MICROMATION

MICROSAT

SATELLITE PROCESSOR BOARD

THEORY OF OPERATION

D01 016 Rev 1

FOREWORD

The MicroSat Satellite processor board is an integral component of Micromation's M/NET Multi-user, Multi-processor System. Conceptually, M/NET is a new implementation of the traditional network approach to data processing. That is, within an M/NET system each operator has a separate computer within the computer at his or her disposal. Thus, programs run independently of eachother, and operators don't suffer the run-time performance degradation common in single processor, multi-user systems -- regardless of the number of users on-line.

Within the M/NET network, each operator is allocated a MicroSat card. On this S-100 type board are a Z80A (running at 4 MHz) and 64K bytes of dynamic RAM (200 ns access time) with transparent refresh. For each operator, a separate MicroSat card is installed. Consistent with the networking concept, a master processor/RAM card, Micromation's Z-64 board (also with a Z80A and 64K of dynamic RAM), handles system I/O chores.

Unlike the traditional network approach where each workstation contains its own intelligence and memory, however, M/NET systems have all processing performed in a single cabinet. The master Z-64 talks with the independent MicroSats via the linking S-100 bus. Also installed within the card cage in a typical M/NET system are the DOUBLER Floppy Disk Controller, M/NET and Multi I/O cards, and Micromation Hard Disk Controller.

There are several advantages to this architecture.

- Since all components (boards and shared peripherals) are centrally located, system upgrading and servicing are performed quickly and easily with a minimum of down time;

- The cost of upgrading is reduced: installing a new workstation requires just a terminal and MicroSat card;

- No special communication protocols need be accommodated; all intra-system data transfers are conducted via the S-100 bus at processor speed.

M/NET systems are available with two operating systems: MP/M from Digital Research or DBOS from TCS, Inc. Both provide CP/M compatibility in a multi-user, multi-processor environment.

TABLE OF CONTENTS

HOW TO FIND INFORMATION QUICKLY

This manual is intended to provide the MicroSat Satellite Processor Board theory of operation for the system integrators and service technicians. Toward this end, it also includes a basic description of the M/NET multi-processor system -- the environment in which the MicroSat is an essential component. This manual does not describe installation of the MicroSat card nor the diagnostic software provided by Micromation. Descriptions of these topics is provided in the M/NET Installation Guide.

The Table of Contents in this manual is arranged in a conceptual top-down order -- more specifically, a hierarchy of systems. M/NET is the highest system explained and digital circuits are the lowest sub-systems explained. Between the highest and lowest system explained is a "center of attention". In this manual, the center of attention is in the form of schematic segments placed at the beginning of each heading.

The information in this manual is arranged into sections, headings and appendixes. Each heading covers about 1-2 pages and has a preceding diagram relevant to the text. Lettered appendices hold loosely organized information, bulky data or other material of peripheral interest which does not fit the flow of headings.

Within each heading the level of explanation can be primary, secondary or tertiary. A primary explanation is very general -- "a white automobile." A secondary explanation includes the most important details -- "a white, 1963 Chevrolet, 2-door sedan." A tertiary explanation contains extensive detail -- "a chalk white, 1963 Super Sport with Posi-traction differential, 350 engine with custom dual carbs, and Michelin radials." Both secondary and tertiary levels have some nested information and gaps to be filled from common knowledge; it is assumed that you know that a Super Sport was a model of Chevrolet, carbs is a abbreviation, and that radials are tires.

In a learning mode, the manual is read through entirely. Subsequently, the reader frequently finds necessary to reference only a certain function or segment of the board. In this case, reference to Figure II - 4 and the Table of Contents indicates the pertinent circuitry and location of the explanation.

We would appreciate your comments regarding the format and contents of this manual. Please note any mistakes you've found in the text and those areas that are confusing. A tear-out Reader Comments sheet is included at the end of this manual for your convenience. All comments will be carefully considered for future revisions of this manual.

SECTION I

THE M/NET SYSTEM

In this section diagrams and text synthesize the MicroSat satellite processor board into the larger M/NET system.

1) M/NET SYSTEM DESCRIPTION

The M/NET multi-processor concept provides a separate MicroSat satellite processor board for each user -- a separate computer within the computer. Each operator has the power of a Z80A and 64K of memory at their disposal for program execution. Thus, the common occurrence of run-time performance degradation suffered in single processor multi-user systems is alleviated, regardless of the number of active users. Those familiar with sophisticated multi-user systems will recognize this architecture as a computer network.

There are several essential components of this scheme:

- A master processor to control access to the shared, and expensive, peripherals;
- Satellite processors, typically one for each operator;
- Peripheral controllers for the floppy disk drive(s), hard disk drive, streaming tape cartridge (to back-up the hard disk quickly and efficiently) and printers; and
- The operating system resident in the master only to coordinate system activities.

Figure I - 1 illustrates these components and their relationships.

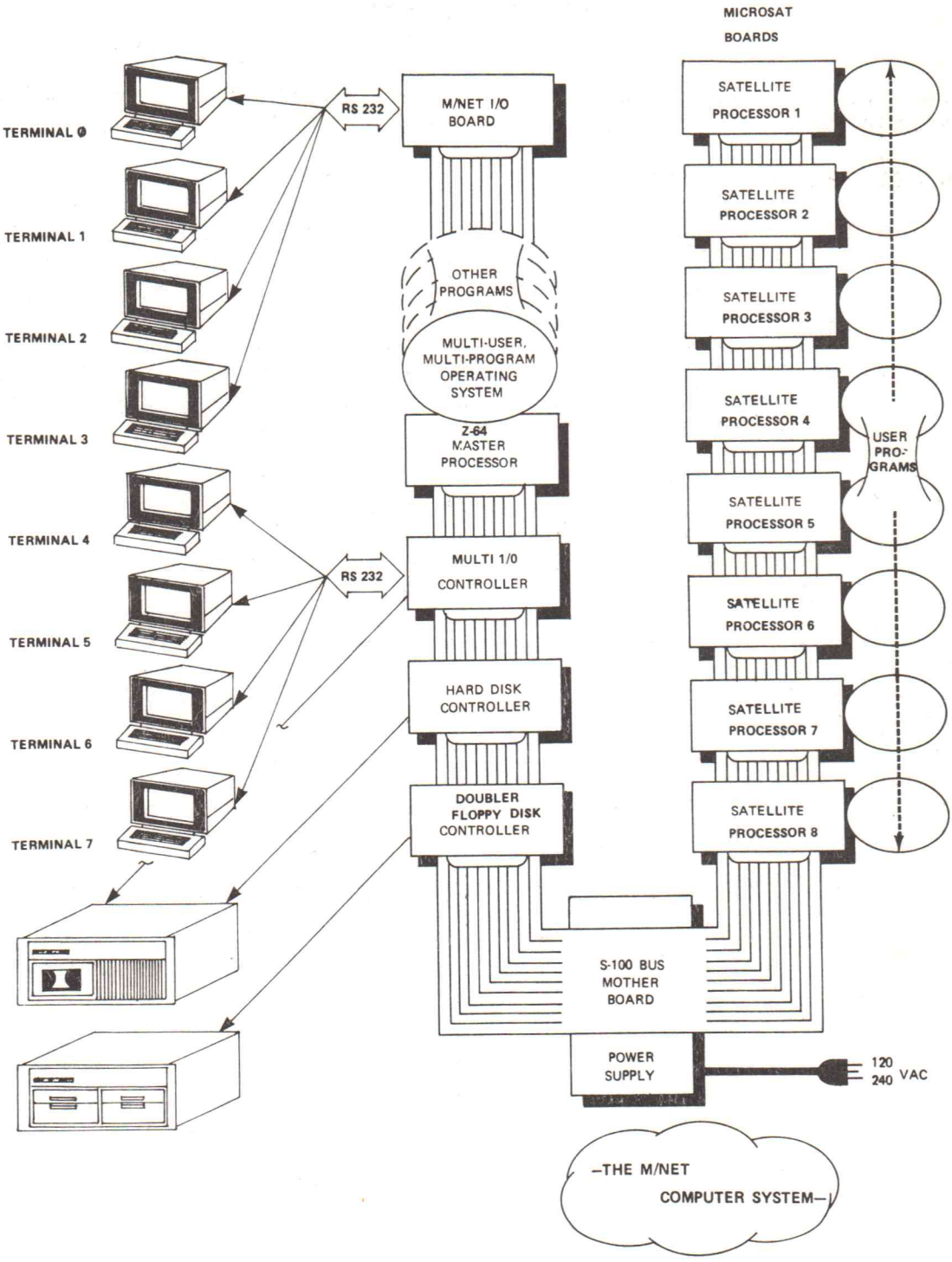


FIGURE I - 1 The M/NET Computer System

A single computer cabinet houses all the board level components in the M/NET system. A S-100 type bus links these boards together for fast inter-card communication.

2) M/NET SYSTEM COMPONENTS

MASTER PROCESSOR

Micromation's Z-64 board acts as the master processor in the M/NET network. The board has a Z80A (running at 4 MHz) and 64K of dynamic RAM (200 ns access time, transparent refresh). Upon a system reset, the multi-user operating system is loaded into the master. From this central focus, satellite allocation and access to the shared peripherals is coordinated. Regarding this latter function: a single system component, the master processor, reconciles potentially conflicting requests to access the peripherals in the network concept. The operating system within the master maintains the appropriate controls and queues to prevent these damaging occurrences. To maximize the master's finite processing capability, the Z-64 is fully interrupt driven.

SATELLITE PROCESSORS

In the network concept, each operator is allocated a separate and independent satellite processor. In the M/NET implementation, this is the MicroSat Satellite Processor Board. Like the Z-64, each MicroSat has a Z80A (running at 4 MHz) and 64K of dynamic RAM (200 ns access time, transparent refresh) on-board. Note that MicroSat-to-user allocation is completely transparent to the operator.

PERIPHERAL CONTROLLERS

A full complement of mass storage peripherals are available for M/NET computers. Separate boards control access to floppy disk drives, hard disk drive, and MicroTape streaming tape drive.

DOUBLER: Micromation's DOUBLER can control up to 4 floppy disk drives. The drives can be single or double sided; the recording formats can be IBM 3740 standard single density or modified IBM 2-D double density. For operator convenience, the DOUBLER feature automatic density compensation: the recording density is automatically determined by the controller without any user intervention.

Hard Disk Controller: Multi-user systems typically need expanded file storage space and faster data transfer than that provided by floppy disk drives. The Micromation Hard Disk Controller provides over 20 megabytes for file storage and transfers data more than 8 times faster than floppies.

Multi I/O: The Hard disk can be backed up to a large group of floppy diskettes or a single streaming tape cartridge. Micromation's MicroTape drive provides the capability to transfer the entire 20+ Mbyte contents of the hard disk to/from a cartridge in a short 14 minutes. This is controlled by the operating system through the Multi I/O. In addition, this board has 4 serial ports for console or printer I/O and a Centronics-type printer interface.

OPERATING SYSTEM

System integrators have a choice of operating systems for the M/NET computers: Digital Research's MP/M and TCS's DBOS. Both maintain CP/M compatibility so that the entire library of application software based upon this popular and familiar operating system can be easily installed.

MP/M: The latest version of the MP/M multi-user, multi-tasking operating system has been enhanced for use in M/NET's multi-processor environment. The significant enhancement is the switch from a single processor design (where each operator relies upon one CPU for all processing and system control functions) to a multi-processor architecture. The remainder of MP/M's structure has been maintained to ensure CP/M compatibility.

DBOS: TCS's Data Base Operating System applies data base management concepts to file management. Thus, data AND program file access is performed with the exceptional speed associated with this type of data management facility. In addition, this operating system was designed from the beginning around M/NET's multi-processor architecture and for CP/M compatibility.

M/NET CONSOLE I/O OPTIONS

**** IMPORTANT ****

MicroSat console I/O control described in the following paragraphs has not been implemented as of this writing. The following description is provided in anticipation of this event.

****

M/NET system console I/O handling is available in two forms: A) console I/O for all users is processed by the system master, or B) each MicroSat controls its own console I/O through the USART mounted on-board.

A) Master Control of Console I/O

When the system master processor card controls console I/O (through the M/NET and Multi I/O boards) there is no hard assignment of MicroSat card to a specific console. This configuration systems provides dynamic satellite allocation; that is, an operator is assigned to any of the available satellite processors for program execution. Another significant feature of this configuration is that an operator has access to all available MicroSats, even all 8 if they are not already active. The operator can, thus, run several programs at once from a single console.

B) MicroSat Console and Printer I/O Control:

Optionally (and this must be specified at the time of purchase -- the operating system is different) the on-board USART can be used for direct console communication. This configuration relieves the master of console control responsibilities and is pertinent for those who won't require the multi-tasking capability. Note that, in this configuration, one MicroSat is always assigned to a specific console.

NOTE: In DBOS based systems, console I/O is always controlled by the master through the M/NET and Multi I/O. The MicroSat's USART is available for non-system printer or secondary console I/O. Back-ground processing for multi-tasking is provided by installing more MicroSat cards in the system than there are consoles.

3) HOW IT WORKS

The Z-64 M/NET master communicates with all MicroSat cards via the S-100 bus. Each satellite is uniquely identified by a jumper indicating the MicroSat number (1 - 8). The Z80A block move command is used to move parallel bytes of data between the two boards at processor speed. The rate of transfer is 181K bytes per second. The master Z-64 controls data transfers in both directions; except for an interrupt and wait signal generated by the MicroSat card, it resides passively on the bus.

Master/satellite communication is conducted on two levels: 1) message transfers between master and satellite pass messages and 2) data transfers between these two boards. The two are related; the messages indicate the nature of the data transfer (e.g., floppy or hard disk read or write, console read or write, etc.). The messages are passed to/from the satellite via a window or "mailbox:" a reserved portion of the satellite's scratch RAM space.

To control satellite program processing, a control program -- a CP/M emulator -- is loaded into the top of the satellite's memory space by the master when the system is reset. This program maintains the MicroSat in idle until a process (program) has been assigned to it. The master transfers the transient program into the satellite's memory (as described below) and the control program relinquishes control to this program. Whenever a CP/M BDOS call is made by the transient software, it is trapped by the program and reduced to a message/command for execution by the master.

The emulator also enables an interrupt line on the system bus that indicates to the master that a satellite requires some action. The master then polls the satellite boards (reads the mailboxes) to determine which one called. The message is read and processed. Return messages (as opposed to program data) is transferred back to the mailboxes by the master.

