Z80 SINGLE BOARD COMPUTER

SBC880

REFERENCE MANUAL

COPYRIGHT NOTICE

Copyright (c) 1982 by COMPUTIME. All Rights Reserved Worldwide. No part of this publication may be reproduced, transmitted, transcribed, stored in a retieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the express written permission of COMPUTIME, 8614 Hamilton Ave. Huntington Beach, California 92646 U.S.A.

DISCLAIMER

SUNTRONICS CO., INC. makes no representations or warranties with respect to the contents of this manual itself, whether or not the product it describes is covered by a warranty or repair agreement. Further, SUNTRONICS CO., INC. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of SUNTRONICS CO., INC. to notify any person of such revision or changes, except when an agreement to the contrary exisits.

TABLE OF CONTENTS

SECTION I

GENERAL DESCRIPTION

INTRODUCTION

The SBC880 Processor Board is a powerful Z80 based design which is compatible with the IEEE S100/696 bus standard. The SBC880 contains enough features to allow its use as a stand alone single board system or as the main CPU board in a larger system.

The board is manufactured using quality components that are conservatively rated to assure long life. All boards are burned in and tested to assure that the board will work properly when you receive it.

Before installing the board read this manual and become familiar with the SBC880 features and options. The board comes configured for a 2708 EPROM. If you want to use one of the other type of EPROM then refer to Section V Board options for modifications.

FEATURES

An on board EPROM can be addressed on any 1K or 2K boundary. A 1K by 8 static RAM may be used in place of the EPROM if desired. Power-on jump is available directly to the EPROM (or RAM). The EPROM may optionally be used in shadow mode to allow the full use of 64K or more of RAM. Devices that can be used in the EPROM location (U23) are the 2708, 2716 EPROMS or the 4118 static RAM. In addition to the EPROM, an additional 1K of static ram is provided and it can be located on any 1K boundary. If the EPROM is not used and the static ram is used instead at the EPROM location, a total of 2K bytes of ram may be present on the board.

The board is equipped with a USART and a RS232 interface. The baud rate is programmable by means of a programmable timer. All model signals required by terminal type equipment is provided for and terminal equipment may be connected directly to the RS232 connector(J1). Reverse channel capability is available for use with buffered type devices such as printers. The reverse channel is occasionally needed as a busy or ready indication from the connected device. It can sense things such as out of paper or ribbon. A 4MHz crystal provides all system timing and can be selected for 2 or 4 MHz operation.

A DMA capability is provided as well as a means of having the MWRT signal generated on the CPU board or elsewhere in the system under control of DMA logic or a front panel.

Two prgrammable timers are available for use by user programs. The timer output and controls are available at the parallel I/O connector J2. A parallel input and a parallel output port is available for use at connector J2.

All S-100 bus signals are fully buffered and regulators are used for all on board voltages to assure an electrically clean and stable design. A quality PC board is used with solder mask on both sides, plated through holes and gold plated contacts.

SECTION II

FUNCTIONAL DESCRIPTION

Z80A CPU

The SBC880 is a single board microcomputer designed around the Z80 microprocssor. The Z80 provides the major control signals required to read and write to memory and the I/O ports. A 16 bit address bus and a 8 bit bi-directional data bus are generatedby the Z80. The SBC880 can be run at either 4 or 2 MHz by changing a simple jumper.

OSCILLATOR

A crystal controller circuit provides all the timing for the S100 bus, the Z80A CPU, the Baud rat timer, and the two programmable timers. Associated with this circuit are the wait state generator for the EPROM and reset circuitry which also generates a power-on clear signal.

STATUS AND CONTROL BUFFERS

The status and control buffers interface the CPU status and control signals to the S-100 bus. These buffers may be tri-stated by a DMA device to allow a transfer of data between the DMA device and memory. The DMA device assumes control on thestatus and control for the duration of the DMA transfer. When a DMA access is requested by activating the CPU DMA request signal, the acknowledgment signal from the CPU is made available on the S-100 bus.

ADDRESS BUFFER

The address buffer is a 16 bit tri-state buffer which drives the CPU's sixteen address bits to the S-100 bus and to other circuitry on the CPU board. The buffer is also tri-stated by DMA devices when a transfer of data is to occur. The DMA device will then provide the address for the duration of the data transfer.

DATA OUT BUFFER

The data out buffer is an 8 bit tri-state buffer which drives the eight data bus signals from the CPU to the S-100 data out bus. The data out bus will only contain valid data during memory write or I/O output cycles. The data out bus will be tri-stated by DMA devices when they are transferring data to memory.

DATA IN BUFFER

The S-100 data in bus is provided to the CPU during memory read or I/O input cycles to devices external to the board. The data in buffers are disabled during memory write or I/O output cycles when the CPU is driving data to the data out buffers. The data in buffers are disabled whenever devices on the CPU board are being accessed to allow the device being accessed to place data on the CPU bus.

MEMORY DECODE AND CONTROL

The memory decode and control circuitry decodes the high order address bits and selects the EPROM or static RAM that is located on the board.

EPROM

The EPROM can be a 1K (2708 type), 2K (2716 type), or a 1K by 8 static RAM (4118 type). The EPROM may be selected on any 1K or 2K boundary and the optional ram can be selected on any 1K boundary.

1K BY 8 STATIC RAM

The 1K of on-board ram may be selected to any 1K boundary. The type of ram provided is static (2114 type) and requires no refresh. This ram may be used to hold the stack when running diagnostic tests on a bad dynamic ram board or it allows the use of the SBC880 as a stand alone system.

I/O DECODE AND CONTROL

The I/O decode and control circuitry decodes the lower 8 bits of the address bus to determine when the USART, timer or parallel I/O ports on the board are being address for input or output operations.

SERIAL I/O

The serial I/O provides asynchronous communication via a RS232 interface. The baud rate is provided by a programmable timer. The USART is a 8251.

PROGRAMMABLE TIMERS

Two programmable timers are available for use on the board. An 8253 or 8254 timer can be used. The timers are clocked from the crystal controlled clock oscillator on the board.

PARALLEL I/O PORTS

An 8 bit parallel output port and a 8 bit parallel input port are provided on the board. The ports are implemented with TTL type circuitry (74LS373 and 74LS374).

SECTION III

INTERFACES

EPROM INTERFACE

The EPROM will be selected for access under the following conditions:

1. When power-on jump is enabled (I to U present) and the power-on jump latch is set. The EPROM U23 is unconditionally selected until the latch is reset. The power on latch is set any time the system is powered up or the system reset button is pressed. The power-on latch is reset when a memory read operation is performed and the address being read compares to the switch settings of the EPROM select switch SW3. For 1K x 8 EPROM (2708 type) the 8131 (U30) compares address bits 10 through 15 to the SW3. For 2K x 8 EPROM (2716 type) U30 compares address bit 11 through 15 to the SW3.

2. When not using the phantom option (Q to R present) and a memory read operation is performed with an address that compares with the EPROM switch settings (as detected by the 8131 comparator)

The EPROM may optionally be replaced by a 4118 RAM. When the RAM is used the MEMRD signal is replaced by MRQ so that the RAM signal is accessed during memory read and write operations. The CPU WR/ signal is jumpered to the RAM WE/ input to allow the CPU to write data into the RAM during memory write cycles when the RAM is selected.

The EPROM or optional RAM is directly connected to the CPU bi-directional data bus and only appears on the S-100 bus indirectly through the data out bus drivers. NOTE that only the CPU can directly access the EPROM or optional RAM. It is not necessary to have the S-100 bus in operational condition to successfully access the EPROM or optional RAM. This feature allows diagnostic programs to be run in EPROM to diagnose a failing S-100 bus. The board can communicate with a terminal and run diagnostic tests even when the S-100 bus is completely inoperative.

PHANTOM MODE

When the phantom EPROM option is used (Q to R cut) the EPROM is only selected after a power-up or when the system reset is pressed and the power-on latch is set. The EPROM will be selected for all memory read operations that occur while the power-on latch is set. During this time a memory write operation will address memory external to the board in a normal fashion. Likewise I/O input and output cycles are unaffected by the power-on latch. Therefore, the program in the EPROM can be used to boot data from an I/O device into memory after a power-up or system reset operation. The EPROM select switch can be set to detect the starting address of the code that the EPROM program boots into memory. A jump to the starting address will then be detected by the 8131 comparator and will reset the power-on latch. When the power-on latch is reset, the EPROM can not be accessed (the comparator cannot select the EPROM because Q to R is open). The data in memory is now accessed in a normal fashion and the EPROM effectively disappears from the system until needed at the next power-up or system reset operation.

RAM INTERFACE

A 1K x 8 block of static RAM is implemented using two 2114 RAM chips on the board. The RAM is selected by a 8131 comparator that compares address bits 10 through 15 to the RAM select switch. When a memory operation is performed by the CPU (MRQ active) and the comparator detects a match between the address and the RAM switch, the CS/ lead goes active (low) at the RAM chips. When a memory write operation is performed the CPU WR/ signal is active (low) at the WE/ inputs of the RAM chips to allow the CPU to write data into the RAM. The RAM on the board is directly connected to the CPU bi-directional data bus and can only be accessed directly by the CPU. The RAM data is indirectly available at the S-100 data out bus. The on-board RAM, EPROM and USART will function independently of the S-100 bus and this allows diagnostics to be performed by the board when the S-100 bus is inoperative. An example of this feature would be running a memory diagnostic on the board while diagnosing a dynamic RAM board that contains the balance of the system memory. The diagnostic programs can use the on-board static RAM for scratchpad and stack operations. The diagnostic routines would run properly even if the RAM board being diagnosed was affecting signals on the S-100 bus. In normal use, the on-board RAM may be located in the same address space as other memory in your system with no conflicts between the memory devices. This will be necessary if your system uses a full 64K of RAM. Whenever the on-board RAM is accessed during a memory read operation the S-100 bus data in bus receivers are disabled and the on-board static RAM is allowed to supply data directly to the CPU. Thus an external device responding to the same memory cycle would have its data ignored by the board and the on-board RAM would supply data instead. When your system RAM is inoperative, the on-board RAM will be available to help you get your system going again.

I/O INTERFACE CIRCUITRY

The CPU I/O devices are selected by an 8131 comparator. The 8131 compares address bits A3 through A7 to the I/O select switch and looks for IORQ to be active (high) indicating that an I/O access is in process. The CPU uses address bits A0 through A7 to address I/O devices. Address bits A3 through A7 are tested by the 8131 comparator and address bits A1 and A2 are decoded with gates to select the individual I/O devices on the board as follows:

RD	WR	A2	A1	A0	Device Selected	Operation
1	0	0	0	0	8253 Timer	Read baud rate time
0	1	0	0	0	" "	Write baud rate time
1	0	0	0	1	" "	Read counter 1
0	1	0	0	1	" "	Write counter 1
1	0	0	1	0	" "	Read counter 2
0	1	0	1	0	" "	Write counter 2
1	0	0	1	1	" "	Illegal
0	1	0	1	1	" "	Write mode word
1	0	1	0	0	Input Port	Read input port
0	1	1	0	0	Output Port	Write output port
1	0	1	0	1	Input Port	Read input port
0	1	1	0	1	Output Port	Write output port
1	0	1	1	0	8251 USART	Read data register
0	1	1	1	0	" "	Write data register
1	0	1	1	1	" "	Read status registe
0	1	1	1	1	" "	Write control register

The final I/O port address for each device is determined by the I/O select switch which determine what state of address bits A3 through A7 will cause the I/O devices on the board to be selected. NOTE that for some devices during an input cycle (RD active) a different operation takes place than for an output cycle (WR active) at the same address. When A2 is high and A1 is low, an input cycle selects the input port, and an outputcycle selects the output port. Adderss bit A0 is ignored when accessing the parallel input and output ports and the same devices are selected with A0 high or low.

8253 TIMER INTERFACE

The board provides a 2Mhz clock from the clock oscillator circuitry to the count inputs of timer zero and timer one of the 8253. Timer zero is used as a programmable baud rate generator for the 8251 USART and its output is connected directly to the transmit and receive clock inputs of the USART. The gate input to timer zero (G0) is tied active (high) to permanently enable this counter. The output of timer one (01) is connected to the input of timer two (CS) and the two timers can be used together to form a 32 bit counter/timer. The gate inputs, count inputs, and count outputs are available for use at connector J2. Refer to Appendix C for a description of the 8253 functions.

8251A USART INTERFACE

Timer zero of the 8253 divides the 2MHz clock down and provides the transmit and receive clocks (TXC and RXC) to the USART. Transmit data at the RS232 connector J1 pin 2 is level shifted by a 1489 RS232 receiver and applied to the receive data input (RXD) of the USART. Transmit data (TXD) from the USART is level shifted by a 1488 RS232 transmitter to J1 pin 3. The reverse channel transmit J1 pin 11 input at the RS232 connector is level shifted by the 1489 and provided to the DSR/ input of the USART. This input can be sensed as a status bit in the status register and has no other affect on the operation of the USART. This allows programs to sense not ready or buffer full conditions on serial I/O devices. The CTS/ input of the USART is tied active (low) permanently at the input of the USART. This tells the USART that the RS232 interface is always ready to transmit data. The actual part used for the USART will be an 8251A or 9551. The older 8251 will not be used on the board. Refer to Appendix B for a functional description of the USART device.

The RS232 connector is configured to allow direct connection to a device without modems. Any modem signals required by the connected device will be satisfied by jumpers on the board. The RS232 connector jumpers together the following RS232 signals at the connector. Request to send is jumpered to Clear to send Pins 4 and 5. Data terminal ready is jumpered to Data set ready and Carrier detect Pins 6, 8 and 20.

BAUD RATE DIVISORS

When the baud rate timer is initialized a divisor must be selected that will divide the 2MHz clock to a frequency that is 16 times the baud rate desired. The following list of baud rate divisors will be of help in selecting the one for your application.

Baud Rate	Divisor
9600	13
4800	26
2400	52
1200	104
600	208
300	417
150	833
110	1136

PARALLEL I/O PORTS

The parallel output port is implemented with a 74LS374 edge-triggered register. The register outputs are buffered on the chip and need no additional buffering. The clock to the output port register is provided to the I/O connector J2. Output data is latched at the rising (low to high) transition of this clock. This clock will transition every time the output port is selected during an I/O output cycle.

The parallel input port is implemented with a 74LS373 octal transparent latch. The tri-state buffers on this chip provide data directly to the CPU bi-directional data bus when the input port is selected during the I/O input cycle. The data present at the latch inputs when the latch select strobe goes from high to low (input port selected) will be latched and then presented to the CPU data bus. The latch strobe signal is made available at the I/O connector J2.

