SERIES 60 (LEVEL 68)

MULTICS HARDWARE AND SOFTWARE FORMATS PROGRAM LOGIC MANUAL ADDENDUM A

SUBJECT

Changes and additions to the Multics Hardware and Software Formats

SPECIAL INSTRUCTIONS

This Program Logic Manual (PLM) describes certain internal modules constituting the Multics System. It is intended as a reference for only those who are thoroughly familiar with the implementation details of the Multics operating system; interfaces described herein should not be used by application programmers or subsystem writers; such programmers and writers are concerned with the external interfaces only. The external interfaces are described in the *Multics Programmers' Manual, Commands and Active Functions* (Order No. AG92), *Subroutines* (Order No. AG93), and *Subsystem Writer's Guide* (Order No. AK92).

This is the first addendum to AN87 Revision 0 dated July 1976. Change bars in the margin indicate technical changes and additions; asterisks denote deletions.

As Multics evolves, Honeywell will add, delete, and modify module descriptions in subsequent PLM updates. Honeywell does not ensure that the internal functions and internal module interfaces will remain compatible with previous versions.

ORDER NUMBER

AN87-00A

March 1980

27079
3C380
Printed in U.S.A.

Honeywell

COLLATING INSTRUCTIONS

To update the manual, remove old pages and insert new pages as follows:

Remove	Insert
iii through vi	iii through vi
1-51, 1-52	1-51, 1-52
1-55, 1-56	1-55, 1-56
	6-5.1, 6-5.2
6-21, 6-22	6-21, 6-22 6-22.1, blank
7-5 through 7-8	7-5 through 7-9, blank
C-1, C-2	C-1, C-2

File No.: 2L13

AN87A

Honeywell

MULTICS HARDWARE AND SOFTWARE FORMATS PROGRAM LOGIC MANUAL

SERIES 60 (LEVEL 68)

RESTRICTED DISTRIBUTION

SUBJECT:

Detailed Formats of Information Read or Shared by Hardware and Formats Used by Multics Software.

SPECIAL INSTRUCTIONS:

This Program Logic Manual (PLM) describes certain internal modules constituting the Multics System. It is intended as a reference for only those who are thoroughly familiar with the implementation details of the Multics operating system; interfaces described herein should not be used by application programmers or subsystem writers; such programmers and writers are concerned with the external interfaces only. The external interfaces are described in the Multics Programmers' Manual, Commands and Active Functions (Order No. AG92), Subroutines (Order No. AG93), and Subsystem Writers' Guide (Order No. AK92).

As Multics evolves, Honeywell will add, delete, and modify module descriptions in subsequent PLM updates. Honeywell does not ensure that the internal functions and internal module interfaces will remain compatible with previous versions.

This PLM is one of a set which, when complete, will supersede the System Programmers' Supplement to the Multics Programmers' Manual (Order No. AK96).

DATE:

July 1976

ORDER NUMBER:

AN87, Rev. 0

PREFACE

Multics Program Logic Manuals (PLMs) are intended for use by Multics system maintenance personnel, development personnel, and others who are thoroughly familiar with Multics internal system operation. They are not intended for application programmers or subsystem writers.

The PLMs contain descriptions of modules that serve as internal interfaces and perform special system functions. These documents do not describe external interfaces, which are used by application and system programmers.

Since internal interfaces are added, deleted, and modified as design improvements are introduced, Honeywell does not ensure that the internal functions and internal module interfaces will remain compatible with previous versions. To help maintain accurate PLM documentation, Honeywell publishes a special status bulletin containing a list of the PLMs currently available and identifying updates to existing PLMs. This status bulletin is distributed automatically to all holders of the System Programmers' Supplement to the Multics Programmers' Manual (Order No. AK96) and to others on request. To get on the mailing list for this status bulletin, write to:

Large Systems Sales Support
Multics Project Office
Honeywell Information Systems Inc.
Post Office Box 6000 (MS A-85)
Phoenix, Arizona 85005

File No.: 2L13

AN87

This PLM contains the detailed formats of status words, control unit data, fault codes, and other information read or shared by hardware. It also includes the formats peculiar to the Multics software environment, such as stack frame formats. In addition, octal masks are included in the detailed description of several formats as a further aid to the reader.

This manual should probably be used in conjunction with the System Dump Analysis PLM, Order No. AN53, when analyzing dumps or system problems.

Many acronyms are used throughout this manual. The acronym, FNP, is used to mean either the DATANET 355 Front-End Network Processor or the DATANET 6600 Front-End Network Processor. (Their use with the Multics system is completely interchangeable.) Several acronyms are defined in text; all of them are defined in Appendix A for the reader's convenience.

CONTENTS

ILLUSTRATIONS

TABLES

SECTION I

LEVEL 68 PROCESSOR

This section describes the registers and register data formats (as seen in an octal store dump) of the program-accessible registers of the Level 68 processors. Only very brief discussions of the roles of the various features in the operation of the processor are given. See Multics Processor Reference Manual, Order No. AL39 for a complete description of the processor operation.

CONTROL UNIT AND OPERATIONS UNIT FORMATS

The control unit (CU) is that portion of the processor hardware that decodes instruction opcodes, loads and unloads registers, responds to internal and external hardware signals, and interfaces with the main store. The operations unit (OU) is that portion of the processor hardware that executes the binary word mode or basic instructions.

Processor Instructions

The basic instruction word format is shown below in Figure 1-1. The PL/I declaration (and the name of the include file) is given below. The following pages contain the instruction opcode charts (Tables 1-1 and 1-2) and an alphabetic list of the processor instructions (Table 1-3).

PL/I Declaration (db_inst.incl.pl1)

```
dcl     1 instr       based (ilc_ptr)   aligned,
         (2 offset    fixed bin (17),
          2 opcode    bit(10),
          2 inhibit   bit(1),
          2 pr_bit    bit(1),
          2 tag       bit(6))           unaligned;

dcl     1 instr_pr    based (ilc_ptr)   aligned,
         (2 pr        bit(3),
          2 offset    fixed bin (14),
          2 pad       bit(18))          unaligned;
```

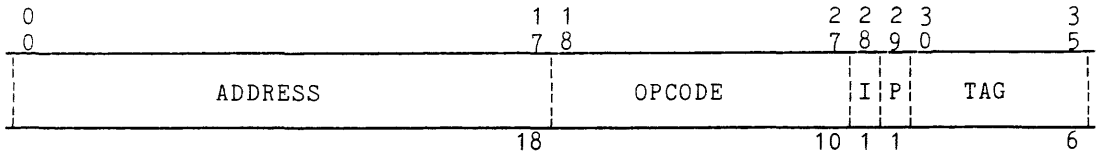


Figure 1-1. Basic Instruction Word Format

Legend:

ADDRESS For P=0; 18-bit procedure segment address.
For P=1; 3-bit pointer register number and 15-bit signed word offset.

OPCODE Instruction operation code.

I Interrupt inhibit bit.

P Pointer register flag.

TAG Instruction address modifier.

Explanation of Opcode Notation

The opcode field of the instruction is 10 bits long (bits 18-27). The normal notation is to express the first nine bits (bits 18-26) in octal followed by either "(1)" or "(0)" for the tenth bit (bit 27). For example, lda is expressed as 235(0), which is 0100111010 in the opcode field, and mlr is expressed as 100(1), which is 0010000001. The two opcode charts on the following pages divide the instruction set into two groups: those with bit 27 equal to 0 and those with bit 27 equal to 1.

Table 1-1. Instruction Opcode Chart, Bit 27 = 0

	000	001	002	003	004	005	006	007
000		mme1	drl		mme2	mme3		mme4
010		nop	puls1	puls2		cioc		
020	adlx0	adlx1	adlx2	adlx3	adlx4	adlx5	adlx6	adlx7
030			ldqc	adl	ldac	adla	adlq	adlaq
040	asx0	asx1	asx2	asx3	asx4	asx5	asx6	asx7
050	adwp0	adwp1	adwp2	adwp3	aos	asa	asq	sscr
060	adx0	adx1	adx2	adx3	adx4	adx5	adx6	adx7
070		awca	awcq	lreg		ada	adq	adaq
100	cmpx0	cmpx1	cmpx2	cmpx3	cmpx4	cmpx5	cmpx6	cmpx7
110		cwl				cmpa	cmpq	cmpaq
120	sblx0	sblx1	sblx2	sblx3	sblx4	sblx5	sblx6	sblx7
130						sbla	sblq	sblaq
140	ssx0	ssx1	ssx2	ssx3	ssx4	ssx5	ssx6	ssx7
150	adwp4	adwp5	adwp6	adwp7	sdbr	ssa	ssq	
160	sbx0	sbx1	sbx2	sbx3	sbx4	sbx5	sbx6	sbx7
170		swca	swcq	lpri		sba	sbq	sbaq
200	cnax0	cnax1	cnax2	cnax3	cnax4	cnax5	cnax6	cnax7
210		cmk	absa	epaq	sznc	cnaa	cnaq	cnaaq
220	ldx0	ldx1	ldx2	ldx3	ldx4	ldx5	ldx6	ldx7
230	lbar	rsw	ldbr	rmcm	szn	lda	ldq	ldaq
240	orsx0	orsx1	orsx2	orsx3	orsx4	orsx5	orsx6	orsx7
250	spri0	spbp1	spri2	spbp3	spri	orsa	orsq	lsdp
260	orx0	orx1	orx2	orx3	orx4	orx5	orx6	orx7
270	tsp0	tsp1	tsp2	tsp3		ora	orq	oraq
300	canx0	canx1	canx2	canx3	canx4	canx5	canx6	canx7
310	eawp0	easp0	eawp2	easp2		cana	canq	canaq
320	lcx0	lcx1	lcx2	lcx3	lcx4	lcx5	lcx6	lcx7
330	eawp4	easp4	eawp6	easp6		lca	lcq	lcaq
340	ansx0	ansx1	ansx2	ansx3	ansx4	ansx5	ansx6	ansx7
350	epp0	epbp1	epp2	epbp3	stac	ansa	ansq	stcd
360	anx0	anx1	anx2	anx3	anx4	anx5	anx6	anx7
370	epp4	epbp5	epp6	epbp7		ana	anq	anaq
400		mpf	mpy			cmg		
410		lde		rscr		ade		
420		ufm		dufm		fcmg		dfcmg
430	fszn	fld		dfld		ufa		dufa
440	sxl0	sxl1	sxl2	sxl3	sxl4	sxl5	sxl6	sxl7
450	stz	smic	scpr		stt	fst	ste	dfst
460		fmp		dfmp				
470	fstr	frd	dfstr	dfrd		fad		dfad
500	rpl					bcd	div	dvf
510				fneg		fcmp		dfcmp
520	rpt					fdi		dfdi
530		neg	cams	negl		ufs		dufs
540	sprp0	sprp1	sprp2	sprp3	sprp4	sprp5	sprp6	sprp7
550	sbar	stba	stbq	smcm	stc1			ssdp
560	rpd					fdv		dfdv
570				fno		fsb		dfsb
600	tze	tnz	tnc	trc	tmi	tpl		ttf
610	rtcd			rcu	teo	teu	dis	tov
620	eax0	eax1	eax2	eax3	eax4	eax5	eax6	eax7
630	ret			rccl	ldi	eaa	eaq	ldt
640	ersx0	ersx1	ersx2	ersx3	ersx4	ersx5	ersx6	ersx7
650	spri4	spbp5	spri6	spbp7	stacq	ersa	ersq	scu
660	erx0	erx1	erx2	erx3	erx4	erx5	erx6	erx7
670	tsp4	tsp5	tsp6	tsp7	lcpr	era	erq	eraq
700	tsx0	tsx1	tsx2	tsx3	tsx4	tsx5	tsx6	tsx7
710	tra			call6		tss	xec	xed
720	lxl0	lxl1	lxl2	lxl3	lxl4	lxl5	lxl6	lxl7
730		ars	qrs	lrs		als	qls	lls
740	stx0	stx1	stx2	stx3	stx4	stx5	stx6	stx7
750	stc2	stca	stcq	sreg	sti	sta	stq	staq
760	lprp0	lprp1	lprp2	lprp3	lprp4	lprp5	lprp6	lprp7
770		arl	qrl	lrl	gtb	alr	qlr	llr

Table 1-2. Instruction Opcode Chart, Bit 27 = 1

	000	001	002	003	004	005	006	007
000								
010								
020	mve				mvne			
030								
040								
050								
060	csl	csr			sztl	sztr	cmpb	
070								
100	mlr	mrl					cmpc	
110								
120	scd	scdr			scm	scmr		
130								
140								
150					sptr			
160	mvt				tct	tctr		
170				lptr				
200			ad2d	sb2d			mp2d	dv2d
210								
220			ad3d	sb3d			mp3d	dv3d
230			lsdr					
240								
250	spbp0	spri1	spbp2	spri3	ssdr			lptp
260								
270								
300	mvn	btd		cmpn		dtb		
310	easp1	eawp1	easp3	eawp3				
320								
330	easp5	eawp5	easp7	eawp7				
340								
350	epbp0	epp1	epbp2	epp3				
360								
370	epbp4	epp5	epbp6	epp7				
400								
410								
420								
430								
440				sareg				spl
450								
460				lareg				lpl
470								
500	a9bd	a6bd	a4bd	abd				awd
510								
520	s9bd	s6bd	s4bd	sbd				swd
530			camp					
540	ara0	ara1	ara2	ara3	ara4	ara5	ara6	ara7
550								sptp
560	aar0	aar1	aar2	aar3	aar4	aar5	aar6	aar7
570								
600	trtn	trtf			tmoz	tpnz	ttn	
610								
620								
630								
640	arn0	arn1	arn2	arn3	arn4	arn5	arn6	arn7
650	spbp4	spri5	spbp6	spri7				
660	nar0	nar1	nar2	nar3	nar4	nar5	nar6	nar7
670								
700								
710								
720								
730								
740	sar0	sar1	sar2	sar3	sar4	sar5	sar6	sar7
750					sra			
760	lar0	lar1	lar2	lar3	lar4	lar5	lar6	lar7
770					lra			

Table 1-3. Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
a4bd	502(1)	Add 4-bit character displacement to AR
a6bd	501(1)	Add 6-bit character displacement to AR
a9bd	500(1)	Add 9-bit character displacement to AR
aarN	56N(1)	Alphanumeric descriptor to ARN
abd	503(1)	Add bit displacement to AR
absa	212(0)	Absolute address to A register
ad2d	202(1)	Add using two decimal operands
ad3d	222(1)	Add using three decimal operands
ada	075(0)	Add to A register
adaq	077(0)	Add to AQ register
ade	415(0)	Add to E register
adl	033(0)	Add low to AQ register
adla	035(0)	Add logical to A register
adlaq	037(0)	Add logical to AQ register
adlq	036(0)	Add logical to Q register
adlxN	02N(0)	Add logical to index N
adq	076(0)	Add to Q register
adwp0	050(0)	Add to word number field of PR0
adwp1	051(0)	Add to word number field of PR1
adwp2	052(0)	Add to word number field of PR2
adwp3	053(0)	Add to word number field of PR3
adwp4	150(0)	Add to word number field of PR4
adwp5	151(0)	Add to word number field of PR5
adwp6	152(0)	Add to word number field of PR6
adwp7	153(0)	Add to word number field of PR7
adxN	06N(0)	Add to index N
alr	775(0)	A register left rotate
als	735(0)	A register left shift
ana	375(0)	AND to A register
anaq	377(0)	AND to AQ register
anq	376(0)	AND to Q register
ansa	355(0)	AND to storage from A register
ansq	356(0)	AND to storage from Q register
ansxN	34N(0)	AND to storage from index N
anxN	36N(0)	AND to index N
aos	054(0)	Add one to storage
araN	54N(1)	ARN to alphanumeric descriptor
arl	771(0)	A register right logical shift
arnN	64N(1)	ARN to numeric descriptor
ars	731(0)	A register right shift
asa	055(0)	Add stored to A register
asq	056(0)	Add stored to Q register
asxN	04N(0)	Add stored to index N
awca	071(0)	Add with carry to A register
awcq	072(0)	Add with carry to Q register
awd	507(1)	Add word displacement to AR
bcd	505(0)	Binary-to-BCD
btd	301(1)	Binary-to-Decimal
call6	713(0)	Call
camp	532(1)	Clear associative memory paged

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
cams	532(0)	Clear associative memory segmented
cana	315(0)	Comparative AND with A register
canaq	317(0)	Comparative AND with AQ register
canq	316(0)	Comparative AND with Q register
canx*N*	30*N*(0)	Comparative AND with index *N*
cioc	015(0)	Connect
cmg	405(0)	Compare magnitude
cmk	211(0)	Compare masked
cmpa	115(0)	Compare with A register
cmpaq	117(0)	Compare with AQ register
cmpb	066(1)	Compare bit strings
cmpc	106(1)	Compare alphanumeric character strings
cmpn	303(1)	Compare numeric
cmpq	116(0)	Compare with Q register
cmpx*N*	10*N*(0)	Compare with index *N*
cnaa	215(0)	Comparative NOT with A register
cnaaq	217(0)	Comparative NOT with AQ register
cnaq	216(0)	Comparative NOT with Q register
cnax*N*	20*N*(0)	Comparative NOT with index *N*
cwl	111(0)	Compare with limits
csl	060(1)	Combine bit strings left
csr	061(1)	Combine bit strings right
dfad	477(0)	DP floating add
dfcmg	427(0)	DP floating compare magnitude
dfcmp	517(0)	DP floating compare
dfdi	527(0)	DP floating divide inverted
dfdv	567(0)	DP floating divide
dfld	433(0)	DP floating load
dfmp	463(0)	DP floating multiply
dfrd	473(0)	DP floating round
dfsb	577(0)	DP floating subtract
dfst	457(0)	DP floating store
dfstr	472(0)	DP floating store rounded
dis	616(0)	Delay until interrupt signal
div	506(0)	Divide integer
drl	002(0)	Derail
dtb	305(1)	Decimal-to-binary convert
dufa	437(0)	DP unnormalized floating add
dufm	423(0)	DP unnormalized floating multiply
dufs	537(0)	DP unnormalized floating subtract
dv2d	207(1)	Divide using two decimal operands
dv3d	227(1)	Divide using three decimal operands
dvf	507(0)	Divide fraction
eaa	635(0)	Effective address to A register
eaq	636(0)	Effective address to Q register
easp0	311(0)	Effective address to segment number field of PR0
easp1	310(1)	Effective address to segment number field of PR1
easp2	313(0)	Effective address to segment number field of PR2
easp3	312(1)	Effective address to segment number field of PR3
easp4	331(0)	Effective address to segment number field of PR4
easp5	330(1)	Effective address to segment number field of PR5
easp6	333(0)	Effective address to segment number field of PR6
easp7	332(1)	Effective address to segment number field of PR7
eawp0	310(0)	Effective address to word and bit fields of PR0
eawp1	311(1)	Effective address to word and bit fields of PR1

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
eawp2	312(0)	Effective address to word and bit fields of PR2
eawp3	313(1)	Effective address to word and bit fields of PR3
eawp4	330(0)	Effective address to word and bit fields of PR4
eawp5	331(1)	Effective address to word and bit fields of PR5
eawp6	332(0)	Effective address to word and bit fields of PR6
eawp7	333(1)	Effective address to word number field of PR7
eaxN	62N(0)	Effective address to index N
epaq	213(0)	Effective pointer to AQ register
epbp0	350(1)	Effective pointer at base to PR0
epbp1	351(0)	Effective pointer at base to PR1
epbp2	352(1)	Effective pointer at base to PR2
epbp3	353(0)	Effective pointer at base to PR3
epbp4	370(1)	Effective pointer at base to PR4
epbp5	371(0)	Effective pointer at base to PR5
epbp6	372(1)	Effective pointer at base to PR6
epbp7	373(0)	Effective pointer at base to PR7
epp0	350(0)	Effective pointer to PR0
epp1	351(1)	Effective pointer to PR1
epp2	352(0)	Effective pointer to PR2
epp3	353(1)	Effective pointer to PR3
epp4	370(0)	Effective pointer to PR4
epp5	371(1)	Effective pointer to PR5
epp6	372(0)	Effective pointer to PR6
epp7	373(1)	Effective pointer to PR7
era	675(0)	Exclusive OR to A register
ersq	677(0)	Exclusive OR to AQ register
erq	676(0)	Exclusive OR to Q register
ersa	655(0)	Exclusive OR to storage with A register
ersq	656(0)	Exclusive OR to storage with Q register
ersxN	64N(0)	Exclusive OR to storage with index N
erxN	66N(0)	Exclusive OR to index N
fad	475(0)	Floating add
fcmg	425(0)	Floating compare magnitude
fcmp	515(0)	Floating compare
fdi	525(0)	Floating divide inverted
fdv	565(0)	Floating divide
fld	431(0)	Floating load
fmp	461(0)	Floating multiply
fneg	513(0)	Floating negate
fno	573(0)	Floating normalize
frd	471(0)	Floating round
fsb	575(0)	Floating subtract
fst	455(0)	Floating store
fstr	470(0)	Floating store rounded
fszn	430(0)	Floating set zero and negative indicators
gtb	774(0)	Gray-to-binary convert
larN	76N(1)	Load ARN
lareg	463(1)	Load address registers
lbar	230(0)	Load base address register
lca	335(0)	Load complement into A register
lcaq	337(0)	Load complement into AQ register
lcpr	674(0)	Load central processor register
lcq	336(0)	Load complement into Q register
lcxN	32N(0)	Load complement into index N
lda	235(0)	Load A register

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
ldac	034(0)	Load A register and clear
ldaq	237(0)	Load AQ register
ldbr	232(0)	Load descriptor base register
lde	411(0)	Load E register
ldi	634(0)	Load indicator register
ldq	236(0)	Load Q register
ldqc	032(0)	Load Q register and clear
ldt	637(0)	Load timer register
ldx*N*	22*N*(0)	Load index *N*
llr	777(0)	Long left rotate
lls	737(0)	Long left shift
lpl	467(1)	Load pointers and lengths
lpri	173(0)	Load pointer registers from ITS pairs
lprp*N*	76*N*(0)	Load pointer register *N* from packed pointer
lptp	257(1)	Load page table pointers
lptr	173(1)	Load page table registers
lra	774(1)	Load ring alarm register
lreg	073(0)	Load registers
lrl	773(0)	Long right logical
lrs	733(0)	Long right shift
lsdp	257(0)	Load segment descriptor pointers
lsdr	232(1)	Load segment descriptor registers
lxl*N*	72*N*(0)	Load index *N* from lower
mlr	100(1)	Move alphanumeric left to right
mme1	001(0)	Master mode entry 1
mme2	004(0)	Master mode entry 2
mme3	005(0)	Master mode entry 3
mme4	007(0)	Master mode entry 4
mp2d	206(1)	Multiply using two decimal operands
mp3d	226(1)	Multiply using three decimal operands
mpf	401(0)	Multiply fraction
mpy	402(0)	Multiply integer
mrl	101(1)	Move alphanumeric right to left
mve	020(1)	Move alphanumeric edited
mvn	300(1)	Move numeric
mvne	024(1)	Move numeric edited
mvt	160(1)	Move alphanumeric with translation
nar*N*	66*N*(1)	Numeric descriptor to AR*N*
neg	531(0)	Negate (A register)
negl	533(0)	Negate long (AQ register)
nop	011(0)	No operation
ora	275(0)	OR to A register
oraq	277(0)	OR to AQ register
orq	276(0)	OR to Q register
orsa	255(0)	OR to storage from A register
orsq	256(0)	OR to storage from Q register
orsx*N*	24*N*(0)	OR to storage from index *N*
orx*N*	26*N*(0)	OR to index *N*
puls1	012(0)	Pulse location 1
puls2	013(0)	Pulse location 2
qlr	776(0)	Q register left rotate
qls	736(0)	Q register left shift
qrl	772(0)	Q register right logical shift
qrs	732(0)	Q register right shift
rccl	633(0)	Read calendar clock

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
rcu	613(0)	Restore control unit
ret	630(0)	Return
rmcm	233(0)	Read memory controller mask
rpd	560(0)	Repeat double
rpl	500(0)	Repeat link
rpt	520(0)	Repeat
rscr	413(0)	Read system controller register
rsw	231(0)	Read switches
rtcd	610(0)	Return control double
s4bd	522(1)	Subtract 4-bit displacement from AR
s6bd	521(1)	Subtract 6-bit displacement from AR
s9bd	520(1)	Subtract 9-bit displacement from AR
sarN	74N(1)	Store ARN
sareg	443(1)	Store address registers
sb2d	203(1)	Subtract using two decimal operands
sb3d	223(1)	Subtract using three decimal operands
sba	175(0)	Subtract from A register
sbar	550(0)	Store base address register
sbaq	177(0)	Subtract from AQ register
sbd	523(1)	Subtract bit displacement from AR
sbla	135(0)	Subtract logical from A register
sblaq	137(0)	Subtract logical from AQ register
sblq	136(0)	Subtract logical from Q register
sblxN	12N(0)	Subtract logical from index N
sbq	176(0)	Subtract from Q register
sbxN	16N(0)	Subtract from index N
scd	120(1)	Scan character double
scdr	121(1)	Scan character double reverse
scm	124(1)	Scan with mask
scmr	125(1)	Scan with mask reverse
scpr	452(0)	Store central processor register
scu	657(0)	Store control unit
sdbr	154(0)	Store descriptor base register
smcm	553(0)	Set memory controller mask
smic	451(0)	Set memory interrupt cells
spbp0	250(1)	Store segment base pointer of PR0
spbp1	251(0)	Store segment base pointer of PR1
spbp2	252(1)	Store segment base pointer of PR2
spbp3	253(0)	Store segment base pointer of PR3
spbp4	650(1)	Store segment base pointer of PR4
spbp5	651(0)	Store segment base pointer of PR5
spbp6	652(1)	Store segment base pointer of PR6
spbp7	653(0)	Store segment base pointer of PR7
spl	447(1)	Store pointers and lengths
spri	254(0)	Store pointer registers as ITS pairs
spri0	250(0)	Store PR0 as an ITS pair
spri1	251(1)	Store PR1 as an ITS pair
spri2	252(0)	Store PR2 as an ITS pair
spri3	253(1)	Store PR3 as an ITS pair
spri4	650(0)	Store PR4 as an ITS pair
spri5	651(1)	Store PR5 as an ITS pair
spri6	652(0)	Store PR6 as an ITS pair
spri7	653(1)	Store PR7 as an ITS pair
sprpN	54N(0)	Store pointer register N packed
sptp	557(1)	Store page table pointers

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
sptr	154(1)	Store page table registers
sra	754(1)	Store ring alarm register
sreg	753(0)	Store registers
ssa	155(0)	Subtract stored from A register
sscr	057(0)	Set system controller register
ssdp	557(0)	Store segment descriptor pointers
ssdr	254(1)	Store segment descriptor registers
ssq	156(0)	Subtract stored from Q register
ssx*N*	14*N*(0)	Subtract stored from index *N*
sta	755(0)	Store A register
stac	354(0)	Store A register conditional
stacq	654(0)	Store A register conditional on Q register
staq	757(0)	Store AQ register
stba	551(0)	Store 9-bit characters of A register
stbq	552(0)	Store 9-bit characters of Q register
stc1	554(0)	Store instruction counter + 1
stc2	750(0)	Store instruction counter + 2
stca	751(0)	Store 6-bit characters of A register
stcd	357(0)	Store control double
stcq	752(0)	Store 6-bit characters of Q register
ste	456(0)	Store E register
sti	754(0)	Store indicator register
stq	756(0)	Store Q register
stt	454(0)	Store timer register
stx*N*	74*N*(0)	Store index *N*
stz	450(0)	Store zero
swca	171(0)	Subtract with carry from A register
swcq	172(0)	Subtract with carry from Q register
swd	527(1)	Subtract word displacement from AR
sxl*N*	44*N*(0)	Store index *N* in lower
szn	234(0)	Set zero and negative indicators
sznc	214(0)	Set zero and negative indicators and clear
sztl	064(1)	Set zero and truncation indicators with bit string left
sztr	065(1)	Set zero and truncation indicators with bit string right
tct	164(1)	Test character and translate
tctr	165(1)	Test character and translate reverse
teo	614(0)	Transfer on exponent overflow
teu	615(0)	Transfer on exponent underflow
tmi	604(0)	Transfer on minus
tmoz	604(1)	Transfer on minus or zero
tnc	602(0)	Transfer on no carry
tnz	601(0)	Transfer on nonzero
tov	617(0)	Transfer on overflow
tpl	605(0)	Transfer on plus
tpnz	605(1)	Transfer on plus and nonzero
tra	710(0)	Transfer
trc	603(0)	Transfer on carry
trtf	601(1)	Transfer on truncation indicator off
trtn	600(1)	Transfer on truncation indicator on
tsp0	270(0)	Transfer and set PR0

Table 1-3. (cont) Alphabetic Listing of Processor Instructions

Mnemonic	Code	Meaning
tsp1	271(0)	Transfer and set PR1
tsp2	272(0)	Transfer and set PR2
tsp3	273(0)	Transfer and set PR3
tsp4	670(0)	Transfer and set PR4
tsp5	671(0)	Transfer and set PR5
tsp6	672(0)	Transfer and set PR6
tsp7	673(0)	Transfer and set PR7
tss	715(0)	Transfer and set slave
tsx*N*	70*N*(0)	Transfer and set index *N*
ttf	607(0)	Transfer on tally indicator off
ttn	606(1)	Transfer on tally indicator on
tze	600(0)	Transfer on zero
ufa	435(0)	Unnormalized floating add
ufm	421(0)	Unnormalized floating multiply
ufs	535(0)	Unnormalized floating subtract
xec	716(0)	Execute
xed	717(0)	Execute double

Instruction Address Modifiers

Instruction address modifiers are divided into four groups, as shown in Table 1-4. These are:

R Register
RI Register then indirect
IR Indirect then register
IT Indirect then tally

Standard Modifiers:

Table 1-4. Standard Instruction Modifier Chart

	00	01	02	03	04	05	06	07	Type
00	none	au	qu	du	ic	al	ql	dl	R
10	0	1	2	3	4	5	6	7	R
20	n*	au*	qu*	IPR	ic*	al*	ql*	IPR	RI
30	0*	1*	2*	3*	4*	5*	6*	7*	RI
40	f1	itp	IPR	its	sd	scr	f2	f3	IT
50	ci	i	sc	ad	di	dic	id	idc	IT
60	*n	*au	*qu	*du	*ic	*al	*ql	*dl	IR
70	*0	*1	*2	*3	*4	*5	*6	*7	IR

(where IPR means IPR fault if detected)

Special Modifiers:

inst	tag	meaning
scpr	00	Store APU history register
	01	Store fault register
	06	Store mode register and cache mode register
	20	Store CU history register
	40	Store OU history register
	60	Store DU history register
lcpr	02	Load cache mode register
	03	Load 0's into all history registers
	04	Load mode register
	07	Load 1's into all history registers

Multics Indirect Words

The Level 68 processor has two special indirect words that are active in Multics mode; the indirect-to-segment (ITS) pointer pair and the indirect-to-pointer (ITP) pointer pair. These words are recognized only when an RI or IR address modification references an even location.

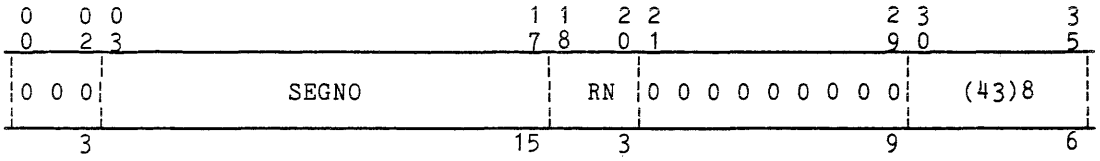
ITS PAIR FORMAT

The indirect-to-segment (ITS) word pair is an indirect pointer to a segment in a Multics process. Segment offset and additional address modification are permitted.

PL/I Declaration (its.incl.pl1)

```
declare 1 its          based aligned,              /* Even word */
       (2 pad1         bit(3),
        2 segno        bit(15),
        2 ringno       bit(3),
        2 pad2         bit(9),
        2 its_mod      bit(6),
        2 offset       bit(18),                    /* Odd word */
        2 pad3         bit(3),
        2 bit_offset   bit(6),
        2 pad4         bit(3),
        2 mod          bit(6)) unaligned;
```

Even Word:



Odd Word:

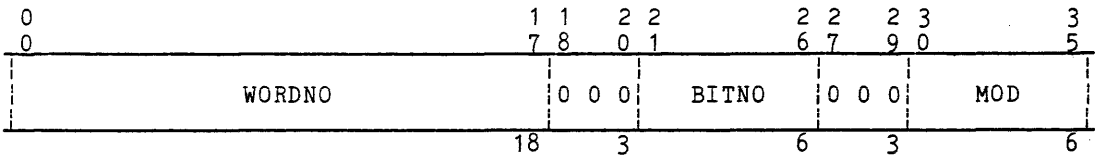


Figure 1-2. ITS Pair Format

Legend:

SEGNO (its.segno) 15-bit segment number.

RN (its.ringno) a lower bound for the value of TPR.TRR (temporary ring register) for the address preparation involving this ITS pair.

(43)8 (its.its_mod) ITS modifier.

WORDNO (its.offset) word offset to be used in calculating the computed address within the segment.

BITNO (its.bit_offset) bit offset to be used in calculating the computed address within the segment.

MOD (its.mod) any valid instruction modifier.

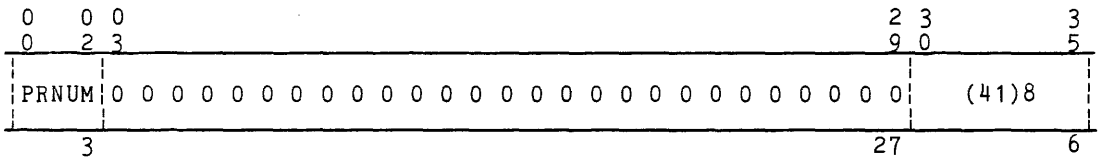
ITP PAIR FORMAT

The indirect-to-pointer (ITP) word pair is similar to the ITS pair except that it specifies the number of a pointer register (PRNUM) that already contains a valid pointer to the segment to which access is desired.

PL/I Declaration (its.incl.pl1)

```
declare 1 itp              based aligned,      /* Even word */
       (2 pr_no            bit(3),
        2 pad1             bit(27),
        2 itp_mod          bit(6),
        2 offset           bit(18),           /* Odd word */
        2 pad2             bit(3),
        2 bit_offset       bit(6),
        2 pad3             bit(3),
        2 mod              bit(6)) unaligned;
```

Even Word:



Odd Word:

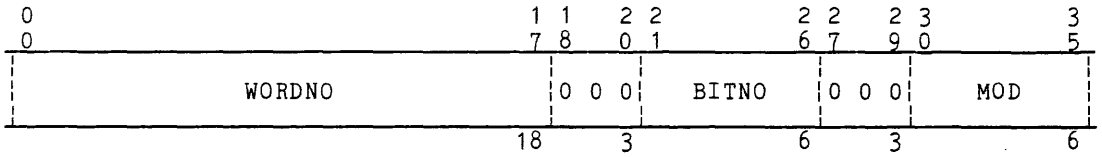


Figure 1-3. ITP Pair Format

Legend:

PRNUM (itp.pr_no) the number (0-7) of the pointer register to be used in the indirect reference.

(41)8 (itp.itp_mod) ITP modifier.

WORDNO (itp.offset) word offset as for ITS.

BITNO (itp.bit_offset) bit offset as for ITS.

MOD (itp.mod) any valid instruction modifier.

APPENDING UNIT FORMATS

After the CU has completed its address modifications, the processor appending unit (APU) converts the final virtual address (segment, page, and offset) to a 24-bit absolute store address, as required for the store access, when a store access is required.

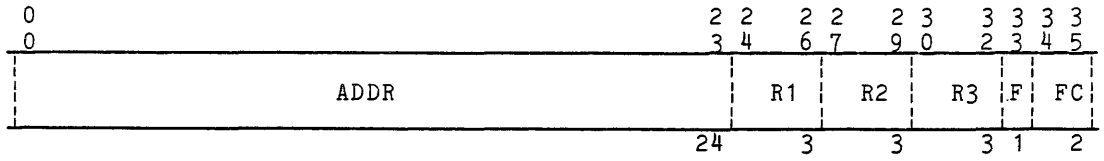
Segment Descriptor Word Format

The segment descriptor word (SDW) pair contains information necessary to control the access to a segment by a process. The SDW for a segment is constructed from data in the directory entry for the segment and in the system segment table (SST) when the segment is faulted upon by the process. The SDW for segment N (unique within the process) is placed at offset 2N in the descriptor segment (dseg) of the process.

PL/I Declaration (sdw.incl.pl1)

```
dcl 1 sdw                 based (sdwp) aligned,     /* Even word */
   (2 add                 bit(24),
    2 (r1, r2, r3)        bit(3),
    2 df                  bit(1),
    2 df_no               bit(2),
    2 pad1                bit(1),                   /* Odd word */
    2 bound               bit(14),
    2 read                bit(1),
    2 execute             bit(1),
    2 write               bit(1),
    2 privileged          bit(1),
    2 unpaged             bit(1),
    2 entry_bound_sw      bit(1),
    2 cache               bit(1),
    2 entry_bound         bit(14)) unaligned;
```

Even Word:



Odd Word:

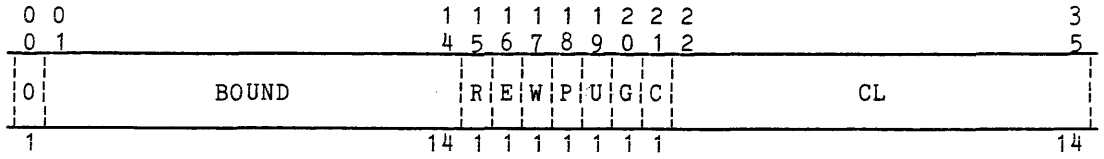


Figure 1-4. Segment Descriptor Word (SDW) Format

Legend:

Mask	Field	Meaning
777777 U 770000 L	ADDR	(sdw.add) base address of segment (U=1) or segment page table (U=0).
007000	R1	(sdw.r1) highest effective read/write ring.
000700	R2	(sdw.r2) highest effective read/execute ring.
000070	R3	(sdw.r3) highest effective call ring.
000004	F	(sdw.df) directed fault indicator. 1 = the necessary unpaged segment or segment page table is in memory. 0 = execute the directed fault specified in FC.
000003	FC	(sdw.df_no) the number of the directed fault (DF0-DF3) to be executed if F=0.
377770	BOUND	(sdw.bound) highest modulo 16 computed address (offset) that may be used in referencing the segment without causing an out_of_segment_bounds fault (ACV-OOSB).
000004	R	(sdw.read) read permission bit.
000002	E	(sdw.execute) execute permission bit (xec and xed excluded).
000001	W	(sdw.write) write permission bit.
400000	P	(sdw.privileged) privileged mode bit. 0 = privileged instructions cannot be executed in this segment. 1 = privileged instructions can be executed in this segment if it runs in ring 0.
200000	U	(sdw.unpaged) paged/unpaged bit. 0 = segment is paged and ADDR is the address of the page table. 1 = segment is unpaged and ADDR is the base address of the segment.
100000	G	(sdw.entry_bound_sw) gate indicator bit. 0 = any call from an external segment must be to an offset less than the value of CL. 1 = any valid segment offset may be called.
040000	C	(sdw.cache) cache control bit. 0 = words (operands or instructions) from this segment cannot be placed in the cache. 1 = words from this segment can be placed in the cache.
037777	CL	(sdw.entry_bound) call limiter. Any external call to this segment must be to an offset less than CL if G=0.

Page Table Word Format

The page table word (PTW) contains location and state information for a page of a paged segment. The PTWs for a paged segment are created in a free entry in the active segment table (AST) area of the SST when the segment is first referenced by some process. Subsequent segment faults by other processes reference the existing page table.

PL/I Declaration (ptw.incl.pl1)

```
dcl 1 ptw        based (ptp) aligned,
    2 add        bit(18) unaligned,
    2 did        bit(4) unaligned,
    2 first      bit(1) unaligned,
    2 processed  bit(1) unaligned,
    2 unusable1  bit(2) unaligned,
    2 phu        bit(1) unaligned,
    2 unusable2  bit(1) unaligned,
    2 nypd       bit(1) unaligned,
    2 phm        bit(1) unaligned,
    2 phu1       bit(1) unaligned,
    2 wired      bit(1) unaligned,
    2 os         bit(1) unaligned,
    2 df         bit(1) unaligned,
    2 df_no      bit(2) unaligned;
```

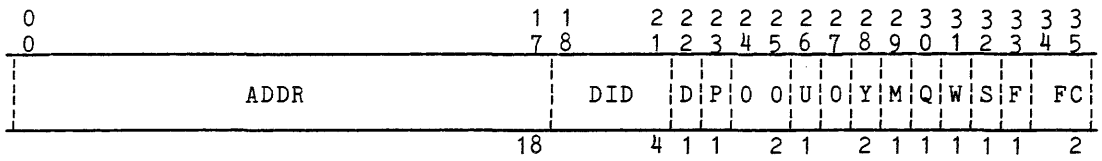


Figure 1-5. Page Table Word (PTW) Format

Legend:

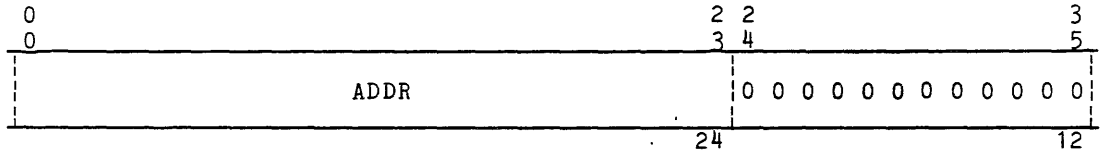
Mask	Field	Meaning
777777	ADDR	(ptw.add) modulo 64 page address if page is in store, or record number of page if page is not in store. For a 1024-word page size, the hardware ignores the four LSB of the in-main-memory page address.
740000	DID	(ptw.did) device identifier for device containing the page (used only by software).
020000	D	(ptw.first) paging device update delay bit (used only by software). 1 = page must not be written to paging device. 0 = page can be written to paging device.
010000	P	(ptw.processed) temporary bit used in post_purging (used only by software).
001000	U	(ptw.phu) used (referenced) bit (set by hardware). 1 = page has been used. 0 = page has not been used.
000200	Y	(ptw.nypd) not yet on paging device bit (used only by hardware). 1 = page has not been updated to paging device. 0 = page has been updated to paging device.
000100	M	(ptw.phm) modified bit (set by hardware). 1 = page has been modified. 0 = page has not been modified.
000040	Q	(ptw.phu1) quantum bit (used only by software). 1 = page has been used during the quantum. 0 = page has not been used during the quantum.

Mask	Field	Meaning
000020	W	(ptw.wired) wired bit (used only by software). 1 = page is wired. 0 = page is not wired.
000010	S	(ptw.os) out of service bit (used only by software). 1 = page is out of service (I/O in progress). 0 = page is in service.
000004	F	(ptw.df) main memory bit (checked by hardware). 1 = page is in main memory. 0 = page is not in main memory. Execute directed fault FC.
000003	FC	(ptw.df_no) directed fault number for page fault (checked by hardware).

Descriptor Base Register Format

The descriptor base register (DBR) specifies the descriptor segment for the process. Because of the high degree of similarity between the SDW and the DBR data, the declaration for the SDW is used for the equivalent fields of the DBR data.

Even Word:



Odd Word:

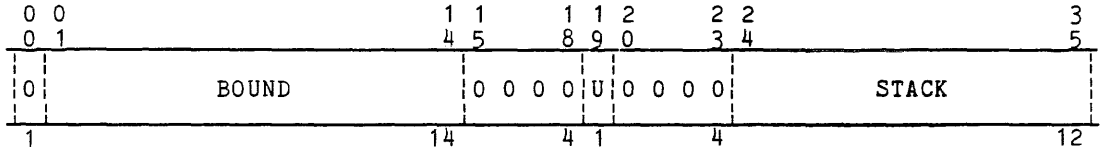


Figure 1-6. Descriptor Base Register (DBR) Format

Legend:

ADDR (sdw.add) base address of descriptor segment (U=1) or descriptor segment page table (U=0).

