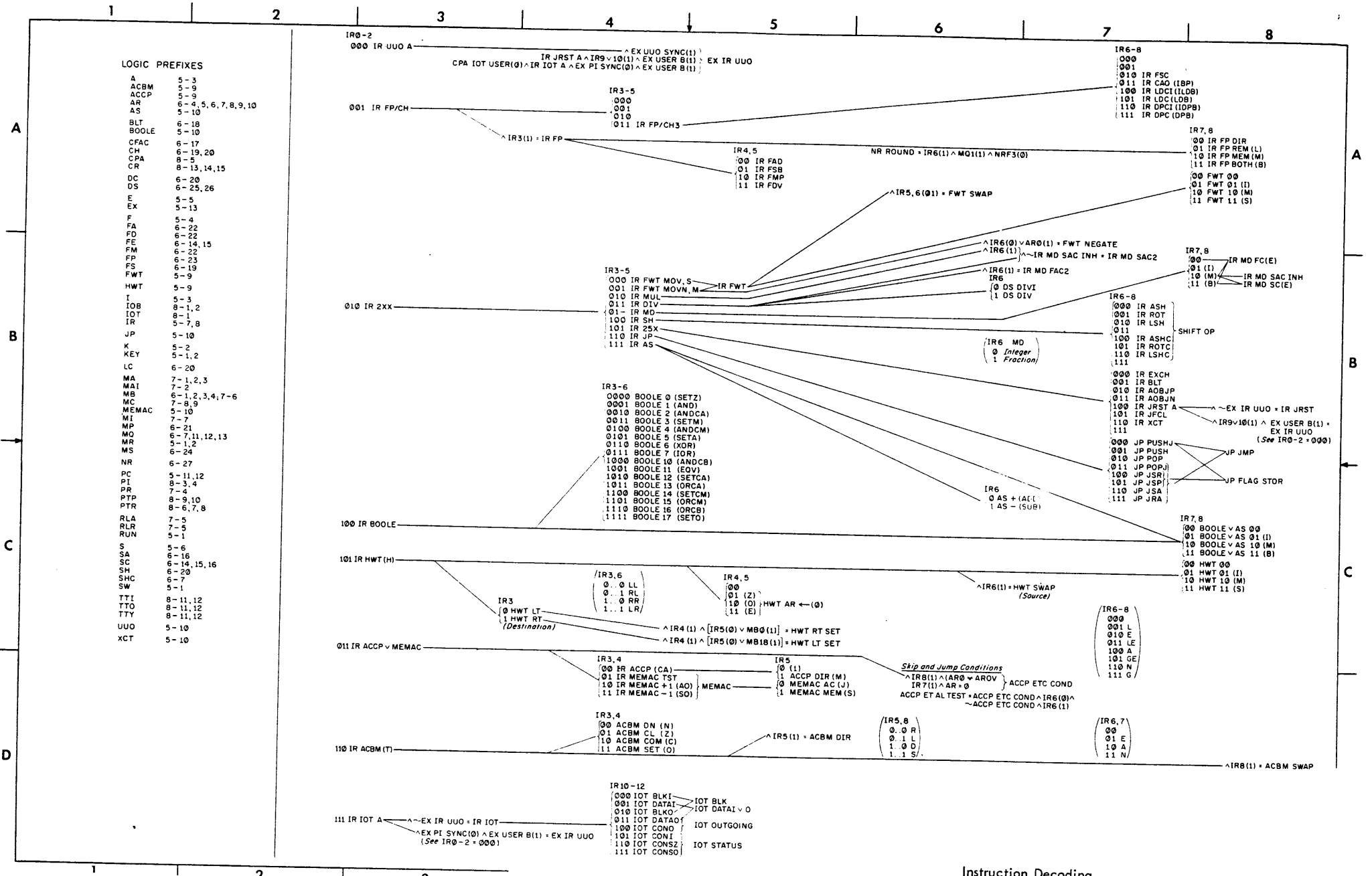


System Block Diagram
(SD-D-166-0-BD)



LOGIC PREFIXES

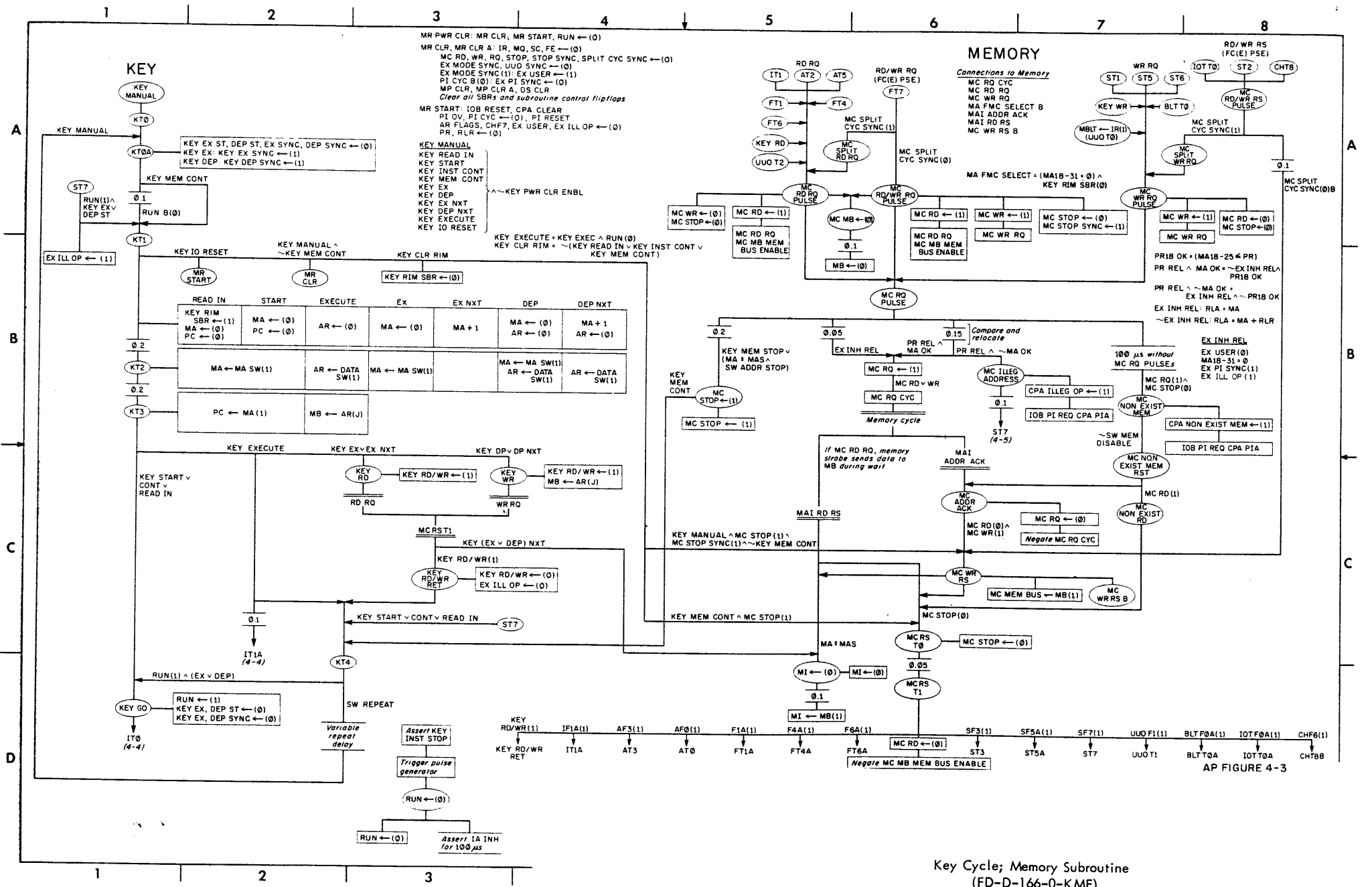
A	A	5-3
	ACBM	5-9
	ACCP	5-9
	AR	6-4, 5, 6, 7, 8, 9, 10
	AS	5-10
	BLT	6-18
	BOOLE	5-10
	CFAC	6-17
	CH	6-19, 20
	CPA	8-5
	CR	8-13, 14, 15
	DC	6-20
	DS	6-25, 26
	E	5-5
	EX	5-13
	F	5-4
	FA	6-22
	FD	6-22
	FE	6-14, 15
	FM	6-22
	FP	6-23
	FS	6-19
	FWT	5-9
	HWT	5-9
	I	5-3
	IOB	8-1, 2
	IOT	9-1
	IR	9-7, 8
	JP	5-10
	K	5-2
	KEY	5-1, 2
	LC	6-20
	MA	7-1, 2, 3
	MAI	7-2
	MB	6-1, 2, 3, 4, 7-6
	MC	7-8, 9
	MEMAC	5-10
	MI	7-7
	MP	6-21
	MQ	6-7, 11, 12, 13
	MR	5-1, 2
	MS	6-24
	NR	6-27
	PC	5-11, 12
	PI	8-3, 4
	PR	7-4
	PTP	8-9, 10
	PTR	8-6, 7, 8
	RLA	7-5
	RLR	7-5
	RUN	5-1
	S	5-6
	SA	6-16
	SC	6-14, 15, 16
	SH	6-20
	SHC	6-7
	SW	5-1
	TTI	8-11, 12
	TTO	8-11, 12
	TTY	8-11, 12
	UOO	5-10
	XCT	5-10

A

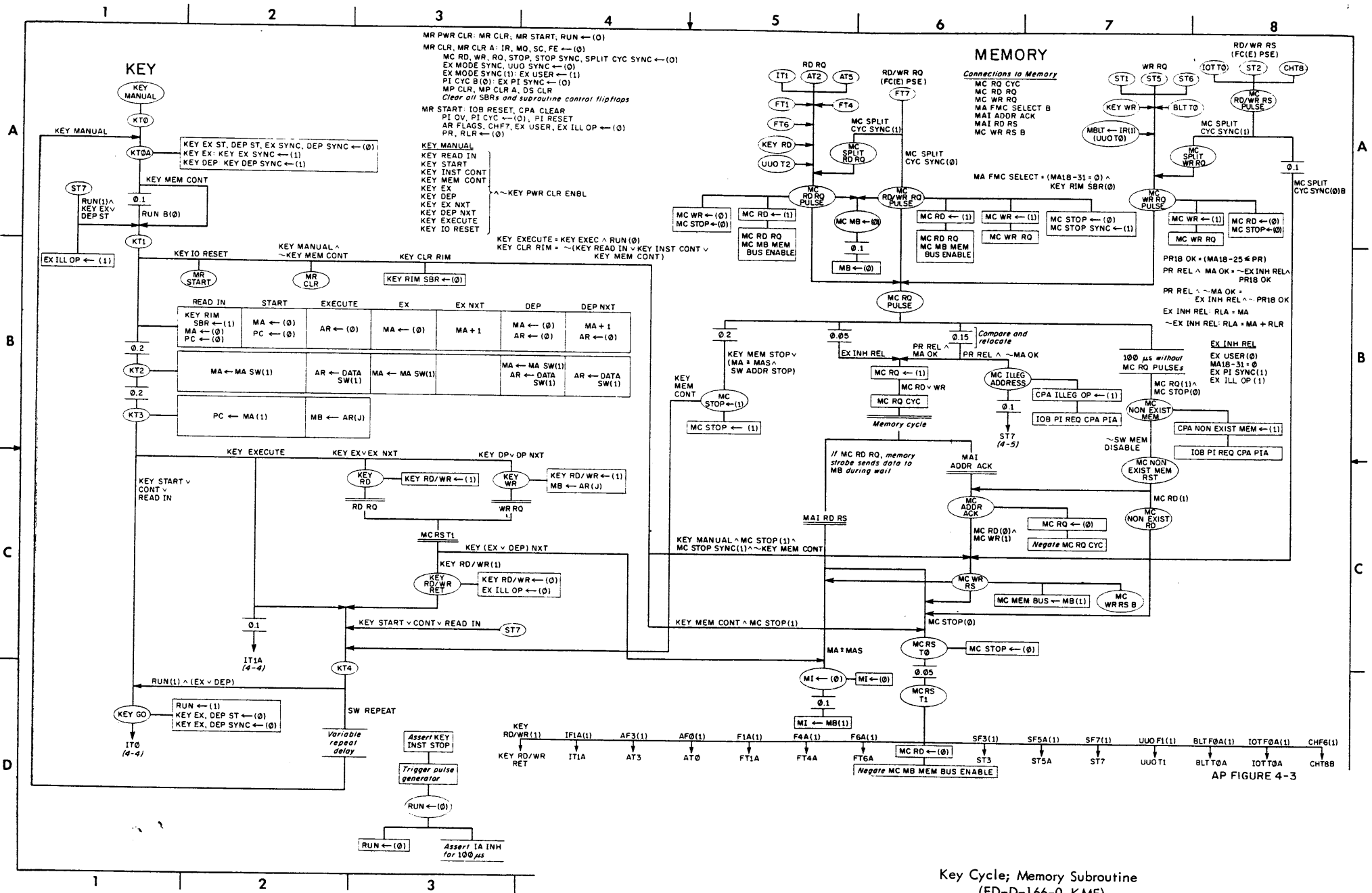
B

C

D



Key Cycle; Memory Subroutine
(FD-D-166-0-KMF)



MR PWR CLR: MR CLR; MR START; RUN ← (0)
 MR CLR, MR CLR A: IR, MQ, SC, FE ← (0)
 MC RD, WR, RD, STOP, STOP SYNC, SPLIT CYC SYNC ← (0)
 EX MODE SYNC, UUD SYNC ← (0)
 EX MODE SYNC (1): EX USER ← (1)
 PI CYC B (0): EX PI SYNC ← (0)
 MP CLR, MP CLR A, DS CLR
Clear all SBRs and subroutine control flipflops
 MR START: IOB RESET, CPA CLEAR
 PI DV, PI CYC ← (0), PI RESET
 AR FLAGS, CHFT, EX USER, EX ILL OP ← (0)
 PR, RLR ← (0)

MEMORY

Connections to Memory
 MC RO CYC
 MC RD RO
 MC WR RO
 MA FMC SELECT B
 MAI ADDR ACK
 MAI RD RS
 MC WR RS B

$MA FMC SELECT = (MA18-31 = 0) \wedge$
 $KEY RIM SBR(0)$

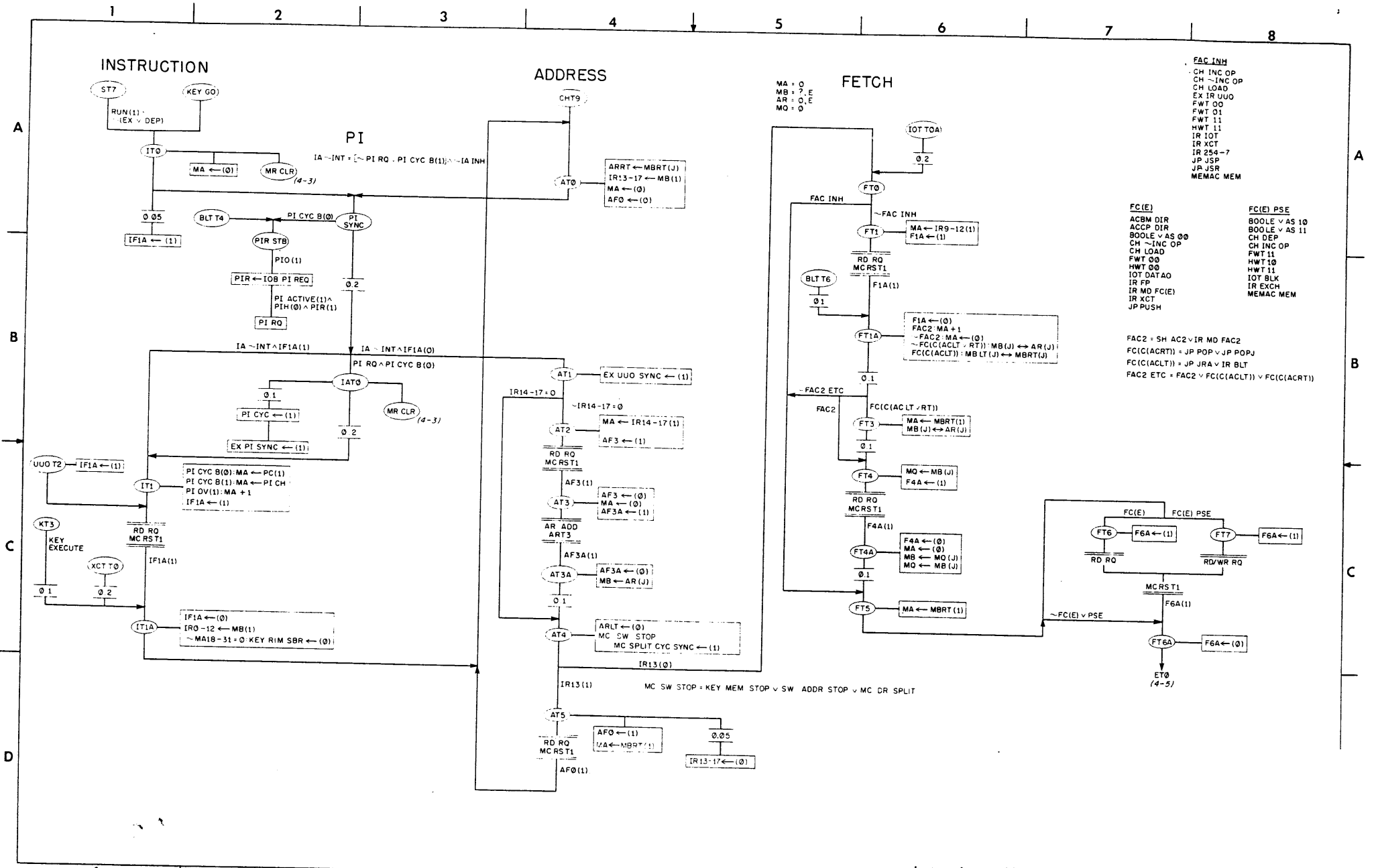
$PR18 OK = (MA18-25 \neq PR)$
 $PR REL \wedge MA OK = \sim EX INH REL \wedge PR18 OK$
 $PR REL = \sim MA OK = \sim PR18 OK$
 $EX INH REL: RLA = MA$
 $\sim EX INH REL: RLA = MA + RLR$

Compare and relocate
 $EX INH REL$
 $EX USER(0)$
 $MA18-31 = 0$
 $EX PI SYNC(1)$
 $EX ILL OP(1)$
 100 μs without MC RO PULSES
 EX INH REL
 EX USER(0)
 MA18-31 = 0
 EX PI SYNC(1)
 EX ILL OP(1)

Memory cycle
 MAI ADDR ACK
 MAI RD RS
 MC ADDR ACK
 MC RD(0) \wedge MC WR(1)
 Negate MC RO CYC
 MC WR RS
 MC MEM BUS ← MB(1)
 MC WR RS B
 MC RD(1)
 MC NON EXIST RD
 MC NON EXIST MEM
 MC NON EXIST MEM
 IOB PI REQ CPA PIA
 CPA NON EXIST MEM ← (1)
 CPA ILLEG OP ← (1)
 MC ILLEG ADDRESS
 MC RO(1) \wedge MC STOP(0)
 IOB PI REQ CPA PIA
 SW MEM DISABLE
 MC STOP(0)

MAI MAS
 MI ← (0) MI ← (0)
 MI ← MB(1)
 MCRS T0
 MCRS T1
 MC STOP ← (0)
 MC STOP(0)
 MC RD ← (0)
 Negate MC MB MEM BUS ENABLE
 SF3(1) SF5A(1) SF7(1) UUDF(1) BLTF0A(1) IOTF0A(1) CHF6(1)
 ST3 ST5A ST7 UUDT1 BLT0A IOT0A CHT8B

Key Cycle; Memory Subroutine (FD-D-166-0-KMF)



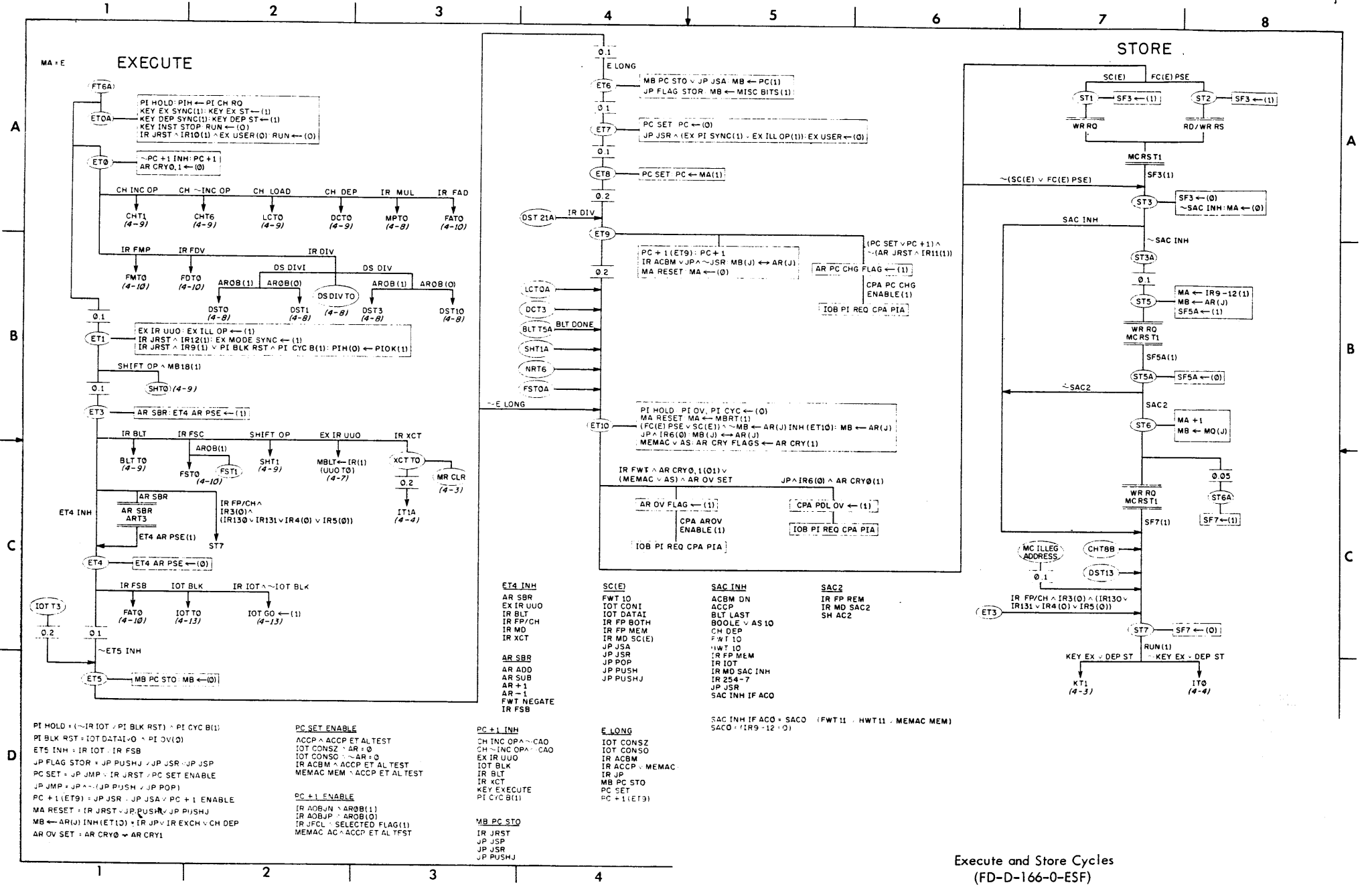
FAC INH
 CH INC OP
 CH ~INC OP
 CH LOAD
 EX IR UO
 FWT 00
 FWT 01
 FWT 11
 HWT 11
 IR IOT
 IR XCT
 IR 254-7
 JP JSP
 JP JSR
 MEMAC MEM

FC(E)
 ACBM DIR
 ACCP DIR
 BOOLE ^ AS 00
 CH ~INC OP
 CH LOAD
 FWT 00
 HWT 00
 IOT DATAQ
 IR FP
 IR MD FC(E)
 IR XOT
 JP PUSH

FC(E) PSE
 BOOLE ^ AS 10
 BOOLE ^ AS 11
 CH DEP
 CH INC OP
 FWT 11
 HWT 10
 HWT 11
 IOT BLK
 IR EXCH
 MEMAC MEM

FAC2 = SH AC2 ^ IR MD FAC2
 FC(C(ACRT)) = JP POP ^ JP POPJ
 FC(C(ACLT)) = JP JRA ^ IR BLT
 FAC2 ETC = FAC2 ^ FC(C(ACLT)) ^ FC(C(ACRT))

Instruction, Address and Fetch Cycles
 (FD-D-166-0-IAFF)



MA + E

EXECUTE

STORE

A

A

B

B

C

C

D

PI HOLD ← (~IR IOT / PI BLK RST) ∨ PI CYC B(1)
 PI BLK RST ← IOT DATA(0) ∨ PI OV(0)
 ETS INH ← IR IOT ∨ IR FSB
 JP FLAG STOR ∨ JP PUSH ∨ JP JSR ∨ JP JSP
 PC SET ← JP JMP ∨ IR JRST ∨ PC SET ENABLE
 JP JMP ∨ JP ∼ (JP PUSH ∨ JP POP)
 PC + 1 (ET9) ← JP JSR ∨ JP JSA ∨ PC + 1 ENABLE
 MA RESET ← IR JRST ∨ JP PUSH ∨ JP PUSHJ
 MB ← AR(J) INH (ET10) ∨ IR JP ∨ IR EXCH ∨ CH DEP
 AR OV SET ← AR CRYO ∨ AR CRY1

PC SET ENABLE
 ACCP ∨ ACCP ET AL TEST
 IOT CONSZ ∨ AR = 0
 IOT CONSO ∨ AR = 0
 IR ACBM ∨ ACCP ET AL TEST
 MEMAC MEM ∨ ACCP ET AL TEST

PC + 1 ENABLE
 IR AOBUN ∨ AROB(1)
 IR AOBUP ∨ AROB(0)
 IR JFCL ∨ SELECTED FLAG(1)
 MEMAC AC ∨ ACCP ET AL TEST

PC + 1 INH
 CH INC OPA ∨ CAO
 CH ∼ INC OPA ∨ CAO
 EX IR UOO
 IOT BLK
 IOT BLT
 IR ACT
 KEY EXECUTE
 PI CYC B(1)

MB PC STO
 IR JRST
 JP JSP
 JP JSR
 JP PUSHJ

E LONG
 IOT CONSZ
 IOT CONSO
 IR ACBM
 IR ACCP ∨ MEMAC
 IR JP
 IR ACT
 MB PC STO
 PC SET
 PC + 1 (ET9)

SAC INH IF ACQ ∨ SAC0 (FWT 11 ∨ HWT 11 ∨ MEMAC MEM)
 SAC0 ∨ IR9 - 12 ∨ 0

Execute and Store Cycles
(FD-D-166-0-ESF)

	1	2	3	4	5	6	7	8					
	HALF WORD TRANSFER 500-577			FULL WORD TRANSFER 200-217		EXCH 250		ARITHMETIC COMPARE 300-377					
	LOGICAL COMPARE 600-677			AOBJP 252		AOBJN 253							
	<i>IR</i> = 101 VXX YZZ <i>IR HWT</i> = IR0-2(101) <i>W</i> specifies destination half HWT LT = IR HWT IR3(0) HWT RT = IR HWT IR3(1) <i>XY</i> specifies action on other half - <i>do nothing, zero one, extend</i> HWT AR = (0) = IR HWT IR4(1) IR5(1) HWT RT SET = HWT LT IR4(1) HWT LT SET = HWT RT IR4(1) <i>Y</i> specifies source half HWT SWAP = IR HWT IR6(1) <i>ZZ</i> specifies mode			<i>IR</i> = 010 O0Y YZZ <i>IR 2XX</i> = IR0-2(010) <i>YY</i> specifies instruction - MOVE, MOVN, MOVW, MOVV IR MOV S = IR 2XX IR3-5(000) IR MOVN M = IR 2XX IR3-5(001) IR MOVW M = IR 2XX IR3-5(002) IR MOVV M = IR 2XX IR3-5(003) IR FWT = IR MOV S = IR MOVN M FWT SWAP = IR FWT IR5,6(01) FWT NEGATE = IR MOVN M IR6(0) V ARO(1) <i>ZZ</i> specifies mode		<i>IR</i> = 010 101 000 <i>IR 2XX</i> = IR0-2(010) <i>IR 25X</i> = IR 2XX IR3-5(101) <i>IR EXCH</i> = IR 25X IR6-8(000) AC → C(E)		<i>IR</i> = 011 VVV XYZ <i>IR ACCP</i> = MEMAC + IR0-2(011) <i>VV</i> specifies instruction type (IR ACCP = MEMAC) ^ IR3,4 MEMAC = MEMAC TST V MEMAC +1 V MEMAC -1 <i>X</i> specifies whether or not action is on condition YZ <i>Y</i> specifies condition "equals" on condition YZ <i>Z</i> specifies condition "less than" <i>XYZ</i> together define relation R		<i>IR</i> = 110 VVV XYZ <i>IR ACBM</i> = IR0-2(110) <i>VV</i> specifies action on masked bits 00 = ACBM DN IR ACBM = (0) = ACBM CL IR3,4(1) = ACBM COM 11 = ACBM SET <i>W</i> specifies source of mask - immediate or memory ACBM DIR = IR ACBM IR5(1) <i>X</i> specifies whether or not action is on condition YZ, "equals zero" <i>XY</i> together define relation R		<i>IR</i> = 010 101 01X <i>IR 2XX</i> = IR0-2(010) <i>IR 25X</i> = IR 2XX IR3-5(101) <i>X</i> specifies instruction IR AOBJP = IR 25X IR6-8(010) IR AOBJN = IR 25X IR6-8(011) AC + 1000001 → AC AOBJP ^ (AC = 0) : E → PC AOBJN ^ (AC = 0) : E → PC IR AOBJP ^ AROB(0) : PC SET IR AOBJN ^ AROB(1) : PC SET	
	<i>00</i> = HWT 00 /C(E), AC → AC <i>01</i> = HWT 01 /((0,E),AC) → AC <i>10</i> = HWT 10 /AC,C(E) → E <i>11</i> = HWT 11 /C(E) → E			<i>00</i> = FWT 00 /C(E) → AC <i>01</i> = FWT 01 /C(E) → AC <i>10</i> = FWT 10 /C(E) → E <i>11</i> = FWT 11 /C(E) → E		<i>000</i> = Never <i>001</i> = <i>010</i> = <i>011</i> = <i>100</i> = Always <i>101</i> = <i>110</i> = <i>111</i> =		<i>00</i> = Never <i>01</i> = <i>10</i> = Always <i>11</i> =					
Initial Registers	HWT 00: AR = AC MB = C(E) HWT 01: AR = AC MB = 0,E HWT 10: AR = AC MB = C(E) HWT 11: AR = 0,E MB = C(E) MQ = 0			FWT 00: AR = 0,E MB = C(E) FWT 01: AR = 0,E MB = ?E FWT 10: AR = AC MB = 0,E FWT 11: AR = 0,E MB = C(E) MQ = 0		AR = AC MB = C(E) MQ = 0		Determination of Skip or Jump Condition ACCP ETC COND = (IR7(1) ^ (AR=0)) V ((IR ACCP V MEMAC) ^ IR8(1) ^ (AR0 → AR OV)) ACCP ET AL TEST = ACCP ETC COND ^ IR6 ARO = AR OV ^ ARO → (AR OV SET ^ ~MEMAC) AR OV SET = AR CRYO → AR CRY1					
Initial Gates	HWT 00: FC(E) HWT 10: FC(E) PSE HWT 11: FAC INH, FC(E) PSE			HWT 00: FAC INH, FC(E) HWT 01: FAC INH HWT 11: FAC INH, FC(E) PSE FWT NEGATE: ET4 INH, AR SBR		FC(E) PSE MB → AR(J) INH(ET10)		MEMAC 320-377 <i>W</i> specifies whether AC or C(E) compared with zero, i.e. whether action is jump or skip ACCP AC = MEMAC ^ IR5(0) MEMAC MEM = MEMAC ^ IR5(1) MEMAC AC AC R 0: E → PC MEMAC ±1: AC ±1 → AC ACCP ET AL TEST: PC SET					
ET0A	HWT 10: MB(J) ↔ AR(J) HWT 11: AR ← MB(J)			FWT 00: 11: AR ← MB(J) FWT 01: 10: MB ← AR(J)		MB(J) ↔ AR(J)		ACCP 300-317 <i>W</i> specifies whether AC compared with E or C(E) ACCP DIR = ACCP ^ IR5(1) IR5(0) ^ AC R(0, E) PC +1 IR5(1) ^ (AC R C(E)) PC +1					
ET1	HWT SWAP: MBLT(J) ↔ MBRT(J) HWT AR ← (0): AR ← (0)			FWT SWAP: MBLT(J) ↔ MBRT(J)				ACBM R (Mask ^ AC) : PC +1 / (Mask, AC) → AC					
ET3				FWT NEGATE: ET4 AR PSE ← (1) AR NEGATE ART3									
ET4	HWT LT SET AR LT COM HWT RT SET AR RT COM HWT LT AR ← MB LT(J) HWT RT AR ← MB RT(J)			ET4 AR PSE ← (0) FWT SWAP: MB(J) ↔ AR(J)									
ET5													
ET10	HWT 10: 11: MB ← AR(J)			AR CRYO: 1(01) AR OV FLAG ← (1) FWT 10: 11: MB ← AR(J)									
Final Gates	HWT 10: FC(E) PSE, SAC INH HWT 11: FC(E) PSE HWT 11: SAC0: SAC INH SAC0 = (IR9-12 = 0)			FWT 10: SC(E), SAC INH FWT 11: FC(E) PSE FWT 11: SAC0: SAC INH SAC0 = (IR9-12 = 0)		FC(E) PSE							
						Initial Registers AR = AC MB = 0,E MQ = 0		AR = AC ~ACCP DIR: MB = 0,E ACCP DIR: MB = C(E) MQ = 0					
						Initial Gates MEMAC ±1: ET4 INH AR SBR E LONG		ACCP DIR: FC(E) ET4 INH AR SBR E LONG					
ET0A						AR ← MB(J)		ACBM SWAP: MBLT(J) ↔ MBRT(J)					
ET1								MB ← AR(0) ACBM COM: AR ← MB (-) ACBM SET: AR ← MB(1)					
ET3				MEMAC +1: ET4 AR PSE ← (1) AR +1 ART3 MEMAC -1: ET4 AR PSE ← (1) AR -1 ART3		ET4 AR PSE ← (1) AR SUB ART3		ET4 AR PSE ← (1) AR +1 LTRT ART3					
ET4				ET4 AR PSE ← (0)		ET4 AR PSE ← (0)		ET4 AR PSE ← (0)					
ET5													
ET6								AR COM					
ET7								ACBM CL MB ← AR(0)					
ET8								AR COM					
ET9	PC SET: PC ← MA(1)			PC SET: PC ← MA(1)				PC SET: PC ← MA(1)					
ET10	PC SET: AR PC CHG FLAG ← (1)			ACCP ET AL TEST: PC +1 AR PC CHG FLAG ← (1)		ACCP ET AL TEST: PC +1 AR PC CHG FLAG ← (1)		MB(J) ↔ AR(J) ACCP ET AL TEST: PC +1 AR PC CHG FLAG ← (1)					
						AR OV SET: AR OV FLAG ← (1) AR CRY FLAGS ← AR CRY(1) MB ← AR(J)							
						FC(E) PSE SAC0: SAC INH		SAC INH					

JUMP AND PUSHDOWN

260-267

IR = 010 110 XXX
 IR 2XX = IR0-2 (010)
 IR JP = IR 2XX ^ IR3-5 (110)
 XXX specifies instruction

000 = JP PUSHJ
 001 = JP PUSH
 010 = JP POP
 011 = JP POPJ
 100 = JP JSR
 101 = JP JSR
 110 = JP JSA
 111 = JP JRA

JP JMP = IR JP ^ (JP PUSH ^ JP POP)
 JP FLAG STOR = JP PUSHJ ^ JP JSR ^ JP JSR
 ~IR IOT ^ PI CYC B(1): PI HOLD

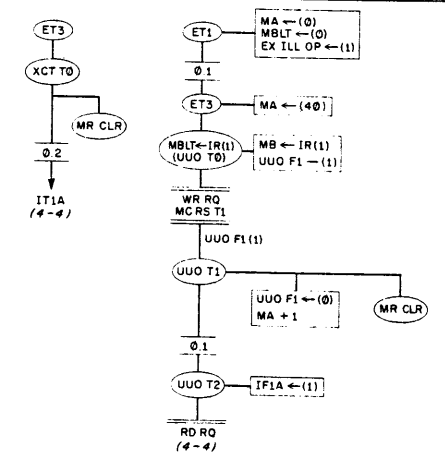
JRST
 254
 IR = 010 101 100
 IR 2XX = IR0-2 (010)
 IR 25X = IR2XX ^ IR3-5 (101)
 IR JRST A = IR 25X ^ IR 6-8 (100)
 IR JRST = IR JRST A ^ ~EX IR UO

JFCL
 255
 IR = 010 101 101
 IR 2XX = IR0-2 (010)
 IR 25X = IR 2XX ^ IR3-5 (101)
 IR JFCL = IR 25X ^ IR 6-8 (010)

XCT
 256
 IR = 010 101 110
 IR 2XX = IR0-2 (010)
 IR 25X = IR 2XX ^ IR3-5 (101)
 IR XCT = IR 25X ^ IR6-8 (110)

UO
 000-077
 IR = 000 XXX XXX
 IR UO A = IR0-2 (000)
 EX IR UO =
 IR UO A ^ EX UO SYNC(1) ^ IR9 ^ 10(1) ^
 IR JRST A ^ EX USER B(1) ^ IR9 ^ 10(1) ^
 IR IOT A ^ EX USER B(1) ^ EX PI SYNC(0) ^ CPA IOT USER(0)

	PUSHJ 260	PUSH 261	POP 262	POPJ 263	JSR 264	JSP 265	JSA 266	JRA 267	JRST 254	JFCL 255	XCT 256	UO 000-077
Initial Registers	AR ← AC MB ← 0, E MQ ← 0	AR ← AC MB ← C(E) MQ ← 0	AR ← AC MB ← 0, E MQ ← C(CIACRTT)	AR ← AC MB ← 0, E MQ ← C(CIACRTT)	AR ← 0, E MB ← 7, E MQ ← 0	AR ← 0, E MB ← 7, E MQ ← 0	AR ← AC MB ← 0, E MQ ← 0	AR ← ACRTLT MB ← 0, E MQ ← C(CIACLT)	AR ← 0, E MB ← 7, E MQ ← 0	AR ← 0, E MB ← 7, E MQ ← 0	AR ← 0, E MB ← C(E) MQ ← 0	AR ← 0, E MB ← 7, E MQ ← 0
Initial Gates	ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(E) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(CIACRTT) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(CIACRTT) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FAC INH E LONG MB ← AR(J) INH (ET10)	FAC INH E LONG MB ← AR(J) INH (ET10)	E LONG MB ← AR(J) INH (ET10)	FC(CIACLT) E LONG MB ← AR(J) INH (ET10)	FAC INH E LONG	FAC INH E LONG	FAC INH FC(E) PC + 1 INH ET4 INH	FAC INH PC + 1 INH ET4 INH
ET0A			MB ← MQ(J)	MB ← MQ(J)	PI HOLD: PIH ← PI CHRQ	MB(J) ← AR(J)	MBLT(J) ← MBRT(J)	MB ← MQ(J)	IR10(1): RUN ← (0) IR11(1): AR FLAG CLEAR (Clears MISC BITS of ET0)			
ET1				MA ← (0)					IR9(1): PIH(0) ← PIOK(1) IR11(1): AR FLAG SET (Transfers MB(1) into MISC BITS) IR12(1): EX MODE SYNC ← (1)			
ET3	ET4 AR PSE ← (1) AR + 1 LTRT ART3	ET4 AR PSE ← (1) AR + 1 LTRT ART3	ET4 AR PSE ← (1) AR - 1 LTRT ART3	ET4 AR PSE ← (1) AR - 1 LTRT ART3								
ET4	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)								
ET5	MB ← (0)				MB ← (0)	MB ← (0)						
ET6	MB ← MISC BITS, PC(1)				MB ← MISC BITS, PC(1)	MB ← MISC BITS, PC(1)	MB ← PC(1)					
ET7	PC ← (0)			PC ← (0)	PC ← (0)	PC ← (0)	PC ← (0)	PC ← (0)			SELECTED FLAG(1): PC ← (0)	
ET8	PC ← MA(1)			PC ← MA(1)	PC ← MA(1)	PC ← MA(1)	PC ← MA(1)	PC ← MA(1)			SELECTED FLAG(1): PC ← MA(1)	
ET9	MB(J) ← AR(J) MA ← (0)	MB(J) ← AR(J) MA ← (0)	MB(J) ↔ AR(J)	MB(J) ↔ AR(J)	PC + 1 CHF7 ← (0)	MB(J) ← AR(J)	MB(J) ↔ AR(J) PC + 1	MB(J) ↔ AR(J)				
ET10	MA ← MBRT(1) MB(J) ↔ AR(J) AR CRY0(1): CPA PDL OV ← (1) IOB PI REQ CPA PIA		MB(J) ↔ AR(J) AR CRY0(1): CPA PDL OV ← (1) IOB PI REQ CPA PIA		PI HOLD: PI OV, PI CYC ← (0)				MA ← MBRT(1)			
Final Gates	SC(E)	SC(E)	SC(E)		SC(E) SAC INH		SC(E)		SAC INH		SAC INH	