When data is to be transferred, the master enables the satellite processor signal (SPn where n = 1 to 8) via the M/NET I/O card. Unless this signal is active, the MicroSat is locked off the system bus; in fact, the system doesn't know it exists until data transfers are requested. (This is true whenever the satellite is in idle mode or is executing a program.)

Active SPn puts the MicroSat's Z80A on hold (via the BUSREQ input). At this time, memory accesses can be controlled through the system bus but not by the resident Z80A. That is, only the Z80A on the master can access the satellite's 64K memory block.

However, the master also has a 64K memory block of its own. To prevent conflicting responses to the same memory access, the PHANTOM line is used. For instance, when a byte is to be transferred from master to satellite, PHANTOM is inactive when the byte is taken from the master's memory location then activated during that part of the cycle wherein the byte is placed in the MicroSat's memory. Thus the master's Z80A only sees a single 64K block of memory despite the fact that it is accessing two parallel blocks.

Data transfers from satellite to master are conducted in a similar manner. The exception is that PHANTOM is active during the first part of the cycle, when the data byte is taken from the satellite's memory block, and inactive during the latter part when the byte is being placed in the master's memory.

NOTE: The description above applies to MP/M based M/NET systems. Under DBOS, data transfers are performed in a similar with two differences: 1) The control program area in the MicroSat's memory includes a 1K window/buffer for data transfers; and 2) the master polls the satellites for I/O requests at regular intervals instead of relying upon the interrupt.

**** IMPORTANT ****

Because of these operational differences, the MicroSat is configured differently for DBOS based systems than for MP/M based systems. Each card is factory configured for the operating system specified on the purchase order; no reconfiguation is necessary

****

NOTES

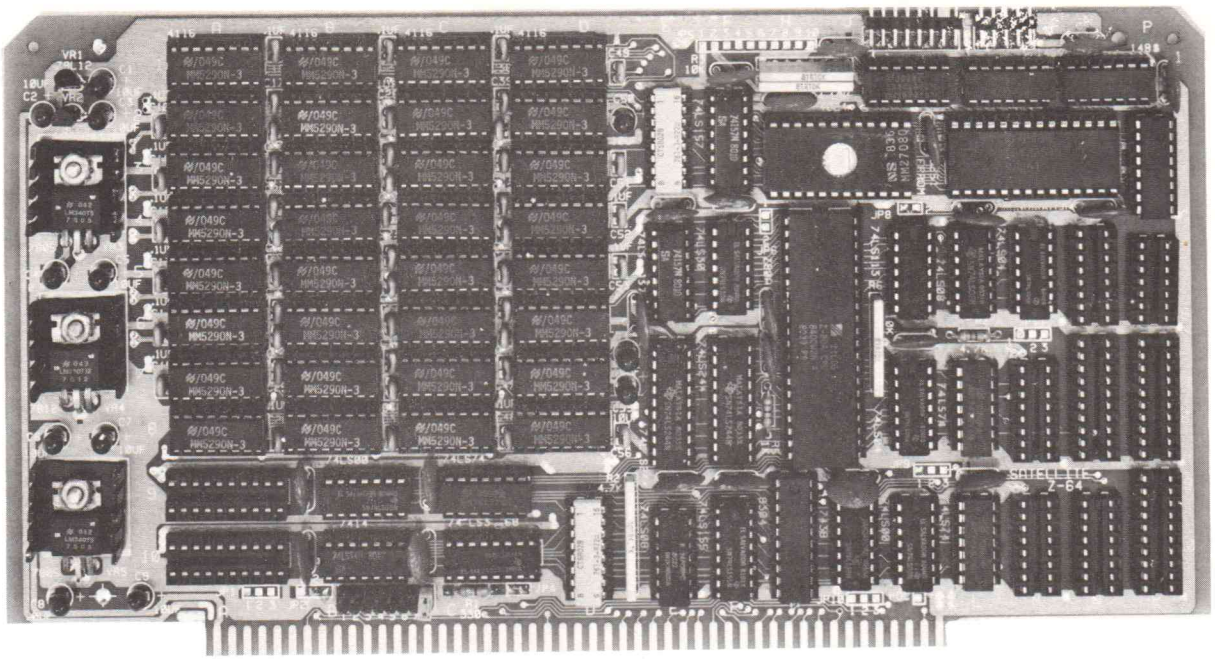


FIGURE II - 1: MicroSat Satellite Processor Board.

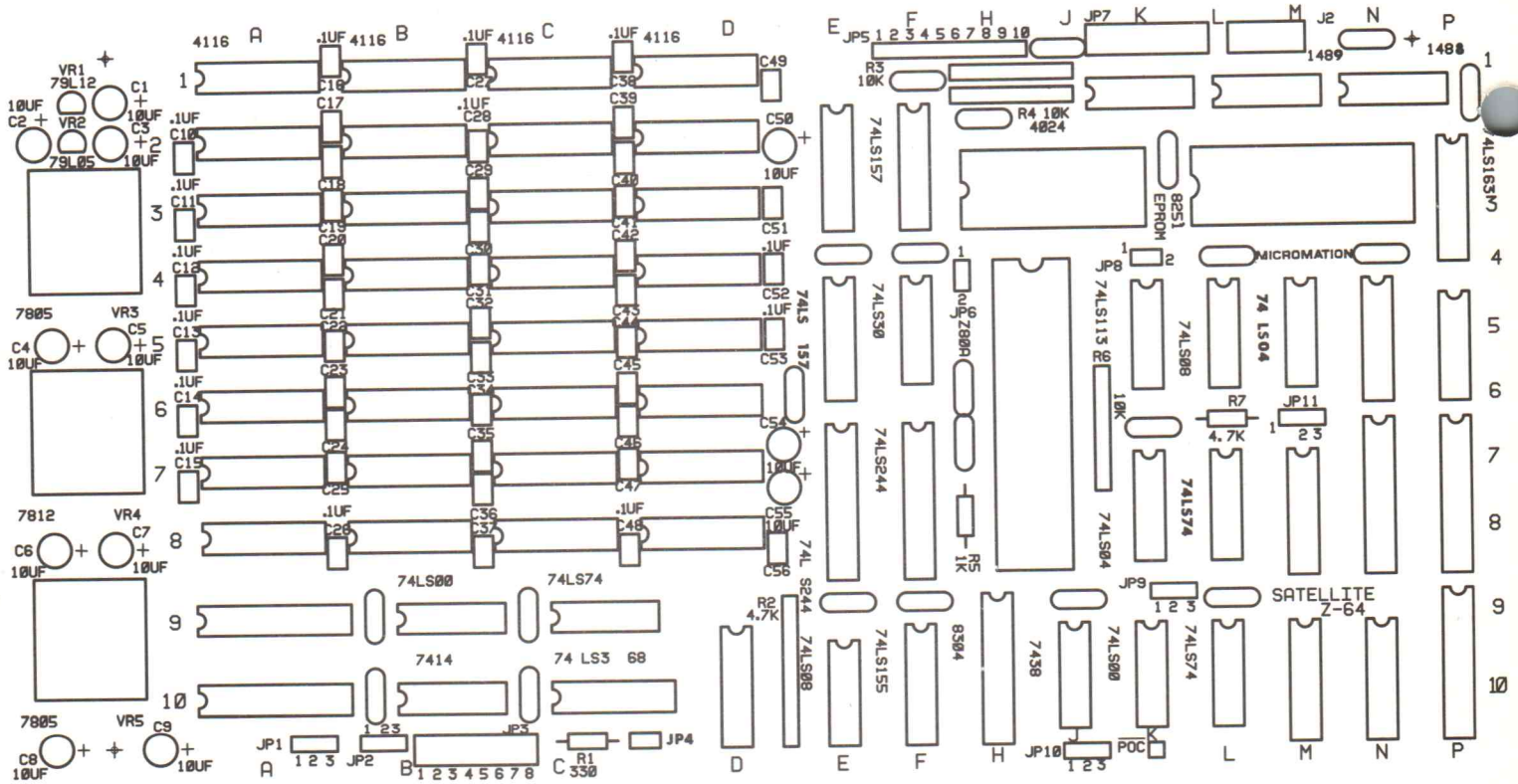


FIGURE II - 2 MicroSat Board Silkscreen

Photograph and silkscreen of the MicroSat Satellite Processor card. Use these figures for board and component location identification.

SECTION II

MICROSAT THEORY OF OPERATION

This section contains two chapters. Chapter 1 provides a functional description of the MicroSat Satellite Processor Board. Diagrams and text break-down the MicroSat board into its functional blocks and their logic circuits and components. Figures II - 1 and II - 2 are a photo of the MicroSat and the silkscreen for board and component location identification, respectively. Simplified diagrams (Figures II - 3 and II - 4) illustrate the logic flow on the MicroSat and the relationship of the headings in Chapter 2 to this functional description.

After understanding the block diagrams at the operations level and a reading of the partial schematics and text at the component level, much servicing can be accomplished with the segmented schematic and location diagrams alone. When a specific board functional segment needs to be identified and the description found, use these diagrams to assist you in your search. Additional assistance may be found in Figure II - 5 at the beginning of Chapter 2, a segmented schematic which indicates the associated subsection number.

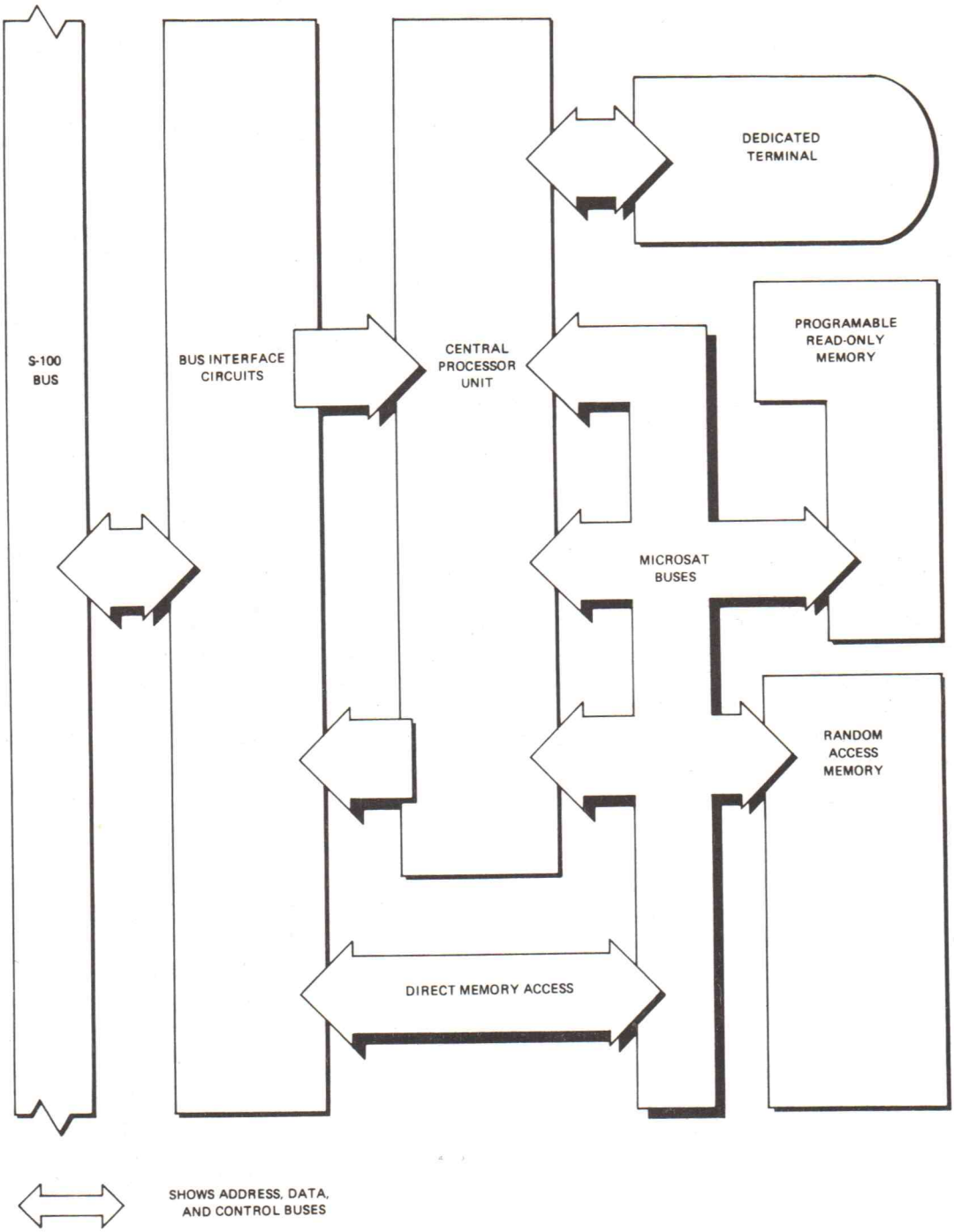


FIGURE II - 3 MicroSat Block Diagram

The MicroSat is connected to the system via the S-100 Bus. Bus interface logic prevents interference from external signals but allows master processor access to satellite RAM. Each MicroSat operates in an M/NET system as an independent computer when locked off the bus.

CHAPTER 1

MICROSAT FUNCTIONAL DESCRIPTION

Figure II - 3 is a simplified block diagram of the MicroSat board. Note that each large block can be represented by several schematic segments (see Figure II - 4).

Regarding the blocks:

The S-100 bus is the link to the rest of the computer system. It carries control, status and data signals between not only the master and satellite but the other system boards (primarily peripheral controllers) as well. To prevent signal conflicts, the master processor controls all bus activity.

The bus interface circuits manage information transfers between master and satellite. This block is controlled by the master. That is, the master signals that it would like to open up the satellite's memory. The bus interface logic responds by enabling the data and address signal driver/buffers for direct memory access by the system master CPU.

The remainder of the functional blocks (CPU, RAM, PROM, and USART) operate as an independent computer whenever the master has not interrupted their operation.

The **Programmable Read Only Memory** is not used in M/NET systems. The socket and supporting select logic are present for a test PROM.

The **Random Access Memory** is split into two segments. A small portion is reserved as scratch RAM. One of the functions of this area is to provide a window or mailbox for passing messages to master and getting responses back. The remainder of the RAM, over 62K, is the transient program area (with a base address of 00 hex). User's programs run here.

A USART on the MicroSat card is optionally available for a **Dedicated Terminal.** I/O mapped at location 40 hex, this device is accessed by the satellite CPU only. The system master CPU does not see this port.

