Soldering PC Boards

Two common causes of trouble with PC boards are bad solder joints or solder bridges. Usually, bad solder joints are caused by either a cold solder joint or contamination. A good solder joint is characterized by a bright shiny and smooth surface (see figure 1).

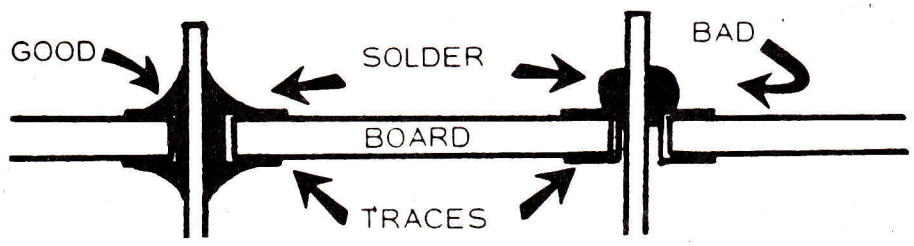


Figure 1. CROSS-SECTION OF A PC BOARD SHOWING GOOD AND BAD SOLDER CONNECTIONS

A cold solder joint is characterized by a dull surface and usually a lumpy or balled appearance. It takes practice and patience to obtain a good solder joint consistently. However, the first step is to apply flux to all connections before the solder. Second, heat the connection for a second or two with the soldering iron. Third, apply solder to the opposite side of the connection. Don't touch the solder to the iron. Flux has a "wetting" effect on solder which causes the solder to flow smoothly, completely filling the connection. If flux is not used or the metal around the connection is contaminated (dirty) it is almost impossible to hav a good solder joint.

Solder bridges are usually caused by using a soldering iron tip that's too large, solder wire that's too large, or trying to rush the job. Use a small spade tip iron (see figure 2). Touch the connection with the flat side of the tip. After the flux bubbles touch the solder to the opposite side of the connection. Again, don't touch the solder to the iron. The connection is hot enough to melt the solder causing it to flow around the connection. Do not use too much solder. Use a little and watch it flow. Solder is like spice for cooking, don't use too much.

Applying heat for extended periods will cause either or both of the following: the trace or pad will lift from the board or the board material will turn brown. Remove the iron before this happens. One hobbyist counts the bubbles that pop in the solder. He found seven to nine bubbles insured good solder flow without over heating.

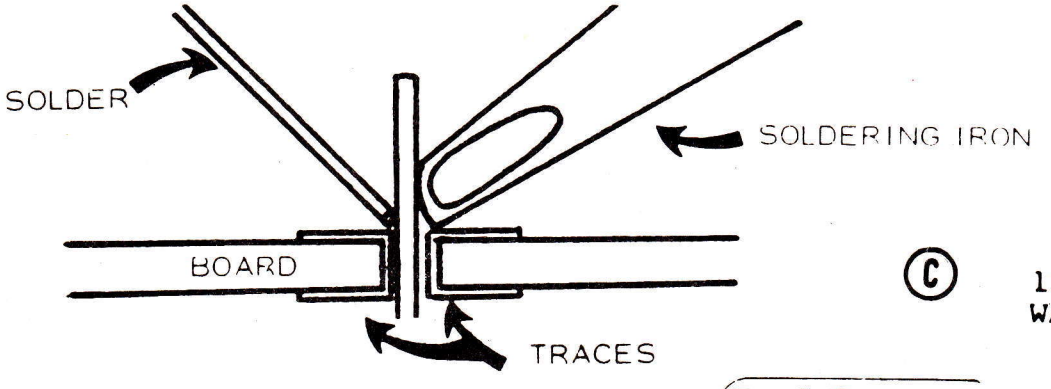


Figure 2.

The MEM-2 is a memory board designed to interface 2114's to the S-100 (WAMECO$_{TM}$) bus (see Tables I and II). Provisions have been made for multiple wait states, memory addressing options, Phantom Disable, and Bank Addressing. Any multiple of two memory chips can be used in the board start and stop address can be effectively set in 4 K boundries anywhere in the 65 K Byte memory of your computer. If 4 K Bytes or less, the board can be configured to occupy only the amount of 4 K Bytes in the memory map of your computer. This selection can be increased by 4 K Byte increments until the full 16 K Bytes is selected.

PARTS LIST

Schematic Identifier	Quantity	Part
U1-U5	5	7805
U6-U21, U37-U44, U47-U54	32	2114
U22, U23	2	DM8098, 8T98, or 74368
U24, U27	2	74LS138
U25, U26	2	7485
U28	1	74LS02
U29	1	7404
U30, U31	2	74LS74
U32	1	74122
U33, U35	2	74LS20
U34	1	74LS32
U36, U45, U46	3	DM8097, 8T97, 74367
C1, C4, C14, C19, C25, C36, C37, C46, C50, C52	10	0.1μF disc ceramic capacitor
C2, C3, C13, C24, C47, C49	6	22μF 16V(or higher) Tantalum Capacitor
C5-C12, C15-C18, C20-C23, C26, C29-C32, C34, C35, C38-C45, C48, C51	33	0.01μF disc ceramic capacitor
C28	1	39pF disc ceramic capacitor
C33	1	680pF disc ceramic capacitor
R1	1	330Ω 1/4 W carbon film resistor
R2	1	100Ω 1/4W carbon film resistor
R3-R20	18	2.7KΩ 1/4W carbon film resistor
S1, S2	2	8 position dip switch
	5	#361 AHAM (or equivalent) heat sinks
	8	14 pin low profile sockets
	9	16 pin low profile sockets
	32	18 pin low profile sockets

TOOLS OR SUPPLIES NEEDED TO ASSEMBLE AND TEST MEM-2

1 Q Tip cotton swab
1 pair needle nose pliers
1 pair diagonal cutting pliers
1 bottle rosin flux
1 tube silicone thermal compound
1 jar solder cleaner
1 roll solder wick
1 Phillips screwdriver
1 small adjustable wrench or socket to fit regulator nut
1 roll (.031" or .040") SN60/40 rosin core solder
1 25 to 40 W soldering iron with small spade tip
1 Strong light
1 magnifying glass
1 XACTO knife with number 16 blade
1 multimeter with leads
1 power supply with variable outputs

PIN	MNEMONIC	TERM.
1	+5V	
2	+15V	
3	XRDY	X
4	VI0	X
5	VI1	X
6	VI2	X
7	VI3	X
8	VI4	X
9	VI5	X
10	VI6	X
11	VI7	X
12		
13		
14		
15		
16		
17		
18	STAT DISABLE	X
19	CIC DISABLE	X
20	UNPROTECT	X
21	SS	X
22	ADDR DSBL	X
23	DO DSBL	X
24	Ø2	X
25	Ø1	X
26	PHLDA	X
27	PWAIT	
28	PINTE	
29	A5	
30	A4	
31	A3	
32	A15	
33	A12	
34	A9	
35	DO1	X
36	DOØ	X
37	A10	
38	DO4	X
39	DO5	X
40	DO6	X
41	DI2	X
42	DI3	X
43	DI7	X
44	SMI	
45	SOUT	
46	SINP	
47	SMEMR	
48	SHLTA	
49	CLOCK (2MHz)	
50	GND	

PIN	MNEMONIC	ALTER. PIN DESIG.	TERM.
51	+5V	A	
52	-15V	B	
53	SSW DSB	C	
54	EXT CLR	D	X
55		E	
56		F	
57		H	
58		J	
59		K	
60		L	
61		M	
62		N	
63		P	
64		R	
65		S	
66		T	
67	PHANTOM	U	
68	MWRITE	V	X
69	PS	W	
70	PROTECT	X	X
71	RUN	Y	X
72	PRDY	Z	X
73	PINT	a	X
74	PHOLD	b	X
75	PRESET	c	X
76	PSYNC	d	X
77	PWR	e	X
78	PDBIN	f	X
79	AØ	h	
80	A1	j	
81	A2	k	
82	A6	l	
83	A7	m	
84	A8	n	
85	A13	p	
86	A14	r	
87	A11	s	
88	DO2	t	X
89	DO3	u	X
90	DO7	v	X
91	DI4	w	X
92	DI5	x	X
93	DI6	y	X
94	DI1	z	X
95	DIØ	AA	X
96	SINTA	AB	
97	SWO	AC	
98	SSTACK	AD	
99	POC	AE	
100	GND	AF	

Table I

S-100 (WAMECO) BUS DESCRIPTION

Pin #	Mnemonic	Enabled State	Description
1	+8 Volts	NA	Unregulated +8 Volts DC. This voltage should not be less than +8 or greater than +11 volts.
2	+16 Volts	NA	Unregulated +16 Volts DC. This voltage should not be less than +16 or greater than +20 Volts.
3	XRDY	Low	Causes CPU to enter WAIT state when enabled.
4	$\overline{\text{VI0}}$	Low	Vectored Interrupt priority 0
5	$\overline{\text{VI1}}$	Low	Vectored Interrupt priority 1
6	$\overline{\text{VI2}}$	Low	Vectored Interrupt priority 2
7	$\overline{\text{VI3}}$	Low	Vectored Interrupt priority 3
8	$\overline{\text{VI4}}$	Low	Vectored Interrupt priority 4
9	$\overline{\text{VI5}}$	Low	Vectored Interrupt priority 5
10	$\overline{\text{VI6}}$	Low	Vectored Interrupt priority 6
11	VI7	Low	Vectored Interrupt priority 7
12	---	NA	Not used
13	---	NA	Not used
14	---	NA	Not used
15	---	NA	Not used
16	---	NA	Not used
17	---	NA	Not used
18	STAT DISABLE	Low	The eight status line buffers on the CPU board enter the high impedance state when enabled.
19	C/C DISABLE	Low	The six command/control line buffers on the CPU board enter the high impedance state when enabled.
20	UNPROTECT	High	Combined with address in an AND gate on a memory board which causes the PROTECT flip-flop to be cleared.
21	SS	High	Indicates the CPU is single stepping.
22	$\overline{\text{ADDR DSBL}}$	Low	The 16 address line buffers on the CPU board enter the high impedance state when enabled.
23	$\overline{\text{DO DSBL}}$	Low	The eight data-out lines on the CPU board enter the high impedance state when enabled.
24	Ø 2	High	Buffered TTL CPU phase 2 clock.
25	Ø 1	High	Buffered TTL CPU phase 1 clock.
26	PHLDA	High	CPU board "Hold Acknowledge" to HOLD-H input.
27	PWAIT	High	CPU output showing a WAIT state is occuring.

Table II

Pin #	Mnemonic	Enabled State	Description
28	PINTE	High	CPU output showing that Interrupts are enabled.
29	A5	High	Address Bit 5
30	A4	High	Address Bit 4
31	A3	High	Address Bit 3
32	A15	High	Address Bit 15
33	A12	High	Address Bit 12
34	A9	High	Address Bit 9
35	DO1	High	CPU Data Out Bit 1
36	DO0	High	CPU Data Out Bit 0
37	A10	High	Address Bit 10
38	DO4	High	CPU Data Out Bit 4
39	DO5	High	CPU Data Out Bit 5
40	DO6	High	CPU Data Out Bit 6
41	D12	High	Data In Bit 2 to CPU
42	D13	High	Data In Bit 3 to CPU
43	D17	High	Data In Bit 7 to CPU
44	SM1	High	CPU output indicating it is performing Fetch Instruction.
45	SOUT	High	CPU output showing it is in an output cycle.
46	SINP	High	CPU output showing it is in an input cycle.
47	SMEMR	High	CPU status signal indicating the current cycle is a Memory Read cycle.
48	SHLTA	High	CPU status signal indicating the CPU is halted.
49	CLOCK(2MHz)	Low	A buffered 2 MHz clock for general use.
50	GND	NA	Ground (common)
51	+8 Volts	NA	(Same as pin 1)
52	-16 Volts	NA	Unregulated-16 Volts DC. This voltage should not be greater than -16 or less than -20 Volts.
53	$\overline{\text{SSW DSB}}$	Low	Sense Switch Disable disables CPU board data input buffers so that CPU can read sense switches.
54	$\overline{\text{EXT CLR}}$	Low	Front panel generated I/O clear signal.
55	---	NA	Not used
56	---	NA	Not used
57	---	NA	Not used
58	---	NA	Not used
59	---	NA	Not used
60	---	NA	Not used
61	---	NA	Not used
62	---	NA	Not used
63	---	NA	Not used
64	---	NA	Not used
65	---	NA	Not used
66	---	NA	Not used
67	PHANTOM	NA	Used for Memory Bank Selection (or for SOL© Systems)

