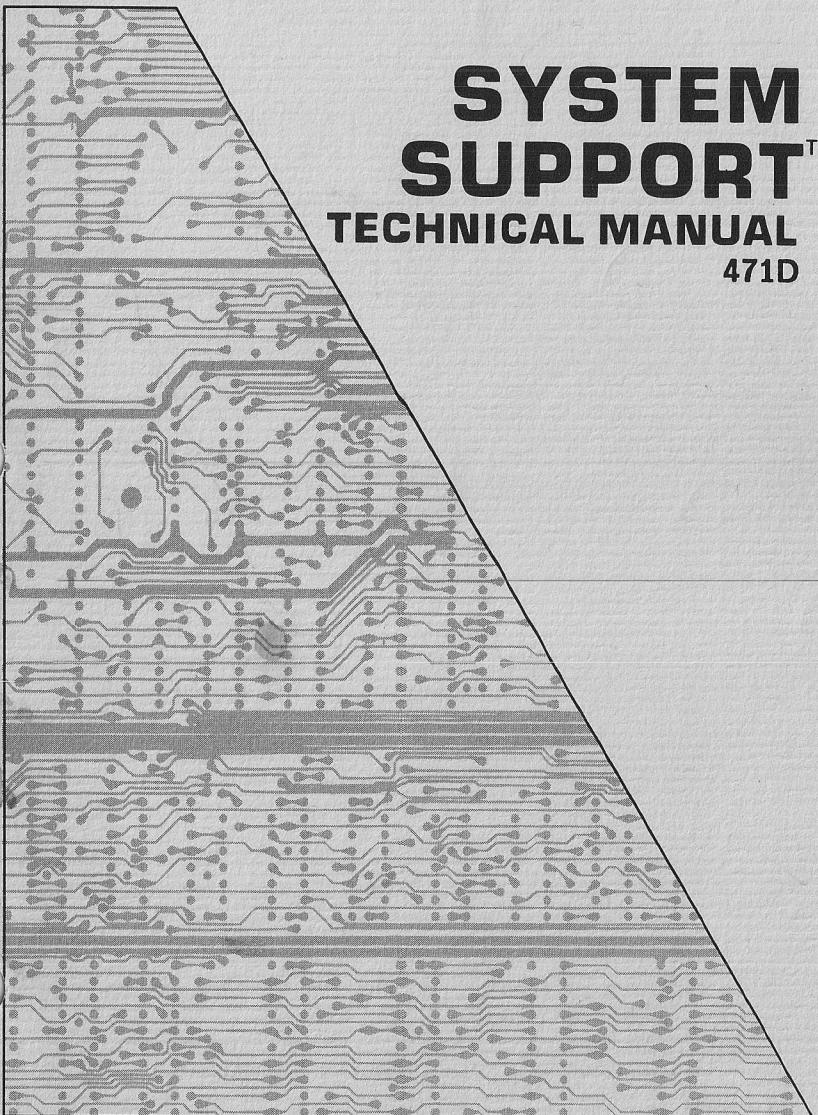


CompuPro™

**SYSTEM
SUPPORT™
TECHNICAL MANUAL
471D**



8261-0084B

June 1988

SYSTEM SUPPORT 2™ Revision D
Technical Manual

System Support 2 Technical Manual
Copyright 1988 Viasyn Corporation
Hayward, CA 94545

Part Number: 8261-0084B
File name: SS2D.MAN

DISCLAIMER - Viasyn Corporation makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, VIASYN reserves the right to revise this publication and to make any changes from time to time in the content hereof without obligation of VIASYN to notify any person of such revision or changes.

Registered Trademarks: Centronics; Centronics Data Computer Corp. CompuPro; Viasyn Corporation.

Compound Trademarks: Concurrent DOS 8-16; Digital Research Inc. and Viasyn Corporation.

Trademarks: The CompuPro logo, CPU 286, CPU 68K, CPU 8085/88, CPU 8086/87, CPU-Z, CPU 32016, DISK 3, System Support 1, System Support 2, ; Viasyn Corporation. Z80; Zilog Corporation.

All rights reserved. No part of this publication may be reproduced or transmitted in any form, or by any means, without the written permission of VIASYN. Printed and assembled in the United States of America.

Preface

This manual describes the features and functions of Revision D of the System Support 2™ board and also includes programming information. This is a reference manual for programmers, hardware engineers and anyone else who needs to understand how the System Support 2 functions in a CompuPro™ computer system. It is neither a troubleshooting guide nor a repair manual.

This manual begins with an overall description of the board and a detailed account of the jumper shunt functions. More details are included in the functional description following the jumper shunt section. Programming considerations, specifications and schematics are also included.

The *Operating System Installation and Customization Guide Concurrent DOS 8-16™* contains information on settings for the operating system to function with the System Support 2 board.

Contents

Overall Description	1
Installing the System Support 2 Board	2
Basic Installation	3
Switch and Jumper Summary	5
Special Situations	13
Functional Description of Parts	15
Interrupt Controllers	15
On-Board Memory	16
Real Time Clock	16
DUART	18
Interval Timer	20
Power Failure Detection Circuit	21
Centronics [®] Port	21
SCSI Port	24
Programming Considerations	27
Interrupt Controllers	27
Enabling Memory Chip Select	29
Real Time Clock/Calendar	30
DUART	35
Interval Timers	37
Centronics [®] Port	38
SCSI Port	40
Appendix A: Specifications	46
Appendix B: Technical Data Sources	47
Appendix C: Schematic Diagram	49
Appendix D: Component Layout	58

LIST OF TABLES

Table 1: Switch S1 Address Bits	5
Table 2: Jumper Shunt JS2 and Jumper J11 Settings	7
Table 3: I/O Port Address Setting	8
Table 4: I/O Port Map	8
Table 5: Summary of RAM/ROM Jumper Settings	11
Table 6: 8259A Interrupt Registers	15
Table 7: RTC Internal Registers	17
Table 8: DUART I/O Bits	18
Table 9: Centronics Signals	22
Table 10: Centronics Status Bits	23
Table 11: Centronics Channel Pin-out	23
Table 12: SCSI Bus Signals	24
Table 13: SCSI Status Register	25
Table 14: SCSI Bus Pin-out	26

REPORT TO THE

CONFIDENTIAL

Overall Description

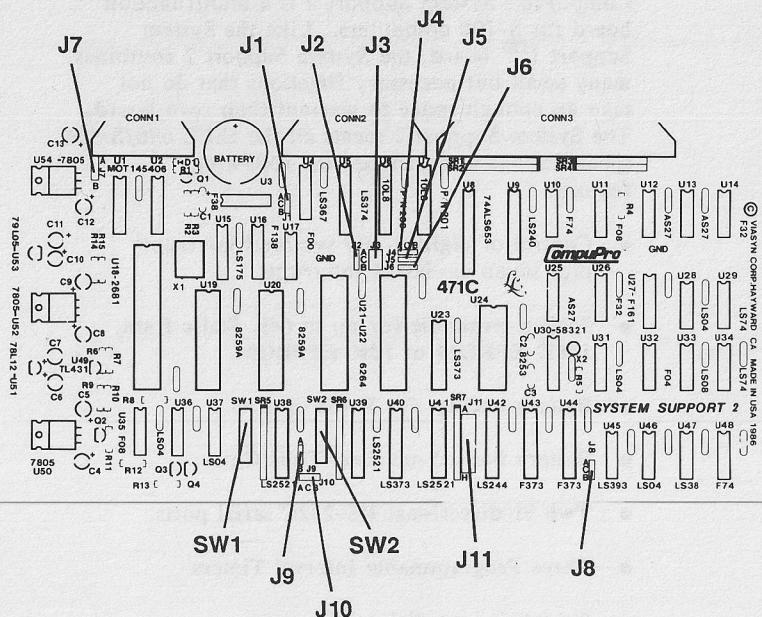
CompuPro's System Support 2 is a multifunction board for S-100 computers. Like the System Support 1TM board, the System Support 2 combines many small but necessary functions that do not take up enough space to warrant their own board. The System Support 2 meets all the IEEE 696/S-100 specifications and includes the following features:

- Control of eight S-100 vectored interrupts plus seven on-board interrupts.
- Sockets available for up to 64K Static Ram, 128K EPROM or 16K EEPROM.
- Battery backup for RAM options.
- Battery backed-up Real Time Clock.
- Two bi-directional RS-232C serial ports.
- Three Programmable Interval Timers.
- Centronics parallel printer port.
- Small Computer Systems Interface (SCSI) port.
- Supported by CompuPro's Concurrent DOS 8-16TM multi-user, multi-tasking operating system.
- Power failure detection.

The separate parts are linked by common Address and Data buses and are orchestrated by common control circuitry.

Further details of the features of the System Support 2 are included in the **Functional Description of Parts** section of this manual.

Installing the System Support 2 Board



STATIC WARNING: Please observe these precautions:

- Store only in anti-static materials. Do not use non-conductive styrofoam or plastic trays or bags.
- Observe proper static discharge techniques when handling boards.
- Keep fabric and other static-generating materials away from boards and circuits.
- Turn off power when inserting or removing boards. Leave all power supplies off for several minutes to be certain no voltage is present.

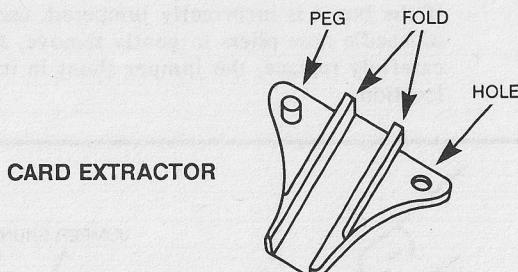
(continued)

Installing the System Support 2 Board

Basic Installation

Step 1. Unpack the Board.

Along with the board, you will find two card extractors in the plastic bag.



Step 2. Install the Card Extractors.

1. Hold the board with the component side toward you. (See diagram below.)
2. Insert the peg on the card extractor into the hole in the right corner of the board. Fold the extractor over the board's edge until the extractor's hole snaps over the peg. Make sure the long edge of the extractor is on the top edge of the board.
3. Repeat for left extractor.

Step 3. Check Switch and Jumper Settings

For standard switch settings for a CompuPro operating system, check the *Operating System Installation and Customization Guide Concurrent DOS 8-16™*. Otherwise, refer to the **Switch and Jumper Summary** in this manual. The locations of the various switches and jumpers on the board are shown in the diagram on the preceding page.

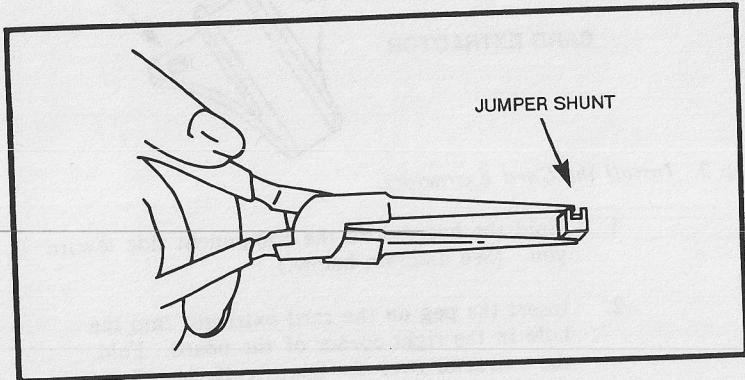
(continued)

Installing the System Support 2 Board

Step 4. How to Install Jumper Shunt Connectors

A jumper shunt is a small plastic part used to connect two pins on the jumper connector. Jumper shunts should be installed notch side up.

