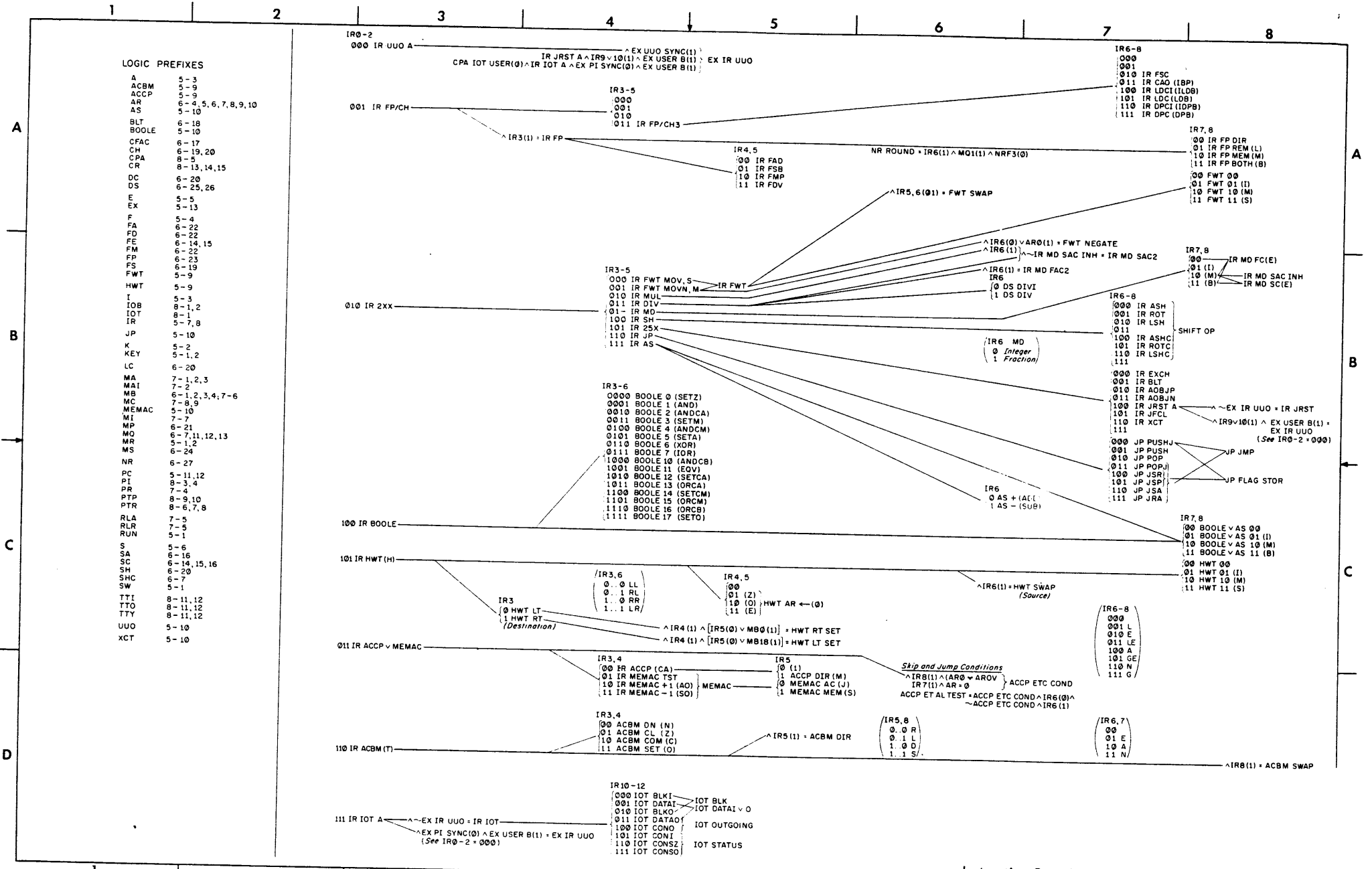
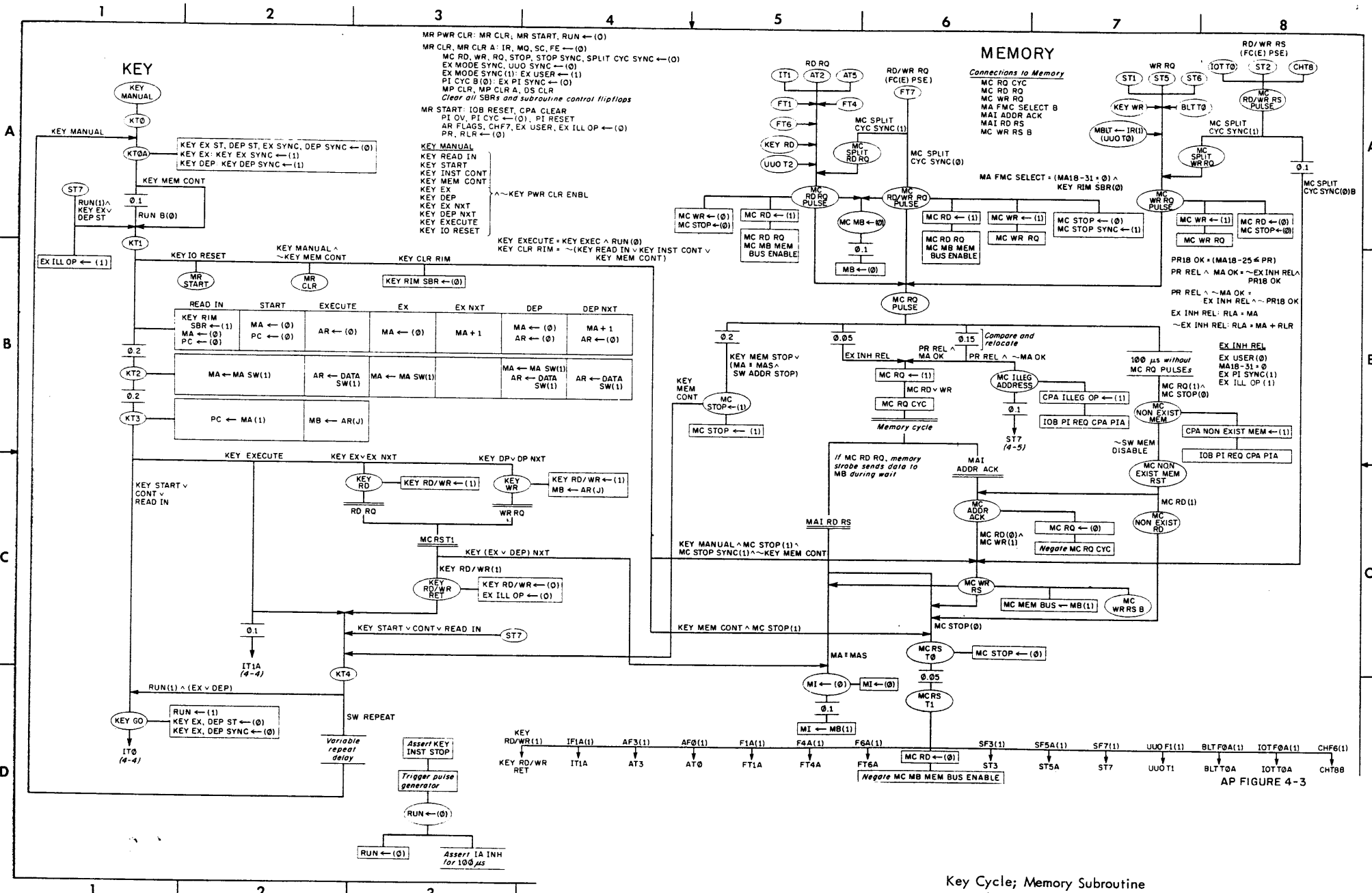


System Block Diagram
(SD-D-166-0-BD)





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, UJ0 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 OV, PI CYC ← (0); PI RESET
 AR FLAGS, CH7, EX USER, EX ILL OP ← (0)
 PR, RLR ← (0)

KEY MANUAL
 KEY READ IN
 KEY START
 KEY INST CONT
 KEY MEM CONT
 KEY EX
 KEY DEP
 KEY EX NXT
 KEY DEP NXT
 KEY EXECUTE
 KEY IO RESET
 KEY CLR RIM

KEY EXECUTE = KEY EXEC ^ RUN (0)
 KEY CLR RIM = ~(KEY READ IN ^ KEY INST CONT ^ KEY MEM CONT)

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) ^ KEY RIM SBR (0)

PR18 OK + (MA18-25 ≤ PR)
 PR REL ^ MA OK + ~EX INH REL ^ PR18 OK
 PR REL ^ ~MA OK + EX INH REL ^ ~PR18 OK
 EX INH REL: RLA + MA
 ~EX INH REL: RLA + MA + RLR

EX INH REL: EX USER (0)
 MA18-31 + 0
 EX PI SYNC (1)
 EX ILL OP (1)

100 μs without MC RQ PULSEs
 CPA NON EXIST MEM ← (1)
 IOB PI REQ CPA PIA

MC NON EXIST MEM RST
 MC RD (1)
 NON EXIST RD

MC ADDR ACK
 MC RD (0) ^ MC WR (1)
 Negate MC RO CYC

MC WR RS
 MC MEM BUS ← MB (1)
 MC WR RS B

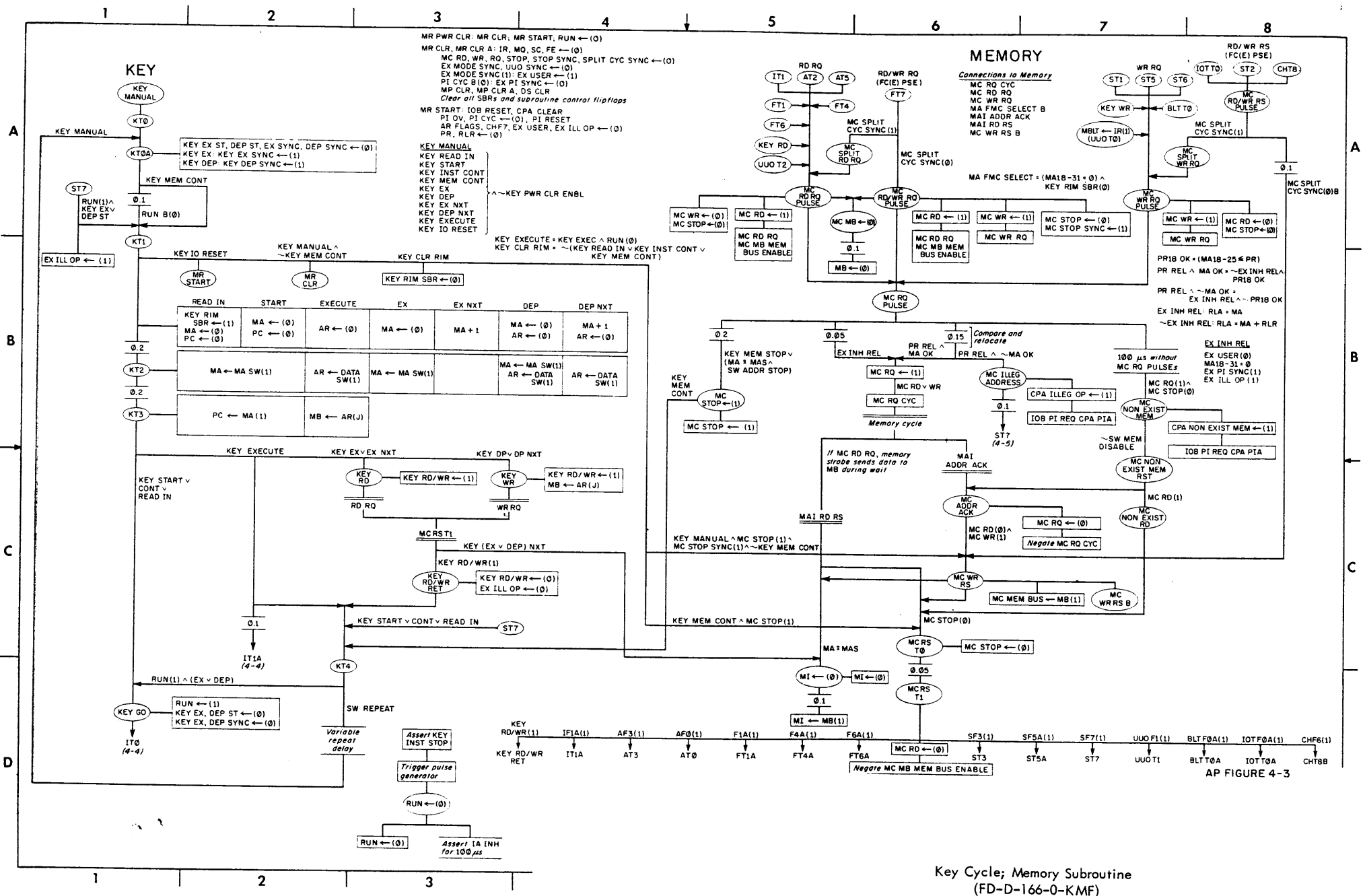
MCRS T0
 MCRS T1
 MC STOP ← (0)

MAI MAS
 MI ← (0)
 MI ← (0)
 MI ← MB (1)

MAI RD RS
 MAI ADDR ACK
 MAI RD RS

MC RD (0) ^ MC WR (1)
 Negate MC MB MEM BUS ENABLE

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, UDO SYNC ← (1)
 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 OV, PI CYC ← (0); PI RESET
 AR FLAGS, CHFT, EX USER, EX ILL OP ← (0)
 PR, RLR ← (0)

KEY MANUAL
 KEY READ IN
 KEY START
 KEY INST CONT
 KEY MEM CONT
 KEY EX
 KEY DEP
 KEY EX NXT
 KEY DEP NXT
 KEY EXECUTE
 KEY IO RESET
 KEY MANUAL
 KEY CLR RIM
 KEY RIM SBR ← (0)

KEY EXECUTE = KEY EXEC ^ RUN (0)
 KEY CLR RIM = ~(KEY READ IN ^ KEY INST CONT ^ KEY MEM CONT)

READ IN	START	EXECUTE	EX	EX NXT	DEP	DEP NXT
KEY RIM SBR ← (1) MA ← (0) PC ← (0)	MA ← (0) PC ← (0)	AR ← (0)	MA ← (0)	MA + 1	MA ← (0) AR ← (0)	MA + 1 AR ← (0)
MA ← MA SW (1)	AR ← DATA SW (1)	MA ← MA SW (1)	MA ← MA SW (1) AR ← DATA SW (1)	AR ← DATA SW (1)		
PC ← MA (1)	MB ← AR (J)					

MEMORY

Connections to Memory

- MC RD CYC
- MC RD RD
- MC WR RD
- MA FMC SELECT B
- MAI ADDR ACK
- MAI RD RS
- MC WR RS B

MA FMC SELECT = (MA18-31 = 0) ^ KEY RIM SBR (0)

PR18 OK = (MA18-25 < PR)

PR REL ^ MA OK = ~EX INH REL ^ PR18 OK

EX INH REL = MA ^ MA

~EX INH REL: RLA = MA + RLR

EX INH REL: EX USER (0)

MA18-31 = 0

EX PI SYNC (1)

EX ILL OP (1)

100 μs without MC RD PULSES

CPA ILLEG OP ← (1)

IOB PI REQ CPA PIA

CPA NON EXIST MEM ← (1)

~SW MEM DISABLE

MC NON EXIST MEM

MC NON EXIST RD

MC RD (1)

MC RD (0) ^ MC WR (1)

Negate MC RD CYC

MC WR RS

MC MEM BUS ← MB (1)

MC WR RS B

MC STOP ← (0)

MA = MAS

MI ← (0) MI ← (0)

MI ← MB (1)

MC RD ← (0)

Negate MC MB MEM BUS ENABLE

MC STOP ← (0)

MCRS T0

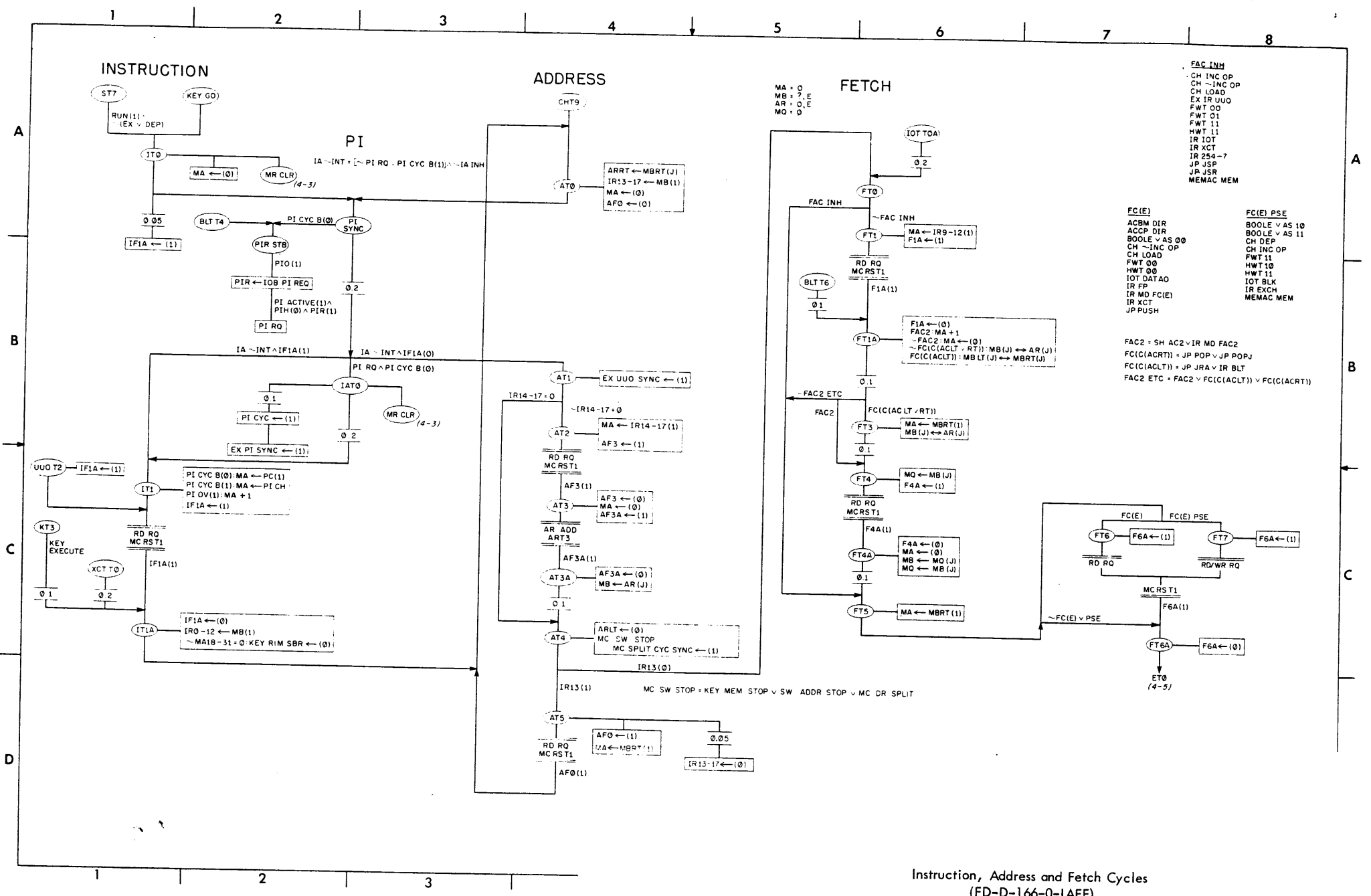
MCRS T1

MC RD ← (0)

Negate MC MB MEM BUS ENABLE

MC STOP ← (0)

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



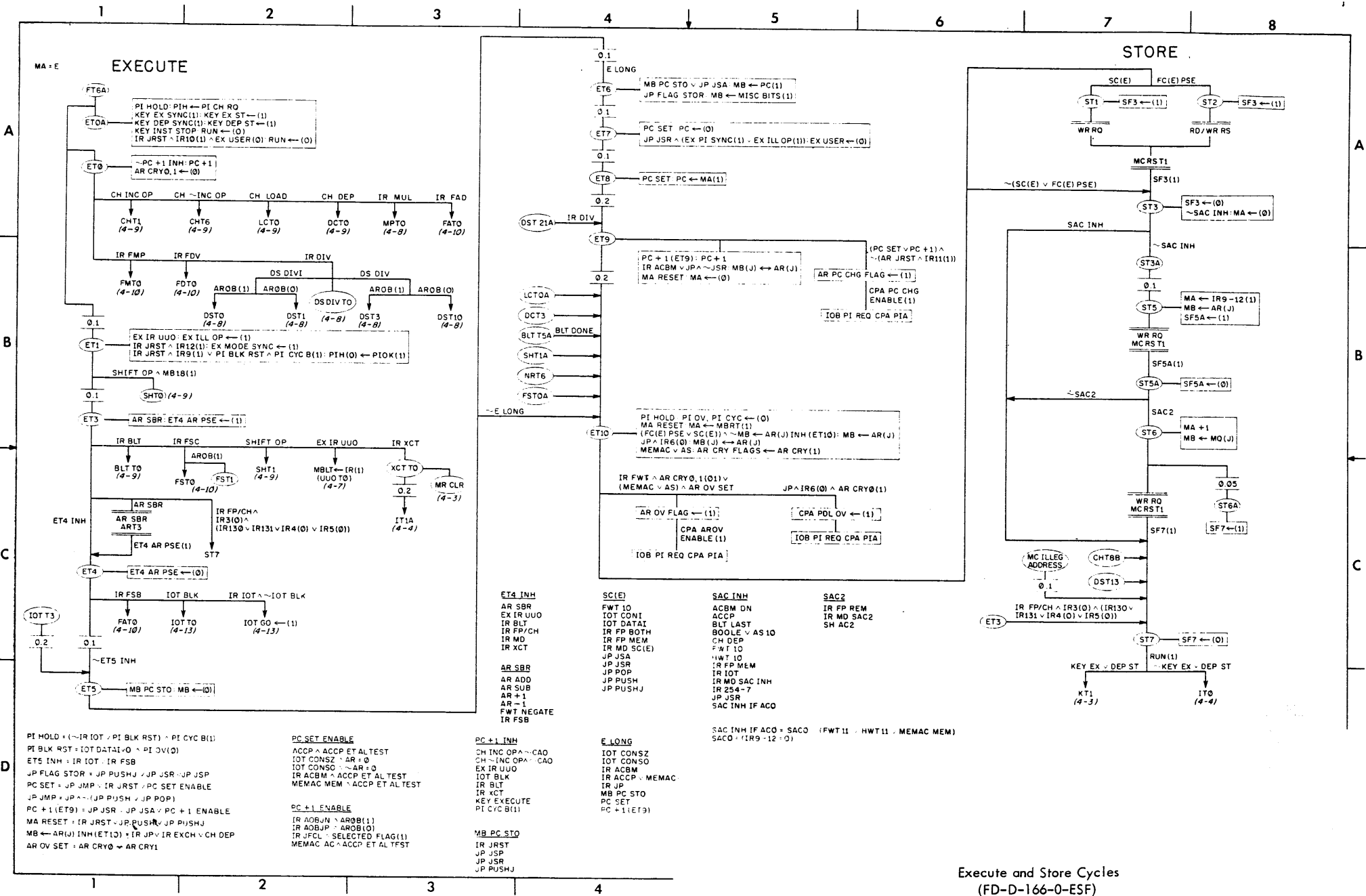
FAC INH
 CH INC OP
 CH ~INC OP
 CH LOAD
 EX IR UUO
 FWT 00
 FWT 01
 FWT 11
 HWT 11
 IR TOT
 IR XCT
 IR 254-7
 JP JSP
 JR JSR
 MEMAC MEM

FC(E)
 ACBM DIR
 ACCP DIR
 BOOLE ^ AS 00
 CH ~INC OP
 CH LOAD
 FWT 00
 HWT 00
 IOT DATAO
 IR FP
 IR MD FC(E)
 IR XCT
 JPPUSH

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))

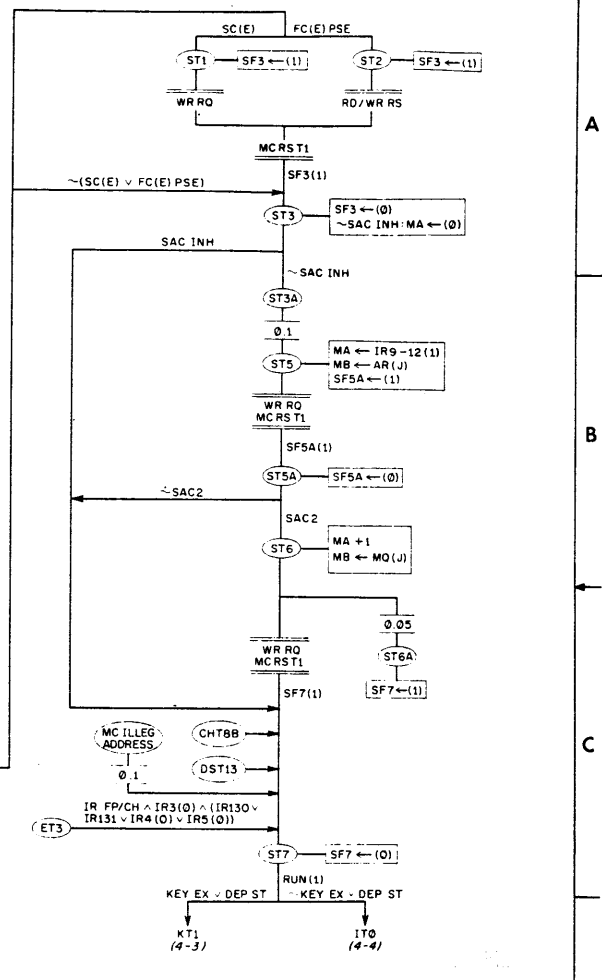
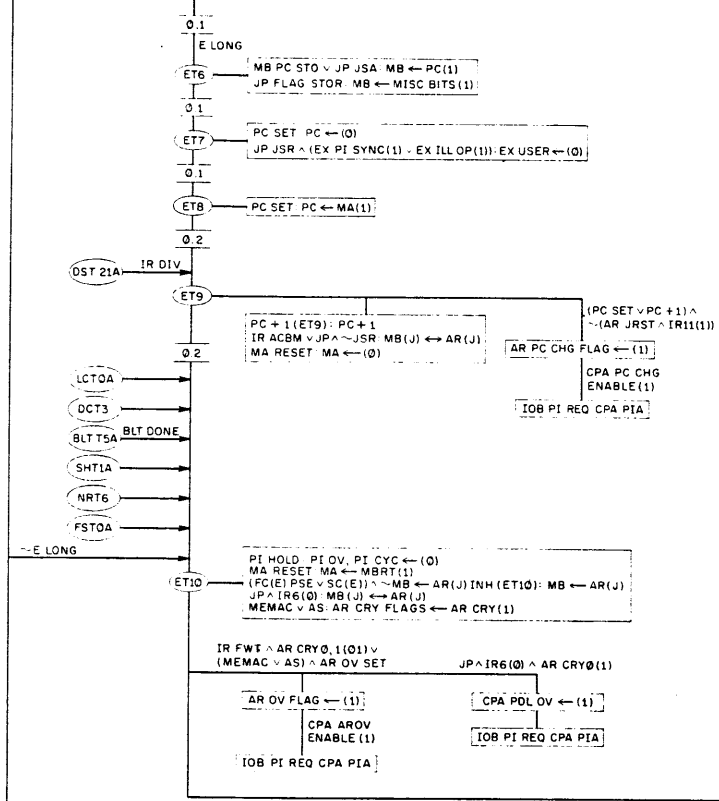
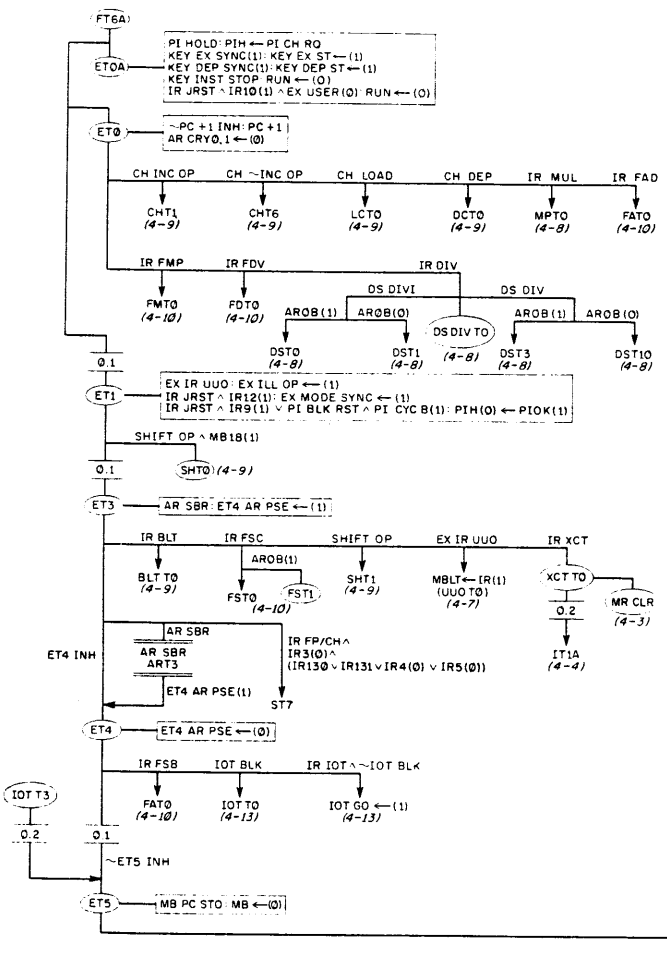
Instruction, Address and Fetch Cycles
 (FD-D-166-0-IAFF)



MA = E

EXECUTE

STORE



PI HOLD: (~IR IOT v 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 v JP JSR v JP JSP
PC SET: JP JMP v IR JRST v PC SET ENABLE
JP JMP: JP ^ (JP PUSH v JP POP)
PC + 1(ET9): JP JSR v JP JSA v PC + 1 ENABLE
MA RESET: IR JRST v JP PUSH v JP PUSHJ
MB ← AR(J) INH(ET10): IR JP v IR EXCH v CH DEP
AR OV SET: AR CRY0 ^ 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 v AROB(1)
IR AOBUP v AROB(0)
IR JCL: SELECTED FLAG(1)
MEMAC AC ^ ACCP ET AL TEST

PC + 1 INH
CH INC OPA: CAO
CH ~INC OPA: CAO
EX IR UUU
IOT BLK
IR BLT
IR FP/CH
IR MD
IR XCT
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 v MEMAC
IR JP
IR JCT
MB PC STO
PC SET
PC + 1(ET9)

SAC INH
ACBM DN
ACCP
BLT LAST
BOOLE v AS 10
CH DEP
FWT IO
HWT IO
IR FP MEM
IR IOT
IR MD SAC INH
IR 254-7
JP JSR
SAC INH IF ACO

SAC2
IR FP REM
IR MD SAC2
SH AC2

SAC INH IF ACO v SAC0 (FWT11 v HWT11 v 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 IR + 101 WXX YZZ IR HWT ← IR0-2(101) <i>W specifies destination half</i> HWT LT ← IR HWT ← IR3(0) HWT RT ← IR HWT ← IR3(1) <i>XX specifies action on other half - do nothing, zero, one, extend</i> HWT AR ← (0) ← IR HWT ← IR4(1) ∨ IR5(1) HWT RT SET ← IR5(0) ∨ MBO(1) HWT LT SET ← HWT RT ← IR4(1) ∨ IR5(0) ∨ MB18(11) <i>Y specifies source half</i> HWT SWAP ← IR HWT ← IR6(1) <i>ZZ specifies mode</i> IR HWT ← IR7,8 <table border="0"> <tr> <td>00</td><td>← HWT 00</td><td>∕ C(E), AC</td><td>→ AC</td> </tr> <tr> <td>01</td><td>← HWT 01</td><td>∕ (0,E), AC</td><td>→ AC</td> </tr> <tr> <td>10</td><td>← HWT 10</td><td>∕ AC, C(E)</td><td>→ AC</td> </tr> <tr> <td>11</td><td>← HWT 11</td><td>∕ C(E)</td><td>→ E</td> </tr> </table>			00	← HWT 00	∕ C(E), AC	→ AC	01	← HWT 01	∕ (0,E), AC	→ AC	10	← HWT 10	∕ AC, C(E)	→ AC	11	← HWT 11	∕ C(E)	→ E	FULL WORD TRANSFER 200-217 IR + 010 O0Y YZZ IR 2XX ← IR0-2(010) <i>YY specifies instruction - MOVE, MOVN, MOVN, MOVN</i> IR MOVN, M ← IR 2XX ∨ IR3-5(0001) IR MOVN, M ← IR 2XX ∨ IR3-5(0001) IR FWT ← IR MOVN, M FWT SWAP ← IR FWT ← IR5,6(01) FWT NEGATE ← IR MOVN, M ∨ IR6(0) ∨ ARO(1) <i>ZZ specifies mode</i> <table border="0"> <tr> <td>00</td><td>← FWT 00</td><td>∕ C(E)</td><td>→ AC</td> </tr> <tr> <td>01</td><td>← FWT 01</td><td>∕ 0,E</td><td>→ AC</td> </tr> <tr> <td>10</td><td>← FWT 10</td><td>∕ AC</td><td>→ E</td> </tr> <tr> <td>11</td><td>← FWT 11</td><td>∕ C(E)</td><td>→ E</td> </tr> </table> IR FWT ← IR7,8		00	← FWT 00	∕ C(E)	→ AC	01	← FWT 01	∕ 0,E	→ AC	10	← FWT 10	∕ AC	→ E	11	← FWT 11	∕ C(E)	→ E	EXCH 250 IR + 010 101 000 IR 2XX ← IR0-2(010) IR 25X ← IR 2XX ∨ IR3-5(101) IR EXCH ← IR 25X ∨ IR6-8(000) AC → C(E)	ARITHMETIC COMPARE 300-377 IR + 011 VVV XYZ IR ACCP ∨ MEMAC ← IR0-2(011) <i>VV specifies instruction type</i> ((IR ACCP ∨ MEMAC) ∨ IR3,4) <table border="0"> <tr> <td>00</td><td>← ACCP</td> </tr> <tr> <td>01</td><td>← MEMAC TST</td> </tr> <tr> <td>10</td><td>← MEMAC +1</td> </tr> <tr> <td>11</td><td>← MEMAC -1</td> </tr> </table> MEMAC ← MEMAC TST ∨ MEMAC +1 ∨ MEMAC -1 <i>X specifies whether or not action is on condition YZ</i> <i>Y specifies condition "equals"</i> <i>Z specifies condition "less than"</i> XYZ together define relation R <table border="0"> <tr> <td>XYZ</td><td>R</td> </tr> <tr> <td>000</td><td>Never</td> </tr> <tr> <td>001</td><td><</td> </tr> <tr> <td>010</td><td>=</td> </tr> <tr> <td>011</td><td>≤</td> </tr> <tr> <td>100</td><td>Always</td> </tr> <tr> <td>101</td><td>≥</td> </tr> <tr> <td>110</td><td>></td> </tr> <tr> <td>111</td><td>></td> </tr> </table> Determination of Skip or Jump Condition ACCP ETC COND ← ((IR7(1) ∨ (AR+0)) ∨ ((IR ACCP ∨ MEMAC) ∨ IR6(1) ∨ (AR0 ← AR OV)) ACCP ET AL TEST ← ACCP ETC COND ∨ IR6 ARO ← AR OV ∨ ARO ← (AR OV SET ∨ MEMAC) AR OV SET ← AR CRY0 ∨ AR CRY1			00	← ACCP	01	← MEMAC TST	10	← MEMAC +1	11	← MEMAC -1	XYZ	R	000	Never	001	<	010	=	011	≤	100	Always	101	≥	110	>	111	>	LOGICAL COMPARE 600-677 IR + 110 VVV XYZ IR ACBM ← IR0-2(110) <i>VV specifies action on masked bits</i> IR ACBM ∨ (0) ← ACBM CL IR3,4 ∨ (0) ← ACBM COM 11 ← ACBM SET <i>W specifies source of mask - immediate or memory</i> ACBM DIR ← IR ACBM ∨ IR5(1) <i>X specifies whether or not action is on condition Y, i.e. equals zero</i> XY together define relation R <table border="0"> <tr> <td>XY</td><td>R</td> </tr> <tr> <td>00</td><td>Never</td> </tr> <tr> <td>01</td><td>W</td> </tr> <tr> <td>10</td><td>Always</td> </tr> <tr> <td>11</td><td>W</td> </tr> </table> <i>Z specifies swapping mask selected by W</i> ACBM SWAP ← IR ACBM ∨ IR6(1) WZ together specify mask as (0,E), (E,0), C(E), C(E) RT LT		XY	R	00	Never	01	W	10	Always	11	W	A0BJP 252 IR + 010 101 01X IR 2XX ← IR0-2(010) IR 25X ← IR 2XX ∨ IR3-5(101) <i>X specifies instruction</i> IR A0BJP ← IR 25X ∨ IR6-8(010) IR A0BJN ← IR 25X ∨ IR6-8(011) AC + 1000001 → AC A0BJP ∨ (AC ← 0) ∨ E → PC A0BJN ∨ (AC ← 0) ∨ E → PC IR A0BJP ∨ AROB(0) ∨ PC SET IR A0BJN ∨ AROB(1) ∨ PC SET
00	← HWT 00	∕ C(E), AC	→ AC																																																																													
01	← HWT 01	∕ (0,E), AC	→ AC																																																																													
10	← HWT 10	∕ AC, C(E)	→ AC																																																																													
11	← HWT 11	∕ C(E)	→ E																																																																													
00	← FWT 00	∕ C(E)	→ AC																																																																													
01	← FWT 01	∕ 0,E	→ AC																																																																													
10	← FWT 10	∕ AC	→ E																																																																													
11	← FWT 11	∕ C(E)	→ E																																																																													
00	← ACCP																																																																															
01	← MEMAC TST																																																																															
10	← MEMAC +1																																																																															
11	← MEMAC -1																																																																															
XYZ	R																																																																															
000	Never																																																																															
001	<																																																																															
010	=																																																																															
011	≤																																																																															
100	Always																																																																															
101	≥																																																																															
110	>																																																																															
111	>																																																																															
XY	R																																																																															
00	Never																																																																															
01	W																																																																															
10	Always																																																																															
11	W																																																																															
A	Initial Registers HWT 00: AR ← AC MB ← C(E) HWT 01: AR ← AC MB ← 0,E HWT 10: AR ← 0,E 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 ← 0,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																																																																										
B	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)																																																																										
	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)																																																																										
	ET1 HWT SWAP: MBLT(J) ← MBRT(J) HWT AR ← (0): AR ← (0)			FWT SWAP: MBLT(J) ← MBRT(J)																																																																												
	ET3 HWT LT SET: AR LT COM HWT RT SET: AR RT COM HWT LT: AR ← MB LT(J) HWT RT: AR ← MB RT(J)			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 HWT 10: 11: MB ← AR(J)			AR CRY0, 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																																																																										
C	Initial Registers AR ← AC MB ← 0,E MQ ← 0			MEMAC ±1: ET4 INH AR SBR		FAC INH FC(E) PSE MEMAC ±1: ET4 INH AR SBR E LONG	ACCP DIR: FC(E) ET4 INH AR SBR E LONG																																																																									
	Initial Gates E LONG			MEMAC ±1: ET4 INH AR SBR		FAC INH FC(E) PSE MEMAC ±1: ET4 INH AR SBR E LONG	ACCP DIR: FC(E) ET4 INH AR SBR E LONG																																																																									
	ET0A AR ← MB(J)			AR ← MB(J)			ACBM SWAP: MBLT(J) ↔ MBRT(J)																																																																									
	ET1 MB ← AR(0) ACBM COM: AR ← MB(←) ACBM SET: AR ← MB(1)			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 AR COM			AR COM			ET4 AR PSE ← (0)																																																																									
	ET6 ACBM CL: MB ← AR(0)			AR COM			PC SET: PC ← (0)																																																																									
	ET7 PC SET: PC ← (0)			AR COM			PC SET: PC ← (0)																																																																									
	ET8 PC SET: PC ← MA(1)			PC SET: PC ← MA(1)			PC SET: PC ← MA(1)																																																																									
	ET9 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)																																																																									
	ET10 AR OV SET: AR OV FLAG ← (1) AR CRY FLAGS ← AR CRY(1)			AR OV SET: AR OV FLAG ← (1) AR CRY FLAGS ← AR CRY(1)		MB ← AR(J)	PC SET: AR PC CHG FLAG ← (1)																																																																									
D	Final Gates FC(E) PSE SAC0: SAC INH			FC(E) PSE SAC0: SAC INH		SAC INH	ACBM ON: 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

