SPECIAL SYSTEMS

CONTROL DATA®

GRID ASSEMBLY SYSTEM

PRELIMINARY

REFERENCE MANUAL

CONTROL DATA CORPORATION

SPECIAL SYSTEMS

CONTROL DATA®

GRID ASSEMBLY SYSTEM

REFERENCE MANUAL

CONTROL DATA CORPORATION

RECORD OF REVISIONS	
Revision	**Notes**
01	Preliminary 7-25-69

GRASS Reference Manual
Publication Number 44624500

Address comments concerning this manual to:

Control Data Corporation
Special Systems Division
Technical Publications
4201 North Lexington Avenue
St. Paul, Minnesota 55112

PREFACE

The GRID Assembly System (GRASS) is a Control Data® 6000 Series central processor program which allows a programmer to write programs in symbolic notation for the GRID Processor. The assembly system accepts instructions in mnemonic form, which are easier to remember and to interpret than actual machine code equivalents.

GRASS assembles symbolic programs from source statements input on Hollerith cards. GRASS provides a printer listing and binary output for each program successfully assembled. The binary output, in GRID octal machine code equivalents, can be input to GRID memory for execution.

This manual describes the GRASS assembler operation (Section 2), the symbolic language (Section 3), and the hardware environment (Section 1).

This manual is intended to be used with the following Control Data publications:

Title	Publication Number
Control Data 6400/6500/6600 Computer Systems Reference Manual	60100000
Control Data 6400/6500/6600 Computer Systems SCOPE 3.1.2 Reference Manual	60189400
Control Data GRID Display Subsystem Hardware Reference Manual	82134500

CONTENTS

FIGURES

TABLES

HARDWARE ENVIRONMENT 1

The GRID Display is a computer-controlled, stored program, visual data processing system made up of two major functional units: the Display Controller and the Display Console. It can be operated on-line by receiving instructions and data from a large-scale data processing system (e.g., CDC 6000 Series), or off-line by using internal memory for program and data storage. Manual input devices — alphanumeric/function keyboard and light pen – allow an operator to interrupt the operational mode for data entry. The GRID Display will operate either in the Processor mode or in Display mode. Due to the use of internal modules in both modes, it will not operate in both modes simultaneously.

BANK CONTROL

The Display Controller may have from one to three memory banks of 4096 12-bit words per bank. Whenever there is more than one bank, bank control becomes necessary. Each bank has $10,000_8$ addresses numbered from 0000_8 to 7777_8. Bank control is necessary to allow selection of banks for relative, direct, and indirect addressing.

For a complete description of the Display Controller or Console, refer to the GRID Display Subsystem Hardware Reference Manual (Control Data publication number 82134500).

ASSEMBLER OPERATION 2

This section describes GRASS operation in terms of control card and I/O formats. GRASS input is from source cards. Output is a printer listing and a binary file.

CONTROL CARDS

A GRASS assembly program is submitted as a job under the SCOPE operating system. It is called when the GRASS control card appears in the control card record of a job.

Control card formats:

```
GRASS.
GRASS (lfn1)
GRASS (,lfn2)
GRASS (lfn1,lfn2)
```

lfn1 the file name of the file containing GRASS input.

lfn2 the file name on which binary output will be produced.

If file names are not specified, the assembler automatically uses the SCOPE system files – INPUT and PUNCHB – for lfn1 and lfn2, respectively.

The program resulting from a GRASS assembly is non-relocatable. GRASS can assemble more than one source program per call. Symbolic program statements are in a fixed coded format. Figure 2-1 shows the input deck structure.