Table II

Pin #	Mnemonic	Enabled State	Description
68	MWRITE	High	CPU output showing Data Out Bus data is to be written into the memory selected by the address lines.
69	$\overline{PS}$	Low	Shows Protect Status of selected memory.
70	PROTECT	High	Combined with address in an AND gate on a memory board which causes the PROTECT flip-flop to be set.
71	RUN	High	Front panel indication that CPU run instruction has been input.
72	PRDY	Low	Causes the CPU to enter the WAIT state when enabled.
73	$\overline{PINT}$	Low	If interrupts have been enabled causes the CPU to enter the Interrupt Acknowledge condition at the conclusion of the current instruction.
74	$\overline{PHOLD}$	Low	CPU input which causes a HOLD status to occur. DMA transfer request signal is $\overline{PHOLD}$.
75	$\overline{PRESET}$	Low	CPU board system reset signal.
76	PSYNC	High	CPU output showing the start of a new machine cycle. This signal is used on the CPU board to enable the loading of the System Status Latch.
77	$\overline{PWR}$	Low	Indication that data on the Data Out Bus is to be written either to a memory or an I/O device.
78	PDBIN	Low	Indication to the selected memory or I/O device that the CPU expects data on the Data In Bus.
79	A0	High	Address Bit 0
80	A1	High	Address Bit 1
81	A2	High	Address Bit 2
82	A6	High	Address Bit 6
83	A7	High	Address Bit 7
84	A8	High	Address Bit 8
85	A13	High	Address Bit 13
86	A14	High	Address Bit 14
87	A11	High	Address Bit 11
88	DO2	High	CPU Data Out Bit 2
89	DO3	High	CPU Data Out Bit 3
90	DO7	High	CPU Data Out Bit 7
91	DI4	High	Data In Bit 4 to CPU
92	DI5	High	Data In Bit 5 to CPU
93	DI6	High	Data In Bit 6 to CPU

Table II

S-100 (WAMECO) BUS DESCRIPTION (Cont.)

Pin #	Mnemonic	Enabled State	Description
94	DI1	High	Data In Bit 1 to CPU
95	DI0	High	Data In Bit 0 to CPU
96	SINTA	High	CPU Interrupt Acknowledge Signal
97	SWO	Low	CPU output indicating the current cycle involves writing to a memory or I/O device.
98	SSTACK	High	CPU output indicating the address bus contains the stack address and the current cycle will have a stack operation.
99	$\overline{\text{POC}}$	Low	Power On Clear reset signal
100	GND	NA	Ground (common)

Table II

I. Assembly of MEM-2

I-1. Before placing any parts on the board, check the board for any hairline shorts (slivers). All boards have been inspected at least three times before shipping. Still, a good hobbyist checks any board he buys.

I-2. Using a strong light and a magnifying glass, very carefully check all leads on the top of the board (this is the side marked COMPONENT SIDE). If any slivers are found, carefully cut and scrape them with an XACTO knife. The underside of the board will be checked after assembly.

I-3. Place all the 14, 16, and 18 pin sockets in their positions on the top side of the board.

I-4. After positioning all the sockets in place, check to ensure that a socket is not in the position S1 or S2. Dip switches will not stay in place in a socket. Place a book on top of the sockets, hold the book tight against the board and turn them over so that the underside of the board is up. Press down on the board and solder one pin on each end of each socket. This will ensure the sockets are flat against the board. When the tacking of all sockets is completed, finish soldering all the other pins of the sockets.

NOTE

DO NOT PUT IC'S IN SOCKETS AT THIS TIME. THEY WILL BE INSTALLED LATER.

I-5. Bend the leads on R3-R20 (2.7KΩ RED, VIOLET, RED) and place in board. Check parts placement drawing (figure 3) for correct locations. Bend the leads of the resistors on the underside of the board to retain them in place until they are soldered. Turn the board over and solder all the resistors. Clip the leads of the resistors flush with the underside of the board with the diagonal pliers.

I-6. Bend the leads on R1 (330Ω ORANGE, ORANGE, BROWN) and place in board. Check parts placement drawing (figure 3) for correct location. Bend the leads of the resistor on the underside of the board to retain it in place until it is soldered. Turn the board over and solder the resistor. Clip the leads of the resistor flush with the underside of the board with the diagonal pliers.

I-7. Bend the leads on R2 (100Ω BROWN, BLACK, BROWN) and place in board. Check parts placement drawing (figure 3) for correct location. Bend the leads on the resistor on the underside of the board to retain it in place until it is soldered. Turn the board over and solder the resistor. Clip the leads of the resistor flush with the underside of the board with the diagonal pliers.

CAUTION

CHECK DISC CAPACITORS FOR PROPER VALUE BEFORE INSERTING IN BOARD. ENSURE .01 μF AND .1 μF DISC CAPACITORS ARE NOT INTERCHANGED.

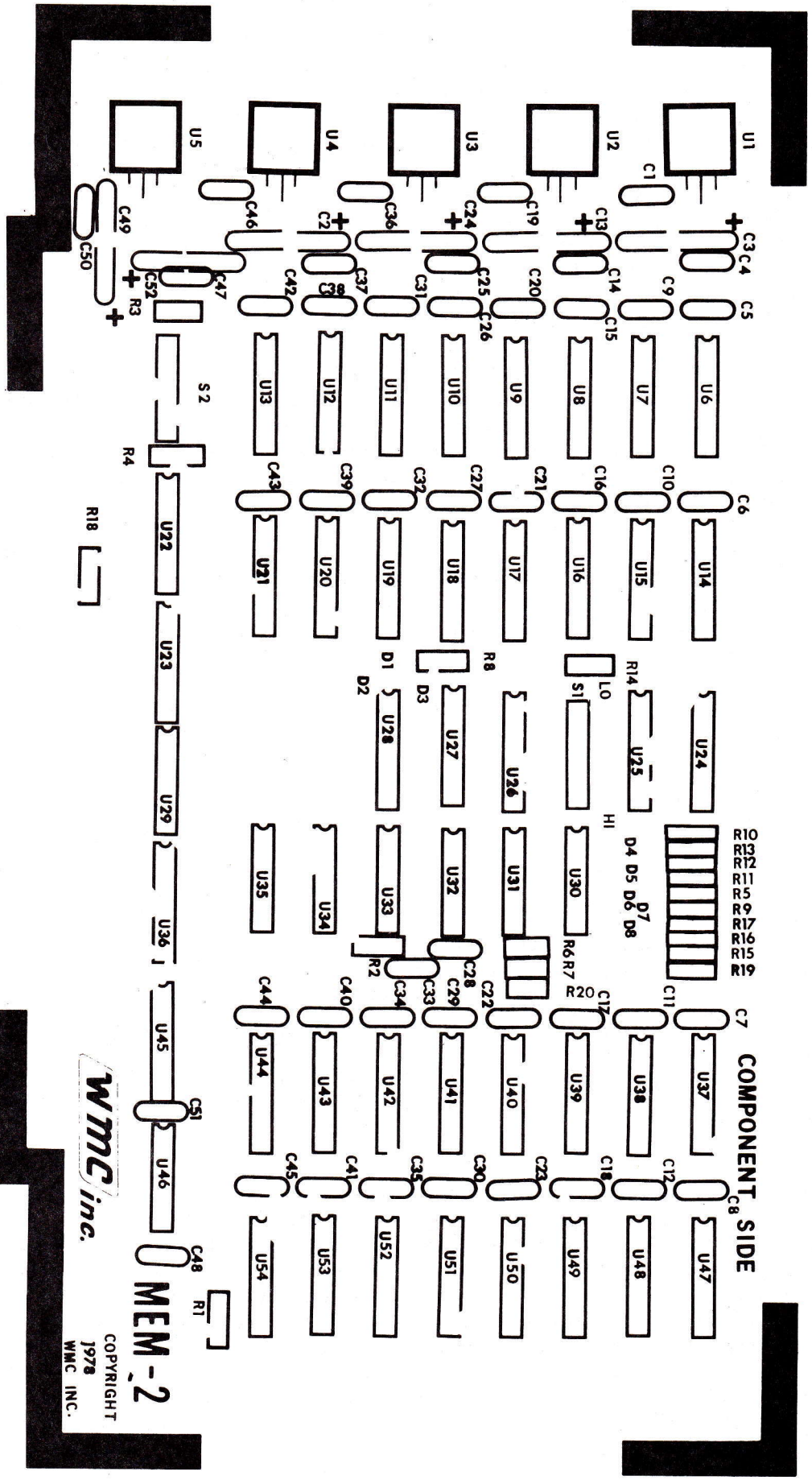


Figure 3. MEM-2 Parts Placement Diagram

I-8. Put the leads of C1, C4, C14, C19, C25, C36, C37, C46, C50, C52 (.1 F) disc capacitors in the board. Check parts placement drawing (figure 3) for proper locations. Bend the leads of the capacitors to retain them in place until they are soldered. Turn the board over and solder the capacitors. Clip the leads of the capacitors flush with the underside of the board with the diagonal pliers.

I-9. Put the leads of C5-C12, C15-C18, C20-C23, C26, C29-C32, C34, C35,C51 C38-C45, C48 (.01 μF) disc capacitors in the board. Check parts placement drawing (figure 3) for proper locations. Bend the leads of the capacitors to retain them in place until they are soldered. Turn the board over and solder the capacitors. Clip the leads of the capacitors flush with the underside of the board with the diagonal pliers.

I-10. Place C2, C3, C13, C24, C47, C49 (22 μF tantalum) in place. Ensure that the polarities are correct. Check parts placement drawing (figure 3) for correct placement and polarity. Bend the leads of the capacitors to retain them in place until they are soldered. Turn the board over and rest it on books. Solder the capacitors in place. Clip the leads flush with the underside of the board with diagonal pliers.

I-11. Put the eight position dip switches in place. Ensure that the switches are installed so the the OFF positions are towards the gold fingers of the board. Bend the two pins at each end of each switch to retain it in place until it is soldered. Turn the board over and rest it on books as before. Solder the eight position dip switches in place.

I-12. Before installing the regulators, it is recommended that they be tested for proper voltage regulation.

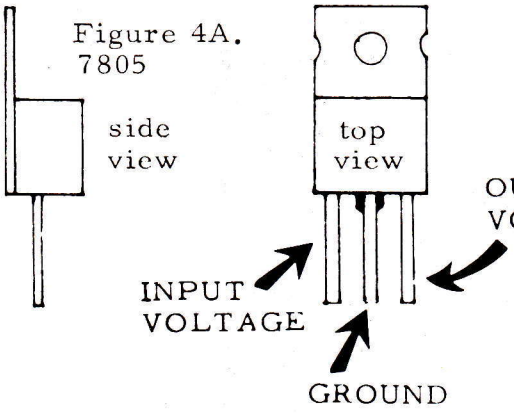


Figure 4A.
7805

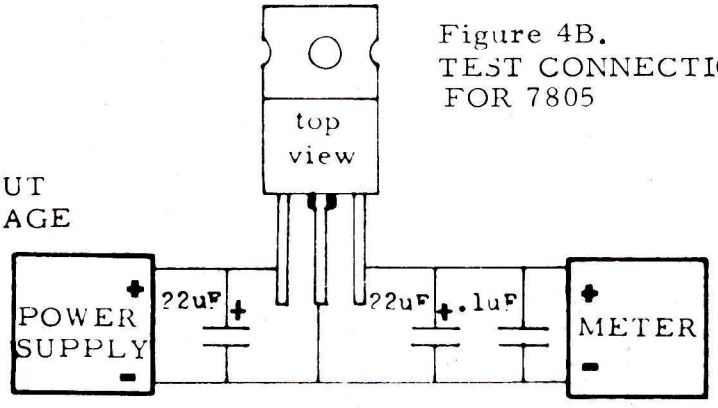


Figure 4B.
TEST CONNECTIONS
FOR 7805

To prevent oscillation of the regulators, assemble a test rig as shown. The capacitors must be installed observing the correct polarity. This test rig is for pre-installation testing only. The filter capacitors installed on the board serve the same purpose in the final assembly. Attach the power supply, multimeter, and capacitors to the 7805 as shown in figure 4B. Place the multimeter in a DC range that will allow 10 volts to be displayed. The regulator needs a 2.0 volt minimum difference between the input voltage and the regulated output voltage. If the power supply has a voltmeter, observe the input voltage during the test using that. If the power supply does not have a voltmeter, switch the + meter lead between the input lead and the output lead. The input and output voltages can thus be observed.

I-13. Slowly increase the input voltage and observe the input and output voltages. When the input voltage is between 7.0 and 7.5 volts, the regulated output of a properly operating 7805 should be between 4.8 and 5.2 volts. Replace the regulator if it does not meet these limits.

