

206

PACKET NUMBER 1

EXTENSIONS AND MODIFICATIONS  
TO THE MARK-8

THE DIGITAL GROUP &  
DR. ROBERT SUDING WØLMD

*SPRINT MARCH 1946*

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
Preface	i
Rights Reservation and Disclaimer	ii
Section I - Hardware Narratives	
Dr. Suding's Modification Narrative	I-1
Digital Group Packet #1 Notes	I-5
Circuit Board Modification Procedure	I-6
Section II - Schematics	
Input Port Extensions	II-1
Output Port Extensions	II-2
Address Latch Modifications	II-3
Octal Keyboard & 7-Segment Octal Display	II-4
Power Supply & 8223 ROM Circuit	II-5
TV Typewriter	II-6
TV Typewriter Memory	II-7
Extended (128-character) ASCII Encoder	II-8
Cassette Interface	II-9
Backplane Wiring Diagram	II-10
Section III - Parts Lists	
Detailed Parts Lists	III-1
Summarized Parts List	III-2
Section IV - Software Documentation	
Narrative	IV-1
Restart to Programs	IV-3
Memory Clear	IV-4
Memory Checker	IV-5
Bit Reverse	IV-6
TV Character Generator Test	IV-7
Keyboard to Memory	IV-8
Number Sorting	IV-9
Running TV Display	IV-10
TV Character Demonstration	IV-12
Keyboard to TV	IV-13
TV Storage Dump	IV-14
Cassette Dumper for Cold Start	IV-16
TV Subroutines	IV-17
Cassette Loader	IV-18
8223 ROM Programming for Cassette Dumper	IV-19
Appendices	
Universal Order Form Master	
Blank Coding Sheet Masters	
Character Generator Codes	

## PREFACE

What we and Dr. Suding have tried to accomplish with this packet is provide:

1. Significant hardware modifications for usability
2. A small 1K operating system to help you get started
3. A reliable cassette medium to allow loading of all software in one step
4. As many comments, recommendations, etcetera that we thought might also help people get going.

If you use a point-to-point wiring approach, you should be able to build directly from the schematics as pin number callouts are included. Please do it slowly and carefully, as this method is very error-prone.

We have tried to be as accurate as possible in reproducing Dr. Suding's documentation but there are bound to be some errors or unclear areas. As you discover the errors or make improvements, please let us know so we can pass them on to others and update future copies. If you have questions on the packet, send them to us and we will try to answer them as quickly as possible or refer them to Dr. Suding. Dr. Suding requests that you write him through us rather than directly as he is not set up to handle any volume of mail and we are.

RIGHTS RESERVATION AND DISCLAIMER

The digital group and its contributing authors reserve all rights to any commercial use of the material herein. All reproduction in any media by any method for any purpose is specifically prohibited without written permission of both the digital group and the contributing author.

The digital group assumes no liability or responsibility for the accuracy of the contents or the infringement on other copyrights or patents by its contributing authors. The digital group also does not represent that the use of the circuits contained herein is free from patent infringement.

## MARK-8 ENHANCEMENTS

Some have claimed that the Mark-8 Microcomputer is a toy. It definitely isn't. It is almost unlimited in potential applications. However, its original capabilities can be vastly enhanced. This pack of information and schematics presents the modifications and enhancements made to the original Mark-8 as well as some additional units.

Several enhancement categories have been investigated and implemented. First, improved abilities to get data and software in and out of the Mark-8 were investigated. Next, better methods of displaying the software and data handling hardware were designed. Finally, software was written which would provide effective support of these enhancements, as well as provide a core of operations subroutines.

The initial modification was the use of plug-in sockets in place of those board-to-board wires which were breaking after many fold-opens. I had some old 47-pin second generation computer cards and sockets. After cutting off the top part of each card, leaving only about 1" of card bearing the pins, I epoxied these 47-pin stumps to each original microprocessor card. #30 Teflon wire was used to connect the new pins to the old lines, maintaining the same numbering sequence. Since the output and the input card have 7 to 8 ports, each port containing 8 bits, a dual set was used on these cards. These dual sets were made by bolting a second 47-pin stub to the original, spaced the socket spacing away. The input and output cards now have 94 pins, eliminating the need for those clumsy Molex connectors. The extra pin set is noted by an "x" following the pin number on the schematics.

The output card was modified by adding another 7402 and six 7475's in the unused card areas near the pin end of the card. This provided a total of seven 8-bit output ports from this card. The eighth output port (Port 7) is included on the LED readout pegboard. A similar number of ports extension on the input card was implemented.

A pegboard mounted set of twelve 7-segment LED's (MAN-1) and 7447 drivers is mounted behind the window formerly occupied by the LED lamp boards. The six address LED's are wired for leading zero blanking except for the least significant LED. The three memory LED's do not have leading zero blanking. Output port 7 is now used for the front panel output port instead of the original port 0. The port 7 LED's are wired for complete leading zero blanking so that a 000 can be outputted to port 7 and completely blanked to reduce readout clutter.

A surplus desk top style 16-key calculator keyboard was purchased for key input of functions and data. Nine of the keys were assembled in a 3 x 3 block for Octal code entry. The rightmost column has a 1, 2, and 4 key, corresponding to the bit value of the least

significant digit of the Octal number. The middle column also has a 1, 2, and 4 key, corresponding to the bit value of the middle digit. The leftmost column contains a 1 & 2 key with a common clear key at the bottom. Operation of this key set consists of punching the needed bits while watching the accumulating value in three 7-segment LED readouts above the keyblock. Eight R-S flip-flops made from four 7400's hold the value as entered. After a given value is utilized to load address or data, the CLEAR key is pressed to reset the eight flipflops to 000 Octal. Three more keyswitches are mounted to the left of the Octal number block. These three are INTERRUPT, EXAMINE, and DEPOSIT, reading from top to bottom. To the right of the Octal number block are four more keyswitches: LOAD HIGH ADDRESS, LOAD LOW ADDRESS, RUN, and SINGLE STEP. Notice that the JAM switch is no longer used since JAM is logically generated in the switch pulse forming circuitry.

The address latch card has been radically changed due to the use of the very "bouncy" SPST keyswitches. Everything except the 74193's was stripped off the address latch board. The new circuitry shown on the address latch schematic was then added in the card area immediately above the 74193's. The 1/3 of the card to the side of 74193's is used for cassette interface logic now.

The cassette interface consists of four IC's. A 566 voltage controlled oscillator (VCO) is used to convert the 0 and 1 output of the least significant bit of output Port 1 to a frequency shift output. A 1 output gives 2125 HZ output and 0 output results in 2975 HZ from the VCO. The triangular wave output of the VCO is fed through a 47K resistor to the microphone input of a \$30 cassette recorder. A special program, LOAD CASSETTE (included), serializes the 8-bit parallel bytes and outputs them to Port 1 asynchronously at a rate of 1K/25 seconds. The receive portion of the cassette interface consists of a 741 limiter, a 5558 dual band-pass active filter for 2125 and 2975 HZ, and 741 output comparator giving the TTL compatible serial output to the least significant bit of input Port 1. After attaching the limiter's input to the phone jack output of my cassette, I start my recorder and a special deserializer program called DUMP CASSETTE (included). The cassette then loads the prerecorded operating system and operational program into storage. The cassette interface circuits were designed to operate on standard Teletype frequencies which allows them to be also used for sending and receiving Teletype signals. Since the programming establishes the encoding and decoding, almost any 850 HZ shift RTTY station can be copied, whether ASCII, BAUDOT, etc., at any speed.

Finally, a special TV readout circuit was built on the former LED board after completely stripping off the previous circuitry. This TV readout is built around a special character generator, the MCM 6571L from Motorola, which gives both upper and lower case alpha characters, as well as numbers, special characters, math symbols, and even the Greek alphabet. Cost is \$27.50 from a distributor, and the characters are generated with a 7 x 12 dot

matrix format vastly easier to read than the 5 x 7 upper case-only R & E Typewriters. The remaining IC's should cost about \$32.00 at current surplus prices. The character memories are seven 1101's which result in 8 lines with 32 characters in each line. While this is less than the R & E TV Typewriter 1's 512 characters, it loads the screen apparently instantaneously, instead of taking 17 seconds. The circuit is much simpler, and the large line-to-line spacing is great for keeping the characters apart in such things as storage dumps, etc.

The TV readout has a few unusual features. All output is via a single output port, in my case, Port 6. The most significant bit is used as a data strobe line. Bits 0 - 6 define which of the 128 possible characters are to be printed, with one important exception. The solid block character (■) consists of all ones. I don't need this character, so when a bit pattern of all ones (including the strobe line) comes from Port 6, the 7430 detects this, and presets the 74193 address counter. The next character entry is then loaded at the upperleftmost screen position controlled by the RAM's. Each character will increment the address counter when entered, also eliminating the need for another port for address output. If you need the capability of random addressing, an output port could be used instead of the 74193 address latches.

My keyboard uses a re-encoded Microswitch keyboard with Hal Effect keyswitches, giving the 128 potential characters. Other 64-character ASCII keyboards could be used if special shifting key(s) and logic are added.

The power supply should have considerably greater potential on the +5 supply when running these modifications. Being rather conservative, and hating to blow CPU chips, I have built in a supply which can easily supply the required voltages. The +5 pass transistor dissipates considerable heat, so use as large a heat sink as possible.

Packaging is inside of a CO-1 LMB cabinet which is horrible overpriced, but looks nice. The output ports are available through some 16-pin second generation IBM computer card sockets (SMS), although 17 or slightly more pins would be better since at least a ground pin is generally needed. I use a separate banana plug for common grounding.

Several basic software routines were required to support my enhanced Mark-8. First, routines for loading and dumping the cassettes were required. Since the data are stored serially on the cassettes, the loading and dumping routines were designed as parallel/serial and serial/parallel converters. Several short subroutines were designed to support the TV readout hardware by a "Erase/Home" subroutine, "Space Over" subroutine, and a "Write Long Character String" subroutine. Basic operational routines have been written

such as a "TV Storage Dump" which displays the memory contents in Octal code.

After evaluating the enclosed circuits and hardware, you can pick those improvements which would seem most worthwhile yourself. You may not feel any need for the modified front panel as well as the TV readout and keyboard entry. You may have already built the TVT circuitry and wish to use it instead of the enclosed circuitry. I would strongly recommend that you build the cassette hardware and software, however.

Some port reassignments have taken place. Port 0, input and output, is now dedicated to the keyboard, although presently only port 0 input is used. Port 1, input and output, is dedicated to serial devices, presently using only bit 0 for the cassette recorder's interface. Ports 2, 3, 4, and 5 are brought out to the rear for external usage. Port 6 output is used for interfacing to the TV readout board. Port 7 output is used to drive a set of 7-segment readouts on the front panel in place of the Mark-8 use of port 0 output.

Dr. Robert Suding



## THE DIGITAL GROUP PACKET #1 NARRATIVE

We'd like to recommend a way to get started implementing Dr. Suding's or your own modifications. First and foremost, make all the boards pluggable. To do this, find a surplus set of PC boards and sockets or a card rack containing cards and connectors. Or buy a new set of PC card headers and connectors. The more pins the better. You'll need about 10 sets. (47-pin double-tier Elco/Varicon sockets and cards are ideal as they are very solid and self-supporting in any plane, but are very expensive new.) Crop off the connectors that are on the surplus card leaving about an extra 1" of card length behind the connector. Bolt the card connectors onto the Mark-8 boards (after stripping off appropriate circuitry). If you can't find 47-pin or larger sockets and connectors (a very likely possibility) and don't want to buy new ones (see note #1), then use 2 22-pin connectors and sockets or equivalent per Mark-8 card. Be aware that the input and output port cards will then require 4 22-pin connectors so physical mounting gets a little bit tricky.

Attach connecting wires between the old pads and new connectors on the boards in the same relative position as on the old board. Next, wire up the octal display and front panel keyboard which will replace the front panel and LED display board. (Please note that the front panel keyboard also requires the modifications to the address latch board for proper operation.) You might want the keyboard and display on a separate module with about a 2-foot cable plugging into the new Mark-8.

Build an extender card. Either convert one of your surplus cards or use a piece of flat metal, rods, bars or whatever for the mechanical mounting of the plug and socket and then interconnect with wires. Then wire up the cassette interface on the address latch card and, after debugging, you're ready for some software. At this point you're pretty free to choose whichever modification or extension you wish to implement next. Our choice of a typical implementation sequence is as follows:

1. TV Typewriter (including full keyboard) - Dr. Suding's or equivalent
2. Output port extensions
3. Input port extensions
4. Typewriter/teletype printer
5. Cassette tape drives

Note #1--If you do want to buy new ones, drop us a SASE for some more ideas on it--there are several ways to go.

