008A MICROCOMPUTER
MANUAL

RGS ELECTRONICS
3650 Charles St., Suite K
Santa Clara, Calif.
95050
(408) 247-0158

TABLE OF CONTENTS

IF YOU SHOULD FIND any errors, corrections, or improvements that should be made in the Manual, please let us know so we can make the changes and send corrections to all holders of the Manual. ---RGS Electronics.

I. CONSTRUCTION

I-1. General description and considerations.

I-1-A. The 008A is a general purpose microcomputer with 1024 words of random access memory (minimum) and a bus type I/O arrangement. The memory is expandable up to 16K words, 2K per card, of RAM or ROM. The I/O serial and parallel interface cards can be plugged into the backplane or installed in external cabinets with the I/O device. The front panel allows the user to view and load memory, step programs to debug software, and stop and start the CPU.. Internal interrupt and ready expansion are also available if needed.

The basic 008A consists of 6 p.c. boards: 2 CPU, 1 memory, 1 backplane, 1 front panel and 1 power supply. The 2 CPU boards and the memory board are 4" x 6"; they plug into the backplane which is 4.8" x 6.75". The 4.8" x 6.75" front panel board mounts on the backplane and contains the switches, LEDs, and associated circuitry. The power supply board is 3.5" x 3.5" and mounts back with the transformer and filter capacitor. The capacity of the power supply is +5V at 5A and -12V at 1A. The -9V for the 8008 is derived from the -12V. The entire assembly will fit in a cabinet 5" high, 9" wide and 12" deep. This includes up to 8 plug-in boards. When expanding the 008A to full memory capacity, another backplane board is needed, making the cabinet at least 15" wide.'

I-1-B. When constructing the 008A, there are several considerations for successful and trouble-free operation. When soldering components to the p.c. boards, use a low wattage, small-tipped soldering pencil, and good soldering practices generally. Due to the complexity of some of the boards, there is not much clearance between runs, and care must be taken to avoid solder bridges. Insert components correctly: pin 1's on the ICs are identified on the board with an extra tab on the

II-1-B. (cont'd)

pad. When using Molex pins, install them very carefully. They must be soldered in straight and vertical. Be sure the shorting bars face each other, then remove them before installing the IC. Always check Molex pins for shorts and opens; if an IC seems bad, check its seating in the Molex pins first.

WARNING: NEVER INSTALL OR REMOVE ICS OR BOARDS WITH POWER ON!!

IC's are identified by a number and a letter (e.g., 1A) representing the intersection of numbers and letters found along the edges of the p.c. boards. The 8008 and the 2102's are packed in foil; leave them there until you are ready to install them, and be sure to ground yourself before you handle them. Static charges can destroy these chips very easily.

I-2. CPU

I-2-A. The CPU section consists of 2 boards, 008A-1 and 008A-2. Assemble these 2 boards, in the following order, using the layout diagram as a reference.

1. Passive components (resistors, capacitors)
2. Molex pins
3. IC's
4. Clean and check

I-2-B. 008A-1 Parts list:

QUAN.	DESCR.
1	220 ohm trimpot
1	2.7K ¼ watt resistor
1	tantalum electrolytic capacitor
1	.002mfd disc ceramic cap.
1	.005mfd " ":
7	.01mfd " " "
1	8008 CPU IC

I-2-B. 008A-1 parts list (cont'd):

QUAN:	DESCR.
2	8833
4	7400
2	7404
1	74L04
4	7408
1	7442
1	7474
1	74107
1	74121
1	008A-1 printed circuit board
1	72-pin edge connector
2	9-pin Molex strips (for 8008)
6	8-pin Molex strips
28	7-pin Molex strips
1	edge guide
1	¼" spacer
2	#4-40 x 1¼" screw
1	#4 lockwasher
1	#4-40 nut

I-2-C. 008A-2 parts list:

QUAN.	DESCR.
1	2.7K ¼ watt resistor
1	tantalum electrolytic cap.
7	.01mfd disc ceramic cap.
1	7400
1	7437
1	7442

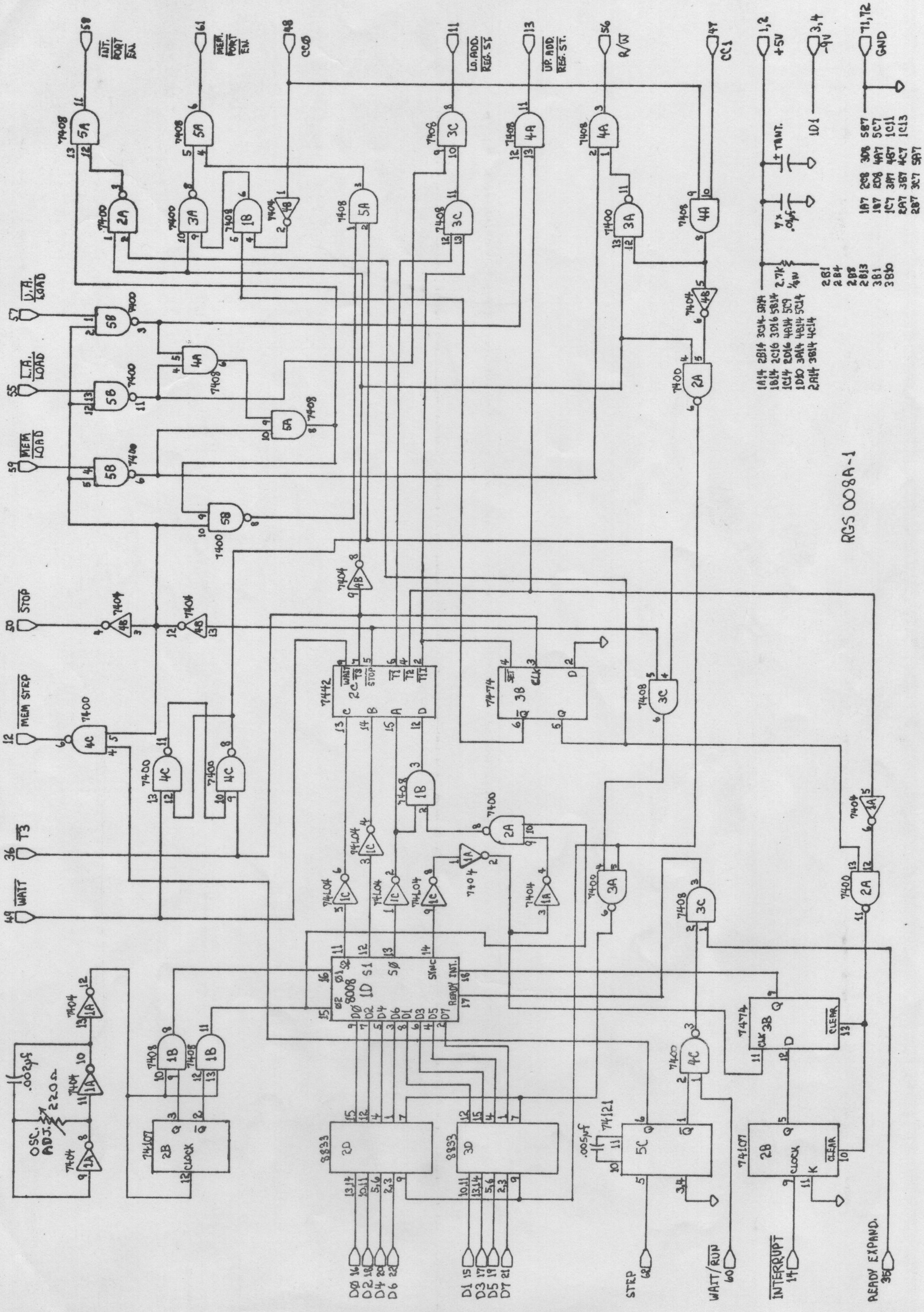
I-2-C. 008A-2 parts list (cont'd):

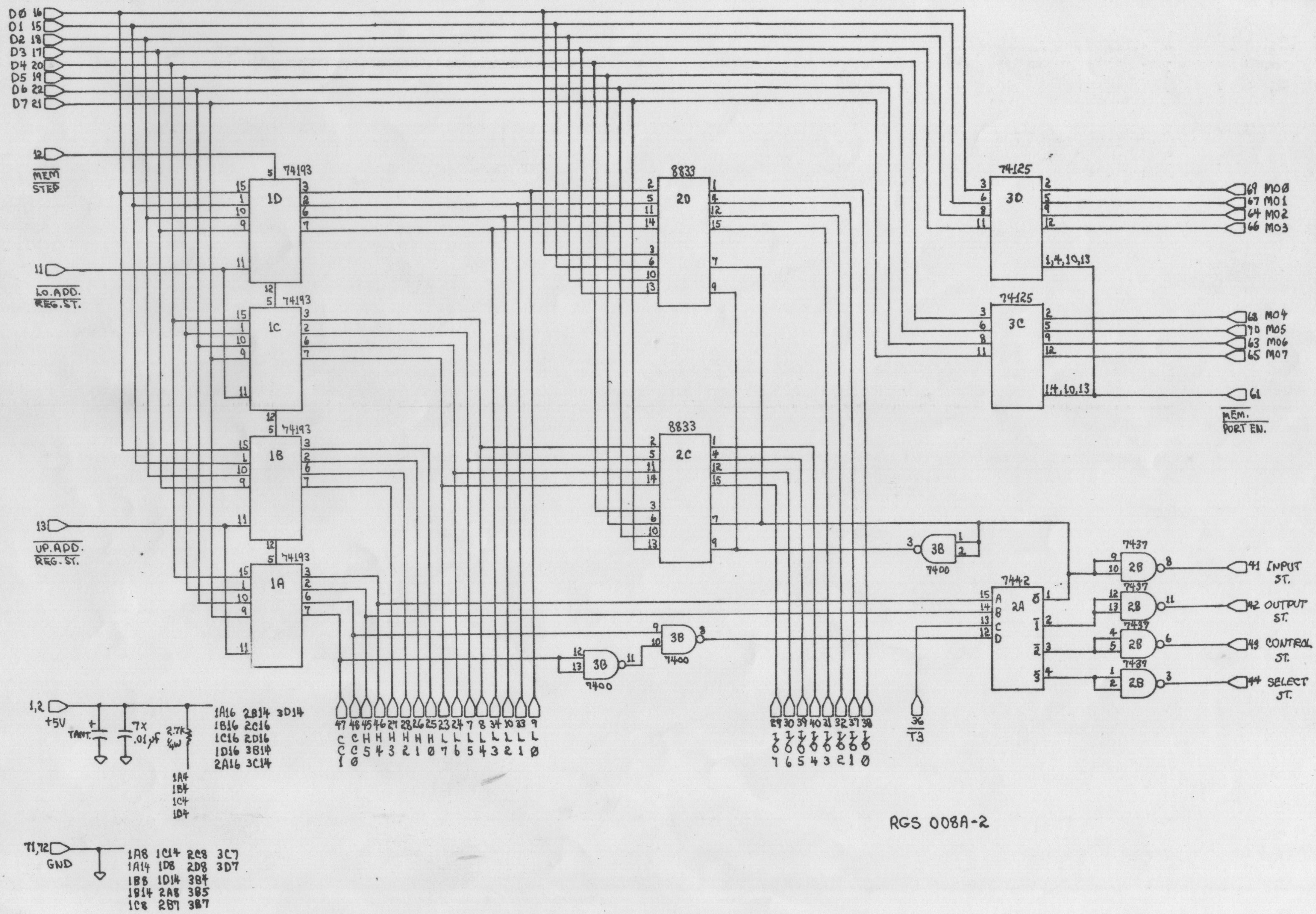
QUAN.	DESCR.
2	74125
4	74193
2	8833
1	008A-2 printed circuit board
1	72-pin edge connector
14	8-pin Molex strips
8	7-pin Molex strips
1	edge guide
1	$\frac{1}{4}$" spacer
1	#4-40 x 1$\frac{1}{4}$" screw
1	#4 lockwasher
1	#4-40 nut

I-2-D. 008A-1 schematic (see p. 5).

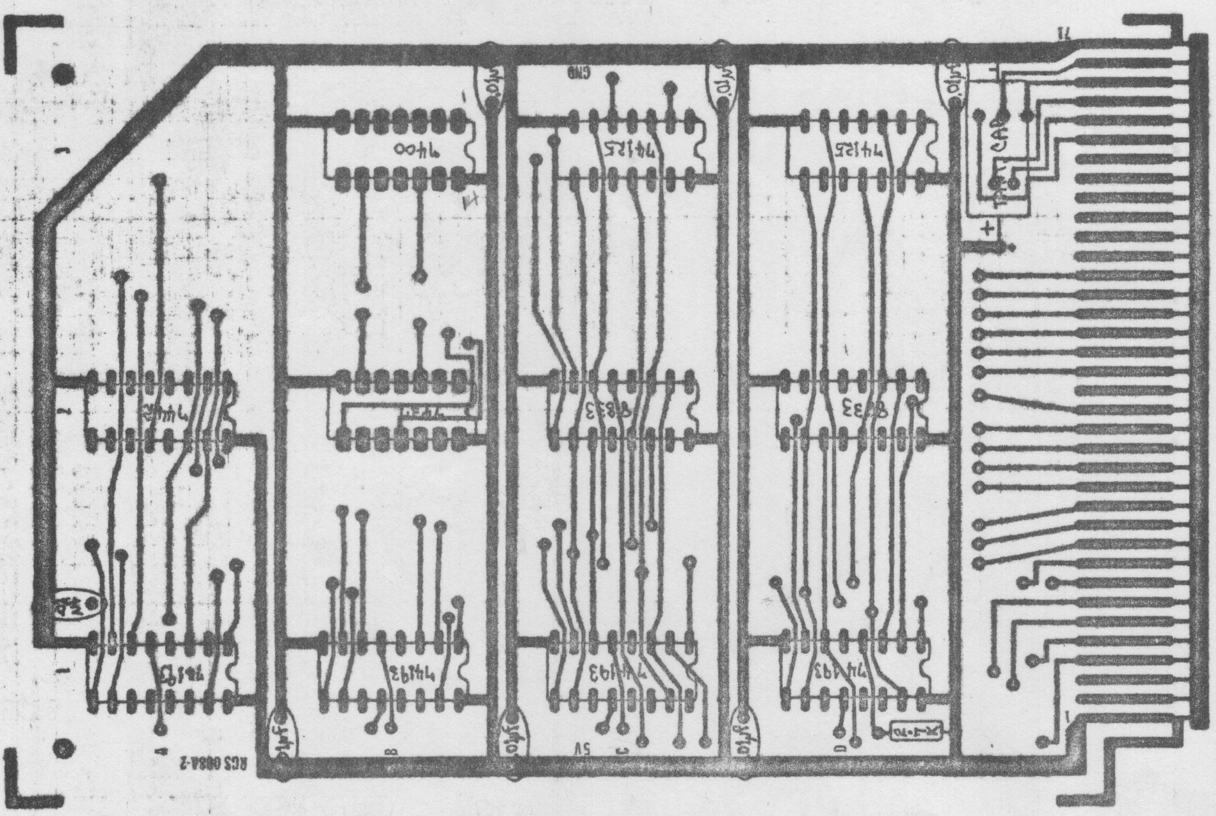
I-2-E. 008A-2 " (see p. 6).