BOUND (sdw.bound) highest modulo 16 offset within the descriptor segment that will not cause an out-of-segment-bounds fault (ACV-OOSB). (This is twice the first out-of-bounds segment number.)

U (sdw.unpaged) paged/unpaged bit.
1 = descriptor segment is unpaged.
0 = descriptor segment is paged.

STACK (sdw.entry_bound) 12 MSB of the 15-bit stack segment number for the process. This field is referenced only by the "call6" instruction.

Associative Memory Formats

The processor contains two associative memories that greatly reduce the number of main store accesses required for the address preparation function. The degree of store access reduction depends on the degree of locality of reference of the procedure executing in the processor.

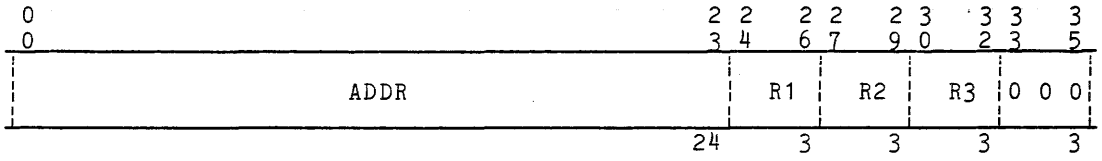
SDW ASSOCIATIVE MEMORY

The segment descriptor word associative memory (SDWAM) contains the SDWs and segment numbers for the 16 most recently used segments for the process currently using the processor. When a given segment is accessed, the SDWAM is queried with the segment number before a main memory access to the descriptor segment is made. If a match is found, the SDWAM returns the SDW and the need for the main memory access is obviated.

Because the SDWAM register holds an SDW with the directed-fault fields cleared, the data in main memory is described by the declaration for the SDW. Because the SDWAM match logic register is not used by Multics, there is no software declaration for the data in main memory.

The ssdr instruction stores the SDWAM registers in the following format:

SDWAM Register Upper Half:



SDWAM Register Lower Half:

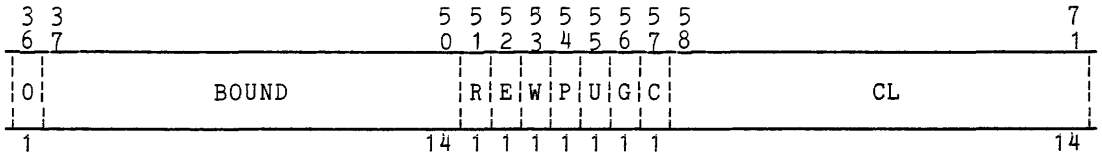


Figure 1-7. SDW Associative Memory (SDWAM) Register Format

Legend:

ADDR	(sdw.add) segment address as in SDW format (see Figure 1-4 above).
R1	(sdw.r1) read/write ring bracket as in SDW format.
R2	(sdw.r2) read/execute ring bracket as in SDW format.
R3	(sdw.r3) call ring bracket as in SDW format.
BOUND	(sdw.bound) segment bound as in SDW format.

R,E,W,P, U,G,C — access control bits as in SDW format.

CL — (sdw.entry_bound) call limiter as in SDW format.

The ssdp instruction stores the SDWAM match logic registers in the following format:

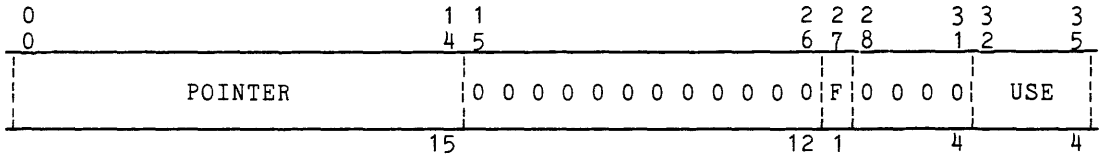


Figure 1-8. SDW Associative Memory (SDWAM) Match Logic Register Format

Legend:

POINTER — 15-bit effective segment number generated when this SDW was fetched from main memory.

F — full/empty bit.
1 = this AM register contains a valid SDW.
0 = this AM register is empty.

USE — usage count.
The "oldest" SDWAM entry has count 00 and the "newest" has count 17.

PTW ASSOCIATIVE MEMORY

The processor page table word associative memory (PTWAM) contains the in-main-memory addresses and the segment and page numbers of the 16 most recently used pages for the process currently using the processor. When a paged segment is accessed, the PTWAM is first queried with the segment and page number. If a match is found, the PTWAM returns the in-main-memory page address and the main memory access to the PTW is obviated.

Because the PTWAM register holds a PTW with some control bits cleared, the data in main memory is described by the declaration for the PTW. Because the PTWAM match logic register is not used by Multics, there is no software declaration for the data in main memory.

The sptr instruction stores the PTWAM registers in the following format:

PTWAM Register:

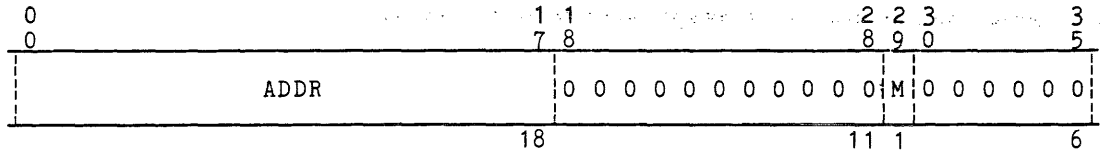


Figure 1-9. PTW Associative Memory (PTWAM) Register Format

Legend:

ADDR (ptw.add) modulo 64 page address as in PTW format (see Figure 1-5 above).

M (ptw.phm) modified bit.
1 = page has been modified.
0 = page has not been modified.

PTWAM Match Logic Register:

The sptp instruction stores the PTWAM pointers in the following format:

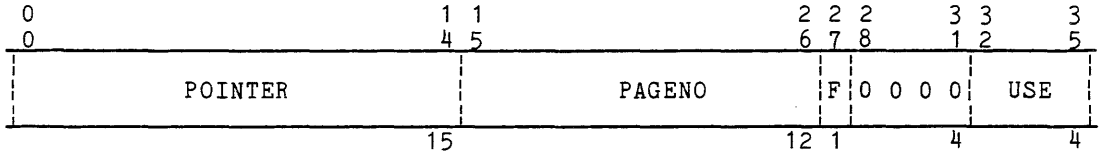


Figure 1-10. PTW Associative Memory (PTWAM) Match Logic Register Format

Legend:

POINTER effective segment number as in SDWAM format (see Figure 1-8 above).

PAGENO page number to which this PTW refers.

For a 1024-word page size, the four LSB are forced to zero by the hardware.

F full/empty bit as in SDWAM format.

USE usage count as in SDWAM format.

DECIMAL UNIT FORMATS

The decimal unit (DU) is that portion of the processor hardware that executes the extended instruction set (EIS) instructions for bit- and character-string processing and for decimal arithmetic.

EIS Multiword Instruction Format

The EIS processor instructions occupy one, three or four words in memory depending upon the number of EIS data descriptors required for their execution. Single word EIS instructions have the same format as the basic instructions already described.

(There is no include file for the declaration of this data.)

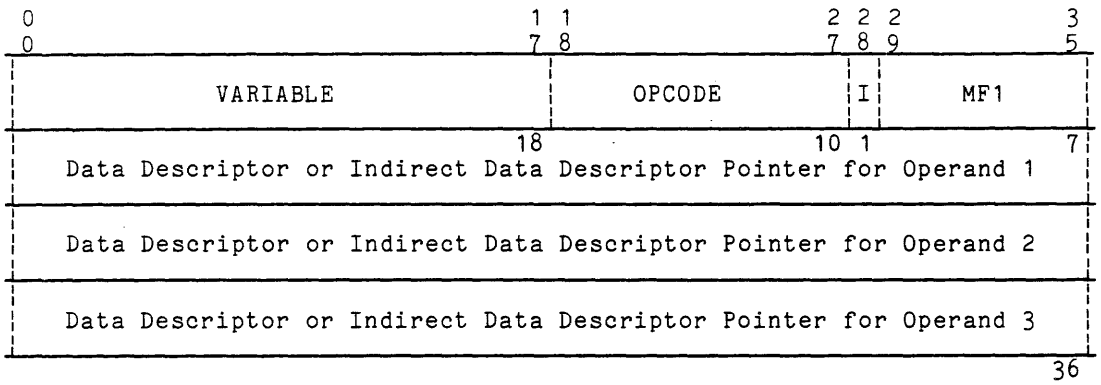


Figure 1-11. EIS Multiword Instruction Format

Legend:

VARIABLE Interpreted according to the requirements of the individual EIS instructions; contains MF2 and MF3 for the second and third data descriptors if needed.

OPCODE Instruction operation code.

I Interrupt inhibit bit.

MF1 Modification field for data descriptor 1.

EIS Data Descriptor Modification Field Format

Many of the EIS data descriptors required by EIS instructions will have a modification field (MF) in the first word of the multiword instruction. The MF fields contain additional address preparation information that cannot be contained in the data descriptor.

PL/I Declaration (derived from eis_bits.incl.alm)

```
dcl 1 mf   based unaligned,
    2 ar   bit(1),
    2 rl   bit(1),
    2 id   bit(1),
    2 reg  bit(4);
```

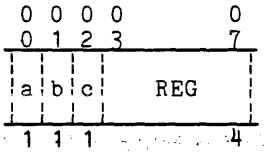


Figure 1-12. EIS Data Descriptor Modification Field (MF) Format

Legend:

Key	Field	Meaning
a	AR	(mf.ar) pointer register control in preparing addresses from this descriptor. 0 = pointer register not to be used. 1 = pointer register is used.
b	RL	(mf.rl) register length control. 0 = N field of the descriptor is operand length. 1 = N field of the descriptor is number of register containing operand length.
c	ID	(mf.id) indirect descriptor control. 0 = operand descriptor follows instruction word in its sequential location. 1 = operand descriptor location contains an indirect pointer to the descriptor.
	REG	(mf.reg) register number for R-type modification of ADDRESS of the descriptor. (see "EIS Instruction Address Modification Codes" below.)

EIS Data Descriptor Formats

The words occupying the data descriptor locations following an EIS instruction are either an indirect data descriptor pointers or any of the three EIS data descriptors.

INDIRECT DATA DESCRIPTOR POINTER FORMAT

(There is no include file for the declaration of this data.)

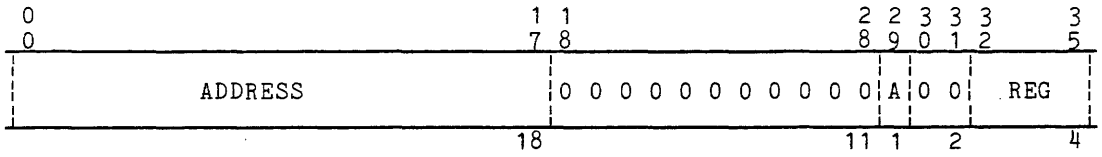


Figure 1-13. EIS Indirect Data Descriptor Pointer Format

Legend:

ADDRESS	For A=0; 18-bit procedure segment address. For A=1; 3-bit pointer register number and 15-bit signed word offset.
A	Pointer register flag.
REG	Address modification code. (see "EIS Instruction Address Modification Codes" below.)

BIT STRING DATA DESCRIPTOR FORMAT

(There is no include file for the declaration of this data.)

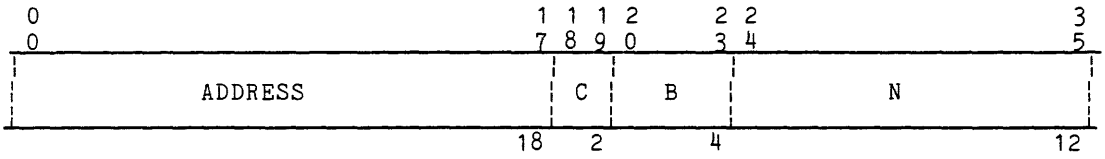


Figure 1-14. EIS Bit String Data Descriptor Format

Legend:

Given that this is descriptor n:

ADDRESS For MFn.AR=0; 18-bit procedure segment address.
For MFn.AR=1; 3-bit pointer register number and 15-bit signed word offset.

C Character position offset within the word determined by ADDRESS. This count is in units of 9-bit characters.

B Bit position offset within the character determined by C.

N For MFn.RL=0; 12-bit bit count.
For MFn.RL=1; eight "0" bits and 4-bit number of register containing bit count.

ALPHANUMERIC DATA DESCRIPTOR FORMAT

(There is no include file for the declaration of this data.)

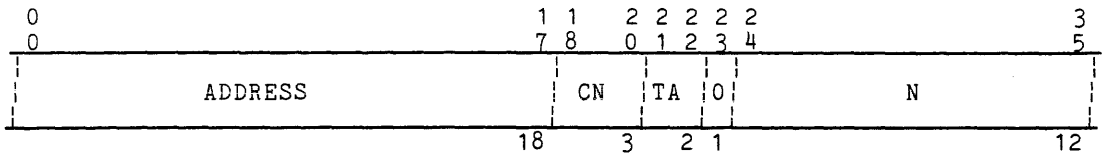


Figure 1-15. EIS Alphanumeric Data Descriptor Format

Legend:

Given that this is descriptor n:

ADDRESS For MFn.AR=0; 18-bit procedure segment address.
For MFn.AR=1; 3-bit pointer register number and 15-bit signed word offset.

CN Character position offset within the word determined by ADDRESS. This count depends on the character size (TA) specified by the descriptor.

9-bit: character positions 0 to 3 are designated by CN = 000,010,100,110 respectively. Other codes are invalid.
6-bit: character positions 0 to 5 are designated by CN = 000 through 101. 110 and 111 are invalid.
4-bit: character positions 0 to 7 are designated by CN = 000 through 111.

TA Alphanumeric character type code

00 = 9-bit character
01 = 6-bit character
10 = 4-bit character
11 = illegal (IPR fault)

N For MFn.RL=0; 12-bit character count.
For MFn.RL=1; eight "0" bits and 4-bit number of register containing character count.

NUMERIC DATA DESCRIPTOR FORMAT

(There is no include file for the declaration of this data.)

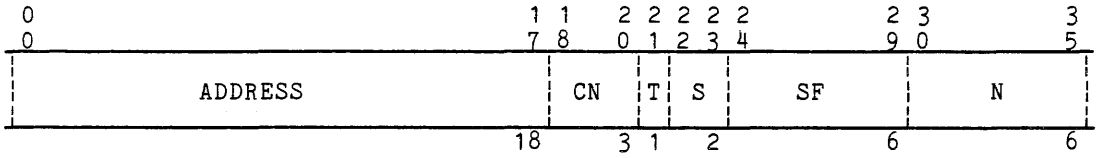


Figure 1-16. EIS Numeric Data Descriptor Format

Legend:

Given that this is descriptor n:

ADDRESS For MFn.AR=0; 18-bit procedure segment address.
For MFn.AR=1; 3-bit pointer register number and 15-bit signed word offset.

CN Character position offset within the word determined by ADDRESS. This count is in units of the character size specified by the descriptor.

T Numeric character type code.
0 = 9-bit character
1 = 4-bit character

S Sign and decimal type code.
00 = Leading sign, floating point
01 = Leading sign, scaled
10 = Trailing sign, scaled
11 = No sign, scaled

SF Scaling factor.

N For MFn.RL=0; 6-bit character count.
For MFn.RL=1; two "0" bits and 4-bit number of register containing character count.

The address modification codes used in the "REG" field of the EIS data descriptor modification field and the indirect data descriptor pointer and in the "N" field of the data descriptors are slightly different from those used in the basic instructions.

Octal Code	Basic "R" Type	"REG" Type (1)	"N" Type MFn.RL=1 (2)
00	None	None	Invalid (3)
01	au	au	au
02	qu	qu	qu
03	du	du	Invalid
04	ic	ic	Invalid
05	al	a	a
06	ql	q	q
07	dl	Invalid	Invalid
10	x0	x0	x0
11	x1	x1	x1
12	x2	x2	x2
13	x3	x3	x3
14	x4	x4	x4
15	x5	x5	x5
16	x6	x6	x6
17	x7	x7	x7

(a) When the "REG" field of an indirect data descriptor pointer contains a register code, the specified register contents are interpreted as a word index value.

When the "REG" field of the EIS data descriptor modification field contains a register code, the specified register contents are interpreted as a character index value. The size of the character is specified by the data type given in the data descriptor.

When the descriptor word does not have an associated MF field in the instruction word, its format is identical to an indirect data descriptor.

The A and Q registers provide for indexing by values greater than the range of an 18-bit field. For address modification codes 05 and 06, low-order bits of the specified register are used as follows:

Bit strings	lowest 24 bits
4- and 6-bit characters	lowest 21 bits
9-bit characters	lowest 20 bits

All index values are taken as unsigned, positive integers.

(b) Except in the cases of address modification codes 05 and 06, the full 18-bit extent of the specified register is used for the value of string length. For codes 05 and 06, the bit extents given in (a) above apply.

(c) Invalid address modification codes cause an IPR fault.

EIS Data Formats

There are six valid formats for data intended for use by the DU: bit string data, three modes of alphanumeric data, and two modes of numeric data.

BIT STRING DATA FORMAT

The data is a string of contiguous bits starting anywhere in a word and having extent without regard to word or character boundaries.

(There is no include file for the declaration of this data.)

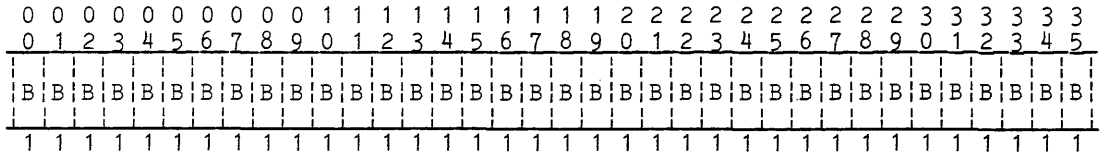


Figure 1-17. EIS Bit String Data Format

ALPHANUMERIC DATA FORMAT

The data is a string of 4, 6, or 9-bit characters starting at any character boundary and having extent without regard to word boundaries. In 4-bit mode, the "0" bits are <u>not</u> part of the data. The DU skips over them in input data and inserts them in output data.

(There is no include file for the declaration of this data.)

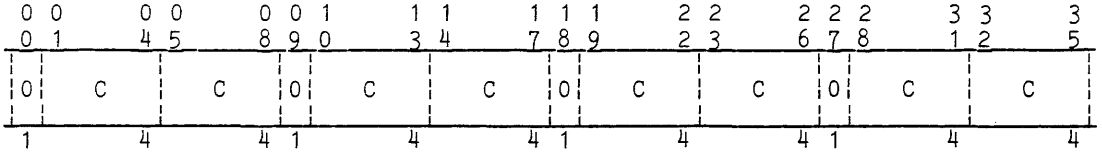


Figure 1-18. EIS Alphanumeric Data Format, 4-bit Mode

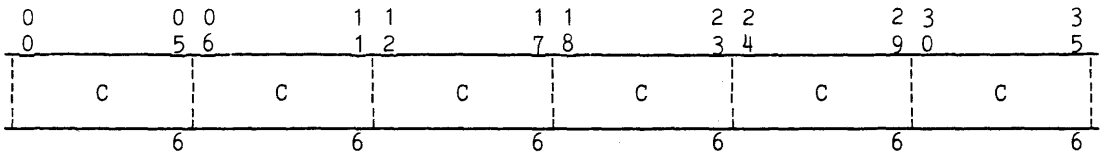


Figure 1-19. EIS Alphanumeric Data Format, 6-bit Mode

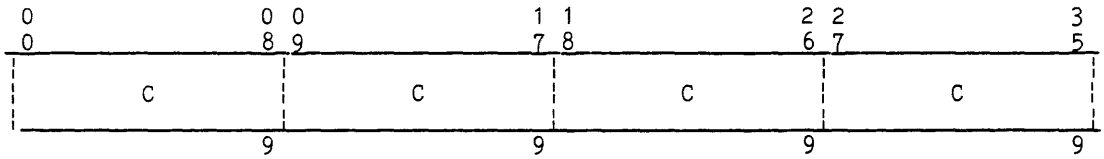


Figure 1-20. EIS Alphanumeric Data Format, 9-bit Mode

NUMERIC DATA FORMAT

The data is a string of numeric digits starting at any digit boundary and having extent without regard to word boundaries. In 4-bit mode, the "0" bits are <u>not</u> part of the data. The DU skips over them in input data and inserts them in output data.

(There is no include file for the declaration of this data.)

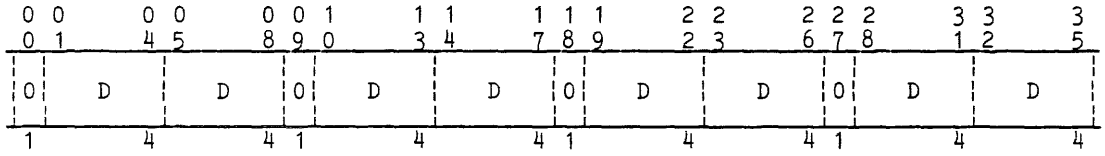


Figure 1-21. EIS Numeric Data Format, 4-Bit Mode

Legend:

Each D represents a digit, a sign, or an exponent, depending on the descriptor. The digits 0 through 9 are represented by a D of 0000 through 1001. A D of 1010 through 1111 where a digit is expected will cause an IPR fault; D's of 1010, 1011, 1100, 1110, and 1111 are interpreted as "+"; 1101 is interpreted as "-". The hardware always generates 1100 or 1011 for "+". If the descriptor indicates floating point, the last two D's form an 8-bit twos-complement exponent.

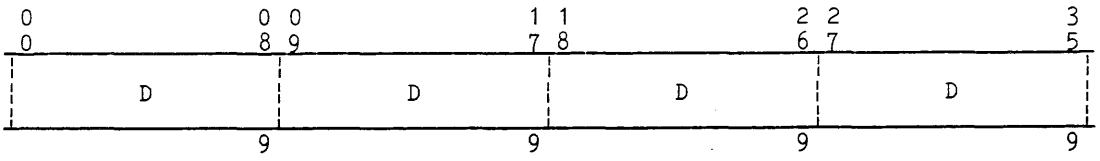


Figure 1-22. EIS Numeric Data Format, 9-Bit Mode

Legend:

Each D represents a digit, a sign, or an exponent. For digits and signs, the low-order 4 bits are interpreted as in 4-bit mode. The hardware always generates octal 060 through 071 for digits, 053 for "+", and 055 for "-". If the descriptor indicates floating point, the low-order 8 bits of the last D is the twos-complement exponent.

DU Pointers and Lengths Format

The following is the format of the eight words of data stored by the spl instruction. The data reflects the exact state of execution of an EIS instruction by the DU. This same data is reloaded into the DU by the lpl instruction.

(There is no include file for the declaration of this data.)

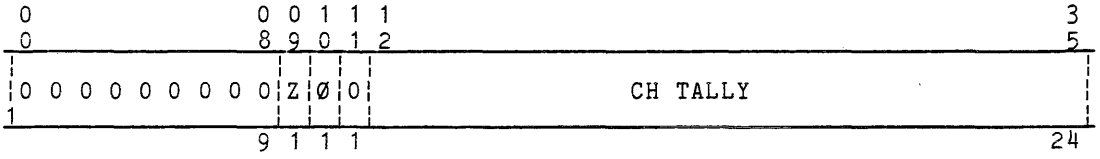


Figure 1-23. DU Pointers and Lengths Format, Word 0

Legend:

Z All bit string instruction results are zero.

Ø Negative overpunch found in 6-4 expanded move.

CH TALLY The number of characters examined by the SCAN, TCT, or TCTR instruction (up to the interrupt or match).

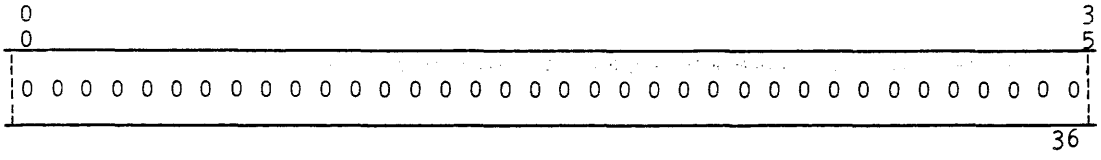


Figure 1-24. DU Pointers and Lengths Format, Word 1

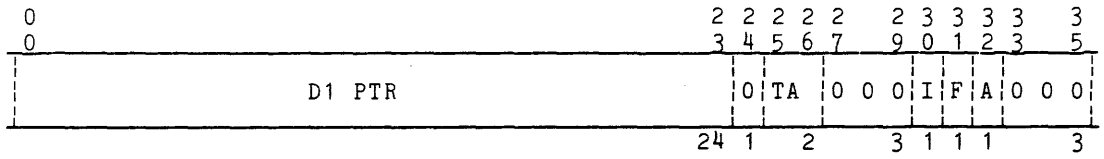


Figure 1-25. DU Pointers and Lengths Format, Word 2

Legend:

D1 PTR Address of last double word accessed by descriptor 1. Bits 17-23 (bit address) valid only for initial access.

TA Alphanumeric type (bits 21-22) of descriptor 1.

I DU interrupted flag; a copy of the MIF fault indicator. This bit is not reloaded by lpl.

F First time. Data in descriptor 1 is valid.

A Descriptor 1 is active.

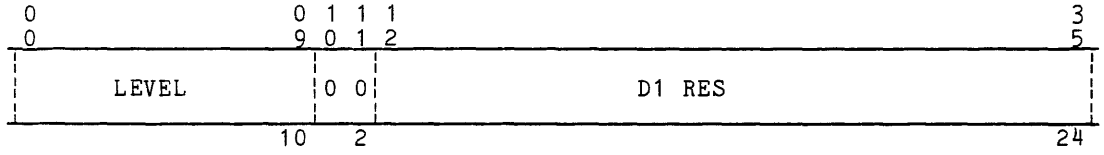


Figure 1-26. DU Pointers and Lengths Format, Word 3

Legend:

LEVEL The difference in the count of characters loaded into the CPU and characters stored back from the CPU.

D1 RES The count of characters remaining in descriptor 1.

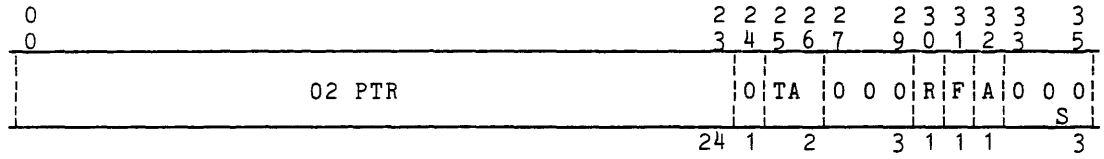


Figure 1-27. DU Pointers and Lengths Format, Word 4

Legend:

D2 PTR Address of the last double word accessed by descriptor 2. Bits 17-23 (bit address) valid only for initial access.

TA Alphanumeric type (bits 21-22) of descriptor 2.

R The last cycle performed must be repeated. This bit is not reloaded by lpl.

F First time. Data in descriptor 2 is valid.

A Descriptor 2 is active.

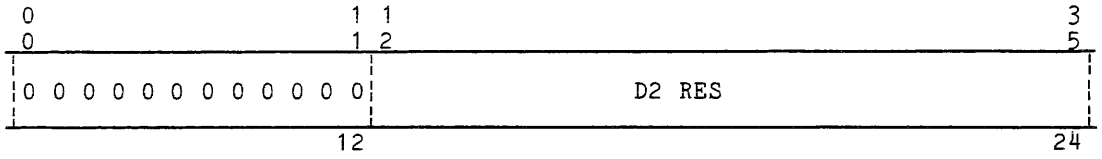


Figure 1-28. DU Pointers and Lengths Format, Word 5

Legend:

D2 RES The count of characters remaining in descriptor 2.

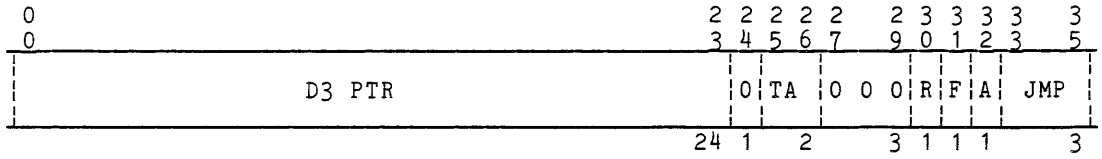


Figure 1-29. DU Pointers and Lengths Format, Word 6

Legend:

D3 PTR Address of the last double word accessed by descriptor 3. Bits 17-23 (bit address) valid only for initial access.

TA Alphanumeric type (bits 21-22) of descriptor 3.

R The last cycle performed must be repeated. This bit is <u>not</u> reloaded by lpl.

F First time. Data in descriptor 3 is valid.

A Descriptor 3 is active.

JMP Number of words to skip to find the next instruction following this multiword instruction.

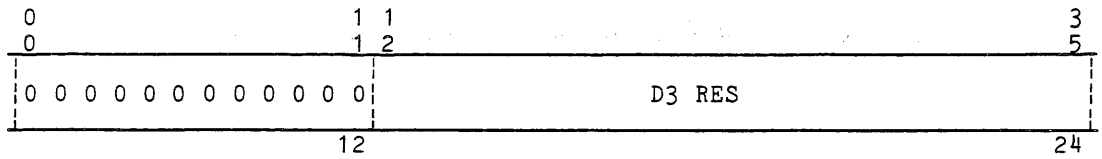


Figure 1-30. DU Pointers and Lengths Format, Word 7

Legend:

D3 RES The count of characters remaining in descriptor 3.

PROCESSOR HISTORY REGISTER FORMATS

Each of the major units of the processor has a set of 16 history registers to store fields and flags from the last 16 execution cycles from that unit. Data is stored in these registers with the scpr instruction.

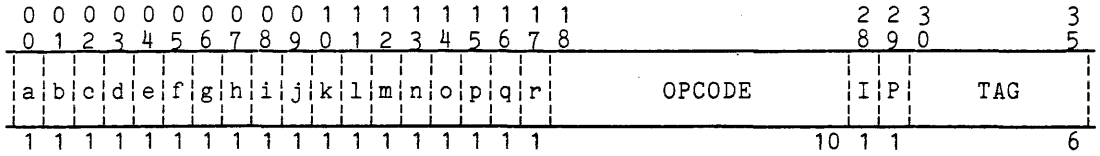
CU History Register Format

The control unit history registers (CU-HR) show the conditions in the CPU control unit for the last 16 CU cycles. Data is entered into the CU history registers at the end of each CU cycle. The last entry shown in a dump of the history registers is the last entry made. True multicycle instructions (such as lpri, lreg, rcu, etc.) will have an entry for each of their cycles.

PL/I Declaration (history_regs.incl.pl1)

```
dcl 1 cuhr      based(cuhrp) aligned,   /* Even word */
   (2 pia       bit(1),
    2 poa       bit(1),
    2 riw       bit(1),
    2 siw       bit(1),
    2 pot       bit(1),
    2 pon       bit(1),
    2 raw       bit(1),
    2 saw       bit(1),
    2 trgo      bit(1),
    2 xde       bit(1),
    2 xdo       bit(1),
    2 ic        bit(1),
    2 rpts      bit(1),
    2 wi        bit(1),
    2 ar        bit(1),
    2 nxip      bit(1),
    2 nflt      bit(1),
    2 np        bit(1),
    2 inst      bit(18),
    2 addr      bit(18),                /* Odd word */
    2 pcmd      bit(5),
    2 psl       bit(4),
    2 xec_int   bit(1),
    2 ins_fetch bit(1),
    2 cus       bit(1),
    2 ous       bit(1),
    2 cul       bit(1),
    2 oul       bit(1),
    2 dir       bit(1),
    2 npcb      bit(1),
    2 pib       bit(1)) unaligned;
```

Even Word:



Odd Word:

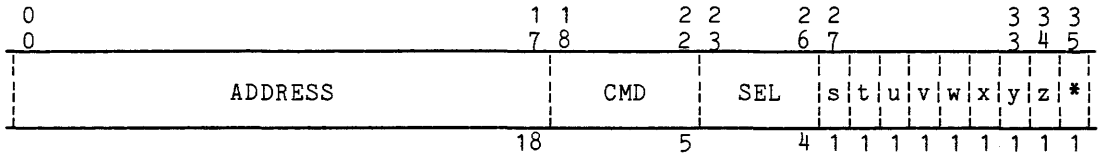


Figure 1-31. CU History Register Format

Legend:

Mask	Key	Field	Meaning
400000	a	PIA	(cuhr.pia) preparing instruction address.
200000	b	POA	(cuhr.poa) preparing operand address.
100000	c	RIW	(cuhr.riw) requesting indirect word.
040000	d	SIW	(cuhr.siw) restoring indirect word.
020000	e	POT	(cuhr.pot) preparing operand tally.
010000	f	PON	(cuhr.pon) preparing operand no tally.
004000	g	RAW	(cuhr.raw) requesting read-alter-rewrite word.
002000	h	SAW	(cuhr.saw) restoring read-alter-rewrite word.
001000	i	TRGO	(cuhr.trgo) transfer GO (conditions met).
000400	j	XDE	(cuhr.xde) executing xed from even ICT.
000200	k	XDO	(cuhr.xdo) executing xed from odd ICT.
000100	l	IC	(cuhr.ic) executing odd instruction of the pair.
000040	m	RPTS	(cuhr.rpts) executing a repeat operation.
000020	n	WI	(cuhr.wi) waiting for instruction fetch.
000010	o	AR F/E	(cuhr.ar) 1 = Address register has valid data.
000004	p	$\overline{\text{XIP}}$	(cuhr.nxip) NOT preparing XIP address.
000002	q	$\overline{\text{FLT}}$	(cuhr.nflt) NOT preparing fault address.
000001	r	$\overline{\text{BASE}}$	(cuhr.np) NOT slave (BAR) mode.
777400		OPCODE	(cuhr.inst, bits 1-10) current operation code.
000200		I	(cuhr.inst, bit 11) interrupt inhibit bit as in Figure 1-41.
000100		P	(cuhr.inst, bit 12) pointer register flag as in Figure 1-41.
000077		TAG	(cuhr.inst, bits 13-16) current address modifier as in Figure 1-41. This modifier is replaced by the contents of the TAG fields of indirect words as they are fetched during indirect chains.
777777		ADDRESS	(cuhr.addr) value of the current computed address.
760000		CMD	(cuhr.pcmd) modulo 2 SCU command.
017000		SEL	(cuhr.psl) port select bits. (Invalid for CP6090 and CP6100 unless ports A through D are selected.)

Mask	Key	Field	Meaning
000400	s	XEC-INT	(cuhr.xec_int) an interrupt is present.
000200	t	INS-FETCH	(cuhr.inst_fetch) performing an instruction fetch.
000100	u	CU-STORE	(cuhr.cus) CU store cycle.
000040	v	OU-STORE	(cuhr.ous) OU store cycle.
000020	w	CU-LOAD	(cuhr.cul) CU load cycle.
000010	x	OU-LOAD	(cuhr.oul) OU load cycle.
000004	y	DIRECT	(cuhr.dir) direct cycle.
000002	z	$\overline{\text{PC-BUSY}}$	(cuhr.npcb) port control logic not busy.
000001	*	BUSY	(cuhr.pib) port interface busy.

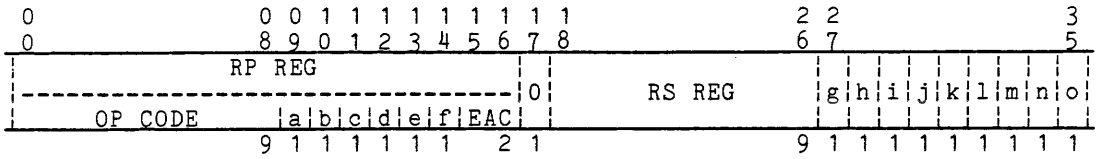
OU History Register Format

The operations unit history registers (OU-HR) show the conditions in the CPU operations unit for the last 16 OU cycles. Data is entered at the end of each OU operation or at the occurrence of a fault. The last entry shown in a dump of the OU history registers is the last entry made. The OU and CU history registers run asynchronously.

PL/I Declaration (history_regs.incl.pl1)

```
dcl 1 ouhr  based(ouhrp) aligned,     /* Even word */
   (2 nopc  bit(9),
    2 itw   bit(1),
    2 ntg   bit(3),
    2 cmod  bit(1),
    2 dir   bit(1),
    2 efad  bit(2),
    2 pad0  bit(1),
    2 rp    bit(9),
    2 opbf  bit(1),
    2 frpf  bit(1),
    2 srf   bit(1),
    2 fgin  bit(1),
    2 fgos  bit(1),
    2 fgd1  bit(1),
    2 fgd2  bit(1),
    2 fgoe  bit(1),
    2 fgoa  bit(1),
    2 fgom  bit(1),                   /* Odd word */
    2 fgon  bit(1),
    2 fgof  bit(1),
    2 fstr  bit(1),
    2 dn    bit(1),
    2 an    bit(1),
    2 qn    bit(1),
    2 x0n   bit(1),
    2 x1n   bit(1),
    2 x2n   bit(1),
    2 x3n   bit(1),
    2 x4n   bit(1),
    2 x5n   bit(1),
    2 x6n   bit(1),
    2 x7n   bit(1),
    2 pad1  bit(3),
    2 ict   bit(18)) unaligned;
```

Even Word:



Odd Word:

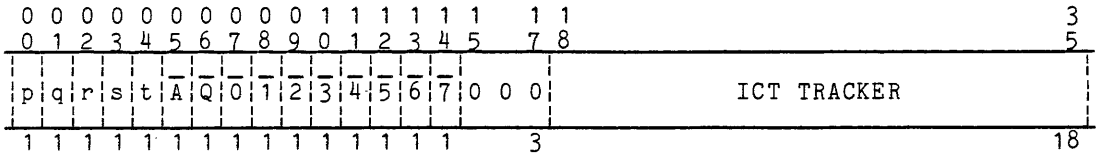


Figure 1-32. OU History Register Format

Legend:

Mask	Key	Field	Meaning
777776		RP REG	primary OU operation register. RP REG receives the instruction operation code and other data from the CU during the CU instruction cycle while the OU may be be busy with a prior operation. RP REG is further substructured as described below.
777000		OP CODE	(ouhr.nopc) the nine MSB of the operation code for the instruction. Basic (non-EIS) operations do not involve bit 27; hence, the 9-bit field is sufficient to define the operation code.
000400	a	9 CHAR	(ouhr.itw) character size for IT mods. 0 = 6-bit character. 1 = 9-bit character.
000340	b,c,d	TAG1,2,3	(ouhr.ntg) three LSB of the modifier of the instruction. This field may contain a character position for an IT character modifier.
000020	e	CR FLG	(ouhr.cmod) character operation flag.
000010	f	DR FLG	(ouhr.dir) direct operation flag.
000006		EAC	(ouhr.efad) effective address counter for lreg/sreg instructions.
777000		RS REG	(ouhr.rp) secondary OU operation register. OP CODE is moved from RP REG to RS REG during the operand fetch and is held until completion of the instruction.
000400	g	RB1 FULL	(ouhr.opbf) OP CODE buffer full.
000200	h	RP FULL	(ouhr.frpf) RP REG full.
000100	i	RS FULL	(ouhr.srf) RS REG full.
000040	j	GIN	(ouhr.fgin) first cycle for all OU operations.
000020	k	GOS	(ouhr.fgos) second cycle for OU multi-ops.
000010	l	GD1	(ouhr.fgd1) first divide cycle.
000004	m	GD2	(ouhr.fgd2) second divide cycle.
000002	n	GOE	(ouhr.fgoe) exponent compare cycle.
000001	o	GOA	(ouhr.fgoa) mantissa alignment cycle.
400000	p	GOM	(ouhr.fgom) general OU cycle.
200000	q	GON	(ouhr.fgon) normalize cycle.
100000	r	GOF	(ouhr.fgof) final OU cycle.
040000	s	STR OP	(ouhr.fstr) OU store data available.

Mask	Key	Field	Meaning
020000	t	DA-AV	(ouhr.dn) data not available.
010000	A	A-REG	(ouhr.an) A register not in use.
004000	Q	Q-REG	(ouhr.qn) Q register not in use.
002000	0	X0-RG	(ouhr.x0n) X0 not in use.
001000	1	X1-RG	(ouhr.x1n) X1 not in use.
000400	2	X2-RG	(ouhr.x2n) X2 not in use.
000200	3	X3-RG	(ouhr.x3n) X3 not in use.
000100	4	X4-RG	(ouhr.x4n) X4 not in use.
000040	5	X5-RG	(ouhr.x5n) X5 not in use.
000020	6	X6-RG	(ouhr.x6n) X6 not in use.
000010	7	X7-RG	(ouhr.x7n) X7 not in use.
777777		ICT TRACKER	(ouhr.ict) the CU ICT value carried as the current offset to the PSR. Since the CU and OU run asynchronously and overlap is usually enabled, the value of ICT TRACKER may not be the address of the OU instruction currently being executed.

DU History Register Format

The decimal unit history registers (DU-HR) show the conditions in the DU for the last 16 CU cycles (since the DU and CU run synchronously). The format is specified as a collection of 72 separate bits since fields are not defined. A minus sign (-) preceding the flag name indicates that the complement of the flag is shown. Unused bits are stored as binary "1"s.

PL/I Declaration (history_regs.incl.pl1)

```
dcl 1 duhr      based(duhrp) aligned,
   (2 pol       bit(1),
    2 pop       bit(1),
    2 ndesc     bit(1),
    2 seladr    bit(1),
    2 dlendr    bit(1),
    2 dfrst     bit(1),
    2 exr       bit(1),
    2 ldfrst    bit(1),
    2 dulea     bit(1),
    2 dusea     bit(1),
    2 redo      bit(1),
    2 wcws      bit(1),
    2 exh       bit(1),
    2 eseq      bit(1),
    2 einst     bit(1),
    2 durw      bit(1),
    2 pradb0    bit(1),
    2 pradb1    bit(1),
    2 aidesc    bit(3),
    2 wrd       bit(1),
    2 nine      bit(1),
    2 six       bit(1),
    2 four      bit(1),
    2 bit       bit(1),
```

```
    2 du_pad1   bit(4),
    2 samplint  bit(1),
    2 sfcsq     bit(1),
    2 adjlen    bit(1),
    2 intind    bit(1),
    2 inhibstc1 bit(1),
    2 du_pad2   bit(1),
    2 duidl     bit(1),
    2 dcldgt    bit(3),
    2 nopl1     bit(1),
    2 nopgl1    bit(1),
    2 nopl2     bit(1),
    2 nopgl2    bit(1),
    2 aoplg1    bit(1),
    2 aoplg2    bit(1),
    2 lrwrg1    bit(1),
    2 lrwrg2    bit(1),
    2 dataav    bit(1),
    2 rw1rl     bit(1),
    2 numstg    bit(1),
    2 anstg     bit(1),
    2 opav      bit(1),
    2 endseq    bit(1),
    2 len128    bit(1),
    2 charop    bit(1),
    2 anpk      bit(1),
    2 exmop     bit(1),
    2 blnk      bit(1),
    2 du_pad3   bit(1),
    2 bde       bit(1),
    2 dbe       bit(1),
    2 shft      bit(1),
    2 flt       bit(1),
    2 rnd       bit(1),
    2 addsub    bit(1),
    2 multdiv   bit(1),
    2 expon     bit(1),
    2 du_pad4   bit(4))unaligned;
```

Mask	Bit	Field	Meaning
400000	0	- FPOL	(duhr.pol) preparing operand length.
200000	1	- FPOP	(duhr.pop) preparing operand pointer.
100000	2	- NEED-DESC	(duhr.ndesc) need descriptor.
040000	3	- SEL-ADR	(duhr.seladr) select address register.
020000	4	- DLEN=DIRECT	(duhr.dlendr) length equals direct.
010000	5	- DFRST	(duhr.dfrst) descriptor being processed for first time.
004000	6	- FEXR	(duhr.exr) extended register modification.
002000	7	- DLAST-FRST	(duhr.ldfrst) last cycle of DFRST.
001000	8	- DDU-LDEA	(duhr.dulea) DU load.
000400	9	- DDU-STAE	(duhr.dusea) DU store.
000200	10	- DREDO	(duhr.redo) redo operation without pointer and length update.
000100	11	- DLVL<WD-SZ	(duhr.wcws) load with count less than word size.
000040	12	- EXH	(duhr.exh) exhaust.
000020	13	DEND-SEQ	(duhr.eseq) end of sequence.
000010	14	- DEND	(duhr.einst) end of instruction.
000004	15	- DU=RD+WRT	(duhr.durw) DU write-back.
000002	16	- PTRA00	(duhr.pradb0) PR address bit 0.
000001	17	- PTRA01	(duhr.pradb1) PR address bit 1.

