

# T A B L E O F C O N T E N T S

## PART I (MC 581)

<u>Section</u>	<u>Subject Matter</u>
0000	Introduction and General Comments on EDPM Systems
0900	Equipment Price Lists
1000	Magnetic Tape Units and Control Units
2000	Random Access Storage Units and Control Units

## PART II (MC 582)

<u>Section</u>	<u>Subject Matter</u>
3000	Input Devices, Except Tapes
4000	Output Devices, Except Tapes
5000	Central Processing Units and Systems Considerations -- Large-Scale EDPMs
6000	Central Processing Units and Systems Considerations -- Small-Scale EDPMs
7000	Auxiliary Equipments
8000	Data Processing Characteristics with EDPMs
8500	Specific Applications Information
9000	Miscellaneous
9900	Index

0000 INTRODUCTION

Our common job--whether we be salesmen, sales support personnel, research engineers, product development engineers, logical designers, applications specialists, programmers, product planners, or administrators--is selling UNIVAC data processing systems. This manual is designed to help you, as the salesmen and sales support staff ultimately responsible for getting the order, do a more effective job. Its sole purpose is to present factual information on our own and competitive EDPMs, together with discussions and evaluations of equipment performance, which will enable us to meet, with the greatest effectiveness possible, the proposals of other manufacturers, and to promote the entire UNIVAC line as not only the first, but also the most efficient, of the electronic data processing equipments.

0001 ORGANIZATION OF THIS MANUAL

The manual is organized into sections dealing with the various types of components of an EDPM system--the input devices, the storage devices, the central processor, etc. With this arrangement, it does not include all comments necessary to compare a Univac II with a 705, for example, in one place; the comparisons are included in the applicable major section which discusses each type of component. The reason for this choice is to eliminate the duplication which would otherwise occur in comparing Univac II with the IBM 705, or the DATAmatic, or the BIZMAC; or the File-Computer with the IBM 650 or the Datatron. Everything said about an IBM tape unit, for instance, applies to any equipment with which it may be associated. Consequently, each type of component is discussed and evaluated in the section pertaining to its general category; the section devoted to central processing units includes not only those factors which are peculiar to the "main frame", but also ties together the many system ramifications possible into one coherent whole and provides an overall evaluation of complete systems.

A straightforward system of decimal index numbering is employed, with major breakdowns indicated in the covering table of contents of this manual. The basic number is four digits in length, any one of which may be sub-indexed with up to three digits to the right of the decimal point. Pages are not numbered sequentially, but each has, in the upper right corner, a large index number which identifies the particular page and its proper sequence. Thus, page 8888.132 follows page 8888.13, which in turn follows 8888.1.

The first, or thousands digit, indicates the major section of the manual. The second, or hundreds, digit is assigned to major manufacturers, although this cannot be carried through all equipment makers. However, a second digit "0" pertains to Remington Rand Univac equipments; a "1" to IBM equipment and a "2" to DATAmatic components. Specific sales pointers are always included in sections with a units digit "9"--i.e., "1109."

Every effort has been made to cross-reference the manual when a knowledge of material contained elsewhere is necessary or desirable for a complete understanding of a topic. To avoid the necessity of numerous correction sheets as information is added, deleted, or changed, such cross-references

0001 (Continued)

usually refer to basic section numbers. Thus it should be understood that a reference to "section 1120" includes the range "1120" through "1129," and a reference to "1122" includes all breakdowns within that index number.

Section 9900 contains a subject matter index of the contents of this manual. An additional index to sections dealing with specific equipments or components is contained in the price lists in section 0900-0999; The entries for each type number or trade name under which equipments and components are known include the manual section in which it is discussed.

The discussion of each equipment or component is broken into three parts: (1) A brief summary of the basic characteristics of the unit under consideration; (2) a discussion and evaluation of the operating features; and (3) sales points pertaining to it. The second part is usually the longest and is developed in sufficient detail to permit a ready understanding of the method of operation of a component and an evaluation of its performance in comparison with the corresponding UNIVAC series equipment. The sales points are largely based upon discussions contained in this second part.

General comments on EDPM processing and information on specific applications is contained in section 8000. This is intended to serve as background information which will be useful in assisting potential customers to gain a proper perspective on the relative performances of various equipments through an understanding of the basic business file processing techniques which are possible with EDPMs of varying general characteristics.

The development of this manual assumes a reasonable familiarity with the characteristics and operations of equipments of Remington Rand UNIVAC. All competitive equipments are explained in a degree of detail sufficient to permit understanding their method of operation; it is intended that these explanations be comprehensive enough to eliminate the need for handbooks or manuals describing their method of operation-- they may not always be available, and frequently are not exhaustive or complete in their explanations.

0002 INTERCHANGE OF INFORMATION

Although every effort has been made to verify the accuracy of all factual and analytical information contained in this manual, it is inevitable that some errors or omissions occur. Any which may be noted should be brought to the immediate attention of the Sales Support Staff, in order that they may be corrected.

Similarly, the development of timely and accurate comparative analyses of equipments or components requires the constant availability of current and complete information on new developments and new areas of

0002      (Continued)

emphasis by competitive firms. Information not present in the manual should be forwarded to the Sales Support Staff as soon as possible, in order that a timely evaluation can be made and "rebuttal" data prepared. This includes not only new equipments or peripheral units, but also new or unusual equipment configurations and processing techniques which may be proposed. Only by knowing of these developments can they be taken into consideration in improving the effectiveness of our UNIVAC sales efforts and in evaluating the impact of new equipments or techniques on the marketability of our own future product plans. The success and usefulness of a manual of this nature is directly correlated to the degree to which the joint efforts of all of us in Remington Rand Univac are co-ordinated to one common end--the widest possible distribution of the accumulated knowledge of every one of us. This manual is intended to provide--at least in part--a medium for accomplishing this.

0900 PRICES OF ELECTRONIC DATA PROCESSING EQUIPMENTS

This section contains the currently-quoted prices for the EDPM equipments and components of leading manufacturers; these are grouped by manufacturer and are in type number or component name sequence, as applicable. Each section includes all EDPMs, or components, which have been or are being made, as well as those which have been announced for future delivery; inclusion does not mean that they are available for current delivery. It also includes type numbers of equipments or components which have been assigned, although the unit may never have been made and may not be planned for production. In general, these are noted.

Where components can be used in more than one basic computing system, notation is made of the possible systems into which it may be incorporated. The notation "Obs." in the column headed "Status," means that the unit is now out of production and no further orders can be taken; most of them are of course still in use.

A question mark preceding the price indicates lack of certainty or contradictory information; these will be corrected as soon as specific and accurate cost data are known.

Some manufacturers assign names or type numbers to such units as power supplies, memory units, etc., although their cost is usually included in that for the assembly with which they are used. These are included in the list for the sake of completeness; in all instances, they are noted to indicate the equipment or component with which used.

0110 IBM ELECTRONIC DATA PROCESSING EQUIPMENTS

The chart on the next page ties together the equipment relationships of the EDPMs offered by the International Business Machines Corporation.

This charts includes all functional components available or announced, but omits many type numbers which are always associated and used with one or more of the equipments shown: these comprise such components as power supplies, memory storage elements, operator's consoles, etc. A complete list of type numbers, with references to the basic EDPM systems with which used, is included in 0910.

0910 PRICES OF IBM EDPM AND PUNCH CARD EQUIPMENTS

Section 0910.1 lists the prices and basic EDPM systems of all known type numbers assigned to this class of equipment. Where available, both rental and purchase prices are included. The listing is in type number sequence, and includes a reference to the section of this manual in which a discussion of the type appears.

The basic processing systems are listed in capital letters; all units not so identified are component or auxiliary parts of one or more of the major systems. Up to the present time, the assignment of type numbers by IBM has been in certain broad classes of components, which permits a partial identification of the nature of the equipment by a knowledge of its type number. This is not, however, universal. The basic groups are:

- 300 Series: Equipments in the RAMAC 305 series
- 600 " : Equipments in the Type 650 systems
- 700 - 709 : Analytical and Control Units, large-scale systems
- 710 - 730 : Card readers, card punches, printers and magnetic tape units
- 731 - 739 : Central processor storage devices--drums and magnetic cores
- 741 - 749 : Power supply and power distribution units
- 750 - 759 : Tape control units
- 760 - 779 : Special purpose tape control units
- 780 - 789 : Operator's consoles

In the past several months, IBM has adopted a general policy of pricing equipments by type numbers; for example, the Analytical and Control Unit of the 705 is priced at \$11,800 monthly (Model 1), although it requires a Type 745 Power Supply Unit (at \$1,200) and a Type 782 Operator's Console (at \$1,000) to be operative. Insofar as practicable, this list groups associated components under the major one and prices the total; the individual type numbers are separately enumerated in the "Description" column, and also appear in proper sequence. An asterisk following a price indicates a unit which is a fundamental part of some other type number.

An "x" keying a type number to a major system indicates a component that can operate under central processor control; it may also be capable of periphery or off-line operation. A "y" key indicates a unit that can be used only in off-line operation in that system.

The discussions of major systems, sections 5000 and 6000, contain a complete description of the various system configurations which are possible and also include examples of what may be considered typical installations of such major systems.

The prices of some punch card equipments manufactured by IBM are listed in 0910.2. This is not exhaustive. It is included for reference purposes in arriving at costs of punch card equipment necessary in an EDPM installation.

MONTHLY COSTS OF IBM EDPM EQUIPMENT

TYPE	MODEL	650	701	702	704	705	709	305	STATUS	DESCRIPTION	COST		SEC-TION
											RENTAL	PURCHASE	
305								x		CENTRAL PROCESSING UNIT, RAMAC 305 Type 305      \$1,250      \$86,900 Type 340      375      26,100	\$1,625	\$113,000	6150
323								x		Card Punch	225	14,600	
340								x		Power Supply Unit for Type 305	375*	26,100*	6150
350								x		Random Access Disc Memory	650	48,880	2110
355		x								Random Access Disc Memory	975	72,200	2115
370								x		Line ("Stick") Printer	300	19,800	4140
380								x		Operator's Console with Card Reader and Input- Output Typewriter	400	27,800	6150
475										Early version of Type 370 Printer	---	---	---
533		x								Card Read Punch Unit (dual feed)	550		3170
537		x								Card Read Punch Unit (single feed)	700		3175
650										MAGNETIC DRUM PROCESSOR			
		x								1,000 Word Capacity      \$1,900      \$100,700	3,250		6100
										2,000 Word Capacity      2,400      127,000	3,750		6100
										Type 655      1,100      58,300			
										Type 650 Optional Features (no type numbers):			
		x								Floating Decimal Point Feature	? 300*		6111
		x								Indexing Registers (3)	? 240*		6112
		x								Printer Input-Output Device for use with Type 407 tabulator (numeric only)	500*		6113
		x								Alphabetic Device for use with Type 407	325*		6113
		x								"11" and "12" Punch Device for use with Types 407 and 533	25*		6113
		x								Special Character Device (all including "11" and "12") for use with Types 407 and 533	100*		6113
652		x								Magnetic Tape and Random Access Control Unit	975-	55,600-	1120
653		x								Record Storage Unit	1,950		1125
655		x								Power Supply Unit for Type 650 with one Input- Output Synchronizer	2,425		1125
											1,100*	58,300*	6100

Monthly Costs of IBM EDPM Equipment (cont'd)

TYPE	MODEL	650	701	702	704	705	709	305	STATUS	DESCRIPTION	COST		SECTION
											RENTAL	PURCHASE	
701			x						Obs.	Additional Synchronizers (each)	\$ 300*		6100
702									Obs.	CENTRAL PROCESSING UNIT			5110
				x					Obs.	10,000 Characters Electrostatic Storage	8,650		5120
703				x					Obs.	20,000 Characters Core Storage	10,150		5120
704	2								Obs.	See Type 770	---	---	---
					x					CENTRAL PROCESSING UNIT:			
					x					4,096-Word Storage (Type 737 Model 1)	15,800		5130
					x					8,192-Word Storage (Type 737 Models 1 and 2)	20,800		5130
					x					32,768-Word Storage (Type 738)	29,700		5130
					x					Type 704 \$9,700 \$523,800			
					x					Floating Point Feature (additional cost)	4,000*		5130
705	1					x				CENTRAL PROCESSING UNIT, 20,000 CHARACTERS	14,000		5140
										Type 705-1 \$11,800 \$613,600			
										Type 745 1,200			
										Type 782 1,000			
705						x				CENTRAL PROCESSING UNIT, 40,000 CHARACTERS	16,500		5140
										Type 705-2 \$14,300 \$749,600			
										Type 745 1,200			
										Type 782 1,000			
706			x						Obs.	Electrostatic Storage Unit for Type 701	1,600*	\$ 28,100*	5110
708									Obs.	Modified Type 704; superseded by Type 709	---	---	---
709	1									CENTRAL PROCESSING UNIT:			
										4,096 Word Storage (Type 737 Model 3)	16,900	948,400	5150
										8,192 Word Storage (Type 737 Models 3 and 4)	20,100	1,141,400	5150
										32,768 Word Storage (Type 738)	32,900	1,803,600	5150
										Type 709-1 \$10,000 \$600,000			
										Type 741-2 1,400 72,800			
										or Type 741-3 1,600 96,000			
										Type 746 1,300 67,600			
711	1		x	x						Punch Card Reader (125/minute)	200		3110
711				x		x				Punch Card Reader (250/minute)	800	52,000	3110
712				x					Obs.	Punch Card Reader (C-T Converter)	1,050		3120
										Type 712 \$750 \$36,800			
										Type 756 300			
714		y	y	x	y	x	y			Punch Card Reader (C-T Converter)	2,400		3120

0910.1

Monthly Costs of IBM EDPM Equipment (cont'd)

TYPE	MODEL	650	701	702	704	705	709	305	STATUS	DESCRIPTION	COST		SEC- TION
											RENTAL	PURCHASE	
										Type 714 \$1,500 \$97,500			
										Type 759 \$ 900			
715					x					Output Recording Typewriter			
716	1		x	x			x			Alphabetic Printer (150 lines/minute)	\$1,200	\$ 78,050	4110
717		y	y	x	y	x	y			Alphabetic Printer (150 lines/minute)	1,800		4120
										Type 717 \$1,200 \$73,950			
										Type 757 600			
719		y		x	y	x	y			Alphabetic Printer (1000/minute, 60 characters)	1,400	74,200	4130
720		y		x	y	x	y			Alphabetic Printer (500/minute, 120 characters)	1,400	74,200	4130
721	1		x	x			x			Punch Card Recorder (Card Punch) (100/minute)	600	39,000	4510
722		y	y	x	y	x	y			Card Punch (T-C Converter)	1,050		4520
										Type 722 \$750 \$44,400			
										Type 758 300			
726			x						Obs.	Magnetic Tape Unit	850	42,100	
727		x	x	x	x	x	y			Magnetic Tape Unit	550	29,800	1100
729	1						x			Magnetic Tape Unit	650	35,750	1105
730		y		x	y	x	y			Alphabetic Printer (1000/minute, 120 characters)	2,775	149,900	4130
731			x						Obs.	Magnetic Drum and Power Unit	1,400	87,450	
										Type 731 \$1,100 \$57,200			
										Power Supply 300 30,250			
732				x						Magnetic Drum and Power Unit	2,300	121,900	2520
733	1				x		x			Magnetic Drum (8,192 36-bit words)	3,100	167,400	2510
733	2				x		x			Magnetic Drum (faster transfer than Model 1)	3,500	189,000	2510
734				x		x				Magnetic Drum (60,000 characters)	2,800		2520
736	1		x		x					Power Unit No. 1; rental included with CPU	---	---	5130
736	2						x			Power Supply Unit	1,100*	57,200*	5150
737	1				x					Magnetic Core Storage (4,096 words)	6,100*		5130
737	2				x					Magnetic Core Storage (Next 4,096 words)	5,000*		5130
737	3						x			Magnetic Core Storage (4,096 words)	4,000*	208,000*	5150
737	4						x			Magnetic Core Storage (Next 4,096 words)	3,200*	193,000*	5150
738	1				x		x			Magnetic Core Storage (32,768 words)	20,000*	1,040,000*	5150
740	1		x		x		x			Cathode Ray Tube Output Recorder	2,700	162,000	4910
741	1		x		x					Power Supply Unit; rental included with CPU	---	---	5130
741	2						x			Power Supply Unit	1,400*	72,800*	5150
741	3						x			Power Supply Unit	1,600*	96,000*	5150
742				x						Power Supply Unit for Type 732; rental included with it	---	---	2520

CTC

Monthly Costs of IBM EDPM Equipment (cont'd)

TYPE	MODEL	650	701	702	704	705	709	305	STATUS	DESCRIPTION	COST		SEC-TION
											RENTAL	PURCHASE	
743				x						Power Supply Unit; rental included with CPU	---	---	5120
744		x								Power Supply Unit; rental included with 650	---		6100
745						x				Power Supply Unit	1,200*		5140
746	1		x		x				Obs.	Power Distribution Unit	1,300*	67,600*	5110
746	2							x		Power Distribution Unit	1,300*	67,600*	5150
746	3							x		Power Distribution Unit	1,300*	67,600*	5150
747		y		x	y	x	y			Power Supply Unit for Type 774	500*		1170
752				x						Tape Control Unit	550	28,600	1130
753			x		x				Obs.	Tape Control Unit	2,500		1135
754				x		x				Tape Control Unit	2,000	104,000	1140
755	1							x		Tape Control Unit	1,800	108,000	1145
756				x						Control Unit for Type 712	300*		3120
757		y		x	y	x	y			Control Unit for Type 717	600*		4120
758		y		x	y	x	y			Control Unit for Type 722	300*		4520
759		y		x	y	x	y			Control Unit for Type 714	900*		3120
760		y		x	y	x	y			Record and Storage Unit	1,850		1150
766	1							x		Data Synchronizer	3,500	210,000	1160
766	2							x		Data Synchronizer	3,500	210,000	1160
770									Obs.	Special Sorter, originally known as Type 703, built for National Security Agency; no commercial plans	---	---	---
771										Data lacking	1,280	83,200	
772										Data lacking	570	37,100	
774		y		x	y	x	y			Tape Data Selector	2,200		1170
										Type 774	\$1,700	\$102,000	
										Type 747	500		
775				x						Record Storage Unit; announced but never delivered	1,850	111,000	
777				x		x				Tape Record Coordinator	3,000	156,000	1180
780	1							x		7-inch Cathode Ray Display Tube	150	8,700	4910
781				x					Obs.	Console and Typewriter; rental included with Type 702	---	---	---
782						x				Console and Typewriter	1,000*		5140
795									Obs.	Special purpose machine built at Northrop Aircraft. no commercial plans	---	---	---
797									Obs.	Core storage version of Type 795; IBM built four but has no commercial plans	---	---	---
838		x								Inquiry Keyboard in RAMAC 650	175	10,000	7110

0910.1

0910.2 PRICES OF IBM PUNCH CARD EQUIPMENT

Following is an incomplete list of IBM punch card equipments, with monthly rentals and sales prices. It does include nearly all of the units which may be required as auxiliaries to an EDPM installation. Prices shown are probably for basic equipments without any of the optional devices which may be added. Some units are indicated with two or three different cost figures; these may represent different models or versions of the same basic equipment, but are not otherwise explained in the partial information from which this list has been compiled. More accurate costs will be provided when available.

<u>Type Number</u>	<u>Equipment</u>	<u>Cost</u>	
		<u>Rental</u>	<u>Purchase</u>
010	Key Punch	\$ 10	\$ 700
016	Duplicating Key Punch	40	1,950
024	Alphabetic Key Punch	35	1,750
026	Printing Key Punch	60	3,700
	" " "	55	3,500
027	Punch Card Proof Machine	145	8,000
028	Printing Punch Card Proof Machine	160	9,500
043	Tape-Controlled Card Punch	70	3,000
046	Tape-to-Card Punch	140	8,400
	" " "	135	8,100
047	Tape-to-Card Printing Punch	160	9,600
056	Verifier	50	2,400
065	Data Transceiver Card Unit	90	4,700
066	Printing Data Transceiver Card Unit	110	6,450
067	Telegraph Signal Unit (Transceiver)	85	4,950
075	Card Counting Sorter	70	4,550
077	Numeric Collator	100	6,600
080	Sorter	40	2,600
	"	25	2,565
082	Sorter	55	3,400
083	Sorter	110	6,550
086	Sorter	100	6,900
089	Alphabetic Collator	220	14,300
101	Statistical Sorter	500	30,000
402	Alphabetic Electric Accounting Machine	420	27,950
	" " " "	395	27,450
403	Alphabetic Electric Accounting Machine	470	31,150
407	Alphabetic Electric Accounting Machine	835	54,250
	" " " "	920	59,850
419	Numeric Electric Accounting Machine	425	26,550
513	Reproducing Punch, Mark-Sense	40	1,350
514	Reproducing Punch, Mark-Sense	125	7,500
	" " " "	110	6,600
519	End-Printing Reproducing Punch, M/S	135	7,650
	" " " "	120	6,850
	" " " "	85	5,150
523	Gang-Summary Punch	85	5,550
	" " "	80	5,200

## 0910.2 (Continued)

<u>Type Number</u>	<u>Equipment</u>	<u>Cost</u>	
		<u>Rental</u>	<u>Purchase</u>
526	Printing Summary Punch	\$ 100	\$ 5,900
	" " "	95	5,700
528	Accumulating Reproducer	440	26,400
549	Ticket Converter	250	12,500
552	Alphabetic Interpreter	75	4,650
557	Alphabetic Interpreter	165	9,400
602	Calculating Punch	245	15,900
	" "	220	26,000
604	Calculating Punch	400	26,000
	" "	510	37,100
605	Calculating Punch (part of CPC)	615	40,000
607	Calculating Punch	650	35,800
608	Transistorized Calculating Punch	1,600	
803	Bank Proof Machine	135	10,125
824	Typewriter Card Punch	95	4,700
826	Typewriter Card Punch	115	5,900
916	Bill Feed	50	1,950
923	Tape-Controlled Carriage	30	1,950

PRICES OF DATAMATIC CORPORATION EDPM EQUIPMENTS

The DATAmatic Corporation is wholly-owned by Minneapolis-Honeywell.

<u>Model</u>	<u>Description</u>	<u>Cost</u>		<u>Section</u>
		<u>Purchase</u>	<u>Rental</u>	
1000	Central Processing Unit, including Model 1090 Console and facilities for controlling up to 8 tape units	\$985,000	\$21,500	5410
1052	Auxiliary Central Processor; controls up to 2 additional tape units	100,000	2,200	5410
1054	Same; controls up to 4 additional tape units	175,000	3,900	5410
1056	Same; controls up to 6 additional tape units	250,000	5,500	5410
1058	Same; controls up to 8 additional tape units	295,000	6,600	5410
1090	Operator's Console; cost included with Model 1000	---	---	---
1100	Magnetic File (Tape) Units	60,000	1,350	1400
1150	Magnetic File Reference Unit, including typewriter	50,000	1,075	7410
1170	Magnetic File Switching Unit	9,600	190	1410
1200	Input Converter (Card-to-Tape Converter); includes card reader but not magnetic tape unit	185,000	3,325	3410
1300	Output Converter (Tape-to-Card or Tape-to-Printer). Does not include tape unit, card punch or printer.	100,000	1,800	4410
1400	Output Converter and high-speed printer. Includes converter and printer, but not tape unit	215,000	4,300	4420

PRICES OF BURROUGHS CORPORATION EDPM EQUIPMENTS

Prices of equipments manufactured by Burroughs Corporation and its Electrodata Division are included. It is understood that all future development of EDPMs will be under the Electrodata Division.

<u>Model</u>	<u>Description</u>	<u>Cost</u>		<u>Section</u>
		<u>Purchase</u>	<u>Rental</u>	
<u>Model E101 Data Processor:</u>				
E101	E101 Computer, with keyboard input semi-ganged printer, 100-word magnetic drum, two pinboard sets	\$ 32,500	\$ 850	6510
	Additional Pinboards	25	---	---
E101	Additional 120 words of drum capacity	3,000	80	---
E101	Paper Tape Reader (20 digits/second)	3,500	95	3610
E101	Paper Tape Punch	3,500	95	4710
<u>DATA TRON Data Processing System:</u>				
204	Basic Datatron Central Processor, including magnetic drum and arithmetic unit	135,000	3,900	6520
205	Magnetic Drum, part of Model 204 and cost included with it	---	---	---
360	Floating Decimal Arithmetic Unit for use with Model 204 Datatron	21,100	725	6520
403	Console, for use with Model 204	7,050	230	6520
405	Consollette (Remote Control), for use with Model 204	1,980	70	6520
406	Console, with photoelectric perforated tape reader and tape punch	14,211	451	6520
409	Console, with photoelectric perforated tape reader (no tape punch)	11,231	362	6520
---	Photoelectric Reader (used with Models 406 and 409)	2,960	95	3620
---	Paper Tape Perforator, used with 406	2,040	61	4720
---	Paper Tape Perforator, high-speed	2,980	89	4720
446	Typewriter (Flexowriter) Control Unit	4,560	137	7620
454	Paper Tape Preparation Unit	?	?	7610
458	Flexowriter, Program-Controlled	3,135	95	7620
500	Punch Card Converter Unit	18,625	567	3450
506	Cardatron:			
	Control Unit and Power Supply	31,000	770	3460
	Input Station	22,500	560	3465
	Output Station	26,300	660	4610
543	Magnetic Tape Control Unit	25,000	750	1520
544	Datareader (Magnetic Tape Unit)	12,000	375	1510
560	Datafile (Multiple Magnetic Tape Bin)	25,000	825	2510
<u>IBM punch card equipments usable with Model 204 Datatron:</u>				
	500 series summary punches (input/output)		60-455	
	089 Collator		190-220	
	400 series Tabulators		205-800	

PRICES OF LOGISTICS RESEARCH, INC., EDPM EQUIPMENT

These prices include only those of components used with the ALWAC 800 EDPM equipment.

Model	DESCRIPTION	COST		Section
		Purchase	Rental	
801	Central Processing Unit (includes 1,000 words of core memory)	\$325,000	\$11,580	5510
810	Magnetic Drum Buffer Unit	28,000	950	2850
815	Magnetic Drum	34,000	1,150	2850
820	Punched Card Buffer Unit	53,000	1,530	4850
840	Magnetic Tape Buffer Unit	25,000	850	1510
845	Magnetic Tape Transport Unit	18,000	610	1520
860	Paper Tape and Keyboard Buffer Unit	15,000	510	3510
865	Electric Keyboard	5,000	170	7510
866	High-Speed Paper Tape Reader	1,500	85	3520
867	High-Speed Paper Tape Punch	1,500	40	4860
881	Additional 1,000-Word Magnetic Core Storage for Type 801 (each)	75,000	2,550	5510
882	Address Modifier Unit for Type 801	6,500	220	5510
885	Floating Point Arithmetic Control Unit	23,000	740	5510

1100 IBM TYPE 727 TAPE UNIT

The Type 727 is the standard tape unit used with the Types 650, 702, 704 and 705 EDPMs and computer systems, both for on-line and off-line operations. A variety of control units, listed in 1102, is provided for various purposes. The Type 727 always requires a control unit to be used.

1100.1 Monthly rental: \$550.

1101 TAPE UNIT AND MAGNETIC TAPE CHARACTERISTICS

Essential Specifications are listed below.

Magnetic Tape and Reel

Tape base:	Acetate (scheduled to be replaced to mylar)
Tape width:	$\frac{1}{2}$ "
Tape length:	200, 600, 1200 or 2400 feet (standard)
Tape reel diameter (standard):	$10\frac{1}{2}$ "
Tape reel weight (empty):	12 ounces
Tape weight:	1 ounce per 100 feet (acetate base)
Plastic container diameter:	11 $\frac{9}{16}$ inches
Plastic container thickness:	1 $\frac{7}{16}$ inches
Tape cost, including reel:	2400 feet of tape -- \$50 1200 feet of tape -- \$32 600 feet of tape -- \$22

Type 727 Tape Unit

Number of recording channels:	7, each group of 7 bits recorded in parallel
Character recording density:	200 characters per inch
Tape speed: forward	75 inches per second
rewind	500 inches per second, except last 400 feet (75 inches/second)
Character transfer rate	15,000 per second
Tape start time:	10 milliseconds
Tape stop time:	6.4 milliseconds maximum
Tape reversal time:	60 milliseconds
Bad spot detection:	None: tape must be perfect
Detecting beginning of tape:	Photoelectrical sensing of reflective spot
Detecting end of tape:	Same, or tape mark
Method of recording:	Non-return-to-zero; every valid character must have a binary "1"
Length of tape record:	Variable, subject to restrictions imposed by CPU characteristics
Rewind time:	Maximum of about 1.2 minutes for any length of tape
Check bit:	Normally even. Odd when recording binary information on 704
Setting of tape unit addresses:	Rotary switch on each unit

1102 CONTROL UNITS AND USES OF TYPE 727

The various configurations of control units and basic computer systems or periphery operations with which the 727 may be associated are listed below. Refer to writeups on the various control units or periphery functions for more detailed information.

Type 650 EDPM:	652 Control Unit, maximum of 6 - 727s
Types 702/705 EDPM:	754 Control Unit, maximum of 10 - 727s
	760 Control Unit, maximum of 2 - 727s
	774 Control Unit, maximum of 1 - 727
Type 705 EDPM:	777 Control Unit, maximum of 8 - 727s
Type 704 EDPM:	755 Control Unit, maximum of 10- 727s

## Periphery Operations:

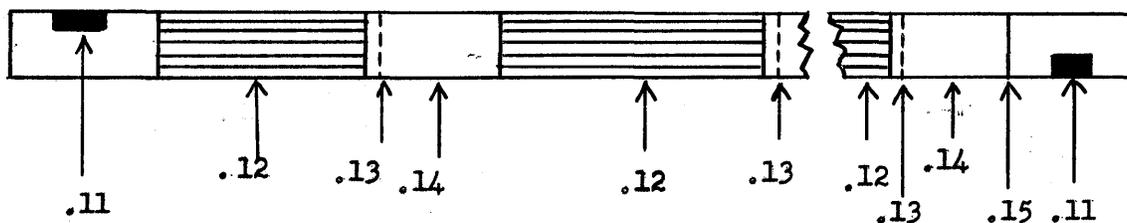
Card-to-Tape Conversion:	712 - 756 - 727	
	714 - 759 - 727	
Tape-to-Card Conversion:	727 - 758 - 722	
Printing:	727 - 757 - 717	
	727 - 760 - 719)	
	727 - 760 - 720)	One or two tape units
	727 - 760 - 730)	may be connected
Punch Card Operations:	727 - 774 - 402 )	
	727 - 774 - 402A)	Tabulators
	727 - 774 - 407 )	
	727 - 774 - 519	Reproducing Punch

1103 MAGNETIC TAPE FORMATS WITH THE TYPE 727

Although the 727 is used with all current IBM computers, the appearance of information stored on tape is a function of the particular equipment which created it. Tapes are nominally interchangeable between and among systems, provided the restrictions or requirements of each are met. Use of this tape unit with the Tape 701 EDPM is not discussed.

1103.1 Common Features of Tape Information Storage

Regardless of the initiating computer, or tapes resulting from C-T conversion, all uses of Type 727 tapes have some common features in the recording of information and tape handling. All specifications of 1101 are applicable to any use. The basic tape layout is shown below, the numbers attached to the arrows indicating the sub-paragraph in this section in which the feature is discussed.



- 1103.11 Reflective Spot: Adhesive aluminum-coated strips, are placed several feet from each end of the tape to indicate the physical beginning and ending points for recording information. The spots are about  $\frac{1}{4}$ " x 1" in size and are placed in the positions indicated, allowing enough footage to provide for the vacuum loop controls and two or three turns around the hub of the tape reel. A minimum of about six feet is required on each end, but 10-15 are more normally used. The spots can be moved quite readily by the user. In both reading and writing, the tape is positioned just past the beginning reflective spot as a part of the loading operation; this is automatically done by the 727 through a photoelectric cell sensing device. It is impossible to read or write on the end of the tape past this spot. In writing only, the end-of-tape reflective spot is similarly sensed and a program-testable signal is generated. Unlike the beginning-of-tape spot, however, it is possible to write beyond the end-of-tape spot. Sensing of the end-of-tape spot occurs only during tape writing; the circuitry is inactive during reading. (See 1105.15).
- 1103.12 Records are written as a variable number of characters, partly dependent upon characteristics of the EDPM being used. Recording is in seven channels, all seven bits being handled in parallel. The record consists of 6-bit information codes plus a check bit.
- 1103.13 Check Character. Each record is followed by a check character, automatically created by the equipment in all input/output operations. In both reading and writing operations, the number of binary "1s" on each of the seven channels is counted and the check character created to make the total number of "1s", including the check character, an even number. Physically, it is separated from the last information character by a space equivalent to three character positions (0.015"). These three "blank" spaces are identical to a recording of seven binary zeros and thus are

## 1103.13 (Continued)

identical to the inter-record gap (1103.14); consequently, every information character must contain at least one "1." The sole use of the check character is to verify the accuracy of reading information previously recorded on tapes. During writing, the character is created and placed on tape at the end of the record; during reading, the character is created while the record is being read and the recreated check character compared with the original. The entire process is automatic and requires no attention by the programmer.

1103.14 Inter-Record Gap. The physical space on tape separating successive records, is  $3/4$ " , from the check character of the previous record to the first information character of the next. The variable lengths of information records have no effect on the length of the gap.

1103.15 Tape Mark. The last record written on a tape is followed by a single special character (0 00 1111), called the tape mark, which is placed there by a special instruction; each system has such an instruction. In writing a full reel, the end-of-tape reflective spot normally is passed over while writing a record, and the tape mark thus is closer to the physical end of the tape than the spot. It may be many feet past it, because sensing the reflective spot does not disable the writing circuits, and several records may be placed on the tape after the spot is reached. On partially-filled reels of tape, the tape mark is placed after the last information record, and is separated from it by an inter-record gap. In reading a tape, sensing of the tape mark generates a program-testable signal indicating the end of recorded information on the reel; this is the only method (other than special program-generated record counts) of recognizing this condition.

1103.2 TAPE RECORD FORMATS OF IBM EDPMS

The only variations of tape recording occur in the records themselves; the reflective spots, check characters and inter-record gaps are common to all equipments. Figure 1103.2 shows the types of record formats possible with various equipments, together with the data in memory which represents the record. The check character is not shown; however, it should be remembered that it follows every tape record, regardless of generating equipment.

The formats contained in Figure 1103.2 are discussed in more detail beginning in 1103.3

1103.3 TAPE FORMAT WITH TYPE 650 EDPM

Two basic formats are possible, one for strictly numeric and one for alphanumeric or mixed information.

- 1103.31 Numeric Information. Numeric data, contained in memory as 7-bit biquinary coded digits in words of 10 digits plus sign, are carried digit-for-digit on tape in standard 702/705 character codes, not in 650 codes. The conversion in either direction is performed automatically in the Type 652 Tape Control Unit. On the basis of read/write times contained in 650 instruction manuals, it appears that the sign digit is not written as a separate character, but is included as a zone code in the units position of each 10-digit word on tape; this is consistent with the 702/705 method of handling signs. Changing of the sign digit to a zone code, and vice versa, is done automatically as a part of input/output operations.

Any number of words up to 60 maximum (600 digits plus 60 signs) may be placed on tape as one record. Numeric tape records used in 650 operations consist, then, of anything from 10 to 600 tape characters in multiples of 10; partial words cannot be written or read. Both reading and writing instructions must specify the exact number of words involved; the exact number specified are always written out and, in reading, if the number of words in the actual record is greater or less than that specified, an error indication results.

- 1103.32 Alphabetic or Mixed Information. Internally in the 650, alphanumeric information requires two biquinary coded digits for each character; one word can be ten digits or five alphanumeric characters; the sign position is disregarded in an alphanumeric word. Each word is an individual entity; strictly numeric words can be mixed with alphanumeric ones.

Alphanumeric or mixed information can be placed on tape only in multiples of ten words: 10, 20, 30, 40, 50 or 60 (maximum). An alphanumeric word in 650 memory is converted automatically to five 702/705 character codes when written on tape, and conversely when read. Alphanumeric and strictly numeric words may be mixed, but any one word must be one or the other.

In 650 memory, the last (tenth) word of each ten-word group is a control word designating the status of the previous nine words. An "8" in a control word digit position indicates that the corresponding word of the group is alphanumeric; a "9," that it is purely numeric. The tenth digit of the control word is always a "9," because it designates the numeric status of the control word itself. Any or all of the remaining nine information words may be alphanumeric. (Control words are obviously not required for strictly numeric information).

In writing on tape, the control word precedes the nine information words, which (presumably) follow in sequence, and is used to control the method of writing out the nine information words. A word with a control digit "8" is written as five (alphabetic) 702/705 character codes; with control digit "9" as ten (numeric) codes, identical with 1103.31. This use of the control word to govern the method of writing out the ten-word group is automatic. In the following example, the nine information words are

identified as "A" for alphanumeric and "N" for numeric; all digits of the control word are shown.

650 memory:	<u>A A N N A N A N A</u>	<u>8899898989</u>
	<u>Information Words</u>	<u>Control Word</u>
Codes on tape:	8899898989	A A N N A N A N A
No. characters:	<u>10</u>	<u>5 5 10 10 5 10 5 10 5</u>
	Control Word	Information Words

The maximum number of alphanumeric characters is a ten-word group is 45 (nine words of five each). The maximum number of alphabetic characters in a single tape record of 60 memory words is 270 (45 x 6). This requires 330 tape characters, because the control words are written out in numeric form.

In reading from tape, the control word of each ten-word group is read first and governs the automatic reconversion of the 702/705 character codes into the correct 650 codes. The position of the control word is changed to the tenth word of each group.

As is the case with numeric information, read and write instructions must specify the exact number of words (which must be multiples of 10) involved; otherwise, errors result.

1103.4 TAPE FORMAT WITH TYPE 702 EDPM

Information character codes used in the 702 are identical with the tape characters. Tape records may be any length from one character up to one less than memory capacity (9999 or 19999). The writing of a tape record (with or without a buffer in the control unit) proceeds character-by-character beginning at the addressed position and continuing until a special character code, called a record mark (1 01 1010), denoted by "†," is reached in memory. The record mark is not placed on tape, but sensing its presence initiates the end of the writing operation (with, of course, writing of the check character and creation of the inter-record gap).

In reading, recognition of the inter-record gap (actually, the three blank positions preceding the check character) terminates the read operation. A record mark is automatically inserted in memory in the position following the last information character read in. Figure 1103.2 will clarify the relationship of the information in memory with the corresponding tape record.

By a special provision in the address of the "write" instruction, it is possible to write out all of memory, record marks included, beginning at any addressed character position and ending in the next-to-last character position (memory address 9998 or 19998), as one tape record. On reading in such a record, record marks in the tape record are treated as any other valid character code. However, at the end of the operation a record mark is automatically inserted in memory following the last information character. If the record was a "memory dump," beginning at address 0000, the record mark would be inserted in 9999 or 19999; this is the reason the last character position is not written out when this special "write" provision is used.

It is important to note that a record mark in memory always terminates a "write" operation on the 702 (with the exception noted above), but it does not appear on the tape record itself. If a record mark is included as a character code in a tape record (as in a memory dump or on a tape prepared on the 705), it is treated as any other valid character; it does not end a reading operation. A record mark is still placed in memory following the last character read in; this is the one generated by sensing the inter-record gap.

Two types of formats are possible with the 704, one for pure binary information and one for character-coded (702/705 codes).

1103.51 Binary Tape Records. The 36-bit binary internal word is represented on tape as six characters of six bits, plus check bit, each. Any number of words may be included in one tape record, but the total number of tape characters normally must be an integral multiple of 6--i.e., full computer words. Unlike the other IBM systems, there is no theoretical limit to the length of a tape record, except the physical length of the tape itself.

There is one other difference in this method of recording. Because long strings of binary zeros are possible in data, an even check bit used in the other systems, could easily result in several successive tape characters of nothing but binary zeros. These characters would be identical with the "blank" spaces indicating the end of each tape record; thus, in a reading operation, their presence would be taken as indicating the end of a record. To overcome this difficulty, an odd check bit, assuring a binary "1" in every information character, is used in binary recording. The check character, however, is based on the even channel bit count used in the other equipments.

1103.52 Character-Coded Tape Records. Although binary in its arithmetic operations, the 704 can contain information in coded form internally; these can be written out as standard 702/705 character codes. It is not necessary to write them out in this fashion, however--they can be handled as if they were binary data. The character codes used in the 704 differ from those of the 702/705 in three respects:

- (1) Decimal zeros are 00 0000 in the 704 and 00 1010 (plus check bit) in the 702/705; they must be converted.
  - (2) All alphabetic zone codes used in the 704 differ from those of the 702/705 and must be converted
  - (3) An even, not odd, check bit must be inserted on tape characters.
- The 36-bit, 6-character coded word in the 704 is written out as six tape characters, in 702/705 codes, each with an added even check bit. Successive groups of six bits are converted automatically to the proper 702/705 character codes. Like binary recording, each tape record normally must be an integral multiple of six characters, but can be of almost unlimited length.

1103.53 Record Lengths Not a Multiple of Six Characters. The 704 writes tapes only in full words--that is, 6 groups of six binary digits or six coded characters. However, it can read a record which is not an even multiple of six tape characters, provided its exact length is known. The requirement of known length is necessary because the method of transferring the last partial group of bits is different from transferring a full group. Evidently, tapes so read must have been prepared on a 650, 702 or 705.

1103.6 TAPE FORMAT WITH TYPE 705 EDPM

Essentially, tape records used with the 705 follow the principle of those used with the 702 (1103.4), with one important addition: The record mark ( $\neq$ ) of the 702 is treated in input/output operations as any other valid character code, and a new code, called the group mark (0 11 1111), designated by " $\neq$ ", serves exactly the same function as does the record mark in the 702. The group mark is not a valid character in the 702, but because it does not appear on tape, a 705-created tape can be used as input to a 702 (or vice versa).

- 1103.61 Record. In IBM terminology, a record is a group of characters, variable in length, terminated in memory by a record mark. On a 702-created tape, the record mark is replaced by the inter-record gap; on a 705 tape, it appears as a valid character code and does not terminate the writing operation.
- 1103.62 Grouped Record. In the 705, several records, separated by record marks, may be treated as a grouped record, terminated in memory by a group mark and on tape by an inter-record gap. In the 705, the record marks terminating each (except the last) record also appear on tape. Grouped records cannot be written on the 702, but they can be read; in this case, a record mark (not a group mark) is inserted following the last character in memory. The group mark is peculiar to the 705.

1103.7 TAPE FORMAT WITH TYPE 709 EDPM

Two types of formats are possible with the 709.

1103.71 Binary Tape Records. Recording is identical with the Type 704. See 1103.51.

1103.72 Character-Coded Tape Records. Recording is identical with the Type 704. See 1103.52.

1103.73 Record Lengths Not a Multiple of Six Characters. Like the 704, the 709 writes tape records, in either method of recording, only as an integral multiple of six characters. Unlike the 704, it can read a record which is not an even multiple of six characters without advance knowledge of the length; the 704 must know the exact length to read such records. The partial group of characters (five or less) is automatically assembled in the high-order (left) end of the 709 word before transmission to memory. Such tapes must of necessity have been prepared on a Type 650, 702 or 705.

1104 ERROR AND SIGNAL INDICATORS PERTAINING TO TYPE 727 USAGE

All systems using Type 727 tape units incorporate checking of tape status and accuracy of read/write operations. There are several basic conditions to be taken into account. Each of them can cause an automatic stop of the EDPM, but some of them can be set for program interrogation and control.

- 1104.1 Tape Unit Not Ready. This condition arises when no tape unit has been set to the address selected, or when a valid tape unit is in what IBM calls a "not ready" status--tape rewinding, no tape reel mounted, or mounted but not yet at the load (starting) point.
- 1104.11 Type 650 error indication not known. Presumably, the computer stops.
- 1104.12 Types 702/705 stop automatically, with the "I/O No Response" neon on the operator's console lighted.
- 1104.13 Type 704 indication not certain, but apparently the computer stops, with the "Read/Write Check Indicator" turned on.
- 1104.2 Read/Write Character Code Errors. In writing, each character code is given a redundancy bit test (except in the 704, where a check bit is merely added to each six memory bits), and the check character is created. In reading, each character is tested and also the check character is re-created and verified against that on the tape. These operations are automatic. Error indications vary.
- 1104.21 Type 650. The "Branch No Tape Signal" conditional transfer instruction permits a conditional transfer to be made if either a tape error or end-of-file (tape) condition exists. This can be followed by a "Branch No End of File" instruction, which distinguishes between the two conditions and causes a program transfer to the appropriate sub-routine.
- 1104.22 Types 702/705. An error in either reading or writing sets the "Read/Write Check Indicator." This is an addressable indicator (address 0902) which, by the setting of a console switch, may be handled either as an automatic machine stop or a program-controlled correction. In the latter case, a conditional transfer instruction on the status of the "Read/Write Check Indicator" provides a branching if an error has been made; otherwise, the transfer address is ignored and the normal program continues.
- 1104.23 Type 705. A character code error turns on the "Tape Check Indicator," on the console, and makes available a program-testable signal. This indicator does not directly cause a machine stop, but the conditional transfer instruction testing it can be branched to an automatic stop if desired. In addition to the validity test, this indicator is turned on if a "read" instruction in binary mode is used for a character-coded tape, or vice versa.
- 1104.24 Type 705. The "Read/Write Check Indicator" may be tested by the "Transfer Any" instruction of the 705, as well as by direct interrogation. This instruction is not included in the 702.
- 1104.3 End-of-Tape. On writing, sensing of the reflective spot signifying approach of the physical end of the tape generates a testable signal. This condition never results in an automatic stop.

- 1104.31 Type 650. The "Branch End of File" instruction causes a conditional transfer to be made when the reflective spot is sensed.
- 1104.32 Types 702/705. An "Input-Output Indicator" is turned on and can be used in a conditional transfer instruction. A separate indicator is provided for each tape unit and selecting a unit automatically makes the indicator available.
- 1104.33 Type 704. The "Tape Check Indicator" is turned on and may be program-tested. Apparently, the existence of an end-of-tape condition can be distinguished from a writing error only by time of occurrence.
- 1104.34 Type 705. The "Input-Output Indicator" may be tested by the "Transfer ANY" instruction.
- 1104.4 End-of File. On reading, the tape mark rather than the reflective spot signifies the end of information on a tape. The indications are the same as end-of-tape: 1104.3; the only difference is the means used to turn them on. The programmer, of course, knows which tapes are used for reading and which for writing, and can readily provide for the proper tape-switching subroutines to be called up by the conditional transfer instruction.
- 1104.5 Incomplete Word on Type 704. An attempt made to read, in the normal manner, a partial group of less than six characters turns on the "Tape Check Indicator." This condition can arise only on the final characters of records on tapes prepared on a 650, 702, or 705, where records are not necessarily in multiples of six characters. Refer to 1103.53 for 704 input records not a multiple of six characters in length.
- 1104.6 Incorrect Number of Words on Type 650. Both read and write instructions on the 650 specify the number of words involved. Because it always writes the number of words specified, no error results in this operation. On reading, if a tape record has fewer or more words than indicated in the instruction, the machine stops with an error indication on the console. This type of error cannot be program-corrected.

1105 IBM TYPE 729 TAPE UNIT

The Type 729 is the tape unit used with the Type 709 EDPM in central processing operations. Essentially, it is a modified version of the Type 727 (section 1100) and produces magnetic tapes which are completely compatible with those of that unit.

1105.1 Cost: Purchase: \$35,750  
Rental 650 monthly

1106 TAPE UNIT AND MAGNETIC TAPE CHARACTERISTICS

Essential specifications are listed below.

Magnetic Tape and Reel

Identical with those used with Type 727; see 1101.

Type 729 Tape Unit

All specifications for the Type 727 Tape Unit, 1101, are applicable. In addition, one new characteristic is added:

Verification of Tape Writing: Characters are read immediately after being written for check bit validity; this is accomplished by having a set of "read" heads immediately following the "write" heads.

Functionally, this is the only feature of the Type 729 which distinguishes it from the Type 727.

1107 CONTROL UNITS AND USES OF THE TYPE 729

In on-line operations, the Type 729 is used only with the 709 EDPM and not with the other major systems of IBM. The actual linkage is accomplished through a Type 755 Tape Control Unit, which provides power and some control circuitry, connected to one channel of a Type 766 Data Synchronizer. Each channel may have up to eight Type 729s connected through one Type 755. One Type 709 may have up to three Data Synchronizers, each with two channels; if eight tape units are connected to each channel (through a Type 755 for each channel), a maximum of 48 Type 729s may be associated with a Type 709 CPU.

With the exception of its use with the Type 774, it is not known definitely if the Type 729 can be used in lieu of a Type 727 in the periphery (off-line) operations listed in 1102. Because it is functionally identical with the Type 727, there appears to be no reason why it is not interchangeable in periphery operations and this assumption will be made until contradictory information is received.

Sales points on the Type 729 begin in section 1109.50.

1108 PURPOSE OF THE READ-CHECK FEATURE OF THE TYPE 729 TAPE UNIT

The functional characteristic which distinguishes the Type 729 from the Type 727 is the inclusion of the "read-check" or verification of tape writing in the 729. The emphasized purpose of this change is to provide facilities for proving the accuracy of tape recording immediately after the writing. The unemphasized reason is probably a good deal more important.

- 1108.1 Causes of Tape Reading Errors. Several causes can result in failure to read a tape record. From the technical standpoint, the list can be rather long, but from the purely functional viewpoint, the following are the more important in the IBM method of recording on and reading tape:
- 1108.11 Incorrect Spacing of Pulses. Reading IBM tapes requires that the pulses follow each other at specific time intervals, plus or minus a tolerance. This is discussed in 1109.07; the inclusion of the read-check feature in the 729 has no effect on tape reading errors arising from this cause.
- 1108.12 Skew. This also is discussed non-technically in 1109.07. Again, the 729 does not correct read failure arising from skew difficulties.
- 1108.13 Faulty Pulse Recording on Good Tape. Even though a parity bit test is made on the pulses sent to the write heads, it does not follow that the resulting magnetization of the tape is good. The energizing of the heads may be such that the resulting pulse is weaker or stronger than normal, and reading may give an error because the pulse is too weak or too strong (saturated, technically); spots on the tape where the oxide coating is somewhat thinner or thicker than nominal may contribute to this. On all magnetic recording devices, there is a certain amount of inherent background "noise," and the information pulses must be sufficiently strong to rise above this noise to be discernible. On the other hand, if the information pulses are too strong, they may "saturate" the heads and several successive pulses may merge into one indistinguishable group. These conditions may arise even though the tape is considered "good."

In the Univac tape system, the automatic re-read features incorporated in the Uniservos permit varying the "gain" of the signals; this means that we can often successfully read tape pulses which are too weak or too strong to be picked up at normal gain. The IBM tape units do not have this feature and always read at a constant "gain" or amplification. The read-check feature of the 729 can cope with this condition, at least to some extent.

- 1108.14 Flaws on Tape. In use, magnetic tape develops flaws, even though it may have been "perfect" when it left the factory. These "flaws" consist of a variety of different phenomena: Chips in the oxide coating, deposits of grease, dust or non-magnetic particles on the surface, nicks or dents in the tape, which prevent the heads from making sufficiently close contact, etc. Some of these conditions can be removed, but, when present, neither writing nor reading can be done with any degree of reliability. With Univac tapes, such flaws can be "punched out" and passed over by the "bad spot" detection circuits in the Uniservos; IBM does not include this facility in its equipment line.

1108.2 Detection of Errors by the Read-Check Heads. Shortly after being written, the newly-created characters pass under the "read check" heads, which test the pulses for readability and give the character a parity bit test. The read heads are physically located about 1/8" from the write heads, which is equivalent to about 25 character positions on tape, or 1.6 ms in time. The circuitry performing the validity read is deliberately designed to work under conditions more adverse than the normal reading circuits. Any failure to pass the parity bit test is an indication of faulty recording, and generates a signal which can be tested by an instruction in the main program of the 709.

Failure to obtain a valid parity bit does not indicate the cause; it could be a pulse too weak or too strong, a flaw on the tape, or a temporary fault in the checking circuitry itself. The same signal is generated, regardless of the cause. The normal tape-writing instruction sequence of a 709 program can be expected to include a sub-routine, entered only when a read-check error occurs. This routine will attempt to write the tape record again, and repeat the cycle three or four times. If continued errors occur, one of two conditions most likely exists: (1) There is a permanent failure in the tape writing circuits themselves; or (2) there is a flaw on the tape.

Isolating the first cause can be accomplished rather easily by typing out a short message on the operator's console whenever a read-check error occurs. If there is a permanent failure in the writing circuits, there should be repeated failures of this nature, and continued type-outs are a signal to the operator to call in the maintenance engineers.

What course of action is available for the second case--a flaw on tape?

1108.3 Program-Controlled "Passing Over" of Bad Spots. Unless there is a failure in the tape writing circuitry, continued inability to record a readable tape block most likely indicates a flaw on tape. Rather evidently, some means must be provided for "passing over" this bad spot; there is no point in writing a block on a piece of tape known to be bad.

In the 709 system, bad spots on tape can be signalled and detected under complete program control. In other words, a program can be written to accomplish the same thing we do with Uniservos with the bad spot "punch outs." This facility is available because there are two methods of recording on tape, and two write and two read instructions which specify the method used. One of these uses an even check bit; the other, an odd. (See 1103.5). The read instruction must specify the type of recording and an error signal, which can be tested by a program instruction, is generated whenever the parity bit does not agree with that specified. Although IBM has not announced the specific techniques it will advocate, the general approach of this programmed bad spot elimination is quite simple.

1108.31 Bad Spot Elimination With Character-Coded Recording. Suppose a tape is recorded with 702/705 character codes, which have an even check bit. If, in the writing operation, a continued read-check failure indicates the probability of a flaw on tape, this technique provides an easy means of detecting the condition on a subsequent reading operation. The routine

## 1108.31 (Continued)

attempting to re-write the tape block will stop after three or four tries, and move in a sequence of instructions which place on the tape several inches of "hash" recording having an odd check bit (binary recording). In reading this tape, the "read" instruction will specify an even check bit; reaching this "hash" block, with the incorrect type, generates a read-error signal which is tested as a normal part of a 709 tape-reading instruction sequence. Because this type of error cannot be distinguished from other types (for example, simple failure to read validly), some additional checking is necessary. This can be done by making the "hash" block consist of bit combinations which repeat a specific character; for instance, it may be nothing but "\$" symbols repeated for 1,000 characters (five inches on tape). If the input-error sub-routine tests several of the character positions for a "\$" character code, positive identification of a bad spot, rather than some other type of failure, can be made. Several positions normally must be tested, because the mere fact that the "hash" block is written over an imperfect piece of tape means that not all character codes will be readable.

1108.32 Bad Spot Elimination With Binary Recording. The technique for a tape being recorded in binary, with odd check bits, is just the reverse of the above. In this case, the "hash" block would be written with an even check bit, each six-bit group repeating a specific bit pattern--for example, 10 1010. This is detected and tested in exactly the same manner as outlined in 1108.31.

1108.4 Comments on Programmed Bad Spot Elimination in the 799/709. In announcing the incorporation of the read-check heads into the 729, IBM has stressed the fact that they permit verifying the readability and validity of tape characters as soon as they are written, and to correct improper recording immediately. The last thing mentioned--and nicely buried at the end of a paragraph--is this:

"The 729 tape unit, by constantly checking the quality of the recording and the quality of tape surface, achieves more error-free passes, extends the useful life of a reel of tape and increases the number of times a record may be read. Programming can prevent the recording of data on defective sections of the tape. Many marginal tape errors will be eliminated."

This verbatim quote is an interesting admission from a company that has required perfect tape with its EDPMs ever since the first one was produced. By the time the first 729 is in use with this feature, IBM 700 series equipments will have been in use for about five years--with perfect tape. The implications of their about-face are obvious.

1108.5 Comparison of IBM and Univac Methods of Bad Spot Elimination. Now that IBM has admitted that the normal tape is "imperfect," it is pertinent to compare the corrective methods adopted by them and by Univac. The Univac technique is to punch out the bad area, a permanent means of eliminating it from possible recording. The IBM method requires detecting the bad spot, and programming a "hash" block, on each tape pass involving writing.

1108.5 (Continued)

The principal differences between the two approaches are two in number: (1) When the bad spot is detected; and (2) the permanency of the corrective action. In the 729 method, the bad spot is discovered immediately, but is never permanently "blacked out;" the corrective routine is performed every time the tape is used as output. The newly-created tape is, however, good insofar as bad spots are concerned--no information is written on an imperfect piece of tape. In the Univac method, the bad spot is discovered, normally, only when the tape is used as an input, either to the CPU or a periphery equipment. The corrective action, by way of contrast, is done once and is permanent. It is, however, possible to record information on imperfect tape and not discover the fact until much later.

Which method is better? The answer, of course, is the one that is the least wasteful of EDPM time and money. Rather obviously, it is a function of the frequency with which new flaws appear on tape and of the total number which may be present on any one reel. The determination of what constitutes imperfect tape is itself not definitive. For example, mylar-based tape being inspected for Univac II usage averages about ten bad spots or punch outs per reel; IBM, using the same type of tape from the same supplier, is getting perfect reels of comparable length. If nothing else, this indicates that Univac inspection standards must be more rigid.

However, the user is interested in performance rather than inspection standards. So long as IBM tapes give satisfactory results, he is not particularly interested in the rigidity of technical tests. Insofar as bad spots are concerned, the method which results in the least loss of good EDPM time (assuming comparable costs) is the best for the customer. Which method has the least time loss--Univac or IBM? Unfortunately, the data to make a specific answer are lacking. The factors to be considered are nonetheless quite clear.

Because a bad spot on Univac tape is normally discovered only in a subsequent input operation, it can be assumed that a rerun of the job creating the tape is necessary. (It is not often that the erroneous recording can be corrected manually, although this is sometimes possible). Consequently, the time loss per bad spot is that necessary to recreate the tape; this typically will run from 10-15 minutes, including machine setup.

In the IBM method, bad spots must be handled on every tape writing operation, at least; the reading instruction sequences can be designed to eliminate re-reads of "hash" blocks. The time loss is partly a function of the number of characters in the record which cannot be written successfully and partly of the number of times it is written before concluding that the tape is imperfect in the place involved. Suppose that three tries are made at writing the information block--this is probably a reasonable number--and further suppose there are 1,000 characters in the information block. The time required for the three "tries" plus that for the "hash" block is about 950 ms (seven passes over the tape block at 83.4 ms each, plus six tape reversals at 60 ms each). Thus, about one second is required for each bad spot, and this is repeated every time a bad spot occurs.

1108.5 (Continued)

If 15 minutes is taken as the time lost by Univac for a newly-discovered bad spot, a total of about 900 could be recognized and program-eliminated by the 729/709 in the same time. The cost-comparison of the two methods then reduces essentially to this determination: Assuming fairly constant and continuous writing of output tapes, what is the cost of Univac main frame time between discoveries of two new bad spots on tapes versus that of the 709 for the time required to pass about 900 bad spots on tape? This requires knowledge of several factors not now available: (1) What is the average number of bad spots passed through Univac output tapes for each newly-discovered imperfection; (2) what is the average number of bad spots per 729/709 tape reel; (3) what are the comparable costs of the main frame plus associated tape units and controls; (4) what are the relationships between the character transfer rates of the two equipments; etc. Some of these are known, or can be determined, but the first two are lacking. A definitive answer cannot be given until these factors are available.

1109 SALES POINTS ON TYPE 727 TAPES AND TAPE UNITS

This section includes actual or claimed desirable features of the 727 and methods of rebutting them, as well as its weak points and how to capitalize on them. There is no significance to the sequence in which the factors are listed.

- 1109.01 "BAD Spot" Detection. The 727 demands perfect tape; the Uniservo does not.

For the 727: Faster read/write times because there are no "bad spots" to skip over.

Against the 727: The percentage of Uniservo tape which is punched out is usually very small and adds little to tape time. Tapes don't wear out uniformly, but get scattered flaws from such things as fingernails, abrasive dust particles, etc. With Univac tape, continued use is possible; with the 727, two shorter pieces result. It should be noted that splicing is not too effective and seldom practiced.

- 1109.02 Tape Capacity. IBM tapes in 2400-foot lengths and a 3/4" gap can contain more information than a Univac metallic tape. Typical capacities are:

	<u>727</u>	<u>Uniservo I</u>	<u>Uniservo II</u>
Number of 720-character records	6,600	2,000	3,600 plus
Number of characters	4,750,000	1,440,000	2,600,000 plus
Number 80-col punch cards	25,000	5,000	5,000
Maximum normal capacity	5,000,000	1,440,000	2,600,000 plus

Mylar tape on Uniservo II will give single-reel capacities about the same as the IBM tape.

For the 727: Holds almost twice as much information as a Univac II reel at a comparable cost (\$50 a reel against \$51 for Univac tape). Initial tape investment is less than for Univac.

Against the 727: There is no argument on present capacities, but mylar tape for Uniservo II will provide equality. However, note that tape capacity on the 727 is often a function of file format; it may or may not approach the figures given above. On Univac, tape capacity is independent of format. On the cost aspect, the point to stress is the cost per time used, not the initial cost. Both are less than a nickel per tape pass and, with Univac tape's punch-out feature for bad spots, the cost per million characters stored over a long period may well be less than the 727's. At any event, any cost differential--in favor of the 727 or of Univac--is a fraction of a cent per million characters stored.

- 1109.03 Rewind Time. The 727 has a high-speed rewind with an approximate maximum of 1.2 minutes for any tape rewind. The Uniservo does not. Rewind is closely related to tape changing, 1109.04.

For the 727: The fast rewind gives several minutes for tape changes.

Against the 727: It needs the fast rewind, because of the slowness of tape changing. The Uniservo rewind is fast enough for its purposes--namely, to permit tape changing on multi-reel operations without stopping the computer. Part of the higher price of the 727

## 1109.03 (Continued)

is due to the high-speed rewind. As a further note, tapes sometimes break during the sudden acceleration into high-speed rewind and an insecurely mounted tape reel may fly off the mounting, with rather disastrous results to the innards of the tape unit. Needless to say, the tape itself isn't much good after this treatment.

1109.04 Tape Changing. The Uniservo is considered to have a clean-cut advantage. Therefore, only the steps involved in changing 727 tapes are listed, with sales points following.

1. Press "Unload" key and wait several seconds for vacuum columns to unload and tape drive mechanism to open up.
2. Open door and manually rewind the last several feet (usually 15 or more) of tape.
3. Turn pressure band knob several times to loosen reel and remove it.
4. Put on new reel, unwinding several feet of tape, and tighten pressure band knob.
5. Feed loose end of tape through retracted tape drive mechanism, into takeup reel and under the holding spring.
6. Manually wind tape onto takeup reel until reflective spot is well past the heads.
7. Close door and press "Load" and "Ready" keys. In 8-15 seconds, unit is ready.
8. Set address selector switch, if necessary.

Against the 727. There are several weak features.

1. Changing a 727 reel takes  $1\frac{1}{2}$ -2 minutes, compared with  $\frac{1}{2}$  minute for Uniservo. This is one reason the 727 needs a high-speed rewind.
2. The reel may not be seated firmly against the back-plate of the mount. One edge of the tape then rubs against the reel flange; tape creasing or wrinkling which results doesn't help reading. The seating of a 727 tape is strictly manual--uniform pressure of the operator's hand as he tightens the pressure band knob with the other hand. Uniservo's locking mechanism is not only faster than the 727's, but it assures firm seating of the reel.
3. The locking knob for the pressure band may not be tightened completely. This is a good way to have the reel fly off during high-speed rewind, which has happened. Conversely, some strong-fisted operator may have given the knob a final extra twist; a weaker-fingered tape changer may fight this for quite a while before he can loosen it. This also happens.
4. The tape may slip out of the holding spring on the hub of the takeup reel while the first few turns are being wound manually; this obviously delays tape changing. Also, getting the end of the tape inserted under the spring may be quite a feat for an "all thumbs" operator; it is strictly a "close quarters" job.

1109.05 Multiple Tape Writing. Several 727 tape units may be set to the same address, permitting simultaneous writing of the same information on more than one tape. The maximum number differs with the various equipments with which used. It is a fast way of tape duplication which requires a separate run on Univac.

1109.05 (Continued)

For the 727. This is a desirable feature when more than one copy of an output tape is necessary and the requirement is known at the time of initial preparation. For example, one reel can be stored in a remote location and serve as a spare if the "working copy" is accidentally destroyed; another use is that of a central office sending duplicate copies of master file information to a number of branches for local use. When so used, tape duplication in this manner can be useful.

Against the 727. There is no good rebuttal if a duplicate must of necessity be prepared. However, to date there is no known use of duplicate tapes as insurance media, and it is well to ask: "How much Univac time would be required to do essential duplication?" If the duplicate is prepared on an IBM equipment simply to avoid a machine return when one output tape can't be read subsequently, the duplication method becomes a fairly expensive form of operating insurance not necessary with Univac. (See 1109.07).

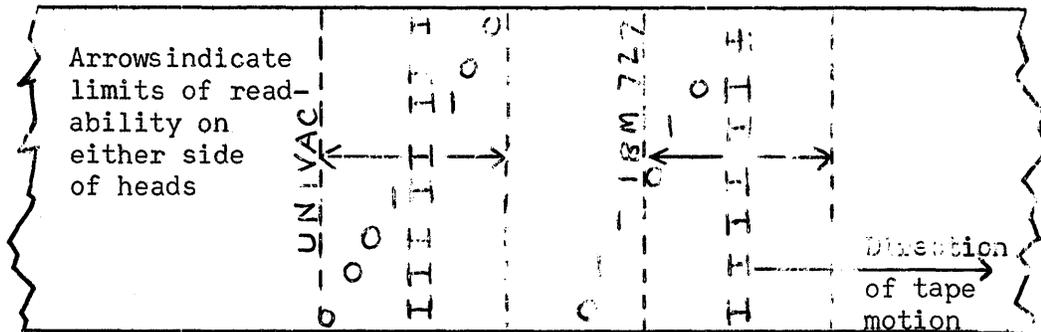
1109.06 Tape Reliability. Occasionally remarks are encountered on the relative merits of plastic versus metallic tape, usually with reference to the fact that reels of the heavier metallic tape are shorter in length and hence contain less information per reel. Capacity is discussed in 1109.02. Technically, today's plastic and metallic tapes are highly reliable, although each has certain technical advantages not possessed by the other. Univac I uses metallic-based tape largely because it was far superior to plastic when a tape choice had to be made several years ago. Our position is best illustrated by noting that Uniservo II will be able to handle either type; as a matter of fact, at least one user has modified Uniservo I to handle plastic tape.

Against the 727. It can use only plastic-based tape, which means that the more permanent storage qualities of metallic tape are not possible with any IBM system. Furthermore, the tape used is acetate-based, not mylar; the latter is being tested on 727s, but so far as known is not actually in production use. (See 1109.07 for some comments on the use of mylar with the 727). Uniservos handle either plastic or metallic; the latter withstands temperatures which reduce plastic to a glop. This can be an important point where irreplaceable records are to be stored for long periods of time.

1109.07 Readability. Univac tapes can be read with a much higher degree of accuracy than those of the 727. There are two good technical reasons for this; because our superior tape readability is a major selling advantage, a simple non-technical explanation of each follows.

Skew. By skew is meant the slight twisting which a tape may undergo as it passes under the read-write heads. What it means is that the pulses don't all reach the heads, when reading, at the same time; the pulses from the lagging side of the tape reach the heads after the leading side pulses, and the worse the skew the longer the lag. Now neither the Uniservo nor the 727 is always alert for pulses; for technical reasons, the heads are "opened" momentarily (a couple microseconds) and the pulses "sampled." A highly-exaggerated illustration will show

why the Uniservo is a better reader:



Here, a "1" represents a magnetized spot on tape; a "0", an unmagnetized spot. As the tape moves under the heads, the "1s" cause a magnetic field to increase in intensity to a maximum value and then die out; reading can occur only when the strength of this magnetic field is above some minimum, or threshold, value. Hence, while the entire rise and fall of the magnetic field may spread over 20 or more microseconds, it may only be strong (big) enough to be usable for a very few microseconds. Obviously, if a tape is skewed, the peak values of the pulses don't occur under all heads at the same time, and if the skew is bad enough, one or two of the magnetized spots may be so far away from the heads that their magnetic field, at the time of sampling, is below the threshold value. And the basic difference between the Uniservo and the 727 is when the pulses are sampled.

We use the sprocket pulse--which is a magnetized spot for every character code on tape--to trigger the sampling. Because it's in the middle of the tape, the leading pulses (when skew is present) are beginning to drop off and the trailing ones are still increasing. But there is still enough signal (unless the skew is quite bad) to read all of them. If, for example, the heads can recognize pulses from magnetized spots which, timewise, are 5 microseconds away from the heads the Uniservo can read a tape which is skewed so badly that the leading side passes under the heads 10 microseconds before the trailing side. (Needless to say, these times are exaggerated).

Now for the 727, it has no sprocket pulse, but uses the first pulse of each character code which comes under the heads to trigger off the sampling of the entire character. It may be that the first pulse is in the middle of a character and no trouble occurs. But because the check bit is one outside channel, the lowest numeric bit is the other, and the odds are that half of the bits in either of these are "1s," it is rather evident that the presence of skew will cause the sampling to be triggered off by an outside bit--the beginning of a character--about half the time. And, using the exaggerated example of the Uniservo, it is equally evident that a pulse trailing by more than 5 microseconds isn't going to be read. In other words, if all other factors were equal, the Uniservo could read a tape skewed twice as badly as could the 727. Of course, the other factors aren't all equal,

1109.07 (Continued)

but it suffices to say that a Uniservo not only reads better, but a lot better, than a 727.

Time Between Pulses. All tape units write pulses at periodic intervals, timed by an electronic clock. For example, Univac II writes one character every 50 microseconds, Univac I about every 78 and the 727 about every 66.7. In reading tapes, Uniservos pay no attention to the time interval between successive pulses--they simply look for a sprocket pulse and know that a character is under the heads when one is sensed. How far apart sprocket pulses may be is immaterial; they can be 200 to the inch, two feet apart (and frequently are if a bad spot punch out is reached), or one per tape. The Uniservo simply reads until it has counted 720 characters, which can take 51 milliseconds or 51 hours.

The 727, however, must time its reading as well as its writing operation. This is necessary first, because there is no sprocket pulse and, second, because the number of characters in a tape record is variable and lack of a character code is its only signal that a record is finished. It expects a character about every 66.7 microseconds, and is only looking ("Sampling") for one that often, with a small deviation (timewise) on either side. Now suppose that a writing tape unit is moving a tape somewhat faster than the normal rated speed--which is a perfectly normal expectancy with a mechanically-actuated operation. (It happens on Uniservos, for instance). The characters then are a bit farther apart than usual, because they are written at fixed time intervals. Now put this tape on another unit moving somewhat slower than normal--also a normal expectancy. Instead of characters coming under the heads every 67 microseconds, they'll be a bit farther apart, and take longer. In extreme cases, it can take enough longer to lie outside the maximum permissible deviation, with the result that the tape unit assumes it's reached the end of the record and disconnects the heads. Usually, no permanent error results, because the check character test fails at best, rereading is necessary and, at worst, the tape must be changed to another tape unit.

For the 727. Not much. One less head, with its associated circuitry, should result in lower cost for the tape unit, and might boost the 727 price if it were included.

Against the 727. The higher reliability and readability of Univac tapes is a major selling point in our systems. Here are some points to drive home.

1. Explain why we read tapes more accurately-- the above is a non-technical reason.
2. It is not uncommon practice to write the tape unit number on 727 output labels and to try to place them on the same physical unit when reading is required. This is often impracticable and usually a nuisance--for example, an output tape which may be destined for an off-line printing operation. Univac tapes--regardless of the equipment used--are completely interchangeable within and between installations; 727 tapes cause difficulty in both cases.

1109.07 (Continued)

3. At least one 702 installation rereads every output tape as soon as it has been created. This means a rewind and a read-- $7\frac{1}{2}$  or 8 minutes with a full reel--during which nothing else is done. They do it because experience has taught them that too many reels can't be read and their conclusion was that rereading all tapes took less time than rerunning portions of jobs only when outputs couldn't be read subsequently. The latter is the conventional Univac practice; Uniservos are reliable enough to make it cheaper to do a rerun for the occasional tape which does turn out to be unreadable.
4. The introduction of mylar tape into the IBM line is already long delayed. The major difficulty is the fact that mylar stretches rather readily and the 727's method of reading obviously won't take much stretch, which increases the time between pulses. The Uniservo, of course, doesn't have this trouble. Ultimately, IBM will probably get it to work, but right now it seems to be all talk and no performance.
5. STRESS UNISERVO RELIABILITY. THE 727 IS A WEAK LINK IN THE ENTIRE IBM COMPUTER LINE AND MANY OF THEIR CLAIMED ADVANTAGES FALL FLAT BECAUSE OF UNRELIABLE TAPE PERFORMANCE.

1109.08 Fixed Versus Variable Block Length. Insofar as tape unit characteristics are concerned, the major implication of the variable record length of all IBM equipments is that tapes can't be read backwards; without knowing the number of characters in a record it can't be reassembled properly in memory.

For the 727. By writing long records on tape, not only is more information stored on one reel, but read-write time is reduced because there are fewer inter-records gaps and start-stops times.

Against the 727. Variable block length is subject to restrictions and systems considerations; the latter, of course, are not a function of tape unit characteristics.

1. Buffer and control unit considerations restrict the size of tape records. In practices, 702/705 records are limited to 1,000 characters; 650 records to 600 decimal digits plus 60 signs; 704 records are the only ones not subject to these restrictions. See Sections 1120 and following for the effects of various types of tape control units on record length.
2. Inability to read backwards causes sorting and merging operations with 727s to take more time than if this were possible. It is necessary to rewind tapes between each sort-merge, whereas Uniservos read the tapes in both directions, eliminating the "idle" time during rewinds.
3. Inability to read backwards is another reason for a high-speed rewind in the 727.

1109.09 End-of-Tape Indication. The 727 provides for testing for end of tapes by the reflective spot (writing) or the tape mark (reading). In Uniservos, which have no comparable facility, output blocks must be counted

1109.09 (Continued)

and tested against a maximum (normal tape capacity) to determine when a tape is full; input blocks are tested for sentinels. The claim is made--correctly--that the 727 method permits the test to be made with fewer instructions and less computing time.

For the 727. IBM points out the ease of testing end-of-tape, both on reading and writing. It takes only one instruction on the 705, and no more than two on the 702, for example.

