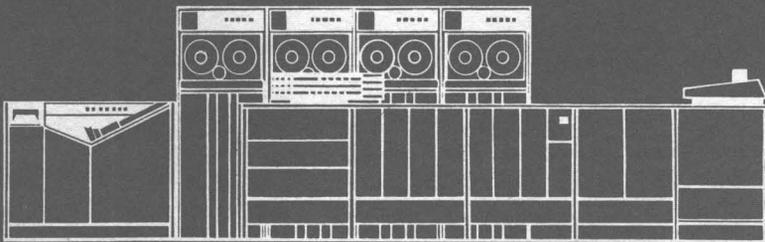


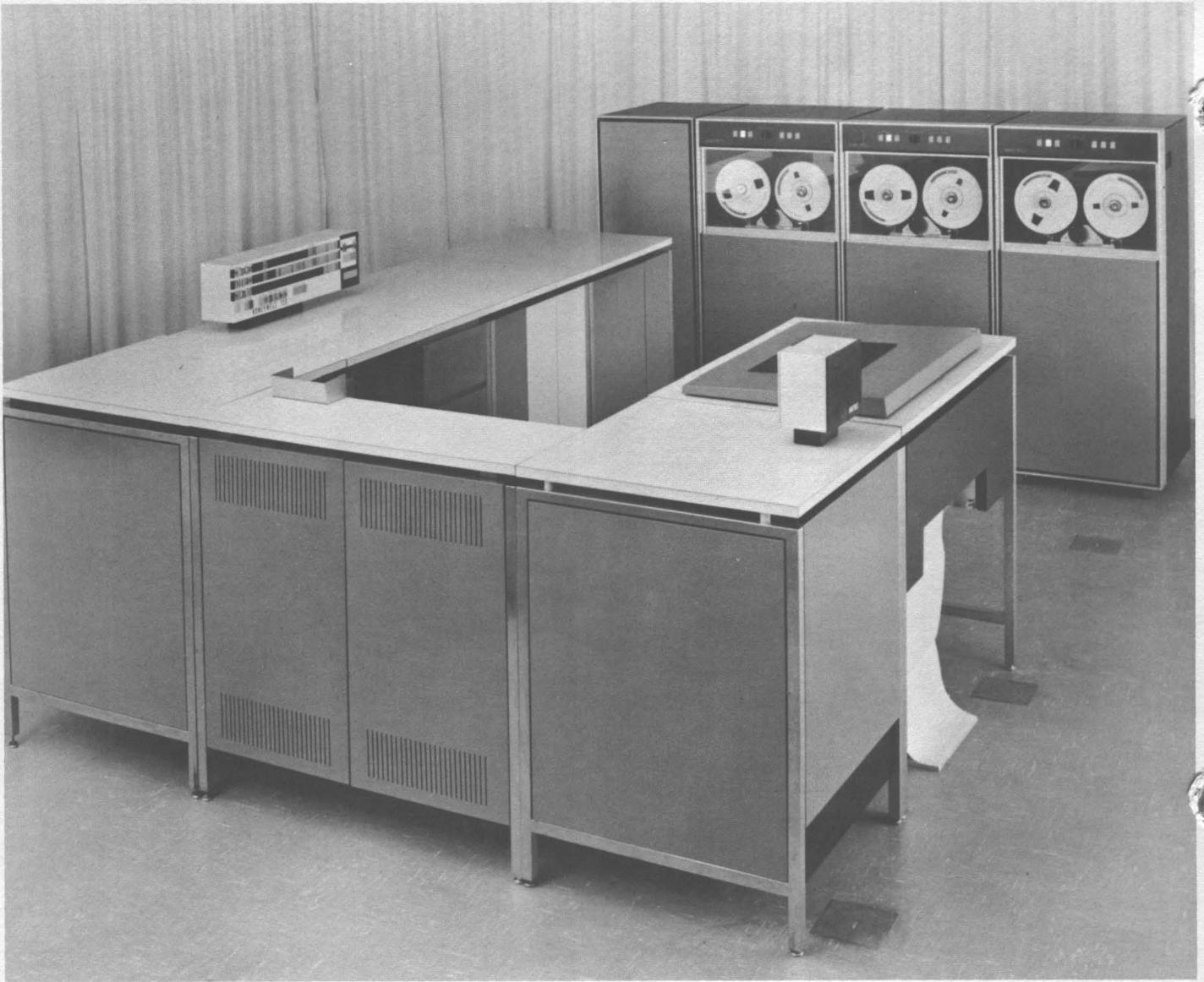
# HONEYWELL SERIES 200

## SUMMARY DESCRIPTION



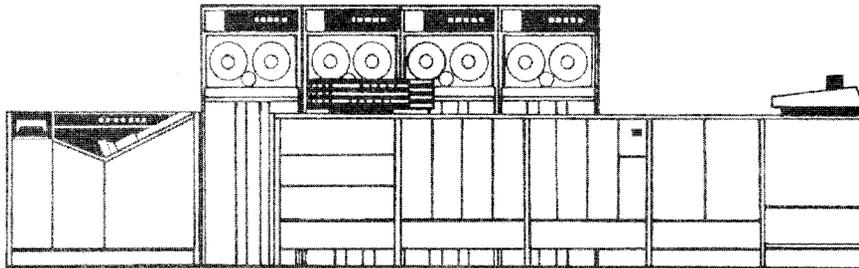
# Honeywell

ELECTRONIC DATA PROCESSING



# **SERIES 200**

## **SUMMARY DESCRIPTION**



# **Honeywell**

**ELECTRONIC DATA PROCESSING**

Questions and comments regarding this manual should be addressed to:

Honeywell Electronic Data Processing  
Information Services  
60 Walnut Street  
Wellesley Hills, Massachusetts 02181



SECOND EDITION  
First Printing, February, 1966

Copyright 1966  
Honeywell Inc.  
Electronic Data Processing Division  
Wellesley Hills, Massachusetts 02181

# Table of Contents

FOREWORD .....	4	5. INSTRUCTIONS .....	25
1. INTRODUCTION .....	5	Instruction Descriptions .....	25
2. PROCESSORS .....	7	Fixed-Point Arithmetic Instructions .....	26
Computing Power .....	7	Floating-Point Arithmetic Instructions .....	27
Memories – Speed and Capacity .....	7	Logic Instructions .....	29
Instructions .....	8	General Control Instructions .....	31
Addressing .....	10	Interrupt Control Instructions .....	32
Simultaneity .....	10	Data Move Instructions .....	33
Read/Write Channels and Input/Output		Edit Instruction .....	34
Trunks .....	10	Input/Output Instructions .....	34
Interrupt Processing Facility .....	11	6. PROGRAMMING SYSTEMS .....	36
Conversion Compatibility .....	11	Series 200/Basic Programming System .....	36
Structural Modularity – Reliability .....	11	Series 200/Operating System .....	38
Summary .....	12	7. APPLICATION SYSTEMS .....	43
3. INPUT/OUTPUT DEVICES .....	13	Industry Application Systems .....	43
Peripheral Controls .....	13	Printing and Publishing .....	44
Control Panel and Consoles .....	13	Manufacturing .....	44
Magnetic Tape Units .....	14	Transportation .....	44
Card Equipment .....	15	Distribution .....	45
Mass Memory File Transports and Control ..	17	Banking .....	46
Random Access Drum Storage and Control ..	18	Finance .....	47
Printers .....	19	Life Insurance .....	47
Paper Tape Equipment .....	20	Fire and Casualty Insurance .....	48
Banking Equipment .....	20	Education .....	48
4. DATA COMMUNICATION FACILITIES ....	21	General-Purpose Application Systems .....	49
Processor Communication Facilities .....	21	Linear Programming .....	49
Data Communication Controls .....	23	Network Modeling .....	49
Fast-Access Storage .....	23	8. TABLES .....	50
Data Station .....	24	Instruction Formats and Timing .....	50
Communication Software .....	24	Correspondence Among Series 200 Central	
		Processor, Card, and Printer Codes .....	54

# List of Illustrations

Memory Capacities Available .....	8	Data Station Peripheral Devices .....	24
Series 200 Instruction Repertoire .....	9	Data Station .....	24
Series 200 Index Registers .....	10	Decimal Add Operation .....	28
Series 200 Processors Input/Output Characteristics .....	10	Branch Conditions for FBI Instruction .....	28
Summary of Processor Characteristics .....	12	Branch Conditions for FBA Instruction .....	28
Type 220-1 Console .....	13	Extract Operation .....	29
Type 220-3 Console .....	14	Partial List of Variant Characters for BCT Instruc- tion .....	30
Type 204B Tape Units .....	15	Partial List of Variant Characters for BCC Instruc- tion .....	30
Magnetic Tape Unit Specifications .....	15	Formation of Logical Product in BBE Instruction	31
Summary of Card Equipment Specifications .....	16	Information Stored by SVI Instruction .....	33
Type 223 Card Reader .....	16	Formation of Translation Table Address of Equiv- alent .....	33
Type 214-2 Card Reader/Punch .....	17	Time Relationships in Execution of PDT Instruc- tion .....	35
Type 251 Mass Memory File Transport .....	17	Peripheral Control and Branch Operation .....	35
Mass Memory File Specifications .....	18	Functions and Data Flow in the Basic Program- ming System .....	37
Type 270 Random Access Drum .....	18	COBOL Source-Language Program .....	40
Type 222 Printer .....	19	Integrated Management Information and Control System for the Distribution Industry .....	45
Summary of Printer Specifications .....	19		
Type 209 Paper Tape Reader .....	20		
Applicability of Honeywell Communication Equip- ment .....	22		

## Foreword

This summary description of the Series 200 is intended for those having a general familiarity with data processing. Machine characteristics and programming aids are described in terms which should aid comparisons between the Series 200 systems and competitive equipment. The equipment characteristics reported herein remain subject to minor revision in order that design improvements may be incorporated.

# 1 Introduction

A computer, like any other tool used by man to tackle a task or problem, is limited in the extent to which it can be applied efficiently. A lightweight truck will probably break down if loaded considerably beyond its design limit. Likewise, a steam shovel is not the economic solution to digging postholes. A modular tool, however, can be applied to a wider range of jobs more efficiently. If the tool has several dimensions, each of which can in turn be modularized, the facility with which it can be tailored to handle specific jobs is enhanced even further.

This is how Honeywell has tackled the problem of matching computers to specific data processing requirements. By breaking computer capability into basic dimensions and providing a range of capability in each dimension, Honeywell is uniquely able to match a computer to a given job. Also, the computer can be expanded or modified very easily to match changes in system requirements. This approach to computer system design is the basis of Series 200.

Series 200 represents an "off the shelf" processing capability consisting of processing, input/output, and software modules that can be brought together in virtually any combination to form systems accurately tailored to solve any business or business-related data processing problem economically. Series 200 includes six compatible processors which display outstanding cost/performance characteristics and offer the user great flexibility in his choice of speed, simultaneity, and memory capacity. A broad array of input/output devices, offered in several performance levels, provides many input/output media alternatives. Software, consisting of programming and application systems, is tailored to match the modularity of hardware.

The six processors of Series 200 are the Models 120, 200, 1200, 2200, 4200, and 8200. The 8200, the most powerful member of the series, is completely compatible not only with other Series 200 models but also with Honeywell's 800 and 1800 systems. Because of its dual processing ability it is described in a separate publication entitled *Honeywell Series 200/Model 8200 Summary Description*, File No. 143.0011.0000.0-191.

Further description of Series 200 within this pub-

lication refers only to the Models 120, 200, 1200, 2200, and 4200.

## Processing Dimension

- Memory speeds ranging from 3 microseconds to 188 nanoseconds per character
- Memory capacities ranging from 2,048 to 524,288 characters, in modular increments
- Up to 30 index registers; flexible nanosecond control memories
- A universal set of powerful instructions affording program compatibility between processors
- Instruction and data compatibility with 1401, 1410, 1460, and 7010 systems
- Advanced programming and memory addressing methods, plus editing, and multiply/divide operations
- Powerful floating-point capability

## Input/Output Dimension

- Up to 16 peripheral operations performed simultaneously with computing
- Up to 48 peripheral control units connected to a processor; each accommodates one or several peripheral devices and is equipped with an automatic program interrupt facility
- A wide variety of peripheral equipment available in a range of performance capabilities, including communication devices, card equipment, magnetic tape and paper tape units, mass storage units, high-speed printers, banking equipment, and memory-to-memory adapter units
- Broad-scale real-time capability that includes an efficient interrupt facility, single- and multi-channel communication controls (the latter accepting data from up to 63 lines simultaneously), multi-level code handling, and a wide range of remote terminal facilities

## Software Dimension

- Basic Programming System, which provides flexible software modules employing self-loading, unit-record techniques for use in small, card-oriented installations
- Operating System – Mod 1, which provides semi-centralized, automatic control for medium-scale tape- or mass-storage installations
- Operating System – Mod 2, which provides completely centralized, highly automated computer management, with a minimum hardware overhead requirement, for medium- and large-scale installations of all types
- Application Systems, which assist directly in the performance of functions which are part of a user organization's operations



# 2 Processors

Series 200 processors are designed primarily for business applications and for jobs involving combined business processing, data communication, and scientific computing. In most data processing, the governing performance dimension is throughput – the quantity of data taken in, processed, and transferred to output media as computed results. High throughput requires not only an ability to transfer large quantities of data into and out of a processor; it also requires the capacity to process the data internally. This capacity includes the performance of all required computations and manipulations, while at the same time servicing demands from input/output devices quickly enough so that these devices can operate at their rated speeds. Therefore, the internal speeds of the processor must be high enough to allow the required combination of computing and input/output servicing. Clearly, then, high-throughput processors must possess a good *balance* of internal speed and potential input/output demand. As the following discussion will demonstrate, Series 200 processors incorporate an optimally balanced mixture of computing power and peripheral simultaneity at all levels of over-all throughput capability.

## Computing Power

The ability of a processor to perform purely internal processing, involving only such operations as arithmetic, logical functions, data transfers, and editing, is largely a function of: (1) the amount of memory available for storing programs, as well as control and working data; (2) control and main memory speeds, which govern the time required to obtain and move instructions and data within the processor; (3) the selection of instructions in the processor's repertoire and the efficiency of the logic by which instructions are implemented; and (4) the memory addressing scheme used. Series 200 processors provide computing power to meet the needs of any business jobs or applications involving business, scientific, and communication processing.

**MEMORIES – SPEED AND CAPACITY:** High internal speeds are assured by main memory cycle times ranging from 3 microseconds to 188 nanoseconds per 6-bit character and control memory cycle times from 500 down to 125 nanoseconds. For example, consider the following statistics, which are based on typical situations:<sup>1</sup>

Operation	Execution Time, Microseconds		
	Model 120	Model 1200	Model 4200
Decimal Add	69	35	10
Compare	57	29	9
Branch if Character Equal	36	18	7
Move Characters to			
Word Mark	54	27	8
Floating Multiply	n/a	56	18

<sup>1</sup> Three-character addresses are used to refer to five-character operand fields. Instruction access times are included in the times shown. The times for floating multiply refer to operations using a 36-bit mantissa and a 12-bit exponent.

The speeds of Series 200 memories are complemented by the wide range of storage capacities available at each speed level. Memory size in the 120 processor ranges from 2,048 to 32,768 six-bit characters. At the other end of the scale, Model 4200 processors are available with 65,536- to 524,288-character memories. The modularity of Honeywell systems is exemplified by the relatively small increments in which main memory can be expanded, even at high capacity levels. Main memories in Models 1200, 2200, and 4200 can be equipped with a "storage protect" capability which shields the contents of one or more designated memory areas against accidental alteration by unrelated programs. High-speed control memories of from 16 to 55 control registers are used in all processors.

Information is stored in main memory locations either in pure binary form, as 6-bit alphanumeric characters, or as signed decimal quantities. Any number of consecutive locations can be grouped to form fields; groups of consecutive fields can be delineated as items.

PROCESSOR MODEL	MEMORY CYCLE TIME	MEMORY CAPACITIES AVAILABLE (thousands of characters)																						
		2	4	8	12	16	20	24	28	32	40	49	57	65	81	98	114	131	163	196	229	262	323	524
120	3 microsec- onds/char.	✓	✓	✓	✓	✓	✓	✓	✓	✓														
200 <sup>1</sup>	2 microsec- onds/char.		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓											
1200	1.5 microsec- onds/char.				✓					✓		✓	✓	✓	✓	✓								
2200	1 microsec- onds/char.				✓					✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
4200	188 nanosec- onds/char.													✓			✓					✓	✓	✓

<sup>1</sup> Three processors are available with the Model 200: the Types 201, 201-1, and 201-2. All references to the Model 200 processor in this publication are to the Type 201-2.

Such groupings are defined by programmed or manual setting of punctuation bits associated with each memory location. (Fields and items are defined, respectively, by word marks and item marks.) Punctuation bits can also be set to form a record, which is defined as any unit of information that is to be transferred between main memory and a peripheral device as the result of a single peripheral data transfer instruction.

There are no reserved input/output areas in main memory. The programmer has complete freedom in specifying the locations and sizes of such areas to meet the needs of any program. This allows both a high degree of programming flexibility and economical usage of memory.

A parity bit in each character position is used to maintain the accuracy of all data. Parity checking, performed automatically, is a method of checking a character each time it is moved in memory to insure that it retains its original value.

**INSTRUCTIONS:** Series 200 processors have available a repertoire of instructions which, with tremendous flexibility and power derived from the use of variant characters, can handle all arithmetic, logical, control, editing, and input/output functions necessary for business data processing. Also included in all processors are instructions for dealing with peripheral and communication interrupts and for manipulating data in codes of up to 12 levels. Hardware multiply and divide operations are standard in all processors except the 120. Models 1200, 2200, and 4200 can be equipped with a floating-point arithmetic facility for use in scientific applications.

Instructions are variable in length. The basic instruction format consists of an operation code which specifies the type of operation to be performed, two operand fields which specify the binary addresses of fields to be used in the operation, and a variant character:

Operation Code	A Address	B Address	Variant Character
----------------	-----------	-----------	-------------------

The variants can be used to expand the meaning of the operation code or to specify literally a piece of data to be used in the operation. However, there are many times when not all of these instruction elements are needed, in which case they may be omitted to minimize both the amount of memory storage required and the time necessary to retrieve and execute an instruction. Peripheral control and input/output instructions have a slightly different basic format from the one described above. The most common Series 200 instruction formats are illustrated below.

Operation Code	A Address	B Address	Variant Character
----------------	-----------	-----------	-------------------

Operation Code	A Address	B Address
----------------	-----------	-----------

Operation Code	A Address	Variant Character
----------------	-----------	-------------------

Operation Code	A Address
----------------	-----------

Operation Code	Variant Character
----------------	-------------------

Operation Code
----------------

The availability of Series 200 instructions to individual processors is shown in the accompanying table.

## Series 200 Instruction Repertoire

Name of Operation	Processor Type <sup>1</sup>					Name of Operation	Processor Type <sup>1</sup>				
	120	200	1200	2200	4200		120	200	1200	2200	4200
<b>Fixed-Point Arithmetic</b>						<b>General Control Functions</b>					
Decimal Add	S	S	S	S	S	Set Word Mark	S	S	S	S	S
Decimal Subtract	S	S	S	S	S	Set Item Mark	S	S	S	S	S
Decimal Multiply	—	S	S	S	S	Clear Word Mark	S	S	S	S	S
Decimal Divide	—	S	S	S	S	Clear Item Mark	S	S	S	S	S
Binary Add	S	S	S	S	S	Halt	S	S	S	S	S
Binary Subtract	S	S	S	S	S	No Operation	S	S	S	S	S
Zero and Add	*	*	S	S	S	Change Addressing Mode	S	S	S	S	S
Zero and Subtract	*	*	S	S	S	Change Sequencing Mode	*	*	S	S	S
<b>Floating-Point Arithmetic</b>						Store Control Registers	S	S	S	S	S
Floating Add	—	—	*	*	*	Load Control Registers	S	S	S	S	S
Floating Subtract	—	—	*	*	*	Load Index/Barricade Register	—	—	*	*	*
Floating Multiply	—	—	*	*	*	Store Index/Barricade Register	—	—	*	*	*
Floating Divide	—	—	*	*	*	<b>Interrupt Control Instructions</b>					
Store Floating Accumulator	—	—	*	*	*	Store Variant and Indicators	S	S	S	S	S
Load Floating Accumulator	—	—	*	*	*	Restore Variant and Indicators	S	S	S	S	S
Floating Test and Branch on Accumulator Condition	—	—	*	*	*	Resume Normal Mode	S	S	S	S	S
Floating Test and Branch on Indicator	—	—	*	*	*	Monitor Call	S	S	S	S	S
Decimal to Binary Conversion	—	—	*	*	*	<b>Data Move Instructions</b>					
Binary to Decimal Conversion	—	—	*	*	*	Move Characters to Word Mark	S	S	S	S	S
Store Low-Order Result	—	—	*	*	*	Load Characters to A-Field Word Mark	S	S	S	S	S
Load Low-Order Result	—	—	*	*	*	Extended Move	*	*	S	S	S
Binary Mantissa Shift	—	—	*	*	*	Move and Translate	*	*	S	S	S
Binary Integer Multiply	—	—	*	*	*	Move Item and Translate	*	S	S	S	S
<b>Logical Functions</b>						<b>Editing</b>					
Extract	S	S	S	S	S	Move Characters and Edit	*	*	S	S	S
Half Add	S	S	S	S	S	<b>Input/Output</b>					
Substitute	S	S	S	S	S	Peripheral Data Transfer	S	S	S	S	S
Compare	S	S	S	S	S	Peripheral Control and Branch	S	S	S	S	S
Branch (Unconditional)	S	S	S	S	S						
Branch on Condition Test	S	S	S	S	S						
Branch on Character Condition, Basic	S	S	S	S	S						
Branch on Character Condition, Extended	*	*	S	S	S						
Branch if Character Equal	*	*	S	S	S						
Branch on Bit Equal	*	*	S	S	S						

<sup>1</sup> S = standard, \* = optional, — = not available.