## MODIFYING THE BOARDS

In order to reclaim space on the PC board, you need to strip off the old circuitry. The recommended method:

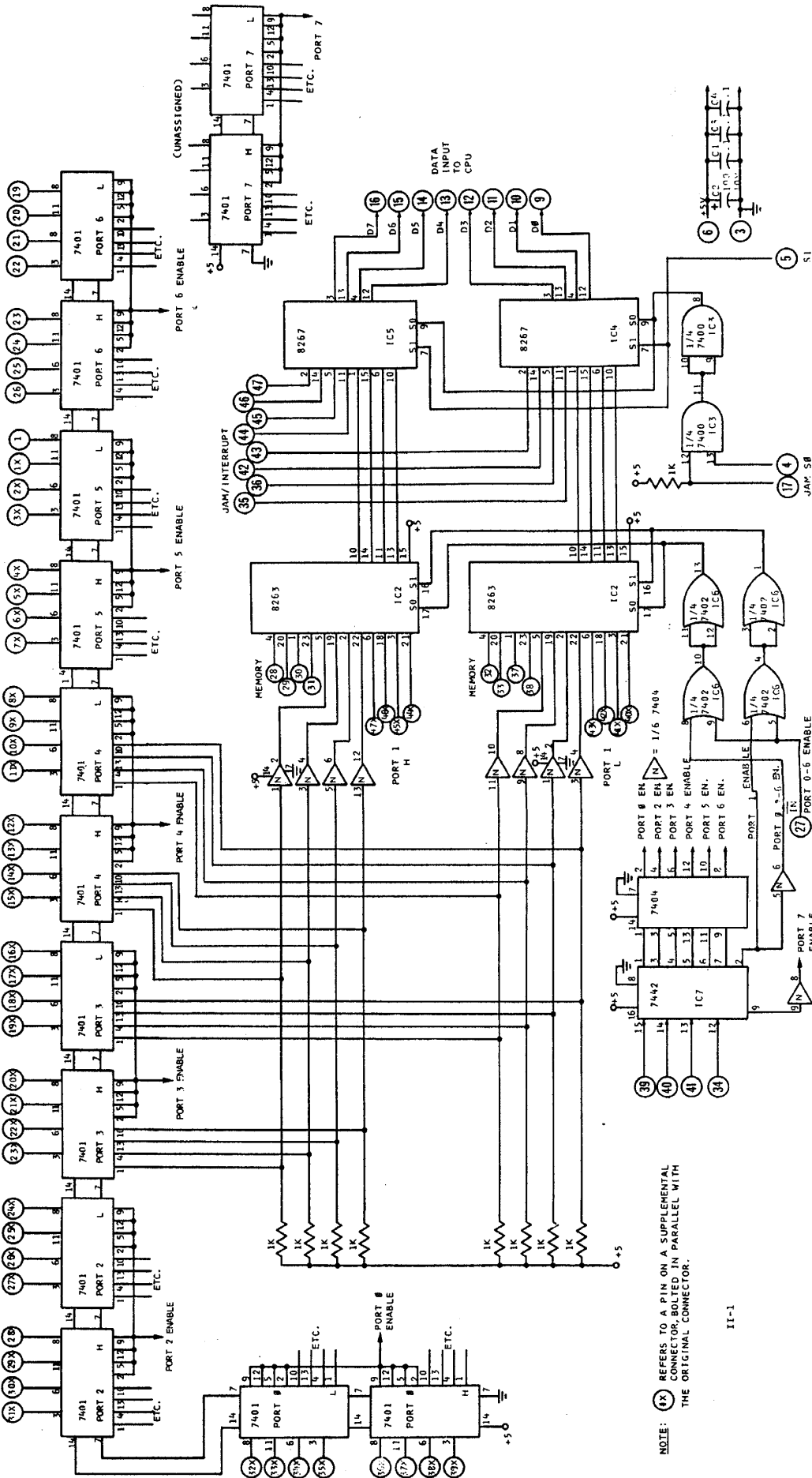
1. If you need to remove only part of a line, cut it at the stopping point with an x-acto knife.
2. Heat the end of the circuit line (usually at a pad) with a small or medium wattage soldering iron and flick up the end of the PC-circuit line with an x-acto knife. What you are trying to do is create what's known as a foil separation.
3. Grab the raised end of the circuit with a pair of needle-nose pliers and pull gently. The line should come away easily and cleanly until you reach the breakpoint you cut in #1 (if any).
4. Continue in this manner until you reclaim all the desired area on both sides of the board.

Building a circuit on the board using a point-to-point approach is as follows:

Lay out a pattern. Drill a 14- or 16-pin DIP pattern with a small drill bit for each IC. Drill appropriate holes near the IC for components. Install the ICs and components in the same direction (front to back) as the other ICs that are already on the board. On the bottom, bend the pins slightly outward to retain the component. Interconnect the ICs and components using thin wire (#30 works well) and soldering directly to the IC pins. Then inter-connect (tie to IC pins on bottom) the circuit to the remaining parts of the old circuit (if any) and the circuit board connector as required. Attach to the old soldered ICs on the bottom of the board via their pins.

An alternative approach would be to wire up the new circuit on Vero or Vectorboard and bolt it to the old board and then connect it to the old circuit and the circuit board connector. This approach is not as dense in terms of component packing that can be achieved with the first approach but can sometimes be easier and quicker.

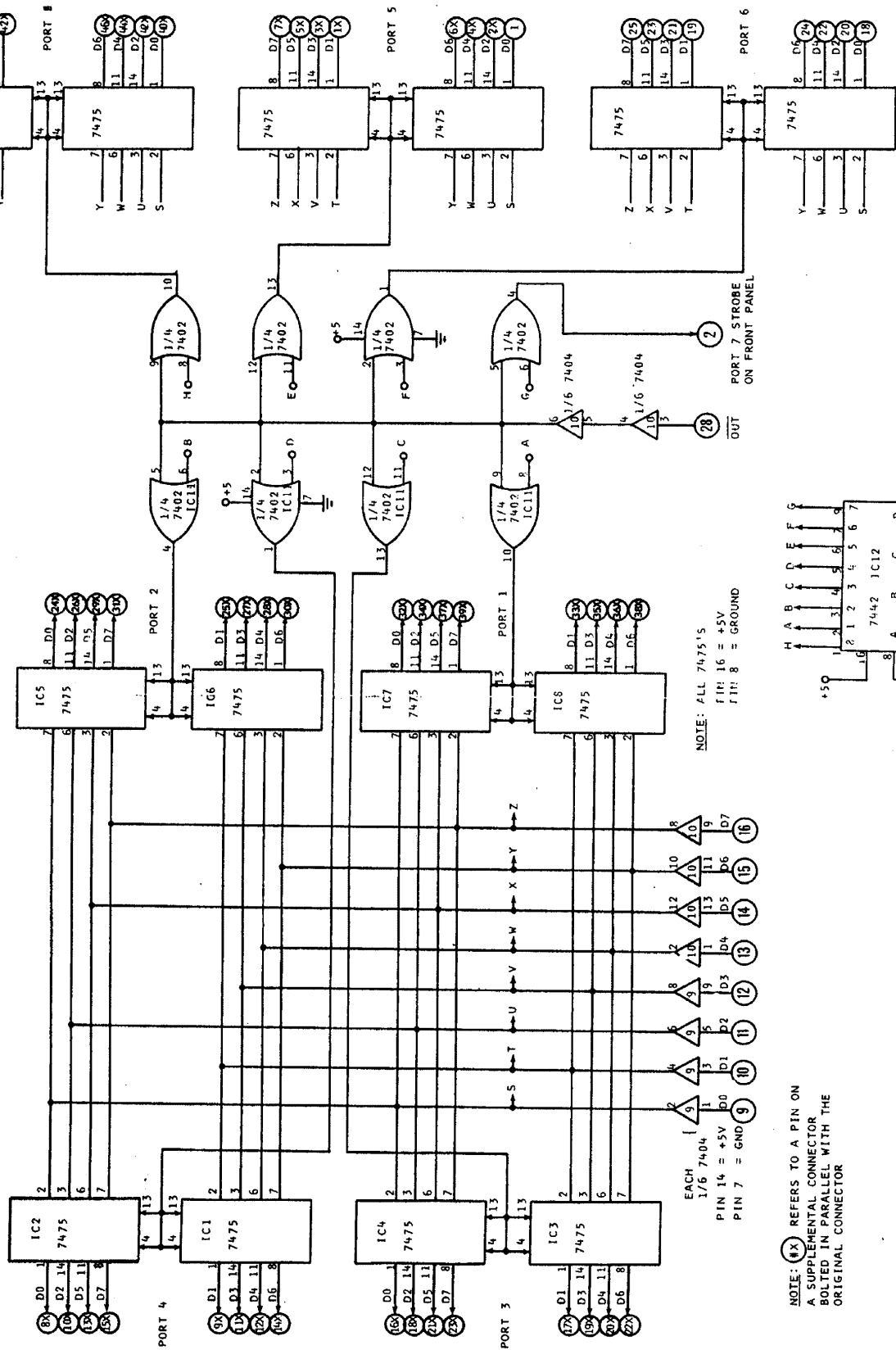
EXTENDED INPUT MULTIPLEXER



NOTE: (X) REFERS TO A PIN ON A SUPPLEMENTAL CONNECTOR, BOLTED IN PARALLEL WITH THE ORIGINAL CONNECTOR.