SECTION IV

BOARD OPTIONS

OPTION 1 ON BOARD EPROM - 2708 EPROM

The SBC880 comes etched for the 2708 EPROM. These defaults are as follows:

1. Z to K open
2. Y to G open
3. F to G shorted
4. H to I shorted
5. J to K shorted
6. L to M open
7. P to O shorted
8. P to N open
9. L to K open
10. H to M open
11. V to W shorted
12. X to V open

OPTION 2 - 2716 EPROM

TI 3 Voltage EPROM

1. F to G shorted
2. H to I shorted
3. J to K shorted
4. L to M shorted
5. P to O open
6. P to N shorted
7. L to K open
8. H to M open
9. V to W shorted
10. X to V open
11. Y to G open
12. Z to K open
13. Switch 6 of SW3 closed

Intel +5 volt EPROM

1. F to G open
2. H to I shorted
3. J to K open
4. L to M open
5. P to O open
6. O to N shorted
7. L to K shorted
8. H to M shorted
9. V to W shorted
10. X to V open
11. Y to G open
12. Z to K open
13. Switch 6 of SW3 closed
14. G to +5

OPTION 3 - 4118 RAM substituted for EPROM

1. F to G open
2. H to I shorted
3. J to K open
4. L to M open
5. L to K open
6. H to M shorted
7. P to O shorted
8. P to N open
9. V to W open
10. X to V shorted
11. Y to G shorted
12. Z to K shorted

OPTION 4 - Power-on Jump no Phantom Mode

An EPROM must be present to use the power-on jump feature. The board is etched for the 2708 EPROM and the power-on jump with no Phantom mode as follows:

1. T to U shorted
2. Q to R shorted

OPTION 5 - Power-on Jump with Phantom Mode

The Eprom must be present to use this option.

1. T to U shorted
2. Q to R open

OPTION 6 - No Power-on Jump

The EPROM or optional RAM may be used with this option.

1. T to U open
2. Q to R shorted

OPTION 7 - No EPROM

To disable address selection of the EPROM entirely.

1. Q to R open
2. T to U open

OPTION 8 - MWRT generated by CPU

This option is etched on the board.

1. C to E shorted
2. D to E open

OPTION 9 - MWRT generated by external devices

1. C to E open
2. D to E open

OPTION 10 - MWRT generated by CPU and external devices

1. C to E open
2. D to E shorted

SECTION V

DETAIL DESCRIPTION

Refer to the SBC880 schematic while reading the descriptions of each functional block that follows.

ADDRESS BUS

The address bus of the CPU is buffered using 74LS241 devices. During DMA operations the DMA device will drive S-100 pin 22 (ADDSB/) low to tri-state the address bus drivers. The DMA device can then place its own address on the bus.

DATA IN BUS

During memory read or I/O input operations, the S-100 data in bus is received and driven to the CPU bi-directional data bus by a 74LS241 type device. Circuitry is provided (1/2 of 7420) to disable the data in buffers under the following conditions:

1. EPROM or optional ram selected
2. On-board static ram selected
3. Programmable timer selected
4. Parallel I/O port selected
5. Memory write or I/O output operation in progress

DATA OUT BUS

The CPU bi-directional data bus is driven to the S-100 data out bus by 74LS241 type buffers. The buffers provide write data from the CPU to devices on the S-100 bus during memory write operations. A DMA device wishing to transfer data on the data out bus will drive pin 23 (DODSB/) on the S-100 bus to a low state. This will disable the data out buffers on the board and allow the DMA device to place data on the bus.

STATUS SIGNALS

Status signals SM1, SMEMR, SINP, SOUT, SINTA, and SWO/ are provided to the S-100 bus by a 8097 type tri-state buffer. A DMA device may drive pin 18 (STATDSB/) low to tri-stat the status buffer to allow the DMA device to gain control of the bus. Control signals PSYNC, PWR/, and PDBIN are provided to the S-100 bus by one section of a 8097 type tri-state buffer. A DMA device may drive pin 19 (C/CDSB/) low to disable this buffer and gain control of the bus. The MWRITE signal is provided to the bus by the other half of the 8097 buffer that was used by the control signals. This buffer may be controlled in several ways: The board as you received it has this buffer permanently enabled so that the CPU board is always the source of the MWRITE signal. If your system contains another device that is to be the source of the MWRITE signal, then you may cut the etch between points C and E to disable the buffer on the board. If desired, the MWRITE buffer may be disabled by the STATDSB/ signal when a DMA device is using the bus. This option is enabled by cutting the etch between points C and E, and then installing a jumper between points D and E. This option requires that the DMA device have a buffer to provided the MWRITE signal that is enabled by the STATDBS/ signal or a signal with the same timing. This is necessary to prevent floating the MWRITE signal on the S-100 bus while transferring control of the bus. Data in memory may be overwritten if this signal is left floating.

The signal SOUT is active (high) when WR/ and IORQ/ signals are active (low) at the CPU. SINP is active (high) when the signals RD/ and IORQ/ are active (low) at the CPU. SMEMR is active (high) when the signals RD/ and MREQ/ are active (low) at the CPU. MWRITE is active (high) when signals WR/ and MREG/ are active (low) at the CPU. SM1 goes active (high) when M1/ goes active (low) at the CPU.

OTHER STATUS SIGNALS

SINTA goes active (high) when M1/ and IORQ/ go active (low) at the CPU. SWO/ goes active (low) when the CPU is not performing any input cycles. This signal is used to provide an early indication that a write or output cycle is going to take place. SWO/ is active (low) when RD/ is inactive (high) or SINTA is inactive (low) will be active (high) at the CPU. The signal SXTRQ/ is not generated by the board because only 8 bit data is required by the CPU. SXTRQ/ is used to request 16 bit data transfers. SHLTA goes active (high) when HLTA/ goes active (low) at the CPU.

CONTROL SIGNALS OUTPUT

The signals PSYNC, PWR/ and PDBIN are tri-stated when C/CDSB/ goes active (low). C/CDSB/ will be driven low by a DMA type device when the device wants to take control of the bus. The PSYNC signal goes active (high) momentarily at the start of any valid I/O or memory cycle. The timing for this signal is developed by a flip flop and produces timing as defined by the IEEE S-100 specification. The PSYNC signal is not produced during CPU memory refresh cycles. PWR/ goes active (low) when WR/ goes active (low) at the CPU. The signal PDBIN goes active when a memory read of I/O input cycle is in process or when an interrupt is being acknowledged (SINTA high). PDBIN is active (high) when RD/ is active (low) at the CPU or when SINTA is active (high). PHLDA goes active (high) when BUSAK/ goes active (low) at the CPU. This signal acknowledges a DMA request and indicates that the requesting device with the highest priority may take control of the bus. The CPU generates this signal in response to the signal BUSRQ/ going active (low). BUSRQ/ goes active (low) when a DMA device drives the S-100 signal PHOLD/ active (low). BUSAK/ only goes active when BUSRQ/ has gone active and the CPU is at a point in its operation where a DMA access can be perfomed properly. PHLDA/ is always driven and cannot be tri-stated.

CONTROL SIGNALS INPUT

The IEEE S-100 bus signal SIXTN/ is not used by the board because only 8 bit accesses are required by the CPU. The signal SIXTN/ is a response to a request for a 16 bit memory access. Since a 16 bit access is never requested by the board, this signal is ignored. The signals XRDY or PRDY when driven low will make the WAIT/ signal go active (low) at the CPU. The EPROM wait state generator can make the WAIT/ signal active for one clock cycle during accesses to the on-board EPROM. The EPROM wait state generator must be enabled by a jumper option. PRDY/ is normally used by slow memory or I/O devices to extend an access cycle by inserting wait states at the CPU. The device being accessed holds PRDY/ active (low) for the number of clock cycles (wait states) desired. XRDY is normally used by front panel type devices to halt or single step the processor. PWAIT/ goes active (high), WAIT/ goes active (low). When PINT/ is driven active (low) at the S-100 bus , the signal INT/ will be active (low) atthe CPU.This is one of the maskable interrupt request input to the CPU. When NMI/ is driven active (low) at the S-100 bus, the signal NMI/ goes active (low) at the CPU. This is the non-maskable interrupt request input to the CPU When PHOLD/ is driven active (low) at the S-100 bus, the signal BUSRQ/ will be active (low) at the CPU. The PHOLD/ signal is used by the DMA devices to request access to the bus.

DMA CONTROL LINES

The primary lines used to tri-state the bus drivers for DMA operations are DODSB/, ADDSB/, STATDSB/, and C/CDSB. DODSB/ tri-states the data out bus drivers when it is driven active (low). ADDSB/ tri-states the address bus drivers when it is driven active (low). STATDSB/ tri-states the status signals SOUT, SINP, SMEMR, SM1, SINTA and SWO/ when it is driven active (low). MWRITE may be tri-stated by STATDSB/ if selected by a jumper option. C/CDSB/ tri-states the signal PSYNC, PWR/ and PDBIN when driven active (low). When a DMA device is granted access to the bus by PHLDA signal going active (high) it will normally activate the DMA control signals and drive its own signals on to the bus.

SYSTEM POWER LINES

A positive 8 volts DC should be present on S-100 bus pins 1 and 51. This voltage is regulated on the board to develop +5 volts. The regulator is decoupled on its input by a 1.5 uf capacitor and the +5 volt output is decoupled by .1 uf capacitors at various places around the board. A negative 16 volts DC should be present at S-100 bus pin 52. The voltage is regulated by two regulators on the board to develop -12 volts and -5 volts. Both regulators are decoupled with 1.5 uf capacitors at thier inputs and outputs. A positive 16 volts DC should be present at S-100 bus pin 2. This voltage is regulated on the board to develop +12 volts. The regulator is decoupled by a 1.5 uf capacitor at its input and a should be present on S-100 bus pins 50 and 53.

SYSTEM CLOCK

The board generates all timing from a 4 MHz crystal controlled oscillator. The 4 MHz clock is divided down to 2 MHz by a flip flop. The 2/4 MHs jumpers selects which clock rate is applied to the CPU and related circuitry. The selected clock is provided to the S-100 bus at 02. An inverted version of 02 is provided as 01. The 2 MHz clock is provided directly to S-100 bus as CLOCK/. This clock is always 2 MHz and is not affected by the 2/4 MHz jumper.

SYSTEM RESET FUNCTIONS

When PRESET is driven active (low) at the S-100 bus, a 100 uf capacitor on this line is discharged. This signal is normally driven low by the system reset button being pressed. The PRESET signal is synchronized to the system clock with a flip flop and is then applied to the following circuits:

1. The Z80 CPU
2. The 8251 USART
3. POC at the S-100 bus
4. The power-on jump latch

The reset signal will remain active (low) after the switch is released for approximately 470 milliseconds due to the time it takes to charge the 100 uf capacitor to a true level. This same circuit de-bounces the reset switch and provides a reset signal during power up.

APPENDIX A

USART - TIMER AND I/O ADDRESSING

Switch SW1 selects the base address range of 8 addresses

RANGE	SW1	SW2	SW3	SW4	SW5
00-07	X	X	X	X	X
08-0F		X	X	X	X
10-17	X		X	X	X
18-1F			X	X	X
20-27	X	X		X	X
28-2F		X		X	X
30-37	X			X	X
38-3F				X	X
40-47	X	X	X		X
48-4F		X	X		X
50-57	X		X		X
58-5F			X		X
60-67	X	X			X
68-6F		X			X
70-77	X				X
78-7F					X
80-87	X	X	X	X	
88-8F		X	X	X	
90-97	X		X	X	
98-9F			X	X	
A0-A7	X	X		X	
A8-AF		X		X	
B0-B7	X			X	
B8-BF				X	
C0-C7	X	X	X		
C8-CF		X	X		
D0-D7	X		X		
D8-DF			X		
E0-E7	X	X			
E8-EF		X			
F0-F7	X				
F8-FF					

First address	X0 or X8	-	Timer 0 Data
2nd "	X1 or X9	-	" 1 "
3rd "	X2 or XA	-	" 2 "
4th "	X3 or XB	-	" Control
5th "	X4 or XC	-	Parallel Input and Output Port
6th "	X5 or XD	-	" " " " "
7th "	X6 or XE	-	USART Data
8th "	X7 or XF	-	" Control

```
SW1 - 5   A7  ┌------
SW1 - 4   A6  |
SW1 - 3   A5  |        ---Base Address Selection
SW1 - 2   A4  |
SW1 - 1   A3  └------
          A2  ┌---------
          A1  |          --I/O Port Selection
          A0  └---------
```

	A2	A1	A0
Timer 0 Data	0	0	0
" 1 "	0	0	1
" 2 "	0	1	0
" Control	0	1	1
Parallel I/O	1	0	0
" "	1	0	1
USART Data	1	1	0
" Control	1	1	1

TABLE 1

CONNECTOR J1 SIGNALS

Pin No	Function
2	RS232 Transmit Data
3	RS232 Receive Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground
8	Carrier Detect
11	Reverse Channel Transmit
20	Data Terminal Ready

TABLE 2

CONNECTOR J2 SIGNALS

PIN NO	FUNCTIONS
1	
2	Output Port Data Bit 0
3	Output Port Data Bit 1
4	Output Port Data Bit 2
5	Output Port Data Bit 3
6	Output Port Data Bit 4
7	Output Port Data Bit 5
8	Output Port Data Bit 6
9	Output Port Data Bit 7
10	Signal Ground
11	Output Port Clock
12	Counter 1 Gate Input
14	Counter 2 Gate Input
15	Input Port Data Bit 0
16	Input Port Data Bit 1
17	Input Port Data Bit 2
18	Input Port Data Bit 3
19	Input Port Data Bit 4
20	Input Port Data Bit 5
21	Input Port Data Bit 6
22	Input Port Data Bit 7
23	Signal Ground
24	Input Port Strobe
	Counter 1 Output (also counter 2
25	input)
	Counter 2 Output

THIS PAGE LEFT BLANK

USART 8251 or 9551

OPERATION AND PROGRAMMING

The computer program controlling the USART performs the following tasks:

1. Output control codes
2. Input status
3. Outputs data to be transmitted
4. Inputs data that has been received

Control codes determine the mode the USART will opearate in and they are used to set or reset control signals output by the USART.

The status register contents will be read by the program monitoring the USART operation in order to determine error conditions, when and how read data, write data or output control codes. Program logic may be based on reading status bit levels, or control signals may be used to request interrupts.

INITIALIZATION

The USART may be initialized following a system reset or prior to starting a new serial I/O sequence. The USART must be reset following power-up and subsequently may be reset at any time following completion of one activity and preceeding a new set of operations. Following a reset, the USART enters an idle state in which it can neither transmit nor receive data.

