6180 DEBUGGERS HANDBOOK

JUNE 1972

CAMBRIDGE INFORMATION SYSTEMS LABORATORY

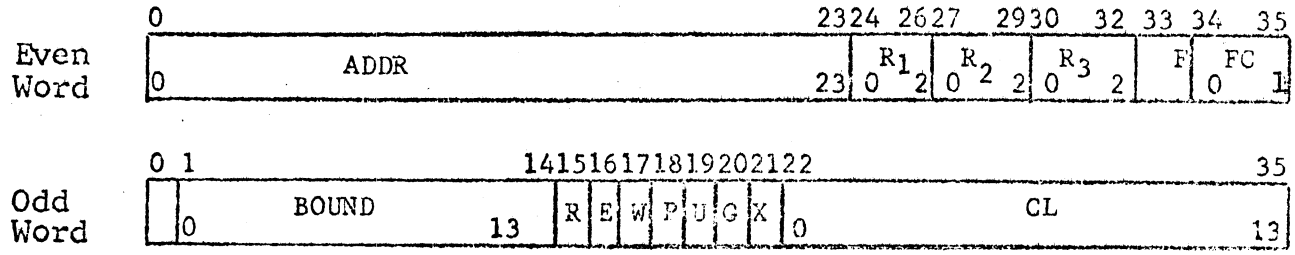
P R E L I M I N A R Y C O P Y

REVISION NUMBER 2.

EDITOR
D.R. Vinograd

TABLE OF CONTENTS

SDW Format



where:

ADDR — is the 24 bit segment address.

R_1, R_2, R_3 — is highest effective ring number of the read/write, the read/execute, and the call brackets respectively for this segment.

F — directed fault indicator.

1 - the necessary unpaged segment or PTW currently resides in memory.

0 - execute the directed fault specified in FC.

FC — indicates (if F = 0) which of the four directed faults is to be executed (DF0-DF3).

BOUND — the highest mod 16 computed address which can be generated without causing an out of bounds fault.

R — read permission bit.

E — execute permission bit (XEC, XED excluded).

W — write permission bit.

P — privileged mode bit.

0 - privileged instructions cannot be executed and interrupts cannot be inhibited via bit 28.

1 - if this procedure segment is executed in ring 0, it can execute privileged instructions and inhibit interrupts via bit 28.

U — the unpaged/paged bit.

0 - this segment is paged, $SDW.ADDR_{00-23}$ is the base address of the segment page table.

1 - this segment is unpaged, $SDW.ADDR_{00-23}$ is the base address of the segment. (Bit 20-23 are assumed = 0).

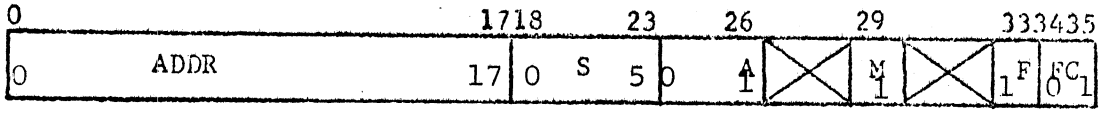
G — the gate indicator.

0 - any call to this segment from a different segment must be directed to a computed address less than the value of the CL field.

1 - any legal segment address may be called.

CL — is the largest computed address +1 in this segment that can be called by another segment if G = 0.

PTW Format



where:

ADDR — is the address of a page. The hardware ignores certain lower order bits of ADDR in order to generate the proper hard-wired page size according to the following table.

Page Size (Words)	ADDR Bits Ignored
64	
128	17
256	16-17
512	15-17
1024	14-17
2048	13-17
4096	12-17

S — is page status information.

A — is the page accessed bit.

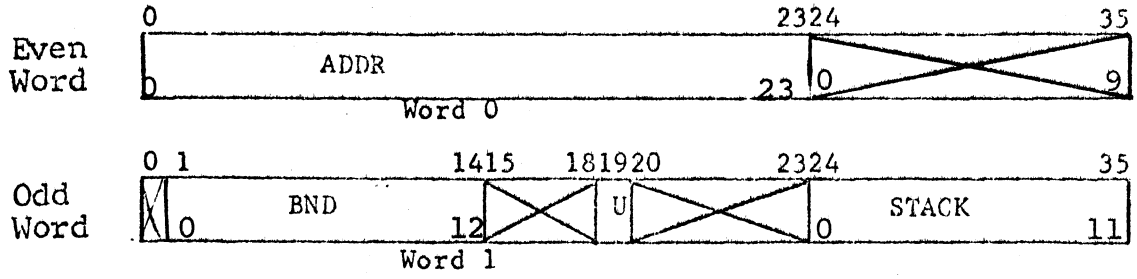
M — is the page modified bit.

F — is the directed fault indicator.

FC — indicates (if F = 0) which of the four directed faults is to be executed (DF0-DF3).

The Descriptor Segment Base Register (DSBR)

The DSBR assembly specifies the descriptor segment for a process in execution. The format of a pair of words in memory which may be used to load the DSBR assembly is:



where:

DSBR.ADDR is the address of description segment (bit 23 assumed = 0)

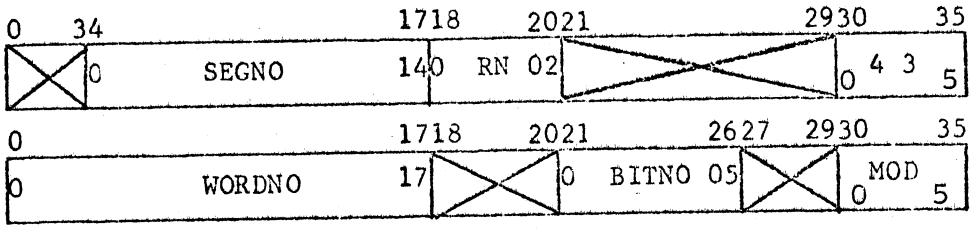
DSBR.BND 01-14 this register is the highest 0 mod 16 descriptor segment address which can be accessed without causing an out of bounds fault

DSBR.U this register (1 bit) specifies whether or not the descriptor segment for a process is unpaged (DSBR.U=1) or paged (DSBR.U=0).

DSBR.STACK this register is the upper 12 bits of the 15-bit stack segment number. This register is used only during the execution of the call instruction.

The ITS Modifier

The format of the ITS word pair in memory shall be:



where: SEGNO = Indirect Segment Number

RN = Maximum level of privilege which can be assigned to the current procedure segment for accessing the specified segment.

43_8 = ITS designator

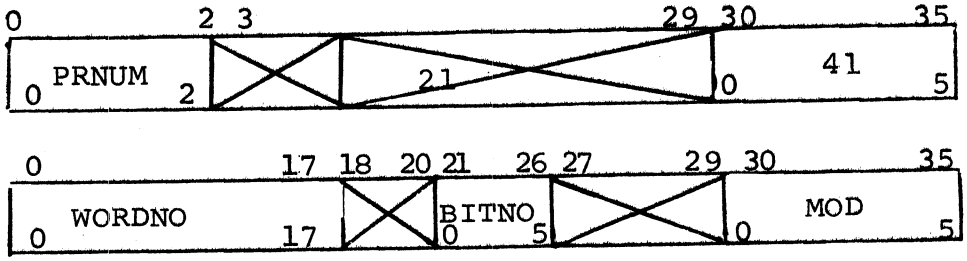
WORDNO = Word offset to be used in generating CA for the indirect segment access.

BITNO = Bit offset

MOD = Is any valid modifier.

ITP Modifier

In the 645F Processor the indirect word for ITP modification shall consist of the following format:



PRNUM (0-2) = the number of the pointer register to use in the indirection,

41_8 = ITP designator,

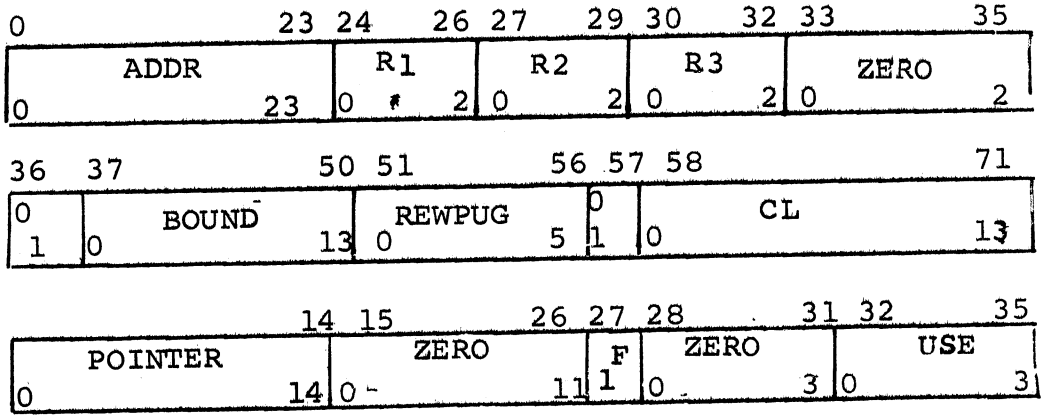
WORDNO (0-17) = Word offset to be used in the effective address formation,

BITNO (21-26) = Bit offset,

MOD (30-35) = any valid modifier.

The SDW Associative Memory Assemblies (SDWAM)

The format of each of the 16 SDWAM registers shall be:



where:

ADDR, R1, R2, R3, BOUND, R,E,W,P,U,G and CL are the contents of the corresponding fields of an SDW fetched from main memory

POINTER: is the effective segment number generated when this SDW was fetched from main memory.

USE: is the relative usage of this segment in relation to the other 15 SDW registers, where 15_{10} (1111) is the most recently referenced. If USE = 0000, this register shall be used for storage of the next SDW fetched from main memory.

F is the full/empty bit.

0 - this register does not contain valid information.

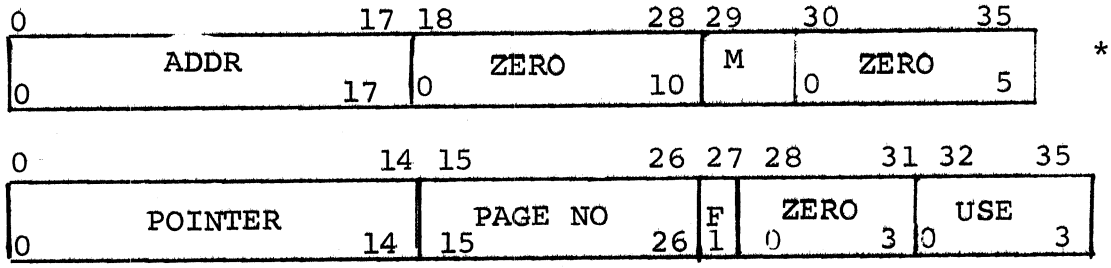
1 - this register contains valid information.
an SDW.

* Format as stored by SSDR.

** Format as stored by SSDP

The PTW Associative Memory Assemblies (PTWAM)

The format of each of the 16 PTWAM registers shall be:



where:

ADDR and M are the contents of the corresponding

POINTER, USE, and F have the same meaning as the associative memory SDW

PAGE is the page number to which this PTW refers. Certain bits of this field shall be forced to zero depending upon the designated page size

Page Size	Bits of Page Forced to Zero
64	None
128	11
256	10-11
512	09-11
1024	08-11
2048	07-11
4096	06-11

* Format as stored by SPTR

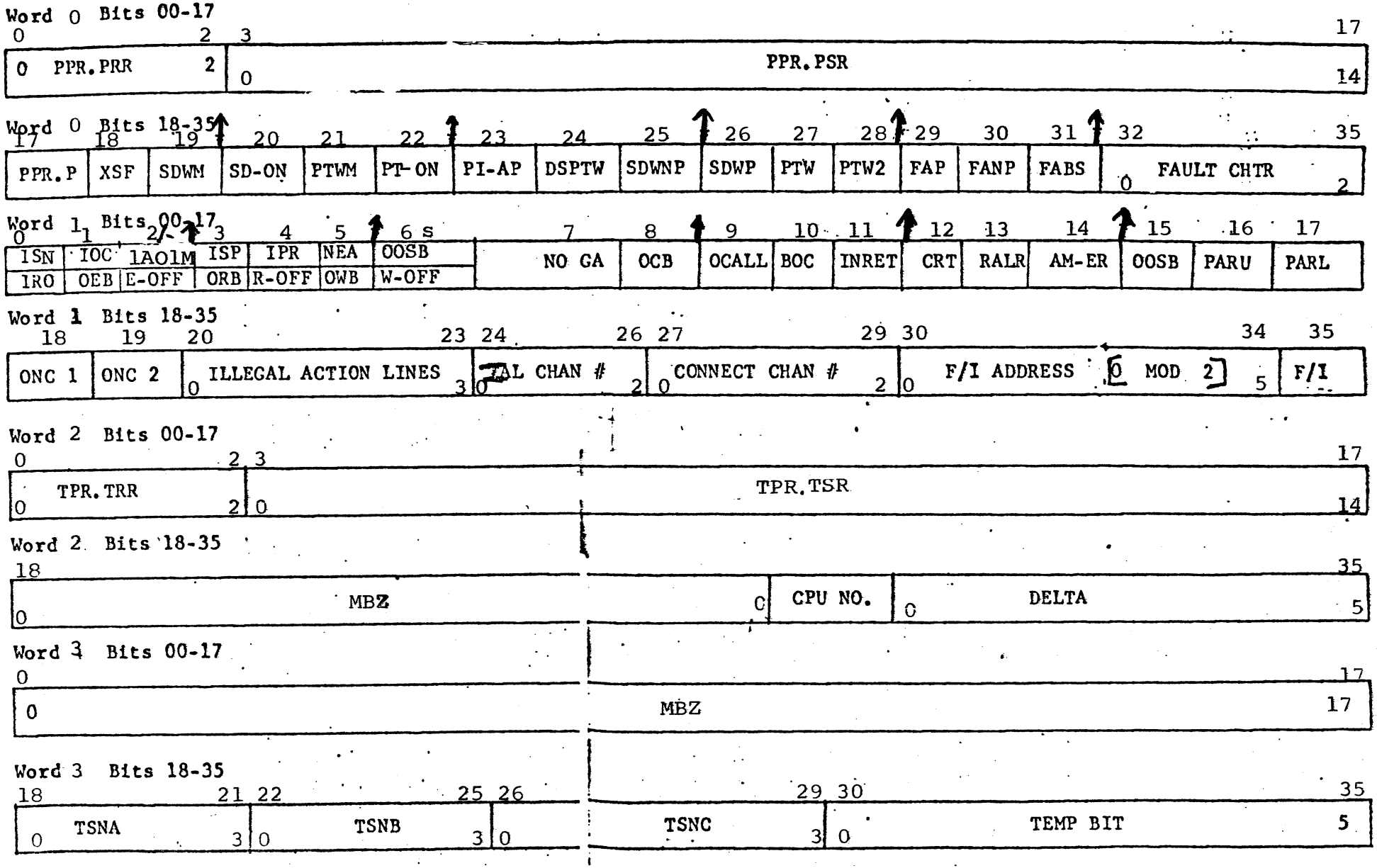
** Format as stored by SPTP

FAULTS

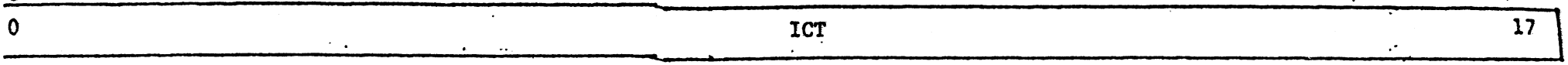
OCTAL NUMBER	DECIMAL NUMBER	NAME	MNEMONIC	PRIORITY	GROUP	MODE
0	0	Shutdown	SDF	27	VII	Multics/GCOS
1	1	Store	STR	10	IV	Multics/GCOS
2	2	Master Mode Entry 1	MME1	11	17	Multics/GCOS
3	3	Fault Tag 1	FTG1	17	V	Multics/GCOS
4	4	Timer Runout	TROF	26	VII	Multics/GCOS
5	5	Command	FCMD	9	IV	Multics/GCOS
6	6	Derail	DRL	15	V	Multics/GCOS
7	7	Lockup	LUF	5	IV	Multics/GCOS
10	8	Connect	CON	25	VII	Multics/GCOS
11	9	Parity	FPAR	8	IV	Multics/GCOS
12	10	Illegal Procedure	IPR	16	V	Multics/GCOS
13	11	Op Not Complete	FONC	4	II	Multics/GCOS
14	12	Startup	SUF	1	1	Multics/GCOS
15	13	Overflow	FOFL	7	III	Multics/GCOS
16	14	Divide Check	FDIV	6	III	Multics/GCOS
17	15	Execute	EXF	2	I	Multics/GCOS
20	16	Directed Fault 0	DFT0	20	VI	Multics
21	17	Directed Fault 1	DFT1	21	VI	Multics
22	18	Directed Fault 2	DFT2	22	VI	Multics
23	19	Directed Fault 3	DFT3	23	VI	Multics
24	20	Access Violation	ACV	24	VI	Multics

OCTAL NUMBER	DECIMAL NUMBER	NAME	MNEMONIC	PRIORITY	GROUP	MODE
25	21	Master Mode Entry 2	MME2	12	V	Multics
26	22	Master Mode Entry 3	MME3	13	V	Multics
27	23	Master Mode Entry 4	MME4	14	V	Multics
30	24	Fault Tag 2	FTG2	18	V	Multics
31	25	Fault Tag 3	FTG3	19	V	Multics
32	26	Unassigned				
33	27	Unassigned				
34	28	Unassigned				
35	29	Unassigned				
36	30	Unassigned				
37	31	Trouble	TRB	3	II	Multics

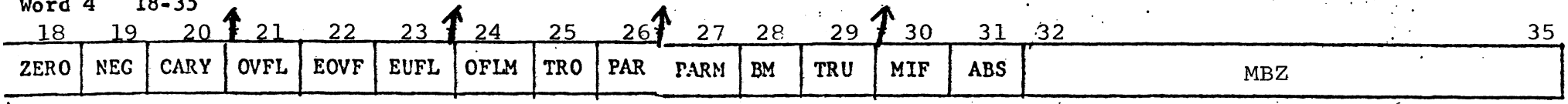
SCU FORMAT


Word 0 Bits 00-17
0 2 3 17
0 PPR.PRR 2
0 PPR.PSR 14
Word 0 Bits 18-35
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 35
PPR.P
XSF
SDWM
SD-ON
PTWM
PT-ON
PI-AP
DSPTW
SDWNP
SDWP
PTW
PTW2
FAP
FANP
FABS
0 FAULT CHTR 2
Word 1 Bits 00-17
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
ISN
IOC
1AO1M
ISP
IPR
NEA
OOSB
IRO
OEB
E-OFF
ORB
R-OFF
OWB
W-OFF
NO GA
OCB
OCALL
BOC
INRET
CRT
RALR
AM-ER
OOSB
PARU
PARL
Word 1 Bits 18-35
18 19 20 23 24 26 27 29 30 34 35
ONC 1
ONC 2
0 ILLEGAL ACTION LINES 3
0 CHAL CHAN # 2
0 CONNECT CHAN # 2
0 F/I ADDRESS 0 MOD 2 5
F/I
Word 2 Bits 00-17
0 2 3 17
0 TPR.TRR 2
0 TPR.TSR 14
Word 2 Bits 18-35
18 35
0 MBZ 0
CPU NO.
0 DELTA 5
Word 3 Bits 00-17
0 17
0 MBZ 17
Word 3 Bits 18-35
18 21 22 25 26 29 30 35
0 TSNA 3
0 TSNB 3
0
TSNC 3
0 TEMP BIT 5

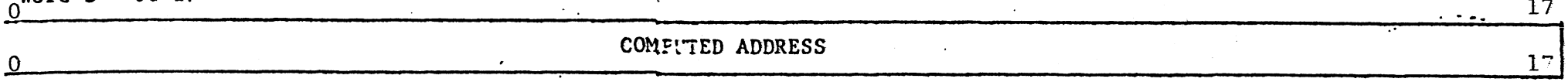
Word 4 00-17



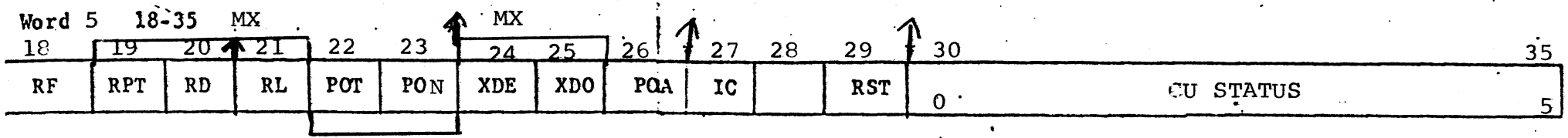
Word 4 18-35



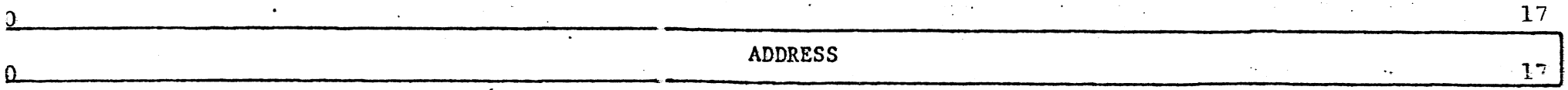
Word 5 00-17



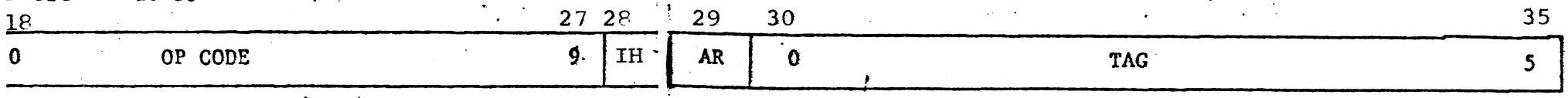
Word 5 18-35



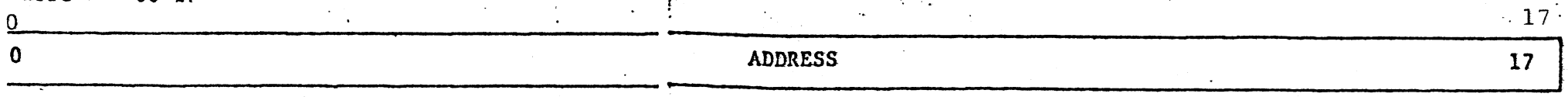
Word 6 00-17



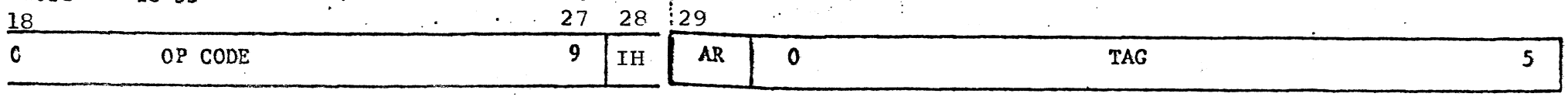
Word 6 18-35



Word 7 00-17



Word 7 18-35



APU STATUS $C(Y)_{19-29}$:

Bit		Mnemonic
18	Privileged Bit	
19	External Segment Flag	XSF
20	SDW Match	SDWM
21	SDWAM-ON	SD-ON
22	PTW Match	PTWM
23	PTWAM-ON	PT-ON
24	Instr. Fetch Append Cycle	PI-AP
25	Fetch DSPTW	DSPTW
26	Fetch SDWNP	SDWNP
27	Fetch SDWP	SDWP
28	Fetch PTW	PTW
29	Fetch Prepage PTW	PTW2
30	Fetch Final Address Paged	FAP
31	Fetch Final Address Nonpaged	FANP
32	Fetch Final Address Absolute	FABS

FAULT CNTR $C(Y)_{33-35}$:

Indicates the number of unsuccessful attempts to execute an instruction.

FAULT WORD $(Y+1)_{0-35}$

Bit	Fault	Mnemonic
0	Illegal Ring Order	
1	Not in execute bracket	ACV
2	Execute bits is off	
3	Not in read bracket	
4	Read bit is off	
5	Not in write bracket	
6	Write bit is off	
7	Not a gate	
8	Not in call bracket	
9	Outward call	
10	Bad outward call	
11	Inward return	
12	Cross ring transfer	
13	Ring alarm	
14	Associative memory	
15	Out of segment bounds	ACV
16	Processor parity error upper	FPAR
17	Processor parity error lower	FPAR
18	SC to Proc. Seq. Error #1	FONC
19	SC to Proc. Seq. Error #2	FONC
20, 21, 22, 23	Illegal Action Lines	Illegal actions
24, 25, 26	Illegal Action Channel	
27, 28, 29	Connect Channel	CON, CON, CON
30, 31, 32, 33	Fault Number/Address	
34		
35	Fault/Interrupt	

Group I

Bit	Fault	Mnemonic
0	Illegal Segment Number	STR
1	Illegal Op Code	IPR
2	Illegal Address and Modifier	IPR
3	Illegal Slave Procedure	IPR
4	All Other Illegal Procedure	IPR
5	Nonexistent Address	STR
6	Out of Bounds	STR

Group II

PORT STATUS $C(Y+1)_{20-35}$:

SYSTEM CONTROLLER ILLEGAL ACTION LINES C(Y+1) 20-23

IA Code (Octal)	Illegal Action	System Controller Priority	Resulting Processor Fault
03	Fault on condition	1	FCMD
14	ZAC parity, proc. to S.C.	2	FPAR
12	Illegal Command	3	FCMD
10	Not control	4	FCMD
02	Non-existent address	5	STR
15	Data parity, proc. to S.C.	6	FPAR
13	Store not ready	7	STR
16	ZAC parity, S.C. to store	8	FPAR
17	Data parity, S.C. to store	9	FPAR
07	Data parity store to S.C. & in store	10	FPAR
06	Data parity in store	11	FPAR
05	Data parity, store to S.C.	12	FPAR
11	Port not enabled	13	FCMD
00	None	14	None

Note: Illegal Action Codes (octal) 01 and 04 are not assigned. The occurrence of an unassigned code will result in an FCMD fault.