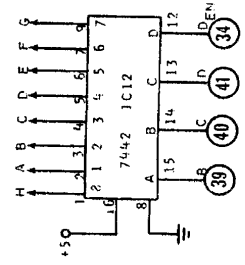
ADDED TO ORIGINAL CARD

EXTENDED OUTPUT LATCH CIRCUIT

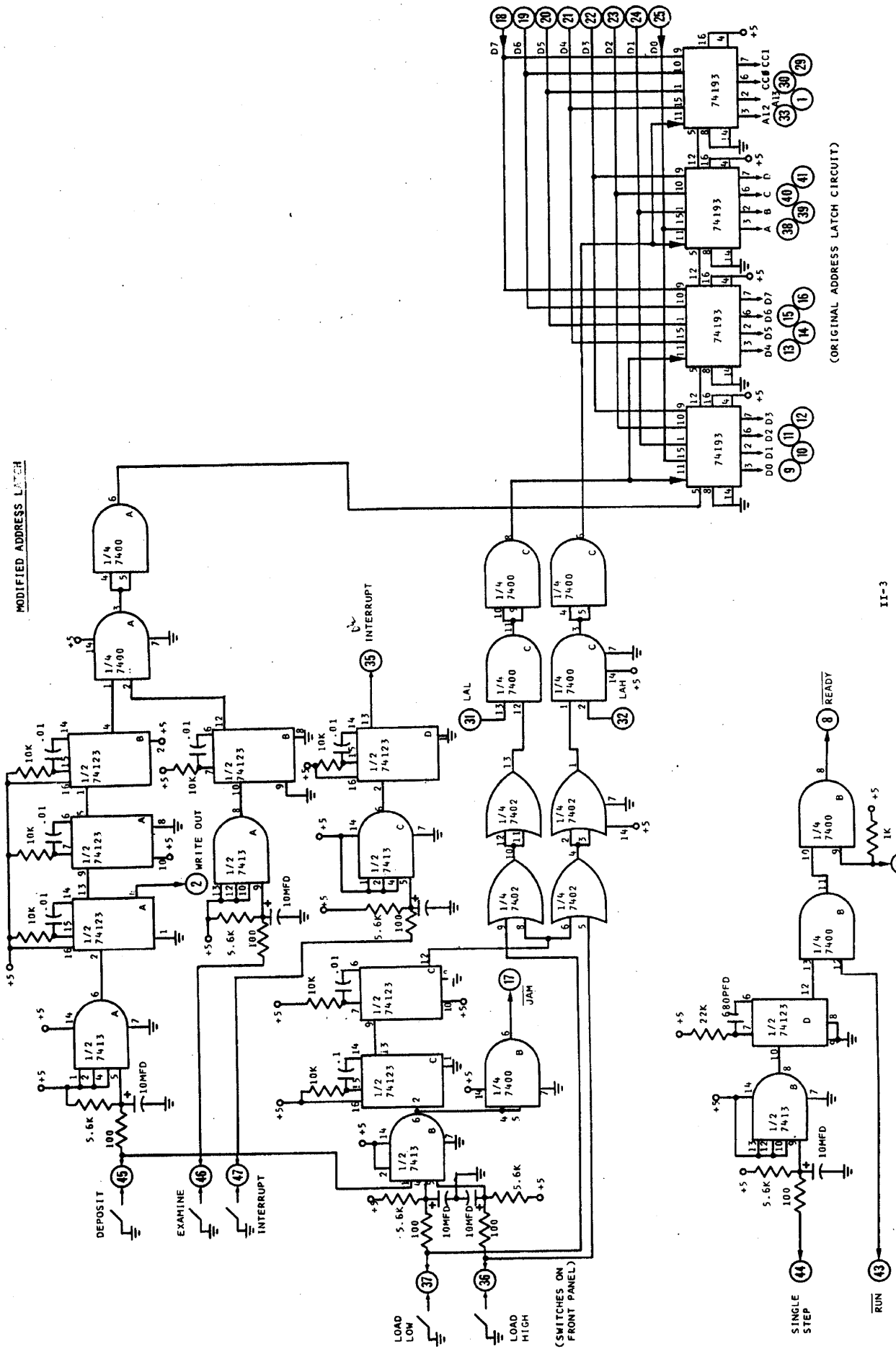


NOTE: ALL 7475'S  
 PIN 16 = +5V  
 PIN 7 = GND

NOTE: (X) REFERS TO A PIN ON  
 A SUPPLEMENTAL CONNECTOR  
 BOLTED IN PARALLEL WITH THE  
 ORIGINAL CONNECTOR

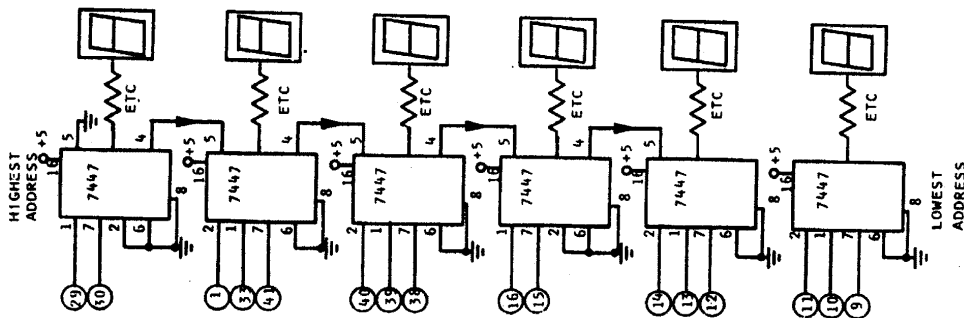


MODIFIED ADDRESS LATCH

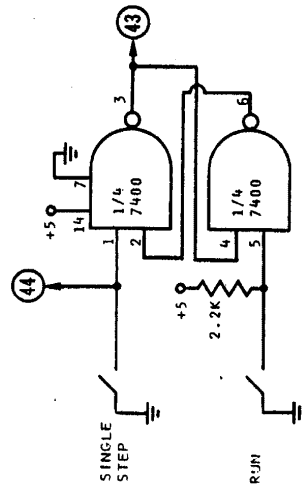
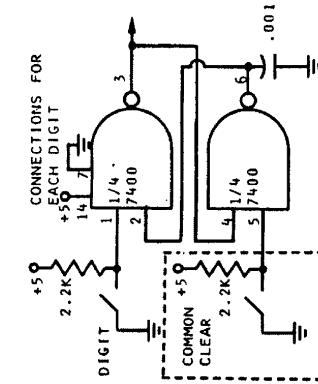
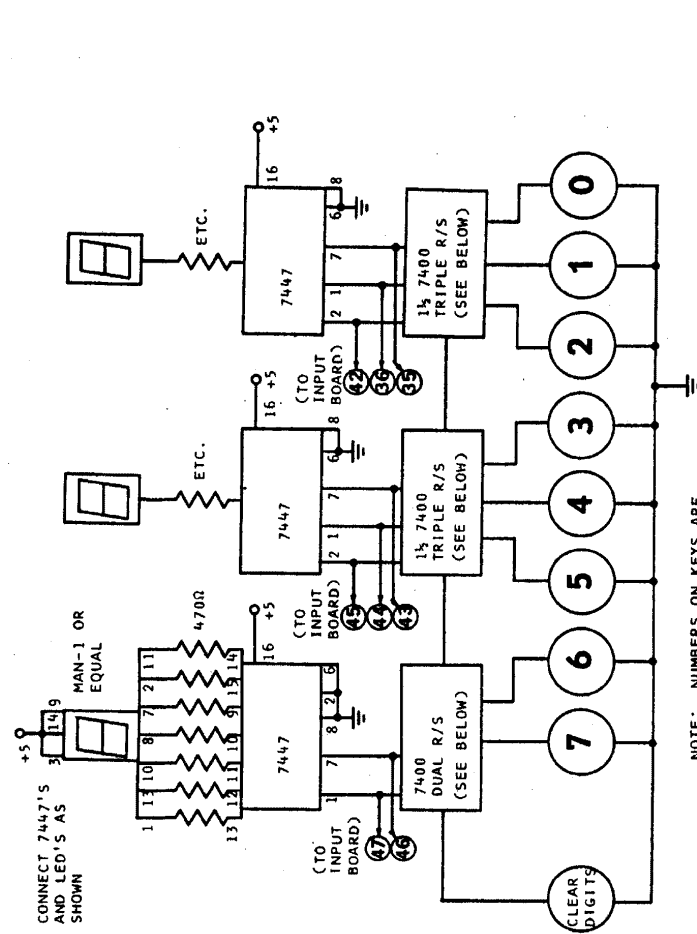
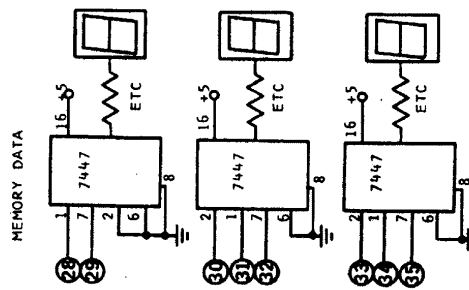
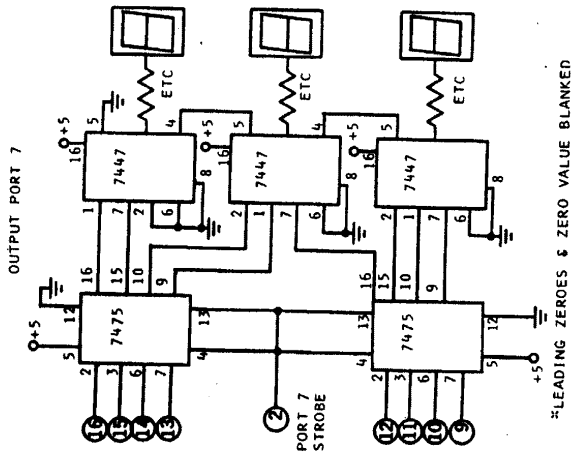


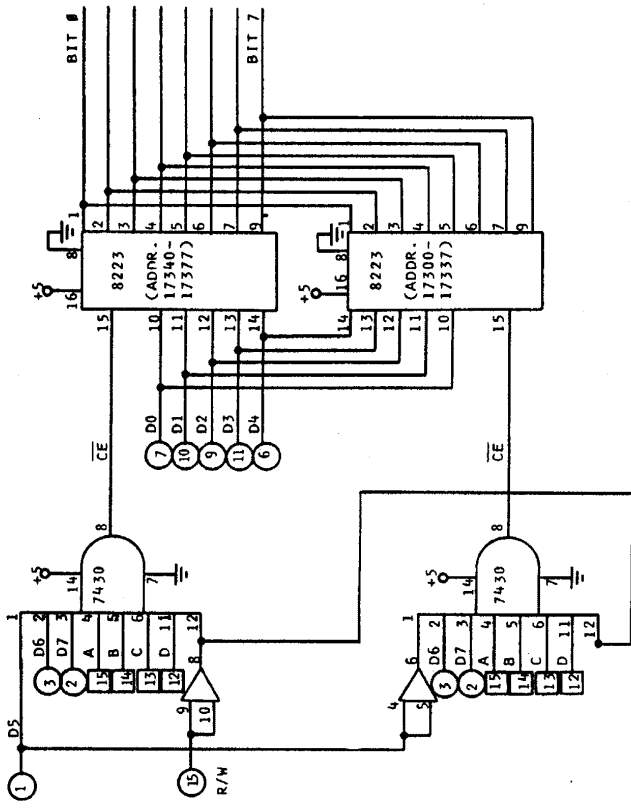
(ORIGINAL ADDRESS LATCH CIRCUIT)

EXTERNAL READY LINE



FRONT PANEL OCTAL DISPLAY AND OCTAL KEYBOARD



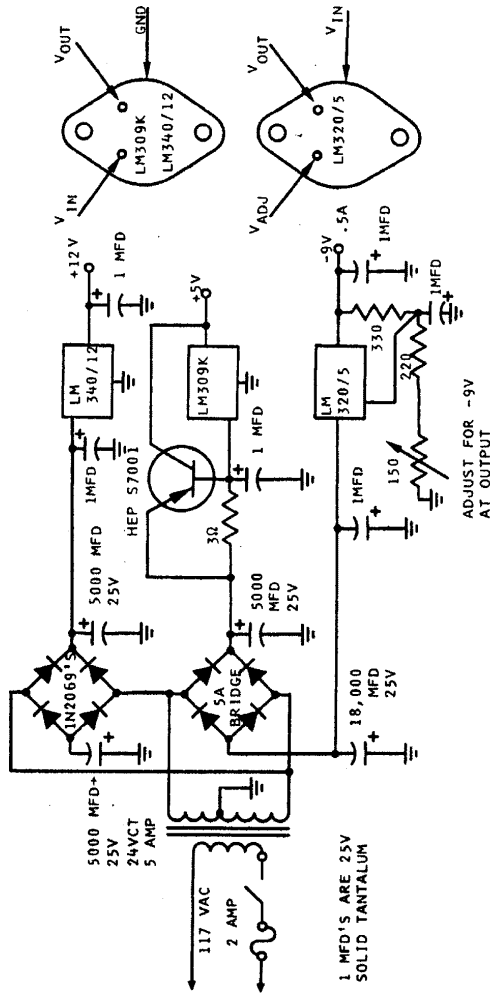


8223 ROM CIRCUIT ON 1101 MEMORY BOARD

R/W & D8 TO D7 CIRCLED NUMBERS ARE 1101 PINS.

A B C D SQUARED NUMBERS ARE 7442 PINS.

THE INVERTERS SHOWN ARE TWO SECTIONS OF IC14 ON MEMORY BOARD 1. THE 1/4 7400 SECTION USING PINS 8, 9, & 10 IS CUT LOOSE FROM ITS PRESENTLY USED PATTERN AND WIRED AS SHOWN.



