

**Honeywell**

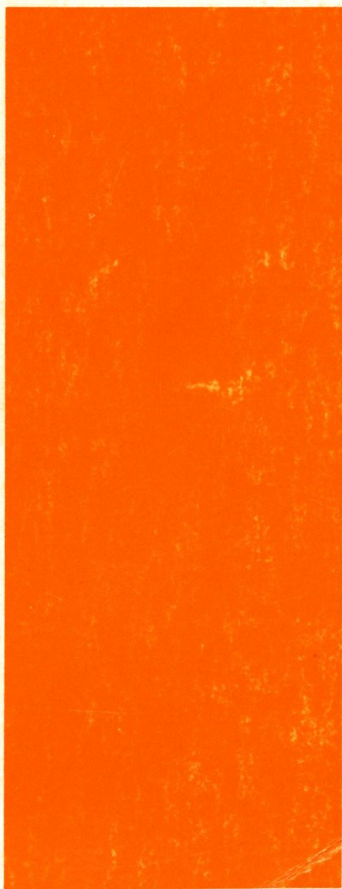
**GMAP  
POCKET GUIDE**

**SERIES 600/6000**

---

**SOFTWARE**

---



# Honeywell

## GMAP POCKET GUIDE

### SERIES 600/6000

#### SUBJECT:

Series 600/6000 GMAP Assembler, I/O System, and Miscellaneous Reference Information.

#### SPECIAL INSTRUCTIONS:

This pocket guide, Order No. BS16, Rev. 1, supersedes the previous edition, BS16, Rev. 0, dated September 1971.

#### SOFTWARE SUPPORTED:

Series 600 Software Release 8.0

Series 6000 Software Release F

#### DATE:

July 1974

#### ORDER NUMBER:

BS16, Rev. 1

User's Name \_\_\_\_\_

## PREFACE

This guide is based on material extracted from the following manuals:

1. Series 600/6000 Macro Assembler Program, BN86
2. Series 600/6000 General Comprehensive Operating Supervisor, BR43
3. Series 600/6000 File and Record Control, BN85
4. Series 600/6000 General Loader, BN90

## CONTENTS

<b>Series 600/6000 Information</b>	
Instruction Format	1
Data Word Formats	2
Instruction Op Codes	4
Instruction Mnemonics	5
EIS Op Codes	7
EIS Mnemonics	8
Extended Memory Op Codes and Mnemonics	9
Indicator Register	9
Effect of Compares on Indicators	10
Instruction Counter Fault Codes	10
<b>GMAP Assembler Information</b>	
Pseudo-Ops	11
Call, Save, Return Pseudo-Ops	13
MME's	14
Literal Formats	15
GMAP Error Flags	15
Relocatable Object Decks	16
<b>I/O System</b>	
I/O Select Sequence	17
File Designator Words	21
File Control Block Macro	21
File Control Block Format	22
File and Record Control I/O Calls	23
Standard System Format	26
<b>Miscellaneous</b>	
MNE GESNAP Snapshot Dumps	27
Octal Patching	28
Slave Program Prefix	29
Compressed Deck Format	30
Standard Character Set	31
State Word	32
Stat1 Word	32
Slave Service Area Symbol Locations	33
Program Switch Word	33
Octal-Decimal Conversion	34
Powers of Two	36

## INSTRUCTION FORMAT

0	17	18	26	27	28	29	30	31	32	35
y		Op Code			i	0	Tag		t <sub>m</sub> t <sub>d</sub>	

y - Address field - even for double-precision operands, even or odd for single-precision operands.

i - Program interrupt inhibit or permit.

t<sub>m</sub> - Type of address modification.

t<sub>d</sub> - Index register designator, mod. variation designator.

Bit 27 - used with EIS and extended memory.

### Address Modification Types (Binary Representation)

t <sub>m</sub> (bits 30-31)		t <sub>m</sub> (bits 30-31)	
00 = R, 01 = RI, 11 = IR		10 = IT	
t <sub>d</sub> (bits 32-35)	Modif.	t <sub>d</sub> (bits 32-35)	Modif.
0000	N	1001	I
1000	X0	1100	DI
1001	X1	1011	AD
...	...	0100	SD
1111	X7	1110	ID
0001	AU	1101	DIC
0101	AL	1111	IDC
0010	QU	1000	CI
0110	QL	1010	SC
0100	IC	0101	SCR
0011	DU	0000	F
0111	DL		

Notes:

SCR for Series 6000 only.

DU\* and DL\* are illegal.

Types 10, 11, ..., 30, 31, ..., and 70, 71, ... refer to index registers.

### Address Modification Types (Octal Representation)

	0	1	2	3	4	5	6	7
0	N	AU	QU	DU	IC	AL	QL	DL
1	0	1	2	3	4	5	6	7
2	N*	AU*	QU*		IC*	AL*	OL*	
3	0*	1*	2*	3*	4*	5*	6*	7*
4	F				SD	SCR		
5	CI	I	SC	AD	DI	DIC	ID	IDC
6	*N	*AU	*QU	*DU	*IC	*AL	*OL	*DL
7	*0	*1	*2	*3	*4	*5	*6	*7

### Indirect Word Formats for IT Modification

Instruction or Previous Indirect Word Modifier

	0	17	18	29	30	31	32	33	35
ID,DI	Address	Ignored							
SCR,SC	Address	Tally	Ignored						Char. Pos.
CI	Address	Ignored		t <sub>b</sub>	0	Char. Pos.			
AD,SD	Address	Tally		Delta					
IDC,DIC	Address	Tally		t <sub>m</sub>			t <sub>d</sub>		

ID - Use, then increment address and decrement tally.

DI - Decrement address, increment tally, then use.

t<sub>b</sub> - Character size (0 = 6-bit 1 = 9-bit).

t<sub>m</sub>\*t<sub>d</sub> - See above.

## INSTRUCTION OP CODES

Op Code	Mnemonic	Op Code	Mnemonic	Op Code	Mnemonic
001	MME	315	CANA	553	SMCM
002	DRL	316	CANQ	554	STC1
011	NOP	317	CANAQ	560	RPD
015	CIOC	32n	LCXn	565	FDV
02n	ADLXn	335	LCA	567	DFDV
032	*LDQC	336	LQ	573	FNO
033	ADL	337	LCAQ	575	FSB
034	*LDAC	34n	ANSXn	577	DFSB
035	ADLA	355	ANSA	600	TZE
036	ADLQ	356	ANSQ	601	TNZ
037	ADLAQ	36n	ANXn	602	TNC
04n	ASXn	375	ANA	603	TRC
054	AOS	376	ANQ	604	TMI
055	ASA	377	ANAQ	605	TPL
056	ASQ	401	MPF	607	TFP
057	*SSCR	402	MPY	614	TEO
06n	ADXn	405	CMG	615	TEU
071	AWCA	411	LDE	616	DIS
072	AWCQ	413	*RSCR	617	TOV
073	LREG	415	ADE	62n	EAXn
075	ADA	421	UFM	630	RET
076	ADO	423	DUFM	634	LDI
077	ADAQ	425	FCMG	635	EAA
10n	CMPXn	427	DFCMG	636	EAQ
111	CWL	430	FSZn	637	LDT
115	CMPA	431	FLD	64n	ERSXn
116	CMPQ	433	DFLD	655	ERSA
117	CMPAQ	435	UFA	656	ERSQ
12n	SBLXn	437	DUFA	66n	ERXn
135	SBLA	44n	SXLn	674	*LCPR
136	SBLQ	450	STZ	675	ERA
137	SBLAQ	451	SMIC	676	ERQ
14n	SSXn	452	SCPR	677	ERAO
155	SSA	454	STT	70n	TSXn
156	SSQ	455	FST	710	TRA
16n	SBXn	456	STE	715	TSS
171	SWCA	457	DFST	716	XEC
172	SWCQ	461	FMP	717	XED
175	SBA	463	DFMP	72n	LXLn
176	SBQ	470	FSTR	731	ARS
177	SBAQ	471	*FRD	732	QRS
20n	CNAXn	472	*DFSTR	733	LRS
211	CMK	473	*DFRD	735	ALS
214	SZNC	475	FAD	736	QLS
215	CNAQ	477	DFAD	737	LLS
216	CNAQ	500	RPL	74n	STXn
217	CNAAQ	505	BCD	750	STC2
22n	LDXn	506	DIV	751	STCA
230	LBAR	507	DVF	752	STCQ
233	RMCM	513	FNEG	753	SREG
234	SZN	515	FCMP	754	STI
235	LDA	517	DFCMP	755	STA
236	LDQ	520	RPT	756	STQ
237	LDAQ	525	FDI	757	STAQ
24n	ORSXn	527	DFDI	771	ARL
255	ORSA	531	NEG	772	ORL
256	ORSQ	533	NEGL	773	LRL
26n	ORXn	535	UFS	774	GTB
275	ORA	537	DUFS	775	ALR
276	ORQ	550	SBAR	776	QLR
277	ORAQ	551	STBA	777	LLR
30n	CANXn	552	STBQ		

\*Series 6000 only

## INSTRUCTION MNEMONICS

Abbreviations: Char. — Character; DP — Double Precision; Flt. — Floating.