Illegal Action Channel Number C(Y+1) 24-26

Indicates CPU Port Number which received illegal action lines.

Illegal Action Channel Number $C(Y+1)_{24-26}$:

Indicates CPU Port Number which received illegal action lines.

Connect Channel Number $C(Y+1)_{27-29}$:

Indicates CPU Port Number which initiated connect fault.

F/I Address $C(Y+1)_{30-34}$:

Indicates relative address (0 mod 2) of fault or interrupt vector.

F/I $C(Y+1)_{35}$:

1 Indicates Fault Occurred

0 Indicates External Interrupt Occurred

TSR STATUS $C(Y+3)_{18-24}$:

TSNA $C(Y+3)_{18-21}$:

Temporary printer register number for non-multiword instructions or for operand descriptor 1 of multiword instructions where:

TSN_{0-2} designates the pointer register number

TSN_3 specifies (if = 1) that TSN_{0-2} is to be used

TSNB $C(Y+3)_{22-25}$:

Temporary pointer register number for operand descriptor 2 of multiword instructions.

TSNC $C(Y+3)_{26-29}$:

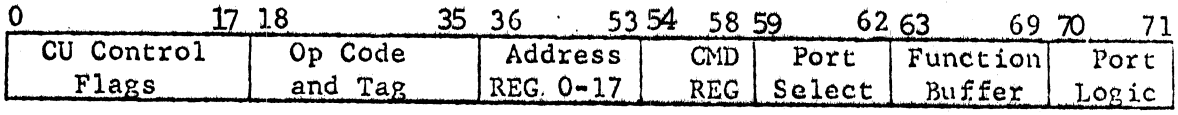
Temporary pointer register number for operand descriptor 3 of multiword instructions.

CU STATUS $C(Y+6)_{18-29}$

BIT		MNEMONIC	
18	Repeat-First Cycle	RF	
19	Repeat	RPT	
20	Repeat Double	RD	*
21	Repeat Link	RL	
22	IT Modification	POT	*
23	Return Type Instruction	PON	
24	Execute Even	XDE	*
25	Execute Odd	XDO	
26	Operand Preparation	POA	
27	ILC Even/Odd	IC	
28			
29	Restart Instruction from Beginning	RST	

* BIts in these fields are mutually exclusive.

CU HISTORY REGISTER FORMAT



where:

00-17 CU Control Flags

Tells what the CU is presently doing.

Bit	Flag	A "1" Means:
0	PIA	Preparing Instruction Address
1	POA	Preparing Operand Address
2	RIW	Requesting Indirect Word
3	SIW	Restoring Indirect Word
4	POT	Preparing Operand Tally
5	PON	Preparing Operand Address - Ignore further indirection
6	RAW	Requesting Alter Rewrite Word
7	SAW	Restoring Alter Rewrite Word
8	TRGO	Transfer Go (conditions met)
9	XDE	Doing an XED from an Even Location
10	XDO	Doing an XED from an Odd Location
11	IC	Executing the Odd location of an instruction pair
12	RPTS	Doing a Repeat Operation

CU HISTORY REGISTER (Continued)

Bit	Flag	A "1" Means:
13	WI	Waiting for (fetching) an Instruction
14	AR F/E	Address Register contains valid information
15	$\overline{\text{XIP*ADR/ZFA}}$	NOT preparing an XIP Address
16	$\overline{\text{FLT*ADR/ZFA}}$	NOT preparing an FLT Address
17	$\overline{\text{ADD BASE}}$	NOT in slave mode

18-35 Op Code and Tag

A copy of bits 18-35 of the Instruction being executed.

36-53 Address Register

A copy of bits 00-17 of the Address Register.

54-58 Command Register

A copy of the Processor Command Register

59-62 Port Select

A copy of the Port Select Lines

63-69 Function Buffer

Tells what the CU has done and the port operation that is starting.

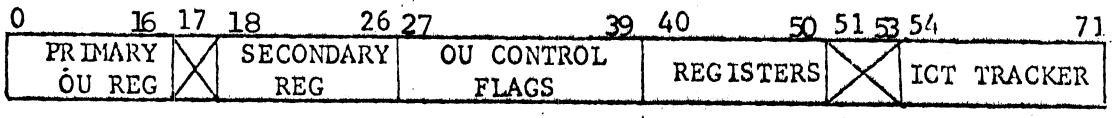
Bit	Function	A "1" Means:
63	XEC-INT	An execute interrupt
64	INS-FETCH	An instruction fetch
65	CU-STORE	Load or store on behalf of the CU or OU
66	OU-STORE	
67	CU-LOAD	
68	OU-LOAD	
69	DIRECT	A direct cycle

70-71 Port Logic

Tells the status of the port logic

70	$\overline{\text{PC BUSY}}$	Port control logic not busy
71	Port BUSY	Port interface busy.

OU HISTORY REGISTER FORMAT



where:

00-16 Primary OU Register (RP)

Holds next instruction. Op code and sets up input data switches.

Bits	Information:
0-8	Bits 18-26 of the next instruction to be executed (Op Code)
9	Bit 30 of IT tally word (6/9 bit data)
10-12	Bits 33-35 of the next instruction to be executed (either latter part of the tag or the character #)
13	Indicates character modification (IT)
14	Indicates direct operation (DL = 10·14, DU = $\overline{10}$·14)
15,16	LREG/SREG Effective Address Counter
17	Spare
18-26	Secondary OU Register (RS)

A copy of bits 0-8 of the RP register. It is strobed when the input data is strobed into the OU. It is used to do the actual instruction execution.

27-39 OU Control Flags

It tells the state of the RB1, RP, & RS registers and tells what OU cycle has been executed.

Bit	Function	A "1" Means:
27	FRB1-FULL	OU OPCODE buffer Full
28	FRP-FULL	Primary Register Full
29	FRS-FULL	Secondary Register Full
30	FGIN	First Cycle for all OU OPs
31	FGOS	Second Cycle for OU Multiple OPs
32	FGD1	First Divide Cycle
33	FGD2	Second Divide Cycle
34	FGOE	Exponent Compare Cycle

OU HISTORY REGISTER (Continued)

Bit	Function	A "1" Means:
35	FG0A	Mantissa Alignment Cycle
36	FG0M	General OU Cycle
37	FG0N	Normalize Cycle
38	FG0F	Final Cycle
39	FSTR-OP-AV	OU store data available (reset by CU)

40-50 Registers

Tell which registers are NOT being used.

Bit	Register
40	DATA NOT available
41	A reg NOT in use
42	Q reg NOT in use
43	X0 reg NOT in use
44	X1 reg NOT in use
45	X2 reg NOT in use
46	X3 reg NOT in use
47	X4 reg NOT in use
48	X5 reg NOT in use
49	X6 reg NOT in use
50	X7 reg NOT in use

51-53 Spare

54-71 ICT Tracker

Tells the address of the OU instruction (skips CU instructions and therefore can point further back than 16 instructions).

DU HISTORY REGISTER FORMAT

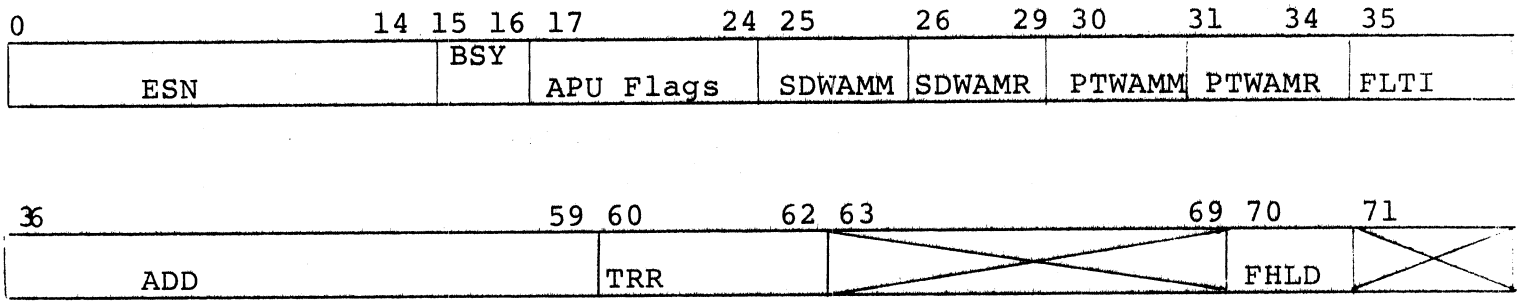
Bit	Mnemonic	Function
0	FPØL	Preparing Operand Length*
1	FPØP	Preparing Pointer*
2	¢NEED-DESC	Need Descriptor*
3	¢SEL-ADR	Select Address Register*
4	DLEN=DIRECT	Length equals direct (length=0; Force direct)*
5	DFRST	Descriptor is being processed for first time*
6	FEXR	Extended Register Modification*
7	DLAST-FRST	Last cycle of DFRST*
8	DDU-LDEA	Decimal Unit Load and Effective Address*
9	DDU-STEA	Decimal Unit Store and Effective Address*
10	DREDØ	Redo (Inhibits update of Pointer and Length)*
11	DLVL < WD-SZ	Load Word Count less than Word Size*
12	¢EXH	Exhaust*
13	DEND-SEQ	End of Sequence
14	DEND	End of Instruction*
15	DU=RD+WRT	Decimal Unit equals Read or Write*
16	¢PTRA00	Pointer Register Address bit 0*
17	¢PTRA01	Pointer Register Address bit 1*
18	FA/I1	Specifies Active/Inactive state of descriptors 1, 2, and 3
19	FA/I2	
20	FA/I3	
21	¢WRD	Word*
22	¢NINE	Nine-bit*
23	¢SIX	Six-bit*
24	¢FOUR	Four-bit*
25	¢BIT	Bit*
26	Not Used	
27	Not Used	
28	Not Used	
29	Not Used	

*data stored in complement form: 0 - true, 1 - false

DU HISTORY REGISTER FORMAT (CONTINUED)

Bit	Mnemonic	Function
30	FSAMPL	Sample for Mid-Instruction Interrupt
31	DFRST-CT	Specific first count of a sequence*
32	ÇADJ-LENINT	Adjust length*
33	FINTRPTD	Interrupt Indicator (mid-instruction)*
34	FINHIB	Inhibit STC1 (Inhibits instruction counter plus one)*
35	Not Used	
36	FDUD	DU Idle
37	FGDLDA	Descriptor Load Gate A*
38	FGDLDB	Descriptor Load Gate B*
39	FGDLDC	Descriptor Load Gate C*
40	FNLD1	Prepare Alignment Count for First Numeric Operand Load
41	FGLDP1	Numeric Operand One Load Gate
42	FNLD2	Prepare Alignment Count for First Numeric Operand Load
43	FGLDP2	Numeric Operand Two Load Gate
44	FANLD1	Alphanumeric Operand One Load Gate
45	FANLD2	Alphanumeric Operand Two Load Gate
46	FLDWRT1	Load Rewrite Register One Gate
47	FLDWRT2	Load Rewrite Register Two Gate
48	DATA-AVLDU	Data Available*
49	FWRT1	Rewrite One Register Loaded
50	FGSTR	Numeric Store Gate
51	FANSTR	Alphanumeric Store Gate
52	FSTR-ØP-AV	Operand Available to be Stored
53	FEND-SEQ	End Sequence Flag*
54	FLEN 128	Length Less Than 128 Flag*
55	FGCH	Character Operation Gate
56	FANPK	Alphanumeric Packing Cycle Gate
57	FEXMØP	Execute MØP Gate
58	FBLNK	Blanking Gate
59	Not Used	
60	DGBD	Binary to Decimal Execution Gates
61	DGDB	Decimal to Binary Execution Gates
62	DGSP	Shift Procedure Gates
63	FFLTG	Floating Result Flag
64	FRND	Rounding Flag
65	DADD-GATE	Add/Subtract Execution Gates
66	DMP+DV-GATE	Multiply/Divide Execution Gates
67	DXPN-GATE	Exponent Network Execution Gates
68-71	Not Used	

APU History Register Format



Where:

BIT	NAME	
0-14	ESN	is the effective segment number generated
15-16	BSY	is from where the ESN was derived as follows: 0 0 ESN from PSP 0 1 ESN from SNR 1 0 ESN from TSR 1 1 Not used
17-24		APU control flags: These flags indicate the kind of cycle that produced this entry in the history register
17	FDSPTW	Fetch of Descriptor Segment Page Table Word
18	MDSPTW	Modification of Descriptor Segment Page Table Word
19	FSDWP	Fetch of Segment Descriptor Word from a paged Descriptor Segment
20	FPTW	Fetch of Page Table Word
21	FPTW2	Fetch of Page Table Word + 1
22	MPTW	Modification of Page Table Word
23	FANP	Fetch of Final Address from a non-paged segment
24	FAP	Fetch of Final Address from a Paged Segment
25	SDWAMM	An SDW match in the SDWAM occurred

APU HISTORY REGISTER FORMAT (CONTINUED

26-29	SDWAMR	If bit 25 is on then these bits specify the SDWAM register which contained the found SDW
30	PTWAMM	A PTW match in the PTWAM occurred
31-34	PTWAMR	If bit 30 is on then these bits specify the STWAM register which contained the found PTW
35	FLT	This cycle produced an access violation or a directed fault
36-59	ADD	This is the 24 bit absolute address produced by this cycle
60-62	TRR	This is the TPR. TRR value developed by this cycle
70	FHLD	An ACVF or DF is waiting to be processed

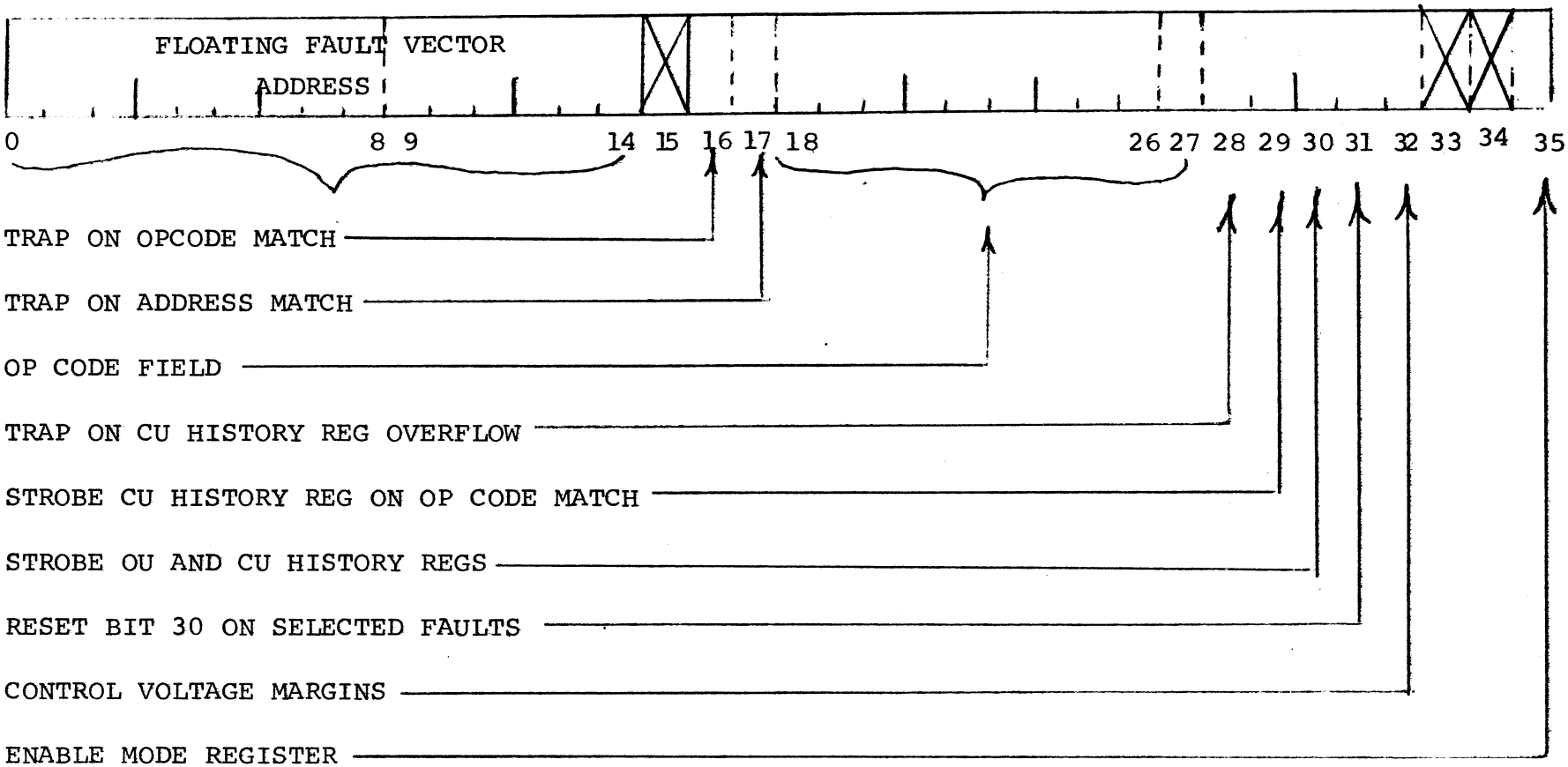
FAULT REGISTER FORMAT

Bit	Fault	Mnemonic	
0	Illegal Ring Order (IRO)		
1	Out of execute bracket (ØEB)	ACV	Group I *
2	Execute bit is off (E-OFF)		
3	Out of read bracket (ØRB)		
4	Read bit is off (R-ØFF)		
5	Out of write bracket (ØWB)		
6	Write bit is off (W-ØFF)		
7	Not a gate (NØGA)		
8	Out of call bracket (ØCB)		
9	Outward call (ØCALL)		
10	Bad outward call (BØC)		
11	Inward return (INRET)		
12	Cross ring transfer (CRT)		
13	Ring alarm (RALR)		
14	Associative memory error (AM-ER)		
15	Out of segment bounds (ØØSB)	ACV	
16	Processor parity error upper	FPAR	
17	Processor parity error lower	FPAR	
18	SC to Proc. Seq. Error #1	FONC	
19	SC to Proc. Seq. Error #2	FONC	
20 21 22 23	Illegal Action Lines		
24 25 26	Illegal Action Channel		
27 28 29	Connect Channel	CON CON CON	
30 31 32 33	Fault Number/Address		
34			
35	Fault/Interrupt		

Bit	Fault	Mnemonic	
0	Illegal Segment Number	STR	Group II*
1	Illegal Op Code	IPR	
2	Illegal Address and Modifier	IPR	
3	Illegal Slave Procedure	IPR	
4	All Other Illegal Procedure	IPR	
5	Nonexistent Address	STR	
6	Out of Bounds	STR	

* Group I is stored for Access Violations and Directed Faults only.

MODE REGISTER



MODE REGISTER FORMAT

0-14 Floating fault vector.

These bits shall constitute the upper 15 bits for the fault pair addresses of the various fault traps enabled by the Mode Register (see bits, 16, 17, and 28).

15 Unassigned

16 Trap on Op code

If bit 16 = 1 and the Op code of the instruction whose address is presently being prepared by the Processor (including indirect cycles) matches bits 18-27 of the Mode Register, the Processor shall do an XED to the second of the fault pairs specified by the Floating fault vector.

17 Trap on Address match.

If bit 17 = 1 and the address in the Processor Address Register matches the address switches on the maintenance panel, the Processor shall do an XED to the fourth of the fault pairs specified by the Floating fault vector.

18-27 Op Code/Software Switches

If either bit 16 or bit 29 = 1, bits 18-27 shall represent an Op code on which the Processor shall trap (see bit 16) or on which it shall strobe the CU History register (see bit 29).

Mode Register (continued)

If both bit 16 and bit 29 = 0, bits 18-27 shall provide the means of software "setting" certain of the Maintenance Panel switches:

18.$\overline{16}$.$\overline{29}$.32 Set CU overlap inhibit. The CU shall wait for the OU to complete its operation on the even instruction operand before it starts the corresponding odd instruction operand. The CU shall also wait for the OU to complete its operation on the odd instruction operand before it fetches the next instruction pair.

19.$\overline{16}$.$\overline{29}$.32 Set Store overlap inhibit. The CU shall wait for the $DA before it issues the next interrupt.

20.$\overline{16}$.$\overline{29}$.32 Set Store Incorrect Data Parity. The CU shall cause incorrect data parity to be sent to the Store for the next store instruction and then reset bit 20.

21 $\overline{16}$ $\overline{29}$ 32 Set Store Incorrect ZAC Parity. The CU shall cause incorrect ZAC parity to be generated on each memory cycle until the $DA of the next store instruction. At this time bit 20 will be reset.

22.$\overline{16}$.$\overline{29}$.32 Set Timing Margin (if the manual switch
23.$\overline{16}$.$\overline{29}$.32 is in the Normal position) accordingly:

24.$\overline{16}$.$\overline{29}$.32 shall control the +5 Voltage Margins
25.16.29.32

	24	25
Low	0	1
High	1	0
Normal	Otherwise	

	22	23
Slow	0	1
Fast	1	1
Normal	X	0

27.$\overline{16}$.$\overline{29}$.32 Unassigned

28 Trap on CU History Register counter overflow.

If bit 28 = 1 and the CU-HR counter overflows, the Processor shall do an XED to the third of the fault pairs specified by the Floating fault vector.

These three traps (Address, Op code, CU-HR counter overflow) shall occur after the next complete odd instruction following their detection. They shall be treated as Group V faults as far as when serviced and when inhibited. Their individual priorities shall be:

1 - CON
2 - TROF
3 - SDF
} Other Group V faults
4 - Op Code trap
5 - CU-HR Counter Overflow
6 - Address match trap
7 - External Interrupts

29 Strobe the CU-HR on Op code.

If bits 29 and 30 (see below) are = 1 and the Op code of the instruction whose address is presently being prepared by the Processor (including indirect cycles) matches bits 18-27 of the Mode Register, the CU-HR will be strobed.

Mode Register (continued)

30 Enable History Registers

If bit 30 = 1, the CU and OU History Registers may be strobed. If bit 30 = 0 or bit 35 = 0 (see below), they will be locked out. This bit will be reset by either an LCPR with the bit corresponding to 30 = 0 or by an Op Not Complete fault. It may be reset by other faults (see bit 31). After being reset, it must be enabled by another LCPR instruction before the History Registers may be strobed again.

- - -

*This means that the CU-HR may be locked out for the latter indirect cycles of an instruction. Also, setting bit 28 shall cause the CU-HR counter to be reset to 0.

Mode Register (continued)

31 Additional resetting of bit 30.

If bit 31 = 1, the following faults will <u>also</u> reset bit 30:

- Lock Up
- Parity
- Command
- Store
- Illegal Procedure

- Shutdown
- Op code trap (bit 16)
- CU-HR counter overflow trap (bit 28)
- Address match trap (bit 17)

Mode Register (Continued)

32 Control Margins

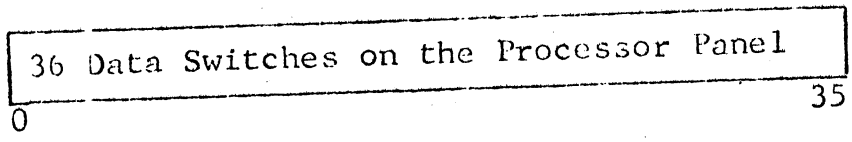
Bit 32 shall be used to inform software when it can control margins. A one shall indicate that software has control (via bits 18-27). A zero shall indicate that hardware (SCAM) has control. Bit 32 shall be set to one when the Normal/Test switch on the Maintenance Panel is in the Test position or when SCAM is sending a Test Mode signal to the Processor.