IR JP ^ IR6-8

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

JP JMP = IR JP ^ (JP PUSH ^ JP POP)
JP FLAG STOR = JP PUSHJ ^ JP JSR ^ JP JSP
~IR 10T ^ 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)

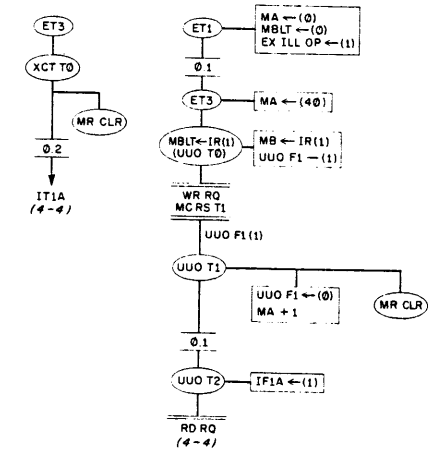
XCT

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

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) ^ IR JOT A ^ EX USER B(1) ^ EX PI SYNC(0) ^ CPA IOT USER(0)
IR, E → 40
EXECUTE 41

	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	AC+1000001 → AC MISC BITS, PC → C(CIACRT) E → PC PI ON OVERFLOW	AC+1000001 → AC C(E) → C(CIACRT) PI ON OVERFLOW	C(CIACRT) → E AC-1000001 → AC PI ON UNDERFLOW	C(CIACRT) → PC AC-1000001 → AC PI ON UNDERFLOW	MISC BITS, PC → E E + 1 → PC	MISC BITS, PC → AC E → PC	AC → E E, PC → AC E + 1 → PC	C(CIACLT) → AC E → PC	E → PC IR9(1): DISMISS PI IR10(1): HALT IR11(1): RESTORE MISC BITS IR12(1): ENTER USER MODE	SELECTED FLAG(1): E → PC CLEAR SELECTED FLAGS	EXECUTE C(E)	IR, E → 40 EXECUTE 41
Initial Gates	AR = AC MB = 0, E MQ = 0	AR = AC MB = C(E) MQ = 0	AR = AC MB = 0, E MQ = C(CIACRT)	AR = AC MB = 0, E MQ = C(CIACRT)	AR = 0, E MB = 7, E MQ = 0	AR = 0, E MB = 7, E MQ = 0	AR = AC MB = 0, E MQ = 0	AR = ACRTL 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
ET0A	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(CIACRT) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(CIACRT) 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
ET0B			MB ← MQ(J)	MB ← MQ(J)	PI HOLD: PIH ← PI CHRO	MB(J) ↔ AR(J)	MB(LT) ↔ MB(RT)(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) ← PTOK(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) EX PI SYNC(1) ^ EX ILL OP(1): EX USER ← (0)	PC ← (0)	PC ← (0)	PC ← (0)			SELECTED FLAG(1): PC ← (0)	
ET8	PC ← MA(1)				PC ← MA(1) EX ILL OP ← (0)	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)		SELECTED FLAGS ← (0)	
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 | |

MULTIPLY-DIVIDE

220 - 237

Levels for fetch and store according to operand type and mode

IR 010 01X YZZ
IR 2XX + IR0-2(C)10
X specifies operation

IR MD = IR 2XX + IR3(0) - IR4(1)
IR MD FC(E) = IR MD - IR7(0) - IR8(1)
IR MD FAC2 = IR DIV - IR6(1)
IR MD SC(E) = IR MD - IR7(1)
IR MD SAC INH = IR MD - IR7(1) - IR8(0)
IR MD SAC2 = IR MD SAC INH - IR DIV - IR MUL - IR6(1)
IR 2XX + IR3-5 - 010 = IR MUL
IR 2XX + IR3-5 - 011 = IR DIV
Y specifies operand type: IR6 is gate in multiply flow; for divide DS DIV1 = IR DIV - IR6(0)
DS DIV = IR DIV - IR6(1)
ZZ specifies mode

IR MUL
Action
For IMUL(XY=00) store only low order half of product
ZZ = 00: AC - C(E) → AC
ZZ = 01: AC - (0,E) → AC
ZZ = 01: AC - (0,E) → AC
ZZ = 10: AC - C(E) → E
ZZ = 11: AC - C(E) → AC, E

IR DIV
Action
For MUL(XY=01) store entire product or high order half
ZZ = 00: AC - C(E) → AC, AC2
ZZ = 01: AC - (0,E) → AC, AC2
ZZ = 10: AC - C(E) → E
ZZ = 11: AC - C(E) → AC, AC2, E

Initial Registers
AR = AC
ZZ = 00 - 10 - 11: MB + C(E)
ZZ = 01: MB + 0, E
MQ = 0

Initial Gates
ZZ = 00 - 10 - 11: FC(E)
ET4 INH

IR MUL
Action
For IMUL(XY=00) store only low order half of product
ZZ = 00: AC - C(E) → AC, AC2
ZZ = 01: AC - (0,E) → AC, AC2
ZZ = 01: AC - (0,E) → AC
ZZ = 10: AC - C(E) → E
ZZ = 11: AC - C(E) → AC, AC2, E

Initial Registers
AR = AC
ZZ = 00 - 10 - 11: MB + C(E)
ZZ = 01: MB + 0, E
MQ = 0

Initial Gates
ZZ = 00 - 10 - 11: FC(E)
ET4 INH

DIV(XY=10)
Action
ZZ = 00: AC - C(E) → AC, AC2
ZZ = 01: AC - (0,E) → AC, AC2
ZZ = 10: AC - C(E) → E
ZZ = 11: AC - C(E) → AC, AC2, E

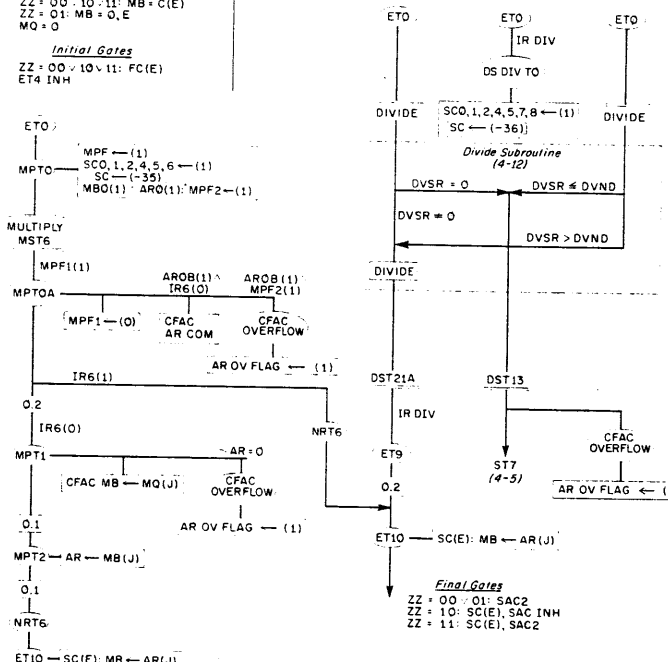
Initial Registers
AR = AC
ZZ = 00 - 10 - 11: MB + C(E)
ZZ = 01: MB + 0, E
MQ = 0

Initial Gates
ZZ = 00 - 10 - 11: FC(E)
ET4 INH

DIV(XY=11)
Action
ZZ = 00: AC, AC2 = C(E) → AC, AC2
ZZ = 01: AC, AC2 = (0,E) → AC, AC2
ZZ = 10: AC, AC2 = C(E) → E
ZZ = 11: AC, AC2 = C(E) → AC, AC2, E

Initial Registers
AR = AC
ZZ = 00 - 10 - 11: MB + C(E)
ZZ = 01: MB + 0, E
MQ = 0

Initial Gates
ZZ = 00 - 10 - 11: FC(E)
ET4 INH



Final Gates
ZZ = 00 - 01: SAC2
ZZ = 10: SC(E), SAC INH
ZZ = 11: SC(E), SAC2

AR OV FLAG ← (1)
CPA AROV ENABLE (1)
IOB PI REQ CPA PIA

BOOLEAN

400-477

IR = 100 XXX YYY
IR BOOLE = IR0-2(100)
XXXX specifies Boolean function, decoded into BOOLE-0-17 as BOOLE N = IR BOOLE - IR3-6(N)

ADD-SUBTRACT

270-277

IR = 010 111 XYY
IR 2XX = IR0-2(010)
IR AS = IR 2XX + IR3-5 (111)
X specifies add or subtract
IR AS + IR6 - 0 = AS +
IR AS + IR6 - 1 = AS -

In both instruction groups YY specifies common mode

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, (0,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 + (0,E)
MQ = 0

Common Initial Gates

BOOLE + AS 00: FC(E)
BOOLE + AS 10 - 11: FC(E) PSE

Initial Gates AS Only

ET4 INH
AR SBR

ET0A	BOOLE 0,3,14,17: AR ← (0) BOOLE 2,4,12,13,15: AR COM
ET1	BOOLE 6,11,14: AR ← MB(-) BOOLE 1,2,15,16: AR ← MB(0) BOOLE 3,4,7,10,13: AR ← MB(1)
ET3	ET4 AR PSE ← (1) AS + AR ADD AR3 AS - AR SUB
ET4	BOOLE 4,10,11,14,15,16,17: AR COM
ET5	ET4 AR PSE ← (0)
ET10	FC(E) PSE: MB ← AR(J) AR OV SET: AR OV FLAG ← (1) AR CRY FLAGS ← AR CRY(1) AR OV SET: AR CRY0 - AR CRY1

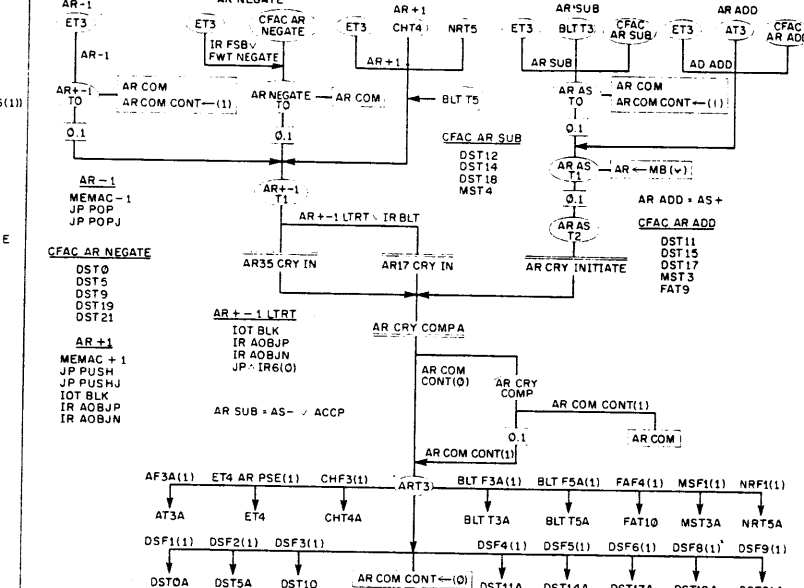
Final Gates

BOOLE + AS 10: FC(E) PSE, SAC INH
BOOLE + AS 11: FC(E) PSE

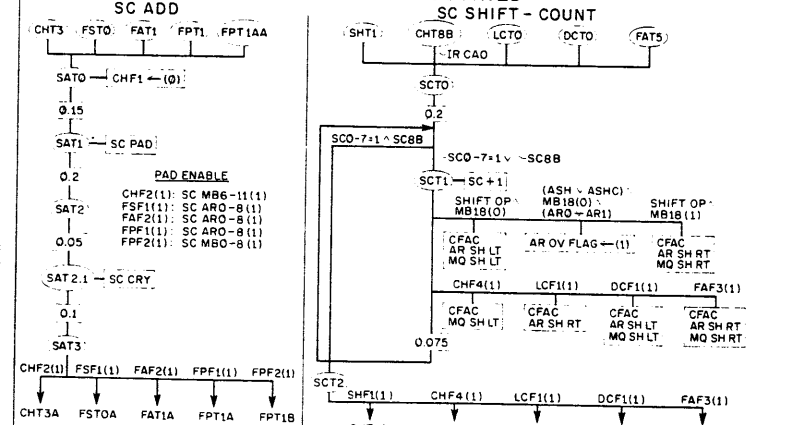
BOOLEAN FUNCTIONS

0 SETZ	1 AND	2 ANDCA	3 SETM
ET0A AR ← (0)	AR COM	AR COM	AR ← (0)
ET1 AR ← MB(1)	AR ← MB(0)	AR ← MB(1)	AR ← MB(1)
ET4 AR COM			
4 ANDCM	5 SETA	6 XOR	7 IOR
ET0A AR COM	AR COM	AR ← MB(-)	AR ← MB(1)
ET1 AR ← MB(1)	AR ← MB(0)	AR ← MB(1)	AR ← MB(1)
ET4 AR COM			
10 ANDCB	11 EQV	12 SETCA	13 ORCA
ET0A AR ← MB(1)	AR ← MB(0)	AR COM	AR COM
ET1 AR ← MB(1)	AR ← MB(0)	AR ← MB(1)	AR ← MB(1)
ET4 AR COM			
14 SETCM	15 ORCM	16 ORCB	17 SETO
ET0A AR ← (0)	AR COM	AR ← MB(-)	AR ← (0)
ET1 AR ← MB(1)	AR ← MB(0)	AR ← MB(0)	AR ← MB(0)
ET4 AR COM	AR COM	AR COM	AR COM

AR SUBROUTINES



SHIFT COUNTER SUBROUTINES



AP FIGURE 4-8

CHARACTER OPERATIONS

133-137

FIRST PART-CHF5(0)

CH INC + (IR LDC1 ∨ IR DPC1 ∨ IR CAO) · CHF5(0)
 CH INC OP = CH INC · CHF7(0)
 CH ~ INC OP = (IR LDC ∨ IR DPC) · CHF5(0) ∨
 CH INC · CHF7(1)

Pointer Format

P	S	I	X	Y
0-5	6-11	12	13	14-17
18-35				

Fetch and increment pointer

Initial Registers

AR = 0, E
 MB = C(E)
 MQ = 0

Initial Gates

CH INC OP: FAC INH, FC(E) PSE
 CH ~ INC OP: FAC INH, FC(E)
 ~IR CAO: PC + 1 INH
 ET4 INH

IR = 001 011 xxx
 IR FP/CH = IR0-2(001)
 IR FP/CH3 = IR FP/CH ∨ IR3-5(011)
 xxx specifies instruction

011 = IR CAO	100 = IR LDC1
101 = IR LDC	110 = IR DPC1
111 = IR DPC	

SECOND PART-CHF5(1)

CH LOAD = (IR LDC ∨ IR LDC1) · CHF5(1)
 CH DEP = (IR DPC ∨ IR DPC1) · CHF5(1)

CH LOAD

CH DEP

Fetch and load character

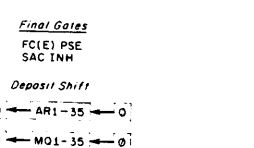
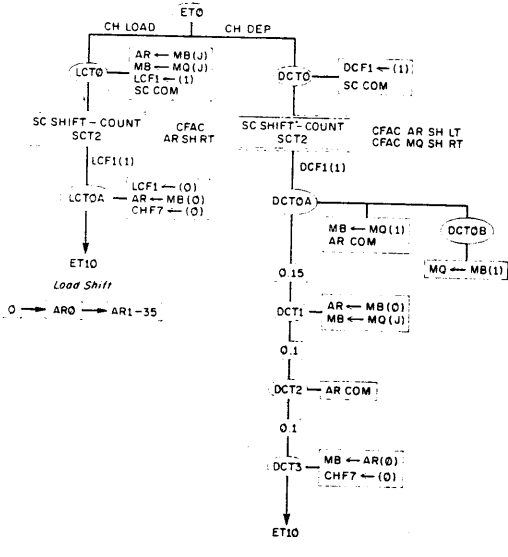
Fetch and deposit character

Initial Registers

AR = 0, E	AR = 0, E
MB = C(E)	MB = C(E)
MQ = 0...0 1...1	MQ = 0...0 1...1

Initial Gates

FAC INH	FC(E) PSE
FC(E)	ET4 INH
ET4 INH	MB ← AR(J) INH(ETO)



BLT

251

IR = 010 101 001
 IR 2XX = IR0-2(010)
 IR 25X = IR 2XX ∨ IR3-5(101)
 IR BLT = IR 25X ∨ IR6-8(001)
 BLT DONE = MQ(0) ∨ PI RQ
 BLT LAST = IR BLT ∨ MQ(0)

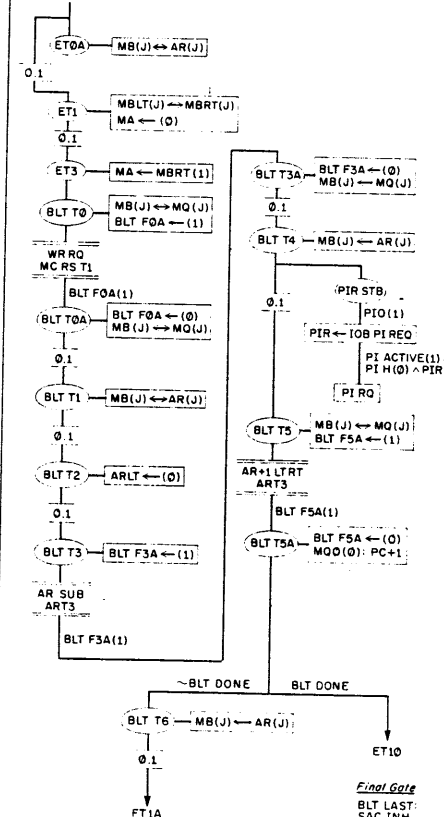
S = Source address
 D = Destination address
 C(S) → D
 (S, D) + 10000001 → (S, D)
 Repeat until D = E

Initial Registers

AR = D, S initially AC RT LT
 MB = 0, E
 MQ = C(S) initially C(C(ACLT))

Initial Gates

FC(C(ACLT))
 PC + 1 INH
 ET4 INH



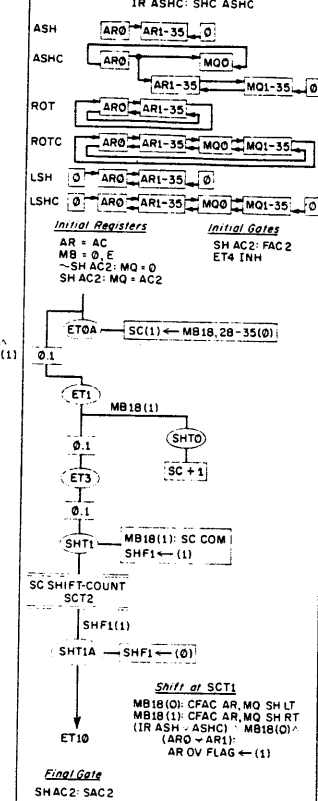
SHIFT OPERATIONS

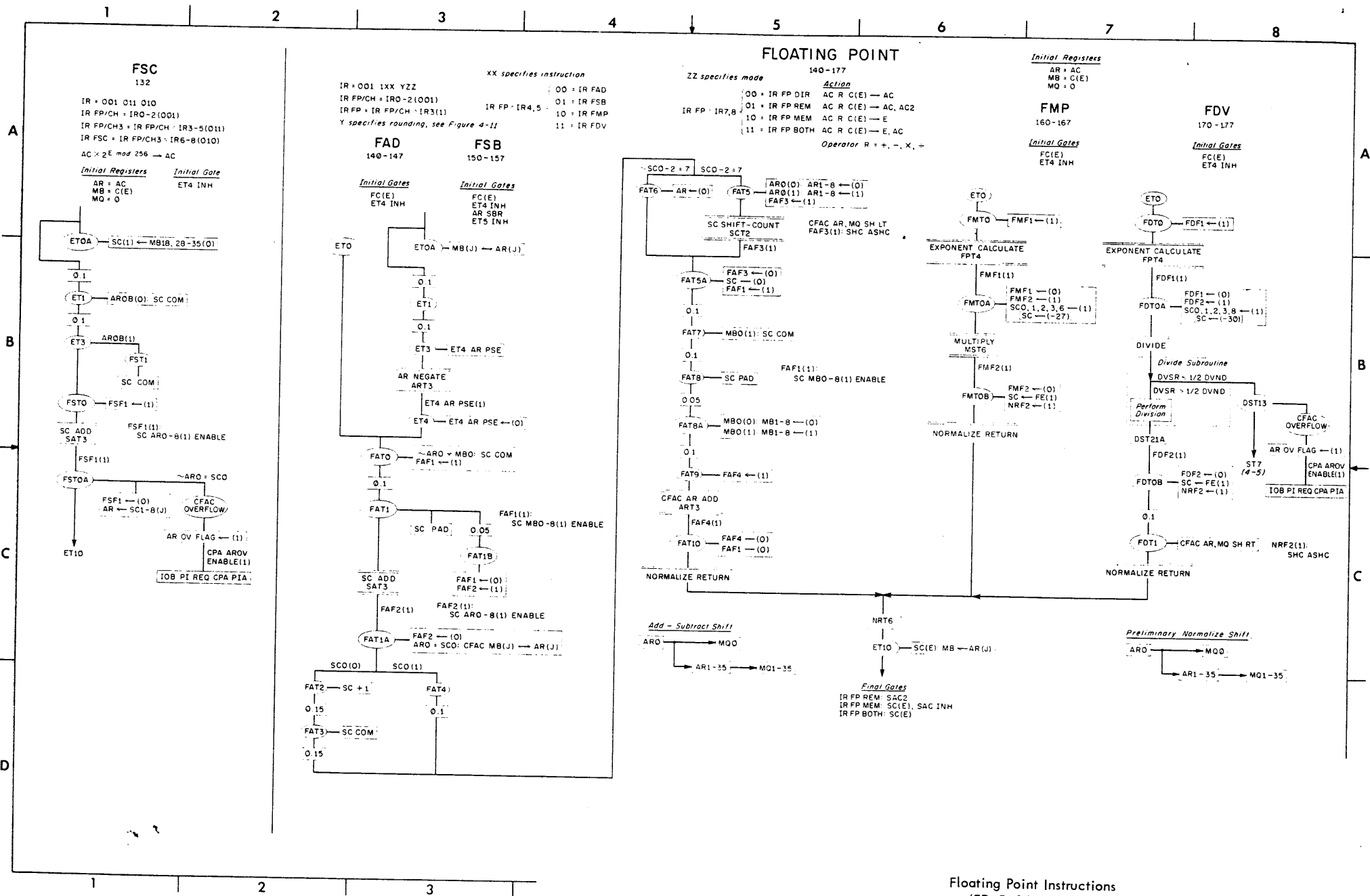
240-242, 244-246

IR = 010 100 xxx
 IR 2XX = IR0-2(010)
 IR SH = IR 2XX ∨ IR3-5(100)

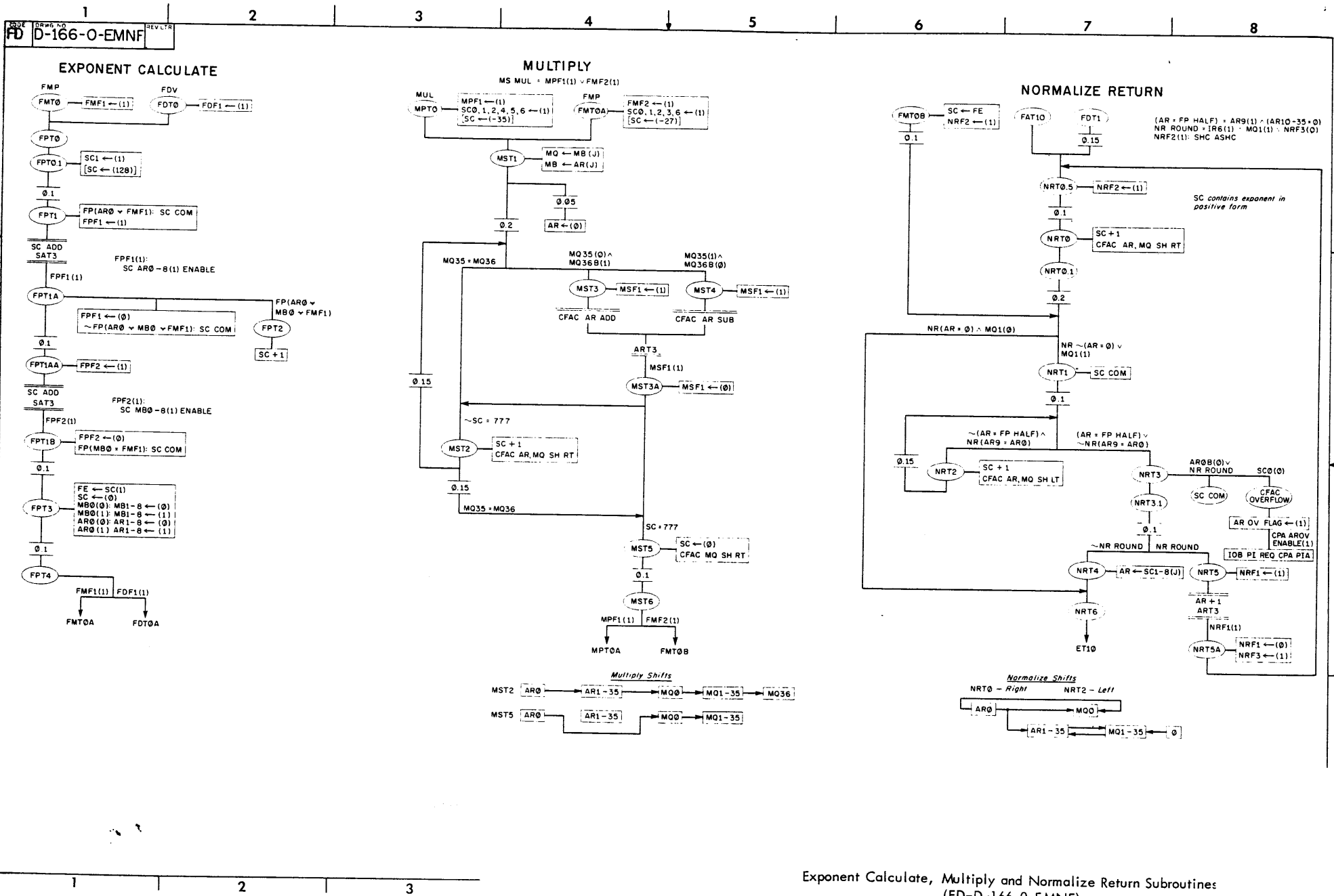
000 = IR ASH	001 = IR ROT
010 = IR LSH	100 = IR ASHC
101 = IR ROTC	110 = IR LSHC

SHIFT OP = IR ASH ∨ IR ROT ∨ IR LSH ∨
 IR ASHC ∨ IR ROTC ∨ IR LSHC
 SH AC2 = IR ASHC ∨ IR ROTC ∨ IR LSHC
 Shift E mod 256 places
 ~SH AC2: / [AC] → AC
 SH AC2: / [AC, AC2] → AC
 SHIFT OP AR, MQ CONNECTIONS
 IR ASHC: SHC ASHC

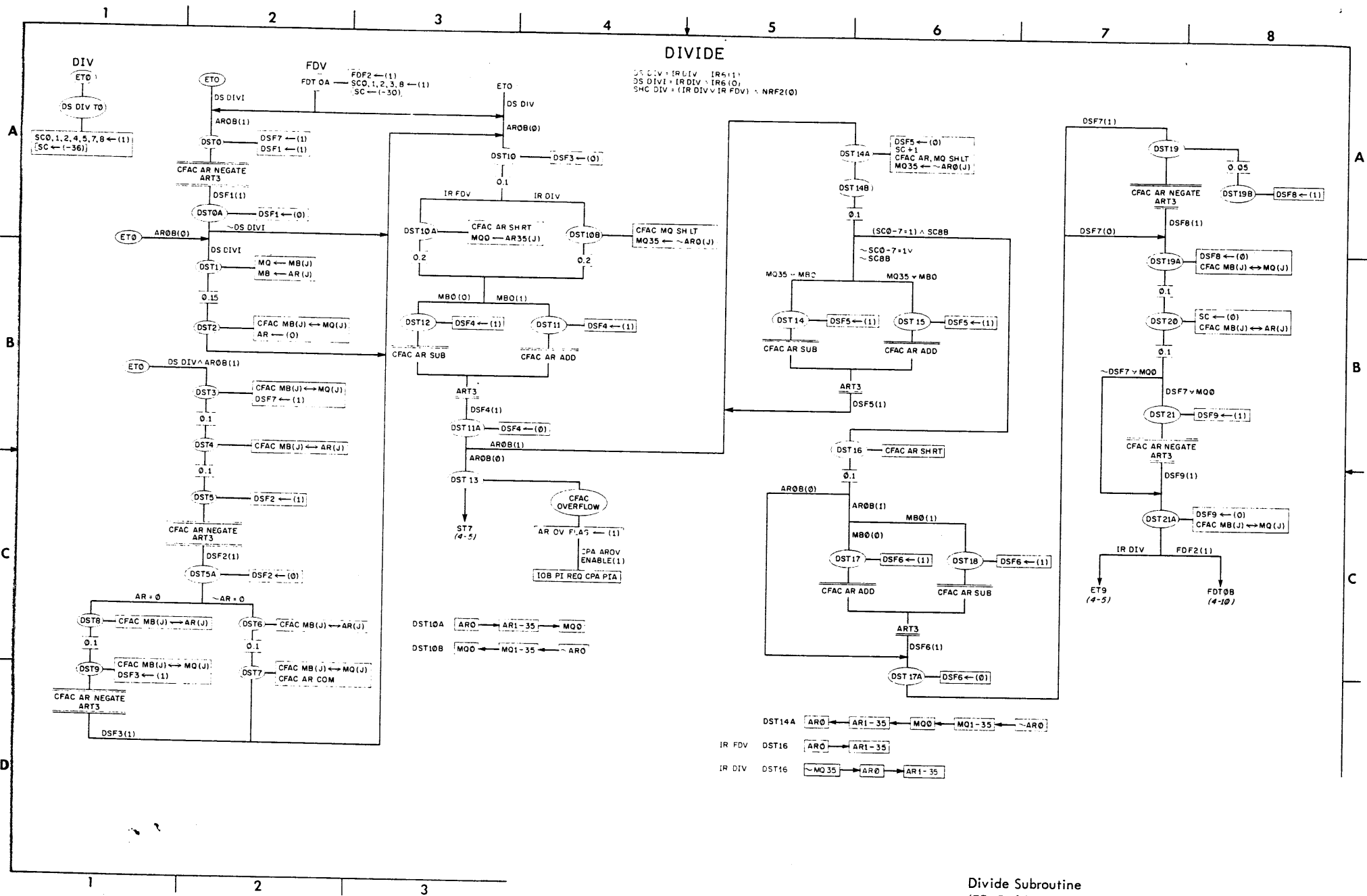




Floating Point Instructions
(FD-D-166-0-FPP)



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



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

IN-OUT TRANSFER

IR = 111 XXX XXX XXX 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 IO33-9 lines on IO bus. Each IO control unit contains a decoding net that allows program code to select device by gating in command signals from bus

Instruction Action

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/O = IOT DATAI ^ IOT DATAO
 PI BLK RST = IOT DATAI ^ IOT DATAO
 PI BLK RST ^ PI CYC B(1): PI HOLD, PI RST
 IOT CONSZ ^ IOT CONSO: E LONG

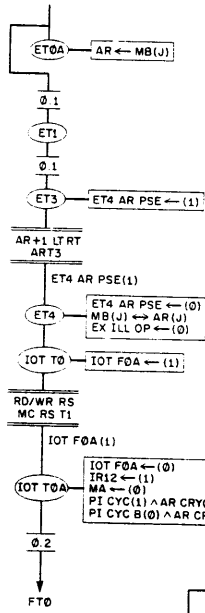
BLK

Initial Registers

MB = C(E)
 AR = 0, E
 MQ = 0

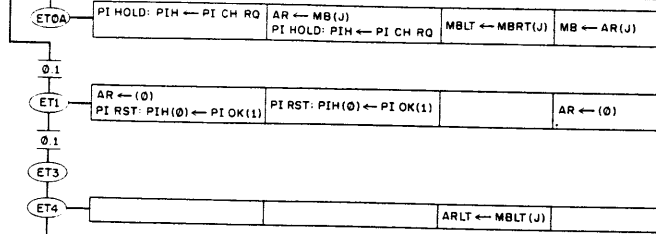
Initial Gates

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

Initial Registers	DATAI	DATAO	CONO	STATUS
	MB = 7, E AR = 0, E MQ = 0	MB = C(E) AR = 0, E MQ = 0	MB = 7, E AR = 0, E MQ = 0	MB = 7, E AR = 0, E MQ = 0
Initial Gates	FAC INH ETS INH	FAC INH FC(E) ETS INH	FAC INH ETS INH	FAC INH ETS INH



IOT GO ← (1)
 IOT RESET
 ~ IOT RESET
 Wait for
 ~ IOT RESET

INITIAL SETUP 1.0

IOT T2

IOT DATAO

IOT CONO

IOT GO ← (0)

IOT DATAI CLEAR

IOT CONO CLEAR

RESTART 1.0

IOT T3

IOT DATAO

IOT CONO

AR ← IOB(1)

IOT DATAI SET

IOT CONO SET

RESET 2.0

~E LONG

ET10

IOT DATAO

IOT DATAI

IOT CONI

PI HOLD: PI CYC ← (0)

MB ← AR(J)

PI HOLD: PI CYC ← (0)

MB ← AR(J)

E LONG

ET6

AR ← MB(0)

ET7

ET8

ET9

IOT CONSZ

IOT CONSO

AR = 0: PC + 1

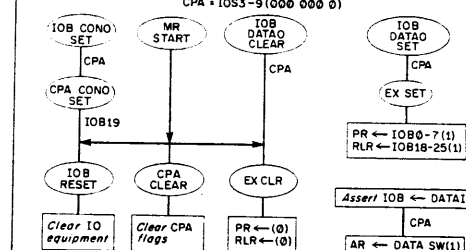
~AR = 0: PC + 1

ET10

Final Gates

IOT DATAI ^ IOT CONI: SC(E)
 SAC INH

CPA

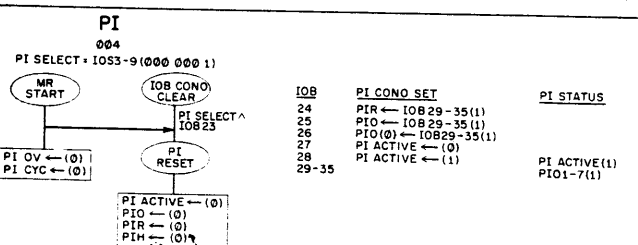


IOB	CPA CONO SET	CPA STATUS
18	CPA PDL OV ← (0)	CPA PDL OV(1)
19	IOB RESET	CPA IOT USER(1)
20	CPA IOT USER ← (1)	EX USER B(1)
21	CPA IOT USER ← (0)	CPA ILLEG OP(1)
22	CPA ILLEG OP ← (0)	CPA NON EXIST MEM(1)
23	CPA NON EXIST MEM ← (0)	CPA NON EXIST MEM(1)
24	CPA CLOCK ENABLE ← (0)	CPA CLOCK ENABLE(1)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

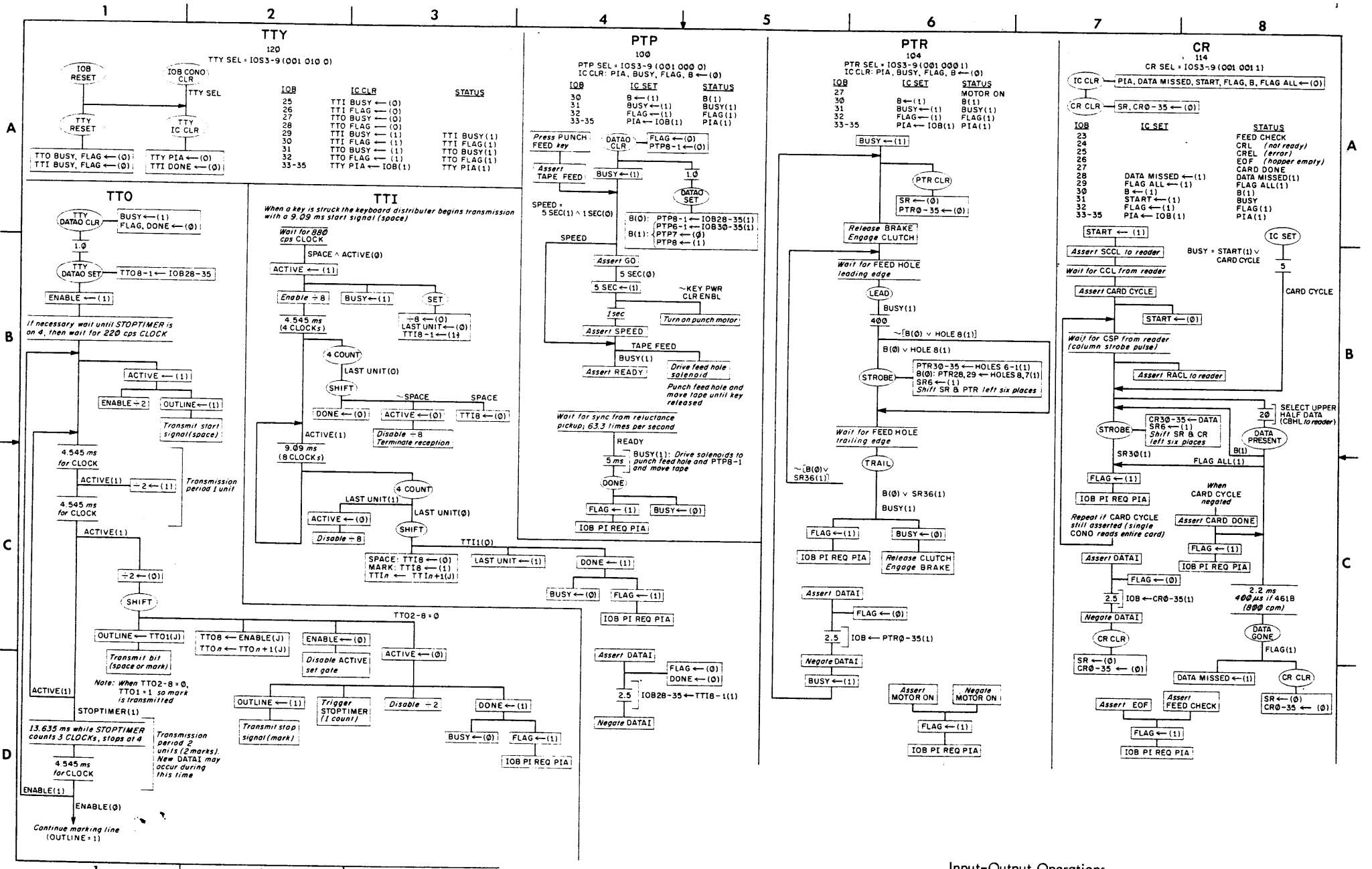
IOB PI REQ CPA PIA

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)

