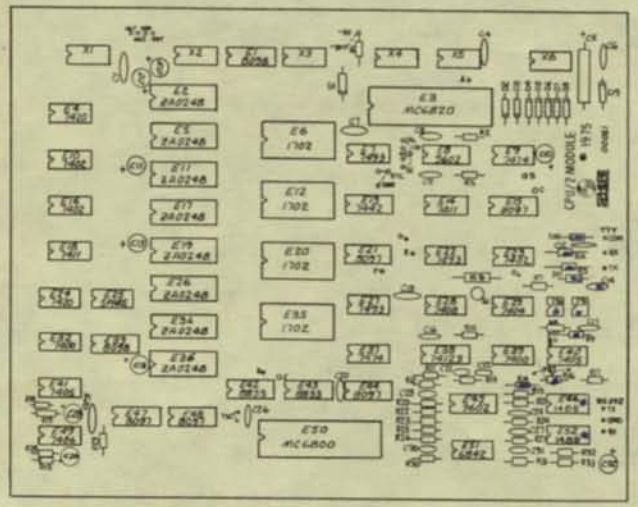
ZONE	LTR	DESCRIPTION	DATE	BY	APPVD
		REVISIONS			



REF	REF	QTY		SCHEMATIC	D000019		
1	1	60		PRINTED CIRCUIT BOARD	D000020		
1	1	39		TRANSISTOR	2N2369A	Q1	
2	2	38		CAPACITOR CERAMIC 68pf 1KV	RP-680	C29,30	CRL
3	3	61		CAPACITOR CERAMIC 330pf 1KV	RD-107	C21,27,31	CRL
3	3	36		CAPACITOR CERAMIC 1000pf 1KV	RDM-102	C17,22,23	CRL
2	2	35		CAPACITOR CERAMIC .01uf 150V	RDM-103	C16,26	CRL
1	1	64		CAPACITOR TANTALUM 100uf 10VDC		C9	MEMCOR
5	5	63		CAPACITOR .47uf 16 VDC	RX47.16	C3,5/10/18	12,29,32 RAY
12	11	62		CAPACITOR CERAMIC .1uf 50V	CK-104	C1-A-3/4-6,8	11,20,25 CRL
2	2	51		RESISTOR 4.7K Ω ¼ W	RC07GF472E	R24,27	
2	2	50		RESISTOR 10 Ω ¼ W	RC07GF100	R23,26	
2	2	49		RESISTOR 22 Ω ¼ W	RC07GF220	R22,25	
3	3	48		RESISTOR 10K Ω ¼ W	RC07GF103	R1,28,30	
4	3	47		RESISTOR 1K Ω ¼ W	RC07GF102	R19,31,32	
1	-	46		RESISTOR 510 Ω ¼ W	RC07GF511	R16	
2	2	45		RESISTOR 470 Ω ¼ W	RC07GF471	R15,33	
1	1	44		RESISTOR	RC07GF	R14	
1	1	43		RESISTOR 33K Ω ¼ W	RC07GF333	R13	
1	1	42		RESISTOR 11K Ω ¼W 1%	RNC65H1102F	R12	
1	1	41		RESISTOR 2.74K Ω ¼W 1%	RNC65H2741F	R11	
1	1	40		RESISTOR 15K Ω ¼ W	RC07GF153	R10	
1	-	39		RESISTOR 2K Ω ¼W	RC07GF202	R9	
5	5	38		RESISTOR 3.3K Ω ¼ W	RC07GF332	R3,5,8,20,29	
1	1	37		RESISTOR 180 Ω 2W	RC42GF181	R6	
1	-	36		RESISTOR 820 Ω 1W	RC32GF821	R5	
1	-	35		RESISTOR 1.1K Ω 1W	RC32GF112	R4	
2	2	34		RESISTOR 34R Ω ¼ W	RC07GF340	R2,7	
1	1	33		RESISTOR 100 Ω ¼ W	RC07GF101	R1	
6	4	32		DIODE	IN4001	D5,6,7,8,10,11	
5	5	31		DIODE	IN914	D2,3,4,9,12	
1	1	30		ZENER DIODE 3V	1N5225B	D1	
8	8	29		22 PIN IC SOCKETS	CA 22-CW1190		CA
2	1	28		40 PIN IC SOCKETS	340-1639-D	E3,5B	AUGAT
2	4	27		24 PIN IC SOCKETS	324-1639-D	E4,7,20,35	AUGAT
6	4	26		14 PIN IC SOCKETS	314-AG39-D	E1-X6	AUGAT
1	-	25		QUAD RECEIVER - RS232	MC489	E37	
1	1	24		QUAD TRANSISTOR	MPQ6842N	E21	
1	1	23		MICROPROCESSOR	MC6800	E30	
1	1	22		QUAD EXCLUSIVE OR	SN7486N	E29	
1	-	21		QUAD TRANSMITTER-RS232	MC1488N	E46	
2	2	20		QUAD TRI-STATE TRANSCEIVER	DM8833N	E42,43	
2	2	19		HEX INVERTER W/OC	SN7406N	E40,41	
1	1	18		DUAL ONE SHOTS	SN74123N	E38	
2	2	17		QUAD 2-INPUT NAND GATE	SN7400N	E32,E39	
2	-	16		PMOD COUPLER	4N33N	E30,31	
1	1	15		HEX INVERTER	SN7404N	E29	
1	1	14		QUAD 2 INPUT AND GATE	SN7408N	E28	
-	-	13		SPARE		E27	
8	8	12		TRI-STATE HEX BUFFER	DM8097N	E15,24,44,47,48	
2	2	11		TRIPLE 3-INPUT AND GATE	SN7411N	E14,18	
1	1	10		4-LINE TO 16 LINE DECODER	SN74154N	E13	
3	3	9		QUAD 2-INPUT NOR GATE	SN7402N	E10,16,23	
2	2	8		DUAL D FLIP FLOP	SN7474N	E9,17	
2	2	7		DUAL ONE SHOT	9602 SN96L02N	E8,45	
3	3	6		4 BIT BINARY COUNTER	SN7493N	E12,22,27	
2	4	5		EPROM 256 X 8 BIT	MM1702A	E6,12,26,35	
2	2	4		8-INPUT NAND GATE	SN7430N	E19,24	
1	-	3		PERIPHERAL INTERFACE ADAPTER (PIA)	MC6820	E3	
9	8	2		4K DYNAMIC RAM MEMORY	2A0248	E2,5,11,17,34,36	
2	2	1		TRI-STATE HEX INVERTER	DM8098N	E31,33	
-OR	-01			ASSEMBLY			
QTY REQ'D	QTY CHGD	ITEM NO.	REV. LETTER	DESCRIPTION OR IDENTIFYING NO.	DESCRIPTION OR PART NO.	SPECIFICATION	MATERIAL OR NOTE

LIST OF MATERIAL OR PARTS

CONTRACT No.		SPHERE
DRAWN BY	G. W. / 22 DEC 75	CPU/2 ASSEMBLY
CHECKED		
MECH.		
ELECT.		
PROJ. ENGR.		
APPROVED Date	SIZE D	000021 REV
APPROVED (for Orders) Date	SCALE NONE	Sheet 1 OF 1

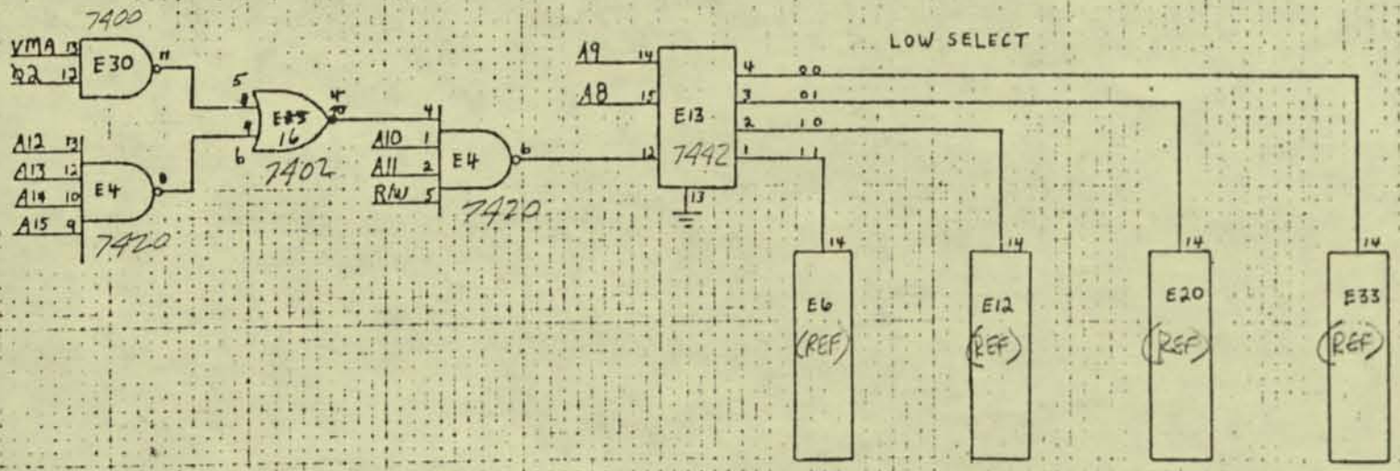
CPU

DATE
CHKD. BY ____ DATE ____

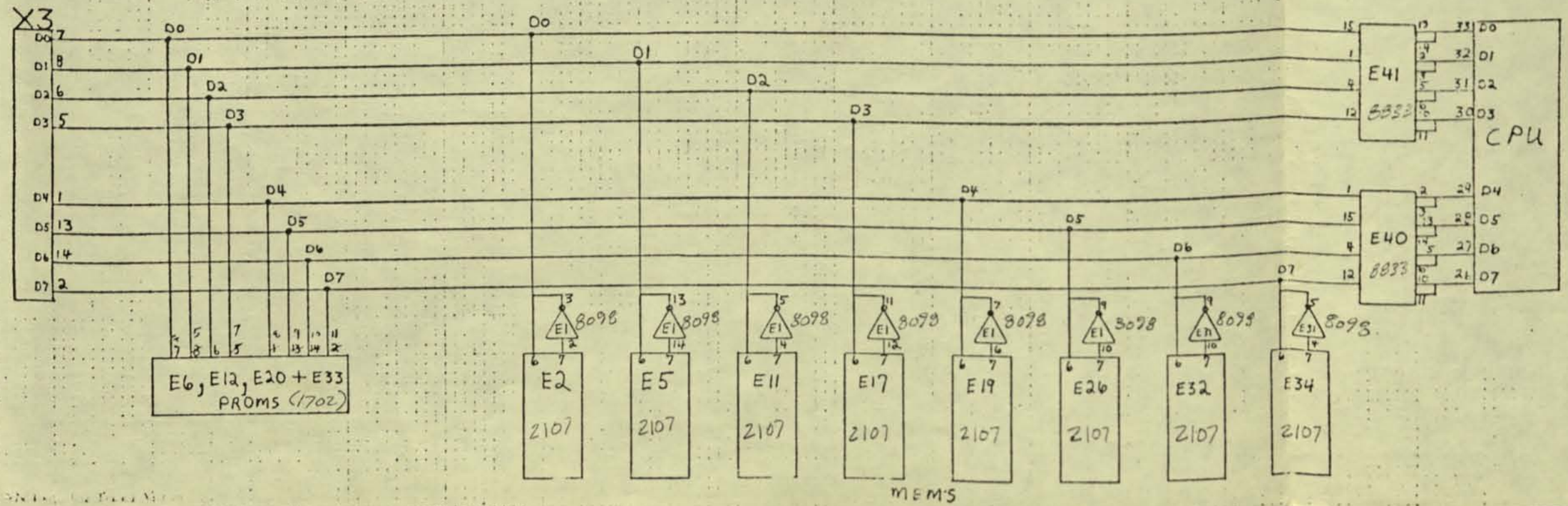
SUBJECT PROM SELECT
DATA DISTRIBUTION

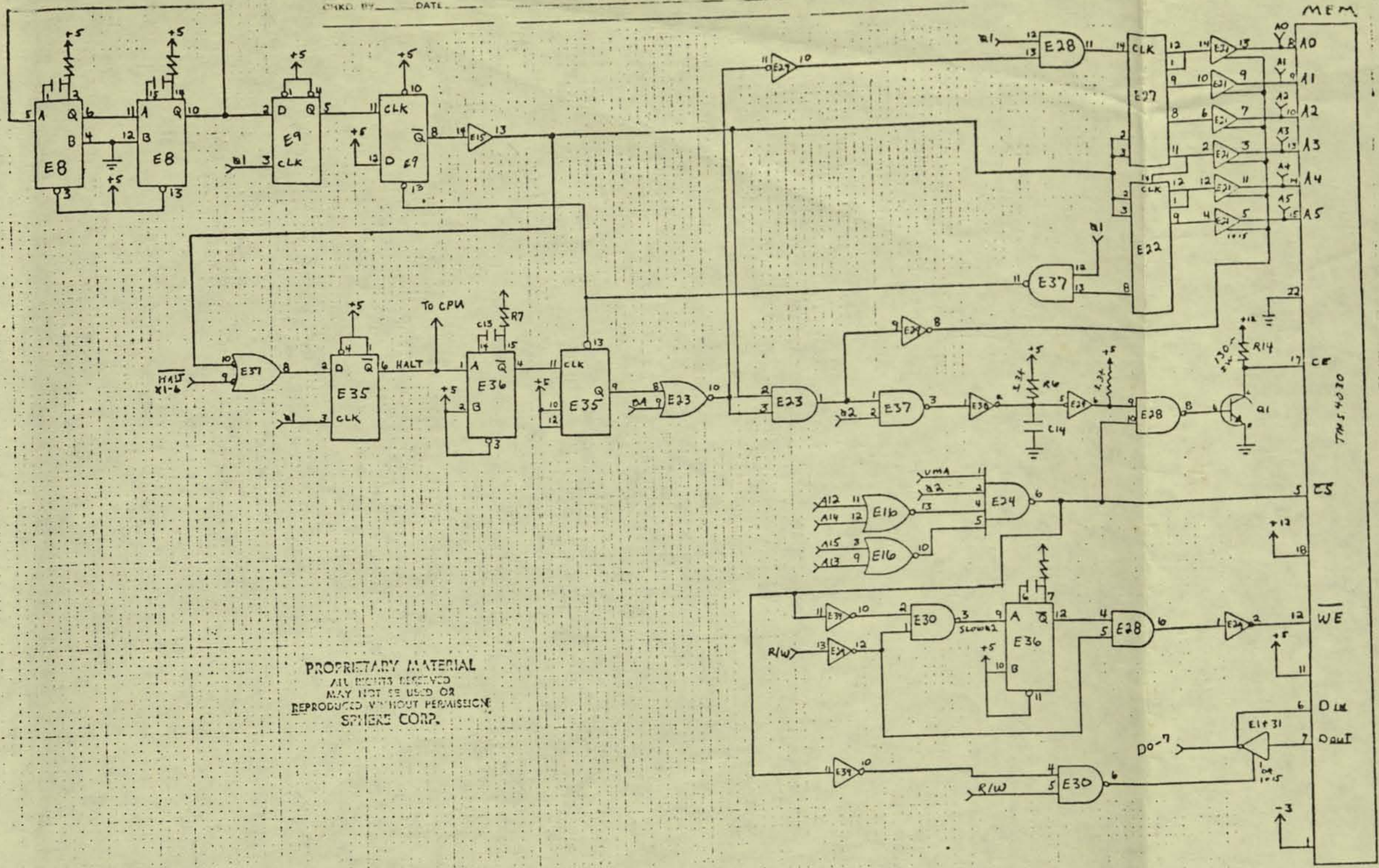
SHEET NO ____ OF ____
JOB NO. ____

CPU



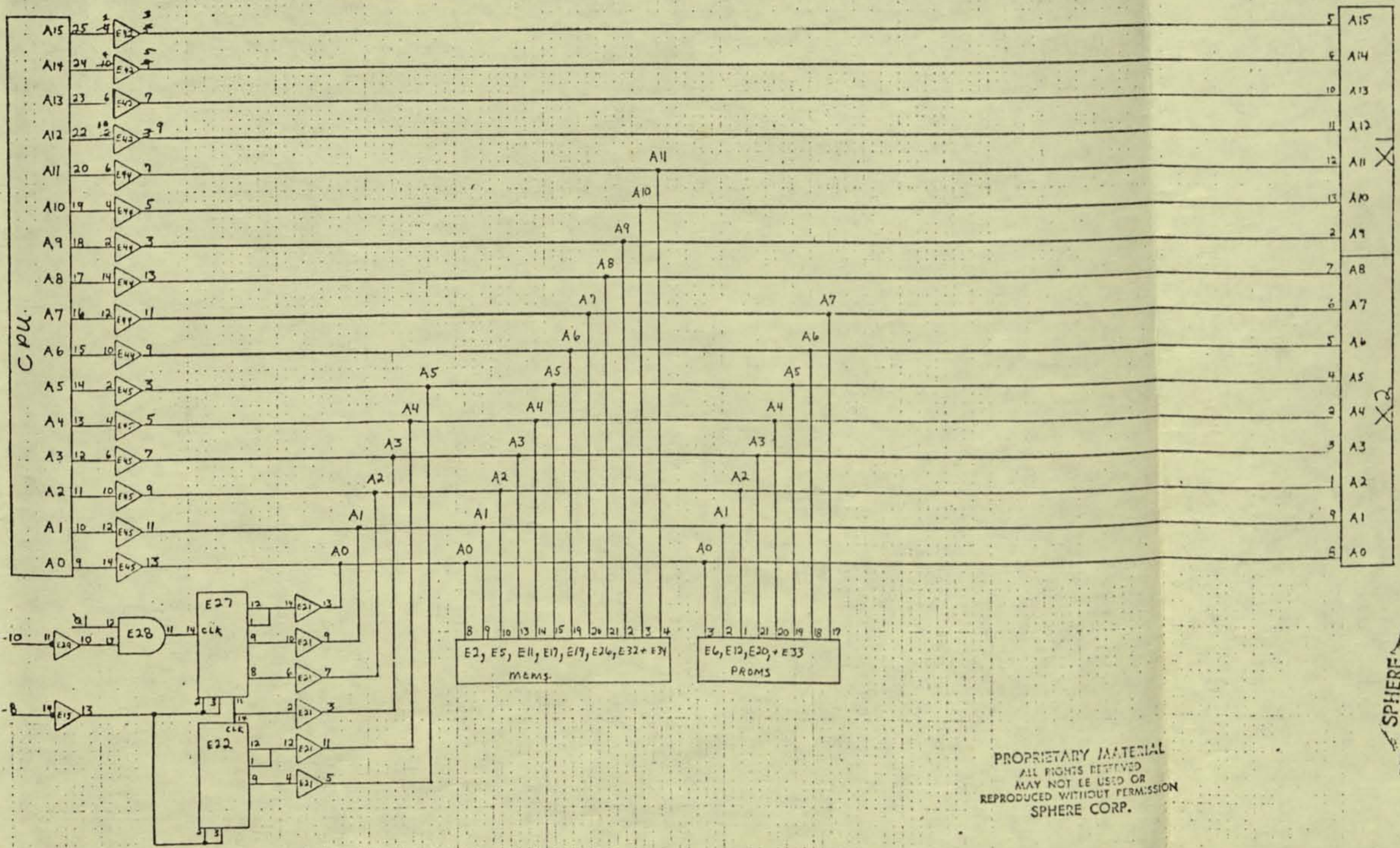
SPHERE
701 SOUTH 400 WEST



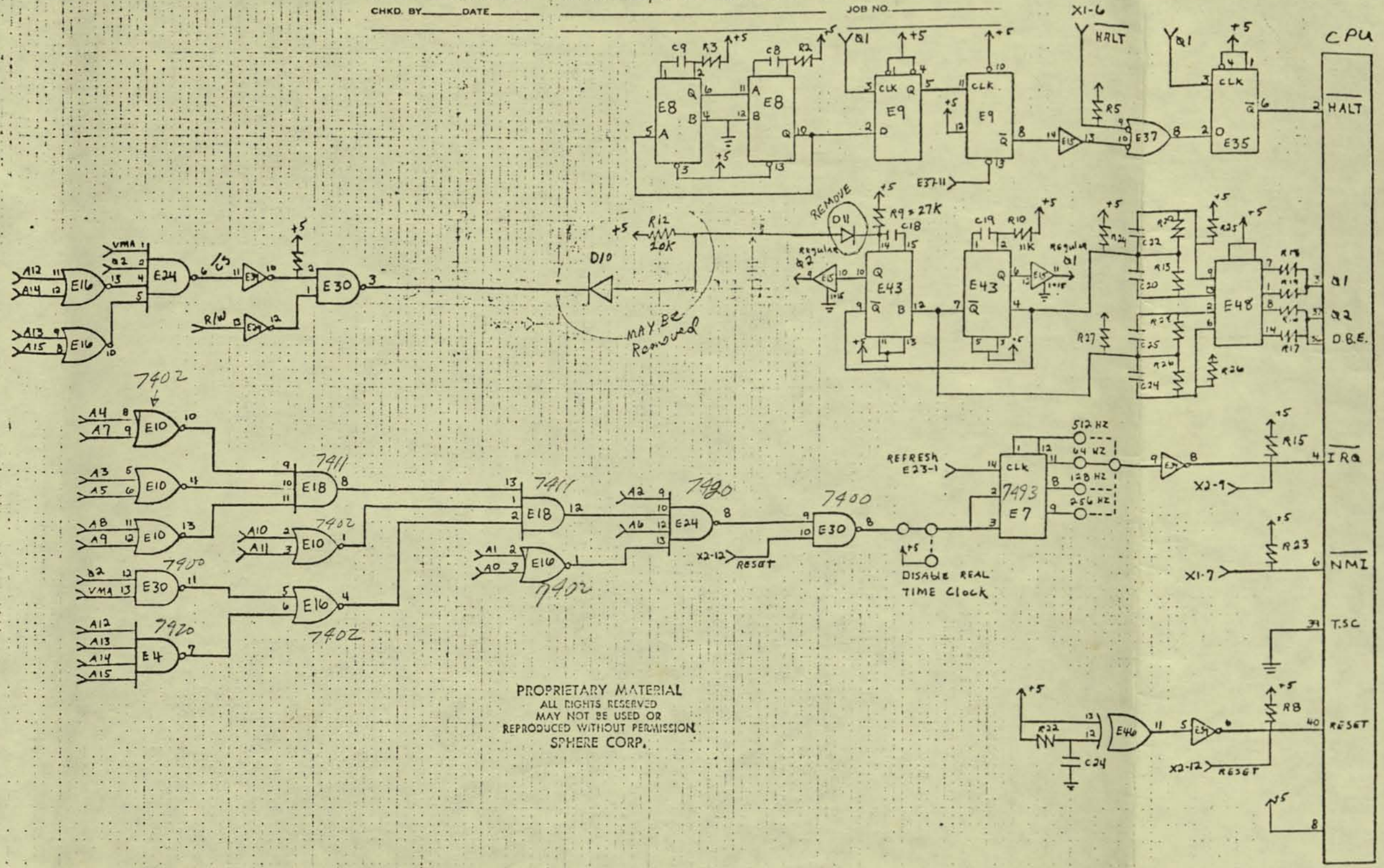

CHKD BY
DATE
MEM
E8
E8
E9
E9
E35
E35
E36
E36
E37
E37
E23
E23
E27
E22
E28
E28
E24
E16
E30
E31
E39
E15
E34
E38
E1+31
TO CPU
HALT
HALT
X1-6
R7
C13
R6
C14
R14
Q1
VMA
R/W
SLOW#2
D0-7
A0
A1
A2
A3
A4
A5
CS
WE
DBE
DBMI
PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.
SPHERE
1791 SOUTH 500 WEST

BY DLS DATE ___ SUBJECT ADDRESS ___ SHEET NO ___ OF ___

CHKD. BY ___ DATE ___ JOB NO. ___

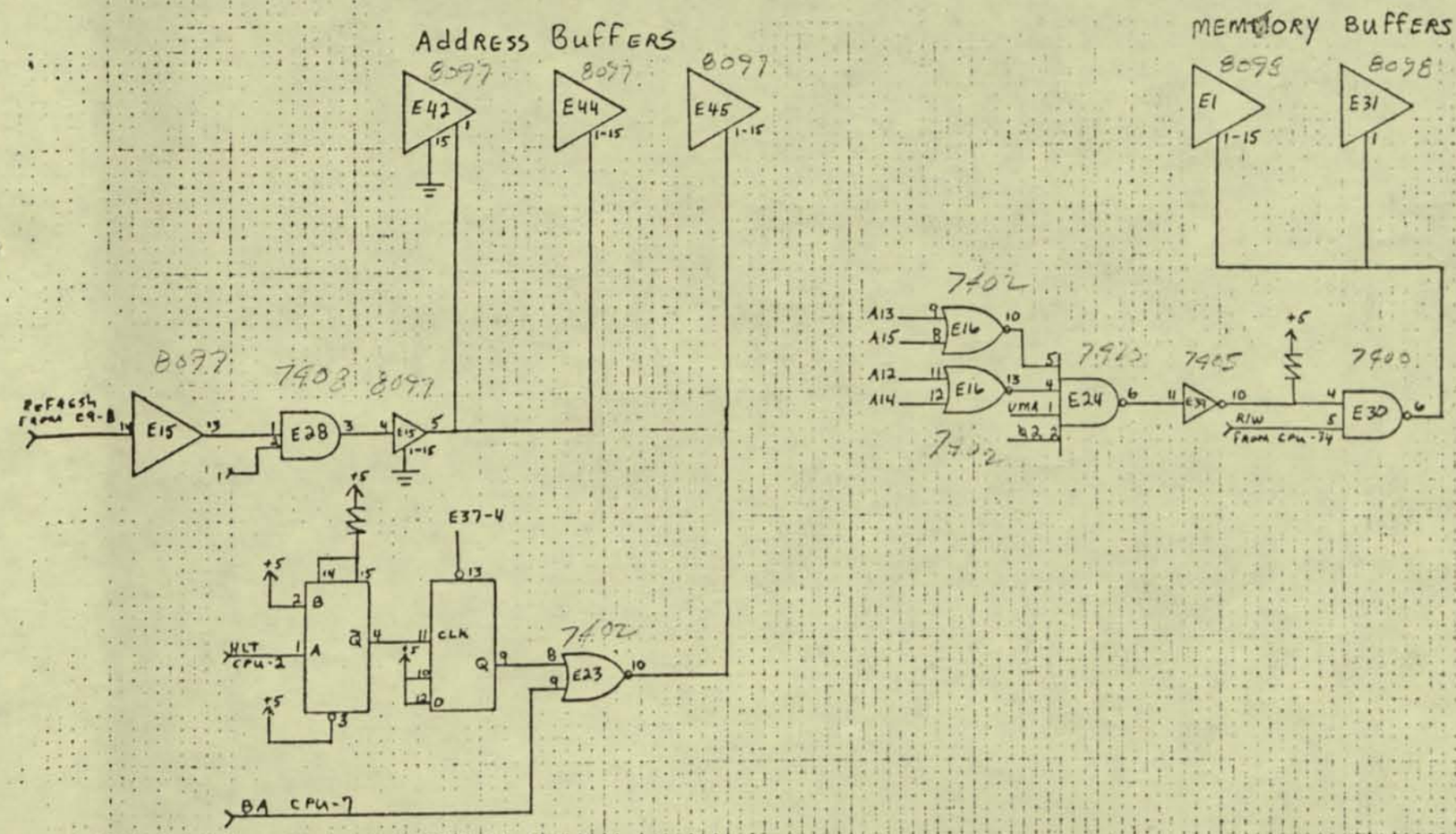


SPHERE
91 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010


CPU
HALT
Q1
Q2
D.B.E.
IRQ
NMI
T.S.C.
RESET
SPHERE
7402
7411
7420
7400
7493
REMOVE
MAY BE Removed
DISABLE REAL TIME Clock
REFRESH E23-1
512 HZ
64 HZ
128 HZ
256 HZ
X1-6
X1-7
X2-9
X2-12
E37-11
R/W
VMA
PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.

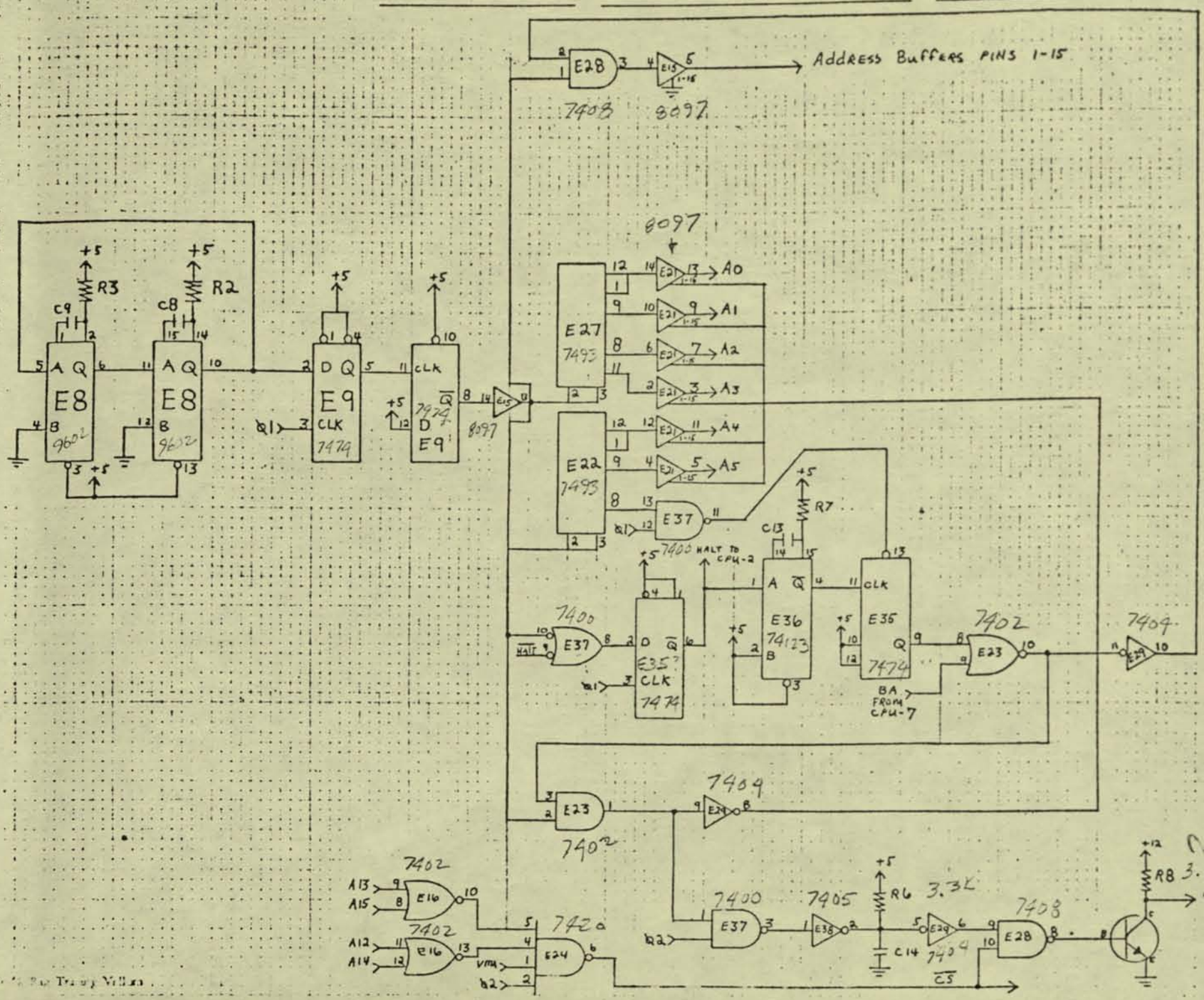
BY _____ DATE _____ PROJECT _____ SHEET NO. _____ OF _____

CHKD. BY _____ DATE _____ JOB NO. _____



SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY DRO DATE SUBJECT REFRESH SHEET NO. OF
CHKD. BY DATE JOB NO.



SPHERE
647 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

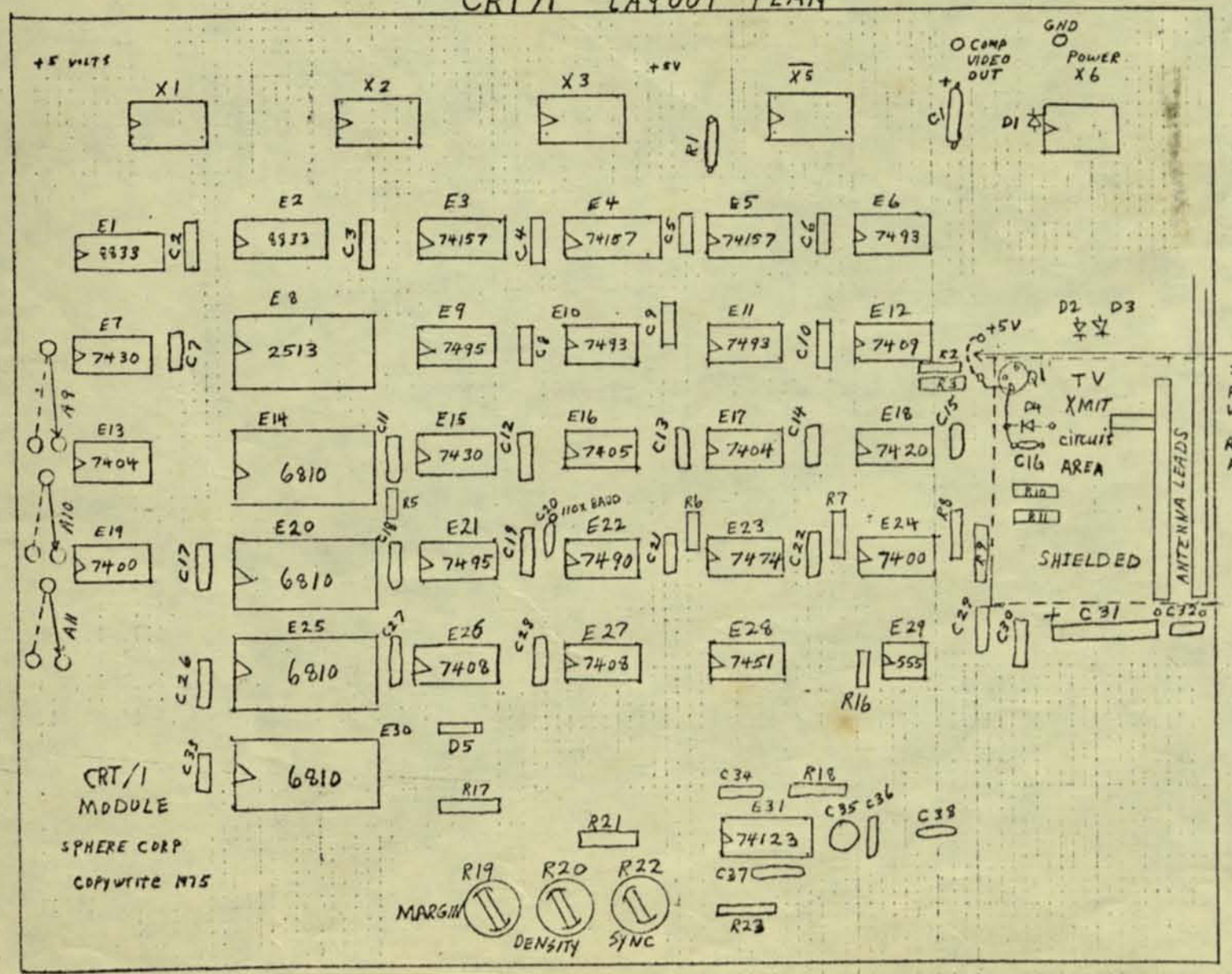
BY _____ DATE _____ SUBJECT CRT/1 SHEET NO 3 OF 3

CHKD BY _____ DATE _____ JOB NO. _____

NOTES:

1. Address as wired is Binary "0". To change to Address "1" cut A9 etch and add jumper along dotted path. To change to address "5" for example, change connection path of A9 and A1.
2. All IC's have pin 1 in lower left corner
3. Power and ground connect to pins 14 and 7 unless shown otherwise.
4. Item's in ( ) not supplied.

CRT/1 LAYOUT PLAN



Item	Part	Discription	Designation	Qnty
0	CRT/1	P.C. Board		1
1		14pin socket	X1 - X3, X6, X5	4 (1)
2	DM8833	Quad T/R	E1, E2	2
3	SN74157	Quad MUX	E3, E4, E5	3
4	SN7493	4 bit Bin. Cntr.	E6, E10 E11	3
5	SN7430	NAND Gate	E7, E15	2
6	2513	ASC II Char. Gen	E8	1
7	SN7495	4 bit Shift Reg.	E9, E21	2
8	SN7409	Quad & Gate	E12	1
9	SN7404	HEX Inverter	E13, E17	2
10	MCM6810	128 x 8 Static RAM	E14, E20, E25, E30	4
11	SN7405	HEX Inverter	E16	1
12	SN7420	Dual Nand gate	E18	1
13	SN7400	Quad Nandgate	E24, E19	2
14	SN7490	4bit Dec. Cntr.	E22	1
15	SN7474	Dual D F/F	E23	1
16	SN7408	Quad AND gate	E26, E27	2
17	SN74123	Dual monostable	E31	1
18	NE555	Timer	E29	1
19	SN7451	dual and/or gate	F28	1
20				
21	2N5129	Transistor	Q1 OR 2N2222A	1
22	2N918	Transistor	Q2	(1)
23	*L1	inductive coil	L1 SEE TVT MODULE IN RADIO ELECT	(1)
24	IN914	Diode	D1 - D5	4 (1)
25	20K	Resistor, var	R19	1
26	5K	Resistor, var	R20	1
27	50K	Resistor, var	R22	1
28	47uf	Cap. -16 vdc	C35	1
29	100uF	Cap. -10 vdc	C1, C31	1 (1)
30	.1uf	Capacitor	C2-C14, C17-C19 C21-C22, C26-28, C30, C33, C36	24
31	.01uf	Capacitor	C29	1
32	.001	Capacitor	C20, C34, C38	1
33	.27pf	Capacitor	C23	(1)
34	47pf	Capacitor	C16, C37	1 (1)
35	470pf	Capacitor	C15, C24, C32	1 (2)
36	8-25pf	Capacitor, VAR	C25	(1)
37	22	Resistor	R13	(1)
38	20K	Resistor	R17, R16	2
39	470	Resistor	R4	(1)
40	1K	Resistor	R2, R3, R6-R9, R12 R15, R18, R23	8 (2)
41	47 1/2w	Resistor	R10	1
42	100 1/2w	Resistor	R11	1
43	~~36K~~	~~Resistor~~	~~R16~~	~~1~~
44	10K	Resistor	R1, R5	2
45	2.2K	Resistor	R14	(1)
46	3.3K	RESISTOR	R21	1

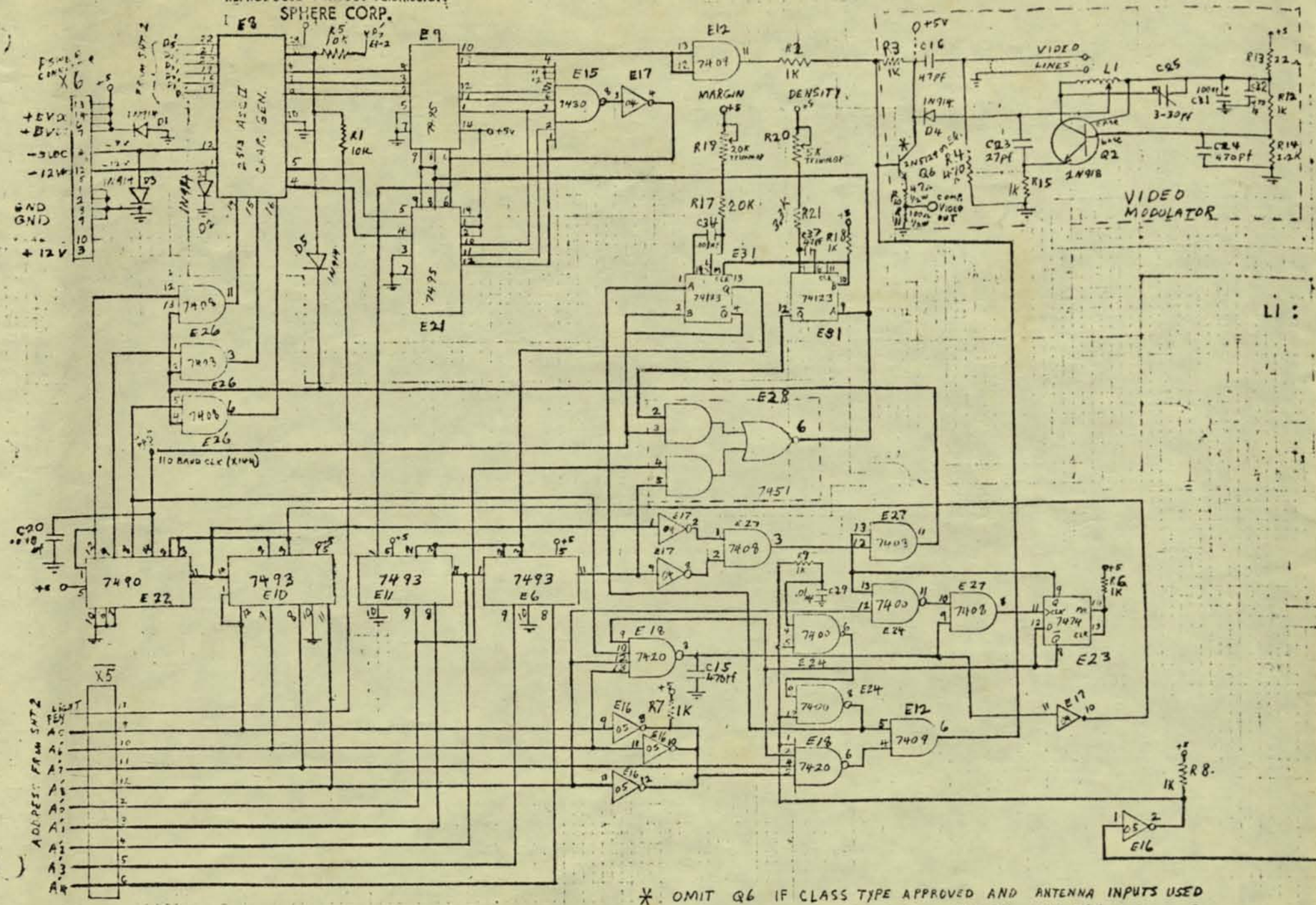
SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY _____ DATE 7/11/75
CHKD. BY _____ DATE _____
REVISED 09-06-75 M. KING

SUBJECT - RTTY MODULE

SHEET NO. 1 OF 3
JOB NO. 1

SPHERE CORP.



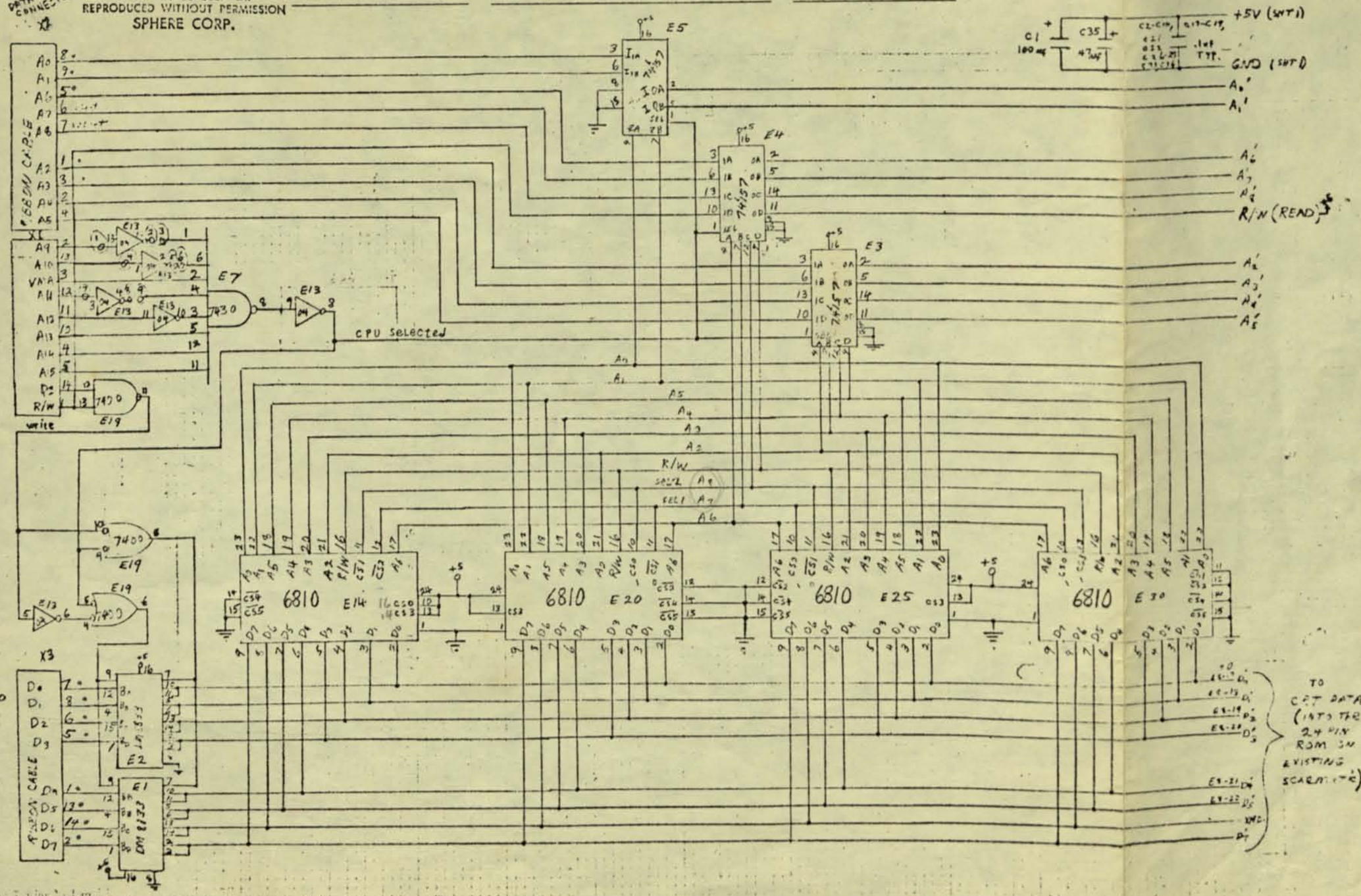
L1: 5 TURNS - TAPPED AT 1 TURN #14 WIRE. 3/8" DIAMETER X 3/4" LONG.

E0 ~ X262

* OMIT Q6 IF CLASS TYPE APPROVED AND ANTENNA INPUTS USED OTHERWISE OMIT D4 AND C24 AND Q1, AND USE COMPOSITE VIDEO OUT TO DRIVE A VIDEO MONITOR. IF L1 IS OMITTED, TIE THE JUNCTION OF D4 AND R3 TO +5V.

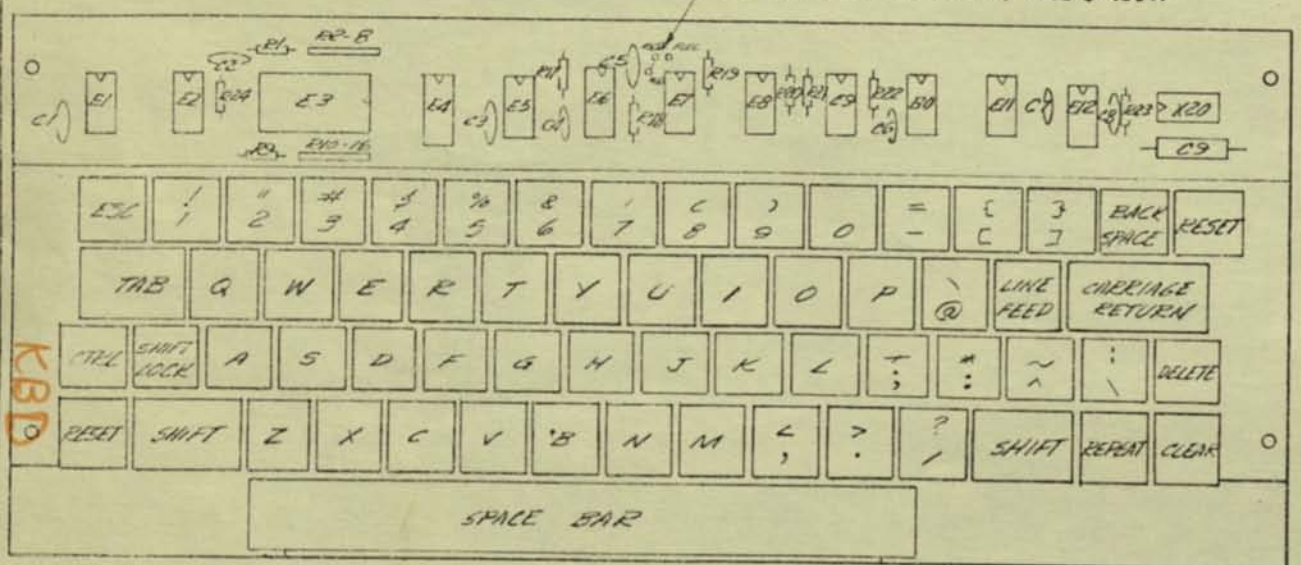
SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY mct/SSC DATE 15 APR 75
CHKD. BY mct DATE 7/1/75
SUBJECT CRT/I MODULE
SHEET NO. 2 OF 3
JOB NO. 1



SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

JUMPER FOR MOD ASC II
(1) ASC II TO MOD.
(2) FULL CONN. TO E7-4, CUT E7-9 TO E7-13
(3) CUT ETCH BETWEEN FULL & ASC II



ITEM NO.	QTY REQD	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO.	SPECIFICATION	MATERIAL OR NOTE
24	REF	SCHEMATIC	E000027		
23	1	PC BOARD	E000028		
22	1	ALPHANUMERIC KEYBOARD	613-10184		MASTER
21	1	CAPACITOR, RADIAL 10V	DDM-103	C9	CRL
20	1	CAPACITOR .01UF		C7	
19	1	CAPACITOR 100PF		C6	
18	1	CAPACITOR .2UF		C5	
17	1	CAPACITOR .001UF		C4	
16	4	CAPACITOR .1UF 50V	CK106	C1,2,C3,C8	CRL
15	1	RESISTOR 47K ¼W	RCO7GF473	R16	
14	1	RESISTOR 10K ¼W	RCO7GF103	R17	
13	2	RESISTOR NETWORK	760 87-R22K	R2-9, R10-16	CTS
12	8	RESISTOR 3.3K ¼W	RC07GF332	R1, 9-24	
11	1	SOCKET, 14 PIN IC	SK-A633D	X20	AUGAT
10	1	DUAL D FLIP FLOP	SN7474N	E12	
9	1	QUAD 2 INPUT NAND	SN7400N	E11	
8	1	QUAD 2 INPUT AND	SN7408N	E10	
7	1	HEX INVERTER	SN7404N	E9	
6	1	QUAD EXCLUSIVE-OR	SN7486N	E7	
5	1	DUAL ONE SHOTS	SN74123N	E6	
4	1	BCD TO DECIMAL DECODER	SN7442N	E4	
3	1	16 TO 1 MULTIPLEXER	SN74150N	E3	
2	2	HEX BUFFER/DRIVER	SN74367N	E2, E8	
1	2	4 BIT BINARY COUNTER	SN7493N	E1, E5	
		ASSEMBLY			

LIST OF MATERIAL

PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.

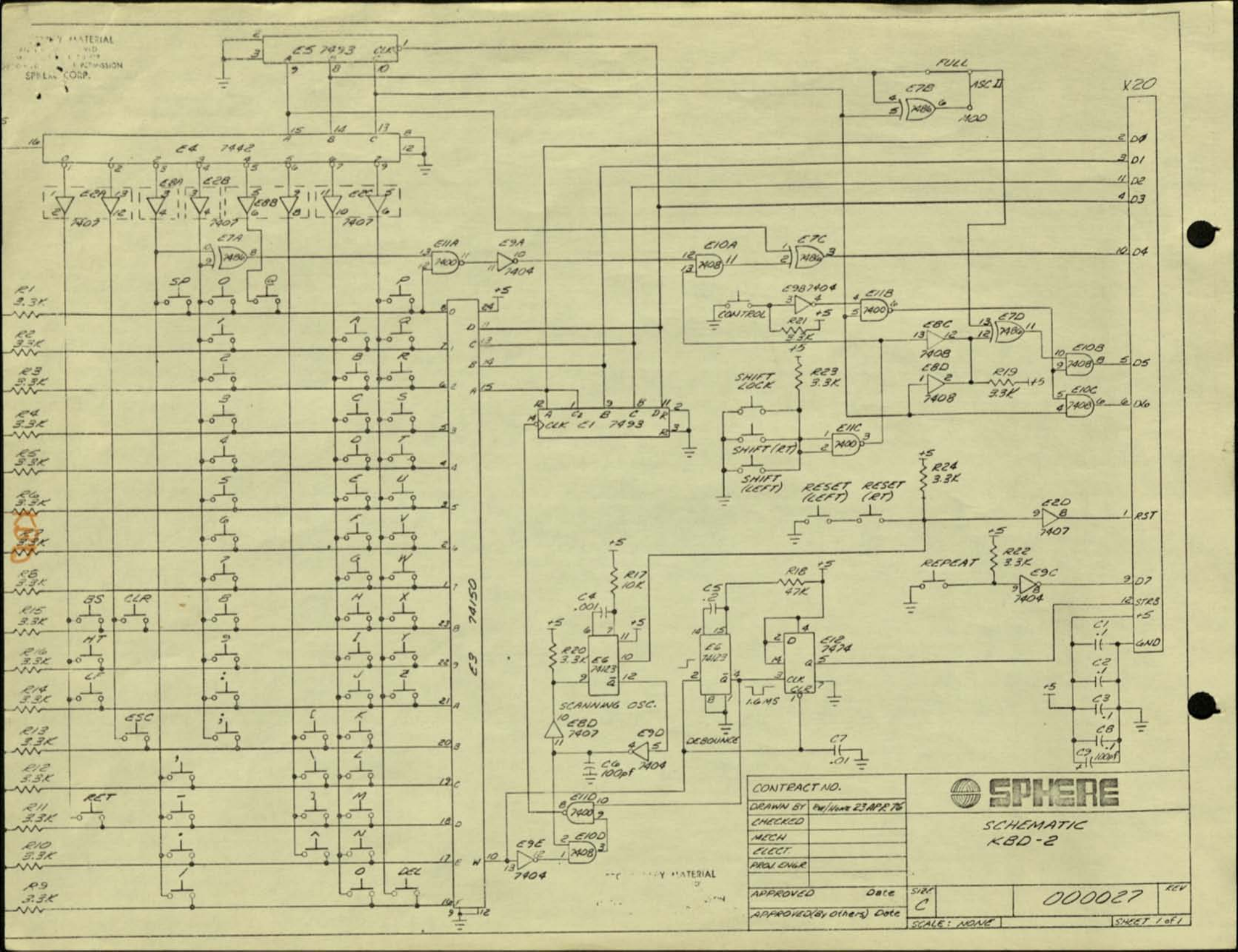
CONTRACT NO.	
DRAWN BY	Hume 17MAR76
CHECKED	
MECH	
ELECT.	
PROJ ENGR.	
APPROVED	Date
APPROVED (By Others)	Date

SPHERE

KBD12
ASSEMBLY

SIZE C | 000025 | REV

SCALE: NONE | SHEET 1 OF 1



CONTRACT NO.	
DRAWN BY	PW/Hume 23 APR 76
CHECKED	
MECH	
ELECT.	
PROJ. ENGR.	
APPROVED	Date
APPROVED (By Others)	Date

SPHERE

SCHEMATIC
KBD-2

SIZE C | 000027 | REV

SCALE: NONE | SHEET 1 of 1

THIS KEYBOARD WILL GENERATE A FULL ASCII CHARACTER SET HOWEVER, THIS IS NOT EASY TO USE IN HEX CODING FROM THE KEYBOARD.

BY MAKING THE JUMPERS SHOWN BELOW OR ON DRAWING A MODIFIED ASCII VERSION OF THE KEYBOARD IS ACHIEVED. THIS VERSION IS MUCH EASIER TO USE IN HEX CODING (THIS VERSION REQUIRES NO SHIFTING OF ALPHA CHARACTERS)

PRINTED CIRCUIT BOARDS THAT HAVE A PART NUMBER OF 00026 WITH NO REVISION SHOULD BE JUMPERED AS FOLLOWS:

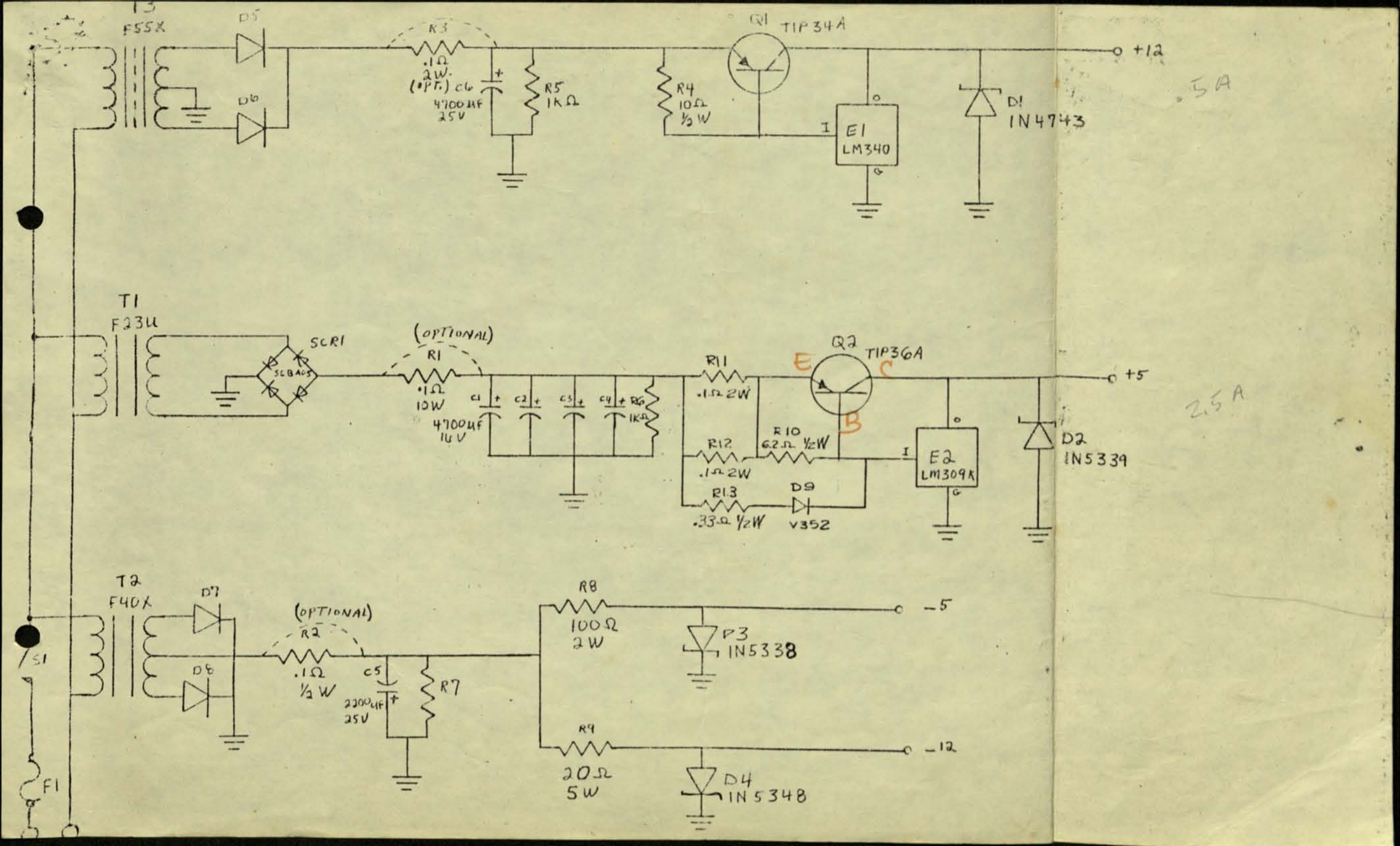
JUMPER FOR MODIFIED ASCII

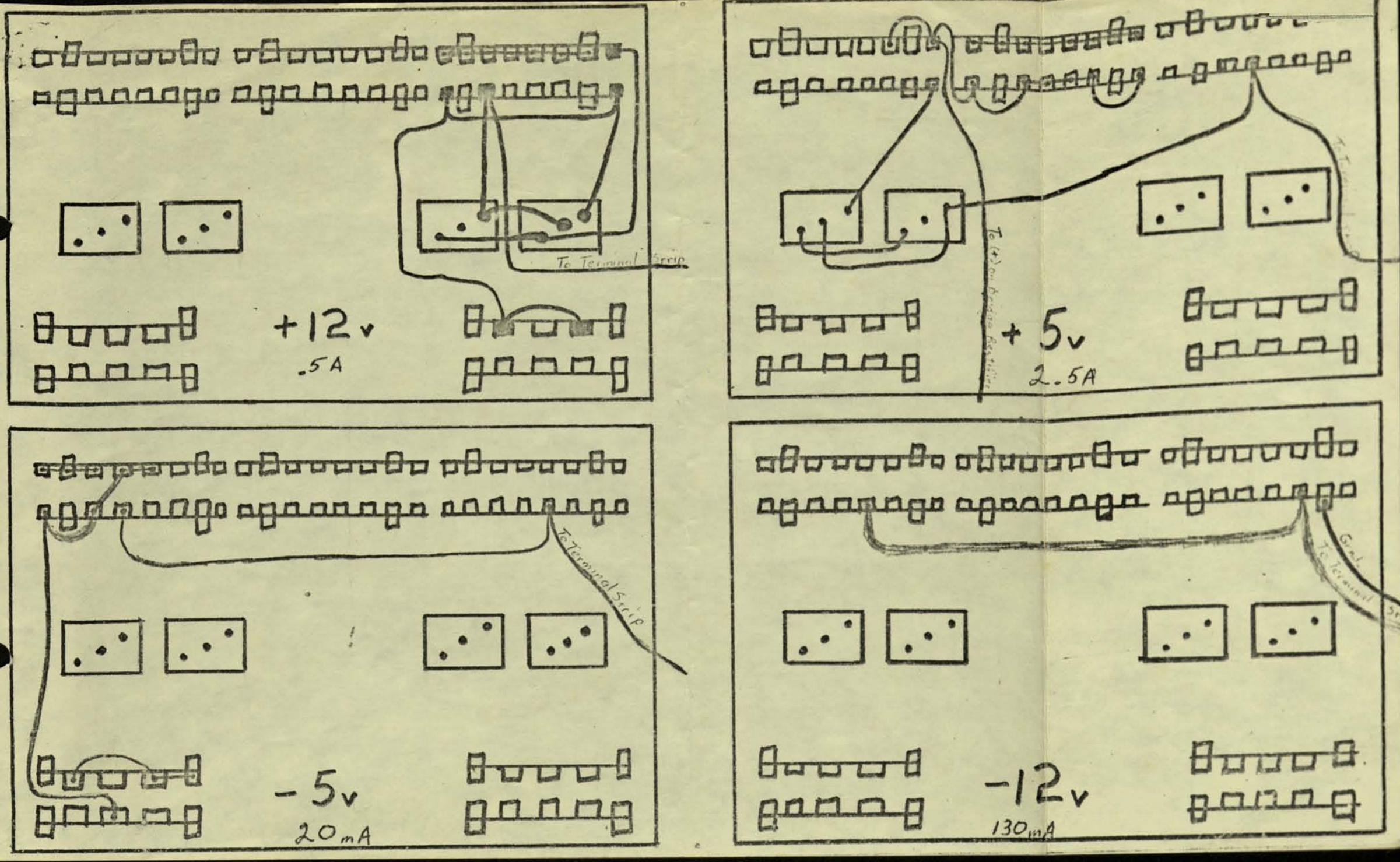
(1) 'ASCII'to E7-4 (use pin), Cut Etch E7-4 to E7-13 (under R19)

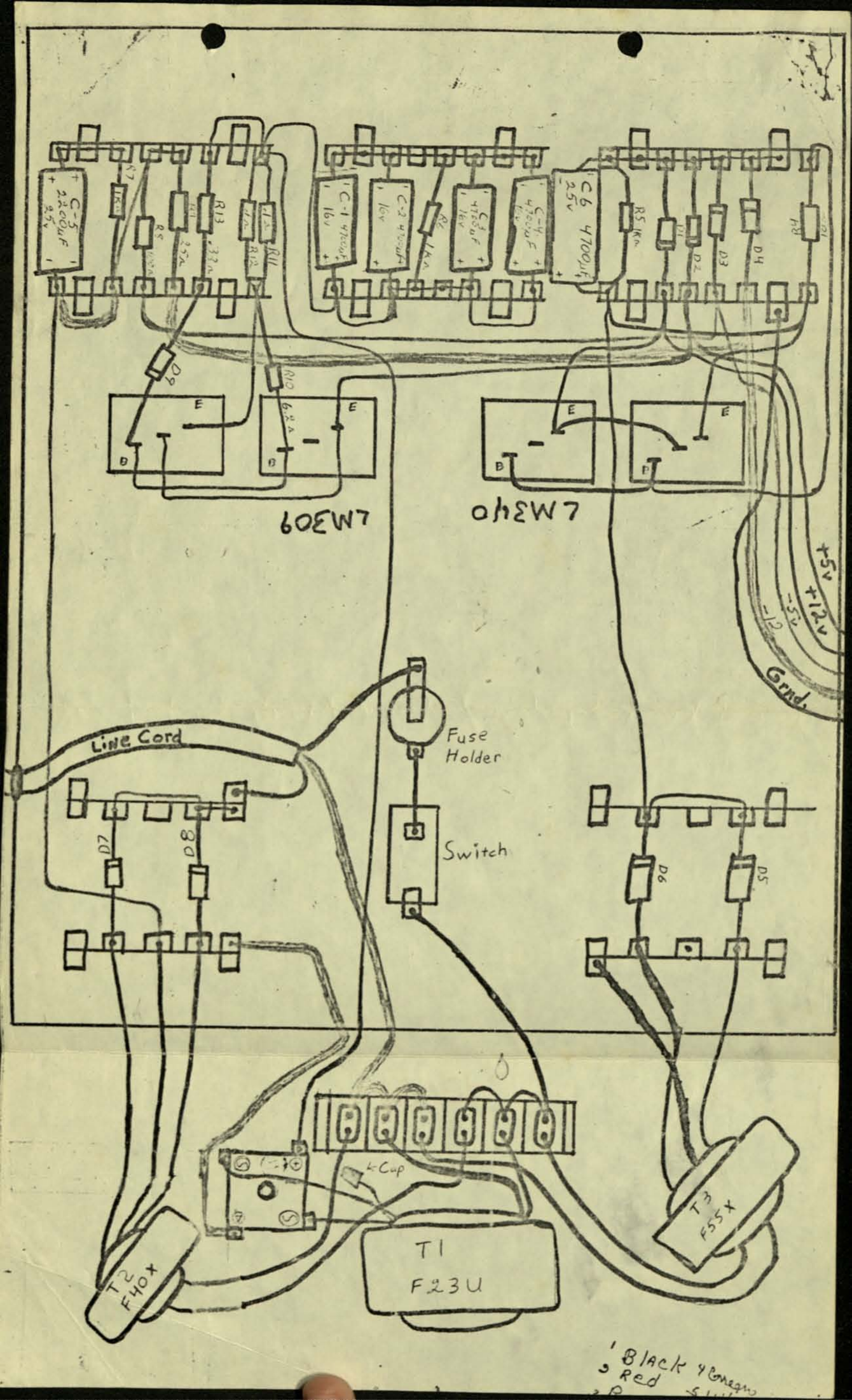
(2) 'MOD' to 'FULL'

(3) Cut Etch Between FULL & ASCII (Backside)

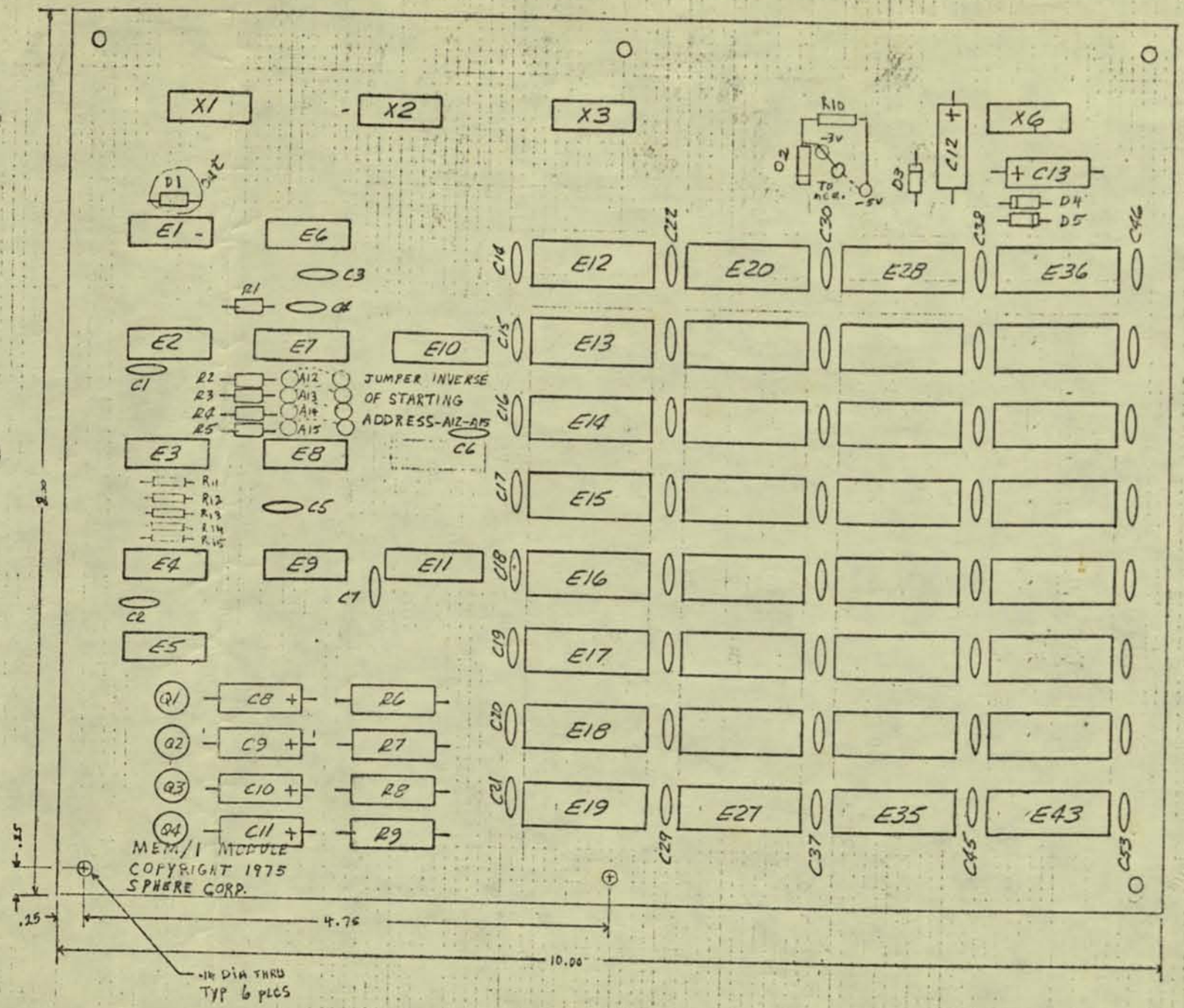
PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.