If the board is incorrectly jumpered, use a pair of needle nose pliers to gently remove, and carefully replace, the jumper shunt in its proper location.



Step 5. Insert the System Support 2 into the S-100 Bus.

The power to the system must be off. Place the board into a slot towards the back of the enclosure. The edge connector is offset, so the board fits only one way. Push down GENTLY until the board is firmly installed.

Switch and Jumper Summary

This section gives a detailed description of all switch and jumper settings for the System Support 2. In the following tables, a switch setting of **on** corresponds to a binary 0 or low, and **off** corresponds to a binary 1 or high.

Switch S1

Switch S1 is an eight-position dip switch located on the lower left side of the board. Switch paddles 3 through 8 set the memory location of the on-board RAM (or ROM), and each paddle corresponds to a particular address bit as described below. The settings allocate a 256K block of memory space, but the actual amount of memory available depends on the type and number of memory chips installed. Paddles 1 and 2 on the switch are not used.

Table 1: Switch S1 Address Bits

<u>Switch Position</u>	<u>Address Bit</u>
3	A23
4	A22
5	A21
6	A20
7	A19
8	A18

To CPUs with 24 bits of addressing, the memory can appear in any one of sixty-four 256K pages as determined by the settings of switch positions 3 through 8. To CPUs with 20 bits of addressing, the memory can appear in any one of four 256K pages, in which case switch positions 3 through 6 are set to **on** (0) and positions 7 and 8 determine the memory location.

(continued)

Switch and Jumper Summary

The two RAM/ROM sockets (U21 and U22) can accommodate memory chips of several sizes. Normally there are no memory chips in these sockets. The default configuration for this board is a single 6264 (8K by 8-bit Static RAM). To address this RAM size, A13 must be set high in software to enable the chip. A17 is used to select between the two RAM/ROM sockets.

With a 6264 in U22, paddles 3-6 on and paddles 7 and 8 off, the memory resides in locations C2000h to C3FFFh. If another 6264 is in U21, this memory is at E2000h to E3FFFh.

Other memory chips that can be used are described in the on-board memory section of this manual.

Jumper Shunt JS2 (Positions 1 - 4)

Jumper shunt JS2 is an eight-position jumper shunt located in the lower left of the board. Jumper shunt positions 1 - 4 set address bits A7 through A4 (see Table 2 below) to determine the base or starting address of the I/O ports on the System Support 2 board. The possible position 1 - 4 jumper shunt settings and their corresponding base addresses are shown in Table 3.

Note: These shunts determine only the high nibble (A7-A4) of the low byte. The low nibble (A3-A0) of the low byte selects which of the 16 ports is addressed. These low nibble bits are addressed in software. The standard CompuPro port map places the 16 I/O ports as shown in Table 4.

(continued)

Switch and Jumper Summary

When the eight position jumper shunt is installed, a connection is made to ground at each position. The ground connection corresponds to a binary 0 or low. To get a binary 1, or high, the connection must be punched out or cut for that position. In the CompuPro standard configuration, positions 2 and 4 are punched out to set bits A7 to A4 as 05h. Positions 5 - 8 are left as 0, but are actually "don't care" in the CompuPro standard configuration.

Address bits A15 through A8, which determine the upper byte for the hex address, are set by positions A through H of Jumper J11, as shown in Table 2. Normally, these bits are hardwired low (ground) by J11. To set the base address above 00F0h, traces on the board must be cut. To do this, carefully cut the trace between the jumper shunt holes of the address bit you wish to set to 1 or high.

The standard CompuPro setting for jumper shunt JS2, positions 1 - 4 places the 16 I/O ports at locations 0050h to 005Fh.

Table 2: Jumper Shunt JS2 and Jumper J11 Settings

Jumper J11 Position	Jumper Shunt JS2 Position	Jumper Shunt JS2 Address Bit	Jumper J11 Address Bit
A	1	A15	A7
B	2	A14	A6
C	3	A13	A5
D	4	A12	A4
E		A11	
F		A10	
G		A9	
H		A8	

(continued)

Switch and Jumper Summary

With the upper address bits fixed at 00, the possible base addresses for the ports are:

Table 3: I/O Port Address Settings

<u>Jumper Shunt JS2 Position:</u>				<u>Base Address of Ports:</u>
1	2	3	4	
off	off	off	off	00F0h
off	off	off	on	00E0h
off	off	on	off	00D0h
off	off	on	on	00C0h
off	on	off	off	00B0h
off	on	off	on	00A0h
off	on	on	off	0090h
off	on	on	on	0080h
on	off	off	off	0070h
on	off	off	on	0060h
on	off	on	off	0050h
on	off	on	on	0040h
on	on	off	off	0030h
on	on	off	on	0020h
on	on	on	off	0010h
on	on	on	on	0000h

Starting from the base location set by jumper shunt JS2, the following port map exists:

Table 4: I/O Port Map

Master 8259A (A0=0)	0h
Master 8259A (A0=1)	1h
Slave 8259A (A0=0)	2h
Slave 8259A (A0=1)	3h
Timer/Counter 0	4h
Timer/Counter 1	5h
Timer/Counter 2	6h
Timer/Control	7h
Centronics Command/Status	8h
Centronics Data	9h
Real Time Clock Address	Ah
Real Time Clock Data	Bh
Duart Address	Ch
Duart Data	Dh
SCSI Command/Status	Eh
SCSI Data/Acknowledge	Fh

Switch and Jumper Summary

Jumper Shunt JS2 (Positions 5 - 8)

Jumper shunt JS2 positions 5 and 6 are connected to the 2681 DUART IP bits 5 and 6, which are user definable input bits (refer to Table 8). Positions 7 and 8 are not used.

Jumper J1 This jumper, located in the upper left hand side of the board, determines which Centronics signal, BUSY* or ACK*, will assert an interrupt to the on-board interrupt controllers (8259As). The ACK* signal is connected via a trace on the board. If BUSY* is to drive the interrupt, cut the trace on the board between the A and C pins and install a jumper shunt between B and C.

Jumpers J2 and J6

Jumpers J2 and J6 reside in the middle of the board. These jumpers select the signals for the memory options in the RAM/ROM sockets. Chips that may be installed in the sockets are: 6264 (standard), 62256, 2764, 27128, 27256, 27512, 2817A EEPROM and 2864A EEPROM.

J2 selects a write enable or address bit 14 for the RAM/ROM socket in U21. If a 6264, 62256, 2817A or 2864A is installed, J2 must be connected from A to C to receive a write enable signal. If a 2764, 27128, 27256 or 27512 is installed, J2 must be connected B to C to receive address bit 14.

(continued)

Switch and Jumper Summary

J6 selects a write enable or address bit 14 for the RAM/ROM socket in U22. If a 6264, 62256, 2817A or 2864A is installed, J6 must be connected from B to C to receive a write enable signal. If a 2764, 27128, 27256, or 27512 is installed, J6 must be connected A to C to receive address bit 14.

NOTE: To the 2764 and 27128 this is the PGM* signal that must be set high in software. If you are writing software for the System Support 2 and need to address a 2764 or 27128, you must be sure the PGM* signal is high. This is done by keeping address line 14 high during any access to the ROM area. To keep A14 high, you must add 4000h to the starting address. For example, if you want to start the address C0000h, you must actually start at C4000h.

- Jumper J3* J3 (located in the middle of the board) selects address bit 14, address bit 15 or +5V power (or none of these) for the RAM/ROM sockets. J3 is the jumper with three rows of pins. The top row (A) selects +5V for the 2764, 27128 and 27256 memory devices. The bottom row (B) selects address bit 14 for the 62256, if it is installed. The middle row (C) selects address bit 15 for the 27512, if it is present. If the 6264, 2817A or 2864A is installed, NO jumper shunt is installed. Pin 3 of the socket is the RDY/BUSY* signal of the 2817A, which is an output signal. If a 2817A is installed in U22, the only chip that can be installed in U21 is another 2817A.

Switch and Jumper Summary

Jumpers J4 and J5

Jumpers J4 and J5 (located in the top middle of the board) determine if the sockets will receive battery backed up power or power from the S-100 bus. J4 controls the power selection for one of the RAM/ROM sockets, and J5 controls the selection for the other socket. With a shunt installed from A to C, the socket will receive battery backed up power. With a shunt installed in the B to C position, the socket will receive power from the +5V supply only. If no chip is installed in the socket, no jumper is required. If there is a ROM or an EEPROM installed, use S-100 power (B to C).

Jumper settings for the RAM/ROM sockets are summarized as follows:

Table 5: Summary of RAM/ROM Jumper Settings

Jumper	6264	62256	2764	27128	27256	27512	2817A	2864A
J2	A-C	A-C	B-C	B-C	B-C	B-C	A-C	A-C
J3	No Shunt	Row B	Row A	Row A	Row A	Row C	No Shunt	No Shunt
J4	A-C	A-C	B-C	B-C	B-C	B-C	B-C	B-C
J5	A-C	A-C	B-C	B-C	B-C	B-C	B-C	B-C
J6	B-C	B-C	A-C	A-C	A-C	A-C	B-C	B-C

Switch and Jumper Summary

Jumper J7 - RS-232 Drive Voltage Select

Jumper J7 selects the positive voltage for the RS-232 output drivers. J7 shunted A-C selects +12V; J7 shunted B-C selects +5V. Whether J7 is set A-C or B-C, the System Support 2 board meets the RS-232 specifications for voltage swings.

If a 79L12 -12V regulator is installed in U53, then J7 must be set A-C and a 78L12 regulator must be installed in U51. If a 79L05 -5V regulator is installed in U53, then J7 must be set B-C and U51 need not be installed.

Jumper J8

This jumper (located on the bottom right) allows the System Support 2 to power up with the RAM/ROM sockets activated or inactivated. With a shunt installed from A to C, the RAM/ROM sockets will not respond on power up. With B to C shunted, the RAM/ROM sockets will respond on power up. Normally, there is a shunt from A to C in this jumper.

Jumper J9

This jumper (located on the bottom left) offers the option of generating an interrupt to the 8259A interrupt controller when RDY/BUSY* is driven high by the 2817A EEPROM (if it is installed). In this case, the signal AS is connected to the 8259A by installing a shunt from A to C. NDEF (S-100 pin 66) can also be selected as an interrupt with a shunt installed from B to C. NDEF is user definable and can be used as a vectored interrupt line. Normally no shunt is installed in this jumper.

Switch and Jumper Summary

- Jumper J10* The System Support 2 can generate PHANTOM* when its memory is accessed, or disable itself when another board asserts PHANTOM*, or do neither. The System Support 2 asserts PHANTOM* when a jumper is installed from B to C in jumper J10 (located on the bottom left). If a shunt is installed from A to C, the memory on the System Support 2 will be disabled whenever other boards assert PHANTOM*. Normally, there is no shunt in this jumper.
- Jumper J11* See the discussion in the section on jumper shunt JS2 for an explanation of this jumper.