Note: Where " is used, this means that all of the words, except the last word, of the preceding full instruction title should be read; e.g.,

DP Flt. Add

"Compare Magnitude

means "DP Flt. Compare Magnitude"

Mnemonic	Op Code	Instruction	Mnemonic	Op Code	Instruction
ADA	075	Add to A	CNAAQ	217	" AQ
ADAQ	077	" AQ	CNAQ	216	" Q
ADE	415	Flt. Add to E	CNAXn	20n	" Xn
ADL	033	Add Low to AQ	CWL	111	Compare with Limits
ADLA	035	Add Logic to A	DFAD	477	DP Flt. Add
ADLAQ	037	" AQ	DFCMG	427	" Compare Magnitude
ADLQ	036	" Q	DFCMP	517	" Compare
ADLXn	02n	" Xn	DFDI	527	" Divide
ADQ	076	Add to Q			Inverted
ADXn	06n	" Xn	DFDV	567	" Divide
ALR	775	A Left Rotate	DFDL	433	" Load
ALS	735	A Left Shift	DFMP	463	" Multiply
ANA	375	AND to A	*DFRD	473	" Floating Round
ANAQ	377	" AQ	DFSB	577	" Subtract
ANQ	376	" Q	DFST	457	" Store
ANSA	355	" Storage A	*DFSTR	472	" Floating Store
ANSQ	356	" Storage Q			Rounded
ANSXn	34n	" Storage Xn	DIS	616	Delay Until Interrupt
ANXn	36n	" Xn			Signal
AOS	054	Add One to Storage	DIV	506	Divide Integer
ARL	771	A Right Logic	DRL	002	Derail
ARS	731	A Right Shift	DUFA	437	DP Unnormalized
ASA	055	Add to Storage From A			Flt. Add
ASQ	056	" Q	DUFM	423	" Multiply
ASXn	04n	" Xn	DUFS	537	" Subtract
AWCA	071	Add with Carry to A	DVF	507	Divide Fraction
AWCQ	072	" Q			
BCD	505	Binary to BCD	EAA	635	Effective Address to A
CANA	315	Comparative AND with A	EAQ	636	" Q
CANAQ	317	" AQ	EAXn	82n	" Xn
CANQ	316	" Q	ERAQ	675	EXCLUSIVE OR to A
CANXn	30n	" Xn	ERAQ	677	" AQ
CIOC	015	Connect I/O Channel	ERQ	676	" Q
CMG	405	Compare Magnitude	ERSA	655	" Storage A
CMK	211	Compare Masked	ERSQ	656	" Storage Q
CMPA	115	Compare with A	ERSXn	64n	" Storage Xn
CMPAQ	117	" AQ	ERXn	66n	" Xn
CMPQ	116	" Q	FAD	475	Flt. Add
CMPXn	10n	" Xn	FCMG	425	Flt. Compare Magnitude
CNAQ	215	Comparative NOT with A	FCMP	515	Flt. Compare
			FDI	525	Flt. Divide
					Inverted
			FDV	565	Flt. Divide

\*Series 6000 only

Mnem- onic	Op Code	Instruction	Mnem- onic	Op Code	Instruction
FLD	431	Flt. Load	ORA	275	OR to A
FMP	461	Flt. Multiply	ORAQ	277	" AQ
FNEG	513	Flt. Negate	ORQ	276	" Q
FNO	573	Flt. Nor- malize	ORSA	255	" Storage A
			ORSQ	256	" Storage Q
*FRD	471	Floating Round	ORSXn	24n	" Storage Xn
FSB	575	Flt. Subtract	ORXn	26n	" Xn
FST	455	Flt. Store			
FSTR	470	Flt. Store Rounded	QLR	776	Q Left Rotate
			QLS	736	Q Left Shift
FSZN	430	Flt. Set Zero & Negative Indicators From Memory	QRL	772	Q Right Logic
			ORS	732	Q Right Shift
			RET	630	Return
			RMCM	233	Read Memory Controller Mask Register
GTB	774	Gray to Binary	RPD	560	Repeat Double
			RPL	500	Repeat Link
LBAR	230	Load BAR	RPT	520	Repeat
LCA	335	Load Comple- ment into A	*RSCR	413	Read SC Register
			SBA	175	Subtract From A
LCAQ	337	" AQ	SBAQ	177	" AQ
LCQ	336	" Q	SBAR	550	Store BAR
*LCPR	674	Load CP Register	SBLA	135	Subtract Logic From A
*LCXn	32n	" Xn	SBLAQ	137	" AQ
LDA	235	Load A	SBLQ	136	" Q
*LDAC	034	Load A & Clear	SBLXn	12n	" Xn
LDAQ	237	" AQ	SBO	176	Subtract From Q
LDE	411	" E (Flt.)	SBXn	16n	" Xn
LDI	634	" IR	*SCPR	452	Store CP Register
LDO	236	" Q	SMCM	553	Set Memory Controller Mask Register
*LDQC	032	Load Q & Clear	SMIC	451	" Interrupt Cells
LDT	637	Load TR	SREG	753	Store Registers
LDXn	22n	Load Xn From Upper	SSA	155	Subtract Stored from A
LLR	777	Long Left Rotate	*SSCR	057	Set SC Register
LLS	737	Long Left Shift	SSQ	156	Subtract Stored from Q
LREG	073	Load Registers	SSXn	14n	" Xn
LRL	773	Long Right Logic	STA	755	Store A
LRS	733	Long Right Shift	STAQ	757	" AQ
LXLn	72n	Load Xn From Lower	STBA	551	Store 9-Bit Char. of A
MME	001	MME	STBQ	552	" Q
MPF	401	Multiply Frac- tion	STC1	554	Store IC+1
MPY	402	Multiply Integer			
NEG	531	Negate A			
NEGL	533	Negate Long			
NOP	011	No Operation			

\*Series 6000 only

Mnem- onic	Op Code	Instruction	Mnem- onic	Op Code	Instruction
STC2	750	Store IC-2	TEO	614	Transfer on Exponent Overflow
STCA	751	Store 6-Bit Char. of A	TEU	615	" Exponent Underflow
STCO	752	" Q	TMI	604	" Minus
STE	456	Store E (Flt.)	TNC	602	" No Carry
STI	754	Store IR	TNZ	601	" Not Zero
STQ	756	Store Q	TOV	617	" Overflow
STT	454	Store TR	TPL	605	" Plus
STXn	74n	Store Xn in Upper	TRA	710	" Uncondi- tionally
STZ	450	Store Zero	TRC	603	" Carry
SWCA	171	Subtract With Carry from A	TSS	715	" & Set Slave
SWCQ	172	" Q	TSXn	70n	" & Set Xn
SXLn	44n	Store Xn in Lower	TTF	607	" Tally Run- out Indica- tor Off
SZN	234	Set Zero & Negative Indicators From Memory	TZE	600	" Zero
			UFA	435	Unnormalized Flt. Add
*SZNC	214	Set Zero & Negative Indicators From Storage & Clear	UFM	421	" Multiply
			UFS	535	" Subtract
			XEC	716	Execute
			XED	717	Execute Double

\*Series 6000 only

#### EIS OP CODES

Op Code	Mnemonic	Op Code	Mnemonic
020(1)	MVE	303(1)	CMPN
024(1)	MVNE	305(1)	DTB
060(1)	CSL	443(1)	SAREG
061(1)	CSR	447(1)	SPL
064(1)	SZTL	463(1)	LAREG
065(1)	SZTR	467(1)	LPL
066(1)	CMPB	500(1)	A9BD
100(1)	MLR	501(1)	A6BD
101(1)	MRL	502(1)	A4BD
106(1)	CMPC	503(1)	ABD
120(1)	SCD	507(1)	AWD
121(1)	SCDB	520(1)	S9BD
124(1)	SCM	521(1)	S6BD
125(1)	SCMB	522(1)	S4BD
160(1)	MVT	523(1)	SBD
164(1)	TCT	527(1)	SWD
165(1)	TCTR	54n(1)	ARAn
202(1)	AD2D	56n(1)	AARn
203(1)	SB2D	600(1)	TRTN
206(1)	MP2D	601(1)	TRTF
207(1)	DV2D	604(1)	TMOZ
222(1)	AD3D	605(1)	TPNZ
223(1)	SB3D	606(1)	TTN
226(1)	MP3D	64n(1)	ARNn
227(1)	DV3D	66n(1)	NARn
300(1)	MVN	74n(1)	SARn
301(1)	BTD	76n(1)	LARn

## EIS MNEMONICS

Abbreviations: Char. — Character; Disp. — Displacement; Spec. — Specified;  
Reg. — Register; Alpha. — Alphanumeric; Trunc. — Truncation.

Mnemonic	Op Code	Instruction
A4BD	502(1)	Add 4-Bit Char. Disp. to Spec. AR
A6BD	501(1)	Add 6-Bit Char. Disp. to Spec. AR
A9BD	500(1)	Add 9-Bit Char. Disp. to Spec. AR
AARn	56n(1)	Alpha. Descriptor to ARn
ABD	503(1)	Add Bit Disp. to Spec. AR
AD2D	202(1)	Add Using 2 Decimal Operands
AD3D	222(1)	Add Using 3 Decimal Operands
ARAn	54n(1)	ARn to Alpha. Descriptor
ARNn	64n(1)	ARn to Numeric Descriptor
AWD	507(1)	Add Word Disp. to Spec. AR
BTD	301(1)	Binary to Decimal Convert
CMPB	066(1)	Compare Bit Strings
CMPc	106(1)	Compare Alpha. Char. String
CMPN	303(1)	Compare Numeric
CSL	060(1)	Combine Bit Strings Left
CSR	061(1)	Combine Bit Strings Right
DTB	305(1)	Decimal to Binary Convert
DV2D	207(1)	Divide Using 2 Decimal Operands
DV3D	227(1)	Divide Using 3 Decimal Operands
LARn	76n(1)	Load Address Reg. n
LAREG	463(1)	Load Address Reg.
LPL	467(1)	Load Pointers and Lengths
MLR	100(1)	Move Alpha. Left to Right
MP2D	206(1)	Multiply Using 2 Decimal Operands
MP3D	226(1)	Multiply Using 3 Decimal Operands
MRL	101(1)	Move Alpha. Right to Left
MVE	020(1)	Move Alpha. Edited
MVN	300(1)	Move Numeric
MVNE	024(1)	Move Numeric Edited
MVT	160(1)	Move Alpha. with Translation
NARn	66n(1)	Numeric Descriptor to ARn
S4BD	522(1)	Subtract 4-Bit Char. Disp. from Spec. AR
S6BD	521(1)	Subtract 6-Bit Char. Disp. from Spec. AR
S9BD	520(1)	Subtract 9-Bit Char. Disp. from Spec. AR
SARn	74n(1)	Store Address Reg. n
SAREG	443(1)	Store Address Reg.
SB2D	203(1)	Subtract Using 2 Decimal Operands
SB3D	223(1)	Subtract Using 3 Decimal Operands
SBD	523(1)	Subtract Bit Disp. from Spec. AR
SCD	120(1)	Scan Char. Double
SCDR	121(1)	Scan Char. Double in Reverse
SCM	124(1)	Scan with Mask
SCMR	125(1)	Scan with Mask in Reverse
SPL	447(1)	Store Pointers and Lengths
SWD	527(1)	Subtract Word Disp. from Spec. AR
SZTL	064(1)	Set Zero and Trunc. Indicators with Bit Strings Left
SZTR	065(1)	Set Zero and Trunc. Indicators with Bit Strings Right
TCT	164(1)	Test Char. and Translate
TCTR	165(1)	Test Char. and Translate in Reverse
TMOZ	604(1)	Transfer on Minus or Zero
TPNZ	605(1)	Transfer on Plus and Nonzero
TRTF	601(1)	Transfer on Trunc. Indicator OFF
TRTN	600(1)	Transfer on Trunc. Indicator ON
TTN	606(1)	Transfer on Tally Runout Indicator ON