- | | |
|-------------------------|------------------------|
| MISC BITS | SELECTED FLAGS |
| 0 AR OV FLAG | IR9(1) AR OV FLAG |
| 1 AR CRY0 FLAG | IR10(1) AR CRY0 FLAG |
| 2 AR CRY1 FLAG | IR11(1) AR CRY1 FLAG |
| 3 AR PC CHG FLAG | IR12(1) AR PC CHG FLAG |
| 4 CHF7 | |
| 5 EX USER - STORE ONLY | |
| EX MODE SYNC - SET ONLY | |

BOOLEAN
400-477

IR + 100 XXX XY
IR BOOLEAN - IRO - 2(100)
XXXX specifies Boolean function,
decoded into IR6(0)-17 as
BOOLE N - IR BOOLE - IR3-6(N)

ADD-SUBTRACT
270-277

IR + 010 111 XY
IR 2XX + IRO - 2(100)
IR AS + IR 2XX - IR3-5(111)
X specifies odd or subtract
Y specifies add or subtract
IR AS + IR 6 - 0 + AS +
10 + BOOLE AS 10
11 : BOOLE AS 11

In both instruction groups YY specifies common mode

(IR BOOLE, IR AS) - IR 7, 8
00 : BOOLE AS 00
01 : BOOLE AS 01
10 : BOOLE AS 10
11 : BOOLE AS 11

Action
BOOLE + AS 00: /AC, C(E) -> AC
BOOLE + AS 01: /AC, (O, E) -> AC
BOOLE + AS 10: /AC, C(E) -> E
BOOLE + AS 11: /AC, C(E) -> AC, E

Initial Registers
AR + AC
BOOLE AS 00: 10: 11: MB + C(E)
BOOLE AS 01: MB + (O, E)
MQ = 0

Common Initial Gates
BOOLE + AS 00: FC(E)
BOOLE + AS 10: FC(E) PSE

Initial Gates As Only
ET4 INH
AR SBR

ETOA	BOOLE 0, 3, 14, 17: AR - (O)
	BOOLE 2, 4, 12, 13, 15: AR COM
ET1	BOOLE 6, 11, 14: AR - MB(v)
	BOOLE 1, 2, 15, 16: AR - MB(O)
	BOOLE 3, 4, 7, 10, 13: AR - MB(1)

ET3	ET4 AR PSE - (1)	AS + AR ADD ART3
ET4	BOOLE 4, 10, 11, 14, 15, 16, 17: AR COM	ET4 AR SUB - (O)

ET10	FC(E) PSE: MB -> AR(J)
------	------------------------

BOOLEAN FUNCTIONS

0 SETZ	1 AND	2 ANDCA	3 SETM
ETOA AR - (O)	AR - (O)	AR COM	AR - (O)
ET1	AR - MB(O)	AR - MB(O)	AR - MB(1)
ET4			

4 ANDCM	5 SETA	6 XOR	7 IOR
ETOA AR COM	AR COM	AR - MB(v)	AR - MB(1)
ET1	AR - MB(1)	AR - MB(v)	AR - MB(1)
ET4	AR COM		

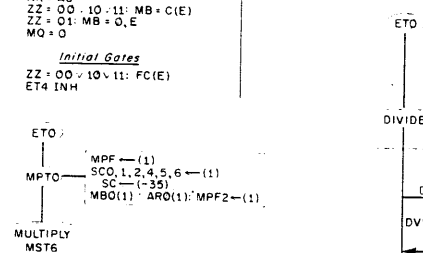
10 ANDCB	11 EQV	12 SETCA	13 ORCA
ETOA AR - MB(1)	AR - MB(v)	AR COM	AR COM
ET1	AR - MB(1)	AR - MB(v)	AR - MB(1)
ET4	AR COM	AR COM	

14 SETCM	15 ORCM	16 ORCB	17 SETO
ETOA AR - (O)	AR COM	AR - MB(O)	AR - (O)
ET1	AR - MB(v)	AR - MB(O)	AR - MB(1)
ET4	AR COM	AR COM	AR COM

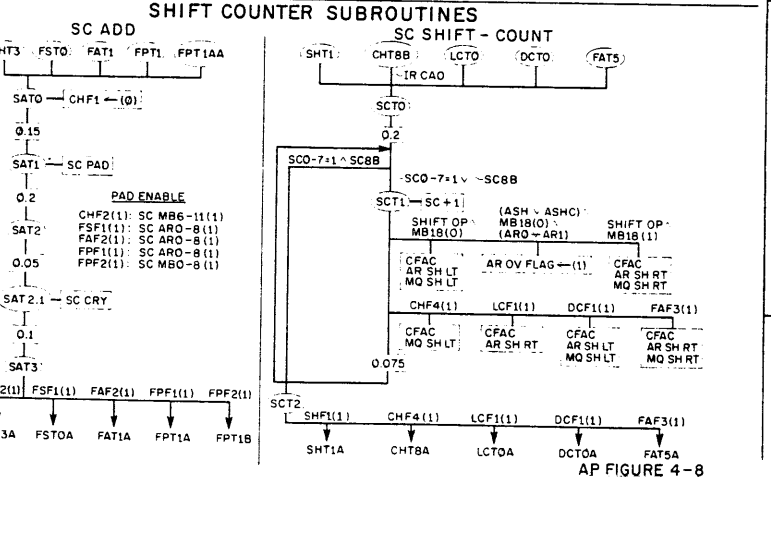
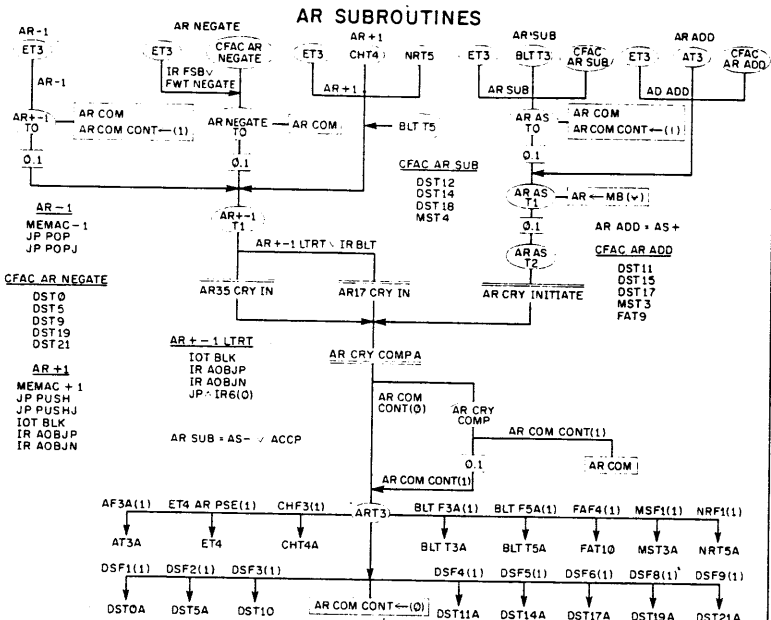
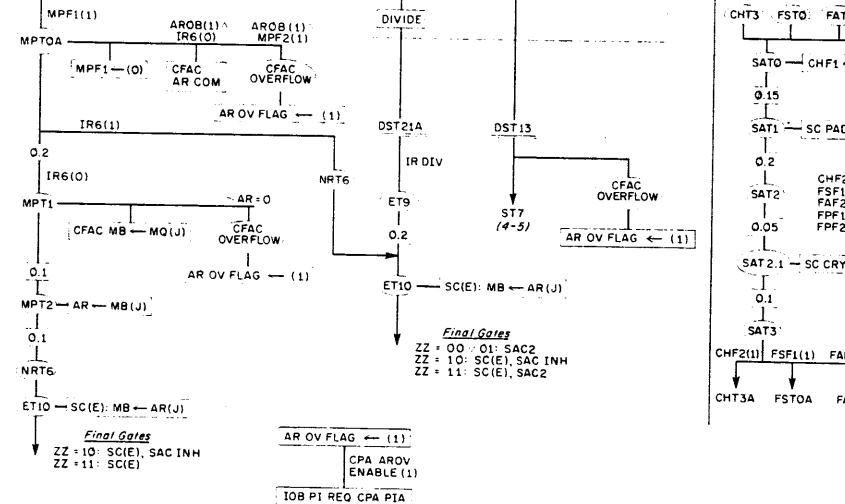
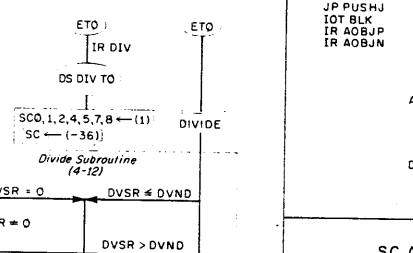
MULTIPLY-DIVIDE
220-237

Levels for fetch and store according to operand type and mode
IR MD + IR 2XX - IR3(O) - IR4(1)
IR MD FC(E) + IR MD - (IR7(O) - IR8(1))
IR MD FLAG2 + IR DIV - IR6(1)
IR MD SC(E) + IR MD - IR7(1)
IR MD SAC INH + IR MD - IR7(1) - IR8(O)
IR MD SAC2 - IR MD SAC INH - (IR DIV - IR MUL - IR6(1))

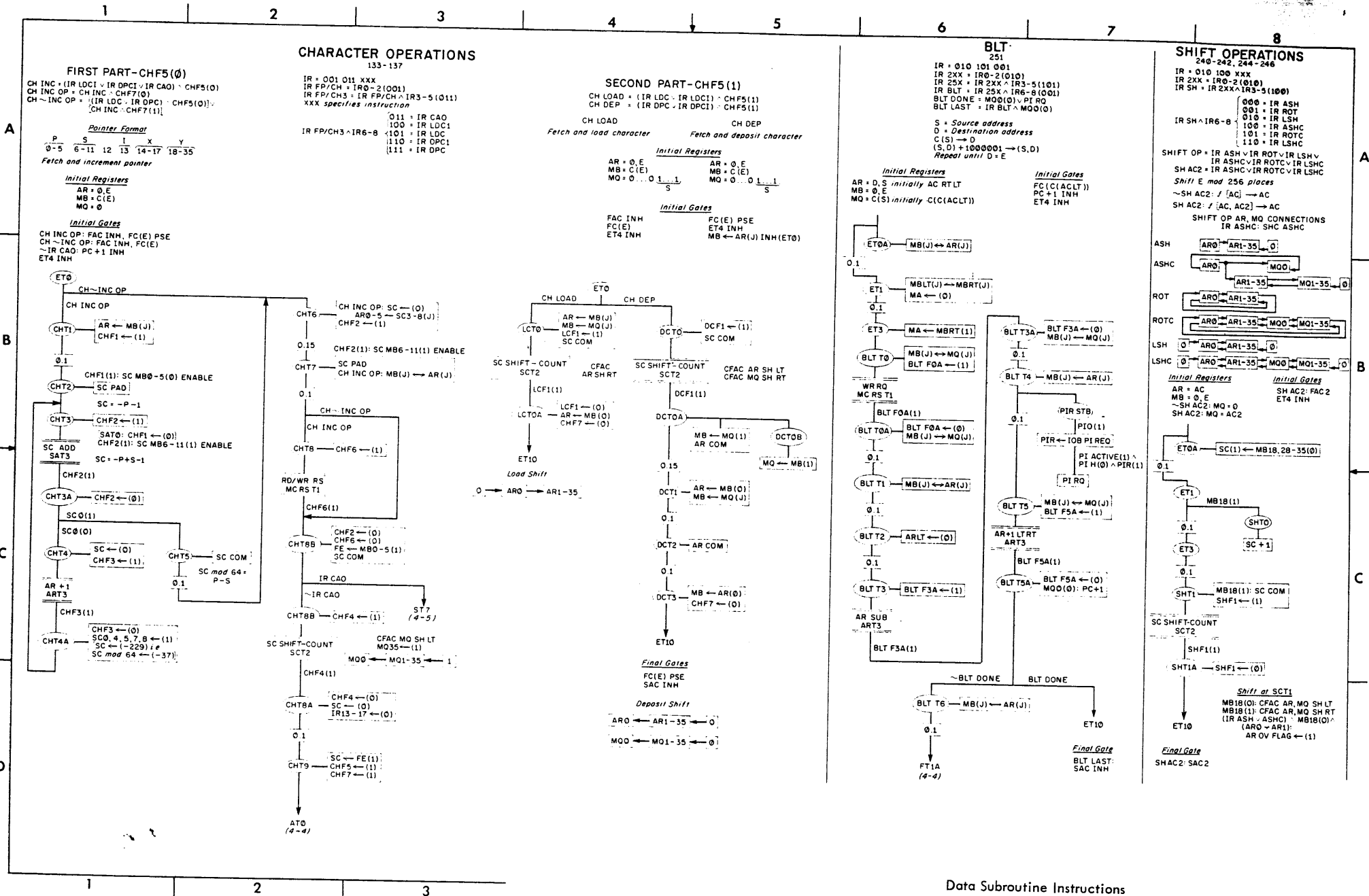
IR MUL Action
ZZ + 00: AC - C(E) -> AC, AC2
ZZ + 01: AC - (O, E) -> AC, AC2
ZZ + 10: AC - C(E) -> E
ZZ + 11: AC - C(E) -> AC, E



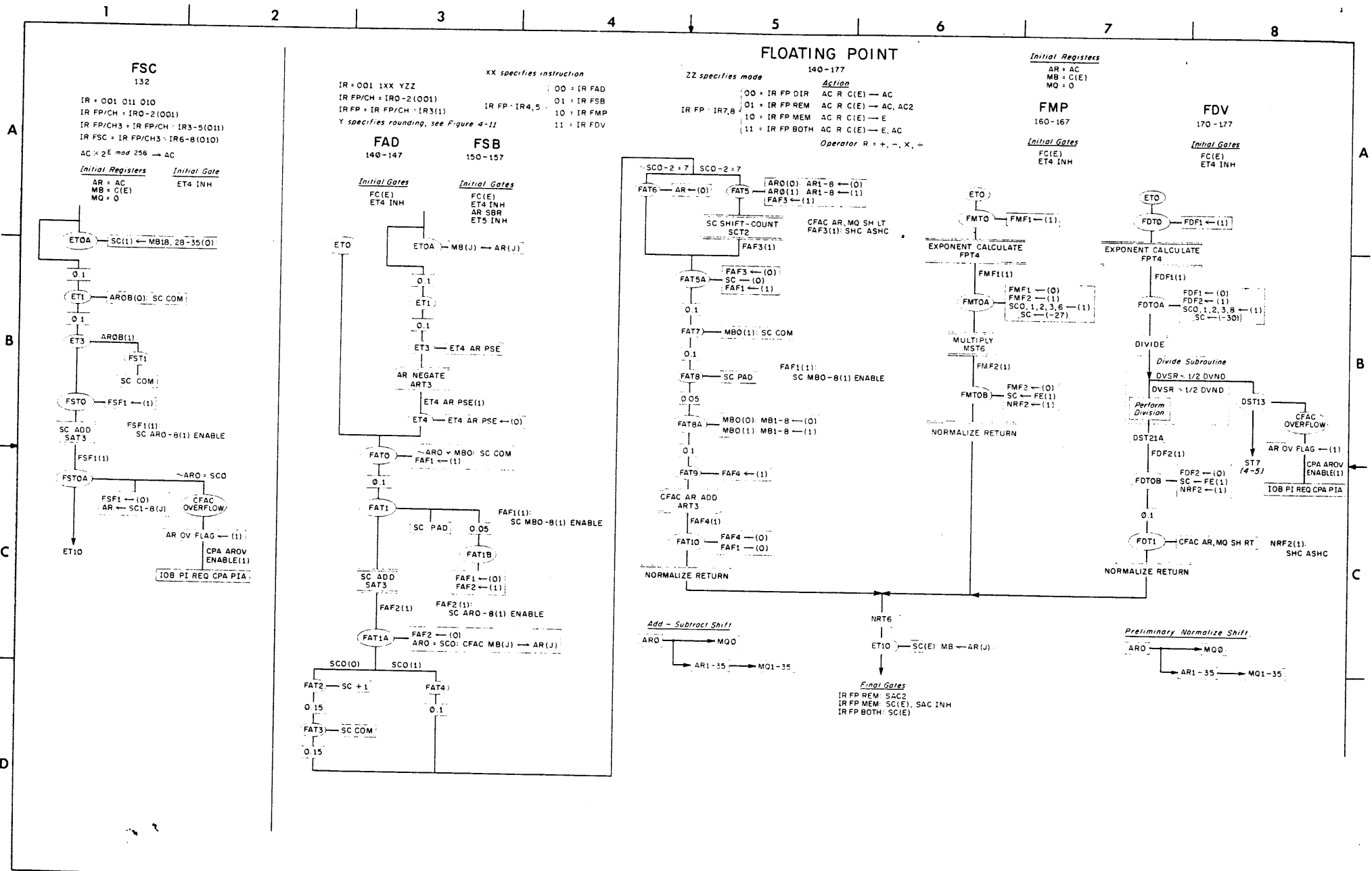
IDIV (XY = 10) Action
ZZ + 00: AC - C(E) -> AC, AC2
ZZ + 01: AC - (O, E) -> AC, AC2
ZZ + 10: AC - C(E) -> E
ZZ + 11: AC - C(E) -> AC, E



Fixed Point and Logical Instructions; AR and SC Subroutines (FD-D-166-0-ALF)



Data Subroutine Instructions
(FD-D-166-0-DSIF)



FLOATING POINT

Initial Registers

AR = AC
MB = C(E)
MQ = O

FMP
160-167

Initial Gates
FC(E)
ET4 INH

FDV
170-177

Initial Gates
FC(E)
ET4 INH

ZZ specifies mode
140-177
Action
00 = IR FP DIR AC R C(E) ← AC
01 = IR FP REM AC R C(E) ← AC, AC2
10 = IR FP MEM AC R C(E) ← E
11 = IR FP BOTH AC R C(E) ← E, AC
*Operator R = +, -, X, **

XX specifies instruction
00 = IR FAD
01 = IR FSB
10 = IR FMP
11 = IR FDV

IR = 001 1XX YZZ
IR FPCH = IR0-2(001)
IR FPCH3 = IR FP/CH = IR3-5(011)
IR FSC = IR FP/CH3 = IR6-8(010)
AC × 2^E mod 256 → AC
Y specifies rounding, see Figure 4-11

FSC
132

IR = 001 011 010
IR FP/CH = IR0-2(001)
IR FP/CH3 = IR FP/CH = IR3-5(011)
IR FSC = IR FP/CH3 = IR6-8(010)
AC × 2^E mod 256 → AC

Initial Registers
AR = AC
MB = C(E)
MQ = O

Initial Gate
ET4 INH

FAD
140-147

Initial Gates
FC(E)
ET4 INH

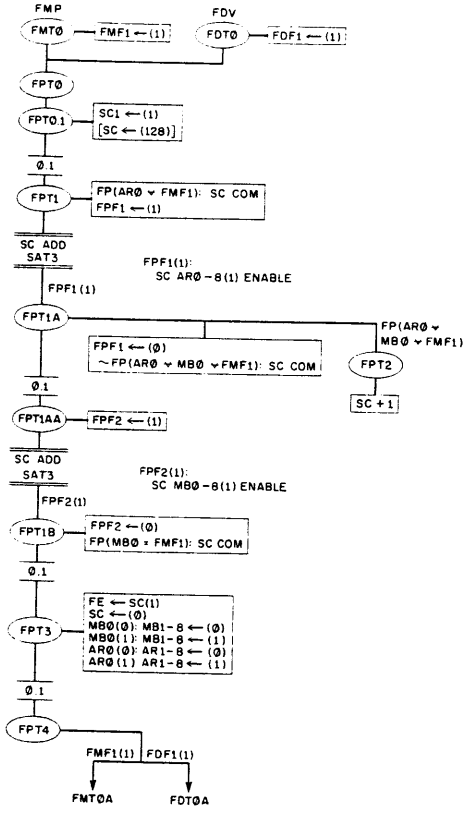
FSB
150-157

Initial Gates
FC(E)
ET4 INH
AR SBR
ET5 INH

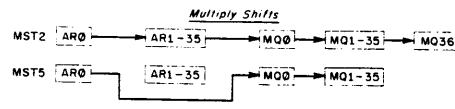
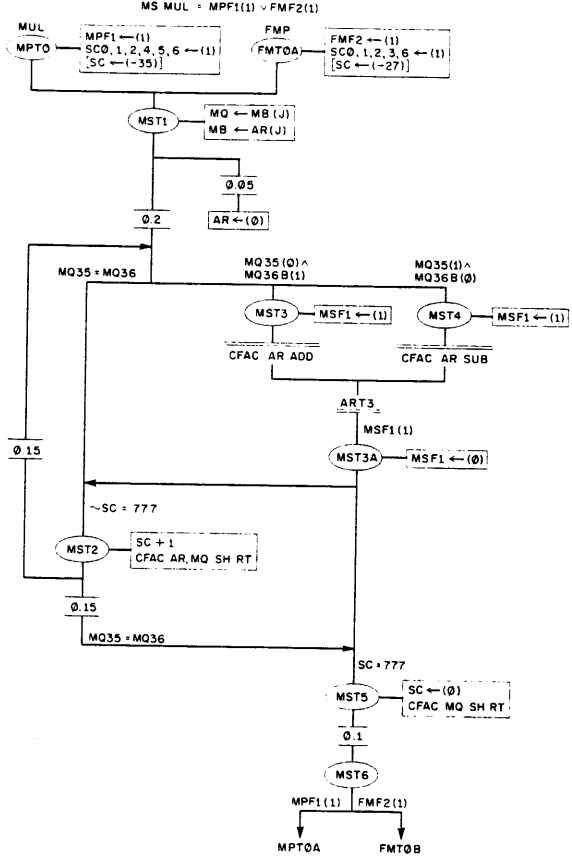
Floating Point Instructions
(FD-D-166-0-FPP)

A
B
C
D

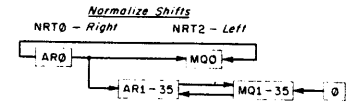
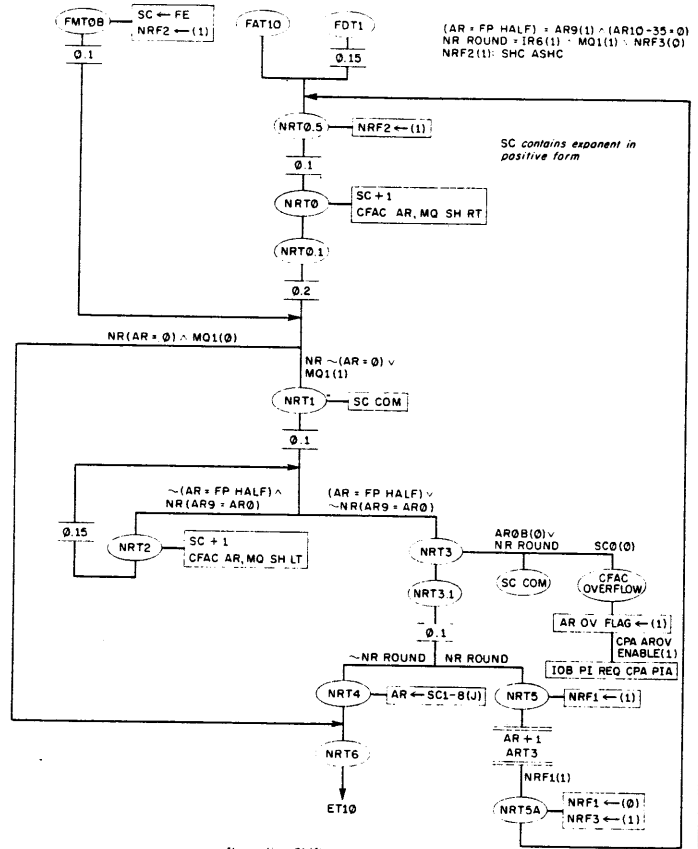
EXPONENT CALCULATE



MULTIPLY



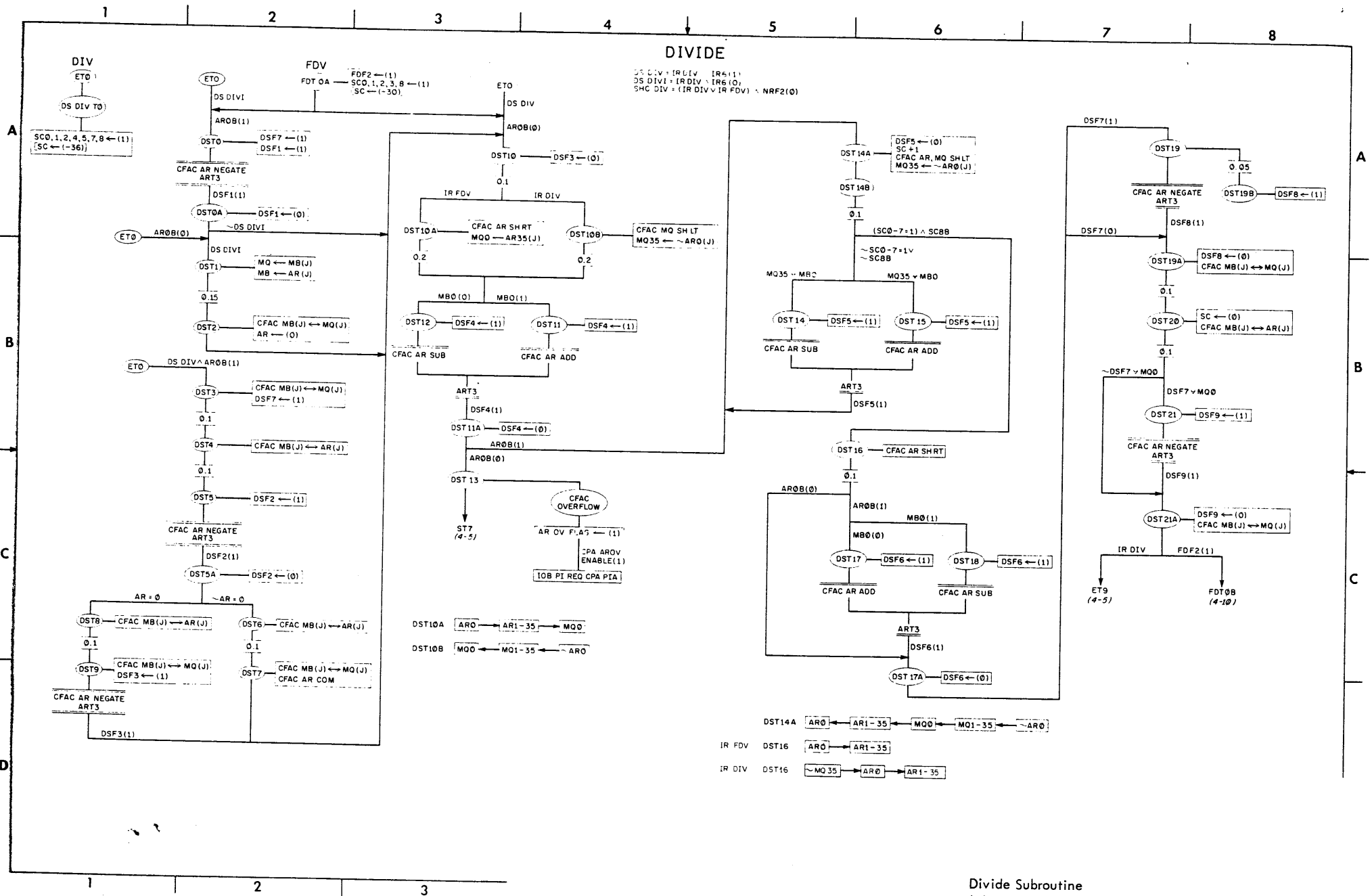
NORMALIZE RETURN



Exponent Calculate, Multiply and Normalize Return Subroutines (FD-D-166-0-EMNF)

DIVIDE

DS DIV ← IR DIV ∨ IR6(1)
 DS DIV1 ← IR DIV ∨ IR6(0)
 5HC DIV ← (IR DIV ∨ IR FDV) ∨ NRF2(0)



Divide Subroutine
(FD-D-166-0-DSF)

IN-OUT TRANSFER

IR = 111 XXX XXX XYY Y
 IR IOTA = IR0-2 (111)
 IR IOT = IR IOTA ^ ~ EX IR UUD
 YYY specifies instruction

IR IOT ^ IR I0-12

000	IOT BLKI
001	IOT DATAI
010	IOT BLKO
011	IOT DATAO
100	IOT CONO
101	IOT CONI
110	IOT CONSZ
111	IOT CONSO

XXX XXX X specifies device. IR3-9 outputs thru bus drivers to I0S3-9 lines on I0 bus. Each I0 control unit contains a decoding net that allows proper code to select device by gating in command signals from bus

Instruction Code

BLKI	C(E) + 1000001 → E. DEVICE BUFFER → C(C(ERT))
DATAI	DEVICE BUFFER → E
BLKO	C(E) + 1000001 → E. C(C(ERT)) → DEVICE BUFFER
DATAO	C(E) → DEVICE BUFFER
CONO	E → DEVICE CONTROL
CONI	STATUS → E
CONSZ	(STATUS ^ E) = 0: PC + 1
CONSO	(STATUS ^ E) ≠ 0: PC + 1

IOT BLK = IOT BLKI ∨ IOT BLKO
 IOT OUTGOING = IOT DATAO ∨ IOT CONO
 IOT STATUS = IOT CONI ∨ IOT CONSZ ∨ IOT CONSO
 IOT DATAI ∨ IOT DATAO ∨ IOT DATAI
 PI BLK RST = IOT DATAI ∨ PI OV(0)
 PI BLK RST ^ PI CYC B(1): PI HOLD, PI RST
 IOT CONSZ ∨ IOT CONSO: E LONG

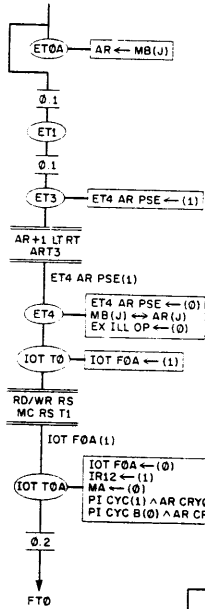
BLK

Initial Registers

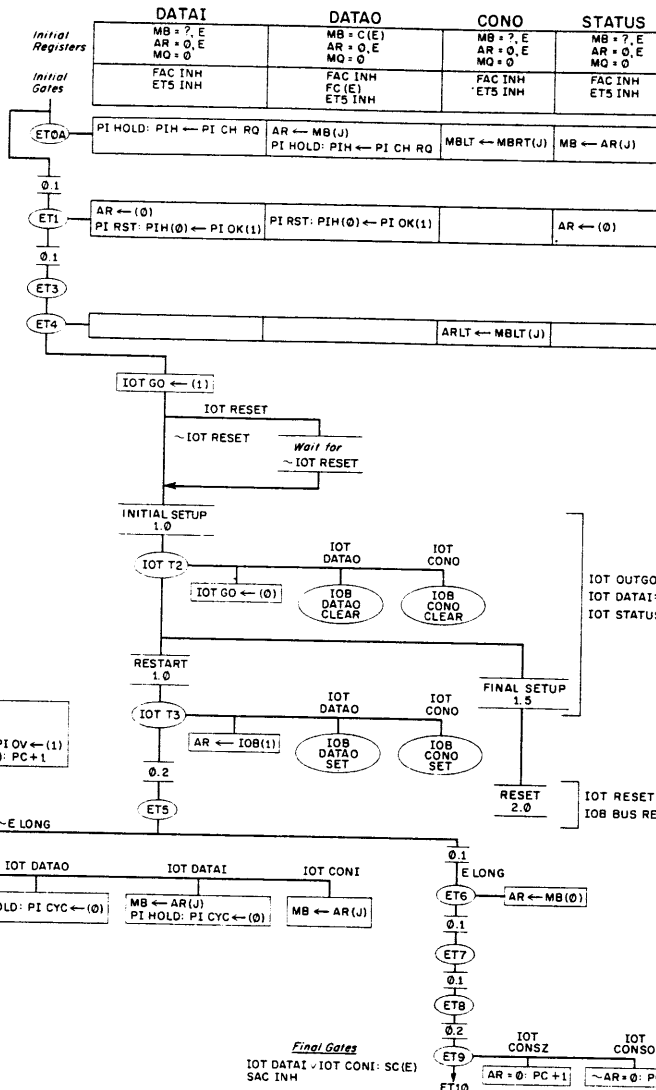
MB ← C(E)
 AR ← 0, E
 MQ ← 0

Initial Gates

FAC INH
 FC(E) PSE
 PC + 1 INH
 ET4 INH
 AR SBR
 ET5 INH

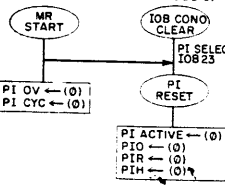


Return to fetch and perform data instruction with C(MBRT) as E

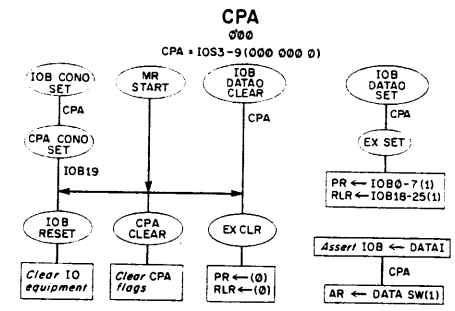


PI

PI SELECT = I0S3-9 (000 000 1)



IOB	PI CONO SET	PI STATUS
24	PI R ← IOB29-35(1)	
25	PI O ← IOB29-35(1)	
26	PI R(0) ← IOB29-35(1)	
27	PI ACTIVE ← (0)	
28	PI ACTIVE ← (1)	
29-35		PI ACTIVE(1) PI O1-7(1)

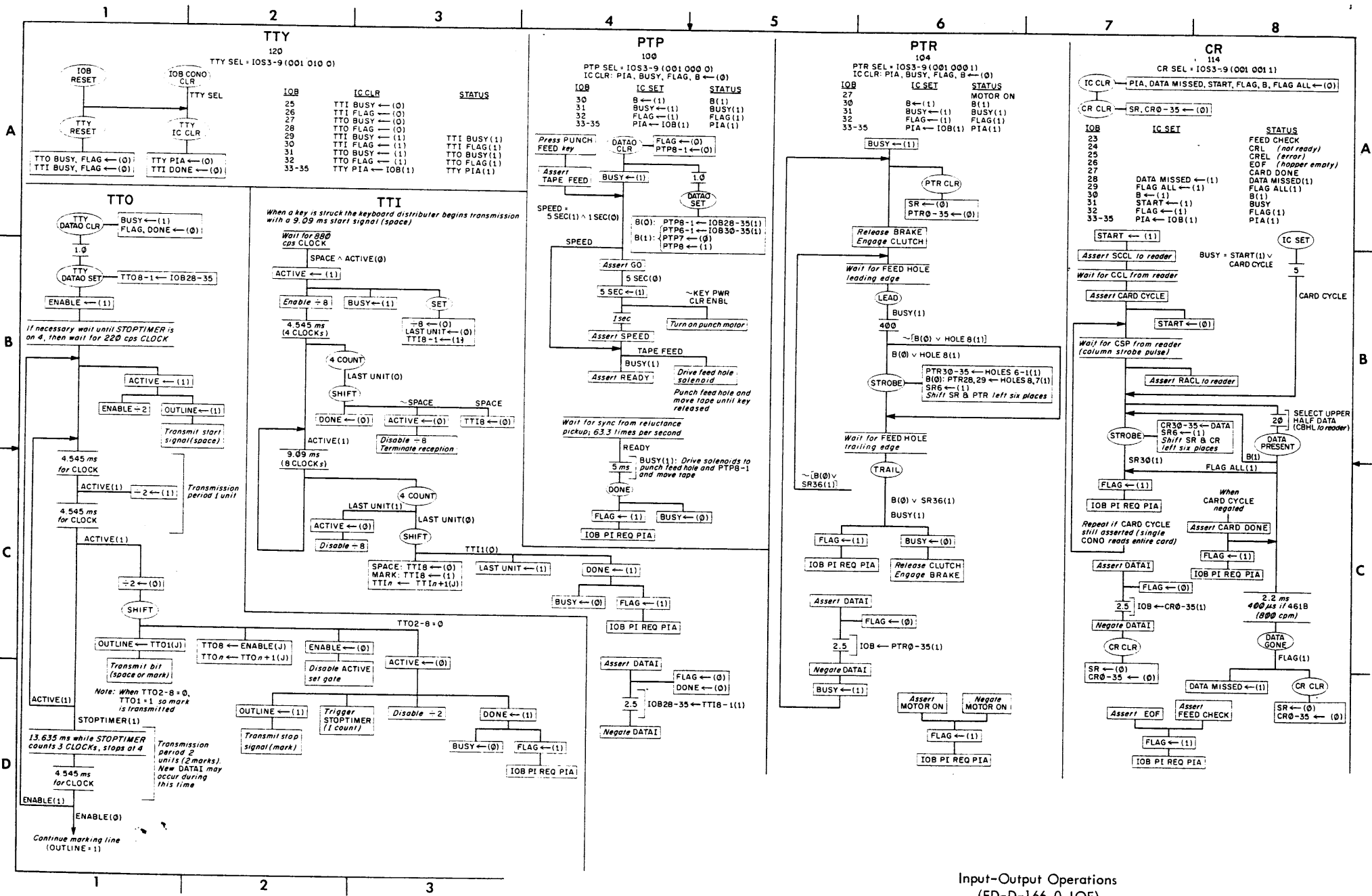


IOB	CPA CONO SET	CPA STATUS
18	CPA PDL OV ← (0)	CPA PDL OV(1)
19	IOB RESET	CPA IOT USER(1)
20	CPA IOT USER ← (1)	EX USER B(1)
21	CPA IOT USER ← (0)	CPA ILLEG OP(1)
22	CPA ILLEG OP ← (0)	CPA NON EXIST MEM(1)
23	CPA NON EXIST MEM ← (0)	CPA NON EXIST MEM(1)
24	CPA CLOCK ENABLE ← (0)	CPA CLOCK ENABLE(1)
25	CPA CLOCK ENABLE ← (1)	CPA CLOCK FLAG(1)
26	CPA CLOCK FLAG ← (0)	
27	CPA PC CHG ENABLE ← (0)	CPA PC CHG ENABLE(1)
28	CPA PC CHG ENABLE ← (1)	AR PC CHG FLAG B(1)
29	AR PC CHG FLAG ← (0)	
30	CPA AROV ENABLE ← (0)	CPA AROV ENABLE(1)
31	CPA AROV ENABLE ← (1)	AR OV FLAG B(1)
32	AR OV FLAG ← (0)	
33-35	CPA PIA ← IOB33-35(1)	CPA PIA B(1)

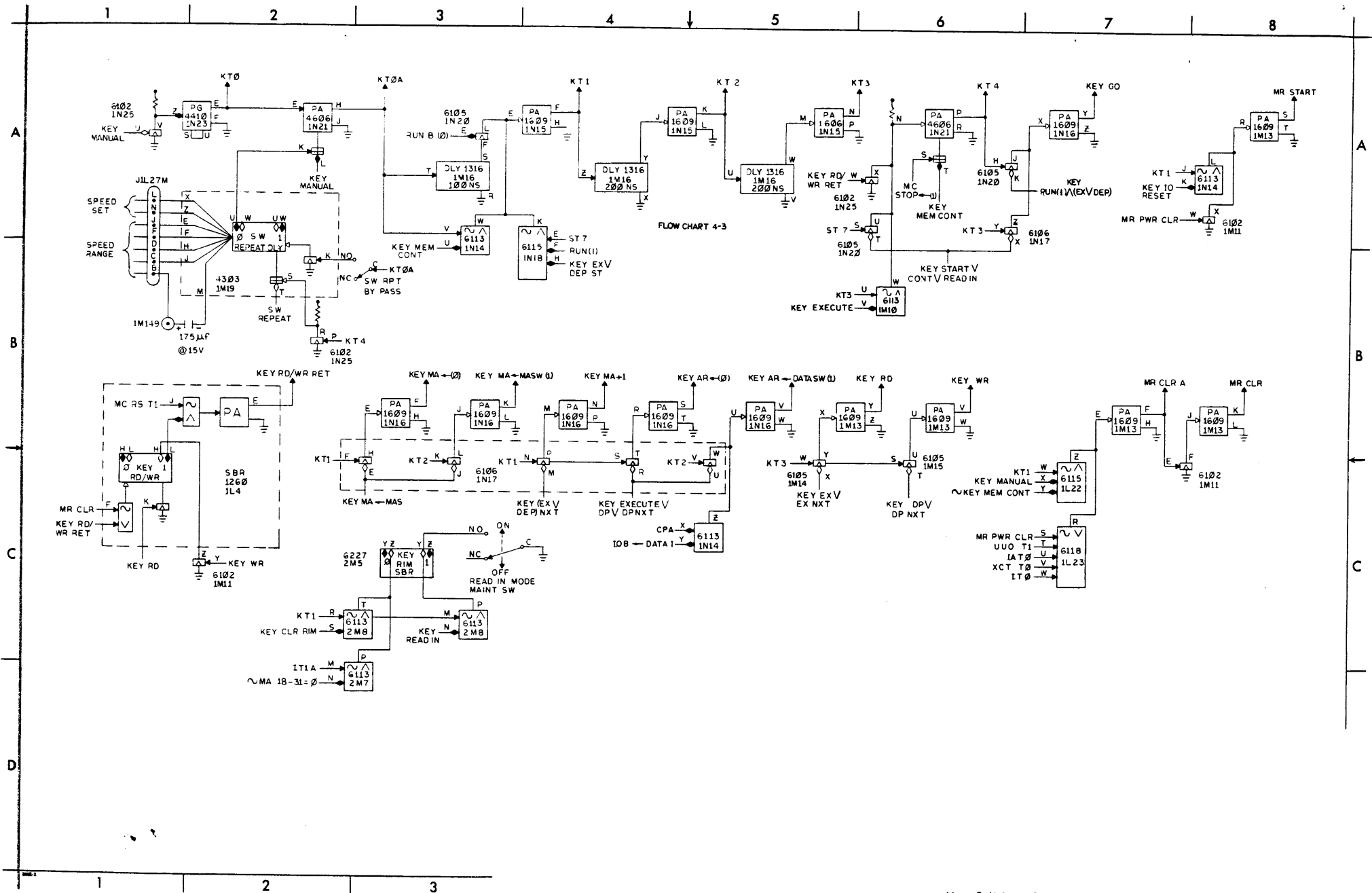
60 times per second from power control: CPA CLOCK FLAG ← (1)
 MC ILLEG ADDRESS: CPA ILLEG OP ← (1)
 MC NON EXIST MEM: CPA NON EXIST MEM ← (1)
 ET10 ^ [MB(J) ← AR(J)](ET10) ^ AR CRY0(1): CPA PDL OV ← (1)

