CAL DATA 1/35
EMULATE BOARD
(P/N C81080210)

DRAWING PACKAGE
C21518021-X3

california data processors

INTRODUCTION

This volume contains engineering drawings for the Cal Data 1/35 Emulate Board, part number C81080210.

Table 1 lists the engineering drawings contained in this document. Material parts lists are included as part of each assembly drawing.

Components are identified, where applicable, by both the suggested manufacturer's part number and the Cal Data part number. In general, commercially available parts conform to specifications published by the manufacturer; however, Cal Data component specifications dictate the performance of all component parts used.

Schematics use industry standard designations where applicable (e.g., 7401, 2N5769). These designations are for convenience only and do not necessarily imply that these specific parts are used. The parts lists contain the accurate identification of individual parts.

Table 1. Drawings in this Document

NUMBER	TITLE
C21518022	Theory of Operation, Cal Data 1/35 Emulate Board
C41900101	Emulation Board Set, Cal Data Basic 1/35
C81080210	Board Assembly, Cal Data 1/35 Emulate
C21080210	Board Schematic, Cal Data 1/35 Emulate
C41200101	Firmware Listing, Cal Data 1/35 Emulation

2019 south ritchey street · santa ana, california 92705 · (714) 558-8211

CAL DATA 1/35
EMULATE BOARD
(P/N C81080210)

DRAWING PACKAGE
C21518021-X3

DOCUMENT C21518021
Revision X3
October 1974

california data processors

$20.00

ENGINEERING SPECIFICATION

CAL DATA BASIC 1/35

EMULATION BOARD SET

ORIGINATOR				
APPROVAL				

california data processors	TITLE: CAL DATA BASIC 1/35 EMULATION BOARD SET		
	SIZE: A	DOCUMENT NUMBER: C41900101	REV.: 01
	DS	SHEET 1 OF 7	PAGE

Form 3- 190

TABLE OF CONTENTS

1.0 SCOPE

This Specification defines the Basic Cal Data 1/35 Emulation Board Set.

2.0 APPLICABLE DOCUMENTS

In addition to the Cal Data documents referenced herein, the following documents provide additional information.

Cal Data Document Number	Description
C41000101	1/35 Instruction Set Specification
C41100101	1/35 Source Deck
C41200101	1/35 Listing
C41300101	1/35 Object Paper Tape
C41400101	Reserved
C41500101	Reserved
C41600101	Reserved
C41700101	Reserved
C41800101	Reserved
C41900101	1/35 ROM/PROM Device Set and Boards

3.0 EMULATION BOARD SET

The following Board Set must be configured with ROM Set as shown in Figures 2 and 3 for this unique Emulation.

Cal Data Part No.	Qty.	Description
C81080190	1	Board Assembly, Engine No. 2
C81080210	1	Board Assembly, Emulate

3.1 Board Assembly Configuration and ROM Set Required

The following ROM Set must be installed on Board Assembly C81080190 as shown in Figure 2.

Item No.	PROM Type	ROM Type	Cal Data Part No.	Board Location Code
1	C51333624	C51331624	C43000000	U55
2			C43000001	U47
3			C43000002	U45
4			C43000003	U43
5			C43000004	U35
6			C43000005	U33
7			C43000006	U25
8			C43000007	U23
9			C43000008	U21
10			C43000009	U13
11			C43000010	U11
12			C43000011	U9
13	C51333624	C51331624	C43000012	U56
14			C43000013	U48
15			C43000014	U46
16			C43000015	U44
17			C43000016	U36
18			C43000017	U34
19			C43000018	U26
20			C43000019	U24
21			C43000020	U22
22			C43000021	U14
23			C43000022	U12
24			C43000023	U10

3.2 Board Assembly Configuration and ROM Set Required

The following ROM Set must be installed on Board Assembly C81080210 as shown in Figure 3.

Item No.	PROM Type	ROM Type	Cal Data Part No.	Board Location Code
25	C51333624	C51331624	C43000024	U28
26			C43000025	U37
27			C43000026	U46
28			C43000027	U55
29	C51333256	C51331256	C43000028	U29
30			C43000029	U38
31			C43000030	U47
32			C43000031	U56

3.2.1 Reference Assembly Information

All ROM/PROM devices are marked with the last 5 digits of the part number, example: C43000014 is marked 00014.

ORIGINATOR		california data processors	TITLE: Cal Data Basic 1/35, Emulation Board Set			
APPROVAL			SIZE: A	DOCUMENT NUMBER: C41900101	REV: 01	
			DS	SHEET 4 OF 7	PAGE	

3.3 Board Set Identification

Both boards used in this Emulation Set must be identified as shown in Figure 1.

Identify this Board Set on the component side of both boards with this specification number. Lettering to be .12 high, black Epoxy paint, vertical Gothic and located approximately as shown.

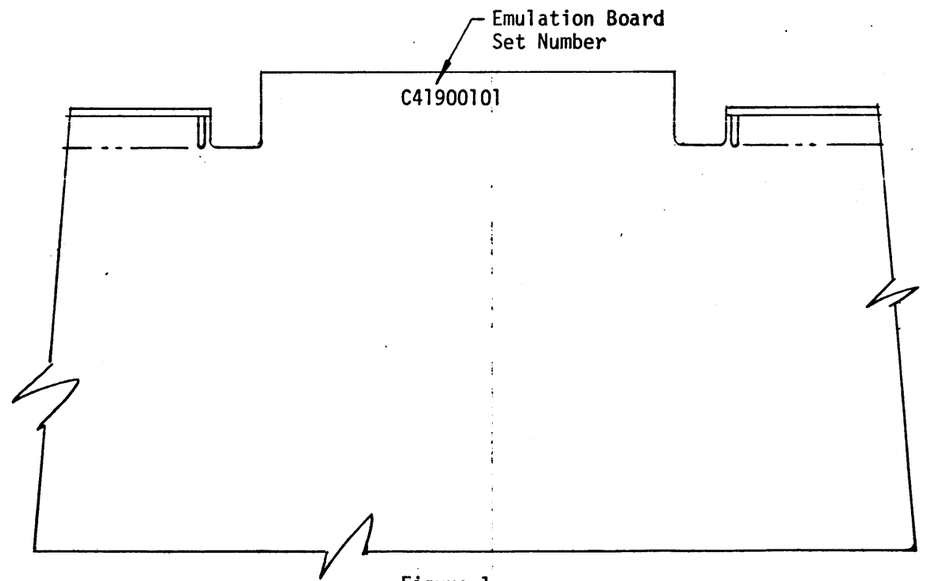


Figure 1

ORIGINATOR		california data processors	TITLE: Cal Data Basic 1/35, Emulation Board Set			
APPROVAL			SIZE: A	DOCUMENT NUMBER: C41900101	REV: 01	
			DS	SHEET 5 OF 7	PAGE	

SPECIFICATION CONTROL DRAWING

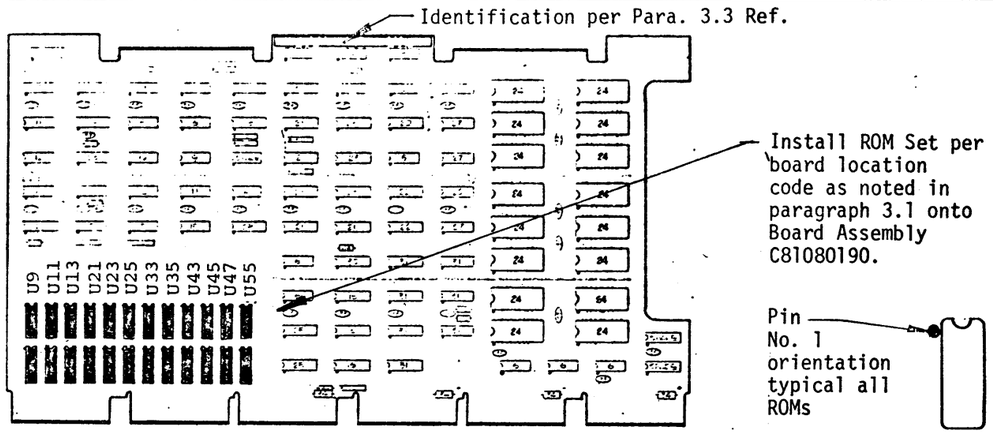


Figure 2

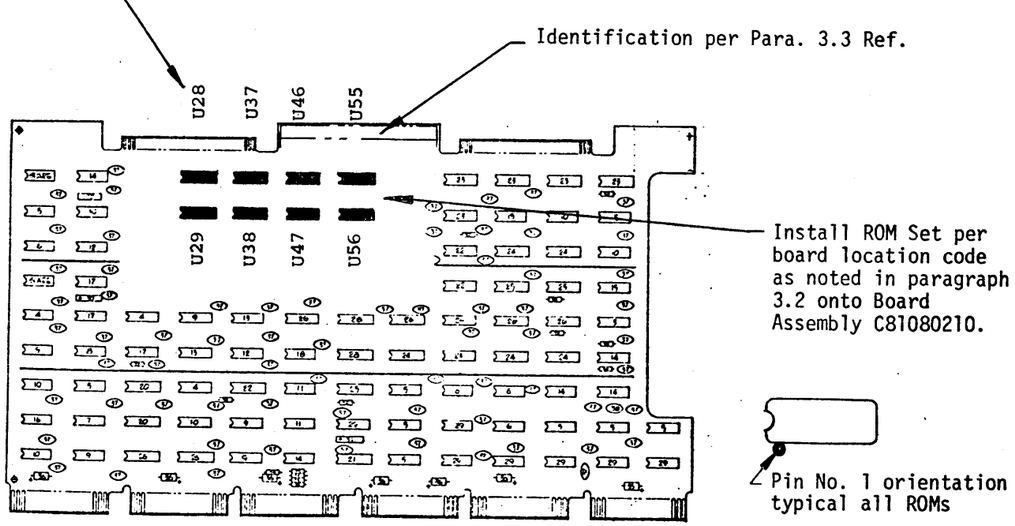


Figure 3

ORIGINATOR		california data processors	TITLE: Cal Data Basic 1/35, Emulation Board Set			
APPROVAL				SIZE	DOCUMENT NUMBER	REV.
				A	C41900101	01
				DS	SHEET 6 OF 7	PAGE

Form 3-173

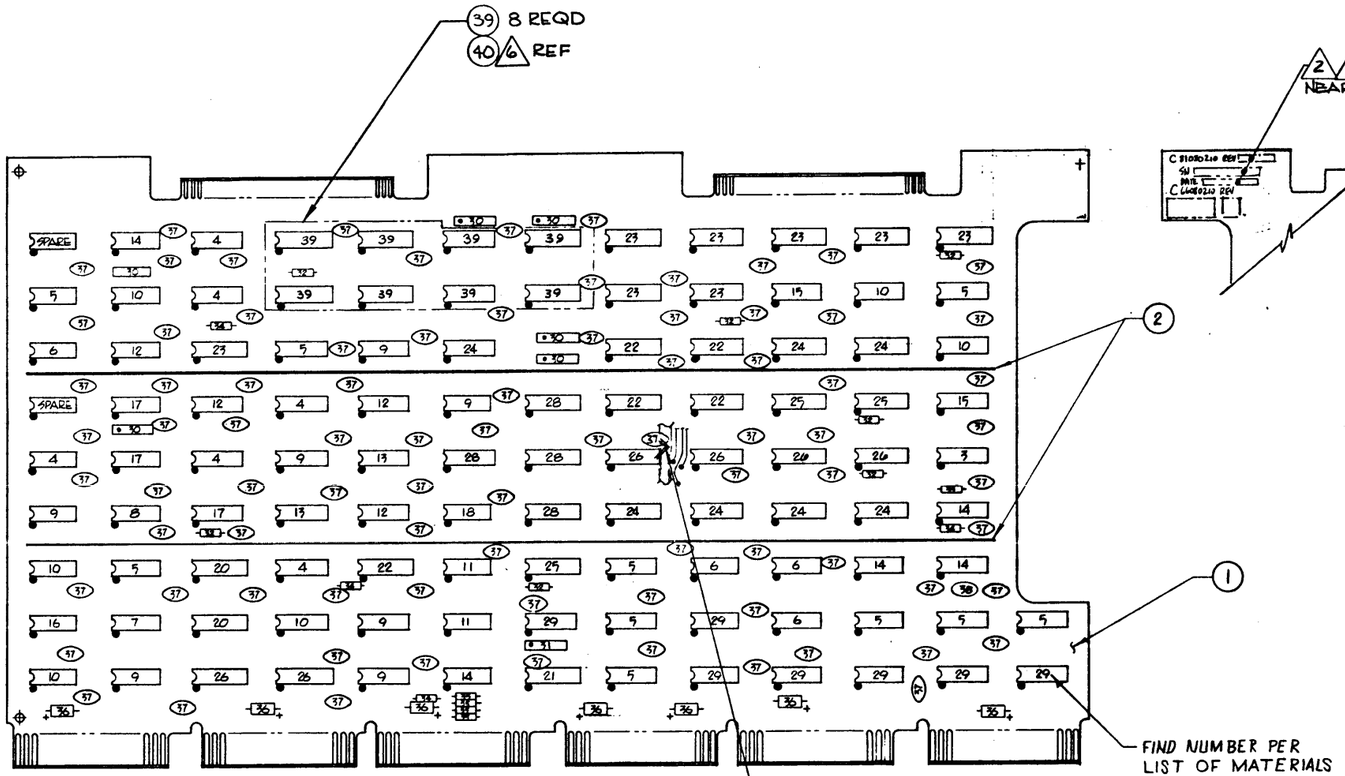
REVISIONS						
REV.	CO		DRFT.	DATE	CHK'D	APPD.
01	~~0~~	Released per D.R.N. DS00130	WB	8-16-79	WB	WB

ORIGINATOR		california data processors	TITLE: Cal Data Basic 1/35, Emulation Board Set		
APPROVAL			SIZE	DOCUMENT NUMBER	REV.
			A	C41900101	01
			DS	SHEET 7 OF 7	PAGE

Form 3-174

RELEASE STATUS			
RELEASE LEVEL		APPROVED	DATE
EXPERIMENTAL			
ENGINEERING	X	[illegible]	3-21-74
PRODUCTION			

REVISIONS					
REV.	ECO.	DESCRIPTION	ORIG.	DATE	APPD.
01	-0-	DCN DS00033	[illegible]	2/1/74	[illegible]
02	DS000056-01	-0-	[illegible]	4/15/74	[illegible]
03	DS000069-01	-0-	RLM	5-8-74	[illegible]
04	DS000063-01	SEE ECO	RLM	5-15-74	[illegible]
05	DS000084-01	SEE ECO	RM	5-22-74	[illegible]
06	DS00114-01	SEE ECO	RM	7-19-74	[illegible]
X	DS00134-01	SEE ECO	CK	8-16-74	[illegible]
07	DS00141-01	SEE ECO	RM	8-22-74	[illegible]
08	DS00162-02	SEE ECO	CK	9-13-74	[illegible]



6. THE R.O.M. SET IS SELECTED AND INSTALLED TO AGREE WITH SALES ORDER MARKETING CODE AS DEFINED BY CONFIGURATION SPECIFICATION C22285009
5. REFERENCE DOCUMENTS: PRODUCT SPECIFICATION: C22410016
   TEST SPECIFICATION: C22413014
4. SERIAL NUMBER TO BE PREFIXED WITH A DIVISION CODE WHICH AGREE WITH PLACE OF MANUFACTURE: DS EQUAL, SANTA ANA - DE EQUAL, EDINA.
3. FOR BOARD SCHEMATIC, SEE C21080210
2. MARK ASSEMBLY LATEST REV. LETTER TO WHICH IT WAS BUILT. ASSY SERIAL NO. AND EIA FOUR DIGIT DATE CODE. LETTERING TO BE .12 HIGH BLACK EPOXY PAINT, VERTICAL GOTHIC AND LOCATED APPROX. AS SHOWN.
1. MAKE PER CAL DATA SPEC C21100010.

NOTES: UNLESS OTHERWISE SPECIFIED

JUMPER CHART	
FROM	TO
U72-16	U73-16
E1	J2-B11

SHOWN ON SHEET 2

SEE DETACHED L/M

ITEM	PART NO.	QTY.	DESCRIPTION

california data processors

TITLE: BOARD ASSEMBLY, C.P.U., EMULATE

ORIGINATOR [illegible] 3-14-74
APPROVAL [illegible] 3-21-74

SIZE D | DOCUMENT NUMBER C81080210 | REV. 08

DS | DO NOT SCALE PRINT | SHEET 1 OF 4

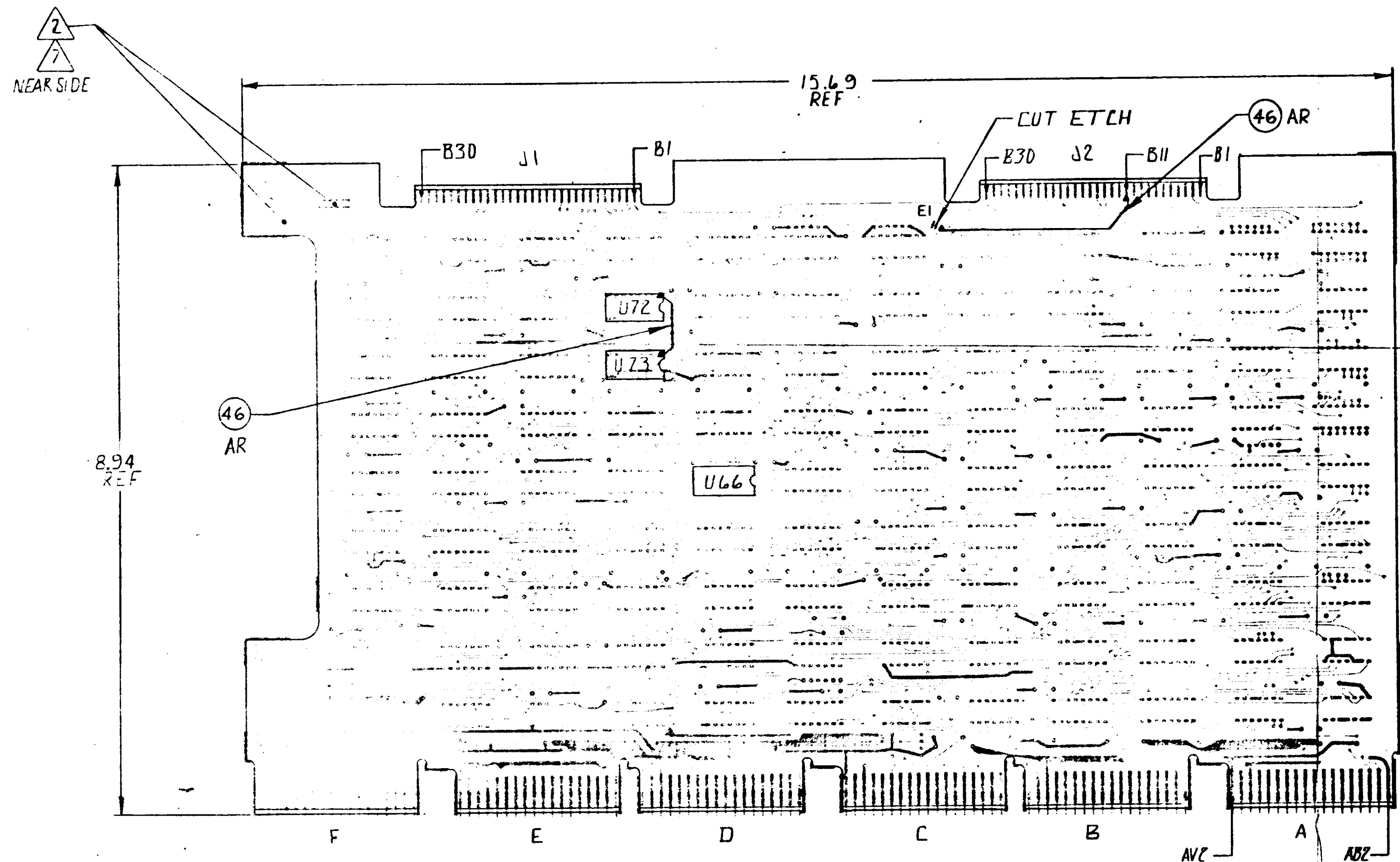
SCALE 1/1

3 PLACE ±.010 | 2 PLACE ±.02 | ANGLES ±0° 30'

UNLESS OTHERWISE SPECIFIED FRACTIONS ± DECIMALS ± ANGLES ±

FORM NO. 3—169

REV.	ECO.	DESCRIPTION	ORIG.	DATE	APPD.
		SEE SHEET 1			



SOLDER SIDE SHOWN

california data processors	TITLE: PRINTED WIRING BOARD C.P.U. EMULAED
ORIGINATOR	
APPROVAL	SIZE: D — DOCUMENT NUMBER: C81C[illegible]210 — REV.: 08
	DS — DO NOT SCALE PRINT — SCALE 1 / 1 — SHEET [illegible] 4 — UNLESS OTHERWISE SPECIFIED: FRACTIONS ± DECIMALS ± ANGLES ±