## EXTENDED MEMORY OP CODES AND MNEMONICS

Mnemonic	Op Code	Instruction
LBER	572	Load Base Extension Register
LBMA	570	Load Master Base Address Register A
LBMB	571	Load Master Base Address Register B
MLDA	235(1)	Extended Memory LDA
MLDQ	236(1)	Extended Memory LDQ
MLDAQ	237(1)	Extended Memory LDAQ
MSTA	755(1)	Extended Memory STA
MSTQ	756(1)	Extended Memory STO
MSTAQ	757(1)	Extended Memory STAQ
SBER	157	Store Base Extension Register
SBMA	555	Store Master Base Address Register A
SMBB	556	Store Master Base Address Register B

## INDICATOR REGISTER BIT POSITION

Bit Position	Indicator
18	Zero
19	Negative
20	Carry
21	Overflow
22	Exponent Overflow
23	Exponent Underflow
24	Overflow Mask
25	Tally Runout
26	Parity Error
27	Parity Mask
28	Master Mode
29	} Must be Zero
30	
31	
32	
33	
34	
35	

- The Tally Runout Indicator will reflect C(Y)<sub>25</sub> regardless of what address modification is performed on the LDI and RET instructions (for Tally Operations).
- The C(Y)<sub>25</sub> will contain the state of the Tally Runout Indicator prior to address modification of the STI and STC1 instructions (for Tally operations).
- A possible change of the status of the Master Mode Indicator takes place as the last part of the RET instruction execution.
- For EIS, bit 29 indicates truncation, bit 30 indicates multiword instruction interruption.

### EFFECT OF COMPARES ON INDICATORS

Note: The following abbreviations are used: Z - Zero Indicator, N - Negative Indicator, C - Carry Indicator.

#### CWL

Zero - If (Y) contained in closed interval [(A); (Q)], i. e.,  
 $(A) \leq (Y) \leq (Q)$  or  $(A) \geq (Y) \geq (Q)$ ,  
 then ON; else OFF.

	NC	(sign)
	00 (Q) > (Y)	(Q) <sub>0</sub> = 0, (Y) <sub>0</sub> = 1
	01 (Q) ≥ (Y)	(Q) <sub>0</sub> = (Y) <sub>0</sub>
	10 (Q) < (Y)	(Q) <sub>0</sub> = 1, (Y) <sub>0</sub> = 0
	11 (Q) < (Y)	(Q) <sub>0</sub> = 1, (Y) <sub>0</sub> = 0
CMPA	(A) :: (Y)	
CMPQ	(Q) :: (Y)	
CMPAQ	(AQ) :: (Y-pair)	
CMPXn	(Xn) :: (Y) <sub>0-17</sub>	

Algebraic (Signed Fixed-Point)  
Comparison

Logic (Unsigned Fixed-Point)  
Comparison

ZNC	(sign)
000 >	( ) <sub>0</sub> = 0, ( ) <sub>0</sub> = 1
001 >	( ) <sub>0</sub> = ( ) <sub>0</sub>
101 =	( ) <sub>0</sub> = ( ) <sub>0</sub>
010 <	( ) <sub>0</sub> = 1, ( ) <sub>0</sub> = 0
011 <	( ) <sub>0</sub> = 1, ( ) <sub>0</sub> = 0

ZC
00 <
11 =
01 >

CMG	[(A) :: (Y)]
FCMG	[(E) (AQ) <sub>0-27</sub> ] :: (Y)
DFCMG	[(E) (AQ) <sub>0-63</sub> ] :: (Y-pair)
FCMP	[(E) (AQ) <sub>0-27</sub> ] :: (Y)
DFCMP	[(E) (AQ) <sub>0-63</sub> ] :: (Y-pair)
SZN	Number (Y) :: 0
FSZN	Mantissa (Y) <sub>8-35</sub> :: 0

#### ZN

00 >
10 =
01 <

### INSTRUCTION COUNTER FAULT CODES

Fault Code	Fault Name	Group (Priority)	IC Contents
1100	Startup	I	N+0, +1, or +2
1111	Execute	I	N+0, +1, or +2
1011	Operation Not Completed	II	N+0, +1, or +2
0111	Lockup	II	N+0, +1, or +2
1110	Divide Check	III	N
1101	Overflow	III	N
1001	Parity	IV	N
0101	Command	IV	N
0001	Memory	IV	N
0010	Master Mode Entry	IV	N
0110	Derail	IV	N
0011	Fault Tag	IV	N
1010	Illegal Op Code	IV	N
1000	Connect	V	N
0100	Timer Runout	V	N
0000	Shut Down	V	N

Notes: N = Last Operation

Group I, II - operations abort unconditionally

Groups III, IV - operations abort after current operation completed

Group V - identical to program interrupts but with priority over program interrupts

### PSEUDO-OPS

Legend: Where ON/OFF is shown, the underlined option (e.g., ON/OFF) is the normal mode. Abbreviations used: = blanks; / = or (e.g., r/ means "remarks or blanks"); e = expression; i = integer; r = remarks; s = symbol.

Columns	2	8	16	Name
Control				*in col. 1, remarks in 2-80
-	ABS		16 = blank	Output Absolute Text
-	CPR		G or H, year	Copyright
-	DATE		16 = blank	Current Date
-	DCARD		i, BCD char	Punch BCD Card
-	DETAIL		ON/OFF	Detail Output Listing
-	EDITP		ON/OFF	Edit Print Lines
-	EJECT		16 = blank	Restore Output Listing
s/-	END		e/-	End of Assembly
-	FLOAT		ON/OFF	Floatable Code
-	FUL		16 = blank	Output Full Binary Text
-	HEAD			Heading (1-7 subfields)
-	INHIB		ON/OFF	Inhibit interrupts
-	LBL		x,y	Label
-	LIST		ON/OFF	Control Output Listing
-	MASK		i/-	Create Mask
⓪	OPD			Operation Def. (1-n subfields)
Ⓛ	OPSYN		op code	Operation Synonym
-	PCC		ON/OFF	Print Control Cards
-	PMC		ON/OFF	Print Macro Expansion
-	PUNCH		ON/OFF	Control Card Output
-	REF		ON/OFF	References
-	REFMA		ON/OFF	Reference Macro
r/-	REM			Remarks (col. 12...)
s/-	TCD		4	Punch Transfer Card
i/-	TTL			Title in cols. 16-72
s/-	TTLDAT			Title Date
i/-	TTLS			Subtitle in cols. 16-72
Location Counter				
-	BEGIN		s,s	Origin of a Location Counter
s/-	EIGHT			Force Loc. Ctr. - Mult. of 8
s/-	EVEN			Force Loc. Ctr. - Even
-	LOC		e	Location of Output Text
s/-	ODD			Force Loc. Ctr. - Odd
s/-	ORG		e	Origin Set By Programmer
-	USE		s/-/PREVIOUS	Use Multiple Location Counters
Symbol Defining				
s	BOOL		e	Boolean
s	EQU		e	Equal To
s	FEQU		s	Special Fortran Equivalence
s	MAX		e,...	Maximum
s	MIN		e,...	Minimum
s	NULL			Null
s	SET		e	Symbol Redefinition
-	SYMDEF		s,...	Symbol Definition
-	SYMREF		s,...	Symbol Reference
Data Generating				
s/-	ASCII		i,s	ASCII Coded Information
s/-	EBCDIC		i,s	EBCDIC Coded Information
s/-	BCI		i,s	BCI Information
s/-	DEC		i,...	Decimal
s/-	DUP		e,0	Duplicate Cards
s/-	OCT		i,...	Octal
s/-	VFD		e,...	Variable Field Definition

## CALL, SAVE, RETURN PSEUDO-OPS

Examples of the CALL, SAVE, and RETURN pseudo-ops in relocatable mode are presented below.

### CALL

AAAAA — Location of the CALL instruction.  
 SUB — Name of subroutine called.  
 MOD — Address modifier.  
 A1-An — Arguments to be used in called subroutine.  
 E1-Em — Locs. for error returns from subroutine.  
 E.I. — Identifier (optional).  
 .E.L.. — Location where error linkage information is stored.

Note: When a CALL to an external subprogram appears within a headed section, the external subprogram must be identified by a 6-character symbol (immune to HEAD). If a CALL is being used to access an internally defined subroutine, the subroutine must be placed ahead of the CALL in the program deck. Also, a SYMDEF pseudo-op with the symbol identifying the subroutine in its variable field must be placed ahead of the CALL in the program deck. Starting the subroutine with a SAVE pseudo-op automatically provides the SYMDEF.

AAAAA CALL SUB, MOD(A1, A2, ..., An) E1, E2, ..., Em'E.I.'

AAAAA	TSX1	SUB, MOD
	TRA	*+2+n+m
	ZERO	.E.L., .E.I.
	ARG	A1
	ARG	A2
	...	
	ARG	An
	TRA	Em
	...	
	TRA	E2
	TRA	E1

### SAVE

BBBBB, which must always be present, is used for referencing by the RETURN pseudo-op. The symbols  $i_1$  through  $i_n$  are integers 0-7 specifying index registers to be saved. .E.L.. defines the location provided for the contents of the error linkage register.

