MICROMATION

Z-64

OPERATOR'S MANUAL

D01 006 REV D

The illustrations on pages 15, 16, 20, 21, 22, 23, 24, 25, and 26, and all of Appendix A are reproduced with the permission of Zilog®, Inc Before this or any other Zilog copyrighted material can be reproduced, in whole or in part, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, prior written approval must be obtained from Zilog, Inc.

Regarding the rest of this document, the same restrictions apply. If you would like to duplicate, regardless of the means, any portion of the text not covered by the statement above, write a letter requesting permission first. Specify the pages needed and the application of the duplicated text when filing the request. We reserve the right to refuse duplication rights.

Note that Z-80, the designation used throughout this manual, and Z80 refer to the same microprocessor and that both designations are registered trademarks of Zilog, Inc.

12/79

4/22/80

Z-64 REVISION 7
ADDENDUM

The replacement pages provided for the Z 64 manual inadvertently omitted instructions for grabbing the reset signal from the S-100 bus. This feature is still available; it was not removed from the board with the revision 7 changes.

There are two options regarding the source of the reset signal to the Z-64:

1) the header mounted on the board
2) S-100 bus from pin 75

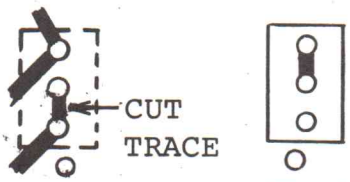
Selection of the two options is a function of jumper JP15. (This jumper is not described in the manual) It is located just to the left of the Z-80 chip and consists of three holes arranged vertically The figure below illustrates the jumper connections relative to each option. Notice that Option 2 includes a jumper wire between the outside pins of J1 in addition to the JP15 cut and jumper.

OPTION 1, RESET FROM J1

JP 15
SOLDER SIDE

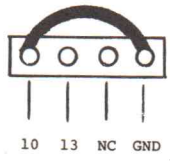
JP 15
COMPONENT SIDE

OPTION 2, RESET FROM BUS



JP 15
SOLDER SIDE

JP 15
COMPONENT SIDE



J1
COMPONENT SIDE

RESET OPTION SELECTION

OPTION 1: This is the default option for all Z-64s shipped in Z-PLUS systems (The Z 64 manual speaks to this option.) A trace is present between the bottom and the middle holes on the solder (back) side of the board precluding the need for a jumper wire

OPTION 2: This is the default option for all Z-64s not shipped in Micromation systems. Board reset is implemented from S 100 bus pin 75. One cut is made and two jumpers are installed to utilize this option

- the trace between the middle and bottom holes of JP15 on the solder (back) side of the board is cut
- a jumper is installed between the top and middle holes of JP15
- an insulated wire is installed connecting the two outside pins of the J1 header

The figure on the following page illustrates the reset circuitry on the Z-64 with the addition of JP15.

MICROMATION

Z-64

The Micromation Z-64 is a complete microcomputer on a single, S-100 bus compatible printed circuit board. It combines 64K of fast dynamic RAM with the popular Z-80A microprocessor. Optional features allow installation of an EPROM and/or a vectored interrupt priority encoder. Installed with the Micromation DOUBLER, the first reliable double density floppy disk controller, users have a complete computer system - CPU, 64K of RAM, floppy disk storage, and serial port suitable for a terminal - on just **two** cards. Users familiar with the early microcomputer systems requiring as many as 20 separate PC boards to perform the same functions will appreciate the ease of installation and use of the Z-64 and DOUBLER.

The Z-64 can be used without the DOUBLER to upgrade S-100 bus systems bound by slower processors and RAM. The Z-64 operates at a fast 4 MHz (jumpers are present to configure for 2 MHz if necessary) and the RAM features a 200 nanosecond access time. And, since the entire 64K is on the same card, there are no problems with RAM board addressing or CPU to RAM communications through the bus.

HOW TO USE THIS MANUAL

This manual is intended primarily for people familiar with microcomputer fundamentals and applications. In fact, many end users will never need to refer to it. System designers and computer dealers will have done the configuring necessary to incorporate the Z-64 into a complete computer system. This is not meant to discourage those curious about micromputers. However, the expressions and concepts require familiarity with computer technology. A short section is included (section III) to introduce some of these notions as they relate to the Z-64, but it is not intended to serve as a tutorial on the subject. (Many introductions to microcomputers can be found at the library or computer store.)

This manual begins with Z-64 installation. **Section I** describes first installation with the DOUBLER and then without. The jumper options are described along with a summary of their purpose. Those installing the Z-64 for the first time without a DOUBLER should not rely upon this section for all the information regarding compatibility with other disk controllers. Section II should be read first. A troubleshooting guide is also included in this section.

Section II contains the Z-64 theory of operation. The board is broken down into the contributing components and analyzed. The complete instructions to option installation and jumper options are presented as part of the component description.

Section III is included to introduce some microcomputer fundamentals and expressions to those unfamiliar with them.

Several appendices conclude this manual. **Appendix A** contains the Z-80 instruction set. **Appendix B** presents a summary of the S-100 bus signals. Micromation provides an assembly language program to test on-board RAM. Those purchasing the Z-64 as part of a complete computer system get this program as a utility on the diskette containing the operating system. A listing of this program is provided in **Appendix C** for those wishing to alter it or those who did not purchase the Z-64 as part of a system. A **schematic** of the Z-64 is provided at the end of this manual.

TABLE OF CONTENTS

Section III: Microprocessors, Microcomputers and the Z-64

Appendices

I. Z-64 INSTALLATION

The Z-64 can be installed in S-100 bus based systems with the Micromation DOUBLER or with another disk controller. This section describes Z-64 installation first with the DOUBLER then installation with a disk controller from another manufacturer. Note that there are certain qualifications if a controller other than the DOUBLER is to be used. These will be described below.

The Z-64 has several optional features. Section I:2 provides a summary of these options and their implementation. The complete description is presented in the relevant portion of section II.

Section I concludes with a troubleshooting guide in the event the Z-64 does not operate as it should.

I:1-1 Z-64 Installation with the Micromation DOUBLER

The Z-64 CPU/RAM card is designed to complement the features of the DOUBLER to provide a complete floppy disk based computer system. With just these two boards, users have a CPU, full 64K of memory, floppy disk interface and serial I/O port suitable for a terminal. Installation is simple and operation equally so.

The first step is to turn the power off and clear your system of other CPU and RAM cards. The only boards in the system should be the Z-64, DOUBLER and I/O board(s) (if any).

At this point, determine if you want the Z-64 to run at 2 or 4 MHz. As shipped from the factory, the jumpers at JP4 are set for 4 MHz operation. (See section I:2-1 below for the changes if 2 MHz is desired.)

The status of the other board jumpers as delivered is:

JP1 (non-maskable interrupt): not connected
JP2 (on-board interrupt): connected
JP3 (off-board interrupt): not connected
JP5 (EPROM power): connected *
JP6 (Power on Jump): not connected
JP7 (EPROM address select): connected **

* This jumper is configured for a 2708 1K EPROM. Refer to sections I:2-3 and II:2-5 for alternate configurations.

** This jumper allows the EPROM to be re-addressed from F800 (the factory setting) to F000. See sections I:2-1 and II:2-5

For installation with the DOUBLER, none of these jumpers need be changed unless an EPROM is to be installed (see sections I:2-1 and II:2-5). However, the DOUBLER must be configured to agree with the Z-64. The table below lists the DOUBLER jumpers and the setting for use with the Z-64.

POJ		installed
PHANTOM		installed
XRDY, PRDY		installed
WAIT:	4 MHz	installed
	2 MHz	removed

The DOUBLER firmware should be addressed for F800H to maximize memory utilization. Other locations for the firmware are permissable and don't require any Z-64 alterations.

If any DOUBLER jumpers require reconfiguring and the board must be removed from the card cage, note the orientation of the cables before removing them. Be sure to reinstall them the same way when the time comes to install the board back in the system.

Once the Z-64 operating frequency has been selected and DOUBLER configured to match, lightly clean off the S-100 edge connector contacts to remove oxidation. **Use a pencil eraser only.** Any other abrasive material will damage the contacts.

Insert the two boards into adjacent S-100 edge connectors. Ensure that each is seated all the way down in the connector. They should both have the component side facing the front. All intercommunication is conducted through the system bus.

If the ribbon cable between the floppy disk drives and DOUBLER was disconnected, reconnect it at this time. Reconnect the terminal cable also if it was disconnected.

The system may now be powered up.

1) Turn on system power
2) Hit the RESET button or switch
3) Insert a write protected floppy disk with the operating system on it in drive A: and close the door.
4) In systems with front panel switches, hit the RUN switch.

The terminal screen should display the operating system sign-on message and prompt.

Remember to hit the RESET button **after** power-up and **before** inserting the system disk every time the system is turned on. During operation, just hit the RESET button to reset the processor and reload the operating system when desired. (There's no need to power down to perform a reset.)

If the procedure above does not perform the indicated result, refer to the Troubleshooting Guide, section I:3 below.

I:1-2 Z-64 Installation without DOUBLER

There's a wide variety of disk controllers available for installation in microcomputers. It would be impossible to cover all contingencies. Before installing the Z-64 in a system with a controller other than the DOUBLER read the rest of the Z-64 documentation to determine compatibility requirements.

Generally, there are two disk controller types that are incompatible with the Z-64.

1) Those disk controllers that use the Western Digital 1771 or 1791 disk controller chips generate a wait state that may extend beyond 2 milliseconds. This paralyzes the Z-80 and prevents refresh of the dynamic RAM. The contents of RAM will be destroyed.

2) Those controllers that use direct memory access (DMA) are not supported. An attempt to reconfigure the Z-64 to allow DMA will void the warranty.

If your disk controller passes both these qualifications, we recommend the installation of a 1K or 2K EPROM as a system monitor. The Z-64 contains power on jump circuitry that accesses the first three memory locations in the EPROM during the first three machine cycles after a reset. These locations should contain a jump instruction and 16 bit address, usually to another location in the EPROM, for a system initialization routine. A phantom line (from S-100 pin 67) is available to disable Z-64 memory (including the EPROM) in deference to the controller firmware. If an EPROM on the Z-64 is to be used as a system monitor, the disk controller power on jump logic should be disabled so as not to conflict with that on the Z-64.

Micromation is not responsible for purchase, programming and burn in of the 2708 (1K) or 2716 (2K) EPROM. This is the sole responsibility of the purchaser.

I:2 Z-64 Option Installation

This section is broken down into three categories:

1) Jumper Options (I:2-1)
2) Interrupt Options (I:2-2)
3) EPROM Option (I:2-3)

The text that follows assumes familiarity with the function of each. The section containing the description of the option is included for your reference.

The Z-64 silkscreen is printed below indicating the location of each jumper and option.

I:2-1 Jumper Options

JP1 - NMI (section II:8-2): This jumper is located between columns D and E near the bottom of the board and is not installed as shipped from the factory.

Users intending to use the non-maskable interrupt feature on the Z-80 must install this jumper. External circuitry must be added to the system to generate a /NMI (low active) signal. Notice that /NMI is connected to line 12 of the S-100 bus. Other manufacturers may use this line for the XRDY2 signal (see section II:1-4). Appropriate jumpers must be installed if both these signals are necessary for system operation. (Note that the Z-80 does not normally recognize the XRDY2 signal.)

JP2, JP3 - Interrupts (section II:8-1): The source of the Z-80 /INT input is controlled by these jumpers located to the right of JP1 under columns H and J. As shipped from the factory, JP2 is installed (connecting /INT with the enable output, pin 19, of the interrupt priority encoder at AB5) and JP3 is not. JP2 and JP3 should be considered mutually exclusive. That is, the circumstances that dictate the installation of one jumper, make the other unnecessary.

With JP2 connected and the 25LS2513 installed at AB5, an interrupting device need only send its signal along one of lines VI0 - VI7 to begin an interrupt response. Either mode 0 (the 8080 interrupt response mode) or mode 1 may be used.

Pin 73 of the S-100 bus (/PINT, see section II:1-4) may also be used to initiate an interrupt. JP3 must be installed first though. This option is appropriate for systems incorporating but a single interrupting device. It is necessary that the Z-80 mode 1 interrupt response be utilized since the various mode 0 restart commands, determined from the AB5 output to the data bus, are not generated and the data bus driver at R4 is in the high impedance mode (as described in section II:3) preventing external access to the data bus.

JP4 - Clock (section II:2-4): The six holes that constitute JP4 (located just below U2) are used to configure the circuitry for operation at 2 MHz or 4 MHz. The diagram below shows the connections for each. Notice that the configuration is different for different revision boards.

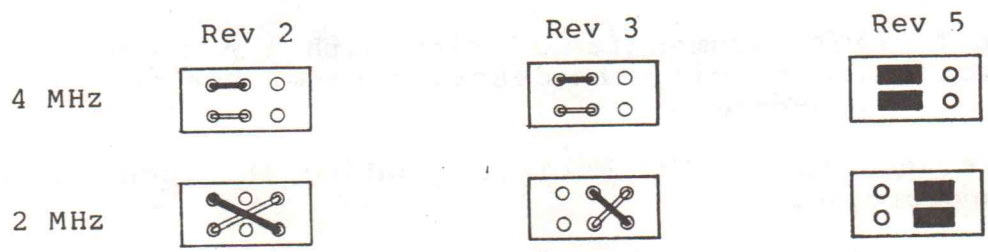


Figure I:2-1.1 JP4 Clock jumpers

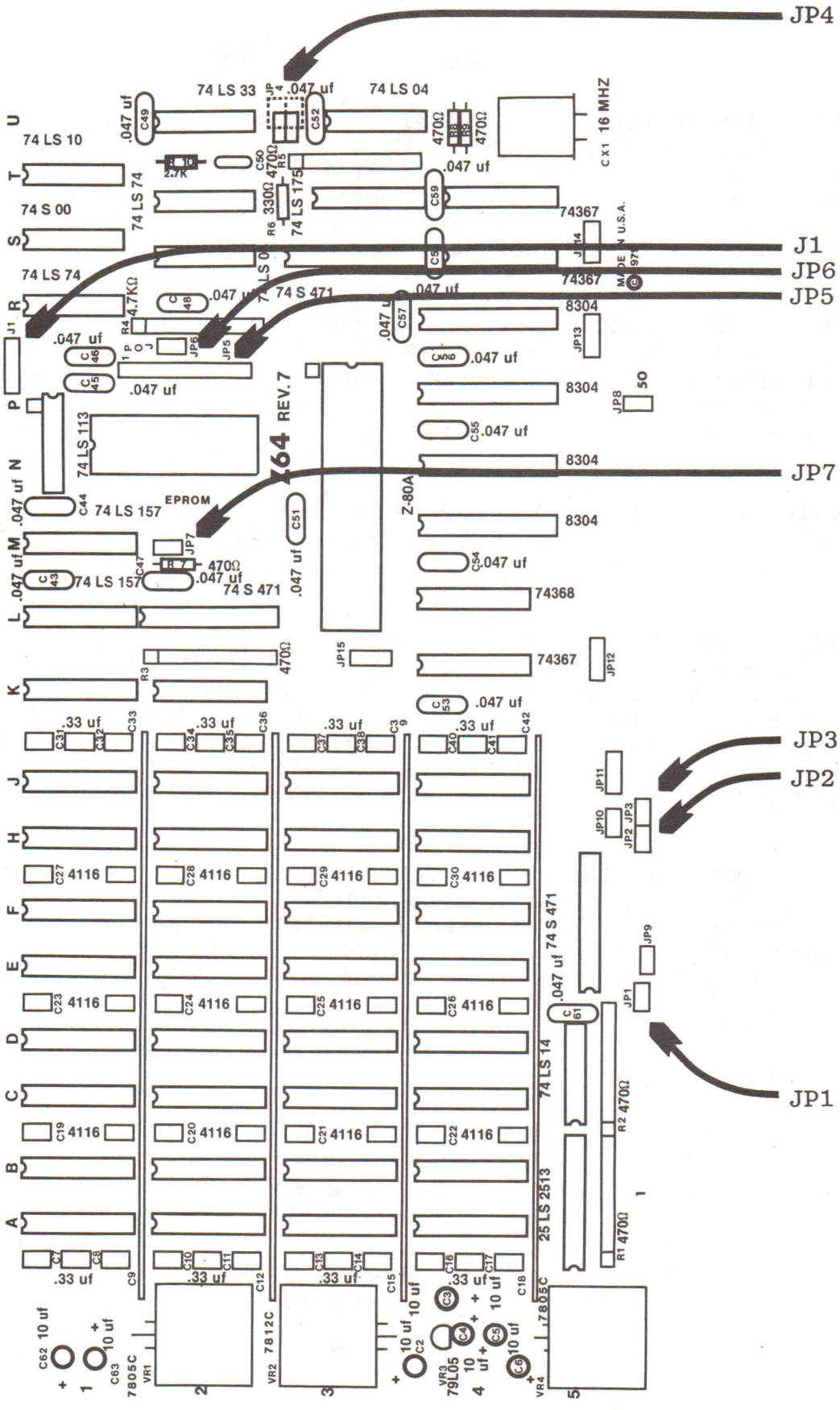


Figure I:2-1.2 Jumper locations

JP5 - EPROM (section II:2-5): Located to the right of the EPROM socket, JP5 is used to configure the circuitry for different types of memory devices (2708 or 2716). See section II:2-5 for a complete description of the jumper options. As shipped from the factory, JP5 is configured for a 2708 EPROM . Regardless of this setting, the board does not require an EPROM to operate.

JP6 - POJ (section II:6): The power on jump enabling jumper, JP6 (in row 2 column R labeled POJ) is not installed Without it, the processor proceeds from location 0000H upon reset. With it, system memory is disabled and the EPROM enabled for three machine cycles after a reset to provide a jump instruction and 2 byte address to a PROM-based system initialization routine.

NOTE: The Micromation DOUBLER has similar POJ circuitry. When the Z-64 is installed with a DOUBLER, only the latter's POJ jumper should be installed.

JP7 - EPROM (section II:2-5): This second EPROM-related jumper allows the selection of one of two starting addresses: F800H or F000H. As shipped from the factory, the jumper is installed rendering F800H as the starting address. Since this is also the standard address for the DOUBLER, the EPROM must be re addressed to accommodate both. With JP7 removed, the EPROM is addressed for F000 - F7FFH leaving F800 - FFFFH for the DOUBLER.

J1 - RST (section II:5): A four pin header is installed at J1 (just to the left of device R1) for installation of the system reset switch. The physical switch should be a push buttom, momentary contact type. (This type is used in Micromation systems.) The appropriate connection of the pins is illustrated below.

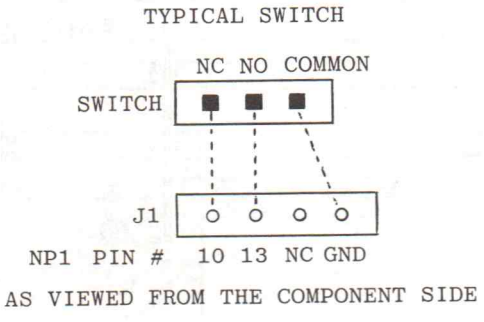


Figure I:2-1.3 Switch wiring

I:2-2 Interrupts (section II:8)

The 25LS2513 tri-state vector priority encoder is optionally availiable and may be easily installed at any time. The Z-64 contains a socket at board location AB5 to facilitate insertion. If this device was not ordered with the Z-64, a jumper is inserted via a plug installed in the socket. Subsequent installation of the device will necessitate removal of the plug allowing insertion of the 24LS2513.

To incorporate this device and utilize the Z-80 mode 0 (8080 standard interrupt response mode), order the 25LS2513 from Micromation, remove the factory installed plug, and install the device in the socket. Pin 1 must be in the lower left hand corner. When a low is received on any of the VI0 - VI7 lines, the signal is decoded and an interrupt response is enabled. The data bus driver (the 8304 at R4) is put in the high impedance mode by an active /INTA which also enables the AB5 output buffer. The unique 3-bit code is passed to data bus lines D3 - D5 (the rest are high level due to pull-up resistors). This single byte designates an 8080-type restart instruction to one of 8 memory locations.

Alternatively to 25LS2513 installation, a single interrupt device may be accommodated by direct input to the Z-80 /INT pin via pin 73 of the S-100 bus. Jumper JP3 must be connected. (As long as the 25LS2513 is **not** installed, there is no need to cut JP2.) The Z-80 must be in interrupt mode 1 for this configuration to respond appropriately. (Mode 1 causes a restart to location 0038H when /INT is active.)

The other interrupt option is to install the /NMI input to the Z-80 by connecting jumper JP3. The non-maskable interrupt is an automatic restart to memory location 0066H (see section II:8-2 and I:2-1).

I:2-3 EPROM (section II:2-5)

Installation vis-a-vis jumper selection of a 2708 1K or 2716 2K (Intel or Texas Instrument) EPROM is discussed in section II:2-5 below. To install the EPROM device after it's been programmed, insert the chip into the socket at board location NP2 with pin 1 in the upper right hand corner.

The Z-64 EPROM is configured for addresses F000H to F7FFH or F800H to FFFFH. Which location is determined by the status of jumper JP7 connecting address line A11 to the bi-polar PROM at L2. F800H - FFFFH is never available as system RAM. With jumper JP7 removed, F000H - FFFFH is not available.

When the EPROM is accessed, a wait period is automatically inserted in the machine cycle (by L2 logic) allowing the much slower EPROM to decode the address and enable its output buffer.

The EPROM is provided for installation of a system monitor for a special application package. Additionally, power on jump circuitry can be used to force the CPU to read the first three locations of the EPROM at every reset or power-up. A jump instruction and 16-bit address should be contained in these locations for the start of a system initialization routine.

Disk-based computer systems don't necessarily need a system monitor; the disk resident operating system performs similar functions. This does not preclude the use of a system monitor but note that disk interface firmware and the Z-64 EPROM cannot occupy the same memory space.

I:3 TROUBLESHOOTING

All Z-64s are comprehensively tested before shipment with a DOUBLER to ensure that all logic subsystems are fully functional.

There are two levels of troubleshooting in the event the system does not boot at power-up or reset.

1) Checks to ensure the Z-64 is configured properly for the system.

2) Probing with a voltmeter, logic probe, and/or oscilloscope to ensure the circuitry is functioning as it should

The first level can be performed by just about anyone. The second requires someone with technical skills and the proper equipment. Regarding the second level: the results of tests should be written down and sent with the board in the event it needs repair. This will aid the technician in debugging and expedite return of the board. Users who choose to perform repair themselves run the risk of voiding the warranty if further damage is caused.

The discussion below is broken down into two parts dependent upon whether the computer loads the operating system or not. If it doesn't boot, review the literature on your disk controller as well as the Z-64 to ensure all compatibility requirements are satisfied. If the machine does boot but does not run properly, refer to section I:3-2.

I:3-1 Computer System Won't Boot

Level 1:

If the Z-64 fails to boot (load the operating system from disk and display the sign-on message and prompt) first power-down the computer. Remove all the printed circuit boards and clean the S-100 edge connectors (both sides) with a pencil eraser only.

While the boards are out, check the jumpers to ensure everything is configured properly. For systems with Z-64 and DOUBLER, pay particular attention to the following.

Z-64 Clock jumpers (JP7) installed properly (see section I:2-1 for configurations). Ensure that the contacts are good and solid. (They may have come loose in shipment.)

All other jumpers should be installed or cut according to the application. Refer to section I:2-1 for a summary of the jumper options.

DOUBLER Ensure that the POJ, XRDY (or PRDY), and PHANTOM jumpers are installed. (Refer to the DOUBLER manual for the locations.)

For 4 MHz operation the WAIT jumper should be installed. For 2 MHz the WAIT jumper should NOT be installed.

Finally, make sure the cables between the DOUBLER and the disk drives and terminal are installed correctly.

Reinstall the boards, power-up, and try again. If the system fails to boot, make notes of the computer's actions (e.g., the lights on the drives, the sounds eminating from the drive during head loads and track changing) and call your dealer for assistance. See section I:3-3 below for the service procedure.

Level 2

This level should only be attempted after level 1 has been performed.

BE AWARE

The presence of this discussion is **not** to be construed as authorization for users to perform repair on the board. If your Z-64 is not functioning as it should call your dealer for assistance (see section I:3-3). In the worst case, you will be told to return the board for repair. If you have some technical experience and the proper equipment (voltmeter, logic probe, and/or oscilloscope), you may perform the tests below and others to isolate the problem. However, if you attempt repair it will be at the risk of voiding the warranty. Any problems identified should be carefully noted and sent with the board. This will help us identify the malfunctioning component and expedite return

****

To set up the Z-64 for these tests, turn off power, remove the Z-64, insert an extender card into a card slot and insert the Z-64 into the extender card. Turn-on the system power and the board is ready to test. There's no need to have a disk in drive A:.

- Use a voltmeter or 'scope and check the 3 voltage levels on the board. The easiest place to check power supply is the inputs to the RAM chips. Check the following pins of one chip in each row.

+5V on pin 9*
-5V on pin 1
+12V on pin 8

* There are two +5V voltage regulators. Each supplies 2 rows of RAM chips (1 and 2 or 3 and 4).