T3
F55X
D5
D6
R3
.1Ω
2W
(OPT.)
C6
4700μF
25V
R5
1KΩ
Q1
TIP34A
+12
R4
10Ω
½W
E1
LM340
D1
IN4743
5A
T1
F23U
SCR1
SCB40S
(OPTIONAL)
R1
.1Ω
10W
C1
C2
C3
C4
R6
1KΩ
4700μF
16V
R11
.1Ω 2W
Q2
TIP36A
E
C
B
+5
R10
6.2Ω ½W
R12
.1Ω 2W
R13
.33Ω ½W
D9
V352
E2
LM309K
D2
IN5339
2.5A
T2
F40X
D7
D8
(OPTIONAL)
R2
.1Ω
½W
C5
2200μF
25V
R7
R8
100Ω
2W
-5
P3
IN5338
R9
20Ω
5W
-12
D4
IN5348
S1
F1


To Terminal Strip
+12 v
.5A
To (+) on Amp Fuse Holder
+ 5v
2.5A
To Terminal Strip
- 5v
20 mA
Gnd
To Terminal Strip
-12v
130 mA


C-5 2200uF 25v
C-1 16v
C-2 16v
C-3 16v
C-4 16v
C6 4700uF 25v
R7 1K
R5 330
R13 33
R12
R11
R10 6.8A
R5 1K
R1
D1
D2
D3
D4
D9
E
B
6EW7
6EW7
+5v
+12v
-5v
-12
Grnd.
Line Cord
Fuse Holder
Switch
D7
D8
D6
D5
T2 F40X
T1 F23U
T3 F55X
4 Cup
1 Black & Green
2 Red

BY _____ DATE _____ SUBJECT MEM/1
CHKD. BY _____ DATE _____
SHEET NO. 3 OF 3
JOB NO. MEM/1 MODULE



ITEM	QTY	PART NO.	DESCRIPTION	DESIGNATION
1	1	MEM/1	P.C. BOARD - 8X10"	MEM/1
2	1	SN74123	ONE SHOT	E7
3	1	SN7400	NAND GATES	E6
4	1	SN7402	NOR GATES	E3
5	1	SN7404	INVERTER	E1
6	1	SN7408	AND GATE	E5
7			SPARE	E2, E4
8	1	SN7483	FULL ADDER	E8
9	2	DM8098	BUFFER	E10, E11
10	1	SN74156	2 LINE TO 4 LINE DECODER	E9
11	32	ZA-0243	4KX1 DYNAMIC RAM	E12-E43
12				
13	4	2N2369A	TRANSISTOR	Q1-Q4
14	3	IN4001	DIODE	D3-D5
15	1	IN914	DIODE	D1
16	1	IN5225B	3.0V ZENER	D2
17	4		RESISTOR, 100Ω, 2W	R6-R9
18	9		" 3.3K ½W	R2-R5 R10-R14
19	1		" 33K "	R1
20	1		" 133Ω 1/4W	R10
21	1		CAPACITOR, 33PF	C4
22	46		" CERAMIC .1uF	C1,C2,C3,C5,C6,C7,C8,C9,C10,C11,C14-C52
23	1		" 100MF-10VDC	C12
24	5		" 47uF 16V	C8-C11, C13
25	4	314-AG39D	16 PIN SOCKET	X1-X3, X6

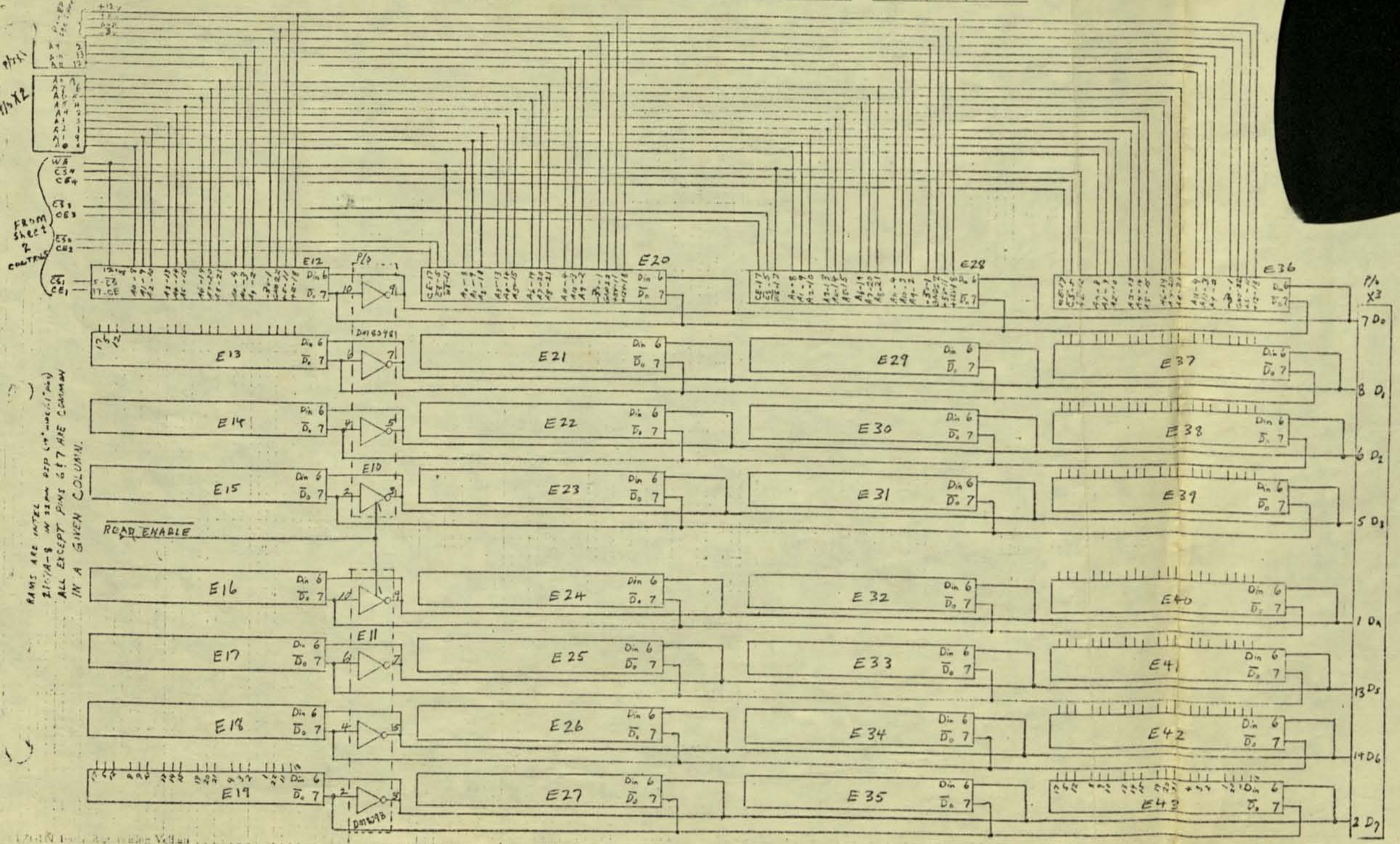
-510Ω

SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY M. L. TEST DATE 29 JULY 75 SUBJECT MEMORY MODULE
CHKD. BY ____ DATE ____
Rev. Oct 1, 1975

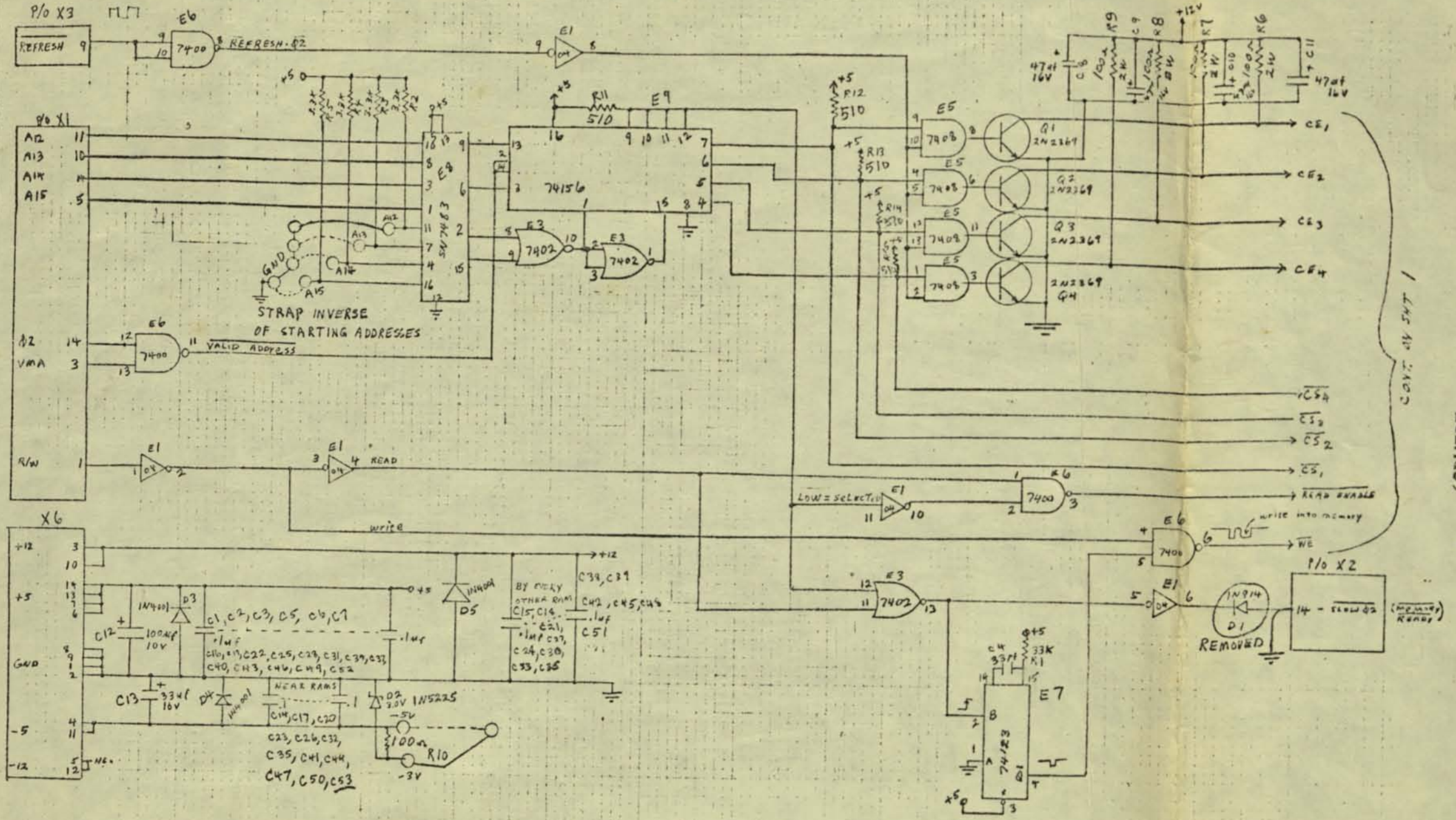
16K x 8 (DYNAMIC RAMS)

SHEET NO. 1 OF 3
JOB NO. MEM/1
MODULE



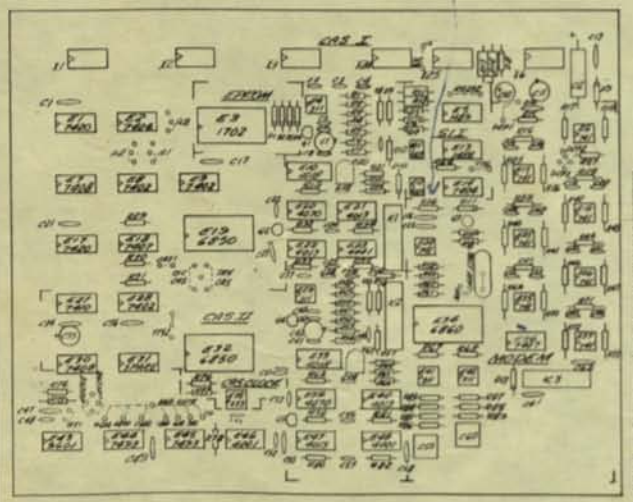
SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY M.C.T. DATE 29 JULY '75 SUBJECT 16K X 8 DYNAMIC MEMORY SHEET NO. 2 OF 2
CHKD. BY DATE JOB NO. MEM/1
Rev. Oct. 1, 1975 MODULE



SPHERE
791 SOUTH 500 WEST

R36



				86	A/R	SCHEMATIC	D 000023		
				85	1	PRINTED CIRCUIT BOARD	E 000024		
-	1	-	-	84	-	CRYSTAL	1 MHZ	XTAL	
-	1	-	-	83	-	RELAY	220-AN-12	K3	TRIPLINE
1	-	1	-	82	-	RELAY	220-AN-12	K1, K2	TRIPLINE
-	1	-	-	81	-	TRANSISTOR	2N2222	Q3	
1	-	1	-	80	-	TRANSISTOR	MPS6521	Q2, Q5	
1	-	1	-	79	-	TRANSISTOR	MPS6529	Q1, Q4	
-	2	-	-	78	-	CAPACITOR, 1μF		C33, 60	
-	-	-	-	77	1	CAPACITOR, 100 PF 500V	DT2-100	C48	CRL
-	-	-	-	76	1	CAPACITOR, 33 PF 1KV	DD-101	C47	CRL
1	-	1	-	75	-	CAPACITOR, 470 PF 1KV	DD-471	C24, 56	CRL
3	12	3	-	74	2	CAPACITOR, .01μF 100V	DDM-103	C1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 61, 62, 63, 64, 65, 66, 67	CRL
-	-	-	-	73	1	CAPACITOR, 100μF 25V		C12	ELKO
1	-	1	-	72	-	CAPACITOR, .0068μF 1KV	MYNE 1KV	C9, 46	
1	-	1	-	71	1	CAPACITOR, 100PF 1KV	DD-101	C8, 43, 52	
2	-	1	-	70	2	CAPACITOR, 47μF 15V	RX47/15	C1, 11, 14, 45	SOLE
8	-	8	-	69	-	CAPACITOR, .001 1KV	DD-102	C26, 27, 31	CRL
1	-	1	-	68	-	CAPACITOR, .047μF 50V	M 50W 50V	C4, 39	
1	-	1	-	67	-	CAPACITOR, .0015μF 1KV	DD-152	C7, 38	CRL
1	-	1	-	66	-	CAPACITOR, 47μF 16V		C2, 37	
3	5	-	-	65	7	CAPACITOR, .1μF 50V	CK-104	C10, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 28, 29, 30, 32, 33, 35, 36, 44, 48, 50, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64	CRL
-	1	-	-	64	-	RESISTOR, 470Ω 1/4W	RC07GF471	R39	
-	3	-	-	63	-	RESISTOR, 1.5K 1/4W	RC07GF152	R28-30	
-	-	-	-	62	-	RESISTOR	RC07GF, R, 1/4	R38	
-	1	-	-	61	-	RESISTOR 57.6K 1/4W 1%		R35	
-	1	-	-	60	-	RESISTOR 23.5K 1/4W 1%		R32	
-	1	-	-	59	1	RESISTOR 6.47K 1/4W 1%		R31, 33	
-	1	-	-	58	-	RESISTOR 34.8K 1/4W 1%		R40	
-	1	-	-	57	-	RESISTOR 6.19K 1/4W 1%		R63	
-	1	-	-	56	-	RESISTOR 3.32K 1/4W	RC07GF34	R59	

QTY PER SYS	QTY PER BD	QTY PER SUB	QTY PER ASSY	ITEM NO	QTY PER PART	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO	SPECIFICATION	MATERIAL OR NOTE

LIST OF MATERIALS OR PARTS

-	2	-	-	77	-	RESISTOR, 1MEG 1/4W	RC07GF105	E19, 60	
-	1	-	-	76	-	RESISTOR, 237K 1/4W 1%		R37	
-	1	-	-	75	-	RESISTOR, 1.24K 1/4W 1%		R36	
-	1	-	-	74	-	RESISTOR, 8.45K 1/4W 1%		R45	
-	1	-	-	73	-	RESISTOR 21.5K 1/4W 1%		R44	
-	1	-	-	72	-	RESISTOR 845Ω 1/4W 1%		R42	
-	1	-	-	71	-	RESISTOR 210K 1/4W 1%		R41	
-	1	-	-	70	-	RESISTOR, 348Ω 1/4W 1%		R40	
-	1	-	-	69	-	RESISTOR, 18.7K 1/4W 1%		R39	
-	3	-	-	68	-	RESISTOR 330Ω 1/4W	RC07GF331	R26, 27, 43	
1	-	1	-	67	-	RESISTOR, 100Ω 1/4W	RC07GF101	R17, 20	
2	-	2	-	66	-	RESISTOR, 10K 1/4W	RC07GF103	R34, 46	
1	-	1	-	65	-	RESISTOR, 20Ω 1/4W	RC07GF200	R24, R21	
-	2	-	-	64	-	RESISTOR, 68K 1/4W 1%		R22, 49	
-	1	-	-	63	-	RESISTOR, 267Ω 1/4W 1%		R23	
-	-	-	1	62	-	RESISTOR, 1.2K 1/4W	RC07GF122	R24	
1	-	1	-	61	-	POTENTIOMETER 2.0K	3352P	R20, 63	BOURNS
1	-	1	-	60	-	RESISTOR, 100Ω 1/4W	RC07GF101	R13, 42	
1	2	1	-	59	-	RESISTOR 22K 1/4W	RC07GF223	R12, 51, 52, 53	
1	2	-	-	58	-	RESISTOR, 15K 1/4W	RC07GF153	R14, 18	
-	1	-	-	57	-	RESISTOR, 620Ω 1/4W	RC07GF621	R15	
-	-	-	-	56	2	RESISTOR, 3.01K 1W	RC32GF301	R11, R16	
-	-	-	1	55	-	RESISTOR, 820Ω 1W	RC32GF821	R12	
-	-	-	1	54	-	RESISTOR, 1.1K 1W	RC32GF112	R21	
1	6	1	0	53	2	RESISTOR, 1K 1/4W	RC07GF102	R10, 19, 25, 41, 44, 47, 48, 64	
1	1	1	0	52	-	RESISTOR, 47K 1/4W	RC07GF473	R8, 33, 61	
1	-	1	0	51	3	RESISTOR, 33K 1/4W	RC07GF333	R1, 31, 62, 65	
3	-	3	0	50	1	RESISTOR, 47K 1/4W	RC07GF473	R2, 3, 9, 10, 50	
4	-	4	0	49	0	RESISTOR, 13K 1/4W	RC07GF133	R4, 5, 6, 7, 35, 36, 37, 38, 54, 55, 56, 57, 58, 59, 60	
2	-	2	0	48	0	RESISTOR, 2.0K 1/4W	RC07GF202	R13, 66, 67	
1	6	1	3	47	7	DIODE	IN 4001	D1-D7	
1	1	-	-	46	2	24 PIN IC SOCKETS	3M 4824-3000	13, 15, 32, 34	
-	-	-	-	45	6	14 PIN IC SOCKETS	3M 4814-3000	11-14, 24, 25	
-	-	-	-	44	2	4 BIT BINARY COUNTER	SN 74393 N	E44, 45	
-	-	-	-	43	1	DIME SWITCH	360 (THUMBWHEEL)	E 42	
-	-	-	-	42	1	PRECISION TIMER	SN 72555 P	E 38	
-	1	-	-	19		MODEM	MC 6860	E34	
-	-	-	-	18		TIMER		E 31	
1	-	-	-	17	-	TRIPLE 3-INPUT NAND GATE	SN 74 10 N	E29	
1	-	-	-	16	1	NOR GATE	CD 4001 B	E23, 26, 28	
2	-	2	-	15	-	DUAL D FLIP FLOP	CD 4013 A	E21, 22, 40, 47	
1	-	1	-	14	-	EXCLUSIVE OR GATE	CD 4070 B	E20, 39	
1	-	-	-	13	1	ACIA	MC 6850	E13, 36	
1	1	1	0	12	1	HEX BUFFER	SN 7407 N	E17, 35	
-	-	-	1	11	-	QUAD TRANSMITTER - RS232	MC 1489 N	E18	
-	-	-	2	10	-	PHOTO COUPLER	4N 33 N	E11, E12	
1	-	1	-	9	-	COUNTER	CD 4020 B	E10, 33	
1	-	-	-	8	2	QUAD 2-INPUT NOR GATE	SN 7402 N	E2, 3, 27	
1	-	-	-	7	1	QUAD 2-INPUT AND GATE	SN 7408 N	E7, 30	
-	2	-	-	6	-	OPERATIONAL AMPLIFIER	LM 741	U1, 2, 3, 4	
-	-	-	1	5	-	QUAD RECEIVER - RS232	MC 1489	E9	
1	2	1	-	4	-	VOLTAGE COMPARATOR	LM 311 N	U5, 6, 7, 8	
-	-	-	-	3	16	E PROM 256 X 8 BIT	MM 1702 A	E8	
-	-	-	1	2	1	HEX INVERTER	SN 7404 N	E14	
-	-	-	1	1	2	8 INPUT NAND GATE	SN 7430 N	E16	
.05	.04	.03	.02		.01	ASSEMBLY			

QTY PER SYS	QTY PER BD	QTY PER SUB	QTY PER ASSY	ITEM NO	QTY PER PART	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO	SPECIFICATION	MATERIAL OR NOTE

LIST OF MATERIALS OR PARTS

03	AUDIO CASSETTE INTERFACE II
04	MODEM
03	AUDIO CASSETTE INTERFACE I
02	SERIAL INTERFACE I
01	STD

PART NO.	TITLE	REVISION LEVEL

A SPECIAL ITEM CONTROLED BY SOFTWARE DEPT.

DRAWN BY	ISSUE RELEASE
CHECKED BY	
R.E.D.	
QA CONT.	
MFG ENG.	
PROD ENG.	
APPROVAL	
APPROVAL	
FINAL APPROVAL	

SPHERE

SIM/1
ASSEMBLY

SIZE		REV
D	000022	A

SCALE NONE | SHEET 1 OF 1

SIM

CUT

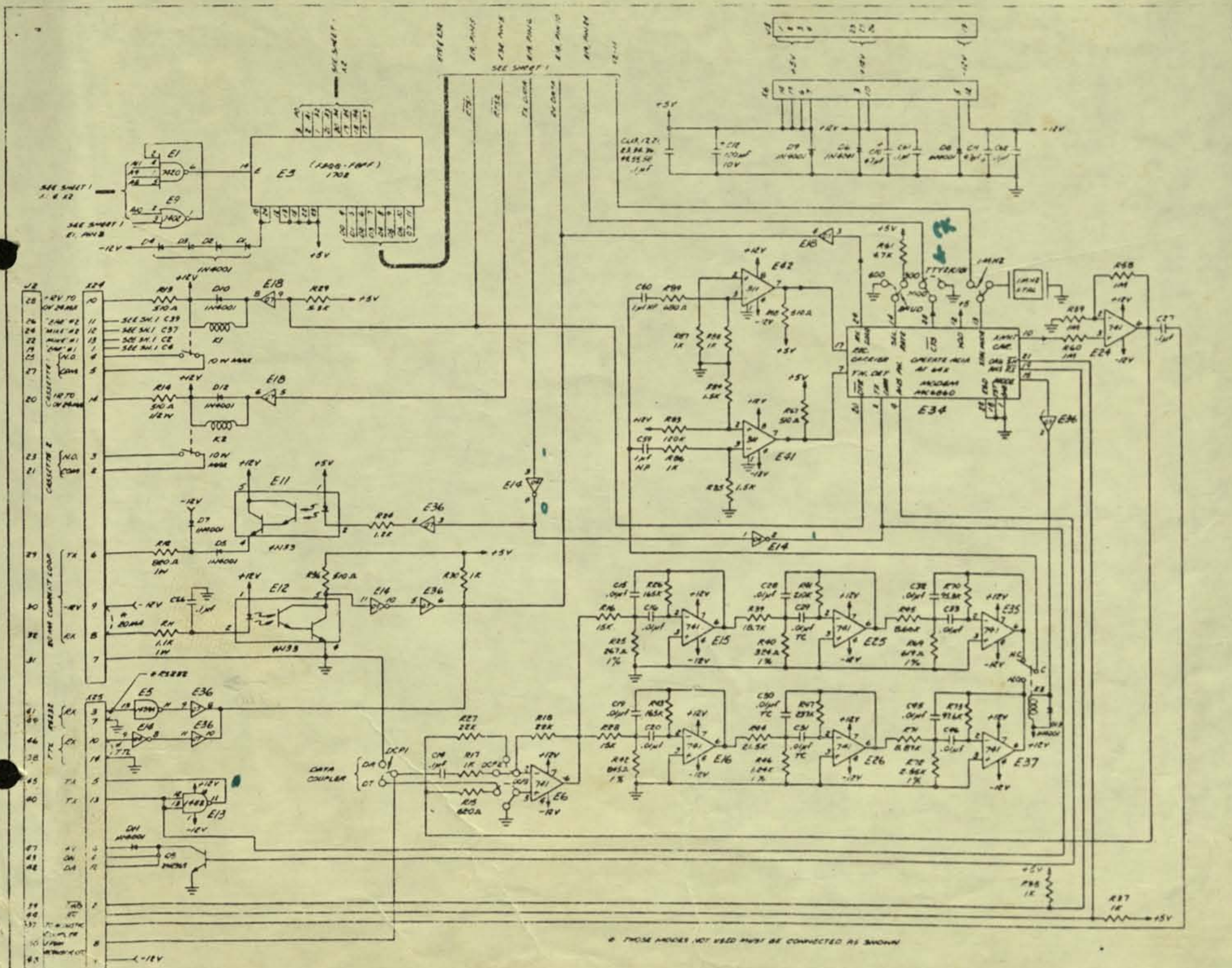
← CK FREQ

#2

#1

NOTE(S)
1. OUTPUTS ARE ±8 VOLTS. SOME RECORDERS NEED ONLY SOME SIGNALS - USE A RESISTIVE DIVIDER FOR SUCH UNITS.

DRAWN BY	M EDDINGTON 8/2/76	SPHERE
CHECKED	D TYLER 10/4/76	SIM/1 SCHEMATIC
R&D		
QA (QUIT)		
MFG ENG		
PROJ ENG		SIZE D — 000023 — REV A
APPROVAL		
APPROVAL		
FINAL APPROVAL		SCALE NONE — SHEET 1 OF 2

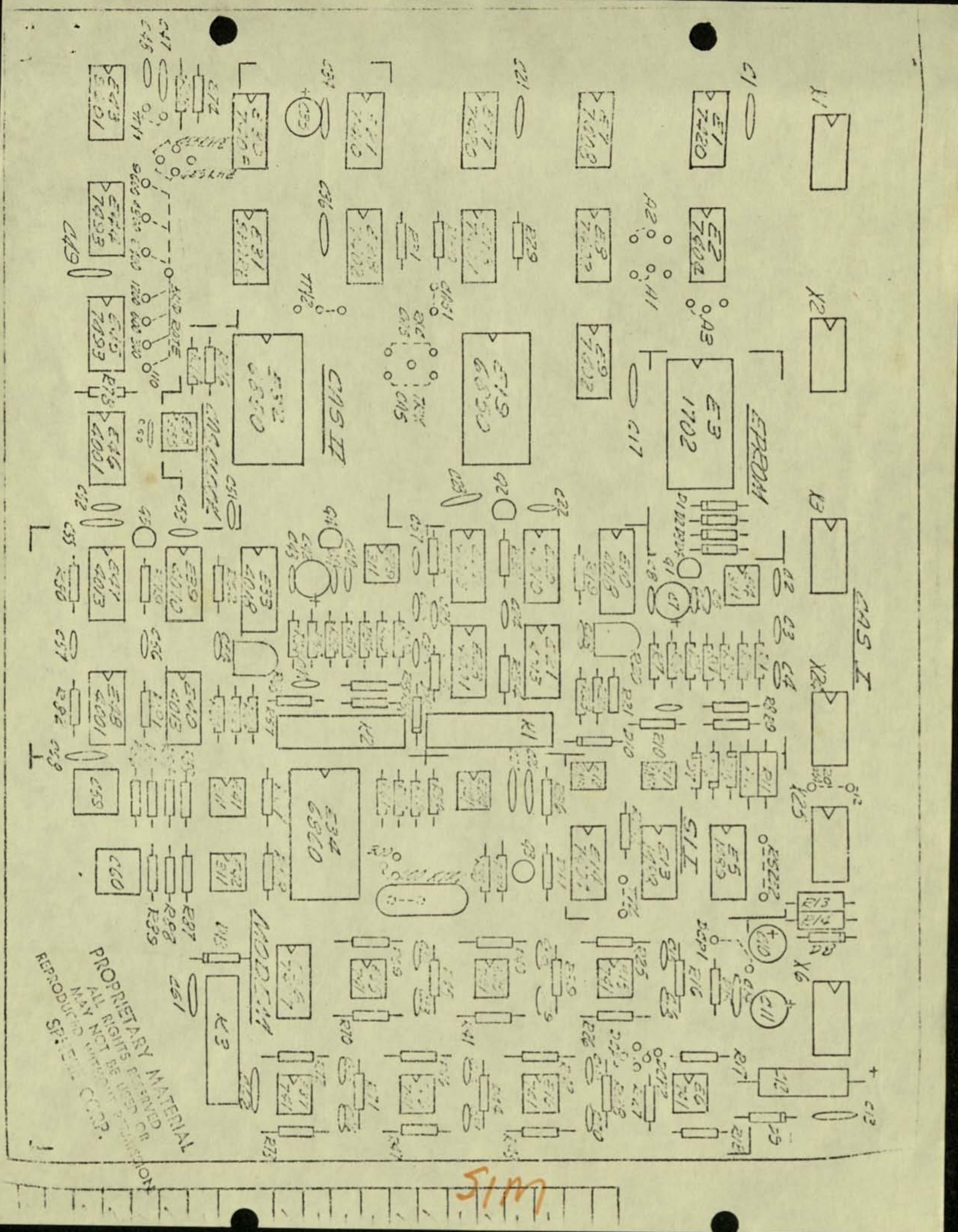


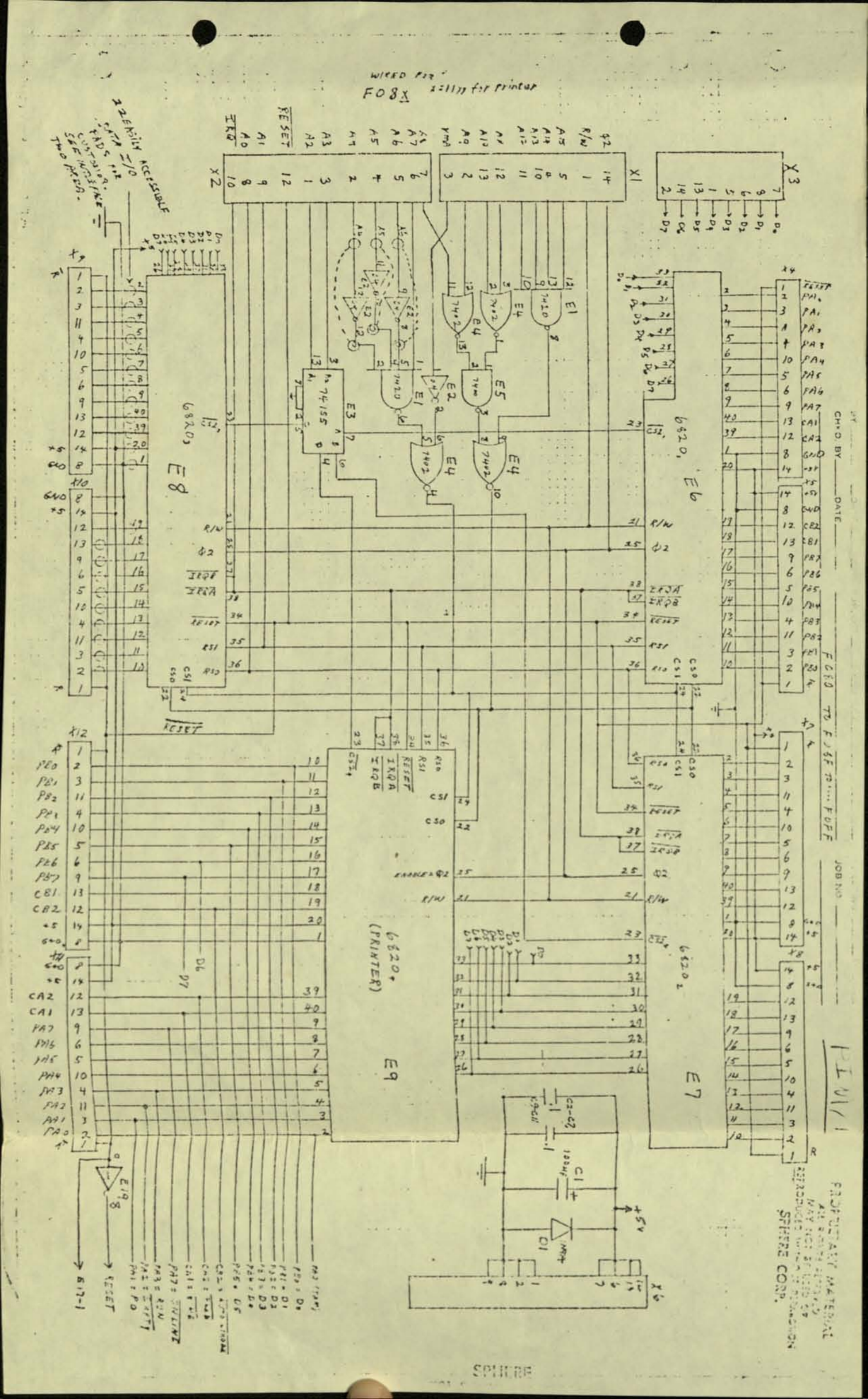
DRAWN BY	M. ROBINSTON 10/7/76
CHECKED	M. TILLER 10/27/76
R & D	
QA CONT	
MFG ENG	
PROJ ENG	
APPROVAL	
APPROVAL	
FINAL APPROVAL	

SPHERE

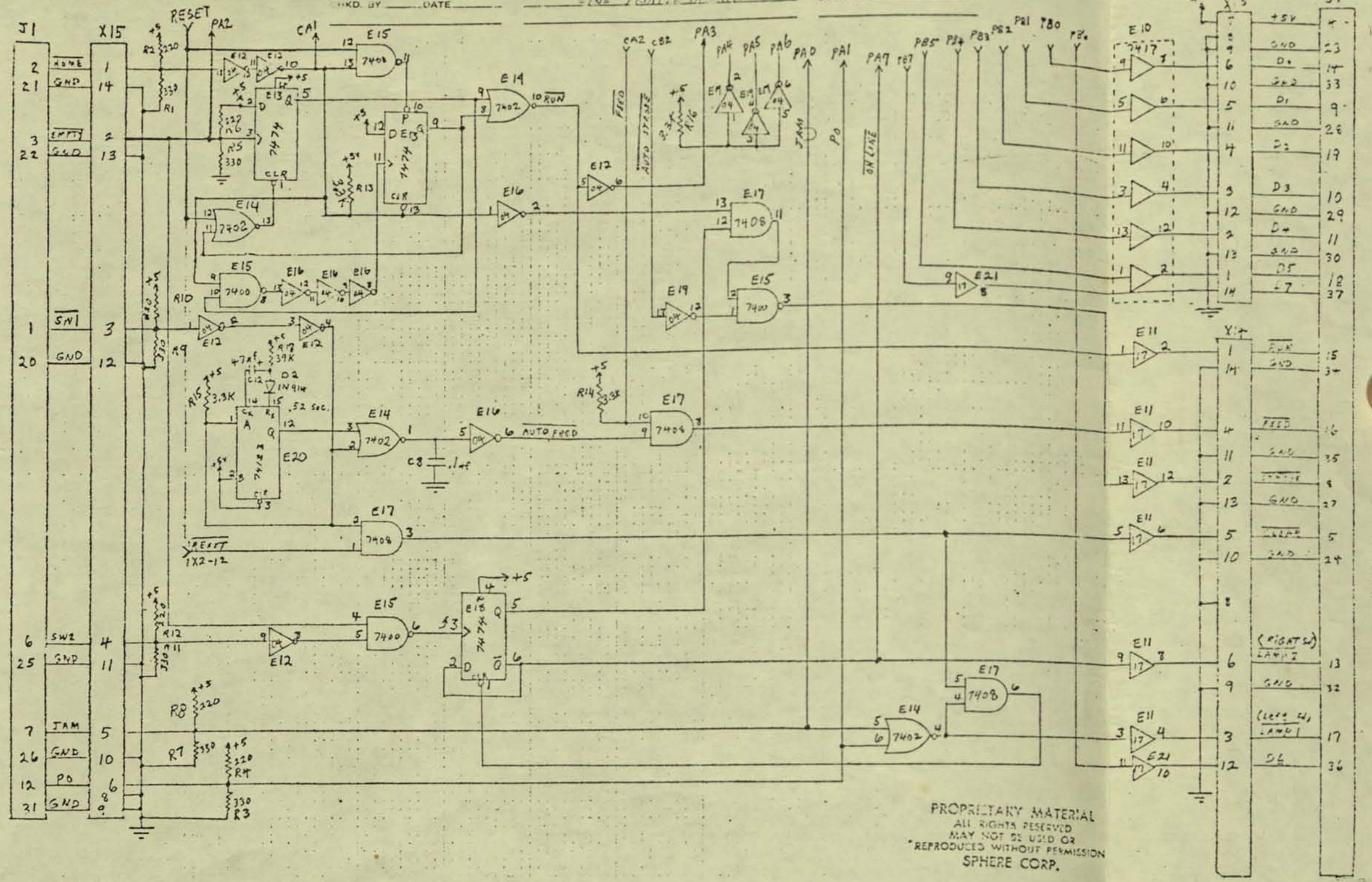
SIM/11
SCHEMATIC

SIZE		REV
D	000023	A
SCALE NONE		SHEET 2 OF 2



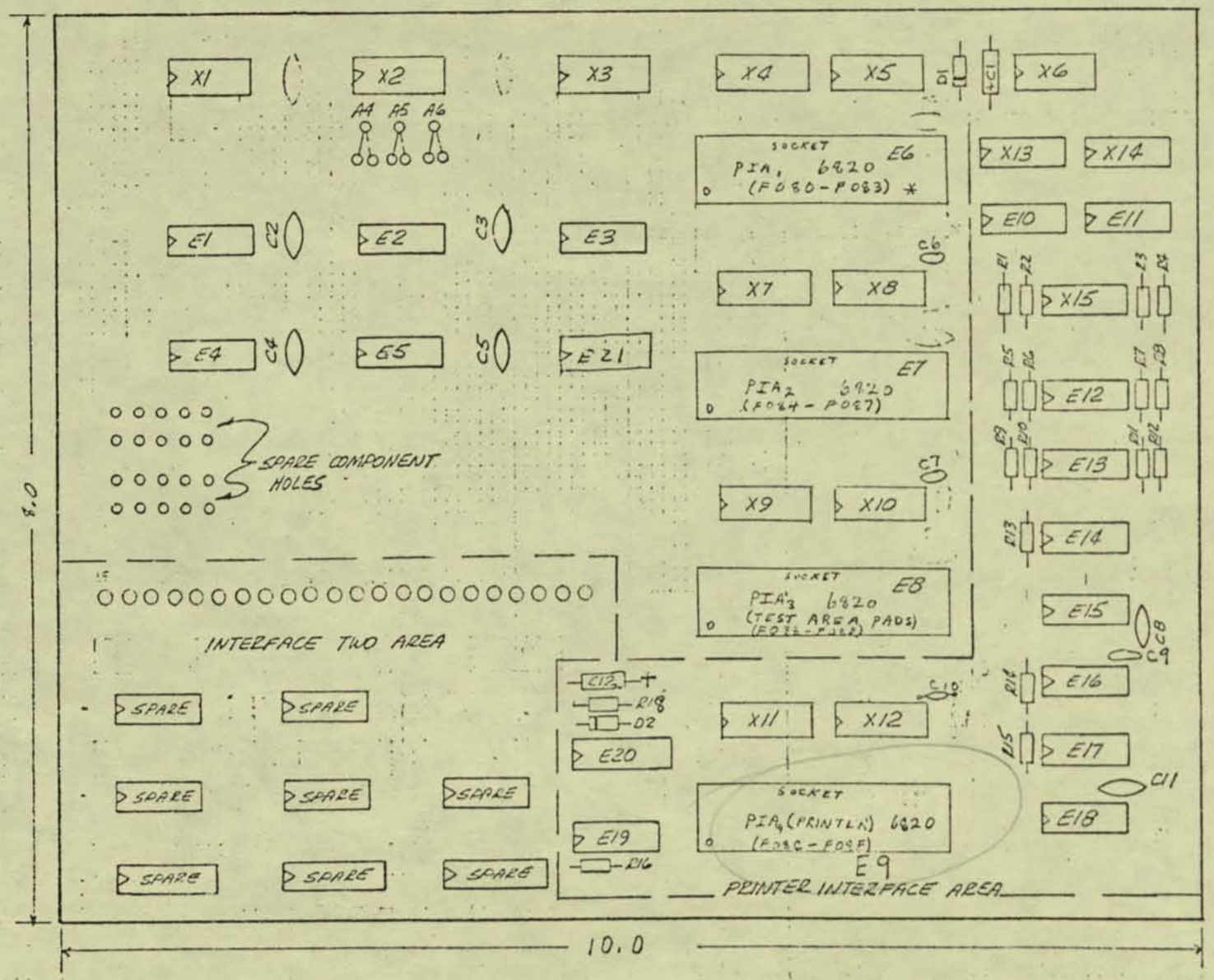

WIRED FOR F08X
2 SERIAL ACCESSIBLE
PARA I/O
PADS FOR
CUSTOMIZING
SEE INSTRUCTIONS
TWO PADS-
X1
X2
X3
RESET
IRQ
R/W
E0
E1
E2
E3
E4
E5
74LS20
74LS02
74LS155
6820, E6
6820, E7
6820, E8
6820, (PRINTER) E9
E19
C1
C2-C7
1000uf
D1
+5v
CH'D BY
DATE
JOB NO
PROPRIETARY MATERIAL
SPHERE CORP.

CHKD. BY _____ DATE _____ LINE PRINTER INTERFACE JOB NO. _____ SHEET NO. _____ OF _____



BY ____ DATE ____ SUBJECT PIM/1 ____ SHEET NO 3 OF ____

CHKD. BY ____ DATE ____ LAYOUT AND PARTS LIST JOB NO ____

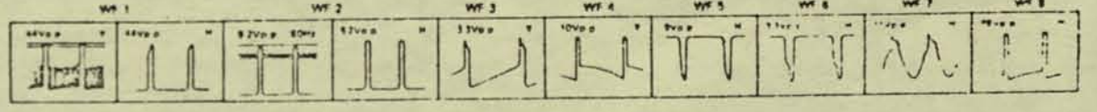


* ADDRESS GIVEN AS WIRED! JUMPERS ON A4, A5, AND A6 WILL ALLOW SELECTION OF 7 ADDITIONAL PIM/1 CARDS WITH ADDRESSES F09X, F0AX, F0BX, ... F0FX.

1	1	PIM/1	P. C. Board	PIM/1
2	4	MC6820	PIA	E6-E9
3	2	SN7400	NAND Gate	E5, 15
4	2	SN7402	NOR Gate	E7, 14
5	4	SN7404	Inverter	E2,12,16,17
6	1	SN7408	AND Gate	E17
7	2	SN7417	Hex Buffer	E10,11, 21
8	1	SN7420	NAND Gate	E1
9	2	SN7474	F/F	E13,18
10	1	SN74123	One Shot	E20
11	1	SN74155	2 line to 4 line Demux	E3
12				
13	15	314-AG390	14 Pin Socket	X1-X15
14	4	340-AG390	40 Pin Socket	E6-E9 sockets
15	2	IN914	Diode	D1, D2
16	6		Resistor, 330 ohm, 1/4 w	R1,3,5,7,9, 11
17	6		Resistor, 220 ohm, 1/4 w	R2,4,6,8, 10, 12
18	4		Resistor 33K 1/4 w	R13-16
19	1		Resistor 39K 1/4 w	R18
20	1		Capacitor 100 Luf.	C1
21	10		Capacitor .1 luf.	C2-C11
22	1		Capacitor 47 luf.	C12

No R17

SPHERE



VIDEO AMP
Q913~Q915 2SC536×3

PICTURE TUBE
V101 240DB4
or 240DB4A

VIDEO OUTPUT
Q101 2SC65Y

CONTRAST

VIDEO

SYNC. SEP.
Q102 2SA201

VERT. OSC.
Q103 2SC536

VERT. DRIVE
Q104 2SC536

VERT. OUTPUT
Q105 2SD024
or 2SD315
or 2SD331

SEESAW SWITCH
SW1

POWER RECT.
D112 PH9d522/1

POWER TRANS.

V-HOLD

HEIGHT

V-LIN

PHASE DET.
D104, D105 1S188×2

VERT. OUTPUT
Q105 2S9474

H.V. RECT.

B-ADJ.

REGULATOR
Q111 2SA608

HORIZ. DRIVE TRANS.

H-FREQ

DAMPER
D106 DG-13
or DG-130

H-HOLD

REGULATOR
Q113 2SB407

REGULATOR
Q112 2SD30

PHASE INVERTER
Q107 2SC536

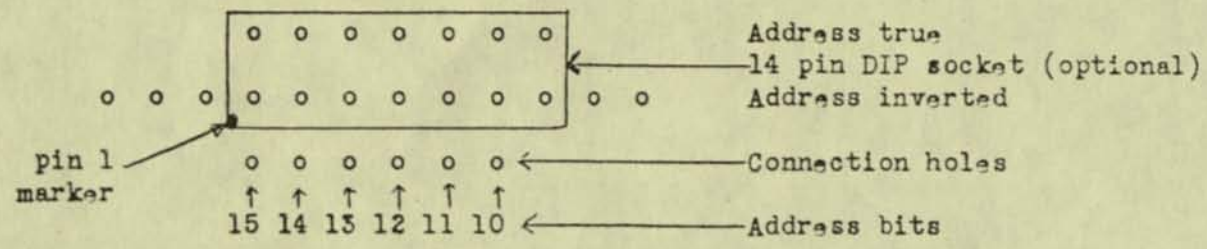
HORIZ. OSC.
Q108 2SC536

HORIZ. DRIVE
Q109 2SB187S

HORIZ. OUTPUT
Q110 2SB375A

SPHERE READ ONLY MEMORY BOARD

The SPHERE ROM/1 board is designed to provide up to 4K of read only memory program space using the commonly available 1702 PROM. The addressing on the board is fully selectable on 1K boundaries by the user. All four banks must be either address strapped or grounded - no banks' address select logic may be left open. If the addressing on your board will change frequently a 14 pin DIP socket may be installed at J1-J4 and 2 inch wire-wrap type wires in the adjoining connecting holes. These wires may then be pushed into the correct hole of J1-J4



To select an address, jumper each connection hole to the appropriate address selection hole. To make a 'don't care' bit (either on or off will do) leave the jumper wire from the connection hole open (no connection). It is for this reason that unused banks must be strapped down.

To select 1000 as the starting address of bank 1 (U13 1000, U12 1100, U11 1200, U10 1300) connect
- 15 to address inverted, pin 1 of J1
- 14 to address inverted, pin 2 of J1
- 13 to address inverted, pin 3 of J1
- 12 to address true, pin 11 of J1
- 11 to address inverted, pin 5 of J1
- 10 to address inverted, pin 6 of J1

To select DC00 as the starting address of bank 3 (U25 DC00, U24 DD00, U23 DE00, U22 DF00) connect
- 15 to address true, pin 14 of J3
- 14 to address true, pin 13 of J3
- 13 to address inverted, pin 3 of J3
- 12 to address true, pin 11 of J3
- 11 to address true, pin 10 of J3
- 10 to address true, pin 9 of J3

To strap a bank of ROM off, connect any (at least one) connection hole to a ground, available feed-throughs near pin 7 of U9, U15, and U21.

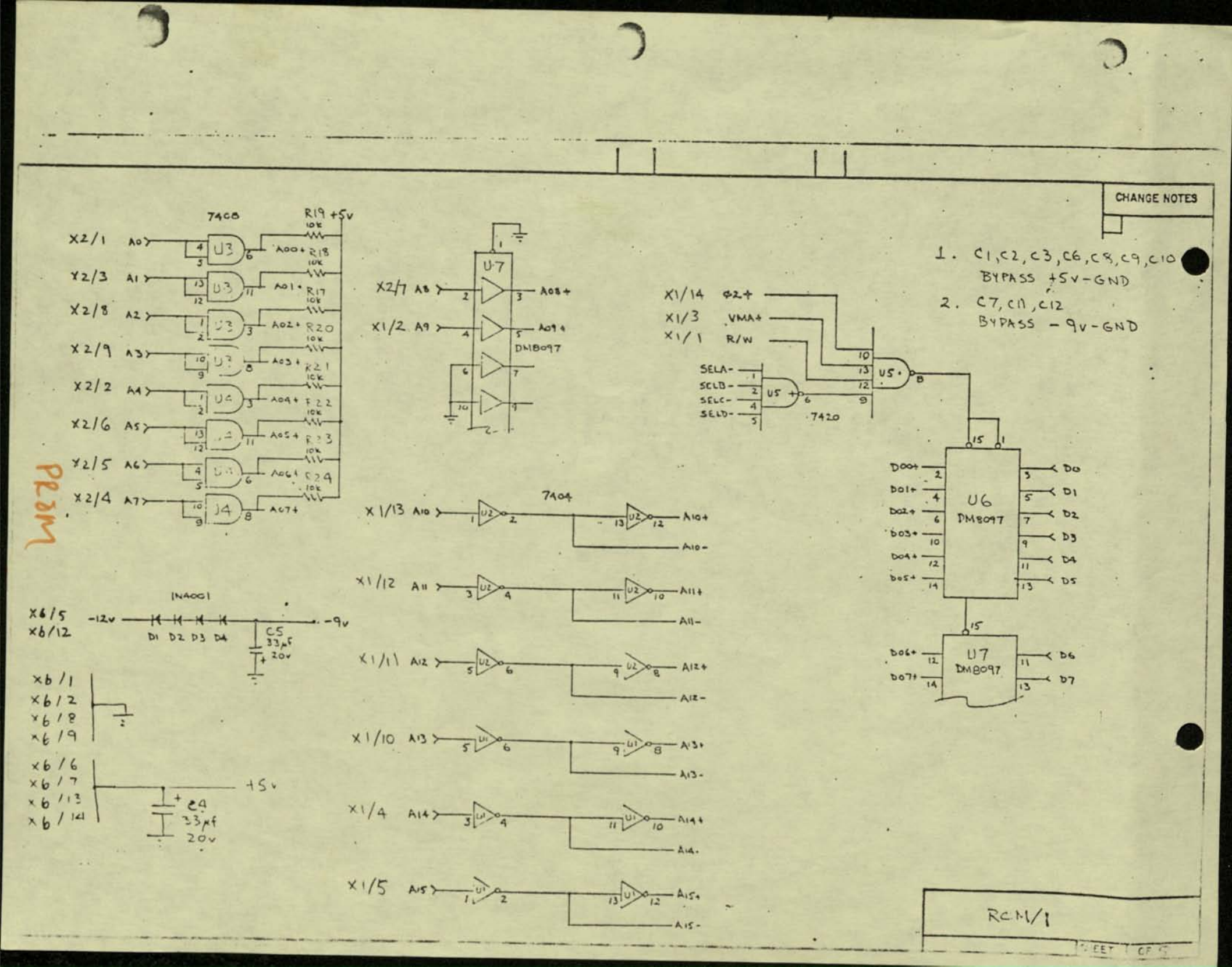
In selecting addresses, remember that below 1000 is dedicated to RAM and above E000 are I/O devices and system ROM's.

Good programming to you all.

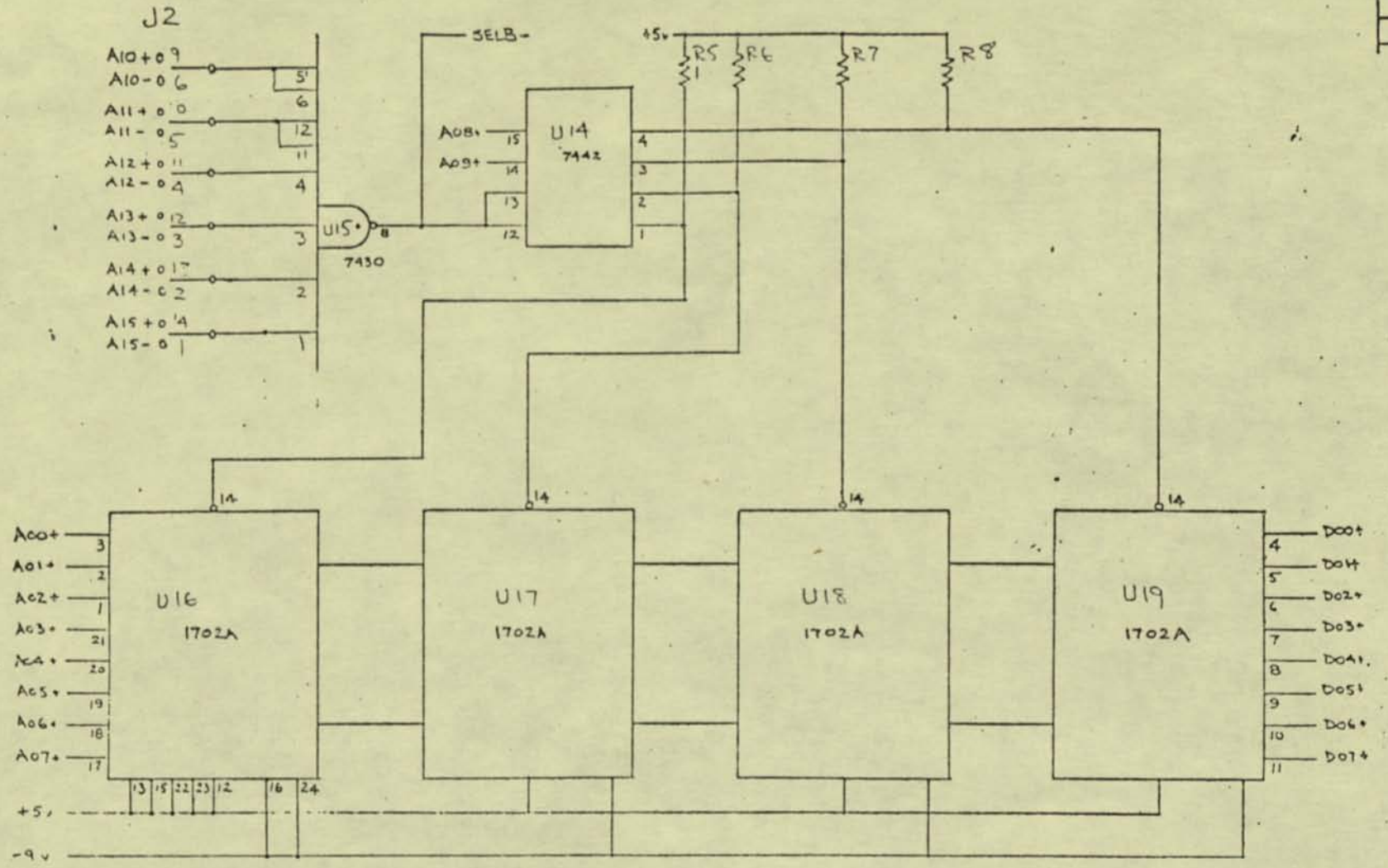
Ernest Dixon

Ernie Dixon
SPHERE Corporation

PROM

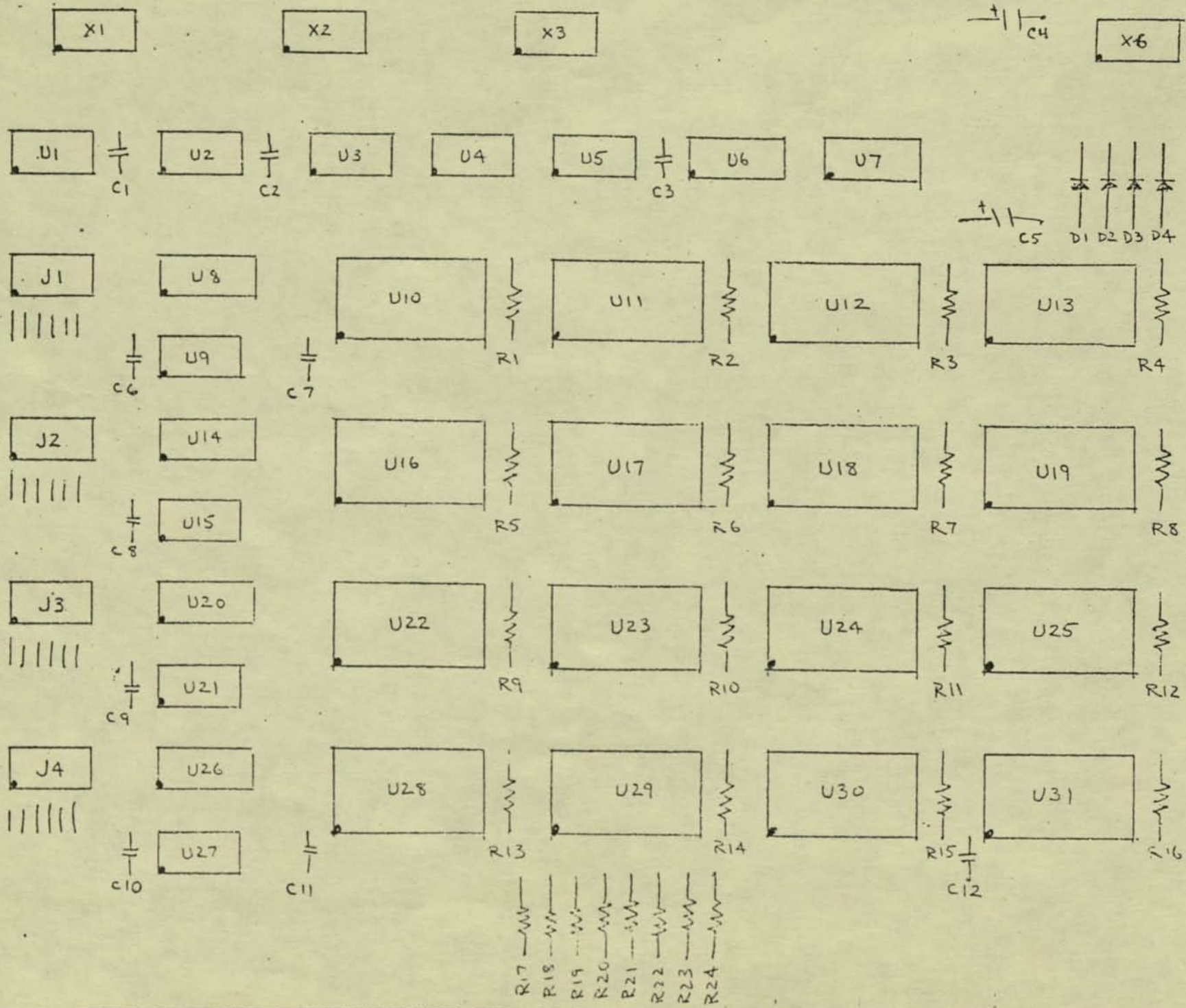

CHANGE NOTES
1. C1,C2,C3,C6,C8,C9,C10 BYPASS +5V-GND
2. C7,C11,C12 BYPASS -9V-GND
7400
R19 +5V
X2/1 A0
X2/3 A1
X2/8 A2
X2/9 A3
X2/2 A4
X2/6 A5
X2/5 A6
X2/4 A7
A00+
A01+
A02+
A03+
A04+
A05+
A06+
A07+
R18
R17
R20
R21
R22
R23
R24
10K
U3
U7
DM8097
X2/7 A8
X1/2 A9
A08+
A09+
X1/14 Φ2+
X1/3 VMA+
X1/1 R/W
SELA-
SELB-
SELC-
SELD-
U5
7420
D00+
D01+
D02+
D03+
D04+
D05+
D06+
D07+
U6
U7
D0
D1
D2
D3
D4
D5
D6
D7
7404
X1/13 A10
X1/12 A11
X1/11 A12
X1/10 A13
X1/4 A14
X1/5 A15
A10+
A10-
A11+
A11-
A12+
A12-
A13+
A13-
A14+
A14-
A15+
A15-
1N4001
X6/5
X6/12
-12v
-9v
D1 D2 D3 D4
C5 33μf 20v
X6/1
X6/2
X6/8
X6/9
X6/6
X6/7
X6/13
X6/14
+5v
C4 33μf 20v
RCM/1
SHEET 1 OF 5

CHANGE NOTES

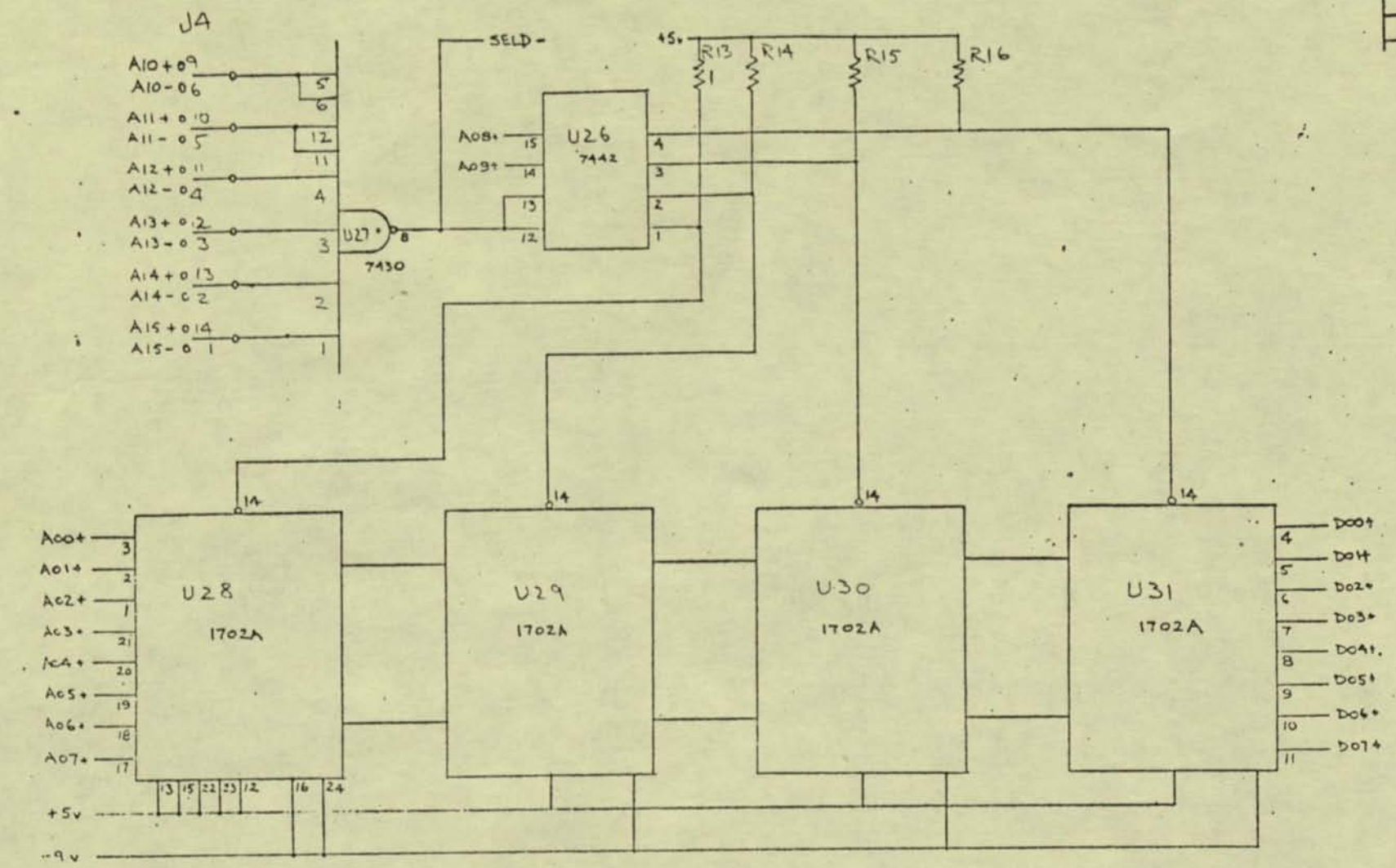


PROM

SHEET 2 OF 5


X1
X2
X3
C4
X6
U1
C1
U2
C2
U3
U4
U5
C3
U6
U7
C5
D1
D2
D3
D4
J1
U8
U9
C6
C7
U10
R1
U11
R2
U12
R3
U13
R4
J2
U14
U15
C8
U16
R5
U17
R6
U18
R7
U19
R8
J3
U20
U21
C9
U22
R9
U23
R10
U24
R11
U25
R12
J4
U26
U27
C10
C11
U28
R13
U29
R14
U30
R15
C12
U31
R16
R17
R18
R19
R20
R21
R22
R23
R24

CHANGE NOTES



PROM

SHEET 5 OF 5

PARTS FOR RCM/1 EHD (PCB ASSY)

	QTY	TYPE	DESCRIPTION	LOC
1	2	7404	IC HEX INVERTER	U1, U2
2	2	7408	IC QUAD 2-INP AND	U3, U4
3	1	7420	IC DUAL 4-INP NAND	U5
4	4	7430	IC 8-INP NAND	U9, U15, U21, U27
5	4	7442	IC 4-10 DECODER	U8, U14, U20, U26
6	2	DM8097	IC HEX BUS DRIVER	U6, U7
7	16	1702A	IC 256X8 PROM	U(10-13), U(16-19), U(22-25), U(28-31)
8	4	IN4001	50v RECTIFIER DIODE	D1, D2, D3, D4
9	2	33µF/20v	CAPACITOR, ELECTROLYTIC	C4, C5
10	10	0.1µf/50v	CAPACITOR, CERAMIC	C1, C2, C3, C(6-12)
11	24	4.7K, ¼w, 10%	RESISTOR, CARBON COMP	R(1-24)
12	4, 4	14-pin sockets (DIP)		X1, X2, X3, X6, (J1, J2, J3, J4)
13	1	RCM/1 EHD	PCB	
14	16	24-pin DIP sockets		U(10-13), U(16-19), U(22-25), U(28-31)

UNDERLINED ITEMS NOT SUPPLIED

PROM

FOR MY USE ONLY

For my bookkeeping please:

One board per order form (copy it if necessary)

One check per order

No charges or COD orders- cash with order!