BBBBB	SAVE	$i_1, i_2, \dots, i_n$
BBBBB	TRA	*+2+n
	LDX( $i_1$ )	** , DU
	.	.
	.	.
	LDX( $i_n$ )	** , DU
	RET	.E.L.
	STI	.E.L.
	STX1	.E.L.
	STX( $i_1$ )	BBBBB+1
	STX( $i_2$ )	BBBBB+2
	.	.
	.	.
	STX( $i_n$ )	BBBBB+n

The following coding is generated when the SAVE variable field is blank:

NAME	TRA	*+2
	RET	.E.L.
	STI	.E.L.
	STX1	.E.L.

Columns				Name
2	8	16		
<b>Storage Allocation</b>				
s/-	BFS	e		Block Followed by Symbol
	BLOCK	s		Block Common
s/-	BSS	e		Block Started by Symbol
s/-	LIT	16 = blank		Literal Pool Origin
<b>Conditional</b>				
-	IFE	e,e,i		If Equal
-	IFG	e,e,i		If Greater Than
-	IFL	e,e,i		If Less Than
-	INE	e,e,i		If Not Equal
s	MARK	c/-		Specify Symbol
<b>Special Word Formats</b>				
s	ARG	x,y		Generate Zero Op Code
-	MAXSZ	i		Maximum Size of Assembly
	NONOP			Undefined Operation
s/-	ZERO	x,y		Generate Word, 2 18-Bit Fields
<b>Address Tally</b>				
	TALLY	a,t,c		Tally - ID, DI, SC, SCR, CI
	TALLYB	a,t,b		Tally Byte - SC, SCR, CI
	TALLYC	a,t, mod		Tally & Continue - DIC, IDC
	TALLYD	a,t,d		Tally & Delta - AD, SD
<b>Macro</b>				
-	CRSM	ON/OFF		Created Symbols
s/-	DELM	s		Delete Macro
-	ENDM	s		End Macro
-	IDRP	i/-		Indefinite Repeat
-	LODM	s		Load Macros
-	LODST	s		Load Symbol Table
s	MACRO	-		Macro Ident.
-	ORGCSM	e		Origin Created Symbols
-	PRGST			Purge Symbol Table
-	PUNM			Punch Macro Prototypes & Controls
-	PUNST			Punch Symbol Table
<b>Program Linkage</b>				
s/-	CALL	⊙		Call—Subroutines
s	SAVE	⊙		Save—Return Linkage Data
s/-	RETURN	⊙		Return—From Subroutines
-	ERLK	16 = blank		Error Linkage—to Subroutines
<b>Miscellaneous</b>				
	ETC	a,b,...		Continuation

- ⊙ New op code
- ⊙ Symbol or op code
- ⊙ See "CALL, SAVE, RETURN PSEUDO-OPS" below.



## RETURN

The instructions generated by a RETURN pseudo-op must make reference to a SAVE pseudo-op within the subroutine being exited from. This is done by the first subfield of RETURN, which must always be present; it specifies a symbol which is defined by its presence in the location field of a SAVE pseudo-op. The second subfield is optional; if present, it specifies the error return to be made (i. e., if the second subfield contains the value k, then the return is made to the kth error return).

### RETURN BBBB

TRA BBBB+1

### RETURN BBBB, K

LXD1 .E. L. . . I  
 SBX1 k, DU  
 STX1 .E. L. .  
 TRA BBBB+1

## MME'S

Symbol	Meaning	Decimal	Octal
GEINOS	Input/Output Initiation	1	1
GEROAD	Roadblock	2	2
GEFADD	Physical File Address Request	3	3
GERELS	Component Release	4	4
GESNAP	Snapshot Dump	5	5
GELAPS	(Elapsed) Time Request	6	6
GEFINI	Terminal Transfer to Monitor	7	7
GEBORT	Aborting of Programs	8	10
GEMORE	Additional Memory or Peripherals	9	11
GEFCON	File Control Block Request	10	12
GEFILS	File Switching Request	11	13
GESETS	Set Switch Request	12	14
GERETS	Reset Switch Request	13	15
GEENDC	Terminate Courtesy Call	14	16
GERELC	Relinquish Control	15	17
GESPEC	Special Interrupt Courtesy Call Request	16	20
GETIME	Date and Time-of-Day Request	17	21
GECALL	System Loader	18	22
GESAVE	Write File in System Format	19	23
GERSTR	Read File in System Format	20	24
GEMREL	Release Memory	21	25
GESYOT	Write on SYSOUT	22	26
GECHK	Check Point	23	27
GEROUT	Output to Remote Terminal	24	30
GEROLL	Reinitiate or Rollback Program	25	31
GEUSER	User-Supplied MME	26	32
GELoop	Loop Protection	27	33
GEWAKE	Call Me When	28	34
GEIDSE	I-D-S I/O Request	29	35
EMM	Enter Master Mode	30	36
GELBAR	Load Base Address Register	31	37
GEFRCE	ISP Abort Processing	32	40
GEFSYE	File System Entry Point	33	41
GEPRIO	I/O Priority	34	42
GENEWS	Spawn New Job	35	43
GESNUM	Supply SNUMB Number	36	44
GEINFO	Information GCOS Entry	37	45

## LITERAL FORMATS

The examples shown below illustrate literal formats.

	Decimal	Octal
=-10	Integer	=O1257
=26.44167E-1	Single-precision floating point	=O37777777742
=1.27743675385D0	Double-precision floating point	
=22.5B5	Fixed-point	

Hollerith	Instruction	Variable Field
=HALPHA1	=MARGBETA	=V10/895,5/37, H6/C, 15/ALPHA
=HGONE	=MLDAB5,1	=V18/ALPHA, O12/235,6/0
=4HGONE		
=7HTHEBEND		

### Literals Modified by DU and DL

(coded literal)	(resultant address field of instruction)	(operand used by instruction)
=100, DL	000144	00000000144
=-1, DU	001000	001000000000
=320., DU	022500	022500000000
=0., DU	400000	400000000000
=O77, DU	000077	000077000000
=2B25, DU	004000	004000000000
=3H00A, DL	000021	000000000021

## GMAP ERROR FLAGS

- U - Undefined - Undefined symbol(s) appear in the variable field.
- M - Multidefined - Multiple-defined symbol(s) appear in the location field and/or the variable field.
- A - Address - Illegal value or symbol appears in the variable field. Also used to denote lack of a required field.
- X - Index - Illegal index or address modification.
- R - Relocation - Relocation error; expression in the variable field will produce a relocatable error upon loading.
- P - Phase - Phase error; this implies undetected machine error or symbols becoming defined in Pass two with a different value from Pass one.
- E - Even - Inappropriate character in column 7.
- C - Conversion - Error in conversion of either a literal constant or a subfield of a data-generative pseudo-operation. Illegal character.
- L - Location - Error in the location field.
- O - Operation - Illegal operation.
- T - Table - An assembly table overflowed, not permitting proper processing of this card completely. Table overflow error information will appear at the end of testing.

## RELOCATABLE OBJECT DECKS

### Preface Card Format

	0	2	3	8	9	11	12	17	18	19	35	
Word 1	100	V	101	Word Count		Length of Subprogram						
Word 2	Checksum of cols. 1-3 and 7-72											
Word 3	Length of BLANK COMMON required by program						2* (No. of Preface Symbols)					
Words 4-23	0	5	6	11	12	17	18	23	24	29	30	35
Word n	Char. 1	Char. 2	Char. 3	Char. 4	Char. 5	Char. 6						
Word n+1	A						K					
	18		19		32		33					

Word 1, Bits 3-8: V ( $5 < V \leq 17$ ) represents size of field within a special relocation entry pointing to the specific preface card entry. Thus,  $V = \log_2 N + 1$ , where N = number of LABELED COMMON and SYMREF entries.

Word 3, Bit 18: If = 1, subprogram or LABELED COMMON region (word n) is to be loaded starting in the next available location which is a multiple of eight.

### Words 4-23:

Symbol in Word n	A in Word n+1	K in Word n+1
Primary SYMDEF	Entry Value	0
Secondary SYMDEF	Entry Value	1
Constant	Entry Value	3
Local LABELED COMMON	Length	4
SYMREF	Zero	5
LABELED COMMON	Length	6
.SYM. LABELED COMMON	Length	7

Word n+1, Bit 18: If = 1 when K = 6 or 7, subprogram region is to be loaded starting in next available location which is a multiple of eight.

Word n+1, Bits 19-32: If = address when K = 5, Bits 19-22 is relative card number, Bits 23-32 is relative block address.

### Text Card Format

	0	2	3	8	9	11	12	17	18	35				
Word 1	010		101	Word Count		Loading Address								
Word 2	Checksum of Cols. 1-3 and 7-72													
	0	4	5	9	10	14	15	19	20	24	25	29	30	34
Word 3	A	B	C	D	E	2	3	4	5	6	7			
Word 4	A	B	C	D	E	8	9	10	11	12	13	14		
Word 5	A	B	C	D	E	15	16	17	18	19				

Word 1, Bits 3-8 indicate symbol relative to which text is loaded. Words 3-5 contain 19 five-bit relocation identifiers. Each 5-bit identifier contains relocation information for each instruction or data word on the card (up to 19 words). The relocation scheme identifiers are interpreted by the General Loader as follows:

- Bit A - 0 (Reserved for future use)
- Bits BC - Left half-word
- Bits DE - Right half-word

To every 18-bit half-word the following code applies:

- 00 Absolute value (not modified by General Loader)
- 01 Relocatable value
- 10 Blank Common
- 11 Special entry value

Words 6-24: Instructions and data.

### I/O SELECT SEQUENCE

Select Sequence	MME	GEINOS
	1/0 Command Word	ZERO
	FILWRD	DCWLST
	ZERO	STSWRD, CCADDR
FILWRD	File Code Word (bits 0-17 = PAT Pointer, 24-35 = File Code).	
DCWLST	Start of DCW List.	
STSWRD	Status Return Word Pair.	
CCADDR	Entry Point to User's Courtesy Call Routine (optional).	

Note: The following I/O requests have a 5-word select sequence: Seek, then Read or Write Disk; Select, then Read or Write Drum; Write, then Read Typewriter. In I/O Command Word 1, if the IOC/IOM Command Code is zero and the Count field contains a two, the Count field contains a two, the select sequence is 5 words.