I-14. When the 7805's have been tested as outlined in I-12, place the 7805's on the board so that the mounting hole on the 7805 lines up with the corresponding hole on the MEM-2. Note where the leads on the 7805's pass over the connection holes on the MEM-2. Bend the leads on the 7805's so that the leads can be inserted into the proper holes. Mount the 7805's on the board using a #6 nut and 5/8" 6-30 screw. Insert a heatsink between the board and the 7805. Solder the leads of the 7805's in place.

I-15. Remove the nuts and screws from the 7805's. Bend the 7805's upward and remove the heatsinks. Place a moderate amount of silicone thermal heat grease on the underside of the 7805's and the underside of the heatsinks with a Q tip cotton swab. Coat all of the area mentioned with an even coating of the heat grease. Reinstall the heatsinks, nuts, and screw. Ensure the nuts are tight.

I-16. Clean off the flux on the underside of the board with flux cleaner.

II. Inspection and Testing

II-1. Use a bright light and magnifying glass to inspect all the traces on the underside of the board. If any slivers are found, cut and scrape them with an Xacto knife. Use the solder wick and soldering iron to remove any solder bridges found. Cover the solder bridge with flux and place a clean piece of solder wick on top of the bridge. Place the soldering on top of the solder wick and hold until solder is seen flowing up into the solder wick. Remove the iron and wick. Check to see if the bridge has been completely removed. If not, repeat the process until the bridge has been removed. Clean the flux off the board with flux cleaner.

NOTE

AT THIS TIME NO IC'S HAVE BEEN INSTALLED ON THE BOARD. DO NOT INSTALL IC'S ON THE BOARD UNTIL CALLED FOR IN THE CHECKOUT PROCEDURE.

II-2. Place all switches of the eight position dip switches in the OFF position. Place the multimeter in the R x 1 scale. Place one probe on the gold finger for pin 1. Place the other probe on all the other fingers sequentially to check for shorts. Repeat this procedure for each pin. There should be only two sets of pins that are shorted; 1 to 51 and 50 to 100. If any other pair of pins are shorted, use a strong light and magnifying glass to locate the solder bridge or sliver causing the short. When the short has been located, correct it as outlined in II-1. If there is no solder bridge or sliver, a component is shorted. Check the MEM-2 schematic (figure 5) to locate the probable component. Lift one lead of the suspected component and recheck between the two fingers that had a bad reading. If the bad reading is now correct, replace the component. If the reading is still bad, continue troubleshooting until the faulty component is located and replaced. Ensure that all components that had a lead lifted have the lead reconnected.

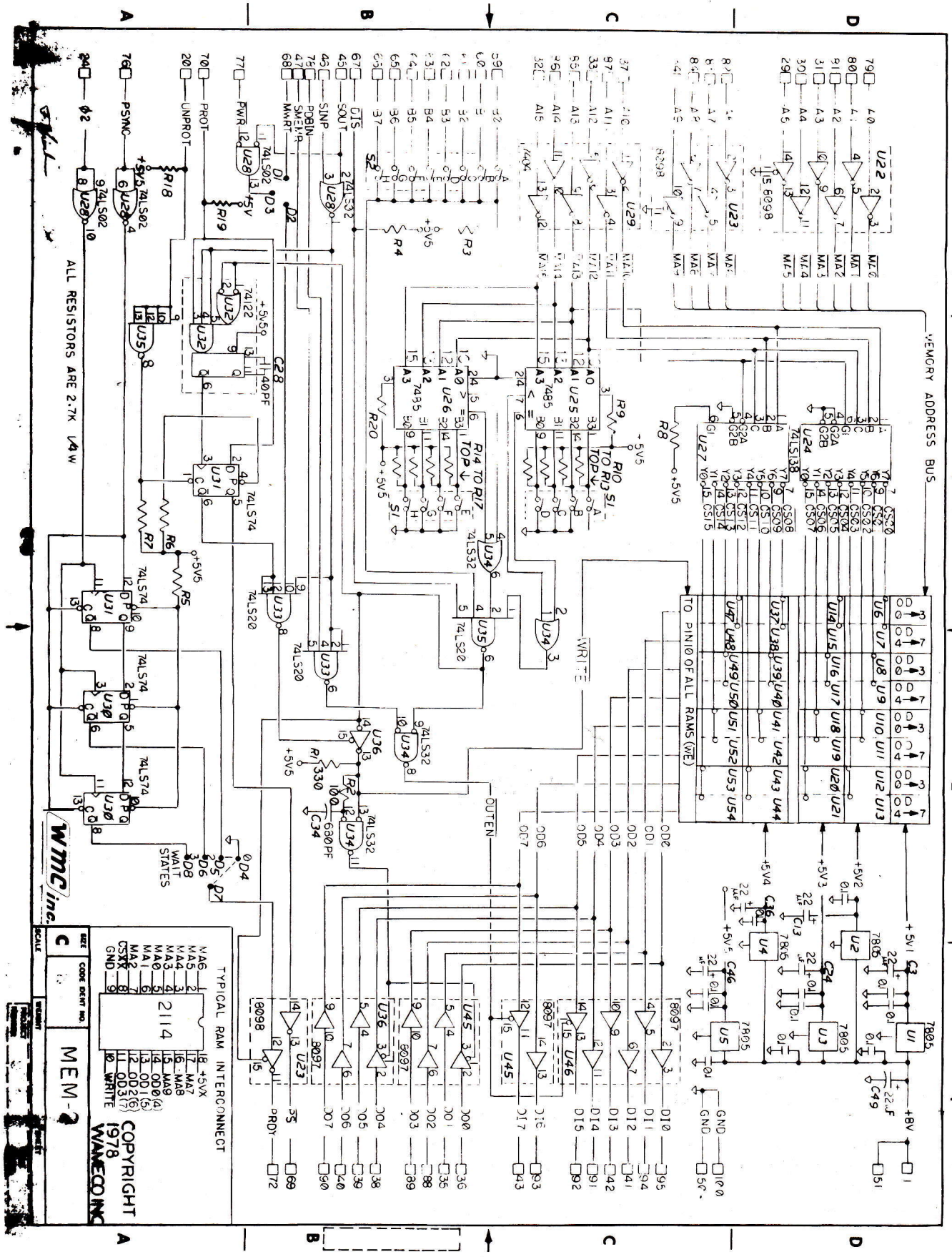


Figure 5. MEM-2 Schematic

WARNING

DO NOT INSTALL OR REMOVE ANY BOARD IN COMPUTER WITH POWER ON. DAMAGE TO BOARDS AND COMPUTER MAY RESULT.

II-3. Ensure computer is OFF. Plug MEM-2 into the motherboard. Check that the MEM-2 is correctly plugged in and that the board is fully seated in the connector. Turn the computer power ON and check the outputs of each regulator on the MEM-2. If the regulators do not have outputs as stated in I-13, turn the computer power OFF and replace the defective regulator. Repeat I-13 to check out the new regulator before installing. If the voltage on the regulator is not now correct, check the voltages on the motherboard. If the voltages on the motherboard are incorrect, repair the power supply as needed. If and when the voltage check good, turn the computer power OFF and remove the MEM-2 from the motherboard.

II-4. Select the proper wait state and MWRITE selection for the board by installing the jumpers on the MEM-2 as shown in Figure 6.

II-5. Clean off the flux on the underside of the board with flux cleaner.

II-6. Install all the IC's on the MEM-2. Check parts placement drawing (Figure 3) for proper location and correct polarity of IC's.

CAUTION

ENSURE ALL IC'S ARE INSTALLED CORRECTLY. INCORRECT POLARIZATION OF IC WILL RESULT IN DAMAGE TO IC AND CAUSE SUBSEQUENT TROUBLES TO APPEAR ON THE BOARD.

II-7. The memory of the MEM-2 is addressed in 4KByte segments. The minimum segment that can be selected is 4K. The board can be populated in 1KByte increments (2 memory chips) at a time. The memory is divided into four separate sections which will be referred to as A through D (see Figure 7). A will always have memory starting at 0000H, 4000H, 8000H, or C000H. B will always have memory starting at 1000H, 5000H, 9000H, or D000H. C will always have memory starting at 2000H, 6000H, A000H, or E000H. D will always have memory starting at 3000H, 7000H, B000H, or F000H.

II-8. Since the memory starting point of the board dictates which section has the starting (lowest) address, it is important that you consider the addressing examples given below.

EXAMPLE A. 5 KBytes of memory to be put on board, starting at 3000H. 8 KBytes of memory space will be used. Section D will be completely filled and U6 and U14 of section A will be filled. Memory addressing increases from top to bottom of each section. Low limit address switches of S1 (A-D) will be OFF, OFF, ON, ON. High limit (E-H) will be OFF, ON, OFF, OFF.

MWRITE SELECTION

No front panel or no MWRITE - Jumper D2 to D3
Front panel with MWRITE - Jumper D1 to D2

WAIT STATE JUMPER SELECTION

WAIT STATE	JUMPER D7 TO
0	D4
1	D5
2	D6
3	D8

NOTE

U30 DOES NOT HAVE TO BE INSTALLED IF 0 OR 1 WAIT STATE IS SELECTED.

MEMORY ADDRESS RANGE SELECT (S1)

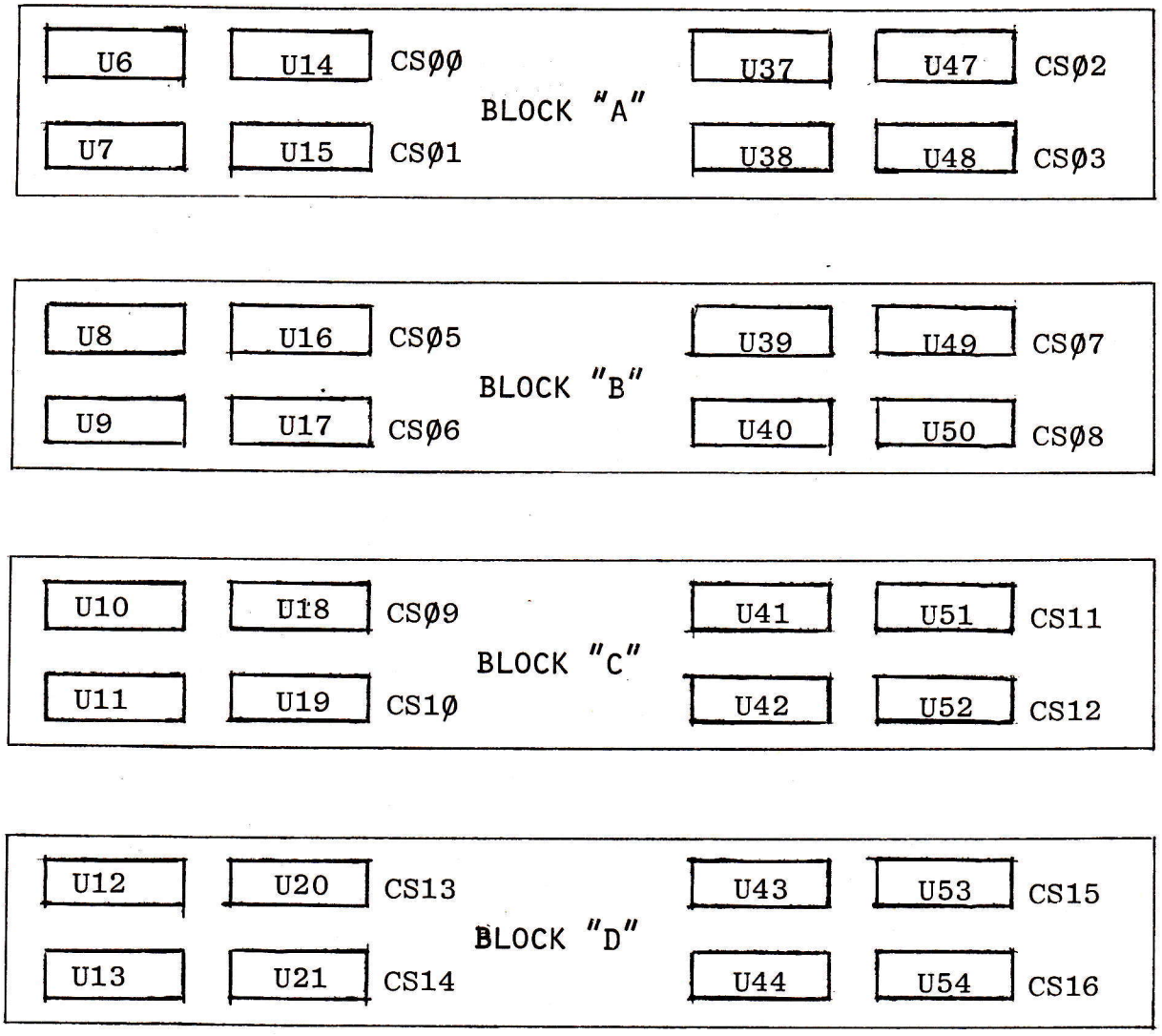
LOW LIMIT				HIGH LIMIT					
A	B	C	D	E	F	G	H	ADDRESS RANGE	SECTION
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	0000-0FFF	A
OFF	OFF	OFF	ON	OFF	OFF	OFF	ON	1000-1FFF	B
OFF	OFF	ON	OFF	OFF	OFF	ON	OFF	2000-2FFF	C
OFF	OFF	ON	ON	OFF	OFF	ON	ON	3000-3FFF	D
OFF	ON	OFF	OFF	OFF	ON	OFF	OFF	4000-4FFF	A
OFF	ON	OFF	ON	OFF	ON	OFF	ON	5000-5FFF	B
OFF	ON	ON	OFF	OFF	ON	ON	OFF	6000-6FFF	C
OFF	ON	ON	ON	OFF	ON	ON	ON	7000-7FFF	D
ON	OFF	OFF	OFF	ON	OFF	OFF	OFF	8000-8FFF	A
ON	OFF	OFF	ON	ON	OFF	OFF	ON	9000-9FFF	B
ON	OFF	ON	OFF	ON	OFF	ON	OFF	A000-AFFF	C
ON	OFF	ON	ON	ON	OFF	ON	ON	B000-BFFF	D
ON	ON	OFF	OFF	ON	ON	OFF	OFF	C000-CFFF	A
ON	ON	OFF	ON	ON	ON	OFF	ON	D000-DFFF	B
ON	ON	ON	OFF	ON	ON	ON	OFF	E000-EFFF	C
ON	ON	ON	ON	ON	ON	ON	ON	F000-FFFF	D