I-2-F. 008A-1 and 008A-2 layout diagrams (see p. 7).


RGS 008A-1
58 INT. PORT EN.
61 MEM. PORT EN.
48 CCØ
11 LD. ADD. REG. ST.
13 UP. ADD. REG. ST.
56 R/W
47 CC1
1,2 +5V
3,4 -9V
71,72 GND
+ TANT.
7 x .01µf
2.7K 1/4W
1D1
1A14 2B14 3C14 5A14
1B14 2C16 3D16 5B14
1C14 2D16 4A14 5C9
1D10 3A14 4B14 5C14
2A14 3B14 4C14
2B1 2B4 2B8 2B13 3B1 3B10
1A7 2B8 3D8 5B7
1B7 2D8 4A7 5C7
1C7 3A7 4B7 1C11
2A7 3B7 4C7 1C13
2B7 3C7 5A7
57 U.A. LOAD
55 L.A. LOAD
59 MEM LOAD
20 STOP
12 MEM STEP
36 T3
49 WAIT
7408 5A
7400 2A
7400 3A
7408 1B
7404 4B
7408 3C
7408 4A
7400 5B
7400 4C
7404 1A
7404 1C
7442 2C
WAIT T3 STOP T1 T2 TII
C B A D
7474 3B
SET CLK D Q
CLK 3B Q D CLEAR
8008 1D
Ø2 Ø1 S2 S1 SØ SYNC READY INT.
DØ D2 D4 D6 D1 D3 D5 D7
8833 2D
8833 3D
74107 2B
CLOCK Q
74107 2B
CLOCK K Q CLEAR
74121 5C
.005µf
.002µf
OSC. ADJ.
220Ω
DØ 16
D2 18
D4 20
D6 22
D1 15
D3 17
D5 19
D7 21
STEP 62
WAIT/RUN 60
INTERRUPT 14
READY EXPAND. 35

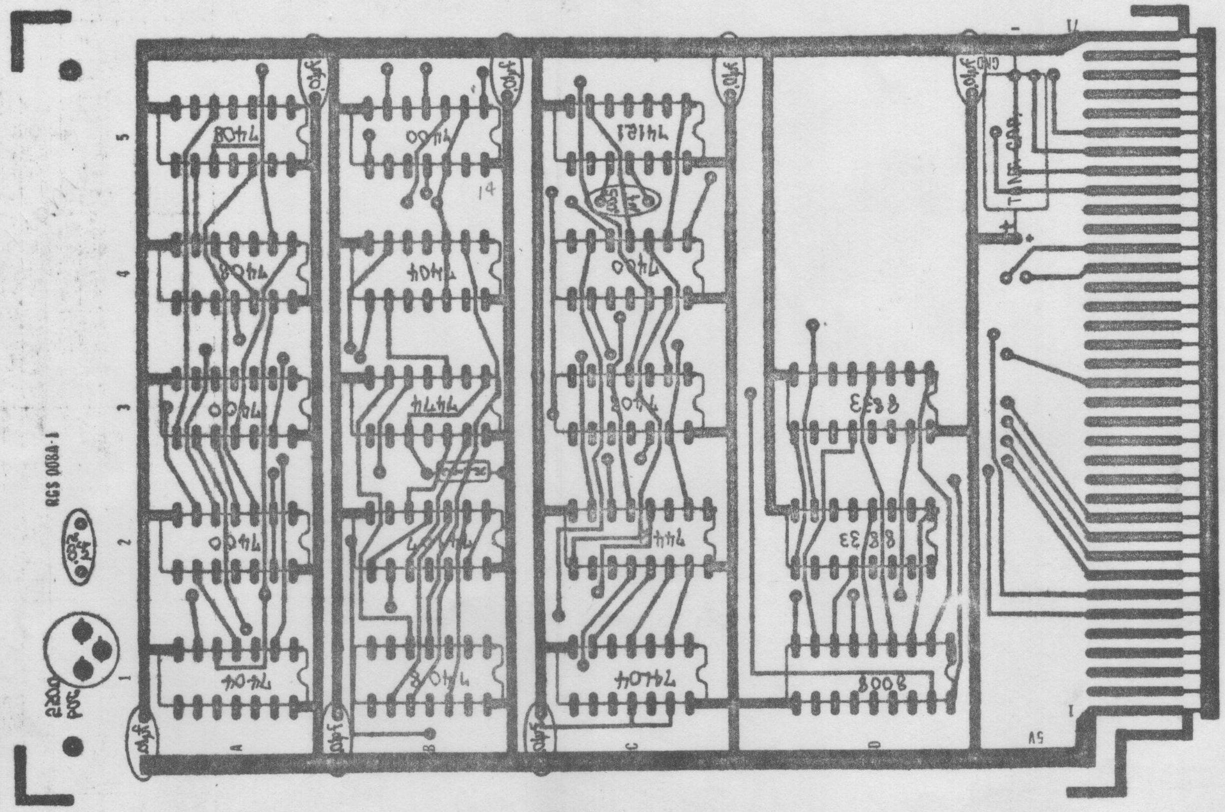

DØ 16
D1 15
D2 18
D3 17
D4 20
D5 19
D6 22
D7 21
12 MEM STEP
11 LO.ADD. REG.ST.
13 UP.ADD. REG.ST.
74193 1D
74193 1C
74193 1B
74193 1A
8833 2D
8833 2C
74125 3D
74125 3C
1,4,10,13
14,10,13
69 MOØ
67 MO1
64 MO2
66 MO3
68 MO4
70 MO5
63 MO6
65 MO7
61 MEM. PORT EN.
7400 3B
7442 2A
7437 2B
41 INPUT ST.
42 OUTPUT ST.
43 CONTROL ST.
44 SELECT ST.
36 T3
47 48 45 46 27 28 26 25 23 24 7 8 34 10 33 9
C C H H H H H H L L L L L L L L
1 Ø 5 4 3 2 1 Ø 7 6 5 4 3 2 1 Ø
29 30 39 40 21 32 37 38
I I I I I I I I
6 6 6 6 6 6 6 6
7 6 5 4 3 2 1 Ø
1,2 +5V
TANT.
7X .01µf
2.7K ¼W
1A16 2B14 3D14
1B16 2C16
1C16 2D16
1D16 3B14
2A16 3C14
1A4
1B4
1C4
1D4
71,72 GND
1A8 1C14 2C8 3C7
1A14 1D8 2D8 3D7
1B8 1D14 3B4
1B14 2A8 3B5
1C8 2B7 3B7

RGS 008A-2



TOP VIEW

008A-2



TOP VIEW

008A-1

I-3. MEMORY

I-3-A. The 008A-3 memory board is capable of 2048 words of memory. If 1K words are used, use sockets 1A, 2A, 3A, 4A, 1B, 2B, 3B, and 4B. Enable A is used to enable this section. Assemble in the following order, using the layout diagram as a reference:

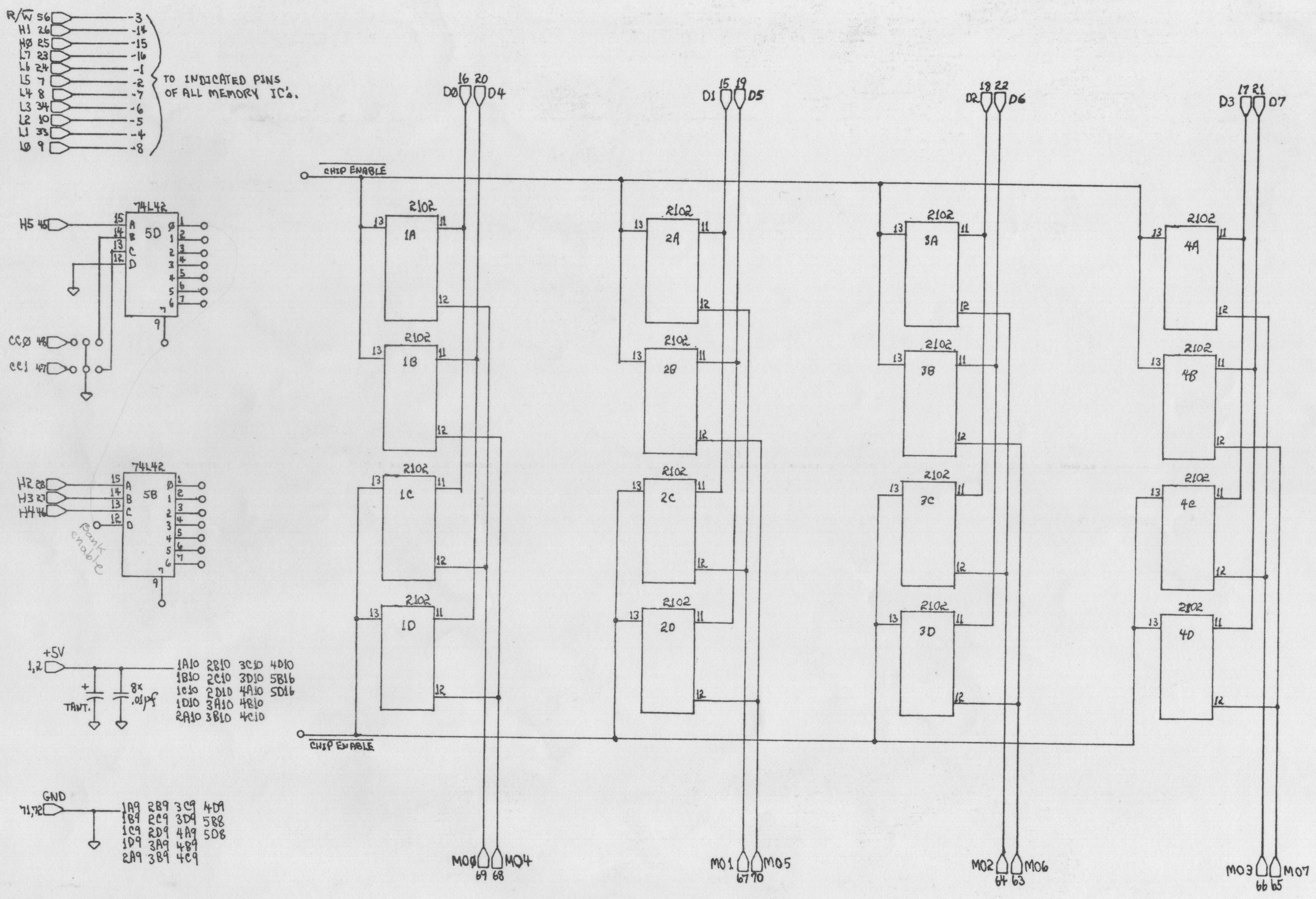
1. Jumpers: the 2 short jumpers next to IC 5D, shown on the layout diagram. 1 jumper from 5D1 to bank enable (lower 8K of memory); 1 jumper from ~~5BØ~~ 5B1 to enable A (first 1K of memory).
2. Passive components (capacitors)
3. Molex pins
4. IC's
5. Clean and check

I-3-B. 008A-3 parts list:

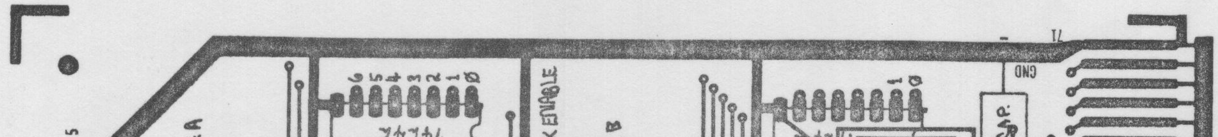
QUAN:	DESCR.
1	tantalum electrolytic cap.
8	.01mfd disc ceramic cap.
2	74L42
8	2102
20	8-pin Molex strips
1	008A-3 printed circuit board
1	72-pin edge connector
1	edge guide
1	¼" spacer
1	#4-40 x 1¼" screw
1	#4 lockwasher
1	#4-40 nut

I-3-C. 008A-3 schematic (see p. 9).

I-3-D. 008A-3 layout diagram (see p. 10).