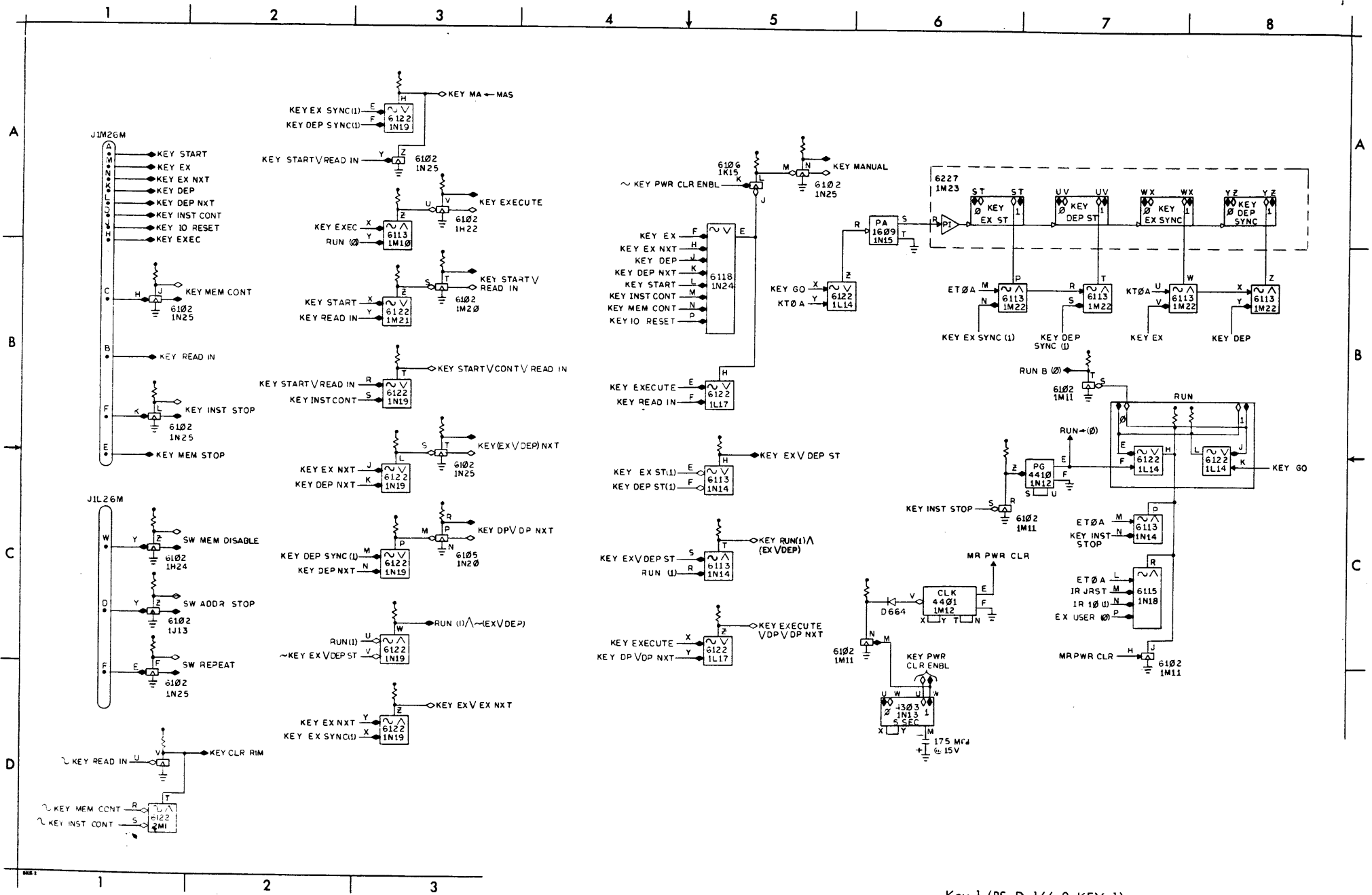


PI	PI CONO SET	PI STATUS
24	PI R ← IOB29-35(1)	
25	PI O ← IOB29-35(1)	
26	PI I(0) ← IOB29-35(1)	
27	PI ACTIVE ← (0)	
28	PI ACTIVE ← (1)	PI ACTIVE(1)
29-35		PI OI-7(1)

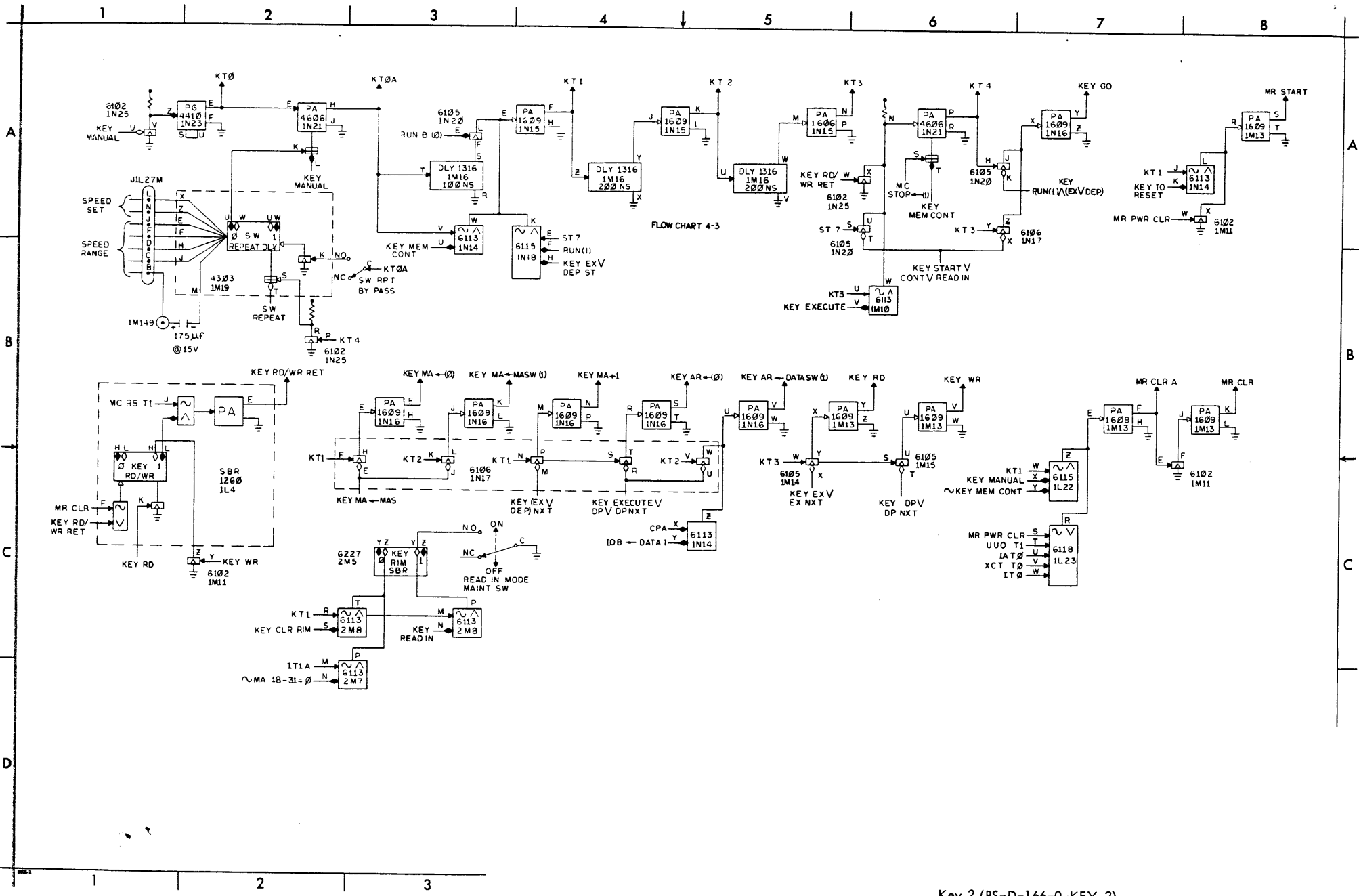
PI ACTIVE ← (0)
 PI O ← (0)
 PI R ← (0)
 PI I ← (0)
 PI H ← (0)



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



Key 1 (BS-D-166-0-KEY-1)



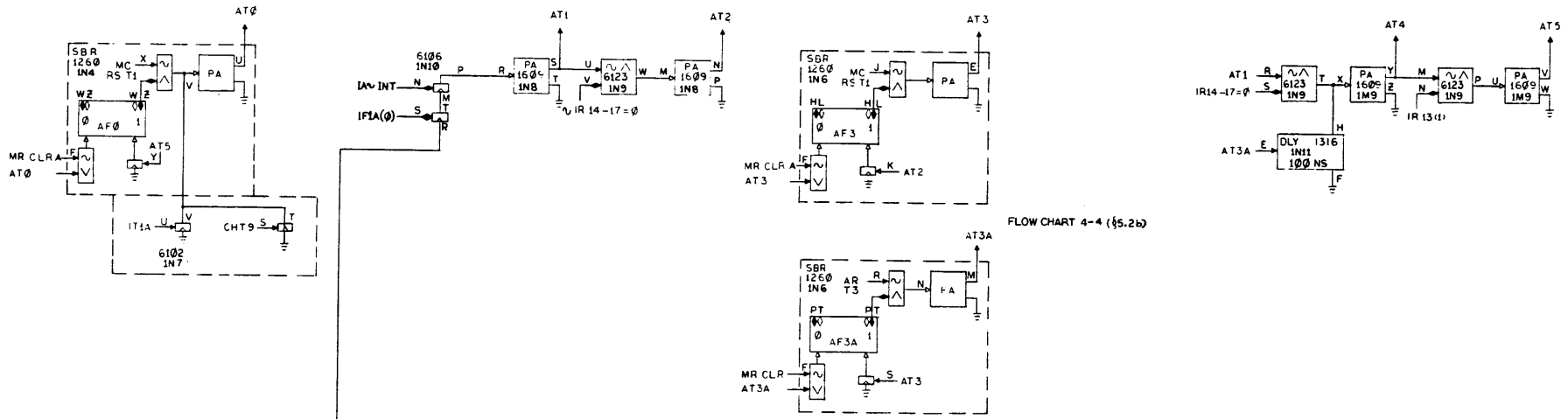
Key 2 (BS-D-166-0-KEY-2)

1 2 3 4 5 6 7 8

A

A

A - ADDRESS CYCLE



FLOW CHART 4-4 (§5.2b)

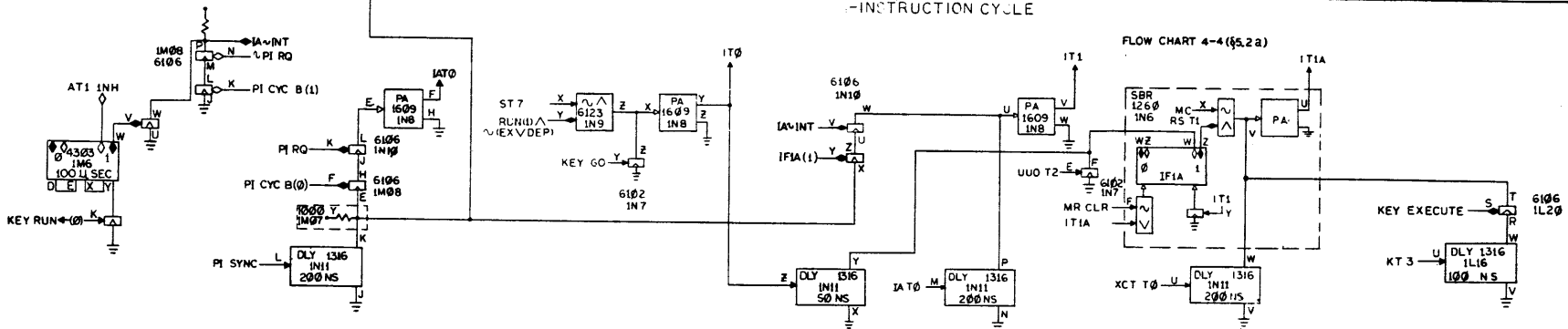
B

B

C

C

I - INSTRUCTION CYCLE



FLOW CHART 4-4 (§5.2a)

D

D

A

B

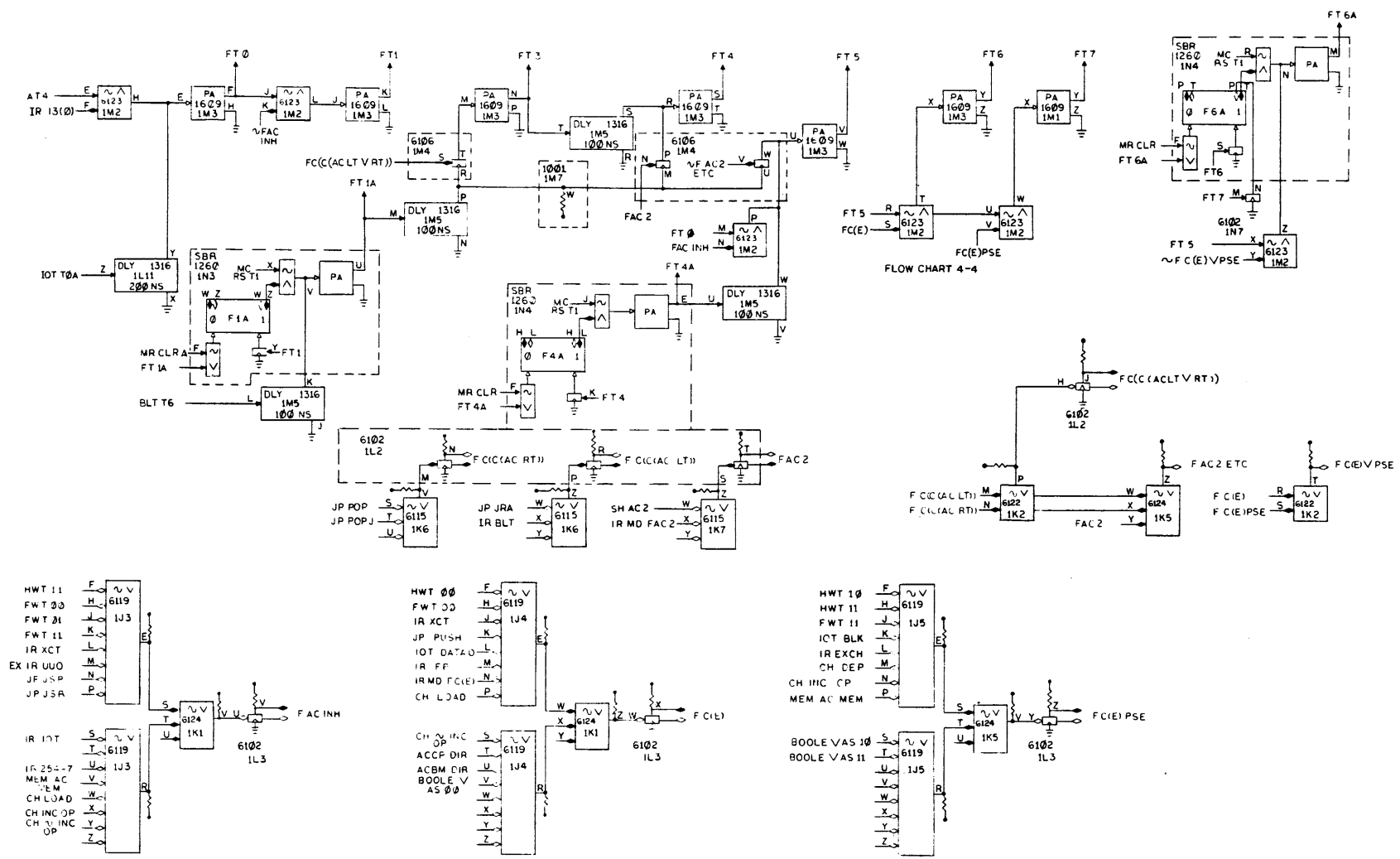
C

D

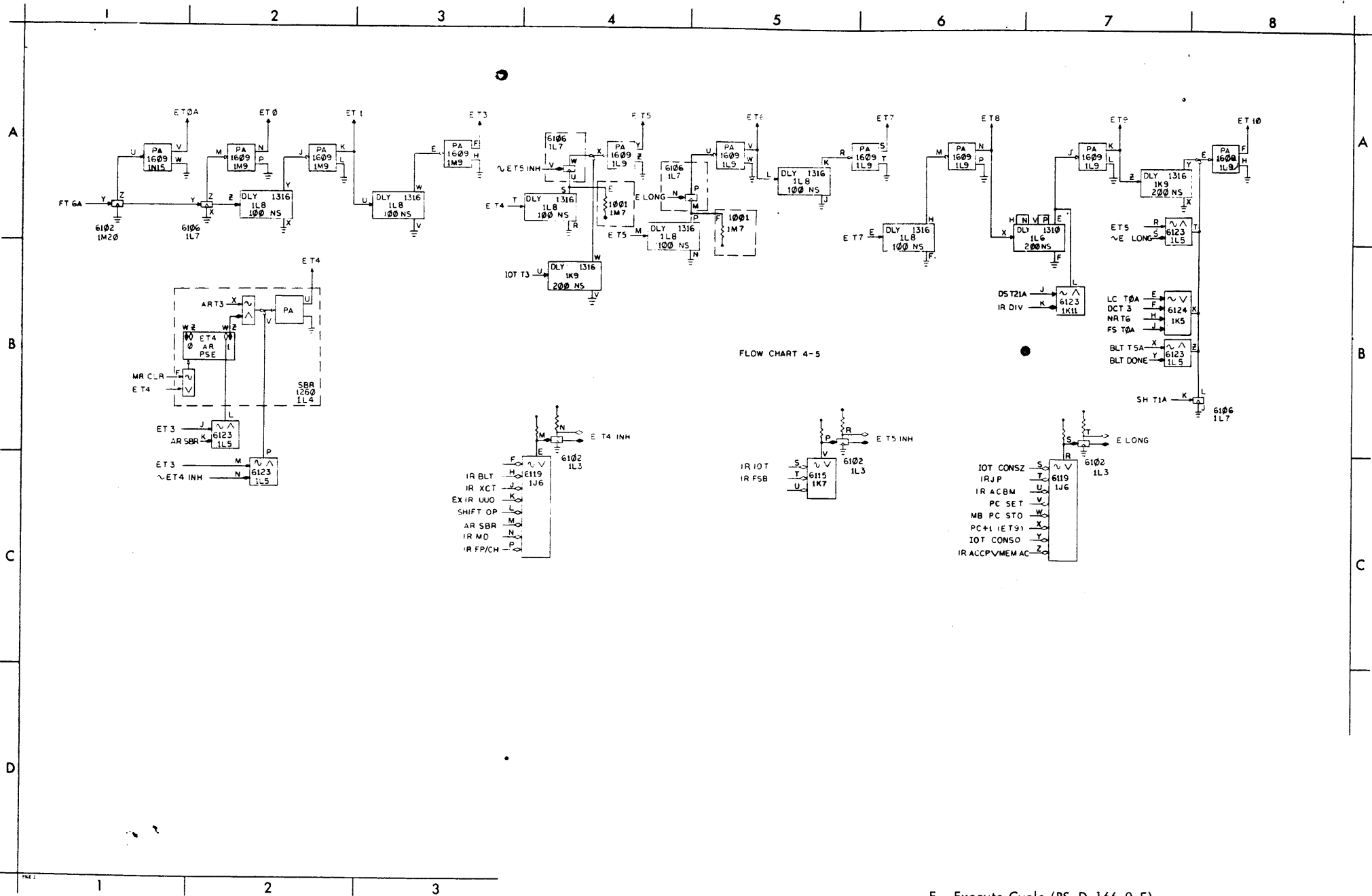
A

B

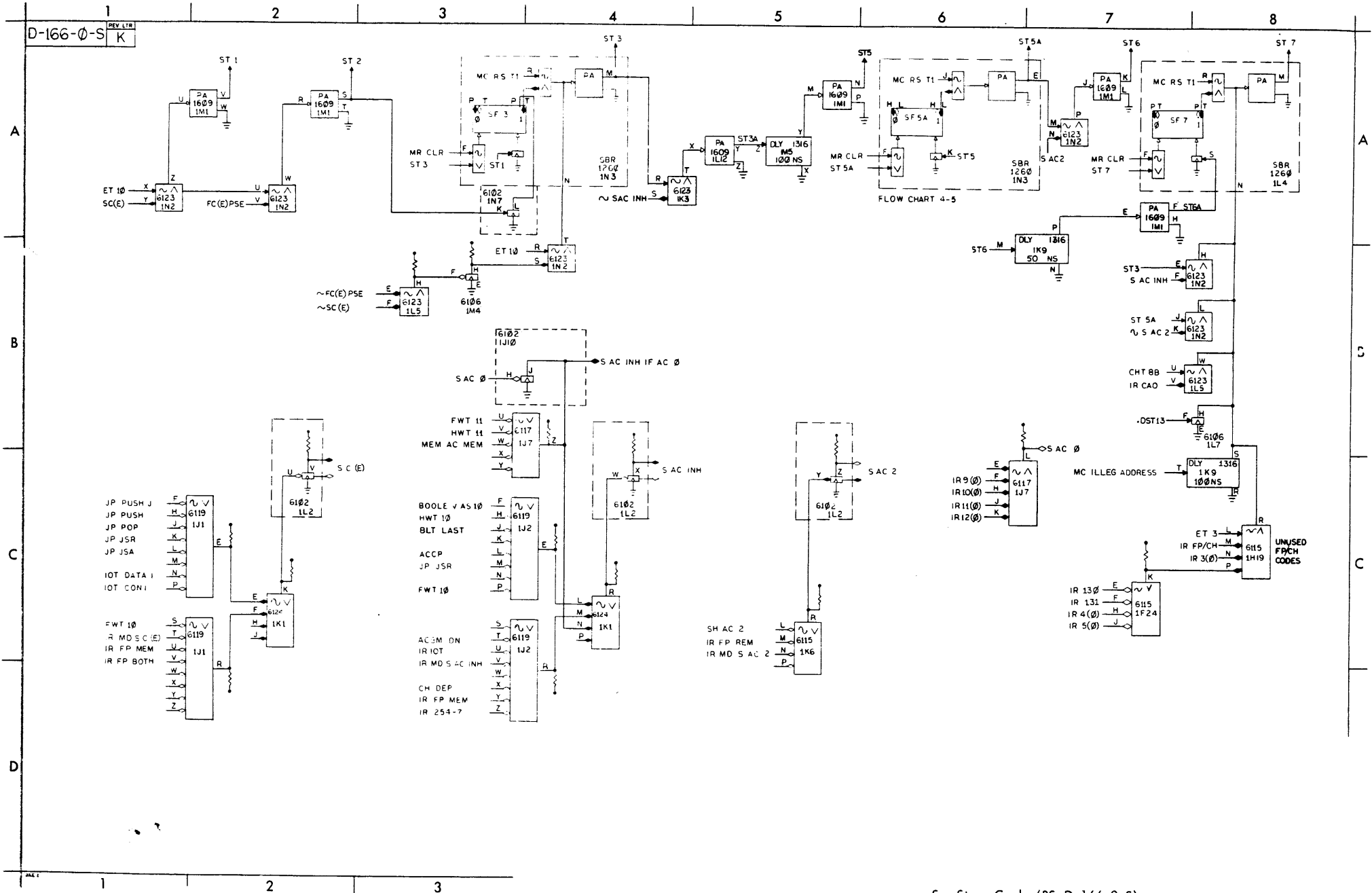
C



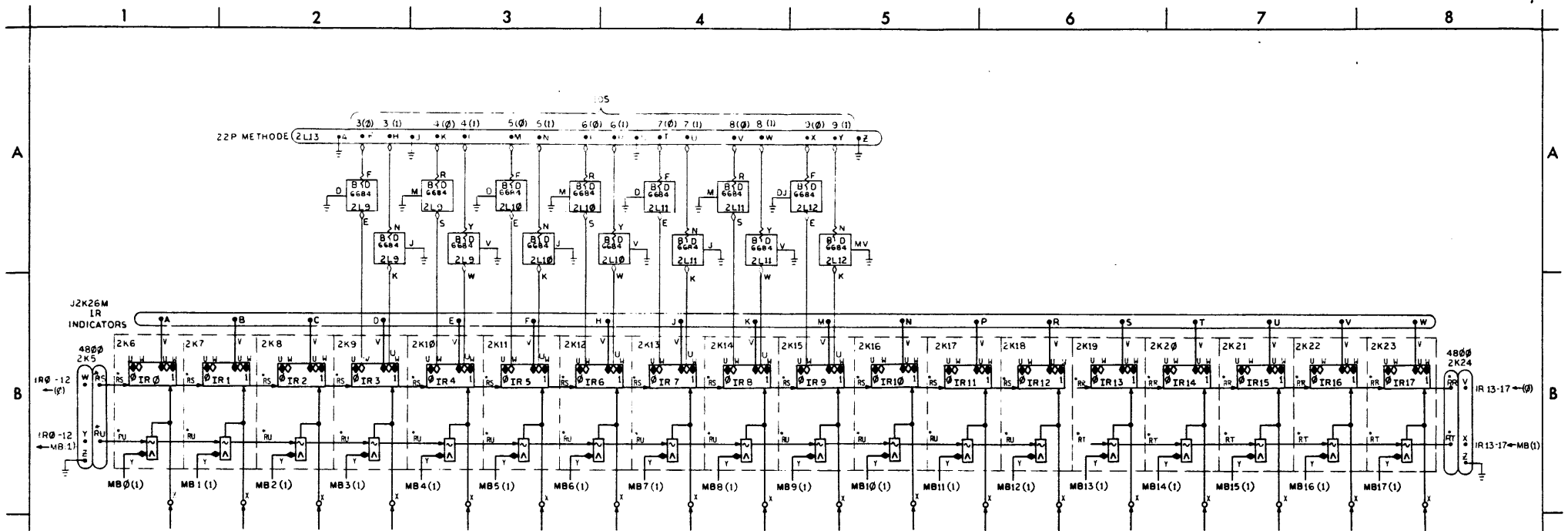
F - Fetch Cycle (BS-D-166-0-F)



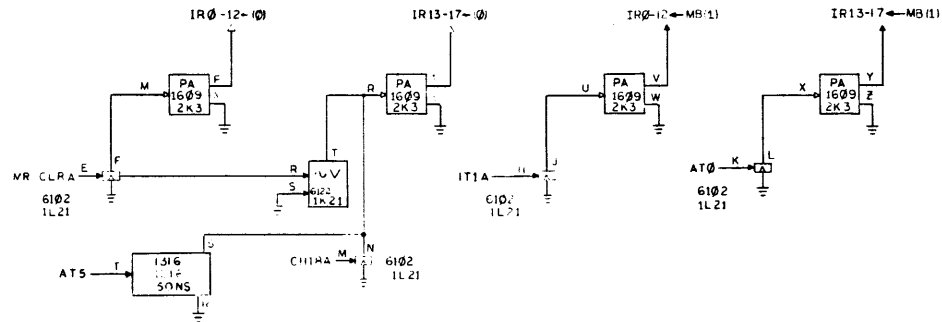
E - Execute Cycle (BS-D-166-0-E)

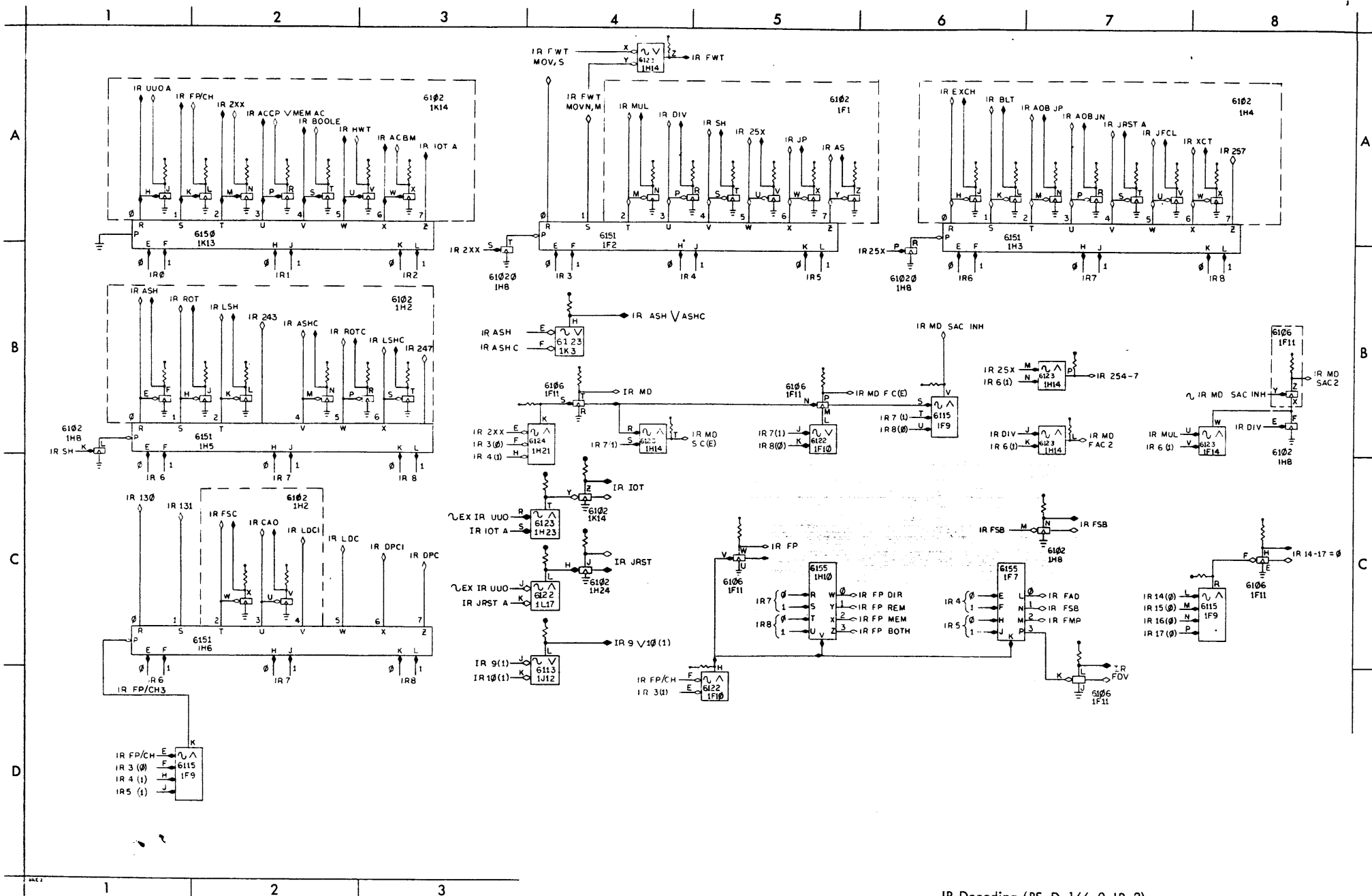


S - Store Cycle (BS-D-166-0-S)

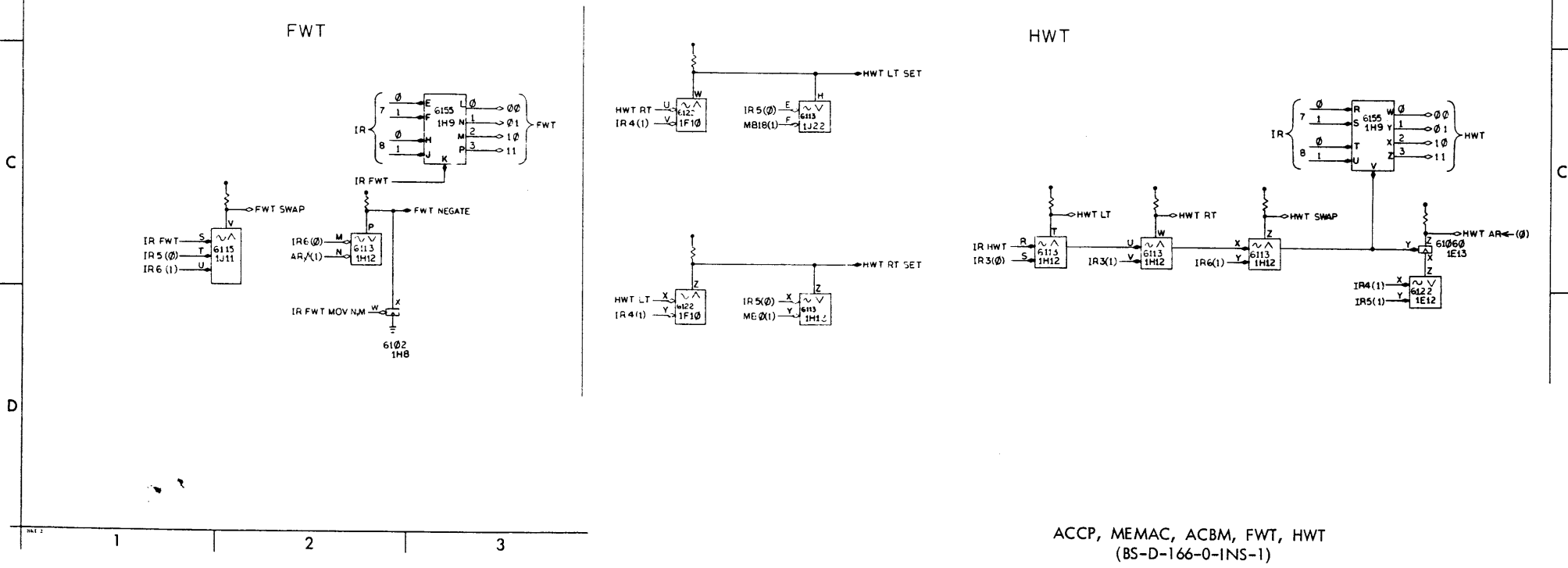
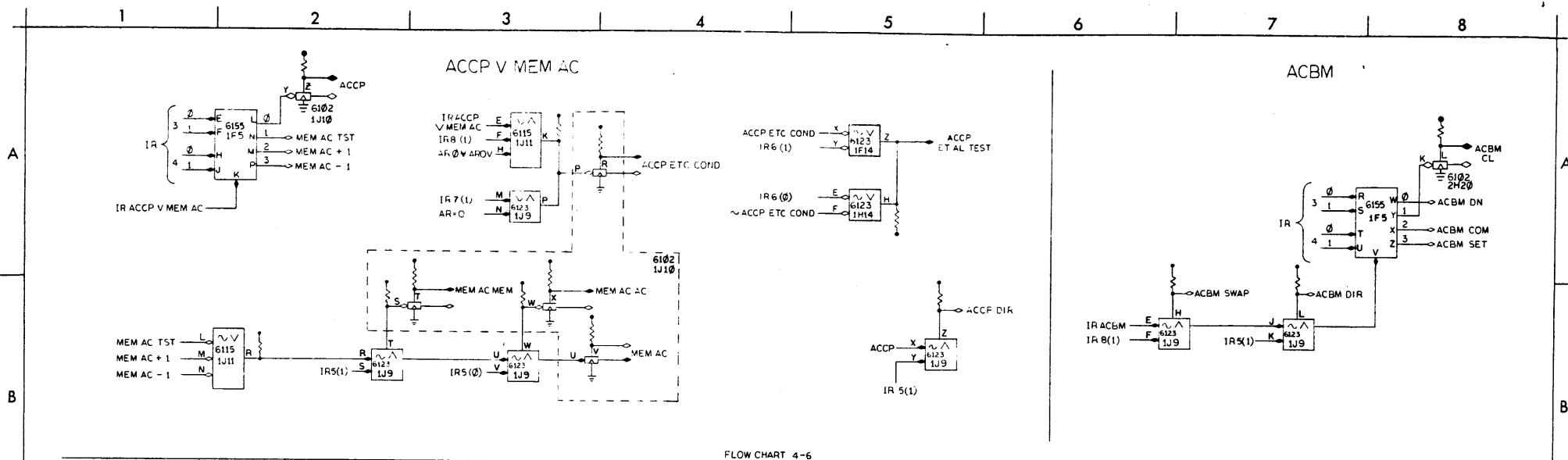


- NOTES:
1. ALL FF PACKAGES ARE 6206
 2. * INDICATES REAR CONNECTOR PIN.
 3. GROUND PIN D AND Z IN ALL 6206'S





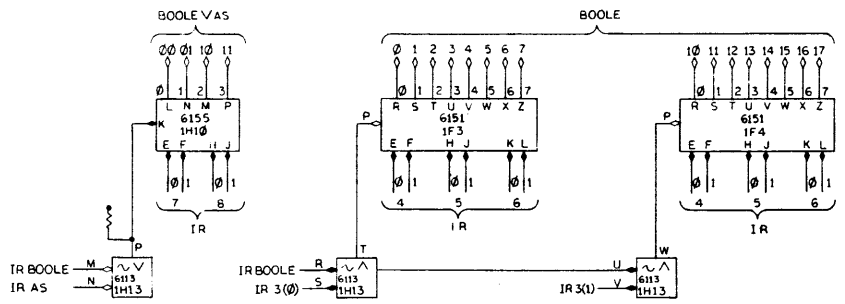
IR Decoding (BS-D-166-0-IR-2)



A

		RESULT BIT FOR OPERAND BIT CONFIGURATION			ET0	ET1	ET4
		T1	T2	T3			
SETZ	MEM ^ AC	0	0	0			
AND	MEM ^ AC	0	0	0			
ANDCA	MEM ^ AC	0	0	1			
SETM	MEM	0	0	1			
ANDCM	MEM ^ AC	0	1	0			
SETA	AC	0	1	0			
XOR	MEM ^ AC	0	1	1			
JOR	MEM ^ AC	0	1	1			
ANDCB	MEM ^ AC	1	0	0			
EDV	MEM ^ AC	1	0	0			
SETCA	AC	1	0	1			
ORCA	MEM ^ AC	1	0	1			
SETCM	MEM	1	1	0			
ORCM	MEM ^ AC	1	1	0			
ORCB	MEM ^ AC	1	1	1			
SETO	MEM ^ AC	1	1	1			