Mask	Bit	Field	Meaning
400000	18	FA/I1	descriptor 1 active.
200000	19	FA/I2	descriptor 2 active.
100000	20	FA/I3	descriptor 3 active. (FA/I1,2,3 collected into duhr.aidesc)
040000	21	- WRD	(duhr.wrd) word operation.
020000	22	- NINE	(duhr.nine) 9-bit character operation.
010000	23	- SIX	(duhr.six) 6-bit character operation.
004000	24	- FOUR	(duhr.four) 4-bit byte operation.
002000	25	- BIT	(duhr.bit) single bit operation.
001000	26		not used.
000400	27		not used.
000200	28		not used.
000100	29		not used.
000040	30	FSAMPL	(duhr.samplint) sample for midinstruction interrupt.
000020	31	- DFRST-CT	(duhr.sfcsq) specified first count of a sequence.
000010	32	- ADJ-LENGTH	(duhr.adjlen) adjust length.
000004	33	INTRPTD	(duhr.intind) midinstruction interrupt.
000002	34	- INHIB	(duhr.inhibstc1) inhibit STC1.
000001	35		not used.
400000	36	DUD	(duhr.duidl) DU idle.
200000	37	- GDLDA	descriptor load gate a.
100000	38	- GDLDB	descriptor load gate b.
040000	39	- GDLDC	descriptor load gate c. (GDLDA,B,C collected as duhr.dcldgt)
020000	40	NLD1	(duhr.nopl1) prepare alignment count for first numeric operand load.
010000	41	GLDP1	(duhr.nopgl1) numeric operand one load gate.
004000	42	NLD2	(duhr.nopl2) prepare alignment count for second numeric operand load.
002000	43	GLDP2	(duhr.nopgl2) second numeric operand load gate.
001000	44	ANLD1	(duhr.aoplg1) first alphanumeric operand load gate.
000400	45	ANLD2	(duhr.aoplg2) second alphanumeric operand load gate.
000200	46	LDWRT1	(duhr.lrwrg1) first load rewrite register gate.
000100	47	LDWRT2	(duhr.lrwrg2) second load rewrite register gate.
000040	48	- DATA-AVLDU	(duhr.dataav) DU data available.
000020	49	WRT1	(duhr.rw1rl) first rewrite register loaded.
000010	50	GSTR	(duhr.numstg) numeric store gate.
000004	51	ANSTR	(duhr.anstg) alphanumeric store gate.
000002	52	FSTR-OP-AV	(duhr.opav) operand available to be stored.
000001	53	- FEND-SEQ	(duhr.endseq) end sequence flag.
400000	54	- FLEN<128	(duhr.len128) length less than 128.
200000	55	FGCH	(duhr.charop) character operation gate.
100000	56	FANPK	(duhr.anpk) alphanumeric packing cycle gate.
040000	57	FEXMOP	(duhr.exmop) execute MOP gate.
020000	58	FBLNK	(duhr.blnk) blanking gate.

Mask	Bit	Field	Meaning
010000	59		not used.
004000	60	DGBD	(duhr.bde) binary-to-decimal execution gate.
002000	61	DGDB	(duhr.dbe) decimal-to-binary execution gate.
001000	62	DGSP	(duhr.shft) shift procedure gate.
000400	63	FFLTG	(duhr.flt) floating result flag.
000200	64	FRND	(duhr.rnd) rounding flag.
000100	65	DADD-GATE	(duhr.addsub) add/substract execute gate.
000040	66	DMP+DV-GATE	(duhr.multdiv) multiply/divide execution gate.
000020	67	DXPN-GATE	(duhr.expon) exponent network execution gate.
000010	68		not used.
000004	69		not used.
000002	70		not used.
000001	71		not used.

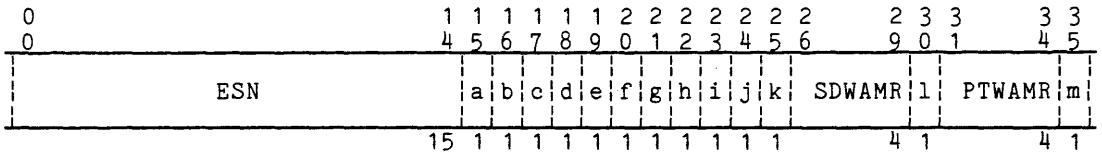
APU History Register Format

The appending unit history registers (APU-HR) show the conditions in the CPU APU for the last 16 address preparation cycles in appending mode.

PL/I Declaration (history_regs.incl.pl1)

```
dcl 1 apuhr     based(aphrp) aligned,      /* Even word */
   (2 esn       bit(15),
    2 bsy       bit(2),
    2 fdsptw    bit(1),
    2 mdsptw    bit(1),
    2 dfsdw     bit(1),
    2 fptw      bit(1),
    2 fptw2     bit(1),
    2 mptw      bit(1),
    2 fanp      bit(1),
    2 fap       bit(1),
    2 sdwmf     bit(1),
    2 sdwamr    bit(4),
    2 ptwmf     bit(1),
    2 ptwamr    bit(4),
    2 flt       bit(1),
    2 add       bit(24),                   /* Odd word */
    2 trr       bit(3),
    2 apu_pad0  bit(3),
    2 cache     bit(1),
    2 apu_pad1  bit(3),
    2 flthld    bit(1),
    2 apu_pad2  bit(1))unaligned;
```

Even Word:



Odd Word:

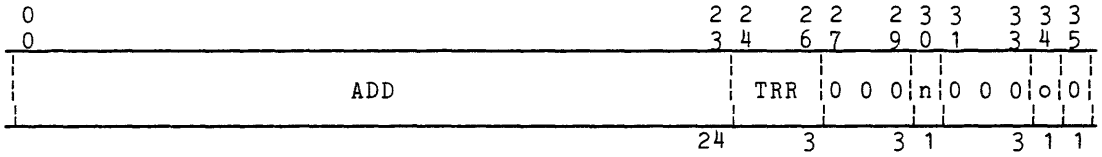


Figure 1-33. APU History Register Format

Legend:

Mask	Key	Field	Meaning
777770		ESN	(apuhr.esn) effective segment number generated.
000006	a,b	BSY	(apuhr.bsy) data source for ESN. 00 = from PSR. 01 = from SNR. 10 = from TSR. 11 = not used.
000001	c	FDSTPW	(apuhr.fdsptw) descriptor segment PTW fetch.
400000	d	MDSPTW	(apuhr.mdsptw) descriptor segment PTW modification.
200000	e	FSDWP	(apuhr.dfsdw) SDW fetch from paged descriptor segment.
100000	f	FPTW	(apuhr.fptw) PTW fetch.
040000	g	FPTW2	(apuhr.fptw2) PTW+1 fetch (prepaging).
020000	h	MPTW	(apuhr.mptw) PTW modification.
010000	i	FANP	(apuhr.fanp) final address fetch from unpaged segment.
004000	j	FAP	(apuhr.fap) final address fetch from paged segment.
002000	k	SDWAMM	(apuhr.sdwmf) SDWAM match ocurred.
001700		SDWAMR	(apuhr.sdwamr) SDWAM register number for SDWAMM=1.
000040	l	PTWAMM	(apuhr.ptwmf) PTWAM match ocurred.
000036		PTWAMR	(apuhr.ptwamr) PTWAM register number for PTWAMM=1.
000001	m	FLT	(apuhr.flt) acv or dftn fault on this cycle.
777777 U 770000 L		ADD	(apuhr.add) 24-bit final address from this cycle.
007000		TRR	(apuhr.trr) ring number from this cycle.
000040	n	CA	(apuhr.cache) segment is encacheable.
000002	o	FHLD	(apuhr.flthld) an acv or dftn is waiting.

FAULT DATA

The processor has 32 active faults in Multics mode and 16 active faults in GCOS mode.

Processor Faults

The fault vector is located at 100(8) and the fault pair for each fault is at 2*OCT relative to 100(8).

Table 1-5. Processor Fault Numbers

Oct	Dec	F/I ADDR in SCU data	Name	Mnemonic	Priority	Group	Mode
0	0	01	Shutdown	sdf	27	VII	M/G
1	1	03	Store	str	10	IV	M/G
2	2	05	Master Mode Entry 1	mme1	11	V	M/G
3	3	07	Fault Tag 1	ftg1	17	V	M/G
4	4	11	Timer Runout	tro	26	VI	M/G
5	5	13	Command	cmd	9	IV	M/G
6	6	15	Derail	drl	15	V	M/G
7	7	17	Lockup	luf	5	IV	M/G
10	8	21	Connect	con	25	VII	M/G
11	9	23	Parity	par	8	IV	M/G
12	10	25	Illegal Procedure	ipr	16	V	M/G
13	11	27	Op Not Complete	onc	4	II	M/G
14	12	31	Startup	suf	1	I	M/G
15	13	33	Overflow	ofl	7	III	M/G
16	14	35	Divide Check	dvck	6	III	M/G
17	15	37	Execute	exc	2	I	M/G
20	16	41	Directed Fault 0	dft0	20	VI	M
21	17	43	Directed Fault 1	dft1	21	VI	M
22	18	45	Directed Fault 2	dft2	22	VI	M
23	19	47	Directed Fault 3	dft3	23	VI	M
24	20	51	Access Violation	acv	24	VI	M
25	21	53	Master Mode Entry 2	mme2	12	V	M
26	22	55	Master Mode Entry 3	mme3	13	V	M
27	23	57	Master Mode Entry 4	mme4	14	V	M
30	24	61	Fault Tag 2	ftg2	18	V	M
31	25	63	Fault Tag 3	ftg3	19	V	M
32	26	65	Unassigned				
33	27	67	Unassigned				
34	28	71	Unassigned				
35	29	73	Unassigned				
36	30	75	Unassigned				
37	31	77	Trouble	trb	3	II	M

Fault Register Format

The CPU fault register contains the conditions in the CPU for several of the hardware faults. The register is stored and cleared by the scpr (tag 01) command. The data is stored into the word pair at location Y and the contents of Y+1 are cleared.

Because the fault register is not used by Multics, there is no software declaration for the data in main memory.

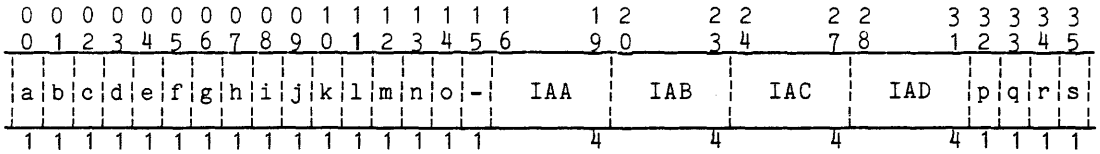


Figure 1-34. Processor Fault Register Format

Legend:

Mask	Key	Field	Meaning
400000	a	ILL OP	illegal operation code.
200000	b	ILL MOD	illegal modifier.
100000	c	ILL SLV	illegal slave procedure.
040000	d	ILL PROC	all other illegal procedures.
020000	e	NEM	nonexistent address.
010000	f	OOB	out of bounds.
004000	g	WRT INH	write inhibit
002000	h	PROC PARU	processor parity upper.
001000	i	PROC PARL	processor parity lower.
000400	j	$CON A	connect to port A.
000200	k	$CON B	connect to port B.
000100	l	$CON C	connect to port C.
000040	m	$CON D	connect to port D.
000020	n	DA ERR1	CPU/SC sequence error 1.
000010	o	DA ERR2	CPU/SC sequence error 2.
000003 U 600000 L		IAA	coded illegal action, port A.
170000		IAB	coded illegal action, port B.
007400		IAC	coded illegal action, port C.
000360		IAD	coded illegal action, port D.
000010	p	CPAR DIR	cache directory parity.
000004	q	CPAR STR	cache storage parity.
000002	r	CPAR IA	illegal action on store.
000001	s	CPAR BLK	cache block parity.

MISCELLANEOUS REGISTER FORMATS

Store Control Unit Data Format

The following is the format of the eight words stored by the scu instruction (the SCU data). The fields marked with (*) are not restored by the restore control unit (rcu) instruction.

PL/I Declaration (mc.incl.pl1)

```
dcl 1 scu              based (scup) aligned,        /* Word 0 */
   (2 ppr,
      3 prr            bit(3),
      3 psr            bit(15),
      3 p              bit(1),
    2 apu,
      3 xsf            bit(1),
```

```
    3 sdwm          bit(1),
    3 sd_on         bit(1),
    3 ptwm          bit(1),
    3 pt_on         bit(1),
    3 pi_ap         bit(1),
    3 dsptw         bit(1),
    3 sdwnp         bit(1),
    3 sdwp          bit(1),
    3 ptw           bit(1),
    3 ptw2          bit(1),
    3 fap           bit(1),
    3 fanp          bit(1),
    3 fabs          bit(1),
  2 fault_cntr      bit(3),                    /* continued below */
```

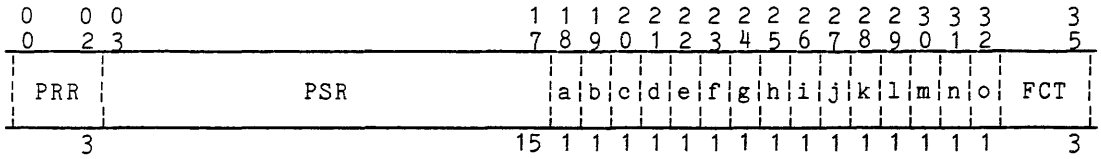


Figure 1-35. SCU Data Format, Word 0

Legend:

Mask	Key		Field	Meaning
700000			PRR	(scu.ppr.prr) procedure ring register.
077777			PSR	(scu.ppr.psr) procedure segment register.
200000	b		XSF	(scu.apu.xsf) external segment flag.
100000	c	*	SDWM	(scu.apu.sdwm) match on SDWAM.
040000	d	*	SD-ON	(scu.apu.sd_on) SDWAM enabled.
020000	e	*	PTWM	(scu.apu.ptwm) match on PTWAM.
010000	f	*	PT-ON	(scu.apu.pt_on) PTWAM enabled.
004000	g	*	PI-AP	(scu.apu.pi_ap) instruction fetch append cycle.
002000	h	*	DSPTW	(scu.apu.dsptw) fetch descriptor segment PTW.
001000	i	*	SDWNP	(scu.apu.sdwnp) fetch SDW, unpaged.
000400	j	*	SDWP	(scu.apu.sdwp) fetch SDW, paged.
000200	k	*	PTW	(scu.apu.ptw) fetch PTW.
000100	l	*	PTW2	(scu.apu.ptw2) fetch prepage PTW.
000040	m	*	FAP	(scu.apu.fap) fetch final address, paged.
000020	n	*	FANP	(scu.apu.fanp) fetch final address, unpaged.
000010	o	*	FABS	(scu.apu.fabs) fetch final address, absolute.
000007			FCT	(scu.fault_cntr) number of unsuccessful attempts to execute the instruction.

```
  2 fd,                                        /* Word 1 */
    3 iro           bit(1),
    3 oeb           bit(1),
    3 e_off         bit(1),
    3 orb           bit(1),
    3 r_off         bit(1),
    3 owb           bit(1),
    3 w_off         bit(1),
    3 no_ga         bit(1),
    3 ocb           bit(1),
    3 ocall         bit(1),
    3 boc           bit(1),
    3 inret         bit(1),
```

```
    3 crt          bit(1),
    3 ralr         bit(1),
    3 am_er        bit(1),
    3 oosb         bit(1),
    3 paru         bit(1),
    3 parl         bit(1),
    3 onc_1        bit(1),
    3 onc_2        bit(1),
  2 port_stat,
    3 ial          bit(4),
    3 iac          bit(3),
    3 con_chan     bit(3),
  2 fi_num         bit(5),
  2 fi_flag        bit(1),                    /* continued below */
```

(Also see declaration for scux in mc.incl.pl1)

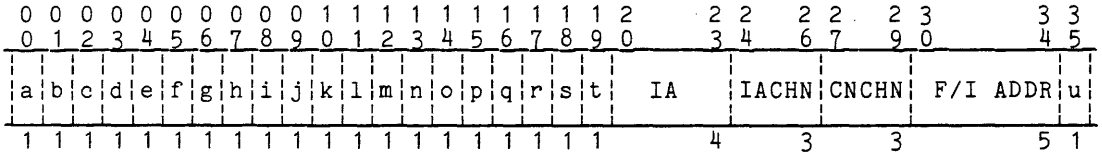


Figure 1-36. SCU Data Format, Word 1

Legend:

Mask	Key		Field	flt addr	Meaning
400000	a	*	IRO	51(acv)	(scu.fd.iro) illegal ring order.
200000	b	*	OEB	51(acv)	(scu.fd.oeb) out of execute bracket.
		*	IOC	25(ipr)	(scux.fd.ioc) illegal op code.
100000	c	*	E-OFF	51(acv)	(scu.fd.e_off) execute bit is off.
		*	IA+IM	25(ipr)	(scux.fd.ia_am) invalid address or modifier.
040000	d	*	ORB	51(acv)	(scu.fs.orb) out of read bracket.
		*	ISP	25(ipr)	(scux.fd.isp) invalid slave procedure.
020000	e	*	R-OFF	51(acv)	(scu.fd.r_off) read bit is off.
		*	IPR	25(ipr)	(scux.fd.ipr) all other illegal procedure.
010000	f	*	OWB	51(acv)	(scu.fd.owb) out of write bracket.
		*	NEA	03(str)	(scux.fd.nea) nonexistent address.
004000	g	*	W-OFF	51(acv)	(scu.fd.w_off) write bit is off.
		*	OOB	03(str)	(scux.fd.oobb) out of bounds.
002000	h	*	NO GA	51(acv)	(scu.fd.no_ga) not a gate, or out-of-call limiter.
001000	i	*	OCB	51(acv)	(scu.fd.ocb) out of call bracket.
000400	j	*	OCALL	51(acv)	(scu.fd.ocall) outward call.
000200	k	*	BOC	51(acv)	(scu.fd.boc) bad outward call.
000100	l	*	INRET	51(acv)	(scu.fd.inret) inward return.
000040	m	*	CRT	51(acv)	(scu.fd.crt) cross ring transfer.

Mask	Key		Field	flt addr	Meaning
000020	n	*	RALR	51(acv)	(scu.fd.ralr) ring alarm.
000010	o	*	AM-ER	51(acv)	(scu.fd.am_er) associative memory error.
000004	p	*	OOSB	51(acv)	(scu.fd.oosb) out of segment bounds.
000002	q	*	PARU	23(par)	(scu.fd.paru) processor parity upper.
000001	r	*	PARL	23(par)	(scu.fd.parl) processor parity lower.
400000	s	*	ONC1	27(onc)	(scu.fd.onc_1) SC/CPU sequence error #1.
200000	t	*	ONC2	27(onc)	(scu.fd.onc_2) SC/CPU sequence error #2.
170000		*	IA	any IA	(scu.port_stat.ial) SC illegal action lines. (see "SC Illegal Action Codes" in Section II.)
007000		*	IACHN	any IA	(scu.port_stat.iac) illegal action CPU port.
000700		*	CNCHN	21(con)	(scu.port_stat.con_chan) connect (CIOC) CPU port.
000076		*	F/I ADDR	any	(scu.fi_num) modulo 2 fault/interrupt vector address.
000001	u	*	F/I	any	(scu.fi_flag) fault/interrupt bit. 0 = interrupt caused the data to be stored. 1 = fault caused the data to be stored.

```
2 tpr,                                          /* Word 2 */
  3 trr          bit(3),
  3 tsr          bit(15),
2 pad2           bit(10),
2 cpu_no         bit(2),
2 delta          bit(6),                        /* continued below */
```

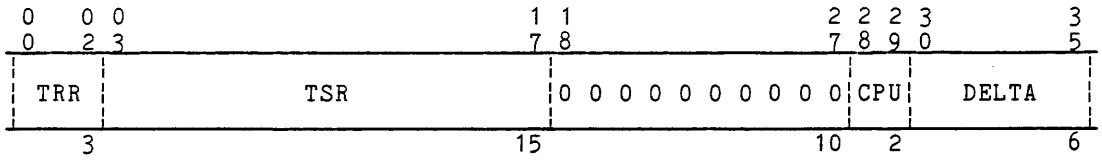


Figure 1-37. SCU Data Format, Word 2

Legend:

TRR (scu.tpr.trr) temporary ring register.

TSR (scu.tpr.tsr) temporary segment register.

* CPU (scu.cpu_no) CPU number (from maintenance panel switches.)

DELTA (scu.delta) address increment for repeats.

```
2 word3          bit(18),                    /* Word 3 */
2 tsr_stat,
  3 tsna,
    4 prn        bit(3),
    4 prv        bit(1),
  3 tsnb,
    4 prn        bit(3),
    4 prv        bit(1),
  3 tsnc,
    4 prn        bit(3),
    4 prv        bit(1),
2 tpr_tbr        bit(6),                     /* continued below */
```

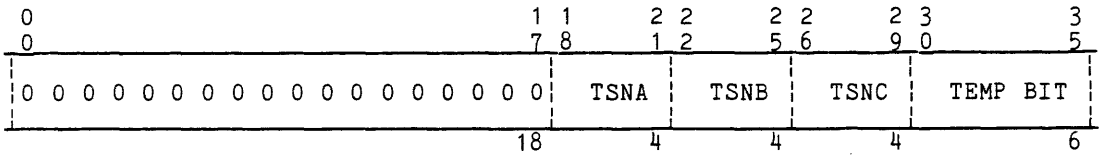


Figure 1-38. SCU Data Format, Word 3

Legend:

TSNA (scu.tsr_stat.tsna...) temporary pointer register number for non-multiword instruction operands or temporary pointer register number for operand descriptor 1 of multiword instructions.

TSN(n)(0-2) pointer register number.
TSN(n)(3) 1 = TSN(n)(0-2) is valid.

TSNB (scu.tsr_stat.tsnb...) temporary pointer register number for operand descriptor 2 of multiword instructions.

TSNC (scu.tsr_stat.tsnc...) temporary pointer register number for operand descriptor 3 of multiword instructions.

TEMP BIT (scu.tpr_tbr) contents of BITNO field of current ITS, ITP, or (if an EIS decimal instruction) ADR pointer.

```
2 ilc            bit(18),                    /* Word 4 */
2 ir,
  3 zero         bit(1),
  3 neg          bit(1),
  3 carry        bit(1),
  3 ovfl         bit(1),
  3 eovf         bit(1),
  3 eufl         bit(1),
  3 oflm         bit(1),
  3 tro          bit(1),
  3 par          bit(1),
  3 parm         bit(1),
  3 bm           bit(1),
  3 tru          bit(1),
  3 mif          bit(1),
  3 abs          bit(1),
  3 pad          bit(4),
```

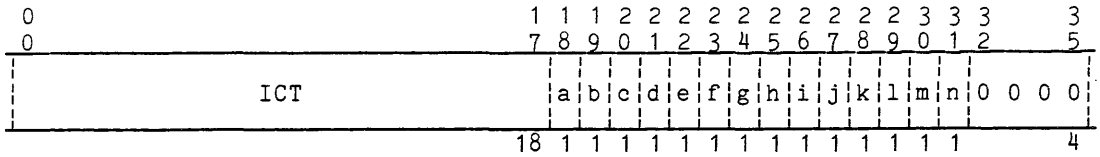


Figure 1-39. SCU Data Format, Word 4

Legend:

Mask	Key	Field	Meaning
777777		ICT	(scu.ilc) instruction counter.
400000	a	ZERO	(scu.ir.zero) zero indicator.
200000	b	NEG	(scu.ir.neg) negative indicator.
100000	c	CARY	(scu.ir.carry) carry indicator.
040000	d	OVFL	(scu.ir.ovfl) overflow indicator.
020000	e	EOVF	(scu.ir.eovf) exponent overflow indicator.
010000	f	EUFL	(scu.ir.eufl) exponent underflow indicator.
004000	g	OFLM	(scu.ir.oflm) overflow mask indicator.
002000	h	TRO	(scu.ir.tro) tally runout indicator.
001000	i	PAR	(scu.ir.par) parity error indicator.
000400	j	PARM	(scu.ir.parm) parity mask indicator.
000200	k	$\overline{BM}$	(scu.ir.bm) BAR mode indicator. If bit is set, CPU is not in BAR (GCOS slave) mode.
000100	l	TRU	(scu.ir.tru) EIS truncation indicator.
000040	m	MIF	(scu.ir.mif) midinstruction fault interrupt (EIS).
000020	n	ABS	(scu.ir.abs) absolute mode.

```
2 ca            bit(18),                    /* Word 5 */
2 cu,
  3 rf          bit(1),
  3 rpt         bit(1),
  3 rd          bit(1),
  3 rl          bit(1),
  3 pot         bit(1),
  3 pon         bit(1),
  3 xde         bit(1),
  3 xdo         bit(1),
  3 poa         bit(1),
  3 rfi         bit(1),
  3 its         bit(1),
  3 if          bit(1),
2 tag           bit(6))  unaligned, /* continued below */
```

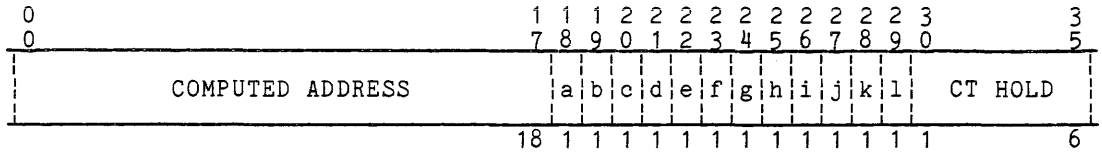


Figure 1-40. SCU Data Format, Word 5

Legend:

Mask	Key	Field	Meaning
777777		* COMPUTED ADDRESS	(scu.ca) effective address value (offset) used in the last address preparation cycle.
400000	a	RF	(scu.cu.rf) first cycle of a repeat operation.
200000	b	RPT	(scu.cu.rpt) executing a repeat.
100000	c	RD	(scu.cu.rd) executing a repeat double.
040000	d	RL	(scu.cu.rl) executing a repeat link.
020000	e	POT	(scu.cu.pot) prepare operand tally. This flag is on until the indirect word of an IT indirect cycle is successfully fetched.
010000	f	PON	(scu.cu.pon) prepare operand notally. This flag is on until the indirect word of a "return" type instruction is successfully fetched. It indicates that there is no indirect chain even though an indirect fetch is being done.
004000	g	XDE	(scu.cu.xde) execute double from even ICT.
002000	h	XDO	(scu.cu.xdo) execute double from odd ICT.
001000	i	ITP	(scu.cu.poa) ITP cycle.
000400	j	RFI	(scu.cu.rfi) restart this instruction at RCU time.
000200	k	ITS	(scu.cu.its) executing ITS indirect cycle.
000100	l	IF	(scu.cu.if) fault occurred during instruction fetch.
000077		CT HOLD	(scu.tag) contents of the "remember modifier" register.

```
2 even_inst     bit (36),          /* Word 6 - continued below */
```

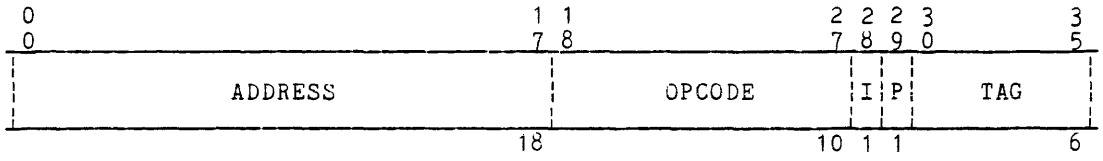


Figure 1-41. SCU Data Format, Word 6

Legend:

ADDRESS current effective 18-bit address.

OPCODE current operation code.

I interrupt inhibit bit.
0 = interrupts permitted.
1 = interrupts inhibited.

P pointer register flag.
0 = Do not use a pointer register for this address preparation cycle. If in appending mode, ADDRESS is an 18-bit offset relative to the procedure segment as specified in PPR.
1 = Decode ADDRESS(0,2) as the number of a pointer register to be used in this address preparation cycle. ADDRESS(3-17) is an offset relative to the base of the segment specified in the pointer register.

TAG current address modifier.

Word 6 is the contents of the "working instruction register" and reflects conditions at the exact point of address preparation when the fault/interrupt occurred. Each instruction of the current pair is moved to this register before actual address preparation begins. At the time this word is stored, it may no longer be identical to the original instruction fetched from memory.

```
2 odd_inst      bit (36);                    /* Word 7 */
```

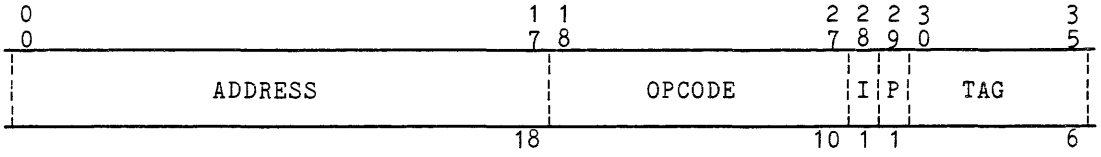


Figure 1-42. SCU Data Format, Word 7

Legend:

ADDRESS address as in word 6.

OPCODE operation code as in word 6.

I interrupt inhibit bit as in word 6.

P pointer register flag as in word 6.

TAG address modifier as in word 6.

Word 7 is the contents of the "instruction holding register". It contains the odd word of the last instruction pair fetched from main memory if word 6 contains the even word. Primarily because of store overlap, this instruction is not necessarily paired with the instruction in word 6.

Read Switches Data Format

The read switches (rsw) instruction provides the ability to interrogate various switches on the processor maintenance and configuration panels. The LSD (bits 15-17) of the instruction address field is used to select the switches to be read. Data is placed in the A register.

rsw xxxxx0:

This is unformatted data; hence, there is no software declaration for the data in main memory.

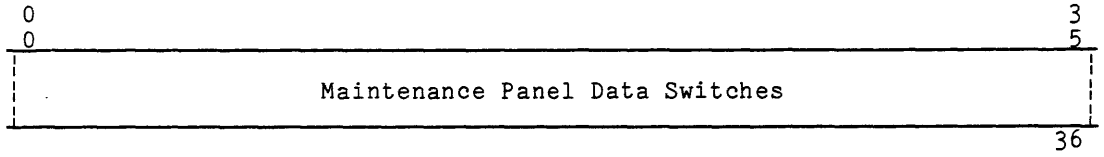


Figure 1-43. rsw xxxxx0 Data Format

rsw xxxxx2:

PL/I Declaration (rsw.incl.pl1)

```
dcl 1 rsw_2          aligned based (rswp),
   (2 mbz            bit(6),
    2 fault_base     bit(7),
    2 mbz2           bit(16),
    2 id             bit(5),
    2 processor_num  bit (2)) unaligned;
```

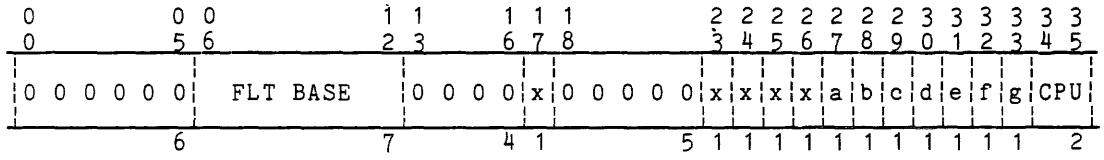


Figure 1-44. rsw xxxxx2 Data Format

Legend:

Key	Field	Meaning
	FLT BASE	(rsw_2.fault_base) seven MSB of the 12-bit fault base address.
x		reserved for future use (presently 0).
a		cache option (not declared). 0 = enabled 1 = disabled
b		extended memory option (not declared). 0 = enabled 1 = disabled
c,d		"01" for EIS cabinet. (keys c,d,e,f,g collected as rsw_2.id)

Key	Field	Meaning
e		EIS option. 0 = enabled. 1 = disabled. (keys c,d,e,f,g collected as rsw_2.id)
f		memory speed option. 0 = slow. 1 = fast. (keys c,d,e,f,g collected as rsw_2.id)
g		overlap option. 0 = no overlap. 1 = overlap. (keys c,d,e,f,g collected as rsw_2.id)
	CPU	(rsw_2.processor_num) processor number.

rsw xxxxx1/3:

For this operation, 1 = ports A,B,C,D and 3 = ports E,F,G,H.

PL/I Declaration (rsw.incl.pl1)

```
dcl 1 rsw_13                aligned based (rswp),
   (2 port_info (0 : 3),
      3 port_assignment     bit(3),
      3 port_enable         bit(1),
      3 initialize_enable   bit(1),
      3 interlace_enable    bit(1),
      3 mem_size            bit(3)) unaligned;
```

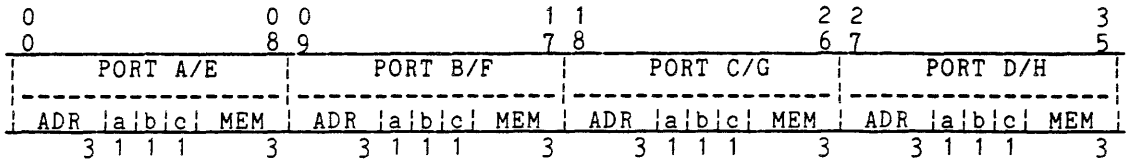


Figure 1-45. rsw xxxxx1/3 Data Format

Legend:

Key	Field	Meaning
	ADR	(rsw_13.port_info.port_assignment) setting of address assignment switches for port.
a		(rsw_13.port_info.port_enable) port enabled flag.
b		(rsw_13.port_info.initialize_enable) initialize control flag.
c		(rsw_13.port_info.interlace_enabled) interlace enabled flag.

Key	Field	Meaning
	MEM	(rsw_13.port_info.mem_size) coded memory size determined as follows:

000	32K
001	64K
010	96K or 160K
011	128K
100	512K
101	1024K
110	2048K
111	256K

rsw xxxxx4:

PL/I Declaration (rsw.incl.pl1)

```
dcl 1 rsw_4                  aligned based (rswp),
   (2 mbz1                   bit(13),
    2 port_interlace (0 : 7),
      3 four                 bit(1),
      3 half_controller      bit(1),
    2 mbz2                   bit(7)) unaligned;
```

```
  0                          1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2          3
  0                          2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9          5
 ____________________________________________________________________________
 |                           | A | B | C | D | E | F | G | H |             |
 |0 0 0 0 0 0 0 0 0 0 0 0 0|---|---|---|---|---|---|---|---|0 0 0 0 0 0 0|
 |                           |a|b|a|b|a|b|a|b|a|b|a|b|a|b|a|b|             |
 ____________________________________________________________________________
                            13 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1            7
```

Figure 1-46. rsw xxxxx4 Data Format

Legend:

Key	Field	Meaning
	A,B,..,H	tags for the eight processor ports.
a		(rsw4.port_interlace.four) interlace mode for port. 1 = 4-word interlace, if interlace enabled for port. 0 = 2-word interlace, if interlace enabled for port.
b		(rsw4.port_interlace.half_controller) memory range for port. 0 = full range. 1 = half range, but "size" is taken as full.

Mode Register and Cache Mode Register Formats

The mode register and cache register contain several dynamic CPU and cache conditions and program accessible controls. The cache mode register is loaded with the lcpr (tag 02) instruction, and the mode register is loaded with the lcpr (tag 04) instruction. The scpr (tag 06) stores the mode register and the cache mode register in a word pair.

PL/I Declaration (mode_reg.incl.pl1)

```
dcl 1 mrg    based(mrp) aligned,          /* Even word */
   (2 ffv    bit(15),
    2 pad0   bit(1),
    2 top    bit(1),
    2 tam    bit(1),
    2 opcss  bit(10),                     /* See switch bits below */
    2 tcuov  bit(1),
    2 scuop  bit(1),
    2 ehr    bit(1),
    2 ehrrs  bit(1),
    2 test   bit(1),
    2 pad1   bit(2),
    2 emr    bit(1)) unaligned;

dcl 1 mrg_sw    based(mrp) aligned,       /* Switch bits in opcss */
   (2 pad0      bit(18),
    2 scuolin   bit(1),
    2 ssolin    bit(1),
    2 ssdpar    bit(1),
    2 sszacpar  bit(1),
    2 stm       bit(2),
    2 svm       bit(2),
    2 pad3      bit(10))unaligned;
```

(The following is derived from cache_mode_reg.incl.alm)

```
dcl 1 cmr                   based aligned,          /* Odd word */
   (2 address_mask          bit(15),
    2 dir_parity            bit(1),
    2 level_full            bit(1),
    2 pad1                  bit(1),
    2 cache_1_on            bit(1),
    2 cache_2_on            bit(1),
    2 operands_from_cache   bit(1),
    2 inst_from_cache       bit(1),
    2 pad2                  bit(1),
    2 cache_to_reg_mode     bit(1),
    2 store_aside           bit(1),
    2 column_full           bit(1),
    2 rro_mask              bit(2),
    2 pad3                  bit(6),
    2 luf_reg_mask          bit(2))unaligned;
```

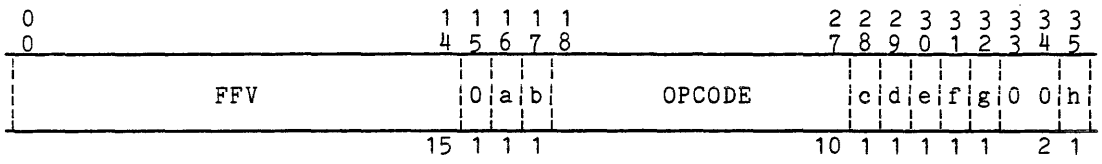


Figure 1-47. Mode Register Format

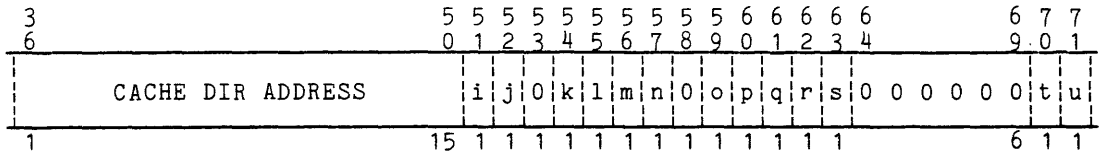


Figure 1-48. Cache Mode Register Format

Legend:

Key	Field	Meaning
	FFV	(mrg.ffv) the "floating fault vector" address. This address consists of the 15 MSB of the modulo 8 base address of four word pairs. These floating faults are generated by other conditions settable by the mode register.
a	OC TRAP	(mrg.top) trap on OPCODE match. (See "Notes" below.) If this bit is set and OPCODE matches the operation code of the instruction for which an address is being prepared (including indirect cycles), generate the second floating fault (xed FFV|2).
b	ADR TRAP	(mrg.tam) trap on FFV match. (See "Notes" below.) If this bit is set and the contents of the address register of the CPU match the setting of the address switches on the maintenance panel, generate the fourth floating fault (xed FFV|6).
	OPCODE	(mrg.opcss) opcode upon which to trap. If either bit 16 (key a) or bit 29 (key d) is set, interpret bits 18-27 as an opcode value. If both bit 16 and bit 29 are not set and bit 32 (key g) is set, interpret bits 18-27 as follows:

bit	meaning
18	(mrg_sw.scuolin) set CU overlap inhibit. The CU waits for the OU to complete execution of the even instruction before it begins address preparation for the associated odd instruction. The CU also waits for the OU to complete execution of the odd instruction before it fetches the next instruction pair.
19	(mrg_sw.ssolin) set store overlap inhibit. The CU waits for completion of a current memory fetch (read cycles only) before requesting a memory access for another fetch.
20	(mrg_sw.ssdpar) set store incorrect data parity. The CU causes incorrect data parity to be sent to the SC for the next data store instruction and then resets bit 20.
21	(mrg_sw.sszacpar) set store incorrect ZAC parity. The CU causes incorrect zone-address-command (ZAC) parity to be sent to the SC for each memory cycle of the next data store instruction and resets bit 21 at the end of the instruction.

Key	Field	Meaning

22,23 (mrg_sw.stm) set timing margins. If bit 32 (key g) is set and the margin control switch on the CPU maintenance panel is in program position, set CPU timing margins as follows:

22,33	margin
0,0	normal
0,1	slow
1,0	normal
1,1	fast

24,25 (mrg_sw.svm) set +5 voltage margins. If bit 32 (key g) is set and the margin control switch on the CPU maintenance panel is in the program position, set +5 voltage margins as follows:

24,25	margin
0,0	normal
0,1	low
1,0	high
1,1	normal

26,27 not used

c — (mrg.tcuov) trap on CU-HR count overflow. (See "Notes" below.)
If this bit is set and bit 30 (key e) is set and the CU-HR counter overflows, generate the third floating fault (xed FFV|4). Further, if bit 31 (key f) is set, reset bit 30, locking the history registers. An lcpr instruction setting bit 28 resets the CU-HR counter to zero.

d — O-C ¢ — (mrg.scuop) strobe CU-HR on OPCODE match.
If this bit and bit 30 (key e) are set and the operation code of the current instruction matches OPCODE, strobe the CU-HR on all CU cycles (including indirect cycles).

e — STROBE ¢ — (mrg.ehr) enable history registers.
If this bit is set, all history registers are strobed at appropriate points in the various CPU cycles. If this bit is reset or bit 35 (key h) is reset, all history registers are locked. This bit is reset with an lcpr instruction providing a 0 bit, an onc fault, and, conditionally, by other faults (see bit 31 (key f) below). Once reset, the bit must be set with an lcpr instruction providing a 1 bit before the history registers again become active.

f — FAULT — (mrg.ehrrs) history register lock control.

RESET — If this bit is set, reset bit 30, locking the history registers, for the following faults and conditions:

luf	Lockup Fault
par	Parity Fault
cmd	Command Fault
str	Store Fault
ipr	Illegal Procedure Fault
sdf	Shutdown Fault
	OPCODE trap
	CU-HR counter overflow trap
	Address match trap

Key	Field	Meaning
g	¢ VOLTAGE	(mrg.test) test mode indicator. This bit is set whenever the TEST/NORMAL switch on the maintenance panel is in TEST position and is reset otherwise. It serves to enable the program control of voltage and timing margins.
h	MR ENABLE	(mrg.emr) enable mode register. When this bit is set, all other bits and controls of the mode register are active. When this bit is reset, the mode register controls are disabled.
	CACHE DIR ADDRESS	(cmr.address_mask) cache block address from cache directory.
i	PAR BIT	(cmr.dir_parity) cache directory parity bit.
j	LEV FUL	(cmr.level_full) cache level is full.
k	CSH1 ON	(cmr.cache_1_on) first two columns of the cache are enabled.
l	CSH2 ON	(cmr.cache_2_on) second two columns of the cache is enabled.
m	OPND ON	(cmr.operands_from_cache) cache is enabled for operands.
n	INST ON	(cmr.inst_from_cache) cache is enabled for instructions.
o	CSH REG	(cmr.cache_to_reg_mode) enabled cache/register.
p	STR ASD	(cmr.store_aside) store aside enabled.
q	COL FUL	(cmr.column_full) column is full.
r	RRO A	round robin counter, bit A.
s	RRC B	round robin counter, bit B. (RRO A,B collected as cmr.rro_mask)
t	LUF MSB	lockup timer setting, most significant bit.
u	LUF LSB	lockup timer setting, least significant bit. (LUF MSB,LSB collected as cmr.luf_reg_mask)

Notes

1. These traps (address match, OPCODE match, CU-HR counter overflow) occur after completion of the next odd instruction following their detection. They are handled as Group VII faults in regard to servicing and inhibition. The complete Group VII priority sequence is:

   1 - con
   2 - tro
   3 - sdf
   4 - OPCODE trap
   5 - CU-HR counter overflow
   6 - Address match trap
   7 - External interrupts

2. The COL FUL, RRO A, RRO B, and CACHE DIR ADDRESS fields reflect different locations in cache depending on the final (absolute) address of the scpr instruction storing this data.