R/W̄ 56 -3
H1 26 -14
HØ 25 -15
L7 23 -16
L6 24 -1
L5 7 -2
L4 8 -7
L3 34 -6
L2 10 -5
L1 33 -4
LØ 9 -8
TO INDICATED PINS OF ALL MEMORY IC's.
74L42
5D
H5 45
A B C D
15 14 13 12
Ø 1 2 3 4 5 6
1 2 3 4 5 6 7
7 9
CCØ 48
CC1 47
74L42
5B
H2 28
H3 27
H4 46
Bank enable
+5V
1,2
TANT.
8x .01pf
1A10 2B10 3C10 4D10
1B10 2C10 3D10 5B16
1C10 2D10 4A10 5D16
1D10 3A10 4B10
2A10 3B10 4C10
GND
71,72
1A9 2B9 3C9 4D9
1B9 2C9 3D9 5B8
1C9 2D9 4A9 5D8
1D9 3A9 4B9
2A9 3B9 4C9
CHIP ENABLE
CHIP ENABLE
DØ 16 20 D4
D1 15 19 D5
D2 18 22 D6
D3 17 21 D7
2102
1A 1B 1C 1D
2A 2B 2C 2D
3A 3B 3C 3D
4A 4B 4C 4D
11 12 13
MOØ 69 68 MO4
MO1 67 70 MO5
MO2 64 63 MO6
MO3 66 65 MO7

RGS 008A-3


RGS 0084-3
1
2
3
4
5
ENABLE A
BANK ENABLE
ENABLE B
74LS42
2102
TANT. CAP.
+
GND
5V
A
B
C
D
1
008A-3

I-4. BACKPLANE AND FRONT PANEL

I-4-A. The backplane and front panel boards are assembled in the following order, using the layout diagram as a reference:

On the 008A-5 board:

1. Resistors
2. Molex pins
3. IC's
4. LED's (watch polarity and height)
5. Switches (mount spring return switches so the part you push is on top)
6. Jumper (shown in lower right of layout diagram)

On the 008A-4 board:

7. 3 edge connectors (next to power connections)

On both boards:

8. Wire the front panel to the back plane. When finished, solder sides of both boards should face each other, and the row of holes on the backplane should be on the bottom. Drill these holes out to 1/8", and mount the edge guides in them, after inserting the boards, for alignment. The ½" spacers go between the front panel and backplane boards. Either one of the edge connectors or an unused socket position is used to wire between the 2 boards. Clip off excess pin lengths of the edge connectors when the wiring is done. The .687" spacers are used to space the 2-board assembly behind the cabinet front panel. The ¼" spacers go between the edge guide and and the p.c. socket. There are 37 wires going between the 2 boards. The I/O socket wiring comes from the backplane and will be done later.

I-4-A. Backplane and front panel (cont'd):

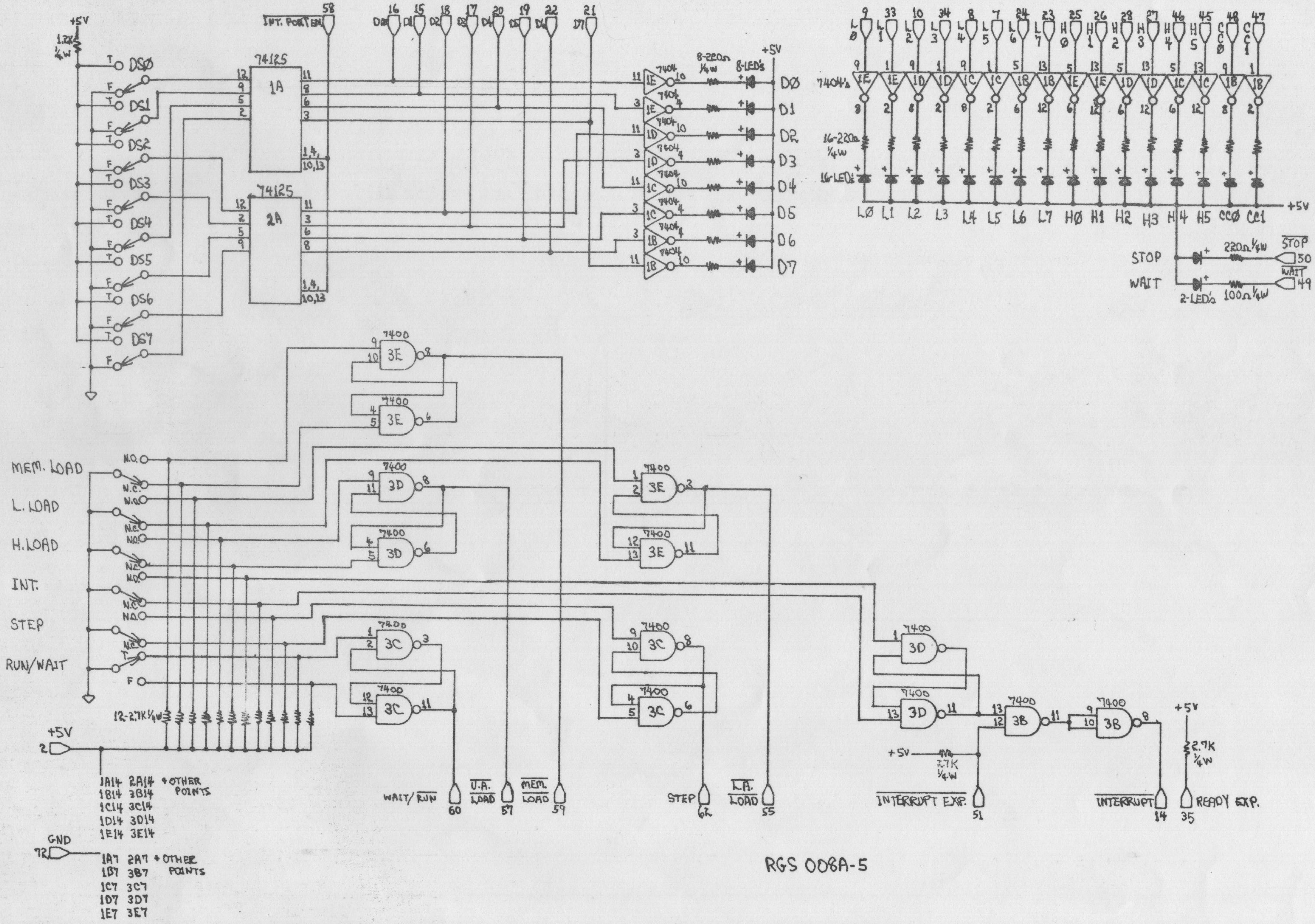
9. Clean and check

I-4-B. Backplane and front panel parts list:

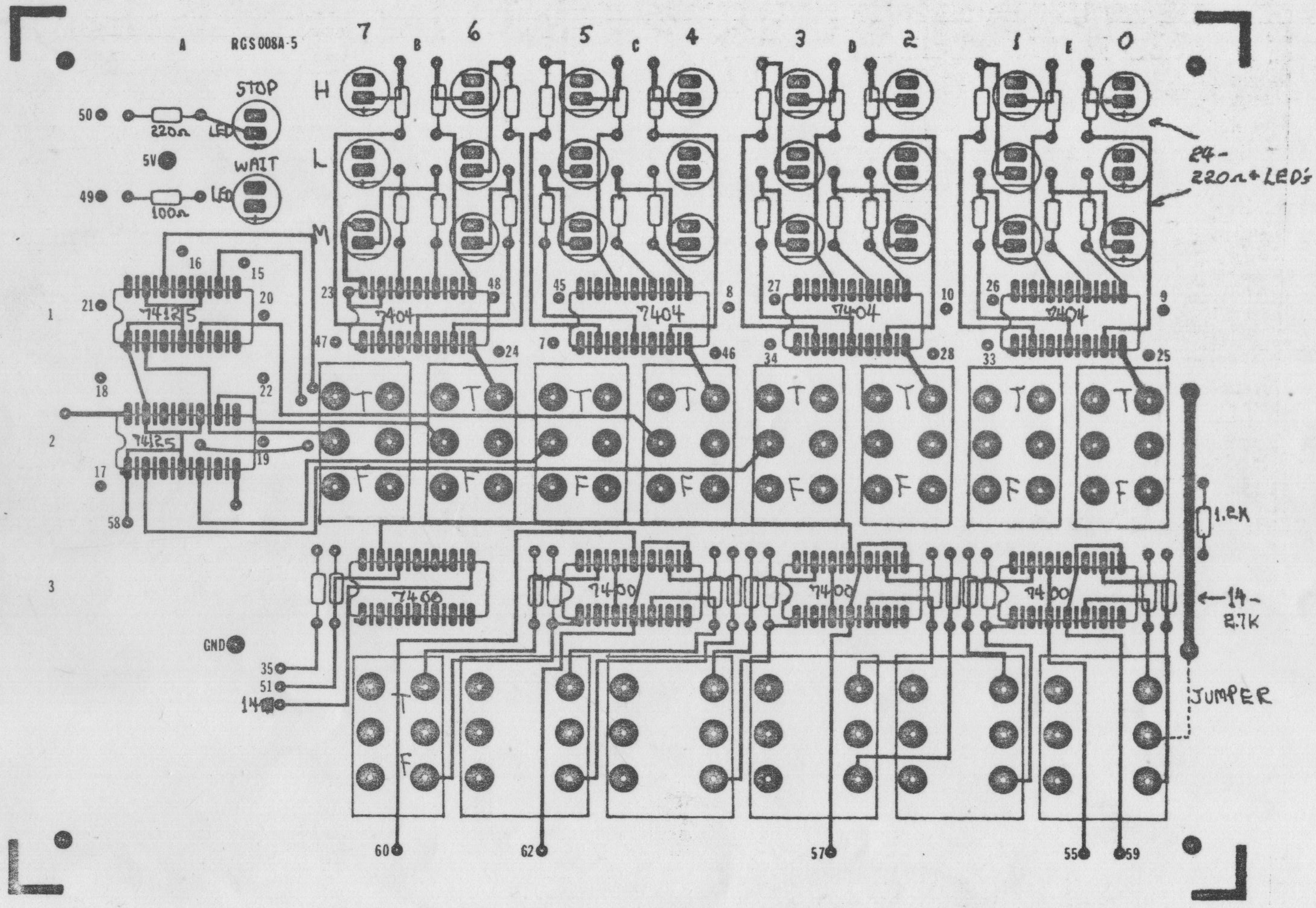
QUAN.	DESCR.
1	100 ohm ¼ watt resistor
25	220 ohm " " "
1	1.2K " " "
14	2.7K " " "
26	LED's
9	DPDT (T/F) rocker switches
5	DPDT spring return rocker switches
4	7400
4	7404
2	74125
20	7-pin Molex strips
1	008A-4 printed circuit board
1	008A-5 " " "
4	.5" spacer
4	.687" spacer
4	#4-40 x 1½" screw
4	#4 lockwasher
4	#4-40 nut

I-4-C. 008A-5 schematic (see p. 13).

I-4-D. 008A-5 layout diagram (see p. 14).

+5V
1.2K ¼W
DSØ
DS1
DS2
DS3
DS4
DS5
DS6
DS7
T
F
74125
1A
2A
INT. PORT EN
58
DØ 16
D1 15
D2 18
D3 17
D4 20
D5 19
D6 22
D7 21
7404
8-220Ω ¼W
8-LED's
DØ
D1
D2
D3
D4
D5
D6
D7
LØ 9
L1 33
L2 10
L3 34
L4 8
L5 7
L6 24
L7 23
HØ 25
H1 26
H2 28
H3 27
H4 46
H5 45
CCØ 48
CC1 47
7404's
16-220Ω ¼W
16-LED's
LØ L1 L2 L3 L4 L5 L6 L7 HØ H1 H2 H3 H4 H5 CCØ CC1
STOP
WAIT
220Ω ¼W
100Ω ¼W
2-LED's
STOP 50
WAIT 49
MEM. LOAD
L. LOAD
H. LOAD
INT.
STEP
RUN/WAIT
N.O.
N.C.
F
12-2.7K ¼W
7400
3E
3D
3C
3B
+5V
2
1A14 2A14 + OTHER POINTS
1B14 3B14
1C14 3C14
1D14 3D14
1E14 3E14
GND
72
1A7 2A7 + OTHER POINTS
1B7 3B7
1C7 3C7
1D7 3D7
1E7 3E7
WAIT/RUN 60
U.A. LOAD 57
MEM. LOAD 59
STEP 62
L.A. LOAD 55
+5V
2.7K ¼W
INTERRUPT EXP. 51
INTERRUPT 14
READY EXP. 35

RGS 008A-5



ALL RESISTORS 1/4 w

008A-5

I-5. POWER SUPPLY.

I-5-A. This is the power supply board construction section. The power supply is initially set up to produce up to 3A at 5V, 1A at +12V and 60mA at -9V. More current at 5V can be supplied, up to 5A, by changing the limiting fuse to a 5A fuse. Several things will be needed to finish construction that are not supplied: a 3-wire grounding line cord, a power switch, standoffs for the 008A-6 p.c. board and screws, a cabinet, a #2 screw and nut and a piece of vectorboard for the I/O socket, wire, fuses, and fuse holders. The 008A-6 is assembled in the following order, using the layout diagram as reference:

1. Resistors
2. Diodes (watch polarity)
3. Capacitors (watch polarity)
4. PNP transistor and 309T
5. Clean and check

Mount transformer, filter capacitor, 008A-6 p.c. board, heatsinks, 2N3055 and 340K-12. Make sure that neither the 2N3055 nor the 340K-12 is shorted to the chassis.