BOOLE



FLOW CHART 4-8

B

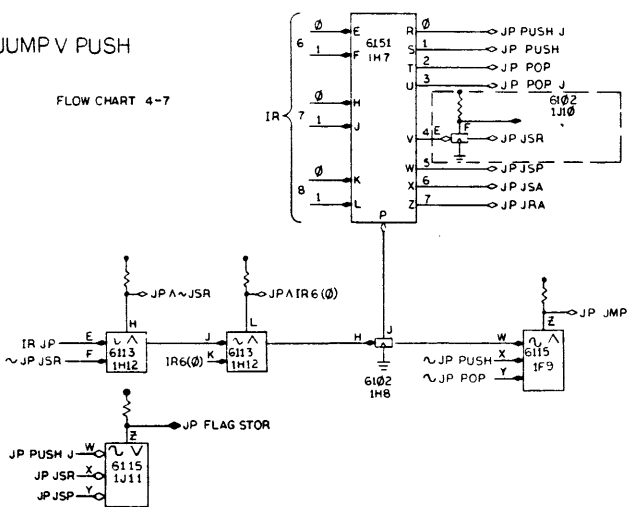
B

C

C

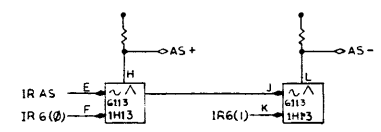
JUMP V PUSH

FLOW CHART 4-7



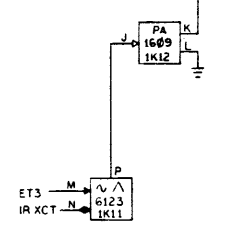
AS

FLOW CHART 4-8



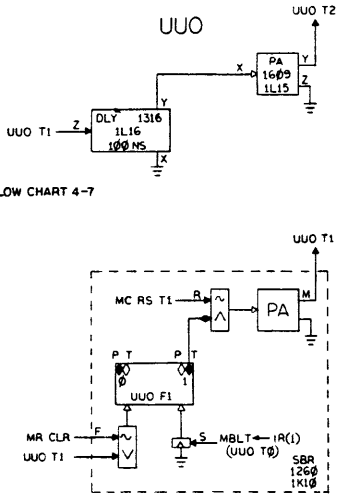
XCT

FLOW CHART 4-7



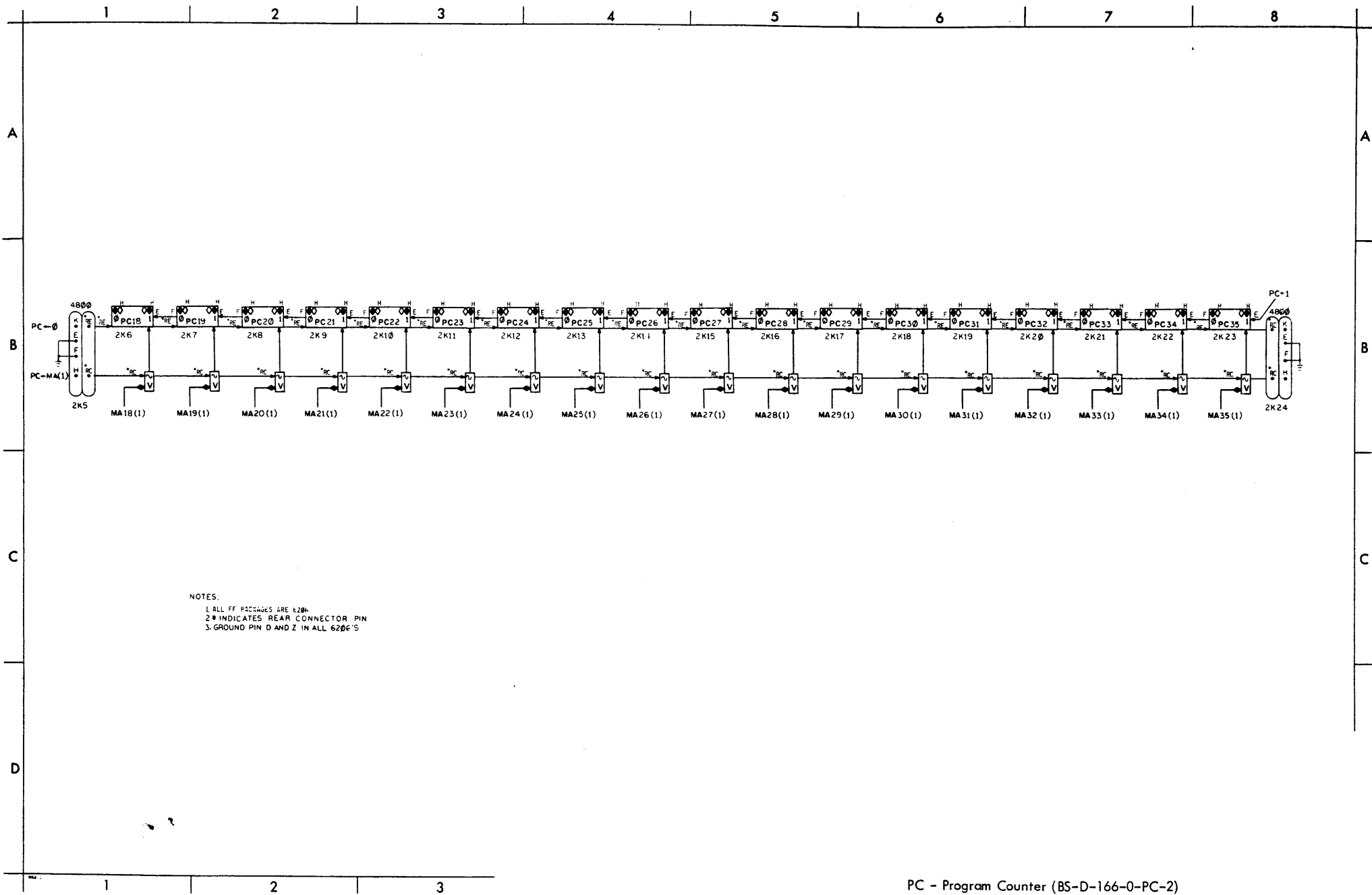
UWO

FLOW CHART 4-7



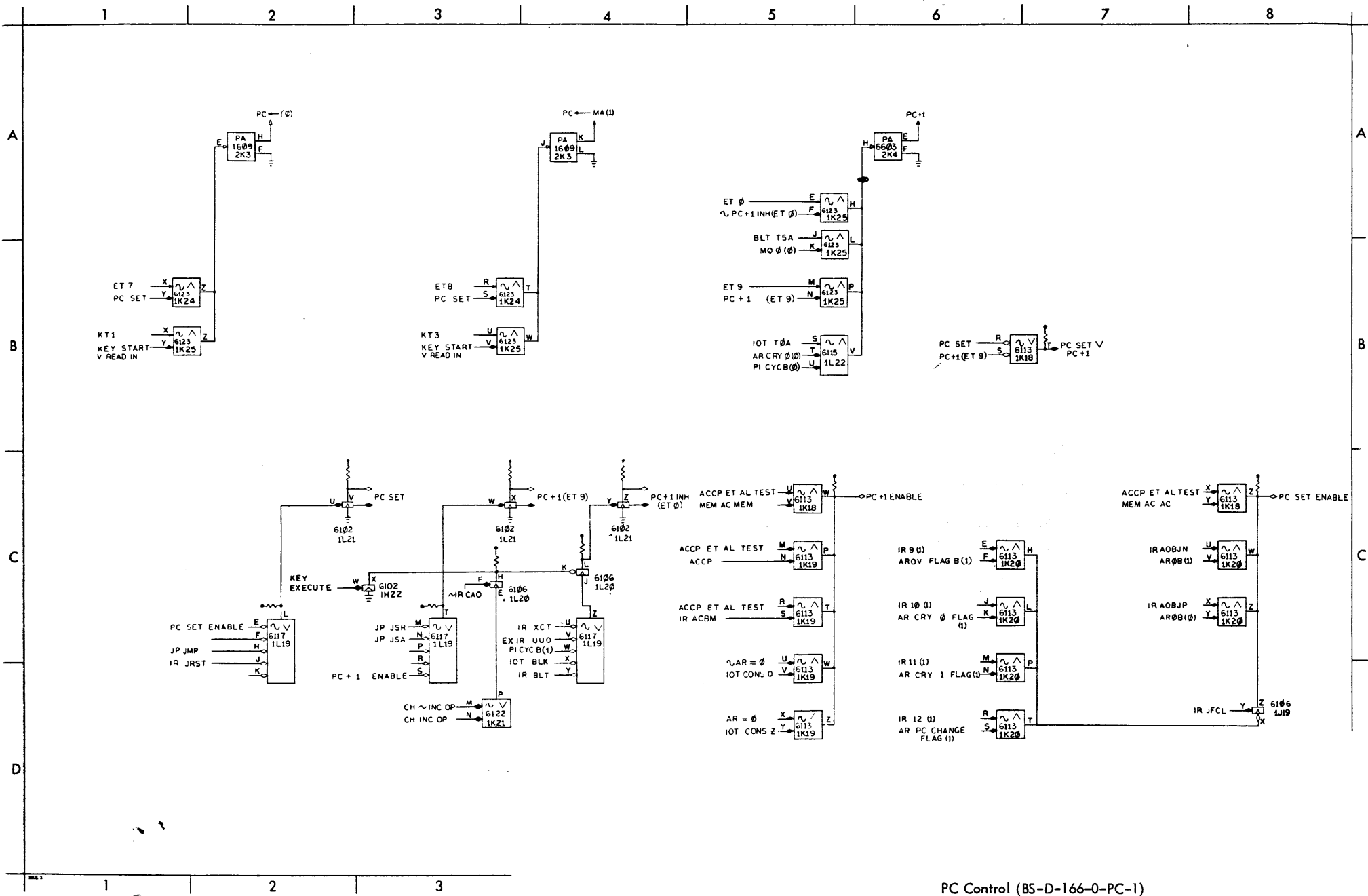
D

D



NOTES:

1. ALL FF PACKAGES ARE 620A.
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 620A'S



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

A

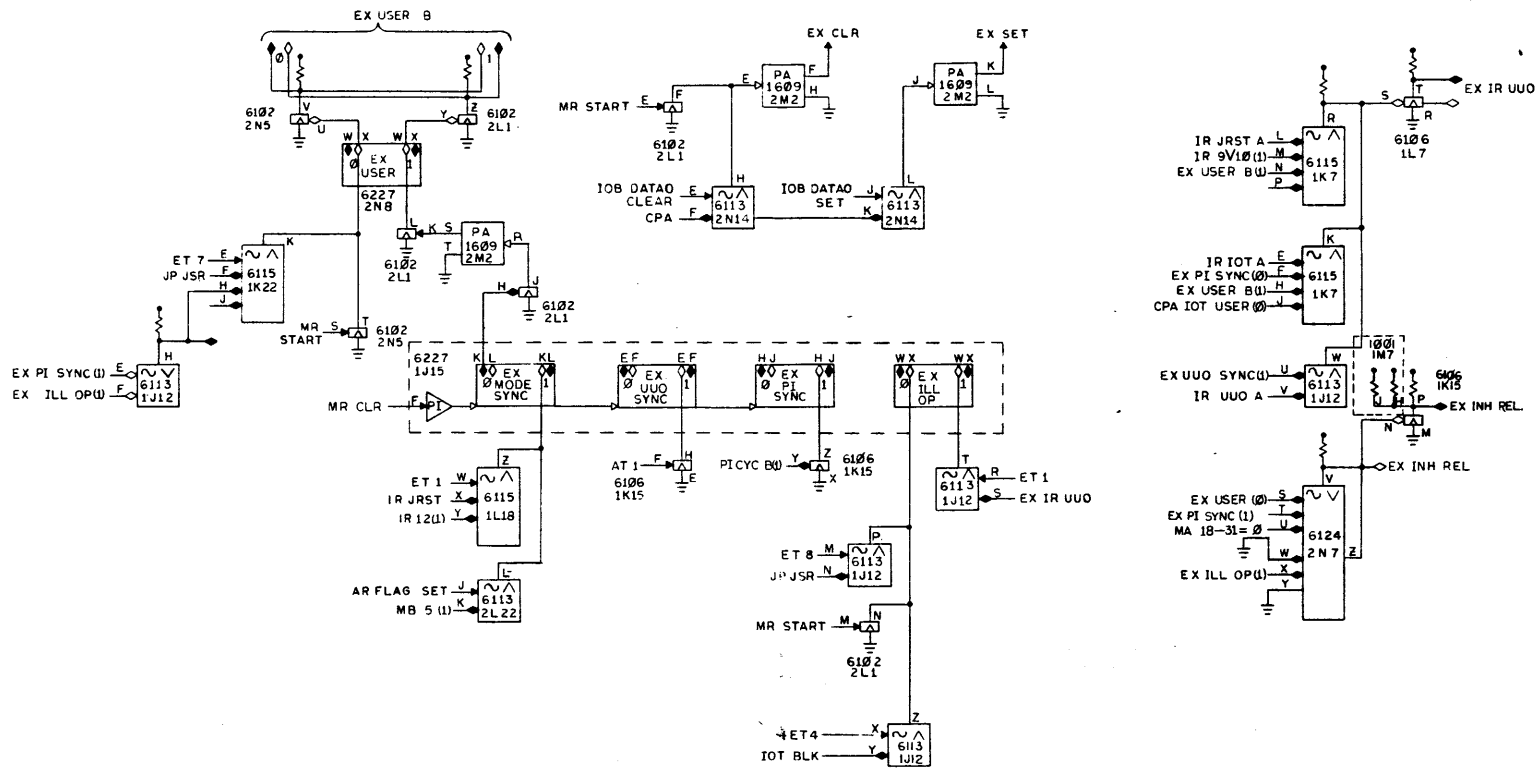
B

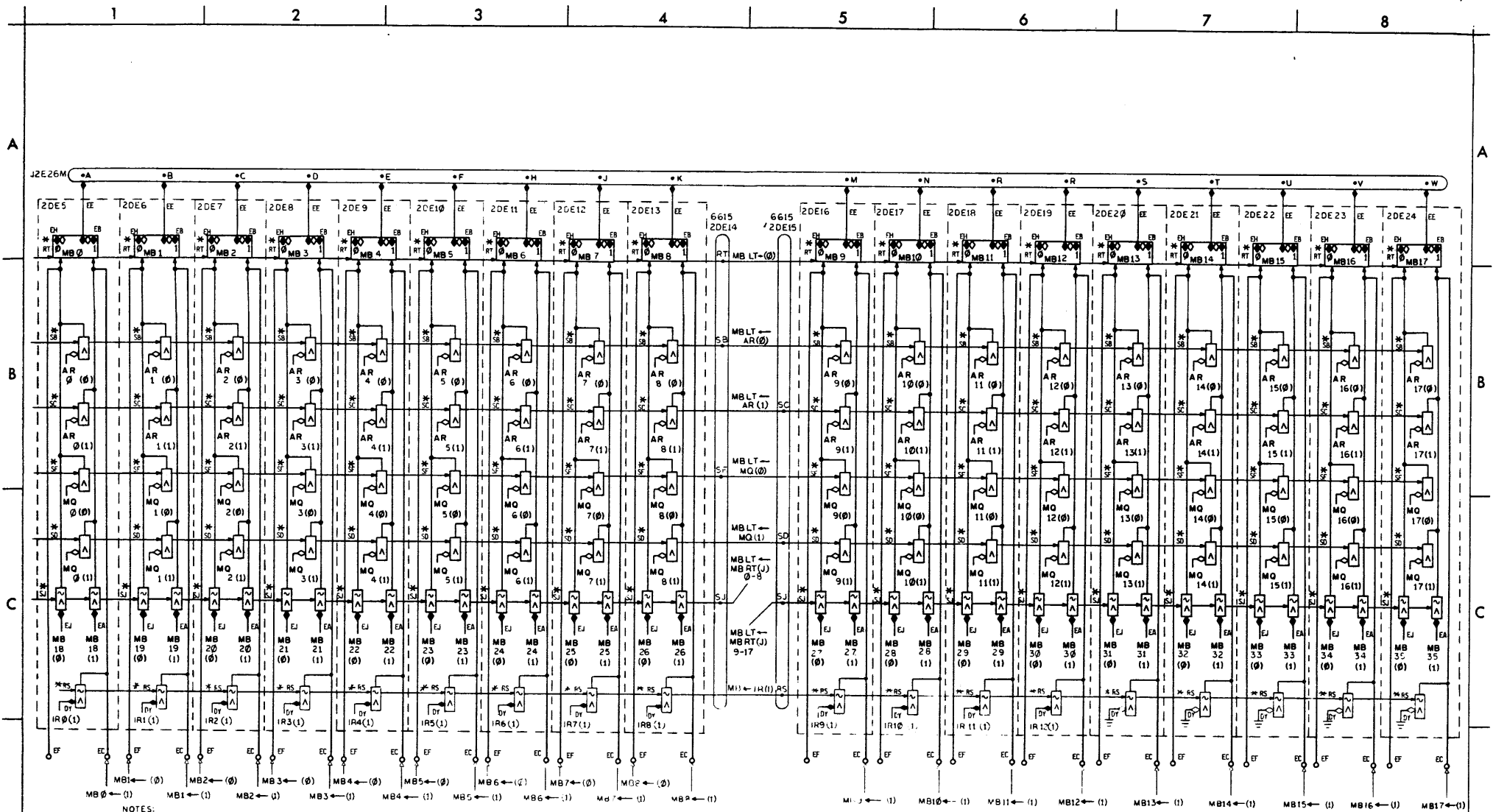
C

D

B

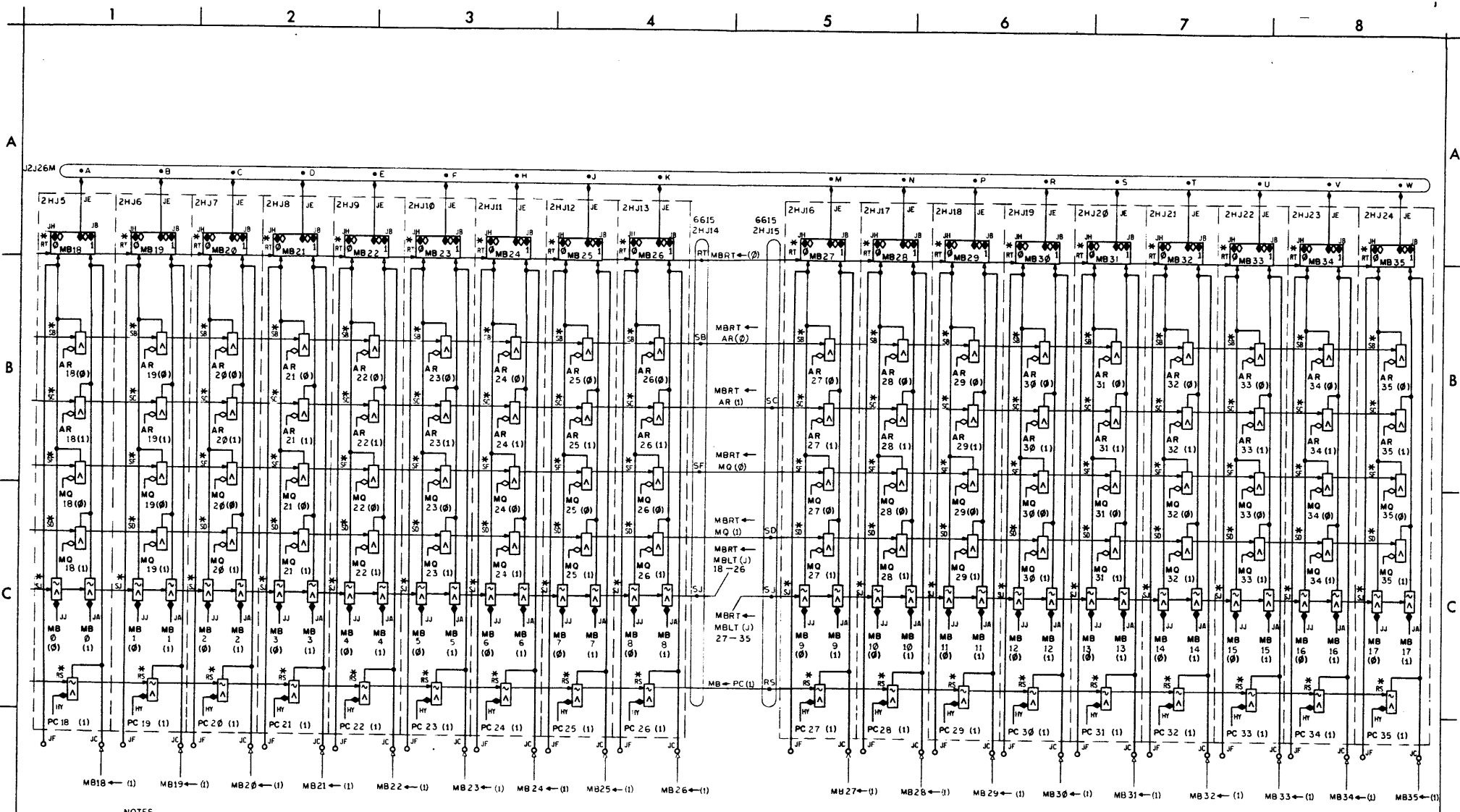
C



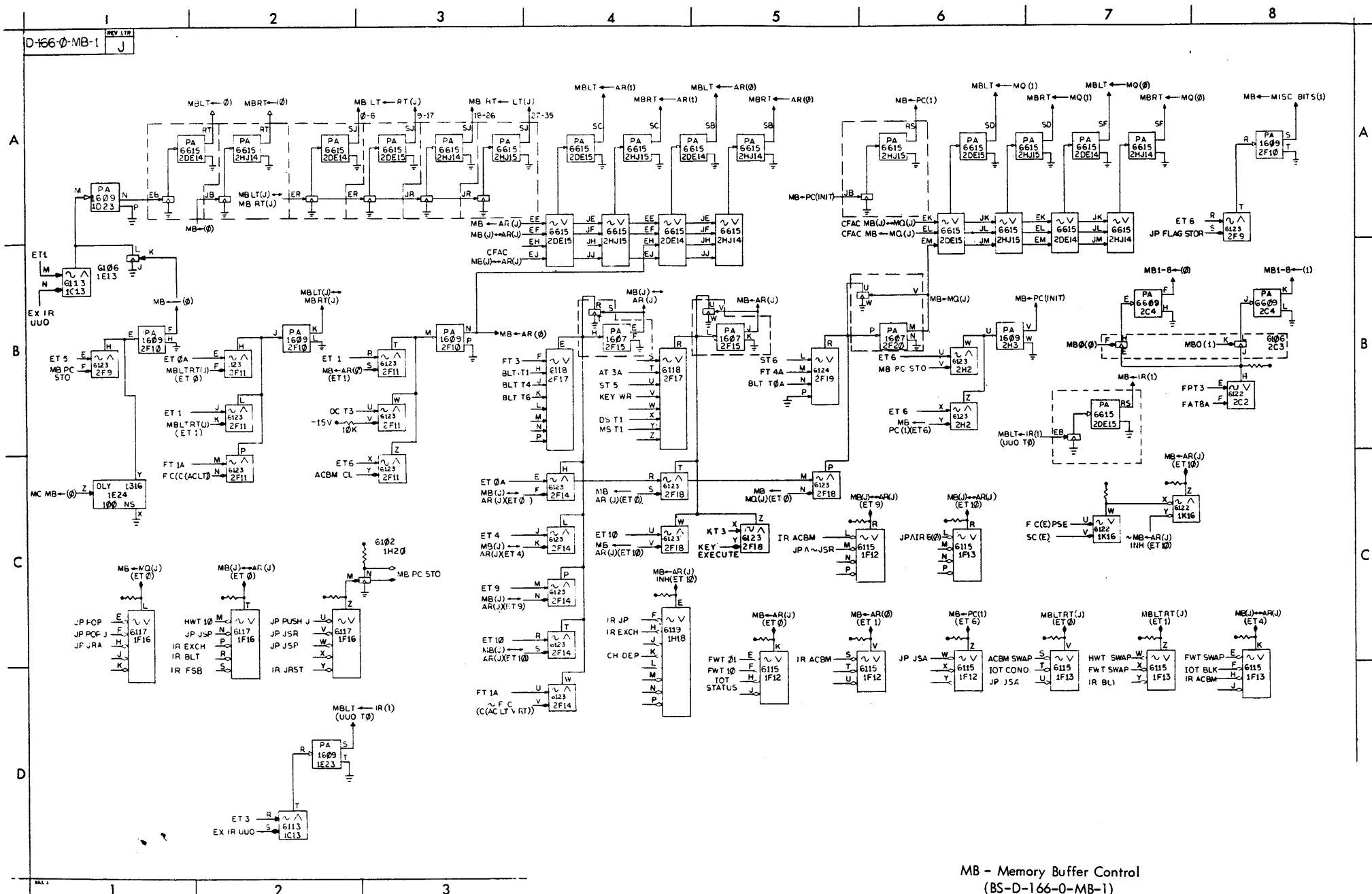


NOTES:
 1. ALL FF PACKAGES ARE 6205
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN D,P AND Z IN ALL 6205'S AND 6615..

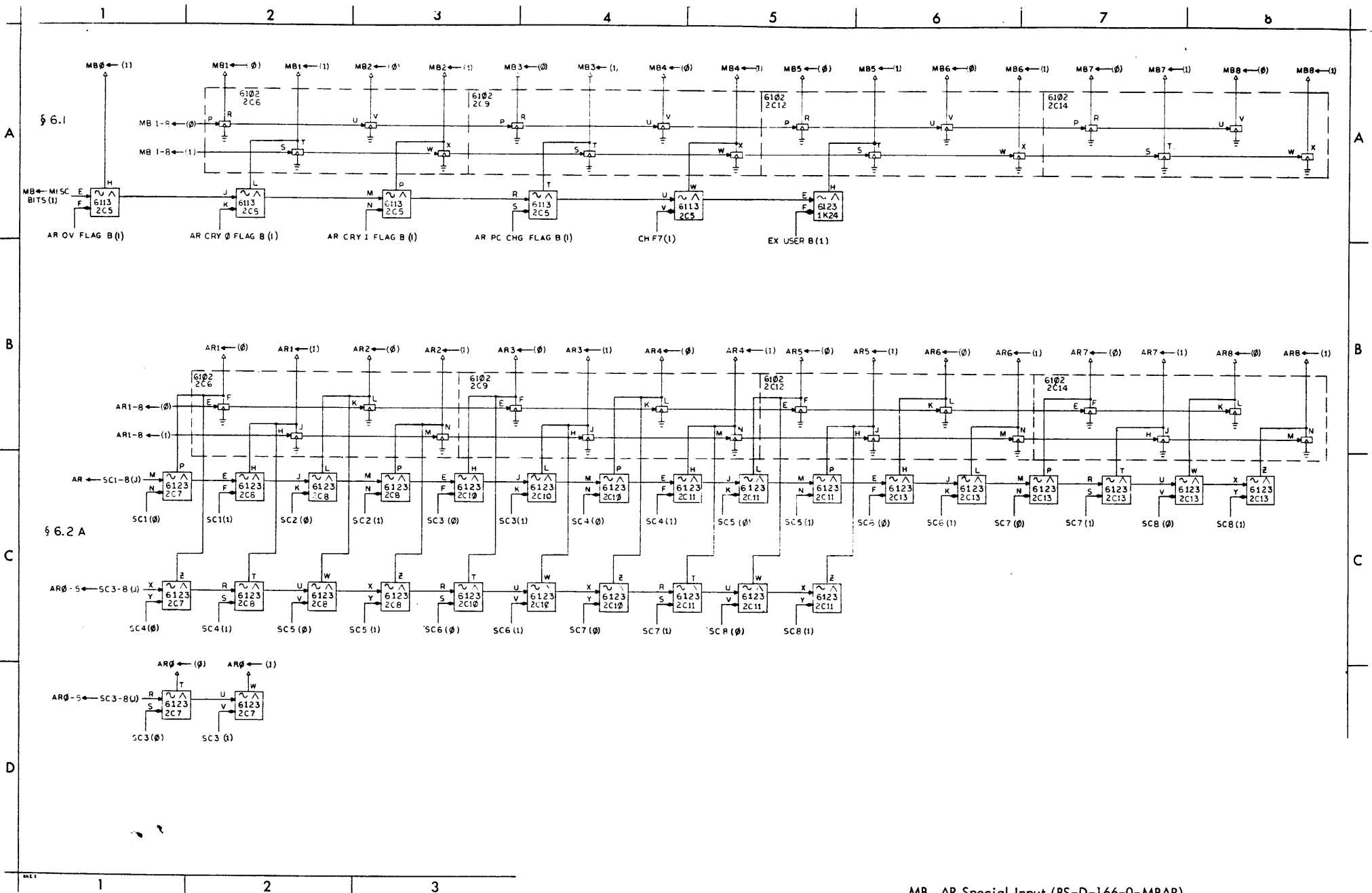
MBLT Register 0-17 (BS-D-166-0-MB-2)

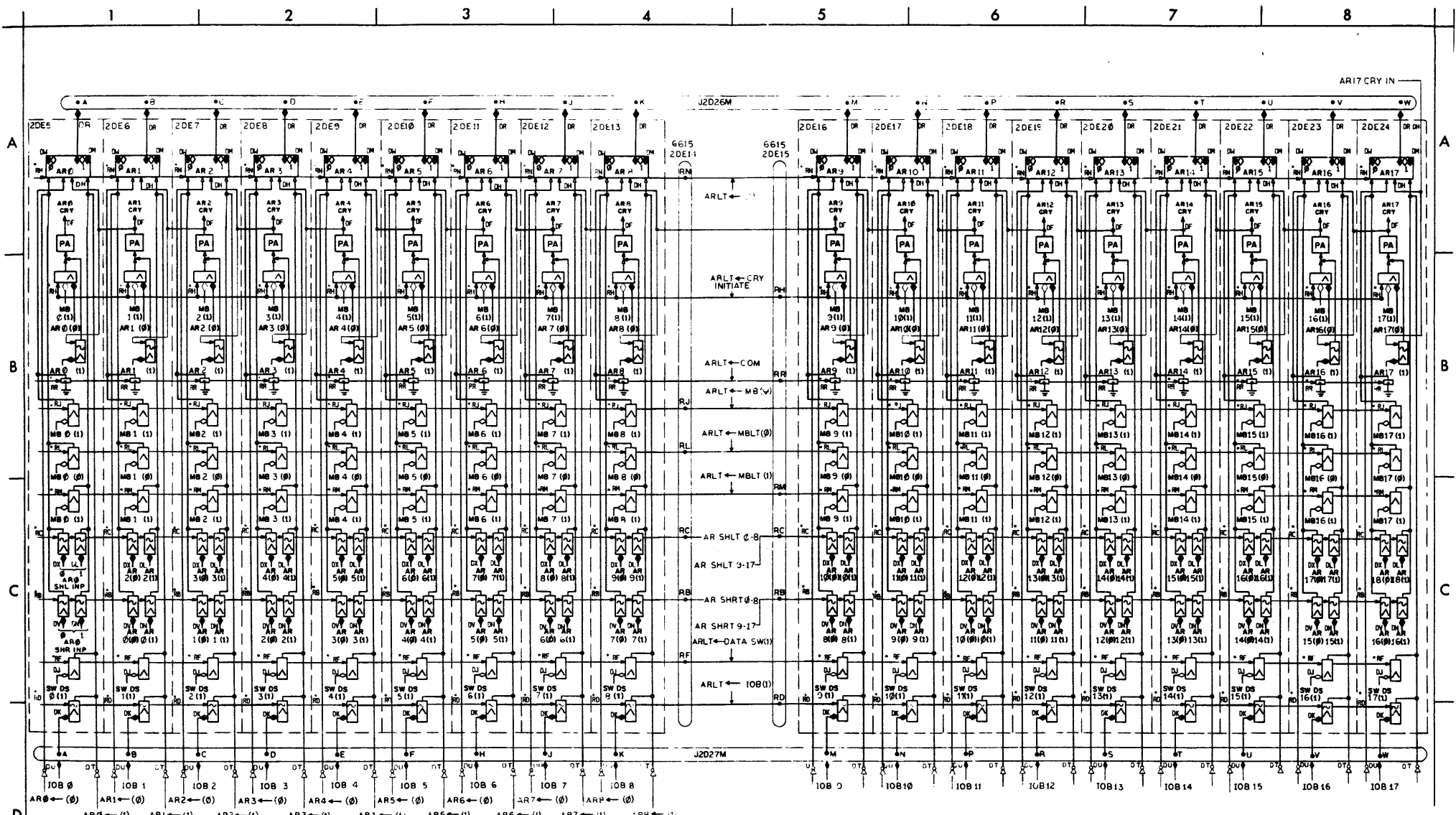


- NOTES
1. ALL FF PACKAGES ARE 6205
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN D,P AND Z IN ALL 6205 AND 66155.

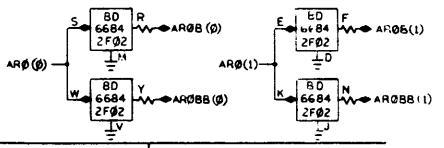


MB - Memory Buffer Control
(BS-D-166-0-MB-1)



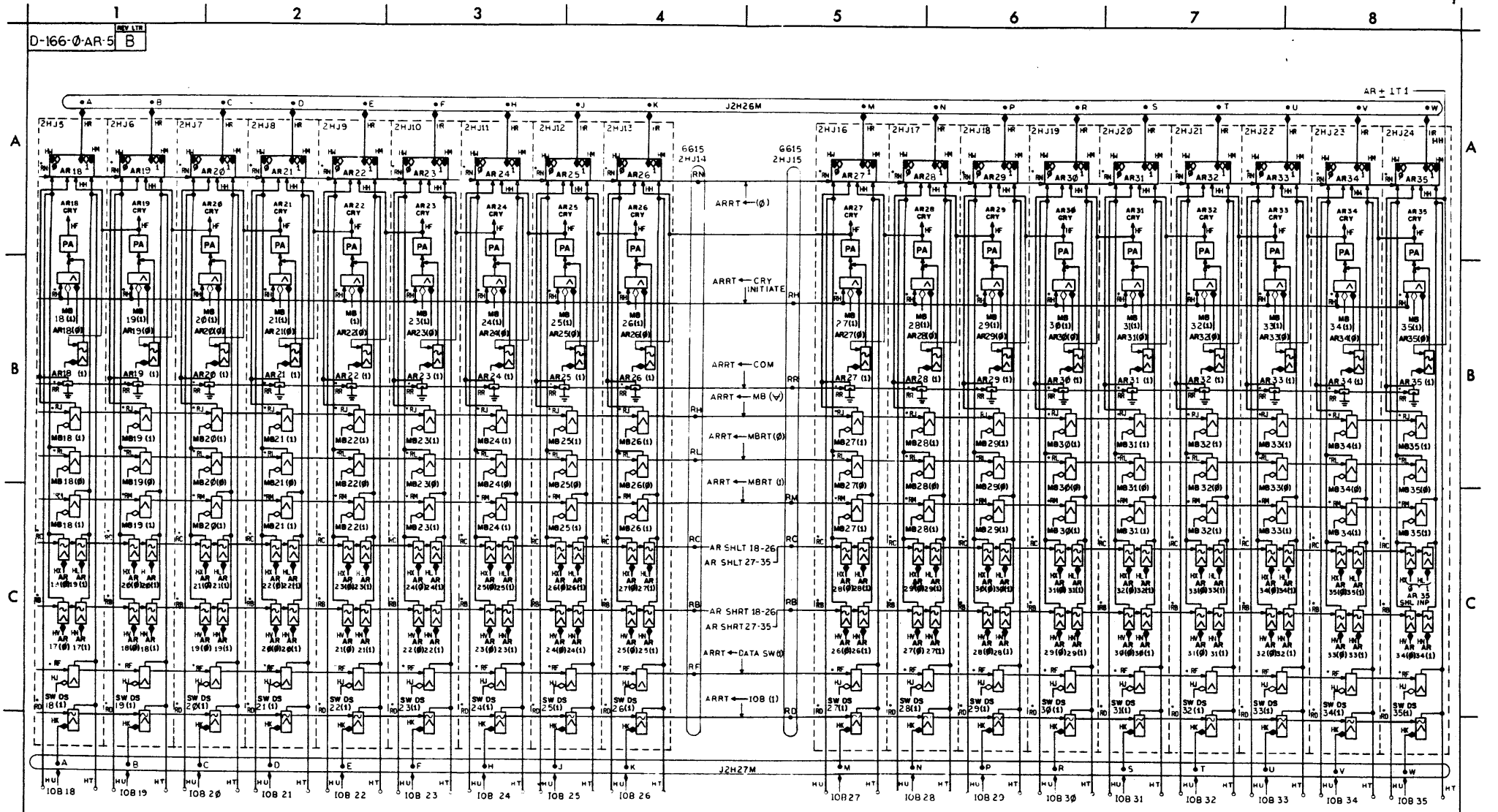


- NOTE
1. * INDICATES FEED DIRECTION PIN
 2. ALL FF PACKAGES ARE 6205
 3. GROUND PINS C AND Z IN ALL 6205 AND 6615.



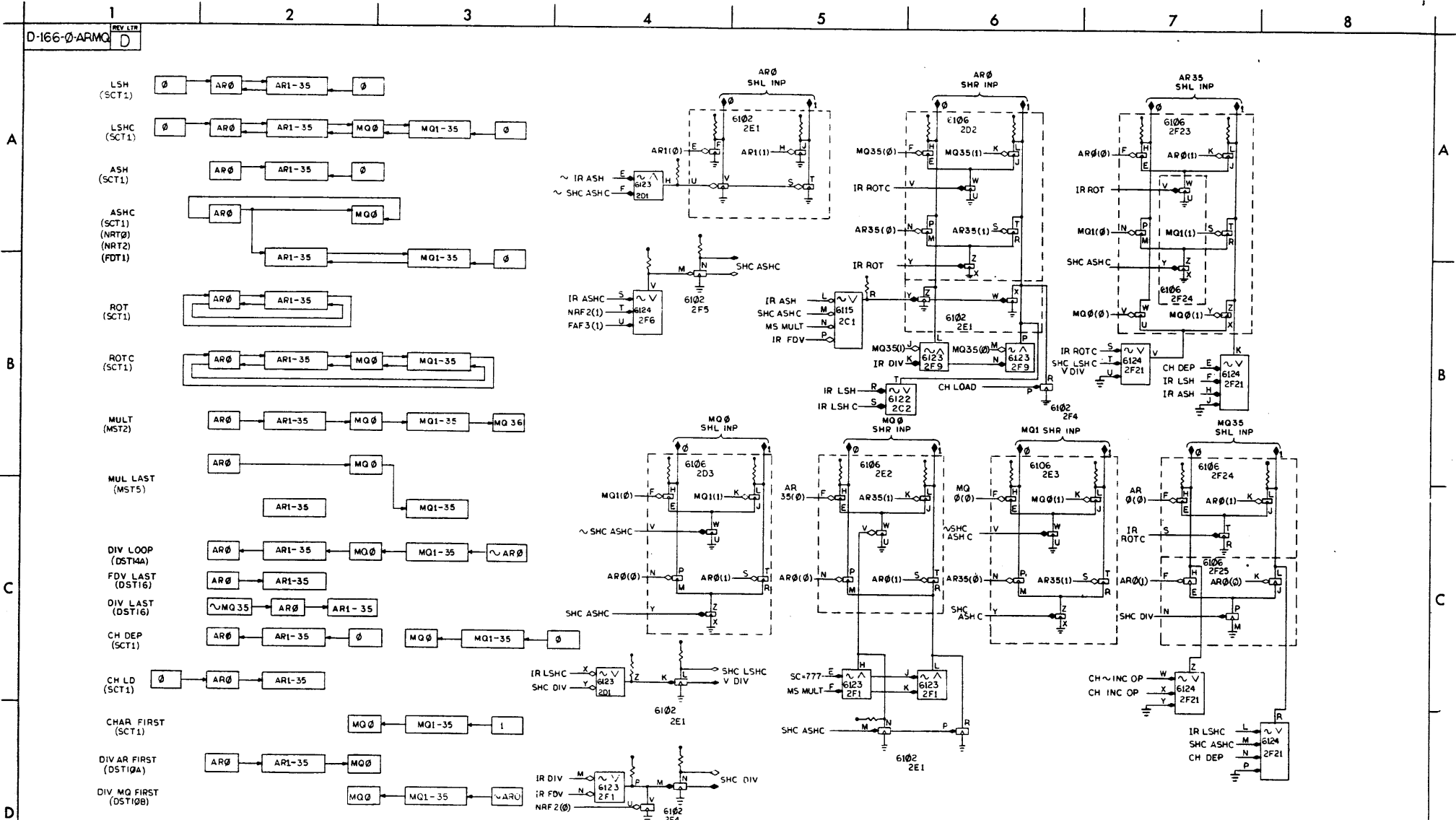
ARLT Register 0-17 (BS-D-166-0-AR-4)

D-166-0-AR-5 B

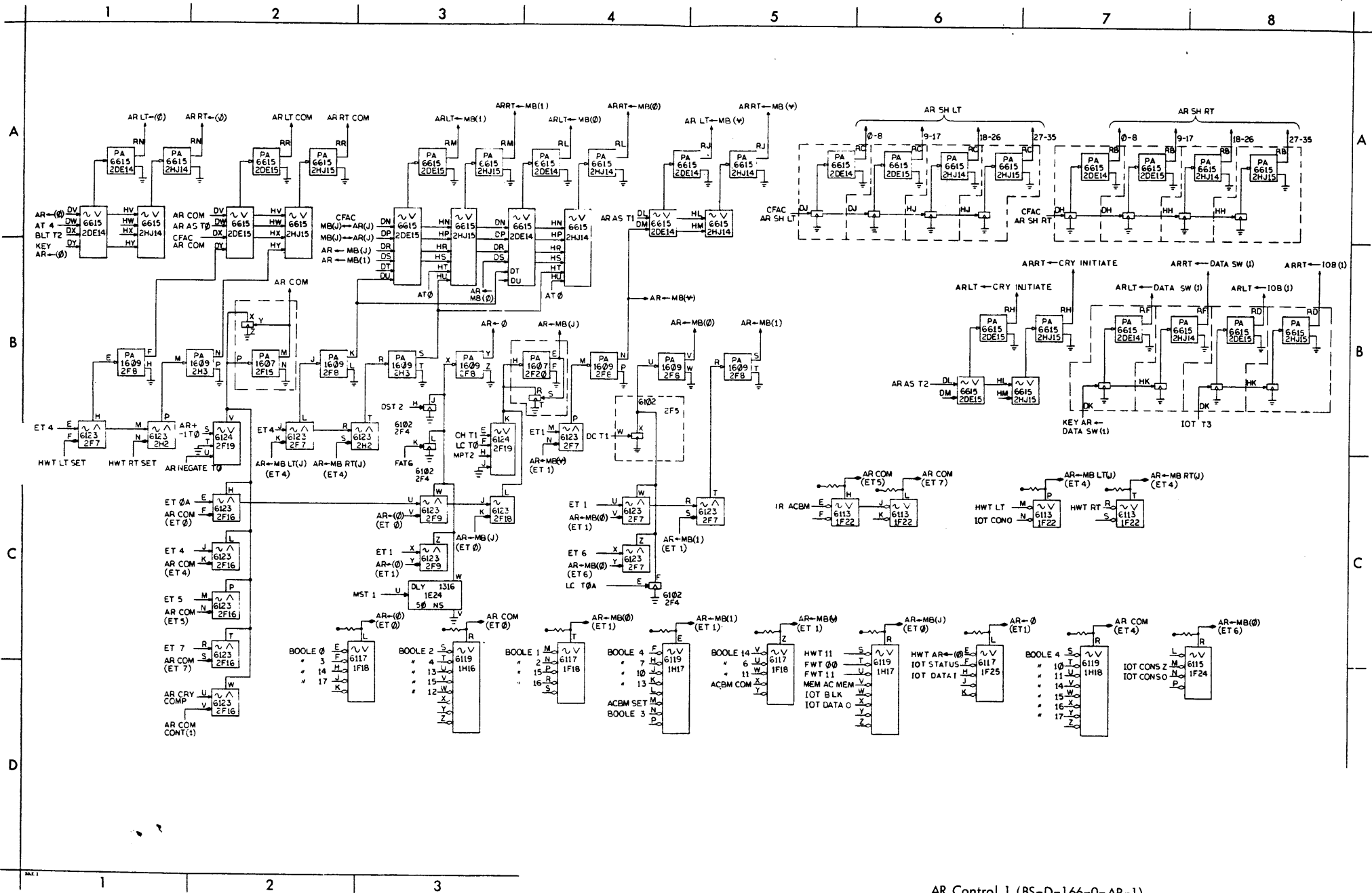


- NOTES:
1. ALL FF PACKAGES ARE 6205
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN D AND Z IN ALL 6205 AND 6615.