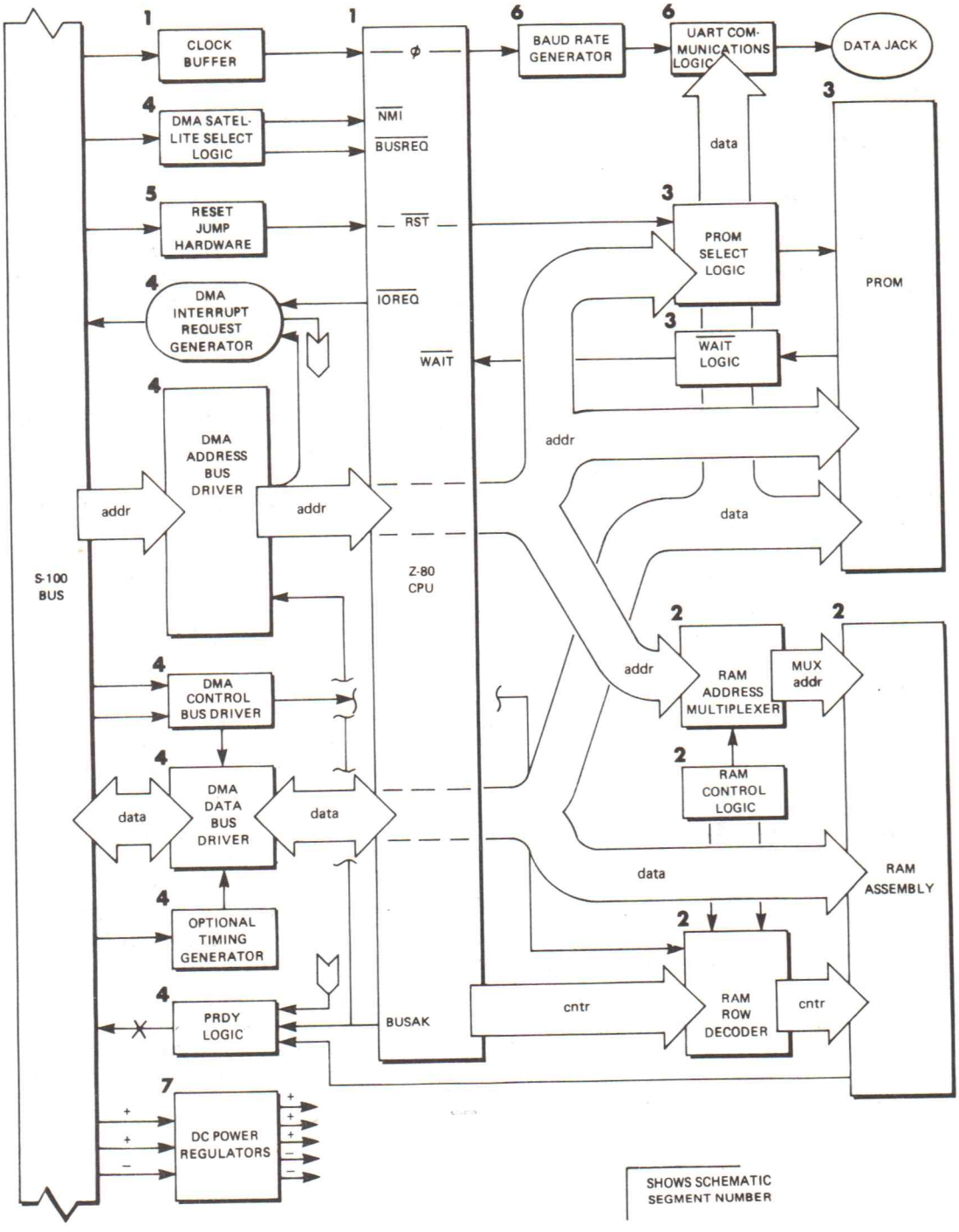


FIGURE II - 4 MicroSat Extended Block Diagram

Each block shown in Figure II - 3 can be broken down into contributing logic segments. This figure relates the functional blocks to the specific circuit segments described in Chapter 2.

In Figure II - 4, each numbered block is the equivalent of one of the schematic segments in Chapter 2 that describe this board on the component and signal level. Note also the functional correspondence of groups of these blocks to the larger blocks on the simplified block diagram (Figure II - 3).

Detailed communication is transmitted between master and satellite via three categories of signals. These are lumped into three subdivisions of the S-100 bus: the address bus, the data bus and control/status signal bus. When interpreting this figure, note that all signals other than those labelled "addr" and "data" should be considered part of the control/status bus. In addition there is a power bus which is usually not given schematic consideration because of its limited extent and simplicity.

The address bus is composed of 16 lines labelled A0 - A15. The data bus consists of two parallel 8-bit groups: DI0 - DI7 for data input to the master CPU and DO0 - DO7 for data output by the master CPU. The remainder of the signals on the S-100 bus fall into the control/status category. Refer to Appendix A if you are not familiar with the S-100 bus and the signal definitions.

Regarding the data and address sub-buses: Notice in this figure that the MicroSat internal data and address buses are used by two devices: the master CPU and the resident CPU. Since only one of these can have access at a given moment, the DMA Address and Data Bus Drivers (enabled by a signal from the Z80A) prevent contention.

NOTE: This figure includes the PROM reference. This is not utilized in M/NET systems except for factory testing.

CHAPTER 2

MICROSAT LOGIC SEGMENT DESCRIPTIONS

Figure II - 5 symbolically divides the satellite hardware into groups of circuits which are both visually obvious and functionally specific, within reason. Its main purpose is to be an index to quickly guide the reader to specific operational information about any small part of the hardware. The following example will illustrate its use.

EXAMPLE:

Suppose that your are a service person and you are investigating a problem which caused a system boot failure only when a particular satellite board was installed. You further discover that one of the SP lines on the S-100 bus has a short when the suspect board is installed, and you conclude that there is something defective on the satellite very close to the bus.

You examine Figure II - 3 and decide that the block labeled Bus Interface Circuits fits the description of what you want to examine. You examine Figure II - 4, and locate the schematic segments that relate to the S-100 bus. Each segment has a number which keys it to the segmented schematic. You now examine Figure II - 5, and locate the SP group of lines in Segment 4.

You turn to the heading for Segment 4 where there is a corresponding schematic segment number. By examining the partial schematic and text at this heading, you learn that jumper JP3 sets the satellite identification number, and that only one terminal pair is ever connected at a time. Since the connected terminal has the same number as the shorted SP line, you think inverter IC B10 may have an internal short.

You now look at Figure II - 2, the silkscreen location diagram, and find B-10 by looking at column B and row 10. The IC at this physical location is a 7414, which matches the number on the schematic. When you replace this IC the problem clears.

The table on the following page is provided for your convenience. It lists segment numbers, descriptions and their page numbers.

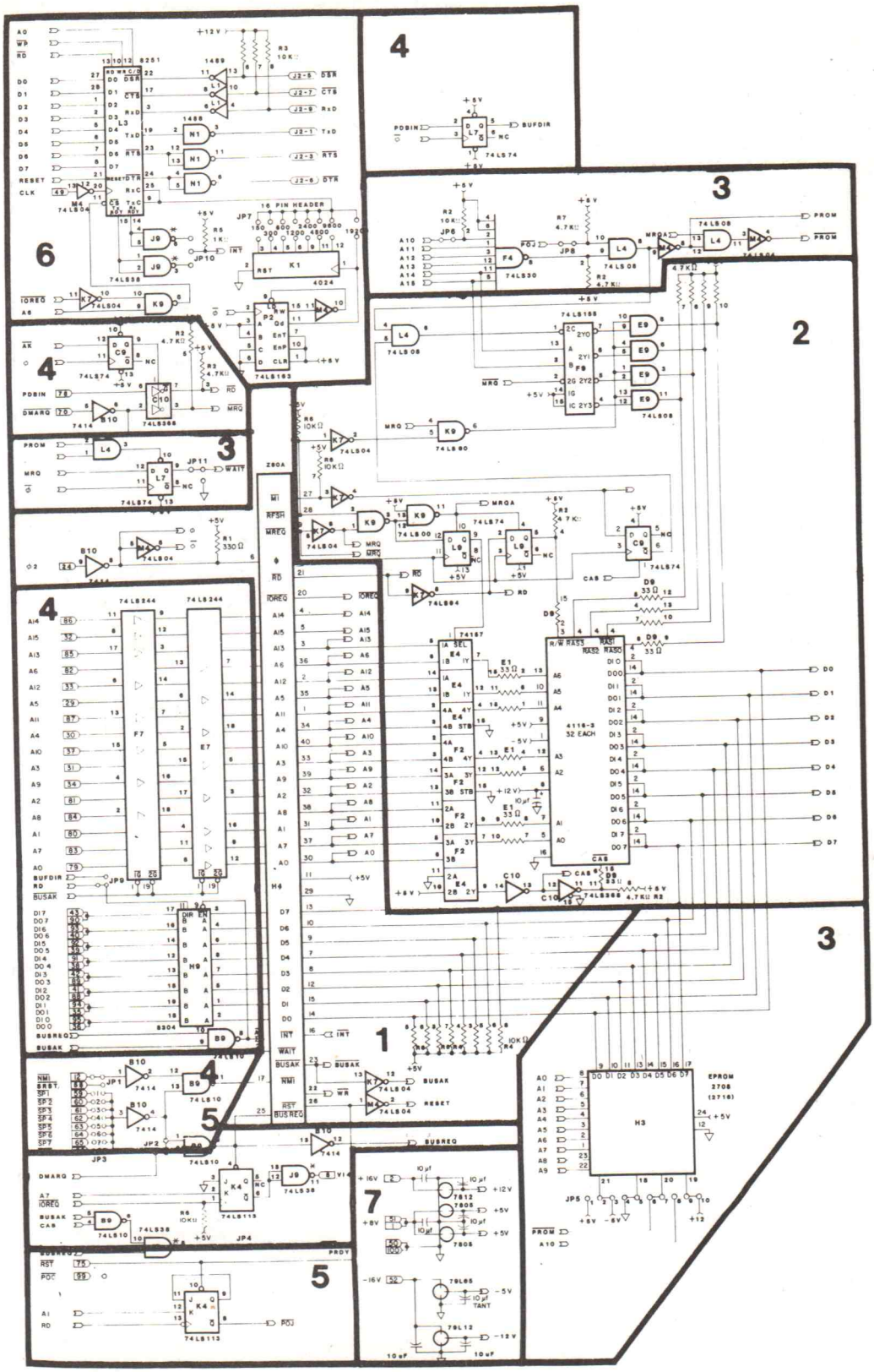


Fig II - 5 Segmented MicroSat Schematic

SCHEMATIC SEGMENT PAGE REFERENCE

SEGMENT	NAME	PAGE
1	Z80A CPU	
2	RAM Address Decoding, Enable and Access	
3	PROM Select Logic	
4	DMA Data Transfer Logic	
5	MicroSat Reset and Initialization Logic	
6	USART	
7	Power	

Note that when appropriate many of these segments are broken down further into smaller functional blocks.

Regarding signal and device location references on the schematic and within the text:

Signals can be either low or high active. Those that are low active have a preceding / in the descriptions that follow. On the schematic, these signals have a bar over them to indicate that they are low active. High active is the default designation. If there is no preceding / or bar over the name, the signal can be assumed to be high active.

Device locations are expressed by their column and row on the board. Columns are identified by capital letters and rows are identified by numbers on both the schematic and in the text. For example N1 indicates the device in column N, row 1. In the text, specific device I/O signals are indicated by their pin number in the form "cr-p" where "cr" are the device column and row designators, respectively, and "p" is the pin number. For instance, N1-3 references pin 3 of the device in column N, row 1.

SCHEMATIC SEGMENT 1: Z80A MICROPROCESSOR

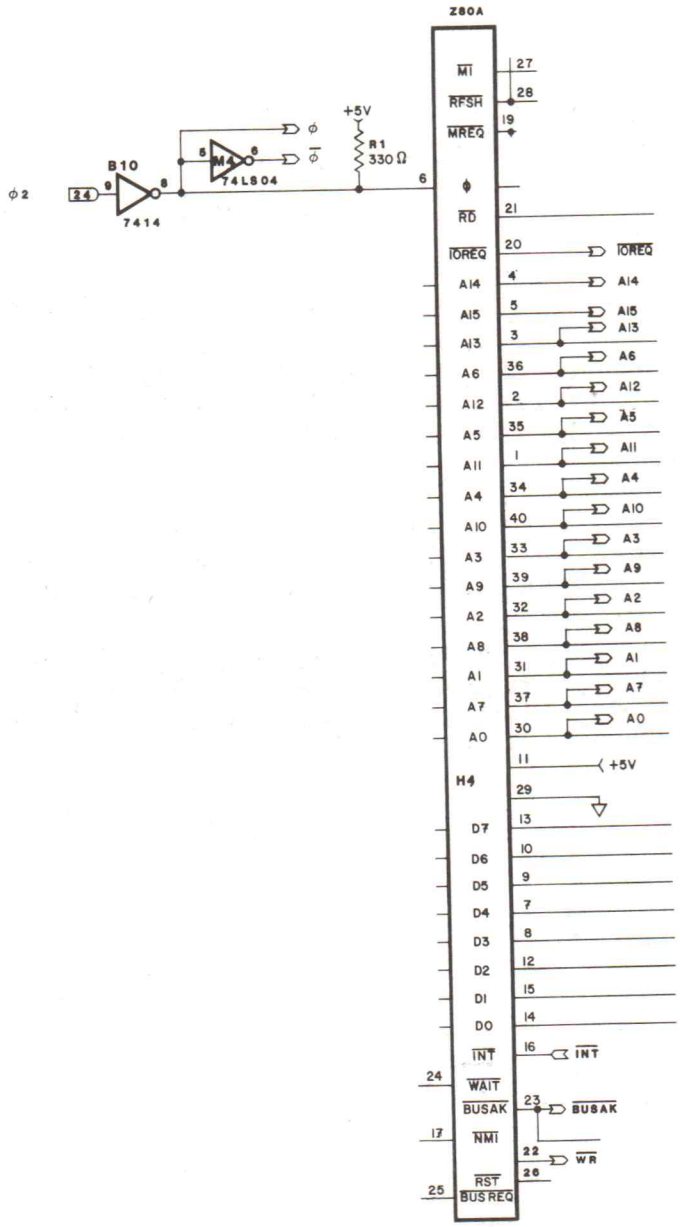


FIGURE II - 6

A Z80A microprocessor is connected to the internal MicroSat address, status/control and data busses. During non-DMA operations it uses these to run programs. During DMA operations, the MicroSat Z80A is put on hold. Bus control is performed by the master's Z80A. Refer to the Z-64 Manual for an extensive description of the Z80A.