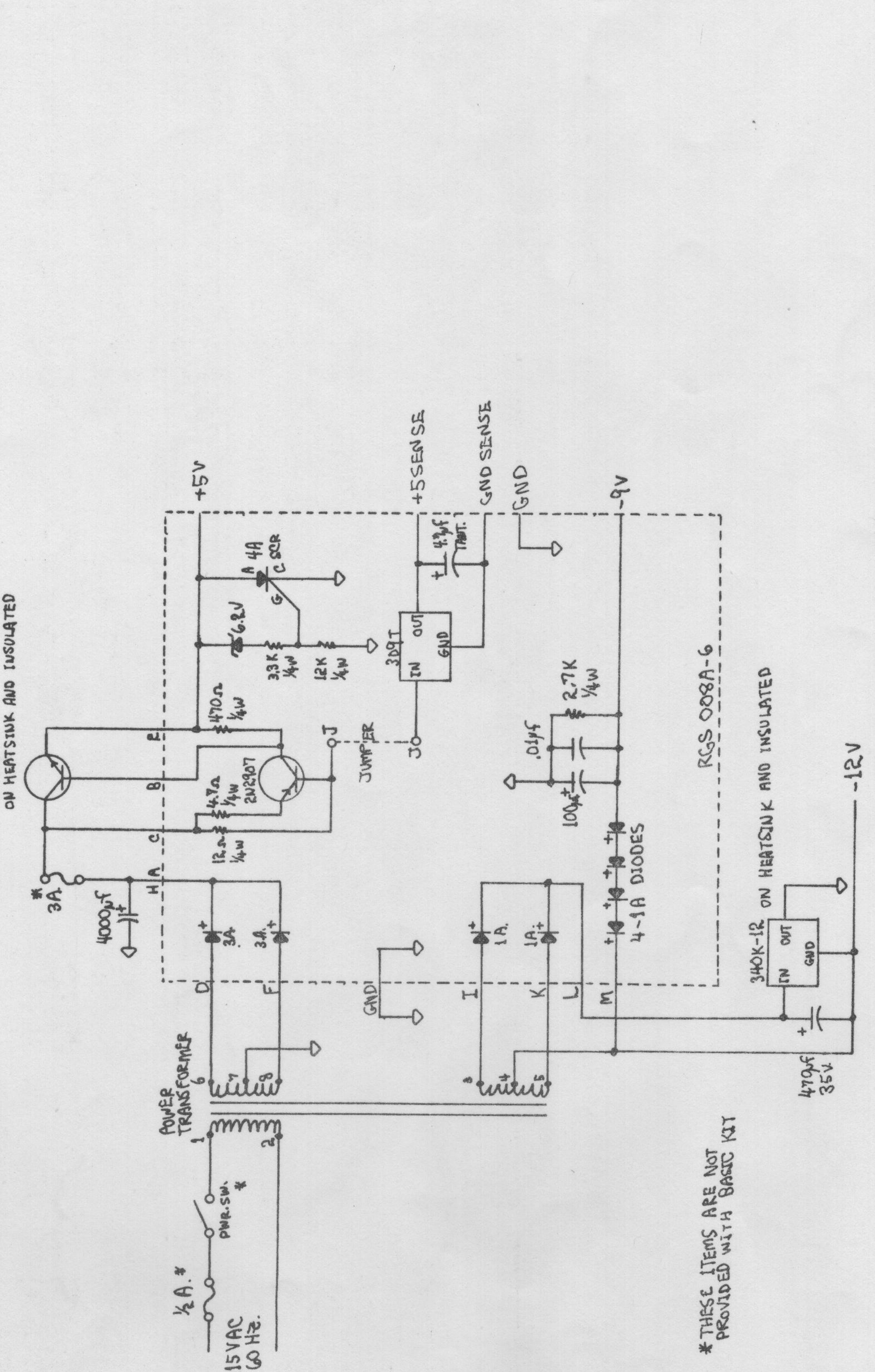
I-5-B. To check the power supply, make sure the sense lines are hooked up, + sense to +5V, - sense to ground. All voltages should be ± 10%.

I-5-C. Power supply schematic (see p. 16).

I-5-D. 008A-6 layout diagram (see p. 17).

I-5-E. Power supply parts list:

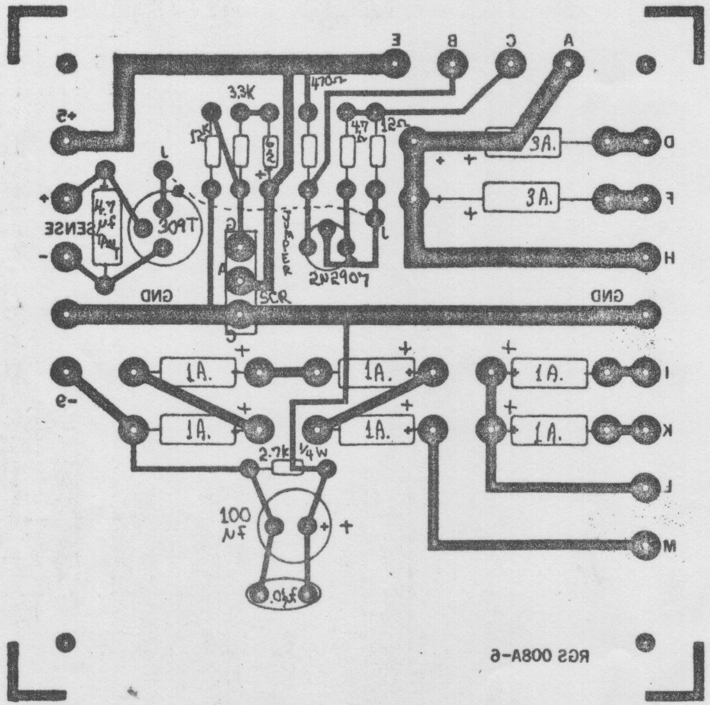
QUAN.:	DESCR.:	QUAN.:	DESCR.:
1	4.7 ohm ¼ watt res.	1	6.2V Zener diode
1	12 ohm ¼ watt res.	6	1A 50PIV silicon rect.
1	470 ohm ¼ watt res.	2	3A silicon rectifiers
1	2.7K ¼ watt res.	1	2N2907
1	12K ¼ watt res.	1	2N3055
1	3.3K ¼ watt res.	1	340K-12
1	.01mfd disc ceramic cap.	1	309T
1	4.7mfd tantalum cap.	1	4A SCR
1	100mfd electrolytic cap.	1	power transformer
1	470mfd 35wv electr. cap.	2	TO-3 heat sinks
1	4000mfd 20wv electr. cap.	2	TO-3 insulators
		1	008A-6 printed circuit board



* THESE ITEMS ARE NOT PROVIDED WITH BASIC KIT

POWER SUPPLY SCHEMATIC

TOP VIEW



008A-6

I-6. FINAL ASSEMBLY.

I-6-A. The final assembly involves putting the assembled boards in a cabinet; choice of the cabinet is up to the builder. A front panel template can be made from the front panel board layout diagram by tracing. The I/O socket and final power supply wiring is done now. The main power wires go to the pads provided on the backplane. The + sense is wired to pin 1 on the socket used for the front panel wiring. The - sense goes to pin 71.

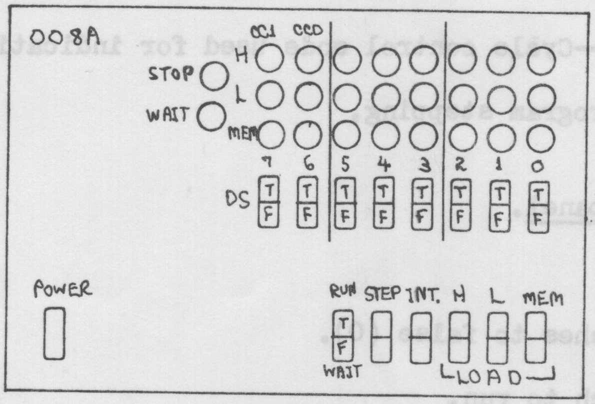
I-6-B. The I/O socket is wired according to the following list:

DESCR.	I/O SOCKET	FRONT PANEL SOCKET
I/O Ø	1	38
I/O 1	2	37
I/O 2	3	32
I/O 3	4	31
I/O 4	5	40
I/O 5	6	39
I/O 6	7	30
GND	8	GND
I/O 7	9	29
INPUT ST.	10	41
OUTPUT ST.	11	42
CONTROL ST.	12	43
SELECT ST.	13	44
READY EXP.	14	35
INTERRUPT EXP.	15	51
-	16	no connection

I-6-C. When finished, go to power on procedure to verify that the computer is working. The p.c. boards can be plugged in and edge guides put on now.

II. FRONT PANEL

II-1. Suggested front panel layout.



II-2. Description of controls and LEDs.

Switches: POWER--turns power supply on and off.

RUN/WAIT--run position is for normal, full speed operation; wait position is for stepping through the program one instruction at a time.

STEP--for stepping through the program, when in the wait mode; for incrementing the memory address register, when in the stop mode.

INT.--for putting in interrupt instructions.

The next three load switches only work in the stop mode.

H--upper address register load.

L--lower address register load.

MEM--memory load.

Data switches (8)--for interrupts and loads.

LEDs: STOP--indicates stop mode.

WAIT--indicates wait mode.

MEM (8)--indicates contents of memory location addressed by H and L.

II-2. Description of controls and LEDs. (cont'd)

LEDs: L (8)--lower address register readout.

H (6)--upper address register readout.

CC0 & CC1--Cycle control code used for indicating operation while program stepping.

II-3. How to use front panel.

II-3-1. Power on.

a) All data switches to false (0).

b) RUN/WAIT switch to run.

c) Turn POWER switch on.

d) Make sure STOP LED comes on.

II-3-2. Memory load or check.

a) Make sure STOP LED is on.

b) Set upper address on DS (data switches); T = 1, F = 0.

c) Push H switch.

d) Verify H loaded correctly.

e) Set lower address on DS.

f) Push L switch.

g) Verify L loaded correctly.

h) The MEM display now shows the contents of the selected memory location. To load that location:

i) Set DS with word to be loaded.

j) Push MEM switch.

k) Verify MEM display.

To check sequential memory locations, do a) through h), then push step switch to get the next location.

II-3-3. To start a program. (Program starting at 00,000)

a) Set DS to 005 (explanation of octal in programming section).

b) Push INT switch.

II-3-4. To stop a program.

a) Set DS to 000,001 or 377.

b) Push INT switch.

Note: The CPU will accept any single word instruction from the data switches, by means of the INTERRUPT switch, without stopping the computer.

II-3-5. To step through a program.

a) RUN/WAIT switch to wait.

If program is not started, go through starting procedure (II-3-3).

b) WAIT LED comes on.

c) Use STEP switch to step through program.

d) When finished, set RUN/WAIT switch to run.

CC (Cycle control code) indicates the operation being performed. One program step is actually one machine cycle step. The programming section indicates how many machine cycles and what CC's are involved.

III. PROGRAMMING

III-1. Formats. The 008A uses octal code for convenience. The 8 binary bits are represented thus:

```
2 1  4 2 1  4 2 1

0 0  0 0 0  0 0 0
```

This is true for memory data, instructions, and lower addresses; however, only the lower 6 bits are used for the upper addresses. The full octal designation for location zero is 00,000. The upper 2 octal digits indicate the page number; the lower 3 octal digits indicate the location in that page (000 to 377). 1024 words of memory is divided into 4 pages, e.g., pages 00 to 03.

III-2. Instructions. This section includes a description of each instruction with notes. A short form version will follow for easy reference.

III-2-1. ODO Increment index register D ≠ 0, D ≠ 7
1 machine cycle, 1 byte, CC* 0, CF+ Z, S, P

Notes: D is the destination register. The 8008 contains 7 registers, the A register or accumulator and 6 other index registers. This particular instruction may only be used to increment the 6 index registers, but in other instructions D may also address the A register and memory. This is a complete chart of registers and their addresses:

0	A register or accumulator
1	B register
2	C "
3	D "
4	E "
5	H "
6	L "
7	MEMORY location addressed by the contents of H and L

In this instruction D can only equal 1-6, because 000 is a Halt instruction, and this instruction may not be used to increment memory.

* CC = cycle control code

+ CF = condition flipflops affected

III-2-1. (cont'd) This instruction can also affect the condition flipflops. There are 4 condition flipflops:

C Carry flipflop — Sets to True (1) as the result of an underflow or overflow, i.e., if you add 1 to all 1's, or subtract 1 from all 0's.

Z Zero flipflop — Sets to True (1) when all bits go to 0 during an operation.

S Sign flipflop — Sets to True (1) when the MSB of the result of an operation is 1.

P Parity flipflop — Sets to True (1) when the parity of a result is true, i.e., when the result has an even number of 1's. Parity should alternate in Increment index register instructions.

These bits are set as a result of each arithmetic and logical operation. Except in the case of add-with-carry and subtract-with-borrow operations, the condition flipflops never interact with subsequent arithmetic and logical operations; they only indicate states at the close of the current operation. They permit conditional branching through conditional Calls, Jumps, and Returns. The carry bit also makes possible multiple precision binary arithmetic.

III-2-2. 0D1 Decrement index register D $\neq$ 0, D $\neq$ 7

1 machine cycle, 1 byte, CC 0, CF Z, S, P

Notes: This instruction works exactly the same as the Increment Index Register instruction, except that the content of D decreases by one instead of increasing by one.

III-2-3. 0E2 Rotate A register E = 0 - 3

1 machine cycle, 1 byte, CC 0, CF C

Notes: This instruction rotates the contents of the A register 1 bit either to the left or to the right, through or past the carry flipflop.

When E equals: The operation performed is:

0 Rotate contents of A register to the left one bit. A_7 goes into A_0 and into carry.

III-2-3. (cont'd)

When E equals: The operation performed is:

1 Rotate contents of A register to the right one bit. A_0 goes into A_7 and into carry.

2 Rotate contents of A register to the left one bit; content of carry flipflop goes into A_0, A_7 goes into carry.

3 Rotate contents of A register to the right one bit; content of carry flipflop goes into A_7, A_0 goes into carry.

III-2-4. 0F4 ALU immediate B_2 = data

2 machine cycles, 2 bytes, CC 0,2, CF C, Z, S, P

Notes: F is the operation performed by the ALU using the contents of the A register and the data in B_2.

When F.is: The operation is:

0 ADD: Add the data in B_2 to the contents of the A register and store the result in the A register. An overflow sets the carry flipflop to True (1).

1 ADD WITH CARRY: Add the data in B_2 and the contents of the carry flipflop to the contents of the A register, and store the result in the A register. An overflow sets the carry flipflop to True (1). C only affects the result of the operation if C = 1 when the operation begins. If the operation doesn't generate a carry, it will clear C, setting it to 0.

2 SUBTRACT: Subtract the data in B_2 from the contents of the A register, and store the result in the A register. An underflow (borrow) sets the carry flipflop to 1. Uses two's complement subtraction.