SECTION II

SERIES 60 SYSTEM CONTROLLER AND MEMORY

This section gives the format of the program accessible registers of the Series 60 Level 66 Controller (SCU), the Level 68 System Controller (SC), and their associated memory.

SYSTEM CONTROLLER ILLEGAL ACTION CODES

The following are illegal action codes for the SC.

Code	Priority	CPU flt	Name
00			no illegal action.
01		12(cmd)	unassigned.
02	5	02(str)	nonexistent address.
03	1	12(cmd)	stop on condition. (Level 68 only)
04		12(cmd)	unassigned.
05	12	22(par)	data parity, store to SC.
06	11	22(par)	data parity in store.
07	10	22(par)	data parity in store and store to SC.
10	4	12(cmd)	not control. (Level 68 only)
11	13	12(cmd)	port not enabled.
12	3	12(cmd)	illegal command.
13	7	02(str)	store not ready.
14	2	22(par)	ZAC parity, active module to SC.
15	6	22(par)	data parity, active module to SC.
16	8	22(par)	ZAC parity, SC to store.
17	9	22(par)	data parity, SC to store.
00	12		no illegal action.
--	--	12(cmd)	not used by system controller.
02	5	02(str)	nonexistent address.
--	--	12(cmd)	not used.
--	--	12(cmd)	not used.
05	10	22(par)	data parity, store to SCU.
06	9	22(par)	data parity in store.
07	8	22(par)	data parity in store and store to SCU.
--	--	12(cmd)	not used.
11	11	12(cmd)	port masked.
12	2	12(cmd)	illegal command.
13	5	02(str)	store not ready.
14	1	22(par)	ZAC parity, active module to SCU.
15	4	22(par)	data parity, active module to SCU.
16	6	22(par)	ZAC parity, SCU to store unit.
17	7	22(par)	data parity, SCU to store unit.

SYSTEM CONTROLLER REGISTERS FORMAT

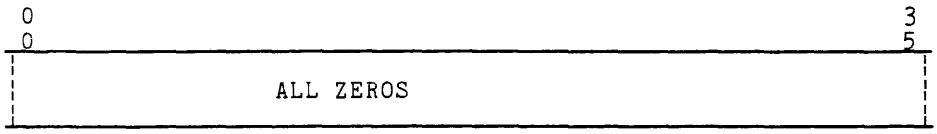
The read system controller register instructions and set system controller register (rscr and sscr) provide the ability to read several registers in SCs and SCUs. The effective absolute address of the instruction selects the SC (or SCU) to be referenced by referring to the port address assignment switches. Bits 3-14 of the instruction address are sent to the SC (or SCU) to specify the register to be referenced. Bits 15-17 are not interpreted since they are used in port selection for normal data and instruction fetches when port interlace is being used. The rscr instruction reads data into the combined A and Q registers of the processor. The sscr instruction sets data from the A and Q registers.

System Controller Mode Register (rscr/sscr 00000X)

PL/I declaration (scr.incl.pl1)

```
dcl 1 scr_mr                aligned,
   (2 pad1                  bit(50),
    2 identification        bit(4),
    2 TS_strobe_margin      bit(2),
    2 GO_strobe_margin      bit(2),
    2 ANSWER_strobe_margin  bit(2),
    2 DA_Strobe_margin      bit(2),
    2 EOC_strobe_margin     bit(2),
    2 PLUS_5_VOLT_margin    bit(2),
    2 parity_override       bit(1),
    2 parity_disable        bit(1),
    2 store_IA_disable      bit(1),
    2 ZAC_parity_error      bit(1),
    2 SGR_accepted          bit(1),
    2 pad2                  bit(1)) unal;
```

Upper Half (A register):



Lower Half (Q register):

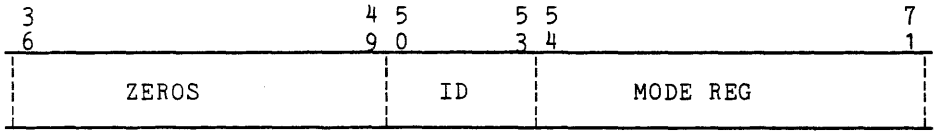


Figure 2-1. Controller Mode Register (rscr/sscr 00000X) Data Format

Legend:

ID (scr.mr.identification) controller ID code.
0000 = 8034, 8035
0001 = Level 68 SC
0010 = Level 66 SCU

MODE REG these fields are used only by T&D.

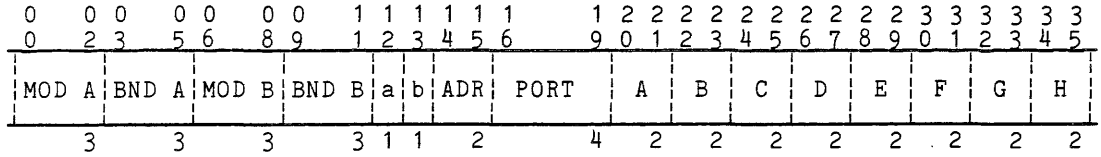
System Controller Configuration Switches (rscr/sscr 00001X)

Note that the configuration switches of an SC cannot be set.

PL/I Declaration (scr.incl.pl1)

```
dcl 1 scr_cfg1          aligned,                    /* Upper half */
   (2 mode_a            bit(3),
    2 bdry_a            bit(3),
    2 mode_b            bit(3),
    2 bdry_b            bit(3),
    2 int               bit(1),
    2 lwr               bit(1),
    2 addr_offset       bit(2),
    2 port_no           bit(4),
    2 port_enable       (0:7) bit(2),
    2 pima              (4) bit(9)) unaligned;  /* Lower half */
```

Upper Half (A register):



Lower Half (Q register):

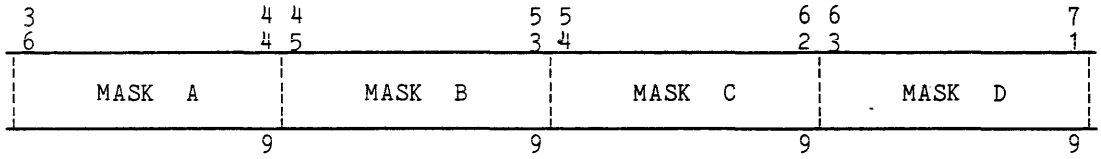


Figure 2-2. SC Configuration Switches (rscr 00001X) Data Format

Legend:

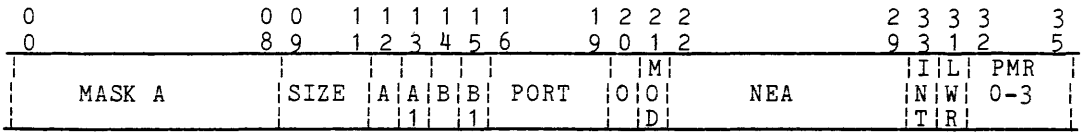
Key	Field	Meaning
	MOD A/B	(scr_cfg1.mode_a/b) state of store A/B. 000 = online. 001 = in test. 010 = offline.
	BND A/B	(scr_cfg1.bdry_a/b) size of memory in store A/B. 000 = 32K. 001 = 64K. 011 = 128K. 111 = 256K.
a		(scr_cfg1.int) interlace flag. 0 = stores are not interlaced. 1 = stores are interlaced.
b		(scr_cfg1.lwr) low-order store flag. 0 = store A is low order. 1 = store B is low order.

Key	Field	Meaning
	ADR	(scr_cfg1.addr_offset) setting of ADDRESS CONTROL OFFSET switch. 00 = no offset. 01 = 16K offset. 10 = 32K offset. 11 = 64K offset.
	PORT	(scr_cfg1.port_no) 4-bit port number of the SC port through which the rscr instruction was received. Port 8 (1000) is the maintenance panel.
	A,B...H	(scr_cfg1.port_enable) port state for each of the eight SC ports. 00 = port disabled. 01 = port in program control. 11 = port enabled.
	MASK A,...,MASK D	(scr_cfg1.pima) EXECUTE INTERRUPT MASK ASSIGNMENT (EIMA) switch settings, i.e., port assignment for each of the four execute interrupt masks. The assigned port corresponds to the bit position within the field. Absence of a bit indicates that the mask is not assigned. Port 8 is the maintenance panel.

PL/I declaration (scr.incl.pl1)

```
dcl 1 scr_cfg2              aligned,
   (2 mask_a_assign         bit(9),
    2 a_online              bit(1),
    2 a1_online             bit(1),
    2 b_online              bit(1)
    2 b1_online             bit(1),
    2 port_no               bit(4),
    2 pad1                  bit(1),
    2 mode                  bit(1)
    2 nea_enabled           bit(1),
    2 nea                   bit(7),
    2 int                   bit(1),
    2 lwr                   bit(1),
    2 port_mask_0_3         bit(4),
    2 mask_b_assign         bit(9),
    2 pad2                  bit(12),
    2 cyclic_prior          bit(7),
    2 pad3                  bit(4),
    2 port_mask_4_7         bit(4)) unal;
```

Upper Half (A register):



Lower Half (B register):

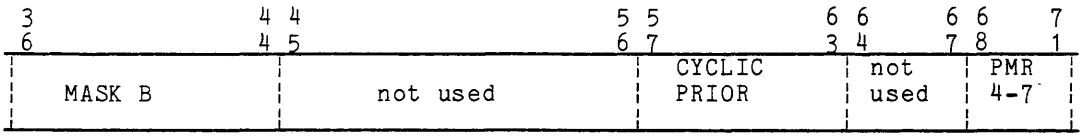


Figure 2-1. SCU Configuration Switches (rscr/sscr 00001x Data Format)

Legend:

MASK A (scr_cfg2.mask_a_assign) EIMA switch setting for mask A. The assigned port corresponds to the bit position within the field. A bit in position 9 indicates that the mask is not assigned.

SIZE (scr.cfg2.size) size of lower store.

000 = 32K
001 = 64K
010 = 128K
011 = 256K
100 = 512K
101 = 1M
110 = 2M
111 = 4M

A (scr_cfg2.a_online) store unit A online.

A1 (scr_cfg2.a1_online) store unit A1 online.

B (scr_cfg2.b_online) store unit B online.

B1 (scr_cfg2.B1_online) store unit B1 online.

PORT (scr_cfg.port_no) 4-bit port number of the SCU port through which the rscr instruction was received. This field cannot be set with the sscr instruction.

MOD (scr_cfg2.mode) program/manual mode. If this bit is a 1, all settable bits of the configuration register may be altered. This bit cannot be set with the sscr instruction.

NEA (scr_cfg2.nea_enabled and scr_cfg2.nea) nonexistent address enable bit and nonexistent address. The first nonexistent address is 32,768 times the switch setting.

INT (scr_cfg2.int) interlace flag.
0 = stores are not interlaced.
1 = stores are interlaced.

LWR (scr_cfg2.lwr) low-order store flag.
0 = store A is low-order.
1 = store B is low-order.

PMR0-3 (scr_cfg2.port_mask_0_3) port enable register for ports 0 through 3.

MASK B (scr_cfg2.mask_b_assign) EIMA switch setting for mask B. (See mask A above.)

CYCLIC PRIOR (scr_cfg2.cyclic_prior) settings of the cyclic port priority ("anti-hogging") switches.

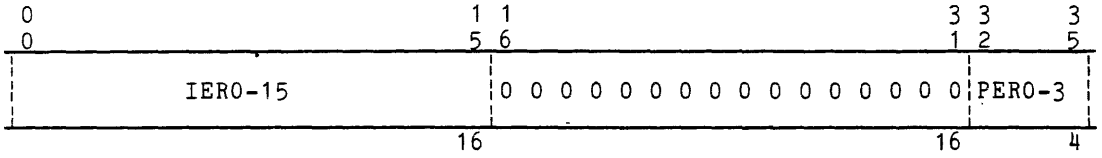
PRM 4-7 (scr_cfg2.port_mask_4_7) port enable register for ports 4 through 7.

System Controller Interrupt Mask Register (rscr/sscr 000N2X)

PL/I declaration (scr.incl.pl1)

```
dcl 1 scr_msk             aligned,
   (2 interrupt_mask_1    bit(16),
    2 pad1                bit(16),
    2 port_mask_1         bit(4),
    2 interrupt_mask_2    bit (16),
    2 pad2                bit(16),
    2 port_mask_2         bit (4)) unal;
```

Upper Half (A register):



Lower Half (Q register):

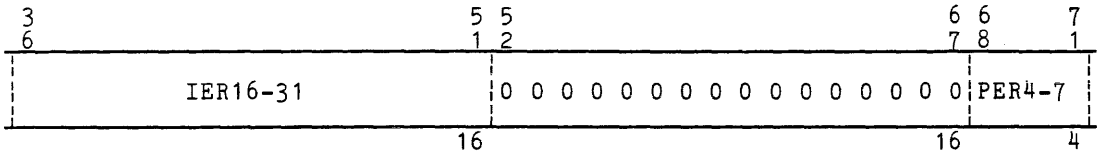


Figure 2-2. Interrupt Mask Register (rscr/sscr 000N2X) Data Format

Legend:

IER0-15 (scr_msk.interrupt_mask_1) program interrupt enable register for interrupts 00 through 15.

PER0-3 (scr_msk.port_mask_1) port enable register for ports 0 through 3. This field is not set by sscr instruction.

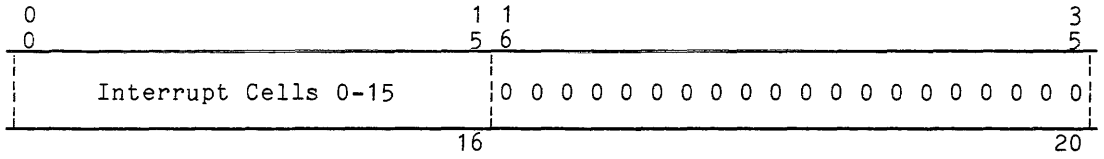
IER16-31 (scr_msk.interrupt_mask_2) program interrupt enable register for interrupts 16 through 31.

PER4-7 (scr_msk.port_mask_2) port enable register for ports 4 through 7. This field is not set by sscr instruction.

System Controller Interrupt Cells (rscr/sscr 00003X)

(There is no include file for the declaration of this data.)

Upper Half (A register):



Lower Half (Q register):

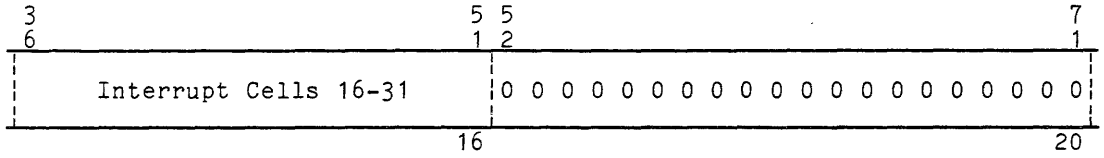


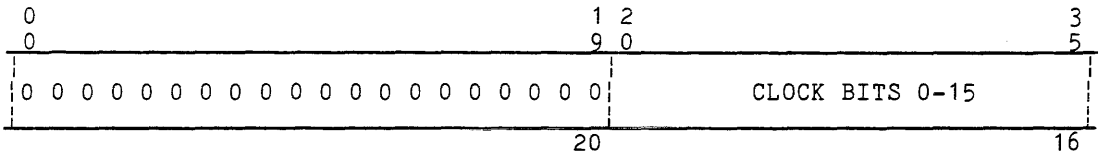
Figure 2-3. Interrupt Cells (rscr/sscr 00003X) Data Format

A bit appearing in any position of the data indicates that the corresponding interrupt cell is set.

System Controller Clock (rscr/sscr 00004X)

(There is no include file for the declaration of this data.)

Upper Half (A register):



Lower Half (Q register):

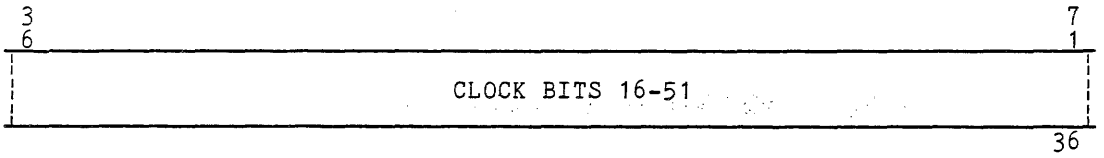


Figure 2-4. System Clock (rscr/sscr/rccl 00004X) Data Format

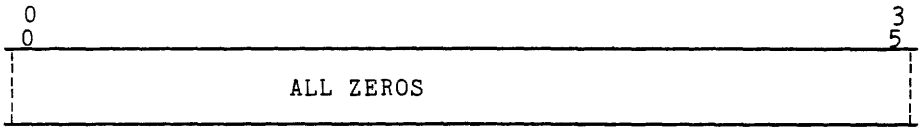
NOTE: The rccl instruction may also be used to read the system clock. The clock in an SC cannot be set with the sscr instruction. It must be set manually. The clock in an SCU cannot be set manually. It must be set using the sscr instruction.

Store Unit Mode Register (rscr/sscr 00006X)

PL/I declaration (scr.incl.pl1)

```
dcl 1 scr_su                    aligned,
   (2 pad1                      bit(36),
    2 ZAC_line                  bit(6),
    2 syndrome                  bit(8),
    2 identification            bit(4),
    2 EDAC_disabled             bit(1),
    2 pad2                      bit(4),
    2 MINUS_5_VOLT_margin       bit(2),
    2 PLUS_5_VOLT_margin        bit(2),
    2 spare_margin              bit(2),
    2 PLUS_19_VOLT_margin       bit(2),
    2 pad3                      bit(1),
    2 SENSE_strobe_margin       bit(2),
    2 pad4                      bit(1),
    2 maint_functions_enabled   bit(1)) unal;
```

Upper Half (A register):



Lower Half (Q Register):

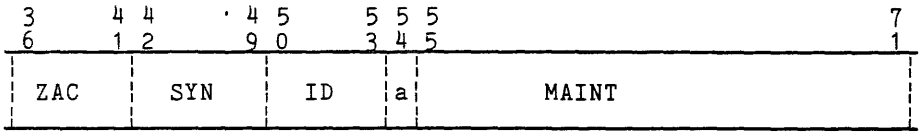


Figure 2-5. Store Mode Register (rscr/sscr 00006X) Data Format

Legend:

Key	Field	Meaning
	ZAC	(scr_su.ZAC_line) address lines.
	SYN	(scr_su.syndrome) failure syndrome.
	ID	(scr_su.identification) store unit type identification. 0000 = high-speed core model AA1. 0001 = high-speed core model AA3. 0100 = 1K chip MOS memory with EDAC enabled. 1100 = 1K chip MOS memory with EDAC disabled. 1111 = 4K chip MOS memory.
a		(scr_su.EDAC_disabled) this bit is turned on when EDAC is disabled.
	MAINT	these fields are used only by T&D.

SECTION III

LEVEL 68 INPUT/OUTPUT MULTIPLEXER

This section gives the formats for the control words and the program accessible registers of the Level 68 Input/Output Multiplexer (IOM).

IOM MAILBOX LAYOUT

The IOM mailbox is a dedicated area in main store used for communication with the IOM and its attached peripherals. Its location is specified by the settings of the INTERRUPT BASE and IOM BASE switches on the IOM configuration panel. Multics currently allows two IOMs and requires that the INTERRUPT BASE for both be set to 1200(8). The IOM BASE settings required are 1400(8) for IOM A and 2000(8) for IOM B.

PL/I Declaration (iom_data.incl.pl1)

```
dcl 1 iom_mailbox$          aligned ext,
    2 imw_array             (32) bit(32),
    2 system_fault_words    (32) bit(36),
    2 spec_status_words     (32) bit(36),
    2 unused                (32) bit(36),
    2 mailboxes             (2),
      3 mailbox             (0:63) like channel_mailbox;
```

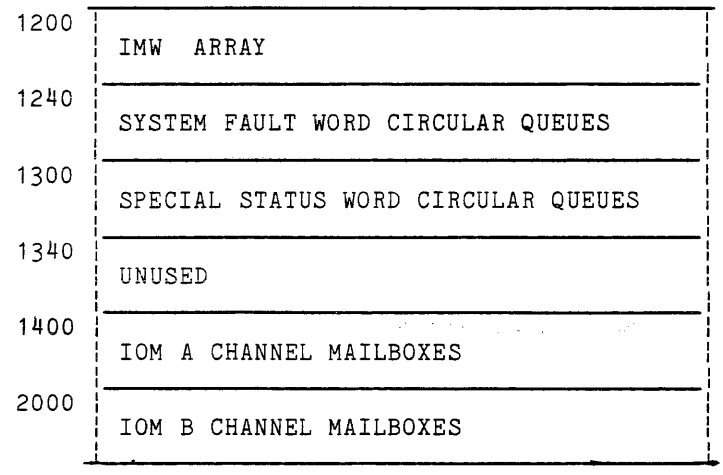


Figure 3-1. IOM Mailbox Layout

The IMW ARRAY is indexed by interrupt number and contains one word for each interrupt. When channel "M" of the IOM signals interrupt "N", the IOM central sets bit "M" of 1200(8)+"N" to 1.

The SYSTEM FAULT WORD CIRCULAR QUEUES contains a 16-word circular queue for the system fault words from each of the allowed IOMs. See "System Fault Status" and Figure 3-9 later in this section.

The SPECIAL STATUS WORD CIRCULAR QUEUES contains a 16-word circular queue for the special status words from each of the allowed IOMs. See "Special Status" and Figure 3-11 later in this section.

The IOM A/B CHANNEL MAILBOXES contain a 4-word mailbox for each of the 64 channels of IOM A/B. See "IOM Channel Mailbox Layout" and Figure 3-2 below.

IOM CHANNEL MAILBOX LAYOUT

Each of the 64 allowed channels of the IOM has a 4-word mailbox located at <IOM_MAILBOX_BASE> + 4*<CHANNEL_NUMBER>.

PL/I Declaration (iom_data.incl.pl1)

```
dcl 1 channel_mailbox based aligned,
    2 lpw              bit(36),
    2 lpwx             bit(36),
    2 scw              bit(36),
    2 dcw              bit(36);
```

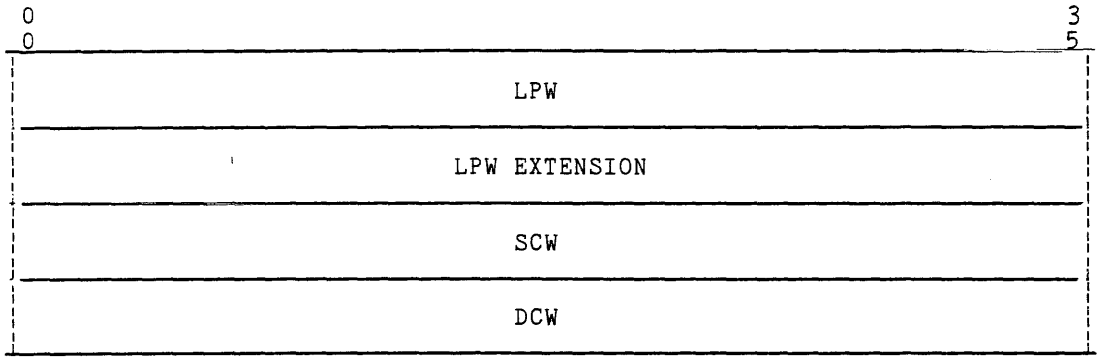


Figure 3-2. IOM Channel Mailbox Layout

Legend:

LPW, LPW EXTENSION
(channel_mailbox.lpw and channel_mailbox.lpwx) See Figure 3-3 below.

SCW (channel_mailbox.scw) See Figure 3-8 below.

DCW (channel_mailbox.dcw) See Figure 3-5 through 3-7 below.

IOM CONTROL WORD FORMATS

List Pointer Word (LPW)

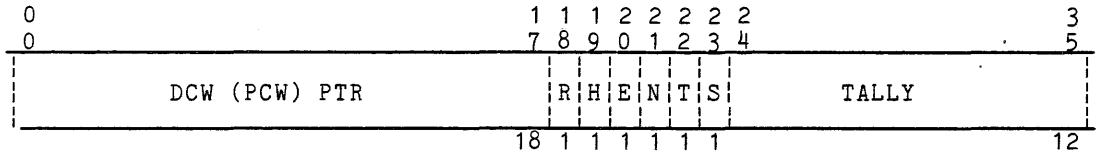
PL/I Declaration (iom_lpw.incl.pl1)

```
dcl 1 lpw        based (lpwp) aligned,
   (2 dcw_addr   bit(18),
    2 res        bit(1),
    2 iom_rel    bit(1),
    2 ae         bit(1),
    2 nc         bit(1),
    2 tal        bit(1),
    2 rel        bit(1),
    2 tally      bit(12)) unal;

dcl 1 lpw_ext based (lpwep) aligned,
   (2 base    bit(9),
    2 bound   bit(9),
    2 idcwp   bit(18)) unal;
```

(Also see Figure 3-2 above.)

LPW:



LPW Extension:

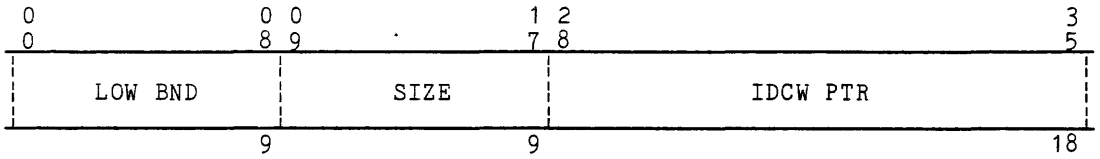


Figure 3-3. IOM List Pointer Word (LPW) Format

Legend:

DCW (PCW) PTR
(lpw.dcw_addr) address of DCW list, or, for connect channel (channel 2) only, PCW address. See Figure 3-4 through 3-7 below.

R (lpw.res) IDCW restrict bit. (The IDCW control bit from every TDCW (tdcw.res) is ORed into this LPW bit. See "Data Control Word" below.)
0 = IDCWs are permitted.
1 = IDCWs are prohibited.

H (lpw.iom_rel) hardware relative addressing bit. A copy of the software relative addressing bit, bit 23, made at IDCW fetch or first list service.

E (lpw.ae) DCW address extension bit. (the address extension control bit from every TDCW (tdcw.ec) is ORed into this bit.)
0 = Fetch DCWs according to the DCW PTR (lpw.dcw_addr) without regard to the address extension value (pcw.ext). All DCWs must reside in lower 256K of store.
1 = Fetch DCWs according to the DCW PTR (lpw.dcw_addr) and the address extension (pcw.ext). DCWs may reside anywhere in main store.

N (lpw.nc) tally control bit.
0 = update TALLY and DCW PTR as DCWs are fetched.
1 = do not update TALLY or DCW PTR.

S (lpw.rel) software relative addressing bit. (The relative addressing control bit from every TCDW (tdcw.rel) is ORed into this LPW bit.)
0 = Perform data transfers to absolute store addresses determined by the address extension value and the DCW data address (dcw.address) without regard to LOW BND and SIZE.
1 = Perform data transfers to store addresses determined by considering the DCW data address as a relative offset to the address extension value and the value of LOW BND. Also, check each DCW data address against the value of SIZE for boundary violations.

T (lpw.tal) tally runout flag.
0 = do not signal tally runout on TALLY exhaust.
1 = signal tally runout on TALLY exhaust.

TALLY (lpw.tally) count of DCWs in list.

LOW BND (lpw_ext.base) mod512 base address for current data transfer.

SIZE (lpw_ext.bound) mod512 bound for current data transfer. (Relative to LOW BND.)

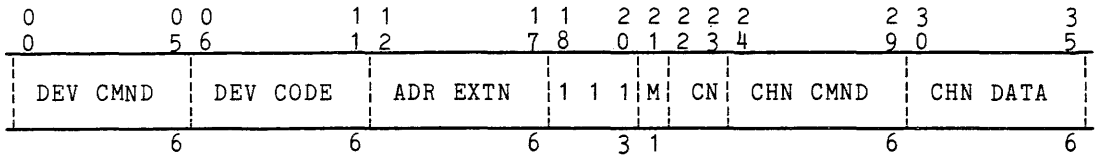
IDCW PTR (lpw_ext.idiwp) address of most recent IDCW.

Peripheral Control Word (PCW)

PL/I Declaration (iom_pcw.incl.pl1)

```
dcl 1 pcw          based (pcwp) aligned,
   (2 command      bit(6),
    2 device       bit(6),
    2 ext          bit(6),
    2 code         bit(3),
    2 mask         bit(1),
    2 control      bit(2),
    2 chan_cmd     bit(6),
    2 count        bit(6),
    2 mbz1         bit(3),
    2 channel      bit(6),
    2 mbz2         bit(27)) unal;
```

Even Word:



Odd Word:

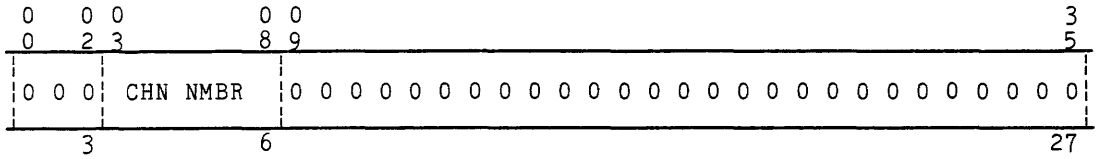


Figure 3-4. IOM Peripheral Control Word (PCW) Format

Legend:

DEV CMND (pcw.command) device command.

DEV CODE (pcw.device) device address.

ADR EXTN (pcw.ext) address extension for addressing beyond 256K.

M (pcw.mask) channel control mask.
0 = normal operation.
1 = mask channel OFF and initialize.

CN (pcw.control) channel control.
00 = terminate at end of I/O operation.
10 = proceed (list service) at end of I/O operation.
11 = set marker interrupt and proceed at end of I/O operation.

CHN CMND (pcw.chan_cmd) channel command.
00 = single record data transfer.
02 = nondata transfer.
06 = multirecord data transfer.
10 = single character record data transfer.

CHN DATA (pcw.count) channel data as required.
(filemark character, backspace count, etc.)

CHN NMBR (pcw.channel) the channel to be connected with this PCW.

Data Control Word (DCW)

There are three types of data control words:

DCW	Data Transmission DCW.
IDCW	Instruction DCW.
TDCW	Transfer DCW.

DATA TRANSMISSION DCW

PL/I Declaration (iom_dcw.incl.pl1)

```
dcl 1 dcw        based (dcwp) aligned,
   (2 address    bit(18),
    2 char_pos   bit(3),
    2 m64        bit(1),
    2 type       bit(2),
    2 tally      bit(12)) unal;
```

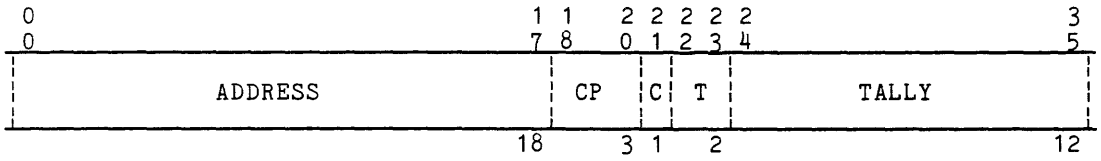


Figure 3-5. IOM Data Transmission DCW Format

Legend:

ADDRESS (dcw.address) data address.

CP (dcw.char_pos) character position address (byte size determined by channel).

C (dcw.m64) tally control.
0 = word tally.
1 = character tally.

T (dcw.type) I/O operation type.
00 = IOTD (transmit and disconnect).
01 = IOTP (transmit and proceed).
11 = IONTP (no transmit and proceed).

TALLY (dcw.tally) element (word or character) count.

INSTRUCTION DCW

PL/I Declaration (iom_pcw.incl.pl1)

```
dcl 1 idcw       based (idcwp) aligned,
   (2 command    bit(6),
    2 device     bit(6),
    2 ext        bit(6),
    2 code       bit(3),
    2 ext_ctl    bit(1),
    2 control    bit(2),
    2 chan_cmd   bit(6),
    2 count      bit(6)) unal;
```

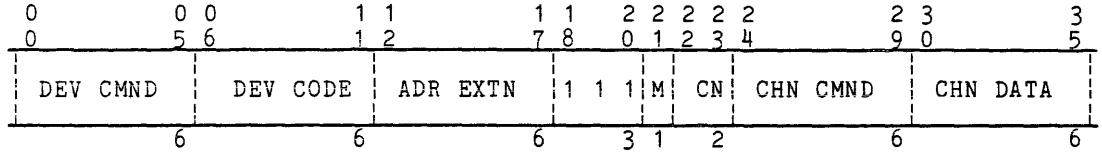


Figure 3-6. IOM Instruction DCW Format

Legend:

DEV CMND (idcw.command) device command.

DEV CODE (idcw.device) device address.

ADR EXTN (idcw.ext) address extension for addressing beyond 256K.

M (idcw.ext_ctl) address extension control.
1 = reset address extension value.
0 = do not reset address extension value.

CN (idcw.control) channel control.
00 = terminate at end of I/O operation.
10 = proceed (list service) at end of I/O operation.
11 = set marker interrupt and proceed at end of I/O operation.

CHN CMND (idcw.chan_cmd) channel command.
00 = single record data transfer.
02 = nondata transfer.
06 = multirecord data transfer.
10 = single character record data transfer.

CHN DATA (idcw.count) channel data as required. (filemark character, backspace count, etc.)

TRANSFER DCW

PL/I Declaration (iom_dcw.incl.pl1)

```
dcl 1 tdcw       based (tdcwp) aligned,
   (2 address    bit(18),
    2 mbz1       bit(4),
    2 type       bit(2),
    2 mbz2       bit(9),
    2 ec         bit(1),
    2 res        bit(1),
    2 rel        bit(1)) unal;
```

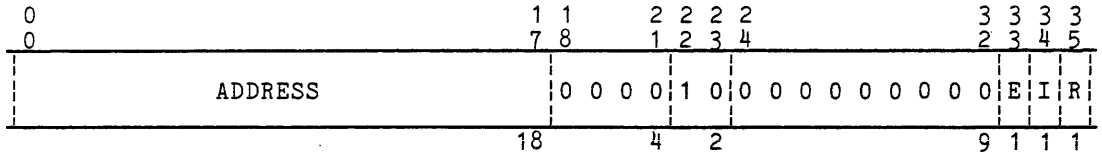


Figure 3-7. IOM Transfer DCW Format

Legend:

ADDRESS (tdcw.address) address of next DCW.

E (tdcw.ec) address extension control, ORed into LPW "E" bit, (See Figure 3-3 above).

I (tdcw.res) IDCW control, ORed into LPW "R" bit (see Figure 3-3 above).

R (tdcw.rel) relative addressing control, ORed into LPW "S" bit (see Figure 3-3 above).

Status Control Word (SCW)

PL/I Declaration (iom_scw.incl.pl1)

```
dcl 1 scw      based (scwp) aligned,
   (2 address  bit(18),
    2 lq       bit(2),
    2 mbz      bit(4),
    2 tally    bit(12)) unal;
```

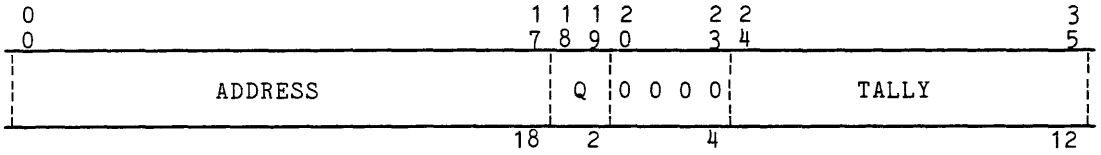


Figure 3-8. IOM Status Control Word (SCW) Format

Legend:

ADDRESS (scw.address) status data address.

Q (scw.lq) status queue control.
00 = store status in normal tallying mode.
01 = store status into a 3-word circular queue.
10 = store status into a 32-word circular queue.
11 = reserved.

TALLY (scw.tally) status tally count.

IOM STATUS WORD FORMATS

System Fault Status

A system fault word is stored as data by the system fault channel (channel 1) of the IOM at the location specified in the fault channel DCW mailbox whenever a system fault is detected by the IOM central.

PL/I Declaration (iom_stat.incl.pl1)

```
dcl 1 faultword          based (statp) aligned,
   (2 mbz1               bit(9),
    2 channel            bit(9),
    2 serv_req           bit(5),
    2 mbz2               bit(3),
    2 controller_fault   bit(4),
    2 io_fault           bit(6)) unal;
```

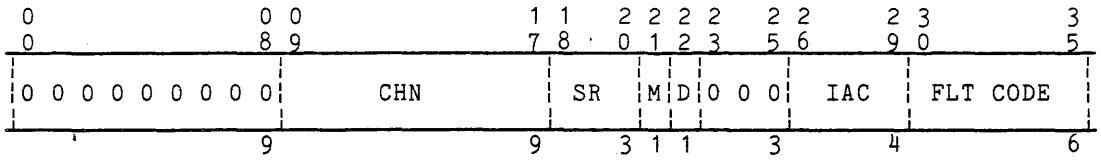


Figure 3-9. IOM System Fault Status Word Format

Legend:

CHN (faultword.channel) channel being serviced when the fault was detected.

SR, M, D (faultword.serv_req) the SR, M, and D fields are decoded together to indicate the service being performed when the system fault occurred.

SR	M	D	service
0	x	x	invalid.
1	1	0	first list service.
	0	x	normal (^first) list service.
	1	1	backup list service.
2	x	x	status service.
3	x	x	program interrupt service.
4	0	0	single-precision indirect data load.
	0	1	double-precision indirect data load.
5	0	0	single-precision indirect data store.
	0	1	double-precision indirect data store.
6	0	0	single-precision direct data load.
	0	1	double-precision direct data load.
	1	0	direct read and clear data load.
7	0	0	single-precision direct data store.
	0	1	double-precision direct data store.

IAC (faultword.controller_fault) illegal action code as received from SC or SCU (See "System Controller Illegal Action Codes" in Section II).

FLT CODE (faultword.io_fault) coded IOM central fault.

Octal value	Meaning
00	no fault.
01	attempt to issue a PCW to a channel with channel number $\geq$ (40)8.
02	a channel requested a service with a service request code of zero, a channel number of zero, or a channel number $\geq$ (40)8. (NOTE: Channel number $\geq$ (40)8 fault is inhibited when IOM is in test.)
03	parity error on the read data when accessing IOM scratchpad.
04	control word address will be incremented to all zeros (256K overflow) and tally will not be decremented to zero.

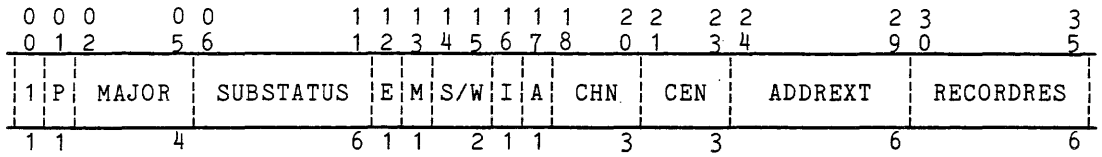
Octal value	Meaning
05	tally was zero for an update LPW (LPW bit 21 = 0) when the LPW was fetched for the connect channel.
06	DCW fetched for the connect channel did not have bits 18-20 = "111"b.
07	DCW fetched for a data service was a TDCW or had bits 18-20 = "111"b.
10	DCW fetched for a 9-bit channel contained an invalid character position.
11	no response to an interrupt from an SC or SCU within 16.5 microseconds.
12	parity error on the read data when accessing an SC or SCU.
13	illegal tally control for an LPW (LPW bits 21-22 = "00"b) when the LPW was fetched for the connect channel.
14	LPW fetched indicates relative address DCWs (LPW bit 23 = "1"b) while operating in Multics mode.
15	fetched a modulo-64 DCW (DCW bit 21 = "1"b) while operating in standard or extended GCOS mode.
16	LPW fetched indicates use of address extension (LPW bit 20 = "1"b) while operating in standard GCOS mode.
17	no port selected during attempt to access main memory.

Channel Status

PL/I Declaration (iom_stat.incl.pl1)

```
dcl 1 status          based (statp) aligned,
   (2 t               bit(1),
    2 power           bit(1),
    2 major           bit(4),
    2 sub             bit(6),
    2 eo              bit(1),
    2 marker          bit(1),
    2 soft            bit(2),
    2 initiate        bit(1),
    2 abort           bit(1),
    2 channel_stat    bit(3),
    2 central_stat    bit(3),
    2 ext             bit(6),
    2 rcount          bit(6),
    2 address         bit(18),
    2 char_pos        bit(3),
    2 r               bit(1),
    2 type            bit(2),
    2 tally           bit(12)) unal;
```

Even Word:



Odd Word:

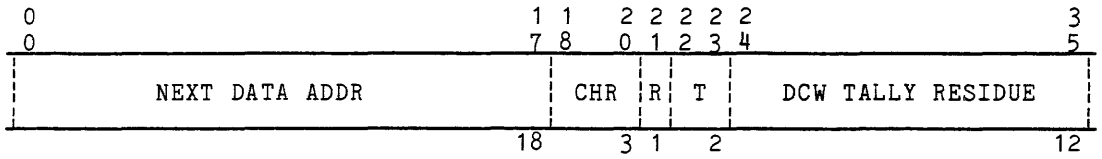


Figure 3-10. IOM Channel Status Data Format

Legend:

P (status.power) device power bit.
0 = device is online and operable.
1 = device is not cabled or is powered off.

MAJOR (status.major) device major status (see Section V or Appendix C under the specific device).

SUBSTATUS (status.sub) device substatus (see Section V or Appendix C under the specific device).

E (status.eo) PSI even/odd bit.
0 = termination occurred after the odd word was stored by a PSI channel operating in binary mode.
1 = termination occurred after the even word was stored by a PSI channel operating in binary mode.
(NOTE: This bit will always be "0" for PSI channels in ASCII mode and for non-PSI channels.)

M (status.marker) marker bit.
0 = initiate/terminate status as per "I" bit described below.
1 = marker interrupt status.

S/W (status.soft) 2-bit field set to 0's by hardware and available for use by software interrupt handler.

I (status.initiate) initiation bit.
0 = terminate/marker status as per "M" bit described above.
1 = initiate status in response to a request status (reqs) or reset status (ress) command.

A (status.abort) software abort bit (set to 0 by hardware).

CHN (status.channel_stat) IOM channel status.

Value	Meaning
0	normal.
1	unexpected PCW (connect while busy).
2	invalid channel instruction in PCW.
3	incorrect DCW on list service.
4	incomplete command sequence.
5	unassigned.
6	parity error at peripheral interface.
7	parity error on I/O bus, data to channel.

CEN (status.central_stat) IOM central status.

Value	Meaning
0	normal.
1	LPW tally runout, not connect channel.
2	two TDCWs.
3	boundary error.
4	address extension change in restricted mode.
5	IDCW in restricted mode.
6	character position/size discrepancy, list service.
7	parity error on I/O bus, data from channel.

ADDREXT (status.ext) address extension value.

RECORDRES (status.rcount) residue in PCW or last IDCW record count field.

NEXT DATA ADDR
(status.address) address of next data word to be transmitted.

CHR (status.char_pos) character position of next character to be transmitted.

R (status.r) read bit.
0 = device is writing.
1 = device is reading.

T (status.type) TYPE field of last DCW.

DCW TALLY RESIDUE
(status.tally) residue in TALLY field of last DCW.

Special Status

A special status word is stored as data by the special status channel (channel 6) of the IOM whenever the appropriate service request is made by a PSI channel. PSI channels store terminate and marker status through their own channel mailboxes, but store status for special interrupts through the special status channel.