- Check the clock input to the Z-80 with an oscilloscope (input pin 6) to ensure the clock generator is functioning. A 2 MHz or 4 MHz square wave should appear on the screen. It won't be perfectly square (transitions and levels will have glitches), but it should not be a single level.

- Check the /M1 output (pin 27) to see if the Z-80 is operating. Refer to the signal illustration in section II:1-2 for comparison. If your 'scope can do it, reference /M1 to the clock.

- Check the /INT input to the Z-80 (pin 16). It should have a high level. If your Z-64 has the 25LS2513 and this line is low, yank this chip (not the Z-80) and check again.

- Check the /WAIT input to the Z-80 (pin 24). This line will

appear active at times but it is inactive at the falling edge of the Z-80 clock cycle T2. (You will need to have a 'scope that can display the clock and /WAIT signals together.)

- These tests check for aberrant input signals that prevent the Z-80 from performing in its normal operating mode. If these check out all right, turn-off power, remove the Z-64, and carefully exchange the RAM chips in row 1 with those in row 2. Be sure that all pins are inserted in the sockets. If a RAM problem was preventing O/S loading, this procedure will move the bad chip above the area where the system enters. Re-install the Z-64 and try booting again. If successful, perform the RAM test described in section I:3-2 below to locate the bad chip.

- If the system boots but the prompt is not displayed, remove the board and carefully exchange the RAM chips in rows 3 and 4 (again, check all the pins). The operating system moves from the first row (low memory) to the last (high memory) after it has been loaded completely from disk. Absence of the prompt indicates a bad RAM cell in the high memory. Try booting again, and if successful, perform the RAM test below.

Make careful notes of the results of these tests and call your dealer (see section I:3-3) for assistance.

I:3-2 System Boots But Something Is Wrong

The following test can be performed by anyone. It requires no technical skills. If errors occur, call your dealer for assistance.

If your system boots but all is not well otherwise, chances are there's a memory related problem. This could involve the multiplex circuitry, refresh circuitry or the RAM array itself. Micromation has a program, RAMTEST.COM, supplied with it's systems to check these areas. If you purchased the Z-64 as part of a Micromation system, you have RAMTEST.COM on the distribution disk. If the Z-64 was purchased to upgrade another system, the RAMTEST listing is given in Appendix D at the end of this manual. Use the editor and assembler provided with the O/S to create this program.

Generally, RAMTEST checks each byte of memory during every pass. This is done completely independently of the disk interface so it is a check of memory only. (A disk access is performed occasionally but it only serves to provide a time delay so the refresh circuitry is tested.) The program causes the processor to fill each byte of memory with a different value as determined by an algotrithm. Once memory has been filled, the program goes back and generates the same value for comparison against that in

the memory byte. If there's a difference, the value in the processor (what the RAM byte should be) and the value in memory (what is stored) are displayed. By comparing the two, the bad chip can be identified.

This test runs very fast. A single error will be displayed on the screen and remain. If an entire chip is bad, 16,384 errors will result and they will be displayed quickly, too quickly to be analyzed. To halt execution to view the results hit ^S (press the control key and the S key at the same time). To restart program execution, hit ^S again. To exit the program hit ^C. (You must have a system disk in drive A: to reboot.) A single pass tests all of memory and takes about 15 minutes. Longer tests may be run to check for intermittent errors or failure due to heat build-up or other cause.

To run RAMTEST, load the operating system and execute RAMTEST. If CP/M (a registered trademark of Digital Research) is your operating system, type 'RAMTEST'

The screen will display instructions. To proceed, hit the RETURN key.

A pass number, in hex, and pattern number, also in hex, are displayed. These display program status (show that it's running and how far it's gone) and do not by themselves indicate an error. If the value written disagrees with the value read, an error message appears:

COMPARE ERROR Location:xxxx Hex, Pattern:xx Hex, Memory Data:xx Hex

There are three sorces of memory problems:

1) faulty RAM chip(s)
2) faulty multiplexer(s)
3) faulty refresh circuitry

1) **A faulty RAM chip** can generate 1 to 16,384 errors. The bad chip can be identified by comparing the PATTERN value (indicating the value written) with the MEMORY DATA value (indicating the value stored in RAM.) Notice that these values are in hex. The table below demonstrates hex to binary conversion. Notice that each digit of a hex number represents 4 binary digits. Thus, the 2 digit hex number represents one byte.

Hex	Binary	Hex	Binary	Hex	Binary	Hex	Binary
0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	A	1010	B	1011
C	1100	D	1101	E	1110	F	1111

Before converting the hex numbers to binary use the

first digit of the **Location** hex value to determine which row the bad chip is in. Table I:3-2.1 equates the number to the row.

Table I:3-2.1
RAM Row Determinator
(from Location Hex Value)

First Hex Digit	RAM Row
0 - 3	1
4 - 7	2
8 - B	3
C - F	4

If the first digit is a 2, the bad chip is in row 1, a 5 indicates row 3, a D points to row 4, etc.

Next, convert the two sets of hex data to their binary form to determine which column the bad chip is in. (Note: each byte of data is stored across an entire row of chips, one bit per chip.)

For instance, if the PATTERN value is AC and the MEMORY data is 8C, the following comparison would determine the bad bit (and chip).

	hex	binary
PATTERN	AC	1010 1100
MEMORY DATA	8C	1000 1100

This would indicate that the third chip in from the right (column F on the board) was bad. (The Location value indicates which row.) On the Z-64, the MSB (most significant bit) is stored in the chip in column J and the LSB (least significant bit) is in column A. Notice that the column letters do not indicate hex numbers.

A bad RAM chip will usually generate more than one error. By analyzing the error messages a distinct pattern will develop pointing to one (or more) chip(s). If no pattern appears, refer to the other two sources below.

2) **A faulty multiplexer** generates error messages but there is no pattern pointing to a distinct chip. The multiplexers switch the RAM chip address inputs from row to column addresses. This must happen at a precise moment in a machine cycle or the address will not be valid.

To determine if one of the multiplexers (there are

two) is bad, compare the PATTERN and MEMORY DATA values as demonstrated above. If no RAM chips are consistently pointed to by the results, one or both of the 74LS157s at L1 or M1 is bad.

3) If no bit cell pattern is apparent **AND** the same MEMORY DATA value is repeated (not necessarily sequentially but does appear throughout a block of RAM), chances are the **refresh circuitry** is not performing its task as it should. It's difficult to pinpoint the source of the problem. The board will need to be returned for repair.

I:3-3 Return Procedure

In the event your board needs to be returned to the factory for repair, follow the procedure below.

1) Make careful notes of the symptoms. The better the notes, the more likely a simple problem can be identified and a fix prescribed.

2) Call the dealer from whom you bought the unit and describe the symptoms. Many dealers have technicians familiar with Micromation products. If a quick remedy is possible, the technician will be able to relay it to you. The technician will also be able to determine if the board requires return to the factory for repair.

3) If the technician or dealer determines the board must be returned to the factory for repair, send the unit to the dealer; do not send it to Micromation. This will allow the dealer to visually inspect and perhaps run some tests on the board. It is in the best interest of all parties to intercept problems and repair them as close to the user as possible. The bottom line of this process is faster turn around.

4) If the board needs to be returned to the factory, the dealer will let you know and send the board. You will be informed by the dealer when the board is repaired and what if any charges will be applied. The board will be returned to you by Micromation.

SECTION II

Section II is written for system builders, programmers, technicians and the curious. The conceptual approach is first to define the basic component then second to state how it is utilized. In basic component descriptions all features are presented. Some of these features may not be used, however; their function replaced by other components or not used at all.

Each subsection describes a different aspect of the Z-64. Section II:1, for instance, describes the Z-80. Section II:2 describes RAM addressing, refresh, the EPROM, and the phantom line. Each of these sections includes that portion of the schematic referenced by the text. A complete schematic is included at the end of this manual.

II:1. Z-80

The Z-80 has proved to be a powerful processor for microcomputer use. In addition to the instruction set of the 8080, it supports many additional commands (total 158) for data manipulation. Those familiar with the 8080 internal registers, will recognize the A (accumulator), F (flags), B, C, D, E, H and L registers. New to them is an identical alternate register set. A single command exchanges A,F with A',F'; another command exchanges the work registers. Of course, the Z-80 includes a 16 bit stack pointer and program counter registers. Special purpose registers include two index registers to hold 16-bit base addresses for indexed addressing modes (IX and IY), a vector interrupt register (I) to store the high order byte of an address for a direct call in response to an interrupt, and a memory refresh register (R) from which dynamic RAM addresses are provided for refresh.

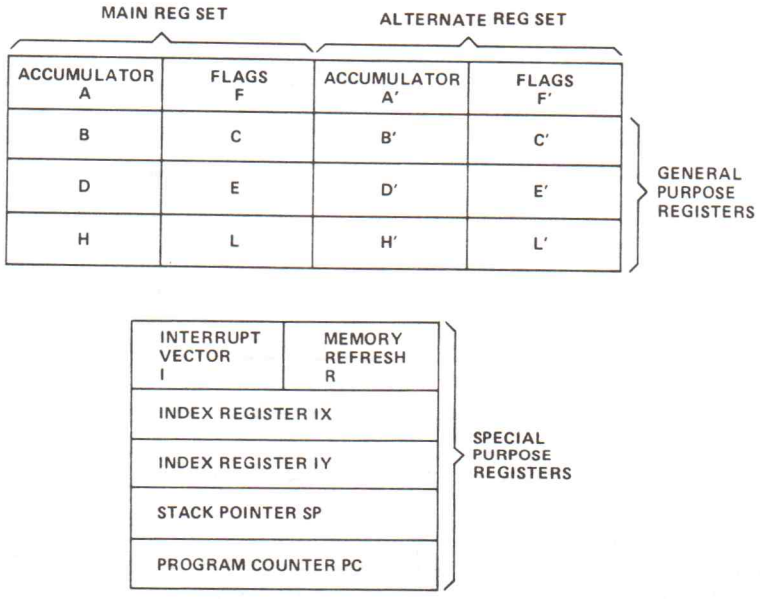


Figure II:1.1 Z-80 Register Set

This section is not intended as a programming guide for Z-80 users. There are many manuals available for this purpose. The focus instead, is upon the Z-80 signals and how they interact on the Z-64 to form a single card computer.

II:1-1 Pin Out and Signal Definition

The Z-80 is contained in a 40 pin dual in-line package. Each pin has a distinct function associated with it. These functions can be broken down into three types:

1) Output Signals
2) Input Signals
3) Bidirectional Signals

The figure below illustrates the relationship between the pins and signals.

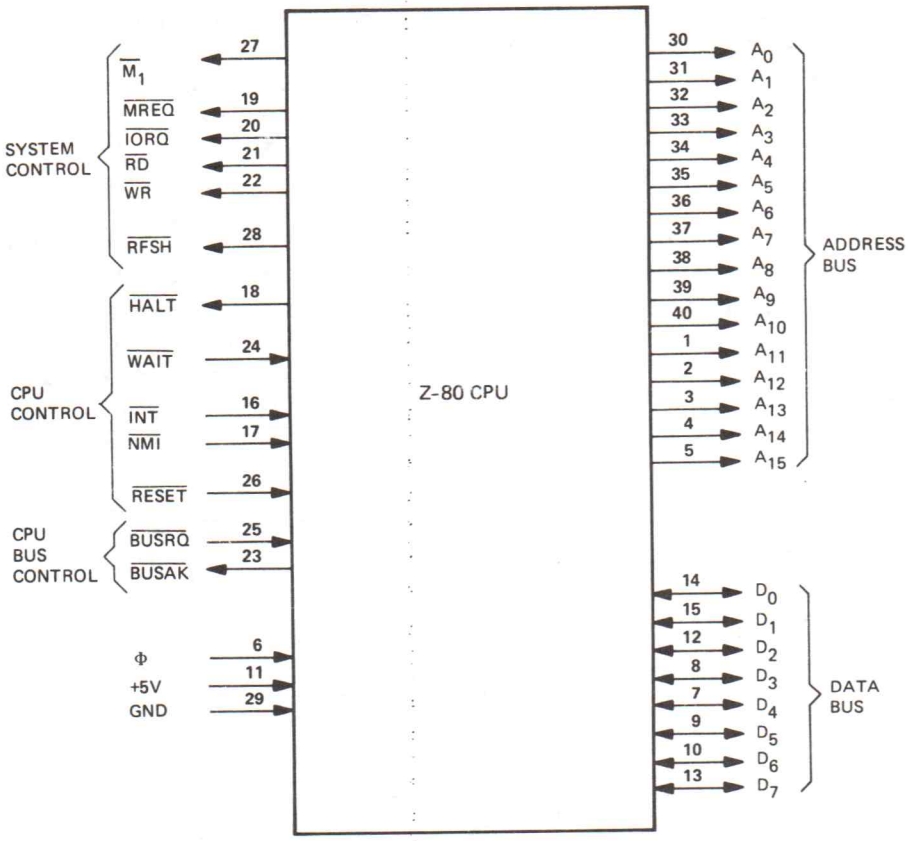


Figure II:1-1.1 Z-80 Pin Outs

Output Signals: This group of signals is generated by the Z-80 to coordinate system activities. The signal name, pin number and definition follows.

Name	Pin(s)	Definition
A0 - A15	30 - 40 1 - 5	Tri-state output (high, low, high impedance), active high, A0 - A15 constitute the 16-bit address bus. Thus, a full 64K of system memory can be accessed. In addition, the 8 low order bits (A0 - A7) are used to directly select up to 256 input or output ports.
/M1	27	Active low (hence the /), /M1 indicates that the current machine cycle is an instruction OP code fetch. /M1 is also used to distinguish the time required to fetch the op-code from the time required to decode and execute it.
/MREQ	19	Tri-state output, active low, /MREQ (memory request) indicates that a valid address is on the address bus for a memory read or write operation.
/IORQ	20	Tri-state output, active low, /IORQ (input/output request) indicates that A0 - A7 contains a valid address for an I/O read or write. With /M1, /IORQ is also used to acknowledge an interrupt. When both are low, an interrupt response vector can be placed on the data bus. Interrupt acknowledge only occurs with M1 (i.e., both must be low). I/O operations do not use an M1 cycle.
/RD	21	/RD (read) is also a tri-state signal, active low. It indicates that the CPU is expecting data from memory or an I/O device. /RD is available to cue memory select or I/O device for data reads.
/WR	22	Again a tri-state signal, active low. /WR (write) indicates to memory or an I/O device that the CPU holds valid data on the data bus for output.
/RFSH	28	Active low, /RFSH (refresh) goes active after the fetch portion of an instruction OP code fetch cycle (i.e., as /M1 goes high) and returns high at the end of the machine cycle. Thus is implemented dynamic RAM refresh. The memory address to be refreshed is taken from the R register. Seven bits of this byte are automatically incremented after each machine cycle. The eigth bit is not used.

Name	Pin(s)	Definition
/HALT	18	Active low, /HALT indicates that the processor has received a software halt instruction and is waiting for a maskable or non maskable interrupt. No processing is performed while /HALT is true (low). NOPs (no operation) are performed to refresh memory while waiting for the interrupt.
/BUSAK	23	Active low, /BUSAK (bus acknowledge) is sent to a device that has requested control of the address bus, data bus and tri-state control bus signals (/MREQ, /IORQ, /RD, /WR). When /BUSAK goes low, these buses are set in the high impedance state allowing the device to take control (e.g., DMA device).

The CPU receives **Input Signals** from other devices in the computer system. Generally, they request the CPU's immediate attention.

Name	Pin(s)	Definition
/WAIT	24	Active low, /WAIT indicates to the CPU that the addressed memory or I/O device is not ready to send data. CPU wait states continue until the device states it is ready (/WAIT returns high). Note that refresh will not be performed during wait states. Extended wait states (longer than 2 milliseconds) will destroy the contents of dynamic memories.
/INT	16	Active low, /INT (interrupt request) is sent to the Z-80 by an I/O device. The request is honored at the end of the current machine cycle. The Z-80 contains an interrupt flip-flop that is set under program control (instruction EI enables, DI disables). The CPU's response to /INT is determined by the status of this flip-flop.
/NMI	17	Negative edge triggered, /NMI (non maskable interrupt) is available for important interruptions of CPU processes (e.g., impending power failure). /NMI is not overridden by the interrupt enable/disable flip-flop. When /NMI has been triggered, the processor will not respond until the end of the current machine cycle. At this time, the contents of the program counter is transferred to the external stack and the CPU automatically jumps to location 0066H (hex) for the service routine.

Name	Pin(s)	Definition
/RST	26	Active low, /RST (reset) forces the program counter to zero and initializes the CPU. This sets the I and R registers to zero, disables the interrupt flip-flop, and returns the Z-80 output signals to their inactive state (high). During reset the address and data buses go to high impedance.
/BUSRQ	25	When /BUSRQ goes low (active) the Z-80 sets the address, data and control output buses in the high impedance state and generates /BUSAK Thus, the CPU allows other devices to take over control of these buses. The CPU waits until the end of the current machine cycle before acknowledging the bus request.
Ø	6	This is the clock signal sent to the Z-80 from an external clock generator. It sets the time periods for machine cycles.

Bidirectional Signals

D0 - D7	7 - 10 12 - 15	Data is received and sent through these pins. This is the data bus. (See the diagram above for the pin to signal assignments.)

II:1-2 CPU TIMING

Each signal is generated, sampled, or interpreted at a specific time. A reference signal, clock, sets the basic operating speed. Machine cycles (M) consist of groups of clock cycles (T). The number of clock cycles in a machine cycle is dependent upon the function of the cycle. For instance, an instruction OP code fetch cycle contains 4 - 6 T (clock) cycles. Subsequent machine cycles to move data between the CPU and memory or a I/O device may last 3 - 5 T cycles. Wait states sent to the CPU by memory or the device will lengthen this period. An instruction cycle is the sum of the machine cycles required to perform a specific instruction (the OP code fetch, decode and execution.) Figure II:1-2.1 illustrates the components of an instruction cycle.

NOTE: Not all of the machine cycles described below are pertinent to the Z-64. For instance, the discussion on Bus Request/Acknowledge Cycle (used in DMA operations) does not apply. These sections are presented for your information and are not meant to indicate that the Z-64 will support these features.

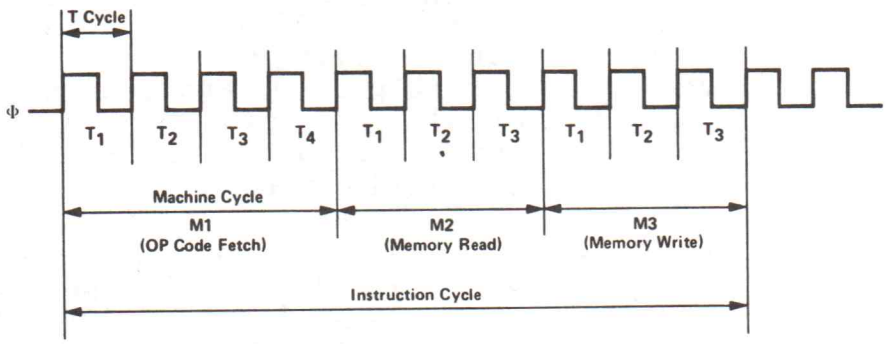


Figure II:1-2.1 Basic CPU Timing

In this instruction cycle, the OP code fetched during M1 called for a memory read followed by a memory write. These are but two types of machine cycles (OP code fetch and memory read/write cycles). The Z-80 has seven basic machine cycles. Each instruction cycle is initiated by a M1 cycle - an OP code fetch. The nature of this command (memory request, I/O request, etc.) determines which of the machine cycles is to follow.

Instruction OP Code Fetch Cycle (M1): Each instruction cycle begins with a M1 cycle. During this time the instruction OP code for the next instruction (pointed to by the Program Counter) is fetched from memory (2 clock cycles) and decoded (2 - 4 clock cycles).

An M1 cycle is identified by /M1 active (low) then /MREQ and /RD active (also low) one half clock period later. The contents of the program counter are transferred to the address bus during T1 and T2. Active /RD and /MREQ indicate that the data is to be output from the memory device to the data bus. Slow memory devices (EPROMs for instance) typically can't supply the data fast enough for the processor. Consequently, the Z-80 checks for an active WAIT line at the falling edge of T2. If /WAIT is active (low), wait states are inserted into the machine cycle and the current signal status (i.e., /MREQ, /RD, and /M1 active) is maintained. The CPU continues to sample the WAIT input at the falling edge of every Tw cycle and inserting wait states until the line is found inactive. The data is sampled from the data bus on the rising edge of clock cycle T3. /MREQ, /RD, and /M1 return inactive soon thereafter.

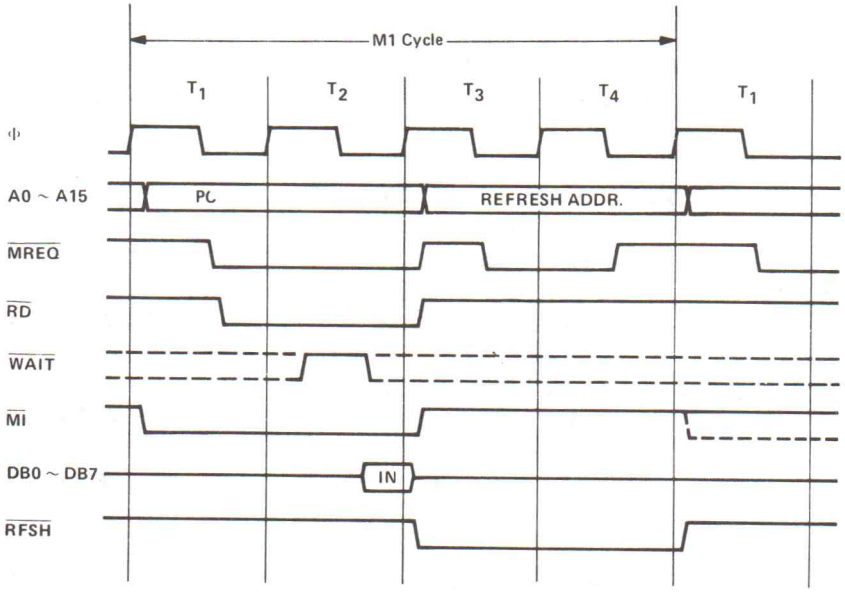


Figure II:1-2.2 MI Cycle

The second part of an M1 cycle is used to decode the instruction and, since the CPU is preoccupied with this task, refresh memory. /RFSH goes low (active) as /M1 goes high (inactive). The contents of the CPU R (refresh) register is loaded onto the address lines A0 - A6. (The value in these seven bits is incremented after each machine cycle so that all memory is refreshed. The reason for a 7-bit address and not 8 is discussed in section II:2.) One half clock cycle later, /MREQ goes low again to enable the dynamic memories. The refresh process is performed by reading a row (but not columns - see section II:2 - 3) of RAM. Since the memory is being refreshed, not read, the /RD line is held high to prevent output of the data to the data bus.

Since refresh occurs during the time the CPU is pre-occupied with instruction decode, it does not affect program execution time and is totally transparent to the user.

Memory Read or Write Cycle: If a memory read or write is called for by the instruction OP code, the following timing sequence results. Again, the address and data buses are used. /WAIT plays a passive role unless slow memories are involved. /MREQ enables memory for the read or write. /RD or /WR indicate to memory the nature of the transaction. (They can't both be active at the same time.) These machine cycles are generally 3 T cycles long unless wait states are inserted.

The process begins with the memory address going onto the address bus. For a memory read, /MREQ and /RD perform as they did in the M1 cyle above, but there's no active /M1 signal, and the data bus isn't sampled until the falling edge of the T3 cycle. Again /WAIT is sampled on the falling edge of T2. If active, (memory indicating the data isn't ready), the CPU holds /MREQ and /RD active until /WAIT is found inactive when sampled again at the falling edge of Tw cycles. After the data bus has been sampled, /MREQ and /RD go high, and the cycle has completed.

Memory write is very similar. Again, 3 clock cycles are required unless wait states occur. /MREQ follows the same pattern as in the previous examples. The /WR signal goes active one clock cycle earlier (upon the falling edge of T2) than the /RD signal, and the data is present on the data bus longer.