**ADDRESSING:** All Series 200 main memory locations are directly addressable. Three additional features facilitate advanced programming and addressing of large memories — indexed and indirect addressing and variable-length address interpretation.

A Series 200 processor may have from 6 to 30 main memory index registers (see table below). These registers provide an automatic means for address modification without altering the instruction in which the address is modified. Indirect addressing enables the user to reference stored information via one or more intermediary addresses. Variable-length address interpretation refers to the ability of Series 200 processors to operate in three different address interpretation modes, allowing the programmer to code instructions using either two-character, three-character, or four-character addresses. This facility provides the flexibility necessary to allow the direct addressing of large memories, while at the same time saving processing time and memory space when working in localized areas of memory.

SERIES 200 INDEX REGISTERS

PROCESSOR MODEL	MINIMUM	MAXIMUM
120	0	6
200	0	15
1200	15	30
2200	15	30
4200	15	30

## Simultaneity

The speed of internal processing is one of the most important standards in evaluating the total throughput of a system; peripheral simultaneity is the other. Series 200 processors possess several features which enable them to provide powerful but easy-to-use simultaneity: program-assignable read/write channels, multiple input/output trunks, and an interrupt processing facility.

**READ/WRITE CHANNELS AND INPUT/OUTPUT TRUNKS:** The use of program-assignable read/write channels enables Series 200 processors to compute while concurrently servicing from 3 (Model 120 maximum) up to 16 (Model 4200 maximum) input/output operations. In addition, Series 200 processors provide facilities for interfacing with a large number of pe-

ripheral controls, ranging from a possible 9 in the case of the 120, up to 48 in the 4200. The high internal speeds of these processors insure that even when the high degree of possible simultaneity is fully exploited, the increased demands on the processor to service peripheral devices will still be satisfied. Perhaps even more significant than the effect is the cause: this capacity is built into every Series 200 processor. It does not depend upon complex software or expanded system configurations.

Series 200 peripheral controls are connected to a processor via input/output trunks. A control that performs only input or output functions (e.g., printer control, card punch control) requires one trunk. A control capable of both reading and writing (e.g., magnetic tape control, Mass Memory File control) connects to a pair of trunks. With the exception of the Model 120, Series 200 processors can be connected with as many peripheral controls as the available input/output trunks in the processor will allow. This number varies according to the processor and ranges up to a possible 48 in the 4200.

SERIES 200 PROCESSORS  
INPUT/OUTPUT CHARACTERISTICS

PROCESSOR MODEL	NUMBER OF I/O TRUNKS	NUMBER OF I/O OPERATIONS SIMULTANEOUS WITH COMPUTING	
		Standard	Maximum
120	Basic: integrated controls for printer, card reader, card punch. Optional: (1) control for magnetic tape; (2) up to 13 I/O trunks.	2*	3
200	8 - 16	3	4
1200	16	4	4
2200	16-32	4	8
4200	32-48	8	16

\* Card reading cannot be performed simultaneously with card punching in the basic 120.

The basic Model 120 is equipped with integrated peripheral controls for a 450-line-per-minute printer, a 400-card-per-minute card reader, and a punch that processes from 100 to 400 cards per minute. Also available is the Type 103 Magnetic Tape Unit Control which can be connected directly to the Model 120 peripheral interface and which accepts four 13,300-character-per-second magnetic tape units. Either of two optional features (Feature 1015 or 1016) provides up to 13 input/output trunks by which standard Series 200 peripheral controls can be connected. The exact number of standard controls which can be added to

the Model 120 varies according to the power requirements of the controls and is generally a maximum of six controls.

Data are transferred between main memory and a trunk (and thus a peripheral device) via a read/write channel assigned by the instruction which initiates the transfer. A read/write channel is a data path across an interface between main memory and a peripheral device. Whenever an input/output operation is to be performed, a program-assigned read/write channel completes the path between the required peripheral device and the main memory.

The degree of peripheral simultaneity achieved by any Series 200 processor depends upon the number of read/write channels with which it is equipped. Standard equipment of the Model 120 processor includes three read/write channels, of which any two can be active at one time. The simultaneous use of the third channel is available with Feature 1016. Sixteen read/write channels are available in the 4200, allowing a like number of input and output operations, in any combination, to go on at the same time as internal processing. In order to appreciate the full power of the read/write channel concept, consider the following statistics. In one minute, a Series 200 system equipped with a Model 200 processor having four read/write channels can:

- read 800 cards;
- punch 10 columns of data into 400 cards;
- print 1300 lines of 120 characters each;
- read or write 4360 tape records of 500 characters;

(or perform any combination of four I/O operations) and in the same minute, execute 1.25 million instructions.

**INTERRUPT PROCESSING FACILITY:** The Series 200 automatic program interrupt facility provides simple but efficient supervision of processing involving combinations of input/output operations and computing. This facility allows automatic branching as necessary between a main program and servicing routines for all input/output devices. It obviates the need for programmed tests to detect the completion of input/output operations. The automatic hardware interrupt has important applications in the field of data communications and other real-time areas, but it is equally applicable to the supervision of operations as universal as reading and punching cards and paper tape, as well as reading and writing magnetic tape.

The Series 200 interrupt processing facility consists of a hardware program interrupt, which signals a particular condition in an input/output control unit, and a set of instructions used in processing interrupts. A program interrupt occurs whenever a peripheral device has completed an input/output operation. For example, an interrupt occurs at the end of data transfer in a tape read or write operation. Likewise, the receipt of a character from a remote station by a communication control may be signalled by a program interrupt. Interrupts from particular peripheral controls can be inhibited by a program as necessary.

A program interrupt is accompanied by: (1) automatic storage of main program indicator values, control register contents, and an indication of interrupt source; and (2) automatic branching to a routine whose address was previously loaded by program into a special control memory register. This routine can then proceed to determine the number and source of existing interrupts and to process the corresponding input/output demands according to whatever priority was specified by the programmer. The interrupt instruction subset is particularly helpful in this regard. After all demands have been processed, only a single instruction is necessary to resume the main program at its point of interruption and to restore all main program indicators and control registers to their previous values.

## Conversion Compatibility

Series 200 processors are designed according to the Liberator concept, which allows the users of various competitive systems to take advantage of the superior performance of Honeywell systems without incurring the prohibitive costs of reprogramming. For example, the instruction repertoire of Series 200 processors is similar enough to those of several other processing systems, viz., the IBM 1400 series, to allow automated, one-time translation of programs written for these competitive systems to a form suitable for execution on higher-performance Series 200 systems.

## Structural Modularity — Reliability

A major feature of the structural design of Series 200 processors is the use of integrated system modules. Each module contains all the circuitry required for a particular system function; for example, one module

contains all the printer control circuitry, another contains the components of the arithmetic unit, etc.

This modularity greatly simplifies expansion of a system; in most cases, expansion involves little more than plugging in additional modules. The reliability of components within each module has been maximized through the use of silicon semiconductors. In addition, Series 200 takes advantage of the latest advances in the application of monolithic integrated circuits.

## Summary

Series 200 processors possess optimum combinations of high memory speeds, modular memory capacities, powerful instructions, efficient addressing methods, and flexible input/output traffic facilities which afford

the computing power and simultaneity necessary for high throughput rates. The productivity of these processors is enhanced by their programming and operating simplicity. Basic hardware compatibility enables users of competitive systems to convert easily to take advantage of superior Series 200 performance. Sound hardware design, always a Honeywell plus, provides modularity and assures reliability. Processors are equipped with:

- Direct, indexed, and indirect addressing
- 2- to 4-character address interpretation
- Program-assignable read/write channels
- Automatic program interrupt
- 5-, 6-, 7- and 8-level code processing facility

Other processor facilities are tabulated below:

SUMMARY OF PROCESSOR CHARACTERISTICS

Processor Model	Main Memory Speed (cycle time)	Memory Capacity (thousands of characters)	Number of Input/Output Trunks	Number of I/O Operations Simultaneous with Computing	Advanced Programming Instructions	Financial Edit Instruction	Multiply and Divide Instructions	Scientific Processing Instructions	Memory Protect Facility
120	3 microseconds per character	2 - 32	Basic print, card read, card punch controls. Tape control and II I/O trunks optional.	2 - 3	*	*	—	—	—
200	2 microseconds per character	4 - 65	8 - 16	3 - 4	*	*	Standard	—	—
1200	1.5 microseconds per character	16 - 131	16	4	Standard	Standard	Standard	*	*
2200	1 microsecond per character	16 - 262	16-32	4 - 8	Standard	Standard	Standard	*	*
4200	188 nanoseconds per character	65 - 524	32-48	8-16	Standard	Standard	Standard	*	*

\* Feature optional.

— Feature not available on this model.

# 3 Input/Output Devices

Series 200 includes a wide variety of input/output devices so as to enable the use of numerous input/output media. The following devices are offered: card readers, a card punch, a card reader/punch, printers, a paper tape reader, a paper tape punch, magnetic tape units, Mass Memory File transports, random-access drum units, communication controls (discussed in a succeeding section), and an operator's console. Most devices are offered in several performance levels, allowing the user to choose a desired input/output medium at an economical processing level. Particularly significant is the fact that all devices described here, except those to which limitations are specifically indicated, can be connected to any Series 200 processor, contingent only upon the availability of the requisite input/output trunk(s). The great flexibility thus provided allows accurate tailoring of system capabilities to satisfy user requirements.

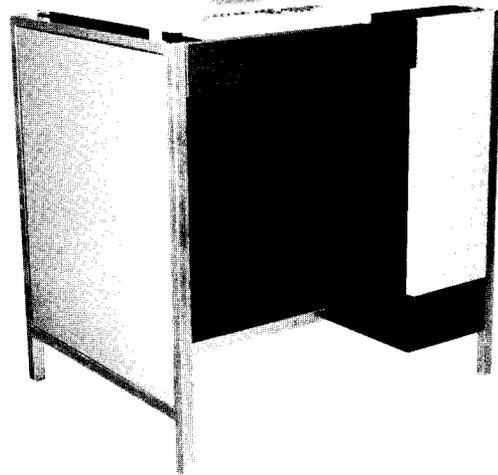
## Peripheral Controls

Peripheral controls are used to regulate the transfer of data between a processor and input/output devices. A significant feature is the fact that these controls operate independently of the central processor and require memory access only when information transfers are performed. In particular, all data validity checks, such as parity checks in magnetic tape transfers, are performed by the controls and do not involve the central processor in any way. Most peripheral controls can also generate an interrupt signal at the completion of data transfer. The initiation of the signal is allowed or inhibited by program control.

Controls for 24-inch-per-second, 1/2-inch magnetic tape units and for 3/4-inch magnetic tape units can each accommodate up to four drives. Controls for other 1/2-inch drives, as well as those for Mass Memory Files and the random-access drum, can each accommodate up to eight devices. Each of the other Series 200 input/output devices requires its own individual control, i. e., multiple devices cannot be connected.

The 120 processor includes integrated controls for: (1) either a card reader/punch or, separately, a 400 CPM card reader and a card punch; and (2) a 450-line-per-minute printer. A control for 24-inch-per-second tape units is also available for this processor. Otherwise, controls for input/output devices are each connected to a processor by means of one or more input/output trunks.

The Type 205 Magnetic Tape Switching Unit enables alternate connection of one or more tape units to different controls.



Type 220-1 Console

## Control Panel and Consoles

A prospective Honeywell customer can choose one of four devices for over-all control of a Series 200 system: a control panel or one of three operator's console models. All devices provide a visual indication of system status and permit manual intervention into system operation.

The control panel, which is actually an integral part of the central processor, contains various control switches by which the operator can start and stop the machine and can load and interrogate both main and control memory locations. Sense switches may be used in conjunction with programmed instructions to stop processing or to select predetermined program paths, thereby increasing the flexibility of a program.

The Type 220-1 Console contains a console typewriter which may be used as a peripheral device, operating under program control, or as a logging typewriter. The central processor control panel is used in conjunction with the Type 220-1.

In the Type 220-2 Console (not available with Model 120) most of the control panel functions, including direct access to the central processor, are performed by means of the console typewriter. In addition, the typewriter can perform the peripheral and logging operations described above. The standard control panel is replaced in the Type 220-2 with a smaller version containing only the main power switches, sense switches, and certain check condition indicators. The control panel of the Type 220-3 Console contains additional indicators used with storage protection and the additional sense switches used with the larger Series 200 processors.

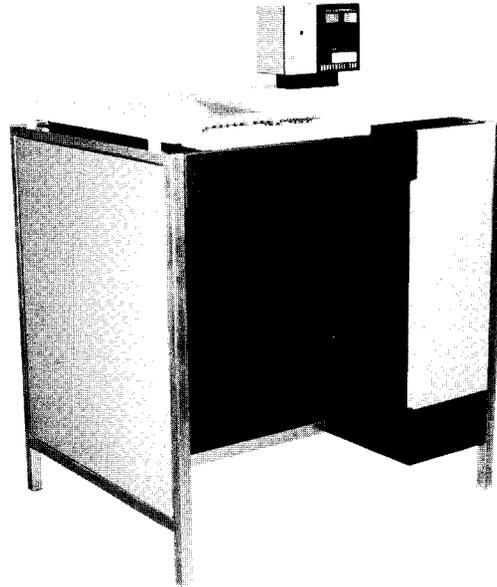
## Magnetic Tape Units

Two complete families of magnetic tape units are provided for use in Series 200 systems:

- Units which process  $\frac{1}{2}$ -inch tape provide: (1) The standard means for storing 6-bit data; and (2) IBM compatibility, including end-of-file mark recognition and the ability to translate between card images in IBM even-parity tape code and Series 200 processor code.
- Units which process  $\frac{3}{4}$ -inch tape provide data compatibility with Honeywell 400/1400/800/1800 systems and, in addition, feature Honeywell's unique Orthotronic control technique for data checking and regeneration.

Programmed tape operations include the following:

- $\frac{1}{2}$ -inch tape units – read forward, write forward, backspace one record, space forward one record, rewind, rewind and release (not available with 24-inch-per-second tape units), and erase; also available is read backward.



Type 220-3 Console

- $\frac{3}{4}$ -inch tape units – read forward, write forward, backspace one record, rewind, release, and regenerate tape channel.

As indicated in the accompanying table, data transfer speeds range from 7,200 to 96,000 characters per second for units processing  $\frac{1}{2}$ -inch tape and from 32,000 to 88,000 characters per second for  $\frac{3}{4}$ -inch units. Also included in the table are "cross-gap" times, the presence of which points to a distinct advantage of Honeywell tape units. When a tape read or write operation is completed, the tape unit begins a deceleration interval which is coincident with the creation of part of the interrecord gap on tape. However, it is not necessary for the unit to stop before beginning to execute a new read or write operation. If such an operation is begun at any time during the deceleration interval, the unit merely accelerates, completes the interrecord gap, and begins the next operation.

The power of Series 200 peripheral simultaneity is evidenced by tape processing statistics: The proportion of available central processor time during a data transfer interval shared with a tape read or write operation ranges from 75% to more than 99%, depending upon the data transfer rate of the tape unit and the speed of the processor being used. Simultaneity is further increased in the case of  $\frac{1}{2}$ -inch tape units: Reading and writing can proceed simultaneously under the direction of a single tape control unit at the same time that computing is in progress (the

tape control for 24-inch-per-second drives does not have this facility).

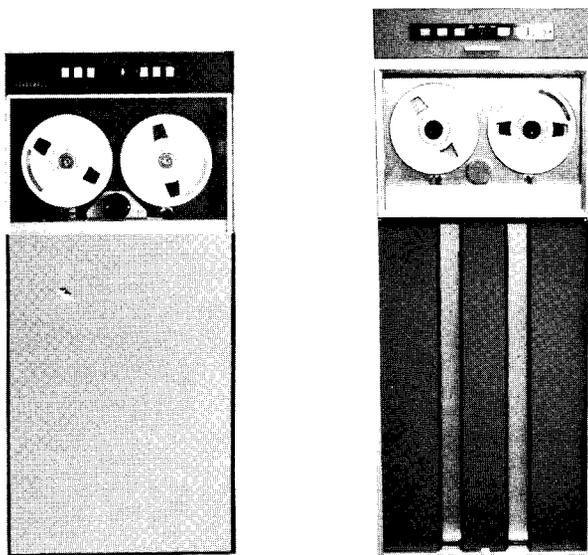
The design of all Honeywell tape units incorporates the vacuum techniques which have earned an outstanding reputation for error-free operation. Vacuum control is used in mounting, driving, and stopping the tape so as to avoid any danger of damage; the reading surface of the tape has physical contact with the read/write head only. A write-enable ring and a manual tape unit switch guard information on tape from accidental destruction by an unintentional write operation.

All information written on 1/2-inch tape is immediately read and checked. During a write operation, a

parity bit is generated for each frame and another is generated for each data channel. The parity bits accompany the data on tape. Frame and channel parity are checked while reading. Failure of any of these checks automatically causes an indicator to be set which can be tested by a programmed instruction. The 3/4-inch tape equipment has the further ability to regenerate any tape channel on the basis of the parity established by the other channels and the frame parity bits.

## Card Equipment

In keeping with the concept of modular processing capability, Honeywell offers a flexible array of punched card equipment. The units described are recent Honeywell developments and include advanced card-handling techniques, such as column-by-column (end-feed) processing. End-feed card processing frees the central processor for other operations during a very high proportion of card equipment cycle time and enables the complete elimination of card cycle clutch points. In card devices employing clutch points, a card input/output instruction can only be acted upon at certain points in the cycle, a situation which limits the device's throughput. The use of end feeding in all Series 200 card equipment enabled Honeywell engineers to incorporate demand feeding, *i.e.*, the execution of card input/output instructions *immediately* upon their receipt by the pertinent control unit. Demand feeding, in combination with the ability of Honeywell's card punch to accelerate over unused card fields, provides maximum rates for continuous card punching.



204  
B-1, B-2, B-3, B-4, B-7, B-11, B-12

204  
B-5, B-9, B-8

MAGNETIC TAPE UNIT SPECIFICATIONS

	Half-Inch Tape Units, Read/Write Speed								3/4-Inch Tape Units, Read/Write Speed			
	24 in./sec.	36 in./sec.		80 in./sec.		120 in./sec.		60 in./sec.	120 in./sec.			
RECORDING DENSITY char./in.	556	200	200 556 or 800 800	200	200 556 or 800 800	200	200 556 or 800 800	200	200 556 or 800 800	533	533	740
TRANSFER RATE thousand char./sec.	13.3	7.2	7.2 20.0 or 28.8 28.8	16.0	16.0 44.5 or 64.0 64.0	24.0	24.0 66.7 or 66.7 96.0 96.0	32.0	32.0 64.0 or 64.0 96.0 96.0	32.0	64.0	88.0
REWIND SPEED inches/sec.	144	108	108	240	240	360	360	180	360	180	360	360
INTERRECORD GAP	.45" .75"	.45" .75"	.45" .75"	.60" .75"	.60" .75"	.70" .75"	.70" .75"	.67"	.67"	.67"	.67"	.67"
CROSS-GAP TIME	18.7 ms 31.2 ms	12.5 ms 20.8 ms	12.5 ms 20.8 ms	7.5 ms 9.4 ms	7.5 ms 9.4 ms	5.8 ms 6.3 ms	5.8 ms 6.3 ms	11.0 ms	11.0 ms	11.0 ms	11.0 ms	5.5 ms
TYPE NUMBER	204B-11, 12	204B-1, 2	204B-7	204B-3, 4	204B-8	204B-5	204B-9	204A-1	204A-2	204A-3	204A-3	204A-3

**SUMMARY OF CARD EQUIPMENT SPECIFICATIONS**

	READERS		PUNCH	READER/PUNCH
<b>SPEED</b> cards/min	400 CPM	800 CPM	100-400 CPM	100-400 CPM
<b>SIMULTANEITY</b>	The central processor, regardless of type, is free to perform other data transfers or computing during at least 99.9% of a card device's transfer interval.			
<b>PROGRAMMED OPERATIONS</b>	Read data and transfer to specified memory area.	1. Read data and transfer to specified memory area. 2. On error card, offset-stack card or go busy.	Punch data from specified area of memory.	1. Punch data from specified area of memory. 2. Read data and transfer to specified memory area. 3. Read and punch same card.
<b>DATA TRANSFER<sup>1</sup> MODE</b>	Automatic translation between Hollerith card code and 6-bit central processor code is standard. Additional transcription mode reading and punching capability also available.			
<b>DATA PROTECTION</b>	Illegal punch check	Validity and cycle checks	Punch check	Illegal punch check on reading; punch check on punching
<b>INPUT HOPPER/OUTPUT STACKER CAPACITY</b>	3000/2500	3000/2500	1200/1300	1200/1300
<b>TYPE</b>	123	223	214-1	214-2

<sup>1</sup> Transcription mode reading and punching not available in Model 120 processor's integrated card equipment controls.