Against the 727. The theory of the 727 method is fine; the practical aspects are something else. The basic objection pertains to the fact that the 727 is completely unchecked in its control functions. This is more thoroughly discussed in 1109.9, but two effect immediately pertinent are listed here.

1. For reasons pertaining to lack of certainty on execution of some tape instructions, some IBM users are numbering all tape records during writing and checking them during reading. This is similar to the Univac block-counting method, but the record numbers are actually placed on tape (adding to read-write time), a practice not often encountered with Univac. This practice, of course, makes block or record control fully as time-consuming as in Univac, offsetting much of the theoretical advantage of reflective spots and tape marks.
2. Both reflective spot and tape mark give unchecked indications. In writing, the spot can fail to give a signal, which results in "writing off the end of the tape." This evidently requires a rerun of the portion of the job creating that tape. Missing a tape mark in reading is normally not so serious, because anything following it on the tape usually is a partly erased record which will give a character-check error. Of course, time is lost in discovering that this type error occurred; the indication of missing the tape mark is exactly the same as failure to read a record on tape correctly.

1109.10 Character Code Checking. Uniservos are built to give each character code a redundancy bit check and a 720-character block count; an error condition results if either fails anyplace during a block. Records on the 727 also undergo a redundancy bit test, but substitute the check character test for the character count; with a variable-length record, something other than a character count is obviously necessary. Insofar as the adequacy of the test are concerned, both methods are perfectly sound and there is nothing to indicate one is any better, or any worse, than the other. However, in certain cases it is possible, with the 727, to isolate an error to a single character through use of the check bit, and to determine which channel (-i.e., bit) is erroneous through a channel error indicator actuated by the check character; such an indicator is incorporated in some of the control units used with 727s.

For the 727: With the channel error indicator and the check bit for the character, it is possible to isolate the specific bit in a specific character which is incorrect (this applies to tape reading). This permits correcting the error at the operator's console, rather than rerunning the job which originally created the tape.

## 1109.10 (Continued)

Against the 727. This is a fine idea, provided the error resulted from a single bit in a single character of a record which actually is too weak a magnetic spot to be read successfully (this implies that it is a binary "1"). This is not usually the case; a bit missed on one reading has an odds-on chance of being read successfully the next time, and it's faster to program a tape reread than to attempt manual correction. It is more attractive as a possibility when a minute tape flaw prevents reading a single bit--a flaw something like 1/200 by 1/15 inch in size at most. That's a pretty small flaw. In fact, in over 15 months of operation of a 702, one user has never been able to correct a record reading error in this fashion. It is a theoretically attractive sales point of relatively minor importance, and of practically no use (if any) in actual operation.

1109.11 Method of assigning Tape Unit Addresses. The logical (program) address of a 727 unit is set by a rotary selector switch on each unit. The address of Uniservos is determined by wiring in a small plugboard on the main frame. Relative advantages and disadvantages of the two methods are probably not of any great importance.

For the 727: If a tape unit malfunctions during a run and the tape can be changed to another unit, the address change can be made by the tape changer in the 727 method. With Uniservos, a plugboard wire--not physically located next to the tape unit--must be changed.

Against the 727: The plugboard method used with Uniservos not only assigns positively the units required for a given run, but also automatically disconnects those not needed. With the 727, units not required for a machine run must be set to an address not used in the program. In addition, there is some chance of error because the tape handler may not change the address of a unit to its new logical address, or may set it incorrectly. However, proper housekeeping programs should test the status of all units needed in a machine run and any incorrect settings can be corrected before the main body of the program begins. At most, inconvenience and some processing delay results.

1109.12 Tape Storage Requirements. Plastic-based magnetic tape--particularly acetate--is best stored in a temperature and humidity-controlled area; this is not mandatory for metallic tapes. In general, it seems most EDPM users are providing good storage facilities for tapes, usually air-conditioned. Metallic tape does seem to withstand shipment--particularly by air--somewhat better than plastic.

For the 727. Adequate storage is normally provided anyway.

Against the 727. As used in IBM equipments, the tape must be at normal computer room temperature before use. If stored in a place below recommended temperature limits, it must be removed and allowed to "warm up" to normal temperature before use. This is not necessary with Univac metallic tapes.

1109.13 Accuracy of Type 727 Tape Writing/Reading. In recent sales proposals, IBM has claimed that experience with Type 727 Tape Units in the first 15-20 Type 705 installations has indicated that slightly less than one unreadable tape error occurs per 8-hour shift. The Type 729, announced for use with the Type 709 EDPM, is a modified 727 which includes an immediate reading of newly-written tapes as its major change.

For the 727: This is a commendable performance record, if true. It is much better than has been experienced by some known 702/705 installations.

Against the 727: If the Type 727 Tape Units and magnetic tape are this good, why has IBM found it expedient to revamp the tape writing mechanism to give an immediate re-read for the avowed purpose of proving legibility? If the 727 is so accurate, is it worth another \$100 a month per tape unit to install additional checking? For sales purposes, note that the 729 cannot be used in central processor operations with the 650, 702, 704 or 705, which require a Type 727. The fact that the 729 is incorporating this additional checking feature can be used, however, to combat claims of the accuracy of the Type 727.

\* 1109.14 Is "Perfect" Tape Possible? The major reason for including the read-check feature in the 729-709 is probably not so much to prove that tape recording is good, but to provide a means of detecting flaws on the tape. (1108). In other words, the principal use of the feature admits that, even if tape is perfect when it leaves the factory, it acquires flaws in use. Unless these flaws can be passed over in some manner, two shorter pieces of tape result. With the 729/709, a means is provided for passing over these bad spots; this is not possible in the other equipments, but ALL OF THEM USE EXACTLY THE SAME TAPE.

\* Against the 727: If IBM's present tape performance is so good (see 1109.13 for claims made), why has the company found it necessary to provide a means of bad spot detection in a system that won't be in use until late 1958? The fact that they HAVE incorporated the feature is a definite indication that today's tape ISN'T AS GOOD as they claim and it does develop flaws. The inclusion of the read-check feature in the 729/709 and provision for program-controlled passing over of bad spots is the best proof available that Type 727 tape performance, and the availability of perfect tape, IS NOT SATISFACTORY.

1109.50 SALES POINTS ON TYPE 729 MAGNETIC TAPE UNIT

Pending operating experience with this tape unit, results of uses of the Type 727 must form the basis for sales evaluations. These sales points on the 727 are pertinent:

1109.01	1109.04	1109.08	1109.11
1109.02	1109.05	1109.09	1109.12
1109.03	1109.06	1109.10	1109.9 (last page of this section)

1109.51 Necessity of Readability Check. The Type 729 functionally differs from the Type 727 only in having the "split head" that reads a newly-written record immediately after recording it on tape; this costs an additional \$100 monthly per tape unit. Recent IBM sales proposals for Type 705 installations, using the 727 tape unit, stress the point that experience with the first 15-20 such installations are having somewhat less than one unreadable tape error per 8-hour shift.

For the 729: This is a nice feature, provided it performs a useful function.

Against the 729: If the performance of Type 727 tape units is as good as claimed in Type 705 proposals, is it worth an extra \$100 a month for each unit to prove tape legibility immediately after writing? Second, how much of IBM's tape troubles are caused by inability to read a tape on other than the unit on which originally prepared--and this is a known occurrence which happens quite frequently. Reading difficulties caused by tape unit "incompatibilities" will not be corrected simply by proving that a tape is good; the fact that a number of "illegible" records can be read properly merely by switching the reel to another tape unit is proof of that.

1109.9 Unchecked Control Errors on the 727. In all EDPMs the control of tape units--reading, writing, rewinding, backspacing, etc.-- is through use of normal machine instructions. All IBM equipments using the 727 are, essentially, completely unchecked insofar as verifying the accuracy of instruction execution is concerned. Univac verifies that the correct operation is actually performed--another feature of Univac's justly-famous "self-checking" circuitry.

For the 727. Because checking is basically a function of main frame circuitry, there are no evident 727 advantages in omitting it.

Against the 727. Each of the types of "busts" listed are known to have occurred in 702 installations; because the other types of equipments use the same basic unchecked method of operating tape units, comparable performance can be expected.

1. Failure to recognize the reflective spot at the end of the tape while writing. This is an unchecked photoelectric sensing which gives only one signal when working properly. It doesn't always give the signal. See also item 2 in 1109.09.
2. Failure to recognize the tape mark when reading. This likewise is a single unchecked signal. However, it is rather easily corrected--at the expense of some programming--by writing two or three tape marks after the last information record. The equipment is bound to pick up one of them.
3. Failure to execute a "backspace" instruction. Although the 727 can never "read backwards," it can be backspaced one record at a time. The instruction is not always executed, with the result that it is perfectly possible to completely drop a record from a master file during an updating operation. It is understood that many customers are allergic to this. It is certain to happen--particularly over a period of time--unless elaborate and time-consuming programming is incorporated to make sure that a backspace instruction actually backspaces. One method is block or record numbering--which takes both program steps and computer time.
4. Writing one output record two or more times. This has happened and is possible in two ways. First is failure of the instruction address counter to advance to the next instruction address--unlike Univac, this is unchecked in at least the 702 or 705. The result is that an instruction is executed twice. If it's a "write" instruction, an output block is duplicated. The seriousness of the error depends on the file involved; it may be only a source of annoyance in a future machine run; it could also result in posting \$5,000 to my checking account twice. The second cause of duplicate output records results when a "write" error is indicated and the tape fails to backspace before it is rewritten. This usually is more of an annoyance than anything else; of course, the odds are good that the tape is to print a management report which looks rather loused up in finished form. This is particularly embarrassing when the output block is repeated four times--which has happened.
5. Failure to select the proper tape unit. Because the equipments do not verify that the tape unit specified by the instruction is the one actually connected, it is possible to write on an erroneous tape (effectively losing information which belongs on another output), or reading in from an incorrect tape. Use of

1109.9 (Continued)

the "protective rings" will eliminate the possibility of accidentally writing on a protected reel, but the other chances of error remain. This type of error often merely "stalls" the machine, because the erroneous address may apply to a non-existent tape unit or tries to "write" on a protected unit which can only be in a "read" status.

1120 IBM TYPE 652 TAPE AND RAMAC CONTROL UNIT

The Type 652 is a control unit used for three separate purposes, alone or in combination, in the IBM 650 EDPM system.

- a. To connect the Type 727 Tape Unit;
- b. To connect the Type 355 Random Access Memory (RAMAC);
- c. To connect the Type 838 Inquiry Keyboard.

Use of either magnetic tapes or RAMACs in the 650 system requires also the inclusion of a Type 653 Record Storage Unit (see 1125) and, to be technically correct, the 653 connects the tape units or RAMACs to the high-speed storage element contained in the 653. Type 838 Inquiry Keyboards can be used only if the 650 installation also includes RAMACs.

1120.1 Monthly Rental. Rental varies depending upon the type of configuration involved.

652 for magnetic tape units only. . . . .	\$ 1,050
652 for RAMACs only . . . . .	975
652 For magnetic tape units and RAMACs. . . . .	1,350
Each Inquiry Control Unit (maximum two), additional . . . . .	300

Monthly rental for a 652 adapted only for RAMACs will be \$975, \$1,275, or \$1,575, for none, one or two Inquiry Control Units; if also equipped to handle magnetic tapes, the corresponding rentals are \$1,350, \$1,650, or \$1,950.

1121 CHARACTERISTICS OF THE TYPE 652

The basic purpose of the 652 is to provide the power and control signals necessary to handle magnetic tape units, RAMACs and Inquiry Keyboards. It does not need to be considered in the writing of programs using magnetic tape units or RAMACs, nor in the operation of Inquiry Keyboards; however, it is involved in the initial setup of the latter. It can control a maximum of six Type 727 Tape Units, four Type 355 RAMACs and ten Type 838 Inquiry Keyboards through one or the Inquiry Control Units. Any combination, up to the maximum of each type of unit indicated, is permissible, provided that the 652 is equipped to handle them (see 1120.1).

1121.1 Characteristics of the Type 652 -- Magnetic Tapes

Number of Type 652 which can be connected to one Type 650 (through the 653)	1
Number of Type 727s connected to one 652 Tape Unit Addresses	1 to 6 (maximum) 8010 through 8015.
Method of Address Selection	Rotary switch on 727s for units position; "801" is built in
Simultaneous read/write operations	No; read only or write only
Restrictions on number of tapes which can read or write	None, within limit of 6
Simultaneous writing	Two tapes, by setting both to same address

1121.2 Characteristics of the Type 652 -- RAMACs. Refer to 2120.

1121.3 Characteristics of the Type 652 -- Inquiry Keyboards. Refer to 7110.  
This also discusses the Inquiry Control Units.

For a fuller discussion, refer to sections on magnetic tapes, RAMACs, Inquiry Keyboards, the 650 and the 650 RAMAC.

1125 IBM TYPE 653 RECORD STORAGE UNIT

The IBM Type 653 Record (or High-Speed) Storage Unit must be used if magnetic tapes or RAMACs are attached to a Type 650 EDPM, but may also be used, without either of these, as an extension of the storage capacity of the Type 650 itself. If used with tapes or RAMACs, a Type 652 Control Unit (see 1120) is also required. Only one Type 653 can be associated with one Type 650.

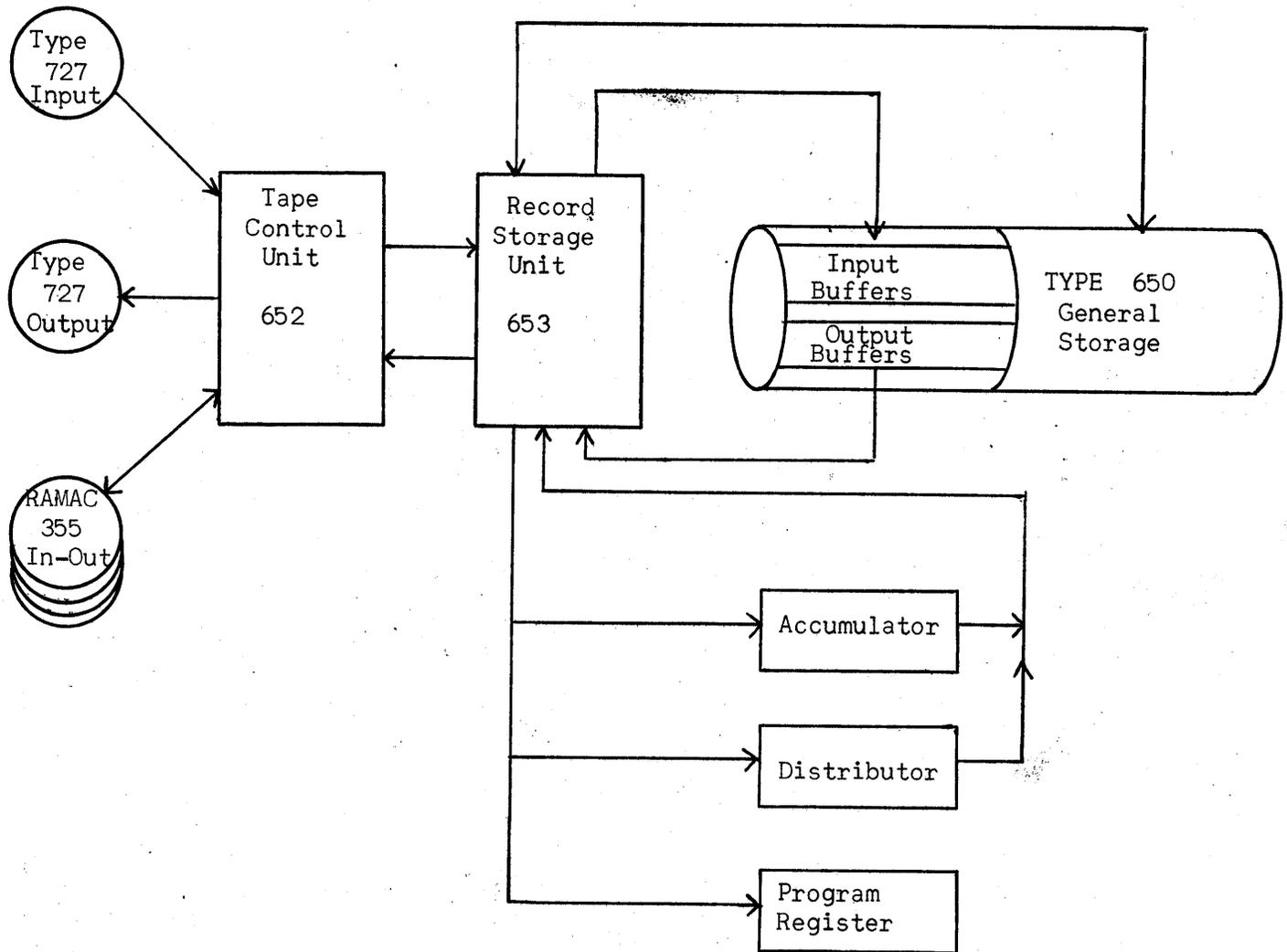
1125.1 Monthly rental:1126 CHARACTERISTICS OF THE RECORD (HIGH SPEED) STORAGE UNIT

The characteristics listed below are common to all uses of the Type 653. More detailed discussion of its use with Type 727 tape units is contained in 1127; with Type 355 RAMACs in 2120 and 6150; with the 650 processor alone in 6100.

Type of storage:	Magnetic core
Character Capacity:	60 words (600 decimal digit plus 60 signs)
Addressing Facilities:	Each word individually addressable
Address Range:	9000 through 9059
Input to High-Speed Storage:	(1) Directly from magnetic tape or RAMAC (2) From any other input device through the input buffer (3) One word transfers from distributor and accumulator (4) Block transfers from general storage on the 650 drum
Output from High-Speed Storage:	(1) Directly to magnetic tape or RAMAC (2) To any other output device through the output buffer (3) One word transfers to distributor, accumulator or program register (4) Block transfers to general storage
Multiplexing Ability:	None. The 653 can carry on only one operation at a time. I.e., if reading from one tape, it cannot write on another, cannot be transferring data to or from a RAMAC, nor do any other type of operation.

1126.1 Block Diagram of Type 652-653 Association in Type 650 EDPM

This diagram shows the relationships of the 653, the 652 Control Unit, tape units and RAMACs, with information flow paths, in the 650 system.



1127 USE OF TYPE 653 IN MAGNETIC TAPE OPERATIONS WITH TYPE 650 EDPM

When used with Type 727 tape units in 650 operations, the Type 653 High-Speed Storage Unit serves both as tape buffer and as an extension of the general storage (the drum) of the machine. It cannot serve both functions simultaneously; when tape is being written or read, the 653 cannot be used in carrying on other instructions, which must be confined to the general storage on the drum. The length of time that High-Speed Storage is tied up depends upon the particular type of operation involved in tape reading or writing; typical times are given in 1127.2.

- 1127.1 Length of Tape Records. Refer to 1103.3 for description of tape formats used. The point from which a tape record is written onto tape, or into which it is read, is determined by a timing ring which serves essentially, as a starting point in the 60-word magnetic code high-speed storage. This timing ring can be set by several means (not discussed here) to any address between 9000 and 9059; in both input and output operations, the data transfers begins with the address of the timing ring and continues through address 9059. Thus numeric information can be read or written in full-word increments of any length from 1 to 60; alphanumeric information is subject to the restriction that it must be in increments which are multiples of ten words.
- 1127.2 Operating Times for Tape Input-Output Functions. The use of magnetic tapes with the 650 imposes the necessity of certain control interlock functions during the execution of the various types of tape instructions. The following table indicates the times necessary for carrying out the types of operations indicated, and also shows the period of time during which a tape unit, the 652 tape control unit, and the 653 storage unit are not available for other work. The times given do not include that necessary for the instruction initiating the action; this averages about 0.3 ms. All times are in milliseconds.

<u>Read/Write Tape:</u>	$10 \neq 0.68N$ , where N is number of characters in the tape record. A 60-word numeric record requires 50.8 ms to read or write. All tape units are unavailable during this period. The 652 is unavailable during this period. The 653 is unavailable during this period.
<u>Write Tape Mark:</u>	11.0 ms. The 652 and all tape units are unavailable during this period. The 653 is available throughout the period.
<u>Backspace Tape:</u>	If tape in write status: $46 \neq 0.068N$ ms; " " in read status: $30 \neq 0.068N$ ms; All tape units except the one addressed and the 652 and 653 are available during this period.
<u>Rewind Tape:</u>	Maximum 1.2 minutes. All tape units except the one addressed and the 652 and 653 are available during this period.

In these remarks, the term "unavailable" means that an instruction addressed to any unit during the time period indicated will cause the 650 to "hang up" on that instruction until the current operation is completed. The maximum tape transfer rate of the 650 is about 11,800 characters per second, combined read and write, or about 5,900 each. Effective transfer rate is much less, because computation may not overlap and, at any event, transfers between general and high-speed storage may be required.

1140 The Type 754 is an unbuffered Tape Control Unit providing power and control functions for up to ten Type 727 Tape Units in 702/705 EDPM installations. In either type of system, up to ten 754s, or combinations of 754s, 760s, and 777s, may be connected; each of the 754s in use may have from one to ten tape units. The maximum tape unit complement of the 702/705 series is 100, achieved with ten 754s.

1140.1 Monthly rental: \$2,000.

1141 BASIC CHARACTERISTICS AND USE OF THE Type 754

See 1103.4 and 1103.6 for explanation of tape formats with the 702 and 705. The 754 provides unbuffered communication between the 702/705 memory and tape units. Because it does not include a buffer, use of the 754 means that input and output do not overlap computation. In addition, the 702 is not able to read and write tape simultaneously; consequently, reading, writing and computation proceed in sequence, with none of the three overlapping. The 705 is able to read on one tape and write on another simultaneously, but nothing else can occur until the longer of the two operations has been completed. With the 754, computation never overlaps input-output operations.

The principal advantage of an unbuffered tape unit is that it permits extremely long tape records; for example, the first sorting routine developed by IBM generated strings of somewhat more than 8,000 characters which were written as one tape record. This not only increases the total character capacity of a tape unit, but also reduces input/output time by cutting down the number of inter-record gaps which must be traversed in a tape operation.

One characteristic of the 754 is worth mentioning: In use, it is not necessary for the tape units to be "stopped" in order for a read/write instruction to be executed; regardless of where it is in the stop cycle, the next instruction commences immediately. Thus, as soon as the last character of a tape record has been read into memory, tests can be initiated to determine whether it is required for processing; the 702/705 are not held up until the tape has stopped. Such a test may take less than a millisecond, at the end of which time a combined read/write instruction (in the 705) will cause both tapes to immediately resume full speed, even though not stopped. The stop curve is such that it always takes 10 ms to cover any remaining portion of the inter-record gap. Therefore, instead of starting and stopping the tape in the gap, which would require a minimum of about 16.4 ms, it is often possible to cover it in about 11 ms, one used for computing (during which time the tape is starting to stop) and 10 for covering the remaining portion of the gap.

The time required to read, write, or read/write is  $10 / 0.067N$  ms, where N is the number of characters involved (in read/write, the number of characters in the longer of the two records). Nothing is done in this period; the time is 77 ms for a 1,000-character record.

1145 IBM TYPE 755 TAPE CONTROL UNIT

The Type 755 is a Tape Control Unit connecting from one to eight Type 729 Tape Units to one channel of a Type 766 Data Synchronizer in 709 EDPM operations. Its basic function is to provide power and a few control and switching circuitry functions to the tape units in on-line processing. It does not need to be taken into account in programming and has no operating features requiring special comment.

A separate 755 is required for each Data Synchronizer channel having tapes attached, regardless of the number associated with each channel.

1145.1 Cost: Purchase: \$108,000  
Rental: 1,800

These prices apply to the Model 1, which is the only one so far announced.

1145.2 Discussion of Use. The unit is of no programming or operating importance. Refer to section 1160, Type 766 Data Synchronizer, for a discussion and evaluation of the use of magnetic tapes in a 709 system.

No sales points are included for this equipment.

1149 SALES POINTS ON THE TYPE 754 TAPE CONTROL UNIT

IBM does not appear to be including 754s in many of its current proposals; this is due to the fact that it is unbuffered, and also to the fact that the cost of the 777 Tape Record Coordinator is only \$1,000 a month more. Occasionally, a 754 will be included in sales proposals where total rental is a critical competitive factor; it permits proposing a system with ten tape units (rather than the eight maximum with one 777) at a cost less than a comparable Univac II. In some instances, has proposed as few as six or seven tape units with one 754 to get the price down to a figure close to a Univac I. The following points should be brought out when a 754 control unit is proposed with a 705; in general, these points are true even though one or more 777s are also proposed.

- 1149.01 Inability to Read/Write and Compute Simultaneously. With the 754, the 705 can write from one area of memory and read into another area simultaneously, but can do nothing else until both operations are completed. All computing must be done outside of tape time. This includes, as a normal minimum, all tape read, write and error test instructions, testing of data to determine if the record is to be processed, possible block reassembly into a new output, and either internal memory transfers to move the input to an output area, or modification of input/output instructions to permit writing out of the area into which reading occurred. The amount of this computing time varies greatly with applications.

Against the 754. Simultaneous read/write operations are no novelty-- Univac has had it since the first one was built, and of course also computes simultaneously. The necessity of computing outside tape time adds--sometimes greatly-- to total processing time.

- 1149.02 Comparative Processing Speeds of a 705-754 and Univac II. The effective speed of a 705-754 is dependent not only upon the amount of computing involved, but also upon the length of the grouped tape records. Greater tape speed is attained by grouping several records into one long tape record, cutting down the number of gaps which must be passed over; of course, the longer the record, the more memory must be aside to hold it and the greater the amount of computing per tape record. A normal maximum is about 1,000 characters, partly because some customers expect to get 777s and set up records initially on that basis. The following table shows typical speeds possible, in numbers of characters per second, for Univac II and the 705-754. Various length tape records are included for the 705 (UNIVAC always uses a 720-character block), and times are shown for instructions varying from 50 to 500 per tape record. The amount of computing time assumes an average of 0.14 ms per instruction for Univac II and 0.07 for the 705; this is normally not valid but the 705 does execute individual instructions at an average rate about twice as fast as Univac II. This does not mean, of course, that it will perform a required processing operation twice as fast, internally, as Univac II.

1149.02 (Continued)

Average Processing Speeds of Univac II and the IBM 705  
Using the Type 754 Tape Control Unit

(Characters per Second)

<u>Number of Instr'ns</u>	<u>Univac II</u>	<u>Length of Tape Records, 705-754</u>				
		<u>200</u>	<u>400</u>	<u>600</u>	<u>800</u>	<u>1000</u>
50	14,300	7,450	9,900	11,100	12,100	12,400
100	14,300	6,600	9,100	10,500	11,500	11,900
200	14,300	5,350	7,850	9,300	10,400	11,000
300	14,300	4,500	6,900	8,400	9,600	10,200
400	11,600	3,900	6,150	7,700	8,800	9,500
500	9,500	3,400	5,500	7,050	8,200	8,900

Estimating the number of instructions required to process one block is frequently difficult, particularly in sales proposals. As a reference point, it may be noted that the typical Univac II block requires about 150 instructions for necessary housekeeping, block reassembly and tests to determine if any item in the block must be processed. The 705 will take fully as many. Actual processing is in addition to this, and varies considerably; it is always additional time in the 705-754, while Univac II can execute a total of about 325 (or something like 175 aside from housekeeping) within tape time. In connection with the 705 speeds noted above, see also the next section.

About the 754. These two points should be brought out:

- (1) Computing is always in addition to tape time, and tape read/write speeds are slower than Univac II.
- (2) Univac II, on the average will out-perform a 705-754 combination by at least 35-40%, and it may run more.

1149.03 Effect of Variations in Length of Tape Records. The above speeds are based upon 705 tape records of identical lengths, a condition automatic in Univac but not in the 705. Because the 705 is tied up in the tape-memory transfers until the longer of the two unequal-length records has been handled, it is obviously desirable to have them all the same length. Many business files, of course, have records with variable amounts of detail information appended to them, and IBM has laid great stress on the ability of the 705 to handle variable-length words and records. If records are variable in length, maintenance of a fixed tape record must be programmed; usually reassembly of an input tape record into parts of the outputs must be provided for, because most file maintenance operations have to provide for insertion and deletion of information. IBM will doubtless point out that its high-speed internal memory transfer makes this easy and fast to accomplish. They will not point out the details necessary to use the high-speed transfer; these seriously slow down the operation. Refer to 5140 for a fuller discussion of the ramifications of the high-speed internal memory transfer.

If tape records are allowed to vary in length, the total time required to read/write a file is greater than a time calculated on the basis of the average length record; how much greater depends upon the variations

## 1149.03 (Continued)

in length and their distribution. For example, the time needed to process a file consisting of alternating 1000 and 100 character tape records is exactly the same as for a file of all 1000-character records. Typically, the time increase, as compared with average record lengths, will range from 15-25%. The alternative is reassembly internally in fixed-length tape records, which requires computing time. There is evidently a balance point; on one side of it the increased tape time is less than the computing necessary for reassembly, and on the other side the converse is true. Determining where this point is for any given file is extremely difficult, if not impossible.

Against the 754: Point out the above facts; it is suggested that the indeterminateness of the increased processing time be emphasized. In practice, it is almost impossible to estimate the time for a file maintenance run with any degree of reliability, even if it is completely programmed. These restrictions and considerations do not exist in Univac.

1149.04 Number of Tape Units Which Can Be Controlled. With ten 754s, the 705 has a maximum of 100 tape units which can be connected. Univac II is limited to 16. The fact that the 705 has a capacity for more may be a factor in some sales proposals.

For the 754: The 705 system can handle more tape units than the Univac systems.

Against the 754: Most Univac II orders do not include even the 16 maximum available. The number which can be used profitably is in large measure a function of the total memory capacity available; if more inputs and outputs are contemplated (and why have more tape units if this is not true?), it is self-evident that more memory is needed for instructions to handle them and, in many instances, more memory will be needed for data storage. The optimum number of tape units for any given installations must be predicated upon consideration of all applications involved; Available knowledge indicates that, for EDPMs of the scale of the 705 and Univac II, from 10 to 16 are all that can be economically justified.

1150 IBM TYPE 760 CONTROL AND STORAGE UNIT

The Type 760 Control and Storage Unit is primarily designed as a control and buffer storage unit for off-line printing operations using any one of the Types 719, 720 or 730 high-speed printers. Additionally, it can be used in 702/705 central processor operations as a tape/printer control and buffer storage unit. These configurations are possible:

1. Off-Line Operations. One 760 with one HSP (719, 720 or 730) and either one or two Type 727 Magnetic Tape Units.
2. On-Line Operations With 702 or 705.
  - a. One 760 with one or two Types 727 Tape Units.
  - b. One 760 with one HSP (any type).
  - c. One 760 with one HSP and either one or two Type 727 Tape Units.

The use of the 760 is mandatory with the IBM 500-and 1000-line a minute printers (Types 719, 720 and 730), either on-line or off-line. Its use for this purpose is discussed in section 4130. This section is limited largely to its use as a tape control unit in on-line operations. In either method of operation, all normal characteristics of the Type 727 Tape Units (1100) remain unchanged.

1150.1 Monthly Rental: \$1,800.

1151 CHARACTERISTICS OF THE TYPE 760

Number of Type 727 Tape Units controlled	1 or 2 (maximum)
Number of printers controlled	1 only
Type of storage (buffer)	Magnetic core
Storage capacity	1,000 702/705 characters
Character transfer rate:	
Memory and 760 storage	0.023 ms/character
Tape and 760 storage	0.067 ms/character
Checking of information transfers	Yes
EDPMs used with in on-line operations	702 or 705
Number of 760s possible in on-line operations	10; or total of ten in any combination with Type 754 Tape Control Units
Permanency of stored information	Non-destructive read-out; character by character replacement on read-in

1152 ON-LINE USE OF THE TYPE 760 WITH 702 or 705 EDPMS

The Type 760 Control and Storage Unit may be used in 702 or 705 CPU operations as a buffer storage unit controlling one or two Type 727 Tape Units; it can serve this function with or without an HSP also attached. This section describes the method of operation as a tape control unit and also introduces the instructions actuating an HSP in on-line operation.

1152.1 Essential Operating Features of the Type 760. This sub-section outlines the basic features of the 760 which pertain to on-line use. Unless otherwise noted, 702 and 705 uses are identical.

1152.11 Method of Addressing the 760, Tape Units and Printers. The address selecting a tape unit or printer controlled by a specific 760 is of the form "02XY," as the address of a "SELECT" instruction.

"02" designates a tape control unit;

"X" is a digit, from 0 to 9, designating the specific 760 involved. It differs from that assigned to any other 760 or to a Type 754 or (probably) Type 774. Thus a maximum of ten such control units, in any combination, is possible;

"Y" designates the specific tape unit or printer:

2 or 7 - Tape unit

4 - Printer

Others - Invalid; anything other than 2, 4 or 7 in units position of address causes a machine stop.

1152.12 Storage Counter. The 760 contains a counter device that operates from 000 to 999 for the 1,000 character positions of buffer storage. It is used to permit loading the storage unit with several successive records from memory; the method is discussed in the sections outlining instructions used with the 760.

1152.13 Control Mark. The end of information in the storage unit is indicated by a special character code called the control mark. Its function is similar to that of the record mark in the 702 (1103.4) or the group mark in the 705 (1103.6). Available information does not indicate the bit combination forming the control mark, but apparently it is a new one not used in any other component of the 702/705 systems. Its use is covered in the following sections on 760 instructions and summarized following that discussion.

1152.14 Checking of Data Transfers. All data transfers into and out of the 760 storage unit are given redundancy bit tests and tape records read in also receive the check character test. Unlike the Type 777 TRC, the check character is not placed into the buffer storage. Error indications are:

1. Memory to 760: 0902 Read-Write Check Indicator. Usually an error also turns on 0901 Machine Check Indicator.
2. 760 to Memory: 0902 Read-Write Check Indicator.
3. Tape to 760 : 0903 Record Check Indicator. This transfer also verifies the check character on the tape record.
4. 760 to Tape : 0903 Record Check Indicator. The check character is also created, but of course not checked.
5. 760 to HSP : 0903 Record Check Indicator.

## 1152.14 (Continued)

In all cases, either automatic machine stop or program-controlled testing and correction is available. Error indications for the first two types of transfers listed are available immediately upon completion of the transfer; for the last three, the check indicator turns on during execution of the next read or write instruction, which may involve a completely different control unit.

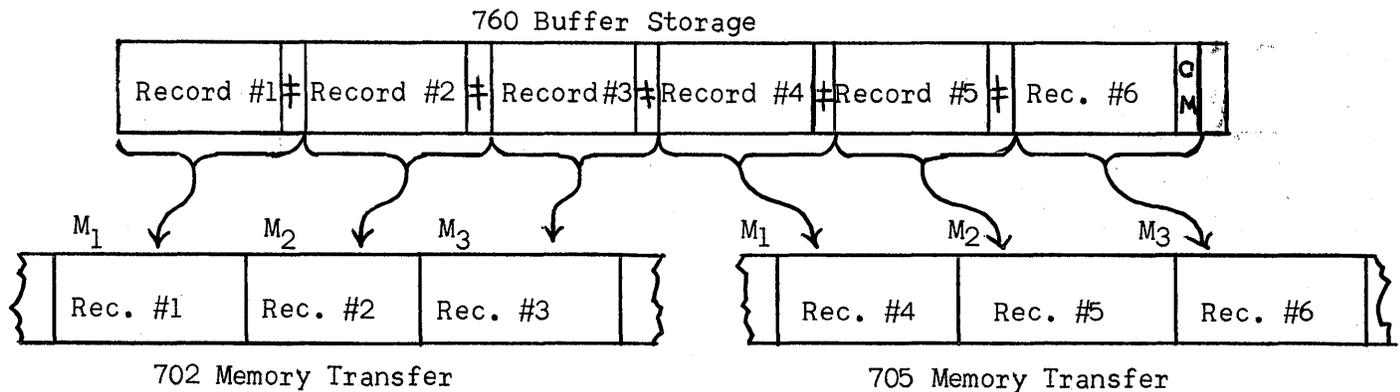
As a matter of interest in the logic of the 702/705 design, note that a tape-to-760 error is indicated by an 0903 Record Check Indicator; a tape-to-777 error, by an 0902 Read-Write Check Indicator. Literature available does not explain the reason for the two different approaches to handling an identical type of machine mistake.

- 1152.15 Tape Unit "Early Start" Feature. Like the Type 777 TRC, it is possible to start a tape in motion during the time a memory-to-760 data transfer is being made, thereby permitting a partial overlap of operations. The feature can be used when the transfer from buffer to memory can be completed in 4 ms or less (early start on "read" tape), or memory to buffer in 8 ms or less (early start on "write" tape). This is time enough to handle 173 and 347 characters, respectively. (IBM uses 170 and 340 as approximations). The reason for the different time intervals is the same as that explained in 1182.21 and 1182.22.
- 1152.5 Instructions Used in Type 760 Operations. Four new instructions are introduced to the 702/705 command list when a 760 is installed. The illustrative examples used in the following sections have addresses 0212 and 0217 for tapes and 0214 for the HSP; thus they refer to a 760 with address "1". Error-testing instructions are omitted, but must of course be included in working programs.
- 1152.21 760-to-Memory Transfers. This type transfer is normally associated with an input tape which is to be read into the 760 buffer after its contents have been placed in memory. Assume 0212 is this input. The 760-to-memory transfer then is accomplished by this set of instructions:

SELECT	0212	Selects 760 control unit "1" and tape unit "2"
READ	m	Transfers one record from 760 buffer storage into 702/705 memory beginning at address m.

The transfer is terminated by a record mark, or the control mark following the last record, in the 760 buffer. Because several records may be contained in the 760, several "read" instructions are necessary to transfer all of them. Each such "read" transfers one record into the memory location designated; rather evidently, successive records in the buffer do not have to be read into successive memory locations. In the 702, the record mark is transferred into memory following the last information character; the control mark following the last record in the buffer is changed automatically into a record mark. In the 705, it is not transferred; the record or control mark terminates the operation, but the last character moved is the one immediately preceding it in the 760 buffer.

## 1152.21 (Continued)



The nature of the successive transfers in both the 702 and 705 is shown in the above illustration. This is illustrative of the transfer--rather obviously one 760 cannot be attached simultaneously to both a 702 and 705. Considering the 702 example, the first "read" instruction moves the first record in the 760 into memory beginning at address  $M_1$ . At its completion, the 760 storage counter (1152.12) is set at the position of the first character of Record #2, and, when the next "read" instruction is executed, the 760 begins transferring at that character position and continues until the next record mark is reached, and so on:

```

SELECT      0212
READ        M1      Read first record into M1
READ        M2      Read second record into M2, etc.

```

Particularly in 705 usage, several successive "read" instructions are normally necessary to transfer the full buffer into memory; this can also be true in the 702, as will become apparent shortly.

1152.22 Memory-to-760 Transfers. This is usually associated with one of the tapes or the printer as ultimate destination of the information; assume it is tape unit 0217. The memory-to-760 transfer requires:

```

SELECT      0217      Select 760 control unit "1" and tape unit "7"
WRITE       m         Transfer into 760 buffer beginning at memory
                       location m

```

In the write-out, the group mark in the 705 terminates the operation, with several records separated by record marks usually involved. In the 702, the record mark terminates the transfer. In a manner similar to reading, it is possible to load several successive records into the 760 by a series of "write" instructions, a procedure which would be applicable particularly to the 702. The storage counter controls the assembly of each record into the successively higher positions of the 760 buffer.

1152.23 CONTROL 0006 (RWS 0006). This new control instruction is used to read in a record from tape, or to write a record on tape or printer. It normally is preceded by a previous "read" or "write" instruction which determines the meaning of "RWS 0006 (Read/Write Storage). Instruction sequences are:

## 1152.23 (Continued)

Reading

SELECT	0212	Select input tape
READ	m	Read (last) record from 760 buffer into memory
RWS	0006	Start tape unit to read next record or grouped record into 760 buffer

Writing

SELECT	0217	Select output tape
WRITE	m	Write record from memory into 760 buffer
RWS	0006	Write contents of 760 buffer onto tape

If the printer (0214) is selected in a "write" operation, one or more lines are printed from the 760 buffer. These points are apropos:

1. RWS 0006 resets the 760 storage counter to 000 at the beginning and completion of the data transfer. Therefore it always reads into or writes from the beginning of the buffer and leaves the counter reset at 000 when the read or write is completed.
2. RWS 0006 causes an automatic conversion of the last record mark in the buffer to a control mark, which terminates a writing operation. In the 705, the group mark terminating a memory-to-760 transfer is changed to a control mark by this instruction (individual record marks within grouped records are unaltered).
3. IBM notes that the 760 can be used in a "tape copying" routine without putting the information through memory:

SELECT	0217	Select output tape
WRITE	X	Any address to put tape in "write" status
SELECT	0212	Select input tape
RWS	0006	Load 760 control unit from input tape
SELECT	0217	Select output tape
RWS	0006	Write 760 buffer contents on output tape

The point of interest here is that the reading from "0212" is the only known instance in which IBM refers to a tape unit's reading or writing without being specifically instructed as to which is involved.

1152.24 CONTROL 0007 (RWT 0007). This instruction is used only with tape units and initiates the early start feature similar to that in the Type 777 TRC. In use, it allows the next "read" instruction to transfer one record from the 760 into memory while the tape is accelerating and before the input record replaces the contents of the 760 buffer. In "writing," one more record can be loaded into the 760 before the actual 760-to-tape transfer starts, but while the tape is accelerating to full speed. In either case, a typical instruction sequence is:

SELECT	0212	Select input (output) tape
RWT	0007	Start selected tape in motion
READ (WRITE)	m	Read (write) one record from (to) 760 buffer, memory address m

## 1152.24 (Continued)

The reading (or writing) begins in the buffer character position designated by the storage address counter, which is not reset to 000 until after completing the read or write instruction following "RWT 0007." (RWT -- Read/Write Tape). Time permits about 170 characters to be read into memory (4 ms maximum at 0.023 ms/character), or 340 to be written from memory (8 ms maximum). The time saved is the difference between standard tape start time of 10 ms and the number of milliseconds required for the memory-760 transfer.

If more than one read or write instruction follows an RWT 0007, the second is "held up" until the tape-760 transfer is completed; because the storage counter is reset to 000 after this transfer, the following read/write begins at this buffer position.

Parenthetically, there is no information as to what happens if the "read" or "write" takes longer than the 4 or 8 ms maximum. Because several possibilities exist, speculation is rather pointless.

1152.25 CONTROL 0008 (RST 0008). This instruction resets (RST -- ReSeT) the storage counter of the selected 760 to 000 without altering the buffer contents. This, coupled with the necessary number of "dummy" reads, permits any desired record in the buffer to be read into memory repeatedly without using a tape unit, or to substitute a revised record into the buffer before writing a new output tape. This may save some time in 760-memory transfers, although the entire contents of the buffer through the record involved must be handled at 0.023 ms per character.

This instruction also completes the methods of resetting the storage counter to 000:

1. By an RST 0008 instruction;
2. At the beginning and completion of an RWS 0006 instruction (automatic);
3. After the read or write instruction following an RWT 0007 (automatic).

It will be noted that it is not possible to read several tape records and accumulate them in the buffer; each begins at 000 and destroys the previous contents. Likewise, several records in the buffer cannot be written out as individual tape records.

1152.26 CONTROL 0009 (PTW 0009). The last of the new instructions is used only with an on-line high-speed printer (719, 720 or 730) when it is desired to both print and record on tape the information in the 760 buffer. (PTW -- Printer and Tape Write). The sequence of instructions is:

SELECT	0214	Select the printer
WRITE	m	Move output record (or last one) into 760
RWS	0006	Print record group
SELECT	0217	Select output tape
PTW	0009	Write record group on output tape

In a 702 program, several "write" instructions may precede the RWS 0006. Although instruction manuals are not specific, it is probable that the

## 1152.26 (Continued)

printing and tape writing are sequential in execution, rather than occurring simultaneously. The reason for the "PTW 0009," rather than an "RWS 0006," instruction for the tape record writing arises from the fact that, in printer operations, the 760 buffer is not loaded solidly, but in sections. (See 4130 for details). Use of "PTW 0009" permits skipping the gaps between sections and recording on tape only the actual data involved in printing.

1152.3 Summary of Type 760 Instructions and Operating Times. The following table summarizes the times required to execute instructions used with the 760; tape and printer times are also shown.

Instr.	Remarks	Operating Time in Milliseconds			
		702	705	Tape	Printer
READ	760 to memory	.138 <del>/</del> .023N <sub>1</sub>	.034 <del>/</del> .023N <sub>1</sub>	*	*
WRITE	Memory to 760	.138 <del>/</del> .023N <sub>1</sub>	.034 <del>/</del> .023N <sub>1</sub>	*	*
RWS 0006	760 and Tape/Prntr	0.163	0.051	10 <del>/</del> .067N <sub>2</sub>	60/line**
RWT 0007	760 and Tape	0.163	0.051	10 <del>/</del> .067N <sub>2</sub> ***	---
RST 0008	Reset storage counter	0.163	0.051	----	---
PTW 0009	Print & write tape	0.163	0.051	10 <del>/</del> .067N <sub>2</sub>	60/line**

N<sub>1</sub> -- Number of characters in a single record

N<sub>2</sub> -- Number of characters in a grouped record

\* -- Not available during 702/705 operating time

\*\* -- Maximum of ten print lines for one instruction

\*\*\*-- The following "read" instruction, not to exceed 4 ms, or "write," not to exceed 8 ms, occurs during tape time

1152.4 Summary of Handling of Record, Grouped and Control Marks. The use of three separate character codes to signify ends of records, or groups of them, and various ways in which they are handled, can cause considerable confusion. This section summarizes their status and use in 760 operations.

1. Units With Which Used:

- a. Control Mark: Peculiar to the 760 storage unit in both 702 and 705 use, on-line and off-line. It never appears in memory.
- b. Group Mark: Peculiar to the 705 and appears in the 760 buffer.
- c. Record Mark: Used in 702, 705 and 760.

2. Tape-to 760 Transfers. The inter-record gap on tape terminates the tape read. A control mark is automatically inserted in the 760 buffer immediately following the last character of the tape record. Record marks on tape are handled as any other valid character. There is no difference in tape reading for either the 702 or 705.

3. 760-to-Tape Transfers. A RWS 0006 or RWT 0007 instruction addressed to a tape in a write status automatically causes a change in the last record mark in the 760 buffer (702), or the group mark (705), to a control mark. The control mark terminates the writing operation and

1152.4 (Continued)

causes the inter-record gap to be created, but is not itself placed on tape. Record marks in the buffer records are handled as any other valid character code.

4. 760 Buffer-to-Memory Transfers. In both 702 and 705 use, a record mark following each record in the buffer, or the control mark following the last one, terminates the transfer.
  - a. In the 702, a record mark is automatically inserted in memory following the last data character. The control mark is automatically converted to a record mark in memory.
  - b. In the 705, the record mark is not inserted in memory. The last character transferred is that immediately preceding the record or control mark in the buffer.
5. Memory-to-760-Buffer Transfers.
  - a. In the 702, a write operation is terminated by a record mark, which is transferred into the 760 buffer. Several records can be loaded successively.
  - b. In the 705, a write operation is terminated by a group mark, which is transferred into the 760 buffer. Record marks within several grouped records are handled as any other valid character.

The group mark of the 705, or the last record mark of the 702, which appear in the 760 buffer, are automatically converted to control marks before tape writing or printing commences.

6. 760-Buffer-to-Printer Transfers. The control mark is inserted in the buffer by the RWS 0006 instruction in the same manner as for tapes. The record mark, or control mark following the last record, stops the transfer of characters for each printed line. Neither type of "mark" is printed.

1152.5 Comments on Record, Group and Control Marks With the 760. From the previous sub-section, it is rather obvious that there is considerable "hardware" detail in handling three kinds of record-terminating marks in different manners. It would be interesting to find out from IBM just why a "control mark"--a completely new character code--is necessary in the 760 when the same function is handled in the Type 777 TRC with the already-existing group mark. (This might be a pertinent question to ask of prospects who are convinced of the logical beauty of the IBM 705 system. A few more are raised a little later).

1152.51 Lack of Record Termination in the 705. It will be noted that the 760-buffer-to-705-memory transfers never include a record mark. This means they must be program-inserted into input areas, as must the group mark necessary to terminate subsequent output operations. An inevitable result is that any file read into a 705 via a 760 must consist of fixed-length records, in a constant number per tape grouped record. Overall, even though the programming is messy, it probably isn't too important, because the 760 is hardly ideal as a tape control unit in on-line 705 operations. (This use is discussed in more detail in 1153).

1152.52 Automatic Conversion to Control Mark. The automatic conversion of a final record mark or group mark has all the earmarks of a potential booby-trap. Let's analyze a few possible cases (which, parenthetically, are all suggested by IBM in its manuals of information).

1. 705 usage. The second of a group of five records in the buffer must be revised and, to save time in memory-to-760 transfers, the storage counter of the 760 is reset to 000 and a "dummy" read, using a glop area of memory, sets the counter to the beginning of the second record. A group mark is program-inserted at the end of the revised record in memory, and a "write" instruction transfers it into its proper place in the buffer, which now looks like this:

Rec. #1 †	Rec. #2 †	Rec. #3 †	Rec. #4 †	Rec. #5	C M
-----------	-----------	-----------	-----------	---------	--------

Just for fun, further assume that Record #5 must be worked on: when re-loaded into the buffer the control mark (C/M) is replaced by a group mark. Now let's write the grouped records on tape. RWS 0006 changes the group mark to a control mark and the record is written, stopping when the control mark is reached. Unfortunately, the tape block now has a group mark terminating Record #2, rather than a record mark. Sometime in the future this tape will be read in again. Now Records #2 and #3 will be transferred into memory together (remember the group mark doesn't stop a 760-to-memory transfer). But suppose Record #3 now has some work to be done on it. Well, obviously it follows Record #2 on the tape--because we laid out the tape we know that much. Easy to handle. Reset counter to 000. Dummy read to a glop area--there's Record #1 out of the way. Another one--there's Record #2. Another good read gives me---whoops! Wha' happened? Record #2--Smith, J. Next record (which we think is #3 but of course is #4)--Smith, N. Well, Smith, M., is a newcomer to the family. Ho, hum. You figure a way out of it.

2. 702 or 705 usage. Let's start with the 760 buffer containing the five records above, and further suppose they are 200 characters each so the control mark is in position 999. This is the contents of the buffer after the tape writing. Now the internal computations in the EDPM develop an exception item, which should be printed out on the directly-connected HSP. This is a 120-character printout and a "write" instruction addressed to this 760, with the printer selected, moves this 120 characters (plus a record mark in the 702 or a group mark in the 705) into the buffer; it goes into the first 121 positions, wiping out most of Record #1 above, because the storage counter was reset to 000 (automatically) after the previous tape write. Now let's print this line. Well, let's see---on the 702 it's followed by a record mark, and that stops the character transfer for one print line. But on the 705 it's followed by a group mark, which doesn't. Ah--but the printing involves a RWS 0006 instruction, which automatically changes the group mark to a control mark---and that not only stops printing one line, but stops printing from the buffer. But, now, wait a minute--there's already a control mark in position 999 and a group mark in 399 (the last position of Record #2 above), plus this one in 120, which

1152.52 (Continued)

is the one we want changed to a control mark. Precisely how does the 760 know this? Ah, from the storage counter, of course. After the transfer into the 760 buffer, it is set at the next character position, so it's easy for the 760 to know that the control mark goes into the previous character position. Really, it's quite simple when analyzed. So that takes care of this situation very nicely. And now, my fine-feathered friend, would you mind taking a look back at the first example, after Record #2 has been moved into the buffer, with the control counter set at the position of the first character of Record #3 when the memory-to-buffer transfer is complete? If the reasoning on this print operation is correct--just how on earth did you get Records #3, #4 and #5 on tape?

The second case is the more important. The 760 buffer is a non-destructive type of storage; i.e., information is replaced character by character only insofar as necessary. It's quite possible (more to the point, a virtual certainty) to be faced with a rather short--say 400 character--record in the buffer, and have all sorts of record, group and control marks left over from previous information in the remaining positions. Just how does the 760 know which one to change into the control mark?

These two examples are facetious treatments of conditions which occur commonly in practice. Their existence is not limited to on-line operations, but could arise in off-line printing. So far as can be determined, there are only two possible explanations: (1) IBM is far from complete in its explanation of how the 760 operates; or (2) its use involves a large volume of housekeeping which is not mentioned or explained. There definitely are inconsistencies and impossible conditions in the methods of operation contained in the IBM information manuals on this unit.

1153 COMMENTS PERTAINING TO ON-LINE USE OF TYPE 760

The principal purpose of the Type 760 is, of course, to serve as a control and buffer storage unit for high-speed printer operations. In fact, the possibility of using it as a tape control unit in CPU work is most likely strictly a by-product of the fact that IBM, consistent with its previous equipments, has seen fit to adapt the high-speed printers to on-line operations under program control.

1153.1 Use of Type 760 as Tape Control Unit Only. The only justification which can be seen for this type operation is the fact that a 760 is necessary for a high-speed printer, and may be available during periods when the printer is not working productively. Even then, it makes for a rather expensive and slow tape control operation:

1. With one tape, the cost is \$2,350 base rental; with two tapes, it is equivalent to \$1,450 each. This is extremely high compared with ten tape units attached to a Type 754, averaging \$750 per tape, or eight with a Type 777 TRC, averaging \$822. (These two pro-rate the cost of the tape control unit).
2. The 760 reads only or writes only; it cannot do both simultaneously. With a maximum tape record size of 999 characters, the maximum transfer rate of characters, reading from one tape and writing on another, is 5,990 per second. This is reduced by the time necessary to unload and load the buffer. The Type 777 can operate at twice this speed as can the Type 754, although the latter cannot overlap tape reading/writing and computation.
3. The 760 buffer and memory transfer rates, at 0.023 ms per character, is extremely slow--this is 23 ms for a 1,000-character record. By way of contrast, the Type 777 unloads a similar record in 9 ms. (And Univac II is of course still faster).
4. In virtually all cases, tape records must be of a fixed and constant length, and the individual items also must be of constant length. Although a variable-length master item is possible on the 705, the fact that neither record nor group marks are transmitted into memory would add considerable housekeeping to a program.

These factors indicate that the only reasonably justifiable uses for the 760 as a tape control unit would involve small subsidiary files during periods when high-speed printing was not being done. It also appears basic that any such usage would permit handling more tape units than the number of 754s or 777s available would permit.

1153.2 Use of Type 760 as On-Line Printer Control Unit Only. An on-line printer is highly useful in program debugging, but Univac philosophy from the beginning has been to divorce mass-output printing from the central processor. With EDPMs of the general logic and characteristics of Univac II and the IBM 702 and 705, there is no economic justification for on-line printing. The closest approach to it occurs when the amount of internal computation is just about equivalent to the time necessary to print one line--in the case of the 760, either 60 ms or 120 ms. Such a balance is

## 1153.2 (Continued)

seldom if ever achieved in practice. The net result is that the printer is "hung up" waiting for the next line to be prepared, or the computer "stalls" waiting for the previous line to be printed. At the costs of these equipments--better than \$20,000 a month for the central processor with its tape units, and from \$3,200-\$3,900 for the printer with its 760--it doesn't take much lost time to add up to considerable money. In the case of the 705, preparation of tapes for off-line printing will always be faster using the Type 777 TRC than with a directly-connected HSP; the difference in buffer loading time is appreciable (0.009 ms per character for the 777 vs. 0.023 ms for the 760).

There is little question that a directly-connected printer can be a convenience. There is no question that it is expensive--six to seven times as costly as a tape unit.

1153.3 Use of Type 760 With Tapes and Printers On-Line. Basically, this combines the remarks of the previous two sections. The new factor introduced is the characteristic of the 760 that permits printing and tape writing of the information in the buffer storage. Unquestionably this is necessary or highly desirable at times. Equally unquestionably, the magnetic tape prepared for an off-line printing operation is exactly the same thing. It also takes less computer time to prepare in the 702 or 705 systems.

In general, there seems to be little economic justification for including a 760 as an on-line part of a 702 or 705 installation. To the extent that the cost of the unit reflects additional hardware necessary to use it in this manner, it can be considered over-priced; it contains more than is needed.

1159 SALES POINTS ON THE TYPE 760 CONTROL AND STORAGE UNIT (ON-LINE OPERATIONS)

Although largely based on the previous discussion, some of these points are also appropriate with reference to high-speed printer operations; these are separately discussed in 4130.

- 1159.01 Contradictory Features of the 760. Point out the examples developed in 1152.52. If nothing else, possibly an explanation of these apparent inconsistencies can be obtained.
- 1159.02 Character Transfer Rate Between 760 and Tapes. Because it reads only or writes only, but not both simultaneously, the effective rate of handling tape information with a 760 is extremely slow. If used at all, in on-line operations, it is probably best confined to reading or writing, but not both.
- 1159.03 Character Transfer Rate Between 760 and Memory. Extremely slow at 0.023 ms per character. Because the EDPM can be doing nothing else during 760 and memory information transfers, it is tied up for an excessive period of time. This is true on either tape or printer operations. Point out Univac's faster buffer unloading and loading--about six times as fast as the 760.
- 1159.04 Cost of Using a 760 as Tape/Printer Control Unit. With two tapes, the per-tape cost of using the 760 as a tape control unit is \$1,450, almost twice as much as using a Type 777 and more than twice as much as the per-tape cost of Univac (pro-rating a part of the Univac II main frame cost to tape control and buffering). Furthermore, both the Type 777 and Univac's rI and rO tanks permit simultaneous reading and writing; this makes the cost-of-performance ratio of a 760-connected tape more than four times that of Univac II.
- 1159.05 On-Line Use of a 760 During Time Not Required for Printing. As noted in 1153.1, it may be that the printing load of an installation does not occupy a full shift, and that the 760 may be available for some 705 or 702 work during this otherwise idle (but paid for) time. This may make it possible to do in one run what might otherwise require two, if the other types of tape control units in the installation are already used to their tape unit capacity. Assuming the use of the 760 in this case does not introduce additional rental, it sounds as if it may have merit.

For the 760: This is definitely a possibility.

Against the 760: There are at least two very definite factors working against such part-time use:

- (1) The off-line printer installation may not be in close proximity to the main EDPM installation; it often is in a separate room. In this case, it may be impossible or impracticable to physically move the 760 and tape units to the computer area. (There is a definite maximum distance that can separate a 760 from the EDPM; the specific distance applicable to the 760 is not known, but most likely does not exceed 50 feet).
- (2) Because the program must be written with the 760 in mind for a specific purpose, the 760 and its tapes must always be available for the machine run. This poses first-rate scheduling problems

1159.05 (Continued)

(particularly when the off-line printer operation has a two-hour breakdown) for both printer and central processor. An additional deterring factor to this type of planning is that both printing and computing workloads and peak periods change as time goes on; it is extremely risky to hang a procedure on the assumption that printing will only take four hours a day next year because that's all it takes today.

1159.06 Inconsistencies Between 760 and 777. Both types of tape control units are equipped with the so-called "early start" feature. The maximum time a buffer and memory transfer can take during the overlapping start of tape motion is:

	<u>760</u>	<u>777</u>
Writing (memory to buffer)	8.0 ms	9.2 ms
Reading (buffer to memory)	4.0 ms	4.6 ms

Both use exactly the same tape unit. It is elsewhere explained why the difference exists in times for reading and writing, but why are the corresponding times different for the 760 and 777? Is it possible that one group of IBM engineers don't know what another has done? This inconsistency is similar in nature to that of the use of the control mark in the 760 which was introduced as a brand new character code doing something apparently handled alright by the existing group mark in the 777.

1160 IBM TYPE 766 DATA SYNCHRONIZER

The IBM Type 766 Data Synchronizer is a control and buffer storage or synchronizer unit used in all Type 709 EDPM input-output operations except those involving information transfers between memory and magnetic drum (Type 733) or memory and cathode ray tube (Types 740 or 780). It is required when using magnetic tapes, or when on-line card reading/punching or printing are involved. Strictly speaking, it is not a tape control unit, but because it incorporates many of the functions commonly associated with such devices, it is included in this section.

The Type 766 has been announced in both Models 1 and 2. There is no known difference in characteristics, function or price, and consequently no distinction is made in this discussion.

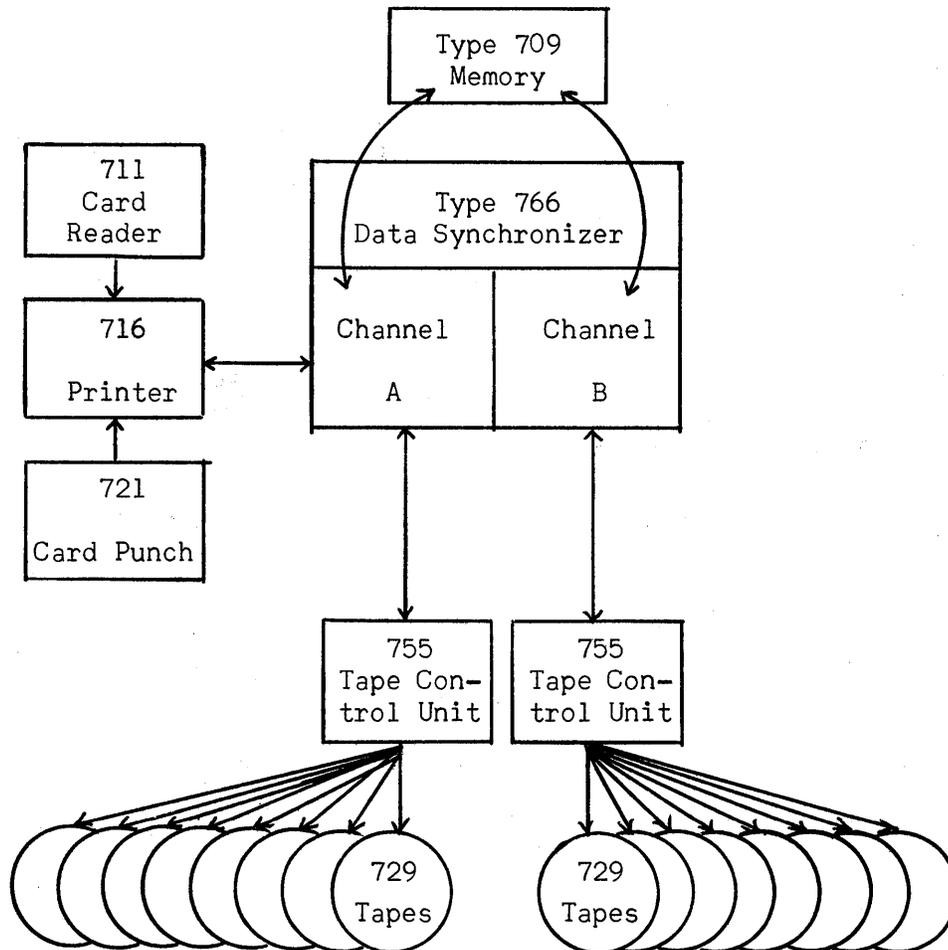
1160.1 Cost: Purchase - \$210,000  
Rental - 3,500 monthly

1161 BASIC CHARACTERISTICS OF THE TYPE 766

Purpose:	Synchronize and control input-output data transfers in Type 709 system
Number associated with 709:	1 to 3 (maximum)
Input-output channels:	2 per synchronizer
Equipments per channel:	1 to 8 (maximum) Type 729 Tape Units (with 1 Type 755 Tape Control Unit)
	1 - Type 716 printer
	1 - Type 711 Model 2 Card Reader
	1 - Type 721 Card Punch
Type of buffer storage:	Magnetic core synchronizing register
Capacity of buffer (synchronizer):	36 bits (one 709 word)
Transfer rate, memory-synchronizer:	12 microseconds per word
Transfer rate, synchronizer-input/output units	Varies with type of unit
Control features:	See 1162

1161.1 EQUIPMENT RELATIONSHIP OF A TYPE 766 IN A 709 SYSTEM

The place of the Type 766 as an input-output synchronizer in a Type 709 installation is depicted graphically in this illustration, which shows the complete array of equipment types which may be associated with a single Type 766. Up to three synchronizers, each with the same maximum array of periphery units, may be attached at one time to the 709 processor.



Any combination of units not exceeding the maximum of each individual type may be associated. The only restriction is that a 716 printer must be used when either a 711 card reader or 721 card punch is attached. Each of the three Data Synchronizers which may be associated with a 709 installation can have a similar maximum number of input-output units, or any lesser number. It is not necessary that each channel have the same configuration of peripheries.

Only one channel in the Data Synchronizer can be used for card equipment (711, 716 or 721): the second channel is available only for tapes. The channels of the first 766 are identified as A and B; of the second, C and D; of the third, E and F. By designation, A, C, and E are used for both tapes and card equipment; B, D and F are restricted to tapes.

1162 METHOD OF OPERATION OF TYPE 766 DATA SYNCHRONIZER

The Type 766 not only provides synchronizing facilities between the 709 memory and input-output units, but also has certain control functions intended to free the CPU of much of the "clerical" instruction execution associated with reading and writing. The nature of these control functions will become clear during the course of the explanation of this section.

The explanation of this section is divided into three phases:

- (1) A general discussion of the features and methods of operation of the Data Synchronizer;
- (2) Specific information on magnetic tape uses; and
- (3) Factors pertaining to the use of card equipment, by which is meant readers, punches and printers used on-line.

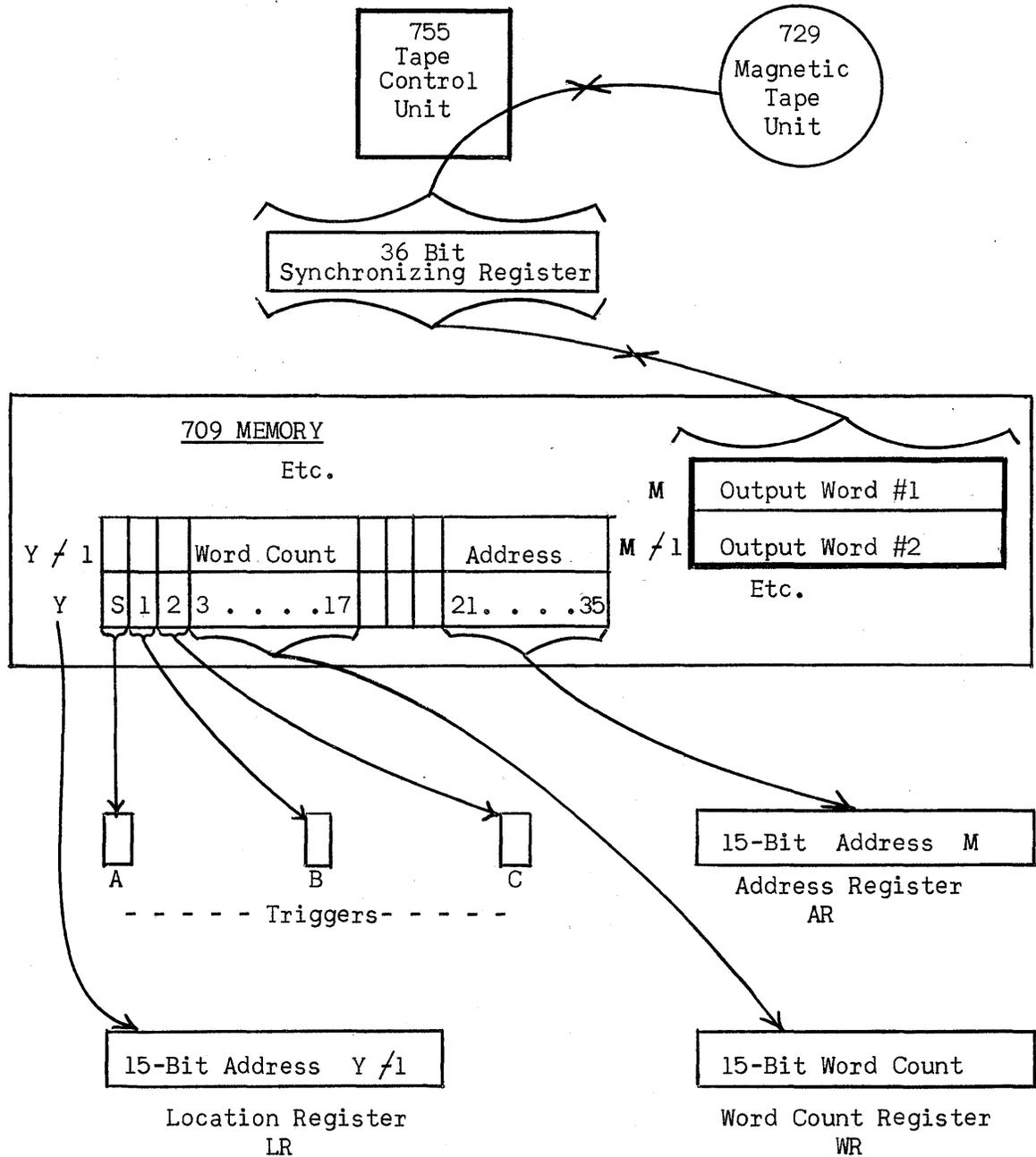
1162.1 Functional Components of Type 766 Data Synchronizer Channel. Each of the two data synchronizer channels (DSC) in a 766 contains hardware to perform some control operations, as well as a synchronizer or buffer to match the internal 709 speed with that of the slower input-output equipment. Illustration 1162.1 on the next page depicts these functional components graphically, and will be found useful in going through the method of operation. The second DSC (B, D or E) is identical with the first, except that it contains only the circuitry used with Type 729 Tape Units and omits that associated with card equipment. In each 766, the two DSCs are completely independent of each other in operation; each contains the features shown in the illustration.

The special registers included in each DSC and a brief statement of their functions follows; their use is further explained in subsequent sections.

- 1162.11 Synchronizing Register. This is a 36-bit register which holds one word from memory while it is being written on tape, or assembles one word (six characters) from tape before high-speed transfer into memory. It thus serves as a one-word buffer between the 709 memory and tape units. Its use is limited to tapes, and for this reason it is sometimes referred to as the Tape Synchronizer. (Although of no importance from a programming viewpoint, it is probable that this register actually consists of two 36-bit halves, the input (or output) to tape alternating between them. This technical detail is not mentioned in preliminary information on the Type 709, but timing considerations indicate this physical configuration).
- 1162.12 Control Word. This is a program constant placed in the Channel by an instruction (or automatically, as will be noted later). The control word contains the information necessary to permit 766 operation independent of main program instructions.
- 1162.13 Word Count Register -- WR. This register contains initially the number of words to be read or written; capacity is 15 bits or a decimal equivalent of 32,767. The contents of WR comes initially from bit positions 3 - 17 of the Control Word. (WR) is automatically reduced by one each time a word is moved to or from memory.

Illustration 1162.1

REGISTERS USED IN TYPE 766 DATA SYNCHRONIZER



This chart shows the makeup of the control word in 709 memory, and its location, with the registers in the 766 into which the various elements of the control word are distributed. The synchronizing register is also a part of the 766.

- 1162.14 Address Register -- AR. AR contains initially the 15-bit memory address of the first word to be read or written; this comes from bit positions 21-35 of the Control Word. (AR) is automatically increased by one each time a word is moved to or from memory.
- 1162.15 Location Register -- LR. LR contains the location of the current control word plus one; i.e., if more than one control word is required for an input/output operation, it contains the address of the next control word. The loading of this 15-bit register is an automatic part of moving a Control Word into the DSC; its contents are set to the location from which it was taken plus one.
- 1162.16 Channel Control Indicator A (Trigger A). The bit in the "S" (sign) position of the Control Word is stored in this trigger. The general interpretation of the bit is:
- 0 - Last control word in an input/output sequence;
  - 1 - More control words following the one being used.
- 1162.17 End-of-Record Control Indicator B (Trigger B). The bit in the "1" position of the Control Word is stored in this trigger. A "1" in this bit position forces a change in a Control Word whenever an end-of-record condition is reached in tape operation; what happens is explained more fully in later sections.
- 1162.18 Trigger C. The bit in the "2" position of the Control Word is placed in this trigger. A "1" in this bit position causes a change in the normal sequence of obtaining new Control Words; this likewise is explained more fully in subsequent sections.
- 1162.19 Summary of Control Word Decomposition in DSCs. The storage of the various positions of the 36-bit Control Word in the DSC occurs as a part of its transfer from memory to the DSC. The bit position distribution is summarized below:

Bit S	--	To Trigger A
1	--	To Trigger B
2	--	To Trigger C
3 - 17	--	To Word Count Register (WR)
18-20	--	Not used
21-35	--	To Address Register (AR)
Y + 1	--	To Location Register (LR), where Y is location of Control Word being transferred into DSC.

The 709 is provided with an instruction which permits storing this channel information in the main memory. The contents of the several registers in the DSC are reassembled into the same Control Word positions, except that (LR) is transferred into bit positions 3-17 and the contents of WR do not appear in the reconstructed word.

1162.2 Basic Method of Input/Output Operation with Type 766. The explanation first outlines the fundamental operating method of the Type 766, with ramifications introduced beginning in 1162.3. For simplicity, tape writing is generally used in the examples; reading is understood to be essentially the converse of writing except where note is made of differences. Notations conform to those in Illustration 1162.1.

1162.21 Instructions Initiating Output (Input) Operations. Data transfer operations are initiated by a sequence of two instructions not necessarily consecutive in a program). The first of these designates the type of transfer (input or output), the Data Synchronizer Channel and the specific unit involved. The second places a Control Word in the DSC registers and initiates the actual transfer of information.

Write (Read) Select Instruction. A "Read Select" or "Write Select" operation code prepares the 709 for the appropriate action with the unit specified by the address part of the instruction. The address includes three factors: (1) The DSC involved; (2) the particular unit (tape or card equipment); and (3) in the case of tapes, the mode of writing or reading (straight binary or character code). The interpretation of the address is in groups of three bits (octal), as follows:

- Bits 21-23 -- Always zeroes and not used.
- " 24-26 -- Selects DSC Channel involved:
  - 1 -- Channel A (Tapes and card equipment)
  - 2 -- " B (Tapes only)
  - 3 -- " C (Tapes and card equipment)
  - 4 -- " D (Tapes only)
  - 5 -- " E (Tapes and card equipment)
  - 6 -- " F (Tapes only)
- " 27-29 -- Distinguishes tape (from card equipment):
  - 2 -- Tape
  - 3 -- Card equipment
- " 30-32 -- (1) If tape is indicated by bits 27-29, designates mode of writing:
  - 0 or 1 -- Character Coded Decimal
  - 2 or 3 -- Binary
 (2) If card equipment is designated, selects unit:
  - 2 -- Card reader
  - 4 -- Card punch
  - 6 -- Printer
- " 33-35 -- (1) If tape is indicated, the specific one of eight maximum is designated (1 through 0, octal, the carry being reflected into bit positions 30-32).
 (2) If card equipment is indicated, this group is always a "1".