FORM NO. 3—175

Type 2

LIST OF MATERIALS

ITEM NO.	QTY. REQD.	REFERENCE DESIGNATION	PART NUMBER	DESCRIPTION	NOTES	VENDOR	VENDOR PART NUMBER
1	1		C66080210	PRINTED WIRING BD. EMULATE			
2	2		C67680025	BUS BAR			
3	1	U98	C51323000	I.C. QUAD 2 NAND		TEX. INST.	SN74H00N-00
4	6	U5, 19, U20, 23, U31, 34	C51324000	QUAD 2 NAND			SN74S00N-00
5	10	U2, 16, U30, 68, U69, 70, U93, 95, U101, 102	C51324004	HEX INV.			SN74S04N-00
6	4	U3, 77, U84, 85	C51323005	HEX INV. W/OC			SN74H05N-00
7	1	U17	C51322008	QUAD 2 AND			SN7408N-00
8	1	U15	C51323010	TRIPLE 3 NAND			SN74H10N-00
9	7	U6, 18, U32, 39, U44, 45, U49	C51324010	TRIPLE 3 NAND			SN74S10N-00
10	6	U7, 9, 11, U35, 85, U96	C51324011	TRIPLE 3 AND			SN74S11N-00
11	2	U52, 53	C51322020	DUAL 4 NAND			SN7420N-00
12	4	U12, 22, U40, 42	C51324020	DUAL 4 NAND			SN74S20N-00
13	2	U33, 41	C51323030	8-IN-NAND			SN74H30N-00
14	5	U10, 54, U92, 99, U100	C51322038	QUAD 2 NAND W/OC			SN7438N-00
15	2	U79, 97	C51323051	DUAL AND-OR-INVERT			SN74H51N-00
16	1	U8	C51324065	AND-OR-INV. OC			SN74S65N-00
17	3	U13, 14, U24	C51322074	DUAL-D FLIP-FLOP			SN7474N-00
18	1	U51	C51324086	QUAD 2-IN EXCL-OR			SN74S86N-00
19							
20	2	U25, 26	C51324112	DUAL JK FLIP-FLOP			SN74S112N-00
21	1	U61	C51324133	13-IN-NAND			SN74S133N-00
22	5	U43, 64, U65, 73, U74	C51322151	DATA SEL MUX			SN74151N-00
23	8	U21, 62, U63, 71, U72, 78, U86, 94	C51324151	DATA SEL MUX			SN74S151N-00
24	7	U48, 67, U76, 80, U83, 88, U91	C51324157	QUAD 2 TO 1 DATA SEL MUX			SN74S157N-00
25	3	U60, 81, U89	C51322175	QUAD D FLIP-FLOP			SN74175N-00
26	6	U27, 36, U66, 75, U82, 90	C51324175	I.C. QUAD D FLIP-FLOP		TEX. INST.	SN74S175N-00
27							

FORM NO. 3—172

ORIGINATOR	R McLendon	July 23, 74
APPROVAL	Bowerman	7-24-74
	Kendall	7-24-74
	Wellington	7-25-74
		7-25-74
		7-25-74

california data processors

TITLE: BOARD ASSEMBLY - EMULATE

SIZE	DOCUMENT NUMBER	REV.
B	C81080210	03
DS	SHEET 3 OF 4	PAGE

LIST OF MATERIALS

ITEM NO.	QTY. REQD.	REFERENCE DESIGNATION	PART NUMBER	DESCRIPTION	NOTES	VENDOR	VENDOR PART NUMBER
28	4	U50, 57, U58, 59	C51324257	I.C. QUAD 2 TO 1 DATA SEL MUX		TEX. INST.	SN74S257N-00
29	7	Z1 THRU Z7	C58111010	HYBRID, HEX LOGIC			58111000
30	6	RM1 - RM6	C54811050	RESISTOR NETWORK 560/620			54811050
31	1	RM7	C54811020	RESISTOR NETWORK 180/390			54811020
32	8	R2, 3, 10 THRU 14,16	C54400285	RESISTOR 1K, 1/4 W, ±5%			RC07GF102J
33	1	R15	C54400275	RESISTOR 390Ω, 1/4 W, ±5%			RC07GF391J
34	3	R1, 5, 4	C54400277	RESISTOR 470Ω, 1/4 W, ±5%			RC07GF471J
35	4	R6, 7, 8, R9	C54400267	RESISTOR 180Ω, 1/4 W, ±5%			RC07GF181J
36	7	C1 - C7	C52326001	CAPACITOR 4.7µf, 20 V, +20%		COMP. INC.	CCD-020-475
37	84	C8 - C91	C52252500	CAPACITOR .01µf, 100 V, 20%		ERIE	805-024-Z5U
38	1	C92	C52213385	CAPACITOR 470pf,		CDE	CD15FD471J03
39	8		C56148000	SOCKET, 16 PIN, LOW PROFILE		SAE	CSA3100-16B
40	AR			ROM SET	6		
41							
42							
43							
44							
45							
46	AR	-	C37002750	WIRE, 24 AWG, SOLID COND. GRN		AM ENKA	T502 (1/24)

FORM NO. 3—172

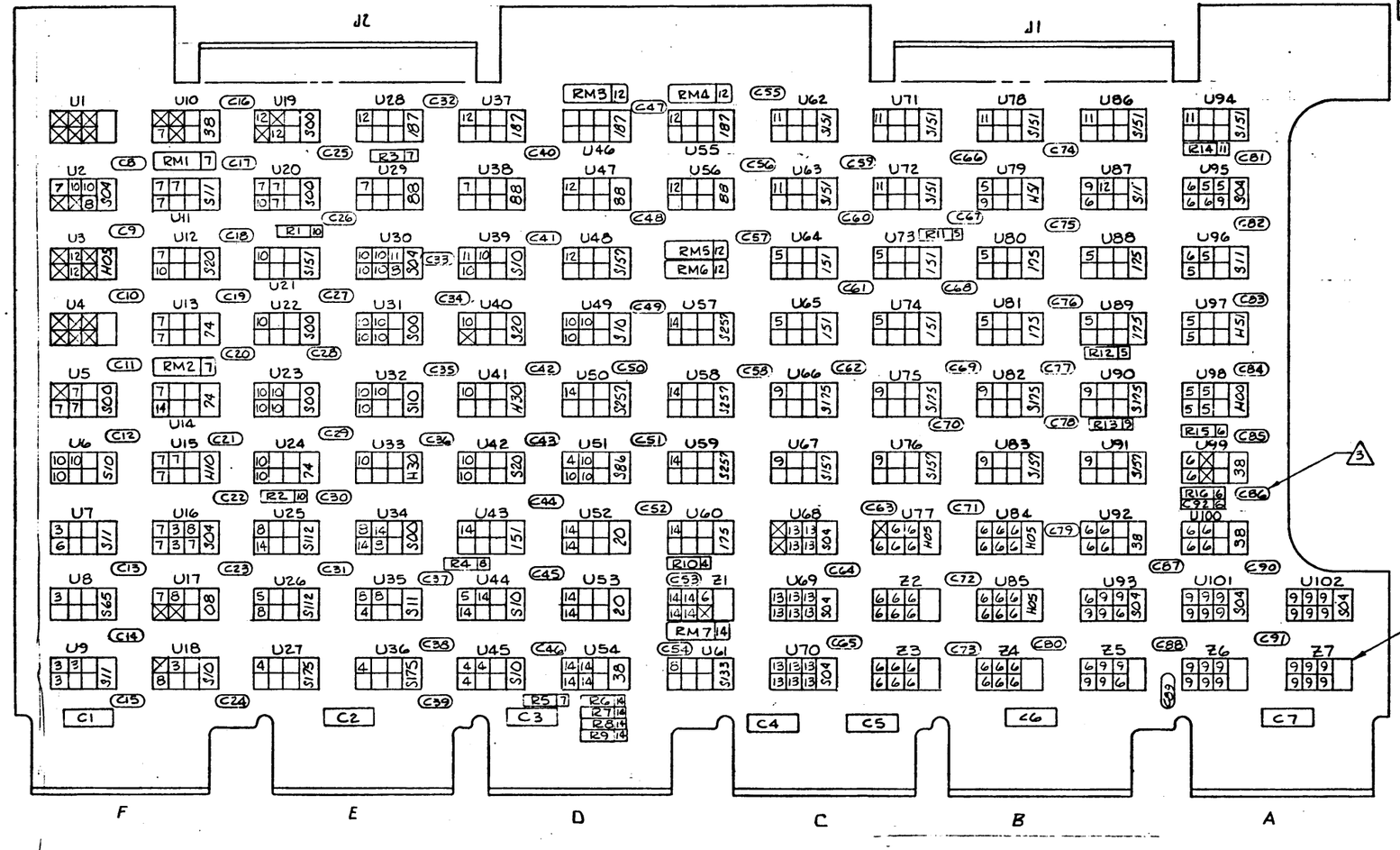
ORIGINATOR C Kennedy	9-17-79	california data processors	TITLE: BOARD ASSEMBLY - EMULATE
APPROVAL			SIZE: B; DOCUMENT NUMBER: C81080210; REV.: 08
			DS; SHEET 4 OF 4; PAGE

RELEASE STATUS		
RELEASE LEVEL	DES SERVICES	DATE
EXPERIMENTAL		
ENGINEERING	[signature]	2-20-74
PRODUCTION		

ATTACHED E.O. RECORD

E.O. NUMBER	DATE

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	01	REL PER DRN DS00028	2-20-74	[signature]
	02	REVISED PER ECO. DS00063-01 SHTS EFFECTED 1 THRU 8, 10, 14,	5-14-74	[signature]
	03	SEE ECO DS00162-02	9-13-74	[signature]



REFERENCE DESIGNATIONS

COMPONENT	HIGHEST NUMBER	NUMBERS NOT USED
CAPACITOR	C92	
HYBRID CKT	Z7	
INT CKT	U102	
RESISTOR	R16	
RES MODULE	RM7	

4. Z1 THRU Z7 ARE 58111010
3. ALL CAPACITORS APPEAR ON SHEET 2
2. REVISIONS TO THIS DOCUMENT WILL BE MAINTAINED ON A PER SHEET BASIS. FOR REVISION LOG SEE SHEET 2
1. FUNCTIONS CODED WITH THIS NOTE ARE ASSIGNED ON THE BACKPLANE AND ARE NOT USED ON THIS ASSEMBLY

NOTES: UNLESS OTHERWISE SPECIFIED

PART/ASSY NO. & QTY PER ASSY	NOTE	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION / MATERIAL	SPEC/SOURCE	CODE IDENT NO.	FIND NO.
			PARTS LIST			1

PART/ASSY REV LTR

PART NO.	NEXT ASSY	USED ON
810300		
	APPLICATION	

DO NOT SCALE DRAWING

SCREW THREADS PER HANDBOOK H-28
COUNTERBORE AND SPOTFACE FILLET RADII TO BE .010 MAXIMUM
REMOVE ALL BURRS AND BREAK SHARP EDGES EQUIVALENT TO .010R
ROUGHNESS OF MACHINED SURFACES 125 PER USAS B46.1
STANDARD HOLE TOLERANCE PER ANSI B32
TOLERANCES ON XX ± .02 XXX ± .010 ANGLES ± 2.0° BY
INTERPRET DIMENSIONS AND TOLERANCES PER USAS Y14.5
DIMENSIONS ARE IN INCHES AND APPLY AFTER HEAT TREAT AND FINISH
UNLESS OTHERWISE SPECIFIED

CONTRACT NO.		
DRAWN	[signature]	12/13/73
CHECK	[signature]	12-13-73
ENGR	[signature]	12-19-73
PROJ. ENGR	[signature]	2-26-74
TECH. DIR.	[signature]	2/25/74
PROD. MGT.	[signature]	2-27-74
PRODUCTION		
Q.A.	[signature]	2-25-74

california data processors
2019 south ritchey street · santa ana, california 92705 · (714) 558-8211

DWG TITLE
BOARD SCHEMATIC
C.P.U., EMULATE

SIZE	CODE IDENT NO.	DWG NO.
D	25	C21080210
SCALE: NONE		REV 03 SHEET 1 OF 14

C21080210 03

SHEET NUMBERS

REVISION LEVEL

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
								01		01	01	01												
02	02	02	02	02	02	02	02		02				02											
03	03	03																						

PAGE NO	1	D	2	PAGE NO
7	[illegible]	A	+5V	
7	[illegible]-H	B	-15V	
	[illegible]		GND	
	[illegible]		[illegible]	
	[illegible]		[illegible]	
			[illegible]	
			[illegible]	
			RESET-L	7
			BUS BG7-IN	
			BUS BG7-OUT	14
			BUS BG6-IN	
			BUS BG6-OUT	14
			BUS BG5-IN	
			BUS BG5-OUT	14
			BUS BG4-IN	
			BUS BG4-OUT	14
			BUS NPG-IN	
			BUS NPG-OUT	

PAGE NO	1	C	2	PAGE NO
13	[illegible]		+5V	
13	[illegible]		-15V	
13	[illegible]		GND	
13	[illegible]		[illegible]-L	13
13	[illegible]		[illegible]	13
13	[illegible]		[illegible]	13
13	[illegible]		[illegible]	13
13			[illegible]	13
13			[illegible]	13
13			[illegible] H	14
14			[illegible] H	14
14			[illegible] H	14
14	[illegible] H		[illegible] H	14
14	[illegible] H		[illegible] H	14
14	[illegible] H		[illegible] H	14
			[illegible] H	14
14	[illegible] H		[illegible] H	14
14	[illegible] H	V	[illegible] H	14

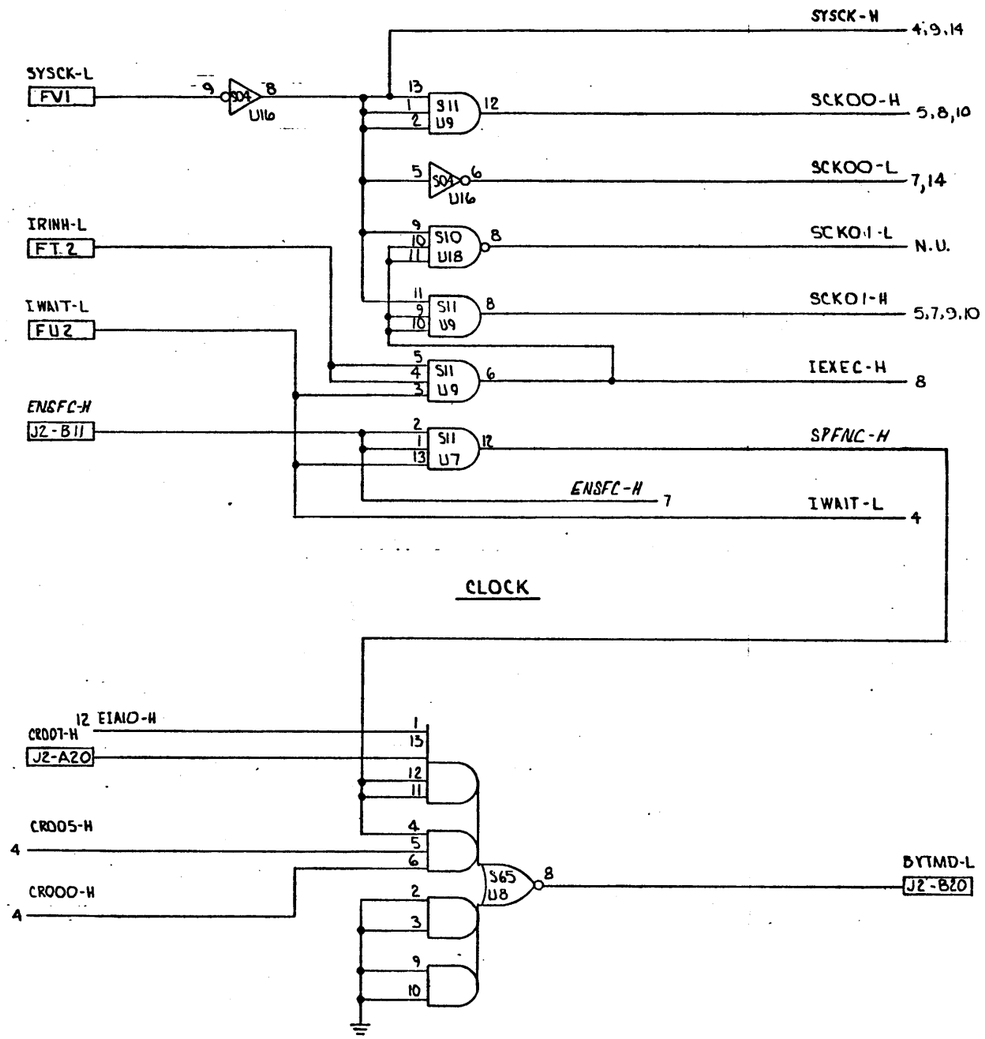
PAGE NO	1	B	2	PAGE NO
	BUS BG6-H			
	BUS BG5-H			
14	BUS BR5-L		GND	
	GND		BUS BR4-L	14
	GND		BUS BG4-H	
	BUS ACLO-L		BUS DCLO-L	
6	BUS A01-L		BUS A00-L	
6	BUS A03-L		BUS A02-L	6
6	BUS A05-L		BUS A04-L	6
6	BUS A07-L		BUS A06-L	6
6	BUS A09-L		BUS A08-L	6
6	BUS A11-L		BUS A10-L	6
6	BUS A13-L		BUS A12-L	6
6	BUS A15-L		BUS A14-L	6
6	BUS A17-L		BUS A16-L	6
	GND		BUS C1-L	6
6	BUS SSYN-L		BUS C0-L	
6	BUS MSYN-L	V	GND	