Automatic translation between standard 12-bit Hollerith card code and Honeywell central processor code is a standard facility on all Series 200 card devices. Transcription mode reading and punching are also available on all devices, except when connected to the integrated controls of Model 120 processors.

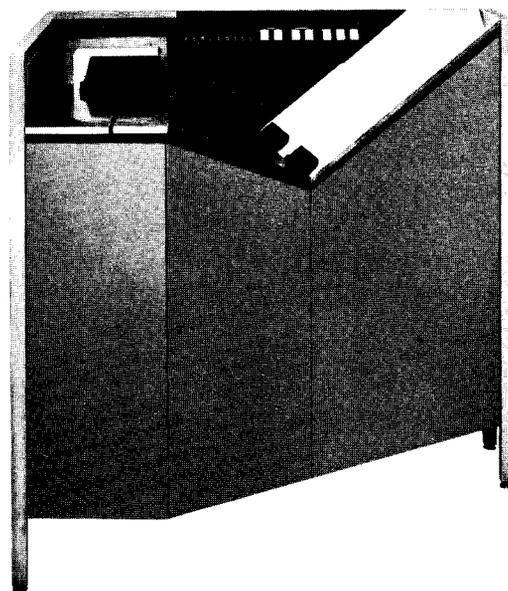
**CARD READERS – 400 AND 800 CPM**

Two high-performance devices are offered for use in Series 200 systems to optically read 80- or 51-column punched cards: a 400-card-per-minute reader, Type 123 (available only with the Model 120); and Type 223, an 800-card-per-minute reader. Processed cards are sent to an output stacker, and those which fail data-protection checks can be offset-stacked under program control. End feeding substantially reduces the time normally required by edge-feed (row-by-row) readers for data transfer to and from main memory; therefore, other peripheral data transfers and computing can be performed during more than 99.9% of a card read cycle. Solid-state electronic components are incorporated in both card reader models to ensure optimum reliability. The speed, simplicity, and reliability of these devices combine to give them the best cost/performance ratios in the industry.

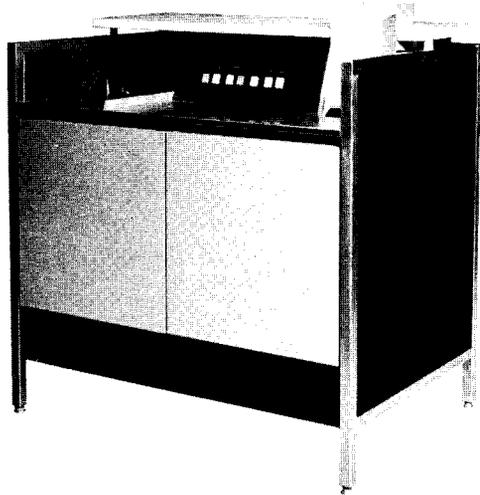
**400 CPM CARD PUNCH**

The Type 214-1 Card Punch operates at speeds of up to 400 cards per minute, depending upon which column is punched last. High-speed column skipping (Feature 066) is available as an option. This feature provides an automatic increase in card advance speed

when unused card columns are detected. This device also incorporates another new feature, dual-character punching, which employs a dual-die mechanism to punch two characters (columns) simultaneously, and adds significantly to the high speeds and reliability of the unit. The Type 214-1 punch was designed for maximum reliability with minimum periodic maintenance. There are no cams, gears, or sliding parts in the feed mechanisms, making lubricating points completely unnecessary. The 214-1 enjoys the same simultaneity advantages afforded other Honeywell end-feed card



Type 223 Card Reader



Type 214-2 Card Reader/Punch

devices: Other processor and peripheral activities can occur during 99.9% of a card punch cycle. Punching errors are detected by a punch check; recognition of an error causes a program-accessible indicator to be set.

#### 400 CPM CARD READER/PUNCH

This dual-purpose device actually has three operational modes; it reads, or punches, or reads a card and punches additional information into the card on the same pass. Punching speed ranges up to 400 cards per minute, depending upon which column is punched last. Operating speed is 400 cards per minute when reading only; if reading and punching during the same pass, the unit operates at its punching speed. This device combines all of the advanced features of the punch and readers described above. That is, the punch station employs dual-character punching, high-speed skipping (optional), as well as high reliability due to the absence of wear-producing cams, gears, and sliding parts. The reading station features optical techniques. Other peripheral data transfers and internal computation can be performed by the central processor during 99.9% of a card processing cycle. The reading and punching stations detect errors by means of illegal punch checks and punch checks, respectively. When a discrepancy is sensed, a program-accessible indicator is set, and the card can be offset-stacked.

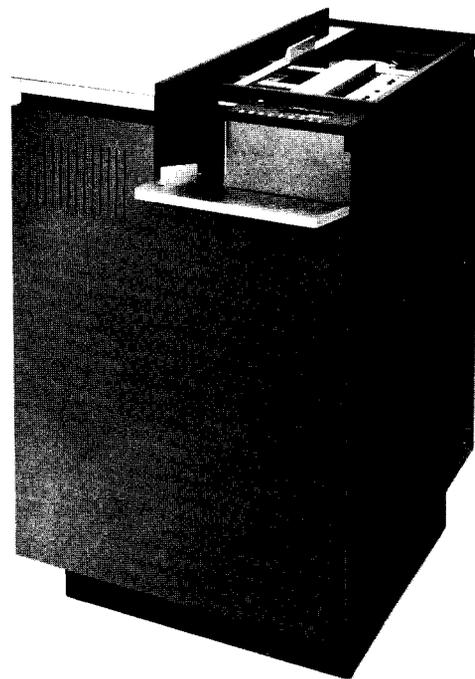
## Mass Memory File Transports and Control

Honeywell's Mass Memory File is a random access, mass storage facility consisting of three transports and a general-purpose control unit. The three transport types constitute a broad range of capacities, and include the most up-to-date design features.

Mass Memory File transports write data onto and read data from Mylar\* tape strips. Each tape strip contains a series of recording tracks in which data is organized, or formatted, according to parameters established by the user. A pack of 512 uniquely notched strips is used during transport operation. When stored off-line, a pack is locked in a lightweight, dust-proof cartridge.

A single program command controls the selection of a strip containing a specified data location. Once a strip has been selected, an accelerator bar moves it down a raceway and onto a read/write drum. Data transfer to or from a specified location is directed by

\* Registered trademark of E. I. du Pont de Nemours Company (Inc.).



Type 251 Mass Memory File Transport

MASS MEMORY FILE SPECIFICATIONS

Type	Storage Capacities					Average Access Times
	Data Characters Per Track	Tracks Per Tape Strip	Packs Per Transport	Millions of Data Characters Per Transport (Nominal)	Millions of Data Characters Per Type 250 Control (Nominal)	
251	968	32	1	15	126	95 ms
252	968	128	1	63	507	150 ms
253	968	128	1-5	63-317	2,537	225 ms

a search and data transfer command. When data transfer has been completed, the strip is released from the drum, passed on to a decelerating capstan and returned to the pack. Strips travel from station to station under their own momentum on a smooth, air-cushioned raceway.

On-line data storage capacities for the three transports are (Type 251) 15 million, (Type 252) 63 million, and (Type 253) 317 million characters. The derivation of these figures is shown in the accompanying table. Up to eight transports in any combination can be connected to a single control unit (Type 250), and thus a single control unit's on-line storage may amount to over two billion characters. On-line storage can be changed simply by exchanging tape packs on the transport with those kept off-line.

The Type 251 transport accesses any on-line data record in an average time of 95 milliseconds, Type 252 in 150 milliseconds, and Type 253 in 225 milliseconds. These access times, plus a high data transfer rate, provide extremely rapid transaction handling. Type 251, for example, can access, read, and write a 900-character record in 115.7 milliseconds, or 132.4 milliseconds if a write check is included.

The Mass Memory File incorporates the same transport techniques which have earned an outstanding reputation for Honeywell's magnetic tape units. Air pressure is used to control and lubricate all tape strip manipulations. Pinch rollers, often the cause of data destruction and surface wear, are not used; the data side of tape strips comes in direct contact only with the read/write heads. Moreover, the notches on Honeywell tape strips never touch the read/write drum or decelerating capstan; thus, notch wear and damage — and the resulting costs and errors — are virtually eliminated.

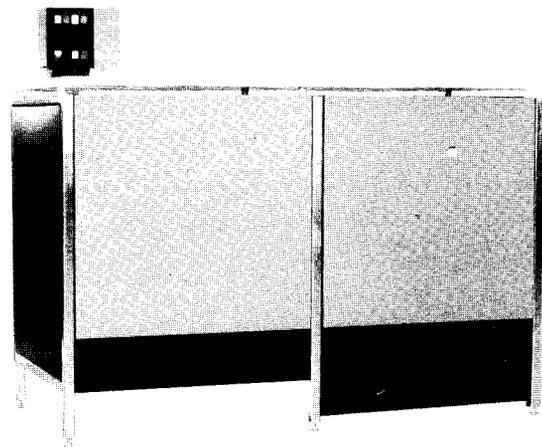
Tape strip data is also protected from unintentional alteration by program through the use of a trio of programmer-designated file safeguards.

## Random Access Drum Storage and Control

The Honeywell Type 270 Random Access Drum Storage and Control provides a highly efficient, random access data storage medium for Honeywell computers. The drum subsystem achieves an optimum combination of high-speed access to large quantities of stored data and low storage cost per unit of information.

One to eight drum files can be connected to a control unit to operate on-line in a Series 200 system. The storage capacity of each drum is 20,480 records of 128 six-bit characters each, or 2,621,440 characters. Thus, a single control/drum subsystem can have a total capacity of over 20 million characters.

Program control of drum operations is maintained by use of two instructions: search-and-write and search-and-read. Both instructions can handle variable as well as fixed-length records.



Type 270 Random Access Drum

A drum rotation speed of 1200 rpm, coupled with the use of 512 read/write heads, provides access to a specified drum record in an average of only 27.5 milliseconds. Data transfer to and from the drum takes place at an average rate of 102,000 characters per second.

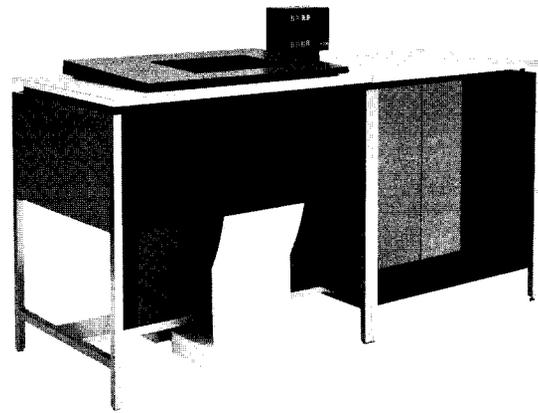
Only one memory cycle of central processor time is required for data transfer between the drum control and main memory. Therefore, the proportion of a data transfer interval available for other central processor operations varies from 69.6% for a Model 120 processor to 92.4% for a Model 4200.

The drum control automatically generates a parity bit for each character to be written. The parity bits accompany the record onto the drum. An automatic character-parity check is performed while reading; any discrepancy results in the setting of a program-accessible indicator. A file protection feature (Feature 075) prevents the accidental programmed alteration of data encoded on any programmer-assigned group of 64 data tracks. As many as eight groups of tracks may be individually protected.

## Printers

Honeywell offers printers to meet a wide variety of requirements. As indicated in the accompanying table, printing speeds offered range from 450 to 950 single-spaced lines per minute for alphanumeric characters and up to 1300 for lines containing a numeric character set; 96 to 132 print positions per line are available. Up to eight carbon copies can be provided.

Printing is performed in response to peripheral data transfer instructions issued to the printer control from the central processor. The peripheral control and branch instruction is used to handle such functions as line and form spacing. An edit instruction allows the



Type 222 Printer

programmer to arrange output data into any desired format.

During printing, an operator-changeable type roll on which characters are embossed moves past print hammers at each print position. Actuated as the proper character moves by, these hammers print the characters indicated by the print instruction. A cycle check technique insures the accuracy of printed information. Standard drums for Honeywell printers have 63 characters available at each print position — 26 alphabetic, 10 numeric, a blank symbol, and a number of special characters (e.g., credit symbol, asterisk, dollar sign, etc.). Each print position of the drum used for high-speed numeric printing has available a special 49-character set which is the same as the standard set except that it contains fewer special characters. Also available is a bar code drum that generates documents readable by the Type 289-8 Data Station Bar Code Reader. Two special symbols of the standard print drum are replaced by the left and right ortho bars in the bar code drum.

SUMMARY OF PRINTER SPECIFICATIONS

	450 LPM PRINTER	650 LPM PRINTER	950 LPM PRINTER
SPEED ATTAINABLE WITH NUMERIC PRINT FEATURE	NA	1300 LPM	1266 LPM
% OF DATA TRANSFER INTERVAL AVAILABLE TO CP FOR OTHER OPERATIONS	Model 120: 84% Model 4200: 99%	Model 120: 75% Model 4200: 98%	Model 120: 60% Model 4200: 97%
PRINT POSITIONS PER LINE	120	96, 108, 120, or 132	120 or 132
CHARACTERS PER PRINT POSITION	63	63 standards; 49 with Numeric Print Feature (034, 035)	
SKIP SPEED inches per second	Up to 55 in/sec attainable as successive lines are skipped.		
VERTICAL SPACING	6 or 8 lines per inch.		
TYPE	222-5, 122 <sup>1</sup>	222-1, -2, -3	222-4

<sup>1</sup> Type 122 Printer only available in Model 120 systems.

## Paper Tape Equipment

The Honeywell paper tape reader (Type 209) processes 5- through 8-level tape at the rate of 600 frames per second; the punch (Type 210) operates at 120 frames per second. Reading and punching, as well as tape rewind and runout on the reader, are under program control. Tape stops within a frame's



Type 209 Paper Tape Reader

length at the end of a reading or punching operation, thus ensuring reliable reading of the first and last frames in a record.

Paper tape control units can be conditioned by programmed instruction to process either codes of 5 and 6 levels or codes of 7 and 8 levels. This facility minimizes the amount of central processor time required for data transfer when processing 5- and 6-level tape.

Data transfer between the central processor and either the reader or the punch involves the central processor for only one memory cycle per 5- or 6-level frame and two memory-cycles per 7- or 8-level frame. Thus, the central processor is free during more than 99.9% of a paper tape read or punch data transfer interval to perform computations and other input/output operations.

Frame parity can be generated by programmed instruction in preparation for punching. Likewise, frame parity can be checked by the program when reading tape. The 210 punch sets a program-accessible indicator when the end of tape is sensed.

## Banking Equipment

The availability of a magnetic ink character recognition (MICR) sorter-reader and a multiple tape lister and controls ideally equips Honeywell systems to handle a full line of banking applications. The MICR reader-sorter operates at the speed of 1,560 documents per minute. The multiple tape lister, working in conjunction with the MICR reader-sorter, prints 1,563 twenty-two column lines per minute.

# 4 Data Communication Facilities

Honeywell provides a broad-scale data communication capability, the highlights of which are:

- Single-channel and multi-channel control units to handle an exceptionally wide array of communication lines, speeds, and terminal devices.
- Fast-access mass storage devices.
- Powerful processor communication features, including an automatic interrupt system, 5-through-8-level code handling capability, an interval timer, and a programmable real time clock.
- An advanced, multi-purpose remote terminal device, the Data Station.
- A full line of software for interrupt processing and message handling.

With the exception of the message-mode control, this entire communication capability is available for use in any Series 200 system, regardless of the processor model chosen by the user; the message-mode control cannot be used with Type 201-0 and 201-1 processors.

## Processor Communication Features

Several features available in Series 200 processors make them especially well suited to handle communication applications. The simultaneity, high internal speeds, and automatic hardware interrupt facility of these central units provide a very significant capability — *effective processing of communications and conventional jobs at the same time*. Flexibility in application design is provided by the ability to process ASCII as well as other 5-, 6-, 7-, and 8-level codes.

### SIMULTANEITY

The use of program-assignable read-write channels in Series 200 processors enables them to direct the data flow to and from several peripheral devices and, con-

currently, to perform computing operations. For example, the Model 200 processor can perform up to four input/output operations at the same time that internal processing is going on. Projected to the Model 4200 processor, this facility allows 16 peripheral data transfers to proceed simultaneously with computing. This greater throughput is simultaneity's chief contribution to integrated communication/business data processing systems.

### INTERNAL SPEED

Concurrent I/O and internal processing must be coupled with internal speeds high enough to allow efficient handling of data received or transmitted to provide an effective computing system. Honeywell memory cycle times ranging from 3 microseconds per character down to only 188 nanoseconds per character provide internal processing speeds which are suitable complements to the aforementioned simultaneity. These speeds enable complete processing of communications data even when transmission is at high-volume rates.

### AUTOMATIC INTERRUPT FACILITY

Available for use in all Series 200 processors is a completely automatic program interrupt facility. The advantage of this interrupt is that it enables simple but efficient direction of processing involving concurrent real time and business or scientific applications. The interrupt facility allows automatic branching, as necessary, between a main program and real time service routines. In particular, the readiness of a communication control to receive data for transmission or to relay data coming in from a line can automatically trigger entrance to a stored routine to service the external demand immediately. (Interrupt routines, applicable to most communication environments, are provided by Honeywell as part of the stan-

dard software.) Automatic signalling of control status obviates the necessity for programmed tests of these units to detect the arrival of data or the readiness to transmit. The interrupt facility also includes automatic storage of main program indicator values and control register contents, as well as an interrupt source indication.

### MULTI-LEVEL CODE-HANDLING FACILITY

All Series 200 processors are equipped with a facility enabling them to bring into memory and manipulate data in many different codes. This feature in-

cludes the ability to translate automatically between character codes of up to 12 levels.

### TIMING DEVICES

Two types of devices are available for use in Series 200 processors to give programs access to real time information; each requires one input/output trunk. The Type 213-3 Interval Timer provides automatic program interrupts at program-specified intervals. The Type 213-4 Time of Day Clock permits a program to determine the current clock time in hours, minutes, seconds, and tenths of seconds. These devices may

APPLICABILITY OF HONEYWELL COMMUNICATION EQUIPMENT

Terminal	Service & Line	Data Set <sup>3</sup>	Transmission Speed	Single-Channel Control Type	Adapter Unit Type <sup>1</sup>
DATASPEED <sup>4</sup> 2	Voice-Grade Private Line DDD	202D 202C	105 cps	281-1H	285-1H
DATASPEED <sup>4</sup> 5 RECEIVERS	Voice-Grade Private Line DDD	402C	75 cps	281-3A	285-3A
DATASPEED <sup>4</sup> 5 SEND UNITS	Voice-Grade Private Line DDD	402D	75 cps	281-4A	285-4A
DIGITRONICS DIAL-O-VERTER <sup>5</sup>	Voice-Grade Private Line DDD	202D 202C	150 cps	281-2C	285-2C
DIGITRONICS TYPE 1 DIAL-O-VERTER <sup>5</sup>	Voice-Grade Private Line DDD	201B 201A	300 cps 250 cps	281-2E	285-2E
FRIDEN COLLECTADATA <sup>6</sup> 30	Voice-Grade Private Line DDD	103F 103A	30 cps	281-1L	285-1L
Honeywell Series 200 Computer <sup>2</sup>	Voice-Grade Private Line DDD	201B 201A	300 cps 250 cps	281-2B	285-2B
	Telpak A 48 KC Broad-Band Channel	301B	5100 cps	281-2F	---
Honeywell Data Station	Voice-Grade Private Line DDD	202D 202C	120 cps	281-1M	285-1M
IBM 1050	W. U. 180 Baud Tel. Co. 150 Baud Voice-Grade Private Line	1181.1A* 816 103F	14.8 cps	281-1E	285-1E
	Tel. Co. TWX-CE Tel. Co. DDD	103A 103A	14.8 cps	281-1K	285-1K
IBM Standard STR Series (7702, 1013, 1009, etc.)	Voice-Grade Private Line DDD	202D 202C	150 cps	281-2A	285-2A
	Voice-Grade Private Line DDD	201B 201A	300 cps 250 cps	281-2D	285-2D
TTY 15, 19, 28	5-Level TTY Circuit	---	60, 66, 75, or 100 wpm	281-1C	285-1C
TTY 33, 35	TWX TWX-CE Tel. Co. 150 Baud DDD	811B	100 wpm	---	285-1N
		103A		281-1B	285-1B
		816		281-1D	285-1D
		103A		281-1B	285-1B
TTY 33, 35, 37 Model 1	Voice-Grade Private Line W. U. 180 Baud	103F 1181.1A*	100 wpm	281-1D	285-1D
UNIVAC 1004/DLT2	Voice-Grade Private Line DDD	201B	300 cps	281-2E	285-2E
		201A	250 cps		
UNIVAC 1004/DLT2B	Telpak A 48 KC Broad-Band Channel	301B	5100 cps	281-2F	---
W.U. TELEX	W. U. Telex	W. U. Adapter*	66 wpm	281-1A	285-1A

<sup>1</sup> References to adapter units imply Type 286 communication controls, since a 285 adapter interfaces each line connected to a multi-channel control.