POWER SUPPLY

ADJUST FOR -9V AT OUTPUT

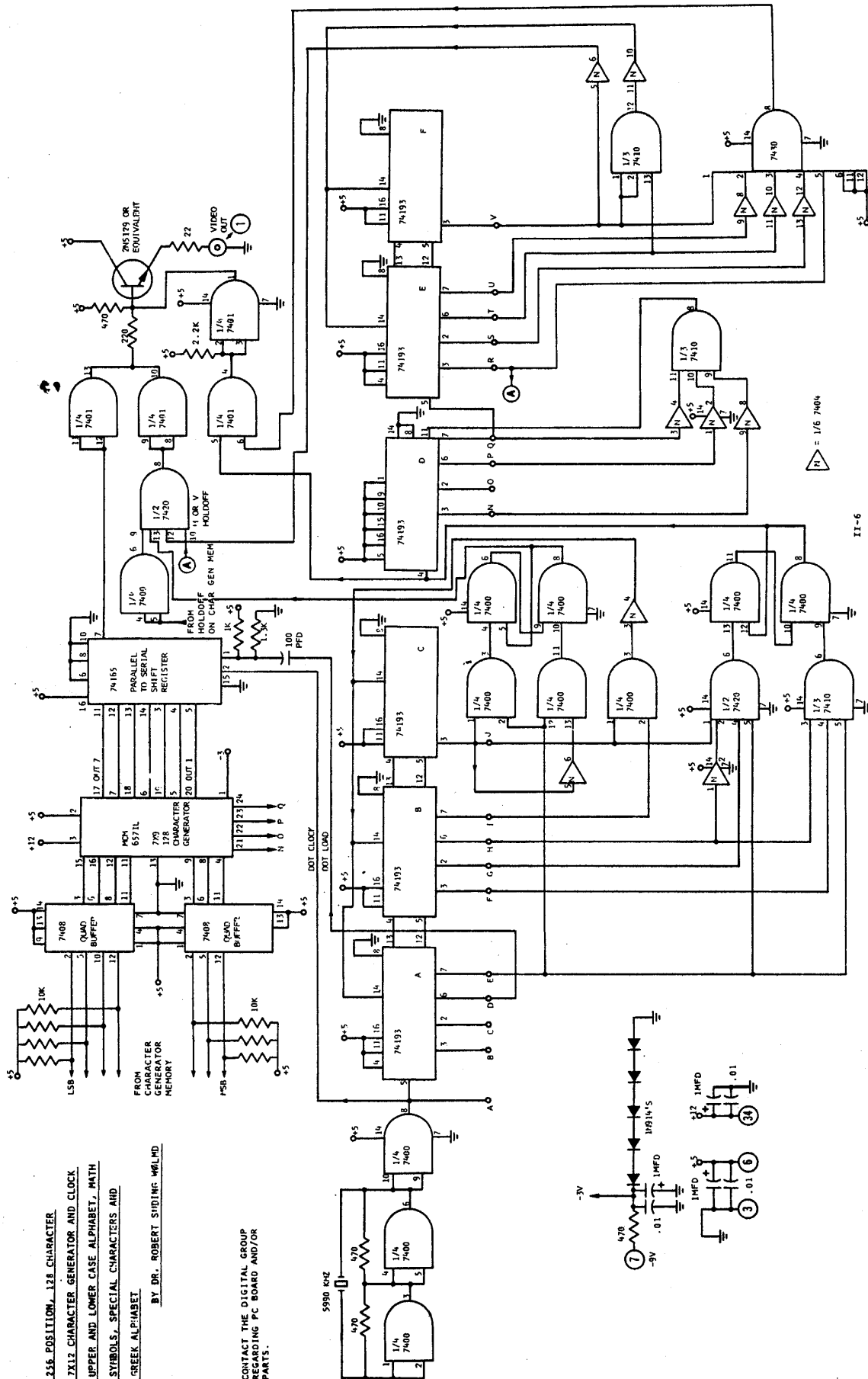
1 MFD'S ARE 25V SOLID TANTALUM

**258 POSITION, 128 CHARACTER**

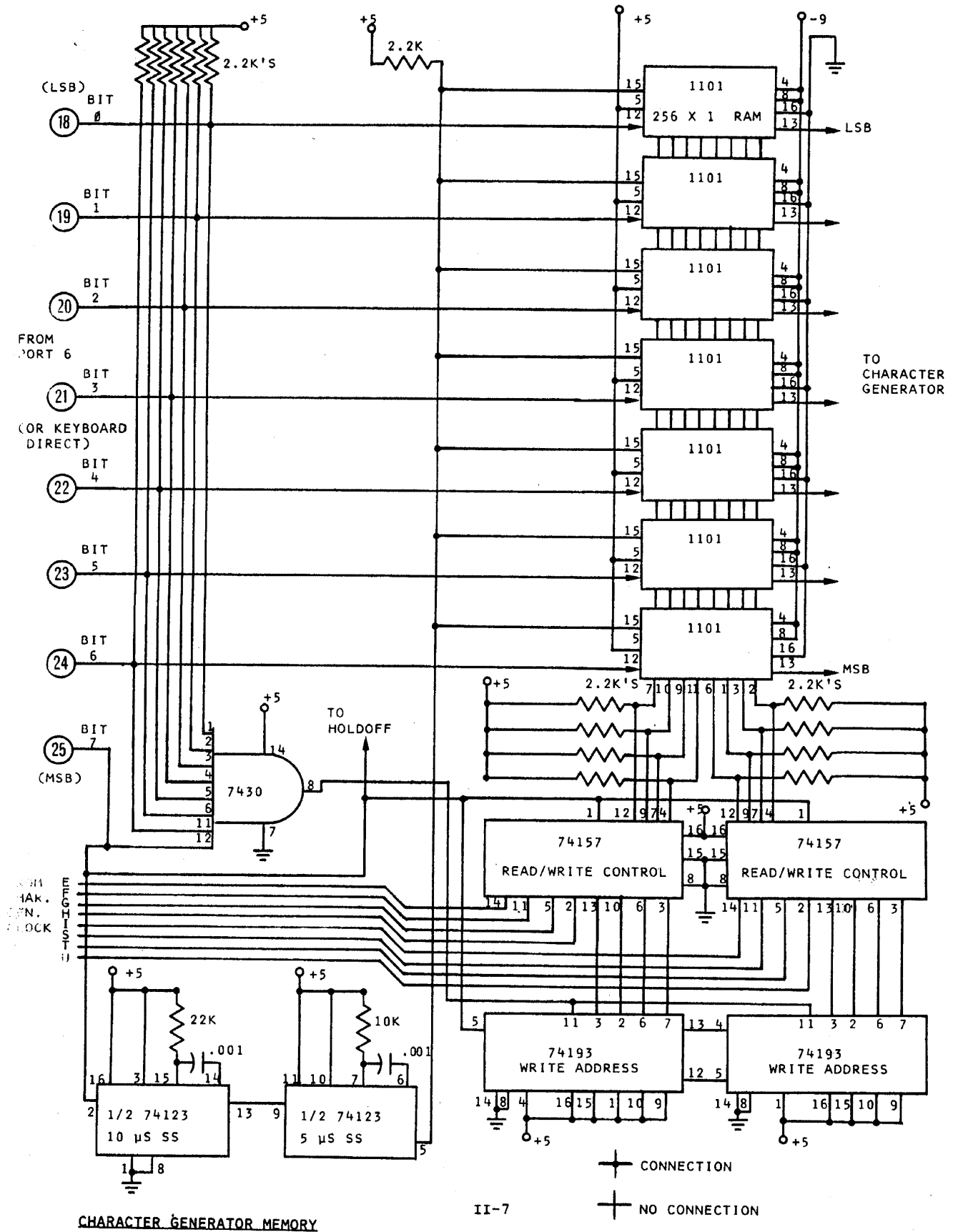
**7X12 CHARACTER GENERATOR AND CLOCK**  
**UPPER AND LOWER CASE ALPHABET, MATH**  
**SYMBOLS, SPECIAL CHARACTERS AND**  
**GREEK ALPHABET**

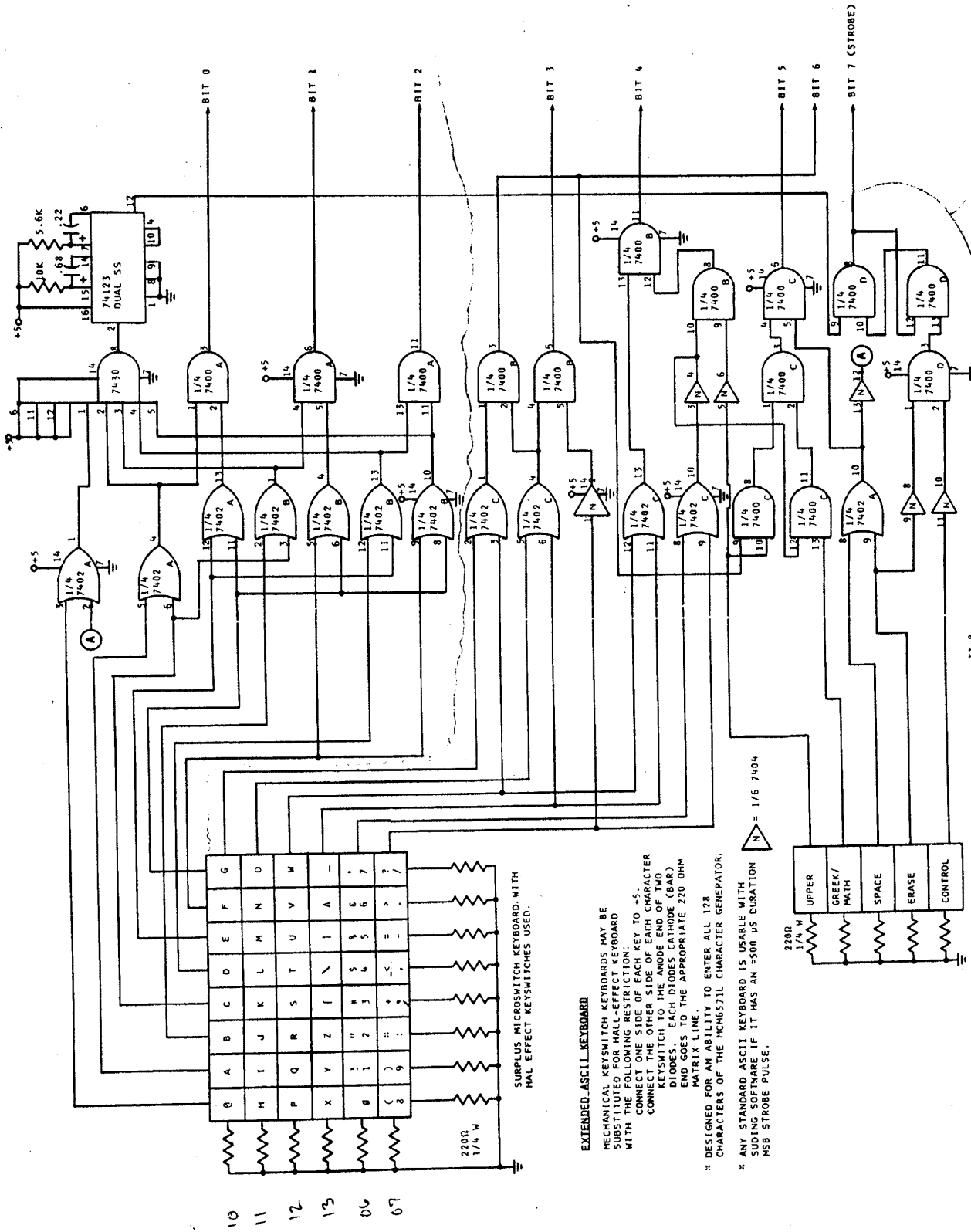
BY DR. ROBERT SIDING WJLHD

CONTACT THE DIGITAL GROUP  
 REGARDING PC BOARD AND/OR  
 PARTS.









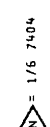
LEDs out are done

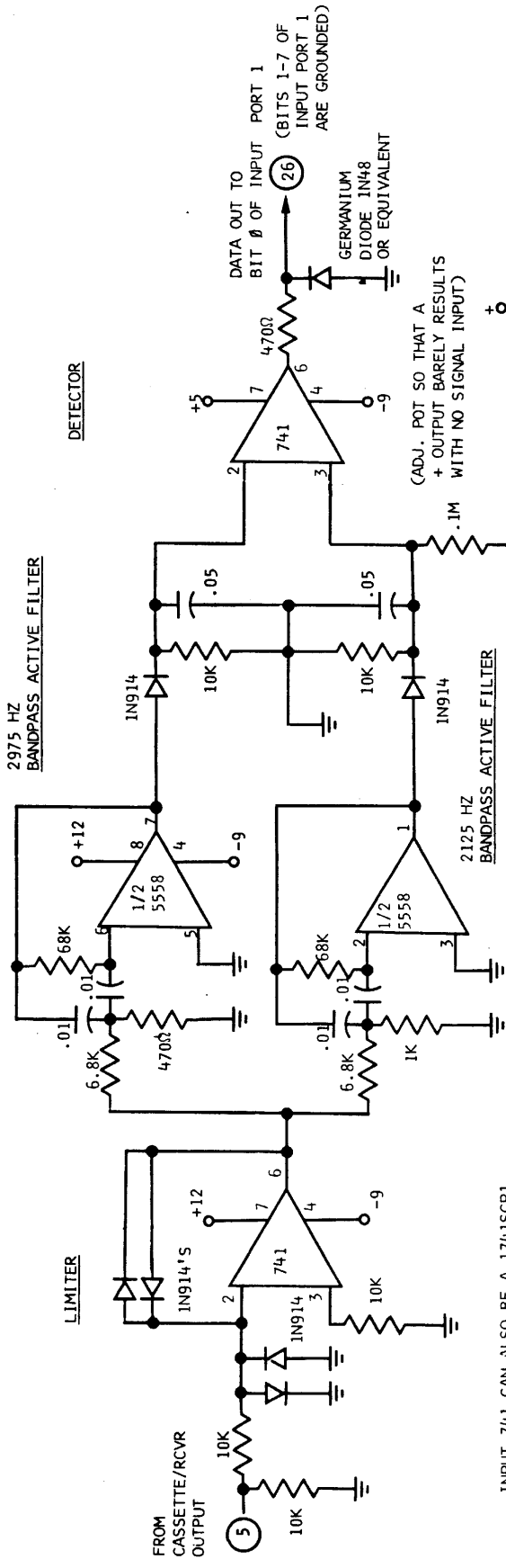
**EXTENDED ASCII KEYBOARD**

MECHANICAL KEYSWITCH KEYBOARDS MAY BE SUBSTITUTED FOR HALF-EFFECT KEYBOARD WITH THE FOLLOWING RESTRICTION: CONNECT ONE SIDE OF EACH KEY TO +5. CONNECT THE OTHER SIDE OF EACH CHARACTER KEYSWITCH TO THE ANODE END OF TWO DIODES. EACH DIODE CATHODE (BAR) END GOES TO THE APPROPRIATE 220 OHM MATRIX LINE.

\* DESIGNED FOR AN ABILITY TO ENTER ALL 128 CHARACTERS OF THE MCM6571L CHARACTER GENERATOR.