The USART is initialized with two, three or four control words from the processor. Figure 1 shows the sequence of control words needed to initialize the USART for synchronous operation, the mode control is followed by one or two SYNC characters, and then a command.

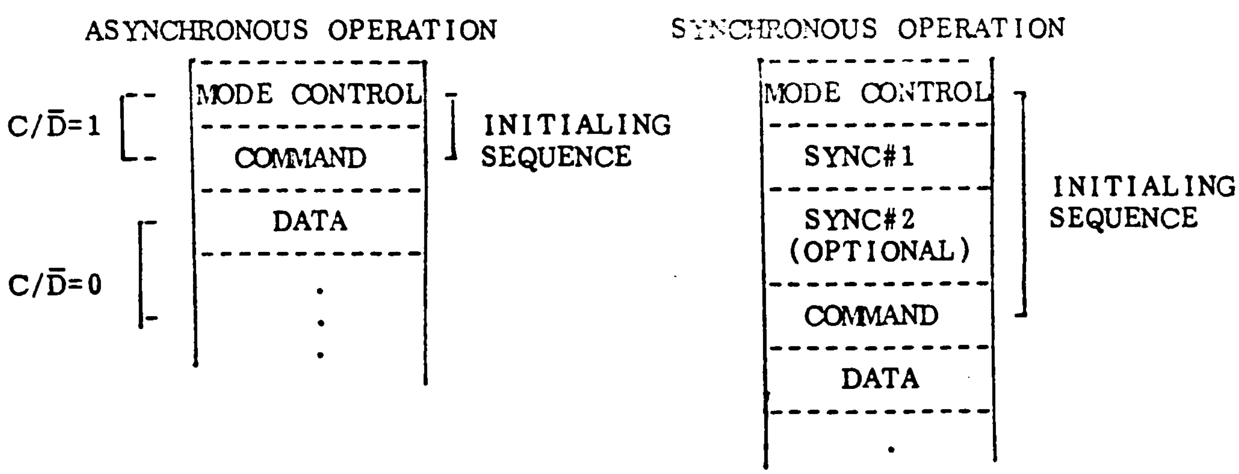


FIGURE 1. Control Word Sequence for Initialization.

Only a single address is set aside for mode control bytes, command bytes and SYNC character bytes. For this to be possible, logic internal to the chip directs control information to its proper destination based on the sequence in which it is received. Following a reset, the first control code output is interpreted as a mode control. If the mode control specifies synchronous operation, then the next one or two bytes (as determined by the mode byte) output as control codes will be interpreted as SYNC characters.

For either asychronous or synchronous operation, the next byte output as a control code is interpreted as a command. All subsequent byte output as control codes are interpreted as commands. There are two ways in which control logic may return to anticipating a mode control input; Following an external reset signal or following an internal reset command.

MODE CONTROL CODES

The USART interprets mode codes as shown in Figure 2 and 3. Control code bits 0 and 1 determine whether synchronous or asynchronous operation is specified. A non-zero value in bits 0 and 1 specifies asynchronous operation and defines the relationship between data transfer, baud rate and receiver or transmitter clock rate. Asynchronous serial data may be received or transmitted on every check pulse, on every 16th clock pulse, or every 64th clock pulse. A zero in both bits 0 and 1 defines the mode of operation as synchronous.

BIT NO	DESCRIPTION
0,1	00 = SYNC MODE
2,3	00 = 5 BITS PER CHARACTER
	01 = 6 BITS PER CHARACTER
	10 = 7 BITS PER CHARACTER
	11 = 8 BITS PER CHARACTER
4	0 = PARITY DISABLE
	1 = PARITY ENABLE
5	0 = ODD PARITY
	1 = EVEN PARITY
6	0 = SYNDET OUTPUT
	1 = SYNDET INPUT
7	0 = 2 SYNC CHARACTERS
	1 = 1 SYNC CHARACTER

FIGURE 2. Synchronous Mode Control Code

For synchronous and asynchronous modes, control bits 2 and 3 determine the number of data bits which will be present in each data character.

For synchronous and asynchronous modes, bit 4 and 5 determine whether there will be a parity bit in each character, and if so, whether odd or even parity will be adopted. Thus in synchronous mode a charater will consist of five, six, seven or eight data bits, an optional parity bit, a preceeding start bit, plus 1, 11/2, or 2 trailing stop bits. Interpretation of subsequent bits differs for synchronous or asynchronous modes.

Control code bits 6 and 7 in asynchronous mode determine how many stop bits will trail each data unit. 11/2 stop bits can only be specified with a 16 x or 64 + baud rate factor. In these two cases, the half stop bit will be equivalent to 8 or 32 clock pulses, respectively.

In synchronous mode, control bits 6 and 7 determine how character synchronization will be achieved. When SYNDET is an output, internal synchonization is specified, one or two SYNC characters, as specified by control bit 7, must be detected at the head of a data stream in order to establish synchronization.

BIT NO	DESCRIPTION
0,1	00 = INVALID
	01 = ASYNC MODE, 1 x BAUD RATE FACTOR
	10 = ASYNC MODE, 16 x BAUD RATE FACTOR
	11 = ASYNC MODE, 64 X BAUD RATE FACTOR
2,3	00 = 5 BITS PER CHARACTER
	01 = 6 BITS PER CHARACTER
	10 = 7 BITS PER CHARACTER
	11 = 8 BITS PER CHARACTER
4	0 = PARITY DISABLE
	1 = PARITY ENABLE
5	0 = ODD PARITY
	1 = EVEN PARITY
6,7	00 = INVALID
	01 = 1 STOP BIT
	10 = 1 1/2 STOP BIT
	11 = 2 STOP BIT

FIGURE 3. Asynchronous Mode Control Code

COMMAND WORDS

Command words are used to initiate specific functions within the USART such as, "reset all error flags" or "start searching for SYNC". Consequently, command words may be issued at anytime during the execution of a program in which specific functions are to be initiated within the communication circuit. Figure 4 shows the format for the command words.

BIT NO	DESCRIPTION
	TxEN
0	0 = DISABLE TRANSMISSION 1 = ENABLE TRANSMISSION
	DTR
1	1 = DTR OUTPUT IS FORCED TO 0
	RxE
2	0 = DISABLE RxRDY 1 = ENABLE RxRDY
	SBRK
3	0 = NORMAL OPERATION 1 = TxD IS FORCED LOW
	ER
4	1 = RESETS ALL ERROR FLAGS IN STAUS REGISTER (PE, OE, FE)
	RTS
5	1 = $\overline{RTS}$ OUTPUT IS FORCED TO 0
	IR
6	1 = RESET FORMAT
	EH
7	1 = ENTER HUNT MODE

FIGURE 4. Control Command.

Bit 0 of the command word is the transmit enable bit (TxEN). Data transmission from the USART cannot take place unless TxEN is set in the command register. Figure 5 defines the way in which TxEN, T x E ant T x RDY combine to control transmitter operation.

Bit 1 is the Data Terminal Ready (DTR) bit. When the DTR command bit is set, The DTR output connection is active (low). DTR is used to advise a modem that the data terminal is prepared to accept or transmit data.

Bit 2 is the Receiver Enable Command bit (R x E). R x E is used to enable the R x RDY output signal. R x E prevents the R x RDY signal from being generated to notify the processor that a complete character is framed in the Receive Character Buffer. It does not inhibit the assembly of data characters at the input, however. Consequently, if communication circuits are active, characters will be assembled by the receiver and transferred to the Receiver Character Buffer. If R x E is disabled,the overrun error (OE) will probably be set; to insure proper operation, the overrun error is usually reset with the same command that enables R x E.

Bit 3 is the Send Break Command bit (SBRK). When SBRK is set, the transmitter output (T x D) is interrupted and a continuous binary "0" level, (spacing) is applied to the T x D output signal. The break will continue until a subsequent command word is sent to the USART to remove SBRK.

Bit 4 is the Error Reset bit (ER). When a command Word is transmitted with the ER bit set, all three error flags in the status register are reset. Error Reset occurs when the command word is loaded into the USART. No latch is provided in the command register to save the ER command bit.

Bit 5 the Request To Send command bit (RTS). Sets a latch to reflect the RTS signal level. The output of this latch is created independently of other signals in the USART. As a result, data transfers may be made by the CPU to the transmit regester; and data may be actively transmitted to the communication line through T x D regardless of the status of RTS.

TxEN	TxE	TxRDY	
1	1	1	Transmit Output Registers and Transmit Character Buffer empty. TxD continues to mark if in the asynchronous mode. TxD will send sync pattern if in the synchronous mode. Data can be entered into buffer.
1	0	1	TransmitOutputRegisteris shifting a character . Transmit Character Buffer is available to receive a new byte from the processor.
1	1	0	Transmit Register has finished sending. A new character is waiting for transmission. This is a transient condition.
1	0	0	Transmit Register is currently sending and an additional character is stored in the Transmit Character Buffer for transmission.
0	0/1	0/1	Transmitter is disabled.

FIGURE 5. Operation of the transmitter section as a function of T x E, T x RDY and T x EN.

Bit 6 the Internal Reset (IR) causes the USART to returnto the idle mode. All functions within the USART cease and no new operation can be resumed until the circuit is reinitialized. If the operating mode is to be altered during the execution of the program, the USART must first be reset. Either the external reset connection can be activated or the Internal Reset command can be sent to the USART. Internal Reset is a momentary function performed only when the command is issued.

Bit 7 is the Enter Hunt command bit. The Enter Hunt mode command is only effective for the USART when it is operating in the synchronous mode. EH causes the receiver to stop assembling characters at the R x D input and start searching for the prescribed sync pattern. Once the "Enter Hunt" mode has been initiated, the search for the sync pattern will continue indefinitely until EH is reset when a subsequent command word is sent, when the IR command is sent to the USART, or when SYNC characters are recognized.

STATUS REGISTER

The Status Register maintains information about the current operation status of the USART. Figure 6 showes the format of the Status Register.

BIT NO	DESCRIPTION
0	TxRDY
1	RxRDY
2	TxE
3	PE - Parity error
4	OE - Overrun error
5	FE
6	SYNDET
7	DSR

FIGURE 6 Status Register

TxRDY signals the CPU that the transmit character buffer is empty and that the USART can accept a new character for transmission.

RxRDY signals the CPU that a complete character is holding in the receive character buffer register for transfer to the CPU

TxE signals the CPU that the transmit register is empty.

PE is the parity error signal indicating to the CPU that the character stored in the receive character buffer was received with an incorrect number of binary "1" bits.

OE is the receiver overrun error. OE is set whenever a byte stored in the receiver character register is overwritten with a new byte before being transfered to the CPU.

FE is the character framing error which indicates that the asynchronous mode byte stored in the receiver character buffer was received with incorrect character bit format, as specified by the current mode.

SYNDET is the synchronous mode status bit associated with internal SYNC detection.

DSR is the status bit set by the external data set ready signal to indicate that the communication data set is operational. All status bits are set by the function described for them. SYNDET is reset whenever the CPU reads the status register. OE, FE, PE are reset only by command.

THIS PAGE LEFT BLANK

APPENDIX C

PROGRAMMABLE INTERVAL TIMER

INTRODUCTION

The Programmable Interval Timer used with this processor board can do functions normally done by software timing loops, such as event counting, time out delays, variable frequency generation, real time clock. With a minimal amount of software overhead, the interval timer can free the CPU of the task of counting and do it faster.

The Programmable Interval Timer has three seperate 16-bit counters (refer to Figure 1) that can count at rates up to 2MHz. Counter 0 is dedicated as the programmable baud rate generator for the USART. Counter 1 and 2 are available for use by the programmer.

The counters can operate in six different modes:

MODE	0	Interrupt on Terminal Count
	1	Programmable One-Shot
	2	Rate Generation
	3	Square Wave
	4	Software Triggered Strobe
	5	Hardware Triggered Strobe

The counters count in binary or BCD in repetitive and single event modes - all synchronous to the CPU clock.

PROGRAMMING

Associated with each counter is one 16-bit write-only control word register and two 8-bit write-only counter latches. To program the counter you initialize the control register and then program the counter latches. The counters can be programmed in any order, as long as each control word is programmed before the counter latches for that particular counter. See Table 1.

A1	A0	
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word

TABLE 1 Counter Addressing

CONTROL WORD REGISTER

The 6-bit control word register controls the counter mode and read/write sequencing of the counter latches. When A0 and A1 are bpthe high D6 and D7 select the control register for each counter. See Table 2.

A1	A0	D7	D6	
1	1	0	0	Counter 0 CW
1	1	0	1	Counter 1 CW
1	1	1	0	Counter 2 CW
1	1	1	1	Illegal

TABLE 2 Control Word Register

It appears that only one register is being programmed because the address A1 and A0 is the same for all control registers. In actuality the upper two bits (D7 and D) of the data word select the individual registers. The lower six bits (D5, D4, D3, D2, D1, D0) are register information. See Table 3.

D5	D4	D3	D2	D1	D0
RL1	RL0	M2	M1	M0	BCD
0	0	Counter latching command			
0	1	Read/Load LSB latch			
1	0	Read/Load MSB latch			
1	1	Read/Load LSB, then MSB latch			

TABLE 3 Control Register Format

RLO and RL1 (D4 and D5) of the control word determine how the two counter bytes are to be accessed when the counter address is selected. They are also decoded to send a special instruction which latches the counter contents. M0, M1 and M2 (D1, D2, D3) of the control word determine in which of the six modes the counter is to operate. BCD (D0) selects binary or BCD counting.

COUNTER LATCHES

A0 and A1 in conjunction with RL0 and RL1 access the counter latches. A0 and A1 determine which of the three paires of counter latches are to be accessed; the read/load (RL) bits of the control word register determine the upper/lower byte selection. If only RL0 is set, the least significant byte is being programmed. If only RL1 is set then the MSB is being programmed. If both bits are set, then a sequence of two write programs the LSB and then the MSB latch. Using this read/load format then requires the performance of the two writes in sequence, if the device is to operate correctly.

It is not necessary to program all 16 bits of a counter, when either the lower or upper byte is a zero. Both latches are automatically cleared when the control word is programmed and remain zero until otherwise programmed.

OPERATING MODES

There are six available modes of counting. MODE 2 and 3 are repetitive and all others are single event modes. Table 4 contains a gate summary for the different modes.