Please send me ONE (1) ROM/1 board wired to the following addresses:

BANK	STARTING ADDRESS Binary	Hex
A	_ _ _ _ _ _ 0 0 0 0 0 0 0 0 0 0	_ _ 0 0
B	_ _ _ _ _ _ 0 0 0 0 0 0 0 0 0 0	_ _ 0 0
C	_ _ _ _ _ _ 0 0 0 0 0 0 0 0 0 0	_ _ 0 0
D	_ _ _ _ _ _ 0 0 0 0 0 0 0 0 0 0	_ _ 0 0

Please remember that addresses below 1000_X and above $E000_X$ are assigned to memory and I/O devices and that SPHERE Corporation reserves the right to modify at any time or add to the addresses of the I/O devices.

Initial orders will be shipped within 10 days. Unshipped orders may be canceled after 45 days. Allow an extra 30 days for non-guaranteed checks to clear.

CORRESPONDENCE ADDRESS: _____

_____

_____

ATTN: _____ ZIP _____

SHIPPING ADDRESS: _____

_____

_____

ATTN: _____ ZIP _____

Send this form and $156.00 (plus 6% California sales tax if applicable) to:

Ernest Dixon
P. O. Box 3102
Culver City, Ca. 90230

PROM

HELLO

PROBLEMS WITH THE 16K MEM BOARD

1) NO JUMPER NEAR D2 AND R10 TO CONNECT -5VDC TO MEMORY BANK
SOLUTION:PUT IN JUMPER

2) BURNED ETCH NEAR E19 PIN 18;+12VDC DOES NOT GET TO + SIDE OF C8,C9,C10,C11
SOLUTION:PUT WHITE JUMPER ON BACK

3) NO CONTACT FROM CHIP TO SOCKET AT PIN 11 ON E5(NO REFRESH ON Q3 BASE)
SOLUTION:BEND PIN 11 ON CHIP AT E5

4) NO CONTACT FROM CHIP TO SOCKET AT PIN 7(GRND) ON E5
SOLUTION:BEND PIN 7 ON CHIP AT E5

COMMENT:TEXAS INSTRUMENT'S SOCKETS ARE BETTER

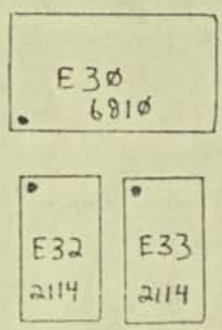
5) 2 WATT RESISTORS AND 100 MFD CAPACITORS TO REPLACE THE 1 WATT AND 47MFD TO GIVE BETTER OPERATION.

WARREN W.WEIMER
12-6-77

8/17/79

MOD: Upgrade crt board from 512 to 1024 bytes.

1) Add 2 18-pin wire wrap sockets just below E30 on the CRT board and label them E32 and E33.



2) Remove 4 6810's at E14, E20, E25 and E30.
3) Connect E32-18 to E33-18 to E30-24 (Vcc).
4) Connect E32-9 to E33-9 to E30-1 (Gnd).
5) Connect E32-5 to E33-5 to E30-23 (A0).
6) Connect E32-6 to E33-6 to E30-22 (A1).
7) Connect E32-7 to E33-7 to E30-21 (A2).
8) Connect E32-4 to E33-4 to E30-20 (A3).
9) Connect E32-3 to E33-3 to E30-19 (A4).
10) Connect E32-2 to E33-2 to E30-18 (A5).
11) Connect E32-1 to E33-1 to E30-17 (A6).
12) Connect E32-17 to E33-17 to E30-10 (A7).
13) Connect E32-16 to E33-16 to E30-13 (A8).
14) Connect X1-2 to E5-13 (A9).
15) Connect E32-15 to E33-15 to E5-12 (A9).
16) Connect E32-14 to E30-2 (D0).
17) Connect E32-13 to E30-3 (D1).
18) Connect E32-12 to E30-4 (D2).
19) Connect E32-11 to E30-5 (D3).
20) Connect E33-14 to E30-6 (D4).
21) Connect E33-13 to E30-7 (D5).
22) Connect E33-12 to E30-8 (D6).
23) Connect E33-11 to E30-9 (D7).
24) Connect E32-10 to E33-10 to E30-16 (R/W).
25) Connect E32-8 to E33-8 to E32-9 (CS always active).
26) Open land going from X1-2 to E13-13 at E13 end.
27) Connect E13-13 to E13-7 (this allows selection of board for addresses E000 - E3FF).
28) Install 2 2114 static rams at E32 and E33.
29) NOTE: If you are going to continue with the 64 char mod, then ignore this step, else,
Connect E5-14 to E5-8 (this ties A'9 inactive so we can access the 1st 512 bytes).

8/17/79

MOD: CHANGE CRT BOARD FROM 32 TO 64 CHARS/LINE

1) IF STEP 29 WAS PERFORMED DURING MEMORY UPGRADE, REMOVE WIRE FROM E5-14 TO E5-8.

2) CUT LAND FROM E10-11 TO E4-14 AT E4 END (ORIG A'8).

3) CUT LAND FROM E10-8 TO E4-5 AT E4 END (ORIG A'7).

4) CUT LAND FROM E10-9 TO E4-2 AT E4 END (ORIG A'6).

5) CUT LAND FROM E10-12 TO E3-11 AT E3 END (ORIG A'5).

6) CUT LAND FROM E6-8 TO E3-14 AT E3 END (ORIG A'4).

7) CUT LAND FROM E6-9 TO E3-5 AT E3 END (ORIG A'3).

8) CUT LAND FROM E11-11 TO E6-1 AT E6 END (CARRY BIT FROM E11).

9) CONNECT E10-11 TO E5-14 (NEW A'9).

10) CONNECT E10-8 TO E4-14 (NEW A'8).

11) CONNECT E10-9 TO E4-5 (NEW A'7).

12) CONNECT E10-12 TO E4-2 (NEW A'6).

13) CONNECT E6-8 TO E3-11 (NEW A'5).

14) CONNECT E6-9 TO E3-14 (NEW A'4).

15) CONNECT E6-1 TO E6-12 (ADD NEW COUNTER STAGE FOR A'3).

16) CONNECT E6-14 TO E11-11 (CARRY BIT FROM E11 TO NEW COUNTER STAGE).

17) CONNECT E6-1 TO E3-5 (NEW A'3).

18) REMOVE C37 AND REPLACE WITH A 4.7 PF CAPACITOR (THIS ALLOWS DENSITY TO BE INCREASED TO ACCOMMODATE 64 CHARS).

19) PLACE A 10K RESISTOR ACROSS R5 (MAKES CURSOR DISPLAY MORE RELIABLE).
NOTE: PERFORM STEP 19 ONLY IF YOUR CURSOR GIVES YOU PROBLEMS WHEN BACKED OVER A PREVIOUSLY TYPED CHARACTER. CHECK FOR THIS SYMPTOM BY GOING INTO EDIT MODE AND TYPING IN ABOUT 10 LETTER H'S. NOW BACK THE CUROSR OVER THE H'S AND IF WHEN THE CURSOR SHOULD BE ON, YOU INSTEAD SEE A 7, THEN DO STEP 19. WHAT YOU ARE ACTUALLY LOOKING AT IS THE COMPLEMENTED CHARACTER WITHOUT THE CURSOR FUNCTION, AND THIS PROBLEM WILL AFFECT ALL CHARACTERS.

20) AS THIS CHANGE WILL AFFECT ALL 3 ADJUSTMENT POTS, A DEFINITE ADJUSTMENT PROCEDURE CANNOT BE GIVEN. HOWEVER, I FOUND THE DENSITY POT (R21) TO REQUIRE THE MOST MOVEMENT, HENCE IT SHOULD BE THE FIRST ONE ADJUSTED.

11

THIS KEYBOARD WILL GENERATE A FULL ASCII CHARACTER SET HOWEVER, THIS IS NOT EASY TO USE IN HEX CODING FROM THE KEYBOARD.

BY MAKING THE JUMPERS SHOWN BELOW OR ON DRAWING A MODIFIED ASCII VERSION OF THE KEYBOARD IS ACHIEVED. THIS VERSION IS MUCH EASIER TO USE IN HEX CODING (THIS VERSION REQUIRES NO SHIFTING OF ALPHA CHARACTERS)

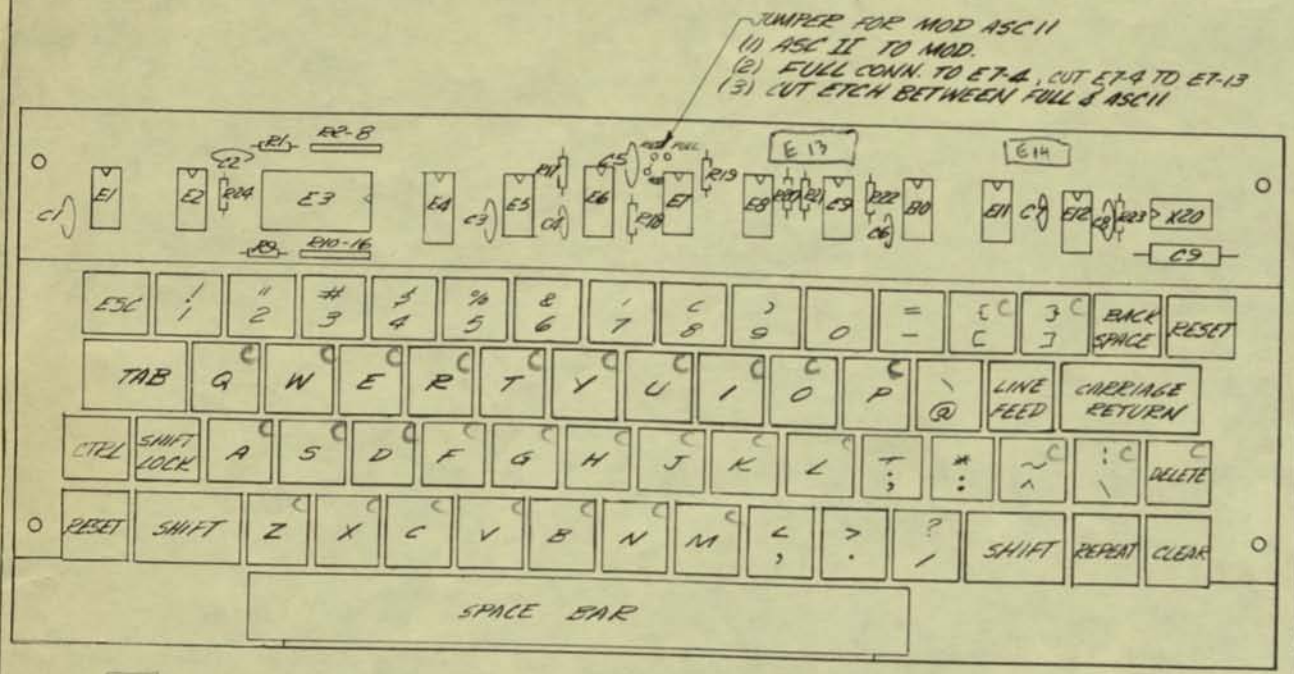
PRINTED CIRCUIT BOARDS THAT HAVE A PART NUMBER OF 00026 WITH NO REVISION SHOULD BE JUMPERED AS FOLLOWS:

JUMPER FOR MODIFIED ASCII

(1) 'ASCII'to E7-4 (use pin), Cut Etch E7-4 to E7-13 (under R19)

(2) 'MOD' to 'FULL'

(3) Cut Etch Between FULL & ASCII (Backside)



[ ] → valid CTRL key

ITEM NO.	QTY REQ'D	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO.	SPECIFICATION	MATERIAL OR NOTE
24	REF	SCHEMATIC	C000027		
23	1	PC BOARD	E000028		
22	1	ALPHANUMERIC KEYBOARD	873-10184		HI-TEK
21	1	CAPACITOR 100 MF 10V	DDM-103	C9	CRL
20	1	CAPACITOR .01 MF		C7	
19	1	CAPACITOR 100 PF		C6	
18	1	CAPACITOR .2 MF		C5	
17	1	CAPACITOR .001 MF		C4	
16	4	CAPACITOR .1 MF 50V	CK104	C1, C2, C3, C8	CRL
15	1	RESISTOR 4.7K ¼ W	RC07GF472	R18	
14	1	RESISTOR 10K ¼ W	RC07GF103	R17	
13	2	RESISTOR NETWORK	760-81-R33K	R2-8, R10-16	CTS
12	8	RESISTOR 3.3K ¼ W	RC07GF332	R1, 9, 19-24	
11	1	SOCKET, 14 PIN IC	316-AG39D	X20	AUGAT
10	1	DUAL D FLIP FLOP	SN7474N	E12	
9	1	QUAD 2 INPUT NAND	SN7400N	E11	
8	1	QUAD 2 INPUT AND	SN7408N	E10	
7	1	HEX INVERTER	SN7404N	E9	
6	1	QUAD EXCLUSIVE-OR	SN7486N	E7	
5	1	DUAL ONE SHOTS	SN74123N	E6	
4	1	BCD TO DECIMAL DECODER	SN7442N	E4	
3	1	16 TO 1 MULTIPLEXER	SN74150N	E3	
2	2	HEX BUFFER/DRIVER	SN7407N	E2, E8	
1	2	4 BIT BINARY COUNTER	SN7493N	E1, E5	
		ASSEMBLY			

LIST OF MATERIAL

CONTRACT NO.

DRAWN BY Howe ITHAKA '76

CHECKED

MECH

ELECT.

PROJ ENGR.

APPROVED Date

APPROVED (By Others) Date

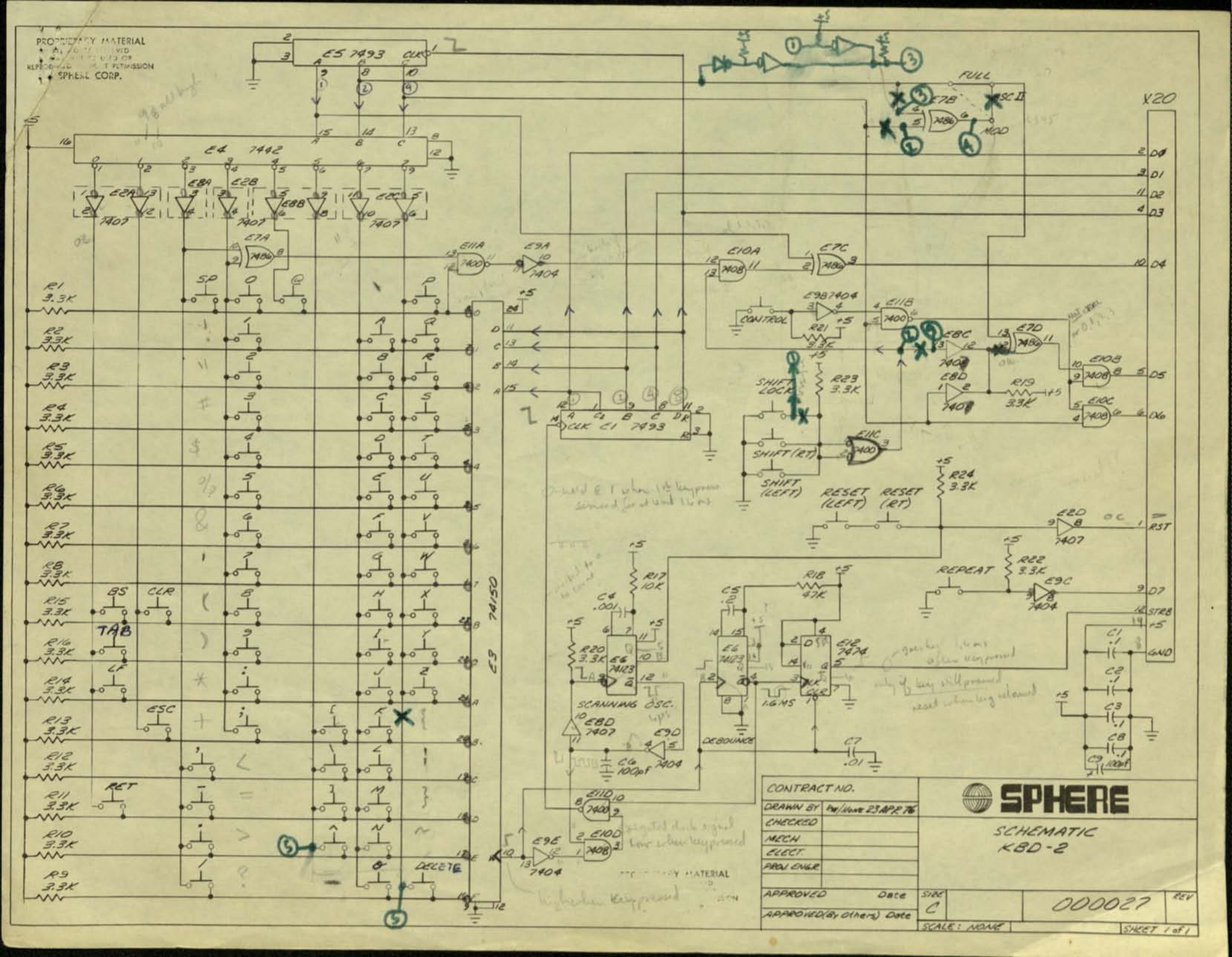
SPHERE

KBD/2

ASSEMBLY

SIZE C | 000025 | CSV

SCALE: NONE | Sheet 1 OF 1



CONTRACT NO.	
DRAWN BY	[illegible] 23 APR 76
CHECKED	
MECH	
ELECT.	
PROJ ENGR.	
APPROVED Date	
APPROVED(By Others) Date	

SPHERE

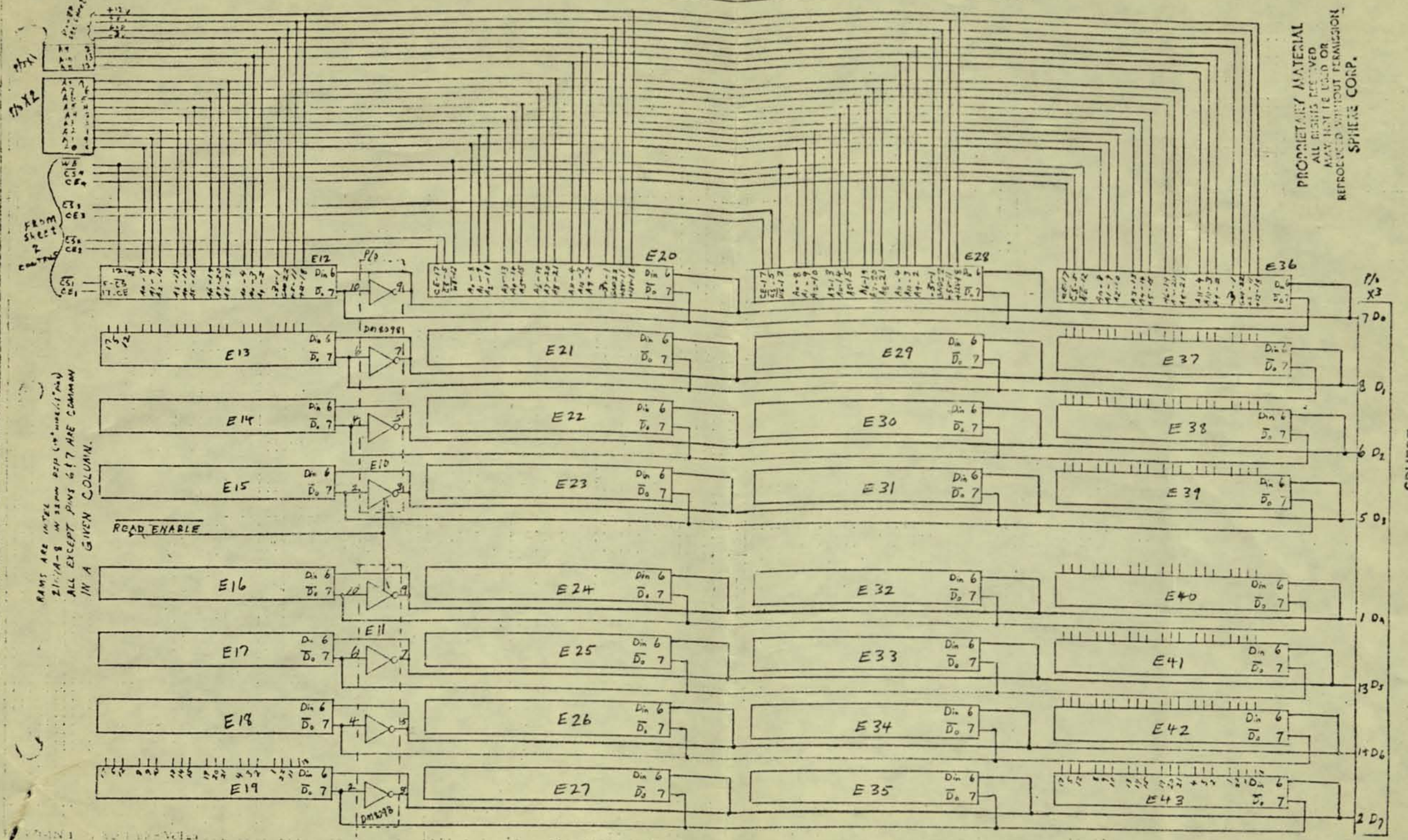
SCHEMATIC
KBD-2

SIZE			REV
C		000027	
SCALE: NONE			SHEET 1 of 1

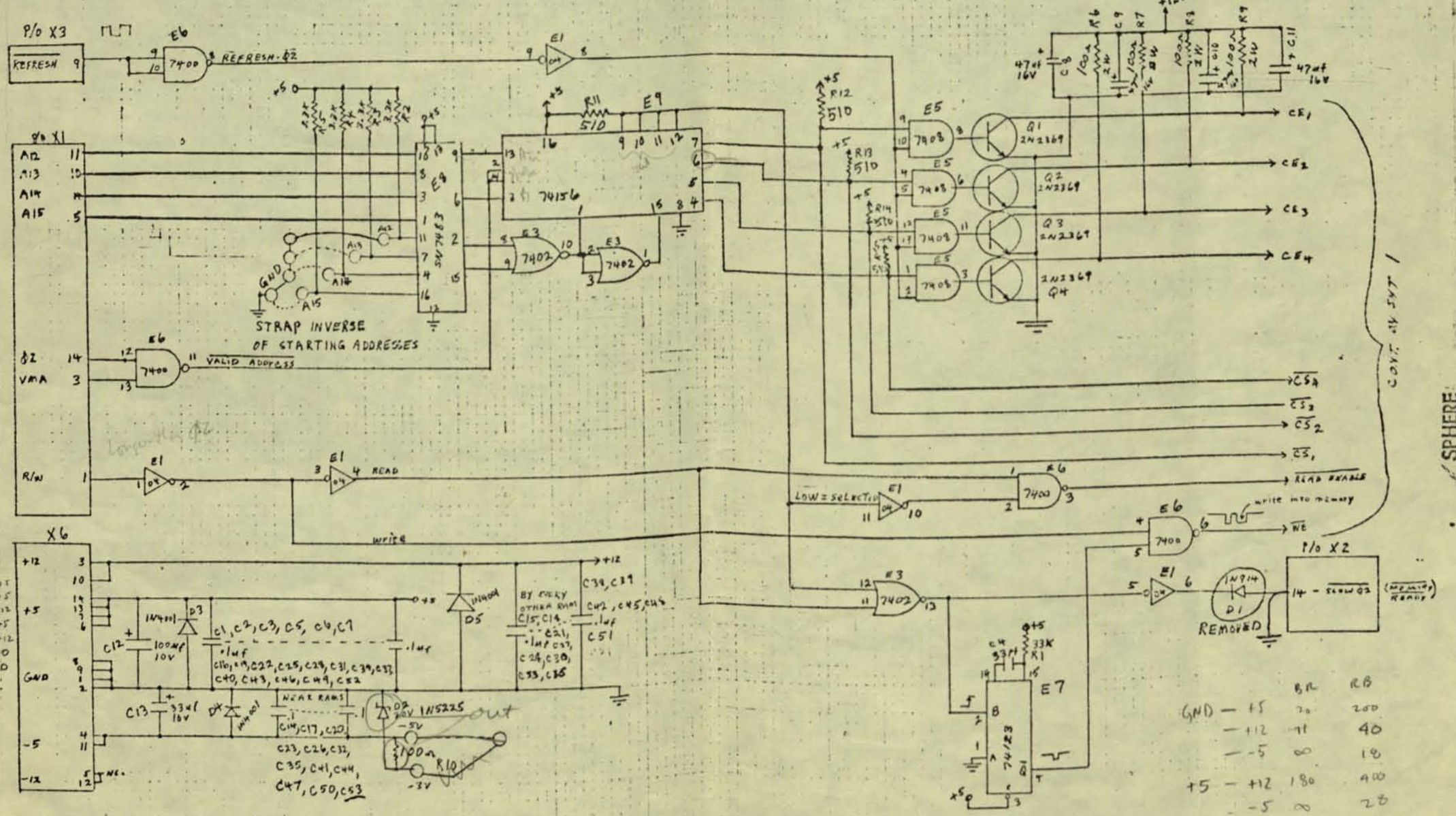
BY R.L. Tyler DATE 29 JULY '75 SUBJECT MEMORY MODULE (16K X 9 DYNAMIC RAMS) SHEET NO. 1 OF 3

CHKD. BY DATE REV. OCT 1, 1975 JOB NO. MEM/1 MODULE

RIGHT HAND SIDE OF P.C. BOARD.



BY M.C.T. DATE 29 JULY '75 SUBJECT 16K X 8 DYNAMIC MEMORY SHEET NO. 2 OF 3
CHKD. BY DATE JOB NO. MEM/11
Rev. Oct. 1, 1975 MODULE

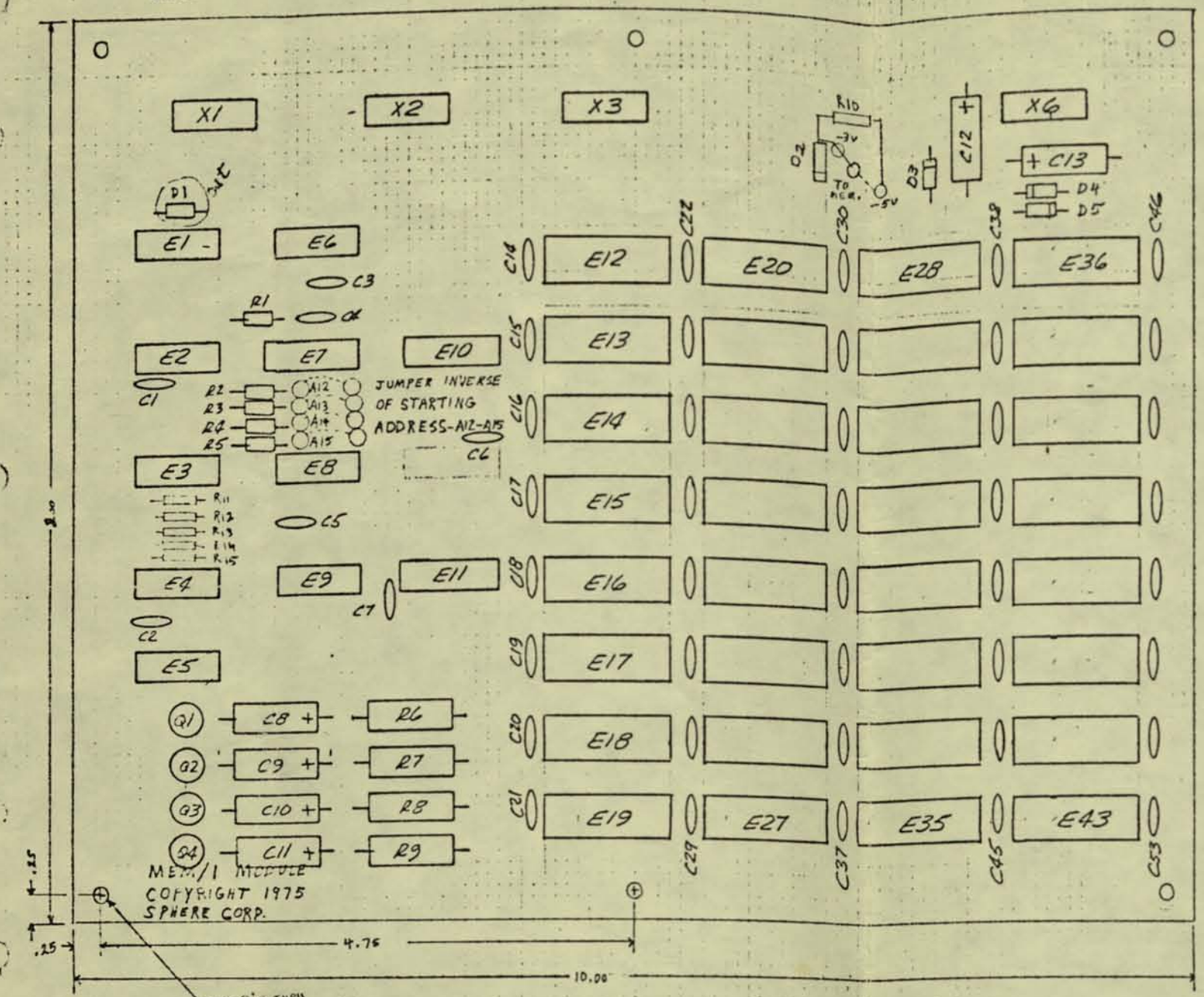


	BR	RB
GND — +5	70	200
— +12	71	40
— -5	∞	18
+5 — +12	180	400
— -5	∞	28
+12 — -5	∞	18

+5 bus get +5, -12, +5
+12 bus get 0, -5
GND get 0, +12
-5 get +12, -12

SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY _____ DATE _____ SUBJECT MEM/1 SHEET NO
CHKD. BY _____ DATE _____ JOB NO.

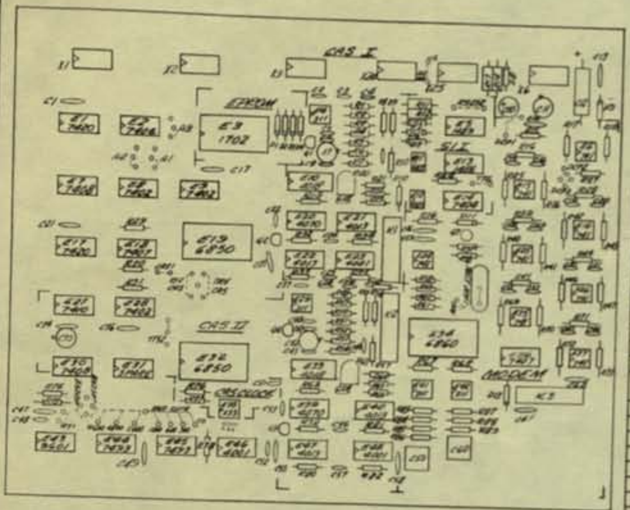


ITEM	QTY	PART No.	DESCRIPTION	DESIGNATION
1	1	MEM/1	P.C. BOARD - 8X10"	MEM/1
2	1	SN74123	ONE SHOT	E7
3	1	SN7400	NAND GATES	E6
4	1	SN7402	NOR GATES	E3
5	1	SN7404	INVERTER	E1
6	1	SN7408	AND GATE	E5
7			SPARE	E2, E4
8	1	SN7483	FULL ADDER	E8
9	2	DM8098	BUFFER	E10, E11
10	1	SN74156	2 LINE TO 4 LINE DECODER	E9
11	32	ZA-6249	4K x 1 DYNAMIC RAM	E12 - E43
12				
13	4	2N2369A	TRANSISTOR	Q1 - Q4
14	3	IN4001	DIODE	D3 - D5
15	1	IN914	DIODE	D1
16	1	IN5225B	3.0V ZENER	D2
17	4		RESISTOR, 100Ω 2W	R6 - R9
18	9		" 3.3K ¼W	R2 - R5 (R11 - R15) -510Ω
19	1		" 33K "	R1
20	1		" 1.2Ω ¼W	R10
21	1		CAPACITOR CERAM 33PF	C4
22	46		" CERAMIC .1μF	C1, C2, C3, C5, C6, C7, C14 - C53
23	1		" 100MF - 10VDC	C12
24	5		" 47MF 16V	C8 - C11, C13
25	4	214-AG39D	16 PIN SOCKET	X1 - X3, X6

SPHERE
791 SOUTH 500 WEST
BOUNTIFUL UTAH 84010

PROBLEM		NUMBER
PROGRAMMER	DATE	PAGE OF

NO.	LOCATOR 1 6	WORD				ID 67 72	COMMENTS
		Op	ADDRESS	TAG	DECREMENT		



QTY -05	QTY -04	QTY -03	QTY -02	ITEM NO	QTY -01	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO.	SPECIFICATION	MATERIAL OR NOTE
				86	UR	SCHEMATIC	D000023		
				85	1	PRINTED CIRCUIT BOARD	E000024		
—	1	—	—	84	—	CRISTAL	1MHZ	XTAL	
—	1	—	—	83	—	RELAY	220-100-12	K3	TRIPIDGE
1	—	1	—	82	—	RELAY	220-100-12	K1,K2	TRIPIDGE
—	1	—	—	81	—	TRANSISTOR	2N2369	Q3	
1	—	1	—	80	—	TRANSISTOR	MPS6521	Q2,Q5	
1	—	1	—	79	—	TRANSISTOR	MPS6523	Q1,Q4	
—	2	—	—	78		CAPACITOR, 1uf		C59,60	
—	—	—	—	77	1	CAPACITOR, 100PF 1000V N750	DTE-100	C48	
—	—	—	—	76	1	CAPACITOR, 33PF 1KV	DD-101	C47	CRL
1	—	1	—	75	—	CAPACITOR, 470PF 1KV	DD-471	C24,56	CRL
3	12	3	—	74	2	CAPACITOR, .01uf 100V	DDM-103	[illegible]	CRL
—	—	—	—	73	1	CAPACITOR, 100uf 10V		C12	CRL
1	—	1	—	72	—	CAPACITOR .0062uf 1KV	AE.0062 1KV	C9,44	SIEMENS
1	—	1	—	71	1	CAPACITOR 100PF 1KV	DD-101	C8,43,52	
2	—	1	—	70	2	CAPACITOR 47uf 16V	EX 47/16	C7,10,11,35,42	
2	—	2	—	69	—	CAPACITOR, .001 1KV	DD-102	C5,6,40,41	ROE
1	—	1	—	68	—	CAPACITOR, .047uf 50V	AE.047M 50V	C4,39	CRL
1	—	1	—	67	—	CAPACITOR, .0015uf 1KV	DD-152	C3,38	
1	—	1	—	66	—	CAPACITOR, 47uf 16V		C2,37	CRL
3	5	—	—	65	7	CAPACITOR, .1uf 50V	CK-104	[illegible]	CRL
—	1	—	—	64	—	RESISTOR, 680Ω ¼W	RC07GF681	R89	
—	3	—	—	63	—	RESISTOR, 1.5K ¼W	RC07GF152	R81-85	
—	—	—	—	62	—	RESISTOR	RC07GF[illegible]	R74	
—	1	—	—	61	—	RESISTOR 57.6K ¼W 1%		R73	
—	1	—	—	60	—	RESISTOR 255K ¼W 1%		R72	
—	1	—	—	59	1	RESISTOR 887K ¼W 1%		R71,75	
—	1	—	—	58	—	RESISTOR 953K ¼W 1%		R70	
—	1	—	—	57	—	RESISTOR, 619Ω ¼W 1%		R69	
—	1	—	—	56	—	RESISTOR, 330K ¼W	RC07GF334	R59	

LIST OF MATERIALS OR PARTS

QTY -05	QTY -04	QTY -03	QTY -02	ITEM NO	QTY -01	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO.	SPECIFICATION	MATERIAL OR NOTE
—	2	—	—	55	—	RESISTOR, 1MEG ¼W	RC07GF105	R59,60	
—	1	—	—	54	—	RESISTOR, 237K ¼W 1%		R47	
—	1	—	—	53	—	RESISTOR, 1.24K ¼W 1%		R46	
—	1	—	—	52	—	RESISTOR, 8.66K ¼W 1%		R45	
—	1	—	—	51	—	RESISTOR 21.5K ¼W 1%		R44	
—	1	—	—	50	—	RESISTOR 845Ω ¼W 1%		R42	
—	1	—	—	49	—	RESISTOR 210K ¼W 1%		R41	
—	1	—	—	48	—	RESISTOR, 324Ω ¼W 1%		R40	
—	1	—	—	47	—	RESISTOR, 18.7K ¼W 1%		R35	
—	3	—	—	46	—	RESISTOR, 510Ω ¼W	RC07GF511	R36,47,48	
1	—	1	—	45	—	RESISTOR, 100K ¼W	RC07GF104	R31,80	
2	—	2	—	44	—	RESISTOR, 10K ¼W	RC07GF103	[illegible]	
1	—	1	—	43	—	RESISTOR, 20K ¼W	RC07GF203	R34,R81	
—	2	—	—	42	—	RESISTOR, 165K ¼W 1%		R26,43	
—	1	—	—	41	—	RESISTOR, 267Ω ¼W 1%		R25	
—	—	—	1	40	—	RESISTOR, 1.2K ¼W	RC07GF122	R24	
1	—	1	—	39	—	POTENTIOMETER 2.0K	3353P	R20,63	BOURNS
1	—	1	—	38	—	RESISTOR, 100Ω ¼W	RC07GF101	R13,62	
1	2	1	—	37	—	RESISTOR 22K ¼W	RC07GF223	R9,21,27,[illegible]	
—	2	—	—	36	—	RESISTOR, 15K ¼W	RC07GF153	R16,28	
—	1	—	—	35	—	RESISTOR, 620Ω ¼W	RC07GF621	R15	
—	—	—	—	34	2	RESISTOR, 510Ω 1W	RC32GF511	R13,R14	
—	—	—	1	33	—	RESISTOR, 820Ω 1W	RC32GF821	R12	
—	—	—	1	32	—	RESISTOR, 1.1K 1W	RC32GF112	R11	
1	6	1	0	31	3	RESISTOR, 1K ¼W	RC07GF102	[illegible]	
1	1	1	0	30	—	RESISTOR, 4.7K ¼W	RC07GF472	[illegible]	
1	—	1	0	29	3	RESISTOR, 3.3K ¼W	RC07GF332	R8,33,61	
3	—	3	0	28	1	RESISTOR, 47K ¼W	RC07GF473	[illegible]	
4	—	4	0	27	0	RESISTOR, 33K ¼W	RC07GF333	[illegible]	
2	—	2	0	26	0	RESISTOR, 2.2K ¼W	RC07GF222	R1,48,73	
1	2	1	3	25	7	DIODE	1N4001	CR1-D13	
1	1	—	—	24	2	24 PIN IC SOCKETS	AH4623D	E12,13,32,34	
—	—	—	—	23	6	14 PIN IC SOCKETS	AH4633D	E1-3,6,24,25	
—	—	—	—	22	2	4 BIT BINARY COUNTER	SN7493N	E44,45	
—	—	—	—	21	1	ONE SHOT	9601(PHILIPS)	E43	
—	—	—	—	20	1	PRECISION TIMER	SN72555P	E38	
—	1	—	—	19		MODEM	MC6860	E34	
—	—	—	—	18		SPARE		E31	
1	—	—	—	17	—	TRIPLE 3 INPUT NAND GATE	SN7410N	E27	
1	—	1	—	16	1	NOR GATE	CD4001B	E21,46,48	
2	—	2	—	15	—	DUAL D FLIP FLOP	CD4013A	E10,22,40,47	
1	—	1	—	14	—	EXCLUSIVE OR GATE	CD4070A	E20,39	
1	—	—	—	13	1	ACIA	MC6850	E9,32	
1	1	1	0	12	1	HEX BUFFER	SN7407N	E18,36	
—	—	—	1	11	—	QUAD TRANSMITTER-RS232	MC1488N	E19	
—	—	—	2	10	—	PHOTO COUPLER	4N33	E11,E12	
1	—	1	—	9	—	COUNTER	CD4018A	E10,33	
1	—	—	—	8	2	QUAD 2-INPUT NOR GATE	SN7402N	E23,28	
1	—	—	—	7	1	QUAD 2 INPUT AND GATE	SN7408N	E7,30	
—	8	—	—	6	—	OPERATIONAL AMPLIFIER	LM741	[illegible]	
—	—	—	1	5	—	QUAD RECEIVER-RS232	MC1489	E5	
1	2	1	—	4	—	VOLTAGE COMPARATOR	LM311N	E25,41,42	
—	—	—	—	3	4	E PROM 256X8 BIT	MM1702A	E3	
—	—	—	1	2	1	HEX INVERTER	SN7404N	E24	
—	—	—	—	1	2	8 INPUT NAND GATE	SN7430N	E17	
-05	-04	-03	-02		-01	ASSEMBLY			

LIST OF MATERIALS OR PARTS

DASH NO.	TITLE	REVISION
05	AUDIO CAS INTERFACE II	d
04	MODEM	c
03	AUDIO CAS INTERFACE I	b
02	SERIAL INTERFACE I	a
01	STD	—

* SPECIAL ITEM CONTROLED BY SOFT WARE DEPT.

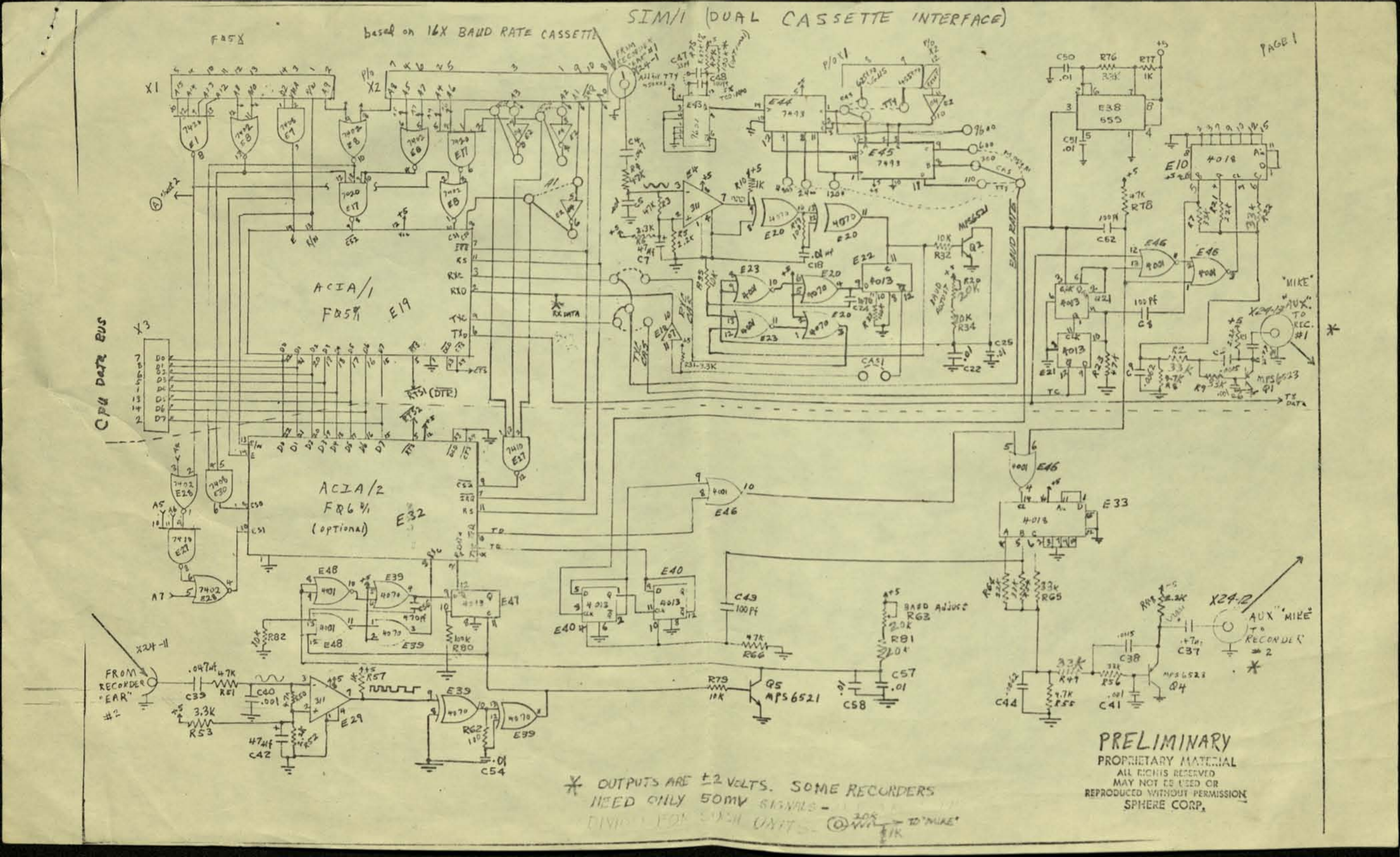
DRAWN BY	28 JAN 76
CHECKED BY	
R&D	
QA CONT	
MFG ENG	
PROD ENG	
APPROVAL	
APPROVAL	
FINAL APPROVAL	

SPHERE

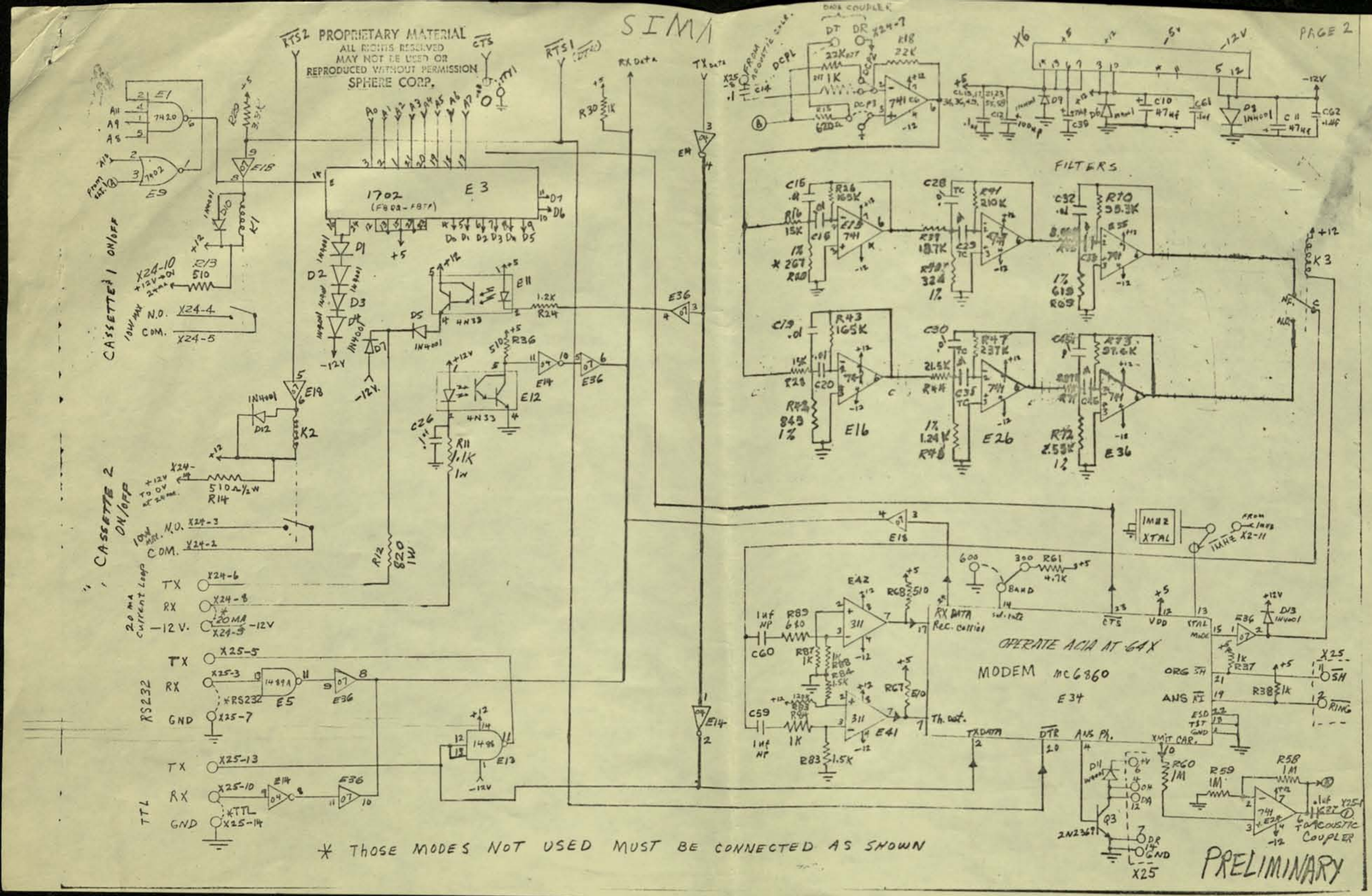
SIM/1 ASSEMBLY

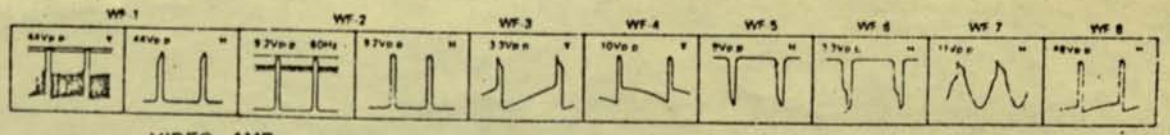
SIZE D | 000022 | REV A

SCALE NONE | SHEET 1 OF 1


SIM/1 (DUAL CASSETTE INTERFACE)
based on 16X BAUD RATE CASSETTE
PAGE 1
FROM RECORDER "EAR" #1
FROM RECORDER "EAR" #2
CPU DATA BUS
ACIA/1
ACIA/2
(optional)
AUX "MIKE" TO REC. #1
AUX "MIKE" TO RECORDER #2
"MIKE"
TX DATA
RX DATA
CAS1
BAUD ADJUST
* OUTPUTS ARE ±2 VOLTS. SOME RECORDERS NEED ONLY 50mV SIGNALS - DIVIDER FOR EACH UNIT
PRELIMINARY
PROPRIETARY MATERIAL
ALL RIGHTS RESERVED
MAY NOT BE USED OR
REPRODUCED WITHOUT PERMISSION
SPHERE CORP.

SIM1





VIDEO AMP.
Q913~Q915 2SC536×3

PICTURE TUBE
V101 24QD84
or 24QD84A

VIDEO OUTPUT
Q101 2SC65Y

VIDEO

HIGH ⟷ 75Ω

CONTRAST

SYNC. SEP.
Q102 2SA201

VERT. OSC.
Q103 2SC536

VERT. DRIVE
Q104 2SC536

BRIGHT

VERT. OUTPUT
Q105 2SC1024
or 2SD315
or 2SD331

L102
DEFLECTION

V-HOLD

HEIGHT

V-LIN

SEESAW SWITCH
SW1

POWER RECT.
D112 PR9d 522/1

POWER TRANS.

PHASE DET.
D104, D105 1S188×2

H.V. RECT.

VERT. OUTPUT
Q106 2SB474

FUSE 1.6A SLOW BLOW

FUSE 0.75A SLOW BLOW

AC 120V 60Hz

B-ADJ.

REGULATOR
Q111 2SA608

HORIZ. DRIVE TRANS.

DAMPER
D106 DG-13
or DG-130

T102
HORIZ. OUTPUT TRANS.

H-FREQ

H-HOLD

REGULATOR
Q113 2SB407

REGULATOR
Q112 2SD30

PHASE INVERTER
Q107 2SC536

HORIZ. OSC.
Q108 2SC536

HORIZ. DRIVE
Q109 2SB187S

HORIZ. OUTPUT
Q110 2SB375A

NOTES
1. All resistance values in ohms K=1,000 M=1,000,000
2. Unless otherwise noted in schematic, all capacitors less than 1 are expressed in mfd, and the values larger than 1 are in pF.
3. Voltage reading taken with "VTVM" from point indicated to chassis ground. Tuner on unused channel, contrast at maximum, other control at normal, line voltage 120 volts.
4. All waveforms measured with strong signal input, contrast set to give normal picture and AGC line operating normally.
5. Voltage reading may vary 20%.
6. This is a fundamental circuit diagram. Some production changes may be made without revision of the diagram.

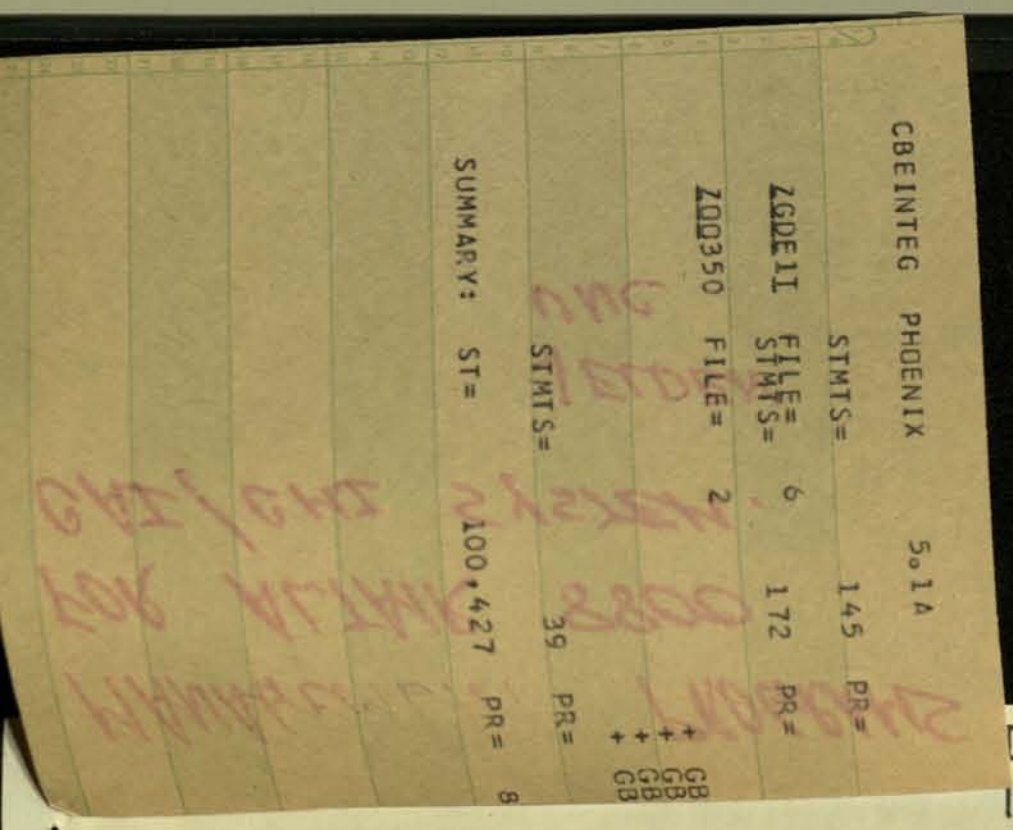
Dr. D. L. JELDEN
University Northern Colorado 49
Greeley, Colorado 80639
MENT SYSTEM

M.I. FILE

MANAGEMENT PROGRAMS
FOR ALTAIR 8800
CAI/CMI SYSTEM.