33-34 Unassigned

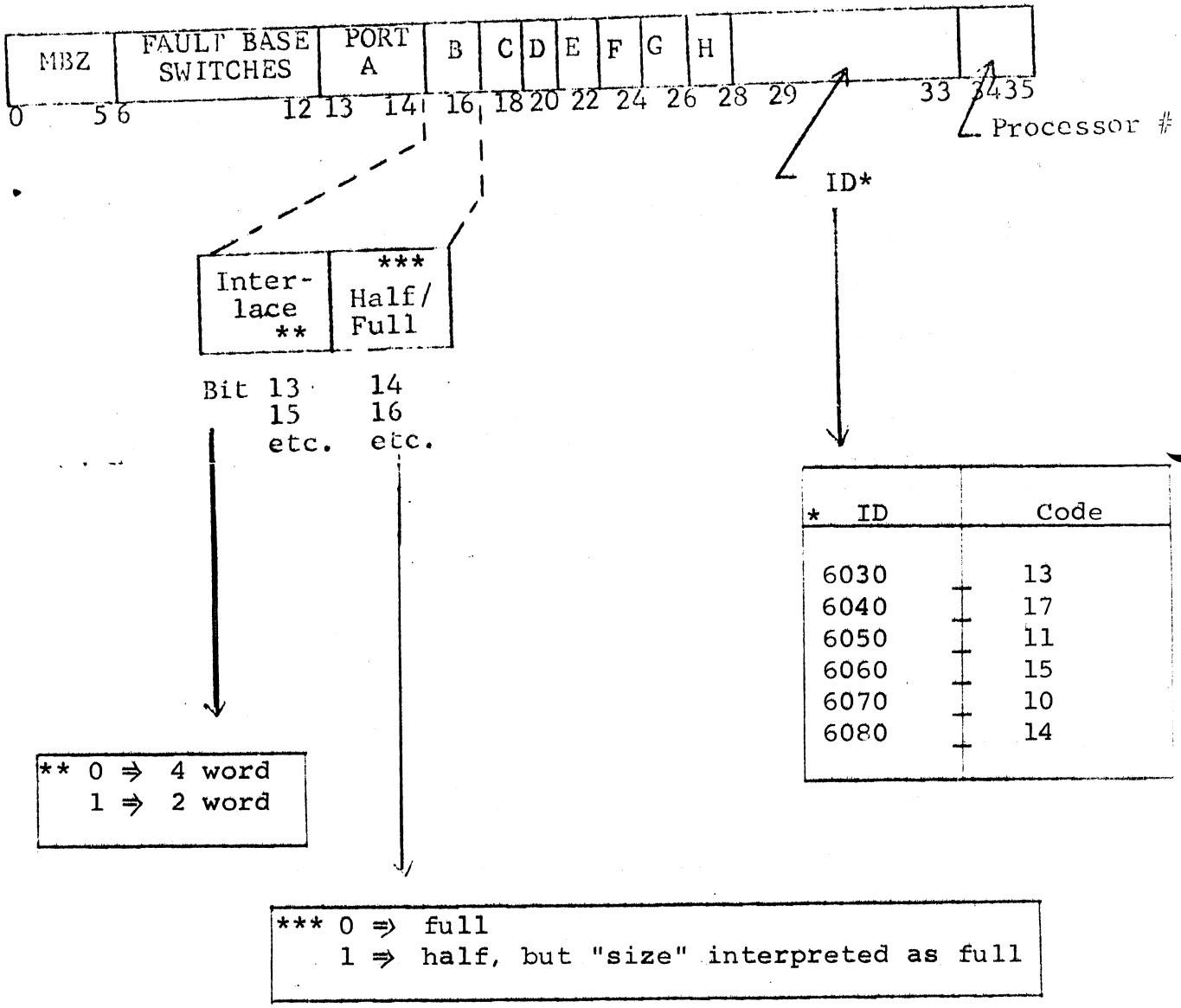
35 Use Mode Register

Unless bit 35 = 1, all other bits in the Mode Register will be ignored and the History Registers shall be locked.

For RSN address (octal) of XXXXX0:



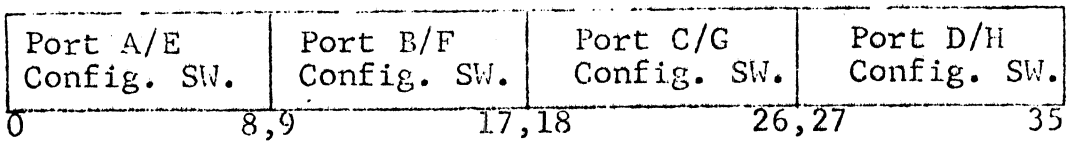
For RSN address (octal) of XXXXX2:

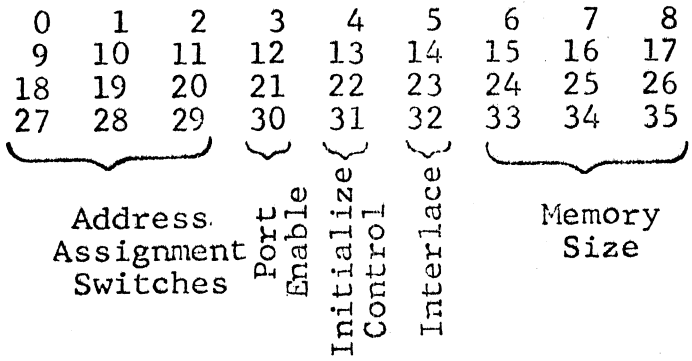


* ID	Code
6030	13
6040	17
6050	11
6060	15
6070	10
6080	14

Processor Number	00	Processor Number 0
	01	Processor Number 1
	10	Processor Number 2
	11	Processor Number 3

For RSW address (octal) of XXXXX1/3:





The fields for the Address Assignment Switches, Port Enable, Initialize Control, Interlace and Memory Size shall be coded as follows:

Field	Code	Meaning
Port Assignment	000 to 111	Reflects Port Address switches:
Port Enable	0	This port not configured into system
	1	This port configured into system
System Initialize Enable	0	Processor can't be initialized from system.
	1	Processor can be initialized from system.
Interlace Enable	0	Port not interlaced.
	1	Enables port to be interlaced with another port pair.
Memory Size	000	32K
	001	64K
	010	96K or 160K
	011	128K
	100	512K
	101	1024K
	110	2048K
	111	256K

Read System Controller Register - Configuration Switches Format

RSCR-CFG (Read System Controller Register - Configuration switches)

The System Controller switches are returned in the following format:

0 2	3 5	6 8	9 11	12	13	14 15	16 19	20 21	22 23	24 25	26 27
Mode A	Bdry A	Mode B	Bdry B	Ø	A/B	Addr. Offset	Port No.	PORT ENABLE 0	1	2	3

EXTENDED

28 29	30 31	32 33	34 35	36 44	45 53	54 62	63 71
PORT ENABLE 4	5	6	7	PROGRAM INTERRUPT MASK ASSIGNMENTS MASK A	MASK B	MASK C	MASK D

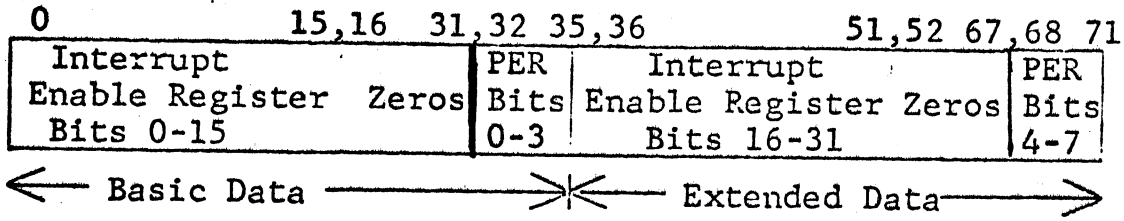
The fields have the following codes and meanings:

Field	Code	Meaning	
Mode A or Mode B fields	000	On Line	Refers to state of store units connected to that system controller port.
	001	Test	
	010	Off Line	
Boundary A or Boundary B fields	000	32K	Refers to size of store connected to that port and is used for address port selection and non-existent address action.
	001	64K	
	011	128K	
	111	256K	
Ø	0	Stores are not interlaced	
	1	Stores are interlaced	
A/B	0	Lower address is in Store A	
	1	Lower address is in Store B	

Field	Code	Meaning	
Address Offset	00	No offset	
	01	16K offset	
	10	32K offset	
	11	64K offset	
Port No.	0000	Port 0	This field tells the requesting system port its own port number assignment in the system controller.
	0001	Port 1	
	0010	Port 2	
	0011	Port 3	
	0100	Port 4	
	0101	Port 5	
	0110	Port 6	
	0111	Port 7	
	1000	Port 8	(Maintenance Panel)
Port Enable Switches (0--7)	00	Port Disabled	
	11	Port Enabled	
	01	Port under program control	
Program Interrupt Mask Assignments A,B,C,D	100 000 000	Assigned to Port 0	
	010 000 000	Assigned to Port 1	
	001 000 000	Assigned to Port 2	
	000 100 000	Assigned to Port 3	
	000 010 000	Assigned to Port 4	
	000 001 000	Assigned to Port 5	
	000 000 100	Assigned to Port 6	
	000 000 010	Assigned to Port 7	
	000 000 001	Assigned to Port 8 (Maintenance Panel)	
	000 000 000	Unassigned	

RSCR-IERn (Read System Controller Register - Interrupt Enable Register Port n) Format

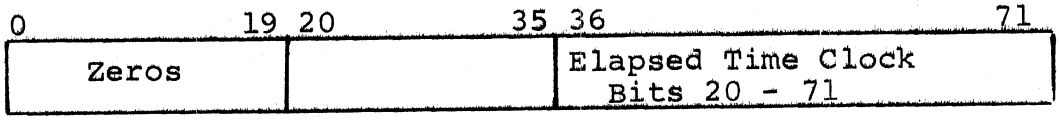
The RSCR-IERn instruction causes the contents of the IER (Program Interrupt Enable Register) assigned to the specified system port of the System Controller loaded into AQ. In addition, the contents of the PER (Port Enable Register, one per system controller) are presented in the same format as for the RMCM instruction. If no Program Interrupt Enable Register is assigned to the port specified, only the PER (Port Enable Register) contents are returned, and the balance of the bits are zeros. The format is as follows:



If no IER is assigned to the specified port, the RSCR-IERn will return only the PER bits and the other fields will be zeros.

RSCR-ETC (Read System Controller Register - Elapsed Time Clock)

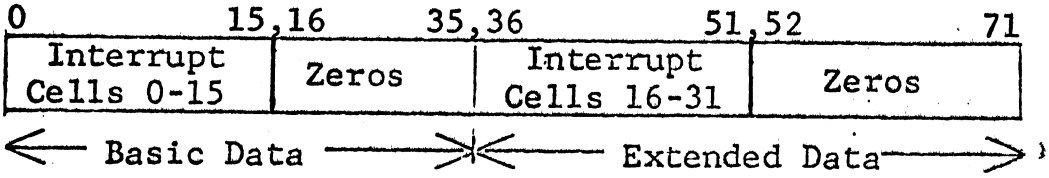
The RSCR-ETC causes the contents of the Elapsed Time Clock Register to be read and loaded into AQ. The format is as follows:



The least bit significance is 1 microsecond. The clock is not settable and turns over every 144 years.

RSCR-IC (Read System Controller Register - Interrupt Cells)

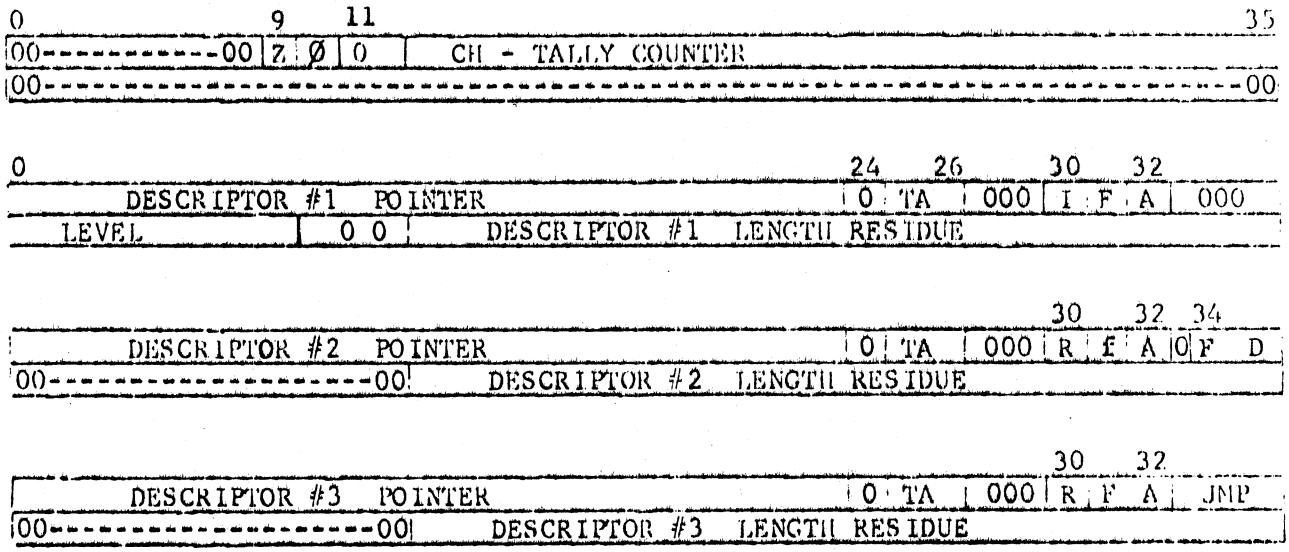
The RSCR-IC instruction causes the contents of the Interrupt cells to be loaded into AQ in the following format:



A RSCR-IC instruction will read (and not reset) the cells.

A SSCR-IC will load new data (1's and 0's) into the cells.

Format of Pointers/Lengths *



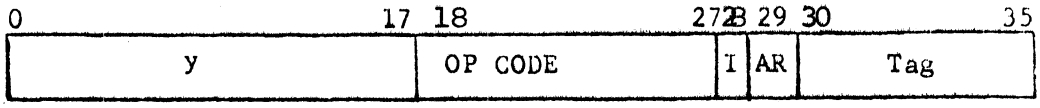
* Format as stored by SPL.

% -	All the bit string instruction results are zero.
Ø -	Negative overpunch found in 6-4 expanded move.
CH-Tally Counter-	The number of characters examined by the SCAN, TCT, or TCTR instruction (up to the interrupt or match).
Descriptor (#1,#2,#3) Pointer -	The last double word accessed by the descriptor (bits 17-23 only valid for initial access).
TA -	Bits 21, 22 (alphanumeric type) of each descriptor.
I -	Ignore EU requests. CU cannot be interruptable by the extension unit. (This bit cannot be loaded.)
F -	First time. (Information in descriptor is valid.)
A -	The descriptor is active. (Information in pair valid.)
LEVEL -	The difference in the number of characters inputted into the processor and the number outputted from the processor.
Descriptor (#1,#2,#3) Length Residue -	The amount of data left in each descriptor.
R -	The last cycle performed must be repeated. (This bit cannot be loaded.)
f (D2 only) -	ORing of first time indicator (bit 34) and Direct indicator (bit 35).
D (D2 only) -	Direct type of operation. Information can be found in the descriptor.
JMP -	Represents the number of words in this multiword instruction.

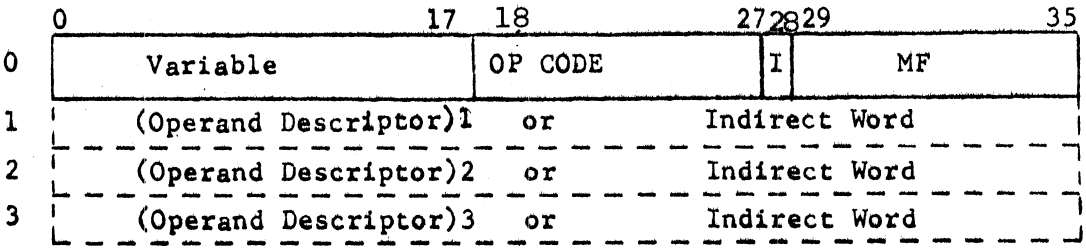
	33	34	35
2 descriptor instruction -	0	1	1
3 descriptor instruction -	1	0	0

INSTRUCTION WORD FORMAT

Standard H6000 Instruction Format

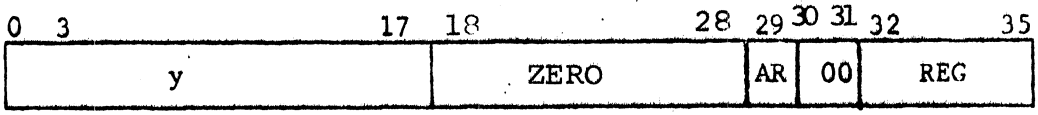


EIS multiword instruction format:

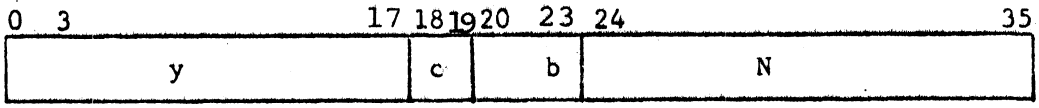


where:

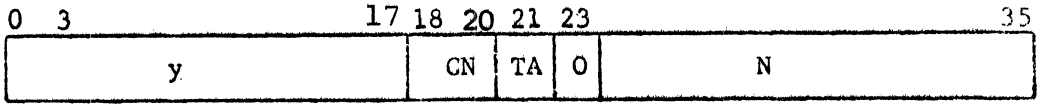
indirect word format:



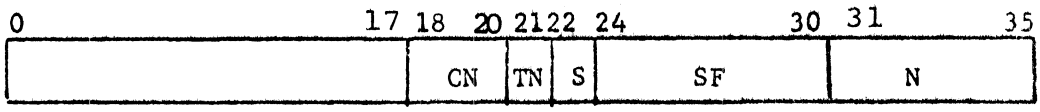
and bit string operand:



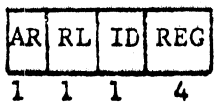
Alphanumeric operand:



Numeric operand:



and modification fields format:



Instruction			
502	(1)	A4BD	Add 4-bit character displacement to AR
501	(1)	A6BD	Add 6-bit character displacement to AR
500	(1)	A9BD	Add 9-bit character displacement to AR
560	(1)	AAR0	Alphanumeric descriptor to AR0
561	(1)	AAR1	Alphanumeric descriptor to AR1
562	(1)	AAR2	Alphanumeric descriptor to AR2
563	(1)	AAR3	Alphanumeric descriptor to AR3
564	(1)	AAR4	Alphanumeric descriptor to AR4
565	(1)	AAR5	Alphanumeric descriptor to AR5
566	(1)	AAR6	Alphanumeric descriptor to AR6
567	(1)	AAR7	Alphanumeric descriptor to AR7
503	(1)	ABD	Add bit displacement to AR
202	(1)	AD2D	Add using 2 decimal operands
222	(1)	AD3D	Add using 3 decimal operands
212	(0)	ABSA	Absolute Address to Accumulator
075	(0)	ADA	Add to Accumulator
077	(0)	ADAQ	Add to A-Q
415	(0)	ADE	Add to Exponent Register
033	(0)	ADL	Add Low to A-Q
035	(0)	ADLA	Add Logical to Accumulator
037	(0)	ADLAQ	Add Logical to A-Q
036	(0)	ADLQ	Add Logical to Quotient
020	(0)	ADLX0	Add Logical to Index 0

Instruction

021	(0)	ADLX1	Add Logical to Index 1
022	(0)	ADLX2	Add Logical to Index 2
023	(0)	ADLX3	Add Logical to Index 3
024	(0)	ADLX4	Add Logical to Index 4
025	(0)	ADLX5	Add Logical to Index 5
026	(0)	ADLX6	Add Logical to Index 6
027	(0)	ADLX7	Add Logical to Index 7
076	(0)	ADQ	Add to Quotient Register
050	(0)	ADWP0	Add to Word Number Field of PR0
051	(0)	ADWP1	Add to Word Number Field of PR1
052	(0)	ADWP2	Add to Word Number Field of PR2
053	(0)	ADWP3	Add to Word Number Field of PR3
150	(0)	ADWP4	Add to Word Number Field of PR4
151	(0)	ADWP5	Add to Word Number Field of PR5
152	(0)	ADWP6	Add to Word Number Field of PR6
153	(0)	ADWP7	Add to Word Number Field of PR7
060	(0)	ADX0	Add to Index 0
061	(0)	ADX1	Add to Index 1
062	(0)	ADX2	Add to Index 2
063	(0)	ADX3	Add to Index 3
064	(0)	ADX4	Add to Index 4
065	(0)	ADX5	Add to Index 5
066	(0)	ADX6	Add to Index 6
067	(0)	ADX7	Add to Index 7

Instruction

775 (0)	ALR	Accumulator Left Rotate
735 (0)	ALS	Accumulator Left Shift
375 (0)	ANA	And to Accumulator
377 (0)	ANAQ	And to A-Q
376 (0)	ANQ	And to Quotient
355 (0)	ANSA	And to Storage Accumulator
356 (0)	ANSQ	And to Storage Quotient
340 (0)	ANSX0	And to Storage Index 0
341 (0)	ANSX1	And to Storage Index 1
342 (0)	ANSX2	And to Storage Index 2
343 (0)	ANSX3	And to Storage Index 3
344 (0)	ANSX4	And to Storage Index 4
345 (0)	ANSX5	And to Storage Index 5
346 (0)	ANSX6	And to Storage Index 6
347 (0)	ANSX7	And to Storage Index 7
360 (0)	ANX0	And to Index 0
361 (0)	ANX1	And to Index 1
362 (0)	ANX2	And to Index 2
363 (0)	ANX3	And to Index 3
364 (0)	ANX4	And to Index 4
365 (0)	ANX5	And to Index 5
366 (0)	ANX6	And to Index 6
367 (0)	ANX7	And to Index 7
054 (0)	AOS	Add One to Storage

Instruction

540 (1)	ARA0	AR0 to Alphanumeric Descriptor
541 (1)	ARA1	AR1 to Alphanumeric Descriptor
542 (1)	ARA2	AR1 to Alphanumeric Descriptor
543 (1)	ARA3	AR3 to Alphanumeric Descriptor
544 (1)	ARA4	AR4 to Alphanumeric Descriptor
545 (1)	ARA5	AR5 to Alphanumeric Descriptor
546 (1)	ARA6	AR6 to Alphanumeric Descriptor
547 (1)	ARA7	AR7 to Alphanumeric Descriptor
771 (0)	ARL	Accumulator Right Logical
640 (1)	ARN0	AR0 to Numeric Descriptor
641 (1)	ARN1	AR1 to Numeric Descriptor
642 (1)	ARN2	AR2 to Numeric Descriptor
643 (1)	ARN3	AR3 to Numeric Descriptor
644 (1)	ARN4	AR4 to Numeric Descriptor
645 (1)	ARN5	AR5 to Numeric Descriptor
646 (1)	ARN6	AR6 to Numeric Descriptor
647 (1)	ARN7	AR7 to Numeric Descriptor
731 (0)	ARS	Accumulator Right Shift
055 (0)	ASA	Add Stored to Accumulator
056 (0)	ASQ	Add Stored to Quotient
040 (0)	ASX0	Add Stored to Index 0
041 (0)	ASX1	Add Stored to Index 1
042 (0)	ASX2	Add Stored to Index 2
043 (0)	ASX3	Add Stored to Index 3
044 (0)	ASX4	Add Stored to Index 4

Instruction

045	(0)	ASX5	Add Stored to Index 5
046	(0)	ASX6	Add Stored to Index 6
047	(0)	ASX7	Add Stored to Index 7
071	(0)	AWCA	Add with Carry to Accumulator
072	(0)	AWCQ	Add with Carry to Quotient
507	(1)	AWD	Add Word Displacement to AR
505	(0)	BCD	Binary to Binary-Coded-Decimal
301	(1)	BTD	Binary to Decimal Convert
713	(0)	CALL SP	Call
532	(1)	CAMP	Clear Associative Memory Paged
532	(0)	CAMS	Clear Associative Memory Segmented
315	(0)	CANA	Comparative And With Accumulator
317	(0)	CANAQ	Comparative And With A-Q
316	(0)	CANQ	Comparative And With Quotient
300	(0)	CANX0	Comparative And With Index 0
301	(0)	CANX1	Comparative And With Index 1
302	(0)	CANX2	Comparative And With Index 2
303	(0)	CANX3	Comparative And With Index 3
304	(0)	CANX4	Comparative And With Index 4
305	(0)	CANX5	Comparative And With Index 5
306	(0)	CANX6	Comparative And With Index 6
307	(0)	CANX7	Comparative And With Index 7
015	(0)	CIOC	Connect I/O Channel

Instruction

405	(0)	CMG	Compare With Magnitude
211	(0)	CMK	Compare Masked
115	(0)	CMPA	Compare With Accumulator
066	(1)	CMPB	Compare Bit Strings
106	(1)	CMPC	Compare Alphanumeric Character String
117	(0)	CMPAQ	Compare with A-Q
303	(1)	CMPN	Compare Numeric
116	(0)	CMPQ	Compare With Quotient Register
100	(0)	CMPX0	Compare With Index 0
101	(0)	CMPX1	Compare With Index 1
102	(0)	CMPX2	Compare With Index 2
103	(0)	CMPX3	Compare With Index 3
104	(0)	CMPX4	Compare With Index 4
105	(0)	CMPX5	Compare With Index 5
106	(0)	CMPX6	Compare With Index 6
107	(0)	CMPX7	Compare With Index 7
215	(0)	CNAA	Comparative Not With Accumulator
217	(0)	CNAAQ	Comparative Not With A-Q
216	(0)	CNAQ	Comparative Not With Quotient
200	(0)	CNAX0	Comparative Not With Index 0
201	(0)	CNAX1	Comparative Not With Index 1
202	(0)	CNAX2	Comparative Not With Index 2
203	(0)	CNAX3	Comparative Not With Index 3
204	(0)	CNAX4	Comparative Not With Index 4
205	(0)	CNAX5	Comparative Not With Index 5
206	(0)	CNAX6	Comparative Not With Index 6