MODE	0	1	2	3	4	5
Initiate count		X	X	X		X
Inhibit count	X		X	X	X	

TABLE 4 Gate Summary

MODE 0 INTERRUPT ON TERMINAL COUNT

In this mode a control word write or writing to any counter latch forces the output low. After the write to the counter is completed, it begins counting. At the completion of the count (counter equals zero), the output goes high and remains high until a new control word or count is loaded. Reloading the counter latches during counting suspends the current count. At the end of reloading, the counter begins counting with the new divisor, the gate input suspends counting when low, and enables counting when high.

MODE 1 PROGRAMMABLE ONE-SHOT

In this mode, the output is high when the counter is not counting. A rising transition of the gate input triggers the counter to begin counting which forces the output low. Upon completion, the output goes high. Since the counter is retriggerable, any rising edge on the gate causes the counter to restart at the beginning. The counter can be reloaded at anytime. Any subsequent trigger initiates the new count.

MODE 3 SQUARE WAVE

Similar to mode 2, except that the output remains high for half the count and low for half the count for even divisors, for odd numbers, the output is high for (N+1)/2 counts and low for (N+1)/2 counts. In other words, the remander of division by 2 is addes to the output high time. If the counter is reloaded while counting, the new divisor becomes effective after the next output transaction. The gate input functions identically to mode 2.

MODE 4 SOFTWARE-TRIGGERED STROBE

In this mode, the output is normally high. Loading the counter latch (es) initiates counting. If counting is in progress at the time of the load, the current count runs to completion and the subsequent count reflects the new value. Upon completion, the output goes low for one clock period.

MODE 5 HARDWARE TRIGGERED STROBE

This mode is the same as mode 1, except that the output is normally high and goes for one clock period upon completion of counting.

COUNTER READING AND LATCHING

The counters can be read in two ways. In one of them, issuing a normal read to the counter's specified address transfers the counter outputs directly to the data bus. If the counter is counting, the contents are counting continuously. For an assured reading of the actual counter contents, the counter must be inhibited by disabling the clock or alternatively by forcing the gate low, if it is in modes 0 or 4. Note that the counter latches are write-only and that the counter itself is read. In reading, as with writing, the read/load bits of the control word register determines the accessing of the counter contents.

The second method of reading the counter uses the counter latching command. Issued like a control word, this command performs a counter latching operation. Freezing the contents in an auxiliary register and giving a stable, readable value. Once latched, the contents can be read out at any time without affecting the counter operation. In operation the user simply issues the latch command for the particular counter (Table 5) at the desired point in time to latch the current contents. The saved counter contents are now read as though one were reading the counter latches.

NOTE: The latch command does not affect the programmed read/load format or mode, so that the bytes read remain as previously programmed by the control word.

A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
1	1	0	0	0	0	X	X	X	X	Latch counter 0
1	1	0	1	0	0	X	X	X	X	Latch counter 1
1	1	1	0	0	0	X	X	X	X	Latch counter 2

TABLE 5 Latch Command

APPENDIX D

Parallel Printer Installation on the SBC-880

DESCRIPTION

Describes the basic interface of a "Centronics" parallel printer including both the cable pin-out, software driver example, and known limitations.

CABLE FABRICATION

SBC-880 parallel port			ribbon cable color		Amphenol 53-30360
pin 8	out	B7	green #2	*strobe	pin #1
pin 1	out	B0	brown #1	data 1	pin #2
pin 2	out	B1	orange #1	data 2	pin #3
pin 3	out	B2	green #1	data 3	pin #4
pin 4	out	B3	violet #1	data 4	pin #5
pin 5	out	B4	white #1	data 5	pin #6
pin 6	out	B5	brown #2	data 6	pin #7
pin 7	out	B6	orange #2	data 7	pin #8
pin 9	gnd		violet #2	data 8	pin #9
pin 14	in	B0	red#1	BUSY	pin #11
pin 22	gnd		gray #2	GND	pin #26

CAUTION: the pinout of the SBC-880 parallel port connector is called out to correspond to like pin numbers on a DB-25 connector (with conductor #26 cut back). Use of a MALE DB-25 connector is strongly recommended as well as proper labelling to prevent damage to your CPU card or printer, from swapping cables with the RS-232C lines.

SOFTWARE DRIVER

The following listing is exactly the same as used in the COMPUTIME CTVIII.2 monitor/bios. If you are using another monitor that does not have a printer driver, it can be added into ROM or your CP/M BIOS.

```
INITIALIZATION          MVI    A,80H
                        OUT    INOUT    ;INOUT IS THE PORT ADDRESS
                                        ; THIS SETS BIT 7 HIGH IN
                                        ; THE OUTPUT PORT
DRIVER          CHK     IN     INOUT
                        ANI    01       ;IS THE PRINTER BUSY?
                        JRNZ   CHK      ;YES--WAIT UNTIL IT IS NOT
                                        ;NO--OUTPUT THE CHARACTER
                        MOV    A,C
                        ORI    80H
                        OUT    INOUT    ;OUTPUT CHARACTER WITH BIT
                                        ;7 HIGH
                        ANI    7FH      ;BIT 7 GOING LOW
                        OUT    INOUT    ;GENERATE LOW GOING STROBE
```

```
                        ORI     80H     ;BIT 7 GOING HIGH
                        OUT     INOUT   ;GENERATE HIGH GOING STROBE
                        RET
```

PRINTER STATUS ROUTINE

```
        PTRSTS          IN      INOUT
                        ANI     01      ;IS THE PRINTER BUSY?
                        MVI     A,0FFH
                        RZ
                        CMA
                        RET
```

Additional status information could be added to this last routine by adding more input lines from the printer such as "out of paper", "off-line", etc. could be monitored.

INTERFACE LIMITATIONS

This particular printer interface will only handle 7 bit parallel data printers. Some printers use the 8th bit to control graphic functions and other features. Check with your dealer for appropriate printers for your consideration. The software status routine looks only at the "BUSY" line of the printer and so may cause the computer to "hang" if it is not powered on and placed on line when you try to print something. A more involved status determining subroutine can be added to make the system more user friendly.

APPENDIX E

INTERFACING NON-IEEE DYNAMIC MEMORY CARDS TO THE SBC-880

BACKROUND

Before the emergence of the IEEE-696 standard for signal specifications there evolved an entire generation of dynamic ram cards that used two additional signals from the cpu card, this note describes modifications to the SBC880 board that will allow proper interfacing.

PC CARD MODIFICATIONS

CIRCUIT BOARD CUTS - none are required for this application.
CIRCUIT BOARD JUMPS - a total of three additional jumpers are required.

JUMPER #1--U19 pin 28 to U32 pin 6
JUMPER #2--U32 pin 7 to edge connector 66 (*rfsh)
JUMPER #3--U10 pin 3 to edge connector 65 (*mreq)

UPDATE THE SCHEMATIC - Always document changes to your system, in case of problems this information can save you untold hours of troubleshooting.

MODIFICATION DESCRIPTION - Early generation dynamic ram memory cards depended on two Z-80 CPU chip derived signals that made their proper operation possible, these two signals presently deleted from the IEEE spec can be added to take advantage of these older memory boards. There are only two limitations on using the CPU generated refresh signals. ONE--be careful not to use DMA type disk controllers and TWO--do not attempt to use non-Z-80 slave processors in your computer; Either violation of the rule will result in unpredictable operation of the system.

THIS PAGE LEFT BLANK

SYSTEM MONITOR

The COMPUTIME MONITOR is designed to be used for trouble shooting your system. The MONITOR is designed to run on the on-board EPROM of the COMPUTIME SBC 880 processor board. The switches should be set as follows to allow operation of the MONITOR. Switches 5 and 6 of SW3 should be ON and switches 1, 2, 3, and 4 of SW3 should be OFF. Switches 2, 4, and 5 of SW1 should be ON and switches 1, 3, and 6 of SW1 should be off. If your system has no RAM memory at address zero then switches 1, 2, 3, 4, 5, and 6 of SW2 should be ON to select the on-board RAM to be at address zero. A 9600 baud RS232 terminal is required by the MONITOR and it should be connected to connector J1.

The CPU board and terminal are all that is required for operation of the MONITOR and they may be used to diagnose problems on any other boards in your system. The COMPUTIME MONITOR also provides features for debug of programs and may be used to get I/O driver and boot routines operating in your system.

With the power-on jump option enabled on the CPU board the MONITOR will be executed when your system is powered up. The MONITOR will be re-initialized each time your system is reset. The MONITOR prompts for input from the user by displaying "/" on the terminal. Input errors are indicated by displaying "*" on the terminal. All MONITOR commands are entered as a single letter followed by parameters as required by the various commands.

Command entry is terminated by hitting the return key. When the designated command has completed, the monitor displays the input prompt "/". parameters are entered as hexadecimal values using the ASCII characters 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F. Parameters are separated by commas or spaces. Following is a list of commands:

D	DISPLAY CONTENTS OF MEMORY IN HEX & ASCII SAMPLE D200,300
F	FILL MEMORY WITH A CONSTANT VALUE SAMPLE F100,400,FF
G	GOTO USER PROGRAM SAMPLE G100
H	GIVES THE SUM AND DIFFERENCE OF 2 HEX NUMBERS SAMPLE HF400,FDE2
I	INPUT FROM AN I/O PORT SAMPLE I2C
J	DESTRUCTIVE MEMORY TEST SAMPLE J400,C000
M	MOVE BLOCK OF MEMORY SAMPLE MF000,F800,1000

O OUTPUT TO I/O PORT
SAMPLE O2C,55

S SUBSTITUTE OR EXAMINE MEMORY
SAMPLE S100

V VERIFY MEMORY AGAINST MEMORY
SAMPLE VF000,F400,1000

DETAILED COMMAND DESCRIPTION

D This command accepts a start and end address as parameters. The contents of memory from the start address up to the end address will be displayed in hexadecimal & ASCII on the terminal device. The command D100,200 would cause the contents of memory from 100 through 200 to be displayed on the terminal. The start and end address will be adjusted to the nearest 16 byte boundaries that include the requested memory range. For example, the command D10F,1F1 would display memory from 100 to 200.

F This command accepts a start address, end address, and a data value as parameters. The data value will be written to memory at the start address through the end address. The command F100,200,55 would fill the memory starting at address 100 through address 200 with the data pattern 55.

G This command accepts a branch address as an input parameter. The command G100 would jump the processor to address 100 and begin execution of the user program at that point.

H This command accepts two hexadecimal numbers (maximum value of FFFF) and displays the sum and the difference of these numbers on the terminal device. The command H200,100 will display 100 300 on the terminal.

I This command accepts an I/O port address and displays the result of an input from the designated port. The command I2C would display the result of an input instruction at I/O port address 2C.

J This command accepts a start address and an end address as input parameters and tests the memory starting at the start address through the end address. The command J0,BFFF will test memory at address 0 through BFFF. Testing of memory will continue until the command is aborted. Pressing any key on the terminal will abort the command. The memory test performed is destructive, however the stack and MONITOR storage memory may be overwritten without affecting operation of the test portion of the routine. The stack is only used by the routine while it is displaying a detected error. The address of the error is displayed followed by the expected data contents and the actual data contents. Testing will resume after an error is reported. A tally is incremented after each pass through the memory. The data pattern is formed by an exclusive OR of the tally and the upper and lower bytes of the current memory address. A complete write of the data pattern is performed followed by a READ/VERIFY test for each pass through the memory.

M This command accepts three parameters. The start address of the source data to be moved is entered first followed by the ending address of the source data to be moved. The third parameter is the start address of the destination for the data block being moved. The MONITOR will begin moving data from the starting source address to the starting destination address and the addresses will be incremented as each byte of data is moved from the source to the destination. This process will continue until the end address of the source data is reached. The command M100,200,400 will move the block of data at address 100 through 200 to address 400 through 500.

O This command accepts an I/O port address and a data byte. The command will cause the data value to be output to the designated I/O address. The command O2C,55 would output the value 55 to I/O port address 2C. The maximum data value is FF.

S This command accepts a memory address as a parameter. The monitor will display the data contained at that address if the space bar or comma is pressed on the terminal. Successive memory locations may be displayed by continually pressing the space bar or comma. If data is entered on the keyboard after displaying a memory location, that data will be written to the location just displayed when the space bar or comma is pressed. The contents of the next location is displayed after the old location is rewritten. The command is terminated by hitting the return key. The command S100 would display the contents of the memory at address 100 when the space bar or comma is pressed. The contents of this location could be modified by typing data at this time followed by pressing the space bar or comma. Pressing the return key would terminate the command.

V This command accepts three parameters. The start address of the first memory block and the end address of the first memory block are followed by the start address of a second memory block. The two memory blocks are compared to one another and any differences are displayed on the terminal. No display would indicate that the two blocks of data contained identical data. The command V100,200,300 would compare the block of data from 100 to 200 to the block of data at 300 to 400. Any differences would be displayed on the terminal.