3 SUBTRACT WITH BORROW: Subtract B_2 and the contents of C from the contents of the A register, and store the result in the A register. As in ADD WITH CARRY, C only affects the result of the operation if C = 1 when the operation begins; and if the operation doesn't generate a borrow, it will set C to 0.

4 LOGICAL AND: Logically AND B_2 and the contents of the A register, and store the result in the A register. When two binary numbers are ANDed, the only columns in the answer which contain a 1 are the columns in which both original numbers had a 1:

$$\begin{array}{lr} A & 11\ 001\ 110 \\ B_2 & \underline{01\ 101\ 101} \\ & 01\ 001\ 100 \end{array}$$

III-2-4. (cont'd)

When F is: The operation is:

5 EXCLUSIVE-OR: Place the exclusive-OR of B_2 and the contents of the A register in the A register. The result of an exclusive-OR operation contains a 1 only in the columns where one or the other, but not both, of the original numbers had a 1:

$$\begin{array}{cr} A & 01\ 011\ 101 \\ B_2 & 11\ 001\ 010 \\ \hline & 10\ 010\ 111 \end{array}$$

6 INCLUSIVE-OR: Place the inclusive-OR of B_2 and the contents of the A register in the A register. The result of an inclusive-OR operation contains a 1 in all columns where either or both of the original numbers had a 1:

$$\begin{array}{cr} A & 01\ 011\ 101 \\ B_2 & 11\ 001\ 010 \\ \hline & 11\ 011\ 111 \end{array}$$

7 COMPARE: Subtracts B_2 from contents of A register. The content of the A register remains unchanged; only the condition flipflops are affected. If $A = B_2$, Z is set to 1. If $A < B_2$, C is set to 1. If $A > B_2$, C and Z are both 0.

All condition flipflops can be affected by these operations. All three logical operations reset C to 0. You can clear C without changing the A register by ANDing immediate with all 1's (377_8). You can invert (or complement) the content of the A register by performing exclusive-OR immediate with all 1's (377_8).

III-2-5. 2FS ALU register F = 0 - 7; S = source register

1 machine cycle, 1 byte, CC 0, CF C, Z, S, P

Notes: F is the operation performed by the ALU using the contents of the A register and the contents of the source register, S. The result is stored in the A register. The operations and effects on condition flipflops are the same as those listed for F under "ALU Immediate". The address digits for S are the same as those listed for D under "Increment Index Register".

When S = 7, the instruction has 2 machine cycles, 2 bytes, and the CC is 0,2. The H and L registers are normally used to store the addresses of memory words to be used during arithmetic and logical operations.

III-2-5. (cont'd) The H register contains the page.number or upper address; the L register contains the word number or lower address. Therefore, when S = 7, the computer checks the H and L registers, and takes the word from memory which is stored at the location addressed by the contents of those registers. In writing programs, it is important to be sure that H and L contain the address of the correct word before you write an "ALU Register" instruction where S = 7.

III-2-6. 0D6 Load data immediate D = destination register

2 machine cycles, 2 bytes, CC 0,2

Notes: The data in B_2 is loaded into the destination register addressed by D. The address digits for D are the same as those listed under "Increment Index Register".

When D = 7, indicating load memory, this is a 3 machine cycle instruction, CC 0,2,3. The data in B_2 is then loaded into the memory location addressed by the contents of registers H and L. You can load specific memory locations by loading H and L with the desired address, then loading memory.

III-2-7. 3DS Load data D and S ≠ 7, because 377 = HALT

1 machine cycle, 1 byte, CC 0

Notes: Used to load data into registers, or to move data from one register to another. D = destination register, S = source register; addresses are as listed for D under "Increment Index Register". If the instruction reads 30X, load the A register with the contents of register X, register X remains unchanged by the operation.

If the instruction reads 37X, transferring the contents of register X into the memory location addressed by registers H and L, this is a 2 machine cycle instruction, and the CC is 0,2.

If the instruction reads 3X7, transferring data from memory to register X, it is a 2 machine cycle instruction, and the CC is 0,3. In this case data would come from the location addressed by registers H and L.

III-2-7. (cont'd) Transferring the contents of a register to itself, for example 300, load A with contents of A, is a NOP (no operation). NOPs can be used as fillers, saving certain memory locations within the program for later data and/or instructions.

III-2-8. 1X4 Unconditional jump X = "don't care"

3 machine cycles, 3 bytes, CC 0,2,2

Notes: This instruction automatically shifts the program to the instruction located at the memory address designated by B_2 (lower address) and B_3 (upper address). After the jump the program continues in sequence from the new location. The middle digit, X, has no effect on the jump instruction; it can be used to code groups of jumps.

III-2-9. 1C0 Conditional jump C = condition: 0 - 7

3 machine cycles, 3 bytes, CC 0,2,2

Notes: The shift to the instruction located at the address in B_2 B_3 takes place if the condition specified in C is met.

If C is:	The jump takes place if:
0	C = 0; carry false
1	Z = 0; zero false
2	S = 0; sign false (MSB of previous result was 0)
3	P = 0; parity false (odd)
4	C = 1; carry true
5	Z = 1; zero true
6	S = 1; sign true (MSB of previous result was 1)
7	P = 1; parity true (even)

The conditional jump does not affect the condition flipflops, merely checks them to see if the condition enabling the jump exists. If the condition does not exist, the program goes to the next instruction in sequence.

III-2-10. 1X6 Unconditional call X = "don't care"

3 machine cycles, 3 bytes, CC 0,2,2

Notes: This instruction jumps the program unconditionally to the address defined in B_2 B_3. The address in the P register (program counter) immediately following B_3 is stored in the pushdown stack, III-2-10. (cont'd) which pops down one level. This instruction and its companion, the unconditional return, are used for often-repeated subroutines. Subroutines may be called and nested up to 7 levels.

III-2-11. 0X7 Unconditional return X = "don't care"

1 machine cycle, 1 byte, CC 0

Notes: The uppermost address in the pushdown stack (i.e., the last one to be stored there) goes into the P register, and the program jumps to the instruction located at that address. The pushdown stack pops up one level. This instruction is used to exit from subroutines.

III-2-12. 1C2 Conditional call C = condition; 0 - 7

3 machine cycles, 3 bytes, CC 0,2,2

Notes: The call proceeds as detailed under "Unconditional Call", IF the condition specified by C exists. Values for C are the same as the ones listed under "Conditional Jump". Like the conditional jump, the conditional call does not affect the condition flipflops. If the condition does not exist to enable the call, the program continues to the next instruction in sequence.

III-2-13. 0C3 Conditional return C = condition; 0 - 7

1 machine cycle, 1 byte, CC 0

Notes: The return proceeds as detailed under "Unconditional Return", IF the condition specified by C exists. Values for C are the same as the ones listed under "Conditional Jump". The conditional return does not affect the condition flipflops. If the condition does not exist, the program continues to the next instruction in sequence.

III-2-14. 0A5 Restart A = 0 - 7

1 machine cycle, 1 byte, CC 0

Notes: This is a one-instruction unconditional call; the address immediately following the restart instruction goes into the pushdown stack, and the program shifts to the location on page 00 indicated by the value of A.

III-2-14. (cont'd)

When A is:	The program goes to location:
0	00 000
1	00 010
2	00 020
3	00 030
4	00 040
5	00 050
6	00 060
7	00 070

Since the 8 locations addressed by this instruction are 8 words apart, the instruction can be used to store 8 8-byte subroutines in the lower 64 words of memory. Use a return instruction to get back, as for any other call.

III-2-15. 1R1 Input/Output R = 0, 2, 4, or 6

2 machine cycles, 1 byte, CC 0,1

Notes: This instruction uses the contents of the A register to manipulate the I/O bus, to get information into and out of the computer.

When R equals: The instruction is:

0 INPUT: 1 byte comes into the A register from whatever input device has been selected on the I/O bus. Changes the content of A.

2 OUTPUT: The data word in the A register goes to be loaded into whatever output device has been selected on the I/O bus. Content of A is not changed.

4 CONTROL: This indicates that the word in the A register is a control word, telling the peripheral what to do, rather than a data word.

6 SELECT: Load the A register with the address of the device you wish to select, then send a SELECT instruction. Selecting a device connects it electrically. To clear the bus, select a non-existent device.

III-2-16. 000
001 Halt
377

1 machine cycle, 1 byte, CC 0

Notes; Suspends the activity of the processor in the STOPPED state. Content of registers and memory is unchanged. The program counter

III-2-16. (cont'd) advances one instruction and the internal dynamic memories continue to be refreshed.

III-3. Instruction reference sheet.

MNEMONIC	CODE	INSTRUCTION	NO. BYTES	CONDITION FLIPFLOPS AFFECTED
HLT	000	Halt	1	none
HLT	001	"	1	none
HLT	377	"	1	none
IN(D)	0D0	Increment index register (D ≠ 0, D ≠ 7)	1	Z, S, P
DC(D)	0D1	Decrement index register (D ≠ 0, D ≠ 7)	1	Z, S, P
R(E)	0E2	Rotate (E = 0 to 3)	1	C
R(C)	0C3	Conditional return	1	none
(F)I	0F4	ALU Immediate (B_2 = data)	2	C, Z, S, P (logical operations reset C to false)
RS(A)	0A5	Restart (address 00,0A0)	1	none
L(D)I	0D6	Load data immediate (B_2 = data)	2	none
RET	0X7	Unconditional return	1	none
J(C)	1C0	Conditional jump (B_2, B_3 = address)	3	none
(R)	1R1	Input/output	1	none
C(C)	1C2	Conditional call (B_2, B_3 = address)	3	none
JMP	1X4	Unconditional jump (B_2, B_3 = address)	3	none
CAL	1X6	Unconditional call " " "	3	none
(F)(S)	2FS	ALU Register	1	C, Z, S, P (logical operations reset C to false)
L(D)(S)	3DS	Load data (D and S ≠ 7)	1	none

X = don't care
B_2 = data or lower address
B3 = upper address
D = destination register
S = source register

D or S	REGISTER
0	A (accumulator)
1	B } Index registers
2	C
3	D
4	E
5	H
6	L
7	MEM (address defined by contents of H and L)

R	FUNCTION
0	Input (INP)
2	Output (OUT)
4	Control (CON)
6	Select (SEL)

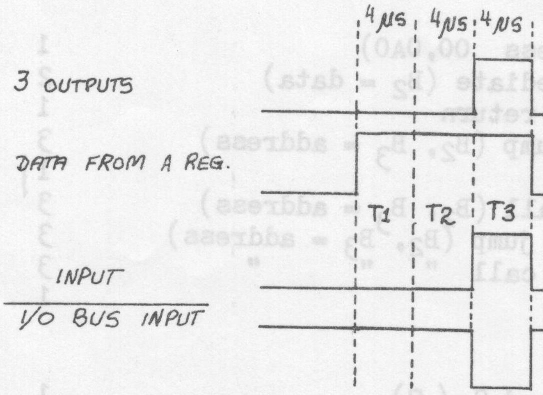
	C	CONDITION
(F)	0	Carry false (C)
	1	Zero false (Z)
	2	Sign false (S, +)
	3	Parity odd (P, false)
(T)	4	Carry true (C)
	5	Zero true (Z)
	6	Sign true (S, -)
	7	Parity even (P, true)

F	OPERATION WITH A REG.
0	Add (AD)
1	Add with carry (AC)
2	Subtract (SU)
3	Subtract with borrow (SB)
4	AND (ND)
5	Exclusive OR (XR)
6	Inclusive OR (OR)
7	Compare (CP)

E	ROTATION
0	Left, A_7 into carry (LC)
1	Right, A_0 into carry (RC)
2	Left through carry (AL)
3	Right through carry (AR)

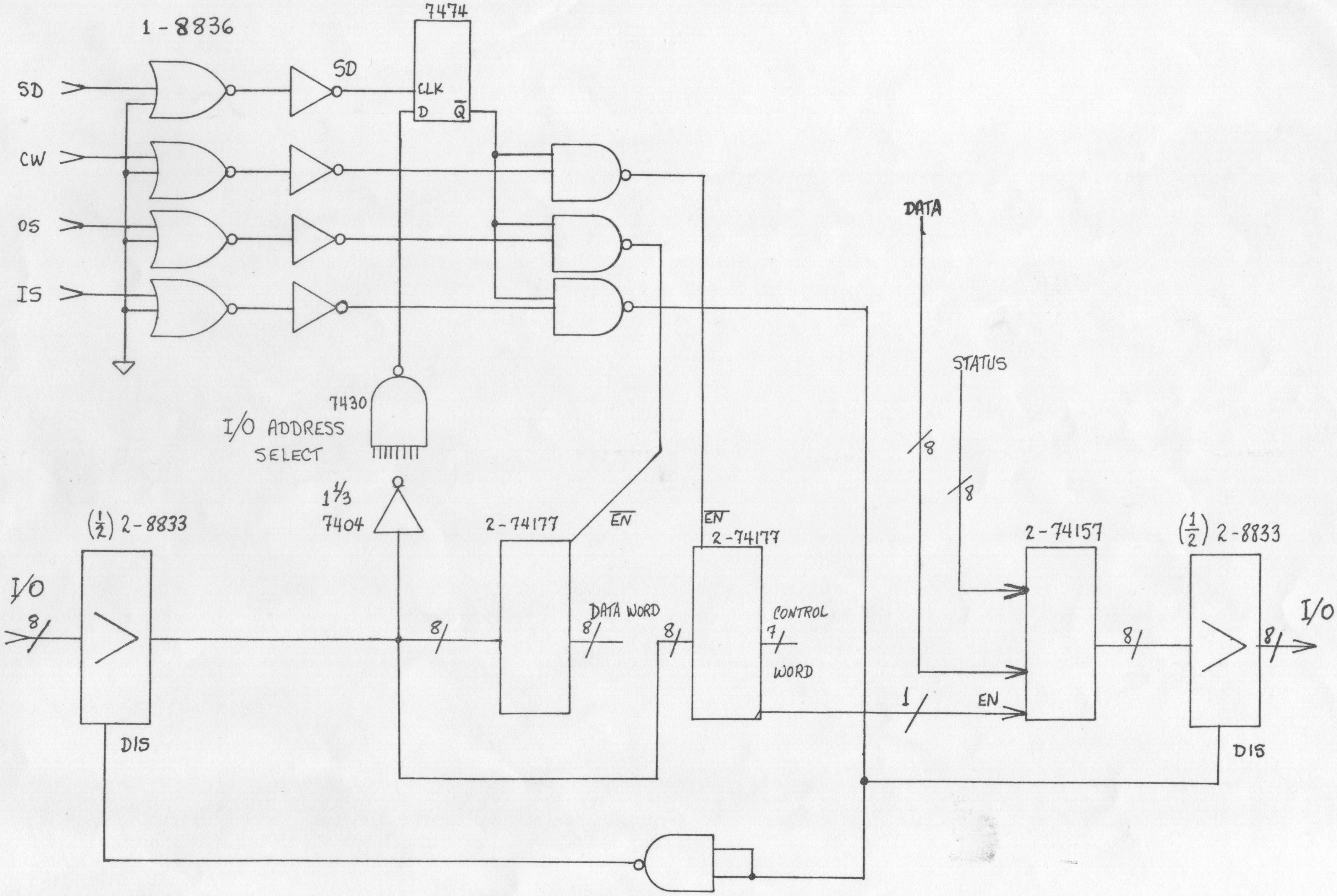
IV. INPUT/OUTPUT

IV-1. Signals. There are four I/O signals available: Input.(INP), Output.(OUT), Control (CON), and Select (SEL). They are active high and available for 4 microseconds. The leading edge of the 3 output signals can be used to strobe data because data is already available. See diagram:



The ready line is also on the I/O cable. When the ready line is brought low, the CPU stops between T_2 and T_3 (WAIT) until ready goes high. Interrupt can be used externally, but this is not recommended with minimum memory because it requires more complicated software. Our suggested universal parallel interface is set up so the status can be read to find out if an I/O device is ready. The block diagram of the interface is on the next page; there will be a kit available for it. Almost any parallel device can be interfaced to the 008A with this interface. More interfacing information will appear in future supplements to this manual, depending on what is requested by the users.

PARALLEL INTERFACE BLOCK DIAGRAM



IV-2. Programming considerations. There are 4 steps to be considered in working with an I/O device:

1. Initialization
2. Control
3. Status
4. Data transfer

Initialization: The condition of the device has to be determined. All necessary startup routines must be done.

Control: The operating mode of the device must be set.

Status: The time at which the device is ready to input or output must' be determined.

Data transfer is self-explanatory.

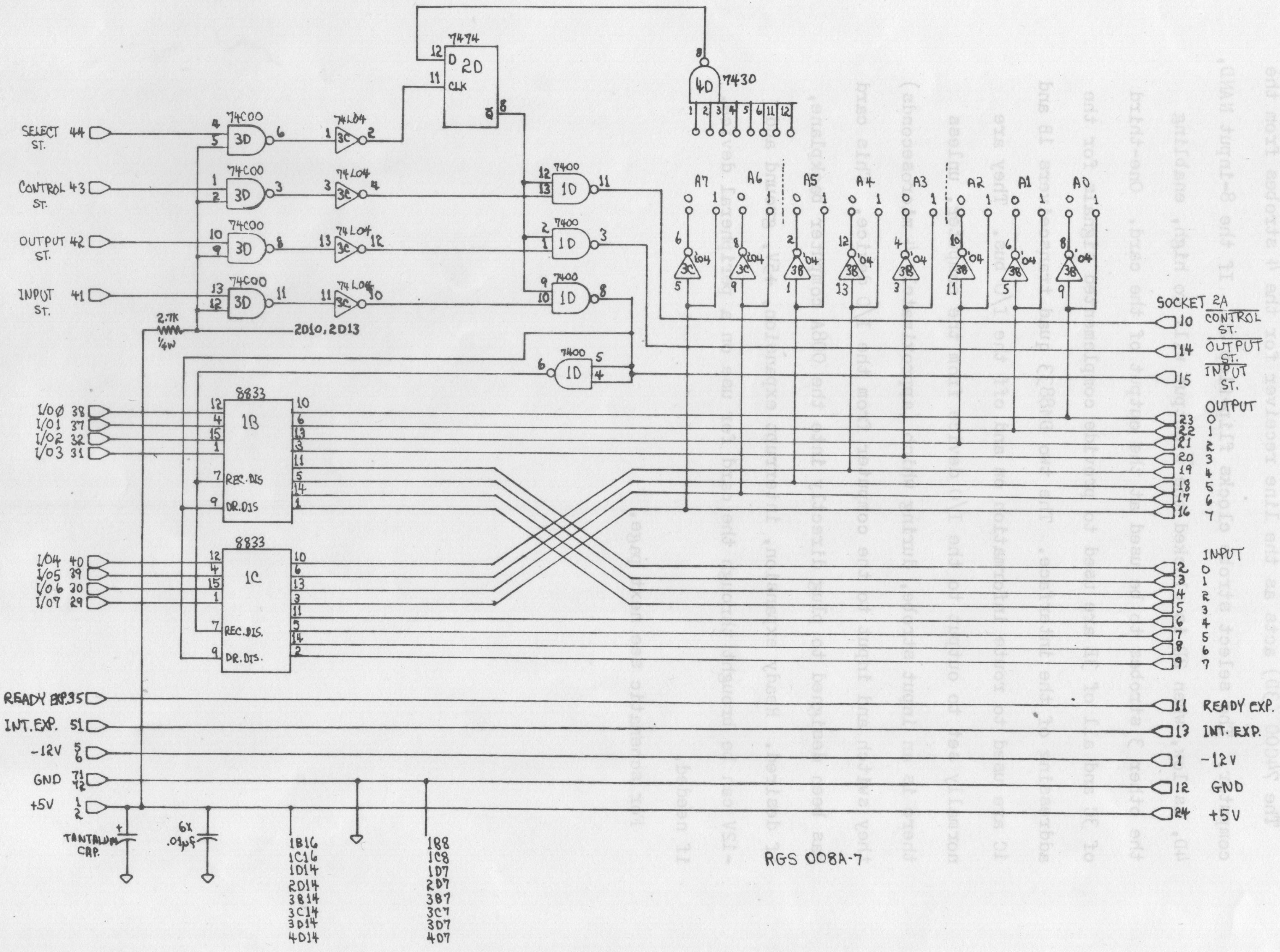
These steps are interrelated to a great degree and must be thought out. Examples and common subroutines will be included in future supplements.

IV-3. PERIPHERAL SCHEMATICS & DESCRIPTIONS.

IV-3-A. Parallel interface.

The 74C00 (3D) acts as the line receiver for the 4 strobes from the computer. The select strobe clocks flip-flop 2D. If the 8-input NAND, 4D, is low, when 2D is clocked the $\overline{Q}$ output will go high, enabling the other 3 strobes to be used at the output of the card. One-third of 3C and all of 3B are used to provide complemented signals for the addressing of the interface. The two DM8833 quad transceivers 1B and 1C are used to route information on and off the I/O bus. They are normally set to output to the I/O device from the computer, unless there is an input strobe, during which (approximately 4 microseconds) they switch and input to the computer from the I/O device. This card has been designed to plug directly into the 008A computer backplane, if desired. Ready expansion, interrupt expansion, +5V, ground and -12V can be brought through the card for use on a peripheral device, if needed.

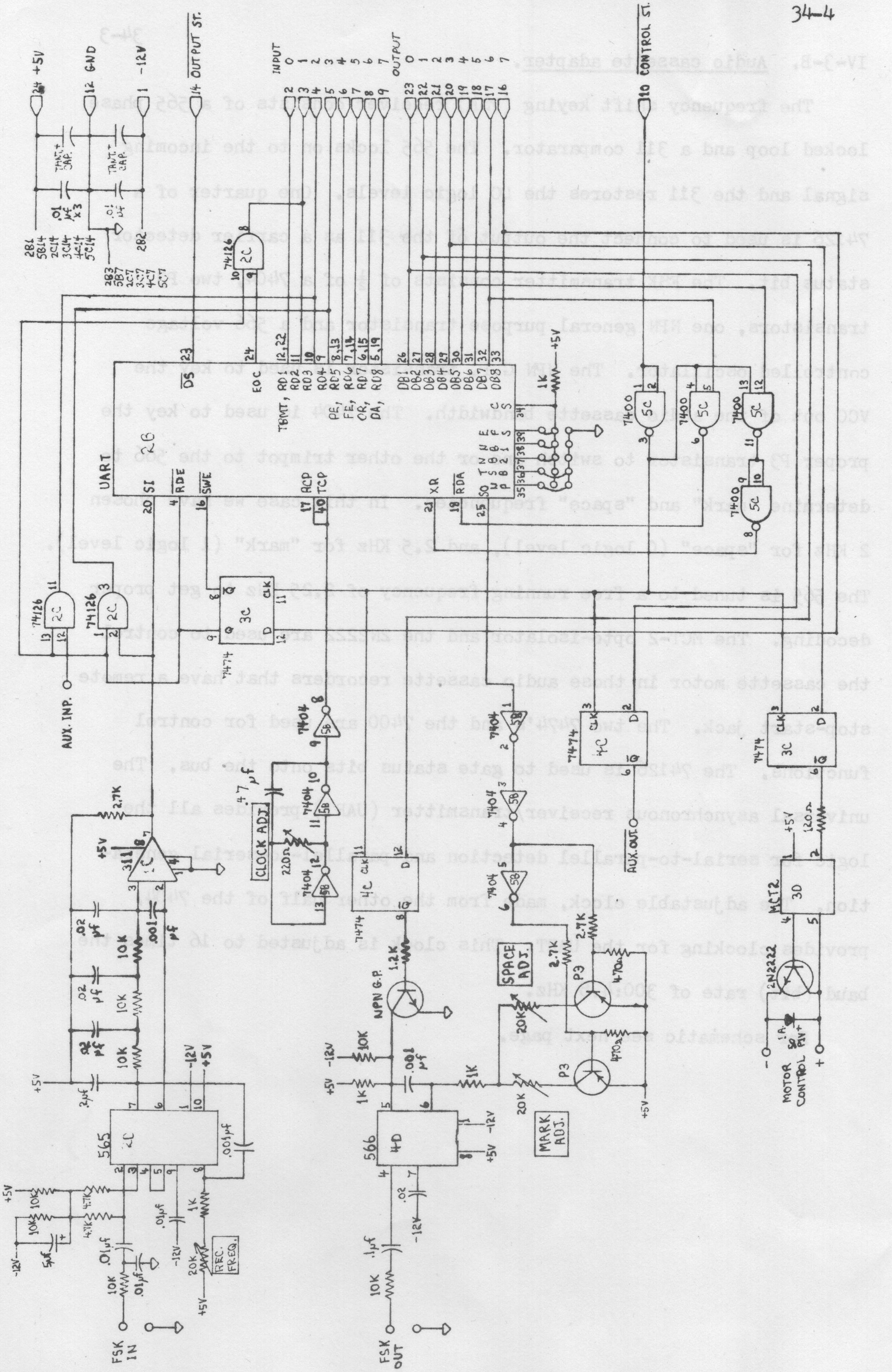
For schematic see next page.


SELECT ST. 44
CONTROL ST. 43
OUTPUT ST. 42
INPUT ST. 41
74C00 3D
74L04 3C
7474 2D
D
CLK
7430 4D
7400 1D
A7
A6
A5
A4
A3
A2
A1
A0
'04 3C
'04 3B
2.7K 1/4W
2D10, 2D13
SOCKET 2A
CONTROL ST. 10
OUTPUT ST. 14
INPUT ST. 15
OUTPUT 0 1 2 3 4 5 6 7
23 22 21 20 19 18 17 16
8833 1B
8833 1C
REC. DIS.
DR. DIS.
I/0Ø 38
I/01 37
I/02 32
I/03 31
I/04 40
I/05 39
I/06 30
I/07 29
INPUT 0 1 2 3 4 5 6 7
READY EXP. 35
INT. EXP. 51
-12V 5 6
GND 71 72
+5V 1 2
READY EXP. 11
INT. EXP. 13
-12V 1
GND 12
+5V 24
TANTALUM CAP.
6X .01µF
1B16 1C16 1D14 2D14 3B14 3C14 3D14 4D14
1B8 1C8 1D7 2D7 3B7 3C7 3D7 4D7
RGS 008A-7

IV-3-B. Audio cassette adapter.

The frequency shift keying (FSK) receiver consists of a 565 phase locked loop and a 311 comparator. The 565 locks on to the incoming signal and the 311 restores the DC logic levels. One quarter of a 74126 is used to connect the output of the 311 as a carrier detector status bit. The FSK transmitter consists of ½ of a 7404, two P3 transistors, one NPN general purpose transistor and a 566 voltage controlled oscillator. The NPN G.P. transistor is used to key the VCO out of the audio cassette bandwidth. The 7404 is used to key the proper P3 transistor to switch one or the other trimpot to the 566 to determine "mark" and "space" frequencies. In this case we have chosen 2 KHz for "space" (0 logic level), and 2.5 KHz for "mark" (1 logic level). The 565 is tuned to a free running frequency of 2.25 KHz to get proper decoding. The MCT-2 opto-isolator and the 2N2222 are used to control the cassette motor in those audio cassette recorders that have a remote stop-start jack. The two 7474's and the 7400 are used for control functions. The 74126 is used to gate status bits onto the bus. The universal asynchronous receiver/transmitter (UART) provides all the logic for serial-to-parallel detection and parallel-to-serial generation. The adjustable clock, made from the other half of the 7404, provides clocking for the UART. This clock is adjusted to 16 times the baud (bit) rate of 300:4.8 KHz.