The Z80A processor fetches program instructions (in what is referred to as a M1 cycle) from memory, interprets and executes them. The results of execution always include fetching the next instruction. In addition, changes may be made to memory, and information may be presented to the external environment. Those lines not part of the address or data busses form the control bus, which sends to and receives logical instructions from, the support hardware ICs.

The Z80A is only active during non-DMA data transfers with the master CPU. At these times, it operates in two modes: idle and user program execution. In both cases the Z80A has full control of the address and data busses and status/control output signals /M1, /RFSH (Refresh), /RD (Segment 2 discusses the relationship of these signals vis-a-vis RAM access) and /IOREQ (I/O Request).

During DMA transfers, the Z80A is disabled by an active /BUSREQ signal supplied by the system master processor board. At these times, the MicroSat's Z80A sets the /BUSAK signal. This, in turn, enables the various S-100 bus driver/buffers to allow the master Z80A to take control on-board address, data and status/-control busses (see Segment 4). In addition, the Z80A memory access control logic is put in the high impedance state. All signals for the transfer of data in and out of the satellite's memory block are provided by the master Z80A with one exception: the /RFSH signal which is necessary to refresh the dynamic RAM.

RESET from S-100 bus is used to reset satellite Z80A. NMI (non-maskable interrupt) is used to initialize the MicroSat after a system reset (see Segment 5 for a description of these functions).

The /INT signal is optionally connected to the USART TxRDY and/or RxRDY lines for interrupt driven I/O control.

The 4 MHz clock signal for Z80A operation is provided by the system master processor board (Z-64) on S-100 pin 24. Note that there is no resident crystal for the clock signal.

Clock buffer relates most of the function of this circuit. It takes the system's 4 MHz clock signal off of the S-100 pin 24, inverts it to provide both phase 1 and phase 2, and drives them onto the chips where they are required, including the Z80A.

SCHEMATIC SEGMENT 2: RAM ARRAY

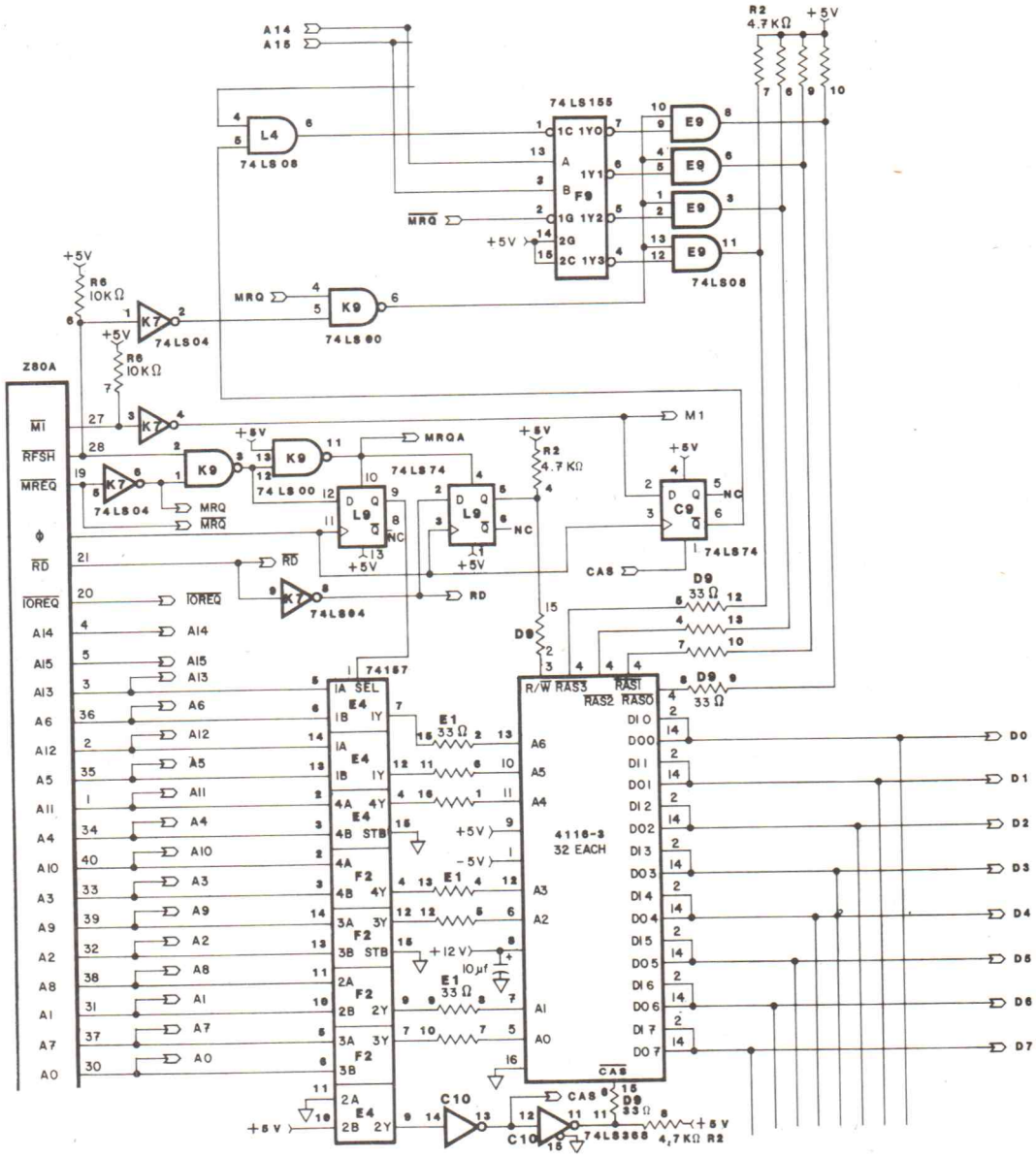


FIGURE II - 7

32 type 4116 Random Access Memory ICs are arranged in the 64K memory array shown as a single chip drawing above. Each is connected to the bi-directional internal data bus and the multiplexed address inputs. Read/Write signal, four Row Address Strobes and Column Address Strobe (CAS) are input.

This "64K" memory actually contains 65,536 eight bit bytes. Each of the 524,288 bits is stored in a memory cell, which is essentially a capacitor whose state of charge, or lack of it, determines the presence of a logical one or zero. These capacitors leak. To keep logical ones from being lost, cells must have their charge refreshed at least once every 2ms. The Z80A does this during the refresh cycles (at the end of each M1 cycle) in which it supplies:

- 1 of 128 refresh register addresses in seven bits
- activation of all four RAS lines
- no CAS activation (in contrast to a normal memory access)

Refresh, write, and read are the three possible activities of this RAM. Refresh requires only partial addressing, since 4096 bit cells are refreshed simultaneously. Writing or reading is accomplished on eight bits simultaneously, i.e., one byte is written or read per memory access.

The physical arrangement of the memory cells is such that each of the eight bits in a data byte is stored in a different chip. The Silkscreen Component Diagram at the beginning of Section II illustrates the 8 x 4 RAM matrix. Each IC holds 16,384 bits. A group of eight ICs form a "bank".

Within each memory IC are 128 "rows" and 128 "columns" of memory cells. It is these rows and columns to which the Row Address Strobe (RAS) and the Column Address Strobe (CAS) make reference. Thus RAS selects one 128 bit row and CAS selects the specific bit within the row yielding one data byte per IC. Headings 2 - 2 and 2 - 3 below describe RAS and CAS enable, respectively.

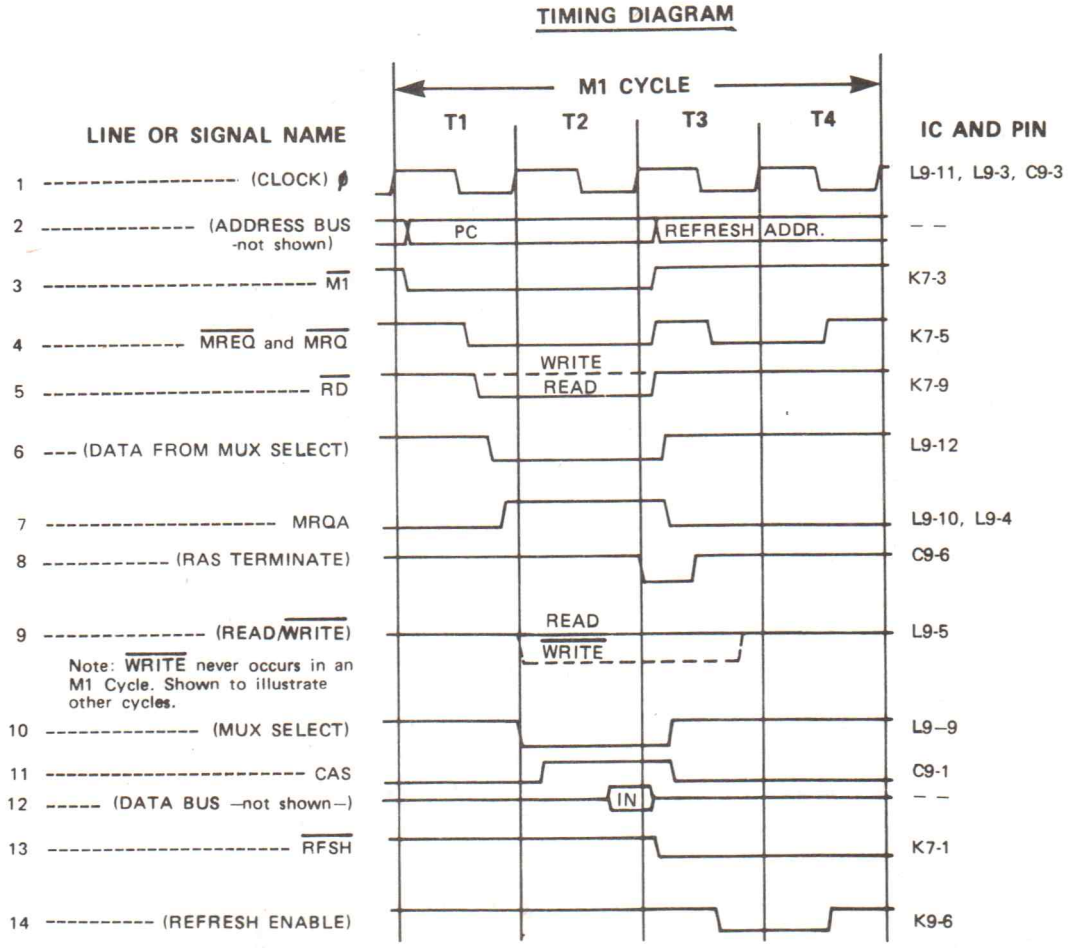
A memory access for write or read is accomplished through the output of a 16-bit address to the internal address bus, soon followed by an /MREQ (memory request) control from the Z80A CPU. The two most significant bits, A15 and A14 are examined by the logic described in heading 2 - 2 below, and are 1-of-4 decoded to enable the correct bank of ICs. The four banks are represented by inputs /RAS0-3.

The remaining 14 bits or the address bus are presented to the multiplexers at board locations E4 and F2. The MUX divides them into two sequential 7-bit groups input to the RAM A0 - A6 address inputs. The first bit group (A0-A6) are input during a RAS. Active-low RAS causes the group to be accepted by the memory. The multiplexer is switches to the second bit group (A7 - A13) by a MUX Select control from Seg. 8. The control signal is intentionally delay-gated by propagating itself through the E4-2Y output of the multiplexer and becomes (/CAS). After passing through two inverters, /CAS causes the second bit group to be accepted by the memory.

The final action in memory access is determined by the R/W (Read/Write) control signal to RAM chip pin 3 If accessed for write (a low voltage input), the memory accepts the eight bits presented on the data bus and stores them in the addressed byte. If accessed for read (high active), the memory copies the contents of the byte and presents them to the data bus.

The 33 ohm resistors provide damping for signal undershoot; they are recommended by most manufacturers of 4116 RAM ICs when driving from TTL logic.

The timing diagrams below illustrate the timing sequences during M1 cycles. Notice that the data bus is always sampled upon the rising edge between T2 and T3 cycles. During the refresh portion of an M1 cycle, the Z80A switches the contents of the address bus from the program counter to a 7-bit address from its refresh register. Thus, in combination with active RØ - R3, an entire RAM row is refreshed in each bank of chips. Data is not output to the bus because CAS is never activitated during the refresh.



M1 CYCLE RAM ACCESS TIMING DIAGRAMS

2 - 1: RAM ROW DECODE LOGIC

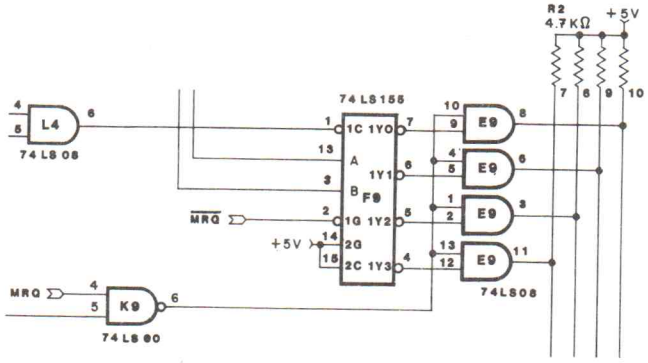


FIGURE II - 8

An AND gate (L4) output, with address and control bus bits, drive a 2-line to 4-line decoder (F9). The output of the decoder passes through four AND gates and onto the Row Address Strobe bus.

It is the function of this segment to distinguish a RAM refresh from a RAM access, inhibit RAM if the PROM is addressed, decode the two most significant bits of the address bus if a RAM access is requested, and terminate the access RAS period.

The four lines of the RAS bus correspond to the four rows of eight ICs . This pattern is visible on the Silkscreen Location diagram in Figure II - 2. The four lines, /RASØ-3 , are activated one at a time to access a specific memory location and four at a time to refresh memory.

Decoding is initiated by an /MRQ strobe-low at decoder F9-2, only if F9-1 sees a data-high. Address bits A14 and A15 are input to F9-13 and F9-2, respectively, decoded to select 1-of-4 possible RAS outputs and sent through gates E9 to the RAS bus. AND gates E9 are transparent to the RAS outputs.