JELDEN
UNC

```
140 PRINT 'PUSH THE RETURN KEY TO CONTINUE.';
150 LINE INPUT A$
160 PRINT C$;
170 PRINT'TYPE IN COURSE NAME AND NUMBER.'
180 PRINT'FOR EXAMPLE: IA 281';
190 INPUT N$
200 PRINT
210 PRINT'LOAD C.M.I. DISK ON DRIVE #1 AND'
220 PRINT'PUSH THE RETURN KEY WHEN READY.';
240 LINE INPUT A$
260 PRINT C$;
270 PRINT'TYPE THE STUDENT INFORMATION AS'
280 PRINT'FOLLOWS;'
290 PRINT'UNDER TYPE:'
300 PRINT'----- -----'
310 PRINT'  S         UNC/SSN I.D. NUMBER'
320 PRINT'  L         STUDENT'S LAST NAME'
330 PRINT'  F         STUDENT'S FIRST NAME'
340 PRINT'  M         STUDENT'S MID INITIAL'
350 PRINT
360 PRINT'USE THE SPACE BAR TO ALIGN'
370 PRINT'THE CURSOR UNDER EACH FIELD'
383 PRINT 'PUSH THE RETURN KEY TO CONTINUE.';
385 LINE INPUT A$
387 PRINT C$;
390 PRINT'EACH LINE WILL BE GIVEN A'
400 PRINT'HEADING. BE CERTAIN TO RECORD'
410 PRINT'THE STUDENT'S C.A.I. I.D. NO.'
420 PRINT'AFTER EACH ENTRY. LET'S BEGIN.'
430 OPEN 'R',#1,N$,1
440 FIELD #1,9 AS S$,15 AS L$,7 AS F$,1 AS M$
450 FOR I=1 TO 24
460 FIELD #1,(I+7)*4 AS A$,2 AS Y$(I),2 AS Z$(I)
470 NEXT I
480 FOR I=1 TO 2046
490 PRINT'PUSH THE RETURN KEY TO CONTINUE.';
```



Dr. D. L. JELDEN
University Northern Colorado 49
Greeley, Colorado 80639

MICROSOFT BASIC LANGUAGE MANAGEMENT SYSTEM
(Altair Computer)

#1

```
LOAD"CMI LOAD",0
OK
LIST

10 DIM Y$(24),Z$(24) ' CREATE AND LOAD C.M.I. FILE
20 C$=CHR$(27) + ";"
30 PRINT C$;
40 PRINT"THE PURPOSE OF THIS PROGRAM IS"
50 PRINT"TO LOAD THE C.M.I. FILE. IT"
60 PRINT"SHOULD BE USED AT THE BEGINNING"
70 PRINT"OF EACH QUARTER. THE PROGRAM"
80 PRINT"ALSO INITIALIZES THE PRE-TEST"
90 PRINT"AND POST-TEST SCORES FOR EACH"
100 PRINT"STUDENT AS WELL AS DETERMINING"
110 PRINT"THE STUDENT'S C.A.I. I.D. NO."
120 PRINT"FOR CROSS-CHECKING THE USER'S"
130 PRINT"IDENTITY."
140 PRINT"PUSH THE RETURN KEY TO CONTINUE.";
150 LINE INPUT A$
160 PRINT C$;
170 PRINT"TYPE IN COURSE NAME AND NUMBER."
180 PRINT"FOR EXAMPLE: IA 281";
190 INPUT N$
200 PRINT
210 PRINT"LOAD C.M.I. DISK ON DRIVE #1 AND"
220 PRINT"PUSH THE RETURN KEY WHEN READY.";
240 LINE INPUT A$
260 PRINT C$;
270 PRINT"TYPE THE STUDENT INFORMATION AS"
280 PRINT"FOLLOWS;"
290 PRINT"UNDER TYPE:"
300 PRINT"----- -----"
310 PRINT"  S        UNC/SSN I.D. NUMBER"
320 PRINT"  L        STUDENT'S LAST NAME"
330 PRINT"  F        STUDENT'S FIRST NAME"
340 PRINT"  M        STUDENT'S MID INITIAL"
350 PRINT
360 PRINT"USE THE SPACE BAR TO ALIGN"
370 PRINT"THE CURSOR UNDER EACH FIELD"
383 PRINT "PUSH THE RETURN KEY TO CONTINUE.";
385 LINE INPUT A$
387 PRINT C$;
390 PRINT"EACH LINE WILL BE GIVEN A"
400 PRINT"HEADING. BE CERTAIN TO RECORD"
410 PRINT"THE STUDENT'S C.A.I. I.D. NO."
420 PRINT"AFTER EACH ENTRY. LET'S BEGIN."
430 OPEN "R",#1,N$,1
440 FIELD #1,9 AS S$,15 AS L$,7 AS F$,1 AS M$
450 FOR I=1 TO 24
460 FIELD #1,(I+7)*4 AS A$,2 AS Y$(I),2 AS Z$(I)
470 NEXT I
480 FOR I=1 TO 2046
490 PRINT"PUSH THE RETURN KEY TO CONTINUE.";
```

```
500 LINE INPUT A$
510 PRINT C$;
520 PRINT
530 PRINT"SSSSSSSSSLLLLLLLLLLLLLLLFFFFFFFM"
540 LINE INPUT B$
550 IF B$="" THEN END
560 FOR J=1 TO 24
570 LSET Y$(J) = MKI$(0)
580 LSET Z$(J) = MKI$(0)
590 NEXT J
600 LSET S$ = LEFT$(B$,9)
610 LSET L$ = MID$(B$,10,15)
620 LSET F$ = MID$(B$,25,7)
630 LSET M$ = RIGHT$(B$,1)
640 PUT #1,I
650 PRINT"STUDENT C.A.I. I.D. NO.=";I
660 PRINT"PLEASE RECORD THE C.A.I. "
670 PRINT"I.D. NUMBER AT THIS TIME"
680 NEXT I
690 END
OK
```

```
LOAD"CMI EXTD",0
OK
LIST

5 REM - CMI EXTD
10 DIM Y$(24),Z$(24) ' EXTEND AN ALREADY CREATED C.M.I. FILE
20 C$ = CHR$(27) + ";"
30 PRINT C$;
40 PRINT"THE PURPOSE OF THIS PROGRAM IS"
50 PRINT"TO EXTEND A C.M.I. FILE AFTER IT"
60 PRINT"HAS BEEN INITIALIZED. IT IS"
70 PRINT"NECESSARY TO KNOW THE LAST C.A.I."
80 PRINT"I.D. NO. THAT WAS ON THE C.M.I."
90 PRINT"FILE. IF THIS PROGRAM IS USED"
100 PRINT"WITH A C.A.I. I.D. NO. OF AN"
110 PRINT"ALREADY EXISTING STUDENT, HIS/"
120 PRINT"HER PRE-TEST/POST-TEST SCORES"
130 PRINT"WILL BE DESTROYED. IF IN DOUBT,"
140 PRINT"THEN USE THE CNTRL-C RESPONSE TO"
150 PRINT"STOP THE RUNNING OF THIS PROGRAM."
160 LINE INPUTA$:PRINT C$;
170 PRINT"BETTER SAFE THAN SORRY. ONCE A"
180 PRINT"MISTAKE IS MADE, IT IS HARD TO"
190 PRINT"RECOVER. SURE YOU WANT TO GO ON."
200 PRINT"ANSWER YES OR NO.";
210 INPUT A$: PRINT C$
220 IF A$<> "YES" THEN END
240 PRINT"TYPE IN COURSE NAME AND NUMBER."
250 PRINT"FOR EXAMPLE:  IA291";
260 INPUT N$: PRINT C$;
280 PRINT"LOAD C.M.I. DISK ON DRIVE #1 AND"
290 PRINT"PUSH THE RETURN KEY WHEN READY.";
300 UNLOAD 1
310 LINE INPUT A$: PRINT C$;
320 MOUNT 1
340 PRINT"TYPE THE STUDENT INFORMATION AS"
350 PRINT"FOLLOWS:"
360 PRINT"UNDER      TYPE:"
370 PRINT"  S        STUDENT I.D. NUMBER"
380 PRINT"  L        STUDENT'S LAST NAME"
390 PRINT"  F        STUDENT'S FIRST NAME"
400 PRINT"  M        STUDENT'S MIDDLE INITIAL"
410 LINE INPUT A$: PRINT C$
420 PRINT"EACH LINE WILL BE GIVEN A"
430 PRINT"HEADING. BE CERTAIN TO RECORD"
440 PRINT"THE STUDENT'S C.A.I. I.D. NO."
450 PRINT"AFTER EACH ENTRY. LET'S BEGIN."
460 OPEN "R",#1, N$,1
470 FIELD #1,9 AS S$,15 AS L$,7 AS F$,1 AS M$
480 FOR I=1 TO 24
490 FIELD #1,(I+7)*4 AS A$,2 AS Y$(I),2 AS Z$(I)
```

```
500 NEXT I
510 FOR I=1 TO 24
520 LSET Y$(I)=MKI$(0)
530 LSET Z$(I)=MKI$(0)
540 NEXT I
550 PRINT"INPUT THE LAST C.A.I. I.D. NO."
560 PRINT"ENTERED ONTO THIS C.M.I. FILE."
570 INPUT J: PRINT C$;
580 J = J + 1
590 FOR I=J TO 2046
600 PRINT
610 PRINT"SSSSSSSSSLLLLLLLLLLLLLLLFFFFFFFM"
620 LINE INPUT B$
630 IF B$="" THEN END
640 LSET S$=LEFT$(B$,9)
650 LSET L$=MID$(B$,10,15)
660 LSET F$=MID$(B$,25,7)
670 LSET M$=RIGHT$(B$,1)
680 PUT #1,I
690 PRINT"STUDENT C.A.I. I.D. NO. = I";
700 NEXT I
710 END
OK
```

(#3)

TO MANAGE ENTRANCE INTO A TEST FILE (PRE-TEST) OR (POST-TEST)

```
LOAD"CAI INTR",0
OK
LIST

1 REM N=CAI I.D. NUMBER
2 REM N$=STUDENT NAME (FIRST NAME)
3 REM S$=SOCIAL SECURITY NUMBER
4 C$=CHR$(27)+";"
5 PRINT C$;
10 PRINT"PLEASE ENTER YOUR CAI I.D. NUMBER"
11 PRINT"THIS IS NOT YOU UNC STUDENT"
12 PRINT"I.D. NUMBER, BUT IT'S THE"
13 PRINT"NUMBER THAT WAS ASSIGNED BY YOUR"
14 PRINT"INSTRUCTOR."
15 INPUT N
16 REM Y=PRETEST SCORES, Z=
17 REM POST TEST
18 OPEN "R",#1,"CMI FILE",1 ' CHANGE CMI FILE TO ACTUAL CLASS NUMBER
19 DIM Y$(24),Z$(24)
20 FIELD #1,9 AS S$,15 AS L$, 7 AS N$,1 AS M$
21 FOR I=1 TO 24
22 FIELD #1,(I+7)*4 AS A$,2 AS Y$(I),2 AS Z$(I)
23 NEXT I
24 GET #1,N
25 PRINT"PLEASE ENTER YOUR UNC STUDENT I.D."
26 PRINT"NUMBER"
27 INPUT A$
28 IF A$<>S$ THEN END
OK
```

(#4)

TO RECORD SCORE OF PRE-TEST

```
LOAD"PRETEST",0
OK
LIST

 1 C$=CHR$(27)+";": PRINT C$;
 5 I=CN*100/(NUMBER OF QUESTIONS)
 10 PRINT"'N$", YOUR SCORE ON THE PRE-TEST"
 20 PRINT"WAS "CN" OR "I"%. IF YOU MISSED MORE THAN"
 30 PRINT"ONE OF THE (?) QUESTIONS YOU MUST"
 40 PRINT"TAKE THE LESSON. TO RECORD THE SCORE"
 50 PRINT"OF YOUR PRE-TEST, TYPE 'R' AND PUSH"
 60 PRINT"THE RETURN KEY."
 70 INPUT R$:PRINT C$;
 80 IF R$<>"R" THEN 110
 85 IF R$="R" THEN 130
 110 PRINT"I DON'T KNOW WHAT YOU PUSHED, BUT"
 120 PRINT"IT WASN'T AN 'R', TRY AGAIN.":GOTO  70
 130 LSET Y$ (LESSON NUMBER)=MKI$ (I)
 140 PUT #1,N
 150 IF I=>80 THEN 170
 160 IF I< 80 THEN 340
 170 PRINT"YOUR SCORE ON THE PRE-TEST WAS"
 180 PRINT"HIGH ENOUGH TO ALLOW YOU TO SKIP"
 190 PRINT"THE LESSON. YOU MAY NOW EITHER"
 200 PRINT"TAKE THE LESSON OR THE POST-TEST."
 210 PRINT"TYPE 'A' FOR THE LESSON AND 'B' FOR"
 220 PRINT"THE POST-TEST."
 230 INPUT Z$
 240 IF Z$="A" THEN 300
 250 IF Z$="B" THEN 330
 260 PRINT"I DON'T KNOW WHAT YOU PUSHED, BUT"
 270 PRINT"IT WASN'T AN 'A' OR 'B'. THIS TIME PUSH"
 280 PRINT"'A' FOR THE LESSON AND 'B' FOR POST-TEST."
 281 PRINT"TRY AGAIN."
 290 GOTO 230
 300 PRINT"PUSH THE RETURN KEY TO CONTINUE)"
 310 GOTO 380
 320 LINE INPUT A$:PRINT C$;
 330 RUN"ELEC$(?)",0:REM - POST TEST CODE
 340 PRINT"I'M SORRY "N$", YOUR SCORE ON THE"
 350 PRINT"PRE-TEST WILL NOT ALLOW YOU TO SKIP"
 360 PRINT"THE LESSON. PUSH THE RETURN AND"
 370 PRINT"WE WILL GET ON WITH IT.":GOTO380:LINE INPUT A$:PRINT C$;
 380 SCRIPT FOLLOWS
OK
```

(#5)

TO RECORD POST-TEST SCORE

```
LOAD"POSTTEST",0
OK
LIST

10 I=CN*100/(NUMBER OF QUESTIONS)
20 PRINT"N$", YOUR SCORE ON THE POST-TEST"
30 PRINT"WAS "CN" OR "I"%."
40 PRINT"TO RECORD THE SCORE OF YOUR POST-TEST,"
50 PRINT"TYPE 'R' AND PUSH THE RETURN KEY."
60 INPUT R$
70 IF R$="R" THEN 110
80 IF R$<> "R" THEN 90
90 PRINT"I DON'T KNOW WHAT YOU PUSHED, BUT"
100 PRINT"IT WASN'T AN 'R'. TRY AGAIN.":GOTO 60
110 LSET Z$ (LESSON NUMBER)=MKI$ (I)
120 PUT #1,N
130 IF I<80 THEN 150
140 IF I=>80 THEN 210
150 PRINT"IF YOU MISSED MORE THAN ONE OF THE"
160 PRINT"(?) QUESTIONS, THEN YOU MUST COME"
170 PRINT"BACK AT A LATER TIME AND TAKE THE "
180 PRINT"POST-TEST AGAIN. ASK YOUR INSTRUCTOR"
190 PRINT"FOR ADDITIONAL INFORMTAION."
200 END
210 PRINT"GOOD GOING. TRY TO REMEMBER "
220 PRINT"WHAT YOU LEARNED HERE AS IT MAY BE USEFUL"
230 PRINT"TO YOU IN THE FUTURE. GOOD LUCK."
240 END
OK
```

(#6)

TO PRINT(DISPLAY) CLASS/STUDENT PROGRESS REPORT FOR TEACHER/FACULTY

```
LOAD"CMI REPT",0
OK
LIST

10 DIM X$(24),Y$(24),Z$(24) ' PRODUCE A PROGRESS REPORT FOR C.A.I. COUR
E
20 DATA " 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8"," 9","10","11","12"
30 DATA "13","14","15","16","17","18","19","20","21","22","23","24"
40 FOR I = 1 TO 24
50 READ X$(I)
60 NEXT I
70 C$ = CHR$(27)+";"
80 PRINT C$;
90 PRINT "TYPE IN COURSE NAME AND NUMBER."
100 PRINT "FOR EXAMPLE:  IA 180";
110 INPUT N$
120 PRINT
130 PRINT "LOAD C.M.I. DISK ON DRIVE #1 AND"
140 PRINT "PUSH THE RETURN KEY WHEN READY."
160 LINE INPUT A$
190 OPEN "R",#1,N$,1
200 FIELD #1,9 AS S$,15 AS L$,7 AS F$,1 AS M$
210 FOR I = 1 TO 24
220 FIELD #1,(I+7)*4 AS A$,2 AS Y$(I),2 AS Z$(I)
230 NEXT I
240 K = 0
245 N = LOF(1)
250 FOR I = 1 TO N
260 PRINT "PUSH RETURN TO CONTINUE.";
270 LINE INPUT A$
280 PRINT C$;
290 GET #1,I
300 PRINT S$,F$,M$,L$
301 J=0
305 FOR K=1 TO 8
310 FOR L = 1 TO 3
315 J=J+1
320 PRINT USING"\       \";N$;
321 PRINT USING"\\";X$(J);
322 PRINT USING "####";CVI(Y$(J));
323 PRINT USING "####";CVI(Z$(J));
324 PRINT USING"\  \";"";
330 NEXT L
335 PRINT
340 NEXT K
360 NEXT I
370 END
OK
```

T3
F55X

D5

D6

R3
.1Ω
2W
(OPT.)

C6
4700 uf
25V

R5
1KΩ

R4
10Ω
½W

Q1 TIP34A

E1
LM340

D1
IN4743

+12

T1
F23U

SCR1

SCR-BRIDGE

(OPTIONAL)
R1
.1Ω
10W

C1 C2 C3 C4 R6
4700uF
16V
1KΩ

18,800μF

R11
.1Ω 2W

R10
62Ω ½W

R12
.1Ω 2W

R13
.33Ω ½W

D9
V352

3Ω

Q2 TIP36A

E2
LM309K

D2
IN5339

GND

+5

– heavy wire!

T2
F40X

D7

D8

(OPTIONAL)
R2
.1Ω
½W

C5
2200uF
25V

R7
1K

R8 1.5W
100Ω
2W

D3
IN5338

-5

125ma

R9 1.5W
20Ω
5W

D4
IN5348

-12

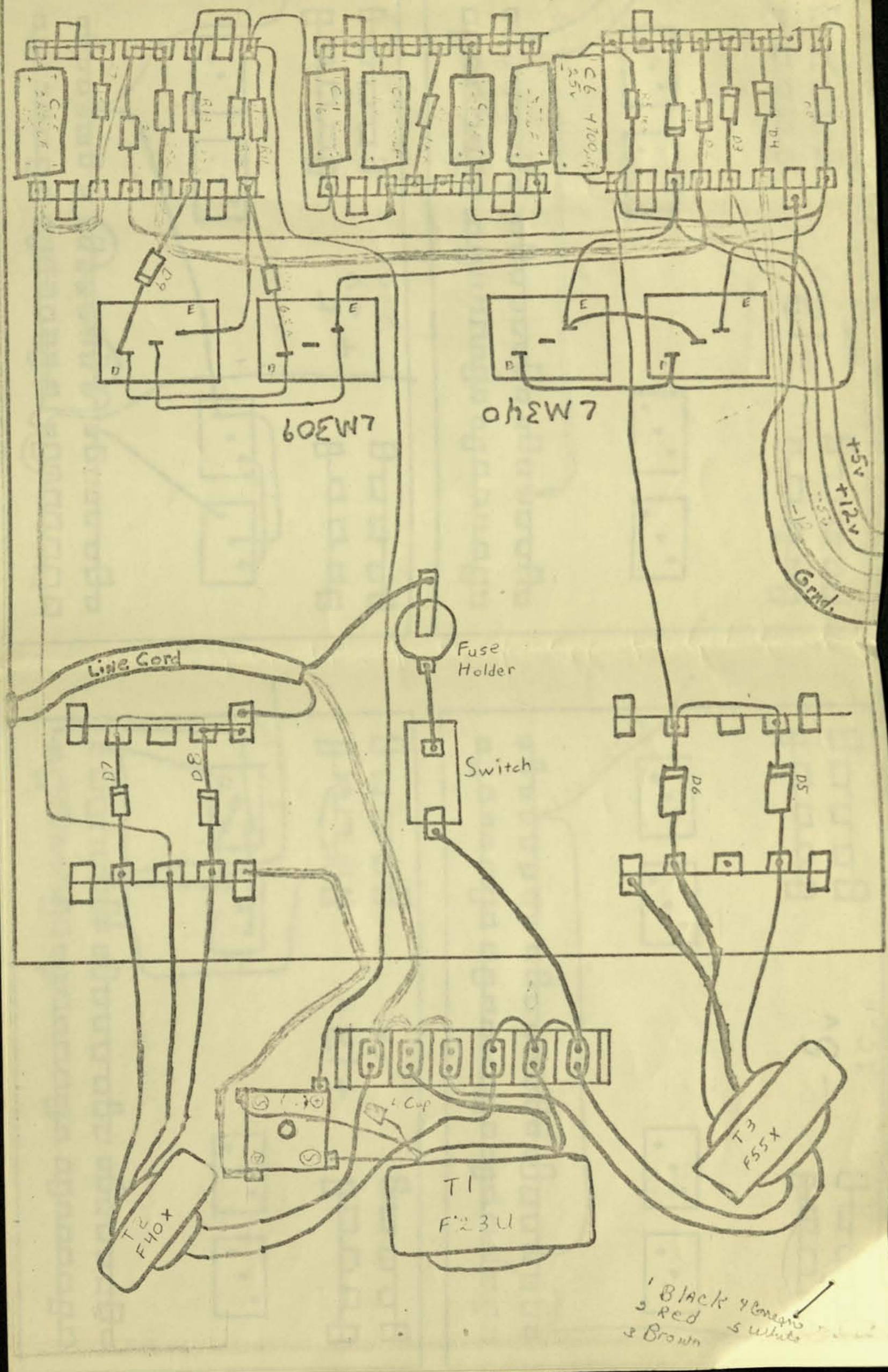
275ma

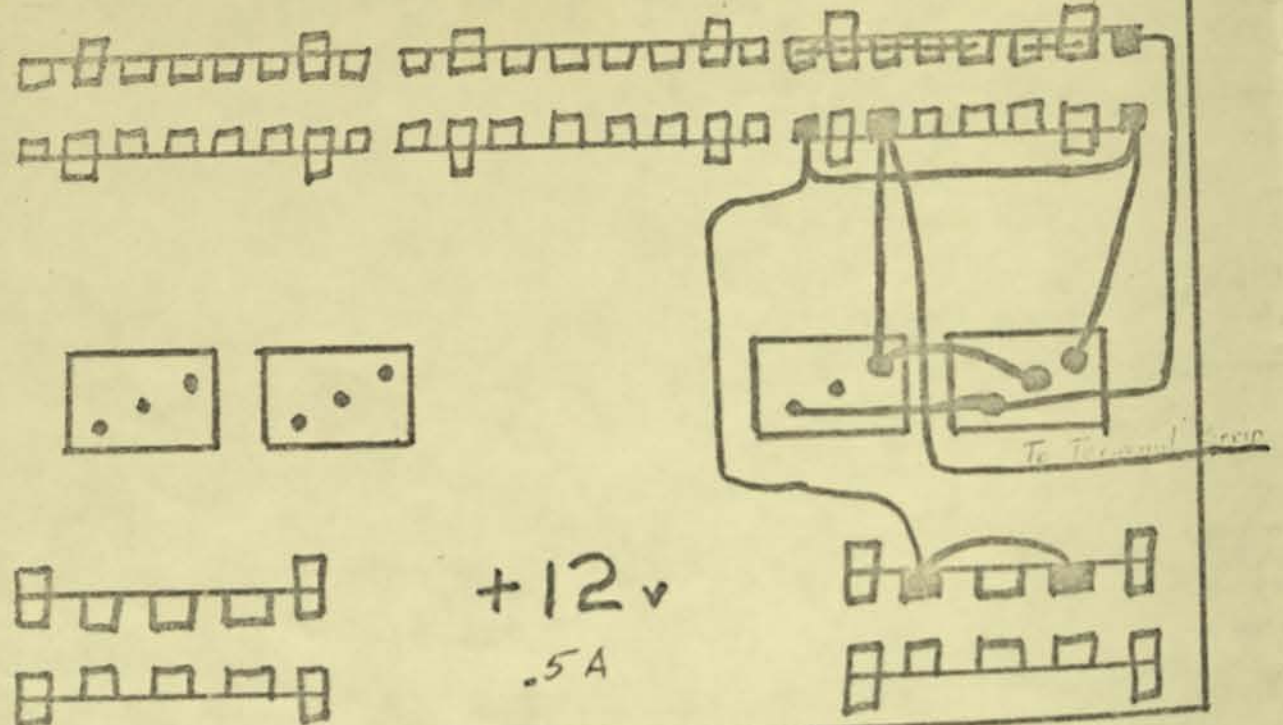
25mw

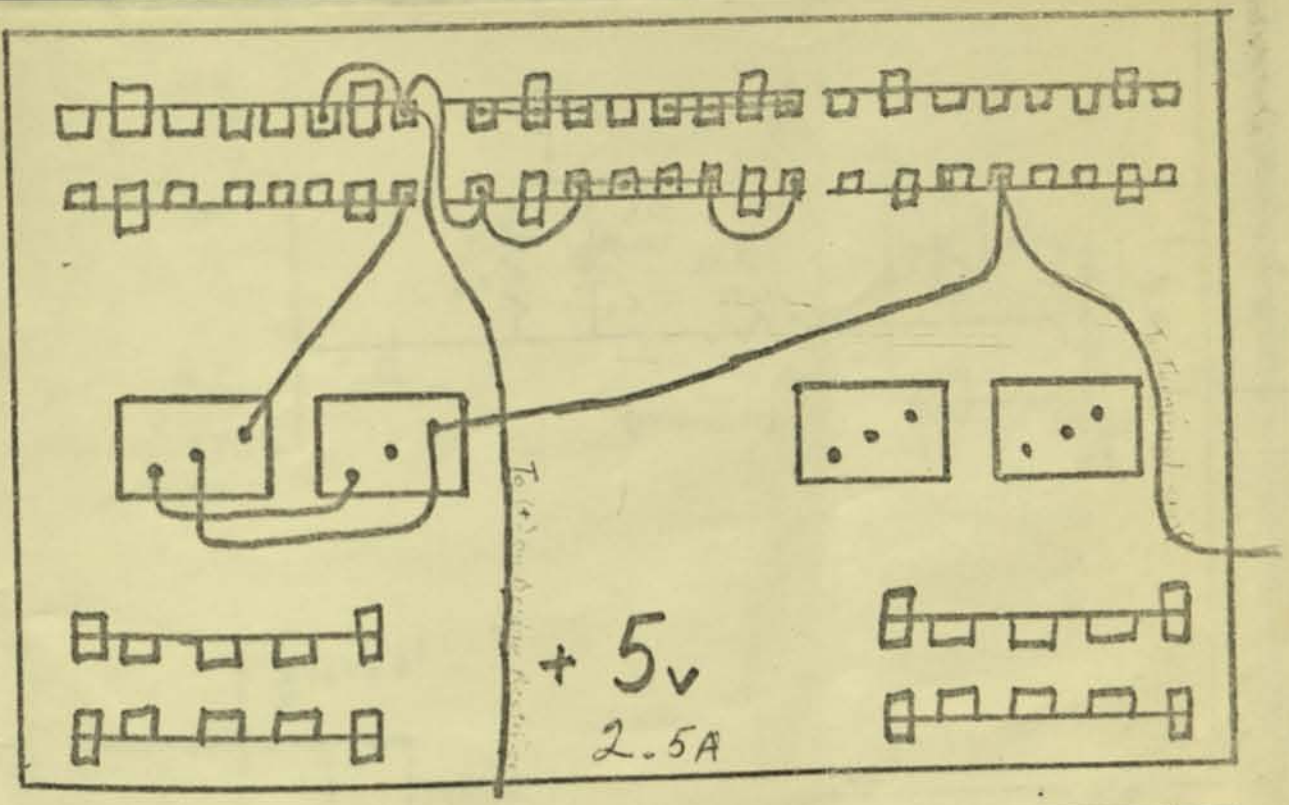
5v 1K ⇒ 5ma

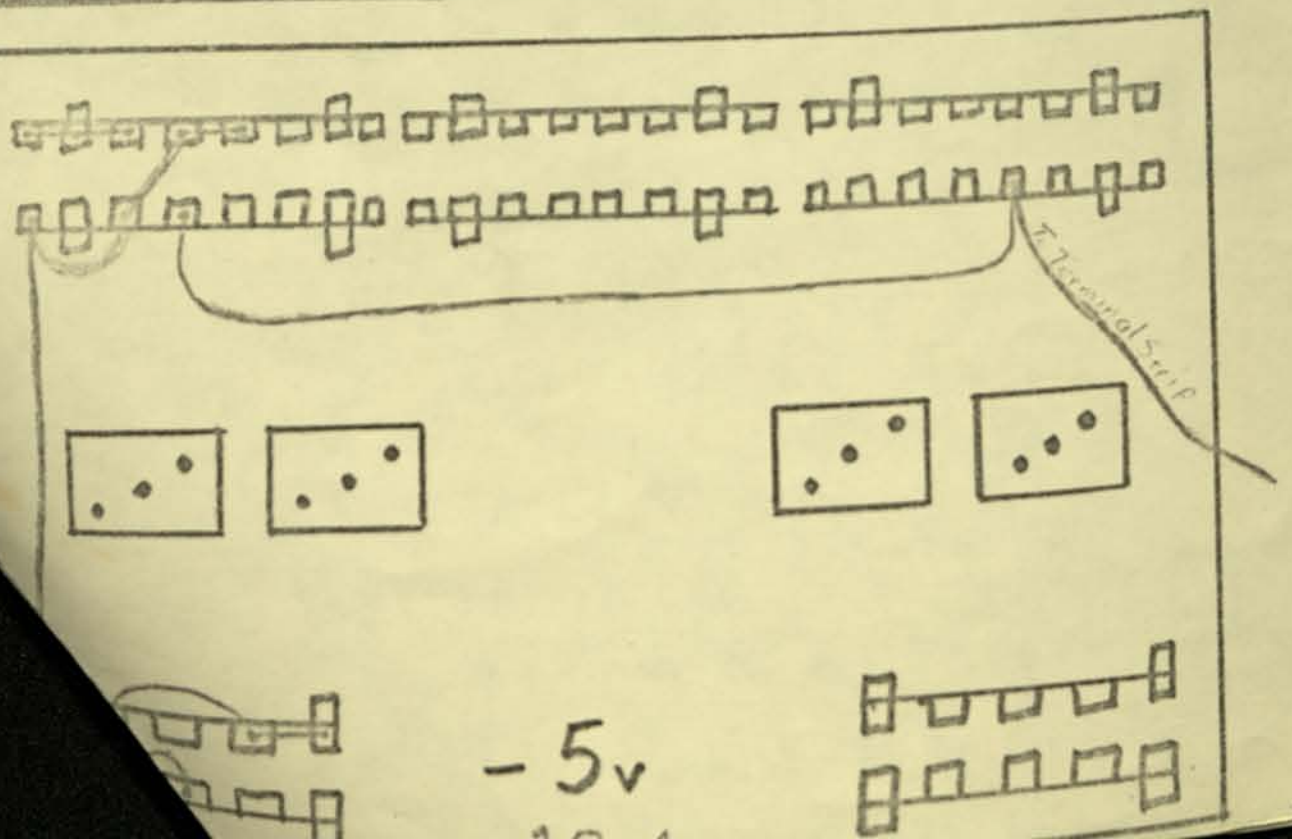
S1

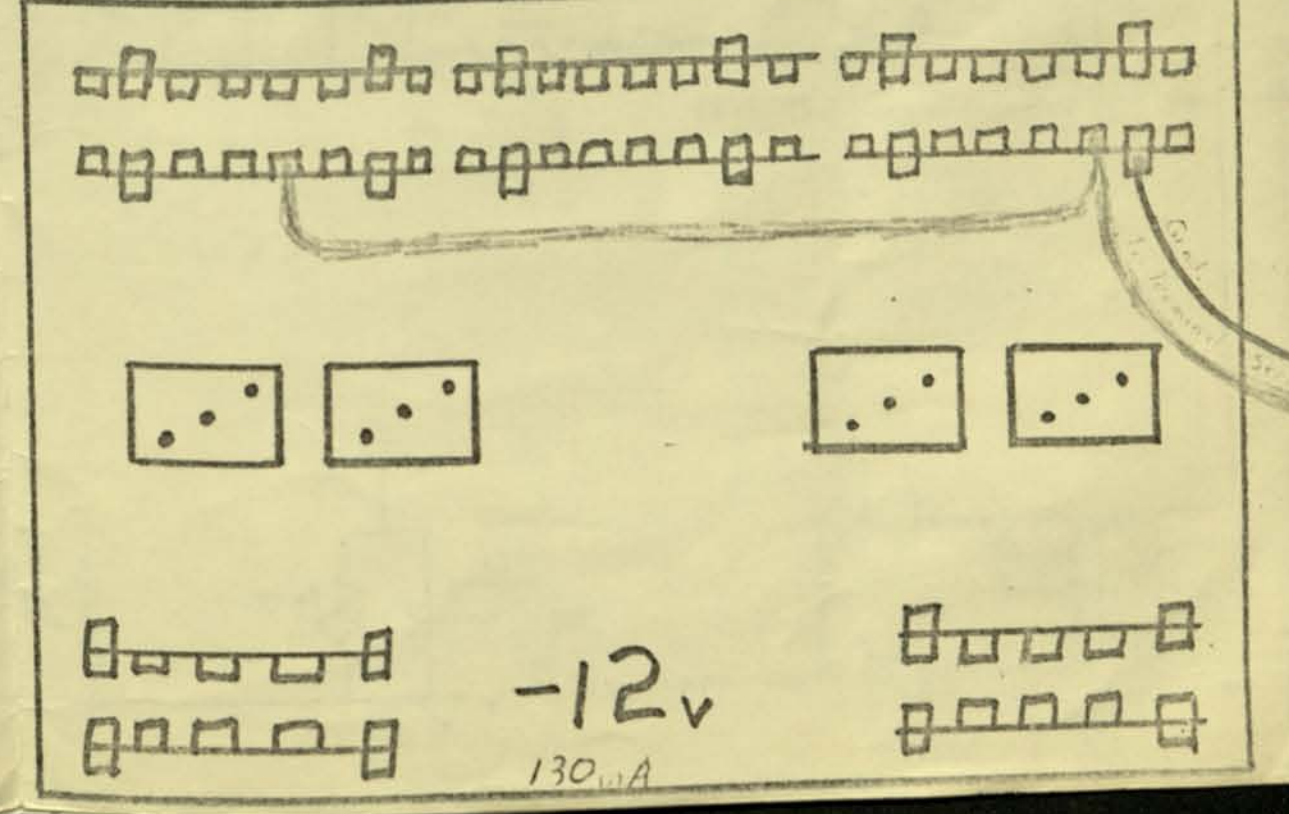
F1


LM309
LM340
+5v
+12v
Grnd.
Fuse
Holder
Line Cord
Switch
D7
D8
D5
D6
C4
T1
F230
T2
F40X
T3
F55X
1 Cap
1 BLACK
2 RED
3 Brown
4 Orange
5 White


+12v
.5A


+5v
2.5A


-5v


-12v
130mA

Phil –

When I first put the memory board back in, I could zero a memory location or two. Then when I checked that location, it wasn't zero! After running a short clear & checksum program (

```
LOOP → CLR 0,X
       EORA 0,X
       INX
       DECB
       BNE LOOP
```

) roughly the checksum was different each time, until it ran for a while & didn't change anything. Then in hot weather, all locations read as FF & I was not able to change them.

(note - the empty 12K were read as 00 - I changed the base address to check that they were OK - then cut the jumper)

I don't have a 'scope, but could it be: 1) low power?
2) refresh too slow?

I tried putting in Charlies CPU board, and I got the erratic data. Same as when I first got the board back last time - but it didn't write to the mem

please call me after 2-3 hours work. 4!45

John Ribble
876-8980

Thanks!

John!

The problem actualy was in the power supply!

The Boards patched on each power cable draw a different Load and due to the small wire cause a different drop on the ground. The way you had it patched caused a .7 volt difference between the Two cables. This caused ringing on the bus to Trip the Logic on the other boards. This caused the 200ns ring in the logic when the $\overline{CE}$ drives were Turned on.

Solution:

Patch it the way that I have patched it or better yet connect a ground from each board to the next then to the power bass screw Terminal ground. It would probably be good to use zip cord or something similar.

Good luck

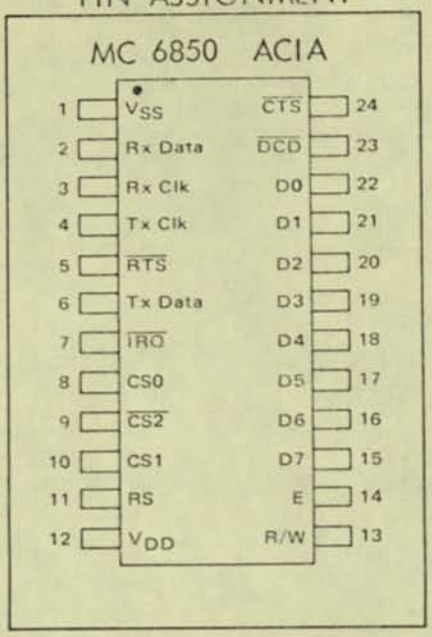
Phil Cameron

MC6850

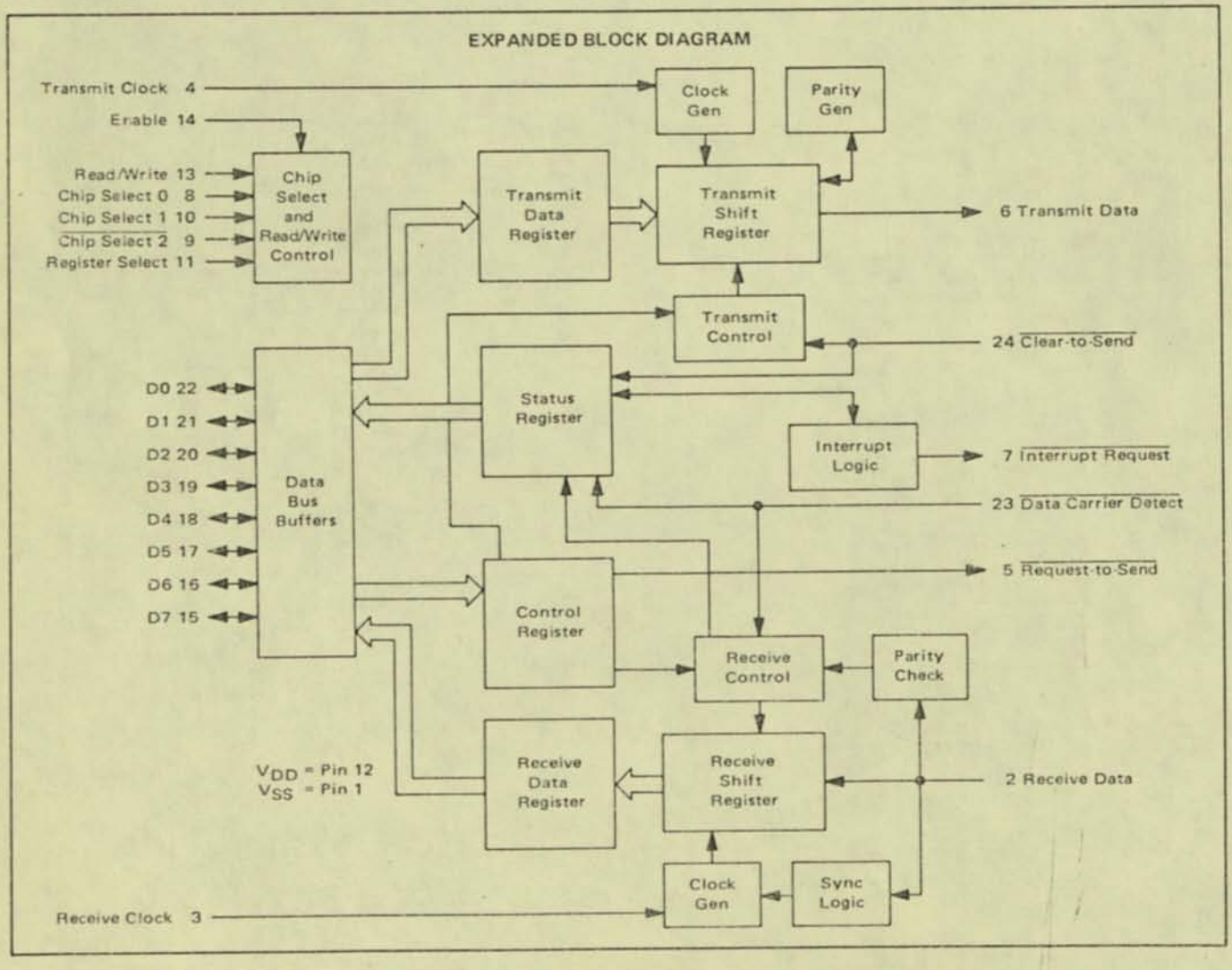
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

PIN ASSIGNMENT



EXPANDED BLOCK DIAGRAM



8602 DUAL TTL/MONOSTABLE MULTIVIBRATOR

general description

The TTL/Monostable DM9602/DM8602 dual retriggerable, resettable monostable multivibrator provides an output pulse whose duration and accuracy is a function of external timing components. The DM9602/DM8602 has excellent immunity to noise on the V_{CC} and ground lines. The DM9602/DM8602 uses TTL inputs and outputs for high speed and high fanout capability and is compatible with all members of the TTL family.

TTL Input Load and Drive Factors

INPUTS	LOAD	
	HIGH	LOW
3, 4, 5, 11, 12, 13	1 U.L.	1 U.L.

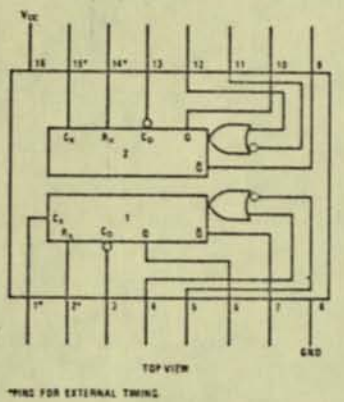
OUTPUTS	DRIVE FACTOR	
	HIGH	LOW
6, 7, 9, 10	16 U.L.	8 U.L.

1 Unit Load (U.L.) = 60μA HIGH/1.6 mA LOW

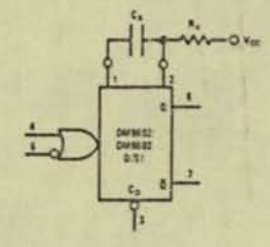
Triggering Truth Table

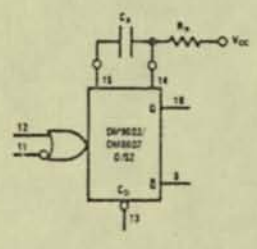
PIN NO'S. 5(11)	PIN NO'S. 4(12)	PIN NO'S. 3(13)	OPERATION
H→L	L	H	Trigger
H	L→H	H	Trigger
X	X	L	Reset

H = High Voltage Level ≥ V_{IH}
L = Low Voltage Level ≤ V_{IL}
X = Don't Care
H→L = High to Low Voltage Level Transition
L→H = Low to High Voltage Level Transition



*PINS FOR EXTERNAL TIMING





8833 QUAD TRI-STATE TRANSCEIVER

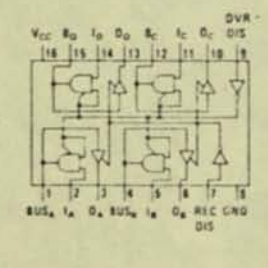
general description

This family of TRI-STATE® party line transceivers offer extreme versatility in bus organized data transmission systems. The data bus may be unterminated, or terminated DC or AC at one or both ends. Drivers in the third (high impedance) state load the data bus with a negligible leakage current. The receiver input current is low allowing at least 100 driver/receiver pairs to utilize a single bus. The bus loading is unchanged when V_{CC} = 0V. The receiver incorporates hysteresis to provide greater noise immunity. All devices utilize a high current TRI-STATE output driver. The DM7833/DM8833 and DM7835/DM8835 employ TRI-STATE outputs on the receiver also, while on the DM7834/DM8834 and DM7839/DM8839 the receiver outputs are standard active pull up T^2L.

The DM7833/DM8833 are non-inverting quad transceivers with a common driver disable control and a common receiver disable control.

The DM7839/DM8839 are non-inverting quad transceivers with a common two-input driver disable control.

The DM7834/DM8834 are inverting quad transceivers with a common two input driver disable control.

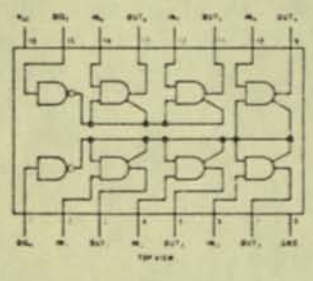


8097 TRI-STATE HEX BUFFER
8098 TRI-STATE HEX INVERTER

general description

Each of the devices described herein are used to convert standard TTL or DTL outputs to TRI-STATE outputs. The DM7095/DM8095 and the DM7097/DM8097 do so with no logic inversion; the DM7096/DM8096 and DM7098/DM8098 provide the logical opposite of the input signal. The DM7095/DM8095 and DM7096/DM8096 control all six devices from common inputs; the DM7097/DM8097 and DM7098/DM8098 control four devices from one input and two from another input.

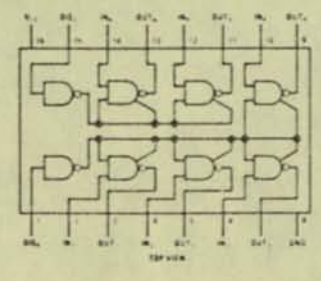
8097



DISABLE DIS_1	INPUT DIS_2	INPUT	OUTPUT
0	0	0	0
0	0	1	1
X	1	X	Hi-Z*
1	X	X	Hi-Z**

*Output 5-6 only
**Output 1-4 only
X = Irrelevant

8098



DISABLE DIS_1	INPUT DIS_2	INPUT	OUTPUT
0	0	0	1
0	0	1	0
X	1	X	Hi-Z*
1	X	X	Hi-Z**

74155 DUAL 2-LINE-TO-4-LINE DECODER/DEMULTIPLEXER

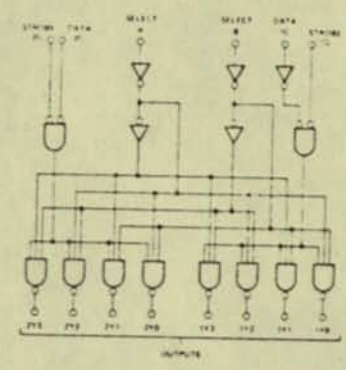
74156 DUAL 2-LINE-TO-4-LINE DECODER/DEMULTIPLEXER

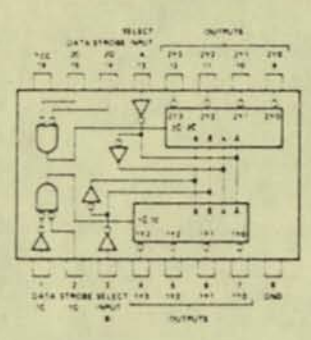
- Choice of Outputs:
  Totem Pole ('155, 'LS155)
  Open-Collector ('156)

DESCRIPTION

These monolithic transistor-transistor-logic (TTL) circuits feature dual 1-line to 4-line demultiplexers with individual strobes and common binary-address inputs in a single 16-pin package. When both sections are enabled by the strobes, the common binary-address inputs sequentially select and route associated input data to the appropriate output of each section. The individual strobes permit activating or inhibiting each of the 4-bit sections as desired.

Data applied to input 1C is inverted at its outputs and data applied at 2C is not inverted through its outputs. The inverter following the 1C data input permits use as a 3- to 8-line decoder or 1- to 8-line demultiplexer without external gating. See typical applications data and the truth tables for more details.





2-LINE TO 4-LINE DECODER OR 1-LINE TO 4-LINE DEMULTIPLEXER

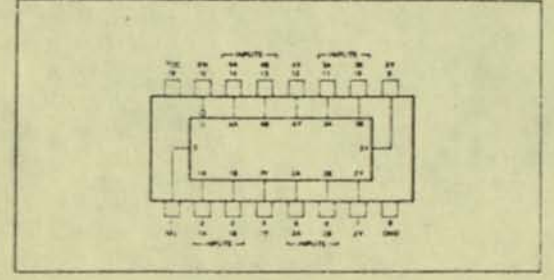
INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	1G	1C	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

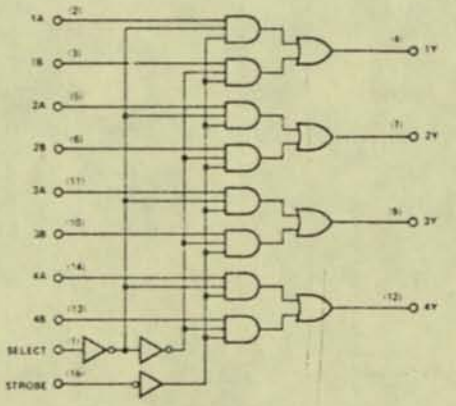
INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

74157 QUAD 2-LINE-TO-1-LINE DATA SELECTOR/MULTIPLEXER

DESCRIPTION

The S54157/N74157 and S54158/N74158 are identical with the exception of the S54158/N74158 being inverted. These devices are logical implementations of a four-pole two-position switch, with the position of the switch being set by the logic levels supplied to the one select input. Both assertion and negation outputs are provided. The enable input ($\overline{E}$) is active low. When it is not activated the negation output is high and the assertion output is low regardless of all other inputs. The devices provide the ability, in one package, to select four bits of either data or control from two sources. By proper manipulation of the inputs, it can generate four functions of two variables with one variable common. Thus any number of random logic elements used to generate unusual truth tables can be replaced. All outputs are low when disabled (enable high). Both inputs and outputs are buffered.





INPUTS			OUTPUT
STROBE	SELECT	A B	Y
H	X	X X	L
L	L	L X	L
L	L	H X	H
L	H	X L	L
L	H	X H	H

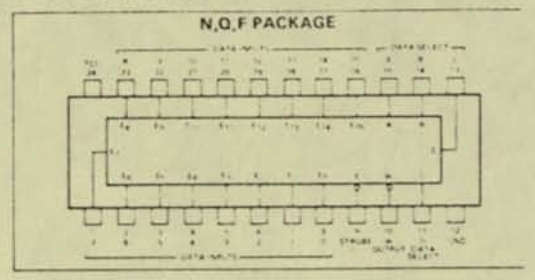
74150 16-LINE TO 1-LINE MULTIPLEXER

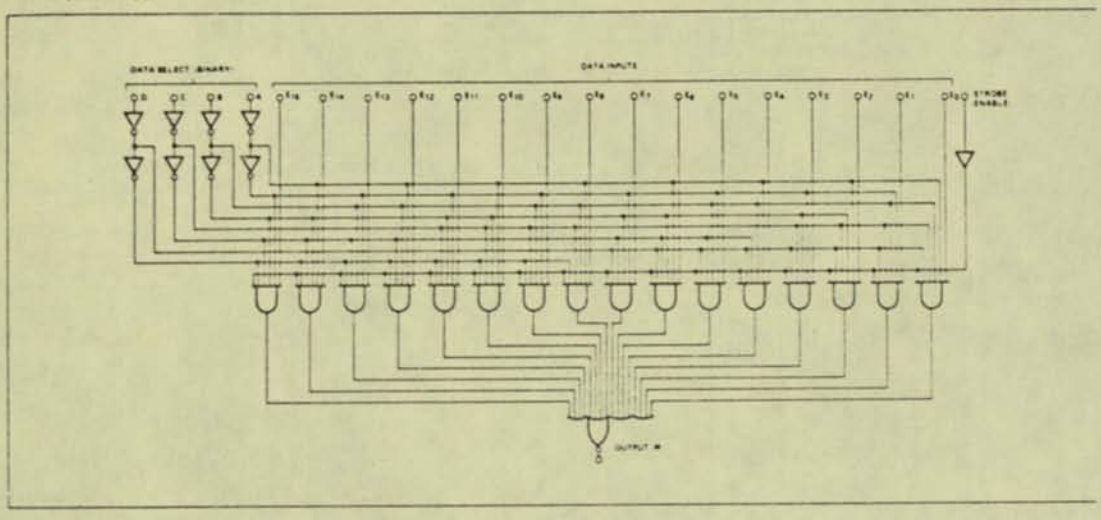
The 54/74150 is a one-of-sixteen data selector which performs parallel-to-serial data conversion. The unit incorporates an enable circuit for chip select. This allows multiplexing from N-lines to one-line.

The S54150/N74150 is provided with a strobe input which, when taken to a logical 0, enables the function of these multiplexers.

This data selector/multiplexer is fully compatible for use with other TTL or DTL circuit. Each input represents only one normalized Series 54/74 load, and full fan-out to 10 normalized Series 54/74 loads is available from each of the outputs in the logical 0 state. A fan-out to 20 normalized Series 54/74 loads is provided in the logical 1 state to facilitate connection of unused inputs to used inputs. Typical power dissipations are

S54150/N74150 – 200 milliwatts.





INPUTS																				OUTPUT
D	C	B	A	STROBE	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	E_{15}	W
X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0	0	0	1	0	X	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
0	0	0	1	0	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0	0	1	0	0	X	X	0	X	X	X	X	X	X	X	X	X	X	X	X	X	1
0	0	1	0	0	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0	0	1	1	0	X	X	X	0	X	X	X	X	X	X	X	X	X	X	X	X	1
0	0	1	1	0	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	0
0	1	0	0	0	X	X	X	X	0	X	X	X	X	X	X	X	X	X	X	X	1
0	1	0	0	0	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	X	0
0	1	0	1	0	X	X	X	X	X	0	X	X	X	X	X	X	X	X	X	X	1
0	1	0	1	0	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	X	0
0	1	1	0	0	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	X	1
0	1	1	0	0	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	X	0
0	1	1	1	0	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	1
0	1	1	1	0	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	X	0
1	0	0	0	0	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	1
1	0	0	0	0	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	0
1	0	0	1	0	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	1
1	0	0	1	0	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	0
1	0	1	0	0	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	1
1	0	1	0	0	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X	0
1	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	1
1	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	0
1	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	0	X	X	X	1
1	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	0
1	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	0	X	X	1
1	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	0
1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	X	1
1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	0
1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	1
1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0

LIST OF CONTENTS

555	TIMER
1702A	2048 BIT ELECTRICALLY PROGRAMMABLE READ ONLY MEMORY
2107A-8	FULLY DECODED RANDOM ACCESS 4096 BIT DYNAMIC MEMORY
2513	64 X 8 X 5 CHARACTER GENERATOR
7400	QUAD 2-INPUT NAND GATE
7402	QUAD 2-INPUT NOR GATE
7404	HEX INVERTER
7405	HEX INVERTER - OPEN COLLECTOR
7407	HEX BUFFERS/DRIVERS
7408	QUAD 2-INPUT AND GATE
7409	QUAD 2-INPUT AND GATE - OPEN COLLECTOR
7410	TRIPLE 3-INPUT NAND GATE
7411	3-INPUT POSITIVE AND GATE
7417	SEE 7407
7420	DUAL 4-INPUT NAND GATE
7427	TRIPLE 3-INPUT NOR GATE
7430	8-INPUT GATE
7442	BCD-TO-DECIMAL DECODER
7451	AND-OR-INVERT GATE DUAL 2-WIDE 2-INPUT
7474	DUAL D FLIP FLOP
7483	4-BIT BINARY ADDER & DUAL 1-BIT BINARY ADDER
7486	QUAD EXCLUSIVE-OR GATE
7490	DECADE COUNTER
7493	4-BIT BINARY COUNTER
7495	4-BIT RIGHT SHIFT/LEFT SHIFT REGISTER
74123	TTL/MONOSTABLE MULTIVIBRATOR
74150	16-LINE TO 1-LINE MULTIPLEXER
74155	DUAL 2-LINE-TO-4-LINE DECODER/DEMULTIPLEXER
74156	DUAL 2-LINE-TO-4-LINE DECODER/DEMULTIPLEXER
74157	QUAD 2-LINE-TO-1-LINE DATA SELECTOR/MULTIPLEXER
8602	DUAL TTL/MONOSTABLE MULTIVIBRATOR
8097	TRI-STATE HEX BUFFER
8098	TRI-STATE HEX INVERTER
8833	QUAD TRI-STATE TRANSCEIVER

NOTE - INTEGRATED CIRCUIT PART NUMBERS ARE PRECEEDED BY A TWO DIGIT ALPHA CODE DENOTING THE MANUFACTURER. A PARTIAL LIST IS OFFERED.

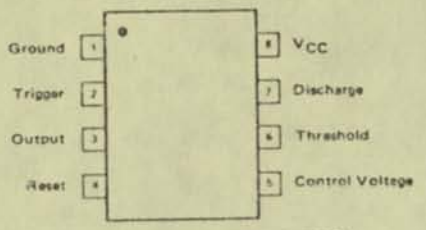
DM	NATIONAL SEMICONDUCTOR
SN	TEXAS INSTRUMENT
SW	STEWART-WARNER
MC	MOTOROLA

555 TIMER

DESCRIPTION

The NE/SE 555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays, or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA or drive TTL circuits.

V PACKAGE

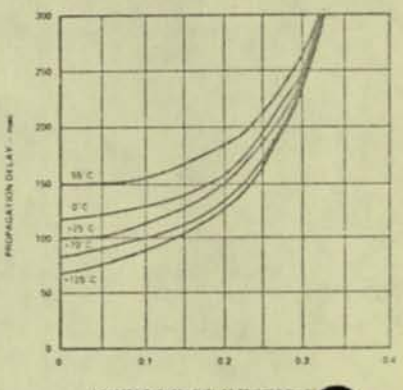


ORDER PART NOS. SE555V/NE555V

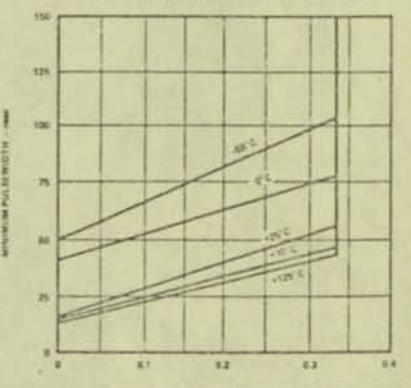
ELECTRICAL CHARACTERISTICS $T_A = 25°C$, $V_{CC} = +5V$ to $+15$ unless otherwise specified

PARAMETER	TEST CONDITIONS	SE 555			NE 555			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Supply Voltage		4.5		18	4.5		16	V
Supply Current	$V_{CC} = 5V$ $R_L = \infty$		3	5		3	6	mA
	$V_{CC} = 15V$ $R_L = \infty$		10	12		10	15	mA
	Low State, Note 1							
Timing Error(Monostable)	R_A, R_B = 1KΩ to 100KΩ							
Initial Accuracy	C = 0.1 μF Note 2		0.5	2		1		%
Drift with Temperature			30	100		50		ppm/°C
Drift with Supply Voltage			0.05	0.2		0.1		%/Volt
Threshold Voltage			2/3			2/3		X V_{CC}
Trigger Voltage	$V_{CC} = 15V$	4.8	5	5.2		5		V
Timing Error(Astable)	$V_{CC} = 5V$	1.45	1.67	1.9		1.67		V
Trigger Current			0.5			0.5		μA
Reset Voltage		0.4	0.7	1.0	0.4	0.7	1.0	V
Reset Current			0.1			0.1		mA
Threshold Current	Note 3		0.1	.25		0.1	.25	μA
Control Voltage Level	$V_{CC} = 15V$	9.6	10	10.4	9.0	10	11	V
	$V_{CC} = 5V$	2.9	3.33	3.8	2.6	3.33	4	V
Output Voltage (low)	$V_{CC} = 15V$							
	$I_{SINK} = 10mA$		0.1	0.15		0.1	.25	V
	$I_{SINK} = 50mA$		0.4	0.5		0.4	.75	V
	$I_{SINK} = 100mA$		2.0	2.2		2.0	2.5	V
	$I_{SINK} = 200mA$		2.5			2.5		
	$V_{CC} = 5V$							
	$I_{SINK} = 8mA$		0.1	0.25				V
	$I_{SINK} = 5mA$					.25	.35	
Output Voltage Drop (low)								
	$I_{SOURCE} = 200mA$		12.5			12.5		
	$V_{CC} = 15V$							
	$I_{SOURCE} = 100mA$							
	$V_{CC} = 15V$	13.0	13.3		12.75	13.3		V
	$V_{CC} = 5V$	3.0	3.3		2.75	3.3		V
Rise Time of Output			100			100		nsec
Fall Time of Output			100			100		nsec

PROPAGATION DELAY vs VOLTAGE LEVEL OF TRIGGER PULSE



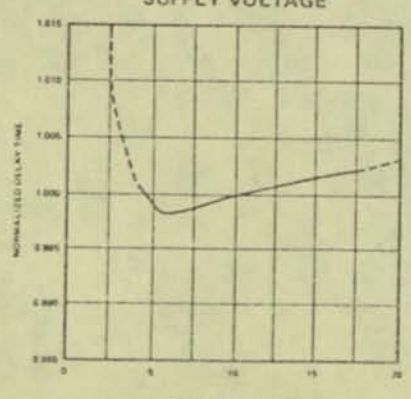
MINIMUM PULSE WIDTH REQUIRED FOR TRIGGERING



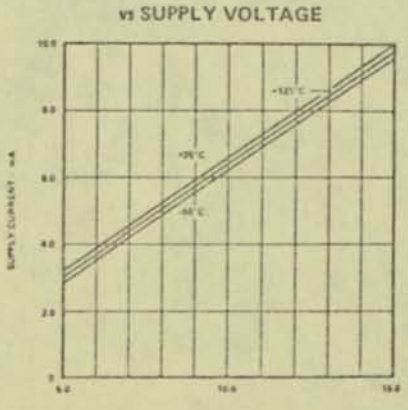
NOTES

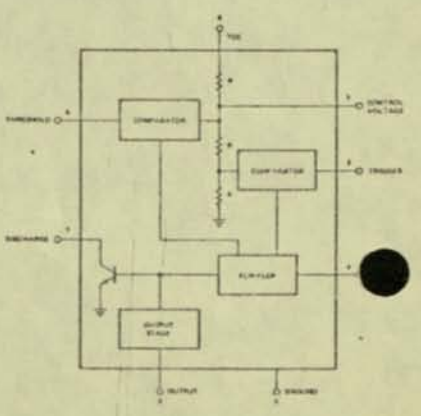
1. Supply Current when output high typically 1mA less.
2. Tested at $V_{CC} = 5V$ and $V_{CC} = 15V$
3. This will determine the maximum value of $R_A + R_B$ For 15V operation, the max total R = 20 megohm.

DELAY TIME vs SUPPLY VOLTAGE



SUPPLY CURRENT vs SUPPLY VOLTAGE





1702A–ERASABLE & ELECTRICALLY REPROGRAMMABLE

- Fast Programming--2 minutes for all 2048 bits
- All 2048 bits guaranteed* programmable--100% factory tested
- Fully Decoded, 256x8 organization
- Static MOS -- No Clocks Required
- Inputs and Outputs DTL and TTL compatible
- Three-state Output--OR-tie Capability
- Simple Memory Expansion--Chip select input lead

The 1602A and 1702A are 256 word by 8-bit electrically programmable ROMs ideally suited for uses where fast turn-around and pattern experimentation are important. The 1602A and 1702A undergo complete programming and functional testing on each bit position prior to shipment, thus insuring 100% programmability.

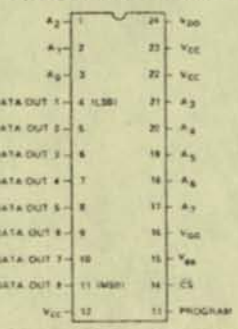
The 1602A and 1702A use identical chips. The 1702A is packaged in a 24 pin dual in-line package with a transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device. The 1602A is packaged in a 24 pin dual in-line package with a metal lid and is not erasable.

The circuitry of the 1602A/1702A is entirely static; no clocks are required.

A pin-for-pin metal mask programmed ROM, the Intel 1302, is ideal for large volume production runs of systems initially using the 1602A or 1702A.

The 1602A/1702A is fabricated with silicon gate technology. This low threshold technology allows the design and production of higher performance MOS circuits and provides a higher functional density on a monolithic chip than conventional MOS technologies.

PIN CONFIGURATION



*THIS PIN IS THE DATA INPUT LEAD DURING PROGRAMMING

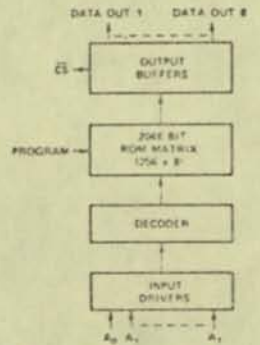
See page 3-8 for operational connection.

*Intel's liability shall be limited to replacing any unit which fails to program as desired.

PIN NAMES

A_0–A_7	Address Inputs
$\overline{CS}$	Chip Select Input
D_{OUT1}–D_{OUT8}	Data Outputs

BLOCK DIAGRAM



NOTE: In the read mode a logic 1 at the address inputs and data outputs is a high and logic 0 is a low.

U.S. Patent No. 3660819

A.C. Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±5%, V_{DD} = −9V ±5%, V_{GG} = −9V ±5% unless otherwise noted

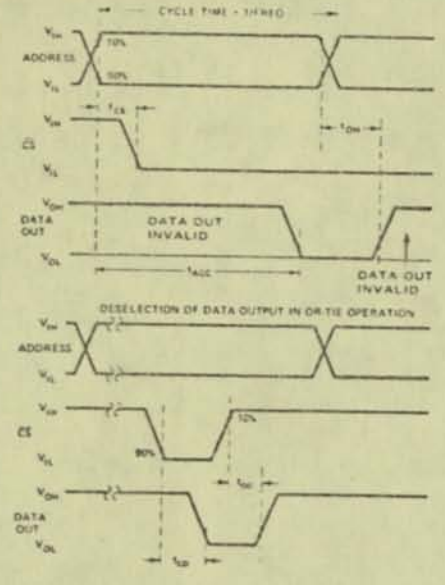
SYMBOL	TEST	MINIMUM	TYPICAL	MAXIMUM	UNIT
Freq.	Repetition Rate			1	MHz
t_{OH}	Previous read data valid			100	ns
t_{ACC}	Address to output delay		0.7	1	µs
t_{DVGG}	Clocked V_{GG} set up (Note 1)	1			µs
t_{CS}	Chip select delay			100	ns
t_{CO}	Output delay from $\overline{CS}$			900	ns
t_{OD}	Output deselect			300	ns
t_{OHC}	Data out hold in clocked V_{GG} mode (Note 1)			5	µs

Switching Characteristics

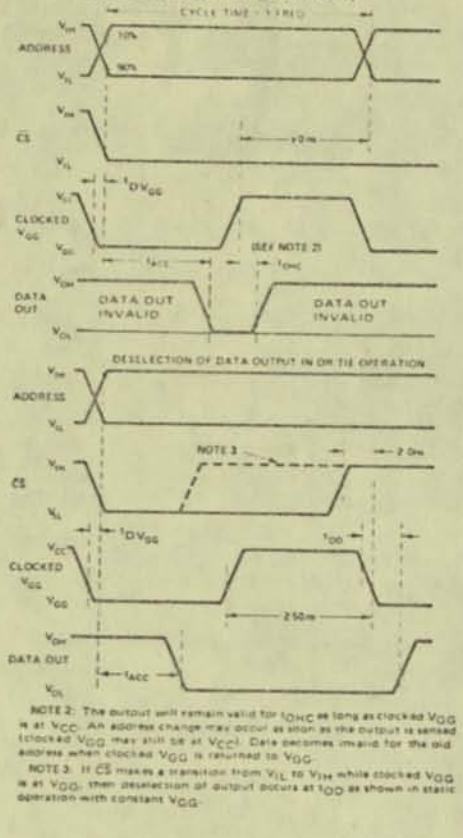
Conditions of Test:

Input pulse amplitudes: 0 to 4V; t_R, t_F ≤50 ns
Output load is 1 TTL gate; measurements made at output of TTL gate (t_{PD}≤15 ns), C_L = 15pF

A) Constant V_{GG} Operation



B) Power-Down Option (See Note 1)



NOTE 2: The output will remain valid for t_{OHC} as long as clocked V_{GG} is at V_{CC}. An address change may occur as soon as the output is sensed. Clocked V_{GG} may still be at V_{CC}. Data becomes invalid for the old address when clocked V_{GG} is returned to V_{GG}.

NOTE 3: If $\overline{CS}$ makes a transition from V_{IL} to V_{IH} while clocked V_{GG} is at V_{GG}, then deselection of output occurs at t_{OD} as shown in static operation with constant V_{GG}.

OPERATION OF THE 1702A IN PROGRAM MODE

Initially, all 2048 bits of the ROM are in the "0" state (output low). Information is introduced by selectively programming "1"s (output high) in the proper bit locations.

Word address selection is done by the same decoding circuitry used in the READ mode (see table on page 3-11 for logic levels). All 8 address bits must be in the binary complement state when pulsed V_{DD} and V_{GG} move to their negative levels. The addresses must be held in their binary complement state for a minimum of 25μsec after V_{DD} and V_{GG} have moved to their negative levels. The addresses must then make the transition to their true state a minimum of 10μsec before the program pulse is applied. The addresses should be programmed in the sequence 0 through 255 for a minimum of 32 times. The eight output terminals are used as data inputs to determine the information pattern in the eight bits of each word. A low data input level (-48V) will program a "1" and a high data input level (ground) will leave a "0" (see table on page 3-11). All eight bits of one word are programmed simultaneously by setting the desired bit information patterns on the data input terminals.