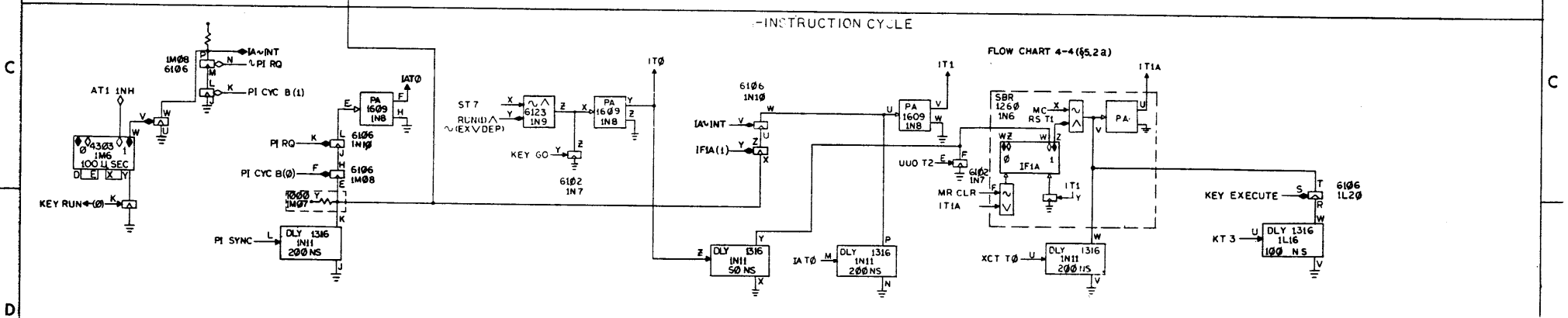
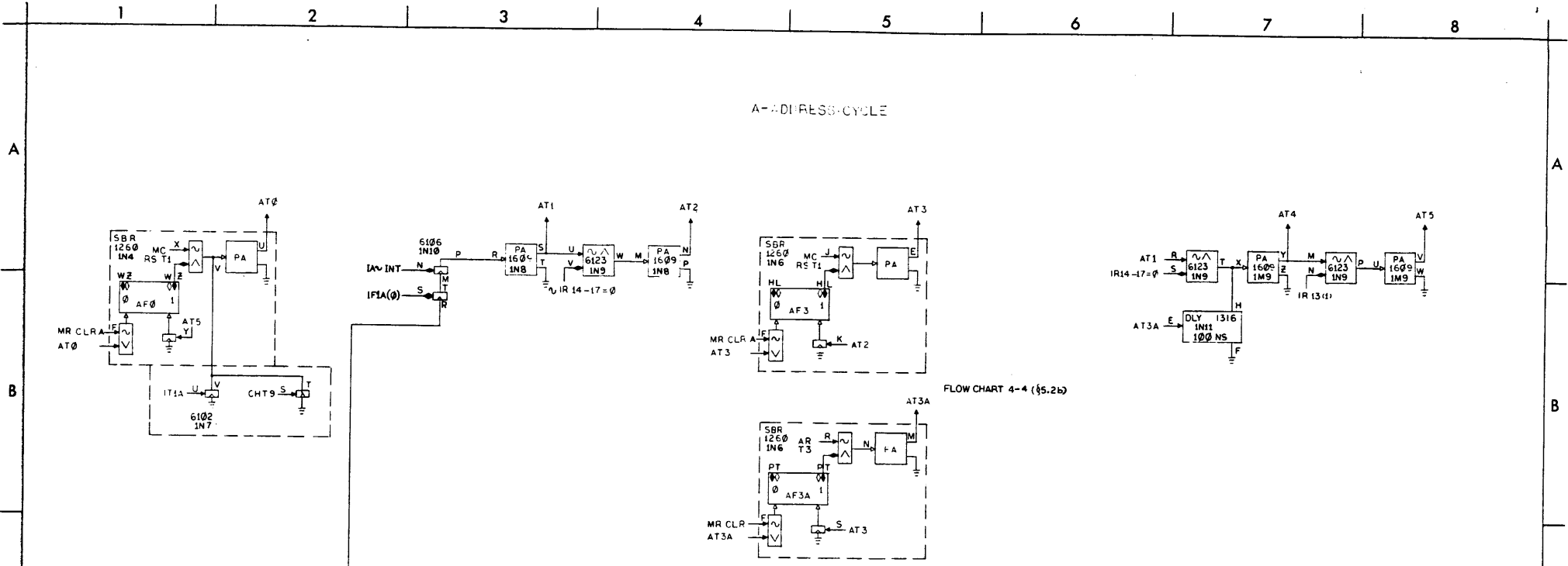
IOB PI REQ CPA PIA

CPA ILLEG OP(1)
CPA NON EXIST MEM(1)
CPA PDL OV(1)
CPA CLOCK ENABLE(1) ^ CPA CLOCK FLAG(1)
CPA PC CHG ENABLE(1) ^ AR PC CHG FLAG B(1)
CPA AROV ENABLE(1) ^ AR OV FLAG(1)



Input-Output Operations
(FD-D-166-0-IOF)





I, A - Instruction and Address Cycles
(BS-D-166-0-1A)

A

B

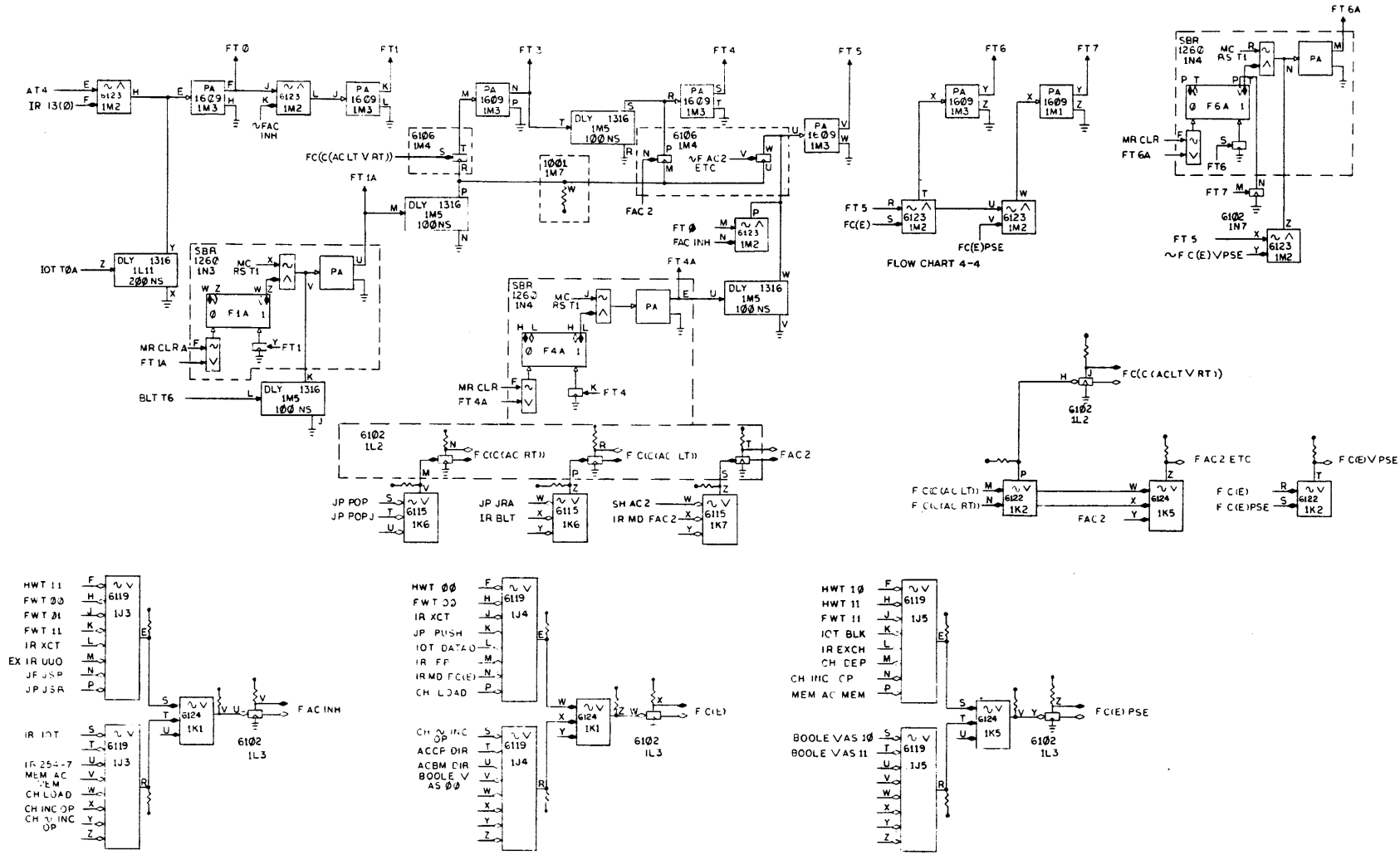
C

D

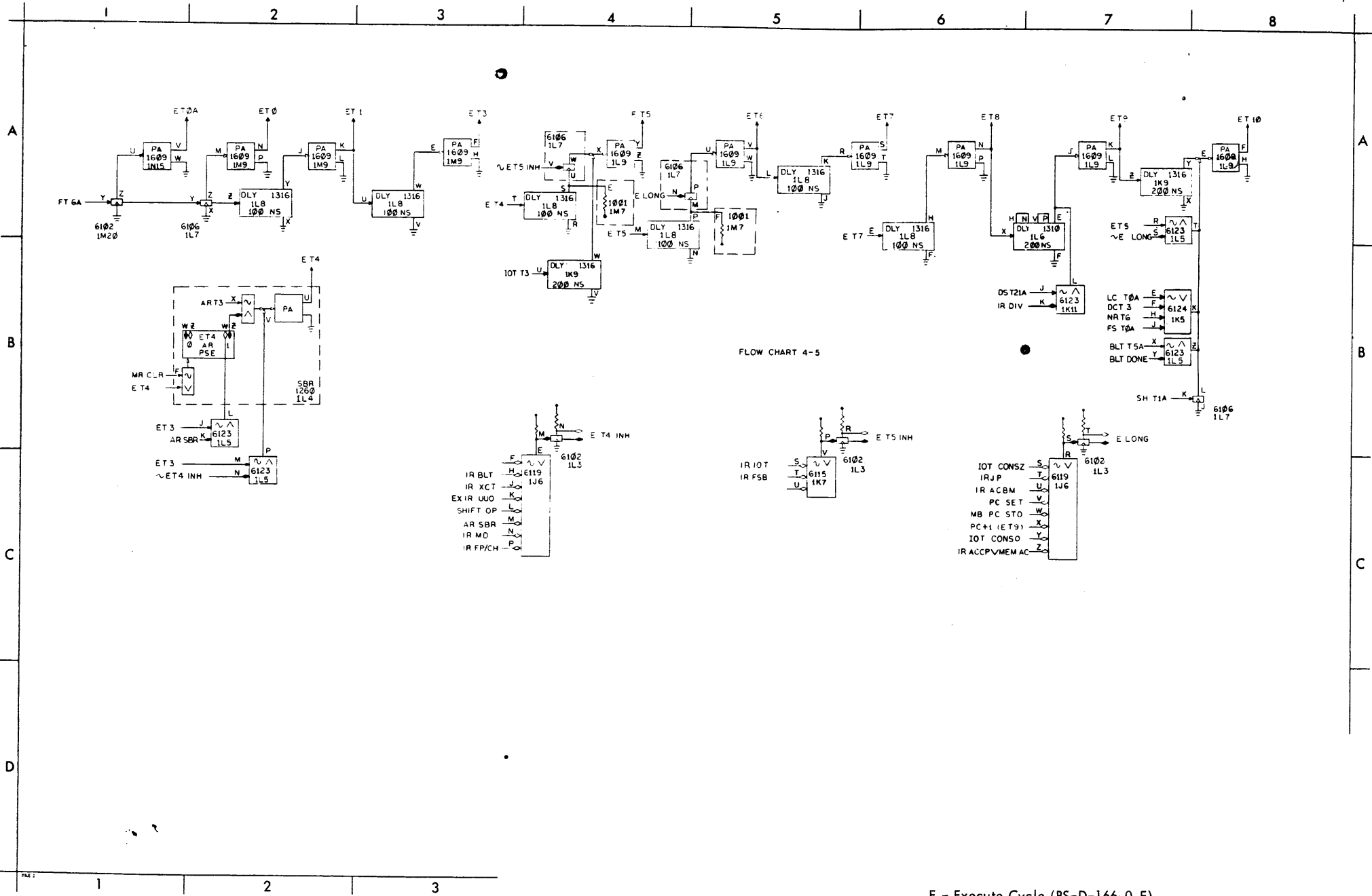
A

B

C

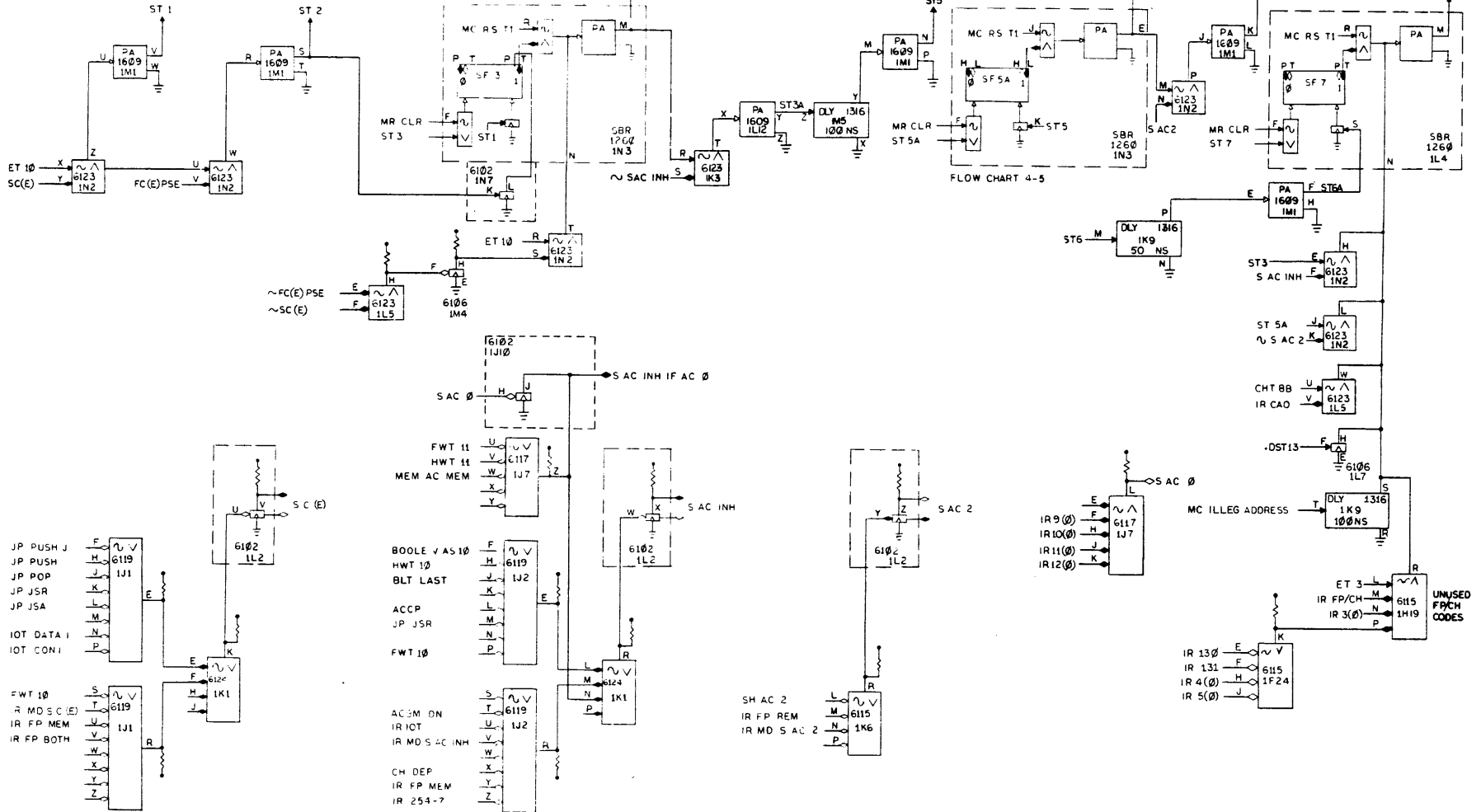


F - Fetch Cycle (BS-D-166-0-F)

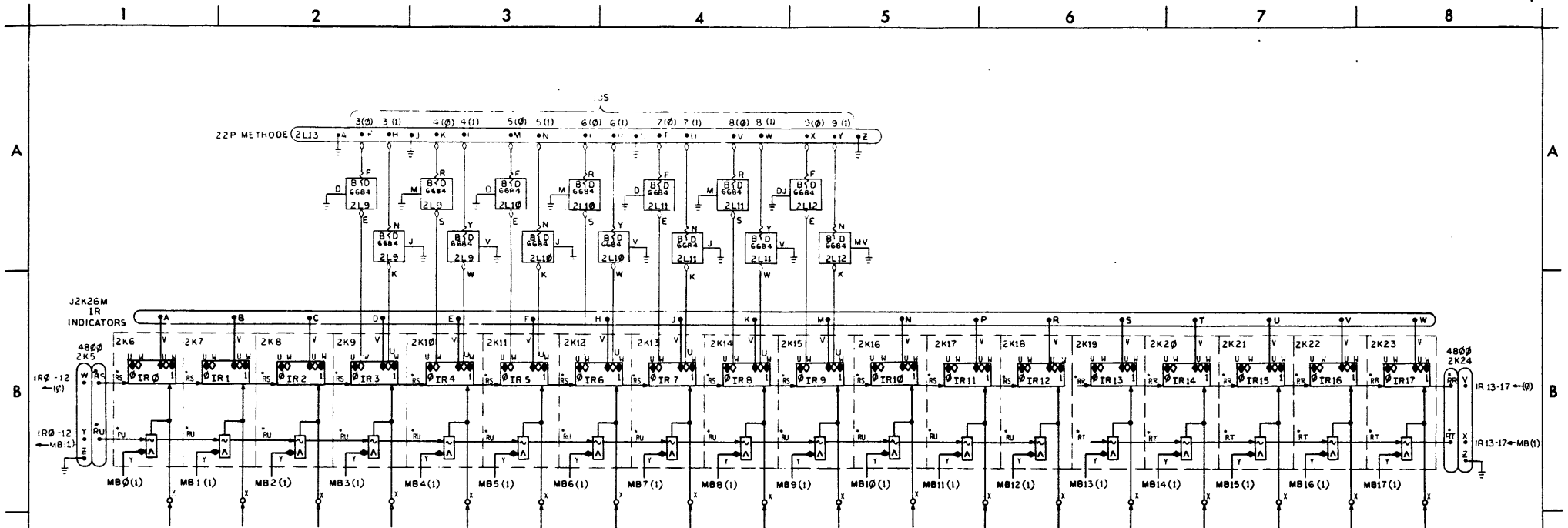


E - Execute Cycle (BS-D-166-0-E)

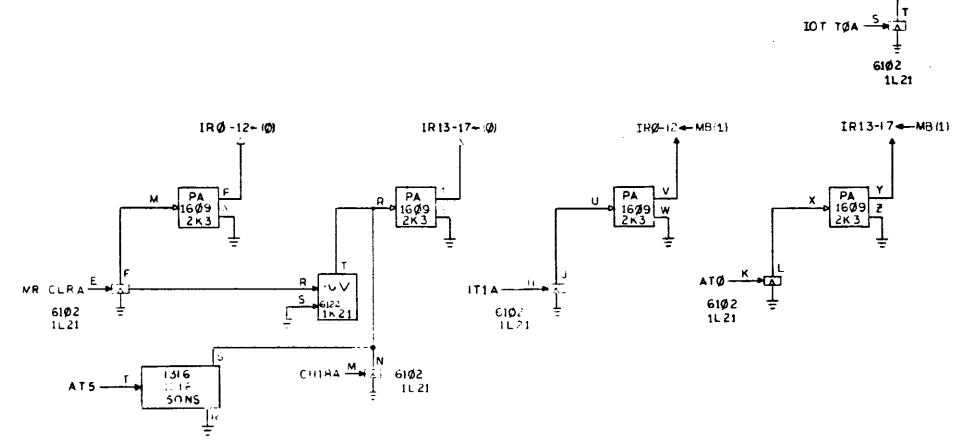
D-166-0-S REV. LTR K



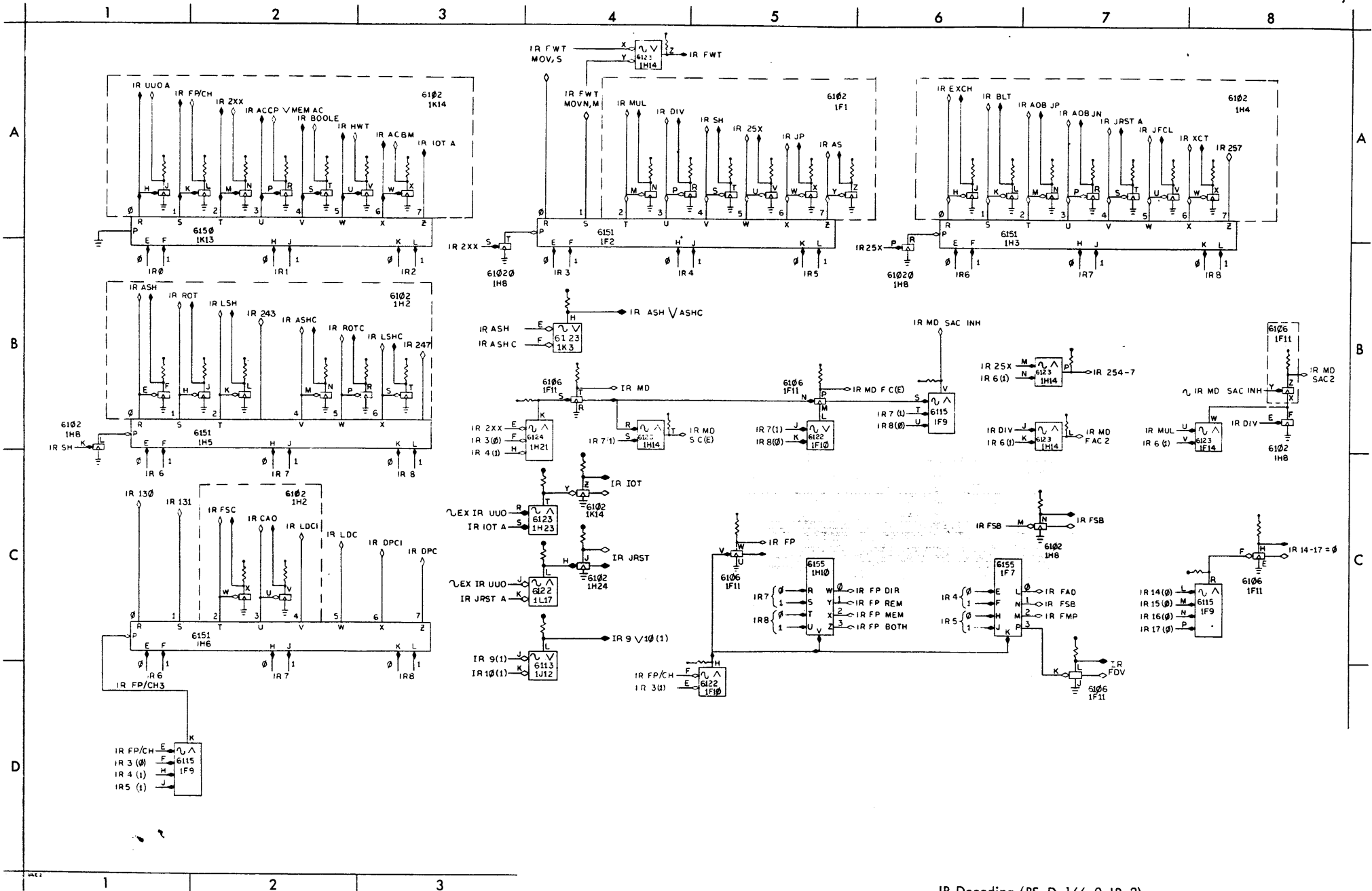
S - Store Cycle (BS-D-166-0-S)



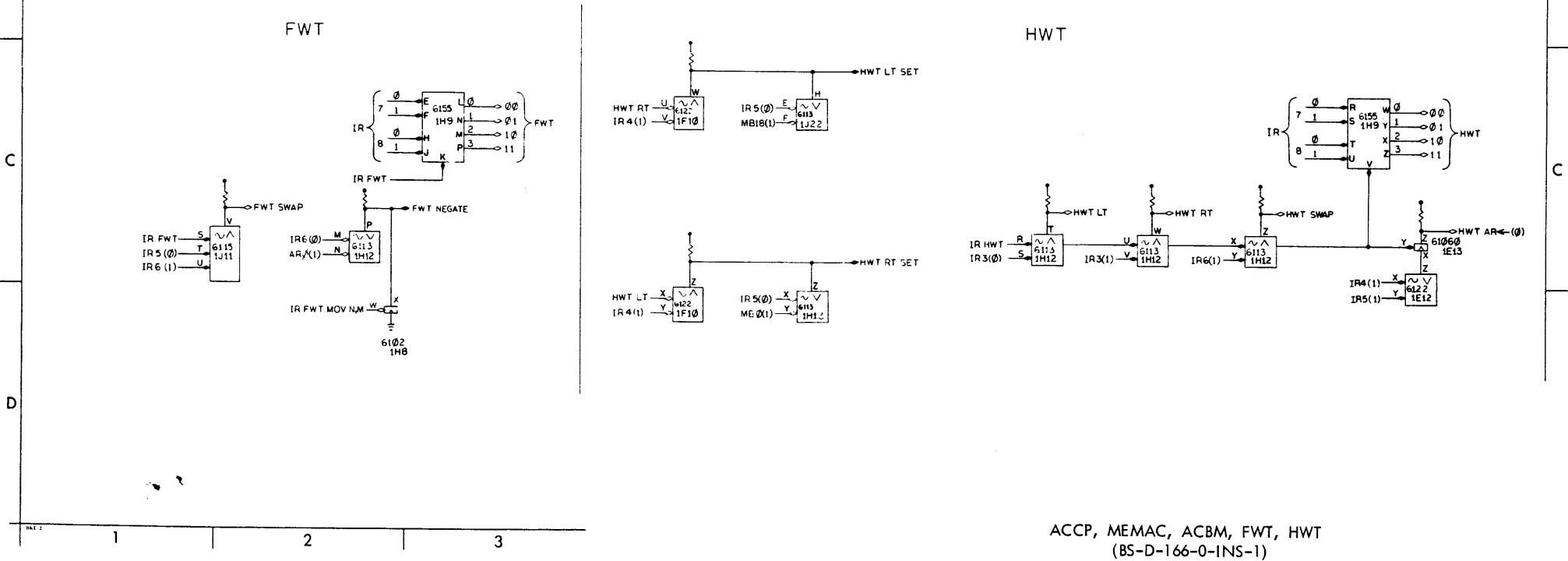
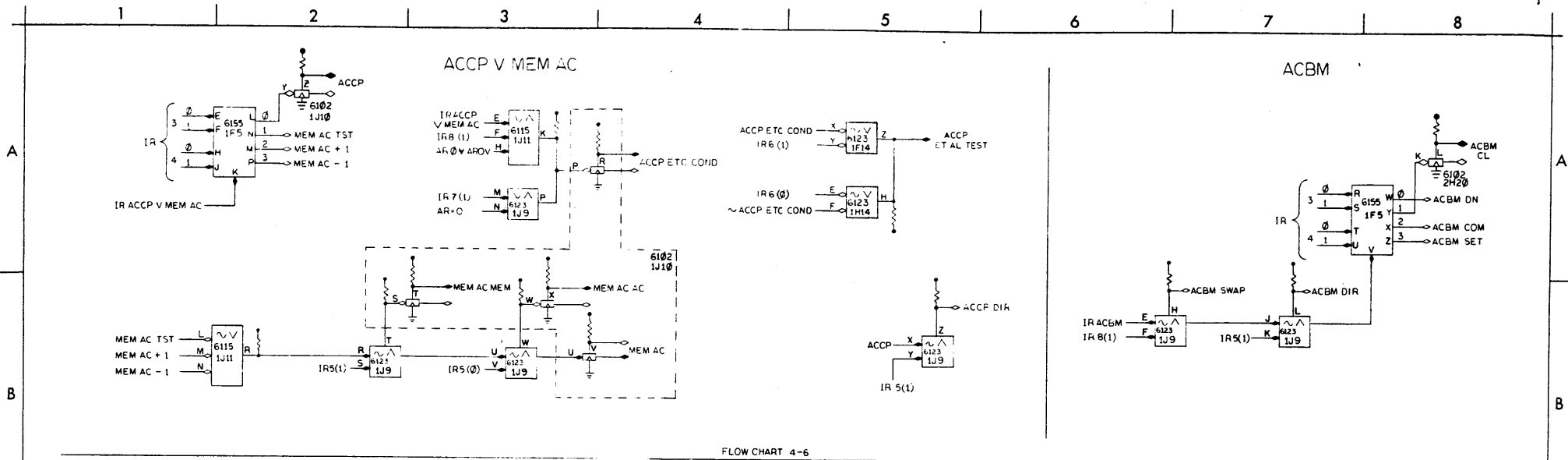
- NOTES:
1. ALL FF PACKAGES ARE 6206
 2. * INDICATES REAR CONNECTOR PIN.
 3. GROUND PIN D AND Z IN ALL 6206'S



IR - Instruction Register
(BS-D-166-0-IR-1)



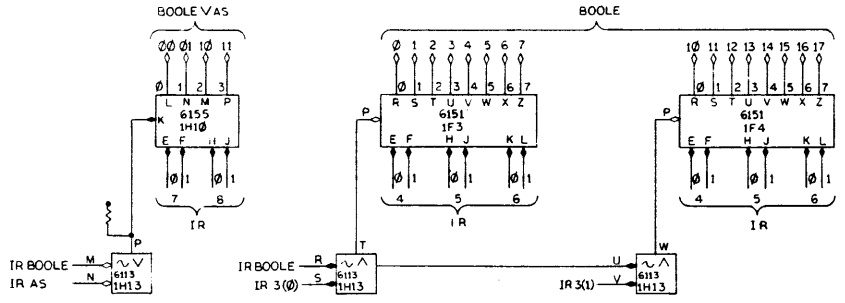
IR Decoding (BS-D-166-0-IR-2)



A

		RESULT BIT FOR OPERAND BIT CONFIGURATION			ET0			ET1			ET4		
					T1	T2	T3						
		MEM AC	MEM AC	MEM AC	AR CLR	AR COM	AR=MB(0)	AR=MB(1)	AR=MB(2)	AR COM			
SETZ	0	MEM A AC	0	0	0	0							
AND	01	MEM A AC	0	0	1	0							
ANDCA	02	MEM A AC	0	1	0								
SETM	03	MEM	0	0	1	1							
ANDCM	04	MEM A AC	0	1	0	0							
SETA	05	AC	0	1	0	1							
XOR	06	MEM V AC	0	1	1	0							
XOR	07	MEM V AC	0	1	1	1							
ANDCB	10	MEM A AC	1	0	0	0							
EDV	11	MEM V AC	1	0	0	1							
SETCA	12	AC	1	0	1	0							
ORCA	13	MEM V AC	1	0	1	1							
ANDCB	14	MEM A AC	1	1	0	0							
SETCM	15	MEM	1	1	0	1							
ORCM	16	MEM V AC	1	1	0	1							
ORCB	17	MEM V AC	1	1	1	0							
SETO	1		1	1	1	1							

BOOLE



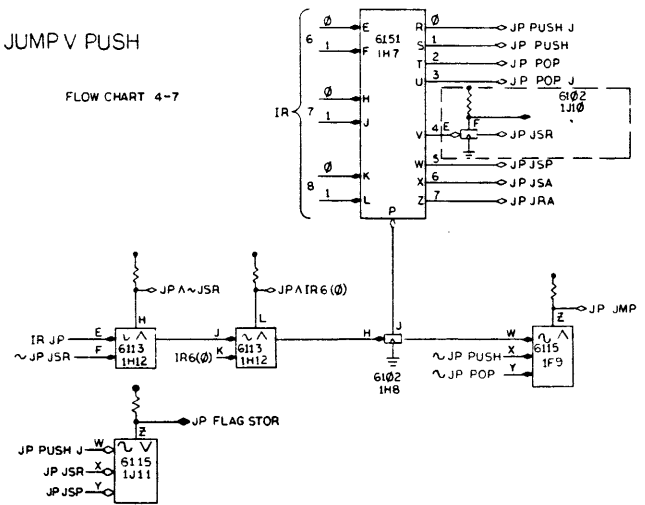
FLOW CHART 4-B

B

B

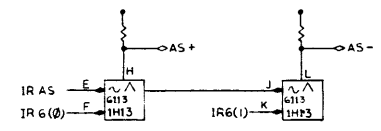
JUMP V PUSH

FLOW CHART 4-7



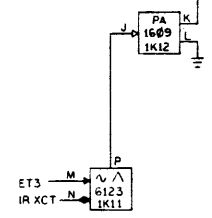
AS

FLOW CHART 4-B



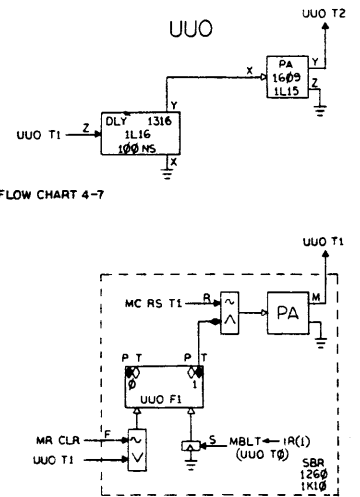
XCT

FLOW CHART 4-7



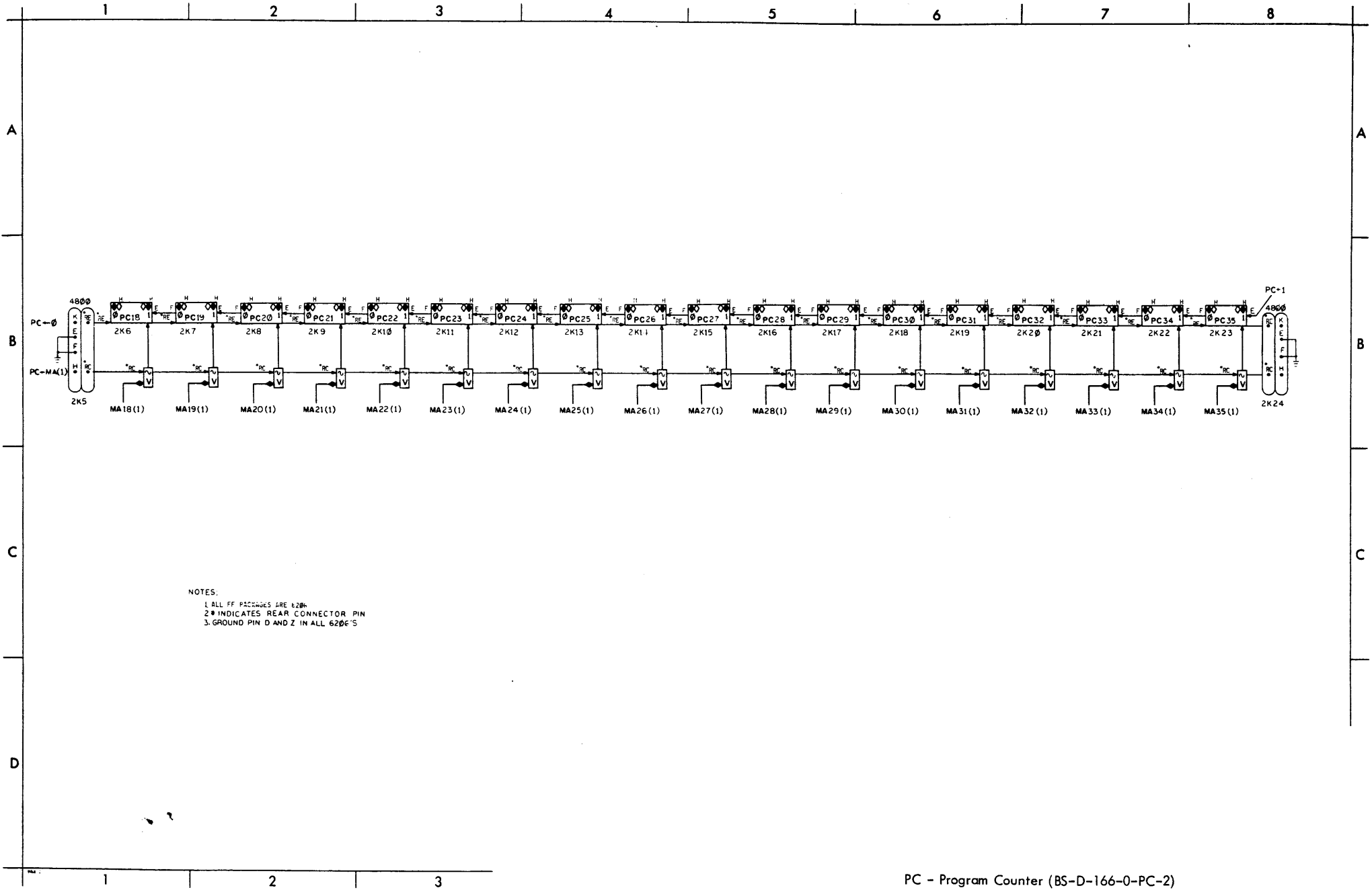
UOO

FLOW CHART 4-7



D

C



NOTES.

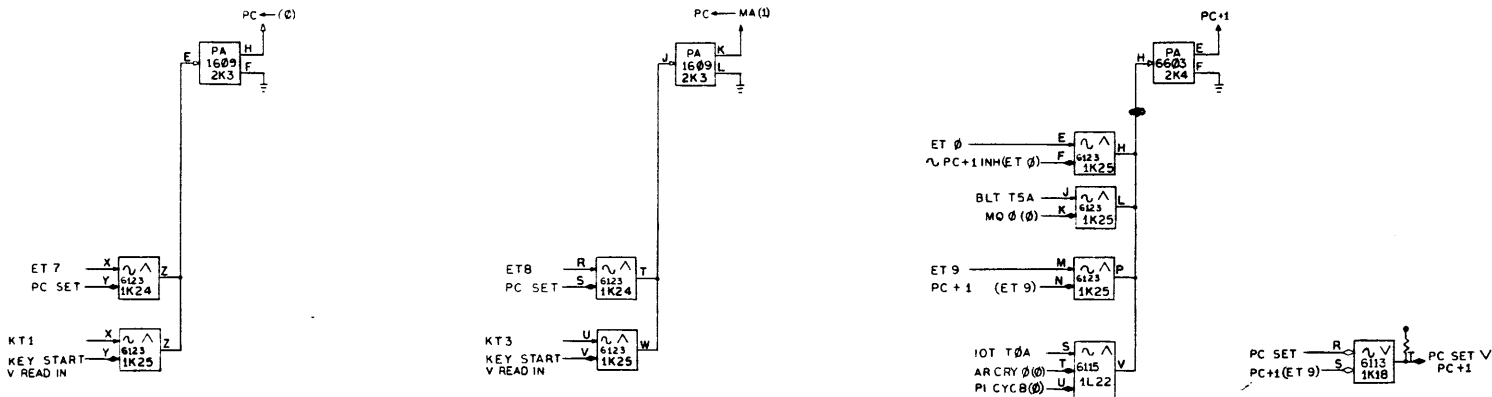
- 1. ALL FF PACKAGES ARE 6204.
- 2. * INDICATES REAR CONNECTOR PIN.
- 3. GROUND PIN D AND Z IN ALL 6204'S.