<sup>2</sup> Type 281-2F single-channel control in Honeywell-to-Honeywell service is available in half-duplex and full-duplex forms.

<sup>3</sup> Except where indicated by an asterisk, data set designations refer to Bell System DATA-PHONE Data Sets.

<sup>4</sup> Trademark of American Telephone & Telegraph Co.

<sup>5</sup> Trademark of Digitronics Corp.

<sup>6</sup> Trademark of Friden, Inc.

be used in such applications as: (1) timing of program runs; (2) logging times of remote inquiries and information input; and (3) starting programs at specified intervals or clock times, as in polling a communication network.

## Data Communication Controls

Single-channel and multi-channel controls are available to enable Series 200 systems to receive and transmit data over toll and leased lines. One of the most outstanding features of these devices is the broad selection of lines, speeds, and terminal devices to which they can be connected — this selection is one of the largest offered by any manufacturer. The compatible services and equipment are indicated in the accompanying table.

### SINGLE-CHANNEL CONTROLS

The Type 281-1, -2, -3, and -4 Single-Channel Communication Controls direct the transmission and reception of messages in 5- to 8-level codes at rates of up to 5,100 characters per second, depending upon what common carrier line is selected. The four controls are distinguished from one another only by the lines and services to which they are connected; functionally, they are identical. These controls are half-duplex devices; i.e., messages are both transmitted and received, but not simultaneously. Additional controls can be added to a system in order to provide full-duplex or multi-channel operation.

### MULTI-CHANNEL CONTROLS

The Type 286-1, -2, and -3 Multi-Channel Communication Controls direct the transmission and reception of messages over as many as 63 half-duplex communication lines. The 286-1 services up to 3 lines, the 286-2 services from 4 to 15 lines, and the 286-3 can handle from 16 to 63 lines. A communication adapter unit is required to interface between the control and each line being used. The processor is interrupted as each character of a message is transferred. Data can be transferred at rates up to 300 characters per second in a single line.

### MESSAGE-MODE CONTROLS

The Type 286-4 and 286-5 Message-Mode Multi-Channel Communication Controls direct the transmission and reception of whole messages over as many as 63 half-duplex lines. The 286-4 services from 2 to 32 lines, and the 286-5 services from 33 to 64 lines.

A communication adapter unit is required as the interface between the control and each line being used. The total throughput capacity (all lines) of either control is 7,000 characters per second. Line-scanning priorities are established under program control.

Initially, a main memory input/output area address is stored in the message-mode control for each line connected to the control. When a character is received (or is to be transmitted) over a line, the address corresponding to that line is transferred directly and automatically to the central processor. The message character is then transferred to/from the main memory address indicated. The address of the next character position to be accessed in the main memory is always stored in the message-mode control, so that the next message character is once again transferred automatically. The processor is interrupted only when an entire message (or a designated portion thereof) has been transferred over any one line.

### DATA PROTECTION

The validity of data being communicated is protected by three different means:

- A transmission lapse results in the automatic setting of a program-accessible indicator in the receiving processor.
- Where applicable, the controlling program can initiate an automatic message-receipting system.
- For codes of more than 6 bits, a frame parity check is available. A long-check feature is also available.

### COMMUNICATION SWITCHING UNITS

Communication switching units of the 215 series enable any two Honeywell systems to share a group of communication lines or to switch between different groups of lines. Up to 63 lines may be switched simultaneously.

## Fast-Access Storage

A major requirement of many communication applications, such as those involving inquiry and message switching, is fast access to information which has been placed in storage. Of course, core memory provides the fastest access possible. But when dealing with large files, core memory becomes too expensive. To fill this need for economical storage, Honeywell offers the complete line of magnetic tape units, Mass Memory File transports, and the random access drum described in the preceding section.

## Data Station

The Honeywell Data Station is a multi-purpose remote terminal device which can be used for a broad range of communication applications, as well as for off-line jobs. This device gives branch offices, warehouses, remote reporting locations throughout a plant, or any other company outposts, the power to prepare source data locally and to communicate directly with a centrally located computer.

The Data Station features a very competitive transmission speed of 120 characters per second and a wide choice of input/output devices, including paper tape and punched card equipment, a keyboard, page printers, and an optical bar code reader that introduces new applications possibilities. The Type 288-1 Control Unit can handle a keyboard plus any combination of four of the peripheral devices included in the accompanying table.

### DATA STATION PERIPHERAL DEVICES

<i>Device</i>	<i>Speed</i>	<i>Type</i>
Card Reader	120 char/sec	289-7
Paper Tape Reader	50 char/sec	289-6
	120 char/sec	289-4
Paper Tape Punch	50 char/sec	289-6
	120 char/sec	289-5
Optical Bar Code Reader	50 char/sec	289-8
Page Printer and Keyboard	10 char/sec	289-2
	40 char/sec	289-3
Keyboard	10 char/sec	289-2A

<sup>1</sup> Registered trademark of American Telephone & Telegraph.



Data Station

## ON-LINE OPERATION

The Data Station transmits over a telephone-grade line using a DATA-PHONE<sup>1</sup> 202C or 202D dataset; party line operation is available. It uses the 8-bit ASCII code which includes parity. Communications can be directed locally by the operator or remotely by the central computer. Parity and long checking are used to protect against transmission errors. An automatic facility is available for initiating of retransmission of data in which errors have been detected.

## OFF-LINE OPERATION

When not being used for on-line operations, the Data Station can be used for local activities such as data preparation. By way of illustration, possible off-line operations include keyboard-to-paper-tape and card-to-paper-tape with simultaneous printing.

## Communication Software

To complement the capability of Series 200 hardware, Honeywell provides a full line of software to aid in implementing a variety of applications. Some of the functions performed by this software are:

**Interrupt Processing** — Determination of which communication line is demanding service and whether the line is incoming or outgoing; recognition of the priority of high-speed lines over low-speed lines in gaining access to processing routines, and optimal distribution of processing time to critical operations and less demanding operations.

**Real Time Input Analysis** — Conversion of the communication code to that used by the processor; interpretation of message headers if necessary; accumulation of messages in memory; generation of storage assignment requests; checking of message format for validity, generation of output queue requests; and addition of control information to messages.

**Output Stacking and Interfacing** — Generation of requests to processor for messages; reception of messages from output queues; transmission of messages; maintenance of queuing and line priority information.

**Random Access Storage and Retrieval** — Allocation of random access storage; reading and writing records; placing and retrieving queue entries; and performance of all message-queuing functions.

**Line Status Direction** — Determination of line availability; over-all control of line utilization.

# 5 Instructions

Series 200 computers possess a repertoire of over 50 powerful instructions that allow the programmer to specify all operations quickly and easily. Each instruction has a standard format which may range in complexity from one simple element — an operation code — to many elements. Each instruction includes an operation code which uniquely identifies the type of operation to be performed. Some instructions also include one or more variant characters which are used either to define an operation in greater detail or to specify literally a piece of data to be used in the operation. Most instructions also include one or two operand addresses.

When an instruction is extracted from memory for execution, the instruction elements are automatically loaded into particular control registers. Specifically, the operand addresses are loaded into the A- and B-address registers, and the variant character, if present, is loaded into the variant register. Thereafter, the contents of these registers are used in controlling the progress of the operation initiated.

Certain instructions can be stated in an abbreviated form to conserve storage space and to shorten instruction execution time. For example, the Extract instruction can be specified in any of the forms:

EXT/A/B  
EXT/A  
EXT

In executing an abbreviated instruction form, the information left in pertinent control memory registers from the execution of previous instructions is used in place of the unstated elements which would be included in the standard format of the instruction being executed. For example, in executing the abbreviated form EXT, the values remaining in the A- and B-address registers from the execution of the previous instruction execution will be used as the A- and B-operand addresses for the EXT operation. A series of instructions linked together in this manner by residual register contents is said to be “chained.”

It is usually possible to use the abbreviated form, Op Code/A, of instructions whose standard format contains two addresses. The actions resulting from using this form take one of two courses depending upon the instruction. In the execution of some instructions, the contents of the B-address register are used as the unstated B-operand address. In the execution of other instructions, the stated A address is used as the B-operand address.

## Instruction Descriptions

Each Series 200 instruction is described on the following pages in terms of four parameters: name, operation code, format, and execution time. The following format is used to describe all instructions.

Instruction Name		Mnemonic Op Code
Description of Instruction Execution		
EasyCoder	Standard and Abbreviated Instruction Formats	Execution Times (microseconds)

For those instructions which cannot be chained, the operations initiated by both standard and allowable abbreviated formats are described. In the case of instructions which can be chained, the operations initiated by abbreviated formats are quite similar to those for the standard formats; therefore, in the case of these instructions, only the operations initiated by the standard format are described. Abbreviated format operations can be inferred from these descriptions. The Op Code/A form of such instructions are footnoted to indicate whether the stated A address or the value in the B-address register is used as the B-operand address (see above).

The execution times given are not the fastest possible times but are based on realistic situations involving three-character addressing mode. The data fields referenced by both the A and B addresses are

assumed to be five characters long. Execution times are rounded off to the nearest whole number. The instruction summary table in Section 6 contains detailed timing information for each instruction.

### CONVENTIONS

The following symbols are used in the instruction descriptions to convey the meanings indicated.

SYMBOL	MEANING
A and B	Addresses of main memory locations.
(A) and (B)	The contents of the fields indicated by A and B.
V	Variant character.

The phrases "field indicated by A" and "field at A" are synonymous and refer to the data field whose rightmost limit is location A and whose leftmost limit is marked by the next word-marked character to the left of location A. The expression "instruction indicated by B" refers to the instruction whose operation code is stored in location B.

### Fixed-Point Arithmetic Instructions

Eight arithmetic instructions perform both decimal and binary arithmetic: decimal add, decimal subtract, binary add, binary subtract, zero and add, zero and subtract, decimal multiply, and decimal divide. The latter two instructions are not available in the Model 120.

Decimal arithmetic instructions treat their operands as signed numeric data. Normal algebraic sign control is in effect during the execution of these instructions. Any zone bit configuration other than 10 in the rightmost character of a decimal field causes the field value to be interpreted as positive; 10 indicates a negative field value.

If (A) is algebraically larger than (B), a recomplement cycle is performed automatically to convert the result to its true form.

An indicator is set at the completion of each decimal arithmetic operation to indicate the presence or absence of a zero result. A different indicator is set if overflow is sensed. The status of either of these indicators can be tested by a subsequent programmed instruction.

In binary arithmetic operations, the operands are treated as unsigned binary numbers; overflow and zero balance are disregarded.

Instruction Format

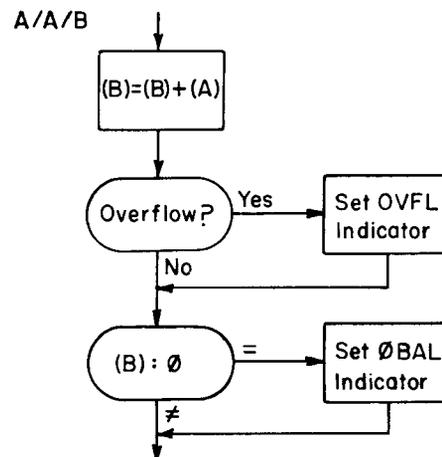
Execution Time  
120 200 1200 2200 4200

#### Decimal Add

A

(A) is added algebraically to (B), and the result is stored in the field at B. B-field zone bits are set to zero except when necessary in the units position to give a true result.

A/A,B	69	48	35	25	10
A/A <sup>1</sup>	60	42	30	22	9
A	51	36	26	19	9



Decimal Add Operation

#### Decimal Subtract

S

(A) is subtracted algebraically from (B), and the result is stored in the field at B. B-field zone bits are set to zero except when necessary in the units position to give a true result.

S/A,B	69	48	35	25	10
S/A <sup>1</sup>	60	42	30	22	9
S	51	36	26	19	9

#### Decimal Multiply

M

The signed decimal integer in the field at A is multiplied by the signed decimal integer in the high-order locations of the field at B. The product is stored in

<sup>1</sup> The stated A address is used as the B-operand address.

Instruction Format	Execution Time				
	120	200	1200	2200	4200

the low-order locations of the field at B.

M/A,B	—	419	316	200	81
M/A <sup>2</sup>	—	413	312	197	81
M	—	407	307	194	80

### Decimal Divide D

The signed decimal integer in the field at A is divided into the signed decimal integer in the field whose leftmost location is B. The quotient and the remainder are stored in the high-order and low-order locations, respectively, of the field at B.

D/A,B	—	219	164	134	50
D/A <sup>2</sup>	—	213	160	131	50
D	—	207	155	128	49

### Binary Add BA

(A) is added to (B), and the sum is stored in the field at B.

BA/A,B	69	46	35	24	9
BA/A <sup>1</sup>	60	40	30	21	9
BA	51	34	26	18	9

### Binary Subtract BS

The ones complement of (A) is added to (B), and a simulated carry is added to the sum of the low-order characters. The result is stored in the field at B.

BS/A,B	69	46	35	24	9
BS/A <sup>1</sup>	60	40	30	21	9
BS	51	34	26	18	9

### Zero and Add ZA

(A) is transferred, right to left, to the field at B. All zone bits in (B), other than those in the sign position (which are normalized) are cleared to zero.

ZA/A,B	54	36	27	19	8
ZA/A <sup>1</sup>	45	30	23	16	8
ZA	36	24	18	13	7

<sup>1</sup> The stated A address is used as the B-operand address.

<sup>2</sup> The value in the B-address register is used as the B-operand address.

Instruction Format	Execution Time				
	120	200	1200	2200	4200

### Zero and Subtract ZS

(A) is transferred, right to left, to the field at B with the opposite sign. All zone bits in (B), other than those in the sign position, are cleared to zero.

ZS/A,B	54	36	27	19	8
ZS/A <sup>1</sup>	45	30	23	16	8
ZS	36	24	18	13	7

## Floating-Point Arithmetic Instructions

Series 200 scientific processing operations, available in Models 1200, 2200, and 4200, use four 48-bit accumulators, a low-order register (LOR), and three floating point indicators for exponent overflow, divide check, and multiply overflow. At the most, two of the four accumulators are used in any one operation; these two are designated X and Y for purposes of description.

Many of the scientific processing operations may be performed in either of two forms: (1) accumulator-to-accumulator; and (2) memory-to-accumulator. Accumulator-to-accumulator operations use the contents of the designated floating-point accumulator for the A operand. Memory-to-accumulator operations obtain the A operand from the main memory field specified by the A address stated in the instruction.

### Floating Add AMA or AAA

AMA/A,XY: (A) is added to (X), and the sum is stored in Y.

AAA/XY: (X) is added to (Y), and the sum is stored in Y.

### Floating Subtract SMA or SAA

SMA/A,XY: The twos complement of (A) is added to (X), and the result is stored in Y.

SAA/XY: The twos complement of (Y) is added to (X), and the result is stored in Y.

**Floating Multiply MAM or MAA**

MAM/A,XY: (X) is multiplied by (A). The high-order product is stored in Y; the low-order product is stored in LOR. — — 68 45 22

MAA/XY: (X) is multiplied by (Y). The high-order product is stored in Y; the low-order product is stored in LOR. — — 56 37 18

**Floating Divide DMA or DAA**

DMA/A,XY: (A) is divided by (X). The quotient is stored in Y; the remainder is stored in LOR. — — 83 55 25

DAA/XY: (Y) is divided by (X). The quotient is stored in Y; the remainder is stored in LOR. — — 71 47 21

**Store Floating Accumulator TAM**

TAM/A,X-: (X) is stored in A. (X) is not altered. — — 24 16 6

**Load Floating Accumulator TMA or TAA**

TMA/A,-Y: (A) is loaded into Y. No normalization occurs. — — 26 17 7

TAA/XY: (X) is loaded into Y. No normalization occurs. — — 12 8 4

**Floating Test and Branch on Accumulator Condition FBA**

FBA/A,XC: The mantissa portion of (X) is tested for the condition specified by C, the low-order octal digit of the variant. If the condition specified by C is satisfied, program control branches to A. — — 15 10 5  
No Branch  
— — 18 12 5  
Branch

Otherwise, the program continues in sequence.

**Floating Test and Branch on Indicator FBI**

FBI/A,OD: The indicators specified by D, the low-order octal digit of the variant, are tested. If any of the indicators are set, control branches to A. Otherwise, the program continues in sequence. All indicators tested are reset. — — 12 8 4  
No Branch  
— — 15 10 4  
Branch

Branch Conditions for FBI Instruction.

D	INDICATOR	D	INDICATOR
0	No branch	6	Divide check or exponent overflow
1	Multiply overflow	7	Divide check, exponent overflow, or multiply overflow
2	Exponent overflow		
3	Exponent or multiply overflow		
4	Divide check		
5	Divide check or multiply overflow		

**Decimal to Binary DTB**

DTB/A,-Y: The 11-character signed decimal integer whose low-order character is A is converted to a 36-bit binary integer, which is stored in the mantissa portion of Y. — — 45 30 13

**Binary to Decimal BTD**

BTD/A,X-: The mantissa portion of (X) is con- — — 44 29 12

Branch Conditions for FBA Instruction.

C	CONDITION	C	CONDITION
0	No branch	4	(X) > 0
1	(X) = 0	5	(X) ≥ 0
2	(X) < 0	6	(X) ≠ 0
3	(X) ≤ 0	7	*Unconditional branch

verted from a binary integer to a signed decimal integer, which is stored in the 11-character main memory field at A.

**Store Low-Order Result TLM or TLA**

TLM/A: (LOR) is stored in A. No normalization occurs.	-	-	23	15	7
TLA/-Y: (LOR) is stored in Y. No normalization occurs.	-	-	9	6	3

**Load Low-Order Result TML or TAL**

TML/A: (A) is loaded into LOR. No normalization occurs.	-	-	23	15	8
TAL/X-: (X) is loaded into LOR. No normalization occurs.	-	-	9	6	3

**Binary Mantissa Shift BMS**

BMS/XM,V: If single precision, the mantissa of (X) is shifted in the mode specified by M, the low-order octal digit of the first variant. If double precision, mantissas of (X) and (LOR) are shifted. The second variant V specifies the number of bits to be shifted.	-	-	21	14	6
---	---	---	----	----	---

**Binary Integer Multiply BIM**

BIM/A,B: The four-character fields in memory whose low-order characters are A and B are treated as 24-bit binary integers. The integers are multiplied together; the product is stored in the field specified by B.	-	-	54	36	17
---	---	---	----	----	----

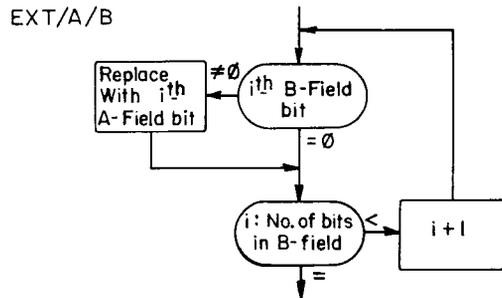
## Logic Instructions

Nine instructions are included in this category. Extract, Half Add, and Substitute manipulate data on an individual bit basis, combining bits from two different fields according to rules based on AND and OR logic. Each of the remaining instructions causes a program branch to be performed unconditionally or contingent upon the existence of a precisely defined condition.

**Extract EXT**

Each 1-bit in the field at B is replaced with the corresponding bit from (A); all zeros in (B) are undisturbed.