PAGE NO	1	A	2	PAGE NO
	BUS INIT-L		[illegible]V	
	BUS INTR-L		GND	
9	BUS D00-L		GND	
9	BUS D02-L		BUS D01-L	9
9	BUS D04-L		BUS D03-L	9
9	BUS D06-L		BUS D05-L	9
9	BUS D08-L		BUS D07-L	9
9	BUS D10-L		BUS D09-L	9
9	BUS D12-L		BUS D11-L	9
9	BUS D14-L		BUS D13-L	9
	BUS PA-L		BUS D15-L	9
	GND		BUS PB-L	
	[illegible]		BUS BBSY-L	
	[illegible]		BUS SACK-L	
	GND		BUS NPR-L	
	GND		BUS BR7-L	14
	BUS NPG-H		BUS BR6-L	14
	BUS BG7-H		GND	

D 0?0210

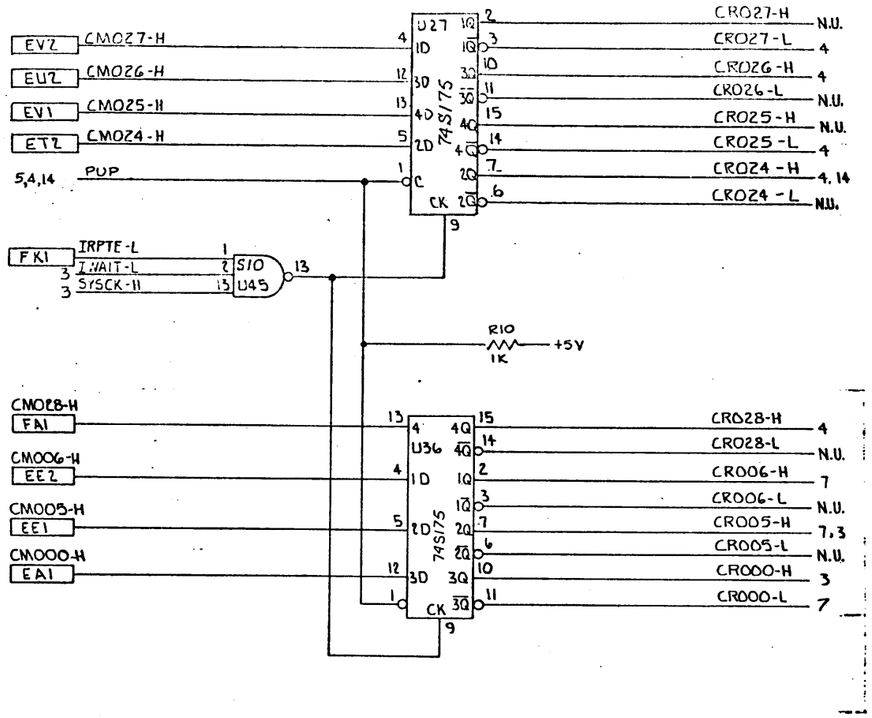
4210080210

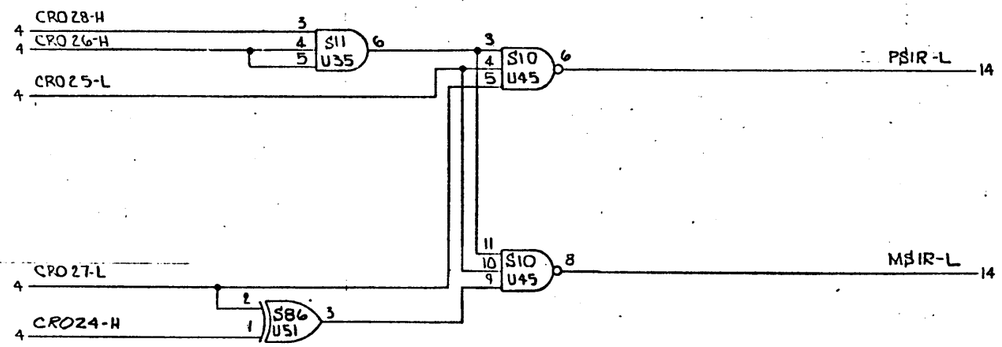
REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		



SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 03	SHEET 3 OF 14

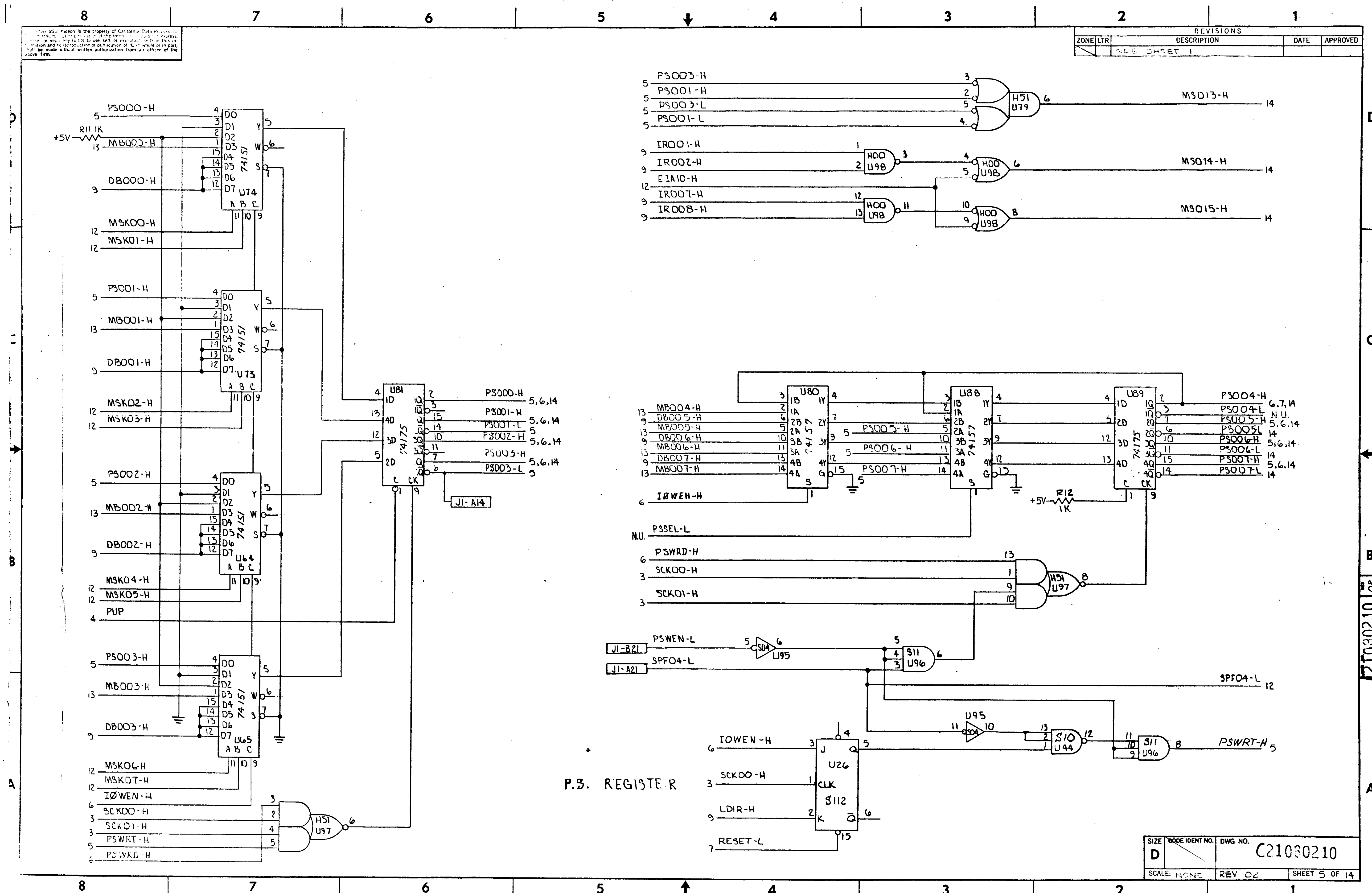
REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		



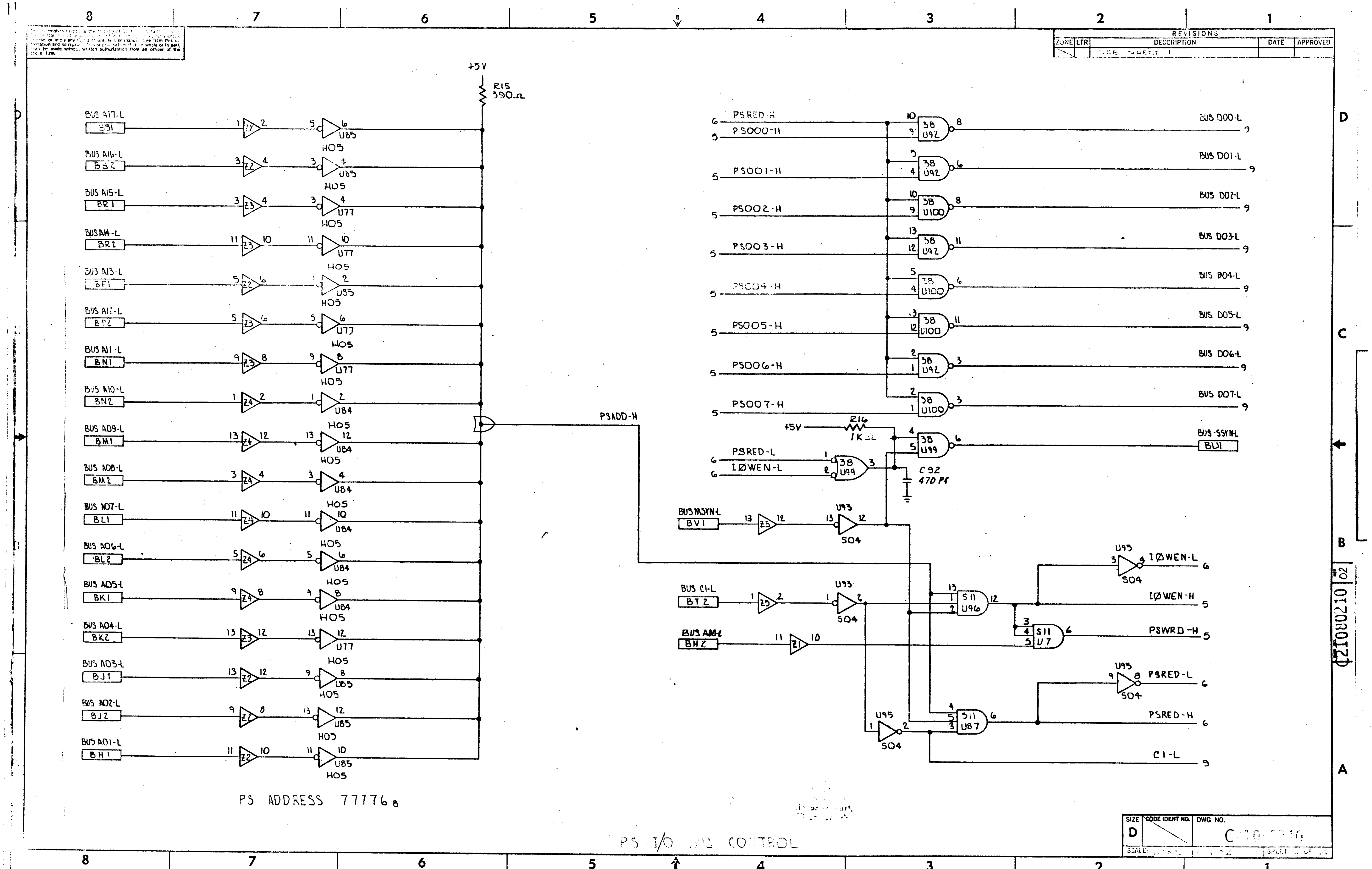


SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 02	SHEET 4 OF 14

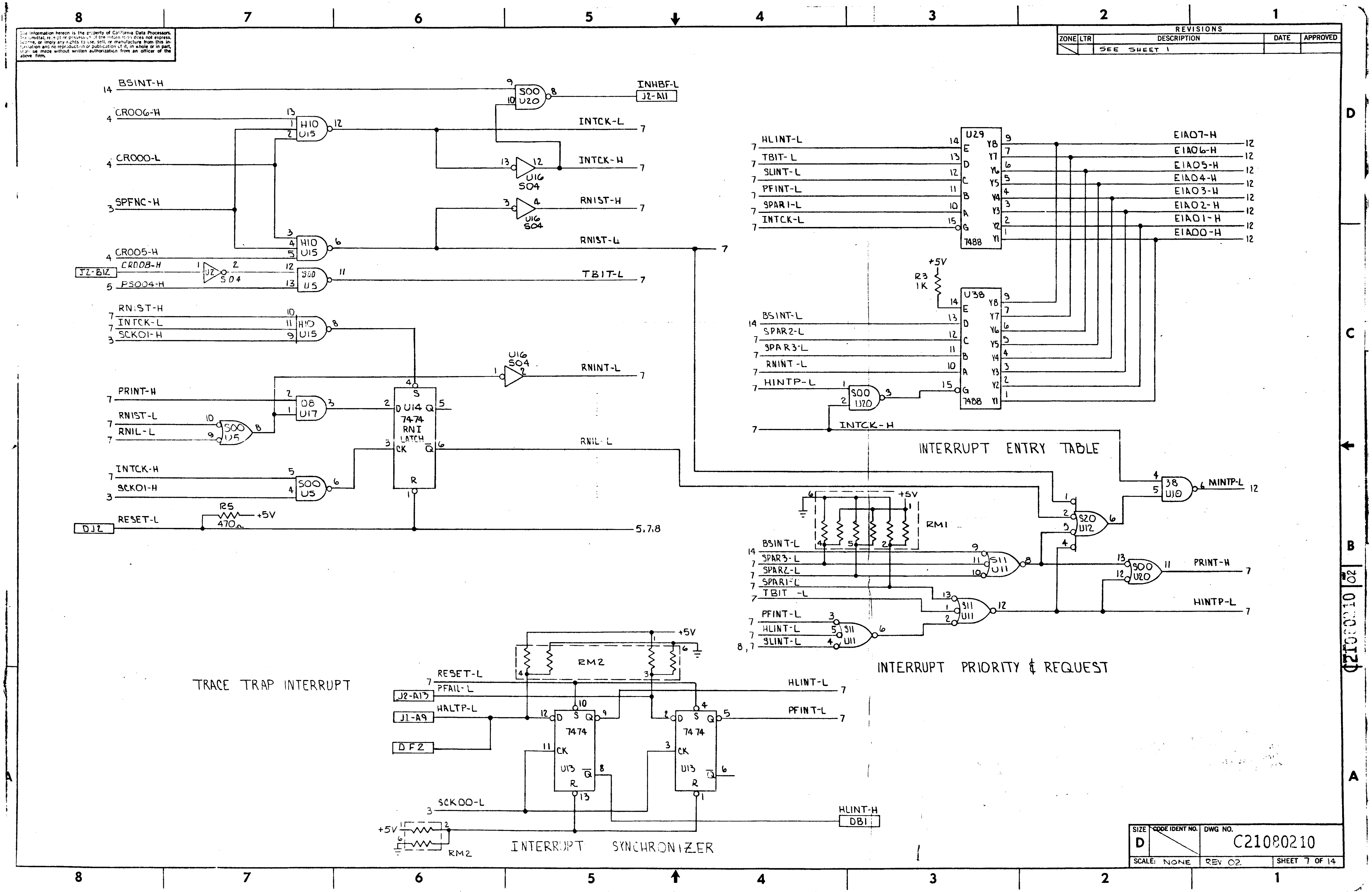
C21080210 02



PS000-H
PS001-H
PS002-H
PS003-H
MB000-H
MB001-H
MB002-H
MB003-H
DB000-H
DB001-H
DB002-H
DB003-H
MSK00-H
MSK01-H
MSK02-H
MSK03-H
MSK04-H
MSK05-H
MSK06-H
MSK07-H
PUP
IØWEN-H
SCK00-H
SCK01-H
PSWRT-H
PSWRD-H
+5V
R11 1K
74151
U74
U73
U64
U65
U81
74175
PS000-H 5,6,14
PS001-H
PS001-L 5,6,14
PS002-H 5
PS003-H 5,6,14
PS003-L 5
J1-A14
H51
U97
PS003-L
PS001-L
U79
MS013-H 14
IR001-H
IR002-H
EIA10-H
IR007-H
IR008-H
H00
U98
MS014-H 14
MS015-H 14
MB004-H
DB005-H
MB005-H
DB006-H
MB006-H
DB007-H
MB007-H
74157
U80
U88
U89
PS005-H
PS006-H
PS007-H
PS004-H 6,7,14
PS004-L N.U.
PS005-H 5,6,14
PS005-L 14
PS006-H 5,6,14
PS006-L 14
PS007-H 5,6,14
PS007-L 14
IØWEH-H
PSSEL-L N.U.
R12 1K
PSWEN-L
J1-B21
SPF04-L
J1-A21
S04
U95
S11
U96
SPF04-L 12
S10
U44
PSWRT-H 5
IOWEN-H
SCK00-H
LDIR-H
RESET-L
U26
S112
CLK
P.S. REGISTER
REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
SEE SHEET 1
SIZE D
CODE IDENT NO.
DWG NO. C21030210
SCALE: NONE
REV 02
SHEET 5 OF 14