A

A

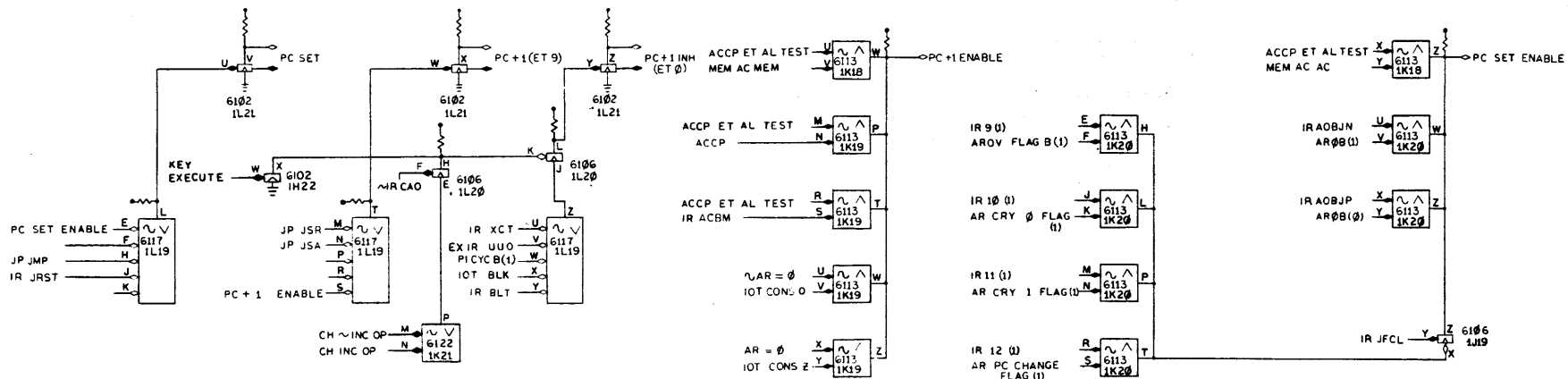
B

B



C

C



D

A

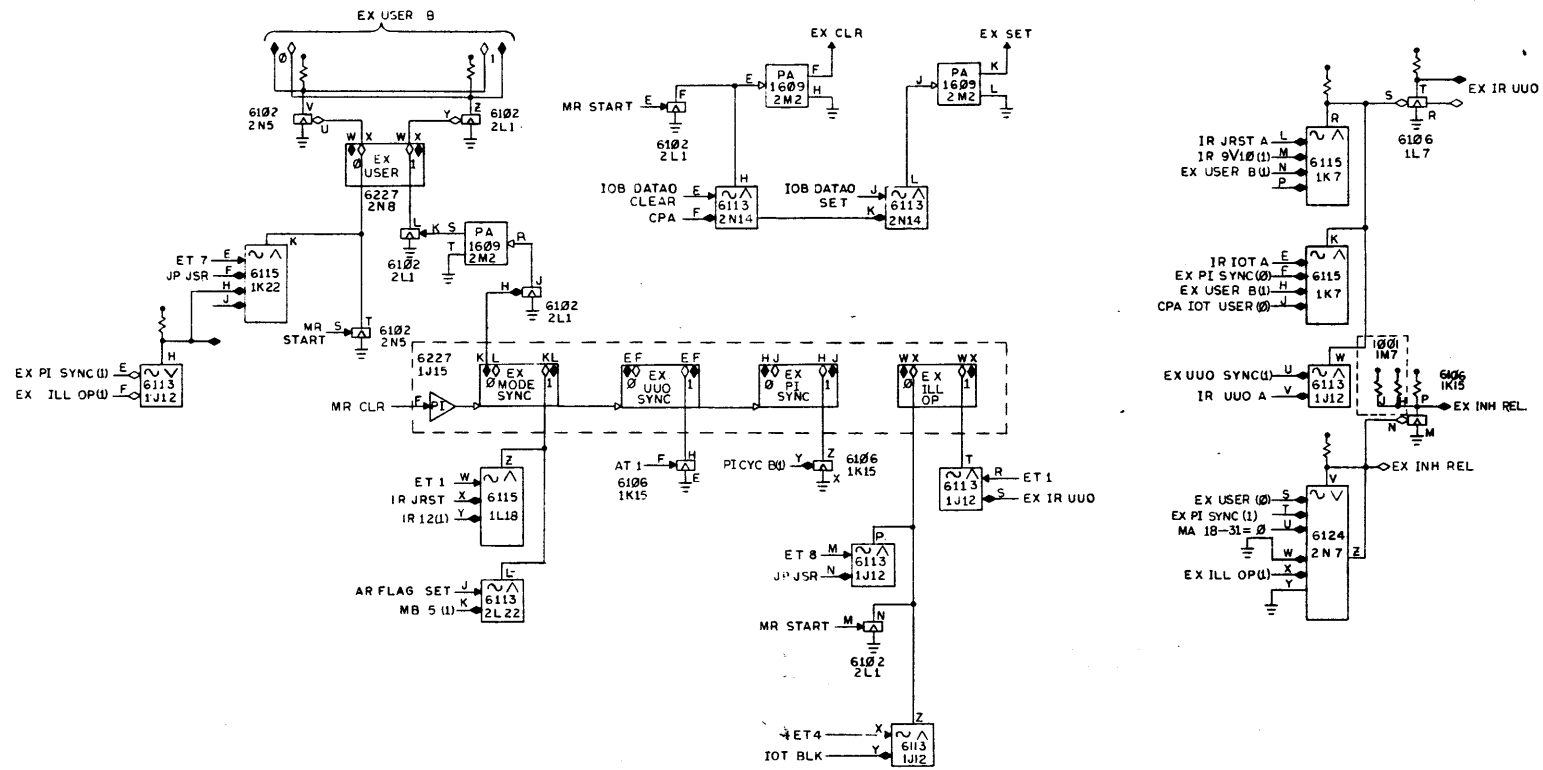
B

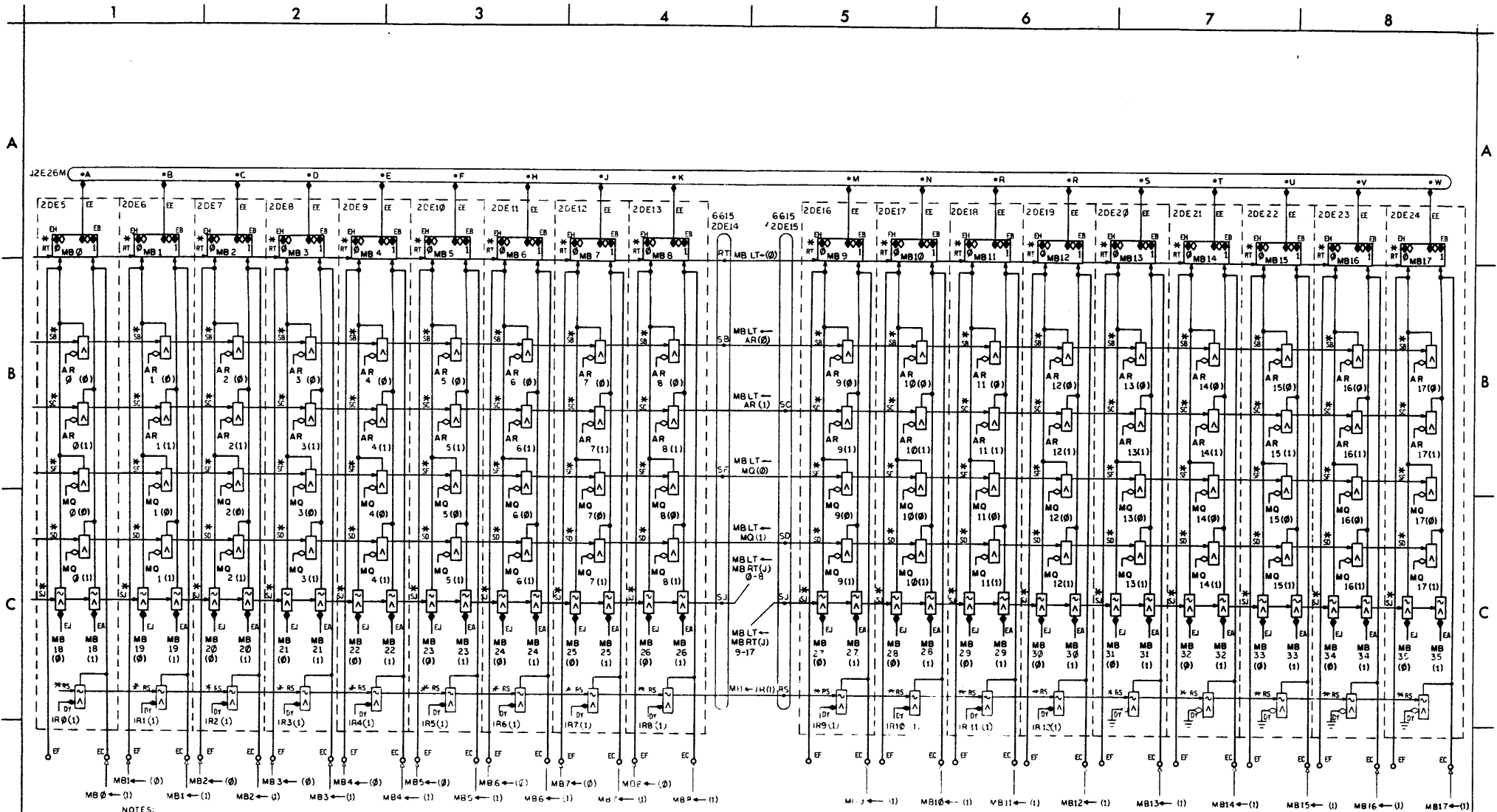
C

D

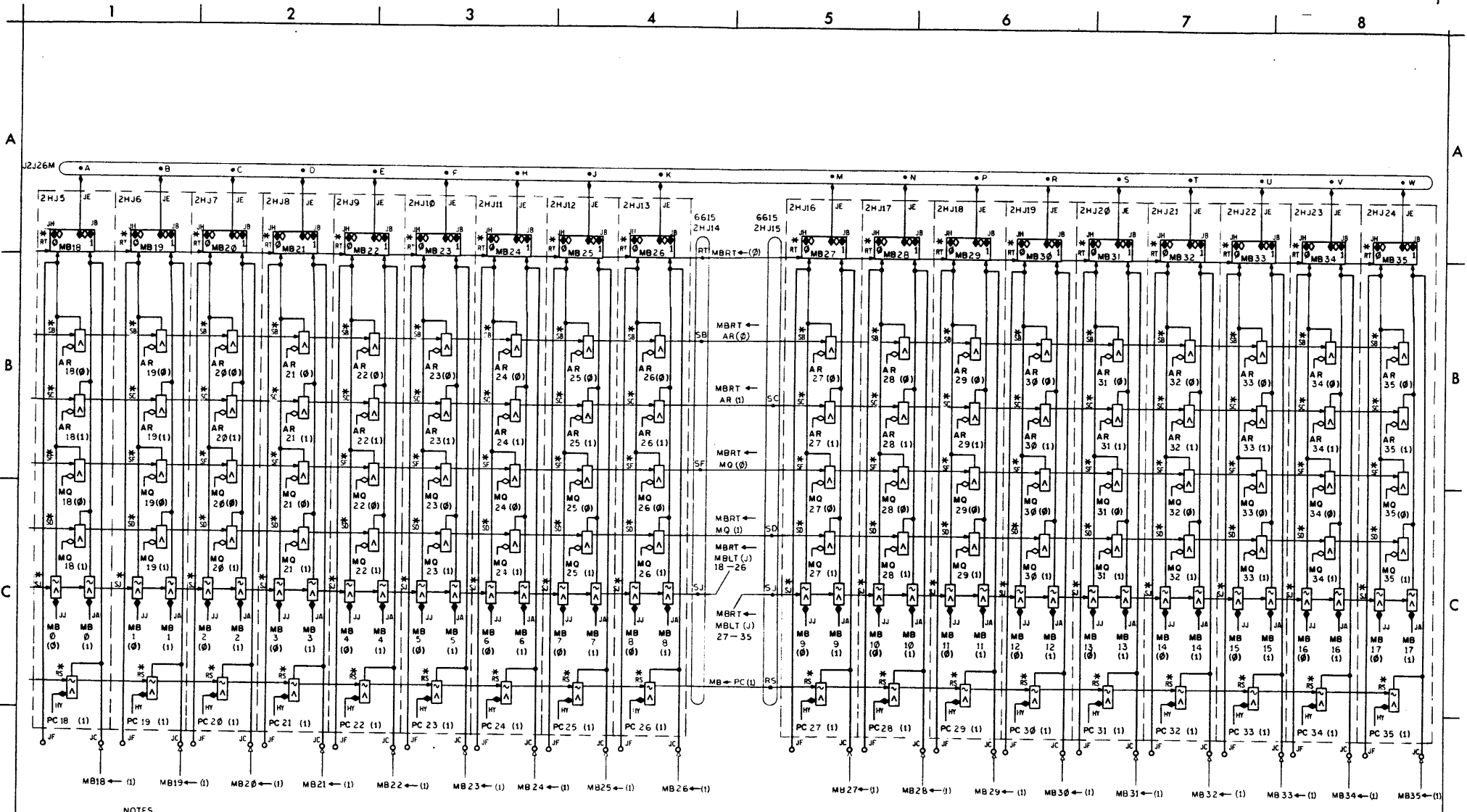
B

C



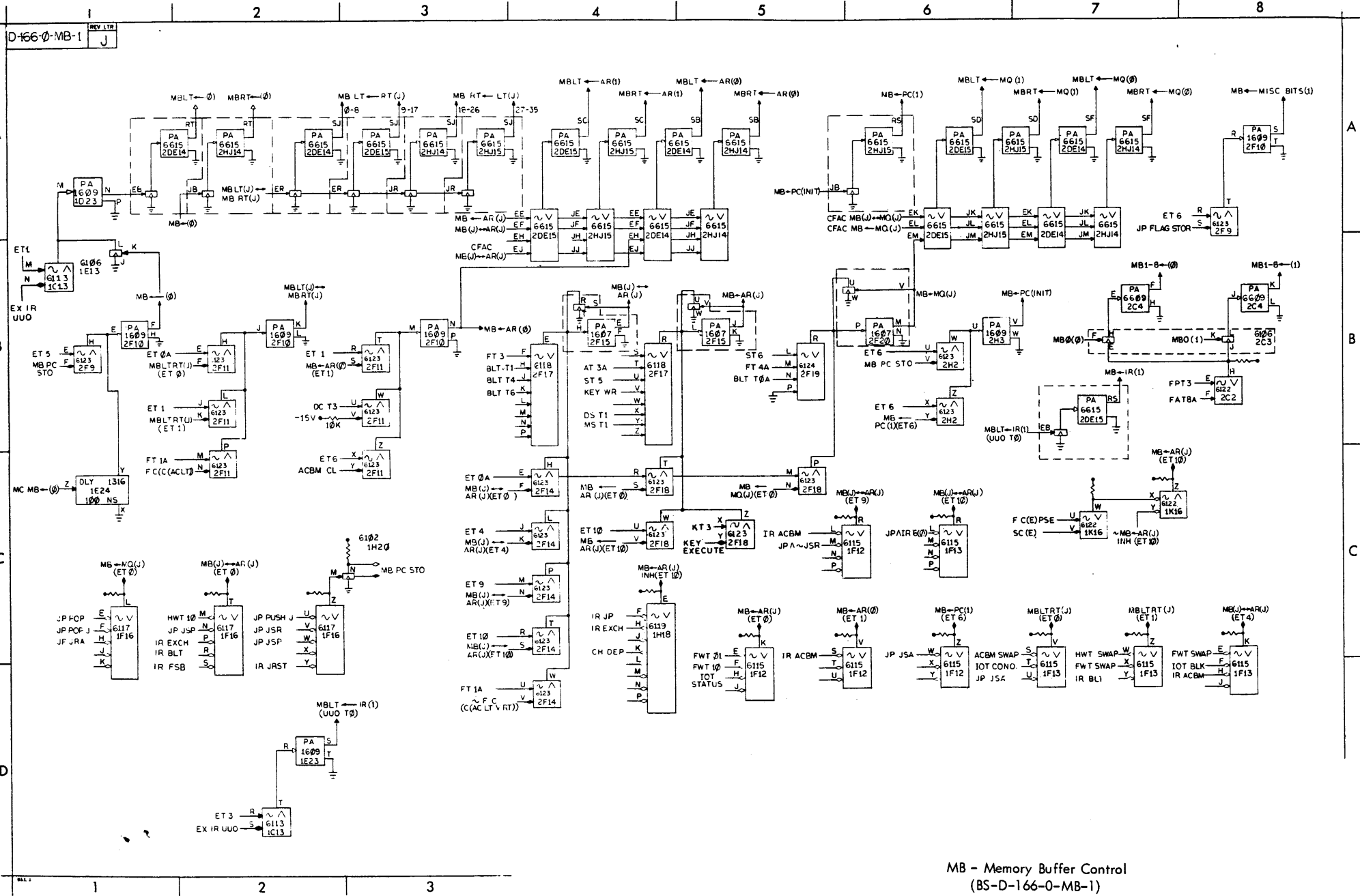


- NOTES:
 1. ALL FF PACKAGES ARE 6205
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN D,P AND Z IN ALL 6205'S AND 6615..

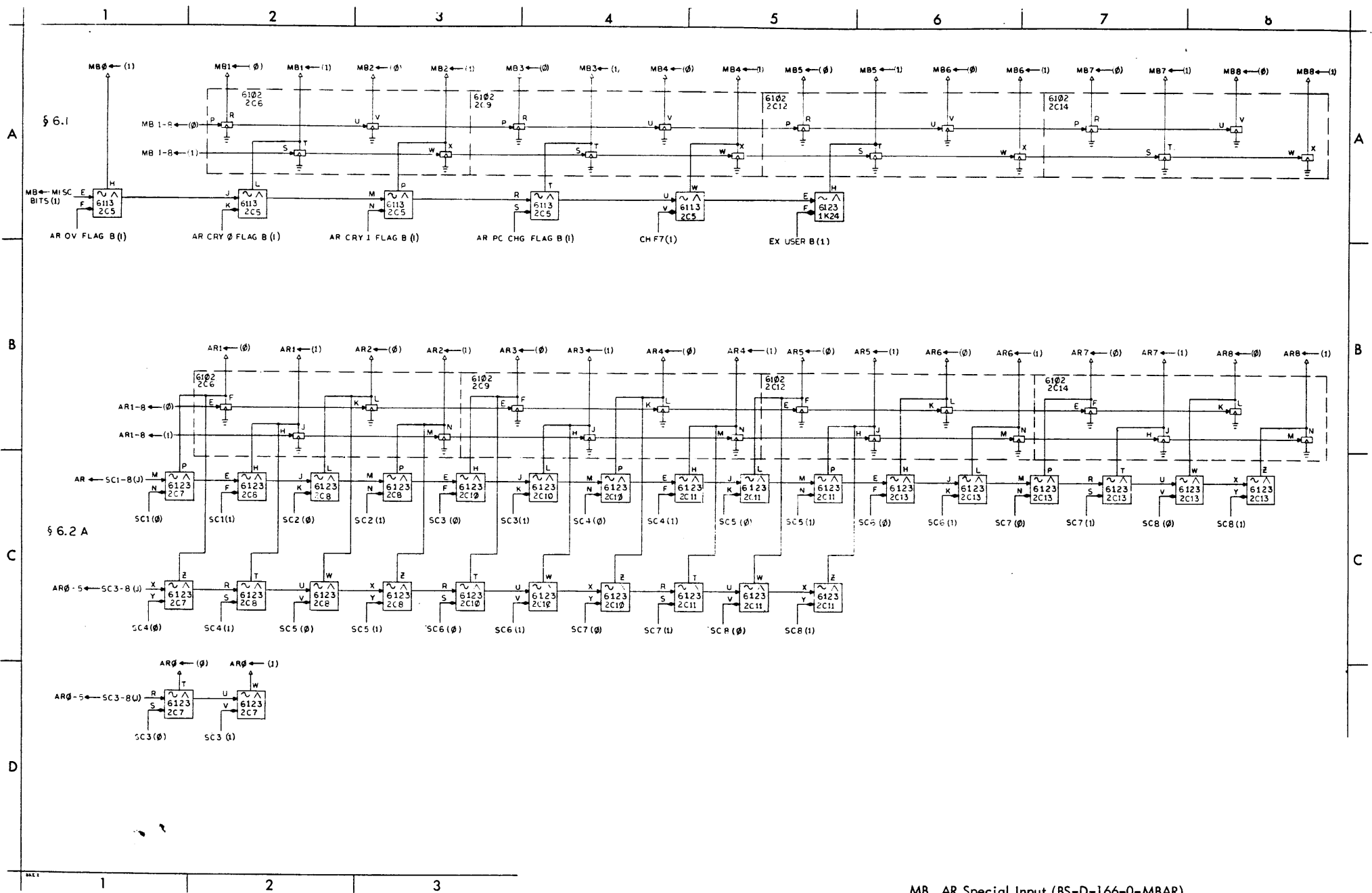


NOTES

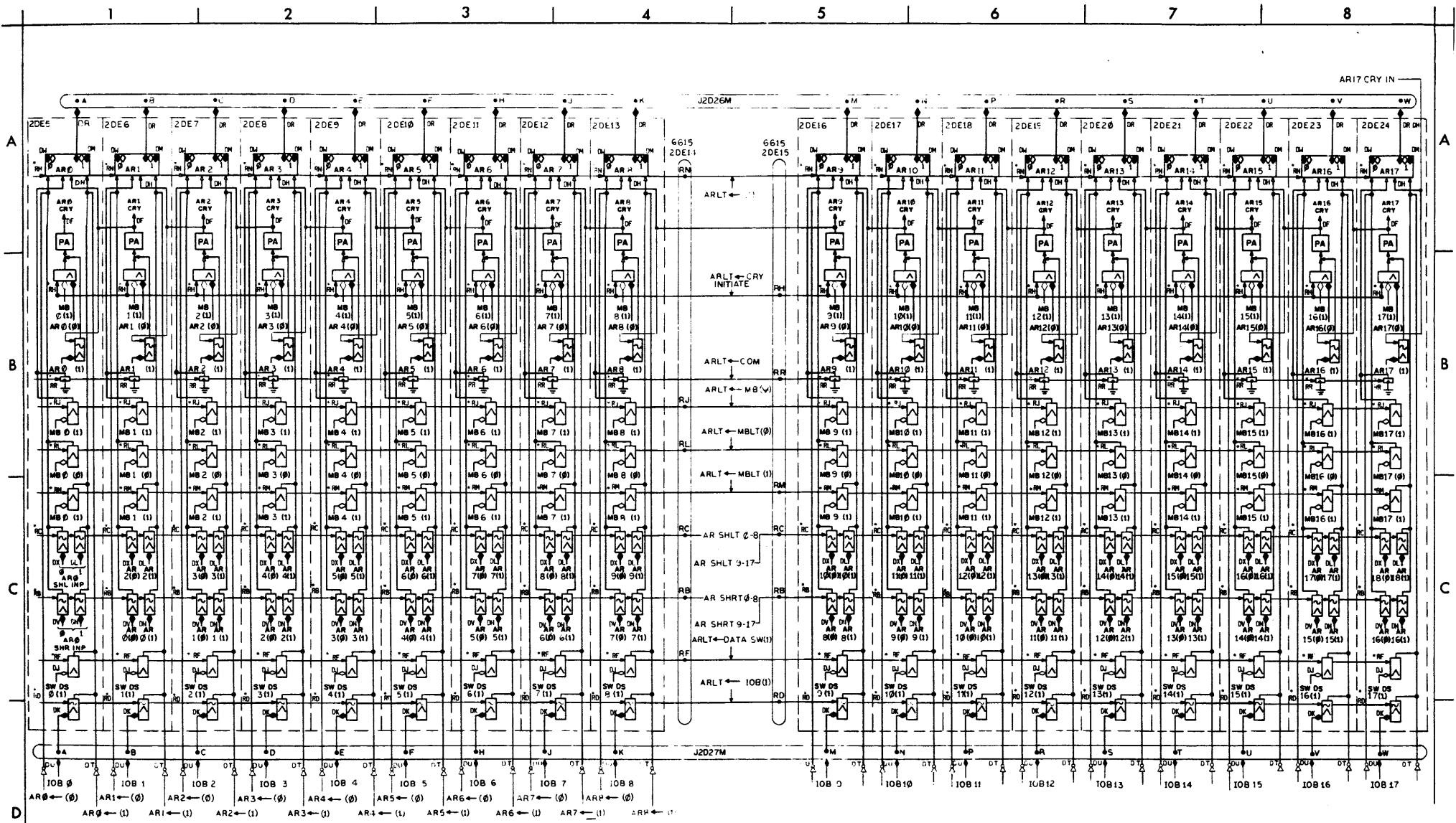
1. ALL FF PACKAGES ARE 6205
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D, P AND Z IN ALL 6205S AND 6615S.



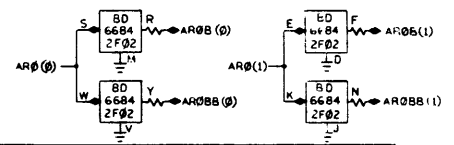
MB - Memory Buffer Control
 (BS-D-166-0-MB-1)



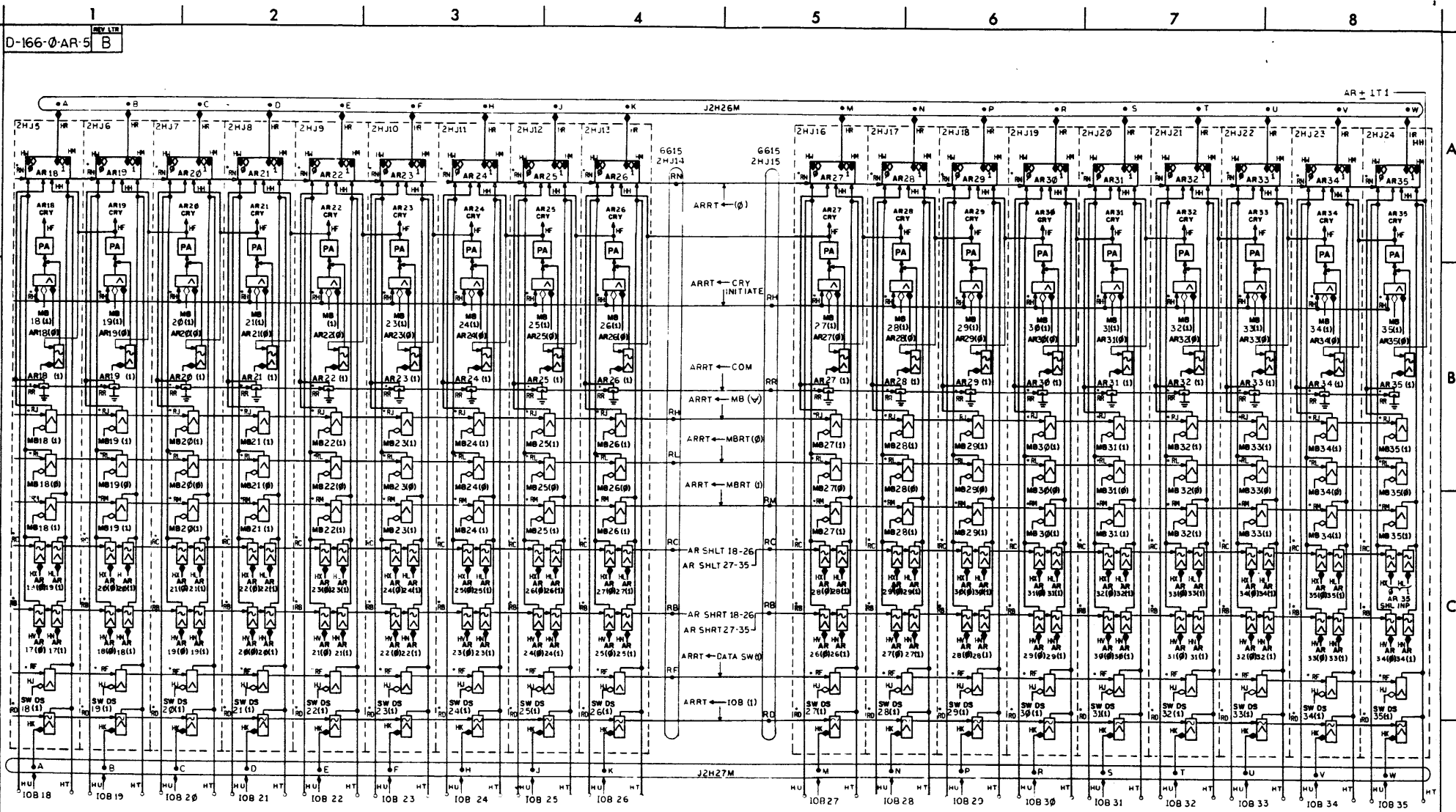
MB, AR Special Input (BS-D-166-0-MBAR)



- NOTE
1. * INDICATES FEED DIRECTION PIN
 2. ALL FF PACKAGES ARE 6205
 3. GROUND PINS C AND Z IN ALL 6205 AND 6615.

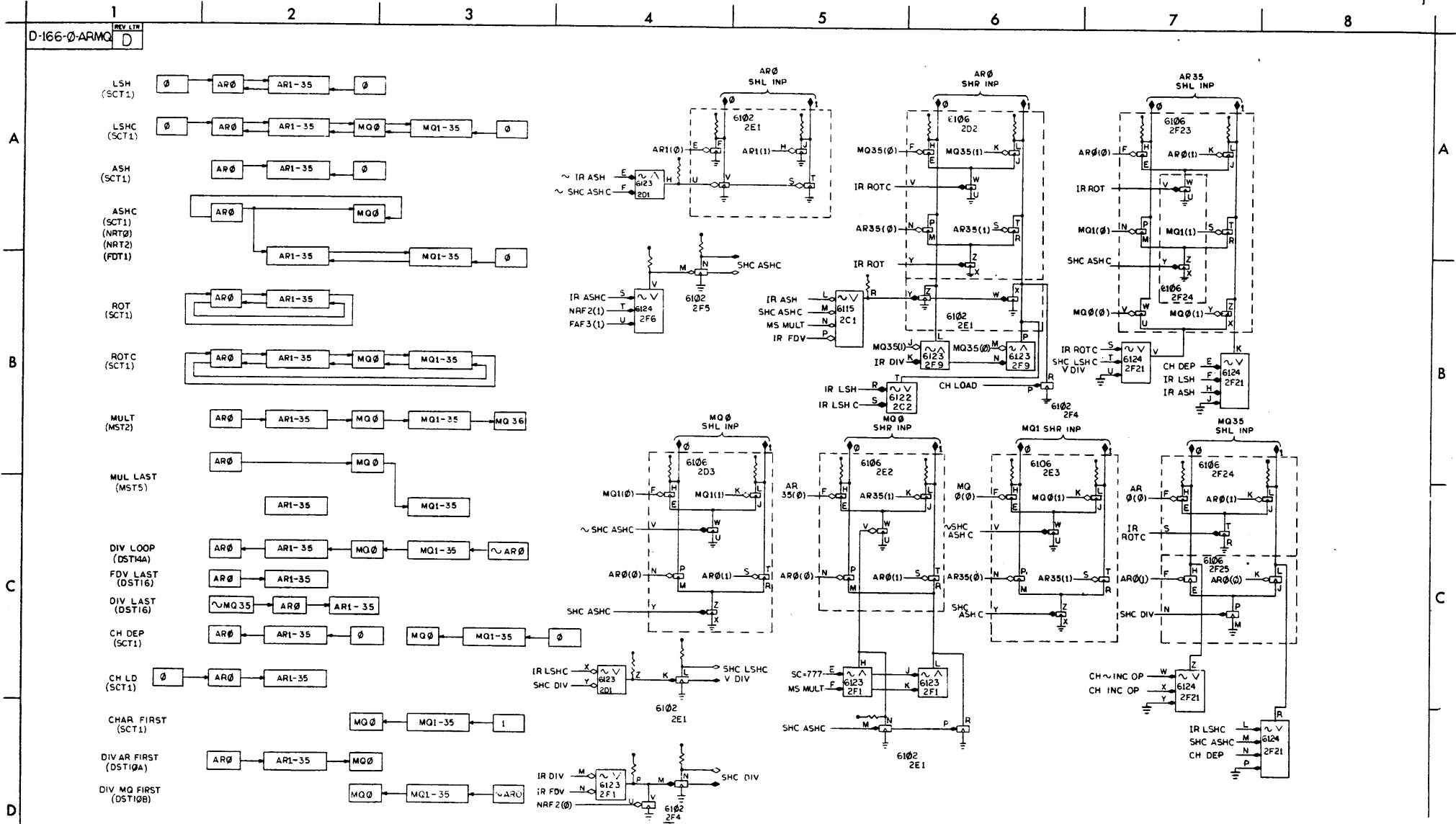


ARLT Register 0-17 (BS-D-166-0-AR-4)

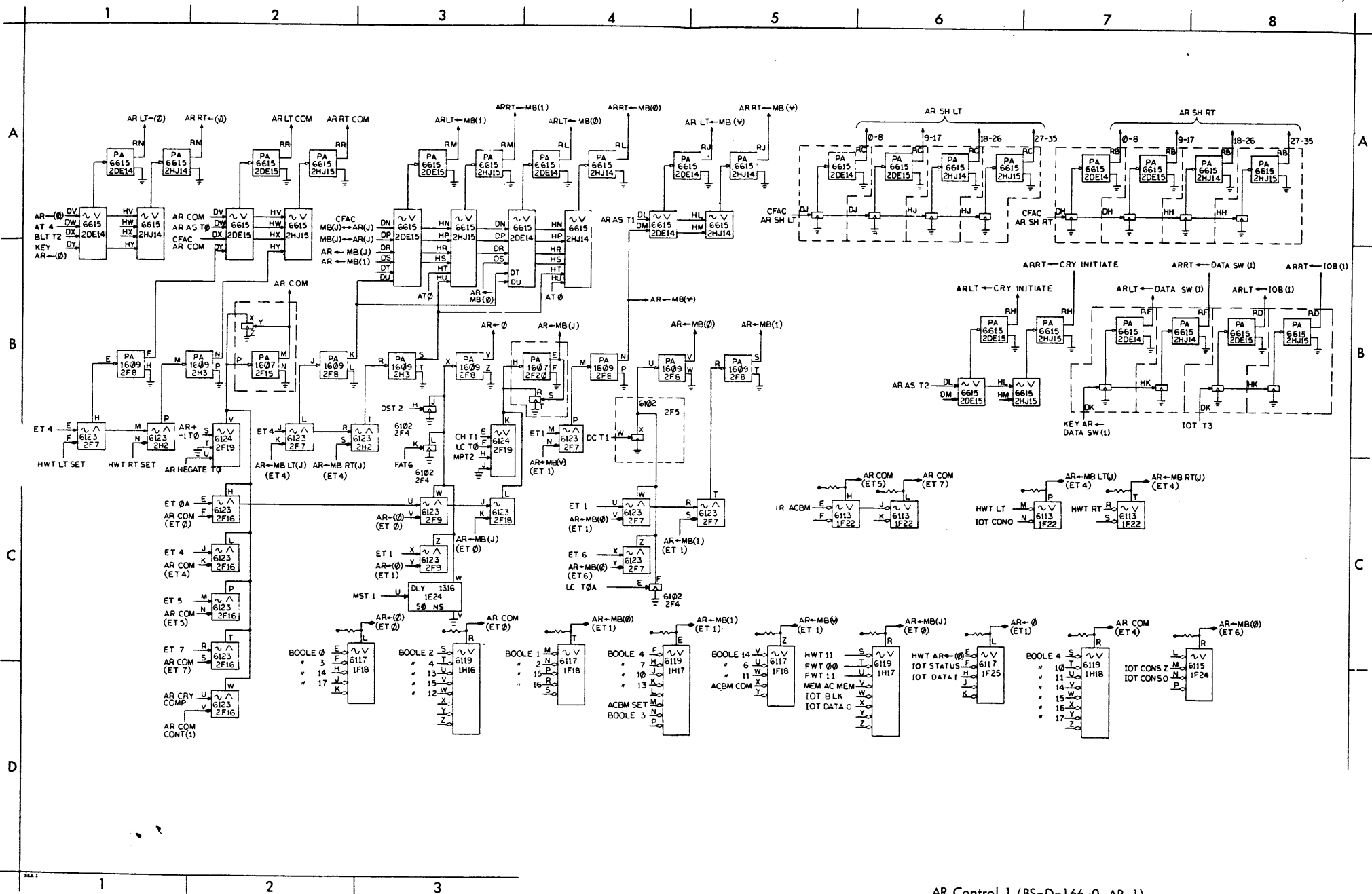


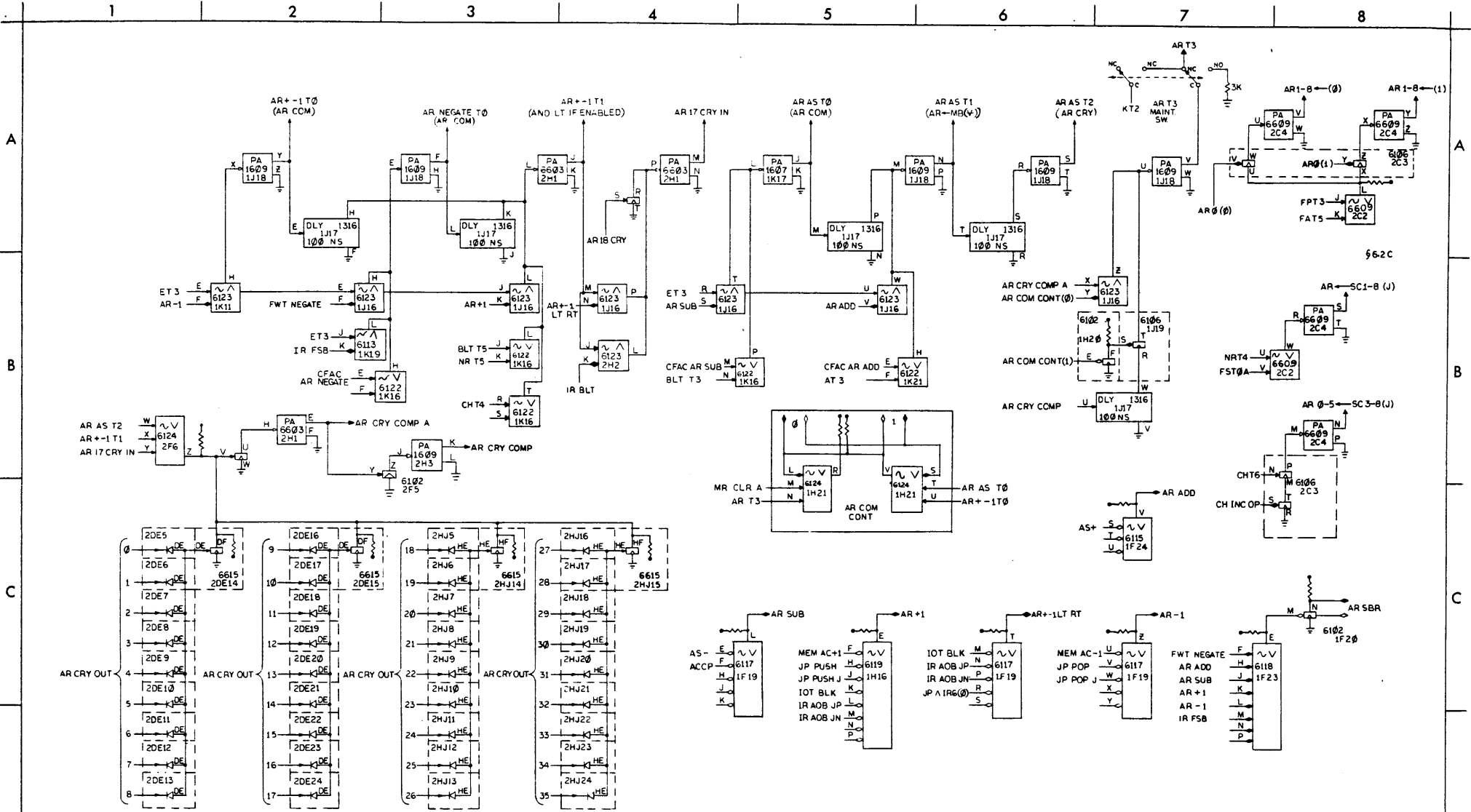
NOTES:

1. ALL FF PACKAGES ARE 6205
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 6205 AND 6615.



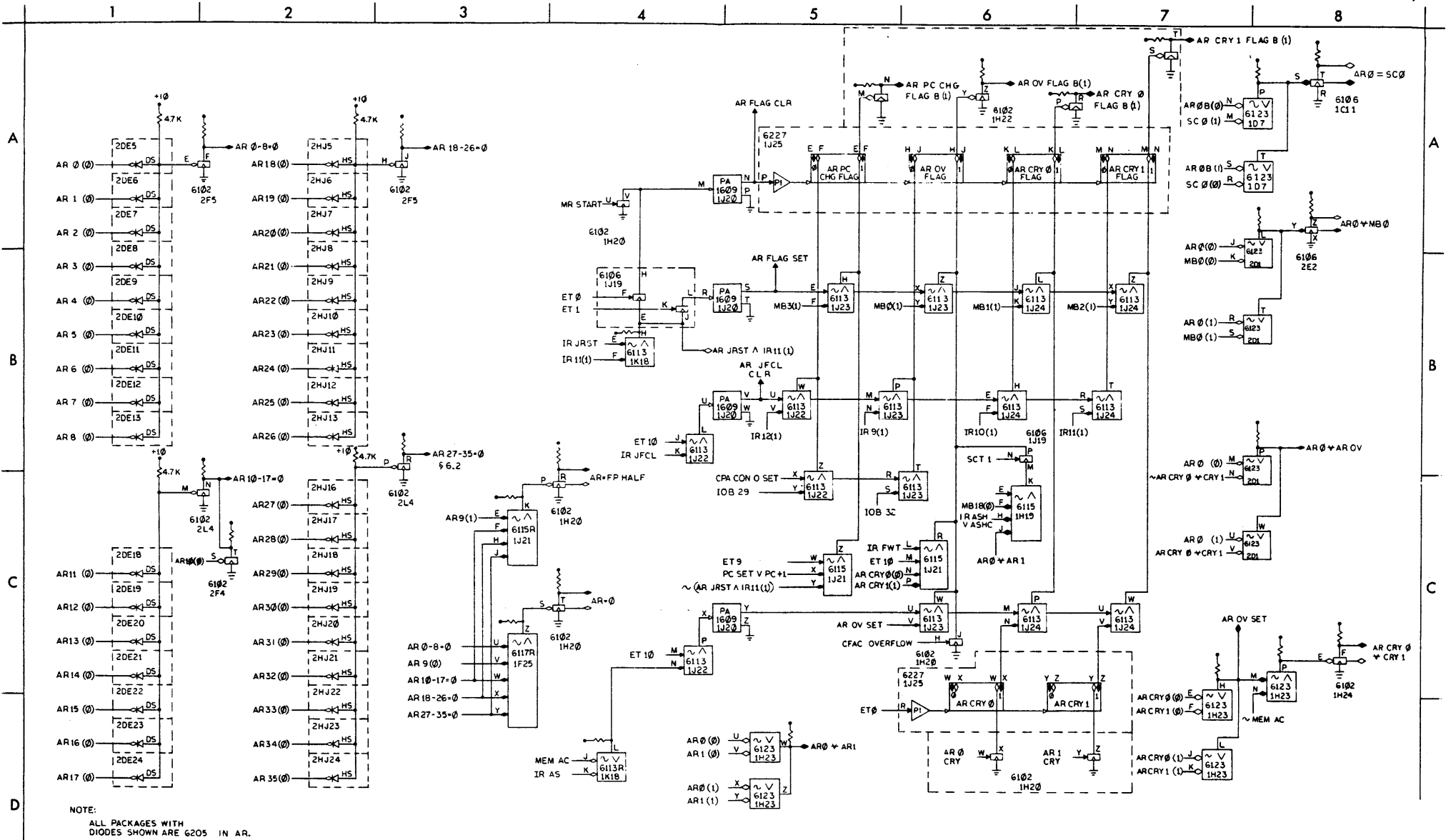
AR, MQ Shift Connections (BS-D-166-0-ARMQ)



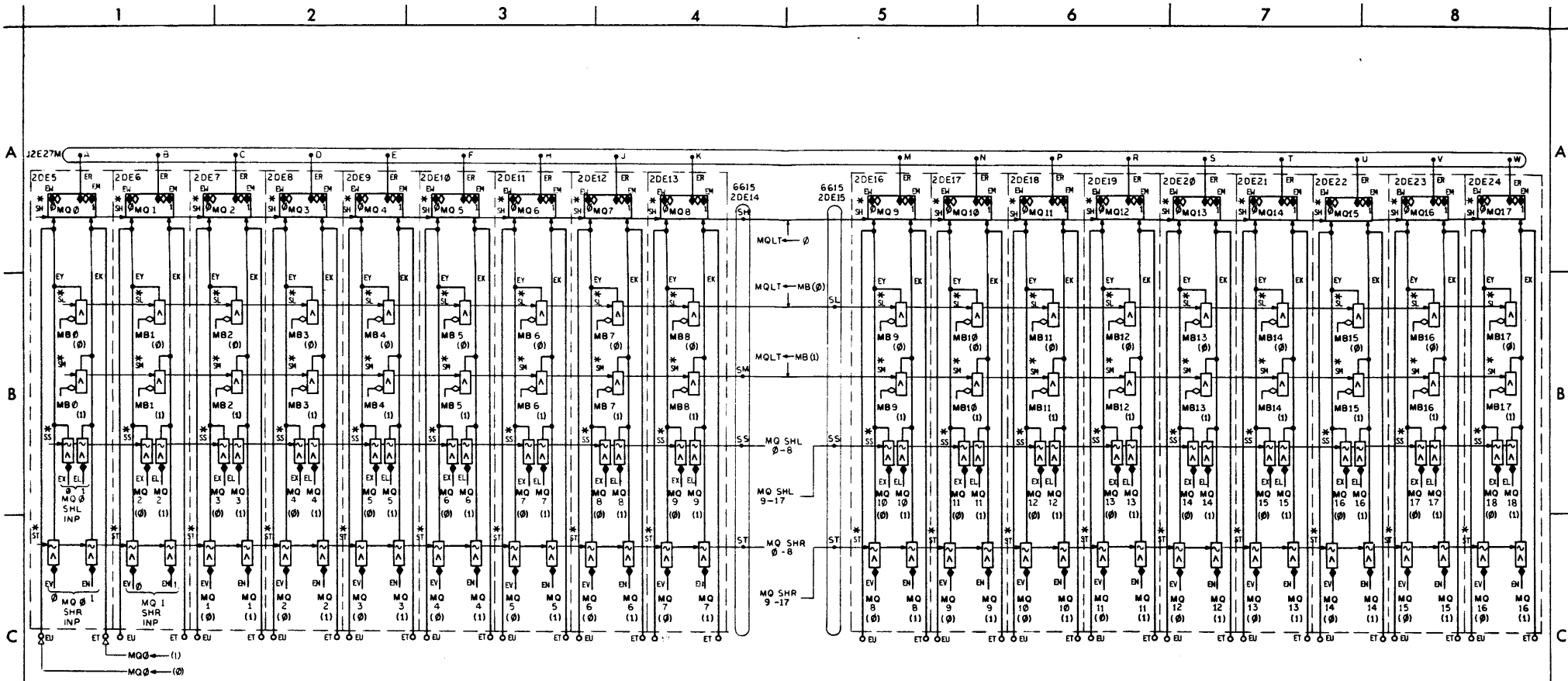


NOTE:
ALL PACKAGES WITH
DIODES SHOWN ARE 6205 IN AR.

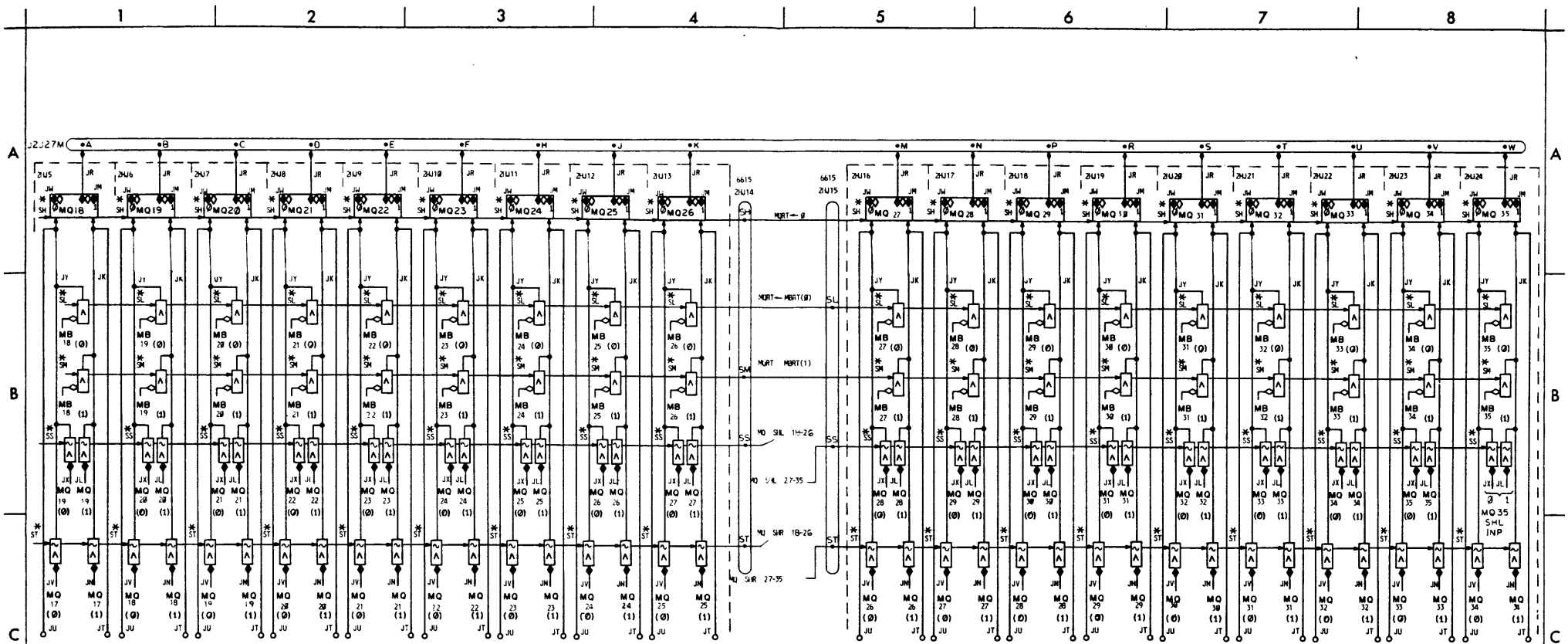
AR CRY COMPLETION
DETECTION



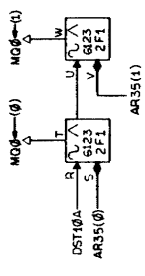
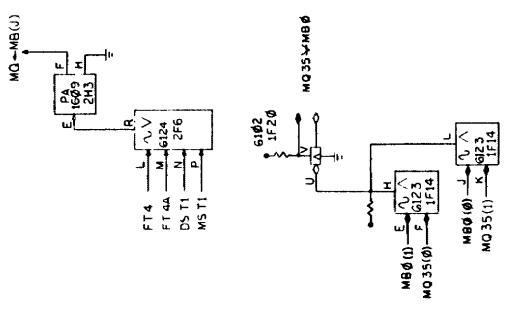
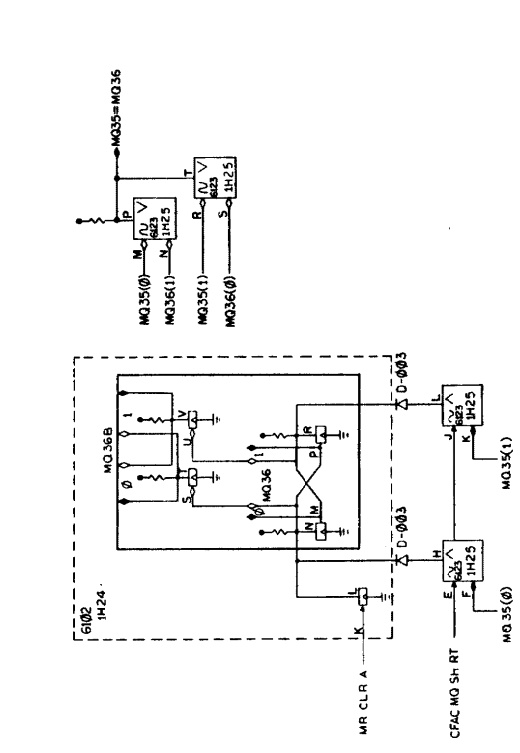
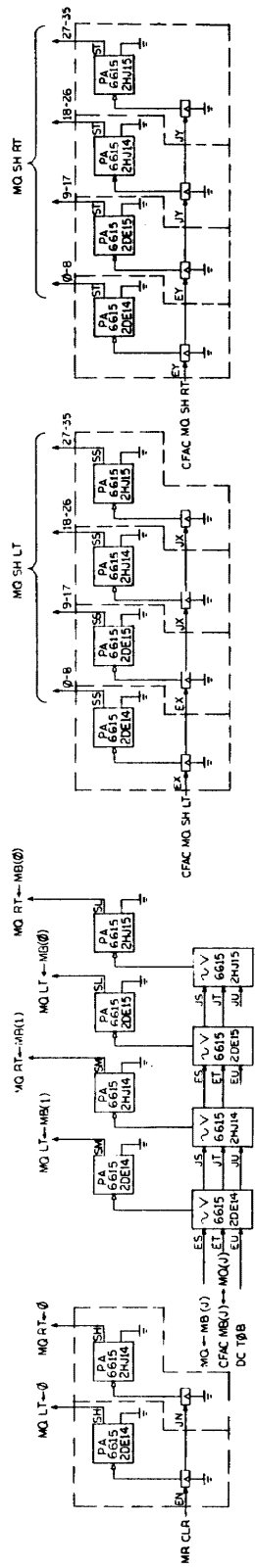
AR Control 3 (BS-D-166-0-AR-3)

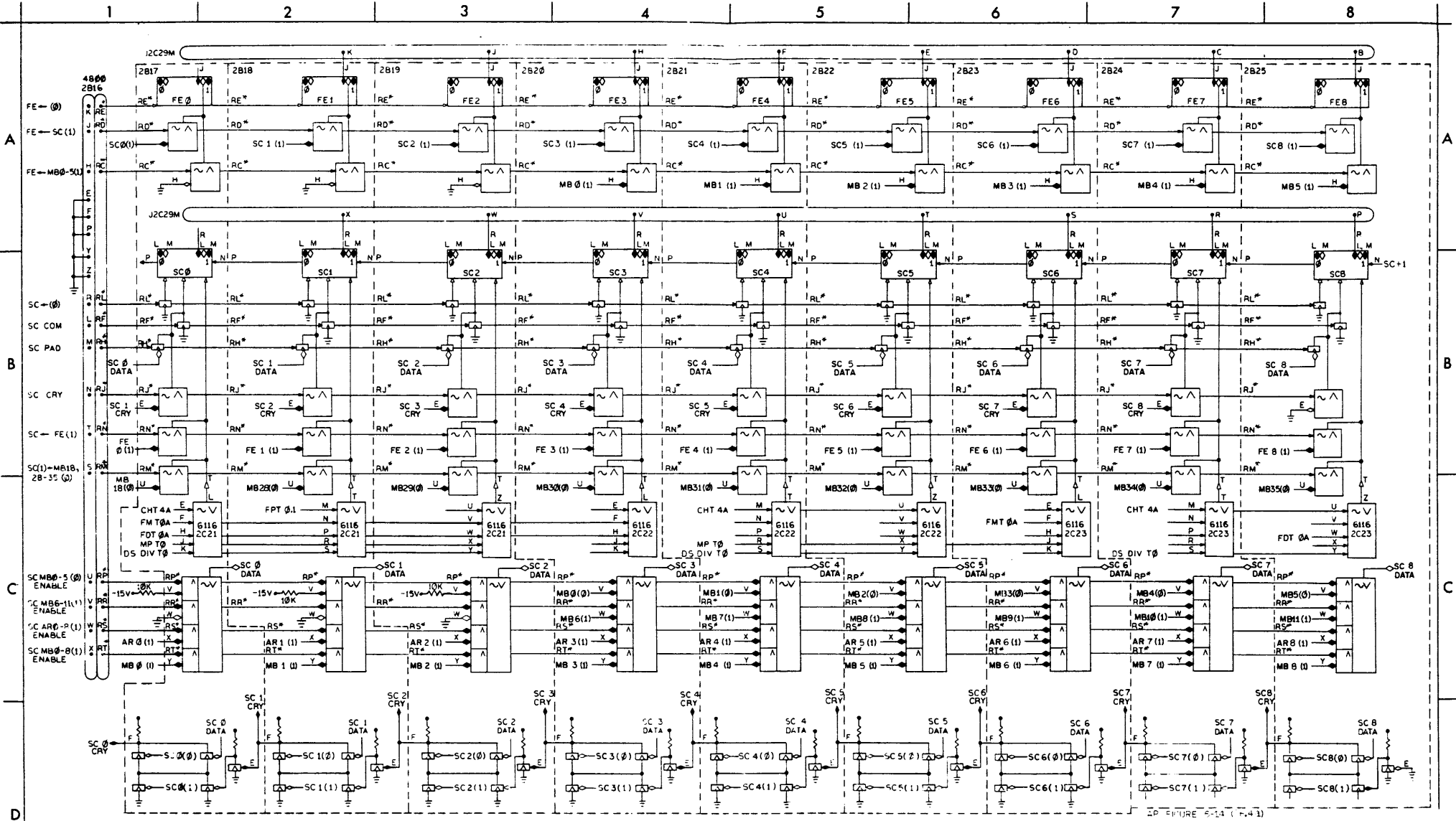


- NOTES:
1. ALL FF PACKAGES ARE 6205
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN DP AND Z IN ALL 6205'S AND 6615'S.



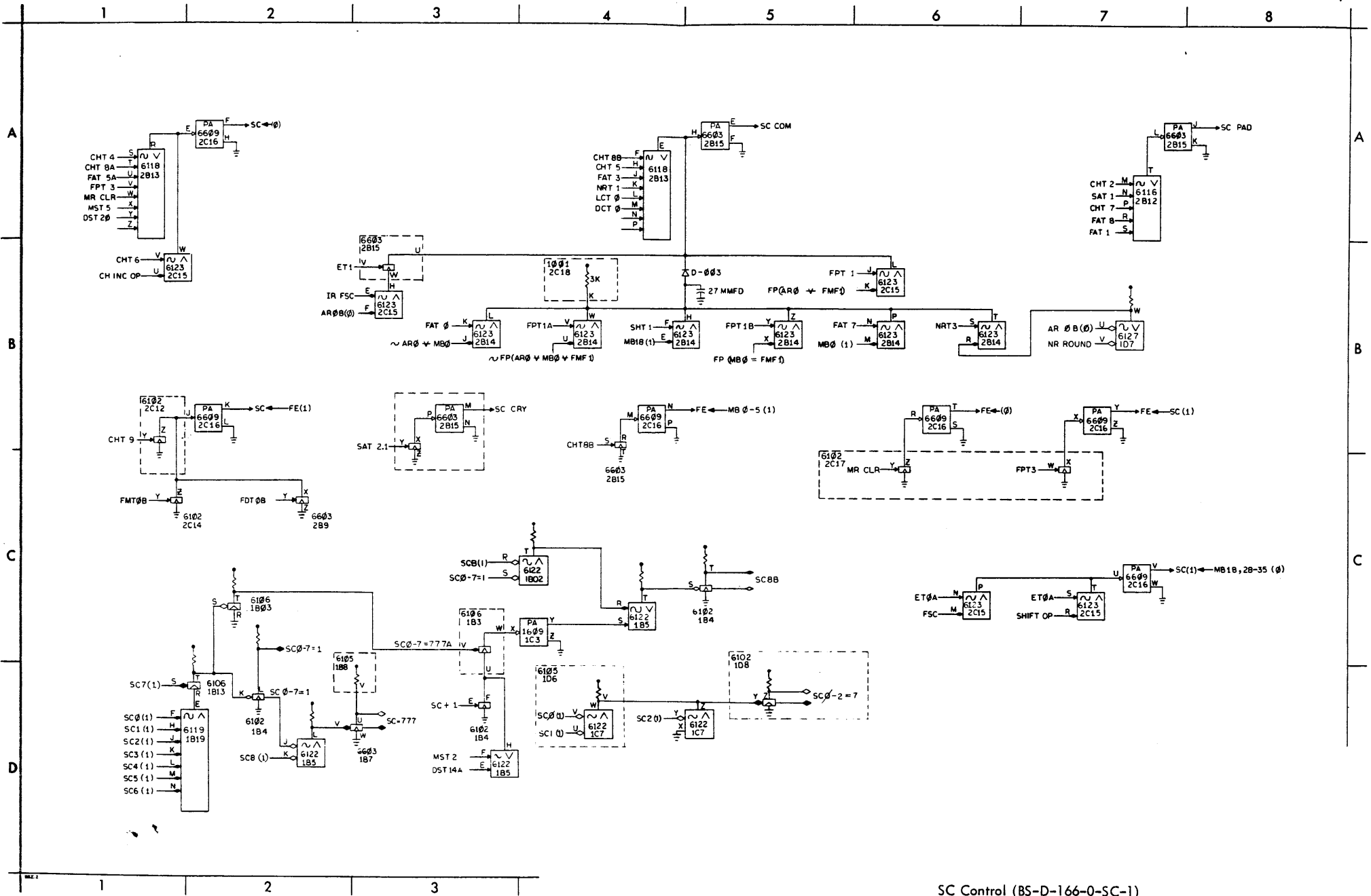
- NOTE:
1. ALL FF PACKAGES ARE 6205.
 2. * INDICATES REAR CONNECTOR PIN.
 3. GROUND PIN D,P AND Z IN ALL 6205'S AND 6615'S.



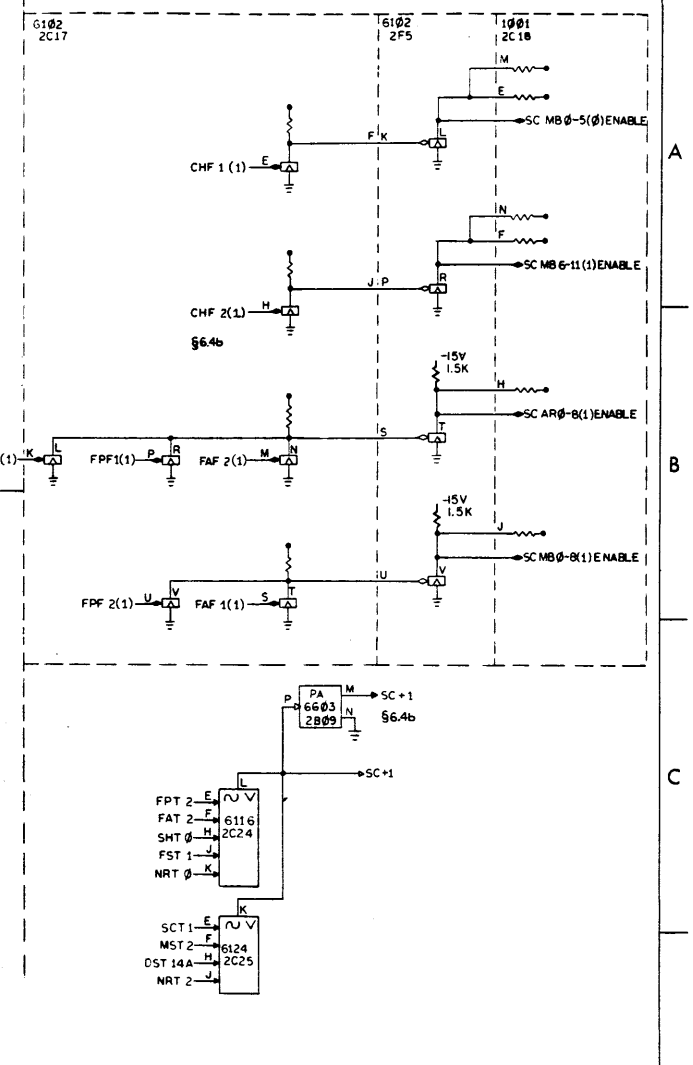
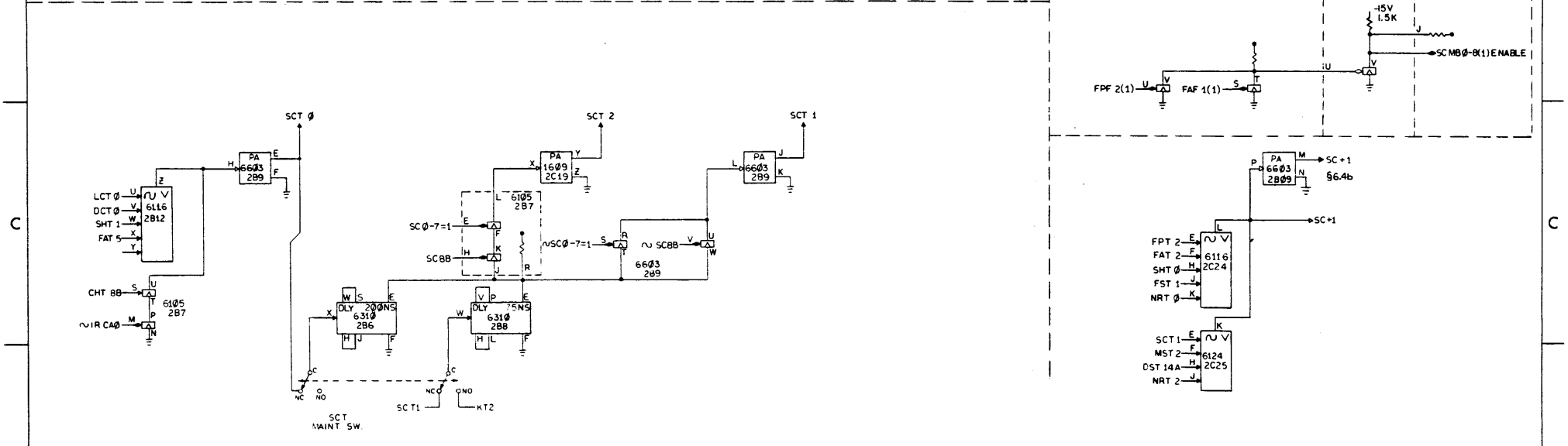
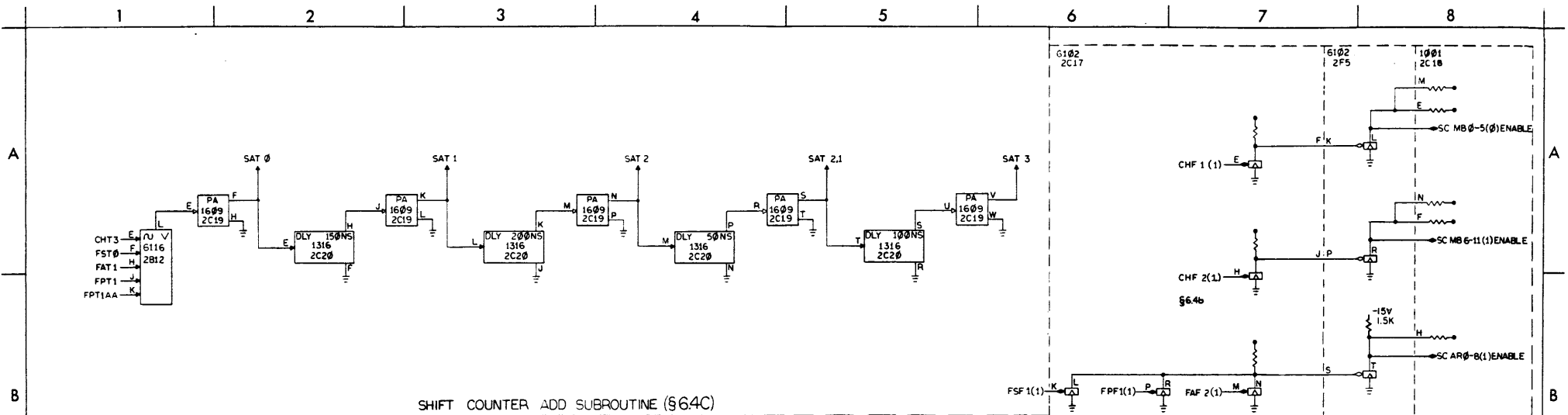


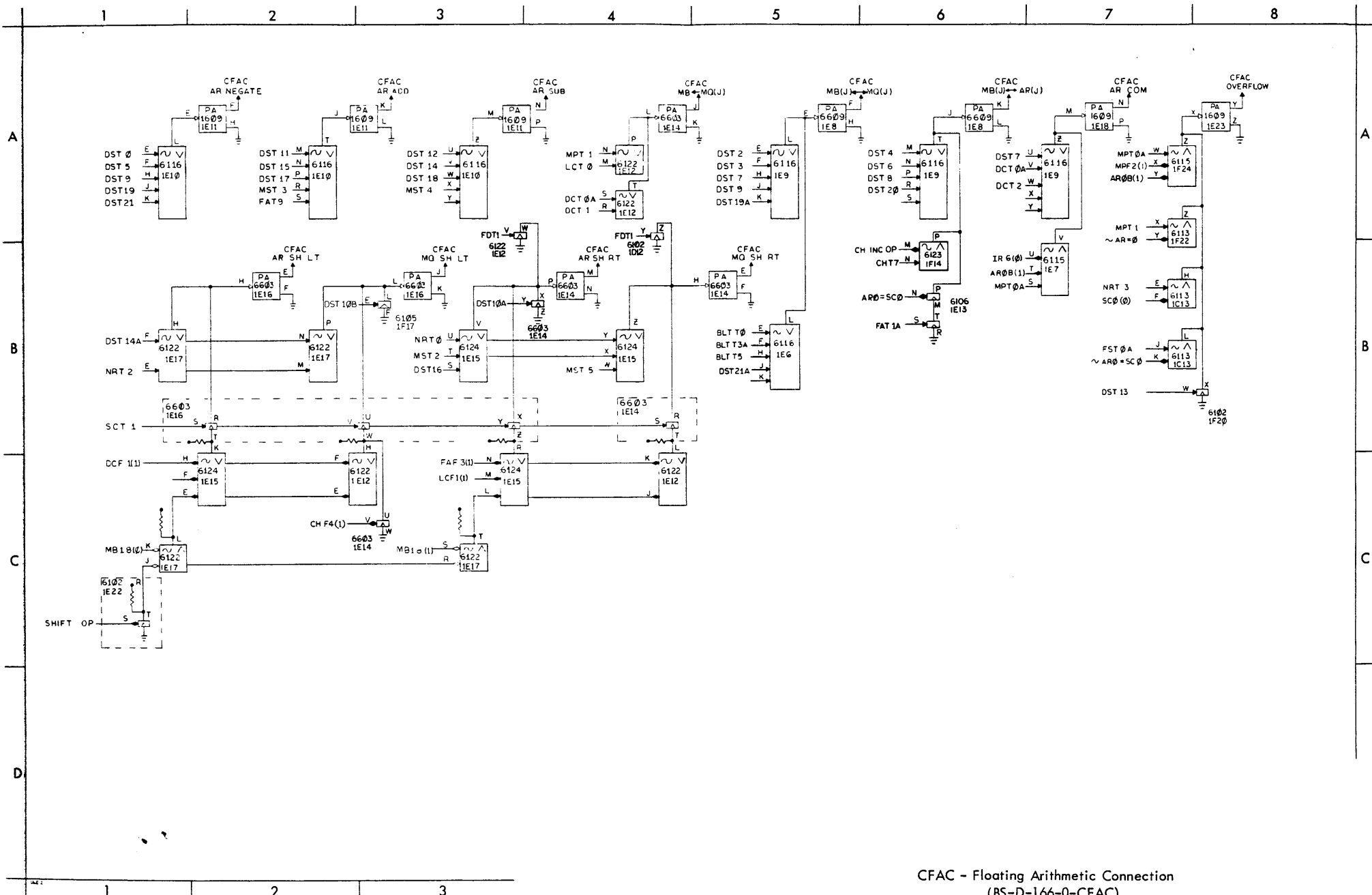
- NOTE:
1. UNLESS OTHERWISE INDICATED ALL PACKAGES ARE 6203s.
 2. * INDICATES REAR CONNECTOR PIN.

SC, FE - Shift Counter, Floating Exponent
(BS-D-166-0-SCFE)

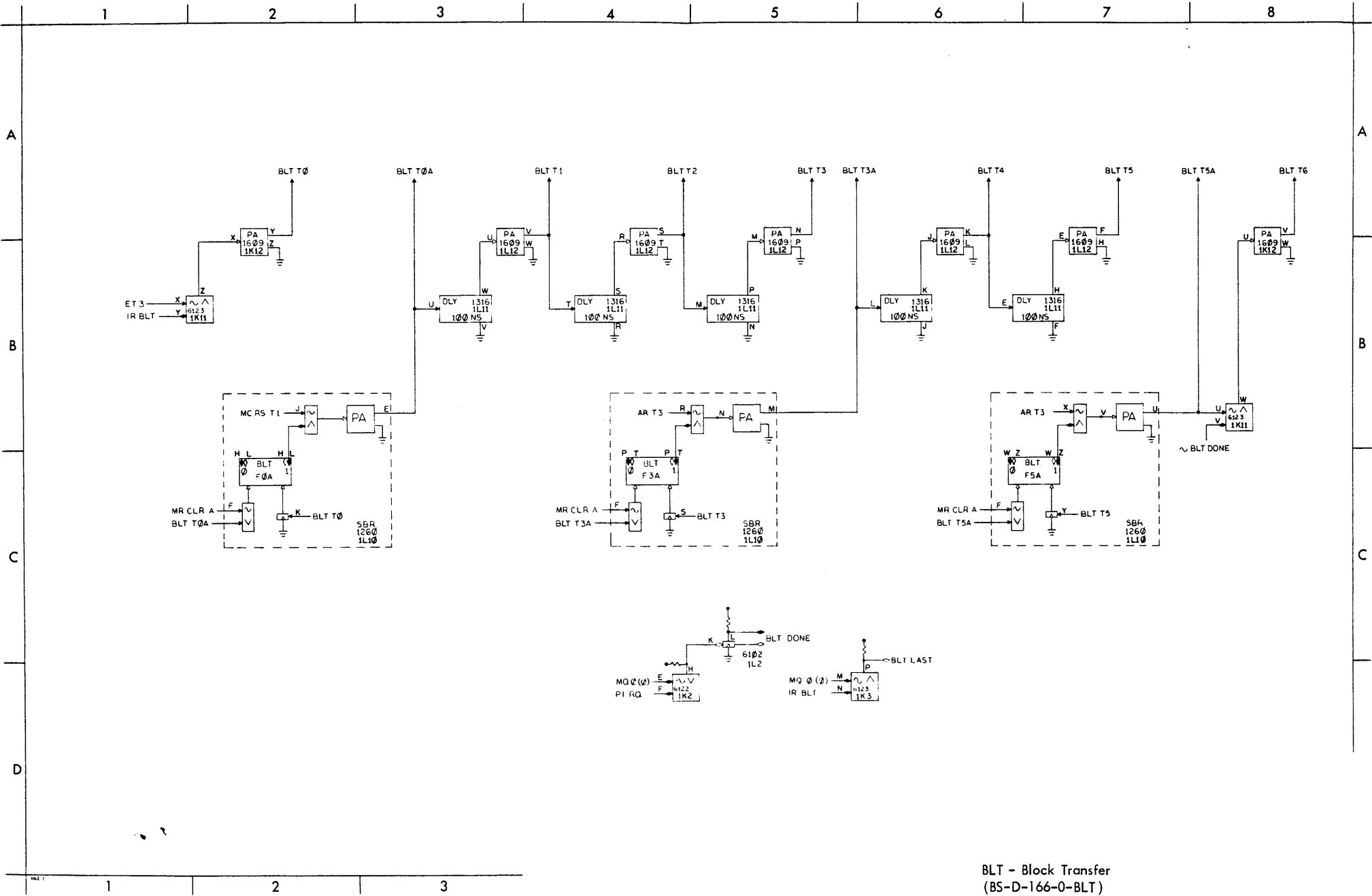


SC Control (BS-D-166-0-SC-1)

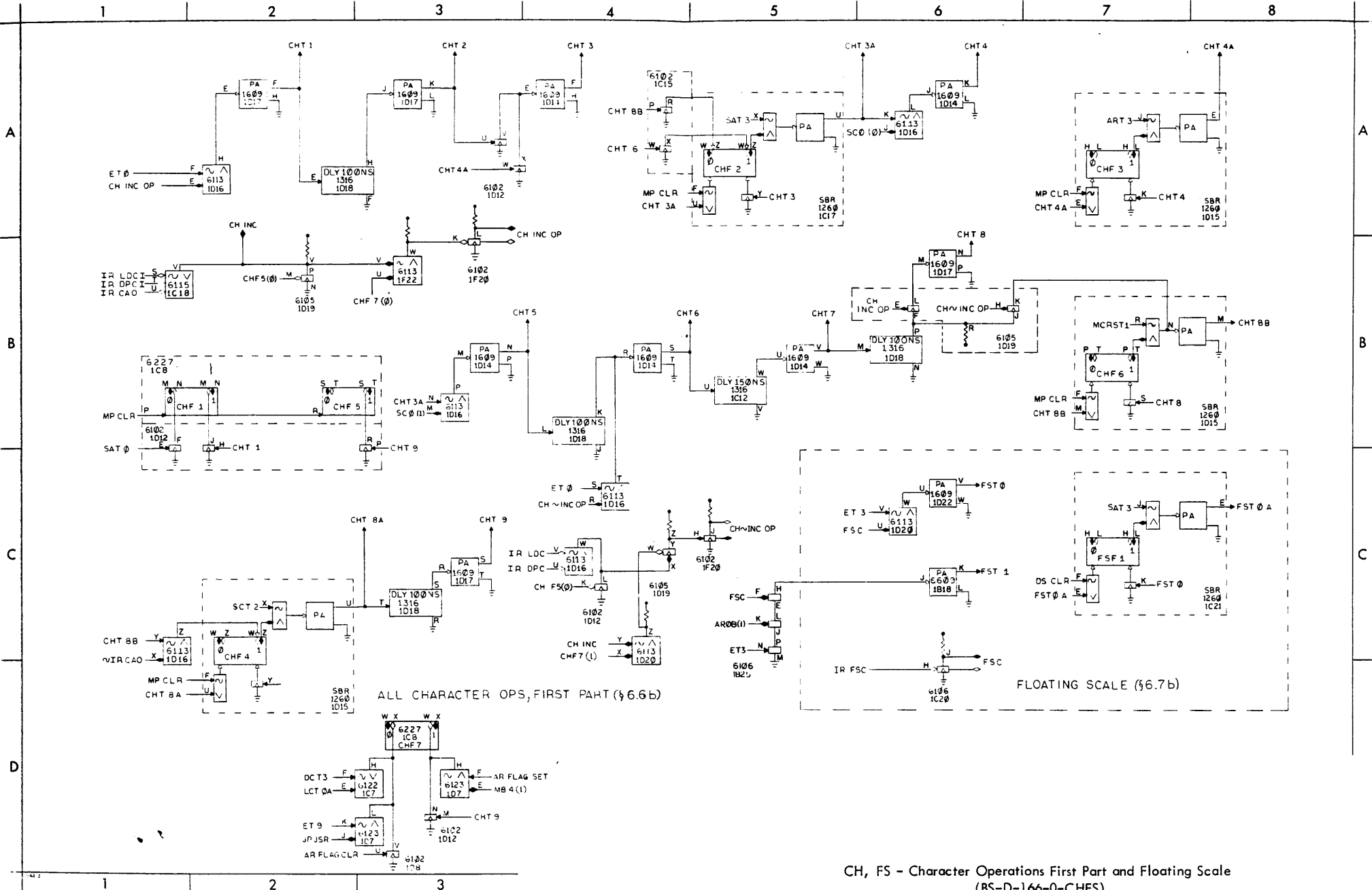




CFAC - Floating Arithmetic Connection
(BS-D-166-0-CFAC)



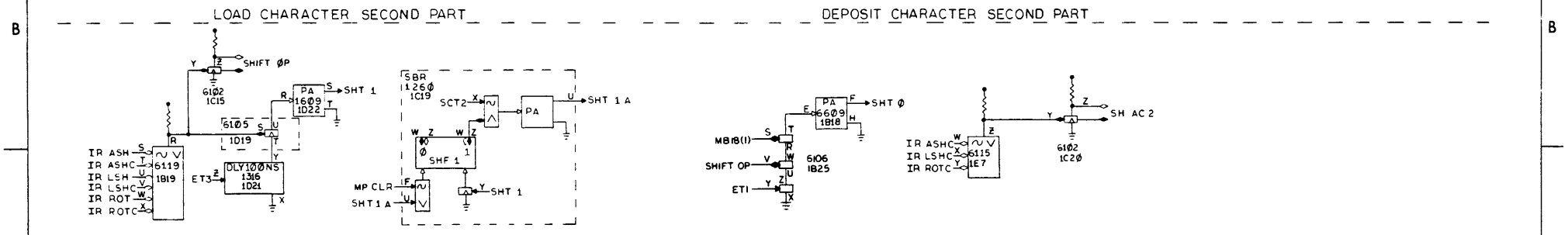
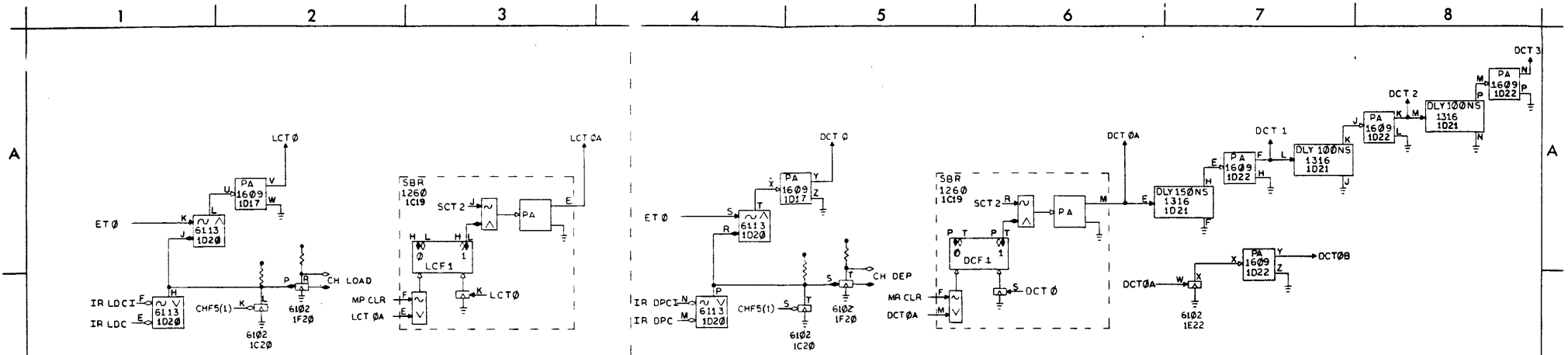
BLT - Block Transfer
(BS-D-166-0-BLT)