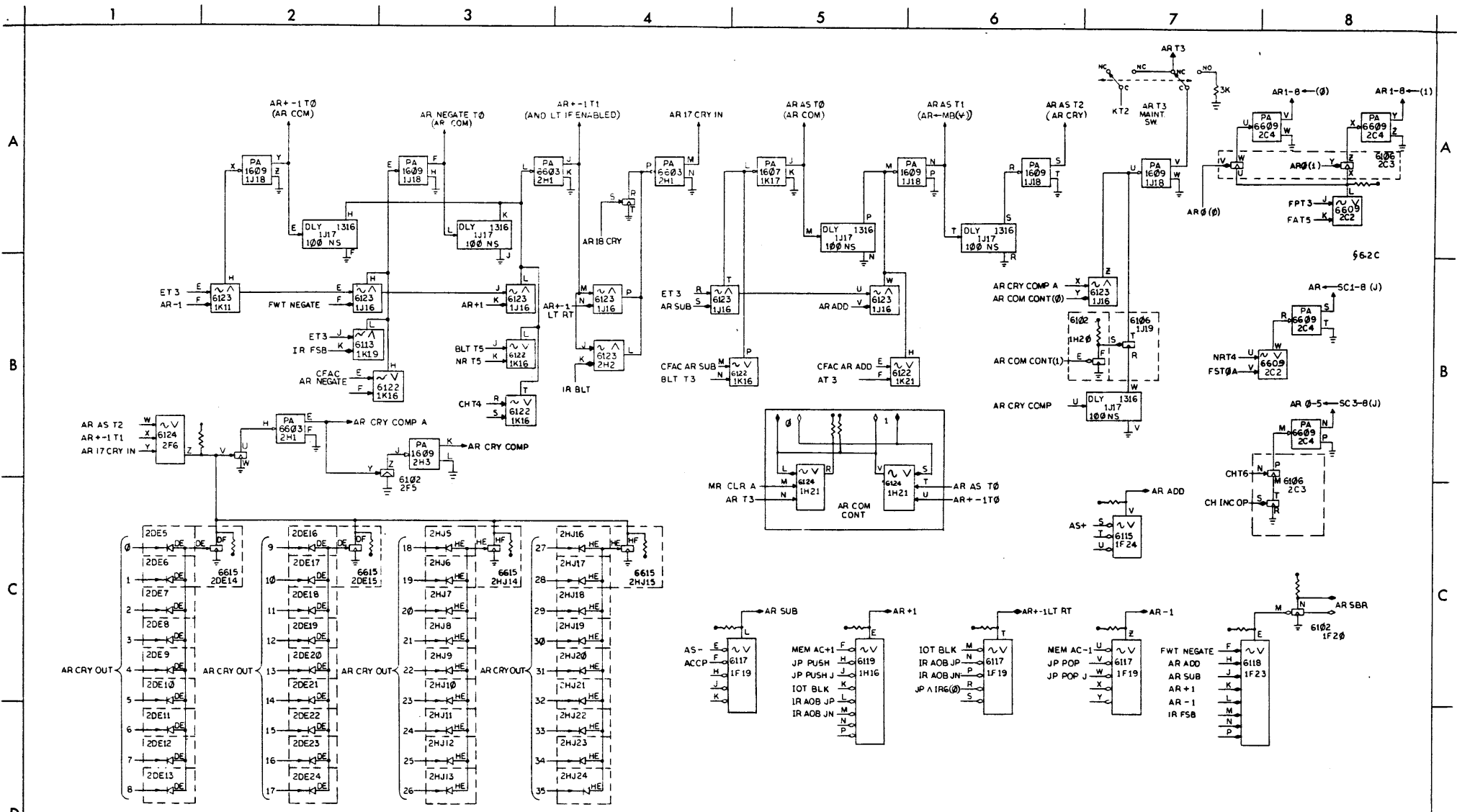
D-166-0-ARMQ



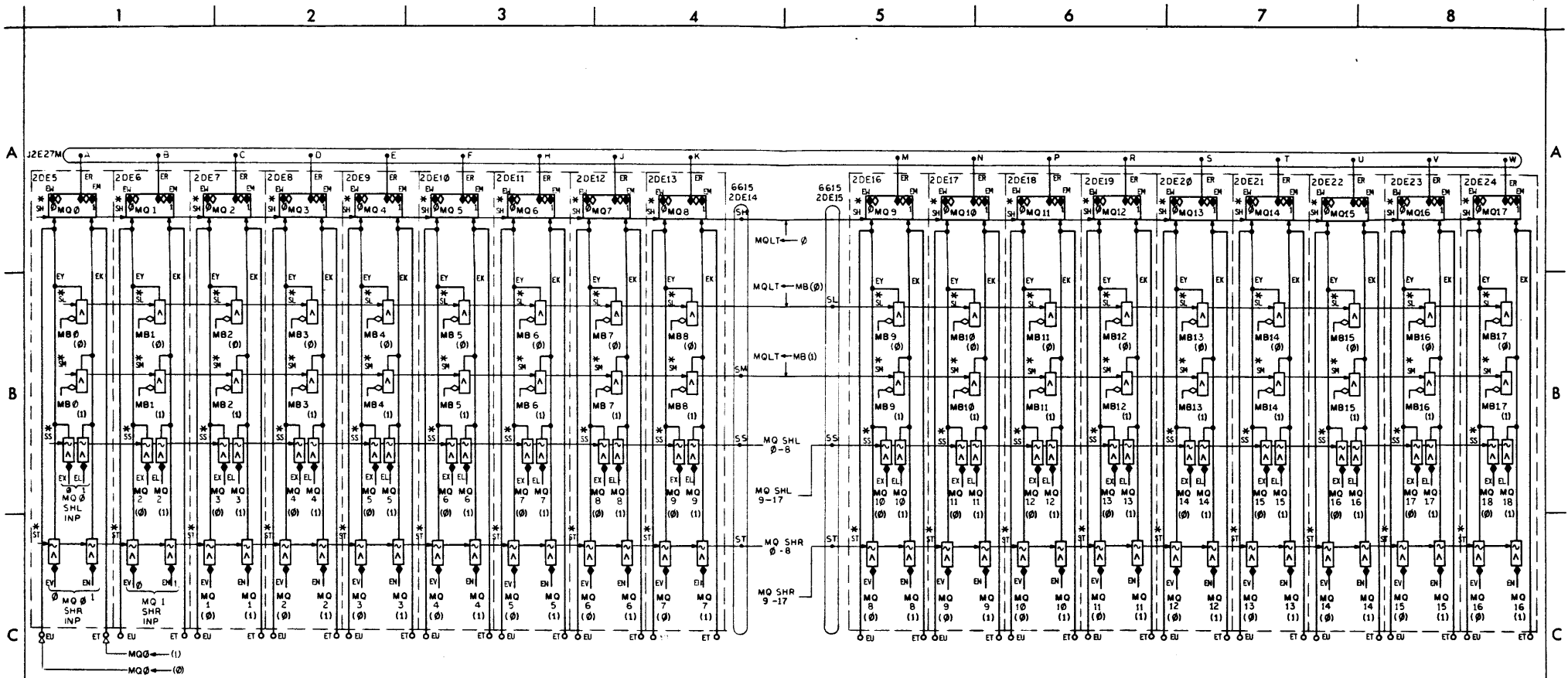
AR, MQ Shift Connections (BS-D-166-0-ARMQ)



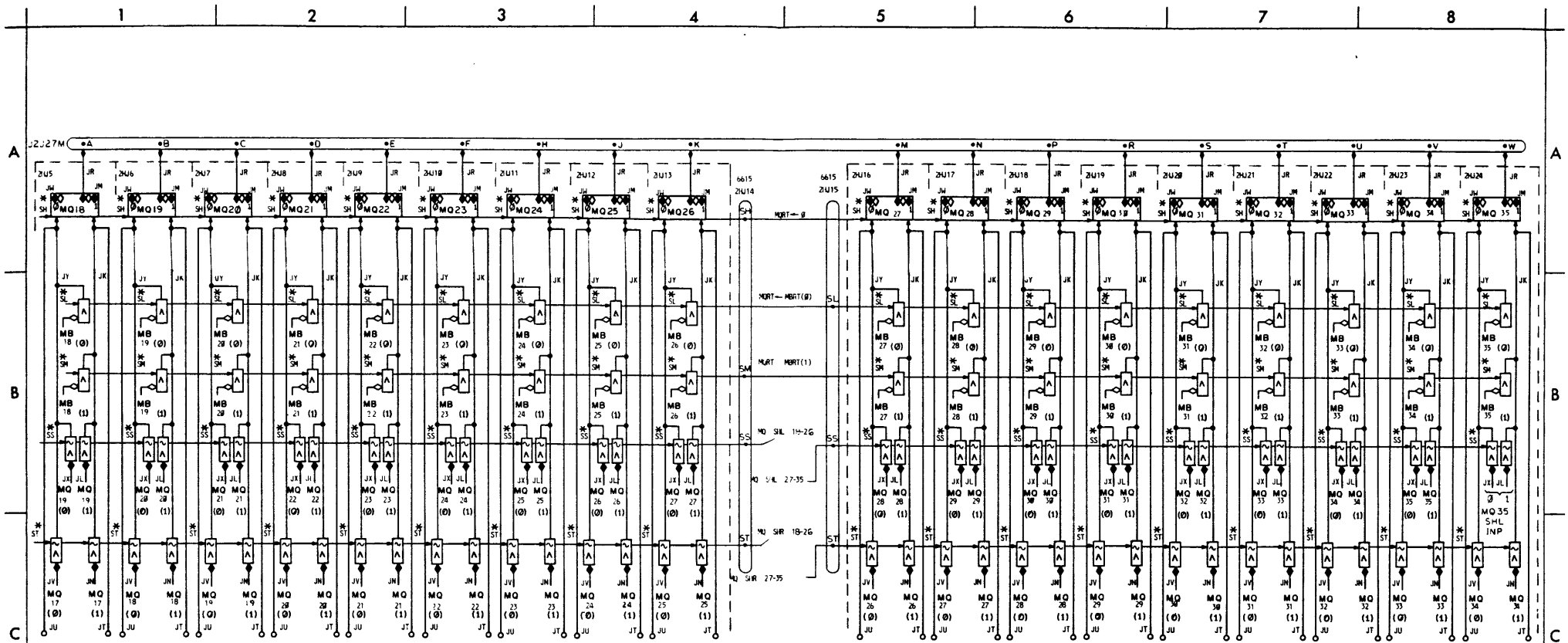
AR Control 1 (BS-D-166-0-AR-1)



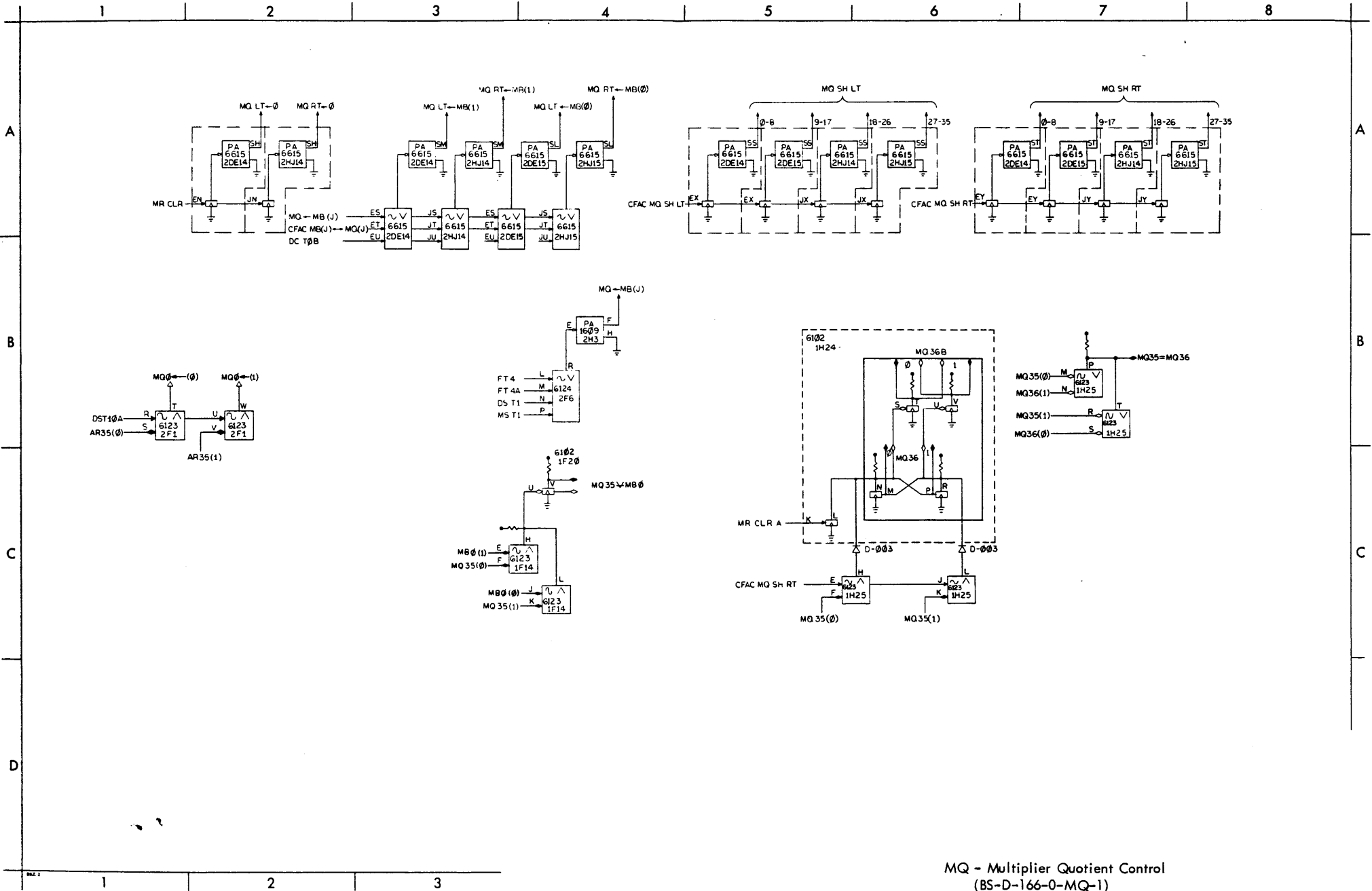
NOTE:
ALL PACKAGES WITH
DIODES SHOWN ARE 6205 IN AR.



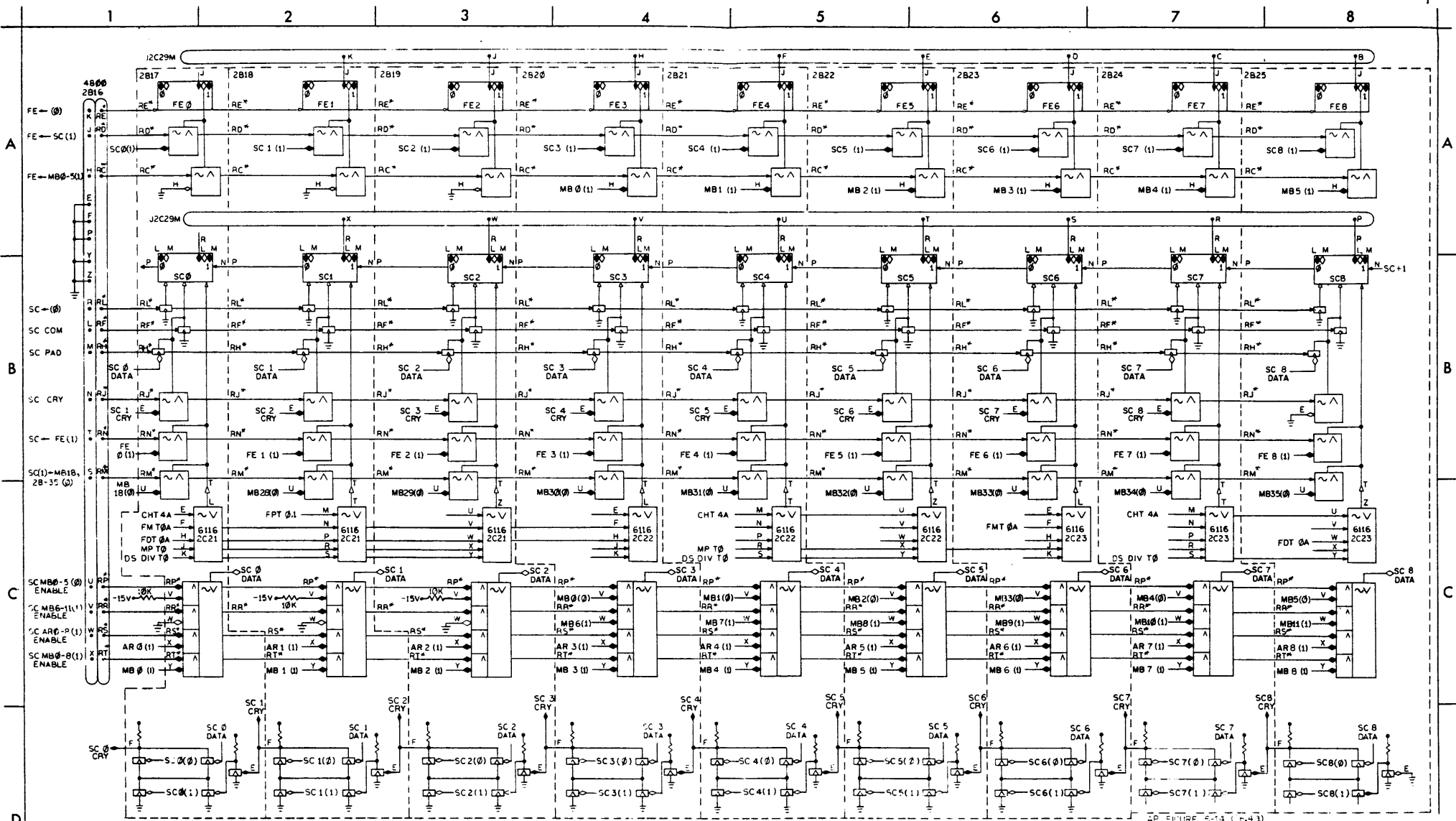
- NOTES:
1. ALL FF PACKAGES ARE 62 ϕ 5
 2. * INDICATES REAR CONNECTOR PIN
 3. GROUND PIN D.P AND Z IN ALL 62 ϕ 5'S AND 6615'S.



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

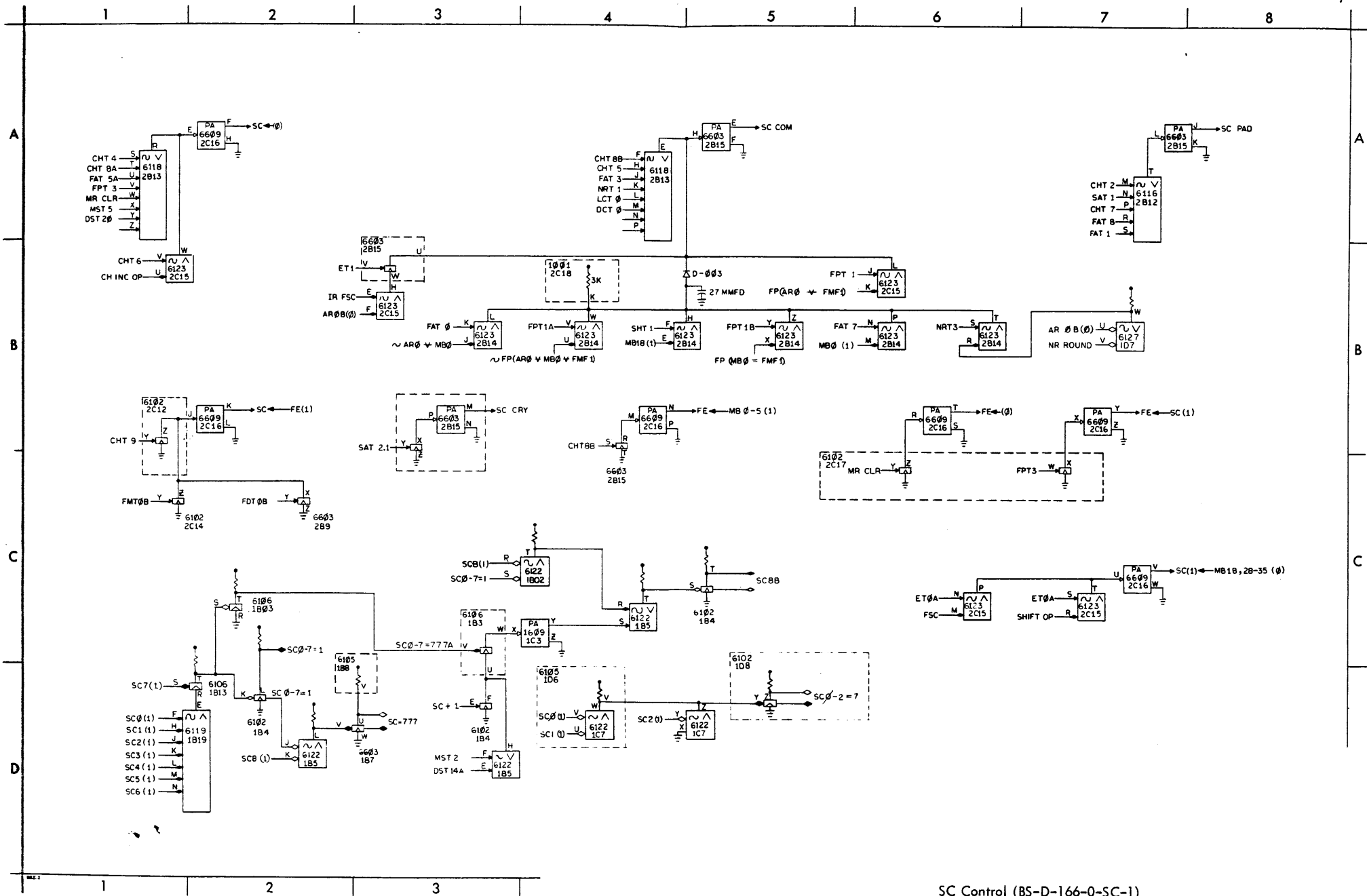


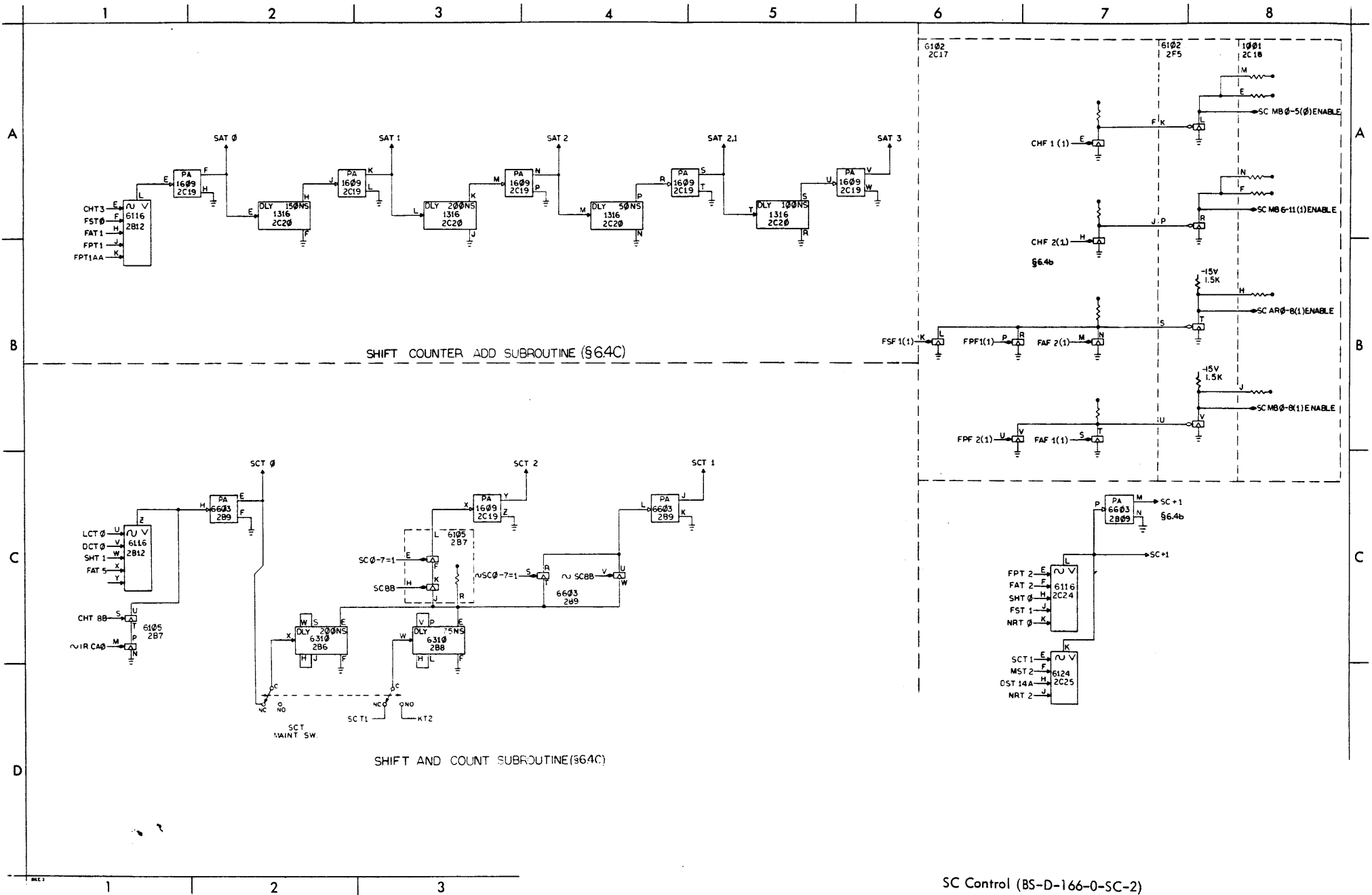
MQ - Multiplier Quotient Control
(BS-D-166-0-MQ-1)



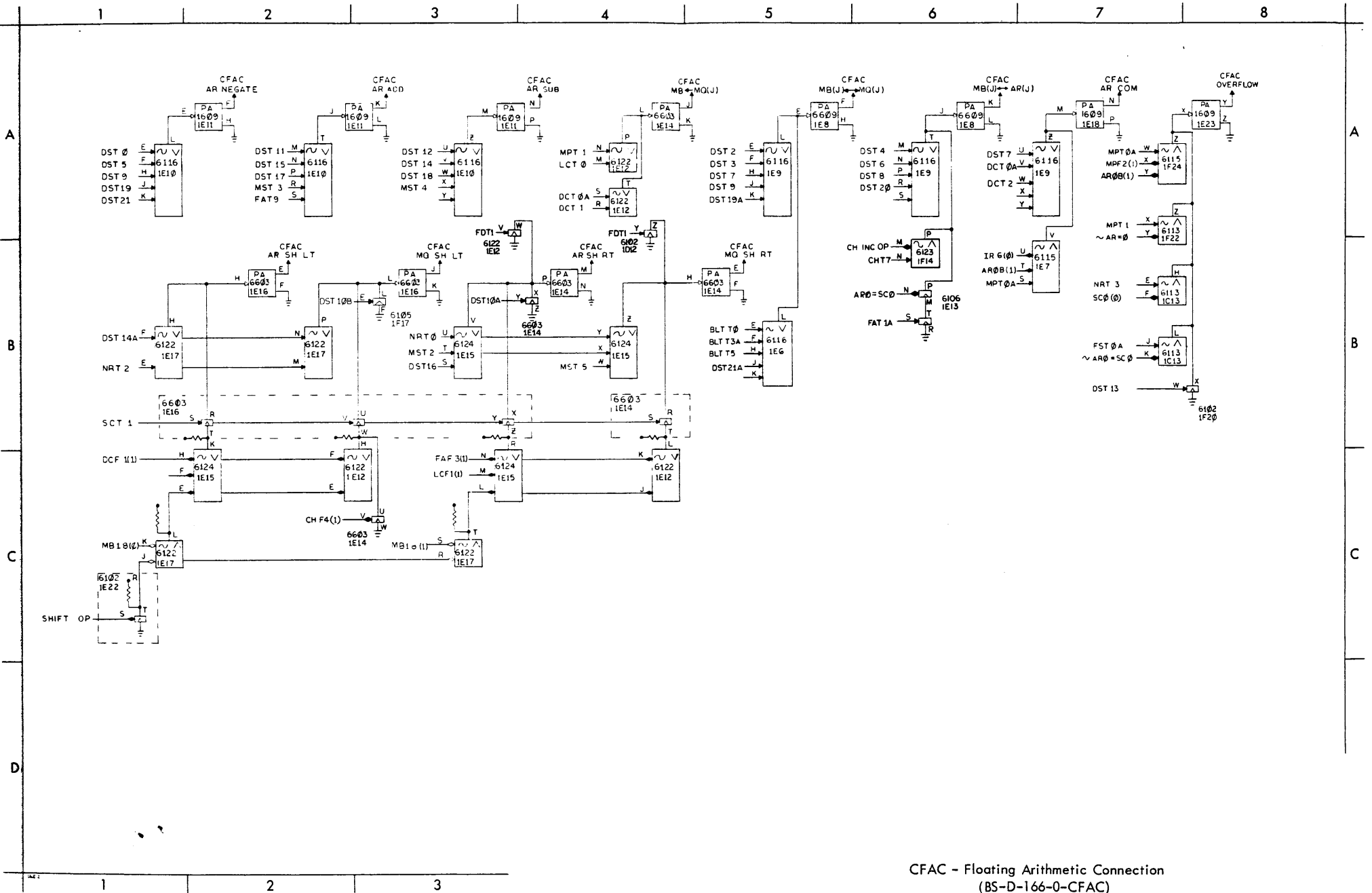
NOTE:
 1. UNLESS OTHERWISE INDICATED
 ALL PACKAGES ARE 6203s.
 2. * INDICATES REAR CONNECTOR PIN.

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

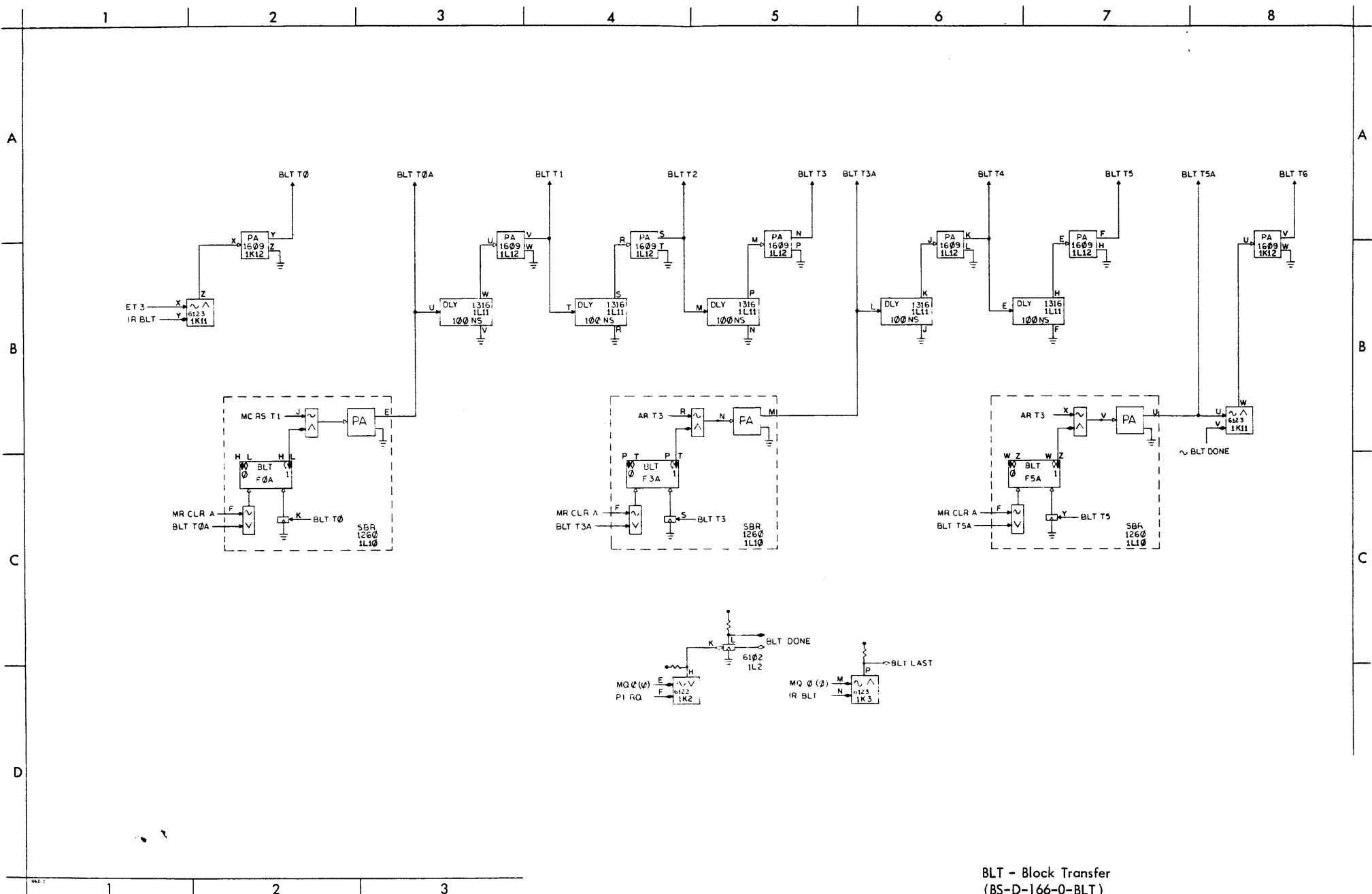




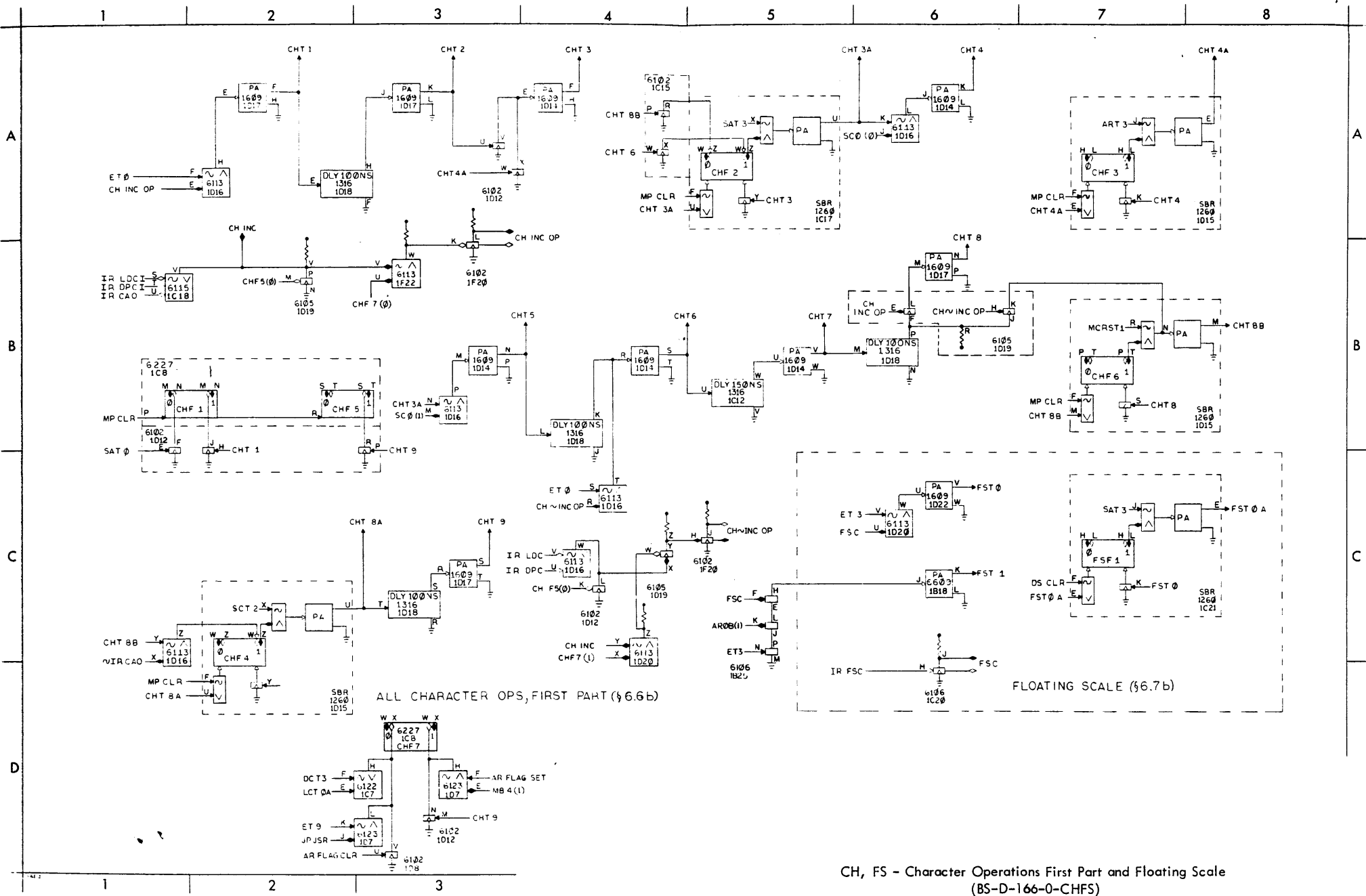
SC Control (BS-D-166-0-SC-2)



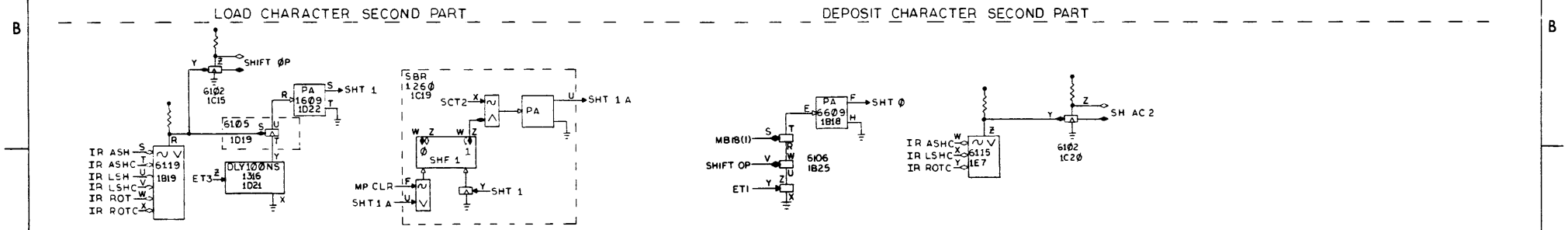
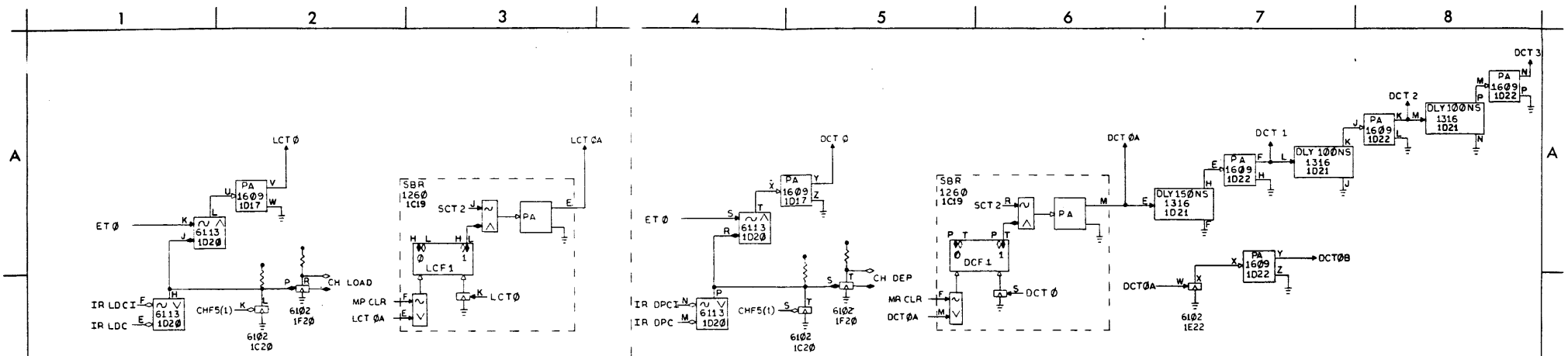
CFAC - Floating Arithmetic Connection
(BS-D-166-0-CFAC)