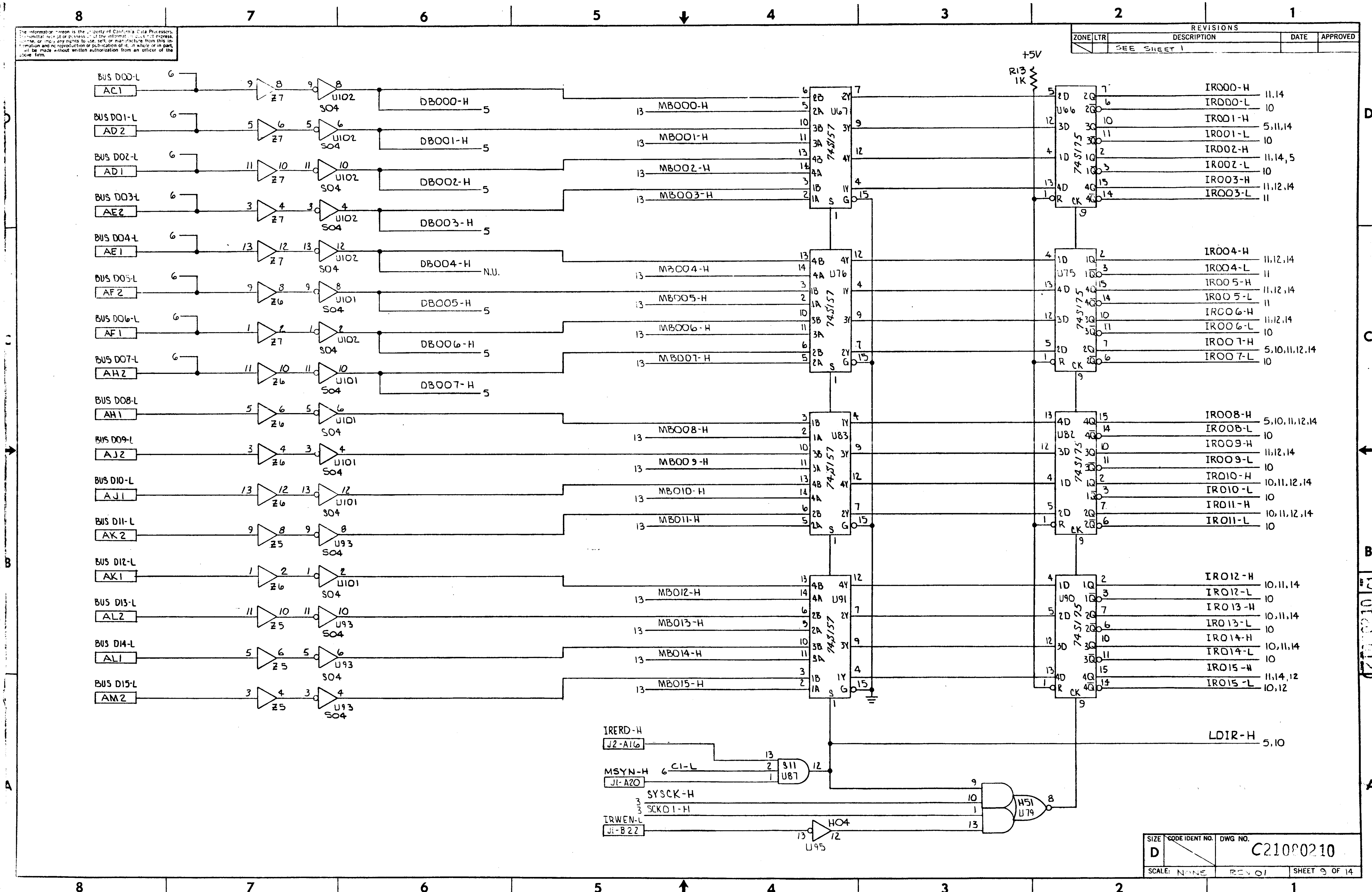
REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
+5V
R15 390Ω
BUS A17-L BS1
BUS A16-L BS2
BUS A15-L BR1
BUS A14-L BR2
BUS A13-L BP1
BUS A12-L BP2
BUS A11-L BN1
BUS A10-L BN2
BUS A09-L BM1
BUS A08-L BM2
BUS A07-L BL1
BUS A06-L BL2
BUS A05-L BK1
BUS A04-L BK2
BUS A03-L BJ1
BUS A02-L BJ2
BUS A01-L BH1
U85 H05
U77 H05
U84 H05
PSADD-H
PSRED-H
PS000-H
PS001-H
PS002-H
PS003-H
PS004-H
PS005-H
PS006-H
PS007-H
38 U92
38 U100
38 U99
BUS D00-L
BUS D01-L
BUS D02-L
BUS D03-L
BUS D04-L
BUS D05-L
BUS D06-L
BUS D07-L
BUS SSYN-L BU1
R16 1KΩ
PSRED-L
IØWEN-L
C92 470 PF
BUS MSYN-L BV1
U93 S04
BUS C1-L BT2
BUS A00-L BH2
S11 U96
S11 U7
S11 U87
U95 S04
IØWEN-L
IØWEN-H
PSWRD-H
PSRED-L
PSRED-H
C1-L
PS ADDRESS 777768
PS I/O BUS CONTROL
SIZE D
CODE IDENT NO.
DWG NO.
SCALE
SHEET


REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
SEE SHEET 1
BSINT-H
INHBF-L
J2-A11
CR006-H
CR000-L
SPFNC-H
CR005-H
CR008-H
J2-B12
PS004-H
INTCK-L
INTCK-H
RNIST-H
RNIST-L
TBIT-L
SCK01-H
RNINT-L
PRINT-H
RNIL-L
74 74 RNI LATCH
RESET-L
R5 470
+5V
DJ2
5,7,8
HLINT-L
TBIT-L
SLINT-L
PFINT-L
SPAR1-L
7488
EIA07-H
EIA06-H
EIA05-H
EIA04-H
EIA03-H
EIA02-H
EIA01-H
EIA00-H
R3 1K
BSINT-L
SPAR2-L
SPAR3-L
HINTP-L
INTCK-H
INTERRUPT ENTRY TABLE
MINTP-L
RM1
INTERRUPT PRIORITY & REQUEST
TRACE TRAP INTERRUPT
RM2
PFAIL-L
J2-A13
HALTP-L
J1-A9
DF2
7474
SCK00-L
HLINT-H
DB1
PFINT-L
INTERRUPT SYNCHRONIZER
SIZE D
CODE IDENT NO.
DWG NO. C21080210
SCALE: NONE
REV 02
SHEET 7 OF 14

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

MB008-L
MB009-L
MB010-L
MB011-L
MB012-L
MB013-L
MB014-L
MB015-L
FILE6-H (J2-B15)
MMINH-L (J2-A5)
ARWEN-L (J1-A2)
U2 S04
IEXEC-H
S10 U18
S133 U61
S00 U34
S08 U17
S11 U35
S112 U25
S112 U26
U30 S04
U16 S04
WRITE-L (J2-A15)
RESET-L
SCK00-H
SLINT-H (J2-A21)
SLINT-L
MB008-H
LDSPF-H (J1-A6)
+5V
R4 470Ω
YELLW-L (J2-B19)

SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 02	SHEET 8 OF 14



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 01	SHEET 9 OF 14

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

IRO11-L
IRO12-L
IRO13-L
IRO14-L
S20 U42
EMS02-H 11,10

IRO15-L
IRO10-L
IRO09-L
IRO08-L
H30 U41
U30 S04
ZR814-H 10

S00 U31
EMS01-H 11,10

IRO11-H
S20 U42
U30 S04
SGL11-H 10
SGL11-L 10

IRO12-H
IRO13-H
IRO14-H
S10 U49
SEVEN-L 10

SGL11-H
IRO10-L
IRO09-L
S10 U39
S20 U22
EMS00-H 11,10

ZR814-H
IRO06-L
IRO07-L
IRO15-L
S20 U40
S00 U31

IRO08-L
EMA07-H

EMULATE FORMAT SELECT

IRO06-L
IRO07-L
S10 U39
S10 U49

IRO08-H
IRO02-L
S10 U6
S86 U51

IRO15-H
IRO06-H
IRO07-H
S10 U49

IRO07-H
IRO01-L
S86 U51
S10 U6
S10 U6
DTECT-H

IRO06-H
IRO00-L
S86 U51

SRCSE-H
IRO09-L
IRO10-L
IRO11-L
H30 U33
SEQDR-L 12

SOURCE EQUAL DESTINATION REGISTER

+5V
R2 1K
PUP1 14

7474 U24
SKE02-L 10
SKE01-L 10

EMLAT-H
J2-A12
SCK01-H
S00 U23
LDIR-H
SCK00-H

EMULATE SCHEDULER

IRO06-H
IRO10-L
IRO09-L
IRO11-H
IRO10-H
IRO08-H
S00 U31
S20 U22

+5V
R1 470Ω
74S151 U21
DSTEN-H 10
DSTIN-H 10

IRO15-L
EMS00-H
EMS01-H
EMS02-H

SGL11-L
SEVEN-L
S10 U32
U30 S04
SRCSE-H 10

SKE01-L
DSTEN-H
S10 U32
S00 U23
U30 S04
IEXRM-H 12
IEXRM-L 12

SKE02-L
SRCSE-H
S10 U32
S00 U23
SCSEL-L 12

SOURCE/ DESTINATION
DECISION & ENABLE LOGIC

EIA13-H
EIA12-H
EIA14-H
EIA15-H
ITROM-H
S04 U12
S20 U12
S00 U20
DSTIN-H 10

SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 02	SHEET 10 OF 14

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

IR008-H
IR004-H
IR000-H
IR012-H
U62 74S151 → EMA00-H → J1-B1

IR009-H
IR005-H
IR001-H
IR013-H
U72 74S151 → EMA01-H → J1-B2

IR010-H
IR006-H
IR002-H
IR014-H
IR011-H
U63 74S151 → EMA02-H → J1-B3

IR015-H
IR007-H
IR003-H
DMZRO-L
U71 74S151 → EMA03-H → J1-B4

EMS02-H
EMS01-H
EMS00-H

IR004-H
IR010-H
U78 74S151 → EMA04-H → J1-B5

IR005-H
DMZRO-L
U86 74S151 → EMA05-H → J1-B6

+5V
R14 1K
U94 74S151 → EMA06-H → J1-B7

EMS02-H
EMS01-H
EMS00-H

EMS02-H → U30 → EMA07-H → J1-B8

IR005-L
IR004-L
IR003-L
U39 → DMZRO-L

SIZE	CODE IDENT NO.	DWG NO.
D		C21000210
SCALE: NONE	REV 01	SHEET 11 OF 14

C21000210 01

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

EMA00-H
EMA01-H
EMA02-H
EMA03-H
EMA04-H
EMA05-H
EMA06-H
EMA07-H

U55 74187
U46 74187
U37 74187
U28 74187
ME

+5V
RM3

EIA15-H → J2-A30
EIA14-H → J2-B30
EIA13-H → J2-A29
EIA12-H → J2-B29
EIA11-H → N.U. J2-A28
EIA10-H → 12 J2-B28
EIA09-H → 12 J2-A27
EIA08-H → 12 J2-B27
EIA07-H → 7,12 J2-A26
EIA06-H → 7,12 J2-B26
EIA05-H → 7,12 J2-A25
EIA04-H → 7,12 J2-B25
EIA03-H → 7,12 J2-A24
EIA02-H → 7,12 J2-B24
EIA01-H → 7,12 J2-A23
EIA00-H → 7,12 J2-B23

ITROM-H
ETROM-L

EMULATE EXECUTE TABLE

STATUS MASK

RM4
RM5
RM6
+5V

U56 7488

MSK07-H → 5
MSK06-H → 5
MSK05-H → 5
MSK04-H → 5
MSK03-H → 5
MSK02-H → 5
MSK01-H → 5
MSK00-H → 5

IRO15-L
SPF04-L
IEXRM-L
MINTP-L

S00 U19

EIA10-H → 3,5,12
ETROM-L → 12
EIA08-H → 12
ITROM-H → 10,12

U3 H05
S11 U87

EIA09-H
IRO15-H

SEQDR-L
IRO11-H
IRO05-H
IRO10-H
IRO04-H
IRO09-H
IRO03-H

74S157 U48

U47 7488

EIA07-H → 7,12
EIA06-H → 7,12
EIA05-H → 7,12
EIA04-H → 7,12
EIA03-H → 7,12
EIA02-H → 7,12
EIA01-H → 7,12
EIA00-H → 7,12

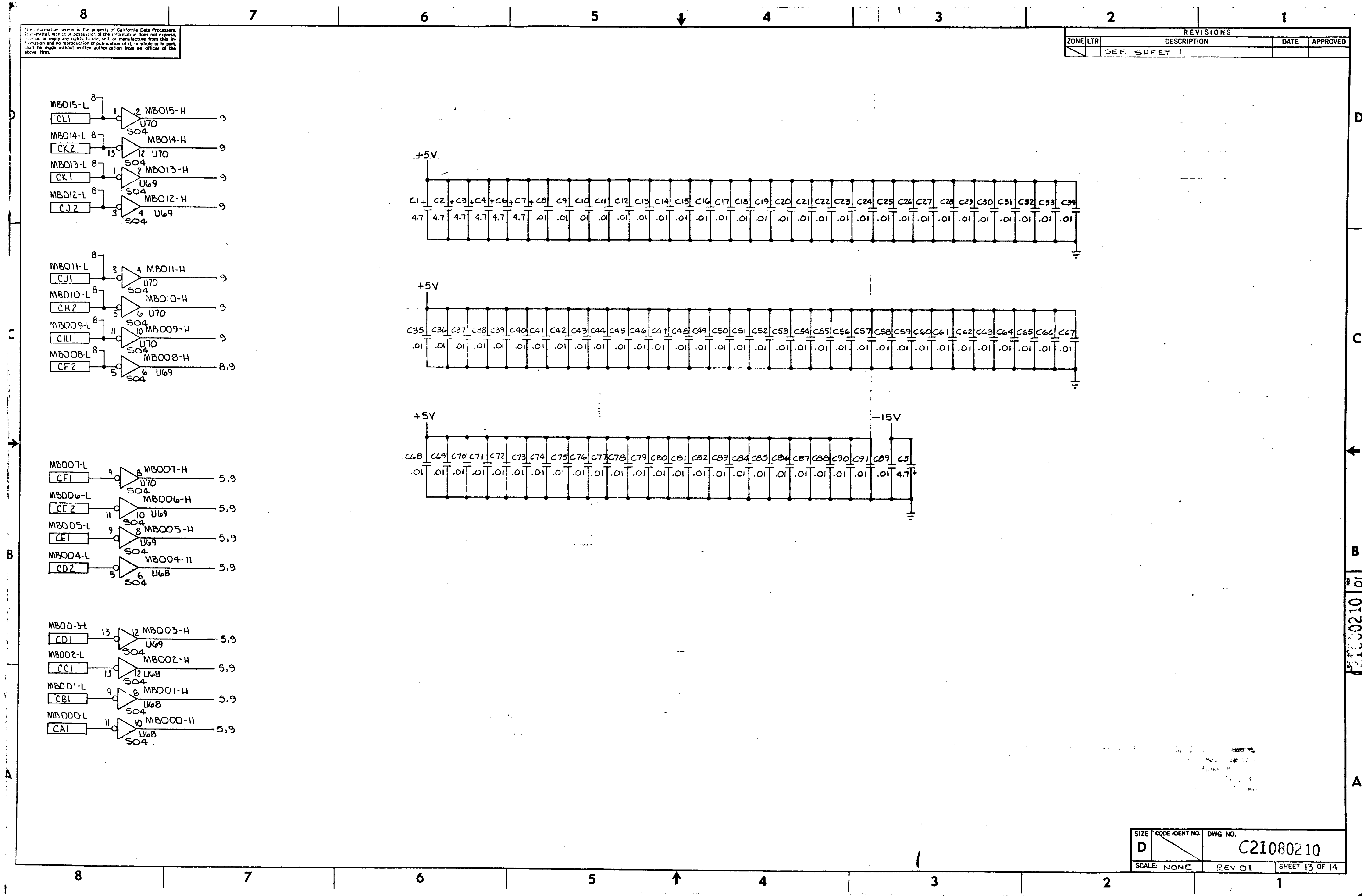
SCSEL-L
IEXRM-H
MINTP-L J2-B2

S00 U19
S04 U2

MINTP-H → N.U.
MINTP-L → 7,12

EMULATE OPERAND TABLE

SIZE	CODE IDENT NO.	DWG NO.
D		C21080210
SCALE: NONE	REV 01	SHEET 12 OF 14


The information hereon is the property of California Data Processors. Transmittal, receipt or possession of the information does not express, license, or imply any rights to use, sell, or manufacture from this information and no reproduction or publication of it, in whole or in part, shall be made without written authorization from an officer of the above firm.
REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
SEE SHEET 1
MB015-L CL1 MB015-H U70 S04
MB014-L CK2 MB014-H U70 S04
MB013-L CK1 MB013-H U69 S04
MB012-L CJ2 MB012-H U69 S04
MB011-L CJ1 MB011-H U70 S04
MB010-L CH2 MB010-H U70 S04
MB009-L CH1 MB009-H U70 S04
MB008-L CF2 MB008-H U69 S04 8,9
MB007-L CF1 MB007-H U70 S04 5,9
MB006-L CE2 MB006-H U69 S04 5,9
MB005-L CE1 MB005-H U69 S04 5,9
MB004-L CD2 MB004-H U68 S04 5,9
MB003-L CD1 MB003-H U69 S04 5,9
MB002-L CC1 MB002-H U68 S04 5,9
MB001-L CB1 MB001-H U68 S04 5,9
MB000-L CA1 MB000-H U68 S04 5,9
+5V
C1–C6 4.7, C7 4.7, C8–C34 .01
+5V
C35–C67 .01
+5V
C68–C91 .01
-15V
C89 .01 C5 4.7
SIZE D
CODE IDENT NO.
DWG NO. C21080210
SCALE: NONE
REV 01
SHEET 13 OF 14