AND gate L4-6 has two inputs. An active-low at L4-4 inhibits a RAS with priority in favor of a PROM access. An active-low L4-5 terminates a RAS decode.

The purpose of the 4 input AND gates E9 is for a simultaneous refresh enable of all four RAS lines. The Z8ØA contains a refresh address register which supplies (after multiplexing) 128 different 7-bit numbers to the RAM column address bus. Each number refreshs an entire row (4Ø96 bits) at the same time. RAS causes latching of the entire row in the chip. Without /CAS no bit is selected for subsequent output to data bus. The latching of the row from active /RASn is what causes the refresh. Every 128 M1 cycles, all of the RAM is refreshed.

2 - 2: RAM CONTROL LOGIC

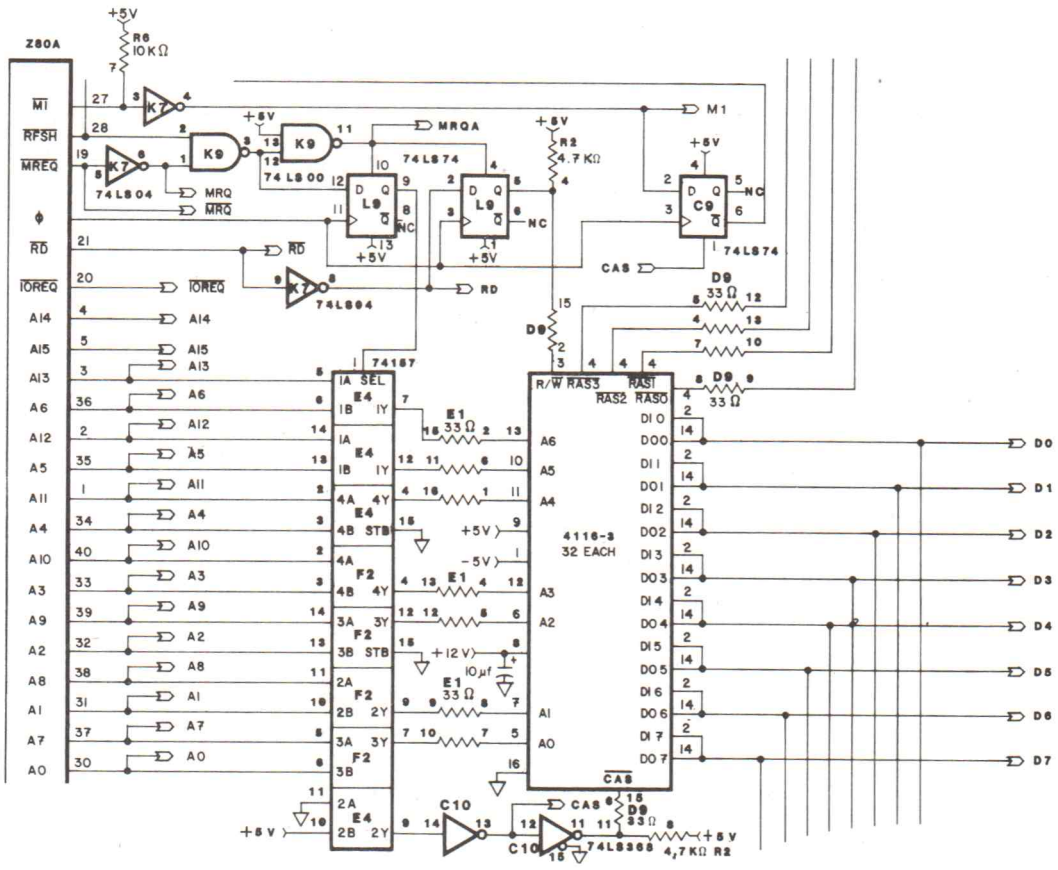


FIGURE II - 9

Three NANDs, three D-type flip-flops, and four inverters are configured in a logic and timing arrangement which accepts six inputs and generates four major and four supporting outputs for RAM access control.

This logic generates the correct timing from the Z80A /M1, /RFSH, /MREQ and /RD to allow it to access RAM. This segment can be divided into four circuits, each producing one of the four major outputs numbered (8), (9), (10) and (14) in the timing diagram above..

NOTE: During DMA transfers the satellite Z80A does not provide /RD or /MRQ. Consequently, to access a memory location, they must be provided elsewhere; i.e., the master Z80A. Refer to Seg. 4 for the origin of the surrogate /RD and /MRQ. /M1 & /RFSH are not used during DMA transfers.

8) RAS Terminate: The falling edge of this output terminates the Row Address Strobe except those generated by a (Refresh Enable). The source of RAS Terminate is /M1 input to inverter K7-3. M1 is presented to the D-type flip-flop C9-2 as an active-high, which is not clocked through because of an inhibiting inactive-low on the CAS line. When CAS goes high, a RAS Terminate is generated by the rising edge of the next clock cycle.

9) Read/Write: The status of this signal determines whether a read from (high) or write to (low) the data bit cell is to be performed. The source of Read/Write is the Z80A /RD signal input to inverter K7-9. The resulting RD is presented to the input of D-type flip-flop L9-2 as data which is not clocked through because of an inhibiting inactive-low on the MRQA line. When MRQA goes high, Read/Write is rendered on the rising edge of the next clock cycle.

10) MUX Select: The low state of this output will cause the address multiplexers to switch from the B (address lines A0 - A6) to the A inputs (address lines A7 - A13) and, additionally, output a /CAS to the RAM. The source of MUX Select is /MREQ or /MRQ (only active during DMA transfers) input to inverter K7-5. The resulting MRQ is NANDed with /RFSH and presented to the input of D-type flip-flop L9-12 as data which is not clocked through because of an inhibiting inactive-low on the MRQA line. When MRQA goes high, MUX Select is generated on the rising edge of the next clock cycle. Note that A0-A6 select the RAM row and A7-A13 select the RAM column; A14 and A15 select the RAM bank.

14) Refresh Enable: The low state of this output causes all of the RAS bus lines to be simultaneously active-low which refreshes 4096 memory cell bits in the RAM. The source of Refresh Enable is /RFSH input to inverter K7-1. The resulting RFSH is NANDed with MRQ to output Refresh Enable. This is inactive during DMA.

The 4116 RAM ICs require sequentially multiplexed addressing in order to keep the pin count low. To address the entire space of those chips takes 14 bits, and that is done on seven pins. The multiplexers shown present seven of the address bits (A0-A6) during the Row Address Strobe, followed by the other seven during Column Address Strobe (A7-A13).

When active-low, the (MUX Select) line selects the A inputs attached to address bits A7-A13. The B inputs are presented to RAM during RAS time, and the A inputs are presented during CAS time. Note that the switch of address lines is made at the beginning of the T2 clock cycle as illustrated in the timing diagram above.

Note that (/CAS) is actually the output of a ground tied to an A input. This is done to insure that /CAS occurs only after the multiplexers have switched from B to A inputs; and to assure, through MRQA reset of the L9 flip-flops, that /CAS goes away when R/W switches at the end of a RAM cycle.

SCHEMATIC SEGMENT 3: PROM

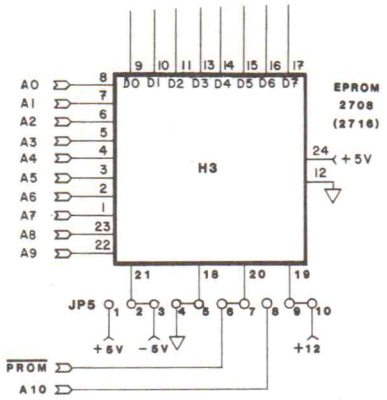


FIGURE II - 10

An Erasable Programmable Read Only Memory (abbreviated as PROM) is connected to the internal address and data busses. It is selected by logic that analyzes address bus signals A10 - A15, active MRQA and, after a reset, /POJ (power on jump). A wait state is inserted in the access cycle to allow for the device to render its bits to the data bus.

NOTE: This PROM is not used in M/NET systems. The socket and select logic described below in heading 3 - 1 are installed for a PROM used during MicroSat factory testing.

The PROM occupies the address space from FC00 through FFFF hex. This space can contain a program to be executed by the resident Z80A. The device is enabled by the /PROM signal described below.

The jumper pads JP5 and JP7 allow installation of a 2K custom PROM. (The standard configuration has these jumpers set for a 2708 1K PROM.) JP5 is provided to accommodate a 2716 2K PROM in either of two power configurations: the INTEL type that uses a single supply voltage, or the TI type which uses three supply voltages. Jumper JP6 must be cut when a 2716 type PROM is installed to enlarge the PROM select range to F000 - FFFF.

3 - 1 PROM SELECT LOGIC

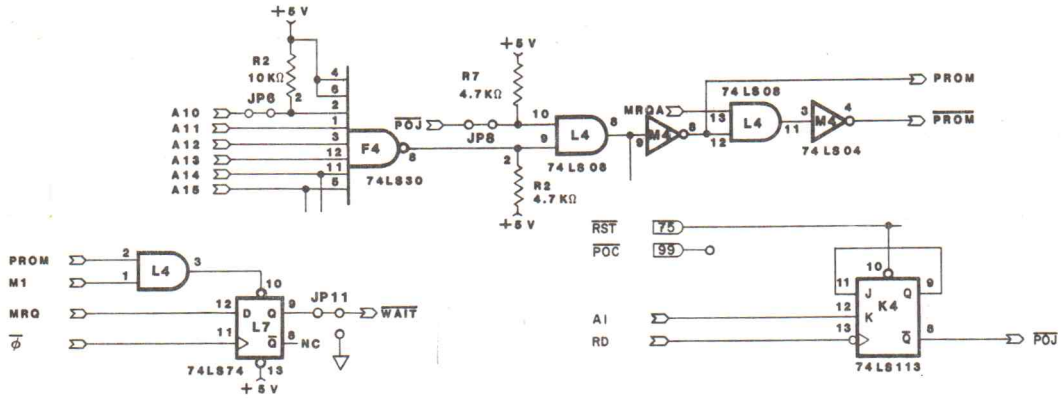


FIGURE II - 11

A 8-input NAND connected to address signals A1Ø - A15 outputs to two AND-inverter combinations outputing the non-complementary pair PROM and /PROM. A third control, to disable RAM, is output from gate L4.

It is the function of this segment to detect PROM addresses FCØØ - FFFF for a 1K PROM or FØØØ - FFFF for a 2K PROM on the six most significant bits of the internal address bus. In a secondary function, RAM is disabled and the PROM is enabled by an active-low /POJ. This is the other half of the Reset Jump Hardware described in Segment 5.

During the principal function, the address matcher at gate F4 watches for memory addresses from FCØØ (FØØØ) to FFFF. It acknowledges them with a low output to gate L4. With /POJ in a normal high state, an active-low out from L4-8 commands a Row Address Strobe inhibit in Segment 2. This is effectively a RAM disable, since the action of a memory Column Address Strobe (/CAS) will be ignored if a valid /RAS didn't precede it.

Note that the inverter M4 makes the PROM control line the TRUE complement of the RAM disable just discussed. This control helps time wait states for the PROM memory. The M4-8 output, labelled PROM, is NANDed with MRQA to output /PROM -- the actual PROM memory select line.

This logic is used to insert a Z8ØA wait state during a PROM access because PROMs typically respond more slowly than the 2ØØns RAM ICs supplied with the MicroSat board (which require no waits). A wait state halts the Z8ØA for one clock cycle allowing the PROM to provide the data byte.

D-type flip-flop L7-9 is normally held in the Preset state rendering an inactive-high on the /WAIT line. When the PROM and M1 (from the Z8ØA) go active-high, the inhibiting low at L7-1Ø is released. Now, an active-high MRQ (memory request) will be clocked through as a /WAIT (wait request) by the next rising edge of the system clock.

SCHEMATIC SEGMENT 4: DMA TRANSFERS

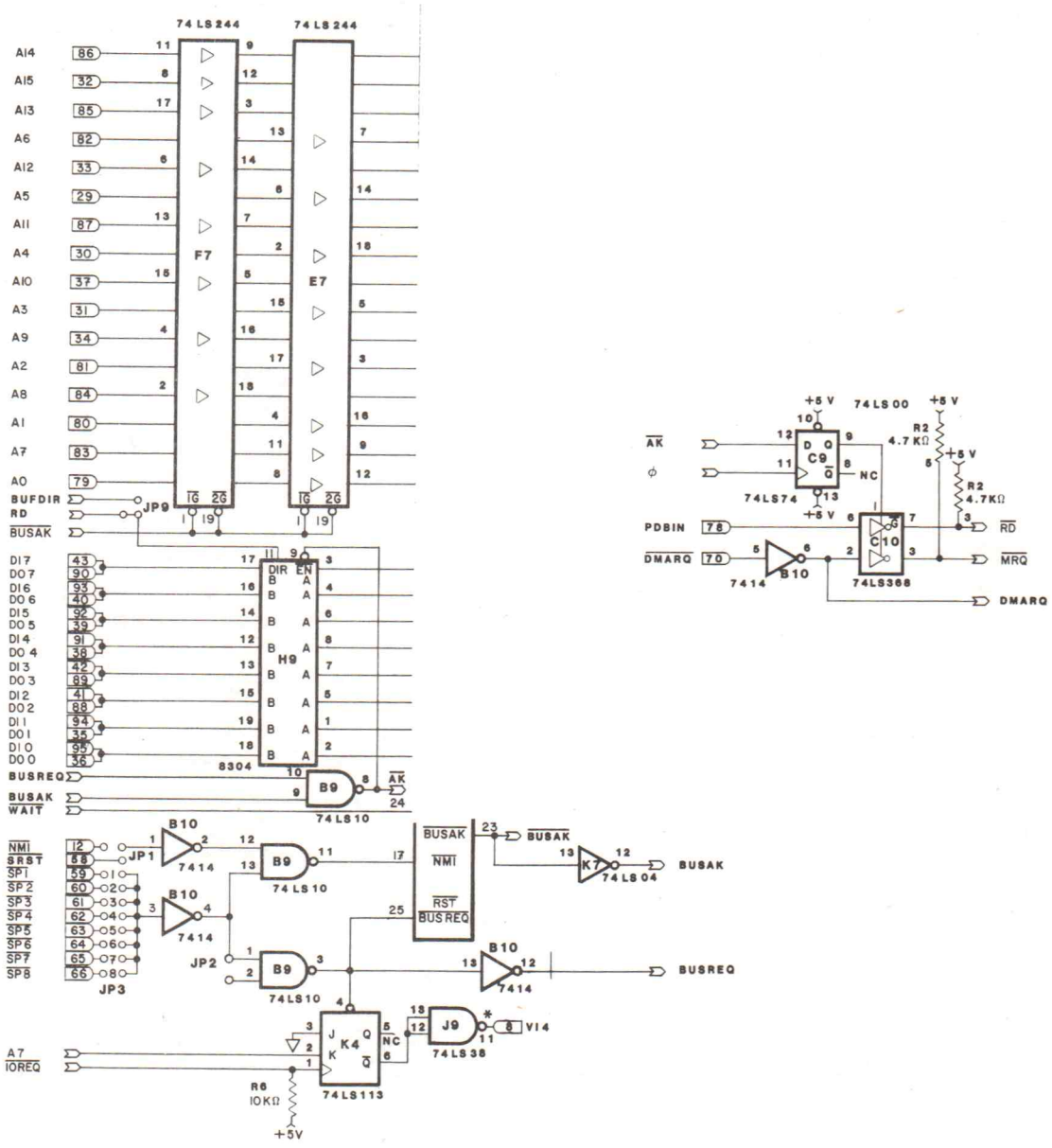


FIGURE II - 12

Two 74LS244 octal buffers/line receivers separate the MicroSat from the inter-board address bus. An 8304 bi-directional buffer/driver does the same for the data bus. Control signals disable the on-board Z80A and enable the buffers during master/satellite DMA data transfers to allow master access to the MicroSat's RAM.