I/O Command Word 1  
File Identification Word 1  
I/O Command Word 2  
File Identification Word 2  
Status Return Word Point

	0	5	6	17	18	22	23	24	29	30	35
I/O Command Word	Device Command Code	0	IOC/IOM Command Code	0	Control Field	Count Field					

Command	Description	Device Command	Code
Request Status		REOS	00
Reset Status		RESS	40
Read Card Binary		RCB	01
Read Card Binary Continuous		RCBC	01
Read Card Decimal		RCD	02
Read Card Decimal Continuous		RDCD	02
Read Card Mixed		RCM	03
Read Card Mixed Continuous		RCMC	03
Write Card Binary		WCB	11
Write Card Binary Continuous		WCBC	11
Write Card Decimal		WCD	12
Write Card Decimal Continuous		WCDC	12
Write Card Decimal Edited		WCDE	13
Write Card Decimal Edited Continuous		WCDEC	13
Write Printer		WPR	10
Write Printer Continuous		WPRC	10
Write Printer Edited		WPRE	30
Write Printer Edited Continuous		WPREC	30

Command Description	Device Command	Code
Read Tape Binary	RTB	05
Re-Read Tape Binary	RRTB	07
Read Tape Decimal	RTD	04
Re-Read Tape Decimal	RRTD	06
Read Tape 9 Channel	RT9	03
Write Tape 9 Channel	WT9	13
Write Tape Binary	WTB	15
Write Tape Decimal	WTD	14
Write End-of-File	WEF	55
Write File Mark	WFM	15
Write File Mark Decimal	WFMD	14
Erase	ERASE	54
Backspace Record(s)	BSR	46
Backspace File	BSF	47
Forward Space Record(s)	FSR	44
Forward Space File	FSF	45
Rewind	REW	70
Rewind and Standby	REWS	72
Set Low Density	SLD	61
Set High Density	SHD	60
Set File Permit	SFPR	63
Tape Load	TPLD	75
Set 200 BPI	S2D	64
Set 556 BPI	S5D	43
Set 800 BPI	S8D	42
Set 1600 BPI	S16D	65
Set High Density Default	SHD	60
Set Low Density Default	SLD	61
Write EBCDIC Record	WTE	34
Read EBCDIC Record	RTE	24
Write ASCII Record	WTA	37
Read ASCII Record	RTA	27
Write ASCII/EBCDIC Record	WTAE	35
Read ASCII/EBCDIC Record	RTAE	25
Seek Disk Address	SDIA	34
Read Disk Continuous	RDIC	25
Write Disk Continuous	WDIC	31
Write Disk Continuous and Verify	WDICV	33
Select Drum Address	SDRA	34
Read Drum	RDR	25
Write Drum	WDR	31
Write Drum and Verify	WDRV	33
Drum Compare and Verify	DRCV	11
Read Perforated Tape	RDPT	02
Write Perforated Tape	WPT	11
Write Perforated Tape Edited	WPTE	31
Write Perforated Tape - Single Character	WPTSC	16
Write Perforated Tape - Double Character	WPTDC	13
Read Typewriter	RTYP	03
Write Typewriter	WTYP	13
Write Typewriter and Return to Road	WTYPR	13
Read DATANET 30/305	RDN	01
Write DATANET 30/305	WDN	10

Series 6000 only

Command Description	Device Command	Code
Write Card ASCII	WCDASC	14
Write Card ASCII Continuous	WCDCAS	14
Write Card EBCDIC	WCDEBC	15
Write Card EBCDIC Continuous	WCDCBC	15
Read Card ASCII	RCDASC	04
Read Card ASCII Continuous	RCDCAS	04
Read Card ASCII Mixed	RCDMAS	05
Read Card ASCII Mixed Continuous	RCDCMA	05
Read Card EBCDIC	RCDEBC	06
Read Card EBCDIC Continuous	RCDCBC	06
Read Card Mixed With Switching to Mixed ASCII	RCMA	07
Read Card Mixed ASCII Continuous	RCDMAC	07
Set Reader Native Mode	SRNM	65
Write Printer ASCII	WPRASC	14
Write Printer ASCII Continuous	WPRCAS	14
Write Printer ASCII Edited	WPREAS	34
Write Printer ASCII Edited Continuous	WPRECA	34
Write Console ASCII	WTYPAS	33
Read Console ASCII	RTYPAS	23
Write Disk ASCII	WDASC	32
Read Disk ASCII	RDASC	23
Read Disk Nonstandard Size	RDNSZ	04

IOC/IOM 00001 = Continuous Mode Non-Data-Transfer

Command 00010 = Card Punch

Codes 00100 = Write Single-Character Record

01000 = Program Load

00000 = Unit Record Transfer

Control Field Zero, except when writing EOF or tape mark on magnetic tape, in which case it contains binary representation of character to be written.

Count Field Zero, except when: (a) IOC/IOM Command Code 00001 used - Count Field must be 00001 unless back- or forward-space n records specified, in which case Count Field contains binary integer; (b) for disk or drum I/O, and for typewriter I/O when select sequence contains a Write and Read command, Count Field in I/O Command Word 1 is initialized by GMAP to value 2, to indicate two commands in select sequence.

DCW	0	17 18	20 21 22	23 24	35
Word	Data Address	Zone Control	0(W) 1(R)	Action Code	Word Count

Bit 21: W = Write, R = Read

Action Codes	Transmit and Disconnect	IOTD	00
	Transmit and Proceed	IOTP	01
	Transfer to DCW	TDCW	10
	Nontransmit and Proceed	IONTP	11

Status Return Word 1	0 1 2	5 6	11 12	15 16	17 18	20 21	23 24	29 30	35
	Major Status	Sub-status	0	I	A	IOC/IOM Error Code	Mem. Error Code	0	Record Code Residue

- 0 1 when I/O and recovery action terminate.
- 1 Power Bit
- 2-5 } Refer to General Comprehensive Operating Supervisor manual for status codes.
- 6-11 }
- 16 0 = Termination Interrupt. 1 = Initiation Interrupt.
- 17 1 = Exception Processing cannot recover error
- 18-20 IOC/IOM error Codes
- 000 - no error
  - 001 - primary mailbox error
  - 010 - secondary mailbox error
  - 101 - data error
  - 110 - overflow address (exceeds 2<sup>18</sup> - 1)
  - 111 - illegal address
  - 100 - peri. data error
- 21-23 Memory Error Codes
- 000 - no error
  - 001 - nonexistent address
  - 010 - not master memory for this IOC/IOM
  - 100 - not the control processor for this IOC/IOM
  - 110 - protection area
  - 111 - cycle not properly completed
  - 011 - parity error-memory
- 30-35 (Tape back- or forward-space operations.) No. records remaining that were not spaced over, due to detection of beginning of tape or EOF mark. (An EOF mark will have been counted as a record.)

Status Return Word 2	0	17 18	20 21 22	23 24	35
	Address + 1 of last complete data word transmitted	Char. Count	Action Code	Word Count Residue	

- 18-20 No. characters transmitted in last word (if partial word); 0 if complete word).
- 21 0 = Write on peripheral. 1 = Read from peripheral.
- 22-23 DCW Action Code.
- 24-35 No. words left to be transmitted under control of the DCW when I/O terminated.

## FILE DESIGNATOR WORDS

VFD 18/fcb, 1/io, 1/open, 2/close, 1/prime, 1/size, 1/req, 1/res, 10/file

- | Bits  | Meaning   |
|-------|---|
| 0-17  | Location of FCB.  |
| 18    | 0 = input file; 1 = output file.  |
| 19    | Rewind-on-OPEN: 0 = do not rewind file, or this option does not apply to device; 1 = rewind file when opened.   |
| 20-21 | Rewind-on-CLOSE: 0 = does not apply to device; 1 = lock file when closed, release via MME GERELS; 2 = do not rewind and do not lock file when closed; 3 = rewind but do not lock file when closed.  |
| 22    | Prime input buffer: 0 = normal; 1 = buffered input file, buffer not to be filled when opened.   |
| 23    | Put-size on output: 0 = normal; 1 = buffered output file for which a CALL PUTSZ command with record size zero is to be issued when closed.  |
| 24    | Optional/required status on OPEN: 0 = normal; 1 = abort if file not present.  |
| 25    | Reserved for system use.  |
| 26-35 | File positioning value for OPEN: 0 = honor rewind option in bit 19, but do not otherwise position device; n = this is nth file on multifile reel, rewind tape and space forward over n-1 files. Note: Rewind for a linked file on mass storage means that the file is to be positioned at the beginning of the assigned area for that file. |

## FILE CONTROL BLOCK MACRO

8	16	Subfield
FILCB	LOCSYM of File Control Block	1
ETC	2-Character File Code	2
ETC	LOCSYM of First Buffer	3
ETC	LOCSYM of Second Buffer	4
ETC	Maximum Block Size	5
ETC	Record Form: 0 = Variable, 1 = Fixed, 2 = Mixed	6
ETC	Record Size: No. if Fixed, LOCSYM if Mixed	7
ETC	1 if Standard Block Serial Nos. not Used	8
ETC	LOCSYM of Error Routine	9
ETC	1 if no Standard Labels	10
ETC	Recording Mode: 0 = Binary, 1 = BCD, 2 = Mixed, 3 = 9-track tape	11
ETC	Recording Density: 0 = High, 1 = Low	12
ETC	1 if File is on Multifile Reel	13
ETC	Retention Period in Days	14
ETC	LOCSYM of Preheader Label Routine	15
ETC	LOCSYM of Postheader Label Routine	16
ETC	LOCSYM of Pretrailer Label Routine	17
ETC	LOCSYM of Posttrailer Label Routine	18
ETC	File name	19
ETC	Exception Processing	20
ETC	Partitioned Record	21

Example:

FILCB FILEA, AA,  
ETC BUFA1, BUFA2, 100, 1, 10

## FILE CONTROL BLOCK FORMAT

	0	17 18	29 30 35	
-14	Preheader Label Exit		Reserved For System	MBZ
-13	Postheader Label Exit		Reserved For System	MBZ
-12	Pretrailer Label Exit		Reserved For System	MBZ
-11	Posttrailer Label Exit		Reserved For System	MBZ
-10	First 6 chars. of File-Name			
-9	Last 6 chars. of File-Name			
-8	Retention Period		1 2	Reel Sequence No.
-7	File Serial No.			RFS
-6	Block Count		3 4 5 6 7 8	File Count
-5	Error Routine Exit			
-4	0	20	Station Code	0 21
-3	IOS - Status Return Word 1			
-2	IOS - Status Return Word 2			
-1	FCB Pointer		22 23 24 25	
LOC-SYM	Current Record Index		26 27 28 2 3 3 3	MBZ
+1	Record Size		MBZ	
+2	Loc. of 1st Buffer		Loc. of 2nd Buffer	
+3	Working Storage Word			
+4	Block Size		MBZ	
+5	Record Size Routine		Reserved For System	MBZ

- |                       |                                |
|-----------------------|--------------------------------|
| 1. Label              | 17. No. of buffers             |
| 2. Name               | 18. Internal control           |
| 3. Operator ignore    | 19. Internal control           |
| 4. Override           | 20. Extended recording mode    |
| 5. MBZ                | 21. Extended recording density |
| 6. Noslew             | 22. Device address             |
| 7. Partitioned record | 23. IOC/IOM                    |
| 8. Zero               | 24. 4-bit channel address      |
| 9. File present       | 25. 6-bit channel address      |
| 10. File open         | 26. Record form                |
| 11. File lock         | 27. Recording mode             |
| 12. I/O               | 28. Perforated tape            |
| 13. EOF               | 29. Recording density          |
| 14. Block serial no.  | 30. Internal control           |
| 15. Standard label    | 31. Disk or tape               |
| 16. Multifile reel    | 32. SYSOUT                     |
|                       | 33. Device type                |

Section	Required for Following Files
1	All files.
2	Files with buffers.
3	Tape files having a file name.
4	Tape files requiring special label processing routine(s) in addition to the standard label routines.
5	Files having mixed-length record type.

## FILE AND RECORD CONTROL I/O CALLS

Note: the term "CALL" must precede each I/O call; e.g., CALL OPEN. . .

Legend: The following arguments are used in the calls.

<b>bcd</b>	Address of user's routine to transfer to when BCD 14-word record is encountered. Not required.
<b>buffer</b>	380-word count.
<b>cc</b>	User-supplied courtesy call routine. (Optional).
<b>cflag</b>	0 = BCD; not 0 = ASCII.
<b>code</b>	Indicates type of error (in bits 9-17 of calling sequence word).
<b>cont</b>	First control word for reading (READ) or writing (WRITE) the record.
<b>cont</b>	Word containing, in lower half, count of words to be converted.
<b>eof</b>	User's eof routine.
<b>fcf</b>	Fcb.
<b>filemark</b>	Word specifying, in bits 30-35, the file mark to be written.
<b>function</b>	40 = do not apply delayed posting, release *A file. 41 = apply delayed posting, release *A file.
<b>image</b>	First word of logical record (line or card) to be inserted (WTREC); or of first word of the print line image to be inserted (PRINT and EPRINT); or of first word of image to be inserted (PUNCH).
<b>isize</b>	Size of file.
<b>istat</b>	Status return word.
<b>iprmis</b>	Permissions 1 = read only 2 = write only 3 = read and write
<b>kind</b>	Word specifying the type of card to be produced (if 0, column binary card, if ≠ 0, Hollerith card).
<b>list</b>	First n consecutive file designator words for n files to be opened (OPEN) or closed (CLOSE); or of first entry in list of control parameters (IOEDIT).
<b>loc-of-fcb</b>	Fcb for the file on which the error occurred.
<b>media</b>	Word containing a code in bits 30-35 describing the ultimate form of the record (1 = column binary card, 2 = Hollerith card, 3 = print line).
<b>mflag</b>	0 = batch, note 0 = time sharing.
<b>mode</b>	0 = linked mass storage file; 1 = random mass storage file; 2 = terminal file.
<b>n</b>	No. files (OPEN, CLOSE); or standard filemarks to bypass (FSTFM, BSTFM); or physical records to bypass (n ≤ 63) (FSREC, BSREC); or entries in list (IOEDIT); or initial words in image not requiring editing (EPRINT).
<b>report</b>	Word containing a report code in bits 30-35.
<b>size</b>	Word specifying, in bits 0-17, the no. of words in the last logical output record to this file.
<b>slew</b>	Word whose contents in bits 30-35 control paper movement.
<b>stor</b>	First cell of a working storage area into which record is to be moved (GET, GETBK); or of first data word of the logical record as it resides in working storage (PUT, PUTBK). (Optional).
<b>user-routine</b>	Error routine as given in subfield 9 in the routine fcb macro for the file on which the error occurred.

### File Preparation

CLOSE (list, n)	Close File
OPEN (list, n)	Open File
SETIN (fcb)	Set as Input File
SETOUT (fcb)	Set as Output File
.GCRE (fcb, isize, mode, istat, mflag)	Create Temporary File
.GTCH (fcb, catfil, iprmis, mode, istat, buffer, mflag, cflag)	Access a File
.GDTC (fcb, istat, buffer, mflag, cflag)	Deaccess a File

File Preparation

CLOSE (list, n)  
 OPEN (list, n)  
 SETIN (fcb)  
 SETOUT (fcb)

Close File  
 Open File  
 Set as Input File  
 Set as Output File

Logical Record Processing

COPY (fcb-out, fcb-in)  
 GET (fcb, eof, stor)  
 GETBK (fcb, eof, stor)  
 PUT (fcb, stor)  
 PUTBK (fcb, stor)  
 PUTSZ (fcb, size)  
 RELSE (fcb)

Write Logical Record From Input Buffer  
 Read Logical Record  
 Read Logical Record From Next Block  
 Write Logical Record  
 Write Logical Record in Next Block  
 Update File Control Block  
 Release Current Buffer

Device Positioning

BSREC (fcb, n, eof)  
 BSTFM (fcb, n)  
 FORCE (fcb)  
 FSREC (fcb, n, eof)

Backspace n Blocks (Physical Records)  
 Backspace to File Mark  
 Force End of Reel  
 Forward Space n blocks (Physical Records)

FSTFM (fcb, n)  
 REWND (fcb)  
 WEF (fcb, filemark)

Forward Space to File Mark  
 Rewind  
 Write End-of-File

Physical Record Processing

READ (fcb, cont, cc)  
 WAIT (fcb, eof)  
 WRITE (fcb, cont, cc)  
 POST (fcb, buffer, function)

Read Physical Record  
 Wait for Physical Record  
 Write Physical Record  
 Apply Delayed Posting

I/O Editor Functions

EPRINT (fcb, image, slew, n)  
 IOEDIT (list, n)  
 PRINT (fcb, image, slew)  
 PUNCH (fcb, image, kiind)  
 RDREC (fcb, eof, bcd)  
 WTREC (fcb, image, media, report)  
 .GBGET (fcb, eof, stor, media)  
 .GBGTB (fcb, stor, media)  
 .GBWTR (fcb, image, media, report)  
 BCDASC (from, to, count)  
 ASCBCD (from, to, count)  
 .GBTA. (from, to)  
 .GBNRY  
 .GBCD  
 .GATB. (from, to)  
 GMEDIA (fcb, media)

Edit and Write Print Line Image  
 I/O Editor Initialization  
 Write Print Line Image  
 Write Punch Card Image  
 Read Input Record  
 Write Output Record  
 Read Logical Record, Conforming  
 Read Logical Record from Next Block, Conforming  
 Write Output Record, Conforming  
 Convert BCD to ASCII  
 Convert ASCII to BCD  
 Convert BCD to ASCII Using Tallies  
 Convert BCD to Binary  
 Convert Binary to BCD  
 Convert ASCII to BCD Using Tallies  
 Media transiteration

Error Procedures

user-routine  
 (loc-of-fcb, code)

User Supplied Routine

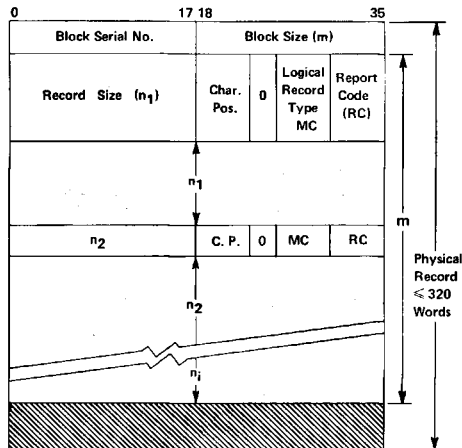
The types of devices that the I/O calls may handle are shown below. Restrictions that may affect the use of the calls are also shown.

Call	Mag. Tape	Paper Tape	Printer	Card Punch	Card Reader	Linked Mass Storage	Random Mass Storage
COPY	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
GET	1, 4	1, 4			1, 4	1, 3, 4, 5	
GETBK	1, 4	1, 4			1, 4	1, 3, 4, 5	
PUT	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
PUTBK	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
PUTSZ	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
RELSE	0	0	2	2	1	3, 4, 5	
BSREC	0					0	
BSTFM	8						
FORCE	0					0	
FSREC	0						
FSTFM	8, 9						
REWND	8					3, 4, 5	
WEF	2, 7					2, 3, 4, 5	
READ	1, 6	1, 6			1, 6		6
WRITE	2, 6	2, 6	2, 6	2, 6			6
PRINT	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
PUNCH	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	
RDREC	1, 4	1, 4			1, 4	1, 3, 4, 5	
WTREC	2, 4	2, 4	2, 4	2, 4		2, 3, 4, 5	

Codes

- 0 - Legal with no restrictions
- 1 - Must be input file
- 2 - Must be output file
- 3 - Must have variable record type
- 4 - Must have at least one buffer
- 5 - Must have block serial numbers
- 6 - Must not have buffers
- 7 - If FM = 17g, must not have labels
- 8 - Must not have labels
- 9 - Must be multifile reel

## STANDARD SYSTEM FORMAT



Block serial number (bits 0-17)	Sequential number of this physical record (within current reel if this file is on tape).
Block size (bits 18-35)	Size of block in words, not including control word.
Record size control word Record size (bits 0-17)	Binary equivalent of record size in words not including control word. When file is assigned to mass storage, value may be zero and word is interpreted as file mark analogous to tape end-of-file marker.
(bits 18-19)	Next available character position in last word: 00 - full word (four characters) used 01 - one character used 10 - two characters used 11 - three characters used
(bits 20-23)	Not used unless bits 0-17 are zero, in which case position contains specific file mark character.
(bits 24-25)	Zeros.
Media Code (bits 26-29)	Logical record type media code: 0 - BCD 1 - binary punch format 2 - BCD punch format 3 - BCD print 4 - user pre-empted 5 - ASCII files consistent with TSS 6 - ASCII standard system format 7 - reserved for user 8 - TSS information record
Report Code (bits 30-35)	Code identifying this record as one belonging to specific report or punch deck.