During the programming, V_{GG}, V_{DD} and the Program Pulse are pulsed signals.

1702A ERASING PROCEDURE

The 1702A may be erased by exposure to high intensity short-wave ultraviolet light at a wavelength of 2537Å. The recommended integrated dose (i.e., UV intensity x exposure time) is 6W-sec/cm². Examples of ultraviolet sources which can erase the 1702A in 10 to 20 minutes are the Model UVS-54 and Model S-52 short-wave ultraviolet lamps manufactured by Ultra-Violet Products, Inc. (5114 Walnut Grove Avenue, San Gabriel, California). The lamps should be used without short-wave filters, and the 1702A to be erased should be placed about one inch away from the lamp tubes.

PIN CONNECTIONS

The external lead connections to the 1602A/1702A differ, depending on whether the device is being programmed or used in read mode. (See following table.) In the programming mode, the data inputs 1-8 are pins 4-11 respectively.

PIN / MODE	12 (V_{CC})	13 (Program)	14 ($\overline{CS}$)	15 (V_{BB})	16 (V_{GG})	22 (V_{CC})	23 (V_{CC})
Read	V_{CC}	V_{CC}	GND	V_{CC}	V_{GG}	V_{CC}	V_{CC}
Programming	GND	Program Pulse	GND	V_{BB}	Pulsed V_{GG} (V_{ILAP})	GND	GND

2107A-8 FULLY DECODED RANDOM ACCESS 4096 BIT DYNAMIC MEMORY

TMS 4030 OR ZA-0248 ARE EQUIVALENT NUMBERS

- Low Cost Per Bit
- Low Standby Power
- Easy System Interface
- Only One High Voltage Input Signal—Chip Enable
- Low Level Address, Data, Write Enable, Chip Select Inputs
- Address Registers Incorporated on the Chip
- Simple Memory Expansion—Chip Select Input Lead
- Fully Decoded—On Chip Address Decode
- Output is Three State and TTL Compatible
- Ceramic and Plastic 22-Pin DIPs

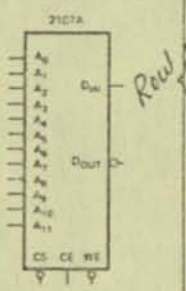
The Intel 2107A is a 4096 word by 1 bit dynamic n-channel MOS RAM. It was designed for memory applications where very low cost and large bit storage are important design objectives. The 2107A uses dynamic circuitry which reduces the operation and standby power dissipation.

Reading information from the memory is non-destructive. Refreshing is accomplished by performing one read cycle on each of the 64 row addresses. Each row address must be refreshed every two milliseconds. The memory is refreshed whether Chip Select is a logic one or a logic zero.

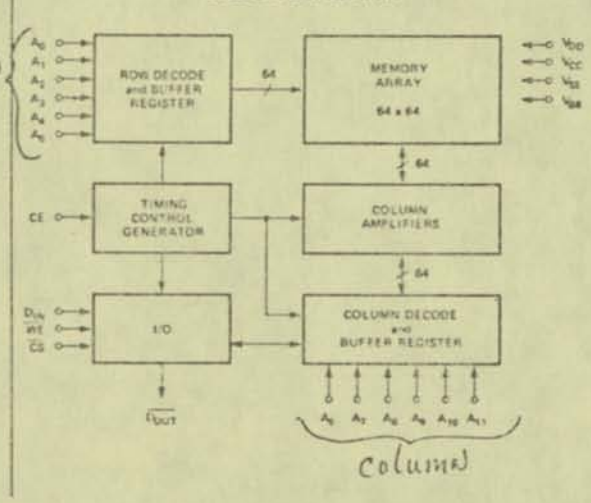
The 2107A is fabricated with n-channel silicon gate technology. This technology allows the design and production of high performance, easy to use MOS circuits and provides a higher functional density on a monolithic chip than other MOS technologies.

PIN CONFIGURATION

LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

D_{IN}	DATA INPUT	CE	CHIP ENABLE
A_0-A_{11}	ADDRESS INPUTS*	$\overline{D_{OUT}}$	DATA OUTPUT
$\overline{WE}$	WRITE ENABLE	V_{CC}	POWER (+5V)
$\overline{CS}$	CHIP SELECT	NC	NOT CONNECTED

*Refresh Addresses A_0-A_5

A.C. Characteristics

T_A = 0°C to 70°C, V_{DD} = 12V ± 5%, V_{CC} = 5V ± 5%, V_{BB} = -5V ± 5%,

READ, WRITE, AND READ MODIFY/WRITE CYCLE V_{SS} = 0V, unless otherwise noted.

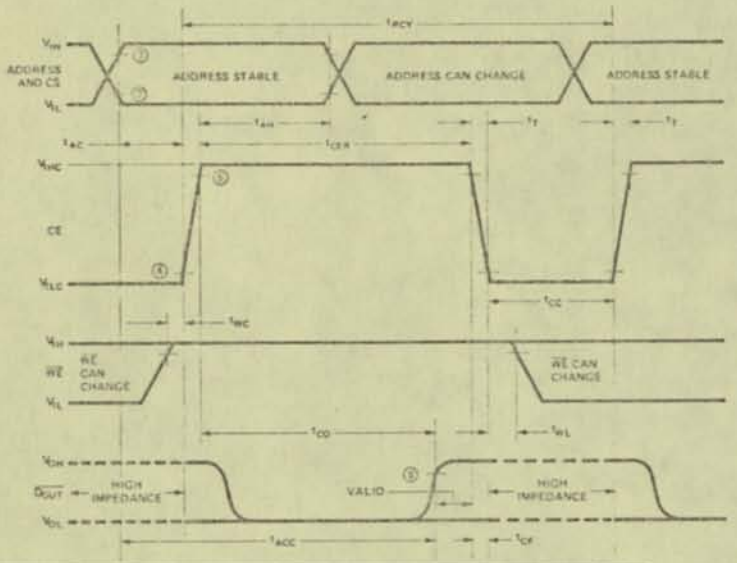
Symbol	Parameter	Min.	Max.	Unit	Conditions
t_{REF}	Time Between Refresh		2	ms	
t_{AC}	Address to CE Set Up Time	0		ns	
t_{AH}	Address Hold Time	100		ns	
t_{CC}	CE Off Time	180		ns	
t_T	CE Transition Time		50	ns	
t_{CF}	CE Off to Output High Impedance State	0		ns	

2107A OR EQUIVALENT NO.S

READ CYCLE

Symbol	Parameter	Min.	Max.	Unit	Conditions
t_{RCY}	Read Cycle Time	500		ns	t_T = 20ns
t_{CER}	CE On Time During Read	280	3000	ns	
t_{CO}	CE Output Delay		280	ns	C_{load} = 50pF, Load = One TTL Gate,
t_{ACC}	Address to Output Access		300	ns	Ref = 2.0V for High, 0.8V for Low.
t_{WL}	CE to $\overline{WE}$ Low	0		ns	$t_{ACC} = t_{AC} + t_{CO} + 1t_T$
t_{WC}	$\overline{WE}$ to $\overline{CE}$ on	0		ns	

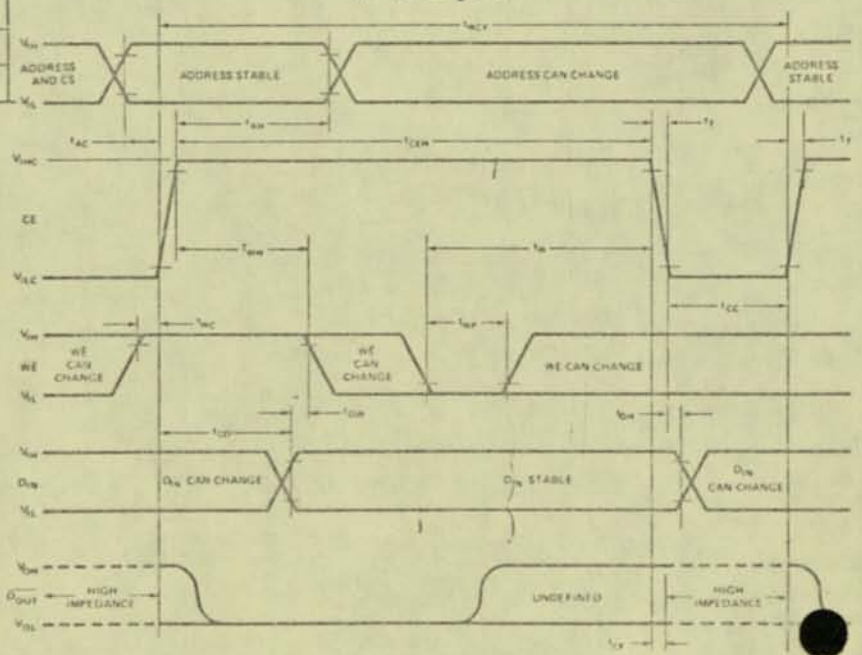
Read and Refresh Cycle



WRITE CYCLE

Symbol	Parameter	Min.	Max.	Unit	Conditions
t_{WCY}	Write Cycle Time	700		ns	t_T = 20ns
t_{CEW}	CE Width During Write	480	3000	ns	
t_W	$\overline{WE}$ to CE Off	340		ns	
t_{CW}	CE to $\overline{WE}$ High	300		ns	
t_{DW}	D_{IN} to $\overline{WE}$ Set Up	0		ns	
t_{CD} [1]	CE to D_{IN} Set Up		50	ns	
t_{DH}	D_{IN} Hold Time	0		ns	
t_{WP}	$\overline{WE}$ Pulse Width	150		ns	
t_{WW}	$\overline{WE}$ Wait	0		ns	
t_{WC}	$\overline{WE}$ to CE On	0		ns	

Write Cycle

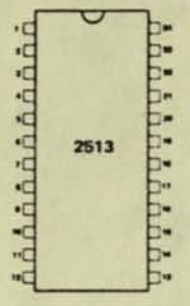


NOTES:
1. For Refresh cycle row and column addresses must be stable before t_{AC} and remain stable for entire t_{AH} period.
2. V_{SS} + 1.5V is the reference level for measuring timing of the addresses, $\overline{CS}$, $\overline{WE}$, and D_{IN}.
3. V_{SS} + 3.0V is the reference level for measuring timing of the addresses, $\overline{CS}$, $\overline{WE}$, and D_{IN}.
4. V_{SS} + 2.0V is the reference level for measuring timing of CE.
5. V_{DD} -2V is the reference level for measuring timing of CE.
6. V_{SS} + 2.0V is the reference level for measuring the timing of $\overline{D_{OUT}}$.

2513 64 X 8 X 5 CHARACTER GENERATOR

DESCRIPTION

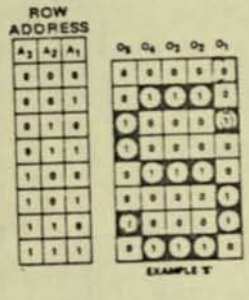
The Signetics 2513 is a high speed 2560-bit Static ROM organized as 64x8x5. A standard 7x5 dot matrix fits well in the 2513. The product uses +5V, -5V and -12V power supplies, TTL level interface signals and Tri-State Outputs for direct, low cost interfacing with TTL, DTL, CMOS and 2500 Series MOS.

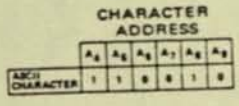


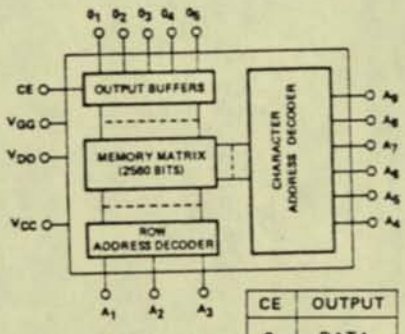
FEATURES

- 450 ns TYPICAL ACCESS TIME
- STATIC OPERATION
- TTL/DTL COMPATIBLE INPUTS
- +5, -5, -12V POWER SUPPLIES
- TRI-STATE OUTPUT CONTROLLED BY CHIP ENABLE FOR BUSSING CAPABILITY
- 2513/CM2140 ASCII FONT STANDARD (7 X 5)
- 24-PIN DIP
- P-MOS SILICON GATE TECHNOLOGY

1.	V_{GG}	24.	V_{CC}
2.	NC	23.	NC
3.	NC	22.	Address 9
4.	Out 1	21.	Address 8
5.	Out 2	20.	Address 7
6.	Out 3	19.	Address 6
7.	Out 4	18.	Address 5
8.	Out 5	17.	Address 4
9.	NC	16.	Address 3
10.	Ground	15.	Address 2
11.	Chip Enable	14.	Address 1
12.	V_{DD}	13.	NC

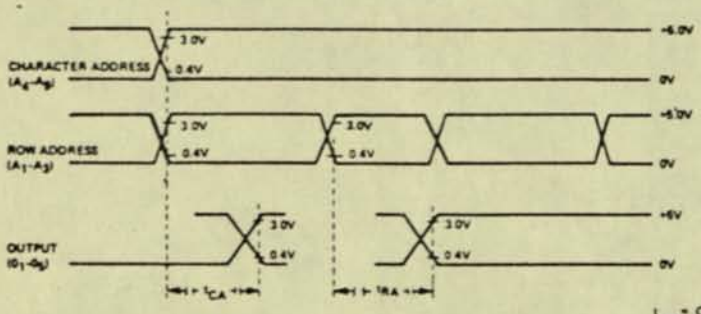




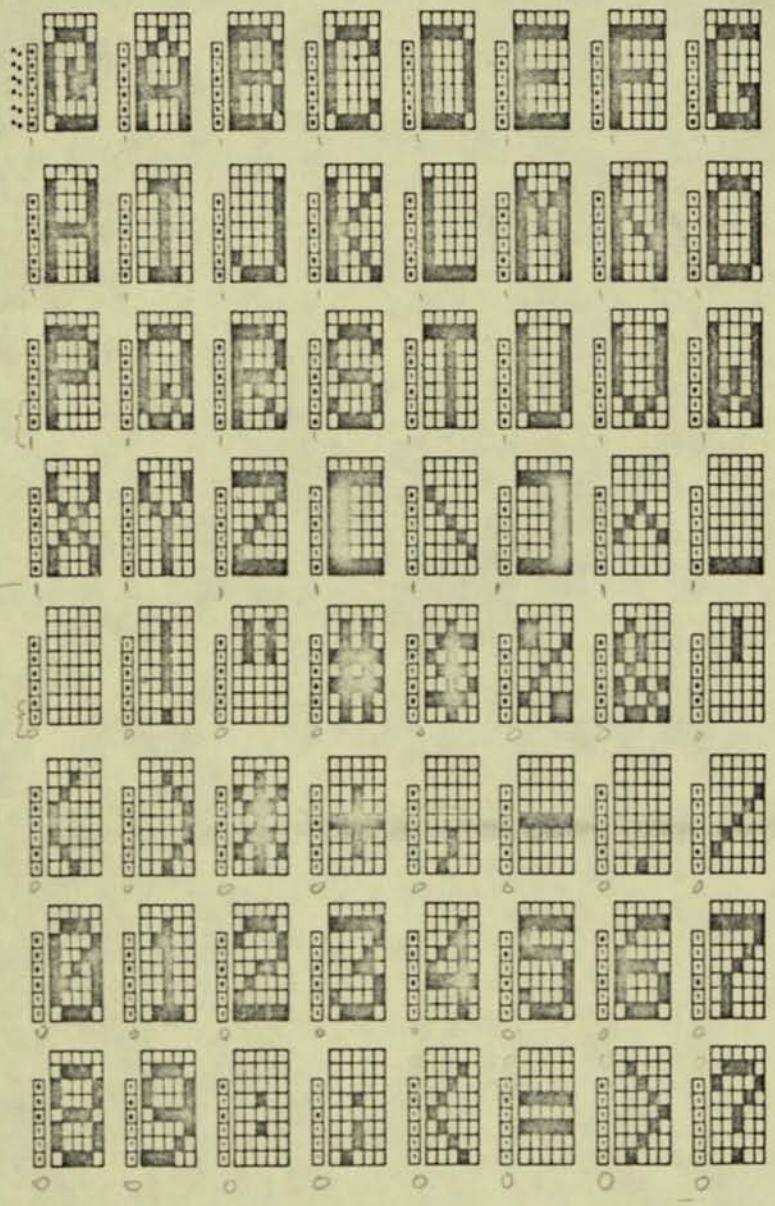


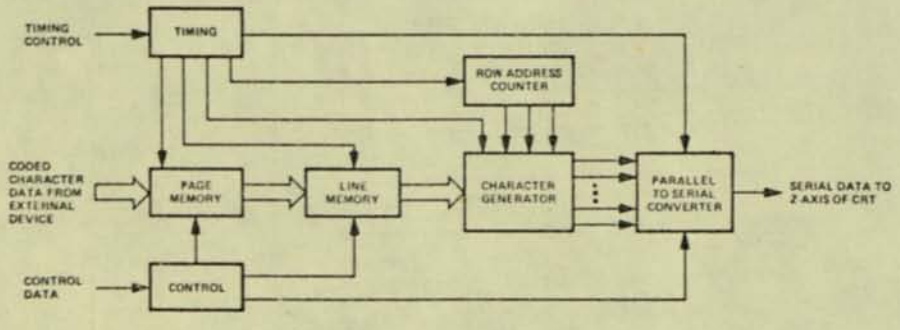
CE	OUTPUT
0	DATA
1	OPEN

TIMING DIAGRAM



t_{CA} = Character Access Time
t_{RA} = Row Access Time





APPLICATION INFORMATION

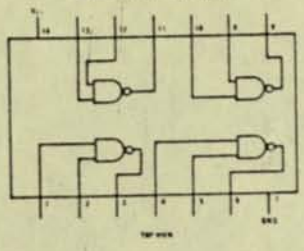
CHARACTER GENERATOR: The 2513 IS DESIGNED TO PROVIDE THE INFORMATION NEEDED TO CONVERT THE CHARACTER CODES INTO A DOT MATRIX FOR DISPLAY.

PAGE MEMORY: THIS MEMORY CONTAINS CHARACTER CODES. TYPICALLY, IT CONTAINS THE SAME NUMBER OF CHARACTER CODES AS THE NUMBER OF CHARACTERS ON A FULL SCREEN.

LINE MEMORY: THIS MEMORY CONTAINS THE CHARACTER CODES FOR ONE LINE OF THE CRT DISPLAY.

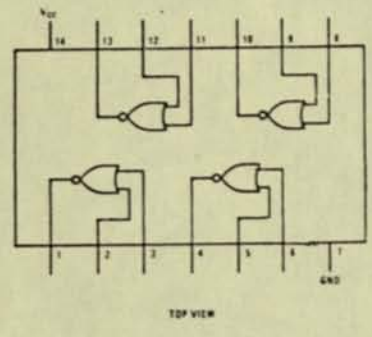
7400 QUADRUPLE 2-INPUT NAND GATE

Dual-In-Line Package



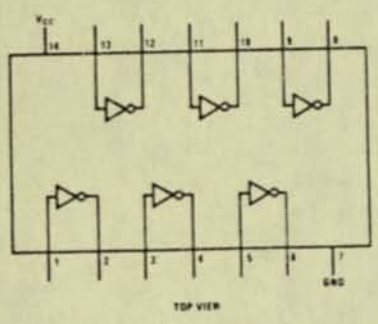
7402 QUAD 2-INPUT NOR GATE

Dual-In-Line Package



7404 HEX INVERTER

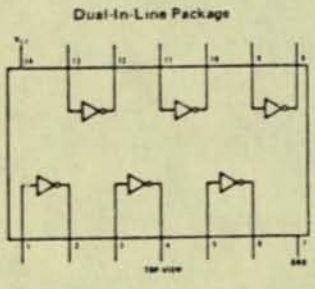
Dual-In-Line Package



7405 HEX INVERTER

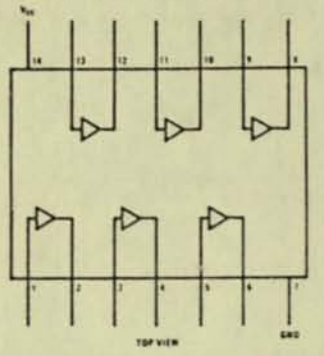
OPEN COLLECTOR

Dual-In-Line Package



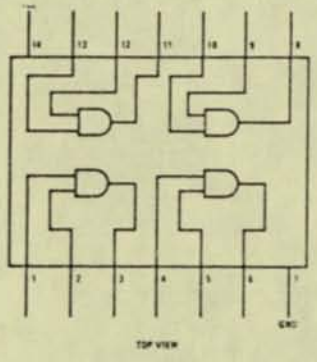
7407 HEX BUFFERS/DRIVERS
7417 HEX BUFFERS/DRIVERS

Dual-In-Line and Flat Package



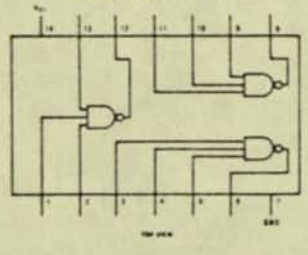
7408 QUAD 2-INPUT AND GATE
7409 QUAD 2-INPUT AND GATE

OPEN COLLECTOR

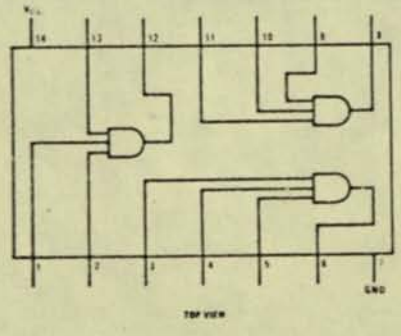


7410 TRIPLE 3-INPUT NAND GATE

Dual-In-Line Package

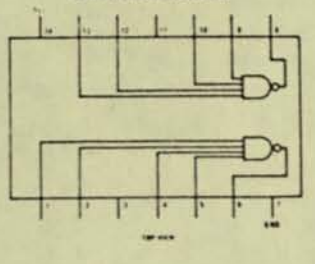


7411 3-INPUT POSITIVE AND GATE

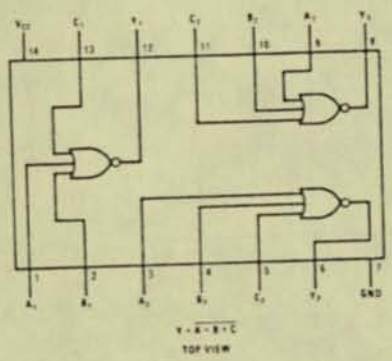


7420 DUAL 4-INPUT NAND GATE

Dual-In-Line Package

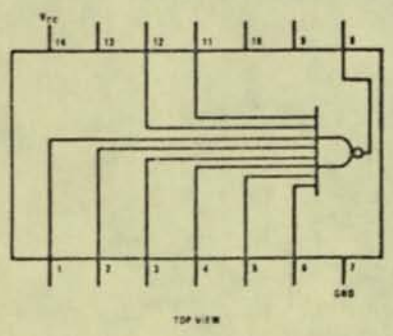


7427 TRIPLE 3-INPUT NOR GATE



7430 8-INPUT GATE

Dual-In-Line Package

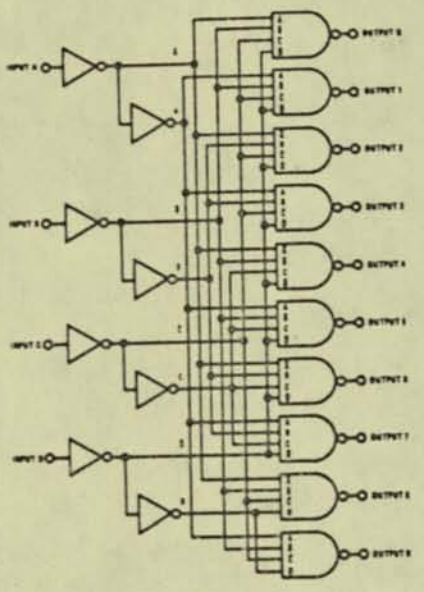


7442 BCD-TO-DECIMAL DECODER

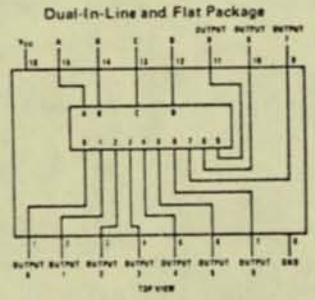
The DM5442/DM7442 utilizes Series 54/74 compatible circuitry to decode a four-bit BCD number to one-of-ten decimal outputs. These ten decimal outputs are capable of driving 10 standard TTL loads each.

The decoding logic is designed such that when binary numbers between 10 and 15 are applied to the inputs, no outputs are enabled.

logic and connection diagrams



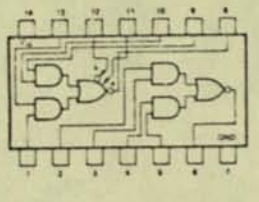
Dual-In-Line and Flat Package



logic table

INPUTS				OUTPUTS							
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

7451 AND-OR-INVERT GATE DUAL 2-WIDE 2-INPUT



4. Make no external connection to X and $\overline{X}$ pins of the S5451 and N7451.
5. A total of four expander gates can be connected to the expander inputs.

7474 DUAL D FLIP FLOP

DESCRIPTION

The S5474/N7474 is a monolithic, dual, D-type, edge-triggered flip-flop featuring direct clear and preset inputs and complementary Q and $\overline{Q}$ outputs. Input information is transferred to the Q output on the positive edge of the clock pulse.

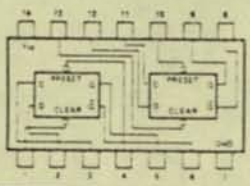
Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive going pulse. After the clock input threshold voltage has been passed, the data input (D) is locked out.

TRUTH TABLE

D_n	Q_{n+1}	$\overline{Q}_{n+1}$
1	1	0
0	0	1

Preset	Clear	Q
1	1	Q
1	0	0
0	1	1
0	0	†

† Both outputs in 1 state
n is time prior to clock
n+1 is time following clock



POSITIVE LOGIC – Low input to preset sets Q to logical 1. Low input to clear sets Q to logical 0. Preset and clear are independent of clock.

7483 4-BIT BINARY FULL ADDER AND DUAL SINGLE-BIT BINARY FULL ADDER

DESCRIPTION

The 54/7483 is a 4-Bit Binary Full Adder for adding two four bit binary numbers. A Carry Look Ahead circuit is included to provide minimum carry propagation delays.

Propagation delays of carry-in to carry-out is typically 12 nsec.

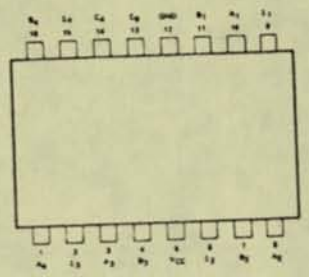
TRUTH TABLE

INPUT				OUTPUT					
				WHEN $C_0 = 0$			WHEN $C_0 = 1$		
				WHEN $C_2 = 0$			WHEN $C_2 = 1$		
A_1 / A_3	B_1 / B_3	A_2 / A_4	B_2 / B_4	Σ_1 / Σ_3	Σ_2 / Σ_4	C_2 / C_4	Σ_1 / Σ_3	Σ_2 / Σ_4	C_2 / C_4
0	0	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	1	0
0	1	0	0	1	0	0	0	1	0
1	1	0	0	0	1	0	1	1	0
0	0	1	0	0	1	0	1	1	0
1	0	1	0	1	1	0	0	0	1
0	1	1	0	1	1	0	0	0	1
1	1	1	0	0	0	1	1	0	1
0	0	0	1	0	1	0	1	1	0
1	0	0	1	1	1	0	0	0	1
0	1	0	1	1	1	0	0	0	1
1	1	0	1	0	0	1	1	0	1
0	0	1	1	0	0	1	1	0	1
1	0	1	1	1	0	1	0	1	1
0	1	1	1	1	0	1	0	1	1
1	1	1	1	0	1	1	1	1	1

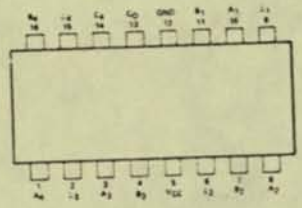
NOTES:
Input conditions at A_1, A_2, B_1, B_2, and C_0 are used to determine outputs Σ_1 and Σ_2, and the value of the internal carry C_2. The values at C_2, A_3, B_3, A_4, and B_4, are then used to determine outputs Σ_3, Σ_4, and C_4.

PIN CONFIGURATIONS

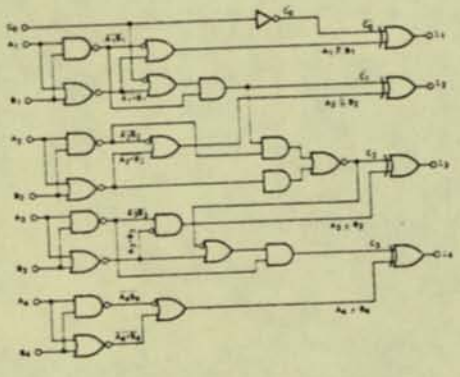
W PACKAGE

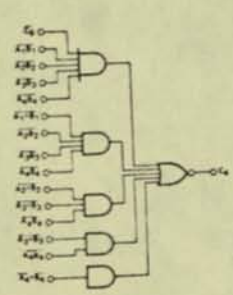


B,F PACKAGE

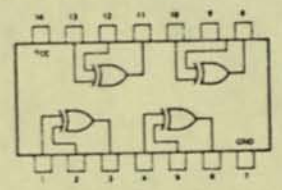


LOGIC DIAGRAM





7486 QUAD EXCLUSIVE-OR GATE



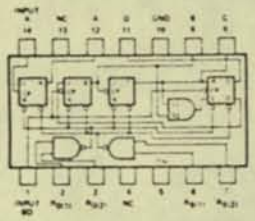
7490 DECADE COUNTER

DESCRIPTION

The S5490/N7490 is a high-speed, monolithic decade counter consisting of four dual-rank, master-slave flip-flops internally interconnected to provide a divide-by-two counter and a divide-by-five counter. Gated direct reset lines are provided to inhibit count inputs and return all outputs to a logical "0" or to a binary coded decimal (BCD) count of 9. As the output from flip-flop A is not internally connected to the succeeding stages, the count may be separated in three independent count modes:

1. When used as a binary coded decimal decade counter, the BD input must be externally connected to the A output. The A input receives the incoming count, and a count sequence is obtained in accordance with the BCD count sequence truth table shown above. In addition to a conventional "0" reset, inputs are provided to reset a BCD 9 count for nine's complement decimal applications.
2. If a symmetrical divide-by-ten count is desired for frequency synthesizers or other applications requiring division of a binary count by a power of ten, the D output must be externally connected to the A input. The input count is then applied at the BD input and a divide-by-ten square wave is obtained at output A.
3. For operation as a divide-by-two counter and divide-by-five counter, no external interconnections are required. Flip-flop A is used as a binary element for the divide-by-two function. The BD input is used to obtain binary divide-by-five operation at the B, C, and D outputs. In this mode, the two counters operate independently; however, all four flip-flops are reset simultaneously.

The 5490/7490 is completely compatible with Series 54 and Series 74 logic familes. Average power dissipation is 160mW.



LOGIC TRUTH TABLES

BCD COUNT SEQUENCE (See Note 1)

COUNT	OUTPUT D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

RESET/COUNT (See Note 2)

RESET INPUTS $R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	OUTPUT D C B A
1	1	0	X	0 0 0 0
1	1	X	0	0 0 0 0
X	X	1	1	1 0 0 1
X	0	X	0	COUNT
0	X	0	X	COUNT
0	X	X	0	COUNT
X	0	0	X	COUNT

NOTES:
1. Output A connected to input BD for BCD count.
2. X indicates that either a logical 1 of a logical 0 may be present.
3. Fanout from output A to input BD and to 10 additional Series 54/74 loads is permitted

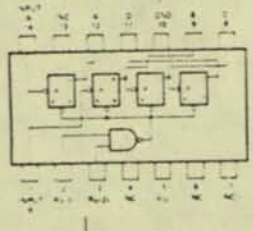
7493 4-BIT BINARY COUNTER

DESCRIPTION

The S5493/N7493 is a high-speed, monolithic 4-bit binary counter consisting of four master-slave flip-flops which are internally interconnected to provide a divide-by-two counter and a divide-by-eight counter. A gated direct reset line is provided which inhibits the count inputs and simultaneously returns the four flip-flop outputs to a logical 0. As the output from flip-flop A is not internally connected to the succeeding flip-flops the counter may be operated in two independent modes:

1. When used as a 4-bit ripple-through counter output A must be externally connected to input B. The input count pulses are applied to input A. Simultaneous divisions of 2, 4, 8, and 16 are performed at the A, B, C, and D outputs as shown in the truth table.
2. When used as a 3-bit ripple-through counter, the input count pulses are applied to input B. Simultaneous frequency divisions of 2, 4, and 8 are available at the B, C, and D outputs. Independent use of flip-flop A is available if the reset function coincides with reset of the 3-bit ripple-through counter.

The S5493/N7493 is completely compatible with Series 54 and Series 74 logic families. Average power dissipation is 32mW per flip-flop (128mW total).



TRUTH TABLE (See Notes 1 and 2)

LOGIC

COUNT	OUTPUT D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0

COUNT	OUTPUT D	C	B	A
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

NOTES
1. Output A connected to input B.
2. To reset all outputs to logical 0, both $R_{0(1)}$ and $R_{0(2)}$ inputs must be at logical 1.

7495 4-BIT RIGHT SHIFT/LEFT SHIFT REGISTER

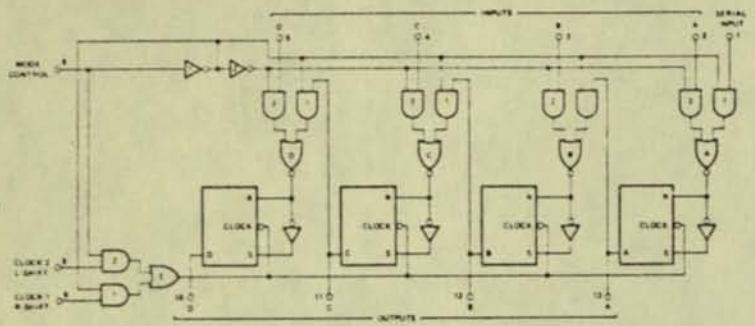
DESCRIPTION

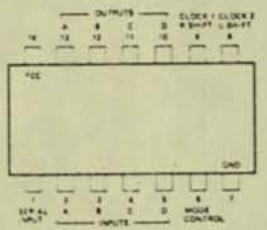
The 54/7495 is a monolithic universal 4-Bit Shift Register designed with standard TTL techniques. The circuit layout consists of 4 R-S master-slave flip-flops, 4 AND-OR-INVERT gates, and 6 inverters configured to form a versatile register which will perform right-shift, left-shift, or parallel-in, parallel-out operations depending on the logical input level to the mode control.

Right-shift operations are performed when a logical 0 level is applied to the mode control. Serial data is entered at the serial input D_S and shifted one position right on each clock 1 pulse. In this mode, clock 2 and parallel inputs D_A thru D_D are inhibited.

Parallel-in, parallel-out operations are performed when a logical 1 level is applied to the mode control. Parallel data is entered at parallel inputs D_A thru D_D and is transferred to the data outputs A_Q thru D_Q on each clock 2 pulse. In this mode, shift left operations may be implemented by externally tying the output of each flip-flop to the parallel input of the previous flip-flop (D_Q to D_C and etc.), with serial data entry at input D_D.

Information must be present at the R-S inputs prior to clocking and transfer of data occurs on the falling edge of the clock pulse.





74123 TTL/MONOSTABLE MULTIVIBRATOR

DESCRIPTION

These monostables are designed to provide the system designer with complete flexibility in controlling the pulse width, either to lengthen the pulse by retriggering, or to shorten by clearing. N74122 has an internal timing resistor which allows the circuit to be operated with only an external capacitor, if so desired. Applications requiring more precise pulse widths and not requiring the clear feature can best be satisfied with N74121.

The output pulse is primarily a function of the external capacitor and resistor. For $C_{ext} > 1000pF$, the output pulse width (t_w) is defined as:

$$t_w = 0.32\ R_T C_{ext} \left(1 - \frac{0.7}{R_T}\right)$$

where

R_T is in kΩ (either internal or external timing resistor)
C_{ext} is in pF
t_w is in ns

For pulse widths when $C_{ext} \leq 1000pF$, see Figure B.

These circuits are fully compatible with most TTL or DTL families. Inputs are diode-clamped to minimize reflections due to transmission-line effects, which simplifies design. Typical power dissipation per one shot is 115 milliwatts, typical average propagation delay time to the Q output is 21 nanoseconds. The N74122 and N74123 are characterized for operation from 0°C to 70°C.

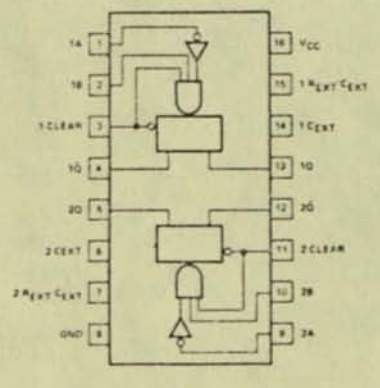
(See Note A)

N74122

INPUTS				OUTPUTS	
A_1	A_2	B_1	B_2	Q	$\overline{Q}$
H	H	X	X	L	H
X	X	L	X	L	H
X	X	X	L	L	H
L	X	H	H	L	H
L	X	↑	H	⊓	⊔
L	X	H	↑	⊓	⊔
X	L	H	H	L	H
X	L	↑	H	⊓	⊔
X	L	H	↑	⊓	⊔
H	↓	H	H	⊓	⊔
↓	↓	H	H	⊓	⊔
	H	H	H	⊓	⊔

S54123,N74123

INPUTS		OUTPUTS	
A	B	Q	$\overline{Q}$
H	X	L	H
X	L	L	H
L	↑	⊓	⊔
↓	H	⊓	⊔



NOTES

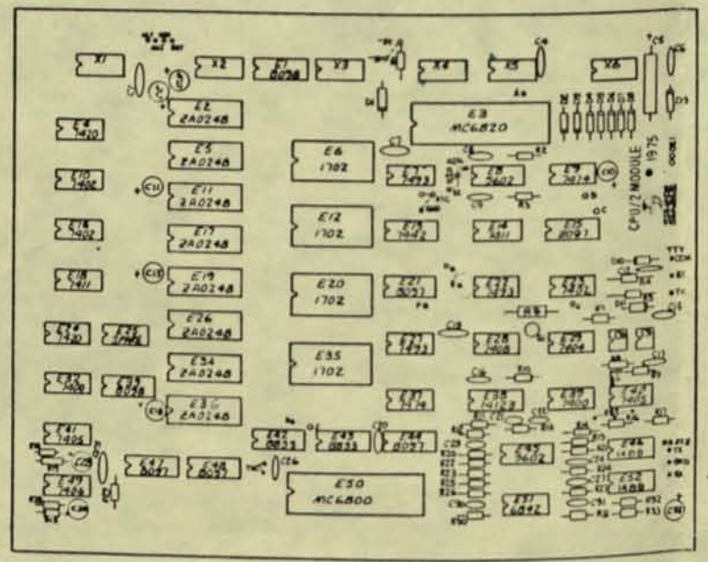
A. H = high level (steady state), L = low level (steady state), ↑ = transition from low to high level, ↓ = transition from high to low level, ⊓ = one high level pulse, ⊔ = one low level pulse, X = irrelevant (any input, including transitions).

B. NC = No internal connection.

C. To use the internal timing resistor of N74122 (10kΩ nominal), connect R_{int} to V_{CC}.

D. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).

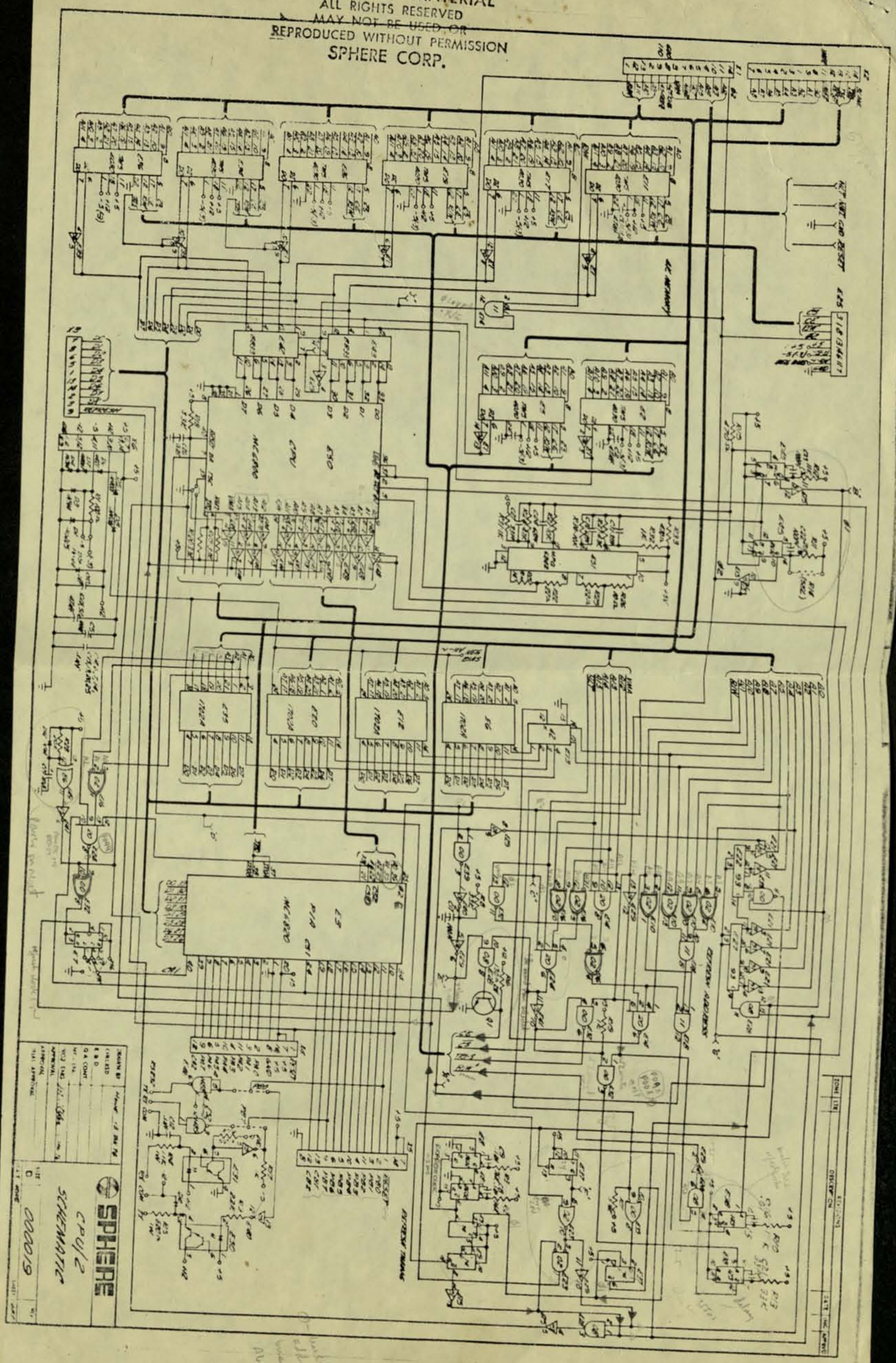
REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	BY



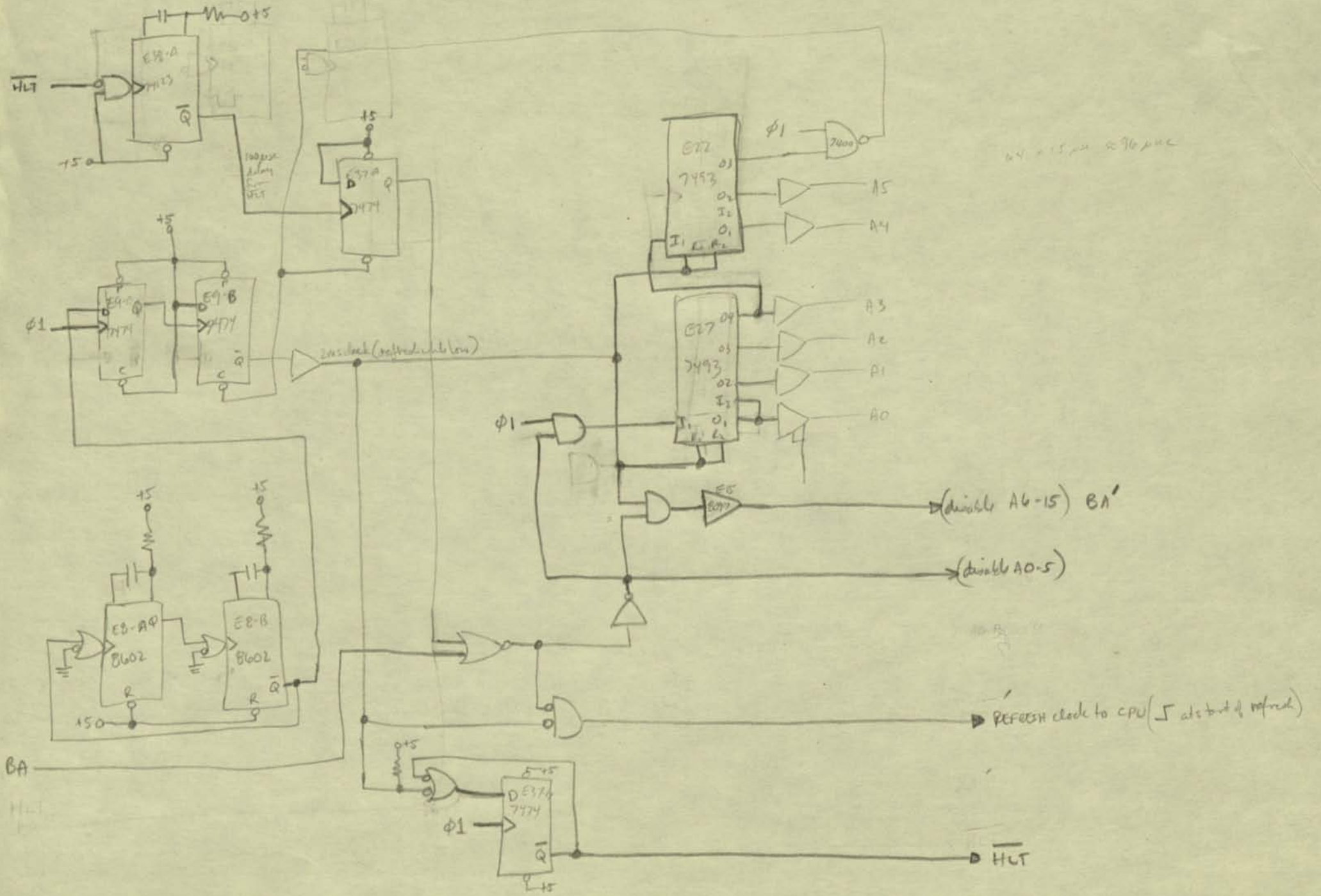
REF	REF	QTY		SCHEMATIC	DOOOOI9		
1	1	60		PRINTED CIRCUIT BOARD	2000020		
1	1	59		TRANSISTOR	2N2369A	Q1	
2	2	58		CAPACITOR CERAMIC 68pf 1KV	DD-680	C29,30	CRL
3	3	57		CAPACITOR CERAMIC 330pf 1KV	DD-101	C1,17,31	CRL
3	3	56		CAPACITOR CERAMIC .001uf 1KV	DDM-102	C15,22,27	CRL
2	2	55		CAPACITOR CERAMIC .01uf 50V	DDM-103	C16,26	CRL
1	1	54		CAPACITOR TANTALUM .00uf 2.5V		C5	MCM 68
9	9	53		CAPACITOR .47uf 16 VDC	ECE-A1VF6	C3,4,6,7,8,9,10,12,24	CRL
12	11	52		CAPACITOR CERAMIC .1uf 50V	DK-104	C14,18,19,21,35	1,2,0,25 CRL
2	2	51		RESISTOR 470K Ω ¼W	RC07GF474	R29,37	
2	2	50		RESISTOR 10 Ω ¼W	RC07GF 20	R23,26	
2	2	49		RESISTOR 22 Ω ¼W	RC07GF220	R27,35	
3	3	48		RESISTOR 10K Ω ¼W	RC07GF103	R2,28,34	
4	3	47		RESISTOR 1K Ω ¼W	RC07GF 22	R17,31,32	
1	–	46		RESISTOR 510 Ω ¼W	RC07GF511	R6	
2	2	45		RESISTOR 470 Ω ¼W	RC07GF471	R15,33	
1	1	44		RESISTOR	RC07GF5GL	R4	
1	1	43		RESISTOR [illegible] Ω ¼W	RC07GF	R3	
1	1	42		RESISTOR 1K Ω ¼W 1%	RN65H-1021	R12	
1	1	41		RESISTOR 2.15K Ω ¼W 1%	RN65H2151F	R11	
1	1	40		RESISTOR [illegible] Ω ¼W	RC07GF	R5	
1	–	39		RESISTOR 2K Ω ¼W	RC07GF202	R8	
5	5	38		RESISTOR 2.7K Ω ¼W	RC07GF332	R7,8,20,29	
1	1	37		RESISTOR 130 Ω 2W	RC42GF131	R14	
1	–	36		RESISTOR 820 Ω 1W	RC32GF821	R13	
1	–	35		RESISTOR 1.1K Ω 1W	RC32GF[illegible]2	R9	
2	2	34		RESISTOR 36K Ω ¼W	RC07GF363	R22,3	
1	1	33		RESISTOR 100 Ω ¼W	RC07GF101	R1	
6	4	32		DIODE	IN4001	D5,6,7,8,9,	
5	5	31		DIODE	IN914	D2,3,4,9,12	
1	1	30		ZENER DIODE 3V	IN5225B	D1	
8	8	29		22 PIN IC SOCKETS	SA 22-C61T33		C4
2	1	28		40 PIN IC SOCKETS	340-1639-10	U13,33	ALCAT
2	4	27		24 PIN IC SOCKETS	324-1639-10	U14,[illegible]20,35	ALCAT
6	4	26		14 PIN IC SOCKETS	314-1639-10	U1-U6	ALCAT
1	–	25		QUAD RECEIVER - RS232	MC 489	U32	
1	1	24		QUAD TRANSISTOR	MPS6842 V	U31	
1	1	23		MICROPROCESSOR	MC6800	U30	
1	1	22		QUAD EXCLUSIVE OR	SN7486 N	U29	
1	–	21		QUAD TRANSMITTER - RS232	MC 488 N	U46	
2	2	20		QUAD TRI-STATE TRANSCEIVER	DM8833 V	U47,48	
2	2	19		HEX INVERTER w/OC	SN7405 V	U40,41	
1	1	18		DUAL ONE SHOTS	SN74123 N	U38	
2	2	17		QUAD 2-INPUT NAND GATE	SN7400 N	U32,U39	
2	–	16		PHOTO COUPLER	4N33 V	U30,31	
1	1	15		HEX INVERTER	SN7404 N	U29	
1	1	14		QUAD 2-INPUT AND GATE	SN7408 N	U28	
–	–	13		SPARE		U25	
6	5	12		TRI-STATE HEX BUFFER	DM8097 V	U42,44,47,48	
2	2	11		TRIPLE 2-INPUT AND GATE	SN7411 N	U14,15	
1	1	10		4 LINE TO 10 LINE DECODER	SN7442 N	U13	
3	3	9		QUAD 2-INPUT NOR GATE	SN7402 N	U10,4,23	
2	2	8		DUAL D FLIP FLOP	SN7474 N	U9,37	
2	2	7		DUAL ONE SHOT	9602 SN74L221	U8,45	
3	3	6		4 BIT BINARY COUNTER	SN7493 N	U7,22,27	
2	4	5		EPROM 256 X 8 BIT	MM 1024	U6,12,20,35	
2	2	4		8-INPUT NAND GATE	SN7430 N	U4,24	
1	–	3		PERIPHERAL INTERFACE ADAPTER (PIA)	MC6820	U3	
9	8	2		4K DYNAMIC RAM MEMORY	2A0248	U2,5,[illegible],14	34,36
2	2	1		TRI-STATE HEX INVERTER	DM8098 N	U43	
-OE	01			ASSEMBLY			
QTY REQ'D	QTY SHIP'D	ITEM NO	MC SPEC.	DESCRIPTION	IDENTIFYING PART NO.	SPECIFICATION	MATERIAL OR PARTS

LIST OF MATERIAL OR PARTS

CONTRACT No.		SPHERE
DRAWN BY	G. W. / [illegible] 22 DEC 75	CPU/2 ASSEMBLY
CHECKED		
MECH.		
ELECT.		
PROJ. ENGR	[illegible]	
APPROVED	DATE	SIZE D / 000021 / REV
APPROVED (By Others)	DATE	SCALE NONE / SHEET 1 OF 1



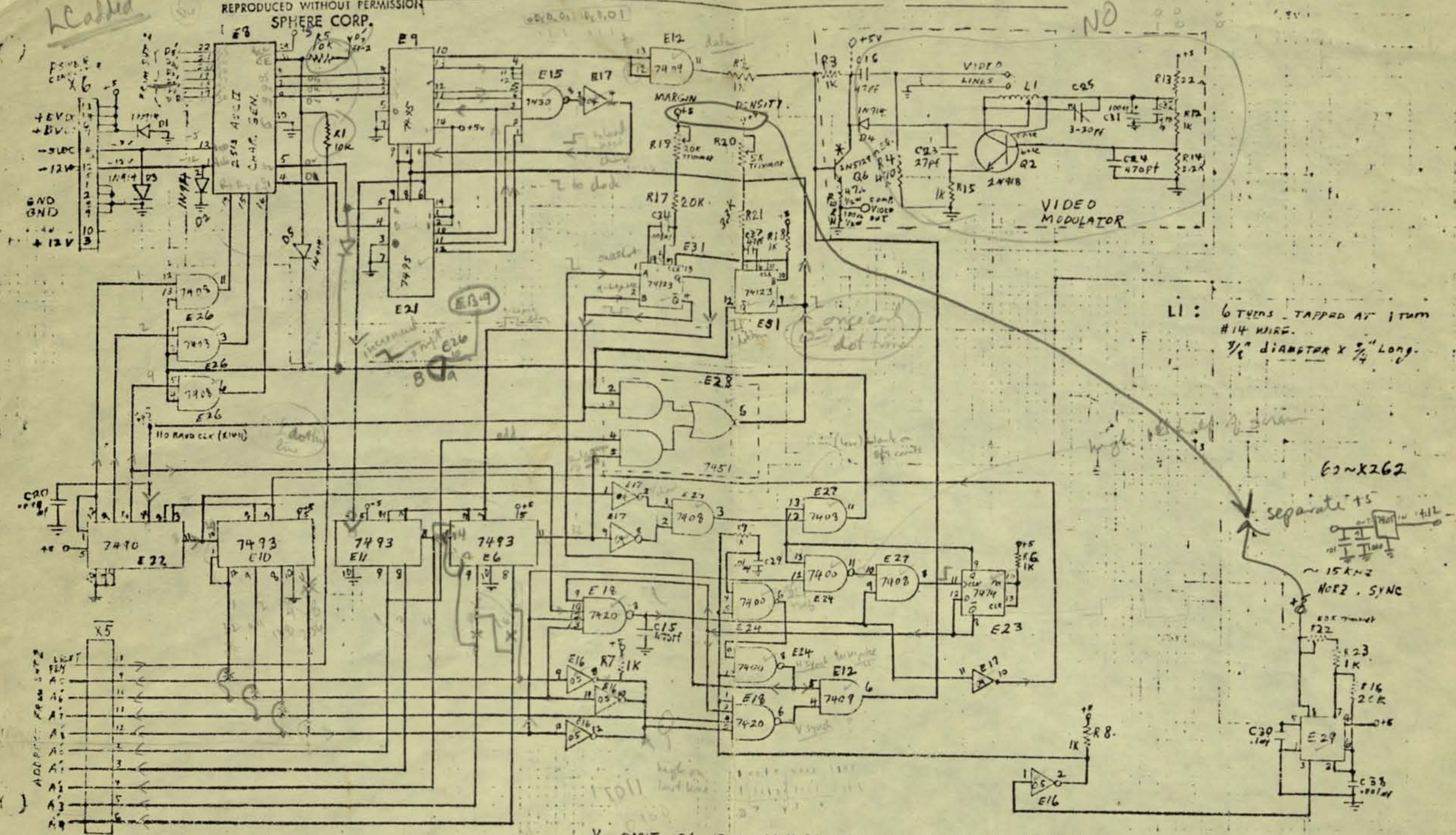
REFRESH


+5
E8-A
HLT
Q
+5
100pF
E37-A
D
Q
7474
+5
E9-A
7474
E9-B
7474
C
Q
Φ1
2ms clock (refresh cycle low)
E22
7453
Φ1
7400
A5
A4
E27
7493
A3
A2
A1
A0
Φ1
E5
8097
(disable A6-15) BA'
(disable A0-5)
+5
E8-A
8602
E8-B
8602
R
Q
150
BA
REFRESH clock to CPU (5 clocks per refresh)
+5
E37-A
7474
D
Q
Φ1
HLT
+5

```
O.ST8   EQU     08H
O.ST9   EQU     04H
;
        CHAIN   FLAGS,2         ;Flag bits
AFLAGS  EQU     FLAGS           ;First byte of flags:
A.CBSY  EQU     80H             ;CCD Busy
A.CERR  EQU     40H             ;CCD Error
A.CFUL  EQU     20H             ;CCD Full
A.BUSY  EQU     10H             ;Timer busy
A.XBSY  EQU     08H             ;RS-232 Busy
A.XMIT  EQU     04H             ;RS-232 Transmit (versus Receive)
;A.BOZZ EQU     02H             ;Annunciator on
BFLAGS  EQU     FLAGS+1         ;Second byte of flags
B.LVAL  EQU     80H             ;Label is valid
B.LONG  EQU     40H             ;Label/Record too long
B.LCMD  EQU     20H             ;Label is command
B.TVAL  EQU     10H             ;Transition is valid
B.DFOR  EQU     08H             ;Last display direction was forward
B.CNFL  EQU     04H             ;CCD is nearly full
B.CEND  EQU     02H             ;CCD is at end of data
B.DOWN  EQU     01H             ;Wand is down
;
        CHAIN   LABUF,RECLEN    ;Label buffer
        CHAIN   RECBUF,RECLEN   ;Record buffer
        CHAIN   LBUF.P,1        ;Label buffer pointer
        CHAIN   RBUF.P,1        ;Record buffer pointer
        CHAIN   RDIS.P,1        ;Record buffer display pointer
        CHAIN   HEDCHR,1        ;Header identifying character
        CHAIN   CMDSEQ,CMDLEN   ;Command identifying sequence
        CHAIN   CSEQ.P,1        ;Command identifying buffer pointer
        CHAIN   X.ATTS,4        ;Transmit attributes
X.BAUD  EQU     X.ATTS          ;Baud rate code
X.BITS  EQU     X.ATTS+1        ;Data bits code
X.PAR   EQU     X.ATTS+2        ;Parity code
X.TERM  EQU     X.ATTS+3        ;Record terminator code
;
        CHAIN   CCDPTS,5*2      ;CCD Pointers (5 two-byte pointers)
C.REDP  EQU     CCDPTS          ;Read pointer
C.CURP  EQU     CCDPTS+2        ;Current pointer
C.WRTP  EQU     CCDPTS+4        ;Write pointer
C.NFUL  EQU     CCDPTS+6        ;"Nearly full" boundary
C.FULL  EQU     CCDPTS+8        ;"Full" boundary
;
        CHAIN   CWLRC,1         ;CCD Write LRC
        CHAIN   CRLRC,1         ;CCD Read LRC
        CHAIN   X.LRC,1         ;Transmit LRC
        CHAIN   X.CHR,2         ;Character currently being transmitted
;
        CHAIN   X.OPRV,3        ;Operating values for transmit routine
X.SLOP  EQU     X.OPRV          ;Slop time
X.BTIM  EQU     X.OPRV+1        ;Number of interrupts per bit
X.BCNT  EQU     X.OPRV+2        ;Number of bits per character
;
        CHAIN   DURTIM,2        ;Duration timer
        CHAIN   DISDUR,2        ;Display duration pre-set
;
;
```

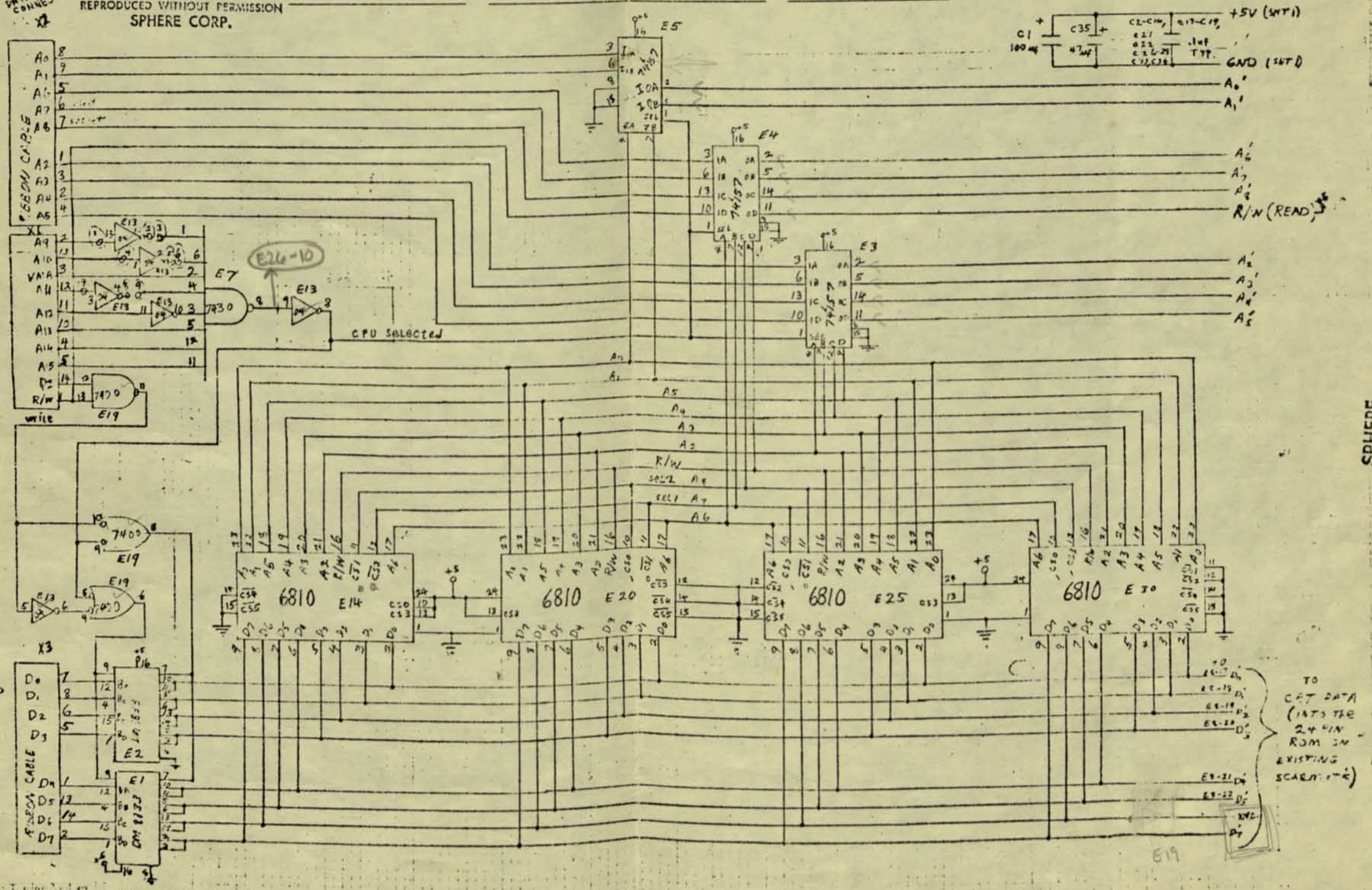
DATE 11/2/75
CHKD. BY
DATE
REVISED 09-06-75 M. KING

SUBJECT CRT/I/O MODULE
SHEET NO. 1 OF 3
JOB NO. 1



SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

BY arct/SGC DATE 15 APR 75
CHKD. BY mat DATE 7/1/75
SUBJECT CRT/I MODULE
SHEET NO. 2 OF 3
JOB NO. 1



SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

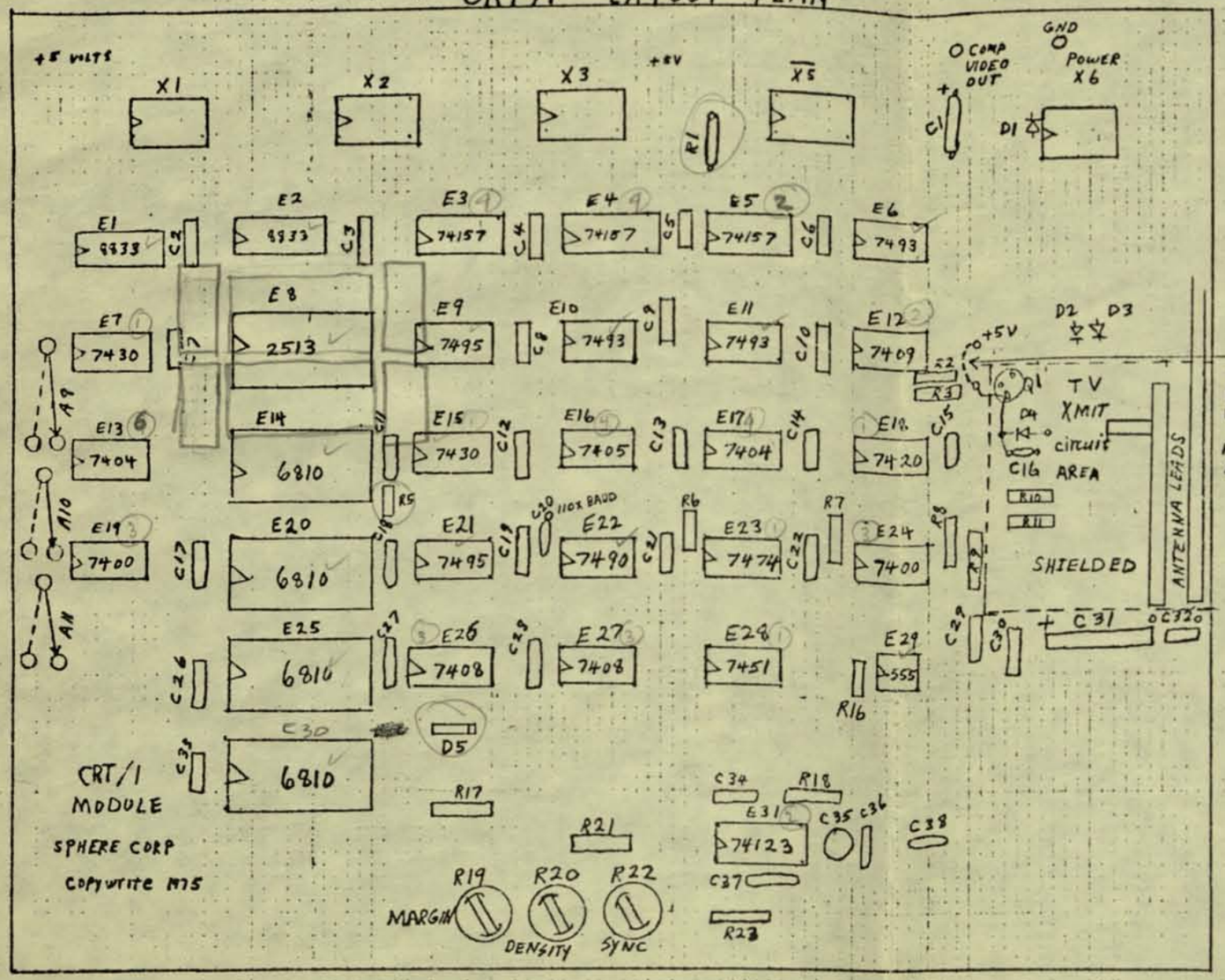
BY_____ DATE_____ SHEET NO. 3 OF 3

CHKD. BY_____ DATE_____ JOB NO._____

NOTES:

1. Address as wired is Binary "0". To change to Address "1" cut A9 etch and add jumper along dotted path. To change to address "5" for example, change connection path of A9 and A1.
2. All IC's have pin 1 in lower left corner
3. Power and ground connect to pins 14 and 7 unless shown otherwise.
4. Item's in ( ) not supplied.