OPERATING HINTS

All MONITOR command entries are terminated by entering a carriage return. If an entry error occurs you will be notified by a "*" being displayed. The "/" prompt from the MONITOR must be present before the MONITOR will accept commands to be executed. If you make an error entering a command just re-enter it correctly after the "*" is displayed. All the command parameters are entered as hexadecimal values with a maximum value of FF for an 8 bit value and FFFF for a 16 bit value. If more hexadecimal numbers than needed are entered the MONITOR will accept the last ones entered and ignore all the others. For example if 1234567 was entered as an address the MONITOR would ignore the 123 and the address accepted by the MONITOR would be 4567. Note that accessing I/O ports 28 through 2F using the "O" command could affect operation of the MONITOR and these I/O port addresses should be avoided. The MONITOR executes at addresses F800 through FFFF. Any user memory at these addresses will be ignored. It is not necessary to disable these external memory devices while the MONITOR is in use. However, the MONITOR will be unable to access external memory from address F800 through FFFF. The on-board RAM functions in a similar fashion to the MONITOR EPROM when accessed. Any external memory at addresses 0000 through 03FF will be ignored by the CPU board. This feature is useful when debugging an inoperative memory board. The on-board RAM is selected to be at address 0 through 3FF and is used to hold the MONITOR stack. Test programs may be typed into RAM at address 100 through 3FF to help diagnose the failing memory board in addition to using the "J" command.

```
                        ;       ****************************************
                        ;       * SYSTEM MONITOR, VERSION I.1 REV A    *
                        ;       * REQUIRES SBC880 CPU BOARD            *
                        ;       * SW1-1=OFF,2=ON,3=OFF,4=ON,5=ON       *
                        ;       * SW2-1 THRU 6 = ON                    *
                        ;       * SW3-1 THRU 5 = OFF, 6 = ON           *
                        ;       * WRITTEN BY R. D. CATILLER            *
                        ;       * COPYRIGHT 1981 (C) COMPUTIME         *
                        ;       ****************************************
                        ;
                        ;
                        .PABS
                        ;
                        ;       MEMORY USED BY MONITOR
                        ;
F800                    BASE    =       0F800H  ;MONITOR BASE ADDRESS.
00FF                    STACK   =       00FFH   ;MONITOR STACK
0010                    REGSTR  =       0010H   ;CPU REGISTER STORAGE
0020                    OLDOP   =       REGSTR+16
0021                    BRKSTR  =       REGSTR+17
0023                    HLSTR   =       REGSTR+19
0038                    RST7    = 38H           ;RST 7 (LOCATION FOR TRAP)
                        ;
                        ;       CONSTANTS FOR MONITOR
                        ;
000D                    CR      = 0DH           ;ASCII CARRIAGE RET
000A                    LF      = 0AH           ;ASCII LINE FEED
                        ;
                        ;       I/O PORTS ON CPU BOARD
                        ;
0028                    T0      =       28H
0029                    T1      =       29H
002A                    T2      =       2AH
002B                    TCTL    =       2BH
002C                    INOUT   =       2CH
002E                    CONDTA  =       2EH
002F                    CONCTL  =       2FH
002F                    CONSTS  =       CONCTL
                        ;
                        ;       PROGRAM CODE BEGINS:
                        ;
F800                    .LOC    BASE
                        ;
                        ;LET US BEGIN
                        ;
F800    C3 F803                 JMP     BEGIN           ;RESET JUMP LATCH
F803    3E00            BEGIN:  MVI     A,0             ;CLEAR REGISTER STORE
F805    21 0010                 LXI     H,REGSTR
F808    77                      MOV     M,A
F809    11 0011                 LXI     D,REGSTR+1
F80C    01 0012                 LXI     B,18
F80F    EDB0                    LDIR
F811    3EC3                    MVI     A,0C3H          ;SET BREAKPOINT TRAP
F813    32 0038                 STA     RST7
F816    21 FB08                 LXI     H,BREAK
```

```
F819    22 0039                 SHLD    RST7+1
 81C    22 001E                 SHLD    REGSTR+14        ;SET INITIAL PC
F81F    31 00FF                 LXI     SP,STACK         ;SET UP STACK
F822    ED73 0018               SSPD    REGSTR+8
F826    3E80                    MVI     A,80H            ;SET UP PRINTER
F828    D32C                    OUT     INOUT
F82A    3EFA                    MVI     A,0FAH   5A      ;SET USART MODE
F82C    D32F                    OUT     CONCTL
F82E    3E36                    MVI     A,36H            ;SET TIMER MODE
F830    D32B                    OUT     TCTL
F832    3E0D                    MVI     A,13             ;SET BAUD RATE TO 9600
F834    D328                    OUT     TO
F836    AF                      XRA     A
F837    D328                    OUT     TO
F839    3E37                    MVI     A,37H    37      ;ENABLE XMT & REC
F83B    D32F                    OUT     CONCTL
F83D    061A            HELLO:  MVI     B,MSGL           ;SAY HELLO
F83F    CD F892                 CALL    MESG
F842    31 00FF         START:  LXI     SP,STACK         ;RESTORE STACK
F845    CD F8BA                 CALL    CRLF
F848    0E2F                    MVI     C,'/'
F84A    CD F8A5                 CALL    CO
F84D    CD F8DF                 CALL    MAININ           ;GET INPUT
F850    FE44                    CPI     'D'
F852    CA F97B                 JZ      DISP             ;D = DISPLAY MEMORY
F855    FE46                    CPI     'F'
 857    CA F9CF                 JZ      FILL             ;F = FILL MEMORY
F85A    FE47                    CPI     'G'
F85C    CA F9E0                 JZ      GOTO             ;G = GOTO USER PROGRAM
F85F    FE48                    CPI     'H'
F861    CA FA37                 JZ      HEXN             ;H = HEX MATH
F864    FE49                    CPI     'I'
F866    CA FA9D                 JZ      INPUT            ;I = INPUT FROM PORT
F869    FE4A                    CPI     'J'
F86B    CA FA46                 JZ      TEST             ;J = RAM TEST
F86E    FE4D                    CPI     'M'
F870    CA FA95                 JZ      MOVE             ;M = MEMORY MOVE
F873    FE4F                    CPI     'O'
F875    CA FAAE                 JZ      OUTPUT           ;O = OUTPUT TO PORT
F878    FE53                    CPI     'S'
F87A    CA FAB7                 JZ      SUBS             ;S = SUBSTITUTE/EXAMINE
F87D    FE56                    CPI     'V'
F87F    CA FAF6                 JZ      VERIFY           ;V = VERIFY MEMORY
F882    FE58                    CPI     'X'
F884    CA FB33                 JZ      XAM              ;X = EXAMINE REGISTERS
F887    31 00FF         ERROR:  LXI     SP,STACK         ;RESTORE STACK
F88A    0E2A                    MVI     C,'*'            ;DISPLAY ASTERISK
F88C    CD F8A5                 CALL    CO
F88F    C3 F842                 JMP     START            ;GO GET INPUT
                        ;
                        ;MESSAGE OUTPUT ROUTINE.
                        ;
F892    21 FB6D         MESG:   LXI     H,MSG
F895    4E              MESG1:  MOV     C,M              ;GET A CHARACTER
F896    23                      INX     H                ;MOVE POINTER
```

```
F897    CD F8A5            CALL    CO              ;OUTPUT IT
 89A    10F9               DJNZ    MESG1           ;KEEP GOING TILL B=0
F89C    C9                 RET
                   ;
                   ;CRLF BEFORE HLBLK ROUTINE
                   ;
F89D    CD F8BA    CRLFHL: CALL    CRLF
                   ;
                   ;PRINT THE CURRENT VALUE OF H&L,
                   ;AND A SPACE.
                   ;
F8A0    CD F8CB    HLBLK:  CALL    DISPHL
                   ;
                   ;PRINT A SPACE ON THE CONSOLE
                   ;
F8A3    0E20       SPACE:  MVI     C,' '
                   ;
                   ;THIS IS THE MAIN CONSOLE
                   ;OUTPUT ROUTINE.
                   ;
F8A5    DB2F       CO:     IN      CONCTL
F8A7    E601               ANI     01H
F8A9    CA F8A5            JZ      CO
F8AC    79                 MOV     A,C
F8AD    D32E               OUT     CONDTA
F8AF    C9                 RET
                   ;
                   ; CONVERT HEX TO ASCII
                   ;
F8B0    E60F       HTA:    ANI     0FH     ;LOW NIBBLE ONLY
F8B2    C690               ADI     90H
F8B4    27                 DAA
F8B5    CE40               ACI     40H
F8B7    27                 DAA
F8B8    4F                 MOV     C.A
F8B9    C9                 RET
                   ;
                   ; CONSOLE CARRIAGE RETURN &
                   ; LINE FEED ROUTINE.
                   ;
F8BA    E5         CRLF:   PUSH    H       ;SAVE HL
F8BB    0602               MVI     B,2     ;CRLF LENGTH
F8BD    CD F892            CALL    MESG    ;SEND CRLF
F8C0    E1                 POP     H
F8C1    C9                 RET
                   ;
                   ;CONSOLE STATUS TEST ROUTINE.
                   ;
F8C2    DB2F       CSTS:   IN      CONCTL
F8C4    E602               ANI     02H
F8C6    3EFF               MVI     A,0FFH
F8C8    C0                 RNZ
F8C9    2F                 CMA
F8CA    C9                 RET
                   ;
```

```
                              ;PRINT H&L ON CONSOLE
                              ;
 F8CB    7C                   DISPHL:  MOV     A,H
 F8CC    CD F8D0                       CALL    DISPB
 F8CF    7D                            MOV     A,L
 F8D0    F5                   DISPB:   PUSH    PSW
 F8D1    0F                            RRC
 F8D2    0F                            RRC
 F8D3    0F                            RRC
 F8D4    0F                            RRC
 F8D5    CD F8D9                       CALL    HTA2
 F8D8    F1                            POP     PSW
 F8D9    CD F8B0              HTA2:    CALL    HTA
 F8DC    C3 F8A5                       JMP     CO
                              ;
                              ;MAIN KEYBOARD ROUTINE
                              ;
 F8DF    CD F95A              MAININ:  CALL    CI              ;GET INPUT
 F8E2    4F                            MOV     C,A             ;ECHO IT
 F8E3    C3 F8A5                       JMP     CO
                              ;
                              ;MAIN PARAMETER GETTING ROUTINE
                              ;
 F8E6    21 0000              GPARAM:  LXI     H,0             ;CLEAR HL
 F8E9    CD F8DF              GPNEXT:  CALL    MAININ          ;GET INPUT
 F8EC    47                   GP1:     MOV     B,A             ;SAVE IT
 F8ED    FE20                          CPI     ' '             ;TEST FOR SPACE
 F8EF    C8                            RZ                      ;RETURN IF SPACE
 F8F0    FE2C                          CPI     ','             ;TEST FOR COMMA
 F8F2    C8                            RZ                      ;RETURN IF COMMA
 F8F3    FE0D                          CPI     CR              ;TEST FOR CR
 F8F5    C8                            RZ                      ;RETURN IF CR
 F8F6    D630                          SUI     '0'             ;TEST < 0
 F8F8    DA F887                       JC      ERROR           ;INPUT ERROR
 F8FB    FE17                          CPI     'G'-'0'         ;TEST IF > F
 F8FD    D2 F887                       JNC     ERROR           ;INPUT ERROR
 F900    FE0A                          CPI     10              ;TEST FOR NUMBER
 F902    DA F90C                       JC      DONE            ;GO SAVE NUMBER
 F905    D607                          SUI     'A'-'9'-1       ;ADJUST LETTER
 F907    FE0A                          CPI     0AH             ;TEST FOR . THRU @
 F909    DA F887                       JC      ERROR           ;INPUT ERROR
 F90C    29                   DONE:    DAD     H               ;SHIFT HL 1 DIGIT
 F90D    29                            DAD     H
 F90E    29                            DAD     H
 F90F    29                            DAD     H
 F910    B5                            ORA     L               ;OR L WITH DIGIT
 F911    6F                            MOV     L,A
 F912    C3 F8E9                       JMP     GPNEXT          ;GET MORE INPUT
                              ;
                              ;GETS START & END ADDRESS AND DETERMINES LENGTH
                              ;
 F915    CD F8E6              RANGE:   CALL    GPARAM          ;GET START ADDRESS
 F918    FE0D                          CPI     CR              ;TEST FOR CR
 F91A    CA F887                       JZ      ERROR           ;INPUT ERROR
 F91D    54                            MOV     D,H             ;PUT HL IN DE
```

```
F91E    5D                      MOV     E,L
F91F    CD F8E6                 CALL    GPARAM          ;GET END ADDRESS
F922    E5                      PUSH    H               ;SAVE IT
F923    B7                      ORA     A               ;END - START
F924    ED52                    DSBC    D
F926    44                      MOV     B,H             ;PUT LENGTH IN BC
F927    4D                      MOV     C,L
F928    62                      MOV     H,D             ;PUT START IN HL
F929    6B                      MOV     L,E
F92A    D1                      POP     D               ;PUT END IN DE
F92B    C9                      RET
                        ;
F92C    CD F915         RANGE2: CALL    RANGE           ;GET 2 PARAMETERS
F92F    FE0D                    CPI     CR              ;TEST FOR CR
F931    C2 F887                 JNZ     ERROR           ;INPUT ERROR
F934    C9                      RET
                        ;
F935    CD F915         RANGE3: CALL    RANGE           ;GET 2 PARAMETERS
F938    FE0D                    CPI     CR              ;TEST FOR CR
F93A    CA F887                 JZ      ERROR           ;INPUT ERROR
F93D    22 0023                 SHLD    HLSTR           ;SAVE START
F940    C5              PARAM1: PUSH    B               ;SAVE B
F941    CD F8E6                 CALL    GPARAM          ;GET 3RD PARAMETER
F944    C1                      POP     B               ;RESTORE BC
F945    FE0D                    CPI     CR              ;TEST FOR CR
F947    C2 F887                 JNZ     ERROR           ;INPUT ERROR
F94A    C9                      RET
                        ;
F94B    E5              ENDTST: PUSH    H               ;SAVE HL
F94C    B7                      ORA     A               ;HL - DE
F94D    ED52                    DSBC    D
F94F    E1                      POP     H               ;RESTORE HL
F950    C9                      RET                     ;RETURN FLAGS
                        ;
F951    CD F935         SDL:    CALL    RANGE3          ;GET 3 PARAMETERS
F954    54                      MOV     D,H             ;DEST TO DE
F955    5D                      MOV     E,L
F956    2A 0023                 LHLD    HLSTR           ;SOURCE TO HL
F959    C9                      RET                     ;BC = LENGTH
                        ;
                        ;MAIN CONSOLE INPUT ROUTINE
                        ;
F95A    DB2F            CI:     IN      CONCTL
F95C    E602                    ANI     02H
F95E    CA F95A                 JZ      CI
F961    DB2E                    IN      CONDTA
F963    E67F                    ANI     7FH
F965    C9                      RET
                        ;
                        ;PRINTER OUTPUT ROUTINE
                        ;
F966    DB2C            PRINT:  IN      INOUT
F968    E601                    ANI     1
F96A    C2 F966                 JNZ     PRINT
F96D    79                      MOV     A,C
```

```
F96E    F680                ORI     80H