Instruction			
207	(0)	CNAX7	Comparative Not With Index 7
111	(0)	CWL	Compare With Limits
060	(0)	CSL	Combine Bit Strings Left
061	(0)	CSR	Combine Bit Strings Right
477	(0)	DFAD	Double Precision Floating Add
427	(0)	DFCMG	Double Precision Floating Compare Magnitude
517	(0)	DFCMP	Double Precision Floating Compare
527	(0)	DFDI	Double Precision Floating Divide Inverted
567	(0)	DFDV	Double Precision Floating Divide
433	(0)	DFLD	Double Precision Floating Load
463	(0)	DFMP	Double Precision Floating Multiply
473	(0)	DFRD	Double Precision Floating Round
577	(0)	DFSB	Double Precision Floating Subtract
457	(0)	DFST	Double Precision Floating Store
472	(0)	DFSTR	Double Precision Floating Store Round
616	(0)	DIS	Delay Until Interrupt Signal
506	(0)	DIV	Divide Integer
002	(0)	DRL	Derail
305	(1)	DTB	Decimal to Binary Convert
437	(0)	DUFA	Double Precision Unnormalized Floating Add
423	(0)	DUFM	Double Precision Unnormalized Floating Multiply
537	(0)	DUFS	Double Precision Unnormalized Floating Subtract
207	(1)	DV2D	Divide Using 2 Decimal Operands
227	(1)	DV3D	Divide Using 3 Decimal Operands
507	(0)	DVF	Divide Fraction
635	(0)	EAA	Effective Address to Accumulator
636	(0)	EAQ	Effective Address to Q

Instruction

311 (0)	EASP0	Effective Address to Segment Number Field of PR0
310 (1)	EASP1	Effective Address to Segment Number Field of PR1
313 (0)	EASP2	Effective Address to Segment Number Field of PR2
312 (1)	EASP3	Effective Address to Segment Number Field of PR3
331 (0)	EASP4	Effective Address to Segment Number Field of PR4
330 (1)	EASP5	Effective Address to Segment Number Field of PR5
333 (0)	EASP6	Effective Address to Segment Number Field of PR6
332 (1)	EASP7	Effective Address to Segment Number Field of PR7
310 (0)	EAWP0	Effective Address to Word Number Field of PR0
311 (1)	EAWP1	Effective Address to Word Number Field of PR1
312 (0)	EAWP2	Effective Address to Word Number Field of PR2
313 (1)	EAWP3	Effective Address to Word Number Field of PR3
330 (0)	EAWP4	Effective Address to Word Number Field of PR4
331 (1)	EAWP5	Effective Address to Word Number Field of PR5
332 (0)	EAWP6	Effective Address to Word Number Field of PR6
333 (1)	EAWP7	Effective Address to Word Number Field of PR7
620 (0)	EAX0	Effective Address to Index 0
621 (0)	EAX1	Effective Address to Index 1
622 (0)	EAX2	Effective Address to Index 2
623 (0)	EAX3	Effective Address to Index 3
624 (0)	EAX4	Effective Address to Index 4
625 (0)	EAX5	Effective Address to Index 5
626 (0)	EAX6	Effective Address to Index 6
627 (0)	EAX7	Effective Address to Index 7

Instruction			
213	(0)	EPAQ	Effective Pointer to A-Q
350	(1)	EPBP0	Effective Pointer at Base to PR0
351	(0)	EPBP1	Effective Pointer at Base to PR1
352	(1)	EPBP2	Effective Pointer at Base to PR2
353	(0)	EPBP3	Effective Pointer at Base to PR3
370	(1)	EPBP4	Effective Pointer at Base to PR4
371	(0)	EPBP5	Effective Pointer at Base to PR5
372	(1)	EPBP6	Effective Pointer at Base to PR6
373	(0)	EPBP7	Effective Pointer at Base to PR7
350	(0)	EPP0	Effective Pointer to PR0
351	(1)	EPP1	Effective Pointer to PR1
352	(0)	EPP2	Effective Pointer to PR2
353	(1)	EPP3	Effective Pointer to PR3
370	(0)	EPP4	Effective Pointer to PR4
371	(1)	EPP5	Effective Pointer to PR5
372	(0)	EPP6	Effective Pointer to PR6
373	(1)	EPP7	Effective Pointer to PR7
675	(0)	ERA	Exclusive Or to Accumulator
677	(0)	ERAQ	Exclusive Or to A-Q
676	(0)	ERQ	Exclusive Or to Quotient
655	(0)	ERSA	Exclusive Or to Storage Accumulator
656	(0)	ERSQ	Exclusive Or to Storage Quotient
640	(0)	ERSX0	Exclusive Or to Storage Index 0
641	(0)	ERSX1	Exclusive Or to Storage Index 1
642	(0)	ERSX2	Exclusive Or to Storage Index 2
643	(0)	ERSX3	Exclusive Or to Storage Index 3
644	(0)	ERSX4	Exclusive Or to Storage Index 4

Instruction

645	(0)	ERSX5	Exclusive Or to Storage Index 5
646	(0)	ERSX6	Exclusive Or to Storage Index 6
647	(0)	ERSX7	Exclusive Or to Storage Index 7
660	(0)	ERX0	Exclusive Or to Index 0
661	(0)	ERX1	Exclusive Or to Index 1
662	(0)	ERX2	Exclusive Or to Index 2
663	(0)	ERX3	Exclusive Or to Index 3
664	(0)	ERX4	Exclusive Or to Index 4
665	(0)	ERX5	Exclusive Or to Index 5
666	(0)	ERX6	Exclusive Or to Index 6
667	(0)	ERX7	Exclusive Or to Index 7
475	(0)	FAD	Floating Add
425	(0)	FCMG	Floating Compare Magnitude
515	(0)	FCMP	Floating Compare
525	(0)	FDI	Floating Divide Inverted
565	(0)	FDV	Floating Divide
431	(0)	FLD	Floating Load
461	(0)	FMP	Floating Multiply
513	(0)	FNEG	Floating Negate
573	(0)	FNO	Floating Normalize
471	(0)	FRD	Floating Round
575	(0)	FSB	Floating Subtract
455	(0)	FST	Floating Store
470	(0)	FSTR	Floating Store Rounded
430	(0)	FSZN	Floating Set Zero and Negative Indicators
774	(0)	GTB	Gray to Binary

Instruction		
760	(1) LAR0	Load AR0
761	(1) LAR1	Load AR1
762	(1) LAR2	Load AR2
763	(1) LAR3	Load AR3
764	(1) LAR4	Load AR4
765	(1) LAR5	Load AR5
766	(1) LAR6	Load AR6
767	(1) LAR7	Load AR7
463	(1) LAREG	Load Address Registers
230	(0) LBAR	Load Base Address Register
335	(0) LCA	Load Complement into A
337	(0) LCAQ	Load Complement into A-Q
674	(0) LCPR	Load Central Processor Register
336	(0) LCQ	Load Complement into Q
320	(0) LCX0	Load Complement into Index 0
321	(0) LCX1	Load Complement into Index 1
322	(0) LCX2	Load Complement into Index 2
323	(0) LCX3	Load Complement into Index 3
324	(0) LCX4	Load Complement into Index 4
325	(0) LCX5	Load Complement into Index 5
326	(0) LCX6	Load Complement into Index 6
327	(0) LCX7	Load Complement into Index 7
235	(0) LDA	Load Accumulator
034	(0) LDAC	Load Accumulator and Clear
237	(0) LDAQ	Load A-Q Register
232	(0) LDBR	Load Descriptor Base Register
411	(0) LDE	Load Exponent Register
634	(0) LDI	Load Indicator Register

Instruction			
236	(0)	LDQ	Load Quotient Register
032	(0)	LDQC	Load Q and Clear
627	(0)	LDT	Load Timer Register
220	(0)	LDX0	Load Index 0
221	(0)	LDX1	Load Index 1
222	(0)	LDX2	Load Index 2
223	(0)	LDX3	Load Index 3
224	(0)	LDX4	Load Index 4
225	(0)	LDX5	Load Index 5
226	(0)	LDX6	Load Index 6
227	(0)	LDX7	Load Index 7
777	(0)	LLR	Long Left Rotate
737	(0)	LLS	Long Left Shift
467	(1)	LPL	Load Pointers and Lengths
173	(0)	LPRI	Load Pointer Register From ITS
760	(0)	LPRP0	Load Pointer Register 0 Packed
761	(0)	LPRP1	Load Pointer Register 1 Packed
762	(0)	LPRP2	Load Pointer Register 2 Packed
763	(0)	LPRP3	Load Pointer Register 3 Packed
764	(0)	LPRP4	Load Pointer Register 4 Packed
765	(0)	LPRP5	Load Pointer Register 5 Packed
766	(0)	LPRP6	Load Pointer Register 6 Packed
767	(0)	LPRP7	Load Pointer Register 7 Packed
257	(1)	LPTP	Load Page Table Pointers
173	(1)	LPTR	Load Page Table Registers
774	(1)	LRA	Load Ring Alarm
073	(0)	LREG	Load Registers
773	(0)	LRL	Long Right Logical
733	(0)	LRS	Long Right Shift

Instruction			
257	(0)	LSDP	Load Segment Descriptor Pointers
232	(1)	LSDR	Load Segment Descriptor Registers
720	(0)	LXL0	Load Index 0 from Lower
721	(0)	LXL1	Load Index 1 from Lower
722	(0)	LXL2	Load Index 2 from Lower
723	(0)	LXL3	Load Index 3 from Lower
724	(0)	LXL4	Load Index 4 from Lower
725	(0)	LXL5	Load Index 5 from Lower
726	(0)	LXL6	Load Index 6 from Lower
727	(0)	LXL7	Load Index 7 from Lower
100	(1)	MLR	Move Alphanumeric Left to Right
001	(0)	MME1	Master Mode Entry 1
004	(0)	MME2	Master Mode Entry 2
005	(0)	MME3	Master Mode Entry 3
007	(0)	MME4	Master Mode Entry 4
206	(1)	MP2D	Multiply Using 2 Decimal Operands
226	(1)	MP3D	Multiply Using 3 Decimal Operands
401	(0)	MPF	Multiply Fraction
402	(0)	MPY	Multiply Integer
101	(1)	MRL	Move Alphanumeric Left to Right
020	(1)	MVE	Move Alphanumeric Edited
300	(1)	MVN	Move Numeric
024	(1)	MVNE	Move Numeric Edited
160	(1)	MVT	Move Alphanumeric with Translation
660	(1)	NAR0	Numeric Descriptor to AR0
661	(1)	NAR1	Numeric Descriptor to AR1
662	(1)	NAR2	Numeric Descriptor to AR2

Instruction			
663	(1)	NAR3	Numeric Descriptor to AR3
664	(1)	NAR4	Numeric Descriptor to AR4
665	(1)	NAR5	Numeric Descriptor to AR5
666	(1)	NAR6	Numeric Descriptor to AR6
667	(1)	NAR7	Numeric Descriptor to AR7
531	(0)	NEG	Negate Accumulator
533	(0)	NEGL	Negate Long
011	(0)	NOP	No Operation
275	(0)	ORA	Or to Accumulator
277	(0)	ORAQ	Or to A-Q
276	(0)	ORQ	Or to Quotient
255	(0)	ORSA	Or to Storage Accumulator
256	(0)	ORSQ	Or to Storage Quotient
240	(0)	ORSX0	Or to Storage Index 0
241	(0)	ORSX1	Or to Storage Index 1
242	(0)	ORSX2	Or to Storage Index 2
243	(0)	ORSX3	Or to Storage Index 3
244	(0)	ORSX4	Or to Storage Index 4
245	(0)	ORSX5	Or to Storage Index 5
246	(0)	ORSX6	Or to Storage Index 6
247	(0)	ORSX7	Or to Storage Index 7
260	(0)	ORX0	Or to Index 0
261	(0)	ORX1	Or to Index 1
262	(0)	ORX2	Or to Index 2
263	(0)	ORX3	Or to Index 3

Instruction

264	(0)	ORX4	Or to Index 4
265	(0)	ORX5	Or to Index 5
266	(0)	ORX6	Or to Index 6
267	(0)	ORX7	Or to Index 7
012	(0)	PULS1	Pulse Location 1
013	(3)	PULS2	Pulse Location 2
776	(0)	QLR	Quotient Left Rotate
736	(0)	QLS	Quotient Left Shift
772	(0)	QRL	Quotient Right Logical
732	(0)	QRS	Quotient Right Shift
633	(0)	RCCL	Read Calendar Clock
613	(0)	RCU	Restore Control Unit
630	(0)	RET	Return
233	(0)	RMCM	Read Memory Controller Mask
560	(0)	RPD	Repeat Double
500	(0)	RPL	Repeat Link
520	(0)	RPT	Repeat
413	(0)	RSCR	Read System Controller Register
231	(0)	RSW	Read Switches
610	(0)	RTCD	Return Control Double
522	(1)	S4BD	Subtract 4-bit Character Displacement from AR
521	(1)	S6BD	Subtract 6-bit Character Displacement from AR
520	(1)	S9BD	Subtract 9-bit Character Displacement from AR
740	(1)	SAR0	Store AR0
741	(1)	SAR1	Store AR1
742	(1)	SAR2	Store AR2
743	(1)	SAR3	Store AR03

Instruction			
744	(1)	SAR4	Store AR4
745	(1)	SAR5	Store AR5
746	(1)	SAR6	Store AR6
747	(1)	SAR7	Store AR7
440	(1)	SAREG	Store Address Registers
203	(1)	SB2D	Subtract Using 2 decimal operands
223	(1)	SB3D	Subtract Using 3 decimal operands
175	(0)	SBA	Subtract from Accumulator
350	(0)	SBAR	Store Base Address Register
177	(0)	SBAQ	Subtract From A-Q
523	(1)	SBD	Subtract Bit Displacement from AR
135	(0)	SBLA	Subtract Logical from Accumulator
137	(0)	SBLAQ	Subtract Logical from A-Q
136	(0)	SBLQ	Subtract Logical from Quotient
120	(0)	SBLX0	Subtract Logical from Index 0
121	(0)	SBLX1	Subtract Logical from Index 1
122	(0)	SBLX2	Subtract Logical from Index 2
123	(0)	SBLX3	Subtract Logical from Index 3
124	(0)	SBLX4	Subtract Logical from Index 4
125	(0)	SBLX5	Subtract Logical from Index 5
126	(0)	SBLX6	Subtract Logical from Index 6
127	(0)	SBLX7	Subtract Logical from Index 7
176	(0)	SBQ	Subtract From Quotient
160	(0)	SBX0	Subtract From Index 0
161	(0)	SBX1	Subtract From Index 1
162	(0)	SBX2	Subtract From Index 2

Instruction

163	(0)	SBX3	Subtract From Index 3
164	(0)	SBX4	Subtract From Index 4
165	(0)	SBX5	Subtract From Index 5
166	(0)	SBX6	Subtract From Index 6
167	(0)	SBX7	Subtract From Index 7
120	(1)	SCD	Scan Character Double
121	(1)	SCDR	Scan Character Double in Reverse
124	(1)	SCM	Scan with Mask
125	(1)	SCMR	Scan with Mask in Reverse
452	(0)	SCPR	Store Control Processor Register
657	(0)	SCU	Store Control Unit
154	(0)	SDBR	Store Descriptor Base Register
553	(0)	SMCM	Set Memory Controller
451	(0)	SMIC	Set Memory Interrupt Cells
250	(1)	SPBP0	Store Segment Base Pointer of PR0
251	(0)	SPBP1	Store Segment Base Pointer of PR1
252	(1)	SPBP2	Store Segment Base Pointer of PR2
253	(0)	SPBP3	Store Segment Base Pointer of PR3
650	(1)	SPBP4	Store Segment Base Pointer of PR4
651	(0)	SPBP5	Store Segment Base Pointer of PR5
652	(1)	SPBP6	Store Segment Base Pointer of PR6
653	(0)	SPBP7	Store Segment Base Pointer of PR7
447	(1)	SPL	Store Pointer and Lengths

Instruction

254	(0)	SPRI	Store Pointer Register as ITS
250	(0)	SPRI0	Store Pointer Register 0 as ITS
251	(1)	SPRI1	Store Pointer Register 1 as ITS
252	(0)	SPRI2	Store Pointer Register 2 as ITS
253	(1)	SPRI3	Store Pointer Register 3 as ITS
650	(0)	SPRI4	Store Pointer Register 4 as ITS
651	(1)	SPRI5	Store Pointer Register 5 as ITS
652	(0)	SPRI6	Store Pointer Register 6 as ITS
653	(1)	SPRI7	Store Pointer Register 7 as ITS
540	(0)	SPRP0	Store Pointer Register 0 Packed
541	(0)	SPRP1	Store Pointer Register 1 Packed
542	(0)	SPRP2	Store Pointer Register 2 Packed
543	(0)	SPRP3	Store Pointer Register 3 Packed
544	(0)	SPRP4	Store Pointer Register 4 Packed
545	(0)	SPRP5	Store Pointer Register 5 Packed
546	(0)	SPRP6	Store Pointer Register 6 Packed
547	(0)	SPRP7	Store Pointer Register 7 Packed
557	(1)	SPTP	Store Page Table Pointers
154	(1)	SPTR	Store Page Table Registers
754	(1)	SRA	Store Ring Alarm Register
753	(0)	SREG	Store Registers

Instruction

155	(0)	SSA	Subtract Stored From Accumulator
057	(0)	SSCR	Set System Controller Register
557	(0)	SSDP	Store Segment Descriptor Pointers
254	(1)	SSDR	Store Segment Descriptor Registers
156	(0)	SSQ	Subtract Stored from Quotient
140	(0)	SSX0	Subtract Stored from Index 0
141	(0)	SSX1	Subtract Stored from Index 1
142	(0)	SSX2	Subtract Stored from Index 2
143	(0)	SSX3	Subtract Stored from Index 3
144	(0)	SSX4	Subtract Stored from Index 4
145	(0)	SSX5	Subtract Stored from Index 5
146	(0)	SSX6	Subtract Stored from Index 6
147	(0)	SSX7	Subtract Stored from Index 7
755	(0)	STA	Store Accumulator
354	(0)	STAC	Store A Conditional
654	(0)	STACQ	Store A Conditional on Q
757	(0)	STAQ	Store A-Q
551	(0)	STBA	Store 9 bit Character of A
552	(0)	STBQ	Store 9 bit Character of Q
554	(0)	STC1	Store Instruction Counter + 1
750	(0)	STC2	Store Instruction Counter + 2
751	(0)	STCA	Store 6 bit Character of A
357	(0)	STCD	Store Control Double
752	(0)	STCQ	Store 6 bit Character of Q
456	(0)	STE	Store Exponent Register
754	(0)	STI	Store Indicators

Instruction			
756	(0)	STQ	Store Quotient
454	(0)	STT	Store Timer Register
740	(0)	STX0	Store Index 0
741	(0)	STX1	Store Index 1
742	(0)	STX2	Store Index 2
743	(0)	STX3	Store Index 3
744	(0)	STX4	Store Index 4
745	(0)	STX5	Store Index 5
745	(0)	STX6	Store Index 6
747	(0)	STX7	Store Index 7
450	(0)	STZ	Store Zero
171	(0)	SWCA	Subtract with Carry from Accumulator
172	(0)	SWCQ	Subtract with Carry from Quotient
440	(0)	SXL0	Store Index 0 in Lower
441	(0)	SXL1	Store Index 1 in Lower
442	(0)	SXL2	Store Index 2 in Lower
443	(0)	SXL3	Store Index 3 in Lower
444	(0)	SXL4	Store Index 4 in Lower
445	(0)	SXL5	Store Index 5 in Lower
446	(0)	SXL6	Store Index 6 in Lower
447	(0)	SXL7	Store Index 7 in Lower
234	(0)	SZN	Set Zero and Negative Indicators
214	(0)	SZNC	Set Zero and Negative Indicators and Clear

Instruction		
064 (1)	SZTL	Set Zero and Truncation Indicators with Bit Strir Left
065 (1)	SZTR	Set Zero and Truncation Indicators with Bit String Right
164 (1)	TCT	Test Character and Translate
165 (1)	TCTR	Test Character and Translate in Reverse
614 (0)	TEO	Transfer on Exponent Overflow
615 (0)	TEU	Transfer on Exponent Underflow
604 (0)	TMI	Transfer on Minus
604 (1)	TMOZ	Transfer on Minus or Zero
602 (0)	TNC	Transfer on No Carry
601 (0)	TNZ	Transfer on Not Zero
617 (0)	TOV	Transfer on Overflow
605 (0)	TPL	Transfer on Plus
602 (1)	TPNZ	Transfer on Plus and Nonzero
710 (0)	TRA	Transfer Unconditionally
603 (0)	TRC	Transfer on Carry
601 (1)	TRTF	Transfer on Truncation Indicator OFF
600 (1)	TRTN	Transfer on Truncation Indicator ON
270 (0)	TSP0	Transfer And Set Pointer Register 0
271 (0)	TSP1	Transfer And Set Pointer Register 1
272 (0)	TSP2	Transfer And Set Pointer Register 2
273 (0)	TSP3	Transfer And Set Pointer Register 3
670 (0)	TSP4	Transfer And Set Pointer Register 4
671 (0)	TSP5	Transfer And Set Pointer Register 5
672 (0)	TSP6	Transfer And Set Pointer Register 6
673 (0)	TSP7	Transfer And Set Pointer Register 7

Instruction

715	(0)	TSS	Transfer And Set Slave
700	(0)	TSX0	Transfer and Set Index 0
701	(0)	TSX1	Transfer And Set Index 1
702	(0)	TSX2	Transfer And Set Index 2
703	(0)	TSX3	Transfer And Set Index 3
704	(0)	TSX4	Transfer And Set Index 4
705	(0)	TSX5	Transfer And Set Index 5
706	(0)	TSX6	Transfer And Set Index 6
707	(0)	TSX7	Transfer And Set Index 7
607	(0)	TTF	Transfer On Tally Indicator Off
606	(1)	TTN	Transfer On Tally Indicator On
600	(0)	TZE	Transfer on Zero
435	(0)	UFA	Unnormalized Floating Add
421	(0)	UFM	Unnormalized Floating Multiply
535	(0)	UFS	Unnormalized Floating Subtract
716	(0)	XEC	Execute
717	(0)	XED	Execute Double

	000	001	002	003	004	005	006	007
000		MME	DRL		(MME2)	(MME3)		(MME4)
020	ADLX0	ADLX1	ADLX2	ADLX3	ADLX4	ADLX5	ADLX6	ADLX7
040	ASX0	ASX1	ASX2	ASX3	ASX4	ASX5	ASX6	ASX7
060	ADX0	ADX1	ADX2	ADX3	ADX4	ADX5	ADX6	ADX7
100	CMPX0	CMPX1	CMPX2	CMPX3	CMPX4	CMPX5	CMPX6	CMPX7
120	SBLX0	SBLX1	SBLX2	SBLX3	SBLX4	SBLX5	SBLX6	SBLX7
140	SSX0	SSX1	SSX2	SSX3	SSX4	SSX5	SSX6	SSX7
160	SBX0	SBX1	SBX2	SBX3	SBX4	SBX5	SBX6	SBX7
200	CNAX0	CNAX1	CNAX2	CNAX3	CNAX4	CNAX5	CNAX6	CNAX7
220	LDX0	LDX1	LDX2	LDX3	LDX4	LDX5	LDX6	LDX7
240	ORSX0	ORSX1	ORSX2	ORSX3	ORSX4	ORSX5	ORSX6	ORSX7
260	ORX0	ORX1	ORX2	ORX3	ORX4	ORX5	ORX6	ORX7
300	CANX0	CANX1	CANX2	CANX3	CANX4	CANX5	CANX6	CANX7
320	LCX0	LCX1	LCX2	LCX3	LCX4	LCX5	LCX6	LCX7
340	ANSX0	ANSX1	ANSX2	ANSX3	ANSX4	ANSX5	ANSX6	ANSX7
360	ANX0	ANX1	ANX2	ANX3	ANX4	ANX5	ANX6	ANX7
400		MPF	MPY			CMG		
420		UFM		DUFM		FCMG		DFCMG
440	SXL0	SXL1	SXL2	SXL3	SXL4	SXL5	SXL6	SXL7
460		FMP		DFMP				
500	RPL					BCD	DIV	DVF
520	RPT					FDI		DFDI
540	(SPRP0)	(SPRP1)	(SPRP2)	(SPRP3)	(SPRP4)	(SPRP5)	(SPRP6)	(SPRP7)
560	RPD					FDV		DFDV
600	TZE	TNZ	TNC	TRC	TMI	TPL		TTF
620	EAX0	EAX1	EAX2	EAX3	EAX4	EAX5	EAX6	EAX7
640	ERSX0	ERSX1	ERSX2	ERSX3	ERSX4	ERSX5	ERSX6	ERSX7
660	ERX0	ERX1	ERX2	ERX3	ERX4	ERX5	ERX6	ERX7
700	TSX0	TSX1	TSX2	TSX3	TSX4	TSX5	TSX6	TSX7
720	LXL0	LXL1	LXL2	LXL3	LXL4	LXL5	LXL6	LXL7
740	STX0	STX1	STX2	STX3	STX4	STX5	STX6	STX7
760	(LPRP0)	(LPRP1)	(LPRP2)	(LPRP3)	(LPRP4)	(LPRP5)	(LPRP6)	(LPRP7)