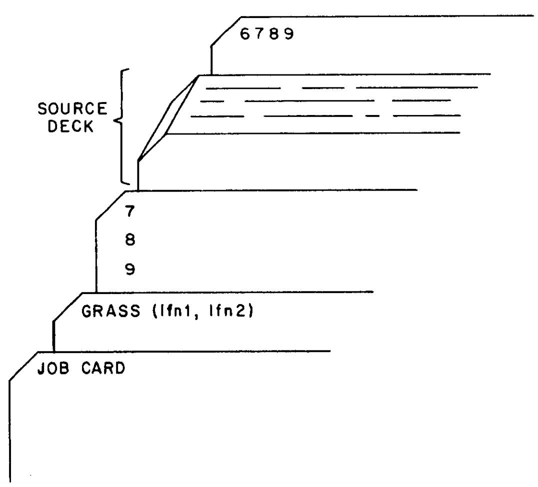
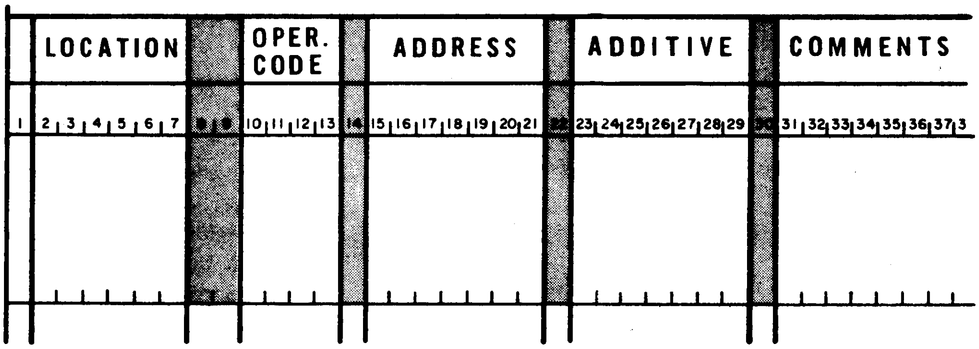


Figure 2-1. GRASS Assembler Input Deck

ASSEMBLER INPUT FORMAT

SOURCE STATEMENT

The source statement is made up of five fields: location, operation code, address, additive, and comments. The operation code (op-code) is mandatory for all source statements, while the other fields are mandatory or optional depending on the op-code. The comments field is always optional. Fields are defined by their fixed positions on the source card as described in this section.



LOCATION FIELD

The symbolic label is from one to six alphanumeric characters in length. Blanks or spaces are not considered characters and should not be used. The first character must be alphabetic. Symbolic labels are stored in the symbol table which has a capacity of 1000_{10} entries.

OPERATION CODE FIELD

This field can contain any of the GRASS symbolic op-codes or pseudo-ops described in the next section.

ADDRESS FIELD

This seven-character field may contain an octal constant, a decimal constant followed by a D, a location symbol, or an asterisk (*), which represents the value of the location counter. The location counter indicates the address of the current instruction. A positive or negative constant or label is denoted by a plus (+) or minus (-) sign in column 15 of the address field. A blank in column 15 denotes a positive quantity. Any location symbol used in an address field must appear in the location field somewhere in the program.

ADDITIVE FIELD

Information specified by the additive field will be added algebraically to information specified in the address field. In backward and forward mode instructions, the sum will be adjusted by the location counter if that sum is relative. Instruction modes are identified and explained in the next section.

The additive field in normal mode may contain an octal constant, a decimal constant followed by a D, a location symbol, and an asterisk (*),which represents the value of the location counter. A positive or negative constant or label is denoted by a plus (+) or minus (-) sign in column 23 of the additive field. A blank in column 23 denotes a positive quantity. Any location symbol used in an additive field must appear in the location field somewhere in the program.

For instructions in Display mode which have a ZZ field, the contents of columns 23 and 24 must be blank and columns 25-30 will be for the special flags: S=SCA, B=BLI, O=OR, D=DSH, L=SIZ, P=LP, and K=AN, as outlined below.