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



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)

1 2 3 4 5 6 7 8

A

B

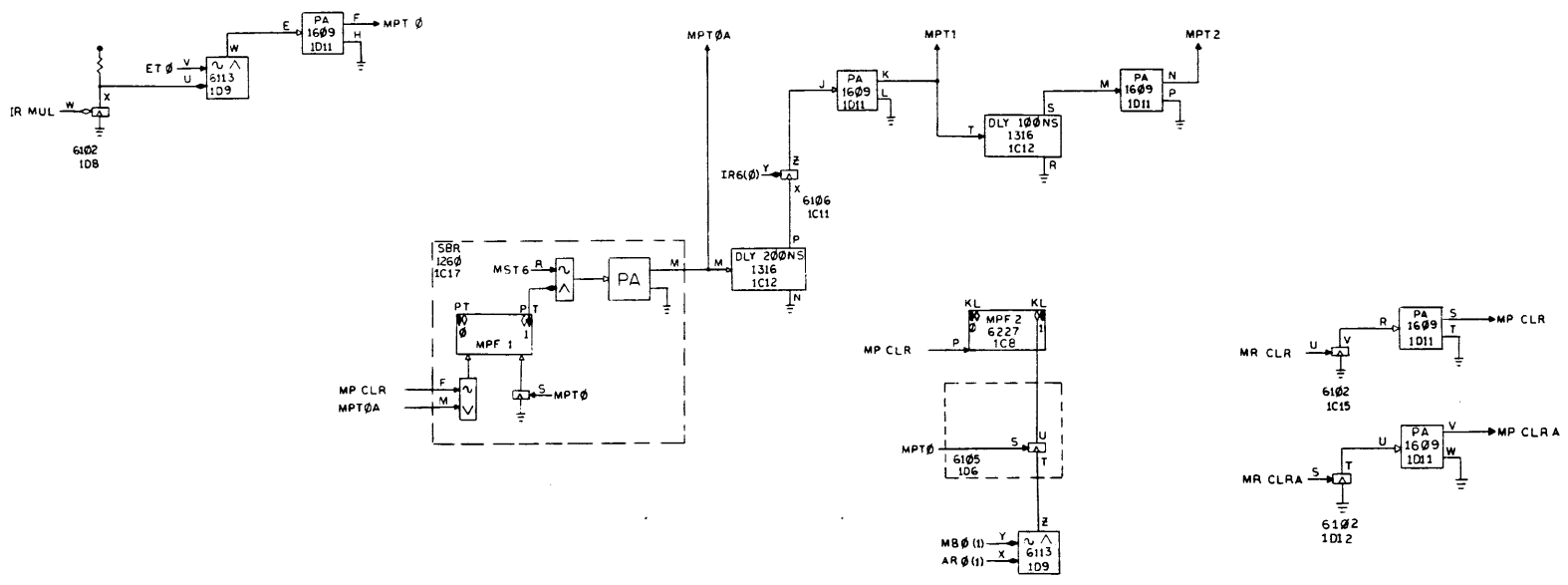
C

D

A

B

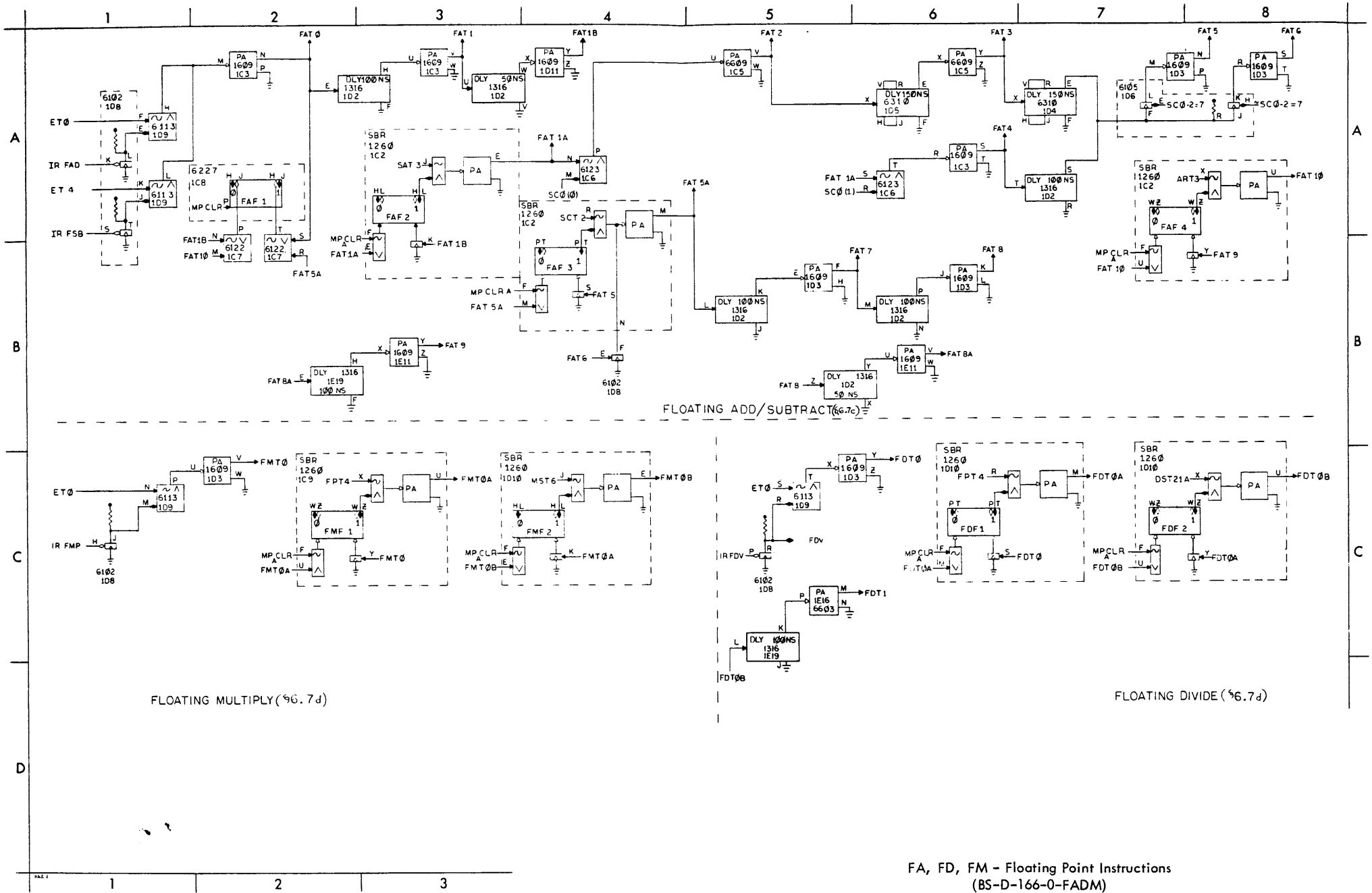
C



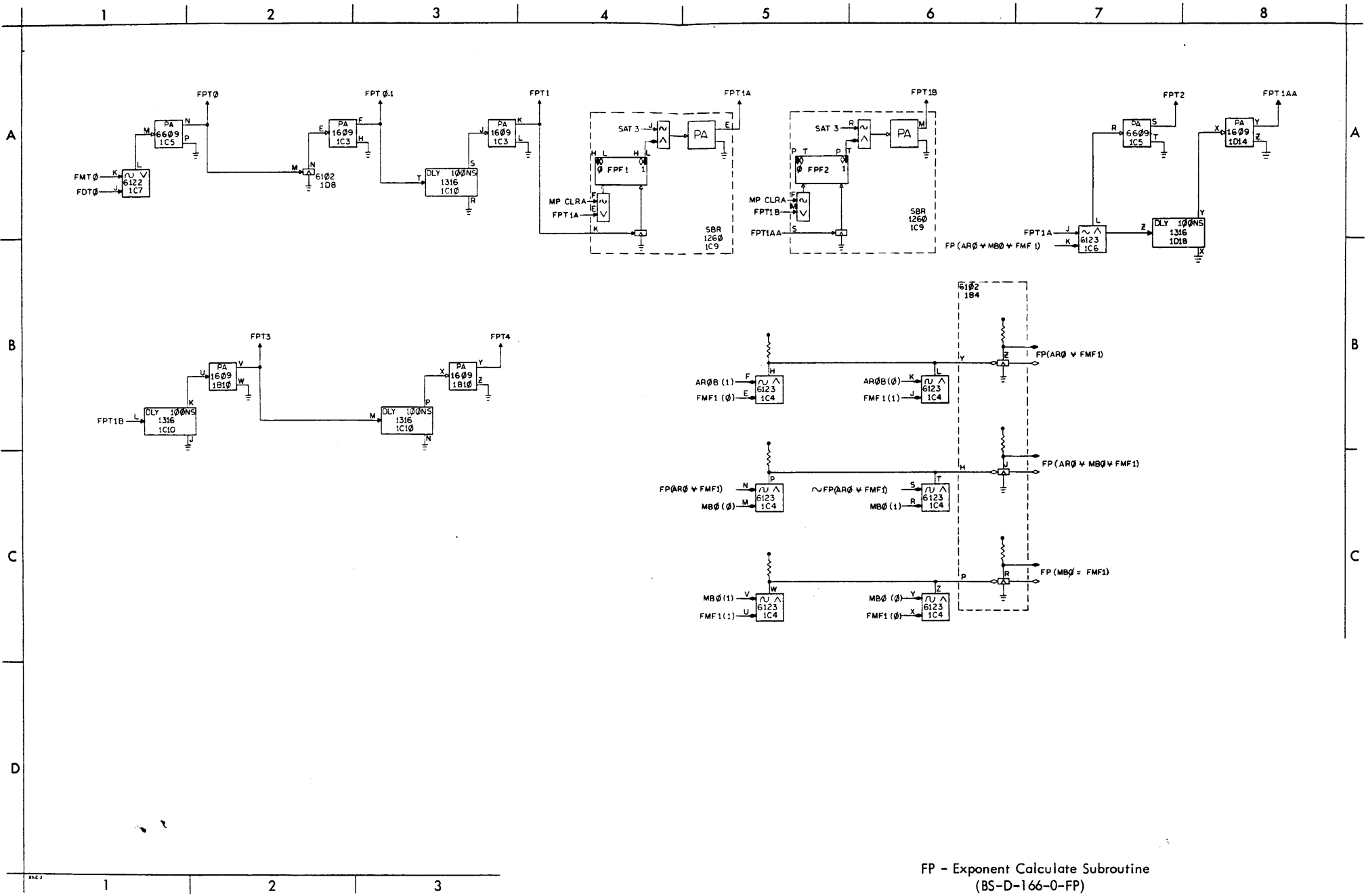
FIXED POINT MULTIPLY

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

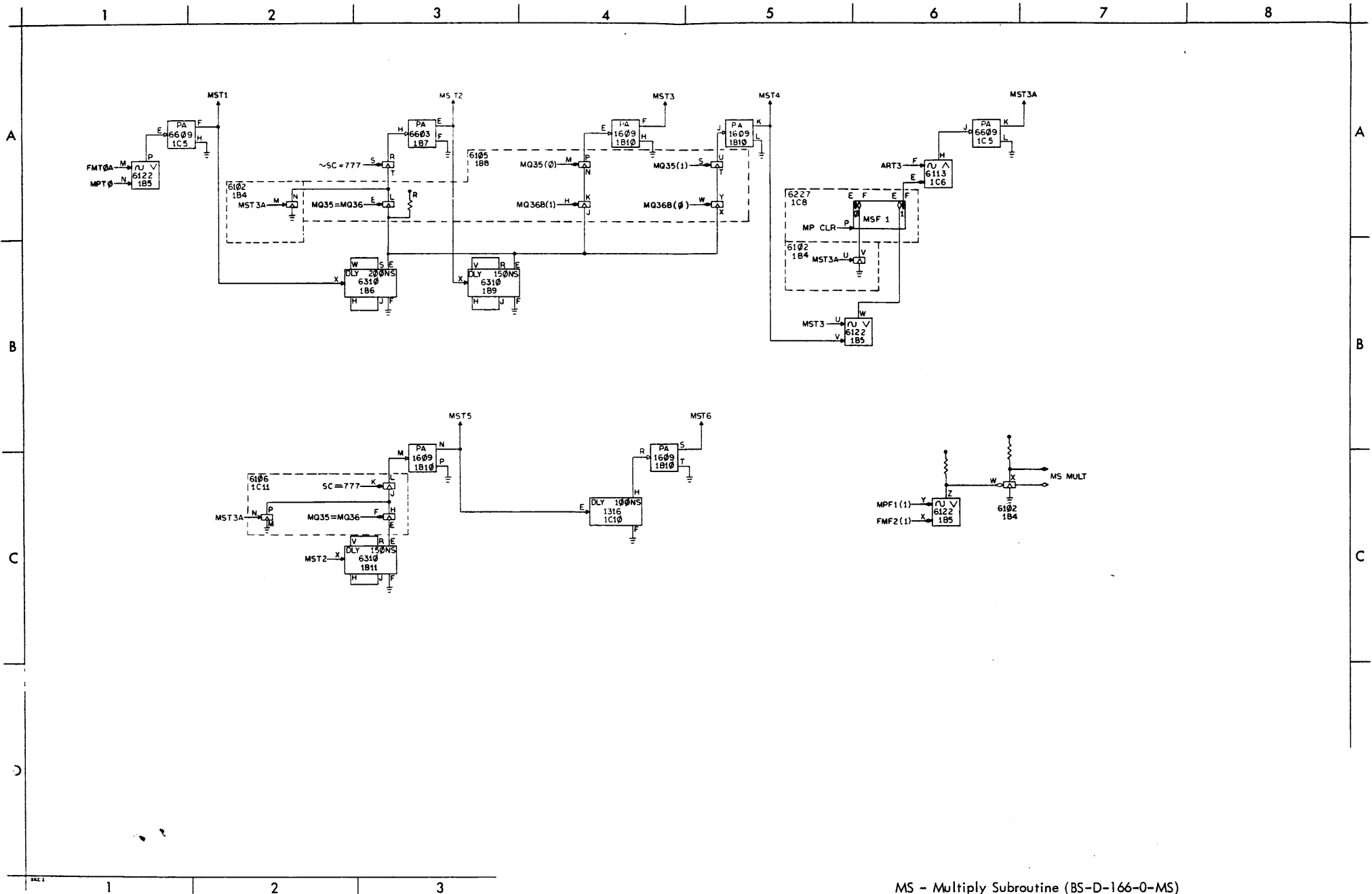
1 2 3



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

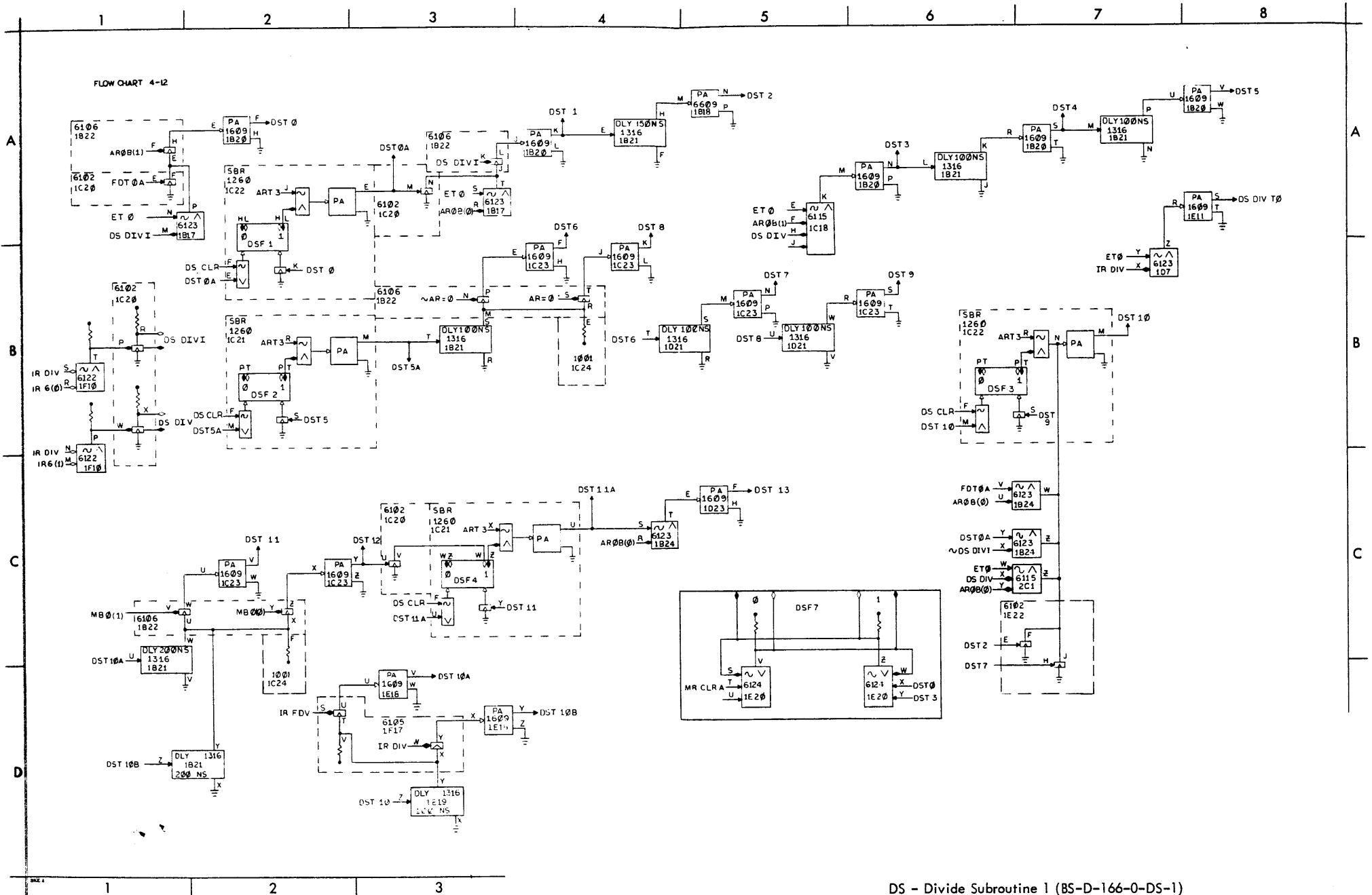


FP - Exponent Calculate Subroutine
(BS-D-166-0-FP)



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

FLOW CHART 4-12



DS - Divide Subroutine 1 (BS-D-166-0-DS-1)

1 2 3 4 5 6 7 8

A

B

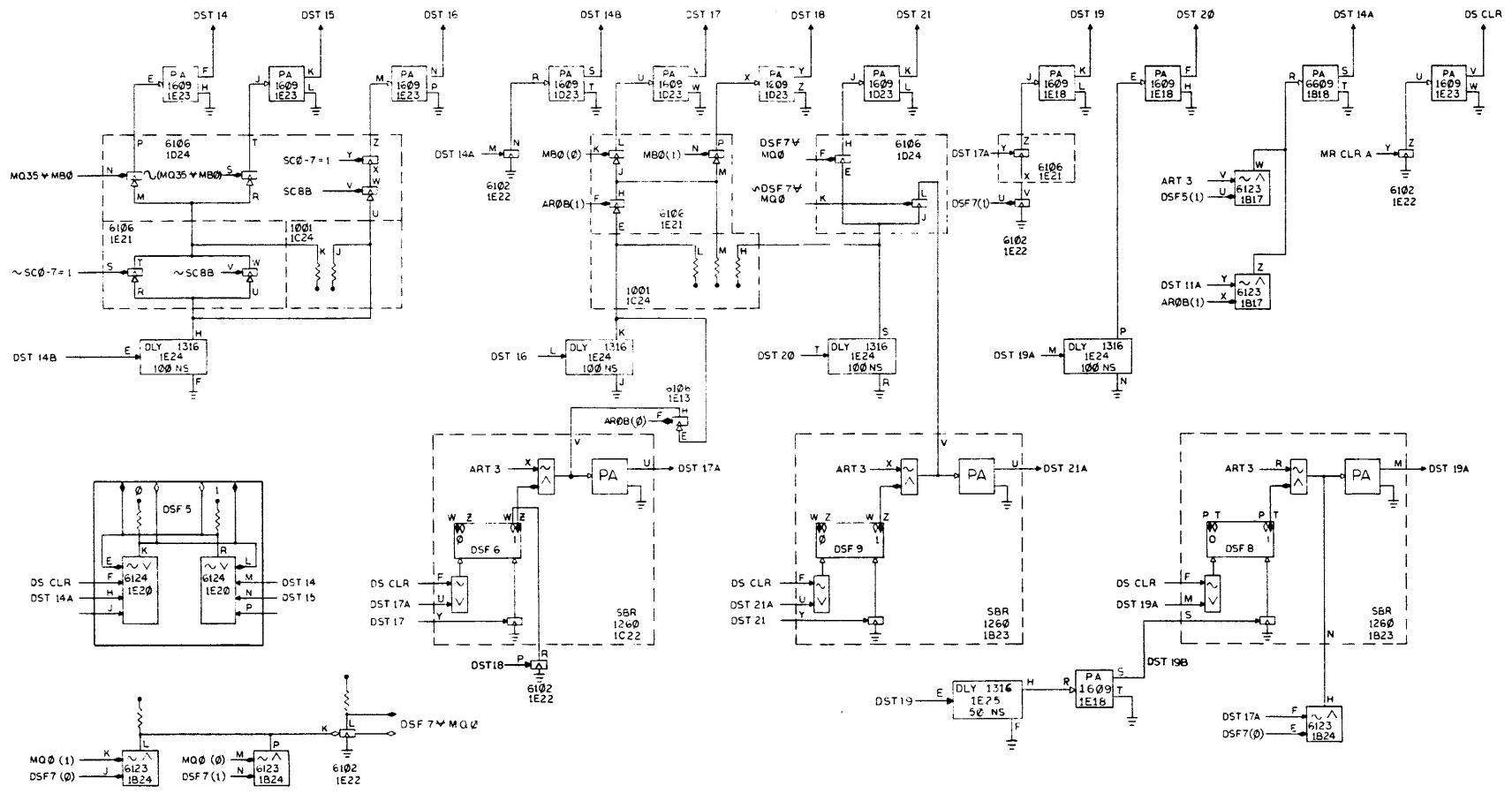
C

D

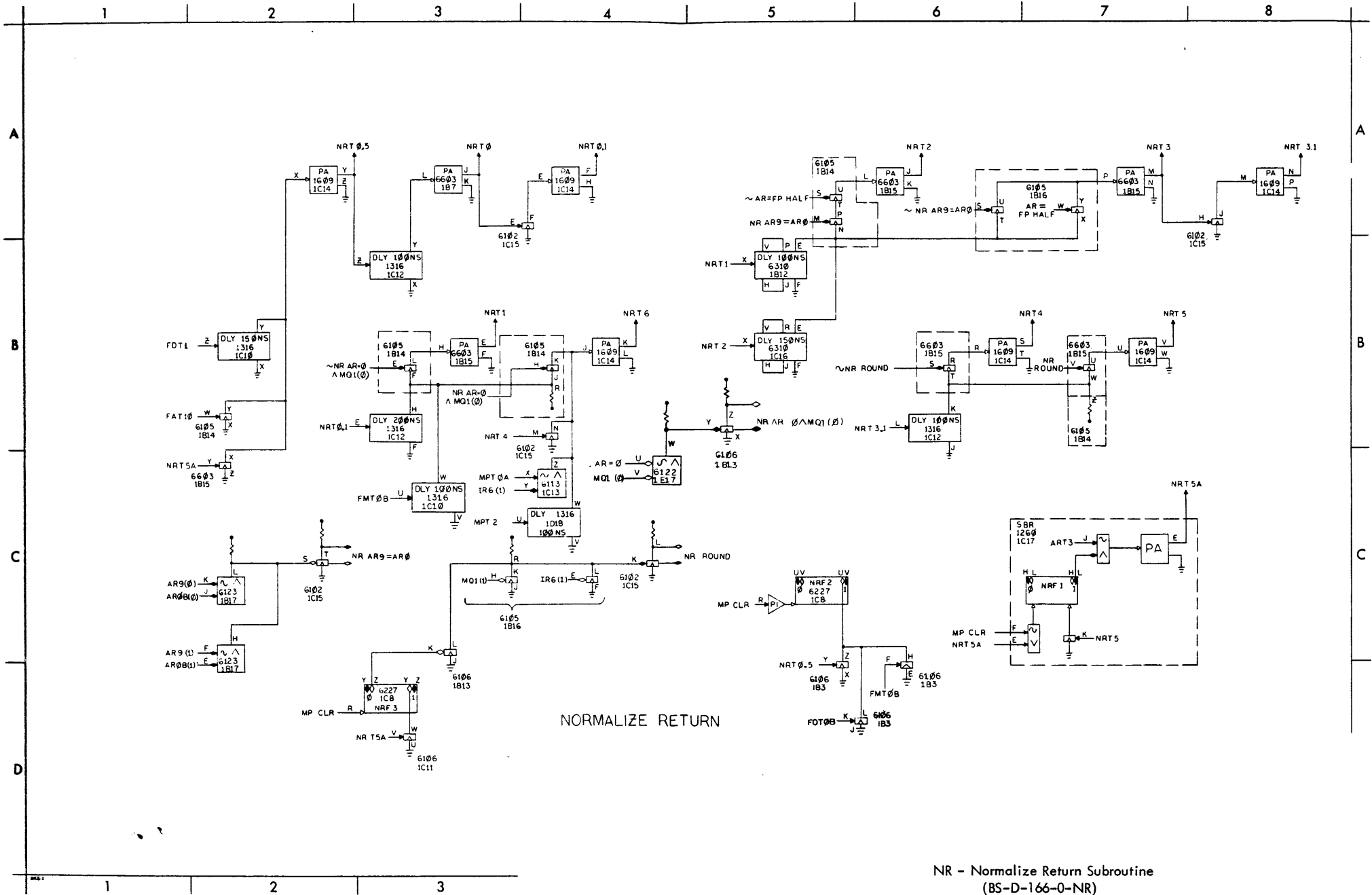
A

B

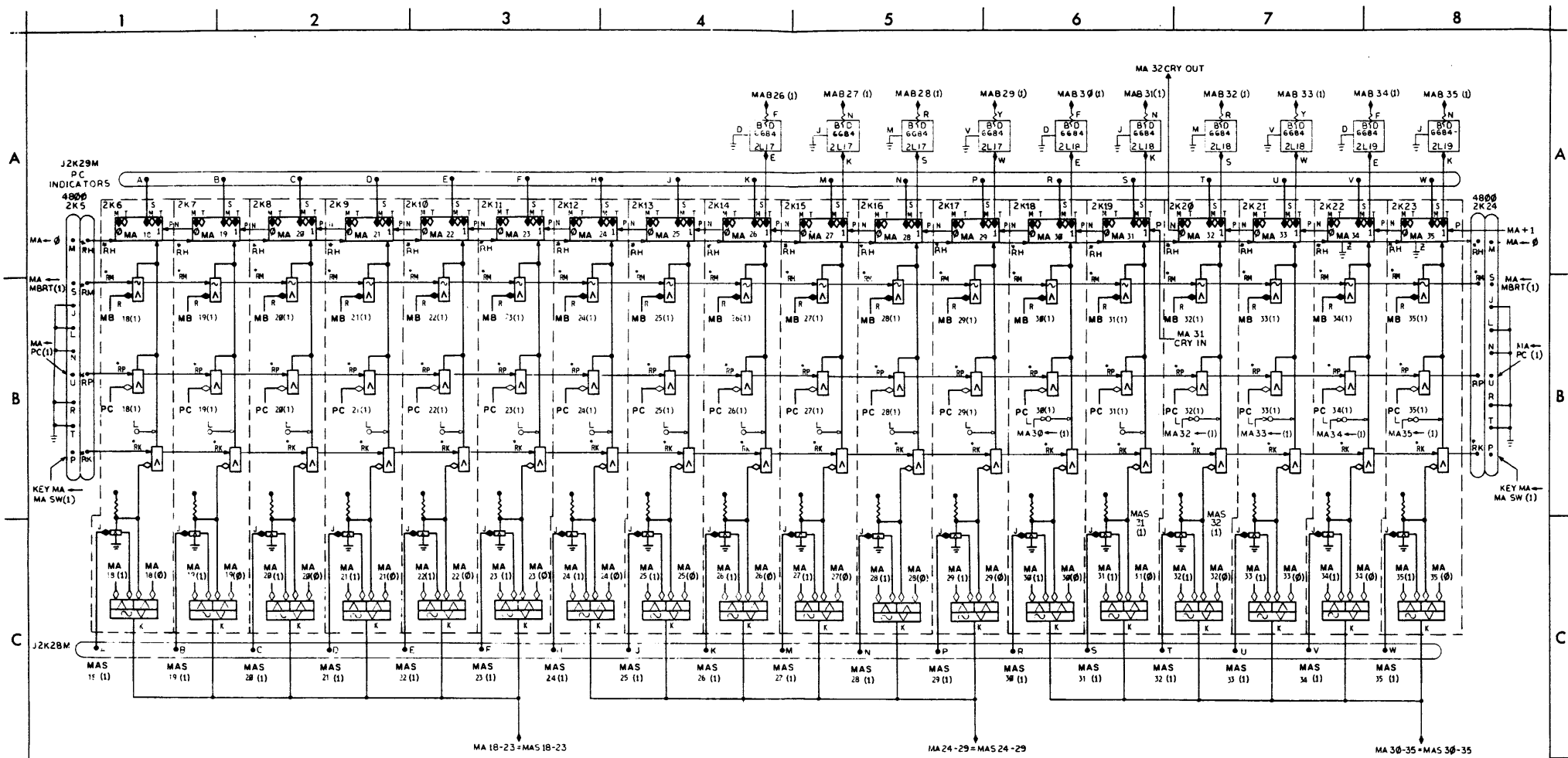
C



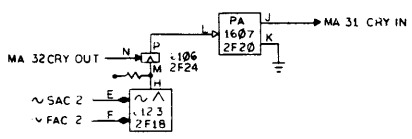
DS - Divide Subroutine 2 (BS-D-166-0-DS-2)



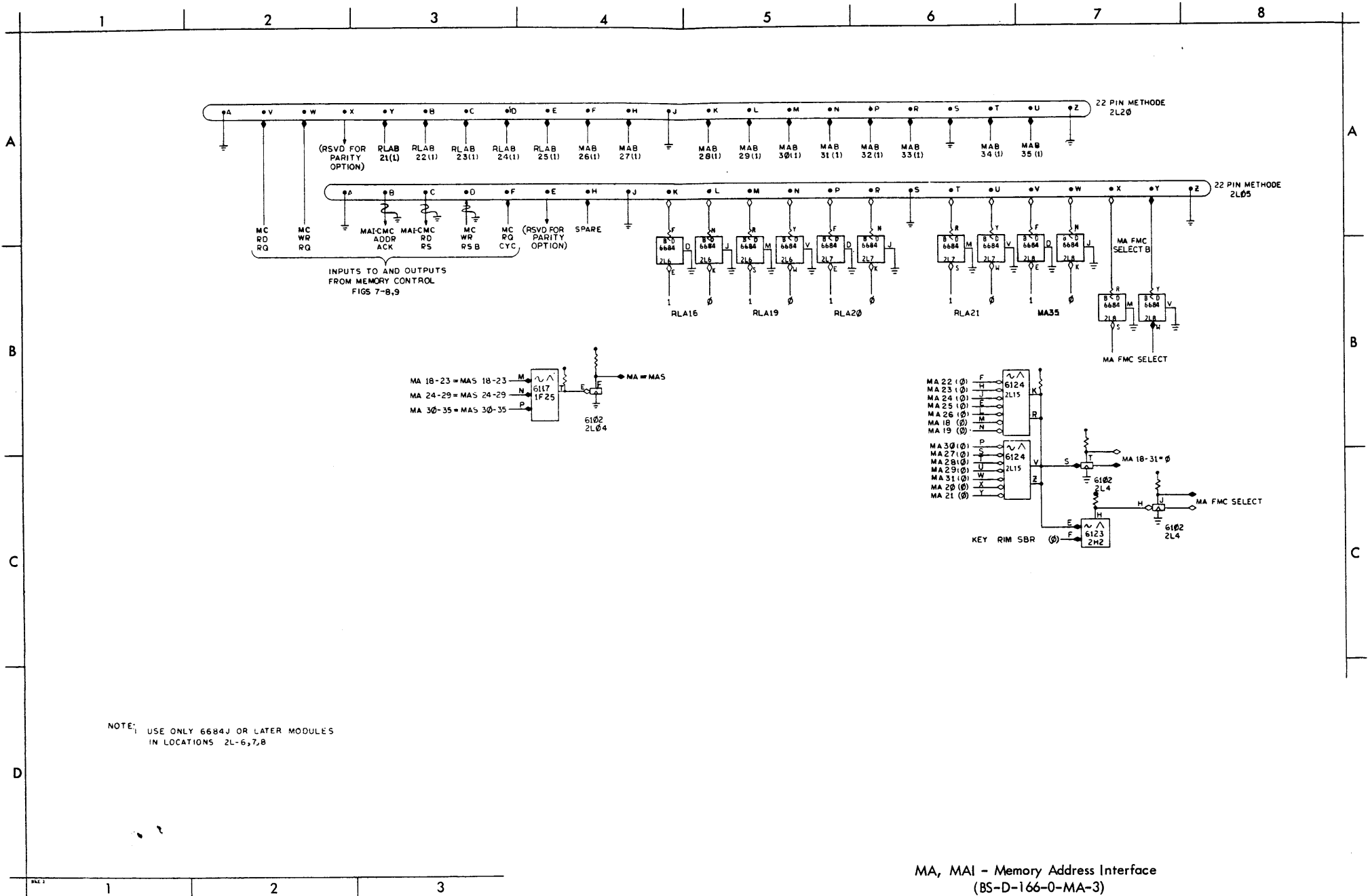
NR - Normalize Return Subroutine
(BS-D-166-0-NR)

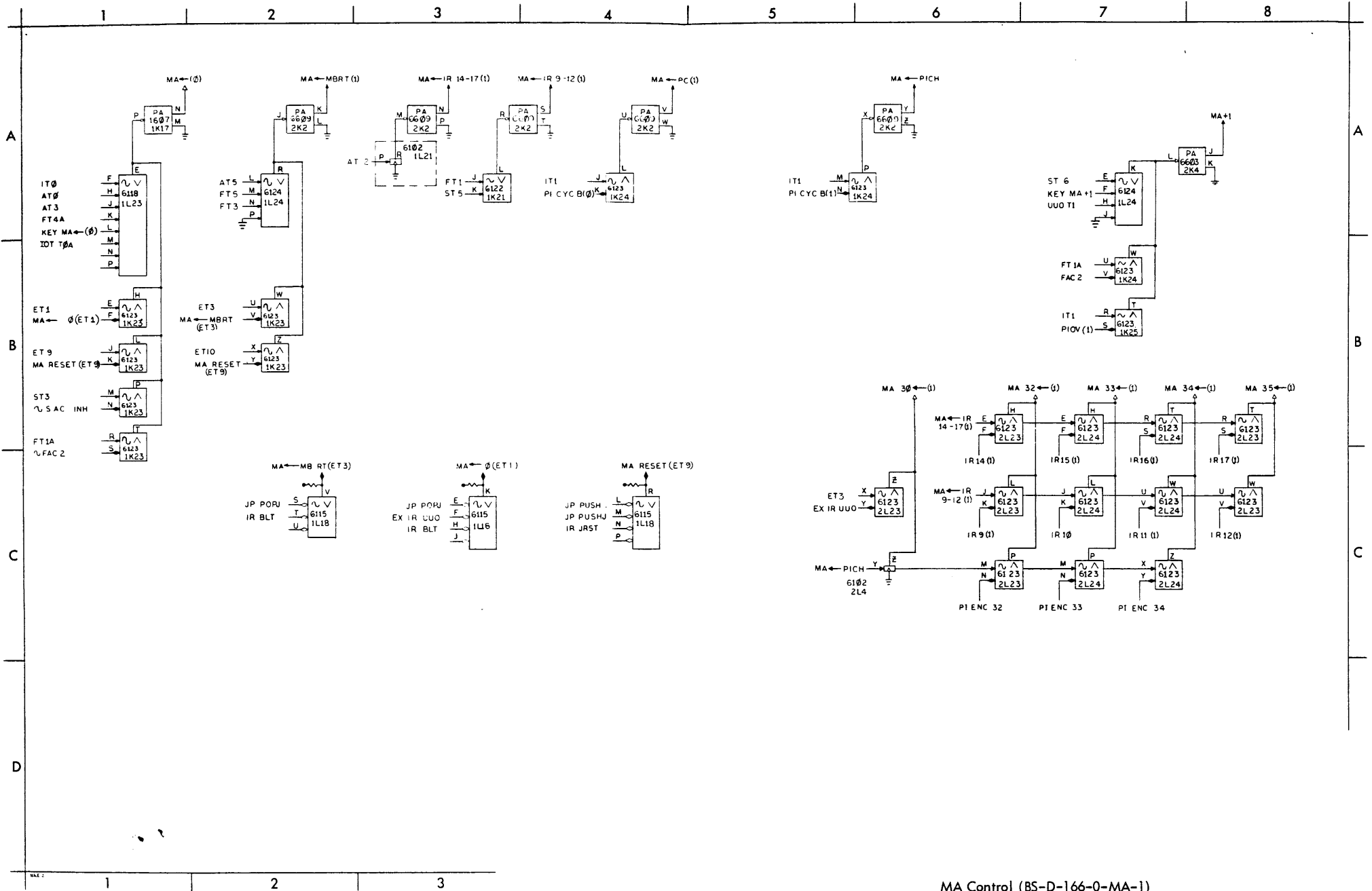


- NOTE
1. ALL FF PACKAGES ARE 6206
 2. - INDICATES REAR CONNECTOR PIN.
 3. GROUND PIN D AND Z IN ALL 6206'S.
 4. USE ONLY 6684 OR LATER MODULES IN LOCATIONS 2L-17,18,19

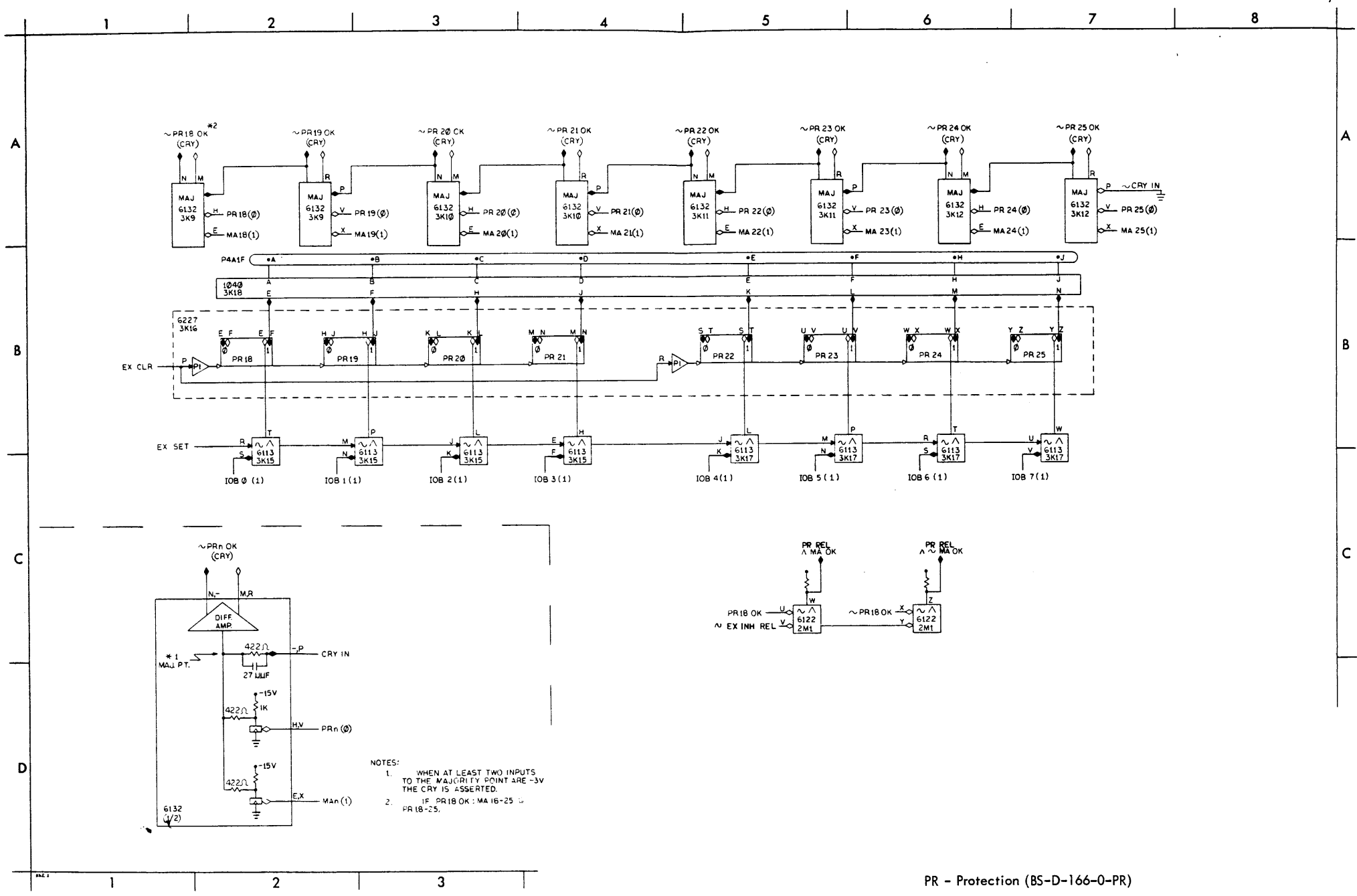


MA - Memory Address 18-35
(BS-D-166-0-MA-2)