CRT/1 LAYOUT PLAN

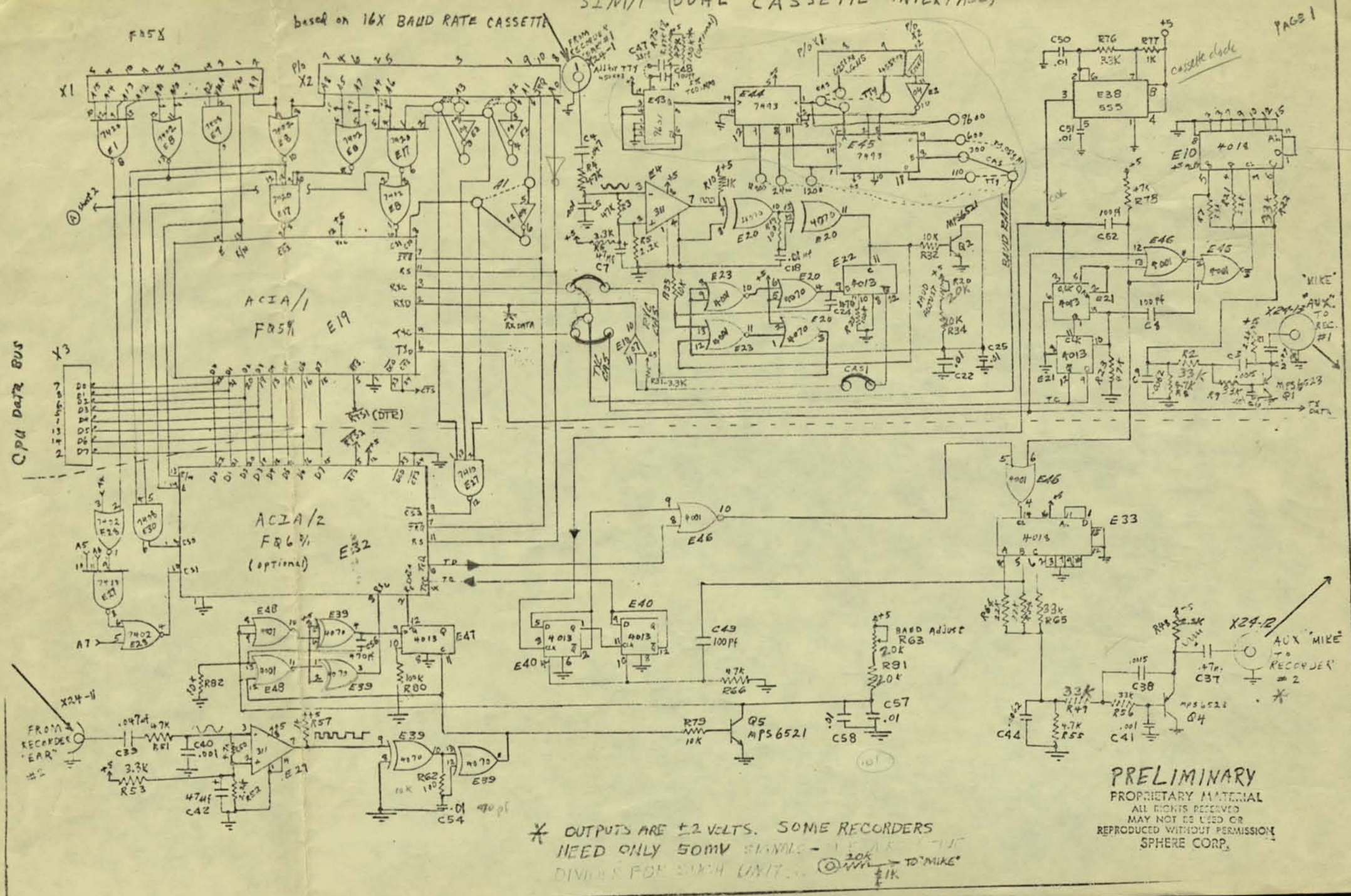


Item	Part	Discription	Designation	Qnty
0	CRT/1	P.C. Board		1
1		14pin socket	X1 - X3, X6, X5	4 (1)
2	DM8833	Quad T/R	E1, E2	2
3	SN74157	Quad MUX	E3, E4, E5	3
4	SN7493	4 bit Bin. Cntr.	E6, E10 E11	3
5	SN7430	NAND Gate	E7, E15	2
6	2513	ASC II Char. Gen	E8	1
7	SN7495	4 bit Shift Reg.	E9, E21	2
8	SN7409	Quad & Gate	E12	1
9	SN7404	HEX Inverter	E13, E17	2
10	MCM6810	128 x 8 Static RAM	E14, E20, E25, E30	4
11	SN7405	HEX Inverter	E16	1
12	SN7420	Dual Nand gate	E18	1
13	SN7400	Quad Nandgate	E24, E19	2
14	SN7490	4bit Dec. Cntr.	E22	1
15	SN7474	Dual D F/F	E23	1
16	SN7408	Quad ANDgate	E26, E27	2
17	SN74123	Dual manostable	E31	1
18	NE555	Timer	E29	1
19	SN7451	dual And/or gate	F28	1
20				
21	2N5129	Transistor	Q1 OR 2N2222A	1
22	2N918	Transistor	Q2	(1)
23	*LI	inductive coil	L1 SEE TVT MODULE IN RADIO ELECT	(1)
24	IN914	Diode	D1 - D5	4 (1)
25	20K	Resistor, var	R19	1
26	5K	Resistor, var	R20	1
27	50K	Resistor, var	R22	1
28	47uf	Cap.-16 vdc	C35	1
29	100uF	Cap.-10 vdc	C1, C31	1 (1)
30	.1uf	Capacitor	C2-C14, C17-C19 C21-C22, C26-28, C30, C33, C36	24
31	.01uf	Capacitor	C29	1
32	.001	Capacitor	C20, C34, C38	1
33	.27pf	Capacitor	C23	(1)
34	47pf	Capacitor	C16, C37	1 (1)
35	470pf	Capacitor	C15, C24, C32	1 (2)
36	8-25pf	Capacitor, VAR	C25	(1)
37	22	Resistor	R13	(1)
38	20K	Resistor	R17, R16	2
39	470	Resistor	R4	(1)
40	1K	Resistor	R2, R3, R6-R9, R12 R15, R18, R23	8 (2)
41	47 1/2w	Resistor	R10	1
42	100 1/2w	Resistor	R11	1
43	~~36K~~	~~Resistor~~	~~R16~~	+
44	10K	Resistor	R1, R5	2
45	2.2K	Resistor	R14	(1)
46	3.3k	RESISTOR	R21	1

SPHERE
791 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010

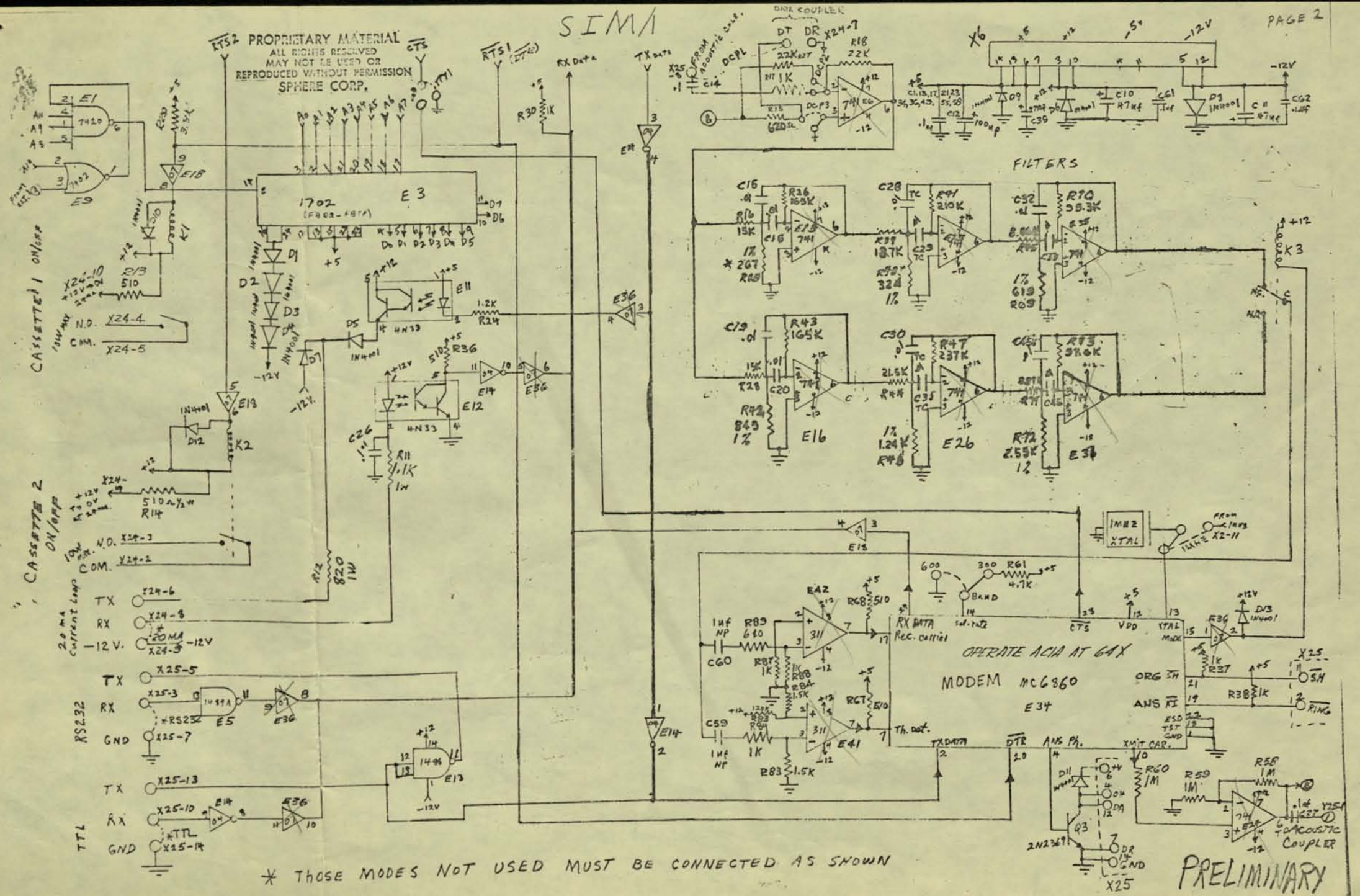
SIM/1 (DUAL CASSETTE INTERFACE)

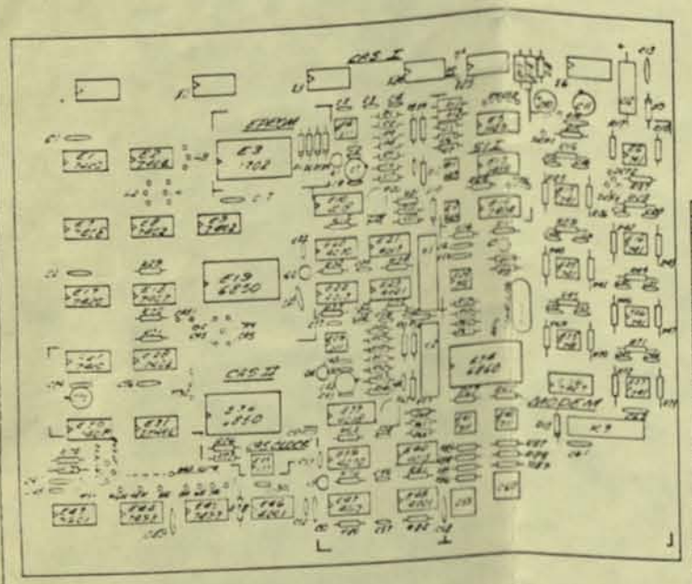
based on 16X BAUD RATE CASSETTE



PAGE 1

* OUTPUTS ARE ±2 VOLTS. SOME RECORDERS NEED ONLY 50mV SIGNAL - ADJUST TO DIVIDE FOR EACH UNIT.





				86	4/R	SCHEMATIC	D100023		
				85	1	PRINTED CIRCUIT BOARD	E000024		
–	1	–	–	84	–	CRYSTAL	1 MHz	XTAL	
–	1	–	–	83	–	RELAY	220-001-12	K3	TRIPRIDGE
1	–	1	–	82	–	RELAY	220-100-12	K1, K2	TRIPRIDGE
–	1	–	–	81	–	TRANSISTOR	2N2369	Q	
1	–	1	–	80	–	TRANSISTOR	MPS6521	Q2, Q5	
1	–	1	–	79	–	TRANSISTOR	MPS6523	Q1, Q3	
–	2	–	–	78		CAPACITOR, 1uf		C53, 60	
–	–	–	–	77	1	CAPACITOR, 100 PF 100V NPO	DTB-100	C48	CRL
–	–	–	–	76	1	CAPACITOR, 33PF 1KV	DD-101	C87	CRL
1	–	1	–	75	–	CAPACITOR, 470 PF 1KV	DD-471	C29,56	CRL
3	12	3	–	74	2	CAPACITOR, .01uf 150V	DDM-103	C16,17,20,21,22,23,24,25,26,27,34,35,37,57 C4,5,30,31,32,33	CRL
–	–	–	–	73	1	CAPACITOR, 100uf 10V		C12	SIEMENS
1	–	1	–	72	–	CAPACITOR .0082uf 1KV	AE0282 1KV	C9, 45	
1	–	1	–	71	1	CAPACITOR 100PF 1KV	DD-101	C8, 43, 52	
2	–	1	–	70	2	CAPACITOR .47uf 16V	RK47/16	C7, 10, 11, 35	ROE
2	–	2	–	69	–	CAPACITOR, .001 1KV	DD-102	C54, 42, 41	CRL
1	–	1	–	68	–	CAPACITOR, .047uf 50V	AE.047M 50V	C44, 39	
1	–	1	–	67	–	CAPACITOR, .0015uf 1KV	DD-152	C9, 38	CRL
1	–	1	–	66	–	CAPACITOR, .47uf 16V		C2, 37	
3	5	0	–	65	7	CAPACITOR, .1uf 50V	CK-10A	C1,3,9,10,14,15,19,28,29,36,40,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64	CRL
–	1	–	–	64	–	RESISTOR, 620Ω 1/4W	RC07GF621	R89	
–	3	–	–	63	–	RESISTOR, 1.5K 1/4W	RC07GF152	R87-85	
–	–	–	–	62	–	RESISTOR	RC07GR394	R74	
–	1	–	–	61	–	RESISTOR 97.6K 1/4W 1%		R73	
–	1	–	–	60	–	RESISTOR 255K 1/4W 1%		R72	
–	1	–	–	59	1	RESISTOR, 8.87K 1/4W 1%		R71, 75	
–	1	–	–	58	–	RESISTOR 35.7K 1/4W 1%		R70	
–	1	–	–	57	–	RESISTOR, 619Ω 1/4W 1%		R69	
–	1	–	–	56	–	RESISTOR, 330Ω 1/4W	RC07GF331	R59	
QTY REQD	QTY REQD	QTY REQD	QTY REQD	ITEM NO	QTY REQ'D	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO	SPECIFICATION	MATERIAL OR NOTE

LIST OF MATERIALS OR PARTS

–	2	–	–	53	–	RESISTOR, 1MEG 1/4W	RC07GF105	R58 60	
–	1	–	–	54	–	RESISTOR, 237K 1/4W 1%		R67	
–	1	–	–	53	–	RESISTOR, 1.24K 1/4W 1%		R66	
–	1	–	–	52	–	RESISTOR, 8.06K 1/4W 1%		R65	
–	1	–	–	51	–	RESISTOR 21.5K 1/4W 1%		R64	
–	1	–	–	50	–	RESISTOR 887Ω 1/4W 1%		R63	
–	1	–	–	49	–	RESISTOR 210K 1/4W 1%		R61	
–	1	–	–	48	–	RESISTOR, 324Ω 1/4W 1%		R62	
–	1	–	–	47	–	RESISTOR, 18.7K 1/4W 1%		R35	
–	3	–	–	46	–	RESISTOR, 510Ω 1/4W	RC07GF511	R34, 37, 38	
1	–	1	–	45	–	RESISTOR, 100K 1/4W	RC07GF104	R31 R	
3	–	3	–	44	–	RESISTOR, 10K 1/4W	RC07GF103	R30, 33, 36	
1		1							
–		–	–	42	–	RESISTOR, 165K 1/4W 1%		R66, 45	
–	1	–	–	41	–	RESISTOR, 267Ω 1/4W 1%		R40	
–	–	–	1	40	–	RESISTOR, 1.2K 1/4W	RC07GF122	R28	
1	–	1	–	39	–	POTENTIOMETER 50K	3352P	R20, 43	50-245
1	–	1	–	38	–	RESISTOR 100Ω 1/4W	RC07GF101	R3, 42	
8	2	8	–	37	–	RESISTOR, 22K 1/4W	RC07GF223	R1, 4, 6, 7, 8, 9, 10, 11	
–	2	–	–	36	–	RESISTOR, 1.5K 1/4W	RC07GF152	R24, 27	
–	1	–	–	35	–	RESISTOR, 200Ω 1/4W	RC07GF201	R25	
–	–	–	–	34	2	RESISTOR, 510Ω 1W	RC32GF511	R32, 44	
–	–	–	1	33	–	RESISTOR, 820Ω 1W	RC32GF821	R12	
–	–	–	1	32	–	RESISTOR, 1.1K 1W	RC32GF112	R13	
1	6	1	0	31	2	RESISTOR, 1K 1W	RC32GF102	R14, 15	
1	1	1	0	30	–	RESISTOR, 47K 1W	RC32GF473	R16, 17	
2	–	1	0	29	1	RESISTOR, 33K 1W	RC32GF333	R18, 19, 21	
3	–	2	0	28	1	RESISTOR, 47K 1W	RC32GF473	R22, 23, 26	
1	–	1	–	27	–	RESISTOR, 33K 1W	RC32GF333	R2, 5, 46	
2	–	2	–	26	–	RESISTOR, 2.2K 1W	RC32GF222	R47, 48	
1	2	1	3	25	7	DIODE	1N4001	D1-D7	
1	1	–	–	24	2	24 PIN IC SOCKETS	3M 3624-3000	11-332-24	
–	–	–	–	23	6	16 PIN IC SOCKETS	3M 3616-3000	11-332-423	
–	–	–	–	22	2	8 BIT BINARY COUNTER	SN 7493N	E44-45	
–	–	–	–	21	1	ONE SHOT	SN74121N	E43	
–	–	–	–	20	1	PRECISION TIMER	SN72555P	E34	
–	1	–	–	19		MODEM	MC6860	E36	
–	–	–	–	18		TIMER		E31	
1	–	–	–	17	–	TRIPLE 3 INPUT NAND GATE	SN7410N	E27	
1	–	1	–	16	1	NOR GATE	SN7402N	E27-42-43	1/
2	–	2	–	15	–	DUAL D FLIP FLOP	SN7474N	E32-33-37	1/
1	–	1	–	14	–	EX-CLUSIVE OR GATE	SN7486N	E30-37	1/
1	–	–	–	13	1	ACIA	MC6850	E35-36	
1	1	1	0	12	1	HEX BUFFER	SN74174N	E17-36	
–	–	–	1	11	–	QUAD TRANSMITTER - RS232	MC1488N	E13	
–	–	–	2	10	–	PHOTO COUPLER	4N33N	E1, E16	
1	–	1	0	9	–	COUNTER	SN7493N	E6, 31	1/
1	–	–	–	8	2	QUAD 2-INPUT NOR GATE	SN7402N	E23, 28	
1	–	–	–	7	1	QUAD 2-INPUT NAND GATE	SN7400N	E7-8	
–	8	–	–	6	–	OPERATIONAL AMPLIFIER	LM741	U1, 3, 4, 7, 8, 9, 10, 11	
–	–	–	1	5	–	QUAD RECEIVER - RS232	MC1489	E5	
1	2	1	–	4	–	VOLTAGE COMPARATOR	LM311N	U2, 5, 6	
–	–	–	–	3	*	EPROM 1K X 8 BIT	MM1702A	E9	
–	–	–	1	2	1	HEX INVERTER	SN7404N	E21A	
–	–	–	–	1	2	8 INPUT NAND GATE	SN7430N	E11	
-05	-04	-03	-02		-01	ASSEMBLY			
QTY REQD	QTY REQD	QTY REQD	QTY REQD	ITEM NO	QTY REQ'D	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO	SPECIFICATION	MATERIAL OR NOTE

LIST OF MATERIALS OR PARTS

05	AUDIO CAS INTERFACE II
04	MODEM
03	AUDIO CAS INTERFACE I
02	SERIAL INTERFACE I
01	STD
DASH NO.	TITLE

05	AUDIO CAS INTERFACE II	d
04	MODEM	c
03	AUDIO CAS INTERFACE I	b
02	SERIAL INTERFACE I	a
01	STD	–
DASH NO.	TITLE	ALPHA SUFFIX

* SPECIAL ITEM, CONTROLED BY SOFT WARE DEPT

DRAWN BY	DATE 28 JAN 76
CHECKED BY	
R&D	
QA CONT	
MFG ENG	
PROD ENG	
APPROVAL	
APPROVAL	
FINAL APPROVAL	

SPHERE

SIM/1

ASSEMBLY

SIZE D 000022 REV A

SCALE NONE SHEET OF

```
00001                    NAM    PDS-SYS2N
00002                    OPT    0,NOG
00003                *
00004                *   PDS SYSTEM 2N  CASSETTE DRIVERS  (SYS2NF)
00005                *
00006                *   PROGRAMMED BY ERIC JAMESON
00007                *
00008                *
00009                *
00010                *   COPYRIGHT 1976 SPHERE CORPORATION
00011                *   940 N.  400 EA.;  NORTH SALT LAKE, UTAH  84054
00012                *
00013                *   P.O. BOX 213;  BOUNTIFUL, UTAH  84010
00014                *
00015                *
00016                *
00017                *
00018                *   SPHERE RESERVES ALL RIGHTS FOR THE REPRODUCTION,
00019                *   DISTRIBUTION AND USE OF THE PDS SOFTWARE.
00020                *   NO COPIES MAY BE MADE OR DISTRIBUTED WITHOUT THE
00021                *   WRITTEN PERMISSION OF SPHERE CORP.
00022                *
00023                *
00024                *
00025                *
00026                *
00027                * THE PROGRAM DEVELOPMENT SYSTEM (PDS SYS2N) IS A SET OF
00028                * PROGRAMS RESIDING ON ERASABLE PROGRAMMABLE READ ONLY
00029                * MEMORY WHICH ALLOW EVEN THE SMALLEST USER TO USE HIS
00030                * SPHERE SYSTEM AS A COMPLETE COMPUTER SYSTEM FOR THE
00031                * DEVELOPMENT OF COMPUTER PROGRAMS.
00032                *     TOWARD THIS END, THE 5 PDS EPROMS CONTAIN A CURSOR
00033                * BASED EDITOR, A MINI-ASSEMBLER, AND THE SPHERE DEBUGGING
00034                * AID (SDA), AS WELL AS A SET OF UTILITY ROUTINES TO DO 16
00035                * BIT MULTIPLY AND DIVIDE, ASCII-TO-BINARY, AND
00036                * BINARY-TO-ASCII ROUTINES, AND ROUTINES TO DO
00037                * INPUT AND OUTPUT TO THE AUDIO CASSETTE.
00038                *
00039                *
00040                *
00041                *     THE SYS2N SOFTWARE IS AN UPGRADE OF THE PDS V3A
00042                * (V3N & V3D) SOFTWARE DESIGNED TO RUN WITH THE CASSETTE
00043                * SYSTEM.  THERE ARE TWO VERSIONS OF THE SYS2 SOFTWARE:
00044                * SYS2N WHICH RUNS WITH THE NEW KEYBOARD AND THE SYS2A
00045                * WHICH RUNS WITH THE ORIGIONAL (KBD/1A) KEYBOARD.  THE
00046                * MAIN DIFFERENCE BETWEEN THE V3A AND SYS2 VERSIONS ARE
00047                * THAT A FIFTH EPROM HAS BEEN ADDED AND THAT THE MINI-
00048                * ASSEMBLER HAS BEEN DELEATED AND REPLACED WITH A SET OF
00049                * COMMANDS TO DO LOADING AND DUMPING OF CASSETTE TAPES.
```

The SYS2NF prom is a version of the SYS2N cassette prom with a software bug fixed. This bug would cause the next block of a multiblock read to be skipped if the checksum on the preceeding block was a 16 and the tape had not been previously used. The change is on page 10 where the ESC test and branch now branches to RDHDR1 instead of RDHDR.

```
00051                    *              MEMORY MAP
00052                    *
00053  0000  TMP    EQU  $00
00054  0002  TMP1   EQU  $02
00055  0004  ARB    EQU  $04        16 BIT ACC. PSEUDO REG B.
00056  0004  AR3    EQU  $04        _HI BYTE OF ARB.
00057  0005  AR2    EQU  $05        _LO BYTE OF ARB.
00058  0006  ARA    EQU  $06        16 BIT ARITH PSEUDO REG A.
00059  0006  AR1    EQU  $06        _HI BYTE OF ARA.
00060  0007  AR0    EQU  $07        _LO BYTE OF ARA.
00061  0008  DIGIT  EQU  $08        BYTE USED BY ASCBIN FOR TMP.
00062  0009  CSTATS EQU  $09        CASSETTE I-O STATUS BYTE.
00063  000A  OUTEND EQU  $0A        END OF OUTPUT BUFFER TEXT.
00064  000C  BUFADR EQU  $0C        START OF I/O BUFFER (PTR)
00065  000E  BUFEND EQU  $0E        PTR. TO END OF I/O BUFFR.
00066  0011  OUTBUF EQU  $11        START OF OUTPUT BUFFER.
00067  0013  CASNUM EQU  $13        PHYSICAL CASSETTE NUMBR.
00068  0014  SRCADR EQU  $14        SOURCE FOR TEXT MOVES.
00069  0016  DSTADR EQU  $16        DEST. ADDR. FOR TEXT MOVE.
00070  001A  ENDMEM EQU  $1A        LAST ADDRES OF REAL MEMORY.
00071  001C  CSRPTR EQU  $1C        PTR TO CURSER ON SCREEN.
00072  001E  BUFPTR EQU  $1E        TEMP PTR USED BY OUTSTR.
00073  0020  BUFFLO EQU  $20        PTR TO END OF LOW EDIT TXT.
00074  0022  BUFFHI EQU  $22        PTR TO START OF HI TEXT.
00075  0024  SCNPTR EQU  $24        PTR. TO BUFFRD TXT START.
00076  0026  SRCASM EQU  $26        PTR TO ASSMBLR SOURCE CODE.
00077  002A  ONDVAL EQU  $2A        HAS ASSMBLR OPERND VALUE.
00078  002C  SYMVAL EQU  $2C        VALUE PUT IN ASSM. SYMTBL.
00079  002E  BRKSAV EQU  $2E        TEMP SAVE FOR BRKPT DATA.
00080  0030  BRKADR EQU  $30        ADDDRESS OF BREAKPOINT.
00081  0032  EDIT   EQU  $32        0 IF EDITOR IS NOT RUNNING.
00082  0033  BLKNAM EQU  $33        CASSETTE BLOCK NAME.
00083  0035  IOBUFF EQU  $35        I-O BUFFER FOR DEBUGGER.
00084  0038  ACIANO EQU  $38        SYS2N CASSETTE ACIA ADDR.
00085  003A  NOPRNT EQU  $3A        CASSETTE NAME PRINT FLAG.
00086  003B  BLKTYP EQU  $3B        CASSETTE BLOCK TYPE CODE.
00087  003C  BFRPTR EQU  $3C        ADDR. OF I-O BUFF. FOR CASS.
00088  003E  BFRSZE EQU  $3E        LENGTH OF CASS. BUFFER.
00089  0040  PCVAL  EQU  $40        PROGRAM COUNTER FOR ASSM.(debug)


00091                    *      FOLLOWING ARE VARIABLE VALUES.
00092                    *
00093  26F0  TIMEA  EQU  $26F0      TIMER COUNTER.
00094  00B1  ON     EQU  $B1        ACIA VALUE TO TURN ON CASS.
00095  0051  OFF    EQU  $51        ACIA VALUE TO STOP CASS.
00096  0003  ETX    EQU  $03        END-OF-TEXT.
00097  0016  SYN    EQU  $16        SYNCHRONISE.
00098  0017  ETB    EQU  $17        END-OF-TRANSMISSION-BLOCK.
00099  001B  ESC    EQU  $1B        EXCAPE TO NONSTANDARD HDR.
00100  0054  ERR4   EQU  $'T        SET  FOR TRAILER ERROR.
00101  0043  ERR5   EQU  $'C        SET  FOR CHECKSUM ERR.
00102  5161  TIME   EQU  20833       TIME CNTR FOR 1/4 SEC.
00103  0009  TIMCNT EQU  9           TIMES FOR 2 & 1/4 SECONDS.
0
```

```
                        PDS SYSTEM 2N CASSETTE DRIVERS
00105                *
00106 FB00                ORG    $FB00
00107                *
00108                *
00109                *          CASSETTE I-O DRIVERS
00110                *
00111                *     THE CASSETTE DRIVERS LOAD AND DUMP A BLOCK OF DATA
00112                * TO AND FROM THE CASSETTE.  THEY HANDLE BOTH THE HEADER
00113                * AND TRAILER FORMATTING.
00114                *     THE DRIVERS ARE SET UP AR A SET OF SUBROUTINES
00115                * CALLED BY THE EXECUTIVE OR THE USER'S PROGRAMS.
00116                *
00117                *
00118                *
00119                *      LOW MEMORY ADDRESSES USED BY THE DRIVERS ARE:
00120                *
00121                *      CSTATS  AT  09    USED TO STORE INPUT ERR CODE.
00122                *      BLKNAM  AT  33    2 CHAR. NAME OF BLOCK.
00123                *      ACIANO  AT  38    PTR. TO THE ACIA CURRENTLY USED.
00124                *      NOPRNT  AT  3A    PRINT FLAG FOR BLOCK NAME.
00125                *      BFRPTR  AT  3C    START OF CASSETTE I-O BUFFER.
00126                *      BFRSZE  AT  3E    END OF CASSETTE DATA BUFFER.
00127                *
00128                *
00129                *     THE DRIVERS CAN BE RUN FROM THE PDS-SYS2N SOFTWARE
00130                * SYSTEM, WHICH CONTAIN THE CASSETTE LOAD AND DUMP
00131                * COMMANDS IN THE EXEC, OR FROM THE PDS-V3A, WHERE THE
00132                * DRIVERS ARE CALLED THROUGH THE DEBUGGER.
00133                *
00134                *     THE DRIVERS DISPLAY PERTINENT DATA ON THE SCREEN
00135                * WHEN WRITING OR READING FROM THE CASSETTES.  ON A WRITE,
00136                * THE CHARACTERS BEING WRITTEN ONTO THE CASSETTE ARE
00137                * DISPLAYED IN THE SECOND CHARACTER POSITION FROM THE
00138                * UPPER RIGHT HAND CORNER OF THE SCREEN.  ON A READ, THE
00139                * CHARACTERS BEING READ IN ARE DISPLAYED IN THE UPPER
00140                * RIGHT HAND CORNER OF THE SCREEN.  THE NAME OF THE BLOCK
00141                * CURRENTLY BEING READ OR SEARCHED OVER IS DISPLAYED ON THE
00142                * RIGHT HAND SIDE OF THE SECOND LINE.  THE ERROR CODE FOR
00143                * A READ IS DISPLAYED ON THE RIGHT SIDE OF THE THIRD LINE.
00144                * NO CHANGE IN CHARACTER MEANS THAT THE READ WAS O.K.
00145                * A "C" MEANS THERE WAS A CHECKSUM ERROR ON THE BLOCK.
00146                * A "T" MEANS THAT THE WRONG NUMBER OF BYTES WERE READ
00147                *  INTO THE BUFFER (TRAILER ERROR).  THIS WOULD OF COURSE
00148                *  IMPLY A CHECKSUM ERROR ALONG WITH THE TRAILER ERROR.
00149                *
00150                *     THE CSTATS (CASSETTE STATUS) BYTE WILL CONTAIN A 0
00151                * IF THE BLOCK READ IN WAS O.K.  IF IT WAS A BAD READ, IT
00152                * WILL CONTAIN A 54 FOR TRAILER ERRROR OR A 43 FOR A
00153                * CHECKSUM ERROR CODE UPON EXIT FROM THE ROUTINE.
00154                *
00155                *     SETTING THE NOPRNT (NO PRINTING) FLAG TO A 0 WILL
00156                * STOP THE DISPLAY OF CHARACTERS ON THE SCREEN DURING
00157                * CASSETTE READ AND WRITE, EXCEPT FOR THE T & C ERROR
00158                * CODES, WHICH ARE ALWAYS DISPLAYED WHEN THEY OCCUR.
```

```
00160          *     IF THE FIRST BYTE OF BLKNAM (BLOCK NAME) IS A 0
00161          * WHEN THE READ BLOCK ROUTINE IS ENTERED, THE NEXT BLOCK
00162          * WILL BE READ FROM TAPE NO MATTER WHAT THE NAME OF THE
00163          * TAPE BLOCK IS.  ON THE SYS2N EXEC, A CONTROL SPACE
00164          * CHARACTER CAN BE TYPED IN AS THE FIRST CHARACTER OF THE
00165          * NAME IN THE LOAD BLOCK COMMAND.  THUS, A (CNTL L)(CNTL
00166          * SPACE)(X) WOULD READ IN THE NEXT BLOCK ON THE TAPE.
00167          *
00168          *
00169          *
00170          *     THE TAPE FORMAT IS:
00171          *       SYN   16
00172          *       SYN   16
00173          *       SYN   16
00174          * SOH   ESC   1B
00175          *       HI BYTE OF 16 BIT BLOCK LENGTH  } (no. of data characters on tape)
00176          *       LOW BYTE OF BLOCK LENGTH        }
00177          *       FIRST CHAR OF BLOCK NAME
00178          *       SECOND CHARACTER OF NAME
00179          * STX   DATA
00180          *       .
00181          *       .
00182          *       .
00183          *       DATA
00184          * ETX   ETB   17
00185          *       CHECKSUM
00186          *       CHECKSUM
00187          *       CHECKSUM
00188          *       CHECKSUM
00189          *
00190          *
00191          *
00192          *     THE CHECKSUM IS CALCULATED FROM THE DATA, WHICH
00193          * IS READ IN FROM THE CASSETTE BUFFER RESIDING IN MEMORY.
00194          *
00195          *
00196          *     THE FORMAT FOR THE CASSETTE BUFFER IS:
00197          *
00198          *          BFRPTR                              BFRSZE
00199          *            |                                   |
00200          *            v                                   v
00201          *          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
00202          *          |C|A|S|S|E|T|T|E| | | |B|U|F|F|E|R|
00203          *          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
00204          *
00205          *     WHEN DATA IS READ IN, THE READ ROUTINE SETS 'BFRSZE'
00206          * TO POINT TO THE LAST CHARACTER READ INTO THE BUFFER.
00207          * NOTE THAT THERE IS NO OVERFLOW CHECK WHEN DATA IS READ
00208          * INTO MEMORY.  ON OUTPUT TO THE CASSETTE, THE BLOCK
00209          * LENGTH IS CALCULATED FROM THE BFRPTR AND BFRSZE POINTERS.
00210          *
00211          *     THE READ BLOCK AND WRITE BLOCK ROUTINES AUTOMATICALLY
00212          * TURN ON AND OFF THE CASSETTE.
```

```
00214                *
00215                *     THE ADDRESSES OF THE ROUTINES FOR CONTROLLING
00216                *     THE CASSETTE ARE AS FOLLOWS:
00217                *
00218                *     INTLZ  - FB00 :  INITIALIZES THE ACIA FOR USE.
00219                *     WRTBLK - FB2D :  WRITES A  FORMATTED BLOCK TO CASSETTE
00220                *     RDBLK  - FB91 :  READS IN A FORMATTED BLOCK OF TAPE
00221                *     WRTMOD - FB2F :  WRITES A BLOCK TO A RUNNING CASSETTE
00222                *     RDMOD  - FB93 :  READS A BLOCK FROM A RUNNING TAPE.
00223                *     CASOUT - FB62 :  WRITES OUT ONTO TAPE THE BYTE IN A.
00224                *     CASIN  - FB7E :  READS IN A BYTE INTO ACCUM. A
00225                *     TURNON - FB77 :  TURNS ON THE CASSETTE DRIVE.
00226                *     TRNOFF - FBB0 :  TURNS OFF THE CASSETTE DRIVE WHEN THE
00227                *                      POINTR TO ACIA IS PASSED IN THE X REG.
00228                *
00229                *    NOTE THAT ALL THE ABOVE ROUTINES ARE SUBROUTINES THAT
00230                *     ARE ENTERED BY A JSR OR BSR CALL.
00231                *
00232                *
00233                *
00234                *
00235                *              USING THE CASSETTE DRIVERS.
00236                *
00237                *     IF THE EPROMS ON THE CPU BOARD ARE THE SYS2N OR THE
00238                * SYS2D PROMS, THE DRIVERS ARE USED BY THE I (INITIALIZE),
00239                * L (LOAD FROM CASSETTE) AND S (STORE ONTO CASSETTE)
00240                * COMMANDS THAT ARE A PART OF THE SYS-2 EXECUTIVE.
00241                *
00242                *     IF THE CPU PROMS BEING USED ARE THE PDS-V3A OR THE
00243                * PDS-V3N OR V3D PROM SETS, THEN THE USER MUST SET UP THE
00244                * POINTERS AND CALL THE ROUTINES HIMSELF, USING THE
00245                * DEBUGGER ON THE CPU PROMS.  (NOTE THAT THE V3A AND V3N
00246                * DEBUGGERS VARY, FOR INSTANCE THE V3A GO COMMAND IS A 'G'
00247                * WHILE THE V3N GO COMMAND IS A 'CNTL G'.  REFER TO THE
00248                * USERS MANUAL FOR DETAILES.)
00249                *     THE FIRST THING TO DO IS OPEN THE LOW MEMORY LOCATIONS
00250                * USED BY THE DRIVERS AS FLAGS AND POINTERS AND INITIALIZE
00251                * THE LOCATIONS.  THE ACIANO, BLKNAM, BFRPTR AND BFRSZE
00252                * SHOULD NOW BE GIVEN VALUES.
00253                *     ACIANO WOULD POINT TO F050 FOR THE FIRST CASSETTE AND
00254                * TO F060 FOR THE SECOND CASSETTE DRIVE.
00255                *     A SIMPLE PROGRAM TO CALL THE ROUTINES SHOULD NOW BE
00256                * WRITTEN.  OPEN LOCATION 900.  TYPE IN THROUGH THE DEBUGGER
00257                * THE INSTRUCTIONS JSR BDXX, JMP FE4F.  XX IS THE SECOND BYTE
00258                * OF THE DESIRED DRIVER ROUTINE, I.E. 00 FOR INITIALLIZATION
00259                * OR 91 FOR READING A BLOCK OR 2D FOR WRITING A BLOCK.  THUS,
00260                * TO INITIALIZE AN ACIA, ACIANO WOULD BE LOADED WITH
00261                * THE ADDRESS OF THE ACIA AND THE USER WOULD THEN JUMP TO THE
00262                * ROUTINE AT 900 BY OPENING 900 AND JUMPING TO IT WITH
00263                * THE 'G' COMMAND.  THE ROUTINE WOULD BE AS FOLLOWS:
00264                *
00265                *CALL 900  BD     JSR
00266                *     901  FB     HI BYTE OF ROUTINE AADDRESS
00267                *     902  00     LOW BYTE OF DRIVER ADDRESS.
00268                *     903  7E     JMP
00269                *     904  FE     ADDRESS OF THE DEBUGGER ON
00270                *     905  4F     THE PDS V3A PROM SET
0
```

```
00272                 *     THE ACIA WOULD NOW BE INITIALIZED AND CONTROL WOULD
00273                 * HAVE RETURNED TO THE DEBUGGER.  TO READ IN A BLOCK,
00274                 * THE NAME WOULD BE PUT IN BLKNAM AND LOCATION 902 WOULD
00275                 * BE CHANGED TO 91.  THE USER WOULD THEN  JUMP TO 900.
00276                 *
00277                 *     TO TEST THE CASSETTE, SET BFRPTR TO E060 AND BFRSZE
00278                 * TO E0DF.  THIS WILL ALLOW THE USER TO WRITE OUT DATA
00279                 * FROM THE FOURTH, FIFTH, SIXTH AND SEVENTH LINE OF THE
00280                 * CRT DISPLAY AND THEN READ IT BACK ONTO THE DISPLAY.  DATA
00281                 * CAN BE TYPED ONTO THE SCREEN BY OPENING A LOCATION WITH
00282                 * THE DEBUGGER AND THEN MOVING THE CURSOR AROUND THE SCREEN
00283                 * TO CHANGE THE CHARACTERS.  THIS IS POSSIBLE BECAUSE THE
00284                 * ROUTINE TO INPUT AN ADDRESS CALLS THE EDITOR FOR INPUT.
00285                 *
00286                 *     THE DRIVER ROUTINES CAN ALSO BE USED TO PERFORM
00287                 * I-O WITH A MODEM OR TELETYPE.  THE MAIN HARDWARE
00288                 * DIFFERENCE BETWEEN THE CASSETTE AND MODEM/TTY IS THAT THE
00289                 * CASSETTE HAS A DIVIDE BY 16 CLOCK AND IS UNIDIRECTIONAL
00290                 * WHILE THE MODEM/TTY HAVE A DIVIDE BY 64 CLOCK AND CAN BE
00291                 * BIDIRECTIONAL.  BECAUSE OF THE CLOCK CHANGE THE ACIA MUST
00292                 * BE TURNED ON WITH A DIFFERENT VALUE BEFORE THE READ BLOCK
00293                 * OR WRITE BLOCK ROUTINES ARE ENTERED.  TO TURN ON THE ACIA
00294                 * STORE THE VALUE '82' INTO LOCATION F050 OR F060.  ONCE
00295  B2 will produce* IT IS TURNED ON, EITHER CASIN OR CASOUT MAY BE CALLED
00296  a signel of eight* REPEATEDLY OR WRITE MODEM BLOCK (WRTMOD) OR READ MODEM
00297  data bits and 2* (RDMOD) CAN BE CALLED ONCE.  WRTMOD & RDMOD ARE THE SAME
00298  stop bits.  If* AS WRTBLK & RDBLK EXCEPT THEY DO NOT TURN ON THE ACIA.  A
00299  the TTY operates* PROGRAM TO READ IN A BLOCK OF DATA FROM A MODEM OR TTY
00300  on a different* WOULD THUS BE (PLACED BEFORE THE 'CALL' ROUTINE):
00301  code, look up *     8FA  DE 38  LDX   ACIANO   LOADS ACIA POINTTER.
00302  the proper init-*     8FC  86 B2  LDA A  #$B2     LOADS STARTUP VALUE.
00303  ialization value*     8FE  A7 00  STA A  0,X      PUTS START CODE INTO ACIA.
00304  in the ACIA   * LOCATION 902 WOULD NOW BE 2F FOR WRITING AND 93 TO READ.
00305                 *     TO READ IN A CHARACTER FROM THE TELETYPE TURN ON THE
00306  section of the * ACIA AND GO TO THE FOLLOWING ROUTINE:
00307                 *     TTYIN  JSR   CASIN   READS IN A CHAR FROM KEYBOARD.
00308  chip description*            JMP   CASOUT  TYPES OUT CHAR ON PRINTER.
00309  appendex.      * THE RS232 SHOULD HAVE BEEN STRAPPED TO HALF DUPLEX.
00310                 *
00311                 *
00312                 *     IT IS TO BE STRESSED THAT  THE RELIABILITY OF THE
00313                 * CASSETTE CONTROLLER DEPENDS ON THE ADJUSTMENT OF THE
00314                 * TRIMMER ON THE SIM BOARD.  IF THE TRIMMER IS OUT OF
00315                 * ADJUSTMENT THE DATA WILL NOT READ IN PROPERLY.  BESIDES
00316                 * USING THE OSCILLOSCOPE TO ADJUST THE TRIMMER, IT CAN ALSO
00317                 * BE ADJUSTED BY READING IN A STRING OF SINGLE CHARACTERS
00318                 * FROM  THE CASSETTE AND ADJUSTING IT UNTIL THE CHARACTERS
00319                 * SYNC IN PROPERLY.  THE BEST CHARACTER TO USE IS A STRING
00320                 * OF 'U'S.  TO READ IN THE STRING FOR TESTING USE, WRITE A
00321                 * LOOP TO GET A CHARACTER FROM THE CASSETTE AND THEN
00322                 * DISPLAY THAT CHARACTER.  A SAMPLE ROUTINE TO DO THIS IS:
00323                 *     A  JSR  $FB7E   LOADS A WITH CASSETTE CHAR.
00324                 *        JSR  $FCAD   PDS-V3A PUTCHR ROUTINE.
00325                 *        TST  $F001   TESTS KEYBOARD FOR A KEY.
00326                 *        BPL  A       SKIPS BACK IF NO INPUT.
00327                 * THE ABOVE ROUTINE WOULD INPUT CHARACCTERS UNTIL A KEY
00328                 * ON THE KEYBOARD WAS DEPRESSED.
0
```

```
00330                    *       INTLZ INITIALIZES THE ACIA CONTROLLER FOR
00331                    *       A SPECIFIC TAPE UNIT AT MOUNT TIME.
00332                    *
00333 FB00 DE 38   INTLZ  LDX     ACIANO   X GETS ACIA ADDRESS.
00334 FB02 86 13          LDA A   #$13     RESETS THE ACIA.
00335 FB04 A7 00          STA A   0,X      .
00336 FB06 86 51          LDA A   #0FF     SETS ACIA TO
00337 FB08 A7 00          STA A   0,X      /16 2 STOPS. BIT FORMAT.
00338 FB0A 39             RTS

00340                    *       WRTHDR FORMATS THE HEADER ON THE TAPE.
00341                    *
00342 FB0B 86 16   WRTHDR LDA A   #SYN     PUTS SYNC CHARS ONTO TAPE.
00343 FB0D 8D 53          BSR     CASOUT4  .
00344 FB0F 8D 51          BSR     CASOUT   .
00345 FB11 8D 4F          BSR     CASOUT   .
00346 FB13 86 1B          LDA A   #ESC     .
00347 FB15 8D 4B          BSR     CASOUT   .
00348 FB17 96 3E          LDA A   BFRSZE   FOLLOWING OUTPUTS LENGTH.
00349 FB19 D6 3F          LDA B   BFRSZE+1 LOADS LO BYTE OF END PTR.
00350 FB1B D0 3D          SUB B   BFRPTR+1 SUBS LO BYTE OF BEGIN PTR.
00351 FB1D 92 3C          SBC A   BFRPTR   SUBS HI BYTE OF START PTR.
00352 FB1F 8D 41          BSR     CASOUT   OUTPUTS HI LENGTH BYTE.
00353 FB21 17             TBA              LOADS LO BYTE OF LENGTH.
00354 FB22 8D 3E          BSR     CASOUT   OUTPUTS LOW LENGTH BYTE.
00355 FB24 96 33          LDA A   BLKNAM   PUTS OUT NAME OF BLOCK.
00356 FB26 8D 3A          BSR     CASOUT   .
00357 FB28 96 34          LDA A   BLKNAM+1 .
00358 FB2A 8D 36          BSR     CASOUT   PUTS OUT LAST OF NAME.
00359 FB2C 39             RTS              RETURNS BACK TO WRTBLK.

00361                    *       WRTBLK WRITES OUT A BLOCK OF DATA TO THE CASSETTE.
00362                    *
00363 FB2D 8D 48   WRTBLK BSR     TURNON   TURNS ON THE CASSETTE.
00364                    *        FOLLOWING WAITES FOR CASSETTE TO GET UP TO SPEED.
00365 FB2F C6 09   WRTMOD LDA B   #TIMCNT  LOADS TIME LOOP COUNTER.
00366 FB31 CE 5161 TIME1  LDX     #TIME    MASTER TIME LOOP (1/4 SEC).
00367 FB34 09      TIME2  DEX              COUNTS CYCLES OF LOOP.
00368 FB35 26 FD          BNE     TIME2    TESTS FOR FIRST TIME OUT.
00369 FB37 5A             DEC B   -        COUNTS TIMES IN LOOP.
00370 FB38 26 F7          BNE     TIME1    SKIPS BACK UNTIL DONE.
00371                    *        THE TIME LOOP IS NOW FINISHED.
00372 FB3A 8D CF          BSR     WRTHDR   WRITES HEADER ON THE TAPE.
00373 FB3C 8D 10          BSR     WRTBFR   WRITES OUT BUFFER DATA.
00374                    *        FOLLOWING WRITES THE TRAILER OUT ONTO THE TAPE.
00375 FB3E 86 17   WRTTLR LDA A   #ETB     OUTPUTS END-OF-BLOCK CHAR.
00376 FB40 8D 20          BSR     CASOUT   ETB IS DISPLAYED AS A "W".
00377 FB42 17             TBA              A GETS CHECKSUM FROM B.
00378 FB43 8D 1D          BSR     CASOUT   OUTPUTS THE CHECKSUM.
00379 FB45 8D 1B          BSR     CASOUT   OUTPUTS TRAILER FILLER
00380 FB47 8D 19          BSR     CASOUT   BYTES.
00381 FB49 8D 17          BSR     CASOUT4  .
00382                    *        END OF TRAILER WRITING ROUTINE.
00383 FB4B 8D 63          BSR     TRNOFF   HALTS CASSETTE DRIVE.
00384 FB4D 39             RTS              -
```

```
00387                    *       WRTBFR WRITES OUT THE CONTENTS  OF THE
00388                    *       BUFFER ONTO THE CASSETTE TAPE.
00389                    *
00390                    *
00391 FB4E 5F      WRTBFR CLR B   -          INIT CHECKSUM COUNT.
00392 FB4F DE 3C         LDX     BFRPTR     ACIA CONTROL MASK.
00393 FB51 A6 00   WBFR1  LDA A   0,X        LOADS CHAR. FROM BUFFER.
00394 FB53 DF 00         STX     TMP        SAVES BUFFER PTR.
00395 FB55 8D 08         BSR     CASOUT     PUTS CHAR ONTO CASSETTE.
00396 FB57 1B            ABA                A GETS A+B.
00397 FB58 16            TAB                B GETS A+B.
00398 FB59 DE 00         LDX     TMP        RESTORES PTR. INTO BUFFR.
00399 FB5B 9C 3E         CPX     BFRSZE     TESTS IF BUFFER EMPTY.
00400 FB5D 27 17         BEQ     CSOEXT     EXITS WHEN EMPTY.
00401 FB5F 08            INX                INC POINTER.
00402 FB60 26 EF         BNE     WBFR1      SKIPS BACK IF CHARS.  LEFT.
```

```
00404                    *       CASOUT TAKES THE CHAR IN A AND PUTS IT OUT
00405                    *       ONTO THE CASSETTE TAPE.
00406                    *
00407                    *
00408 FE62 36      CASOUT PSH A   .          SAVES CHAR TO READ OUT.
00409 FB63 DE 38         LDX     ACIANO     X GETS PHYSICAL ACIA ADDR.
00410 FB65 86 02         LDA A   #2         LOADS CONTROL TEST BITS.
00411 FB67 A5 00   CASO1  BIT A   0,X        TESTS IF ACIA BUFFER EMPTY.
00412 FB69 27 FC         BEQ     CASO1      LOOPS BACK UNTIL READY.
00413 FB6B 32            PUL A   .          GETS ORIG CHAR.
00414 FB6C A7 01         STA A   1,X        STORS CHAR INTO ACIA BUFFER.
00415 FB6E 70 003A       TST     NOPRNT     TESTS IF PRINTOUT ALLOWED.
00416 FB71 27 03         BEQ     CSOEXT     SKIPS PRINTING IF A 0.
00417 FB73 B7 E01E       STA A   $E01E      DISPLAYS CHAR ON SCREEN.
00418 FB76 39      CSOEXT RTS
```

```
00420                    *       TURNON TURNS ON THE CASSETTE DRIVE.
00421                    *
00422                    *
00423 FB77 DE 38   TURNON LDX     ACIANO     LOADS CASSETTE ACIA ADDRESS.
00424 FB79 86 B1         LDA A   #ON
00425 FB7B A7 00         STA A   0,X        TURNS ACIA ON.
00426 FB7D 39            RTS
```

```
00429                    *      CASIN READS IN A CHARACTER FROM THE CASSETTE TAPE
00430                    *      INTO THE A ACCUMULATOR.
00431                    *
00432                    *
00433 FB7E DE 38   CASIN  LDX    ACIANO   X GETS THE ACIA ADDRESS.
00434 FB80 86 01         LDA A  #1       LOADS TEST BITS.
00435 FB82 A5 00   CASIN1 BIT A  0,X      TESTS IF ACCIA BUFFR FULL.
00436 FB84 27 FC         BEQ    CASIN1   SKIPS BACK IF NOT IN YET.
00437 FB86 A6 01         LDA A  1,X      LOADS IN CHAR FROM TAPE.
00438 FB88 7D 003A       TST    NOPRNT   TESTS IF PRINT IS OFF.
00439 FB8B 27 03         BEQ    CINEXT   SKIPS DISPLAYING IF 0.
00440 FB8D B7 E01F       STA A  $E01F    DISPLAYS CHR ON TV.
00441 FB90 39       CINEXT RTS            .



00443                    *      RDBLK READS IN A BLOCK FROM THE CASSETTE
00444                    *      TAPE INTO BUFFER MEMORY.
00445                    *
00446                    *
00447 FB91 8D E4   RDBLK  BSR    TURNON   TURNS ON TAPE DRIVE.
00448 FB93 8D 20   RDMOD  BSR    RDHDR    READS IN THE HEADER.
00449 FB95 8D 55         BSR    RDBFR    READS IN DATA INTO BUFFER.
00450                    *       FOLLOWING READS IN TRAILER AND CHECKS CHECKSUM.
00451 FB97 4F      RDTLR  CLR A  -        LOADS A 0 FOR A GOOD READ.
00452 FB98 97 09         STA A  CSTATS   SETS STATUS BYTE TO NO ERR.
00453 FB9A 8D E2         BSR    CASIN    INPUTS END-OF-BLOCK CHAR.
00454 FB9C 81 17         CMP A  #ETB     ETB DISPLAYS AS A "W".
00455 FB9E 27 04         BEQ    RTLR1    SKIPS IF NO ETB ERROR.
00456 FBA0 86 54         LDA A  #ERR4    LOADS TRAILER ERROR CODE.
00457 FBA2 20 07         BRA    RTLR2    SKIPS TO STORE ERROR CODE.
00458 FBA4 8D D8   RTLR1  BSR    CASIN    READS IN CHECKSUM.
00459 FBA6 11            CBA             TESTS CHECKSUM.
00460 FBA7 27 07         BEQ    TRNOFF   SKIPS IF OK.
00461 FBA9 86 43         LDA A  #ERR5    SETS CHECKSUM ERROR CODE.
00462 FBAB 97 09   RTLR2  STA A  CSTATS   SETS ERROR STATUS BYTE.
00463 FBAD B7 E05F       STA A  $E05F    DISP.  ERR CODE ON SCREEN.
00464                    *       END OF TRAILER READ IN.
00465                    *       FOLLOWING TURNS OFF THE CASSETTE DRIVE.
00466 FBB0 86 51   TRNOFF LDA A  #OFF     LOADS COMMAND TO TURN
00467 FBB2 A7 00         STA A  0,X      OFF ACIA CASSETTE DRIVE.
00468 FBB4 39            RTS             .
```

LDX ACIANO

```
00471                    *       RDHDR FINDS THE START OF THE BLOCK ON THE TAPE,
00472                    *       FIGURES THE BUFFER END AND CHECKS THE NAME.
00473                    *
00474                    *
00475 FBB5 8D C7   RDHDR  BSR     CASIN     FOLLOWING FINDS THE
00476 FBB7 81 16   RDHDR1 CMP A   #SYN      START OF THE HEADER.
00477 FBB9 26 FA          BNE     RDHDR     .
00478 FBBB 8D C1          BSR     CASIN     TESTS FOR HEADR CHAR.
00479 FBBD 81 1B          CMP A   #ESC      START-OF-HEADER.
00480 FBBF 26 F6          BNE     RDHDR1    GOES BACK IF NOT GOOD HDR.
00481 FBC1 8D BB          BSR     CASIN     READS IN HI BYTE OF LEN.
00482 FBC3 16             TAB               SAVES HI LENGTH BYTE.
00483 FBC4 8D B8          BSR     CASIN     INPUTS LO SIZE (LEN) BYTE.
00484 FBC6 9D 3D          ADD A   BFRPTR+1  FORMS POINTER TO THE
00485 FBC8 D9 3C          ADC B   BFRPTR    TOP BYTE OF THE BUFFER.
00486 FBCA 97 3F          STA A   BFRSZE+1  SAVES THE HI BUFF. PTR.
00487 FBCC D7 3E          STA B   BFRSZE    TO THE CASSETTE BUFFER.
00488 FBCE 8D AE          BSR     CASIN     READS IN BLOCK NAME.
00489 FBD0 36             PSH A   -         SAVES FIRST CHAR OF NAME.
00490 FBD1 8D AB          BSR     CASIN     READ IN SECOND CHAR INTO A.
00491 FBD3 D6 3A          LDA B   NOPRNT    TESTS IF PRINT IS OFF.
00492 FBD5 33             PUL B   -         RESTORES FIRST NAME CHAR.
00493 FBD6 27 06          BEQ     RHDR1     SKIPS IF PRINT FLAG IS 0.
00494 FBD8 F7 E03E        STA B   $E03E     DISPLAYS BLOCK NAME ON
00495 FBDB B7 E03F        STA A   $E03F     THE CRT SCREEN.
00496 FBDE 7D 0033 RHDR1  TST     BLKNAM    TESTS IF NAME IS CHECKED.
00497 FBE1 27 08          BEQ     RHDR2     SKIPS IF NO NAME CHECK.
00498 FBE3 D1 33          CMP B   BLKNAM    TESTS FIRST CHAR OF NAME.
00499 FBE5 26 CE          BNE     RDHDR     SKIPS BACK IF BAD NAME
00500 FBE7 91 34          CMP A   BLKNAM+1  TESTS SECOND NAME CHAR.
00501 FBE9 26 CA          BNE     RDHDR     SKIPS BACK IF BAD NAME.
00502 FBEB 39        RHDR2 RTS               .

00504                    *       RDBFR READS DATA INTO THE MEMORY BUFFER FROM
00505                    *       THE CASSETTE.
00506                    *
00507                    *

00508 FBEC 5F        RDBFR CLR B   -         INIT B FOR CHECKSUM.
00509 FBED DE 3C          LDX     BFRPTR    LOADS START OF BUFFER.
00510 FBEF DF 00     RBFR1 STX     TMP       .
00511 FBF1 8D 86          BSR     CASIN     A GETS CHAR READ IN.
00512 FBF3 DE 00          LDX     TMP       X GETS BUFFER PTR.
00513 FBF5 A7 00          STA A   0,X       STORS CHAR INTO BUFFER.
00514 FBF7 1B             ABA               A GETS A+B.
00515 FBF8 16             TAB               B GETS A.
00516 FBF9 9C 3E          CPX     BFRSZE    TESTS IF BUFFER FULL.
00517 FBFB 27 EE          BEQ     RHDR2     SKIPS TO EXIT IF ALL IS IN.
00518 FBFD 08             INX               INC TO NEXT CHAR POSITIN.
00519 FBFE 26 EF          BNE     RBFR1     GOES BACK IF ANY LEFT.
                          END
```

interpretation & block I/O

+BLOCK
(
-->
;S
BLOCK
BUFFER
ERASE-CODE
FLUSH
CONTINUED
LOAD
UPDATE
BLK

LINE
LINELOAD
LIST

definition vocabulary

WORD
WHERE
[
]
VOCABULARY
LAST
FORTH
FLUSH
FORGET
REMEMBER
DISCARD
FIND
EDITOR
DEFINITIONS
DLIST
CURRENT
CONTEXT
CHAIN

basic word defining words

:
;
;;
; CODE
CODE
EDOC
CONSTANT
VARIABLE
SET
LIT

FDA1 – FE49 are free!