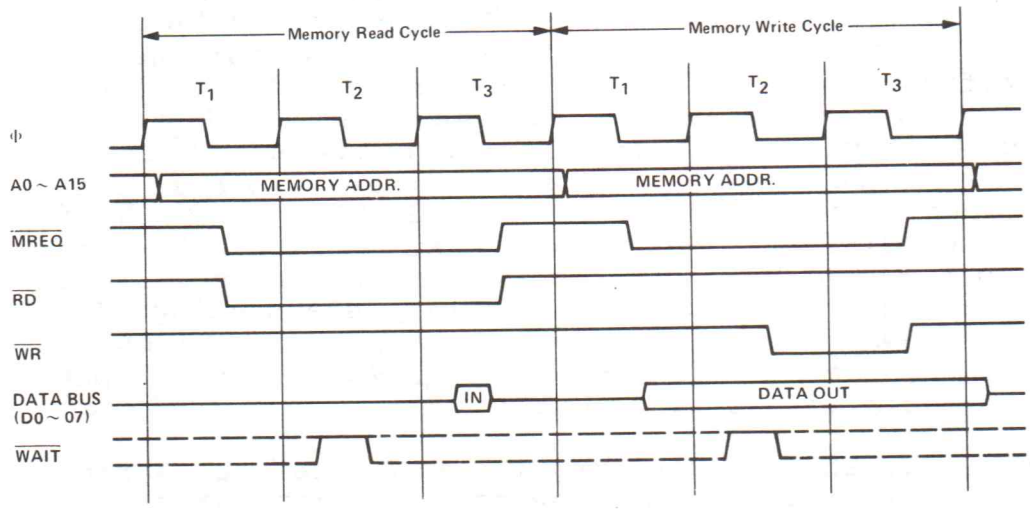


Figure II:1-2.3 Memory Read or Write Cycle

Input or Output Cycles: While similar to the previous memory R/W discussion, there is a crucial difference when the data is going to or coming from a peripheral device: two wait cycles (Tw) are automatically inserted between T2 and T3. This allows the device addressed to decode the address and activate the /WAIT signal if necessary. (The time period between /IORQ going active and the normal T2 /WAIT sampling point being too short for this function.) If the /WAIT signal is activated, additional wait cycles will be inserted and the /IORQ, /RD or /WR, and D0 - D7 (in the event of a write) lines held in their active states. /RD is used to enable the addressed port (as indicated by address lines A0 - A7) onto the data bus as in the case of a memory read. The bus is sampled at the falling edge of T3. For a write, /WR is used to clock the port. The data is maintained on the data bus so that the rising edge of /WR can be used as a data clock.

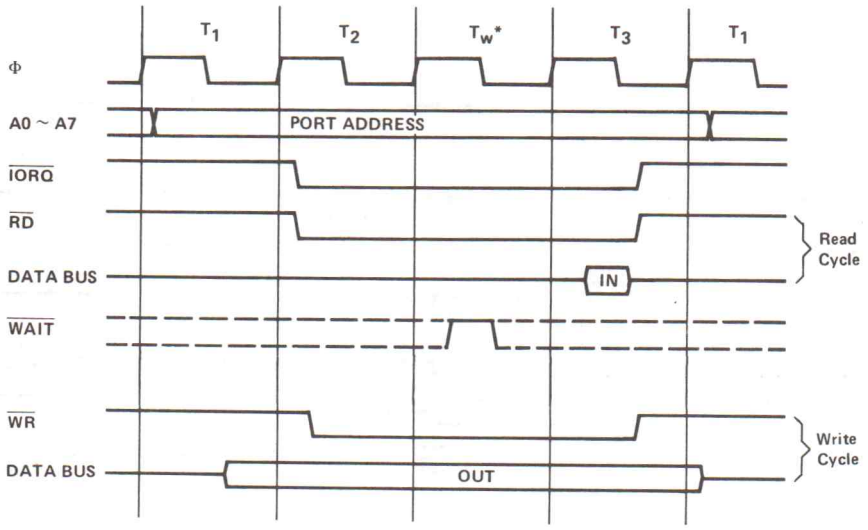


Figure II:1-2.4 I/O Request Cycle

Bus Request/Acknowledge Cycle: Computer systems that rely upon DMA (Direct Memory Access) for I/O device to RAM data transfers use /BUSRQ (bus request) and /BUSAK (bus acknowledge) to wrest control of the address, data and tri-state control buses from the CPU. (The Z-64 does not support DMA at this time. When installed with the Micromation DOUBLER, the operating system uses byte oriented transfer of data from disk, through the CPU, and on to RAM. The following Z-80 signal timing is presented for your information.)

The CPU samples the /BUSRQ line at the leading edge of the last clock cycle of each machine cycle. At the leading edge of the next clock cycle, the CPU sends /BUSAK low and sets the address, data, and tri-state control signals in the high impedance state. Control of these signals remains external to the CPU until /BUSRQ goes high indicating the external device is done. Control is returned (/BUSAK goes high) within one clock cycle from this discovery and program execution continues. During a bus request cycle, neither /NMI nor /INT will interrupt the DMA processor or be recognized by the CPU if they return to inactive before it has reacquired control.

An important consideration during DMA transfers is memory refresh. Since the tri-state control signals /RD, /MREQ and /RFSH are set in the high impedance state, the memory will not be refreshed by the Z-80. Although memory refresh can be performed by some DMA devices, this is not supported on the Z-64.

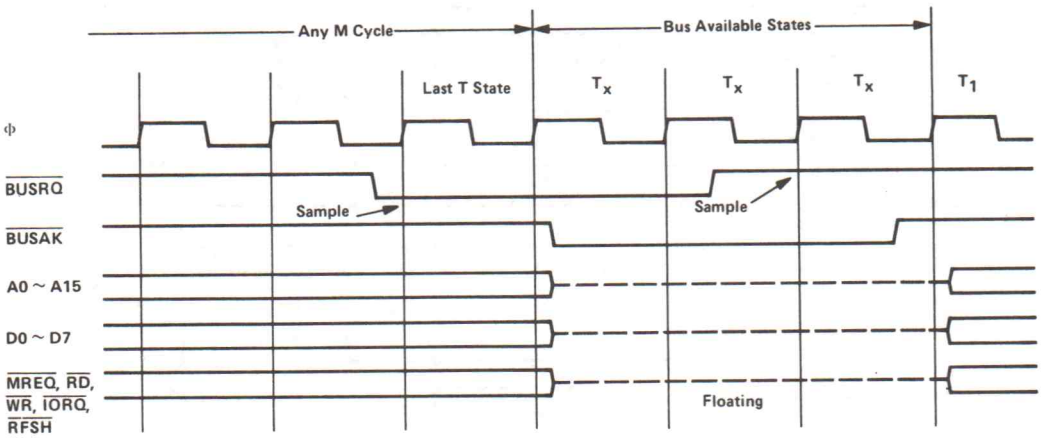


Figure II:1-2.5 Bus Request/Acknowledge Cycle

Interrupt Request/Acknowledge Cycle: The figure below indicates CPU timing in response to an interrupt request. As with /BUSRQ, the CPU samples the /INT line at the leading edge of the last clock cycle during every machine cycle. The interrupt will not be acted upon unless /BUSRQ and /NMI are inactive (high) and the interrupt enable flip-flop is set. If these conditions are met, a special M1 cycle is generated by the CPU. This M1 cycle differs from the OP code fetch cycle in that /IORQ is used with /M1, instead of /MREQ, and /RD never goes active.

Two wait states (Tw below) are automatically inserted between T2 and T3 to provide sufficient time for a ripple priority interrupt scheme to be implemented. The /WAIT input is sampled (at the falling edge of the second Tw) to determine if enough time has been allocated.

There are three modes of interrupt response. Mode 0 is the standard 8080-type interrupt acknowledge in which the contents of the data bus, with inputs from the priority interrupt encoder, are read indicating a restart instruction and address. The Mode 1 acknowledge disregards the contents of the data bus and performs a restart from location 0038H. Mode 2 is a special Z-80 response mode and is not supported by the Z-64. Refer to section II:8 for a complete discussion of Z-64 interrupt acknowledge capabilities.

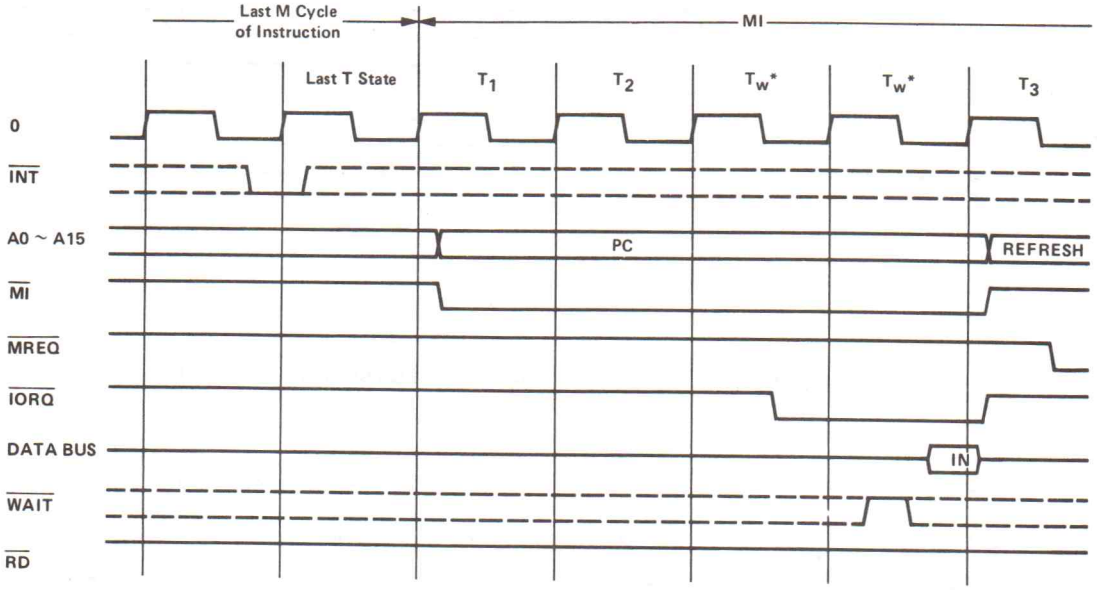


Figure II-1-2.6 Interrupt Request/Acknowledge Cycle

Non Maskable Interrupt Request/Acknowledge Cycle: As indicated earlier, the CPU samples the /NMI line at the rising edge of the last clock cycle of every machine cycle. An active /NMI (low) overrides /INT but takes a back seat to an active /BUSRQ. It is not subject to the interrupt enable flip-flop.

Active /NMI generates a special M1 cycle. During this machine cycle, the contents of the program counter is pushed to the external stack (location indicated by the SP register) and a jump to location 0066H is performed. Refresh is performed with the rising edge of /M1 as in other M1 cyles.

Exit from HALT Instruction: The /HALT signal is set under program control. Once set (low), the CPU performs NOPs (no operation) until a low /NMI or /INT is sensed. Each NOP is an M1 cycle that lasts 4 clock cycles and performs no operation but does perform memory refresh. /NMI and /INT are sampled at the leading edge of T4. If one or both are active (in which case /NMI has priority), the CPU responds as indicated in the previous /NMI or /INT timing diagrams.

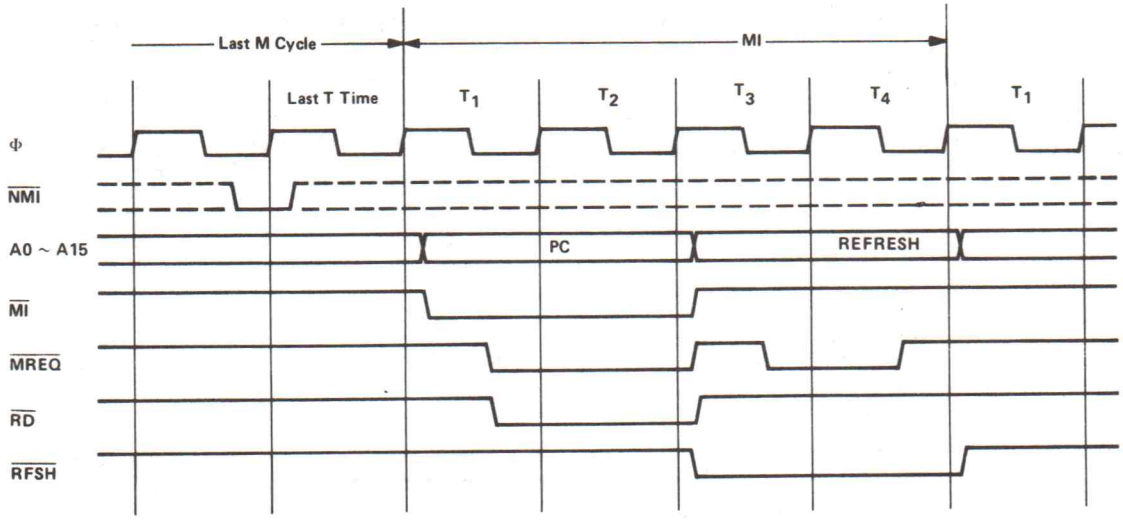


Figure II:1-2.7 NMI Request/Acknowledge Cycle

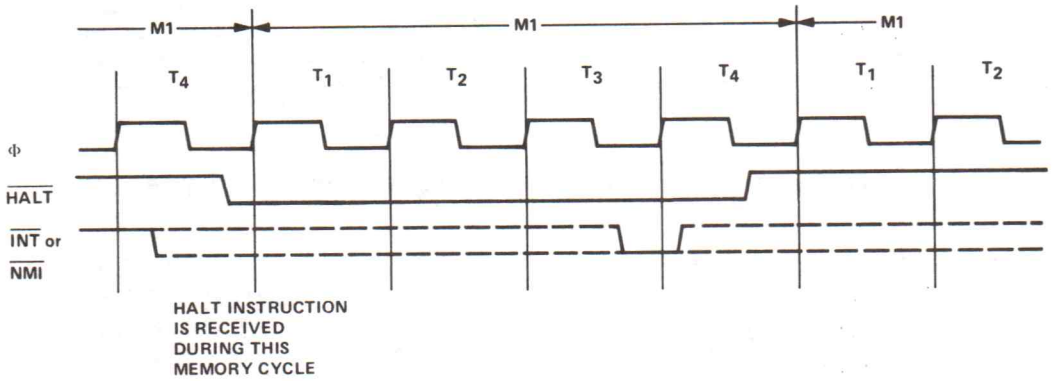


Figure II:1-2.8 Exit from HALT

The timing diagrams of section II:1-2 should be used as a reference throughout Section II. The interrelationship between the signals demonstrated above are the basis for the processes described below and will be referred to in the text.

II:1-3 THE Z-80 ON THE Z-64

The Z-80 description above applied to the Z-80 in general, not necessarily to its application on the Z-64. Some signals used in the timing illustrations are not used and others take their place. For instance, the /RD (memory read) signal is inverted to enable the memory for a write operation. /WR is not used for this purpose at all. (The reason for this is described in section II:2-2 below.) Other differences will be pointed out in the following sections.

The diagram on the following page is excerpted from the Z-64 schematic. It designates the Z-80 signals vis-a-vis their utilization. The signals are batched according to function. The function and the corresponding section below are identified.

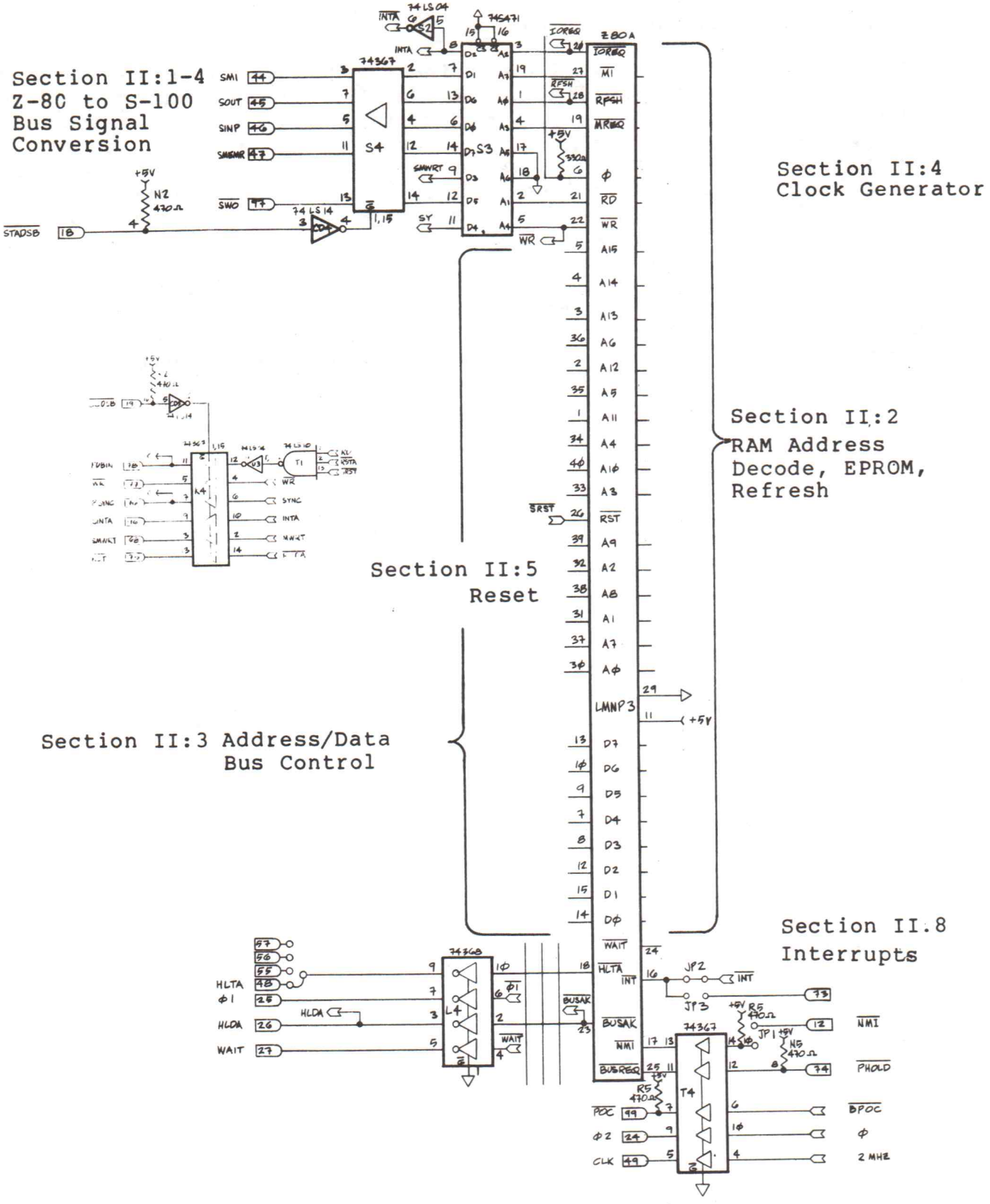


Figure II:1-3.1 The Z-80 on the Z-64

II:1-4 Z-80 to S-100 Bus Signal Conversion

The S-100 bus was the early de facto standard upon which microcomputer systems were built. CPU signals (based upon the 8080 processor) were given a dedicated line on the bus for communication between printed circuit boards. Other signal lines were allocated anticipating expansion, control and I/O requirements. Appendix B contains a description of the S-100 bus signals and the trace assignments. The following section expects familiarity with them.

Some signals on the S-100 bus match directly with Z-80 signals (/RST for instance). Others are the product of two or more CPU signals. These combinations are typically used for reference by external devices. For instance, a printer must be informed that it is selected and told what character to print. (You don't want the printer selected all the time.) In this case the Z-80 signals /IOREQ and /WR are activated and yield SOUT to the bus. This prompts the printer to go active and print the character. An input device (a terminal) would look for SINP before it yielded its data to the bus.

There are several signals that are totally independent of the CPU. These are generated by external devices for control of the data and address buses and the buffers that pass status and control signals.

Many S-100 signals are not essential to Z-64 operation. They are present to satisfy industry convention and to ensure compatibility with external (8080-based) devices.

The S-100 bus signals can be broken down into five classes:

1) Input directly to the CPU
2) Output directly from the CPU
3) Product of two or more CPU output signals
4) Product of two or more signals that become a single CPU input
5) Status signals that never reach the CPU and are used for signal control

The table on the following page references the S-100 signals used on the Z-64 according to the classes above. Included in the descriptions are Z-64 components that contribute to their generation. The data bus (D0 - D7), address bus (A0 - A15), clock signals and vector interrupt lines (VI0 - VI7) are not included. The S-100 pin number is shown in parentheses.

TABLE II:1-4.1

Z-64 S-100 Bus Signals by Class

1) Input Direct to CPU

NMI (12) (special case)
PINT (73)
PHOLD (74)
PRESET (75)

2) Output Direct from CPU

HLDA (26)
SM1 (44)
HLTA (48)
/PWR (77)

3) Output Combinations

WAIT (27)
SOUT (45)
SINP (46)
SMEMR (47)
/SWO (97)
SMWRT (68)
PSYNC (76)
PDBIN (78)
SINTA (96)

4) Input Combinations

XRDY (3)
PRDY (72)

5) CPU-Independent Control Signals

/PHANTOM (16 or 67)
/STADSB (18)
/CCDSB (19)
/ADDSB (22)
/DODSB (23)
/POC (99)

1) Input Directly to CPU

In the next two groups of signals the Z-64 relative signal is shown in parentheses underneath the S-100 signal.

/NMI (12)
(XRDY2)

This signal is redefined for the Z-64. With an 8080 it can be used along with XRDY (pin 3) to indicate to the processor that the indicated I/O device is ready to send/receive data. On the Z-64 pin 12 is /NMI (low active) and causes a non maskable interrupt. This signal is jumper selectable and not installed when shipped from the factory. The hex bus driver at T4 buffers this line.

/PINT (73)
(/INT)

An external device sends this signal to the processor to request an interrupt. This CPU command line is superceded by the interrupt priority encoder at AB5. However, a jumper can be installed at JP3 to utilize this signal with the mode 1 interrupt acknowledge. See section II:8 for a complete discussion.

/PHOLD (74)
(/BUSREQ)

A low appears on this CPU command line when an external device requests control of the address, data and processor control signals. This is the signal activated for DMA functions. It is buffered at T4.

/PRESET (75)
(/RST)

Although not connected directly between the bus and processor, the intervening circuitry is functionally inconsequential. This is the processor command reset signal (see section II:5).

2) Output Direct from CPU

HLDA (26)
(/BUSAK)

Upon receipt of a /PHOLD (/BUSREQ), the Z-80 acknowledges the request when it's ready (afterthe current machine cycle) and releases control to the requesting device. On the Z-80, this control signal is active low. However, the S-100 equivalent is active high. Consequently, the bus driver at L4 inverts /BUSREQ.

SM1 (44)
(/M1)

This S-100 status signal is the inverse of the Z-80 /M1 signal indicating that the processor is in an OP code fetch portion of an M1 cycle. It is inverted as a function of the bi-polar PROM at S3 (see section II:1-4.1 below for a discussion of this PROM).

HLTA (48)
(/HLTA)

This is a CPU status output that acknowledges a software HALT instruction. Notice that it is low active from the Z-80 but high active on the bus. Consequently, it is inverted at L4.

/PWR (77)
(/WR)

/PWR is a processor control output that indicates a write is to take place. External boards can use this as the data bus buffer enable. Notice that the output of this signal to the S-100 bus is controlled by the status of /CCDSB at K4.

3) Output Combinations

WAIT (27)

On the Z-80, /WAIT is a low active input. On the S-100 bus, WAIT is a CPU control, high active output indicating the processor is in a wait state (i.e., it acts as a wait acknowledge). On the Z-64 several inputs are NANDed for input to the Z-80. This input is then inverted at L4 for output to the bus. The circuit diagram below illustrates the input and output lines.

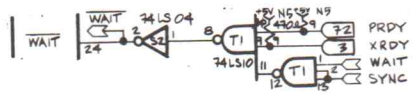


Figure II:1-4 WAIT Signal Generation

Three of the signals are S-100 related: PRDY, XRDY and SYNC. The fourth, WAIT, is input to T1 from the PROM at L2 described in section II:2-5 below.

Notice that a low on PRDY or XRDY (indicating an external device is not ready to send or receive data as requested) causes a wait state in the processor and an active WAIT to the S-100 bus. If both these signals are high (inactive) and L2 sends an active WAIT when SYNC is active, the processor will also enter a wait state and S-100 WAIT will be active. (This second case happens only when addresses F800H - FFFFH, the location of the EPROM, are accessed and the phantom line is not enabled.)

The following four signals are all a product of the bi-polar PROM at S3. Refer to section II:1-4.1 below for a description of this PROM. Their output to the S-100 bus is controlled by the status of S-100 signal /STADSB.

SOUT (45) — This processor status signal is active whenever /IOREQ and /WR are active together. It indicates that the address bus contains the 8 bit output device address. SOUT is active high.

SINP (46) — The input equivalent of the SOUT processor status signal, SINP is active (high) whenever /IOREQ and /RD are active together. It indicates the address bus has an input device's 8-bit address.

SMEMR (47) — The status of /MREQ and /RD determine SMEMR. This CPU status signal is active (high) when both of these Z-80 signals are active. It indicates that the address bus contains a 16-bit memory location to be read.

/SWO (97) — A processor status signal, /SWO indicates a write to memory or output device operation. Notice that, unlike the three above, it is low active.

Like the four signals described above, the following four are controlled by the status of a signal. In this case, it's the /CCDSB signal. Active (low) /CCDSB disables the bus driver at K4 (puts it in the high impedance state).

SMWRT (68) — This signal is also a product of S3. SMWRT is a processor status signal that is active (high) whenever /MREQ and /WR are active together. It indicates that a memory write (A0 - A15 contain the address) is to take place.

PSYNC (76) — PSYNC is a CPU control signal. It is active (high) for one clock cycle and straddles CPU clock states T1 and T2 of every machine cycle. On the Z-64 it is a product first of the S3 PROM output SY and second of the 74LS175 at board location T3. (This location is also used for the RESET function; see section II:2-5 below.)