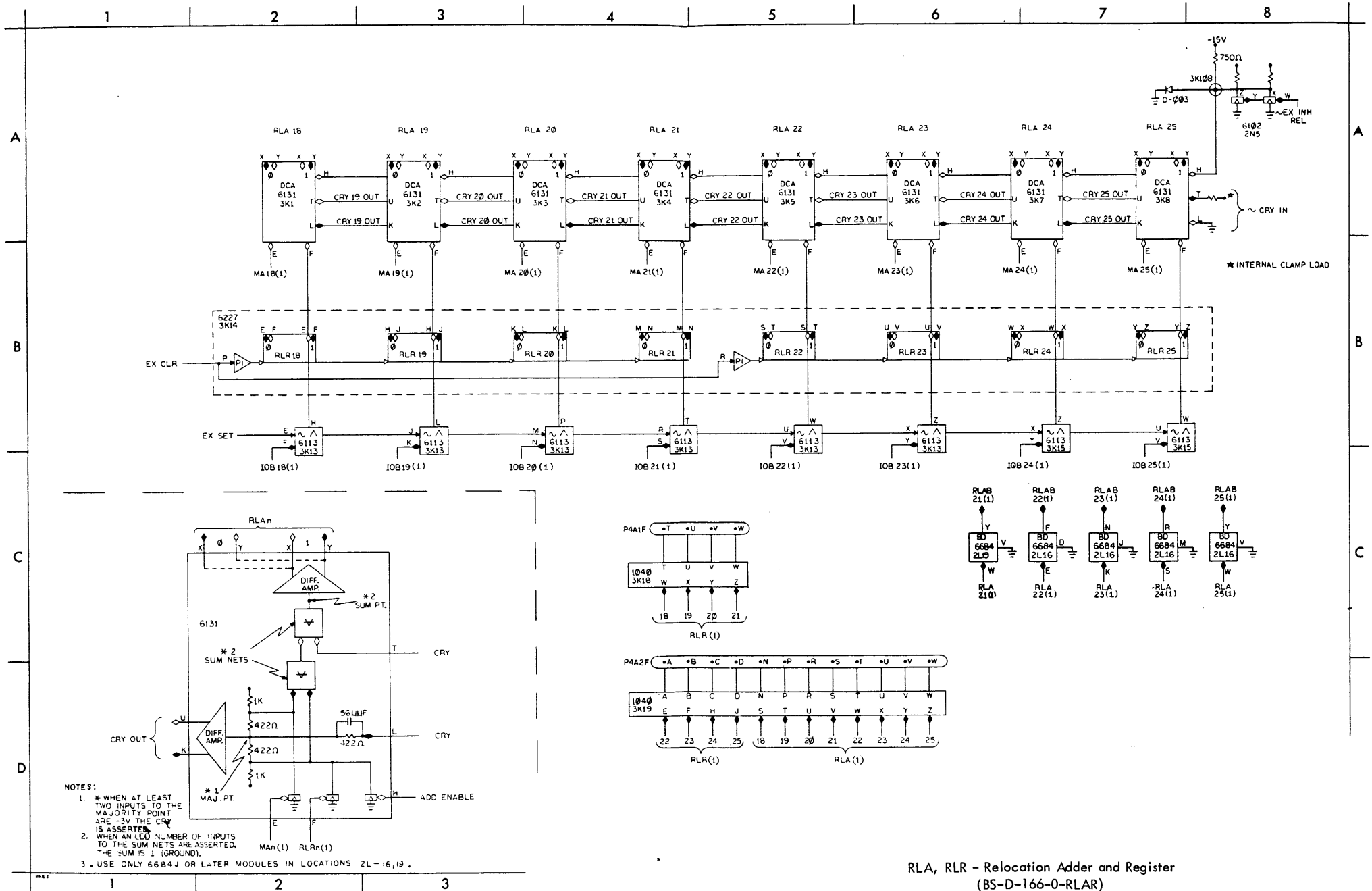




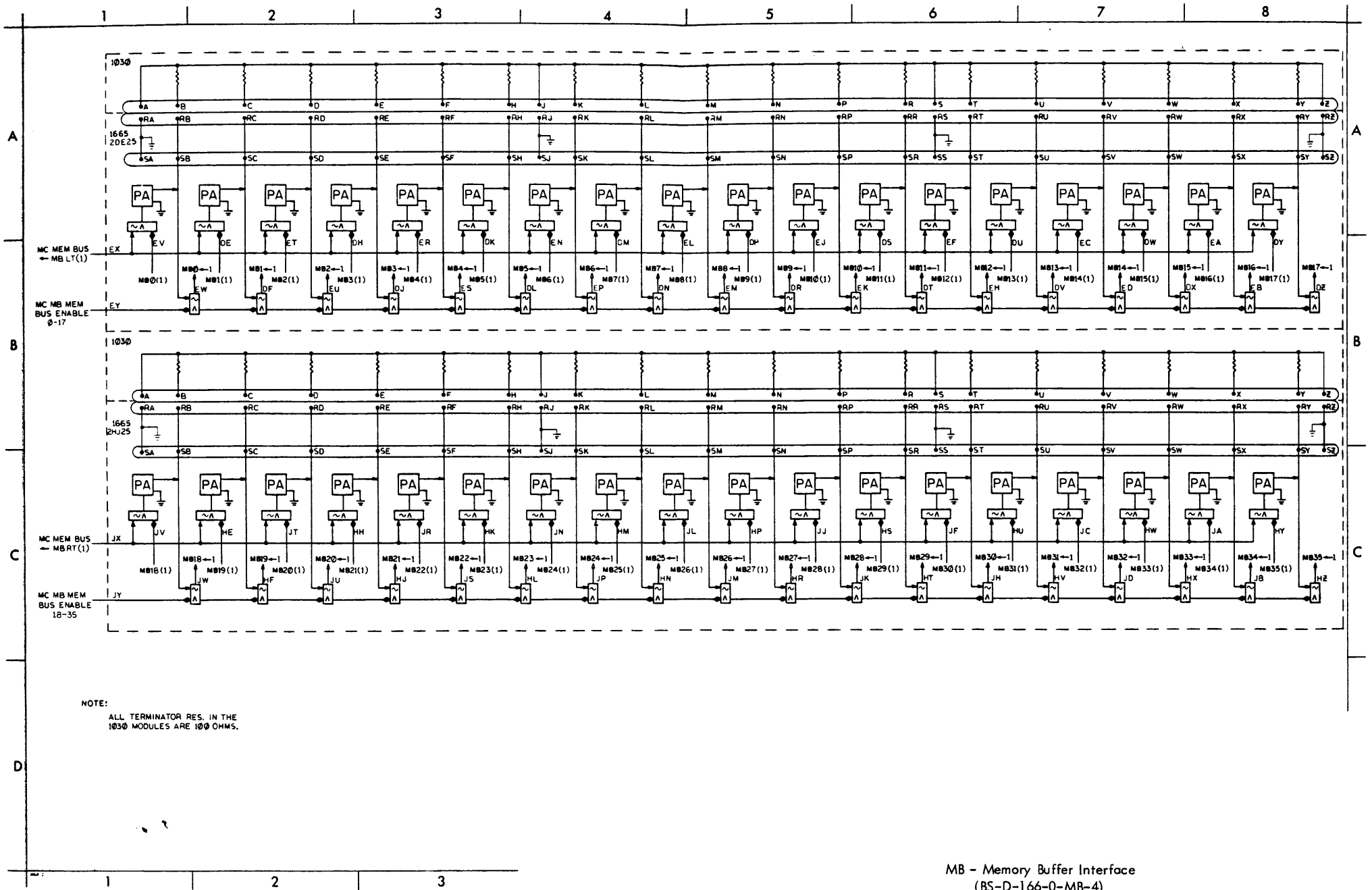
MA Control (BS-D-166-0-MA-1)

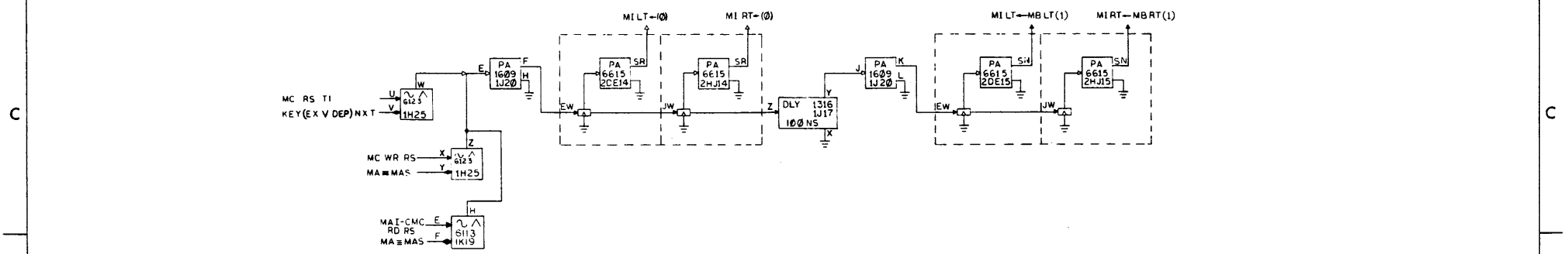
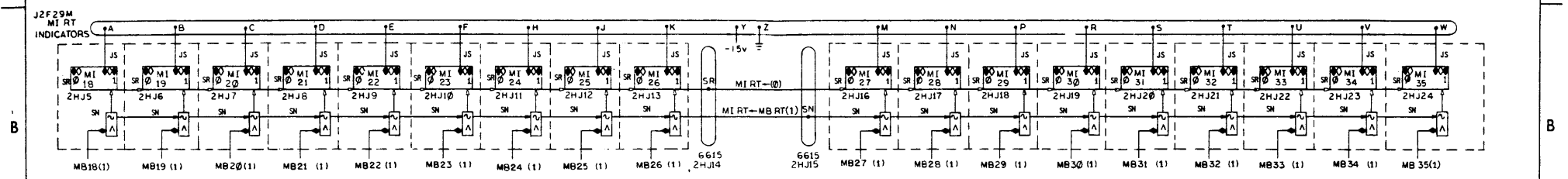
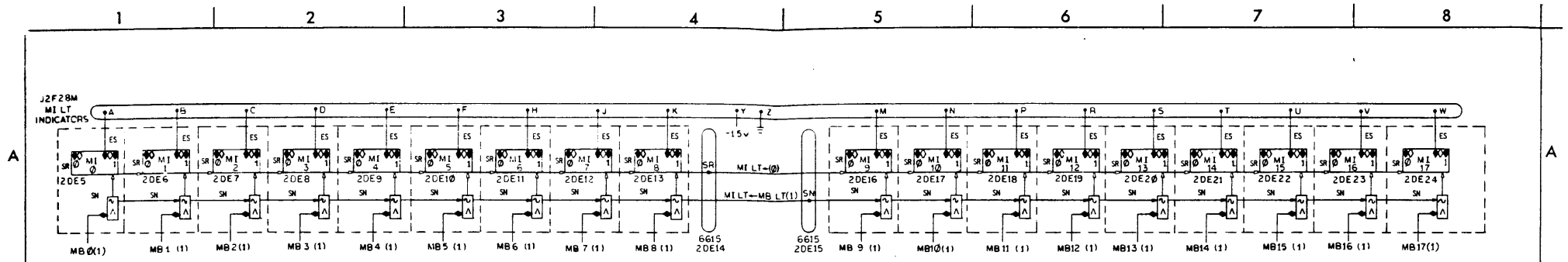


- NOTES:
1. WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
 2. IF PR18 OK : MA18-25 = PR18-25.

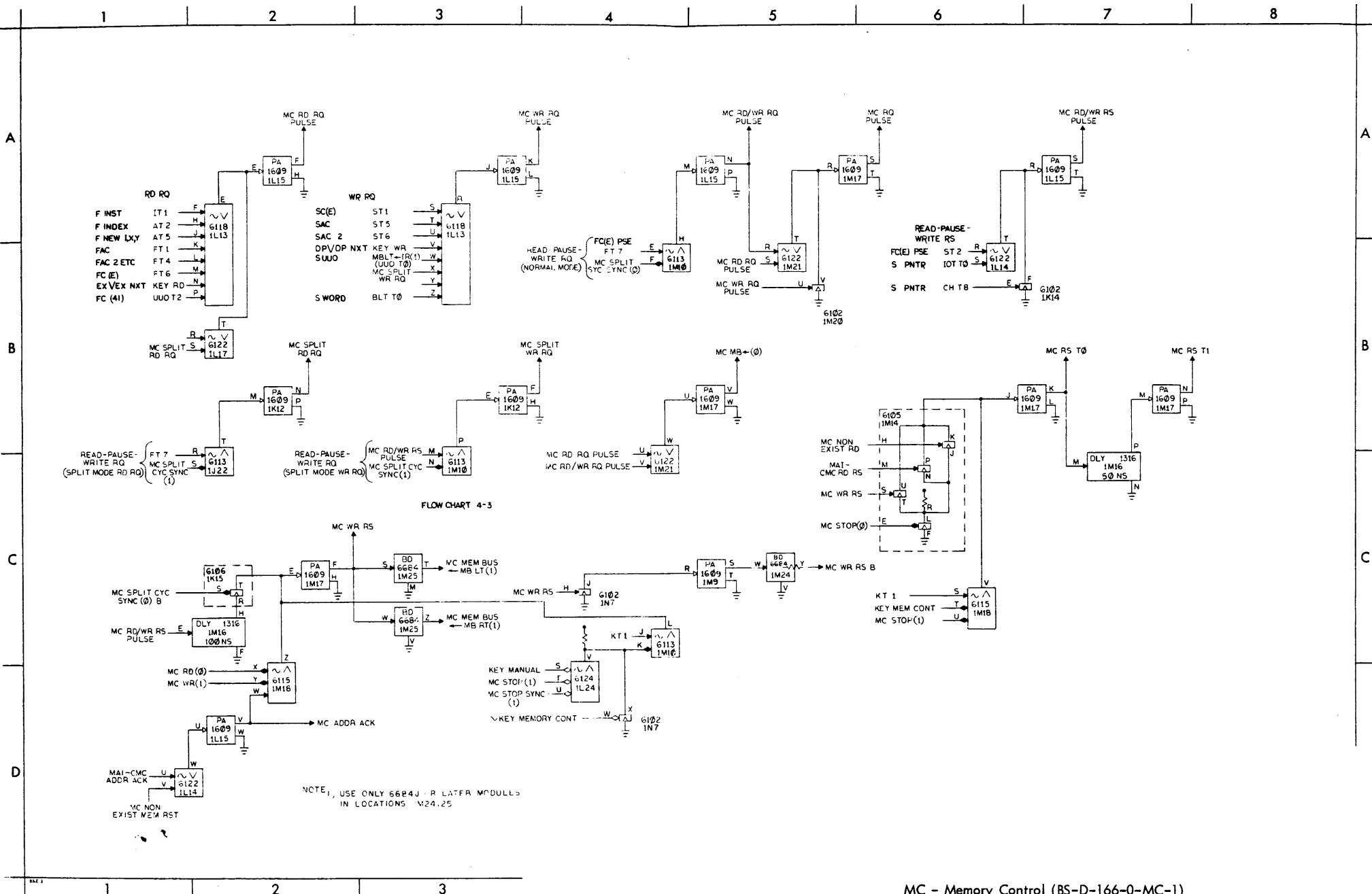


RLA, RLR - Relocation Adder and Register (BS-D-166-0-RLAR)





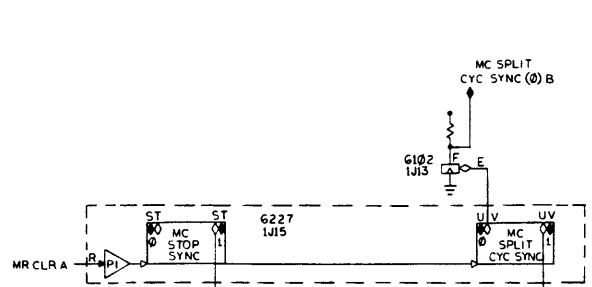
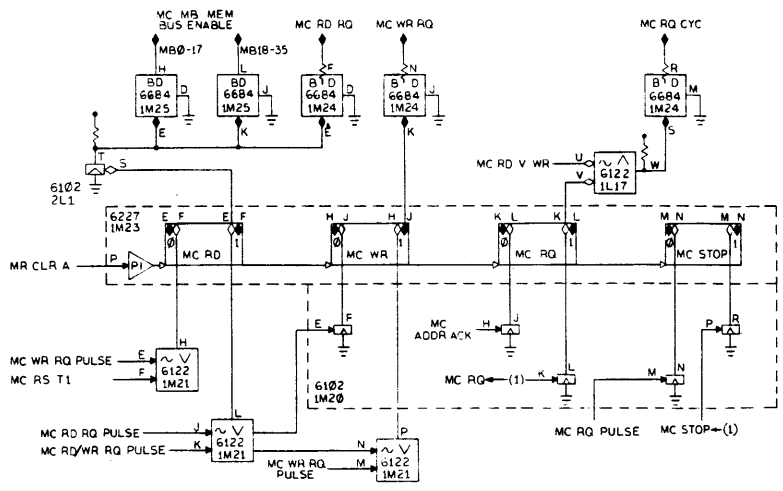
NOTE:
1. ALL FF PACKAGES ARE 6205
2. SN AND SR ARE PINS IN LOWER REAR CONNECTOR



MC - Memory Control (BS-D-166-0-MC-1)

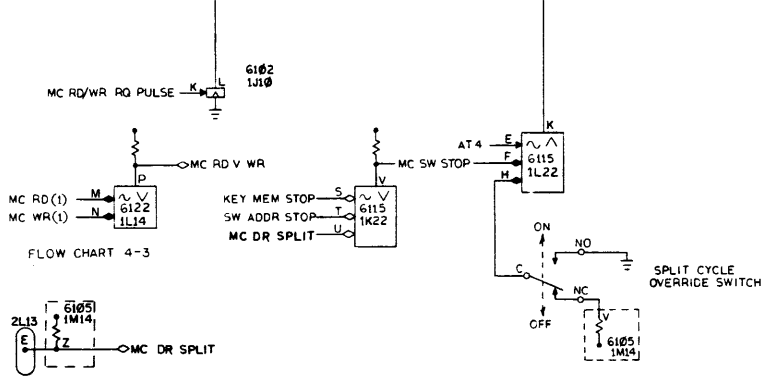
A

A



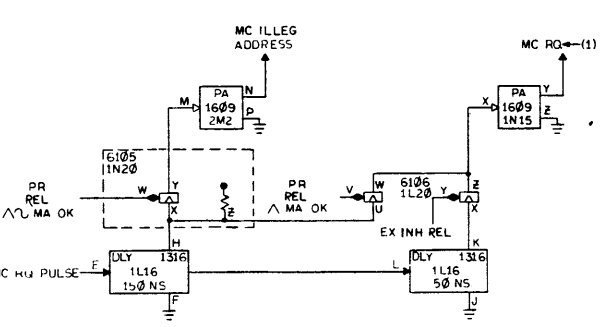
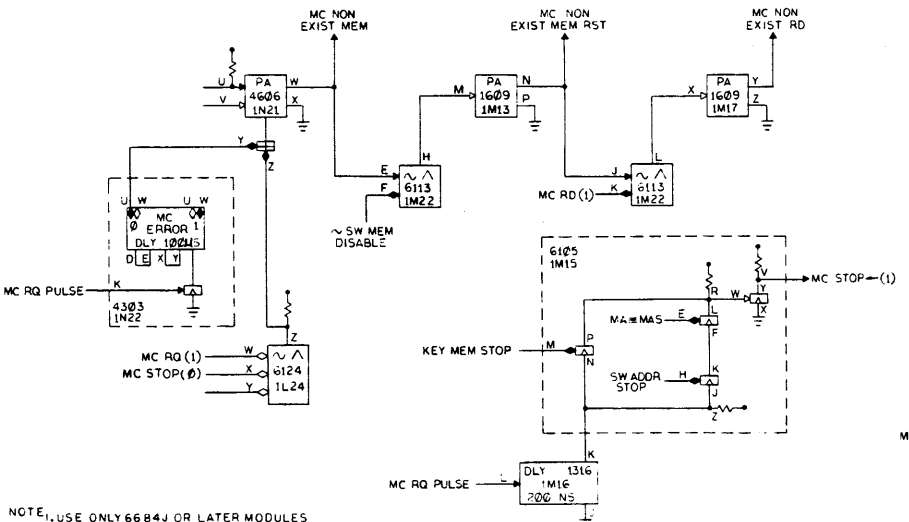
B

B



C

C



D

NOTE: USE ONLY 6684J OR LATER MODULES IN LOCATIONS 1M24, 1M25

IO BUS (§ 8.4)

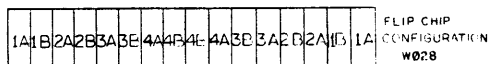
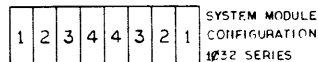
MEM BUS (§§ 7.1, 7.2)

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

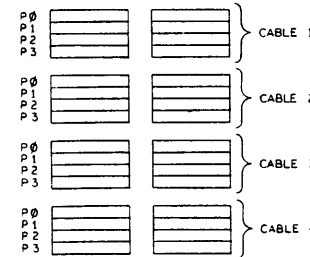
(SOURCE 3L6) (SOURCE 3L7) (SOURCE 2L13) (SOURCE 2N25)

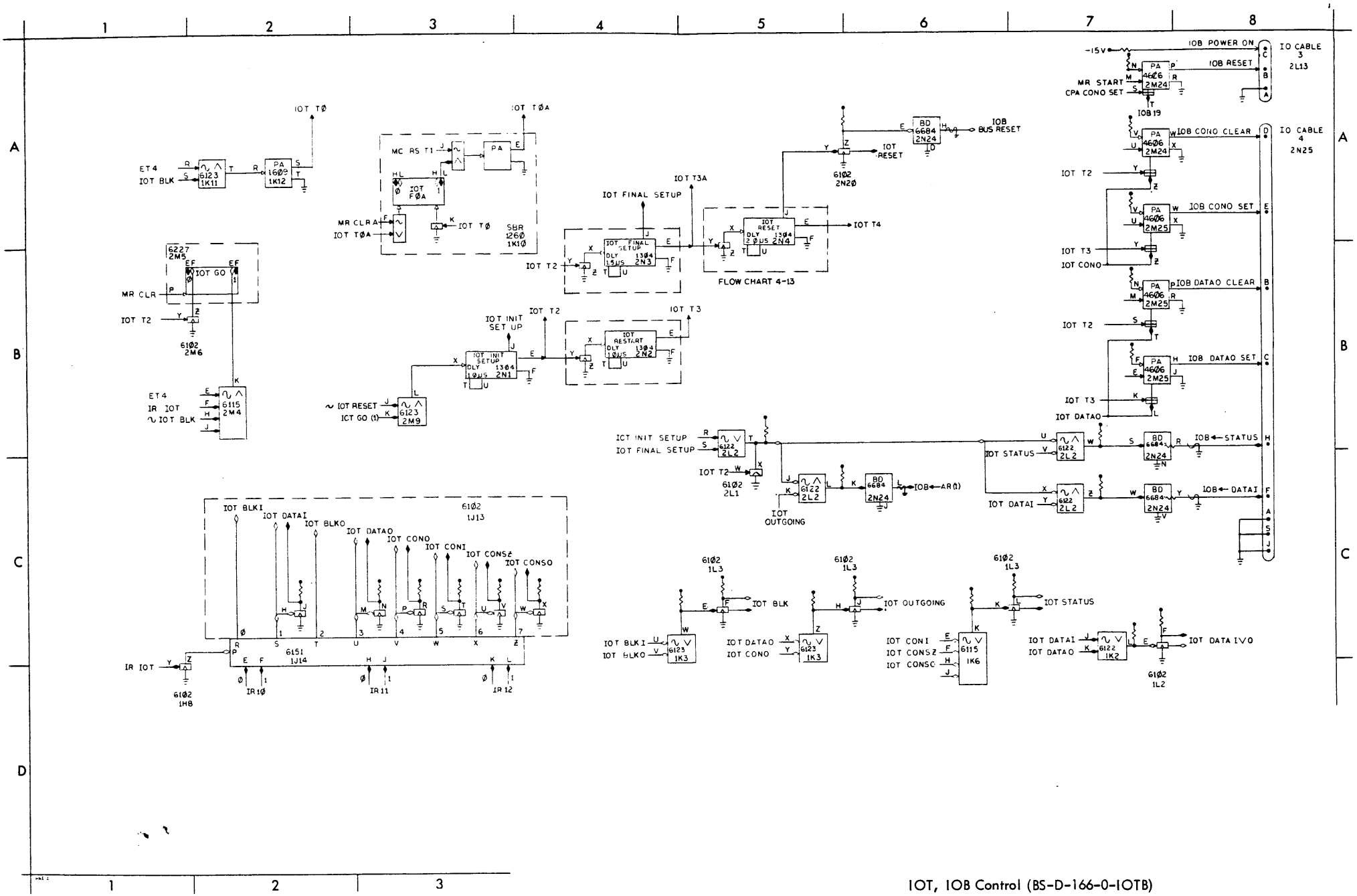
FLIP CHIP PIN	SYSTEM MODULE PIN	MEM CABLE 1	MEM CABLE 2	MEM CABLE 3	MEM CABLE 4
C	A	GND	GND	GND	GND
D	B	ADDR ACK →	MA 22 (1) →	MB 0 (1) →	MB 18 (1) →
E	C	RD RS →	MA 23 (1) →	MB 1 (1) →	MB 19 (1) →
H	D	WR RS →	MA 24 (1) →	MB 2 (1) →	MB 20 (1) →
K	E	PAR (1) →	MA 25 (1) →	MB 3 (1) →	MB 21 (1) →
M	F	RQ CYCLE →	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	MAFMC } →	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 2L5) (SOURCE 2L20) (SOURCE 2E25) (SOURCE 2J25)

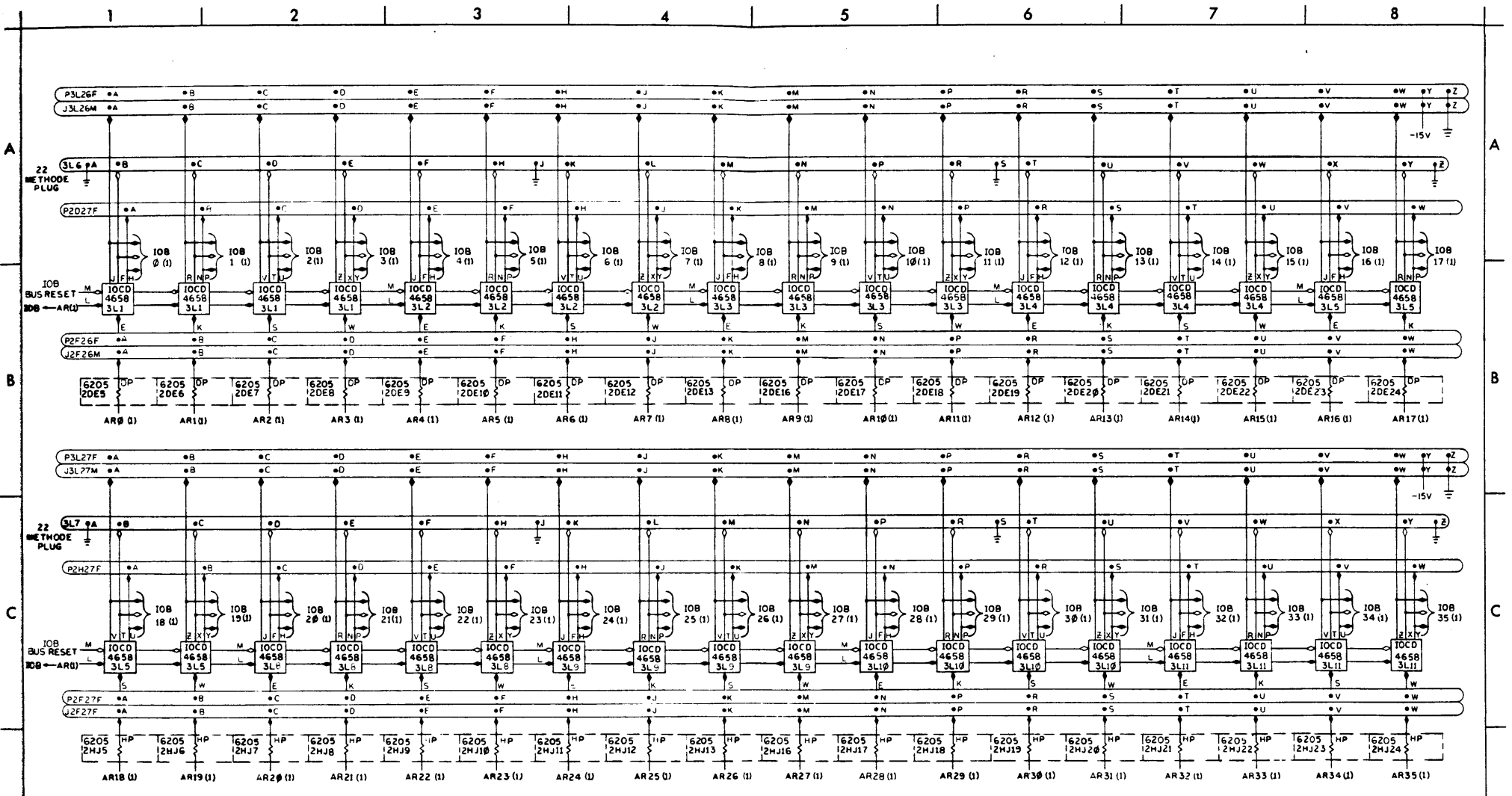


NOTE: 1 SYSTEM MODULE CONNECTOR (1032) = 2 FLIP CHIP CONNECTORS (W02B)

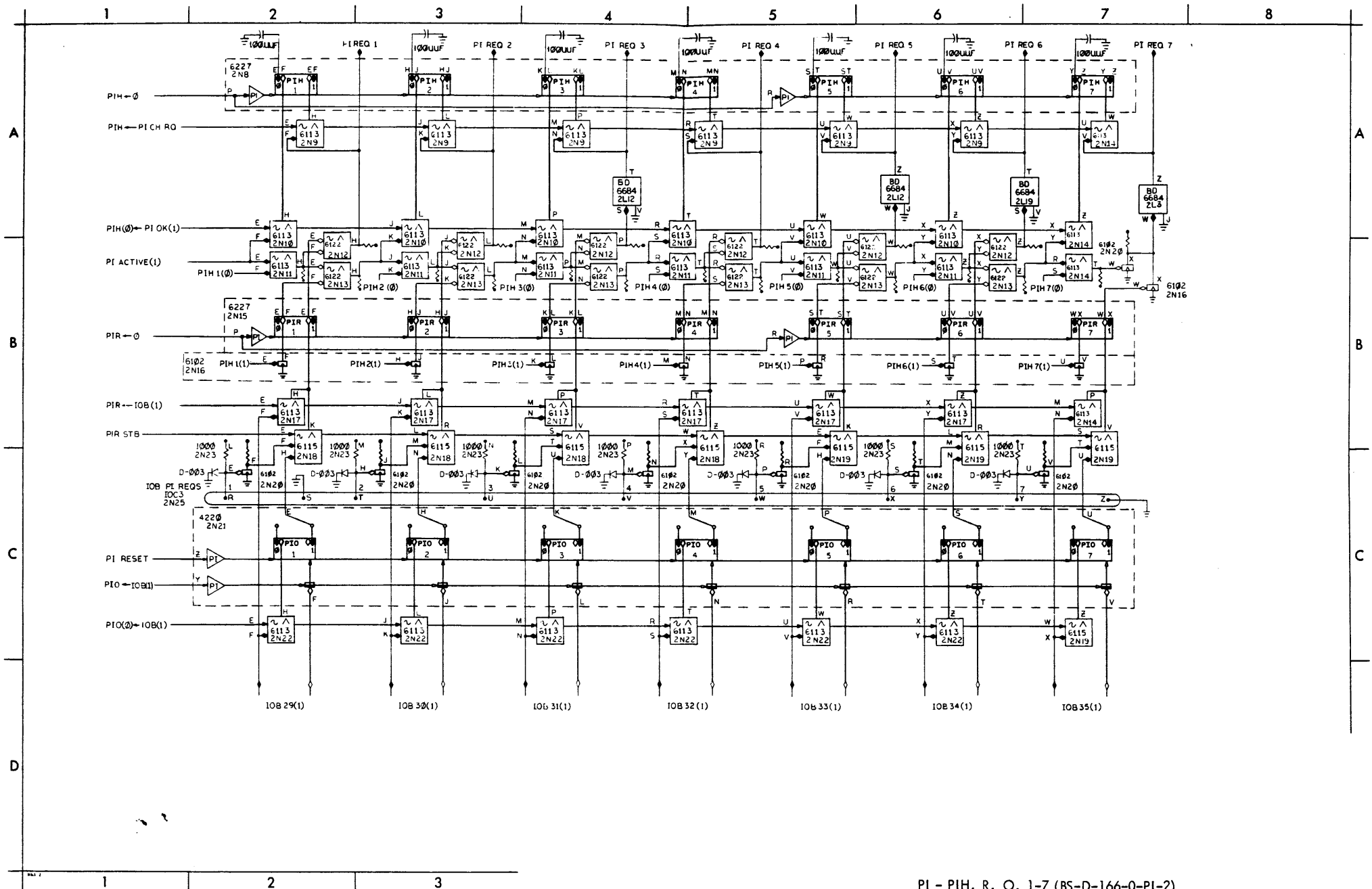




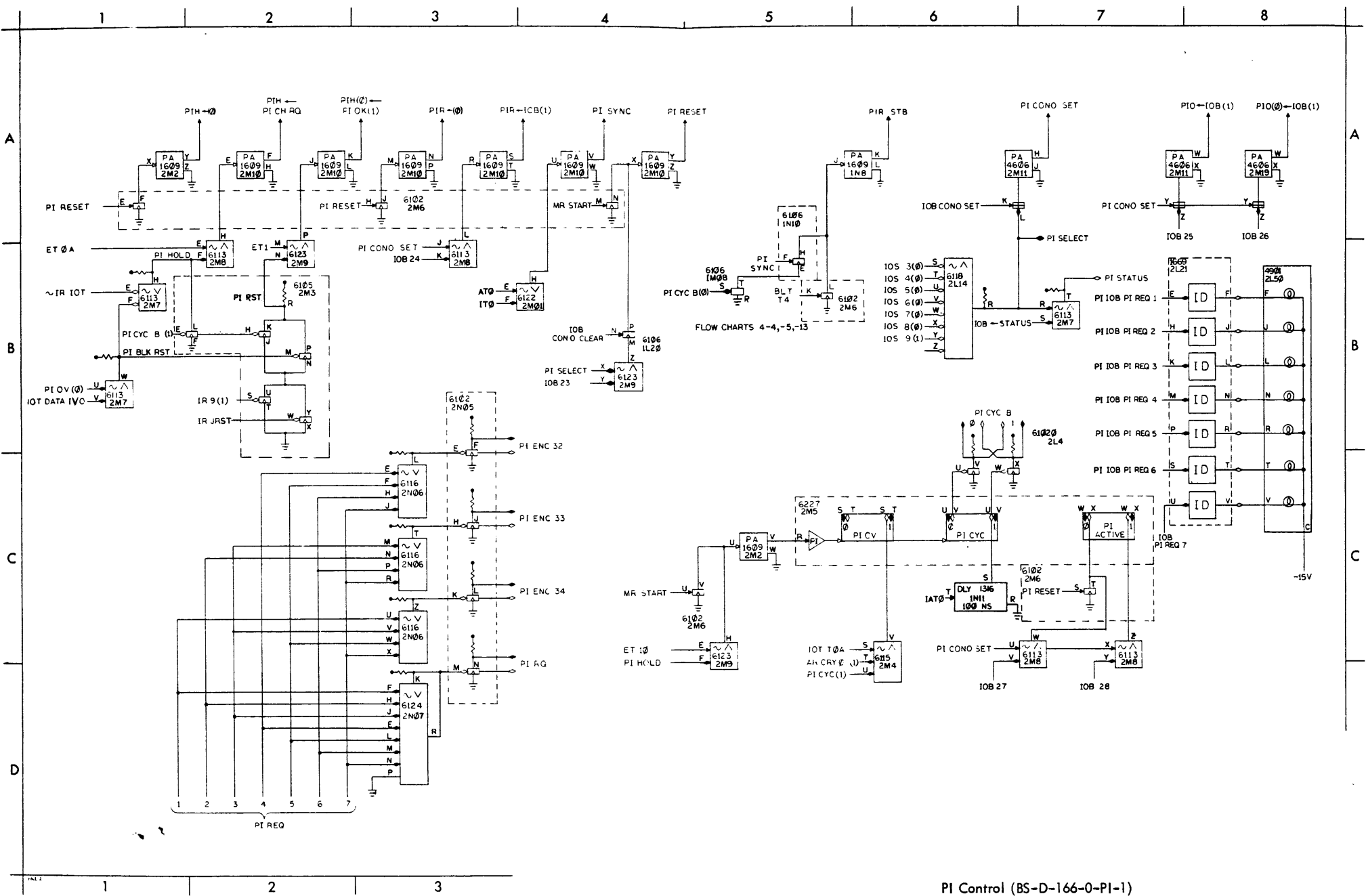
IOT, IOB Control (BS-D-166-0-IOTB)