Thus, a "Read Select 02207" (octal notation) alerts tape unit #7 of DSC B to prepare to read in character code fashion; "Write Select 03361" alerts the printer attached to Channel C.

Loading the Control Word. The loading of the control word is carried out by executing a "Reset and Load Channel" instruction contained in the main program. This instruction must be executed within 3 ms after alerting a specific DSC for a read operation, or 7 ms for a write. The channel

1162.21 (Continued)

involved is indicated as a part of the operation code itself; thus there are six separate "Reset and Load Channel" instructions in the 709 repertoire. (This is probably necessary, because several "Read (Write) Select" instructions may be encountered before any one of them is followed by a "Reset and Load Channel;" thus the Channel must be repeated).

Assume DSC A is selected. Execution of this instruction transfers the control word, whose location is given as the address part of the "Reset and Load Channel A" instruction, into the DSC A registers, the storage of the various parts of the Control Word being accomplished as depicted in Illustration 1162.1. The Control Word (CW) contains the memory address of the first word to be written, the number of words to be written, and possible bit entries in position S, 1 and 2 of the word. These are moved into the Address Register, the Word Count Register and Triggers A, B and C. Simultaneously, the location of the Control Word plus one (i.e., the address of the "Reset and Load Channel A" plus one) is stored in the Location Register. All of these registers are a part of DSC A and are separate from the registers of the 709 itself. Upon completion of executing this instruction, the 709 central processor is free to continue with other computation; from this point on, the Data Synchronizer Channel circuitry takes over the handling of the input or output operation.

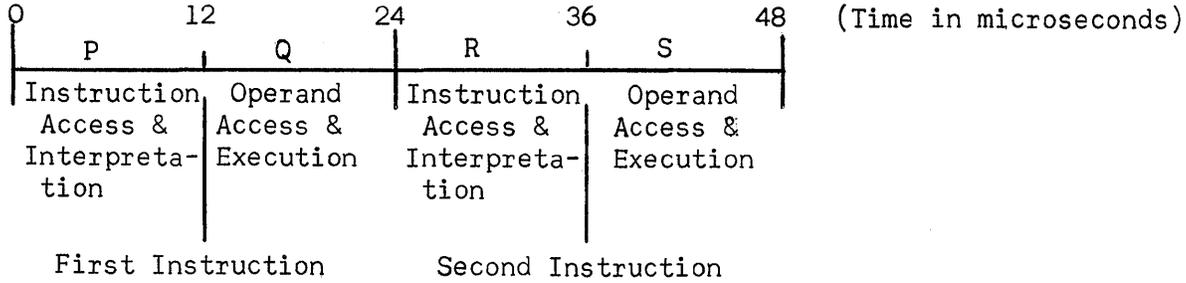
1162.22 DSC Functions in Output (Input) Operations. As soon as the Control Word has been broken up into the various DSC registers, one access (12 microseconds) is made to the memory location indicated in the Address Register and the word found there is transferred to the (Tape) Synchronizing Register. Simultaneously, (AR) is increased by one and (WR) decreased by one. The 36-bit word contained in the synchronizer is now written onto the selected output tape, each group of six bits being written in parallel with a check bit automatically added. Thus one word becomes six character codes on tape; about 400 microseconds are required for this operation at 67 microseconds per character.

When the synchronizer is empty, the DSC inspects the Word Count Register. If this is not zero, more words are to be written and the location of the next output word is contained in the Address Register. At this point, the DSC interrupts the computing operations of the 709 central processor to transfer this one word into the synchronizer. In general, this is an interruption of computing; the DSC "steals" one cycle to make the transfer of the required word. It may occur at any point in a normal 709 instruction; for example, the interruption may come after the first cycle of a two-cycle (24 microsecond) instruction. It always occurs at the end of 709 cycle and always takes one cycle to complete. This causes a 12-microsecond delay in the 709, but does not change any of the registers used in the main processor--all of the necessary information is stored in the DSC registers. The reason for interrupting, rather than overlapping, 709 computing is that most of the 24- and 36-microsecond instructions in the 709 require a memory access on each cycle and only one access is possible per cycle. Thus control circuitry is probably simplified by considering all these short-time instructions as needing accesses each cycle, even though such operations as register shifts require only an instruction access to main memory.

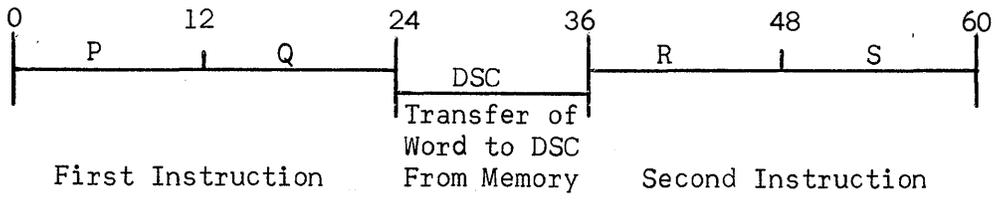
1162.22 (Continued)

Graphically, the timing of the interruption may be depicted quite easily; the normal uninterrupted execution of two successive instructions is shown first, followed by two possible timings of the DSC interruption:

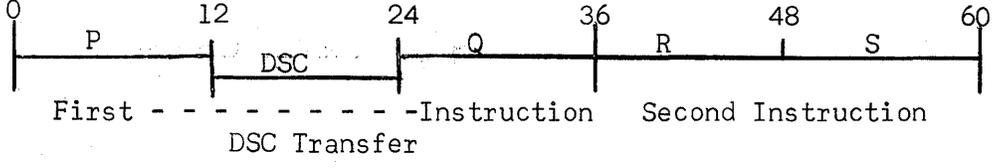
Normal Instruction Execution



DSC Interruption Between Instructions



DSC Interruption Within Instruction



Let us immediately answer the rather obvious question which now arises: "What time does this save--it appears to sandwich in the transfer of information to and from the DSC synchronizer by delaying execution of instructions, adding to total computing time." The answer is that it doesn't save any time in the transfer of data between memory and the buffer--but what difference does it make, overall, if 1,000 words are transferred in one chunk requiring, for example, 12 ms, or in 1,000 separate chunks of 12 microseconds? The total time is the same. What is saved is rather expensive buffer hardware--a one-word synchronizer costs a lot less than a 1,000 buffer (which is nothing but another name for synchronizer). The concept used in the 709-766 is several years old, but this is one of the first equipments announced which will incorporate the feature. As a minor note, it is quite probable that we have seen the last of the large-scale EDPMs with big buffers; they very likely will be limited to equipments now in production or so far along in the design stage that the introduction of newer techniques cannot be incorporated without a major overhaul of the equipment logic. Although the specific method used in the 709-766 is different, a similar approach is incorporated in the design logic of LARC which we are now building.

## 1162.22 (Continued)

There is one class of instructions in the 709 which permits the memory access required by the DSC to actually overlap internal computation; these are the multiply and divide operations, which take from 2 to 20 cycles, during most of which no memory access is required. If a DSC needs a new word from memory while the 709 is engaged in executing one of these instructions, the transfer of the word will occur during the same cycle in which part of the computation is performed. In business data processing, multiplication and division instructions require only a very small part of total computing time and this feature is of little significance; in scientific computations, however, these instructions may take from 15-40% of total computing time, and some reduction in total processing time may be achieved by this simultaneous DSC-memory transfer.

Now back to the synchronizer operation. The write-out of information on tape continues word by word, one memory access being captured by the DSC each time the synchronizer is emptied--in other words, about once every 400 microseconds. Each time the contents of the Word Count Register is inspected and reduced by one, until ultimately it reaches zero. At this point, the fundamental method of the writing operation is terminated; the check character is added to the tape record, the unit is disconnected from the Data Synchronizer and the tape comes to a stop. The Data Synchronizer is available for some other input operation, or output, as soon as the check character has been written.

Reading tape records of known length is basically the converse of writing. The end of the tape record, signified by the end-of-record gap, and the reduction of the Word Count Register to zero occur simultaneously, and the input operation is completed with the transfer of the last word into the 709 memory.

With the basic method of operation thoroughly understood, we are now in a position to go into some of the variations possible.

- 1162.3 Use of Multiple Control Words. If, when the Word Count Register is reduced to zero, Trigger A is "on"--i.e., a "1" was contained in the "S" position of the Control Word-- the writing operation is not discontinued, but instead the DSC obtains a new Control Word from memory and continues writing, using the contents of the new Control Word to determine the next memory word required and the number of words to be written. In other words, successive Control Words can be loaded, as required, in the DSC, each designating an initial memory address and word count. New Control Words are obtained from successively higher memory locations; it will be recalled that the Location Register holds the address of the current Control Word plus one--i.e., the address of the next higher memory location. So long as the "S" position of a Control Word is a binary "1," the DSC automatically secures the next successive Control Word each time the Word Counter is reduced to zero. There is no limit on the number of Control Words which can be used; theoretically, it is possible to write a complete tape solidly from beginning to end using a large number of Control Words. Uses of this feature are discussed beginning in 1163. Thus, the writing operation continues until the current Control Word has a "0" in the "S" position, which means the A Trigger is set to "off."

## 1162.3 (Continued)

In reading from tapes, if Trigger A is "on," end-of-record gaps are ignored and the tape continues reading until the Word Count Register is reduced to zero. Thus several tape records may be read successively and transferred into memory as one larger grouped record. This permits passing over the gaps at full speed in 10 ms, rather than requiring the longer time involved if the tape stops after reading each record.

1162.4 Reading/Writing Under Trigger B Control. A binary "1" in bit position 1 of a Control Word places Trigger B "on" and the input-output operation is under what IBM calls "record control." Because the sequence of events is somewhat different for input as compared with output, the two types of operation are discussed individually.

1162.41 Reading Under Trigger B (Record) Control. Under Trigger B control, when an end of record (inter-record gap) is detected on tape, the next successive Control Word (determined by the address in the Location Register) is transferred into the DSC regardless of the status of the Word Count Register or of Trigger A, and the next following record is read from the tape. Trigger B thus overrides Trigger A. Uses of this feature are discussed in subsequent sections.

1162.42 Writing Under Trigger B (Record) Control. If Trigger B is on during a writing operation, the status of Trigger A must be taken into account. When the Word Count Register is reduced to zero, the next action depends on the status of the A Trigger:

- (1) If the A Trigger is "on," (i.e., a "1" in the "S" position of the Control Word), an inter-record gap is forced on tape and the next Control Word in sequence is moved into the DSC. Writing continues from this point on the basis of addresses and word counts in the new Control Word.
- (2) If Trigger A is "off" (i.e., a "0" in the "S" position of the Control Word), the output operation has been completed, and the DSC disconnects the tape unit involved.

Thus a single "Reset and Load Channel" instruction executed by the 709 central processor may initiate the reading or writing of a large number of tape records without further intervention by the main program. So long as Control Words contain a "1" in bit position 1, Trigger B is turned "on" and one more record is read or written; this cycle continues until the Control Word of the last tape record involved is moved into the DSC; the "0" in bit position 1 turns off the B Trigger and the transfer reverts to the fundamental method of 1162.2.

1162.5 Altering the Sequence of Transferring Control Words. Normally, successive Control Words are obtained sequentially from memory locations indicated by the address in the Location Register. There are two methods of altering the normal sequence of obtaining Control Words, one automatic and one under program control.

1162.51 Altering Sequence by Trigger C. If in a sequence of Control Words, one contains the bits "001" in the S, 1 and 2 positions (Triggers A and B

1162.51 (Continued)

off, C on), the normal use of the Control Word does not apply. Instead of being taken as the next Control Word, the address portion only (bit positions 21-35) is used as the location of the next Control Word; this address is placed in the Location Register and an access made to that memory location for a new Control Word. Succeeding Control Words, if called for, are obtained in the normal manner beginning with this new location. Thus, when Trigger C is on, the normal location of the next Control Word is used only to obtain the address of the one actually to be used; this is virtually identical to the "indirect addressing" used in the 709 central processor.

1162.52 Altering Sequence by Main Program Instruction. The second method of changing the normal sequence of obtaining Control Words is through use of a "Load Channel" instruction in the main program. The initial "Reset and Load Channel" can seldom be used, because it is executed immediately, regardless of the status of the DSC operation which may be going on.

The "Load Channel" instruction differs in that it does not take effect until completing the action required by a Control Word which has a binary "1" in bit position 2 and also in position S and/or 1—in other words, Trigger C and Trigger A and/or B set to "on." With this bit combination, the DSC will do one of two things:

- (1) Execute a "Load Channel" instruction provided one is waiting; or
- (2) Disconnect the selected input/output unit and terminate the operation if no "Load Channel" instruction is waiting.

In either event, the action specified by the Control Word containing the necessary bit combinations in the S, 1 and 2 positions is completed before the "Load Channel" is executed or input/output terminated. Use of this program-interrupt feature requires synchronizing the input/output routine with the main program because the "Load Channel" instruction must be available at the time the DSC completes the action of a Control Word having the specified bit combinations.

1162.6 Status of Data Synchronizer Registers. It may be necessary, during the course of a main program routine, to know the status of a specific input-output operation previously initiated. This is accomplished by a "Store Channel" instruction, which transfers the contents of the several DSC registers into a memory location indicated by the address portion of the instruction. The reconstructed word is similar in format to a Control Word, except that the contents of the Location Register is inserted in the normal Word Count bit positions, and the Word Count Register is not sent into the reconstructed word. The status of the three Triggers, of the Address Register and the Location Register are transmitted. This instruction interrupts, but does not terminate or alter, the DSC operation. The instruction execution is interlocked to prevent the transfer of the register information during the time a new Control Word may be in process of movement, or during a change of the contents of the Word Count and Location Registers. The normal input/output sequence being carried out by the DSC continues without change.

1162.7 Testing Status of Data Synchronizer Channels. Two conditional transfer instructions are included in the 709 repertoire to test the condition of a selected DSC. These are "Transfer if Channel A in Operation" and "Transfer if Channel A Not in Operation," with similar instructions for the other five channels. A channel is considered to be in operation if it is not ready to execute a new input/output instruction.

1162.8 Check Indicators and Verifying Accuracy of Data Transfers. A number of special indicators are provided for each of the channels to permit program testing of various conditions applicable to input-output units and of the accuracy of the data transfer. These are:

- (1) End-of-tape tests; (one in each DSC and in each tape unit)
- (2) Beginning-of-tape tests;
- (3) Redundancy indicator, turned on when a check bit or check character error occurs;
- (4) End-of-file indicator (turned on by a tape mark).

The redundancy indicator is operative whether or not the contents of a tape record are being transmitted to memory. It should be noted that it is not necessary to read into memory an entire tape record; a dummy control Word can be inserted when it is discovered that a particular record is not required, eliminating the transfer of the contents of the synchronizing register; the tape, however, continues moving until an inter-record gap is reached and the normal testing of tape characters is operative during this period.

1162.9 Interaction of Triggers A, B and C; Word Count; Tape Records. The various actions taken by a DSC after transmitting each successive word of information are determined by the setting of the three Triggers and of the Word Count Register; in addition, in tape or card reading, the end of record (card) also must be considered. The various possibilities are summarized in the tables on the next page, which groups the combinations causing each type of DSC operation. In these tables, "W" is the contents of the Word Count Register, "0" being zero and "Ø" being non-zero. These tables are included for reference purposes only; the basic methods of operation of the DSC with magnetic tapes have been outlined in the previous paragraphs of this section; specific comments applicable to card equipment are contained in 1163.

Table 1162.9 (1)

Tape and Card Reading Operation of DSC

<u>Triggers</u>				<u>End of Record</u>	<u>DSC Action</u>
<u>A</u>	<u>B</u>	<u>C</u>	<u>WR</u>		
0	0	0	∅	No	Continue transmitting words from I/O unit into memory
1	0	0	∅	No	
1	0	1	∅	No	
0	1	0	∅	No	
0	1	1	∅	No	
1	1	0	∅	No	
1	1	1	∅	No	
1	0	0	0	No	Bring in next Control Word from location specified by Location Register
1	0	0	0	Yes	
0	1	0	0	Yes	
0	1	0	∅	Yes	
1	1	0	0	No	
1	1	0	0	Yes	
1	1	0	∅	Yes	
0	0	1	0	No	Bring Control Word into DSC from location specified in address part of this Control Word
0	0	1	0	Yes	
0	0	1	∅	No	
0	0	1	∅	Yes	
1	0	1	0	Yes	If "Load Channel" instruction is waiting, execute it. If not, disconnect input/output unit.
0	1	1	0	Yes	
0	1	1	∅	Yes	
1	1	1	0	Yes	
1	1	1	∅	Yes	
1	0	1	0	No	If "Load Channel" instruction is waiting, execute it. If not, disconnect unit but move it until it reaches end-of-record
1	1	1	0	No	
0	0	0	∅	Yes	Ignore end of record and continue transmitting words from I/O unit into memory
1	0	1	∅	Yes	
1	0	1	∅	Yes	
0	0	0	0	No	Disconnect unit from 709, but move it to end-of-record.
0	0	0	0	Yes	Disconnect unit from 709
0	1	0	0	No	Read tape without transmitting words to storage. At end of record, bring in Control Word from address specified by Location Register.
0	1	1	0	No	
0	1	1	0	No	Read tape without transmitting words to storage. At end of record, if "Load Channel" instruction is waiting, execute it. If not, disconnect unit.

"∅" means "not zero"

Table 1162.9 (2)

Tape and Card Writing Operation of DSC

Triggers

<u>A</u>	<u>B</u>	<u>C</u>	<u>WR</u>	
0	0	0	}	Continue transmitting words from memory to I/O unit
1	0	0		
1	0	1		
0	1	0		
0	1	1		
1	1	0		
1	1	1		
1	0	0		
1	1	0		
0	0	1	}	Bring Control Word into DSC from location specified in address part of this Control Word
0	0	1		
0	0	0	0	Write end-of-record designation and disconnect unit
1	0	1	}	If load channel instruction is waiting, execute it. If not, write end-of-record and disconnect unit
0	1	0		
0	1	0	}	Write end-of-record indication and bring in new Control Word
0	1	1		
0	1	1	}	Write end-of-record indication. If "Load Channel" instruction is waiting, execute it. If not, disconnect unit
1	1	1		
1	1	1	}	If "Load Channel" instruction is waiting, execute it. If not, disconnect unit

1163 METHOD OF OPERATION OF TYPE 766 DATA SYNCHRONIZER (Continued)

Section 1162 outlined the basic characteristics of the Type 766 DSC and the general method of operation. Some of the comments pertained only to magnetic tape units, because some types of operations are peculiar to that type of input/output device. This section discusses in more detail the use of the DSC in operations concerned with both magnetic tapes and card equipment. The last part of this section has some summary comments on the use of DSCs in the 709 system.

- 1163.1 Modes of Reading/Writing With Magnetic Tapes. Section 1162.2 describes the basic method of operation of the DSC in reading or writing, and these general comments apply fully to magnetic tapes. The basic additional note to be covered is the observation that the 709, like the 704, is capable of two separate modes of writing on magnetic tape: Binary or character-coded decimal. Which method is to be used in a given operation is determined by the address of the instruction selecting the tape unit; each tape unit has two logical addresses, one for each method of writing or reading.

If information is to be written on tape in binary recording, the successive words of 36 bits each are recorded as six groups of six bits on the tape. To each group of six bits an odd check bit is automatically added; thus each word from memory becomes six separate character codes on tape, each group of six bits and the added check bit being written successively. There is no change in the six data bits recorded; they are simply transferred to the tape. Reading is the converse of writing, the six bits from each character code on tape being transferred without change into the synchronizing register. The check bit is not included in this assembly of successive characters on tape, but is checked to determine validity of reading and a check error arises if the bit is invalid.

If information is to be written on tape in character-coded form, the successive groups of six bits in the words moved into the synchronizing register are not written as they appear, but are automatically converted from the internal 709 code to the 702/705 character codes. (Refer to 5152 for a table of the internal and external character codes used in the 704/709 systems). In this case, an even check bit is added to each group of six bits after the conversion; the resulting recording is identical with that of the 702/705 systems. It should be noted that information in binary form cannot be converted to a character-coded form, but must be written on tape in binary. The reason is fairly evident--there are only 48 valid character codes of six bits each in the internal language, although 64 are possible; the 16 invalid bit combinations could not be converted and most likely would give rise to an input/output error. Reading is the converse of writing--the tape characters are automatically converted into the internal codes and assembled into the synchronizing register before transfer of each word into memory.

The "READ" instruction must specify whether binary or coded characters are to be read; because of the check bit differences it is obvious that the equipment must know in advance which type of recording is being used as input. Although there appears to be no reason mixed types of recording

## 1163.1 (Continued)

could not be combined on a single output tape, the difficulties of providing the proper read instruction for each successive tape record precludes this as a practicable method. Any one tape reel will be in binary, or in character-coded decimal, but not both.

It should be observed, however, that anything in the 709 can be written as binary information, even though the information actually represents character codes. The distinction of data as being binary or character-coded is purely human; the 709 doesn't know the difference and handles everything internally (with a few exceptions) as if it were binary. In use, character-coded magnetic tape will be used for off-line printing, punching and card reading functions, or for preparation of tapes to be used as input into 650, 702 or 705 installations. Tapes, including master files, which are to be used as later input into the 709 itself will almost always be written in the binary mode.

- 1163.2 Use of Type 711, Model 2, Card Reader With Type 766. One channel of a DSC may have attached to it, under program control, one Type 711, Model 2, card reader. It is also required that a Type 716 Printer be attached in order to use the card reader. Only one card reader can be associated with each Data Synchronizer, using only Channels A, C or E. Thus the maximum in one 709 installation is three.

The basic method of card reading is described in section 3110 and is not changed in 709 usage. 72 columns of the card are read and converted into 24 "card image words," each word having a bit combination corresponding to the presence or absence of holes in 36 columns of one row of the card. These 24 words are read sequentially and transferred into memory, through the synchronizing register, according to data in the control word. The principle of the transfer is identical with that of reading from magnetic tapes. The basic difference is in the timing of the movement of the successive word images; the DSC automatically adjusts itself to the timing of the card reader. This is explained in 3110.

It is important to bear in mind that the information transferred into memory is in card image form and thus is neither binary nor character coded (unless the card read is punched in binary fashion as discussed in 3110; this is almost exclusively peculiar to scientific input and does not occur in punch card business input). A rather extensive conversion program is necessary to change the 24-word card image into either straight binary or character coded form for internal 709 processing. It does not appear that the special "conversion" instructions incorporated into the 709 are particularly applicable in this case; if this is true, a long and time-consuming program will be necessary to effect the conversion.

- 1163.3 Use of Type 716 Printer with Type 766. One channel of a DSC may have attached to it, under program control, one Type 716, Model 1, printer. Only one can be associated with each Data Synchronizer, using only Channels A, C or E. The maximum in one 709 installation is three.

The basic method of printing is described in 4110 and is not changed in

1163.3 (Continued)

709 usage, and is essentially the reverse of the card reading mode summarized in 1163.2. Each word transferred from 709 memory through the DSC to the printer is a 36-bit representation of 36 columns of one card row; thus 24 words must be transferred to permit printing a 72-character line. Information maintained in either binary or character-coded form in the 709 must be program-converted to the proper punch card word images. As in the card reader, the DSC automatically adjusts the transfer of words from memory to the timing necessary for the operation of the printer.

- 1163.4 Use of Type 721 Card Punch With Type 766. One channel of a DSC may have attached to it, under program control, one Type 721, Model 1, card punch. It is also required that a Type 716 Printer be attached in order to use the punch. Only one card punch can be associated with each Data Synchronizer, using only Channels A, C or E. The maximum in one 709 installation is three.

The basic method of card punching is identical with that of printing, and is discussed in section 4510. As in printing, the DSC automatically adjusts the transfer of words from memory to the timing necessary for the operation of the punch. A maximum of 72 columns of a card can be punched.

- 1163.5 Interruption of Data Transfers. Each Data Synchronizer Channel is capable of handling one--and only one--data transfer at a time. That is, if it is writing on one tape, it cannot simultaneously read a punch card, or read from another tape. Once an input/output unit has been selected, and electronically connected to the DSC, it commences the required operation and continues with that operation until one of these two conditions occurs: (1) Failure of the DSC to receive the first control word within a specified time limit after the unit has been selected--the time varies depending upon the type of unit involved; or (2) completion of the specified transfer as indicated by the word count and status of Triggers A, B and C held in the DSC.

In some applications, however, "real time" data of an intermittent nature may be used as input to a 709; such information usually requires immediate processing, which means interrupting whatever the EDPM is doing at the moment to process the urgent data. The only means of input to the 709 is through a Data Synchronizer Channel; where real time information is involved, it may then be necessary to interrupt an input or output operation to permit receipt of these data. The exact method of doing this is not detailed in available information on the 709, but the method of accomplishment is this: A control or key word with a specified bit combination is fed into the 709 (in a manner not described) where a matching bit pattern is retained. Receipt of this key word automatically transfers program control to memory location 0004 and simultaneously stores the location of the next instruction in the interrupted sequence in location 0002. If the DSC through which the urgent data is to be received is busy in some input or output operation, the transfer is stopped by a "Reset and Load Channel" instruction which has a zero word count and a "0" bit for Trigger A. This instruction would be included in the sequence beginning in location 0004; the entry of a new Control Word into the DSC is forced by the instruction, and the unit involved in the transfer is dis-

1163.5 (Continued)

connected, freeing the DSC for the "real time" information. It should be noted that entry of the key word into the 709 does not of itself stop a DSC transfer; this is done by programming included as a part of the subroutine beginning in location 0004. The actual stoppage is accomplished by the "Reset and Load Channel" instruction, which places a new Control Word into the DSC registers.

Rather evidently, a "Reset and Load Channel" instruction which would stop input/output action could be executed at any time in a program for any reason, including whims of the programmer. An example of its use in other than "real time" instances is in the reading of a fairly long string of tape records with a sequence of Control Words. A tape reading error could occur early in the string and it would then be desirable to terminate the reading to permit a corrective re-read subroutine to be entered. This would be done by inserting several instructions testing the status of the error indicators at periodic intervals--say every 20 ms. If an error is detected, the program transfers to a corrective subroutine which includes a "Reset and Load Channel" instruction to stop the reading immediately, rather than cycling through the remaining tape records specified by the normal sequence of Control Words. The use of this instruction, then, is a third means of stopping the transfer of data through a DSC.

1163.6 Reason for the Number of Data Synchronizer Channels. The 709 can be equipped with up to three Data Synchronizers, or a maximum of six channels. This limitation in number does not appear to be based upon any processing considerations--i.e., there is no indication by IBM that six is the optimum or maximum number of inputs and outputs which can be handled by the 709 internally--but is a limit imposed because of technical considerations.

The reason is this, using writing on tape as an example. When the 36 bits in the synchronizer have been written on tape, the next word must be obtained from 709 memory and moved into the synchronizer and this must be done by the time the next character is to be written on tape. The transfer of the word takes 12 microseconds. But now let us look at the worst possible condition. Suppose the last word written on tape reduced the word count to zero, but did not terminate the writing sequence. This means that another Control Word must be moved into the DSC. Further, it might be an indirectly-addressed Control Word. This means that it is possible, in the worst case, to require an access for the indirectly-addressed word, another access for the new Control Word and, finally, an access for the information word to be written on tape. This total of three accesses requires 36 microseconds. With six DSCs, it is possible--again in the worst case--that each of the six requires exactly the same treatment. In the worst case, a total of 18 memory accesses, requiring 216 microseconds, might be required. With a nominal 400 microseconds tape time for handling one word, and the engineering requirement for a good margin of tolerance on this figure--an insistence on something like 50% is customary--it is apparent that this "worst case" is an engineering limitation. It is necessary to have sufficient time to take care of the worst possible occurrences, and addition of one more Data Synchronizer, which would add 72 microseconds to the time requirement, would add to more than can be considered safe.

## 1163.6 (Continued)

This summarization also makes it evident that the "synchronizing register" is not a 36-bit affair, but rather consists of two 36-bit halves which alternate back and forth. With only one 36-bit register, it would be necessary to obtain the next word within 67 microseconds, less some tolerance; this is the nominal time between successive tape characters. It is obvious that  $6 \times 12$  is 72, and with single registers it wouldn't even be possible to move six new data words into the DSCs within one character time, let alone provide for the possibility of new Control Words. This is a purely technical consideration; the programmer doesn't need concern himself with this fact.

1164 INPUT/OUTPUT TECHNIQUES POSSIBLE WITH TYPE 766 DATA SYNCHRONIZER

The control circuitry included as a part of each channel of a data synchronizer makes possible a number of input-output techniques which, heretofore, have been beyond the scope of EDPM capabilities. One of the basic purposes of the unit is to free the central processor of much of the so-called "clerical" effort associated with input-output operations. There is no question that the DSC logic incorporates some highly desirable features, although some complexities have been introduced and not all characteristics are without disadvantages.

At this time, there is no available information on the various input-output operations and techniques which IBM will advocate as promoting the most efficient use of the 709 system. Consequently this section is not exhaustive, but it does include a number of possibilities which are immediately obvious from the characteristics of the DSC and some comments on their general applicability. The following discussion should therefore be considered only as preliminary; as time permits, more detailed analysis will be made and the information contained herein supplemented or revised as necessary.

1164.1 Elimination of Internal Memory Transfers. Probably the most significant single characteristic typifying the 709 EDPM system is a definite effort by IBM to virtually eliminate multi-word internal memory transfers. Three characteristics of the equipment shirk this conclusion:

- (1) Absence of any form of multi-word memory transfer instruction;
- (2) The "indirect addressing" scheme incorporated into both the 709 central processor and DSCs; and
- (3) Accomplishment of input-output operations by the use of multiple Control Words, if desired.

The attempt to circumvent the normal requirements for internal transfers is, of course, aimed essentially at freeing the central processor from a time-consuming clerical chore. As such, it is not concerned directly with the Data Synchronizers. Nonetheless, it is the inclusion of the multiple Control Word feature that completes the logic necessary to attempt the scheme. In short, a fundamental objective of the 709 system is to eliminate multi-word internal memory transfers and to substitute in lieu thereof single-word transfers of words which become, ultimately, DSC Control Words.

The internal 709 characteristics which are intended to accomplish this objective are discussed in section 5150. The possibilities and techniques included in this section are concerned largely with DSC use, and mention the 709 only to the extent necessary to understand the basic methods of operation possible.

1164.2 Sequencing of Words as Part of Output Operation. Through the use of multiple Control Words, each of which can transfer the source of output information to a new memory location, it is possible to assemble individual records into an index-number sequence as a part of the output operation, rather than by internal memory transfers. This likewise can be done as a part of input to the 709, but the utility of the technique is not so obvious as it is in output. There are two major areas of interest in EDPM

## 1164.2 (Continued)

applications--sort-merge functions and the reassembly of master file information to incorporate additions and deletions into the file.

- 1164.21 Sort-Merge Technique. Sorting and merging in the 709 will consist not of sequencing the individual records by means of multi-word memory transfers, but rather of arranging the locations (i.e., memory addresses) of the index words into a sequence such that the individual records will be in proper order if they are taken in the succession of location words. This is accomplished by placing the addresses of the index words into a series of other words and, through "indirect addressing," sorting these location words into the desired sequence. (The internal use of "indirect addressing" and amplification of this technique are contained in the discussion of the 709 central processor, 5150). As a final fillip, these location words become Control Words for output operations, and the actual sequencing of the complete individual records does not occur until they are written on tape. Thus the internal memory transfers normally associated with a sort-merge operation reduce to transfers of one-word "location words," which are so set up in format that they are directly usable as DSC output Control Words.

For example, the record size may be such that a thousand of them can be accommodated comfortably in memory, which would require the use of 1,000 Location-Control Words. Initially, these would all have a "binary" "1" in the S position; after the sequencing of the Control Words and prior to initiating the writing operation, a "0" would be placed in the 1,000-th word. The first word is placed into the applicable DSC with a "Reset and Load Channel" instruction, after which the "1" in the S position (which turns on Trigger A) causes a new Control Word to be moved into the DSC every time the Word Count is reduced to zero (i.e., every time a complete record has been transferred to tape). The "0" in the S position of the last Control Word stops the write-out, and the entire 1,000 records have been placed on tape in proper sequence. The same basic principle can be used for either sorting or merging, although the details vary depending upon the type involved.

Insofar as the storage on tape of a sequenced string of records is concerned, it may be noted that there is no particular reason for them to be all of the same number of words. Each Control Word has its own Word Count, and these can be different in successive Control Words. The writing operation, then, can easily accommodate records of varying lengths. Reading them back in for subsequent merging passes adds a few difficulties, requiring some additional internal programming, but presents no insurmountable barriers. Again, refer to the 709 central processor discussion for further details. This brief explanation can be concluded with one additional statement: The sorting of variable-length records may well be economic in the 709-766; it certainly is always possible.

- 1164.22 Master File Tape Block Reassembly. The conventional techniques of providing for additions and deletions within a master file call for the reassembly of an input block into parts of two output blocks by means of multi-word internal memory transfers. Not only does the 709 not have such an instruction, but it is quite evident that the technique to be advocated

1164.22 (Continued)

will consist of creating Control Words to accomplish additions and deletions of information. This certainly can be done--the 709 has a number of "bit manipulation" instructions which facilitate the creation of such Control Words, although at this time the optimum approach is far from clear. One possible method is outlined briefly in the next paragraph, not for the reason that it is of necessity particularly apropos, but because it illustrates a use of the "indirectly addressed" Control Word, (See 1162.51), a feature whose application is not too obvious.

Suppose a master file is maintained in tape blocks of 1,000 words maximum size. Now establish 20 output Control Words, each moving 50 data words from memory to the tape. Imagine also that the fourth group of words (in other words, in the 151-200 range) has a 10-10 word deletion. This could be handled in this fashion: Replace the fourth Control Word with one containing the Trigger C control and an address which is beginning of a "Control Word subroutine." The subroutine Control Words are created to write out, say, 15 words by the first one and 25 by the second, the address of the first starting the writeout at the beginning of the 50-word group and the second after the 10-word deletion. These two new Control Words are followed by another under Trigger C control which returns to the main sequence of Control Word instructions. This appears to be a quite logical technique; further, it tends to corroborate the belief that the indirectly addressed Control Word is of most utility as a means of initiating a subsequence altering the normal course of a read or write operation.

1164.3 Multiple-Block Reading and Writing. The maximum effective transfer rate of information between tapes and memory is achieved with long tape blocks, which minimize the occurrence of inter-record gaps and the time necessary to pass over them. Extremely large tape blocks require, of course, considerable memory space if the entire block is to be held. However, the percentage increase in effective transfer rates drops rapidly as blocks become longer. For example, a 500-word 709 block results in an effective tape speed of about 95.5% of maximum; increasing the block to 1,000 words boosts this to about 97.6%. The effective transfer rates for the two tape block sizes are about 14,300 and 14,600, respectively, compared with the 15,000 instantaneous rate. Rather evidently, any increase above 1,000 words (6,000 tape characters occupying 30 inches) gives only a marginal improvement in tape transfer rates.

Several factors indicate that shorter tape blocks are also a common occurrence--the requirements of periphery units is a good example, and the incidence of tape read/write errors also mitigates against extremely long blocks. In some instances, file format considerations, particularly with variable length records, may also tend to shorten the number of inches of tape packed with characters before a gap is reached. In these cases--and they are probably going to be quite common--some tape time can be saved if the gap can be covered at full speed, rather than having the tape stop for each one.

Break-up of an output into several records or reading of multiple blocks on input are both possible with the Data Synchronizer; the settings of the three triggers and the status of the Word Count Register which make

## 1164.3 (Continued)

this possible are shown in Tables 1162.9 (1) and (2). This facility has the effect of permitting multiple blocks to be read and written with only one initiating action by the central processor and, secondly, of giving a somewhat higher effective transfer rate by covering the inter-record gaps at full tape speed.

Some techniques known to be in use on the Type 704 EDPM indicate that the following possibility of handling variable-length master file records may be advocated by IBM. The basic premise of the technique is that records vary not only in the number of detail elements which may be appended to them, but also may have a significant percentage of "zero" entries within the elements which exist. For example, a stock control file may have words for on hand quantities, back orders, due ins and total demands for the past three months. For many of the stock items, some or all of these may be zero. Comparable conditions exist in many types of business files. The method of approach is this:

One word is set up as a "control word" for 36 other possible ones, a "1" in a bit position indicating that the corresponding word is present and a "0" that it is not. The only words which are included in the file are those corresponding to "1s." If a word has a non-significant entry, it simply does not appear at all. Thus, if a control word has at its first ten bits a "1110000101," only five words of information follow; the bits in the control word are used to identify their nature. With this technique, individual records within one file can, of course, vary widely in the number of words of information associated. In general, considerable input-output time can be saved by handling records in this variable-length form.

Suppose the maximum size of any given item is 100 709 words, and that memory space for ten such items is available. This makes it possible to read or write ten at a time, but it would be highly convenient to know where they were in memory; considerable computer time can be spent in "bootstrapping" through variable length records.

First of all, let's write out ten such records. The ten Control Words each contains as Word Count the actual number of words in the corresponding record, and each record begins (assume) in a memory location ending in "00," and of course the memory addresses in the Control Words are each 100 higher than the preceding one. With a bit combination "010" in the S, 1 and 2 positions of the Control Words, this permits writing out each of the ten records as an individual record on tape--i.e., a gap is "forced" between each one. Because each Control Word has the actual word count of the item, only the significant words of information are written out.

The read-in is likewise accomplished with the "010" bit combination in all Control Words, each with a Word Count entry of "100". (The maximum individual record size). The memory addresses in these Control Words are also the successively higher "00" locations. With this type of control (Trigger B or end-of-record), either reduction of the word count to zero or presence of a gap on tape causes a new Control Word to be loaded into the DSC. In other words, each record is read into a location beginning

## 1164.3 (Continued)

at a memory address ending in "00", and continuing in successively higher addresses until then inter-record gap is reached on tape. The next tape record is stored beginning at the next higher "00" address. Thus, regardless of individual lengths, the beginning memory address of each record is in a known location and, at the same time, the tape time necessary for transferring "pad characters" or zero entry words is eliminated.

These features of the DSC make available for the first time an IBM data processor which is considered flexible enough to handle the variable-length records which comprise the vast majority of business files.

1164.4 Passing Over Unwanted Information. In some processes, it is necessary to read over large volumes of information to secure a relatively small amount needed for actual processing. If the file does not require updating (that is, if it does not need rewriting), considerable time may be saved internally in the EDPM if the transfer of unneeded records into memory can be avoided. Usually, this requires that the first few characters or words of each tape block be read and tested, but if this can be accomplished quickly, it may be possible simply to pass over most of the block without actually transferring it into memory.

Although there are other methods of accomplishing essentially the same thing, the logic of the 709 requires that two conditions be met before the objective can be realized:

- (1) It is necessary to know when the required number of words to be tested have actually been read in from the tape and transferred to memory; and
- (2) The equipment must be capable of suppressing the transfer into memory at any point along the tape block.

Both of these requirements are fairly obvious.

In the 709-766, the first is met by the "Store Channel" instruction (see 1162.6). This permits a program to obtain access to the status of the various DSC registers and, by program testing, to determine if the required number of words has been transferred into memory. If they have, it is known that new information is in the low-address end of the input area and the necessary tests can be made to determine if the item is required for the computing process. If it is not, the program can execute an immediate "Reset and Store Channel" instruction, with a zero word count entry and a bit in the 1 position to turn Trigger B on. This stops the transfer of information into memory, but the tape continues moving to the end of the record. This takes care of the second requirement listed above.

1164.5 Continuous Reading/Writing. The nature of the memory accesses for data transfers to and from the DSCs makes it evident that there is no necessity of waiting for a complete block to be transferred into memory before writing it out. As a matter of fact, it is perfectly possible for one DSC to access a specific memory location on one cycle and another DSC access it on the next cycle. Provided necessary computing can be done in

## 1164.5 (Continued)

the time available, there is no logical reason to prevent an input tape packed solidly from beginning to end (no gaps) being read in, processed and written out on an equally-solid tape. For example, the input can be read into 1,000 words of memory, the Control Words transferring the input back to the beginning location at the end of every 1000th word. Output can be on the same cycle, for the same memory area, lagging the input by, say, 500 words. The only operating requirement is that of keeping the input and output in reasonable synchronism; tape unit variations mean that the reading rate will differ, normally, from the average by a slight amount. Thus it is possible for reading to be slightly faster than the average character rate, which means that the input might eventually "catch up" with the output; or, conversely, the reading may be slower, with the result that the output would overtake the input. The occasional insertion of "Store Channel" instructions would permit checking the lag between output and input and, if required, taking some corrective action.

As stated above, the idea is purely idealistic. The accuracy of tape reading would dictate some much shorter tape block size, and the increase in effective transfer rates is so minute with blocks of more than 1,000 709 words that longer ones give no significant gains in speed and may have offsetting disadvantages in terms of accuracy. These are, however, practical considerations; there is certainly nothing in the DSC logic that makes them impossible.

INDEX

IBM Type 777 Tape Record Coordinator

- 1180 IBM Type 777 Tape Record Coordinator (General statement)
- 1180.1 Monthly Rental
- 1181 Characteristics of the Type 777 TRC
- 1182 Operating Characteristics of the type 777 TRC
  - 1182.1 TRC-Tape Transfers (Operating Mode A)
    - 1182.11 Tape to TRC (A1)
    - 1182.12 TRC to Tape (A2)
    - 1182.13 TRC to Tape and Tape to TRC (Read While Writing) (A3)
    - 1182.14 Comments on Operating Mode A
  - 1182.2 Memory-TRC-Tape Transfers (Operating Mode B)
    - 1182.21 Memory-to-TRC-to-Tape With Early Tape Start (B1)
    - 1182.22 TRC-to-Memory and Tape-to-TRC with Early Tape Start (B2)
    - 1182.23 Memory-to-TRC-to-Tape and Tape-to-TRC (B3)
    - 1182.24 Comments on Operating Mode B
  - 1182.3 Memory-TRC Transfers (Operating Mode C)
  - 1182.4 "Microsecond Interrogation" (Operating Mode D)
  - 1182.5 By-Pass Coordinator (Operating Mode E)
  - 1182.6 Checking in the TRC
  - 1182.7 Testing Status of TRC
  - 1182.8 Summary of TRC Operating Times and Modes
- 1183 Length of Tape Records and TRC Operations
  - 1183.1 Effect of Length of Grouped Records of Constant Length
  - 1183.2 Effect of Variable Lengths of Grouped Records
  - 1183.3 File Maintenance Using the Type 777 TRC
- 1184 File Maintenance with the Type 777--Records Passing Through Memory
  - 1184.1 Maintenance of Files With Fixed Master Item Length
    - 1184.11 Factors Pertaining to 705-777 File Processing
    - 1184.12 Factors Pertaining to Univac II File Processing
    - 1184.13 Comparison of Univac II and IBM 705-777
  - 1184.2 Maintenance of Files With Variable Length Master Items, Fixed Tape Blocks
    - 1184.21 Factors Pertaining to 705-777 Processing
    - 1184.22 Factors Pertaining to Univac II Processing
    - 1184.23 Comparison of Univac II and IBM 705-777
  - 1184.3 File Maintenance With Variable Length Tape Records
    - 1184.31 Factors Pertaining to 705-777 Processing
    - 1184.32 Factors Pertaining to Univac II Processing
    - 1184.33 Comparison of Univac II and IBM 705-777
- 1185 File Maintenance Using "Microsecond Interrogation"
  - 1185.1 File Maintenance Facts Inherent in "Microsecond Interrogation"
  - 1185.2 Level of Activity in File Maintained by "Microsecond Interrogation"
  - 1185.3 Determination of Processing Time Savings
  - 1185.31 Comparison with File Maintenance Using Type 754
  - 1185.4 Possible Duplication of Information in an Interrogated File

INDEX (Page 2)

- 1186 Use of Two or More Type 777 TRCs in the IBM 705
  - 1186.1 Implications of "Double Buffer" Operation Using Two TRCs
  - 1186.2 Factors to be Considered in a Double-Buffer Process
    - 1186.21 TRC Control Sequences
    - 1186.22 Testing of Master File Items
    - 1186.23 Input-Output Areas
    - 1186.24 Number of Tape Units
    - 1186.25 Multiplexing of Processing
    - 1186.26 Machine Runs Before and After a Double-Buffer Process
  - 1186.3 "Microsecond Interrogation" Processing With Two TRCs
  - 1186.4 Double-Buffered "Read Into Memory" Processing
    - 1186.41 Implications of Block Reassembly
    - 1186.42 Testing of Master File Items
    - 1186.43 Control Sequence
    - 1186.44 Productive Processing
  - 1186.5 Mixed "Microsecond Interrogation" and "Read Into Memory"
  
- 1187 Systems Factors Affecting Double-Buffered Processing
  - 1187.1 Tape Units and Number of Input-Output Files
  - 1187.11 Necessary Duplication of Tape Units
  - 1187.12 Other Inputs and Outputs
  - 1187.2 Duplication in Memory
  - 1187.3 Non-Overlapping Operations
    - 1187.31 Machine and Data Transfer Errors
    - 1187.32 Check Points
    - 1187.33 Typewriter Printouts
    - 1187.34 Machine Failure
  - 1187.4 Summary of Double-Buffer Processing Speedup
  - 1187.5 Multiplexing of Two Different Jobs
  
- 1188 Comparison of Univac II With Double-TRC Type 705
  - 1188.1 Costs of Typical EDPM Installations
  - 1188.2 Comparable Installations, Univac II and IBM 705
    - 1188.21 Two TRCs With One Input and One Output
    - 1188.22 Two TRCs with Multiplexing
  - 1188.3 Cost of Performance, Single-Buffer Type Operation
  - 1188.4 Cost of Performance, Double-Buffer Type Operation
    - 1188.41 "Microsecond Interrogation,"
    - 1188.42 "Reading Into Memory"
  - 1188.5 Summary of Comparative Costs
  
- 1189 Sales Points on Type 777 Tape Record Coordinator

1180 IBM TYPE 777 TAPE RECORD COORDINATOR (TAPE CONTROL UNIT)

The Type 777 Tape Record Coordinator (usually referred to as "TRC") is a tape control unit used to connect Type 727 magnetic tape units to the 705 central processor and to provide buffer storage facilities to permit internal computation to proceed during tape read/write operations. The 705 can be equipped with up to ten TRCs, or up to ten total TRCs, 754s, 760s and 774s in any combination. (This statement may be incorrect; the addressing structure is such that ten TRCs in addition to ten other types of tape control units could be used. A definitive answer is of no immediate importance, because no known or projected 705 installation involves anything close to ten such tape control units for on-line operation).

Because IBM is lending heavy sales emphasis on the use of two or more TRCs to provide a "double-buffered" 705, the method of use and the systems implications of the 777 will be examined in considerable detail.

1180.1 Monthly Rental: \$3,000.

1181 CHARACTERISTICS OF THE TYPE 777 TRC

Number of Tape Units Controlled:	1 to 8 (maximum) (Type 727)
Type of Buffer Storage:	Magnetic Core
Character Capacity of Buffer:	1,024 7-bit characters; 1,022 usable positions
Number of Buffers:	1
Transfer Rate, Tape-Buffer:	0.067 ms per character
Transfer Rate, Memory-Buffer:	0.009 ms per character
Tape Unit Addressing Structure:	06X0 to 06X3, 06X5 to 06X8, where: "06" designates TRC use; "X" is 0 to 9 and designates specific TRC (of ten maximum) Unit digit is specific tape unit attached to TRC. 06X4 and 06X9 have special meanings, discussed in 1182
Checking of Information Transfers:	(1) Check bit for each character (2) Check character for each record

The characteristics of the various modes of operation of which the TRC is capable are explained in detail in 1182.

1182 OPERATING CHARACTERISTICS OF THE TYPE 777 TRC

The TRC is capable of five separate modes of operation in a 705 installation; these are designated as A, B, C, D, E. The applicable mode is determined by instruction sequences in a 705 program or by the address of the instruction selecting the TRC. The discussion which follows assumes an understanding of the tape format used in the 705 (1103.6). The explanations all use tape unit addresses in the range 0600 through 0609; they are thus associated with the TRC designated by the ten's digit "0".

1182.1 TRC-TAPE TRANSFERS (OPERATING MODE A)

This mode of operation involves transfers of data between the buffer storage of the TRC and tapes; memory is not involved. The maximum record which can be handled is 1,022 characters. There are three sub-modes of operation.

- 1182.11 Tape to TRC (A1). A grouped tape record can be read from any one of the eight tape units into the buffer; the sequence of instructions (using 0602 as the tape unit from which reading is to be done) is:

SELECT	0602	Selects tape unit
CONTROL (RTS)	0016	Initiates read only operation

The first instruction alerts the proper tape unit; the RTS 0016 initiates the actual transfer. Each requires 0.051 ms of internal computing time; the 705 is free to continue with other computations during the actual transfer from tape to the buffer. The time required for the transfer is  $10/0.067N$  ms, where N is the number of characters in the record read in. The characters of the grouped tape record are read into the buffer beginning at its lowest position and continuing until all characters have been read. Recognition of the inter-record gap on tape causes a group mark to be placed into the next higher buffer position and the check character is placed immediately after the group mark. Necessity for inserting the check character and group mark limits the tape record to a maximum of 1,022 characters.

- 1182.12 TRC to Tape (A2). A grouped record in the TRC buffer can be written on any one of the eight tapes, without reading from another, by this sequence of instructions, again using 0602 as the tape unit:

SELECT	0602
CONTROL (WST)	0017

After executing the WST 0017 instructions, the 705 main frame is free to continue with other operations. Actual writing requires  $10/0.067N$  ms, where N is the number of characters in the buffer record. The writing operation continues character by character until the group mark is reached in the buffer; it is not placed on tape but generates the inter-record gap and terminates the writing operation.

- 1182.13 TRC to Tape and Tape to TRC (Read While Writing) (A3). It is possible to write a grouped record from the TRC buffer onto one tape while simultaneously reading a record from another tape into the buffer. Using 0601 as the read tape and 0602 as the write tape, the sequence of instruction to accomplish this is:

1182.13 (Continued)

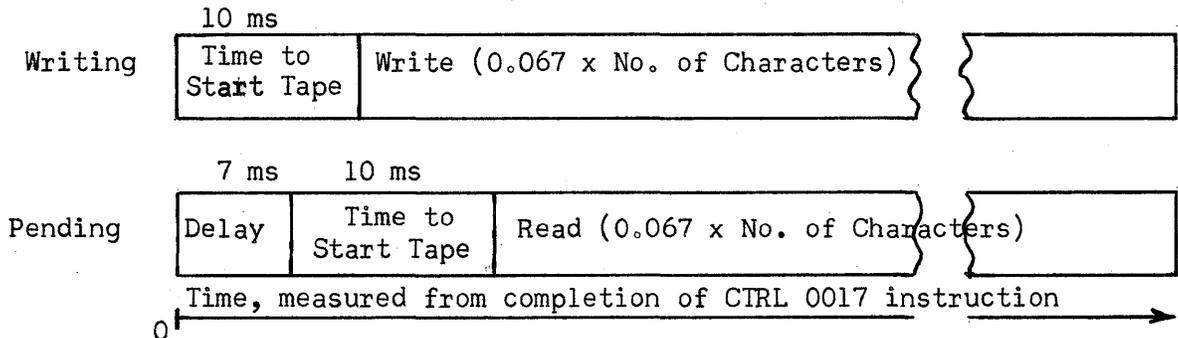
SELECT 0601 Select the tape for reading  
 CONTROL(PRW)0015 Prepare to read while writing (RWW) --

SELECT 0602 Select the tape for writing  
 CONTROL 0017 Write from TRC onto 0602 and read from 0601  
 into TRC buffer

The "read" tape is always the first selected. Once these instructions have been executed (0.204 ms), the 705 central processor is free to continue with other computations while the actual data transfers occur.

This sub-mode of operation is one of the three principal ones which IBM is using to promote the use of "double buffers" on the 705 with two (or more) TRCs.

Because the TRC has only a single buffer, it is necessary to assure that the record being read in does not destroy the information to be written out. This is accomplished by automatically delaying the beginning of the read operation for 7 ms. This chart shows the timing of the two operations:



The 7ms is sufficient to assure that the reading in of information will involve buffer positions already written out; it will be noted that the lag is such that the reading is about 105 character positions after the writing. Thus the writing from and reading into the buffer occur simultaneously, the reading "lagging" the writing by 7 ms, or about 105 characters. If  $N_1$  is the number of characters to be written, and  $N_2$  the number to be read, the total time required for the operation is the greater of

$$10 + 0.067 N_1$$

$$\text{and } 17 + 0.067 N_2$$

RWW, using a TRC, places certain restrictions on the two tape units involved. The maximum of eight which can be associated with a TRC are divided into two groups, depending upon the units digit of the tape address:

<u>Group I</u>	<u>Group II</u>
0600	0602
0601	0603
0605	0607
0606	0608

## 1182.13 (Continued)

In mode A3, one tape must come from each group; i.e., if the read tape is in Group I, the write tape must be in Group II, or vice versa. The restriction applies only to each pair of read-write operations; that is, one set can be Read 0600--Write 0602 and the next Read 0603--Write 0605.

1182.14 Comments on Operating Mode A. Modes A1 and A2 do not appear to be of major importance. Although on occasion they will be convenient and necessary, either operation can often be combined with some other type of data transfer using other TRC modes. The "RWW" sub-mode, however, is always involved in an operation using the so-called "microsecond interrogation" feature of the TRC (see 1182.4), and is fundamental in the use of this feature.

1182.2 MEMORY-TRC-TAPE TRANSFERS (OPERATING MODE B).

This mode involves the transfer of information between the 705 memory and the TRC buffer, together with either a tape reading or writing operation. In each of the three sub-modes, the memory-TRC transfer requires 0.009ms per character (9 ms for 1,000 characters), plus instruction execution time, during which period nothing else can be done in the 705.

1182.21 Memory-to-TRC-to-Tape With Early Tape Start (B1). Because the time to start a tape in motion and accelerate it to full speed is 10 ms, and the time necessary to load 1,022 characters (the maximum possible) from memory into the TRC buffer is 9.2 ms, it is possible to start an output tape in motion at the same time the transfer from memory to the TRC buffer begins. By the time the tape has reached writing speed, the buffer will be fully loaded. The sequence of instructions, which does not overlap other instructions using the TRC is:

SELECT	0603	Select output tape unit 3
Write	(m)	Write record from memory into TRC and from TRC to tape unit 3, with early tape start

As usual, the group mark following the last information character or record mark in memory terminates the memory-to-TRC transfer. Likewise, the group mark in the TRC terminates the tape writing operation.

1182.22 TRC-to-Memory and Tape-to-TRC With Early Tape Start (B2). The converse of B1 is to transfer the contents of the TRC buffer into memory and to read an input record into the buffer, using early tape start for the input. Unlike writing--which can be electronically timed to begin exactly 10 ms after the execution of the "write" instruction--reading must begin when the first character is reached; because of mechanical variations in stopping the tape, it is possible that the first character will be reached before 10 ms have elapsed. With 9.2 ms required to transfer a full buffer load into memory, it is therefore possible to reach the first tape character before the TRC has been emptied. To provide a sufficient margin of safety the execution of the tape movement in this sub-mode recognizes two conditions.

(1) If the record in the TRC buffer is 512 characters or less (maximum transfer time to memory is then 4.6 ms). the acceleration of the

## 1182.22 (Continued)

read tape occurs simultaneously with the beginning of the TRC-to-memory transfer.

- (2) If the record is longer than 512 characters, the tape start is delayed until after the 512th character has been moved into memory; i.e., it is delayed for 4.6 ms.

No program attention is required to achieve the delay if necessary; the "hardware" checks the number of characters in the buffer and inserts the delay automatically. The sequence of instruction in both cases is:

SELECT	0605	Select input tape 5
READ	(m)	Transfer the contents of the TRC buffer into memory, then read tape 5 into the TRC with early tape start.

The TRC buffer contents are transferred into memory, character by character, until the group mark is reached in the TRC. This terminates the transfer, but the group mark itself is not transferred into memory; the last character transferred is the one preceding it in the buffer. The 705 can do nothing else during the TRC-to-memory transfer, at 0.009 ms per character, but is free to continue other computations during the tape reading operation. If N is the total number of characters in the tape record to be read, the total time until the reading has been completed is either  $10 \neq 0.067N$  or  $14.6 \neq 0.067N$  ms, depending upon whether the initial buffer contents exceeded 512 characters.

- 1182.23 Memory-to-TRC-to-Tape and Tape-to-TRC (B3). This sub-mode is quite similar to A3, except that the output record is transferred from memory to the TRC buffer as a part of the instruction sequence, rather than being in the buffer already. The sequence of instructions is similar to that of 1182.13:

	SELECT	0601	Select the input tape, 1
*	PRW	0015	Prepare to read while writing
	SELECT	0602	Select the output tape, 2
	WRITE	(m)	Transfer the record beginning in (m) into the TRC buffer, then write it on tape unit 2, beginning the movement of the tape at the same time the transfer into the TRC commences. Then read one record from tape unit 1 into the TRC buffer, delaying the start of the "read" tape for 7 ms after the start of the "write."
*			
*			
*			

The restriction on tape unit number assignments discussed in 1182.13 applies here also. Otherwise, the method of execution is identical with the individual transfers previously discussed. The 705 central processor is tied up during instruction execution and the transfer of the output record from memory to the TRC at 0.009 ms per character. The tape write operation requires  $10 \neq 0.067 N_1$  ms, and the read  $17 \neq 0.067 N_2$ , where  $N_1$  and  $N_2$  are the number of characters in the output and input records.

## 1182.23 (Continued)

\* Mode B3 is a combination of mode B1 and the tape reading portion of mode  
 \* B2. The dumping of the output record into the TRC and writing it, with  
 \* early tape start, is identical with mode B1. The start of the read tape  
 \* is delayed for an arbitrary 7 ms. for the same reason as the delay in  
 \* the "read while writing" mode A3, section 1182.13. The timing of the com-  
 \* plete cycle is the same as that shown in the chart of 1182.13, the loading  
 \* of the TRC with the output record occurring during the 10 ms start time  
 \* of the output tape.

1182.24 Comments on Operating Mode B. Sub-mode B3 is the most important, because  
 it is the one normally used in file maintenance operations. In execution,  
 it is the TRC operation corresponding most closely to the 5n (write) and  
 3n (read) orders in a Univac. The chief difference is that Univac has  
 separate input and output buffers, and no delay in the read tapes is  
 necessary, as it is in the 705-777. The transfer rate between memory and  
 \* buffer is also considerably faster in Univac II--approximately 2.75 ms  
 for a full block of 720 characters, compared with 6.48 ms for a comparable  
 block in the 705; the main frame is tied up for a considerably longer  
 period in the 705. One factor which will be referred to frequently in  
 subsequent discussions evaluating the use of the TRC should also be noted:  
 \* The group mark is never transmitted from the TRC to memory, but it must  
 \* be at the end of the record to terminate a memory-to-TRC transfer. The  
 \* "early tape start" is likewise nothing new; all it means is that loading  
 \* or dumping of a buffer overlaps some of the tape start time, which Univac  
 \* has been doing since 1951.

1182.3 MEMORY-TRC TRANSFERS (OPERATING MODE C)

In mode C, data are transferred from memory to the TRC buffer, or vice  
 versa, with no tape action involved. This mode of operation is designated  
 by a "4" in the units position of the address selecting the tape unit;  
 it will be recalled that "4" and "9" are not valid tape unit numbers. The  
 form of instruction sequence for either reading (TRC to memory) or writing  
 (memory to TRC) is:

```
SELECT      0604  Select the TRC; the "4" indicates mode C
READ (WRITE) (m)  Read (write) beginning at memory address m
```

The transfer of information is terminated by the group mark in memory  
 (writing) or the TRC (reading). On reading, the group mark is not trans-  
 ferred into memory, but on writing it is placed in the TRC. The time  
 required for the transfer in either direction is 0.009 ms per character,  
 plus instruction execution time, during which period no other computing  
 can be done.

1182.4 "MICROSECOND INTERROGATION" (OPERATING MODE D)

The 777 is equipped with a feature, called "microsecond interrogation,"  
 which permits transferring from the buffer into memory a control field of  
 a few characters, rather than the entire record. By testing an index num-  
 ber in the control field, it can be determined if the record is required  
 for processing, and it is not brought into memory unless it is. The  
 time saved is the 0.009ms per character required to bring in the entire  
 record. The length of the control field transferred can be of 8, 16,

1182.4 (Continued)

32 or 64 characters; a switch on the 777 panel, manually set, determines the number. Operating mode D is indicated by a "9" (not a valid tape unit address) in the units position of the instruction selecting the TRC; e.g., "SELECT 0609" followed by "READ (m)" transfers 8 (16, 32, 64) characters into memory beginning at address m. In use, "microsecond interrogation" is coupled with the "RWW" submode A3 to handle the tapes.

Mode D is being advocated by IBM for the maintenance of files which have a low activity rate. It is the last of three principal modes of operation of the TRC.

1182.5 BY-PASS COORDINATOR (OPERATING MODE E)

In this final mode of operation, the buffer storage of the TRC is by-passed, and the record is read in from tape directly to memory, or the converse. In this mode, its use is identical with that of the Type 754 Tape Control Unit (1140); because nothing else can be done during the transfer of information. mode E normally would be used only for handling records longer than 1,022 characters. Unlike the 754, however, "read while writing" cannot be done in this mode; either one, but not both together. The E mode of operation is indicated by this sequence of instructions:

SELECT	0601	Select tape unit, 1
CONTROL (BPC)	0018	By-pass TRC
READ (WRITE)	(m)	Read (write) record beginning in memory location m

The "CONTROL 0018" instruction is used only for this one purpose, and causes the buffer storage unit to be by-passed only for the succeeding read or write instruction. The TRC reverts to normal status at the completion of the instruction. Although the buffer is not used, the TRC is tied up during the entire transfer of information  $10 \neq 0.067N$  ms, as well as the central processor.

1182.6 CHECKING IN THE TRC

All transfers of information using the TRC, whether from tape-to-buffer or buffer-to-memory, and all combinations thereof, are given the normal IBM validity tests. Each character is inspected for a proper check bit, and a new check character is created and tested against that following the record (TRC-tape transfers only; the check character does not appear in memory). Errors turn on check indicators as follows:

- (1) Errors made in memory-to-TRC or TRC-to memory transfers turn on the read-write check indicator (address 0902).
- (2) Errors made in a Tape-to-TRC transfer turn on the read-write check indicator (0902).
- (3) Errors made in a TRC-to-Tape transfer turn on the record check indicator (0903).

The signal indicating an error in a memory-TRC transfer is available

1182.6 (Continued)

immediately upon completion of the information transfer; the read-write check indicator can be set either to automatic stop or to program control (normally done), and the signal program-tested by a conditional transfer instruction.

The signals in a TRC-tape transfer are not available until the following read or write instruction addressed to the TRC. Again, this is normally under program control rather than automatic stop. Because it is highly desirable to know that a transfer of information has been done correctly, it appears that almost all 705 users will introduce a program convention of a "fictitious" write instruction to determine the validity of a previous read/write operation. This will consist of selecting any tape unit and using a "write" instruction whose address is a group mark location in memory. Because the group mark is not written out, nothing will be written on the tape selected, no tape movement will occur, and the only effect will be to make the error signal, if one exists, available.

It should also be noted that a memory-to-TRC error will frequently turn on the machine check indicator (0901) in addition to the read-write check indicator; the corrective subroutines must provide for turning both of them off.

1182.7 TESTING STATUS OF TRC

The possibility of connecting more than one TRC to the 705 makes it desirable to test the status of any which may be in use; if a TRC is engaged in some previously-initiated operation, it may be that the 705 can be used for some other computation. To enable testing the status of a TRC, the normal "Transfer Signal" instruction of the 705 is divided into two parts when one or more 777s are connected.

- (1) If the address of the "Transfer Signal" is coded for accumulator 00, the normal meaning of the instruction applies.
- (2) If the address is coded for accumulators 01 to 15, it becomes a "Transfer Ready" instruction for the selected TRC, the transfer being effected if the TRC is ready to carry out more instructions; i.e., if it has completed a previous input or output operation.

Through use of the "Transfer Ready" instruction, the status of any TRC may be tested periodically in a program, and some other type of computation performed until it has completed the data transfers specified previously in a program. The use of "Transfer Signal," "Transfer Ready" and "Transfer Any" instructions can be done in any of the modes in which the TRC operates.

1182.8 SUMMARY OF TRC OPERATING TIMES AND MODES

For reference purposes, the operating times and modes of the Type 777 TRC are summarized in the following table. All times are in milliseconds. In the column on TRC times, the letter "N" designates the number of characters to be read or written when only a single tape operation is involved. If both reading and writing are involved, "N<sub>1</sub>" designates the number of characters written and "N<sub>2</sub>" the number read.

Instruc'n	Mode	Unit Selected	CPU Time	Tape Time	Data Flow
CTRL 0015		Prepare	0.051		None
CTRL 0016	A1	Any Tape	0.051	10 / 0.067N	Tape to TRC
CTRL 0017	A2	Any Tape	0.051	10 / 0.067N	TRC to Tape
CTRL 0017 after CTRL 0015	A3	Any Tape	0.051	10 / 0.067N <sub>1</sub> or 17 / 0.067N <sub>2</sub>	TRC to Tape Tape to TRC
WRITE	B1	Any Tape	0.051/0.009N	10 / 0.067N	Mem. to TRC to Tape
READ	B2	Any Tape	0.051/0.009N	10 / 0.067N - 0.009N 14.6/0.067N - 0.009N	Tape to TRC
WRITE after CTRL 0015	B3	Any Tape	0.051/0.009N	10 / 0.067N <sub>1</sub> or 17 / 0.067N <sub>2</sub>	Mem. to TRC to Tape Tape to TRC
READ	C1	06x4	0.051/0.009N	0.051 / 0.009N	TRC to Mem.
WRITE	C2	06x4	0.051/0.009N	0.051 / 0.009N	Mem. to TRC
READ	D	06x9	0.051/0.009N	0.051 / 0.009N	TRC to Mem.
READ preceded by CTRL 0018	E	Any Tape	10/0.067N	10 / 0.067N	Tape to Mem.
WRITE preceded by CTRL 0018	E	Any Tape	10/0.067N	10 / 0.067N	Mem. to Tape

Notes:

Modes A3 and B3: Use longer TRC time.

Mode B2: Use second TRC time if N equals or is greater than 512.

Mode D: N is 8, 16, 32 or 64, depending on switch setting.

Mode E: TRC time and tape time are identical.

1183 LENGTH OF TAPE RECORDS AND TRC OPERATIONS

The three most important operating modes of the Type 777 TRC are A3 and B3 (read while writing) and D (microsecond interrogation). In the typical file maintenance operation, an updated block of the master file is written out at the same time a block from the old file is read in; these are accomplished most rapidly in modes A3 and B3. Periodically, it may be necessary to either read or write a single block--for example, to read a new block of transaction items to be processed against the master file or to write out a block of transaction information. These may not occur at points in the program where a simultaneous read and write is possible. In general, however, Univac experience is that more than 90% of read-write operations occur in pairs.

Because the TRC has only a single buffer, used for both input and output, it is tied up until both operations are completed, in an A3 or B3 mode, or until a single operation is completed in the other modes. In combined reading-writing if the reading takes longer than writing, the TRC is not available for further data transfers until the reading has been completed; the converse is also true. With reading lagging writing by 7 ms in a combined operation, the two types of transfers would end at the same time if the record read in were about 105 characters shorter than that written out. In practice, of course, it is impossible to continue very long with all input records 105 characters shorter than the output. The most efficient use of the TRC, in terms of TRC-tape transfer operations, occurs when both input and output records are the same length. In this case, the combined operation requires  $17 + 0.067N$  ms, where N is the number of characters in each record. This corresponds exactly to the principle of UNIVAC blocks, which are always exactly 720 characters; the only difference is that the length of the tape record in the 705 can be any number of characters up to 1,022 maximum.

1183.1 Effect of Length of Grouped Records of Constant Length. In order to minimize the number of inter-record gaps on a tape--each of which adds 17 ms to total tape time--it is desirable to group individual records into longer blocks, or grouped records in IBM terminology, on tape; this is standard IBM practice in its recommendations. The effect of this grouping on tape speed, in characters per second, is evident in the following table, which gives the net speed for tape grouped records of varying lengths; each allows 0.75 ms for non-overlapping computations necessary to use the TRC (this is about the minimum time which will be possible; frequently it will be more than this).

<u>Number of Characters in Tape Record</u>	<u>Net Tape Speed Characters per Second</u>
100	4,100
200	6,450
300	7,950
500	9,750
750	11,000
1,000	11,800
1,022	11,900

1183.1 (Continued)

It is evident that tape records 750 characters long or longer are desirable to permit passing over unneeded master file information at the highest possible rate of speed. Likewise, it should be noted that the fastest tape speed possible, using the TRC buffer, is only 83% of the rate of Univac II. At best, the 705 with a single TRC takes 20% longer to pass a file than does Univac II. It is equally important to note that the character rates given in the previous table do not include getting the information into memory, but only the time necessary to pass the file through the TRC buffer. The Univac II rate of 14,300 characters per second of course, is based upon the tape data going through memory.

1183.2 Effect of Variable Lengths of Grouped Records. If successive grouped records on a Type 727 magnetic tape are not all of the same length, the determination of tape reading/writing time becomes much more complicated. The following table is indicative of what occurs, and will be used to explore the implications of variable length tape records in more detail. The table is based upon simultaneous reading and writing, using the conventional times for Mode A3 operation; the grouped record read in on one line becomes the record written out on the next.

<u>Number of Characters</u>		<u>Tape Time (Milliseconds)</u>		
<u>Write</u>	<u>Read</u>	<u>Write</u>	<u>Read</u>	<u>Net</u>
(1000)	1000 A	77.0	84.0	84.0
1000A	900 B	77.0	77.3	77.3
900 B	600 C	70.3	57.2	70.3
600 C	850 D	50.2	74.95	74.95
850 D	100 E	67.95	23.7	67.95
100 E	1000 F	16.7	84.0	84.0
1000 F	800 G	77.0	70.6	77.0
800 G	900 H	63.6	77.3	77.3
900 H	1000 I	70.3	84.0	84.0
1000 I	100 J	77.0	23.7	77.0
100 J	( 300)	16.7	37.7	<u>37.7</u>

Total times. . . . . 811.5

The average length of the ten grouped records identified as A through J is 725 characters. If the tape consisted of constant records of this length, the time to write one out and read one in would be 65.6 ms; with the variability shown above the average time is 73.8 ms, or about 12½% longer. The reason for the greater time is obvious by noting that the two short tape records, E and J, have no effect on net TRC time; the longer records on either side override.

Some further considerations arise from more detailed study of the above table. One highly significant statement can be made, and is important enough to justify considerable emphasis: The tape time of a file of variable length items, processed through a TRC, is ALWAYS GREATER THAN the tape time of the file based upon the average of their lengths. In practice, the percentage increase over tape time based on the average is extremely difficult, if not impossible, to determine.

1183.2 (Continued)

The reason that tape time with variable-length grouped records is always greater than with constant is easy to see: It only takes one short or long record to produce the result. The reason that it is so difficult to determine the percentage increase is because it is determined not so much by the variations in tape record lengths as by their distribution as successive grouped records in the file. For example, a file consisting of alternate grouped records of 1,000 and 100 characters -- 1,000 - 100 - 1,000 - 100, etc. -- would require an average of 80.5 ms per read/write operation (84 to write 100 and read 1,000; 77 to write 1,000 and read 100), which is 73% greater than the time to handle constant 550-character grouped records, the average of this file. But if the same file consisted of all 1,000-character grouped records first, followed by all the 100-character grouped records, the total time would be almost the same as a file of constant 550-character tape records. So, even if the number of grouped records of various possible sizes is known, tape time cannot be determined until the distribution of the tape records is known. This almost never will be available and, if it were, would usually be of historical interest only, because the distribution normally could be expected to change every time the file was processed.

About all that can be said is this: The tape time for a file consisting of variable-length tape records usually will be from 10% to 25% greater than the time based on the average tape record size of the file. If the percentage variation between minimum and maximum tape record size is not too greater--say, 800 minimum and 1,000 maximum--then an increase on the order of 10% appears more reasonable; if the variation is wide--say, 200 minimum and 1,000 maximum--then 25% or more will be the expected differential.

1183.3 File Maintenance Using the Type 777 TRC. The comments of 1183.1 and 1183.2, plus the general discussion on magnetic tape file format and maintenance in 8110, are fundamental to an understanding and evaluation of the use of the TRC in business data processing. For convenience, this evaluation will be broken into three separate sections:

- 1184 - File Maintenance With Tape Records Passing Through Memory.
- 1185 - File Maintenance Using "Microsecond Interrogation."
- 1186 - File Maintenance Using Two or more TRCs.

Where applicable, Univac II file maintenance factors are developed for comparative purposes. Section 1186 draws heavily on the analyses included in the other two sections.

1184 FILE MAINTENANCE WITH THE TYPE 777 -- RECORDS PASSING THROUGH MEMORY

One method of using the TRC for file maintenance of processing is to read all tape records into the memory of the 705; the basic master file instructions are Modes B3, for tape reading and writing, and C for dumping the input record into memory. The discussion and evaluation will be considered in the three major types of file format which can be employed.

- 1184.1 -- Constant-length master records assembled into constant-length grouped records on tape.
- 1184.2 -- Variable-length master records assembled into constant-length tape records.
- 1184.3 -- Variable-length master records assembled into variable-length tape records.

1184.1 Maintenance of Files With Fixed Master Item Length. If a file is set up initially with all items of the same length, they can be grouped into the largest number that will not exceed 1,022 characters. Some types of files are susceptible to this sort of treatment; the basic requirement is that all items be of practically the same length at all times; e.g., the variation from the shortest to the longest possible should not be very many characters. Any "padding" with blanks necessary to fill out shorter items to the fixed length results in increased tape time. Preferably, the item length should be such that an integral number of them combined will be close to 1,000 to 1,022 characters; see 1183.2 for the effect on tape speed, in characters per second, of varying numbers of characters in tape records. 50, 100 or 200 characters are good lengths; such lengths as 175, 350 or 525 are less efficient in terms of tape time.