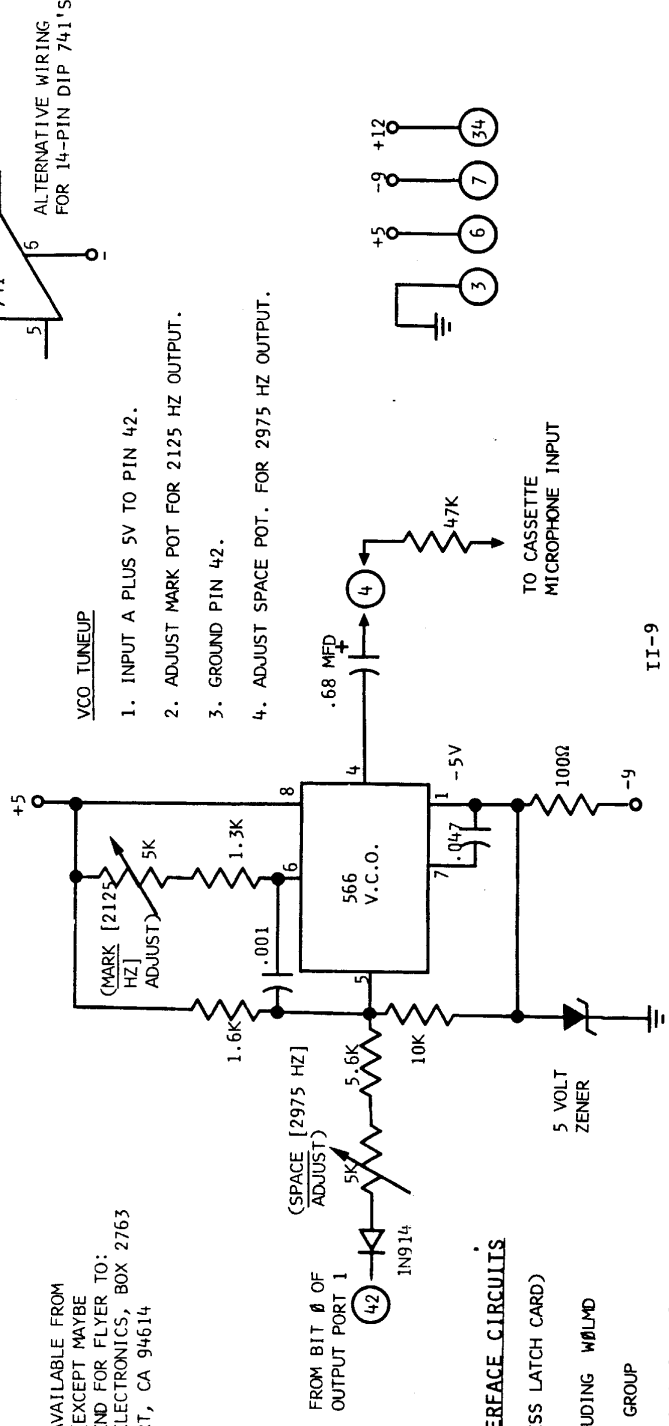
\* ANY STANDARD ASCII KEYBOARD IS USABLE WITH SUDING SOFTWARE IF IT HAS AN +500 μS DURATION MSB STROBE PULSE.





INPUT 741 CAN ALSO BE A 1741SCP1 WHICH HAS A HIGHER SLEW RATE AND IS A PIN FOR PIN EQUAL TO MINI-DIP 741. (HARDER TO FIND, THOUGH)

ALL IC'S ARE AVAILABLE FROM BILL GODBOUT (EXCEPT MAYBE 1741SCP1) - SEND FOR FLYER TO: BILL GODBOUT ELECTRONICS, BOX 2763 OAKLAND AIRPORT, CA 94614



\*8008 MICROPROCESSOR BACK-PLANE BOARD INTERCONNECTIONS\*

	CPU	IN	IN-X	ADDRESS	MEM-1	MEM-2	OUT	OUT-X	TV	LED
A1	1 X	∅PORT5	1	X	R-1		∅PORT5	1	VIDEO	X
	2			W	W	R-1	G			X
GND	3 X	X		X	X	X	X		X	X
SLO	4 X	X	PORT5	CASS IN				PORT5		
SLZ	5 X	X		CASS OUT						
+5V	6 X	X		X	X	X	X		X	X
-7V	7 X		7	X	X	X		7	X	
READY	8 X		∅	X				∅		
D0	9 X	X		Y	Y	Y	Y			Y
D1	10 X	X		Y	Y	Y	Y			Y
D2	11 X	X	PORT4	Y	Y	Y	Y	PORT4		Y
D3	12 X	X		Y	Y	Y	Y			Y
D4	13 X	X		Y	Y	Y	Y			Y
D5	14 X	X		Y	Y	Y	Y			Y
D6	15 X	X	7	Y	Y	Y	Y	7		Y
D7	16 X	X	∅	Y	Y	Y	Y	∅		Y
	17	J		J						
D7	18 X			X	X	X	Z		Z	Q-1
D6	19 X	∅	PORT3	X	X	X	Z	PORT3	Z	Q-2
D5	20 X			X	X	X	Z		Z	Q-3
D4	21 X			X	X	X	Z		Z	Q-4
D3	22 X	PORT6		X	X	X	Z		Z	Q-5
D2	23 X		7	X	X	X	Z	7	Z	Q-6
D1	24 X		∅	X	X	X	Z	∅	Z	Q-7
D0	25 X			X	X	X	Z		Z	Q-8
	26 X	7								X-1
D7	27 X	X	PORT2	DATA OUT	X	X		PORT2		X-2
D6	28 X	M		XTRNRDY	M	M	X			X-3
D5	29 X	M		X-1	M	M				M
D4	30 X	M		X-2	M	M				M
D3	31 X	M	7	X	M	M		7		M
D2	32 X	M	∅	X	M	M		∅		M
D1	33 X	M		X-3	M	M				M
D0	34 X	X		+12	M-1	M-1	X		+12	M-1
	35 X	Q-1	PORT1	X	M-2	M-2		PORT1		M-2
	36 X <sup>1</sup>	Q-2		A						A
	37 X <sup>1</sup>	M-1		A						A
	38	M-2		X	X	X				X
	39	X	7	X	X	X	X	7		X
	40	X	∅	X	X	X	X	∅		X
	41	X		X	X	X	X			X
	42	Q-3		DATA IN						X
	43	Q-4	PORT∅	A				PORT∅		A
	44	Q-5		A						A
	45	Q-6		A						A
	46	Q-7		A						A
	47	Q-8	7	A				7		A

## BACKPLANE WIRING INTERCONNECTION NOTES

THIS BACKPLANE DIAGRAM IS BASED ON THE DIGITAL GROUP'S CONFIGURATION WHICH EVOLVED FROM DR. SUDING'S. IT IS INCLUDED FOR INFORMATIONAL PURPOSES ONLY. YOU WILL HAVE TO ADAPT TO YOUR OWN CONNECTOR SYSTEM. AS YOU DO SO WE WOULD HIGHLY RECOMMEND EITHER ADAPTING OUR DIAGRAM OR CONSTRUCTING YOUR OWN. IT IS PROBABLY THE MOST SIGNIFICANT SINGLE DOCUMENT YOU CAN HAVE ABOUT YOUR SYSTEM FOR LATER REFERENCE.

AS FAR AS POSSIBLE WE TRIED TO MAINTAIN COMPATABILITY WITH THE ORIGINAL MARK-8 PINOUT. ALSO, MOST CONNECTIONS WERE MADE IN PARALLEL BETWEEN CONNECTORS. LIKE SINGLE LETTERS (HORIZONTALLY) ARE CONNECTED TOGETHER. LETTERS WITH NUMBERS (Q-1, ETC.) ARE CONNECTED TO THE SAME LETTER-NUMBER PAIR (Q-1 TO Q-1, ETC.) AND MAY NOT NECESSARILY BE IN PARALLEL. INPUT AND OUTPUT PORTS ARE BROUGHT OUT TO A PADDLE BOARD CONNECTOR ON THE REAR OF THE CHASSIS.

<sup>1</sup>BROUGHT OUT BUT NOT USED

MODIFICATIONS AND EXTENSIONS PARTS LISTS

TV Typewriter  
Character Generator

<u>Qty</u>	<u>Part</u>
3	7400
1	7401
2	7404
2	7408
1	7410
1	7420
2	7430
1	74123
2	74157
1	74165
8	74193
7	1101
1	MCM6571L
1	5990 KHz crystal
8	1N914 diodes
1	22 ohm 1/2w resis.
1	220 ohm
4	470 ohm
1	1K
1	1.3K
16	2.2K
1	2.4K
8	10K
1	22K
1	100 pfd capacitor
2	.001 mfd capac.
3	.01 mfd
3	1 mfd Tantalum caps.

Input Port  
Extensions

<u>Qty</u>	<u>Part</u>
14	7401
3	7404
8	1K 1/2w Resis.

Output Port  
Extensions

<u>Qty</u>	<u>Part</u>
1	7402
6	7475

Extended ASCII  
Keyboard Encoder

<u>Qty</u>	<u>Part</u>
4	7400
3	7402
1	7404
1	7430
1	74123
19	220 ohm resis.
1	5.6K
1	10K
1	.68 mfd tantalum
1	.22 mfd tantalum
	capacitor

Cassette  
Interface

<u>Qty</u>	<u>Part</u>
1	5558
1	556
2	741
7	1N914 diodes
1	1N48 Germanium diode or equivalent
1	5V Zener diode
1	100 ohm 1/2w resis.
2	470 ohm
1	1K
1	1.3K
1	1.6K
1	5.6K
2	6.8K
6	10K
1	47K
2	68K
1	100K
1	50K Trimpot
2	5K Trimpot
1	.001 mfd capacitor
4	.01 mfd
1	.047 mfd
2	.05 mfd
1	.68 mfd electrolytic capacitor

Address Latch

<u>Qty</u>	<u>Part</u>
3	7400
1	7402
3	7413
4	74123
6	110 ohm resis.
1	1K
6	5.6K
7	10K
1	22K
1	680 pfd capac.
6	.01 mfd
1	.1 mfd
6	10 mfd elec. cap.

Front Panel  
Keyboard

<u>Qty</u>	<u>Part</u>
5	7400
3	7447
3	Man-1 or equal
21	100 - 470 ohm resistors
9	2.2K
1	16-key keyboard
	<u>7-Segment Octal Display</u>
	<u>Qty</u> <u>Part</u>
12	7447
2	7475
12	Man-1 or equal
84	100 - 470 ohm resis.

Power Supply

<u>Qty</u>	<u>Part</u>
4	1N2069 or equal
1	5Amp Bridge
1	HEP S7001
1	LM 309K
1	LM 340/12
1	LM 320/5
1	24 VCT 1/2Amp Transforme
1	2 Amp Fuse
1	18,000 Mfd 25V Cap or Larger
3	5,000 Mfd 25V Caps
6	1 Mfd 25V solid Tant.
1	3 ohm
1	220 ohm
1	150 ohm trimpot

IC AND MAJOR PARTS SUMMARY FOR ALL MODIFICATIONS AND EXTENSIONS

for bulk orders if desired (does not include power supply)

<u>Quantity</u>	<u>Part</u>
15	7400
15	7401
5	7402
6	7404
2	7408
1	7410
3	7413
1	7420
3	7430
15	7447
8	7475
6	74123
2	74157
1	74165
8	74193
1	5558
1	556
2	741
1	MCM6571L
15	Man-1 or equal
1	5990 KHz Crystal
1	16-key keyboard
7	1101

hardware things to do 10/1/76

- 1) order parts for front panel display (see other side) and 256 byte ram + prom
- 2) modify boards for plugin
- 3) get tape to work again (probably works already)
- 4) fix keyboard strobe
- 5) build multiplex display with 0.30" displays I bought today
- 6) invert keyboard output with 110K buffer
- 7) cassette recorder fix!

Software 1) program mini monitor using

- a) multiplex display
  - b) keyboard
  - c) option for tape I/O
- 2) ~~use~~ assembler on S9513
  - 3) simulator on CYBER
  - 4) assembler on CYBER
  - 5) calculator (octal and decimal)

## SOFTWARE NARRATIVE

The software programs and routines are all sequentially contained on the cassette. They are designed to fit into a 1K system. To dump the cassette into memory, hand enter the program "Cassette Dumper for Cold Start" at location 003000 (and following) from the documentation (page IV-16). Enter "104" (jump unconditional) at location 000000, "000" at 000001, and "003" at 000002. Do a Restart "005" which is the following:

Enter "005" on the data register  
Press STOP  
Press INTERRUPT

Begin playing the cassette. After the constant leader tone stabilizes, press RUN and the Cassette Dumper program should begin dumping the cassette into storage from address 000000 to 003377.

The cassette will return to a constant tone at the end of the data. The computer should halt and display 003056 at that time. Stop the cassette. Do a Restart to the program you wish to execute. If the Cassette Dumper program halted before the data stopped playing, the load was bad and should be redone. (After the interface is completed and debugged this is very rare and is usually due to a mistake in entering the Cassette Dumper program, a failing memory chip, or dirty heads on the recorder.) After you have successfully loaded the distribution cassette into your system, and verified its contents, we would recommend writing a new cassette with the "Cassette Loader" program. This will negate any speed discrepancies between systems and recorders and will also provide a form of backup.

### Cassette Contents and 1K Storage Map as Distributed

<u>Address</u>	<u>Program/Routine</u>
000000 - 000046	Restart to Programs
000047 - 000071	Memory Clear
000100 - 000147	Memory Checker
000150 - 000176	Bit Reverse
000200 - 000220	TV Character Generator Test
000230 - 000270	Keyboard to Memory
000300 - 000336	Number Sorting
001000 - 001137	Running TV Display
001240 - 001373	TV Character Demonstration
002000 - 002077	Keyboard to TV
002100 - 002351	TV Storage Dump
003000 - 003056	Cassette Dumper
003100 - 003120	Home Erase Subroutine
003122 - 003133	Spacer Subroutine
003135 - 003146	Writer Subroutine
003150 - 003160	"Dumped ok" constant
003162 - 003176	Timer Subroutine
003200 - 003327	Cassette Loader



## Compatibility

If your system does not match our configuration (a very real possibility) you will have to modify the software package and/or upgrade the hardware. The major changes will involve different I/O devices and Port assignments. After you have made the appropriate changes, save them on a different cassette and you're in business. Note also that the TVT is supported in a modular fashion via subroutines. If you are using a TVT I or a TVT II, you should only have to change the subroutines and their call addresses within the programs if they change. The Suding TVT is, of course, fully supported.