PL/I Declaration (iom_stat.incl.pl1)

```
dcl 1 special_status based (statp) aligned,
   (2 t              bit(1),
    2 channel        bit(8),
    2 pad1           bit(3),
    2 device         bit(6),
    2 pad2           bit(1),
    2 byte2          bit(8),
    2 pad3           bit(1),
    2 byte3          bit(8)) unal;
```

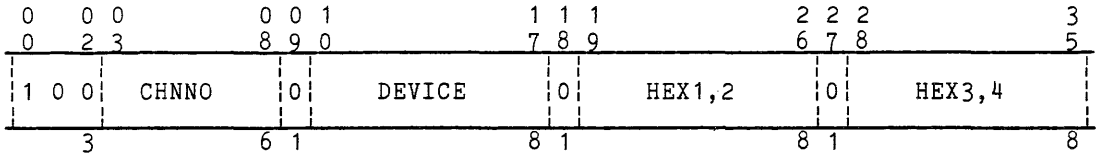


Figure 3-11. IOM Special Status Word Format

Legend:

CHNNO (special_status.channel) the number of the channel storing this special status.

DEVICE (special_status.device) the device address of the device causing the special interrupt.

HEX1,2 (special_status.byte2) the first 8-bit status byte from the MPC.

HEX3,4 (special_status.byte3) the second 8-bit status byte from the MPC.

DEVICE SPECIAL INTERRUPTS

If DEVICE in Figure 3-11 above is nonzero, the special interrupt was caused by a signal from a device attached to the MPC and the status description is as follows:

HEX1,2 (octal)	HEX3,4 (octal)	Meaning
000	000	printer to run: normal.
000	001	disk pack changed or, tape drive(*) malfunction or, reader/punch to ready or, printer to run: print one line.
000	002	disk drive released or, tape drive(*) released or, reader/punch released or, printer to run: forward space.
000	003	printer to run: forward to top.
000	004	tape drive(*) standby loaded or, printer to run: invalid line.
000	005	printer to run: reverse rewind.
000	006	printer to run: backspace.
000	007	printer to run: backspace top.
000	010	tape drive(*) to standby.
000	020	tape drive(*) to ready.
000	040	tape drive(*) unload complete.
000	100	tape drive(*) rewind complete.

(*) status bits from tape drives may be ORed together to show multiple status conditions.

If DEVICE in Figure 3-11 above is zero, the special interrupt was caused by an internal controller condition and the status description is as follows:

HEX1,2 (octal)	HEX3,4 (octal)	Meaning
001	000	suspend command accepted.
002	000	release command accepted.
004	002	completed Test LAELT or CSELT#1.
004	004	completed Test ELT#2.
004	005	completed Test CSELT#2.
004	006	completed Test MMLT.
004	023	completed Test ELT#1.
004	121	completed Test CAITR1 for MTS500.
004	122	completed Test CAITR2 for MTS500.
004	123	completed Test CAITR3 for MTS500.
004	146	completed Test BTLT.
004	312	completed Test CAITR1 or CAITR2 for DSS181/DSS190.
xxx	xxx	if the first two bits of HEX1,2 are "01"b, then the operator has pressed the INTERRUPT key on the MPC and: For DSS190 or URC - setting of thumbwheel switches. For DSS181 or MTS500 - setting of configuration switches.

SECTION IV

LEVEL 68 BULK STORE

This section gives the formats of the control words and status words for the bulk store.

BULK STORE MAILBOX LAYOUT

The bulk store mailbox is a dedicated area in main store used for communication with the Bulk Store Subsystem. Its location is determined by the setting of the CONTROL BASE switches for the port group being used by Multics on the bulk store controller (BSC) configuration panel. Multics currently requires this setting to be 1100(8).

(There is no include file for the declaration of this data.)

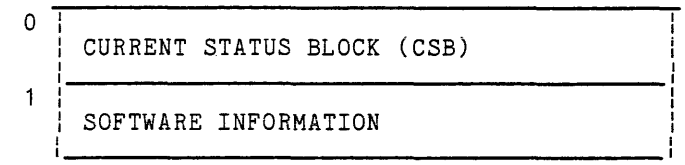


Figure 4-1. Bulk Store Mailbox Layout

CURRENT STATUS BLOCK FORMAT

PL/I Declaration

Word 0:

```
dcl 1 csb          aligned based,
    2 dcb_address  bit(24) unaligned,
    2 rel          bit(1) unaligned,
    2 mbz          bit(6) unaligned,
    2 status       unaligned,
      3 sse        bit(1) unaligned,
      3 nde        bit(1) unaligned,
      3 spe        bit(1) unaligned,
      3 ss         bit(1) unaligned,
      3 busy       bit(1) unaligned,
```

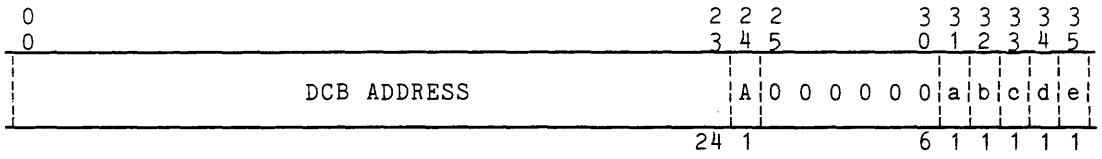


Figure 4-2. Bulk Store Current Status Block (CSB) Format, Word 0

Legend:

Key	Field	Meaning
	DCB ADDRESS	(csb.dcb_address) address of current data control block (DCB).
	A	(csb.rel) relative/absolute bit. 0 = DCB ADDRESS is relative to mailbox base. 1 = DCB ADDRESS is absolute.
a	SSE	(csb.status.sse) status storage error. The BSC was unable to store status properly. DCB ADDRESS contains the address of the DCB for which status was to be stored. Flag applies to both DCB status block storage (see Figures 4-7 through 4-11 following) and single-word status storage. The BSC has halted and reset the BUSY bit (see below).
b	NDE	(csb.status.nde) next DCB error. The BSC was unable to read the NEXT DCB ADDRESS in the DCB referenced by the CSB. The BSC has stopped and reset the BUSY bit.
c	SPE	(csb.status.spe) status pointer error. The BSC was unable to access and use the DCB status pointer. DCB ADDRESS contains the DCB for which status was to be stored. The BSC has stopped and reset the Busy bit.
d	SS	(csb.status.ss) service started. This bit is set to "1"b by the BSC when it responds to a connect. It remains set until the the completion of the service and then is reset.
e	BUSY	(csb.status.busy) busy. 0 = BSC is stopped. 1 = BSC is busy.

DATA CONTROL BLOCK FORMAT

PL/I Declaration

```
dcl 1 dcb (1),
    2 abs_thread          bit(24) unaligned,
    2 rel                 bit(1) unaligned,
    2 mbz                 bit(9) unaligned,
    2 op_started          bit(1) unaligned,
    2 mbz1                bit(1) unaligned,
    2 status,
      3 status_block_ptr  bit(23) unaligned,
      3 rel               bit(1) unaligned,
      3 unused            bit(10) unaligned,
```

```
2 mem_addr          bit(24) unaligned,
2 tally             bit(12) unaligned,
2 store_addr        bit(24) unaligned,
2 control_field     unaligned,
  3 tis             bit(1) unaligned,
  3 tad             bit(1) unaligned,
  3 sps             bit(1) unaligned,
  3 ieo             bit(1) unaligned,
  3 seo             bit(1) unaligned,
  3 mbz             bit(1) unaligned,
  3 int             bit(1) unaligned,
  3 dcw             bit(1) unaligned,
  3 instr           bit(3) unaligned,
```

Word 0:

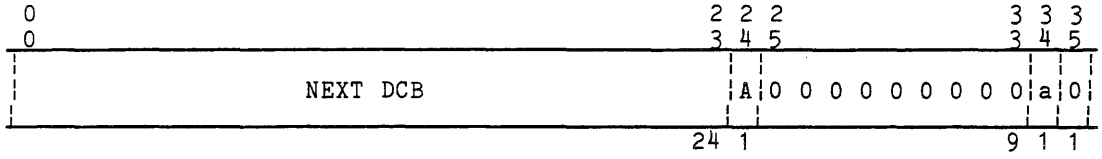


Figure 4-3. Bulk Store Data Control Block (DCB) Format, Word 0

Legend:

Key	Field	Meaning
	NEXT DCB	(dcb.abs_thread) address of next DCB in DCB chain. If address is zero, the BSC stops after completion of this DCB execution.
	A	(dcb.rel) relative/absolute bit. 0 = NEXT DCB is relative to mailbox base. 1 = NEXT DCB is absolute.
a		(dcb.op_started) op_started. A software flag used by the bulk_store_control program to signal that the DCB is active.

Word 1:

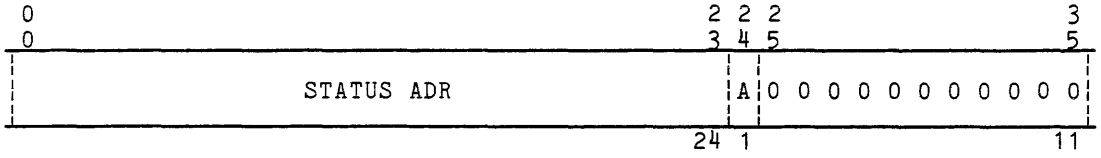


Figure 4-4. Bulk Store Data Control Block (DCB) Format, Word 1

Legend:

STATUS ADR
(dcb.status.status_block_ptr) address of DCB status block.

A (dcb.status.rel) relative/absolute flag.
0 = STATUS ADR is relative to mailbox base.
1 = STATUS ADR is absolute.

Word 2:

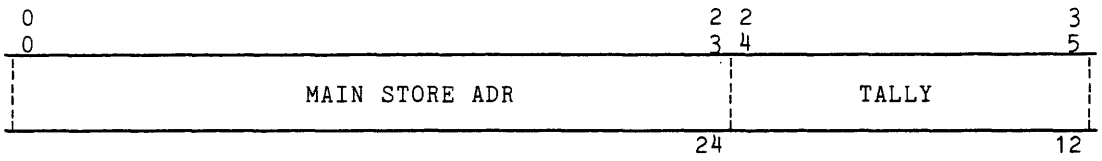


Figure 4-5. Bulk Store Data Control Block (DCB) Format, Word 2

Legend:

MAIN STORE ADR
(dcb.mem_addr) main memory address for data transfer.

TALLY (dcb.tally) tally count for data transfer. (See TIS field in Figure 4-6 below for size of increment.)

Word 3:

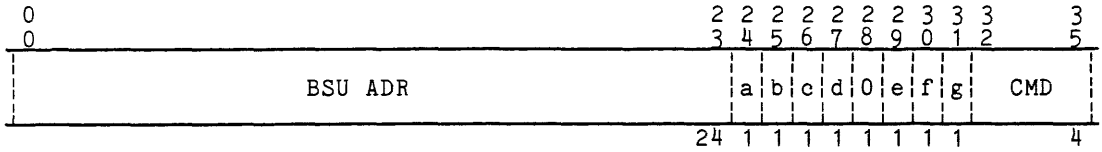


Figure 4-6. Bulk Store Data Control Block (DCB) Format, Word 3

Legend:

Key	Field	Meaning
	BSU ADR	(dcb.store_addr) Bulk Store Unit (BSU) address for data transfer.
a	TIS	(dcb.control_field.tis) tally increment selector. Selects the increment to be used on the TALLY field of DCB Word 2. 0 = 64-word increment. 1 = 1-word increment.
b	T&D	(dcb.control_field.tad) T&D mode indicator. The command in CMD is redefined as a Test & Diagnostic command.
c	SPS	(dcb.control_field.sps) status pointer selector. Used to define the mode of status storage. 0 = store single-word status into DCB, word 1. 1 = store DCB status block at address given in DCB, word 1.
d	IOE	(dcb.control_field.ioe) interrupt on error. If set, the BSC generates a program interrupt at the completion of DCB execution and status storage if the status is other than SUBSYSTEM READY (See Bulk Store Peripheral Status).
e	SOE	(dcb.control_field.soe) stop on error. If set, the BSC will stop at the completion of DCB execution and status storage if the status is other than SUBSYSTEM READY.
f	INT	(dcb.control_field.int) interrupt. If set, the BSC generates a program interrupt at the completion of DCB execution and status storage.

Key	Field	Meaning
g	DCW	(dcb.control_field.dcw) DCW control flag. 0 = DCB, word 2, is a main store address for data transfer. 1 = DCB, word 2, is the main store address of a GCOS type DCW list for data transfer control.
	CMD	(dcb.control_field.inst) coded BSC command.

00	nop
02	load configuration
04	load base and limit (not used by Multics)
05	power off enable
06	read configuration
10	write zeros
11	write
12	write conditional
13	write and verify
14	compare
15	read
16	read nontransfer

(Undefined commands will cause a BSC abort.)

BULK STORE STATUS BLOCK FORMAT

(This bulk store status has the same format whether it appears as word 0 of the status block or as word 1 of the DCB.)

PL/I Declaration

```
dcl 1 dcb status_block,
    2 status              bit(36),
    2 dcw_residue,
      3 abs_addr          bit(24) unaligned,
      3 tally             bit(12) unaligned,
    2 hardware_indicators bit(36),
    2 dcw_pointer         bit(36);
```

Word 0:

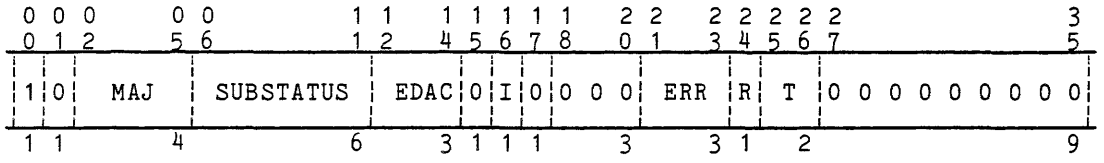


Figure 4-7. Bulk Store DC3 Status Block Format, Word 0

Legend:

MAJOR, SUBSTATUS
: These fields are directly analogous to the MAJOR and SUBSTATUS fields as stored by the IOM. See "Bulk Store Peripheral Status" below for a description of these fields.

EDAC
: error indicators.

I initiation interrupt flag.
Used only with DCB COMMAND REJECT status (See "Bulk Store Peripheral Status" below).

ERR error indicators.

R read flag.
0 = data was read from the BSU to main store.
1 = data was written from main store to the BSU.

T type code.
This field is set to the DCW type if DCWs are used.

Word 1:

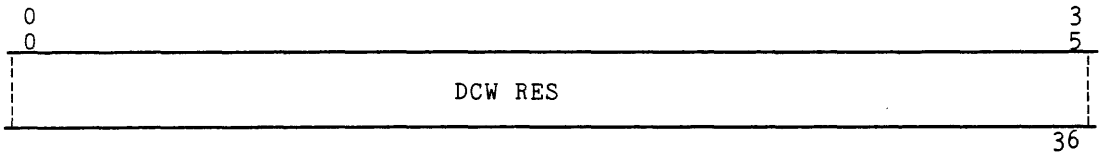


Figure 4-8. Bulk Store DCB Status Block Format, Word 1

Legend:

DCW RESIDUE
This word contains the 24-bit main store address and 12-bit residual tally after the final word of the DCB execution is transmitted.

Word 2:

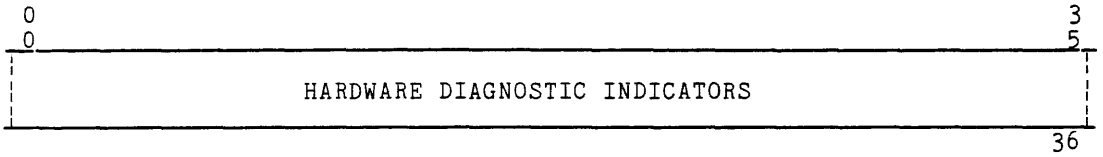


Figure 4-9. Bulk Store DCB Status Block Format, Word 2

Word 3:

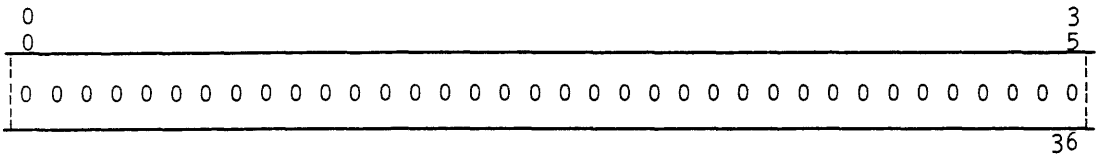


Figure 4-10. Bulk Store DCB Status Block Format, Word 3

Not used by Multics.

Word 4:

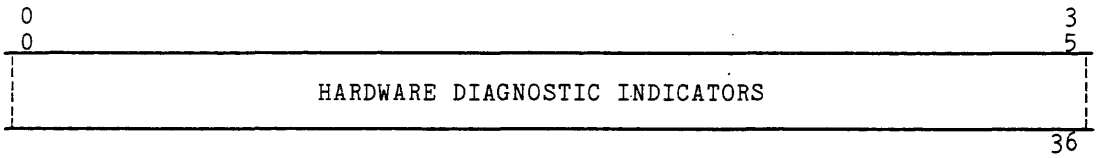


Figure 4-11. Bulk Store DCB Status Block Format, Word 4

BULK STORE PERIPHERAL STATUS

The MAJOR and SUBSTATUS fields in Figure 4-7 are interpreted according to the list below.

MAJOR SUBSTATUS

40 SUBSYSTEM READY.

- 00 subsystem ready.

42 SUBSYSTEM ATTENTION.

- 01 hardware write inhibited.
  A write operation was attempted to a bulk store unit (BSU) that had its WRITE INHIB switch ON.
- 02 no response from BSU.
  The addressed BSU did not respond within the allowable time.
- 04 error detected in BSU.
  The selected BSU detected a parity error at the address specified, or the selected BSU was offline or powered down.
- 20 BSU address not present.
  No BSU is configured for the address specified in the command.

43 DATA ALERT.

- 01 uncorrectable BSU data error.
  One of the following occurred:
  1. One or more multiple-bit errors were detected in the data transferred from the BSU.
  2. The hardware EDAC syndrome indicated the wrong BSU location was addressed during a read operation.
  3. A data parity error was detected when the EDAC function was inactive.
- 02 data parity error.
  A parity error was detected within the bulk store controller (BSC), by the system controller (SC), or on the BSC/SC interface.
- 04 write conditional inhibited.
  A write conditional command was attempted but was not executed because the first BSU word addressed did not contain zeros.

10 hardware detected control error.
One of the following occurred:
1. The BSC detected an internal parity error on an address or tally used for data transfer control.
2. The SC did not respond to a BSC request within the allowable time.
3. The SC reported an illegal action (IA) which was not data parity, out-of-bounds, or nonexistent memory.

20 write verification failed.
A write and verify command was attempted and the data read from the BSU contained uncorrectable errors.

40 failed to compare.
A compare command was executed and the data did not compare.

44 END OF FILE.

00 The tally in DCB, word 2, (see Figure 4-5 above) exhausted before tally exhaust in a given DCW string. Multics does not use this DCW feature and this status should never be seen.

45 DCB COMMAND REJECT.

01 invalid command.
The BSC is unable to recognize the command code in the DCB.

02 DCB parity error.
A parity error occurred during the reading of the DCB from main store.

04 invalid BSU address.
The bulk store unit (BSU) address for data transfer was not 0 modulo 4.

10 hardware detected control error.
One of the following occurred:
1. An error was detected by the BSC while reading the third and fourth words of the DCB.
2. An IA other than data parity was returned by the SC during the control sequence.

55 DCW REJECT.

01 invalid DCW.

02 hardware detected data error.

04 DCW out of bounds.

10 hardware detected control error.

SECTION V

PERIPHERALS

This section gives a brief summary of the peripheral devices supported by Multics, the commands for these devices, and the status they return. If more detail is required for a particular status, consult Appendix C. Peripheral status is shown in two ways. The major and substatus is given. In addition, the status is shown in octal as it appears in the four MSD of an IOM status word.

PERIPHERALS SUPPORTED BY MULTICS

The various peripherals supported on the Multics system are listed below.

Card Readers	CRZ201 CRU1050
Card Punches	CPZ201 PCU0120
Line Printers	PRT201/202 PRT300/301 PRT303 PRU1200/1600
Disk Storage Subsystems	DSS181 DSS190/191 NDM400
Magnetic Tape Subsystems	MTS400 MTS500
System Consoles	CO8030 CSU6001 (EMC655) CSU6002 (SCC655)

DISK STORAGE CHARACTERISTICS

	MSU0454	MSU0400	DSU190	DSU181
sectors per track	40	40	31	18
tracks per cylinder	19	19	19	20
sectors per cylinder	760	760	589	360
cylinders per device	814	410	410	202
sectors per device	618640	311600	241490	72720
Multics records per cylinder	47	47	36	22
Unused sectors per cylinder	8	8	13	8
Multics records per device	38258	19270	14760	4444
Avg. seek time	25ms	30ms	30ms	34ms
Avg. Rotational latency	8.3ms	8.3ms	8.3ms	12.5ms
Transfer time for Multics records	6.7ms	6.7ms	8.6ms	22.5ms

CARD READER

Command	Octal Code
Request Status	00
Reset Status	40
Read Card Binary	01
Read Card Alphanumeric	02
Read Card Mixed	03
Read Card ASCII (not in CPL)	04
Read Card ASCII Mixed (not in CPL)	05
Read Card EBCDIC (not in CPL)	06
Read Card Mixed ASCII (not in CPL)	07
Reserve Device (not in CPL)	66
Release Device (not in CPL)	67
Set Native Mode (not in CPL)	65

Status	Major	Substatus	Octal
Channel Ready	0000		
51-Column Cards		000001	4001
Attention	0010		
Off-Line		000000	4200
Hopper/Stacker		xxx0x1	4201
Manual Halt		xxx01x	4202
Last Batch		xxx1x1	4205
Feed Alert		0x10xx	4210
Card Jam		x1x0xx	4220
Read Alert		1x00xx	4240
Sneak Feed		1x10xx	4250
Data Alert	0011		
Transfer Timing Alert		000001	4301
Validity Alert		000x10	4302
Dual Read Alert		0001x0	4304
No Read Instruction		001000	4310
Command Reject	0101		
Invalid Op Code		0000x1	4501
Invalid Device Code		00001x	4502
Parity, IDCW/LC#		000100	4504
MPC Attention	1010		
IAI Error		000001	5201
DAI Error		000010	5202
DA Transfer Error		000100	5204
Invalid Punch		001000	5210
MPC Data Alert	1011		
Transmission Parity		000001	5301
DAI Error		000101	5305
MPC Command Reject	1101		
Illegal Procedure		000001	5501
Illegal LC#		000010	5502
Device Reserved		001000	5510

CARD PUNCH

Command	Octal Code
Request Status	00
Reset Status	40
Punch Card Binary	11
Punch Card Alphanumeric	12
Punch Card Edited Alphanumeric	13
Punch Card ASCII (not in CPI version)	14
Punch Card EBCDIC (not in CPI version)	15
Reserve Device (not in CPI version)	66
Release Device (not in CPI version)	67

Status	Major	Substatus	Octal
Channel Ready	0000		
Ready		000000	4000
Attention	0010		
Off-Line		000000	4200
Hopper/Stacker		0xxxx1	4201
Manual Halt		0xxx1x	4202
Chad Box Full		0xx1xx	4204
Feed Failure		0x1xxx	4210
Card Jam		01xxxx	4220
Data Alert	0011		
Transfer Timing Alert		000xx1	4301
Transmission Parity Alert		000x1x	4302
Punch Alert		0001xx	4304
Command Reject	0101		
Invalid Op Code		000001	4501
Invalid Device Code		000010	4502
Parity Error, IDCW/LC#		000100	4504
MPC Attention	1010		
IAI Error		000001	5201
DAI Error		000010	5202
DA Transfer Error		000100	5204
MPC Data Alert	1011		
Transmission Parity		000001	5301
DAI Error		000101	5305
PSI Data Overflow		000110	5306
MPC Command Reject	1101		
Illegal LC#		000010	5502
Illegal Procedure		000001	5501
Device Reserved		001000	5510

PRINTERS

Command (Models PRT203/303, PRU1200/1600)	Octal Code
Request Status	00
Reset Status	40
Print Nonedited BCD, Slew Zero Lines	10
Print Nonedited BCD, Slew One Line	11
Print Nonedited BCD, Slew Two Lines	12
Print Nonedited BCD, Slew Top of Page	13
Print Edited BCD, Slew Zero Lines	30
Print Edited BCD, Slew One Line	31
Print Edited BCD, Slew Two Lines	32
Print Edited BCD, Slew Top of Page	33
Print Nonedited ASCII, Slew Zero Lines	14
Print Nonedited ASCII, Slew One Line	15
Print Nonedited ASCII, Slew Two Lines	16
Print Nonedited ASCII, Slew Top of Page	17
Print Edited ASCII, Slew Zero Lines	34
Print Edited ASCII, Slew One Line	35
Print Edited ASCII, Slew Two Lines	36
Print Edited ASCII, Slew Top of Page	37
Slew One Line	61
Slew Two Lines	62
Slew Top of Page	63
Load Image Buffer (ASCII mode only)	01
Read Status	03
Reserve Device	66
Release Device	67
Load VFC Image	05

Command (Model PRT202/300)	Octal Code
Print in Edited Mode (data controls slewing)	30
Print in Edited Mode--slew single line	31
Print in Edited Mode--slew double line	32
Print in Edited Mode--slew to Top of Page	33
Print in Nonedited mode--slew no lines	10
Print in Nonedited mode--slew single line	11
Print in Nonedited mode--slew double line	12
Print in Nonedited mode--slew to Top of Page	13
Slew single line--no print	61
Slew double line--no print	62
Slew to Top of Page--no print	63
Load Image Buffer--no slew	14
Reset Status	40
Request Status	00

Status (Models PRT303, PRU1200/1600)	Major	Substatus	Octal
Ready	0000		
Normal		000000	4000
Print One Line		000001	4001
Forward Space		000010	4002
Forward To Top		000011	4003
Invalid Line		000100	4004
Reverse Rewind		000101	4005
Backspace		000110	4006
Backspace Top		000111	4007
Attention	0010		
Power Fault		000000	4200
Out of Paper		000001	4201
Manual Halt		000010	4202
VFC Image Error/Tape Alert		000100	4204
Check Alert		001000	4210
Data Alert	0011		
Image Buffer Alert/Invalid Character Code		000000	4300
Transfer Timing Alert		0000x1	4301
Alert Before Print		00001x	4302
Alert After Start of Print		000100	4304
Paper Low		001000	4310
Paper Motion Alert/Slew Error		010000	4320
Top of Page Echo		1000x0	4340
Command Reject	0101		
No VFC		000000	4500
Invalid Command Code		000xx1	4501
Invalid Device Code		000x1x	4502
Parity error on command or device code		0001xx	4504
No Belt Image		001000	4510
Slew Error on Last Operation		010000	4520
Top of Page Echo on Last Slew		100000	4540
MPC Attention	1010		
IAI Error		000001	5201
DAI Error		000010	5202
MPC Data Alert	1011		
Transmission Parity		000001	5301
Sum Check Error		000011	5303
DAI Error		000101	5305
PSI Data Overflow		000110	5306
MPC Command Reject	1101		
Illegal Procedure		000001	5501
Illegal Logical Channel No.		000010	5502
Device Reserved		001000	5510

OPERATOR'S CONSOLE

Command	Octal Code
Read	03
Write	13
Write Alert	51
Reset Status	40
Request Status	00
Write ASCII	33
Read ASCII	23
T&D Read	07

Status	Major	Substatus	Octal
Channel Ready	0000		
No Substatus		000000	4000
Device Attention	0010		
No Substatus		000000	4200
Data Alert	0011		
Transfer Timing Error		000001	4301
Transmission Parity Error		0x0010	4302
Operator Input Error		000100	4304
Operator Distracted		001000	4310
Incorrect Format		0100x0	4320
Message Length Alert		100000	4340
Command Reject	0101		
Invalid Instruction Code		000001	4501

TAPE

Command	Octal Code
Request Status	00
Reset Status	40
Request Device Status	50
Reset Device Status	51
Survey Devices	57
Read Control Registers	26
Write Control Registers	16
Set File Protect	62
Set File Permit	63
Rewind	70
Tape Load	75
Rewind/Unload	72
Reserve Device	66
Release Device	67
Set 200 BPI	64
Set 556 BPI	61 or 43
Set 800 BPI	60 or 42
Set 1600 CPI	65
Forward Space One Record	44
Forward Space One File	45
Backspace One Record	46
Backspace One File	47
Control Store Overlay	10
Load From Device	05
Erase	54
Write End-of-File Record	55
Write Tape Nine	13
Read Tape Nine	03
Write Binary Record	15
Read Binary Record	05
Reread Binary Record	07
Write BCD Record	14
Read BCD Record	04
Reread BCD Record	06
Write EBCDIC Record	34
Read EBCDIC Record	24
Write ASCII Record	37
Read ASCII Record	27
Write ASCII/EBCDIC Record	35
Read ASCII/EBCDIC Record	25
Diagnostic Mode Control	31
Main Memory Overlay	11

Status	Major	Substatus	Octal
Peripheral Subsystem Ready	0000		
Ready		000000	4000
Write Protected		xx0xx1	4001
Positioned at BOT		000x1x	4002
9-Track Handler		xxx1xx	4004
2-Bit Fill		010x0x	4020
4-Bit Fill		100x0x	4040
6-Bit Fill		110x0x	4060
ASCII Alert		001100	4014

Device Busy	0001		
In Rewind		000001	4101
Device Reserved		100000	4140
Alternate Channel in Control		000010	4102
Device Loading		000100	4104
Device Attention	0010		
Write Protected		00xx01	4201
No Such Handler		000010	4204
Handler in Standby		0xx10x	4204
Handler Check		0x1x0x	4210
Blank Tape on Write		01xx00	4220
Device Data Alert	0011		
Transfer Timing Alert		000001	4301
Blank Tape on Read		000010	4302
Bit Detected During Erase Operation		xxxx11	4303
Transmission Parity Alert		xxx1xx	4304
Lateral Tape Parity Alert		xx1xxx	4310
Longitudinal Tape Parity Alert		x1xxxx	4320
End of Tape Mark		1xxxxx	4340
End of File	0100		
End of File Mark (7-Track)		001111	4417
End of File Mark (9-Track)		010011	4423
Data Alert Condition		111111	4477
Single Character Record		xxxxxx	44xx
Command Reject	0101		
Invalid density		000000	4500
Invalid Op Code		000xx1	4501
Invalid Device Code		000x1x	4502
Invalid IDCW Parity		0001xx	4504
Positioned at Bot		001000	4510
Forward Read After Write		010000	4520
9-Track Error		100000	4540
MPC Device Attention	1010		
Configuration Switch Error		000001	5201
Multiple Devices		000010	5202
Illegal Device ID Number		000011	5203
Incompatible Mode		001000	5210
TCA Malfunction		0011xx	5214
MTH Malfunction		010000	5220
Multiple BOT		010001	5221
MPC Device Data Alert	1011		
Transmission Alert		000001	5301
Inconsistent Command		000010	5302
Sum Check Error		000011	5303
Byte Locked Out		000100	5304
PE-Burst Write Error		001000	5310
Preamble Error		001001	5311
T&D Error		001010	5312
Multitrack Error		010000	5320
Skew Error		010001	5321
Postamble Error		010010	5322
NRZI CCC Error		010011	5323
Code Alert		010100	5324
Marginal Condition		100000	5340
MPC Command Reject	1101		
Illegal Procedure		000001	5501
Illegal LC Number		000010	5502
Illegal Suspended LC Number		000011	5503
Continue Bit Not Set		000100	5504

DISKS

Command	Octal Code
Seek	34
Special Seek (T&D)	36
Preseek	37
Restore	42
Read	25
Read ASCII	23
Write	31
Write ASCII	32
Write and Compare	33
Read Nonstandard Size	04
Read Track Header	27
Format Track	17
Request Status	00
Reset Status	40
Read Control Register	26
Write Control Register	16
Read Status Register	22
Read EDAC Register	21
Release	76
Reserve Device	77
Set Standby	72
Bootload CS	10
ITR Boot	11
Execute Device Command (DLI)	30

Status	Major	Substatus	Octal
Channel Ready	0000		
No Substatus		000000	4000
Retries (xx = Retry count)		0000xx	400x
Device in T&D		0010xx	4010
Busy	0001		
Positioning		000000	4100
Alternate Channel		100000	4140
Attention	0010		
Write Inhibit		000001	4201
Seek Incomplete		000010	4202
Device Inoperable		001000	4210
Device in Standby		010000	4220
Device Off-Line		100000	4240
Data Alert	0011		
Transfer Timing		000001	4301
Transmission Parity		000010	4302
Invalid Seek Address		000100	4304
Header Verification		0x1000	4310
Cyclic Check		x1x000	4320
Compare Alert		1x0000	4340
End-of-File	0100		
Good Track		000000	4400
Last Consecutive Block		0000x1	4401
Block Count Limit		00001x	4402
Defective Track-Alt. Assg.		000100	4404
Defective Track-No Alt.		001000	4410
Alt. Track Det.		010000	4420

Command Reject	0101		
Invalid Op Code		000001	4501
Invalid Device Code		000010	4502
IDCW Parity		000100	4504
Invalid Inst. Sequence		001000	4510
Channel Busy	1000		
No substatus		xxxxxx	5000
MPC Device Attention	1010		
Configuration Error		000001	5201
Multiple Device		000010	5202
Device No. Error		000011	5203
CA Error or OPI Down		001011	5213
Alert EN1		001100	5214
CA EN1 Error		001101	5215
CA Alert (no EN1)		001110	5216
MPC Device Data Alert	1011		
Transmission Parity		000001	5301
Inconsistent Command		000010	5302
Sum Check Error		000011	5303
Byte Lockout		000100	5304
EDAC Parity		001110	5316
Sector Size Error		010001	5321
Nonstandard Sector Size		010010	5322
Search Alert (1st)		010011	5323
Cyclic Code (≠ 1st)		010100	5324
Search Alert (≠ 1st)		010101	5325
Sync Byte ≠ Hex 19		010110	5326
Error in Alt. Track Processing		010111	5327
EDAC Corr. - Last Sect.		011001	5331
EDAC Corr. ≠ Last Sect.		011010	5332
EDAC Corr. Block Count Limit		011011	5333
EDAC Uncorrectable		011100	5334
EDAC Corr. Short Block		011101	5335
MPC Command Reject	1101		
Illegal Procedure		000001	5501
Illegal Logical Channel Number		000010	5502
Illegal Suspended		000011	5503
Continue Bit Not Set		000100	5504

SECTION VI

DATANET 6600 FRONT-END NETWORK PROCESSOR

This section gives information on the formats of the status and control words for the DATANET 6600 Front-End Network Processor (FNP).

FNP PROCESSOR DATA

The following paragraphs describe the processor instruction formats and instruction opcodes.

Instruction Word Formats

The FNP instruction word has three formats:

Store Reference Instructions
Nonstore Reference Instructions - Group 1
Nonstore Reference Instructions - Group 2

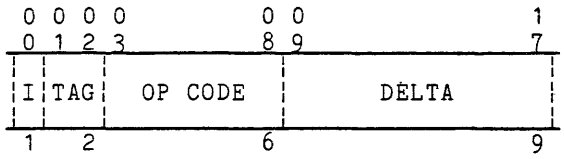


Figure 6-1. FNP Store Reference Instruction Format

Legend:

I	indirect addressing flag.
TAG	index to be used in address preparation.
OP CODE	instruction operation code.
DELTA	offset to be used in address preparation.

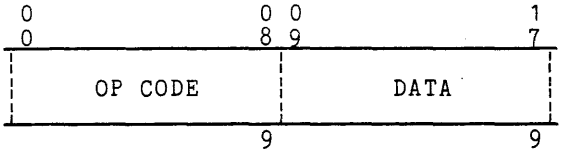


Figure 6-2. FNP Nonstore Reference Instruction Format - Group 1

Legend:

OP CODE instruction operation code.

DATA data for instruction execution.

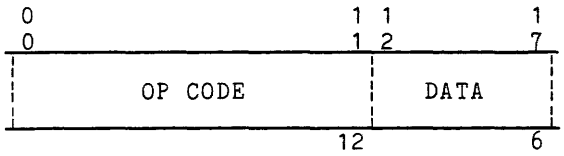


Figure 6-3. FNP Nonstore Reference Instruction Format - Group 2

Legend:

OP CODE instruction operation code.

DATA data for instruction execution.

FNP Operation Code Charts

Table 6-1. Store Reference Instruction Opcodes

	0	1	2	3	4	5	6	7
0		mpf	adcx2	ldx2	ldaq		ada	lda
1	tsy		(grp1)	stx2	staq	adaq	asa	sta
2	szn	dvf	(grp1)	cmpx2	sbaq		sba	cmpa
3	ldex	cana	ansa	(grp2)	ana	era	ssa	ora
4	adcx3	ldx3	adcx1	ldx1	ldi	tnc	adq	ldq
5	stx3		(grp1)	stx1	sti	tov	stz	stq
6	cioc	cmpx3	ersa	cmpx1	tnz	tpl	sbq	cmpq
7	stex	tra	orsa	(grp1)	tze	tmi	aos	

Table 6-2. Nonstore Reference Instruction Opcodes (Group 1)

	x = 0	1	2	3	4	5	6	7
x12	rier				ria			
x22	iana	iora	icana	iera	icmpa			
x52	sier				sic			
x73	sel	iacx1	iacx2	iacx3	ilq	iaq	ila	iaa

Table 6-3. Nonstore Reference Instruction Opcodes (Group 2)

	0	1	2	3	4	5	6	7
033			cax2		lls	lrs	als	ars
133					nrml		nrm	
233		nop	cx1a		llr	lrl	alr	arl
333		inh	cx2a	cx3a			alp	
433		dis	cax1	cax3			qls	qrs
533								
633				caq			qlr	qrl
733		eni		cqa			qlp	

Table 6-4. Alphabetic Listing of FNP Instruction Opcodes

Mnemonic	Code	Meaning
ada	06	Add to A register
adaq	15	Add to AQ register
adcx1	42	Add character address to index1
adcx2	02	Add character address to index2
adcx3	40	Add character address to index3
adq	46	Add to Q register
alp	3336	A register left parity rotate
alr	2336	A register left rotate
als	0336	A register left shift
ana	34	AND to A register
ansa	32	AND to storage from A register
aos	76	Add one to storage
arl	2337	A register right shift logical
ars	0337	A register right shift
asa	16	Add stored to A register
cana	31	Comparative AND with A register
caq	6333	Copy A register into Q register
cax1	4332	Copy A register into index1
cax2	0332	Copy A register into index2
cax3	4333	Copy A register into index3
cioc	60	Connect I/O channel
cmpa	27	Compare with A register
cmpq	67	Compare with Q register
cmpx1	63	Compare with index1
cmpx2	23	Compare with index2
cmpx3	61	Compare with index3
cqa	7333	Copy Q register into A register
cx1a	2332	Copy index1 into A register
cx2a	3332	Copy index2 into A register
cx3a	3333	Copy index3 into A register
dis	4331	Delay until interrupt
dvf	21	Divide fraction
eni	7331	Enable interrupts
era	35	EXCLUSIVE OR to A register
ersa	62	EXCLUSIVE OR to storage from A register
iaa	773	Immediate add to A register
iacx1	173	Immediate add character address to index1
iacx2	273	Immediate add character address to index2
iacx3	373	Immediate add character address to index3
iana	022	Immediate AND to A register

Table 6-4 (cont). Alphabetic Listing of FNP Instruction Opcodes

Mnemonic	Code	Meaning
iaq	573	Immediate add to Q register
icana	222	Immediate comparative AND with A register
icmpa	422	Immediate compare with A register
iera	322	Immediate EXCLUSIVE OR to A register
ila	673	Immediate load A register
ilq	473	Immediate load Q register
inh	3331	Inhibit interrupts
iora	122	Immediate OR to A register
lda	07	Load A register
ldaq	04	Load AQ register
ldex	30	Load external channel
ldi	44	Load indicators
ldq	47	Load Q register
ldx1	43	Load index1
ldx2	03	Load index2
ldx3	41	Load index3
llr	2334	Long left rotate
lls	0334	Long left shift
lrl	2335	Long right shift logical
lrs	0335	Long right shift
mpf	01	Multiply fraction
nop	2331	No operation
nrm	1336	Normalize
nrml	1334	Normalize long
ora	37	OR to A register
orsa	72	OR to storage from A register
qlp	7336	Q register left parity rotate
qlr	6336	Q register left rotate
qls	4336	Q register left shift
qrl	6337	Q register right shift logical
qrs	4337	Q register right shift
ria	412	Read interrupt address
rier	012	Read interrupt enable register
sba	26	Subtract from A register
sbaq	24	Subtract from AQ register
sbq	66	Subtract from Q register
sel	073	Select I/O channel
sic	452	Set interrupt cells
sier	052	Set interrupt enable register
ssa	36	Subtract stored from A register
sta	17	Store A register
staq	14	Store AQ register
stex	70	Store external channel
sti	54	Store indicators
stq	57	Store Q register
stx1	53	Store index1
stx2	13	Store index2
stx3	50	Store index3
stz	56	Store zero
szn	20	Set zero and negative indicators
tmi	75	Transfer on minus
tnc	45	Transfer on no carry
tnz	64	Transfer on not zero
tov	55	Transfer on overflow
tpl	65	Transfer on plus

Table 6-4 (cont). Alphabetic Listing of FNP Instruction Opcodes

Mnemonic	Code	Meaning
tra	71	Transfer unconditionally
tsy	10	Transfer and store IC
tze	74	Transfer on zero

FNP IOM DATA

IOM Hardware Communications Region Layout

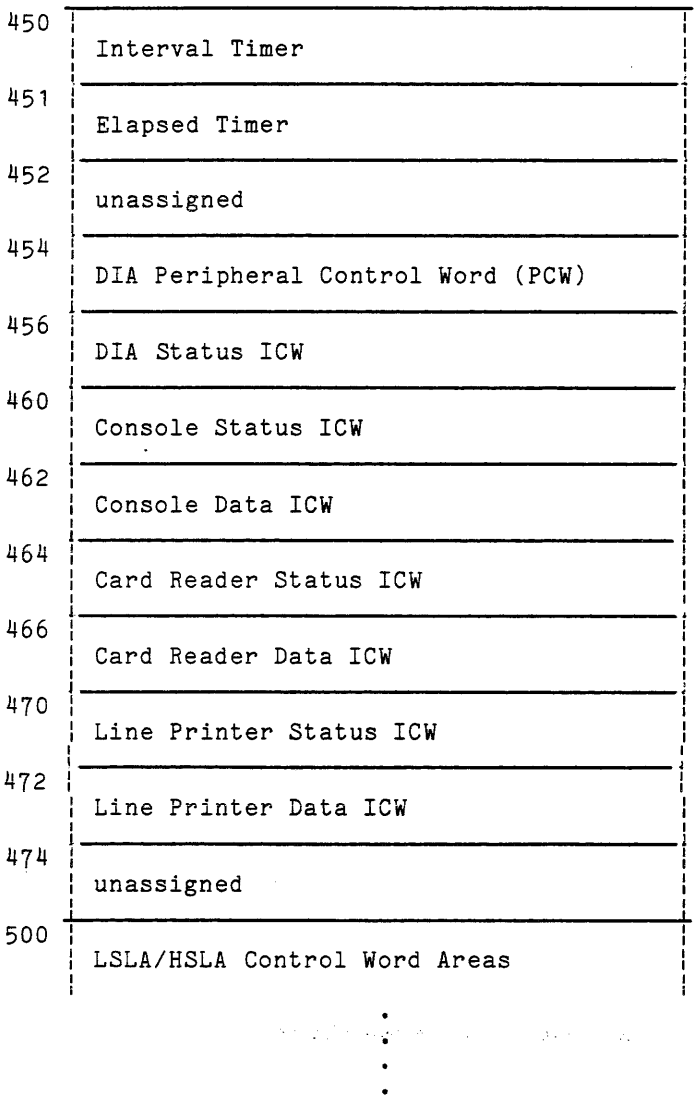
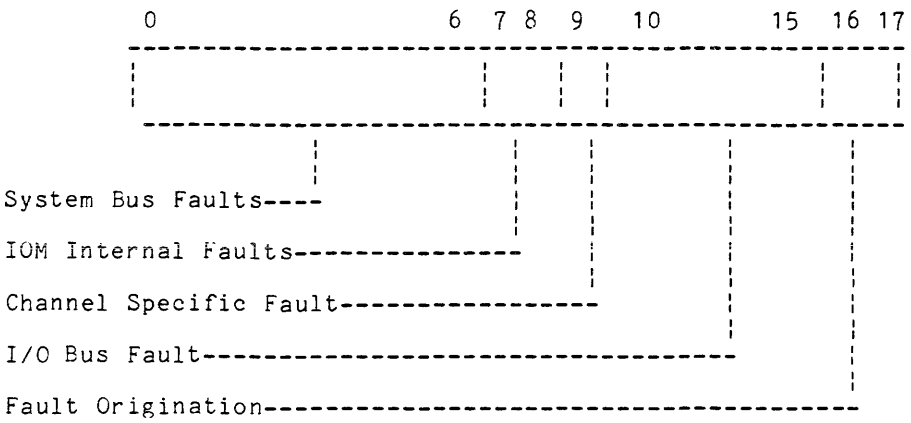


Figure 6-4. FNP IOM Hardware Communications Region Layout

IOM STATUS FORMAT FOR DN6670 FNP



System Bus Faults

mask
400000
when one, the IOM detected an uncorrectable error indication (red) from the HNP main storage unit.