ALL CHARACTER OPS, FIRST PART (§6.6b)

FLOATING SCALE (§6.7b)

CH, FS - Character Operations First Part and Floating Scale (BS-D-166-0-CHFS)



LC, DC, SH - Character Operations Second Part
and Shift Operations
(BS-D-166-0-LDCS)

A

A

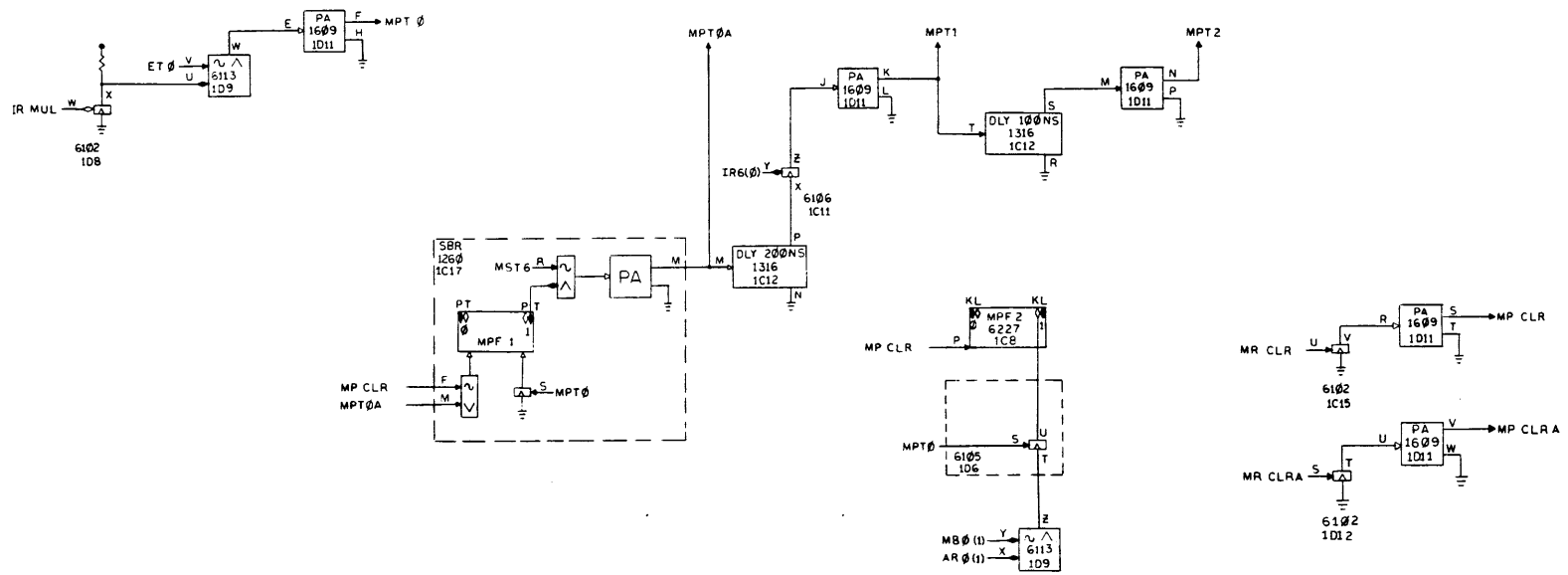
B

B

C

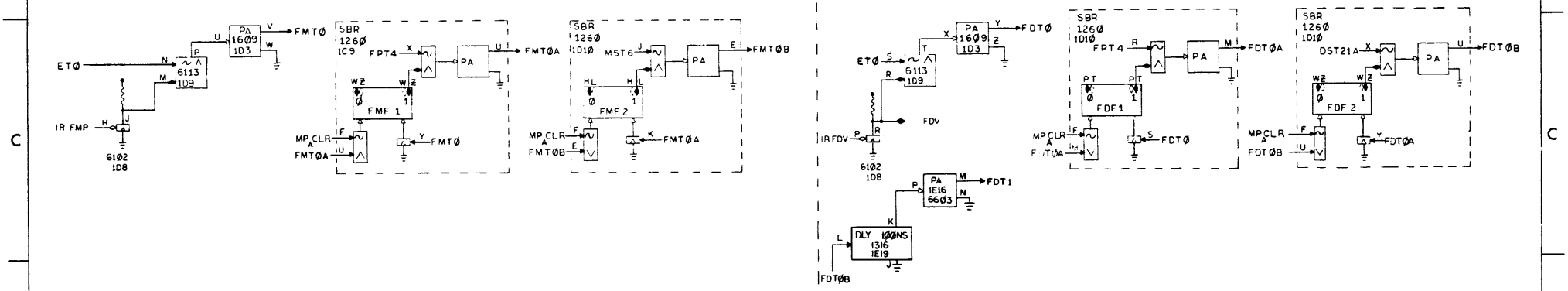
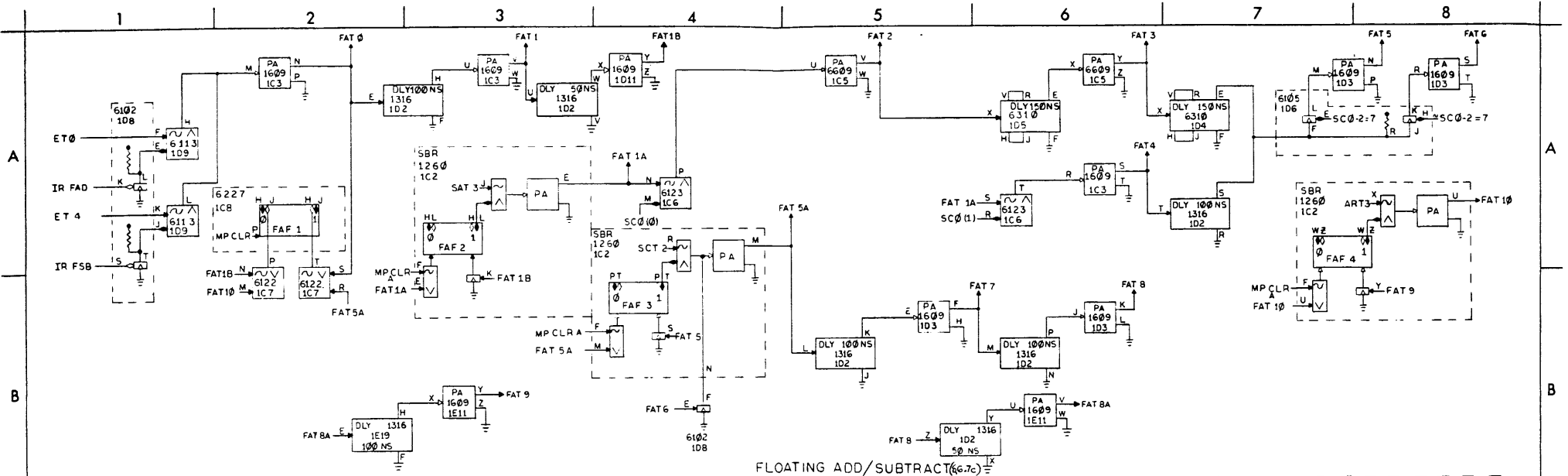
C

D

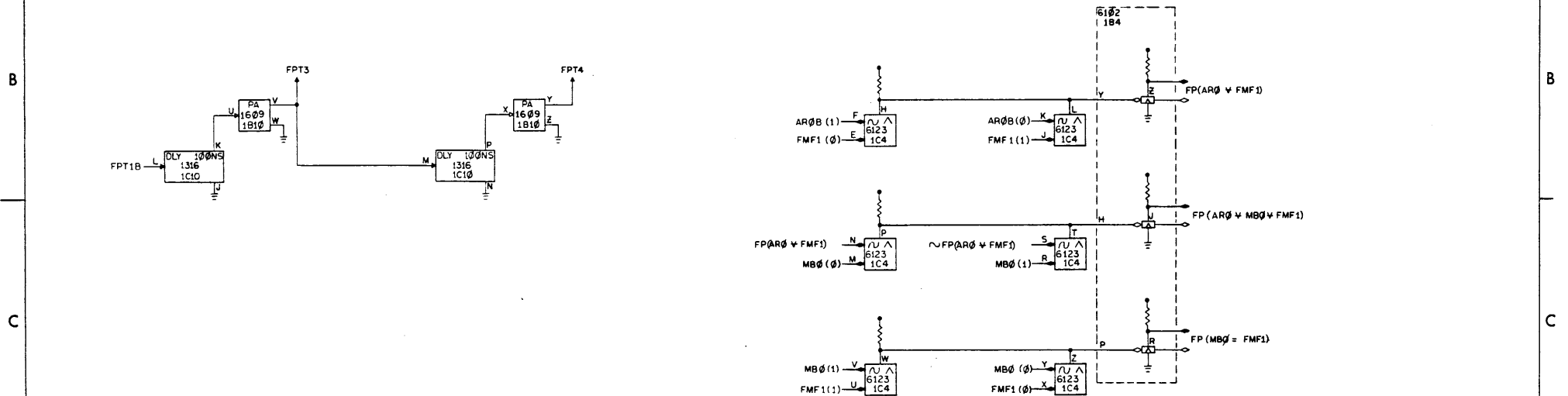
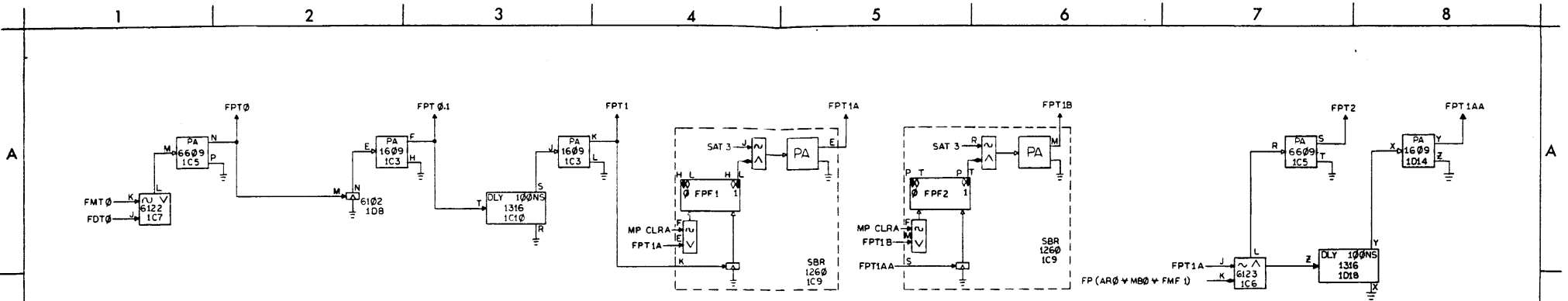


FIXED POINT MULTIPLY

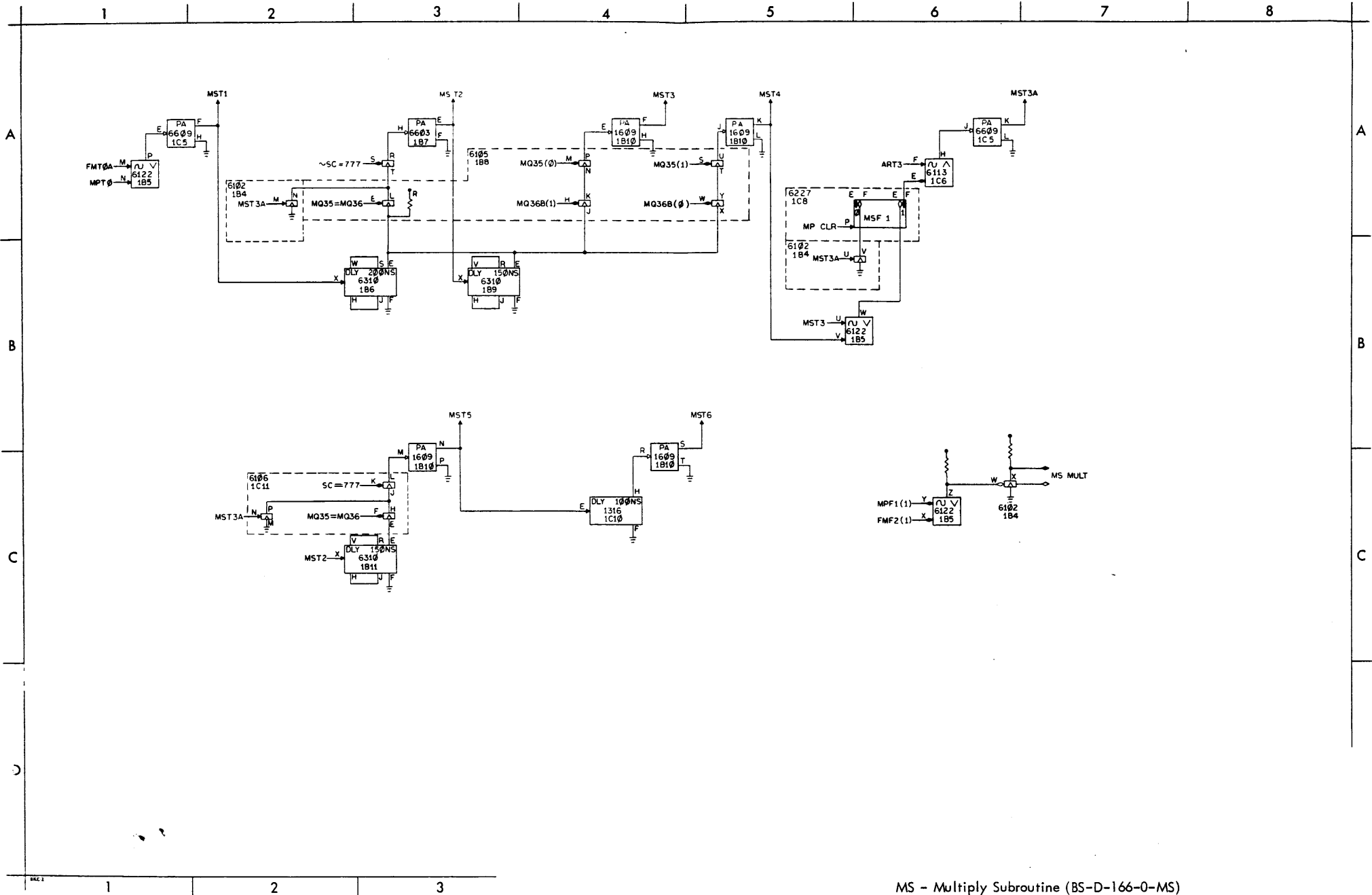
MP - Fixed Point Multiply
(BS-D-166-0-MP)



FA, FD, FM - Floating Point Instructions
(BS-D-166-0-FADM)



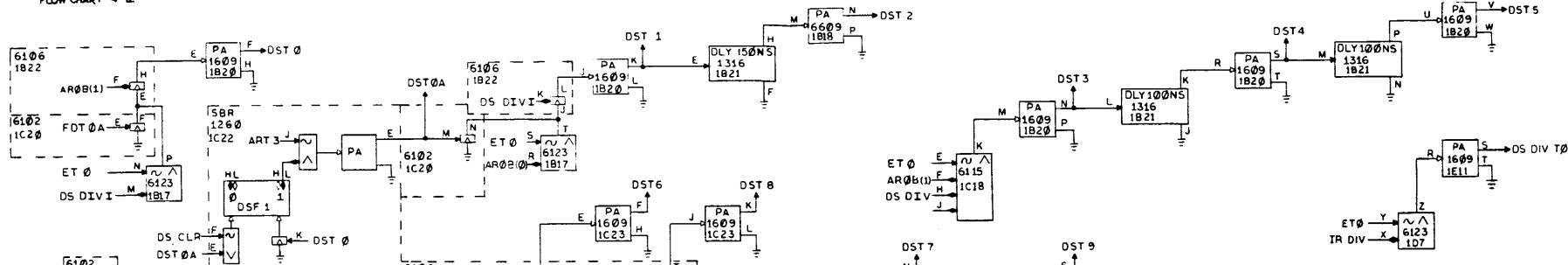
FP - Exponent Calculate Subroutine
(BS-D-166-0-FP)



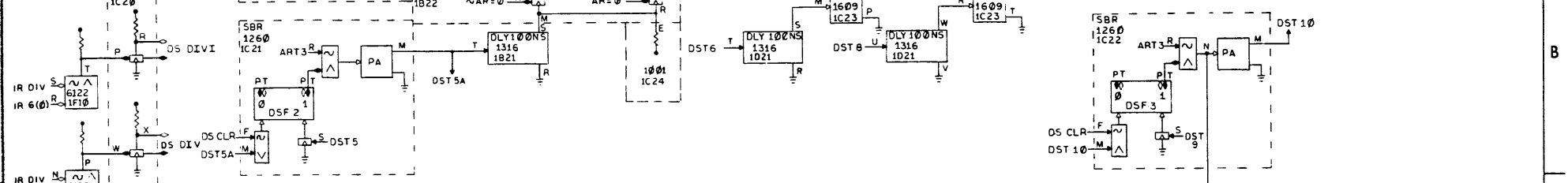
MS - Multiply Subroutine (BS-D-166-0-MS)

FLOW CHART 4-12

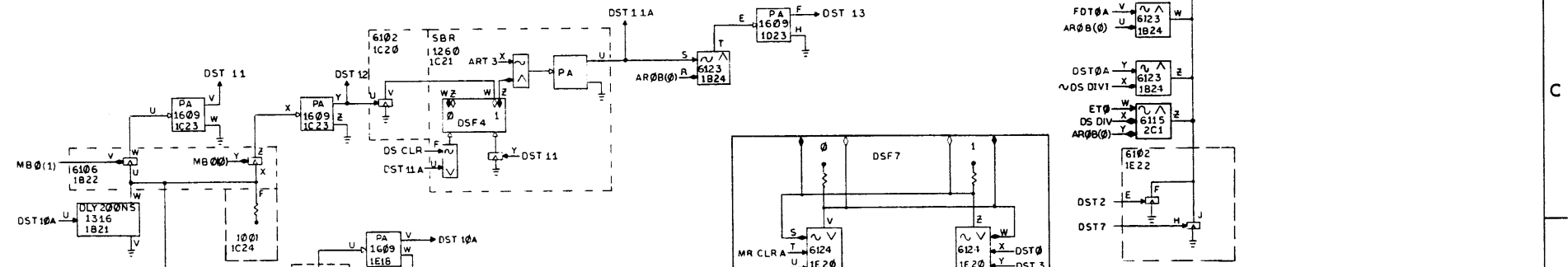
A



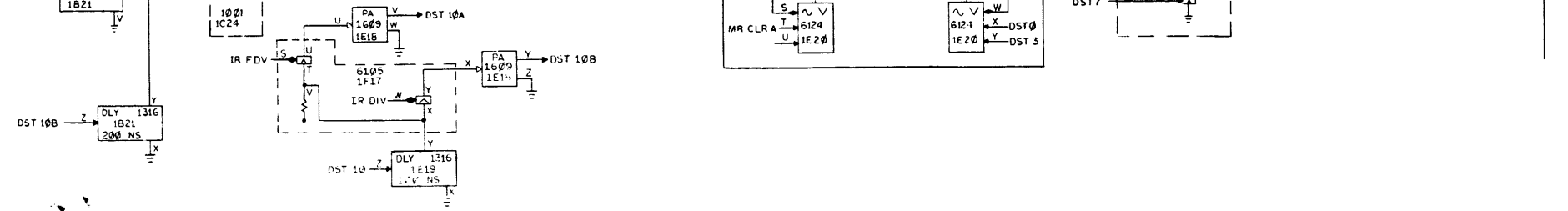
B



C



D



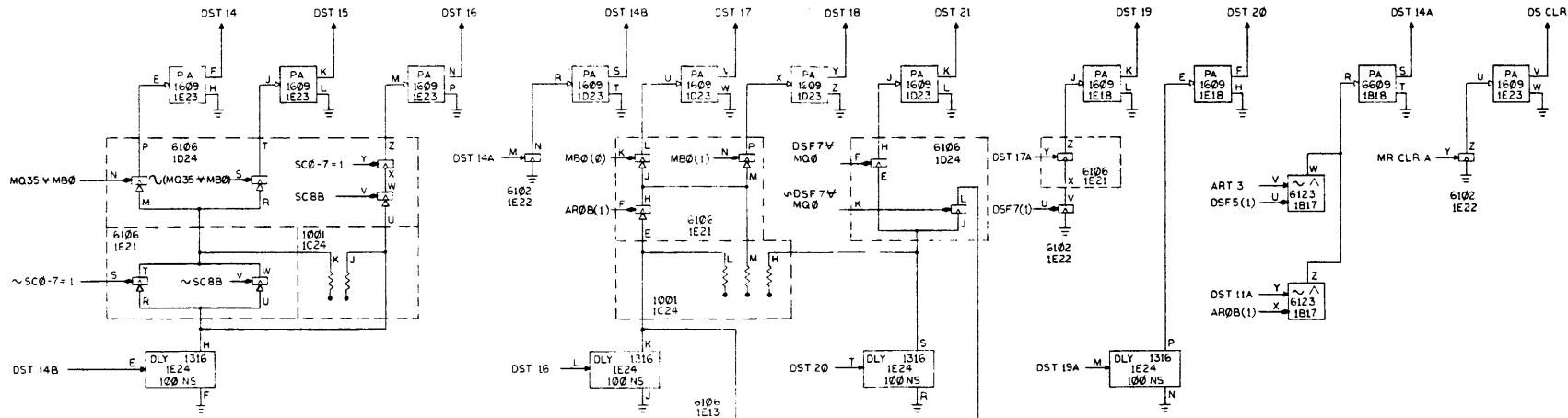
A

B

C

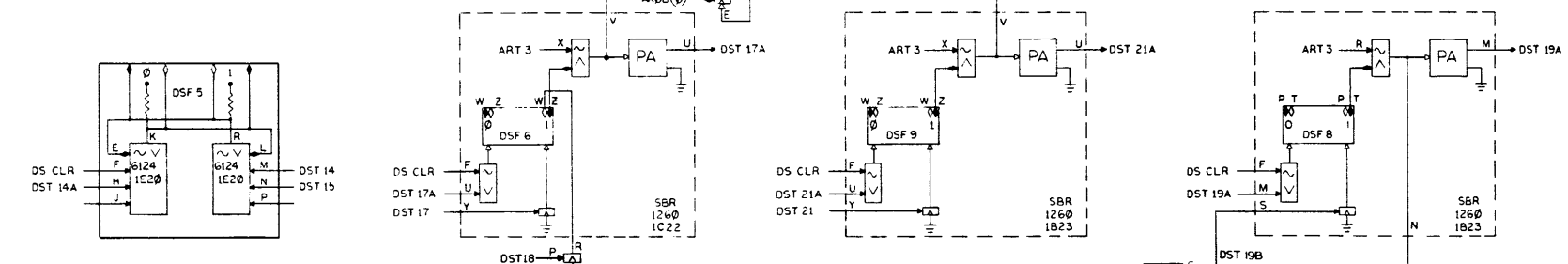
A

A



B

B

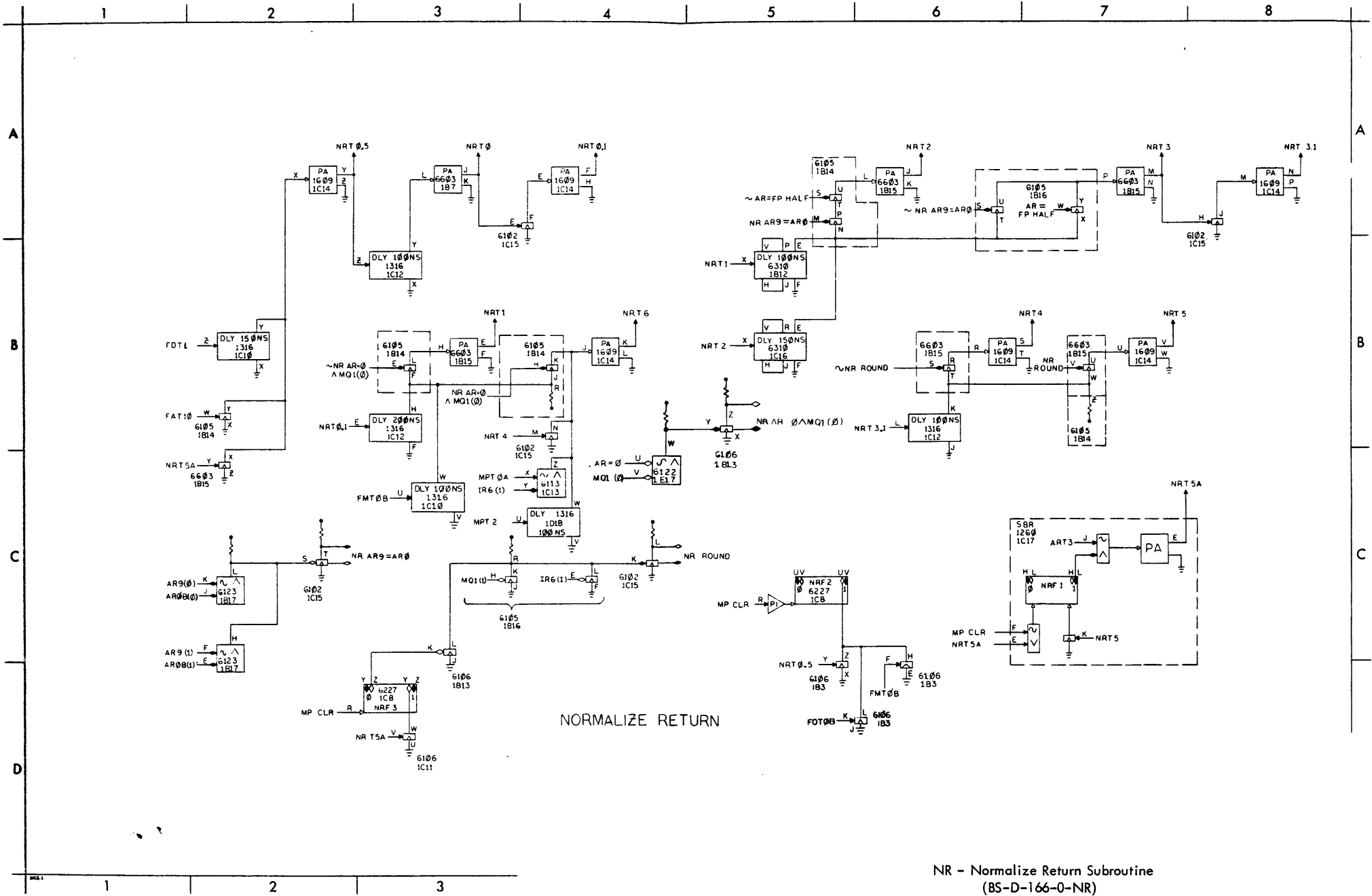


C

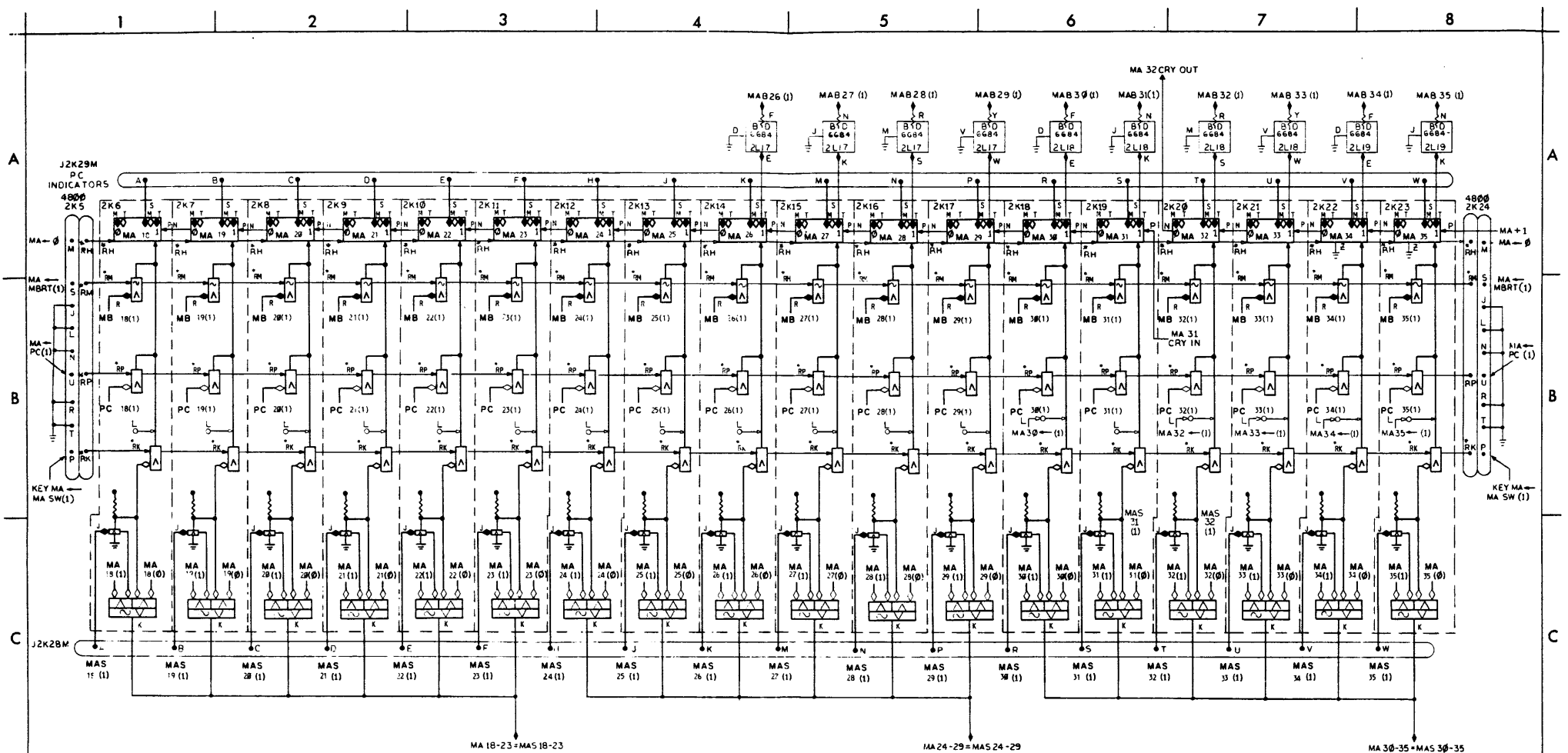
C



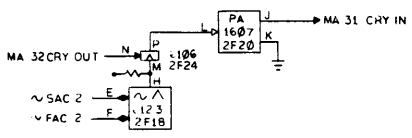
D



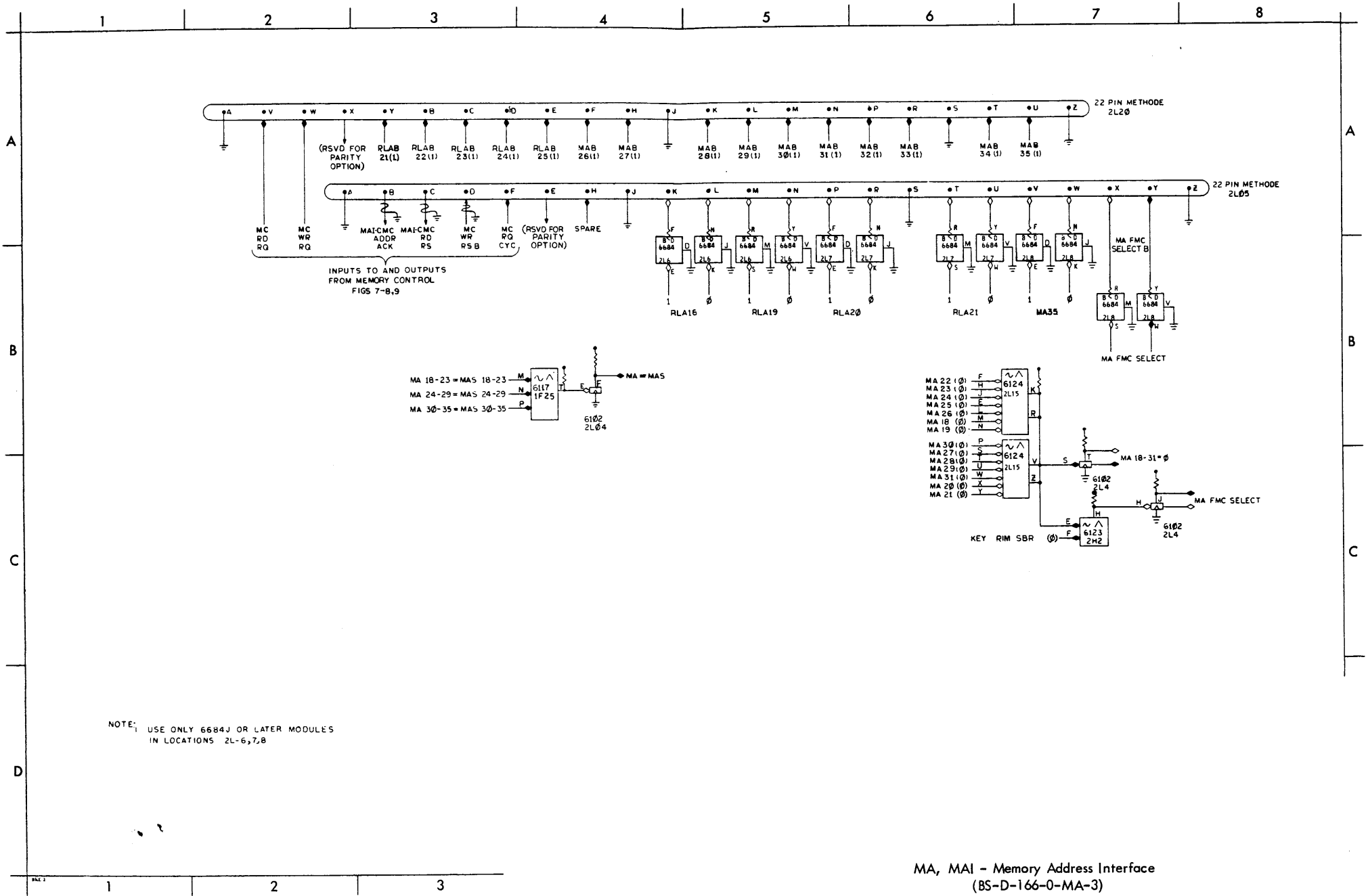
NR - Normalize Return Subroutine
(BS-D-166-0-NR)

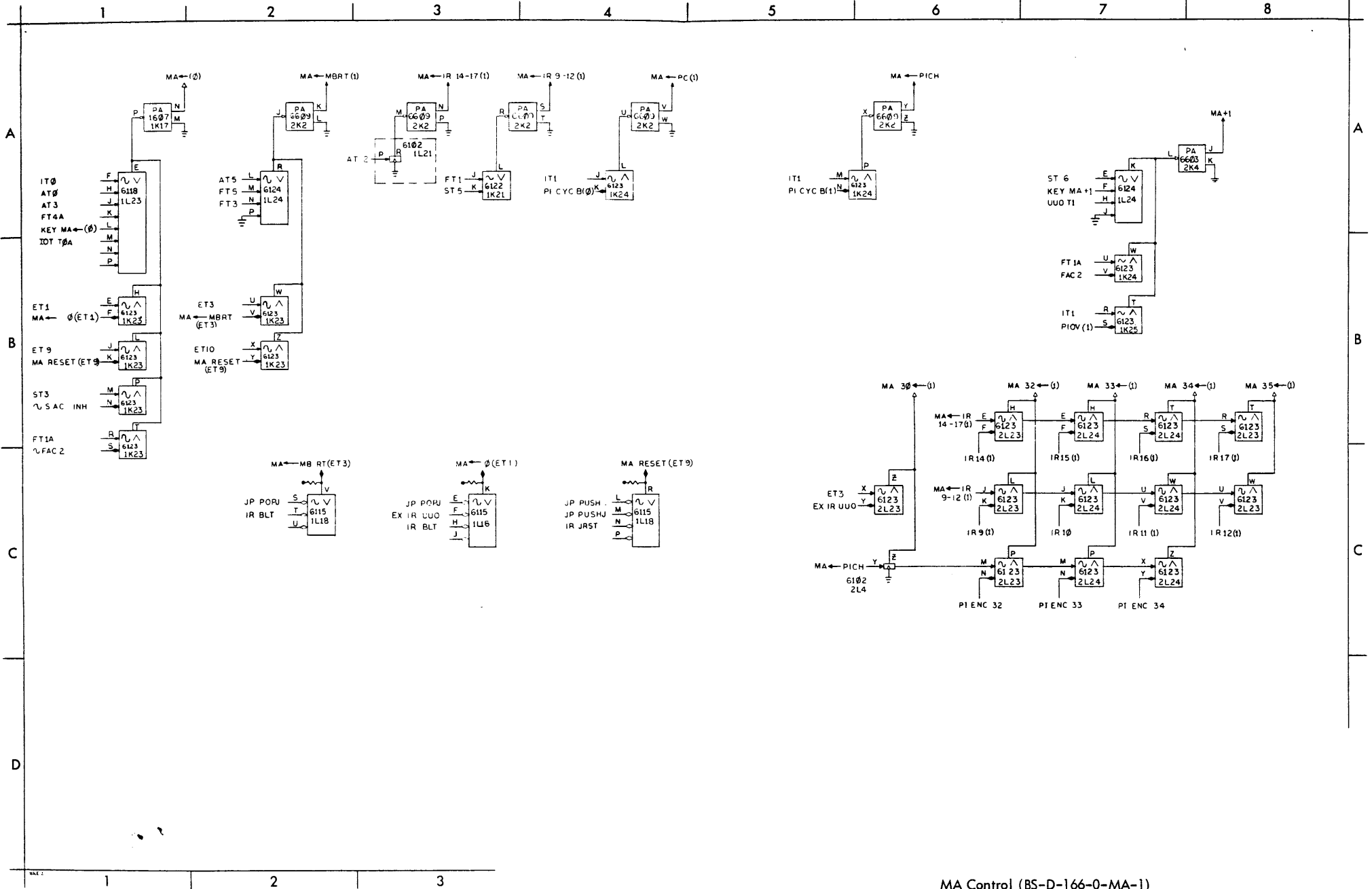


- NOTE
1. ALL FF PACKAGES ARE 6206
 2. - INDICATES REAR CONNECTOR PIN.
 3. GROUND PIN D AND Z IN ALL 6206'S.
 4. USE ONLY 6684 OR LATER MODULES IN LOCATIONS 2L-17, 18, 19



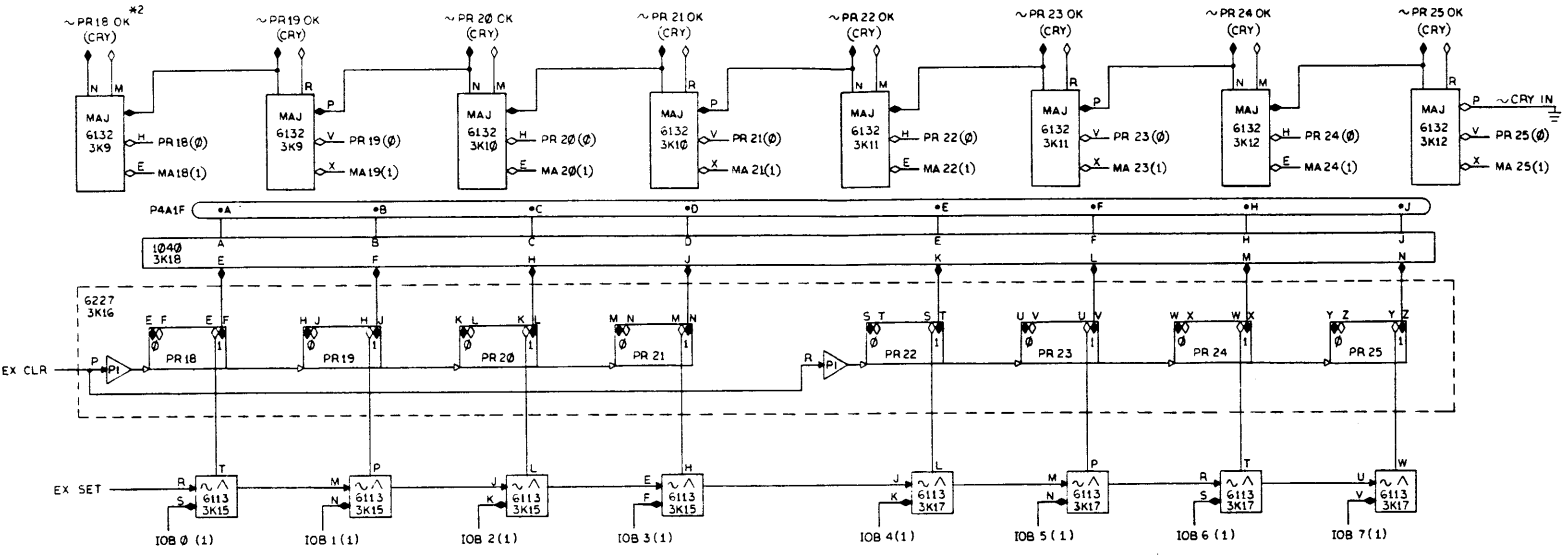
MA - Memory Address 18-35
(BS-D-166-0-MA-2)