F970    D32C                OUT     INOUT
F972    E67F                ANI     7FH
F974    D32C                OUT     INOUT
F976    F680                ORI     80H
F978    D32C                OUT     INOUT
F97A    C9                  RET
                    ;
                    ;DISPLAYS CONTENTS OF MEMORY IN HEX & ASCII
                    ;
F97B    CD F92C     DISP:   CALL    RANGE2          ;GET PARAMETERS
F97E    7D                  MOV     A,L             ;ADJUST START ADDRESS
F97F    E6F0                ANI     0F0H
F981    6F                  MOV     L,A
F982    7B          DISP1:  MOV     A,E             ;ADJUST END ADDRESS
F983    E60F                ANI     0FH
F985    CA F98C             JZ      DISP2
F988    13                  INX     D
F989    C3 F982             JMP     DISP1
F98C    CD F89D     DISP2:  CALL    CRLFHL          ;DISPLAY CRLF & ADR
F98F    CD F8A3     DISP3:  CALL    SPACE           ;DISPLAY SPACE
F992    7E                  MOV     A,M             ;GET DATA
F993    CD F8D0             CALL    DISPB           ;DISPLAY IT
F996    7D                  MOV     A,L             ;TEST FOR END OF LINE
F997    E60F                ANI     0FH
F999    FE0F                CPI     0FH
F99B    CA F9A2             JZ      DISP4           ;DISPLAY ASCII
F99E    23                  INX     H               ;NEXT
F99F    C3 F98F             JMP     DISP3
F9A2    CD F8A3     DISP4:  CALL    SPACE           ;DISPLAY A SPACE
F9A5    CD F8A3             CALL    SPACE           ;DISPLAY A SPACE
F9A8    7D                  MOV     A,L             ;BACK UP ADR
F9A9    E6F0                ANI     0F0H
F9AB    6F                  MOV     L,A
F9AC    7E          NEXTA:  MOV     A,M             ;GET DATA
F9AD    E67F                ANI     7FH             ;KILL PARITY
F9AF    FE20                CPI     ' '             ;TEST IF < SPACE
F9B1    D2 F9B6             JNC     DISP6           ;TEST FURTHER
F9B4    3E2E        DISP5:  MVI     A,'.'           ;REPLACE WITH PERIOD
F9B6    FE7C        DISP6:  CPI     7CH             ;> LOWER CASE Z
F9B8    D2 F9B4             JNC     DISP5           ;REPLACE IT
F9BB    4F                  MOV     C,A             ;DISPLAY IT
F9BC    CD F8A5             CALL    CO
F9BF    23                  INX     H               ;STEP TO NEXT
F9C0    7D                  MOV     A,L             ;TEST FOR END OF LINE
F9C1    E60F                ANI     0FH
F9C3    C2 F9AC             JNZ     NEXTA           ;DO NEXT
F9C6    CD F94B             CALL    ENDTST          ;TEST FOR END
F9C9    C2 F98C             JNZ     DISP2           ;NEXT LINE
F9CC    C3 F642             JMP     START
                    ;
                    ;FILL MEMORY WITH A CONSTANT
                    ;
F9CF    CD F935     FILL:   CALL    RANGE3          ;GET 2 PARAMETERS
F9D2    7D                  MOV     A,L             ;PUT DATA IN A
```

```
F9D3    2A 0023              LHLD    HLSTR           ;START ADR TO HL
 9D6    77                   MOV     M,A             ;WRITE DATA AT START
F9D7    54                   MOV     D,H             ;COPY HL TO DE
F9D8    5D                   MOV     E,L
F9D9    13                   INX     D               ;DEST = SOURCE + 1
F9DA    0B                   DCX     B               ;ADJUST LENGTH
F9DB    EDB0                 LDIR                    ;WRITE DATA
F9DD    C3 F842              JMP START
                     ;
                     ;GOTO USER PROGRAM WITH OPTIONAL BREAKPOINT
                     ;
F9E0    CD F95A      GOTO:   CALL    CI              ;GET INPUT
F9E3    FE0D                 CPI     CR              ;TEST FOR CR
F9E5    CA FA16              JZ      GOTO1           ;USE OLD PC VALUE
F9E8    4F                   MOV     C,A             ;ECHO INPUT
F9E9    CD F8A5              CALL    CO
F9EC    FE20                 CPI     ' '             ;TEST FOR SPACE
F9EE    CA FA04              JZ      GOTO2           ;OLD PC, NEW BRKPOINT
F9F1    FE2C                 CPI     ','             ;TEST FOR COMMA
F9F3    CA FA04              JZ      GOTO2           ;OLD PC, NEW BRKPOINT
F9F6    21 0000              LXI     H,0             ;CLEAR HL
F9F9    CD F8EC              CALL    GP1             ;GET PARAMETER
F9FC    22 001E              SHLD    REGSTR+14       ;SET NEW PC
F9FF    FE0D                 CPI     CR              ;TEST FOR CR
FA01    CA FA16              JZ      GOTO1           ;JUMP TO NEW PC
FA04    CD F8E6      GOTO2:  CALL    GPARAM          ;GET BREAKPOINT ADR
FA07    FE0D                 CPI     CR              ;TEST FOR CR
FA09    C2 F887              JNZ     ERROR           ;INPUT ERROR
FA0C    7E                   MOV     A,M             ;GET OLD OP
FA0D    32 0020              STA     OLDOP           ;SAVE IT
FA10    22 0021              SHLD    BRKSTR          ;SAVE BREAKPOINT ADR
FA13    3EFF                 MVI     A,0FFH          ;STORE A BREAKPOINT
FA15    77                   MOV     M.A
FA16    2A 0010      GOTO1:  LHLD    REGSTR          ;GET PSW
FA19    E5                   PUSH    H
FA1A    F1                   POP     PSW
FA1B    ED4B 0012            LBCD    REGSTR+2        ;GET BC
FA1F    ED5B 0014            LDED    REGSTR+4        ;GET DE
FA23    ED7B 0018            LSPD    REGSTR+8        ;GET SP
FA27    2A 001E              LHLD    REGSTR+14       ;GET PC
FA2A    E5                   PUSH    H
FA2B    2A 0016              LHLD    REGSTR+6        ;GET HL
FA2E    DD2A 001A            LIXD    REGSTR+10       ;GET IX
FA32    FD2A 001C            LIYD    REGSTR+12       ;GET IY
FA36    C9                   RET                     ;GOTO USER PROGRAM
                     ;
                     ;HEXADECIMAL MATH ROUTINE
                     ;
FA37    CD F92C      HEXN:   CALL    RANGE2          ;GET PARAMETERS
FA3A    19                   DAD     D               ;ADD PARAMETERS
FA3B    C5                   PUSH    B               ;SAVE DIFFERENCE
FA3C    CD F89D              CALL    CRLFHL          ;DISPLAY SUM
FA3F    E1                   POP     H               ;GET DIFFERENCE
FA40    CD F89D              CALL    CRLFHL          ;DISPLAY IT
FA43    C3 F842              JMP     START
```

```
                                ;
                                ;EXTEDED MEMORY TEST
                                ;
FA45    CD F92C         TEST:   CALL    RANGE2          ;GET 2 PARAMETERS
FA48    44                      MOV     B,H             ;SAVE START IN BC
FA4A    4D                      MOV     C,L
FA4B    3E00                    MVI     A,0             ;CLEAR I
FA4D    ED47            LOOP:   STAI
FA4F    ED57            FILLIT: LDAI                    ;BUILD DATA
FA51    AD                      XRA     L
FA52    AC                      XRA     H
FA53    77                      MOV     M,A             ;WRITE DATA
FA54    23                      INX     H               ;NEXT DATA
FA55    7C                      MOV     A,H             ;TEST FOR END
FA56    BA                      CMP     D
FA57    C2 FA4F                 JNZ     FILLIT          ;CONTINUE WRITING
FA5A    7D                      MOV     A,L             ;TEST FOR END
FA5B    BB                      CMP     E
FA5C    C2 FA4F                 JNZ     FILLIT          ;CONTINUE WRITING
FA5F    60                      MOV     H,B             ;RESTORE START
FA60    69                      MOV     L,C
FA61    ED57            TEST1:  LDAI                    ;BUILD DATA
FA63    AD                      XRA     L
FA64    AC                      XRA     H
FA65    BE                      CMP     M               ;COMPARE IT
FA66    C4 FA83                 CNZ     MERR            ;DISPLAY ERRORS
FA69    23                      INX     H
FA6A    7C                      MOV     A,H             ;TEST FOR END
FA6B    BA                      CMP     D
FA6C    C2 FA61                 JNZ     TEST1           ;CONTINUE TEST
FA6F    7D                      MOV     A,L             ;TEST FOR END
FA70    BB                      CMP     E
FA71    C2 FA61                 JNZ     TEST1           ;CONTINUE TEST
FA74    60                      MOV     H,B             ;RESTORE START
FA75    69                      MOV     L.C
FA76    DB2F                    IN      CONCTL          ;TEST KEYBOARD
FA78    E602                    ANI     02H
FA7A    C2 F842                 JNZ     START           ;ABORT IF KEY PRESSED
FA7D    ED57                    LDAI                    ;INCREMENT TALLY
FA7F    3C                      INR     A
FA80    C3 FA4D                 JMP     LOOP            ;ANOTHER PASS
                                ;
FA83    C5              MERR:   PUSH    B               ;SAVE BC
FA84    F5                      PUSH    PSW             ;SAVE DATA
FA85    CD F89D                 CALL    CRLFHL          ;DISPLAY ADDRESS
FA88    F1                      POP     PSW             ;DISPLAY DATA
FA89    CD F8D0                 CALL    DISPB
FA8C    CD F8A3                 CALL    SPACE           ;DISPLAY A SPACE
FA8F    7E                      MOV     A,M             ;GET MEM DATA
FA90    CD F8D0                 CALL    DISPB           ;DISPLAY IT
FA93    C1                      POP     B               ;RESTORE BC
FA94    C9                      RET                     ;CONTINUE TESTING
                                ;
                                ;MOVE BLOCK OF MEMORY
                                ;
```

```
FA95    CD F951     MOVE:   CALL    SDL             ;SRC, DEST, LNGTH
FA98    EDB0                LDIR                    ;DO MOVE
FA9A    C3 F842             JMP     START
                    ;
                    ;INPUT DATA FROM AN I/O PORT
                    ;
FA9D    CD F940     INPUT:  CALL    PARAM1          ;GET PARAMETER
FAA0    4D                  MOV     C,L             ;PUT IO ADR IN C
FAA1    ED78                INP     A               ;INPUT DATA TO A
FAA3    F5                  PUSH    PSW             ;SAVE IT
FAA4    CD F8BA             CALL    CRLF            ;DISPLAY CRLF
FAA7    F1                  POP     PSW             ;GET DATA
FAA8    CD F8D0             CALL    DISPB           ;DISPLAY IT
FAAB    C3 F842             JMP     START
                    ;
                    ;OUTPUT DATA TO AN I/O PORT
                    ;
FAAE    CD F92C     OUTPUT: CALL    RANGE2          ;GET 2 PARAMETERS
FAB1    4D                  MOV     C,L             ;OUTPUT DATA
FAB2    ED59                OUTP    E
FAB4    C3 F842             JMP     START
                    ;
                    ;SUBSTITUTE AND EXAMINE MEMORY
                    ;
FAB7    CD F8E6     SUBS:   CALL    GPARAM          ;GET PARAMETER
FABA    CD F89D     SUBS1:  CALL    CRLFHL          ;DISPLAY ADDRESS
FABD    7E                  MOV     A,M             ;GET DATA
FABE    CD F8D0             CALL    DISPB           ;DISPLAY IT
FAC1    CD F8A3             CALL    SPACE           ;DISPLAY A SPACE
FAC4    CD F95A             CALL    CI              ;GET INPUT
FAC7    FE0D                CPI     CR              ;TEST FOR CR
FAC9    CA F842             JZ      START           ;DONE
FACC    FE20                CPI     ' '             ;TEST FOR SPACE
FACE    C2 FAD5             JNZ     SUBS2           ;TEST FURTHER
FAD1    23          SUBS4:  INX     H               ;ADDRESS + 1
FAD2    C3 FABA             JMP     SUBS1           ;CONTINUE DISPLAY
FAD5    FE7F        SUBS2:  CPI     7FH             ;TEST FOR RUBOUT
FAD7    C2 FADE             JNZ     SUBS3           ;LOOK FOR PARAMETER
FADA    2B                  DCX     H               ;ADDRESS - 1
FADB    C3 FABA             JMP     SUBS1           ;CONTINUE DISPLAY
FADE    4F          SUBS3:  MOV     C,A             ;ECHO CHARACTER
FADF    CD F8A5             CALL    CO
FAE2    E5                  PUSH    H               ;SAVE ADR
FAE3    21 0000             LXI     H,0             ;CLEAR HL
FAE6    CD F8EC             CALL    GP1             ;GET PARAMETER
FAE9    47                  MOV     B,A             ;SAVE DELIMITER
FAEA    7D                  MOV     A,L             ;PUT DATA IN A
FAEB    E1                  POP     H               ;GET ADR
FAEC    77                  MOV     M,A             ;STORE DATA
FAED    78                  MOV     A,B             ;TEST FOR CR
FAEE    FE0D                CPI     CR
FAF0    CA F842             JZ      START           ;DONE
FAF3    C3 FAD1             JMP     SUBS4           ;CONTINUE
                    ;
                    ;VERIFY BLOCK OF MEMORY AGAINST MEMORY
```

```
                                ;
 AF6    CD F951    VERIFY: CALL    SDL           ;SRC, DEST, LNGTH
 FAF9   1A         VER1:   LDAX    D             ;GET DEST DATA
 FAFA   EDA1               CCI                   ;COMPARE TO SOURCE
 FAFC   E2 F842            JPO     START         ;DONE
 FAFF   2B                 DCX     H             ;ADJUST ADDRESS
 FB00   C4 FA83            CNZ     MERR          ;DISPLAY ERRORS
 FB03   23                 INX     H             ;RESTORE ADDRESS
 FB04   13                 INX     D             ;DEST + 1
 FB05   C3 FAF9            JMP     VER1          ;CONTINUE
                                ;
                                ;RETURN FROM BREAKPOINT HERE
                                ;
 FB08   F5         BREAK:  PUSH    PSW           ;SAVE PSW
 FB09   ED43 0012          SBCD    REGSTR+2      ;STORE BC
 FB0D   ED53 0014          SDED    REGSTR+4      ;STORE DE
 FB11   22 0016            SHLD    REGSTR+6      ;STORE HL
 FB14   E1                 POP     H             ;GET PSW
 FB15   22 0010            SHLD    REGSTR        ;STORE PSW
 FB18   E1                 POP     H             ;GET PC
 FB19   2B                 DCX     H             ;PC - 1
 FB1A   22 001E            SHLD    REGSTR+14     ;STORE PC
 FB1D   2A 0021            LHLD    BRKSTR        ;GET BREAKPOINT ADR
 FB20   3A 0020            LDA     OLDOP         ;RESTORE OLD OP
 FB23   77                 MOV     M,A
 FB24   ED73 0018          SSPD    REGSTR+8      ;STORE SP
 FB28   DD22 001A          SIXD    REGSTR+10     ;STORE IX
 FB2C   FD22 001C          SIYD    REGSTR+12     ;STORE IY
 FB30   31 00FF            LXI     SP,STACK      ;RESTORE STACK
                                ;
                                ;EXAMINE AND DISPLAY CPU REGISTERS
                                ;
 FB33   21 FB5D    XAM:    LXI     H,XAMM        ;POINT TO MESSAGE
 FB36   11 0010            LXI     D,REGSTR      ;POINT TO REGISTERS
 FB39   CD F8BA            CALL    CRLF          ;DISPLAY CRLF
 FB3C   0E08               MVI     C,8           ;SET TALLY
 FB3E   0602       XAM1:   MVI     B,2
 FB40   C5                 PUSH    B             ;SAVE BC
 FB41   CD F895            CALL    MESG1         ;DISPLAY REG NAME
 FB44   CD F8A3            CALL    SPACE         ;DISPLAY SPACE
 FB47   E5                 PUSH    H             ;SAVE HL
 FB48   1A                 LDAX    D             ;GET REGISTER DATA
 FB49   6F                 MOV     L,A
 FB4A   13                 INX     D
 FB4B   1A                 LDAX    D
 FB4C   67                 MOV     H,A
 FB4D   13                 INX     D             ;DEST + 1
 FB4E   CD F8A0            CALL    HLBLK         ;DISPLAY REG DATA
 FB51   CD F8A3            CALL    SPACE         ;DISPLAY SPACE
 FB54   E1                 POP     H             ;GET HL
 FB55   C1                 POP     B             ;GET BC
 FB56   0D                 DCR     C             ;TALLY - 1
 FB57   CA F842            JZ      START         ;DONE
 FB5A   C3 FB3E            JMP     XAM1          ;CONTINUE
                                ;
```

```
FB5D                    XAMM:
FB5D    414642434445    .ASCII  'AFBCDEHLSPIXIYPC'
                        ;
                        ;       MONITOR SIGN-ON MESSAGE
                        ;
FB6D    0D0A            MSG:    .BYTE   CR,LF
FB6F    53595354454D            .ASCII  'SYSTEM MONITOR I.1 REV A'
001A                    MSGL    =       .-MSG
                        ;
                        ;END OF PROGRAM
                        .END
```