Operation Code Map Bit 27 = 0
(Left Half)

010		011	012	013	014	015	016	017
000		NOP	PULS1	PULS2		CIOC		
020			LDQC	ADL	LDAC	ADLA	ADLQ	ADLAQ
040	(ADWP0)	(ADWP1)	(ADWP2)	(ADWP3)	AOS	ASA	ASQ	SSCR
060		AWCA	AWCQ	LREG		ADA	ADQ	ADAQ
100		CWL				CMPA	CMPQ	CMPAQ
120						SBLA	SBLQ	SBLAQ
140	(ADWP4)	(ADWP5)	(ADWP6)	(ADWP7)	(SDBR)	SSA	SSQ	
160		SWCA	SWCQ	(LPRI)		SBA	SBQ	SBAQ
200		CMX	(ABSA)	(EPAQ)	SZNC	CNAA	CNAQ	CNAAQ
220	LBAR	RSW	(LDBR)	RMCM	SZN	LDA	LDQ	LDAQ
240	(SPRI0)	(SPBP1)	(SPRI2)	(SPBP3)	(SPRI)	ORSA	ORSQ	(LSDP)
260	(TSP0)	(TSP1)	(TSP2)	(TSP3)		ORA	ORQ	ORAQ
300	(EAWP0)	(EASP0)	(EAWP2)	(EASP2)		CANA	CANQ	CANAQ
320	(EAWP4)	(EASP4)	(EAWP6)	(EASP6)		LCA	LCQ	LCAQ
340	(EPP0)	(EPBP1)	(EPP2)	(EPBP3)	(STAC)	ANSA	ANSQ	(STCD)
360	(EPP4)	(EPBP5)	(EPP6)	(EPBP7)		ANA	ANQ	ANAQ
400		LDE		RSCR		ADE		
420	FSZN	FLD		DFLD		UFA		DUFA
440	STZ	SMIC	SCPR		STT	FST	STE	DFST
460	FSTR	FRD	DFSTR	DFRD		FAD		DFAD
500				FNEG		FCMP		DFCMP
520		NEG	(CAMS)	NEGL		UFS		DUFS
540	SBAR	STBA	STBQ	SMCM	STC1			(SSDP)
560				FNO		FSB		DFSB
600	(RTCD)			(RCU)	TEO	TEU	DIS	TOV
620	RET			(RCCL)	LDI	EAA	EAQ	LDT
640	(SPRI4)	(SPBP5)	(SPBI6)	(SPBP7)		ERSA	ERSQ	(SCU)
660	(TSP4)	(TSP5)	(TSP6)	(TSP7)	LCPR	ERA	ERQ	ERAQ
700	TRA			(CALL)		TSS	XEC	XED
720		ARS	QRS	LRS		ALS	QLS	LLS
740	STC2	STCA	STCQ	SREG	STI	STA	STQ	STAQ
760		ARL	QRL	LRL	GTB	ALR	QLR	LLR

- Operation Code Map Bit 27 = 0
(Right Half)

000		001	002	003	004	005	006	007
000 020 040 060	MVE CSL	 CSR			MVNE SZTL	 SZTR	 CMPB	
100 120 140 160	MLR SCD MVT	MRL SCDR			 SCM TCT	 SCMR TCTR	CMPC	
200 220 240 260			AD2D AD3D	SB2D SB3D			MP2D MP3D	DV2D DV3D
300 320 340 360	MVN	BTD		CMPN		DTB		 DVDR
400 420 440 460				 SAREG LAREG				 SPL LPL
500 520 540 560	A9BD S9BD ARA0 AAR0	A6BD S6BD ARA1 AAR1	A4BD S4BD ARA2 AAR2	ABD SBD ARA3 AAR3	 ARA4 AAR4	 ARA5 AAR5	 ARA6 AAR6	AWD SWD ARA7 AAR7
600 620 640 660	TRTN ARN0 NAR0	TRTF ARN1 NAR1	 ARN2 NAR2	 ARN3 NAR3	TMOZ ARN4 NAR4	TPNZ ARN5 NAR5	TTN ARN6 NAR6	 ARN7 NAR7
700 720 740 760	 SAR0 LAR0	 SAR1 LAR1	 SAR2 LAR2	 SAR3 LAR3	 SAR4 LAR4	 SAR5 LAR5	 SAR6 LAR6	 SAR7 LAR7

Operation Code Map Bit 27 = 1
(Left Half)

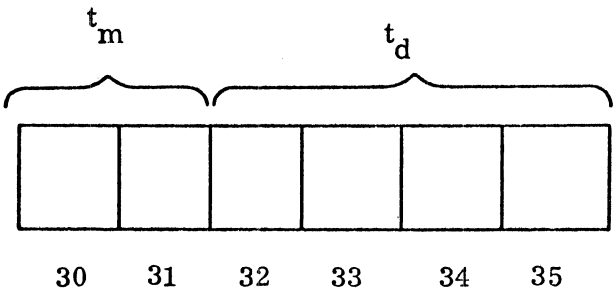
010		011	012	013	014	015	016	017
000 020 040 060								
100 120 140 160				(LPTR)	(SPTR)			
200 220 240 260	(SPBP0)	(SPRI1)	(LSDR) (SPBP2)	(SPRI3)	(SSDR)			(LPTP)
300 320 340 360	(EASP1) (EASP5) (EPBP0) (EPBP4)	(EAWP1) (EAWP5) (EPP1) (EPP5)	(EASP3) (EASP7) (EPBP2) (EPBP4)	(EAWP3) (EAWP7) (EPP3) (EPP7)				
400 420 440 460								
500 520 540 560			(CAMP)					(SPTP)
600 620 640 660	(SPBP4)	(SPRI5)	(SPBP6)	(SPRI7)				
700 720 740 760					(SRA) (LRA)		(SPTR)	

Operation Code Map Bit 27 = 1
(Right Half)

MODIFIERS FOR

00	n	20	n*	40	f1	60	*n
01	au	21	au*	41	itp	61	*au
02	qu	22	qu*	42		62	*qu
03	du	23		43	its	63	
04	ic	24	ic*	44	sd	64	*ic
05	al	25	al*	45	scr	65	*al
06	ql	26	ql*	46	f2	66	*ql
07	dl	27		47	f3	67	
10	0	30	0*	50	ci	70	*0
11	1	31	1*	51	i	71	*1
12	2	32	2*	52	sc	72	*2
13	3	33	3*	53	ad	73	*3
14	4	34	4*	54	di	74	*4
15	5	35	5*	55	dic	75	*5
16	6	36	6*	56	id	76	*6
17	7	37	7*	57	idc	77	*7

* Indirect



R RI IR

For t_m = 00, 01, 11

t_d	Mnemonic	Action
0000	n	y → Y
0001	AU	y + $C(A)_{0-17}$ → Y
0010	QU	y + $C(Q)_{0-17}$ → Y
0011	DU	y, 00...0 is the operand
0100	IC	y + C(IC) → Y
0101	AL	y + $C(A)_{18-35}$ → Y
0110	QL	y + $C(Q)_{18-35}$ → Y
0111	DL	0 ...0, y is the operand
1000	X0, 0	y + C(Xn) → Y
1001	X1, 1	
1010	X2, 2	
1011	X3, 3	
1100	X4, 4	
1101	X5, 5	
1110	X6, 6	
1111	X7, 7	

For t_m = 10

t_d	Mnemonic	Action
0000	F1	Fault 1 (processor fault).
0001	ITP	Indirect through pointer register.
0010	- -	--
0011	ITS	Indirect to segment.
0100	SD	Subtract Delta (from address field)
0101	SCR	Sequence Character Reversed.
0110	F2	Fault 2 (processor fault).
0111	F3	Fault 3 (processor fault).
1000	CI	Character from Indirect.
1001	I	Indirect.
1010	SC	Sequence Character.
1011	AD	Add delta (to address field).
1100	DI	Decrement address, increment tally.
1101	DIC	Decrement address, increment tally and continue (using modifier field of indirect word).
1110	ID	Increment address, decrement tally.
1111	IDC	Increment address, decrement tally, and continue.

Non Standard Modifiers

INST	TAG_8	MEANING
SCPR	01	store Fault register
	06	store Mode register
	20	Store Control Unit History Register
	40	Store Operations Unit History Register
	00	Store Appending Unit History Register
	60	Store Decimal Unit History Register
LCPR	02	Load Lock Up Fault Register
	04	Load Mode Register
	03	Zero Control Unit History Registers
		Zero Operations Unit History Registers
		Zero Appending Unit History Registers
		Zero Decimal Unit History Registers
	07	Set to one Central Unit History Register
		Set to one Operations Unit History Register
		Set to one Appending Unit History Register
		Set to one Decimal Unit History Register

System Modes

	GECOS OPERATIONAL		MULTICS OPERATIONAL			
	GECOS Master	GECOS Slave	MULTICS Absolute	MULTICS Privileged	MULTICS Nonprivileged	MULTICS BAR
Address Mode	Absolute			Append		
Absolute Indicator (ABS)	*2 1	*2 1	1	0	0	0
Master Mode Indicator	1	0	*3	*3	*3	*3
$\overline{\text{BAR}}$ Mode Indicator	*3	*3	1	1	1	0
PPR.P *1	*2	*2	1	1	0	1 or 0
H6000 M/M Instructions	OK	CMD FLT	OK	OK	IPR FLT Priv. Inst.	IPR FLT Priv. Inst.
MULTICS Privileged Instructions	IPR FLT (IOC)	IPR FLT (IOC)	OK	OK	IPR FLT Priv. Inst.	IPR FLT (IOC)
MULTICS Nonprivileged Instructions	IPR FLT (IOC)	IPR FLT (IOC)	OK	OK	OK	IPR FLT (IOC)
TSS Instructions	OK	TRA	OK	OK	OK	TRA
ITS Modifier	IPR FLT	IPR FLT	OK	OK	OK	IPR FLT
Instruction Bit 28	Inhibits	Inhibits	Inhibits	Inhibits	Ignored	Inhibits
LBAR	OK	CMD FLT	OK	OK	OK	IPR FLT Priv. Inst.

*1 Procedure Pointer Register Assembly - privileged bit

*2 Ignored - that is, stored as zero and not loaded to a zero.

*3 When operating in the GECOS Mode the indicator is defined as Master/Slave; while in the MULTICS Mode it is defined as the $\overline{\text{BAR}}$/BAR indicator.

MEMORY MAP FOR MULTICS SYSTEM

This document describes how primary memory will be utilized for the follow-on system. It describes how bootstrap1 and BOS lay out the data at bootload time and also how the memory is used during normal Multics operation and normal BOS operation. Reference is made to core blocks which are in units of 1024 word blocks of memory configured.

During its operation, bootstrap 1 lays out memory as shown below:

BLOCK NUMBER(S)	ADDRESSES	USE
0	0	INTERRUPT VECTOR
	100	FAULT VECTOR
	200	INTERRUPT TRANSFER PAIRS
	300	INTERRUPT SCU PAIRS
	400	FAULT TRANSFER PAIRS
	500	FAULT SCU PAIRS
	600	DN355 MAILBOX #1
	1020	PROCESSOR UTILITY SEGMENT
	1100	BULK STORE MAILBOX
	1200	IMW AREA
	1400	IOM MAILBOX #1

1	2000	IOM MAILBOX #2
	2400	DN355 MAILBOX #2
1	2620	UNUSED (CAN BE RESERVED FOR FUTURE EXPANSION OF IOM's AND DN355's.)
2	4000	BOS TOEHOLD
3	6000	CONFIGURATION DECK
4,5	10000	BOOTSTRAP1
6	14000	TEMPORARY DESCRIPTOR SEGMENT
7,8,	14000	SLT
9,10,11,12,13	22000	NAME TABLE
14	34000	PHYSICAL RECORD BUFFER
15→N-1	36000	PERM WIRED
N→M-1		FREE
M→n-1		PAGED SEGMENTS

Where N and M depend on the particular system being bootloaded.

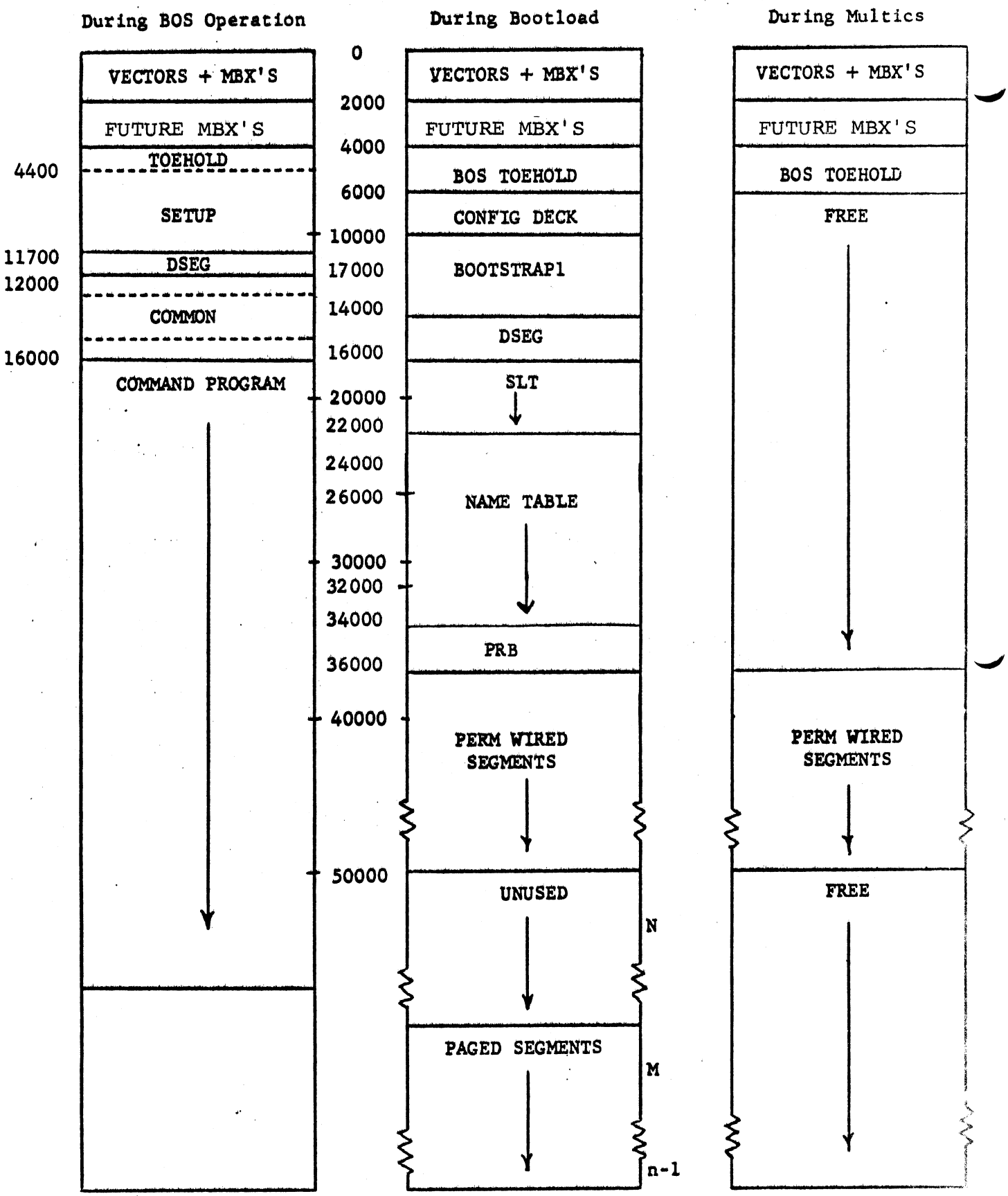
After initialization certain pages are freed (added to the paging pool) and the layout appears as follows:

0	0	AS ABOVE (MBX's + VECTORS)
1	2000	FREE (FOR PAGING)
2	4000	AS ABOVE (BOS)
3→14	6000	FREE (FOR PAGING)
15→N-1	36000	PERM WIRED
N→n-1		FREE (FOR PAGING)

When BOS is running, it utilizes memory as follows:

0	0	AS ABOVE (MBX's + VECTORS)
1	2000	UNUSED
2,3,4	4000	BOS TOEHOLD
	4400	SETUP (BOS MAIN CONTROL)
	11700	DESCRIPTOR SEGMENT
5,6	12000	BOS COMMON AREA
	12000	SAVED MACHINE CONDITIONS
	12400	BOS INTERNAL VARIABLES
	14000	CONFIGURATION DECK
7-22	16000	COMMAND PROGRAM

The following Diagrams are meant to describe the three memory layouts:


During BOS Operation
During Bootload
During Multics
VECTORS + MBX'S
FUTURE MBX'S
TOEHOLD
SETUP
DSEG
COMMON
COMMAND PROGRAM
4400
11700
12000
16000
0
2000
4000
6000
10000
17000
14000
16000
20000
22000
24000
26000
30000
32000
34000
36000
40000
50000
VECTORS + MBX'S
FUTURE MBX'S
BOS TOEHOLD
CONFIG DECK
BOOTSTRAP1
DSEG
SLT
NAME TABLE
PRB
PERM WIRED SEGMENTS
UNUSED
PAGED SEGMENTS
N
M
n-1
VECTORS + MBX'S
FUTURE MBX'S
BOS TOEHOLD
FREE
PERM WIRED SEGMENTS
FREE

INTERRUPT CELL ASSIGNMENTS

Decimal	Octal	DEVICE
0	0	DN355 #ø TERMINATE
1	1	DN355 #1 TERMINATE
2	2	BULK STORE CONTROLLER #0
3	3	
4	4	IOM #0 OVERHEAD
5	5	IOM #1 OVERHEAD
6	6	IOM #2 OVERHEAD
7	7	IOM #3 OVERHEAD
8	10	
9	11	
10	12	BULK STORE CONTROLLER #1
11	13	
12	14	IOM #0 TERMINATE
13	15	IOM #1 TERMINATE
14	16	IOM #2 TERMINATE
15	17	IOM #3 TERMINATE
16	20	DN355 #ø EMERGENCY
17	21	DN355 #1 EMERGENCY
18	22	SOFTWARE (SYSTEM TROUBLE)
19	23	
20	24	IOM #0 MARKER
21	25	IOM #1 MARKER
22	26	IOM #2 MARKER
23	27	IOM #3 MARKER
24	30	} STET
25	31	
26	32	SOFTWARE (Stop)
27	33	SOFTWARE (Quit)
28	34	IOM #0 SPECIAL
29	35	IOM #1 SPECIAL
30	36	IOM #2 SPECIAL
31	37	IOM #3 SPECIAL

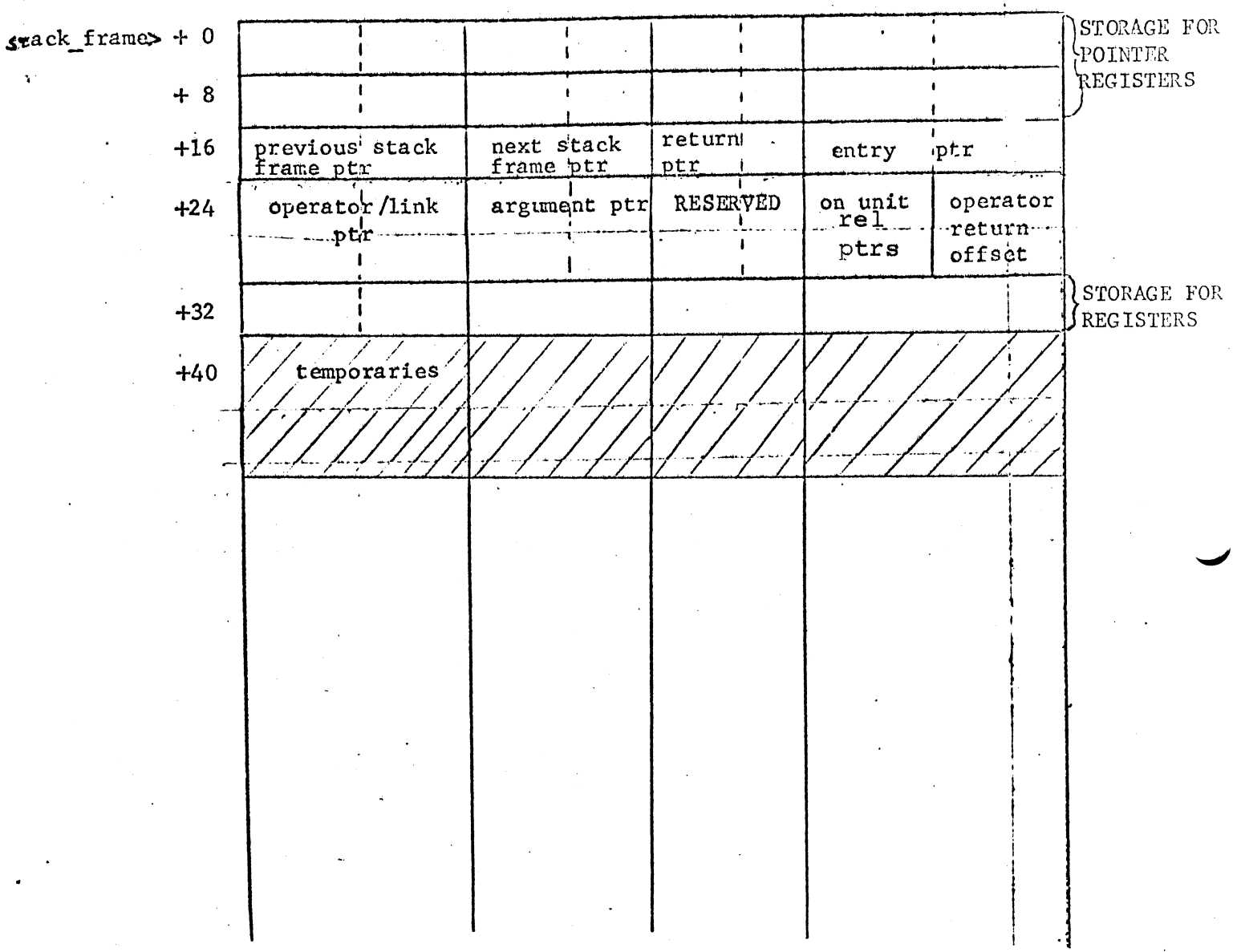
Multics IOM CHANNEL ASSIGNMENTS

CHANNEL NUMBER		DEVICE
Octal	Decimal	
10	8	DSU-270
12	10	Tape
15	13	Printer
16	14	Card Reader
17	15	Card Punch
20	16	Console
30 - 37	24 - 31	DSU-181

STACK HEADER

<stack> + 0		ring validation number	level	old lot ptr	
+ 8					
16	null	ptr	stack begin ptr	stack end ptr	lot ptr
24	signal ptr		process info ptr	pl1_operators_ ptr	call op ptr
32	push op ptr		return op ptr	short return op ptr	entry op ptr
40	RESERVED		RESERVED	RESERVED	RESERVED
48					

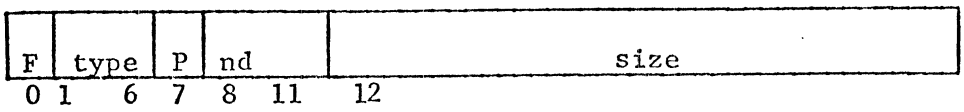
STACK FRAME FORMAT



ARGUMENT LIST FORMAT

The argument list must begin in an even word boundary. The pointers in the argument list need not be ITS pointers, however, they must be pointers which can be indirected through. Packed (unaligned) pointers cannot be used.