EXT/A,B	69	46	35	24	9
EXT/A <sup>1</sup>	60	40	30	21	9
EXT	51	34	25	18	8



Extract Operation

**Half Add HA**

The binary fields at A and B are added without carry, and the result is stored in the field at B.

HA/A,B	69	46	35	24	9
HA/A <sup>1</sup>	60	40	30	21	9
HA	51	34	25	18	8

**Substitute SST**

Each bit in the character at B which corresponds to

<sup>1</sup> The value in the B-address register is used as the B-operand address.

Instruction Format	Execution Time				
	120	200	1200	2200	4200

a 1-bit in V is replaced by the corresponding bit in the character at A; other B-bits remain undisturbed.

SST/A,B,V	36	24	18	13	7
SST/A,B	33	22	17	12	7
SST/A <sup>1</sup>	24	16	12	9	7
SST	15	10	8	6	6

### Compare

C

(B) is compared bit-by-bit with an equal number of characters in the field indicated by A; indicators are set to show the result of the comparison.

C/A,B	57	38	29	21	9
C/A <sup>1</sup>	48	32	25	18	9
C	39	26	21	15	8

### Branch

B

The contents of the sequence register are stored in the B-address register, and a program branch to the instruction at A is performed.

B/A	18	12	9	7	5
-----	----	----	---	---	---

### Branch on Condition Test

BCT

If the indicator specified by V is set, the contents of the sequence register are stored in the B-address register and a program branch to the instruction at A is performed; otherwise, program continues in sequence.

BCT/A,V	21	14	11	8	5
BCT	9	6	5	4	4

Partial list of variant characters for BCT instruction.

VARIANT	INDICATOR	VARIANT	INDICATOR
00	Unconditional		
01	SENSE Sw 1 ON	41	$B < A$ (low compare)
02	SENSE Sw 2 ON	42	$B = A$
		43	$B \neq A$
		44	$B > A$ (high compare)
04	SENSE Sw 3 ON	45	$B \neq A$
10	SENSE Sw 4 ON	46	$B = A$
		50	Overflow
		60	Zero Balance

The conditions in the righthand (or lefthand) column can be combined.

<sup>1</sup> The value in the B-address register is used as the B-operand address.

Instruction Format	Execution Time				
	120	200	1200	2200	4200

### Branch on Character Condition

BCC

If the character at B contains the type of punctuation and/or sign indicated by V, the contents of the sequence register are stored in the B-address register, and a program branch to the instruction at A is performed; otherwise, the program continues in sequence.

BCC/A,B,V	36	24	18	13	6
BCC/A,B	33	22	17	12	6
BCC/A <sup>1</sup>	24	16	12	9	6
BCC	15	10	8	6	5

Partial list of variant characters for BCC instruction.

VARIANT	BRANCH CONDITION
02	B-bit is 1
06	Negative sign
10	Word mark or record mark
12	B-bit is 1 and word mark bit is 1
16	Negative sign and word mark bit is 1
20	Item mark or record mark

### Branch if Character Equal

BCE

A program branch to the instruction at A occurs if the character at B is the same as V; otherwise, the program continues in sequence.

BCE/A,B,V	36	24	18	13	7
BCE/A,B	33	22	17	12	6
BCE/A <sup>1</sup>	24	16	12	9	6
BCE	15	10	8	6	5

### Branch on Bit Equal

BBE

The single character at B is combined bit-by-bit with V according to the rules shown below. This logical product is tested but is not stored in memory. If the result is not equal to zero, the contents of the sequence register are stored in the B-address register, and a program branch to the instruction at A is performed. If the result equals zero, the program continues in sequence.

BBE/A,B,V	36	24	18	13	6
BBE/A,B	33	22	17	12	6
BBE/A <sup>1</sup>	24	16	12	9	6
BBE	15	10	8	6	5

## General Control Instructions

The instructions in this category are used to manipulate data within the control memory, to prepare main memory storage areas for the processing of data fields, and to control the sequential selection and interpretation of instructions in the stored program.

### Set Word Mark SW

Word marks are set in the locations specified by A and B; the data in these locations are undisturbed.

SW/A,B	30	20	14	11	5
SW/A <sup>1</sup>	21	14	11	8	5
SW	12	8	6	5	4

### Set Item Mark SI

Item marks are set in A and B; the data in these locations are undisturbed.

SI/A,B	30	20	14	11	5
SI/A <sup>1</sup>	21	14	11	8	5
SI	12	8	6	5	4

### Clear Word Mark CW

Locations A and B are cleared of word marks; the data at these addresses are undisturbed.

CW/A,B	30	20	15	11	5
CW/A <sup>1</sup>	21	14	11	8	5
CW	12	8	6	5	4

### Clear Item Mark CI

Locations A and B are cleared of item marks; the data at these addresses are undisturbed.

CI/A,B	30	20	15	11	5
CI/A <sup>1</sup>	21	14	11	8	5
CI	12	8	6	5	4

Formation of Logical Product in BBE Instruction.

BIT IN B FIELD	BIT IN V	RESULTING BIT
1	1	1
1	0	0
0	1	0
0	0	0

<sup>1</sup> The stated A address is used as the B-operand address.

### Halt H

H: The machine is halted unconditionally. The sequence register contains the address of the instruction following the halt.

H/A: The contents of the sequence register are transferred to the B-address register, and A is placed in the sequence register; then the machine is halted.

H/A,B: The machine is halted unconditionally. A and B are stored in the address registers as halt identification symbols.

H/A,B,V: The machine is halted unconditionally. A and B are stored in the address registers, and V is stored in the variant register.

H	9	6	5	5	4
H/A	18	12	9	8	5
H/A,B	27	18	14	11	5
H/A,B,V	30	20	15	12	6

### No Operation NOP

The contents of the sequence register are incremented; no other operation is performed.

NOP	9	6	3	3	3
-----	---	---	---	---	---

### Change Addressing Mode CAM

The addresses in all following instructions are interpreted as specified by V until the next CAM instruction is executed. Either or both of the following conditions may be specified by V:

1. Addressing mode (2-, 3-, or 4-character).
2. The "trap" mode of instruction execution. This mode is used to trap operation codes which are not implemented in a given installation and to provide an automatic branch to routines which simulate the non-existent operation codes.

CAM/V	12	8	5	5	3
CAM	9	6	3	4	3

### Change Sequencing Mode CSM

CSM: The contents of the sequence register and the change sequence register are interchanged, and the program branches to the address which was previously stored in the change sequence register.

CSM/A: The operations described above are performed. In addition, the A address is loaded into

Instruction Format	Execution Time				
	120	200	1200	2200	4200

the A-address register.

CSM/A,B: The operations described above are performed. In addition, the B address is loaded into the B-address register.

CSM/A,B,V: The operations described above are performed. In addition, the variant character is loaded into the variant register.

CSM	12	8	5	5	4
CSM/A	21	14	9	8	5
CSM/A,B	30	20	14	11	5
CSM/A,B,V	33	22	15	12	5

### Store Control Registers SCR

The value in the control register indicated by V is stored in the field indicated by A. V can be specified to indicate any control register except those used as floating-point accumulators.

SCR/A,V	30	20	15	12	5
SCR/A	27	18	14	11	5
SCR	18	12	9	8	4

### Load Control Registers LCR

The value in the field indicated by A is moved to the control register indicated by V. V can be specified to indicate any control register except those used as floating-point accumulators; if the sequence register is specified, a program branch to the instruction at A is performed.

LCR/A,V	30	20	15	12	5
LCR/A	27	18	14	11	5
LCR	18	12	9	8	4

### Load Index/Barricade Register LIB

The single character at A is loaded into the index/barricade register, specifying the number of a 4,096-character portion of main memory whose low-order character location is the low-order boundary of a protected memory area. The high-order boundary of the protected area is the high-order location of memory. The 15 additional index registers which function with a protected memory area are located in the low-order character locations of the portion designated by the character at A.

LIB/A	21	14	11	8	5
LIB	12	8	6	5	5

Instruction Format	Execution Time				
	120	200	1200	2200	4200

### Store Index/Barricade Register SIB

The contents of the index/barricade register are stored in A. Unused high-order bit positions in A are set to zero.

SIB/A	21	14	11	8	5
SIB	12	8	6	5	4

## Interrupt Control Instructions

The normal processing sequence can be interrupted by any one of four sources: (1) a peripheral control; (2) the operator's control panel or console; (3) an internal processor condition which violates a protected portion of memory; or (4) a programmed instruction. Program control automatically branches to a stored routine which identifies the source of interruption and services the condition which caused the interruption.

Four instructions are used in conjunction with the automatic interrupt facility. The Monitor Call instruction is one of the four interrupt sources. The remaining three instructions, coded in the interrupt routine, help identify the source, store information which will be needed when the normal processing sequence is resumed, restore this information, and cause the return to the interrupted program.

### Monitor Call MC

The processor is interrupted as follows: the source indicators are set to show that the Monitor Call instruction is the source of interruption, the settings of pertinent processor indicators are stored, the arithmetic indicators are cleared, the contents of the sequence register and the external interrupt register are interchanged, the program branches to the address previously stored in the external interrupt register, and the processor enters the three-character, non-trap, external interrupt mode of operation.

MC	9	6	3	4	3
----	---	---	---	---	---

### Store Variant and Indicators SVI

Up to six characters of indicator and register status information, as specified by V, are stored in consecutive locations following V (see accompanying table).

SVI/V	27	18	15	11	6
-------	----	----	----	----	---

Instruction Format	Execution Time			
	120	200	1200	2200 4200

**Restore Variant and Indicators RVI**

Up to five consecutive characters starting at A are loaded into the processor control indicators and registers specified by V. These characters were stored previously by an SVI instruction. The correspondence between V bit positions and the registers and indicators whose previous contents are being restored is the same as for the SVI instruction, except that bit 5 of V is not used by RVI.

RVI/A,V	33	22	15	12	6
---------	----	----	----	----	---

**Resume Normal Mode RNM**

The contents of the pertinent interrupt register and the sequence register are interchanged, and program control branches from the interrupt routine being executed to the address contained in the sequence register. Pertinent control information, previously stored when interruption occurred, is restored, and A and B (if stated) are stored in the respective address registers.

RNM/A,B	30	20	14	12	5
RNM/A <sup>1</sup>	21	14	9	9	5
RNM	12	8	5	6	3

## Data Move Instructions

These instructions are used to move data within the main memory. Control over the placement of punctuation can be exercised, and data can be translated to other forms as it is moved.

**Move Characters to Word Mark MCW**

Data and item marks are moved from the field indicated by A to the field indicated by B.

Information Stored by SVI Instruction.

BIT	CORRESPONDING REGISTERS AND INDICATORS
0	Variant register
1	Arithmetic, comparison, address mode, and trap mode indicators
2	Auxiliary indicators register
3	Scientific unit indicators
4	Storage protection indicators; in some cases, the internal interrupt mode indicators
5	Interrupt source indicators

<sup>1</sup> The value in the B-address register is used as the B-operand address.

Instruction Format	Execution Time			
	120	200	1200	2200 4200

MCW/A,B	54	36	27	19	8
MCW/A <sup>1</sup>	45	30	23	16	8
MCW	36	24	18	13	7

**Load Characters to A-Field Word Mark LCA**

The data and punctuation in the field indicated by A are moved to the field indicated by B.

LCA/A,B	54	36	27	19	8
LCA/A <sup>1</sup>	45	30	23	16	8
LCA	36	24	18	13	7

**Extended Move EXM**

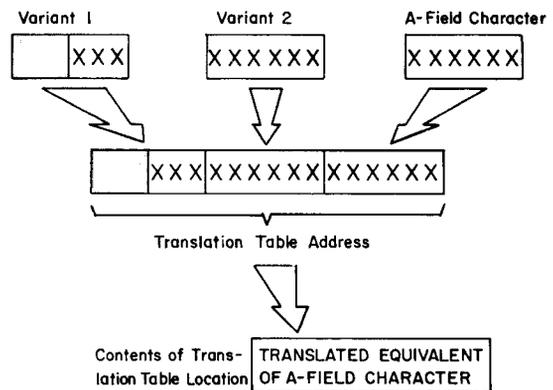
A specified amount of the data and/or the punctuation in the field indicated by A are moved to the field indicated by B. Different variant characters can be specified to control:

1. whether or not punctuation marks are moved;
2. the type(s) of punctuation moved;
3. whether or not data is moved;
4. the direction in which the source field is to be scanned; and
5. how much of the source field is to be scanned.

EXM/A,B,V	57	38	29	20	9
EXM/A,B	54	36	27	19	8
EXM/A <sup>1</sup>	45	30	23	16	8
EXM	36	24	18	13	7

**Move and Translate**

The characters in the field indicated by A are successively translated according to the contents of a stored



Formation of Translation Table Address of Equivalent

translation table; their equivalents are placed in corresponding successive locations in the field indicated by B. The translation table address (binary) of an equivalent character is formed by placing together the binary equivalents of  $V_1$ ,  $V_2$ , and the character to be translated, in sequence.

This instruction and MIT, below, find particular application in the solution of code conversion problems. For example, it can be used to convert easily and efficiently between central processor codes and peripheral media codes having different configurations.

MAT/A,B,V<sub>1</sub>,V<sub>2</sub>                      72    48    36    28    11

**Move Item and Translate**                      **MIT**

The code characters in the item indicated by A are successively translated according to the contents of a stored translation table; their equivalents are placed in the item indicated by B. Characters to be translated and their equivalents may be up to 12 bits in length; the size of these characters (i.e., whether contained in one or two six-bit character locations) is specified by  $V_3$ . The translation table address of an equivalent character is formed by combining  $V_1$ ,  $V_2$ , and the character to be translated in a manner similar to MIT, above.

MIT/A,B,V<sub>1</sub>,V<sub>2</sub>,V<sub>3</sub>                      75    50    38    26    14

## Edit Instruction

**Move Characters and Edit**                      **MCE**

The contents of the field indicated by A are moved into replaceable character positions in the edit control word in the field indicated by B; and then the contents of the B field are edited to suppress unneeded credit and minus symbols, zeros, and commas, and to insert blanks, asterisks, and dollar signs where necessary.

MCE/A,B                                      99    66    50    34    14  
MCE/A<sup>1</sup>                                        90    60    45    31    14  
MCE    81    54    41    28    13

<sup>1</sup> The value in the B-address register is used as the B-operand address.

## Input/Output Instructions

Effective control over data transfers between the central processor and peripheral units and over the peripheral units themselves is maintained by the use of two basic instructions: Peripheral Data Transfer (PDT) and Peripheral Control and Branch (PCB). The PDT instruction is used to initiate data transfer operations and certain other related operations, such as backspace magnetic tape and erase magnetic tape.

The PCB instruction can perform four distinct functions: (1) cause a program branch to be performed contingent upon the setting of peripheral condition indicators; (2) change the operational mode of a peripheral control; (3) initiate strictly mechanical (non-data transfer) operations; and (4) allow a peripheral control to interrupt (or direct it not to interrupt) the central processor when data transfer is completed.

**Peripheral Data Transfer**                      **PDT**

PDT/A, C<sub>1</sub>, C<sub>2</sub>, . . . . . , C<sub>n</sub>: Data are transferred between the field indicated by A and the control unit and peripheral device indicated by control characters C<sub>2</sub> and C<sub>3</sub>, using the read/write channel indicated by C<sub>1</sub>. The number of control characters (n in the instruction format) varies with the device being addressed.

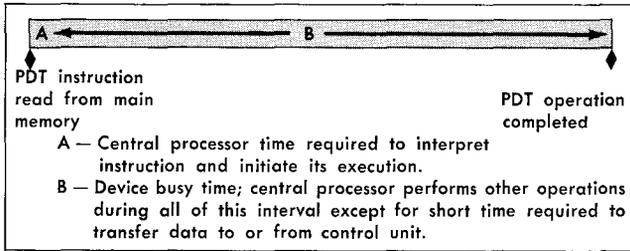
A write operation is terminated when all of the data in the output field has been transferred. A read operation is terminated when the memory input area has been filled or when a standard unit of information, such as a record or a card, has been read.

The timing of a PDT instruction involves three considerations:

1. the time required by the central processor to interpret the instruction (eight memory cycles average time);
2. the time required by the processor to transfer data to or from the peripheral control involved (approximately one memory cycle per character); and
3. the amount of time the peripheral device is busy (varies according to the device involved).

The amount of central processor time required during the entire peripheral operation is relatively small (from .001% to approximately 30%, depending on the processor and peripheral device involved).

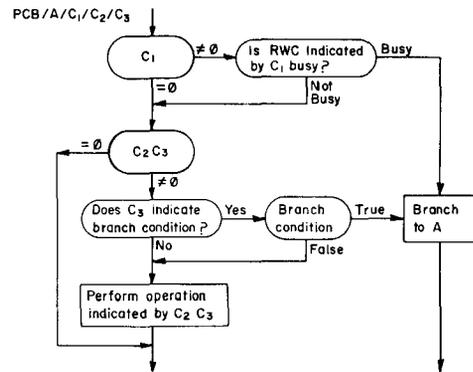
TIME RELATIONSHIPS IN EXECUTION OF PDT INSTRUCTION



Peripheral Control and Branch PCB

PCB/A,C<sub>1</sub>, . . . ,C<sub>n</sub>: If C<sub>1</sub> specifies a read/write channel, the program branches to the instruction at A if the channel is busy. If the channel is not busy, or if no channel is specified, the operation(s) specified by control characters following C<sub>1</sub> are performed. These operations are summarized below.

1. Test and branch operations – PCB instruction configurations are available to test for conditions such as peripheral control busy, error in last card punched, loss of transmission error detected by communication control, etc. If the condition exists, a program branch to the instruction at A is performed.
2. Mode change operations – The PCB instruction can direct the central processor to condition a specified peripheral control for operating in a specific mode. For instance, the card reader control can be conditioned to read subsequent cards in Hollerith mode or indirect transcription mode, depending on the control characters of the PCB instruction. That control could also be directed to reject cards which contain illegal punches, to generate a busy signal if illegally punched cards are read, or both, depending on the particular control characters specified.



Peripheral Control and Branch Operation

3. Single-occurrence peripheral device operations – The PCB instruction can be used to direct the single performance of a non-data transfer peripheral operation (for example, rewind magnetic tape or seek a mass memory file tape strip).
4. Peripheral interrupt operations – Any Series 200 peripheral control can be allowed to interrupt (or directed not to interrupt) the central processor. If allowed, the interrupt signal is generated automatically at the conclusion of a data transfer operation.

Central processor time required for executing a PCB instruction varies with the length of the particular instruction and the processor under discussion. Instruction length varies in turn with the type of operation specified, with A-address length, and with whether or not performance of the instruction includes a program branch. In any case, very little central processor time is required (approximately eight memory cycles).

# 6 Programming Systems

Series 200 hardware reflects the most advanced technology in the computer industry. To complement the hardware, Honeywell has developed a comprehensive array of software that capitalizes on the processing capabilities of the hardware. The software is supplied in several versions tailored to fit various equipment configurations and operating environments. The equipment configurations may cover a range of many different memory capacities and peripheral device combinations.

A distinctive feature of Series 200 programming aids is that they are operationally compatible with each other and the object programs that they produce. Object programs produced by a variety of program preparation aids as well as programs from the software library may all be intermixed on run tapes and processed together.

Program compatibility is a built-in feature of Series 200 software. A single machine language is used with all of the Models 120/200/1200/2200/4200, allowing the user to run on the Model 4200 programs written for the smaller machines. Thus, software and software-produced object programs which run on even the smallest processor can also run on the larger configurations, and with considerable gain in performance due to the faster cycle times and increased input/output simultaneity of the larger processors.

The Series 200 software available now and planned for the future is grouped into two general categories:

Series 200/Programming Systems — Software which performs computer management functions such as language processing, program checkout and maintenance, operation control, I/O control, data editing and transcription, and mathematical processing.

Series 200/Application Systems — Special-purpose software which performs jobs closely related to the functions of the user's organization (e.g., inventory control and linear programming). These systems are described in Section 7.

Series 200/Programming Systems are of two types:

the Series 200/Basic Programming System, consisting of self-loading, unit-record programs for the 4K to 12K environment, and the Series 200/Operating System. The Series 200/Operating System is supplied in two implementation levels:

Series 200/Operating System—Mod 1 — Applies generally to the range 12K to 65K; and

Series 200/Operating System—Mod 2 — Applies generally to the range 65K to 524K.

## Series 200/Basic Programming System

The Series 200/Basic Programming System is designed for card-oriented Series 200 installations which assemble and execute programs individually. Each program in the Series 200/Basic Programming System is an independent entity consisting of facilities for self-loading, data manipulation/specialization, and diagnostic analysis. The prime requirements of an "open shop" installation, where jobs are scheduled and executed on a demand basis, are flexibility and simplicity of operation. These requirements are reflected in the design features of the Basic Programming System, enabling a self-loading, unit-record type of operation for Series 200 processors having core memory capacities in the approximate range of 4K to 12K characters.