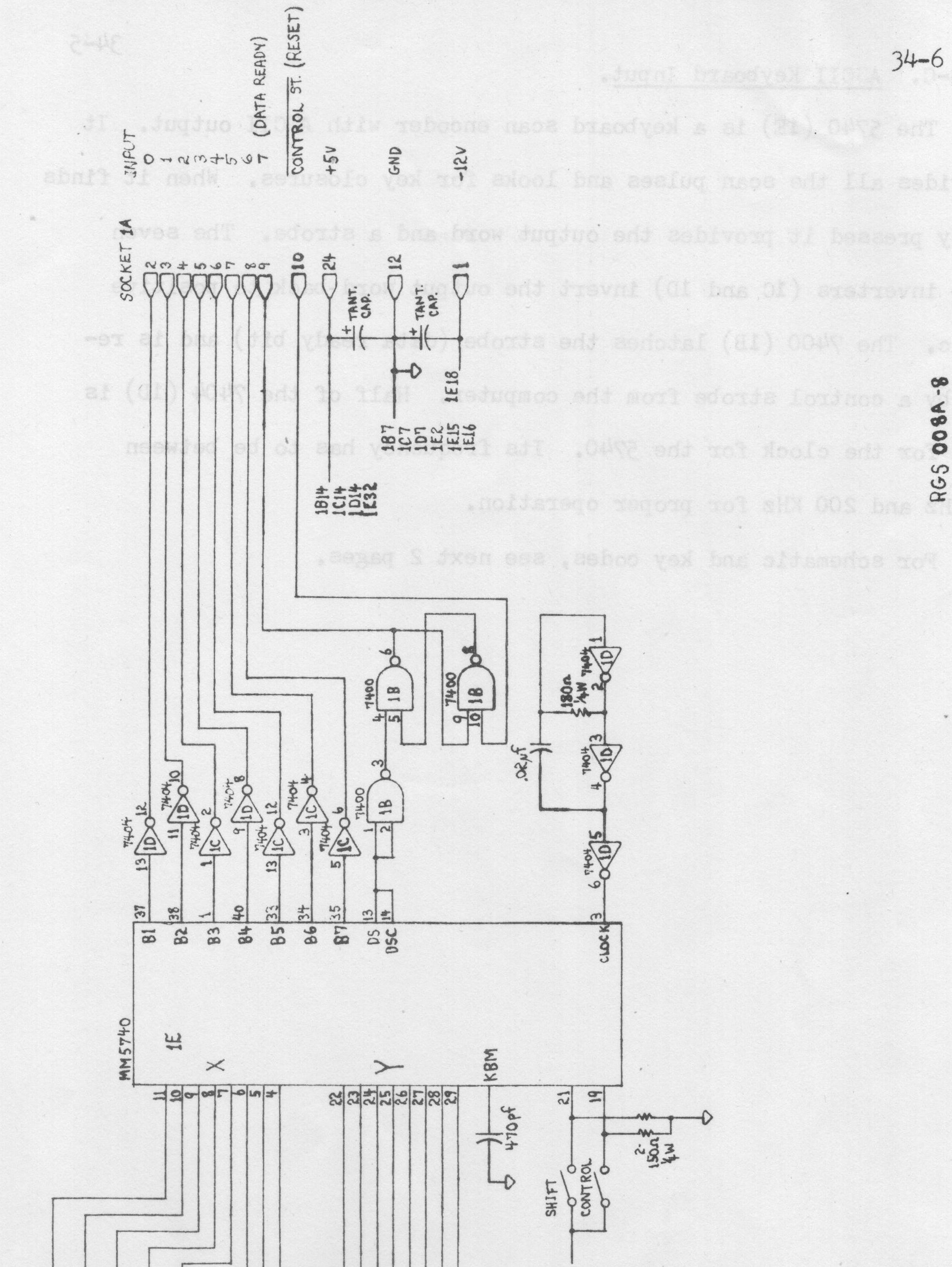
For schematic see next page.


RGS 008A-9
24 +5V
12 GND
1 -12V
14 OUTPUT ST.
TANT. CAP.
.01 µf X3
.01 µf
2B1 5B14 2C14 3C14 4C14 5C14
2B3 5B7 2C7 3C7 4C7 5C7
2B2
INPUT 0 1 2 3 4 5 6 7
OUTPUT 0 1 2 3 4 5 6 7
10 CONTROL ST.
74126 2C
UART 2G
TBMT RD1 RD2 RD3 RD4 RD5 RD6 RD7 RD8
PE FE OR DA
EOC
DS
DB1 DB2 DB3 DB4 DB5 DB6 DB7 DB8
SI RDE SWE RCP TCP XR RDA
SO NSB NB2 NB1 EPS PI CS
1K +5V
7400 5C
7474 3C
7474 4C
CLK D Q
AUX. INP.
AUX. OUT
7404 5B
CLOCK ADJ.
220
4.7 µf
311
2.7K
10K
.02 µf
.001 µf
+5V -12V
565 2C
3µf
.001µf
.01µf
REC. FREQ.
20K 1K 4.7K 10K
5µf
FSK IN
566 4D
1µf
.02
FSK OUT
NPN G.P.
1.2K
.001 µf
SPACE ADJ.
MARK ADJ.
P3
2.7K
470Ω
20K
1K
MCT2 3D
120
2N2222
MOTOR CONTROL
RLY

IV-3-C. ASCII Keyboard Input.

The 5740 (1E) is a keyboard scan encoder with ASCII output. It provides all the scan pulses and looks for key closures. When it finds a key pressed it provides the output word and a strobe. The seven 7404 inverters (1C and 1D) invert the output word back to positive logic. The 7400 (1B) latches the strobe (data ready bit) and is reset by a control strobe from the computer. Half of the 7404 (1D) is used for the clock for the 5740. Its frequency has to be between 10 KHz and 200 KHz for proper operation.

For schematic and key codes, see next 2 pages.


SOCKET 1A
INPUT
0
1
2
3
4
5
6
7 (DATA READY)
CONTROL ST. (RESET)
+5V
GND
-12V
2
3
4
5
6
7
8
9
10
24
12
11
TANT. CAP.
TANT. CAP.
1E18
1B7
1C7
1D7
1E2
1E15
1E16
1B14
1C14
1D14
1E32
7400
1B
7400
1B
7400
1B
7404
1D
7404
1C
180Ω ¼W
.02µf
MM5740
1E
B1
B2
B3
B4
B5
B6
B7
DS
DSC
CLOCK
X
Y
KBM
470pf
SHIFT
CONTROL
2-150Ω ¼W
+5
X2
X3
X4
X5
X6
X7
X8
X9
Y7
Y6
Y5
Y4
Y3
Y2
Y1
Y10
SPACE

RGS 008A-8

KEY CODES FOR 008A-K ASCII KEYBOARD INPUT KIT

UNSHIFT		SHIFT		CONTROL		SHIFT & CONTROL	
1	061	!	041	1	061	!	041
2	062	"	042	2	062	"	042
3	063	#	043	3	063	#	043
4	064	$	044	4	064	$	044
5	065	%	045	5	065	%	045
6	066	&	046	6	066	&	046
7	067	'	047	7	067	'	047
8	070	(	050	8	070	(	050
9	071	)	051	9	071	)	051
Ø	060	Ø	060	Ø	060	Ø	060
:	072	*	052	:	072	*	052
Q	121	Q	121	DC1	021	DC1	021
W	127	W	127	ETB	027	ETB	027
E	105	E	105	ENQ	005	ENQ	005
R	122	R	122	DC2	022	DC2	022
T	124	T	124	DC4	024	DC4	024
Y	131	Y	131	EM	031	EM	031
U	125	U	125	NAK	025	NAK	025
I	111	I	111	HT	011	HT	011
O	117	_	137	SI	017	US	037
P	120	@	100	DLE	020	NUL	000
-	055	=	075	-	055	=	075
A	101	A	101	SOH	001	SOH	001
S	123	S	123	DC3	023	DC3	023
D	104	D	104	EOT	004	EOT	004
F	106	F	106	ACK	006	ACK	006
G	107	G	107	BEL	007	BEL	007
H	110	H	110	BS	010	BS	010
J	112	J	112	LF	012	LF	012
K	113	[	133	VT	013	ESC	033
L	114	\	134	FF	014	FS	034
;	073	+	053	;	073	+	053
Z	132	Z	132	SUB	032	SUB	032
X	130	X	130	CAN	030	CAN	030
C	103	C	103	ETX	003	ETX	003
V	126	V	126	SYN	026	SYN	026
B	102	B	102	STX	002	STX	002
N	116	↑	136	SO	016	RS	036
M	115	]	135	CR	015	GS	035
,	054	<	074	,	054	<	074
.	056	>	076	.	056	>	076
/	057	?	077	/	057	?	077
SP	040	SP	040	SP	040	SP	040

V. SOFTWARE AND APPLICATIONS

This section introduces the software and applications exchange service. All schematics, block diagrams and ideas should be submitted in as much detail as possible. We will try them out, consolidate as necessary, then print them and send them out to all manual holders. Please use the format below for any programs and/or subroutines you send in to the exchange service. We will send out a format for cassette tape when we are ready to handle it.

SOFTWARE FORMAT

```
ADDRESS          INSTRUCTION                           COMMENTS
 (octal)         (octal code)  (mnemonic)
0 0, 0 0 0        000 0           H L T                A halt instruction
1 2  3 4 5   6   7 8 9      10   11 12 13       14     15 . . . . . . . . . . 32
```

If every manual holder sends in one program or one application during the next year, we will all have a fantastic and valuable volume of reference for future use. We ourselves will be working on software and applications for everyone's use. All software and applications submitted become public domain and are not patentable, nor should any patents be infringed upon without written permission. Address all software and applications contributions, and any further questions about anything in the kit, to:

RGS Electronics
Software and Applications
3650 Charles St. Suite K
Santa Clara, California 95050

008A-C CASSETTE TAPE BOOTSTRAP

I. Description and use.

This program loads the first record off the tape recorder. There are no error detection steps in the bootstrap program, so if the first program doesn't work the bootstrap has to be run again. The program first selects the tape recorder (address 300_8) and starts the motor. Next, it waits for the carrier. When the carrier is detected it starts a .5 second timer. When the timer runs out, it sets the start address, then looks for the first word. The words, coming in, are loaded into memory until the last word is detected. The motor then stops and control is returned to the calling program.

II. Special instructions.

This bootstrap is set up for loading the tape read/write off tape. For other programs, 3 words have to be changed. The contents of addresses 00,145 and 00,147 are, respectively, the high and low starting address of the program to be read. The contents of address 00,170 is the total number of words of the program to be read.

III. Operating instructions.

Manually load the bootstrap from the front panel into the addresses specified. Call the bootstrap from one of the restart areas.

IV. Program.

```
00 100  006  LAI  00 121  000          00 142  130          00 163  101  INP
   101  300          122  041  DCE        143  000             164  370  LMA
   102  161  SEL     123  110  JFZ        144  056  LHI        165  060  INL
   103  006  LAI     124  116             145  001             166  306  LAL
   104  230          125  000             146  066  LLI        167  074  CPI
   105  141  CON     126  016  LBI        147  000             170  313
   106  101  INP     127  006             150  006  LAI        171  110  JFZ
   107  044  NDI     130  006  LAI        151  310             172  150
   110  010          131  310             152  141  CON        173  000
   111  150  JTZ     132  141  CON        153  101  INP        174  006  LAI
   112  106          133  101  INP        154  002  RLC        175  000
   113  000          134  002  RLC        155  100  JFC        176  141  CON
   114  046  LEI     135  100  JFC        156  153              177  007  RET
   115  040          136  133             157  000
   116  031  DCD     137  000             160  006  LAI
   117  110  JFZ     140  011  DCB        161  010
   120  116          141  110  JFZ        162  141  CON
```

008A-C CASSETTE TAPE READ/WRITE

I. Description and use.

This program is used to read from and write on a cassette tape. An error on read is indicated by a halt at location 01,134. A six word header is used for identification and location of the program. The format is:

xxx - lower record number
xxx - upper " "
xxx - lower starting address
xxx - upper " "
xxx - lower number of words (not counting header)
xxx - upper " " " " " "

In the write mode, the program writes 2 seconds of mark, the header, the main text, then .1 seconds of mark. It then stops the recorder. In the read mode, it starts the motor, then looks for the mark frequency. When it finds mark, it times out, then checks again for mark. If mark is there, the program looks for the header. The header tells it where to put the main text. When finished, it stops the recorder.

II. Special instructions.

The first program on tape should be started right at the beginning of the magnetic tape. When writing, locations 00,100 to 00,105 have to be loaded with the header for that program. These locations are also used during read. Cassette address is 300.

III. Operating instructions.

To read the next program from tape: call location 01,000. To write a program on tape, put recorder in "record" and call location 01,166.

IV. Header for this program.

000 000 000 001 313 000.