All data transfers between master processor card and MicroSat satellite processor are controlled by the master; the satellite sits passively on the bus during this time. To perform this task, the satellite's Z80A is put in a hold state via the BUSREQ input. The Z80A's acknowledge signal, /BUSAK, is then used to enable the bus buffer/drivers. These open the otherwise shut-out RAM to control by the master.

Because of the complexity of this schematic segment, it has been broken down into several sub-segments.

- The first describes the interrupt generated by the MicroSat that indicates to the master that the former is requesting an action.

- The second describes how a single MicroSat in an M/NET system is selected from the many that may be installed. The satellite's response to selection is also described.

- In the third sub-segment, the circuit and signals active during the data transfer are described.

- Finally, the optional timing for data transfers is described.

Recall that data transfers between master and satellite processor cards is done with the Z80A block move command. This command was originally designed by Zilog to allow a block of data to be moved from one area in memory to another. Judicious and timely use of the /PHANTOM line, however, expands the utility of this facility to include interboard data transfers.

To summarize how it's done, let us assume that the data flow is from master to satellite and the selected MicroSat board's Z80A has already been put in the suspended operation state. During the fetch from memory and write block move cycle, the address bus is used to first grab a byte of data from the master's memory. /PHANTOM is not active during this part of the cycle. Soon thereafter, the master's CPU address the destination location. Address matching logic on the M/NET I/O board detects that the destination address is on the satellite and enables the /PHANTOM line. Thus the master's memory is "disappeared" and the satellite's memory is the only block accessible. Hence, the byte is deposited in the latter.

For satellite to master data transfers (which are also controlled by the master) /PHANTOM is active during the first part of the cycle only, when the data byte is being fetched from the satellite's memory block. During the second part, /PHANTOM goes inactive when the byte is written to the master's RAM.

4 - 1 DMA INTERRUPT REQUEST GENERATOR

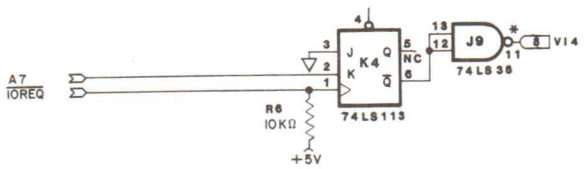


FIGURE II - 13

An address line and two controls are connected to a JK flip-flop whose /Q output is driven onto S-100 bus line VI4 through an inverter-wired gate J9.

The satellite board requests a DMA by the master CPU board when a program executing in the satellite memory requires data I/O (i.e., data to or from terminal, printer or mass storage device.). The call is trapped by the satellite operating program which writes a message in the scratch RAM window and causes a Z80A OUT instruction to be executed enabling the /IOREQ signal.

A Vectored interrupt to the master is generated on S-100 bus pin 8, line VI4. VI4 is used by all the satellite boards for this purpose. If /IOREQ is strobed while address bit A7 is high, the JK flip-flop toggles and sets VI4 low.

The master subsequently does a DMA of each board and examines each board's mailbox to discover the source of the interrupt. Since all the satellites interrupt on VI4, each window or mailbox is examined.

Each time the master does a satellite DMA, /BUSREQ becomes active. In the process, the JK flip-flop at K4-4 is presented with a Preset pulse. If the JK flip-flop was already present, nothing changes; the master examines the mailbox, finds that this is not the interrupting board, and goes to the next board. If, however, the JK flip-flop had been toggled by an interrupt request, the Preset pulse would toggle it back again and the interrupt on VI4 would be cleared and the mailbox examined.

4 - 2 DMA SATELLITE SELECT AND CONTROL LOGIC

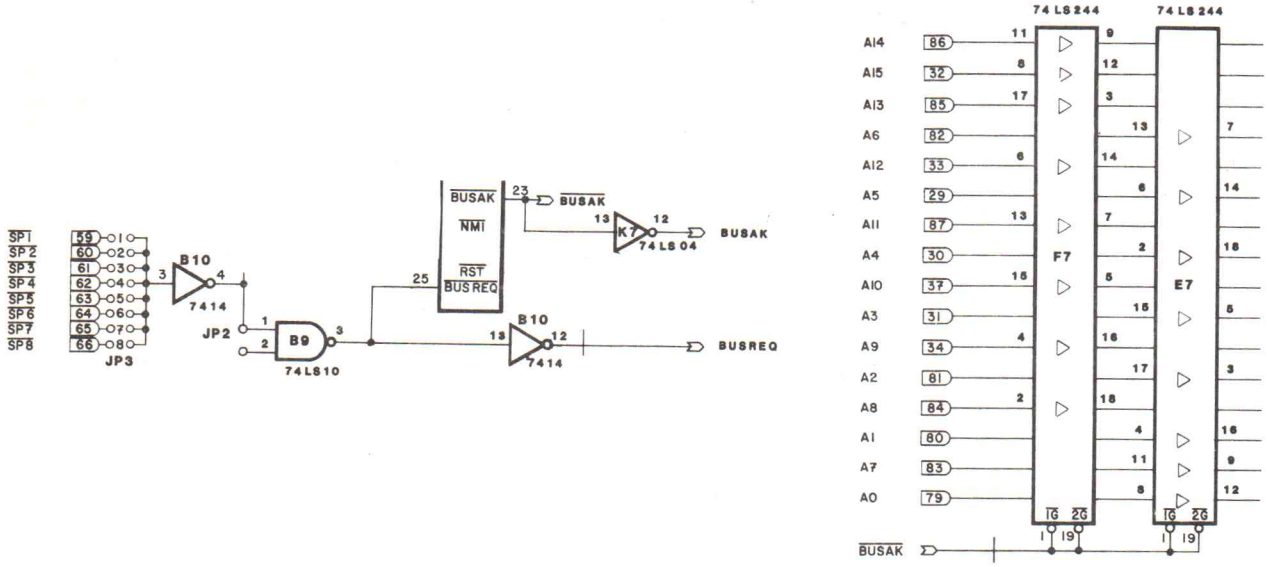


FIGURE II - 14

Jumper pad JP3 determines the MicroSat physical number for DMA transfers. This signal renders /BUSREQ (bus request) to the MicroSat Z80A. The Z80A is then put in the hold state and generates the /BUSAK (bus acknowledge) to enable the S-100 address, data and control bus buffers/drivers.

These circuits allow the master CPU board, to gain control of the satellite through the S-100 bus. The two satellite control operations provided by this hardware are Direct Memory Access of the satellite, and Initialization of the satellite software. The master initiates DMA by presenting a low on S-100 bus lines SP1-SP8 via the M/NET I/O board. Only the satellite wired to a particular SPn line, through jumper JP3, responds to the low.

The signal from JP3 is inverted at B10 and inverted again at B9 (see heading 4 - 4 below for a description of JP2; it is not relevant to the current discussion) to render /BUSREQ to the resident Z80A. This signal interrupts the current operation, letting it finish first, and pushes the program counter value onto the stack. The Z80A subsequently goes on hold. This renders the its address, data and control signal pins in the high impedance state with the exception of one signal -- /BUSAK. This signal is maintained in its active state (low) until /BUSREQ goes inactive.

/BUSAK is input to the S-100 address, data and control bus driver/buffers and enables them. Thus the master Z80A gains access to the MicroSat's RAM.

4 - 3 DMA RAM ACCESS CONTROL

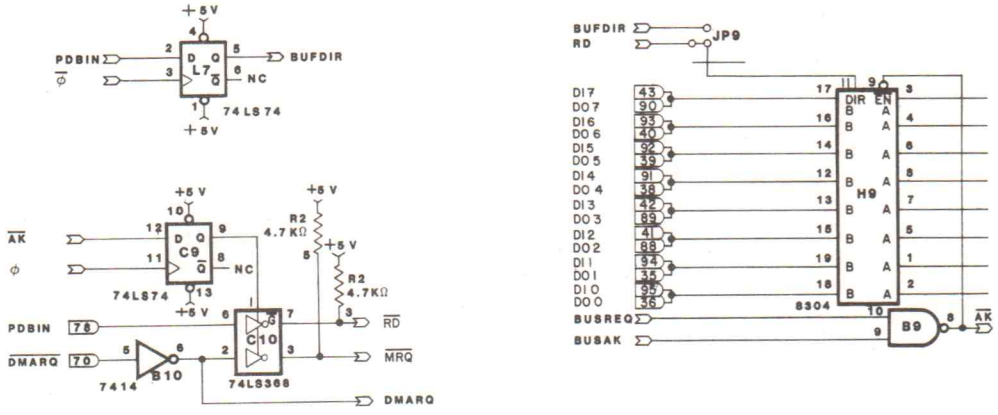


FIGURE II - 15

An 8-bit bidirectional transceiver buffers the internal data bus from the S-100 data bus. A NAND gate enables the device during DMA transfers.

This transceiver is in operation only during satellite DMA, when it can drive data in either direction. The enable for the buffer during DMA is generated by AND gate B9-8. The enable, /AK, is active-low during the period when the master has requested bus control (active /BUSREQ) and after the Z80A has responded (active /BUSAK). Note that /BUSREQ is a high priority interrupt. It causes the Z80A to terminate a program after the current instruction has completed and to push the value of the program counter onto the stack. Inactive /BUSREQ re-enables the Z80A.

Data direction is determined by the state of RD, which provides buffer turnaround timing during normal DMA. When RD is high at H9-11, the buffer transmits the data byte from the MicroSat RAM to the S-100 bus and onto the master. When RD is low, the buffer transmits the data byte from the bus (as generated by the master) to the MicroSat for a memory write.

The system clock signal, along with /AK, enables hex driver C10. This allows the control signals that enable the RAM chips -- PDBIN and /DMARQ -- through from the S-100 bus. Notice that PDBIN replaces the Z80A's /RD signal and DMARQ becomes the Z80A /MREQ signal. Notice that this signal is held active (low) during the entire DMA transfer. The C10 enable signal goes inactive (high) upon the rising edge of clock after /AK goes inactive (high). Refer to Segment 2 for a description of how these are used to access specific memory locations.

Jumper JP9 allows use of the BUFDIR signal to set the data direction. Similar to RD, BUFDIR is set by PDBIN. However, unlike RD, it is enabled by the inverse of the system clock signal. This causes a half clock cycle delay. Note that this option was installed for test purposes only and should not be selected.

4 - 4 OPTIONAL TIMING GENERATOR

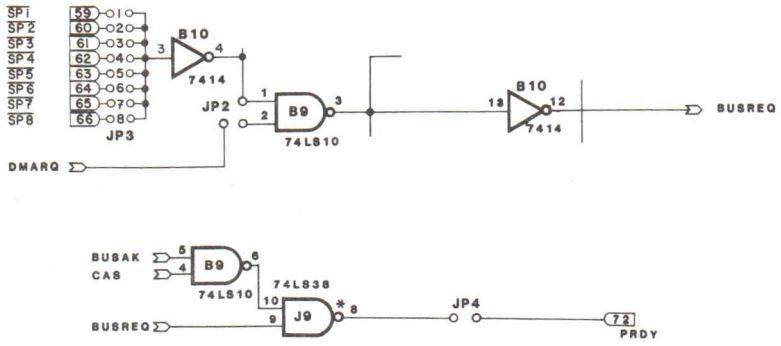


FIGURE II - 16

Two jumpers interrupt the DMA transfer process to allow dynamic RAM refresh and insert a wait state in the master's MicroSat RAM access cycle.

Blocks of data up to 256 bytes can be transferred between master and satellite while the satellite's Z80A is on hold. However, if the Z80A is in limbo much longer than the time required to transfer this much data, damage to the unrefreshed RAM will result. (Note that while BUSREQ is active, the Z80A's refresh control logic is shut off.)

The circuitry illustrated in this segment prevents this mishap when longer data blocks are moved. This logic is enabled by jumpers JP2 and JP4.

The JP2 jumper allows the DMARQ signal to reset the Z80A /BUSREQ input. By connecting B9 to the DMARQ (which is active once per fetch/write cycle) signal from the M/NET I/O board, the MicroSat's Z80A is taken off hold when its memory is not being accessed and allowed to continue with RAM refresh. Otherwise, when JP2 is connected to the SPn line only, the Z80A is kept on hold during the entire transfer.

A wait state is also inserted during this cycle. BUSAK, CAS and BUSREQ condition the PRDY output on the S-100 bus pin 72 via jumper JP4. Note that this signal state is high active indicating that the processor can proceed. It is connected to the master's Z80A /WAIT input. Thus a low, causing a wait state in the master Z80A memory request cycle, is generated when CAS goes is active (low). This allows the master CPU to wait for the longer RAM access time required in the DBOS timing DMA transfer.

NOTE: The JP2 and JP4 options are only set to the configuration described in this heading for DBOS systems.