Special Situations

Note: The System Support 2 was designed to run with the 12 MHz, two-cycle CPU 286™ and may not work with older processors. S-100 pins 3 and 21 are grounded on the System Support 2. Pin 3 is XRDY and pin 21 is an NDEF pin, previously used by CompuPro as Φ disable. Pins 3 and 21 are inputs on the CPU 8085/88™, CPU 8086/87™. Pin 3 is an input on DISK 3™ boards earlier than Revision J. On the CPU 68K™, pin 21 is a jumperable input (J11) and pin 3 is used only on boards earlier than Revision H.

To modify these boards for use with the System Support 2, disconnect pins 3 and 21 from their IC inputs (or remove jumpers) and pull the disconnected inputs on the ICs on the board up to +5V. (This is not required if a jumper is present.)

Also, the System Support 2 will not work with operating systems prior to Concurrent DOS 5.0. Refer to your operating system installation guide to see if the System Support 2 is supported.

(continued)

Switch and Jumper Summary

CompuPro standard operating systems will only support one System Support board. If you have a System Support 1 board, remove it from the system before installing the System Support 2.

If you wish to keep a System Support 1 board in the same machine with a System Support 2 (not supported by CompuPro), you will have to set the ports for one of the boards in a location other than 50-5Fh. You will also have to disable the 8259A interrupt controllers on one of the boards. See the System Support 1 technical manual for information on setting the I/O port space on the System Support 1 and for instructions on how to disable the 8259As.

To disable the 8259As on the System Support 2, perform the following steps:

1. Carefully remove the IC in U44. Bend pin 4 of the IC up and away from the chip.
Reinstall the IC, making sure the bent-out pin does not contact either the socket or any other IC pin.
2. Carefully remove the IC in U14. Bend pin 8 of the IC up and away from the chip.
Reinstall the IC, making sure that the bent out pin does not contact either the socket or any other IC pin.
3. On the solder side of the board, solder a jumper wire between U44 pin 4 and U44 pin 1 (which is grounded).

Functional Description of Parts

Interrupt Controllers

The System Support 2 uses two 8259A chips (one master, one slave) as interrupt controllers. These chips control and prioritize the eight vectored interrupts from the S-100 bus, plus seven on-board interrupts. Any (or all) of the interrupts may be masked. The 8259A accepts commands from and releases information to 8085- and 8086-type CPUs and does not support Z80TM - type CPUs.

The 8259s are addressed through relative ports 00-03 (50-53h standard). Interrupts controlled by these chips are the eight vectored interrupts of the S-100 bus, plus interrupts from the 8253 interval timer, the Centronics port, the SCSI port, and the DUART on the board. Jumper 9 allows the selection of an interrupt from the RAM/ROM socket when an EEPROM is present, or an interrupt from the S-100 NDEF pin 66. The interrupt lines for the master and slave are as follows:

Table 6: 8259A Interrupt Registers

<u>Interrupt Register</u>	<u>Interrupt Signal</u>
Master:	
IR0	VI 0 (S-100)
IR1	VI 1 (S-100)
IR2	VI 2 (S-100)
IR3	VI 3 (S-100)
IR4	VI 4 (S-100)
IR5	VI 5 (S-100)
IR6	VI 6 (S-100)
IR7	Slave 8259A
Slave:	
IR0	VI 7 (S-100)
IR1	Interval Timer 0
IR2	Interval Timer 1
IR3	Interval Timer 2
IR4	Centronics ACK* or BUSY*
IR5	SCSI
IR6	DUART
IR7	EEPROM or NDEF

Functional Description of Parts

For more details, see Appendix B for information on how to obtain a data sheet and application notes on the 8259A.

On-Board Memory

Two sockets reside on the System Support 2 for RAM or ROM with 8-bit data bus access. Some chips that can be put in these sockets are: 62256 (32K by 8-bit Static RAM), 2764 (8K by 8-bit EPROM), 27128 (16K by 8-bit EPROM), 27256 (32K by 8-bit EPROM), 27512 (64K by 8-bit EPROM), 2817A (2K by 8-bit EEPROM) and 2864A (8K by 8-bit EEPROM). The RAM/ROM chips in U21 and U22 are in the lower 28 pins of a 32-pin footprint. The 32-pin footprint can be used to accommodate pin-compatible one megabit EPROM's.

With the 2817A in place, pin 3 is the output signal RDY/BUSY*. Be sure no shunt is in J3 if a 2817A is present. This signal can be used to generate an interrupt at the 8259A by installing a shunt from A to C in jumper J9.

Battery backup power is available through Jumpers J4 and J5 for RAM chips. See the **Switch and Jumper Summary** section of this manual for more information on jumper settings for the RAM/ROM sockets.

Real Time Clock/Calendar

The System Support 2 has a complete time-of-day clock on board that will count seconds, minutes, and hours of the day as well as keep track of the date, day of the week, month and year. The Real Time Clock (RTC) is addressed at relative ports 0Ah and 0Bh and has battery backup so time and date information is not lost when system power is off.

(continued)

Functional Description of Parts

To write to the RTC, the BUSY* signal must be in the high portion of its cycle. This signal can be monitored through the IP bit 4 of the DUART. If the BUSY* signal is disasserted, the chip can be written to by first writing to the address port 0Ah and then sending the data to data port 0Bh. The RTC can be read using the BUSY* signal as described above, or it can be read without monitoring the BUSY* signal. To do this, write to the address port of the register you wish to read and then read from data port 0Bh. The port should be read twice and the results compared in software to determine if the value read is valid.

The internal registers of the RTC are as follows. The address of the register must be sent to the System Support 2 port 0Ah. There are two registers each for seconds, minutes, hours, days, months and years. One register is for the one's place and the other is for the ten's place.

Table 7: RTC Internal Registers

<u>Register</u>	<u>Address</u>
Seconds (1)	0h
Seconds (10)	1h
Minutes (1)	2h
Minutes (10)	3h
Hour (1)	4h
Hour (10)	5h
Day of the Week	6h
Date (1)	7h
Date (10)	8h
Month (1)	9h
Month (10)	Ah
Year (1)	Bh
Year (10)	Ch

See the programming section of this manual for more information on addressing the RTC. For more details on the 58321 RTC, see Appendix B for information on obtaining a data sheet.

Functional Description of Parts

DUART A 2681 Dual Asynchronous Receiver/Transmitter (DUART) controls the two serial channels of the System Support 2. The 2681 can control baud rate, word length, parity and stop bits in RS-232 communications.

The DUART is addressed with relative ports 0Ch and 0Dh. Port 0Ch selects the register in the DUART that is being addressed, and port 0Dh receives/sends data, commands and status. There are 16 internal ports on the DUART that can be addressed with port 0Ch. Data bits D3 to D0 map directly into the A3 - A0 pins on the DUART. The DUART data sheet explaining the addressing of these 16 ports is available on request from Signetics (see Appendix B).

The DUART has eight output bits (OP bits) and seven input bits (IP bits). These bits are defined as follows.

NOTE: To set an OP bit low (0), a high (1) must be written to the set OP register. To set an OP bit high (1), a high (1) has to be written to the reset OP register.

Table 8: DUART I/O Bits

<u>Bit</u>	<u>Function</u>
OP 0	CTS out A - Clear to Send output for channel A. This bit goes to the RS-232 CTS pin for channel A, which is pin 8 for DB-9s (pin 5 for DB-25s).
OP 1	CTS out B - Clear to send output for channel B. This bit goes to the RS-232 CTS pin for channel B, which is pin 8 for DB-9s (pin 5 for DB-25s).

(continued)

Functional Description of Parts

Table 8: DUART I/O Bits
(Continued)

OP 2	RR - Enables the chip select on the RAM/ROM socket. Works in conjunction with jumper J8. The OP bits power up with a high level. If a jumper shunt is installed from B to C, the RAM/ROM sockets will power up with chip select asserted. If a jumper shunt is installed from A to C in J8, the RAM/ROM sockets will power up with the chip selects disasserted. The OP 2 bit then has to be set low to enable the RAM/ROM chip select.
OP 3	AUTOFD - Centronics printer control signal Auto Feed.
OP 4	INIT - Centronics printer control signal Initialize.
OP 5	SLCTIN - Centronics printer control signal Select In.
OP 6	RTS - Request to Send output to RS-232 port for channels A and B which is pin 7 for DB-9 connectors (pin 6 for DB-25s). (Not requesting input on reset.)
OP 7	Not used.
IP 0	DTR in A - Data Terminal Ready input A. This bit comes from the RS-232 DTR pin for channel A, which is pin 4 for DB-9s (pin 20 for DB-25s).
IP 1	DTR in B - Data Terminal Ready input B. This bit comes from the RS-232 DTR pin for channel B, which is pin 4 for DB-9s (pin 20 for DB-25s).
IP 2	DCDA - Data Carrier Detect on channel A. This bit comes from the RS-232 Data Carrier Detect pin 1 on DB-9s (pin 8 on DB-25s) channel A. This bit is normally only used when the System Support 2 is DTE.

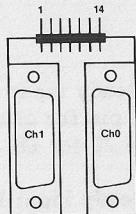
Functional Description of Parts

Table 8: DUART I/O Bits
(Continued)

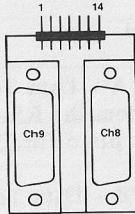
<u>Bit</u>	<u>Function</u>
IP 3	DCDB - Data Carrier Detect on channel B. This bit comes from the RS-232 Data Carrier Detect pin 1 on DB-9s (pin 8 on DB-25s) channel B. This bit is normally only used when the System Support 2 is DTE.
IP 4	RTCBSY* - Busy signal from the Real Time Clock. The RTC is busy and must not be written to when this signal is low.
IP 5	User definable as low or high with jumper shunt JS2, position 5.
IP 6	User definable as low or high with jumper shunt JS2, position 6.

Signals from the DUART go to a 14-pin edge connector. The signals then travel over a ribbon cable to an I/O backplane board.

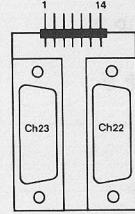
System Support 2
or SP186 device 0



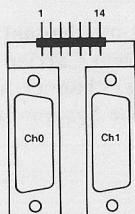
SP186 device 4



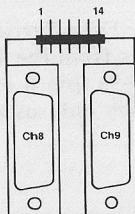
SP186 device 11



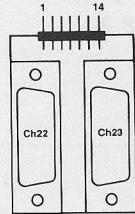
System Support 2
or SP186 device 0



SP186 device 4



SP186 device 11



See Appendix B for information on obtaining RS-232 specifications.

Functional Description of Parts

Interval Timer

There is a programmable interval timer (8253) on the board that can generate time delays under software commands. This chip is addressed at relative ports 04-07 and its internal registers can be addressed through address bits A0 and A1.

There are three output signals from the 8253 that can generate interrupts to the 8259A interrupt controller. The timer's clock value is 2 MHz.

See Appendix B for information on obtaining a data sheet for the 8253.

Power Failure Detection Circuit

The System Support 2 includes a circuit that allows for early detection of power failure in the system. This circuit will assert PWRFAIL* when the +5V signal drops below threshold.