SY is active (high) from S3 under certain conditions (see section II:1-4.1 below) and begets SYNC through T3. Notice that the clock for T2 is the same signal as the system clock except offset one half clock period (via inversion at U3). Thus SYNC goes high half way through machine clock cycle T1 then returns low in the middle of the next clock cycle (T2).

PDBIN (78) Inverted /RD signal NANDed with /RSTA and /SRST (to prevent spurious reads during reset) is used for PDBIN. This processor control signal indicates to external devices that the data bus is in the input mode.

SINTA (96) INTA is the resident Z-64 processor status signal which acknowledges an interrupt It is output from S3 whenever /M1 and /IOREQ are active together (which only occurs during the special Z-80 M1 interrupt cycle). SINTA is the S-100 bus equivalent gated by K4.

4) Input Combinations

Both of these inputs can cause extended wait states which could prove to be the demise of the information stored in memory. Recall that refresh does not occur during a wait period. Some peripheral devices and disk controllers may cause extended wait states. Refer to the disk controller manual to determine if this could be a problem. (The Micromation DOUBLER does not generate extended wait states.)

XRDY (3) When the XRDY input is high, the sending device is stating, "I'm ready to send/receive data." When it's not ready, the device pulls this line low. Notice that XRDY is one of the inputs that contributes to the Z-80 /WAIT input and the S-100 WAIT signal (discussed above). Whenever it goes low at the right time (the /WAIT input is only sampled by the Z-80 on the falling edge of clock cycle T2), a wait state is inserted in the machine cycle.

PRDY (72) This is the processor command/control input that controls the run state of the CPU. This line is sometimes connected to the RUN/STOP switch on systems with a front panel. (This switch should be avoided since it inserts an extended - longer than 2 milliseconds - wait state.) This signal also is a contributor to the Z-80 /WAIT and S-100 WAIT signals. When PRDY is low, a wait state is inserted.

5) CPU Independent Control Signals

This group of S-100 signals operate independently of the CPU. That is, they are generated by external devices. The function of the first group below (/STADSB, /DODSB, /ADDSB, and /CCDSB) is to control the gating of the CPU status signals, data bus buffer, address bus buffer, and CPU control signals, respectively, onto the S-100 bus. Most often, they are used to disable these bus drivers effectively eliminating the influence of the CPU in

deference to an external DMA controller. (DMA is not presently supported by the Z-64. The PROM at EF5 is not programmed to recognize their active state.)

/STADSB (18) Status signal disable is used to control the output of the CPU status signals SM1, SOUT, SINP, SMEMR, and /SWO. It is active low. Notice that the enable input to S4 is low active. Consequently, /STADSB is inverted before input to the bus driver.

/CCDSB (19) The control signal disable controls the output of the CPU command/control signals to the S-100 bus. This too is low active and hence inverted to accommodate the hex bus driver at K4.

/ADDSB (22) An active (low) address bus disable will do just that: disable the address bus drivers. Notice that this input is to the PROM at EF5 and would disable the 8304 bi-directional bus drivers at P4 and M4 if the PROM was encoded to recognize it.

/DODSB (23) Similar to /ADDSB, the data bus disable performs the same function for the data bus. Also input to EF5, /DODSB would disable the 8304 bidirectional bus driver at R4.

Two other signals are necessary for system operation.

/PHANTOM (16, 67) The phantom line is used to disable system memory in favor of firmware occupying the same address space on an external device. Notice that two lines on the S-100 bus are used, depending upon the manufacturer: 16 or 67. The Z-64 uses line 67. If your system uses line 16, a jumper must be installed. (Line 16 is not recognized by the Z-64.)

/POC (99) Power on clear performs a similar function to RST. Notice that it is a product of the R-C pair input to T2. This combination holds the /BPOC signal low (active) for a period of time allowing CPU and external device set-up.

These two signals, unlike /STADSB, /ADDSB, /CCDSB, and /DODSB, are implemented on the Z-64.

1-4.1 S3 Bi-Polar PROM: This PROM generates the S-100 bus signals from Z-80 signals. The S-100 bus was originally designed for use with the 8080 CPU. The Z-80 simplifies many operations but must generate 8080 sympathetic signals to maintain compatibility with its predecessor.

The table below indicates the S-100 signals and the Z-80 signals that activate them. These combinations indicate the only time the output is active. In any other combination, the given S-100 signal is inactive.

S-100	Z-80
SINTA	/M1, /IOREQ
SM1	/M1, /MREQ, /RD
SOUT	/IOREQ, /WR
SINP	/IOREQ, /RD
SMEMR	/M1, /MREQ, /RD /MREQ, /RD
SMWRT	/MREQ, /WR
/SWO	/IOREQ, /WR
PSYNC (SY)	/M1 nothing active

II:2 RAM

The Z-64 contains 64K of fast (200 nanosecond access time) dynamic RAM. Contained in a 16-pin dual in-line package, the memory chips are arranged in 4 rows of 8 chips each on the board. A byte of data is stored across an entire row, one bit per chip. Each row contains 16K (16,384) addresses for data storage. The figure below designates the pin assignments.

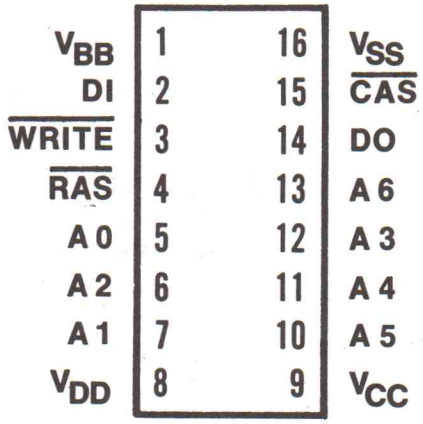


Figure II:2.1 RAM Chip Pin Out

A0 - A6: Address Inputs
DI: Data In
DO: Data Out
VBB: Power (-5V)
VDD: Power (+12V)

/CAS: Column Addr. Strobe
/RAS: Row Address Strobe
/WRITE: Read/Write Input
VCC: Power (+5V)
VSS: Ground

II:2-1 RAM Cell Addressing

Each RAM chip contains a 128 rows of 128 bits. (128 is 2 to the seventh power.) A bit is uniquely identified by indicating which row it's in then which column. The row and column identification is delivered to the chip on input lines A0 - A6. The separate 7-bit codes are gated into the chip address decoders by the /RAS (row address strobe) and /CAS (column address strobe) inputs (pins 4 and 15 respectively). The CPU address lines A0 - A6 and A7 - A13 are multiplexed onto these 7 RAM address lines. When /RAS is active (low), the 7 bits on A0 - A6 identify the row. Subsequently when /CAS goes active (also low), the 7 bits identify the column. CPU lines A0 - A6 can be thought of as the row address and A7 - A13 as the column address. CPU address lines A14 and A15 are used to select one of the 4 banks* of RAM chips. (The absence of the /RAS active signal on the 3 banks not selected prevents the input/output of data upon application of an active /CAS.)

* Bank in this context indicates a row of RAM chips as distinguished from a row within a RAM chip.

Pin 3 on the RAM chip, write enable, determines whether the data bit is to be read from or written to memory. When active (low), data is taken from the data bus and stored. A high input sends the data out for a CPU read.

The timing that separates the row address from the column address on RAM lines A0 - A6 is crucial. Recall from the previous discussion on the instruction OP code fetch (a memory read) that the location in memory is selected and read within two clock cycles. This means that the RAM must read and decode the row address, read and decode the column address and have the data bit ready within this period. The figure below illustrates the timing sequence.

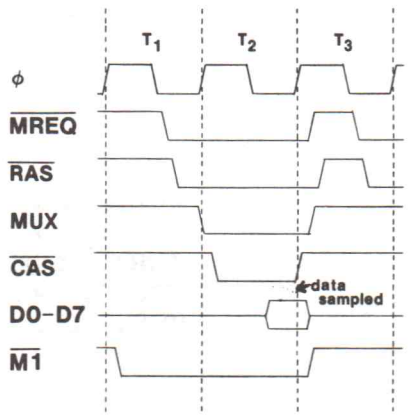


Figure II:2-1.1 Data Access Timing During MI (worst case)

As can be seen, the change from row address (A0 - A6) to column address (A7 - A13) is performed at the rising edge of the T2 clock cycle.

Within each RAM chip, the bit selection process is initiated by an active /RAS (low). The row address is decoded and each bit of the selected row (all 128) are gated to sense amplifiers that discriminate, latch and rewrite the logic level of each cell. Subsequent /CAS application selects the specific column and renders the bit to the output buffer. Activation of the entire row means that refresh can be performed by applying /RAS and a row address without /CAS. (Recall that the Z-80 R [refresh] register contains a row address. This is gated onto address lines A0 - A6 during clock cycles T3 and T4.) A complete description of the refresh process is presented in section II:2-3 below.

II:2-2 Address Decode

Z-80 and memory interaction involves the address bus (A0 - A15), the data bus (D0 - D7) and 5 control signals (/M1, /RFSH, /MREQ, clock and /RD). /WR, memory write, is not used on the Z-64. Instead, /RD is inverted.

Address lines A0 - A13 are multiplexed in the 74LS157s at board locations L1 and M1 (see Figure II:2-2.1 below) to render the 7 input lines for the RAM. Notice that the address inputs are paired, A0/A7, A1/A8 . . . A6/A13. Thus, the low order 7 bits (corresponding to the row address) are passed to the RAM chip, then the high order 7 bits (corresponding to the column address) are passed. A high on the select line (pin 1) selects the B inputs (A0 - A6); a low selects the A inputs (A7 - A13). The select signal to the multiplexers must make the high to low transition at the rising edge of clock cycle T2.

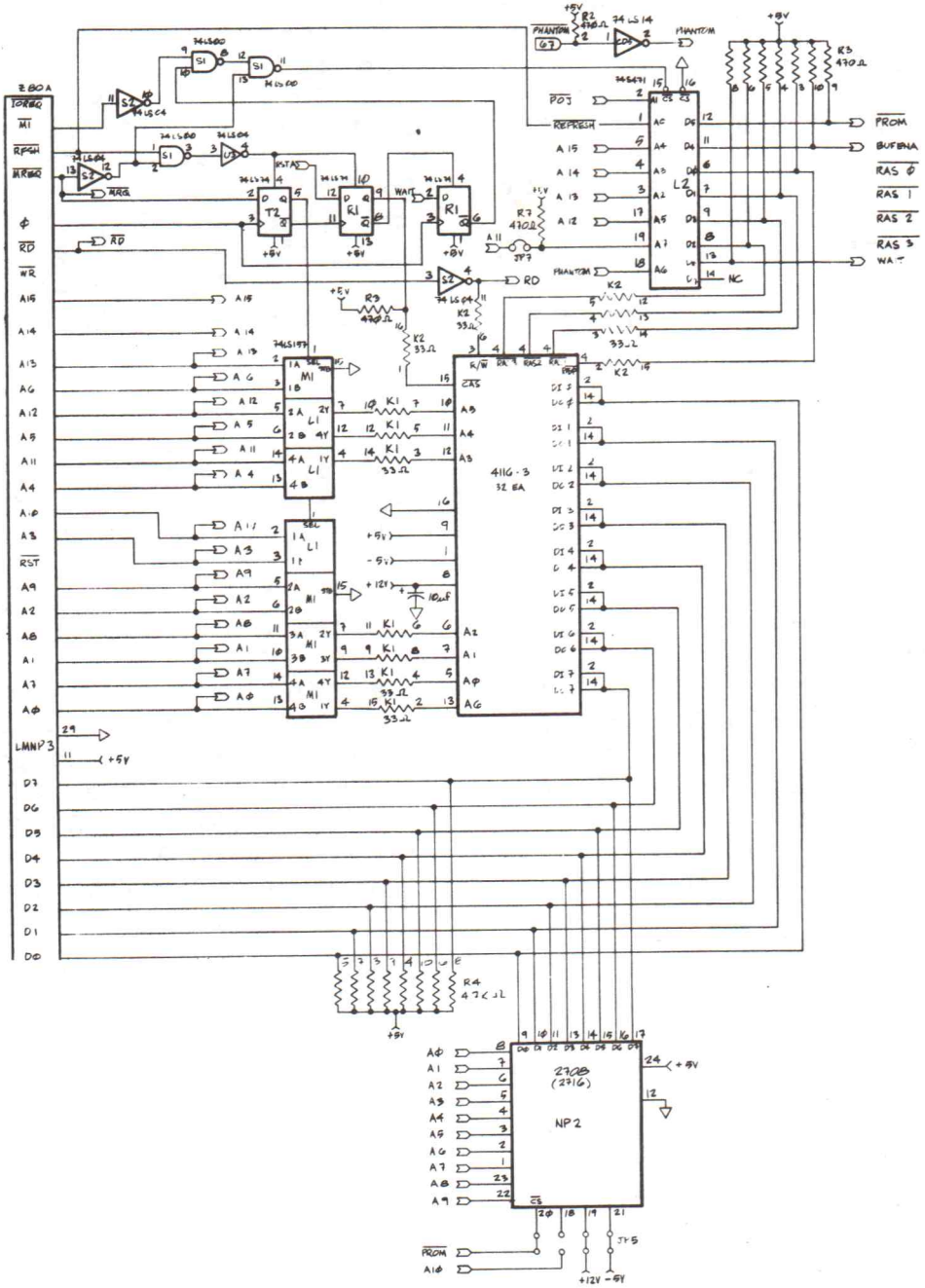


Figure II:2-2.1 RAM and Address Decode Circuitry

/MREQ, /RFSH and clock are used to change this select line. /MREQ is inverted at S2 (yielding MREQ) and NANDed with /RFSH at S1. This output is then inverted in another section of S1 to provide the PRESET input (pin 4) to the 74LS74 at board location T2. Other inputs to T2 are /MREQ and clock. The CLEAR input (pin 1) is held high. The timing diagram below illustrates the process for changing the Q output for the address multiplexers. Again, an M1 instruction OP code fetch (worst case) is used as reference.

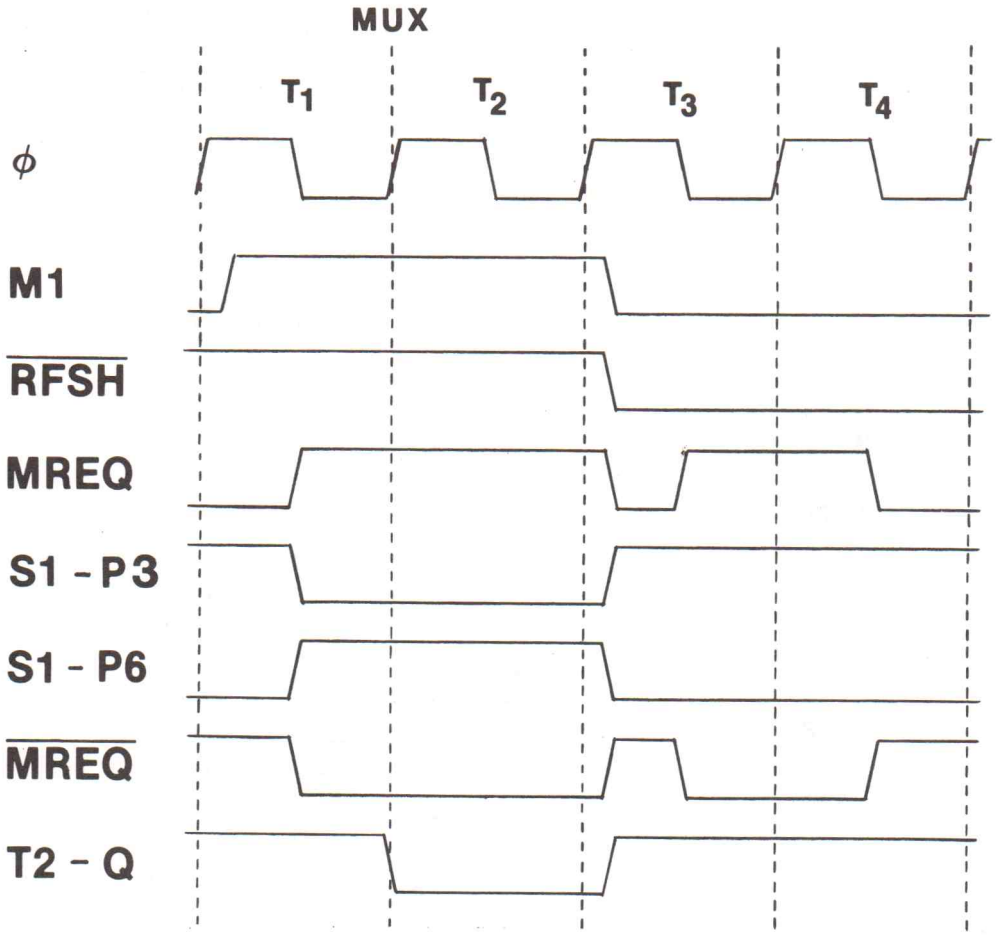


Figure II:2-2.2 Select Input to Multiplexers

NOTES: 1) Clock input to T2.
2) D input to T2.
3) PRESET input to T2, output of S1.

The 74LS74 logic is edge sensitive to the clock signal. In other words, the Q and /Q status only change with the rising edge of the clock. In this case, the rising edge of clock cycle T2 of M1.

Before the change is made to the MUX select pin, the /RAS (one of the four /RAS lines 0 - 3) is active (low) indicating the address to RAM pins A0 - A6 is the row address. At the change of Q from high to low, the /CAS line must be activated to indicate that the column address is now on the lines.

The 74LS74 at R1 is used to generate /CAS. The D input is subject to the RSTA signal from the reset circuitry. This signal is normally low during program execution. The exception is when the reset button is pressed. At this time, RSTA is high which ensures that the Q output stays high preventing spurious writes to memory during reset. CLEAR (pin 13) is again held high. PRESET input is the same as the 74LS74 at location T2. R1 is clocked by the /Q output from T2. /Q supplies the rising edge to R1 to activate the logic.

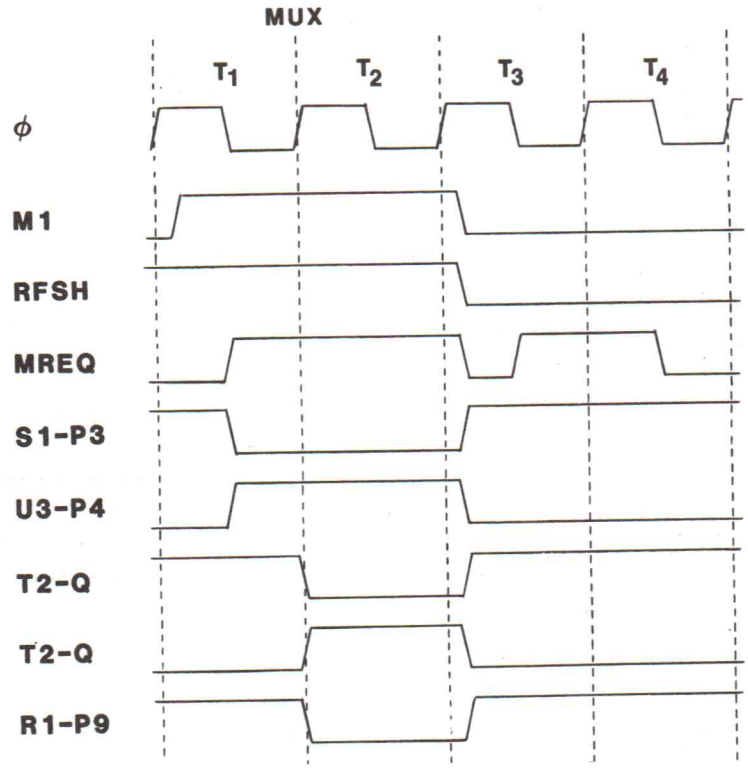


Figure II:2-2.3 /CAS Enable

Close inspection of T2 and R1 reveals that both Q outputs do the same thing (i.e., go low at the rising edge of clock cycle T2). In an ideal state this would indeed occur. Processing the signals, however, causes a time delay between the low to high change of the T2 /Q output and the resultant high to low change of the R1 Q output which allows the new address (A7 - A13) to stabilize after the multiplex switch.

/RAS signal generation involves more logic than /CAS. One reason for this is that the /RAS signal is used to enable a specific bank of RAM chips, /RAS0 the first bank, /RAS1 the second, etc. CPU address lines A14 and A15 are used to select the appropriate bank. Other considerations that enter in /RAS generation are refresh (/RFSH), the phantom line (for disabling memory in deference to external memory devices), and the power on jump circuitry (/POJ). These inputs are fed to the 74S471 bi-polar PROM at L2. Outputs from the PROM, besides the aforementioned /RAS 0 - 3, provide PROM select (to enable an optional 2708 or 2716 at NP2), a wait signal (WAIT) and a buffer enable for the data bus (BUFENA).

Among the functions performed at L2 is /RAS enable. For the purpose of this discussion disregard the EPROM at NP2; only the four rows of memory chips will be referenced. (EPROM is discussed in section II:2-5 below.) The pertinent inputs are /L2EN (chip select, active low), A14 and A15. The relevant outputs are /RAS0, /RAS1, /RAS2, and /RAS3.

A RAM chip is selected via the /RAS enable. If /CAS without /RAS is applied to the chip, no output will result. On the Z-64, /RAS(0-3) serves two functions: bank select as well as row address gating.

In Figure II:2-2.4, notice that the input to L2 pin 15 is active low. (The other /CS is held low.) The signal /L2EN is similar to /MREQ. Thus, the chip select is active for both memory request and refresh. When active first, address lines A0 - A6 (the row address) are gated from the multiplexers to RAM. During the second active state, the contents of the refresh register are passed to the chips. This is the sequence in an OP code fetch (M1) cycle, the worst case.

/MREQ isn't used as the chip select (/RAS enable) because of timing considerations relative to the RAM chips. /MREQ is inactive for only 100 nanoseconds between the OP code fetch and refresh. This isn't enough time to satisfy RAM precharge requirements. This period is extended slightly with the WAIT signal.

The row of chips is selected by a low on one of lines /RAS(0 - 3). The inputs from which this decision is made are address lines A14 and A15. A high on neither, one or the other, or both is decoded and the appropriate /RASn selected. (All 16 address lines have now been accounted for. A0 - A6 provide the row address, A7 - A13 provide the column address, and A14 -A15 determine /RASn.)

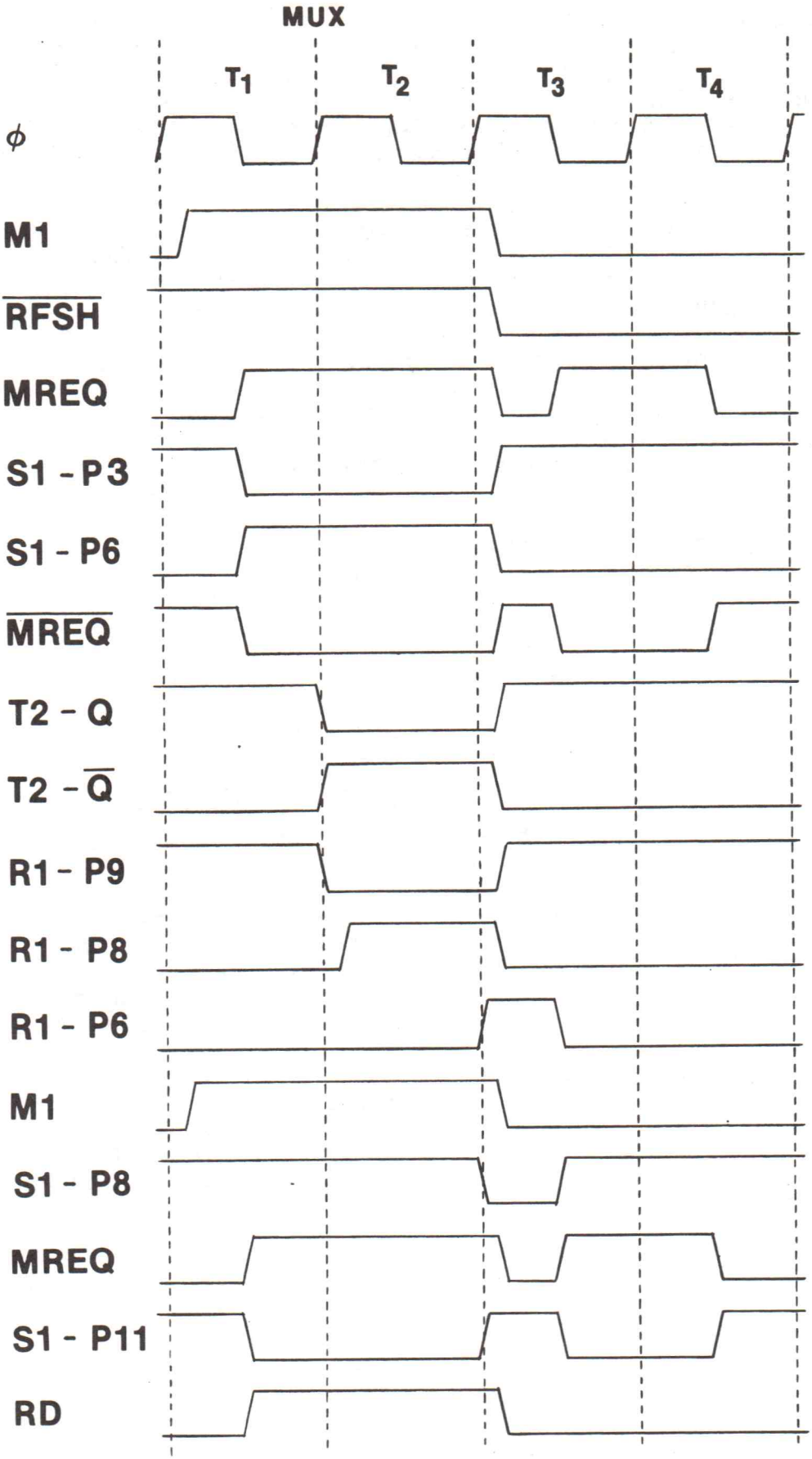


Figure II:2-2.4 /RAS Enable

II:2-3 Memory Read/Write

The discussion thus far has focused on address selection. The RAM array must also be told whether to perform a read or write. The read/write enable (RAM chip pin 3) is low active. This means that a low level at a certain point indicates that the data on D0 - D7 should be written into memory. A high level indicates the data is to be yielded to the data bus from the output buffer. There are several options available to circuit board designers to implement this function.