## Clock Synchronization

The cassette was written with a 4 Mhz crystal clock which is the frequency specified for the original Mark-8. If your system is not based on a 4 Mhz crystal (or equivalent with different divisors) you may have trouble reading the cassette in. To recover, use the documentation and hand enter the programs and write your own cassette which will then be in sync with your system.

## Port Assignments

Dr. Suding's and the digital group's modified Micros have the following configuration which is supported by the 1K system:

Input Port 0	= Keyboard
Input Port 1 (Bit 0)	= Cassette In
Output Port 1 (Bit 0)	= Cassette Out
Output Port 6	= TVT
Output Port 7	= Front Panel LEDs

## Cassette Recorder Note

There are no required modifications to the cassette deck. However, your cassette recorder may have an automatic speaker cutoff when a plug is inserted into the auxiliary output. We would recommend bypassing the cutoff so you can hear the cassette in operation. Bypass usually involves shorting two pins on the auxiliary output jack.

## Keyboard Program Loader

The Keyboard Program Loader is not included in this original distribution of Packet #1. Some bugs developed which we are in the process of shooting plus we wanted to incorporate a number of new features. A copy of the new program will be distributed to all purchasers of Packet #1 without charge at a later time.

PROGRAM: RESTART TO PROGRAMS

OCTAL ADDRESS	OCTAL CODE	OPERATION
000000	104	Jump unconditional
000001	LLL	
000002	00H	(Beginning point of operational program)
000003		
000004	}	Reserved for ROM Operating System
000005		
000006		
000007		
000010	104	Jump unconditional
000011	200	
000012	003	Load Cassette
000013		
000014		
000015		
000016		
000017		
000020	104	Jump unconditional
000021	000	
000022	003	Dump Cassette
000023		
000024		
000025		
000026		
000027		
000030	104	Jump unconditional
000031	000	
000032	002	TV Keyboard
000033		
000034		
000035		
000036		
000037		
000040	104	Jump unconditional
000041	100	
000042	002	TV Storage Dump
000043		
000044		
000045		
000046		

PROGRAM: MEMORY CLEAR - Sets all of storage above this routine to zero.

<u>OCTAL ADDRESS</u>	<u>OCTAL CODE</u>	<u>OPERATION</u>
000047	016	Load B with 000
000050	000	
000051	056	Load H with 000
000052	000	
000053	066	Load L with 071
000054	071	
000055	371	Store B in memory
000056	060	Increment L
000057	110	Jump not zero
000060	055	
000061	000	
000062	050	Increment H
000063	305	Load A with H
000064	074	Compare A with 004*
000065	004	
000066	110	Jump not equal
000067	055	
000070	000	
000071	000	Halt

COMMENTS: \*Set to byte above highest available address  
1K= 004  
1.5K= 006  
2.0K= 010, etc.

Start this program by loading "047" at 000001 and "000" at 000002, then do a Restart "005". The program should halt with a "000071" in the address registers and "000" in all memory positions above 072. All routines in storage above 072 will be lost! This routine is helpful when all storage is to be set to zero prior to initially building a program. Setting all of unused storage to zero can greatly aid in finding bad jumping and calling routines, since a "000" is a halt instruction, and the halting address will be displayed. Comparing this address with your code will usually show the error.

PROGRAM: MEMORY CHECKER - Exercise those surplus 1101's!

<u>OCTAL ADDRESS</u>	<u>OCTAL CODE</u>	<u>OPERATION</u>	<u>OCTAL ADDRESS</u>	<u>OCTAL CODE</u>	<u>OPERATION</u>
000100	016	Load B with 000	000124	110	Jump if not equal
000101	000		000125	106	
000102	056	Load H with 000	000126	000	
000103	000		000127	301	Load A with B
000104	066	Load L with 150	000130	074	Compare A with 377
000105	150		000131	377	
000106	371	Store B in memory	000132	150	Jump if equal
000107	307	Load A from memory	000133	100	
000110	271	Compare A to B	000134	000	
000111	110	Jump not equal	000135	010	Increment B
000112	141		000136	104	Jump uncond
000113	000		000137	102	
000114	060	Increment L	000140	000	
000115	110	Jump not zero	000141	137	Out 7
000116	106		000142	000	Halt
000117	000		000143	305	Load A with H
000120	050	Increment H	000144	137	Out 7
000121	305	Load A with H	000145	306	Load A with L
000122	074	Compare A with	000146	137	Out 7
000123	004*	004*	000147	000	Halt

COMMENTS: All routines above 000147 will be lost. Enter "100" at 000001 and "000" at 000002. Do a restart 005. Address and Memory Data LED's continually "twinkle" if every bit OK. If the program stops, out 7 has the failing bit. Doing a restart to 000143 and single stepping will show the failing address.

\*Set address at 123 according to your storage capacity:

256 bytes - 001  
512 bytes - 002  
768 bytes - 003  
1K bytes - 004 etc.

PROGRAM: BIT REVERSE by John Nall, Tallahassee, FL

OCTAL ADDRESS	OCTAL CODE	OPERATION
000150	006	Load # in A
000151	#	(01001001 now) (111 Octal)
000152	066	Load L with 000
000153	000	
000154	016	Load B with 8
000155	010	
000156	350	Load H with A
000157	305	Load A with H
000160	012	Shift A right
000161	350	Load H with A
000162	044	AND A with 200
000163	200	
000164	206	ADD L to A
000165	002	Shift left
000166	360	Load L with A
000167	011	Decrement B
000170	301	Load A with B
000171	110	Jump if not zero
000172	157	
000173	000	
000174	306	Load A with L
000175	137	Out 7 (10010010 will be displayed)
000176	000	Halt (222 octal)

COMMENTS: Load a byte into the A register.  
This routine does a bit for bit swap, LSB through MSB.  
Start by entering a sample # at 000151. Enter "150"  
at 000001 and a "000" at 000002, then do a Restart 005.

PROGRAM: TV CHARACTER GENERATOR TEST

<u>OCTAL</u> <u>ADDRESS</u>	<u>OCTAL</u> <u>CODE</u>	<u>OPERATION</u>
000200	006	Load A with 377
000201	377	
000202	135	Out 6
000203	220	Clear A
000204	135	Out 6
000205	310	Load B with A
000206	301	Load A with B
000207	064	OR A with 200
000210	200	
000211	135	Out 6
000212	220	Clear A
000213	135	Out 6
000214	010	Increment B
000215	110	Jump not zero
000216	206	
000217	000	
000220	000	Halt

COMMENTS: The 127 valid characters of the MCM6571L character generator will be displayed on the TV set sequentially. In addition, a random pattern of 128 characters will follow with a ■ ending. Start by loading "200" at 000001 and "000" at 000002. Restart 005 will then display the characters. This is a very handy routine for initially testing the TV readout generator system - all functions are exercised by this routine.

PROGRAM: KEYBOARD TO MEMORY

OCTAL ADDRESS	OCTAL CODE	OPERATION	FUNCTION
000230	066	Load L with 000	Starting address
000231	000		
000232	056	Load H with 001	Get character
000233	001		
000234	101	Input Ø	
000235	074	Compare A with 200	
000236	200		
000237	140	Jump if less	
000240	234		
000241	000		
000242	300	NOP	
000243	300	NOP	
000244	370	Store A	Delay to clear strobe bit
000245	016	Load B with 000	
000246	000		
000247	010	Increment B	
000250	110	Jump not zero	
000251	247		
000252	000		
000253	306	Load A with L	
000254	137	Out 7	
000255	060	Increment L	
000256	110	Jump not zero	512 character load
000257	234		
000260	000		
000261	050	Increment H	
000262	305	Load A with H	
000263	074	Compare A with 003	
000264	003		
000265	110	Jump not equal	
000266	234		
000267	000		
000270	000	Halt	

COMMENTS: Start by loading "230" at 000001 and "000" at 000002. Then do a Restart 005. The 512 bytes of memory from 001000 to 002377 are used for character storage, so the routines formerly occupying those locations will have to be restored by redumping the cassette.

PROGRAM: NUMBER SORTING - ASCENDING ORDER

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
000300	056	Load H with 001	000320	110	Jump not zero
000301	001	(Starting Page)	000321	306	
000302	066	Load L with 000	000322	000	
000303	000	(Starting Address)	000323	000	Halt
000304	026	Load C with	000324	317	Load B with Mem
000305	XXX	(Ending addr to be sorted)	000325	061	Decrement L
000306	307	Load A with Memory	000326	307	Load A with Mem
000307	060	Increment L	000327	371	Load Mem with B
000310	277	Compare Mem with A	000330	060	Increment L
000311	150	Jump if equal	000331	370	Load Mem with A
000312	317		000332	306	Load A with L
000313	000		000333/21	<del>127</del>	Out 7
000314	100	Jump if not less	000334	104	Jump unconditional
000315	324		000335	300	
000316	000		000336	000	
000317	021	Decrement C			

COMMENTS: This is a slight modification of Dr. George Haller's sorting routine. Bytes 001000 to 001377 will be sorted into ascending order. Restore the original programs by redumping the cassette. Start this program by loading byte 000001 with "300" and 000002 with "000" Then do a Restart 005.



PROGRAM: RUNNING (RIGHT TO LEFT) TV DISPLAY  
 Simulates an alphanumeric self-scan panel.

PAGE 1 OF 2

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
001000	106	Call (Name Erase)	001053	370	Store A
001001	100		001054	030	Increment D
001002	003		001055	300	NOP
001003	046	Load F with 150	001056	300	NOP
001004	150		001057	300	NOP
001005	106	Call Spacer	001060	110	Jump not zero
001006	122		001061	050	
001007	003		001062	001	
001010	056	Load H with 001	001063	066	Load L with 377
001011	001		001064	377	
001012	066	Load L with 116	001065	374	Store E
001013	116		001066	300	NOP
001014	046	Load E with 022	001067	300	NOP
001015	022		001070	006	Load A with 377
001016	106	Call Writer	001071	377	
001017	135		001072	135	Out 6
001020	003		001073	220	Clear A
001021	106	Call Timer	001074	135	Out 6
001022	162		001075	066	Load L with 340
001023	003		001076	340	
001024	056	Load H with 000	001077	307	Load A with Mem
001025	000		001100	135	Out 6
001026	006	Load A with 240	001101	220	Clear A
001027	240		001102	135	Out 6
001030	066	Load L with 340	001103	060	Increment L
001031	340		001104	110	Jump not zero
001032	370	Store A	001105	077	
001033	060	Increment L	001106	001	
001034	110	Jump not zero	001107	104	Jump uncond.
001035	032		001110	037	
001036	001		001111	001	
001037	036	Load D with 341	001112	000	Halt
001040	341		001113	000	
001041	101	Input $\emptyset$	001114	000	
001042	074	Compare A with 200	001115	000	
001043	200		001116	322	R
001044	140	Jump if less	001117	365	u
001045	040		001120	356	n
001046	001		001121	356	n
001047	340	Load E with A	001122	351	i
001050	363	Load L with D	001123	356	n
001051	307	Load A with Mem	001124	347	g
001052	061	Decrement L	001125	240	

<u>OCTAL ADDRESS</u>	<u>OCTAL CODE</u>	<u>OPERATION</u>
001126	324	T
001127	326	V
001130	240	
001131	304	D
001132	351	i
001133	363	s
001134	360	p
001135	354	l
001136	341	a
001137	371	y

COMMENTS: Entered characters from the keyboard appear at the right hand side of the screen. Subsequent characters push the previous character one position to the ~~right~~.<sup>left</sup> After 32 characters, the leftmost character is pushed off the screen.

Start by loading "000" at 000001 and "001" at 000002.  
Then do a Restart 005.

PROGRAM: TV CHARACTER DEMONSTRATION

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
001240	106	Call Home Erase	001316	003	
001241	100		001317	046	Load E with 012
001242	003		001320	012	
001243	056	Load H with 001	001321	106	Call Writer
001244	001		001322	135	
001245	066	Load L with 330	001323	003	
001246	330		001324	000	Halt
001247	046	Load E with 033	001325		
001250	033		001326		
001251	106	Call Writer	001327		
001252	135		001330	324	T
001253	003		001331	326	V
001254	046	Load E with 045	001332	240	
001255	045		001333	303	C
001256	106	Call Spacer	001334	310	H
001257	122		001335	301	A
001260	003		001336	322	R
001261	046	Load E with 170	001337	301	A
001262	170		001340	303	C
001263	304	Load A with E	001341	324	T
001264	135	Out 6	001342	305	E
001265	220	Clear A	001343	322	R
001266	135	Out 6	001344	240	
001267	026	Load C with 360	001345	304	D
001270	360		001346	305	E
001271	016	Load B with 000	001347	315	M
001272	000		001350	317	O
001273	010	Increment B	001351	316	N
001274	110	Jump not zero	001352	323	S
001275	273		001353	324	T
001276	001		001354	322	R
001277	020	Increment C	001355	301	A
001300	110	Jump not zero	001356	324	T
001301	271		001357	311	I
001302	001		001360	317	O
001303	040	Increment E	001361	316	N
001304	304	Load A with E	001362	240	
001305	074	Compare A with 377	001363	316	N
001306	377		001364	305	E
001307	110	Jump not equal	001365	301	A
001310	263		001366	324	T
001311	001		001367	240	
001312	046	Load E with 050	001370	310	H
001313	050		001371	325	U
001314	106	Call Spacer	001372	310	H
001315	122		001373	277	?