Centronics Port

There is a Centronics parallel interface port on the System Support 2 to connect directly to Centronics style printers. The Centronics' port is addressed at relative ports 08 and 09 and consists of eight data lines plus status and control lines. The output strobe line conforms to the timing specifications of Centronics interface printers. The printer control signals AUTOFD*, INIT* and SLCT IN* are controlled by OP bits 3-5 of the 2681 DUART.

An interrupt is generated with the ACK* pulse when the printer is ready to receive more data. An interrupt can be generated on BUSY* with Jumper J1. To do this, cut the trace between A and C and install a shunt from B to C on J1.

A summary of Centronics' signals is on the next page. Consult your printer manual for information on how these signals work in your printer.

Functional Description of Parts

Table 9: Centronics Signals

<u>Name</u>	<u>Signal</u>
ACK*	Acknowledge - A status signal from the printer that indicates its operation is complete and it is ready to receive more data.
AUTOFD*	Auto feed - A control signal from the System Support 2 that sets the automatic line feed.
BUSY	Busy - A status signal from the printer that indicates the printer is busy and is not ready to receive data. This signal is inverted for use on the board.
ERROR*	A status signal from the printer indicating a printer error.
INIT*	Initialize - A signal sent to the printer from the System Support 2 for initialization.
PE	Paper error - A status signal from the printer indicating it is out of paper.
SLCT	Select - A status signal from the printer that indicates the printer is selected.
SLCTIN*	Select in - A control signal sent to the printer to select it.
STROBE*	Data strobe pulse signal from the System Support 2.

Functional Description of Parts

The status bits appear at data bits BD0 - BD4 when relative port 08 is read. The status bits are defined as follows:

Table 10: Centronics' Status Bits

<u>Data Bit</u>	<u>Signal</u>
D0	BUSY* - Printer busy when low.
D1	ACK* - Transfer acknowledge low pulse
D2	PE - Paper error when high
D3	ERROR* - Printer error when low
D4	SLCT* - Printer selected when high

The Centronics' signals pass from the System Support 2 to the back panel of the computer over a ribbon cable to a DB-25 (D Subminiature) female connector. The pin-out for the connector is as follows. For more information on the Centronics' specifications, see Appendix B.

Table 11: Centronics Cable Pin-out

<u>Pin Number</u>	<u>Signal</u>
1	STROBE*
2	Data Bit 0
3	Data Bit 1
4	Data Bit 2
5	Data Bit 3
6	Data Bit 4
7	Data Bit 5
8	Data Bit 6
9	Data Bit 7
10	ACK*
11	BUSY
12	PE
13	SLCT
14	AUTOFD*
15	ERROR*
16	INIT*
17	SLCTIN*
18-26	Ground

Functional Description of Parts

SCSI Port

The System Support 2 also contains a SCSI port for communications with peripheral I/O devices, residing at relative ports 0Eh and 0Fh. Data transfers to and from this port go through port 0Fh. Status from the SCSI port comes in inverted from the SCSI bus through port 0Eh. SCSI port commands SEL* and RST* go out through port 0Eh.

As defined by the SCSI specification, an **initiator** in SCSI information transfers is a device that requests the performance of an operation by another SCSI device. The initiator is usually the host system. A **target** is the device that performs the operation requested by the initiator. The System Support 2 supports a single initiator, non-arbitrating SCSI system and uses the following SCSI signals:

Table 12: SCSI Signals

Name	Signal
ACK*	Acknowledge - An initiator (System Support 2) driven signal that indicates an acknowledge for a REQ/ACK data transfer handshake.
BSY*	Busy - A status signal from the target that indicates that the SCSI bus is in use.
C*/D	Control/Data - A status signal from the target that indicates if the information on the data bus is control information or data. A low indicates control information.
DB0-7	Data bus bit 0 to 7.
DI*/O	Data/I/O - A control signal from the target that indicates the direction of the data transfer on the bus with respect to the initiator.

(continued)

Functional Description of Parts

Table 12: SCSI Signals
(Continued)

<u>Name</u>	<u>Signal</u>
MSG*	Message - A status signal that the target drives low during the message phase.
REQ*	Request - A status signal that indicates that a target is requesting a REQ/ACK data transfer handshake.
RST*	Reset - A control signal from the System Support 2 that causes a reset condition.
SEL*	Select - A control signal from the System Support 2 used to select a target.

The status bits are read through relative port 0Eh with the data bits as indicated below. These bits are inverted from the SCSI bus signals.

Table 13: SCSI Status Register

<u>Data Bit</u>	<u>Signal</u>
D0	BSY - High when SCSI bus is in use
D4	MSG - High when in Message Phase
D5	C/D* - High in Command Phase
D6	DI/0* - High in Data In Phase
D7	REQ - High when target requests data or commands

Functional Description of Parts

The pin-out for the SCSI bus follows. All signals are connected to a 50-pin edge connector and have pull up and pull down resistors as indicated in the SCSI specification.

Table 14: SCSI Bus Pin-out

<u>Pin</u>	<u>Signal</u>
2	Data bit 0
4	Data bit 1
6	Data bit 2
8	Data bit 3
10	Data bit 4
12	Data bit 5
14	Data bit 6
16	Data bit 7
36	BSY*
38	ACK*
40	RST*
42	MSG*
44	SEL*
46	C*/D
48	REQ*
50	DI*/O

Odd pins 1 to 49, even pins 18 to 30 and pin 34 are all grounded. Pin 32 is tied to +5V with a pull-up resistor.

See Appendix B for more information regarding the SCSI standard.

Programming Considerations

The following software examples are designed to run with the CompuPro standard operating system. If you are programming in another environment, study the code listings before adapting your code. No representation is made that this is the best way to program the elements of the System Support 2. Rather, the code is written to illuminate the quirks and pitfalls of programming the System Support 2.

Programming the Interrupt Controllers

The 8259As reside at relative ports 00 to 03. Ports 00 and 01 address the master's ports, and ports 02 and 03 address the slave's ports. In the first port of each set, the address bit 0 is equal to 0. In the second port, the address bit 0 is equal to 1.

The chips require several control words for initialization. In this example, four initialization control words (ICW's) and two operational control words (OCW's) are sent to the master and to the slave. These words tell the chip how to handle the interrupts. The following code sets up the interrupt controllers with the following features: level triggered, cascade mode, ICW4 needed, IR7 input has a slave, special fully nested mode, non-polled mode.

```
BASE EQU 50H ;Port base address
MPRTA EQU BASE ;master port A (AO=0)
MPRTB EQU BASE+1 ;master port B (AO=1)
SPRTA EQU BASE+2 ;slave port A (AO=0)
SPRTB EQU BASE+3 ;slave port B (AO=1)
;
MOV AL,1DH ;ICW1 master
OUT MPRTA,AL;ICW4 needed
;
MOV AL,40H ;ICW2 master
OUT MPRTB,AL;address table
;starts at 80H
;
```

(continued)

Programming Considerations

```
        MOV AL,80H    ;ICW3 master
        OUT MPRTB,AL ;slave on IR7
;
        MOV AL,11H    ;ICW4 master
        OUT MPRTB,AL ;special fully
                      ;nested mode
;
        MOV AL,7FH    ;OCW1 master
        OUT MPRTB,AL ;slave input mask
;
        MOV AL,08H    ;OCW3 master
        OUT MPRTA,AL ;non polled mode
;
        MOV AL,1DH    ;ICW1 slave
        OUT SPRTA,AL ;ICW4 needed
;
        MOV AL,48H    ;ICW2 slave
        OUT SPRTB,AL ;address table
                      ;starts at 48H
;
        MOV AL,07H    ;ICW3 slave
        OUT SPRTB,AL ;slave identifier
;
        MOV AL,11H    ;ICW4 slave
        OUT SPRTB,AL ;special fully
                      ;nested mode
;
        MOV AL,0      ;OCW1 slave
        OUT SPRTB,AL ;enable all interrupts
;
        MOV AL,080H   ;OCW3
        OUT SPRTA,AL ;non polled mode
```

on to other code ...

Sample code fragment to enable an interrupt:

```
CLI           ;disable interrupts while
IN  AL,MPRTB   ; modifying mask register
AND AL,11111110b ;unmask IRO of master
OUT MPRTB,al
STI
```

(continued)

Programming Considerations

Sample interrupt routine:

```
V10_ENT: PUSH AX
          :
          Other code can be done here
          :
          MOV AL,60H ;specific end of interrupt
          OUT MPRTB,AL ; for IRO on master
          POP AX
          IRET
```

Enabling the Chip Select to the RAM/ROM Socket

The OP 2 bit of the DUART enables the chip select on the RAM/ROM socket. This bit works in conjunction with jumper J8. The OP bits power up with a high level. If a jumper shunt is installed from B to C, the RAM/ROM sockets will power up with chip select enabled. If a jumper shunt is installed from A to C in J8, the RAM/ROM sockets will power up with the chip selects disasserted. The OP 2 bit then must be set low to enable the RAM/ROM chip select.

In the following example, J8 is assumed to have a shunt from A to C. The chip select to the RAM/ROM socket is then enabled through OP 2. The DUART is at ports 5Ch and 5Dh.

```
DUADD EQU 5C
DUDATA EQU 5D
MOV AL,0EH ;DUART set command reg
OUT DUADD,AL
MOV AL,04H ;set OP2=0
OUT DUDATA,AL
```

Then to disable the chip select:

```
MOV AL,0FH ;DUART reset command reg
OUT DUADD,AL
MOV AL,04H ;set OP2=1
OUT DUDATA,AL
```

Programming Considerations

Programming the Real Time Clock

The RTC resides at relative ports 0Ah and 0Bh. The 0Ah port is used to select the internal register of the clock, and the 0Bh port is the data port. In addition, the RTC uses the IP bit 4 of the DUART, which resides at ports 0Ch (address) and 0Dh (data). The clock is read and set by accessing its internal registers. To read and write the RTC, follow the guidelines given below.

To read the RTC:

1. Write the number of the register to be read to 0Ah.
2. Read 0Bh.
3. Store the results.
4. Read 0Bh again.
5. Compare the data of the two reads.
6. If the data read is the same, the data is valid.
7. If the data read is not the same, read 0Bh again and compare. Data is valid when two readings are the same.

To write the RTC:

1. Select the DUART IP register through 0Ch.
2. Read 0Dh.
3. If IP4 is high, continue. If IP4 is low, read 0Dh again.
4. Write the address of the clock register to 0Ah.
5. Write the data to be written to 0Bh.
6. Reset the 1 MHz clock by writing 0Dh to 0Ah and 00 to 0Bh.