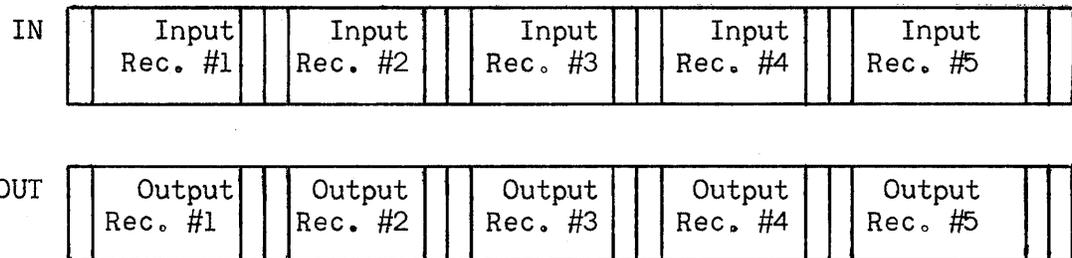
By definition of the master item, any transaction affecting one already in the file cannot change its length; a new item must be a complete master to be set up in the pre-determined length, and a deletion will like wise be a complete item. Consequently, where deletions and additions occur (the normal case in file maintenance), they are relatively easy to handle by reassembly of the input grouped record into a new output record using the high-speed internal memory transfer. Parenthetically, it may be noted that although IBM points out the advantages of variable word and record length, many of the examples it cites show records of a constant length--in other words, they use variable words but constant records in many illustrative examples.

In some respects, the establishment of magnetic tape files with items of constant length is a holdover from punch-card philosophy, where the 80 or 90-column limitations of the card make fixed-length items almost an universal custom. Although many files of business information are inherently variable, some are susceptible to fixed-item length treatment; this is true particularly in the early stages of an EDPM operation and in the pre-ordering days when the potential customer is analyzing his job and evaluating equipments. Consequently, files consisting of fixed-length items are more prevalent today than the nature of data indicates. For this reason, the treatment of this sort of file is worth detailed investigation.

It is self-evident that any file which can be set up as fixed-length items on the 705 can be handled the same way with Univac. (This does not imply that it should be).

1184.11 Factors Pertaining to 705-77 File Processing. The basic (and self-evident) requirement of a fixed-item and fixed tape record file is that every grouped record on tape contain the same number of master items. Because the normal file maintenance processing must provide for the addition and deletion of master items, it follows that the typical input tape record will become part of two output tape records. In the 705, this reassembly is handled fastest by use of the high-speed internal memory transfer; this imposes a restriction on the size of the master item. It must be an integral multiple of 5 characters, including the record mark which terminates the item.

In examining the efficient use of the 705 in processing a file of this nature, it is expedient to assume a given status of the grouped record on both the input and output tapes, and see just what is involved in a program to do the necessary item testing and block reasonably. Therefore assume that records of 200 characters, including the record mark, are being handled; these are grouped into five records on tape. The input area in memory begins in character location 1000; the output area in memory location 3000:



The input records begin in 1000, 1200, 1400, etc. with the record marks terminating each one in 1199, 1399, 1599, etc. A group terminating the input area may be program-inserted into 2000; it is not mandatory but if used for any purpose must be placed there by the program--it is not inserted into memory during a TRC buffer to memory transfer. The output area is similarly set up; in this case the group mark in 4000 is necessary and must be program-inserted at the beginning of the machine run. It does not need to be re-inserted after that.

Now suppose that previous additions and deletions to the file have resulted in the condition that the last three records of one input become the first three of an output. This means that Records #1 and #2 of the next input from tape must be added to the end to fill out the output. Omitting test necessary for determining if any input item must be processed, it remains to consider the use of the high-speed memory transfer in effecting the re-assembly. There are two characteristics of this instruction in the 705 that are pertinent: (1) The data transferred must begin in a memory location ending in a "0" or a "5;" (2) The transfer is terminated by a record mark in the final position of a group of five characters, i.e., in a memory address ending in "4" or "9." The file layout above meets this requirement.

It will be noted that the nature of the internal memory transfer of the 705 permits only one record at a time to be moved. Since each grouped

1184.11 (Continued)

record consists of five individual items or records, five separate memory transfers are required for a complete reassembly of the input block; in this example, two of them move Records #1 and #2 to the end (Records #4 and #5) of an output block, and the remaining three move #3 through #5 to the first of the next output. There are two basic methods of doing this--by "loops" or straight-line coding--and numerous variations of each. In fact, it is somewhat pointless to develop a sample program for this example, because any particular file considered may well have characteristics permitting a faster program to be devised. Therefore, only some general comments on the grouped record reassembly are noted.

- (1) The "loop" method usually takes fewer instructions, overall, than straight-line coding, but generally takes more computer time. It may be the preferred method if memory space is tight.
- (2) Straight-line coding is usually faster overall, but takes more instructions than the "loop" method. Its feasibility arises from the fact that the distribution of an input block into two output blocks can change only when an item is added or deleted; in many files, this may be only once in several hundred records. When it does happen, the memory transfer instructions must be adjusted to meet the new conditions for input block reassembly, but they then remain constant until the next change occurs; this may be many blocks further along in the file.
- (3) The minimum time spent in memory transfers is easily determined. Each transfer takes  $0.051 / 0.0036N$  milliseconds, where N is the number of characters (and a multiple of 5). If X records are combined into one grouped record, the minimum total time for all transfers necessary to reassemble one input block is  $X (0.051 / 0.0036N)$ . In the above example, where  $X = 5$  and  $N = 200$ , this time is 3.775 ms. Note that the minimum can never be achieved in practice, because at some point address modification is required. Using a "loop" to effect the transfers can more than double this time.

One additional point of importance should be noted. At the time an output block is ready to be written, the TRC buffer contains an input record from the previous read/write operation. This must be moved into memory before the output can be written--otherwise it will be destroyed-- and it can't be put into the area already in use because that still contains a portion of the previous input, the last three records in this case. Figure it any way you like, the 705 either needs two input areas and one output, or two output and one input. One or the other choice must be made. One result of this requirement is more instructions--and more computing time--either to modify addresses or to transfer complete grouped records to a common area.

The computing time necessary for the tape record reassembly, plus any time required to make the tests to determine if master file items are needed for processing, almost always can be done while the TRC is in a read-write operation; consequently it often is not important that the re-assembly time is quite variable. Total tape time, however, is capable of

1184.11 (Continued)

relatively accurate determination and in many--if not most--file maintenance operations this is the critical factor. The minimum program for the tape handling follows, with times in milliseconds.

SELECT	0604	0.051	Select applicable TRC
WRITE	2000	0.068	Write group mark; turns on indicators
TRANSFER ANY	X	0.034	Transfer if <u>any</u> error or special condition exists in the 705; this tests accuracy of last TRC-to-Tape and Tape-to-TRC transfers
READ	1000	0.051/ 0.009N	If no error, transfer N characters from TRC to memory, beginning at address 1000
TRANSFER ANY	X	0.034	To test accuracy of above transfer
SELECT	0605	0.051	Select input tape unit
RWW	xxxx	0.034	Prepare to read-while-writing
SELECT	0602	0.051	Select output tape unit
WRITE	3000	0.051/ 0.009N	Mode B3 write and read operation
TRANSFER ANY	Z	0.034	Test accuracy of memory-to-TRC transfer in Mode B3 operation

This is a standard program in IBM examples, and requires a total of 18.459 ms of internal computer time, of which 9.425 do not overlap tape movement. (This time is based on the 1,000-character grouped record). The actual transfer of the output grouped record (next to last instruction) overlaps the start of the output tape, and the final instruction also occurs during tape time. The total non-overlapping time may be summarized as 0.425/0.009N; the total internal computer time is 0.459 / 0.018N.

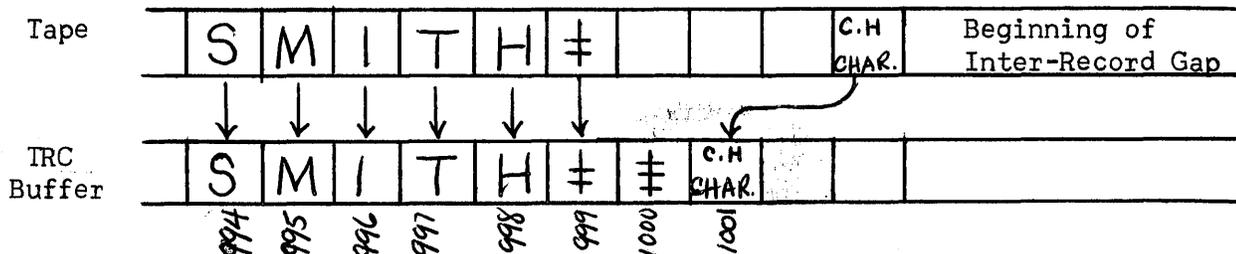
The total 705 time to write a block, read the next input, and do non-overlapping computations may then be summarized, using a 1,000-character tape record as the block size. 17/0.067N for tape time plus 9.425 computing time, a total of 93.425 ms for 1,000-character grouped record. These are IBM times for a grouped record of this size, and they are incorrect. As it frequently does, IBM has oversimplified in two respects: (1) The minimum program shown above probably will not be used in normal practice; and (2) the transfer times themselves are incorrect.

For a discussion of why the program itself is probably oversimplified, refer to the discussion on the limitations and implications of the "TRANSFER ANY" instruction in section 5140. Consider now why the transfer times are incorrect and note one statement: NONE OF THE TIMES IBM IS QUOTING ON 727 TAPE TIMES OR 777-memory ARE ACCURATE.

First take tape time. The standard quotation for time is 10/0.067N, or 17/0.067N ms. The correct time is 10/0.067(N/4) or 17/0.067 (N/4). The reason for the extra four characters is easy to see. Tape read/write operations are not completed until the record has been finished--and the record consists of the data characters followed by three blank spaces and the check character. Every tape record requires 0.268 ms more time to read or write than published figures show. Let's see what actually

1184.11 (Continued)

happens when reading a tape record into the TRC buffer, for simplicity taking only the last few characters of the tape record of 1,000 characters (the last then goes into TRC buffer position 999).



Characters are transferred from tape to the TRC buffer through the record mark in 999; recognition of the three "blanks" on tape causes the group mark to be inserted into position 1,000 of the buffer and the recreated check character into 1,001 after comparing with the character on tape. Because the tape cannot miraculously "jump over" the gap, it evidently requires four character times to complete the checking operation. IBM includes this in none of its tape time estimates--but it is fundamental in checking the validity of the transfers.

Similar reasoning will show that 2 character cycles of 0.009 ms each are omitted from both TRC-to-memory and memory-to-TRC transfers. Through the record mark, everything that happens is fine; this is the "N" IBM uses. One additional character cycle must be added for the transfer of the group mark to the TRC in memory-to-TRC transfers or for recognition of it in TRC-to-memory transfers (remember that the group mark does not get moved in this case). One more cycle must be added for inserting the check character in a memory-to-TRC transfers, or for comparing the recreated one against the original in a TRC-to-memory transfer.

The correct times for IBM tape and TRC-memory transfers are based on these formulas, not on those published by IBM.

Tape:  $10 \neq 0.067 (N \neq 4)$  or  $17 \neq 0.067 (N \neq 4)$  ms  
 TRC:  $0.009 (N \neq 2)$  ms (plus instruction execution time)

Using the corrected formulas, tape time for the transfer of 1,000-character records is 84.268 ms; total time in the example above is 94.711 ms, 9.443 ms not overlapping tape operations. Of the total time, 18.495 ms are required for instructions and TRC-memory transfers, leaving 75.216 available for useful computation during the tape read/write operation. Total tape time, non-overlapping computations and the time available for productive computations vary, of course, with the length of the tape records in the file.

1184.12 Factors Pertaining to Univac II File Processing. In the handling of fixed-length master items in Univac II, the same principles of block re-assembly mentioned as possibilities in the 705 can be used. There is one important difference. Unlike the 705, the size of the master item has no relationship to the output block, which is always 720 characters. Block

1184.12 (Continued)

reassembly can be handled by Y-Z and V-W instruction pairs, changed only as an addition or deletion to the file requires, or by movement of individual items. This is also true of the 705; the choice of method of reassembling input blocks into new outputs often is predicated in large measure on what tests or operations have to be performed on the items as the file passes through the EDPM. In general, "loop" methods of reassembly would not be used unless the item size is an even sub-multiple of 60 words; if it is not, the straight-line block reassembly may be preferable. Univac block time, of course, is a constant 51 ms, and there are no non-overlapping instructions to be performed. As a further note of interest, Univac does not demand the use of two input and one output area, or vice versa, as does the 705; because rI and rO are separate buffers, there is no necessity of unloading an input block before an output can be written. Normally standby blocks are used to permit input-output operations to continue during the occasional processes which may require more than tape time to complete. This does not alter the fact that rI serves in many file maintenance processes as a 60-word extension of memory; the TRC buffer cannot be used for this purpose.

1184.13 Comparison of Univac II and IBM 705-777. Insofar as handling single inputs of fixed tape block and fixed master item sizes is concerned, either the 705 or Univac is relatively straightforward. Internally, such a file is most ideally adapted to the high-speed internal memory transfer of the 705, and it appears that block reassembly can be done in about half-the time of Univac; there will be some variation depending upon the number of records per tape block in the 705, whereas this need have no effect on Univac's internal time. In many file maintenance runs the speed of handling tape is more critical than internal speed, provided that all necessary internal work can be done within tape time. For both equipments, the EDPM housekeeping necessary to test input blocks to determine if they require action, reassemble them into new outputs, and execute tape handling instructions can be done in much less than tape time. This often involves a fairly high percent of the total blocks in the file. Using Univac's standard 51-ms block time and the figures developed in 1184.11 for the 705 time with 1,000-character tape records, the comparative speeds of the two equipments per 1,000 characters are:

Univac:	1,000 characters in 70.8 ms, or 14,300/second
705 :	1,000 characters in 94.7 ms, or 10,550/second

Thus Univac is 33% faster than the 705; using eight tape units on each equipment (the maximum possible with one TRC), the base monthly rental for the Univac is \$22,140; for the 705, \$21,400. In terms of cost, then, Univac does an equivalent amount of work about 30% cheaper than the 705.

Because the 705 is hardly competitive with Univac II, either in time or cost, IBM is pushing the "double buffer" concept, using two or more TRCs, to obtain a better cost-of-performance figure. This will be considered in detail in 1186.

## 1184.2 MAINTENANCE OF FILES WITH VARIABLE LENGTH MASTER ITEMS AND FIXED LENGTH TAPE BLOCKS

1184.1 discussed in detail some of the aspects involved in the maintenance of files with all items of the same length. The more typical business file has items of variable length, and the degree of variation can be very large; files have been encountered in which the maximum size of an item is 50 times greater than the minimum. Because the EDPM has no advance knowledge of how many characters or words are involved in any one item, data must be included within each item which permits determining its length; some of the methods in use are discussed in 8110.

Insofar as this subsection is concerned, it should be noted that the Univac treatment is practically identical with that of 1184.1; it writes out in constant blocks of 720 characters which have no relationship to the item involved. Consequently, most of this discussion is devoted to the 705.

1184.21 Factors Pertaining to 705/777 Processing. In the fixed-item, fixed-tape block method, the processing techniques with the 705 are quite similar to Univac's when the item size is a sub-multiple of 60—say 12 words. The basic difference is that the block size with the 705 is adjusted to an integral number of items, close to 1,022 total characters, and the file then treated as of constant block size size. With different files, the block size can vary, but it remains constant for any one. The length of the tape grouped records is then an integral and unchanging number of items, and the end of an output record always comes in the same character position, which is also the end of an item.

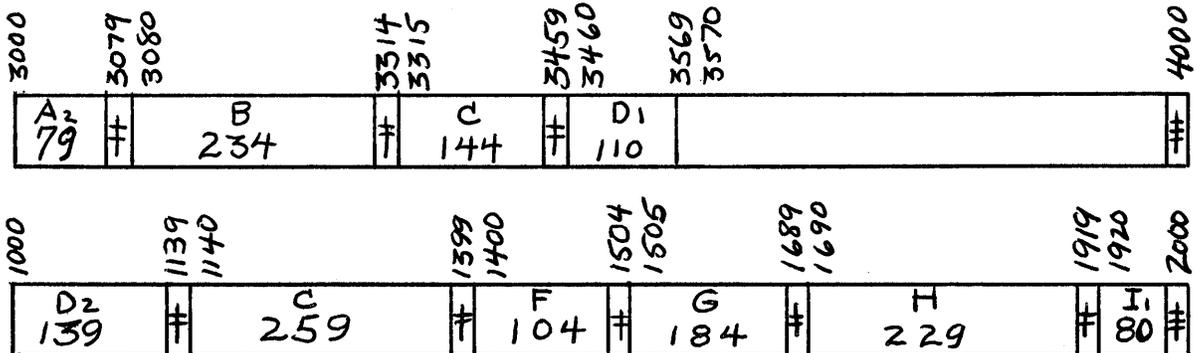
Because constant-length tape records permit most efficient utilization of the TRC in terms of tape handling, let us examine the implications and feasibility of handling variable-length master items in a fixed-length tape block. This means, of course, that a tape block will not, in general, end at the end of an item, but will terminate in the middle someplace. Note that this is identical with the typical Univac situation when variable-length items are being processed. Well, it may be asked, what's the difference in the 705? Why not handle it as we do in Univac? And the difference is this: In the 705, internal memory transfers now become much more complicated because of the way they are accomplished; they are geared directly to a record. Univac doesn't care how many or what records are involved; it transfers a specific number of words.

Before investigating what is now involved in 705 memory transfers, let us note that block reassembly is still required. In fact, in the normal file with variable-length items additions and deletions occur more frequently than in a fixed-item file. Not only are we concerned with complete item additions and deletions, but also with additions and deletions of the variable sub-items within items. In many files, these are of much more frequent occurrence than changes in entire items. This is an inherent characteristic of variable-length items, although the amount of file re-juggling naturally varies greatly in different types of applications.

An understanding of the difficulties now involved in the internal memory

1184.21 (Continued)

is made easier by reference to a typical partially-filled output area and the next input block, some of which is required to fill the output. Both input and output blocks are to be of constant length, by definition, and this is taken as 1,000 characters. The figures between record marks indicate the number of characters in the record; as always, this figure plus one (for the record mark) must be an integral multiple of five. The output block is first.



By previous operations, the output block has been filled with 570 characters, through position 3569, and 430 are needed from the input block to complete it. This requires the first 30 characters of item F, in addition to D<sub>2</sub> (which is the rest of the D items) and all of E. And because F is 105 characters long, including its record mark, it is obvious that the high-speed memory transfer isn't going to stop after 30 characters, but will transfer the next 75 into addresses 4000-4074, removing the group mark in 4000 as it does so. And because the group mark is what stops the writing operation, its elimination is undesirable.

Well, there are numerous methods of handling this undesirable situation. In fact, just as in the discussion of reassembling the fixed item, fixed block, there are so many possibilities that the development of a sample method is somewhat futile; any specific file might be handled in a shorter time. One way is to store the 30th character of item F, insert a record mark in its place and do the high-speed transfer. Upon completion, put the stored character in the last position of the output block. A second method is to set one of the auxiliary storage units to the required number of characters in the last record, and move them one at a time. A third is to establish the output area at a size large enough to hold 1,000 characters plus the longest item in the file; the entire item is transferred, overflowing beyond the output block size, after which the group mark is restored to position 4000 to stop the writing operation. All--and more methods--are possible, and all require computing time.

One statement, however, can be made with certainty, Straight-line coding for the block reassembly can never be done; the loop method, with its address modifications and tests for exit, must always be used. The effective speed of the high speed internal memory transfer then is reduced because of the computing time required for "housekeeping" addresses. How seriously this reduces the effective speed of internal transfers is a function of the average length of items, whether or not the items are moved to a common "working storage" area for testing operations and productive work,

1184.21 (Continued)

the method of indicating the length of the master item, and similar considerations. Internal memory transfers become, in effect, one more highly important consideration to be taken into account with the other complex factors associated with the determination of the most efficient file format for a given processing.

There is one other facet of the reassembly process that deserves mention. The maintenance of a fixed-length tape block in general means that part of the final record in one block is carried over to the beginning of the next. Furthermore, any record in the file can be so split, and usually will over a period of time. Because its length must be an integral number of five characters, including the record mark, in order to use the high-speed internal memory transfer, it follows that the portion of the record at the end of one record cannot be terminated by a record mark; if it did, the combined halves of the record would have two record marks and then not be an integral multiple of five. Needless to say, there are ways out of this predicament, but--like so many other things--they take instructions and computer time. Referring to the illustrations on the previous page, what is required is a means of stopping the transfer of the  $I_1$  partial record (the remainder is in the next input block) when 80 characters have been moved. Perhaps the most obvious way to stop the transfer is to place in addresses 2000-2004 something like four Z's and record mark; they would, of course, be transferred but could easily be written over in the output area as the next input block is reassembled. This costs only one additional memory transfer cycle, but introduces another complication which takes some computer time. Occasionally  $I_1$  will be a complete record and have a record mark in 1999, which will stop the memory transfer. This means additional programming--and time--to determine the exact nature of the last partial record, which may turn out to be complete.

1184.22 Factors Pertaining to Univac II Processing. Because Univac II's input/output operations and high-speed internal memory transfers are both based upon fixed numbers of words, either 60 or specified in instructions, the fact that items happen to be of variable lengths causes no complications in either type of instruction. Just as in the fixed-size item, it is perfectly possible and feasible to adjust the memory transfer instructions only when additions or deletions are made, and use them as straight-line coding until the next change occurs. In other words, the internal memory transfers of Univac are, or at least can be, independent of the information being transferred.

The use of standby blocks, block advancement in input and output areas, and similar well-established techniques in Univac I installations can be carried over bodily into Univac II practice; these methods likewise are independent of the item size, whether it be fixed or variable.

1184.23 Comparison of the Univac II and IBM 705-777. With either equipment, the processing of variable item length files requires programmed advancement, based on data elements within the item, to step from one index number to the next to determine if processing against a transaction is required; this is also necessary if each item must be inspected for a possible action dependent on something in the item (such as an expiration date for a magazine subscription) and not upon a transaction.

1184.23 (Continued)

This requirement cannot be eliminated by repeating the high-order index number at the first of the tape record, because, with block reassembly involved, it would be necessary to redetermine this index number for each new output block--and this would require "bootstrapping" through successive items in the input file. Consequently, both equipments require a loop-type program to advance through successive master items. The 705, in addition to this, requires a loop to do block reassembly. Univac does not.

The net result of this requirement is that internal memory transfers in the 705 become, in general, more time-consuming than they are in Univac. Although the internal transfer of the 705 is, nominally, about twice as fast for an equivalent number of characters, the housekeeping involved in address modifications and limit tests in the 705 may take more time than the transfers themselves.

This leads to a significant conclusion: Unless record being transferred are of constant length, the so-called high-speed internal memory transfer of the IBM 705 is, in general, no faster than that of Univac and in many practical applications is slower. This statement also highlights another fact of general importance. It is fallacious to compare EDPMs on the basis of the speed with which they execute comparable instructions. The important factor is how long it takes to do a specific job.

Insofar as ability to process the file is concerned, the comparative tape speeds of 1184.13 apply here; there is no change in the minimum non-overlapping computer time needed by the 705. However, with variable-length master items, the available computing time in the 705 is less than with fixed-length items; in Univac II it is, or at least can be, the same.

In file operations in which the use of standby blocks is advantageous, the usual condition, Univac II has, in general, a clear-cut advantage. It can always be handled with a Y6-Z6 pair, which requires 5.02 ms. to complete; the 705 record-by-record transfer would necessitate a loop-type program in which address modification and exit tests would normally exceed the actual transfer time.

1184.3 FILE MAINTENANCE WITH VARIABLE-LENGTH TAPE RECORDS

The third possibility of file maintenance with the IBM 705-777 combination is assembling variable-length master items into grouped records consisting of the largest number of complete items which can be contained in 1,022 character position, rather than attempting to construct fixed-length tape records. This possibility obviously does not exist with Univac, which always writes and reads fixed blocks of 720 characters.

- 1184.31 Factors Pertaining to 705-777 Processing. In 1184.21, the ramifications (in part) involved in handling variable-record, fixed-tape block lengths in the 705 were discussed. Some of the difficulties which arose were centered around the fact that, in general an input block ended in the middle of a record, and reassembly of the output block normally required splitting up an input record into two pieces. Although the condition can be handled, both instructions and computer time, not necessary in the Univac system, were required. The question naturally arises as to the possibility of recording information on tape in the form of full records, letting the length of the tape block itself vary.

Initially, consider the internal ramifications now encountered. The first to be noted is that an input grouped record consists of an unknown number of records and characters, each record terminated by a record mark. There is nothing distinctive about the last one; the group mark, it will be recalled, is not transferred from the TRC buffer into memory. Suppose the input area is established as 1,022 characters, the maximum size a grouped record could be. These 1,022 positions always have something in them, because a TRC buffer-to-memory transfer replaces only the characters actually transferred. Further assume that records can vary from 50 to 300 characters in length; this means a grouped record may be 1,022 characters (in rare instances, of course), but also may be as short as 725 (in the instance in which the next record is 300 characters). We can be sure, without special provisions, that a specific item is the last one in a grouped record only if that last item extends (in this case) into position 973, because then we know that even the shortest possible item cannot be included without overflowing the buffer capacity. We can also be sure that if an item ends in position 722 or earlier that another item follows it in the grouped record, because even the longest can be contained within the maximum of 1,022. But if any given item ends between 723 and 972, in this case, we cannot be certain whether or not another follows; it depends completely on the length of the item.

The immediately obvious way out is to say that a test on the index number will indicate whether there is another item following, or just "hash" from some previous, and longer, input record. A moment's reflection will indicate that this may not be so simple as it appears at first sight. It is barely possible--one of those 1 in 10,000 occurrences-- that an index number from some previous input actually occupies those character positions, and any detailed examination of the characters involved would give a positive result. Adding a sequence test would be necessary to be certain, and of course analyzing several characters to see whether or not they had the pattern of the index number would be quite a bit of programming. Transfer high or transfer low tests are certainly not much good; data from some previous input could well give the same indication as an

1184.31 (Continued)

Index number. Inserting a specific character code which never appears any place else in the record is not, of itself, sufficient; it is again possible that some previous index number actually does occupy the place being tested. Testing of the specific character and the index number (for sequence with the current input) would probably be satisfactory. Just as effective, and possibly simpler overall, is a "grouped record terminator" consisting of four Z's and a record mark, for instance, following the record mark of the last information item in the grouped tape record. The important thing is the test must be completely positive, and it requires instructions and computer time to ascertain definitely where a record from tape actually terminates.

The reassembly of the input into two outputs is subject to the same programming complexities of 1184.21; it would require a loop type operation, testing each record to see whether or not it can be contained in the remaining portion of the 1,022-character maximum output block. Finally, once the last item had been placed into the output area, it would be necessary to determine the last character position (i.e., the location of the final record mark), and program-insert a group mark into the next higher address; this is necessary to terminate the output writing.

Everything considered, the housekeeping necessary to use this method appears to be fully as complex as with the fixed-block tape record in lieu of special provisions to terminate the assembly of part of a record into the output we substitute complexities of determining where an input ends.

In terms of tape time, the assembly of information into variable-length tape blocks introduces additional tape time, as compared with that which would be required to handle constant blocks of the average size of a given file. This is discussed in 1183.2, and the remarks made there are applicable in this case. How much additional tape time would be required depends upon the variations in grouped record lengths which are possible, and their actual distribution through the file. Typically, in this case (very short tape records are not likely to occur, but ranges of something like 700 or 800 minimum to 1,022 maximum), the percentage increase will range from 5-15% over that required to handle constant blocks of the average size contained in a specific file.

For many typical files, it appears that the "housekeeping" involved is about the same for fixed-length and variable-length tape records; both require considerable programming and computing time. Because the latter method always increases tape time (see 1183.2), it would be expected that the fixed-block tape record would be preferred. Although this appears to be a perfectly straightforward and logical statement, it is remarkable that no example known to have been published by IBM adopts this approach. As a further note, it may well be that the difficulties of handling variable-length records may be the reason that IBM advocates--and most of its examples are based upon--fixed-length records.

1184.32 Factors Pertaining to Univac Processing. None. Univac cannot operate in this manner.

1184.33 Comparison of Univac II and IBM 705-777. There appears to be little to be added to the comments in 1184.23 on variable-item, fixed tape record files. Under either approach, the characteristics of the internal memory transfer and writing operations on the 705 pose difficulties and require programming not necessary with Univac. Of the two possible approaches with the 705, there seems to be little to gain, and some tape time to lose, with variable-length tape records.

1185 FILE MAINTENANCE USING "MICROSECOND INTERROGATION" (TRC MODE D)

The processing of many business files involves the searching of a large volume of master items to find the relatively few which are subject to action because of some form of transaction input--and order for a stock item, a change of address for a magazine subscriber, etc. Many blocks of information may be passed over before the next master item is found. TRC Mode D, called "microsecond interrogation" by IBM, is a form of operation included to speed up passing over the unneeded information in a master file; refer to 1182.4 for the method of operation of the TRC in this mode.

Because Univac is not capable of operating in this fashion, this section will discuss in the detail necessary the implications of this mode in a 705-777 installation.

- 1185.1 File Maintenance Facts Inherent in "Microsecond Interrogation." The basic fact to be noted, with respect to tape files and operating speeds, is that only those grouped records which contain a master item to be processed are transferred from the TRC buffer into the 705 memory. Those grouped records which do not contain such an item are written directly onto the output tape without going through the memory.

An immediate consequence of this method of passing over the master file is that reassembly of the input blocks into new outputs is impossible. Put another way, all additions and deletions to the file, whether of complete records or portions of them, must be done in the grouped record in which the master item occurs. This is fundamental, and is applicable to master files of both fixed and variable item length.

In practice, what this means is that, as the file goes through several cycles of the maintenance operation, the records become more and more variable in length, and tape time increases, as discussed in 1183.2. For example, suppose the file starts out with all items grouped into 1,000-character tape records. If an item is deleted from one of these records, the remainder may be, say, 900 characters. If an item is to be added, it is inserted where it belongs, and, say, 100 characters from the end of the record are written out as a new and separate tape record. As time goes on, the variability of records on tape in the file increases and so does tape time, until ultimately the entire master file is rewritten to restore the tape records to approximately equal size. The increase in tape time is rapid; for example, if a file of 10,000 grouped records of about 1,000 characters each has added to it 100 items of 100 characters each, the increase in the number of characters in the file is only 0.1%, but the increase in tape time is 1%--ten times as great. And a similar condition holds for deletions; a drop in the file size of 1% may change tape time by less than 0.1%. The effects of combined additions and deletions, which is the normal expectancy, is even more spectacular. The net file size can remain unchanged, but the total tape time can increase by 15% to 25% or more, within a very few cycles of processing. How rapidly the tape time increases is rather evidently a function of the percentage of additions and deletions which occur during the successive cycles of processing.

1185.1 (Continued)

The above discussion permits the establishment of certain criteria which a file should meet to be susceptible of maintenance in the "microsecond interrogation" method:

- (1) The percentage of additions and deletions of information in the file should be extremely low per maintenance cycle--no exact criteria can be established, but well under 1% seems desirable.
- (2) Most likely, the file should consist of fixed-length master items in the sense that necessary information in each item is practically a constant number of characters. Any "padding" necessary to make all items constant simply adds to tape time. The reason for the fixed length item, rather than those which are inherently variable, is that additions and deletions are usually much more likely in a variable-length record than in a fixed one; in the latter, entire items must be added or deleted. In the former, changes within items must also be considered.
- (3) As a corollary of the above, the processing involved in the file maintenance should only change data elements already in the master items, not add or delete information.
- (4) All processing to be done must be initiated by transaction inputs, rather than by information within the master item. The only exception would be those cases where criteria for possible action based on something in the master item could be included in the 8, 16, 32 or 64 characters moved into memory. This is self-evident, since nothing can be done on data in the TRC buffer.

These are basic requirements of the file which must (or at least should) be met before microsecond interrogation can be considered as an economical or practical method of file maintenance. One additional factor--volume of activity--must also be met; this is discussed in the next subsection.

Parenthetically but importantly, it should be noted that very few business files meet all four of the criteria for microsecond interrogation which are listed above.

1185.2 Level of Activity in a File Maintained by "Microsecond Interrogation." IBM is advocating "microsecond interrogation" file maintenance for low activity files, but so far as known has published no definitions of what it means by "low", or at what point some other procedure may be more suitable. These criteria will be developed in this section and an explicit mathematical formula developed to permit ready determination of when a file can be processed more economically by this method than by some other.

To begin, consider how a 705 must maintain a file through successive cycles in this mode of TRC operation. A grouped record from an input tape is read into the TRC buffer at the same time the previous record is written out. The first 8 (16, 32, 64) characters are transferred to memory and tested against the next transaction input item to find out if the corresponding master is in this grouped record. If not, the buffer

## 1185.2 (Continued)

is written out and simultaneously the next input block read in. There is only one mode of operation to do this simultaneous read-write--A3 (see 1182.13).

If a given grouped record in the buffer contains a master item required for processing, the entire group is transferred into memory, the computation performed and the updated grouped record transferred back into the TRC buffer. Then the combined write-read begins. During the entire TRC-to--memory, computing, and memory-to-TRC operations, no tape movement can occur, regardless of how long or how short the computation. Obviously, nothing can be written out, because the last previous input record has already been so handled. Likewise, the next input cannot be read into the buffer. Why? Because it would have to be moved into memory in order to write out the block requiring computation-- and moving blocks into memory is precisely what "microsecond interrogation" seeks to avoid. Therefore whenever a grouped record requires processing, tape movement stops.

We can summarize the information necessary to establish a formula to permit determination of the economy of this method of file maintenance. The time saved by "microsecond interrogation" is the time necessary to transfer a complete grouped record from the TRC to memory and back again for every input block; the time lost is that required for computation, during which period no tape movement can occur. This is the general statement; it will be defined and developed more explicitly in the next few paragraphs.

First of all, the specific formulation desired is one that will enable a determination of which is faster, "microsecond interrogation" methods or one which brings all of every tape record into memory; the latter method is that discussed in section 1184, and involves TRC Mode B3. In both methods, certain instructions do not overlap tape movement, and it is well to see what differences, if any, exist. First, consider a sequence of instructions to handle a file by "microsecond interrogation". It begins at the point where the input grouped record is in the TRC buffer.

SELECT	0609		Set up TRC Mode D
WRITE	mm		Write group mark (to turn on indicators)
TRANSFER ANY	X		Transfer if any error exists
READ	1000	0.051/0.009N	Transfer N characters into memory N = 8, 16, 32 or 64
COMPARE	Y	0.034/0.017M	Compare index numbers, M is length or index
TRANSFER HIGH	Z	0.034	Detail is greater than Master
TRANSFER EQUAL	A	0.034	Detail is equal to Master
SELECT	0602	0.051	Select input tape unit
RWW	xxxx	0.034	Prepare to read-while-writing
SELECT	0601	0.051	Select output tape unit
CONTROL (WST)	0007	0.051	Write-read in Mode A3
Transfer to another program			

This is the minimum program to accomplish the work which cannot overlap tape time. It is noted that the time is variable in two respects: (1)

## 1185.2 (Continued)

The number of characters (8, 16, 32 or 64) transferred from the TRC buffer into memory, and (2) the number of characters in the index being tested, which can be any number up to and including whatever was moved in from the TRC. However, the minimum and maximum times, together with three intermediate "average" times, based upon index numbers of 8, 12 and 24 characters, are summarized below:

<u>Type</u>	<u>N</u>	<u>M</u>	<u>Time (ms)</u>
Minimum (1-character index)	8	1	0.582
Average I (8-char. index)	8	8	0.701
Average II (12-char. index)	16	12	0.841
Average III (24-char. index)	32	24	0.919
Maximum (64-char. index)	64	64	2.157

The range of 0.7-0.9 ms can be considered most typical. Parenthetically, the program on the previous page is taken exactly as presented by IBM in its manual on the Type 777, and has a rather obvious omission, which required one instruction and 0.034 ms more to execute: As presented by them, the accuracy of the transfer from TRC-to-memory of the control word is not tested; the first "READ" instruction should be followed by a "TRANSFER ANY" to do this. So even this minimum program is not sufficient.

The development of a formula to express the sum of tape time and non-overlapping computation is quite simple, and is accurate provided all tape records are of identical lengths. Let "L" be the number of characters in the tape grouped record, "N" be 8, 16, 32 or 64, choosing it to be the smallest that will contain "M", the number of characters in the index number. Recalling that tape time in Mode A3 is  $17 \neq 0.067L$ , and adding up the constant and unchanging times from the program above (including IBM's omitted 0.034 ms), the following formula gives total tape passing time.

$$17.527 \neq 0.067L \neq 0.009N \neq 0.017M \text{ milliseconds.}$$

Typically, this will be 84.7-84.9 ms. The entire file, of course, is not processed at this rate; this covers only the time to pass over tape blocks not required for processing. To total tape time must be added the amount of non-overlapping computation.

This is, of course strictly a function of the processing required and how often it occurs. Before discussing it, a few remarks on "low" activity in an EDPM are in order. Although IBM advocates "microsecond interrogation" for "low" activity files without further defining the word, in many business applications, a file maintenance process in which input transactions run from  $1\frac{1}{2}\%$ - $3\%$  of the number of items in the master file is low. With a transaction density this miniscule, "microsecond interrogation" sounds attractive. Then suddenly it is realized that several master items are grouped together in one tape record--it may be only a couple, but may be eight or ten. And it further becomes apparent that with a transaction density of, say,  $2\%$  and six items per block, something short of  $12\%$  of the blocks will be involved. (It won't be quite 12, of course, because some blocks will have two or three transactions to be posted into them).

## 1185.2 (Continued)

For the further development of the formula for the total processing time, it is necessary to assign symbols for other quantities involved; the following are all involved:

- L = Length (or average length) of the grouped tape records.
- M = Number of characters in the index number
- N = 8, 16, 32 or 64, whichever is the smallest that will contain M.
- T = Total number of Transaction items.
- X = Number of items in the master file.
- Q = Average number of master file items per tape record.
- P = T/X - Transactions as a percentage of the master file.
- K = Average processing time per transaction item processed.

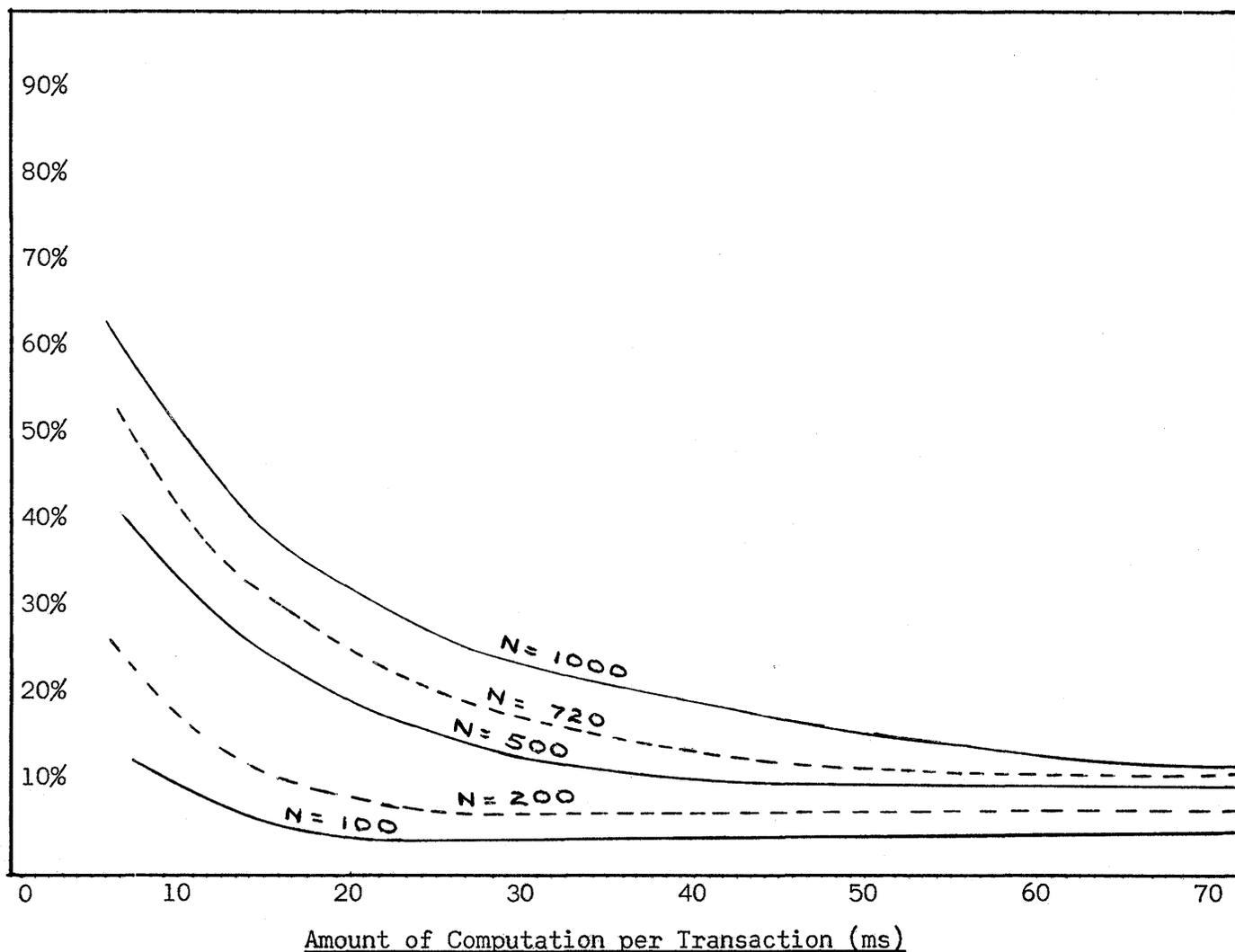
The additional time involved is that necessary to move T input records into the memory from the TRC buffer, plus Computing time. This assumes that each transaction will affect a different tape record; although some tape records will have two or three transactions to be posted, the total number will be quite close to the number of transaction items. This is particularly true when transactions, as a percentage of the master file, are very low; if they number 2% of the size of the main file and, on the average, affect an item every eight or ten tape blocks, there usually will be very few blocks having two or more transactions. The formula for the total processing time then is:

$$(17.527L + 0.067L^2 + 0.009N^2 + 0.017M) X/Q + T(0.051L + 0.009L^2 + K)$$

If not known, T, the number of transaction items, is equivalent to PQ, and both of these are usually known. Actually, knowing P implies that T is known, but people will usually give an estimated percentage of transactions, even though they may not be willing to state a definite average number.

The expression above can be adapted to any given file maintenance process. The amount of actual processing time, K, usually will not be known, but often an EDPM technical consultant can estimate it within 15—20% by a not too time-consuming analysis of the job, and an estimate accurate within this order of magnitude is probably more than is justified; the prospective customer will change his mind many times before the job actually goes onto a computer.

Although use of the formula developed above will always give the time required to maintain a file by "microsecond interrogation," it may be more useful to obtain a general indication of the conditions under which it is potentially applicable. With the 705-777 combination, the alternative method is to read every record into memory, which of course takes more time than merely reading a control word of 8, 16, 32 or 64 characters. But now the computing time, K, in many instances completely overlaps the tape read-write operations. If the percentage of transactions is high enough, the latter course may be faster, overall, than "microsecond interrogation." The three variable factors to be considered are length of individual records, percentage of transactions, and amount of computing time (average) per transaction. In the next section, 1185.3, is developed a graph relating these three variables for various combinations and values and explaining the use of the graph.

1185.3 Determination of Processing Time Savings Using "Microsecond Interrogation"

The chart above gives "break-even" curves which permit a fairly close approximation of whether "microsecond interrogation" is a faster means of file maintenance, using the 705-777, than entering all master file items into memory. Five curves are shown, for tape records (or grouped records) of 100, 200, 500, 720 and 1,000 characters. For a given amount of computation per transaction, "microsecond interrogation" will be faster if the percentage of transactions is below the curve; If the percentage is above the curve, it will be slower than reading everything through memory. Tabular values on which this chart is based are included on the next page.

For both types of processing, lms has been allowed for essential machine housekeeping and tape movement instructions; in practice, this is seldom enough if the file is read into memory. The basic effect is that the amount of computing time shown must include, in this case, that which is required for block reassembly and similar types of operations--anything that will cause the 1 ms allowance to be exceeded. It will be noted that

## 1185.3 (Continued)

each curve becomes a straight line when computing time reaches a certain point; this point occurs when the available computing time for a tape record of a specific size equals the amount of necessary computation. For example, with 100 character records, the available computing time during the tape handling of one pair of records is 21.9 ms; if the actual computation requires more than that, the 705 is computer limited in both cases.

It is important to note that the percentage points are based on blocks, not on transactions. For example, with 1,000-character tape records and 30 ms of computing per transaction, the break-even point between the two methods is 25.1% of the tape records requiring processing. If there are two master items per tape record, this corresponds to a transaction rate of 11.5%; if ten items per tape record, to a rate of 2.3%. The percentage figure used in this chart are equivalent to the transaction rate multiplied by the average number of master items per tape record.

Tabular value for this chart are given below: "K" is the computing time in ms and "%" means percentage of tape grouped records requiring processing.

100		200		500		720		1,000	
K	%	K	%	K	%	K	%	K	%
5	16.0	5	27.0	5	43.0	5	56.6	5	64.4
10	8.7	10	15.6	10	31.0	10	38.8	10	47.5
15	6.0	15	11.0	20	18.5	20	24.2	20	31.2
20	4.5	20	8.5	30	13.2	30	17.6	30	23.1
≤ 21.9	4.3	25	6.9	40	10.4	40	13.8	40	18.4
		≤ 26.8	6.7	≤ 41.5	9.9	50	11.4	50	15.3
						≤ 52.3	10.9	60	13.1
								≤ 66	12.0

In general, the chart and table may be considered as giving the maximum tape record processing percentage and in most practical applications the actual break-even point will be somewhat lower than that indicated. One obvious reason is that a master file processed by "microsecond interrogation" may have duplicated index numbers in the control field, thus adding to the size of tape records and tape time when compared with the method of reading everything through memory. However, both are useful as a simple means of estimating approximate break-even points for the two types of file maintenance processing.

1185.31 Comparison With File Maintenance Using Type 754. IBM has been promoting "microsecond interrogation" for low-activity files, as one means of speeding up overall processing. They have been a good bit more negligent in pointing out that use of a 754 is still faster in this type of operation. As a matter of fact, this statement can be made categorically. The use of a single TRC in "microsecond interrogation" file maintenance is never as fast as a single Type 754 (unbuffered); the latter is ALWAYS about 6ms per block faster. The reason is that the 754 does not impose an automatic tape stop between blocks, but can cover the gap in 10 ms plus essential

## 1185.31 (Continued)

non-overlapping computation, which will average  $1-1\frac{1}{2}$  ms; this is about  $11-11\frac{1}{2}$  ms per tape record compared with 17 ms using the 777. In both cases, all processing is outside tape time, and it is rather evident that the TRC, in a "microsecond interrogation" mode, can never catch up with the unbuffered 754.

An obvious extension is the fact that using the TRC in the mode of reading all items through memory results in break-even curves, when compared with the 754, which have lower percentages than the comparison of the two modes of TRC use. The reason is rather evident: For every tape block, the 754 "gains" about  $5\frac{1}{2}-6$  ms over the TRC, and the accumulated gains can be used to offset computations when these are required. It also gains additional milliseconds which the TRC buffer requires to unload its contents into the 705 memory; with 1,000-character tape records, this is another 9 ms. If a file is established with tape records of this size and the amount of computation per block is under about 16 ms, the 754 will process the file faster than the TRC even if 100% of the blocks are involved. It does not appear that IBM has ever presented its prospects with any comparisons of the relative merits of the unbuffered Type 754 and the buffered TRC. From these few remarks, it is rather evident that the mere fact that a buffer is available does not automatically mean decreased processing time; as a matter of note, many business file updatings can be done faster through use of a 754 than with a TRC.

These comparisons are applicable only when discussing a 705 with one TRC or one 754; as soon as two or more TRCs enter the picture, the 754 loses much of its advantage. This is explored more fully in 1186.

1185.4 Possible Duplication of Information in an Interrogated File. The fact that "microsecond interrogation" permits testing control or index numbers only from the front end of a grouped record on tape means that, in almost all normal applications, a duplication of the high index number from the group at the first of the record. This is rather evident, because in testing a transaction input item against the master file, it is necessary to know which tape record it is in, and this cannot be done simply by looking at the first of a group of items in one tape block. If the high-order index number is duplicated, total tape time is increased by a percentage directly proportional to the length of the duplicated index number as a percentage of the remaining characters in the tape grouped record.

In some instances, this proviso may be unnecessary; for example, if each grouped record consists of the same number of items as all others, and the index number is short enough, it may be possible to put all of them in the first 64 character positions, or less, and not repeat them within the block. This is not quite so desirable if different grouped records have varying numbers of items, because now it is necessary to determine which index is the last in the tape block. In some files, it may be feasible to arrange items within the block in the sequence reversed from the main block-to-block sequence; i.e., to arrange them in descending sequence in the block, although the file itself is handled in ascending sequence. This places the high index number as the first item in the block, and removes the necessity for duplication. Other possibilities may exist in files of a specific nature, but as a general rule it may be stated that some redundancy--adding to tape time--will exist in files processed by "microsecond interrogation".

The second source of duplication of information which may arise is in the area of what may be called, to relate it to conventional punch card practices, "header data." For example, in a name and address file arranged geographically as primary sequence, the most efficient magnetic tape format places the city and state only once at the beginning of the items pertaining to it; it is not necessary to repeat the city and state, or a coded abbreviation, into every master item. This may not be possible in a file processed by "microsecond interrogation," without the inclusion of special programming (for recognizing a change in city) which may add to tape time by placing a short tape record whenever the change occurs, or by additional testing of the control field, which does not overlap tape movement during the testing of the tape record to determine if it is required for processing.

As a general rule, any proposal to use "microsecond interrogation" with the 705-777 implies some redundant tape information, or additional non-overlapping programming, either or both of which add to the total time necessary to process a given master file. The degree of such duplication may vary widely, but it is pertinent to point out to potential users the fact that it almost always will exist.

1186 USE OF TWO OR MORE TYPE 777 TRCs IN THE IBM 705

The previous few sections have discussed in some detail the basic principles of operation of the Type 777 TRC, and factors pertaining to its use in file maintenance operations using both "microsecond interrogation" and the method of reading all information into memory. Using either procedure, the 705 is inferior to Univac II, a major reason being that Univac has an effective tape speed at least 30% faster than the best that can be obtained in a 705 installation. With one TRC, the 705 is simply not competitive with Univac II on a cost-of-performance basis.

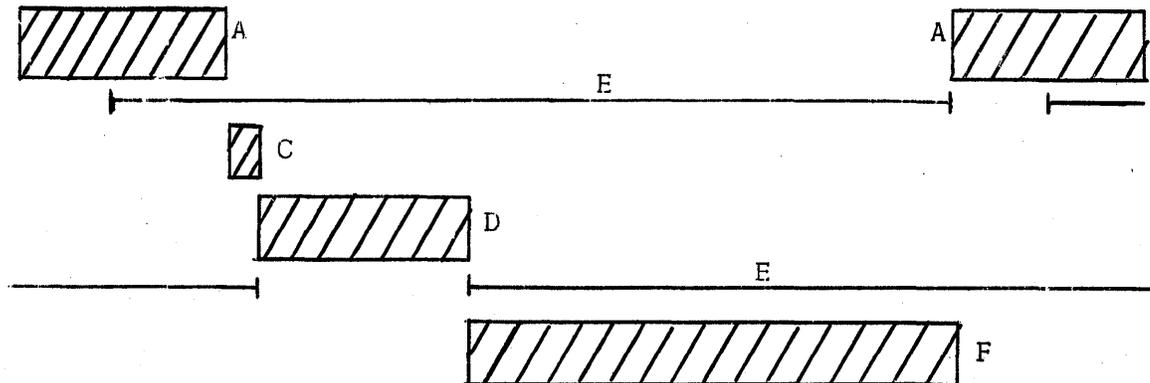
For the past several months, IBM has been strongly advocating the use of two (and in some instances, three) TRCs to permit multi-programming of EDPM operations. Actually, there are several reasons for this sort of proposal, and IBM is avoiding mention of some of them.

- (1) With two TRCs, it is possible to split a file into two halves, and process the two halves in parallel, thereby cutting file maintenance in half. This is, of course, the point IBM has been stressing.
- (2) IBM has advocated multi-plexing of two different jobs in exactly the same manner.
- (3) With one Type 777, the 705 can handle only eight tape units; with one unbuffered Type 754, only ten. With the larger memories of presentday EDPMs, it is often desirable for an installation to have more than this number. This means at least two tape control units for the 705. IBM has not stressed this point.
- (4) The cost of an installation processing a file in sections is greater than that of handling it as one file. This fact is not stressed except by the implication that "twice as much work can be done;" there is no attempt made to demonstrate this point.

This section will be devoted to an analysis of the two-TRC, or "double-buffered" operation of the 705. The development draws heavily on the discussions in 1183-1185, which should be understood before attempting to read this section.

1186.1 Implications of "Double Buffer" Operation Using Two TRCs. In the double-buffer approach, the various methods of handling a file maintenance process with one TRC can be applied to two, doing the job in parallel, or the two TRCs may be operating with a different technique--"microsecond interrogation" in one and reading everything into memory with the other. All of the comments made on the various types of operation with one TRC apply to the use of two, and additional factors must now also be taken into consideration.

Consider, initially, the objective of the double-buffered approach. To permit overlapping the movement of two sets of tapes. Timewise, this may be depicted graphically, assuming that the tape records from both sets of tapes are read into memory from the TRC Buffers.



During time A, the 705 is occupied in testing an input tape record, dumping from the TRC buffer into memory and loading the output record from memory into the TRC; in mode B3, tape advance begins simultaneous with loading the buffer. During time B one record is written on an output tape and one read in to the first TRC buffer. As soon as time A is completed, the 705 checks to see if the second TRC has finished its previous operation; this is time C. If it has, it enters into time D, during which it unloads the second TRC buffer into memory, tests it for action and reloads the buffer, simultaneously starting the output tape into motion. The second TRC writes one record from buffer and reads another in during time E. As soon as time D is completed, the 705 is free to do useful computation until the first (or second) TRC has completed its write-read operation. Thus the two tapes from two different TRC move in parallel, with, of course, a slight time lag between them. This is the basic theory of the operation, and IBMs claims are that the 705 is so fast internally that times A and D are so short that sufficient computing time (E) is normally available to permit processing to overlap the dual tape read-write operations. It now remains to examine the implications of this method of operation.

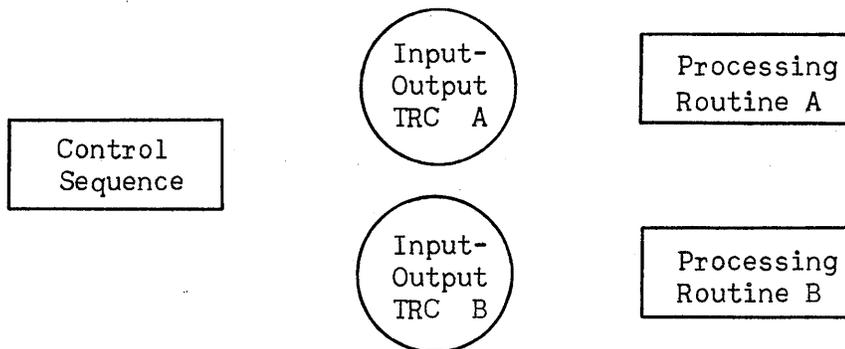
First, there is the matter of control--that is, setting up a routine to switch back and forth between the TRCs. This can be quite short, because it is possible to test the status of the TRC (1182.7). The method is to select one TRC and execute a "Transfer Ready" instruction; if the TRC has completed a read-write operation, transfer to the processing routine. If it hasn't, test the other in the same fashion; if it is not ready, go back to the first. Thus, when no computation is being performed, the 705

1186.1 (Continued)

continues cycling through this control loop until one or the other of the TRCs has completed a previously-initiated read-write operation. In the time-sequence chart above, this loop occupies time C and, using IBM-suggested programs, it lasts about 0.2 ms.

1186.2 Factors to be Considered in A Double-Buffer Process. A double-buffer operation on the 705 is, fundamentally, nothing more than combining two single-TRC processes into one machine run. So far as the work being done through each TRC is concerned, it is subject to all of the considerations discussed in 1184 and 1185, depending upon its mode of operation. In addition, the joint use of the two TRCs introduces new factors caused by their interaction. These are superimposed upon the structure already developed, and this section is devoted to an analysis of these new factors and their effect on the machine operation.

1186.21 TRC Control Sequences. A single TRC program can be thought of as being divided into two logical phases: (1) The tape handling or input-output routine, which may also include master item testing against a transaction, and tape block reassembly, if applicable; and (2) the processing routine, which performs the computations required when a master item is to be processed. In a double-buffer program, these two phases exist for each TRC. One more phase also is added: A "control sequence" to provide the linkage between the two TRCs. These program phases may be depicted graphically:



These deliberately are unconnected. Not only are different linkages possible, but what is done in each of the phases is subject to a wide amount of variation. As an example, the two input-output sequences may be so linked that there really is no "control sequence" as such.

1186.22 Testing of Master File Items. The tape records from each TRC must be examined to determine if processing is required; this may be limited to matching against an input transaction item, or may involve tests on data in every master file item in the file. There may be a separate test sequence for each TRC, or a common sequence for both. In the latter case, a "control sequence" is necessary, in some form, to return the program to the proper place when testing is completed.

1186.23 Input-Output Areas. Depending upon the mode of processing adopted (i.e., "microsecond interrogation" or reading everything through memory), dupli-

## 1186.23 (Continued)

cate input and output areas in memory may be necessary. The amount of duplication could be fairly small--in "microsecond interrogation," for instance--or run into several thousand characters. It also should be noted that such duplication is not limited to the master file, in the case where it is processed in two sections, but may carry over into several subsidiary input and output files.

The import of this duplication of memory areas is at once obvious: Whatever duplication is necessary effectively serves to reduce the size of memory below that which otherwise is available. A direct effect is that less memory is available for instructions.

1186.24 Tape Units. If one long file maintenance run on a master file is performed by splitting it between two TRCs, duplication in tape units must exist. Instead of four for the master file (two input and two output, both sets "ping-ponged"), there are eight--four for each TRC. Instead of one transaction input tape, there are two. Other input and output tapes may also be duplicated. The net effect is either to increase the total number of tape units in an installation, or to reduce the number available for different purposes in a double-buffered machine run.

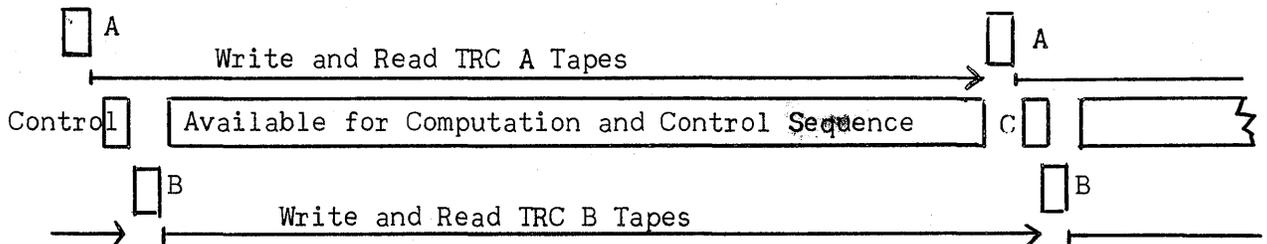
1186.25 Multiplexing of Processing. The use of double-buffer processing may consist of splitting one file into two sections and processing both sections in parallel, doing exactly the same type of work on both halves. Or it may consist of doing two completely different jobs in one machine run, each TRC controlling the tape units for each job, and the two separate programs "time-sharing" the available EDPM computing time.

1186.26 Machine Runs Before and After a Double-Buffer Process. A double-buffer machine run may impose specific requirements on preceding or following runs, or may require creation of runs not otherwise necessary. For example, a tape of daily transactions must not only be sorted into sequence, but must be split into two parts, on separate tape reels, corresponding to the division of the master file; controlling the split may add to sorting time. Similarly, several minor outputs which could be written on separate tape units in a single-buffer type operation may--because of duplicate tape unit requirements, for handling the master file--need to be consolidated on one output tape, and separate for printing (as an example) in a completely new machine run. All EDPM spent in this sort of extra work is, of course, properly chargeable as a cost of a double-buffer operation.

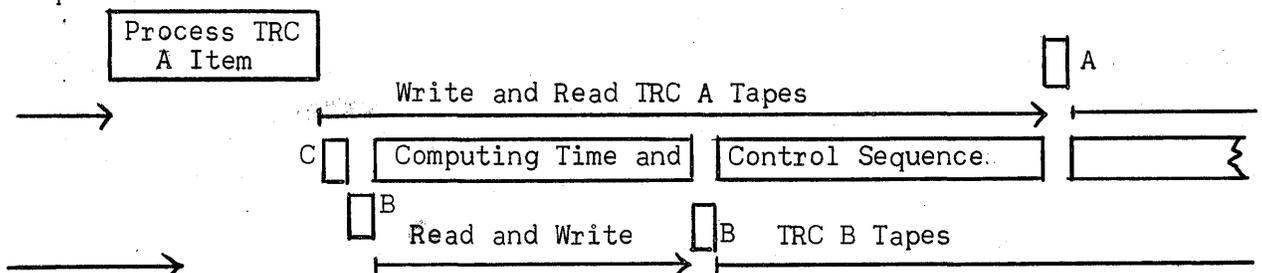
1186.3 "Microsecond Interrogation" Processing With Two TRCs. In this mode of operation (see 1185), the internal 705 time necessary to examine one tape record to determine whether or not it requires processing and to execute the instructions for moving the tapes take relatively little time with average-length index numbers--something between 0.8 - 1.0 ms is enough, using IBM suggested programming. It then appears to be quite attractive to use two TRCs in this mode of operation, because only about 2.0 ms are required for internal tape handling and testing computations out of the 85 ms available in reading and writing tape blocks of 1,000 characters. Such use of two--or even three--TRCs offers an attractive means of increasing the effective number of characters-per-second handled by the 705. Let us now examine more closely this superficial statement.

First of all, the discussion of 1185 pointed out that few business files have characteristics making the "microsecond interrogation" approach particularly appropriate. Even if the tape records are periodically re-grouped into ones of constant length, additions and deletions soon cause them to vary, and tape time per thousand characters goes up. These facts should be understood clearly.

Now the splitting of such a file between two TRCs does not quite double the transfer rate from tape units. One new factor at a minimum is added: The possibility that one TRC becomes ready and must wait, while an item in the other is undergoing processing. Note that it is logically highly impracticable to attempt to interrupt computations simply to keep the first TRC going. There are two reasons for this: (1) The length of tape blocks is variable and there is no foreknowledge of how long it will take to read any specific "next record," and (2) because one set of tapes is stopped during computation, the two sets do not move synchronously and even if all tape records were of a known and constant length, the relative "starting points" of two corresponding blocks changes every time a computation occurs. A time illustration similar to that of 1186.1, may facilitate understanding this phenomenon:



This represents the initial starting condition of the two halves of the master file, with records of identical lengths being handled. The tape handling sequences for the two TRCs are designated by A and B. With no computation required, the time available for computing is spent in cycling through the control sequence until (in this case), TRC A is ready with its next record. Suppose it requires some action; this is the time sequence now:



## 1186.3 (Continued)

Tapes from TRC A, of course, are stopped during the computation period; this is inevitable. But, likewise TRC B is held up, because it is ready with its next tape block before the 705 is ready for it. Now this tape stoppage isn't going to happen every time an item is processed. In the above case, for instance, an item from TRC B could be handled without delaying the tapes from TRC A unless the amount of computation were rather long or the next tape record from TRC A were rather short. However, there is going to be some variation in record lengths, and even more variation in the amount of computing time necessary to handle each item which requires processing; the special sub-routines necessary to handle each type of transaction make that certain. It is thus quite evident that the movement of the two sets of tapes is not going to be synchronous and that some computations are going to stop both TRCs. In other words, there will be some interference between the TRCs. Less obvious, but equally true, the effect of this interference on total processing time is going to be difficult to determine; it depends not only upon the amount of computing, which usually varies from item to item, but also upon the variable lengths of tape records. Some computations will be performed with no loss of time on the other TRC; others will stop it for almost the full processing time, as in the example above. An actual estimate of the time loss is not too important, because if the process is a low activity file maintenance of the type suggested by IBM as suitable for "microsecond interrogation," the interference time is only a small percentage of the tape time.

It may be illustrative to consider a hypothetical example of a 100,000-item master file, grouped five per tape record averaging 750 characters each. Assume 5,000 transaction (a 5% transaction density) are to be processed (involving 25% of the grouped tape records), at 20 ms processing each. Using an 8-digit index number, these times can be established:

Read 10,000 grouped records on one TRC at 67.52 ms per second (this is half the master file) . . . . .	675.2 sec.
Non-overlapping tape-handling instructions and control field interrogations . . . . .	7.0
Processing time, 2500 items at 20 ms each . . . . .	50.0
Non-overlapping TRC-to-memory transfers, 2500 @ 6.75 ms each . . . . .	16.9
10% (assumed) variable record length allowance . . . . .	67.3
Interference factor with other TRC: Assume 33% interference probability on 2500 records, average delay of 16.4 ms (one- half of processing time plus TRC-to-memory and memory-to- TRC transfers) . . . . .	13.7
Total time . . . . .	830.1

The processing of the other half of the master file, from the second TRC, overlaps this time. Thus two grouped records are processed every 83 ms, or one in 41.5 ms; this is an average rate of 1,000 characters every 55.3 ms, as compared with a theoretical maximum of about 42.5 ms. With a single TRC, the time per 1,000 characters is not quite double 55.3 ms; the saving is the interference factor which amounts to only 0.13 ms per

## 1186.3 (Continued)

tape record. It appears, then, that the use of two TRCs nearly doubles the EDPM output at an increase in equipment rental of 26-34% (depending upon the number of tape units associated with each TRC).

Realistically, the percentage improvement should be based upon the performance possible with one unbuffered Type 754 Tape Control Unit, which is always faster than a single TRC operating in the "microsecond interrogation" mode. The file postulated above could be processed in about 70.7 ms average per tape grouped record; now the 41.5 ms possible with two TRCs represents about a 71% increase in production at a 30-40% increase in equipment rental. Additional factors discussed in 1187 will show that even this percentage increase is never attainable in practice; there are many EDPM operations which will halt the entire operation regardless of how many TRCs are involved.

Most 705 installations can be expected to have two TRCs; although one 754 might provide enough tape units with its ten, it is unbuffered and, in general, the additional \$1,000 a month for a TRC is profitable. This probably means two will be ordered to provide for a total of 16 tapes, because very few installations can successfully operate with fewer than ten or twelve. With two TRCs on hand, the discussion so far makes it evident that a double-buffered "microsecond interrogation" process does almost double production compared with one TRC. We shall subsequently consider some of the additional factors (1187) which serve to make the actual expected increase somewhat less.

1186.4 Double-Buffered "Read Into Memory" Processing. If all master file tape records are dumped into the 705 memory, net tape speed is reduced compared with "microsecond interrogation" and more internal computing time is involved in tape handling. As a compensating factor, much of the actual productive processing may now overlap tape time, rather than being outside it. The maximum transfer rate is achieved with constant 1,022-character records, which will be used for evaluation purposes because it gives the most effective 705 performance.

Time for reading one input and writing one output grouped record (1,022 characters each) . . . . .	85.7 ms
Non-overlapping between-record computation (minimum) . . . . .	0.6
TRC-to-memory transfer (non-overlapping) . . . . .	<u>9.2</u>

Total time per tape grouped record . . . . . 95.5 ms

Handling another set of tapes on a second TRC can be accomplished in the same time; this is at a rate of 2,044 characters ever 95.5 ms, or 1,000 characters in 46.8 ms. Of the 95.5 ms, the following are the minimum required for essential operations not related to effective computation:

TRC-to-memory transfers and back (two TRCs at 18.4 ms each) ..	36.8 ms
Non-overlapping between-records computation (minimum) . . . . .	1.4
Data transfers to reassemble inputs into new outputs, not including loop housekeeping (minimum) . . . . .	<u>7.5</u>

Total time per pair of tapes . . . . . 45.7 ms

1186.4 (Continued)

Thus it appears that about 50 ms are available for testing records to determine if they are required for processing and for actual computations. This is usually sufficient to permit unchecked computations, in normal types of processing, to overlap completely the input-output tape operations. Let us now examine the validity of this apparently available 50 ms.

1186.41 Implications of Block Reassembly. The times cited assume blocks of a constant 1,022 characters and will not be significantly less if the actual blocks are close to 1,000. To account for additions and deletions, this means reassembly of an input block into two output blocks as a normal occurrence. Normally, then, an output block is filled with some of the input, and the remainder of the input is carried over to the next following output. Because there is only one dual-purpose buffer in the TRC, the next input block must be unloaded into memory before the output can be written. And, as noted in 1184.11, this means either two input blocks and one output, or two outputs and one input, must be set up. The only alternative is using one TRC only for writing, or only for reading, at a great increase in tape time. The handling of two inputs (or outputs) means two areas in memory, and this requires repetitive address modification shifting back and forth between the two areas. (The solution of transferring inputs to a common work area is possible, but may require excessive memory transfer time in the 705). The computing time necessary to do this address modification may vary considerably. It may be almost none if separate instruction sequences can be used for each area, but this is achieved only at the cost of memory space for non-productive instructions. If the memory space is to be freed for useful instructions, then address modification time is sacrificed. The latter choice, which overall is probably preferable, may add anything from one ms up to several, depending upon other parts of the program which may be affected--as an example, exit points in processing routines may require changing. Something like 2.0 ms, which allows a maximum of about seven changes, is not an extreme expectancy. With a total of two TRCs affected, this adds up to about 4.0 ms as a practical estimate.

There still remains the housekeeping in the individual reassembly program sequences. As noted in 1184.11, this also can be quite variable, but assuming it will take as much time as the actual memory transfers is, if anything, somewhat conservative. This, then, is another 3.75 ms for each TRC or 7.5 ms total.

Thus we have arrived at about 11.5 ms to be added to the block reassembly time, reducing the time available for computing from 49.8 to 38.3 ms. The alternative to doing block reassembly is the approach of "microsecond interrogation;" this immediately adds to tape time because of variable-length tape records.

1186.42 Testing of Master File Items. Unless the high-order index number in each tape block is in a known location, or is repeated at the front of the block, each item must be examined to determine if a transaction applies to it. Such a test requires, at a minimum, two instructions per master item; the time required depends upon the number of characters, but a representative figure is 0.221 ms for an 8-digit index number. If the

## 1186.42 (Continued)

file averages five records per tape block, another 1.1 ms is consumed in direct comparisons and at least an equal amount in address modification (part of the loop testing can be combined with block reassembly). This totals 4.4 ms for two TRCs, with a probable range of around 3.0 ms as a minimum up to more than 10.0 ms for a longer index number and more items per block. And if the user desires an accurate comparison--which may be made mandatory by an outside auditor or a governmental regulatory body--multiply these times by anything from a minimum of two to more than five.

Any significant reduction in this time presupposes that the high order index number in each tape record is in a known location--which means all items are of constant length--or is repeated at the beginning of the block, as in "microsecond interrogation." There is nothing to gain in this latter method in the "reading into memory" mode, because block re-assembly would require its recalculation on each processing cycle.

Using an average of 5.0 ms for unchecked master item testing against a transaction input, the time available for computing is now reduced to 33.3 ms. If the customer is old-fashioned enough to want accuracy in his processing, take off another ten to thirty ms.

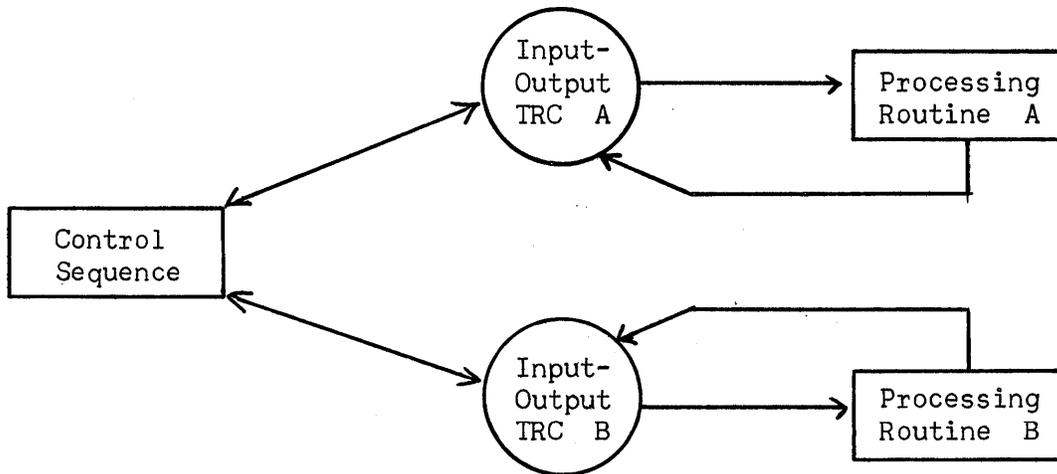
1186.43 Control Sequence. The control sequence and tape handling instructions suggested by IBM are quite short, quite fast, and quite full of holes. For one thing, they are concerned only with reading and writing the master file (so far, this analysis has been limited to that phase). By the time necessary provisions have been made for reading and writing other inputs and outputs, the control sequence may be a good deal more complicated. Likewise, some customers are going to want--or the auditor is going to insist on--safeguards against completely losing a master file block--and using IBM suggested tape-handling sequences makes this possibility a certainty. Instead of using 0.6 ms of non-overlapping computation, it may be anything from 1.0 ms up. A reasonable minimum is on the order of 1.5 ms. This increases the total tape time by 0.9 ms (the difference between the 0.6 ms and the 1.5 ms now used) to 96.4 ms, and reduces the time available to 32.4 ms.

1186.44 Productive Processing. It would appear\* that this 32.4 ms is enough for most normal processing without throwing the 705 into a computer-limited status. This figure is, of course, only a rough average; the amount of time may range from nothing up to possibly 45 ms with 1,022-character tape records, enough for around 225-450 instructions. This seems ample.

Now let us consider the two basic ways of doing necessary actions on a master file being processed in two parallel sections. Each has its own input areas, and, similarly, each has its own input area to hold the next detail or transaction item awaiting processing when the matching master shows up.