SBC880

Assembly Instructions

Construction of the SBC880 from the bare board is intended for those who have experience in electronic assembly. If you do not have this experience we strongly suggest you obtain help in assembling this board.

Be sure you have the proper tools available: a small soldering iron (20 W, 700 degree optimum tip temperature), Rosin Core solder (preferably 63/37), diagonal cutters, a flat blade screw driver and needle nose pliers.

CAUTION
USE EYE PROTECTION WHILE SOLDERING OR CUTTING WIRE

1. ( ) Check the parts received against the partslist. If anything ismissing please contact us and report the shortage.

2. ( ) Install the following sockets in the locations indicated, but DO NOT solder them at this time.

   14 pin sockets at: U1-7,U10,U11,U14,U18,U20,U21

   16 pin sockets at: U17,U26,U29,U30,U31,U32

   18 pin sockets at: U24,25

   20 pin sockets at: U12,U13,U15,U16,U22,U27,U28, U33

   24 pin sockets at: U9,U23

   28 pin socket at: U8

   40 pin socket at: U19

   After all of the sockets have been inserted place a flat piece of cardboard or styrofoam over the sockets and turn the board over. Now solder alternate corner pins of each socket, i.e. pins 1 and 9 of a 16 pin socket. After this has been done for each socket, turn the board over and inspect each socket to determine that it has been seated flat against the board. If this is not the case, re-heat the two soldered pins while pressing down on the socket.

   Now solder the remaining pins of each socket. Do this carefully as the density of the board makes it easy to miss soldering a pin. After you have soldered all the pins, inspect each connection for solder bridges and cold joints.

3. ( ) Install the five SIP resistor packs in the locations indicatedon the board (RP1,RP3-RP5). Be sure to install them properly. The resistor packs can be soldered to the board by a method like that used for the IC sockets.

4. ( ) Install and solder the 1K DIP resistor pack at RP2.

5. ( ) Install and solder the six 1K resistors at locations R1,R3,R4,R6,R7,and R8.

6. ( ) Install and solder the 4.7K resistor at R2.

7. ( ) Install and solder the 330 ohm resistor at R5.

8. ( ) Install and solder the .001 uf capacitor at C5, C6,C7.

9. ( ) Install and solder the 100uf electrolytic cap. at C8. Observe the proper polarity.

10. ( ) Install and solder the .1uf capacitors at C10, C12-C26

12. ( ) Install and solder the 1.5uf tantalum capacitors at C1-C4,C9 and C11. Be sure the + and - polarity is observed.

13. ( ) Install and solder the 4MHz crystal at Y1.

14. ( ) Install the 5 volt regulator and heatsink at VR1. If you have a good heat sink compound, we suggest you use it. Insert a 6-32 machine screw through the board from the solder side, place the heatsink over the screw and align it with the pattern on the board, place the regulator on the screw with the flat side of the regulator against the heatsink, place a lockwasher and #6 nut on the screw and tighten snugly without forcing. Carefully, using needle nose pliers, bend the leads of the regulator such that they can be inserted into the solder holes labeled "IN", "G", and "O". Solder the leads to the board.

15. ( ) Install the 7812 +12 regulator at VR2, a heat sink is not required.

16. ( ) Install the 7905 -5 regulator at VR3, a heat sink is not required.

17. ( ) Install the 7912 -12 regulator at VR4, a heat

sink is not required.

18. ( ) Install header pins at J1 and J2 and solder.

19. ( ) Install and solder the wire-wrap pins or header pins at 2M/4M position. The center to bottom position selects a 4MHz clock and the center to top position selects a 2MHz clock.

Before continuing with the insertion of the ICs in the sockets, inspect the board carefully for shorts or opens caused by solder bridges or cold solder joints. Now insert the un-populated board into your system and check for the proper supply voltages. The output voltages from all the regulators can be measured on the pin facing toward the top of the board (away from the S100 connector). Be careful not to let your probes short the voltage regulator pins together. If all the voltages are up to par (plus or minus half a volt or so), continue to step 20, otherwise, check the board for shorts. Find the short before you install any IC's.

All integrated circuits on this board (except U1,U2,U3,U4) should be insertedwith pin 1 toward the bottom of the board.

20. ( ) InserttheICsat theindicated locations on the assembly drawing.

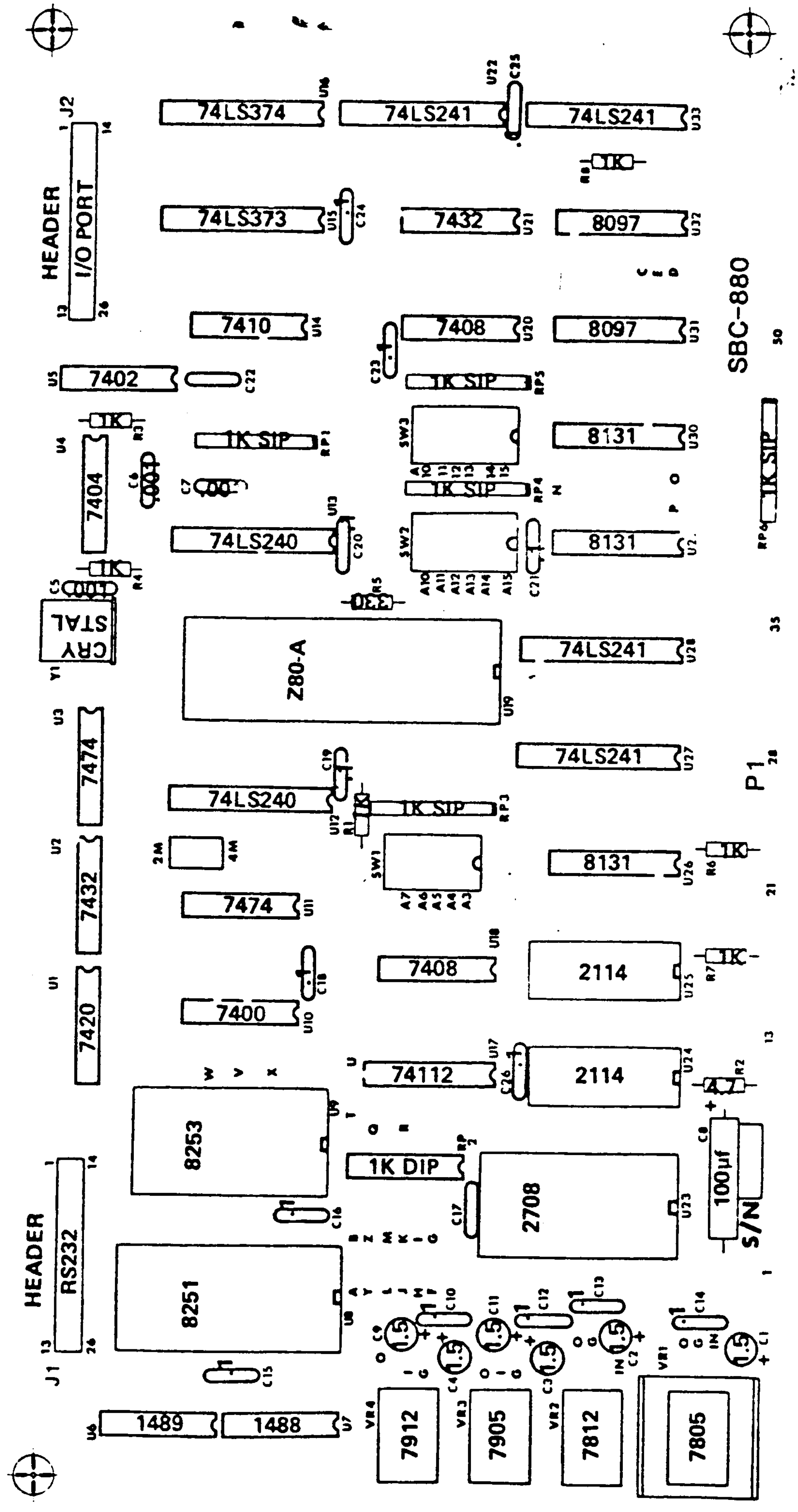
21. ( ) Install whatever options you have chosen from the option list.

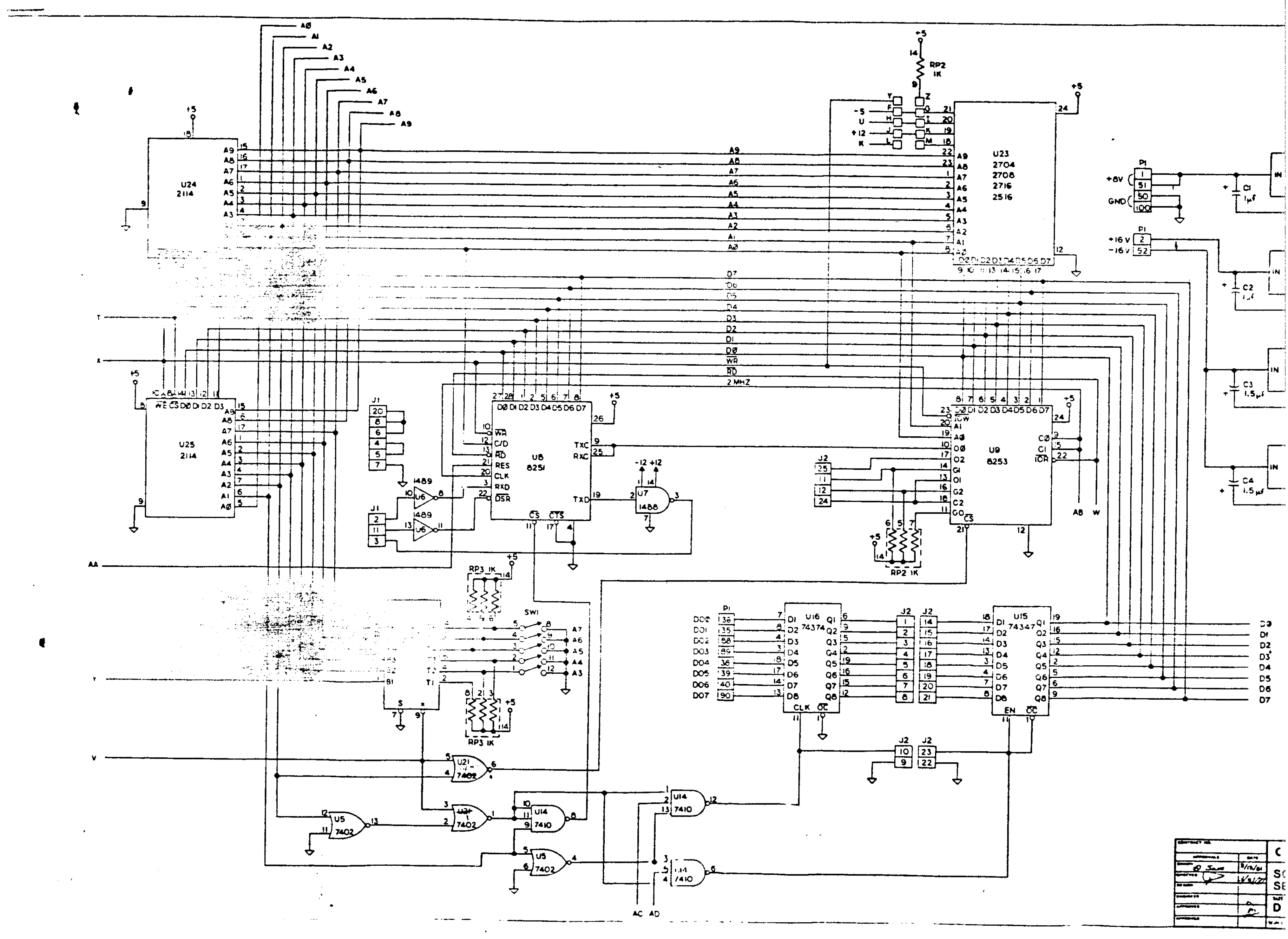
This completes the assembly of the SBC880 board.