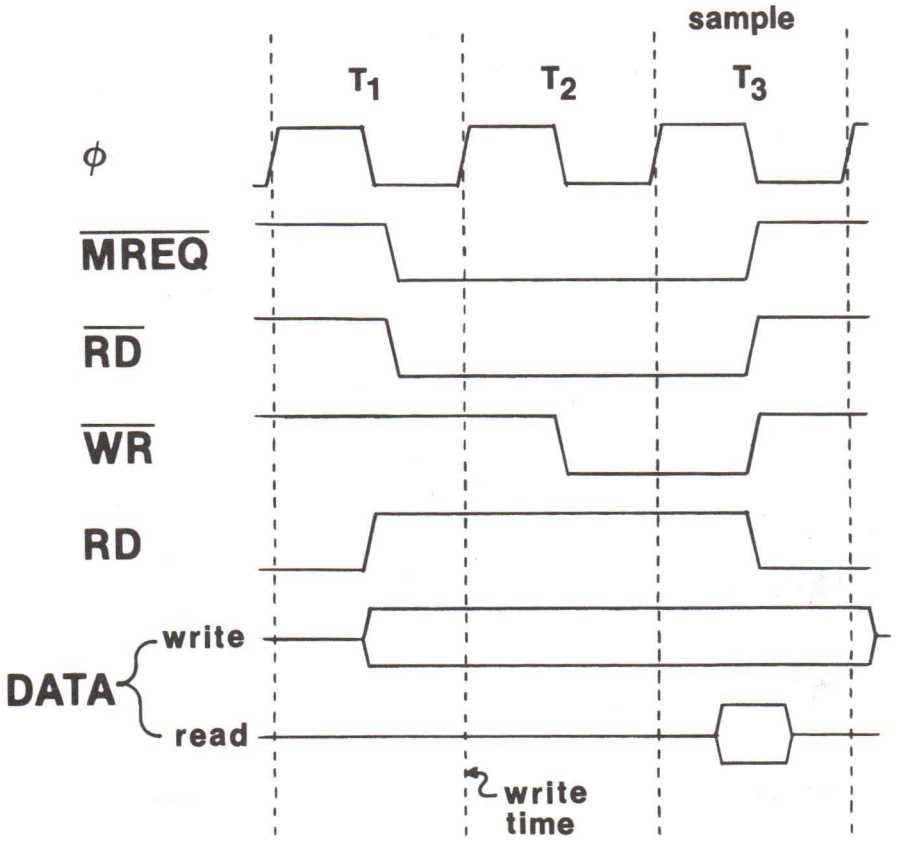


Figure II:2-3.1 CPU Read or Write Cycle

The Z-64 uses the /RD signal inverted to indicate read or write. The upshot of this is that during write cycles the data is strobed into RAM upon the falling edge of /CAS (the rising edge of T2) since the write enable input (RD) is low during the entire memory write machine cycle. Use of /RD allows the inputs and outputs to be tied together (RAM pins 2 and 14) but requires that the data be ready on the bus earlier (the Z-80 handles this).

II:2-4 REFRESH

Dynamic RAM requires refreshing. Any given row within the memory array must be refreshed (if it hasn't been accessed) at least once every 2 milliseconds. (Some chips can go longer without refresh, but it isn't good policy to extend this period.) Although only one row of memory (128 bits per memory cell) in the selected row of chips is refreshed per OP code fetch, more than enough M1 cycles occur during 2 milliseconds to refresh all memory at either 4 MHz or 2 MHz operation.

The row to be refreshed is indicated by 7 bits in the Z-80 R (refresh) register. The register contains 8 bits but the high order bit isn't used. After every OP code fetch, the indicated row is refreshed and the R register incremented.

The sequence of events during refresh is illustrated below. Note that these are the Z-80 generated signals. Their application on the Z-64 is discussed below. The illustration picks up the M1 cycle during clock cycle T2.

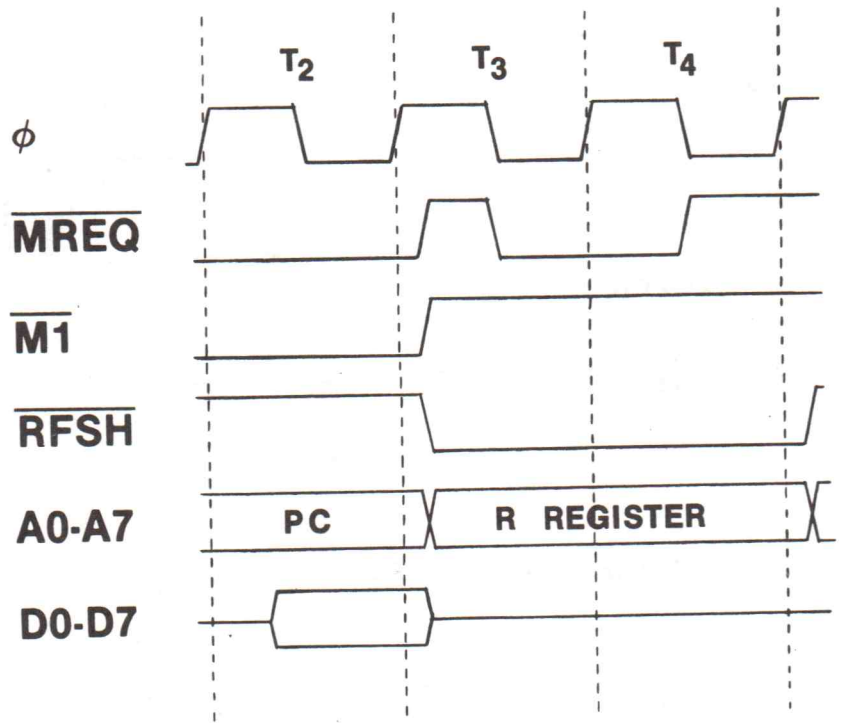


Figure II:2-4.1 RAM Refresh Timing

The refresh process begins as /M1 returns high after the OP code fetch. At the same time /RFSH goes low, and /MREQ goes high to terminate the current memory operation then low to indicate another is to be done. /RD goes high and remains there to prevent the data refresh process from passing data to the bus. The Z-80 also puts the contents of the R register on address lines A0 - A6 at this time (see Figure II:2-2.4).

The Z-64 refresh logic uses these signals to set RAM refresh in motion. The first priority is to return the MUX select input to the high state so that A0 - A6 are sent to the RAM array. This occurs when the PRESET input to chip T2 goes low soon after the rising edge of clock cycle T3. The select signal remains high for the duration of the M1 cycle so A7 - A13 don't get MUXed in and /CAS won't be activated.

Unlike memory reads and writes, all four /RASn lines are enabled. The active /RFSH on PROM L2 distinguishes a refresh from a R/W cycle. Thus, a bit row in all four rows of chips is refreshed. Only 128 M1 cycles are needed to refresh the entire 64K of memory.

II:2-5 Optional EPROM

The Z-64 is socketed to allow installation of a 1K or 2K EPROM at board location NP2. The EPROM is normally addressed for locations F800H to FBFFH for a 1K device or F800H to FFFF for a 2K device. Since this is the usual address for the DOUBLER firmware, an alternate location of F000 - F7FFH is available. The system memory from F800H to FFFFH (or F000H to FFFFH) contained in the dynamic RAM array is never accessed. This is a function of the PROM at L2 and prevents contention of memory locations between different memory devices.

When power on jump is installed (via jumper JP6) in conjunction with the EPROM, a system reset or power up disables memory in favor of the EPROM for three machine cycles. The first three EPROM addresses are read by the CPU and should contain a jump instruction (first byte) and an address (next two bytes) to a system initialization routine. This routine can also be contained in the EPROM.

In non-power on jump operations, the EPROM is selected when address lines A11 - A15 are all high. Notice that they are high (logical 1) only when A0 - A15 is F800H and up (F8 = 1111 1000). The logic in the 74S471 PROM responds to these addresses by enabling the /PROM signal (no /RAS is enabled) and automatically installing a wait state (WAIT active) in the machine cycle to allow the slower EPROM to decode the address. Address inputs A0 - A10 for the 2K version, A0 - A9 for a 1K, are decoded and the output buffer is enabled. When /PROM is inactive (high), the EPROM output buffer is maintained in the high impedance state.

Jumper JP7 allows the EPROM to be re-addressed for F000-F7FFH. The jumper, when installed, provides a bridge for address line A11 between the processor and pin 19 of L2. When JP7 is removed, the line is held high by a pull-up resistor connected between +5 and pin 19, regardless of the level of A11. Thus, the EPROM is enabled (system memory disabled) beginning with location F000H.

NOTE:

The EPROM cannot be programmed on the Z-64. The device should be programmed elsewhere before installation.

JUMPER INSTALLATION: Several different EPROM devices are jumper selectable for installation in the Z-64. Jumper area JP5, immediately to the right of the EPROM, is used to configure the board for the desired device. The discussion below indicates the appropriate JP5 connections for a 2708 1K device (the setting installed at the factory), the Texas Instrument TMS 2716 2K device, and the Intel 2716 2K device. The last two devices are not interchangeable; the JP5 jumpers for one preclude use of the other.

2708 1K EPROM: Address line A10 is not used for a 1K EPROM. Four jumpers must be installed to provide -5V to EPROM pin 21 and +12 to pin 19, to ground the PROGRAM pin (only high when the chip is programmed), and to put chip select to pin 20.

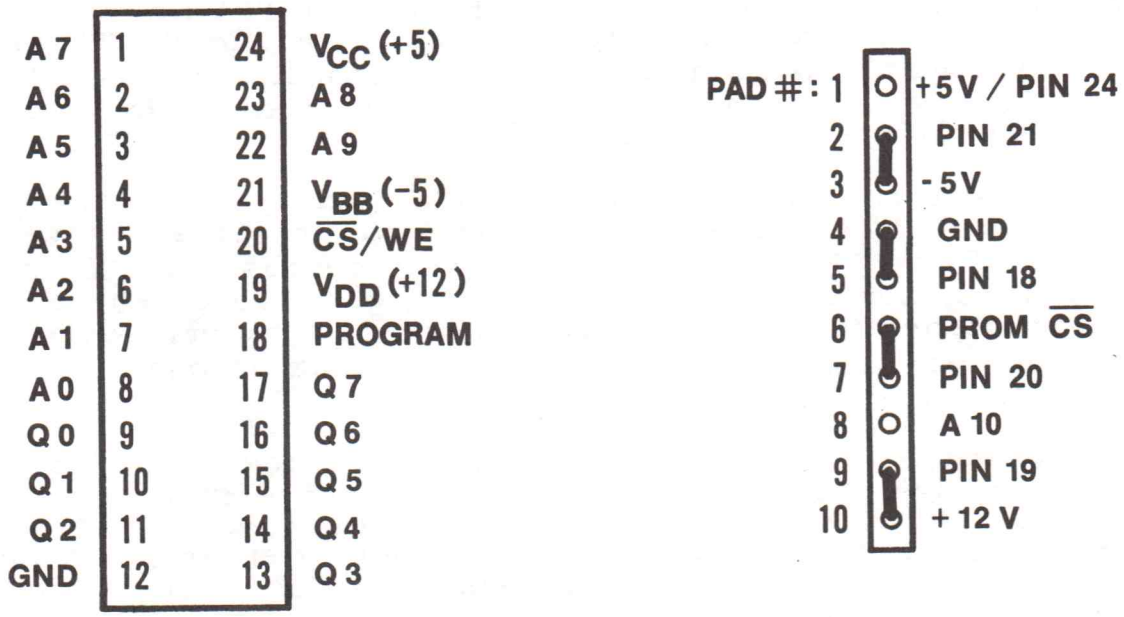


Figure II:2-5.1 JP5 Jumpers for 2708

Texas Instruments TMS 2716 2K EPROM: Jumpers are installed to connect -5 to EPROM pin 21, chip select to pin 18, address line A10 to pin 20 and +12 to pin 19.

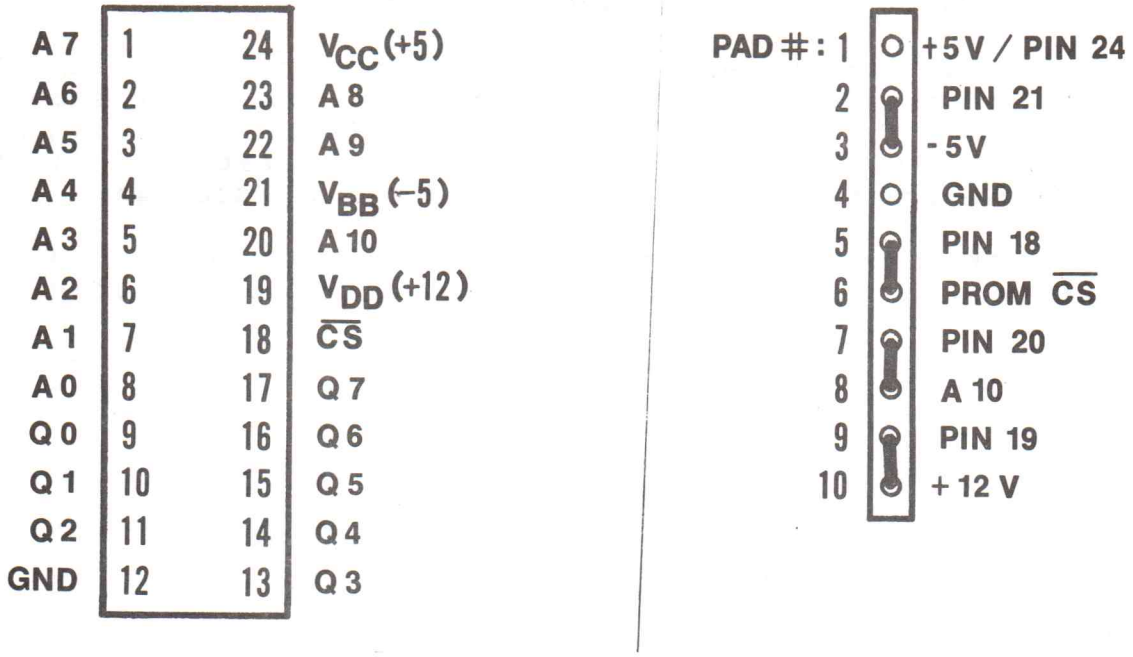


Figure II:2-5.2 JP5 Jumpers for TMS2716

Intel 2716 2K EPROM: The jumpers indicated below connect +5 to EPROM pin 21 (a higher voltage, about +25, is applied to this pin to program it), chip select to pin 18, ground to pin 20 and address line A10 to pin 19.

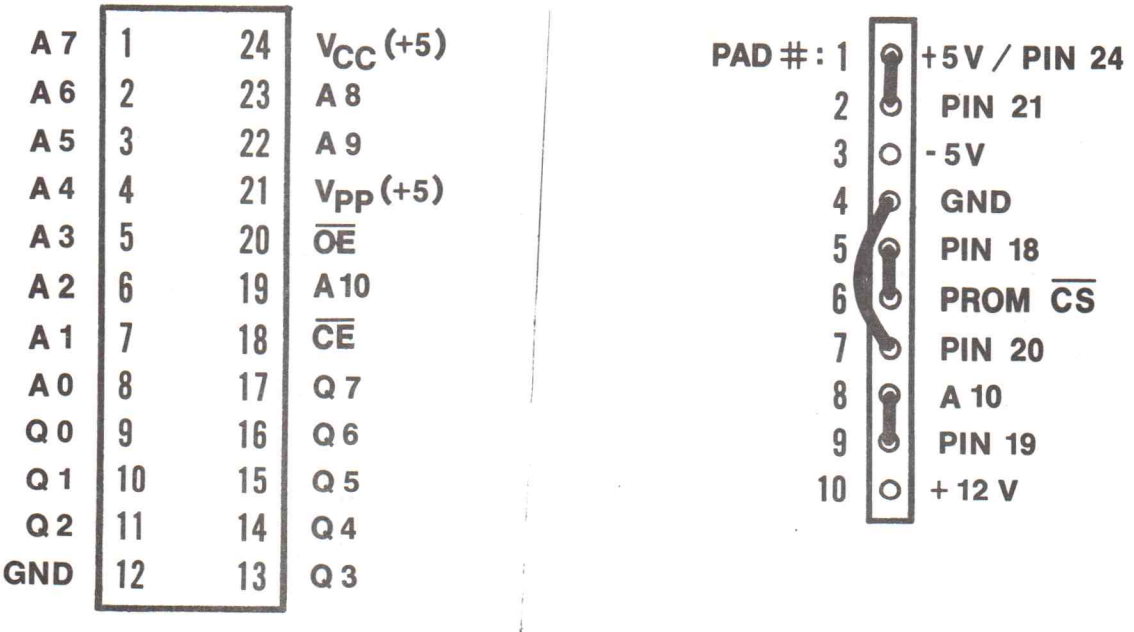


Figure II:2-5.3 JP5 Jumpers for Intel 2716

II:2-6 Phantom Line

The phantom line, S-100 bus pin 67, is available to disable on-board memory in favor of an external device. For instance, the Micromation DOUBLER floppy disk interface uses the phantom line to disable the system memory during disk accesses.

Refresh occurs unimpeded when the phantom line is enabled. Disabling memory with active phantom is a function of the same PROM (at location L2) as the memory enable (/RASn) and refresh functions. All /RASn lines are enabled during the refresh portion of an M1 cycle regardless of the status of the phantom line.

Notice that the /PHANTOM signal at input pin 67 of the S-100 bus is active low. It is inverted at CD5 so that it is active high at input to the L2 PROM.

II:3. ADDRESS/DATA BUS CONTROL

Control of the address and data buses centers around th 8304 tri-state bidirectional transceivers at M4, N4, P4 and R4 and the 74S471 bi-polar PROM at EF5. The transceivers determine the direction of the data flow and tri-stating via the DIR and /CS (chip select) inputs, respectively. A change is effected by the outputs of the PROM. The inputs to the PROM are control and status signals that indicate where the data is coming from and where it should go.

The figure below illustrates the components of this section. Notice that although M4 and P4 appear to overlap, they, in fact, do not. M4 handles address lines A0 - A7; P4 address lines A8 - A15. N4 performs a function relative to Z-64 compatibility with an 8080 environment (through the S-100 bus). This is described in section II:3-2 below.

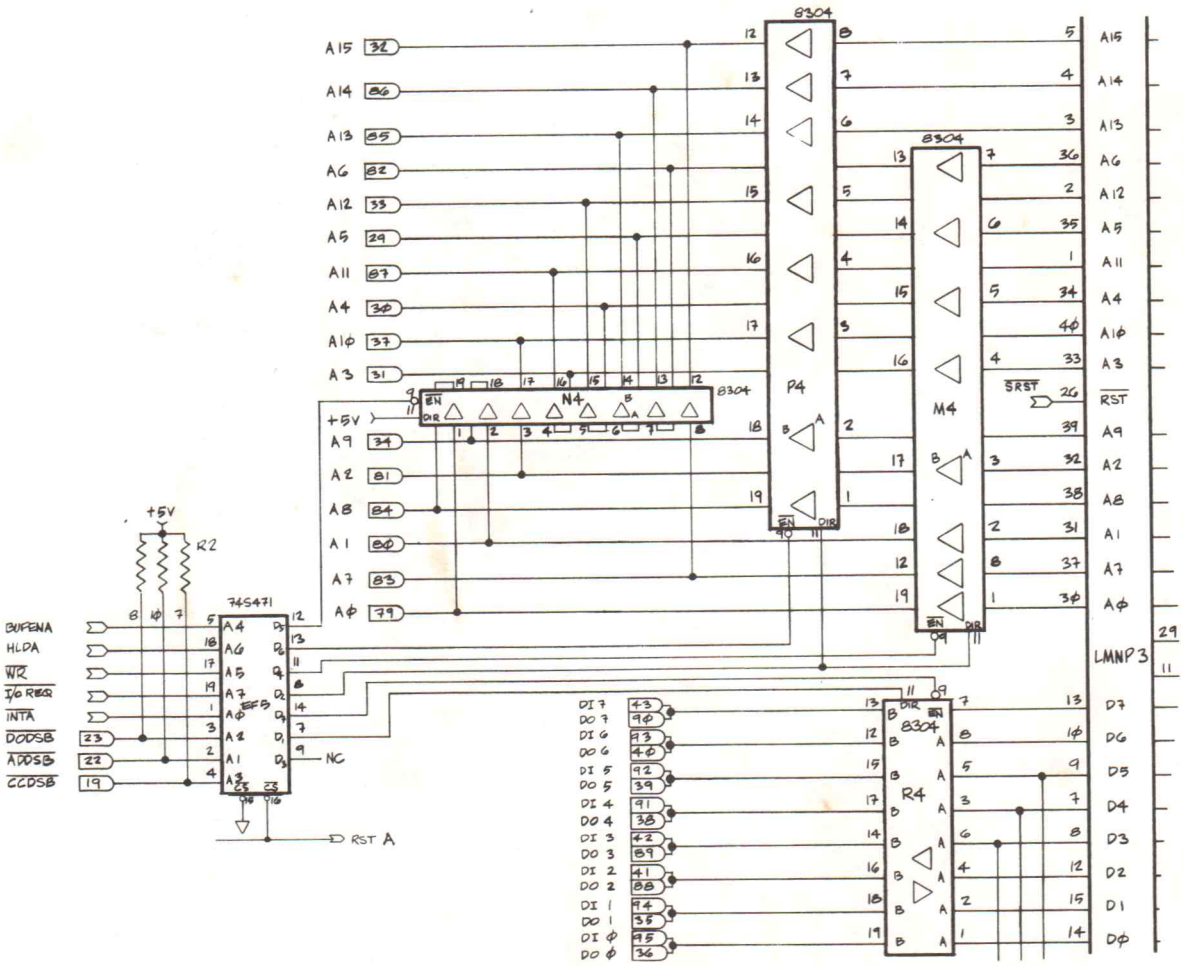


Figure II:3.1 Data/Address Bus Control Circuitry

II:3-1 Signal Definition

The following signals are input to the PROM at EF5.

BUFENA: This is the output of the bi-polar PROM at L2 (see section II:2). It is active (high) when the phantom input to L2 is high or the L2 PROM select is not active. This second condition is important. Since L2 is not enabled during I/O requests, BUFENA is high enabling the appropriate address and data buffers. Of course when the phantom line is enabled, BUFENA is also high enabling these buffers (but with different results.)

HLDA: HLDA is the hold acknowledge output (the Z-80 calls it /BUSAK, bus acknowledge) from the processor. The bus request (8080 HOLD, Z-80 BUSRQ) and acknowledge signals are used to perform DMA and are not implemented on the Z-64.

PDBIN: This signal is the processor data bus in control signal. It is a function of RD (inverted processor signal /RD). When active (high), the data bus drivers send the data to the processor.

/IOREQ: This signal is used to distinguish an I/O request from a memory request. It also indicates that an 8-bit I/O device port address is on the low order byte of the address bus. The status of PDBIN determines the direction of the data.

/INTA: When an interrupt is received by the processor a special M1 cycle results in which /IOREQ goes active without /RD or /WR active. /IOREQ and /M1 active begets /INTA. Thus it is distinguished from a I/O request in which /IOREQ is active with /RD or /WR but /M1 is inactive.

/DODSB, /ADDSB, /CCDSB: These signals are used to control bus functions in response to a bus request. All active would turn the direction of the address, data, and tri-state control bus from the out direction (normal under processor control) to the in direction allowing an external processor or controller to control the buses. This is most frequently used in DMA processes. They are not implemented on the PROM at this time.

II:3-2 Bus Driver at N4

When the /CS input is active (low), the contents of address lines A0 - A7 are duplicated on lines A8 - A15. Note that the direction is held high so the contents are transferred in one direction only.

The purpose of this buffer is to maintain compatibility with the 8080 vis-a-vis I/O addressing. On the Z-80, the I/O address is placed on address lines A0 - A7. On the 8080, the 8-bit port address appears on both A0 - A7 and A8 - A15. Fortunately, A8 - A15 are not significant during I/O requests by the Z-80. The two bytes thus can be equivalent without requiring an address multiplex nor affecting system performance. N4 is only active when /INTA is inactive (high) and /IOREQ is active (low).