The first method is to set up individual programs for each TRC, identical insofar as what is done is concerned, but having addresses tailored to each of the two sets of data storage areas. The functions depicted in 1186.21 are then connected in this manner:

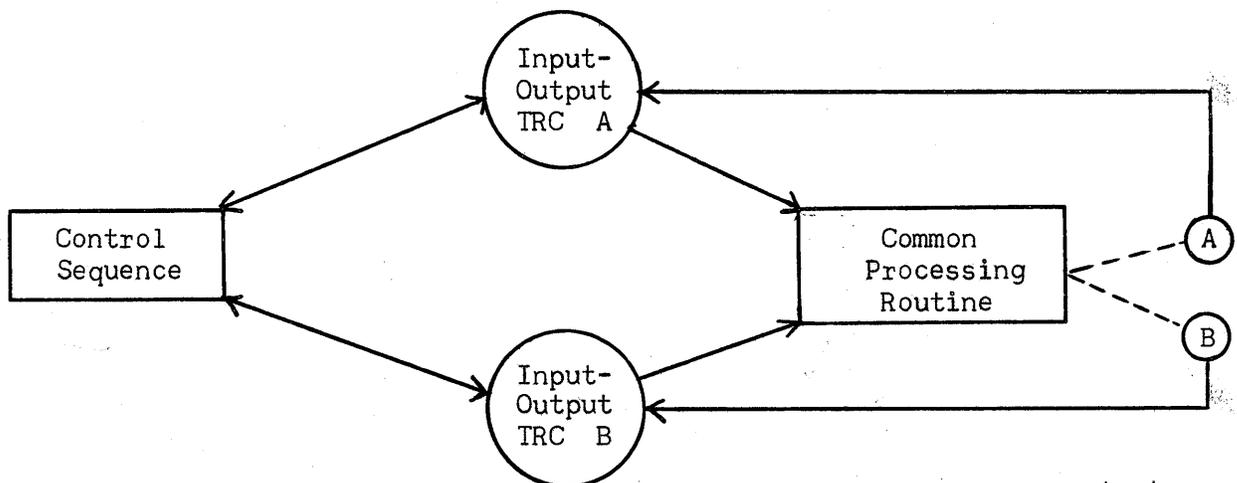
1186.44 (Continued)



This is quite straightforward; as soon as the control sequence finds one TRC ready with its next record, it transfers to the input-output sequence for that TRC. This sequence handles the tape reading and writing, examines the new input to see if any item requires processing, does whatever is required when a "match" occurs, reassembles the input into a new output and then--when this complete cycle is finished--transfers back to the control sequence. There's only one thing wrong with it: Except for the control sequence and program constants, everything in memory is duplicated. Compare with a single-buffer type operation, this approach can accommodate well less than half as many productive instructions. It's well less than half, rather than about half, because both instruction space and input-output space for the second TRC would be available for more useful instructions if only a single set of input-output tapes were being handled.

So, although straightforward, this approach often must be discarded because it seriously restricts the amount of work which can be done in a single pass of the data through the 705. However, it should be noted that this is the precise method followed if two different jobs are being multiplexed on the 705.

The second method is to use a common processing routine for handling items from either TRC; there are still separate input-output sequences for each TRC (probably including the master item tests), but both of these transfers to a common program for any productive computation required. The graphic relationship is this:



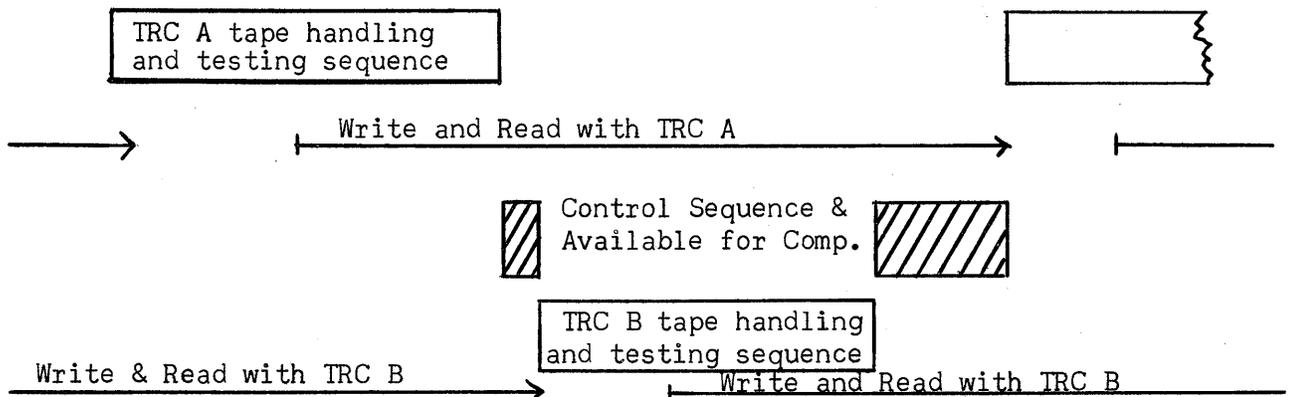
1186.44 (Continued)

It appears that this uses memory more effectively, because now a program twice as long as the first method can be contained. Actually, it is not all gain, because more housekeeping is required. Obviously, the exit from the common routine must be set to the proper input-output sequence before it is entered. This is not enough. Because the inputs can be in two different areas, it is necessary either to move them to a common working storage area or to do considerable address modification within the common routine to adapt it to the particular TRC involved. In most practical cases, the former choice is probably preferable, but this may not always be true. It is, for example, the conventional method used in Univac processing.

In the double-TRC case, it is necessary to move both the master file and transaction input items to the common working area. Although the so-called high-speed memory transfer most likely can be used, it still takes time. Moving a 200-character master and a 40-character transaction item, plus setting the exit from the common routine, take a minimum of 1.15 ms after the transfer addresses have been set. Likewise, the time required to restore the updated master item into its proper place in the new output always is greater than a straight input-to-output transfer.

Thus it is seen that the normal method of using a common processing routine takes some time in housekeeping, but not a damaging percentage of available computing time. Apparently, most processing can still be done within tape time and a double-buffered 705 can process master files at a rate of 1,000 characters about every 47 ms.

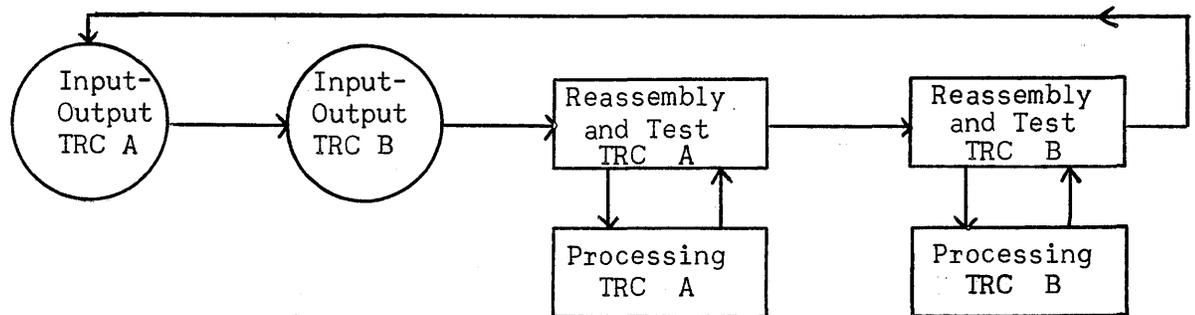
A timing chart becomes useful for further analysis. This is the basic picture:



Assume this is a base condition, with TRC B leading TRC A by a time just sufficient to permit completion of all TRC A tape handling, master file testing and block reassembly by the time TRC B is ready with its next tape record. Now if TRC A has an item requiring processing, the movement of its tapes can continue without interruption, but TRC B becomes "stalled" while the computation is being completed. It then gets out of "synchronism" with TRC A and the tendency is to stay that way. The effect, then, is similar to that encountered in "microsecond interrogation." The two TRCs interfere with each other, and an "interference allowance" must be added to the tape passing time.

1186.44 (Continued)

The question naturally arises: "Can the sequencing of the programming be so arranged that this interference is avoided?" The obvious solution is to do the tape handling in two successive operations, and perform all other operations after tape instructions are out of the way; the graphic chart may be considered in this fashion:



Here the "control sequence" vanishes in a straight-line operation; TRC A is always handled before TRC B. Both new input records are moved into memory, both outputs moved into the TRCs, both tape writing and reading operations are initiated and then the functions of master item testing, any processing required, and block reassembly are done. So long as these latter computations do not extend beyond the period when TRC A is ready with its next record, internal processing can apparently overlap completely the tape movement from both TRCs. This is exactly the objective of the double-buffer process and is precisely what IBM is claiming as possible with two TRCs. Moreover, it gives an effective tape handling rate of 1,000 characters about every 48 ms, definitely faster than Univac's 720 every 51 ms.

Well, what--if anything--is wrong with it? It seems to be quite logical and to cover the normal processing requirements. But it nonetheless has not one, but several, loopholes which we will now take a look at.

Only an engineer would think of the first one--but he'd think of it immediately. The example has postulated tape records of fixed lengths--1,022 characters--as being most favorable to the 705. Two of these are being read simultaneously and the peak performance noted above assumes that reading two records of identical length takes an equal amount of time. Mechanically, this is almost impossible to achieve; it demands absolute uniformity in the speed of tapes during both the writing and reading operations. Even with perfect maintenance, this uniform time of reading cannot be achieved in practical operation. The tape passing time of both TRCs, under the processing method above, is slowed down to the speed of the slower input tape. How much is it? Well, one unit 3% slower than the other adds more than 1.0 ms per 1,000 characters, and a variation from rated speed of this amount is probably more common than a lesser one. Not much, perhaps, but EDPM operations are replete with one millisecond figures--which have a nasty habit of cropping up several places in a program and suddenly adding minutes to a machine run.

The second weakness in this approach is that it requires constant-length

## 1186.44 (Continued)

tape records, which with the 705 almost inevitably means a fixed-length master item. It has already been pointed out (1184.2) that typical business files consist of items inherently variable in length and forcing them into a fixed-length pattern means one (or both) of two things: (1) "padding" with blanks to fill out shorter items to the standard length, or (2) elimination of the variable elements from the file and putting them on a separate tape. The first increases tape-handling time; the second either generates another machine run or moves the bothersome variable elements to another tape unit--where they still must be taken into account and read into the machine in some manner.

Establishing a file in variable-length format immediately rules out the practicability of the processing in this last manner and makes a "control sequence" mandatory for most efficient operation. And--at the same time --it introduces the variable record length allowance (as in "microsecond interrogation") and an interference factor between the two TRCs. The latter factor is of more importance than it was in "microsecond interrogation" for two reasons: (1) There is even less computing time available for productive processing because more is taken up in housekeeping, and (2) the percentage of transactions usually is higher, resulting in more chances for interference. (Even IBM does not recommend "microsecond interrogation" for high-activity files). The variable record length allowance may be somewhat less than in the other mode, because block reassembly can eliminate the extremely short records. But at the same time, the internal housekeeping instructions necessary to use the high-speed memory transfer in block reassembly now take more time. In general, it may be considered that the combined effects of increased reading time because of variable-length tape records and the interference between TRCs will add a minimum of around 15% to the nominal tape time computed above, increasing the 48.2 ms per 1,000 character rate to something on the order of 55-56 ms. This is about the same tape-handling rate possible with "microsecond interrogation," under favorable conditions, but it does have the advantage of permitting more transactions to be processed.

1186.5 Mixed "Microsecond Interrogation" and "Reading Into Memory." It is possible to use one TRC in the "microsecond interrogation" mode and the other in reading into memory. This could occur, for example, if two different jobs were multiplexed and each was best handled in the mode adopted.

However, it may be (in fact, has been) proposed by IBM as a method of maintaining a master file. One example is magazine subscription file in which the name and address is placed on one tape for "microsecond interrogation" processing, and other special information for promotional purposes placed on another tape, the latter being read into memory. The information on this second tape contains everything needed for much normal processing--expiration date of subscription, special information to key promotional literature, etc. The name and address from the other tape are moved into memory only when a change in the mailing list label data is involved. Similar examples can be found in all segments of business; the basic master file characteristic is that some of its data elements must be examined or possibly changed on every maintenance cycle, while others are subject to only infrequent revision. These latter are handled

## 1186.5 (Continued)

through the second TRC and not read into memory--saving the TRC-to-memory transfer time--unless a change is required:

IBM may propose this scheme where part of the information in a master file has a frequent 100% (using a 500 or 1,000 line printer) or high-density printing requirement (using a 774-407 printer in the latter case). The elements printed go on one tape, in completely edited format if necessary, and those not printed on the other. The lack of a printer control panel on the high-speed printers, rather than processing considerations, is the major reason for this approach.

From the standpoint of file maintenance, splitting a master file into two (or more) sections introduces several weaknesses in addition to those already discussed under the two basic modes of operation. Because the two sets of tapes must be processed in complete synchronism, both parts of each item must contain an identifying index number. This duplication adds to the total number of characters in the file and this adds to tape-handling time; it also requires otherwise unnecessary internal comparisons to make sure that corresponding records are being processed.

Both sets of tapes now move record by record, rather than independently. This means that both are limited to the speed of the slower (probably the one with the most information); it is a rare occurrence when both parts of such a split file are of approximately the same number of characters.

A further restriction on tape speed arises when requirements of a printing tape limit the maximum block to 600 characters; this condition arises whenever the print line exceeds 98 characters. A tape block of this size is read about 15% slower than one of around 1,000 characters insofar as master file time is concerned.

The processing rate, in time per 1,000 characters, evidently is subject to considerable variation. If encountered in any sales proposal, an approximate time or rate can be determined quite readily by using the principles and factors discussed in the previous sections. In general, something on the order of 70-75 ms per 1,000 characters is considered to be a rather better-than-average expectation.

1187 SYSTEMS FACTORS AFFECTING DOUBLE-BUFFERED PROCESSING

The discussion of previous sections has been limited largely to factors affecting the processing rate of the master file. Their basic objective has been to explore program and operating facts which derive from the 705's particular logic and to point out practical considerations which must be taken into account in evaluating double-buffered file processing under either major mode of TRC operation. Some mention, but only cursory, has been made of other factors which affect the time and cost of this type processing. These are examined more fully in this section.

- 1187.1 Tape Units and Number of Input-Output Files. The typical EDPM run involves more than merely feeding a master file and a transaction file through the machine to arrive at an updated master. There may be other inputs and practically always other outputs exist. Any proposed EDPM processing scheme must provide for handling all necessary inputs and generating all required outputs.

In a sequential magnetic tape processor, all inputs must be in the sequence of the master file. (The fact that the 705 may have a magnetic drum, which might permit handling small volumes of data not in master file sequences, is not considered in this section. Refer to section 2520 for a discussion of its use). Outputs likewise are generated in this sequence, although sorting or other types of additional work may be necessary to put them into the proper order for a final report or auxiliary tape file record. In general, some outputs are to be used in the master file sequence, while others may require sorting into a new sequence.

- 1187.11 Necessary Duplication of Tape Units. In double-buffered processing of a master file, duplication of tape unit assignment is inevitable. In the typical single-buffer EDPM run--both Univac and 705--four tape units are assigned for handling the master file, two input and two output. This method permits continued reading or writing with one unit while the other is rewinding and being changed. In addition, either one or two tape units are assigned to the transaction file; for simplicity, assume it is short compared with the master file and that one is enough. Five tape units, then, are required for these two files.

In a double-buffered process, five are needed for each TRC, or a total of ten. The master file uses four of them, and a separate transaction tape is required for each half of the master file. (It is easy to show that a single transaction tape would not work).

This is the minimum number which can be assigned. If a proposal involves less, it means that processing from one TRC will be suspended during rewind and tape changing, on both input and output tapes, which usually will not terminate simultaneously (although, in some instances, they can be so set up). Processing of the tape with the other TRC could continue during this period, but would require additional programming in the 705. If either the input or output is limited to one tape unit instead of two, master file handling time is increased a minimum of about 50%--about three minutes rewind and tape changing time are added to the 6.4 minutes required to read an average full reel of tape in a tape-limited operation. If both are limited to one tape unit, master file handling time is about doubled.

1187.12 Other Inputs and Outputs. Tape units are also needed for other inputs and outputs. Inputs must always be assigned in pairs, one for each TRC. Outputs need not be, because a subsequent machine run can separate or sort them as required. Let us consider the two basic types of output and their effect on a double-buffer operation.

1187.121 Outputs in Sequence of Master File. Some outputs should be in the same sequence as the master file; examples are transaction register tapes for printing, exception on error item tapes, and transaction history tapes. In a single-TRC 705 or a Univac II run, such outputs are obtained automatically sequenced during the normal processing. In a double-TRC process, they are available in correct sequence only if two tape units are assigned. This, of course, costs money--and uses up tape units pretty fast.

If a single tape unit is assigned for such outputs, there are two adverse effects. The immediately obvious one is that a separate machine run is required to separate the interspersed items onto two tapes; this of course reduces the ostensible time savings of a double-buffered process. Second, the fact that the entire output is handled through one TRC tends to increase the number of characters it must transfer as compared with the other one, and this again increases the time for master file processing. This can be offset, at least partially, by using each TRC for one such output, but seldom will they be of a nature or volume requiring the same time to handle.

One other facet of the TRC method of operation also becomes evident at this point. Univac has a separate input and output buffer which never interfere with each other. The Type 777 uses one buffer for both purposes; this means that every output instruction must recognize the possibility of an input block in the buffer and provide for dumping it into memory--in its proper place--before the output block is moved. Often this can be highly inconvenient; it is just possible that there is no convenient place available for it. Similarly, master file tape handling orders must recognize the possibility of there being no input block in the TRC at the time a read-write order normally would be executed. The net effect is that tape control program sequences using either one or two TRCs are more complicated than those of Univac and increase tape handling instruction time above the figures arrived at by using IBM's oversimplified sample routines.

The dual use of the TRC buffer is responsible for one other big difference between 705 and Univac II programs. In Univac, input and output instructions are completely independent of each other; after a subsidiary output instruction is underway, the program can shop around at leisure to see if another input can be used, or can read in another master file block. With the TRC, this determination must be made before the output is disposed of. Once it becomes engaged in either a single read or a single write, it is completely tied up ~~until~~ the tape-buffer transfer is finished. There is with the 705 a much greater probability of single-tape operation than with Univac, which can use either buffer without regard to the status of the other.

- 1187.122 Outputs Not in Sequence of Master File. Some outputs, which are of course generated sequentially in the file processing, must be sorted into a different sequence for other purposes. For such outputs, there is no basic loss in placing items from both TRCs onto one tape. As in the previous case, some loss of time most likely will occur because one TRC must handle more characters than the others, and the comments of the last paragraph are applicable here too.
- 1187.2 Duplication in Memory. It has already been noted that the double-buffer approach requires duplication of memory areas for both data and for instructions. This may be relatively small; it could be only a few hundred extra character positions in "microsecond interrogation." In the "reading into memory" mode it involves several thousand, or a significant percentage of the space available. If a master file is set up in tape records of 1,000 characters or so, each TRC requires an input-output area of about 3,000 character positions, or 30% of the total memory--15% for each TRC. More duplication occurs in the transaction input area, although this could be held to single item size at some sacrifice in input tape speed (and probably a considerable sacrifice in the final merge routine that sorted the input items). More duplication occurs in the two sets of tape handling instructions and block reassembly sequences, and still additional requirements are imposed by the control sequence and special instructions necessary to set program "switches" and otherwise provide for conditions not met with in single-buffer processing. At a minimum, a double-buffer 705 operation will cost at least 4,000 character positions in duplicated functions--20% of memory. From 5-7,000 is a more normal expectancy. Obviously, this means that less than usual can be done in one machine run; there simply is no way around the fact that a given area in memory can be used for only one thing at a time--and 5,000 characters of duplicated memory is equivalent to 1,000 instructions. Even if 705 memory is expanded to 40,000 characters (at a monthly cost of \$2,500 more), a double-TRC run means something between 800-1200 fewer productive instructions. (Refer to 2520 for comments on use of a drum for storing instructions which exceed memory capacity).
- 1187.3 Non-Overlapping Operations. A surprising number of mundane operating requirements can stop an EDPM process completely. We have already mentioned some practical factors which will stall one TRC and thus prevent full movement of both sets of tapes. These basically are stoppages which arise because of the logically necessary implications of a double-buffer operation and some of the characteristics of the 705/777. This section points out still more factors which enter into the picture.
- 1187.31 Machine and Data Transfer Errors. Every machine error or tape read-write error causes a temporary machine stoppage or rerun which affects both TRCs. Even failure to read a tape block on the first try--a common occurrence on IBM's Type 727 Tape Units--causes a quarter-second delay with average size blocks--and it would be a beaut of a tape re-read subroutine which would keep the other TRC moving during the correction. Similarly, any restart procedure initiated by a machine error, of any type, consumes considerable time in repositioning the tapes to the proper point. Because IBM tapes cannot be read backwards, they are conventionally rewound and then individually advanced to the proper position. This may take ten or fifteen minutes. Of course, such a point may not be reached for quite a few hours, but also may occur two or three times within a single hour. Restarts may average once every two or three hours.

## 1187.31 (Continued)

to once in ten or twelve, depending upon the specific 705; for some reason, EDPMs seem to have their own personalities and two successive units off the production line can (and usually do) have widely different "quirks."

The slowdown effect of machine errors is extremely difficult to predict, but is definitely significant. For example, ten minutes delay in a ten-hour machine run sounds nominal, but adds almost one millisecond to the time necessary to handle 1,000 characters. And ten minutes is hardly enough to correct one machine error with a programmed restart procedure. And one machine error in ten hours is a rate even less than IBM admits exists; they acknowledge that one completely unreadable tape error occurs on the average every eight hours. The actual incidence of this specific type error is known to be considerably higher in several installations.

1187.32 Check Points. Standard EDPM operating procedure calls for periodic write-outs of memory on a "scratch tape" as a part of program-controlled restart procedures. Although the frequency varies with different installations, two minutes is a recommended interval (oftener than with Univac, by way of note) and one adopted by some 702/705 installations for which the period is known. Let's forget the fact that programming and computing time, usually is some form of block count, are necessary to measure the interval between successive check points. Just count the time required to write out 20,000 characters, backspace the tape and read the write-out to verify its accuracy. About 1.4 seconds for each phase, or 4.2 seconds. Doing this every two minutes adds about 1.7 ms to the per-1,000 character reading rate. Remember--nothing else occurs during this period.

1187.33 Typewriter Printouts. The typical EDPM program provides for considerable chit-chat between the machine and the operator; instructions to change tape reels, announcements of machine or tape error subroutines, etc., keep the operator from getting lonesome. On Univac, most of these are one-word typeouts which don't slow down processing; a one-word buffer permits Univac to overlap computation and typing. The 705 comes to a halt during such printouts, because its typewriter is unbuffered.

This apparently minor difference can have an appreciable effect on tape handling times. For example, in a double-TRC 705 operation, the typeouts to change master file tape reels occur at a nominal rate of four every six to seven minutes. If they took only one second each--shorter than the conventional reel-changing instruction--the four require 1% of total reel time, adding 0.5 ms to the per 1,000 character rate. And this is only one of several types of printouts which may occur on a routine or special basis. Just to add another example, consider the typeout for tape labels of new output reels--at four or five seconds each.

1187.34 Machine Failure. Although we should never mention to the customer the fact that the machine fails, the mere presence of maintenance engineers is a sufficient reminder. It goes without amplification that any maintenance required on a double-buffered operation stops two sets of tape units rather than just one, and has a considerably greater effect on the amount of production per dollar.

1187.4 Summary of Double-Buffer Processing Speedup. The detailed analysis which has been developed makes it readily apparent that a double-TRC operation is never going to result in doubling the processing rate of a file, regardless of how much IBM preaches this magnitude of gain. Of a given machine run, part of the total time represents work which can be split between two TRCs at a definite time improvement, but nothing like a 50% reduction; the remainder is time that is not eliminated or reduced. For example, a 10-hour machine run might involve 7 hours of master file work plus 3 hours of handling subsidiary files, error correction routines and printouts. To process this file on two TRCs might reduce the 7 hours to something like 4 or slightly more; the 3 would remain essentially intact. The net reduction, then, is from ten to seven hours, or 30%, not 50%. In general, a double-TRC operation may be considered as capable of saving from about 25% to 40% of processing time, the actual saving depending upon the cumulative effects of the numerous factors which have been discussed in this and previous sections. This is equivalent to increasing the amount of productive work per hour by from 33% to about 70%, but nothing close to doubling it. When all factors are taken into account, it does not appear that an increase of more than this percentage (70%) can be realized in practice.

1187.5 Multiplexing of Two Different Jobs. Previous sections have paid a good deal of attention to factors affecting the processing of a large master file by multiplexing it through two TRCs. It is also possible to handle two completely different jobs, one on each TRC, and IBM has advocated this type of approach also. (For a rather obvious reason--many installations spend only a small part of their time on files big enough to justify split-TRC processing). Little has been said about the requirements which must be met before this multiplexed processing is practicable. The more important are listed here (not necessarily in order of importance).

1. Both jobs must take about the same machine time when performed singly. This is obvious.
2. Both must be definitely tape-limited when performed singly. If either is computer-limited, or close to it, there is no gain in multiplexing them.
3. The combined number of tape units must not exceed the number in the installation. More specifically, each should be self-contained within the maximum of eight which one TRC can control, although this is not mandatory.
4. The combined program and data storage areas of the two jobs must not exceed the capacity of memory. This is either 20,000 or 40,000 characters.
5. Both jobs should be handled on the same processing cycle. Actually, one could be done singly on a daily cycle and the other multiplexed with it only one day a week; evidently, any savings accrue only when multiplexing is done.

All comments previously made apply in this case. In general, it appears that interference between TRCs will be of more significance, although this may not always be true.

1188 COMPARISON OF UNIVAC II WITH DOUBLE-TRC TYPE 705

Except in cases where IBM proposes a minimum 705 installation to try to beat its cost down to that of a Univac I, it can be expected that most proposals in competition with Univac II will involve two, or even three, TRCs. This is inevitable, because two are necessary to provide buffered control facilities for more than eight tape units and, in addition, a 705 with only one TRC inevitably loses out to Univac II simply on ability to transfer information to and from tapes. The real reason competition will advocate double-buffered operation is simply that it is possible anyway if a workable number of tapes is included in an installation. It is thus pointless to try to convince a prospect that a double-TRC operation is not profitable; if he is going to get a 705, he will definitely profit by multiplexing all jobs he possibly can. Our approach can be much more direct: In jobs not multiplexed, Univac II will out-perform a 705 by a minimum of 35% in speed and 30% in cost-of-performance. If multiplexing is involved, Univac II may be somewhat slower, but will match the 705 in cost-of-performance under optimum conditions for the latter. Because optimum conditions cannot always exist, Univac II will always give across-the-board superior and more economical performance than the 705.

- 1188.1 Costs of Typical EDPM Installations. Because the cost comparisons to be developed involve only the main frame and tape units, nothing else need be taken into account. Monthly rental figures are shown.

<u>Number of Tape Units</u>	Univac II	IBM Type 705 (20,000 Characters)		
		<u>One TRC</u>	<u>Two TRCs</u>	<u>Three TRCs</u>
6	\$ 21,240	\$20,300	\$ 23,300	\$ ---
7	21,690	20,850	23,850	---
8	22,140	21,400	24,400	---
9	22,590	---	24,950	---
10	23,040	---	25,500	28,500
11	23,490	---	26,050	29,050
12	23,940	---	26,600	29,600
13	24,390	---	27,150	30,150
14	24,840	---	27,700	30,700
15	25,290	---	28,250	31,250
16	25,740	---	28,800	31,800

The cost comparisons which are developed assume that both Univac II and IBM 705 will use instructions tape, rather than program cards for the latter equipment. This is the cheaper way for the 705 to operate, because a card reader costs several times as much as a tape unit.

- 1188.2 Comparable Installations, Univac II and IBM 705. Equipment configurations providing essentially the same processing capabilities depend upon how the TRCs are used.

- 1188.21 Two TRCs With One Input and One Output. If one TRC is used only for input and the other only for output (i.e., no multiplexing), the 705/777 requirements for tape units are essentially the same as for Univac. The basic difference is that each TRC can control only eight tape units, which might make difficult holding to using each for only a single purpose.

## 1188.21 (Continued)

In this type of operation, comparative equipment costs can be obtained from the Univac II and second 705 columns in 1188.1. Note that Univac II always costs less than a comparable 705 installation.

- 1188.22 Two TRCs With Multiplexing. When complete multiplexing is involved, a 705 installation will always have more tape units than a comparable Univac II. This is true if the major use of multiplexing is processing a large master file in split sections, or is parallel handling of two different jobs in one machine run. The former is much the more common case, and requires a minimum of five tape units more than a Univac II doing the same job. This is based upon four for the master file and one for the transaction input. Costs of comparable installations then are:

<u>Univac II</u>		<u>IBM 705 With Two TRCs</u>	
<u>Uniservos</u>	<u>Monthly Rental</u>	<u>Tape Units</u>	<u>Monthly Rental</u>
6	\$ 21,240	11	\$ 26,050
7	21,690	12	26,600
8	22,140	13	27,150
9	22,590	14	27,700
10	23,040	15	28,250
11	23,490	16	28,800

An IBM Type 705 installation costs roughly 25% more than a comparable Univac II.

- 1188.3 Cost of Performance, Single Buffer Type Operation. If the 705 does not use each TRC for both reading and writing, the relative processing rates of the two equipments are directly comparable. In a tape-limited operation, Univac II will pass a master file at a rate of 720 characters every 51 ms, equivalent to 70.8 ms per 1,000 characters. The 705 will pass a 1,000-character record about every 82.2 ms, reading in mode B2. The relative cost of performance for varying numbers of tape units may then be calculated, using the Univac II cost in each case as 100:

<u>Number of Tape Units</u>	<u>Monthly Rentals</u>		<u>705 Cost of Performance</u>
	<u>Univac II</u>	<u>IBM 705</u>	
8	\$ 22,140	\$24,400	127.9
9	22,590	24,950	128.2
10	23,040	25,500	128.4
11	23,490	26,050	128.6
12	23,940	26,600	128.9
13	24,390	27,150	129.1
14	24,840	27,700	129.3
15	25,290	28,250	129.6
16	25,740	28,800	129.8

The IBM 705, then, costs 28-30% more than Univac II per unit of production. Further, this is the minimum differential favoring Univac; it applies when the 705 is processing master files records of constant length, thus permitting complete parallel operation of the tapes. In the more typical case of variable length items--and normally variable length tape records--the 705 time per 1,000 characters goes up, resulting in a still higher cost-of-performance ratio.

## 1188.3 (Continued)

These figures simply establish factually what is obvious from Univac's faster input-output rate: The IBM Type 705 is never competitive when reading and writing single tapes. The sole possible exception is a highly computer-limited operation, handling factors of very short lengths, and in which accuracy is either not important or can be verified in other ways; in this relatively uncommon case (at least in business data processing), the ability of the 705 to operate on short words faster than Univac II may give it a better cost-of-performance ratio.

1188.4 Cost of Performance, Double-Buffer Type Operations. In a double-TRC process using each for both reading and writing, the comparative cost-of-performance depends upon the actual speed of operation attained. Previous sections have discussed in considerable detail the numerous aspects to be taken into account. Cost comparisons can be summarized by postulating 8 Uniservos with a Univac II and 13 tape units with a 705; this is not to be construed as meaning that an 8-Uniservo installation is a good Univac II configuration, but is used simply because one TRC controls that number. Any upward change to a maximum of 11 Uniservos or 16 Type 727 tape units will cause only a small change in the cost indices developed below.

1188.41 "Microsecond Interrogation." The theoretically-maximum transfer rate is 1,000 characters every 42.5 ms; this of course allows no time for processing and assumes that all tape records remain a constant 1,000-plus characters. The more typical maximum expectancy, as discussed in 1186.2, is at a rate of about 1,000 characters every 55-56 ms. When some of the non-overlapping operations discussed in 1187 are considered, a rate of 1,000 characters every 60 ms appears to be an optimistic production hope. Univac II does not operate in this mode, but handles characters at the rate of 1,000 every 70.8 ms. The relative costs of performance may be calculated for several time rates of the 705, again using Univac II as 100.

<u>705 Time per 1,000 Characters</u>	<u>Relative Cost of Performance</u>
42.5 ms	73.6
50.0	86.6
55.0	95.3
56.0	97.0
60.0	104.0
65.0	112.7
70.0	121.2

Thus it will be observed that under the optimum normal conditions, the 705 using two TRCs in "microsecond interrogation" is barely better than Univac II in cost of performance. In the more normal condition, with 60-65 ms rates per thousand characters, it is somewhat more, the differential becoming appreciable for a 65 ms rate. In practice, the difference is even more marked because of the introduction of other factors--such as "padding" of tape records--which virtually negate the applicability of this mode of operation.

1188.42 "Reading Into Memory." The maximum 705 character transfer rate with two TRCs reading into memory is 1,000 every 48 ms, with a normal maximum of 1,000 every 55-56 ms. Like "microsecond interrogation," this is probably a better rate than can be achieved in practice, and something between 60-65 ms is considered the optimum working rate which can be expected. Because the tape unit complement is identical with the case previous, the relative cost-of-performance figures are also the same. Thus, under the best possible conditions, the 705 will have a slightly better cost-of-performance ratio than Univac II; under the typical working conditions, it will range from slightly to definitely more expensive.

1188.5 Summary of Comparative Costs. The determination of Univac II time for a given operation is much simpler than a double-buffered 705; in our case, it is not necessary to take into account such definite--but indeterminable--factors as possible interference between the two sets of tapes and variations in tape record lengths. IBM can be expected to propose methods most favorable to double-buffered processing; with the basic logic of the 705, this almost inevitably means an emphasis on fixed-length master items (a punch-card approach that has seldom been pronounced in Univac applications), and this, in turn, entails at least some padding of records. The discussion of previous sections has gone into considerable detail on the ramifications in the two basic modes of operations of a two-TRC 705, with the significant points in both file format, tape handling and normal operation developed. The net conclusion is that under favorable circumstances, the two TRC 705 is approximately cost-competitive with Univac II and in more normal circumstances is definitely more expensive.

It must be remembered that not all operations are susceptible to this type of treatment. IBM itself does not make this claim. And when the two TRCs cannot be used for reading and writing in each of them, Univac is always faster and cheaper. And accurate. It takes very little time per 8-hour shift before any savings made on a two-TRC operation vanish. For example, if the 705 can do double-buffered work at a rate of 1,000 characters every 50 ms--an exceptional rate--on some files and only single-buffer type operations the rest of the time, it would require 5½ hours per 8-hour shift of double-buffered operation before the 705 became, overall, competitive with Univac II. In the normal optimum expectancy, with a cost-of-performance margin of 3-5% over Univac II as maximum, a half-hour a day of other work would more than offset any cost savings which might accrue from the double-buffered period. And in almost any imaginable application, more than that time would be spent in single-type work occurring during a double-buffered machine run.

The net conclusion is that a double-buffered 705 is rarely cost-competitive with Univac II; the margin will range from a minimum of 10% up to more than 25% for the Univac. And Univac's accuracy is a built-in and free bonus to this better cost-of-performance ratio.

1189 SALES POINTS ON TYPE 777 TAPE RECORD COORDINATOR

Many of the points included in this section refer to the detailed discussions of 1180-1188. A complete evaluation of the Type 777 TRC can become extremely complex; however, all of the essential factors have been explained and analyzed in detail and should be referred to as necessary in combatting IBM claims for the performance of this unit.

- 1189.01 "The 777 Doubles the Speed of the Computer." IBM has been advertising the TRC as doubling the speed of an EDPM by permitting simultaneous read-write and also overlapping computation.

For the 777: True, with some restrictions.

Against the 777: Well, it depends upon the reference point. Insofar as IBM is concerned, the statement is true. Of course, Univac I had the facility in 1951, still has it, and probably always will. Simultaneous read-write-compute is not new; it was in the first EDPM ever built for business data processing--Univac I. Well, it took IBM only  $5\frac{1}{2}$  years to almost catch up.

- 1189.02 Speed-up Possible With Double-TRC Processing. IBM states, in the introductory paragraphs to the Type 777: "If the file is split in half, the over-all speed of processing should be approximately double."

Against the 777: Never true. Consider these factors, among others:

- (1) Effect of variable-length tape records (1183.2).
- (2) Padding records to constant length, in either "microsecond interrogation" or reading into memory. (1184.1, 1185.4).
- (3) TRC interference and non-overlapping computation (1186.2, 1186.3).
- (4) All factors in 1187.

- 1189.03 Use of "Microsecond Interrogation." IBM may propose maintenance of files in this mode as a time-saver.

For the 777: The mode is possible.

Against the 777: These factors work against this mode of operation:

- (1) With only a single TRC, use of the unbuffered Type 754 Tape Control Unit is always faster and cheaper.
- (2) With two TRCs, and for basic implications of this method of processing, refer to 1185, 1186.3, 1187.

- 1189.04 Cost Comparisons. See 1188. Note that a single-TRC 705 is never competitive with Univac II and a double-TRC installation always costs more.

- 1189.05 Detailed Discussions of TRC Operating Implications. Refer to the index preceding Section 1180 for the sections discussing the various types of operations and their processing implications.

1400 DATAMATIC MODEL 1100 MAGNETIC FILE (TAPE) UNIT

The Model 1100 is the standard tape unit for use with the DATAmatic 1000 data processing system, both for on-line and off-line operations.

1400.1 Cost: Purchase: \$60,000  
Rental: 1,350 monthly.

1401 TAPE UNIT AND MAGNETIC TAPE CHARACTERISTICSMagnetic Tape and Reel:

Tape base:	Mylar; ferric oxide magnetic surface is laminated between two sheets of mylar
Tape width:	3"
Tape length:	675, 1350 or 2700 feet
Tape reel diameter:	14"
Tape cost:	675 feet -- \$230, including two reels
	1350 feet -- 400, " " "
	2700 feet -- 775 " " "

Model 1100 Magnetic File Unit:

Number of recording channels:	36: 5 control channels 31 information channels
Character recording density:	Approximately 170-175 bits/inch. (1200 numeric or 800 alphanumeric characters per linear inch of tape)
Tape speed (forward & rewind):	100 inches/second
Character Transfer rate:	
Instantaneous:	120,000 digits or 80,000 characters/second
Maximum Effective:	60,000 digits or 40,000 characters/second
Tape start time:	6 ms
Tape stop time:	4 ms (maximum)
Tape reversal time:	Not known, but not instantaneous
Bad spot detection:	Permanent recording in control channel
Detecting physical ends of tape:	Permanent recording in control channel
Method of Recording	Frequency modulated signals
Mode of recording:	31 channels in parallel
Length of tape record (block):	Fixed; 62 words, 52 bits each. Each of 31 channels contains two words
Check bits:	4 bits of each word used for checking
Length of inter-block gap:	Approximately .625 inches
Recording in gap:	Yes; gaps are filled by recording in backward movement of tape
Tape capacity (2700-foot)	37,200,000 decimal digits 24,800,000 alphanumeric characters 3,100,000 DATAmatic words 50,000 DATAmatic blocks

1402 CONTROL UNITS AND USES OF THE MODEL 1100

The various configurations of control units with which the Model 1100 may be used are listed below, for both on-line and off-line operations. Refer to write-ups on the various control units (converters) for more detailed information.

Central Processor Operations:

Model 1000 Central Processor	Maximum of 100 tape units
Model 1000 basic unit has controls for	8 tape units
Model 1052 Auxiliary Central Processor has controls for (additional to basic)	2 tape units
Model 1054, as above, for (additional to basic)	4 tape units
Model 1056, as above, for (additional to basic)	6 tape units
Model 1058, as above, for (additional to basic)	8 tape units

(Models 1052, 1054, 1056 and 1058 apparently may be combined to provide the control units needed for the maximum installation of 100 tape units).

Periphery Operations:

Card-to-Tape Conversion:	Card Reader - 1200 - 1100
Tape-to-Card Conversion:	1100 - 1300 - Card Punch
Tape-to-Printer:	1100 - 1300 - Printer
Tape-to-Printer (high-speed):	1100 - 1400 - High-Speed Printer

1403 MAGNETIC TAPE FORMAT WITH THE MODEL 1100

The magnetic tape used with the DATAmatic system is always written and read in 36 channels of recording, of which 5 are control and 31 are information channels. The tape format is identical, for both on-line and off-line (periphery) operations.

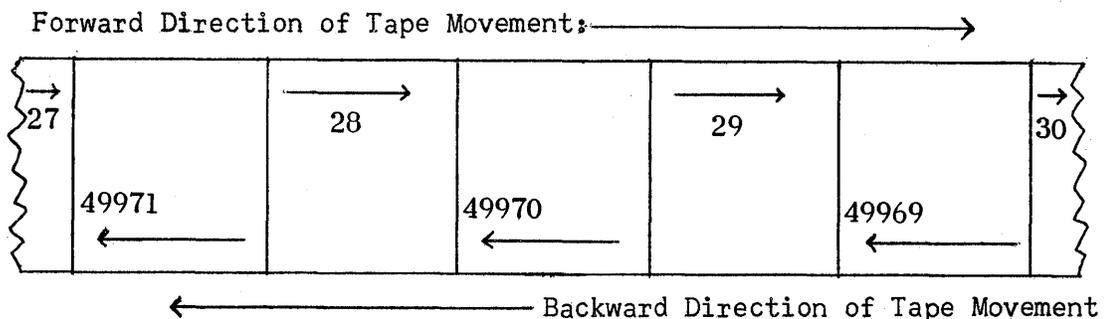
1403.1 Uses of Control Channels. The specific purpose of each of the five control channels is not certain, but these functions are accomplished:

1. "Block" marks, dividing the tape into usable sections of recording surface. Bad spots, or areas of the tape with flaws, are rejected and by-passed by these block marks.
2. Markers indicating the physical beginning and end of useful recording area on the tape.
3. Separation or distinguishing of block marks to denote those written in the forward direction of tape movement from those written in the backward direction.

A 2700-foot length of tape is divided into 50,000 usable blocks. It is probable that the five control channels are permanently recorded at the factory and, in operation, are used only for reading.

1403.2 Division of the Tape Into Blocks. The division of the tape into permanently-recorded blocks results in "gaps" between successive logical blocks which are the same length as the blocks themselves--about .67 inches. In the forward direction of tape movement, every other block space is used; in the reverse direction, the intervening spaces are utilized. Thus a full reel of DATAmatic tape can be packed solidly with data, except for the imperfect areas rejected during original inspection.

The interlacing of blocks results in a tape recording depicted in this illustration (using a 50,000-block tape):



The control channels on the tape contain recordings which indicate the physical limits of the tape; these prevent execution of a tape instruction calling for continued movement in the same direction. It is, thus, not possible to read or write "off the end of the tape."

- 1403.3 Format of a Block. A block of 62 DATAmatic words, each of 52 bits, is recorded in 31 information channels on the tape. Each channel contains two words, recorded serially; a block thus consists of 31 information channels, each with 104 bits of recording.

Omitting the five control channels, the recording of a DATAmatic tape may be depicted graphically:

Channel 1	Word 1 (52 bits)	Word 2 (52 bits)
Channel 2	Word 3 (52 bits)	Word 4 (52 bits)
Channel 3	Word 5 (52 bits)	Word 6 (52 bits)
.	.	.
.	.	.
Channel 29	Word 57 (52 bits)	Word 58 (52 bits)
Channel 30	Word 59 (52 bits)	Word 60 (52 bits)
Channel 31	Word 61 (52 bits)	Word 62 (52 bits)

The 31 channels are read in parallel; thus, one bit from each of 31 words is read and assembled in the input buffer. The writing operation is the converse of this.

- 1403.4 Method of Handling Tape "Skew." Problems engendered because of "skew" in the tape--that is, the slight twisting of the tape that keeps all bits from coming under the heads simultaneously--are discussed in 1109.07. With a 3-inch wide tape, it would be expected that skew difficulties would make successful reading almost impossible. In the DATAmatic, this difficulty is overcome by what is called "self-sprocketing" of each channel.

In other types of tape recording techniques--such as those used in Univac and the IBM series--there are actually magnetized and unmagnetized "spots" on tape. Because the gaps between tape blocks are also unmagnetized, it follows that each character code on tape must have at least one binary "1," and the entire character is read in parallel. Excessive skew rather evidently causes erroneous reading. In the DATAmatic method, there are no unmagnetized spots on the tape. For practical purposes, it can be considered that a binary "1" is a "big one" and a "0" a "little one" on the tape. Every bit position has either a "big" or "little" one, and each channel has exactly 104 magnetized spots in one block. Thus each channel can be made "self-sprocketing," and the bits assembled into a buffer independent of any other channel--it is only necessary to be sure that 104 are read from each of the 31 channels. Essentially, this is the technique followed in the Model 1100 Tape Unit.

2520 IBM TYPE 734 MAGNETIC DRUM STORAGE UNIT

The Type 734 is a magnetic drum for use in the 702 and 705 EDPs in an on-line operation under central processor control.

2520.1 Cost:      Purchase:      \$  
                                  Rental:                      2,800 per month

2521 CHARACTERISTICS OF THE TYPE 734

Type of data recording:	702/705 character codes
Drum capacity:	60,000 characters
Division of Information:	300 sections
Characters per section:	200
Addressability:	Each section addressable
Address range:	1000-1299 for first drum; 1300-1599 for second drum; etc., to 9700-9999 for 30th drum
Number of drums possible:	Maximum of 30
Drum size (physical):	10.7" diameter by 12.5" long
Drum rotation speed:	3750 rpm
Time per revolution:	16 ms
Average access (latency) time:	8 ms
Character transfer rate:	0.04 ms per character (40 microseconds)
Method of Transfer:	Bit-parallel for each character, serial by character
Buffer:	None
Number of information bands:	30
Sections per band:	10 (interlaced)

2522 METHOD OF OPERATION OF THE TYPE 734 MAGNETIC DRUM

The basic method of operation of the Type 734 is identical in both the 702 and 705 systems, with the minor exception noted in 2522.1. The time required to execute the instruction sequences writing on or reading from the drum are, of course, different, the 705 being faster.

- 2522.1 Writing on the Drum. Transfer of data from memory to the drum requires a two-instruction program sequence:

SELECT	xxxx	where xxxx is the drum section address
WRITE	m	where m is the memory location of the first character to be written.

In executing the write instruction, the EDPM waits until the first character position of the addressed drum section is under the heads; this average latency time is half a drum revolution, or 8 ms. Once this point is reached, the first (addressed) character in memory is moved to the drum and thereafter characters are transferred from successively higher memory locations to the drum at the rate of one every 40 microseconds. The operation is terminated by sensing a group mark (705) or record mark (702) following the last data character in memory. The group or record mark is not itself placed on the drum, but is automatically converted to a drum mark (character code 0 00 0000) which is inserted following the last data character on the drum.

The central processor is "tied up," and can do no other operations, during the entire time required to complete the transfer from memory to the drum. This time averages  $8 \frac{1}{2} .04 N$  ms, where N is the number of characters (including the group or record mark) to be written. The time for any given transfer, of course, is variable, depending upon the relative position of the addressed section to the heads at the time the "write" instruction is executed.

- 2522.2 Reading From the Drum. This is the exact converse of writing, and also requires a two-instruction sequence:

SELECT	xxxx	where xxxx is the drum section address
READ	m	where m is the memory location into which the first character is to be placed.

Once the addressed section is under the heads, the first character is transferred into the addressed memory location, and successive characters from the drum are moved into the next higher memory addresses at the rate of one every 40 microseconds. The transfer is terminated by sensing the drum mark, which is automatically converted to a record mark (702) or group mark (705) following the last data character placed into memory. The average time for executing a read instruction is identical with that for a write.

- 2522.3 Number of Characters in a Drum Record. Although each drum is divided into 300 addressable sections of 200 characters each, records of any length may be written on or read from it. The record or group mark in memory terminates the writing operation; the drum mark ends the reading. Records of more than 200 characters are placed in the drum sections following the one

2522.3 (Continued)

addressed, the drum mark following the final data character. Thus, although the drum sections consist of only 200 characters each, any number of them may be chained together to store records of longer length. The following comments are pertinent:

1. The drum section addressed always reads or writes beginning with its first character position; i.e., it is not possible to begin reading or writing someplace in the middle of a section.
2. The drum mark follows the last valid data character, and may appear in any character position within a section. Thus, a 60-character record would have a drum mark in the 61st position; a 200-character record would have a drum mark in the first position of the next drum section.
3. Those character positions within a section following a drum mark are inaccessible and unusable. Reading or writing always starts with the first position and always terminates with the drum mark.
4. It is not possible to "write off the end of the drum." That is, if the last section of any one drum is filled with data characters and more characters (or the record or group mark) still remain to be read, an error condition exists and an appropriate signal is generated. This statement is also true if two or more drums are included in an installation; it is not possible, for example, to split a 1,000-character record into the last three sections of one drum and the first two of the other (this can, of course, be programmed with two sets of drum write instructions).

The number of characters in any one drum record is completely variable from a single character up to one less than drum capacity.

2522.4 Errors in Drum Transfers. Each character transferred to or from the drum is given a parity bit test; the read-write check indicator is turned on whenever an invalid bit combination occurs. This indicator can be set to automatic machine stop or placed under stored-program interrogation and control. An attempt to write off the end of the drum also turns on this indicator.

Although magnetic drum recording is the most reliable of any of the magnetic recording media, it is interesting to note that, in one 702 installation, IBM customer engineers have recommended that programs provide for immediate re-reading of new records if they are to remain on the drum for any length of time.

COMPARISON OF UNIVAC II AND IBM 702/705 WITH DRUMS

Because Univac II does not have a drum, a direct comparison between it and a 702/705 with a drum is not possible. There are unquestionably applications in which the availability of the drum permits a 705 to perform productive work more cheaply than without one. A single application is not, of course, a proper criterion. Because anything that can be done with a drum can also be done with magnetic tapes (in a different manner), the question reduces to the determination of the system configuration which provides the most economical overall performance.

At \$2,800 a month, a magnetic drum represents an approximate 10-13% increase in the rental of a typical 702/705 installation (central processor and tape units only); the exact increase varies with the particular tape and tape control unit setup included. To be economically justified, a drum should enable the system to turn out least that percentage more work. The addition of more drums beyond the first requires a proportionately greater increase in the volume of productive output. The comparison of the cost-of-performance of a Univac II and a 705 with drums becomes, then, the determination of the time and cost required to accomplish the total overall work of a given installation under the two methods. Rather evidently, this determination cannot be given in specific terms, but is subject to analysis of each customer's particular processing requirements; some general guidelines to assist in making this evaluation can be given.

2531.1 Table Lookup. The term "table" conventionally refers to such types as mathematical values, freight rate classifications, public utility rates, mileage between cities, loan repayment schedules, etc. As used herein, it is extended to include also such types as cost distributions and statistical summaries which are prepared for presentation in tabular form. When related to the sequence of the data generating the lookup requirement, most of these are referred to at random, but some mathematical tables are referenced sequentially for each iteration of a problem solution.

When stored on a Type 734 drum, a table is normally broken down into sections each approximating 200 or fewer characters. To be used, these sections must be "indexed" in the main memory; that is, it is necessary to maintain in memory an index showing either the low or high table entry in each section, together with its drum address. A part of the memory must be set aside for holding this index, and both memory and computing time are needed for the instructions necessary to search the index and find the drum location of a particular table value needed. The searching method most economical of instructions -- sequentially through the index by a loop -- is unfortunately the most wasteful of computer time; the method normally most economical of time -- binary searching -- is probably the most wasteful of instructions. (Occasionally, the drum address can be calculated from the entry itself, eliminating this search). With 200-character records on the drum, average time to obtain any one at random is 16 ms (8 ms average latency plus 8 ms transfer time for 200 characters), plus the computing time necessary to determine the correct drum location; this can add anything from another 5 to 20 or more ms.

## 2523.1 (Continued)

The first type of table mentioned above is normally not changed during the processing, and is subject to reading only. The second type must be continually revised as new data are posted, and require both reading and writing. With records of 200 characters, this adds another 16 ms, approximately, to the time required to make one reference to the drum.

Without a drum, there are two basic methods of accomplishing on magnetic tape the equivalent of a table lookup. The first is to store the table sequentially on tape, and to scan this tape forward or backward, as appropriate, to find the required sections. The time necessary for this is basically a function of the size (number of blocks) in the table, but it always requires considerably more actual time than a drum reference. Like a drum reference, it requires computation to determine the particular tape block needed for any specific table value.

The second method is to sort the input data into sequence on the table index numbers and process these sequentially against a magnetic tape containing the table. With the first type table, this often requires a double sort -- first into the sequence of the table and second back into the original sequence. With the second type of table (tabular information), it usually requires only one sort. The double sort of the first type table lookup can sometimes be reduced to one by proper planning; the original sort can place the items in sequence for the sequential processing against the table.

Sometimes a third possibility exists; this is to calculate the table values as they are required. Obviously, this can only be done in those cases in which a definite formula can be used to calculate a particular value.

In table lookup operations, the determination of which is faster and cheaper -- drum reference or tape processing -- revolves principally around the time required for sorting. This, in turn, is a function of the number of words in an item and their total number. For fairly small volumes and short items, the additional pre-sorting (if necessary), sequential processing, and final sorting may take less time, overall, than the drum lookup and processing. If the items involved are long, or many in number, the odds become more in favor of the drum lookup. Standard sorting tables can be used to assist in computing the relative times, and costs, of the two methods.

2523.2 Instruction Overflow. When programs become too large for the memory capacity of the equipment, a drum becomes a convenient place to store the overflow routines, which usually are those to handle more or less "exception" conditions, or correction sub-routines. These are read in from the drum as required in a manner quite similar to that customary when using magnetic tape for the overflow.

Because of its "random access" characteristics, the drum practically always makes it possible to obtain an instruction sequence in less time than is required with tapes. The time saved is largely a function of the number of overflow instruction blocks incorporated into a program.

2523.3 Data Overflow. Occasionally, it is necessary to provide temporary storage for part of an item which is too long to contain in its entirety within memory. For example, information in the master portion at the beginning of an item may require change based on an action in a trailer portion which may be several blocks away. In this case, the first part of the item needs to be placed temporarily on tape or drum until the detail processing has been completed. Either one can be used for this purpose, and it is standard practice in Univac installations to use a "scratch tape" for this purpose.

Comparing Univac II with the 705, the actual time required to write out the overflow is about the same; drum latency average time and tape start time in the two systems are about the same, and successive characters are transferred every 40 microseconds in both systems. However, Univac can continue reading in and processing subsequent blocks during most of the tape writing period; the 705 can read into a tape control buffer (this assumes it has Type 777 Tape Control Units), but can do no processing during the memory to drum transfer. To offset this, Univac would need time to rewind the overflow tape after the last temporary block had been placed on it; the 705 does not require a corresponding action with the drum. Because of tape reversal times, the 705 to drum method will usually be somewhat faster, but not significantly so.

2523.4 Summary Comments on Drum Evaluation. The fact that the 705 can have one or more drums can be disregarded when building up the machine runs needed by Univac II to do the job of a potential customer; the goal should be, as always, the best Univac procedure. A drum is NEVER A NECESSITY -- it is a convenience, at times, and can be justified only to the extent that it pays its own way. It frequently does save time, and if a potential customer emphasizes processing time without regard to cost, there isn't much that we can do. Fortunately, most customers are cost conscious, and they can be impressed by the fact that each drum must give them an average of AN EXTRA HALF DAY OF WORK EVERY WEEK. Some savings on one or two jobs are not enough -- it must justify its cost over a full base period of operation as a minimum.

It has been noted in some instances that an unjustified assumption often is associated with use of a drum. The fact that it may save time in one or two runs involving table lookup becomes the basis for jumping to the conclusion that savings accrue whenever it is used. Thus, for example, it may be assumed that using the drum for overflow program instructions is the most economical processing method; the fact may well be that extraction of a work tape and subsequent processing in another machine run is the more economical approach. One of the pitfalls of efficient EDPM processing is to view the drum with the same outlook as the mountain climber contemplating Mount Everest -- simply because it's there.

2529 SALES POINTS ON THE IBM TYPE 734 MAGNETIC DRUM

These points will be useful in combating IBM proposals involving one or more magnetic drums in a 702 or 705 EDPM system.

- 2529.01 A Drum Is a Necessity. The claim may be made that the expansion of memory because of a drum makes possible a reduction in machine operating time, facilitates securing overflow instructions and permits "table lookup" type processes.

For the 734: These are all possible uses of a drum.

Against the 734: See 2523 for the economic considerations which must be taken into account. A drum is economically justified only insofar as it "pays its way."

- 2529.02 Justifying the Cost of a Drum. The monthly rental of a drum is \$2,800, or 10-13% of the cost of a typical 705 central processing unit.

Against the 734: Restricting consideration to the 705, a drum must add at least as much productive work as it costs; this is equivalent to a minimum of another half-day a week. It cannot be justified, on cost grounds, unless it does add this much output to the installation.

- 2529.03 Univac II Without Drums Against the 705 With Drums. The best counter-proposal to a 705-with-drums proposal is a sound, well-thought-out Univac method. There is ALWAYS AN ALTERNATIVE METHOD OF PROCESSING, and the development of the Univac procedure need not consider what IBM may propose with drums. The customer's decision will practically never be based on the availability of drums alone; he is looking for the best data-processing system. A drum is only one part of the overall installation.

- 2529.04 Characteristics of the 705 With Drums. These are covered in 2522. In discussing the counter-proposals with a customer, the advantages and limitations of the Type 734 can be outlined. IBM may not always point out that there is a good deal of lost time in using it with the 705.

3110 IBM TYPE 711 CARD READER

The IBM Type 711 Card Reader is designed to read standard 80-column punch cards in on-line operation only in the IBM Type 701, 704 and 709 EDPMs. It is not capable of independent off-line operation, but operates always strictly under stored-program control. Two models, differing in card reading speed, are available; in general, Model 1 is used in the 701 system, but can be used with the 704; Model 2 is normally used with the 704 and, so far as available information indicates, is the only version which can be connected to the 709.

Model 1 -- 150 cards per minute

Model 2 -- 250 cards per minute

3110.1	<u>Cost.</u>	<u>Purchase</u>	<u>Monthly Rental</u>
	Model 1		\$ 200
	Model 2	\$ 52,000	800

3111 CHARACTERISTICS OF THE TYPE 711

With the exceptions noted, characteristics of the two models are identical.

Type of cards read:	80-column
Number of columns readable:	72
Type of card punching read:	Conventional punch card code or binary (see 3112)
Card reading speed (continuous)	Model 1 -- 150/minute Model 2 -- 250/minute
Control panel:	Yes (see 3112)
Number of reading stations:	Two (one for control punches only, one for reading).
Checking of card reading	None
Memory in card reader:	No
Method of reading:	Card image: 36 columns of one card row are transferred as a 36-bit word
Types of selector pickups:	(1) Immediate (2) 11-12 punch ("X" pickup) (3) 9-12 punch (digit selection) (4) Card cycle (5) Split column control

The card feed is a slight modification of that used in the Type 402 Tabulator. The control and reading stations correspond to the second and third reading stations of the 402.

3112 METHOD OF OPERATION OF THE TYPE 711 CARD READER

Both models of the Type 711 read 72 columns of an 80-column card and transmit the information into the EDPM memory as 24 words, each of 36 bits; each word consists of binary 1's or 0's corresponding to the presence or absence of a punch in each of 36 columns of one card row. The memory representation thus is a complete image of the 72 card columns read.

Through control panel wiring, it is possible to select any 72 card columns desired and to rearrange them in any manner desired to create the 36-bit words. Variation of word formats (i.e., card columns which are read into specific bit positions) is possible both within a single card and between cards through use of the pickup selectors. For the purpose of such data rearrangement, the control panel is provided with 12 five-position co-selectors and two ten-position pilot selectors. To simplify the explanation following, it is assumed that the first 72 card columns are being read and that the first 36 are converted into one set of 12 words and the last 36 into the other set.

Each row of a card is read in two separate reading cycles of 36 columns each; these are called the "left-read" and "right-read." Specifically, all 80 columns are actually read in each cycle, but only the 36 columns wired into the "entry left" and "entry right" hubs, respectively, generate impulses which can be transferred to the EDPM. Reading of a card is 9-edge first, with the 24 cycles following each other as indicated in this illustration:

	1		36 37		72 73 -
12		\ 23		24	80
11		21		22	
00		19		20	Unused
11		17		18	Columns
2		15		16	
3		13		14	
4		11		12	
5		9		10	
6		7		8	
7		5		6	
8		3		4	
9		1		2	

Once the feeding of a card into the reader has been initiated, the movement of a card through the read brush station is automatic, the successive 36-bit words becoming available at specified time intervals. The essential difference between the Model 1 (150 cards/minute) and the Model 2 (250 cards/minute) is the time interval between successive card rows

3112 (Continued)

and also between the end of one card and the beginning of the next. The timing chart for the Model 1 card reader is depicted on the next page; that for the Model 2 is similar, except for shorter time intervals.

The cards read may be punched either in pure binary or conventional card codes. In the former case, relatively common in scientific applications, it is customary to prepare a card with up to 24 binary words, each of 36 bits (sign plus 35 data bits), each word occupying 36 successive columns of a row. (Normally, columns 1-36 and 37-72 are used). A punch is converted to a binary "1" and no punch to a binary "0". Cards prepared in this fashion result in a computer word which can be used without change in computation.

If a card is punched in conventional card codes, the resulting transfer of 24 words into memory results only in a "card image," and the conversion program is necessary to transform the information into a form the EDPM can use. For example, a "6" in card column 36 is transferred into memory as a binary "1" in the right-most bit position of the 7th word read in; the program must recognize this particular word as representing the decimal digit "6" and provide for converting each bit in the word to the binary representation for a "6." The actual program can become very detailed, since it is also necessary to know whether a particular bit position represents the units, tens, hundreds, etc., position of a decimal word.

Alphabetic information in punch card code is converted into two binary "1's" in two different words transferred into memory. An "E" in column 36, for example, becomes a "1" in the right-hand position of the 9th word read in (5-row left) and 23rd word (12-row left). The conversion of alphanumeric information into a form suitable for processing in an EDPM basically binary internally requires extensive programming and a considerable amount of computer time. In the Type 701, for example, the conversion of a 72 column input involving some non-numeric fields cannot be accomplished within the 400 ms card reading cycle.

In either type of card recording, the impulses representing each word are not stored, but are available only for a limited period of time during each of the 24 reading cycles on wires reading directly into the EDPM. If it is not ready to receive the word at that time, the information is lost. The timing impulses determining when the reading of each successive word occurs are a function of the clock mechanism of the card reader itself, rather than the EDPM. Because there is a difference in the manner of transferring information from the card reader into the 701/704 as compared with the 709, the two methods are discussed separately in the next two sections. The remarks of this section are common to all three EDPMs.

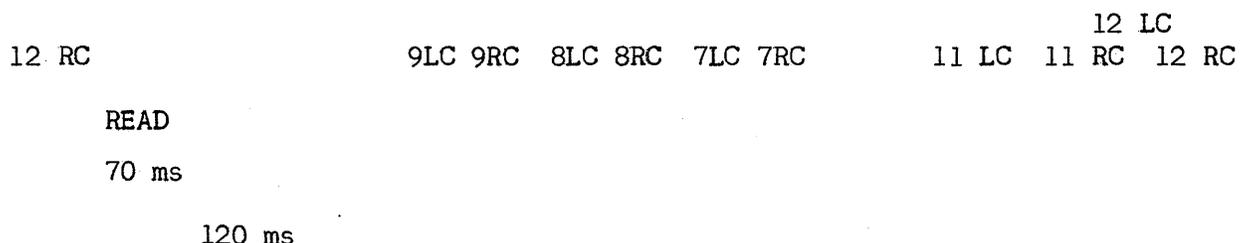
3112.1 Card Reader to Memory Transfers -- Types 701 and 704. In both of these equipments, the transfer of information from the card into memory requires stored-program instructions. Two separate ones are involved:

- (1) A "Prepare to Read," with an address of the card reader, causes a card to feed through the reading stations.
- (2) A separate "Copy" instruction for each word of the card to be transferred into memory. The address of this instruction designates the memory location into which the word is to be read.

Once the card feed has been initiated, the movement of the card through the reading station is automatic, and the successive words become available at specified time intervals. The program must provide that "Copy" instructions are ready for execution at these required times in order to assure transfer of all information in the cards. Consequently, care is required to assure that the time limits between the card reader instructions are observed.

A brief discussion of the time limits applicable to the Type 711 Model 1 will show the considerations involved. If continuous card motion is maintained, the complete card cycle will require 400 ms (at 150 cards/minute). Of this, 292 ms are involved in the time limits the programmer must observe; the remaining 108 ms are required for execution of "Copy" instructions and for the necessary tolerances in the mechanical card feeding mechanism.

The essential portions of the timing chart, beginning with the transfer of the last word of the previous card, are depicted:



These times add up to about 292 ms, the maximum safe available computing time during the card reading cycle. The various time intervals which are involved are described briefly.

Execution of "Read" Instruction. For continuous card movement, a "read" instruction must be executed within 70 ms after reading the last word (12 RC = 12 right copy) of the previous card. However, the "read" cannot be executed until 20 ms has elapsed after the 12 RC; consequently, full card speed with maximum internal computing time is attained if the next "read" is executed between 20 and 70 ms after the previous 12 RC. If it is reached in the program before the 20 ms has elapsed, the EDPM simply "hangs up" until the instruction can be executed; there is in this case no loss of card speed, but some loss of computing time.

Tolerance Time Before First Word From Card. At full card speed, an average time of 120 ms elapses between reading the 12-row of one card and

## 3112.1 (Continued)

the 9-row of the next. However, only 70 ms is available for useful computation; the remainder is necessary for tolerances in the card feeding mechanism. If the card reader is stopped at the time a "Read" instruction is executed, an average of 270 ms elapses before the 9-row is under the read brushes, but again only 50 ms can be used for computing.

Time Between Left and Right Words of One Row. Each word is transferred in parallel (36 bits at once) from the read brushes to the MQ-register and thence into memory by a "Copy" instruction. The safe computing time available between the two words of each card row is 540 microseconds. Normally, this time is used to modify the memory address of the "Copy" instruction in the so-called "copy loop."

Time Between Successive Card Rows. 15 ms of useful computing time is available between reading the right-hand word of one row and the left-hand word of the next row. The nature of the computation is immaterial.

Because the word transfer from the card reader into memory is through the MQ-register, it is normally necessary to be certain no useful data is stored there at the time a "Copy" is given; its previous contents are destroyed. The MQ-register can be used, without restriction, during the computing time available between the successive "copy" instructions.

In the event a word is ready for transfer before a "copy" instruction has been reached in the program, the EDPM stops with a "Copy Check" light on the Operator's Console. The card continues feeding to the end of the normal cycle.

Reading of a card may be stopped at any point by executing another "Read" instruction; the card moves to the end of the cycle, but no additional words can be transferred into memory. In this case, the "Copy Check" light does not go on and the EDPM continues in operation.

If a 25th "Copy" instruction is reached in a program, the EDPM recognizes the fact that the end of the card has been reached, and automatically skips to the third following instruction. This feature is conventionally used as a means of exit from the "copy loop;" it eliminates the need for counting the number of words which have been moved into memory.

The various time intervals available for useful computation during the card reading cycle are beyond the control of the programmer. Necessary impulses are generated by the card reader itself, and "copy" (or "read") instructions must be waiting execution when a word is ready for transfer. The limits indicated are the maximum safe computing times available; the actual times are somewhat greater, but the difference is needed to provide a sufficient margin of tolerance for the mechanically moving card reader.

It will be noted that a card rate of 150/minute is achieved only in continuous card movement; this requires the next "read" instruction to be executed between 20 and 70 ms after the last "copy" for the previous card. Intermittent card movement reduces the rate to about 125 cards/minute.

3112.2 Card Reader to Memory Transfers -- Type 709. One Type 711 Model 2 Card Reader can be connected to one channel of each Type 766 Data Synchronizer in a 709 installation. Thus a maximum of three can be connected to the system. In this use, it is necessary to have a Type 716 Printer also attached to each channel using a Type 711.

The basic method of card reading is identical to that of the 701/704 EDPs, with the exception that the "copy loops" of those systems are not necessary; the control circuitry of the Type 766 takes over these functions. (Refer to 1160 for a discussion of the characteristics and operation of the Type 766). The successive "words" from the card are transferred into the synchronizing register of the Type 766, and thence into memory, using memory addresses and word counts contained in the Control Word. The essential method of transfer is the same as that applicable to tape units, which is discussed in detail in 1160. As is the case in tape reading, the transfer of the word from the synchronizing register into memory is accomplished by "interrupting" the main program for one cycle; this is completely automatic and requires no attention by the programmer. The transfer of the word from the reading brushes to the synchronizing register is controlled by timing impulses generated by the card reader itself.

The Data Synchronizer Channel can be used for only one input or output operation at a time. Therefore, when reading a card, it is tied up for 250 ms (at 250 cards/minute), and is unavailable for other inputs or outputs during that period. The card reading may, of course, be interrupted at any point by one of the means mentioned in 1160, but the card completes its movement at the end of the cycle.

3112.3 Checking of Card Reading. Both models of the Type 711 have only one effective reading station and the transfer of information is unchecked. Similarly, the sensing of control punches in the first reading station is unchecked. It should be noted that there are at all times (except the end of a deck of cards) two cards in the feed path; the "read" instruction advances the card which has just passed the first (control punch) station through the reading station and makes its information available for transfer to the EDP. Simultaneously, a card is fed from the hopper and passes through the first station, where it is read for control punches only.

3112.4 Timing for the Type 711 Model 2. A timing chart is not available for the Model 2 Card Reader. It is, of course, similar to that of the Model 1, section 3112.1, except that most intervals will be shorter. Knowledge of maximum safe computing times at various points of the card reading cycle is necessary with the Type 704, which requires "copy loops," but is not needed with the Type 709.

3113 USE OF THE TYPE 711 CARD READER IN BUSINESS DATA PROCESSING

Until the Type 709 is available for delivery (announced for late 1958), use of the Type 711 Card Readers is limited to the 701 and 704, both of which are used primarily for scientific computations. There are, however, some 704 installations which are planning to do some commercial data processing, particularly payrolls; in fact, suggestions from these users are primarily responsible for many of the internal changes planned for incorporation in the 709. The discussion of this section is limited largely to use of the 711 in a 709 business data processing installation.

Of course, it should be noted that some potential 709 users will be combining both scientific and commercial problems on the equipment. It is fairly common to find scientific source data in binary form and, in such instances, uses of the 711 is almost mandatory. One card is capable of containing up to 24 36-bit words of binary information, and this can be transferred into the EDPM only through the 711. Independent card-to-tape conversion is impossible; all of them handle character codes only, not straight binary information. If a 709 is to be used partially for scientific calculations, and also if sufficient binary input is involved to justify a directly-connected card reader, then the 711 may be warranted. It should be observed that its use demands a Type 716 printer also, and thus the economics of a 711 is closely wrapped up with a 716; some discussion of the printer is contained in 4110. Because a combined 711/716 costs \$2,000 monthly, their total use should justify this expenditure of money.

In a business installation (either exclusively or in part), the use of a Type 711 must be weighed against an off-line Type 714 Card Reader, which with a tape unit aggregates a minimum of \$2,950 monthly. This section discusses some of the factors to be considered.

- 3113.1 Effect of Alphanumeric Nature of Business Data. Business data are almost exclusively decimal or alphabetic in nature; although master files on magnetic tape may be maintained partially in binary (at least on the 709), basic inputs and outputs consist of decimal digits, letters and special characters. Most of us, certainly, are not particularly facile at reading \$5,621.87 in binary notation--10001001010000001011. Inputs and outputs in commonly-understood alphanumeric representation are mandatory.

Because the Type 711 transfers only a 24-word "card image" into memory, an extensive conversion program would be necessary to create binary-coded decimals or, in the case of numeric data, binary words. How long this will take has not yet been determined; as a point of comparison, the 701 could not convert 72 characters in the 400 ms card reading cycle. Although the 709 is faster internally, it is rather evident that it will spend considerable time, and require a fairly lengthy program, to do this conversion. Effectively, then, it will be tied up for a considerable part of the card reading cycle on transforming the card image input into a usable form. By reducing its ability to do other productive work (using other Data Synchronizer channels for input and output), the 709 becomes an extremely expensive input device. It may be useful to develop a simple

3113.1 (Continued)

economic comparison. The question to be answered is this: Which is cheaper, reading cards directly or through a Card-to-Tape conversion? Suppose the 709 CPU rents for \$50,000 monthly (somewhat on the low side). Further assume that the internal conversion of a card read on the Type 711 takes 100 ms--which probably is not too far from the actual time. This means that 40% of the card reading cycle (about 240 ms) is spent on transforming the input information. Thus, during the time it is reading cards, the 709 is costing the equivalent of \$20,000 a month for this purpose. An off-line C-T converter costs about \$3,000 a month--15% as much. Thus if the 709 spends more than 15% of its time reading cards, the percentage of its costs chargeable to card reading exceeds that of off-line operation. 15% of an 8-hour day is 1.2 hours, time for 18,000 cards at full reading speed. That's a fairly modest card volume.

3113.2 Character Transfer Rate. At 72 characters maximum per card, the transfer rate of information into memory, at full reading speed, is about 300 characters per second. This is pretty slow. Although it may not be critical in some applications--for example, posting a low-density file--in others it may cause considerable EDPM "stalling" waiting for the next card to be read. This is an example of the typical case of the mismatch occurring between a slow input/output unit and a much faster CPU. As a further note, this transfer rate is about 2 $\frac{1}{2}$ % of that obtainable from tapes. Not only is one Data Synchronizer channel working at much less than capacity, but the internal computing time required to transform the card image is many times that necessary for a tape resulting from C-T conversion.

3113.3 Lack of Checking. A major disadvantage of the Type 711 in a business process is its complete lack of checking. A card is read once--and only once. Because the reading mechanism is electro-mechanical, the error incidence to be expected is much higher than in internal EDPM processes. There is no economic method of assuring accurate card reading, or of detecting errors when they do occur. Batch proof totals may be used to check the accuracy of reading the fields involved, but there is no certainty that the other (an usually more numerous) columns have been correctly read. Foolproof error-detection could be provided only by accumulating "hash totals" for all card fields, recreating these totals internally as the cards are read, and comparing against periodic "hash total" cards. Such a procedure would indicate an error, but would not provide for correcting it. It also is uneconomic; the proof totals would have to be computed prior to the card reading, on equipment external to the EDPM; at least one special card pass would be necessary on a tabulator to compute these totals, and more might be required.

Without such "hash totals," it is extremely doubtful if any auditor would accept any accounting procedure using Type 711 input to an EDPM. As a useful means of data input, this unchecked card reader is definitely unattractive.

3119 SALES POINTS ON THE TYPE 711 CARD READER

Index numbers 3119.01--3119.49 are reserved for pointers applicable largely to business operations; 3119.50--3119.99, for those of use primarily in scientific applications.

- 3119.01 Lack of Checking. The Type 711 (both models) has only one effective data reading station. There is no verification that the word images are a correct representation of the actual punches in the card.