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		SEE SHEET 1		

IR, PS & MS MUX

SIZE	CODE IDENT NO.	DWG NO.
D		C210000-10
SCALE: NONE		REV 02
		SHEET 14 OF 14

```
                        FILE DEFINITIONS
0000
  X0    .EQU  %0
0001
  X1    .EQU  %1
0002
  X2    .EQU  %2
0003
  X3    .EQU  %3
0004
  X4    .EQU  %4
0005
  X5    .EQU  %5
0006
  X6    .EQU  %6
0007
  X7    .EQU  %7
0008
  X8    .EQU  %8
0009
  X9    .EQU  %9
000A
  X10   .EQU  #10
000B
  X11   .EQU  #11
000C
  X12   .EQU  #12
000D
  X13   .EQU  #13
000E
  X14   .EQU  #14
000F
  X15   .EQU  #15
0007
  PC    .EQU  #7
                        REGISTER DEFINITIONS
0010
  DR    .EQU  %10               ; OUTPUT REGISTER
0011
  AR    .EQU  %11               ; BUS ADDRESS REG.
0012
  ER    .EQU  %12               ; E REGISTER
0013
  LR    .EQU  %13               ; STK LIMIT REG.
0013
  SL    .EQU  %13
0014
  IR    .EQU  %14               ; INSTRUCTION REG.
0015
  PS    .EQU  %15               ; MACRO STATUS REG.
0016
  RR    .EQU  %16               ; INPUT REGISTER
001C
  LC    .EQU  %1C               ; LOOP COUNTER
001D
  MSR   .EQU  %1D               ; MICRO-STATUS REG.
```

```
                001E
                  CC      .EQU  %1E               : ROM ADDRESS REG.
                001A
                  HLS     .EQU  %1A               : BIT 15 = 0, HLT SW ON
                001F
                  CS      .EQU  #31               : MICRO-STACK REG.
                0019
                  XY      .EQU  %19
                0017
                  XS      .EQU  %17               : 0S OR 1S DETERMINED BY N STATUS
                000C
                  PCX     .EQU  %C                : LOAD ADDR. FILE
                000B
                  XRA     .EQU  %B                : NEW DATA FILE
                0100
                  INHT    .EQU  %100              : INHIBIT T-BIT INTERRUPT
                000D
                  TCS     .EQU  %D                : CONTROL SWITCH FILE
                0018
                  SFR     .EQU  %18
                D0FF
                          .QET  %D0FF

0000
                          .ORP  %0

                          PROCESSOR INTERRUPT DETERMINATION ROUTINE

0000  704840000178
                  CPUI    MVA   X8,X8             : CANCEL ANY BRANCH
                          BRU   CPUX              : GO TO NXT LOC
                          SKPNX

                          DETERMINE REGISTER OR BUS ADDRESS

0001  7F7F0F80C005
                  RBA     TSBF  X15,#12,BAD       : CHECK FOR REGISTER ADDR.
0002  7C510C000000
                          MVA   PCX,AR
0003  CA1F0C000030
                          ADD   PCX,'%30          : IS ADDR A REG ADDR
                          SNGDF                   : IF BUS ADDR, SKIP NXT
0004  CA1F0C000040
                          ADD   PCX,'%40          : IS ADDR A REG ADDR
                          SNGDF                   : IF REG. ADDR, SKP NXT
0005  7E9E5F100600
                  BAD     ADT   CS,CC             : UPDATE RTN ADDR + 2
                          SKPNX                   : SKIP NXT
                          POP                     : ADJUST STK PTR
0006  C21F0C000038
                          ADD   PCX,'%38          : ARE ONE OF THE LOWER 8
                          SNGDT                   : FILES BEING ADDRESSED
```



```
0007  79524C00000A
                          MVA   PCX,ER            : NO, ONE OF THE UPPER 8,
                          BRU   UFA               : GO TO UPPER 8 EXIT
                          SKPNX
0008  7C520C000400
                          MVA   PCX,ER            : REGISTER ADDR. TO ER
                          RTN                     : RETURN TO MAIN FLOW
0009  7E9F280C0000
                          INC   *X8               : X8 TO 1
                          MYA                     : MODIFY A OPR OF NXT
000A  7E9E1F000600
                    UFA   INC   CS,CC             : UPDATE RTN ADDR + 1
                          POP                     : ADJUST STACK
000B  7E9F280C0000
                          INC   *X8               : 0001 TO X8
                          MYA                     : MODIFY A OPR OF NXT

                          UPDATE DISPLAY ADDRESS

000C  71DF48008801
                    UPA   XOR   X3,X8             : CLEAR X8
                          BRU   RBA               : DETERMINE REG OG BUS
                          SKPNX                   : SKIP NXT
                          SAVE                    : SAVE RELINK
000D  7E0C4800C400
                          ADD   X8,PCX,PCX        : UPDATE REG ADDR BY 1
                          SKPNX                   : SKIP NEXT
                          RTN                     : RETURN TO MAIN FLOW
000E  7E0C4300C400
                          ADD   X8,PCX,PCX        : UPDATE UPPER REG ADDR BY 1,
                          SKPNX                   : RETURN TO MAIN FLOW
                          RTN
000F  7E916C100400
                          ADT   *PCX,AR           : UPDATE BUS ADDR BY 2
                          SKPNX                   : SKIP NEXT
                          RTN                     : RETURN TO MAIN FLOW

                          SWAP INSTRUCTION EXECUTION - SWAP
                          BUS DESTINATION

0010  FFBF00000000
                    SWAB  CHA                     : READ THE DSTN DATA
0011  FF1036000004
                          SWAB  *RR,DR            : SWAP UPPER/LOWER BYTES
                          WRT                     : WRITE RESULT TO MEM
0012  7C5F16000021
                    SCU1  MVA   RR, HS            : UPDATE SWAP CONDITIONS
                          BYTE                    : ON LOWER BYTE

                          FETCH NEXT INSTRUCTION ROUTINE

0013  7C5107000006A
                    FNI   MVA   PC,AR             : ADDR OF NXT TO AR
                          DONXT
                          RNI, IHQ, REDI          : SET RNI INTRP, INTORR
                                                    INTRPS, READ NXT INSTR
```

```
0014  FF951D000000
              MMS   MSR,PS            ; UPDATE MACRO STATUS
                                      INTRP, READ NXT INSTR

              POWER UP ROUTINE

0015  777F1A00F020
        PWP   TSBT  HLS,#15,HALT      ; TEST FOR HALT SW ON
0016  FC6600000004
              MVB   ,'%4,X6
0017  FC6800000014
              MVB   ,'%14,X8          ; SET PWR UP TRAP VECTOR
0018  7E9108100002
              ADT   X8,AR             ; READ NEW PSW FROM TRAP 1
              REDD
0019  7C5108000000
              MVA   X8,AR             ; ADDR OF NEW PC
001A  7C5516000002
              MVA   RR,PS             ; PUT NEW PSW IN PS       12
              REDD                    ; READ NEW PC FROM TRAP
001B  704756000B2
              MVA   RR,PC             ; ADDR OF PWR UP ROUTINE
              BRU   FETCH             ; GO FETCH THE INSTR
              SKPNX

              EXECUTE SUB INSTRUCTION - BUS DESTINATION

001C  73BF00000013
        SUB   CMA                     ; READ DSTN OPR DATA
              BRU   FNI               ; GO TO FETCH NXT, DO NXT FIRST
001D  7E3016F08004
              SUB   RR,X8,DR; USLO    ; SUB SRC FROM DSTN
              WRT                     ; WRITE TO DSTN

              HALT EXECUTION AND CONSOLE ACCESS

001E  7C5F08300000
        HLTP  MVA   X8                ; DUMMY TO NEGATE MXAB
001F  7F7FD500F8FF
              TSBF  PS,#15,ILL        ; IS USER, HALT ILLEGAL
              SKPIB
0020  7C4C07500000
        HALT  MVA   PC,PCX; CLO       ; RESET FLAGS, SAVE PC
0021  7C4B00000000
              MVA   X0,XRA            ; SAVE REG ZERO DATA
0022  FC712903FF60
        HLTT  MVB   *X9,'%FF60,AR     ; CONTROL SW ADDR TO AR
0023  7DC803000002
              XOR   X8,X8,X8
              REDD                    ; READ CONTROL SWITCHES
        >>>>  (02)  DISABLE MMU
0024  7F7FDA000026
              TSBF  HLS,#2,HLTG       ; PWR FAIL INTR PENDING
              SKPIB
```



```
0025  73FF40000025
              BRU   .                 ; YES, LOCK UP
              SKPNX
0026  776D96007048
        HLTG  TSBT  RR,#7,EXCH,TCS    ; IS CONSOLE INTR SET
              DNXIB                   ; IF YES,DO NEXT
0027  FC6800000AC
              MVB   ,'%AC,X8          ; SET CONSOLE INTR. ADDR
0028  777F8D0030B3
              TSBT  TCS,#3,IFETC      ; IS CONTINUE SET
              DNXIB                   ; IF YES,DO NEXT
0029  7C5107000000A
              MVA   PC,AR             ; FETCH NEXT INSTR.
              REDI
002A  777F8D00405B
              TSBT  TCS,#4,CPRS       ; IS START SET
              DNXIB                   ; IF YES,DO NEXT
002B  7C470C000000
              MVA   PCX,PC            ; MOVE NEW PC TO PC
002C  7F7FCD000030
              TSBF  TCS,#0,HLT1       ; IS LOAD ADDR SET
              SKPIB                   ; IS NOT SET, SKIP NXT
002D  FC712A00FF78
              MVB   *X10,'%FF78,AR    ; READ DATA SWITCHES
002E  7C4F0D000002
              MVA   TCS,X15           ; PANEL CONTROL TO X15
              REDD
        >>>>  (28)  DISABLE MMU, READ BITS 16, 17 FROM SW.
002F  704C56500042
              MVA   RR,PCX; CLO       ; STORE LOAD ADDR
              BRU   HLTA              ; GO TO DISPLAY
              SKPNX
0030  7F7FCD001039
        HLT1  TSBF  TCS,#1,HLT2       ; IS EXAMINE SET
              SKPIB                   ; IS NOT SET, SKIP NXT
0031  15DFC840880C
              XOR   X8,X8; CO         ; CLEAR DPST SIGN
              BCYST UPA               ; IF UPDATE, DETRM ADDR
              SKPIB                   ; SKIP NXT IF BRANCH
              SAVE                    ; SAVE RELINK IF BRANCH
0032  71DF48398301
              XOR   X0,X8; SL         ; SET EXAM SIGN
              BRU   RBA               ; DETRMN BUS / REG ADDR
              SKPNX                   ; SKIP NEXT
              SAVE                    ; SAVE RELINK
0033  704B47000042
              MVA   X7,XRA            ; REG DATA TO TEMP SAVE
              BRU   HLTA              ; GO TO DISPLAY
              SKPNX
0034  704B4F000042
              MVA   X15,XRA           ; REG DATA TO TMP SAVE
              BRU   HLTA              ; GO TO DISPLAY
              SKPNX                   ; SKIP NEXT
```

```
0035  7F7FCD006037
                TSBF   TCS,#6,RPHM      : IF PHYSICAL SW SET,
                SKPIB                   : SKIP NXT INSTR.
0036  7C514C000002
                MVA    PCX,AR           : READ FROM BUS ADDR
                SKPNX                   : SKIP NEXT INSTR.
                REDD                    : EXAMINE VIRTUAL SPACE
              >>>>  (48)  USE 16, 17 ADDR BITS IF APPLICABLE
0037  7C510C000002
        RPHM    MVA    PCX,AR           : EXAMINE PHYSICAL SPACE
                REDD                    : EXAMINE PHYSICAL SPACE
              >>>>  (48)  DISABLE MMU/ USE 16,17 ADDR. BITS
0038  704B56000042
                MVA    RR,XRA           : BUS DATA TO TMP SAVE
                BRU    HLTA             : GO TO DISPLAY
                SKPNX                   : SKIP NEXT
0039  7F7FCD002042
        HLT2    TSBF   TCS,#2,HLTA      : IS DEPOSIT SET
                SKPIB                   : IS NOT SET, SKIP NXT
003A  7C510A100002
                MVA    X10,AR; CL       : DATA SWITCH ADDR TO AR
                REDD                    : READ SWITCHES
              >>>>  (08)  DISABLE MMU
003B  244BD610080C
                MVA    RR,XRA; CL       : CLEAT EXAM. SIGN
                BOVST  UPA              : IF UPDATE, DETRM ADDR
                SKPIB                   : SKIP NXT IF BRANCH
                SAVE                    : SAVE RELINK IF BRANCH
003C  71DF48608801
                XOR    X8,X8; SO        : SET DEPST SIGN
                BRU    RBA              : DETRMN BUS / REG ADDR
                SKPNX                   : SKIP NEXT
                SAVE                    : SAVE RELINK
003D  707F6700B042
                MVB    *X7,XRA          : SW. DATA TO REG
                BRU    HLTA             : GO TO DISPLAY
                SKPNX
003E  707F6F00B042
                MVB    *X15,XRA         : SW DATA TO REG AND TMP
                BRU    HLTA             : GO DISPLAY DATA
                SKPNX                   : SKIP NEXT
003F  7F7FCD006041
                TSBF   TCS,#6,WPHM      : IF PHYSICAL SW SET,
                SKPIB                   : SKIP NXT INSTR.
0040  7C5048000004
                MVA    XRA,DR           : SWITCH DATA TO DR, WRT
                SKPNX                   : SKIP NEXT INSTR.
                WRT                     : DEPOSIT INTO VIRTUAL SPACE
              >>>>  (40)  USE 16, 17 ADDR BITS IF APPLICABLE
0041  7C500B000004
        WPHM    MVA    XRA,DR           : DEPOSIT INTO PHYSICAL SPACE
                WRT                     : DEPOSIT INTO PHYSICAL SPACE
              >>>>  (48)  DISABLE MMU/ USE 16,17 ADDR. BITS
```



```
0042  7F7F0D008045
        HLTA    TSBF   TCS,#8,HLTD      : IS DISPLAY ADDR SET
0043  705109000022
                MVA    X9,AR            : LITE ADDR TO AR
                BRU    HLTT             : AFTER NXT, DO AGAIN
0044  7C500C000004
                MVA    PCX,DR           : LOAD ADDR TO DR
                WRT                     : DISPLAY ADDR
              >>>>  (08)  DISABLE MMU
0045  7C500B000004
        HLTD    MVA    XRA,DR           : DATA TO DR
                WRT                     : DISPLAY DATA
              >>>>  (08)  DISABLE MMU

                BUS INTERRUPT ROUTINE

0046  FC7300000001
        BIO     MVB    ,'%1,SL          : DO BR ACKNOWLEDGE
0047  7C4816000000
                MVA    RR,X8            : SET VECTOR ADDR
0048  7F7FDD00B0B2
                TSBF   MSR,#11,FETCH    : IS INTR. REQUESTED
                SKPIB                   : SKIP NXT IS NO INTRP
0049  7C4E08000000
                MVA    X8,X14
004A  FC7300000002
                MVB    ,'%2,SL          : YES, DO INTR. ACK.

                TRAP REGISTER EXCHANGE ROUTINES

0048  7EB119000002
        EXCH    DEC    XY,AR            : READ PSW VIA BUS
                REDD
              >>>>  (09)  DISABLE MMU
004C  7E9108100000
                ADT    X8,AR            : TRAP ADDR OF NEW PS
004D  7C4C16000002
                MVA    RR,X12           : SAVE OLD PSW
                REDD                    : READ NEW PSW FROM TRAP VECTOR
              >>>>  (04)  FORCE SUPERVISOR MODE
004E  FD5F2E008000
                ORI    *X14,'%8000      : SET PSW ORIGIN
004F  7C5108000000
                MVA    X8,AR            : TRAP ADDR OF NEW PC
0050  7C5516000000
                MVA    RR,PS            : LOAD NEW PSW INTO PS
              >>>>  (01)  SHIFT CURRENT MODE TO PREVIOUS MODE
0051  7C500C000002
                MVA    X12,DR           : OLD PSW TO OUTPUT REG.
                REDD                    : READ NEW PC ADDR
              >>>>  (04)  FORCE SUPERVISOR MODE
0052  7EB126100004
                SBT    *X6,AR           : PUT OLD PSW INTO STK
                WRT
```

```
0053  7C5007000000
            MVA   PC,DR             : OLD PC ADDR TO OUTPUT
0054  7EB126100004
            SBT   *X6,AR            : PUT OLD PC INTO STK
            WRT
0055  FC73000000C0
            MVB   ,'%C0,SL          : RESET S SYNC/D SYNC
0056  7F7F15004059
            TSBF  PS,#4,TBNO        : TEST FOR T BIT
0057  7C4716000000
      RPRS  MVA   RR,PC             : NEW ADDR TO PC
0058  FC7331000100
            MVB   *AR,'%100,SL      : RESET YELLOW STK LMT
0059  FCDF2E007FFF
      TBNO  AND   *X14,'%7FFF       : REMOVE PSW ORIGIN
005A  7C514700006A
      NTB   MVA   PC,AR             : READ NXT INSTR
            SKPNX                   : SKIP NEXT MICROINSTRUCTION
            RNI; INQ; REDI
005B  71D50800080B2
      CPRS  XOR   X8,X8,PS          : CLEAR PSW
            BRU   FETCH
005C  FC73000005F0
            MVB   ,'%5F0,SL         : RESET ALL CONDITIONS

            MFPI EXECUTION ROUTINE - MEM SOURCE

005D  7C5F08000002
      MFPI  MVA   X0
            REDD                    : READ FROM PREV SPC
      >>>>  (02)  FORCE PREVIOUS MODE
005E  7C5016000000
            MVA   RR,DR             : PREV. DATA TO OUTPUT REG
005F  72B126100013
      MFXX  SBT   *X6,AR            : DECR STK PTR, TO AR
            BRU   FNI               : GO TO FETCH NEXT AFTER NEXT
0060  7C5F10800004
            MVA   DR; MS            : SET CONDITIONS
            WRT                     : PUT PREV. DATA ON STK

            MFPI EXECUTION ROUTINE - REG. SOURCE

0061  704848000062
      MFPR  MVA   X8,X8             : ABORT  GO TO FETCH NXT
            BRU   MFXX              : BYPASS FNI JUMP
            SKPNX
0062  73FF000C005F
      MFRX  BRU   MFXX              : GO PUT DATA ON STK AFTER NEXT
            MVA                     : MODIFY A OPR OF NEXT
0063  7C5007000000
            MVA   X7,DR             : PREV DATA TO OUTPUT REG
      >>>>  (10)  FORCE PREVIOUS MODE FOR 1 CLOCK

            MTPI EXECUTION ROUTINE - BUS SOURCE
```



```
0064  FCDF3200003F
      MTPI  AND   *ER,'%3F          : MASK OUT DSTN MODE/REG
0065  81DF1200001E
            XOR   ER,'%1E           : IS DSTN @-(R6)
            SZRDT                   : IF SO, SKIP NEXT
0066  89DF1200002E
            XOR   ER,'%2E           : IS DSTN @(R6)+
            SZRDF                   : IF NOT, SKIP NEXT