The associated peripheral array for the Series 200/Basic Programming System need include only a card reader and a card punch, although most installations will also have use for a high-speed printer. Magnetic tape drives may be added to this array, providing a more efficient storage medium. Alternatively, program storage and data input/output functions can be allocated to paper tape equipment. Finally, the Basic Programming System includes provisions for storage of frequently used systems or production programs on a self-loading tape (SLT) to increase the efficiency of a card-oriented environment.

## PROGRAM PREPARATION AND MAINTENANCE

The program preparation and maintenance function includes language processing as well as program editing and maintenance.

### LANGUAGE PROCESSING

Among the language processors in the Series 200/Basic Programming System are three Easycoder assemblers (Easycode Assembly A, A(P), and B), COBOL Compiler B, and two Liberator conversion programs, Bridge and Easytran. COBOL Compiler B may be used separately or as an integral part of a conversion system called Easytab which permits users to effect a transition for tab equipment to a computer system.

#### Easycoder

The Easycoder assemblers provide a choice of storage media which includes punched cards, magnetic tape, and paper tape. Easycoder symbolic programs may include calls to macro routines which are stored on either punched cards or card-image tapes. If the source program includes macro statements, a library processor program is used prior to assembly to specialize the desired routines according to the parameters furnished in the macro statements and to insert each routine into the program at the requested point(s). The output of the library processor then becomes the input to an Easycoder assembler.

The output of Easycoder assembly can be punched cards, magnetic tape, or punched paper tape, depending on the needs of a given application.

#### COBOL Compiler B

The COBOL B language is a simple programming tool which employs commonly used English-language business terms. COBOL Compiler B translates the programmer's COBOL source statements into a machine-language program, i.e., a program intelligible to and executable by the computer.

Honeywell's COBOL compilers are noted for their compilation speed and the efficiency of the resulting machine-language programs. COBOL B requires only 8,192 characters of memory for compilation, the least amount of memory yet required for COBOL, but it provides all the language elements necessary to express virtually any business-oriented data processing function.

#### Liberator

Liberator conversion programs embody a uniquely

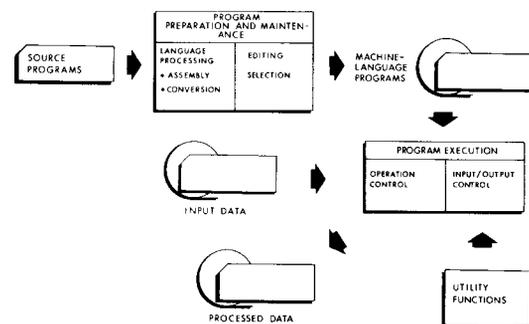
effective approach to competitive language translation which allows users of IBM 1400-series systems to harness the superior throughput and cost/performance characteristics of Series 200 processors without reprogramming. Program translation can be performed on either the symbolic- or machine-language level. Additional functions such as input/output operations can also be performed, including appropriate conversions and/or substitutions for competitive routines, thus producing converted programs with optimum operational characteristics on Series 200 processors.

#### Easytab

The Easytab system is an effective transition tool for tab equipment users who are moving up to computers. It consists of two elements: a set of preprogrammed utility routines (see below) for the performance of common tab functions, and COBOL B for those jobs which do not lend themselves to performance by the utility routines. Easytab retains tab procedures but substitutes the speed and processing capabilities of Series 200 computers for the slower, less efficient operation of tab equipment. The basic processing unit is still an 80-character item, thus retaining much of the basic structure of the tab user's present data processing operations, and thereby precluding costly restructuring of data files and facilitating the orientation of tab installation personnel to the computer environment.

## PROGRAM EDITING AND MAINTENANCE

The program editing and maintenance facilities include the ability to create and maintain files of programs on self-loading magnetic tapes. The tape files are created in card-image format, thereby retaining the operational characteristics of punched card programs. Facilities are also included for copying programs from an existing tape file onto a new tape, inserting new programs, and replacing or deleting old programs.



Functions and Data Flow in the Basic Programming System

## OPERATION CONTROL

Operation control is provided by the applicable loader for the medium from which the object program is loaded into memory. The loader provides for optimal use of available core memory through the facility for loading programs on a segmented basis, thereby allowing subsequent segments to overlay the instructions of a previous segment after they have been executed. In addition, the loader transfers control to the starting location of the object program after the program is loaded into memory. The loader also provides for load-and-halt or load-and-start operation. These characteristics apply to both the card and paper tape loaders. For the magnetic tape loader, the self-loading features of the card-image object programs are increased by the addition of a search routine which enables several programs to be stored in a tape file, from which the desired programs can be selected and loaded according to program name.

## INPUT/OUTPUT CONTROL

Data input/output operations are performed by macro routines which are specialized to the user's requirements. Magnetic tape input/output, paper tape input, communication control, and console typeins and typeouts are among the operations performed. For example, tape operations such as checking tape labels, identifying files, blocking and unblocking records, and checking for read/write errors and end-of-reel conditions are performed by tested routines which permit the user to handle input/output operations with simple macro statements. Among the communication control functions performed are interrupt processing, real time input analysis, output stacking and interfacing, and random access storage and retrieval.

## UTILITY FUNCTIONS

The Basic Programming System employs an extensive set of utility routines which perform a variety of data transcription, editing, and mathematical processing functions. Data transcription and editing packages perform sorting and collating of data stored on magnetic tape, tape handling, media conversion, e.g., card-to-tape, report generation, and tabulator simulation. An extensive library of routines oriented to scientific tasks complement the processing capabilities of the Basic Programming System. This library includes basic routines to perform square root, exponential, trigonometric, and logarithmic functions, as

well as matrix, statistical, and other more comprehensive routines.

Easytab includes seven preprogrammed (in COBOL) utility routines which handle the bulk of the tab user's workload. These routines are ready to use as soon as the user receives them; he need only compile the routines (a one-time operation), insert parameter cards to specialize the routines to his applications — and programs are ready to run. Only one or a few parameter cards are required to specialize the utility routines to a given application.

The functions performed by the Easytab utility routines are sorting, merging, selecting, altering, totaling, reproducing, and performing basic input/output functions (for which there is no tab equivalent).

## SUMMARY OF FEATURES

- Flexibility and simplicity
- Self-loading, unit-record operation
- Increased efficiency in card environment
- Choice of storage media
- Superior throughput and cost/performance characteristics
- Competitive language translation
- Ease of maintaining and updating program files
- Optimal use of available memory
- Macro routines specialized to user's requirements

## Series 200/Operating System

The Series 200/Operating System is designed for implementation in various types of environments at differing stages of growth. The user may select the desired functions, the degree of centralized control for these functions, and the methods of implementation which are specifically oriented to his equipment configuration and operating environment. At all hardware levels, the operating system provides automatic control using minimum hardware overhead. The characteristic fixed overhead, which makes many operating systems economically unfeasible, is eliminated by the modular design of the Series 200/Operating System. The user is thus able to apply the system according to his needs without incurring costs for unnecessary functions and features.

## IMPLEMENTATION LEVELS

The operating system is designed in two implementation levels to relieve the user of a host of complex programming and execution supervision tasks. The

Mod 1 implementation level of the operating system is the unifying element for medium-scale, tape-oriented installations having three or more tape drives and minimum core memory capacities of 12K characters. Flexibility at the Mod 1 level permits efficient use of mass storage, communications, paper tape, and punched card devices in both independent and semi-centralized operations. The Mod 2 implementation level utilizes two main types of environments in which core memory capacities range from 65K to 524K characters: Either an all-tape or a mixed tape-and-mass-storage environment may be employed.

The Mod 2 all-tape environment permits the user to effect a gradual transition from the semi-centralized job control functions of the Mod 1 level to the totally centralized job control functions of Mod 2. The operating system takes advantage of new hardware facilities as they are added. Moreover, program compatibility is assured as the installation grows. Continuously mounted tapes provide extensive storage capacity, and techniques have been devised to provide efficient processing in several operating modes. Source and systems subprograms can be stored on several tapes and/or partly stored on punched cards to avoid lengthy tape searching and copying.

The Mod 2 mixed tape-and-mass-storage environment provides an optimum random access processing capability in a totally integrated system whose standard mode of operation is oriented to continuous, stacked-job processing — a technique whereby “open-shop” installations can achieve rapid turnaround time, especially during checkout phases of program development. In this environment, a mass storage device is used for the storage of both systems and object programs so that they can be accessed randomly for program generation or execution. Magnetic tapes can be used for intermediate work storage.

## PROGRAM PREPARATION AND MAINTENANCE

This category includes three subfunctions: language processing, program editing, and program updating.

### LANGUAGE PROCESSING

The user is provided a choice of language processing programs with specific versions that are tailored to his installation. These programs perform the following translations:

1. symbolic (EasyCoder) language into machine language;

2. commercial (COBOL) language into machine language;
3. scientific (FORTRAN) language into machine language; and
4. competitive languages into Honeywell Series 200 language on both source- and machine-language levels.

### *EasyCoder Assemblers*

The assemblers for the Series 200/Operating System consist of two elements — a symbolic language and an assembly program which translates source programs written in the symbolic language into machine language. The assemblers take full advantage of large tape- or mass-storage-oriented configurations to provide both speed and flexibility. Programming with the EasyCoder Assembly Language is greatly simplified through the use of macro instructions which cause the generation of appropriate sequences of machine instructions or the insertion of library routines into an object program. Symbolic program analysis and specialization of symbolic library programs are also performed by the assembly program. Symbolic program analysis includes the process of extracting symbolic tags, references (to each tag, to index registers, and to absolute addresses), and calls to library routines from the symbolic input programs. Symbolic library processing is provided as an extension of the assembly language. By writing macro calls, the programmer can cause the incorporation of precoded assembly-language routines in his program.

### *COBOL Compilers*

The English-language statements of COBOL provide a relatively machine-independent method of expressing a business-oriented problem to a Series 200 computer. Commonly used nouns, verbs, and connectives are used in the procedural portion of a COBOL program to construct easily understood sentences. The excellent documentation provided by COBOL — problem definition as well as method of solution — enables more than one programmer to work on a particular problem with minimal duplication of effort.

To complement the modularity of the Series 200 hardware, the various COBOL compilers implement a set of language modules, expanding the features of COBOL as the machine capacity is increased. This design approach allows the COBOL user to enhance the power of the source language and to produce larger object programs as the need arises.

The four Series 200 COBOL compilers are syntax-directed; the smallest version can operate in a configuration consisting of three magnetic tape units, a card reader, an on-line printer, and a processor containing an 8,192-character memory. Other compilers are available for memory sizes of 16,384 characters and larger. Honeywell COBOL compilers are known for their high performance, and the smallest version of the Series 200 compiler is no exception: Compilations of typical programs on a Model 200 processor require on the order of one to two minutes.

All Series 200 COBOL compilers are modularly expandable and self-adapting to memories larger than the minimum. They accept batched source programs and will operate in a batch-compile, load-and-go mode.

Series 200 COBOL compilers possess several significant operating features:

1. Library facilities for source-language files.
2. An object-time, data distribution system, plus dumping facilities to expedite program testing.
3. Dynamic reassignment of read/write channels at object time.
4. Fast diagnostic scans for new programs.
5. Integrated output listing containing imbedded diagnostics, memory map, and cross reference to machine language.

### FORTRAN Compilers

Series 200 FORTRAN consists of two basic elements: a source language (FORTRAN IV) whose structure closely resembles the language of mathematics, and compilers which translate the statements and formulas written in the source language into Series 200 machine-language programs.

Programs are written directly as algebraic expressions and arithmetic statements. Additional statements, such as transfer, decision, indexing, and input/output statements, control the processing of the algebraic expressions and arithmetic statements. The smallest compiler version translates a major portion of FORTRAN IV, including logical statements and testing, data initialization, labelled COMMON areas, and type statement declarations. Even more sophisticated language elements are accepted by the larger versions.

All Series 200 FORTRAN compilers are designed for rapid compilation and optimum efficiency of object coding. Translated programs can be combined with other previously compiled and assembled programs and immediately executed to obtain fast results. The smallest version requires as few as 16,384 characters

PROGRAMMER		FOR		PAGE		OF	
PROGRAM		DATE		IDENT		NO	
SERIAL	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
IDENTIFICATION DIVISION.							
PROGRAM-ID. UPDATE.							
DATE-WRITTEN. FEBRUARY 10, 1965.							
DATE-COMPILED. FEBRUARY 11, 1965.							
AUTHOR. JOHN PARNELL.							
INSTALLATION. JONES DEPARTMENT STORE.							
SECURITY. CONFIDENTIAL TO COMPANY.							
REMARKS. THIS PROGRAM READS PURCHASE RECORDS FROM A MAGNETIC							
TAPE FILE, UPDATES AND EDITS INFORMATION IN EACH RECORD, AND							
PRODUCES A TOTAL-RECORDS MAGNETIC TAPE FILE; ERROR PRINTOUTS							
ARE PROVIDED WHERE NECESSARY.							
ENVIRONMENT DIVISION.							
CONFIGURATION SECTION.							
SOURCE-COMPUTER. MODEL-200							
OBJECT-COMPUTER. MODEL-200; MEMORY SIZE 16384 CHARACTERS.							
S. TAPE-UNITS, I READER-PUNCH, I PRINTER.							
SPECIAL-NAMES. CONSOLE-TYPEWRITER IS ERROR-PRINTER.							
INPUT-OUTPUT SECTION.							
FILE-CONTROL. SELECT PURCHASE-FILE, ASSIGN TO TAPE-UNIT AA, TAPE-UNIT							
AB. SELECT TOTAL-FILE, ASSIGN TO TAPE-UNIT AC, TAPE-UNIT AD.							
I-O-CONTROL. APPLY TRAILING-COUNT ON PURCHASE-FILE, TOTAL-							
FILE.							
DATA DIVISION.							
FILE SECTION.							
FD PURCHASE-FILE; BLOCK CONTAINS 50 RECORDS; LABEL RECORDS							
ARE STANDARD; VALUE OF IDENTIFICATION IS "PURCHASE"; DATA							
RECORD IS PURCHASE-RECORD.							
01 PURCHASE-RECORD							
02 LOCATION NUMBER; PICTURE IS 99999.							
02 ORDER NUMBER; PICTURE IS 9999.							
02 STOCK-NUMBER; PICTURE IS 99999999.							
02 UNITS-ON-HAND; PICTURE IS 999.							
02 UNIT-PRICE; PICTURE IS 999.							
02 UNIT-CODE PICTURE IS XX.							
WORKING-STORAGE SECTION.							
77 ERROR-INDICATOR-1 PICTURE IS 99, VALUE IS ZERO.							
01 LABEL-RECORDS							
02 DATE, OCCURS 5 TIMES.							
03 MONTH PICTURE IS 99.							
03 YEAR PICTURE IS 99.							
PROCEDURE DIVISION.							
BEGIN SECTION.							
OPEN FILES OPEN INPUT PURCHASE-FILE; OUTPUT TOTAL-FILE.							
COMPUTING SECTION.							
READ-LOOP. READ PURCHASE-FILE; AT END GO TO END-PROCESSING.							
IF STOCK-NUMBER IS NOT NUMERIC GO TO BAD-STOCK-NUMBER.							
COMPUTE TOTAL-PRICE. MULTIPLY UNITS-ON-HAND BY UNIT-PRICE							
GIVING TOTAL-PRICE; ON SIZE ERROR GO TO EXCEEDS-ROUTINE.							
MOVE TOTAL-PRICE TO EDITED-PRICE.							
TRANSFER-DATA. MOVE PURCHASE-RECORD TO TOTAL-RECORD.							
ERROR-SWITCH. GO TO PRODUCE-OUTPUT.							
PRODUCE-OUTPUT. WRITE TOTAL-RECORD. GO TO READ-LOOP.							
ERROR-1 SECTION.							
BAD-STOCK-NUMBER. ALTER ERROR-SWITCH TO PROCEED TO							
MARK-STOCK-NUMBER. GO TO COMPUTE-TOTAL-PRICE.							

### COBOL Source-Language Program

of memory, plus four magnetic tape units, a card reader, card punch, and printer. Larger versions, which exploit the added features and instructions of the scientific hardware option, can process programs utilizing very large core storage capacities, up to 524,000 characters. Special features of the FORTRAN compilers include object code optimization and a highly sophisticated diagnostic system.

### Easytran Symbolic Translator

The Operating System's Easytran conversion program accepts as input symbolic source programs written in SPS and/or Autocoder language. The source program is completely analyzed and then translated statement by statement. During this process, most symbolic statements are replaced on a one-for-one basis with equivalent Easycoder statements. Those

statements which have no direct Easycoder equivalent are replaced either with in-line macrocoding or with calls to Easytran subroutines which perform the desired functions; those whose functions are automatically handled by Series 200 hardware are deleted.

The principal output of Easytran is a symbolic program in the proper form for input to an Easycoder assembler. Additional outputs include a parallel listing of the Autocoder and Easycoder symbolic programs, a cross-reference listing of all labels (tags) used in the input program, and an English-language diagnostic listing which points out any areas where programmer hand-tailoring may be required.

The translations described above define the capabilities of the four types of language processing provided in the Series 200/Operating System. For optimal program preparation, the user can select the method(s) of language processing which best conform to his requirements. Each of the language processors produces programs in a standard format. Programs produced by any of these language processors can be combined so that they can be executed in a single sequential operation.

#### *PROGRAM EDITING*

The output of language processing is used as input to a program editing process which is directed by definitive control statements to produce an executable object program. The purpose of editing is to combine into a program those subprograms that have been individually compiled, assembled, or converted. The requested subprograms are selected from the appropriate files and memory addresses are modified for loading. Programs are then ordered into sequential files for execution.

#### *PROGRAM UPDATING*

The program updating function includes such processes as storage and modification of both systems programs and user programs. Through the correction of existing programs, the addition of new programs, and the deletion of unwanted programs, files may be created or updated which contain only the required systems programs and subprograms in the sequence most suited to job requirements.

#### *PROGRAM EXECUTION*

Program execution functions include operation control, input/output control, and program test control.

#### *OPERATION CONTROL*

Operation control is the most significant element of an operating system and includes the interrelated functions of job scheduling, program sequencing, and multiprogramming. Job scheduling involves the allocation of memory and peripheral facilities among jobs. Jobs are made up of a series of logically related programs which are usually independently generated and made compatible through the common interface of the operating system. Jobs may be selected for execution on the basis of assigned priorities. The ability to alter assigned priorities at any time gives the operating system scheduling flexibility.

Program sequencing involves two principal operations: program selection and program loading. The operating system provides facilities for storage, modification, and maintenance of binary executable programs on one or more master tapes. From these master tapes, programs may be selected and ordered in a sequence most suited to specific run requirements in order to create tapes having only the required systems and processing programs.

The area of memory into which a program is to be loaded is specified at object time, thus increasing the operational flexibility afforded the programmer. Program loading makes efficient use of core memory, since object programs are segmented into optimum-size loading units so that only the required portion of a program is loaded into memory at any one time. The loading process retains the same functional structure in all storage media and incorporates the operating features that are most practical and convenient for a given hardware/software complex.

Multiprogramming is based on a control interrupt process in conjunction with other hardware facilities. This process provides the ability to automatically interrupt a job or program (on the basis of specified priorities) for a peripheral or communication demand and to recover after the demand is serviced. Throughput is increased by the simultaneous operation of programs that share equipment resources. Whenever a job is using only a portion of the system or is delayed due to input/output operations, efficient reallocation of memory cycles saturates the processing capacity of the system and allows programs to be sequenced independently of each other. In addition, the multiprogramming capability provides great flexibility in scheduling equipment and jobs.

## INPUT/OUTPUT CONTROL

The primary input/output control functions are: initiation of all data transfers (including dynamic allocation of input/output channels) and error detection and correction. In addition, the input/output control function includes the performance of secondary logical operations such as checking tape labels, checking file identification, and blocking and unblocking records.

On the Mod 1 level, input/output control functions are decentralized to reduce system overhead to a minimum. For example, source program calls for input/output control operations are replaced by specialized macrocoding at assembly or compile time. The result is a flexible system for the control of operations in small-to-medium-sized equipment configurations. On the Mod 2 level, all such functions are centralized for maximum throughput efficiency.

## PROGRAM TEST CONTROL

The program test function is performed by a collection of routines which may be used separately or as part of an automatic checkout subsystem. The checkout subsystem can process several programs or a job and automatically produce the necessary documentation to evaluate the programs. Processes such as automatic sequencing from one program to the next, printouts of messages and operator instructions, generation of test data, tape dumps, and dynamic, terminal, and emergency memory dumps are among the program test operations performed.