Flag	Bit Setting	Effect
SCALE (SCA) - Bit 5		
	0	allows positioning; point unblank
	1	allows positioning and inhibits point unblank
BLINK (BLI) - Bit 4		
	0	inhibits blinking
	1	selects blinking
ORIENTATION (OR) - Bit 3		
	0	plots symbols in normal orientation
	1	plots symbols rotated 90° CCW
DASH (DSH) - Bit 3 - VECTOR MODE ONLY		
	0	line vectors
	1	dashed line vectors
LIGHT PEN (LP) - Bit 2 (EOD and DOD operations only)		
ALPHANUMERIC KEYBOARD (AN) - Bit 0		
	0	status of device unchanged
	1	identifies device as the device to be affected by the EOD or DOD operation

SIZE (SIZ) - Bit 1

Bit 1 determines automatic spacing between symbols.

Bit Setting	Spacing	Symbols/Line
0	14_8	86
1	20_8	64

COMMENTS FIELD

This field may contain 0 to 50_{10} Hollerith characters.

REMARKS CARD

Remarks can be inserted in the assembly deck to be printed on the assembly listing by using a remarks card. The remarks card is identified by an asterisk punched in column 1 with the desired remarks punched in columns 2 through 80. A remarks card does not generate any object code in the assembly of a program.

PRINTER LISTING

GRASS generates, as output, an assembly listing on the line printer and a punched binary card file. The assembly listing format is:

Columns	Contents
3-6	error codes field
8	bank number
10-13	octal address
19-22	octal code for first word GRID instruction
25-29	this field may contain one of the three below: 1. octal code for second word GRID instruction 2. octal constant 3. counter setting as a result of PRG, or ORG
33-144	source card image

Any columns not specified are to be assumed not used.

ERROR MESSAGES

Error codes listed below may appear on an assembly listing to explain error conditions detected in the source input.

Error code	Explanation
R	operand or operand address out of range
D	symbol defined more than once
O	illegal op code
A	illegal address or additive field
U	undefined symbol in address or additive field

Any one of the above errors will cause suppression of GRASS binary output for the program in which the error occurred.

A portion of an assembly printer listing would appear like this:

```
                                        GRID  ASSEMBLY LISTING                          PAGE    2

  0 0055     1314                       LPB   BETTY
R 0 0056     1317               EROR 3  LPB   SAM              ERROR
  0 0057     0307               SAM     SCN   7                SELECTIVE  COMPLEMENT
  0 0060     1435                       SCD   35
  0 0061     1457                       SCD   SAM
  0 0062     1500     0057              SCM   SAM
```

BINARY OUTPUT

A binary output file is automatically generated as a result of an assembly that completes without any errors. Any of the errors previously listed causes suppression of the binary output. The GRASS binary card format is:

Column(s)	Rows	Contents
1	12-3	word count in 6000 central memory words
1	7, 9	7-9 punch
2	12-9	checksum
3	8-9	bank number (0, 1, or 2)
3	0-6	number of GRID memory words on this card
4	12-9	GRID address for first memory word in column 8
5-7		(Not used)
8-77	12-9	GRID 12-bit memory words, one word per column
78-79		(Not used)
80	12-9	card sequence

If the output is not on cards, GRASS generates one 6000 Series binary file, made up of one or more 15 central memory word blocks.

Word 1	byte 0	bank number, number of GRID words in record
	byte 1	GRID address for first word
	bytes 2-4	not used
Word 2	byte 0	1st GRID word (12 bits)
	byte 1	2nd GRID word
	byte 2	3rd GRID word
	byte 3	4th GRID word
	byte 4	5th GRID word
•	•	•
•	•	•
•	•	•
Word 15	byte 0	66th GRID word
	byte 1	67th GRID word
	byte 2	68th GRID word
	byte 3	69th GRID word
	byte 4	70th GRID word
Word 16	byte 0	bank number, number of GRID words in record
	byte 1	GRID address for first word
	bytes 2-4	not used

SYMBOLIC INSTRUCTIONS 3

This section introduces:

- Word format
- Address modes
- Processor mode instructions
- Display mode instructions
- Pseudo instructions

WORD FORMATS

A one-word instruction has a 6-bit function code (F) and a 6-bit execution address (E). Most instructions require only one word of storage but expanded instructions occupy two words. Figure 3-1 shows the processor instruction word formats. The second word contains a 12-bit address or operand (G), depending on the instruction. Both words 1 and 2 must be located in sequential storage addresses in the same memory bank.