## MNE GESNAP SNAPSHOT DUMPS

The MME GESNAP is used to obtain partial memory printouts during the debugging of a program (except that it cannot be used during a courtesy call). The snapshot dump produced will be written on the system output file P\* if P\* is assigned to SYSOUT; otherwise it will be written to the execution report (J\*). The number of print lines produced by GESNAP is considered as part of SYSOUT print lines. Entry to this routine is made from the fault vector module as the result of the MME GESNAP. The calling sequence is:

```

Entry      L      MME GESNAP
          L+1    VFD 18/A, 2/P, 1/S, 15/N
          L+2    Return
    
```

where:

- A - Starting location of the snapshot. A may be symbolic. If A = modulo 8, N+7 is truncated to modulo 8. If A ≠ modulo 8, N+14 and A are truncated to modulo 8. If A is outside the program limits, only the panel will be dumped.
- S - Slew indicator: 0 = slew one line before SNAPSHOT dump. 1 = slew to top of page before snapshot dump.
- N - Number of words to be snapped.
- P - Panel indicator:  
00 = panel, EIS registers  
01 = no registers, memory only  
10 = panel and EIS registers only  
N > 0, registers will always be dumped.

To dump all of allocated memory, when allocated memory is greater than 32K:

```

L      MME GESNAP
L+1    36/0
L+2    Return
    
```

No slew will be made. With allocated memory less than 32K, a dump of all of allocated memory with a slew may be obtained by:

```

L      MME  GESNAP  or  L      MME  GESNAP
L+1    ZERO  ,377777  or  L+1    VFD  O36/377777
    
```

The calling sequence can be replaced by:

```

L      MME GESNAP      for multiple panels/page
L+1    OCT 400000
      or
      OCT 700000      for one panel/page
    
```

## OCTAL PATCHING

Corrections are made by the use of one or more octal correction cards. These are inserted in the object deck immediately preceding the \$DKEND card. The octal correction card has the following format:

### Cols.

- 1-6 Octal address — the contents of which are to be changed.  
 7 Blank.  
 8-12 The word OCTAL in BCD characters.  
 13-15 Blank.  
 16-72 This field contains one or more subfields as follows:

- A) If only one subfield is specified, it will replace the contents of the address specified in Cols. 1-6. Multiple subfields, which must be separated by commas, replace successive addresses starting with the one specified.
- B) A subfield may contain from zero to twelve octal digits. If less than twelve are specified, they will be right justified with leading zeroes inserted. A null field (,) constitutes one zero word.
- C) Each subfield may be prefixed and/or suffixed by the letter R. Prefix R will cause the high order 18 bits to be relocated. Suffix R results in the low order 18 bits being relocated. Absence of the letter R will imply absolute.

Following the terminal blank, comments may be added.

Octal corrections should not be made to SYMDEFs or SYMREFs. In particular, do not patch any instruction that has 030, 003, or 033 relocations bits in the GMAP listing. The General Loader needs the information in these words. If you patch these locations, the information will be destroyed and catastrophe will result.

It is also possible to make octal corrections to object programs on mass storage or tape libraries.

## SLAVE PROGRAM PREFIX

Note: Word numbers are shown below, first in decimal, and then (in parentheses) in octal.

**Fault Vector** — Each pair of the Fault Vector has the format: 1st Word: Bits 0-17 = IC, Bits 18-35 = IR. 2nd Word = Vector (If = 0, program aborts; if ≠ 0, registers are restored and control transferred to vector. Examples: to ignore fault, set to RET \*1. To process with routine at FAULT, set to TRA FAULT.)

- |       |         |   |
|-------|---------|---|
| 0-1   | (0-1)   | : Illegal Op Code, Command Faults.              |
| 2-3   | (2-3)   | : Memory Fault.                                 |
| 4-5   | (4-5)   | : Fault Tag Modifier.                           |
| 6-7   | (6-7)   | : Divide Check Fault.                           |
| 8-9   | (10-11) | : Overflow Fault.                               |
| 10    | (12)    | : Lockup Fault (always causes abort).           |
| 11    | (13)    | : Location of abort; reason code of last abort. |
| 12-13 | (14-15) | : Derail Instruction.                           |

### Used by GCOS

- |    |      |  |
|----|------|--|
| 14 | (16) | : Address to which control returned after checkpoint recovery.                   |
| 15 | (17) | : Bits 0-17 = Address of FCB in activity.<br>Bits 18-35 = No. FCB's in activity. |
| 16 | (20) | : File and Record Control switch word  |
| 17 | (21) | : GELBAR timer setting   |
| 18 | (22) | : IC&I for GELBAR  |
| 19 | (23) | : Fault vector for GELBAR  |

### General Loader

- |    |      |   |
|----|------|---|
| 20 | (24) | : Bits 0-17 = Address of entry to main program.<br>Bits 18 = 0 (normal highload), = 1 (lowload).<br>Bits 19-35 not used.  |
| 21 | (25) | : Bits 0-17 = Logical Unit Table address.<br>Bits 18-35 = Destination Code Table address.   |
| 22 | (26) | : Bits 0-16, pointer to Proc. History Reg. buffers, Bit 17, flag for RETRY. Bits 18-35 reserved for GCOS.   |
| 23 | (27) | : Bits 0-17 = Wrap-Up Routine address. Bit 18 = 0 (dump), = 1 (no dump).  |
| 24 | (30) | : Not used.   |
| 25 | (31) | : Bits 0-17, BAR from MME GELBAR. Bit 18, not used.<br>Bit 19, fault occurred in GELBAR. Bit 20, I/O interrupt in MME GELBAR. Bit 21, overflow. Bit 22, exponent overflow. Bit 23, exponent underflow. Bits 24-28, MBZ. Bit 29, divide check. Bits 30-35, fault type. |

### GCOS Initiation and Termination Sequence

- |       |         |   |
|-------|---------|---|
| 26-29 | (32-35) | : MME GECALL<br>BCI 1, GELOAD<br>ZERO<br>ZERO |
|-------|---------|---|

### Other Information

- |       |         |   |
|-------|---------|---|
| 30    | (36)    | : Bits 0-5 = Activity No.<br>Bits 6-35 = Job Sequence No.                             |
| 31    | (37)    | : Bits 0-17 = General Loader Lower Limit.<br>Bits 18-35 = General Loader Upper Limit. |
| 32-39 | (40-47) | : Safe-Store Area #1 for slave program registers.                                     |
| 40-47 | (50-57) | : Safe-Store Area #2 for main level registers.  |
| 48    | (60)    | : EOF Reissue buffer.   |
| 49-53 | (61-65) | : Reserved for GCOS   |
| 54-63 | (66-77) | : Image of columns 16-72 of \$ IDENT card.  |



## COMPRESSED DECK FORMAT

The compressed deck (COMDK) format is produced by removing sequences of 3 or more blanks from Hollerith coded card images, and packing the information in standard column binary form. To do this, a Hollerith card is considered as being composed of a series of fields (a segment of the card containing no sequences of over 2 consecutive blanks, except at the beginning), and strings (that portion of a field obtained by deleting leading blanks).

Each field specification consists of: first, an octal count,  $A(10 \leq A \leq 67_8)$ , of the total characters in the field; second, an octal count,  $B(0 \leq B \leq 67_8)$ , of the number of characters in the string; third, the actual characters that constitute the string.

If a field exceeds  $67_8$ , it is segmented into separate fields; e.g.,  $70_{10}$  consecutive nonblank characters are treated as two fields:

Field 1 A =  $67_8$ , B =  $67_8$  (octal values)

Field 2 A =  $17_8$ , B =  $17_8$  (octal values)

The field specifications (A, B, string) are packed sequentially on a binary card (see format below). A field specification may be started on a COMDK card (X) and completed on the following card (X+1); but in doing so, the field is split into two parts. The first two characters of card X+1 will be the A and B characters for the second part.

The following codes for A are used to designate specific conditions. The B character is not present in such cases.

A = 0 End compressed card. Continue decoding on next card.  
 A =  $77_8$  End encoded string for given Hollerith card image.  
 A =  $76_8$  End of the compressed deck segment.  
 A =  $70_8 - 75_8$  Available for extension.

### COMDK Card Layout

Word 1: 0-2 = Column binary card type 5.

3-8 = Zeros.

9-11 = 101 (7-9 punches). 12-35 = Binary sequence No.

Word 2: Checksum of word 1 and words 3-24.

Words 3-24: Compressed card image.

Words 25-27: Hollerith label or zeros (supplied by I/O Editor).