Against the 711: In business operations, particularly applications in which accuracy is mandatory or highly desirable, extensive external checking features must be incorporated into the processing system to assure correct card reading. Any feasible method of providing such checks can be only an error-detection scheme, which would not indicate the specific card read incorrectly, but only isolate an error to a batch of cards for which a series of proof or "hash" totals has been generated prior to the card reading.

- 3119.02 Reading of Alphanumeric Information. The Type 711 reads a card and transfers the data into memory as a series of 24 "card images," each representing 36 columns of one card row. Information in standard punch card code requires programmed conversion into either internal character codes or pure binary numbers.

For the 711: The reading method permits easy handling of pure binary information. In addition, since a complete (72-column) card image is moved into memory, multi-punched columns can be read without special handling.

Against the 711: The internal program time required to convert the 24-word card image into a usable internal language is considerable. Business data is almost exclusively in punch card codes, not binary.

- 3119.03 Elimination of a C-T Converter if a Type 711 is Otherwise Required. In proposals in which a Type 711 may be necessary to read binary scientific input, IBM may propose that this reader be used in lieu of a C-T converter (Type 714), simply to save the \$2,950 cost (including tape unit).

For the 711: This is a possible solution, and may result in a saving of about \$3,000 in basic installation rental.

Against the 711: There are two rebuttals.

- (1) With the 709, a Type 716 is required with a Data Synchronizer channel before the Type 711 can be connected. This boosts the cost of the two units to \$2,000 monthly, rather costly unless the printer can be utilized economically.
- (2) The proposal is economic only for small card volumes, because of the extremely high CPU cost of converting alphanumeric card input. For any specific proposal, an analysis similar to that of 3113.1 can be developed.

- 3119.04 Mismatch of Card Reading Rate and Computing Speed. The usual remarks on the speed mismatch between the relatively slow card reader and the fast CPU are applicable. Because computation time and card rate seldom are equal, one or the other is used at less than maximum efficiency.

3130 IBM TYPES 712 AND 714 PUNCH CARD READERS (CARD-TO-TAPE CONVERTERS)

The IBM Types 712 and 714 Punch Card Readers are designed to read standard 80-column punch cards and transfer the reading, in 702/705 character codes, either into 702 or 705 memory, under program control, or onto magnetic tape in an off line card-to-tape (C-T) conversion. The resulting tapes may be used as input to a 702 or 705, as well as to a 650 or 704, provided tape format restrictions of the latter two EDPs are met.

The Type 712 is an early version provided with the first Type 702s delivered. It has been superseded by the Type 714 and, as far as known, is no longer being manufactured, although some are still in use. It differs from the 714 in not having a plugboard and, as a C-T converter, in not verifying the accuracy of the tape writing. The discussion of this section is limited to the 714.

Both units consist of two physical cabinets: A card reader and a control unit providing power, timing and decoding circuitry. A Control Unit is required for each Type 712 or 714.

3130.1 Monthly Rental: Type 712 -- \$1,050 (including Type 756 Control Unit)  
Type 714 -- 2,400 (including Type 759 Control Unit)

There is a current tendency on IBM's part to quote separate rentals for each type number of an associated and integral group of type numbers. The 712 is separately quoted at \$750 monthly and the 756 at \$300, totalling the \$1,050 given above. Separate figures are not available for the 714/759; however, both must be used together.

When used in an off-line C-T conversion, both require a Type 727 Tape Unit at \$550 monthly additional rental.

3131 BASIC CHARACTERISTICS OF THE TYPE 714/759 CARD READER

Purpose of equipment	Direct on-line card reader or off-line C-T conversion
Type of punch card	IBM 80-column cards only
Type of buffer storage	Magnetic core
Character capacity of buffer	92 characters
Conversion of punch card codes	Automatic, to 702/705 character codes
Card reading speed	250 cards per minute
Magnetic tape unit (off-line)	Type 727
Transfer rate, buffer to tape	0.067 ms per character, plus start
Transfer rate, buffer to memory	0.0335 ms per character (702 and 705)
Plugboard editing features	Yes; see 3132
Multi-card conversion	One or two cards, not to exceed 92 characters total
Checking of operations	Yes -- all are checked
Length of converted record	Variable, up to 92 characters

In an off-line conversion, all normal characteristics of the Type 727 Tape Units are unaltered (see 1101).

3132 METHOD OF OPERATION OF THE TYPE 714/759 CARD READER

This section describes the plugboard editing features and briefly discusses on-line and off-line operation.

- 3132.1 Type 714 Control Panel. The entire operation of reading a card into the 714 buffer is controlled by plugboard wiring.
1. Reading Stations. There are two reading stations. In the first, a series of check bits are created. The second reading is the one actually transferred into the buffer after comparing with the check bits for accuracy. Wiring from the reading brushes into the storage entry hubs is completely flexible, and any desired rearrangement of the card format can be accomplished.
  2. Column Splits. Twelve sets of column split hubs permit separating "11" and "12" control punches from 0-9 information punches, and either recording them in separate character positions of the buffer, or eliminating them.
  3. Digit Emitters. Four digit emitters are provided: Two are available on the "12" cycle only, one on the "11" cycle, and one on the 0-9 cycles. These automatically generate impulses on the applicable card row cycles and permit including in the buffer record characters not punched into the card. The 0-9 emitter is normally wired through a digit selector to select the specific digits wanted.
  4. Digit Selectors. Two digit selectors are available. These permit breaking up a multiple-punched card column into several different character positions in the buffer, and are also used in conjunction with the 0-9 digit emitter to create constant characters not present in the cards.
  5. Record and Storage Mark Emitters. These are special-purpose emitters for creating the punch-card pattern for a record mark (which may be in the card also) and a record storage mark (RSM), a character code peculiar to the Type 714.
    - a. In buffer-to-memory or buffer-to-tape transfers, the operation is terminated by the RSM. Thus only the number of characters required is written out, provided the RSM follows the last data character in the buffer. If there is no RSM, the entire 92 characters in the buffer are transferred. The RSM is not placed in memory or on tape, but simply terminates the transfer operation. (The character code of the RSM is not known, and is immaterial. Its use is confined to the Type 714).
    - b. Emitting of record marks is provided solely as a convenience in future EDPM processing. They are handled as any other valid character code in buffer-to-memory/tape transfers.
  6. Alternators. These are used when two cards are to be converted into the buffer before its contents are transferred into memory or onto tape. Up to 46 characters from each of two successive cards can be

3132.1 (Continued)

handled in this manner. Identical columns must be read from each of the two cards in exactly the same manner; the alternator is essentially a 46-position "selector" which automatically shifts from one state to the other on each card cycle, enabling the two successive (partial) card images to be converted into different buffer positions.

All control panel function hubs are duplicated to permit verification of the card reading in the two read stations; this includes the emitters. (Instruction manuals state that there are 24 column splits and four digit selectors. However, because checking of the second reading against the first is automatic and includes everything placed in the buffer, these two features always require use of hubs in pairs, reducing the effective number to 12 and 2, respectively).

3132.2 Checking of Card Reading. Results of card reading at the two stations are checked adequately. The method of checking differs appreciably from that of the Univac C-T converter, but is considered fully satisfactory.

3132.3 Checking of Buffer-to-Memory/Tape Transfers. Direct transfers from the buffer into memory receive a check bit test only. Because the buffer record has already been verified, only a double error in the transfer of a single character code would be undetected. The probability of such an error is too small to worry about.

Transfers from buffer to tape, in C-T conversion, likewise are checked, but again in a manner different from Univac. The tape recording is re-read for both check bit and check character agreement, but is not compared against the actual buffer record. Again, the probability of an undetected error is negligible.

3132.4 Card Capacity per Tape Reel. The card capacity of a magnetic tape reel, in C-T conversion, varies with the length of the records converted and whether one or two cards are placed in each. These are typical maximum capacities for a full 2400-foot reel of tape; normal capacity is about 5% less.

<u>Characters per Tape Record</u>	<u>Cards per Record</u>	
	<u>One</u>	<u>Two</u>
40	30,300	60,600
60	27,400	54,800
80	25,000	50,100
92 (maximum)	23,800	47,600

For two cards per tape record, the number of characters from each card is half the figure in the left-hand column.

3132.5 Direct EDPM Operation of the Card Reader. The card reader is capable of operation hooked directly to either a 702 or 705 (with a 759 Control Unit) under stored-program control. In this type of use, the transfer of the contents of the buffer into memory automatically initiates the action of reading the next card (or cards); plugboard features are all operative during the conversion into the buffer. Up to 100 card readers can be handled simultaneously. Transfer of the buffer record into memory

3132.5 (Continued)

is accomplished in the conventional manner: A "Select" instruction, with an address in the range 0100-0199, alerts the specific card reader desired; a following "Read" instruction starts the transfer. This is executed at the rate of 0.0335 ms per character (in both 702 and 705), plus instruction time. This is approximately 30 characters per millisecond, rather slow compared with the internal speed, and as usual no other operations can occur during this transfer time.

An error in either the card reading or transfer from buffer to memory turns on the "0902 Read-Write Check Indicator," which can be set to automatic stop or to program control. The indicator does not distinguish between the two sources of error; therefore it is usually necessary to feed the card just ejected into the stacker as well as the two at the read stations back through the card reader.

Although the card reading rate of 250/minute (240 ms per card) is extremely slow compared with the EDPM speed, IBM has rather consistently advocated a directly-connected reader for loading machine programs, and for certain types of data input, such as those in which the time to find master file items stored on magnetic tape averages more than the card reading time (low activity file maintenance operations are typical). This latter use is probably secondary to the loading of program cards; if a 714 is to be connected for this purpose, it might as well be used for something during the main body of a machine run. The justification for maintaining programs in punch card, rather than magnetic tape, form is that corrections can be key-punched and manually inserted, without tying up the central processor; if programs are stored on an instruction tape, an EDPM run is obviously necessary to update or correct them.

3132.6 The 714/759 as a Card-to-Tape Converter. As an off-line operation, the 714/759 can be used with a standard Type 727 Tape Unit as a card-to-tape converter. There are no particularly significant features in this mode; it operates basically the same as the Univac C-T Converter. The tape prepared in this operation is acceptable to either the 702 or 705, as well as to the Type 650 and 704. For the latter two equipments, the conversion must be into a tape format acceptable to the respective EDPM (see 1103). There are, at this time, no known plans to use it with the 650, which has other (and cheaper) card readers in its periphery array. Its cost is probably too high to justify including it in a 650 installation unless special considerations make independent card-to-tape conversion necessary. There is, however, no technical reason to prevent its being so used.

3175 IBM TYPE 537 CARD READ-PUNCH UNIT

The Type 537 Card Read-Punch Unit is an auxiliary input-output device for the Type 650 EDPM, which is designed to permit both reading from and punching into a single card. It thus differs from the Type 533 Read-Punch Unit, which cannot punch into the card being read.

3175.1 Monthly Rental: \$ 700.

The 537 requires one input-output synchronizer on the Type 650 drum; one of these is included in the basic 650 rental, with two more optional at \$300 monthly each. Normally, the 537 will require an additional synchronizer. Alphabetic character devices may also be extra; see 6113.

3176 BASIC CHARACTERISTICS OF THE TYPE 537

The 537 card feed is essentially the same as the card-feed units used with IBM's Types 604, 607 and 608 calculating punches. Functional operations, however, are quite similar to those of the 533.

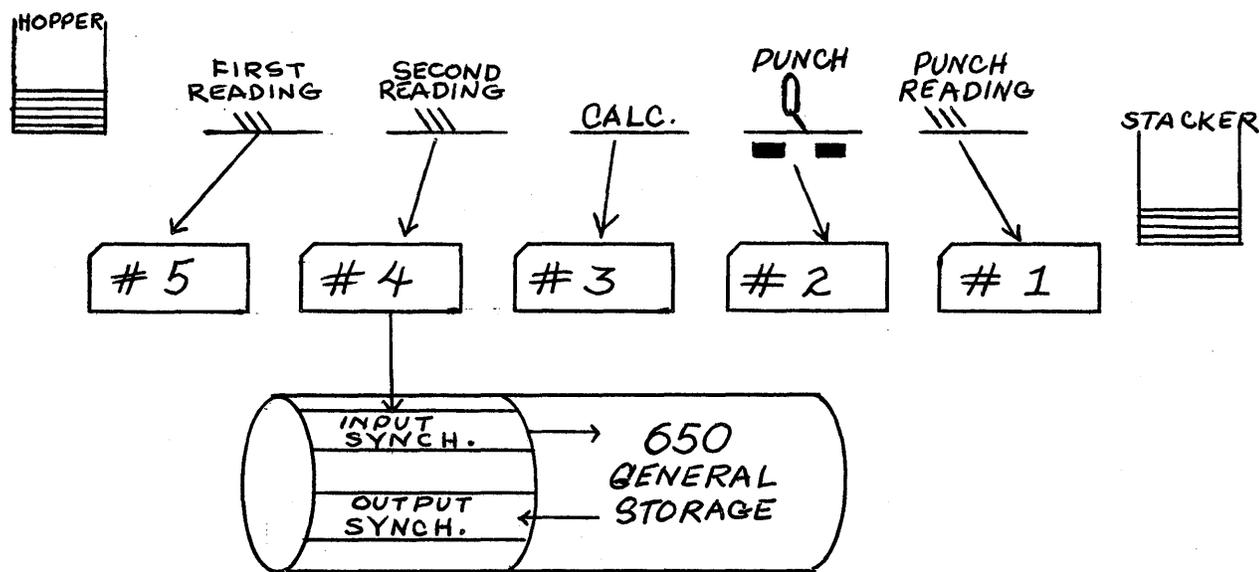
Card feeds	One
Card speed (maximum)	155 per minute, read and punch
Number of card stations	Five
Type of control	Plugboard and 650 stored program
Number of stackers	One
Handling of error cards	Offset in stacker

The plugboard (control panel) layout and terminology is identical with the one used with the Type 533 (see 3172), with the exception of a pair of "switch" hubs wired if the 537 is used for punching only. Wiring principles are identical with those of the 533, with a few exceptions. In principle, the first and second reading stations of the 537 are treated in about the same manner as the read feed of the 533; the punch and punch read stations correspond closely to the punch feed of the 533.

3177 METHOD OF OPERATION OF THE TYPE 537

The basic use of the Type 537 follows closely the principles of the Type 533. The same plugboard is used, although a few hubs have different interpretations.

3177.1 Type 537 Card Feed. The card feed is graphically portrayed thus:



There are five cards, one at each station, at all times (except initial loading and final runout); advance is initiated for all five simultaneously by feeding in a card from the hopper. This is done by a 650 "Read" or "Write" instruction. The five card stations and their uses are:

Card #1 is checked for double punches and blank columns in the Punch Reading station. Information in this card may also be plugboard-wired to gang-punch into the card at the Punch Station.

Card #2 is punched from the output synchronizer as it passes through the Punch Station. This normally is the result of computation performed by the 650 during the preceding card cycle.

Card #3 is in a "wait" status at the Calculating Station; the 650 is computing on data pertaining to this card during this cycle.

Card #4 is being read into the 650 input synchronizer in the Second Read station.

Card #5 is being read for control punches and selector pickups in the First Read station.

The control panel may be wired to read a card into the 650 input synchronizer from either the first or second reading station. The latter is normally used.

The 537 may be used for reading only, for punching only, or for both reading and punching.

4130 IBM HIGH-SPEED PRINTERS -- TYPES 719, 720 AND 730

The IBM line of high-speed printers designed for EDPM use consists of three similar versions differing only in maximum printing speed and number of characters per line:

Type 719 -- 1,000 lines per minute, 60 printing positions  
 Type 720 -- 500 " " " , 120 " "  
 Type 730 -- 1,000 " " " , 120 " "

The basic method of operation is identical for each type and all are discussed in this section. Each can be connected directly to a Type 702 or 705 EDPM, in program-controlled operation, or used in off-line printing in the same manner as the Univac High Speed Printer. In either case, each HSP requires one Type 760 Control and Storage Unit to provide power, control functions and storage facilities; off-line use requires also a Type 727 Magnetic Tape Unit, but two may be used. None of the printers can be used at any time without the Type 760.

In off-line operations, output tapes from the Types 650 or 704 EDPMs can also be used as printer input, subject to certain restrictions which are discussed subsequently.

4130.1 Monthly Rental: Type 719 -- \$ 1,400  
 Type 720 -- 1,400  
 Type 730 -- 2,100

Total monthly rental for working printer assemblies, including the control and tape units, are:

	On-Line Operation	Off-Line Operation	
		<u>1 Tape Unit</u>	<u>2 Tape Units</u>
Types 719/720	\$ 3,200	\$ 3,750	\$ 4,300
Type 730	3,900	4,450	5,000

These costs are for each printer. The figures for on-line operation do not include the costs of any tape units which may be associated with the 760 controlling the printer.

4131 PRINTER CHARACTERISTICS AND METHOD OF PRINTING

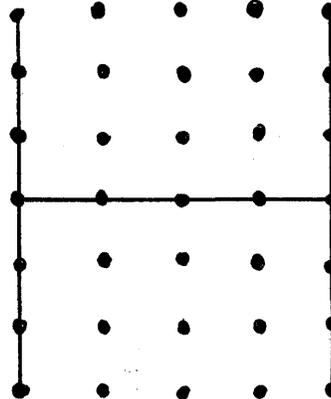
The characteristics and features in this section are common to all three printers, except for differences noted.

4131.1 Basic Printer Characteristics and Common Features

Maximum printing speed	1,000 lines/minute (719 and 730) 500 lines/minute (720)
Number of printing positions	120 (720 and 730) (60 (719))
Characters printed	47 (figures 0-9, letters A-Z, special characters & . - \$ * / , % # @ $\square$ )
Printing positions per inch	10 (6-inch print line on 719, 12-inch maximum on 720 and 730)
Vertical line spacing	6 per inch
Tape-controlled carriage	Yes; three form spacing options: (1) Single space (6 lines/inch) (2) Double space (3 lines/inch) (3) Tape control
Type of paper feed	Continuous form, standard marginal punching pin-feed
Maximum form dimensions	Width - 16 3/4" overall, including marginal strip Length- 22 inches maximum
Minimum form dimensions	Width - not stated Length- no minimum
Paper skipping or fast-feed	Yes, but may cause drop in printing speed
Number of carbon copies	Original plus seven
Storage of information to be printed	Stored in Type 760 buffer
Plugboard editing features	None; no plugboard
Character suppression	Print suppression levers permit suppressing specific groups of print positions
Checking of printer accuracy	Checked for accuracy of character code transfers and character printed

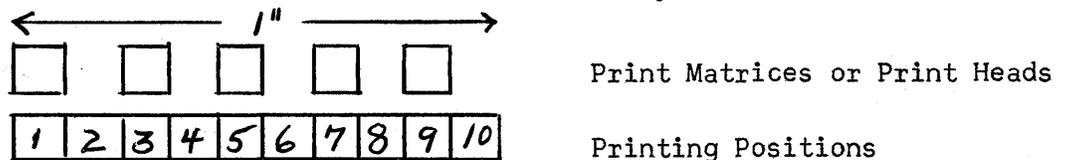
The discussion which follows requires an understanding of the method of using the storage buffer in the Type 760 Control and Storage Unit; this is covered in 1150.

4131.2 Method of Printing. Printing is accomplished by creating each character pattern from a print matrix or printing head of 35 wires, arranged in a 5x7 grid. Individual characters are formed by extending a matrix pattern of wires which differs for each of the characters permissible. This illustration is a head-on view of a print matrix, with lines connected the wires extended to form the letter "H:"



Mechanically, the formation of a character is accomplished by running the matrix wires back into rotating "code rods;" these have grooves and lands which cause some wires to project and others to remain retracted. These code rods are rotated and stopped at the angular position that has the groove and land combination creating a desired character. The actual mechanism, needless to say, is a good deal more complicated than this simple explanation.

On the Type 719, there are 30 of these print matrices, placed in alternate character positions; that is, spaced five to the inch. In one print sub-cycle, the odd-numbered characters are formed and printed; for the next



sub-cycle, the entire assembly of 30 matrices shifts one position to the right and the even-number characters are formed and printed. This completes a print line. On the next line, the even-numbered characters are printed first, the matrix assembly is shifted one position left, and the odd-numbered characters are printed. Printing on any line begins in whatever position the print heads were last stopped. The mechanics of this printing method are "built in," and require no attention by the user.

The operation of the Type 730, which also prints at a rate of 1,000 lines a minute, is identical, except there are 60 print matrices alternated over the 120 character printing positions.

The Type 720, which prints at a rate of 500 lines a minute in a 120 character line, has only 30 matrices, spaced every fourth printing posi-

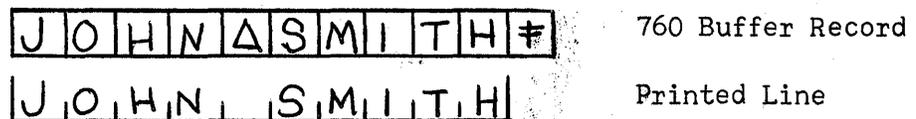
4131.2 (Continued)

tion. The printing of each line requires a total of four sub-cycles and three shifts of the matrix assembly. The sub-cycle time is the same as that of the other two printers; because the 720 takes twice as many sub-cycles for each line, its speed is only half as great--500 instead of 1,000 lines a minute.

The sub-cycle time is 1/2000 of a minute, or 30 ms.

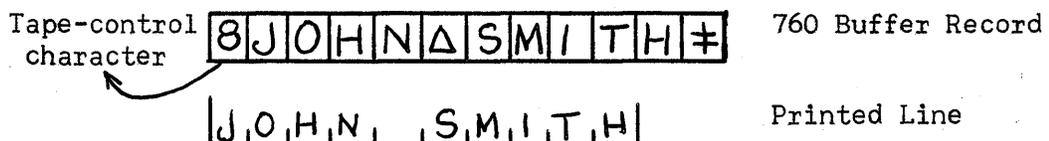
4131.3 Format of EDPM Output and of Printed Line. All three HSPs print a line which is an exact image, character for character, of the data contained in the 760 buffer, which in turn is an exact image of a portion of memory (whether or not tape storage has intervened). An obvious result is that all editing of information into the printing format desired must be done as a part of internal EDPM processing--insertion of blank spaces, commas, decimal points, dollar signs, etc.

The line record in the buffer consists of one character for each print position involved, with a record mark following the last significant character (a control mark for the last record in the buffer). This is printed exactly as it appears, beginning with the first print matrix position; the transfer or formation of characters in the print matrices is terminated by the record (control) mark, which stops or terminates the line, but is not itself printed:



A maximum of 60 or 120 characters, depending upon the version of the HSP, can be printed; buffer records then are normally limited to this number of characters plus the record mark as a maximum.

If tape-controlled carriage printing is desired (as is customary for form spacing), the first character position of the record is used solely for this purpose. IBM does not assign special characters for the purpose of fast feeding; consequently, every record will be handled with the first character as a tape-control symbol when this type of printing is involved. The tape-control character does not count as one of the 60 or 120 maximum printing characters; therefore a maximum buffer print line consists of either 62 or 122 characters (60 or 120 to be printed, 1 for carriage control and the record mark). In this type of printing, the second character of the buffer record is printed from the first print matrix, and so on:



The type of line-spacing is determined by the setting of a three-position

## 4131.3 (Continued)

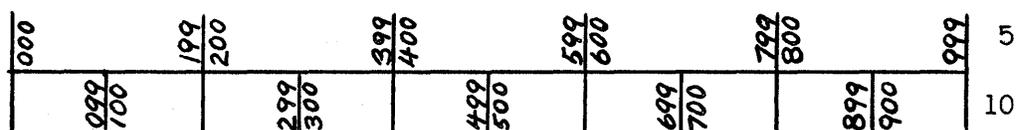
switch located on the printer; these are the options available:

- Setting 1: The carriage single-spaces for each line of printing. Overflow ejection only can be handled by the control tape. The first character of the record is printed.
- Setting 2: The carriage double-spaces for each line of printing. The other two remarks of Setting 1 apply.
- Setting : Tape-controlled carriage operation. The first character of each record, inserted as a part of the EDPM program, is used as the carriage-control symbol and is not printed. There is no automatic spacing--every record must contain such a symbol.

This switch, which is manually set for each printing operation, is called the Program Switch.

Although, in general, every character in the buffer record is printed in the same sequence, except for the carriage-control symbol and the terminating record mark, there are two exceptions to this rule.

- 4131.31 Print Suppression Levers. The printer has eight manually-set Print Suppression Levers which permit locking out groups of printing positions; their function appears to be identical to that of the short hammerlocks in conventional IBM tabulators (402, 405, and 407). Printing never occurs in positions so locked out, regardless of the contents of the corresponding character positions in the buffer record. Available data does not indicate how many printing positions are controlled by each lever, or whether they are flexible as to number suppressed. Additional information is solicited.
- 4131.32 Overflow Characters on Type 719. Although the Type 719 has a maximum of 60 printing positions, the buffer record can be longer; only the first 60, exclusive of the carriage-control symbol, print. This is unlike the 720 and 730, in which a machine error is indicated if the print record is longer than 122 characters.
- 4131.4 Print Record Grouping in the Type 760 Storage Device. Particularly when preparing tapes for off-line printing operations, it is desirable to group several print records into one longer tape record in order to minimize total EDPM time devoted to this operation. To handle such grouped records in printing, the 1,000-character core storage device of the 760 control unit is divided into sections, with the data for each print line placed wholly within one section. The buffer is divisible into five sections of 200 characters each, or ten of 100, as follows:



With a Type 719 attached to the 760, the division into ten sections is automatic. With the Types 720 and 730, division into either five or ten sections is accomplished by a two-position Record Grouping Switch which

4131.4 (Continued)

is located on the printer itself. It is manually set to 5 or 10 for each printing operation, depending upon the number of characters in the printed line and the number of records grouped together on tape or in memory.

In transferring data from memory to tape into the buffer for printing, the first character of a record is placed in the first position of a section, and succeeding characters moved into successively higher buffer positions. The record mark (or control mark of the last record) terminating each record is also stored, but sensing it causes the next character to be stored in the first position of the next higher buffer section. When loaded for printing, the buffer is therefore not packed solidly with meaningful information, but only in sections.

In off-line printing operations, loading of the 760 buffer in this manner is automatic; the inter-record gap on tape causes a control mark to be placed after the final information character of the last record. In on-line operations under EDPM control, recognition of the digit "4" in the units position of the instruction selecting the printer causes the loading from memory to take place in sections. (With a tape unit selected, loading is solid from 000, even though a printer is attached to the 760).

Although sections consist of either 100 or 200 character positions, error conditions or garbled printing result if certain maxima are exceeded. The maximum number of characters to be printed which can be placed in a section is:

	<u>Without Carriage Control Symbol</u>	<u>With Carriage Control Symbol</u>
Ten Sections	98	97
Five Sections	120	121

The record or control mark is in addition to the above. Exceeding the maximum of 99 or 122 characters (which includes the record or control mark) causes an error condition in most cases. There is an exception in the 719, which can print the first 60 characters of a longer record and skip to the next section, omitting any characters after the 60th and not resulting in an error.

The sectional loading of the buffer necessitates the "PTW 0009" instruction used when a grouped record to be both printed and written on tape occurs in an on-line operation (1152.26). This instruction causes the tape writing operation to skip the "garbage" characters following the record mark within each section, and to advance to the first character of the next section. The tape record thus is limited only to printed data and is identical with a tape which would be used for an off-line printing operation.

4131.5 Calculating Operating Speeds of the Printers. In practice, the rated speeds of all three printers are reduced (or at least may be) because of two factors: (1) The length of skips in printed output; and (2) the number of characters being loaded into the 760 buffer. Both of these serve to decrease the effective number of lines per minute.

- 4131.51 Calculating the Effect of Form Skipping on Printing Speeds. Normal single- and double-spacing of forms, either automatic or under programmed tape control, is accomplished without affecting printing speeds. Each print line requires two (719 and 730) or four (720) sub-cycles. Each skip of 3 2/3 inches or less requires an additional two sub-cycles. For skips of greater than 3 2/3 inches, each additional two inches requires one more sub-cycle. For example, a 5-inch skip takes three sub-cycles.
- 4131.52 Calculating the Effect of Buffer Loading on Printing Speeds. The time required to load the 760 buffer with the data to be printed varies with the number of characters; if it cannot be loaded within the printing sub-cycle, a delay in speed results. Buffer loading time also varies with the source of the data; information in memory is transferred at a rate of 0.023 ms per character, while that from tape requires 0.067 ms per character plus a constant 10 ms tape start time. The table below gives the effective printing speeds for grouped records of varying lengths and numbers of records (print lines) per group.

		TYPE 719			TYPE 720				TYPE 730				
Char. per Group	From CPU	0-600			0-600	601-1000			0-600	601-1000			
	From Tape	0-120	121-320	321-600	0-120	121-320	321-767	768-1000	0-120	121-320	321-767	768-1000	
Records/Group		LPM	LPM	LPM	LPM	LPM	LPM	LPM	LPM	LPM	LPM	LPM	
1		1000			500				1000				
2		1000	800		500	444			1000	800			
3		1000	857		500	461	428		1000	857	750		Note 1
4		1000	889	800	500	471	444		1000	889	800		
5		1000	909	833	500	476	454		1000	909	833		
6		1000	923	857	500	480	461		1000	923	857		Note 2
7		1000	933	875	500	482	467		1000	933	875		
8		1000	942	889	500	485	471	457	1000	942	889	842	
9		1000	948	900	500	487	474	461	1000	948	900	857	
10		1000	952	909	500	488	476	465	1000	952	909	869	

Note 1--Group records either by 5 or 10. Maximum record length 120 or 98 printing characters.

Note 2--Group records by 10 only. Maximum record length 98 printing characters.

To use this table, find the column which contains the number of characters per grouped record, including carriage-control symbols and record marks to obtain the total. Find the row corresponding to the number of individual records per group. Read the speed at the intersection.

In this table, each column to the right of the one containing the rated speed of the printer represents one additional sub-cycle for loading the buffer. For example, to load 1,000 characters requires three sub-cycles; the entry is in the third column to the right.

With records of more than 98 characters, to be printed on the 720 or 730, only the upper half of the table can be used; in this case, the maximum grouping is five.

4131.53 Sample Calculation of Effective Printing Speed. An example may serve to illustrate the method of calculating net printing speed, taking into account both form skipping and 760 buffer loading. Suppose one grouped record contains the information to print a 10-line invoice, using the program-controlled carriage. The name and address of the customer are printed at the left and require 40 printing positions; the remaining information needs 90 (both including all blanks necessary to fill out the line to the farthest right character position). The individual records then total 42 and 92 characters, counting carriage-control symbol and record marks, or a total of 720 characters. The printing is to be done on the Type 730 and there are two form skips of 1" and 4½" each, plus a page ejection of 4" to the first print line of the next form. The calculation of sub-cycles necessary then becomes:

Load 720 characters from tape to 760 buffer. . . . .	2
Print ten lines at 2 sub-cycles each . . . . .	20
One skip of less than 3 2/3 inches . . . . .	2
One skip of 4 inches . . . . .	3
Page ejection (skip) of 4 inches . . . . .	<u>3</u>
Number of sub-cycles per invoice . . . . .	30

With 2,000 sub-cycles per minute and ten lines printed in 30 sub-cycles, the effective speed then is

$$\frac{2000}{30} \times 10 = 667 \text{ lines per minute effective rate}$$

The printing speed for any type of printed output can be calculated in a similar fashion. Where skipping or page ejection is not of particular significance--as in a straight single-space listing--the chart in 4131.52 may be used. It gives a speed slightly higher than that which can be attained in practice, because any type of form or page ejection other than double-spacing has an effect--however slight--on net speed.

4132 COMMENTS ON THE USE OF IBM HIGH-SPEED PRINTERS

An evaluation of the performance effectiveness of the Types 719, 720 and 730 requires recognition and analysis of only a relatively few basic factors. Probably the more important are:

1. Lack of a plugboard, which requires that each print line be completely edited as a part of internal EDPM processing.
2. The reduction in printing speed which occurs as the grouped record loaded into the 760 buffer increases in length.
3. The reduction in printing speed because of form spacing.
4. The increase in printing speed as the number of characters to be printed decreases; the EDPM output rate is also increased.
5. Evaluation of the economy of using two tape units, rather than one, in off-line operations, to permit printing to continue from one tape unit while the other is rewinding and being changed.
6. Development of an approximation of the relative effective speeds of the printers over a number of printing operations, and of the total cost per thousand or million lines of the entire printing and print-related processes.

Each of these will be discussed, together with comparative evaluations against the Univac II-Univac High Speed Printer.

- 4132.1 Effect of Lack of a Plugboard. In an EDPM operation, the cost of printing cannot be confined to the X dollars a month that the printer costs; it must include the cost of preparing the information into a form that it can accept. Because the IBM printers do not have plugboards, but print information, character for character, exactly as it is received in the printer memory, the internal cost of assembling data into that format is an integral part of the overall printing cost.

The IBM claim is that the internal speed of the EDPM is so great that internal editing is faster and cheaper than plugboard editing; as a matter of note, this claim was back when the 702 was first announced, and it is slower internally than the 705. The instruction repertoire of both equipments includes commands specifically designed or useful in output data editing and assembly: "Store for Print," "Normalize and Transfer," and the high-speed internal memory transfer. These, together with 17 microsecond cycle rate of the 705, are cited as reasons for not needing a plugboard on the IBM printers.

Now an exhaustive discussion of all the possible ways in which an output line for printing may be generated is virtually impossible. Some of the typical possibilities, together with their effect on such factors as input data and master file format, are discussed in 8120. However, some points of common applicability can be noted. They are, all too frequently, not taken into account.

4132.11 Internal Assembly of Data to be Printed. Typically, printing consists of data elements assembled from two or more files, plus possibly some computed results, by means of internal programming. In general, the format used for magnetic tape storage or EDPM processing differs from the one of the printed page. Separation of entries by blank spaces, insertion of punctuation marks, decimal points, commas, dollar signs, etc., make it look different, even if the basic information content is the same. This has a definite effect on the internal programming of the 702/705.

First of all, the high-speed memory transfer can seldom be used. There are three reasons for this: (1) Characters moved by this means are not rearranged in any manner; (2) it can start only in every fifth character position; and (3) it is stopped only by a record mark, also in a specific position. This means most transfers into the output assembly area will be on a character-by-character basis, which takes 5 to 20 times longer.

Second, although the "Store for Print" instruction is the fastest way in the 702/705 to change non-significant zeroes to blanks and to insert decimal points and commas, it is not particularly fast in actual use. Each field must be initialized to the pattern every time it is used, and then the new output quantity moved into it. The normal minimum computing time--assuming an Auxiliary Storage Unit set to the proper field length--involves at least three separate character cycles for each one moved, and may run into more. For example, a 6-digit field-length dollar amount in the main accumulator (as the result of a computation) is to be placed into the printing area with the decimal point and comma (if necessary) inserted. This requires a minimum of 0.372 ms, or 0.062 ms per character, and it can well take more.

Third, some (or many) numeric quantities used in calculating operations may have the units position "signed" in conventional manner, are not even in condition to use the high-speed memory transfer; they must be transferred via an accumulator to detach the sign. If this were the case in the previous example, the time would increase to 0.454 ms, or 0.075 ms per character.

The wide variations in printing formats make it impossible to set down any close estimate of the internal time required to assemble a complete line in the 702/705. (The 702 takes longer, because it's slower). Assuming data elements averaging six characters--the length of the typical business item--it appears that anything from a minimum of around 0.03 ms up to over 0.1 ms per character may be expected. An overall average of around 0.06 - 0.07 ms appears reasonable. For a full line of around 100 significant printing characters, an expectancy of 6-7 ms for output assembly appears to be, if anything, somewhat optimistic; it may run a good bit more and usually will not be much less. Shorter lines, of course, would take less overall time.

How does this compare with Univac II? Well, ten one-word transfer take 2.4 ms; if extraction is required for some data elements, it might go up to 3-4 ms. But it can also be a good deal less. Because the plugboard on the printer permits complete rearrangement of data as a part of printing, the design of file formats can often take account of data elements needed in frequent or high-volume printing operations and group them into successive words. Many printed outputs of this nature can be assembled with two or three V-W transfers; time, about 1.25 ms.

## 4132.11 (Continued)

At this point, it may be apropos to repeat again a highly significant fact: How fast a computer may be in its ability to execute a specific instruction is not nearly so important as how fast it can do an assigned job. The 705 can manage an operation on one character at a pretty fair rate of speed, but by the time it finishes doing something useful it often has spent much more time than the "slower" Univac II. And note also than when Univac does something, it is done correctly-- thanks to its built-in checking.

The internal logic and printing flexibility of Univac II permits it to assemble data to be printed at a rate which will average three to six times as fast as the 705.

- 4132.12 Assembling Different Kinds of Printed Lines. Many business printing operations require a number of different line formats for various types of information; an invoice or bill, for example, may have from two or three to more than a dozen different kinds of line arrangements. It is necessary for the EDPM program to be certain that each line is assembled in memory to print properly.

The 702/705 are equipped with a special write instruction which is intended to facilitate this operation. "Write and Erase" not only transfers information in memory in the same manner as the normal "write" instruction, but also replaces each character by a blank, thus "initializing" the area for the next output line. Everything is erased except the group mark in the 705, which means that record marks separating individual lines of a grouped output must be reinserted by programming. It is also necessary to "initialize" numeric fields which are to be printed with decimal points and commas.

Although the "write and erase" instruction provides the fastest 702/705 means of clearing an output area to facilitate setting up the next series of lines, it has the rather evident disadvantage of making correction of "write errors" more complicated; the output must be completely reassembled from the beginning.

With Univac II, of course, the two most frequently occurring line formats are covered by the multi-line feature of the plugboard, which always provides at least two arrangements of the printed line independent of the physical sequence in the output blockette. This feature eliminates a good deal of housekeeping time in the EDPM. Again, the "high speed" 705 doesn't get a job done as fast as the "slower" Univac II.

- 4132.13 High-Frequency Printing. Certain business files must be printed, in large part or completely, almost as often as they are processed. Here the format for printing is often of overriding importance if excessive EDPM time devoted to turning out a special printing tape is to be avoided. Probably the classic example is a magazine subscription file from which address labels must be prepared. Usually, not quite all of the file must be printed--some coded information not included in the labels is present in the master file. The plugboard editing and multi-line features of the Univac high-speed printer make it possible to print directly from the

4132.13 (Continued)

master file; this is much more difficult (and usually impossible) with the IBM HSPs, which have no editing ability.

Another known example is a technical file used in military supply operations. This file is updated on a quarterly cycle for inventory purposes and printed at the same frequency as a supply reference catalog. The result is that the file itself--processed on an IBM 702--is set up in final printing format--and consists of about 50% blanks, because of the frequency of lines having a maximum of 32 printed characters. With Univac's multi-line printing feature, these secondary lines would be packed three or four to the blockette, giving a tremendous speed-up to the file maintenance process.

Univac's flexibility makes it highly probable that it will process files of this nature much faster than the 705.

4132.14 Duplication of Printing. In some applications, it is desirable to repeat all or part of the information on one line in two different sections of a printed form; repetition of billed amounts in statements with a detachable "stub" are typical. The "split wiring" feature of the Univac HSP permits this to be done without repeating data in the EDPM output record. In fact, many such printings can be multi-lined as well as split-wired. This is not possible with the IBM HSPs; the same data must be placed into two positions of the output record. More internal programming and more tape writing time are the result.

4132.2 Reduction in Printing Speeds as Grouped Records Grow Longer. The table in 4131.52 shows the printing speeds possible with grouped records of different sizes and number of lines included. Clearly, as tape records get longer, printing speed drops. And as printing speed goes up through the use of shorter records, EDPM tape writing time also increases. Because the EDPM is several times as expensive as the printer, it almost always is desirable to minimize 705 tape writing time and accept a reduction in HSP speed. The following table shows the time, in minutes, to write tapes for 100,000 printed lines for ranges of 40 to 120 printed positions (including all blanks) and for one, five and ten records grouped together on tape:

<u>Number of Print Positions</u>	<u>Number of Records per Group</u>		
	<u>One</u>	<u>Five</u>	<u>Ten</u>
40	21.7 min.	8.0 min.	6.3 min.
50	22.8	9.1	7.4
60	23.9	10.2	8.5
70	25.0	11.3	9.6
80	26.1	12.5	10.8
90	27.3	13.6	11.9
98	28.2	14.5	12.8
100	28.4	14.7	----
110	29.5	15.8	----
120	30.5	16.9	----

The savings in main frame time by the largest possible grouping of print records is evident.

4132.3 Reduction in Printing Speed Because of Form Spacing. The reduction of printing speed because of "fast-feeding" varies considerably. It is almost nil when single-space printing on a 22" deep form with 1" top and bottom margins--the page ejection takes two sub-cycles compared with 240 printing sub-cycles (at 1,000 lines a minute), or less than  $\frac{1}{120}$ %. The other extreme may require almost as many fast-feed sub-cycles as does the actual printing; one-line checks or earnings statements could be examples. Probably from 10-30% is an average range, the former applying to more or less tabular listings and the latter to preprinted forms for operations similar to billing.

Although IBM manuals do not state the paper speed during fast feeding, it can be calculated as about 67 inches/second (two inches in one sub-cycle of 30 ms). This compares with 20 inches/second on the Univac HSP, apparently giving a significant advantage to the IBM equipments.

In evaluating the importance of this fast feed skipping speed, however, the bare figures cited are subject to important qualifications:

1. Univac suspends printing only for the actual skip time, while the IBM printers lose at least two printing sub-cycles, 60 ms, for every skip, and then increase in increments of a sub-cycle, 30 ms, for longer skips. On anything up to a seven-line skip, Univac loses less printing time.
2. Univac makes a slightly longer initial skip with no loss in print speed.

In general, the faster feeding over short gaps more than offsets Univac's longer time on large gaps; in "average" business printing jobs there are more of the former. In the IBM printers, the percentage loss is greater on the 1,000-line versions.

4132.4 Effect of Line-Length on Printing Speed. The shorter the length of the printed line, the greater the printing speed on grouped records, as is shown in the table of 4131.52. Because the record mark terminates the data to be printed in each line, it is necessary to assemble characters only for the positions actually to be printed. For example, if only the first 40 printing positions are required, the records are set up in that length, and only that many characters written out on tape. Thus internal computing and tape writing time is conserved, and the printer operates at a fairly high effective rate.

Superficially, this appears to be a definite advantage over Univac, which always prints from full 120-character blockettes. In practice, however, anything involving up to about 70-75 print positions has 60 or fewer printing characters--the rest being blank to separate columns--and Univac would multiline these on the HSP. The basic region in which the IBM computer time is significantly less is in printings ranging from about 61 up to 85 or 90 significant characters.

Where lines are of varying lengths within one printing format, programs for the 705 probably would be based upon the maximum length, although they could be handled as variable. The internal programming necessary to as-

## 4132.4 (Continued)

semble variable length printing records and insert a terminating record mark normally becomes excessive, unless they are written out individually, and this more than doubles tape writing time.

It should be noted that the advantages of short line lengths can accrue to the IBM printers only if all lines begin at the left-side printing position. The setting up of printing characters can be stopped at any position, but every line begins at the first print matrix. For example, name and address lines on an invoice in a position 20 character positions to the right of the detail line items requires 44 printing positions rather than the 24 actually involved as significant.

4132.5 Economy of Using Two Tape Units. The Type 760 is equipped to handle two tape units in off-line printing operations, for the avowed purpose of enabling continuous printing from one tape while the other is rewinding and being changed. This means an extra rental of \$550 for the second tape. The economy of this deserves investigation.

A tape reel is printed in the shortest possible time if the printer can operate at full rated speed and print the fewest possible number of lines from the reel. This condition occurs with tape records of 120 characters, of which about 21,300 can be contained on one full reel of tape. Printing time on the Type 730 is 21.3 minutes; rewind and tape changing about 3.2 minutes, allowing two minutes for the change (although IBM states in its literature that it takes  $1\frac{1}{2}$ ). Two tape units permit printing during this period, giving a 15% increase in output at a 12.3% increase in rental (\$4,450 vs. \$5,000). There is thus a marginal economic advantage in printing costs in this case--and it is the most favorable to two-tape operation. To achieve it adds roughly three minutes to 705 tape-writing time as compared with grouping records in fives. The net result is that the saving of 3.2 minutes on the \$4,450 or \$5,000 a month printer takes almost that much time on the \$25,000-odd a month 705. The economy is hardly obvious.

In most practical cases, of course, the net gain in printer output will be less than the percentage increase in cost for the second tape unit. It is easy to calculate the minimum printing time, omitting skips, for one reel of tape and determine the percentage increase in output if two tapes are used. In virtually no practical case will it be as much as 10% more, compared with 12.3% additional for the second tape. About the only time it can be justified economically is the relatively rare and practically unpredictable case in which an extra 20-60 minutes per day is the difference between one-shift and overtime printer operation. It is doubted if anyone has achieved a degree of operating stability (not to mention printer reliability) reflecting such a condition. Likewise--have you ever met an IBM supervisor who would schedule an equipment to almost 100% of rated speed?

The other instance in which two tape units may be justified is in the printing of a number of different fairly small reports. The practical objection is that very few users have such a condition. In any event, form changing is often necessary, and this prevents continuous printer operation.

4133 COMPARISON OF IBM TYPE 719 AND UNIVAC HIGH SPEED PRINTER

The 1,000 line a minute, 60-character Type 719 appears to be essentially a special-purpose machine intended for large-volume printing jobs involving short lines; public utility and insurance premium billing are possible uses. If the form design is of necessity such that two-up printing cannot be done, the \$700 a month saving over the Type 730 is attractive. It does not appear that form restrictions will be this rigid in very many applications. When two-up printing is possible, the Type 730 gives much higher production per dollar.

It is not considered that the 719 is particularly attractive except in highly specialized cases. Its limited line length makes its use for printing management reports, accounting summaries, inventory analyses, and similar outputs somewhat impracticable.

The Univac HSP rebuttal to the 719 is two-up printing, resulting in more net output at lower cost and time. Information is requested on proposals of 705 installations involving this printer.

4134 COMPARISON OF UNIVAC HSP AND IBM TYPES 720 and 730

The comparative performances of the Univac HSP and IBM's two 120-character printers requires summarization of internal EDPM costs as well as those directly attributable to the printer.

- 4134.1 Internal Data Editing. In general, Univac's plugboard-controlled features allow internal data editing to be accomplished in from one-third to one-sixth the time of the 705. As a corollary to this, the amount of memory necessary for output assembly instructions is also much less. The amount of time saved per print line is extremely variable, but an across the board reduction of 4-10 ms, with an average of 6-7 ms, appears reasonable.
- 4134.2 Tape Writing Time. The time to prepare magnetic tapes for printing on the Univac HSP is a constant 101 ms per block, as blockette output; this results in anything from 6 to 36 printed lines. The 705 time is highly variable, depending upon both line length and method of grouping. In general, with grouped records, tape writing time is less than Univac II, particularly for outputs not susceptible to multi-lining. The increased tape time is about equal to or somewhat less than the savings, in Univac, in output assembly time. For example, the 705 can write 120-character output lines in groups of five at a rate of six lines in 60 ms, compared with Univac's 101 ms. However, it will take about 35-40 ms longer in assembling these lines and the time to load the output buffer is also longer than in Univac. The result is an approximate stand-off in terms of computer time, which may give Univac a slight cost advantage (because an average Univac II CPU with tapes costs less than a comparable 705).
- 4134.3 Comparative Per-Line Printing Costs. Experience has indicated that the Univac HSP is capable of printing at sustained rates of 375-400 lines per minute, counting tape rewinds and changes, form spacing and downtime. There is as yet no comparable experience for even the 500-line IBM HSPs. For the most efficient 705 usage, grouped records for printing are mandatory and this factor reduces the actual maximum print rate for the 1,000 line versions to 833, 869 or 909 lines/minute, with the former two somewhat more probable. Form spacing will cause a further reduction of from 10-30% in many applications. Finally, unscheduled machine downtime will reduce output. (And a ribbon change every two hours, at three to five minutes, should also be considered). A range of from about 600-825 net lines per minute, exclusive of downtime, may be anticipated with the Type 730, and about 400-460 with the 720.

On a cost-per-line basis, the 730 should turn out about 30% more printing at 50% greater cost (two tape units), or 25% more at 33% greater cost (one tape unit) than the Univac HSP. The 720 may be expected to average from 75-95% of Univac HSP's print rate at 13% more cost, assuming one tape unit. It is never economical to hook two tape units on the IBM printers (the maximum increase in output is under 10%, and the extra tape adds 12.3% to the printer rental).

- 4134.4 Normal Printing Costs. Per-line printing costs become pertinent only when the volume approaches one-shift capacity, or is so great that several printers are necessary. Assuming a Univac HSP rate (net) of 375 lines

## 4134.4 (Continued)

per minute, its capacity is 180,000 per shift. For any installation having a total printing load not exceeding this volume, the Univac HSP is always more economical, simply because its base rental is less than either IBM version. One shift on the 730 may produce anything from 250,000 to 300,000 lines, and single-shift operation with this printer may be somewhat cheaper than a two-shift Univac HSP when volume ranges between 180,000 and the Type 730 limits. Above 300,000 lines, two shifts are necessary for both types, and again the Univac HSP is more economical.

Large-volume printing operations tend to be somewhat specialized; magazine address label preparation, public utility and insurance premium billing, etc., are common examples. Probably 80% or more of EDPM applications are for users with under a million lines a week; in all of these, Univac II, with its HSP, is almost always going to handle printing requirements at a smaller cost for the printer and about the same or somewhat less central processor cost.

4134.5 Printer Reliability. Univac's HSP has been proved out in many installations since June, 1954. IBM has not yet had completely satisfactory results with its 500-line Type 720, of which a very few are installed, and there are no known production installations of the 1,000-line version. Although the only safe assumption, for customer relations purposes, is that IBM ultimately will get them to work, it should be noted that they are mechanically much more complex than ours and may be a long time reaching the high degree of reliability which the Univac HSP has achieved.

4134.6 Number of Carbons. IBM claims more copies of legible carbons than does Univac--seven against four. Possibly the IBM version does give a higher maximum. In the final analysis, the adequacy of carbon impressions is determined by the user and the purposes to which the carbons are put. Also, the nature of the printing is important; alphanumeric copy may be legible by context on copies where numbers could not be deciphered. Because the customer is the final judge, the best answer is the one which has been rather consistently followed by Univac: If question arises as to the legibility and number of carbons, have the potential customer send in 50 or so sets of the form involved and run them off. This is quick, cheap -- and definitive.

4134.7 Appearance of Characters. Because IBM HSP characters are formed from a 5x7 grid of wires, characters are not quite so nicely shaped as Univac's. Actually, the first carbon impression from matrix printing often looks nicer than the original, because of the tendency of the carbon to cause the individual dots of the original to blend or merge together. Generally speaking, matrix printing is acceptable in quality, assuming the printer is working reliably. It takes only a few malfunctioning wires to cause many characters to be incompletely formed; this may cause misinterpretation of what the character is, as well as reducing the printing quality. With either 1,050 or 2,100 of these wires in a machine, there are numerous chances for a few not to be functioning correctly. A slight malfunctioning in Univac causes, at most, a somewhat uneven alignment of printing; almost anything else stops the HSP. Univac HSP pre-formed characters always look nicer, are never a disadvantage as a sales point, and may be a definite feature when quality of printing is desired.

4139 SALES POINTS ON IBM HIGH-SPEED PRINTERS

Unless noted, these points apply to all three versions.

- 4139.01 Availability of Printers. As of this date, a few 500-line Type 720's are in 702/705 installations, but some (if not all) are backed up by 150-line Type 717's on a no-rent basis because of the generally poor performance of the HSPs. The 1,000-line versions reportedly have been returned to the laboratories for clearing up "bugs." Although it must be assumed that IBM will get them to work, they are not yet satisfactory.

Against the IBM Printers. They are not available. About the time IBM delivered its first 705, its sales staff began asking customers:

"Have you ever seen a Univac II?" How about returning the query:  
"Have you ever seen an IBM high-speed printer?"

- 4139.02 Printer Reliability. The Univac HSP is proved out in operation; IBM is talking futures.

Against the IBM Printers.

- (1) Mechanically, they are much more complex than the univac HSP. This indicates a continuing need for more maintenance and a larger amount of machine downtime during production runs. It appears doubtful that they will ever give as high a percentage of good production time.
- (2) Malfunctioning can cause partially-formed letters, missing one or two dots each. This lessens the attractiveness of printing.

- 4139.03 "Standby Printers." IBM may point out to prospects who are concerned about availability of its HSPs that Type 717s have been made available, at no cost to the customer, to handle printing if the HSPs are not functioning properly.

For IBM. Well, at least the customer gets something; he is not stranded with only a 10-character a second typewriter.

Against the IBM Printers. Type 717s are no substitute for the HSPs for two reasons:

- (1) They are slower. It would take four of them, at least, to equal the effective output of one 1,000-line printer.
- (2) They accept single-line tape records only, not grouped records. A program written for an HSP tape output would require revision if a Type 717 were used. More important, CPU time would increase by 100-250% for the tape writing.

- 4139.04 Plugboard vs. No Plugboard. Univac HSP has it; IBM does not.

For the IBM Printers.

- (1) It is not necessary to invest in a number of plugboards.
- (2) Short printing runs can be edited by internal programming and printed without taking the time to place a plugboard in the machine and test it. Several such reports can be run off successively without changing the machine setup.

4139.04 (Continued)Against the IBM Printers.

- (1) All printing, long as well as short jobs, requires complete internal editing. The cost of the memory required for editing instructions, plus the additional CPU time to do it, far outweighs the one-time cost of plugboards.
- (2) It is not the short printing runs that give volume; if a prospective customer has enough work to require an HSP instead of a 150-line printer, he has at least one high-volume job. The computer savings in this one run more than offsets any additional time which might be required to insert a new plugboard.
- (3) "The 705 is so fast internally that programmed editing is more efficient than plugboard." See 4132.1 for an analysis of this statement.
- (4) Multi-line printing cannot be done. Univac has this feature, saving EDPM tape writing time when it can be used.
- (5) Duplication of characters requires internal assembly of one data element into two or more memory locations. Univac does it in one step, and can combine duplication with multi-lining.

4139.05 Internal Assembly and Tape Writing Time. For short printed lines, the 705 can write out tape records faster than Univac, because it uses only the amount or number of characters involved, whereas Univac always writes in blockettes.

For the IBM Printers. This premise may be true in some instances.

Against the IBM Printers. For short-line printings, Univac output will provide for multilining. On the average, central processor plus tape writing time in Univac will not exceed that of the 705, and may be considerably less.

4139.06 Printer Speed. The IBM 1,000-line Type 730, at \$4,450, is 67% faster than the Univac HSP and costs only 35% more.

For the IBM Printers. Based solely on rated speeds and costs, this statement is true.

Against the IBM Printers. The statement is true only in a meaningless sense.

- (1) For printing volumes of up to about 180,000-200,000 lines per day, the Univac HSP is always cheaper, simply because it rents for less than any of the three IBM printers.
- (2) Effective speeds over a long period, not rated speeds, are the important factor. No IBM printer will operate at rated speed--even on a straight listing--without a murderously slow 705 operation. (See 4131.52). Univac HSP will.
- (3) The 500-line Type 720 always costs more than Univac HSP.
- (4) Univac HSP has been proved out at about 400 lines/minute over long print runs. What the Type 730 will net is sheer guesswork.

4139.07 Two Tape Units. The IBM printers can use two tape units, permitting their alternating back and forth to eliminate holding up printing during rewind and tape changing; Univac HSP has only one.

For the IBM Printers.

- (1) The possibility exists.
- (2) It may not be necessary to acquire the second tape unit specifically for the printer; a main frame unit can be borrowed when not needed in the machine run being processed.

Against the IBM Printers.

- (1) The extra tape unit adds 12.3% to a Type 730 and 14.3% to a Type 720 rental; to be economically justified, printed output must increase by at least this percentage. For long, multi-reel printing operations, it almost certainly will increase by less (4132.5). There may be some merit to the idea if numerous short runs exist and can be printed successively; this seldom occurs.
- (2) Borrowing a unit from the 705 may not be so easy as it sounds. It adds to scheduling difficulties--and the printer may be in a different room from the CPU.
- (3) The possibility is attractive principally when the printing volume is so close to a full shift that gaining 20-60 minutes extra available time might avoid overtime or additional shift rental. Unpredictable downtime alone--particularly on an untried printer--precludes scheduling to this degree of closeness.
- (4) The argument that the extra tape unit permits printing to be completed more quickly is rather specious. On full reels of tape, the "time saving" will run 6-8 minutes per hour; for a number of short runs, the percentage is better but the absolute time savings are still trivial.

4139.08 Direct Connection of Printer to Central Processor. The IBM printers, through a Type 760 Control Unit, can operate with the CPU under program control, as well as off-line. Univac HSP has only off-line operation.

For the IBM Printers. This type of operation is possible.

Against the IBM Printers.

- (1) Output from memory in this mode of operation takes  $2\frac{1}{2}$  times as long as to magnetic tape--0.023 ms per character compared to 0.009 ms. This increases the amount of time the CPU is tied up.
- (2) Almost perfect balance between computing and printing times is required for efficient use of both EDPM and printer; this practically never exists. Either the EDPM or the printer is standing idle part of the time.
- (3) Machine failure in either printer or CPU stops both. With a mechanically complex printer, why chance hinging a \$25,000 a month EDPM to the performance of a \$4,450 a month auxiliary?
- (4) Why pay for capabilities which are a convenience in program debugging, but not economic for productive work?

4139.09 Use of Type 720. The 500-line version seems to be the first which will give fairly adequate performance; the others are not yet in a reliable performance state.

Against the 720. No comparison with the Univac HSP. It turns out less work and costs more.

4510 IBM TYPE 721 CARD PUNCH

The IBM Type 721 Card Punch is designed to punch standard 80-column cards in on-line operation only in IBM Type 701, 704 and 709 EDPMs. It is not capable of independent off-line operation (T-C conversion), but operates always strictly under stored-program control. Only one version, designated as Model 1, has been announced.

4510.1 Cost.      Purchase:      \$ 39,000  
                                  Rental:                      600 per month

4511 CHARACTERISTICS OF THE TYPE 721

Type of cards punched:	80-column
Number of columns punched:	72 from EDPM memory; remaining 8 can be punched (see 4512)
Type of card punching:	Conventional punch card code or binary (see 4512)
Card punching speed (continuous):	100 cards/minute
Control panel:	Yes (see 4512)
Number of stations:	Two (punch station and punch read station)
Checking of card punching:	None
Memory in card punch:	No
Method of punching:	From card image in memory; 36 columns of one card row are punched from one word in memory

The Type 721 is, essentially, a slightly modified Type 521 Reproducing punch.

4512 METHOD OF OPERATION OF THE TYPE 721 CARD PUNCH

The method of punching a card is quite similar to that of reading with the Type 711; this is discussed in 3110, and an understanding of its operation is assumed in this section.

In a manner similar to the Type 711, a card is punched in 24 cycles, each punching 36 columns of one card row. Through control panel wiring, the specific card columns corresponding to each bit of a 36-bit internal word can be varied. Punching of a full 72 columns in all 12 card rows requires creation of a 24-word "card image" in memory (by internal programming), a "0" representing no punch and a "1" representing a punch in a specific row and column position. Punching may be either in conventional card codes or in pure binary; in the latter case, one card can contain a maximum of 24 36-bit binary numbers.

Some flexibility and partial checking of the card are possible through control panel wiring. Double-punch and blank column detection can be wired into any or all of the 80 card columns; this permits detecting failure to punch or erroneous double-punching. Four 10-position selectors are provided; these may be actuated in two ways:

- (1) Two sense hubs, which are program-addressable by a "sense" instruction, provide some flexibility in card format. For example, some fields of a detail card may be punched in a manner different from a master card, even though the information occurs in identical bit positions of the words in the "card image."
- (2) Six control brushes, which can be set to any desired card columns at the read station, are provided. These can, if desired, be used to pickup one or more selectors to permit gangpunching of data in the card at the read station into the card at the punch station. Gangpunching may be continuous (independent of the EDPM) or under program control. A punch delay hub is available to permit offset (interspersed) gangpunching from a master card into following detail cards.

Ten column splits are also available. Thus, although only 72 bits per card row are transferred from the EDPM to the card punch, the use of column splits and/or gangpunching permits the full 80 columns of a card to be punched.

In the Types 701 and 704, transfer of the successive words of the "card image" from memory to the card punch is through the MQ-register. In the Type 709, it is through the synchronizing register in one of the two channels of the Type 766 Data Synchronizer.

The only checking of the card is that provided by double punch and blank column detection.

4512.1 Memory to Card Punch Transfers -- Types 701 and 704. The method of operation is almost identical with that of the Type 711 Card Reader, section 3112.1, which should be understood before reading this discussion.

Punching is accomplished by a programmed "write" instruction, with the address of the card punch, followed by one or more "copy" instructions, each of which transfers one word of the card image from memory through the MQ-register to the punch dies. Punching is 9-row first, and the successive copy instructions must follow within specific time limits; the essential parts of the timing chart are:

12 RC

9LC 9RC 8LC 8RC

11 LC 11 RC 12 LC

12 RC

WRITE

25 ms

95 ms

In continuous card punching, 442 ms of the card cycle of 600 ms (at 100 cards/minute) are available for useful calculating. The remaining 158 ms are required for execution of "copy" instructions and for safe synchronizing of the mechanical punching components. The times in the chart above are the maximum safe computing times available between successive "copy" instructions. A brief explanation of the various limits is:

Execution of "Write" Instruction. For continuous card feeding, this must be executed within 25 ms after the last "copy" of the previous card. It cannot be executed until 25 ms has elapsed; this means that for full utilization of the computing time, it must be reached exactly 25 ms after the final "copy;" because programs normally cannot be timed this closely, some of the available 25 ms is usually lost. The EDPM "hangs up" on the "write" instruction, if necessary, until the 25 ms has elapsed.

Time Between Execution of "Write" Instruction and First "Copy." 70 ms of useful computing time is available between the execution of the "write" instruction and the first (9-left) "copy." This is true whether or not continuous card motion is involved. If the card punch is not in motion (the write instruction executed after the 25 ms interval), about 400 ms is required before the 9-row is in position for punching. However, only 70 ms is available for computation in this case also. The differences are necessary to assure proper synchronization of the mechanical card feed with the electronic circuitry.

Time Between Left-and Right-Copies. The two copy instructions for one card row must follow each other by not more than 540 microseconds. In practice, this time interval is usually used to calculate the memory address for the second "copy."

Time Between Successive Card Rows. A maximum of 31 ms of useful computing time is available between the right-copy of one row and the left-copy of the next.

4512.1 (Continued)

Although the transfer of each word from memory to the punch is through the MQ-register, it is used only for the time required to execute the "copy" instructions. It can be used for any computing purposes desired during the available time between successive "copies," but its contents are destroyed during the transfer of the word.

As in the Type 711 card reader, the timing impulses governing the punch originate in it, not the EDPM. If the program reaches a "copy" instruction before the punch is ready, the EDPM simply waits until the impulse actuates the word transfer. Even if the program uses the full available computing time between successive copies, there is some delay in executing them; the time limits indicated allow for safe synchronization of the variances in the card feed mechanism. A "copy" instruction can be reached at any time during this available computing period; the EDPM simply waits until the card punch is ready for the word.

If the full 24 words are not required to transfer essential information to the card (as may be the case in punching binary information, or numbers only in punch card code), the writing can be stopped at any point by merely omitting "copy" instructions. The card movement continues to the end of the cycle, but no additional punching in the card is possible; i.e., it is not possible to skip punching in a couple card rows and then resume in the same card. Gangpunching, controlled by control panel wiring, is not affected by this suspension of information transfers from memory.

Unlike the card reader, an error condition results if more than 24 copy instructions are given without an intervening "write." Thus there is no automatic end-of-record skip and the "copy loop" must provide for making sure that the proper number of words are transferred to the punch.

4512.2 Memory to Card Punch Transfers -- Type 709. The operation is analogous to card reading (3112.2). The transfer of successive words from memory to the Data Synchronizer register is under control of the memory address and word count in the Control Word. Impulses from the punch initiate the transfer of each word out of the synchronizing register and the Data Synchronizer automatically interrupts the main program for the one cycle required to access the next output word from memory.

As is the case with the card reader, the Data Synchronizer Channel cannot be used for any other input or output purposes during the 600 ms required to punch the card. The transfer can, of course, be stopped at any point in the punching cycle by one of the normal methods of terminating an input/output operation in the Data Synchronizer (1160).

Like the card reader, the Type 721 can be connected to a Type 766 Channel only if a Type 716 Tabulator is also attached to that channel.

4513 USE OF TYPE 721 CARD PUNCH IN BUSINESS DATA PROCESSING

The use of this card punch in business data processing applications is subject to the same considerations applicable to the Type 711 Card Reader. Consequently, the discussion of 3113 is fully applicable to this unit. It should be noted, of course, that the conversion of data is the reverse of that encountered with the card reader--from internal language to punch card code. This conversion requires programming which is essentially the reverse of that required with the card reader.

One other point of difference exists. With a punching speed of 100 cards per minute maximum, and 72 information columns per card, the maximum transfer rate of character coded data is 120 per second. If cards are punched with binary words, the maximum rate is 40 36-bit (35 plus sign) words/second, equivalent to about 400-440 decimal digits.

4519 SALES POINTS ON THE TYPE 721 CARD PUNCH

Index numbers 4519.01--4519.49 are reserved for pointers applicable to business operations; 4519-50--4519.99, for those of use primarily in scientific applications.

As might be expected, many of these pointers are corollaries of those developed for the Type 711 Card Reader.

- 4519.01 Lack of Checking. The Type 721 can be wired to do double-punch and blank column detection, but cannot otherwise verify the accuracy of punching.

Against the 721: For any type of punch card output entailing accurate reproduction, as in many accounting applications, any method of checking requires additional work. Possibly the most feasible way is to accumulate proof and "hash" totals as a part of internal programming, and punch out periodic "batch total cards." The batch can then be proof-listed on a tabulator. This method entails additional work--and equipment--and only detects an error in a batch without isolating it to a specific card or column.

- 4519.02 Punching of Alphanumeric Information. The Type 721 requires a "card image" of up to 24 words; this must be created, by programming, from the internal binary or character coded machine language.

For the 721: The punching method permits the easy handling of pure binary information; 24 36-bit words can be punched into one card.

Against the 721: The internal programming time required to convert from internal machine language to punch card code is considerable. Business output data are almost exclusively in punch card codes, not binary.

- 4519.03 Elimination of a T-C Converter if a Type 721 Is Otherwise Required. In proposals in which a Type 721 is necessary to punch out binary scientific data, IBM may propose its use in lieu of a T-C converter (Type 722), simply to save its \$1,600 cost (including tape unit).

For the 721: This is a possible solution, and may result in a saving of about \$1,600 in basic installation rental.

Against the 721: There are two rebuttals:

- (1) With the 709, a Type 716 is required before a Type 721 can be connected to a Data Synchronizer channel. This boosts the cost of the two units to \$1,800 monthly, more than the cost of a T-C converter. The economy of this would hinge on the effective utilization of the printer.
- (2) The proposal is economic only for small punch card output volumes. For any specific proposal, an analysis similar to that developed for the card reader (3113.1) can be developed.

- 4519.04 Mismatch of Card Punching Rate and Computing Speed. The usual remarks on the speed mismatch between the relatively slow card punch and the fast CPU are applicable. Because computation time and card rate seldom are even approximately equal, one or the other is used at less than maximum efficiency.

5120 IBM TYPE 702 ELECTRONIC DATA PROCESSING MACHINE

The IBM Type 702 EDPM was the first data processor for business purposes produced and delivered by the International Business Machines Corporation. The Type 702 specifically pertains only to the central processing unit, other type numbers applying to all input, output and periphery units. In general usage, "702" refers to an EDPM installation with this equipment as the central processor.

First deliveries to customers were made in March 1955 and production was discontinued about one year later. A total of 15 were built and are in use.

5120.1 Monthly Rental: \$8,800 with 10,000 characters electrostatic storage  
9,800 with 20,000 characters magnetic core storage

These rentals are for the central processor, including arithmetic and logical unit, control functions, memory, operator's console and necessary power supplies.

5121 CHARACTERISTICS OF THE TYPE 702

The 702 is an earlier and slower version of an equipment that became the 705. Logically, it is quite similar to the later, and differs in only a few points. Because the 702 is not now a factor in the competitive market, its characteristics are of purely historical interest. For reference purposes, the basic differences between the 702 and 705 are summarized. Refer to 5140 for a description of the 705.

1. Operating speed of the 702 is slower than the 705.
  - a. Basic character cycle time of the 702 is 23 microseconds; of the 705, 17 microseconds.
  - b. The 702 has serial access to instructions, giving a minimum time of 138 microseconds for any instruction (five accesses plus one cycle minimum for execution); the 705 has parallel access to all five characters of an instruction, and thus a minimum instruction time of 34 microseconds (one access and one cycle for execution).
2. The 702 originally had 10,000 character positions of electrostatic storage; this has been changed to 20,000 positions of magnetic core, and it is believed that all 702s have been altered. However, there has been no announcement of increasing this to the 40,000 possible in the 705.
3. Input-output are always unbuffered. The 702 can read, write or compute only one at a time; there is no overlap of any of the three. It cannot use a buffered tape control unit, with the unimportant exceptions of the Types 760 and 774. The 705 differs in two respects:
  - a. Without a buffered tape control unit, it can read and write simultaneously but cannot compute during the input-output period.
  - b. It can be buffered with the Type 777 TRC, which permits almost complete overlap of reading, writing and computation.
4. The 702 has two 512-position accumulators; the 705 has one of 256, one of 32 and 14 of 16 positions.

5121 (Continued)

5. The 702 has no high-speed memory transfer, while the 705 does.
6. There are differences in the manner of terminating individual records of data; see 1103.
7. The 702 has a "read check" instruction which is omitted in the 705.
8. The first 702s had a Type 712 Card Reader with no control panel. These were superseded by the Type 714, which does; no 705s have the earlier model.
9. The 705 is, or can be, equipped with several instructions not in the 702 repertoire. These include the conditional "Transfer Any" and several control instructions peculiar to use of the Type 777 Tape Record Coordinator.
10. The record mark character code of the 702 is almost identical in function to the group mark of the 705. In the latter, the record mark is treated like any other character code, except in high-speed memory transfer.

5122 COMPARISON OF IBM TYPE 702 AND UNIVAC I

The 702 was IBM's first answer to Univac I, and it may be interesting to note briefly the relative effectiveness of an equipment first delivered  $2\frac{1}{2}$  years after the initial Univac I. This covers only highlights--and these briefly--because there no longer is any competitive factor involved.

- 5122.1 Internal Computing Speed. The 702 has a slower execution rate than Univac I on all but a few instructions. When average latency time on Univac I is considered, it appears that the 702 may be faster. Such a conclusion overlooks completely the large amount of "housekeeping" involved in handling variable word-length in the 702. With normal minimum latency coding, Univac I is almost always faster in internal computations, allowing for extractions which are required with Univac's fixed word length.
- 5122.2 Input-Output Transfer Rates. The 702 can read or write tapes at an effective rate of about 11,900 characters per second (with 1,000-character tape blocks), compared with Univac's 7,200; instantaneous rates are 15,000 and 12,800, respectively. However, Univac reads and writes simultaneously, while the 702 does one or the other, but not both. Thus Univac reads and writes a total of 14,400 characters per second, which is also the effective rate in tape-limited operations. Because the 702 is unbuffered and internal computation occurs outside tape time, its effective rate is greatly reduced--with its fairly slow execution speed, it is doubtful if normal file processing can exceed about 6,000 characters per second combined read and write, or about 40% of Univac's production.
- 5122.3 Comparison Summary. The cost of a typical 702 installation is, in general, somewhat less than a comparable Univac I; the principal cost differences arise in the periphery units rather than in the processor and tapes. A 702 with ten Type 727 Tape Units totals \$16,150 monthly with 10,000 characters of high-speed storage or \$17,650 with 20,000 characters. Univac I totals \$16,590 with 1,000 words (12,000 characters). In almost all applications, Univac I will turn out at least twice as much work as the 702, and may exceed three times as much, depending partly upon the amount

5122.3 (Continued)

of checking done in a 702 program. Including peripheral unit costs, Univac I can be considered, with a considerable safety margin, to be twice as productive, per dollar, as the Type 702.

5123 INSTALLATIONS OF IBM TYPE 702

As of 31 January 1957 the IBM Type 702 installations were:

Bank of America	San Francisco, California
State of California (Department of Employment Security)	Sacramento, California
Consolidated Edison	Chicago, Illinois
Esso Standard Oil	Baton Rouge, Louisiana
Ford Motor Company	Detroit, Michigan
General Electric Company (2)	Hanford, Washington (AEC)
	Schenectady, New York
IBM	Poughkeepsie, New York
Monsanto Chemical Company	St. Louis, Missouri
Pratt and Whitney	Hartford, Connecticut
Prudential Insurance Company	Newark, New Jersey
U. S. Air Materiel Command (2)	Dayton, Ohio
	Oklahoma City, Oklahoma
U. S. Navy Aviation Supply Office	Philadelphia, Pennsylvania
U. S. National Security Agency	Washington, D. C.

5129 SALES POINTS ON IBM TYPE 702

Because production has been discontinued, sales points are omitted. It is possible that some installations of 702s will be released and become available for reassignment. If this equipment becomes a factor in a sales proposal, and assistance is required in meeting it, refer details to the Home Office Sales Support Staff.



5141 BASIC CHARACTERISTICS OF THE TYPE 705

The characteristics included here are limited to those essentially a part of the central processor. Characteristics of the periphery units are part of the discussion of each of them, and some of these can be considered as main frame attributes. However, none are repeated here. Refer to 5142 for a chart of all equipment type numbers which can be associated with the 705 and to 0910.1 for a cross-reference between type numbers and manual sections.

Type of storage:	Magnetic core
Storage capacity:	20,000 or 40,000 characters
Information representation:	Binary-coded characters of 6 bits each plus check bit
Word length:	Variable, one character up to memory capacity
Address range:	00000-19999 or 00000-39999
Memory addressability:	Each character addressable; one address to each character position
Type of operation mode:	Serial by character, bit-parallel within character. (5-character parallel ac- cess to instructions)
Type of instruction:	Single address modified. (1) One character operation code (2) Four digit numeric address (3) Register designation by zone bits in address portion
Basic cycle rate:	17 microseconds
Machine checking features:	(1) Check bit on data transfers (2) Validity of operation codes and addresses in instructions (3) Partial check on adder
Direct inputs into memory:	(1) Magnetic tape (2) Punch card reader
Direct outputs from memory:	(1) Magnetic tape (2) Card punch (3) Printer
Buffering of input-output:	Both buffered and unbuffered possible, depending upon periphery unit used
Registers:	Arithmetic and logical use: (1) 256-character general purpose Accumulator (2) 15 limited-purpose auxiliary storage units, one of 32 positions and 14 of 16 positions Instruction Register 2 Memory Address Counters High, Equal and Low Comparison flip-flops

With the exception of basic differences noted in 5121, these character-istics are applicable also to the Type 702.