	11 — 6	5 — 0
Word 1	Function Code (F)	Execution Address (E)

	11 — 0
Word 2	Address Or Operand (G)

Figure 3-1. Processor Instruction Word Formats

ADDRESS MODES

The seven memory address modes listed in Table 3-1 provide a maximum 6-bit address flexibility.

TABLE 3-1. MEMORY ADDRESS MODES

Address Mode	Description
No Address (N)	When the E portion of a processor instruction is a 6-bit operand, the instruction initiates arithmetic and logical operations using this 6-bit operand as a constant. By combining constant and instruction, this mode conserves memory locations.
Direct Address (D)	Refers to a 12-bit operand in one of the first 100_8 memory locations in the direct memory bank.
Indirect Address (I)	Provides for operand references and jump address. For processor instructions employing indirect addressing, E refers to one of 77_8 of the first 100_8 memory locations starting at location 0001 of the direct storage bank. The contents of this address becomes the address of the operand or the jump address.
Relative Address Forward (F)	Generates operand or jump address by adding E to the current contents of P. This specifies one of 77_8 addresses immediately following the address of current processor instruction.
Relative Address Backward (B)	Generates operand or jump address by subtracting E from the current contents of P. This specifies one of 77_8 addresses immediately preceding the address of the current processor instruction.
Constant Address (C)*	The G portion of a 24-bit processor instruction contains the operand.
Memory Address (M)*	The G portion of a 24-bit processor instruction contains the address of the operand.

*This mode uses two sequential storage locations. The designation for the second location is G. In this mode, E must always equal zero.

The following paragraphs contain examples of the address modes listed on the previous page.

NO-ADDRESS MODE (N)

In no-address mode, E is the lower 6 bits of an implied 12-bit operand. The upper 6 bits of the operand always equal zero. Thus, the E portion of the instruction word becomes the operand.

Example:

Location	F	E
(r)0100	LDN	43 (load no address)
(r)0101	next instruction	

Instruction generated:

Bank	Address	Octal Code(1)	Code (2)
0	0100	0443	—

A load instruction transmits the operand to the A register. The 12-bit operand for LDN 43 is 0043.* The number 0043 goes to the A register. At the completion of a no-address (N) instruction, control continues at the location in the relative storage bank (r) specified by the contents of P+1. In this case, control continues at location (r)0101.

DIRECT ADDRESS MODE (D)

In the direct address mode, E selects one of the first 100_8 locations in the direct storage bank (d) as the operand address.

Example:

Location	F	E
(d)0076	12	34
(r)1075	LDD	76 (load direct)
(r)1076	next instruction	

Instruction generated:

Bank	Address	Octal Code(1)	Code(2)
0	1075	2076	—

E specifies the operand address as (d)0076. This address contains the quantity 1234 which will be transferred to the A register. At the completion of a direct address (D) instruction, control continues at the location in the relative storage bank specified by the contents of P+1. In this case, control continues at (r)1076.

*All numbers in example programs are octal unless stated otherwise.

INDIRECT ADDRESS MODE (I)

In Indirect Address mode, E references 77_8 of the first 100_8 locations, starting at location 0001 of the direct storage bank (d). The location (d)00E is then referenced and the contents of (d)00E become the operand address in the indirect bank (i).