(continued)

Programming Considerations

RTC Internal Registers:

<u>Register</u>	<u>Address</u>
Seconds (1)	0h
Seconds (10)	1h
Minutes (1)	2h
Minutes (10)	3h
Hour (1)	4h
Hour (10)	5h
Day of the Week	6h
Date (1)	7h
Date (10)	8h
Month (1)	9h
Month (10)	Ah
Year (1)	Bh
Year (10)	Ch

A sample program follows:

```
; Display and set the clock on the SS2. This program
; prints out the contents of the clock and then sets
; the clock to January 1, 1986.
;

SS2_BASE EQU 50H ;base address of board
CLOCK_CMD EQU SS2_BASE + 10 ;clock command port
CLOCK_DATA EQU SS2_BASE + 11 ; " data port

DUART_CMD EQU SS2_BASE + 12
DUART_DATA EQU SS2_BASE + 13

DUART_INPUTEQU 13 ;the input register of
; DUART
HOUR10 EQU 6 ;24hr flag in this port

CR EQU 'M'-40H
LF EQU 'J'-40H

EXTRN COUT:NEAR
EXTRN PSTRING:NEAR

CSEG

MOV DX,OFFSET HELLO ;Explain our output
CALL PSTRING
```

Programming Considerations

```
CALL  READTIME
CALL  PRINTCLOCK;print the current time

        MOV   CX,12      ;number of digits to write
        MOV   BX,OFFSET TIMESTR;time to set clock to
        MOV   AL,0       ;start at the beginning
WRITELOOP: MOV   AH,[BX]    ;get the data
        SUB   AH,'0'     ;strip ASCII bias
        CALL  WRITEDIGIT;send out the digit
        INC   BX        ;go to next input digit
        INC   AL        ; and next output digit
        LOOP  WRITELOOP;and repeat

        MOV   DX,OFFSET TIME_RESET
        CALL  PSTRING

        CALL  READTIME
        CALL  PRINTCLOCK

        RETF

; PrintClock : display the time/date from the SS2 clock
;
PRINTCLOCK: MOV   CX,6      ;number of digit pairs
            ; to read
        MOV   AL,0       ;start at the beginning
PRINTLOOP: CALL  PRINTDIGIT;print out the next
        CALL  PRINTDIGIT; pair of digits
        CMP   AL,12     ;print a hyphen between
        JE    NOHYPHEN  ; pairs of digits
        XCHG AH,AL
        MOV   AL,'-'
        CALL  COUT
        XCHG AH,AL
NOHYPHEN:  LOOP  PRINTLOOP
        RET

; Print the clock digit in AL, auto-increment the counter
;
PRINTDIGIT: CALL  READDIGIT
        XCHG AH,AL
        ADD   AL,'0'
        CALL  COUT
        XCHG AH,AL
        INC   AL
        RET
```

Programming Considerations

```
; ReadDigit: AL = digit to read.  
; Returns AH = contents of specified port.  
; Only AX affected  
;  
READDIGIT: PUSH CX ! PUSH BX  
    MOV CL,AL      ;save our digit number  
    MOV BX,OFFSET ADDR_VAR  
    XLAT BX        ;find out where the data  
    MOV BL,AL      ; is saved  
    XOR BH,BH  
    MOV AL,TIME_BUFFER1[BX]  
    AND AL,0FH     ;get rid of high order  
                   ; nibble  
    CMP CL,HOUR10 ;if hour10,  
    JNZ READD_EXIT; bits D2 and D3 must go  
    AND AL,3H      ;mask off the 24hr flags  
READD_EXIT: MOV AH,AL      ;set up our return codes  
    MOV AL,CL  
    POP BX ! POP CX  
    RET  
  
; ReadTime - Read the SS2 clock data twice to insure we  
; do not read during a digit change.  
; Results stored in time_buffer1.  
;  
READTIME: PUSH AX ! PUSH CX ! PUSH SI ! PUSH DI  
READAGAIN: MOV DI,OFFSET TIME_BUFFER1 ;read it once  
            CALL READTSTR  
            MOV DI,OFFSET TIME_BUFFER2 ;read it twice  
            CALL READTSTR  
            MOV CX,12                ;if the 12 bytes  
            MOV SI,OFFSET TIME_BUFFER1 ; are not the  
            MOV DI,OFFSET TIME_BUFFER2 ; same, then  
            REPE CMPSB  
            JNE READAGAIN           ; read it again  
            POP DI ! POP SI ! POP CX ! POP AX  
            RET
```

Programming Considerations

```
; ReadTStr: read the time from the SS2, and save the
; string at es:di. No checking for validity of data
; or masking is done.
;
READTSTR:    MOV AH,0          ;clock port to read
              MOV CX,13        ;number of digits to read
READT_LOOP:   MOV AL,AH        ;tell clock what we want
              OUT CLOCK_CMD,AL
              IN  AL,CLOCK_DATA;get the data from clock
              STOSB           ;save our input
              INC AH           ;point to next digit
LOOPREADT_LOOP ; and repeat
RET

;
; WriteDigit: AH = digit to be written; AL = digit #
; to be written to.
;
WRITEDIGIT:  PUSH CX ! PUSH BX ! PUSH AX
WRITEWAIT:   MOV AL,DUART_INPUT ;make sure the clock
              OUT DUART_CMD,AL ; is not busy
              IN  AL,DUART_DATA
              AND AL,10H         ;strip off to just IP4
              JZ  WRITEWAIT      ;go back to waiting
              POP AX ! PUSH AX
              MOV CX,AX          ;save our digits
              MOV BX,OFFSET ADDR_VAR ;where are we
              XLATBX            ; putting this?
              OUT CLOCK_CMD,AL  ;and ask clock for it
              MOV AH,CH          ;find new digit again
              CMP CL,HOUR10
              JNZ WRITED_DONE
              OR  AH,8H          ;mask in 24-hour flag
WRITED_DONE: MOV  AL,AH
              OUT CLOCK_DATA,AL; and give it to clock
              MOV  AL,ODH         ;point to the reset
              OUT CLOCK_CMD,AL
              OUT CLOCK_DATA,AL
              POP AX ! POP BX ! POP CX
              RET

DSEG

ORG 100H
```

Programming Considerations

```
        DB      'MMDDYYHHMISS='
TIMESTR    DB      '01018600001'

        ; port addresses for:
        ; Month10,Month1Day10,Day1,Year10,Year1
        ; Hour10,Hour1,Mn10,Min1,Sec10,Sec1
ADDR_VAR    DB      10,9,8,7,2,11
                DB      5,4,3,2,1,0

HELLO       DB      CR,LF,'System Support 2 Clock'
                DB      ' Demonstration Version 1.2',CR,LF
                DB      CR,LF,'The time is ',0
TIME_RESET   DB      CR,LF,'Time reset to ',0

TIME_BUFFER1 RS      16
TIME_BUFFER2 RS      16
```

end

Programming the DUART

The dual serial channels are addressed through relative ports 0Ch and 0Dh. To initialize the serial channels, several mode and command words must be written to the DUART. Code initializing the DUART follows. In this example, the DUART is initialized to send characters at 19200 baud with eight bits per character, two stop bits and no parity.

```
;Routine to initialize the serial channels

DUADD EQU 5CH          ;uart address port
DUDATA EQU 5DH          ;uart data port
CSEG
ORG 0
INIT: MOV AL,04H         ;send 80 to the ACR reg
      OUT DUADD,AL        ;o select BAUD
      MOV AL,80H
      OUT DUDATA,AL
;
      MOV AL,0EH           ;set OPO To 0
      OUT DUADD,AL         ;TS/CTS on 0
      MOV AL,01H
      OUT DUDATA,AL
;
```

Programming Considerations

```
MOV     AL,00H      ;Send 13 to mode reg A
OUT    DUADD,AL    ;8 bit, no parity
MOV     AL,13H
OUT    DUDATA,AL

;

MOV     AL,00H      ;Send 0F to mode reg A
OUT    DUADD,AL    ;2 stop bits
MOV     AL,0FH
OUT    DUDATA,AL

;

MOV     AL,01H      ;Send CC to CSRA reg
OUT    DUADD,AL    ;19200 BAUD
MOV     AL,0CCH
OUT    DUDATA,AL

;

MOV     AL,02H      ;Send 15 to command reg A
OUT    DUADD,AL    ;enable transmitter/receiver
MOV     AL,15H
OUT    DUDATA,AL

;

MOV     AL,0EH      ;put OP1 to 0
OUT    DUADD,AL    ;RTS/CTS on bit 1
MOV     AL,02H
OUT    DUDATA,AL

;

MOV     AL,08H      ;Send 13 to mode reg B
OUT    DUADD,AL    ;8 bit, no parity
MOV     AL,13H
OUT    DUDATA,AL

;

MOV     AL,08H      ;Send 0F to mode reg B
OUT    DUADD,AL    ;2 stop bits
MOV     AL,0FH
OUT    DUDATA,AL

;

MOV     AL,09H      ;Send CC to the CSR B reg
OUT    DUADD,AL    ;19200 BAUD
MOV     AL,0CCH
OUT    DUDATA,AL

;

MOV     AL,0AH      ;Send 15 to command reg B
OUT    DUADD,AL    ;enable transmitter/receiver
MOV     AL,15H
OUT    DUDATA,AL
```

On to the rest of the program -

Programming Considerations

Programming the Interval Timers

The 8253 interval timers reside at relative ports 04h-07h. Port 04h is the data port for the counter 0, port 05h is the data port for the counter 1, and port 06h is the data port for the counter 1. Port 07h is the command port for all of the counters.

Control words must be sent to each counter for initialization. In the control words, data bits 6 and 7 are used to select the counter. Data bits 4 and 5 determine in what order the data bytes will be read. Bits 1, 2 and 3 select the mode of operation, and data bit 0 sets the counter as binary or BCD (binary coded decimal).

The code that follows will set up the counters to be square wave generators. The maximum count available to the counters is 65536 with the count going down to zero from the loaded count. The clock is set to 2 MHz.

```
CTRL    EQU  57
CNT0   EQU  54
CNT1   EQU  55
CNT2   EQU  56
;
MOV    AL,3EH ;command word for counter 0
OUT    CTRL   ;square Wave generator
MOV    AL,7EH ;command word for counter 1
OUT    CTRL   ;square Wave generator
MOV    AL,0BEH ;command word for counter 2
;
MOV    AL,0AH ;period of square wave 5 msec
OUT    CNT0,AL ;send LSB to counter 0
OUT    CNT0,AL ;send MSB to counter 0
OUT    CNT1,AL ;send LSB to counter 1
OUT    CNT1,AL ;send MSB to counter 1
OUT    CNT2,AL ;send LSB to counter 2
OUT    CNT2,AL ;send MSB to counter 2
```

on to other programming...