SCHEMATIC SEGMENT 5: MICROSAT RESET AND INITIALIZATION

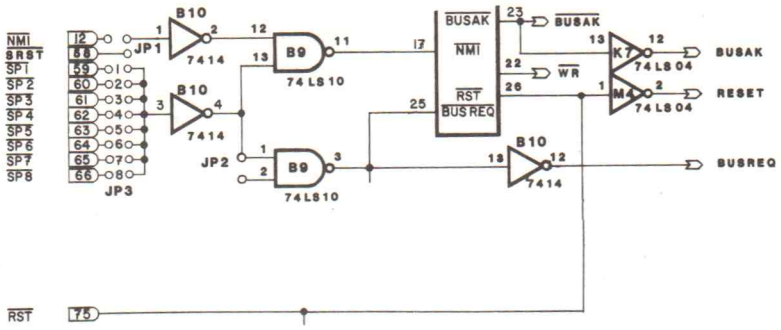


FIGURE II - 17

S-100 signal /NMI and /SPn are used for MicroSat cold boot initialization and warm restart. Alternately, a JK flip-flop takes inputs from on-board signals A1 and RD and S-100 signal /RST causing a board reset and power on jump into the satellite PROM firmware for three machine cycles.

Each MicroSat board in an M/NET system is reset via the S-100 /RST signal from pin 75. Not until the master processor card has loaded the CP/M emulator program into the satellite is it operational. Loading this program is performed after the master operating system has been loaded from disk and other system components (USARTS, hard disk, etc.) have been initialized. Until this has been done, the MicroSat is in limbo. That is, the Z80A begins executing from location 0000 but there are no instructions there (just NOPs).

Cold boot initialization involves then two steps: 1) loading the CP/M emulator program and 2) performing a warm restart to have the Z80A jump to the idle routine. (Recall that when not running a program, the MicroSat is executing an idle routine.) The first step is fairly straightforward. The emulator program is loaded into the top of each satellite's memory via DMA by the master. This process does not indicate to the Z80A CPU where it should begin execution, however.

Initialization of the satellite software is a houskeeping operation initiated by the master to terminate a process and prepare a satellite for execution of a user's program (a warm restart). It is also the final step in the cold boot initialization process. The master begins initialization with a normal DMA during which /BUSREQ is held active-low. The master then strobes the /NMI S-100 bus pin 12. Since /BUSREQ has priority over /NMI, nothing happens immediately except that the transient Non-Maskable Interrupt is latched in for use at the end of the /BUSREQ .

During the /BUSREQ , the master uses DMA to insert a three byte Z80A jump instruction starting at location 0066 hex. This jump is vectored into the CP/M BIOS emulator routine.

The latched /NMI takes control of the satellite when the master releases /BUSREQ . /NMI always forces a jump to 0066 hex. From there it jumps to COLDBOOT . The master will wait about 20ms for COLDBOOT to complete initialization, although only about 1ms is actually required.

Jumper JP1 allows Micromation the future option of supplying the /NMI strobe on S-100 bus pin 58, instead of the current pin 12.

The circuitry shown above also supports a power-on-jump into the PROM option. This facility is not utilized on the MicroSat board since the resident program is not stored in an EPROM.

When the system reset button is pressed, /RST becomes active (low). This signal lasts three clock cycles (as set by the Z-64) on the Preset pin of the JK flip-flop. The JK Preset state forces Q high and /Q low. /Q becomes /POJ which is used to select the PROM address space instead of location 0000 (the location addressed by the Z80A upon reset).

When /RST goes inactive, the /Q output remains active until the K input (K4-12) goes high. Since this input is tied to address line A1, this does not occur until the third memory fetch after a reset when location 02 (0000 0010 hex) is requested by the Z80A. /Q remains low through the machine cycle until the falling edge of RD is encountered. At this point, /Q goes high.

With /POJ enabled the first three locations of the PROM are fetched. A JUMP instruction and 2 byte location can be stored pointing to a routine that begins anyplace in the PROM memory space (base FC00 or F800). When /POJ goes inactive, the Z80A's program counter value is replaced by the new address as per the JUMP protocol.

SCHEMATIC SEGMENT 6: USART COMMUNICATIONS LOGIC

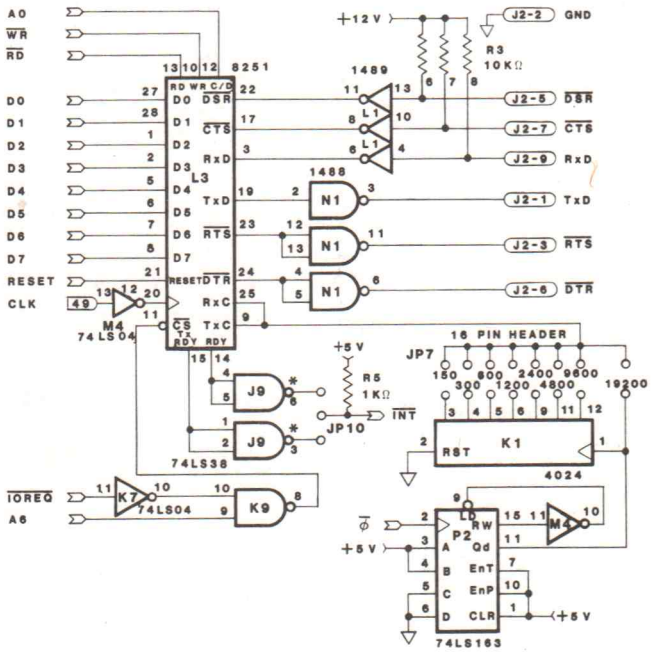


FIGURE II - 18

An 8251 USART is connected to the on-board data bus and to address bus bits A0 and A6 via logic at K9-8. RS-232 line drivers and receivers connect the UART to a data jack.

NOTE: As of this edition of the MicroSat manual, this device has not been implemented. The following description is provided in anticipation of its use.

The on-board 8251 can be used for two purposes. At the time of purchase the system integrator can specify an M/NET system wherein the satellites communicate directly with the user console via this device. Alternately, in systems where console I/O is handled by the master (via the M/NET and Multi I/O boards), this device can be used for any serial peripheral. Note that this latter option may require substantial alteration to the MicroSat firmware.

The USART is enabled by active /IOREQ and A6 signals input to the K7 and K9 NANDs. The result is input to the 8251's /CS pin. The I/O addressing on the satellite board is not fully decoded, so the UART will be selected anytime a Z80A IN or OUT instruction is executed to an I/O address with bit 6 high (40 hex to 7F

hex). Note that either an IN or an OUT instruction renders /IOREQ active (low).

Two types of data are transferred between CPU and USART on the MicroSat data bus: control/status and data. Both types can be read or written. The 8251's control registers are accessed by setting the C/D input, L3-12, (from address line A0) low and enabling either the /RD (processor read), L3-13, or /WR (processor write), L3-10. inputs. Data is transferred to/from the device when the A0 input is high.

Regarding other inputs to the 8251:

CLK: The operating speed of the 8251 is set by the 2 MHz clock signal from pin 49 on the S-100 bus.

RESET: The 8251 is reset whenever the MicroSat card is by active RST from S-100 pin 75.

TxC and RxC: These inputs set the transmitted and received data baud rates in the 8251. Notice that they are tied together. The input signal is furnished by the baud rate generator described in Segment 4 below.

Note that the I/O signals between the 8251 and the peripheral are buffered by a 1489 (L1 for input signals) and 1488 (N1 for output). These devices translate the TTL level signals to the levels compatible with the 8251 (+/- 12v to 5/0v levels indicating high and low respectively).

Jumper JP10 allows the additional option of implementing the USART in an interrupt environment. By connecting the pins of JP10, the TxRDY (transmitter ready), the RxRDY (receiver ready), or both may be connected to generate an interrupt on the /INT line to the Z80A. The alternative is the use of a polled environment. Either environment would required supporting software not currently extant.

6 - 1 BAUD RATE GENERATOR

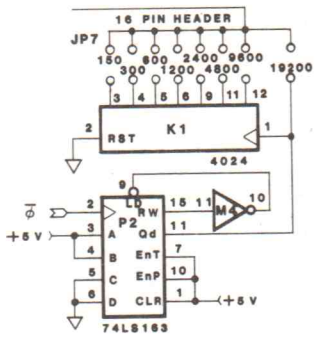


FIGURE II - 19

A synchronous 4-bit counter outputs to an 8-bit binary counter for baud rate generation. Rates from 19.2K to 150 bps are available (jumper selectable via the 16-pin header next to the 10-pin I/O header) for the 8252 USART.

This segment of the MicroSat generates a 19.2K baud signal from P2-11. This signal is input to a 8-bit binary, divide-by-two counter (K1) to render lower frequency signals. The value selected by a header jumper determines the baud rate for communication between the USART (described in Segment 1) and a serial peripheral.

The phase 2 clock signal (4 MHz from Segment 10) is used to clock a 4-bit counter. At the 0th clock pulse (of 0 thru 12) an output from P2-11 is sent to chip K1-1. Also at the 0th, P2-15 is output, inverted, and used in a feedback loop to self-strobe its own Load pin. This action loads a binary 3 into the counter. The 3 comes from bits hardwired to the counter's data inputs: P2-3 to P2-6.

Clock pulses 0 through 12 advance the counter from 3 through 15. At the end of count-value 15, a carry is output which loads a "3" to repeat the cycle. The output at P2-11 is the system clock frequency divided by 13 or 19.2K bps. This is sent to the next divider at K1.

The K1 divider outputs a choice of seven different frequency-divided outputs. These are shown in the schematic as a jumper JP7 choice of eight different baud rates (including the input to K1). The jumper selectable baud rates are

19,200	2,400	300
9,600	1,200	150
4,800	600	

bits per second.

SCHEMATIC SEGMENT 7: DC POWER REGULATORS

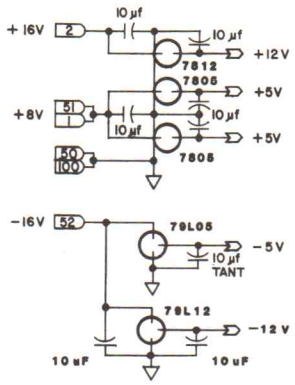


FIGURE II - 20

Three unregulated DC voltages are input from the S-100 bus and connected to five voltage regulators, which output five regulated voltages for use by the components on the internal power bus. Eight capacitors reduce AC ripple on the bus.

A requirement of the S-100 system design is that all boards must regulate their own voltages for their own components. In the case of the satellite boards, five voltages are required which are different from those supplied on the S-100 bus.

Supply voltages must also meet tolerance specifications in order to maintain signal isolation between unrelated circuits on the board. If tolerances were not maintained, for example, a high current square wave driving the 32 chip memory assembly might pull a supply voltage in and out of regulation, with every cycle. In effect, the memory square wave might potentially modulate the output of every other IC which shared that same regulator.

The following table lists the eight supply voltages found on the MicroSat board with the peak-to-peak AC ripple tolerance for each:

	VOLTAGE	RIPPLE,MAX	TEST POINT
UNREGULATED	+16/18.5	1.5VPP	S-100 PIN 2
	+8/10	1VPP	" " 1 & 51
	-16/18.5	1.5VPP	" " 52
			GRND " 50 & 100
REGULATED	+12	+/-10%	(Check the voltage
	+5 #1	+/-10%	level at any device
	+5 #2	+/-10%	to determine if it
	-5	+/-10%	is accurate.)
	-12	+/-10%	

APPENDIX A:

S-100 BUS COMPATIBILITY

Nine S-100 bus signals are used in a non-standard fashion to support Micromation's unique multi-processor architecture. The table below indicates the S-100 pin numbers of these signals and their application.

S-100 Pin #	Signal Name	M/NET Application	IEEE Definition
59	SP1		A19
60	SP2	The 8 SPn signals are used	SIXTN*
61	SP3	to select one of the poten-	A20
62	SP4	tial eight MicroSat cards	A21
63	SP5	for DMA transfers from	A22
64	SP6	Master to satellite data	A23
65	SP7	transfers.	NDEF
66	SP8		NDEF
70	DMARQ	Supplies MREQ on MicroSat during DMA transfers	GND
98		Power from batteries to M/NET I/O Calendar/Clock chip.	ERROR

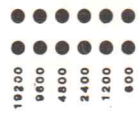
These signals, except for the last, are generated by the M/NET I/O card and are used by the MicroSat cards only. The remainder of the board level components in the M/NET system to not use these signals.

APPENDIX B

REFERENCES FOR THE MP/M BASED M/NET SYSTEMS

B - 1 MicroSat Jumper Settings for MP/M

The following table lists the proper MicroSat jumper settings for MP/M based M/NET systems. The MicroSat will not work unless these jumpers are set as described below. Place the MicroSat before you with the component side up and the S-100 fingers toward you when comparing the illustrations to the board.

Jumper	Setting	Description
JP1		Connects S-100 /NMI to Z80A /NMI
JP2		Second inversion of SPn to render /BUSREQ
JP3		Selects MicroSat number (different for each MicroSat card in a system)
JP4		PRDY to Master (not used in MP/M based systems)
JP5		Set for Intel 2708 PROM (unnecessary in M/NET systems)
JP6		Cut for a 2K PROM (unnecessary in M/NET systems)
JP7		Sets baud rate for USART (not implemented as of this edition of this manual)
JP8		Enables cold-start /POJ for PROM (unnecessary in M/NET systems)
JP9		Always set to select RD
JP10		Enables TxRDY and/or RxRDY for Z80A /INT
JP11		WAIT state for PROM reads (unnecessary in M/NET systems)

B - 2 Direct Memory Access and Mailbox Communication

The M/NET system is designed so that the master processor can directly read and write the memory of the satellite processors. Directly, means that the satellite processors suspend processing (the resident Z80A is put on hold with active /BUSREQ) for an instant while the system master processor board's Z80A accesses the former's memory space for a DMA.

The master processor communicates with the satellite's memory via the S-100 bus. During DMA, the satellite address, data, and control busses are controlled by the master. Only one satellite memory may be accessed at one time.

The satellite DMA enables are initiated with the master sending a bus request (/BUSREQ) to the distant processor by enabling one of lines SP1 - SP8. This sets the satellite Z80A's bus connections to a floating, high-impedance state, and sets the bus acknowledge (/BUSAK) signal active opening the data, address and control bus buffers. Thus the master gains access to the satellite's memory block.

Each satellite has an area of memory reserved for a mailbox buffer: FB00 through FBFF hex. This area contains several subareas, the most important of which are the incoming mailbox and the outgoing mailbox. Outgoing contains messages to the master; incoming holds messages to the satellite.

Messages take the form of five operation codes, which have different meanings depending on which direction the message is being sent. The table on the following page lists the Op Codes and the register detail utilized. These codes allow the master to move data around in the satellite memory, and begin satellite's execution; they allow the satellite to get master execution of less common CP/M subroutine calls and to request termination of the program in satellite memory.