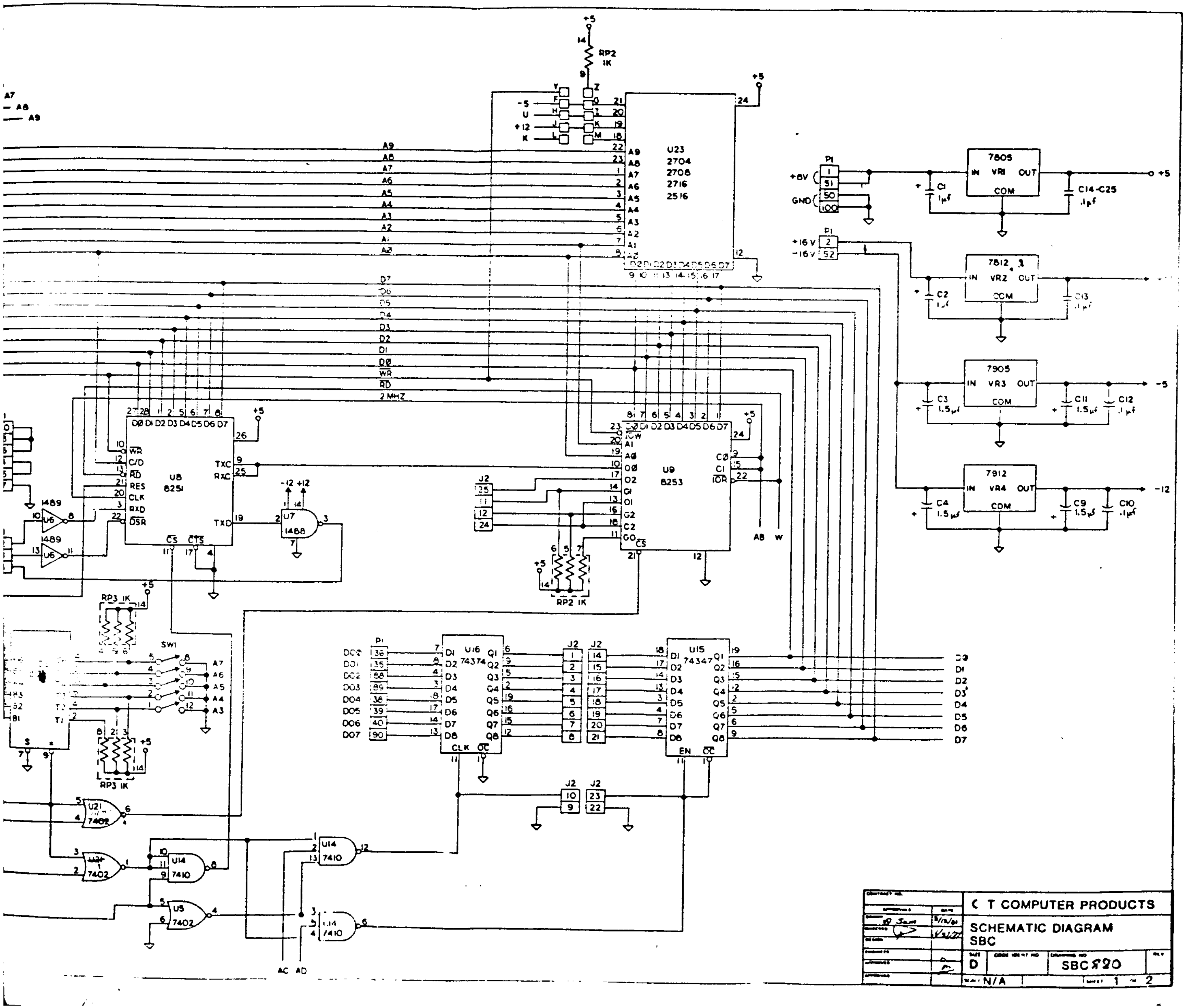
PARTS LIST

Part Number	Quantity	Description
C1-4,9,11	6	1.5 uf Tantalum Capacitor
C10,11-26	16	.1 uf Ceramic Disc Capacitor
C5,6,7	3	.001 uf Ceramic Disc Capacitor
C8	1	100uf Electrolytic Capacitor
R1,3,4,6-8	6	1K ohm, 1/4 watt Resistor
R2	1	4.7K ohm, 1/4 watt Resistor
R5	1	330 ohm, 1/4 watt Resistor
RP1,RP3-6	5	1K 8 pin SIP Resistor Pack
RP2	1	1K ohm 14 pin DIP Resistor Pack
U1	1	7420 DUAL 4 IN. NAND GATES
U2,U21	2	7432 QUAD 2 IN. OR GATES
U3,U11	2	7474 DUAL D FLIP FLOP
U4	1	7404 HEX INV.(do not use 74LS04)
U5	1	7402 NOR GATE
U6	1	1489
U7	1	1488
U8	1	8251 USART
U9	1	8253 OR 8254 TIMER
U10	1	7400 QUAD 2 IN. NAND GATES
U12,U13	2	74LS240 OCTAL BUFFERS
U14	1	7410 TRIPLE 3 IN. NAND GATES
U15	1	74LS373 OCTAL D TYPE LATCHES
U16	1	74LS374 OCTAL D FLIP FLOP
U17	1	74112 DUAL J-K FLIP FLOP
U18.U20	2	7408 QUAD 2 IN. AND GATES

U22,U27,U28,U33	4	74LS241 OCTAL BUFFERS
U23	1	2708 or 2716 EPROM
U24,U25	2	2114 RAM
U26,U29,U30	3	8131 COMPARATORS
U31,U32	2	74LS367 or 8097 HEX BUFFERS
VR1	1	7805 5 Volt Regulator (TO 220)
VR2	1	7812 12 Volt Reg.(TO 220)
VR3	1	7905 -5 Volt Reg. (TO 220)
VR4	1	7912 -12 Volt Reg. (TO 220)
Y1	1	4MHz Crystal
SW1,SW2,SW3	3	Dip Switch 6 Position
J1,J2	2	26 Pin Header (Right Angle)
	13	14 pin Sockets
	6	16 pin Sockets
	2	18 pin Sockets
	8	20 pin Sockets
	2	24 pin Sockets
	1	28 pin Socket
	1	40 pin socket
	3	Wire-Wrap Pins/Headers
	1	Heatsinks
	2	6-32 Screws, Lockwashers, & Nuts


SBC-880
HEADER I/O PORT
J2
HEADER RS232
J1
74LS374
74LS241
74LS241
74LS373
7432
8097
7410
7408
8097
7402
1K SIP
1K SIP
8131
7404
1K SIP
74LS240
8131
CRY STAL
Z80-A
74LS241
7474
74LS241
P1
74LS240
1K SIP
7432
8131
7474
7408
2114
7420
7400
74112
2114
8253
1K DIP
2708
100µf
S/N
8251
1489
1488
7912
7905
7812
7805

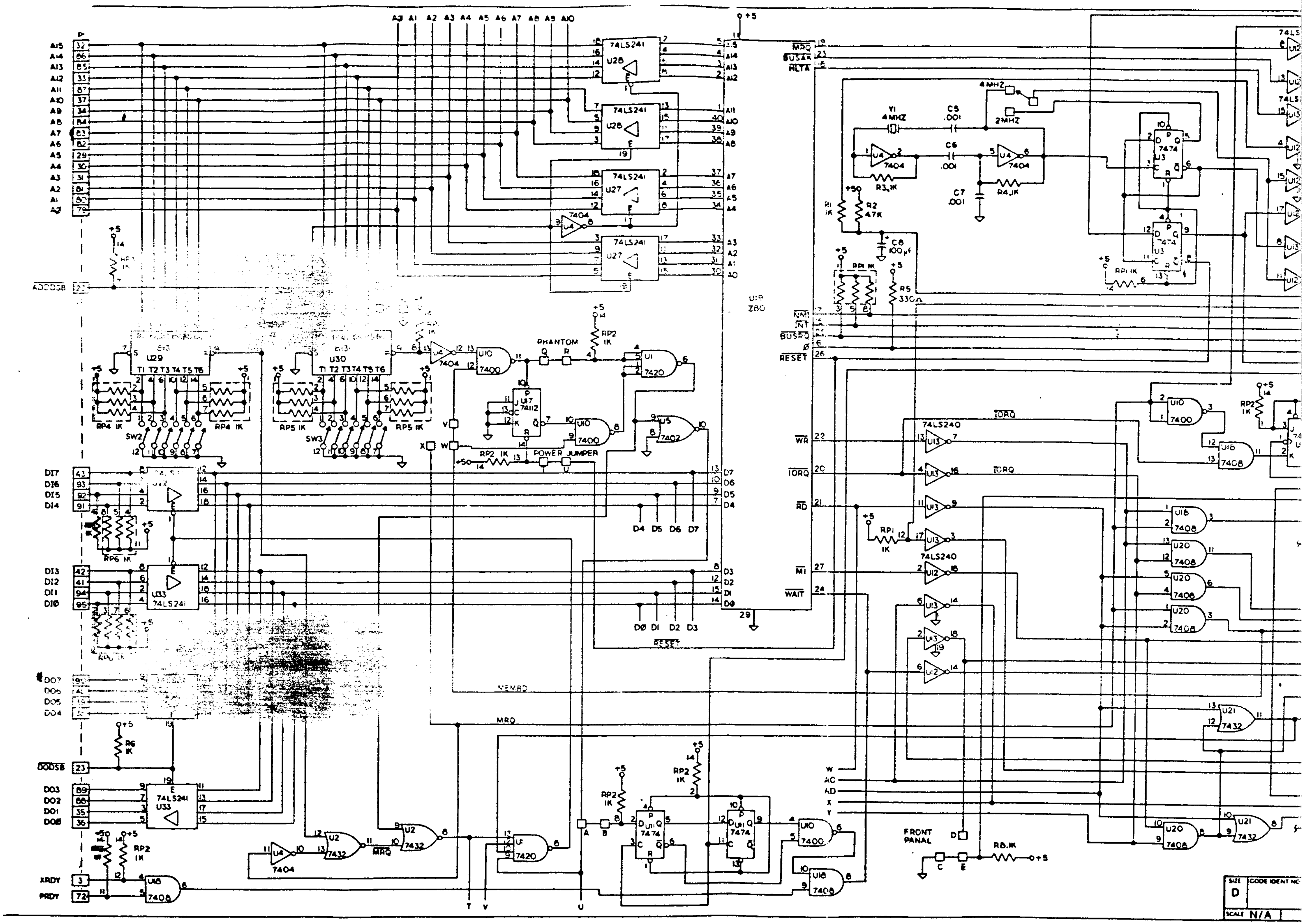

U24
2114
U25
2114
U8
8251
U9
8253
U23
2704
2708
2716
2516
U16
74374
U15
74347
U6
1489
U7
1488
U5
7402
U14
7410
U21
SW1
RP2 1K
RP3 1K
2 MHZ
+8V
GND
+16 V
-16 V

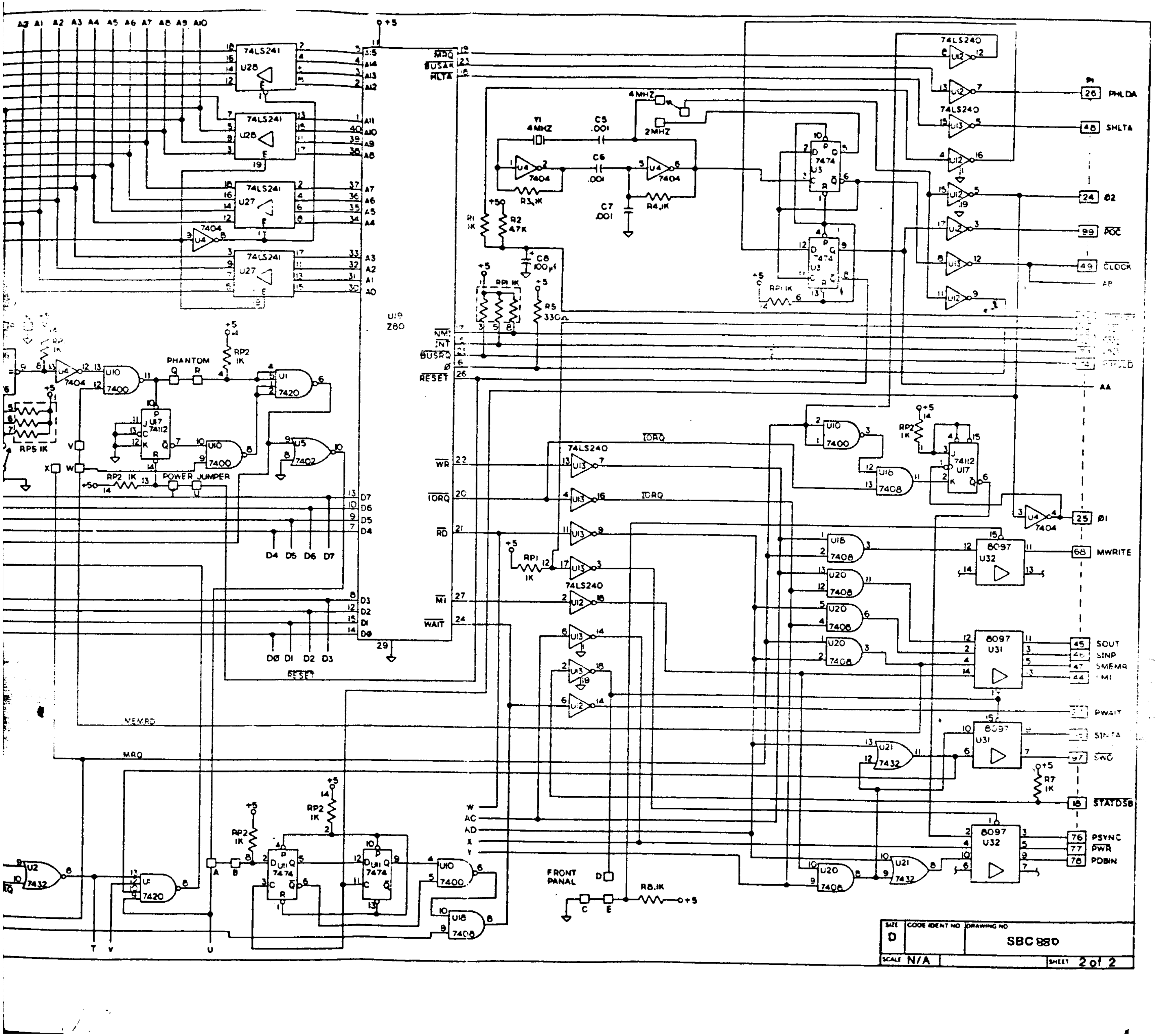


C T COMPUTER PRODUCTS

SCHEMATIC DIAGRAM
SBC

D | SBC 820 | N/A | SHEET 1 of 2


74LS241
U28
U27
U19
Z80
U29
U30
SW2
SW3
RP4 1K
RP5 1K
RP6 1K
RP1 1K
RP2 1K
74LS240
7404
7400
7420
7402
7408
7432
7474
74112
PHANTOM
POWER JUMPER
MEMRD
MRQ
RESET
IORQ
FRONT PANEL
Y1 4MHZ
4MHZ
2MHZ
C5 .001
C6 .001
C7 .001
C8 100µf
R1 1K
R2 4.7K
R3 1K
R4 1K
R5 330Ω
R6 1K
R8 1K
ADDSB
DODSB
XRDY
PRDY
SIZE D
CODE IDENT NO
SCALE N/A



SIZE	CODE IDENT NO	DRAWING NO
D		SBC 880
SCALE N/A		SHEET 2 of 2