COMMENTS: Start by loading "240" at 000001 and "001" at 000002. Then do a Restart 005. This routine contains timing loops to slow it down. For a full speed version, NOP (enter 300) for bytes 001267 - 001302.

PROGRAM: KEYBOARD TO TV

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
002000	106	Call unconditional -	002040	101	Input Ø
002001	100	Home Erase	002041	074	Compare A with 200
002002	003		002042	200	
002003	046	Load E with 150	002043	140	Jump if less
002004	150		002044	040	
002005	106	Call unconditional -	002045	002	
002006	122	Spacer	002046	135	Out 6
002007	003		002047	220	Clear A
002010	056	Load H with 002	002050	135	Out 6
002011	002		002051	016	Load B with 000
002012	066	Load L with 062	002052	000	
002013	062		002053	010	Increment B
002014	046	Load E with 16	002054	110	Jump not zero
002015	016		002055	053	
002016	106	Call unconditional -	002056	002	
002017	135	Writer	002057	104	Jump unconditional
002020	003		002060	040	
002021	016	Load B with 000	002061	002	
002022	000		002062	324	T
002023	026	Load C with 000	002063	326	V
002024	000		002064	240	
002025	020	Increment C	002065	313	K
002026	110	Jump not zero	002066	345	E
002027	025		002067	371	Y
002030	002		002070	342	B
002031	010	Increment B	002071	357	O
002032	110	Jump not zero	002072	341	A
002033	023		002073	362	R
002034	002		002074	344	D
002035	106	Call unconditional -	002075	240	
002036	100	Home Erase	002076	317	O
002037	003		002077	316	N

COMMENTS: Begin this program by doing a Restart 035.

PROGRAM: TV STORAGE DUMP - Dumps storage in octal onto TV screen  
 PAGE 1 OF 2

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
002100	106	Call unconditional -	002160	106	Call unconditional -
002101	100	Home Erase	002161	240	Character
002102	003		002162	002	
002103	046	Load E with 050	002163	026	Load C with 004
002104	050		002164	004	
002105	106	Call unconditional -	002165	046	Load E with 001
002106	122	Spacer	002166	001	
002107	003		002167	106	Call unconditional -
002110	056	Load H with 002	002170	122	Spacer
002111	002		002171	003	
002112	066	Load L with 300	002172	347	Load E from memory
002113	300		002173	106	Call unconditional -
002114	046	Load E with 020	002174	240	Character
002115	020		002175	002	
002116	106	Call unconditional -	002176	060	Increment L
002117	135	Writer	002177	021	Decrement C
002120	003		002200	110	Jump not zero
002121	046	Load E with 114	002201	165	
002122	114		002202	002	
002123	106	Call unconditional -	002203	046	Load E with 012
002124	122	Spacer	002204	012	
002125	003		002205	106	Call unconditional -
002126	046	Load E with 037	002206	122	Spacer
002127	037		002207	003	
002130	106	Call unconditional -	002210	306	Load A with L
002131	135	Writer	002211	044	AND A with 037
002132	003		002212	037	
002133	056	Load H with $\emptyset$	002213	300	NOP
002134	000		002214	074	Compare A with 000
002135	066	Load L with $\emptyset$	002215	000	
002136	000		002216	110	Jump not equal
002137	101	Input $\emptyset$	002217	153	
002140	074	Compare A with 240	002220	002	
002141	240		002221	306	Load A with L
002142	110	Jump not equal	002222	074	Compare A with 000
002143	137		002223	000	
002144	002		002224	110	Jump not equal
002145	106	Call unconditional -	002225	137	
002146	100	Home Erase	002226	002	
002147	003		002227	050	Increment H
002150	300	NOP	002230	305	Load A with H
002151	300	NOP	002231	074	Compare A with 004*
002152	300	NOP	002232	004*	
002153	345	Load E with H	002233	110	Jump not equal
002154	106	Call unconditional -	002234	137	
002155	240	Character	002235	002	
002156	002		002236	000	Halt
002157	346	Load E with L	002237	000	

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
002240	304	Load A with E	002305	317	O
002241	044	AND A with 300	002306	322	R
002242	300		002307	301	A
002243	002	Shift left	002310	307	G
002244	002	Shift left	002311	305	E
002245	064	OR with 260	002312	240	
002246	260		002313	304	D
002247	135	Out 6	002314	325	U
002250	220	Clear A	002315	315	M
002251	135	Out 6	002316	320	P
002252	304	Load A with E	002317	240	
002253	044	AND A with 070	002320	320	P
002254	070		002321	362	r
002255	012	Shift right	002322	345	e
002256	012	Shift right	002323	363	s
002257	012	Shift right	002324	363	s
002260	064	OR with 260	002325	240	
002261	260		002326	323	S
002262	135	Out 6	002327	320	P
002263	220	Clear A	002330	301	A
002264	135	Out 6	002331	303	C
002265	304	Load A with E	002332	305	E
002266	044	AND A with 007	002333	240	
002267	007		002334	353	k
002270	064	OR with 260	002335	345	e
002271	260		002336	371	y
002272	135	Out 6	002337	240	
002273	220	Clear A	002340	346	f
002274	135	Out 6	002341	357	o
002275	007	Return uncond.	002342	362	r
002276	000		002343	240	
002277	000		002344	360	p
002300	324	T	002345	341	a
002301	326	V	002346	347	g
002302	240		002347	351	i
002303	323	S	002350	356	n
002304	324	T	002351	347	g

COMMENTS: \*Set A comparison to byte address higher than highest byte of your storage capacity:

1K = 004 (shown)

1.5K = 006

2K = 010

Begin this program by doing a Restart 045.

PROGRAM: CASSETTE DUMPER FOR COLD START - SHORT FORM

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
003000	056	Load H with 000	003030	016	Load B with 040
003001	000		003031	040	
003002	066	Load L with 000	003032	011	Decrement B
003003	000		003033	110	Jump not zero
003004	026	Load C with 010	003034	032	
003005	010		003035	003	
003006	036	Load D with 000	003036	021	Decrement C
003007	000		003037	110	Jump not zero
003010	103	Input 1	003040	024	
003011	044	AND A with 001	003041	003	
003012	001		003042	373	Store D in mem
003013	110	Jump if not zero	003043	060	Increment L
003014	010	<i>Loop until not 0</i>	003044	110	Jump not zero
003015	003		003045	004	
003016	016	Load B with 060	003046	003	
003017	060		003047	050	Increment H
003020	011	Decrement B	003050	305	Load A with H
003021	110	Jump not zero	003051	074	Compare A with 004
003022	020		003052	004	
003023	003		003053	110	Jump not equal
003024	103	Input 1	003054	004	
003025	203	Add D to A	003055	003	
003026	012	Shift right	003056	000	Halt
003027	330	Load D with A			

COMMENTS: This routine is hand keyed into the upper portion of the 1K 8008 microprocessor when power is first applied. The program deserializes the output of the cassette, and loads the 8-bit bytes into memory starting at byte 0. The speed is approximately 40 bytes/second (1K in 25 sec.)

Begin this program by doing a Restart 025 while the cassette is playing the constant tone leader prior to the data portion. The timing constants at 003017 and 003031 assume the 20µs cycle time of the Mark-8 (use of a 4Mhz crystal).

*Let in 3033 NG*

PROGRAM: TV - HOME ERASE, SPACER, TIMER, and WRITER SUBROUTINES

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
"Home Erase"					
003100	006	Load A with 377	003140	135	Out 6
003101	377		003141	060	Increment L
003102	135	Out 6	003142	041	Decrement E
003103	220	Clear A	003143	110	Jump not zero
003104	310	Load B with A	003144	135	
003105	135	Out 6	003145	003	
003106	137	Out 7	003146	007	Return uncond.
003107	006	Load A with 240	003147		
003110	240		003150	304	D
003111	135	Out 6	003151	365	u
003112	220	Clear A	003152	355	m
003113	135	Out 6	003153	360	p
003114	010	Increment B	003154	345	e
003115	110	Jump not zero	003155	344	d
003116	107		003156	240	
003117	003		003157	317	O
003120	007	Return uncond.	003160	313	K
"Spacer"			"Timer"		
003122	006	Load A with 240	003162	016	Load B with 000
003123	240		003163	000	
003124	135	Out 6	003164	026	Load C with 000
003125	220	Clear A	003165	000	
003126	135	Out 6	003166	020	Increment C
003127	041	Decrement E	003167	110	Jump not zero
003130	110	Jump not zero	003170	166	
003131	122		003171	003	
003132	003		003172	010	Increment B
003133	007	Return uncond.	003173	110	Jump not zero
"Writer"			003174	164	
003135	307	Load A from Mem.	003175	003	
003136	135	Out 6	003176	007	Return uncond.
003137	220	Clear A			

COMMENTS: 003100 - 003106 = Homes counter  
 003107 - 003115 = Enters 256 blanks  
 003122 - 003133 = E Register should contain the number of blanks when calling this subroutine  
 003135 - 003145 = Enters desired character. H & L should first storage address; E should contain the number of sequential characters starting at this address.  
 003162 - 003176 = Timer is a 5 second delay. By entering at 003164 with the B Register preset by the calling routine, a controlled delay of about 19.5 milliseconds per count decrementing from B = 377 can be obtained (B = 377 gives a 19.5 ms delay, B = 376 gives a 39 ms delay, ...).

ABI 3PM  
 AMI 4PM



PROGRAM: CASSETTE LOADER (loads storage contents onto cassette)-  
SHORT FORM

OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE	OPERATION
003200	006	Load A with 001	003254	<del>123</del>	Out 4
003201	001		003255	016	Load B with 100
003202	<del>123</del>	Out 4	003256	100	
003203	<del>125</del>	<del>Out 2</del> NOP	003257	011	Decrement B
003204	026	Load C with 377	003260	110	Jump not zero
003205	377		003261	257	
003206	016	Load B with 377	003262	003	
003207	377		003263	305	Load A with H
003210	011	Decrement B	003264	273	Compare A with D
003211	110	Jump not zero	003265	150	Jump if equal
003212	210		003266	300	
003213	003		003267	003	
003214	021	Decrement C	003270	060	Increment L
003215	110	Jump not zero	003271	110	Jump not zero
003216	206		003272	230	
003217	003		003273	003	
003220	056	Load H with 000	003274	050	Increment H
003221	000		003275	104	Jump unconditional
003222	066	Load L with 000	003276	230	
003223	000		003277	003	
003224	036	Load D with 003*	003300	306	Load A with L
003225	003*		003301	274	Compare A with E
003226	046	Load E with 377	003302	150	Jump if equal
003227	377		003303	311	
x 003230	026	Load C with 011	003304	003	
003231	011		003305	060	Increment L
003232	302	Load A with C	003306	104	Jump unconditional
003233	022	Rotate left thru car.	003307	230	
003234	307	Load A from memory	003310	003	
003235	022	Rotate left thru car.	003311	026	Load C with 377
003236	<del>123</del>	Out 4	003312	377	
003237	016	Load B with 040	003313	016	Load B with 177
003240	040		003314	177	
003241	011	Decrement B	003315	011	Decrement B
003242	110	Jump not equal	003316	110	Jump not zero
003243	241		003317	315	
003244	003		003320	003	
003245	032	Rotate rht thru car.	003321	021	Decrement C
003246	021	Decrement C	003322	110	Jump not zero
003247	110	Jump not zero	003323	313	
003250	236		003324	003	
003251	003		003325	220	Clear A
003252	006	Load A with 001	003326	125	Out 2
003253	001		003327	000	Halt

COMMENTS: \*Set byte 003225 to the address of the highest byte page in your system:

1K = 003 (shown)

1.5K = 005

2K = 007

Begin this program by doing a Restart 015 after having placed the cassette in record and running clear of the leader