ADDRESS SELECT REQUIREMENT

LOW LIMIT SETTING DESIRED ADDRESS RANGE HIGH LIMIT SETTING

Figure 6. MEM-2 Board Configuration

Fig. 7


U6
U14
CSØØ
U37
U47
CSØ2
BLOCK "A"
U7
U15
CSØ1
U38
U48
CSØ3
U8
U16
CSØ5
U39
U49
CSØ7
BLOCK "B"
U9
U17
CSØ6
U40
U50
CSØ8
U10
U18
CSØ9
U41
U51
CS11
BLOCK "C"
U11
U19
CS1Ø
U42
U52
CS12
U12
U20
CS13
U43
U53
CS15
BLOCK "D"
U13
U21
CS14
U44
U54
CS16

EXAMPLE B. 12 KBytes of memory to be put on board, starting at 9000H. 12 KBytes of memory space will be used. Section B will be the lowest address block. Sections B, C, and D will be completely filled. Low limit switches of S1 (A-D) will be ON, OFF, OFF, ON. High limit (E-H) will be ON, OFF, ON, ON.

II-9. Select the address range desired on S1. Insert the memory chips as dictated by the selected memory address. Insure that the polarity of the memory chips are correct (see Figure 3).

II-10. The two horizontially adjacent memory chips in each section are the chip pairs that form a one KByte block. Memory increases in address from top to bottom of each section.

II-11. Ensure computer is OFF. Plug the MEM-2 into the motherboard. Check that MEM-2 is correctly plugged in and that the board is fully seated in the connector.

NOTE

WHEN POWER IS APPLIED TO AN 8080 SYSTEM, THE MICROPROCESSOR DOES NOT COME UP IN ANY DETERMINABLE STATE. TO CORRECTLY INITIALIZE THE COMPUTER, HOLD THE STOP SWITCH IN STOP AND PUSH THE RESET TO RESET.

II-12. There are three different programs given to check out the memory of your board. They are:

1. Memory Address/Checkerboard Test (RTEST)
2. Walking One (WLKON)
3. Walking Zero (WLKZR)

II-13. All three tests require that your board be configured for address 0000H. If your computer will not allow this address to be used, you will have to modify the programs to use an allowable address range.

II-14. After configuring your board as above, turn your computer ON and input your choice of the memory tests given. RTEST will only take about a second to run once it has been inputted, WLKON and WLKZR take an expodential time to run as more and more memory is tested. It is therefore recommended that you not run more than 2 KBytes at a time on these two tests.

II-15. These programs are very basic and are given to help you debug your board. They will stop upon completion or when an error is found. It may therefore be needed to run them a series of times until all errors have been found and corrected. It is possible to modify the programs so that they will execute until completion and give an output of all errors found. We leave this modification up to you as an exercise in assembly language programming. The programs are written to be input from your front panel. If you don't have a front panel or if you wish to use another input device, you will have to modify the programs to meet your system requirements.

III. BANK ADDRESSING

III-1. Switch S2 controls the bank addressing capabilities of MEM-2. If S2 is not installed, the board will respond if the address presented to it falls within the limits selected by switch S1.

III-2. If S2 is installed, MEM-2 will respond if and only if the address is within the limits selected by S1 and the CPU has selected a memory bank corresponding to the setting of S2.

III-3. If MEM-2 is to be used in an application using bank addressing, use Figure 8 to select the correct bank.

BANK ADDRESSING SELECTION (S2)

A	B	C	D	E	F	G	H	BANK
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	1
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	2
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	3
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	4
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	5
OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	6
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	7
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	8

Figure 8. MEM-2 Bank Addressing Selection

CAUTION

ENSURE ONLY ONE SWITCH IS ON AT ANY TIME. MULTIPLE ON SETTINGS WILL CONFUSE THE BOARD.

NOTE

ALL SWITCHES OFF DISABLE THE BANK SELECT OPTION.

IV. GENERAL

IV-1. The WAMECO INCORPORATED product you have purchased has an unlimited guarantee good for a period of ninety (90) days from date of purchase from your dealer against defects in manufacturing. Upon receipt of the defective board by WAMECO INCORPORATED, pre-paid freight or mailing, the board will be replaced and your shipping charges cheerfully refunded. The guaranty is limited to replacement of the board with an equivalent board even though the board may be defective through negligence in manufacturing or through other fault.

IV-2. For future reference, a print of the front and back traces of the MEM-2 is shown (see Figure 9A and B).

IV-3. We sincerely hope that the MEM-2 will give you long and satisfactory service. If you have any problems with the MEM-2, or if you just want to comment on the board, please write to me personally.

Norm Walters
President
WAMECO INCORPORATED
3107 Laneview Drive
San Jose, Ca. 95132