5142 EQUIPMENT CONFIGURATIONS IN TYPE 705 INSTALLATIONS

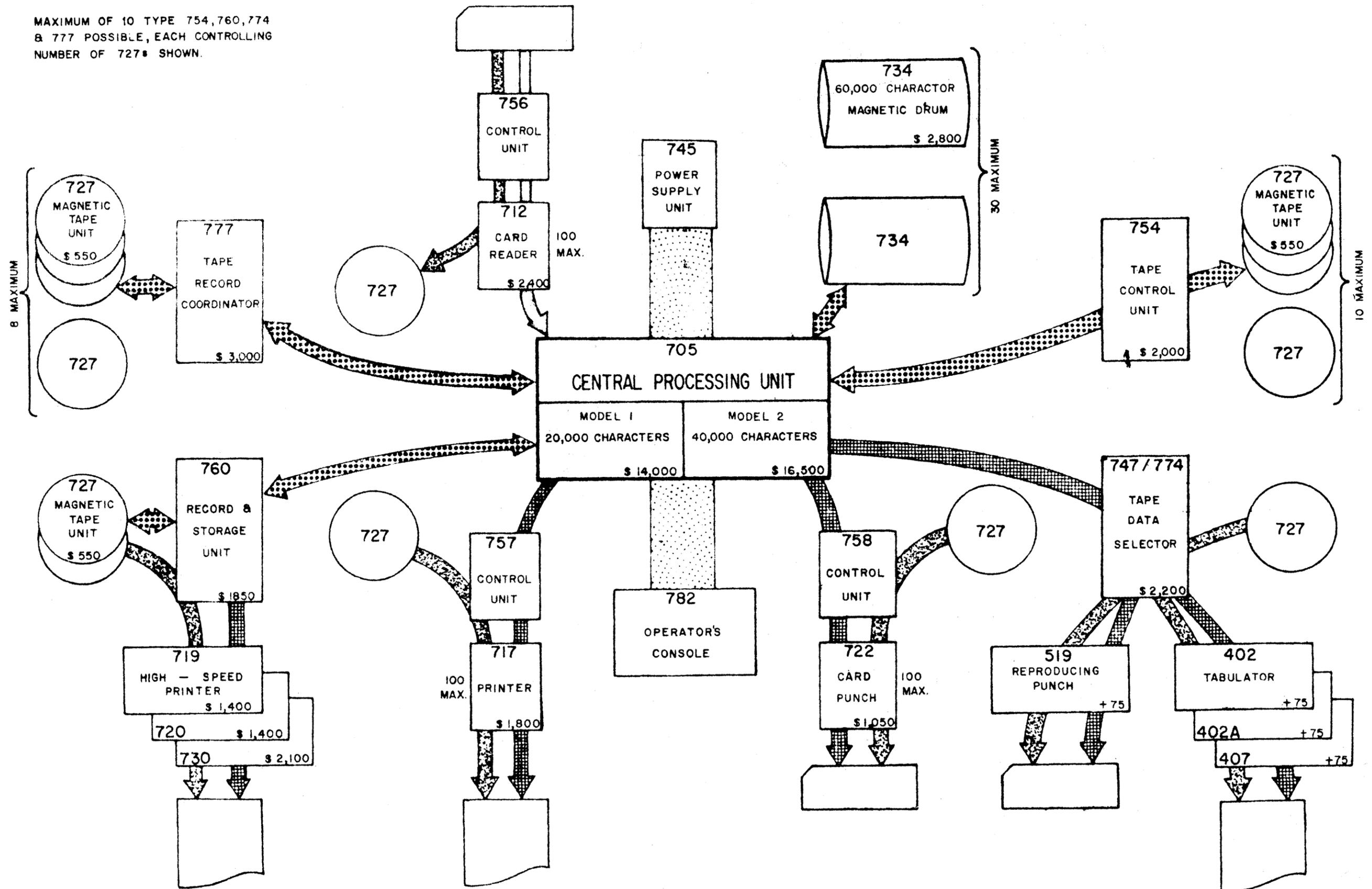
The chart on the next page depicts graphically the various equipments which may be associated with a Type 705 installation in both on-line and off-line operation. For convenience, the various units possible are listed below, with certain additional information pertaining to each unit.

<u>Type Number</u>	<u>Description</u>	<u>Monthly Rental</u>	<u>Operation</u>		<u>Maximum Number On-Line</u>	<u>Address Range On-Line</u>	<u>Maximum Number Tape Units</u>
			<u>On- Line</u>	<u>Off- Line</u>			
705	Central Processing Unit (\$11,800)						
745	Power Supply Unit (\$1,200)	\$14,000	Yes	No	1	----	----
782	Operator's Console (\$1,000)						
714/759	Card Reader	2,400	Yes	Yes	100	0100-0199	--
722/759	Card Punch	1,050	Yes	Yes	100	0300-0399	--
717/757	Printer	1,800	Yes	Yes	100	0400-0499	--
734	Magnetic Drum	2,800	Yes	No	30	1000-9999	--
727	Magnetic Tape Unit	550	Yes	Yes	100	0200-0299	*
754	Tape Control Unit	2,000	Yes	No		0200-0299	10
760	Record and Storage Unit	1,850	Yes	Yes	10	0200-0299	2
774	Tape Data Selector	2,200	Yes	Yes		0200-0299	1
777	Tape Record Coordinator	3,000	Yes	No		0600-0699	8
719	High-Speed Printer	1,400	Yes	Yes	--	02X4	--
720	" " "	1,400	Yes	Yes	--	02X4	--
730	" " "	2,100	Yes	Yes	--	02X4	--
	Typewriter	75	Yes	No	100	0500-0599	--

\*The maximum number of tape units in a 705 installation depends upon which tape control units are used. With ten Type 754's, 100 tape units are possible; this is the upper limit. There is no limit in the number of units which may be used off-line.

# IBM TYPE 705 EDPM

MAXIMUM OF 10 TYPE 754, 760, 774 & 777 POSSIBLE, EACH CONTROLLING NUMBER OF 727\* SHOWN.



5142.1 "Normal" Type 705 Installations. A "normal" Type 705 installation will vary considerably in cost, depending not only upon the number of tape units and types of control units, but also upon the array of periphery equipments required. For many applications, one card reader at 250 per minute and one punch at 100 per minute are enough; there may even be little use of the latter. Similarly, the nature and volume of printing will affect the choice of a printer and its cost. With a print volume of not more than about 70,000 to 80,000 lines per shift, this example may be considered a roughly representative installation:

1 -- Type 705 CPU, 20,000 characters. . . . .	\$ 14,000
2 -- Type 777 Tape Record Coordinators. . . . .	6,000
10-16 -- Type 727 Tape Units. . . . .	5,500 - 8,800
1 -- Type 714 Card Reader with 759 Control. . . . .	2,400
1 -- Type 722 Card Punch with 758 Control. . . . .	1,050
1 -- Type 717 Printer with 757 Control. . . . .	1,800

Total monthly rental . . . . . \$30,750 - 34,050

Because tape units are necessary to operate the peripheries off-line, ten is a relatively low complement -- 12 or 13 is probably the minimum number which can provide efficient CPU and off-line operation. The card punch may be omitted, but still leaves the cost of a reasonably good installation at more than \$30,000 monthly. When the addition of one or two Type 734 magnetic drums, replacement of the Type 717 by a higher-speed Type 720/760 printer, and possible inclusion of a Type 774 Tape Data Selector are taken into account, it becomes apparent that \$35,000 monthly is a more appropriate rental for an average basic 705 installation.

5142.2 Stripped-Down Type 705 Installations. To combat the proposal of a Univac I in areas within its potential scope, IBM has proposed greatly stripped-down 705 installations strictly to meet the Univac I cost. Typically, such an installation may consist of:

1 -- Type 705 CPU, 20,000 characters. . . . .	\$ 14,000
1 -- Type 754 Tape Control Unit . . . . .	2,000
7 -- Type 727 Tape Units. . . . .	3,850
1 -- Type 714 Card Reader with Type 759 Control . . . . .	2,400
1 -- Type 717 Printer with Type 757 Control . . . . .	1,800

Total monthly rental . . . . . \$ 24,050

This is just about bedrock for a 705 installation. It provides no buffering of tape input and output (this could be made available for another \$1,000 monthly), which means that all computation occurs outside tape time. The limited number of tape units necessitates quite inefficient processing on many jobs; sorting, for example, would be restricted to a two-way merge, rather than the faster 3- and 4-way merges. There is no card punch output whatsoever and only a 150-line printer. A Univac I of exactly the same units would run \$21,485 (which includes buffered input and output, a 600-line printer and two more tape units for the printer and card reader). A full-fledged Univac I with ten on-line Uniservos, a T-C and C-T converter and an HSP would run only \$24,760, or little more than the stripped-down 705.

5142.3 Maximum Type 705 Installation. To provide its best possible cost-of-performance ratio, IBM may propose a wider array of equipments than either of the preceding two versions. They are particularly common in proposals hinging on the handling of extremely large master files.

1 -- Type 705 CPU, 40,000 characters. . . . .	\$ 16,500
3 -- Type 777 Tape Record Coordinators. . . . .	9,000
16 -- Type 727 Tape Units. . . . .	8,800
1 -- Type 714 Card Reader with Type 759 Control . . . . .	2,400
1 -- Type 760 Record Storage Unit . . . . .	1,850
1 -- Type 720 Printer . . . . .	1,400
1 -- Type 722 Card Punch with Type 758 Control. . . . .	1,050
1 -- Type 734 Magnetic Drum . . . . .	2,800

Total monthly rental. . . . . \$ 43,800

The sales emphasis by IBM on a proposal of this general configuration--indeed on any proposal involving two or more Type 777s--is on the faster tape speed claimed to be possible by paralleling input and output in two or three of them. This is discussed exhaustively in 1160, which develops detailed time and cost data for this type of operation, together with comparisons against Univac II performance. The use of the third TRC in theory makes even greater processing speeds possible, but the analysis of 1160 indicates definitely that the percentage increase is not equivalent to IBM's claims. A 705 proposal involving three TRCs may indicate that a job can be done faster than on Univac II, but the cost of doing a unit of work will be greater.

Subsequent paragraphs in this section will bring out additional points on the cost-of-performance ratio of the IBM 705. The three "typical" configurations drawn up are only indicative of the minimum, average and maximum proposals which may be encountered in meeting IBM competition.

5150 IBM TYPE 709 ELECTRONIC DATA PROCESSING MACHINE

The IBM Type 709 is a large-scale EDPM designed for both business data handling and scientific computation. It is a further extension of the Type 704, having the same basic logic and operating speed, but incorporating a number of new instructions to facilitate the handling of business-type information and providing greatly improved input-output facilities. First deliveries are scheduled for the latter part of 1958.

The type designation 709 refers specifically to the Analytical and Control Unit, without the high-speed storage (memory) or power supplies necessary for an operative central processor. In common usage, it also refers to an EDPM installation having a 709 as the central processor.

Information available is not specific as to the exact units necessary to form various configurations of the central processor. Those known or believed to be associated with the main frame are included in the price listing of the next section. The use of the Type 736 Power Supply Unit is uncertain, but it probably is required at all times. The difference between the two models of the Type 741 also is not known; a possible explanation is that the more expensive one is used with the 32,768-word magnetic core storage, and the cheaper with the smaller ones.

5150.1 Cost. Rental and purchase prices of the various units necessary to make a central processor are:

	<u>Purchase</u>	<u>Rental</u>
Type 709 -- Analytical Control Unit. . .	\$ 600,000	\$ 10,000
Type 736 -- Power Supply Unit. . . . .	57,200	1,100
Type 737 -- First 4,096 words of Magnetic Core Storage (Model 3) . . .	208,000	4,000
Type 737 -- Next 4,096 words of Magnetic Core Storage (Model 4) . . .	193,000	3,200
Type 738 -- 32,768 word Magnetic Core Storage. . . . .	1,040,000	20,000
Type 741 -- Power Supply (Model 2) . . .	72,800	1,400
Type 741 -- Power Supply (Model 3) . . .	96,000	1,600
Type 746 -- Power Distribution Unit (Model 2 or 3) . . . . .	67,600	1,300

Assuming that a Type 736 is necessary in all configurations and that the Type 741 Model 3 replaces the Model 2 with the large core memory, these prices result for the three versions of the 709 central processor:

Type 709 -- 4,096-word core storage. . .	\$1,005,600	\$ 17,800
Type 709 -- 8,192-word core storage. . .	1,198,600	21,000
Type 709 -- 32,768-word core storage . .	1,860,800	34,000

5151 BASIC CHARACTERISTICS OF THE IBM TYPE 709

The characteristics included here are limited to those essentially a part of the central processor. Characteristics of the periphery units are part of the discussion of each of them, and some of these can be considered as main frame attributes. Refer to 5152 for a chart of all equipment type numbers which can be associated with the 709, either on- or off-line, and to O910.1 for a cross-reference between type numbers and manual sections.

Type of storage:	Magnetic core
Storage capacity:	4,096 word (Type 737 Model 3 memory) 8,192 word (Type 737 Model 3 and 4) 32,768 word (Type 738)
Word length:	36 bits (fixed)
Information representation:	Binary (35 bits plus sign bit) or character code (six 6-bit codes)
Memory addressability:	Word-addressable; 15 bits or five octal digits maximum address
Type of operation mode:	Binary on one word
Type of memory access:	One word in parallel
Type of instruction:	Single address modified to include Index Register of B-Box address
Arithmetic operation:	Binary
Decimal-to-binary conversion:	Table lookup
Binary-to-decimal conversion:	Table lookup
Internal checking:	None; there are no check-bits attached to words
Basic cycle rate:	12 microseconds (includes access and operation)
Instruction execution times:	24 or 36 microseconds for almost all in- structions except multiply and divide
Registers:	(1) Accumulator (P; Q, 1-35) (2) MQ Register (S, 1-35) (3) Storage Register (essentially high- speed bus) (4) 36 Sense Indicators (5) Instruction Register (6) 3 Index Registers (B-Boxes)
Direct inputs into memory:	(1) Magnetic tape (2) Punch card reader (3) Magnetic Drum
Direct outputs from memory:	(1) Magnetic tape (2) Magnetic drum (3) Cathode ray tube (4) Printer (5) Card punch

5152 EQUIPMENT CONFIGURATIONS POSSIBLE IN A TYPE 709 INSTALLATION

The chart on the next page depicts graphically the various equipments which may be associated with a Type 709 in on-line and off-line operation. Certain pertinent data are summarized in the table below.

<u>Type Number</u>	<u>Description</u>	<u>Monthly Rental</u>	<u>Operation</u>		<u>Maximum Number On-Line</u>	<u>Address Range On-Line*</u>	<u>Maximum Number Tape Units</u>
			<u>On-Line</u>	<u>Off-Line</u>			
<u>Magnetic Core Storage Units</u>							
737-3	4,096 words	\$4,000	Yes	No	1	00000-07777	--
737-3&4	8,192 words	7,200	Yes	No	1	00000-17777	--
738-1	32,768 words	20,000	Yes	No	1	00000-77777	--
<u>Input-Output Units</u>							
711-2	Card Reader (250/min)	800	Yes	No	3	0X321	--
716-1	Printer (150/min)	1,200	Yes	No	3	0X361	--
721-1	Card Punch (100/min)	600	Yes	No	3	0X341	--
729	Magnetic Tape Unit	650	Yes	Yes	48	See note**	--
755	Tape Control Unit	1,800	Yes	No	6	Non-Addressable	8 each
766	Data Synchronizer	3,500	Yes	No	3	See note***	16 each
733	Magnetic Drum \$2,800 or 3,500	3,500	Yes	No	2	00301-00310	--
740	Cathode Ray Tube	2,700	Yes	No	1		--
780	Cathode Ray Tube	150	Yes	No	1		--
727	Magnetic Tape Unit	550	No	Yes	--	--	--
774	Tape Data Selector	2,200	No	Yes	--	--	--
760	Record Storage Unit	1,850	No	Yes	--	--	--
719	Printer	1,400	No	Yes	--	--	--
720	Printer	1,400	No	Yes	--	--	--
730	Printer	2,775	No	Yes	--	--	--
717/757	Printer	1,800	No	Yes	--	--	--
722/758	I-C Converter	1,050	No	Yes	--	--	--
714/759	C-I Converter	2,400	No	Yes	--	--	--

\*Octal addresses. "X" is 1, 3, 5.

\*\*Magnetic tape addresses depend upon Data Synchronizer Channel, mode of writing and tape unit number.

\*\*\*See Section 1160.

5152.1 Equipment Configurations in Type 709 Installations. The specific array of periphery units and size of memory in a 709 installation can be expected to show rather wide variation, depending upon the major purpose of the operation. A customer with predominantly scientific uses for the equipment may sacrifice on some of the input-output devices, because these requirements are only nominal. Conversely, an installation devoted mainly to business data handling may be of considerably different composition. The possible installations shown in the next few sections are only roughly representative and will be revised as soon as IBM indicates a pattern in its sales proposals, and the actual arrays of equipments ordered by the first few customers are available.

5152.2 Minimum Type 709 Installation, Scientific Purposes. Desires expressed by users in the past indicate that the largest feasible internal memory is a definite attraction in the scientific computer field. With the 32,768-word core storage memory announced for both the Type 704 and Type 709, the exact place of each equipment in the marketplace has not yet been evaluated. The essential difference is the faster input-output rate of the 709 (in terms of internal computing time required), although this newer equipment also has some advantage in internal arithmetic operations because of its more extensive instruction repertoire. The relative speed improvement in typical mathematical computations is not known. However, a minimum-type 709 may be predicated upon this equipment array:

Type 709 -- Analytical Control Unit . . . . .	\$ 10,000
Type 736 -- Power Supply Unit . . . . .	1,100
Type 741 -- Power Supply Unit (Model 2) . . . . .	1,400
Type 746 -- Power Distribution Unit . . . . .	1,300
Type 737 -- 8,192-Word Magnetic Core Storage . . . . .	7,200
Type 733 -- Magnetic Drum . . . . .	2,800
Type 766 -- Data Synchronizer . . . . .	3,500
Type 711 -- Punch Card Reader (Model 2) . . . . .	800
Type 716 -- Printer . . . . .	1,200
Type 721 -- Card Punch . . . . .	600
8 Type 729 -- Magnetic Tape Units . . . . .	5,400
Type 755 -- Tape Control Unit . . . . .	1,800
	\$ 37,100
Total Monthly Rental. . . . .	

This probably is about bedrock for a 709 installation.

5152.3 Minimum Type 709 Installation, Business Purposes. The approximate minimum 709 installation for business data processing is predicated upon the fact that sufficient input-output units must be provided to justify the cost of the configuration. The theoretical internal speed of the 709 is quite fast, and the new instructions permit a rather definite saving in internal computing time as compared with the 705. To use this additional speed economically means more input and output units to be handled in parallel; because IBM has been promoting the use of two TRCs with the 705 as a means of souping up its data transfer rate, a comparable technique must be employed with this equipment. Consequently, for business purposes it is considered that a minimum of two Data Synchronizers will be required. Even here, the economy of a 709 as compared with the slower but cheaper 705 must be kept in mind. It is not at all certain that the minimum con-

## 5152.3 (Continued)

figuration is economically justified against a 705 of approximately equal production capabilities.

	Type 709 -- Analytical Control Unit . . . . .	\$ 10,000
	Type 736 -- Power Supply Unit . . . . .	1,100
	Type 741 -- Power Supply Unit (Model 2) . . . . .	1,400
	Type 746 -- Power Distribution Unit . . . . .	1,300
	Type 737 -- 8,192-Word Magnetic Core Storage . . . . .	7,200
2	Type 766 -- Data Synchronizers . . . . .	7,000
4	Type 755 -- Tape Control Units . . . . .	7,200
16	Type 729 -- Magnetic Tape Units . . . . .	10,400
	Type 711 -- Punch Card Reader (Model 2) . . . . .	800
	Type 721 -- Card Punch . . . . .	600
	Type 717 -- Printer With Type 757 Control Unit . . . . .	1,800
	Total Monthly Rental . . . . .	\$ 48,800

This minimum configuration does not provide a particularly large memory. Punch card input and output is not efficient, requiring program conversion as a part of internal computation, and probably would not be suitable if any significant volume of either type operation existed. Printing is on an off-line basis only, at a rated 150 lines per minute, or a maximum expectancy of about 50,000 net lines per 8-hour shift. Thus the installation is not especially attractive.

5152.4 Typical 709 Installation, Business Purposes. An equipment configuration which appears more suitable for a large-volume business data handling operation, of a scope for which the 709 capabilities may be attractive, is as follows:

	Type 709 -- Analytical Control Unit . . . . .	\$ 10,000
	Type 736 -- Power Supply Unit . . . . .	1,100
	Type 741 -- Power Supply Unit (Model 3) . . . . .	1,600
	Type 746 -- Power Distribution Unit . . . . .	1,300
	Type 738 -- 32,768-Word Magnetic Core Storage . . . . .	20,000
3	Type 766 -- Data Synchronizers . . . . .	10,500
6	Type 755 -- Tape Control Units . . . . .	10,800
20	Type 729 -- Magnetic Tape Units . . . . .	13,000
	Type 760 -- Record Storage Unit . . . . .	1,850
	Type 730 -- High-Speed Printer (off-line only) . . . . .	2,775
	Type 714 -- Punch Card Reader With Type 759 Control Unit . . . . .	2,400
	Type 722 -- Card Punch With Type 758 Control Unit . . . . .	1,050
2	Type 727 -- Magnetic Tape Units (for off-line equipments) . . . . .	1,100
	Total Monthly Rental . . . . .	\$ 77,475

This obviously is not cheap, but also can be shaved in several areas. However, a range of \$60,000 and up monthly appears to be a reasonable conglomeration of units for this system. It is rather evidently priced well above anything else so far announced in the business data processing field and, for this reason, is not strictly competitive with either the Univac II or IBM's own Type 705.

5152.5 IBM Announcements of Type 709 Costs. In its publicity releases and advertising on the 709, IBM has been citing monthly rentals of around \$55,000 as "typical" costs of an installation. From the comparative rentals developed in the preceding two paragraphs, it is rather obvious that IBM's "typical" 709 is, in some respects, a quite stripped-down data processor. For one thing, the input-output array of 5152.3 is just about as minimum as can be visualized, costwise, for a commercial EDPM; another \$6,000 worth of equipment, which would bring it up to \$55,000, won't add up to much. It certainly won't admit of a 32,768-word memory, which is one of the plugged features of the equipment. In some respects, the publication of this figure has the appearance of a "loss leader;" get the customer's order at all costs and then gradually break the news that another ten to twenty thousand a month ought to be added to get a decent installation.

5153 OPERATING FEATURES OF TYPE 709 EDPM

This section is presented in preliminary form only and simply highlights some of the features of the 709 which have been added to make it more adaptable to business data processing. A complete revision of this material will be forwarded shortly. This section does not include the characteristics of the Type 766 Data Synchronizer, which is written up in detail in 1160. Characteristics of the Type 729 Tape Unit are included in Section 1105.

- 5153.1 Magnetic Tape -- Memory Transfer Rates. IBM's advertising and promotional literature make it appear as if the tape speed or transfer rate has been improved over that available in the 702, 704 and 705. Nothing of the kind has developed, and nothing new has been added with the sole exception of the verification of tapes immediately after writing. Their statements of tape holding the equivalent of 350 decimal digits per inch, with a transfer rate of 25,000 digits per second, are nothing but another way of stating what can be done with the existing Type 704. The explanation is simple. If information is in binary form internally and written on tape in that fashion, then the 1200 information bits per inch of tape (6 bits per character, plus check bit, at 200 characters per inch) is the equivalent of about 350 decimal digits, provided practically all bits represent information. This will not always be the case; the internal word is 35 bits plus sign, which is equivalent to about 11 decimal digits. Unless typical numeric elements of business information (which average much less than 11 digits) are packed two or three into the word, there may be only a few--10 or 20--significant information bits. Consequently, the statement that tape holds the "equivalent of 350 decimal digits" is the maximum condition; in practice, it will practically always be less, and often a whale of a lot less. The 729 tape units used with the 709 still pack at 200 to the inch, still move the tapes at 75 inches per second, and still handle exactly 15,000 tape characters per second maximum transfer rate.
- 5153.2 Indirect Addressing. A major change from the 704 is the inclusion of a form of "indirect addressing" in an instruction; many, but by no means all, instructions can be so treated. An example will serve to illustrate how it operates.

Suppose memory address 1234 contains the number 1111. In the normal single-address instruction of the form

ADD    1234

the EDPM goes to address 1234 and takes its contents, 1111, to add to some other quantity. In "indirect addressing," the contents of 1234 are not added, but instead used as the location of another number which is the quantity actually used. In this example, instead of adding "1111," the 709 would go to location or address 1111 and actually add the quantity in that location. This is the meaning of "indirect addressing;" the address attached to the instruction is used only to find the actual location for the operand really involved. In the 709, the indirect addressing use of an instruction adds 12 microseconds to the normal execution time.

## 5153.2 (Continued)

A major use of this feature is in the sorting and merging of raw data. The conventional methods of sorting involve the constant reshuffling of the items internally, by means of memory transfers, to place them in proper sequence. With "indirect addressing," the items themselves are never moved. Instead, a "table" of addresses of each index number in a group of unsorted items is set up in memory, and the comparison instructions refer to the addresses of each word of the table, not of the data. The table itself is sorted into the sequence of the items, but this involves only one-word memory transfers rather than the movement of long strings of information. When coupled with the use of multiple control words in the Type 766 Data Synchronizer, see 1160, the entire sorting operation, including writing out of a long sequenced string of items, is done without ever moving the data from the original input locations. The "table" words are so set up in format that they are used directly as Control Words in the output operation. As an additional note, this technique makes it feasible to sort items of different lengths, rather than treating each item as a constant number of words or characters identical with all other items.

Indirect addressing may be used with or without index registers applying to the same instruction.

5153.3 Conversion Instructions. Three new instructions have been added to the equipment which permit the conversion of information from one number or character code system to another. The specific and most obvious intent is to facilitate conversion from one system of character codes to another or from decimal numbers to binary numbers and the converse. The method of approach is sufficiently general to permit conversion from almost any type of character code system to any other; for example, Univac to 705 codes or vice versa.

The three instructions all involve the use of tables. The general use of the instructions follows the principle outlined: Each 709 word is treated as six 6-bit code combinations. Each 6-bit group is added to a base table address, in binary addition, to create a new address within the table. The contents of this new address are two-fold: (1) a 6-bit group which is the new code combination corresponding to the old one; and (2) a new base address in the table from which to begin the addition for the next 6-bit code combination in the word to be converted. The new base address may be the same as the original one, or completely different, thus permitting repeated references within a single table or a "bootstrapping" operation through a complex, segmented table. This is the basic technique of conversion by replacement, the purpose of two of the three instructions, and results in a new word of six 6-bit code combination.

As an example of how this works, let's convert a "\$" in Univac character code to the same symbol in the 709 code. Omitting check bits, the Univac character code for "\$" is 10 1101, which as a 6-bit binary number is the same as decimal "45". The 709 internal code for "\$" is 10 1011. Now let us set up in memory a table beginning at location 1000 and containing 64 words, the maximum number of code combinations in each of the two types

## 5153.3 (Continued)

equipments. (Decimal notation will be used in this explanation for memory addresses, although the actual operations in the machine occur, of course, in binary). In location 1045 is a word which consists of two parts: (1) its left-most six bits are "10 1011," which is the 709 internal code for "\$;" and (2) its rightmost bits are the number "1000" (in binary, of course).

By a separate instruction, the word in Univac character codes is placed in the MQ register. Now a "Convert by Replacement from MQ" instruction, with address 1000, acts as follows: The low-order six bits of the word in the MQ register are added to 1000; with a dollar sign in this position, the result is 1045. Then the contents of location 1045 are obtained and the left-most six bits placed in the left end of the accumulator. The rightmost bits reset the starting point back to 1000. Now both the MQ register and accumulator are shifted six bits to the right and the cycle is repeated. This continues until six groups have been converted, at the conclusion of which the accumulator contains, in proper order, the new codes corresponding to those which were in the MQ. Provision is made for terminating the conversion after any number of characters if fewer than six are involved.

Two of the three conversion instructions operate in this manner and are designed for character replacement. It is not necessary to return to a pre-set address at the beginning of the table for each group of six bits to be converted; each "word" in the table is used to designate the origin point for the next character to be converted. It is at once obvious to the cryptographically-inclined that this technique can be used to generate virtually unbreakable codes, although the significance of this with respect to business information is not at all evident in other than the communications area.

The final conversion instruction operates by addition, and is the one used to convert from decimal to binary and vice versa. In this case, 20 bits from the words in the table are added successively to the contents of the accumulator to arrive at a total.

The conversion instructions are used with others in small sequences to perform the entire cycle. IBM quotes these time figures for typical conversions:

- Replacing leading zeroes by blanks in a six-digit coded word: Three instructions and 144 microseconds total execution time.
- Same for a 12-digit (two-word) field: 168 or 336 microseconds, depending upon whether there are more than five zeroes to be replaced.
- Converting six 6-bit character codes to another code system: Four instructions and 168 microseconds total execution time.
- Converting a six-digit coded number to binary form: Five instructions and 204 microseconds total execution time.
- Converting a 20-bit binary number to coded decimal: Five instructions and 360 microseconds total execution time.

- 5153.4 Index Transmission Instructions. The use of index registers in the 704 is subject to some restrictions which have been eliminated in the 709 by incorporation of several new instructions to permit more flexible manipulation and use of the "b-box" technique. These instructions have been incorporated along the lines of recommendations made by the SHARE organization, a group of IBM 704 users.
- 5153.5 Variable Length Multiply and Divide. The most time-consuming instructions in almost all EDPMs are the multiply and divide operations. Rather than always multiplying full words, or developing full word quotients, the 709 incorporates special instructions which permit the programmer to designate the number of bits involved in a multiplier or quotient and thus eliminate the time required to pass through the unused bit positions. Although of use in business applications, which seldom involve multipliers or quotients of the size of the 11-digit (approximate) word of the 709, the number of such operations is relatively few and overall time savings are not too significant. However, it is a feature.
- 5153.6 Zero Skipping on Multiplication. Somewhat akin in objective to the previous feature is the modification of the multiplication operation to permit speeding up execution time by rapidly passing over zeroes in the multiplier. If a multiplier is all zeroes, the answer is created in two machine cycles, 24 microseconds. Normal multiplications vary from 72 to 240 microseconds, depending upon the number and distribution of zero bits in the multiplier.
- 5153.7 External Signal and Floating Point Traps. In some types of on--line data reduction work (real-time processing), it may be desirable to have information from an external source interrupt the program being processed so that the new information can be handled immediately. An external signal meeting predetermined requirements will cause the 709 to complete the instruction it is on and jump arbitrarily to memory address 0004 for the next instruction. Simultaneously, the address of the instruction which would otherwise have been next executed is placed in 0003.
- The "floating point trap" is somewhat akin to the automatic overflow logic of Univac. If an overflow or underflow occurs in floating point arithmetic operations, the address of the normally-next instruction is placed in location 0000 and the 709 skips to location 0002 rather than continuing in sequence. This trap is operative only in floating point arithmetic and not in normal binary instructions.
- 5153.8 Sense Indicators. A new 36-bit register has been added to provide what amounts to 36 "selectors." Each position can be set to "1" (on) or "0" (off) and singly or in groups tested to provide alteration switches in a program. They can be used also for extracting bits within a 709 word or for permitting tests to be made on groups of bits within a word.
- 5153.9 Number of Instructions in the 709. The 709 has 189 instructions, compared with 87 in the 704. Of the 87, 84 are included in the 709 and perform identical functions, although the actual execution of some of them is changed (i.e., in multiplying). The single "redundancy tape test" of the 704 becomes six separate instructions, one for each of the six possible

## 5153.9 (Continued)

Data Synchronizer channels. The "copy" instruction used for all input-output operations in the 704, is restricted to drums and cathode ray tube in the 709. For all other devices, the Data Synchronizers and their array of instructions have been added. Finally, the "transfer on MQ overflow" of the 704 has been dropped; a new set of operational codes has been added to handle this condition.

Of the 189 instructions, 42 are devoted to operations of the six Data Synchronizer channels although they accomplish only seven different functions. Each of the seven functions has six separate operation codes, one for each channel. Similarly, a number of the other new instructions added are essentially the converse of operations already possible; i.e., "Transfer on Storage Zero" and "Transfer on Storage Not Zero." These are programming conveniences rather than separate operations. Most of them have been added, however, as a result of suggestions made by the SHARE organization.

FIXED POINT ARITHMETIC

Code	Time (us)	Type	Index	Indir	Instruction	Description
ADD	24	B	X	X	ADD	$(m)_{S,1-35} + (AC)_{S,1-35} \rightarrow (AC)_{S,-135}$
ADM	24	B	X	X	Add Magnitude	$(m)_{1-35} + (AC)_{1-35} \rightarrow AC_{1-35}$
ACL	24	B	X	X	Add and Carry Logical Word	$(m)_{S,1-35} + (AC)_{p,1-35} \rightarrow AC_{p,1-35}$ A carry from $AC_p$ adds into $AC_q$ and $AC_{35}$ ; a carry from $AC_q$ is lost.
SUB	24	B	X	X	Subtract	$(AC)_{S,1-35} - (m)_{S,1-35} \rightarrow AC_{S,1-35}$
SBM	24	B	X	X	Subtract Magnitude	$(AC)_{1-35} - (m)_{1-35} \rightarrow AC_{1-35}$
MPY	24-240	B	X	X	Multiply	$(m) \times (AC) \rightarrow AC_{1-35}$ (most significant bits) and $MQ_{1-35}$ (least significant). $0 \rightarrow AC_{q,p}$ $AC_S = MQ_S$ Conditions: If $(m) = 0$ , $0 \rightarrow AC$ and $MQ$ and proceed to next instruction. If $(m) \neq 0$ , execution time depends on significant bits in $MQ$ .
MPR	24-240	B	X	X	Multiply and Round	Executes a Multiply (MPY) followed by a Round (RND).
DVH	240	B	X	X	Divide or Halt	If $(m) > (AC)$ : $(AC)_{S,q,p,1-35}$ and $MQ_{S,1-35}$ $(m)_{S,1-35} \rightarrow MQ$ , remainder to $AC_{S,1-35}$ . If $(m) \leq (AC)$ , no division occurs, EDPM stops with divide check indicator on.
DVP	240	B	X	X	Divide or Proceed	If $(m) > (AC)$ : Division occurs as above. If $(m) \leq (AC)$ : No division, but computer proceeds with Divide Check Indicator on.
RND	24	E	-	-	Round	If $(MQ)_1 = 1$ : "1" is added to $AC_{35}$ . If $(MQ)_1 = 0$ : No change in AC.

5154.1 Instructions of the Type 709 EDPM (continued)

FIXED POINT ARITHMETIC (continued)

Code	Time (us)	Type	Index	Indir.	Instruction	Description
VIM	24	E	X	-	Variable Length Multiply	<p><math>(m) \times (MQ)</math> number of positions specified by count Product is in <math>AC_{S, 1-35}</math> (most significant) and in high order positions of MQ (least significant). Remainder of MQ contains original contents of high order positions of MQ. (<math>AC_S = MQ_S</math>).</p> <p>Timing is dependent upon sequential zeroes in MQ. If count is zero, instruction is interpreted as No Operation.</p>
VDH	24	E	X	-	Variable Length Divide or Halt	<p>If <math>(m) (AC): (AC)_{S,q,p,1-35}</math> and <math>(MQ)_{S,1-35}</math></p> <p><math>\div (m)_{S,1-35} \rightarrow MQ_{low\ order\ positions\ specified\ by\ count.}</math> Remainder <math>\rightarrow AC_{S,1-35}</math> and MQ (high order positions minus count).</p> <p>If <math>(m) (AC): \leq</math> No division occurs, EDPM stops with Divide Check Indicator on. If count is zero, instruction interpreted as No Operation</p>
VDP	24	E	X	-	Variable Length Divide or Proceed	<p>Same as VDH, except that if <math>(m) \leq (AC)</math>, no division occurs, EDPM proceeds with Divide Check Indicator on.</p>

## 5154.1 Instructions of the Type 709 EDPM (Continued)

## FLOATING POINT ARITHMETIC

Code	Time (us)	Type	Index	Indir	Instruction	Description
FAD	84+	B	X	X	Floating Add	<p>(m) + (AC) → AC and MQ.</p> <ol style="list-style-type: none"> <li>1. MQ is cleared.</li> <li>2. (m) → SR.</li> <li>3. IF <math>(SR)_{1-8} &lt; (AC)_{1-8}</math>: (SR) and (AC) are interchanged. (Smaller characteristic always is in AC before addition). If <math>AC_p</math> or <math>q = 1</math>: Modification occurs and an incorrect answer results.</li> <li>4. <math>(AC)_S \rightarrow (MQ)_S</math>.</li> <li>5. If the difference in characteristics <math>&gt; 63</math> <math>0 \rightarrow AC</math>. Otherwise, <math>(AC)_{9-35}</math> shifts right x positions (x is difference in characteristics). Bits shifted past <math>AC_{35} \rightarrow MQ_9</math> Bits shifted past <math>MQ_{35}</math> are lost.</li> <li>6. <math>(SR)_{1-8} \rightarrow (AC)_{1-8}</math>.</li> <li>7. <math>(SR)_{9-35} + (AC)_{9-35} \rightarrow AC_{S,9-35}</math>. If signs are different, 1's complement of AC is added.</li> <li>8a. If <math>(AC)_S = (SR)_S</math> and <math> \text{sum}  \geq 1</math>, a carry is added to <math>AC_8</math>; * <math>(AC)_{9-35}</math> and <math>(MQ)_{9-35}</math> shifted right one place; <math>1 \rightarrow AC_9</math>.</li> <li>8b. If <math>(AC)_S \neq (SR)_S</math> and <math> \text{sum}  \geq 1</math>, <math>(SR)_S \rightarrow AC_S</math> and <math>MQ_S</math>; carry from <math>AC_9</math> is lost. If <math>(MQ)_{9-35} = 0</math>: 1 added to <math>AC_{35}</math>. If <math>(MQ)_{9-35} \neq 0</math>: 2's complement of <math>(MQ)_{9-35} \rightarrow MQ_{9-35}</math>.</li> <li>8c. If <math>(AC)_S \neq (SR)_S</math> and <math> \text{sum}  &lt; 1</math>: 1's complement of <math>(AC)_{9-35} \rightarrow AC_{9-35}</math>.</li> <li>9a. If <math>(AC)_{9-35} = 0</math>; AC is cleared.</li> <li>9b. Results are normalized by shifting left until <math>AC_q = 1</math>. Characteristic is reduced by one for each shift.</li> <li>10. <math>(AC)_{1-8} \rightarrow MQ_{1-8}</math>, unless <math>(AC) = 0</math>, when zeroes <math>\rightarrow MQ_{1-8}</math>.</li> </ol>

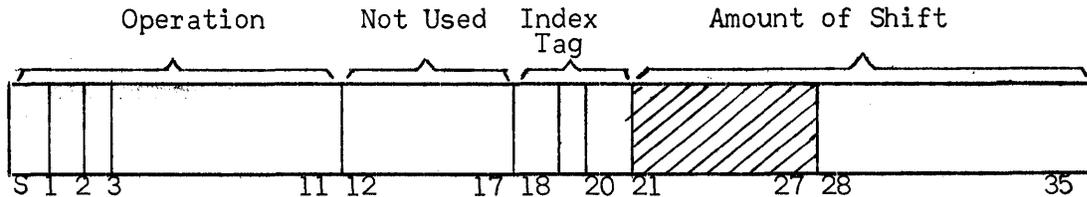
## 5154.1 Instructions of the Type 709 EDPM (Continued)

## FLOATING POINT ARITHMETIC (Continued)

Code	Time (us)	Type	Index	Indir.	Instruction	Description
FAM	84	B	X	X	Floating Add Magnitude	Same as Floating Add (FAD), except: 2. $ m  \rightarrow SR$ .
UFA	72	B	X	X	Unnormalized Floating Add	Same as Floating Add (FAD), omitting Steps 9a and 9b.
UAM	72	B	X	X	Unnormalized Floating Add Magnitude	Same as Floating Add Magnitude (FAM), omitting Steps 9a and 9b.
FSB	84	B	X	X	Floating Subtract	} Same as Floating Add Series, except the contents of memory are considered negatively.
FSM	84	B	X	X	Floating Subtract Magnitude	
UFS	72	B	X	X	Unnormalized Floating Subtract	
USM	72	B	X	X	Unnormalized Floating Subtract Magnitude	
FMP	24-204	B	X	X	Floating Multiply	$(m) \times (MQ) \rightarrow AC$ (most significant) and $(MQ)$ (least significant). Product is in normalized form only if multiplier and multiplicand are normalized.
UFM	24-204	B	X	X	Unnormalized Floating Multiply	Same as Floating Multiply (FMP) without normalizing steps.
FDH	216	B	X	X	Floating Divide or Halt	$(AC)_{S,1-35} \div (m) \rightarrow MQ$ , remainder $\rightarrow AC$ . 1. Divisor and dividend must be in normalized form to obtain correct quotient. 2. EDPM stops with Divide Check Indicator on for any of these conditions: a. $(AC)_{q,p} \neq 0$ . b. $(m) = 0$ . c. $(AC)_{9-35} \geq 2(m)_{9-35}$ .
FDP	216	B	X	X	Floating Divide or Proceed	Same as Floating Divide or Halt (FDH), except EDPM proceeds without division if $(m) = 0$ or $(AC)_{9-35} \geq 2(m)_{9-35}$ . Divide Check Indicator turned on in this case.)
FRN	24	B	-	-	Floating Round	If $(MQ)_{9-35} = 1$ , a "1" is added to $AC_{35}$ .

5154.1 Instructions of the Type 709 EDPM (Continued)

SHIFT INSTRUCTIONS

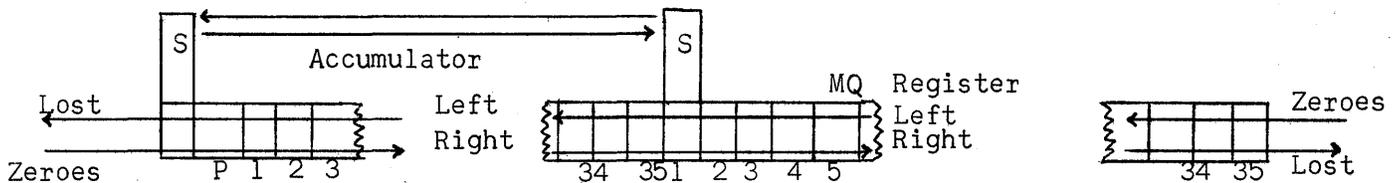


In the "Amount of Shift" field, only the last 8 bits are interpreted. A larger quantity may be used and is interpreted modulo 256.

Timing: 24 us if shift  $\leq$  8 places; add 12 us for each additional 12 places or fraction thereof.

This instruction format is considered to be Type "B."

Code	Time (us)	Type	Index	Indir	Instruction	Description
ALS	24	B	X	-	Acc Left Shift	(AC) <sub>q,p,1-35</sub> shifts left number of bit positions indicated by "Amount of Shift."
ARS	24	B	X	-	Acc Right Shift	(AC) <sub>q,p,1-35</sub> shifts right.
LLS	24	B	X	-	Long Left Shift	(AC) <sub>q,p,1-35</sub> and (MQ) <sub>1-35</sub> shift left; (MQ) <sub>S</sub> → AC <sub>S</sub> .
LRS	24	B	X	-	Long Right Shift	(AC) <sub>q,p,1-35</sub> and (MQ) <sub>1-35</sub> shift right; (AC) <sub>S</sub> → MQ <sub>S</sub> .



LGL	24	B	X	-	Logical Left	(AC) <sub>q,p,1-35</sub> and (MQ) <sub>S,1-35</sub> shift left.
LGR	24	B	X	-	Logical Right	(AC) <sub>q,p,1-35</sub> and (MQ) <sub>S,1-35</sub> shift right. These two instructions are similar to the long shifts illustrated, except that the S position of the MQ register is treated as another bit position preceding bit 1, and moves through the ACC positions.
RQL	24	B	X	-	Rotate MQ Left	(MQ) <sub>S,1-35</sub> rotated left; S bit re-enters via position 35.

In all shifts except Rotate MQ Left, a non-zero bit shifted into or through AC<sub>p</sub> turns on AC overflow indicator. Bits shifted left past q or right past 35 are lost. Positions made vacant are filled with zeroes.

5154.1 Instructions of the Type 709 EDPM (Continued)EXCHANGE INSTRUCTIONS

Only the operation code is decoded on these two instructions.

Code	Time (us)	Type	Index	Indir.	Instruction	Description
XCA	24	B	-	-	Exchange AC and MQ	$(AC)_{S,1-35} \leftrightarrow (MQ)_{S,1-35}; 0 \rightarrow AC_{q,p}$ .
XCL	24	B	-	-	Exchange Logical AC and MQ	$(AC)_{p,1-35} \leftrightarrow (MQ)_{S,1-35}; 0 \rightarrow AC_{S,q}$ .

WORD TRANSMISSION INSTRUCTIONS

Register to Memory:

STO	24	B	X	X	Store	$(AC)_{S,1-35} \rightarrow m$ .	
SLW	24	B	X	X	Store Logical Word	$(AC)_{p,1-35} \rightarrow m$ .	
STA	24	B	X	X	Store Address	$(AC)_{21-35} \rightarrow m_{21-35}$ .	Memory positions not included in these instructions are unaltered
STD	24	B	X	X	Store Decrement	$(AC)_{3-17} \rightarrow m_{3-17}$ .	
STP	24	B	X	X	Store Prefix	$(AC)_{p,1,2} \rightarrow m_{S,1,2}$ .	
STT	24	B	X	X	Store Tag	$(AC)_{18-20} \rightarrow m_{18-20}$ .	
STQ	24	B	X	X	Store MQ	$(MQ) \rightarrow m$ .	
SLQ	24	B	X	X	Store Left-Half MQ	$(MQ)_{S,1-17} \rightarrow m_{S,1-17}$ .	$(m)_{18-35}$ unaltered.
STZ	24	B	X	X	Store Zero	$+0 \rightarrow m$ .	
STL	24	B	X	X	Store Instruction Location Counter	Location of STL instruction + 1 $\rightarrow m_{21-35}$ .	

Memory to Register:

CLA	24	B	X	X	Clear and Add	$(m) \rightarrow (AC)_{S,1-35}; 0 \rightarrow AC_{q,p}$ .
CLS	24	B	X	X	Clear & Subtract	$(m) \rightarrow AC_{S,1-35}; 0 \rightarrow AC_{q,p}$ .
CAL	24	B	X	X	Clear and Add Logical Word	$(m) \rightarrow AC_{p,1-35}; 0 \rightarrow AC_{S,q}$ .
LDQ	24	B	X	X	Load MQ	$(m) \rightarrow MQ$ .

5154.1 Instructions of the Type 709 EDPM (Continued)INDEX TRANSMISSION INSTRUCTIONS

## Memory to Index:

Code	Time (us)	Type	Index	Indir.	Instruction	Description
LXA	24	B	-	-	Load Index from Address	$(m)_{21-35} \rightarrow \text{Index.}$
LXD	24	B	-	-	Load Index from Decrement	$(m)_{3-17} \rightarrow \text{Index.}$
LAC	24	B	-	-	Load Complement of Address in Index	2's complement of $(m)_{21-35} \rightarrow \text{Index.}$
LDC	24	B	-	-	Load Complement of Decr't in Index	2's complement of $(m)_{3-17} \rightarrow \text{Index.}$

## Instruction to Index:

AXT	24	B	-	-	Address to Index True	21-35 of instruction in $m \rightarrow \text{Index.}$
AXC	24	B	-	-	Address to Index Complemented	2's complement of 21-35 of instruction in $m \rightarrow \text{Index.}$

## Accumulator to Index:

PAX	24	B	-	-	Place Address in Index	$(AC)_{21-35} \rightarrow \text{Index.}$
PDX	24	B	-	-	Place Decrement in Index	$(AC)_{3-17} \rightarrow \text{Index.}$
PAG	24	B	-	-	Place Complement of Address in Index	2's complement of $(AC)_{21-35} \rightarrow \text{Index.}$
PDC	24	B	-	-	Place Complement of Decrement in Index	2's complement of $(AC)_{3-17} \rightarrow \text{Index.}$

## Index to Memory:

SXA	24	B	-	-	Store Index in Address	$(\text{Index}) \rightarrow m_{21-35}.$
SXD	24	B	-	-	Store Index in Decrement	$(\text{Index}) \rightarrow m_{3-17}.$

## Index to Accumulator:

PXA	24	B	-	-	Place Index in Address	$0 \rightarrow \text{AC}; (\text{Index}) \rightarrow \text{AC}_{21-35}.$
PXD	24	B	-	-	Place Index in Decrement	$0 \rightarrow \text{AC}; (\text{Index}) \rightarrow \text{AC}_{3-17}.$

5154.1 Instructions of the Type 709 EDPM (Continued)INDEX CONTROL INSTRUCTIONS

Code	Time (us)	Type	Index	Indir.	Instruction	Description
TIX	24	A	-	-	Transfer on Index	If (Index) $\gt$ Decrement: (Index) - Decrement $\rightarrow$ Index; control $\rightarrow$ m. If (Index) $\leq$ Decrement: (Index) unchanged, control maintains normal sequence.
TNX	24	A	-	-	Transfer on No Index	If (Index) $\leq$ Decrement: (Index) unchanged, control $\rightarrow$ m. If (Index) $\gt$ Decrement: (Index) - Decrement $\rightarrow$ Index; control maintains sequence.
TXH	24	A	-	-	Transfer on Index High	If (Index) $\gt$ Decrement: Control $\rightarrow$ m. If (Index) $\leq$ Decrement: Control maintains sequence.
TXL	24	A	-	-	Transfer on Index Low or Equal	If (Index) $\leq$ Decrement: Control $\rightarrow$ m. If (Index) $\gt$ Decrement: Control maintains sequence.
TXI	24	A	-	-	Transfer with Index Incremented	(Index) + Decrement $\rightarrow$ Index; Control $\rightarrow$ m.
TSX	24	B	-	-	Transfer & Set Index	2's complement of location on TSX instruction $\rightarrow$ Index; control $\rightarrow$ m. Use: Address modification by Index; address is reduced by contents of Index. (Subtracting 2's complement is equivalent to addition of original value).

CONVERSION INSTRUCTIONS

CVR	24	C	-	-	Convert by Replacement from Accumulator	(AC) <sub>p,1-35</sub> handled as 6-bit character code representations (right to left). (AC) <sub>30-35</sub> added to instruction address in SR. Contents of this new address $\rightarrow$ SR. (AC) shifted right 6 bits. (SR) <sub>S,1-5</sub> $\rightarrow$ AC <sub>p,1-5</sub> . Count is decreased by 1. Steps are repeated until count equals zero.
CRQ	24	C	-	-	Convert by Replacement from MQ	Similar to above, using MQ. Rotation is left to right.
CAQ	24	C	-	-	Convert by Addition from MQ	(MQ) <sub>S,1-35</sub> handled as 6-bit representations. Same procedure as CRQ above, except (SR) are added to AC <sub>q,p,1-35</sub> .

Uses: CVR replaces a word in BCD form with a word in another BCD form (code conversion). CRQ is used for format control; i.e., to replace leading zeroes with blanks. CAQ is used for decimal to binary and binary to decimal conversion.

5154.1 Instructions of the Type 709 EDPM (Continued)

SENSE INDICATOR INSTRUCTIONS

Accumulator  $\longleftrightarrow$  Sense Indicator (Instruction bits 12-35 not used)

PAI	24	B	-	-	Place Accumulator in Indicators	$(AC)_{p,1-35} \longrightarrow SI_{0-35}$ .
OAI	24	B	-	-	OR Accumulator to Indicators	$(AC)_{p,1-35}$ matched with $(SI)_{0-35}$ . Logical "OR" operation $\rightarrow$ SI. Example: (AC): 001100001100001100000000111111000000 (SI): 000000111111000000001111111100000111 before (SI): 001100111111001100001111111111000111 after
RIA	24	B	-	-	Reset Indicators from Accumulator	$(AC)_{p,1-35}$ matched with $(SI)_{0-35}$ . A "1" bit in AC sets corresponding position in SI to "0". A "0" bit in AC leaves corresponding position in SI unaltered. Examples: (AC): 001100001100001100000000111111000000 (SI): 000000111111000000001111111100000111 before (SI): 000000110011000000001111000000000111 after
IIA	24	B	-	-	Invert Indicators from Accumulator	$(AC)_{p,1-35}$ matched with $(SI)_{0-35}$ . A "1" bit in AC complements corresponding bit position in SI. A "0" bit in AC leaves corresponding position in SI unaltered. Example: (AC): 001100001100001100000000111111000000 (SI): 000000111111000000001111111100000111 before (SI): 001100110011001100001111000011000111 after
PIA	24	B	-	-	Place Indicators in Accumulator	$(SI)_{0-35} \longrightarrow AC_{p,1-35}$ . $0 \longrightarrow AC_{S,q}$ .

Memory  $\longleftrightarrow$  Sense Indicators

LDI	24	B	X	X	Load Indicators	$(m)_{S,1-35} \longrightarrow SI_{0-35}$ .
OSI	24	B	X	X	OR Storage to Indicators	Same as OR Accumulator to Indicators, using (m) instead of (AC).
RIS	24	B	X	X	Reset Indicators from Storage	Same as "Reset Indicators from Accumulator," using (m) instead of (AC)
IIS	24	B	X	X	Invert Indicators from Storage	Same as "Invert Indicators from Accumulator," using (m) instead of (AC).
STI	24	B	X	X	Storage Indicators	$(SI)_{0-35} \longrightarrow m_{S,1-35}$ .

5154.1 Instructions of the Type 709 EDPM (Continued)

SENSE INDICATOR INSTRUCTIONS (continued)

Instruction  $\longleftrightarrow$  Sense Indicator (Instruction bits 18-35 are control or F-field)

Code	Time (us)	Type	Index	Indir.	Instruction	Description
SIL	24	D	-	-	Set Indicators of Left Half	F-field matched with (SI) <sub>0-17</sub> ; logical OR $\rightarrow$ SI.
SIR	24	D	-	-	Set Indicators of Right Half	F-field matched with (SI) <sub>18-35</sub> ; logical OR $\rightarrow$ SR.
RIL	24	D	-	-	Reset Indicators of Left Half	F-field matched with (SI) <sub>0-17</sub> ; a (1) bit in F-field resets corresponding bit position of SI to "0."
RIR	24	D	-	-	Reset Indicators of Right Half	As above, on (SI) <sub>18-35</sub> .
IIL	24	D	-	-	Invert Indicators of Left Half	F-field matched with (SI) <sub>0-17</sub> ; a "1" bit in F-field complements corresponding bit position in SI.
IIR	24	D	-	-	Invert Indicators of Right Half	As above, on (SI) <sub>18-35</sub> .

Transfer and Testing of Sense Indicators

TIO	24	B	X	X	Transfer When Indicators On	(AC) matched with (SI). A "1" bit in AC examines corresponding bit position in SI. If all examined positions of SI are 1's, control $\rightarrow$ m; if not, next instruction in sequence.
TIF	24	B	X	X	Transfer When Indicators Off	As previous instruction; if all examined bits in SI are 0's control $\rightarrow$ m.
ONT	24	B	X	X	On Test for Indicators	(m) matched with (SI), in same manner as two previous instructions. If all examined positions are 1's, skip next instruction.
OFT	24	B	X	X	Off Test for Indicators	As previous instruction; if all examined bit positions are 0's, skip next instruction.
LNT	48	D	-	-	Left Half Indicators ON Test	F-field matched with (SI) <sub>0-17</sub>
RNI	48	D	-	-	Right Half Indicators ON Test	F-field matched with (SI) <sub>18-35</sub>
LFT	48	D	-	-	Left Half Indicators OFF test	F-field matched with (SI) <sub>0-17</sub>
RFT	48	D	-	-	Right Half Indicators OFF Test	F-field matched with (SI) <sub>18-35</sub>

} If all examined bit positions are 1's, skip next instruction

} If all examined bit positions are 0's skip next instruction

5154.1 Instructions of the Type 709 EDPM (Continued)LOGICAL BIT MANIPULATION INSTRUCTIONS

Code	Time (us)	Type	Index	Indir	Instruction	Description
CIM	24	E	*	-	Clear Magnitude	$0 \rightarrow AC_{q,p,1-35}$
COM	24	E	*	-	Complement Magnitude	Complement $AC_{q,p,1-35}$
CHS	24	E	*	-	Change Sign	Change sign of AC
SSP	24	E	*	-	Set Sign Plus	Make sign of AC plus
SSM	24	E	*	-	Set Sign Minus	Make sign of AC minus
ANS	48	B	X	X	AND to Storage	$(AC)_{p,1-35}$ matched with $(m)_{S,1-35}$ . If corresponding bits of BOTH AC and memory are 1's, $1 \rightarrow m$ . If either bit is a "0", $0 \rightarrow m$ .
ANA	36	B	X	X	AND to Accumulator	Same test as above; result $\rightarrow AC_{p,1-35}$ $0 \rightarrow AC_{S,q}$ .
ORS	24	B	X	X	OR to Storage	$(AC)_{p,1-35}$ matched with $(m)_{S,1-35}$ . When corresponding bit of EITHER AC or memory is a "1," $1 \rightarrow m$ . When corresponding bit of BOTH AC and memory is a zero, $0 \rightarrow m$ .
ORA	24	B	X	X	OR to Accumulator	Same test as above; result $\rightarrow AC_{p,1-35}$ $0 \rightarrow AC_{S,q}$ .
ERA	36	B	X	X	Exclusive OR to Accumulator	$(AC)_{p,1-35}$ matched with $(m)_{S,1-35}$ . When corresponding bits of AC and memory are equal, $0 \rightarrow AC$ . When corresponding bits are not equal, $1 \rightarrow AC$ .

\* - Indexing not prohibited, but may cause change in the operation.

5154.1 Instructions of the Type 709 EDPM (Continued)TRANSFER OF CONTROL INSTRUCTIONS

## Unconditional Transfers

Code	Time (us)	Type	Index	Indir	Instruction	Description
TRA	24	B	X	X	Transfer	Control $\rightarrow$ m (unconditional transfer).
HTR	24	B	X	X	Halt and Transfer	EDPM STOP; Manual start causes control $\rightarrow$ m.
TTR	24	B	X	X	Trap Transfer	Control $\rightarrow$ m regardless of Trapping Mode

## Conditional Transfers -- Accumulator

TZE	24	B	X	X	Transfer on Zero	If $(AC)_{q,p,1-35} = 0$ , control $\rightarrow$ m.
TNZ	24	B	X	X	Transfer on No Zero	If $(AC)_{q,p,1-35} \neq 0$ , control $\rightarrow$ m.
TPL	24	B	X	X	Transfer on Plus	If $(AC)_S = 0$ (+), control $\rightarrow$ m.
TMT	24	B	X	X	Transfer on Minus	If $(AC)_S = 1$ (-), control $\rightarrow$ m.
PBT	24	E	*	-	P-Bit Test	If $(AC)_p = 1$ , skip next instruction.
LBT	24	E	*	-	Low Order Bit Test	If $(AC)_{35} = 1$ , skip next instruction.

## Conditional Transfers -- MQ Register

TQP	24	B	X	X	Transfer on MQ Plus	If $(MQ)_S = 1$ , (+), control $\rightarrow$ m.
TLQ	24	B	X	X	Transfer on Low MQ	If $(MQ) < (AC)$ , control $\rightarrow$ m. (+ 0 > - 0).

## Conditional Transfers -- Indicators

TOV	24	B	X	X	Transfer on Overflow	If overflow indicator ON, turn off and control $\rightarrow$ m.
TNO	24	B	X	X	Transfer on No Overflow	If overflow indicator OFF, control $\rightarrow$ m; if ON, turned OFF, but sequence maintained.
TOF	24	B	X	X	Transfer on Overflow in Floating Point	If Floating Point Indicator ON, turn OFF and control $\rightarrow$ m.
TUF	24	B	X	X	Transfer on Underflow in Floating Point	If Floating Point Underflow Indicator ON, turn OFF and control $\rightarrow$ m.
TOU	24	B	X	X	Transfer on Overflow or Underflow in Floating Point	If Floating Point Overflow and/or Underflow Indicator ON, turn both OFF and control $\rightarrow$ m.
DCT	24	E	*	-	Divide Check Test	If Divide Check Indicator is ON, turn OFF and maintain instruction sequence. If OFF, skip next instruction.

5154.1 Instructions of the Type 709 EDPM (Continued)TRANSFER OF CONTROL INSTRUCTIONS (continued)

## Conditional Transfers -- Comparison

Code	Time (us)	Type	Index	Indir.	Instruction	Description
CAS	36	B	X	X	Compare Accumulator With Storage	(AC) <sub>S,1-35</sub> compared with (m) <sub>S,1-35</sub> . If (AC) Greater: Instruction sequence maintained. Equal: Skip next instruction. Less: Skip next two instructions.
LAS	36	B	X	X	Logical Compare, Accumulator With Storage	(AC) <sub>p,1-35</sub> compared with (m) <sub>S,1-35</sub> . Same control as "Compare Accumulator With Storage."
ZET	24	B	X	X	Storage Zero Test	If (m) <sub>1-35</sub> = 0, skip next instruction.
NZT	24	B	X	X	Storage Not Zero Test	If (m) <sub>1-35</sub> ≠ 0, skip next instruction.

MISCELLANEOUS CONTROL INSTRUCTIONS

EIM	24	E	-	-	Enter Trapping Mode	Trapping Indicator and console light turned ON. During operation in Trapping Mode (program debugging): All transfer instructions (except Trap Transfer), if the transfer is made, send the location of the transfer instruction to address part of location 00000 and automatically transfer instruction control to 00001.
LTM	24	E	-	-	Leave Trapping Mode	Trapping Indicator and console light turned OFF. Normal EDPM operation resumed.
ENK	24	E	-	-	Enter Keys	Contents of the 36 panel input switches transferred to MQ register. Switch down: 1 Switch up: 0
HPR	24	B	-	-	Halt and Proceed	Computer STOP. On manual start, instruction sequence is maintained.
NOP	24	B	-	-	No Operation	No instruction executed; sequence maintained.
STR	24	A	-	-	Store Location and Trap	Location of STR instruction plus one replaces address part of location 00000. Control transferred to 00002. (Only positions S, 1 and 2 of the instruction are used).

5154.1 Instructions of the Type 709 EDPM (Continued)

INPUT-OUTPUT INSTRUCTIONS

Data Transfer Instructions

Code	Time (us)	Type	Index	Indir.	Instruction	Description
CPY	Var.	B	X	-	Copy	Transfers one word: (m) ↔ Input/Output, via MQ register. Used only with a drum or cathode ray tube recorder.
CAD	Var.	B	X	-	Copy and Add Logical	Copy instruction, plus "Add and Carry Logical Word" instruction.
RDS	24*	B	X	-	Read Select	Prepare to read from the unit specified by the address portion.
WRS	24*	B	X	-	Write Select	Prepare to write on the unit specified by the address portion.
WEF	24*	B	X	-	Write End-of-File	End-of-file space (gap) and end-of-file character code placed on tape unit specified by address portion.
RLA	36	B	X	X	Reset and Load Channel A	(m) → DSC Control Registers. If specified DSC has been previously selected:
RLB	36	B	X	X	Reset and Load Channel B	Location RL-instruction+1 → Location Reg. (m) <sub>3-17</sub> → Word Count Register
RLC	36	B	X	X	Reset and Load Channel C	(m) <sub>21-35</sub> → Address Register
RLD	36	B	X	X	Reset and Load Channel D	(m) <sub>S</sub> → Trigger A } Triggers are considered "ON" if a "1" bit is placed in them. (m) <sub>1</sub> → Trigger B } (m) <sub>2</sub> → Trigger C }
RLE	36	B	X	X	Reset and Load Channel E	If specified DSC NOT selected: I/O Check Indicator turned ON; Instruction interpreted as No Operation.
RLF	36	B	X	X	Reset and Load Channel F	
LCA	36*	B	X	X	Load Channel A	(m) DSC Control Registers (Conditional).
LCB	36*	B	X	X	Load Channel B	If specified DSC is selected and a control word is needed: Same procedure as "Reset and Load Channel." If a control word is not needed, EDPM waits until it is. (Word Count and Triggers A,B,C govern requirement for another Control Word).
LCC	36*	B	X	X	Load Channel C	
LCD	36*	B	X	X	Load Channel D	
LCE	36*	B	X	X	Load Channel E	
LCF	36*	B	X	X	Load Channel F	
SCA	24	B	X	X	Store Channel A	Zero → m, then (DSC Control Registers) → m.
SCB	24	B	X	X	Store Channel B	(Address Register) → m <sub>21-35</sub>
SCC	24	B	X	X	Store Channel C	(Location Register) → m <sub>3-17</sub>
SCD	24	B	X	X	Store Channel D	(Triggers A,B,C) → M <sub>S,1,2</sub> if not zero.
SCE	24	B	X	X	Store Channel E	If DSC Control Registers are being changed, EDPM waits with "Store Channel -" until change has been completed.
SCF	24	B	X	X	Store Channel F	

5154.1 Instructions of the Type 709 EDPM (Continued)INPUT-OUTPUT INSTRUCTIONS (Continued)

## Input-Output Control and Test Instructions

Code	Time (us)	Type	Index	Indir.	Instruction	Description
BTT	24	E	X	-	Beginning of Tape Test	Beginning-of-tape Indicator for selected DSC is tested. If ON: Turn off, maintain instruction sequence. If OFF: Skip next instruction.
EIT	24	E	X	-	End of Tape Test	End-of-Tape Indicator for selected DSC is tested. If ON: Turn off, maintain instruction sequence. If OFF: Skip next instruction. (Address of BTT and EIT instructions specify particular DSC involved).
IOT	24	E	-	-	Input-Output Test	Input-Output Indicator (Address 00005) tested. If ON: Turn off, maintain instruction sequence. If OFF: Skip next instruction.
TAO ...	24	B	X	X	Transfer on DSC A in Operation	If selected DSC is in operation, control → m. If not in operation, maintain sequence. Six instructions in this group.
TFO	24	B	X	X	Transfer on DSC F in Operation	
TAN ...	24	B	X	X	Transfer on DSC A Not in Operation	If selected DSC is not in operation, control → m. If in operation, maintain sequence. Six instructions in this group.
TFN	24	B	X	X	Transfer on DSC F Not in Operation	
TAR ...	24	B	X	X	Transfer on DSC A Redundancy Check	Test Tape-Check Indicator for selected DSC. If ON: Turn OFF, control → m. If OFF: Maintain instruction sequence Six instructions in this group.
TFR	24	B	X	X	Transfer on DSC F Redundancy Check	
TAF ...	24	B	X	X	Transfer on DSC A End-of-File	Test End-of-File Indicator for selected DSC. If ON: Turn OFF, control → m. If OFF: Maintain instruction sequence. Six instructions in this group.
TFF	24	B	X	X	Transfer on DSC F End-of-File	
BST	24	B	X	-	Backspace Tape	Specified tape is backspaced one record.
BSF	24	B	X	-	Backspace File	Specified tape is backspaced until end-of-file indicator or load point is reached.
REW	36	B	X	-	Rewind	Specified tape is rewound to starting (load) point.

5154.1 Instructions of the Type 709 EDPM (Continued)

INPUT-OUTPUT INSTRUCTIONS (Continued)

Sense Instructions

Code	Time (us)	Type	Index	Indir.	Instruction	Description
PSE	24	E	-	-	Plus Sense	<p>Console sense switches (lights) and punch or printer hubs are tested and manipulated, according to address part of PSE instruction:</p> <p>0140 All Sense Lights turned OFF.</p> <p>0141 } to } Specified Sense Light turned ON. 0144 }</p> <p>0161 Corresponding Sense Switch tested. to } If ON: Skip next instruction. 0166 } If OFF: Maintain sequence</p> <p>1341 } Impulse sent to exit hub of punch 1342 } control panel of DSC A. (3341-2 and 5341-2 same for DSC C &amp; E)</p> <p>1360 } Entry hub of printer control panel of 3360 } DSC A,C,E is tested. 5360 } If impulse present: Skip next instruction If no impulse: maintain sequence</p> <p>1361 } Impulse sent to exit hub of printer to } control panel of DSC A. 1372 } 3361-72, 5361-72 same for DSC C, E.</p>
MSE	24	E	-	-	Minus Sense	<p>Console Sense Lights are tested; address (0141 to 0144) determines which one. If ON: Turn OFF, skip next instruction. If OFF: Maintain instruction sequence</p>

Sense switches are equivalent to Alteration Switches on the 702/705 and similar to Breakpoint Switches on Univac.

5412.2 Instructions of the DATAmatic 1000 EDPM.ARITHMETIC INSTRUCTIONS

All operands and results are handled as 11-digit plus sign numeric words.

Code		Cycles	Instruction	Description
ADD	1	8	Add	$(A) + (B) \rightarrow C$ . If overflow, control $\rightarrow$ 1988 (automatic).
SUB	1	8	Subtract	$(A) - (B) \rightarrow C$ . If overflow, control $\rightarrow$ 1988 (automatic).
MUL	1	35	Multiply	$(A) \times (B) \rightarrow C$ (rounded high-order half of 22-digit product) $\rightarrow$ "Remainder" register (1995) (low-order half).
DIV	1	31- 130	Divide	$(B) \div (A) \rightarrow C$ (most significant 11 digits) $\rightarrow$ "Remainder" register (1995) (remainder)
SST	1	7	Substitute	Extract (A) and (C) into C. If a bit of B is "1", insert corresponding bit from A; If a bit of B is "0," preserve the cor- responding bit in C. Extraction is bit by bit, but may operate on digits or characters by proper setup of the "extractor" word in B.

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued).SHIFT INSTRUCTIONS

In these instructions, A and C denote high-speed memory locations and B denotes the amount of the shift. (A) is always the word shifted; (A) remains unchanged.

Code				
SLP	1	$7+B$	Shift Left, Pre-serving Sign	(A) shifted left number of digit spaces designated by B, keeping the sign unchanged and unshifted. Numeric zeroes fill in at right. Result $\rightarrow$ C. $0 \leq B \leq 12$ .
SRP	1	$7+B$	Shift Right, Pre-serving Sign	(A) shifted right number of digit positions indicated by B, keeping sign unchanged and unshifted. Numeric zeroes fill in at left. Result $\rightarrow$ C. $0 \leq B \leq 12$ .
SLA	1	$7+\frac{3}{2}B$	Shift Left, Alphabetic	(A) shifted left the number of alphabetic spaces designated by B. Result $\rightarrow$ C. $0 \leq B \leq 8$ . (If $B=0$ , one two-bit shift occurs).
SRA	1	$7+\frac{3}{2}B$	Shift Right, Alphabetic	As above, shifting right.
SLN	1	$7+B$	Shift Left Numeric, Sign Included	As SLP, including sign position. $0 \leq B \leq 12$ .
SRN	1	$7+B$	Shift Right Numeric, Sign Included	As SRP, including sign position. $0 \leq B \leq 12$ .

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)BUFFER-MEMORY TRANSFER INSTRUCTIONS

Instruction Word Format No. 2. Address A is the first memory location from or into which words are transferred.  $B_1$  and  $B_2$  must be in the range 1-32. Address C is used for the address of a subsequence call, if applicable.

Each basic order can apply to either the A or B section of the buffer. The third letter of the mnemonic operation code designates one of the four possible conditions:

- A - Transfer from the A section of the Input Buffer.
- B - Transfer from the B section of the Input Buffer.
- S - Transfer from the same section of the buffer to which the last previous connection was made (by a Transfer In, Double Transfer and Select, Transfer and Select, or Read instruction).
- D - Transfer from the opposite section of the buffer to which the last previous connection was made.

## Transfer In Instructions:

Code		Cycles	Instruction	Description
TIA	2	$4+B_2$	Transfer $B_2$ Words from Buffer to Memory With Inter- lock	If buffer is still being loaded from tape as result of the last previous Read instruction, hold up via the interlock until the Read is completed.  If or when interlock is released, transfer $B_2$ words from Input buffer to memory, beginning at memory Address A.  If $B_2$ is greater than number of words in buffer, fill out with Sequence Change Instruction words (A and B address void, 1985 in C).  Examine words transferred for sentinels. Store first sentinel in Sentinel Register (1997). If no sentinel present, store a Pass Instruction.  Ignore $B_1$ .
TIB	2	$4+B_2$		
TIS	2	$4+B_2$		
TID	2	$4+B_2$		
TBA	2	$4+B_2$	Transfer $B_2$ Words	Same as previous instructions, ignoring status of interlock. (This permits transferring Input Buffer contents to memory after a Read Instruction but before the tape transfer actually begins).
TBB	2	$4+B_2$	From Buffer to	
TBS	2	$4+B_2$	Memory Without	
TBD	2	$4+B_2$	Interlock	

Upon completing these instructions, the next is taken from Address C (subsequence call).

An error (erroneous weight count) in any word transferred from the buffer results in an automatic subsequence call to address 1985.