0067  705051000060
            MVA   AR,DR             : SAVE AS CURRENT DATA
            BRU   MTOX              : GO GET PREV. STK ADDR
            SKPNX                   : SKIP NEXT INSTR
0068  7C4911000000
            MVA   AR,X9             : SAVE PREV SPC ADDR
0069  7C5106000002
            MVA   X6,AR             : READ FROM CRNT STACK
            REDD
006A  7E9F26100000
            ADT   *X6               : INCR CRNT STK PTR
006B  705016800013
            MVA   RR,DR; MS         : CRNT MODE DATA TO OUTPUT REG
            BRU   FNI               : GO TO FETCH NEXT
006C  7C5109000004
            MVA   X9,AR             : PREV. SPC ADDR TO AR
            WRT                     : WRITE CRNT DATA TO PREV SPC
      >>>>  (02)  FORCE PREVIOUS MODE
006D  7C5106000002
      MTOX  MVA   X6,AR             : READ FROM CURRENT STK
            REDD
006E  7E9F26100000
            ADT   *X6               : INCR CRNT STK PTR
006F  705116000013
            MVA   RR,AR             : ADDR IN PREV SPC TO AR
            BRU   FNI               : GO FETCH NXT INSTR
0070  7C5F10800004
            MVA   DR; MS            ;SET CONDITION CODES
            WRT                     : WRITE CRNT DATA TO PREV SPACE
      >>>>  (02)  FORCE PREVIOUS MODE

            MTPI EXECUTION ROUTINE - REG SOURCE

0071  706871006072
      MTPR  MVB   *AR,X6,X8         : STK ADDR TO AR
            BRU   MTRX              : BYPASS FNI JUMP
            SKPNX
0072  7E9F26110002
      MTRX  ADT   *X6               : INCR STK PTR BY 2
            REDD                    : READ FROM STACK
            MXD
0073  704756800013
            MVA   RR,X7; MS         : STK DATA TO REG
            BRU   FNI               : GO FETCH NXT INSTR
            SKPNX                   : SKIP NEXT INSTR
      >>>>  (10)  FORCE PREVIOUS MODE FOR 1 CLOCK

            MARK EXECUTION ROUTINE
```

```
0074  7C4814000021
        MARK  MVA   IR,X8            : GET DISPLACEMENT
              BYTE
0075  7E0808008021
              ADD   X8,X8,X8         : COMPUTE STK ADJUSTMENT
              BYTE
0076  7E1107008002
              ADD   PC,X8,AR         : COMPUTE NEW STK ADDR
              REDD                   : READ FROM STK
0077  7C4705000000
              MVA   X5,PC            X5 TO PC
0078  728611100082
              ADT   AR,X6            : INCR STK PTR BY 2
              BRU   FETCH            : GO FETCH NXT INSTR
0079  7C4516000000
              MVA   RR,X5            : LOAD X5 FROM STK

              SUBROUTINE JUMP INSTRUCTION EXECUTION

007A  7C4811000000
        JSR   MVA   AR,X8            : JMP ADDR TO X8
007B  7EB126140000
              SBT   *X6,AR           : DECR STK PTR
              MXA                    : MODIFY A OPR OF NXT
007C  7C5007000004
              MVA   X7,DR            : SRC REG TO OUTPUT REG
              WRT                    : WRITE SRC REG CONTENTS TO STK
007D  7C4807000000
              MVA   PC,X11           : SAVE CRNT PC
007E  7C5F0F040000
              MVA   X15              : DELAY 1 CLOCK
              MXA                    : MODIFY A OPR OF NEXT
007F  707F2700B0B2
              MVB   *X7,X11          : SAVE  OLD PC IN SRC REG
              BRU   FETCH            : GO FETCH NEXT INSTR
0080  7C4708000000
              MVA   X8,PC            : JMP ADDR TO PC

              RETURN FROM INTERRUPT ROUTINE
              IGNORE T BIT

0081  FC7F68300FFF
        RTT   MVB   *X8,'%FFF; SL    : INDICATE RTT EXIT
              SKPNX                  : SKIP NEXT

              RETURN FROM INTERRUPT INSTRUCTION EXECUTION

0082  FC7F28100FFF
        RTI   MVB   *X8,'%FFF; CL    : INDICATE RTI EXIT
0083  7C5106000002
              MVA   X6,AR            : READ PC ADDR FROM STK
              REDD
```



```
0084  7E9F26100000
              ADT   *X6              : UPDATE STK REG
0085  7C4716000000
              MVA   RR,PC            : NEW PC ADDR TO PC
0086  7C5106000002
              MVA   X6,AR            : STK ADDR TO AR
              REDD                   : READ NEW PSW FROM STK
0087  7E9F26100000
              ADT   *X6              : UPDATE STK REG
0088  777FD500F08D
              TSBT  PS,#15,RNM       : IS CURRENT MODE USER
              SKPIB                  : IF SO, STAY IN USER
0089  7CDF36008000
              AND   *RR,X8           : AND OUT MODE IN STK PSW
008A  7C8915000080
              OCA   PS,X9            : ADJUST USER/ SUPRV PS BITS
008B  7CE809008000
              NDB   X9,X8,X8         : SAVE MODE IN CURRENT PS
008C  155556008090
              ORI   RR,X8,PS         : FORM NEW PSW
              BCYST RTEX             : IF RTT, INHIBIT T-BIT INTRP
              SKPNX
008D  145556000090
        RNM   MVA   RR,PS            : NEW PSW TO PS
              BCYST RTEX             : IF RTT, INHIBIT T-BIT INTRP
              SKPNX                  : SKIP NEXT MICROINSTRUCTION
008E  7DDF38008000
              XOR   X8,X8            : DUMMY TO GET 1 CLOCK
008F  7C5147000060
              MVA   PC,AR            : NEW PC TO AR
              SKPNX
              RNI; INQ; REDI         : READ NXT INSTR
0090  7C5147000160
        RTEX  MVA   PC,AR            : NEW PC ADDR. TO AR
              SKPNX                  : SKIP NEXT MICROINSTRUCTION
              RNI; INQ; REDI         : SET RHI INTRP, DO INTRP INQUIRY,
              FORM  INHT             : DO INSTR. FETCH, INHIBIT T-BIT INTRP

              RED STACK LIMIT AND DOUBLE SLAVE SYNC ERROR ROUTINE

0091  7E9F26100000
        SKE   ADT   *X6              : ADJUST STK REG
0092  FC7300▷005C0
        SKE1  MVB   ,'%5C0,SL        : RESET STK LMT/ SLV SYNC/ DBL SLV SYNC
                                       DBL SLV SYNC IND.
0093  405F8E060097
              MVA   X14              : EXAMINE X14 FOR PSW ORIGIN
              BNGDT RSXX             : IF FROM STACK RTN,
              DNXIB                  : DO NEXT
0094  7C500C000000
              MVA   X12,DR           : FROM STK RTN, OLD PS IN X12
0095  7EB119000002
              DEC   XY,AR            : READ PSW VIA BUS
              REDD
        >>>>  (08)  DISABLE MMU
```

```
0096  7C5016000000
            MVA   RR,DR             : OLD PSW TO OUTPUT REG
0097  7E9108100002
      RSXX  ADT   X8,AR             : TRAP ADDR OF NEW PSW
            REDD                    : READ NEW PSW FROM TRAP
        >>>>  (08)  DISABLE MMU
0098  7C5108000000
            MVA   X8,AR             : TRAP ADDR OF NEW PC
0099  7C5516000002
            MVA   RR,PS             : LOAD NEW PSW INTO PS
            REDD                    : READ NEW PC ADDR
        >>>>  (05)  FORCE SUPERVISOR MODE, SHIFT CURRENT
                    MODE TO PREVIOUS MODE
009A  7EB108100004
            SBT   X8,AR             : STORE OLD PSW IN LOC 2
            WRT
        >>>>  (0C)  DISABLE MMU, FORCE SUPERVISOR MODE
009B  7C5007000000
            MVA   PC,DR             : OLD PC ADDR TO OUTPUT
009C  7DD108008004
            XOR   X8,X8,AR          : PUT OLD PC ADDR IN
            WRT                     : LOC 0
        >>>>  (0C)  DISABLE MMU, FORCE SUPERVISOR MODE
009D  7C4611008000
            MVA   AR,X6             : STACK REG TO ZERO
009E  FCDF2E007FFF
            AND   *X14,'%7FFF       : REMOVE PSW ORIGIN
009F  704716000050
            MVA   RR,PC             : NEW PC ADDR TO PC
            BRU   NTB               : ENTER STACK ROUTINE
00A0  777FDA00F020
            TSBT  HLS,#15,HALT      : TEST FOR HALT SW ON
            SKPIB                   : IF ON, GO TO HALT

                  BREAKPOINT TRAP ROUTINE

00A1  71DF08000040
      BKP   XOR   X8,X8             : GO TO STK PSW/PC
            BRU   EXCH-#1

                  T-BIT INTERRUPT ROUTINE

00A2  FC6840000000
      TBIT  MVB   ,'%C,X8           : SET ADDR OF T BIT/BKP TRAP
            SKPNX

                  YELLOW STACK LIMIT INTERRUPT ROUTINE

00A3  FC6800000004
      YSL   MVB   ,'%4,X8           : SET ADDR OF Y STK LMT TRAP
00A4  7E8119008002
      YSL1  DEC   XY,AR             : READ PSW VIA BUS
            REDD
        >>>>  (08)  DISABLE MMU
```



```
00A5  7E9108100000
            ADT   X8,AR             : TRAP ADDR OF NEW PS
00A6  7C4C16000002
            MVA   RR,X12            : SAVE OLD PSW
            REDD                    : READ NEW PSW FROM TRAP VECTOR
        >>>>  (04)  FORCE SUPERVISOR MODE
00A7  FD5F2E000300
            ORI   *X14,'%8000       : SET PSW ORIGIN
00A8  7C5108000000
            MVA   X8,AR             : TRAP ADDR OF NEW PC
00A9  7C5516000000
            MVA   RR,PS             : LOAD NEW PSW INTO PS
        >>>>  (01)  SHIFT CURRENT MODE TO PREVIOUS MODE
00AA  7C500C000002
            MVA   X12,DR            : OLD PSW TO OUTPUT REG.
            REDD                    : READ NEW PC ADDR
        >>>>  (04)  FORCE SUPERVISOR MODE
00AB  7EB126100004
            SBT   *X6,AR            : PUT OLD PSW INTO STK
            WRT
00AC  7C5007000000
            MVA   PC,DR             : OLD PC ADDR TO OUTPUT
00AD  7EB126100004
            SBT   *X6,AR            : PUT OLD PC INTO STK
            WRT
00AE  73FF40000057
            BRU   RPRS              : ENTER STACK ROUTINE
            SKPNX                   : SKIP NEXT INSTR.

                  RESET INSTRUCTION EXECUTION

00AF  7C5F08000000
        RSTC  MVA   X8
00B0  7F7FD500F0B2
            TSBF  PS,#15,FETCH      : IF USER, RESET = NOP
            SKPIB
00B1  FC7300000010
        RSET  MVB   ,'%10,SL        : RESET BUSS

                  INSTRUCTION FETCH ROUTINE

00B2  7C51C700004A
        FETCH MVA   PC,AR           : PC TO MEMORY ADDR.
            SKPIB                   : MAKE INTRP INQR. AND
            INQ; REDI               : READ INSTR, CHK INTRPS
00B3  7DC928400000
        IFETC XOR   *X8,X8,X9; CO   : CLR SRC DATA SAVE
00B4  FCCC15000001
            AND   PS,'%1,X12        : SAVE MACRO CARRY
00B5  7E9127100000
            ADT   *PC,AR            : UPDATE PC BY 2
00B6  7C1214000000
            EML   IR,ER             : COPY FETCHED INSTR.
```

```
00B7  FF101D464000
                SLLC  MSR,DR; UL        : MOVE +1,+2 BITS
                MXAB                    : TO X15,MODIFY NEXT
                                          BY SOURCE REG. REF.

                POWER FAIL INTERRUPT ROUTINE

00B8  FC7300000200
        PFI     MVB   ,'%200,SL         : RESET POWER FAIL IND
00B9  FC6800000014
                MVB   ,'%14,X8          : SET TRAP VECTOR ADDR
00BA  73FF4000004B
                BRU   EXCH              : DO STACK PUSH
                SKPNX

                FETCH SOURCE OPERAND - MODE 0

00BB  7C0807800080
        SMZ     EML   X7,X8; MS         : MOVE SRC DATA TO X8
                QBYT                    : A BYTE IF DESIRED
00BC  7C5107060000
                MVA   PC,AR             : MOVE PC ADDR TO AR
                MXAB                    : MODIFY A AND B OPR OF NXT

                FETCH SOURCE OPERAND - MODE 1

00BD  7C5107000012
        SM1     MVA   X7,AR             : MOVE SRC OPR ADDR TO AR
                REDD*                   : READ WORD/BYTE OF SRC DATA
00BE  7C0816800080
        SM1X    EML   RR,X8; MS         : GO TO DSTN OPR FETCH
                QBYT                    : MOVE WORD/BYTE OF SRC DATA
00BF  7C5107060000
                MVA   PC,AR             : MOVE PC TO AR
                MXAB                    : MODIFY A AND B OPR OF NXT

                FETCH SOURCE OPERAND - MODE 2

00C0  7C5107040012
        SM2     MVA   X7,AR             : MOVE SRC OPR ADDR TO AR
                MXA                     : MODIFY A OPR. OF NXT
                REDD*                   : READ WORD/BYTE OF SRC DATA
00C1  729F6720000BE
                ADQ   *X7               : UPDATE SRC REG. BY 1 OR 2
                BRU   SM1X              : ENTER COMMON MODE EXIT
                SKPNX                   : SKIP NXT CLOCK EXEC.

                FETCH SOURCE OPERAND - MODE 3

00C2  7C5107040002
        SM3     MVA   X7,AR             : MOVE SRC OPR ADDR TO AR
                MXA                     : MODIFY A OPR. OF NXT
                REDD                    : READ ADDR OF SRC DATA
```



```
00C3  729F271000BE
                ADT   *X7               : UPDATE SRC REG. BY 2
                BRU   SM1X              : ENTER COMMON MODE EXIT
00C4  7C5116000012
                MVA   RR,AR             : ADDR OF SRC DATA TO AR
                REDD*                   : READ WORD/BYTE OF SRC DATA

                FETCH SOURCE OPERAND - MODE 4

00C5  7EB127200012
        SM4     SBQ   *X7,AR            : DECR. SRC DATA ADDR BY 1 OR 2
                REDD*                   : READ WORD/BYTE OF SRC DATA
00C6  73FF400000BE
                BRU   SM1X              : ENTER COMMON MODE EXIT
                SKPNX                   : SKIP NXT CLOCK EXEC.

                FETCH SOURCE OPERAND - MODE 5

00C7  7EB127100002
        SM5     SBT   *X7,AR            : DECR. SRC DATA ADDR BY 2
                REDD                    : READ ADDR OF SRC DATA
00C8  73FF000000BE
        SM5X    BRU   SM1X              : ENTER COMMON MODE EXIT
00C9  7C5116000012
                MVA   RR,AR             : ADDR OF SRC DATA TO AR
                REDD*                   : READ WORD/BYTE OF SRC DATA

                FETCH SOURCE OPERAND - MODE 6

00CA  7DDF08008002
        SM6     XOR   X8,X8             : A DUMMY
                REDD                    : READ NXT WORD FROM PC
00CB  729F271A00BE
                ADT   *PC               : UPDATE PC BY 2
                BRU   SM1X              : ENTER COMMON MODE EXIT
                MYB                     : MODIFY B OPR OF NXT BY SRC
00CC  7E1116007012
                ADD   RR,X7,AR          : COMPUTE ADDR OF SRC DATA
                REDD*                   : READ WORD/BYTE OF SRC DATA

                FETCH SOURCE OPERAND - MODE 7

00CD  7DDF08008002
        SM7     XOR   X8,X8             : A DUMMY
                REDD                    : READ NXT WORD FROM PC
00CE  729F271A00C8
                ADT   *PC               : UPDATE PC BY 2
                BRU   SM5X              : EXIT TO READ SRC DATA
                MYB                     : MODIFY B OPR OF NXT BY SRC
00CF  7E1116007002
                ADD   RR,X7,AR          : COMPUTE ADDR OF SRC DATA ADDR
                REDD                    : READ ADDR OF SRC DATA

                FETCH DESTINATION OPERAND - MODE ZERO
```

```
00D0  7C0907020080
        DM2   EML   X7,X9            : MOVE SRC REG. DATA TO X9
              MXB                    : MODIFY B OPR OF NXT
              QBYT                   : MOVE BYTE IF SPECIFIED
00D1  70762B017013
              MVB   *X11,X7,RR       : MOVE DSTN DATA TO X11,RR
              MXD                    : MODIFY D OPR OF NXT
              BRU   FNI              : GO FETCH NXT, DO EML FIRST

              FETCH DESTINATION OPERAND - MODE 1

00D2  7C0907020000
        DM1   EML   X7,X9            : MOVE SRC REG. DATA TO X9
              MXB                    : MODIFY B OPR OF NXT
00D3  7C7100007000
              MVB   ,X7,AR           : MOVE DSTN ADDR TO AR

              FETCH DESTINATION OPERAND - MODE 2

00D4  7C4907000000
        DM2   MYA   X7,X9            : MOVE SRC REG. DATA TO X9
00D5  FF1F104C4000
              SLLO  DR; UL           : SET UPDATE BY 1 OR 2 CONDITION
              MYA                    : MODIFY A OPR OF NXT
00D6  7C11070C0000
              EML   X7,AR            : MOVE DSTN ADDR TO AR
              MYA                    : MODIFY A OPR OF NXT
00D7  7E9F27200000
              ADQ   *X7              : INCR DSTN REG BY 1 OR 2

              FETCH DESTINATION OPERAND - MODE 2 - SM = 0, SR = DR

00D8  FE1F104C8000
        DM2A  ADD   DR,'%8000; UL    : SET UPDATE BY 1 OR 2 CONDITION
              MYA                    : MODIFY A OPR OF NXT BY DSTN REG
00D9  7C51070C0000
              MYA   X7,AR            : SAVE DSTN ORP ADDR IN AR
              MYA                    : MODIFY A OPR OF NXT BY DSTN REG
00DA  7E9F27240000
              ADQ   *X7              : UPDATE DSTN REG.
              MXA                    : MODIFY A OPR OF NXT BY SRC REG
00DB  7C0907410000
        DM2X  EML   X7,X9; CO        : GO TO EXECITE, SRC REG TO SRC OPR ADDR
              MXA                    : MODIFY A OPR OF NXT BY SRC REG
00DC  7C4807809080
        DM2Y  MYA   X7,X8; MS        : SAVE CURRENT SRC DATA
              QBYT                   : WORD OR BYTE

              FETCH DESTINATION OPERAND - MODE 3

00DD  7C71006C7002
        DM3   MVB   ,X7,AR; SO       : MOVE DSTN ADDR TO AR
              MYA                    : MODIFY A OPR OF NXT
              REDD                   : READ DSTN OPR ADDR
```



```
00DE  729F271400EC
              ADT   *X7              : UPDATE DSTN REG. BY 2
              MXA                    : MODIFY A OPR OF NXT
              BRU   DX0
00DF  7C0907000000
              EML   X7,X9            : MOVE SRC REG. DATA TO X9

              FETCH DESTINATION OPERAND - MODE 3 - SM = 0, SR = DR

00E0  7C71006C7002
        DM3A  MVB   ,X7,AR; SO       : MOVE DSTN ADDR TO AR
              MYA                    : MODIFY A OPR OF NXT BY DSTN REG
              REDD                   : READ DSTN OPR ADDR
00E1  729F271000DB
              ADT   *X7              : UPDATE DSTN REG. BY 2