The i'th argument pointer points at the i'th argument directly. The i'th descriptor pointer points at the descriptor for the i'th argument. The format for a descriptor is as follows:



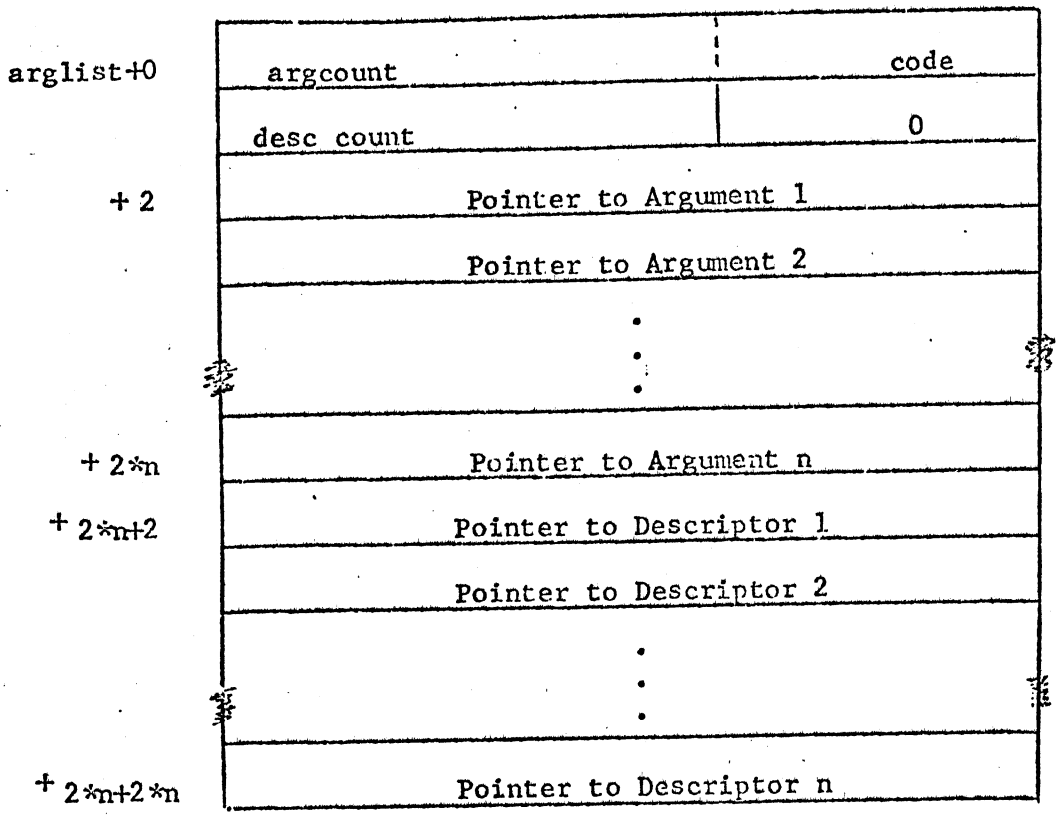
F is a flag specifying that this is a new type descriptor. It is a 1 if it is a PL/I version II descriptor and 0 for the old format descriptors.

type specifies the data type of the variable being described. The PL/I documentation contains a mapping of the actual codes used.

P indicates, if 1, that the data item is packed.

nd is the number of dimensions of an array. The array bounds follow the descriptor head in a format described in the PL/I documentation.

size holds the size (in bits or characters) of string data, the number of structure elements for structure data, or the scale and precision (as two, 12 bit fields) for arithmetic data.


arglist+0
argcount
code
desc count
0
+ 2
Pointer to Argument 1
Pointer to Argument 2
+ 2*n
Pointer to Argument n
+ 2*n+2
Pointer to Descriptor 1
Pointer to Descriptor 2
+ 2*n+2*n
Pointer to Descriptor n

PL1 Descriptors

A. Version 1 Descriptors

Value	Type
1	single precision real integer
2	double precision real integer
3	single precision real floating-point
4	double precision real floating-point
5	single precision complex integer (2 words)
6	double precision complex integer (4 words)
7	single precision complex floating-point (2 words)
8	double precision complex floating-point (4 words)
13	pointer data
14	offset data
15	label data
16	entry data
17-24	arrays of types 1-8
29-31	arrays of types 13-15
514	structure
518	area
519	bit string
520	character string
521	varying bit string
522	varying character string

data types which are not Multics standard

Value	Type	
523	array of structures	data types which are not Multics standard
524	array of areas	
525	array of bit strings	
526	array of character strings	
527	array of varying bit strings	
528	array of varying character strings	

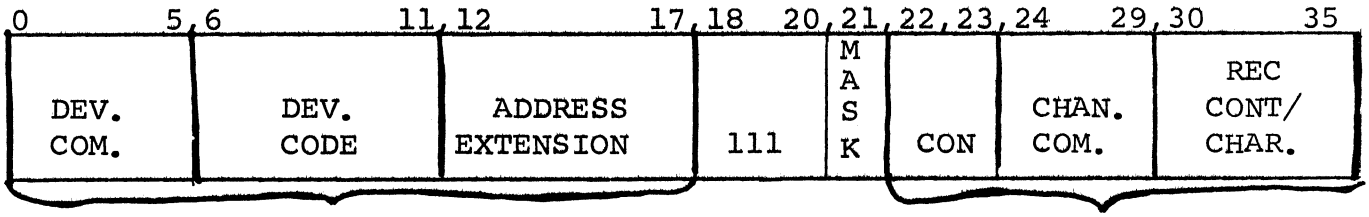
B. Version 2 Descriptor

Value	Type
1	real fixed binary short
2	real fixed binary long
3	real float binary short
4	real float binary long
5	complex fixed binary short
6	complex fixed binary long
7	complex float binary short
8	complex float binary long
9	real fixed decimal
10	real float decimal
11	complex fixed decimal
12	complex float decimal
13	pointer
14	offset

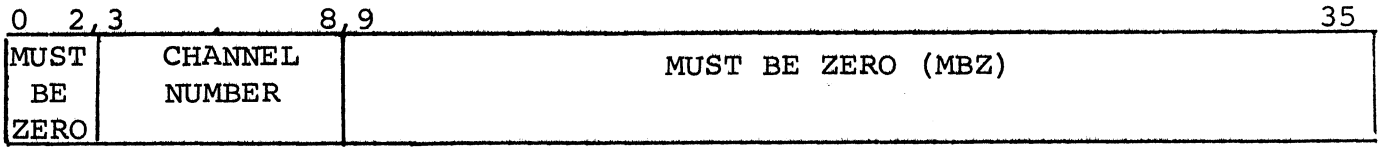
Value	Type
15	label
16	entry
17	structure
18	area
19	bit string
20	varying bit string
21	character string
22	varying character string
23	file

IOM DATA SHEET

PCW + IDCW



PCW (CP CH) + IDCW



PCW ODD WORD

DEV. COM. is the device command

DEV. CODE is the device code (MBZ in IDCW)

MASK if 1 in PCW, will turn off channel

if 1 in IDCW, will change address extension

CON 00 = terminate at end of I/O operation

10 = proceed at end of I/O operation

11 = send marker and proceed at end of I/O

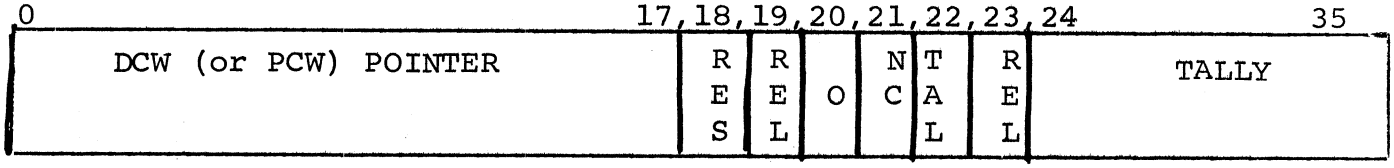
CHAN. COM. 00 = unit record transfer

02 = non-data transfer

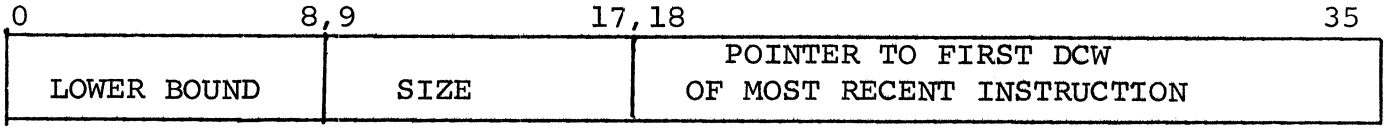
06 = multi-record transfer

10 = single character record

LPW



LPW



LPW Extension

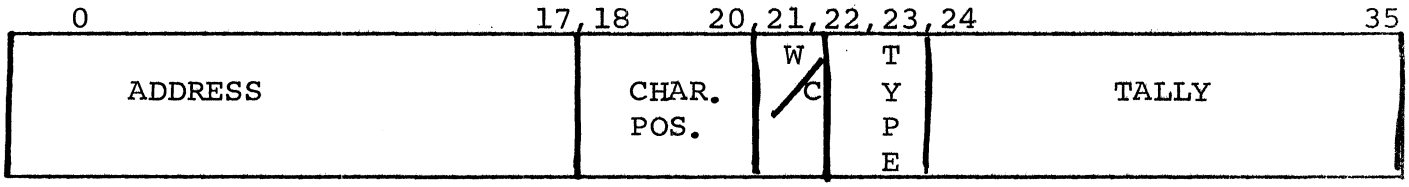
DCW POINTER — is the address of the DCW list

RES — = 1 to restrict use of IDCW

NC — = 1 to suppress updating of address and tally

TAL — = 1 to enable TRO fault

DCW

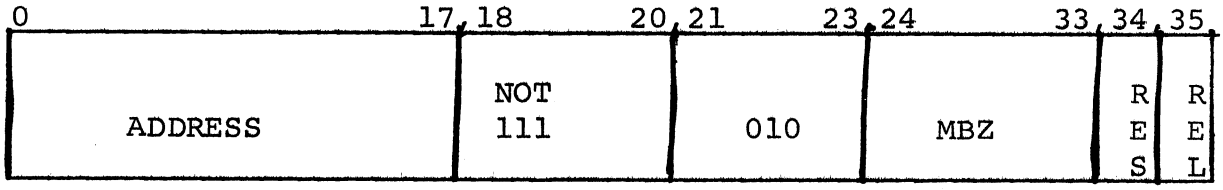


TYPE — IOTD (00), IOTP (01) IONTP (11)

CHAR. POS. — Character position (byte size is defined by the channel)

W/C — 0 = word tally
1 = character tally

TDCW

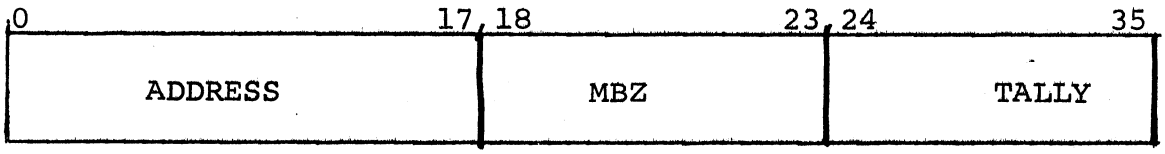


TDCW

RES is OR'ed into LPW RES

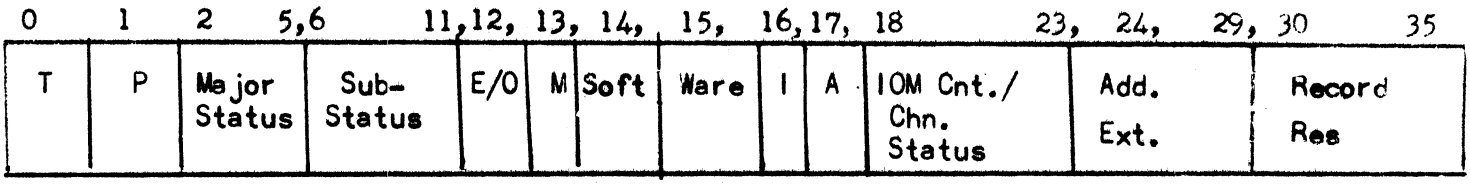
REL should be zero

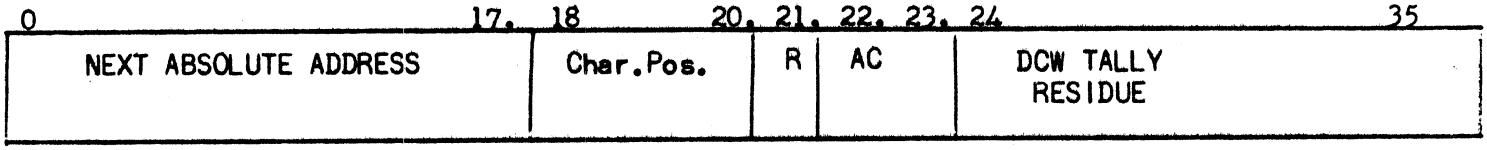
SCW



SCW

STATUS





T 1 = entry present (software can reset this).

P 1 = peripheral absent or power off bit.

M = 1 Marker Status

= 0 Terminate Status

E/O Even/Odd Data Transfer (PSIA)

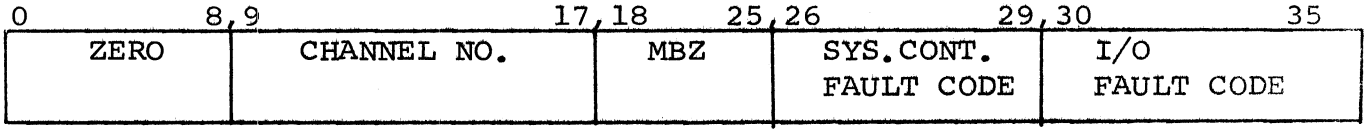
I is initiation bit

A is software abort bit (set to 0 by hardware)

R = 1 if device was reading

AC is TYPE of last DCW

SYSTEM FAULT WORD



CHANNEL STATUS

Bit 18	19	20	Cause
0	0	0	None
0	0	1	Unexpected PCW (connect while busy)
0	1	0	Illegal instruction to channel in PCW
0	1	1	Incorrect DCW, list service
1	0	0	Incomplete instruction sequence
1	0	1	Not used
1	1	0	Parity error, peripheral interface
1	1	1	Parity error, I/O Bus, data to channel

CENTRAL STATUS

Bit 21	22	23	Cause
0	0	0	None
0	0	1	LPW Tally Run-Out (Not connect channel)
0	1	0	2 TDCW's
0	1	1	Boundary error
1	0	0	None
1	0	1	IDCW in restricted mode
1	1	0	Character position/size discrepancy, list service
1	1	1	Parity error, I/O Bus, data from channel

SYSTEM CONTROLLER FAULT CODES

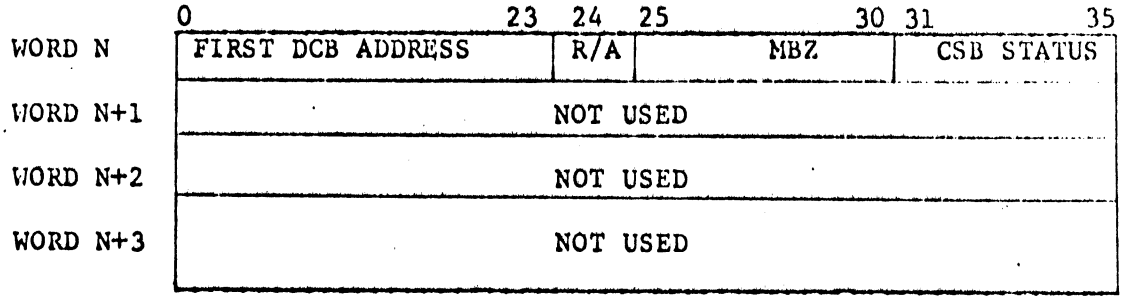
Bit	26	27	28	29	CAUSE
	0	0	0	0	None
	0	0	0	1	(Not used by system controller)
	0	0	1	0	Non-existent address
	0	0	1	1	Fault on condition
	0	1	0	0	(Not used)
	0	1	0	1	Data parity, store to SC
	0	1	1	0	Data parity, in store
	0	1	1	1	Data parity, store to SC and in store
	1	0	0	0	Not control port
	1	0	0	1	Port not enabled
	1	0	1	0	Illegal instruction
	1	0	1	1	Store not ready
	1	1	0	0	ZAC parity, active module to SC
	1	1	0	1	Data parity, active module to SC
	1	1	1	0	ZAC parity, SC to store unit
	1	1	1	1	Data parity, SC to store unit

I/O FAULT CODES

Bit	30	31	32	33	34	35	Cause
	0	0	0	0	0	0	None
	0	0	0	0	0	1	Illegal channel number
	0	0	0	0	1	0	Illegal service request
	0	0	0	0	1	1	Parity error, scratchpad read
	0	0	0	1	0	0	256K overflow
	0	0	0	1	0	1	LPW tally run-out, connect channel
	0	0	0	1	1	0	Control word not PCW, connect channel
	0	0	0	1	1	1	Character position field all one's, data service
	0	0	1	0	0	0	Character position/size discrepancy, data service
	0	0	1	0	0	1	No System Controller response
	0	0	1	0	1	0	Parity error, Core Store read
	0	0	1	0	1	1	Tally control error, Connect Channel (LPW has bits 21, 22=00)

BULK STORE INFORMATION

CSB FORMAT



CURRENT STATUS BLOCK (CSB)

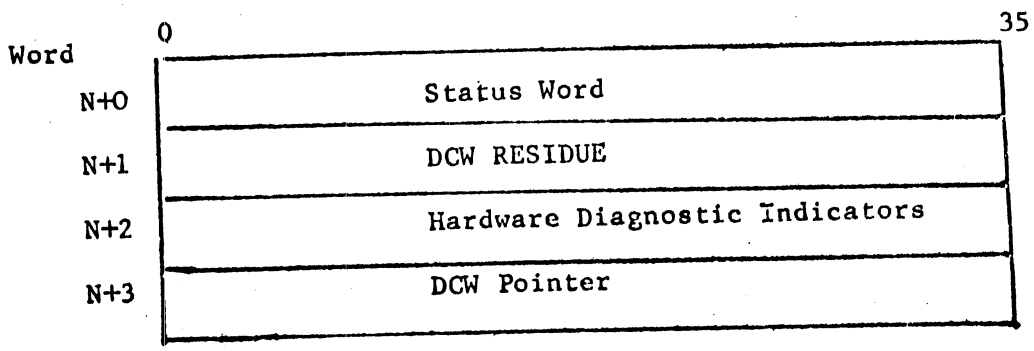
The CSB provides the location of the first DCB to be serviced and an indication of the operating status of the BSC. The fields in the CSB are described as follows:

Bits	Definition
0-23	First DCB Address - When the connect is issued to the BSC, this field will contain the address of the first DCB to be serviced. After operation of the BSC has started, this field will contain the address of the DCB currently being processed or, after operation has halted, of the last DCB processed before the BSC halted. The First DCB Address may be an absolute address or it may be relative to the base address of the mailbox area (from the switches). The two low order bits of this address must be zero. If the address is zero, the controller will stop operating and wait for the next connect strobe.

Bits	Application
25,26	Type code - set to match the DCW type code if a DCU was used. These bits will be zero if no DCW was used.

DCB Status Block

The DCB Status Block is a block of four words that are stored into main memory at a location specified by software. The other words will contain a DCW residue, a DCW pointer and one word of hardware diagnostic indicators.



DCB Status Block

DCW Residue

The DCW residue is composed of an address and a tally remainder. The address is a 24 bit main memory address which specifies the first word location in main memory after the last word to be successfully transferred. The tally specifies the number of words remaining to be transferred for that DCW.

Bits	Definition
24	Relative/Absolute - If this bit is a zero, the First DCB Address is relative to the base address of the mailbox. If this bit is a one, the First DCB address is an absolute main memory address.

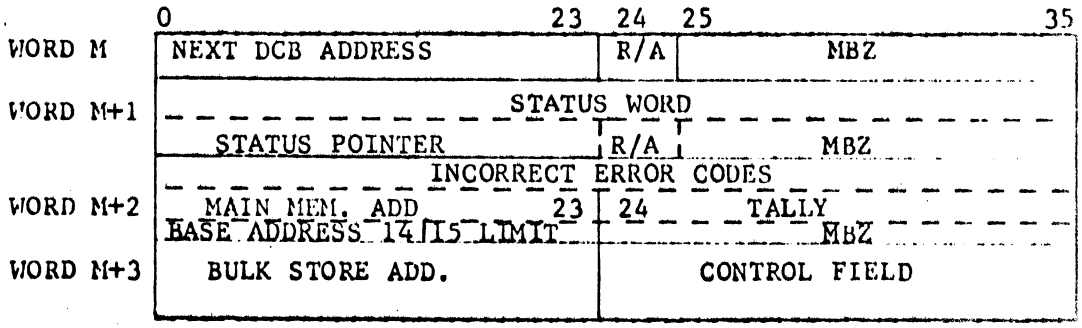
Bits	Definition
25-30	Not used, must be zero.
31-35	CSB Status -
31	Status Storage Error - This bit indicates that the BSC was unable to store status properly. The CSB indicates the DCB for which the status was to have been stored. This indicator applies to both the DCB status Block storage operation and to the single word status storage operation. The BSC has halted and reset the Busy Bit.
32	Next DCB Error - The BSC was not able to read the Next DCB Address in the DCB that the CSB is pointing to. The BSC is stopped (Busy Bit off).
33	Status Pointer Error - The BSC was unable to read and use a DCB status pointer. The CSB indicates the DCB for which status was to have been stored. The BSC has halted with the Busy Bit turned off.
34	Service Started - This bit is stored in the CSB by the BSC as soon as the BSC responds to a connect. It remains on until the service is completed, and the BSC resets it and the Busy Bit.
35	Busy Bit - This bit indicates the state of the BSC immediately after the BSC stores the CSB.

Bits	Definition
	If this bit is a ONE, then the controller is in the busy state. If this bit is a ZERO, the controller is stopped. The use of this bit is described in section 2.8.

Data Control Block (DCB)

The Data Control Block (DCB) is a block of four words which serves as the primary control element for data transfers. It is also the means by which software controls and monitors the Bulk Store Controller.

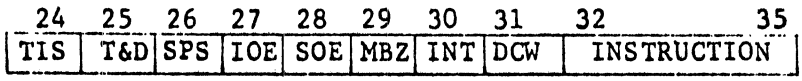
Contained in the Data Control Block are Bulk Store control words, a status word or status pointer and a pointer to the next DCB in the chain.



DATA CONTROL BLOCK

DCB Control Field

The control field contained in the fourth DCB word (Bulk Store Control Word) contains an instruction code and several control and instruction modifier bits. The format of this field is as follows:



The Instruction field contains the basic operation to be performed. The instructions and their codes are given in the following table.

BIT 32	33	34	35	INSTRUCTION
0	0	0	0	NOP
0	1	0	0	LOAD BASE & LIMIT
0	1	1	0	READ CONFIGURATION
0	0	1	0	LOAD CONFIGURATION
1	0	0	1	WRITE
1	0	0	0	WRITE ZEROS
1	0	1	1	WRITE & VERIFY
1	0	1	0	WRITE CONDITIONAL
1	1	0	1	READ
1	1	0	0	COMPARE
1	1	1	0	READ NONTRANSFER
0	1	0	1	POWER OFF ENABLE

Bulk Store Instruction Codes

The definitions of the instructions are given in the Instruction section. Unassigned instruction codes will cause the DCB operation to abort, with the storage of "Instruction Reject" status.

The other control field bits have definitions as stated below. All bits except the T&D bit (bit 25) use the basic instruction as defined. Special T&D operations will be defined in the T&D instructions section.

Bits	Definition
24	Tally Increment Selector - This bit selects the increment for the Tally field (bits 24-25) in the third DCB word. A zero in this bit gives an increment of 64 words. A one gives an increment of one word.
25	Test & Diagnostic - This bit is used to redefine the instruction code for test purposes. The redefined instructions are given in the T&D instruction section.
26	Status Pointer Selector - This bit is used to define the use of word one of the DCB. If this bit is a zero, status will be stored in the DCB word. If this bit is a one, the word will contain an address that will be used to store the DCB Status Block.

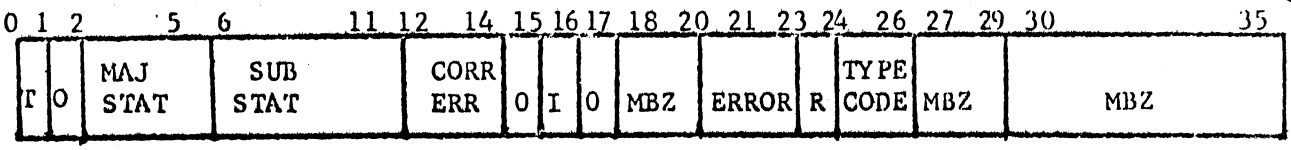
Bits	Definition
27	Interrupt on Error - A ONE in this position will cause the controller to generate a program interrupt after the DCB execution is completed, if the termination status for the operation is other than Subsystem Ready. This interrupt is generated after the DCB status word or block has been stored.
28	Stop on Error - If there is a ONE in this position and the DCB operation terminates in other than a Subsystem Ready condition, the controller will store status in the DCB status word or block and will halt, rather than continuing to the next DCB.
29	MUST BE ZERO - This bit is not assigned.
30	Interrupt - A ONE in this position will cause the controller to generate a program interrupt after this DCB operation has been completed and the DCB status stored.
31	Data Control Word - This bit is used to direct the Bulk Store Controller to a list of GECØS type one word DCW's. If this bit is a ONE, the Main Memory Address in the third word of the DCB will be used as the first DCW location. A ZERO will mean that the address is the first data transfer address.

DCB Status

At the completion of each DCB operation, status is stored to indicate the results of that operation. This status is stored as a single word or as a block of four words, depending on the setting of bit 26 in the DCB Control Field. The single status word and the first word of the status block will have identical formats.

Status Word

The format of the status word is shown in the following Figure.



This word is stored as the second DCB word or as the first word of the DCB status block. The intent of the structure of this word is to achieve as close a match as possible with the status word stored by the IOM.