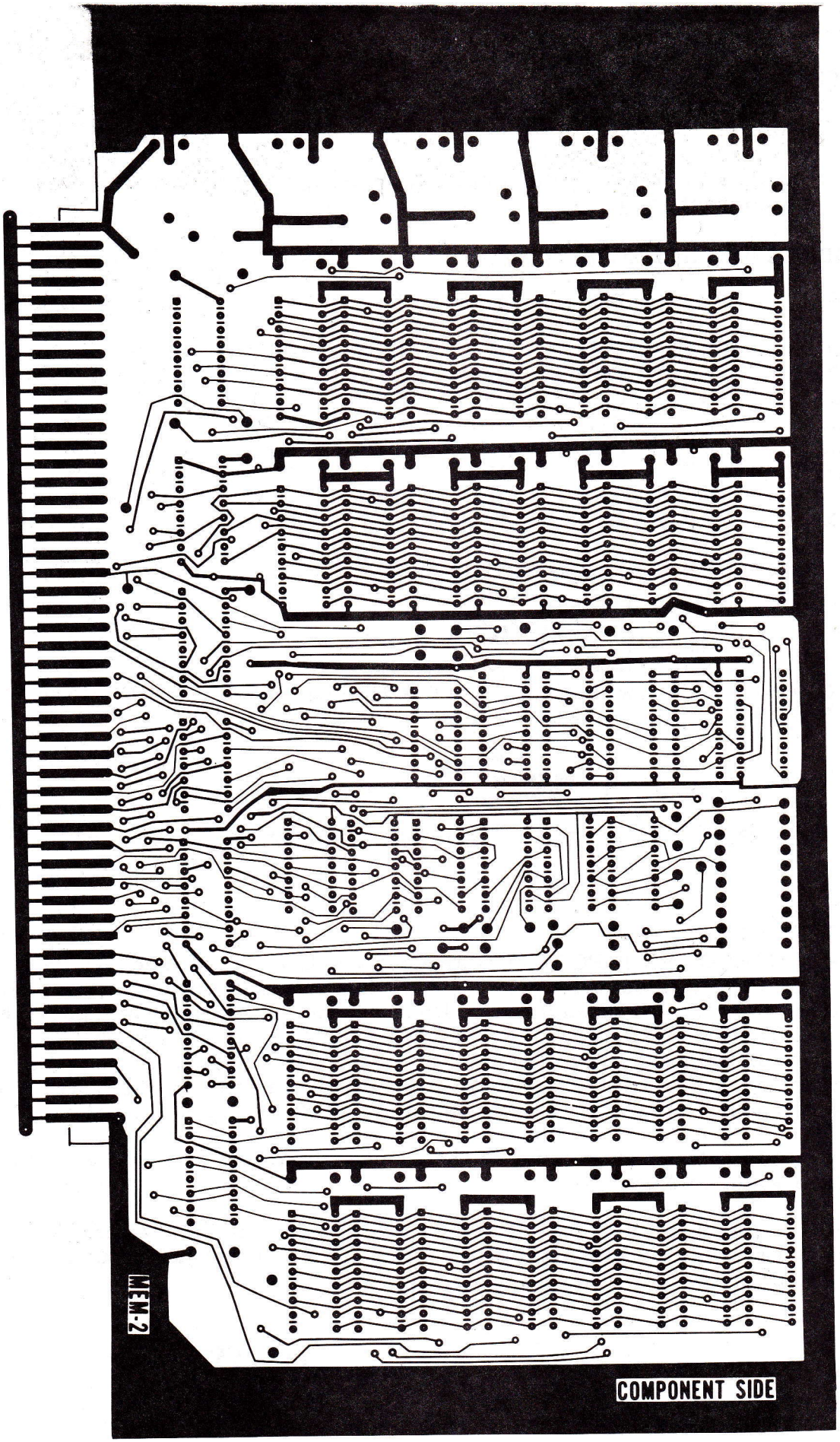


Figure 9A. MEM-2 Component Side

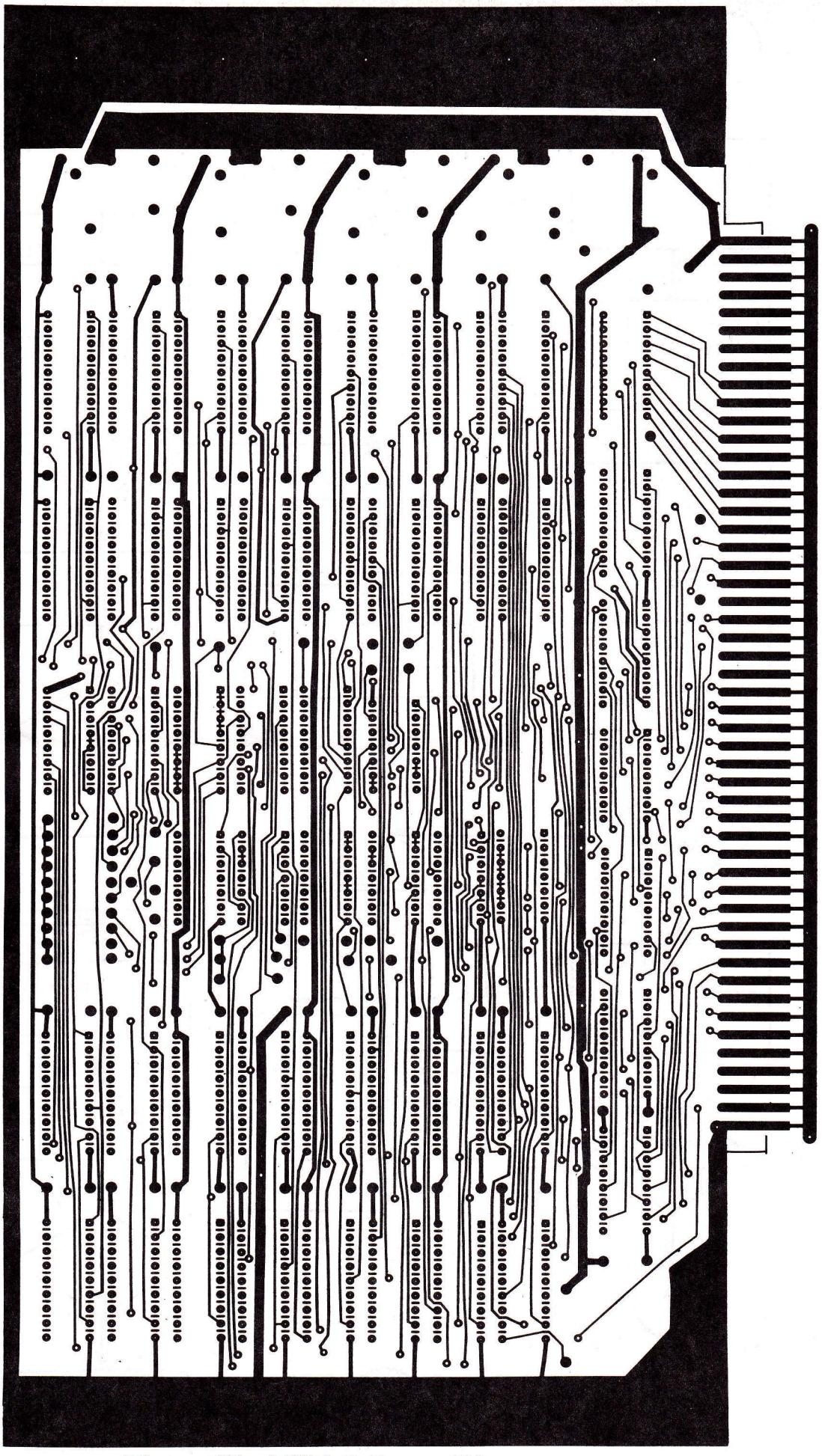
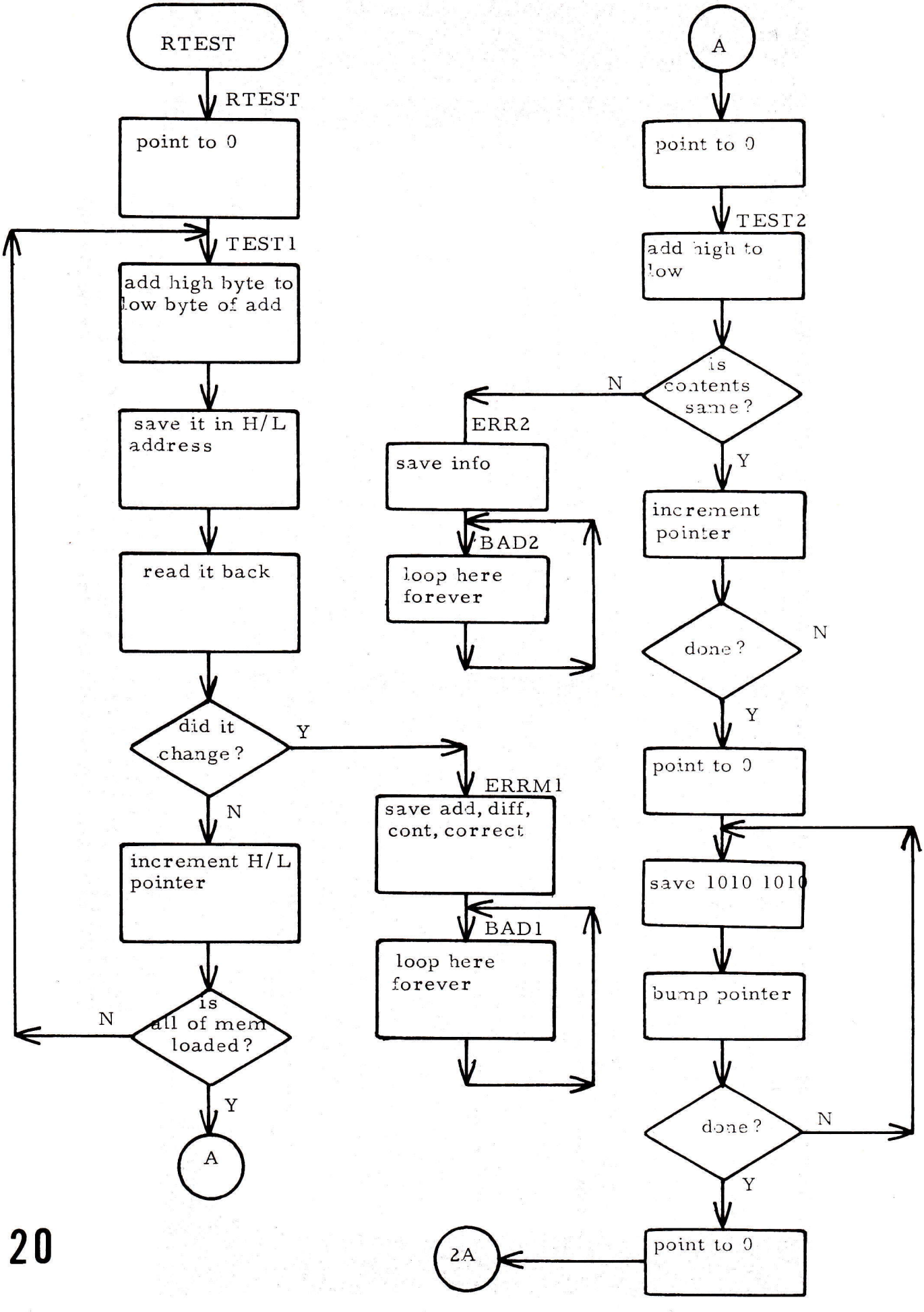
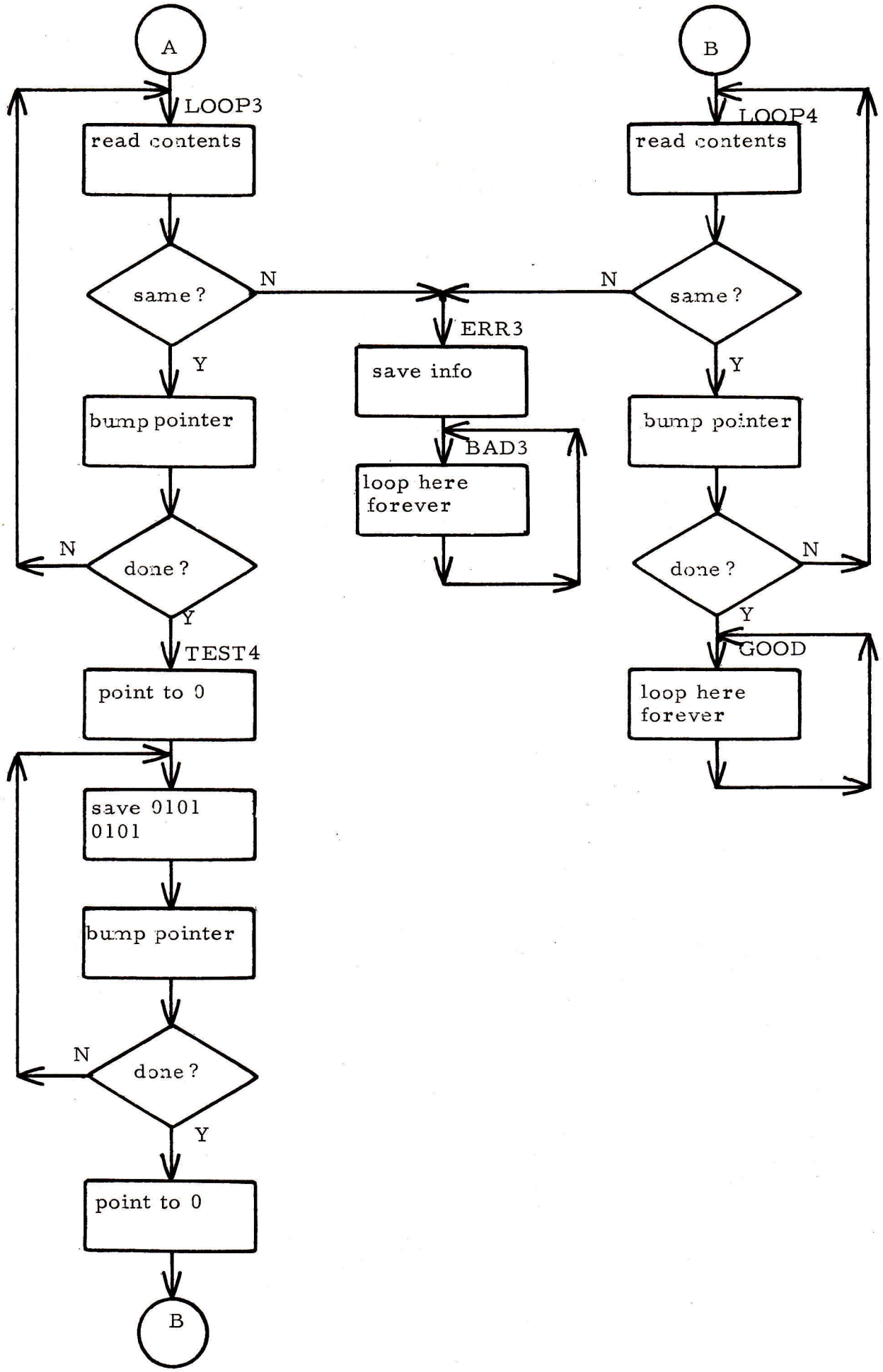


Figure 9B. MEM-2 Trace Side


RTEST
RTEST
point to 0
TEST1
add high byte to low byte of add
save it in H/L address
read it back
did it change?
Y
N
ERRM1
save add, diff, cont, correct
BAD1
loop here forever
increment H/L pointer
is all of mem loaded?
N
Y
A
A
point to 0
TEST2
add high to low
is contents same?
N
ERR2
save info
BAD2
loop here forever
Y
increment pointer
done?
N
Y
point to 0
save 1010 1010
bump pointer
done?
N
Y
point to 0
2A


A
LOOP3
read contents
same ?
N
Y
bump pointer
done ?
N
Y
TEST4
point to 0
save 0101
0101
bump pointer
done ?
N
Y
point to 0
B
ERR3
save info
BAD3
loop here
forever
B
LOOP4
read contents
same ?
N
Y
bump pointer
done ?
N
Y
GOOD
loop here
forever