200000
when one, the IOM received an illegal function code from a component on the system bus.

100000
when one, the IOM detected a parity error on data bus lines A and 0 through 7 (left byte).

040000
when one, the IOM detected a parity error on data bus lines B and 8 through 15 (right byte).

020000
when one, the IOM detected a parity error on the address bus lines.

010000
when one, the IOM performed a dead main timeout on the system bus.

004000
when one, the IOM detected a bus logic test error or a bus continuity error on the sytem bus. This condition will never initiate the fault reporting sequence but will only be set as an HNP system status indication.

IOM Internal Faults

mask
002000
when one, the IOM detected a read-only storage (ROS) parity error. Any ROS parity error detected while attempting to report any fault halts the I/O processor.

001000
when one, the IOM page table unit has indicated a fault.

Channel Specific Fault

mask
000400
Not used.

I/O Bus Faults

mask
000200
when one, the IOM received an illegal function code from a channel on the I/O bus.

Bit 11
000100
when one, the IOM detected a parity error on data bus lines A and 0 through 7 (left byte).

Bit 12
000040
when one, the IOM detected a parity error on data bus lines B and 8 through 15 (right byte).

Bit 13
000020
when one, the IOM detected a parity error on the address bus bits (0 through 7) signal lines.

Bit 14
000010
when one, the IOM received an illegal NAK response on the I/O bus.

Bit 15
000004
when one, the I/O bus has failed the bus logic text or the IOM has detected an I/O bus continuity error. This condition will never initiate the fault reporting sequence but will only be set as an HNP system status indication.

Fault Origination

mask
000003
when one, the IOM detected the fault and originated this fault status word.

IOM Fault Status Format

Whenever the FNP IOM detects a channel fault, it stores a fault status word at 420(8) + <channel_number> and interrupts on level 0 for that channel.

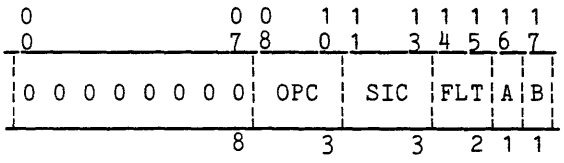


Figure 6-5. FNP IOM Fault Status Word Format

Legend:

OPC channel data operation code.
- 0 no data cycle.
- 1 load.
- 2 store.
- 3 add to store.
- 4 subtract from store.
- 5 AND to store.
- 6 OR to store.
- 7 invalid.

SIC set interrupt cell operation code.
- 0 none.
- 1 unconditional.
- 2 tally = 0 (TY0).
- 3 tally = 1 (TY1).
- 4 negative.
- 5 zero.
- 6 overflow.
- 7 invalid.

FLT fault type code.
- 0 none.
- 1 all other memory illegal actions.
- 2 parity error.
- 3 invalid channel request.

A parity error in IOM channel logic.

B parity error in IOM central logic.

The following combinations of OPC and SIC will cause an invalid channel request fault.

OPC	SIC
7	any
any	7
0	0
0	2
0	3
0	4
0	5
0	6

Indirect Control Word Formats

The indirect control word (ICW) is used consistently throughout the FNP I/O to control the transmission of data to and from channels of the FNP IOM. Individual channels expect particular conditions in their ICWs and will fault if unexpected conditions are found.

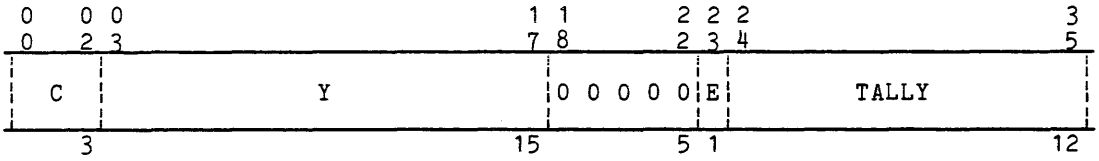


Figure 6-6. FNP IOM ICW Format

Legend:

C — character control.
0 = treat data as 18-bit words.
1 = treat data as 36-bit words.
2 = treat data as 9-bit bytes starting with byte 0 of Y.
3 = treat data as 9-bit bytes starting with byte 1 of Y.
4 = treat data as 6-bit bytes starting with byte 0 of Y.
5 = treat data as 6-bit bytes starting with byte 1 of Y.
6 = treat data as 6-bit bytes starting with byte 2 of Y.
7 = indirect idle, no data transmission.

Y — FNP data address.
Some channels will force the LSB of this address to zero in order to ensure access to word pairs.

E — tally runout.
This bit is set when a tally runout condition is detected. If the bit is intentionally set by the software, tallying and address incrementing are suppressed.

TALLY — count of memory accesses needed for data transfer.

PERIPHERAL STATUS/CONTROL WORD FORMATS

Formats of the status words and control words for peripherals are described in the following paragraphs.

Direct Interface Adapter

Direct Interface Adapter (DIA) Peripheral Control Word (PCW) - (454-455):

The location of the DIA PCW (454) is normally specified as the effective Y-address of the CIOC instruction, i.e., the CIOC operand word and the PCW are the same word.

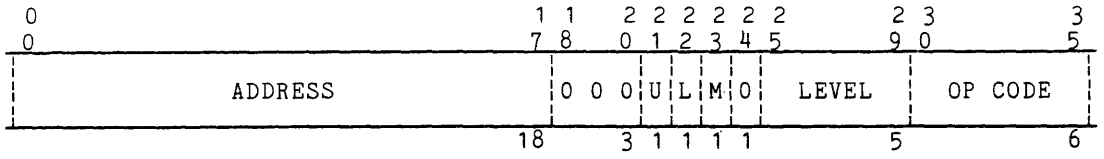


Figure 6-7. FNP DIA PCW Format

Legend:

ADDRESS address of a "list ICW" in FNP pointing to a list of "command DCWs".

U parity bit for 0-17 giving odd parity for the even FNP word.

L parity bit for 18-35 giving odd parity for the odd FNP word.

M channel mask bit.

LEVEL interrupt level to be sent to the Multics IOM.

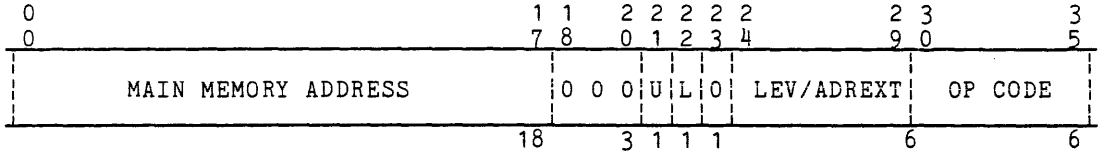
OP CODE DIA operation code.
73 signal an interrupt at LEVEL to the Multics IOM.

$\overline{73}$ with DIA not busy -- fetch command DCWs using list ICW at ADDRESS.
with DIA busy -- invalid connect.

DIA Command Data Control Words (DCWs):

The DIA data control words are located at the address specified in the list ICW. The list ICW is located at the address given in the address field of the PCW (see Figure 6-7 above). The list ICW used to access the DIA DCWs must specify 36-bit addressing.

First Word Pair



Second Word Pair

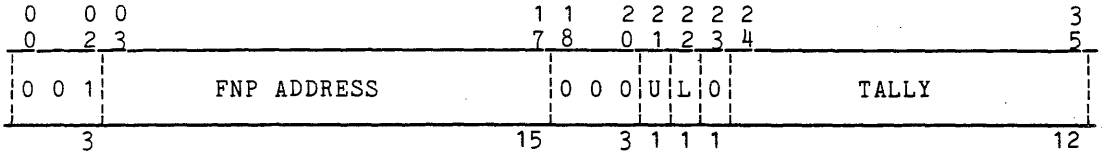


Figure 6-8. FNP DIA DCW Format

Legend:

MAIN MEMORY ADDRESS
18 low-order bits of 24-bit Multics absolute main memory address for data.

U parity bit for 0-17 giving odd parity for the even FNP word.

L parity bit for 18-35 giving odd parity for the odd FNP word.

LEV/ADREXT
interrupt level for Multics IOM interrupt or six high-order bits of Multics absolute main memory address for data.

OP CODE DIA operation code.
65 = read and clear 6180; OR to storage FNP.
70 = disconnect and interrupt FNP.
71 = interrupt FNP.
72 = jump (similar to IOM TDCW).
73 = interrupt Multics at LEV.
74 = report configuration status.
75 = data transfer; FNP to Multics.
76 = data transfer; Multics to FNP.

FNP ADDRESS
FNP store address for data.

TALLY count of memory accesses needed for data transfer.

DIA Status Word:

The DIA status word location is controlled by the DIA status ICW at (456-457).

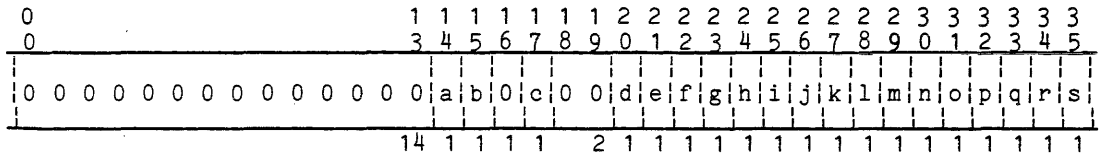


Figure 6-9. FNP DIA Status Word Format

Legend:

Mask	Key	Meaning
000010	a	DIA internal parity error.
000004	b	FNP software parity error.
000001	c	Multics IOM/DIA ready.
100000	d	invalid connect from FNP.
040000	e	invalid command from FNP.
020000	f	list ICW tally runout.
010000	g	data DCW not direct-36.
004000	h	Multics main memory address less than lower bound.
002000	i	Multics main memory address greater than upper bound.
001000	j	while inhibited by the restricted cycle switches, an attempt was made to perform a read and clear on main memory and OR to FNP storage; a read interrupt cells; or a data transfer command (FNP to Multics).

Mask	Key	Meaning
000400	k	test command received while busy.
000200	l	invalid command from Multics.
000100	m	no answer from Multics.
000040	n	List ICW accessed with bit 23 on.
000020	o	Multics IOM parity error.
000010	p	command error in Multics IOM.
000004	q	U-bus error in Multics IOM.
000002	r	data parity error in Multics IOM.
000001	s	Multics IOM system fault.

Console

Console Peripheral Control Word (PCW):

This word is located at the effective Y-address specified by the CIOC instruction.

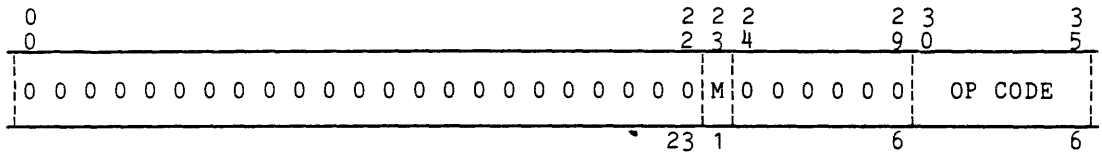


Figure 6-10. FNP Console PCW Format

Legend:

M	channel mask bit.
OP CODE	channel operation code. 00 = request status. 44 = write. 50 = read. 54 = wraparound mode.

Console Data Format

Data is transmitted as 9-bit characters under control of the data ICW at (462-463).

Console Status Word Format

Status is stored as a 9-bit byte under control of the status ICW at (460-461).

bit	name
0	device ready.
1	timer runout.
2	tally runout.
3	pre-tally runout.
4	transfer timing error.
5	control character.
6	connect while busy.
7	invalid PCW.
8	parity on read.

Card Reader

Card Reader Peripheral Control Word (PCW):

This word is located at the effective Y-address specified by the CIOC instruction.

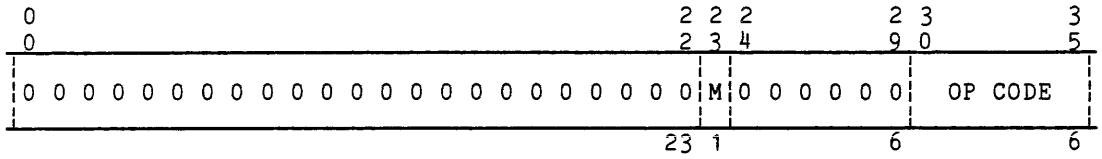


Figure 6-11. FNP Card Reader PCW Format

Legend:

M channel mask bit.

OP CODE channel operation code.
00 = request status.
01 = read card binary.
02 = read card decimal.
03 = read card mixed.
40 = reset status.

Card Reader Data Format

Data is read as 6-bit characters under control of the data ICW at (466-467).

Card Reader Status Word Format

Status is stored as a 36-bit peripheral status word under control of the status ICW at (464-465). See "Channel Status" in Section III for format and "Card Readers" in Section V or Appendix C for a description of the appropriate fields.

NOTE: The FNP does not support the CRU1050. The older CR10 and CR20 sometimes used for the FNP store the same status as the CRZ201.

Line Printer

Line Printer Peripheral Control Word (PCW):

This word is located at the effective Y-address specified by the CIOC instruction.

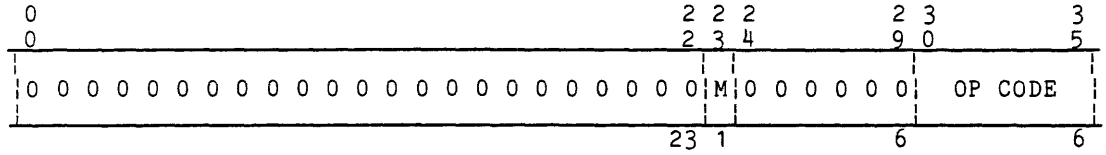


Figure 6-12. FNP Line Printer PCW Format

Legend:

M — channel mask bit.

OP CODE — channel operation code.
- 00 = request status.
- 10 = write nonedited, no slew.
- 11 = write nonedited, slew one line.
- 12 = write nonedited, slew two lines.
- 13 = write nonedited, slew to top.
- 30 = write edited, no slew.
- 31 = write edited, slew one line.
- 32 = write edited, slew two lines.
- 33 = write edited, slew to top.
- 40 = reset status.
- 61 = slew one line.
- 62 = slew two lines.
- 63 = slew to top.

Line Printer Data Format

Data is written as 6-bit characters under control of the data ICW at (472-473).

Line Printer Status Word Format

Status is stored as a 36-bit peripheral status word under control of the status ICW at (470-471). See "Channel Status" in Section III for format and "Line Printers" in Section V or Appendix C for a description of the appropriate fields.

NOTE: The FNP does not support the PRU1200 or PRU1600.

Low-Speed Line Adapter

Low-Speed Line Adapter (LSLA) Peripheral Control Words (PCWs):

This word is located at the effective Y-address specified by the CIOC instruction.

PCW0

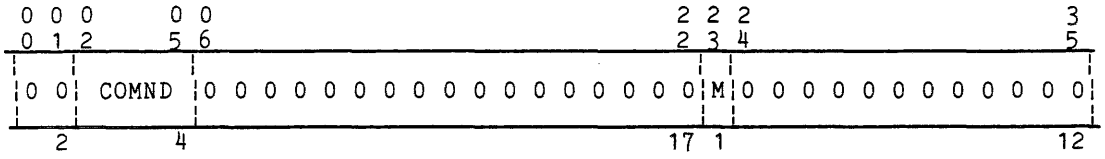


Figure 6-13. FNP LSLA PCW0 Format

Legend:

COMND channel command. (See "LSLA PCW Commands" below.)

M channel mask bit.

PCW1

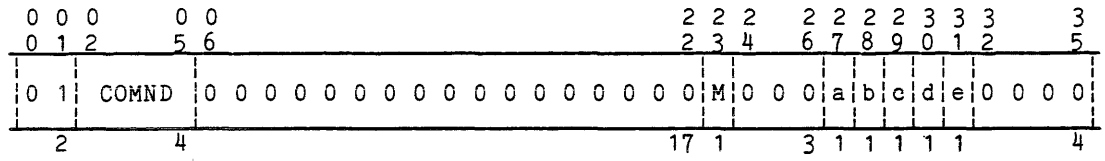


Figure 6-14. FNP LSLA PCW1 Format

Legend:

Mask	Key	Field	Meaning
170000		COMND	channel command. (See "LSLA PCW Commands" below.)
010000		M	channel mask bit.
000400	a		set receive mode.

Mask	Key	Field	Meaning
000200	b		set send mode.
000100	c		set wraparound mode.
000040	d		set data terminal ready.
000020	e		set request to send.

LSLA PCW Commands

COMND	Command
00	no command (needed for broadside channel commands).
01	input status request.
02	output status request.
03	configuration status request.
06	switch receive ICW.
07	switch send ICW.
10	initialize.
14	resynchronize.

LSLA Control Word Area

Each LSLA has a dedicated 16-word control area. See "FNP Store Map" later in this section for area locations.

relative area addr	function
0-1	primary receive ICW
2-3	secondary receive ICW
4-5	primary send ICW
6-7	secondary send ICW
10-11	not used
12-13	not used
14-15	active status ICW
16-17	configuration status mailbox

NOTE: All data ICWs specify 9-bit characters.

LSLA Active Status Word Format

Active status is stored as one 36-bit word under control of the status ICW at <control_word_area>|14.

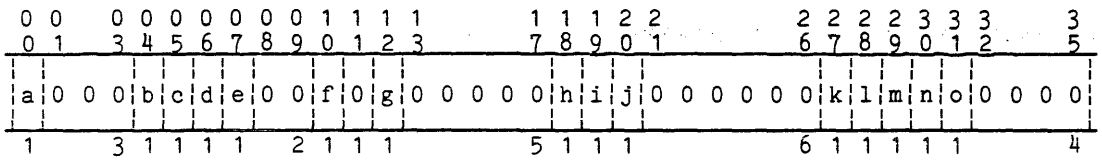


Figure 6-15. FNP LSLA Active Status Word Format

Legend:

Mask	Key	Meaning
400000	a	status type. 0 = send status. 1 = receive status.
020000	b	active buffer. 0 = primary buffer (ICW) in use. 1 = secondary buffer (ICW) in use.
010000	c	1 = buffers (ICWs) switched after status store.
004000	d	1 = TY0 tally condition.
002000	e	1 = TY1 tally condition.
000200	f	data set status change (receive only). If data set ready changes state or if a data terminal ready PCW is sent and either clear to send or carrier detect (i or j below) changes state, an active status interrupt occurs and receive status is stored with this bit set.
000040	g	transfer timing error.
400000	h	data set ready.
200000	i	clear to send.
100000	j	carrier detect.
000400	k	receive mode.
000200	l	send mode.
000100	m	wraparound mode.
000040	n	data terminal ready.
000020	o	request to send.

LSLA Configuration Status Word Format

Configuration status is stored as one 36-bit word into the configuration status mailbox at <control_word_area>|16.

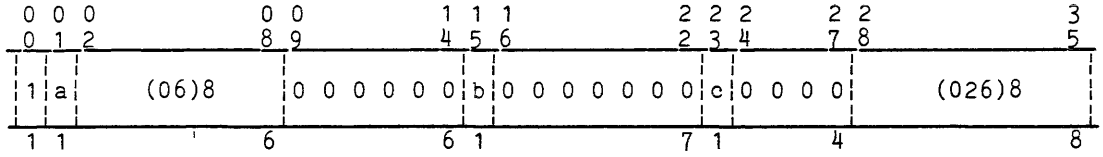


Figure 6-16. FNP LSLA Configuration Status Word

Legend:

Key	Field	Meaning
a		1 = synchronous.
	(06)8	subchannel type (always 06 for LSLA).
b		1 = two send ICWs being used.
c		1 = 8-bit characters.
	(026)8	line synchronizing character.

LSLA Device Command Characters

The LSLA is able to send device commands to the modems on its subchannels and to exercise the T&D subchannel by means of command character sequences transmitted as data. Device status returned from the modems and the T&D subchannel are recognized by a similar character sequence. The character sequence consists of an ESC character with odd parity (233)8, followed by any number (including zero) of fill characters (037)8, followed by one of the command/status characters below.

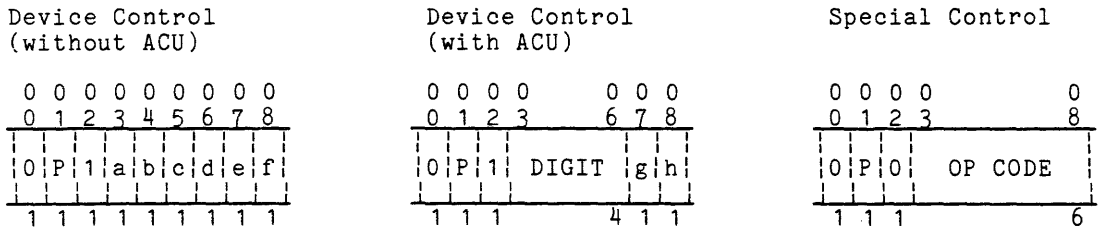


Figure 6-17. FNP LSLA Device Control Characters

Legend:

Mask	Key	Field	Meaning
200		P	parity bit giving odd parity to the character.
040	a		frequency select.
020	b		answer control for Bell 103E modem.
010	c		busy.
004	d		data terminal ready.
002	e		request to send.
001	f		line break transmit.
074		DIGIT	binary value of next digit to be dialed.
002	g		call request.
001	h		digit present.
077		OP CODE	special channel command. 40 = error count command. 41 = unused. 44 = low-speed wraparound reset. 45 = low-speed wraparound set. 50 = high-speed wraparound. 51 = configuration mode command. 54 = disable protect. 55 = channel status request.

LSLA Device Status Characters

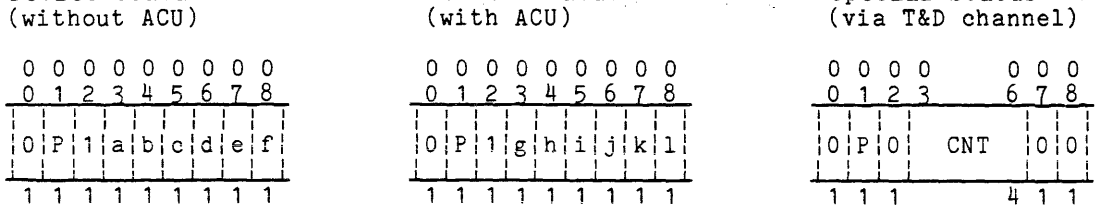


Figure 6-18. FNP LSLA Device Status Character Formats

Legend:

Mask	Key	Field	Meaning
		P	parity bit giving odd parity to the character.
040	a		data set ready.
020	b		restraint.
010	c		clear to send.
004	d		ring.
002	e		carrier detect.
001	f		line break.
040	g		power indicator.
020	h		data set status.
010	i		present (send) next digit.
004	j		data line occupied.
002	k		abandon call and retry.
001	l		used only by T&D.
		CNT	binary error count.

High-Speed Line Adapter

High-Speed Line Adapter (HSLA) Peripheral Control Word (PCW) Formats:

This word is located at the effective Y-address specified by the CIOC instruction.

PCW0

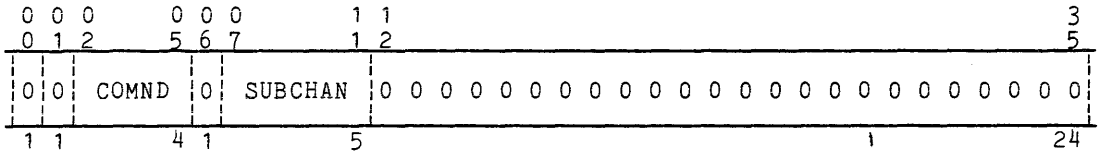


Figure 6-19. FNP HSLA PCW0 Format

Legend:

COMND subchannel command. (See "HSLA PCW Commands" below.)

SUBCHAN subchannel number.

PCW1

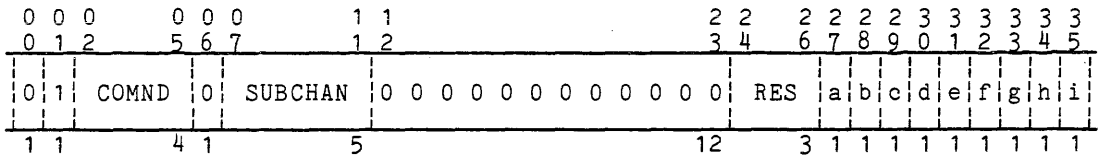


Figure 6-20. FNP HSLA PCW1 Format

Legend:

Mask	Key	Field	Meaning
170000		COMND	subchannel command. (See "HSLA PCW Commands" below.)
003700		SUBCHAN	subchannel number.
007000		RES	unassigned, reserved for broadside commands.
000400	a		set receive mode.
000200	b		set send mode.
000100	c		set wraparound mode.
000040	d		set data terminal ready.
000004	g		supervisory send.
000002	h		ACU call request.
000001	i		spare.

PCW2

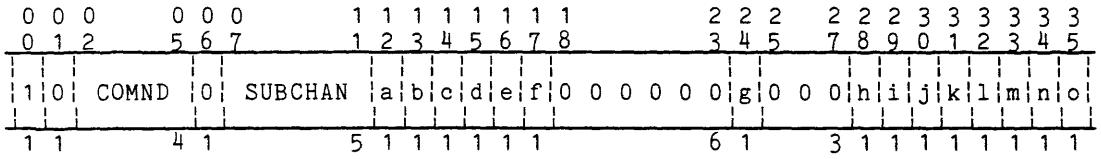


Figure 6-21. FNP HSLA PCW2 Format

Legend:

Mask	Key	Field	Meaning
170000		COMND	subchannel command. (See "HSLA PCW Commands" below.)
037000		SUBCHAN	subchannel number.
000040	a		receive data has parity.
000020	b		send data has parity.
000010	c		use odd parity.
000004	d		use two send ICWs.
000002	e		enable character control table (CCT).
000001	f		spare.
004000	g		0 = one stop bit. 1 = two stop bits.
000200	h		set 110 baud.
000100	i		set 134.5 baud.
000040	j		set 150 baud.
000020	k		set 300 baud.
000010	l		set 1050 baud.
000004	m		set 1200 baud.
000002	n		set 1800 baud.
000001	o		set optional baud rate (e.g., 75 or 600).

PCW3

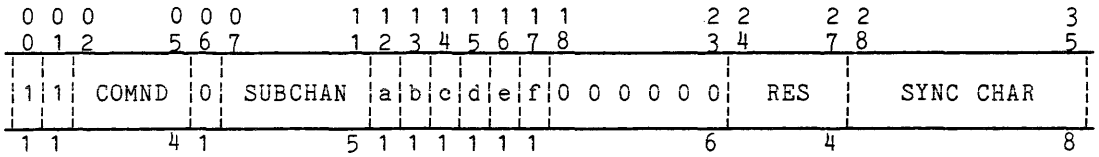


Figure 6-22. FNP HSLA PCW3 Format

Legend:

Mask	Key	Field	Meaning
170000		COMND	subchannel command. (See "HSLA PCW Commands" below.)
003700		SUBCHAN	subchannel number.
000040	a		receive data has parity.
000020	b		send data has parity.
000010	c		use odd parity.
000004	d		use two send ICWs.
000002	e		enable character control table (CCT).
000001	f		spare.
007400		RES	reserved for subchannel use.
000377		SYNC CHAR	subchannel synchronizing character.

HSLA PCW Commands

PCW0,1	COMND	Command
	00	no command (needed for broadside commands).
	01	subchannel input status request.
	02	subchannel output status request.
	03	subchannel configuration status request.
	04	set subchannel mask register bit.
	05	reset subchannel mask register bit.
	06	switch subchannel receive data buffers (ICWs).
	07	switch subchannel send data buffers (ICWs).
	10	initialize (all subchannels).
	11	store mask register (into subchannel 0 control word area).
	12	not used.
	13	not used.
	14	resynchronize subchannel.
	15	transmit line break.
	16	not used.
	17	not used.

PCW2,3	COMND	Command
	00-07	reserved.
	10-13	not used.
	14	set 5-bit character (asynchronous).
	15	set 6-bit character.
	16	set 7-bit character.
	17	set 8-bit character.

HSLA Control Word Areas

Each HSLA subchannel has a dedicated 16-word control word subarea located at 16 * SUBCHAN within the control word area for the HSLA. See "FNP Store Map" later in this section for HSLA control word area locations.

relative area addr	function
0-1	primary receive ICW.
2-3	secondary receive ICW.
4-5	primary send ICW.
6-7	secondary send ICW.
10	base address word.
11	unused.
12-13	mask register (subchannel 0 only).
14-15	active status ICW.
16-17	configuration status mailbox.

Base Address Word Format

The base address word (BAW) is used by the character control feature of the HSLA to prepare addresses for referencing of the character control table (CCT).

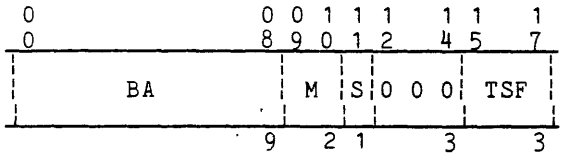


Figure 6-23. FNP HSLA BAW Format

Legend:

BA	base address of character control table (CCT).
M	modifier (used for CCT packing).
S	short table indicator.
TSF	table switch field.

Character Control

The character control feature of the HSLA allows each subchannel to employ its own arbitrary set of control characters. If character control is enabled (see Figures 6-21 and 6-22 above), a reference is made for each data character received to a character control table (CCT) that specifies the action to be taken for that character.

Character control characters are stored as 9-bit characters in the CCT and are selected by the following addressing algorithm:

```
L(CCC) = 00 || B1 || (BAW.BA + BAW.TSF) || (BAW.M | (B7 || B6))
         || (B5 || B4 || B3 || B2)
```

where:

"||" -> concatenation.
"|" -> logical OR.
Bn -> nth bit of data character (B1 = LSB).
"." -> PL/I structure qualifier flag.

BAW.M is used to pack CCTs for short (5-, 6-, 7-bit) codes and BAW.TSF is a dynamic offset, which may be changed by a reference to a character control character (CCC).

Character Control Character Format

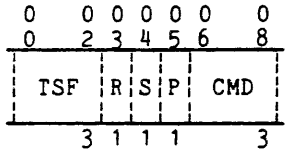


Figure 6-24. FNP HSLA CCC Format

Legend:

TSF table switch field for next CCT reference.

R resynchronize.

S switch buffers.

P inhibit parity.

CMD command field. (All codes, except 6, store character.)
0 = no special action.
1 = terminate after next character.
2 = terminate after second character.
3 = terminate now.
4 = set marker status bit only.
5 = marker interrupt after next character.
6 = do not store character.
7 = marker interrupt now.

Mask Register Word Format for DN355 or DN6632 FNPs

(This word is stored for subchannel 0 only.)

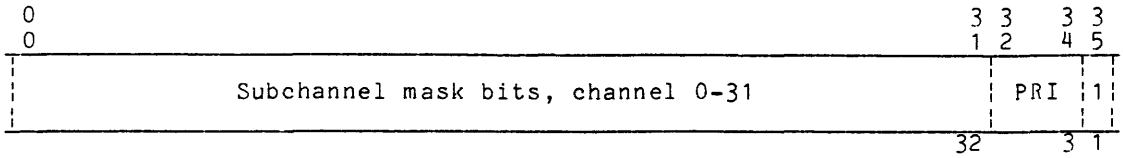


Figure 6-25. FNP HSLA Mask Register Word Format

Legend:

PRI indicates which subchannels will receive priority service from the HSLA central.

PRI	Meaning
000	No high priority scan
001	Subchannels 0 and 1
010	Subchannels 0 through 3
011	Subchannels 0 through 7
100	Subchannels 0 through 15

Mark Register Word for DN6670 FNPs

(This word is stored for subchannels 0, 8, 16 and 24.)

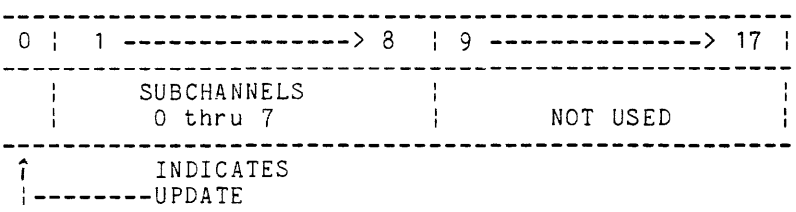


Figure 6-26. DN6670 Mask Status Word

HSLA Active Status Word Format

Active status from subchannels is stored under control of the status ICW at (14-15) in the subchannel control word area.

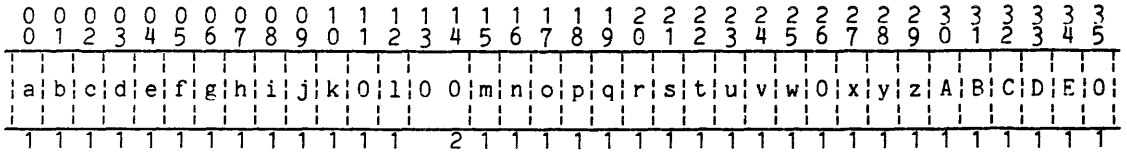


Figure 6-27. FNP HSLA Active Status Word Format

Legend:

Mask	Key	Meaning
400000	a	status type. 0 = send status. 1 = receive status.
200000	b	normal marker character received.
100000	c	delayed marker character received.
040000	d	terminate character received.
020000	e	secondary buffer (ICW) is active.
010000	f	switch buffers (ICWs) after status store.
004000	g	TY0 tally condition.
002000	h	TY1 tally condition.
001000	i	received character parity error.
000400	j	command to unimplemented subchannel.

This page intentionally left blank.

Mask	Key	Meaning
000200	k	change in data set status occurred.
000040	l	transfer timing error.
000004	m	no stop bit received.
000002	n	data line occupied (ACU).
000001	o	power (ACU).
400000	p	data set ready.
200000	q	clear to send.
100000	r	carrier detect.
040000	s	supervisory receive.
020000	t	abandon call and retry (ACU).
010000	u	data set status (ACU).
004000	v	ring indicator.
002000	w	line break.
000400	x	receive mode.
000200	y	send mode.
000100	z	wraparound mode.
000040	A	data terminal ready.
000020	B	request to send.
000010	C	make busy.
000004	D	supervisory send.
000002	E	call request (ACU).

HSLA Configuration Status Word Format

Subchannel configuration status is stored directly into the configuration status mailbox at (16-17) in the subchannel control word area.

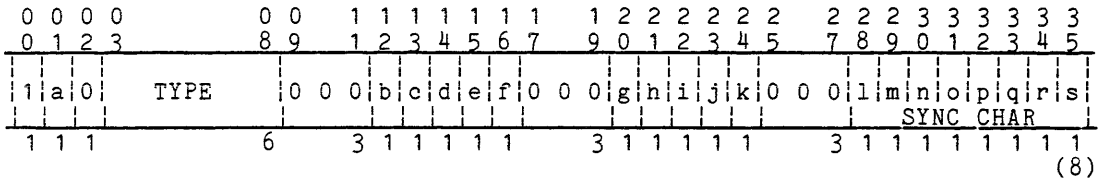


Figure 6-27. FNP HSLA Configuration Status Word Format

Legend:

Mask	Key	Meaning
200000	a	0 = asynchronous subchannel. 1 = synchronous subchannel.
077000	TYPE	subchannel type. 00 = invalid. 01 = general purpose. 02 = general purpose with ACU. 03 = dual synchronous. 04 = dual synchronous with ACU. 05 = dual asynchronous (EIA). 06 = reserved for synchronous line adapter. 07 = dual asynchronous (direct). 10-77 = unassigned.

Mask	Key	Meaning
000040	b	check parity on receive.
000020	c	generate parity on send.
000010	d	0 = use even parity. 1 = use odd parity.
000004	e	use two send ICWs.
000002	f	use BAW (enable character control).
100000	g	5-bit characters if asynchronous.
040000	h	6-bit characters.
020000	i	7-bit characters.
010000	j	8-bit characters.
004000	k	two stop bits.
000200	l	110 baud if asynchronous.
000100	m	134.5 baud if asynchronous.
000040	n	150 baud if asynchronous.
000020	o	300 baud if asynchronous.
000010	p	1050 baud if asynchronous.
000004	q	1200 baud if asynchronous.
000002	r	1800 baud if asynchronous.
000001	s	optional baud rate (e.g., 75 or 600) if asynchronous.
000377	SYNC CHAR	synchronizing character if synchronous.

FNP ENVIRONMENT

The following paragraphs explain the hardware environment in which the FNP code executes.

Interrupt Assignments

The FNP has 256 interrupts organized into 16 levels of 16 interrupts each. Each interrupt within a level corresponds to a bit in the interrupt cell word for that level. When an interrupt occurs, a tsy instruction is forced that makes an indirect reference to location 20(8)*(bit position) + (level).

Table 6-5. FNP Interrupt Assignment Map

	000	001	002	003	004	005	006	007
0	cnsF	cnsS	cnsT	diaS00	H0A00	H0A16	H0C00	H0C16
10	H1A00	H1A16	H1C00	H1C16	H2A00	H2A16	H2C00	H2C16
20	cdrF	cdrS	cdrT	diaS01	H0A01	H0A17	H0C01	H0C17
30	H1A01	H1A17	H1C01	H1C17	H2A01	H2A17	H2C01	H2C17
40	prtF	prtS	prtT	diaS02	H0A02	H0A18	H0C02	H0C18
50	H1A02	H1A18	H1C02	H1C18	H2C02	H2A18	H2C02	H2C18
60				diaS03	H0A03	H0A19	H0C03	H0C19
70	H1A03	H1A19	H1C03	H1C19	H2A03	H2A19	H2C03	H2C19
100	diaF		diaT	diaS04	H0A04	H0A20	H0C04	H0C20
110	H1A04	H1A20	H1C04	H1C20	H2A04	H2A20	H2C04	H2C20
120				diaS05	H0A05	H0A21	H0C05	H0C21
130	H1A05	H1A21	H1C05	H1C21	H2A05	H2A21	H2C05	H2C21
140	H0F			diaS06	H0A06	H0A22	H0C06	H0C22
150	H1A06	H1A22	H1C06	H1C22	H2A06	H2A22	H2C06	H2C22
160	H1F			diaS07	H0A07	H0A23	H0C07	H0C23
170	H1A07	H1A23	H1C07	H1C23	H2A07	H2A23	H2C07	H2C23
200	H2F			diaS08	H0A08	H0A24	H0C08	H0C24
210	H1A08	H1A24	H1C08	H1C24	H2A08	H2A24	H2C08	H2C24
220	L0F	L0A	L0C	diaS09	H0A09	H0A25	H0C09	H0C25
230	H1A09	H1A25	H1C09	H1C25	H2A09	H2A25	H2C09	H2C25
240	L1F	L1A	L1C	diaS10	H0A10	H0A26	H0C10	H0C26
250	H1A10	H1A26	H1C10	H1C26	H2A10	H2A26	H2C10	H2C26
260	L2F	L2A	L2C	diaS11	H0A11	H0A27	H0C11	H0C27
270	H1A11	H1A27	H1C11	H1C27	H2A11	H2A27	H2C11	H2C27
300	L3F	L3A	L3C	diaS12	H0A12	H0A28	H0C12	H0C28
310	H1A12	H1A28	H1C12	H1C28	H2A12	H2A28	H2C12	H2C28
320	L4F	L4A	L4C	diaS13	H0A13	H0A29	H0C13	H0C29
330	H1A13	H1A29	H1C13	H1C29	H2A13	H2A29	H2C13	H2C29
340	L5F	L5A	L5C	diaS14	H0A14	H0A30	H0C14	H0C30
350	H1A14	H1A30	H1C14	H1C30	H2A14	H2A30	H2C14	H2C30
360	tmrF	itr	etr	diaS15	H0A15	H0A31	H0C15	H0C31
370	H1A15	H1A31	H1C15	H1C31	H2A15	H2A31	H2C15	H2C31

Legend:

A*cc*	active data interrupt, subchannel *cc*
C*cc*	configuration data interrupt, subchannel *cc*
cdr	card reader
cns	FNP console
dia	direct interface adapter
etr	elapsed time rollover interrupt
F	fault interrupt
H*n*	high-speed line adapter *n*
itr	interval timer runout interrupt
L*n*	low-speed line adapter *n*
prt	printer
S	special interrupt
S*xx*	special interrupt from DIA mailbox *xx*
T	terminate interrupt
tmr	timer channel

Table 6-6. FNP Interrupt Cells

level	abs addr	bit position 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
0	400	IOM channel fault interrupts
1	401	IOM channel special interrupts
2	402	IOM channel terminate interrupts
3	403	DIA special interrupts
4	404	HSLA#0 subchannels 0-15, active
5	405	HSLA#0 subchannels 16-31, active
6	406	HSLA#0 subchannels 0-15, configuration
7	407	HSLA#0 subchannels 16-31, configuration
10	410	HSLA#1 subchannels 0-15, active
11	411	HSLA#1 subchannels 16-31, active
12	412	HSLA#1 subchannels 0-15, configuration
13	413	HSLA#1 subchannels 16-31, configuration
14	414	HSLA#2 subchannels 0-15, active
15	415	HSLA#2 subchannels 16-31, active
16	416	HSLA#2 subchannels 0-15, configuration
17	417	HSLA#2 subchannels 16-31, configuration

n o t u s e d

FAULT VECTORS

The processor fault vector base in the FNP is 440(8), and there are eight hardware faults defined.

Address	Fault
440	power off.
441	power on.
442	memory parity.
443	invalid operation code.
444	overflow.
445	invalid memory operation.
446	divide check.
447	invalid program interrupt.

In addition to these eight hardware faults, there are two simulated faults that are set by the software for the condition specified. There are no "fault vector" locations associated with the simulated faults.

1. unexpected interrupt.
2. console abort command.