The fields in the status word and their meanings are as follows:

Bits	Application
0	Terminate Bit - set to a one by the BSC to indicate that status has been stored.
1	Not used - set to zero
2-5	Major Status
6-11	Substatus - comparable to common peripheral major status and substatus.
12-14	Corrected Error Indicators - These bits are used to indicate hardware corrected error conditions.
15, 17, 30-35	Used by software. Set to zero by the BSC.
16	Initiate Interrupt Indicator - This bit is used for software compatability. It is present with and equivalent to the Command Reject Major Status. It will be set whether an interrupt is issued or not.
18-20 27-29	Not used, set to zero by the BSC
21-23	Error Indicators - used to indicate certain control sequence errors.
24	Read - set to one if the operation being performed was not a write (main memory to BSU).

MAJOR STATUS CODE	CONDITION	SUBSTATUS CODE
0000	Subsystem Ready	000000
0010	Subsystem Attention	
	BSU Address Not Present	000000
	No Response from BSU	000010
	Error Det. in BSU	000100
	Hardware Write Inhibited	000001
0011	Data Alert	
	Hardware Detected Data Error	00X010
	Hardware Detected Control Error	0010X0
	Failed to Compare	100000
	Write Verification Failed	010000
	Write Conditional Inhibited	000100
0100	End of File	000000
0101	DCB Command Reject	
	Invalid Instruction Code	000001
	Invalid BSU Address	000100
	Hardware Detected Control Error	0010X0
	Hardware Detected Data Error	00X010
1101	DCW Reject	
	Hardware Detected Control Error	0010X0
	Hardware Detected Data Error	00X010
	DCW Out of Bounds	000100
	Invalid DCW	000001

MAJOR STATUS & SUBSTATUS ASSIGNMENTS

DN355 INSTRUCTION CHART

TABLE 1
LISTING OF MACHINE MNEMONICS BY FUNCTIONAL CLASS

Octal Code	Store Reference	Instructions	Execution Time (Microseconds)
	Data Movement - Load		
07	LDA	Load A	2.0
47	LDQ	Load Q	2.0
04	LDAQ	Load AQ	2.4
43,03,41	LDXn	Load Xn (n = 1,2,3)	2.0
44	LDI	Load Indicator register	2.0
	Data Movement - Store		
17	STA	Store A	2.1
57	STQ	Store Q	2.1
14	STAQ	Store AQ	2.1
53,13,50	STXn	Store Xn (n = 1,2,3)	2.1
54	STI	Store Indicator register	2.1
56	STZ	Store Zero	2.1
	Arithmetic - Addition		
06	ADA	Add to A	2.0
46	ADQ	Add to Q	2.0
15	ADAQ	Add to AQ	2.4
42,02,40	ADCXn	Add character address to Xn	2.0
16	ASA	Add stored to A	2.5
76	AOS	Add one to storage	2.5
	Arithmetic - Subtraction		
26	SBA	Subtract from A	2.0
66	SBQ	Subtract from Q	2.0
24	SBAQ	Subtract from AQ	2.4
36	SSA	Subtract stored from A	2.5
	Arithmetic - Multiplication		
01	MPF	Multiply fraction	6.7
	Arithmetic - Division		
21	DVF	Divide fraction	7.5
	Boolean Operations		
34	ANA	AND to A	2.0
32	ANSA	AND to storage A	2.5
37	ORA	OR to A	2.0
72	ORSA	OR to storage A	2.5
35	ERA	EXCLUSIVE OR to A	2.0
62	ERSA	EXCLUSIVE OR to Storage A	2.5

	Comparison		
27	CMPA	Compare with A	2.0
67	CMPQ	Compare with Q	2.0
63,23,61	CMPXn	Compare with Xn	2.0
20	SZN	Set zero and negative indicators from memory	2.0
31	CANA	Comparative AND with A	2.0
	Transfer of Control		
71	TRA	Transfer unconditionally	1.0
10	TSY	Transfer and store (IC) in Y	2.2
74	TZE	Transfer on zero	1.0
64	TNZ	Transfer on not zero	1.0
75	TMI	Transfer on minus	1.0
65	TPL	Transfer on plus	1.0
45	TNC	Transfer on no carry	1.0
55	TOV	Transfer on overflow	1.0
	Input/Output		
60	CIOC	Connect input/output channel	3.3
30	LDEX	Load external (I/O) channel	3.0
70	STEX	Store external (I/O) channel	3.1

Octal Code	Nonstore Reference Instruction		Execution Time (Microsecond)
	GROUP 1		
	Immediate Load		
673	ILA	Immediate load A	1.3
473	ILQ	Immediate load Q	1.3
	Immediate Add		
773	IAA	Immediate add to A	1.3
573	IAQ	Immediate add to Q	1.3
173,273, 373	IACXn	Immediate add character address to Xn	1.3
	Immediate Boolean		
022	IANA	Immediate AND to A	1.3
122	IORA	Immediate OR to A	1.3
322	IERA	Immediate EXCLUSIVE OR to A	1.3
	Immediate Compare		
422	ICMPA	Immediate compare with A	1.3
222	ICANA	Immediate comparative AND to A	1.3

	Interrupt Control		
012	RIER	Read Interrupt Level Enable Register	2.2
412	RIA	Read Interrupt Address	2.5
052	SIER	Set Interrupt Level Enable Register	2.3
452	SIC	Set Interrupt Cells	2.5
	GROUP 2		
	Data Movement		
7333	CQA	Copy Q register into A register	1.3
6333	CAQ	Copy A register into Q register	1.3
4332, 0332,4333	CAXn	Copy A register into Xn register	1.3
2332, 3332,3333	CXnA	Copy Xn register into A register	1.3
	Interrupt Control		
3331	INH	Interrupt inhibit mode ON	1.3
7331	ENI	Interrupt enable mode ON	1.3
4331	DIS	Delay until interrupt	1.3
2331	NOP	No operation	1.3
	Shifts		
0337	ARS	A right shift	1.2 + 0.25N*
4337	QRS	Q right shift	
0335	LRS	Long right shift	
0336	ALS	A left shift	
4336	QLS	Q left shift	
0334	LLS	Long left shift	
2337	ARL	A right logic	
6337	QRL	Q right logic	

* N = Number of bits shifted

2335	LRL	Long right logic	
2336	ALR	A left rotate	
3336	ALP	A left parity rotate	
6336	QLR	Q left rotate	
7336	QLP	Q left parity rotate	
2334	LLR	Long left rotate	
1336	NRM	Normalize	1.4 + 0.35N
1334	NRML	Normalize long	1.4 + 0.25N*

Table 1. Instruction Repertoire by Functional Class

* N = Number of bits shifted

DN355 MEMORY MAP

Address (octal)	Interrupt Level (decimal)	Function
00000	0	Console — fault
001	1	Console — special (request)
002	2	Console — terminate
003	3	DIA-355 special #0
004	4	HLA-355 #1 — active subchannel 0
005	5	HLA-355 #1 — active subchannel 16
006	6	HLA-355 #1 — config. subchannel 0
007	7	HLA-355 #1 — config. subchannel 16
010	8	reserved for HLA-355
011	9	reserved for HLA-355
012	10	reserved for HLA-355
013	11	reserved for HLA-355
014	12	reserved for HLA-355
015	13	reserved for HLA-355
016	14	reserved for HLA-355
017	15	reserved for HLA-355
020	0	Card Reader — fault
021	1	Card Reader — special
022	2	Card Reader — terminate
023	3	DIA-355 Special #1
024	4	HLA-355 #1 — active subchannel 1
025	5	HLA-355 #1 — active subchannel 17
026	6	HLA-355 #1 — config. subchannel 1
027	7	HLA-355 #1 — config. subchannel 17
030	8	reserved for HLA-355 #2
031	9	reserved for HLA-355 #2
032	10	reserved for HLA-355 #2
033	11	reserved for HLA-355 #2
034	12	reserved for HLA-355 #3
035	13	reserved for HLA-355 #3
036	14	reserved for HLA-355 #3
00037	15	reserved for HLA-355 #3

DATANET 355 Interrupt Vectors

Address (octal)	Interrupt Level (decimal)	Function
00040	0	Line Printer: Fault
41	1	Line Printer: Special
42	2	Line Printer: Terminate
43	3	DIA-355 Special #2
44	4	HLA-355 #1: active subchannel 2
45	5	HLA-355 #1: active subchannel 18
46	6	HLA-355 #1: config. subchannel 2
47	7	HLA-355 #1: config. subchannel 18
48	8	reserved for HLA-355 #2
51	9	
52	10	
53	11	
54	12	reserved for HLA-355 #3
55	13	
56	14	
57	15	
60	0	
61	1	
62	2	
63	3	DIA-355 Special #3
64	4	HLA-355 #1: active subchannel 3
65	5	HLA-355 #1: active subchannel 19
66	6	HLA-355 #1: config. subchannel 3
67	7	HLA-355 #1: config. subchannel 19
70	8	reserved for HLA-355 #2
71	9	
72	10	
73	11	
74	12	reserved for HLA-355 #3
75	13	
76	14	
00077	15	

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00100	0	DIA-355 Fault
101	1	unassigned
102	2	DIA-355 terminate
103	3	DIA-355 Special #4
104	4	HLA-355 #1 — active subchannel 4
105	5	HLA-355 #1 — active subchannel 20
106	6	HLA-355 #1 — config. subchannel 4
107	7	HLA-355 #1 — config. subchannel 20
110	8	reserved for HLA-355 #2
111	9	reserved for HLA-355 #2
112	10	reserved for HLA-355 #2
113	11	reserved for HLA-355 #2
114	12	reserved for HLA-355 #3
115	13	reserved for HLA-355 #3
116	14	reserved for HLA-355 #3
117	15	reserved for HLA-355 #3
120	0	unassigned
121	1	unassigned
122	2	unassigned
123	3	DIA-355 Special #5
124	4	HLA-355 #1 — active 5
125	5	HLA-355 #1 — active 21
126	6	HLA-355 #1 — config. 5
127	7	HLA-355 #1 — config. 21
130	8	reserved for HLA-355 #2
131	9	reserved for HLA-355 #2
132	10	reserved for HLA-355 #2
133	11	reserved for HLA-355 #2
134	12	reserved for HLA-355 #3
135	13	reserved for HLA-355 #3
136	14	reserved for HLA-355 #3
00137	15	reserved for HLA-355 #3

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00140	0	HLA-355 #1 fault
141	1	unassigned
142	2	
143	3	DIA-355 Special #6
144	4	HLA-355 #1: active subchannel 6
145	5	HLA-355 #1: active subchannel 22
146	6	HLA-355 #1: config. subchannel 6
147	7	HLA-355 #1: config. subchannel 22
150	8	reserved for HLA-355 #2
151	9	
152	10	
153	11	
154	12	reserved for HLA-355 #3
155	13	
156	14	
157	15	
160	0	HLA-355 #2 fault
161	1	unassigned
162	2	
163	3	DIA-355 Special #7
164	4	HLA-355 #1: active subchannel 7
165	5	HLA-355 #1: active subchannel 23
166	6	HLA-355 #1: config. subchannel 7
167	7	HLA-355 #1: config. subchannel 23
170	8	reserved for HLA-355 #2
171	9	
172	10	
173	11	
174	12	reserved for HLA-355 #3
175	13	
176	14	
00177	15	

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00200	0	HLA-355 #3 fault
201	1	unassigned
202	2	unassigned
203	3	DIA-355 Special #8
204	4	HLA-355 #1: active subchannel 8
205	5	HLA-355 #1: active subchannel 24
206	6	HLA-355 #1: config. subchannel 8
207	7	HLA-355 #1: config. subchannel 24
210	8	reserved for HLA-355 #2
211	9	reserved for HLA-355 #2
212	10	reserved for HLA-355 #2
213	11	reserved for HLA-355 #2
214	12	reserved for HLA-355 #3
215	13	reserved for HLA-355 #3
216	14	reserved for HLA-355 #3
217	15	reserved for HLA-355 #3
00220	0	LLA-355 fault
221	1	active configuration
222	2	active configuration
223	3	ICA-355 Special #9
224	4	HLA-355: active subchannel 9
225	5	HLA-355: active subchannel 25
226	6	HLA-355: config. subchannel 9
227	7	HLA-355: config. subchannel 25
00230	8	reserved for HLA-355 #2
231	9	reserved for HLA-355 #2
232	10	reserved for HLA-355 #2
233	11	reserved for HLA-355 #2
234	12	reserved for HLA-355 #3
235	13	reserved for HLA-355 #3
236	14	reserved for HLA-355 #3
00237	15	reserved for HLA-355 #3

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00240	0	unassigned
241	1	
242	2	
243	3	DIA-355 Special #10
244	4	HLA-355 #1 — active subchannel 10
245	5	active subchannel 26
246	6	config. subchannel 10
247	7	config. subchannel 26
250	8	reserved for HLA-355 #2
251	9	
252	10	
253	11	
254	12	reserved for HLA-355 #3
255	13	
256	14	
257	15	
260	0	unassigned
261	1	
262	2	
263	3	DIA-355 Special #11
264	4	HLA-355 #1 — active subchannel 11
265	5	active subchannel 27
266	6	config. subchannel 11
267	7	config. subchannel 27
270	8	reserved for HLA-355 #2
271	9	
272	10	
273	11	
274	12	reserved for HLA-355 #3
275	13	
276	14	
00277	15	

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00300	0	unassigned
301	1	
302	2	
303	3	DIA-355 Special #12
304	4	HLA-355 #1 — active subchannel 12
305	5	HLA-355 #1 — active subchannel 28
306	6	HLA-355 #1 — config. subchannel 12
307	7	HLA-355 #1 — config. subchannel 28
310	8	reserved for HLA-355 #2
311	9	
312	10	
313	11	
314	12	reserved for HLA-355 #3
315	13	
316	14	
317	15	
320	0	unassigned
321	1	
322	2	
323	3	DIA-355 Special #13
324	4	HLA-355 #1 — active subchannel 13
325	5	HLA-355 #1 — active subchannel 29
326	6	HLA-355 #1 — config. subchannel 13
327	7	HLA-355 #1 — config. subchannel 29
330	8	reserved for HLA-355 #2
331	9	
332	10	
333	11	
334	12	reserved for HLA-355 #3
335	13	
336	14	
00337	15	

DATANET 355 Interrupt Vectors (cont.)

Address (octal)	Interrupt Level (decimal)	Function
00340	0	unassigned
341	1	
342	2	
343	3	DIA-355 Special #14
344	4	HLA-355 #1 — active subchannel 14
345	5	active subchannel 30
346	6	config. subchannel 14
347	7	config. subchannel 30
350	8	reserved for HLA-355 #2
351	9	
352	10	
353	11	
354	12	reserved for HLA-355 #3
355	13	
356	14	
357	15	
360	0	timer channel fault
361	1	interval timer runout
362	2	elapsed timer rollover
363	3	DIA-355 Special #15
364	4	HLA-355 #1 — active subchannel 15
365	5	active subchannel 31
366	6	config. subchannel 15
367	7	config. subchannel 31
370	8	reserved for HLA-355 #2
371	9	
372	10	
373	11	
374	12	reserved for HLA-355 #3
375	13	
376	14	
00377	15	

DATANET 355 Interrupt Vectors (cont.)

Processor Fault Vectors

Absolute Address (octal)	Function
0040	Power off
0041	Power on
0042	Memory Parity
0043	Illegal Operation Code
0044	Overflow
0045	Illegal Memory Operation
0046	Divide Check
0047	Illegal Program Interrupt

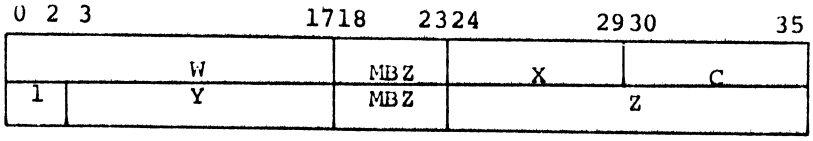
IOC Fault Status Locations

Address (octal)	Function
00420	I/O Chan. 0 (console)
1	I/O Chan. 1 (card reader)
2	I/O Chan. 2 (line printer)
3	Not used
4	I/O Chan. 4 ICA-355
5	Not used
6	I/O Chan. 6 (HLA-355)
00427	Not used (reserved for HLA-355 #2)
00430	Not used (reserved for HLA-355 #3)
1	I/O Channel 11_8 (LLA-355)
2	Not used (reserved for LLA-355 #2)
3	Not used (reserved for LLA-355 #3)
4	Not used (reserved for LLA-355 #4)
5	Not used (reserved for LLA-355 #5)
6	Not used (reserved for LLA-355 #6)
00437	Timer Channel

I/O Communications Region

Memory Address absolute octal	Function
0450 - 0451	interval timer, elapsed timer
0452 - 0453	unassigned
0454 - 0455	DIA-355 PCW
0456 - 0457	DIA-355 status ICW
0460 - 0461	Console — Status ICW
0462 - 0463	Console — Data ICW
0464 - 0465	Card Reader — Status ICW
0466 - 0467	Card Reader — Data ICW
0470 - 0471	Line Printer — Status ICW
0472 - 0473	Line Printer — Data ICW
0474 - 0477	unassigned
0500 - 0517	LLA-355 #1 control words
0520 - 0777	unassigned
1000 - 1777	HLA-355 #1 control words

DIA (DATANET 355) List Indirect Control Word (List ICW) FORMAT

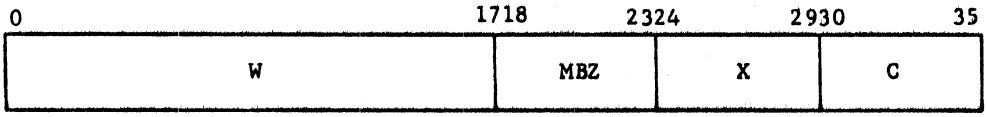


COMMAND	OPCODE (C-FIELD)	GE-600 ADDRESS FIELD (W-FIELD)	X-FIELD	DATANET 355 ADDRESS (Y-FIELD)	Z-FIELD
XEC 600	66	used for port select only	not used	loc for XEC data	not used
CON. 600	67	specifies addr sent with CON	not used	not used	not used
DISCONNECT	70	not used	not used	not used	not used
SXC 355 (level 3)	71	not used	specifies channel number	not used	not used
JUMP	72	not used	not used	specifies addr of new list ICW	not used
SXC 600	73	used for port sel. only	no. of int. cell to be set	not used	not used
CONFIG.	74	bits 16-17 used only	not used	1st loc of config. data	Tally
DATA XFER 355-600	75	600 starting address	not used	355 starting address	Tally
DATA XFER 600-355	76	600 starting address	not used	355 starting address	Tally
355 DATA WRAPAROUND	77	not used	not used	355 starting addr.	Tally (must be even)
READ CLEAR 600 OR TO STORE 355	65	600 starting address	not used	355 starting addr.	Tally

Zeroes ($W_{16 - 17}$=11)

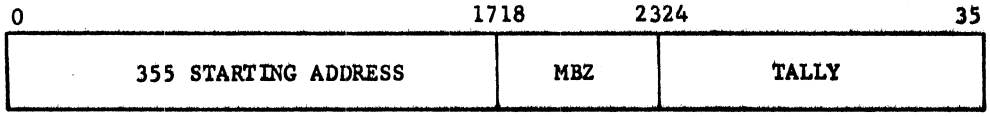
Format (loc Y): always stored as a word of zeroes.

DCW-DIA GE-600) Format

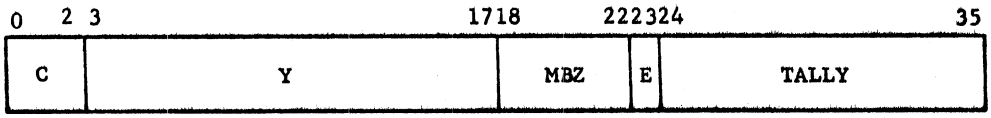


COMMAND	OPCODE (C-FIELD)	GE-600 ADDRESS FIELD (W)	X-FIELD
SXC 355 (level 3)	71	not used	cell no.
BOOTLOAD	72	specifies 600 "Boot ICW" address*	not used
SXC 600	73	specifies 600 address (used for port sel. only)	cell no.
TEST DATA XFER (355-600)	75	specifies 600 "Test ICW" address*	not used
TEST DATA XFER(600-355)	76	specifies 600 "Test ICW" address*	not used

* The "Test ICW" and "Boot ICW" has the following format:



DIA (DATANET 355) Status ICW



C field - 001 (indirect 36)

Y field - 355 store address where status will be stored

E bit - exhaust bit (set to one when tally exhausted)

Tally field - number of 36-bit words to be transferred.

The ICC DATANET 355 Status word has the following format:

BIT	FUNCTION (INDICATION)
0-13	MBZ
14	DIA instruction parity
15	355 parity error
16	MBZ
17	IOM powered up
18	No answer from 355
19	MBZ
20	illegal CON from 355
21	illegal opcode from 355
22	list ICW TRO
23	DCW did not specify 36 bit word
24	address < lower bound
25	address > upper bound
26	6000 write inhibit
27	6000 test command while busy
28	illegal opcode from 6000
29	No answer from processor
30	"E" bit set in list ICW
31	parity error (detected by DIA on 6000)
32, 33, 34	parity error in 6000
35	system fault in 6000 IOM

HIGH SPEED LINE ADAPTER

HLA-355 General Information

- I/O Channel = 6*
- Interrupt Vectors:

1. level 4 (all) - active subchannels 0-15
2. level 5 (all) = active subchannels 16-31
3. level 6 (all) - configuration subchannels 0-15
4. level 7 (all) - configuration subchannels 16-31

* hardware patchable

LOW SPEED LINE ADAPTER (LLA-355)

General Information

- I/O Channel (through LLA-355 channel card) -
  11 (octal) 9 (decimal)
- Interrupt Vectors (octal)
  active - 221
  configuration - 222
- Control Words (character control does not apply) (see Note 1)
  1. PCW's See page 220
  2. ICW's See page 221
- Character Control - does not apply
- Status (via status ICW's) (see Note 2)
- LLA-355 (On LSC-355) Control and Memory Map

ICW Address (octal)	Function
500 - 501	receive ICW (primary)
502 - 503	receive ICW (secondary)
504 - 505	send ICW (primary)
506 - 507	send ICW (secondary)
510 - 511	not used
512 - 513	not used
514 - 515	active
516 - 517	configuration status

Note 1: LLA-355 uses PCW0 and PCW1 only, as follows:
PCW0: bits 0-5 and 23 are read. Within command field (2-5), the LLA-355 will not recognize 4_8, 5_8, 11_8, 15_8, 16_8, 17_8.

PCW1: bits 0-5, 23, and 27-31 are read. Within command field (2-5), the LLA-355 will not recognize 4_8, 5_8, 11_8, 15_8, 16_8, 17_8.

Note 2: Status same as HLA-355 status, except as follows:

CONFIGURATION STATUS - only bits 0, 3-8, 15, 23, 28-35 are used. Channel type (bits 3-8) is coded 06 .

ACTIVE STATUS - only bits 0, 4-7, 10, 12, 18-20, 27-31 are used.

LSLA STATUS

BIT 0	ACTIVE STATUS 0 =send, 1 =receive	CONFIGURATION =1 (configuration)
1	*normal marker character received	sync. (if=1), async. (if=0)
2	*delayed marker character received	spare - reserved for subchannel
3	*terminate character received	
4	alternate buffer is active (if=1)	
5	switch buffers after status store (if=1)	
6	TYO (if = 1)	subchannel type
7	TY1 (if = 1)	
8	*lateral parity on received character	
9	*cmd. sent to unimplemented sub ch.	(spare)-reserved for subchannel
10	*change in date set status occurred	(spare)
11	(spare - MBZ)	(spare)
12	transfer timing error (if = 1)	check lat. parity receive (max. 8 bit code)
13	(spare)- reserved for subchannel (MBZ)	generate lat. send parity (codes 8 bits only)
14	(spare)- reserved for subchannel (MBZ)	lat. parity odd (if=1), even (if = 0)
15	No stop bit received	two send ICW's (if = 1)
16	DLO (data line occupied) (ACU)	use BAW (if = 1)
17	PWI (power indicator) (ACU)	(spare)
18	data set ready	(spare)-reserved for subchannel
19	clear to send	(spare)-reserved for subchannel
20	carrier detect	5 bit char. (asynchronous only)
21	supervisory receive	6 bit character
22	ACR (abandon call and retry)	7 bit character
23	data set status (ACU)	8 " "
24	ring indicator	use two stop bits (if=1)
25	line break	synchronous
26	(spare)	(spare) reserved for subchannel
27	receive mode	(spare) reserved for subchannel
28	send mode	(spare) reserved for subchannel
29	wraparound mode	
30	data terminal ready	
31	request to send	
32	make ready	See Configuration Status
33	supervisory transmit	
34	call request (ACU)	
35	(spare - reserved for subchannel)	
	* - used with receive status only	

COMMAND DESCRIPTION

The controller will provide the following command set for use by the External User Systems. The last column indicates those commands which will be rejected with invalid device code based on the incorrect use of the device code.