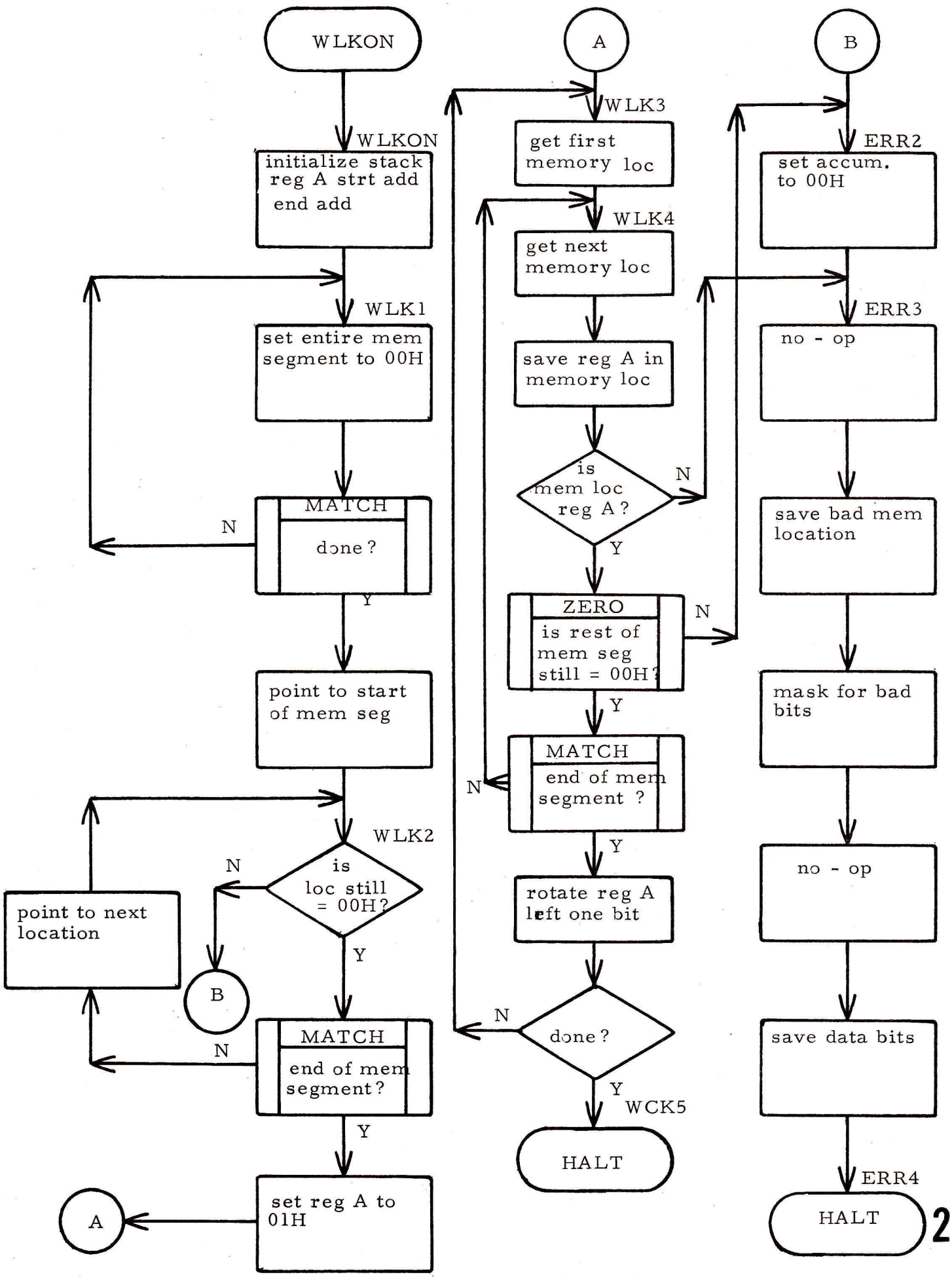
```
ASM80 :F1:8KRAM PRINT(:LP:)


ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        RAM8K    PAGE    1
8K RAM TEST

  LOC  OBJ        SEQ        SOURCE STATEMENT

                    1 $       MACROFILE DEBUG NOSYMBOLS XREF
                    2 $       TITLE ('8K RAM TEST')
                    3
                    4 ;  8K RAM TEST - RAM BOARD AT LOCATION 0 IS TESTED
                    5 ;
                    6 ;  A GOOD TEST LOOPS AT X05C
                    7 ;
                    8 ;  A FAILURE TO WRITE THE ADDRESS CORRECTLY LOOPS AT X06D
                    9 ;
                   10 ;  A FAILURE TO READ BACK THE ADDRESS CORRECTLY LOOPS A X081
                   11 ;
                   12 ;  A FAILURE IN THE CHECKERBOARD TEST LOOPS AT X092
                   13 ;
                   14 ;  ALL FAILURE ROUTINES LEAVE
                   15 ;       LOC             CONTENTS
                   16 ;       0-1             FAILING RAM ADDRESS
                   17 ;       2               DIFFERENCES IN THE RAM LOCATION
                   18 ;       3               EXPECTED RAM CONTENTS
                   19 ;       4               ACTUAL RAM CONTENTS
                   20 ;
                   21
                   22         NAME RAM8K
                   23
                   24         CSEG
                   25
                   26 ;
                   27 ;  8K RAM ADDRESS TEST
                   28 ;
                   29
0000 210000        30 RTEST:  LXI H,0
0003 7D            31 TEST1:  MOV A,L
0004 84            32         ADD H           ;ADD HIGH BYTE TO LOW
0005 77            33         MOV M,A         ;STORE RESULT
0006 46            34         MOV B,M         ;READ IT BACK
0007 B8            35         CMP B           ;IS IT THE SAME?
0008 C25F00   C    36         JNZ ERR1        ;NO, IMMEDIATE READ BACK ERROR
000B 23            37         INX H
000C 7C            38         MOV A,H
000D FE20          39         CPI 20H         ;DONE WITH PASS I?
000F C20300   C    40         JNZ TEST1       ;NO, LOOP AGAIN
                   41
0012 210000        42         LXI H,0
0015 7D            43 TEST2:  MOV A,L
0016 84            44         ADD H           ;ADD HIGH BYTE TO LOW
0017 BE            45         CMP M           ;COMPARE RAM LOCATION
0018 C27000   C    46         JNZ ERR2        ;DIFFERENT - SOMEONE ELSE OVERWROTE IT
001B 23            47         INX H           ;THIS LOCATION IS OK
001C 7C            48         MOV A,H
001D FE20          49         CPI 20H         ;DONE WITH PASS II?
001F C21500   C    50         JNZ TEST2       ;NO, LOOP AGAIN
                   51
                   52 ;
```

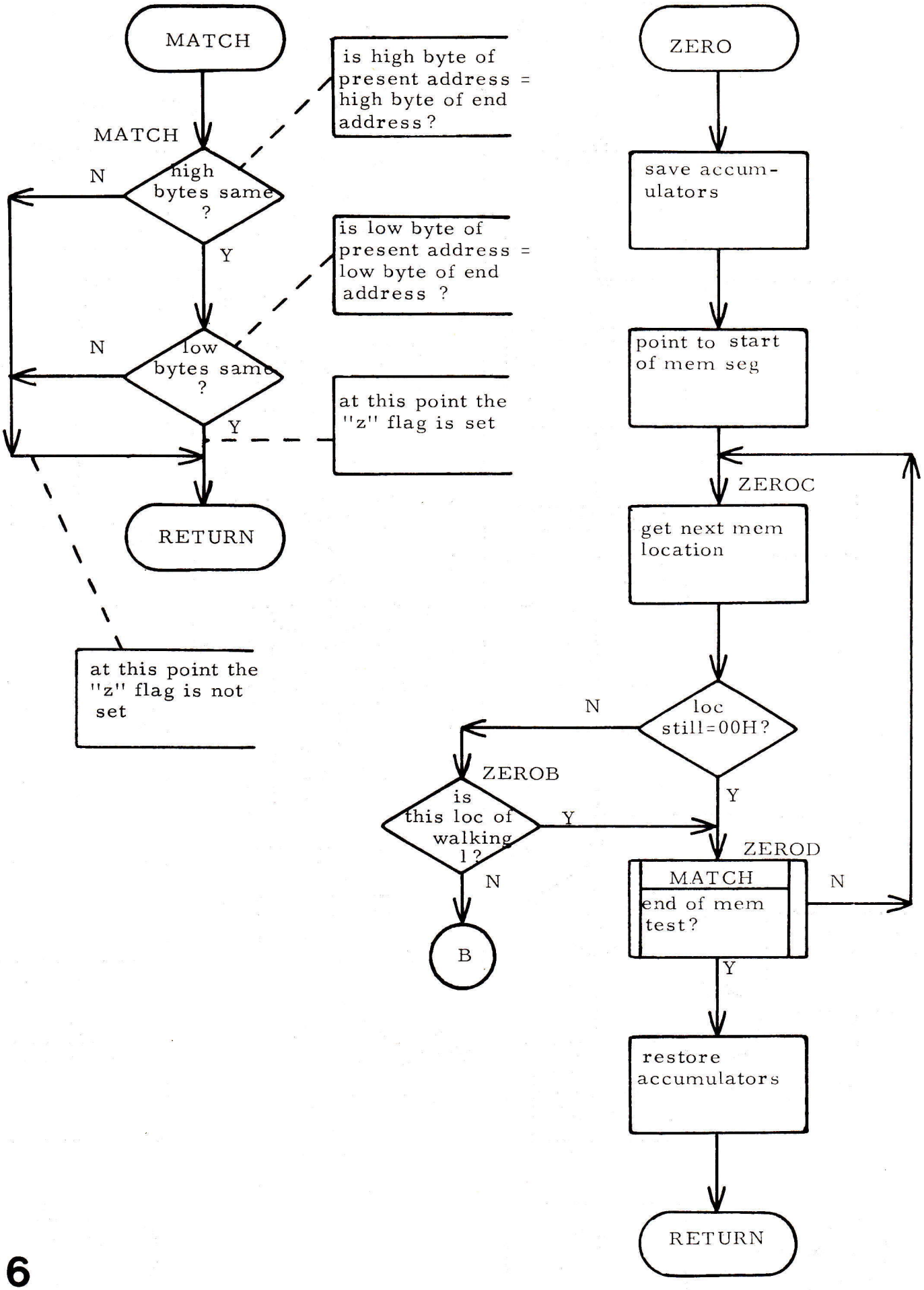
```
LOC  OBJ          SEQ         SOURCE STATEMENT

                   53 ;  CHECKERBOARD TEST
                   54 ;
                   55
0022 210000        56          LXI H,0
0025 3E20          57          MVI A,20H
0027 36AA          58 TEST3:   MVI M,0AAH      ;SAVE 1010 1010
0029 23            59          INX H
002A BC            60          CMP H           ;DONE YET?
002B C22700    C   61          JNZ TEST3       ;NO, LOOP AGAIN
002E 210000        62          LXI H,0         ;POINT TO START
0031 06AA          63          MVI B,0AAH
0033 7E            64 LOOP3:   MOV A,M         ;READ BACK CONTENTS
0034 B8            65          CMP B           ;STILL 0AAH?
0035 C28400    C   66          JNZ ERR3        ;NO, CHECKERBOARD ERROR
0038 23            67          INX H
0039 7C            68          MOV A,H
003A FE20          69          CPI 20H         ;DONE?
003C C23300    C   70          JNZ LOOP3       ;NO
003F 3E20          71          MVI A,20H
0041 210000        72          LXI H,0         ;POINT TO START OF TEST
0044 3655          73 TEST4:   MVI M,55H       ;SAVE 0101 0101
0046 23            74          INX H
0047 BC            75          CMP H           ;END OF SEGMENT?
0048 C24400    C   76          JNZ TEST4       ;NO, LOOP AGAIN
004B 210000        77          LXI H,0         ;DONE, POINT TO START
004E 0655          78          MVI B,55H
0050 7E            79 LOOP4:   MOV A,M         ;READ BACK CONTENTS
0051 B8            80          CMP B           ;STILL 55H?
0052 C28400    C   81          JNZ ERR3        ;NO, CHECKERBOARD ERROR
0055 23            82          INX H
0056 7C            83          MOV A,H
0057 FE20          84          CPI 20H         ;END OF SEGMENT?
0059 C25000    C   85          JNZ LOOP4       ;NO, LOOP AGIAN
                   86
005C C35C00    C   87 GOOD:    JMP GOOD        ;GOOD TEST !!!!!!
                   88
                   89 ;
                   90 ;  ERROR ROUTINES
                   91 ;
                   92
005F 220000        93 ERR1:    SHLD 0          ;SAVE BAD ADDRESS
0062 320300        94          STA 3           ;SAVE WHAT CONTENTS SHOULD BE
0065 A8            95          XRA B
0066 320200        96          STA 2           ;SAVE DIFFERENCES
0069 78            97          MOV A,B
006A 320400        98          STA 4           ;SAVE ACTUAL CONTENTS
006D C36D00    C   99 BAD1:    JMP BAD1        ;AND LOOP AT BAD1
                  100
                  101 ;  READ BACK ERROR - ADDRESS TEST
                  102
0070 220000       103 ERR2:    SHLD 0          ;SAVE BAD ADDRESS
0073 46           104          MOV B,M
0074 7C           105          MOV A,H
0075 85           106          ADD L
0076 320300       107          STA 3           ;SAVE WHAT CONTENTS SHOULD BE
```

8K RAM TEST

```
 LOC  OBJ        SEQ        SOURCE STATEMENT

 0079 A8          108        XRA B
 007A 320200      109        STA 2          ;SAVE DIFFERENCES
 007D 78          110        MOV A,B
 007E 320400      111        STA 4          ;SAVE ACTUAL CONTENTS
 0081 C38100   C  112 BAD2:   JMP BAD2       ;AND LOOP AT BAD2
                  113
                  114 ; CHECKERBOARD ERROR
                  115
 0084 220000      116 ERR3:   SHLD 0         ;SAVE BAD ADDRESS
 0087 320400      117        STA 4          ;SAVE ACTUAL CONTENTS
 008A A8          118        XRA B
 008B 320200      119        STA 2          ;SAVE DIFFERENCES
 008E 78          120        MOV A,B
 008F 320300      121        STA 3          ;SAVE WHAT CONTENTS SHOULD BE
 0092 C39200   C  122 BAD3:   JMP BAD3       ;AND LOOP AT BAD3
                  123
                  124        END

ASSEMBLY COMPLETE,  NO ERRORS
```

```
 ISIS-II ASSEMBLER SYMBOL CROSS REFERENCE, V2.0

 BAD1     99#    99
 BAD2    112#   112
 BAD3    122#   122
 ERR1     36     93#
 ERR2     46    103#
 ERR3     66     81    116#
 GOOD     87#    87
 LOOP3    64#    70
 LOOP4    79#    85
 RAM8K    22
 RTEST    30#
 TEST1    31#    40
 TEST2    43#    50
 TEST3    58#    61
 TEST4    73#    76

 CROSS REFERENCE COMPLETE
```