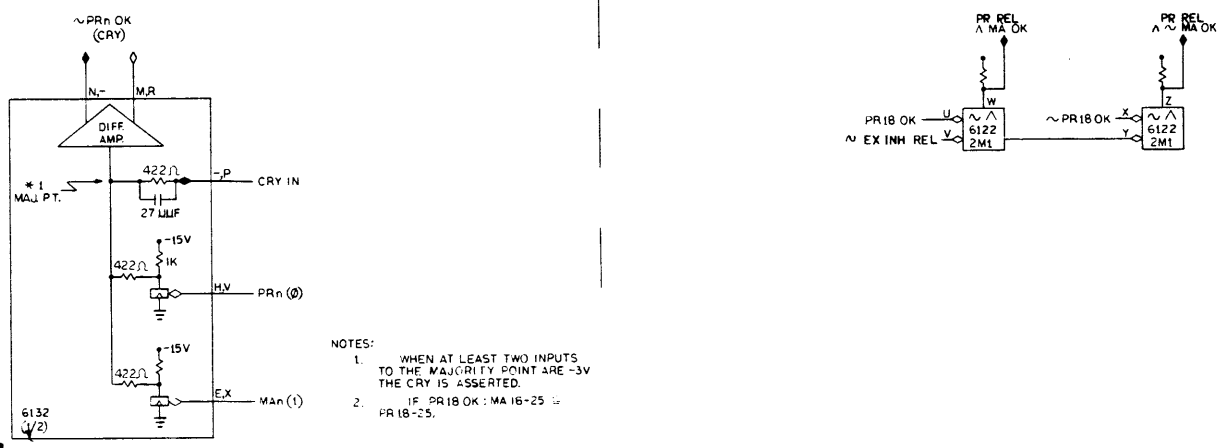
MA Control (BS-D-166-0-MA-1)

A A



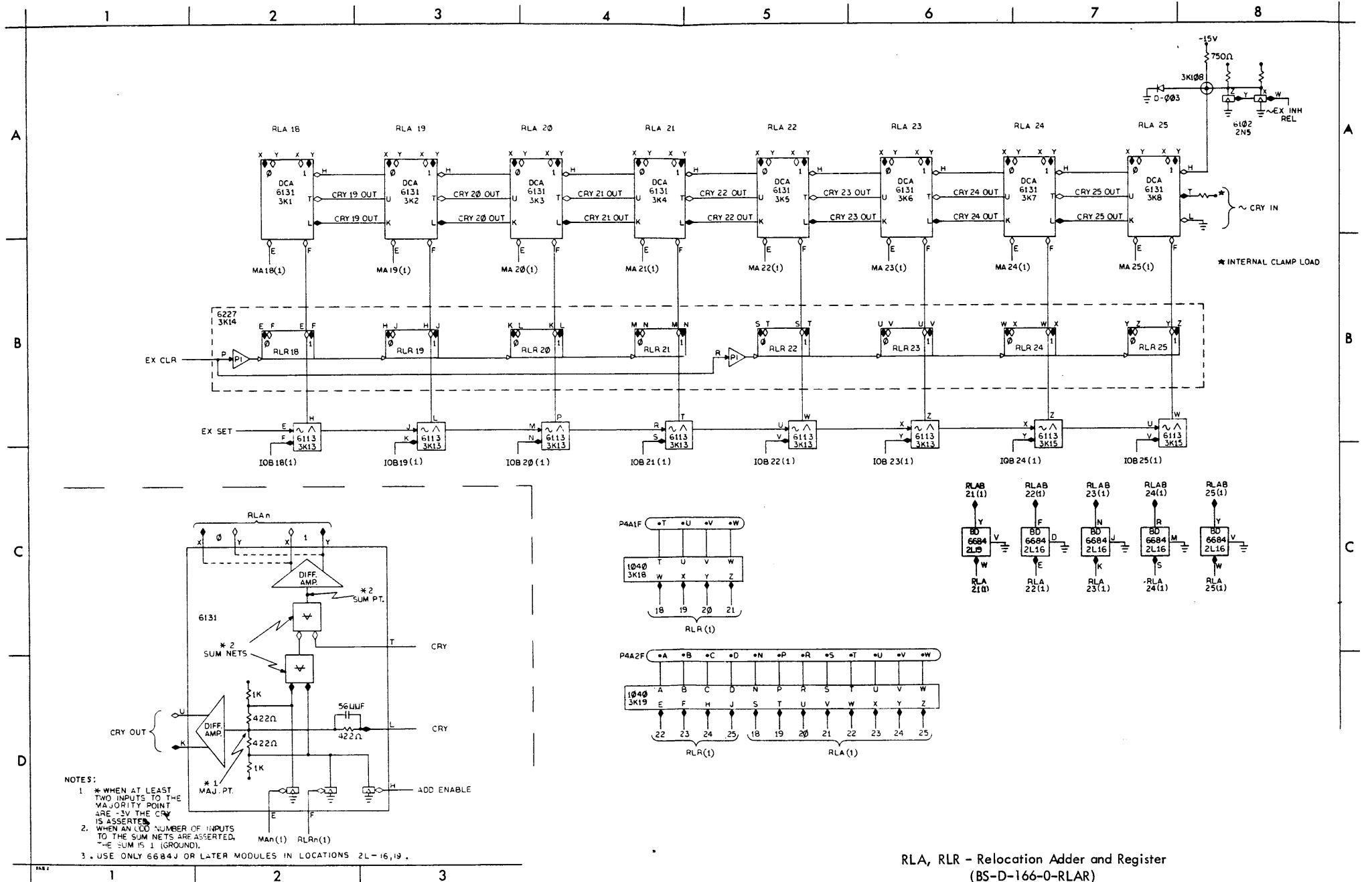
B B

C C



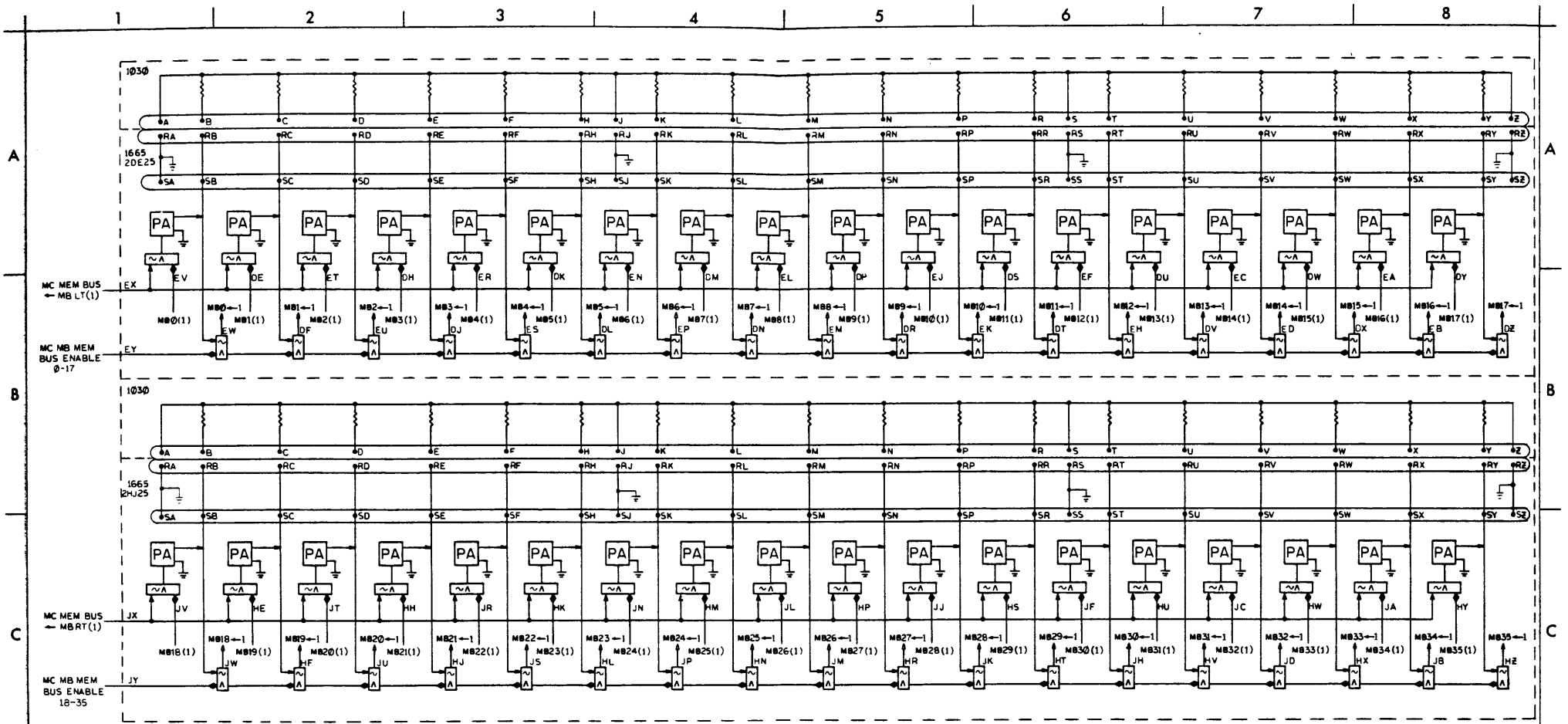
D

- NOTES:
1. WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
 2. IF PR18 OK : MA16-25 & PR18-25.



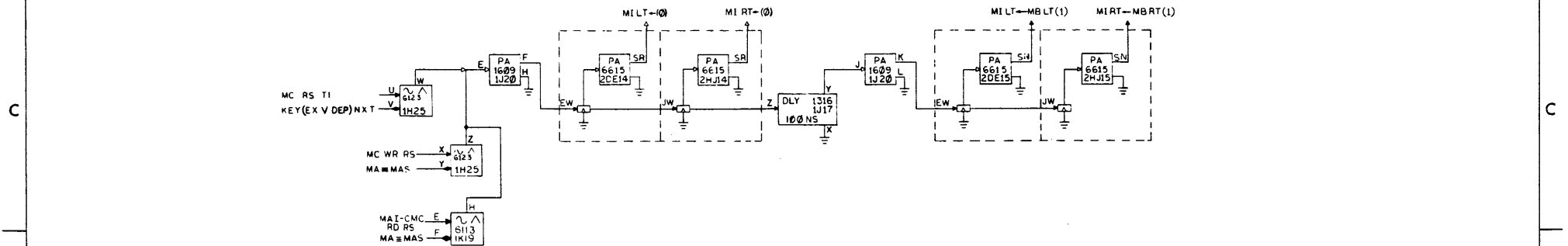
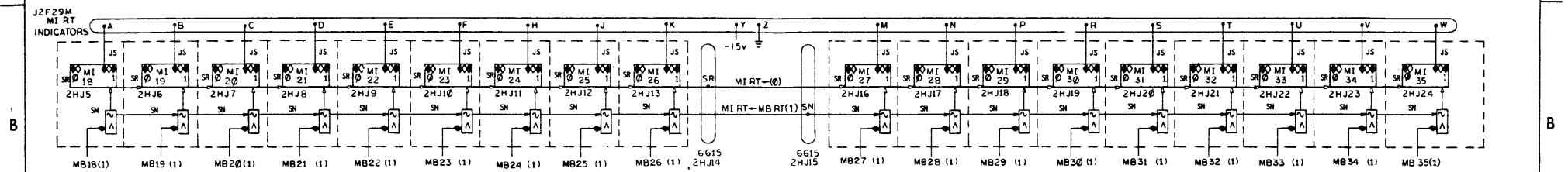
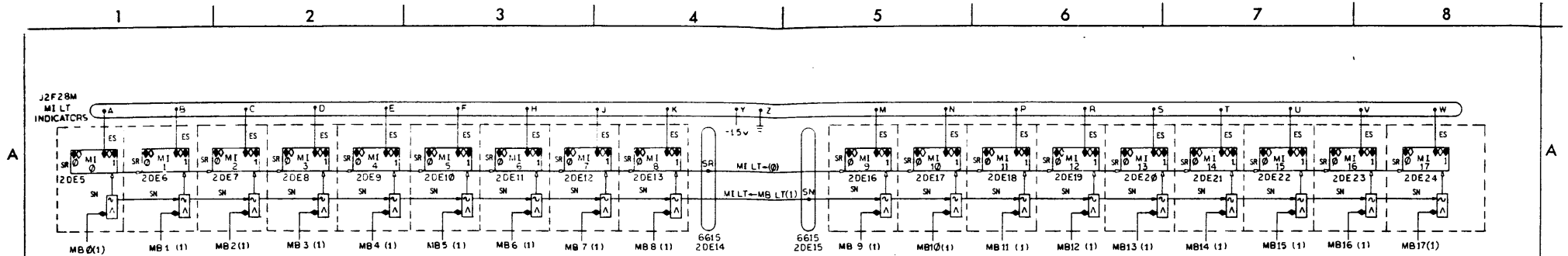
- NOTES:
- * WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
 - WHEN AN LED NUMBER OF INPUTS TO THE SUM NETS ARE ASSERTED, THE SUM IS 1 (GROUND).
 - USE ONLY 66B4J OR LATER MODULES IN LOCATIONS 2L-16,19.

RLA, RLR - Relocation Adder and Register
(BS-D-166-0-RLAR)



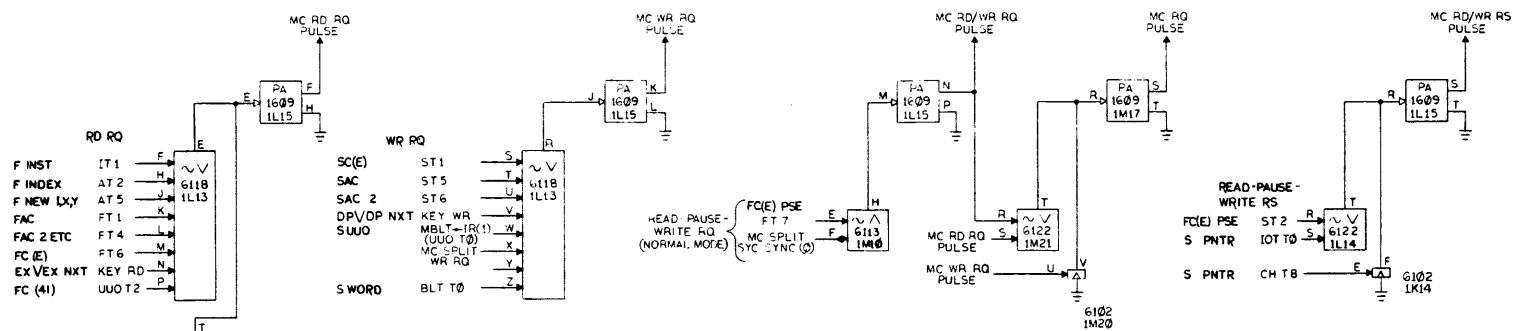
NOTE:
ALL TERMINATOR RES. IN THE
1030 MODULES ARE 100 OHMS.

MB - Memory Buffer Interface
(BS-D-166-0-MB-4)

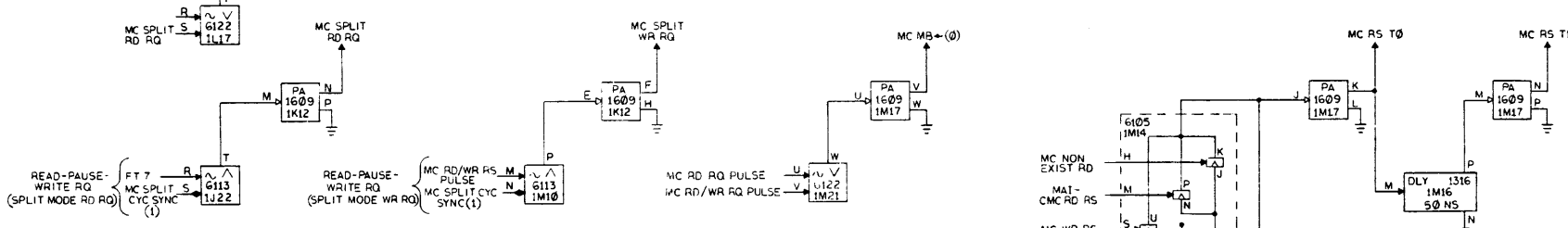


- NOTE:
1. ALL FF PACKAGES ARE 6205
 2. SN AND SR ARE PINS IN LOWER REAR CONNECTOR

A

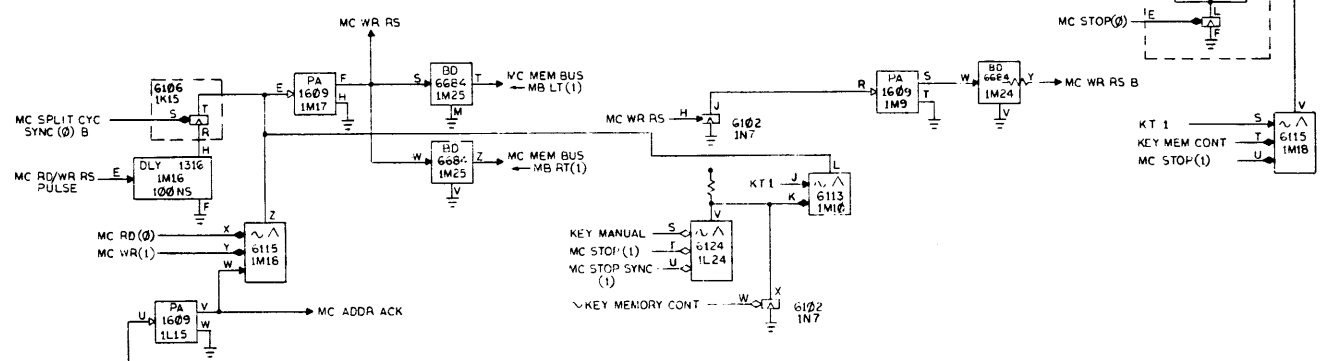


B



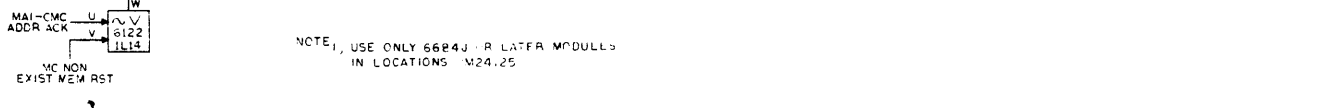
FLOW CHART 4-3

C



NOTE 1, USE ONLY 6684J OR LATER MODULES IN LOCATIONS M24,25

D



A

B

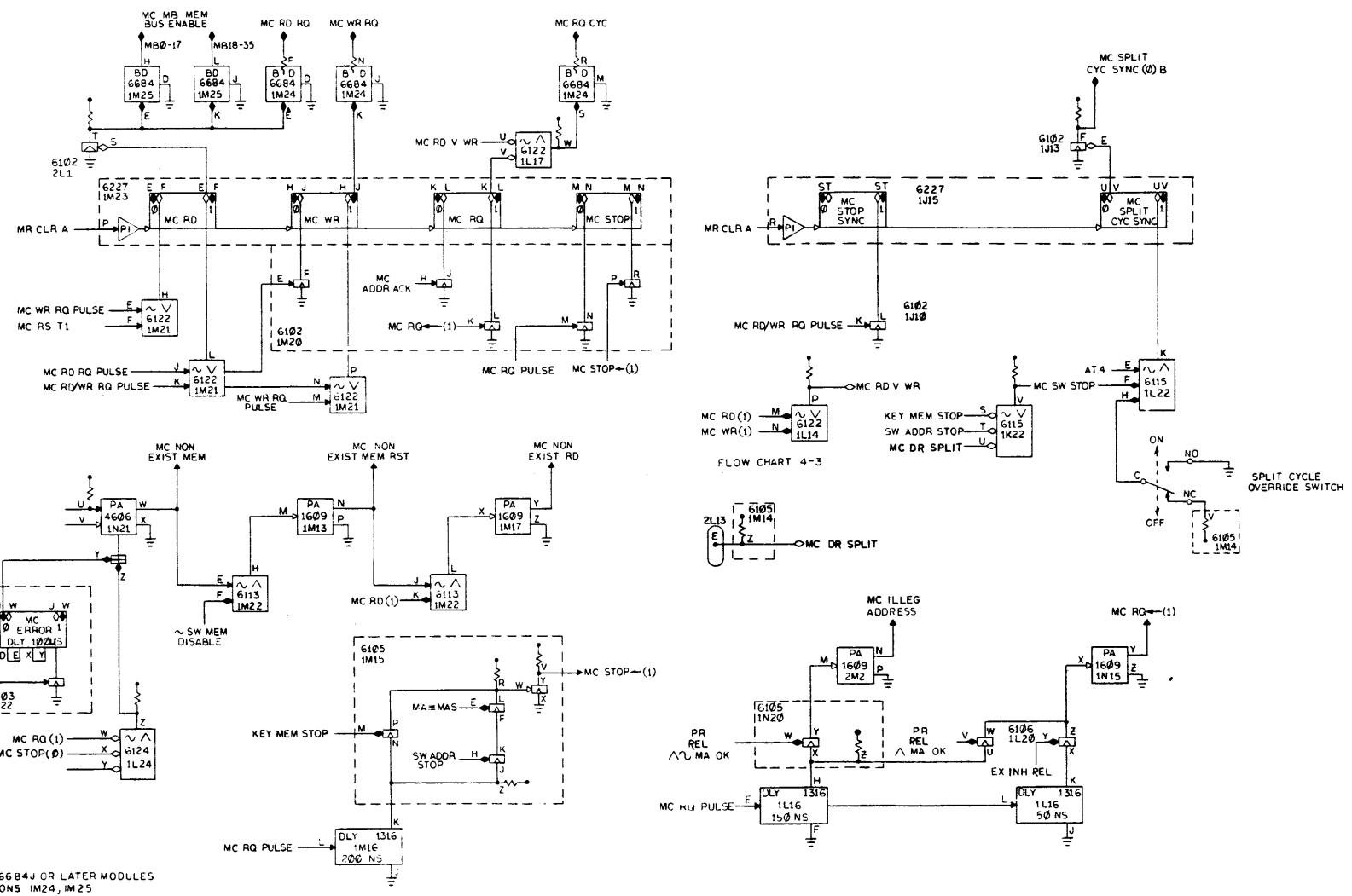
C

D

A

B

C



IO BUS(§ 8.4)

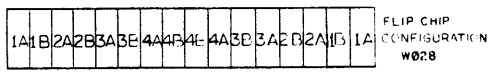
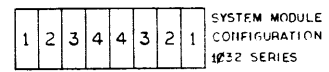
MEM BUS(§§ 7.1a, 7.2)

FLIP CHIP PIN	SYSTEM MODULE PIN	IO CABLE 1	IO CABLE 2	IO CABLE 3	IO CABLE 4
C	A	GND	GND	GND	GND
D	B	IOB 0 (1) →	IOB 18 (1) →	IOB RESET →	DATA0 CLEAR →
E	C	IOB 1 (1) →	IOB 19 (1) →	POWER ON -15V	DATA0 SET →
H	D	IOB 2 (1) →	IOB 20 (1) →		CON 0 CLEAR →
K	E	IOB 3 (1) →	IOB 21 (1) →	MC DR SPLIT →	CON 0 SET →
M	F	IOB 4 (1) →	IOB 22 (1) →	IOS 3 (0) →	IOB ← DATA1 →
P	H	IOB 5 (1) →	IOB 23 (1) →	IOS 3 (1) →	IOB ← STATUS →
N	J	GND	GND	GND	GND
S	K	IOB 6 (1) →	IOB 24 (1) →	IOS 4 (0) →	
T	L	IOB 7 (1) →	IOB 25 (1) →	IOS 4 (1) →	
V	M	IOB 8 (1) →	IOB 26 (1) →	IOS 5 (0) →	
D	N	IOB 9 (1) →	IOB 27 (1) →	IOS 5 (1) →	
E	P	IOB 10 (1) →	IOB 28 (1) →	IOS 6 (0) →	
H	R	IOB 11 (1) →	IOB 29 (1) →	IOS 6 (1) →	PI REQ 1 →
J	S	GND	GND	GND	GND
K	T	IOB 12 (1) →	IOB 30 (1) →	IOS 7 (0) →	PI REQ 2 →
M	U	IOB 13 (1) →	IOB 31 (1) →	IOS 7 (1) →	PI REQ 3 →
P	V	IOB 14 (1) →	IOB 32 (1) →	IOS 8 (0) →	PI REQ 4 →
S	W	IOB 15 (1) →	IOB 33 (1) →	IOS 8 (1) →	PI REQ 5 →
T	X	IOB 16 (1) →	IOB 34 (1) →	IOS 9 (0) →	PI REQ 6 →
V	Y	IOB 17 (1) →	IOB 35 (1) →	IOS 9 (1) →	PI REQ 7 →
U	Z	GND	GND	GND	GND

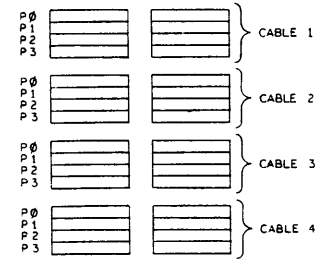
FLIP CHIP PIN	SYSTEM MODULE PIN	MEM CABLE 1	MEM CABLE 2	MEM CABLE 3	MEM CABLE 4
C	A	GND	GND	GND	GND
D	B	ADDR ACK →	MA 22 (1) →	MB 0 (1) →	MB 18 (1) →
E	C	RD RS →	MA 23 (1) →	MB 1 (1) →	MB 19 (1) →
H	D	WR RS →	MA 24 (1) →	MB 2 (1) →	MB 20 (1) →
K	E	PAR (1) →	MA 25 (1) →	MB 3 (1) →	MB 21 (1) →
M	F	RQ CYCLE →	MA 26 (1) →	MB 4 (1) →	MB 22 (1) →
P	H	SPARE	MA 27 (1) →	MB 5 (1) →	MB 23 (1) →
N	J	GND	GND	GND	GND
S	K	MA 18 (1) →	MA 28 (1) →	MB 6 (1) →	MB 24 (1) →
T	L	MA 18 (0) →	MA 29 (1) →	MB 7 (1) →	MB 25 (1) →
V	M	MA 19 (1) →	MA 30 (1) →	MB 8 (1) →	MB 26 (1) →
D	N	MA 19 (0) →	MA 31 (1) →	MB 9 (1) →	MB 27 (1) →
E	P	MA 20 (1) →	MA 32 (1) →	MB 10 (1) →	MB 28 (1) →
H	R	MA 20 (0) →	MA 33 (1) →	MB 11 (1) →	MB 29 (1) →
J	S	GND	GND	GND	GND
K	T	MA 21 (1) →	MA 34 (1) →	MB 12 (1) →	MB 30 (1) →
M	U	MA 21 (0) →	MA 35 (1) →	MB 13 (1) →	MB 31 (1) →
P	V	MA 35 (1) →	MC RD RQ →	MB 14 (1) →	MB 32 (1) →
S	W	MA 35 (0) →	MC WR RQ →	MB 15 (1) →	MB 33 (1) →
T	X	MA FNC } →	PAR OPTION →	MB 16 (1) →	MB 34 (1) →
V	Y	SELECT B } →	MA 21 (1) →	MB 17 (1) →	MB 35 (1) →
U	Z	GND	GND	GND	GND

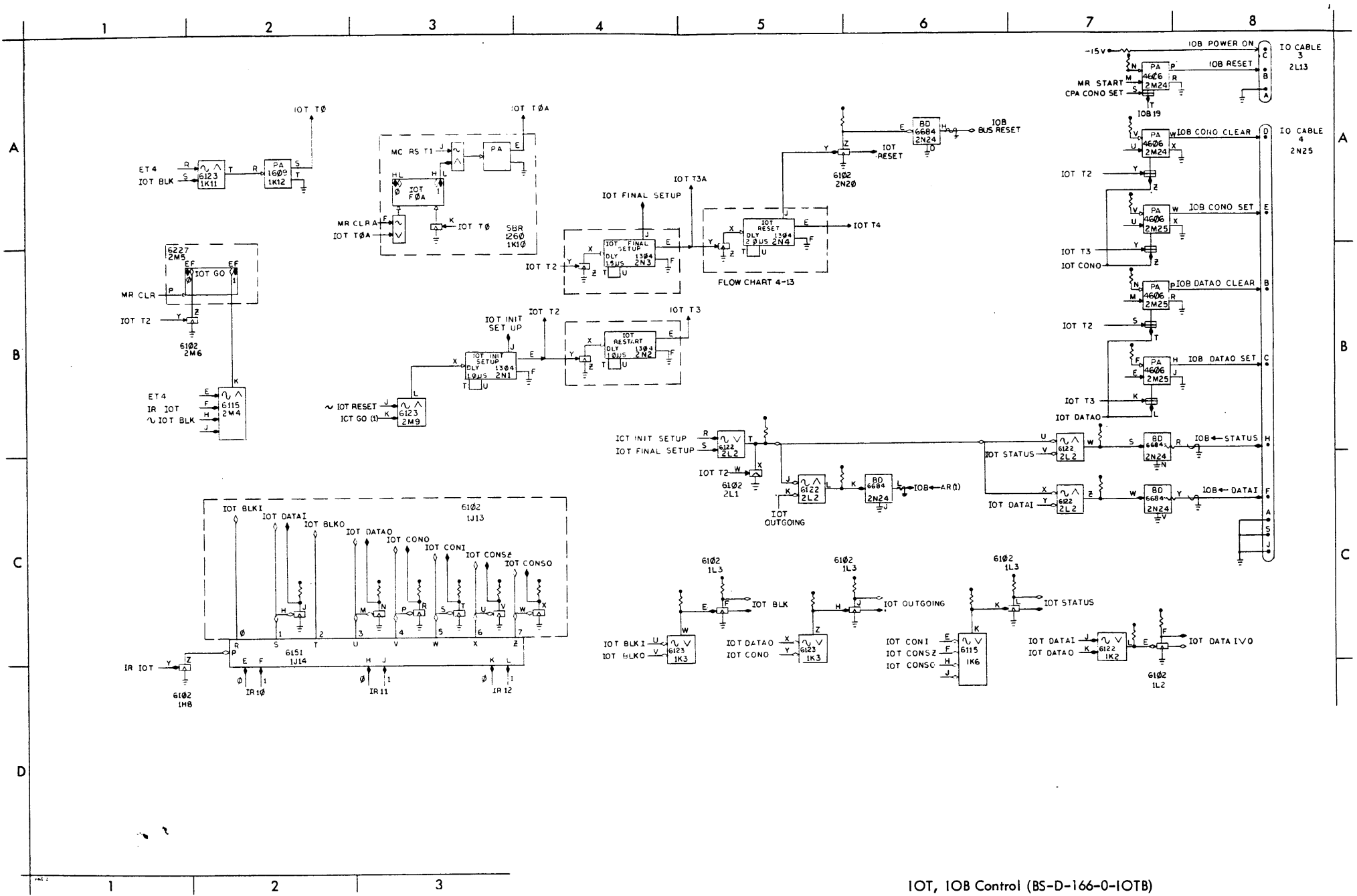
(SOURCE 3L6) (SOURCE 3L7) (SOURCE 2L13) (SOURCE 2N25)

(SOURCE 2L5) (SOURCE 2L20) (SOURCE 2E25) (SOURCE 2J25)

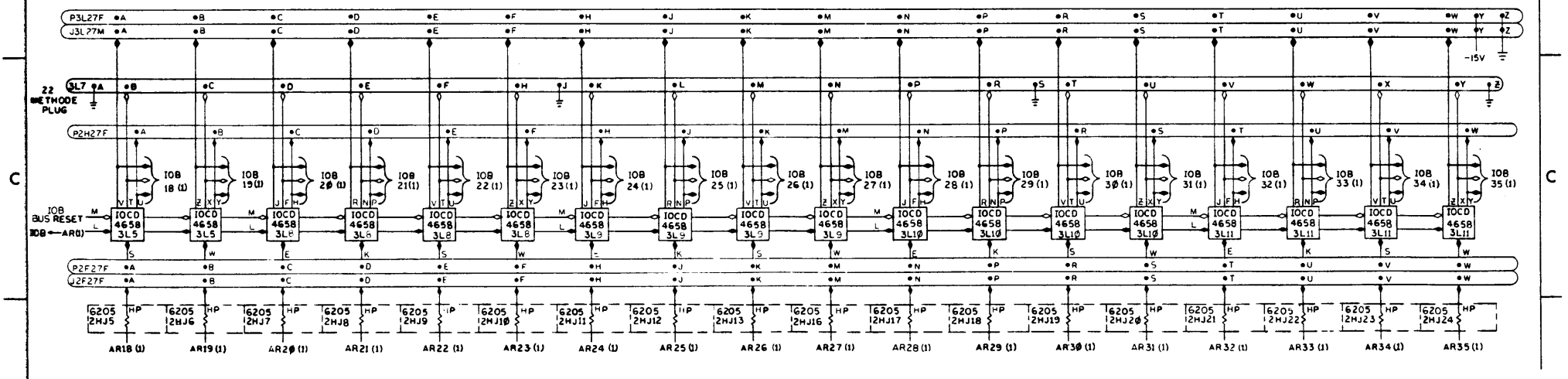
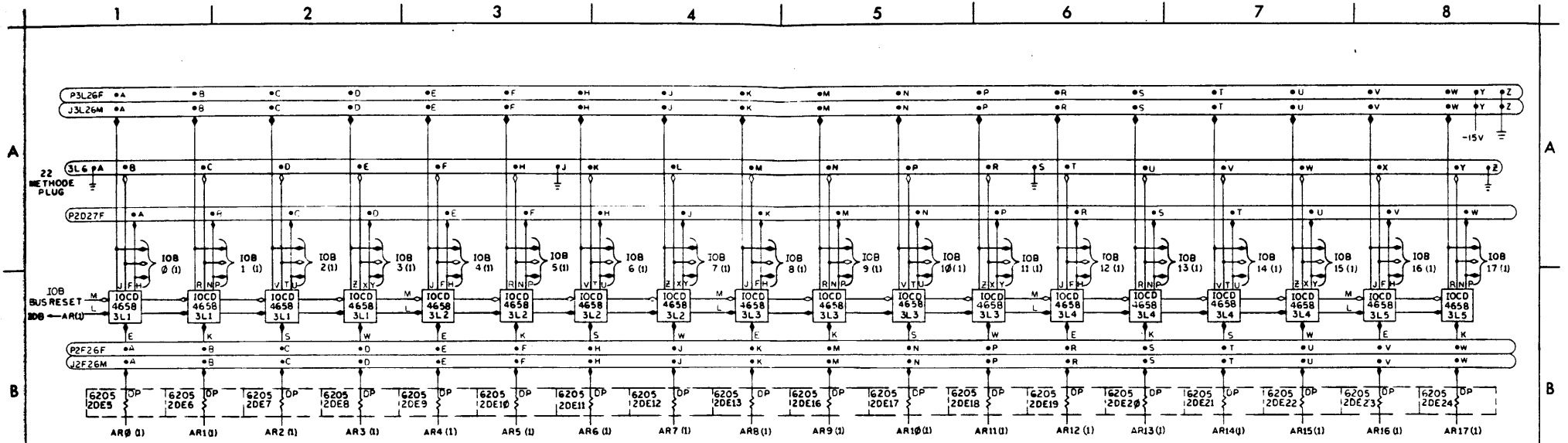


NOTE: 1 SYSTEM MODULE CONNECTOR (1032) = 2 FLIP CHIP CONNECTORS (W02B)

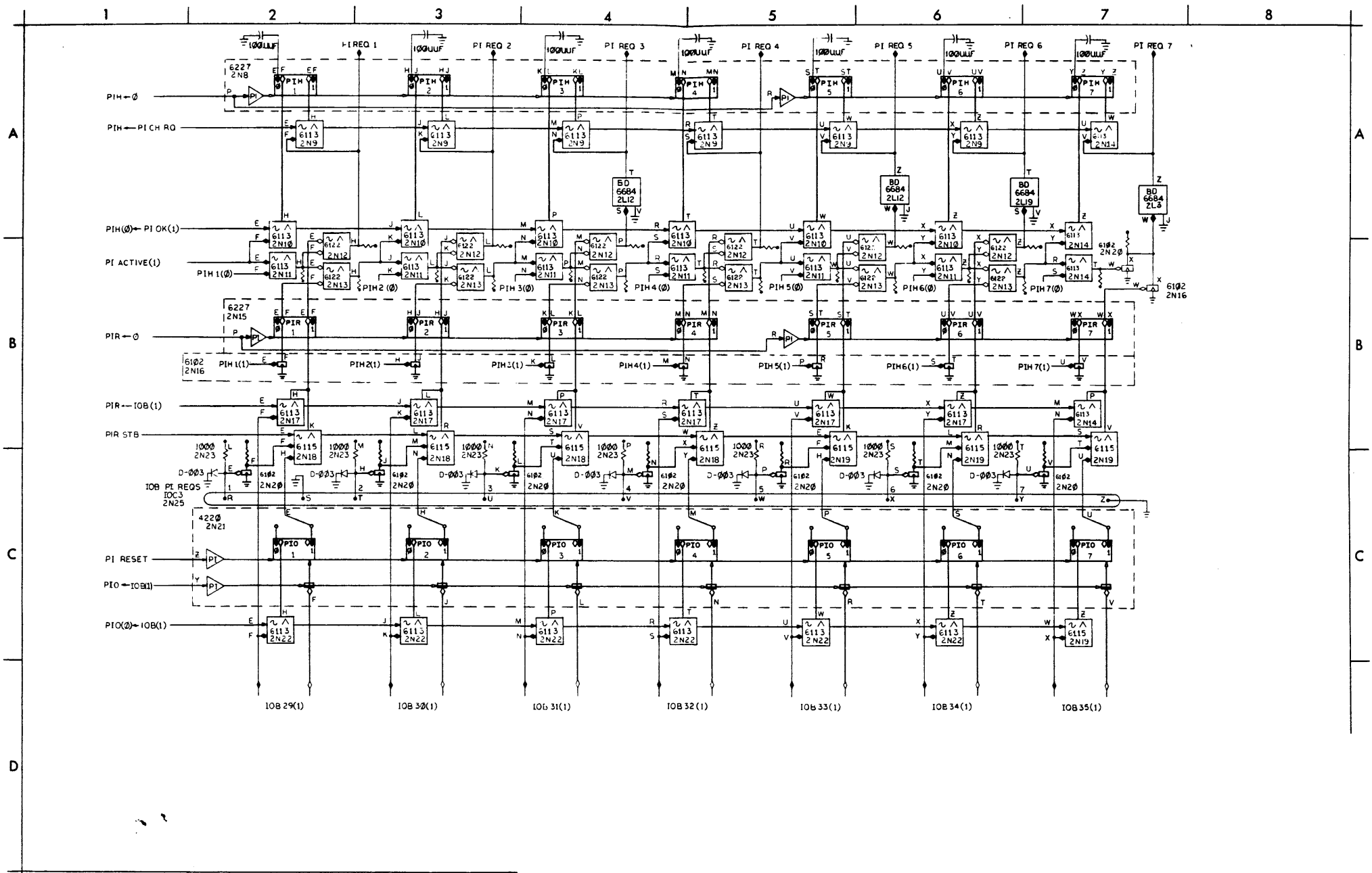




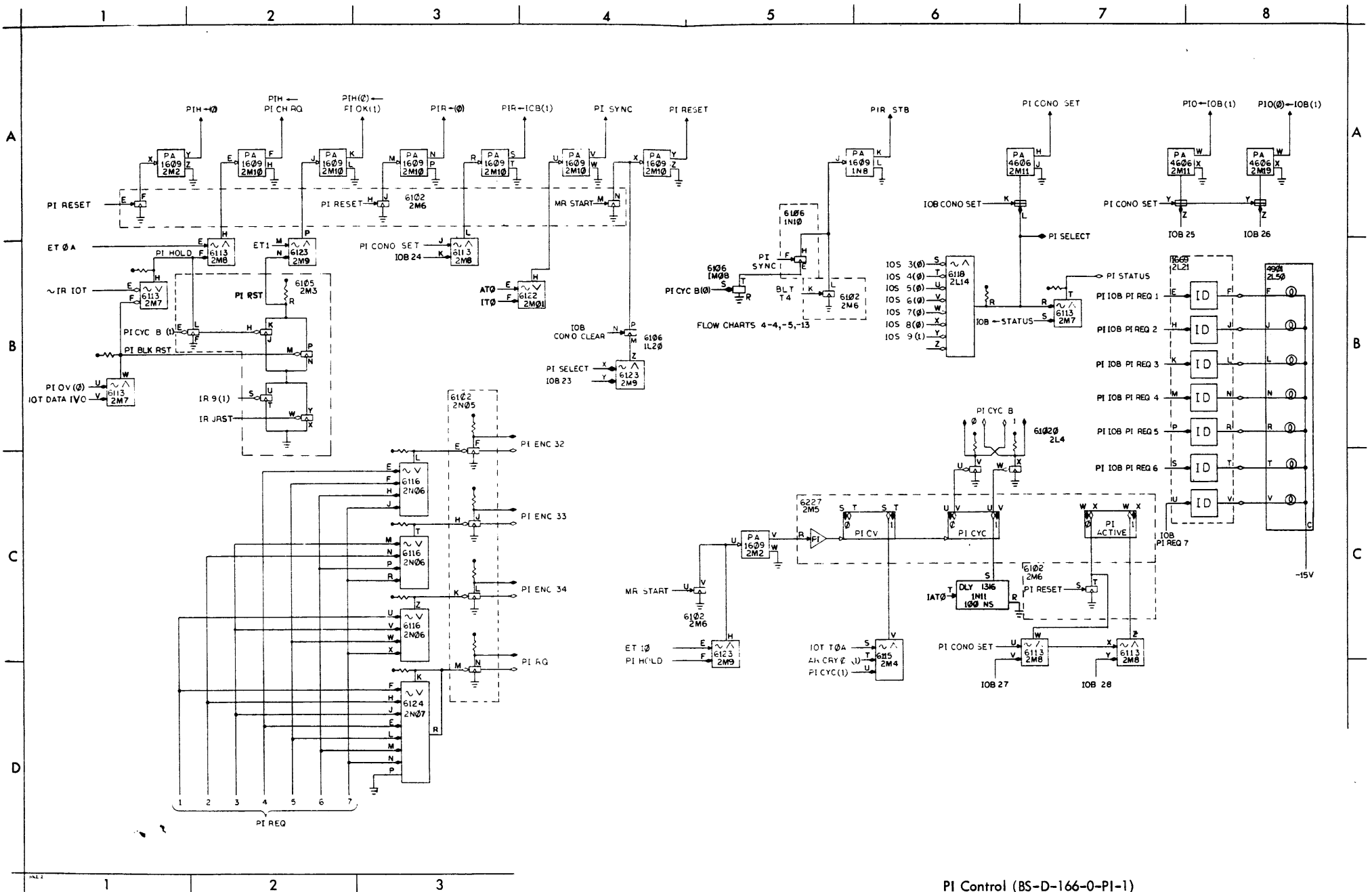
IOT, IOB Control (BS-D-166-0-IOTB)