Example:

Location	F	E
(d)0045	33	65
(1)3365	46	57
(r)4121	LDI	45 (load indirect)
(r)4122	next instruction	

Instruction generated:	Bank	Address	Octal Code(1)	Code(2)
	0	4121	2145	–

E calls for referencing (d)0045, which contains the address 3365. A final reference is now made to (i)3365, which contains the number 4657. The quantity 4657 goes to the X register. Notice that both the direct (d) and indirect (i) storage bank controls are involved in the indirect address mode. At completion of an (i) instruction, control continues at the location in the relative storage bank specified by the contents of P+1. In the above example, control continues at (r)4122.

There are two indirect instructions which are exceptions to the above rules, JPI and JFI:

- JPI initially references location (d)00E. A transfer of control then takes place within the relative (r) bank to the location specified by the contents of (d)00E.
- JFI initial reference is relative forward. A transfer of control then takes place within the relative (r) bank to the address specified in the relative forward reference.

RELATIVE FORWARD ADDRESS MODE (F)

In relative forward address mode, E adds to the contents of the P register. This sum becomes the effective operand address in the relative storage bank (r).

Example:

Location	F	E
(r)0233	LDF	22 (load forward)
(r)0234	next instruction	
(r)0255	77	03

Instruction generated:	Bank	Address	Octal Code(1)	Code(2)
	0	0233	2222	–

Adding E to the P register yields address (r)0255. The contents of (r)0255 go to the X register. At completion of this instruction, X contains 7703. At completion of an (F) instruction which does not cause transfer of control, control continues in relative storage bank (r) at the location specified by contents of P+1. In the above example, control continues at location (r)0234. Forward Jump instructions transfer control E locations forward in the relative bank.

RELATIVE BACKWARD ADDRESS MODE (B)

The relative backward address mode functions similar to relative forward (F) mode. In relative backward address mode, subtraction of (E) from the contents of the P register forms an address in the relative storage bank.

CONSTANT ADDRESS MODE (C)

All constant address mode instructions occupy two sequential storage locations. The G portion of the 24-bit instruction word contains the operand. E always equals zero.

Example:	Location	F	E	G
	(r)0101	LDC	00	7337 (load constant)
	(r)0103	STC	00	2345 (store constant)
	(r)0105	next instruction	00	

Instruction generated:	Bank	Address	Octal Code(1)	Code(2)
	0	0101	2200	7337

A load constant (LDC) instruction is at location (r)0101. The operand address is (r)0102. The quantity 7337 goes to the X register. Upon completion of a (C) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P+2. In this case, control continues at (r)0103.

This address contains a store constant (STC) instruction. This transfers X register contents to the operand address. In the above example, the operand address of the STC instruction is (r)0104. The quantity 7337, in the X register as a result of the LDC instruction in (r)0101, goes to location (r)0104, replacing the constant 2345 now in (r)0104. Final contents of (r)0104 are 7337. Control continues at (r)0105.

MEMORY ADDRESS MODE (M)

All memory address mode instructions occupy two sequential storage locations. The G portion of the 24-bit instruction word contains the address of the operand. E always equals zero.

Example:

Location	F	E	G
(r)3477	LDM	00	1111 (load memory)
(r)3501	STM		0024 (store memory)
(r)3503	next instruction		
(i)1111	67	66	
(i)0024	02	34	

Instruction generated:	Bank	Address	Octal Code(1)	Code(2)
	0	3477	2100	1111

Location (r)3477 contains a load memory (LDM) instruction. The location (i)1111 becomes the operand address. The quantity 6766 goes to the X register. Upon completion of an (M) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P+2. In this case, control continues at location (r)3501 which contains a store memory (STM) instruction. The operand address of this instruction becomes (i)0024. This stores X register quantity 6766 in location (i)0024, replacing quantity 0234. Control continues at location (r)3503.

PROCESSOR MODE INSTRUCTIONS

In the processor mode of operation (activated by manual interrupt, interface function code, operator panel, or display jump instructions), the program stored in memory controls processor operation.