## UTILITY FUNCTIONS

An integral part of the operating system is an extensive set of utility programs which performs a variety of data transcription, editing, and mathematical proc-

esses. Sorting and collating of data stored on magnetic tape, tape handling, media conversions, report generation, and tabulator simulation are some of the data transcription and editing functions available. The mathematical processing capability provides an extensive library of routines oriented to scientific jobs. These routines perform operations such as square root, exponential, trigonometric, and logarithmic computation as well as numerical analysis, matrix, statistical, and other more comprehensive processes. All the routines related to the mathematical processing function can be used with or without the scientific hardware option.

## SUMMARY OF ADVANTAGES

- Increases throughput by more effective use of the hardware/software complex
- Relieves the operator of detailed and burdensome execution supervision
- Minimizes idle time and job setup time
- Minimizes turnaround time
- Modular design permits operation with only the needed functions and features
- Enables running of user-written programs, library, and systems programs as integral parts of the operating system
- Provides execution of stacked jobs without operator intervention
- Standardizes operating procedures
- Allows tailoring and specialization of precoded, fully tested library routines to meet program needs
- Enables automatic monitoring and documentation of test and production runs
- Provides for segmentation of programs to conserve memory space

# 7 Application Systems

To supplement its hardware and computer management software, Honeywell provides its customers with application systems, or "packages," to facilitate the use of their computers. These application systems are of two types: industry-oriented and general-purpose.

The industry-oriented packages perform functions peculiar to a specific industry. CART, for instance, automatically rates shipments for trucking companies; STET is a typesetting program for printers and publishers; CASH does the accounting for distribution companies, and so on. These packages incorporate the systems know-how that Honeywell has acquired through many years of experience in working with customers in a large variety of industries.

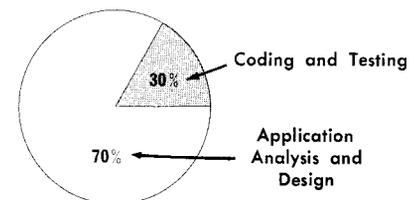
The general-purpose packages are oriented to the solution of complex management problems common to most industries. These packages employ operations research techniques to assist management in tasks such as optimally allocating available resources, scheduling and monitoring research and development projects, and other jobs which cut across industry lines.

Application systems are extremely valuable to a company whether it is taking its first venture into electronic data processing or acquiring a replacement computer to provide greater data processing capability. By applying the appropriate application system, the user can significantly reduce the amount of effort required in planning his initial and subsequent applications, developing an integrated system, and solving a variety of management problems.

## Industry Application Systems

A company installing a computer faces a sizable system startup cost. This cost is made up primarily of the salaries of systems and programming personnel needed to analyze, design, and implement the desired computer applications. The system startup cost can be

divided into two major elements, as represented in the following diagram.



The major portion of the systems cost is the problem-solving phase; that is, determining in detail:

- the objectives of the job;
- the inputs and outputs of the job, and
- how the job can be performed by a data processing system.

As indicated above, this phase constitutes the largest portion (typically 70 percent) of system startup costs; the remaining 30 percent covers the coding and testing (translating the system into computer language).

Honeywell application packages provide a solution to the greater portion of the system startup problem. Each one consists of an English language narrative, general and detailed flow charts, item designs, program logic, and input/output formats. In addition, to ease implementation problems, Honeywell provides fully coded and tested programs for applications which are of a general nature and common to the needs of many companies.

### THE BUILDING-BLOCK APPROACH

Honeywell recognizes that while it is true that the data processing needs of many companies are similar, they are not identical. Therefore, Honeywell's application packages are composed of "building blocks" at both the system design and programming levels. The

building-block approach facilitates individual specialization and tailoring of systems.

The building-block philosophy is also present in Honeywell's over-all approach to application packages for a specific industry. Each package is developed as a subsystem of an integrated management information system. Individual packages are designed to meet the needs of a restricted area of the organization, but with the needs of the whole organization in mind. Thus, the particular package can be used as a separate entity or as part of an integrated system. Once a user has put a basic application into operation, other system functions can be computerized with a minimum of additional information, time, and money.

The following industries are among those served by Honeywell application packages:

- Distribution
- Manufacturing
- Insurance
- Publishing and Printing
- Motor Freight
- Education
- Banking
- Finance

Additional application packages for these and other industries are continually under active analysis and consideration. Following are brief descriptions of Honeywell's available industry application systems.

## PRINTING AND PUBLISHING

Honeywell has several systems to offer the newspapers, magazines and general publishing companies. These include the two STET (Specialized Technique for Efficient Typesetting) system for handling hyphenation and/or justification for hot-metal or photocomposition typesetters and an integrated classified advertising system.

STET-1 is a comprehensive typesetting system which provides facilities for the layout, justification, and hyphenation of all types of copy (straight material, classified copy, and tabular material) for hot-metal linecasters. STET-1 is one of a series of linked subroutines written to facilitate modification, extension, and integration into a large system.

A STET monitor is also available which allows a single central processor to do both accounting and typesetting on a shared time basis.

Photo-STET is a generalized hyphenation and justification system written primarily for photocomposition typesetters, but which is adaptable to other composing

devices. Photo-STET includes all the layout facilities of the STET program but has added messages which permit the exploitation of the full flexibilities of any existing photocomposition device. These include disk or grid changes, point-size changes, line spacing, and many others. Care has been taken to ensure compatibility between the STET hot-metal and the Photo-STET (six-level only) input routines; this enables the same operator to keyboard either hot-metal work or photocomposing in a shop using both processes.

All the STET programs have been designed to be readily incorporated into integrated systems. Additional applications are being explored in the areas of classified advertising, book cost estimating, and subscription fulfillment.

## MANUFACTURING

FICS—Forecasting for Inventory Control System—is a highly automated method of inventory management designed to help manufacturers achieve three principal objectives: (1) to improve customer service, (2) to maintain inventory at lowest possible cost, and (3) to reduce management involvement in routine decision-making.

With FICS, the computer makes all the decisions for forecasting and inventory maintenance once governing policy has been established and incorporated into the program in the form of decision rules. An advanced exponential smoothing technique is used in combination with the traditional components of inventory control—economic order quantity, reorder point, and safety stock—to accomplish the management objectives.

The two primary components are: (1) a set of programs which maintain a sales history file, analyze and update forecast parameters, forecast sales, and calculate inventory control parameters; and (2) documentation which describes precisely the formats of the interfacing data flow between FICS and the other functions in the user's data processing system.

A simulator (ISIM) is also provided as part of FICS to test the system's accuracy and to predict the effects of proposed changes in policy. ISIM accepts as input live data concerning the user's products and then simulates and analyzes inventory conditions. It can therefore be used to test the applicability of FICS to a user's inventory before implementing this system.

## TRANSPORTATION

Honeywell has three application programs for the transportation industry. They are called *CART*, *Dispatch*, and *Schedule*.

*Dispatch* schedules the transportation for a distribution operation. It groups orders by truck or carload and provides the best delivery route consistent with predetermined travel and unloading times.

CART – Computerized Automatic Rating Technique – performs the highly complex task of rating shipments for trucking companies. CART handles the total rating job for commodity, class, and exception rates for the various tariff bureaus. Due to the variety of routes and types of shipments involved and the tight controls of the Interstate Commerce Commission and state regulating agencies, the task of rating manually is time-consuming and error-prone. CART automatically applies the proper freight rate, as well as pertinent accessorial charges, to shipments. Then it transmits the rated bills, or PRO's, to the terminals responsible for delivering goods to consignees.

In addition to shipment rating, CART enables transportation companies to improve operations by virtue of its accounting and management reports as well as

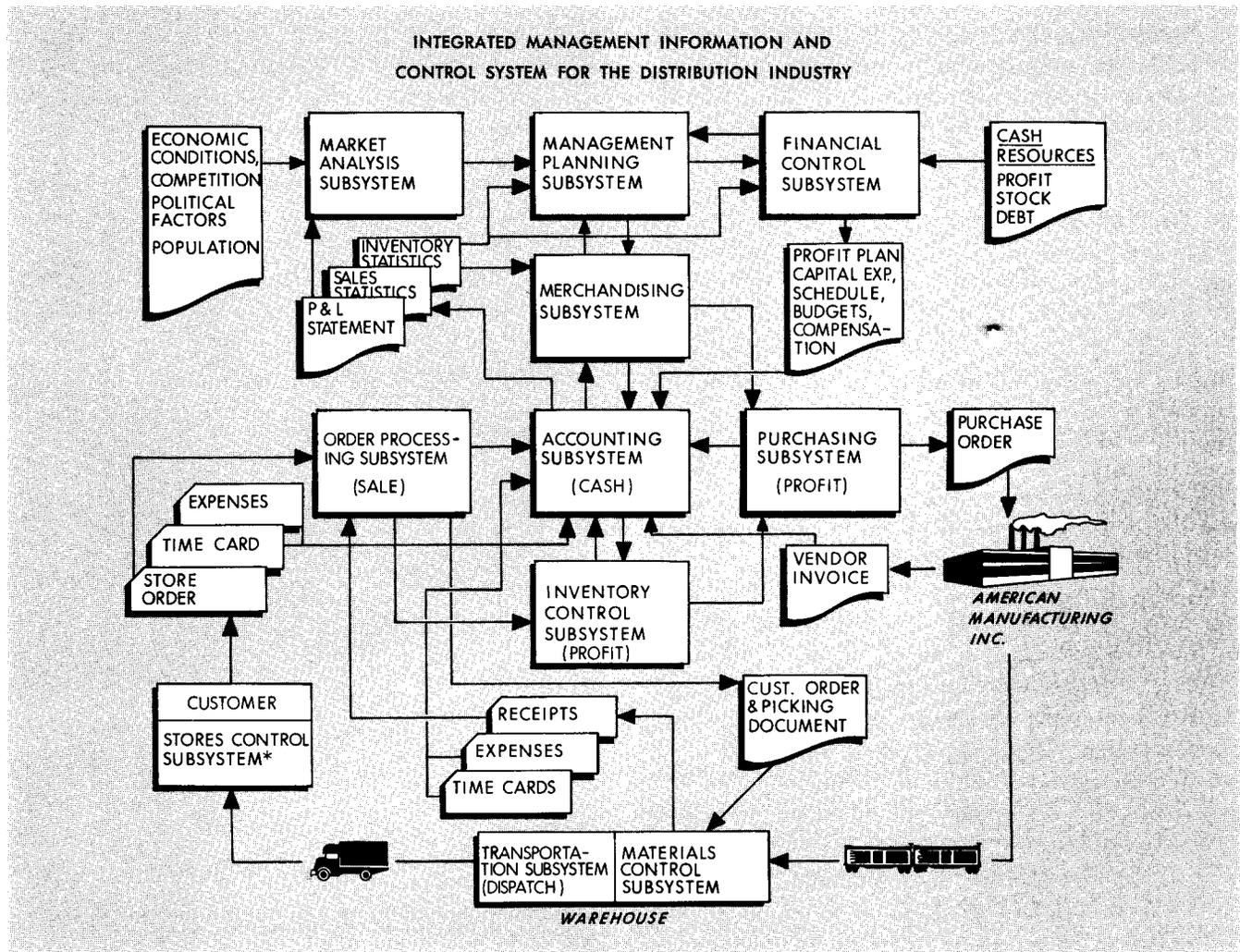
by providing valuable input to other trucking and computer functions.

*Schedule* is a run-cutting program that performs the difficult task of assigning operators to local transportation runs. With *Schedule*, the task is simplified and expedited; the scheduler is given all possible alternative combinations of straight and split runs and pieces of work, each combination closely approximating a normal 8-hour work day. *Schedule* is extremely flexible, enabling it to adapt to the shifting schedules created by holidays, weather, and special events such as parades and sporting events.

## DISTRIBUTION

Honeywell's integrated systems for the distribution industry are based on four fundamental applications:

*Sale* – an order-processing application which enters orders, issues customer invoices and warehouse



picking documents, and updates the appropriate accounting records.

**CASH** – Computerized Accounting System by Honeywell – an accounting application which integrates accounts receivable, accounts payable, and general ledger accounting, and also summarizes and analyzes data for management reports.

**PROFIT** – Programmed Reviewing, Ordering, and Forecasting Inventory Technique – a dynamic order-strategy application for inventory control and purchasing which optimizes inventory levels and replenishment orders at the most desirable customer service level.

*Dispatch* – an application that schedules the transportation for a distribution operation; it groups orders by truck or carload and provides the best delivery route consistent with predetermined travel and unloading times.

*Sale* handles order processing, keeping track of goods at the warehouse by recording each item's identity, quantity, and bin location. This information is used to provide picking documents for warehousemen, customer invoices, and reorder information for the distributor's purchasing subsystem.

**CASH** is an integrated accounting system that concentrates on three areas: accounts receivable, accounts payable, and general ledger. From the transactions entered in these areas, **CASH** produces various sales and purchasing analyses as well as a number of weekly, monthly, quarterly, and annual reports.

**PROFIT** is an inventory management tool which enables the distributor to maintain, *at least cost*, the requisite inventory to support a desired level of customer service. Taking over the time-consuming and complex tasks involved in inventory management, **PROFIT** reviews the inventory records and determines when to order and then issues the required purchase orders. **PROFIT** uses exponential smoothing, a statistical forecasting method, to estimate future demand based on an exponentially weighted moving average of past demands.

**PROFIT** determines the optimum inventory level for each item in the warehouse, using in its considerations the desired level of customer service, past demand history, previous levels of forecast error, vendor lead times, vendor discounts, and lot size. It also saves money by joint replenishment, generation of economic order quantities, and by balancing purchasing and carrying costs to achieve the least total cost for the inventory.

*Dispatch* maximizes the efficiency of shipment deliveries and facilitates the scheduling and routing of trucks from a single point to many destinations. Using order data, the requested time of arrival, and constraints such as number of trucks available, travel time and unloading time, *Dispatch* applies mathematical techniques in order to yield assignment sheets which list sizes of trucks to be employed, orders which are to be grouped together, routes to be followed, and times of departure from the shipping terminal.

The above applications may be used in building-block fashion to construct the core of a total system such as the one depicted in the illustration. These Honeywell-supplied applications can also be used separately or in various combinations as the user desires.

## BANKING

Honeywell has two applications packages for the banking community. One is a Demand Deposit Accounting System (DDAS) which includes a MICR entry run and the other is a Teller Unit Monitoring Program for on-line banking.

*Demand Deposit Accounting* – Banks of any size may avail themselves of the Honeywell Demand Deposit Accounting System. The system's modularity





makes it suitable to small, medium, or large banks and permits specific tailoring to individual user needs.

The DDAS functions include: check sorting, settlement accounting, clearing and collection, customer posting, exception procedures (stop drafts, large checks, overdrafts, etc.), special services for large depositors, and miscellaneous services such as reports and statements.

The MICR Entry Run program is a part of the DDAS, but is also a complete application package. For example, it contains input, output, print and check-digit routines in separate sections which can be easily replaced or modified.

*Teller Unit Monitoring* — The Teller Unit Monitoring Program is a flexible, modular system which processes on-line transactions and inquiries from remotely located teller terminals such as window machines, video display units and audio-response devices. Under control of this program, Series 200 computer systems accept, process, and transmit on-line messages as normal batch processing proceeds simultaneously.

Taking advantage of the program interrupt feature of Series 200 computers, the program automatically halts batch processing, assumes control and brings the teller message into computer memory. The program then retrieves the pertinent account record from the

random access file and processes transactions against the record for updating purposes. When this operation is completed, the program assembles the proper reply and initiates its transmission back to the inquiring terminal.

The Teller Unit Monitoring Program is modular in structure and contains a series of individual transaction and inquiry routines to handle various types of messages from a variety of terminal types. The routines may be used selectively, deleted, or replaced by specially designed user routines to meet specific requirements of individual banks. Although designed primarily to handle transactions for savings and mortgage loan accounting, the program layout is readily expandable to a full-scale, on-line, central information file system to meet the needs of other banking applications.

## FINANCE

Banks, brokers, investment fund trustees, mutual fund administrators, investment counselors, and insurance companies are the potential users of Honeywell's financial analysis application package. Anyone who invests in (or gives advice on) common stocks can benefit by taking advantage of the application package called Computerized Portfolio Analysis.

The programs in this package enable a subscriber to Standard & Poor's Compustat Tape Service to evaluate and compare the status and performance of several companies rapidly and simultaneously. The system is capable of calculating various important ratios and indicators of a company's growth, stability and prospects. It also produces a company-versus-industry comparison allowing an even more informed analysis to be performed. The analysis can also be used to select companies which meet criteria specified by the investment analyst. The programs perform the complete and lengthy calculations necessary to produce the analytical reports.

With Honeywell's portfolio analysis programs, investors and investment counselors can analyze and compare stocks efficiently and rapidly and thus gain a better basis for investment decisions.

## LIFE INSURANCE

SOLO — (System for Ordinary Life Operations) is Honeywell's total system approach to data processing in the life insurance industry. Outstanding features of the SOLO system are the policy master record development, the integrated daily cycle, new issue and

reserve calculation functions, and its modular design which allows it to be adapted to individual company needs.

Basic inputs common to all life insurance operations are defined to provide a common ground for linking the user's current operations with SOLO. The policy master file, general ledger file and agent master file are all updated in the daily cycle. Complete policy status is prepared on all automatic file update transactions and for each new issue entering the system. Premium notices are prepared and data are collected for month-end reports. A new issue building-block operation accomplishes the issuance of ordinary life policies and their entry into the policy master file. Inquiries, policy exhibit, agency accounting and other functions normally associated with daily business are incorporated into the SOLO daily cycle.

To round out the total systems approach, SOLO includes a means of preparing monthly, quarterly and annual reports. A reserve calculation and cash value system is provided. Mortgage loan processing, stock and bond updating and new rate book calculations are also included in the SOLO system.

## FIRE AND CASUALTY INSURANCE

FACILE (Fire And Casualty Insurance Library Editions) is a total data processing system developed for the fire and casualty insurance industry. Among the more important data processing functions encompassed by FACILE are: handling of new business, policy changes, agent accounting, claims handling, etc. In addition, the system integrates several files into a single master file and prepares reports on an exception basis.

FACILE is open-minded and modular in design. It is an expandable system designed to provide optimal data processing with a variety of systems configurations.

Not only are a company's present business problems efficiently and effectively handled, but the basic building-block design of FACILE provides expansion to solve tomorrow's problems without compromising today's systems requirements. Through this philosophy, a company can select components applicable to today's particular operation and at the same time provide easy transitional steps to a larger and more powerful system as the company grows.

## EDUCATION

Honeywell has developed a comprehensive group of application packages to form the nucleus of a total

school information system. The broad areas of application which lead to an improvement in the quality of education are pupil management, instructional aids, and educational research.

*Pupil Management* — In this area, Honeywell's PROF (Pupil Registering and Operational Filing) includes applications which perform cumulative record maintenance, pupil assignment, attendance reporting, grade reporting, and test analyses. These packages are aimed at alleviating teachers' and administrators' paperwork burdens and automating the maintenance and analysis of pupil data for use in career and curriculum planning.

Registration data, as well as addresses, permanent record information, yearly update, etc., are processed in PROF's cumulative record maintenance system. In the pupil assignment system, pupils are assigned to classes under the following constraints: No conflict may exist in period, day, or semester; only a predetermined number of consecutive study periods may be selected; pupil assignments to course sections must be balanced. If assignment cannot be made after attempting the thousands of possible schedules, a conflict report is printed.

PROF's attendance accounting system relieves the teacher of clerical drudgery. Beginning with pupil registration, PROF maintains attendance records and performs many accounting chores such as calculating the average daily and monthly attendance.

The grade-reporting system which is a part of PROF maintains cumulative pupil records and produces report cards, transcripts, academic lists, and guidance reports. Guidance reports are provided to counselors automatically — as problems are detected — or on request.

PROF's test analysis system permits the establishment of local standardization norms for tests. It produces reports which relate individual school ratings (percentiles, stanines, grade equivalences, etc.) to local, state, and national norms. Test scores are also recorded in each pupil's cumulative record.

*Educational Research* — PROF has an item of long-term significance in its educational research facility. A permanent file, maintained on tape for each pupil from the day he first enters the system, provides the raw data necessary for a variety of statistical studies having important implications in the planning of his future curricula. Correlation studies and factor analyses relating items such as academic achievement and standardized test scores with indices of postgraduate academic and career success can be carried out effectively on Series 200 computers.

*Instructional Aid* — A most significant application of the computer in a school system is its use as an instructional device. EDP training is becoming vitally important in areas such as mathematics, science, business, and general education. Honeywell has adapted WORDCOM,<sup>1</sup> a simplified programming language, to Series 200 for training beginning students in computer technology. With WORDCOM, the teacher can inject realism and practical experience into the teaching of computer fundamentals.