              BRU   DM2X             : GO TO COMMON EXIT
00E2  7C5116040000
              MYA   RR,AR            : DSTN OPR ADDR TO AR
              MXA                    : MODIFY A OPR OF NXT BY SRC REG.

              FETCH DESTINATION OPERAND - MODE 4

00E3  FE1F104C8000
        DM4   ADD   DR,'%8000; UL    : SET UPDATE BY 1 OR 2 CONDITION
              MYA                    : MODIFY A OPR OF NXT
00E4  7EB127240000
              SBQ   *X7,AR           : DECR DSTN REG BY 1 OR 2
              MXA                    : MODIFY A OPR OF NXT
00E5  7C0947000000
              EML   X7,X9            : MOVE SRC REG. DATA TO X9
              SKPNX                  : SKIP EXEC OF NXT CLOCK

              FETCH DESTINATION OPERAND - MODE 4 - SM = 0, SR = DR

00E6  FE1F104C8000
        DM4A  ADD   DR,'%8000; UL    : SET UPDATE BY 1 OR 2 CONDITION
              MYA                    : MODIFY A OPR OF NXT BY DSTN REG
00E7  72B1272400DC
              SBQ   *X7,AR           : DECR DSTN REG BY 1 OR 2
              BRU   DM2Y             : GO TO COMMON EXIT
              MXA                    : MODIFY A OPR OF NXT BY SRC REG.
00E8  7C0907440000
              EML   X7,X9; CO        : GO TO EXECITE, SRC REG TO SRC OPR ADDR
              MXA                    : MODIFY A OPR OF NXT BY SRC REG.

              FETCH DESTINATION OPERAND - MODE 5

00E9  7C48086C0000
        DM5   MYA   X8,X8; SO        : A DUMMY
              MYA                    : MODIFY A OPR OF NXT
00EA  7EB127140002
              SBT   *X7,AR           : DSTN OPR ADDR TO AR
              REDD                   : READ DSTN OPR ADDR
              MXA                    : MODIFY A OPR OF NXT
```

```
00EB  7C0907000000
        DXX   EML   X7,X9           : MOVE SRC REG. DATA TO X9
00EC  7C5116400000
        DX0   MVA   RR,AR; C0       : MOVE DSTN OPR ADDR TO AR

              FETCH DESTINATION OPERAND - MODE 5 - SM = 0, SR = DR

00ED  7C5F086C0009
        DM5A  MVA   X8; S0
              MVA                   : MODIFY A OPR OF NXT BY DSTN REG
00EE  7E8127140002
              SBT   *X7,AR          : ADDR OF DSTN OPR ADDR TO AR
              MXA                   : MODIFY A OPR OF NXT BY SRC REG.
              REDD                  : READ ADDR OF OPR
00EF  704907000DDC
        DM5X  MVA   X7,X9           : SAVE SRC REG A SRC ADDR
              BRU   DM2Y            : GO TO COMMON EXIT
00F0  7C1116440000
              EML   RR,AR; C0       : GO TO EXECUTE, DSTN OPR ADDR TO AR
              MXA                   : MODIFY A OPR OF NXT BY SRC REG.

              DESTINATION MODE 6 - SM = 0, SR = DR OR SR = 7

00F1  7C5F08000002
        DM6A  MVA   X8              : READ DSTN OPR DISP.
              REDD
00F2  729F271200DB
              ADT   *PC             : UPDATE PC
              BRU   DM2X            : GO TO COMMON EXIT
              MXB                   : MODIFY B OPR OF NXT BY DSTN REG
00F3  7E1116047000
              ADD   RR,X7,AR        : DSNT OPR ADDR TO AR
              MXA                   : MODIFY A OPR OF NXT BY SRC REG.

              FETCH DESTINATION OPERAND - MODE 6

00F4  7C4907000002
        DM6   MVA   X7,X9           : MOVE SRC REG. DATA TO X9
              REDD                  : READ DSTN OPR DISPL.
00F5  7E9F27140000
              ADT   *PC             : UPDATE PC ADDR.
              MXA                   : MODIFY A OPR OF NEXT
00F6  7C0907020000
        DM6X  EML   X7,X9           : GO TO EXECUTE, SCR ADDR TO X9
              MXB                   : MODIFY B OPR OF NXT
00F7  7E1116007000
              ADD   RR,X7,AR        : COMPUTE DSTN OPR ADDR.

              FETCH DESTINATION OPERAND - MODE 7

00F8  7C4808000002
        DM7   MVA   X8,X8           : A DUMMY
              REDD                  : READ DSTN OPR DISPL.
```



```
00F9  7E9F27100000
              ADT   *PC             : UPDATE PC ADDR.
00FA  73FF000200EB
        DM7X  BRU   DXX             : GO TO COMMON ENTRY
              MXB                   : MODIFY B OPR OF NXT
00FB  7E1116047002
              ADD   RR,X7,AR        : COMPUTE ADDR OF DSTN OPR ADDR
              REDD                  : READ DSTN OPR ADDR
              MXA                   : MODIFY A OPR OF NXT

              DESTINATION MODE 7 - SM = 0, SR = DR OR SR = 7

00FC  7C5F08000002
        DM7A  MVA   X8              : READ DSTN OPR DISP.
              REDD
00FD  729F271200EF
              ADT   *PC             : UPDATE PC
              BRU   DM5X            : GO TO COMMON EXIT
              MXB                   : MODIFY B OPR OF NXT BY DSTN REG
00FE  7E1116047002
              ADD   RR,X7,AR        : ADD DISP TO DSTN REG FOR DSTN OPR ADDR
              MXA                   : MODIFY A OPR OF NXT BY SRC REG.
              REDD                  : READ DSTN OPR ADDR
00FF
              .ORP  %FF

              ILLEGAL/RESERVED INSTRUCTION TRAP ROUTINE

00FF  71DF0800804B
        ILL   XOR   X8,X8
              BRU   EXCH            : GO TO SAVE REGISTERS
0100  FC6800000008
              MVB   ,'X8,X8         : SET ILLEGAL INST VECTOR


              WAIT INSTRUCTION EXECUTION

0101  705F28000101
        WAIT  MVA   *X8
              BRU   .               : BRANCH ON SELF
0102  7C5FE8A00040
              MVA   *X8
              SKPIB                 : INTR SKIP NXT
              INQ                   : DO INTR INQUIRY

              EMULATE TRAP INSTRUCTION EXECUTION

0103  705F28000048
        EMT   MVA   *X8             : NEGATE MODIFY NXT
              BRU   EXCH            : DO REGISTER EXCHANGE
```

```
0104  FC6800000018
              MVB   ,'%18,X8          : SET EMT TRAP LOC

              I/O TRAP INSTRUCTION EXECUTION

0105  705F2800004B
        IOT   MVA   *X8               : NEGATE MODIFY NXT
              BRU   EXCH              : DO REGISTER EXCHANGE
0106  FC6800000010
              MVB   ,'%10,X8          : SET IOT TRAP LOC
              TRAP INSTRUCTION EXECUTION

0107  705F2800004B
       TRAP   MVA   *X8               : NEGATE MODIFY NXT
              BRU   EXCH              : DO REGISTER EXCHANGE
0108  FC680000001C
              MVB   ,'%1C,X8          : SET TRAP VECTOR


              MOVE WITH REG AS DSTN

0109  7C2708800080
        MOVR  SXA   X8,X7; MS         : MOVE SRC TO DSTN REG.
              QBYT                    : ALLOW BYTE OPERATION

              SUBTRACT ONE AND BRANCH INSTRUCTION EXECUTION

010A  32BFE7800302
        SOB   DEC   *X7               : SUBTRACT 1 FROM REG.
              BZRDT FETCH             : IF RESULT ZERO, GO TO FETCH
              SKPIB
010B  FF0812004000
        JMB   SLLO  ER,X8             : DOUBLE BR VECTOR
010C  FCDF2800007F
              AND   *X8,'%007F        : MASK OUT UPPER 9 BITS
010D  7E316700806A
              SUB   *PC,X8,AR         : COMPUTE NXT INST ADDR
              SKPNX                   : DO INTR INQ, SET RNI,
              RNI; INQ; REDI          : READ NXT INST.

              SXT EXECUTION ROUTINE
              BUS DESTINATION

010E  7C5017800004
        SXT   MVA   XS,DR; MS         : CLEAR OR SET ALL BITS
              URT
010F  73FF40000013
              BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
              SKPNX

              SXT EXECUTION ROUTINE
              REGISTER DESTINATION
```



```
0110  7C4717800000
        SXR   MVA   XS,X7; MS         : CLEAR OR SET ALL BITS

              EX-OR INSTRUCTION EXECUTION
              BUS DESTINATION

0111  73BF00000013
        XOR   CMA                     : READ DSTN OPR DATA
              BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
0112  7DD016809004
              XOR   RR,X9,DR; MS      : EX-OR SRC WITH DSTN
              URT

              EX-OR INSTRUCTION EXECUTION
              REGISTER DESTINATION

0113  7DC70B809000
       XORR   XOR   X11,X9,X7; MS     : EX-OR SRC WITH DSTN

              CLEAR WORD/BYTE INSTRUCTION EXECUTION
              REGISTER DESTINATION

0114  7C4708800080
        CLRR  MVA   X8,X7; MS         : CLEAR REG, HALF OR ALL
              QBYT                    : ALLOW BYTE OPERATION

              ADD CARRY INSTRUCTION EXECUTION
              REGISTER DESTINATION
0115  7E070BF0C080
        ADCR  ADD   X11,X12,X7; US,ULO : ADD CRY TO OPR
              QBYT                    : ALLOW BYTE OPERATION

              NEGATE INSTRUCTION EXECUTION
              REGISTER DESTINATION

0116  7E2708F0B080
        NEGR  SUB   X8,X11,X7; USLO   : SUBTRACT FROM ZERO
              QBYT                    : ALLOW BYTE OPERATION

              ONES COMPLEMENT INSTRUCTION EXECUTION
              REGISTER DESTINATION

0117  7C8716800080
        COMR  OCA   RR,X7; MS         : 1S COM OF DATA TO REG.
              QBYT                    : ALLOW BYTE OPERATION

              BIT CLEAR INSTRUCTION EXECUTION
              REGISTER DESTINATION

0118  7CE716800080
        BICR  NDB   RR,X8,X7; MS      : CLEAR BITS BY B
              QBYT                    : ALLOW BYTE OPERATION

              BRANCH IF CARRY SET INSTRUCTION EXECUTION
```

```
0119  70501000011F
        BCS    MVA    DR,DR            ; CANCEL MODIFY NXT
               BRU    NOS              ; IF NO BR, FETCH NXT
                                         BUT DO NXT FIRST
011A  B8DF15000001
               AND    PS,'%1           ; TEST CRY = 1
               SZRDF                   ; IF EQUAL 1, BRANCH

               SET CONDITION CODES INSTRUCTION EXECUTION

011B  704812000013
        SCC    MVA    ER,X8            ; COND. TO BE SET TO X8
               BRU    FNI              ; GO TO FETCH NXT AFTER
                                         DOING NXT
011C  7D5D15008000
               ORI    PS,X8,MSR        ; COND. TO MICRO STATUS

               CLEAR CONDITION CODES INSTRUCTION EXECUTION

011D  704812000013
        CCC    MVA    ER,X8            ; COND. TO BE RESET TO X8
               BRU    FNI              ; GO TO FETCH NXT AFTER
                                         DOING NXT
011E  7CFD15008000
               NDB    PS,X8,MSR        ; COND. TO MICRO STATUS

               NO BRANCH EXIT TO BRANCH INSTRUCTIONS

011F  7C514700006A
        NOS    MVA    PC,AR            ; NO BRANCH - FETCH
               SKPNX                   ; NEXT INSTRUCTION
               RNI; INQ; REDI

               BRANCH INSTRUCTION EXECUTION

0120  7C5014800021
        BR     MVA    IR,DR; MS        ; BRANCH TAKEN - MOVE
               BYTE                    ; BRANCH VECTOR TO DR
0121  FF0810004000
               SLLO   DR,X8            ; ADJUST VECTOR
0122  7C3F28000021
               SXA    *X8              ; SIGN EXTEND VECTOR
               BYTE
0123  7E1167000006A
               ADD    .*PC,X8,.AR      ; ADD VECTOR TO PC
               SKPNX                   ; FETCH NEW INSTR
               RNI; INQ; REDI

               RTS INSTRUCTION EXECUTION

0124  7C6700007000
        RTS    MVB    ,X7,PC           ; RTN ADDR TO PC
```



```
0125  7C5106000002
        RTM    MVA    X6,AR            ; GET STACK ADDR
               REDD                    ; READ TOP OF STK
0126  729F26110013
               ADT    *X6              ; UPDATE STK PTR
               BRU    FNI              ; GO FETCH NXT INSTR
               MXD                     ; AFTER DOING NXT
                                         MODIFY DSTN REG
0127  7C4716000000
               MVA    RR,X7            ; STACK DATA TO REG

               ROTATE LEFT INSTRUCTION EXECUTION
               REGISTER DESTINATION

0128  73104C408129
        ROLR   SRLO   X12,DR; UL       ; GET CARRY CTATUS
               BRU    ROLG             ; ABORT FETCH EXIT
               SKPNX
0129  73FF00010013
        ROLG   BRU    FNI              ; GO TO FETCH NXT
               MXD                     ; MODIFY DSTN OF NXT
012A  FF0716E04080
               SLLL   RR,X7; US, UO    ; ROTATE LEFT + CARRY
               QBYT                    ; ALLOW BYTE OPERATION

               ROTATE RIGHT INSTRUCTION EXECUTION
               REGISTER DESTINATION

012B  73104C40812C
        RORR   SRLO   X12,DR; UL       ; GET CARRY CTATUS
               BRU    RORG             ; ABORT FETCH EXIT
               SKPNX
012C  73FF00010013
        RORG   BRU    FNI              ; GO FETCH NEXT INSTR
               MXD                     ; MODIFY DSTN OF NXT
012D  FF0736E08080
               SRLL   *RR,X7; US, UO   ; ROTATE RIGHT, + CARRY
               QBYT                    ; ALLOW BYTE OPERATION

               ARITHMETIC SHIFT RIGHT INSTRUCTION EXECUTION
               REGISTER DESTINATION

012E  FF070BF0C080
        ASRR   SRAS   X11,X7; US,ULO   ; DO ARITH RIGHT SHFT
               QBYT                    ; ALLOW BYTE OPERATION

               ARITHMETIC SHIFT LEFT INSTRUCTION EXECUTION
               REGISTER DESTINATION

012F  FF070BF04080
        ASLR   SLLO   X11,X7; US, ULO  ; DO ARITH LEFT SHFT
               QBYT                    ; ALLOW BYTE OPERATION

               BRANCH IF LESS THAN INSTRUCTION EXECUTION
```

```
0130  705010000011F
        BLT   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0131  B8DF1D002000
              AND   MSR,'%2000     : TEST N XOR V = 0
              SZRDF                : IF EQUAL 1, BRANCH

              BIT SET INSTRUCTION EXECUTION
              REGISTER DESTINATION

0132  7D4716808080
        BISR  ORI   RR,X8,X7; MS   : OR SRC WITH DSTN DATA
              QBYT                 : ALLOW BYTE OPERATION

              BRANCH IF LESS OR EQUAL INSTRUCTION EXECUTION

0133  705010000011F
        BLE   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0134  B8DF1D003000
              AND   MSR,'%3000     : TEST(Z OR(N XOR V)=1)
              SZRDF                : IF EQUAL 1, BRANCH

              BRANCH IF CARRY CLEAR INSTRUCTION EXECUTION

0135  705010000011F
        BCC   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0136  B0DF15000001
              AND   PS,'%1         : TEST CRY = 0
              SZRDT                : IF EQUAL 0, BRANCH

              BRANCH IF OVERFLOW SET INSTRUCTION EXECUTION

0137  705010000011F
        BVS   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0138  B8DF15000002
              AND   PS,'%2         : TEST OVF = 1
              SZRDF                : IF EQUAL 1, BRANCH

              BRANCH IF LOWER OR SAME INSTRUCTION EXECUTION

0139  705010000011F
        BLOS  MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
013A  B8DF15000005
              AND   PS,'%5         : TEST C OR V = 1
              SZRDF                : IF EQUAL 1, BRANCH

              BRANCH IF OVERFLOW CLEAR INSTRUCTION EXECUTION
```



```
013B  705010000011F
        BVC   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
013C  B0DF15000002
              AND   PS,'%2         : TEST OVF = 0
              SZRDT                : IF EQUAL 0, BRANCH

              BRANCH IF HIGHER INSTRUCTION EXECUTION

013D  705010000011F
        BHI   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
013E  B0DF15000005
              AND   PS,'%5         : TEST C OR V = 0
              SZRDT                : IF EQUAL 0, BRANCH

              JUMP INSTRUCTION EXECUTION

013F  7C4751000006A
        JMP   MVA   AR,PC          : NEW ADDR TO PC
              SKPNX                : SKIP REGULAR FETCH
              RNI; INQ; REDI       : SET RNI INTRP, CHK
                                     INTRP INQ, READ NXT
                                     INSTRUCTION

              BRANCH IF MINUS INSTRUCTION EXECUTION

0140  705010000011F
        BMI   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0141  B8DF15000008
              AND   PS,'%8         : TEST NEG = 1
              SZRDF                : IF EQUAL 1, BRANCH

              BRANCH IF PLUS INSTRUCTION EXECUTION

0142  705010000011F
        BPL   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
0143  B0DF15000008
              AND   PS,'%8         : TEST NEG = 0
              SZRDT                : IF EQUAL 0, BRANCH

              BRANCH IF EQUAL INSTRUCTION EXECUTION

0144  705010000011F
        BEQ   MVA   DR,DR          : CANCEL MODIFY NXT
              BRU   NOS            : IF NO BR, FETCH NXT
                                     BUT DO NXT FIRST
```

```
0145  B8DF15000004
              AND    PS,'%4              : TEST ZERO SET
              SZRDF                      : IF = 0, BRANCH

              BRANCH IF GREATER OR EQUAL INSTRUCTION EXECUTION

0146  70501000011F
        BGE   MVA    DR,DR               : CANCEL MODIFY NXT
              BRU    NOS                 : IF NO BR, FETCH NXT
                                           BUT DO NXT FIRST
0147  B0DF1D002000
              AND    MSR,'%2000          : TEST N XOR V = 0
              SZRDT                      : IF EQUAL 0, BRANCH

              DECREMENT INSTRUCTION EXECUTION
              REGISTER DESTINATION

0148  7E270B900080
        DECR  SUB    X11,X8,X7; PL1, US,UO   : DECR DSTN BY 1
              QBYT                       : ALLOW BYTE OPERATION

              SUBTRACT CARRY INSTRUCTION EXECUTION
              REGISTER DESTINATION

0149  7E270BF0C080
        SBCR  SUB    X11,X12,X7; US,ULO  : SUB CRY TO OPR
              QBYT                       : ALLOW BYTE OPERATION

              BRANCH IF GREATER THAN INSTRUCTION EXECUTION