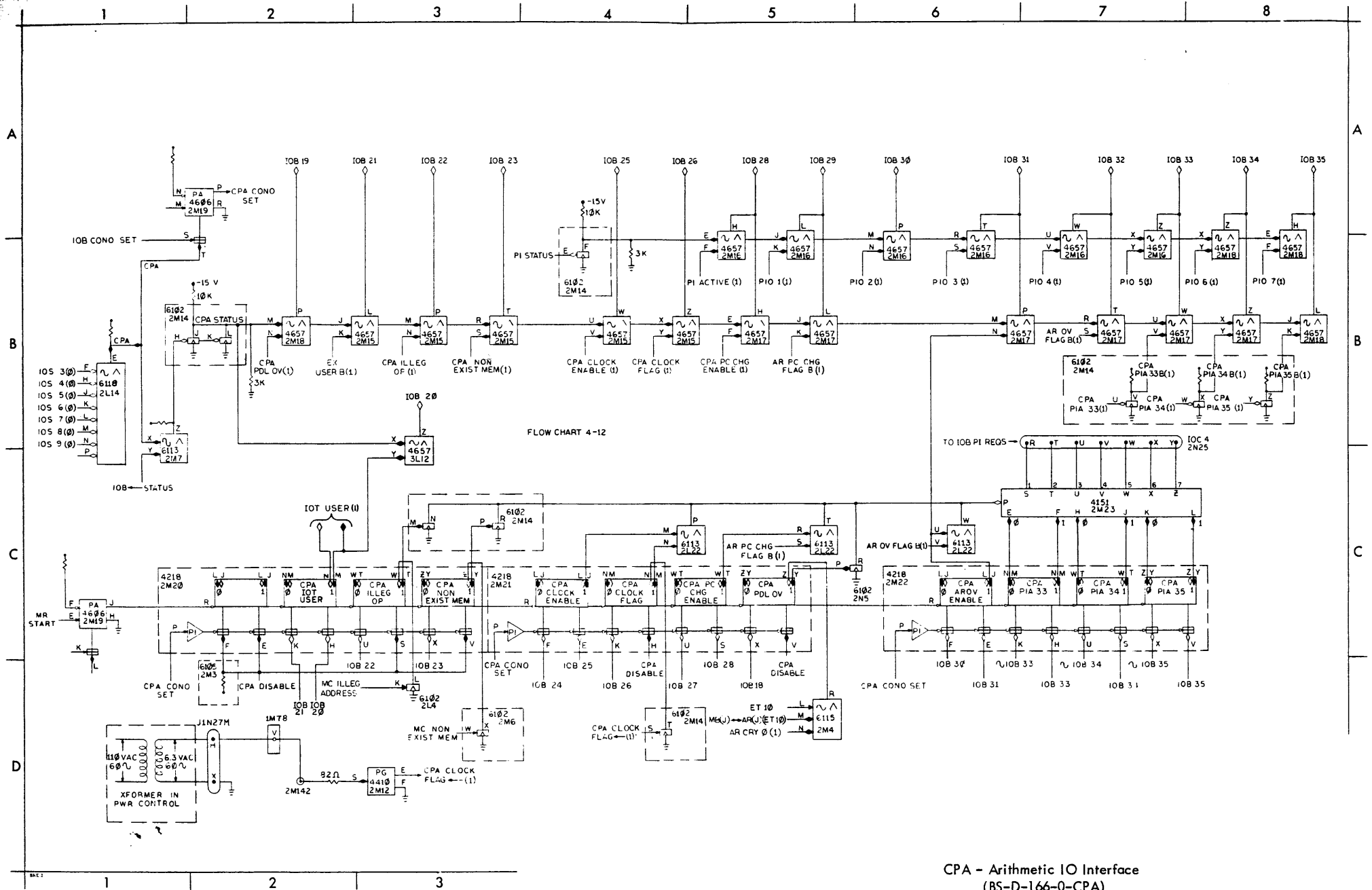
IOB - IO Bus (BS-D-166-0-IOB)



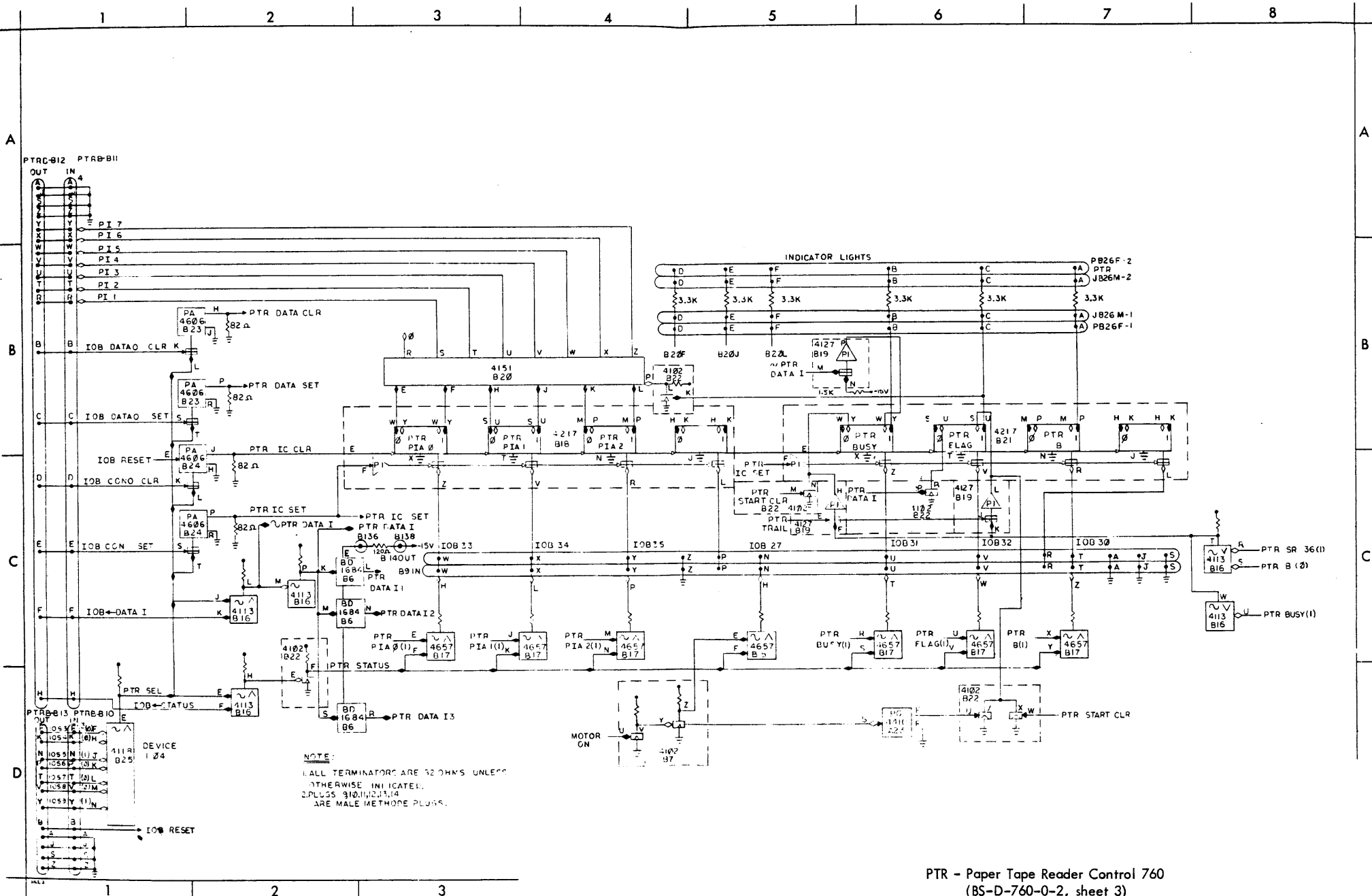
PI - PIH, R, O, 1-7 (BS-D-166-0-PI-2)



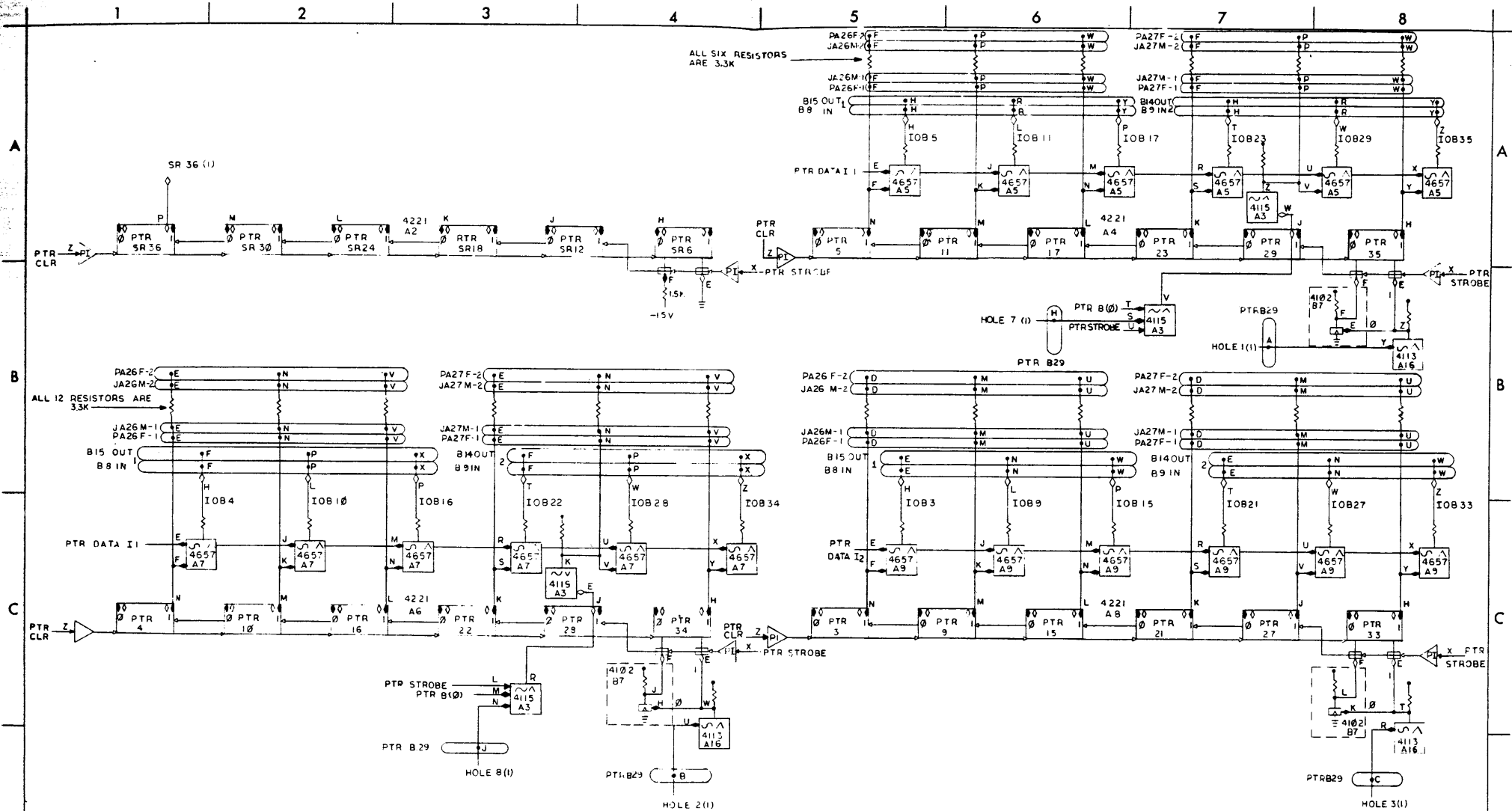
PI Control (BS-D-166-0-PI-1)



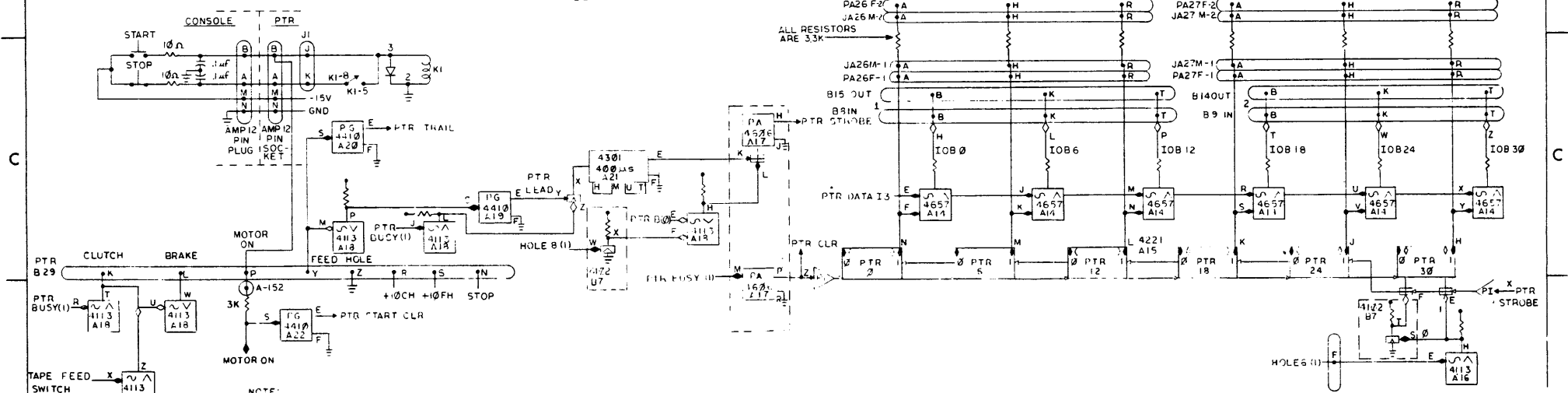
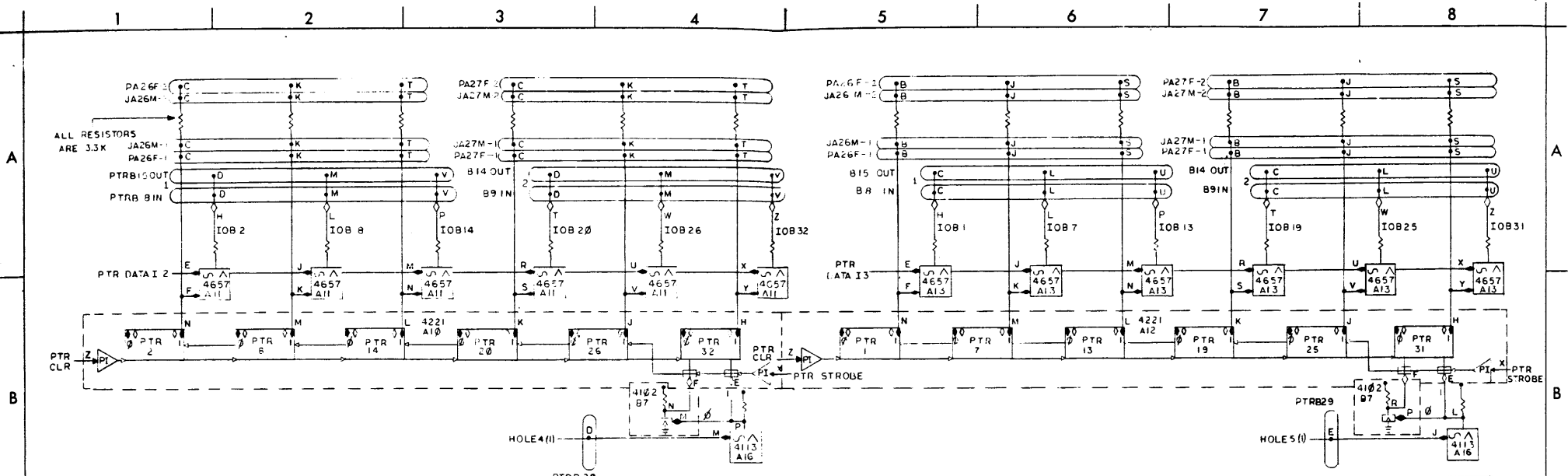
CPA - Arithmetic IO Interface
(BS-D-166-0-CPA)



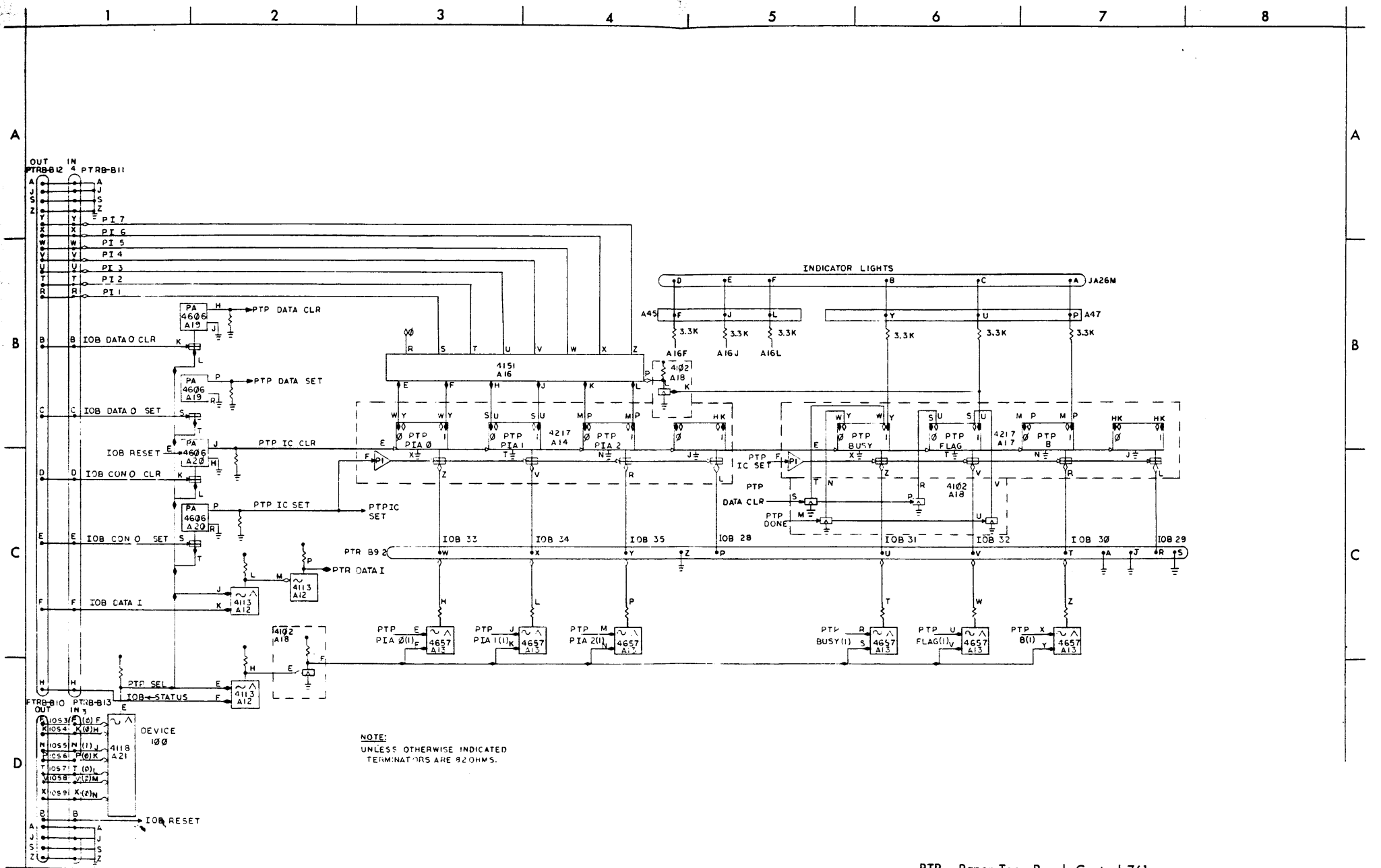
PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 3)



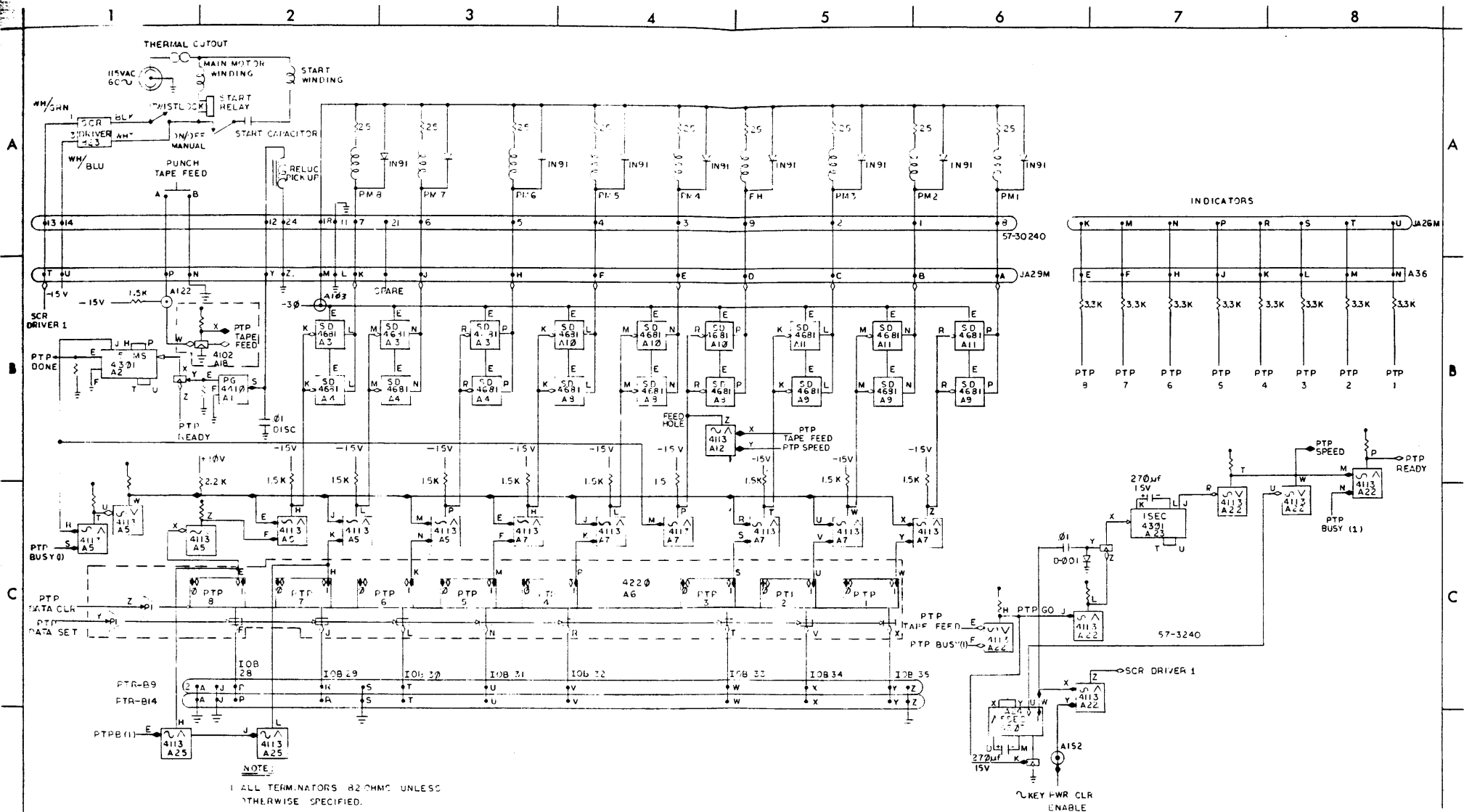
PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 1)



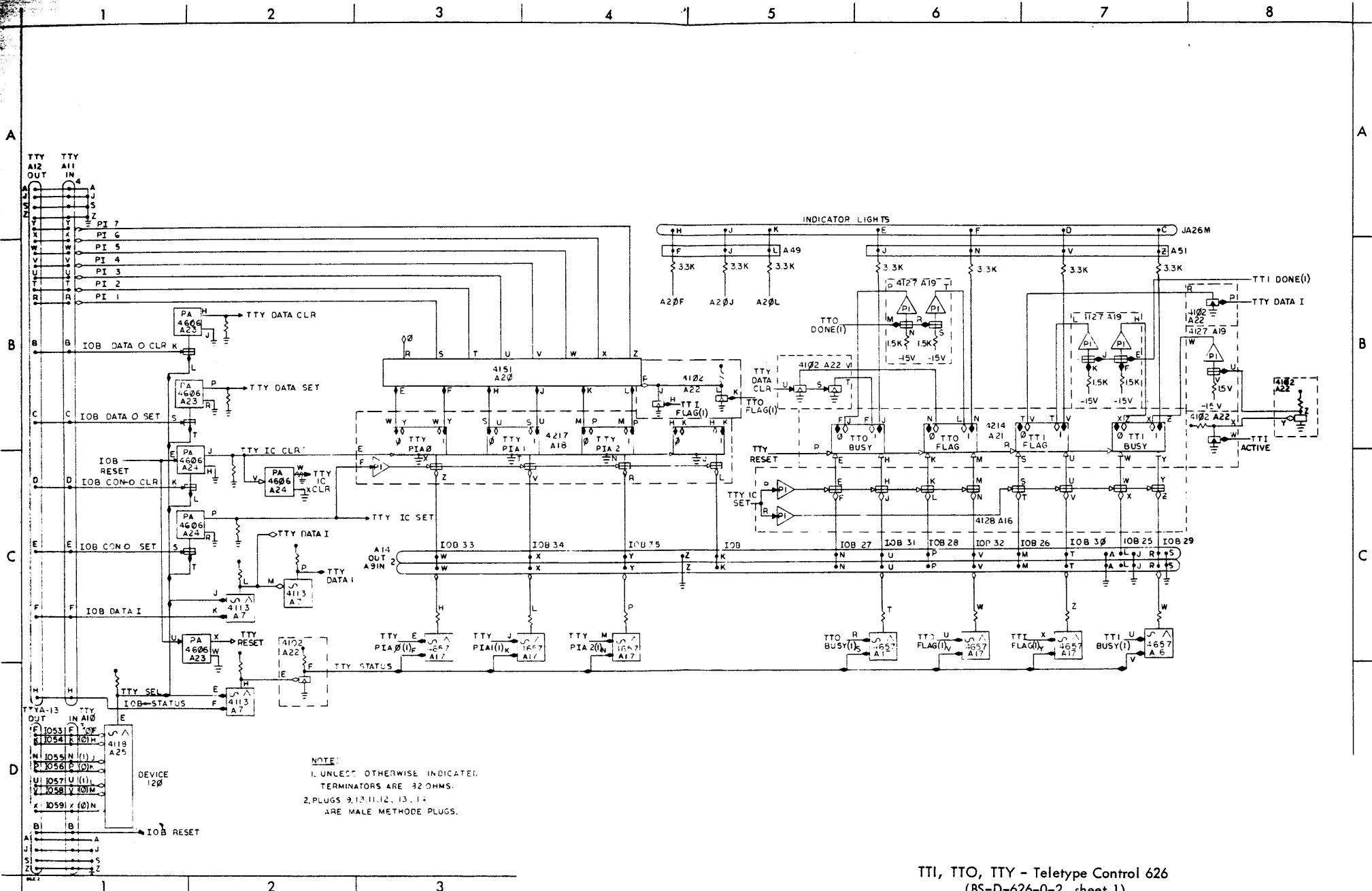
- NOTE:**
1. ALL TERMINATORS 20 OHMS UNLESS OTHERWISE SPECIFIED.
 2. SIZE SHOWN ON D-760-0-2 ARE FOR MC BUDF AND FEED HOLES.
 +10FXD = NORMAL +10V
 +10VAR = MARGINAL CHECK ONLY



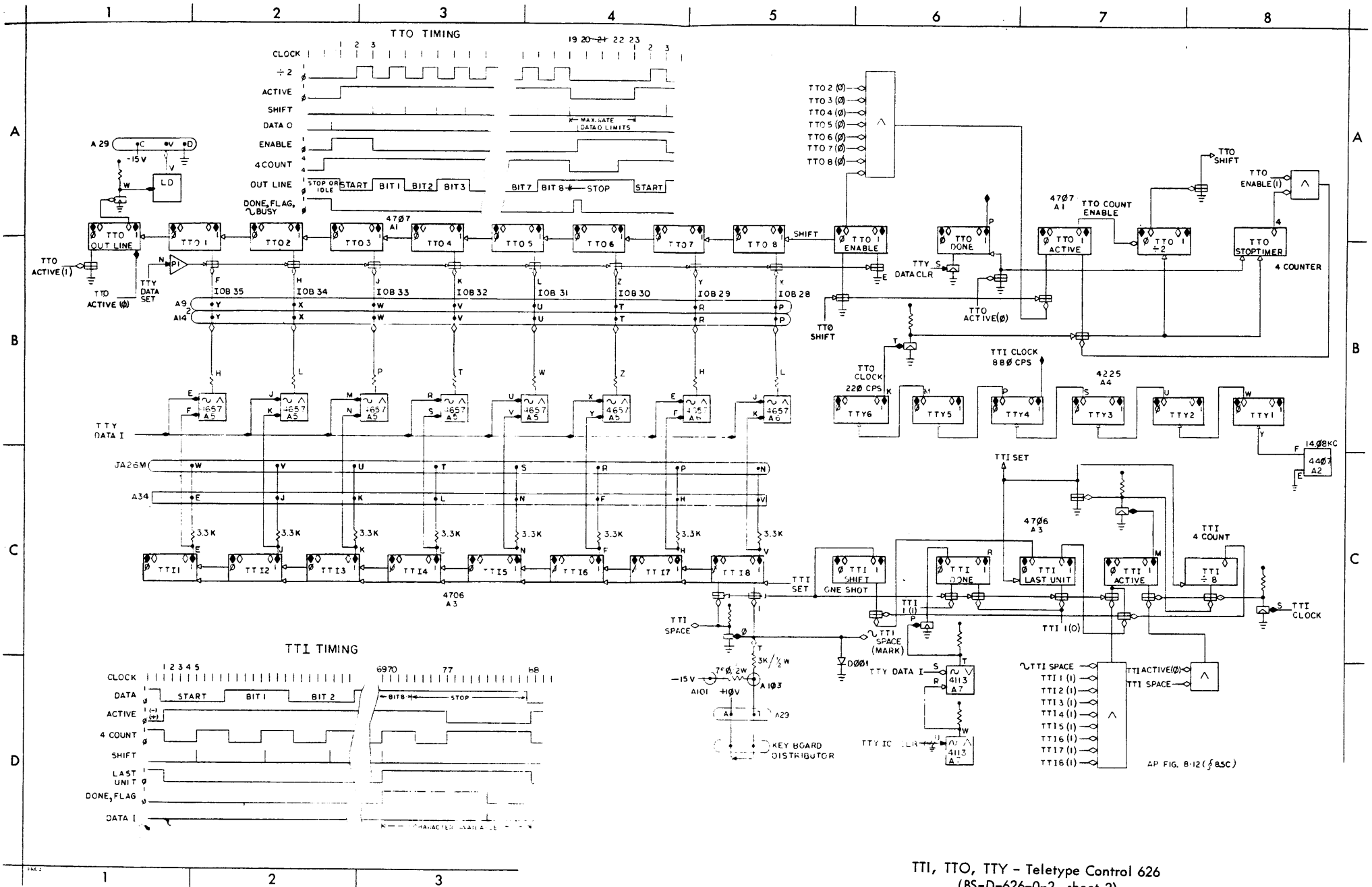
PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 2)



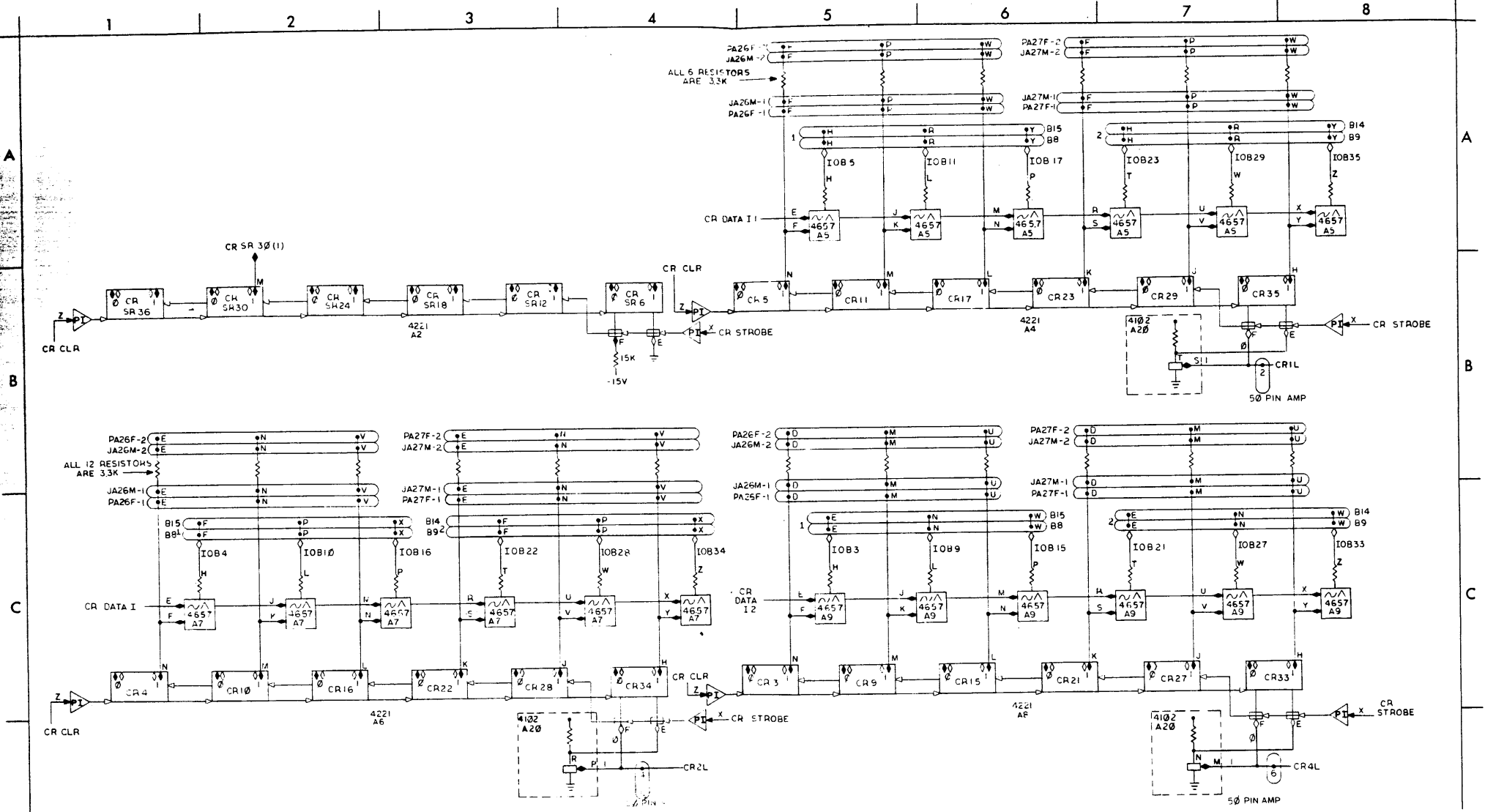
PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 1)



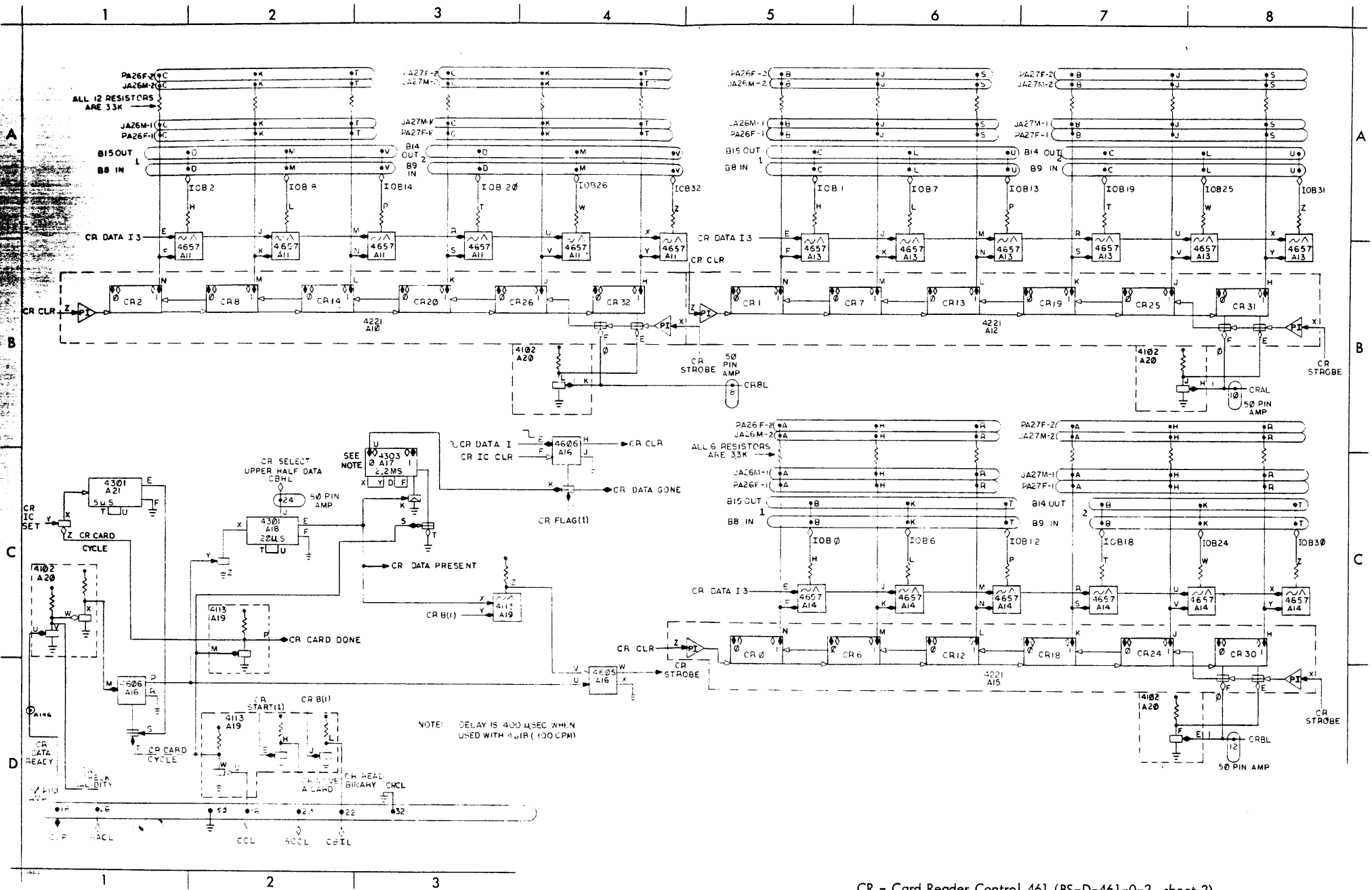
NOTE:
 1. UNLESS OTHERWISE INDICATED,
 TERMINATORS ARE 32 OHMS.
 2. PLUGS 9, 12, 11, 12, 13, 14
 ARE MALE METHODE PLUGS.



TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 2)



CR - Card Reader Control 461 (BS-D-461-0-2, sheet 1)



CR - Card Reader Control 461 (BS-D-461-0-2, sheet 2)