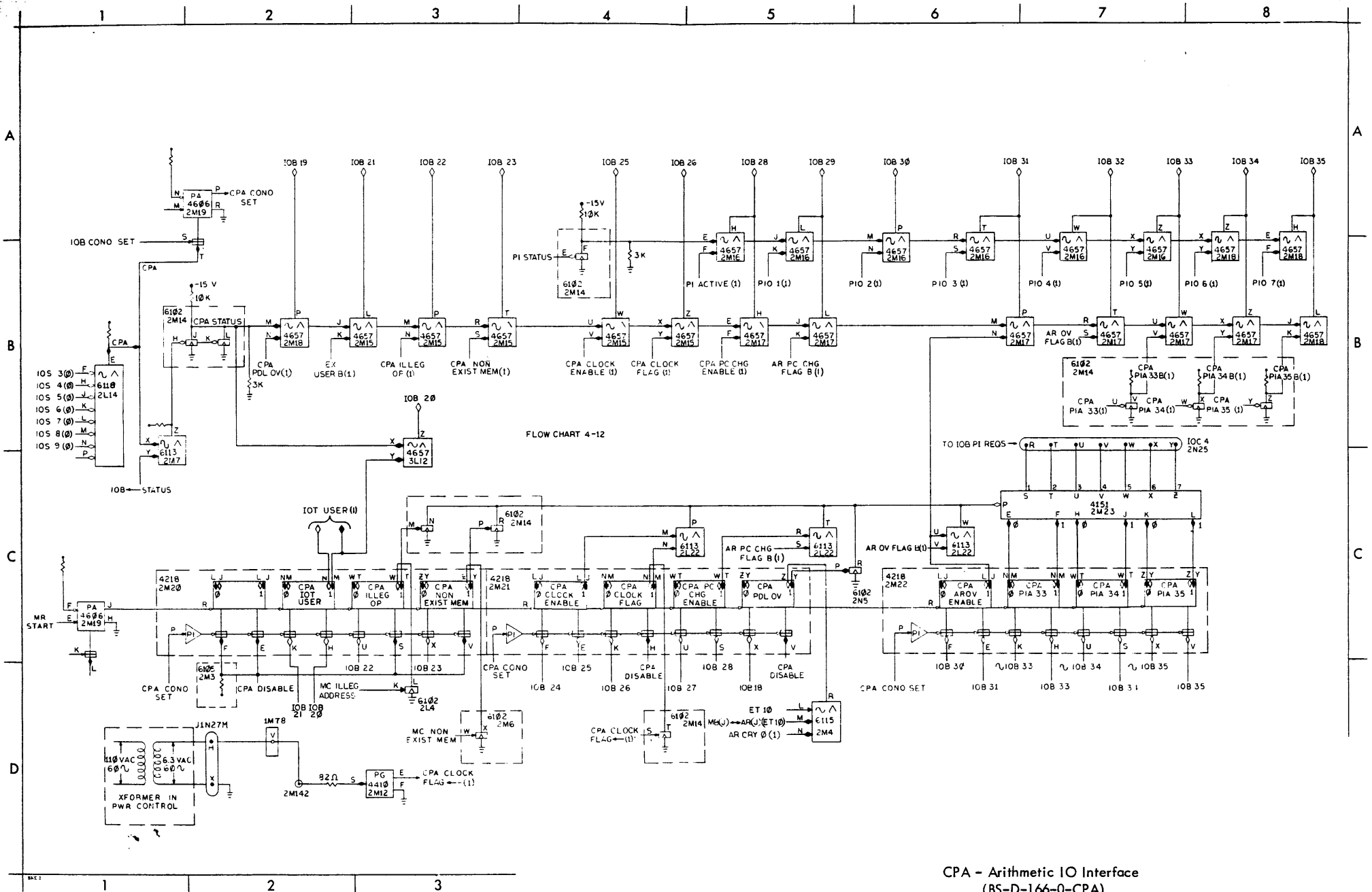
IOB - IO Bus (BS-D-166-0-IOB)

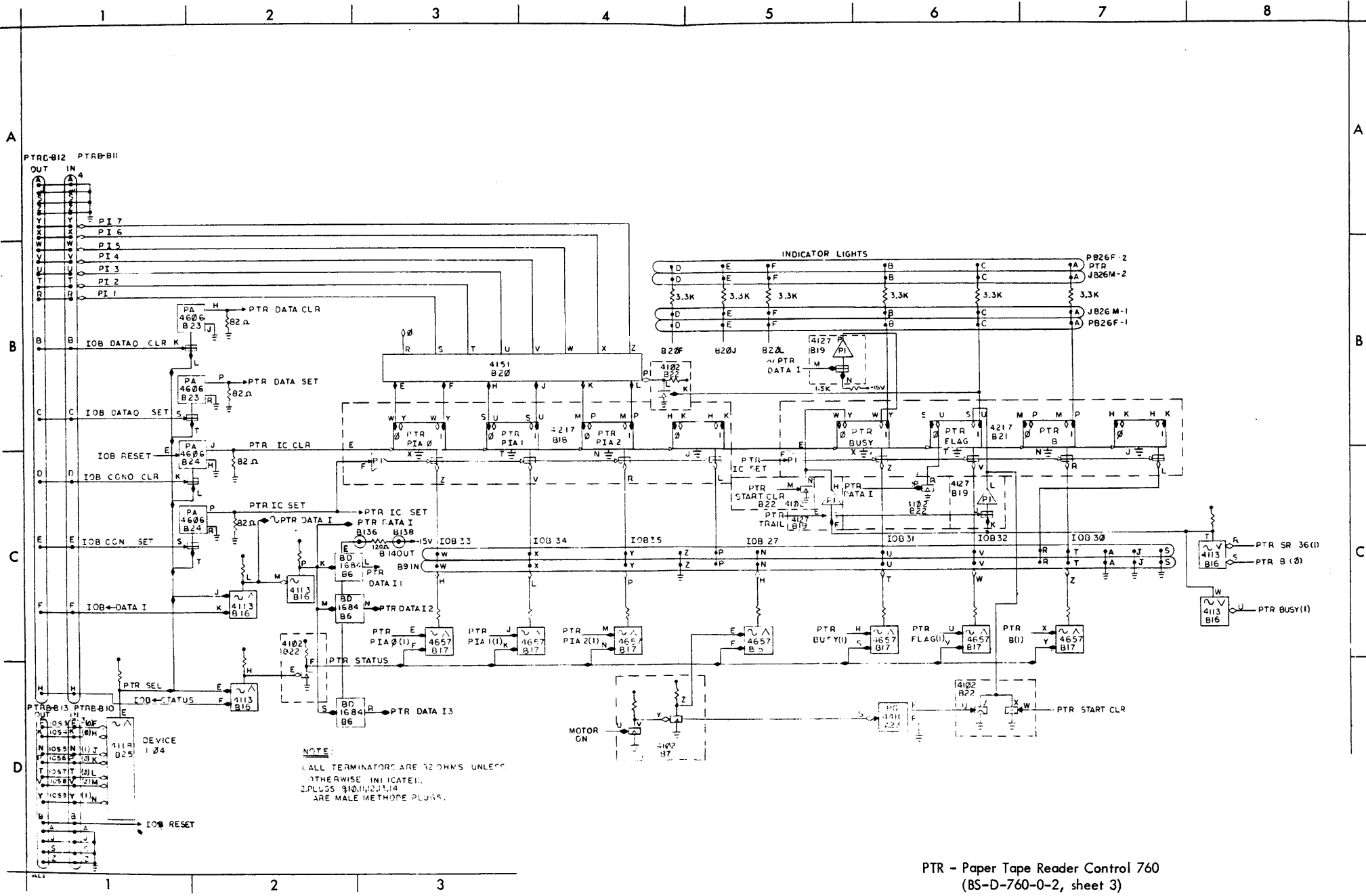


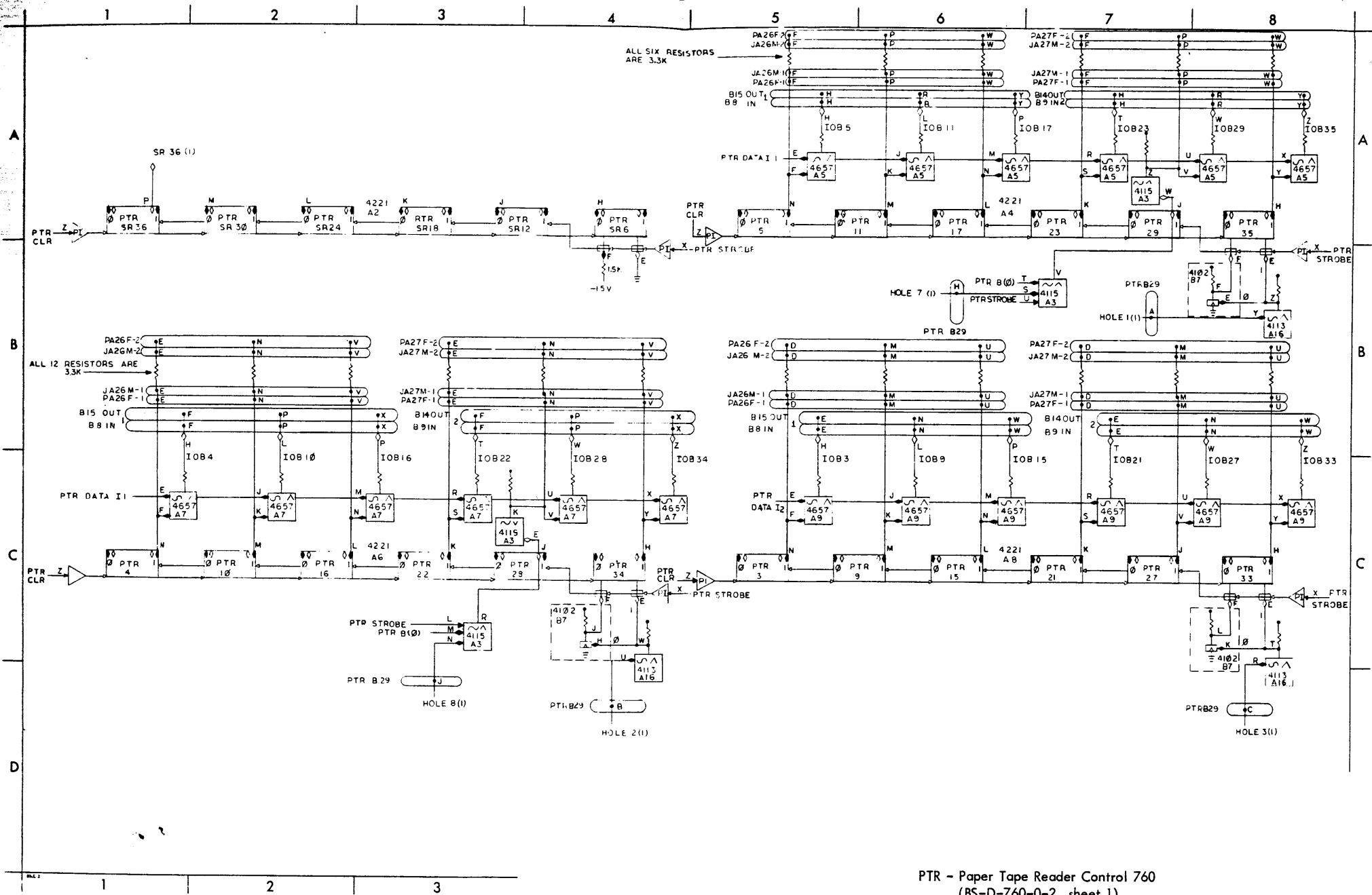
PI - PIH, R, O, 1-7 (BS-D-166-0-PI-2)

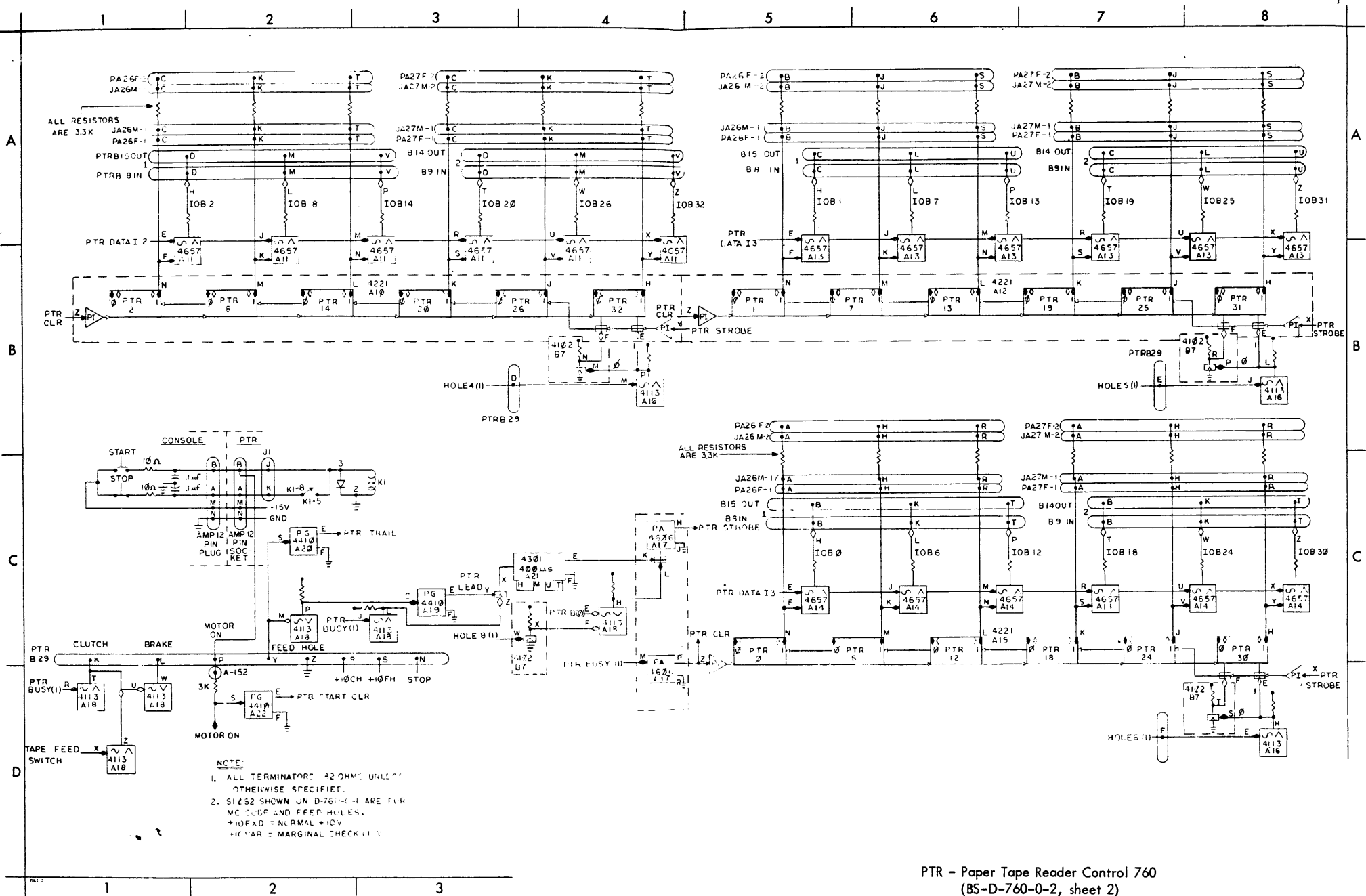


PI Control (BS-D-166-0-PI-1)

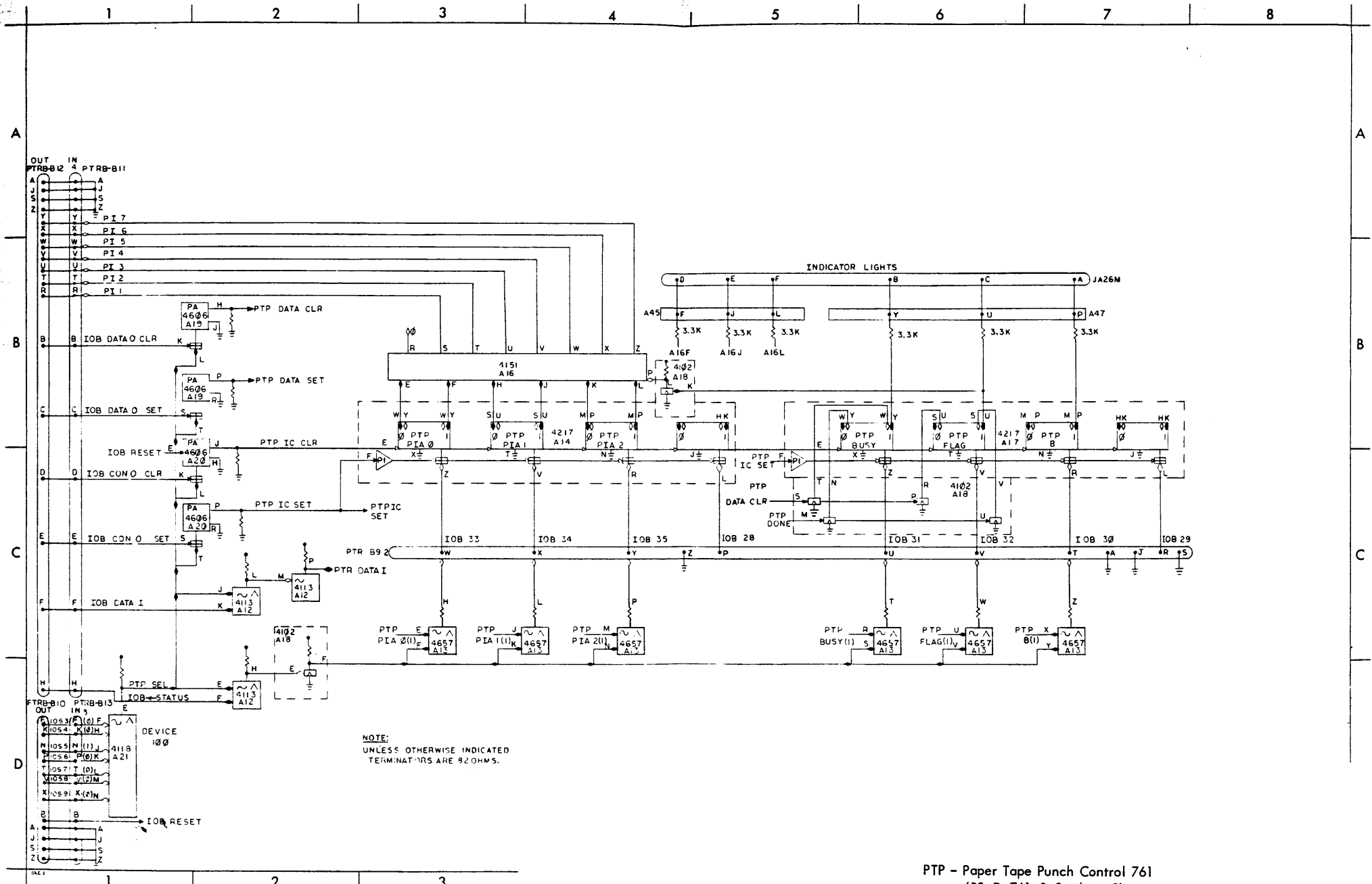




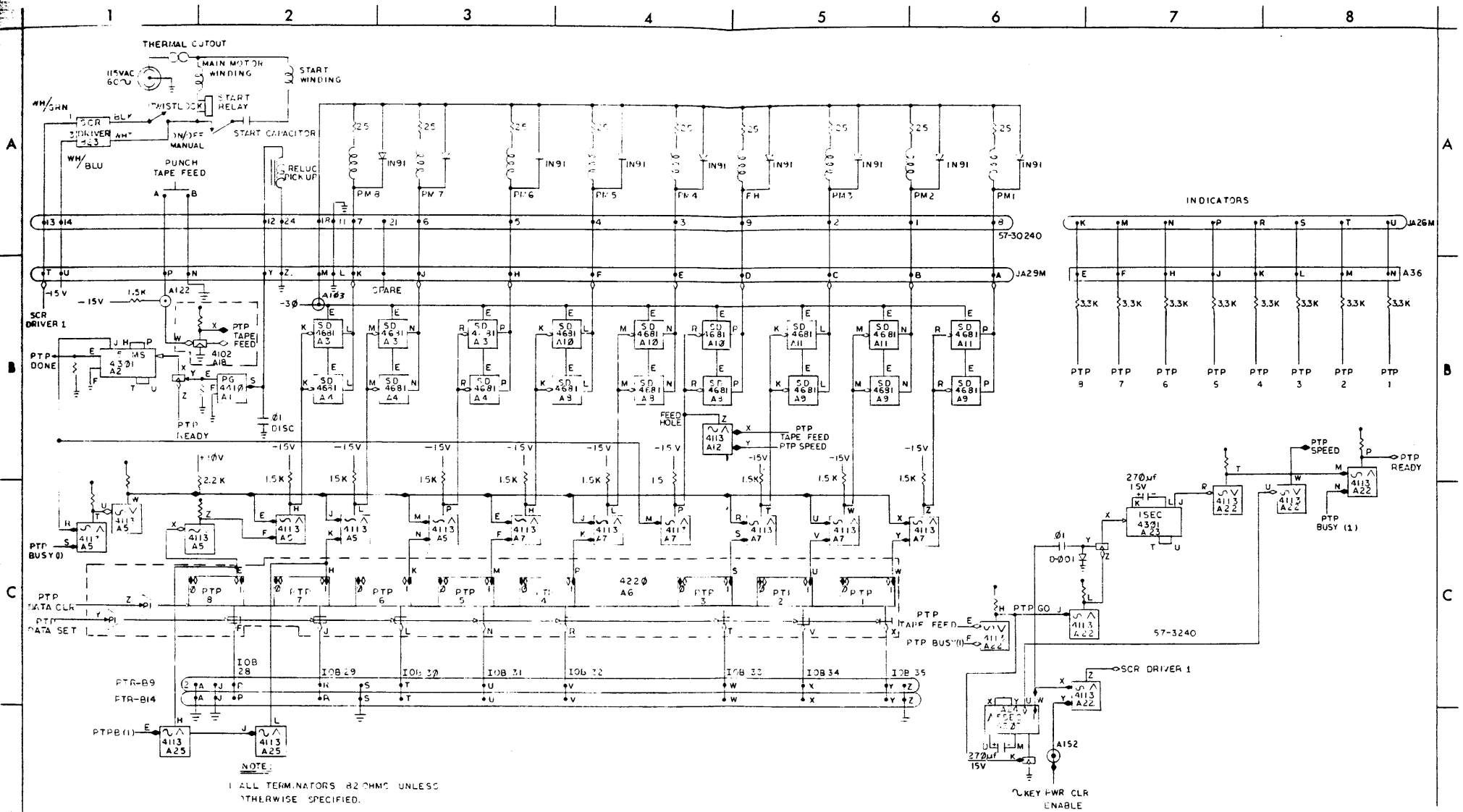




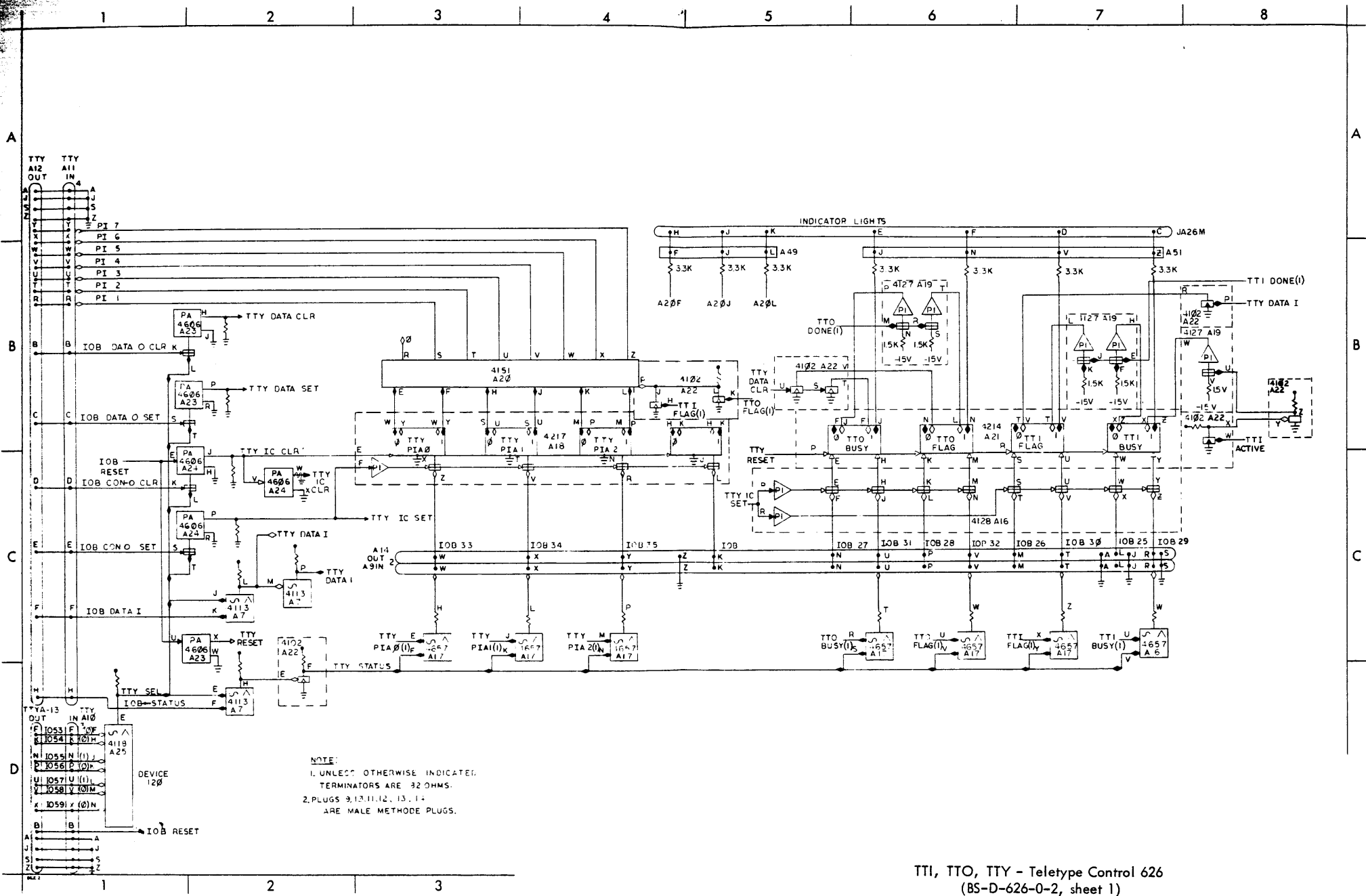
PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 2)

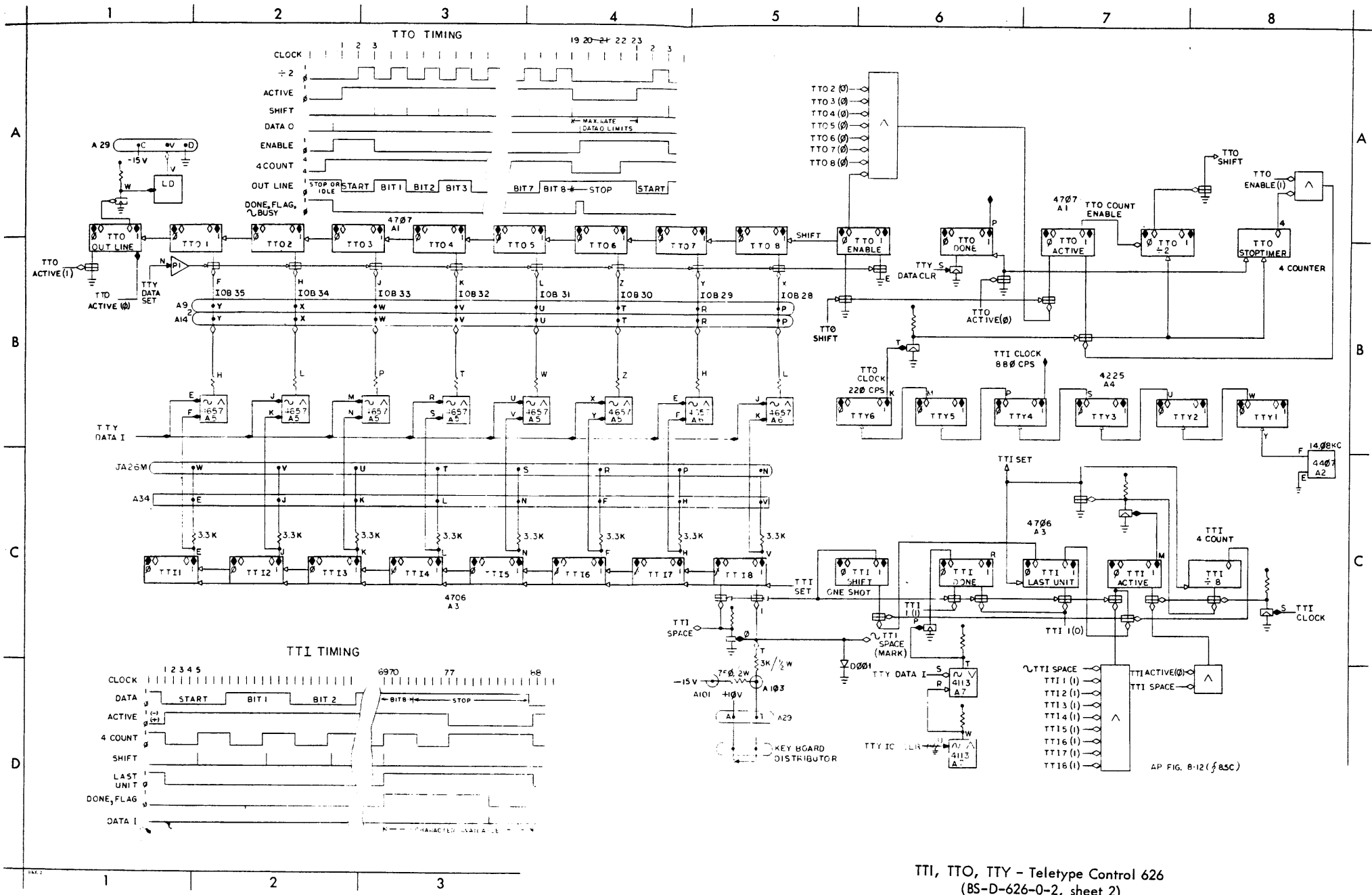


NOTE:
UNLESS OTHERWISE INDICATED
TERMINATORS ARE 920HMS.

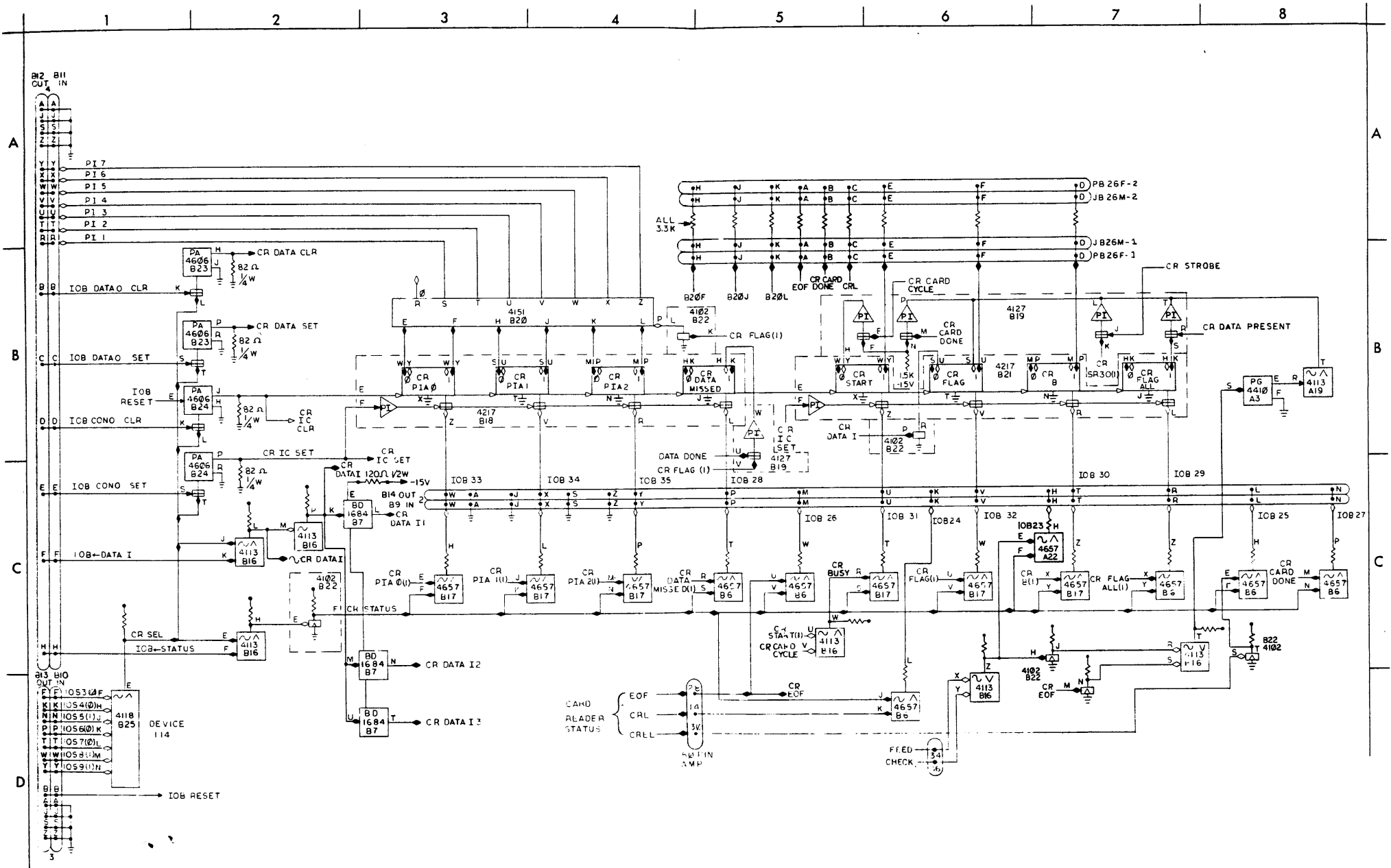


PTP - Paper Tape Punch Control 761
(85-D-761-0-2, sheet 1)

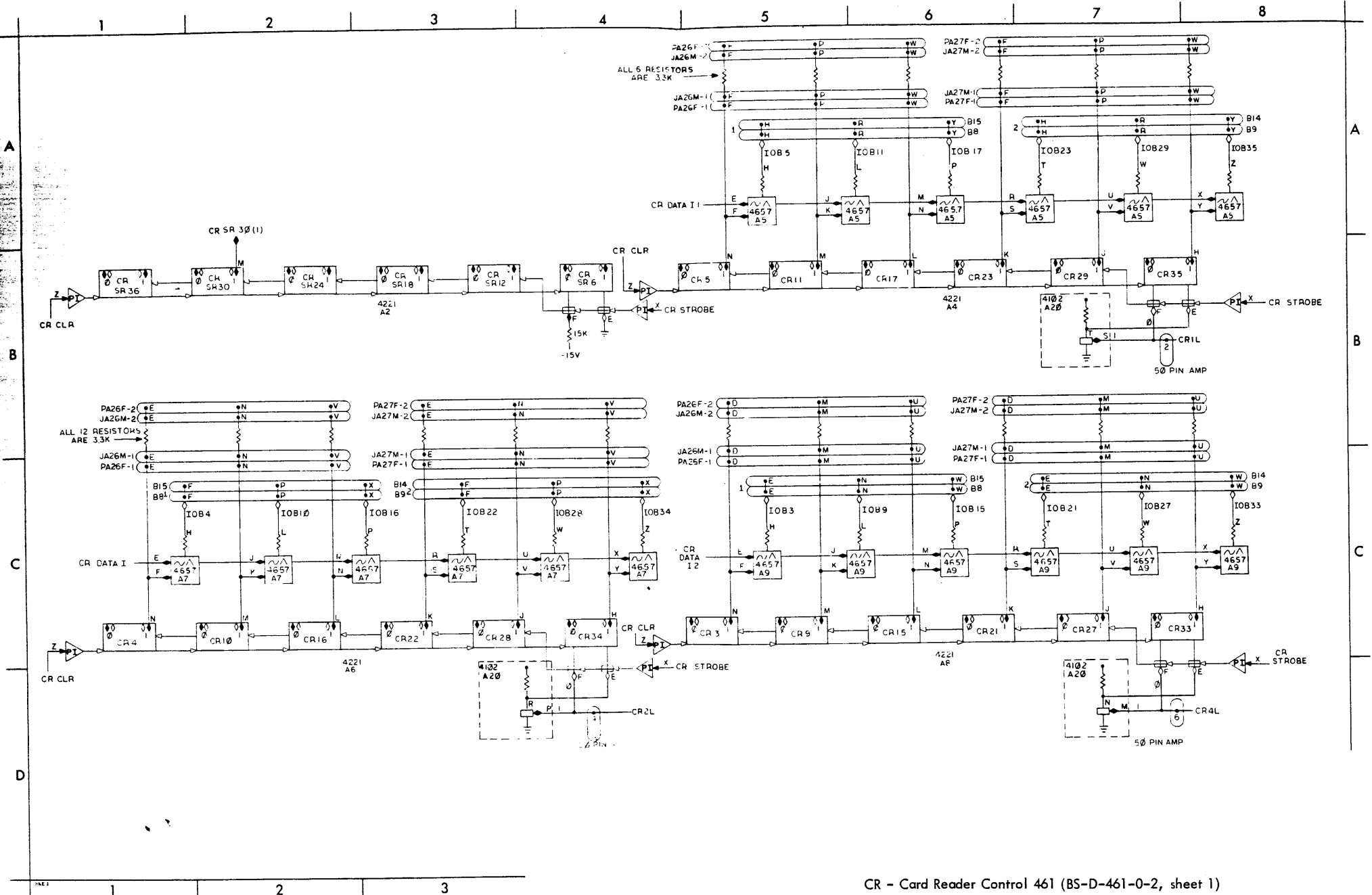




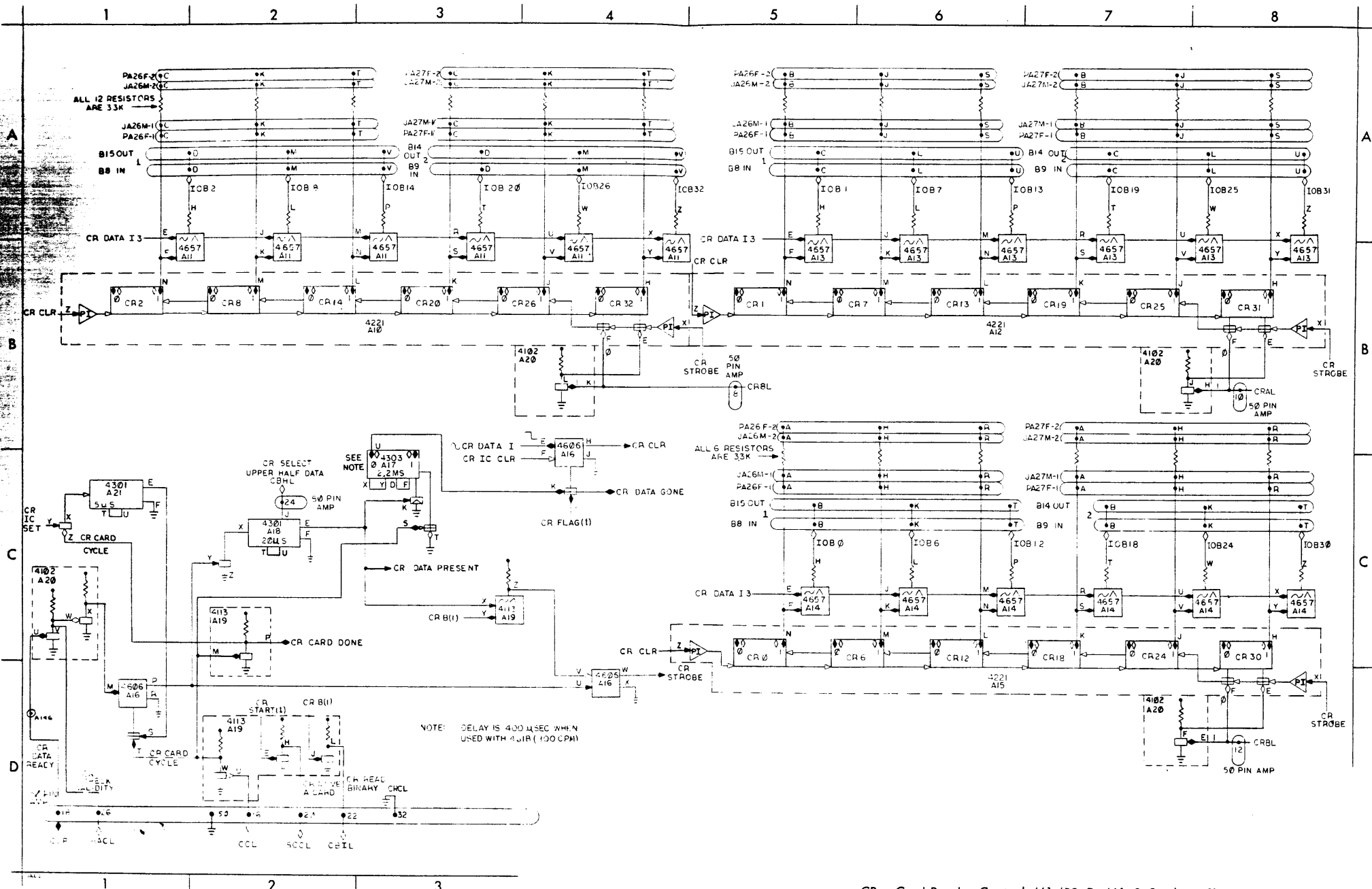
TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 2)



CR - Card Reader Control 461 (BS-D-461-0-2, sheet 3)



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



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