V. <u>Program.</u>

```
01  000  106  CAL  CALL SRS (READ) 01  070  000
    001  047                           071  066  LLI
    002  001                           072  100
    003  006  LAI  MOTOR ON            073  036  LDI
    004  210                           074  006
    005  141  CON                      075  046  LEI
    006  101  INP  CAR. DET.           076  000
    007  044  NDI                      077  007  RET  END
    010  010                       01  100  347  LEM  (LDR)
    011  150  JTZ                      101  061  DCL
    012  006                           102  337  LDM
    013  001                           103  061  DCL
    014  046  LEI  .5 S. TIMER         104  327  LCM
    015  040                           105  061  DCL
    016  106  CAL  CALL TIM            106  317  LBM
    017  056                           107  361  LLB
    020  001                           110  352  LHC
    021  101  INP  CAR. DET.           111  007  RET  END
    022  044  NDI                      112  040  INE  E + 1 (RED)
    023  010                           113  006  LAI  RESET DA
    024  150  JTZ                      114  310
    025  006                           115  141  CON
    026  001                           116  101  INP  IS DA?
    027  106  CAL  CALL LHR            117  002  RLC
    030  067                           120  100  JFC
    031  001                           121  116
    032  106  CAL  CALL RED            122  001
    033  112                           123  044  NDI  ERROR?
    034  001                           124  340
    035  106  CAL  CALL LDR            125  150  JTZ
    036  100                           126  135
    037  001                           127  001
    040  106  CAL  CALL RED            130  310  LBA
    041  112                           131  006  LAI
    042  001                           132  000
    043  006  LAI  MOTOR OFF           133  141  CON
    044  000                           134  377  HLT
    045  141  CON                      135  006  LAI  GET DATA
    046  007  RET  END                 136  010
    047  006  LAI  SEL. TAPE (SRS)     137  141  CON
    050  300                           140  101  INP
    051  161  SEL                      141  370  LMA
    052  006  LAI  STATUS & RESET      142  006  LAI  RETURN TO STATUS
    053  220                           143  210
    054  141  CON                      144  141  CON
    055  007  RET  END                 145  031  DCD  DECR. COUNTER
    056  031  DCD  TIMER (TIM)         146  110  JFZ
    057  110  JFZ                      147  156
    060  056                           150  001
    061  001                           151  041  DCE
    062  041  DCE                      152  110  JFZ
    063  110  JFZ                      153  156
    064  056                           154  001
    065  001                           155  007  RET
    066  007  RET  END                 156  060  INL  INCR. ADDR.
    067  056  LHI  (LHR)               157  110  JFZ
```

V. Program. cont'd

```
01  160  113
    161  001
    162  050  JNH
    163  104  JMP
    164  113
    165  001       END
    166  106  CAL  CALL SRS (WRITE)
    167  047
    170  001
    171  006  LAI  REC. REQ. LITE
    172  201
    173  141  CON
    174  101  INP  REC. VERIFY
    175  044  NDI
    176  004
    177  150  JTZ
01  200  174
    201  001
    202  006  LAI  MOTOR ON
    203  213
    204  141  CON
    205  046  LEI  2 S. TIMER
    206  200
    207  106  CAL  CALL TIM
    210  056
    211  001
    212  106  CAL  CALL LHR
    213  067
    214  001
    215  106  CAL  CALL WTE
    216  237
    217  001
    220  106  CAL  CALL LDR
    221  100
    222  001
    223  106  CAL  CALL WTE
    224  237
    225  001
    226  046  LEI  .1 S. TIMER
    227  010
    230  106  CAL  CALL TIM
    231  056
    232  001
    233  006  LAI  STOP MOTOR
    234  000
    235  141  CON
    236  007  RET  END
    237  040  INE  E + 1 (WTE)
    240  101  INP  TBMT?
    241  012  RRC
    242  100  JFC
    243  240
    244  001
    245  307  LAM  LOAD A
    246  121  OUT  OUTPUT
    247  031  DCD  DECR. COUNTER
```

```
01  250  110  JFZ
    251  262
    252  001
    253  041  DCE
    254  110  JFZ
    255  262
    256  001
    257  104  JMP
    260  272
    261  001
    262  060  INL  INCR. ADDR.
    263  110  JFZ
    264  240
    265  001
    266  050  INH
    267  104  JMP
    270  240
    271  001
    272  101  INP  CHAR. DONE?
    273  044  NDI
    274  003
    275  074  CPI
    276  003
    277  110  JFZ
01  300  272
    301  001
    302  101  INP
    303  044  NDI
    304  003
    305  074  CPI
    306  003
    307  110  JFZ
    310  272
    311  001
    312  007  RET  END
```

008A-K KEYBOARD BOOTSTRAP PROGRAM

I. Description and use.

This program is used when it is desirable to load the first program in by means of the keyboard. Up to 256 words can be loaded in without change to the program. The program halts at the end of the page.

II. Special instructions.

Address 00,001 is the page address, and 00,003 is the start address in that page, of the target area to be loaded. These words are loaded when the program is keyed in. In the case of a mistake, an interrupt 000 will stop the program.

III. Operating instructions.

Key the program in from the front panel, then do an interrupt RSØ (005) to start it. Key in three octal digits, from the keyboard, for each word in sequence. Keyboard I/O address is 100_8.

IV. Program.

```
ADDRESS   OCTAL   MNEMONIC        ADDRESS   OCTAL   MNEMONIC
00 000     056      LHI           00 034     110     JFZ
   001     000                       035     010
   002     066      LLI              036     000
   003     000                       037     377     HLT
   004     006      LAI              040     101     INP
   005     100                       041     064     ORI
   006     161      SEL              042     000
   007     141      CON              043     120     JSF
   010     045      RS4              044     040
   011     044      NDI              045     000
   012     003                       046     141     CON
   013     012      RRC              047     007     RET
   014     012      RRC
   015     310      LBA
   016     045      RS4
   017     044      NDI
   020     007
   021     002      RLC
   022     002      RLC
   023     002      RLC
   024     261      ORB
   025     310      LBA
   026     045      RS4
   027     044      NDI
   030     007
   031     261      ORB
   032     370      LMA
   033     060      INL
```

KEYBOARD OCTAL LOADER - 008A-K

I. Description and use.

This program allows words to be written into memory in octal format. Programs can also be called from the keyboard.

II. Special instructions.

Pressing the "A" key will allow address entry at any time.

III. Operating instructions.

To start program: jump to location 01,313. Press the "A" key and enter full 5 octal digit address. Every time 3 octal digits, to be loaded, are entered, the address is incremented. If an "E" is pressed, in place of the first octal digit of a word, then the program will call the program starting at the latest address. A new address has to be entered when control returns to the keyboard routine.

IV. Header for this program.

001 000 313 001 214 000.

V. Program.

```
01   313  006  LAI  KEYBOARD SELECT
     314  100
     315  161  SEL
     316  141  CON
     317  016  LBI
     320  101
     321  026  LCI
     322  007
   * 323  106  CAL  CALL INP
     324  116
     325  002
     326  271  CPB  IS A?
     327  150  JTZ  YES JUMP TO A
     330  003
     331  002
     332  074  CPI  IS E?
     333  105
     334  150  JTZ  YES JUMP TO E
     335  077
     336  002
     337  044  NDI  PREP. DIGIT
     340  003
     341  012  RRC
     342  012  RRC
     343  330  LDA
     344  106  CAL  CALL INP
     345  116

01   346  002
     347  271  CPB  IS A?
     350  150  JTZ  YES JUMP TO A
     351  003
     352  002
     353  242  NDC  PREP. DIGIT
     354  002  RLC
     355  002  RLC
     356  002  RLC
     357  263  ORD
     360  330  LDA
     361  106  CAL  CALL INP
     362  116
     363  002
     364  271  CPB  IS A?
     365  150  JTZ  YES JUMP TO A
     366  003
     367  002
     370  242  NDC  PREP. DIGIT
     371  263  ORD
     372  370  LMA  LOAD MEM
     373  060  INL  INCR. H & L
     374  110  JFZ  *
     375  323
     376  001
     377  050  INH
```

V.   Program.

```
02  000  104  JMP  *
    001  323
    002  001
 A  003  106  CAL  CALL INP
    004  116
    005  002
    006  271  CPB  IS A?
    007  150  JTZ  YES JUMP TO A
    010  003
    011  002
    012  242  NDC  PREP. DIGIT
    013  002  RLC
    014  002  RLC
    015  002  RLC
    016  350  LHA
    017  106  CAL  CALL INP
    020  116
    021  002
    022  271  CPB  IS A?
    023  150  JTZ  YES JUMP TO A
    024  003
    025  002
    026  242  NDC  PREP. DIGIT
    027  265  ORH
    030  350  LHA
    031  106  CAL  CALL INP
    032  116
    033  002
    034  271  CPB  IS A?
    035  150  JTZ  YES JUMP TO A
    036  003
    037  002
    040  044  NDI  PREP. DIGIT
    041  003
    042  012  RRC
    043  012  RRC
    044  360  LLA
    045  106  CAL  CALL INP
    046  116
    047  002
    050  271  CPB  IS A?
    051  150  JTZ  YES JUMP TO A
    052  003
    053  002
    054  242  NDC  PREP. DIGIT
    055  002  RLC
    056  002  RLC
    057  002  RLC
    060  266  ORL
    061  360  LLA
    062  106  CAL  CALL INP
    063  116
    064  002
    065  271  CPB  IS A?
    066  150  JTZ  YES JUMP TO A
    067  003

02  070  002
    071  242  NDC  PREP. DIGIT
    072  266  ORL
    073  360  LLA
    074  104  JMP
    075  323
    076  001
 E  077  335  LDH  LOAD CALL ADD.
    100  346  LEL
    101  056  LHI
    102  002
    103  066  LLI
    104  111
    105  374  LME
    106  060  INL
    107  373  LMD
    110  106  CAL  CALL PROGRAM
    111  000
    112  000
    113  104  JMP
    114  313
    115  001
    116  101  INP  KEYBOARD INPUT
    117  064  ORI
    120  000
    121  120  JSF
    122  116
    123  002
    124  141  CON
    125  101  INP
    126  007  RET
```

APPLICATION NOTE #1

The I/O cable can be made with 2 16-pin IC component carriers and some wire. The 4 strobes, interrupt exp. and ready exp. should be twisted pair to eliminate crosstalk. The twisted pair ground wires can be connected to the ground pin at each end by means of a piece of bus wire. This arrangement has been tested and is being used here at RGS. It also has the advantage of being very inexpensive to implement.

RGS.

CHANGES AND ADDITIONS TO 008A MICROCOMPUTER MANUAL

Page 2, top of page:

Delete the 4 sentences, "When using Molex....Molex pins first." Add, in place of this, "If an IC seems to be bad, check its seating in the socket."

Page 2, Par. I-2-A:

#2 of list, CHANGE "Molex pins" TO "IC sockets".

Page 3, Par. I-2-B:

CHANGE "2 9-pin Molex strips (for 8008)" TO "14 14 pin IC socket, solder type";
CHANGE "6 8-pin Molex strips" TO "3 16 pin IC socket, solder type";
CHANGE "28 7-pin Molex strips" TO "1 18 pin IC socket, solder type (for 8008)".

Page 4, Par. I-2-C:

CHANGE "14 8-pin Molex strips" TO "4 14 pin IC socket, solder type";
CHANGE "8 7-pin Molex strips" TO "7 16 pin IC socket, solder type".

Page 5:

Right side of schematic, IC #5A, reverse pins 1 and 2.

Page 8, Par. I-3-A:

#1 of list, change "5BØ" to "5B1";
#3 of list, change "Molex pins" to "IC sockets".

Page 8, Par. I-3-B:

CHANGE "20 8-pin Molex strips" TO "10 16 pin IC sockets, solder type".

Page 9:

Left side of schematic, show IC #5D, pins 13 and 14 being connected to the ground pins with jumpers;
Left side of schematic, label IC #5B, pin 12 "bank enable".

Page 11, Par. I-4-A:

#2 of list, change "Molex pins" to "IC sockets";
#5 of list, add "(break screw mount tabs off first by bending back and forth. If there is any trouble with multiple counts when using the step or int. switch, cut the top and bottom run connecting the 2 poles of the switch apart. Do not cut the runs going to the ICs or the run going down the middle wiper connections of the switches.)"
DELETE Step 6;
CHANGE "Step 7" TO "Step 6";
CHANGE "Step 8" TO "Step 7".

CHANGES AND ADDITIONS (cont'd)

Page 12, Par. I-4-A:

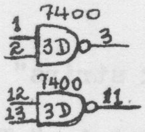
CHANGE "Step 9" TO "Step 8".

Page 12, Par. I-4-B:

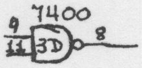
CHANGE "9 DPDT (T/F) Rocker switches" TO "9 SPDT Rocker switches";
CHANGE "20 7-pin Molex strips" TO "10 14-pin IC sockets, solder type";
CHANGE "4 .687" spacer" TO "4 .687" threaded spacer";
CHANGE "4 #4-40x1½" screw" TO "4 #4-40x7/8" screw";
CHANGE "4 #4 lockwasher" TO "4 #4-40x3/8" screw";
CHANGE "4 #4-40 nut" TO "4 #4 lockwasher (for use between threaded spacer and front panel p.c.b.);
Add "1 16-pin IC socket, wire-wrap type".

Page 13:

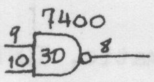
Schematic, lower right corner, IC #3D, add missing pin numbers:



Schematic, middle left, IC #3D, CHANGE:



TO:



Page 14:

Layout, lower right, delete "jumper" and dotted line.

Page 15, Par. I-5-E:

CHANGE "1 2N2907" TO "1 P3, TO-92 case".

Page 16:

Schematic, middle, change "2N2907" to "P3".

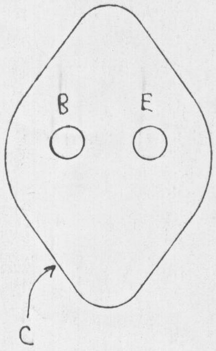
Page 17:

Layout, middle, change "2N2907" to "P3" and show flat spot on bottom of circle for that transistor.

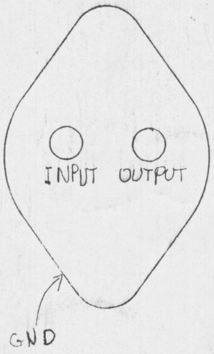
Page 26, Par. III-2-7:

Line 10, CHANGE "and the CC is 0,2" TO "and the CC is 0,3";
Line 13, CHANGE "and the CC is 0,3" TO "and the CC is 0,2".

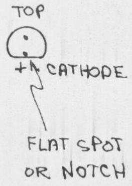
2N3055
BOTTOM VIEW



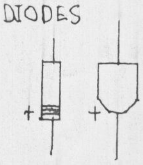
340K-12
BOTTOM VIEW



LED's

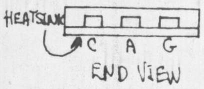


DIODES



3A. DIODES
HAVE THE THICKER
LEADS.

SCR



END VIEW

ADDENDUM:

PAGE 3 CHANGE 2 #440 x 1¼" SCREWS TO
1 #4-40 x 1¼" SCREWS

TANTALUM CAPACITOR VALUE IS 15μf

CS13B
4.7μf TANTALUM CAPACITOR IS PART NO. T475K