## General-Purpose Application Systems

Another group of programs uses operations research techniques for planning, scheduling, and project management. These programs are called the linear optimization packages and are based on two approaches: linear programming and network modeling. Both types provide optimum solutions involving large sets of linear constraints.

### LINEAR PROGRAMMING

Linear programming packages are used to determine the best assignment of resources such as money, machines, manpower, and materials among competing activities so that some value (stated in mathematical terms) is either maximized, e.g., profit, or minimized, e.g., cost. The result shows the most efficient method of operation, given a set of existing conditions. It can also show how much the conditions can change before another method is more efficient.

For example, linear programming might be used to determine the lowest cost combination of grains and other nutrients to put in an animal feed mix and still meet prescribed minimum nutrient requirements for each feed. It then might be desirable to find what effect a relaxation or increase in nutrient requirements might have on the cost of a feed mix. It would also be possible to determine how much of a change could be made in the amount of each grain in a feed mix without changing the cost of the feed.

The linear programming approach is one of complete simultaneity. All activities are simulated at once; no time relationship or sequence is established among

<sup>1</sup> A complete description of WORDCOM can be found in *Automatic Data Processing*, second edition, by Gregory and Van Horn, published by the Wadsworth Publishing Company, Belmont, California.

activities. The answer is quantitative in terms of an amount of dollars, man-hours, space, etc., to assign to various activities.

### NETWORK MODELING

The network modeling packages — Critical Path Method (CPM) and *Manage* — determine the time relationships among project activities, and this information can be interpreted in terms of cost.

*Critical Path Method* — The CPM package assists the project manager in estimating and controlling the time and technical performance required to gain project objectives. The first step in using CPM is to establish the project's major events and supporting activities and to develop a network. Then, a time estimate specifying the expected duration is applied to each activity. The program then sums the activity time estimates over the network to give the total expected elapsed time for every forward path through the network. The expected elapsed time for the end event of the project is of special importance: the longest expected time path from the beginning event to this event is the *critical path*. Any slippage in an activity on the critical path delays the whole project.

Using the CPM output reports, the project manager can identify critical and noncritical activities and can expedite activities selectively, never expediting any but critical activities and avoiding across-the-board overtime and premium costs.

CPM is particularly useful in planning projects where cost and time estimates for each activity can be made accurately because of abundant previous experience (such as in building construction and automobile manufacture).

*Manage* — Honeywell has developed a specialized use of the network model for planning computer installations and applications. *Manage* assists in planning all phases of a computer's installation and in evaluating alternative approaches to objectives, enabling the manager to monitor and control progress.

*Manage* even permits the user to consider the complexity of the programming task and to compare it with the skill and experience of assigned programmers in preparing a schedule. *Manage* can also be used after the computer is installed to plan the implementation of new or additional applications. With *Manage*, the user makes decisions based on current, sound information and can readily change plans, if necessary.

# 8 Tables

## Instruction Formats and Timing

Each Series 200 instruction is described in the following table in terms of its operation code, formats, and timing formulas. In addition, reference is made in each case to the page where the operations initiated by the instruction are described. The formulas given in the table provide execution times in memory cycles.

The internal operation of the Model 4200 processor differs from that of the other Series 200 processors in that data is moved in groups of four characters (a word) rather than singly. Consequently, the 4200 timing formulas differ considerably and are listed separately from those of the other processors.

Equivalent expressions for symbols used in the table are as follows:

SYMBOL	MEANING
A	Address of A-operand field.
B	Address of B-operand field.
V	Variant characters.
$N_i$	Number of characters in the instruction.
$N_a$	Number of characters in the field indicated by A.
$N_b$	Number of characters in the field indicated by B.
$N_w$	The number of characters in the A- or B-operand field, whichever is shorter.
$N_s$	Number of characters stored.
$N_j$	Number of character locations bypassed to reach the next sequential op code.
$N_r$	Number of characters referenced.
$N_{ic}$	Number of information units (6-bit or 12-bit characters) to be translated.
$NB_{it}$	Number of six-bit character locations occupied by each B-item information unit (1 or 2).
Z	Number of characters scanned during zero suppression.
\$	Number of characters scanned during dollar-sign insertion.
$W_i$	Number of four-character words used to store one more than the total number of characters in the instruction.
$N_{aw}$	Number of words in the field indicated by A.
$N_{bw}$	Number of words in the field indicated by B.
$N_{ww}$	Number of words in the A- or B-operand field, whichever is shorter.
N	Number of words used to store the data.
$N_{ws}$	Number of words stored.
$N_{wj}$	Number of words bypassed to reach the next sequential op code.
$N_{wr}$	Number of words referenced.
$N_{ct}$	Number of characters translated.
$N_{ia}$	Number of words in the item to be translated.
$N_{ib}$	Number of words in the result item.
$N_{ic}$	Number of information units (6 or 12-bit characters) to be translated.
$Z_w$	Number of words scanned during zero suppression.
$\$w$	Number of words scanned during dollar-sign insertion.
$X_0$	Zero if no second scan (zero suppression); 1 if the scan is performed.
$Y_0$	Zero if no third scan (dollar sign insertion); 1 if the scan is performed.
n	Number of bit positions shifted for automatic formatting.
$N_1$	Number of binary ones in a multiplier.
$N_{sh}$	Number of shifts.
G	Number of groups of two or more consecutive ones in the multiplier.
K	Number of single ones in the multiplier.
p	Total number of ones in the quotient.
r	Number of 01 groups in the quotient.

MONOP CODE	INSTRUCTION NAME	INSTRUCTION FORMAT(S)	TIMING FORMULAS (Memory Cycles)		DESCRIPTION PAGE
			120/200/1200/2200 <sup>(1)</sup>	4200	
<b>FIXED-POINT ARITHMETIC INSTRUCTIONS</b>					
A	Decimal Add	A/A,B A/A A	No Re complement <sup>(2)</sup> $N_i + 2 + N_w + 2N_b$ Re complement <sup>(2)</sup> $N_i + 2 + N_w + 4N_b$	No Re complement $W_i + .5N_{ww} + 2.5N_{bw} + 4.5$ Re complement $W_i + .5N_{ww} + 5N_{bw} + 4.5$	26
S	Decimal Subtract	S/A,B S/A S	No Re complement <sup>(2)</sup> $N_i + 2 + N_w + 2N_b$ Re complement <sup>(2)</sup> $N_i + 2 + N_w + 2N_b$	No Re complement $W_i + .5N_{ww} + 2.5N_{bw} + 4.5$ Re complement $W_i + .5N_{ww} + 5N_{bw} + 4.5$	26
M	Decimal Multiply	M/A,B M/A M	See page 27 for representative times.	See page 27 for representative times.	26
D	Decimal Divide	D/A,B D/A D	See page 27 for representative times.	See page 27 for representative times.	27
BA	Binary Add	BA/A,B BA/A BA	$N_i + 1 + N_w + 2N_b$	$W_i + .5N_{ww} + 2.5N_{bw} + 4.5$	27
BS	Binary Subtract	BS/A,B BS/A BS	$N_i + 1 + N_w + 2N_b$	$W_i + .5N_{ww} + 2.5N_{bw} + 4.5$	27
ZA	Zero and Add	ZA/A,B ZA/A ZA	$N_i + 1 + N_w + N_b$	$W_i + N_{ww} + N_{bw} + 4.5$	27
ZS	Zero and Subtract	ZS/A,B ZS/A ZS	$N_i + 1 + N_w + N_b$	$W_i + N_{ww} + N_{bw} + 4.5$	27
<b>SCIENTIFIC PROCESSING INSTRUCTIONS<sup>(3)</sup></b>					
AMA	Floating Add	AMA/A,XY	$N_i + 13 + [n/4]$	$W_i + 11.5 + n/6$	27
AAA	Floating Add	AAA/XY	$11 + [n/4]$	$W_i + 6.5 + n/6$	27
SMA	Floating Subtract	SMA/A,XY	$N_i + 13 + [n/4]$	$W_i + 11.5 + n/6$	28
SAA	Floating Subtract	SAA/XY	$11 + [n/4]$	$W_i + 6.5 + n/6$	28
MAM	Floating Multiply	MAM/A,XY	$N_i + 21 + [N_i/2] + [n/4]$	$W_i + 16.5 + 5G/6 + K/3 + n/6$	28
MAA	Floating Multiply	MAA/XY	$19 + [N_i/2] + [n/4]$	$W_i + 11.5 + 5G/6 + K/3 + n/6$	28
DMA	Floating Divide	DMA/A,XY	$N_i + 40 + [n/4]$	$W_i + 16.5 + p/3 + r/3 + n/6$	28
DAA	Floating Divide	DAA/XY	$38 + [n/4]$	$W_i + 11.5 + p/3 + r/3 + n/6$	28
TAM	Store Floating Accumulator	TAM/A,X-	$N_i + 10$	$W_i + 6.5$	28
TMA	Load Floating Accumulator	TAM/A,-Y	$N_i + 11$	$W_i + 7.5$	28
TAA	Load Floating Accumulator	TAA/XY	8	$W_i + 4$	28
FBA	Floating Test and Branch on Accumulator Condition	FBA/A,XC	$N_i + 4$ (No branch) $N_i + 6$ (Branch)	$W_i + 3.8$ (No branch) $W_i + 4.5$ (Branch)	28
FBI	Floating Test and Branch on Indicator	FBI/A,OD	$N_i + 2$ (No branch) $N_i + 4$ (Branch)	$W_i + 3$ (No branch) $W_i + 3.7$ (Branch)	28
DTB	Decimal to Binary	DTB/A,-Y	$N_i + 24$	$W_i + 12.5 + N$	28
BTD	Binary to Decimal	BTD/A,X-	$N_i + 23$	$W_i + 11.5 + N$	28
TLM	Store Low-Order Result	TLM/A	$N_i + 9^{(3)}$	$W_i + 7.5$	29
TLA	Store Low-Order Result	TLA/-Y	$6^{(3)}$	$W_i + 3.5$	29
TML	Load Low-Order Result	TML/A	$N_i + 9^{(3)}$	$W_i + 8.5$	29
TAL	Load Low-Order Result	TAL/X-	$6^{(3)}$	$W_i + 3.5$	29
BMS	Binary Mantissa Shift	BMS/XM,V	$9 + [N_{sh}/4]^{(3)}$	$W_i + 4.5 + N_{sh}/6$	29
BIM	Binary Integer Multiply	BIM/A,B	$N_i + 20 + [N_i/2]^{(3)}$	$W_i + 12.5 + 5G/6 + K/3$	29

Mnemonic Code	Instruction Name	Instruction Format(s)	Timing Formulas (Memory Cycles)		Description Page
			$120/200/1200/2000$	4200	
<b>LOGIC INSTRUCTIONS</b>					
EXT	Extract	EXT/A,B EXT/A EXT	$N_i + 1 + 3N_w$	$W_i + 3N_w + 4.5$	29
HA	Half Add	HA/A,B HA/A HA	$N_i + 1 + 3N_w$	$W_i + 3N_w + 4.5$	29
SST	Substitute	SST/A,B,V SST/A,B SST/A SST	$N_i + 4$	$W_i + 6.85$	29
C	Compare	C/A,B C/A C	$N_i + 2 + N_w + N_b^{(4)}$	$W_i + 1.5N_w + N_{bw} + 4.5$	30
B	Branch (Unconditional)	B/A	$N_i + 2$	$W_i + 4$	30
BCT	Branch on Condition Test	BCT/A,V BCT	$N_i + 2$	$W_i + 4$	30
BCC	Branch on Character Condition	BCC/A,B,V BCC/A,B BCC/A BCC	$N_i + 4$	$W_i + 5.5$	30
BCE	Branch if Character Equal	BCE/A,B,V BCE/A,B BCE/A BCE	$N_i + 4$	$W_i + 5.5$	30
BBE	Branch on Bit Equal	BBE/A,B,V BBE/A,B BBE/A BBE	$N_i + 4$	$W_i + 5.5$	30
<b>GENERAL CONTROL INSTRUCTIONS</b>					
SW	Set Word Mark	SW/A,B SW/A SW	$N_i + 3^{(5)}$	$W_i + 4.5$	31
SI	Set Item Mark	SI/A,B SI/A SI	$N_i + 3^{(5)}$	$W_i + 4.5$	31
CW	Clear Word Mark	CW/A,B CW/A CW	$N_i + 3$	$W_i + 4.5$	31
CI	Clear Item Mark	CI/A,B CI/A CI	$N_i + 3$	$W_i + 4.5$	31
H	Halt	H H/A H/A,B H/A,B,V	$N_i + 2^{(4)}$	$W_i + 4.5$	31
NOP	No Operation	NOP	$3^{(6)}$	$W_i + 3.5$	31
CAM	Change Addressing Mode	CAM/V CAM	$N_i + 2^{(6)}$	$W_i + 3.5$	31
CSM	Change Sequencing Mode	CSM CSM/A CSM/A,B CSM/A,B,V	$N_i + 3^{(6)}$	$W_i + 3.5$	31
SCR	Store Control Registers	SCR/A,V SCR/A SCR	$N_i + 5^{(4)}$	$W_i + N + 3.5$	32
LCR	Load Control Registers	LCR/A,V LCR/A LCR	$N_i + 5^{(4)}$	$W_i + N + 3.5$	32
LIB	Load Index/Barricade Register	LIB/A LIB	$N_i + 3$	$W_i + N + 4$	32
SIB	Store Index/Barricade Register	SIB/A SIB		$W_i + N + 3.5$	32

OPERATING OP CODE	INSTRUCTION NAME	FORMAT	FORMULA (Model 1200)	FORMULA (Model 2200)	DEFINITION PAGE
<b>INTERRUPT CONTROL INSTRUCTIONS</b>					
MC	Monitor Call	MC	$N_i + 2^{(6)}$	$W_i + 3.5$	32
SVI	Store Variant and Indicators	SVI/V	$N_i + 1 + N_s + N_j^{(7)}$	$W_i + N_{ws} + N_{wj} + 5.5$	32
RVI	Restore Variant and Indicators	RVI/A,V	$N_i + 2 + N_r^{(6)}$	$W_i + N_{wr} + 4.5$	33
RNM	Resume Normal Mode	RNM/A,B RNM/A RNM	$N_i + 3^{(8)}$	$W_i + 4$	33
<b>DATA MOVE INSTRUCTIONS</b>					
MCW	Move Characters to Word Mark	MCW/A,B MCW/A MCW	$N_i + 1 + 2N_w$	$W_i + 2N_{ww} + 4.5$	33
LCA	Load Characters to A-Field Word Mark	LCA/A,B LCA/A LCA	$N_i + 1 + 2N_a$	$W_i + 2N_{aw} + 4.5$	33
EXM	Extended Move	EXM/A,B,V EXM/A,B EXM/A EXM	$N_i + 1 + 2N_a$	$W_i + 2N_{aw} + 4.5$	33
MAT	Move and Translate	MAT/A,B,V <sub>1</sub> ,V <sub>2</sub>	$N_i + 3N_a^{(9)}$	$W_i + N_{ct} + 2N_{aw} + 5.1$	33
MIT	Move Item and Translate	MIT/A,B,V <sub>1</sub> ,V <sub>2</sub> ,V <sub>3</sub>	$N_i + N_a + 2N_{ic}(NB_u)$	$W_i + N_{ia} + N_{ib} + N_{ic} + 6$	34
<b>FOR INSTRUCTION</b>					
MCE	Move Characters and Edit	MCE/A,B MCE/A MCE	$N_i + 1 + N_a + 2N_b + 2Z + 2\$$	$W_i + N_{aw} + 2.3N_{bw} + 2Z_w + 2\$_w + 5.5 + X_o + Y_o$	34
<b>INPUT/OUTPUT INSTRUCTIONS</b>					
PDT	Peripheral Data Transfer	PDT/A,C <sub>1</sub> ,C <sub>2</sub> ,...C <sub>n</sub>	See description of PDT instruction.	See description of PDT instruction.	34
PCB	Peripheral Control and Branch	PCB/A,C <sub>1</sub> ,C <sub>2</sub> ,...C <sub>n</sub>	See description of PCB instruction.	See description of PCB instruction.	35

- (1) Add one memory cycle to these formulas when calculating Model 2200 times, except where the formula is followed by footnote (4), (7), (8), or (9).
- (2) Subtract one memory cycle from this formula if the instruction is being executed in the Model 120 or 1200 processor.
- (3) These formulas apply only to the Models 1200 and 2200 processors; the scientific unit is not available with the Models 120 and 200.
- (4) Add two memory cycles to this formula if the instruction is being executed in the Model 2200 processor.
- (5) Subtract one memory cycle from this formula if the instruction is being executed in the Model 1200 processor in the format Op Code/A,B.
- (6) Subtract one memory cycle from this formula if the instruction is being executed in the Model 1200 processor.
- (7) Add one memory cycle to this formula if the instruction is being executed in the Model 1200 processor; add two cycles to the formula if the instruction is executed in the Model 2200 processor.
- (8) Add two memory cycles to this formula if the instruction is being executed in the Model 2200 processor; subtract one cycle from the formula if the instruction is executed in the Model 1200 processor.
- (9) Add four memory cycles to this formula if the instruction is being executed in the Model 2200 processor.

# Correspondence Among Series 200 Central Processor, Card, and Printer Codes

Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer	Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer
0	0	000000	00	0		X, 0 or X <sup>(1)</sup>	100000	40	—
1	1	000001	01	1	J	X, 1	100001	41	J
2	2	000010	02	2	K	X, 2	100010	42	K
3	3	000011	03	3	L	X, 3	100011	43	L
4	4	000100	04	4	M	X, 4	100100	44	M
5	5	000101	05	5	N	X, 5	100101	45	N
6	6	000110	06	6	O	X, 6	100110	46	O
7	7	000111	07	7	P	X, 7	100111	47	P
8	8	001000	10	8	Q	X, 8	101000	50	Q
9	9	001001	11	9	R	X, 9	101001	51	R
#	8, 2	001010	12	'		X, 8, 2	101010	52	#
	8, 3	001011	13	=	\$	X, 8, 3	101011	53	\$
@ Space	8, 4	001100	14	:	*	X, 8, 4	101100	54	*
	Blank	001101	15	Blank		X, 8, 5	101101	55	"
	8, 6	001110	16	>		X, 8, 6	101110	56	≠
	8, 7	001111	17	&	—	X or X, 0 <sup>(1)</sup>	101111	57	½ or I <sup>(2)</sup>
A	R, 0 or R <sup>(1)</sup>	010000	20	+		8, 5	110000	60	<
	R, 1	010001	21	A	/	0, 1	110001	61	/
B	R, 2	010010	22	B	S	0, 2	110010	62	S
C	R, 3	010011	23	C	T	0, 3	110011	63	T
D	R, 4	010100	24	D	U	0, 4	110100	64	U
E	R, 5	010101	25	E	V	0, 5	110101	65	V
F	R, 6	010110	26	F	W	0, 6	110110	66	W
G	R, 7	010111	27	G	X	0, 7	110111	67	X
H	R, 8	011000	30	H	Y	0, 8	111000	70	Y
I	R, 9	011001	31	I	Z	0, 9	111001	71	Z
	R, 8, 2	011010	32	:		0, 8, 2	111010	72	@
•	R, 8, 3	011011	33	.	,	0, 8, 3	111011	73	,
	R, 8, 4	011100	34	)	%	0, 8, 4	111100	74	(
□	R, 8, 5	011101	35	%		0, 8, 5	111101	75	C <sub>R</sub>
	R, 8, 6	011110	36	□		0, 8, 6	111110	76	□
&	R, or R, 0 <sup>(1)</sup>	011111	37	?		0, 8, 7	111111	77	¢

(1) Special code, designated by a card read or punch PCB instruction, which provides compatibility with Honeywell 400 and 800 systems.

(2) The exclamation point replaces the one-half symbol on a type roll containing the Mark II character font.

HONEYWELL EDP TECHNICAL PUBLICATIONS  
USERS' REMARKS FORM

TITLE: SERIES 200  
SUMMARY DESCRIPTION

DATED: FEBRUARY 1966  
FILE NO: 143.0005.0000.0-012

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME \_\_\_\_\_ DATE \_\_\_\_\_  
COMPANY \_\_\_\_\_  
TITLE \_\_\_\_\_  
ADDRESS \_\_\_\_\_  
\_\_\_\_\_

Cut Along Line

**BUSINESS REPLY MAIL**

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

**HONEYWELL**

ELECTRONIC DATA PROCESSING DIVISION

**60 WALNUT STREET**

**WELLESLEY HILLS, MASS. 02181**

ATT'N: PUBLICATIONS AND WRITING SERVICES DEPARTMENT

FIRST CLASS  
PERMIT NO. 39531  
WELLESLEY HILLS  
MASS.



City Line

**Honeywell**  
ELECTRONIC DATA PROCESSING

22



HONEYWELL  
ELECTRONIC  
DATA  
PROCESSING

WELLESLEY HILLS,  
MASSACHUSETTS 02181