II:3-3 74S471 PROM at EF5

This bi-polar PROM presently responds to inputs from BUFENA, PDBIN, /IOREQ, and /INTA. The outputs control the direction of the data flow on the address and data buses (including no flow at all - high impedance mode). There are six different states relative to the status of these inputs:

1) Default
2) I/O write
3) I/O read
4) External memory write
5) External memory read
6) System reset

1) Default: In this state, the data bus buffer/driver (R4) is not selected and hence in the high impedance mode. The data bus is available to the memory array or interrupt priority decoder only. P4 and M4 are enabled with the contents going out. N4 is disabled. This is the condition when none of the other states are in effect.

2) I/O Write: When the program calls for a write to a peripheral device, the appropriate address buffers must be activated and the direction of the data buffer set. In this case, the I/O port should appear on both address lines A0 - A7 and A8 - A15 to appease Z-80 and 8080 based I/O conventions. Consequently, P4 is disabled (high on pin 9) to lock out the high order byte on the address bus and N4 is enabled to place the low order byte on A8 - A15. Of course, the data bus driver at R4 is enabled (low on pin 9) and direction is set to out (high to pin 11). Lows on the /IOREQ and PDBIN inputs are the cues for this response.

3) I/O Read: The conditions of M4 (off), P4 (on),and N4 (on) are the same as above. The only difference is the direction set on R4. In this case a low (for data in) is put to pin 11. This state is a result of a low /IOREQ with a high on PDBIN.

4) External Memory Write: This differs from I/O request in that a complete address (all 16 bits) is being used. Also, the phantom line must be enabled to disable the Z-64 memory array. Signal /IOREQ is inactive (high) and PDBIN is inactive (low). This combination disables N4 and enables P4 and M4. R4 is enabled and the direction of data flow is out.

5) External Memory Read: This situation is similar to 4 above except PDBIN signal is high (active). Given /IOREQ inactive again, P4 and M4 are enabled, N4 disabled (all 16 address line used) and R4 enabled, data going in.

6) System Reset: Chip select for EF5 is set by the RSTA signal output from the reset circuitry. The only time the device is not selected is when the system reset button is depressed. This prevents spurious outputs or inputs from the system busses (data and address) during reset.

In states 2 - 5 above, /INTA, /DODSB, /ADDSB, /CCDSB and HLDA are inactive and BUFENA is active. If any of these signals are otherwise, the data and address buses go to the default condition (state 1).

Of special interest is an interrupt acknowledge (see section II:8 below for this discussion if not familiar with Z-64 interrupt acknowledge procedures). In Mode 0, the data and address default conditions are appropriate; i.e., the address bus is enabled and the data bus is in the high impedance state. With this configuration, the interrupt priority encoder at AB5 is the only device (since the L2 PROM is not enabled) that has access to the data bus. The information provided by the device provides one of 8 addresses for the interrupt vector. Mode 1 is no problem as well since the processor is automatically sent to location 0038H (the contents of the data bus are ignored). Mode 2 is not operative. In the data and address bus default condition, the data bus buffer is in the high impedance mode preventing the external device from sending the 8-bit address to the processor.

Direct memory access (DMA) is another operation of special interest. The Z-64 circuitry appears to support it, but the PROM does not.

II:3-4 Summary

The following is provided as a summary of the 5 states relating to input signals to the buffer status. The default condition is shown first and exists when the other four do not.

1) Default Condition

R4: high impedance
M4: enabled, out
P4: enabled, out
N4: high impedance

2) I/O Write

Inputs		Effect
BUFENA:	H	R4: enabled, out
HLDA:	L	M4: high impedance
PDBIN:	L	P4: enabled, out
/IOREQ:	L	N4: enabled, out
/INTA:	H	
/DODSB:	H	
/ADDSB:	H	
/CCDSB:	H	

3) I/O Read

Inputs		Effect
BUFENA:	H	R4: enabled, in
HLDA:	L	M4: high impedance
PDBIN:	H	P4: enabled, out
/IOREQ:	L	N4: enabled, out
/INTA:	H	
/DODSB:	H	
/ADDSB:	H	
/CCDSB:	H	

4) External Memory Write

Inputs		Effect
BUFENA:	H	R4: enabled, out
HLDA:	L	M4: enabled, out
PDBIN:	L	P4: enabled, out
/IOREQ:	H	N4: high impedance
/INTA:	H	
/DODSB:	H	
/ADDSB:	H	
/CCDSB:	H	

5) External Memory Read

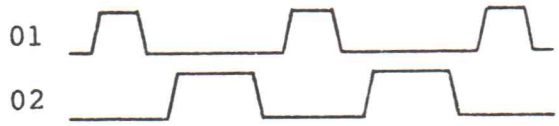
Inputs		Effect
BUFENA:	H	R4: enabled, in
HLDA:	L	M4: enabled, out
PDBIN:	H	P4: enabled, out
/IOREQ:	H	N4: high impedance
/INTA:	H	
/DODSB:	H	
/ADDSB:	H	
/CCDSB:	H	

II:4 CLOCK

The Z-80 requires one square wave clock signal at either 2 or 4 MHz. The 8080, on the other hand, requires two non-overlapping clock signals at 2 MHz. The clock signal generating circuitry on the Z-64 provides both.

Ø is usually used to indicate the clock signal. Unfortunately, the typesetting device does not support this character. Consequently, "0" is used instead.

By convention, the 8080 signals are named 01 and 02 and appear as follows.



Notice that they are out of phase. This is crucial to 8080 operation. The Z-80 runs on the 8080 02 square wave signal (but not 01). The 8080 requires a 2 MHz frequency for 01 and 02. The Z-80, however, operates at both 2 MHz and 4 MHz.

The figure below illustrates the clock generation circuitry. The 74LS93 is used as a divide by two counter. The configuration of the jumpers at JP4 determine if the clock U2 QC output (74LS93 pin 8) has a 2 or 4 MHz frequency. The three configurations are shown below the circuit diagram. The appropriate jumper is determined by the revision number and desired clock speed.

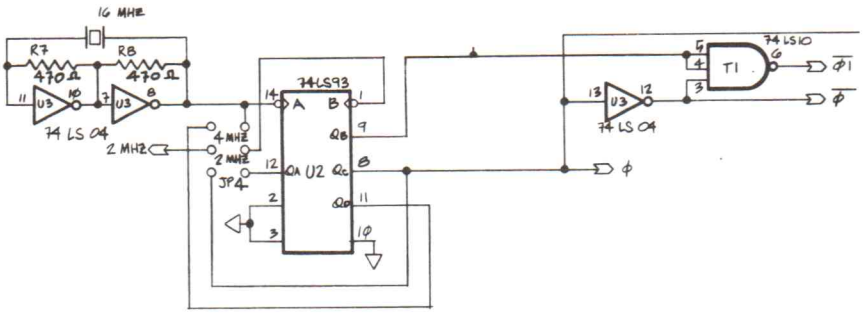


Figure II:4.1A Clock Generation Circuitry

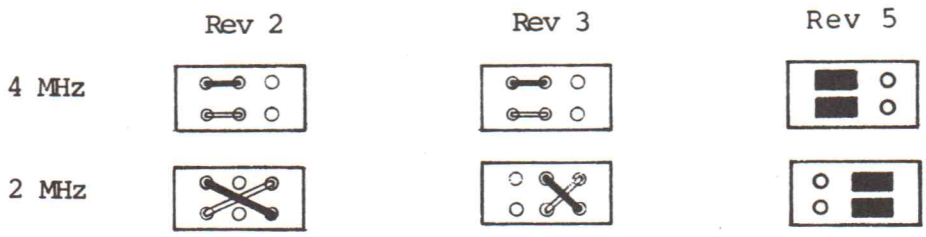


Figure II:4.1B JP4 Jumper Configurations

Notice that regardless of the jumpers, installed a 2 MHz clock signal (square wave) is output. This signal makes its way down to T4 for output to the S-100 bus, pin 49.

The source signal for U2 is a 16 MHz signal from the free running crystal oscillator. The output from U2 Q[C] is the clock signal used by the Z-80. The deciding factor between the two frequencies is whether the A and B inputs to U2 are tied together (4 MHz) or if input B is tied to output Q[A] (2 MHz). Output Q[A] is a product of input A only. Outputs Q[B - D] are a product of the B input. Q[A] and Q[B] are divide by two outputs of A and B while Q[C] is a divide by two output clocked by Q[B]. Q[D] is a divide by two output clocked by Q[C].

The 01 signal for use by the 8080 is generated from the 74LS10 3-input NAND at board location T1. The important factor is to phase it properly in reference to the square wave clock signal 02. The timing diagram below illustrates the inputs and outputs. Notice that 01 is actually the inverse of this signal. Inversion is performed at L4 before this signal is output to the S-100 bus, pin 25. 01 is a function of the system clock. If you need 01 with a 2 MHz frequency, the Z-64 will have to operate at 2 MHz.

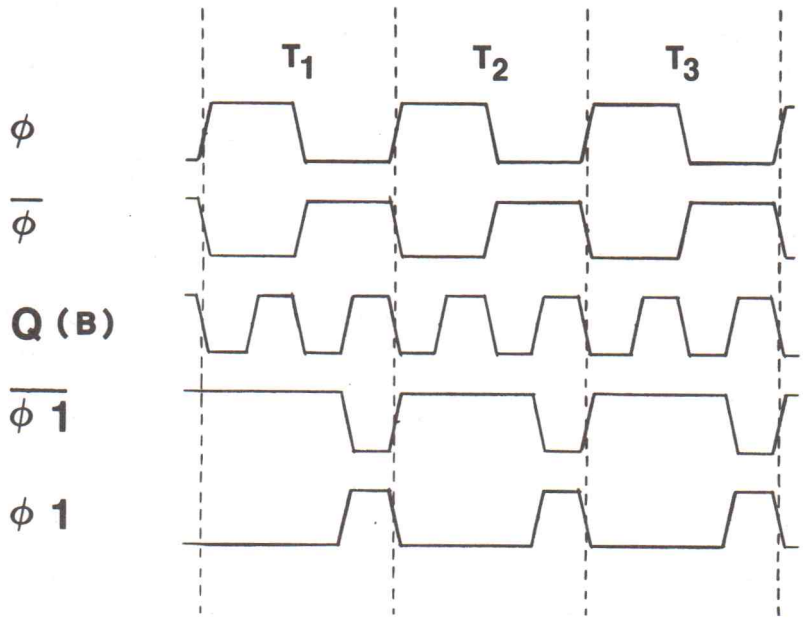


Figure II:4.2 φ1 Generation

The clock signals are output to the S-100 bus at two locations: the 74368 at board location L4 (01) and the 74367 at T4 (02 and CLK). The '368 is an inverter. Notice that in 4 MHz operation 02 is 4 MHZ and CLK is 2 MHz. In 2 MHz operation, they're both the same. The frequency of 01 matches the operating clock frequency of the Z-64.

II:5 RESET

The figure on the following page illustrates the RESET circuitry. To maximize IC utilization, the 74LS175 at T3 is also used to generate SYNC and /SYNC. These signals are not related to the RST function. See Section II:1-4 for a description of their purpose.

System reset is generally used to terminate program operation or to get back to the operating system after a program crash. Functionally, it forces the program counter to zero and initializes the CPU. In the Z-80, initialization

- disables the interrupt flip-flop
- sets registers I and R (interrupt and refresh respectively) to 00H
- sets the Z-80 to Interrupt Mode 0

There are two circumstances that will generate a low (active) /SRST to the Z-80: 1) power-up and 2) user initiated reset.

1) At power-up, the resistor-capacitor pair keeps the T2 PRESET input (and /BPOC) low. This renders a low output at /Q (/SRST), regardless of the clock input. The CLEAR input is high at this time since all /Q outputs of the 74LS175 are high when its CLEAR input (from /BPOC) is low. This condition remains until the 39uF capacitor charges up. Once charged, the PRESET input is held high.

   Although the status of PRESET has changed, the output signals remain the same: Q = H, /Q = L. They will remain in this state until the T2 CLR (clear) input changes. A change in CLR happens 3 clock cycles later when the /Q4 output of T3 goes from high to low. Notice that T3 CLR is the /BPOC signal. When /BPOC went high, it enabled T3. (Low CLR to a 74LS175 holds the Q output low and the /Q output high.) It takes 3 clock cycles for the change to be manifested at Q4 since the inputs are connected to the outputs of the previous section and changes occur one per clock cycle (from the inverted system clock signal). When /Q4 goes low, it drops T2 CLR and the Q and /Q output flip-flop.

   The same procedure in T3 is repeated to set the T2 CLR input to high. (It takes 3 clock cycles.) Once T2 CLR is high, the outputs will remain as they are (Q = L, /Q = H) until a rising edge is detected on the clock input.

2) A rising edge on the clock input to T2 only occurs after the system reset button is switched. A section of the 74LS113 at NP1 is used for this purpose. The device is necessary to debounce the switch.

   When the button is pressed, it connects the NP1 clock input to ground and disconnects the NP1 PRESET input from ground. The 74LS113 is triggered by a high to low transition on the clock

input. Since PRESET is now high, a low is rendered at the Q output, signal /RSTA. This is input to T2.

Notice that the clock input was forced low when the switch was set. Recall, however, that the 74LS74 clock input is rising edge sensitive. Consequently, /SRST isn't actually set until the system reset switch is released. Reconnecting NP1 PRESET to ground causes a high output on the Q output. This supplies the rising edge for T2.

/SRST will stay low for 3 clock cycles as described above. This comforms with the Z-80 time requirements for clearing the internal registers.

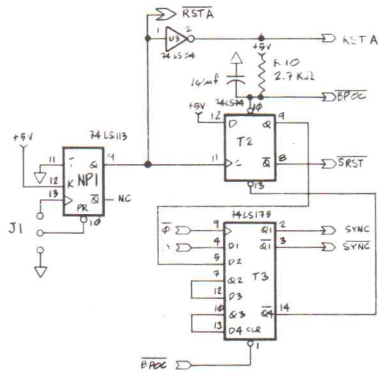


Figure II:5.1 RESET Circuitry

The reset button on the computer cabinet must be attached to jumper J1.

II:6 Power on Jump

The power on jump, POJ, logic is used to disable memory and enable the EPROM at NP2. The circuitry shown in Figure II:6.1 sends an active (low) /POJ at reset or power-up to the bi-polar PROM at L2 which enables /PROM instead of a /RASn. /POJ is held active for 3 machine cycles. During this time, the CPU is fetching addresses 0000H - 0002H. Since /PROM is active, the contents of F800H - F802H, the first three locations of the EPROM, are yielded instead. These locations should contain a jump instruction and 16-bit address for the system initialization routine.

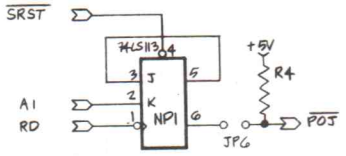


Figure II:6.1 Power on Jump Circuitry

The 74LS113 is a dual J-K negative edge triggered with preset device. On the Z-64 the J input is tied to the Q output, address line A1 to the K input, and the inverted CPU /RD signal acts as the clock. The decision making process is a function of the PRESET input, /SRST. During power-up and reset, /SRST is low (see section II:5 above). When PRESET is low on this device, all other inputs are disregarded and the Q and /Q outputs are forced high and low, respectively. /Q is the /POJ line; input to the PROM at L2.

After generating an active /POJ, the logic must turn it off at the proper time (after 3 machine cycles) so the /PROM signal is disabled and the normal addressing mode of the CPU can take over. The timing diagram below illustrates this sequence. Notice that the diagram picks up the reset as the /SRST returns high. (/SRST stays low for 3 clock cycles.)

The K input, address line A1, is low as addresses 0000H and 0001H are fetched by the CPU. Not until 0002H is read does this line go high. When K and J inputs are read high at the falling edge of RD, the outputs are toggled. /POJ goes high by which time the jump instruction and address (both bytes) have been loaded into the CPU.

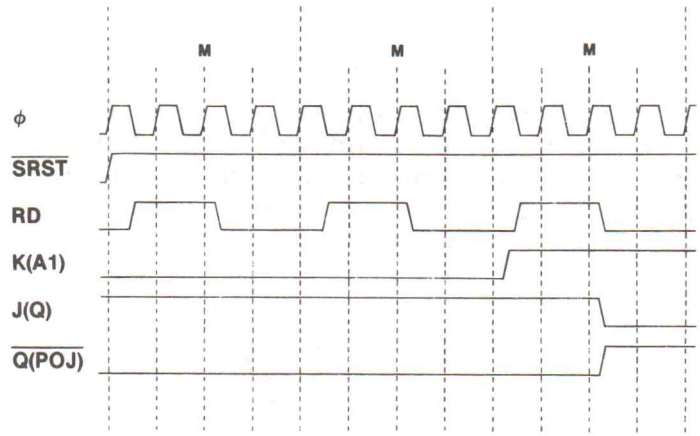


Figure II:6.2 POJ Timing

During operation the RD signal continuously clocks further comparisons of J and K inputs. Notice that the next comparison after /POJ goes high contains a low on J and either a high or low on K (depending upon the address of the initialization routine). The status of K is irrelevant in this instance as long as J is low. A high K will render Q and /Q low and high respectively (the inactive state). A low will maintain the two outputs in their current status (which is inactive). Thus, the power on jump is not activated again until the /SRST signal goes low.

II:6-1 POJ without EPROM

When the Z-64 is shipped from the factory, jumper JP6 is **not** installed, and, hence, the POJ logic not implemented. Power on jump is supplied as a service for those who intend to install an EPROM at NP2.

When the Z-64 is mated with the DOUBLER disk controller, no POJ logic is required. The DOUBLER has its own POJ circuitry that enables the phantom line for three machine cycles upon reset or power-up and supplies the jump instruction and address. (Active PHANTOM also disables system RAM.) The initialization routine, in the form of an operating system boot, is thus initiated.

If a different disk controller is to be used, it should perform similarly to the DOUBLER. That is, at power-up or reset it should have the requisite firmware to initiate a system initialization routine. (Remember: the firmware must do this in response to addresses 0H - 2H.) Again, loading the O/S is a suitable routine. If your controller does not contain this firmware, the EPROM will be necessary and POJ jumper JP6 should be installed.

II:7 POWER

The S-100 bus typically provides +8V on pins 1 and 51, +16V on pin 2 and -16V on pin 52. Ground is available on pins 50 and 100. The Z-64 requires +5, -5 and +12. The reduction is performed by the regulators at board locations VR-1 and VR-4 (+5), VR-2 (+12), and VR-3 (-5). The figure below illustrates the circuitry.

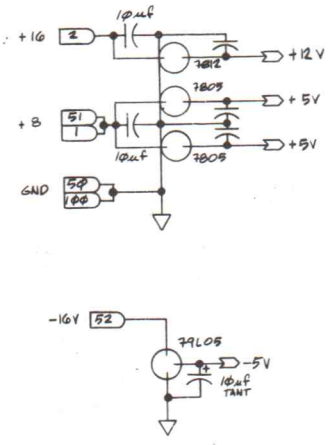


Figure II:7.1 Power Regulators

The power requirements of the Z-64 are

+8V at 700 Ma
+16V at 200 Ma
-16V at 10 Ma

Most ICs require +5V as the supply voltage. A couple of ICs require +12V and others -5V (the RAM chips, for instance, need all three).

Capacitors are used throughout the Z-64 to remove 'noise' caused by the voltage level transitions in the ICs. Within the RAM array, 0.1uF ceramic caps connected between +12V and ground are alternated with 0.1uF ceramics between -5V and ground. 0.01uF caps are connected between +5V and ground to provide decoupling which prevents noise from affecting TTL logic. These appear between every four or five devices. Low frequency decoupling is provided with 10uF tantalum capacitors between +12V and ground every 16 devices and the same between -5V and ground every 32 devices.

II:8 INTERRUPTS

There are two types of interrupts optionally available on the Z-64: a software maskable interrupt and a non maskable interrupt. The first is invoked by a low input to the Z-80 on the /INT line. It is acknowledged only if the software controlled interrupt enable flip-flop has been previously enabled. (If disabled, the interrupt is ignored.) Non maskable interrupts are performed upon a low input on the /NMI line. An /NMI is only overridden by a bus request. The CPU responds to /INT with a special M1 cycle and a low output on the /IOREQ (I/O request) at the end of the current machine cycle. /NMI causes a jump to location 0066H and a read cycle. Both /INT and /NMI are sampled at the rising edge of the last clock cycle of every machine cycle (see Section II:1 for timining illustrations).

II:8-1 Software Maskable Interrupt

The figure below illustrates the maskable interrupt circuitry. The interrupting devices are serviced by the 25LS2513 at AB5. This is a tri-state priority encoder fed by S-100 lines VI0 - 7. The highest priority is given to line VI0, the lowest to VI7. A low on any of the inputs begins the interrupt sequence.

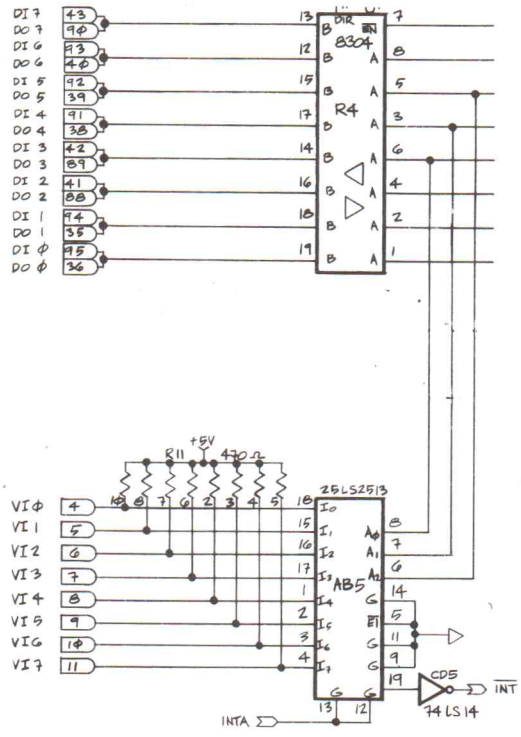


Figure II:8-1.1 Maskable Interrupt Circuitry

The interrupt response is initiated when a low is received on any of AB5 inputs VI0 - 7. If two lines are active (low), the lowest numbered has the priority. The enable output (from pin 19) line goes high indicating that an interrupt requires servicing. This is immediately inverted at CD5 and forwarded to the CPU on /INT. On the rising edge of the last clock cycle of every machine cycle, the /INT line is sampled by the CPU. When this signal is active and the enable flip-flop is set, the processor responds in one of three ways depending upon the mode set by the programmer.

Mode 01: This mode is identical to the 8080A interrupt response mode. It is also the default upon power-up and reset. (The others are set under program control.) When in Mode 0, the interrupt is acknowledged by a special M1 cycle. /M1 and /IOREQ outputs go low, and the CPU reads the data bus for a 8 bit interrupt vector. The combination of these two signals active begets INTA and /INTA via the 74S471 bi-polar PROM at S3. Notice that the /MREQ line remains high. This is to prevent RAM from cluttering the data bus with conflicting data. The only device that can output data is the 25LS2513.

Outputs A0 - A2 are maintained in the high impedance state until interrupt acknowledge, INTA, goes high. When INTA goes high, the contents of A0 - A2 are passed to the data bus lines D3 - D5. The interrupting device is thus uniquely identified by the relationship between VI0 - VI7 and A0 - A2 shown in the truth table below. Note that enable input /EI, pin 5, is held low.

Table II:8-1.1
25LS2513 Truth Table

/EI	/I0	/I1	/I2	/I3	/I4	/I5	/I6	/I7	A0	A1	A2	/EO
L	H	H	H	H	H	H	H	H	L	L	L	L
L	X	X	X	X	X	X	X	L	H	H	H	H
L	X	X	X	X	X	X	L	H	L	H	H	H
L	X	X	X	X	X	L	H	H	H	L	H	H
L	X	X	X	X	L	H	H	H	L	L	H	H
L	X	X	X	L	H	H	H	H	H	H	L	H
L	X	X	L	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	L	L	L	H

The CPU reads D0 - D7 and interprets that data. The entire byte of data provides an OP code RST instruction of which there are 8. Each one is a restart instruction to a separate location in memory. Thus, the CPU is directed to the appropriate service routine for the specific interrupting peripheral. The table below lists the 8 instructions and the addresses set aside for a service routine or jump instruction.

Table II:8-1.3
Restart Group

Call Address*	0000	0008	0010	0018	0020	0028	0030	0038
OP code	C7	CF	D7	DF	E7	EF	F7	FF
	RST0	RST8	RST16	RST24	RST32	RST40	RST48	RST56

* in hex

The 3 bits D3 - D5 indicate the absolute address in hex for the beginning of the service routine. (The decimal equivalent is indicated by the number following RST.)

When an interrupt is serviced, the next instruction in the program counter is pushed to the external stack. A RETI (return from interrupt) instruction at the end of the peripheral service routine pops this value back to the program counter to continue program execution.

When **Mode 1** has been selected by the programmer, the CPU disregards the contents of the data bus and performs a restart to address 0038H. This is very similar to non maskable interrupt response discussed below and suitable when only one peripheral need be serviced or a number devices can be serviced with the same routine. In mode 1, instruction RST56 (OP code FF) is always executed.