PROGRAM: 8223 ROM PROGRAMMING FOR CASSETTE DUMPER

OCTAL ADDRESS	OCTAL CODE	BINARY CODE				OCTAL ADDRESS	OCTAL CODE	BINARY CODE			
		Bit7	(ROM1)	Bit0			Bit7	(ROM2)	Bit0		
017300	056	0 0	1 0 1	1 1 0	017340	324	1 1	0 1 0	1 0 0		
017301	000	0 0	0 0 0	0 0 0	017341	017	0 0	0 0 1	1 1 1		
017302	066	0 0	1 1 0	1 1 0	017342	373	1 1	1 1 1	0 1 1		
017303	000	0 0	0 0 0	0 0 0	017343	060	0 0	1 1 0	0 0 0		
017304	026	0 0	0 1 0	1 1 0	017344	110	0 1	0 0 1	0 0 0		
017305	010	0 0	0 0 1	0 0 0	017345	304	1 1	0 0 0	1 0 0		
017306	036	0 0	0 1 1	1 1 0	017346	017	0 0	0 0 1	1 1 1		
017307	000	0 0	0 0 0	0 0 0	017347	050	0 0	1 0 1	0 0 0		
017310	103	0 1	0 0 0	0 1 1	017350	305	1 1	0 0 0	1 0 1		
017311	044	0 0	1 0 0	1 0 0	017351	074	0 0	1 1 1	1 0 0		
017312	001	0 0	0 0 0	0 0 1	017352	010*	0 0	0 0 1	0 0 0		
017313	110	0 1	0 0 1	0 0 0	017353	110	0 1	0 0 1	0 0 0		
017314	310	1 1	0 0 1	0 0 0	017354	304	1 1	0 0 0	1 0 0		
017315	017	0 0	0 0 1	1 1 1	017355	017	0 0	0 0 1	1 1 1		
017316	016	0 0	0 0 1	1 1 0	017356	104	0 1	0 0 0	1 0 0		
017317	060	0 0	1 1 0	0 0 0	017357	003	0 0	0 0 0	0 1 1		
017320	011	0 0	0 0 1	0 0 1	017360	000	0 0	0 0 0	0 0 0		
017321	110	0 1	0 0 1	0 0 0							
017322	320	1 1	0 1 0	0 0 0							
017323	017	0 0	0 0 1	1 1 1							
017324	103	0 1	0 0 0	0 1 1							
017325	203	1 0	0 0 0	0 1 1							
017326	012	0 0	0 0 1	0 1 0							
017327	330	1 1	0 1 1	0 0 0							
017330	016	0 0	0 0 1	1 1 0							
017331	040	0 0	1 0 0	0 0 0							
017332	011	0 0	0 0 1	0 0 1							
017333	110	0 1	0 0 1	0 0 0							
017334	332	1 1	0 1 1	0 1 0							
017335	017	0 0	0 0 1	1 1 1							
017336	021	0 0	0 1 0	0 0 1							
017337	110	0 1	0 1 0	0 0 0							

\* Schedule of storage capacity  
byte 017352:

1K	004	0 0	0 0 0	1 0 0
1.5K	006	0 0	0 0 0	1 1 0
2K	010	(shown & recommended)		

COMMENTS: This ROM set is located on memory board 1 with its pair of 7430's and 2/4 7400 already on this board. The ROM's address is at the top most end of the total 4K potential. The ROM directs control back to position 000003 following a successful load. Set byte 017352 to byte above top RAM position available. Placing a 000 at address 000003 in RAM will produce a halt after the dump. Placing a Jump, Call, or Restart instruction at address 000003 will enable a load and go operation.

PROGRAM \_\_\_\_\_

MICROPROCESSOR CODING SHEET

PAGE \_\_\_ OF \_\_\_

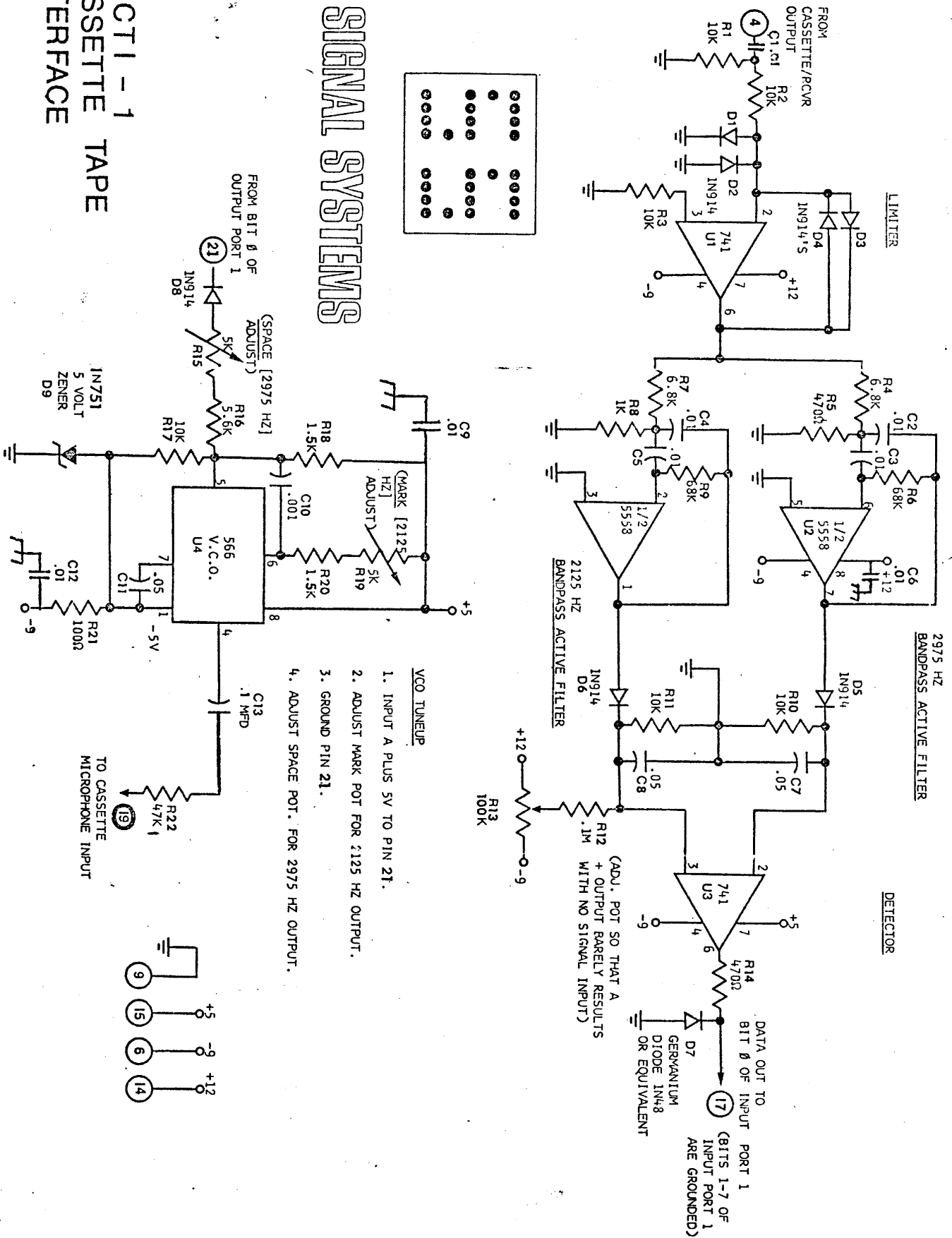
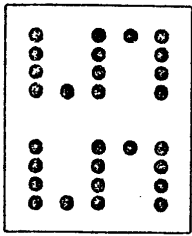
OCTAL ADDRESS	OCTAL CODE	OPERATION	OCTAL ADDRESS	OCTAL CODE
00 000			00 040	
00 001			00 041	
00 002			00 042	
00 003			00 043	
00 004			00 044	
00 005			00 045	
00 006			00 046	
00 007			00 047	
00 010			00 050	
00 011			00 051	
00 012			00 052	
00 013			00 053	
00 014			00 054	
00 015			00 055	
00 016			00 056	
00 017			00 057	
*00 020			00 060	
00 021			00 061	
00 022			00 062	
00 023			00 063	
00 024			00 064	
00 025			00 065	
00 026			00 066	
00 027			00 067	
00 030			00 070	
00 031			00 071	
00 032			00 072	
00 033			00 073	
00 034			00 074	
00 035			00 075	
00 036			00 076	
00 037			00 077	

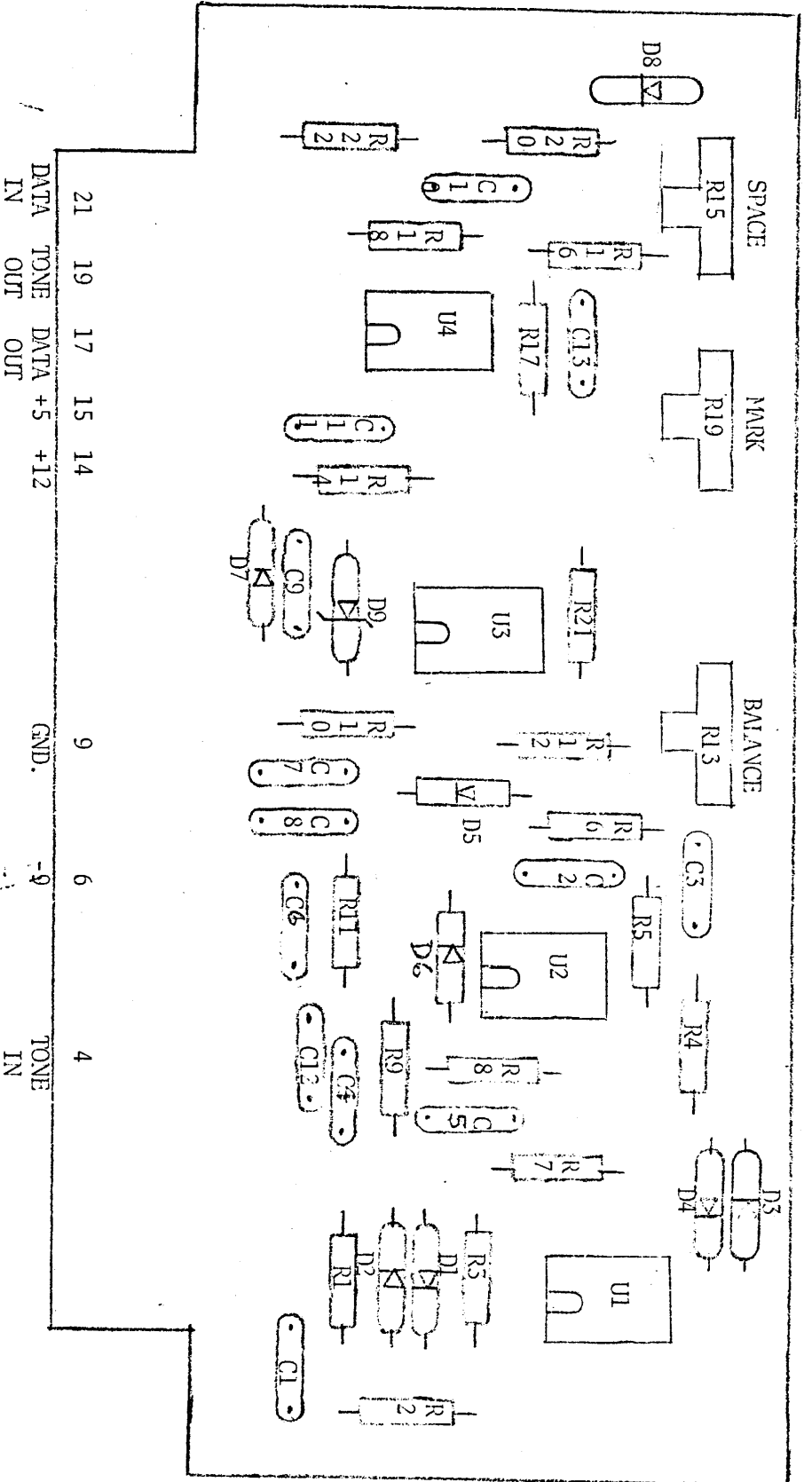
COMMENTS:



# CT1-1 CASSETTE TAPE INTERFACE

## SIGNAL SYSTEMS





Connector - Amphenol 145-022-01 or equivalent

- R1, 2, 3, 10, 11, 17 - 10K 1/4w (Brn, Blk, Orn)
- R4, 7 - 6-8K (Blu, Gry, Red)
- R5, 14 - 470 (Y10, Vio, Brn)
- R6, 9 - 68K (Blu, Gry, Orn)
- R8 - 1K (Brn, Blk, Red)
- R12 - 100K (Brn, Blk, Y10)
- R13 - 100K pot (Mouser 32RV)
- R15, 19 - 5K Pot (Mouser 32RV)
- R16 - 5.6K (Grn, Blu, Red)
- R18, 20 - 1.5K (Brn, Grn, Red)
- R21 - 100 (Brn, Blk, Brn)
- R22 - 47K (Y10, Vio, Orn)
- R23 - .1 Disc (104)
- R24 - .01 Mylar (Green) (Z-J)
- R25 - .01 Disc (503)
- R26 - .001 Disc
- R27 - .05 Mylar (Green)
- R28 - Silicon IN914, IN4148, etc.
- R29 - Germanium IN48
- R30 - Zener 1N751
- R31 - 3 - Op. Amp. LM741CN
- R32 - Dual Op. Amp. N5558V, RF4558
- R33 - Volt. Cont. Osc. LM566CN, N566V
- C1 - .01
- C2 - .01
- C3 - .01
- C4 - .01
- C5 - .01
- C6 - .01
- C7 - .01
- C8 - .01
- C9 - .01
- C10 - .01
- C11 - .01
- C12 - .01
- C13 - .01
- D1 - Diode
- D2 - Diode
- D3 - Diode
- D4 - Diode
- D5 - Diode
- D6 - Diode
- D7 - Diode
- D8 - Diode
- D9 - Diode
- U1 - Op. Amp.
- U2 - Op. Amp.
- U3 - Op. Amp.
- U4 - Op. Amp.

CT1-1