Valid processor mode mnemonic codes are processed by GRASS to form the corresponding processor instruction words.

The following table lists the operation, the mnemonic, octal code, address mode, and instruction time.

TABLE 3-2. PROCESSOR MODE MNEMONIC CODES

Operation	Mnemonic	Code	Mode	Time (microseconds)
ERROR STOP	ERR	0000	n	2.4
NO OPERATION	NOPX	000X	n	2.4
MEMORY BANK CONTROLS (b = Bank Number)	SRJb	001b	n	2.4
	SICb	002b	n	2.4
	IRJb	003b	n	2.4
	SDCb	004b	n	2.4

TABLE 3-2. (Cont'd)

Operation	Mnemonic	Code	Mode	Time (microseconds)
MEMORY BANK CONTROLS (b = Bank Number)	DRJb	005b	n	2.4
	SIDb	006b	n	2.4
	ACJb	007b	n	2.4
MEMORY BANK CONTROL TO A	CTA	0130	n	2.4
TRANSFER INTERNAL REGISTERS	STA	0100	n	2.4
	PTA	0101	n	2.4
	YTA	0170	n	2.4
	R1TA	0171	n	2.4
	R2TA	0172	n	2.4
	RSTA	0173	n	2.4
	ATY	0174	n	2.4
	ATA1	0175	n	2.4
	ATA2	0176	n	2.4
	ATRC	0177	n	2.4
LEFT SHIFT ONE	LS1	0102	n	2.4
LEFT SHIFT THREE	LS3	0110	n	2.4
LEFT SHIFT SIX	LS6	0111	n	2.4
MULTIPLY BY 10	MUT	0112	n	2.4
CLEAR INTERRUPT LOCKOUT	CIL	0120	n	2.4
SET INTERRUPT LOCKOUT	SIL	0121	n	2.4
INTERRUPT DATA SOURCE	IDS	0122	n	2.4
LOGICAL PRODUCT	LPN	02XX	n	2.4
	LPD	10YY	d	4.8
	LPM	1100	m	6.8
	LPI	11YY	i	6.8
	LPC	1200	c	4.8
	LPF	12XX	f	5.6
	LPB	13XX	b	5.6
SELECTIVE COMPLEMENT	SCN	03XX	n	2.4
	SCD	14YY	d	4.8
	SCM	1500	m	6.8
	SCI	15YY	i	6.8

TABLE 3-2. (Cont'd)

Operation	Mnemonic	Code	Mode	Time (microseconds)
SELECTIVE COMPLEMENT	SCC	1600	c	4.8
	SCF	16XX	f	5.6
	SCB	17XX	b	5.6
LOAD	LDN	04XX	n	2.4
	LDD	20YY	d	4.8
	LDM	2100	m	6.8
	LDI	21YY	i	6.8
	LDC	2200	c	4.8
	LDF	22XX	f	5.6
	LDB	23XX	b	5.6
LOAD COMPLEMENT	LCN	05XX	n	2.4
	LCD	24YY	d	4.8
	LCM	2500	m	6.8
	LCI	25YY	i	6.8
	LCC	2600	c	4.8
	LCF	26XX	f	5.6
	LCB	27XX	b	5.6
ADD	ADN	06XX	n	2.4
	ADD	30YY	d	4.8
	ADM	3100	m	6.8
	ADI	31YY	i	6.8
	ADC	3200	c	4.8
	ADF	32XX	f	5.6
	ADB	33XX	b	5.6
SUBTRACT	SBN	07XX	n	2.4
	SBD	34YY	d	4.8
	SBM	3500	m	6.8
	SBI	35YY	i	6.8
	SBC	3600	c	4.8
	SBF	36XX	f	5.6
	SBB	37XX	b	5.6
STORE	STD	40YY	d	4.8
	STM	4100	m	6.8
	STI	41YY	i	6.8

TABLE 3-2. (Cont'd)