Programming Considerations

Programming the Centronics' Port

The Centronics' port resides at relative ports 08h and 09h. In addition, there are 3 OP bits from the DUART used to send command signals to the printer. The DUART port address is 0Ch and 0Dh. The signals AUTOFD, INIT and SLCTIN are controlled by OP3, OP4 and OP5 respectively. These signals are inverted before they are sent to the printer. Port 09h is the Centronics data port. The status register bits at port 08h are as follows:

DATA BIT SIGNAL

D0	BUSY* - Printer busy when low.
D1	ACK* - Transfer acknowledge low pulse
D2	PE - Paper error when high
D3	ERROR* - Printer error when low
D4	SLCT* - Printer selected when high

Here is a program that initializes the printer and sends a message to it:

```
;*           Centronics Test Program
;*           This program sends a repeating message
;*                   to a printer.
;*
BASE EQU      50H
CESRD EQU     BASE+8H
CEDWR EQU     BASE+9H
DUADD EQU     BASE+0CH
DUDATA EQU    BASE+0DH
CSEG
ORG   00H
MOV   AL,0EH      ;Set up 2681 OP bits
OUT   DUADD,AL    ;set the bits
MOV   AL,38H      ;OP3=0, OP4=0, OP5=0
OUT   DUDATA,AL    ;to send a 0 to AUTOFD and
MOV   AL,0FH      ; INIT and a 1 to SLCT IN
OUT   DUADD,AL    ;set OP3=1, OP4=1, OP5=0
```

Programming Considerations

```
MOV AL,18H
OUT DUDATA,AL
MOV AL,0EH ;Set up 2681 OP bits
OUT DUADD,AL
MOV AL,038H ;OP3=1, OP4=1, OP5=1
OUT DUDATA,AL ;to send a 1 to AUTOFD and
MOV AL,0FH ; INIT and a 0 to SLCT IN
OUT DUADD,AL ;set OP3=0, OP4=0, OP5=1
MOV AL,20H
OUT DUDATA,AL
READ: MOV CX,5
PUSH CX
MOV SI,offset MESSAGE ;Point to message start
CALL PMSG ;Send message to printer
POP CX
LOOP READ ;Do read status, write
MOV SI,offset CRLF ; data again
CALL PMSG
JMPS READ

;
PMSG1: PUSH SI
CALL SENTOUT
POP SI
PMSG: LODSB ;Pick up character
OR AL,AL ;See if end of string
JNZ PMSG1 ;Print char if not end
RET ;of string

CENTOUT:
PUSH AX
CENTW: IN AL,CESRD ;Bring in status from printer
AND AL,1FH ;Mask off upper 3 bits
CMP AL,1BH ;See if printer ready
JNE CENTW ;If not, check again
POP AX
OUT CEDWR,AL ;Send to printer
RET

dseg
ORG 100H
MESSAGE db 'TEST ',0
CRLF DB OAH,ODH,0
```

Programming Considerations

Programming the SCSI Port

The SCSI port resides at relative ports 0Eh and 0Fh. Port 0Eh is the status and command port, and port 0Fh is the data port.

The status register bits at relative port 0Eh are as follows:

Data Bit	Signal
D0	BSY - High when SCSI bus is in use
D4	MSG - High when in Message Phase
D5	C/D* - High in Command Phase
D6	DI/O* - High in Data in Phase
D7	REQ - High when target requests data or commands

In the following example, the SCSI target (a hard disk) is first reset. The SCSI status register is then checked to determine if the bus is free. When the bus is free, a SEL* signal is sent to the target, and the status port is then monitored to determine if the target was selected. The status port is then read to check if the target is ready to receive command words. When ready, the command words are sent. The status port is monitored again to see if the target is ready to receive data. When ready, the data is sent. The data is then read back in a similar manner and compared to the data sent to make sure the transfer was made correctly.

(continued)

Programming Considerations

Sample program:

```
; This is a very simple program that will read and
; write different data every 64K to absolute sector 0.
; There is no initialization and very little error
; checking. This code is intended as a programming
; sample and may not be the most efficient way to
; program the SCSI channel.

SCDATA EQU 5FH ;SCSI data port
SCSC EQU 5EH ;SCSI status/command port
SASIREAD EQU 14H ;SCSI read command value
SASIWRITE EQU 13H ;SCSI write command value
;
TRUE EQU OFFFFH
FALSE EQU NOT TRUE
;
CR EQU 0AH
LF EQU 0DH
EOS EQU '$'

CSEG
ORG 00H
;
;This first section resets the SCSI device
;
INITSC: MOV AL,00H ;Assert rst*
        OUT SCSC,AL ;Send it
        MOV BX,OFFFFH ;Set up delay
        MOV CX,28H
DLOOP: PUSH AX ! POP AX ;Delay for SCSI bus
        PUSH AX ! POP AX
        LOOP DLOOP
        MOV AL,02H ;Disassert rst*
        OUT SCSC,AL ;Send it
        MOV DL,3 ;Set up delay
TDLY: MOV CX,-1
AROUND: PUSH AX ! POP AX
        PUSH AX ! POP AX
        LOOP AROUND
        DEC DL
        JNZ TDLY
        PUSH DS ! POP ES ;Set up buffer
        CLD
        MOV DVAR,0FFFFH
```

Programming Considerations

```
SCSILP: INC      DVAR      ;Bump to next word to test
        MOV      DI,offset TMPBUF;Point to sector buffer
        MOV      AX,DVAR   ;Pick up data to fill buf
        MOV      CX,1024/2
        REP      STOSW    ;Fill buffer with test data

;
;This section calls the main routines of the program
;

        CALL    SELECT     ;Selects the scsi device
        MOV     SI,offset WRCMND ;Point to wt cmd block
        CALL    SENDCMD   ;Sends the write cmd
        MOV     SI,offset TMPBUF
        CALL    DTOUT     ;Send the data
        CALL    STATUS     ;Checks two status words

;
;At this point the data has been sent
;

        MOV     DI,offset TMP2BUF
        XOR    AX,AX
        MOV     CX,1024/2
        REP    STOSW     ;Clear out the sec buffer
        CALL    SELECT     ;Reselect the scsi device
        MOV     SI,offset RDCMND ;Point to rd cmd block
        CALL    SENDCMD   ;Send read command
        CLD
        MOV     DI,offset TMP2BUF ;Set up data area
        CALL    DTIN      ;Bring data to storage
        CALL    STATUS     ;Clear the status bytes

;
;The data is now in storage
        MOV     SI,offset TMPBUF ;Point to both buffer
        MOV     DI,offset TMP2BUF ; for string compare
        MOV     CX,1024/2
        REPE   CMPSW     ;Compare the two buffers
:jump to error routine if sectors did not match
        JNE    ERROR
        MOV     DX,offset OKMSG ;Print msg to show
        CALL   PRINT     ;sector read/wrote ok
        JMPS  SCSILP    ;jump back to do rd/wt
                    ;      over

;
ERROR:  MOV     DX,offset ERRMSG
        CALL   PRINT
        MOV     CX,0       ;Exit to CP/M
        MOV     DX,0
        INT    224      ;Go back to cpm
        RETF   ;Should never return
```

Programming Considerations

```
;  
; GENERAL SCSI utility routines  
;  
;This routine checks to see if the SCSI device is not  
;busy, sends SEL*, sends id code, and disasserts SEL*  
;  
SELECT:    CLD  
BSYHI:     IN    AL,SCSC ;Check is busy=0 on board  
           AND   AL,01H  ;Mask off all but d0  
           CMP   AL,00H  ;See if it's 0  
           JNZ   BSYHI  ;If it's not, try again  
           MOV   AL,01H  ;Assert sel*  
           OUT   SCDATA,AL;Send it  
           MOV   AL,03H  ;Send device id  
           OUT   SCSC,AL  
BSYLO:     IN    AL,SCSC ;See if busy=1 on board  
           AND   AL,01H  ;Mask  
           CMP   AL,01  ;See if it's 1  
           JNZ   BSYLO  ;If not, try again  
           MOV   AL,02H  ;Disassert sel*  
           OUT   SCSC,AL  
RET  
;  
;  
;This routine sends a command to the SCSI device  
; SI points to command to send  
;  
SENDCMD:   CALL  WAITRQ ;See if req* is asserted  
           IN    AL,SCSC  
           AND   AL,60H  
           CMP   AL,20H  ;See if still in cmd phase  
           JNZ   DONE    ;Quit if not in cmd phase  
           LODSB       ;Pick up cmd byte to send  
           OUT   SCDATA,AL;Send cmd byte to SCSI  
           JMP$  SENDCMD  
DONE:      RET
```

Programming Considerations

```
;  
;This routine sends the data to the SCSI port  
; SI points to data to send  
;  
DTOUT:      MOV    DX,SCDATA  
DTOUT1:     CALL   WAITRQ  
             IN     AL,SCSC      ;See if still in data  
             AND   AL,20H       ;    phase  
             CMP   AL,00H  
             JNZ   DTOUTDONE   ;If in data phase, send  
                         LODSB      ;    more  
             OUT   DX,AL  
             JMPS  DTOUT1  
DTOUTDONE:  RET           ;If not stop sending  
  
;  
;  
;This routine brings the data from the device and  
;stores it in memory  
; DI points to destination buffer  
;  
DTIN:       MOV    DX,SCDATA  
DTIN1:      CALL   WAITRQ  
             IN     AL,SCSC      ;Still in data phase ?  
             AND   AL,20H  
             CMP   AL,20H  
             JZ    DTINDONE  
             IN     AL,DX  
             STOSB  
             JMPS  DTIN1  
DTINDONE:   RET           ;No more data, go back  
;  
  
;This routine brings in the two status words from the  
;device. The words are disregarded. An error routine  
;could be added here.  
;  
STATUS:     IN     AL,SCDATA  
             MOV   CL,AL      ;Store in cl  
             CALL  WAITRQ  
             IN     AL,SCDATA  
             RET
```

Programming Considerations

```
;  
;This is the routine that waits for the REQ* to be  
;asserted.  
;  
WAITRQ:    IN     AL,SCSC  
            AND    AL,80H  
            CMP    AL,80H  
            JNZ    WAITRQ  
            RET  
  
;  
; General system utility routines  
;  
; Print a message on console  
; DX points to message to print  
;  
PRINT:     MOV    CL,9  
            INT    224  
            RET  
;  
          DSEG  
          ORG    100H  
DVAR      DW     0000  
ERRMSG    DB     CR,LF,'ERROR: SECTOR DID  
           ' NOT VERIFY',CR,LF,EOS  
OKMSG     DB     ' SCSI ',EOS  
RDCMND    DB     SASIREAD,0,0,0,0,0  
WRCMND    DB     SASIWRITE,0,0,0,0,0  
TMPBUF    RB     1024  
TMP2BUF   RB     1024  
           DB     0  
           END
```

Appendix A

Specifications

Size:	
Length	253 millimeters (10 inches)
Depth	13 millimeters (0.5 inches)
Height	127 millimeters (5 inches)
Weight	371 grams (13 ounces)
Edge Connectors	34 Pin Shrouded Right Angle 26 Pin Shrouded Right Angle 50 Pin Shrouded Right Angle
Timing	Meets all IEEE 696/S-100 specifications. Meets CompuPro two cycle system timing at clock rates up to 12.5 MHz.
Processors	Compatible with most CompuPro supported CPU boards.*
S-100 Address Space	Occupies 256K byte memory space and 16 I/O ports
S-100 Memory Address	Switch selectable to any 256K byte page
Standard I/O Address 0050h to 005Fh	
Power Consumption	+8V at 1000 mA typical, 1900 mA maximum; +/-16V at 120 mA maximum

* NOTE: The System Support 2 will not support interrupts on the CPU-ZTM and CPU 32016TM.