FFFA - 0100

```
        LDX #$FE4A
        STX $101
        LDA #$7C
        STA $100
```

```
EXEC1:  JSR INPCHR
        CMP A,#45
        BNE EXEC2
        BSR EDITOR
EXEC2:  CMP A,#52
        BNE EXEC3
        BSR REEDIT
EXEC3:  CMP A,#44
        BNE
```

```
EXEC    BSR HOME
        BSR CLEAR
EXEC0   BSR EXEC1
        BSR CR1
        BRA EXEC0
EXEC1   JSR INPCHR
        CMP A,#45
        BEQ EDITOR
        CMP A,#52
        BEQ REEDIT
        CMP A,#4
```

50

```
100  SWI      JMP FE4A
104  IRQ      RTI
108  NMI      RTI
10C  EXECUTE  RTS
```

```
00001                    NAM    PDS-V3N
00002                    OPT    O,NOG
00003                *
00004                *
00005                *
00006                *       PROGRAMMED BY ERIC JAMESON
00007                *
00008                *
00009                *
00010                *       COPYRIGHT 1976 SPHERE CORPORATION
00011                *       940 N.  4TH EA.; NORTH SALT LAKE, UTAH  84054
00012                *
00013                *
00014                *
00015                *       SPHERE RESERVES ALL RIGHTS FOR THE REPRODUCTION,
00016                *       DISTRIBUTION AND USE OF THE PDS SOFTWARE.
00017                *       NO COPIES MAY BE MADE OR DISTRIBUTED WITHOUT THE
00018                *       WRITTEN PERMISSION OF SPHERE CORP.
00019                *
00020                *
00021                *
00022                *
00023                * THE PROGRAM DEVELOPMENT SYSTEM (PDS V3N) IS A SET OF
00024                * PROGRAMS RDSIDING ON ERASABLE PROGRAMMABLE READ ONLY
00025                * MEMORY WHICH ALLOW EVEN THE SMALLEST USER TO USE HIS
00026                * SPHERE SYSTEM AS A COMPLETE COMPUTER SYSTEM FOR THE
00027                * DEVELOPMENT OF COMPUTER PROGRAMS.
00028                *     TOWARD THIS END, THE 4 PDS EPROMS CONTAIN A CURSOR
00029                * BASED EDITOR, A MINI-ASSEMBLER, AND THE SPHERE DEBUGGING
00030                * AID (SDA), AS WELL AS A SET OF UTILITY ROUTINES TO DO 16
00031                * BIT MULTIPLY AND DIVIDE, ASCII-TO-BINARY, AND
00032                * BINARY-TO-ASCII ROUTINES.
00033                *
00034                *
00035                *
00036                *     THE PDS-V3N PROM SET WAS WRITTEN IN ORDER TO RUN THE
00037                * NEW KEYBOARD.  CHANGES WERE MADE IN THE EDITOR AND THE
00038                * DEBUGGER.  AS THE NEW PROMS ARE A GREAT IMPROVEMENT OVER
00039                * THE V3A PROMM SET, A VERSION KNOWN AS PDS-V3D WAS MADE
00040                * WOULD RUN ON THE OLD KEYBOARDS.  THE ONLY DIFFERENCE
00041                * BETWEEN THE V3D AND THE V3N PPROMS ARE THAT THE PIA
00042                * ADDRESS IS CHANGED FROM F040 ON V3N TO F000 ON V3D.
00043                * CHANGES WERE ALSO MADE IN THE DEBUGGER AND THE
00044                * EDITOR.  THE EDITOR CHANGES WERE THAT INSERT AND DELETE
00045                * ARE NOW AT THE TOP OF THE PAGE AND THAT THERE IS A REEDIT
00046                * COMMAND IN THE EXEC (CTRL R) TO ALLOW RE-EDITING.  THE
00047                * ENTRY TO THE DEBUGGER FROM THE BREAKPOINT INSTRUCTION
00048                * HAS BEEN CHANGED SO THAT THE RETURN ADDRESS FOR THE
00049                * BREAKPOINT IS NOW CALCULATED WHEN THE BREAKPOINT IS
00050                * ENCOUNTERED AND GOES TO THE DEBUGGER.  IN ADDITION
00051                * THERE ARE 2 NEW INSTRUCTIONS:  ↑J FOR DOING A JSR TO
00052                * A ROUTINE AND ↑X TO EXIT BACK TO THE EXEC.
00053                *
```

```
00055                *                    MEMORY MAP
00056                *
00057  0000  TMP     EQU  $00
00058  0002  TMP1    EQU  $02
00059  0004  ARB     EQU  $04      16 BIT ACC. PSEUDO REG B.
00060  0004  AR3     EQU  $04      _HI BYTE OF ARB.
00061  0005  AR2     EQU  $05      _LO BYTE OF ARB.
00062  0006  ARA     EQU  $06      16 BIY ARITH PSEUDN REG A.
00063  0006  AR1     EQU  $06      _HI BYTE OF ARA.
00064  0007  AR0     EQU  $07      _LO BYTE OF ARA.
00065  0008  DIGIT   EQU  $08      BYTE USED BY ASCBIN FOR TMP.
00066  000A  OUTEND  EQU  $0A      END OF OUTPUT BUFFER TEXT.
00067  000C  BUFADR  EQU  $0C      START OF I/O BUFFER (PTR)
00068  000E  BUFEND  EQU  $0E      PTR. TO END OF I/O BUFFR.
00069  0011  OUTBUF  EQU  $11      START OF OUTPUT BUFFER.
00070  0014  SRCADR  EQU  $14      SOURCE FOR TEXT MOVES.
00071  0016  DSTADR  EQU  $16      DEST. ADDR. FOR TEXT MOVE.
00072  001A  ENDMEM  EQU  $1A      LAST ADDRES OF REAL MEMORY.
00073  001C  CSRPTR  EQU  $1C      PTR TO CURSER ON SCREEN.
00074  001E  BUFPTR  EQU  $1E      TEMP PTR USED BY OUTSTR.
00075  0020  BUFFLO  EQU  $20      PTR TO END OF LOW EDIT TXT.
00076  0022  BUFFHI  EQU  $22      PTR TO START OF HI TEXT.
00077  0024  SCNPTR  EQU  $24      PTR. TO BUFFRD TXT START.
00078  0026  SRCASM  EQU  $26      PTR TO ASSMBLR SOURCE CODE.
00079  002A  ONDVAL  EQU  $2A      HAS ASSMBLR OPERND VALUE.
00080  002C  SYMVAL  EQU  $2C      VALUE PUT IN ASSM. SYMTBL.
00081  002E  BRKSAV  EQU  $2E      TEMP SAVE FOR BRKPT DATA.
00082  0030  BRKADR  EQU  $30      ADDDRESS OF BREAKPOINT.
00083  0032  EDIT    EQU  $32      0 IF EDITOR IS NOT RUNNING.
00084  0035  IOBUFF  EQU  $35      I-O BUFFER FOR DEBUGGER.
00085  0040  PCVAL   EQU  $40      PROGRAM COUNTER FOR ASSM.
00086                *
00087  E01F  FRSTLN  EQU  $E01F    RIGHTMOST CHAR OF LINE ONE.
00088  E1E0  LASTLN  EQU  $E1E0    LEFT SIDE OF BOTTOM OF CRT.
00089  E1FF  LASTCH  EQU  $E1FF    LAST CHAR ON CRT DISPLAY.
00090  F040  KBDPIA  EQU  $F040    ADDRESS OF PIA FOR KBD/2.
00091                *KEYBOARD PIA ADDRESS FOR OLD KEYBOARD (KBD/1A) IS F000.

       E000  HOMECH  EQU  $E000
       0060  ENDCHR  EQU  $60


00093  FE71  INPCHR  EQU  $FE71    INPUTS A CHARACTER.
00094  FE64  DEBUG   EQU  $FE64    DEBUGGER ROUTINE.
00095  FF22  ASCBIN  EQU  $FF22    ASCII TO BINARY ROUTINE.
```

```
00097                    *                INITIALIZATION
00098                    *
00099                    *
00100                    *
00101                    *   THE INITIALIZATION ROUTINES SET UP THE INITIAL VALUES
00102                    * UPON SYSTEM RESET.
00103                    *
00104                    *
00105 FC00                         ORG     $FC00
00106 FC00 8E 01FF RESTRT LDS      #$1FF    SETS STACK POINTER
00107 FC03 30              TSX              MOVES STK PTR TO INDEX REG
00108 FC04 DF 26           STX     SRCASM   SETS ASSEMBLR OUTPUT PTR
00109 FC06 DF 0C           STX     BUFADR   INIT INPUT BUFFR ADDR.
00110 FC08 86 1F           LDA A   #$1F     INTLZ. KEYBOARD PIA.
00111 FC0A B7 F041         STA A   KBDPIA+1 PIA CONTROL REG. ADDRESS.
00112 FC0D CE 0FFF         LDX     #$FFF    LAST LOCATN OF MEMORY.
00113 FC10 DF 0E           STX     BUFEND   INIT END-OF-EDIT BUFFR.
00114 FC12 DF 1A           STX     ENDMEM   INIT END-OF-MEM ADDR.
00115                    *
00116                    *
00117                    *
00118                    *
00119                    *
00120                    *                COMMAND LANGUAGE
00121                    *
00122                    *    THIS EXECUTIVE ACCEPTS COMMANDS FROM THE KEYBOARD TO
00123                    * DETERMINE WHAT UTILITY IS TO BE RUN.  INVALID COMMANDS
00124                    * WILL SPACE THE CURSOR DOWN ONE LINE. DO NOT SPACE OFF THE
00125                    * BOTTOM OF THE SCREEN.
00126                    *
00127                    *
00128 FC14 8D 21    EXEC   BSR     HOME     CSRPTR IS HOMED.
00129 FC16 8D 25           BSR     CLEAR    CLEARS SCREEN
00130 FC18 8D 73    EXEC1  BSR     CR1      CRLF FOR NEW LINE.
00131 FC1A BD FE71         JSR     INPCHR   GETS & DISPLAYS CHR.
00132 FC1D 81 01           CMP A   #$01     TESTS IF ASSEMBLY COMMD. CTRL 'A'
00133 FC1F 26 03           BNE     EXEC2    SKIPS IF OTHER COMMAND.
00134 FC21 BD FDA1         JSR     ASMBLR   JUMPS TO ASSEMBL PRROGRM.
00135 FC24 81 05    EXEC2  CMP A   #$05     TESTS IF CONTRL 'E'.
00136 FC26 26 02           BNE     EXEC3    SKIPS IF NOT EDIT CMD.
00137 FC28 8D 3D           BSR     EDITOR   JUMPS TO EDIT TEXT.
00138 FC2A 81 12    EXEC3  CMP A   #$12     TESTS FOR A ^R COMMAND.  CTRL 'R'
00139 FC2C 26 02           BNE     EXEC4    SKIPS IF NOT A REEDIT COMND|00140 FC2E
 8D 3F          BSR    REEDIT   GOES TO REEDIT  TEXT.
00141 FC30 81 04    EXEC4  CMP A   #$04     TESTS FOR CONTRL 'D'.
00142 FC32 26 E4           BNE     EXEC1    SKIPS BACK FOR A NEW COMND.
00143 FC34 7E FE64         JMP     DEBUG    JUMPS TO DEBUGGER.
```

```
00145                *                    THE EDITOR
00146                *
00147                *
00148                *     THE EDITOR ALLOWS INPUT FROM THE KEYBOARD INTO A
00149                * BUFFER MEMORY.  INPUT IS DISPLAYED ON THE SCREEN.  WHEN
00150                * IT IS TYPED IN, THE SCREEN TEXT CAN THEN BE EDITED BY USE
00151                * OF THE CURSOR.  WHEN THE SCREEN IS FULL OR EDITING IS
00152                * FINISHED, THE DATA IS SCROLLED OFF THE SCREEN INTO THE
00153                * EDIT BUFFER.  WHEN TEXT IS SCROLLED OFF THE TOP OF THE
00154                * SCREEN, IT IS STORED FROM THE BUFFER ADDRESS POINTER
00155                * (BUFADR) TO THE LOW BUFFER POINTER (BUFFLO).  BUFFLO
00156                * POINTS TO THE END OF TEXT + ONE (I.E.  IT POINTS TO THE
00157                * FIRST UNUSED BYTE).  WHEN IT IS SCROLLED OFF THE BOTTOM
00158                * OF THE SCREEN, IT IS STORED IN THE TOP OF THE EDIT BUFFER.
00159                * THE TEXT GOES FROM THE HIGH BUFFER POINTER (BUFFHI) TO
00160                * THE END OF BUFFER POINTER (BUFEND).  BUFFHI POINTS TO
00161                * THE END OF TEXT - ONE (I.E.  IT POINTS TO THE LAST UNUSED
00162                * BYTE IN THE BUFFER).  WHEN THE TEXT IS SCROLLED UP
00163                * OFF THE TOP OF THE SCREEN, TEXT IS TAKEN FROM THE HIGH
00164                * AREA OF THE EDIT BUFFER AND DISPLAYED ON THE LAST LINE OF
00165                * THE SCREEN.  WHEN TEXT IS SCROLLED DOWN OFF THE BOTTOM,
00166                * TEXT, IF ANY EXISTS, IS MOVED FROM THE LOW EDIT BUFFER
00167                * AREA TO THE TOP LINE OF THE SCREEN.
00168                *
00169                *
00170                *
00171                *
00172                *
00173                *
00174                *
00175                *
00176                *
00177                *
00178                *                    POINTERS USED
00179                *
00180                * CSRPTR   POSITION OF CHARACTERS INSERTED ON THE SCREEN.
00181                * SCNPTR   POSITION OF START OF EDITED TEXT ON SCREEN.
00182                * BUFADR   START OF TEXT BUFFER IN MAIN MEMORY.
00183                * BUFEND   END OF TEXT BUFFER IN MAIN MEMORY.
00184                * BUFFLO   END OF TEXT SCROLLED OFF TOP OF SCREEN.
00185                * BUFFHI   START OF TEXT SCROLED OFF BOTTOM OF SCREEN.
00186                * BUFLEN   NOT CURRENTLY USED.
00187                *
00188                *
00189                *
00190                *
00191                *
00192                *                    EDITOR COMMANDS
00193                *
00194          'Q    * "UP ARROW"  MOVES CURSOR UP ONE LINE; CSRPTR GETS
00195                *  CSRPTR-32; CALL NDRFLO.
00196                *
00197          'S    * "DOWN ARROW"  MOVES CURSOR DOWN ONE LINE; CSRPTR GETS
00198                *  CSRPTR+32; CALL OVRFLO.
00199                *
00200          'R    * "RIGHT ARROW"  MOVES CURSOR ONE POSITION RIGHT; CSRPTR
00201                *  GETS CSRPTR+1; CALL OVRFLO.
```

```
00203          ^T  * "LEFT ARROW"  MOVES CURSOR ONE POSITION LEFT; CSRPTR
00204              *  GETS CSRPTR-1; CALL NDRFLO.
00205              *
00206          ^   * "CONTROL & LEFT ARROW (ON KEYBOARD)"  LEFT JUSTIFY CURSOR;
00207          —   *  CSRPTR GETS CSRPTR TRUNCATED; CALL NDRFLO FOR SCNLOC CHK.
00208              *
00209              * "PUTCHR"  OUTPUTS CHARACTER; CSRPTR GETS
00210              *  CSRPTR+1; GOES TO OVRFLO.
00211              *
00212          ESC * "ENDCHR"  TERMINATION CHAR; CLEAR EDIT FLAG;
00213              *  EXIT THE EDITOR.
00214              *
00215      ^@ - ^L * "HOME"  HOMES CURSOR POINTER; CSRPTR GETS E000; NDRFLO.
00216              *
00217      ^U - ^^ * "CLEAR"  CSRPTR TO END OF THE SCREEN GETS SPACES.
00218              *
00219              * "CTRL I"  INSERT A LINE AT THE ~~LAST~~ FIRST LINE ON THE SCREEN;
00220              *  CALL OVR1 (SCROLLS UP ONE LINE); CSRPTR GETS ~~E1E0~~.
00221              *
00222              * "CTRL D"  DELETE ~~LAST~~ FIRST LINE; SCROLL ~~DOWN~~ UP (UNDR2);
00223              *  CSRPTR GETS ~~E1E0~~.
00224              *                                  E000   1st line - scroll up
00225              *
00226              *
00227              *
00228              * OVERFLOW CHECKS IF SCROLL UP IS NEEDED; IF IT IS, IT
00229              *  SCROLLS UP AND MOVES DATA TO & FROM THE BUFFEERS.
00230              *
00231              * OVRFLO: IF CSRPTR < E200 THEN RETURN; IF EDIT IS ON THEN
00232              * OVR1:   BUFFLO+ GETS SCNPTR TO 'C.R.';
00233              * DSTADR GETS CSRPTR GETS E1E0 (LAST LINE ON SCREEN);
00234              * IF EDIT IS ON AND BUFFHI < BUFEND THEN MOVE THE TEXT
00235              * (THE STRING FROM BUFFHI TO 'C.R.') TO THE LAST LINE.
00236              *
00237              *
00238              *
00239              *
00240              * UNDERFLOW CHECKS IF SCROLL DOWN IS NEEDED AND MOVES
00241              *  DATA TO AND FROM THE BUFFERS.  CURSOR HAD BEEN
00242              *  MOVED OFF THE TOP OF THE SCREEN AND IS NOW PUT AT THE
00243              *  HOME POSITION ON THE SCREEN.
00244              *
00245              * NDRFLO:  IF CSRPTR > DFFF THEN RETURN (GO TO OVRFLO);
00246              *  IF EDIT FLAG IS ON THEN MOVE LAST LINE TO BUFFHI
00247              *   ON DOWN; SCRLDN; CSRPTR GETS E000; MOVE LINE FROM
00248              *   BUFFLO TO FIRST LINE ON THE CRT.
00249              *
00250              *
00251              *
00252              *
00253              *  NOTE:  DON'T SCROLL OFF SCREEN IN EXEC UNTIL AFTER
00254              *    THE EDITOR HAS BEEN RUN.
00255              *
00256              *  NOTE:  EVERY LINE MUST HAVE A 'C.R.  ON IT.
```

```
00258 FC37 CE E000 HOME   LDX    #$E000   LOADS HOME POSITION.
00259 FC3A DF 1C          STX    CSRPTR   STORES HOME IN CURSOR PTR.
00260 FC3C 39             RTS             RETURNS TO CALLER.
00261                *
00262                *
00263 FC3D C6 60    CLEAR LDA B  #$60      LOADS BLANK (C.R.).
00264 FC3F CE E200        LDX    #LASTCH+1 LOADS END-OF-SCREEN PTR.
00265 FC42 09       CLEAR1 DEX             DECREMENTS BLANKING PTR.
00266 FC43 E7 00          STA B  0,X       BLANKS LOCATION.
00267 FC45 9C 1C          CPX    CSRPTR    TESTS IF DONE.
00268 FC47 26 F9          BNE    CLEAR1    BRANCHES BACK IF NOT DONE.
00269 FC49 39             RTS             RETURNS.
00270                *
00271                *
00272                *
00273                *       GETCHR INPUTS A CHARACTER INTO ACC A WITHOUT
00274                *       MOVING THE CURSOR, AND BLINKS THE CURSOR.
00275                *
00276 FC4A DE 1C    GETCHR LDX    CSRPTR   LOADS CRT CURSOR POSITION.
00277 FC4C 63 00          COM    0,X      COMPLIMENT (FLASH POSITION).
00278 FC4E CE 26F0        LDX    #9968    LOADS BLINK COUNT VALUE.
00279 FC51 09       GET1  DEX             COUNT GETS COUNT-1.
00280 FC52 27 F6          BEQ    GETCHR   RESETS CTR WHEN TIMED OUT.
00281 FC54 86 40          LDA A  #$40     LOADS MASK FOR CA2 FLAG.
00282 FC56 B5 F041        BIT A  KBDPIA+1 TESTS IF A CHAR. TYPED IN.
00283 FC59 27 F6          BEQ    GET1     BRANCH IF CHAR NOT ENTERED.
00284 FC5B DE 1C          LDX    CSRPTR   LOADS CURSOR POSITION.
00285 FC5D A6 00          LDA A  0,X      TESTS IF BLINKMD (SOLID).
00286 FC5F 2A 02          BPL    GET2     SKIPS IF NOT BLINKED.
00287 FC61 63 00          COM    0,X      CLEARS THE CHARACTER.
00288 FC63 B6 F040  GET2  LDA A  KBDPIA   LOADS A WITH KEYBRD CHAR.
00289 FC66 39             RTS             RETURNS TO CALLER.
00290                *
00291                *
00292                *
00293                *
00294                *       EDITOR IS THE MAIN ENTRY POINT FOR EDITING.
00295                *
00296 FC67 DE 0C    EDITOR LDX    BUFADR   BUFFLO GETS THE
00297 FC69 DF 20          STX    BUFFLO   VALUE OF BUFADR.
00298 FC6B DE 0E          LDX    BUFEND   BUFFHI GETS THE
00299 FC6D DF 22          STX    BUFFHI   VALUE OF BUFEND.
00300 FC6F 8D C6    REEDIT BSR    HOME     ENTRY POINT FOR
00301 FC71 8D CA          BSR    CLEAR    RE-EDITING TEXT.
00302 FC73 97 32    EDITRD STA A  EDIT     TURNS ON EDIT MODE.
00303 FC75 DE 1C    EDITIN LDX    CSRPTR   SETS SCNPTR TO CSRPTR.
00304 FC77 DF 24          STX    SCNPTR
00305                *
00306                *
00307 FC79 8D CF    EDREAD BSR    GETCHR   X GETS CSRPTR & A GETS CHR.
00308 FC7B 81 1B    ENDCHR CMP A  #$1B     TESTS FOR AN "ESC" CHAR.
00309 FC7D 26 04          BNE    ED1      SKIPS IF NOT EDIT END
00310 FC7F 7F 0032        CLR    EDIT     TURNS OFF EDIT FLAG.
00311 FC82 39             RTS             EXITS THE EDITOR.
00312 FC83 8D 0A    ED1   BSR    INSERT   EDITS CHARACTER.
00313 FC85 20 F2          BRA    EDREAD   GOES FOR NEXT CHARACTER.
```

```
00315                    *
00316                    *
00317                    *       FOLLOWING IS THE MAIN EDITOR EXECUTION LOOP.
00318                    *
00319                    *
00320                    *
00321                    *
00322                    *
00323 FC87 81 0D   CR     CMP A   #$0D      TESTS FOR CARRIAGE RETURN.
00324 FC89 2D AC          BLT     HOME      SKIPS IF HOME CURSR COMND.
00325 FC8B 2E 12          BGT     RTCSR     GOES TO NEXT COMND TEST.
00326 FC8D 86 60   CR1    LDA A   #$60      LOADS INTERNAL C. R.  VALUE.
00327                    *
00328                    *
00329                    *
00330                    *
00331 FC8F 81 09   INSERT CMP A   #$09      TESTS FOR A CONTROL 'I'.
00332 FC91 2D 16          BLT     DELETE    SKIPS TO DELETE COMMD.
00333 FC93 2E F2          BGT     CR        SKIPS FOR NEXT TEST.
00334 FC95 D6 32   INSRT1 LDA B   EDIT      TESTS IF EDITOR IS ON.
00335 FC97 27 03          BEQ     INSRT2    SKIP TO EXIT IF EDITOR OFF.
00336 FC99 BD FD46        JSR     MOVE2     MOVES LAST LINE TO BUFFHI.
00337 FC9C 7E FD74 INSRT2 JMP     SCRLDN    MOVES ALL LINES DOWN ONE.
00338                    *
00339                    *
00340                    *
00341 FC9F 81 12   RTCSR  CMP A   #$12      TESTS FOR RIGHT ARROW.
00342 FCA1 2D 28          BLT     SUB32     SKIPS IF AN "UP ARROW".
00343 FCA3 2E 08          BGT     LFTCSR    SKIPS IF A "LEFT ARROW".
00344 FCA5 DE 1C   RTARRO LDX     CSRPTR    LOADS CURSOR POINTER.
00345 FCA7 20 1F          BRA     PUTCH1    STORES & INCREMENTS CSR.
00346                    *
00347                    *
00348                    *
00349 FCA9 8D 67   DELETE BSR     OVR1A     SCROLLS UP ONE LINE.
00350 FCAB 20 40          BRA     OVR3      MOVES NEW LAST SCREEN LINE.
00351                    *
00352                    *
00353                    *
00354 FCAD 81 14   LFTCSR CMP A   #$14      TESTS IF "LEFT ARROW".
00355 FCAF 2D 24          BLT     ADD32     SKIPS IF "DOWN ARROW".
00356 FCB1 2E 03          BGT     CLER      SKIPS FOR NEXT TEST.
00357 FCB3 09             DEX               SUB. 1 FROM CSRPTR.
00358 FCB4 20 25          BRA     ADD2      STORES CURSOR POINTER.
00359                    *
00360                    *
00361                    *
00362 FCB6 81 1F   CLER   CMP A   #$1F      TESTS FOR CTRL BACK ARROW.
00363 FCB8 2D 83          BLT     CLEAR     GOES TO CLEAR SCREEN.
00364 FCBA 27 41          BEQ     LFTJST    MOVES CSR TO LEFT OF SCREEN.
00365                    *
00366                    *
00367                    * ALL OTHER CHARACTERS FALL THRU TO PUTCHR.
00368                    *
```

```
00370                    *       PUTCHR DISPLAYS A CHARACTER ON THE CRT DISPLAY AND
00371                    *         INCREMENTS THE CURSOR POINTER AS WELL AS CHECKING
00372                    *         AND HANDLING CARRIAGE RETURNS.
00373                    *
00374                    *
00375                    *
00376 FCBC DE 1C   PUTCHR LDX     CSRPTR    LOADS OLD CSRPTR.
00377 FCBE 81 0D         CMP A    #$0D      TESTS FOR EXTERNAL C. R.
00378 FCC0 27 04         BEQ      CRLF1     SKIPS TO DO A C. R.  L. F.
00379 FCC2 A7 00         STA A    0,X       DISPLAYS CHAR ON SCREEN.
00380 FCC4 81 60         CMP A    #$60      TESTS FOR INTERNAL C. R.
00381 FCC6 27 4C   CRLF1  BEQ      CRLF      SKIPS FOR CR.  LF.
00382 FCC8 08      PUTCH1 INX                INCREMENTS CSRPTR.
00383 FCC9 20 10         BRA      ADD2      TESTS FOR OVRFLO & UNDRFLO.



00385 FCCB DE 1C   SUB32  LDX     CSRPTR    LOADS CURRENT CRSR POSITION.
00386 FCCD C6 20         LDA B    #32       LOADS LOOP COUNT.
00387 FCCF 09      SUB32A DEX                DECREMENTS CSRPTR.
00388 FCD0 5A            DEC B    -         DECREMENTS LOOP COUNTR.
00389 FCD1 26 FC         BNE      SUB32A    SKIPS BACK IF NOT DONE.
00390 FCD3 20 06         BRA      ADD2      SKIPS TO CHECK UNDRFLO.



00392 FCD5 C6 20   ADD32  LDA B    #32       LOADS LOOP COUNTER.
00393 FCD7 08      ADD32A INX                INCRE.  CSRPTR IN INDEX.
00394 FCD8 5A            DEC B    -         DCEREMENTS LOOP COUNTER.
00395 FCD9 26 FC         BNE      ADD32A    SKIPS BACK IF NOT DONE.
00396 FCDB DF 1C   ADD2   STX      CSRPTR    SAVES CSRPTR.



00398                    * NDRFLO (UNDERFLOW) CHECKS FOR THE CURSOR GOING OFF THE
00399                    *  TOP OF THE SCREEN.  THE INDEX REG. CONTAINS THE CURSOR
00400                    *  POINTER WHEN THE ROUTINE IS ENTERED.
00401                    *
00402                    *
00403 FCDD 8C E000 NDRFLO CPX     #$E000    TESTS IF CSRPTR >= DFFF.
00404 FCE0 2C 04         BGE      OVRFLO    SKIPS IF CSRPTR GREATER.
00405 FCE2 8D B1         BSR      INSRT1    SCROLLS DOWN & MOVES LINE.
00406 FCE4 8D 32         BSR      MOVE3     MOVES BUFFLO TO TOP OF CRT.
```

```
00408                    *
00409                    *
00410                    *     OVERFLOW CHECKS FOR SCROLLING UP (CURSOR IS OFF
00411                    * THE BOTTON OF THE SCREEN); INDEX CONTAINS THE CURSOR
00412                    * POINTER UPON ENTRY.
00413                    *
00414                    *
00415 FCE6 8C E200 OVRFLO CPX     #$E200   TESTS AND EXITS IF
00416 FCE9 2B 18          BMI     OVREXT   CURSOR ON SCREEN.
00417 FCEB 8D 17          BSR     OVR1     DOES OVR1 CHECKING.
00418 FCED D6 32    OVR3  LDA B   EDIT     TESTS IF EDIT IS ON.
00419 FCEF 27 12          BEQ     OVREXT   EXITS IF IT IS OFF.
00420 FCF1 DE 22          LDX     BUFFHI   LOADS HI TEXT POINTR.
00421 FCF3 9C 0E          CPX     BUFEND   TESTS IF PTRS NOT EQU.
00422 FCF5 27 0C          BEQ     OVREXT   EXITS IF NO TEXT.
00423 FCF7 8D 3C          BSR     MOVE1A   MOVES CHRS TO LAST LINE.
00424 FCF9 DE 14          LDX     SRCADR   RESETS NEW BUFFHI
00425 FCFB DF 22          STX     BUFFHI   LOCATION.
00426                    *
00427                    *
00428                    *
00429                    *
00430                    *     FOLLOWING ROUTINE MOVES THE CURSER TO THE LEFT.
00431                    *
00432 FCFD D6 1D    LFTJST LDA B  CSRPTR+1 LOADS LOW BYTE OF PTR.
00433 FCFF C4 E0          AND B   #$E0     TRUNCATES TO LEFT OF LINE.
00434 FD01 D7 1D          STA B   CSRPTR+1 SAVES L. J. ED PTR.
00435 FD03 39       OVREXT RTS             RETURNS TO EDITOR.
00436                    *
00437                    *
00438                    *
00439                    *
00440                    *     OVR1 DOES ACTUAL SCROLLING UP.
00441                    *
00442 FD04 D6 32    OVR1  LDA B   EDIT     TESTS IF EDIT IS ON.
00443 FD06 27 0A          BEQ     OVR1A    SKIPS IF EDIT OFF.
00444 FD08 DE 20          LDX     BUFFLO   LOADS TEXT PTR LOW.
00445 FD0A DF 16          STX     DSTADR   DESTINATION OF TEXT MOVE.
00446 FD0C DE 24          LDX     SCNPTR   SOURCE FOR MOVE.
00447 FD0E 8D 26          BSR     MOVE1    MOVES LIN1 TO BUFFFLO.
00448 FD10 DF 20          STX     BUFFLO   SAVES NEW BUFFLO PTR.
00449 FD12 20 4B    OVR1A BRA     SCRLUP   SCROLLS SCREEN UP 1.




00451                    *     FOLLOWING ROUTINE MOVES THE CURSOR.
00452                    *
00453 FD14 8D BF    CRLF  BSR     ADD32    LINE FEED.
00454 FD16 20 E5          BRA     LFTJST   CARRIAGE RETURN.
```

```
00456                    *     MOVE INSTRUCTIONS MOVE FROM ONE BUFFER AREA TO
00457                    * ANOTHER BUFFER AREA.
00458                    *
00459                    *
00460                    *
00461                    *     MOVE3 CALCULATES THE SOURCE ADDRESS OF THE DATA IN
00462                    * BUFFLO (IF IT EXISTS) FOR MOVING TO THE FIRST LINE ON
00463                    * THE CRT.  MOVE 1 IS THEN ENTERED TO DO THE MOVING.
00464                    *
00465                    *
00466 FD18 DE 24    MOVE3  LDX     SCNPTR   CSRPTR GETS E000 (HOME).
00467 FD1A DF 16           STX     DSTADR   SETS MOVE ADDRESS.
00468 FD1C DE 20           LDX     BUFFLO   LOADS LO BUFFR ADDR.
00469 FD1E 9C 0C           CPX     BUFADR   TESTS IF STRING EXISTS.
00470 FD20 27 23           BEQ     MOVEXT   EXITS IF EMPTY.
00471 FD22 09              DEX              MOVES BACK FROM BLANK.
00472 FD23 9C 0C    MV31   CPX     BUFADR   TESTS IF SRCADR = BUFFADR.
00473 FD25 27 08           BEQ     MV32     MOVES IF START OF LINE.
00474 FD27 09              DEX              NEXT LOWER CHAR.
00475 FD28 E6 00           LDA B   0,X      GETS SOURCE CHAR FOR TEST.
00476 FD2A C1 60           CMP B   #$60     TESTS FOR "C. R. ".
00477 FD2C 26 F5           BNE     MV31     SKIPS BACK UNTIL "C. R. ".
00478 FD2E 08              INX              POINTS BACK TO FIRST CHAR.
00479 FD2F DF 20    MV32   STX     BUFFLO   SAVES LO ADDRESS.
00480 FD31 20 03           BRA     MOVE1    MOVES DATA.


00482                    *          MOVE1
00483                    *
00484                    *     MOVE1 MOVES A SET OF CHARACTERS FROM EITHER THE TOP
00485                    * LINE OF THE SCREEN TO BUFFLO OR FROM BUFFHI TO THE
00486                    * BOTTOM LINE OF THE SCREEN.  THE SOURCE ADDRESS IS PASSED
00487                    * IN THE INDEX REG., THE DESTINATION ADDRESS IN DSTADR,
00488                    * AND THE MOVE IS TERMINATED BY A "C. R. " IN THE LINE OF
00489                    * TEXT BEING MOVED.
00490                    *
00491                    *
00492                    *
00493                    *
00494 FD33 DE 14    MOVE   LDX     SRCADR   LOADS SOURCE ADDRESS INTO X. E
00495 FD35 08       MOVE1A INX              POINTS TO NEXT SOURCE CHAR.
00496 FD36 E6 00    MOVE1  LDA B   0,X      LOADS SOURCE CHARACTER.
00497 FD38 DF 14           STX     SRCADR   SAVES THE SOURCE POINTER.
00498 FD3A DE 16           LDX     DSTADR   LOADS DESTINATION ADDRESS.
00499 FD3C E7 00           STA B   0,X      STORES CHAR. IN DESTINATION.
00500 FD3E 08              INX              NEXT DESTINATION ADDRESS.
00501 FD3F DF 16           STX     DSTADR   SAVES DESTINATION PTR.
00502 FD41 C1 60           CMP B   #$60     TESTS IF MOVE FINSHED (CR).
00503 FD43 26 EE           BNE     MOVE     SKIPS BACK IF NOT DONE.
00504 FD45 39       MOVEXT RTS              RETURNS TO CALLER.
```

```
00506                    *          MOVE2 SUBROUTINE
00507                    *
00508                    *     THE MOVE2 SUBROUTINE MOVES THE LAST LINE ON THE
00509                    * SCREEN TO THE HIGH AREA OF THE BUFFER (BUFFHI) DURING
00510                    * SCROLLING.  THE TEXT IS TEMPORARILY STORED ON THE STACK
00511                    * DURING THE MOVE.  THE MOVE IS TERMINATED BY A "C. R. ".
00512                    * THE TEXT IS STORED AT BUFFHI ON DOWN.
00513                    *
00514                    *
00515                    *
00516                    *
00517 FD46 CE E1E0 MOVE2  LDX    #LASTLN   X GETS ADDR OF LAST LINE.
00518 FD49 5F             CLR B  -         SETS TERMINATION FOR
00519 FD4A 37        MV21 PSH B  STACK     POPPING.
00520 FD4B E6 00          LDA B  0,X       LOADS SOURCE CHAR.
00521 FD4D 08             INX              POINTS TO NEXT CHAR.
00522 FD4E C1 60          CMP B  #$60      TESTS IF LINE TO "C. R. "
00523 FD50 26 F8          BNE    MV21      MOVED TO STACK.
00524 FD52 DE 22     MV22 LDX    BUFFHI    INIT. DESTINATION.
00525 FD54 E7 00     MV23 STA B  0,X       STORES CHAR.
00526 FD56 09             DEX              POINTS TO NEXT LOCATION.
00527 FD57 DF 22          STX    BUFFHI    UPDATES BUFFER PTR.
00528 FD59 33             PUL B  -         GETS NEXT CHAR.
00529 FD5A C1 00          CMP B  #00       TESTS IF ALL CHRS STORED.
00530 FD5C 26 F6          BNE    MV23      SKIPS BACK IF NOT STORED.
00531 FD5E 39        MOVEX RTS              RETURNS TO CALLER.


00533                    *          SCROLLUP MOVES ALL LINES UP 1, & CLEARS LAST LINE.
00534                    *                   #HOMECH
00535 FD5F CE E000 SCRLUP LDX    ##E000    SETS CRT HOME POSITION.
00536 FD62 E6 20   SCRP1  LDA B  $20,X     GETS CHAR FROM NEXT LINE.
00537 FD64 E7 00          STA B  00,X      STORES CHAR ON PREV. LINE.
00538 FD66 08             INX              POINTS TO NEXT LINE.
00539 FD67 8C E1E0        CPX    #LASTLN   TESTS IF MOVE DONE.
00540 FD6A 26 F6          BNE    SCRP1     GOES BACK IF NOT DONE.
00541 FD6C DF 1C          STX    CSRPTR    SETS CSRPTR TO LAST LINE.
00542 FD6E DF 16          STX    DSTADR    INIT DEST FOR NEXT MOVE.
00543 FD70 BD FC3D        JSR    CLEAR     CLEARS LAST LINE.
00544 FD73 39             RTS              EXITS.
```

LDX —
PSH A
LOOP LDA B E1E0
CMP B 0,X
BEQ
INC B
INX
BRA LOOP

```
00546                    *      SCRLDOWN MOVES ALL LINES DOWN ONE AND
00547                    * CLEARS THE TOP LINE ON THE SCREEN.
00548                    *
00549 FD74 CE E1DF SCRLDN LDX    #LASTLN-1  INITIALIZES THE POINTER.
00550 FD77 E6 00   SCRD1  LDA B  0,X        LOADS DATA TO BE MOVED.
00551 FD79 E7 20          STA B  $20,X      MOVES DATA DOWN ONE LINE.
00552 FD7B DF 1C          STX    CSRPTR     SAVES CURSOR.
00553 FD7D 09             DEX               POINTS TO NEXT BYTE.
00554 FD7E 8C DFFF        CPX    #$DFFF     TESTS IF MOVE FINISHED.
00555 FD81 26 F4          BNE    SCRD1      SKIPS BACK IF NOT DONE.
00556 FD83 C6 60          LDA B  #$60       LOADS BLANK TO CLEAR LINE.
00557 FD85 08      SCRD2  INX                POINTS TO NEXT CHARACTER.
00558 FD86 E7 00          STA B  0,X        CLEARS BYTE ON LINE 1.
00559 FD88 8C E01F        CPX    #FRSTLN    TESTS IF LINE 1 CLEARED.
00560 FD8B 26 F8          BNE    SCRD2      SKIPS BACK IF NOT CLEARED.
00561 FD8D 39             RTS               RETURNS.


00563                    *      OUTSTRING PRINTS OUT THE STRING BETWEEN THE
00564                    *      OUTBUF POINTER AND THE BUFEND POINTER.
00565                    *
00566 FD8E DE 11   OUTSTR LDX    OUTBUF     BUFPTR GETS START OF TEXT.
00567 FD90 A6 00   OUT1   LDA A  0,X        LOADS CHAR TO BE PUT OUT.
00568 FD92 DF 1E          STX    BUFPTR     SAVES SOURCE POINTER.
00569 FD94 BD FCBC        JSR    PUTCHR     PRINTS CHARACTER.
00570 FD97 DE 1E          LDX    BUFPTR     RESTORES POINTER.
00571 FD99 9C 0A          CPX    OUTEND     TESTS FOR END-OF-TEXT.
00572 FD9B 27 03          BEQ    OUT2       EXITS IF END OF TEXT.
00573 FD9D 08             INX               INCRE. PTR TO NEXT CHAR.
00574 FD9E 20 F0          BRA    OUT1       GOES BACK FOR NEXT CHAR.
00575 FDA0 39      OUT2   RTS               EXITS ROUTINE.
00576                    *
00577                    *      END OF EDITOR PROGRAM.
00578                    *
```

```
00580                  *          THE MINI-ASSEMBLER
00581                  *
00582                  *
00583                  *     THE MINI-ASSEMBLER IS A FIXED-FIELD ONE INSTRUCTION
00584                  * PER LINE 2 PASS ASSEMBLER.  THE MINI-ASSSEMBLER FORMAT
00585                  * IS DESCRIBED ON PAGES 9-2 AND 9-3 OF THE SPHERE
00586                  * OPERATORS REFERENCE MANUAL.
00587                  *     THE TWO PASSES ARE REQUIRED TO FORM THE LABEL
00588                  * ADDRESSES.  THE SECOND PASS EQUATES THE ADDRESS FOR
00589                  * LABELS REFRENCED BEFORE THEY ARE DEFINED IN THE PROGRAM.
00590                  *
00591                  *
00592                  * ON ENTRY:
00593                  * SRCASM = ADDRESS OF SOURCE TEXT TO BE ASSEMBLED.
00594                  * BUFFLO = ADDRESSED OF OBJECT CODE PRODUCED.
00595                  *
00596                  * ON EXIT:
00597                  * PCVAL (PROGRAM COUNTER VALUE) = LAST LOCATION OF
00598                  * THE ASSEMBLED OBJECT PROGRAM.
00599                  *
00600                  *
00601                  *
00602                  *
00603                  * ALGORITHM:
00604                  *
00605                  *ASMBLR:  SET PASS COUNT TO ZERO; SET PCVAL TO DSTASM;
00606                  *ASM1A:  OPERAND VALUE FORMED IN "ONDVAL":
00607                  * A GETS CHAR IN X6 (OPERAND TYPE); X GETS X+7;
00608                  * IF CHAR X6 IS A "@" THEN ONDVAL GETS VALUE FROM SYMBOL
00609                  *   TABLE ELSE ONDVAL GETS VALUE FROM ASCBIN CONVERSION;
00610                  *SYMBL:  EQUATES SYMBOL (PC VALUE IS THE " " SYMBOL
00611                  *[LABEL]) TO A LABEL VALUE:
00612                  * SYMVAL GETS PCVAL;
00613                  * IF X(1) IS AN "=" THEN SYMVAL GETS ONDVAL;
00614                  * IF X(1) IS NOT A "=" OR A SPACE THEN IF SECOND PASS THEN
00615                  *   EXIT ELSE START SECOND PASS;
00616                  * LABEL ENTRY IN SYMBOL TABLE GETS SYMVAL;
00617                  *LDOP:  PUT OPERATION CODE INTO THE OBJECT CODE:
00618                  * CONVERT X(2)-X(3) INTO BINARY;
00619                  * SAVE PCVAL;
00620                  * P.C.  GETS P.C. +1;
00621                  *OPRND:  FORM OPERAND IN OBJECT CODE:
00622                  * FORM ONDVAL INTO PROPER SIZE BASED ON CODE IN X(6);
00623                  * STORE NEW OPERAND VALUE IN MEMORY;
00624                  * P.C.  GETS P.C. +1 OR 2;
00625                  * GET NEXT LINE OF SOURCE;
00626                  * GO TO ASM1A;
00627                  *
00628                  *
```

```
00630 FDA1 7F 0004 ASMBLR CLR     AR3       INIT. PASS CTR TO FRST PASS. )
00631 FDA4 DE 20   ASM1   LDX     BUFFLO    SETS PC CNTR TO START OF
00632 FDA6 DF 40          STX     PCVAL     OBJECT CODE.
00633 FDA8 DE 26          LDX     SRCASM    LOADS ADDR FOR FIRST LINE.
00634 FDAA DF 02   ASM1A  STX     TMP1      SAVES ADDR OF CURRENT LINE.
00635 FDAC A6 08          LDA A   8,X       LOADS SYMBOL (LABEL).
00636 FDAE E6 07          LDA B   7,X       LOADS OPERAND TYPE CODE.
00637 FDB0 C1 40          CMP B   #$'@      IF @, LOADS DATA IN SYMBOL
00638 FDB2 27 6B          BEQ     INDADR    ADDRESS, GOES TO SYMBL.
00639 FDB4 08             INX               SETS INDEX TO START OF
00640 FDB5 08             INX               OPERAND NUMBER.
00641 FDB6 08             INX
00642 FDB7 08             INX
00643 FDB8 08             INX
00644 FDB9 08             INX
00645 FDBA 08             INX
00646 FDBB BD FF22        JSR     ASCBIN    CONVRTS # TO BINARY IN B-A.
00647 FDBE D7 2A   ASM1B  STA B   ONDVAL    STORES OPERAND VALUE IN
00648 FDC0 97 2B          STA A   ONDVAL+1  ONDVAL.
00649              *
00650              *
00651              *
00652              * FOLLOWING FORMS THE VALUE FOR THE LABEL.
00653              *
00654 FDC2 DE 02   SYMBL  LDX     TMP1      LOADS ORIG LINE PTR INTO X.
00655 FDC4 A6 00          LDA A   0,X       LOADS SYMBOL (LABEL).
00656 FDC6 E6 01          LDA B   1,X       LOADS LABEL CONTROL CHAR.
00657 FDC8 DE 40          LDX     PCVAL     LABEL VALUE GETS PCVAL.
00658 FDCA DF 2C          STX     SYMVAL    .
00659 FDCC C1 3D          CMP B   #$'=      TESTS IF LABLE IS EQUATED.
00660 FDCE 26 06          BNE     ASM2      SKIPS IF NOT EQUATED.
00661 FDD0 DE 2A          LDX     ONDVAL    LABEL VALUE (SYMVAL) GETS
00662 FDD2 DF 2C          STX     SYMVAL    THE OPERAND VALUE.
00663 FDD4 20 0E          BRA     ASM3      CONTINUES EVALUATION.
00664 FDD6 C1 20   ASM2   CMP B   #$'       TESTS FOR END-OF-PROGRAM.
00665 FDD8 27 0A          BEQ     ASM3      SKIPS IF SPACE (NOT END).
00666 FDDA 7D 0004        TST     AR3       TESTS IF SECOND PASS.
00667 FDDD 27 01          BEQ     ASM2A     EXITS IF SECOND PASS.
00668 FDDF 39             RTS               EXITS THE ASSEMBLER.
00669 FDE0 D7 04   ASM2A  STA B   AR3       SETS CTR TO SECOND PASS.
00670 FDE2 20 C0          BRA     ASM1      GOES BACK FOR SECOND PASS.
00671              *
00672              *
00673              *
00674              * FOLLOWING PUTS THE LABEL VALUE IN THE SYMBOL TABLE.
00675              *
00676 FDE4 8D 41   ASM3   BSR     SYMPTR    X GETS SYMBL TABL ENTRY ADR. E
00677 FDE6 96 2C          LDA A   SYMVAL    STORES THE LABEL
00678 FDE8 A7 00          STA A   0,X       ADDRESS (SYMVAL) INTO THE
00679 FDEA 96 2D          LDA A   SYMVAL+1  SYMBOL TABEL.
00680 FDEC A7 01          STA A   1,X
```

```
00682                    * FOLLOWING FORMS THE OPERATION CODE.
00683                    *
00684 FDEE DE 02   LDOP   LDX     TMP1      LOADS ORIG LINE POINTER.
00685 FDF0 08             INX               SETS X TO POINT TO
00686 FDF1 08             INX               THE OP CODE CHARS.
00687 FDF2 08             INX
00688 FDF3 A6 00          LDA A   0,X       GETS OP CODE CHAR INTO A.
00689 FDF5 81 20          CMP A   #$'       TESTS IF OP CODE EXISTS.
00690 FDF7 27 0A          BEQ     OPRND     SKIPS IF NONEXISTANT.
00691 FDF9 BD FF22        JSR     ASCBIN    CONVRTS OP CODE TO BINARY.
00692 FDFC DE 40          LDX     PCVAL     LOADS POINTR TO OBJECT CODE.
00693 FDFE A7 00          STA A   0,X       STORS OP INTO OBJECT CODE.
00694 FE00 08             INX               SETS TO NEXT OBJ CODE LOCTN.
00695 FE01 DF 40          STX     PCVAL     SAVES P. C.  POINTER.
00696                    *
00697                    *
00698                    *
00699                    * FOLLOWING STORES INTO THE OBJECT CODE THE SIZED OPERAND.
00700                    *
00701 FE03 DE 02   OPRND  LDX     TMP1      LOADS SOURCE LINE POINTER.
00702 FE05 A6 06          LDA A   6,X       LOADS OPERAND SIZE CHAR.
00703 FE07 DE 40          LDX     PCVAL     LOADS X WITH OBJ CODE PTR.
00704 FE09 81 45          CMP A   #$'E      TESTS LENGTH TYPE.
00705 FE0B 2E 31          BGT     RELTIV    SKIPS IF AN "R" OPERAND.
00706 FE0D 27 21          BEQ     EXTEND    SKIPS IF AN "E" SIZE OPRND.
00707 FE0F 81 44          CMP A   #$'D      TESTS IF SIZE CHR EXISTS.
00708 FE11 27 22          BEQ     DIRECT    SKIPS IF "D"COMND EXISTS.
00709                    *
00710                    *
00711                    *
00712                    * FOLLOWING GETS THE NEXT LINE.
00713                    *
00714 FE13 DE 02   ASM4   LDX     TMP1      LOADS START OF LINE IN
00715 FE15 08      ASM4A  INX               ORDER TO FIND NEXT LINE.
00716 FE16 A6 00          LDA A   0,X       LOADS CHAR FROM SORCE LINE.
00717 FE18 81 60          CMP A   #$60      TESTS FOR A CARRAGE RETURN.
00718 FE1A 26 F9          BNE     ASM4A     SKIPS BAK UNTIL C.R.  FOUND.
00719 FE1C 08             INX               POINTS TO FIRST LINE CHAR.
00720 FE1D 20 8B          BRA     ASM1A     GOES BACK TO ASSM.  NEXT LINE
00721                    *
00722                    *
00723                    *
00724                    * THE FOLLOWING ARE SUBROUTINES USED BY THE MAIN CODE.
00725                    *
00726                    *
00727 FE1F 8D 06   INDADR BSR     SYMPTR    GETS CONTENTS OF
00728 FE21 EE 00          LDX     0,X       SYMBOL LOCATION.
00729 FE23 DF 2A          STX     ONDVAL    STORES AS OPERAND.
00730 FE25 20 9B          BRA     SYMBL     RETURNS TO FIX LABEL VALUE.
00731                    *
00732 FE27 48      SYMPTR ASL A   -         MULT LABEL BY 2 TO FORM
00733 FE28 5F             CLR B   -         POINTR INTO SYMBOL TABLE.
00734 FE29 97 01   LOADX  STA A   TMP+1     LOADS POINTER INTO THE
00735 FE2B D7 00          STA B   TMP       SYMBOL TABLE INTO X.
00736 FE2D DE 00          LDX     TMP       RETURNS TO CALLER.
00737 FE2F 39             RTS               RETURNS.
0
```

```
00739 FE30 D6 2A   EXTEND LDA B  ONDVAL   STORES HI BYTE OF OPERAND
00740 FE32 E7 00          STA B  0,X      INTO OBJECT CODE.
00741 FE34 08             INX             INC PC TO POINT TO NXT WD.
00742 FE35 96 2B   DIRECT LDA A  ONDVAL+1 STORES LO BYTE OF OPERAND
00743 FE37 A7 00          STA A  0,X      INTO OBJECT CODE.
00744 FE39 08             INX             INC & SAVE P.C. TO POINT TO
00745 FE3A DF 40          STX    PCVAL    NEXT BYTE.
00746 FE3C 20 D5          BRA    ASM4     GOES TO WORK ON NEXT LINE.
00747              *
00748 FE3E 08      RELTIV INX             INCREMENT P.C. PTR TO POINT
00749 FE3F DF 40          STX    PCVAL    TO NXT BYTE & SAVE P.C..
00750 FE41 96 2B          LDA A  ONDVAL+1 LOADS LO BYTE OF OPERAND.
00751 FE43 90 41          SUB A  PCVAL+1  FORMS RELATIVE OFFSET.
00752 FE45 09             DEX             INSERTS RELATIVE BYTE INTO
00753 FE46 A7 00          STA A  0,X      OBJECT CODE.
00754 FE48 20 C9          BRA    ASM4     GOES TO ASSMBL NEXT LINE.
00755              *
00756              *       END OF THE ASSEMBLER PROGRAM.
00757              *
```



```
00759                      END
.OTAL ERRORS 00000
```

```
00081                 *          DEBUGGER
00082                 *
00083                 *     THE DEBUGGER FOR THE PDS SYSTEM WAS DESIGNED TO
00084                 * PROVIDE A VERSATILE TOOL FOR USE IN PROGRAM TESTING AND
00085                 * DEBUGGING.  IT ALLOWS FOR BREAKPOINTS, MINI-ASSEMBLER
00086                 * SYMBOL TABLE REFERENCING, STACK MANIPULATION AND
00087                 * INPUT IN EITHER HEXADECIMAL, OCTAL OR DECIMAL.
00088                 * THE DEBUGGER PRINTS A PROMPT CHARACTER ">" ON EVERY NEW
00089                 * LINE.  AN INSTRUCTION CAN BE TYPED IN WHENEVER THE CURSOR
00090                 * IS BLINKING, EXCEPT WHEN A NUMBER IS BEING TYPED IN.
00091                 * THE DEBUGGER CALLS THE EDITOR WHENEVER A NUMBER IS TO BE
00092                 * INPUT, SO CORRECTIONS CAN BE MADE IF THE WRONG DIGIT IS
00093                 * TYPED IN.  THE POINTER "PCVAL" POINTS TO THE CURRENTLY
00094                 * OPENED BYTE LOCATION.  THE DEBUGGER OPERATES ON WHATEVER
00095                 * BYTE IS POINTED TO BY PCVAL.  FOR FURTHER DETAILS SEE THE
00096                 * SECTION ON THE DEBUGGER IN THE OPERATORS REFERENCE MANUAL.
00097                 *
00098                 *     THE DEBUGGER IS IMPLEMENTED BY A SMALL ROUTINE TO SET
00099                 * UP ENTRY (DEBUG) AND A LARGE ROUTINE WHICH DOES A RANGE
00100                 * COMPARE TO FIND THE PROPER COMMAND AND THEN EXECUTES THE
00101                 * COMMAND (RUNBUG).  NOTE THAT SINCE COMMANDS ARE
00102                 * DIFFERENTIATED BY RANGE, ANY KEY STRUCK WILL PRODUCE
00103                 * A COMMAND EXECUTION, SUCH AS A "," BEING INTERPRETED
00104                 * AS A "+" COMMAND.
00105                 *
00106                 *
00107                 *
00108                 * COMMANDS:
00109                 *
00110                 * "C. R. " LINE - PRINTS ">" OUT ON A NEW LINE.
00111                 * " " CHANGE - THE SPACE COMND CHANGES CONTENTS FROM Y TO Z.
00112                 * "+" OPNNXT - OPENS NEXT LOCATION.
00113                 * "-" OPNPRE - OPENS PREVIOUS LOCATION.
00114                 * "↑B" BRKSET - SETS A BREAKPOINT AT THE OPENED LOCATION.
00115                 * "↑C" CLRBRK - CLEARS BRKPOINT.  MUST BE DONE BEFORE EXIT.
00116                 * "↑E" EXIT - PERFORM RTI - EXECUTE AT BRKPOINT LOCATION.
00117                 * "↑G" GOLOCN - STARTS EXECUTION AT OPENED LOCATION.
00118                 * "↑J" JUMP   - JUMP TO USERS SUBROUTINE.
00119                 * "↑O" OPNLOC - OPENS LOCATION THAT IS TYPED IN AFTER "O".
00120                 * "↑R" OPNREG - OPENS THE TOP-OF-STACK LOCATION.
00121                 * "↑S" SETSTK - SETS THE STACK TO THE OPENED LOCATION.
00122                 * "↑I" OPNTBL - OPENS LOCATION IN SYMBOL TABLE OF NEXT CHR.
00123                 * "↑X" GOEXEC - EXITS THE DEBUGGER - GOES BACK TO EXEC.
00124                 *
00125                 *
00126                 *
00127                 * SUBROUTINES:
00128                 *
00129                 * INPCHR - INPUTS A CHAR. INTO A AND PRINTS IT.
00130                 * INPNUM - INPUTS A NUMBER INTO B-A FROM THE KEYBOARD.
00131                 * PNTBYT - PRINTS ACC A AS 2 HEX DIGITS ON THE SCREEN.
00132                 * PNTDIG - PRINTS B-A AS 4 HEX DIGITS ON THE SCREEN.
00133                 * NEWLIN - PRINTS A C.R. AND A ">" ON THE SCREEN.
00134                 * DSPADR - PRINTS BYTE ADDRESS (XXXX) AND BYTE CONTENTS (YY)
00135                 *   AS >XXXX YY ON THE SCREEN.
00136                 *
```

```
00138 FE4A                    ORG    $FE4A
00139                 *
00140                 *
00141                 *
00142                 *       FOLLOWING IS LOCATION OF ENTRY OF THE BRKPT VECTOR.
00143                 *
00144 FE4A 30         BKENTR TSX                    INDEX GETS STACK POINTER.
00145 FE4B E6 05              LDA B  5,X            LOADS HI RETURN ADDRESS.
00146 FE4D A6 06              LDA A  6,X            LOADS LOW BYTE OF ADDRESS.
00147 FE4F 80 01              SUB A  #1             SUB 1 FROM RETURN ADDRESS.
00148 FE51 C2 00              SBC B  #0
00149 FE53 E7 05              STA B  5,X            RESTORES RETURN ADDR. TO
00150 FE55 A7 06              STA A  6,X            THE BREAKPOINT LOCATION.
00151 FE57 20 0B              BRA    DEBUG          GOES TO THE DEBUGGER.
00152                 *
00153                 *
00154 FE59 81 0D      LINE    CMP A  #$0D           TESTS FOR A C.R.  (LINE).
00155 FE5B 2D 7D              BLT    JMPLCN         GOES TO 'JSR' (↑J) ROUTNE.
00156 FE5D 2E 4E              BGT    OPNREG         SKIPS FOR NEXT (↑R) TESTS.
00157 FE5F BD FCCB            JSR    SUB32          MOVES CURSOR UP ONE LINE.
00158 FE62 31         POPLIN  INS                   CLEANS UP STACK FOR
00159 FE63 31                 INS                   DISPLAY OF C.R.  >.
00160                 *
00161                 *
00162 FE64 8D 76      DEBUG   BSR    NEWLIN         PRINTS "C.R.>".
00163 FE66 8D 09      DBUG1   BSR    INPCHR         READS IN COMMAND.
00164 FE68 BD FCA5            JSR    RTARRO         INSERT BLANK.
00165 FE6B DE 40              LDX    PCVAL          LOADS CURRENTLY OPENED LOC.
00166 FE6D 8D 09              BSR    RUNBUG         EXECUTES DEBUG COMMAND.
00167 FE6F 20 F5              BRA    DBUG1          GOES BACK FOR NEXT COMND.
00168                 *
00169                 *
00170 FE71 BD FC4A    INPCHR  JSR    GETCHR         READ IN CHAR INTO A.
00171 FE74 BD FCBC            JSR    PUTCHR         DISPLAYS CHARACTER.
00172 FE77 39                 RTS                   RETURNS TO CALLER.
00173                 *
00174                 *
00175                 *
00176 FE78 81 03      RUNBUG  CMP A  #$03           TESTS FOR A "↑C" COMMAND.
00177 FE7A 2D 25              BLT    BRKSET         SKIPS IF A "↑B" COMMAND.
00178 FE7C 2E 45              BGT    EXIT           SKIPS FOR NEXT COMND TEST.
00179 FE7E DE 30      CLRBRK  LDX    BRKADR         GETS ADDRESS OF BREAKPOINT.
00180 FE80 96 2E              LDA A  BRKSAV         LOADS ORIG BYTE CONTENTS.
00181 FE82 A7 00              STA A  0,X            RESTORES BYTE DATA.
00182 FE84 20 66              BRA    DSPADR         GOES TO OPEN THE LOCATION.
00183                 *
00184                 *
00185 FE86 81 20      CHANGE  CMP A  #$'            TESTS OFR A SPACE COMND.
00186 FE88 2D 12              BLT    EXECTV         SKIPS TO EXIT BACK TO EXEC.
00187 FE8A 2E 09              BGT    OPNPRE         SKIPS FOR OTHER CMND TESTS.
00188 FE8C 8D 56      SPACE   BSR    INPNUM         INPUTS NEW BYTE CONTENTS.
00189 FE8E DE 40              LDX    PCVAL          LOADS OPENED BYTE LOCATION.
00190 FE90 A7 00              STA A  0,X            STORES NEW BYTE CONTENTS.
00191                 *
00192                 *
00193 FE92 08         OPNNXT  INX                   FORMS NEXT LOCATION ADDRES.
00194 FE93 20 57              BRA    DSPADR         GOES TO OPEN LOCATION BYTE.
```

```
00196 FE95 81 2D   OPNPRE CMP A  #$'-     TESTS FOR A "-" COMMAND.
00197 FE97 2D F9          BLT    OPNNXT   SKIPS FOR A "+" COMMAND.
00198 FE99 09             DEX             FORMS PREV. LOCATION ADDR.
00199 FE9A 20 50          BRA    DSPADR   GOES TO OPEN THE LOCATION.
00200                     *
00201                     *
00202 FE9C 31   EXECTV    INS             CLEANS UP THE STACK.
00203 FE9D 31             INS
00204 FE9E 7E FC14        JMP    EXEC     RETURNS TO THE EXECUTIVE.
00205                     *
00206                     *
00207 FEA1 A6 00  BRKSET  LDA A  0,X      LOADS DATA OF OPNED LOCATN.
00208 FEA3 97 2E          STA A  BRKSAV   SAVES DATA OF OPNED BYTE.
00209 FEA5 DF 30          STX    BRKADR   SAVES ADDR. OF BREAKPOINT.
00210 FEA7 86 3F          LDA A  #$3F     LOADS SOFTWARE INTUP COMND.
00211 FEA9 A7 00          STA A  0,X      SETS AN SWI AT OPNED BYTE.
00212 FEAB 20 B5          BRA    POPLIN   GOES TO NEXT LINE FOR COMND.
00213                     *
00214                     *
00215 FEAD 81 12  OPNREG  CMP A  #$12     TESTS FOR "↑R" (STACK TOP).
00216 FEAF 2D 0A          BLT    OPNLOC   GOES TO OPEN A LOCATION.
00217 FEB1 2E 19          BGT    OPNTBL   SKIPS FOR NEXT TEST (↑T).
00218 FEB3 30             TSX             OPENS TOP-OF-STACK.
00219 FEB4 08             INX             PCVAL GETS STACK POINTER.
00220 FEB5 08             INX             (CLEANS UP THE STACK).
00221 FEB6 20 34          BRA    DSPADR   GOES TO DISPLAY THE T-O-S.
00222                     *
00223                     *
00224 FEB8 35   SETSTK    TXS             STACK POINTER GETS PCVAL.
00225 FEB9 20 A9          BRA    DEBUG    RETURNS TO INPUT COMMAND.
00226                     *
00227                     *
00228 FEBB 8D 27  OPNLOC  BSR    INPNUM   LOADS A 16 BIT NUMBER.
00229 FEBD D7 40  OPNLC1  STA B  PCVAL    STORES NEWLY OPENED
00230 FEBF 97 41          STA A  PCVAL+1  LOCATION ADDRESS.
00231 FEC1 20 2B          BRA    DSPAD1   DISPLAYS CONTNTS OF LOCATN.
00232                     *
00233                     *
00234 FEC3 81 07  EXIT    CMP A  #$07     TESTS IF AN EXIT (↑E) COMD.
00235 FEC5 27 11          BEQ    GOLOCN   SKIPS FOR THE "GO" COMMAND.
00236 FEC7 2E 90          BGT    LINE     SKIPS FOR NEXT COMMD TEST.
00237 FEC9 31             INS             CLEARS UP THE STACK.
00238 FECA 31             INS
00239 FECB 3B             RTI             RETURNS FROM BREAKPOINT.
00240                     *
00241                     *
00242 FECC 81 14  OPNTBL  CMP A  #$14     TESTS IF A "↑T" (TABLE).
00243 FECE 2D E8          BLT    SETSTK   GOES TO SET STACK PTR (↑S).
00244 FED0 2E B4          BGT    CHANGE   SKIPS FOR NEXT TEST (SPACE).
00245 FED2 8D 9D          BSR    INPCHR   LOADS A WITH SYMBOL (LABL).
00246 FED4 48             ASL A  .        ALIGNS ADDRESS FOR
00247 FED5 5F             CLR B  .        SYMBOL TABLE ENTRY.
00248 FED6 20 E5          BRA    OPNLC1   SAVES AND DISPLAYS ADDRESS.
00249                     *
00250                     *
00251 FED8 31   GOLOCN    INS             CLEANS UP THE STACK.
00252 FED9 31             INS
00253 FEDA 6E 00  JMPLCN  JMP    0,X      JUMPS TO USERS PROGRAM.
```

```
00255                    *
00256                    * FOLLOWING ARE SUBROUTINES USED BY THE DEBUGGER.
00257                    *
00258 FEDC 86 0D   NEWLIN LDA A  #$0D      LOADS A CARRIAGE RETURN.
00259 FEDE 8D 3E          BSR    PNTBF1    PRINTS A CARRIAGE RETURN.
00260 FEE0 86 3E          LDA A  #'>       LOADS A PROMPT CHARACTER.
00261 FEE2 20 3A          BRA    PNTBF1    DISPLAYS PROMPTER CHAR.
00262                    *
00263                    *
00264                    * FOLLOWING INPUTS A 16 BIT NUMBER INTO THEE BA REGISTER.
00265                    *
00266 FEE4 BD FC75 INPNUM JSR    EDITIN    INPUTS A STRING OF DIGITS.
00267 FEE7 DE 24          LDX    SCNPTR    LOADS ADDR. OF FIRST DIGIT.
00268 FEE9 8D 37          BSR    ASCBIN    CONVERTS TO BINARY # IN BA.
00269 FEEB 39             RTS              RETURNS TO CALLER.
00270                    *
00271                    *
00272                    * FOLLOWING DISPLAYS THE LOCATION ADDR. & CONTENTS.
00273                    *
00274 FEEC DF 40   DSPADR STX    PCVAL     SAVES OPENED LOCATION ADDR.
00275 FEEE 8D EC   DSPAD1 BSR    NEWLIN    PRINTS A "C.R." AND ">".
00276 FEF0 8D 0A          BSR    PNTDIG    PRINTS OUT "PCVAL" IN HEX.
00277 FEF2 BD FCA5        JSR    RTARRO    PRINTS A SPACE.
00278 FEF5 DE 40          LDX    PCVAL     LOADS PTR. TO OPEND LOC.
00279 FEF7 A6 00          LDA A  0,X       LOADS DATA FROM LOCATN.
00280 FEF9 8D 07          BSR    PNTBYT    PRINTS DATA IN HEX FORMAT.
00281 FEFB 39             RTS              RETURNS TO INPUT COMMAND.
00282                    *
00283                    *
00284 FEFC 96 40   PNTDIG LDA A  PCVAL     PRINTS THE 2 HI HEX DIGITS
00285 FEFE 8D 02          BSR    PNTBYT    OF OPENED ADDRESS.
00286 FF00 96 41          LDA A  PCVAL+1   PRINTS OUT 2 LOW HEX DIGITS.
00287                    *
00288                    *
00289                    * FOLLOWING PRINTS OUT 2 HEX DIGITS.
00290                    *
00291 FF02 CE 0010 PNTBYT LDX    #16       LOADS 16 FOR BASE.
00292 FF05 DF 04          STX    ARB       STORES FOR CONVERSION.
00293 FF07 5F             CLR B  -         CLEARS HI 2 DIGITS.
00294 FF08 CE 0035        LDX    #IOBUFF   LOADS OUTPUT BUFF ADDRESS.
00295                    *
00296                    *
00297                    * FOLLOWING CONVERTS BYTE TO HEX WITH LEADING ZEROS.
00298                    *
00299 FF0B D7 36   CONVRT STA B  IOBUFF+1 CLERS BYTE FOR SECOND DIGIT..
00300 FF0D BD FF64        JSR    BINASC    CONVERTS TO ASCII DIGITS.
00301 FF10 96 35          LDA A  IOBUFF    LOADS HI DIGIT.
00302 FF12 D6 36          LDA B  IOBUFF+1 TESTS BOTH DIGITS CONVTD.
00303 FF14 26 04          BNE    PNTBUF    SSKIPS IF BOTH DIGITS CONVTD00304 FF16
 97 36          STA A  IOBUFF+1 SETS UP LOW DIGIT.
00305 FF18 86 30          LDA A  #$'0      HIGH DIGIT GETS A "0".
00306                    *
00307 FF1A 8D 02   PNTBUF BSR    PNTBF1    PRINTS OUT HI DIGIT.
00308 FF1C 96 36          LDA A  IOBUFF+1 LOADS LOW DIGI
00309 FF1E BD FC8C PNTBF1 JSR    PUTCHR    DISPLAYS CCHARACTER.
00310 FF21 39             RTS              RETURNS TO CALLING PROGRAM.
00311                    *
00312                    *        END OF DEBUGGER
```

```
00314                *      UTILITY PROGRAMS
00315                *
00316                *
00317                *
00318                *
00319                *           ASCII TO BINARY CONVERSION.
00320                *
00321                *     THE ASCII TO BINARY ROUTINE CONVERTS FROM AN ASCII
00322                * NUMBER STRING POINTED TO BY X TO AN UNSIGNED 16 BIT
00323                * BINARY NUMBER IN BA (ACC B HAS THE HI BYTE, ACC A HAS
00324                * THE LO BYTE).  THE ASCII STRING IS TERMINATED BY A NON
00325                * HEXADECIMAL CHARACTER.  UPON EXITING, THE INDEX REGISTER
00326                * WILL POINT TO THE NEXT CHARACTER AFTER THE NUMBER
00327                * STRING.  THE BASE OF THE NUMBER STRING IS PASSED TO
00328                * THE ROUTINE IN ARA (ARA IS THE ARITHMETIC REGESTER A
00329                * LOCATED IN BYTES 06 AND 07 OF LOW MEMORY).  IF THE
00330                * ROUTINE IS ENTERED WITH A KNOWN BASE, PUT THE BASE
00331                * (BETWEEN 2 AND 16) IN ARA AND ENTER THE ROUTINE AT
00332                * THE ENTRY POINT ENTR2.
00333                *
00334                *
00335                *
00336                *     CONVERSION FORMULA:
00337                * ASCII NUMBBER STRING X[4],X[3],X[2],X[1] IN
00338                *BASE Y;
00339                * BINARY NUMBER =
00340                *     X[4]*Y↑3+X[3]*Y↑2+X[2]*Y↑1+X[1]*Y↑0          OR
00341                * BINARY NUMBER =
00342                *     (((0*Y+X[4])*Y+X[3])*Y+X[2])*Y+X[1]
00343                *WHERE ↑ IS THE EXPONENT OPERATOR,
00344                * X IS A CHARACTER & Y IS THE BASE.
00345                *
00346                *
00347                *
00348                *     ALGORITHM:
00349                *ASCBIN:  FORM THE BASE IN ARA BASED ON THE FIRST CHAR.
00350                *  OF THE NUMBER STRING; INCREMENT CHAR. PTR. IN X;
00351                *ENTR2:  NUMBER (IN BA) GETS 0;
00352                *NXTCHR:  IF THE CURRENT CHAR. POINTED TO BY X IS NOT A
00353                *  DIGIT THEN EXIT ELSE INCREMT CHARACTER PTR IN INDEX;
00354                *  CONVERT DIGIT TO BINARY;
00355                *  NUMBER GETS NUMBER * BASE;
00356                *  NUMBER GETS NUMBER + DIGIT;
00357                *  GO TO OPERATE ON THE NEXT DIGIT (NXTCHR);
00358                *
```