WLKON
WLKON
initialize stack reg A strt add end add
WLK1
set entire mem segment to 00H
MATCH
done?
N
Y
point to start of mem seg
WLK2
is loc still = 00H?
N
Y
point to next location
B
MATCH
end of mem segment?
N
Y
set reg A to 01H
A
A
WLK3
get first memory loc
WLK4
get next memory loc
save reg A in memory loc
is mem loc reg A?
N
Y
ZERO
is rest of mem seg still = 00H?
N
Y
MATCH
end of mem segment ?
N
Y
rotate reg A left one bit
done?
N
Y
WCK5
HALT
B
ERR2
set accum. to 00H
ERR3
no - op
save bad mem location
mask for bad bits
no - op
save data bits
ERR4
HALT

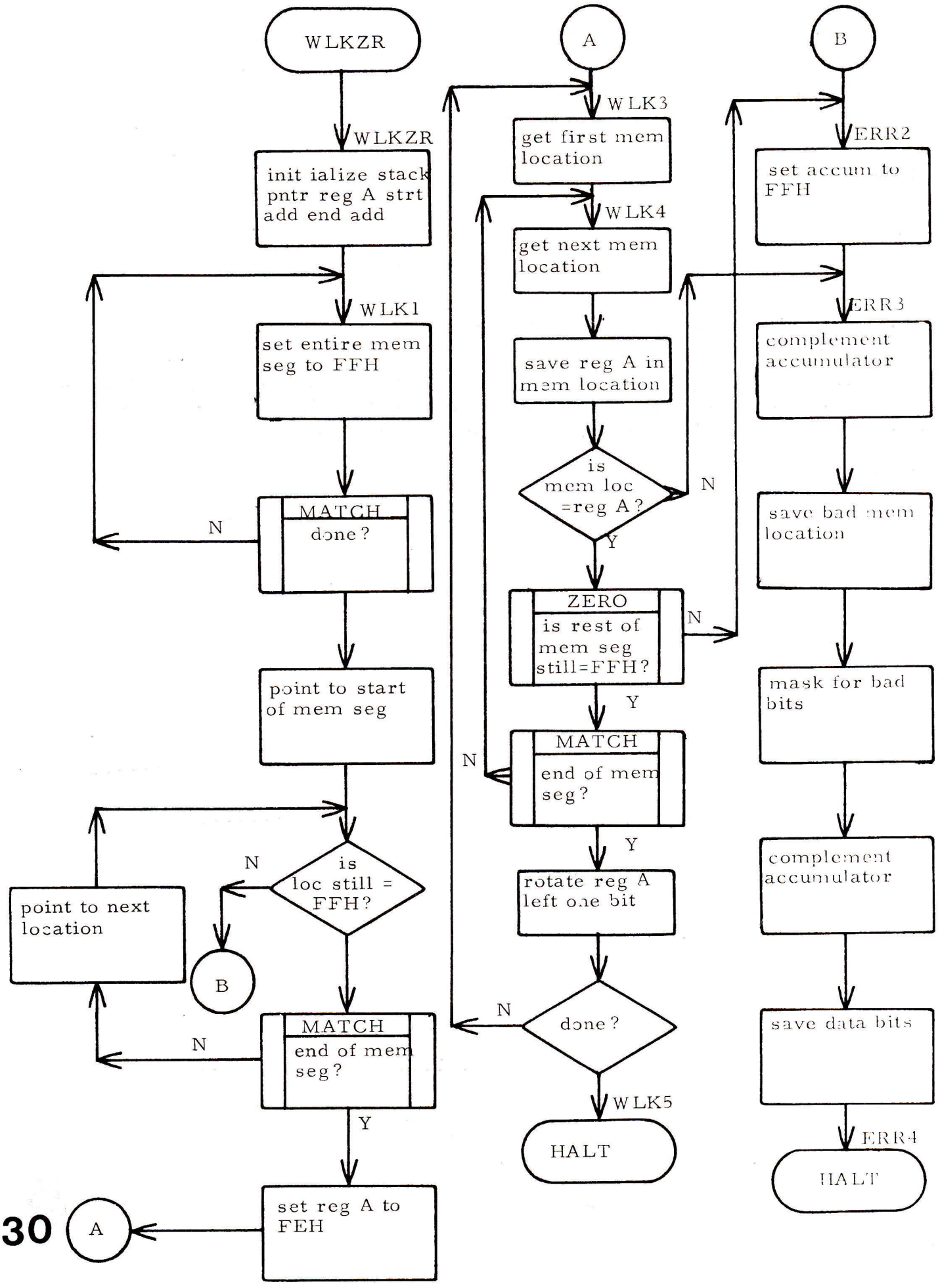

MATCH
is high byte of present address = high byte of end address?
MATCH
high bytes same ?
N
Y
is low byte of present address = low byte of end address ?
low bytes same ?
N
Y
at this point the "z" flag is set
RETURN
at this point the "z" flag is not set
ZERO
save accum-ulators
point to start of mem seg
ZEROC
get next mem location
loc still=00H?
N
Y
ZEROB
is this loc of walking 1 ?
Y
N
B
ZEROD
MATCH
end of mem test?
N
Y
restore accumulators
RETURN

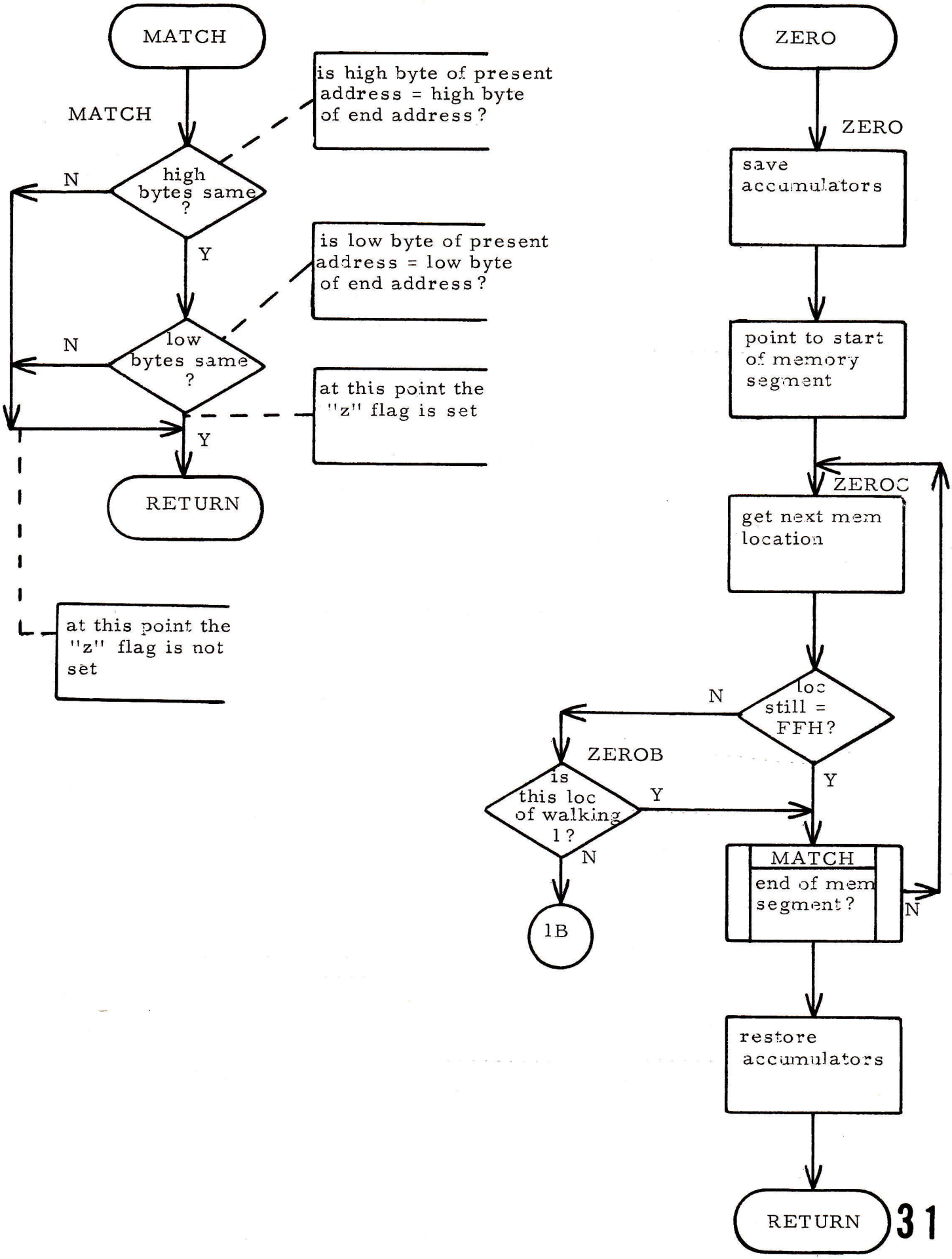
```
ASM80 :F1:WLKON PRINT(:LP:) MACROFILE


ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        WLKON    PAGE    1
WALKING ONE MEMORY TEST - V1.0

 LOC  OBJ         SEQ         SOURCE STATEMENT

                    1 $       TITLE ('WALKING ONE MEMORY TEST - V1.0')
                    2
                    3 ;
                    4 ;  WALKING ONE
                    5 ;
                    6 ;  THIS ROUTINE WILL ZERO MEMORY, THEN SET ONE BIT HIGH
                    7 ;  AND CHECK ALL REMAINING BITS IN MEMORY FOR ZERO.
                    8 ;  THIS PROCESS OCCURS FOR EVERY BIT IN MEMORY.
                    9 ;
                   10 ;  DUE TO EXECUTION TIME, WE SUGGEST YOU TEST NOTHING
                   11 ;  LARGER THAN A 2K SEGMENT (800H)
                   12 ;
                   13
                   14         NAME WLKON
                   15
                   16         CSEG
 0000 3E00         17 WLKON:  MVI A,0
 0002 311300   D   18         LXI SP,BBITS+13          ;SET STACK POINTER
 0005 2A0000   D   19         LHLD STOPA       ;MOVE STOP ADDRESS TO D/E
 0008 EB           20         XCHG
 0009 2A0200   D   21         LHLD STRTA       ;GET START ADDRESS TO H/L
 000C 23           22 WLK1:   INX H
 000D 77           23         MOV M,A          ;ZERO THIS LOCATION
 000E CD7300   C   24         CALL MATCH       ;END OF MEMORY?
 0011 C20C00   C   25         JNZ WLK1         ;NO, DO NEXT LOCATION
 0014 2A0200   D   26         LHLD STRTA       ;GET START
 0017 23           27 WLK2:   INX H
 0018 BE           28         CMP M            ;STILL ZERO?
 0019 C23F00   C   29         JNZ ERR2         ;NO, ERROR
 001C CD7300   C   30         CALL MATCH       ;EOM?
 001F C21700   C   31         JNZ WLK2         ;NO
 0022 3E01         32         MVI A,1          ;PUT A ONE IN ACC
 0024 2A0200   D   33 WLK3:   LHLD STRTA       ;GET START ADDRESS
 0027 23           34 WLK4:   INX H
 0028 77           35         MOV M,A          ;SAVE WALKING ONE IN MEM
 0029 BE           36         CMP M            ;WAS IT SET?
 002A C24100   C   37         JNZ ERR3         ;NO, ERROR
 002D CD4D00   C   38         CALL ZERO        ;REST OF MEMORY STILL ZERO?
 0030 3600         39         MVI M,0          ;YES, RESTORE LOC TO ZERO
 0032 CD7300   C   40         CALL MATCH       ;EOM?
 0035 C22700   C   41         JNZ WLK4         ;NO, TO NEXT LOCATION
 0038 07           42         RLC              ;ROTATE ONE BIT LEFT
 0039 D22400   C   43         JNC WLK3         ;NOT DONE YET
 003C C33C00   C   44 WLK5:   JMP WLK5         ;JUMP TO SELF - ALL LOCS TESTED GOOD
                   45
 003F 3E00         46 ERR2:   MVI A,0
 0041 00           47 ERR3:   NOP
 0042 220400   D   48         SHLD BADDR       ;SAVE BAD ADDRESS
 0045 AE           49         XRA M
 0046 00           50         NOP
 0047 320600   D   51         STA BBITS        ;SAVE BAD BITS
 004A C34A00   C   52 ERR4:   JMP ERR4         ;JUMP TO SELF - ONE LOCATION TESTED BAD
```

```
LOC  OBJ          SEQ        SOURCE STATEMENT

                    53
                    54 ;
                    55 ;  THIS SUBROUTINE WLL CHECK MEMORY TO SEE IF IT IS STILL ZERO.
                    56 ;  IF NOT, IT CHECKS TO SEE IF THIS IS OUR WALKING ONE LOCATION.
                    57 ;
                    58
004D 4F             59 ZERO:   MOV C,A          ;SAVE REGS
004E C5             60         PUSH B
004F E5             61         PUSH H
0050 2A0200    D    62         LHLD STRTA       ;POINT TO START OF MEM
0053 3E00           63 ZEROC:  MVI A,0
0055 23             64         INX H            ;GET NEXT LOC
0056 BE             65         CMP M            ;ZERO?
0057 C26400    C    66         JNZ ZEROB        ;NO
005A CD7300    C    67 ZEROD:  CALL MATCH       ;EOM?
005D C25300    C    68         JNZ ZEROC        ;NO, DO SOMEMORE
0060 E1             69         POP H
0061 C1             70         POP B            ;RESTORE REGS
0062 79             71         MOV A,C
0063 C9             72         RET
0064 C1             73 ZEROB:  POP B
0065 C5             74         PUSH B
0066 7C             75         MOV A,H          ;IS THIS WHERE WALKING
0067 B8             76         CMP B            ;  ONE IS STORED?
0068 C23F00    C    77         JNZ ERR2         ;NO, ERROR #2
006B 7D             78         MOV A,L          ;MAYBE
006C B9             79         CMP C
006D C23F00    C    80         JNZ ERR2         ;NO, ERROR #2
0070 C35A00    C    81         JMP ZEROD        ;YES, DO NEXT LOCATION
                    82
                    83 ;
                    84 ;  THIS SUBROUTINE WILL DETERMINE IF WE HAVE REACHED
                    85 ;  THE END OF MEMORY
                    86 ;
                    87 ;  FLAG Z=0 IF NOT END OF MEMORY
                    88 ;  FLAG Z=1 IF END OF MEMORY
                    89 ;
                    90
0073 47             91 MATCH:  MOV B,A          ;SAVE ACC
0074 7C             92         MOV A,H
0075 BA             93         CMP D            ;DOES HIGH BYTE MATCH?
0076 C27B00    C    94         JNZ MTCHA        ;NO
0079 7D             95         MOV A,L
007A BB             96         CMP E            ;COMPARE WITH LOW BYTE
007B 78             97 MTCHA:  MOV A,B          ;RESTORE ACC
007C C9             98         RET
                    99
                   100         DSEG
0000 FF07          101 STOPA:  DW 7FFH          ;END ADDRESS
0002 9600          102 STRTA:  DW 96H           ;START OF TEST SEGMENT - 1
0004 0000          103 BADDR:  DW 0             ;BAD ADDRESS LOCATION
0006 00            104 BBITS:  DB 0             ;BAD BITS LOCATION
                   105
0000           C   106         END WLKON
```