Operation	Mnemonic	Code	Mode	Time (microseconds)
STORE	STC	4200	c	4.8
	STF	42XX	f	5.6
	STB	43XX	b	5.6
LEFT SHIFT 1 AND REPLACE	SRD	44YY	d	5.6
	SRM	4500	m	9.2
	SRI	45YY	i	9.2
	SRC	4600	c	7.2
	SRF	46XX	f	8.0
	SRB	47XX	b	8.0
REPLACE AND ADD	RAD	50YY	d	7.2
	RAM	5100	m	9.2
	RAI	51YY	i	9.2
	RAC	5200	c	7.2
	RAF	52XX	f	8.0
	RAB	53XX	b	8.0
REPLACE AND ADD 1	AOD	54YY	d	7.2
	AOM	5500	m	9.2
	AOI	55YY	i	9.2
	AOC	5600	c	7.2
	AOF	56XX	f	8.0
	AOB	57XX	b	8.0
ZERO JUMP	ZJF	60XX	f	3.2
	ZJB	64XX	b	3.2
NONZERO JUMP	NZF	61XX	f	3.2
	NZB	65XX	b	3.2
POSITIVE JUMP	PJF	62XX	f	3.2
	PJB	66XX	b	3.2
NEGATIVE JUMP	NJF	63XX	f	3.2
	NJB	67XX	b	3.2
JUMP INDIRECT	JPI	70YY	d	4.4
RETURN JUMP	JPR	7100	m	6.8
JUMP FORWARD INDIRECT	JFI	71XX	fi	5.2
JUMP TO DISPLAY	JPDb	740b	m	4.4
JUMP RESUME DISPLAY	JRDb	744b	m	4.4
INPUT/OUTPUT	INP	72XX	fi	
	OUT	73XX	fi	
	EXC	7500	m	

TABLE 3-2. (Cont'd)

Operation	Mnemonic	Code	Mode	Time (microseconds)
INPUT/OUTPUT	EXF	75XX	f	
	INA	7600	n	
	OTA	7677	n	
HALT	HLT	77XX	n	2.4

DISPLAY MODE INSTRUCTIONS

The Display mode of operation (activated by interface function code, processor jump-display instruction, or operator panel) enables displaying symbols, points, and vectors on the CRT.

While in Display mode, mnemonic codes are processed by GRASS to form the corresponding display instruction words.

The following table lists the operation, the mnemonic, octal code, Address mode, and the lower six bit settings.

TABLE 3-3. DISPLAY MODE MNEMONIC CODES

Operation	Mne-monic	Octal Code	Mode	Lower Six Bits (ZZ) Settings					
				5	4	3	2	1	0
PLOT POINT MODE I	PM1	40ZZ	n		BLI				
PLOT POINT MODE II	PM2	41ZZ	n	SCA	BLI				
PLOT SYMBOL MODE I	SM1	42ZZ	n		BLI	OR		SIZ	
PLOT SYMBOL MODE II	SM2	43ZZ	n	SCA	BLI	OR		SIZ	
TABULAR SYMBOL MODE I	TM1	44ZZ	n		BLI	OR		SIZ	
TABULAR SYMBOL MODE II	TM2	45ZZ	n	SCA	BLI	OR		SIZ	
VECTOR MODE I	VM1	46ZZ	n		BLI	DSH			
VECTOR MODE II	VM2	47ZZ	n	SCA	BLI	DSH			
JUMP TO PROCESSOR	JMPb	520(4+b)	m						
JUMP TO DISPLAY	JMDb	520b	m						
RETURN JUMP	RTJ	5300	m						
IDENTIFIER	IDW	54XX	n						
START REFRESH CYCLE	RFS	5500	n						
ENABLE OPERATOR DEVICE	EOD	56ZZ	n				LP		AN
DISABLE OPERATOR DEVICE	DOD	57ZZ	n				LP		AN

PSEUDO INSTRUCTIONS

Pseudo instructions permit the programmer to control the assembly of a symbolic program. They are used to alter the setting of the location counter, to define data and symbolic location, specify banks for assembled instructions, define end-of-program, or control printer output.