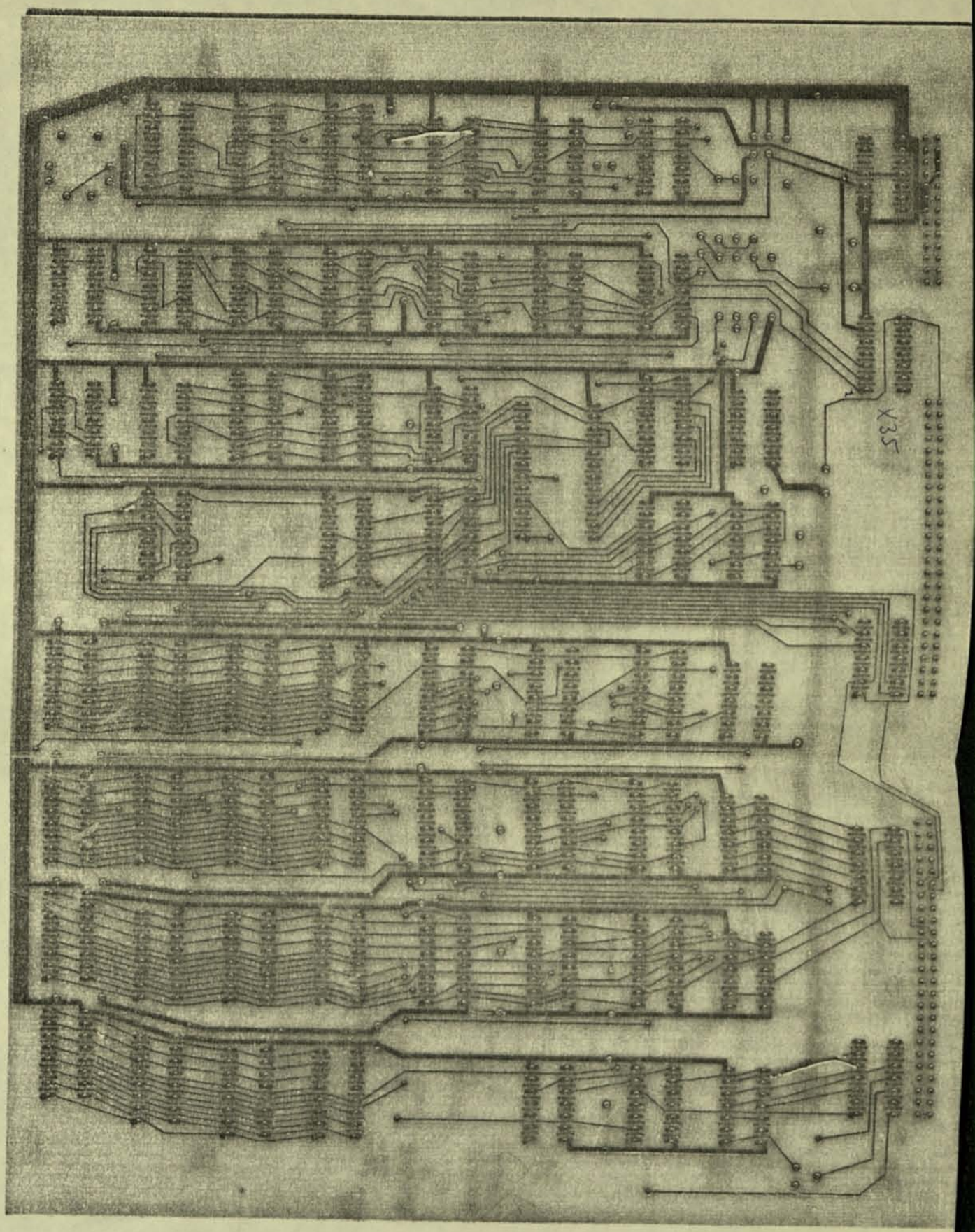
```
00360 FF22 A6 00   ASCBIN LDA A  0,X      GETS CHR TO FORM BASE.
00361 FF24 81 2E           CMP A  #$'.    TESTS FOR DECML STRNG.
00362 FF26 2D 06           BLT    OCT     SKIPS IF BASE 8 (*).
00363 FF28 2E 09           BGT    HEX     SKIPS IF BASE 16.
00364 FF2A 86 0A           LDA A  #10     LOADS BASE 10 FOR CONVERSN.
00365 FF2C 20 02           BRA    ASC1    SKIPS TO INC. TEXT POINTR.
00366 FF2E 86 08   OCT     LDA A  #8      LOADS BASE 8 FOR CONVERSION.
00367 FF30 08      ASC1    INX            INCREMENT PTR TO NEXT CHAR.
00368 FF31 20 02           BRA    ASC2    SKIPS TO SAVE BASE.
00369 FF33 86 10   HEX     LDA A  #16     LOADS BASE 16 FOR CONVERN.
00370 FF35 97 07   ASC2    STA A  AR0     SAVES BASE IN BASE#.
00371              *
00372              *
00373 FF37 5F      ENTR2   CLR B  NUMBER  GETS 0.
00374 FF38 37              PSH B  -       (LOW NUMBER ON STACK).
00375 FF39 D7 06           STA B  AR1     CLEARS HI OF BASE.
00376 FF3B A6 00   NXTCHR  LDA A  0,X     GETS CHAR TO CONVERT.
00377 FF3D 08              INX            INC TO NEXT CHARACTER.
00378 FF3E 81 30           CMP A  #$'0    TESTS FOR END-OF-STRING.
00379 FF40 2D 20           BLT    AEXIT   EXITS IF END.
00380 FF42 80 30           SUB A  #$'0    FORMS B.C.D. NUMBER.
00381 FF44 81 0A           CMP A  #10     TESTS IF DECIMAL DIGIT.
00382 FF46 2D 0A           BLT    ASC3    SKIPS IF DECIMAL.
00383 FF48 81 10           CMP A  #16     TESTS FOR END OF STRING.
00384 FF4A 2F 16           BLE    AEXIT   EXITS IF NOT A HEX DIGIT.
00385 FF4C 80 07           SUB A  #7      FORMS A HEX B.C.D. DIGIT.
00386 FF4E 81 10           CMP A  #16     TESTS FOR END-OF-STRING.
00387 FF50 2C 10           BGE    AEXIT   EXITS IF CHAR > "F".
00388 FF52 97 08   ASC3    STA A  DIGIT   SAVES DIGIT FOR ADD.
00389              *
00390 FF54 DF 00   CNVASC  STX    TMP     SAVES INDEX REG FOR MULT.
00391 FF56 32              PUL A  -       RESTORES LO OF "NUMBER".
00392 FF57 8D 3A           BSR    MULT    NUMBER GETS NUMBER * BASE.
00393 FF59 9B 08           ADD A  DIGIT   NUMBER GETS NUMBER + DIGIT.
00394 FF5B C9 00           ADC B  #0
00395 FF5D 36              PSH A  -       SAVES LO OF NUMBER.
00396 FF5E DE 00           LDX    TMP     RESTORES STRING POINTER.
00397 FF60 20 D9           BRA    NXTCHR  GOES TO CONVRT NEXT CHAR.
00398 FF62 32      AEXIT   PUL A  -       RESTORES "NUMBER" IN BA.
00399 FF63 39              RTS            RETURNS TO CALLING PROGRAM.
```

```
00401                *          BINARY TO ASCII
00402                *
00403                *
00404                *     THE BINARY TO ASCII CONVERSION ROUTINE CONVERTS
00405                * A 16 BIT BINARY NUMBER IN THE BA REGISTR (REG B & REG A)
00406                * TO A STRING OF ASCII DIGITS.  THE ASCII STRING CAN BE IN
00407                * ANY BASE FROM BASE 2 THROUGH BASE 41.  THE VALUE OF THE
00408                * BASE IS LOCATED IN THE ARITHMETIC PSEUDO-REGISTER ARB
00409                * (ARB IS LOOCATED IN BYTE AR3 [LOC 4] AND AR2 [LOC 5]).
00410                * WHEN THE ROUTINE IS ENTERED, THE POINTER TO THE OUTPUT
00411                * LOCATION IS PASSED IN THE INDEX REG.  WHEN THE ROUTINE
00412                * EXITS, THE INDEX POINTS TO THE LAST DIGIT IN THE STRING
00413                * PLUS ONE.
00414                *     CONVERSION IS DONE BY THE METHOD OF REPEATED
00415                * DIVISION.  THE LOW ORDER DIGIT IS FORMED FIRST.  THE
00416                * DIGITS ARE THEN PLACED ON THE STACK UNTIL CONVERSION IS
00417                * COMPLETED.  THE DIGITS ARE THEN POPPED OFF THE STACK
00418                * AND PLACED IN THE OUTPUT STRING.  THE TOP-OF-STACK IS
00419                * INITIALIZED TO HEX FF TO TELL WHEN ALL THE DIGITS
00420                * HAVE BEEN POPPED OFF THE STACK.  AFTER THE DIVISION, THE
00421                * DIGIT (THE REMAINDER OF THE DIVISION OPERATION)
00422                * IS LOCATED IN THE AR 0 PART OF ARA (BYTE 7).  WHEN
00423                * THE QUOTIENT OF THE DIVISION IS 0, THEN THE CONVERSION
00424                * IS COMPLEATED.
00425                *
00426                *
00427                *
00428 FF64 DF 00    BINASC STX    TMP       SAVES OUTPUT POINTER.
00429 FF66 34               DES             /SETS THE TOP-OF-STACK TO
00430 FF67 30               TSX             /ALL ONES TO TELL END OF
00431 FF68 6F 00            CLR    0,X      /CHAR STRING (LAST CHAR IS
00432 FF6A 63 00            COM    0,X      /PUT ON STACK FIRST).
00433 FF6C DE 04    BIN1    LDX    ARB      RESTORES DIVISOR (BASE).
00434 FF6E DF 06            STX    ARA
00435 FF70 8D 3D            BSR    DIVIDE   * QUOTIENT IN BA GETS THE
00436                * REMAINDER OF # TO BE CONVERTED; REMAINDER IN ARA GETS
00437                * THE LOW ORDER DIGIT.
00438 FF72 97 02            STA A  TMP1     SAVES A OF BA.
00439 FF74 96 07            LDA A  AR0      LOAD DIGIT (REMAINDER).
00440 FF76 36               PSH A  -        STACK DIGIT (REVERSE ORDER). G
00441 FF77 96 02            LDA A  TMP1     RESTORES A OF BA.
00442 FF79 4D               TST A  -        /TESTS IF QUOTIENT IS = 0
00443 FF7A 26 F0            BNE    BIN1     /(SIGNIFYING THAT
00444 FF7C 5D               TST B  -        /THE CONVERRSION
00445 FF7D 26 ED            BNE    BIN1     /IS DONE).
00446                *
00447 FF7F DE 00    BINSTR  LDX    TMP      RESTORES OUTPUT POINTER.
00448 FF81 32       BIN3    PUL A  -        UNSTACK A DIGIT.
00449 FF82 4D               TST A  -        TESTS IF NEG (END?).
00450 FF83 2A 01            BPL    BIN4     SKIPS IF A DIGIT.
00451 FF85 39               RTS             EXITS FROM SUBROUTINE.
00452 FF86 81 09    BIN4    CMP A  #9       TESTS IF RESULT IS HEX.
00453 FF88 2F 02            BLE    BIN5     SKIPS IF DIGIT NOT HEX.
00454 FF8A 8B 07            ADD A  #7       FORMS HEX VALUE OF DIGIT.
00455 FF8C 8B 30    BIN5    ADD A  #$'0     FORMS DECIMAL CHARACTER.
00456 FF8E A7 00            STA A  0,X      OUTPUTS CHARACTER.
00457 FF90 08               INX             POINTS TO NEXT CHARACTER.
00458 FF91 20 EE            BRA    BIN3     GOES BACK FOR NEXT DIGIT.
```

```
00460                    *          MULTIPLY ROUTINE
00461                    *
00462                    *     THE MULTIPLY ROUTINE MULTIPLIES TWO 16 BIT BINARY
00463                    * NUMBERS TOGETHER TO PRODUCE A 16 BIT RESULT.  THE BA
00464                    * REGISTERS AND ARA (BYTES 6 & 7) REGISTER ARE USED.
00465                    * THE CONTENTS OF ARA ARE UNCHANGED UPON PROGRAM EXIT.
00466                    *
00467                    *
00468                    *  BA GETS BA * ARA
00469                    *
00470                    *
00471                    *     MULTIPLYING IS ACCOMPLISHED BY REPEATED ADDITIONS
00472                    * OF ONE OF THE OPERATORS (OPERATOR ARA) INTO THE RESULT.
00473                    * THE RESULT STARTS OUT WITH A ZERO VALUE AND IS SHIFTED
00474                    * OVER ONE AFTER EACH ADDITION.  THE HIGHEST ORDER VALUE
00475                    * IS ADDED IN FIRST AND THEN, GOING TO THE RIGHT,
00476                    * (THUS SHIFTING THE ANSWER LEFT ONE TO BRING IN THE NEXT
00477                    * RIGHTMOST DIGIT) GETTING THE NEXT LONERMOST SIGNIFICANT
00478                    * DIGIT.  THE NEXT RIGHTMOST BIT OF THE OTHER OPERAND
00479                    * (THE ONE ORIGINALLY IN BA) IS TESTED, AND IF ONE,
00480                    * ANOTHER ADDITION TAKES PLACE.  THIS IS REPEATED UNTIL
00481                    * THE FINAL SUM IS FORMED.
00482                    *
00483                    *
00484                    *
00485                    *  MULTIPLY ALGORITHM:
00486                    *
00487                    *MULT:  STACK BA; BA GETS 0; SET COUNT VALUE TO 16;
00488                    *MUL1:  SHIFT BA LEFT 1;
00489                    *  SHIFT LEFT ORIG BA VALUE ON STACK INTO CARRY;
00490                    *  IF CARRY = 0 THEN GO TO MUL2
00491                    *  BA GETS BA + ARA;
00492                    *MUL2:  DECREMENT COUNT;
00493                    *  IF COUNT # 0 THEN GO TO MUL1 ELSE EXIT.
00494                    *
00495                    *
00496                    *
00497                    *
00498 FF93 36            MULT   PSH A  -        PUTS THE ORIGINAL CONTENTS
00499 FF94 37                   PSH B  -        OF BA ONTO THE STACK.
00500 FF95 86 10                LDA A  #16      LOADS COUNT VALUE
00501 FF97 36                   PSH A  -        ONTO THE STACK.
00502 FF98 4F                   CLR A  -        BA GETS ZEROED.
00503 FF99 5F                   CLR B
00504 FF9A 30                   TSX             SET INDEX TO STACK.
00505 FF9B 48            MUL1   ASL A  -        SHIFT LEFT BA.
00506 FF9C 59                   ROL B
00507 FF9D 68 02                ASL    2,X      SHIFTS ORIG. BA OPERAND
00508 FF9F 69 01                ROL    1,X      ONE LEFT INTO CARRY.
00509 FFA1 24 04                BCC    MUL2     SKIPS ADDING IF CARRY = 0.
00510 FFA3 9B 07                ADD A  AR0      BA GETS BA + ARA.
00511 FFA5 D9 06                ADC B  AR1
00512 FFA7 6A 00         MUL2   DEC    0,X      TESTS IF DONE.
00513 FFA9 26 F0                BNE    MUL1     GOES BACK IF NOT DONE.
00514 FFAB 31                   INS             CLEANS UP THE STACK.
00515 FFAC 31                   INS
00516 FFAD 31                   INS
00517 FFAE 39                   RTS             EXITS ROUTINE.
```

```
00519                    *          DIVIDE ROUTINE
00520                    *
00521                    *     THE DIVIDE SUBROUTINE DIVIDES THE 16 BIT NUMBER
00522                    * IN THE BA REGISTERS BY THE 16 BIT NUMBER IN THE PSEUDO
00523                    * REGESTER ARA (LOCATED IN BYTES 6 & 7).  UPON EXITING,
00524                    * BA WILL CONTAIN THE QUOTIENT OF THE DIVISION AND ARA
00525                    * WILL CONTAIN THE REMAINDER.  THE DIVIDEND BA IS DIVIDED
00526                    * BY THE DIVISOR ARA (I.E.  BA/ARA ).
00527                    *
00528                    *
00529                    *
00530                    *
00531                    *  DIVIDE ALGORITHM:
00532                    *
00533                    *DIVIDE:  X3,4 GETS BA (BA IS PUT ON THE STACK);
00534                    *  X1,2 GETS THE 16 BIT ARA VALUE (ARA PUT ON THE STACK);
00535                    *  COUNT GETS 1 + THE NUMBER OF NONSIGNIFICANT BITS IN
00536                    * THE DIVISOR (LEFT JUSTIFY X1,2 TO FIRST 1 BIT, COUNT
00537                    * GETS 1 + THE # OF LEADING ZEROS IN X1,2) (COUNT WILL BE
00538                    * FROM 1 TO 17);
00539                    *  BA GETS X3,4 (RESTORES BA);
00540                    *  X3,4 GETS 0 (INITIALIZES THE QUOTIENT);
00541                    *DIV3:  BA GETS BA - X1,2;
00542                    *  IF THERE WAS A BORROW (I.E. DIVIDEND IN BA < DIVISOR
00543                    * IN X1,2 (ARA)) THEN BA GETS BA + X1,2 (ORIGINAL BA
00544                    * VALUE RESTORED) & CARRY CLEARED ELSE CARRY IS SET;
00545                    *DIV5:  X3,4 (QUOTIENT) GETS CARRY LEFT SHIFTED IN;
00546                    *  X1,2 (DIVISOR) GETS SHIFTED RIGHT ONE PLACE WITH ZERO
00547                    * FILLED IN FROM THE LEFT SIDE;
00548                    *  DECREMENT COUNT;
00549                    *  EXIT IF DONE ELSE GO TO DIV3;
00550                    *
00551                    *
00552                    *
00553                    *
00554                    *
00555 FFAF 36       DIVIDE PSH A  -        LOADS DIVIDEND INTO X3,4.
00556 FFB0 37              PSH B
00557 FFB1 96 06           LDA A  AR1      LOADS DIVISOR FROM ARA.
00558 FFB3 D6 07           LDA B  AR0
00559 FFB5 37              PSH B  -        PUTS DIVISOR INTO X1,2.
00560 FFB6 36              PSH A
00561 FFB7 34              DES             SET UP SPACE FOR COUNT.
00562 FFB8 30              TSX             INDEX GETS STACK POINTER.
00563 FFB9 86 01           LDA A  #1       INITIALIZE COUNT.
00564 FFBB 6D 01           TST    1,X      TESTS FOR HI DIVISR BIT ON.
00565 FFBD 2B 0B           BMI    DIV2     SKIPS IF ON.
00566 FFBF 4C       DIV1   INC A  -        COUNTS LEADING ZEROS.
00567 FFC0 68 02           ASL    2,X      LEFT JUSTIFIES X1,2.
00568 FFC2 69 01           ROL    1,X
00569 FFC4 2B 04           BMI    DIV2     SKIPS IF NO LEADING ZERO.
00570 FFC6 81 11           CMP A  #17      TESTS FOR ALL ZERO DIVISOR.
00571 FFC8 26 F5           BNE    DIV1     GOES BACK IF BITS LEFT.
00572 FFCA A7 00    DIV2   STA A  0,X      SETS COUNTER.
00573 FFCC E6 03           LDA B  3,X      BA GETS ORIGINAL
00574 FFCE A6 04           LDA A  4,X      DIVIDEND VALUE.
00575 FFD0 6F 03           CLR    3,X      CLEARS X3,4 FOR FORMATION
00576 FFD2 6F 04           CLR    4,X      OF THE QUOTIENT.
```

```
00578 FFD4 A0 02   DIV3   SUB A  2,X      START OF DIVIDE LOOP.
00579 FFD6 E2 01          SBC B  1,X
00580 FFD8 24 07          BCC    DIV4     SKIP IF DIVIDEND < DIVISOR.
00581 FFDA AB 02          ADD A  2,X      RESTORES DIVIDEND IN BA.
00582 FFDC E9 01          ADC B  1,X
00583 FFDE 0C             CLC             CLEARS THE CARRY.
00584 FFDF 20 01          BRA    DIV5     SKIPS WITH CARRY CLEAR.
00585 FFE1 0D      DIV4   SEC             SETS CARRY TO 1.
00586 FFE2 69 04   DIV5   ROL    4,X      SHIFT CARRY INTO
00587 FFE4 69 03          ROL    3,X      QUOTIENT X3,4
00588 FFE6 64 01          LSR    1,X      SHIFTS DIVISOR X1,2
00589 FFE8 66 02          ROR    2,X      RIGHT ONE.
00590 FFEA 6A 00          DEC    0,X      DECREMENTS COUNTER.
00591 FFEC 26 E6          BNE    DIV3     GOES BACK IF NOT DONE.
00592 FFEE D7 06          STA B  AR1      STORES REMAINDER IN ARA.
00593 FFF0 97 07          STA A  AR0
00594 FFF2 31             INS             CLEANS UP THE STACK.
00595 FFF3 31             INS
00596 FFF4 31             INS
00597 FFF5 33             PUL B  STORES   QUOTIENT IN BA.
00598 FFF6 32             PUL A
00599 FFF7 39             RTS             EXITS ROUTINE.
00600                *
00601                *
00602                *
00603 FFF8 0104    IRQ    FDB    $0104    INTERRUPT REQUEST VECTOR.
00604 FFFA FE4F    SWI    FDB    BKENTR   SOFTWARE INT. VECTOR ADDR.
00605 FFFC 0108    NMI    FDB    $0108    NON-MASKABLE-INT. VECT.
00606 FFFE FC00    RST    FDB    $FC00    RESTART VECTOR ADDRESS.
00607                *
00608                *
00609                *
00610                *  END OF PDS SOURCE LISTING.
00611                *
00612                *
00613                     END
TOTAL ERRORS 00000
```



CRT-8 backside

SYSTEM DOCUMENTATION	FORM	SYSTEM	DATE	INDEX	PAGE
			5-23-78	III-B	2

HEADING

BASIS ADJUSTMENT TRANS

SUBSYSTEM

MODULE

field	validity
(1)	{ NL → cancel in Asm-file
(2)	{ NL → cancel in TT-file get record
(3)	{ NL → actual date MO/DA/YR must be greater than eff date of TT-file entry
(4)-(7)	- { NL → ϕ $ amt
(8)-(10)	- any alpha

ASSET CODE [7 (1)]

[Asset name (1a)]
[Line 2 (1b)]

THIS HOLDING? (Y/N/C) [1 (2)] HO-DATE [6 (2a)] #3 DATE [6 (2b)] #4 DATE [6 (2c)] #5 DATE [6 (2d)]

Effective date: [8 (3)]

Holding adj amt: [12 (4)]

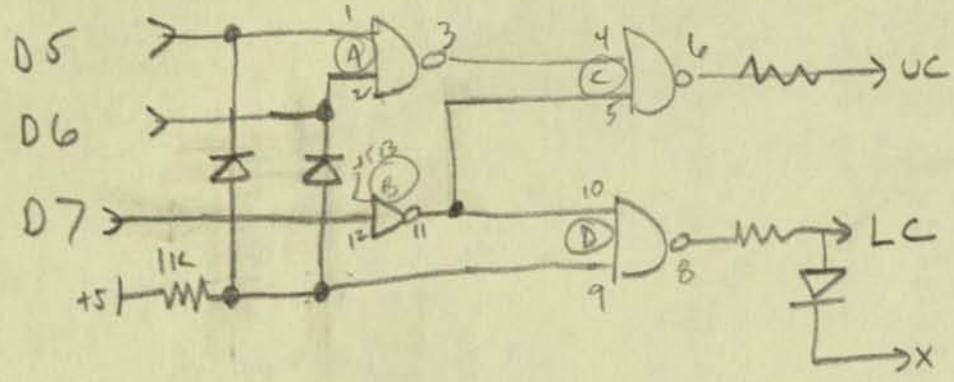
#3 adj amt [12 (5)]

#4 adj amt [12 (6)]

#5 adj amt [12 (7)]

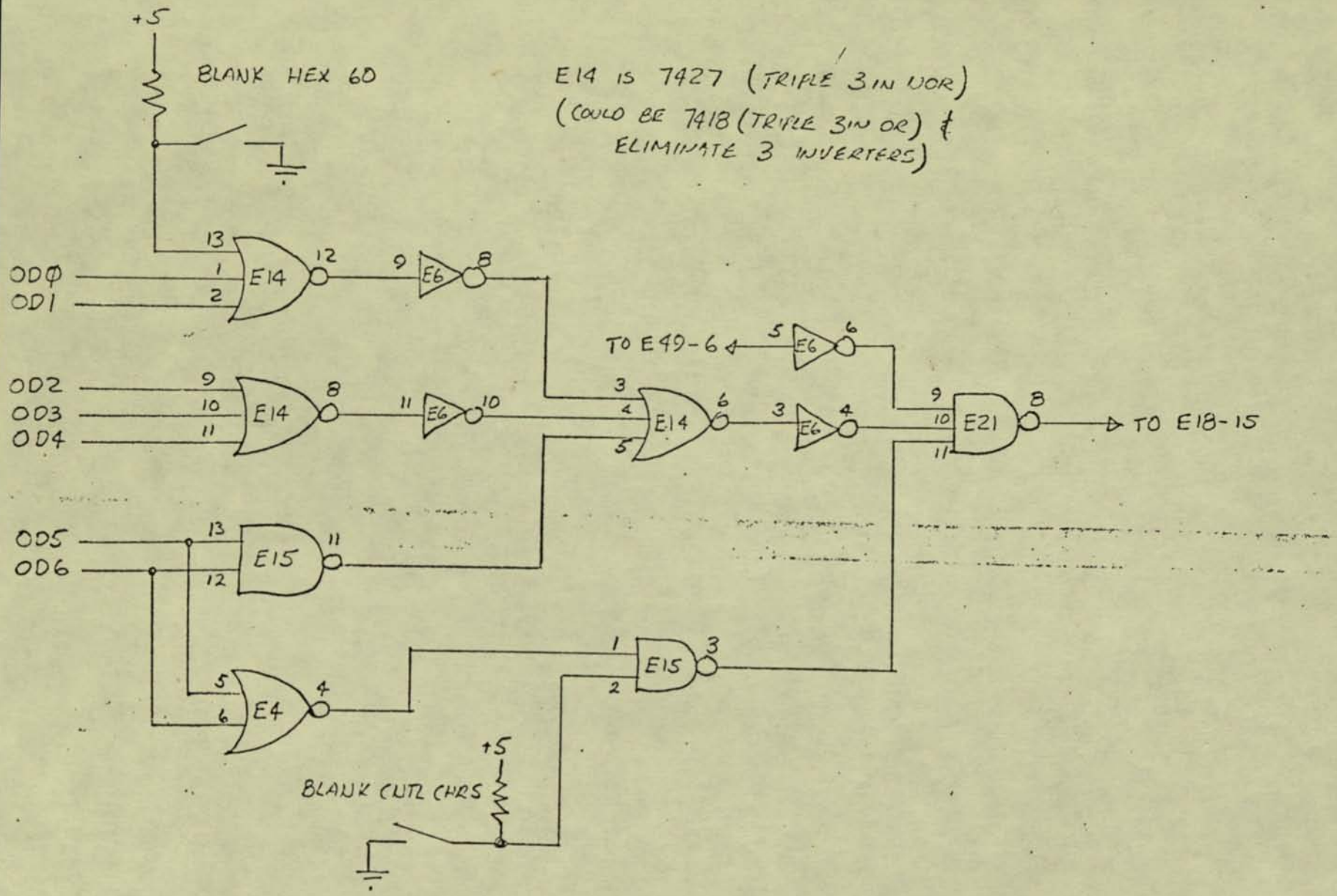
Description

[40 (8)]
[40 (9)]
[40 (10)]



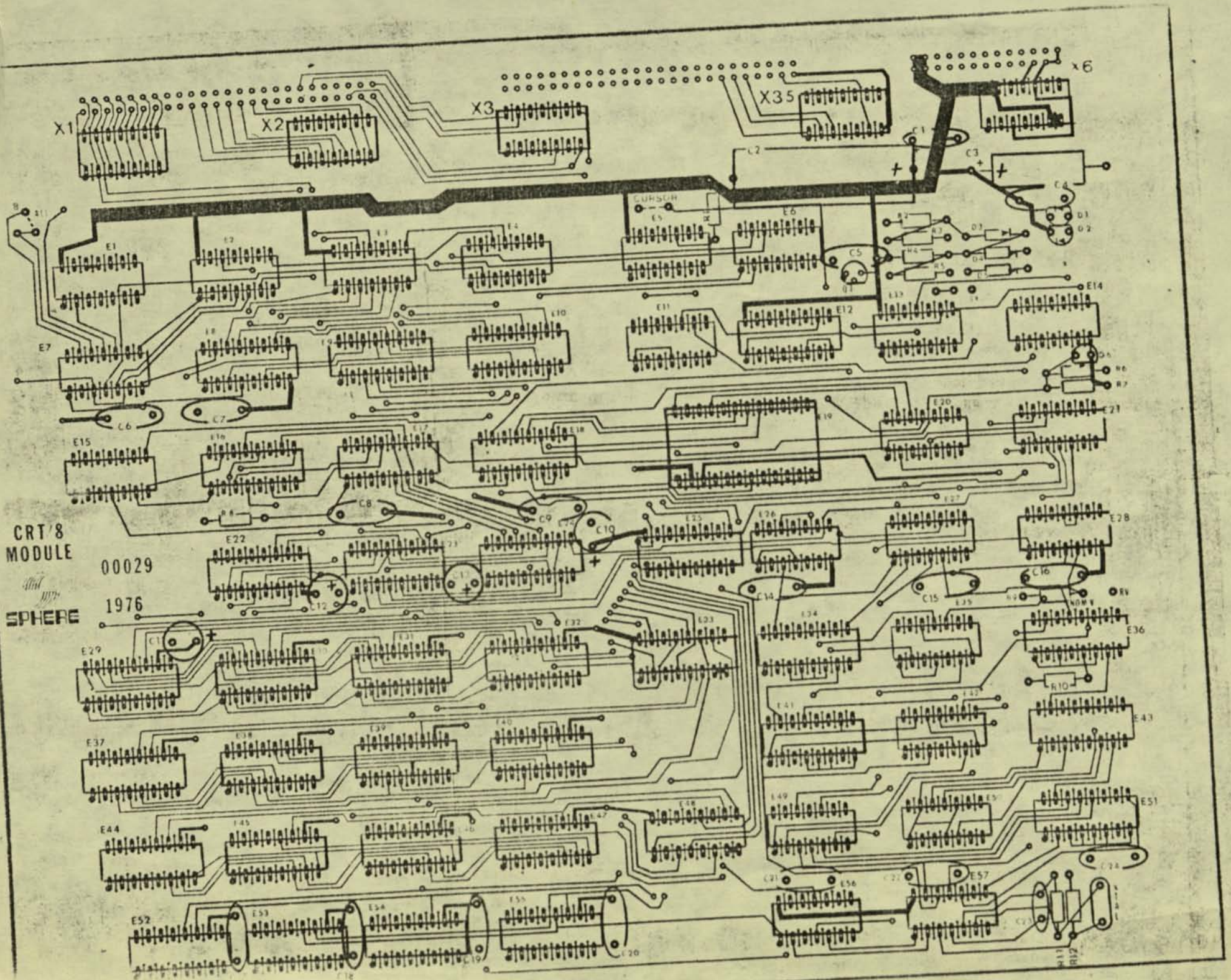
E27-11 — E26-9
E13-9 — E26-10
✓ E26-8 — (E26-2,4,13, diodes tiepoint)

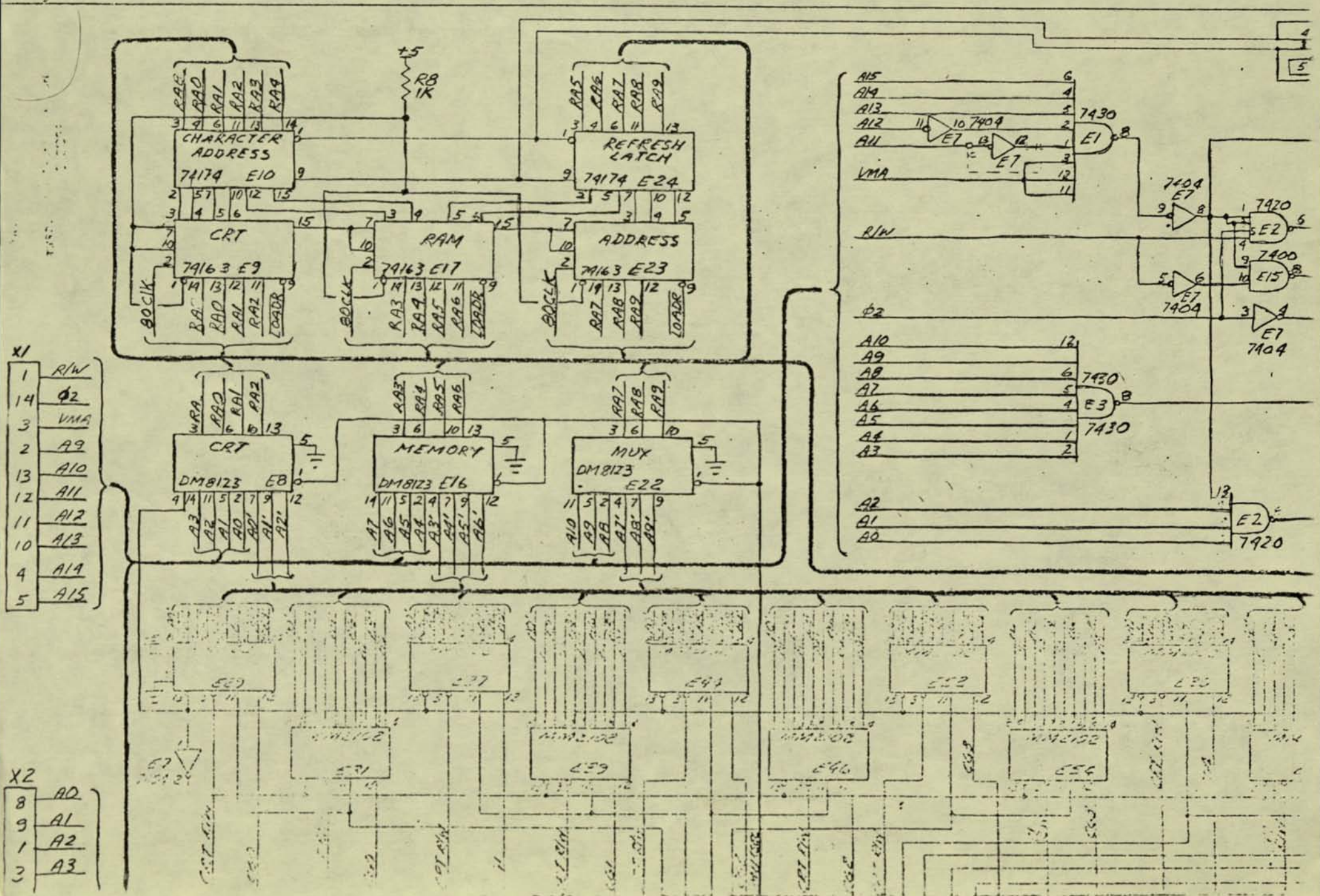
LOWER CASE MOD
(doesn't have control char, so
they still show UC



CIRCUIT TO SELECTIVELY BLANK SPHERE CARRIAGE RETURNS AND/OR ALL CONTROL CHARACTERS

R. J. Mason
Feb '77


X1
X2
X3
X35
X6
CRT/8
MODULE
00029
1976
SPHERE


+5
R8
1K
CHARACTER ADDRESS
74174 E10
REFRESH LATCH
74174 E24
CRT
74163 E9
RAM
74163 E17
ADDRESS
74163 E23
BCLK
LOADR
CRT
DM8123 E8
MEMORY
DM8123 E16
MUX
DM8123 E22
7430
7404
7420
7400
X1
R/W
Φ2
VMA
A9
A10
A11
A12
A13
A14
A15
X2
A0
A1
A2
A3

PROBLEM AREA AND CHANGES

* X1-6 etch should go to X1-5
* Possible short at feed through below pins 6 & 7 of E48
* Possible grounding of leads with card rack near XTAL (lower right)

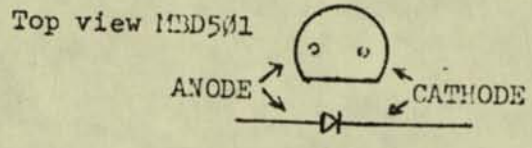
G ASSEMBLY INSTRUCTIONS

G.1 KIT ASSEMBLY INSTRUCTIONS

G.1.6 ASSEMBLY INSTRUCTIONS FOR CRT/8

SOCKET INSERTION- Place the 14 pin sockets in X1,X2,X3,X6 and X35. Place the 16 pin sockets in E34,E36,E41,E43 and E51. Place the 24 pin socket in E19. Watch the alignment of pin 1. Solder sockets carefully into place and verify good solder joints with an OHM meter. All of the IC's could be socketed but this is not required. The memory chips (MM2102N) are of a high quality and a failure is not common if the chips are handeld properly therefore sockets are not manditory.

DIODES AND RESISTORS- Carefully bend the leads of the resistors and diodes and insert them in the board from the front. Solder and cut off excess leads.



TRANSISTORS- Insert transistors on to front side of the board, solder and cut off excess lead length.

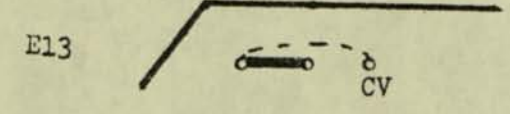
CRYSTAL-Insert the crystal, solder and cut off excess lead length. Be as careful when soldering the crystal down as you are when soldering other heat sensitive parts.

CAPACITORS- Insert on the front side of the board in the proper positions the capacitors. Be careful of polarity on those marked with a + lead on the PC board.

INTEGRATED CIRCUITS- Insert and solder down all 7400 and 8000 type chips. Where sockets were provided, just insert them into the socket. Be sure pin 1 is proprly aligned. Carefully remove the memory chips (MM2102N) and, one by one, insert and solder in place with a grounded soldering iron. Remove the SM3088L character generator from its packing material and insert it in the socket. Use extreme caution with this chip because it costs more than any other chip to replace.

OPTIONS

Composite Video- To provide composite video jumper the CV pad above E13 to the second pad to the left:

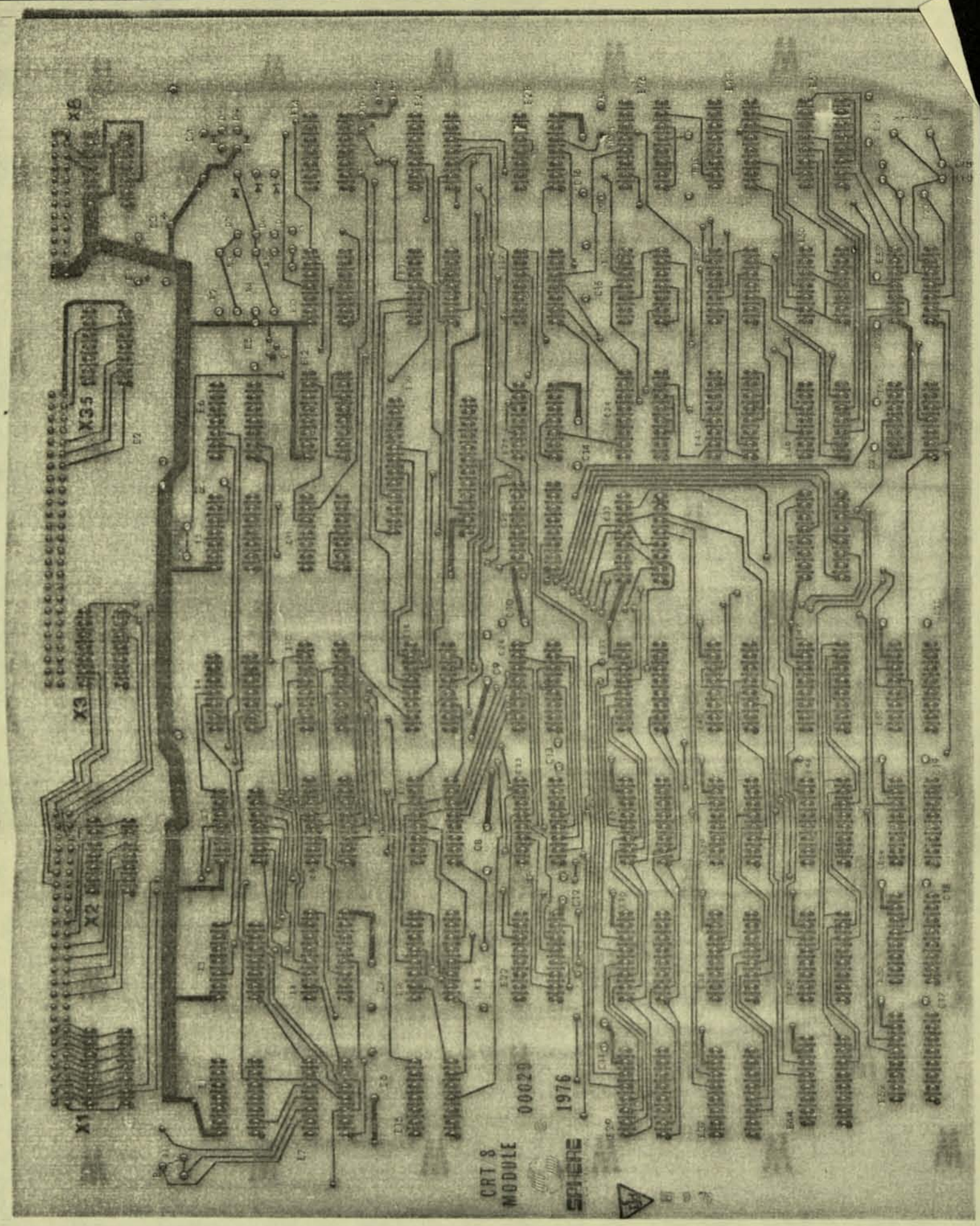


It is not necessary to cut free the middle pad if no connection is made at X35-3. The composite video signal is at X35-8 with ground at X35-7.

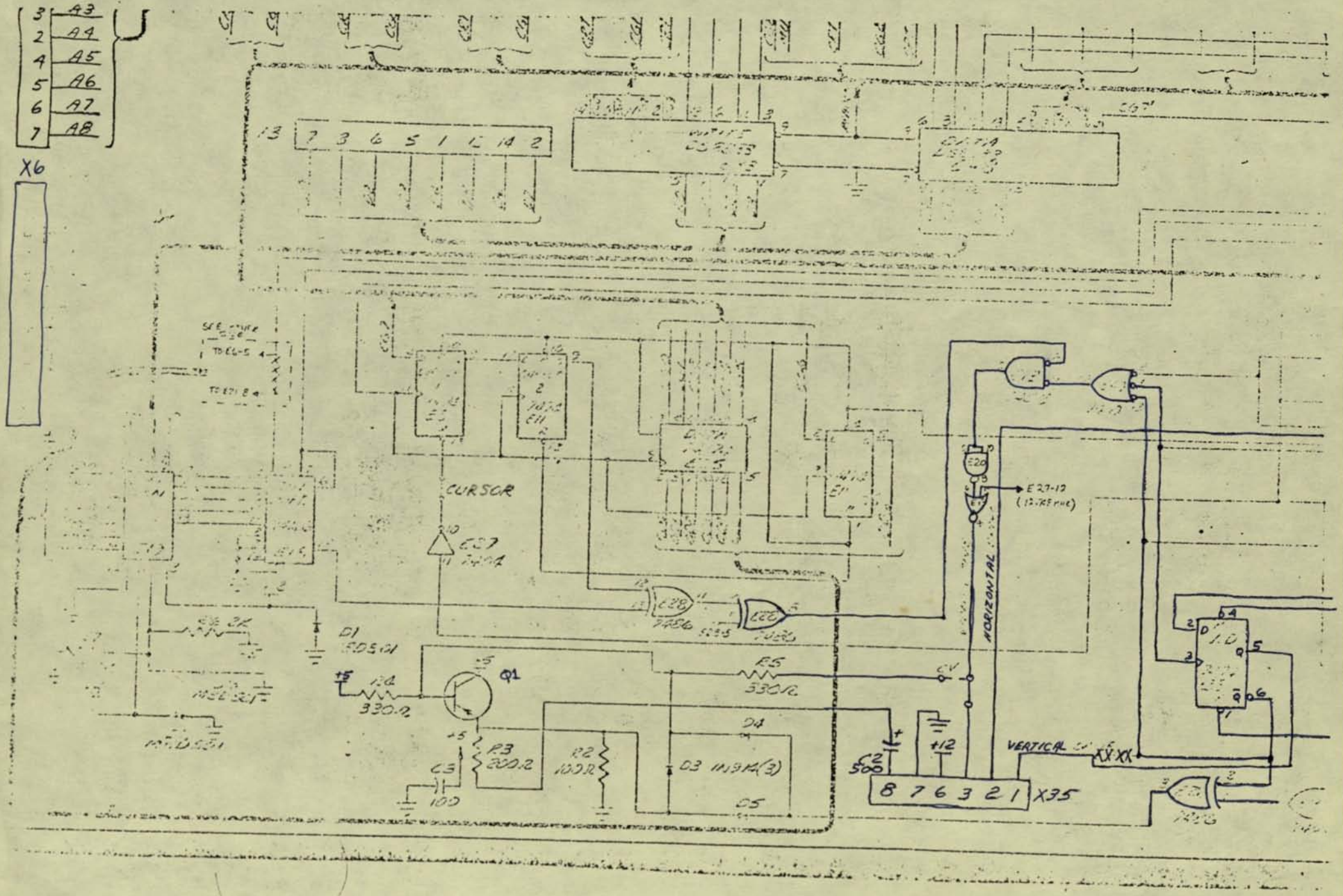
Reverse Video Screen- Remove R9 and jumper NOM V to RV to make the whole screen reverse video.

Curser- To produce a small underline as a cursor instead of a reverse video position, jumper the cursor location above E5.

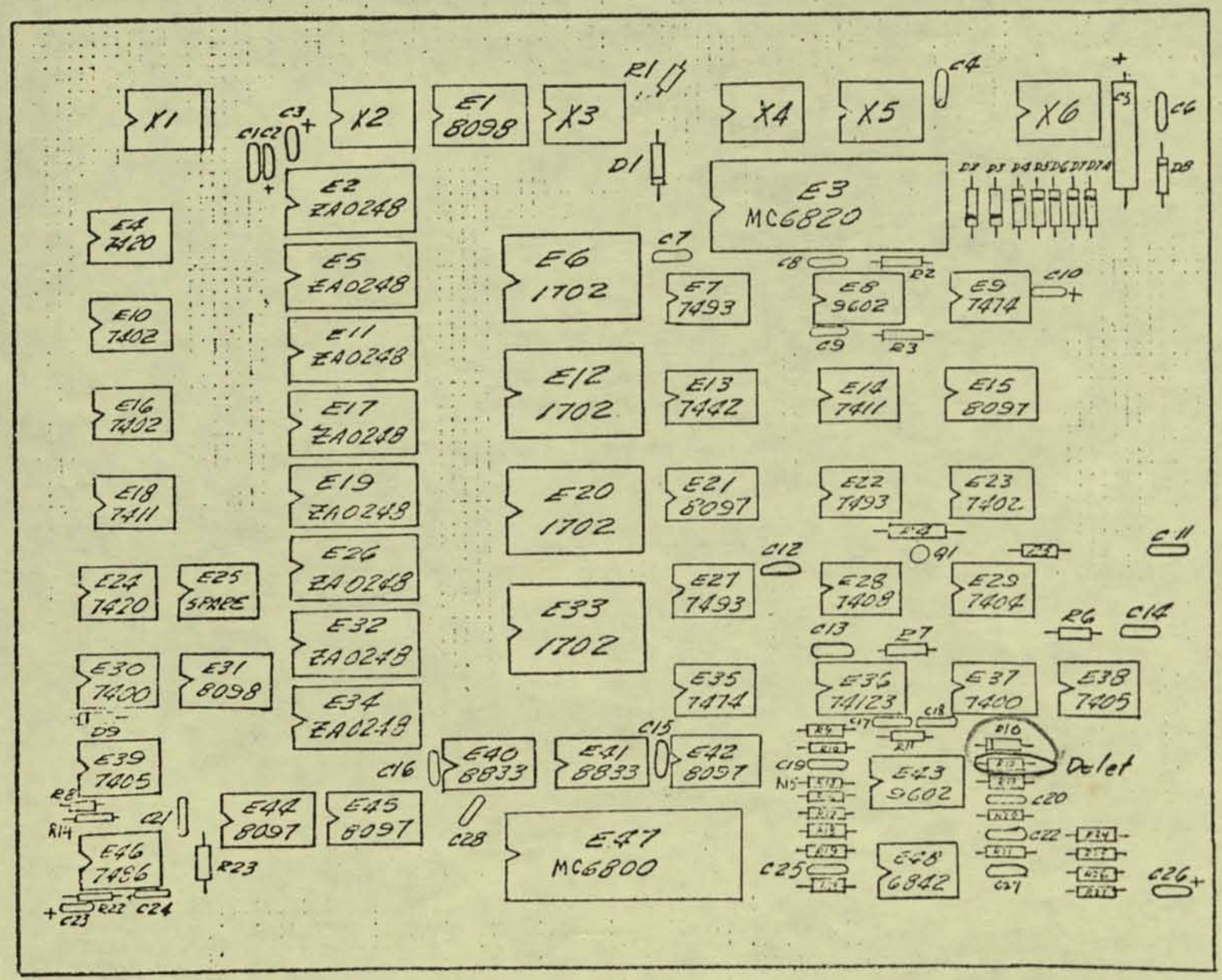
Second Board- To address a second CRT/8 board, cut & jumper A11 near E1.



CRT-8 front side


3 A3
2 A4
4 A5
5 A6
6 A7
7 A8
X6
CURSOR
HORIZONTAL
VERTICAL
+12
Q1
330Ω
330Ω
200Ω
100Ω
100
500
X35
8 7 6 3 2 1

CHKD. BY ____ DATE ____ SUBJECT PARTS LAYOUT SHEET NO ____ JOB NO. ____



E1, E31	8098
E2, E5, E11, E17, E19	
E26, E32, E34	2107
E3	6820
E4, E24	7420
E6, E12, E20, E33	1702
E7, E22, E27	7493
E8, E43	9602
E9, E35	7474
E10, E16, E28	7402
E13	7442
E14, E18	7411
E15, E21, E42, E45, E47	
	8097
E25	SPARE
E28	7408
E29	7404
E30, E37	7400
E36	74123
E38, E39	7405
E40, E41	8833
E46	7486
E44	6800
E48	6842
R1	130Δ
R2, R3	36 K
R4 2W	100Δ
R5, R6, R8, R15, ~~R16~~ R25	3.3 K
R7	6.8 K
R9	27.1K ~~8.2K~~
R10	11 K
R11	33 K
~~R12~~	~~30 K~~
R11, R22 ~~R24~~	10 K
R14, R25, R26	1 K
R16, R18	22 Δ
R17, R19	10 Δ
R20, R21	4.7 K
R24, R27	470Δ
~~R23~~	~~R27K~~
C1, C4, C6, C7, C8, C9	
C11, C12, C15, C16	
C21, C26, C28	.1 μF
C2, C3, C10, C23, C24	47 μF
C5 10V	100μF
C13	.01 μF
C14, C18	27
C	

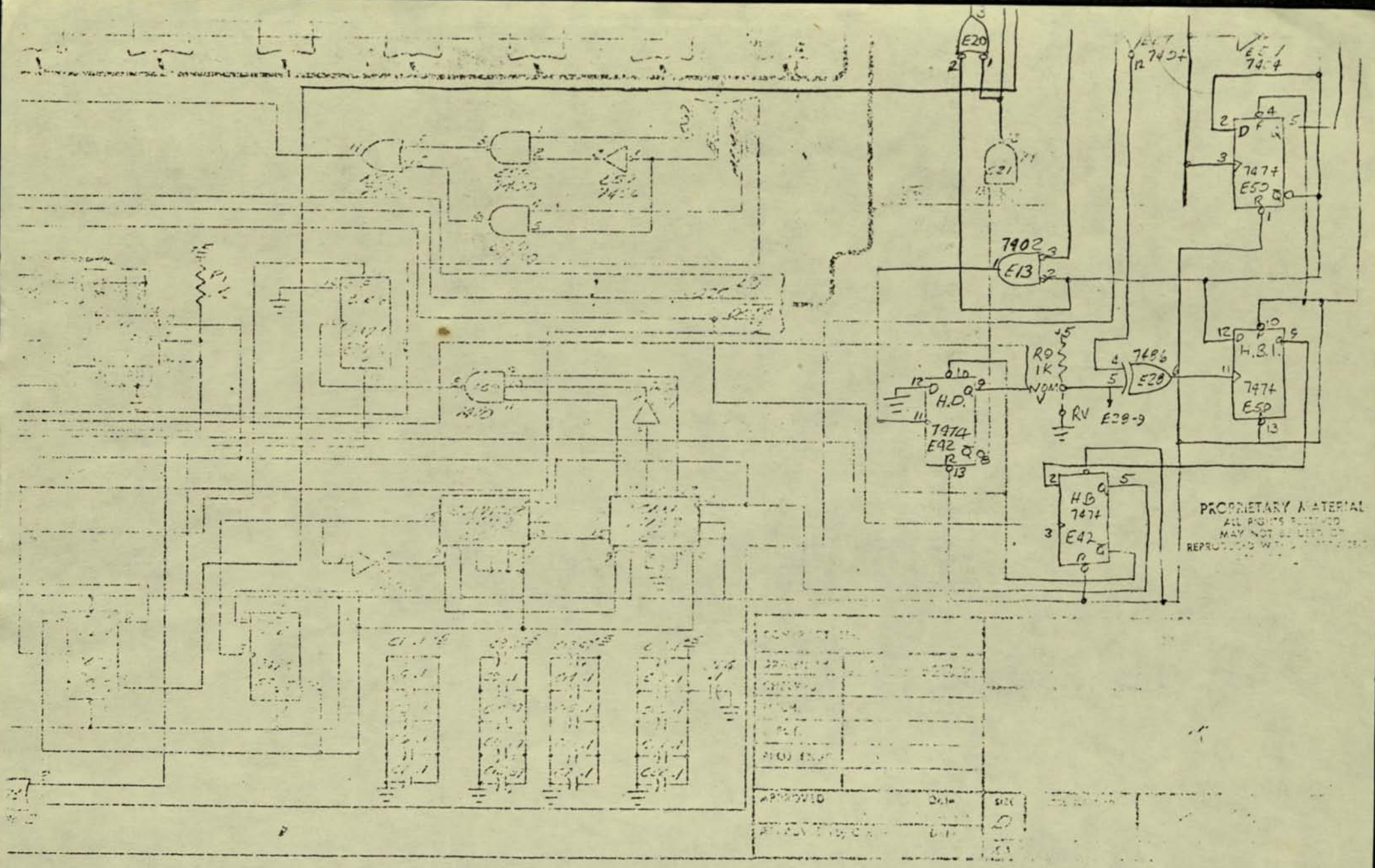
D1	1, 2 EA. IN5226
D2, D3, D4, D5	
D6	IN914
D7, D8, D9, D10	IN4001
Q1	2N2369A

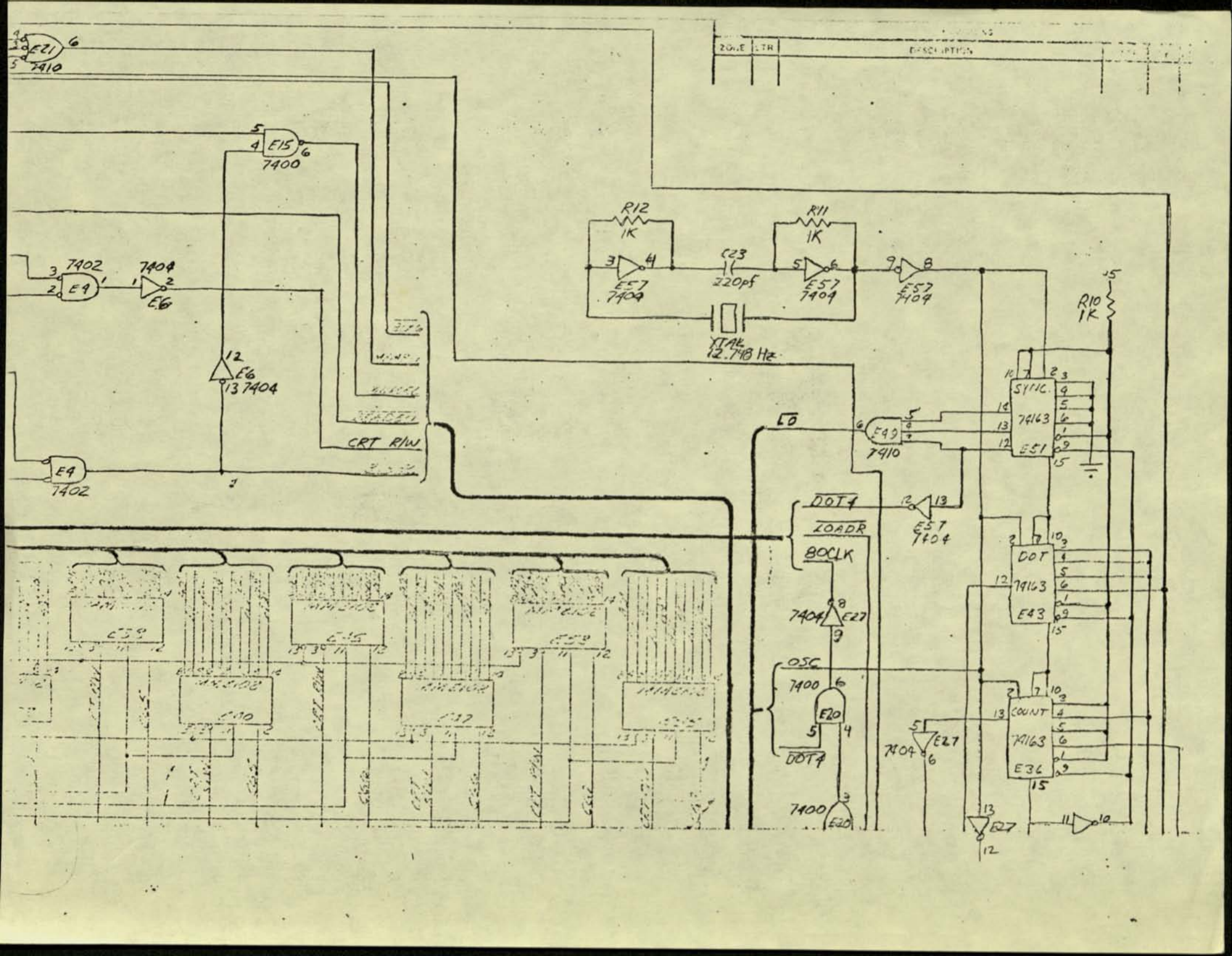
Darlon Left Side and Xerox For Layout.

Change to 15K

Ribb

SPIRE
701 SOUTH 500 WEST
BOUNTIFUL, UTAH 84010




ZONE
LTR
DESCRIPTION
E21
7410
E15
7400
7402
E9
7404
E6
R12
1K
R11
1K
C23
220pf
E57
7404
XTAL
12.748 Hz
R10
1K
CRT R/W
LD
E49
7410
SYNC
74163
E51
DOT4
LOADR
BOCLK
7404
E27
OSC
7400
E20
DOT
E43
COUNT
E36

ITEM NO	QTY REQD	NOMENCLATURE OR DESCRIPTION	IDENTIFYING OR PART NO.	SPECIFICATION	MATERIAL OR NOTE
35	REF	SCHEMATIC	000027		
34	1	24 PIN I.C. SOCKET	324-AG39D	X-E19	
33	5	14 PIN I.C. SOCKETS	314-AG39D	X1-X3, X6, X35	
32	1	TRANSISTOR	2N4400	Q1	
31	1	RESISTOR 3.3K, 1/4W, 5%	RC07GF332J	R7	
30	1	2K, 1/4W, 5%	RC07GF202J	R6	
29	2	330Ω, 1/4W, 5%	RC07GF330J	R4, R5	
28	1	200Ω, 1/4W, 5%	RC07GF200J	R3	
27	1	100Ω, 1/4W, 5%	RC07GF100J	R2	
26	6	RESISTOR 1K, 1/4W, 5%	RC07GF102J	R1, R8-R12	
25	3	DIODE	IN914	D3-D5	
24	3	DIODE	MBD501	D1, D2, D6	MOTOR
23	1	CRYSTAL	12.748MHz	XTAL	STAND CRYST
22	1	CAPACITOR 220pf 1KV	DD-221	C23	CRL
21	4	47μf 16V	EK47/16	C10-C13	ROE
20	1	100μf 10V	B41283 100/10	C3	SIEME
19	1	500μf 25V	TVA-1209	C2	SPRA
18	17	CAPACITOR .1μf 50V	CK-104	C1, 4-9, 14-22, 24	CRL
17	2	QUAD TRANSCEIVER	DS8833N	E33, E48	
16	—	BLANK		E14	
15	16	1024 BIT FULLY DECODED STATIC RAM	MM2102N	E29-32, 37-40 E44-47, 52-55	
14	1	QUAD 2-INPUT EXCL OR GATE	SN7486N	E28	
13	2	3-INPUT POS-NAND GATE	SN7410N	E21, E49	
12	1	CHARACTER GERERATOR	SCM3088L	E19	
11	1	SHIFT REGISTER	SN74166N	E18	
10	3	QUAD 2-INPUT POS-NAND GATE	SN7400N	E15, E20, E56	
9	3	HEX/QUAD FLIP-FLOP	SN74174N	E10, E24, E25	
8	9	4 BIT COUNTER	SN74163N	E9, 12, 17, 23, 34 36, 41, 43, 51	T.I
7	3	8CHAN. DIG. MULTIPLEXER	DM8123N	E8, E16, E22	
6	4	HEX INVERTER	SN7404N	E6, 7, 27, 57	
5	6	FLIP-FLOP	SN7474N	E5, E11, E26 E35, E42, E50	
4	2	QUAD 2-INPUT POS-NOR GATE	SN7402N	E4, E13	
3	1	DUAL 4-INPUT POS-NAND GATE	SN7420N	E2	
2	2	8-INPUT POS-NAND GATE	SN7430N	E1, E3	
1	1	PRINTED CIRCUIT BOARD	000028		SPH

LIST OF MATERIAL OR PARTS

ISE SPECIFIED

CONTRACT No. OCT 27 1976

SOHEQE