The on-board CP/M emulator control program contains two main routines: the BIOS emulator and the BDOS emulator (the acronyms are those of the CP/M vocabulary). Both routines accept user program subroutine calls to CP/M through the 16-position BIOS jump vector, or the 41 BDOS function codes. (The BDOS emulator also accepts XDOS codes 145, 152 and 155.)

The emulator communicates information to the master board by placing data in the Out Mailbox (6 bytes at FB0A) and executing a software OUT instruction. This generates a Z80A /IOREQ control, which together with a high A7 address bit, drives an active-low vectored interrupt VI4 onto the S-100 bus. In response to the interrupt, the master first attempts to reset each MicroSat board to find out which one generated the interrupt. The Master then reads the satellite mailboxes; the interrupting board will have a non-zero operation code in the mailbox.

The master acts according to the operation code. If the request is for execution of one of the BDOS functions, the appropriate BDOS code is passed to the master.

Since the master reads and writes mailboxes with DMA while the satellite processor is stopped, the satellite can't directly see the actions of the master CPU. The satellite goes into a wait loop which continuously samples the In mailbox for a non-zero operation coes. This will occur when the master has finished the requested operation.

A satellite process (program) termination is accomplished when the master initializes the satellite memory via the Non-Maskable Interrupt (/NMI line).

Outgoing Mailbox Register Detail and Codes:

Op Code	C Reg	E Reg	D Reg	Disk DMA Address
FB0A	FB0B	FB0C	FB0D	FB0E - FBOF

NOTE: Interrupt flag is the MSB of the Op Code register.

Outgoing Op Codes:

0 - Inactive mailbox
1 - CP/M BDOS call
2 - Data sent from buffer
3 - Data is in buffer
4 - Terminate this program

Incoming Mailbox Register Detail and Codes:

Op Code	A Reg	L Reg	H Reg
FB10	FB11	FB12	FB13

Incoming Op Codes:

0 - Inactive Mailbox
1 - CP/M BDOS call
2 - Send (A) bytes of data to address (H,L)
3 - Get (A) bytes of data from address (H,L)
4 - Start program execution at address 0100

B - 3 MicroSat Bootstrapping

A single event, power-on or pushbutton Reset, must trigger a chain of events which ultimately allows execution of the user's program. This chain of events, from Reset to the appearance of the operator's prompt, is bootstrapping. Another way of looking at this procedure is that it synchronizes all the cyclical events and logical paths throughout the computer. Without a bootstrap, each processor in the system (including those of the satellite and disk controller boards) begins to execute from an undefined memory location . At power-on this memory area contains undefined states and the processor would execute randomly.

When the system is reset, the MicroSat is a tabula rasa -- there are no programs there to run. As part of the system cold boot process, the master attempts to initialize each satellite. If it is successful, the CP/M emulator program is loaded into the top of the satellite's memory space via DMA. Subsequently a warm restart (active /NMI to the MicroSats) is executed to complete the initialization process and set each board in the idle mode.

A map is maintained in the master indicating the number of satellites in the system. The master assumes 8 satellites when the system is reset. Satellites are removed when they fail the initialization test (i.e., there is either no satellite there or the satellite installed is faulty and does not respond properly).

B - 4 MicroSat Selection Tests

It is the function of the master CPU to assign and schedule user programs to an available satellite. Therefore, it must test satellite boards for working condition and record their current status.

When the master assigns a satellite to a particular user program, a busy-status flag is set for that satellite. When the master is commanded to assign another user program it consults the list of satellites, beginning with the first, and checks the flag for busy. If busy, the master checks the next satellite, and so on.

If the satellites are all busy, the master will check its own base memory segment, the Absolute Transient Program Area, for a not-in-use status. If the Absolute TPA is also in use, a diagnostic message is returned to the user's terminal where the program was submitted:

```
Abs Tpa not free
```

If the master does locate a free satellite, the satellite is selected and initialized through the interrupt hardware of Segment 16 (see the description for this segment for the discussion of the use of the /NMI line and Direct Memory Access by the master). The master waits for initialization to finish.

The satellite is again accessed through DMA. The master leaves an ASCII message in the satellite's incoming mailbox buffer and a SENDDATA mailbox operation code. If the satellite is functional, it moves the ASCII message from the buffer to memory location 0100 hex and signals the master it has performed the operation by writing the proper completion code to the mail box.

The master recovers the code during the next DMA. The recovered information is compared with the expected completion code. If they match, the satellite passes satellite selection tests and a COM type program will be loaded and run.

If the satellite fails the tests, the master will increment the list to the next satellite physically present and restart the test cycle. In this way a defective satellite board will repeatedly be tested and rejected. When this happens the only effect on the user is that there are fewer free and available satellites. No workstations (consoles) are brought down by a malfunctioning MicroSat card.

B - 5 Software Initialization

Initialization of the satellite board software is accomplished by a satellite firmware routine appropriately named INIT, which can be called by either COLDBOOT or WARMBOOT.

When the master does DMA and strobes the /NMI line, a Non-Maskable Interrupt is latched in. During the DMA period for initialization, the master writes a three byte jump at location 0066 hex. At the end of the DMA, the latched /NMI forces the processor to begin execution at location 0066. The jump is to the COLDBOOT routine.

COLDBOOT sets the stack pointer to zero (stack empty), zeros out the CP/M base page, zeros out the mailbox buffer, and calls INIT .

INIT writes three jump instructions; one each at location 0000, 0005, and FB06. It sets the default DMA address in the outgoing mailbox to 0080, and it checks the terminal buffer at FBC0 for leftover characters and prints them. Then it zeros the terminal buffer pointer and buffered character counter, and fills the terminal buffer with "$" characters. Finally, it sets the clock flag in the mailbox buffer.

WARMBOOT will also call INIT, but it does little else besides sending the master a "terminate program" operation code via the mailbox.

B - 6 MicroSat Memory Organization

Each MicroSat card in an M/NET system is a stand-alone computer that communicates with a system master via the S-100 bus. The memory is organized in a similar fashion to a single-user CP/M system. Notice that the top of the transient program area is at FAFE. This avails each user to a TPA of over 62K bytes.

MicroSat Memory Organization

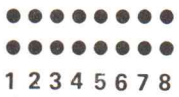
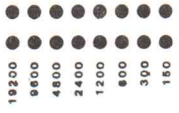
FC30 - FFFF	BIOS and BDOS Emulator Program
FC00 - FC30	BIOS Jump Table
FBC0 - FBFF	Terminal Buffer
FB14 - FBBF	Remainder of Mailbox Buffer
FB10 - FB13	Incoming Mailbox (to MicroSat from Master)
FB0A - FB0F	Outgoing Mailbox (to Master from MicroSat)
FB09	Clock Flag
FB06 - FB08	JMP BDOS (this location is BUFFER+6)
FB00 - FB05	Zeros (start of Mailbox buffer)
0100 - FAFF	Transient Program Area (NOTE: stack grows down from FAFF)
0080 - 00FF	Default DMA Address and Command Line Tail
005C - 007F	Default FCBs
004C - 005B	Reserved
0040 - 004B	Scratchpad area
003B - 003F	Free
0008 - 003A	Reserved
0005 - 0007	JMP BUFFER+6 (jump to BDOS)
0003 - 0004	Free
0000 - 0002	JMP WARMBOOT

APPENDIX C

REFERENCES FOR THE DBOS BASED M/NET SYSTEMS

C - 1 MicroSat Jumper Settings for DBOS

The following table lists the proper MicroSat jumper settings for DBOS based M/NET systems. The MicroSat will not work unless these jumpers are set as described below. Place the MicroSat before you with the component side up and the S-100 fingers toward you when comparing the illustrations to the board.

Jumper	Setting	Description
JP1		Connects S-100 /NMI to Z80A /NMI
JP2		NANDs SPn with DMARQ to allow RAM refresh during DMA
JP3		Selects MicroSat number (different for each MicroSat card in a system)
JP4		PRDY to Master to install wait state during master/satellite DMA
JP5		Set for Intel compatible 2708 PROM (these jumpers are irrelevant in DBOS base systems)
JP6		Cut for a non-standard 2K PROM
JP7		Sets baud rate for USART (not implemented as of this edition of this manual)
JP8		Enables cold-start /POJ for PROM (unnecessary in DBOS based systems)
JP9		Always set to select RD
JP10		Enables TxRDY and/or RxRDY for Z80A /INT
JP11		WAIT state for PROM reads (unnecessary in DBOS based systems)

C - 2 DBOS Operation

DBOS -- Data Base Operating System -- is a CP/M compatible operating system developed specifically for M/NET's multi-processor architecture. As the name implies, DBOS uses data base concepts for file AND data management.

Similar to MP/M based systems, each MicroSat has a stripped down operating system present on-board loaded by the master into the top of its memory block when the system is reset. This program controls program execution trapping CP/M BDOS calls as in MP/M based systems. (DBOS is completely CP/M compatible.) Messages relating the nature of the call are deposited in a dedicated portion of memory for pick-up by the master. The size of this operating system and the window described below leaves a transient program area of around 62K for user programs.

There are several substantial differences regarding satellite utilization between MP/M and DBOS. (Refer to the M/NET Installation Guide and the DBOS User's Guide for a description of the operational differences.)

1) MicroSat polling is used instead of interrupts for I/O requests; and
2) Data transfers are performed through a window/buffer.

Addressing each difference separately:

1) MicroSat Polling

Instead of an interrupt, each MicroSat is interrogated for a message on a rotating basis. That is, the master processor board maintains a table of currently active satellites and polls each window on a regular basis to determine if an I/O request has been made. This brings up an important difference between MP/M and DBOS based M/NET systems: each MicroSat card is dedicated to a specific workstation in DBOS systems.

2) Data Transfer Windows

DMA is used in DBOS based systems to transfer data between master and satellite. The data blocks can be up to 1024 bytes in length and are transferred from/ transfer data between master and satellite. The data blocks can be up to 1024 bytes in length and are transferred from/into a RAM window -- a dedicated portion of the MicroSat's memory. From here, the satellite control program moves the data into the appropriate locations of memory. Because of the extended data block size, DMA must be interrupted to perform RAM refresh. Consequently, a refresh cycle is performed between transferred bytes.

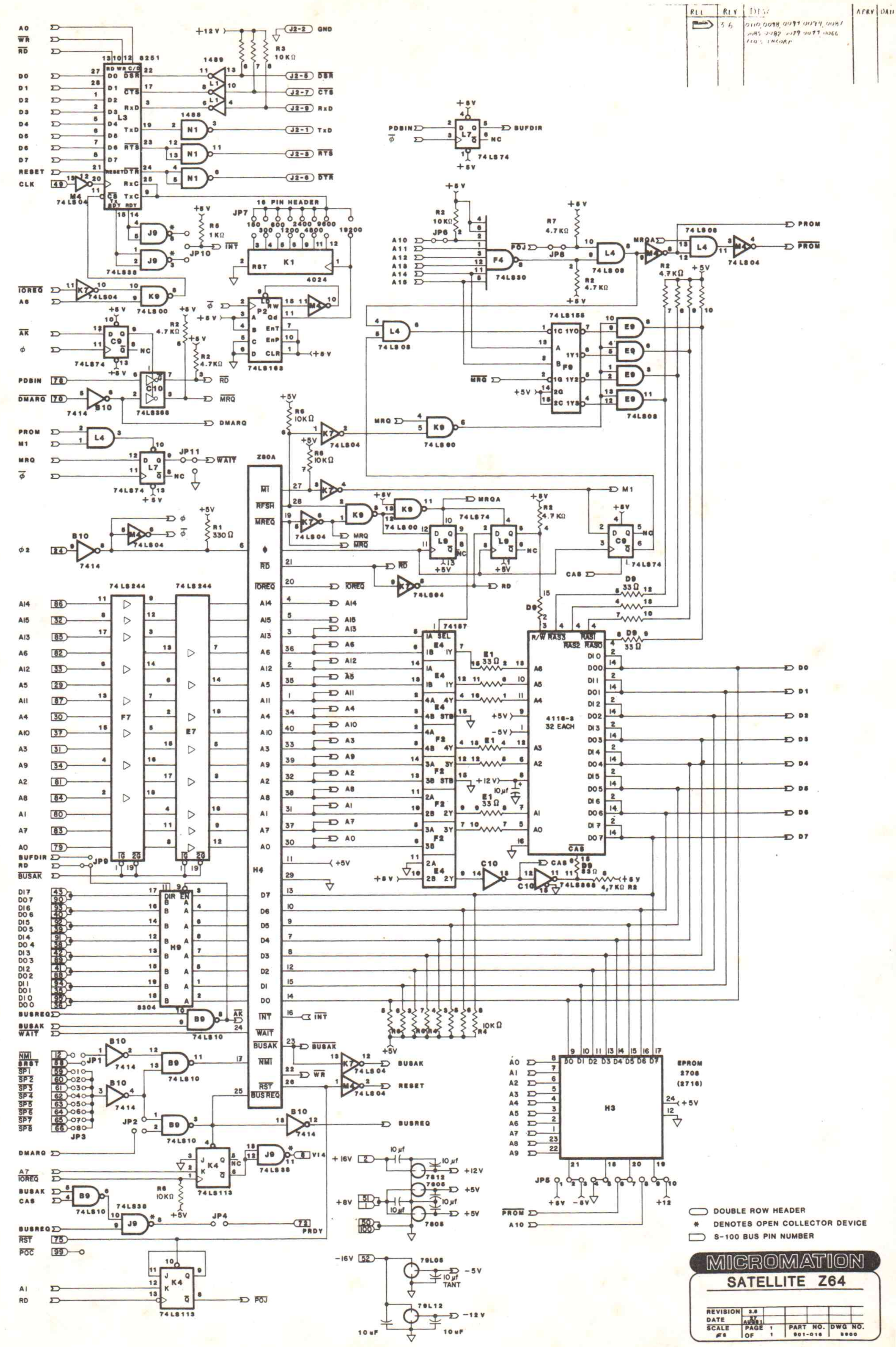
APPENDIX D

REVISION LOG

Manual Number	Name
D01 016	MicroSat Theory of Operation

Revision Number	Date	Notes
1	11/20/1981	Initial release of manual for Rev 3.6 MicroSat board.

NOTES


MICROMATION
SATELLITE Z64
DOUBLE ROW HEADER
* DENOTES OPEN COLLECTOR DEVICE
S-100 BUS PIN NUMBER
Z80A
EPROM 2708 (2716)
4116-3 32 EACH