LOCATION COUNTER CONTROL

ORG

Statement format:

	LOCATION		OPER. CODE		ADDRESS		ADDITIVE		COMMENTS
			ORG		address		addend		
1	2 3 4 5 6 7	8 9	10 11 12 13	14	15 16 17 18 19 20 21	22	23 24 25 26 27 28 29	30	31 32 33 34 35 36 37 3

Sets the location counter to the value of the algebraic sum of the address and additive (AA) fields. This value is the location to which the next instruction (after an ORG) is assembled. Symbols appearing in either AA field must be defined before ORG is encountered. An ORG can not be used to continue an assembly. Blank AA fields set the location counter to 0.

PRG

Statement format:

	LOCATION		OPER. CODE		ADDRESS		ADDITIVE		COMMENTS
			PRG		address		addend		
1	2 3 4 5 6 7	8 9	10 11 12 13	14	15 16 17 18 19 20 21	22	23 24 25 26 27 28 29	30	31 32 33 34 35 36 37 3

Has all the properties of ORG but is used principally to continue an assembly at a new location.

BSS

Statement format:

	LOCATION		OPER. CODE		ADDRESS		ADDITIVE		COMMENTS
			BSS		address		addend		
1	2 3 4 5 6 7	8 9	10 11 12 13	14	15 16 17 18 19 20 21	22	23 24 25 26 27 28 29	30	31 32 33 34 35 36 37 3

Advances the location counter by the amount specified in the address plus additive field. Any symbol in the location field will be assigned to the first numeric address in the block.

DATA DEFINITION

BSSZ

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
	BSSZ	address	addend	

Performs the same as BSS except that the block will be filled with zeros.

DATA

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
	DATA	address	addend	

Causes the sum of the address and additive fields to be assembled in the current location, and advances the location counter by one.

SYMBOL DEFINITION

EQU

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
symbol	EQU	alpha	omega	

Assigns the Algebraic sum of the address and additive fields of the EQU pseudo-op to the symbol in the location field, and places this symbol and the numeric value representing the location in the symbol table.

ASSEMBLY CONTROL

BNKb

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
	BNKb			
1 2 3 4 5 6 7 8 9	10 11 12 13 14	15 16 17 18 19 20 21 22	23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 3

Causes the instructions following the pseudo-op to be assembled for bank b (b may be 0, 1 or 2).

NOTE

The BNKb pseudo-op does not affect the location counter. The location counter should be reset immediately before or after a BNKb card.

END

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
	END			
1 2 3 4 5 6 7 8 9	10 11 12 13 14	15 16 17 18 19 20 21 22	23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 3

Must be used to mark the end of a program, and therefore must be present to terminate a logical program.

PRINT CONTROL

SPACEx

Statement format:

LOCATION	OPER. CODE	ADDRESS	ADDITIVE	COMMENTS
	SPACE	x		
1 2 3 4 5 6 7 8 9	10 11 12 13 14	15 16 17 18 19 20 21 22	23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 3

This instruction controls line spacing on the printer. The number of lines to be skipped is specified by the "x" entry. Legal values of x are $1 \leq x \leq 9$.

EJECT

Statement format:

	LOCATION		OPER. CODE		ADDRESS		ADDITIVE		COMMENTS
			EJEC	T					
1	2 3 4 5 6 7	8 9	10 11 12 13	14	15 16 17 18 19 20 21	22	23 24 25 26 27 28 29	30	31 32 33 34 35 36 37 3

This instruction causes the next line of assembler output to be printed at the top of the next page on the printer.

CONTROL DATA CORPORATION
SUNNYVALE OPERATIONS
215 Moffet Park Drive, Sunnyvale, California 94086