FNP Store Map

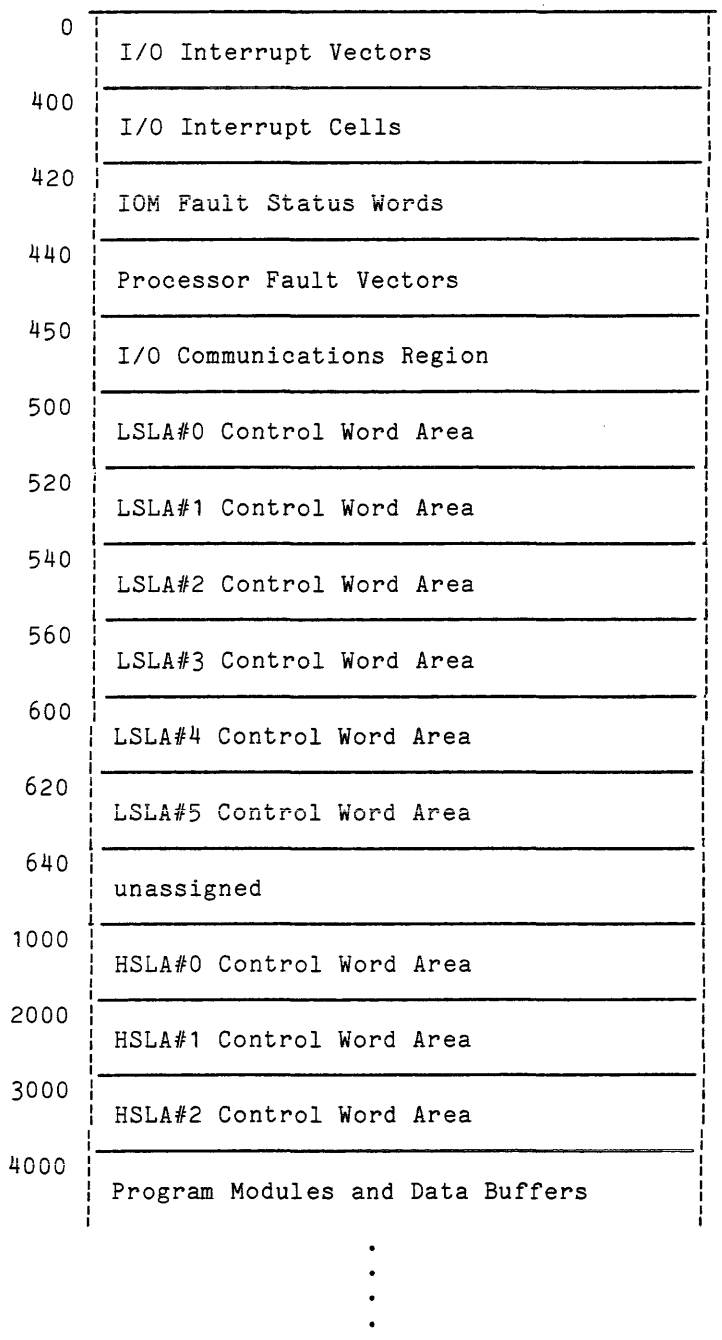


Figure 6-28. FNP Store Map

SECTION VII

MULTICS ENVIRONMENT

This section describes very broadly the environment in which Multics and the Multics user processes execute. The reader desiring more detail is referred to the entire set of Multics Program Logic Manuals (PLMs) and to the Multics module listings.

MAIN MEMORY MAPS

The following paragraphs describe the gross allocation of main memory during the three distinctly different Multics operational environments: BOS, bootstrap1, and service.

BOS Environment

BOS operates in segmented, nonpaged appending mode with exactly eight defined segments. The eight pointer registers are loaded with fixed segment numbers and the segment base and bound values are manipulated according to the requirements of the code.

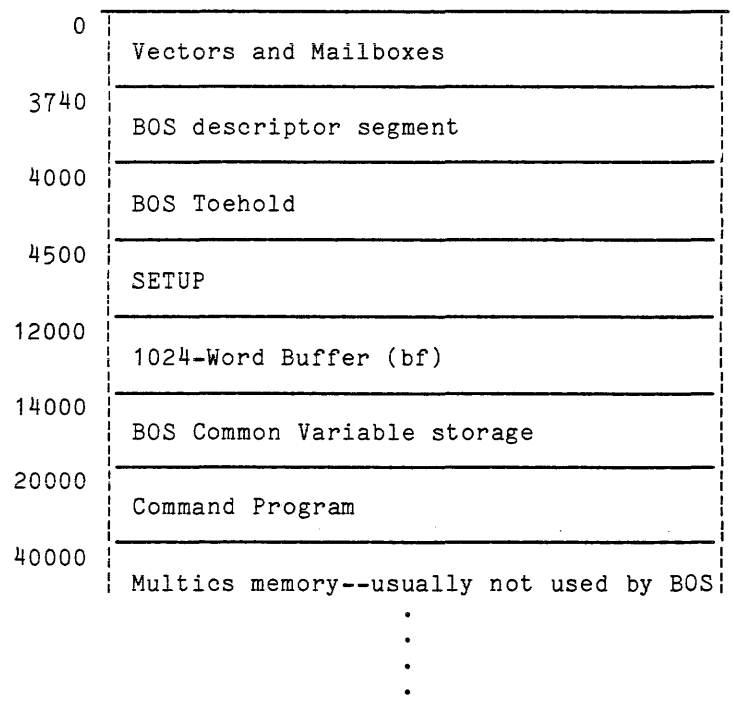


Figure 7-1. Main Memory Map for BOS

Bootstrap1 Environment

The bootstrap1 program runs in fully segmented, unpaged appending mode.

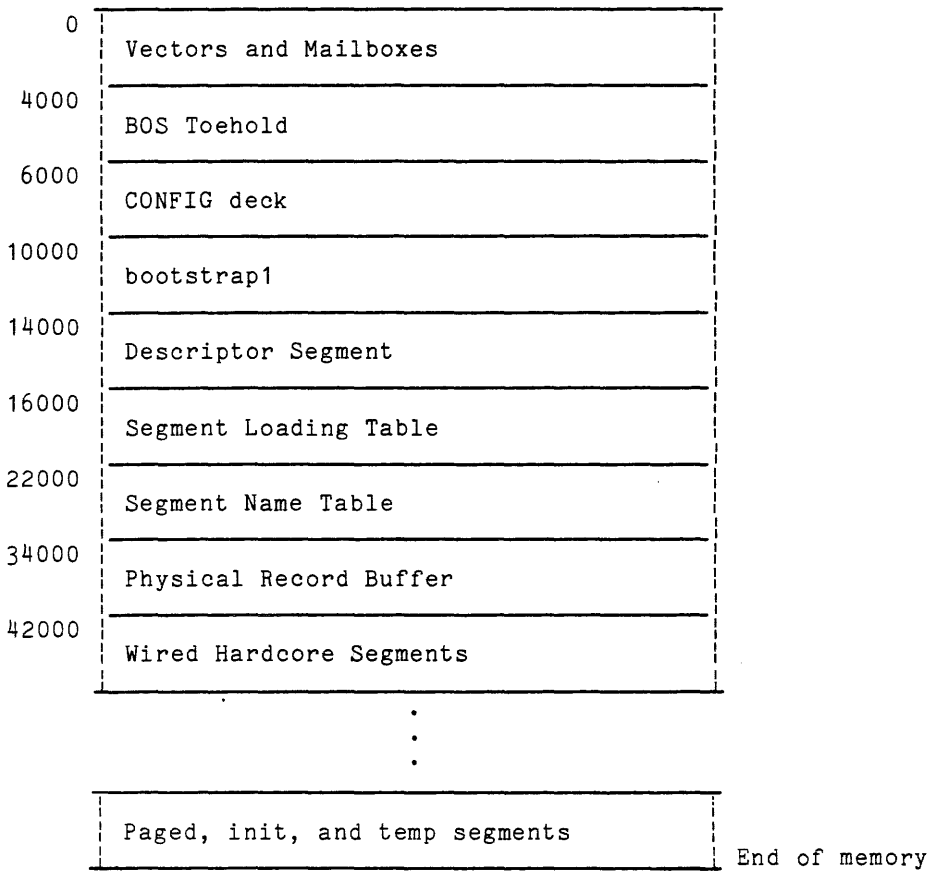


Figure 7-2. Main Memory Map for Bootstrap1

Service Environment

Multics service mode runs in fully segmented, fully paged appending mode.

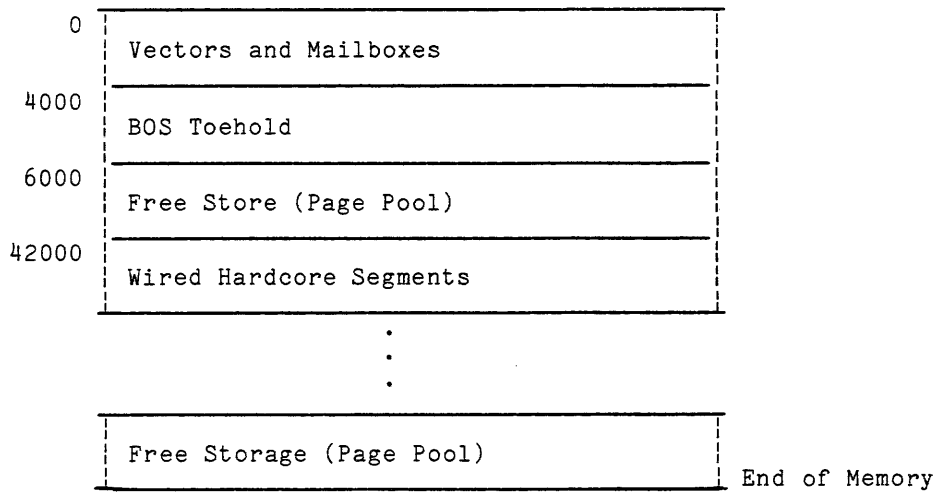


Figure 7-3. Main Memory Map for Multics Service

INTERRUPT ASSIGNMENTS

Table 7-1. Interrupt Assignments

Dec	Oct	F/I ADDR in SCU data	Assignment
0	0	00	
1	1	02	
2	2	04	BSC #0
3	3	06	
4	4	10	IOM #0 overhead
5	5	12	IOM #1 overhead
6	6	14	IOM #2 overhead
7	7	16	IOM #3 overhead
8	10	20	
9	11	22	
10	12	24	BSC #1
11	13	26	
12	14	30	IOM #0 terminate
13	15	32	IOM #1 terminate
14	16	34	IOM #2 terminate
15	17	36	IOM #3 terminate

```
16  20      40
17  21      42
18  22      44      software (system trouble)
19  23      46      software (syserr log)
20  24      50      IOM #0 marker
21  25      52      IOM #1 marker
22  26      54      IOM #2 marker
23  27      56      IOM #3 marker

24  30      60      software (processor initiate)
25  31      62      software (preempt)
26  32      64      software (stop)
27  33      66      software (quit)
28  34      70      IOM #0 special
29  35      72      IOM #1 special
30  36      74      IOM #2 special
31  37      76      IOM #3 special
```

MACHINE CONDITIONS DATA LAYOUT

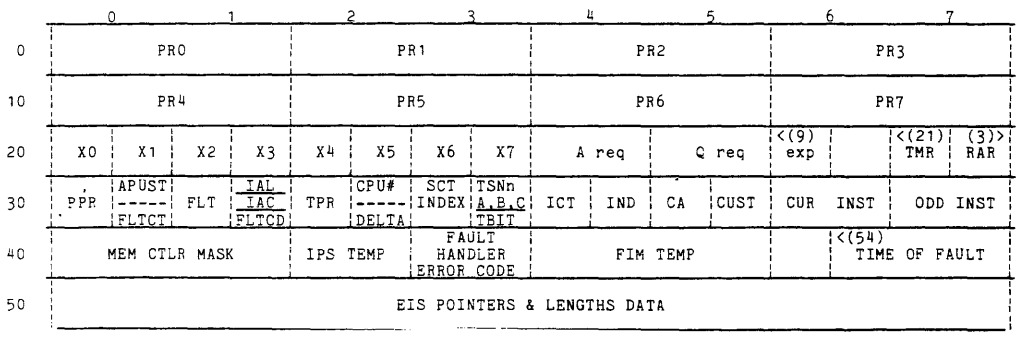


Figure 7-4. Machine Conditions Data Layout

STACKS

Stack Header Layout

PL/I Declaration (stack_header.incl.pl1)

```
dcl 1 stack_header              based (sb) aligned,
    2 pad1 (4)                  fixed bin,
    2 old_lot_ptr               ptr,                    /* obsolete */
    2 pad2 (10)                 fixed bin,
    2 null_ptr                  ptr,
    2 stack_begin_ptr           ptr,
    2 stack_end_ptr             ptr,
    2 lot_ptr                   ptr,
    2 signal_ptr                ptr,
    2 bar_mode_sp               ptr,
    2 pl1_operators_ptr         ptr,
    2 call_op_ptr               ptr,
```

```
2 push_op_ptr              ptr,
2 return_op_ptr            ptr,
2 return_no_pop_op_ptr     ptr,
2 entry_op_ptr             ptr,
2 trans_op_tv_ptr          ptr,
2 isot_ptr                 ptr,
2 pad3 (2)                 fixed bin,
2 unwinder_ptr             ptr,
2 stack_header_end         fixed bin;
```

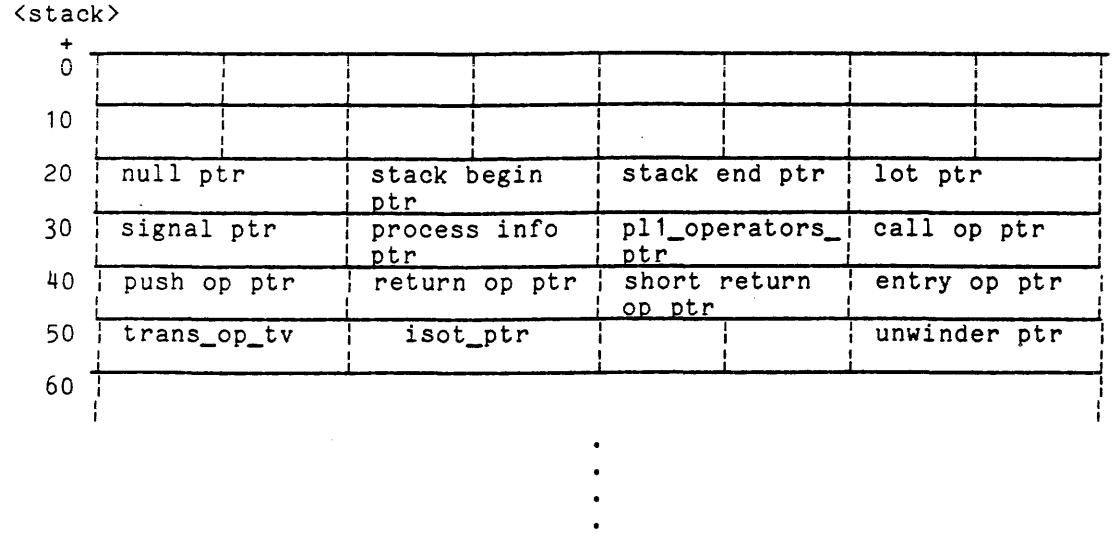


Figure 7-5. Stack Header Layout

Stack Frame Layout

PL/I Declaration (stack_frame.incl.pl1)

```
dcl 1 stack_frame based(sp)           aligned,
    2 pointer_registers(0 : 7)        ptr,
    2 prev_sp                         ptr,
    2 next_sp                         ptr,
    2 return_ptr                      ptr,
    2 entry_ptr                       ptr,
    2 operator_and_lp_ptr             ptr,
    2 arg_ptr                         ptr,
    2 static_ptr                      ptr unaligned,
    2 reserved bit(36),
    2 on_unit_relp1 bit(18)           unaligned,
    2 on_unit_relp2 bit(18)           unaligned,
    2 translator_id bit(18)           unaligned,
    2 operator_return_offset bit(18)  unaligned;
```

The argument list structures are explained more fully in the MPM Subsystem Writers' Guide, Order No. AK92. Briefly, call_type tells what kind of call is being made. The following values are defined: 0, for a quick (intra-segment) call; 4, for a non-quick call; 8, for a call made through an entry variable. In this last case, an environment pointer is also passed, and the second form of arg list is the one used.

<arglist>

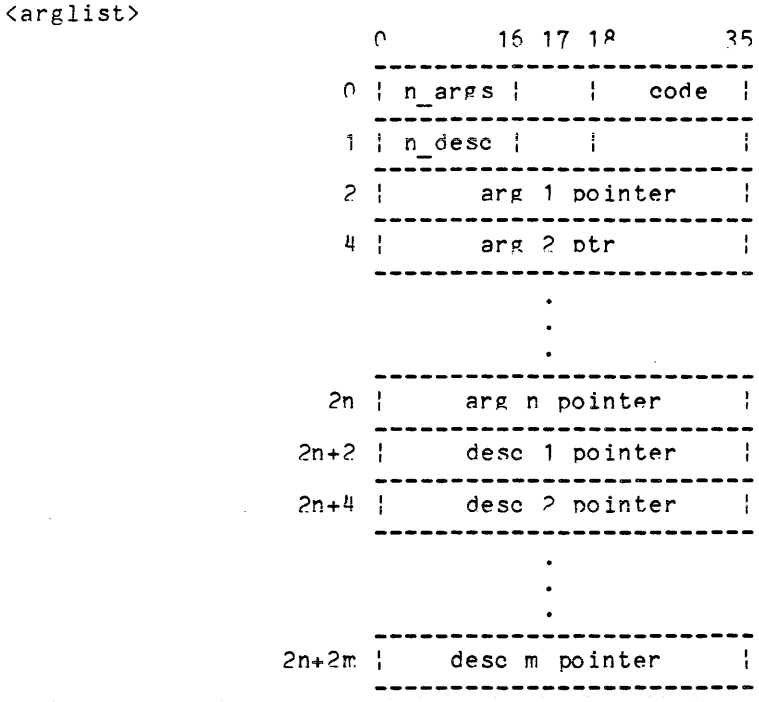


Figure 7-7. Argument List Layout For Call Without Environment Pointer

Argument Descriptor

An argument descriptor is pointed to by a descriptor pointer in an argument list. (For a full discussion of descriptors, refer to the MPM Subsystem Writers' Guide, Order No. AK92.) Its format is given by the following structure:

PL/I Declaration (arg_descriptor.incl.pl1)

```
dcl 1 arg_descriptor              based aligned,
    2 flag                        bit(1) unal,
    2 type                        fixed bin(6) unsigned unal,
    2 packed                      bit(1) unal,
    2 number_dims                 fixed bin(4) unsigned unal,
    2 size                        fixed bin(24) unsigned unal;

dcl 1 fixed_arg_descriptor        based aligned,
    2 flag                        bit(1) unal,
    2 type                        fixed bin(6) unsigned unal,
    2 packed                      bit(1) unal,
    2 number_dims                 fixed bin(4) unsigned unal,
    2 scale                       fixed bin(11) unal,
    2 precision                   fixed bin(12) unsigned unal;
```

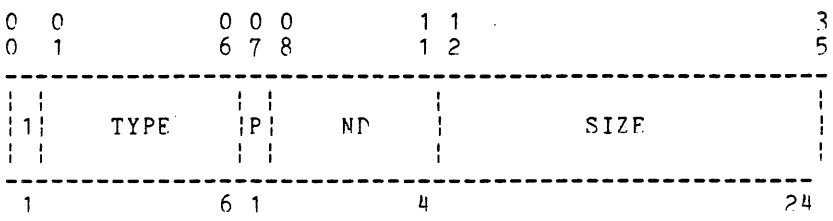


Figure 7-8. Argument Descriptor Format

Legend:

TYPE (arg_descriptor.type) descriptor type for data being described. Refer to the MPM Subsystem Writers' Guide, Order No.AK92 for a complete list of descriptor types.

P (arg_descriptor.packed)
0 = unpacked.
1 = packed.

ND (arg_descriptor.number_dims) number of dimensions if item is an array.

SIZE (arg_descriptor.size) length of string data, or number of members for structure data.

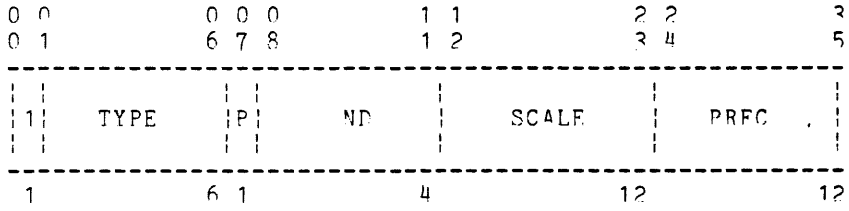


Figure 7-9. Fixed Point Argument Descriptor

Legend: see legend for Argument Descriptor Format

SCALE (fixed_arg_descriptor.scale) arithmetic scale factor of data.

PREC (fixed_arg_descriptor.precision) precision of data.

Table 7-2. ASCII Character Chart

	0	1	2	3	4	5	6	7
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
010	BS	HT	NL	VT	NP	CR	SO	SI
020	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
030	CAN	EM	SUB	ESC	FS	GS	RS	US
040		!	"	#	$	%	&	'
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	{		}	~	PAD

This page intentionally left blank.

APPENDIX A

ACRONYM DEFINITIONS

ACU	automatic call unit
APU	appending unit
APU-HR	appending unit history registers
ASCII	American Standard Code for Information Interchange
AST	active segment table
BAR	base address register
BAW	base address word
BCD	Binary-Coded Decimal
BOS	Bootload Operating System
BOT	beginning of tape
BSC	bulk store controller
BSU	bulk store unit
CA	controller adapter
CCC	character control character
CCT	character control table
CPI	common peripheral interface
CPU	central processor unit
CSB	current status block
CU	control unit
CU-HR	control unit history registers
DA	device adapter
DAI	device adapter interface
DBR	descriptor base register
DCB	data control block
DCW	data control word
DIA	direct interface adapter
DLI	device level interface
DSBR	descriptor segment base register (usually referred to as DBR)
DU	decimal unit
DU-HR	decimal unit history registers
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EDAC	error detection and correction
EIA	Electronic Industries Association
EIMA	execute interrupt mask assignment
EIS	extended instruction set
EOF	end of file
ESC	escape
ESN	effective segment number
FIM	fault intercept module
FNP	DATANET 6600 Front-End Network Processor or DATANET 355 Front-End Network Processor
HSLA	high-speed line adapter
IA	illegal action
IAI	internal adapter interface
ICT	instruction counter
ICW	indirect control word
IDCW	instruction DCW
IMW	interrupt multiplex word
IOC	illegal opcode
IOM	input/output multiplexer
IPR	illegal procedure
IPS	interprocess signal
ITP	indirect-to-pointer (pointer pair)
ITS	indirect-to-segment (pointer pair)

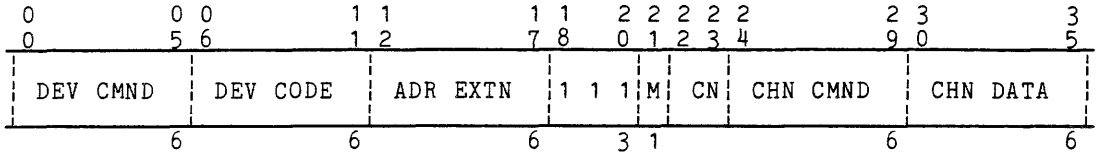
LPW	list pointer word
LSB	least significant bit(s)
LSD	least significant digit(s)
LSLA	low-speed line adapter
MF	modification field
MFM	modified frequency modulation
MOS	metal oxide semiconductor
MPC	microprogrammed peripheral controller
MSB	most significant bit(s)
MSD	most significant digit(s)
NRZI	nonreturn to zero; change on ones (tape drive modes)
OPI	operational-in (line)
OU	operations unit
OU-HR	operations unit history registers
PCW	peripheral control word
PE	phase encoded (tape drive modes)
PRNUM	pointer register number
PRR	procedure ring register
PSI	peripheral subsystem interface
PSR	procedure segment register
PTW	page table word
PTWAM	page table word associative memory
RPS	rotational position sensing
RSCR	read system controller registers
RSR	read status register (MPC command)
SC	system controller
SCU	store control unit
SCW	status control word
SDW	segment descriptor word
SDWAM	segment descriptor word associative memory
SNR	segment number register (part of pointer register)
SST	system segment table
T&D	test and diagnostics
TCA	tape controller adapter
TDCW	transfer DCW
TPR	temporary pointer register
TRR	temporary ring register
TSR	temporary segment register
URC	unit record controller
URMPC	unit record microprogrammed peripheral controller
VFC	vertical format control
VFU	vertical format unit
ZAC	zone address control

APPENDIX B

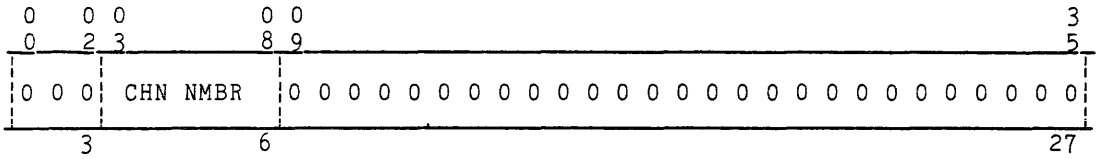
DATA AND CONTROL WORD FORMATS

This appendix consists of information described in more detail in other sections of this manual. The information is repeated here to provide a quick and easy reference for user convenience.

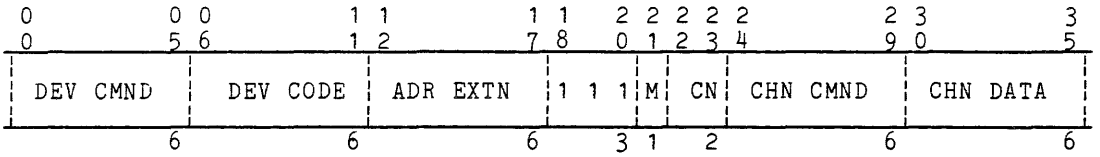
Even Word:



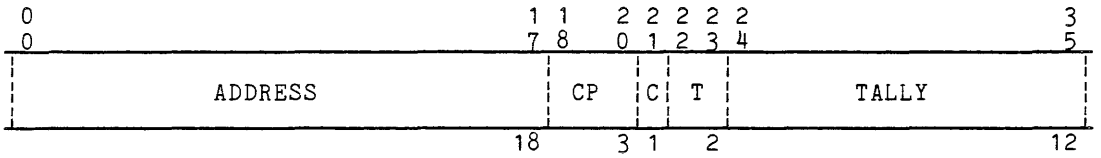
Odd Word:



IOM PCW & DCW Format

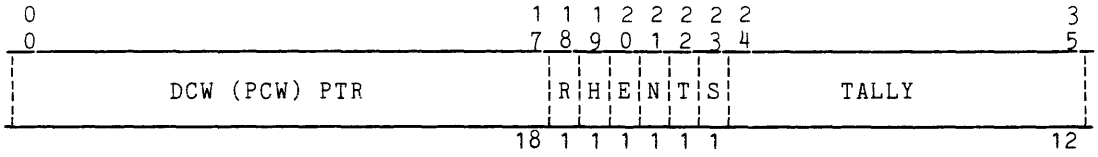


IOM DCW Format

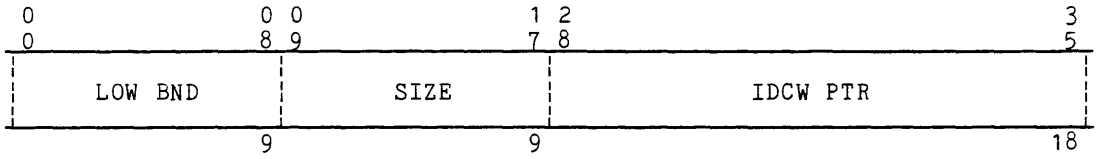


IOM TDCW Format

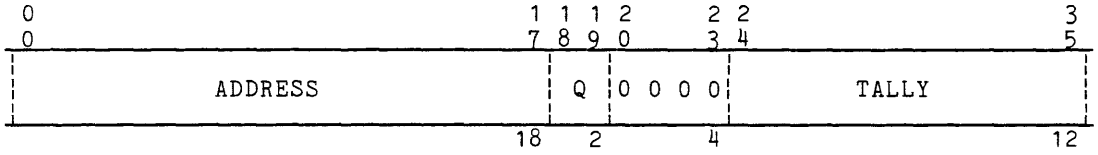
Figure B-1. IOM Formats



IOM LPW Format

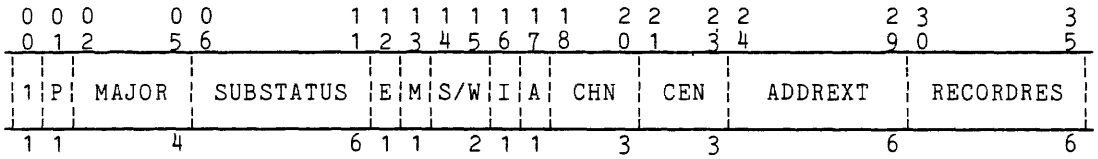


IOM LPW Extension Format

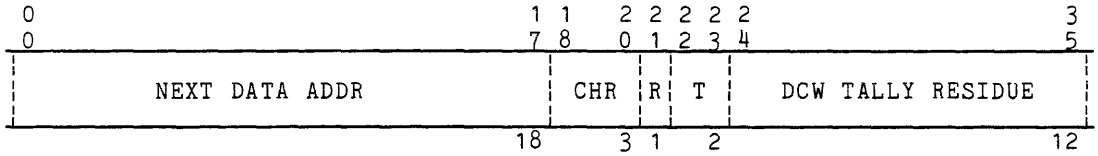


IOM SCW Format

Even Word:



Odd Word:



IOM Status Format

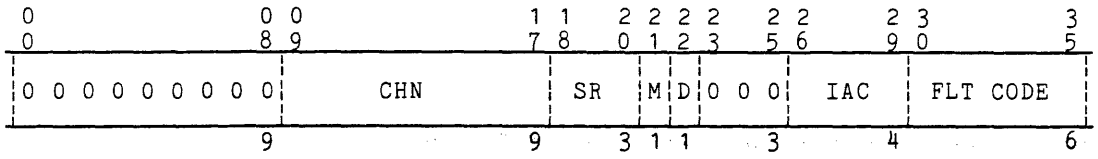


Figure B-1 (cont). IOM Formats

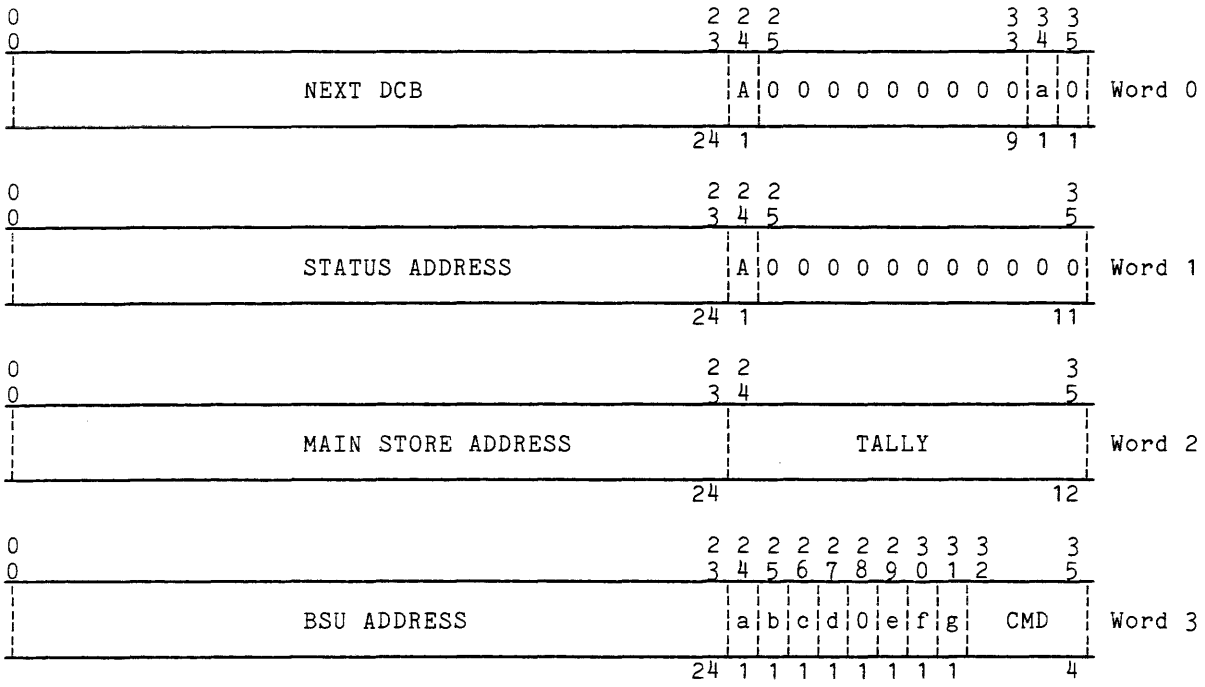


Figure B-2. Bulk Store Data Control Block (DCB) Format

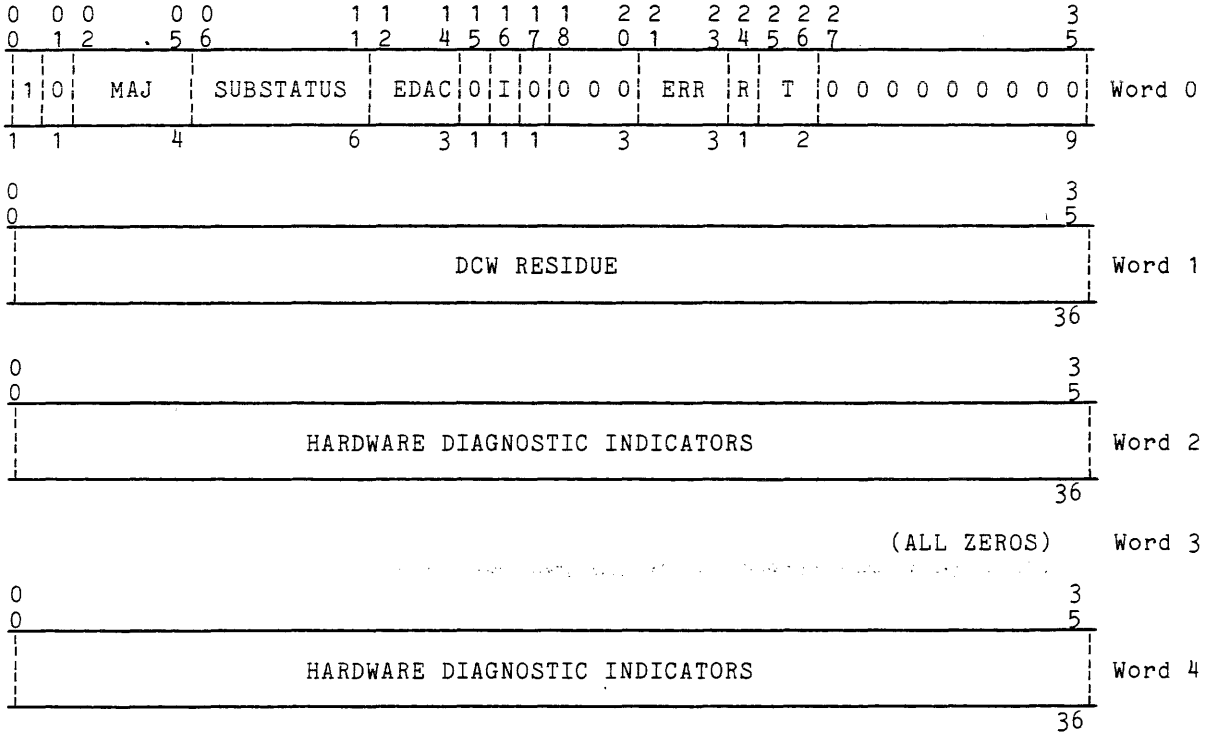


Figure B-3. Bulk Store DCB Status Block Format

Figure B-4. Bulk Store Current Status Block (CSB) Format

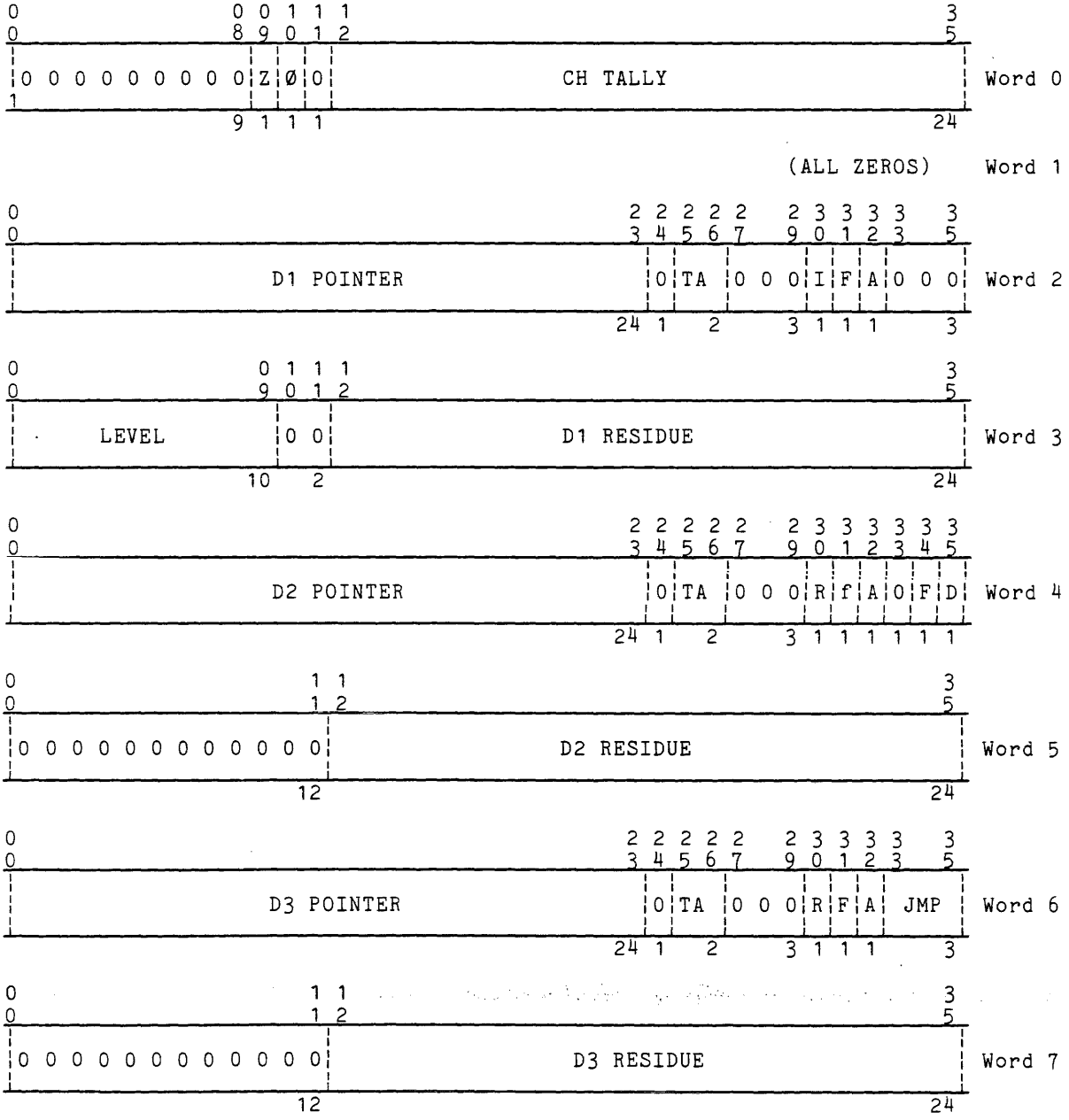


Figure B-5. DU Pointers and Lengths Format

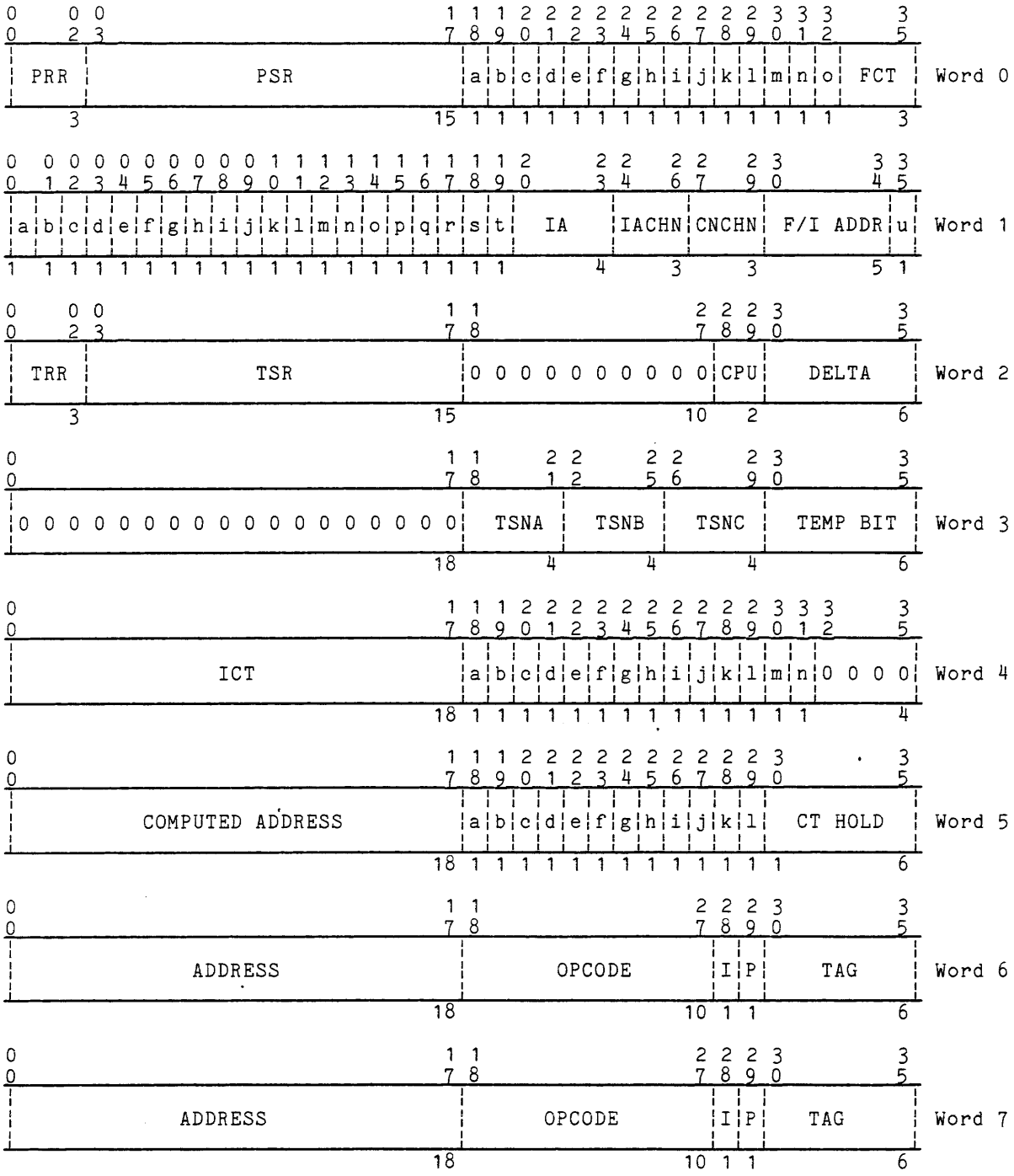


Figure B-6. SCU Data Format

Table B-1. Processor Fault Numbers

Oct	Dec	F/I ADDR in SCU data	Name	Mnemonic	Priority	Group	Mode
0	0	01	Shutdown	sdf	27	VII	M/G
1	1	03	Store	str	10	IV	M/G
2	2	05	Master Mode Entry 1	mme1	11	V	M/G
3	3	07	Fault Tag 1	ftg1	17	V	M/G
4	4	11	Timer Runout	tro	26	VI	M/G
5	5	13	Command	cmd	9	IV	M/G
6	6	15	Derail	drl	15	V	M/G
7	7	17	Lockup	luf	5	IV	M/G
10	8	21	Connect	con	25	VII	M/G
11	9	23	Parity	par	8	IV	M/G
12	10	25	Illegal Procedure	ipr	16	V	M/G
13	11	27	Op Not Complete	onc	4	II	M/G
14	12	31	Startup	suf	1	I	M/G
15	13	33	Overflow	ofl	7	III	M/G
16	14	35	Divide Check	dvck	6	III	M/G
17	15	37	Execute	exc	2	I	M/G
20	16	41	Directed Fault 0	dft0	20	VI	M
21	17	43	Directed Fault 1	dft1	21	VI	M
22	18	45	Directed Fault 2	dft2	22	VI	M
23	19	47	Directed Fault 3	dft3	23	VI	M
24	20	51	Access Violation	acv	24	VI	M
25	21	53	Master Mode Entry 2	mme2	12	V	M
26	22	55	Master Mode Entry 3	mme3	13	V	M
27	23	57	Master Mode Entry 4	mme4	14	V	M
30	24	61	Fault Tag 2	ftg2	18	V	M
31	25	63	Fault Tag 3	ftg3	19	V	M
32	26	65	Unassigned				
33	27	67	Unassigned				
34	28	71	Unassigned				
35	29	73	Unassigned				
36	30	75	Unassigned				
37	31	77	Trouble	trb	3	II	M

APPENDIX C

PERIPHERAL STATUS

This section describes the MAJOR and SUBSTATUS fields of the IOM channel status data shown in Figure 3-10. MAJOR and SUBSTSATUS in this section are given in octal form.