Examples (all numbers are octal; ° = blank)

1. COMDK °°°ABC° is encoded as: 0505COMDK0703ABC77

2. °°°°°LDAQ°°°°GROSS°°°° is 1204LDAQ1105GROSS77

## STANDARD CHARACTER SET

Character	Internal	Octal	Card	ASCII	UASCI	EBCDIC
0	000000	00	0	060	060	360
1	000001	01	1	061	061	361
2	000010	02	2	062	062	362
3	000011	03	3	063	063	363
4	000100	04	4	064	064	364
5	000101	05	5	065	065	365
6	000110	06	6	066	066	366
7	000111	07	7	067	067	367
8	001000	10	8	070	070	370
9	001001	11	9	071	071	371
[	001010	12	2-8	133	133	
#	001011	13	3-8	043	043	173
@	001100	14	4-8	100	100	174
:	001101	15	5-8	072	072	172
>	001110	16	6-8	076	076	156
?	001111	17	7-8	077	077	157
␣	010000	20	(blank)	040	040	100
A	010001	21	12-1	141	101	301
B	010010	22	12-2	142	102	302
C	010011	23	12-3	143	103	303
D	010100	24	12-4	144	104	304
E	010101	25	12-5	145	105	305
F	010110	26	12-6	146	106	306
G	010111	27	12-7	147	107	307
H	011000	30	12-8	150	110	310
J	011001	31	12-9	151	111	311
&	011010	32	12	046	046	
.	011011	33	12-3-8	056	056	113
]	011100	34	12-4-8	135	135	
(	011101	35	12-5-8	050	050	115
<	011110	36	12-6-8	074	074	114
\	011111	37	12-7-8	134	134	
†	100000	40	11-0	136	136	
J	100001	41	11-1	152	112	321
K	100010	42	11-2	153	113	322
L	100011	43	11-3	154	114	323
M	100100	44	11-4	155	115	324
N	100101	45	11-5	156	116	325
O	100110	46	11-6	157	117	326
P	100111	47	11-7	160	120	327
Q	101000	50	11-8	161	121	330
R	101001	51	11-9	162	122	331
.	101010	52	11	055	055	140
\$	101011	53	11-3-8	044	044	133
*	101100	54	11-4-8	052	052	134
)	101101	55	11-5-8	051	051	135
;	101110	56	11-6-8	073	073	136
'	101111	57	11-7-8	047	047	175
+	110000	60	12-0	053	053	116
/	110001	61	0-1	057	057	141
S	110010	62	0-2	163	123	342
T	110011	63	0-3	164	124	343
U	110100	64	0-4	165	125	344
V	110101	65	0-5	166	126	345
W	110110	66	0-6	167	127	346
X	110111	67	0-7	170	130	347
Y	111000	70	0-8	171	131	350
Z	111001	71	0-9	172	132	351
←	111010	72	0-2-8	137	137	
^	111011	73	0-3-8	054	054	153
%	111100	74	0-4-8	045	045	154
=	111101	75	0-5-8	075	075	178
"	111110	76	0-6-8	042	042	177
!	111111	77	0-7-8	041	041	132

## STATE WORD

### BITS

0	= 1,	In execution
1	= 1,	Program number in queue
2	= 1,	Program loading from SSA
3	= 1,	Exception Processing in control
4	= 1,	Exception Processing needed
5	= 1,	Abort in control
6	= 1,	Abort request received
7	= 1,	Swap in control
8	= 1,	Swap requested
9	= 1,	Program in courtesy call
10	= 1,	At least one courtesy call waiting
11	= 1,	Waiting for core store size change (program dead)
12	= 1,	YSOOUT writing
13	= 1,	Relinquished
14	= 1,	Roadblock
15-17	= 1,	Abort substate (15 = 1, terminate job)
	= 00,	FINI
	= 01,	FAULT
	= 10,	BORT
	= 11,	KILL/TERM
18	= 1,	MME .EMM allowed
19		Reserved for GCOS
20		Reserved for GCOS
21	= 1,	Do not set abort bit on Exception Processing — system programs
22	= 1,	Wrapup done
23	= 1,	Enable request
24	= 1,	Gated module busy for program
25	= 1,	Swap this
26	= 1,	Do not move or swap program
27	= 1,	An I/O has completed since last link
28	= 1,	Alarm set for program
29		Reserved for GCOS
30	= 1,	MME GECHek done
31	= 1,	MME GEIDSE done
32	= 1,	No. 3 processor cannot execute this
33	= 1,	No. 2 processor cannot execute this
34	= 1,	No. 1 processor cannot execute this
35	= 1,	No. 0 processor cannot execute this

## STAT1 WORD

### BITS

0	= 1,	Compress next activity
1	= 1,	A disaster fault has occurred
2	= 1,	MME GELBAR in effect
3	= 1,	I/O has terminated for job in termination
4-17		Reserved for GCOS
18-35		Address of first .ENTRY word in threaded list of all modules specially loaded within the slave area itself

## SLAVE SERVICE AREA SYMBOL LOCATIONS

.SACT	777057	.SPR01	777142
.SACT	777065	.SPRT	777061
.SACTY	777066	.SPTBE	777071
.SALIM	777053	.SPUSH	777153
.SALT	777774	.SREG	777000
.SAVG	777064	.SREGS	777050
.SCCAL	777141	.SREMT	777140
.SCKSM	776014	.SRQCT	777143
.SCONT	777060	.SSA	776000
.SFSYS	777211	.SSAPA	777752
.SGCPA	777745	.SSAPT	777733
.SGCPT	777731	.SSSAL	777055
.SGEPR	777043	.SSAK	777045
.SGNPA	777765	.SSTRT	777154
.SGNPT	777734	.SSYIO	777212
.SICI	777144	.SSYOT	777073
.SIOQ	777225	.START	777063
.SLOAD	777041	.STAT1	777067
.SLPRT	777152	.STATE	777040
.SLTT	777151	.STCHT	777066
.SMDSK	777203	.STEMP	777156
.SMSZ	777137	.STOTY	777725
.SNIO	777072	.STPPA	777735
.SNPAT	777070	.STPPT	777730
.GNTRY	776015	.SUID	777776
.SPATP	777724	.SURG	777202
.SPDPA	777757	.SWIT	777773
.SPDPT	777732		

## PROGRAM SWITCH WORD

0	4 5 6	17 18	35
System Options	System Options	User Switches	

Bit	Bit = 1	Bit = 0
0	DUMP Option	NDUMP Option
1	FORTY or GMAP	Neither FORTY nor GMAP
2	SYMTAB/XREF	NSYMTAB/NXREF
3	PURGE implicit file	NPURGE
4	CLEAR core	NCLEAR
5	Execution activity	No execution activity
6	COMDK/ON1	NCOMDK
7	DECK/ON2	NDECK
8	LSTOU/ON3	NLSTOU
9	UPDATE/ON4	No UPDATE
10	LSTIN/ON5	NLSTIN
11	DEBUG, NGMAC, STAB/ON6	NSTAB, GMAC, NDEBUEG
12	(Bit 1 = 0) Abort Subactivity (Bit 1 = 1) MAP	(Bit 1 = 0) No Abort Subactivity (Bit 1 = 1) NOMAP
13	GMAP/System Editor Interface (Bit 1 = 0) COPY/FORTRAN or COBOL created G* File (Bit 1 = 1) FORM	No GMAP Interface (Bit 1 = 0) NCOPY/standard G* (Bit 1 = 1) NFORM
15	(Bit 1 = 0) GESAVE/PRODUCT (Bit 1 = 1) LNO	(Bit 1 = 0) No GESAVE/PRODUCT (Bit 1 = 1) NLNO
16	(Bit 1 = 0) EXTEND (Bit 1 = 1) BCD	(Bit 1 = 0) No EXTEND (Bit 1 = 1) ASCII
17	(Bit 1 = 0) PRODUCT (Bit 1 = 1) OPTZ	(Bit 1 = 0) No PRODUCT (Bit 1 = 1) NOPTZ
18	For programmer use. Switches set by MME GESETS or MME GEREST	
35		

## OCTAL-DECIMAL CONVERSION

Octal	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

## OCTAL-DECIMAL CONVERSION (Continued)

Octal	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

POWERS OF TWO

POSITIVE POWERS OF TWO

n	2 <sup>n</sup>
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576
21	2097152
22	4194304
23	8388608
24	16777216
25	33554432
26	67108864
27	134217728
28	268435456
29	536870912
30	1073741824
31	2147483648
32	4294967296
33	8589934592
34	17179869184
35	34359738368
36	68719476736
37	137438953472
38	274877906944
39	549755813888
40	1099511627776
41	2199023255552
42	4398046511104
43	8796093022208
44	17592180444416
45	35184360888832
46	70368721777664
47	140737443555328
48	281474887110656
49	562949774221312
50	1125899548442624
51	2251799096885248
52	4503598193770496
53	9007196387540992
54	18014392775081984
55	36028785550163968
56	72057571100327936
57	144115142200655872
58	288230284401311744
59	576460568802623488
60	1152921137605246976
61	2305842753210493952
62	4611685506420987904
63	9223371012841975808
64	18446742025683951616
65	36893484513767903232
66	73786969027535806464
67	147573938055071612928
68	295147876110143225856
69	590295752220286451712
70	1180591504440572903424
71	2361183008881145806848
72	4722366017762291613696

NEGATIVE POWERS OF TWO

n	2 <sup>n</sup>
0	1.0
1	0.5
2	0.25
3	0.125
4	0.0625
5	0.03125
6	0.015625
7	0.0078125
8	0.00390625
9	0.001953125
10	0.0009765625
11	0.00048828125
12	0.000244140625
13	0.0001220703125
14	0.00006103515625
15	0.000030517578125
16	0.0000152587890625
17	0.00000762939453125
18	0.000003814697265625
19	0.0000019073486328125
20	0.00000095367431640625
21	0.000000476837158203125
22	0.0000002384185791015625
23	0.00000011920928955078125
24	0.000000059604644775390625
25	0.0000000298023223876953125
26	0.00000001490116119384765625
27	0.000000007450580596923828125
28	0.0000000037252902884619140625
29	0.00000000186264514923095703125
30	0.000000000931322574615478815625
31	0.0000000004656612873077392578125
32	0.000000000232830643652896625
33	0.0000000001164153218269348453125
34	0.0000000000582076609134674072265625
35	0.00000000002910383045673370361328125
36	0.000000000014551915228366851806640625
37	0.0000000000072759576141834259033203125
38	0.0000000000036379789070917295188015625
39	0.000000000001818989403545856475830078125
40	0.0000000000009094947017729282379150390625
41	0.00000000000045474735088646411895751953125
42	0.000000000000227373675443232059478759765625
43	0.0000000000001136868377216160297393798828125
44	0.000000000000056843418868080148968984140625
45	0.00000000000002842170943040400743484970703125
46	0.0000000000000142108547152020037174224853515625
47	0.00000000000000710542735760100185871124287578125
48	0.0000000000000035527136788005092935621337890625
49	0.000000000000001776357763568394046776106689453125
50	0.00000000000000088817841970012523233890533447265625

NOTES