NOTE: The discussion of Mode 2 is for your information only. This mode of interrupt response is not presently supported on the Z-64.

Mode 2 allows an indirect call to any memory location for interrupt response. The contents of the I (interrupt vector) register is appended to a byte from the interrupting device (high order and low order bytes respectively) to provide a 16-bit pointer to the specific location.

This mode requires special support from peripheral device to provide the 8-bit programmed vector during interrupt acknowledge. If you decide to modify the Z-64 to accommodate mode 2 (which will void the warranty) check the documentation for the peripheral device to ensure the necessary hardware and software components are present.

II:8-2 Non Maskable Interrupt (NMI)

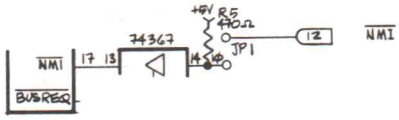


Figure II:8-2.1 NMI Circuitry

The NMI feature on the Z-64 is user jumperable. As shipped, the board does not have the jumper installed. To incorporate this feature, connect the holes at JP1 (between columns D and E at the bottom of the board). This input to the Z-80 is normally held high by the pull-up resistor at R5. The 74367 hex bus driver at board location T4 protects the Z-80 from high voltage inputs and reduces the "noise" acquired on the S-100 bus.

When NMI is active, the processor responds with a special machine cycle in which the contents of the data bus are ignored, the program counter is pushed to the external stack and the CPU jumps to location 0066H.

/NMI is not subject to the internal interrupt enable flip-flop, but defers to a /BUSREQ active.

/NMI should be used when immediate emergency overrides are necessary (e.g., impending power failures during program execution). Memory location 0066H should contain the interrupt service routine or a jump instruction to it. An NMI routine is exited with the RETN (return from non maskable interrupt) command.

SECTION III

Microprocessors, Microcomputers and the Z-64

Figure III.1 illustrates a complete computer system. The Z-64 is a computer on a card (a printed circuit board). There is an important distinction here. A computer performs data processing. A computer system combines this capability with a facility for data entry and computer control (the terminal), another for program and data storage (the floppy disk drives), and an operating system (stored as a program on a floppy disk) to coordinate communication between these components.

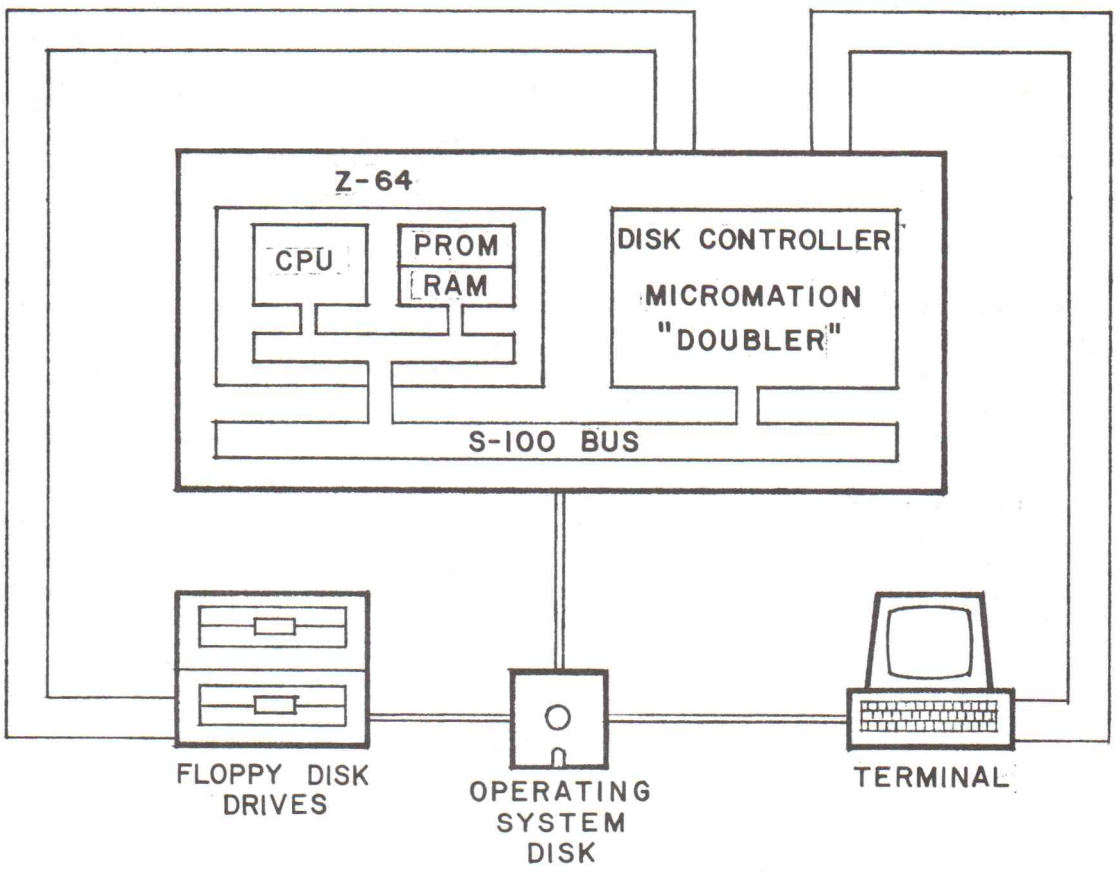


Figure III:1.1 Computer System

This section is about the Z-64, one of the parts in a computer system. It contains two components crucial to microcomputer system operation: the CPU (Central Processing Unit) and RAM. These two components do not a computer system make but do provide the foundation upon which to build one.

III:1 THE CPU

In a very basic sense, the CPU (central processing unit) can be thought of as the brain of the computer. This isn't totally accurate, but it is a good place to start. As the brain, the CPU doesn't think but instead recognizes and responds to certain commands. These commands are called the instruction set. The CPU, then, has the capability to perform a given set of operations, but the programmer has to speak its language.

Command recognition is resident to the CPU. That is, it contains the intelligence necessary for the CPU to recognize a program command and execute it. However, the CPU has no facility to store the sequences of commands that make up a program. This is the function of the memory. Random Access Memory or RAM is the expression used to describe this component.

III:2. RAM

The important thing to remember about RAM is its organization. In the Z-64, as in other computers, memory consists of thousands of units, each unit 8 bits long. This unit is referred to as a byte. When information is stored in memory, either in the form of a program or data, each bit contains either a zero (0) or a one (1). Thus, information is stored in byte form. In addition, each byte has a unique address. When the CPU writes a byte of data in memory at a specific location, it can rest assured that that information will be there upon future requests. The Z-64 has 65,536 separate and uniquely identifiable locations to

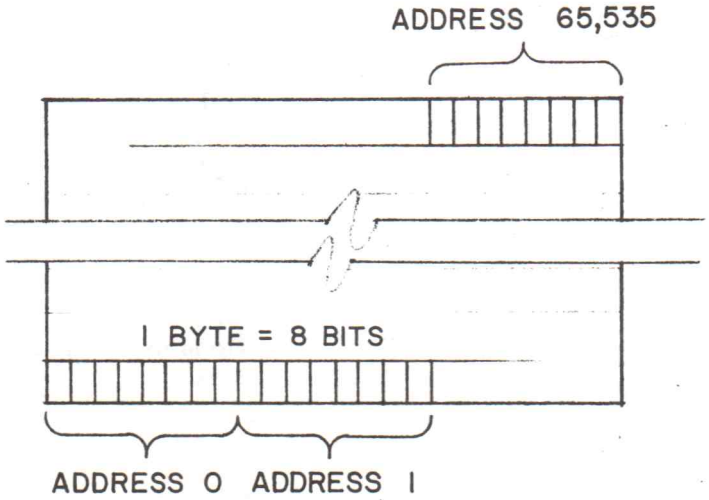


Figure III:1.2 RAM Organization

store information. Since this is a cumbersome number, it is rounded down to 64,000 which is further abbreviated to 64K (where K means kilo or thousand).

There is one other important point about RAM. When the computer system is turned off, the information contained within disappears. All programs and data are destroyed. There are two ways to save this information. One is on a floppy disk (or other magnetic storage device such as tape or cassette). Data can be moved from memory and written on the disk. The other way to permanently store information is in EPROMs, Erasable Programmable Read-Only Memory. These will be discussed in section III:3 below. But first a word about programs.

III:2-1 PROGRAMS

Programs are logical sequences of instructions to the CPU. These instructions are written in the words familiar to the processor; i.e., the instruction set. Logical in this context means the instructions proceed in the proper order. For instance, the instruction set includes a command to add two numbers. But the programmer has to tell the CPU which numbers to add. Unless these are indicated before the command to add is given, the processor is going to be very confused and the program will crash.

Crucial to program execution is the organization of memory. The CPU reads the instructions from sequential addresses in memory. Once a specific location has been read and the instruction executed, the CPU is guided to the next location in memory (by an internal component called the program counter) for its next instruction.

III:3 EPROM

In the discussion on RAM above an important limitation was stated: when power is turned off, the information stored within is destroyed. In most computer systems, it is necessary to have programs that are stored in memory permanently. To resolve this problem, EPROMs are used. Erasable Programmable Read-Only Memory is like RAM in that it is a part of the total memory space available to the computer. It is unlike RAM in that turning off the computer does not destroy the information stored in it. Thus, it provides a permanent record of programs crucial to computer operation.

III:3-1 FIRMWARE

Firmware is a program that is crucial to computer operation. Resident (physically located) in EPROM, it performs a task that is repeated frequently during operation. For instance, computer systems that rely upon floppy disks to store the operating system (a master program that controls all input and output between the user, the computer and the floppy disks) require firmware to instruct the CPU to read the operating system from the disk and transfer it into memory.

III:3-2 EPROM AND THE Z-64

The Z-64 does not contain any firmware. It doesn't need it. The board is designed to work with the Micromation DOUBLER floppy disk controller. The DOUBLER contains the requisite firmware for booting (reading and transferring) the operating system. The DOUBLER has power on jump circuitry so that when the computer system is turned on, the CPU is automatically pointed to its firmware for the start-up routine.

The Z-64 does have the socket and circuitry for EPROM installation. The Z-64 will accommodate either a 1K or 2K (one thousand or 2 thousand) byte EPROM. That is, the firmware program can be up to 2 thousand bytes long. Similar power-on-jump circuitry to that on the DOUBLER is user selectable via a jumper.

III:4 SYSTEM BUSES

Section III thus far has described the symbiotic relationship between the CPU and memory with respect to program execution. Together they form the foundation upon which to build the computer and from that the computer system. When the computer is running, data is constantly being transferred between the CPU and memory. This is done on dedicated paths called buses. There are two: the **address bus** and the **data bus.** The address bus consists of 16 separate lines. The data bus consists of 8, one for each bit of data to be transferred. (Remember that a byte, the basic unit of information, consists of 8 bits.) When the Z-64 needs a byte of data from memory it first sends out the address on the address bus then brings the information from that location back on the data bus. To write a byte of data in memory, the CPU sends out the address then sends the data to that spot.

The address and data buses are part of a larger bus; the **S-100 bus.** This is used for communication between printed circuit boards. Each PC board has a edge connector consisting of 100 fingers, numbered 1 - 50 on the front of the board and 51 - 100 on the back. The edge connector fits into a slot on another board called the motherboard. The sole purpose of this board is to provide the lines for communication between PC boards; for instance the Z-64 and the DOUBLER.

Thus far, three buses have been defined: the data, address and the S-100. The S-100 has 100 separate lines for communication, the data bus uses 16 (8 for data into the CPU and 8 for data out) and the address bus 16. That leaves 68 for other signals.

III:5 CONTROL/STATUS SIGNALS

The rest of the lines on the S-100 bus are used to provide power to the PC boards (from a separate power supply) and for internal control. The CPU receives and sends special signals on these other lines. The outgoing signals inform other components of the computer system as to the state of the processor; e.g., "I'm busy right now, hold on;" "I'm ready to accept some data;" and others. Incoming signals are used by external devices to request the processors attention, to interrupt the processor (i.e., distract it from its current program to perform another), to reset the whole computer system (terminate the current operation and start over again), and to tell the processor to wait until the external device is ready. A complete listing of the signals and their function is presented in Appendix B.

III.5-1 RESET AND INTERRUPT

These two functions terminate the on-going program. However, the results are different. An interrupt sent to the processor from another device will cause program execution to stop and be temporarily replaced by another program. The processor returns to the original program and continues execution after the interrupt has been serviced. A RESET signal sent to the processor, usually originating from the computer system operator, terminates the program in progress and initializes the system (terminates the program, restarts the program counter at the first location in memory - location 0 - and returns the status and control signals to their inactive state). The program that was in progress will not be resumed. A RESET is similar to turning off the computer then turning it on again.

III:6 The Z-64

All the features described in this section are present in the Z-64. The Z-64 utilizes the Z-80 microprocessor as its CPU, clocked at a fast 4 Megahertz. The use of this popular processor means that there's an extensive library of operating systems and application programs available to the user. The full complement of memory accompanies the Z-80 so there's no fussing with separate RAM boards and proper addressing. The density of this configuration does not preclude optional features. A socket is provided for either a 1K or 2K EPROM for use as a system monitor or in a special application. Vectored interrupts are available with installaton of another optional IC. Drawing less than 1 amp, the Z-64 is ready for installation in a number of S-100 systems where speed, reliability, and flexibility are essential to system development and use.

APPENDIX A: Z-80 INSTRUCTION SET

ADC HL, ss	Add with Carry Reg. pair ss to HL
ADC A, s	Add with carry operand s to Acc.
ADD A, n	Add value n to Acc.
ADD A, r	Add Reg. r to Acc.
ADD A, (HL)	Add location (HL) to Acc.
ADD A, (IX+d)	Add location (IX+d) to Acc.
ADD A, (IY+d)	Add location (IY+d) to Acc.
ADD HL, ss	Add Reg. pair ss to HL
ADD IX, pp	Add Reg. pair pp to IX
ADD IY, rr	Add Reg. pair rr to IY
AND s	Logical 'AND' of operand s and Acc.
BIT b, (HL)	Test BIT b of location (HL)
BIT b, (IX+d)	Test BIT b of location (IX+d)
BIT b, (IY+d)	Test BIT b of location (IY+d)
BIT b, r	Test BIT b of Reg. r
CALL cc, nn	Call subroutine at location nn if condition cc if true
CALL nn	Unconditional call subroutine at location nn
CCF	Complement carry flag
CP s	Compare operand s with Acc.
CPD	Compare location (HL) and Acc. decrement HL and BC
CPDR	Compare location (HL) and Acc. decrement HL and BC, repeat until BC=0
CPI	Compare location (HL) and Acc. increment HL and decrement BC
CPIR	Compare location (HL) and Acc. increment HL, decrement BC repeat until BC=0
CPL	Complement Acc. (1's comp)
DAA	Decimal adjust Acc.
DEC m	Decrement operand m
DEC IX	Decrement IX
DEC IY	Decrement IY
DEC ss	Decrement Reg. pair ss
DI	Disable interrupts
DJNZ e	Decrement B and Jump relative if B≠0
EI	Enable interrupts
EX (SP), HL	Exchange the location (SP) and HL
EX (SP), IX	Exchange the location (SP) and IX
EX (SP), IY	Exchange the location (SP) and IY
EX AF, AF'	Exchange the contents of AF and AF'
EX DE, HL	Exchange the contents of DE and HL
EXX	Exchange the contents of BC, DE, HL with contents of BC', DE', HL' respectively
HALT	HALT (wait for interrupt or reset)
IM 0	Set interrupt mode 0
IM 1	Set interrupt mode 1
IM 2	Set interrupt mode 2
IN A, (n)	Load the Acc. with input from device n
IN r, (C)	Load the Reg. r with input from device (C)
INC (HL)	Increment location (HL)
INC IX	Increment IX
INC (IX+d)	Increment location (IX+d)
INC IY	Increment IY
INC (IY+d)	Increment location (IY+d)
INC r	Increment Reg. r
INC ss	Increment Reg. pair ss
IND	Load location (HL) with input from port (C), decrement HL and B
INDR	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B=0
INI	Load location (HL) with input from port (C); and increment HL and decrement B

INIR	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B=0
JP (HL)	Unconditional Jump to (HL)
JP (IX)	Unconditional Jump to (IX)
JP (IY)	Unconditonal Jump to (IY)
JP cc, nn	Jump to location nn if condition cc is true
JP nn	Unconditional jump to location nn
JP C, e	Jump relative to PC+e if carry=1
JR e	Unconditional Jump relative to PC+e
JP NC, e	Jump relative to PC+e if carry=0
JR NZ, e	Jump relative to PC+e if non zero (Z=0)
JR Z, e	Jump relative to PC+e if zero (Z=1)
LD A, (BC)	Load Acc. with location (BC)
LD A, (DE)	Load Acc. with location (DE)
LD A, I	Load Acc. with I
LD A, (nn)	Load Acc. with location nn
LD A, R	Load Acc. with Reg. R
LD (BC), A	Load location (BC) with Acc.
LD (DE), A	Load location (DE) with Acc.
LD (HL), n	Load location (HL) with value n
LD dd, nn	Load Reg. pair dd with value nn
LD HL, (nn)	Load HL with location (nn)
LD (HL), r	Load location (HL) with Reg. r
LD I, A	Load I with Acc.
LF IX, nn	Load IX with value nn
LD IX, (nn)	Load IX with location (nn)
LD (IX+d), n	Load location (IX+d) with value n
LD (IX+d), r	Load location (IX+d) with Reg. r
LD IY, nn	Load IY with value nn
LD IY, (nn)	Load IY with location (nn)
LD (IY+d), n	Load location (IY+d) with value n
LD (IY+d), r	Load location (IY+d) with Reg. r
LD (nn), A	Load location (nn) with Acc.
LD (nn), dd	Load location (nn) with Reg. pair dd
LD (nn), HL	Load location (nn) with HL
LD (nn), IX	Load location (nn) with IX
LD (nn), IY	Load location (nn) with IY
LD R, A	Load R with Acc.
LD r, (HL)	Load Reg. r with location (HL)
LD r, (IX+d)	Load Reg. r with location (IX+d)
LD r, (IY+d)	Load Reg. r with location (IY+d)
LD r, n	Load Reg. r with value n
LD r, r'	Load Reg. r with Reg. r'
LD SP, HL	Load SP with HL
LD SP, IX	Load SP with IX
LD SP, IY	Load SP with IY
LDD	Load location (DE) with location (HL), decrement DE, HL and BC
LDDR	Load location (DE) with location (HL), decrement DE, HL and BC; repeat until BC=0
LDI	Load location (DE) with location (HL), increment DE, HL, decrement BC
LDIR	Load location (DE) with location (HL), increment DE, HL, decrement BC and repeat until BC=0
NEG	Negate Acc. (2's complement)
NOP	No operation
OR s	Logical 'OR' or operand s and Acc.
OTDR	Load output port (C) with location (HL) decrement HL and B, repeat until B=0
OTIR	Load output port (C) with location (HL), increment HL, decrement B, repeat until B=0
OUT (C), r	Load output port (C) with Reg. r
OUT (n), A	Load output port (n) with Acc.
OUTD	Load output port (C) with location (HL), decrement HL and B
OUTI	Load output port (C) with location (HL), increment HL and decrement B

POP IX	Load IX with top of stack
POP IY	Load IY with top of stack
POP qq	Load Reg. pair qq with top of stack
PUSH IX	Load IX onto stack
PUSH IY	Load IY onto stack
PUSH qq	Load Reg. pair qq onto stack
RES b, m	Reset Bit b of operand m
RET	Return from subroutine
RET cc	Return from subroutine if condition cc is true
RETI	Return from interrupt
RETN	Return from non maskable interrupt
RL m	Rotate left through carry operand m
RLA	Rotate left Acc. through carry
RLC (HL)	Rotate location (HL) left circular
RLC (IX+d)	Rotate location (IX+d) left circular
RLC (IY+d)	Rotate location (IY+d) left circular
RLC r	Rotate Reg. r left circular
RLCA	Rotate left circular Acc.
RLD	Rotate digit left and right between Acc. and location (HL)
RR m	Rotate right through carry operand m
RRA	Rotate right Acc. through carry
RRC m	Rotate operand m right circular
RRCA	Rotate right circular Acc.
RRD	Rotate digit right and left between Acc. and location (HL)
RST p	Restart to location p
SBC A, s	Subtract operand s from Acc. with carry
SBC HL, ss	Subtract Reg. pair ss from HL with carry
SCF	Set carry flag (C=1)
SET b, (HL)	Set Bit b of location (HL)
SET b, (IX+d)	Set Bit b of location (IX+d)
SET b, (IY+d)	Set Bit b of location (IY+d)
SET b, r	Set Bit b of Reg. r
SLA m	Shift operand m left arithmetic
SRA m	Shift operand m right arithmetic
SRL m	Shift operand m right logical
SUB s	Subtract operand s from Acc.
XOR s	Exclusive 'OR' operand s and Acc.

APPENDIX B

S-100 BUS SIGNALS

The following table summarizes the signals on the S-100 bus. Each signal is described by the trace number, its active state, the mnemonic used to reference it, the name, and finally its function. Pins 1 - 50 are on the front (component side) of the board; 51 - 100 are on the back (solder side).

Not all S-100 signals are used on the Z-64; those that are have an asterisk, '*', before the pin number. Regarding active state, the designation in this column describes the logical true state according to the following codes.

H = high true
L = low true

	Pin No.	Active State	Symbol	Name	Function
*	1		+8V	+8 volts	Unregulated input to +5 volt regulators
*	2		+16V	+16 volts	Positive unregulated voltage to +12V regulator
*	3	H	XRDY	External Ready	XRDY indicates that the I/O device is ready to send/receive data. A low means the device is not ready and a wait state is inserted into the machine cycle.
*	4	L	VI0	Vectored Interrupt line 0	
*	5	L	VI1	Vectored Interrupt line 1	
*	6	L	VI2	Vectored Interrupt line 2	
*	7	L	VI3	Vectored Interrupt line 3	
*	8	L	VI4	Vectored Interrupt line 4	
*	9	L	VI5	Vectored Interrupt line 5	
*	10	L	VI6	Vectored Interrupt line 6	

Pin No.	Active State	Symbol	Name	Function
* 11	L	VI7	Vectored Interrupt line 7	
* 12	L	/NMI	Non-Maskable interrupt	A low sensed on this line by the Z-80 at the rising edge of T4 will cause a restart at location 0066H
13	H	AA15		
14	H	AA14		
15	H	A18		
16	H	A16	Phantom line	Some systems use this as the phantom line. The Z-64 does not, pin 67 replaces it.
17	H	A17		
* 18	L	/STADSB	Status disable	Allows the buffer for the 8 status lines to be tri-stated (low = high impedance mode) See note 1 below
* 19	L	/CCDSB	Control disable	Allows the buffer for the 6 command/control lines to be tri-stated (again, low = high impedance mode). See note 1 below.
20	H	UNPROT	Memory unprotect	
21	L	/DIDSB	Data in Disable	
* 22	L	/ADDSB	Address bus disable	Allows the buffers for 16 address lines to be tri-stated. (Low = high impedance state) See note 1 below.
* 23	L	/DODSB	Data out Disable	Allows the buffer for the 8 data lines to be tri-stated (low = high impedance) See note 1.

NOTE 1: Present but not fully implemented on the Z-64.

Pin No.	Active State	Symbol	Name	Function
* 24		02	Phase 2 Clock	
* 25		01	Phase 1 Clock	
* 26	H	HLDA	Hold Acknowledge	Processor control output appearing in acknowledgment of an Hold request. From /BUSAK on Z-64. See note 1 above.
* 27	H	WAIT	Wait Acknowledge	Output to the bus, this signal indicates that the processor is in a wait state. This should not be confused with the Z-80 /WAIT input.
28	H	INTE	Interrupt Enable	
* 29		A5	Address Line 5	
* 30		A4	Address Line 4	
* 31		A3	Address Line 3	
* 32		A15	Address Line 15	
* 33		A12	Address Line 12	
* 34		A9	Address Line 9	
* 35		DO1	Data Out Line 1	
* 36		DO0	Data Out Line 0	
* 37		A10	Address Line 10	
* 38		DO4	Data Out Line 4	
* 39		DO5	Data Out Line 5	
* 40		DO6	Data Out Line 6	
* 41		DI2	Data In Line 2	
* 42		DI3	Data In Line 3	
* 43		DI7	Data In Line 7	
* 44	H	SM1	M1	Status output signal indicating the processor OP code fetch cycle.