CARD READERS

If the device is a card reader, the MAJOR and SUBSTATUS fields are interpreted according to the list below. Substatuses marked with an asterisk (*) may be ORed within the same major status.

MAJOR SUBSTATUS

40 CHANNEL READY.

- 00 channel ready. If received as an initiation interrupt (I = "1") in response to a reqs or ress command, the channel and device are ready to accept a new command. If received as a termination interrupt, the last command was executed error free and the channel and device are ready to accept a new command.
- 01 (CRU1050) 51-column cards.
  As above except that the input card hopper contains 51-column cards.

41 DEVICE BUSY.

- 00 (CRZ201) one the following occurred:
  1. A "feed" or "stack" command was being executed and a reqs or ress command was received.
  2. A "feed" command was being executed and another feed command was received.
  3. A "stack" command was being executed and another stack command was received.
  4. A command was received with a card in the read head.

42 ATTENTION.

- 00 (CRU1050) offline (device power off).
  The unit record controller (URC) MPC could not communicate with the device. The operational-in (OPI) line of the device adapter interface (DAI) is reset.

(CRU1050) hopper/stacker alert.
The input card hopper is empty and/or the output card stacker is full.

02* manual halt.
The MANUAL HALT switch has been pressed or a safety interlock is open.

05* (CRZ201) last batch.
The LAST BATCH switch has been pressed and the input card hopper is empty.

10* feed alert.
The next card from the input hopper failed to feed properly.

20* card jam.
The trailing edge of a card failed to reach a photocell station in the card track within the specified time after the detection of the leading edge of the card at the station.

40* (CRZ201) read alert.
One or more of the following occurred:
1. read photocell light current error.
2. read photocell dark current error.
3. read strobe count error.
4. card-in-head error.
5. internal parity error.
6. read error test check failure.

(CRU1050) read alert.
One or more of the following occurred:
1. read photocell light current error.
2. read photocell dark current error.
3. read strobe count error.
4. card-in-head timing error.

50* (CRZ201) sneak feed.
Prior to receipt of the current command, one or more cards passed through the card reader without a command having been given.

43 DATA ALERT.

01 (CRZ201) transfer timing alert.
The IOM failed to accept (read) data characters at a rate compatible with the transfer rate of the card reader.

02* validity alert.
During execution of a "read card decimal" command, an invalid character was detected. An ignore character ("?" = 17(8)) is stored in the card image in place of each invalid character.

04* dual read failure.
A discrepancy was detected in the contents of a card column as read by the dual read head of the card reader. In decimal mode, an ignore character ("?" = 17(8)) is stored in the card image in place of the invalid column. In binary mode, two ignore characters are stored in place of the invalid column.

10 (CRZ201) no read command.
A card fed by a "feed card" command entered the read station before a "read" command was received. The "read" command must be received within 9 milliseconds of the preceding "feed card" command.

45 COMMAND REJECT.

01 invalid command.
The device is unable to recognize the command code in the PCW or IDCW.

02 (CRZ201) no card committed.
A stack command was received at a time other than within 6 milliseconds after a card left the read head.

(CRU1050) invalid device code.
The device code specifies a device that is not configured.

04 (CRZ201) late read command.
A "read" command was received after a card entered the read station. Also see DATA ALERT, substatus 10, no "read" command described above.

(CRU1050) IDCW parity.
A parity error occurred on the logical channel number field in an IDCW from the IOM.

47 LOAD OPERATION COMPLETE.

00 (CRZ201) load complete.
A load card (boot) sequence has completed with no DATA ALERT or ATTENTION conditions.

52 MPC DEVICE ATTENTION.

01 (CRU1050) IAI error.
A parity error was detected on the internal adapter interface (IAI) between the multiplexer adapter and the URMPC.

02 (CRU1050) DAI error (no media movement).
One of the following was detected on the device adapter interface (DAI) between the URMPC and the device adapter (DA):
1. Parity error detected by the DA.
2. Parity error detected by the URMPC.
3. Error timeout detected by the URMPC.

04 (CRU1050) DA transfer error.
A timing error was detected by the DA during device operation.

10 (CRU1050) invalid punch.
An invalid decimal punch combination (two or more punches in rows 1-7) was detected by the DA. No character substitution is made in the card image.

53 MPC DEVICE DATA ALERT.

01 (CRU1050) transmission parity error.
A parity error was detected by the peripheral subsystem interface (PSI) during transfer of data from the IOM to the URMPC.

05 (CRU1050) DAI error (with media movement).
One of the following was detected on the DAI between the URMPC and the DA:
1. Parity error detected by the DA.
2. Parity error detected by the URMPC.
3. Error timeout detected by the URMPC.

55 MPC COMMAND REJECT.

01 (CRU1050) illegal procedure.
The URMPC is in suspend mode and will accept only special controller commands.

02 (CRU1050) illegal logical channel number.
The logical channel number sent with an IDCW was illegal (not 00-07 hexadecimal).

10 (CRU1050) device reserved.
The device requested is reserved to another PSI and is not available for use.

60 POWER OFF.

00 (CRZ201) power off.
The device is powered off or is not cabled to the CPI channel in the IOM.

(CRU1050) power off.
The URMPC is powered off, is not cabled to the PSI channel in the IOM, or has lost its personality firmware.

CARD PUNCHES

For a card punch, the MAJOR and SUBSTATUS fields are interpreted in the following manner. Substatuses marked with an asterisk (*) may be ORed within the same major status.

MAJOR SUBSTATUS

40 CHANNEL READY.

00 channel ready.
If received as an initiation interrupt (I = "1") in response to a reqs or ress command, the channel and device are ready to accept a new command. If received as a termination interrupt, the last command was executed error free and the channel and device are ready to accept a new command.

42 ATTENTION.

00 (PCU0120) offline.
The unit record controller (URC) MPC could not communicate with the device. The operational-in (OPI) line of the device adapter interface (DAI) is reset.

01* hopper/stacker alert.
The input card hopper is empty and/or the output card stacker is full.

02* manual halt.
The MANUAL HALT switch has been pressed or a safety interlock is open.

04* chad box full.
The chad receptacle is full.

10* feed failure.
A card from the input hopper failed to feed into the punch mechanism.

20* card jam.
One or more cards were improperly loaded in the input hopper or a card failed to progress at the proper time from one station to the next in the card track.

43 DATA ALERT.

00 or 01* (CPZ201) transfer timing alert.
The IOM did not send (write) data characters at a rate compatible with the transfer rate of the card punch.

02* (CPZ201) transmission parity alert.
A parity error was detected on a data character received from the IOM.

04* (CPZ201) punch alert.
A count of holes punched in a card was compared with a calculated hole count and the counts did not agree.

10 (PCU0120) punch alert.
A count of holes punched in a card was compared with a calculated hole count and the counts did not agree.

45 COMMAND REJECTED.

01 invalid command.
The channel is unable to recognize the device command code in the PCW or IDCW.

52 MPC DEVICE ATTENTION.

01 (PCU0120) IAI error.
A parity error was detected on the internal adapter interface (IAI) between the multiplexer adapter and the URMPC.

02 (PCU0120) DAI error.
One of the following was detected on the device adapter interface (DAI) between the URMPC and the device adapter (DA):
1. Parity error detected by the DA.
2. Parity error detected by the URMPC.
3. Error timeout detected by the URMPC.

04 (PCU0120) DA transfer error.
A timing error was detected by the DA during device operation.

53 MPC DEVICE DATA ALERT.

01 (PCU0120) transmission parity error.
A parity error was detected by the peripheral subsystem interface (PSI) during transfer of data from the IOM to the URMPC.

05 (PCU0120) DAI error.
One of the following was detected on the DAI between the URMPC and the DA:
1. Parity error detected by the DA.
2. Parity error detected by the URMPC.
3. Error timeout detected by the URMPC.

06 (PCU0120) PSI data overflow.
More than 256 characters were received from the IOM.

55 MPC COMMAND REJECT.

01 (PCU0120) illegal procedure.
The URMPC is in suspend mode and will accept only special controller commands.

02 (PCU0120) invalid logical channel number.
The logical channel number sent with an IDCW was invalid (not 00-07 hexadecimal).

10 (PCU0120) device reserved.
The device requested is reserved to another PSI and is not available for use.

60 POWER OFF.

00 (CPZ201) power off.
The device is powered off or is not cabled to the CPI channel of the IOM.

(PCU0120) power off.
The URMPC is powered off, is not cabled to the PSI channel of the IOM, or has lost its personality firmware.

LINE PRINTERS

For a line printer, the MAJOR and SUBSTATUS fields are interpreted in the following manner. Substatuses marked with an asterisk (*) may be ORed within the same major status.

MAJOR SUBSTATUS

40 CHANNEL READY.

00 channel ready.
If received as an initiation interrupt (I = "1") in response to a reqs or ress command, the channel and device are ready to accept a new command. If received as a termination interrupt, the last command was executed error free and the channel and device are ready to accept a new command.

01 print one line.
Same as substatus 00; in addition printer control button 1 (PRINT 1 LINE) has been activated.

02 forward space.
Same as substatus 00; in addition printer control button 2 (FORWARD SPACE) has been activated.

03 forward to top of page.
Same as substatus 00; in addition printer control button 3 (FORWARD TOP) has been activated.

04 invalid line.
Same as substatus 00; in addition printer control button 4 (INVALID LINE) has been activated.

05 reverse/rewind.
Same as substatus 00; in addition printer control button 5 (REVERSE REWIND) has been activated.

06 backspace.
Same as substatus 00; in addition printer control button 6 (BACK SPACE) has been activated.

07 backspace top of page.
Same as substatus 00; in addition printer control button 7 (BACK SPACE TOP) has been activated.

42 ATTENTION.

00 (PRT300/301) power fault.
One of the following has occurred:
1. A thermal fault.
2. A printer power fault.
3. A feed fault.
4. A power ON/OFF sequence.
5. The printer is powered off.

(PRT303) power fault.
One of the following has occurred:
1. A thermal fault in the printer mechanism.
2. Power not on in device electronics.
3. Power fault in the 36 volt supply.
4. Power fault in the printer mechanism.
5. Power fault in the -5 or -12 volt supplies.

(PRU1200/PRU1600) power fault.
One of the following has occurred:
1. A thermal fault in the printer mechanism
2. Power not on in device electronics.
3. Print power supply fault.
4. Slew power supply fault.
5. Phase fault on AC primary line.
6. Short circuit fault on hammer drivers.
7. Finger sensor fault.
8. Breaker AC.
9. Air flow check.

01* out of paper.
One of the following has occurred:
1. The forms detectors failed to sense the presence of a form.
2. A top-of-page occurred after a paper low condition.

02* manual halt.
One of the following has occurred:
1. The MANUAL HALT switch or one of the printer control buttons has been activated.
2. The POWER ON switch was activated while the printer was powered off.
3. The printer yoke has been opened.

04* (PRT202, PRT300/301, PRT303) VFU tape alert.
One of the following has occurred:
1. VFU tape horizontal parity error.
2. VFU tape was not present.
3. VFU tape was not properly installed.
4. Holes were punched in both channel 5 (start automatic slew) and channel 6 (stop automatic slew) of the same vertical line position of the VFU tape.

10* check.
One or more of the following has occurred:
1. Hammer driver fuse failure.
2. Paper slew fuse failure.
3. Incomplete printout; all characters received from the IOM were not printed.
4. (PRT202) Print wheel out of sequence.

43 DATA ALERT.

00 (PRT300/301, PRT303) invalid character code or image buffer alert.
One of the following has occurred:
1. An image load alert in which an invalid character was detected in the image or less than 288 characters were received during image loading.
2. An image buffer overflow condition in which more than 288 characters were received during image loading.
3. A print incomplete condition in which one or more data characters failed to compare with an image character during a complete cycle of the print train.

(PRU1200/PRU1600) invalid character code or image buffer alert.
One of the following has occurred:
1. An image load alert in which an invalid character was detected in the image or less than 240 characters were received during image loading.
2. An image buffer overflow condition in which more than 240 characters were received during image loading.
3. A print incomplete condition in which one or more data characters failed to compare with an image character during a complete cycle of the print belt.

01* transfer timing alert.
The IOM did not send (write) data characters at a rate compatible with the transfer rate of the printer.

02* alert before printing started.
One of the following has occurred:
1. A parity error was detected on a data character received by the printer.
2. Print buffer overflow was sensed when more than 136 characters (160 FOR PRU1200/1600 with 160-character option) were received from the IOM before receipt of a slew character or end-data-transfer signal.
3. A transfer timing alert condition exists.
4. A top-of-page echo occurred while the printer was busy.

04* alert after printing started.
A parity error was detected on a data character in the print buffer.

10* paper low warning alert.
The last page of the form has passed the first forms detector and approximately 2.4 inches of form remains.

20* slew/paper motion alert.
More than two top-of-page indications were sensed within a single slew operation.

40* (PRT202, PRT300/301, PRT303) top-of-page echo.
The form has slewed to top-of-page as a result of a slew command other than the explicit slew-to-top-of-page.

45 COMMAND REJECTED.

00 (PRU1200/PRU1600) VFC image not loaded.
A print or slew command was issued before the VFC image of the printer was loaded.

01* invalid command.
The channel was unable to recognize the device command code in the PCW or IDCW.

02* (PRT300/301, PRT303, PRU1200/1600) invalid device code.
An invalid device was detected in an IDCW for the printer.

04* (PRT300/301) device/command code parity alert.
A parity error was caused by an incorrect device and/or command code.

10 (PRU1200/PRU1600) train image not loaded.
A print or slew command was issued before the train image of the printer was loaded.

20* feed alert on last slew operation.
The previous operation resulted in a slew error. See also DATA ALERT, substatus 20 (slew/paper motion alert) above.

40* top-of-page echo on last slew operation.
The last command resulted in a termination interrupt with DATA ALERT, top-of-page echo (substatus 40, described above).

52 MPC DEVICE ATTENTION.

01 (PRT303, PRU1200/1600) IAI error.
A parity error was detected on the internal adapter interface (IAI) between the multiplexer adapter and the URMPC.

02 (PRT303, PRU1200/1600) DAI error.
One of the following was detected on the device adapter interface (DAI) between the URMPC and the device adapter (DA):
1. Parity error detected by the DA.
2. Parity error detected by the URMPC.
3. Error timeout detected by the URMPC.

53 MPC DEVICE DATA ALERT.

01 (PRT303, PRU1200/1600) transmission parity error
A parity error was detected on the PSI during data transfer from the IOM to the URMPC.

05 (PRT303, PRU1200/1600) DAI error.
One of the following was detected on the DAI between the URMPC and the DA:
1. Parity error detected by the URMPC.
2. Error timeout detected by the URMPC.

06 (PRT303, PRU1200/1600) PSI data overflow.
More than 512 characters were received from the IOM.

55 MPC COMMAND REJECT.

01 (PRT303, PRU1200/1600) illegal procedure.
The URMPC was in suspend mode and will accept only special controller commands.

02 (PRT303, PRU1200/1600) invalid logical channel number
The logical channel number sent with the IDCW was invalid (not 00-07 hexadecimal).

10 (PRT303, PRU1200/1600) device reserved.
The DA is reserved to another PSI and is not available for use.

60 POWER OFF.

00 (PRT202, PRT300/301) power off.
The device is powered off or not cabled to the common peripheral interface (CPI).

(PRT303, PRU1200/1600) power off.
The URMPC is powered off, not cabled to the PSI of the IOM, or has lost its personality firmware.

MAGNETIC TAPES

For a magnetic tape, the MAJOR and SUBSTATUS fields are interpreted in the following manner. Substatuses marked with an asterisk (*) may be ORed within the same major status.

MAJOR SUBSTATUS

40 CHANNEL READY.

00 channel ready.
If received as an initiation interrupt (I = "1") in response to a reqs or ress command, the channel and device are ready to accept a new command. If received as a termination interrupt, the last command executed error free and the channel is ready to accept a new command (the device may still be busy).

01* last tape unit write inhibited.
Same as substatus 00; in addition, the reel on the last tape drive addressed had no write-permit ring.

02* tape reel on load point.
Same as substatus 00; in addition, the last tape drive addressed was positioned at load point and is ready to process the first physical record.

04* ASA 9-track tape unit.
Same as substatus 00; in addition, the last tape drive addressed was an ASA 9-track unit.

14 (MTS500) ASCII alert.
The read-after-write check has detected an invalid EBCDIC character during a write ASCII/EBCDIC command.

20* (MTS500) 2-bit fill.
The final character from a 7-track read, a 9-track read ASCII, or a 9-track read EBCDIC has been padded with two low-order zero bits.

40* (MTS500) 4-bit fill.
The final character from a 7-track read, a 9-track read ASCII, or a 9-track read EBCDIC has been padded with four low-order zero bits.

60* (MTS500) 6-bit fill.
The final character from a 7-track read, a 9-track read ASCII, or a 9-track read EBCDIC has been padded with six low-order zero bits.

41 DEVICE BUSY.

01 (MTS500) in rewind.
The addressed tape drive is rewinding.

02 (MTS500) alternate channel in control.
The addressed tape drive is executing a command on the alternate channel.

04 (MTS500) device loading.
The addressed tape drive is in a tape loading cycle.

40 (MTS500) device reserved.
The addressed tape drive is reserved to the alternate channel as a result of a reserve device command.

42 ATTENTION.

01* tape write inhibited.
A write command was issued to a tape drive containing a reel without a write-permit ring or to a tape drive that has been protected with the set file protect command.

02* no such tape unit.
A command was issued to a tape drive that is not configured or is in offline mode.

04* tape unit standby.
A command was issued to a tape drive that is in standby mode.

10* (MTS500) tape unit check.
A malfunction in the addressed tape drive has rendered it inoperable.

20* blank tape on write.
A write operation was started on the addressed tape drive, but the read-after-write check was unable to detect any characters.

43 DATA ALERT.

01 transfer timing alert.
The IOM did not send (write) or accept (read) data characters at a rate compatible with the transfer rate of the tape subsystem.

02* blank tape on read.
After receipt of a read command, 30 inches (25 feet for MTS500) of tape were passed over without detection of a data character.

03* bit detected during erase.
A bit was detected in the portion of the tape that should have been erased as a result of an erase or write end-of-file command.

04* transmission parity alert.
Incorrect parity was detected on a data character received from the IOM during a write operation.

10* lateral parity alert.
A missing data character or a lateral (character) parity error was detected.

20* longitudinal parity alert.
The calculated check character did not agree with the recorded check character.

40* end-of-tape mark.
The tape drive detected the reflective end-of-tape foil during a write operation.

44 END OF FILE.

(C) single data character "C".
The single character "C" was read as a valid record during a read, backspace, or forward space operation.

17 EOF marker (7 track).
A valid 7-track end-of-file mark was detected.

23 EOF marker (9 track).
A valid 9-track end-of-file mark was detected.

77 data alert.
A DATA ALERT condition was detected during reading of an end-of-file record.

45 COMMAND REJECTED.

01* invalid command.
The channel was unable to recognize the device command code in the PCW or IDCW.

02* invalid device code.
The channel was unable to recognize the device code in the PCW or IDCW.

04* parity alert on device/command code.
A parity error was detected in the command code and/or device code of the PCW or IDCW.

10* tape on load point.
A "backspace" or "backspace file" command was issued to a tape drive positioned at load point.

20* attempted read after write on same unit.
A "read" or "forward space" command was issued to a tape drive immediately after a "write" command.

40* 9-track alert.
A 9-track command was issued to a 7-track tape drive.

47 LOAD OPERATION COMPLETE.

00 (MTS400) load complete.
A program load (boot) operation was completed error free.

50 CHANNEL BUSY.

00 (MTS400) busy.
The command was accepted but execution will be delayed until current command sequences are complete because the command requires the entire subsystem.

52 MPC DEVICE ATTENTION.

01 (MTS500) configuration error.
The personality firmware (control program) loaded into the MPC does not agree with the settings of the MPC configuration switches.

02 (MTS500) multiple devices.
The MPC has detected at least two devices with the same logical ID number.

03 (MTS500) device number error.
The MPC has detected at least one device with a logical ID number outside the allowed range of ID numbers.

10 (MTS500) incompatible mode.
The tape drive mode (PE or NRZI) and the data mode recorded on the tape reel did not agree.

13 (MTS500) CA OPI down.
The controller adapter (CA) operational-in (OPI) line is reset.

14 (MTS500) TCA malfunction.
A fault was detected within one of the tape controller adapters (TCAs). The two low-order bits of the substatus indicate the internal adapter interface (IAI) port number to which the malfunctioning TCA is connected.

15 (MTS500) CA EN1 error.
An unexpected interrupt occurred during operation.

16 (MTS500) CA alert - no interrupt.
A CA alert occurred while a device number was being read during a select operation and the alert was not attributed to a cyclic code error on (read) status EN1.

20 (MTS500) MTH malfunction.
The MPC has detected an apparent malfunction in a tape drive and the drive did not signal a malfunction.

21 (MTS500) multiple beginning of tape.
Additional beginning-of-tape (BOT) reflective foils were detected after a tape was moved away from load point.

53 MPC DEVICE DATA ALERT.

01 (MTS500) transmission parity alert.
A parity error was detected during execution of a special controller command.

02 (MTS500) inconsistent command.
One of the following occurred during execution of a special controller command:
1. Word count was zero for "read controller main memory," "write controller main memory," or write control store commands.
2. Execution of "read controller main memory" or "write controller main memory" referenced nonexistent memory.
3. Lock byte number specified was invalid.
4. The continue bit was zero in the IDCW for a special controller command.

03 (MTS500) checksum error.
An error occurred in the checksum used by the "write control store" command.

04 (MTS500) byte locked out.
The lock byte referenced by the "conditional write lock byte" command was nonzero.

10 (MTS500) PE-burst write error.
The MPC was unable to write the PE-burst on the tape properly.

11 (MTS500) preamble error.
An error in a PE record preamble was detected or there was apparently no data following a preamble.

12 (MTS500) T&D error.
This substatus is returned by the "device wraparound special controller" command to indicate an error byte and byte count.

20 (MTS500) multiple track error.
A data record contained errors in more than one recording track.

21 (MTS500) skew error.
Excessive skew was detected during a read or write operation in PE mode or during a write operation in NRZI mode.

22 (MTS500) postamble error.
The postamble of the PE record may have been in error. The error may have occurred in the data portion of the PE record such that a postamble appeared to be present. Also, errors may have occurred when entering the postamble so that the data appeared to continue past the data portion of the record. In either case, the postamble was not properly detected.

23 (MTS500) NRZI CCC error.
The 800-bpi, NRZI record just read contains correctable errors and may be reread.

24 (MTS500) code alert.
A character was detected that was not in the code translation tables.

40 (MTS500) marginal capstan speed.
Marginal capstan speed was detected during a write operation.

55 MPC COMMAND REJECT.

01 (MTS500) illegal procedure.
One of the following occurred:
1. The MPC was not in suspend mode when "write controller main memory" and "write control store" commands were received.
2. A special controller command did not precede an "initiate write data transfer" or "initiate read data transfer" command.

02 (MTS500. invalid logical channel number.
An invalid logical channel number was detected.

03 (MTS500) invalid suspend logical channel number.
The MPC is suspended and an IDCW was addressed to a logical channel other than the one over which the suspend controller command was received.

04 (MTS500) continue bit not set.
The first IDCW of a two-IDCW command (special controller command) did not have the continue bit set.

60 POWER OFF.

00 (MTS400) power off.
The tape controller is powered off or is not cabled to the CPI.

00 (MTS500) power off.
The MPC is powered off, is not cabled to the PSI, or has lost its personality firmware.

For disk storage, the MAJOR and SUBSTATUS fields are interpreted in the following manner. Substatuses marked with an asterisk (*) may be ORed within the same major status. Major STATUS 40 + 53 are 00 + 13 under TOITS, BECAUSE it drops left most BIT

MAJOR SUBSTATUS

40 CHANNEL READY.

00 channel ready.
If received as an initiation interrupt (I = "1") in response to a reqs or ress command, the channel and device are ready to accept a new command. If received as a termination interrupt, the last command was executed error free and the channel is ready for a new command (the device may still be busy).

0x* automatic retries.
When automatic retry is performed by the MPC, "x" is the count of retries performed.

10* device in T&D.
The device is in T&D mode.

20 (DSS191) error recovery - EDAC correct.
The MPC is attempting automatic retry, EDAC correction, and positioning offset to correct an error.

41 DEVICE BUSY.

00 file positioning.
The addressed device is busy positioning the actuator and could not accept a new command.

40 alternate channel in control.
The addressed device is busy executing a command on the alternate channel.

42 ATTENTION.

01* write inhibit.
A "write" command was issued to a device that had its write protect switch (PROTECT) in protect position.

02* seek incomplete.
The actuator mechanism of the addressed device failed to lock and/or unlock.

10 device inoperable.
The addressed device was online but did not respond correctly and requires maintenance attention.

20 device in standby.
The MPC detected a fatal error in the addressed device and continued operation would produce erroneous results.

40 device offline.
The addressed device is configured but is powered down or in offline mode.

43 DATA ALERT.

01 transfer timing alert.
The IOM did not send (write) or accept (read) data characters at a rate compatible with the transfer rate of the subsystem.

02 transmission parity alert.
A parity error was detected on a data character from the IOM during a write operation or on a data character between the MPC and the device.

04 invalid seek address.
On a "seek disk address" command, an invalid control character was detected or there were not exactly six control characters.

10* header verification failure.
The final position of the actuator did not correspond to the header address of the block being addressed by the current "seek disk address" command.

20* check character alert.
The check character generated by the MPC did not agree with the check character recorded on the disk.

40* data compare alert.
The data recorded on the disk did not compare with the data from the IOM during a "compare and verify" command.

44 END OF FILE.

00 good track detected.
A good track was detected at the specified sector address when a defective or alternate track was expected.

01* last consecutive block.
The last consecutive block available to the present actuator position was reached and the current command is incomplete.

02* sector count limit.
The sector count limit specified in the previous seek disk address command was reached.

04 defective track - alternate track assigned.
When an alternate track is assigned a read or write operation was attempted to a defective track or an overflow was detected to or from an alternate track.

10 defective track - no alternate track assigned.
An alternate track is not assigned when a read or write operation was attempted to a defective track or an overflow was detected to or from an alternate track.

20 alternate track detected.
A read or write operation was attempted to an alternate track when the track condition indicators from the previous "seek disk address" command did not indicate an alternate track operation.

45 COMMAND REJECTED.

01 invalid command.
The channel was unable to recognize the device command code in the PCW or IDCW.

02 invalid device code.
An invalid device code was received from the IOM or no device with the given code is configured to the subsystem.

04 parity alert on IDCW.
The MPC detected a parity error on the device or command code from the IOM.

10 invalid command sequence.
A data transfer command without a prior "seek disk address" command was received or the "data transfer" command contained a device code different from that given in the "seek disk address" command.

50 CHANNEL BUSY.

00 busy.
The command was accepted but execution will be delayed until current command sequences are complete because the command requires the entire subsystem.

52 MPC DEVICE ATTENTION.

01 configuration error.
The personality firmware loaded into the MPC does not agree with the settings of the MPC configuration switches.

02 multiple devices.
The MPC has detected at least two devices with the same logical ID number.

03 device number error.
The MPC has detected at least one device with a logical ID number outside the allowed range of ID numbers.

13 CA OPI down.
The controller adapter (CA) operational-in (OPI) line is reset.

14 alert EN1 unexpected interrupt.
The CA detected an abnormal condition during operation.

15 CA EN1 error.
An unexpected interrupt occurred during operation.

16 CA alert - no interrupt.
A CA alert occurred while a device number was being read during a select operation and was not attributed to a cyclic code error on (read) status EN1.

Under T.O's

13 53 MPC DEVICE DATA ALERT.

01 transmission parity.
A transmission parity error was detected during execution of a special controller command.

02 inconsistent command.
One of the following occurred during the execution of a special controller command:
1. The word count was zero for "read controller main memory", "write controller main memory", or "write control store" commands.
2. The execution of "read controller main memory" or "write controller main memory" referenced nonexistent memory.
3. The lock byte number specified was invalid.
4. The continue bit was zero in the IDCW for a special controller command.

03 checksum error.
An error occurred in the checksum used by the "write control store" command.

04 byte locked out.
The lock byte referenced by the "conditional write lock byte" command was nonzero.

16 (DSS190, DSS191) EDAC parity error.
An MPC hardware error was detected during EDAC generation.

21 sector size error.
The data field length read from the track was not as specified for the read function.

22 nonstandard sector size.
An attempt was made to read a sector that was not standard size.

23 (DSS190, DSS191) search alert on first search.
A double index was encountered on the first search of a "seek disk address" command and the MPC could not find a sector number.

24 (DSS190, DSS191) cyclic code error (not first search).
The MPC encountered a cyclic code error in the count field during a search that followed the initial search.

25 (DSS190, DSS191) search error (not first search).
The sector number did not compare on the second or subsequent search or the MPC encountered no count field on the track after head switching.

26 (DSS190, DSS191) sync byte error.
The MPC could not find the proper sync byte.

27 (DSS190, DSS191) error in automatic alternate track processing.
An error occurred in going to, processing, or returning from an alternate track.

31 (DSS190) EDAC correction - last sector.
An error was detected in the last sector transmitted, but the error was corrected and the transmission completed.

32 (DSS190) EDAC correction - not last sector.
An EDAC error was detected in a sector other than the last sector and was corrected. A new operation was generated by the MPC for the remaining sectors.

33 (DSS190) EDAC correction - block count limit.
An EDAC was detected and corrected on the last sector requested.

34 (DSS190) uncorrectable EDAC error.
An EDAC error was detected and found to be uncorrectable.

35 (DSS190) EDAC correction - short block.
One of the following conditions occurred:

1. If an EDAC error was reported after the DCW exhausted (i.e., within a sector but outside that part of the sector transmitted), a CHANNEL READY status is returned.
2. If an EDAC error was reported before the DCW exhausted, the EDAC correction for substatus 31(above) is applied.
3. If the DCW exhausted at the end of a sector and an EDAC error is in the next sector, the subsystem recognizes that the DCW string is modulo 64 and returns a CHANNEL READY status. This occurs when the DCW exhausts on a sector boundary because the hardware, in terminating the operation, must read the sector even though the DCW exhausted and the EDAC error was encountered in the second sector.

45 MPC COMMAND REJECT.

01 illegal procedure.
One of the following occurred:
1. The MPC was not in suspend mode when "write controller main memory" and "write control store" commands were received.
2. A special controller command did not precede an "initiate write data transfer" or "initiate read data transfer" command.

02 invalid logical channel number.
An invalid logical channel number was detected.

03 invalid suspend command.
The MPC is suspended and an IDCW was addressed to a logical channel other than the one over which the "suspend controller" command was received.

04 continue bit not set.
The first IDCW of a two-IDCW command (special controller command) did not have the continue bit set.

60 POWER OFF.

00 power off.
The MPC is powered off, is not cabled to the PSI, or has lost its personality firmware.

SYSTEM CONSOLES

For system consoles, the MAJOR and SUBSTATUS fields are interpreted in the following manner. Substatuses marked with an asterisk (*) may be ORed within the same major status.

MAJOR SUBSTATUS

40 CHANNEL READY.

00 channel ready.
If received as an initiation interrupt (I ="1") in response to a reqs or ress command, the console is ready to accept a new command. If received as a termination interrupt, the last command was executed error free and the console is ready to accept a new command.

42 ATTENTION.

00 attention.
The console is unable to accept a command because of some inoperable condition.

43 DATA ALERT.

01 transfer timing alert.
The IOM did not receive (read) or send (write) data characters at a rate compatible with the transfer rate of the console.

02* transmission parity alert.
Incorrect parity was detected on a data character received from the IOM. This error can occur only during a write operation.

04 operator input error.
The operator has pressed the OPERATOR INPUT ERROR key on the console.

10 operator distracted.
An interval of 30 seconds has elapsed without input during a read operation.

20* incorrect format.
An escape character is followed by a invalid character in a message received from the IOM or a control character ("?" = 17(8), " " = 77(8)) is not preceded by the proper number of escape characters.

40 message length alert.
The operator has entered more characters than were specified by the DCWs referenced by the "read" command.

45 COMMAND REJECTED.

01 invalid command.
The channel was unable to recognize the device command code in the PCW or IDCW.

03 command parity error.
A parity error was detected on the device command received from the IOM.

50 CHANNEL BUSY.

00 channel busy.

60 POWER OFF.

00 power off.
The console is powered off or is not cabled to the IOM channel.

MPC EXTENDED STATUS

The microprogrammed peripheral controller (MPC) maintains detailed, extended status for each device connected. This extended status is obtainable with the "read status register" (RSR) special controller command and is transmitted as a series of 8-bit bytes in binary data mode.

Multics currently types the hexadecimal representation of these extended status bytes on the console for each disk error.

DSU181 Extended Status

(There is no include file for the declaration of this data.)

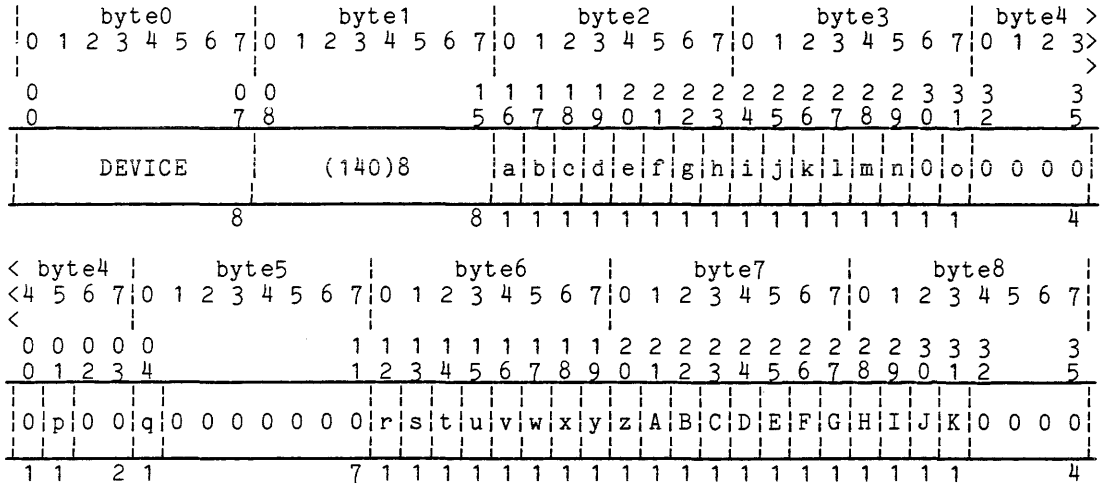


Figure C-1. DSS181 Extended Status

Legend:

Field	Meaning
DEVICE	the logical ID number of the drive.
(140)8	the octal value of the hexadecimal code for DSS181.
a	device reserved.
b	device seized.
c	device in standby.
d	positioner busy.
e	DLI fault.
f	device protected.
g	device fault.
h	device in T&D mode.
i	command parity error.
j	no or multiple command decode.
k	invalid command.
l	invalid command sequence.
m	state violation.
n	protection violation.
o	data parity error.
p	spindle speed loss.
q	seek incomplete.
r	erase current unsafe.
s	DC write unsafe.
t	AC write unsafe.
u	heads unsafe.
v	erase gate and busy.
w	write gate and busy.
x	write gate and no erase current.
y	voltage unsafe.

Field	Meaning
z	brush at stop.
A	pack on.
B	lid on.
C	index block in.
D	attention latch.
E	heads flying.
F	zero speed.
G	online.
H	positioner overtemperature.
I	positioner overvelocity.
J	position out of limits.
K	positioner voltage out of limits.

DSU190A Extended Status

CAUTION: The extended status formats for DSU190A and DSU190B devices are different. BE SURE YOU REFER TO THE FORMAT FOR YOUR DEVICES.

(There is no include file for the declaration of this data.)

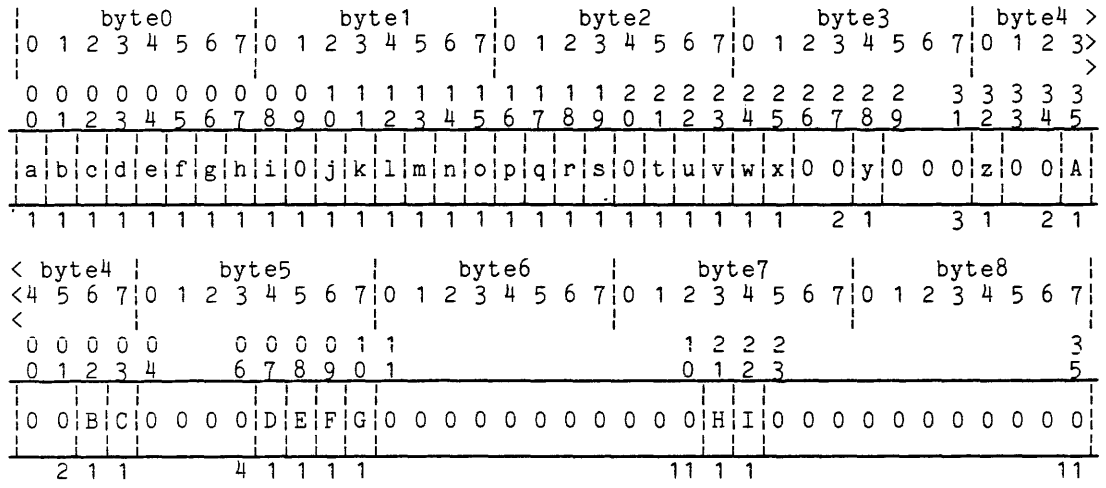


Figure C-2. DSS190A Extended Status

Legend:

Field	Meaning
a	device reserved.
b	device seized.
c	device in standby.
d	positioner busy.
e	DLI fault.
f	device protected.
g	device fault.
h	device in T&D mode.

Field	Meaning
i	command parity error.
j	invalid command.
k	invalid command sequence.
l	state violation.
m	protection violation.
n	transfer timing error.
o	data parity error.
p	loss of write current.
q	write current without write command.
r	loss of AC write current.
s	no or multiple head selection.
t	spindle speed loss.
u	overtemperature.
v	loss of voltage.
w	seek incomplete.
x	positioner overtravel.
y	RPS error.
z	fine servo.
A	brush cycle incomplete.
B	forward set.
C	reverse set.
D	heads retracted.
E	positioner offset.
F	read clock offset.
G	write and read.
H	low air flow.
I	read amplitude low.

DSU190B Extended Status

CAUTION: The extended status formats for DSU190A and DSU190B devices are different. BE SURE YOU REFER TO THE FORMAT FOR YOUR DEVICES.

(There is no include file for the declaration of this data.)

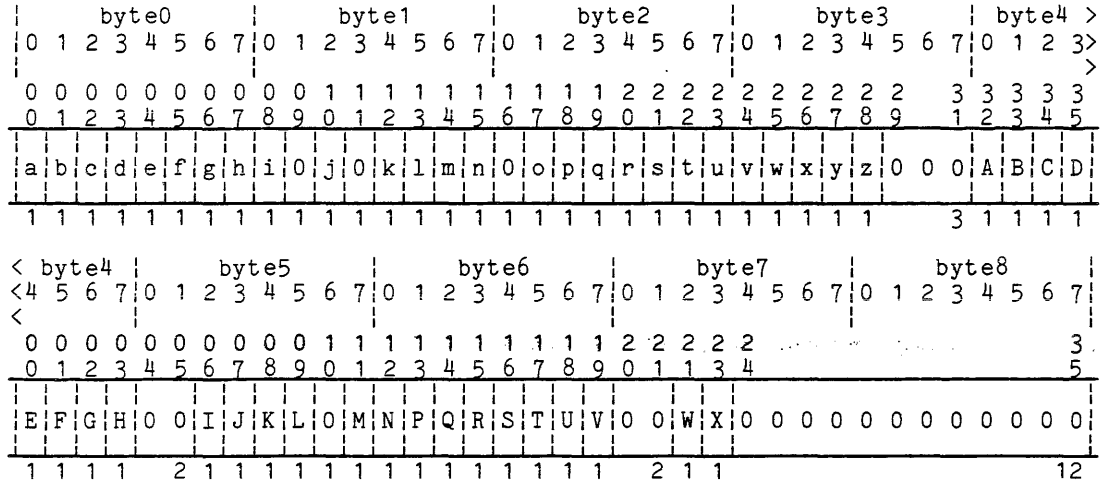


Figure C-3. DSS190B Extended Status

Legend:

Field	Meaning
a	device reserved.
b	device seized.
c	device in standby.
d	positioner busy.
e	DLI fault.
f	device protected.
g	device fault.
h	device in T&D mode.
i	command parity error.
j	invalid command.
k	state violation.
l	protection violation.
m	transfer timing error.
n	data parity error.
o	write current without write command.
p	loss of write current.
q	no or multiple head selection.
r	incomplete start cycle.
s	spindle speed loss.
t	positioner overtemperature.
u	DC power loss.
v	seek incomplete.
w	positioner overtravel.
x	positioner internal fault.
y	positioner sense fault.
z	RPS fault.
A	positioner overspeed.
B	invalid cylinder address.
C	loss of index.
D	emergency retract occurred.
E	loss of velocity.
F	positioner off track.
G	invalid head address.
H	positioner offset.
I	read or write counter error.
J	write precompensation fault.
K	KFK decoder fault.
L	read command timing fault.
M	read or write clock fault.
N	loss of read signal.
P	incorrect write current.
Q	loss of position signal.
R	loss of positioner current.
S	loss of power amplifier input.
T	write fault sense.
U	position motor/pack overtemperature.
V	loss of blower.
W	clogged coarse filter.
X	clogged fine filter.

MSU0451 Extended Status

(There is no include file for the declaration of this data.)

byte0								byte1								byte2								byte3								byte4			
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2				3	3	3	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8			1	2	3	4	5
a	b	c	d	e	f	g	h	i	0	j	k	l	m	n	o	p	q	r	s	0	t	0	u	v	w	0	x	0	0	0	0	y	z	0	A

byte4				byte5					byte6								byte7								byte8							
4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	0	0	0			0	0	0	1	1	1																							3
0	1	2	3	4			7	8	9	0	1	2																							5
0	B	C	D	0	0	0	0	E	F	G	H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure C-4. MSU0451 Extended Status

Legend:

Field	Meaning
a	device reserved.
b	device seized.
c	device in standby.
d	positioner busy.
e	DLI fault.
f	device protected.
g	device failure.
h	device in diagnostic mode.
i	command parity error.
j	invalid command.
k	invalid command sequence.
l	state violation.
m	protect violation.
n	transfer timing error.
o	data parity error.
p	write command without write current.
q	write current without write command.
r	loss of AC write current.
s	no or multiple head selection.
t	spindle speed loss.
u	loss of voltage.
v	seek incomplete.
w	positioner overtravel.
x	rotational position sensing fault.

Field	Meaning
y	fine servo status.
z	tester address error.
A	first seek interlock cycle incomplete.
B	restricted air flow.
C	forward FF set.
D	reverse FF set.
E	heads retracted.
F	positioner offset.
G	read clock offset.
H	write and read.

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE: SERIES 60 (LEVEL 68)
MULTICS HARDWARE AND SOFTWARE FORMATS
PROGRAM LOGIC MANUAL
ADDENDUM A

ORDER NO. AN87-00A

DATED MARCH 1980

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. ☐

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

_____

CUT ALONG LINE

FOLD ALONG LINE

PLEASE FOLD AND TAPE—
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

FOLD ALONG LINE

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

16028, 2.5C676, Printed in U.S.A.

AN87, Rev. 0