Appendix B

Technical Data Sources

Here is a list of sources for more detailed technical information than that included in this manual:

Centronics Specifications

Centronics Data Computer Corporation
1 Wall Street
Hudson, New Hampshire 03051

Telephone: (603) 883-0111

DUART SCN 2681

Signetics Corporation
811 East Arques Avenue
P.O. Box 409
Sunnyvale, California 94086

Telephone: (408) 991-2000

Programmable Interrupt Controller 8259A Programmable Interval Timer 8253

Intel Literature Department
3065 Bowers Avenue
Santa Clara, California 95051

Telephone: (800) 538-1876
California: 800 672-1833

(continued)

Appendix B

RS-232C Standard

Electronic Industries Association
Engineering Department
2001 I Street N.W.
Washington D.C. 20006

Real Time Clock - RTC 58321

Epson America, Inc.
23600 Telo Street
Torrance, California 90505

Telephone: (213) 373-9511

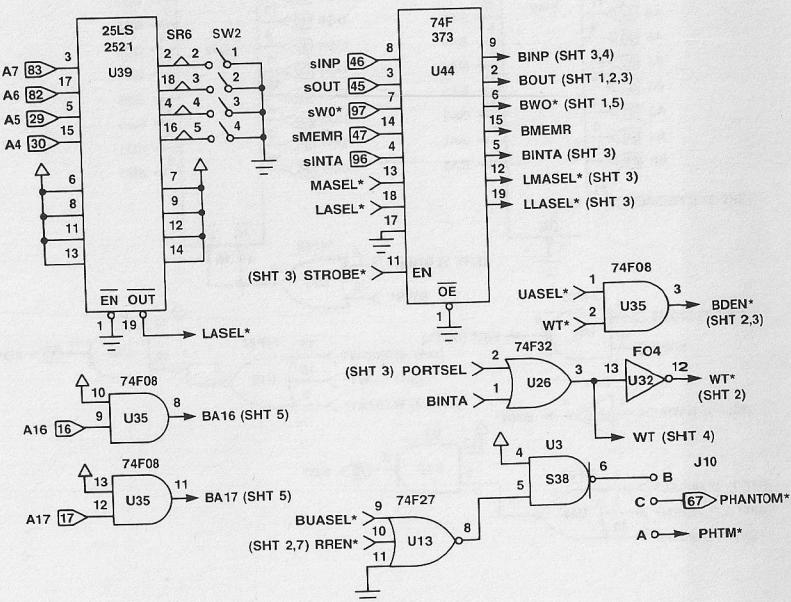
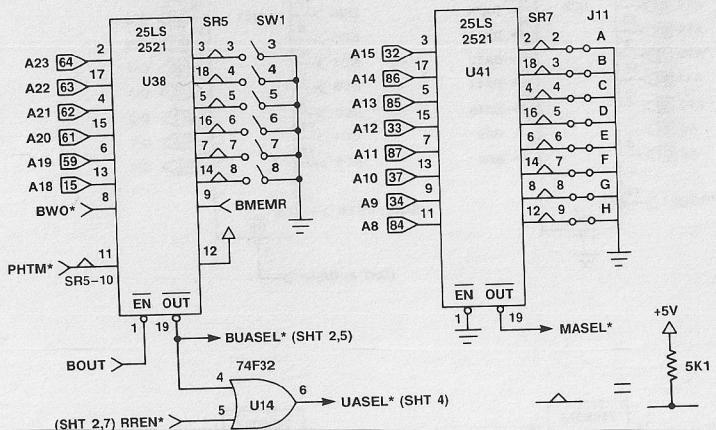
SCSI Standard

U.S. Department of Commerce
National Bureau of Standards/Technology 4-216
Washington D.C. 20234

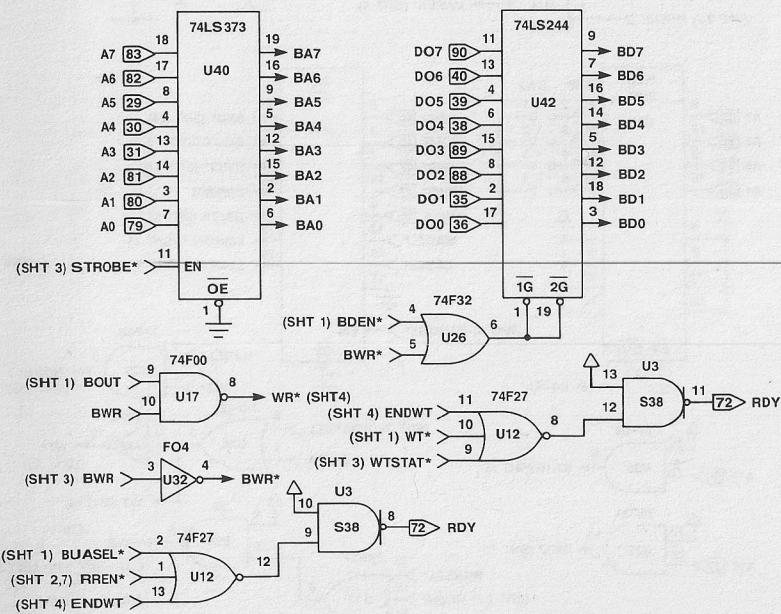
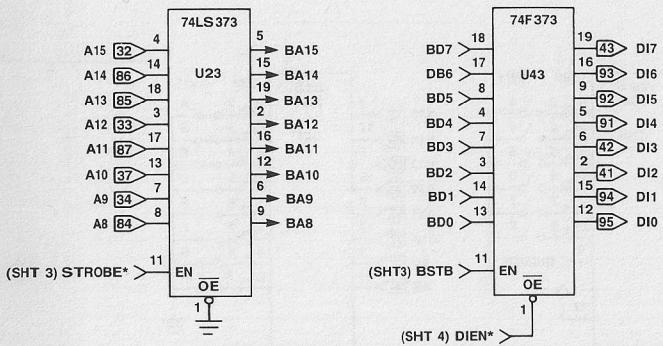
Telephone: (301) 921-3723