Table of commands valid for device/controller

COMMANDS	BIT CODE	OCTAL CODE	LEGAL DEVICE CODE
Seek	011 100	34	D
Special Seek (T&D)	011 110	36	D
Preseek	011 111	37	D
Restore	100 010	42	D
Read	010 101	25	D
Read ASCII	010 011	23	D
Write	011 001	31	D
Write ASCII	011 010	32	D
Write and Compare	011 011	33	D
Read Nonstandard Size	000 100	04	D
Read Track Header	010 111	27	D
Format Track	001 111	17	D
Request Status	000 000	00	E
Reset Status	100 000	40	E
Read Control Register	010 110	26	E
Write Control Register	001 110	16	D
Read Status Register	010 010	22	E
Read EDAC Register	010 001	21	D
Release	111 110	76	D
Reserve Device	111 111	77	D
Set Standby	111 010	72	D
Bootload CS	001 000	10	C
ITR Boot	001 001	11	C
Execute Device Command (DLI)	011 000	30	D

Special Controller Commands - see Section 3.21

D = Device Only
E = Either
C = Controller Only

COMMAND (BIT) STRUCTURE

High \ Low	000	001	010	011	100	101	110	111
000	Req. Status				Read Non-Standard Sector Size			
001	Boot C.S.	ITR BOOT					WCR Write Control Register	Format
010		Read EDAC RER	RSR	Read ASCII		Read	RCR	Read Header
011	Execute Device	Write	Write ASCII	Write & Compare	Seek		Special Seek	Pre-Seek
100	Reset Status		Re-store					
101								
110								
111			Set Standby				Release	Reserve Device

SPECIAL CONTROLLER COMMAND
(BIT)STRUCTURE

High \ Low	000	001	010	011	100	101	110	111
000	Suspend Controller		Read Controller Main ASCII		Read Lock Byte		Initiate Read Data	
001	Write Control Store		Write Controller Main ASCII		Write Lock Byte		Initiate Write Data	
010	Release Controller		Read Controller Main Binary					
011	Execute Control Store		Write Controller Main Binary		Conditional Write Lock Byte			
100								
101								
110								
111								

DATA ALERT	0011
Transmission Parity Alert	000010
Transfer Timing Alert	000001
Invalid Seek Address	000100
Header Verification Failure	0X1000
Cyclic Checkword Alert	X1X000
Compare Alert	1X0000
MPC DEVICE DATA ALERT	1011
Transmission Parity	000001
Inconsistent Command	000010
Sum Check Error	000011
Byte Locked Out	000100
Nonstandard Sector Size	010010
Sector Size Error	010001
EDAC Error Uncorrectable	001010
Error Correction Required	001001
END-OF-FILE	0100
Last Consecutive Block	0000X1
Block Count Limit	00001X
Defective Track (Alt. Track Assigned)	000100
Defective Track (No Alt. Assigned)	001000
Alternate Track Detected	010000
Good Track Detected	000000
ATTENTION	0010
Device Inoperable (Fault)	001000
Device Off-Line	100000
Seek Incomplete	000010
Write Inhibit	000001
Device Standby	010000

MPC DEVICE ATTENTION	1010
Configuration Error	000001
Device Number Error	000011
Multiple Device	000010
CA Error	001011
DEVICE BUSY	0001
Alternate Channel in Control	100000
Device Positioning	000000
CHANNEL READY	0000
No Substatus	000000
Retry was Performed	000001
Retry Twice	000010
Retry Three Times	000011
Device in T&D Mode	0010XX

	INSTRUCTION REJECTED	IDCW Parity Alert	Invalid Op Code	Invalid Device Code	Invalid Instruction Sequence	DATA ALERT	Transmission Parity Alert	Transfer Timing Alert	Invalid Seek Address	Header Verification Failure	Cyclic Checkword Alert	Compare Alert	END-OF-FILE	Last Consecutive Block	Sector Count Limit	Defective Track	Defective Track	Alternate Track	Good Track Detected	ATTENTION	Device Inoperable	Device Off-Line	Seek Incomplete	Write Inhibit	Device in Standby
SEEK	x		x	x		x			x											x	x	x			
SPECIAL SEEK	x		x	x		x			x											x	x	x	x		
PRESEEK	x		x	x	x	x			x											x	x	x	x		x
RESTORE	x		x	x																x	x	x			x
READ	x		x	x	x	x	x			x	x		x		x	x	x	x	x	x	x				x
READ ASCII	x		x	x	x	x	x			x	x		x		x	x	x	x	x	x	x	x			x
WRITE	x		x	x	x	x	x			x	x		x		x	x	x	x	x	x	x	x	x		x
WRITE ASCII	x		x	x	x	x	x			x	x		x		x	x	x	x	x	x	x	x	x		x
WRITE & COMPARE	x		x	x	x	x	x			x	x	x	x		x	x	x	x	x	x	x	x			x
READ NONSTANDARD	x		x	x	x	x	x			x	x									x	x	x			x
READ TRACK HEADER	x		x	x	x	x	x			x	x									x	x	x			x
FORMAT TRACK	x		x	x	x	x	x		x	x	x									x	x	x	x		x
REQUEST STATUS	x		x	x		x	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x		x
RESET STATUS	x		x	x																x	x	x	x		x
WCR WRITE CONTROL REG.	x		x	x		x														x	x				
RCR READ CONTROL REG.	x		x	x		x																			
RSR READ STATUS REG.	x		x	x		x														x	x				x
RER READ STATUS REG.	x		x	x		x																			
RELEASE DEVICE	x		x	x																x	x				x
RESERVE DEVICE	x		x	x																x	x				x
SET STANDBY	x		x	x																x	x				x
SPECIAL CONTROLLER CMDS	x		x	x																					
BOOTLOAD C.S.																									
ITR BOOT																									
EXECUTE DEVICE	x		x	x																					

	DEVICE BUSY Alternate Channel in Control	Device Positioning	CHANNEL READY No Substatus	Retry Was Performed	Retry Twice	Retry Three Times	T&D Mode	MPC COMMAND REJECT Illegal Procedure	Illegal Logical Chan. Number Illegal Chan. No. to Suspend	Continue Bit Not Set	MPC DEVICE DATA ALERT Transmission Parity	Inconsistent Command	Sumcheck Error	Byte Locked Out	Error Correction Required	EDAC Error Uncorrectable	Sector Size Error	Nonstandard Sector Size
SEEK	x		x	x	x	x			x									
SPECIAL SEEK	x		x	x	x	x			x									
PRESEEK	x		x	x	x	x			x									
RESTORE	x		x						x									
READ	x		x	x	x	x			x						x	x	x	x
READ ASCII	x		x	x	x	x			x						x	x		
WRITE	x		x	x	x	x			x									
WRITE ASCII	x		x	x	x	x			x									
WRITE & COMPARE	x		x	x	x	x			x									
READ NONSTANDARD	x		x	x	x	x												
READ TRACK HEADER	x		x	x	x	x			x						x	x		
FORMAT TRACK	x		x						x								x	x
REQUEST STATUS	x	x	x						x						x	x	x	x
RESET STATUS	x	x	x						x									
WCR WRITE CONTROL REG.			x						x									
RCR READ CONTROL REG.			x						x									
RSR READ STATUS REG.	x		x						x									
RER READ EDAC REG.			x						x									
RELEASE DEVICE	x		x						x									
RESERVE DEVICE	x		x						x									
SET STANDBY	x		x						x									
SPECIAL CONTROLLER CMDS			x					x	x	x	x	x	x	x				
BOOTLOAD C.S.			x						x									
ITR BOOT			x						x									
EXECUTE DEVICE			x															

	MPC DEVICE ATTENTION	Configuration Error	Device Number Error	Multiple Devices	CA Unexpected Interrupt	
SEEK			x	x	x	
SPECIAL SEEK			x	x	x	
PRESEEK			x	x	x	
RESTORE			x	x	x	
READ						
READ ASCII						
WRITE						
WRITE ASCII						
WRITE & COMPARE						
READ NONSTANDARD						
READ TRACK HEADER						
FORMAT TRACK						
REQUEST STATUS	x		x	x	x	
RESET STATUS	x		x	x	x	
WCR WRITE CONTROL REG.						
RCR READ CONTROL REG.						
RSR READ STATUS REG.						
RER READ EDAC						
RELEASE			x	x	x	
RESERVE DEVICE			x	x	x	
SET STANDBY			x	x	x	
SPECIAL CONTROLLER CMDS						
BOOTLOAD C.S.						
ITR BOOT						
EXECUTE DEVICE						

STATUS

Table 4 lists the possible status returns to the various instruction terminations. The major statuses and substatuses are described in detail in the paragraphs below:

CHANNEL READY (0000)

No Substatus (000000)

This status indicates that the controller/device is on line and ready and no detectable error conditions exist. When received at termination of a seek type instruction, this substatus indicates that the on-line portion has been successfully completed. The controller is free to receive another instruction for that channel. When automatic retry is performed by the controller, the binary number coded in the substatus is the number of times the retry was performed.

Automatic Retry was performed	(00X001)
Once	(00X010)
Twice	
Three Times	(00X011)
Device in T&D Mode	(0010XX)

DEVICE BUSY (0001)

The device busy status in response to an instruction, indicates that the addressed storage unit is operating but temporarily not available.

Alternate Channel in Control (100000)

This substatus is sent to the EUS which is attempting to gain control of the device or to a release, reserve. set standby, set local instruction directed to a device reserved by a different channel (dual channel operation).

Device Positioning (000000)

This substatus indicates that the addressed device was busy positioning. This status will be returned only for Request Status and Reset Status.

ATTENTION (0010)

The Attention status indicates that a condition has occurred which may require manual intervention.

Write Inhibit (0000X1)

This substatus is sent to the EUS as the result of a write type instruction being issued to a file that is write inhibited.

Seek Incomplete (00001X)

This substatus results when the access mechanism of the addressed device has failed to position after a determined period of time. The controller has attempted to correct the condition three times with restore and seek sequences. Unless there is a permanent hardware failure, this attention condition may be corrected by issuing additional Restore instructions to the addressed device from the EUS. (The restore must be followed by a Seek before device data transfer can occur.)

Device Off-Line (100000)

This substatus is sent to the EUS to indicate that the addressed device is off-line power off, or not connected to the subsystem. OPI is down from the device.

Device Inoperable (001000)

This substatus indicates that the device is on-line but is operating in a fault condition and must be investigated. The fault status indicates the detection of an illegal combination of conditions within the device.

Device in Standby (010000)

This substatus indicates that the device is in a standby state, that is, the spindle motor has been turned off and heads retracted.

DATA ALERT (0011)

The Data Alert status indicates that an error was detected during execution of the last instruction on that I/O channel. Any new instruction, except Request Status, will reset the Data Alert status.

Transfer Timing Alert (000001)

A Transfer Timing Alert during data transfer will occur when data is lost because some hardware stage of the data transfer logic cannot keep up the transfer rate.

Transmission Parity Alert (000010)

This condition is defined as occurring when a parity error is detected by the MPC during the transmission of data bytes from the EUS or to or from the device.

Invalid Seek Address (000100)

A Special Seek was issued to a user cylinder, or a normal seek was issued to the T&D cylinder. The seek address exceeds the limit of one device. If the format track verification data does not match the seek address. If less than five bytes or more than five bytes are transferred as seek data.

This status will be returned also when the check character test on the format instruction has a compare mismatch.

Header Verification Failure (0X1000)

This status is returned to the EUS when the subsystem is unable to locate the desired sector on the addressed track, or when a cylic check alert is detected when the controller reads the count field of a particular sector. Also, if a cyclic check error occurs as a result of reading Home Address (HA), this substatus shall be returned.

Check Character Alert (X1X000)

This substatus indicates that the field cyclic check character generated in the CA did not compare with the one recorded on the media. This status will be set for all check character failures, whether they occur on a count or data field or the home address. If the cyclic check occurs on the data field, the status returned will be (010000). For write and compare operations, the status will be (110000) if the compare operation failed in addition to the cyclic check error on the data field.

Since the controller checks the count field for each sector in all sector read and write operations (excluding format), it is possible to get a cyclic check condition on the count field. When this happens, the controller will return a status of (011000), which is a combination of header verification failure and check character alert.

Compare Alert (1X0000)

This substatus resulting from a write and compare instruction indicates that the data recorded on the media does not compare to data received from the EUS.

END OF FILE (0100)

End of file status indicates that the data transfer for the last instruction tried to exceed some defined boundary. This boundary may be a physical limit (end of cylinder), last block of an alternate track or a program-imposed limit (sector count limit). An EOF shall be indicated any time the track indicators change or upon entering or leaving a track formatted as alternate.

End-of-File - Good Track Detected (000000)

This status will be returned by the controller if the sector address in the count field is correct but the controller detected a Good Track Indicator when it was expecting an Alternate or Defective Track. This status will occur only when the EUS has issued a seek to a track with "good" track indicators when it was expecting something else.

Last Consecutive Block (0000X1)

This condition shall occur when the last consecutive block of the addressed actuator position has been reached and the operation presently being executed has not been completed, or when the controller detects an overflow from an alternate track.

Sector Count Limit (00001X)

This condition indicates that the total number of data sectors specified in the sector count limit were accessed but the processing system has not issued a Terminate Out Signal.

Defective Track Detected, Alternate Assigned (000100)

This condition shall occur when a read or write type instruction is attempted on a defective track or when overflow is detected to a defective track. The track indicators on the defective track are binary 10.

Defective Track Detected, No Alternate Assigned (001000)

This status is similar to the other Defective Track status except that the track indicators are binary 11. This indicates that no alternate has been assigned to the defective track.

Alternate Track Detected (010000)

This status will be returned if the sector address in the count field is correct but the controller detected an alternate track when a good or defective track was expected.

INSTRUCTION REJECTED (0101)

The Instruction Rejected status indicates that the instruction just received is not acceptable to the sub-system and was not initiated. The status is reset when any new instruction is received.

Invalid Operation Code (000001)

This substatus indicates that an operation code invalid for this subsystem was received from the EUS.

Invalid Device Code (000010)

This substatus indicates that an invalid device code for this subsystem was received from the EUS. Further, no device with the given address is connected to the subsystem. If the device is connected but does not have power on, the status will be Attention Device Off-Line or a controller only command (special controller, bootload, ITR boot) was received with a nonzero device code.

Parity Alert on IDCW (000100)

This substatus indicates that a parity error was detected by the LA on the IDCW.

Invalid Instruction Sequence (001000)

This condition shall occur when the subsystem receives a Data Transfer instruction without a prior valid seek on the same logical channel. Restore and Preseek are not valid seeks for data transfer.

CHANNEL BUSY (1000)

The channel busy status is reflected by the channel and indicates that the controller is in the process of executing an instruction from the EUS where the channel must be dedicated to the operation.

MPC STATUS EXTENSIONS

The following new major status categories are established, for use by MPC controllers:

MPC Device Attention	1010
MPC Device Data Alert	1011
MPC Command Reject	1101

These three new categories are to be considered extensions of the currently existing Attention, Data Alert and Command Reject major status classifications, and each will be used to reflect the same basic type of status as its previously defined counterpart.

Illegal Logical Chan Number 000010

Illegal logical channel number on all types of IDCW's.

Illegal Logical Chan Number to "Suspended" Controller 000011

When the controller is suspended and an IDCW is addressed to a logical channel other than the one over which the Suspend command is issued.

Continue Bit Not Set 000100

The first IDCW of a two-IDCW command does not have the continue bit set. (Special controller commands.)

CHANNEL ABORT - CONTROLLER ERROR INTERRUPT

The controller will detect internal hardware errors, such as parity errors on internal registers, parity errors on main memory (read/write) data, parity errors on microinstructions accessed from control store, etc.

The detection of any one of these internal hardware errors will result in the automatic execution of an error interrupt, which forces the controller to branch to a fixed control store location, and establishes an "error interrupt in progress" state of the machine.

The microprogram which is automatically entered as a result of the error interrupt will first test the state of Configuration Switch #14 of the Maintenance Panel. If the switch is set, an immediate branch will be made to the Integrated Test Routine (ITR) module, located in the first 512 locations of control store. (The exact entry point to be determined during implementation.) Setting of switch 14 therefore implies that the MPC controller will be put in the ITR mode upon the occurrence of an error interrupt.

The error interrupt occurrence will go undetected by the external system, however, if no logical channel of the PSI interface is active or initiated, during the time the "operational-in" line to the EUS is down.

As described above, execution of an error interrupt sequence will result in the safestoring of pertinent registers in main memory. This safestore area can be accessed by the external system for diagnostic purposes as required, using the "Read Controller Main Memory" special controller command.

MAJOR/SUBSTATUS STATUS PRIORITIES

The priority of major status returns will be as follows:

Instruction Rejected (Highest)	(0101)
MPC Command Reject	(1101)
Data Alert	(0011)
MPC Device Data Alert	(1011)
End-of-File	(0100)
Attention	(0010)
MPC Device Attention	(1010)
Device Busy	(0001)
Channel Ready (Lowest)	(0000)

The following table defines the codes and priorities for substatus within major status. Substatus is listed in order from the highest to the lowest priority.

INSTRUCTION REJECTED	0101
IDCW Parity Alert	000100
Invalid Op Code	000001
Invalid Device Code	000010
Invalid Instruction Sequence	001000
MPC COMMAND REJECT	1101
Illegal Procedure	000001
Illegal Logical Channel Number	000010
Illegal, Suspended	000011
Continue Bit Not Set	000100

Substatus under these new major status classifications will be encoded as one of 64 possible 6-bit codes (as opposed to bit encloding).

Substatus codes 00_8 - 07_8 under MPC Device Attention, MPC Device Data Alert and MPC Command Reject are reserved for status returns which are generated independent from the device personality of the controller.

MPC Device Attention (1010)

- Configuration Error (000001)
- Device Number Error (000011)
- Multiple Devices (000010)
- CA Unexpected Interrupt (after initialize) or CA OPI down (001011)

Configuration Switch Error

This error condition will occur if the personality firmware loaded into a controller does not agree with the configuration switches on the MPC operator panel.

Multiple Devices

This substatus indicates that the controller has detected at least two devices with the same identification number.

Device Number Error

At least one device has an identification number which is outside the range of legal device numbers for the subsystem.

CA Unexpected Interrupt

The configured switch error will probably be returned as termination status for the first command issued to the controller after the personality firmware has been loaded. The Multiple Devices substatus and Device Number Error substatus will most likely be returned as termination status for the first command issued to the controller following the loading of the personality firmware. However, either may occur at any point in time. Also, both may occur in which case the Device Number Error will take priority.

MPC Device Data Alert (1011)

Transmission Parity 000001

Parity error on data transmission.

Inconsistent Command 000010

All lock byte commands if illegal lock byte number is specified.

Sum Check Error 000011

Sum check error on data written to the controller.

Byte Locked Out 000100

Conditional write lock byte is referenced to a lock byte which is nonzero (locked).

Error Correction Required 001001

EDAC Error Uncorrectable 001010

Sector Size Error 010001

Nonstandard Sector Size 010010

MPC Command Reject (1101)

Illegal Procedure 000001

Write controller main memory and write control store when the controller is not in "Suspend" mode.

Execute control store microprogram and initiate read/write data transfer IDCW not preceded by special controller command IDCW.

If switch 14 is not set, the error interrupt microprogram will take the following actions:

a. Safestore in a fixed main memory area (to be defined) the current contents of all pertinent hardware registers.

b. Terminate all existing activity in progress, including device movement. This must be evaluated and implemented on a device type basis. The execution of an active channel program (DCW list) will be aborted.

c. Reset the error interrupt level, and return the controller to normal operation (that is, waiting for command from central system).

The occurrence of the error interrupt will force the "operational-in" line of the PSI interface to the external system to revert to the "nonoperational" state. This line will stay in the "nonoperational" state until the "error interrupt in progress" state is reset by the microprogram.

Detection of Error Interrupt Sequence by External User System

The external indication that the controller has taken an error interrupt is the dropping of the "operational-in" line of the PSI interface to the external system. This line will be down for the duration of time the controller is in the "error interrupt in progress" state, which is a function of how many microinstructions must be executed by the error interrupt microroutine. As an order of magnitude, it can be assumed the "operational-in" line will be down for approximately 15 microseconds. Repeated error detections will result in repeated error interrupts, and the PSI adapter in the IOM will detect the PSI "operational-in" line dropping if a channel program (DCW list) is currently being executed for any one of the eight logical channels of the PSI. The adapter will generate a status storage, and mask the logical channel.

ASCII CHARACTER SET ON MULTICS

	0	1	2	3	4	5	6	7
000								BEL
010	BS	HT	NL	VT	NP		RRS	BRS
020			HLF		HLR			
030				MC				
040	Space	!	"	#	$	%	&	'
050	(	)	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7
070	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G
110	H	I	J	K	L	M	N	O
120	P	Q	R	S	T	U	V	W
130	X	Y	Z	[	\	]	^	_
140	`	a	b	c	d	e	f	g
150	h	i	j	k	l	m	n	o
160	p	q	r	s	t	u	v	w
170	x	y	z	{	|	}	~	

Multics Definitions:

NL	New Line (carriage return and line feed)
HLF	Half-Line Forward Feed
HLR	Half-Line Reverse Feed
RRS	Red Ribbon Shift
BRS	Black Ribbon Shift
MC	Mode Change
NP	New Page

MULTICS CHARACTER SET - BCD

Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code	Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code
0	00 0000	00	0	↑	10 0000	40	11-0
1	00 0001	01	1	J	10 0001	41	11-1
2	00 0010	02	2	K	10 0010	42	11-2
3	00 0011	03	3	L	10 0011	43	11-3
4	00 0100	04	4	M	10 0100	44	11-4
5	00 0101	05	5	N	10 0101	45	11-5
6	00 0110	06	6	O	10 0110	46	11-6
7	00 0111	07	7	P	10 0111	47	11-7
8	00 1000	10	8	Q	10 1000	50	11-8
9	00 1001	11	9	R	10 1001	51	11-9
[	00 1010	12	2-8	-	10 1010	52	11
#	00 1011	13	3-8	$	10 1011	53	11-3-8
@	00 1100	14	4-8	*	10 1100	54	11-4-8
:	00 1101	15	5-8	)	10 1101	55	11-5-8
>	00 1110	16	6-8	;	10 1110	56	11-6-8
?	00 1111	17	7-8	'	10 1111	57	11-7-8
↑	01 0000	20	(blank)	+	11 0000	60	12-0
A	01 0001	21	12-1	/	11 0001	61	0-1
B	01 0010	22	12-2	S	11 0010	62	0-2
C	01 0011	23	12-3	T	11 0011	63	0-3
D	01 0100	24	12-4	U	11 0100	64	0-4
E	01 0101	25	12-5	V	11 0101	65	0-5
F	01 0110	26	12-6	W	11 0110	66	0-6
G	01 0111	27	12-7	X	11 0111	67	0-7
H	01 1000	30	12-8	Y	11 1000	70	0-8
I	01 1001	31	12-9	Z	11 1001	71	0-9
&	01 1010	32	12	←	11 1010	72	0-2-8
.	01 1011	33	12-3-8	,	11 1011	73	0-3-8
]	01 1100	34	12-4-8	%	11 1100	74	0-4-8
(	01 1101	35	12-5-8	=	11 1101	75	0-5-8
<	01 1110	36	12-6-8	"	11 1110	76	0-6-8
\	01 1111	37	12-7-8	!	11 1111	77	0-7-8

OCTAL/DECIMAL CONVERSIONS

Octal	10000	20000	30000	40000	50000	60000	70000
Decimal	4096	8192	12288	16384	20480	24576	28672

Octal	100000	200000	300000	400000	500000	600000	700000	1000000
Decimal	32768	65536	98304	131072	163840	196608	229376	262144

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

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

Octal	0400 to 0777
Decimal	0256 to 0511

Octal	0	1	2	3	4	5	6	7
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	1400 to 1777
Decimal	0768 to 1023

Octal	0	1	2	3	4	5	6	7
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

Occal	10000	20000	30000	40000	50000	60000	70000
Decimal	4096	8192	12288	16384	20480	24576	28672

Octal	100000	200000	300000	400000	500000	600000	700000	1000000
Decimal	32768	65536	98304	131072	163840	196608	229376	262144

Octal	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

Octal	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

Octal	2400 to 2777
Decimal	1280 to 1535

Octal	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

Octal	3400 to 3777
Decimal	1792 to 2047

Octal	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

Octal	10000	20000	30000	40000	50000	60000	70000
Decimal	4096	8192	12288	16384	20480	24576	28672

Octal	100000	200000	300000	400000	500000	600000	700000	1000000
Decimal	32768	65536	98304	131072	163840	196608	229376	262144

Octal	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	2354	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

Octal	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

Octal	6400 to 6777
Decimal	3328 to 3583

Octal	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

Octal	7400 to 7777
Decimal	3840 to 4095

Octal	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

Octal	10000	20000	30000	40000	50000	60000	70000
Decimal	4096	8192	12288	16384	20480	24576	28672

Octal	100000	200000	300000	400000	500000	600000	700000	1000000
Decimal	32768	65536	98304	131072	163840	196608	229376	262144

Octal	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

Octal	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

Octal	4400 to 4777
Decimal	2304 to 2559

Octal	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

Octal	5400 to 5777
Decimal	2816 to 3071

Octal	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071