Pin No.	Active State	Symbol	Name	Function
* 45	H	SOUT	I/O Out	Status output signal indicating the address bus contains the 8 bit address of an output device
* 46	H	SINP	I/O In	Status input signal indicating the address bus contains the 8 bit address of an input device.
* 47	H	SMEMR	Memory Read	Status output signal indicating that the data bus will be used for a memory read.
* 48	H	HLTA	Halt Acknowledge	Status output signal indicating a processor halt acknowledge
* 49	H	CLK	2 MHz clock	
* 50		GND	Ground	
* 51		+8V	+8 volts	Unregulated input to 5V regulators
* 52		-16V	-16 volts	Unregulated input to -12V or -5V regulators
53	L	/SWDSB	Sense Switch Disable	
54	L	/EXCLR	External Clear	
* 55 - 64 unused				
65		AA12		
66		AA13		
* 67	L	A19	/PHANTOM	The phantom line: 'turns off' system memory in deference to external memory device
* 68	H	SMWRT	Memory Write	Status output signal indicating the address bus contains a 16 bit address for a memory write
69	L	/PS	Protect Status	

Pin No.	Active State	Symbol	Name	Function
70	H	PROT	Protect	
71	H	RUN	Run	
* 72	H	PRDY	Ready	Processor command/control input that controls the run state of the processor
* 73	L	/PINT	Interrupt Request	Processor input from interrupt device. Low at T4 rising edge causes interrupt acknowledge cycle
* 74	L	/PHOLD	Hold Request	Processor command input signal requesting control of the address, data, and status/control buses by external device (/BUSRQ with Z-80)
* 75	L	/RST	Reset	Processor command input to reset the processor
* 76	H	/PSYNC	Sync	Processor command output indicating the beginning of each machine cycle
* 77	L	/WR	Write	Processor control output indicating a write operation
* 78	H	PDBIN	Data Bus In	Processor control output indicating the data bus is in the input mode
* 79		A0	Address Line 0	
* 80		A1	Address Line 1	
* 81		A2	Address Line 2	
* 82		A6	Address Line 6	
* 83		A7	Address Line 7	
* 84		A8	Address Line 8	
* 85		A13	Address Line 13	
* 86		A14	Address Line 14	

Pin No.	Active State	Symbol	Name	Function
* 87		A11	Address Line 11	
* 88		DO2	Data Out Line 2	
* 89		DO3	Data Out Line 3	
* 90		DO7	Data Out Line 7	
* 91		DI4	Data In Line 4	
* 92		DI5	Data In Line 5	
* 93		DI6	Data In Line 6	
* 94		DI1	Data In Line 1	
* 95		DI0	Data In Line 0	
* 96	H	SINTA	Interrupt Acknowledge	Status output signal acknowledging an interrupt request
* 97	L	/SWO	Write output	Status output signal indicating a write operation is to be performed
* 98	H	SSTACK	Stack	
* 99	L	/POC	Power on Clear	
*100		GND	Ground	

APPENDIX C

RAMTEST LISTING

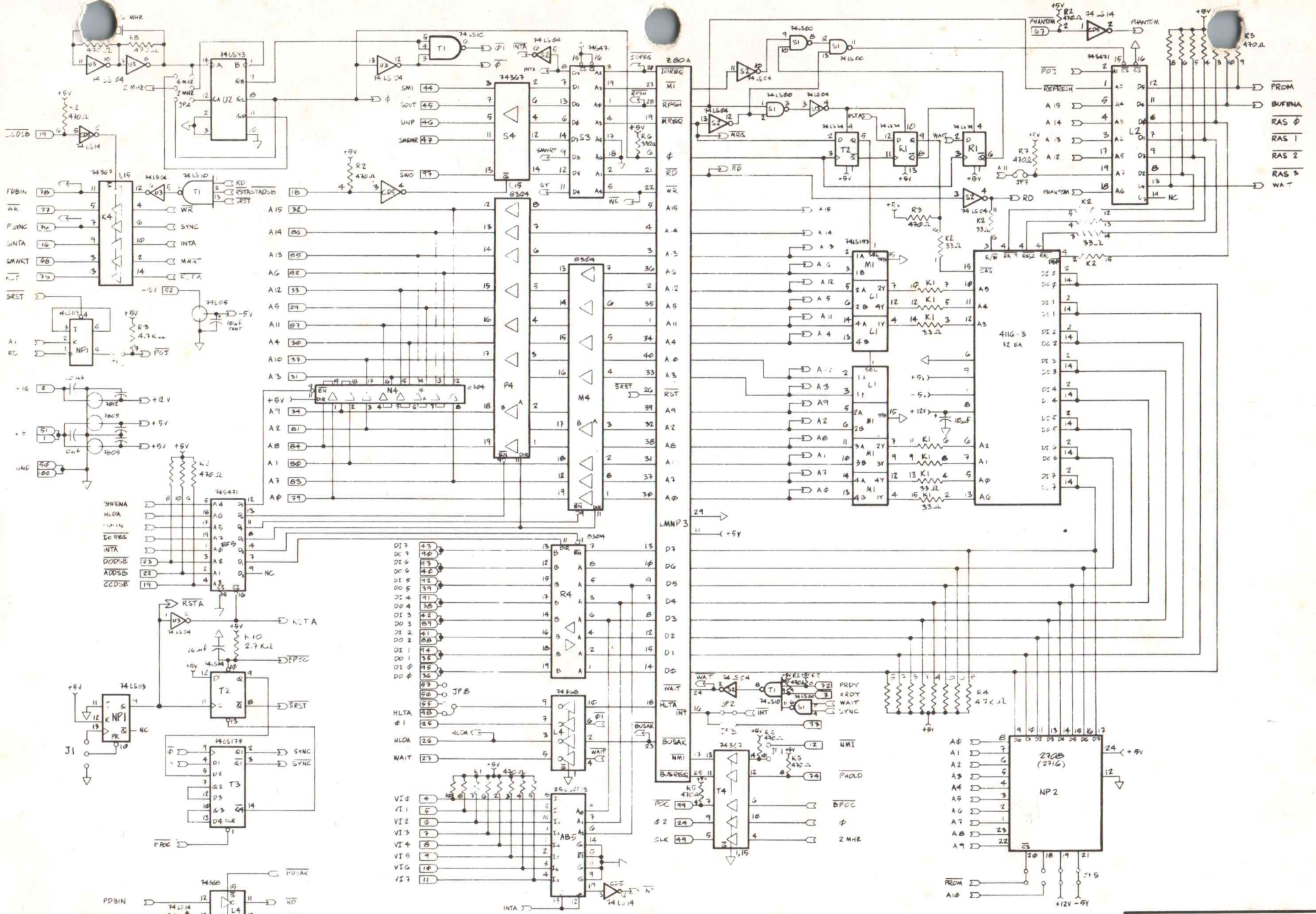
```
*****************************************************************************
*       ;TESTER TO TAKE CARE OF YOUR MEMORY WORRIES IN A FLASH              *
*       ;JUST ASSEMBLE FOR YOUR MEMORY SIZE AND DOUBLER LOCATION.           *
*       ;N.B.-THIS TEST WRITES OVER CP/M ETC.,SO PROGRAM MUST BE            *
*       ;TERMINATED WITH A COLDBOOT (^C) FROM KEYBOARD (OR RESET).          *
*       ;Last updated  November 15, 1979   by  M.GILLILAND                  *
*****************************************************************************
BASE    SET     BOTTOM-X        ;START ADDRESS OF RAM TO BE TESTED

ENDPAGE EQU     0F800H          ;SET THIS TO THE LAST PAGE + 1

DESTINY EQU     0000H           ;THE TEST PROGRAM WILL GET MOVED HERE

* Note: Be sure that the program is not within the test area.

HOLDOFF EQU     1000            ;NUMBER OF SYNC TRIES BETWEEN PATTERNS
                                ;IN AN ATTEMPT TO KILL MEMORY REFRESH
                                ;AT ABOUT 250 uS PER COUNT (4000 = 1 SEC)

DOUBLER EQU     0F800H          ;DOUBLER BASE ADDRESS
*****************************************************************************
BUFF    EQU     DOUBLER+400H    ;RAM BUFFER ON DOUBLER
COMMAND EQU     BUFF+3
WRCONT  EQU     DOUBLER+500H    ;HARDWARE PORTS ON DOUBLER
WRCLK   EQU     WRCONT+1        ;SYNC REGISTER
SYNCPRT EQU     WRCONT+7        ;SYNCPORT
URTSTAT EQU     WRCONT+0AH      ;UART STATUS PORT
URTDATA EQU     WRCONT+2H       ;UART DATA PORT
*****************************************************************************
        ORG     100H
        JMP     MOVER           ;MOVE TESTER TO LOW MEMORY

X       EQU     $-DESTINY       ;ADDRESS OFFSET FOR BLOCK MOVE

STARTAD LDA     COMMAND         ;GET THE DEFAULT DOUBLER COMMAND
        ORI     4               ;SET THE DENSITY BIT
        STA     WRCONT          ;SET THE DOUBLER PLO FOR SINGLE DENITY
        MVI     B,0H            ;CLEAR PATTERN MODIFIER IN B REG.
        LXI     D,0H            ;CLEAR PASS COUNTER
LOOP1   CALL    PASSMSG-X       ;OUTPUT PASS COUNT AND PATTERN TO CRT
LOOP    LXI     H,BASE          ;LOAD STARTING ADDRESS OF TEST
FILL    MOV     A,L             ;MOVE LOW ADDRESS INTO ACC.
        XRA     H               ;GENERATE TEST BYTE
        XRA     B               ;MODIFY WITH B REG.
        MOV     M,A             ;STORE TEST BYTE IN MEMORY CELL
        CALL    ENDTEST-X       ;NEXT ADDRESS TO DO...SETS ZERO FLAG IF DONE
        JNZ     FILL-X          ;IF NOT ZERO, CONTINUE TO GENERATE TEST BYTES
```

```
SYNC     LXI     H,WRCLK          ;NOW GET READY TO SYNC ON DISK(TEST REFRESH)
         MVI     M,70H            ;NONEXISTENT SYNC BYTE TO LATCH ON
         LXI     H,HOLDOFF        ;COUNT THE NUMBER OF SYNC TRIES
         INX     H                ;CORRECT THE COUNT
         PUSH    D                ;SAVE PASS COUNTER
         LXI     D,SYNCPRT        ;HARDWARE PORT ON DOUBLER
SYNLOOP  LDAX    D                ;SYNC ON UP,GUYS (0FFH CLOCK IN HEADER)
         DCX     H                ;DECREMENT SYNC COUNT
         MOV     A,H              ;GET THE HIGH ORDER COUNT
         ORA     A                ;COMPLETED NUMBER OF TRIES IN 'HOLDOFF'?
         JNZ     SYNLOOP-X        ;NOT YET,TRY AGAIN
VERIFY   MVI     E,3              ;LOAD"VERIFY LOOP"COUNT(TRY EACH CELL 3 TIMES)
TEST1    LXI     H,BASE           ;START ADDRESS OF MEMORY BLOCK TO TEST
TEST2    MOV     A,L              ;LOW BYTE USED TO GENERATE PATTERN
         XRA     H                ;GENERATE TEST BYTE
         XRA     B                ;MODIFY WITH CONTENTS OF B REG.
         MOV     C,M              ;GET MEMORY DATA BYTE
         CMP     C                ;COMPARE WITH MEMORY CELL CONTENTS (IN C REG.)
         CNZ     ERRMSG-X         ;IF NOT EQUAL,OUTPUT ADDRESS OF FAULTY CELL
REBOOT   CALL    CONSTAT-X        ;SHOULD TEST END??
         JZ      NEXT-X           ;IF ZERO FLAG,THEN NO CHARACTER: CONTINUE TEST
         CALL    CONIN-X          ;GET CHARACTER
         CPI     13H              ;IS IT A ^S ?
         CZ      CONIN-X          ;IF YES, ENTER WAIT LOOP (STOP SCREEN SCROLL)
         CPI     3H               ;IS IT A ^C ?
         JZ      EXIT-X           ;IF YES, REBOOT SYSTEM
NEXT     CALL    ENDTEST-X        ;NEXT ADDRESS TO DO...SETS ZERO FLAG IF DONE
         JNZ     TEST2-X          ;CONTINUE THIS TEST LOOP,IF NOT FINISHED.
         DCR     E                ;VERIFY LOOP-1 (CELL VALUES CHECKED AGAIN)
         JNZ     TEST1-X          ;REPEAT THIS VERIFY LOOP 5 TIMES
         POP     D                ;RESTORE PASS COUNTER IN DE
         MOV     A,B              ;HAS 1 PASS (256 PATTERN LOOPS) BEEN DONE?
         CPI     0FFH             ;
         CZ      UPDATE-X         ;UPDATE PASS COUNTER IF YES
         INR     B                ;OTHERWISE,GENERATE NEXT PATTERN MODIFIER....
         JMP     LOOP1-X          ;AND EXECUTE NEXT PATTERN LOOP
UPDATE   INX     D                ;PASS COUNT + ONE
         RET
ENDTEST  INX     H                ;NEXT ADDRESS TO DO
         MOV     A,H              ;GET PAGE ADDRESS
         CPI     ENDPAGE/100H     ;DONE???(SET ZERO FLAG IF YES)
         RET
*****************************************************************************
*                ;PRINT OUT ROUTINES FOR TESTER
*****************************************************************************
CONSTAT  LDA     URTSTAT          ;READ UART STATUS
         ANI     2                ;IS THERE A CHARACTER?
         RET                      ;RETURNS WITH ZERO SET IF NOT READY
CONIN    CALL    CONSTAT-X        ;
         JZ      CONIN-X          ;LOOP UNTIL INPUT FROM KEYBOARD
         LDA     URTDATA          ;INPUT CHARACTER
         ANI     5FH              ;STRIP PARITY BIT
         RET
```

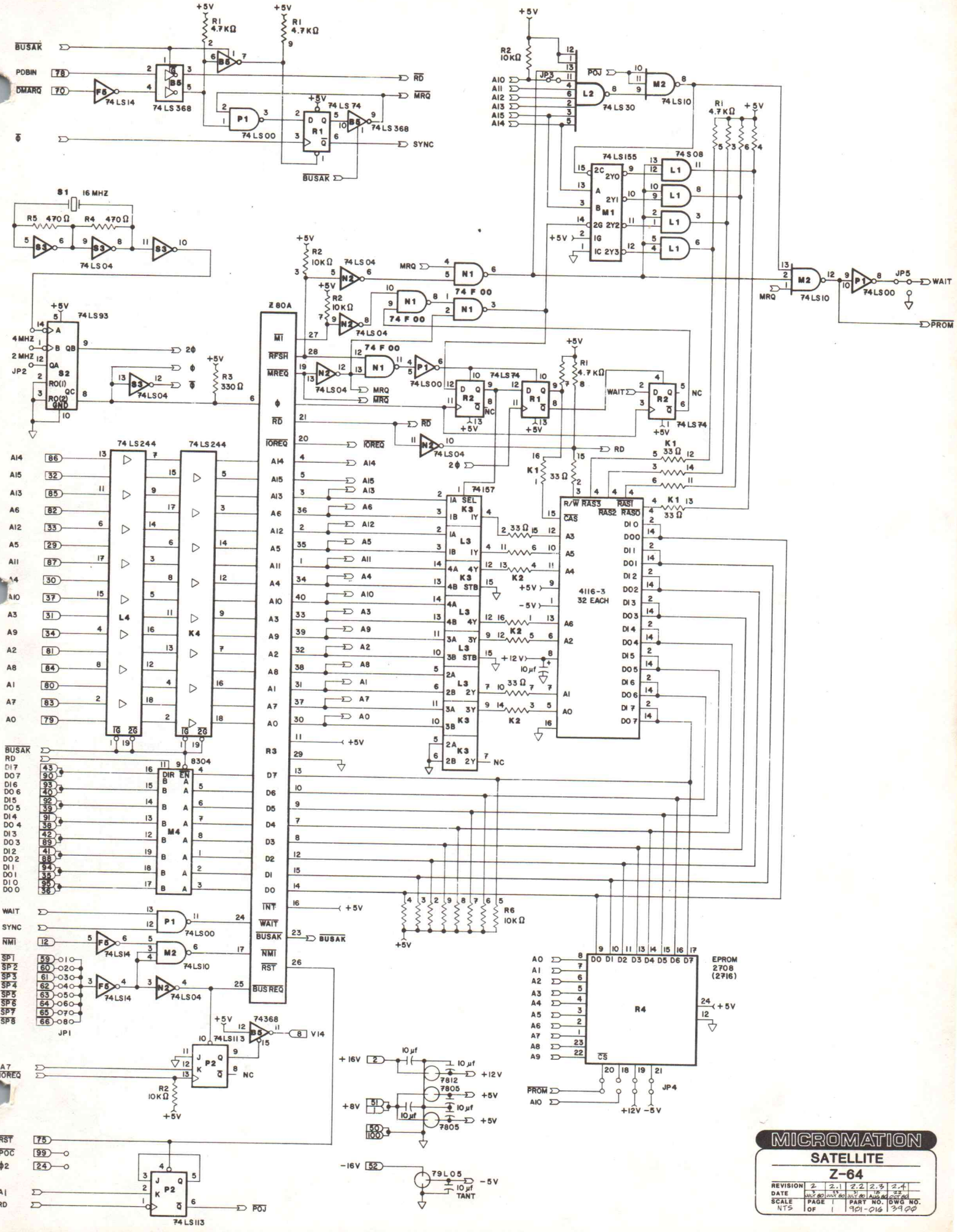
```
PRINT    MOV     A,M             ;GET CHARACTER IN MESSAGE STRING
         CPI     '$'             ;IS IT THE END OF THE STRING?
         RZ                      ;RETURN IF DONE
         MOV     C,A             ;PUT CHARACTER INTO C REG.
         CALL    CHAROUT-X       ;OUTPUT TO SCREEN
         INX     H               ;NEXT CHARACTER IN STRING
         JMP     PRINT-X         ;REPEAT THE LOOP
CHAROUT  LDA     URTSTAT         ;GET UART STATUS
         ANI     1               ;
         JZ      CHAROUT-X       ;WAIT FOR UART READY FLAG
         MOV     A,C             ;PUT OUTPUT CHARACTER IN ACC.
         STA     URTDATA         ;OUTPUT TO UART
         RET                     ;DONE
MOVHEX   MOV     A,D             ;HIGH BYTE
         CALL    PRNHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
         MOV     A,E             ;LOW BYTE
         CALL    PRNHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
         RET
PRNHEX   PUSH    PSW             ;SAVE FOR LSNIBBBLE CONVERSION
         RAR!RAR!RAR!RAR!        ;SHIFT MSN TO LSN
         CALL    SAVEHEX-X       ;CONVERT HEX TO ASCII
         POP     PSW             ;NOW DO THE LSN
SAVEHEX  ANI     0FH             ;
         ADI     30H             ;SHIFT TO ASCII VALUE
         CPI     3AH             ;OVER 9?
         JC      STORE-X         ;SAVE IN STRING FOR OUTPUT
         ADI     7               ;SHIFT TO ALPHA
STORE    MOV     M,A             ;STORE IN MSG STRING LOCATION
         INX     H               ;NEXT MSG LOCATION
         RET
PASSMSG  PUSH    H               ;SAVE CURRENT ADDRESS OF MEMORY CELL
         PUSH    D               ;SAVE PASS COUNT
         PUSH    B               ;SAVE PATTERN MODIFIER
         LXI     H,MSG1-X        ;LOCATION TO INSERT PASSCOUNT MESSAGE
         CALL    MOVHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE (2 BYTES)
         MOV     A,B             ;GET PATTERN-TYPE INTO ACC.
         LXI     H,MSG0-X        ;LOCATION TO INSERT PATTERN-TYPE MESSAGE
         CALL    PRNHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
         LXI     H,STRING1-X     ;PASS COUNT MESSAGE
         CALL    PRINT-X         ;OUT TO SCREEN
         POP     B               ;RESTORE STATUS
         POP     D               ;
         POP     H               ;
         RET
```

```
ERRMSG  PUSH    H               ;SAVE STATUS OF TEST
        PUSH    D               ;
        PUSH    B               ;
        XCHG                    ;PLACE TEST CELL PAGE VALUE IN D REG.
        LXI     H,MSG3-X        ;LOCATION TO INSERT PATTERN BYTE INTO STRING
        CALL    PRNHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
        MOV     A,C             ;GET MEMORY BYTE FOR DISPLAY (IN C REG.)
        LXI     H,MSG4-X        ;LOCATION TO INSERT MEMORY DATA INTO STRING
        CALL    PRNHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
        LXI     H,MSG2-X        ;LOCATION TO INSERT MEMORY ADDRESS INTO STRING
        CALL    MOVHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE (2 BYTES)
        LXI     H,STRING2-X     ;LOCATION OF ERROR STRING
        CALL    PRINT-X         ;OUTPUT ERROR MESSAGE TO SCREEN
        POP     B               ;RESTORE STATUS
        POP     D               ;
        POP     H               ;
        RET
EXIT    LXI     H,SIGNOFF-X     ;Load the mess.
        CALL    PRINT-X         ;Put it on the screen
        JMP     DOUBLER
*****************************************************************
*               ;MESSAGE AREA: STAY CLEAR                       *
*****************************************************************
CR      EQU     0DH
LF      EQU     0AH
STRING1 DB      CR
        DB      'MEMORY TEST: Pass #'
MSG1    DB      '0000 Hex; Pattern #'
MSG0    DB      '00 Hex$'
STRING2 DB      CR,LF
        DB      '** COMPARE ERROR ** Location:'
MSG2    DB      '0000 Hex, Pattern:'
MSG3    DB      '00 Hex, Memory Data:'
MSG4    DB      '00 Hex'
        DB      LF,'$'
SIGNOFF DB      CR,LF,'Insert system diskette in Drive A to coldboot$'

BOTTOM  EQU     $

SIGNMSG DB      CR,LF
        DB      'Welcome to the Memory Test for a DOUBLER system based at '
MSG5    DB      '0000 Hex'
        DB      CR,LF
        DB      'This version tests memory from '
MSG6    DB      '0000 Hex up to '
MSG7    DB      '0000 Hex.'
        DB      CR,LF
        DB      'Remove any diskettes and press <RETURN> to start the test'
        DB      CR,LF
        DB      'Press <^C> to terminate the test'
        DB      CR,LF
        DB      'Press <^S> to toggle the test ON/OFF'
        DB      CR,LF,'$'
```

```
*************************************************************************
        ;MOVER IS SOURCE FOR BLOCK MOVE TO PLACE TESTER AT DESTINY
*************************************************************************

MOVER:  LXI     H,STARTAD       ;BLOCK START
        LXI     B,DESTINY       ;BLOCK DESTINATION=0000H
MOVIT:  MOV     A,M             ;GET PROGRAM BYTE
        STAX    B               ;STORE AT NEW LOCATION
        INX     H               ;NEXT BYTE TO MOVIT
        INX     B               ;NEXT LOCATION TO INSERT DATA
        MOV     A,H             ;TEST FOR END OF BLOCK
        CPI     BOTTOM/100H + 1 ;
        JC      MOVIT           ;IF NOT DONE, CONTINUE
                                ;OTHERWISE EXECUTE SIGNON
        LXI     SP,BUFF+3FH     ;DEFINE STACK ADDRESS
SIGNON  LXI     D,DOUBLER       ;GET CONTROLLER ADDRESS
        LXI     H,MSG5          ;LOCATION FOR INSERT INTO SIGNON MSG
        CALL    MOVHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
        LXI     D,BASE          ;GET MEMORY START ADDRESS
        LXI     H,MSG6          ;LOCATION TO INSERT INTO SIGNON
        CALL    MOVHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
        LXI     D,ENDPAGE       ;GET MEMORY SIZE
        LXI     H,MSG7          ;LOCATION TO INSERT INTO SIGNON
        CALL    MOVHEX-X        ;HEX TO ASCII CONVERSION AND STORAGE
        LXI     H,SIGNMSG       ;ADDRESS OF MESSAGE STRING FOR SIGNON
        CALL    PRINT-X         ;OUTPUT ROUTINE
GO      CALL    CONIN-X         ;WAIT FOR KEY ENTRY (USER RESPONSE)
        CPI     3               ;IS IT ^C (REBOOT)???
        JZ      EXIT-X          ;IF YES,THEN REBOOT
        CPI     0DH             ;IS IT CR?
        JNZ     GO              ;IF NO, WAIT FOR CR
        JMP     DESTINY         ;IF YES, EXECUTE TESTER

        END     100H
```



MICROMATION						
CPU						
PROTO MASTER						
Revision	2	3	4	5	7	
Date	15 OCT 81	31 OCT 81	9 NOV 81	JAN 82	28 FEB 82	
DWG. NO: 3701	page 1 of 1					


MICROMATION
SATELLITE
Z-64
REVISION 2 2.1 2.2 2.3 2.4
DATE JULY 80 MAY 80 JULY 80 AUG 80 OCT 80
SCALE NTS
PAGE 1 OF 1
PART NO. 901-016
DWG NO. 3900
Z 80A
74 LS 368
74 LS 74
74 LS 00
74 LS 14
74 LS 04
74 LS 10
74 LS 30
74 LS 08
74 LS 155
74 LS 93
74 LS 244
74 LS 157
74 F 00
74 LS 113
74 S 08
8304
74368
4116-3
32 EACH
EPROM
2708
(2716)
16 MHZ
4.7 KΩ
10 KΩ
470 Ω
330 Ω
33 Ω
BUSAK
PDBIN
DMARQ
RD
MRQ
SYNC
WAIT
PROM
NMI
RST
POC
BUSREQ
IORQ
INT
JP1
JP2
JP3
JP4
JP5
+5V
+12V
-5V
+16V
-16V
10 µf
TANT
7812
7805
79L05