PUBLIC SYMBOLS

```
EXTERNAL SYMBOLS


USER SYMBOLS


ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        WLKON    PAGE    3
WALKING ONE MEMORY TEST - V1.0

BADDR  D 0004    BBITS  D 0006    ERR2   C 003F    ERR3   C 0041    ERR4   C 004A    MATCH  C 0073    MTCHA  C 007B
STOPR  D 0000    STRTA  D 0002    WLK1   C 000C    WLK2   C 0017    WLK3   C 0024    WLK4   C 0027    WLK5   C 003C
WLKON  C 0000    ZERO   C 004D    ZERO8  C 0064    ZEROC  C 0053    ZEROD  C 005A

ASSEMBLY COMPLETE,  NO ERRORS
```


WLKZR
WLKZR
init ialize stack pntr reg A strt add end add
WLK1
set entire mem seg to FFH
MATCH
done?
N
point to start of mem seg
is loc still = FFH?
N
point to next location
B
MATCH
end of mem seg?
N
Y
set reg A to FEH
A
A
WLK3
get first mem location
WLK4
get next mem location
save reg A in mem location
is mem loc =reg A?
N
Y
ZERO
is rest of mem seg still=FFH?
N
Y
MATCH
end of mem seg?
N
Y
rotate reg A left one bit
done?
N
WLK5
HALT
B
ERR2
set accum to FFH
ERR3
complement accumulator
save bad mem location
mask for bad bits
complement accumulator
save data bits
ERR4
HALT


MATCH
MATCH
high bytes same ?
N
Y
is high byte of present address = high byte of end address?
low bytes same ?
N
Y
is low byte of present address = low byte of end address?
at this point the "z" flag is set
RETURN
at this point the "z" flag is not set
ZERO
ZERO
save accumulators
point to start of memory segment
ZEROC
get next mem location
loc still = FFH?
N
Y
ZEROB
is this loc of walking 1?
Y
N
1B
MATCH
end of mem segment?
N
restore accumulators
RETURN

```
ASM80 :F1:WLKZR PRINT(:LP:) MACROFILE


ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        WLKZR    PAGE    1
WALKING ZERO MEMORY TEST - V1.0

 LOC  OBJ         SEQ         SOURCE STATEMENT

                    1 $       TITLE ('WALKING ZERO MEMORY TEST - V1.0')
                    2
                    3 ;
                    4 ;  WALKING ZERO
                    5 ;
                    6 ;  THIS ROUTINE WILL SET MEMORY TO ONES, THEN SET ONE BIT TO ZERO
                    7 ;  AND CHECK ALL REMAINING BITS IN MEMORY FOR ONES.
                    8 ;  THIS PROCESS OCCURS FOR EVERY BIT IN MEMORY.
                    9 ;
                   10 ;  DUE TO EXECUTION TIME, WE SUGGEST YOU TEST NOTHING
                   11 ;  LARGER THAN A 2K SEGMENT (800H)
                   12 ;
                   13
                   14         NAME WLKZR
                   15
                   16         CSEG
 0000 3EFF         17 WLKZR:  MVI A,0FFH
 0002 311300   D   18         LXI SP,BBITS+13          ;SET STACK POINTER
 0005 2A0000   D   19         LHLD STOPA       ;MOVE STOP ADDRESS TO D/E
 0008 EB           20         XCHG
 0009 2A0200   D   21         LHLD STRTA       ;GET START ADDRESS TO H/L
 000C 23           22 WLK1:   INX H
 000D 77           23         MOV M,A          ;SET THIS LOCATION
 000E CD7300   C   24         CALL MATCH       ;END OF MEMORY?
 0011 C20C00   C   25         JNZ WLK1         ;NO, DO NEXT LOCATION
 0014 2A0200   D   26         LHLD STRTA       ;GET START
 0017 23           27 WLK2:   INX H
 0018 BE           28         CMP M            ;STILL ONES?
 0019 C23F00   C   29         JNZ ERR2         ;NO, ERROR
 001C CD7300   C   30         CALL MATCH       ;EOM?
 001F C21700   C   31         JNZ WLK2         ;NO
 0022 3EFE         32         MVI A,0FEH               ;PUT A ZERO IN ACC
 0024 2A0200   D   33 WLK3:   LHLD STRTA       ;GET START ADDRESS
 0027 23           34 WLK4:   INX H
 0028 77           35         MOV M,A          ;SAVE WALKING ZERO IN MEM
 0029 BE           36         CMP M            ;WAS IT SET?
 002A C24100   C   37         JNZ ERR3         ;NO, ERROR
 002D CD4D00   C   38         CALL ZERO        ;REST OF MEMORY STILL ONES?
 0030 36FF         39         MVI M,0FFH       ;YES, RESTORE LOC TO ONES
 0032 CD7300   C   40         CALL MATCH       ;EOM?
 0035 C22700   C   41         JNZ WLK4         ;NO, TO NEXT LOCATION
 0038 07           42         RLC              ;ROTATE ONE BIT LEFT
 0039 D22400   C   43         JNC WLK3         ;NOT DONE YET
 003C C33C00   C   44 WLK5:   JMP WLK5         ;JUMP TO SELF - ALL LOCS TESTED GOOD
                   45
 003F 3EFF         46 ERR2:   MVI A,0FFH
 0041 2F           47 ERR3:   CMA
 0042 220400   D   48         SHLD BADDR       ;SAVE BAD ADDRESS
 0045 AE           49         XRA M
 0046 2F           50         CMA
 0047 320600   D   51         STA BBITS        ;SAVE BAD BITS
 004A C34A00   C   52 ERR4:   JMP ERR4         ;JUMP TO SELF - ONE LOCATION TESTED BAD
```

WALKING ZERO MEMORY TEST - V1.0

```
 LOC  OBJ          SEQ        SOURCE STATEMENT

                    53
                    54 ;
                    55 ;  THIS SUBROUTINE WLL CHECK MEMORY TO SEE IF IT IS STILL ONES.
                    56 ;  IF NOT, IT CHECKS TO SEE IF THIS IS OUR WALKING ZERO LOCATION.
                    57 ;
                    58
 004D 4F            59 ZERO:   MOV C,A         ;SAVE REGS
 004E C5            60         PUSH B
 004F E5            61         PUSH H
 0050 2A0200   D    62         LHLD STRTA      ;POINT TO START OF MEM
 0053 3EFF          63 ZEROC:  MVI A,0FFH
 0055 23            64         INX H           ;GET NEXT LOC
 0056 BE            65         CMP M           ;ONES?
 0057 C26400   C    66         JNZ ZEROB       ;NO
 005A CD7300   C    67 ZEROD:  CALL MATCH       ;EOM?
 005D C25300   C    68         JNZ ZEROC       ;NO, DO SOMEMORE
 0060 E1            69         POP H
 0061 C1            70         POP B           ;RESTORE REGS
 0062 79            71         MOV A,C
 0063 C9            72         RET
 0064 C1            73 ZEROB:  POP B
 0065 C5            74         PUSH B
 0066 7C            75         MOV A,H         ;IS THIS WHERE WALKING
 0067 B8            76         CMP B           ;  ZERO IS STORED?
 0068 C23F00   C    77         JNZ ERR2        ;NO, ERROR #2
 006B 7D            78         MOV A,L         ;MAYBE
 006C B9            79         CMP C
 006D C23F00   C    80         JNZ ERR2        ;NO, ERROR #2
 0070 C35A00   C    81         JMP ZEROD       ;YES, DO NEXT LOCATION
                    82
                    83 ;
                    84 ;  THIS SUBROUTINE WILL DETERMINE IF WE HAVE REACHED
                    85 ;  THE END OF MEMORY
                    86 ;
                    87 ;  FLAG Z=0 IF NOT END OF MEMORY
                    88 ;  FLAG Z=1 IF END OF MEMORY
                    89 ;
                    90
 0073 47            91 MATCH:  MOV B,A         ;SAVE ACC
 0074 7C            92         MOV A,H
 0075 BA            93         CMP D           ;DOES HIGH BYTE MATCH?
 0076 C27B00   C    94         JNZ MTCHA       ;NO
 0079 7D            95         MOV A,L
 007A BB            96         CMP E           ;COMPARE WITH LOW BYTE
 007B 78            97 MTCHA:  MOV A,B         ;RESTORE ACC
 007C C9            98         RET
                    99
                   100         DSEG
 0000 FF07         101 STOPA:  DW 7FFH         ;END ADDRESS
 0002 9600         102 STRTA:  DW 96H          ;START OF TEST SEGMENT - 1
 0004 0000         103 BADDR:  DW 0            ;BAD ADDRESS LOCATION
 0006 00           104 BBITS:  DB 0            ;BAD BITS LOCATION
                   105
 0000         C    106         END WLKZR

PUBLIC SYMBOLS
```

```
EXTERNAL SYMBOLS


USER SYMBOLS


ISIS-II 8080/8085 MACRO ASSEMBLER, V2.0        WLKZR    PAGE    3
WALKING ZERO MEMORY TEST - V1.0

BADDR  D 0004    BBITS  D 0006    ERR2   C 003F    ERR3   C 0041    ERR4   C 004A    MATCH  C 0073    MTCHA  C 007B
STOPA  D 0000    STRTA  D 0002    WLK1   C 000C    WLK2   C 0017    WLK3   C 0024    WLK4   C 0027    WLK5   C 003C
WLKZR  C 0000    ZERO   C 004D    ZEROB  C 0064    ZEROC  C 0053    ZEROD  C 005A

ASSEMBLY COMPLETE,  NO ERRORS
```