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)BUFFER-MEMORY TRANSFER INSTRUCTIONS (Continued)

Double Transfer and Selection Instructions:

Code				
DTA	2	$6+B_2$	Transfer to Output Buffer and From Input Buffer With Selection and Interlock	<ol style="list-style-type: none"> <li>1. Transfer <math>B_2</math> words, starting at memory address A, to Output Buffer Storage. Do not examine overflow for sentinels.</li> <li>2. Transfer <math>B_2</math> words from Input Buffer to memory, beginning at address A. <ol style="list-style-type: none"> <li>a. Delay the transfer, via the interlock, until the buffer has been loaded as a result of the last previous Read Instruction.</li> <li>b. Store first sentinel in Sentinel Register; if none, store Pass Order.</li> <li>c. If B is greater than number of words in buffer, fill out with Sequence Change Instruction words.</li> </ol> </li> <li>3. Extract one digit from the <math>B_1</math> word in the Input Buffer and add to the units digit of Address C (with carry permitted). The digit extracted depends on the word in the Extractor Register (previously loaded).</li> <li>4. Store the original instruction (DTA, DTB, DTS or DTD) in Select Order Register (1994).</li> <li>5. Make subsequence call to instruction in Address C location modified by Step 3.</li> </ol>
DTB	2	$6+B_2$		
DTS	2	$6+B_2$		
DTD	2	$6+B_2$		
DBA	2	$6+B_2$	Transfer to Output Buffer and From Input Buffer With Selection and Without Interlock	Same as the previous instructions, except the status of the interlock is ignored in Step 2a. The transfer is made at once.
DBB	2	$6+B_2$		
DBS	2	$6+B_2$		
DBD	2	$6+B_2$		
Transfer and Select Instructions:				
TSA	2	$6+B_2$	Transfer from Input Buffer With Selection and Interlock	Same as DTA,...,DTD instructions, omitting Step 1. (Nothing is transferred from memory to the Output Buffer).
TSB	2	$6+B_2$		
TSS	2	$6+B_2$		
TSD	2	$6+B_2$		
BSA	2	$6+B_2$	Transfer from Input Buffer With Selection and Without Interlock	Same as DBA,...,DBD instructions, omitting Step 1. (Nothing is transferred from memory to the Output Buffer).
BSB	2	$6+B_2$		
BSS	2	$6+B_2$		
BSD	2	$6+B_2$		

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)

BUFFER-MEMORY TRANSFER INSTRUCTIONS (Continued)

Transfer Out Instructions:

Code		Cycles	Instruction	Description
TXO	2	$4 + B_2$	Transfer $B_2$ Words From Memory to Output Buffer	Transfer to the Output Buffer $B_2$ words beginning at Address A. If Output Buffer overflows, examine overflow words for sentinels. a. If present, store first sentinel word in Sentinel Register (1997). b. If no sentinel in overflow, store a Pass Order in Sentinel Register. Address C may be used for a subsequence call if desired. $1 \leq B_2 \leq 32$ . $B_1$ is not used.

INTERNAL MEMORY TRANSFER INSTRUCTIONS

TXI	3	$3 + 2C_2$	Transfer Internally	$C_2$ words beginning at address A to new location beginning at Address B. $1 \leq C_2 \leq 32$ . Range of address specified by A and C or B and C should not include both 0999 and 1000 or any special registers.
TTX	1	5	Twin Transfer	Contents of Select Order Register $\rightarrow$ A and (B) $\rightarrow$ C. This order is used with Double Transfer and Select and Transfer and Select in sorting operations.
TXS	1	5	Transfer and Subsequence Call	(A) $\rightarrow$ B; sequence call $\rightarrow$ C. Neither B or C should be 1990.

COMPARISON INSTRUCTIONS

Numeric and Alphabetic Comparison Instructions:

LCN	1	7	Numeric "Less Than" Comparison	If $(A) \leq (B)$ , control $\rightarrow$ C. Otherwise, next instruction in sequence.
ICN	1	7	Numeric Inequality Comparison	If $(A) \neq (B)$ , control $\rightarrow$ C. Otherwise, next instruction in sequence. In both LCN and ICN, $+0 = -0$ .
LCA	1	7	Alphabetic "Less Than" Comparison	If $(A) \leq (B)$ , control $\rightarrow$ C. Words are treated as alphabetic.
ICA	1	7	Alphabetic Inequality Comparison	If $(A) \neq (B)$ , control $\rightarrow$ C.

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)TAPE CONTROL INSTRUCTIONS

In the table blocks, Words 1 and 2 occupy the Key Channel; words 3-62, the Satellite Channels. Either may be in a Read or Write status independent of the condition of the other.

In tape instructions, the three addresses have these meanings:

Address A: Units and tens digits indicate the tape unit (00 to 99).

Address B: Can be used for changing Sequence Counter (transfer of control).

Address C: Can be used for making a subsequence call.

Code		* Cycles	Instruction	Description
RFA	1	7	Read Tape Forward - A	Read one tape block forward into Section A of the Input Buffer.
RFB	1	7	Read Tape Forward - B	Read one tape block forward into Section B of the Input Buffer.
RFD	1	7	Read Tape Forward - Alternate	Read one tape block forward into section of Input Buffer <u>not</u> read into by last previous read order (i.e., change sections). <u>Connect the section not read into memory.</u>
RBA	1	7	Read Tape Backward-A	Read one tape block backward into Section A of the Input Buffer.
RBB	1	7	Read Tape Backward-B	Read one tape block backward into Section B of the Input Buffer.
RBD	1	7	Read Tape Backward - Alternate	Same as RFD, reading tape backward
RFK	1	7	Read Tape Forward - Key Channel	Put Key Channel of designated tape unit in Read status and Satellite Channels in Write status. Read Key Channel of one block, forward, into Section B of Input Buffer.
RBK	1	7	Read Tape Backward - Key Channel	Same as previous instruction, reading tape backward.
WFA	1	7	Write Tape Forward	Place Key and Satellite Channels of designated tape unit in write status. Fill Output Buffer to 62 words, if not full. Fillers are sentinel Sequence Change instructions. <u>Write one block of 62 words on tape.</u>
WFP	1	7	Write Tape Forward, Except on Key Channel	Place Key Channel in read status and Satellite Channels in write status. Fill Output Buffer to 62 words, if necessary. <u>Write one block on Satellite Channels only.</u>

\* - Computer time necessary to execute instructions. Tape time depends on state of tape unit in relation to action required by current instruction.

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)TAPE CONTROL INSTRUCTIONS (Continued)

## Search Instructions:

General comments: Input Buffer A is organized to include ten 2-word sections addressable by the digits 0 through 9. The "search" instructions following permit reading the Key Channel only of a tape block into the section having the same number as the units digit of the tape address. Up to ten separate tape units may be "searched" simultaneously (a separate instruction is required for each), provided the units digit of the tape address differs for each one. (Thus, tape units 12 and 42 cannot be searched simultaneously, but 11 and 12 can).

Code	Type	Cyc.	Instruction	Description
SFR	1	7	Search Forward, Reading	Place Key and Satellite Channels in read status. Read Key Channel only of one block, forward, into matching section of Input Buffer A.
SFW	1	7	Search Forward, Writing	Place Key Channel in read status and Satellite Channels in write status. Read Key Channel only of one block, forward, into matching section of Input Buffer A.
SBR	1	7	Search Backward, Reading	Same as SFR above, except read backwards one block.
SBW	1	7	Search Backward, Writing	Same as SFW above, except read backwards one block.
REW	1	7	Rewind Tape Unit	Rewind the specified tape unit. Upon completion, tape is positioned for reading or writing on the first block.

KEY COMPARISON INSTRUCTIONS

FKC	1		First Key Comparison	<ol style="list-style-type: none"> <li>1. Move the <u>first</u> word in the Input Buffer A section designated by the <u>units</u> digit of Address A to Address A. (I.e., if A is "0936," first word of section 6 is moved to 0936).</li> <li>2. Compare (A) and (B). If (A) <math>\geq</math> (B), control <math>\rightarrow</math> C. Comparison is alphabetic, not numeric.</li> <li>3. If a previous "search" is not completed, action is delayed by interlock until Key Channels have been read in. Only the section involved is interlocked; others may be used if not interlocked.</li> <li>4. Interlocking occurs from a "search" instruction; if not interlocked, (whatever is in section designated is used.</li> </ol>
SKC	1		Second Key Comparison	Identical with above, except the <u>second</u> word in the designated section of Input Buffer A is transferred to Address A. (Step 1)

5412.2 Instructions of the DATAmatic 1000 EDPM (Continued)PRINT INSTRUCTIONS

Code	Type	Cyc.	Instruction	Description
PRA	1	6	Print Alphabetic	(A) printed as 8 alphabetic characters on console typewriter. B may be used to change Sequence Register. C may be used as subsequence address.
PRN	1	6	Print Numeric	(A) printed as 12 digits on console typewriter. B and C may be used as in PRA.

CONTROL INSTRUCTIONS

PSS	1	6	Pass	No operation. Next instruction taken from Sequence Register. Addresses A, B, C immaterial.
OST	1	5	Optional Stop	Stop, or not, depending on console switch setting and (B). (B) is a 12-digit word of numeric 0's and 1's. The optional stop depends on the particular (B) and the setting of the switches. (B) are numbered 1 to 12, from left to right. If the digit in position 12 is a "1," STOP, unconditionally, regardless of switch settings. If positions 12 and 11 are both "0," setting of Optional Stop Switch is next controlling factor: If, set to "Stop," STOP. If set to "Proceed," PROCEED. If set to "Compare," STOP if any of the digits 1 through 10 is a "1" <u>and</u> corresponding Breakpoint switch is ON; if none is on, PROCEED.
SCS	1	6	Sequence Change	Change Sequence Register to Address B, Subsequence Call to Address C. A is disregarded but must be numeric.
BAR	1	7	Branch and Return	Subsequence Call to Address C. Change Sequence Register to Address B. Store in Address A an order which can be used later to change Sequence Register back to its current reading. (This storage occurs after Sequence Register has been incremented by "1.")

8110 BASIC PRINCIPLES OF FILE MAINTENANCE ON A MAGNETIC TAPE PROCESSOR

Univac I--the first of the modern electronic data processing machines adapted to the processing of business data--and all of its large-scale successors and competitors have magnetic tapes as the major medium for the storage of file information. Some of the medium-scale machines use only punch cards for this purpose--such as the IBM Type 650 in its first two years of availability--but the current trend is for all general-purpose business data processors to provide, at some stage of development, magnetic tape storage; this is certainly true in the above-\$100,000 purchase price range.

Magnetic tapes constitute the only form of mass information storage now available. Access to this type of storage is sequential; that is, every item in the file must be passed over to reach any specific one desired. The other major type of storage device now being developed for EDPM use is the random access storage, of which the first announced versions are the IBM RAMAC disc files. This type of storage is differentiated from the sequential magnetic tape files in that access to any given item is essentially of the same time; it is not necessary to pass over unwanted information in the manner required with tapes.

The discussion which follows is concerned solely with the characteristics of handling files which are on magnetic tapes.

8111 DEFINITION OF TERMS USED

There are no uniform standards in the computer industry, or among users, for the terminology applied to the various factors comprising a magnetic tape file of information. Those used here are not proposed as such standards, but in this section the various elements will be referred to specifically, as defined below; in other sections some of the alternate terms mentioned are used. The basic reason is that lack of standardization makes it desirable to know not only our own terminology, where it is definitive, but also the terminology which is used by competitors for the same factor.

8111.1. File. A file is one or more magnetic tapes containing information pertaining to some business data processing function. As such, it is very variable, depending upon what information is included. For example, one EDPM user may have a file containing information necessary to process payrolls and another containing personnel information, much of which may be repeated in the payroll file. Another user might combine the two into one file. Any magnetic tape input to or output from an EDPM is a file, because they all contain information pertaining to some facet of business data. A master file is generally restricted to a major file of information which is used over and over again, the pieces of information being being changed as necessary to reflect current conditions.

8111.2 Item. An item is all the information pertaining to a given major entry in the file. The file itself is sequenced according to some identifying information which distinguishes each item from all others in the file. For example, the item may be all the constant (or reasonably so) information necessary to calculate an employee's weekly pay. Name, clock number

8111.2 (Continued)

hourly pay rate, department number, year-to-date gross earnings, number of departments or income tax table classification, various types of deductions, etc. Normally, this type of file is sequenced on clock number; the file itself consists of one such item for every employee of the company. In IBM terminology, an Item is usually referred to as a record; their term is also used with other meanings.

8111.3 Data Element. A data element is one of the pieces of information which constitute an item--clock number, employee name, etc. It can be considered as a complete and independent unit of information.

8111.4 Magnetic Tape Record. A magnetic tape record is the number of characters or words of information written between successive gaps (no information) on tape. It may be a fixed number of characters or words, always constant for a given equipment, or the number may vary, not only between files but for successive tape records within one file. In Univac, a tape record is always 60 words or 720 characters, and is called a block; in 705 usage, several records are normally combined into one grouped record. Other terms are also encountered.

8111.5 Fixed and Variable Word Lengths. In some EDPs--such as Univac--all operations are performed on a specific number of characters, called a word. Several short data elements may be combined into one word, with extraction being required to separate out any specific one needed. In other EDPs--such as the 705--operations are performed serially on one character after another, separate provisions (other than a constant number of characters) terminating the individual operations; i.e., an instruction may work on one character or 100. For practical purposes, a word and a data element can be considered as identical in a variable word length machine; they are not in a fixed word length equipment, in which a data element may be only part of a word or consist of several words.

8112 CHARACTERISTICS AND IDENTIFICATION OF BUSINESS ITEMS

The basic unit of information upon which an EDPM-- or any other type of data processor--operates is an item. It is difficult to imagine any sort of productive work on a file which does not require searching it, in some manner, for one or more specific items, and performing some sort of operations on one or more of the data elements which are, or can be, a part of the item. For this reason, it is necessary to understand clearly the general nature of business items and the various means which may be used to distinguish them one from another. This section is devoted to this subject.

8112.1 General Variability of Business Items. It is to be expected that the number of words (or characters) in items from different files would not in general be the same; each must be long enough to contain the information necessary for its specific purpose. It is equally true that, within a single file, items are of different lengths, and frequently the amount of the variation is extremely large. In other words, the typical business file does not consist of items of constant length. To be sure, such files do exist; in some, the nature of the information to be represented is such that every item contains an identical number of words. In others--and this is probably closer to the truth--they are set up in constant length to meet the requirements of some data processing system. This is an artificial consideration which is not inherent in the nature of the information to be represented.

Example of files which contain items inherently variable in length are easy to enumerate; those which contain items of constant, or nearly so, length, much more difficult. The most obvious case of the latter which come to mind is a name and address file used for magazine subscription fulfillment and mailing operations; here almost every item has the same number of data elements--the basic variation occurs in coded information representing conditions which may be useful for promotional purposes.

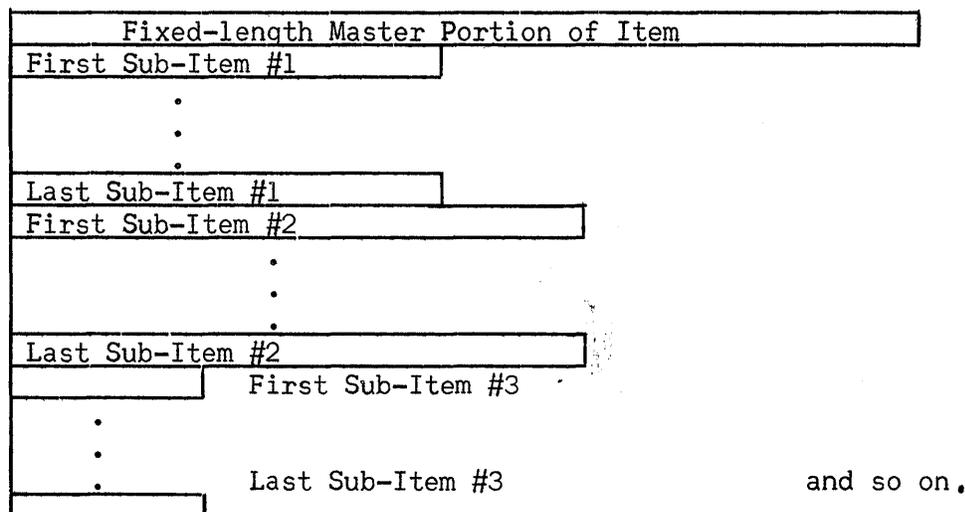
But consider the possibilities for variation in item length. A master insurance policy file contains items of lengths varying with the type of policy, with outstanding loans, with delinquencies in payment, etc. A central inventory file has items whose lengths may vary widely depending upon the number of warehouses carrying the item, the number of back orders attached to the item, the number of purchases due in, etc. A department store charge account file may have provisions for a detailed listing of payments made, of charges incurred, and of other persons authorized to make charges against a basic account. In other words, the mythical "typical" business file is characterized by a marked degree of variation in the length (i.e., amount of information) of items within the file.

In general, a business file consists of a number of items each of which consists of a fixed master portion comprising data elements common to all, or practically all, items, and to which are added or appended a variable number of other data elements which may range in number from none to several hundred.

8112.1 (Continued)

There is nothing novel or revolutionary in this statement. We are all familiar with Kardex files, in which the fixed master portion comprises the visible information, plus possibly additional data elements, and in which the number of line entries representing various types of transactions which have been posted varies from nothing to the capacity of several cards. In punch card operations, the concept of "master" and "detail" cards is also commonplace; there may be several kinds of detail cards in one file, the number often being determined by selector capacity to distinguish them from each other and to provide for different kinds of treatment. Also commonplace is the break-up of closely related information into several types of files, usually because of the limited processing capabilities of the machines; thus a payroll master card may be a completely different punch card from a year-to-date earnings cards, which carries cumulative totals or earnings and deductions. It should be noted that the separation into two files is not inherent in the data, but is made because of the restrictions of processing equipment

The typical business item--if there be such an animal-- then may be considered as a fixed master portion followed by a variable number of detail sub-items, thus:



The various types of sub-items each consist, normally, of a specific number of data elements comprising a specific number of words or characters. In general, each type is of a length different from the others, but at any rate its length is set by the amount of information which it contains. Any given item consists of the master portion (always) plus anything from none to some maximum number of each of the possible sub-items the sub-items can occur, of course, in all conceivable combinations.

This sort of arrangement is not the most complicated which can exist. It is perfectly possible for one or more of the sub-items to be of variable length--in other words, to consist of some sub-master portion plus a variable number of sub-sub-items. The principal is identical to the one-stage process depicted above; the complexity of processing may

8112.1 (Continued)

naturally be increased as "details are piled on details are piled on details." But from the standpoint of what is necessary to represent the information which must be contained within the item, there is no reason we can't have sub-sub-sub-items within sub-sub-items within sub-items within items. Within files, of course.

8112.2 Arrangement or Sequencing of Items Within Files. A file may consist of from a very few to a large number of items--for example, the basic wage file maintained by the Social Security Board exceeds 120,000,000 separate items or accounts. As soon as the number exceeds a handful, it becomes convenient, if not essential, to arrangement the items comprising the file in some logical sequence. The particular sequence into which any given file is ordered is that which is most appropriate for its use; it is quite common for two different files to contain essentially the same information (data elements), but to be maintained in two different sequences for some processing consideration.

The criterion or factor which determines the position of any given item within the file is a data element in the fixed master portion, usually referred to as an index or control number. Frequently it is created and placed into the item as a data element used largely for sequencing purposes; such index numbers are employee clock numbers, social security account numbers, insurance policy numbers, etc., are of this nature. Many of them are strictly numeric; one major reason for their prevalence is that sorting (i.e., sequencing) of numeric index numbers on punch card equipment is faster than on alphabetic numbers. Many such numbers are of course, alphabetic or alphanumeric--names, part numbers, stock numbers, etc.

The items within the file are placed in order according to the index number associated with each. For this purpose, the digits, letters and special characters which comprise the index are ordered in a specific sequence, which frequently is related to a sequence recognized by some of the processing equipment. The sequence of alphanumeric characters in a Remington-Rand collator, for example, is different from that in an IBM collator, and consequently a file with identical index numbers will be in a different order if Remington-Rand equipment is used from what it will be with IBM hardware. The specific sequence is somewhat immaterial, so long as the processing equipment handles it correctly and always works with the same ordering; communication between two different types of processing installations may be somewhat complicated. The number of characters in an index number may vary greatly; they range from a normal minimum of four or five up to 22 or 24 for a name-sequence file and occasionally exceed this number (i.e., the standard descriptive number used by the bearing industry is about 40 digits).

In some files, the nature of the establishment of index numbers precludes the possibility of duplication; this is usually true of clock numbers, insurance policy numbers, and other types of identifying numbers whose assignment is controlled by the user. This is not always the case, and duplication may occur; i.e., in a file sequenced on names. In a file where index numbers of two different items may be identical, there is almost always a secondary index number which is used to arrange the items in a specific order within the prime index. This may be carried through

## 8112.2 (Continued)

several successive stages to provide a specific ordering where primary, secondary, tertiary, etc., index numbers may be identical. Thus, in a geo-alpha file, the principal sequencing may be on state, the secondary on city within state, the third on postal zone within city, then on name within postal zone, then on street address and finally on apartment number--to account for two John Smiths living in the same apartment building in New York 56, N. Y.

8112.21 Two Basic Principles of File Sequencing. Because they are fundamental in the handling of magnetic tape files, it may be well to state two basic principles of file sequencing.

- (1) Every Index number which is, or can be, used to determine the proper ordering of the items in a file must be contained in the master portion of the item and each is assigned a specific and unchanging level or stage in relation to the other index numbers.
- (2) If two different items have identical index numbers through all levels or stages which have been established, then the order of the two items in the file is immaterial. Because this condition is, in general, to be avoided, this statement may be considered tantamount to saying sufficient levels of index numbers must be established to assure that every item in the file differs from any other in at least on index number.

8112.22 Sequencing of Sub-Items. Because an item may consist of a variable number of sub-items associated with a master portion, it is necessary to provide some method of sequencing them in some orderly manner following (usually) the master portion. The same principles and concepts used to order the items themselves are applicable here; the sub-items, and their sub-sub-items, etc., are ordered in some logical manner on index numbers (data elements) which are a part of the information in the sub-items. Thus the entire file is sequenced or ordered in some more or less logical fashion, according to clear-cut, exhaustive and unchanging rules.

8112.3 Identification of Items. Because items frequently are of variable length, it is necessary not only to know the order they are in, but also to have some means of ascertaining when one item has been passed and the next one reached. In a manually-maintained file, such as a Kardex file, this is relatively simple; each item is a separate pocket, even though there may be several cards. In a punch card file, an item may or may not be contained in a single card; usually it isn't. There are numerous conventional techniques of identifying the various types of cards by control punches which differ for each sub-item and for the master portion. Further, for master portion or sub-items which may exceed the capacity of a single card, it is common practice to number the cards in sequence, from "1" up, in addition to the control punch, to assure their proper ordering. Such control punches and numeric codes become a part of the sequencing criteria, with a specific relationship to other index numbers; they are among the data elements contained in the master portion or sub-items of an item, although strictly speaking they are not inherent in the information necessary to be carried with the item. They are data elements

## 8112.3 (Continued)

artificially introduced because of characteristics of the processing system, not because of the inherent information requirements of the file. Similar examples can be found with other types of data processors, such as the notches in an addressograph plate.

The foregoing may be summarized briefly: All business files are ordered or sequenced in some logical manner, based upon a definite number of index or control data elements which are a part of the items, the index numbers having a specific order of importance. In addition, the master portion and the various sub-items which comprise the item are identified, one from the other, in some unique manner. The number of indices upon which a file can be sequenced, the nature of the control symbols identifying the various sub-items which comprise an item, and the amount of information which can be associated with each often is predicated upon the characteristics of the processing system used.

EDPMs handle business information in the same sort of files as the more conventional types of data processors. Like them, the files must be sequenced in some orderly and predictable manner, the various pieces of an item must be identified, and the equipment must be able to find any given item, or portion thereof. The basic principles of magnetic tape files are identical with those discussed here; the techniques vary because the characteristics of the data processing equipment differ from those used in the past. In the next section, consideration will be given to some of the methods which can be used to establish magnetic tape files; following that, the implications of the normal means advocated for working with the file will be investigated.

8113 EDPM REPRESENTATION AND IDENTIFICATION OF BUSINESS ITEMS

The basic logic for establishing the format for a file of variable-length items in an EDPM magnetic tape system is identical with that for manual, punch card, or any other sort of file. Similarly, the logical operations involved in handling a manual file are identical with those for handling a tape file. The techniques vary because the nature of the processor is totally different, but exactly the same operations and conditions must be accounted for in both systems.

8113.1 Characteristics of Magnetic Tapes and Processors Affecting File Formats.

Like any other type of data processors, EDPMs and magnetic tapes have characteristics affecting the format of files to be processed. One of the principal advantages of an EDPM, as compared with more conventional processors, is its extreme speed and capacity for storing information; rather than being limited to a few counters (as adding machine) or a few score (as a punch card tabulator or calculator), it has the ability to retain a large number of characters--20,000 in the 705, 24,000 in Univac II, and options in both equipments for still greater storage. They have great flexibility in their aptitudes for transferring information around inside the machine; among other things, this means that an index number doesn't always have to be "in card columns 10-24."

The magnetic tapes have special characteristics, also. Just as Kardex or punch cards, they are nothing more than a medium for storing information. At some point, the information contained in them is transferred into the "memory" of an EDPM, where it is an exact character-by-character image of the tape record; there is an exact counterpart in the transfer of information contained as holes in a punch card into the counters, relays, selectors, etc., of some type of punch card equipment. The EDPM differs, in one important respect, in its ability to juggle this "tape image" around as it sees fit.

Now consider the tape itself, and imagine a file of information stored on it. Physically, it is a counterpart of a Kardex cabinet or a box of punch cards. It has one important difference: It's a physical entity which can't "lose" a piece of itself. In this respect, it is unlike the other two, in which it is possible to lose a card. It has other direct comparisons. Just as a Kardex or punch card file can overflow a box as new items are added, the tape can get "too full;" the operation of taking a handful of cards and inserting them at the beginning of the next "box" has its magnetic corollary in taking the end of one tape's information and writing it on the beginning of another.

But a tape isn't packed solidly with information from beginning to end. It stores its information in records or blocks which are separated by a blank space; the reason for this is that present-day memories, at least, aren't big enough to hold everything we could pack on a reel of tape, so the information stored is broken up into pieces of a size we can handle. These pieces are called blocks or records, and their size varies with different EDPMs. Compared with the information capacity of a punch card, a block is much bigger; compared with the capacity of a Kardex card, it is usually smaller. In a punch card file, information is added and deleted by inserting and removing cards; sometimes by replacing with a new

## 8113.1 (Continued)

card. The same thing occurs in a tape file; the big difference is that the size of the block is bigger, and the additions and deletions often mean "expanding" or "compressing" an existing block rather than creating one of completely new information. But the principle is the same. And, just as adding enough punch cards eventually necessitates reassembling the boxes holding the file, so adding enough information to a tape means reassembling them.

From the standpoint of file formats, of course, one of the salient differences distinguishing them from punch card operations is the fact that pieces of tape can't get lost; the processor doesn't need redundant information to identify successive sub-items and associate them with the master portion; and, finally, the basic processing capacity and information holding ability of the EDPM are great enough to remove the limits (at least for practical purposes) of how much information any single item can contain.

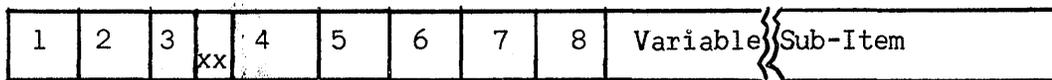
8113.2 Tape Formats for Constant Length Items. If the items of a file are of constant, or practically constant, length, they can be established in a format of a specific number of characters or words; all items are then of the same length. Each data element is in a specific location or "field" within the item. This is a direct counterpart of the typical punch card layout, with the exception that the number of characters can be much greater than 80 or 90. The size of the item does not need to be an even sub-multiple of the tape block or record; a 17-word item in a Univac causes no difficulty. The fact that some items will be split between two tape records does not mean that an identifying index number must be repeated; the EDPM isn't going to "lose" a tape record and is perfectly capable of holding two or three tape blocks in its memory. This does not mean that the size of the tape block can be ignored completely in determining the most appropriate format; other functions of which the EDPM may be capable should be considered. For example, the multiple-tape "search" feature of the DATAmatic system may make it advisable to place one item in each tape block, even though the entire 62 words are not required. Similarly, limitations of the high-speed memory transfer in the IBM 705 may make it advisable to group the largest number of full items which can be contained in 1,022 characters, and not attempt to write the maximum size tape block possible. These considerations, of course, are caused by characteristics of the EDPMs and not of the tapes themselves.

8113.3 Tape Formats for Variable Length Items. More typically, the establishment of an item format for a magnetic tape processor must recognize the existence of a variable length item. Because the EDPM has no advance knowledge of the length of any individual item, or of the types of sub-items or the number of them that may be included, it is necessary to provide, in some manner for three factors: (1) Where does the item begin in memory? (2) How long (i.e., how many characters or words) is it? (3) What sub-item does it include? The answers to these three questions must be contained within the item itself, or must be capable of determination from something in the item. Thus, because the tape format is nothing but a representation of the initial transfer of the tape block into

the central processor, it follows that the tape format must contain the necessary information to answer these questions. The following paragraphs describe some of the techniques which have been developed or proposed.

First, however, it may be well to state specifically a factor which has been implied in the previous sections: The development of an item format for a magnetic tape file is identical with that of the layout of input and output areas in memory.

8113.31 Word or Character Counts Contained in the Item. One method of keeping track of variable length items is to include as a part of the master portion a data element which is in the number of words or characters in the item. This is placed in a fixed and constant position within the master portion. As an example, consider an item format development for a Univac tape file. It has a fixed master portion of eight words, and may have up to 90 words or variable information appended to it; thus it never exceeds 100 words in length. Two digits then can express the length of any given item. For convenience, assume that they are the last two digits in the third word of the master portion; which word is of no consequence as long as it is constant for all items.



Fixed Master Portion of Item

Now suppose that this item begins in word 13 of a Univac block this puts the word count in word 15. Since we designed the item layout, a knowledge of where the first word is located immediately enables us to find the word which has the counter; the converse is also true. The word count is a pair of digits indicating the number of words in the item; in this case, anything from 08 (no sub-items attached to the master portion) to 98 (the maximum size possible). Suppose it is 23. This signifies that the next item begins 23 words further along than the current item; because the current item starts at word 13, the next starts at 36 ( $25 + 13$ ). Likewise, it's also indicated by a word count, which is known to be in the third word of the item, or in word 38. Thus the word count included as a part of the master portion of the record can be used to modify addresses in an instruction sequence to enable us to "bootstrap" ourselves along the block in memory. Each item contains an indication of how long it is, and consequently tells us where the next item begins. It is not even necessary that an item be contained completely within one block in the case of Univac, which has a 60-word block, the calculation of the location of the next item is always tested against 60; if the new address is greater than 60, the next item is known to be in the next block. This is, of course, a programming detail which really has nothing to do with the item layout, but is necessary because the EDPM cannot read an entire file into memory at one clip.

There are numerous variations of this basic technique. Because the master portion of an item is always fixed in length, it is really not necessary to include its number of words in the total count; only the number of

## 8113.31 (Continued)

words in the variable sub-items is required. The various types of sub-items may all be of different word lengths, in which case a word count in each sub-item present may be used not only for the bootstrap operation, but also to identify the particular type represented. In a file consisting of extremely long items, it may not be possible to include the total word count in the master portion, because memory may not be big enough to hold an entire item at once; obviously, the length cannot be known until the entire item has been handled. In this case, the word count may be calculated only to advance the bootstrapping to the next block, which contains another counter to permit another advancement, and so on.

This method requires continual address modification to advance through successive items in a file; the amount of computing time required to do the modifications and test for the end of a tape record must be considered in determining whether a single counter per item, or several, can be used. Use of this method also requires that the word count be recalculated and corrected every time the total length of the item is adjusted by addition or deletion of sub-items. It is, however, a commonly-used and perfectly acceptable method of enabling an EDPM to keep track of where an item begins and how long it is.

8113.32 Coded Item and Sub-Item Identifications. Another possibility is to carry over into EDPM practice the basic logic of "control punches" used in punch card operations. In this method, the master portion and various sub-items carry identifying characters in specific word or character locations; these serve not only to identify the particular piece being examined, but also enable a program to substitute the length of the particular piece into an address modification routine to permit "bootstrapping" through the file. This is always possible, because the format allocates a specific number of words to each type of sub-item or to the master portion. Thus a coded key might be placed in the first character position of the second word of each piece of the item:

	A									X						
	Y					Z				Z						
Z					A										Y	

Here the "A" identifies the master portion, 8 words long; the "X" the first variable sub-item of 6 words; "Y" the second sub-item of 5 words; and "Z" the third sub-item of 4 words. The method of using these codes to progress through a file is somewhat different from the previous type in technique, although the basic principles is quite similar. Suppose the first "A" has been located. This is known to be an 8-word master portion and hence the next key code lies eight words further along. It is not known what it is, but tests of the code against constant A, X, Y, and Z characters stored as a permanent part of an EDPM program enable its identification, which in turn not only tells us what the item is, but--since the various sub-item lengths were initially established at a fixed number of words for each type--how long it is; i.e., where the

8113.32 (Continued)

next piece of information begins. In the normal EDPM program, the "bootstrapping" requires more computing time, because the advancement is by master portions and sub-items, rather than complete items. The logic of the technique is quite straightforward and this is a standard wrinkle in the repertoire of any good computer programmer. It is extensively used in Univac I installations, for example, although the length of sub-items and master portions is usually either 2, 4 or a multiple of ten words. The reason for this restriction has nothing to do with the generality of the technique, but is based upon the fact that Univac I has only two and ten-word internal transfer orders.

Key codes can, of course, be combined with word or character counts of identify the pieces of information which comprise a business file.

8113.33 Key Words. An approach rather different is the use of key words to signify which parts of an item are present. Unlike the previous two methods, this is generally limited in its applicability to fixed-word length EDPMs such as Univac, and is useful not so much to indicate the number of words present in highly variable files but to indicate which of a number of data elements are present in a given master portion or sub-item. One of the more or less undesirable features of magnetic tape storage is that formats must be laid out in a manner similar to punch card fields, allowing for entries which may not exist in some of the items. For example, in a central stock control file, it may be desired to include for each warehouse the quantity on hand, the amount on order, the total of back orders and a summary of sales for the past three months--four data elements in addition to the warehouse identification. For many items, there will be only one or two non-zero entries at a number of warehouses, yet each sub-item must be filled out with zeros to bring it to the constant size. An alternative approach is to establish a key-word consisting of nothing but "1" and "0" entries, each position corresponding to a given data element in the item or sub-item. A "1" indicates that the corresponding element is present in the record; a "0" indicates that it is absent. Zero words are not included in the item--only the significant entries. The sequence of "1s" and "0s" in the key word indicates which ones are included. For two reasons this sort of technique has not gained much headway at the present time: (1) In character-coded machines such as Univac, only 12 data elements can be keyed in one word, and many words might be required to key an extremely long record; and (2) the amount of programming necessary to analyze the key words and manufacture addresses for desired data elements can get excessive unless special address-modification and word shifting instructions are provided.

In an EDPM which is binary internally, this technique has definite possibilities for some types of files; the key words can be very few in number because each binary bit can signify an entire word and binary machines usually have bit-shifting and conditional transfer instructions so far not included in straight business EDPMs. Some organizations having IBM 704s are contemplating the use of this technique in handling payroll jobs on the 704.

8420 WORD LENGTH IN ELECTRONIC DATA PROCESSING MACHINES

The "words," or basic units of information, comprising business records are highly variable in length or number of characters. They range all the way from a single character, usually information in coded form, to several score -- a part description may require two or three hundred characters. An analysis of a large number of business records indicates that about two-thirds of all "words" are six or fewer characters in length and most of the rest do not exceed twelve characters. As a further note, "words" greater than this latter length tend to be mostly alphabetic or alphanumeric, such as names, addresses and noun descriptions. Strictly numeric information seldom exceeds ten or eleven digits.

In order to be an efficient business data processor, an EDPM must be capable of performing operations on these variable-length units of information. There are several methods of doing this and the choice of alternate approaches has given rise, among other things, to the present controversial statements and assertions on the relative merits of "fixed" versus "variable" word length in an EDPM. The purpose of this section is to analyze some of the various logics which have been developed to permit the handling of variable length business "words."

Most of the electronic computers and data processors which have been built or announced are "fixed" word length machines, within the meaning of the definition given in 8421. The major "variable" word length machines which have been developed are the IBM 702 and 705 and the RCA BIZMAC. The different logics of these two major equipments (the 705 being little more than a slightly modified 702) involve only two of a number of possible techniques which could be developed to handle "variable" words. To limit the scope of this discussion, most of the comments will be based upon the specific methods used in these two machines to handle "variable length" words, leaving out alternative methods which have not yet been incorporated into announced EDPMs.

Sections 8421 and 8422 are concerned with more or less detailed aspects of the subject. The essential sales and procedural implications of fixed vs. variable word length are contained in 8423, and a knowledge of the other two sections is not necessary to make use of the information contained in 8423.

8421 DEFINITION OF TERMS

Further discussion will be more explicit with a precise definition of what is meant by "fixed" and "variable" word length. It should be noted that the terms are generally used as describing characteristics or attributes of an EDPM; instead of referring to "fixed word length" as a noun, it is customarily an adjectival modifier, as in "fixed word length machine." The definitions are predicated upon this latter usage.

8421.1 Basis for Classification. The distinction between a "fixed" and "variable" word length EDPM is based upon the number of bits operated upon by an instruction. This can be expressed in another way: The difference arises from the number of memory accesses for operands required to execute one machine instruction. The term "memory access" means the operation of setting up the internal switching circuitry to make one basic transfer of information (binary bits) to or from memory. In most EDPMs, such an access requires one basic machine cycle of time; in general, the carrying out of one instruction takes an integral number of these machine cycles. The definitions following will modify these statements slightly, but the concept is still valid.

8421.2 Fixed Word Length Machine. A fixed word length EDPM is one which performs its operations on a constant and fixed number of binary digits, called a word. Stated another way, all instructions requiring transfer of information to and from memory always involve a fixed number of bits; the exact number is a part of the built-in logic of each EDPM and cannot be varied by the user. In general, the memory access for one word takes one basic machine cycle of time. The "word" may be a pure binary number, as in most scientific computers, or a specific number of binary-coded characters, as in most business EDPMs. The machine itself **does** not know the difference and simply operates upon a known and constant number of bits.

Fixed word length EDPMs may be further subdivided into three categories, depending upon the method of making memory accesses. The distinction is in many respects technical, but a knowledge of the classes facilitates understanding the reason behind the fixed word length concept.

8421.21 Parallel Word Transfers. Some EDPMs access the entire word in parallel; that is, the switching circuitry has enough "lines" to provide one for each bit of the word and all bits are moved simultaneously. Typical examples are the Univac Scientific (1103A) and the IBM Types 701-704-709. All have a 36-bit binary word and make one access in one basic machine cycle for the entire 36 bits. To be of this type, a machine must have a memory of a form permitting simultaneous access to a complete word; for practical purposes, this is limited at the present time to Magnetic core and the obsolescent electrostatic storage devices.

8421.22 Parallel-Serial Word Transfers. In this type EDPM, several bits are accessed in parallel, but more than one such access is required to obtain or deliver an entire word. All magnetic drum processors are of this type; here the bits of each character are handled in parallel but the successive characters follow one another in sequence. The Univac File Computer and IBM 650 are in this category; the former moves twelve

8421.22 (Continued)

characters in sequence (serially), handling the bits of each character in parallel, while the latter does the same thing for eleven character codes (ten digits plus sign). In scientific drum computers, the internal language may be binary, in which case the bits moved in parallel are some integral sub-multiple of the word length of the complete binary word. In magnetic drum computers, the term "word time" is used to indicate the basic overall time required to "access" one complete word.

Magnetic drum EDPs are of the parallel-serial type as a matter of economy rather than technical necessity. They could easily be modified to handle an entire word in parallel, but the additional cost of the switching circuitry would add appreciably to the cost of the equipment. Likewise, increasing the speed of accesses by a factor of ten or twelve would be useful only if accompanied by faster operating circuits, which again cost much more than circuits geared to the speed of character-at-a-time access.

- 8421.23 Serial Word Transfers. In a few EDPs, the transfer of the bits comprising a word is completely serial--that is, one bit at a time. This type of transfer is typical of EDPs having a so-called "circulating" memory, in which the bits are available in a serial fashion only. The outstanding example is Univac I; its mercury delay-line memory is a serial type storage device, with the bits available one after another as they go through the amplifying circuitry before being reintroduced into the mercury tanks. In general, EDPs of this type have an acoustic delay line of some form as the high-speed memory.
- 8421.3 Variable Word Length Machine. A variable word length EDP is one which performs its operations on a varying number of binary digits, rather than a fixed number. Each character code of an operand requires a separate memory access, each one requiring one basic machine cycle of time. The number of accesses required to handle a complete "word" varies with the number of characters to be handled; because this is usually indeterminate from the instruction itself, special provisions are incorporated into the machine logic to signify the end of a piece of information. Although there are numerous methods of accomplishing handling of variable numbers of characters, only the two equipments which have the feature are discussed.
- 8421.31 IBM 702/705 Version. It might be more correct to refer to these equipments as having "variable field length" rather than "variable word length." IBM's variable word length concept, as exemplified in the 702-705, is identical with the standard meaning of variable fields in punch card methodology. Once a field size for a given word has been established, that word is always handled in the predetermined length.

The 702 is strictly serial-by-character in all memory accesses. The 705 does have a five-character fixed word length transfer for access to instructions and for the high-speed memory transfer; in both instances, the full 35 bits of five characters are obtained and moved in parallel, using the same switching circuitry. With these two exceptions (the first of

8421.31 (Continued)

which is beyond control of the programmer), the 705 is a purely serial-by-character processor.

8421.32 BIZMAC Version. The BIZMAC performs its operations on a character-by-character basis, similar to the 702/705, with one significant difference: Under certain conditions, an operation can be stopped as soon as all significant characters of a word have been operated upon. As an example, a seven-digit numeric field might contain, in a record, an entry "0000012," which is to be added to something. Only the last two digits are significant and BIZMAC can stop an operation as soon as they have been processed. This is variable word length processing taken one step farther than the IBM equipments. The use of this feature requires that non-significant zeros and blanks in variable length fields be changed to "item separators"—special character codes having no other meaning. The BIZMAC makes successive memory accesses, one character at a time, until an item separator is encountered; this stops the operation regardless of the field size. If no separators exist, the specified number of characters are processed. Item separators must be inserted by programmed instructions; they are not generated automatically as a part of machine operations.

8421.4 Instruction Execution Times. In a fixed-word length EDPM, each instruction requires a specific time for execution; this time never varies because the operation is always performed on an identical number of characters. (There are, of course, some instructions, such as shifts, which require varying execution times). The execution time is almost always an integral number of basic machine cycles; different types of instructions require different numbers of cycles and similar instructions in two different computers may differ in the number of cycles required. For instance, in Univac I and II, which have similar logics, the transfer of one word from memory to a working register requires three basic cycles (excluding latency time in Univac I): (1) One cycle to obtain the instructions; (2) one cycle to interpret the instruction and set up the circuitry to perform it; and (3) one cycle to obtain the operand from memory and place it in the register. Some EDPMs do this in two cycles, the first two steps being combined into one.

A variable word length EDPM has execution times dependent upon the number of characters involved in an operand. The total time is the sum of a constant time necessary to obtain the instruction, interpret it and set up the circuitry (usually two or three basic cycles), plus one cycle for each character operated upon.

8422 SIGNIFICANCE OF WORD LENGTH IN AN EDPM

From the user's standpoint, whether an EDPM is of fixed or variable word length in internal operations is of purely academic interest. He is--or should be--interested primarily in the amount of productive work per dollar which the processor can deliver. To the extent that the word length in an EDPM affects total processing time and cost, it is a question of direct interest to him. In this section an attempt is made to outline the principal implications of fixed word length machines and the 702/705 and BIZMAC versions of variable word length.

- 8422.1 The User's Point of View. The EDPM user is interested in the equipment solely as a tool capable of doing productive data processing. His yardsticks are cost and time of doing accurate and useful work. The factors allocated to cost and time include, obviously, the total charges of the equipment and its operators, but also encompass "set-up charges"--in the case of EDPMs, programming and coding.

The performance of productive work by an EDPM involves the time spent in internal data handling and necessary machine housekeeping--such things as additions, subtractions, comparisons, data transfers, instruction modifications, etc. In addition, it embraces the time required to move information into the processor to be worked upon and that necessary to transfer completed work into output storage devices; in today's data processors, this is tantamount to saying "tape read and write time."

The effect of word length insofar as it affects the ability of the EDPM to do productive work is of legitimate interest to the user. Academic discussions of abstract theory belong in research, not in a computer installation.

- 8422.2 Implications of a Fixed Word Length EDPM. In this section are listed a number of statements of fact and certain operational considerations pertaining to data processing on a fixed word length machine; these are basically uncontroversial.

1. The word length should be long enough to contain the large majority of business data elements; it is desirable to minimize the number of double-word (or "double-precision") instruction sequences. Word lengths of 10-12 digits or characters, and particularly the latter, meet this requirement.
2. With a given number of digits assigned to the address portion of an instruction, the EDPM memory holds several times as many characters as the maximum possible address. For example, the 12-character word and 4-digit instruction address of Univac II make possible a total of 10,000 words, or 120,000 characters.
3. The working registers of the EDPM--such as Univac's rA, rX, rF and rL--can all be of one-word capacity; this is possible because the maximum number of characters which must be operated upon by any one instruction is pre-determined and constant.
4. The most efficient use of memory requires that short data elements be combined, several to one word. Otherwise, there is inefficient use

8422.2 (Continued)

of the space available for storage of information to be processed.

5. The most efficient use of input-output facilities also requires that short data elements be combined into a word. Unless this is done, a significant percentage of the information on magnetic tape may be "padding," contributing nothing to production and decreasing the effective rate at which information can be read into and written out from the processor. Ideally, the information placed on tape should consist entirely of useful information with no padding.
6. The combining of several data elements into one word necessitates "extracting" specific ones when they are required in the processing. This means that "extract patterns" must be stored in memory, using some of the available storage capacity, and instructions must be placed in programs to move the extractors to working registers. The latter requirement takes some memory space for instructions and some computer time for their execution.
7. A single data element contained in two different records may not be located in corresponding character positions of the two words containing them. In this case, shifting of one is necessary before such operations as comparison or addition can occur. The shift instructions take memory space and computer time.
8. Because of the existence of the possibility just mentioned, it is desirable to place data elements which must be added or compared in corresponding character positions of their respective words. This requirement introduces another element which must be taken into account in the establishment of file formats.
9. Because arithmetic operations are always performed on a fixed number of digits, the handling of the sign poses no problems. As a matter of fact, it can be assumed to be positive unless a specific "minus" character code is included.

This list is not exhaustive, but includes some of the more important aspects applicable to a fixed word length EDPM.

8422.3 Implications of the 702/705 Variable Word Length Method. In this section are listed a number of statements of fact and certain operational considerations applicable to the specific method chosen by IBM to obtain variable word length on these two equipments. Like the foregoing, it is not exhaustive.

1. With a given number of digits assigned to the address portion of an instruction, the EDPM can contain only as many characters as can be expressed by that number of digits. With the 4-digit instruction address of these equipments, IBM has overcome the 10,000-character limitation only by resorting to use of the zone codes in one digit of the address to achieve an increase to 40,000.
2. Because there is no indication in the instruction of the number of characters to be operated upon, special provisions are necessary to signify when an operation has been completed.

8422.3 (Continued)

- a. Arithmetic operations are terminated by a non-numeric character at the left of the field involved. Because it is not always possible to locate the number in a position which has an invariably non-numeric data character in the required place, it may be necessary to put an otherwise unnecessary character (usually a blank) in the position. The various working registers automatically adjust themselves to the size of the operands.
  - b. The transfer of non-numeric information between memory and the working registers is terminated by a special character code "setting" the register size. This requires either that special instructions adjusting the register to the specified size be included in programs, or that one of the auxiliary storage registers be permanently set aside for a specific data element length. The method chosen by IBM also requires that different instructions be used to move non-numeric data than are used for those involving arithmetic operations.
3. Because the number of characters to be operated upon varies, registers in the 705 are much larger than those in a fixed-word length machine. It contains one of 256 positions, one of 32 and 14 of 16. It is necessary for instructions to specify which register is involved; the means of accomplishing this uses the zone code positions of two digits in the address part of the instruction.
  4. In the 702/705, arithmetic operations in the registers strip the zone bits from the character codes. Because the address portion of the instruction uses the zone code bits in three of the four digit positions, addresses can be modified only by the "add to memory" instruction. Comparison of address limits for exits from loops requires constants equal in both the numeric and zone portions to the particular address being tested.
  5. In EDPMs, arithmetic operations are performed in the same method used by human beings--from the units position to the left (This is done to simplify the handling of "carry" digits). Because the 702/705 secure only one digit at a time from memory, it is then necessary that the sign be obtained with or before the units digit. IBM has chosen to indicate the sign with the zone code bits of the units digit. In effect, the units digit is identical with an alphabetic character: A through I for a plus sign, J through R for a minus. Instructions are necessary to be sure that the units position is "signed," and more are required to remove the sign when final output information is being prepared. These instructions require memory space and computing time.
  6. All memory accesses in the 702 are character-by-character, including those for instructions. As a result, the machine is quite slow, despite a 23 microsecond cycle time. The 705 was modified to provide parallel access to instructions; the same circuitry is used for the so-called "high-speed memory transfer," but for nothing else.

## 8422.3 (Continued)

7. Because the units or addressed position of instructions involving memory accesses is always matched up with the "units" position of the specified working register, no attention need be paid to the relative locations of two corresponding data elements in two different files.
8. Because the word length is variable, within the limits of maximum register size, only one instruction is required to perform an operation on any number of characters. "Double-precision" operations required in fixed word length machines are avoided, saving some instructions and, possibly, some computer time.
9. Although not specifically a part of the variable word concept, the restrictions surrounding the use of the "high-speed memory transfer"-- which is an almost mandatory method of moving data inside the 705-- must be considered in setting up file formats. One effect is to increase the number of non-significant "blanks" terminating operands involved in arithmetic instructions.

8422.4 Implications of BIZMAC Variable Word Length Method. Although the BIZMAC is subject to some of the comments applicable to the 702/705, the basic use of variable word length is quite different. In this equipment, not only can field sizes vary in length, but different elements within one field can be handled to eliminate non-essential zeros and blanks. The principal implication of the method of using "item separators" is the fact that they are not generated automatically, but must be program-inserted. In addition, even though two operands in an arithmetic instruction may have separators to terminate the execution cycles, no separator is placed in the answer; it also must be programmed.

Internally, the use of variable length words in BIZMAC requires a rather considerable amount of programming. There is some question as to whether the technique was adopted for internal reasons; the separators are used to "compress" information stored on magnetic tape. That is, only the significant digits or characters plus one separator of each variable-length word are written on or read from tape. The method adopted does introduce some programming to place succeeding elements read into memory locations which are known.

COMPARISON OF FIXED AND VARIABLE WORD LENGTH EDPMs

From the comments of 8422, it is evident that both fixed and variable word length EDPMs have both advantages and disadvantages. Where the fixed word Univac requires extractors and instructions to use them, the 702/705 "signing" of arithmetic operands or setting of register lengths and the instructions to do these things, and the BIZMAC needs item separators with the instructions necessary to place them. Where Univac results in some "padding" to fill out occasional words or to expand an item into a whole number of words, the 705 introduces "padding" to terminate arithmetic fields or to fill items out to a multiple of five characters (to use the "high-speed memory transfer"). Any comparison of the relative merits of fixed versus variable word length, as exemplified by these equipments, must consider not only the time required to execute any one instruction, but also the time required to get ready to execute it. It must go even beyond this; the effect on input-output time is of definite importance.

An impartial evaluation of the relative merits of fixed word length, as used in Univac II, and variable word length, as in the 705, is literally impossible to make. There are too many ramifications which almost inevitably enter into consideration and are often only partly related to the word itself. Similarly, the application under discussion is important; completely different conclusions can be derived simply because of differences in data element sizes, processing requirements, the number of inputs, and the number and type of outputs. As a typical case of the seemingly remote factors which must be taken into account, consider this one simple facet: Univac's fixed word length would be of vastly different significance if the card-to-tape converter and the high speed printer did not have plugboards. With them, several short data elements in one word present no particular problem; control panel wiring can scramble or unscramble them as desired. With them, internal editing often consists of **nothing** more than a pair of V-W instructions moving several adjacent words, each of which may have two or three data elements; without them, more internal instructions would be required to facilitate data edition. This is precisely what occurs in the variable word length 705, which does not have a printer plugboard. Yet, seemingly, control panels have no direct relationship to how Univac or the 705 operate internally.

What can be attacked--and highly successfully -- is the approach that variable word length is an end unto itself -- a sort of "sacred cow" of data processing methodology. Yet precisely this concept is all too often encountered. What also can be attacked is the implication that only the variable word length machine has the "miraculous" ability to handle data elements of variable sizes. Univac, with its fixed word, is fully as capable as the 705 of handling variable length data elements; the mere fact that Univac does it in a different manner and uses different instructions does not mean that it lacks the ability. One simple fact is often overlooked -- The logic of a fixed word length EDPM is such that it operates upon all the characters of a word, or any part thereof, in exactly the same time. As long as it has the ability to isolate a part of a word from the remainder, it can handle variable length data elements. Virtually all fixed word length EDPMs have this ability through "extract" or "buffering" features.

8423

(Continued)

In a similar vein, neither is fixed word length an end unto itself. It is simply one technique--and a highly efficient one--of internal EDPM operations. The fact that these are done on a fixed number of characters does not mean that data elements must be of that fixed size. After all, a data processing machine intended for business applications would be rather ill-designed if it assumed they were.

At this point, the question may be raised: "Yes, the statements you have made are all well and good. But they are--and I think you will agree--rather general. What I want to know is this: What can I do to show my 'brainwashed' prospect that variable word length is not sacrosanct?" Now this is a perfectly legitimate query. It must be conceded that specifics are easier to sell than generalities. Are there any of these "specifics?" In the vernacular of the day, "You betcha."

Before considering some of these detail points, let's iterate one thing. We are selling a data processing system--and a darned good one. We are not selling either fixed or variable word length, and they are important only insofar as they affect how "darned good" our products are. This point should always be kept uppermost in mind. Now for some brass tacks.

8423.1 What Equipment Is the Competition Proposing? Our principal competition in the EDPM field is coming from International Business Machines Corporation. IBM also happens to be the source of most of the claims for the desirability of variable word length. And IBM also happens to make fixed word length machines and to propose them as efficient data processors. Here are the three major classes of equipments IBM is proposing for data processing, with approximate monthly rental ranges:

Type 650	\$ 5,000 - \$14,000	monthly	Fixed Word Length
Type 705	24,000 - 40,000	"	Variable Word Length
Type 709	50,000 and up	"	Fixed Word Length

Needless to say, IBM claims all of them are the most efficient data processors there are in the respective price ranges.

This simple three-line listing leads to an obvious conclusion: IBM preaches the "advantages" of variable word length only if the prospect is a potential 705 user. Their salesmen are not even going to mention the word "variable" if a potential 650 or 709 sale is impending. To be blunt, IBM talks "variable word length" only to 705 prospects and only because the 705 happens to have it. This is, of course, a common and accepted sales approach--seize upon something you have that the competition doesn't and stress its importance.

The use to which this can be put varies with potential customers; the Univac salesman is the best judge of how to turn this fact of the IBM line to our advantage. A few possibilities are listed.

To a 705 prospect--which means our proposal involves a Univac II--it may be apropos to point out IBM's own inconsistencies and apparent uncertainties. If variable word length is so awfully attractive, why don't the

## 8423.1 (Continued)

650 and 709 have this feature? If variable word length makes the 705 a better data processor, why would the same thing not hold true with the others?

Perhaps IBM may claim that the volume of work to be performed on a 705 makes variable word length more desirable. There are at least two good rebuttals to this. For one thing, small volume applications often require just as many different types of operations as the large ones; there are as many types of calculations in a 1,000-man payroll and labor cost distribution as in a 50,000-man one. The only difference is that the latter has 50 times as many repetitions. If variable word length permits a job to be done faster on the 705, the same thing would hold true for the 650. Secondly, if variable word length is desirable to permit more work to go through the 705, it would be even more desirable on the 709, which supposedly can handle even more volume.

Or IBM may observe that the logics of the three EDPMs are different, and only the 705 can profit from variable word length. The obvious retort: "Why are they different?" Data processing is data processing and many typical business functions involve pretty much the same number of different operations, regardless of volume. Or does the volume of work change the logic of a machine? If so, why is the small machine fixed word, the medium variable, and the large fixed?

8423.2 Internal Processing of Data Elements. The advocates of variable word length EDPMs lay heavy stress on the fact that this feature permits short data elements to be handled more rapidly than long ones. Why take the time to handle twelve characters when the element only has two? A little later, some general remarks will be made about the veracity of this statement. Right now, consider a few specific facts.

8423.21 Processing Frequency of Various Sized Data Elements. About two-thirds of business data elements are six or fewer characters in length. This is generally true, but a more pertinent point is this: What is the frequency of processing data elements of varying lengths? It may be perfectly true that a given business item has 80% of its data elements three or fewer characters long. To conclude from this statement that 80% of the work done involves elements of three or fewer characters is a totally unwarranted assumption. 90% of the work may be done on elements of ten or twelve characters. If this is the case, the "attractiveness" of variable word length largely vanishes. If there is any advantage to variable word length machines, it arises only when the average item processed is fairly short.

A number of frequency studies have been made on the number of characters in operands from typical Univac I programs. Unfortunately, these are all, so far as known, static counts; a program is taken and each instruction is entered as a spread-sheet tally according to the number of characters in the data element it is working on. Much more pertinent, and necessary, is a dynamic count; that is, a breakdown of the number of characters operated upon counted once for each time an instruction is executed in a

8423.21 (Continued)

machine run. The fact that an exception routine has 22 instructions operating upon elements of one or two characters is not particularly significant if that routine is used only twice in a 50,000-item master file process. As a further note, some of the static counts include "house-keeping" instructions--such as those to modify memory addresses--and thus do not afford a reliable picture of even the static frequency of instructions operating upon data. And, finally, counts of the lengths of data elements made from Univac--or 705 or File Computer or 650--programs are subject to one other restriction which may partially invalidate their significance: To some extent, the "field size" of a data element is subject to numerous considerations other than central computer processing, and may be shorter or longer than necessary to facilitate periphery operations.

Although the factors outlined above restrict our detailed knowledge of the effects of variable data elements, some specific comments can, nonetheless, be made.

- 8423.22 Tape Limited Processing. So long as an EDPM is in a tape-limited process, the actual computing time required to do whatever is necessary is practically immaterial. If a tape input/output rate of 1,000 characters every 50 milliseconds is the limit that can be achieved with the tape units in use, it is immaterial whether the processing on that 1,000 characters takes 45 milliseconds, or 5. There is no gain in processing speed by increasing the amount of time the main frame is "spinning its wheels." How much time does the average EDPM spend in this status? Well, on Univac I, about 50%. On Univac II, it is estimated that 65% of the time in an average installation will be tape-limited.

Now this may be countered by IBM by noting that the 705 is "so fast" internally that it can use two or three Type 777 Tape Record Coordinators and handle several sets of tapes simultaneously. This **is** perfectly true. It can. And it is still slower than Univac II with its one faster set of input-output buffers and tapes. It also costs more. The **proof** of this is developed in detail in Section 1180-1189.

- 8423.23 Computer Limited Processing. Although there is no gain by speeding up computational speed when the machine run is tape limited, it would be an advantage in the computer-limited runs. The claim is that the 705 can do this with its variable word length instruction execution. In some cases, this quite possibly is the case. However, let's consider some of the typical cases which are computer limited.

Some sort-merge operations are in this class, particularly with fairly short items. Aside from the tape-handling sequences, this type of run involves three basic phases: (1) Comparison of index numbers; (2) modification of addresses; and (3) internal transfers of items. The last has nothing to do with the length of words. How about the first two? Index numbers tend to be longer than "average" data elements; they are not often less than six characters, and such common indices as names, part numbers, customer account numbers, social security numbers, job order

## 8423.23 (Continued)

or project number, etc., run anywhere from eight or nine up to 24 or more characters. And in this range the 705 takes as long as Univac II to handle an operand.

Consider address modification, which is common in sort-merge routines. Both Univac II and the 705, for example, have four-digit addresses. The fastest Univac modification is a "B" and "AH" pair of instructions, which take a total of 360 microseconds. The 705, using a "Load" and "Add to Memory" pair, takes 204, which is obviously faster. Furthermore, it may be possible to keep the modifying increment in an auxiliary storage register and reduce the time to 102 microseconds. The 705, then, is definitely faster than Univac II by better than three to one. Of course, this conclusion overlooks the fact that I can modify two addresses in Univac in the same amount of time--and, surprisingly enough, both addresses of a V-W pair to move an item to its proper place may be involved, depending upon the exact sort-merge technique followed. Furthermore, it neglects to consider that limit tests on the new address are necessary--and Univac may do this with generalized overflow. The 705 won't--at a minimum, two more instructions ("Compare" and a conditional transfer), totalling 136 microseconds, are required. And it may take more if constants can't be contained in one of the auxiliary storage registers. (one of the baffling aspects of competitive proposals is how IBM assumes that 15 auxiliary storage registers can hold 60 or 80 constants at one time).

Another common type of computer-limited process is a payroll and labor cost distribution; in terms of processing requirements, such functions as insurance premium and cash value determination, public utility customer billing and accounting, and inventory analysis are similar. What are the sizes of data elements involved? To be sure, some comparisons on one-or two-character codes, but the bulk of computer time (aside from transfers into and out of working storage and address modification) is spent on arithmetic operations: Multiplications of two factors ranging from three to six or seven digits to arrive at rounded products of from five to nine or ten digits; testing of answers of six to eight digits against limits of the same size; accumulating a wide variety of totals ranging from six to ten or eleven digits in length. Much of the computing time is spent on medium to fairly long length data elements, not on extremely short ones. Of course, the fact that Univac can deliver a rounded product as desired, whereas the 705 cannot, possibly is unrelated to word length; the fact that it is fairly easy to build in this facility with a fixed word length machine and a good deal more difficult in one with variable word length does have a definite effect on comparative abilities to arrive at required answers.

Likewise foreign to fixed versus variable word length is the fact that Univac gives correct answers because of its self-checking circuitry. It is understood that some auditors have acquired the habit of insisting that accounting procedures provide for accuracy; although this requirement likewise is unrelated to an EDPMs word length, it has an adverse effect on the 705's ability to deliver reliable results.

Original 4/15/57

8423.3 Relationship of Word Length to Item Length. A business item (record, in IBM terminology) consists of a number of data elements of varying lengths and items within one file typically are of variable length. In the establishment of file formats it is necessary to consider not only the method by which the processor handles variable length data elements, but also the means available to transfer information internally to reassemble a new master file, taking into account expansions and contractions, or to reassemble various pieces of several files into a new output--say, one for printing.

In the 705, IBM has utilized one version of a variable data element or word length machine--and at the same time made it somewhat awkward to use with variable length items or records. The 702 was strictly a variable word length machine--every instruction, every operand and every transfer was obtained and moved character by character. To be more precise, the 702 might be called a machine with a fixed word length of one character, with instructions operating on successive "words" until something stops the cycle. With the 705, IBM recognized that this character by character access--in other terms, their "variable word length"--was not always a desirable attribute. For example, every EDPM instruction must be obtained from memory and, because it is always of fixed length, why not obtain it in one memory access instead of several? So they did--put in a five-character access line. By doing this, IBM admitted that variable word length was no panacea. But they went even further--they put in hardware to use this same five-character access line to speed up internal memory transfers.

Well, the choice made gears memory transfer to a record or item, and imposes restrictions on the item format. The basic difficulty is that the use of the five-word memory transfer, when variable length items are being handled, involves a lot of address modification and associated limit tests, all of which take instructions and time. Moreover, it can't always be used, particularly in assembling bits and pieces of several items into a new one, necessitating slow character-by-character movement. (This is discussed in detail in the sections on the Type 777 Tape Record Coordinator, 1180, and the 705 central processor, 5140). They have gone even farther, input from and output to tape is also geared to records and is most efficient when all tape blocks are the same size. In effect, they have built an equipment which operates internally on variable length words, in part, but is used most efficiently with fixed length items.

Now consider Univac. Its internal operations on words, its method of making memory transfers and its systems of storing information on tapes are all completely independent of data element or item size. It handles fixed or variable words and fixed or variable items with complete impartiality and flexibility.

Now the fact that an EDPM operates internally on a variable word length basis may seem only remotely connected to memory transfers or storage of information on tape. In theory, this may be true, but we are selling hardware against hardware, not theory. In attempting to overcome certain weaknesses in a completely variable word length machine, IBM has limited the effectiveness of the 705 internal memory transfer and seriously restricted flexibility, and increased the processing time, of magnetic tape files.

8423.4 Does Variable Word Length Provide Faster Internal Processing? The proponents of variable word length claim that processing is faster because instructions operate only upon the specific number of characters in a data element. It is obvious, they claim, that a comparison on a one-character element is faster if the equipment looks only at that character than if it handles a full word of, say, 12 characters. How valid is this claim?

Before considering the facts, a precautionary word should be noted. It is unfortunate that, throughout the EDPM field, comparisons between two or more equipments are made on the superficial and totally inadequate basis of a single instruction execution time. As always, the true measure of a data processor's efficiency is its cost of doing a certain amount of productive work. One EDPM, for example, may be able to execute individual instructions twice as fast as another. But if it takes three times as many instruction executions to perform a given process, it rather obviously is not the faster machine. The types of instructions provided are at least equally important with the speed of their individual executions.

Although the generality of the conclusions is subject to restrictions for these reasons, it is possible to analyze, at least in part, the validity of this claim. For this purpose, let us consider existing equipments with known characteristics, rather than attributes which could be included in an electronic data processor, but haven't been.

The equipments to be used are Univac II and the IBM 704 and 705, all of which are single-addressed EDPMs. Of these, only the 705 has the so-called "variable word length." Let us look at the times required to execute the addition instruction; in all three equipments, this instruction obtains an operand from memory, adds it to the contents of a register and places the sum in the same register. The following tabular presentation, with times in microseconds, is applicable:

<u>Factor</u>	<u>Univac II</u>	<u>704</u>	<u>705</u>
Word Length (characters)	12	36 bits	Variable
Basic Cycle Time	40	12	17
Cycles to Perform an Addition	5	2	2 $\frac{1}{N}$

Some interesting conclusions can be derived. The logics of the equipments differ in terms of the basic cycles required to complete the addition. The 704, which has the fastest cycle, requires only two of them to add two 35-bit plus sign numbers; this is equivalent to ten or eleven decimal digits. The 705, which has a basic cycle about 45% longer, can add two one-digit numbers in 51 microseconds--more than twice the time required by the 704. It takes 204 microseconds to add two 10-digit numbers--700% longer than the 704, although its cycle rate is only 45% longer.

Now compare Univac II and the 705. Univac's cycle rate is 135% longer. And quaintly enough, Univac's addition time for a three-digit number is almost exactly 135% more than that of the 705. And this leads directly to a pertinent observation: In adding anything over three-digit numbers, the 705's faster execution time is due not to its variable word length

## 8423.4 (Continued)

but to its faster cycle time. Or take Univac's "Bm" instruction, which loads one word into rA in 120 microseconds or three basic cycles, with the 705's "Load" order, which moves one character into an accumulator in 51 microseconds and succeeding characters in 17 microseconds more each. Univac's time is almost exactly 135% more than the 705's time for one character; in other words, for many comparable instructions, any apparent speed advantage of the 705 over Univac is due solely to its faster cycle rate, regardless of the number of characters moved. And it might be added that the 705's actual time is greater if more than five characters are involved.

A number of studies have been made of the comparative speeds of Univac II and the 705 in executing actual operating instructions. These indicate that, on the average and instruction for instruction, **the 705 can go through about  $2\frac{1}{4}$  -  $2\frac{1}{2}$  instructions to one for Univac.** It may only be coincidence that Univac's cycle time is about  $2\frac{1}{3}$  times that of the 705. Whether it is or not, one conclusion is inescapable: The fact that the 705 has a faster instantaneous instruction execution rate than Univac II is traceable almost exclusively to its faster cycle time; the variable word length feature is only a contributing factor of much lesser importance.

The statement in the previous paragraph of the relative instruction execution rates of Univac II and the 705 should not be misconstrued; it is based on individual instructions. Interpreting it to mean that the 705 can do  $2\frac{1}{4}$  -  $2\frac{1}{2}$  times as much work as Univac II is completely unjustified. And fallacious.

8423.5 Programming Ease. In theory, there should be no difference in the programming ease or difficulty of a fixed as compared with a variable word length EDPM. In actuality, the particular gimmicks incorporated in both the 702/705 and BIZMAC make them more difficult to program than Univac. With the 705 particularly, the incidence of clerical errors in coding appears to be much higher than that of our own equipments. This is not Remington Rand Univac opinion. It has been stated by several independent, and impartial, management consultants who have had occasion to analyze the operations of the two equipments. The section on the 705 central processor discusses this aspect in more detail.

8423.6 What Is the Future of Variable Word Length Machines? The remarks of 8423.4, coupled with the fact that IBM--as the major exponent of variable word length--is proposing EDPMs of both types for business data handling, naturally give rise to the question: "What type of word length will the EDPM of the future have?" This is a perfectly valid request for a customer to make; it is indeed somewhat confusing for one manufacturer to extol the virtues of both types, and to do it simultaneously.

Any attempt to forecast the future is risky. It is particularly hazardous in a field which has been progressing as rapidly--both technically and operationally--as that of electronic data processing equipments. The development and perfection of new techniques, availability of improved components and circuitry, perfection of faster hardware and the lowering of

## 8423.6 (Continued)

production costs of components and circuits are of such frequent occurrence as to be prosaic. From the product planning aspect, the knowledge and experience gained from existing computers, coupled with technical developments, resulting in system concepts unthought of or impracticable only a short time ago. Although it cannot be expected that the rate of growth and improvement will continue indefinitely at the pace of the last few years, there is no present indication of an immediate slowdown. And so, because sudden breakthroughs on the research front--both technical and logical--may open up completely new avenues for exploration, a look to the future--even the relatively near future--becomes pretty much a "crystal ball" process.

But for what it's worth, one person's thoughts for the very shorth term--say three or four years--follows. The 705 and BIZMAC are the last of the equipments which will have variable word length of the type they represent; modified versions may come out, of course. This statement refers to the large-scale, million-dollar class of machine and is based upon this line of reasoning. Magnetic core storage appears to be the principal type which will be used in this class equipment for the next few years. Although constant speed and cost improvements are being realized, it still represents a fairly expensive cost-per-bit storage medium, and the cost increases rapidly as the "speed" of the cores is increased; here "speed" means the rapidity with which successive accesses to storage can be made. Existing core storage units use what is called "destructive read-out;" that is, the information in the cores is destroyed when it is read and must be restored or "regenerated" to keep it available. Consequently, the memory access time, usually equivalent to a basic machine cycle, is in two parts: (1) The reading phase, at the conclusion of which the bits are available in the working registers of the equipment; and (2) the regeneration phase. Cycle times of 12 microseconds are available today, and this may be cut in half, at no additional cost, in the next two or three years. Today, storage of this speed represents a selling price of about one dollar per bit, not exactly inexpensive.

Considerably faster storage devices can be built today, but the cost-per-bit is prohibitively expensive if any volume is involved. They are, however, not so expensive that a few of them can't be used as the working registers. This is what is done in Univac II, the Univac Scientific and the IBM 704 and 709. These processors consist of large core storage memories of (relatively) slow speed and a few much faster registers--faster by a factor of possibly ten to twenty. Consequently, it becomes feasible to develop a computer with instructions executed along these lines: (1) Obtain the instruction during the first half of the basic cycle and move it into the high-speed instruction register; (2) in the next half-cycle, "regenerate" the instruction in memory and simultaneously decode it in the instruction register and set up the necessary circuitry for execution; (3) in the next half-cycle, obtain the operand from memory; and (4) in the next half-cycle, regenerate the operand in memory and simultaneously process it in the high-speed registers. This is the essential cycle in the Univac Scientific and the IBM 704 and 709. Now it is evident that so long as two memory accesses are required for the

8423.6 (Continued)

instruction and the operand, there is actually a loss of processing speed if the equipment secures operanda character by character when it is fast enough to operate on several characters during the necessary regeneration half-cycle. As long as the present methods of using magnetic core storage exist, fixed word length operation will be much faster than variable, even allowing for instruction time which might be necessary to handle extractors for data elements shorter than the word length.

Of course, progress is being made on cheaper methods of assembling magnetic core matrices, on cores which can be operated at higher rates of speed without a cost increase, and on methods of non-destructive readout, which will eliminate the need for regeneration. Although the availability of these improved mass storage units may change present concepts, it appears that, at least for the immediately foreseeable future, a large-scale data processor can economically include a few working registers capable of operating at speeds several times as great as the memory. If this condition prevails, fixed word length operation probably will provide the faster processing.

The foregoing comments may assist in placing the variable word length 702/705 in an historical perspective. Their basic logic was settled several years ago, at a time when IBM was using electrostatic storage as the high-speed memory. Just why the character-by-character access of the 702 was adopted, when a similar type memory used a full word access in the 701, is not known; it may well have provided an economic technique at the time. Memory switching requirements are somewhat less and registers can be slower than a fixed word length machine. Thus it may well have been that consideration of available techniques and costs of achieving them were highly important criteria in settling upon the single character mode. Certainly the fact that IBM is building both variable and fixed word length processors indicates that there is within the corporation no unanimity of opinion that variable word length is best for the user.

In the large-scale EDPM field, then, one opinion is that fixed word length will be a characteristic of equipments to be announced in the very few years to come. In the medium and low-price fields, there is possibility of even more diversity of opinion. The medium range may utilize fairly slow, but not too expensive, magnetic core storage units of small size, backed up by drums or random access files. In this case, the internal word (but not necessarily the storage word) may be fixed in length; this is by no means certain. In the lower-priced field and in the medium-cost equipments using magnetic drums as the main storage medium, fixed word length probably can be expected. Although magnetic drum computers could be built with a variable length word, the existence of latency time and the desire for low selling cost makes any gain from variable word length memory access quite marginal. Consequently, one school of thought feels that variable word length of the type used in the 702/705 is not a particularly attractive possibility in this range of equipments.

There are obviously other opinions. With progress continuing at impressive speed, there is nothing more uncertain than attempting to predict what an EDPM of two or three years from now may incorporate.

8423.7 Conclusion. The foregoing are just some of the possibilities for the not too distant future. They may or may not be indicative of what will happen. A technical development announced tomorrow could easily make completely new concepts possible.

But, if nothing else, this discussion may have helped to place the variable word length feature of the 702 and 705 in its proper place as only a part of a data processing equipment, and to consider it as just another step in the evolution of electronic data processing techniques and logic. There would be little objection if IBM claims for the variable word length feature of these two equipments were based largely on its being a technically feasible and, possibly, economical method of computer operation. Unfortunately, such an approach has little sales appeal. But there is definite objection to IBM claims that the variable word length concept of the 702 and 705 provide the best systems approach for handling variable length business data elements.

IBM itself has provided part of the rebuttal by building--and of course claiming superiority for--fixed word length data processors. Several other rebuttal points are included within this section. They may be summed up quite briefly: The function of an EDPM is to process business data. Provided an equipment operating with a fixed length internal word can handle variable length data elements, there is no reason it cannot be an effective processor.