014A  70501000011F
        BGT   MVA    DR,DR               : CANCEL MODIFY NXT
              BRU    NOS                 : IF NO BR, FETCH NXT
                                           BUT DO NXT FIRST
014B  B0DF1D003000
              AND    MSR,'%3000          : TEST(Z OR(N XOR V)=0)
              SZRDT                      : IF EQUAL 0, BRANCH

              ADD INSTRUCTION EXECUTION
              REGISTER DESTINATION

014C  7E070BF00000
        ADDR  ADD    X11,X8,X7; USLO     : ADD SRC TO DSTN DATA

              SUBTRACT INSTRUCTION EXECUTION
              REGISTER DESTINATION

014D  7E270BF00000
        SUBR  SUB    X11,X8,X7; USLO     : SUB SRC FROM DSTN

              INCREMENT INSTRUCTION EXECUTION
              REGISTER DESTINATION
```



```
014E  7E070B900080
        INCR  ADD    X11,X8,X7; PL1, US,UO   : INCR DSTN BY 1
              QBYT                       : ALLOW BYTE OPERATION

              BRANCH IF NOT EQUAL INSTRUCTION EXECUTION

014F  70501000011F
        BNE   MVA    DR,DR               : CANCEL MODIFY NXT
              BRU    NOS                 : IF NO BR, FETCH NXT
                                           BUT DO NXT FIRST
0150  B0DF15000004
              AND    PS,'%4              : TEST ZERO SET
              SZRDT                      : IF = 1, BRANCH

              ONES COMPLEMENT INSTRUCTION EXECUTION - COM
              BUS DESTINATION

0151  738F00000013
        COM   CMA                        : READ DSTN DATA
              BRU    FNI                 : GO TO FETCH NXT, DO NXT FIRST
0152  7C9016800094
              OCA    RR,DR; MS           : ONES COMPLEMENT DATA
              QBYT; WRT*                 : WORD OR BYTE, WRITE IN DSTN

              NEGATE - TWOS COMPLEMENT INSTRUCTION EXECUTION - NEG
              BUS DESTINATION

0153  FFBF00000000
        NEG   CMA                        : READ THE DSTN DATA
0154  704B16000013
              MVA    RR,X11              : PUT DSTN DATA IN FILE
              BRU    FNI                 : GO TO FETCH NXT, DO NXT FIRST
0155  7E3008F0B094
              SUB    X8,X11,DR; USLO     : TWOS COMPLEMENT THE DATA
              QBYT; WRT*                 : WORD OR BYTE, WRITE IN DSTN

              ADD CAPRY INSTRUCTION EXECUTION - ADC
              BUS DESTINATION

0156  738F00000013
        ADC   CMA                        : READ THE DSTN DATA
              BRU    FNI                 : GO TO FETCH NXT, DO NXT FIRST
0157  7E1016F0C094
              ADD    RR,X12,DR; USLO     : ADD CARRY TO DATA
              QBYT; WRT*                 : WORD OR BYTE, WRITE IN DSTN

              SUB CARRY INSTRUCTION EXECUTION - SBC
              BUS DESTINATION

0158  738F00000013
        SBC   CMA                        : READ THE DSTN DATA
              BRU    FNI                 : GO TO FETCH NXT, DO NXT FIRST
```

```
0159  7E3016F0C094
            SUB   RR,X12,DR; USLO   : SUB CARRY TO DATA
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            TEST INSTRUCTION EXECUTION - TST

015A  738F00000013
      TST   CMA                     : READ THE DSTN DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
015B  7C5016800080
      TSTR  MVA   RR,DR; MS         : SET STATUS OF DATA
            QBYT                    : WORD OR BYTE

            INCREMENT INSTRUCTION EXECUTION - INC
            BUS DESTINATION

015C  738F00000013
      INC   CMA                     : READ THE DSTN DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
015D  7E1016908094
            ADD   RR,X8,DR; PL1, USO  : ADD 1 TO DATA
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            DECREMENT INSTRUCTION EXECUTION - DEC
            BUS DESTINATION

015E  738F00000013
      DEC   CMA                     : READ THE DSTN DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
015F  7E3016908094
            SUB   RR,X8,DR; PL1, USO  : SUB 1 FROM DATA
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            ARITHMETIC SHIFT RIGHT INSTRUCTION EXECUTION - ASR
            BUS DESTINATION

0160  738F00000013
      ASR   CMA                     : READ THE DSTN DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
0161  FF1016F0C094
            SRAS  RR,DR; US, ULO    : SHIFT DATA RIGHT 1 BIT
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            ARITHMETIC SHIFT LEFT INSTRUCTION EXECUTION - ASL
            BUS DESTINATION

0162  738F00000013
      ASL   CMA                     : READ THE DSTN DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
0163  FF1016F04094
            SLLO  RR,DR; USLO       : SHIFT DATA LEFT 1 BIT
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            ROTATE LEFT INSTRUCTION EXECUTION - ROL
            BUS DESTINATION
```



```
0164  FFBF80003000
      ROL   CMA                     : READ THE DSTN DATA
0165  73100C408013
            SRLO  X12,DR; UL        : SHIFT CARRY INTO LINK
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
0166  FF1016E04094
            SLLL  RR,DR; USO        : ROTATE DATA LEFT 1 BIT
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            ROTATE RIGHT INSTRUCTION EXECUTION - ROR
            BUS DESTINATION

0167  FFBF80000000
      ROR   CMA                     : READ THE DSTN DATA
0168  73100C408013
            SRLO  X12,DR; UL        : SHIFT CARRY INTO LINK
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
0169  FF1016E08094
            SRLL  RR,DR; USO        : ROTATE DATA RIGHT 1 BIT
            QBYT; WRT*              : WORD OR BYTE, WRITE IN DSTN

            EXECUTE MOV INSTRUCTION - BUS DESTINATION

016A  7C5008800094
      MOV   MVA   X8,DR; MS         : MOVE SRC DATA TO DR
            QBYT; WRT*              : WRT SRC TO DSTN
016B  73FF40000013
            BRU   FNI               : GO TO FETCH NXT
            SKPNX                   : SKIP EXEC. OF NXT CLOCK

            EXECUTE BIS INSTRUCTION - BUS DESTINATION

016C  738F00000013
      BIS   CMA                     : READ DSTN OPR DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
016D  7D5016800094
            ORI   RR,X8,DR; MS      : SET BITS SPECIFIED BY SRC
            QBYT; WRT*              : WRT SRC TO DSTN

            EXECUTE BIT INSTRUCTION

016E  738F00000013
      BIT   CMA                     : READ DSTN OPR DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
016F  7CD016800080
      BITR  AND   RR,X8,DR; MS      : TEST BITS SPECIFIED BY SRC
            QBYT                    : BYTE OR WORD

            EXECUTE BIC INSTRUCTION - BUS DESTINATION

0170  738F00000013
      BIC   CMA                     : READ DSTN OPR DATA
            BRU   FNI               : GO TO FETCH NXT, DO NXT FIRST
```

```
0171  7CF016808094
                NDB   RR,X8,DR; MS     : CLEAR BITS SPECIFIED BY SRC
                QBYT; WRT*             : WRT SRC TO DSTN

                EXECUTE CMP INSTRUCTION

0172  FFBF00000000
        CMP     CMA                    : READ DSTN OPR DATA
0173  704B16000013
                MVA   RR,X11           : MOVE DSTN DATA TO X11
                BRU   FNI              : GO TO FETCH NXT

                COMPARE INSTRUCTION EXECUTION
                REGISTER DESTINATION

0174  7E3008F0B080
        CMPR    SUB   X8,X11,DR; USLO  : SUB DSTN FROM SRC
                QBYT                   : ALLOW BYTE MODE

                EXECUTE ADD INSTRUCTION - BUS DESTINATION

0175  73BF00000013
        ADD     CMA                    : READ DSTN OPR DATA
                BRU   FNI              : GO TO FETCH NXT, DO NXT FIRST
0176  7E1016F03004
                ADD   RR,X8,DR; USLO   : ADD SRC AND DSTN DATA
                WRT                    : WRITE TO DSTN

                SWAP INSTRUCTION EXECUTION - SWAP
                REGISTER DESTINATION

0177  730776000012
        SWAR    SWAB  *RR,X7           : SWAP UPPER/LOWER BYTES
                BRU   SCU1             : GO UPDATE SWAP CONDITION CODES
                SKPNX                  : SKIP NXT INSTR


                FATAL INTERRUPT DETECT

0178  777F9D00A189
        CPUX    TSBT  MSR,#10,PIXX     : TEST ODD ADDR ERROR
                DNXIB                  : IF YES, DO NXT AND BRANCH
0179  FC73000004E0
                MVB   ,'%4E0,SL        : RESET ADDR ERROR/SLV SYNC
017A  777F9D007091
        RENT    TSBT  MSR,#7,SKE       : TEST RED STACK LIMIT
                DNXIB                  : IF YES, DO NXT AND BRANCH
017B  FC6800000004
                MVB   ,'%4,X8          : SET STK ADDR/TRAP ADDR
017C  7F7FDD006186
                TSBF  MSR,#6,DSY       : TEST MMU ERROR
                SKPIB                  : SKIP NXT IF NO MMU ERROR
```



```
017D  FC7300000400
                MVB   ,'%400,SL        : RESET MMU ERROR IND.
017E  FC68000000A8
                MVB   ,'%A8,X8         : SET MMU TRAP VECTOR
017F  2C5FCF000183
                MVA   X15
                BOVSF MUO              : DID MMU ERROR OCCUR IN
                SKPIB                  : DM3 OR DM5, IF NOT, GO TO MUO
0180  FCDF32007FC0
                AND   *ER,'%7FC0       : MASK CURRENT INSTR.
0181  B9DF12000D80
                XOR   ER,'%0D80        : IS INSTR AN MTPI
                SZRDF                  : IF NOT, SKIP NXT
0182  729F66100048
                ADT   *X6              : UPDATE R6
                BRU   EXCH             : GO TO STACK PROCESSOR
                SKPNX                  : SKIP NEXT INSTR
0183  485FCE00004B
        MUO     MVA   X14              : EXAMINE MMU ERROR ORIGIN
                BNGDF EXCH             : IF NOT IN STCK PS/PC GO TO STACK
                SKPIB                  : AND SKIP NEXT
0184  70560C00004C
                MVA   X12,RR           : SAVE OLD PSW
                BRU   EXCH+#1          : ENTER STACK PSW/PC
0185  FCDF2E007FFF
                AND   *X14,'%7FFF      : REMOVE ORIGIN FLAG
0186  777F9D000092
        DSY     TSBT  MSR,#8,SKE1      : TEST DOUBLE SYNC ERROR
                DNXIB                  : IF YES, DO NXT AND BRANCH
0187  FC6800000004
                MVB   ,'%4,X8          : SET STK ADDR/TRAP ADDR
0188  7F7FDD009015
                TSBF  MSR,#9,PWP       : TEST SLAVE SYNC ERROR
                SKPIB                  : IF NOT, SKIP NXT
0189  605F06000092
        PIXX    MVA   X6               : TEST STK FOR ODD ADDR
                BODDT SKE1             : IF ODD, DO RED STACK
018A  FC6800000004
        PIX     MVB   ,'%4,X8          : SET TRAP VECTOR ADDR
018B  73FF40000048
                BRU   EXCH             : GO TO TRAP STACK
                SKPNX                  : SKIP NXT
```

```
      D0FF
                .QET  %D0FF

                EMULATE TABLE CONTENTS


0001
                .ORE  %1
0001  1109
                .DAT  %1000+MOVR
0002  0174
                .DAT  CMPR
0003  116F
                .DAT  %1000+BITR
0004  1118
                .DAT  %1000+BICR
0005  1132
                .DAT  %1000+BISR
0006  014C
                .DAT  ADDR
0009
                .ORE  %9
0009  1589
                .DAT  %1400+MOVR
000A  0574
                .DAT  %400+CMPR
000B  156F
                .DAT  %1400+BITR
000C  1518
                .DAT  %1400+BICR
000D  1532
                .DAT  %1400+BISR
000E  014D
                .DAT  SUBR
0011
                .ORE  %11
0011  196A
                .DAT  %1800+MOV
0012  0172
                .DAT  CMP
0013  116E
                .DAT  %1000+BIT
0014  1970
                .DAT  %1800+BIC
0015  196C
                .DAT  %1800+BIS
0016  0975
                .DAT  %800+ADD
0019
                .ORE  %19
0019  1D6A
                .DAT  %1C00+MOV
001A  0572
                .DAT  %400+CMP
```



```
001C  1D70
                .DAT  %1C00+BIC
001D  1D6C
                .DAT  %1C00+BIS
001E  081C
                .DAT  %800+SUB
0024
                .ORE  %24
0024  1313
                .DAT  %1200+XORR
0027
                .ORE  %27
0027  F30A
                .DAT  %F200+SOB
002C
                .ORE  %2C
002C  1B11
                .DAT  %1A00+XOR
002F
                .ORE  %2F
002F  F30A
                .DAT  %F200+SOB
0030
                .ORE  %30
0030  D18A
                .DAT  %D000+PIX
0031  D18A
                .DAT  %D000+PIX
0032  F103
                .DAT  %F000+EMT
0033  F107
                .DAT  %F000+TRAP
0034  F07A
                .DAT  %F000+JSR
0035  F07A
                .DAT  %F000+JSR
0036  F103
                .DAT  %F000+EMT
0037  F107
                .DAT  %F000+TRAP
0048
                .ORE  %48
0048  5514
                .DAT  %5400+CLRR
0049  3517
                .DAT  %3400+COMR
004A  454E
                .DAT  %4400+INCR
004B  4548
                .DAT  %4400+DECR
004C  0516
                .DAT  %400+NEGR
004D  0515
                .DAT  %400+ADCR
```

```
004E  0549          .DAT  %400+SBCR
004F  2558          .DAT  %2400+TSTR
0050  0528          .DAT  %400+RORR
0051  0528          .DAT  %400+ROLR
0052  052E          .DAT  %400+ASRR
0053  052F          .DAT  %400+ASLR
0054  F274          .DAT  %F200+MARK
0055  E861          .DAT  %E800+MFPR
0056  E871          .DAT  %E800+MTPR
0057  9310          .DAT  %9200+SXR
0068                .ORE  %68
0068  5D6A          .DAT  %5C00+MOV
0069  3D51          .DAT  %3C00+COM
006A  4D5C          .DAT  %4C00+INC
006B  4D5E          .DAT  %4C00+DEC
006C  0D53          .DAT  %C00+NEG
006D  0D56          .DAT  %C00+ADC
006E  0D58          .DAT  %C00+SBC
006F  255A          .DAT  %2400+TST
0070  0D67          .DAT  %C00+ROR
0071  0D64          .DAT  %C00+ROL
0072  0D60          .DAT  %C00+ASR
0073  0D62          .DAT  %C00+ASL
0074  F274          .DAT  %F200+MARK
0075  E85D          .DAT  %E800+MFPI
0076  E864          .DAT  %E800+MTPI
0077  980E          .DAT  %9A00+SXT
```



```
0081                .ORE  %81
0081  F120          .DAT  %F000+BR
0082  F14F          .DAT  %F000+BNE
0083  F144          .DAT  %F000+BEQ
0084  F146          .DAT  %F000+BGE
0085  F130          .DAT  %F000+BLT
0086  F14A          .DAT  %F000+BGT
0087  F133          .DAT  %F000+BLE
0088  F142          .DAT  %F000+BPL
0089  F140          .DAT  %F000+BMI
008A  F13D          .DAT  %F000+BHI
008B  F139          .DAT  %F000+BLOS
008C  F13B          .DAT  %F000+BVC
008D  F137          .DAT  %F000+BVS
008E  F135          .DAT  %F000+BCC
008F  F119          .DAT  %F000+BCS
00A4                .ORE  %A4
00A4  D18A          .DAT  %D000+PIX
00A5  D18A          .DAT  %D000+PIX
00A6  D18A          .DAT  %D000+PIX
00A7  D18A          .DAT  %D000+PIX
00A8  F124          .DAT  %F000+RTS
00AC                .ORE  %AC
00AC  2177          .DAT  %2000+SWAR
00B4                .ORE  %B4
00B4  F13F          .DAT  %F000+JMP
00B5  F13F          .DAT  %F000+JMP
```



```
00B6  F13F          .DAT  %F000+JMP
00B7  F13F          .DAT  %F000+JMP
00BA                .ORE  %BA
00BA  011D          .DAT  CCC
00BB  0118          .DAT  SCC
00BC  2810          .DAT  %2800+SWAB
00BD  2810          .DAT  %2800+SWAB
00BE  2810          .DAT  %2800+SWAB
00BF  2810          .DAT  %2800+SWAB
00C0                .ORE  %C0
00C0  F01E          .DAT  %F000+HLTP
00C1  F101          .DAT  %F000+WAIT
00C2  F082          .DAT  %F000+RTI
00C3  F0A1          .DAT  %F000+BKP
00C4  F105          .DAT  %F000+IOT
00C5  F0AF          .DAT  %F000+RSTC
00C6  F081          .DAT  %F000+RTT
                    UPPER INTERRUPT ROM TABLE
0000                .ORD  %0
0000  0020          .DAT  -HALT
0001  0020          .DAT  -HALT
0002  0020          .DAT  -HALT
0003  0020          .DAT  -HALT
0004  0020          .DAT  -HALT
0005  0020          .DAT  -HALT
0006  0020          .DAT  -HALT
0007  0020          .DAT  -HALT
```

```
0008  0020          .DAT  -HALT
0009  0020          .DAT  -HALT
000A  0020          .DAT  -HALT
000B  0020          .DAT   HALT
000C  0020          .DAT  -HALT
000D  0020          .DAT   HALT
000E  0020          .DAT  -HALT
000F  0020          .DAT   HALT
0010  00A2          .DAT   TBIT
0011  00A2          .DAT   TBIT
0012  00A2          .DAT   TBIT
0013  00A2          .DAT   TBIT
0014  00A2          .DAT   TBIT
0015  00A2          .DAT   TBIT
0016  00A2          .DAT   TBIT
0017  00A2          .DAT   TBIT
0018  00A3          .DAT   YSL
0019  00A3          .DAT   YSL
001A  00A3          .DAT   YSL
001B  00A3          .DAT   YSL
001C  0088          .DAT   PFI
001D  0088          .DAT   PFI
001E  0000          .DAT   %0000
001F  FFFF          .DAT   %FFFF
0000                .ORD   %0
      0088          .EQU   SMZ
      008D          .EQU   SM1
```



```
      00C0          .EQU   SM2
      00C2          .EQU   SM3
      00C5          .EQU   SM4
      00C7          .EQU   SM5
      00CA          .EQU   SM6
      00CD          .EQU   SM7
0000                .ORD   %0
      00D0          .EQU   DMZ
      00D2          .EQU   DM1
      00D8          .EQU   DM2A
      00E0          .EQU   DM3A
      00E6          .EQU   DM4A
      00ED          .EQU   DM5A
      00F1          .EQU   DM6A
      00FC          .EQU   DM7A
0000                .ORD   %0
      00D0          .EQU   DMZ
      00D2          .EQU   DM1
      00D4          .EQU   DM2
      00DD          .EQU   DM3
      00E3          .EQU   DM4
      00E9          .EQU   DM5
      00F4          .EQU   DM6
      00F8          .EQU   DM7
0000                .END
```