Appendix C

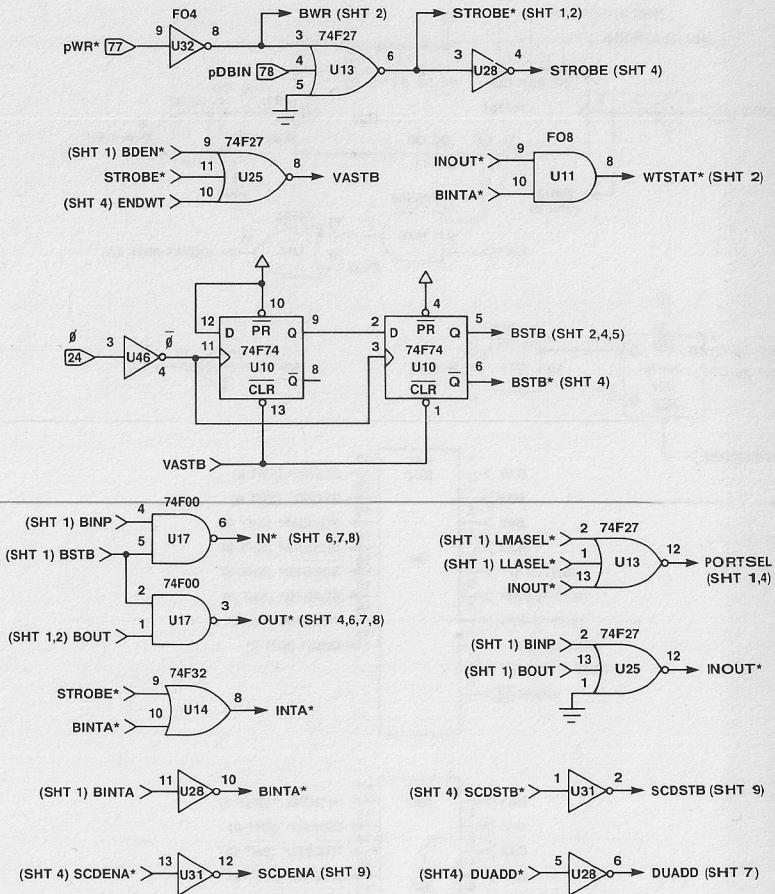
Schematic Diagrams



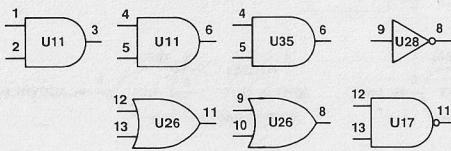
Appendix C



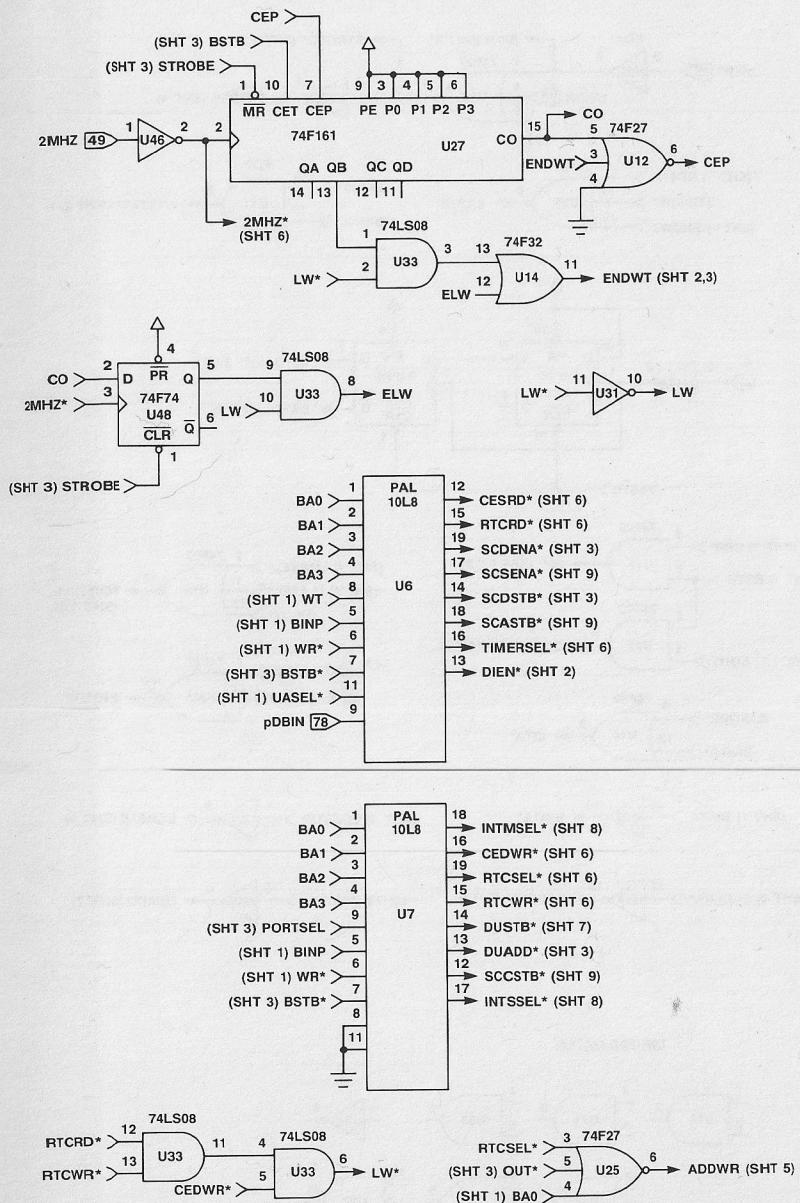
Appendix C



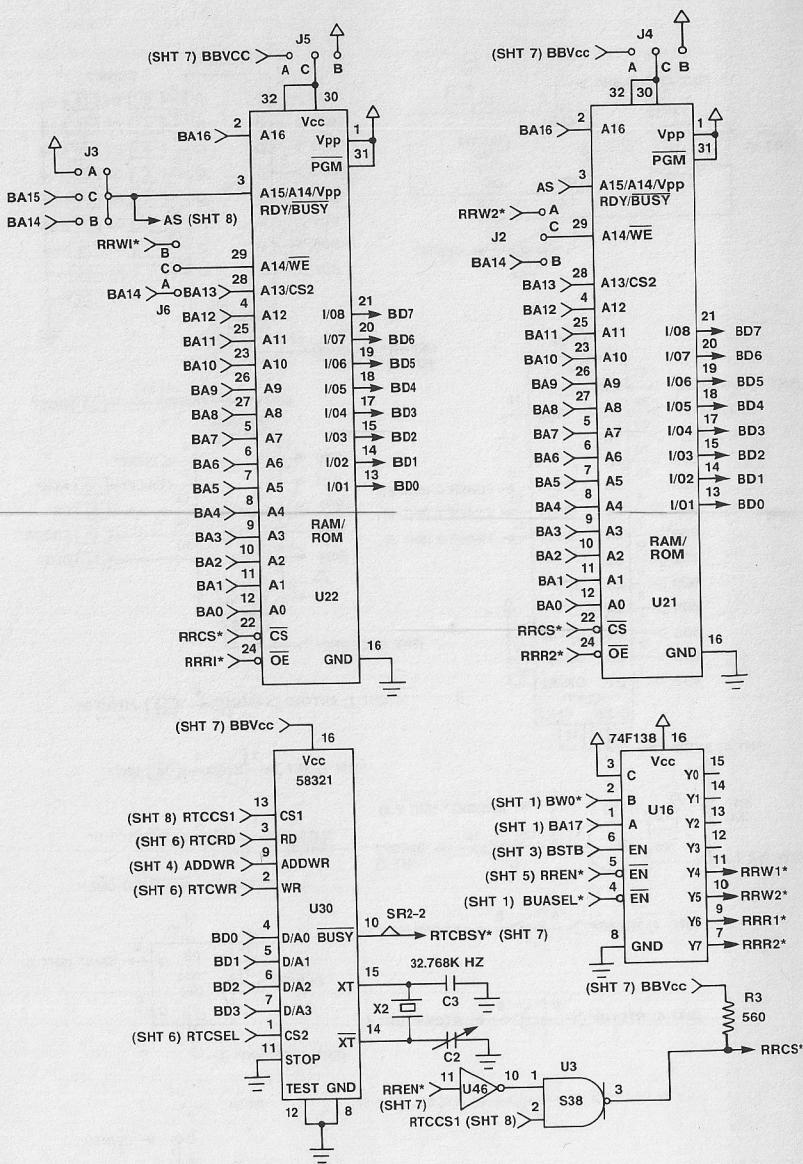
UNUSED GATES



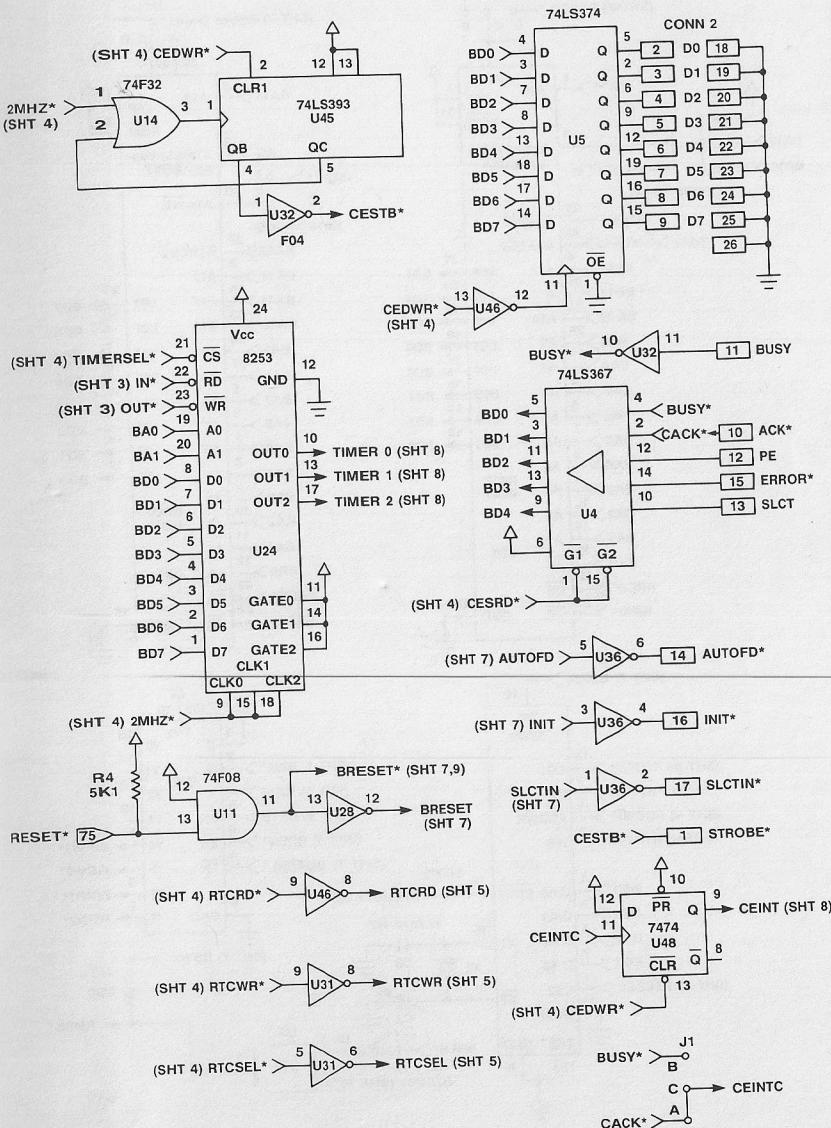
Appendix C



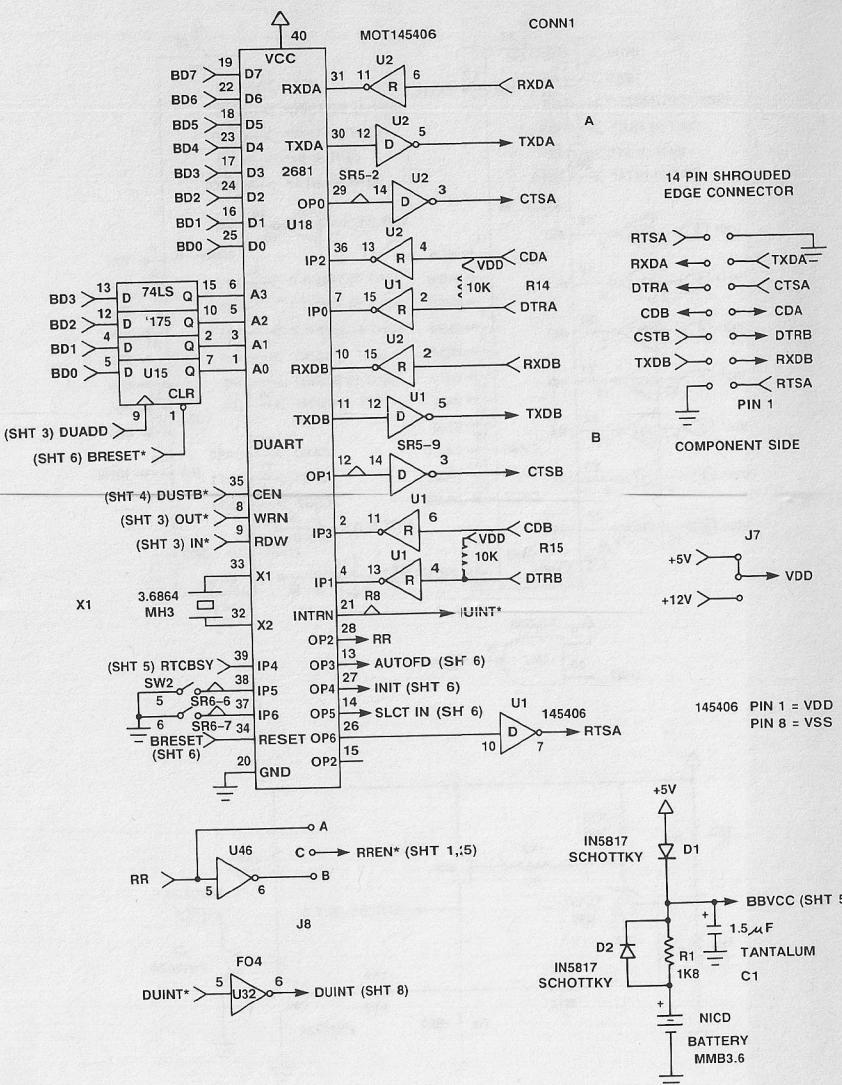
Appendix C



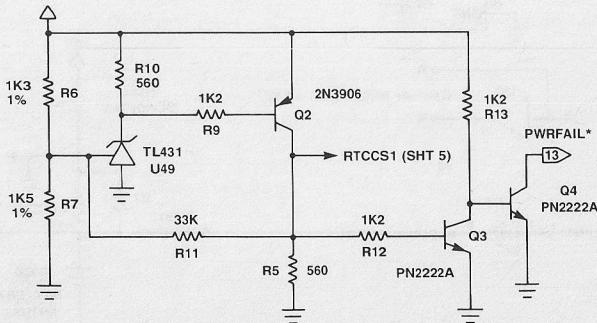
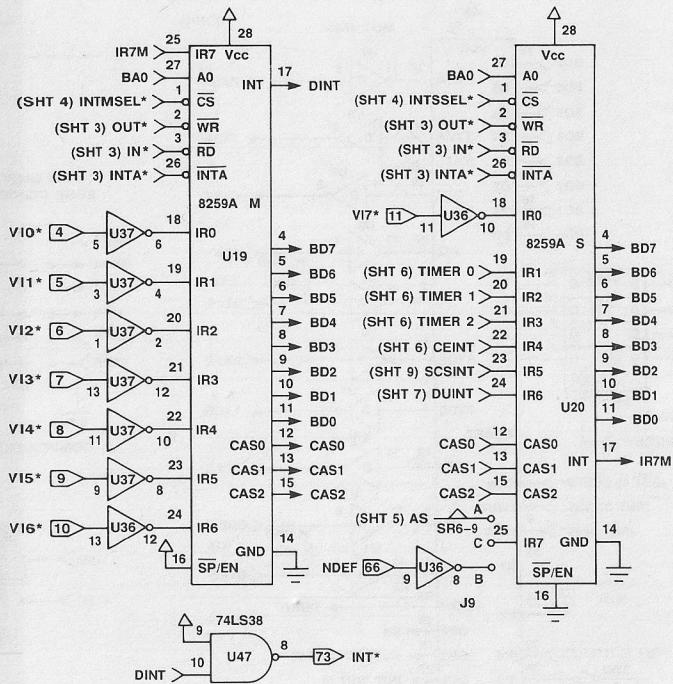
Appendix C



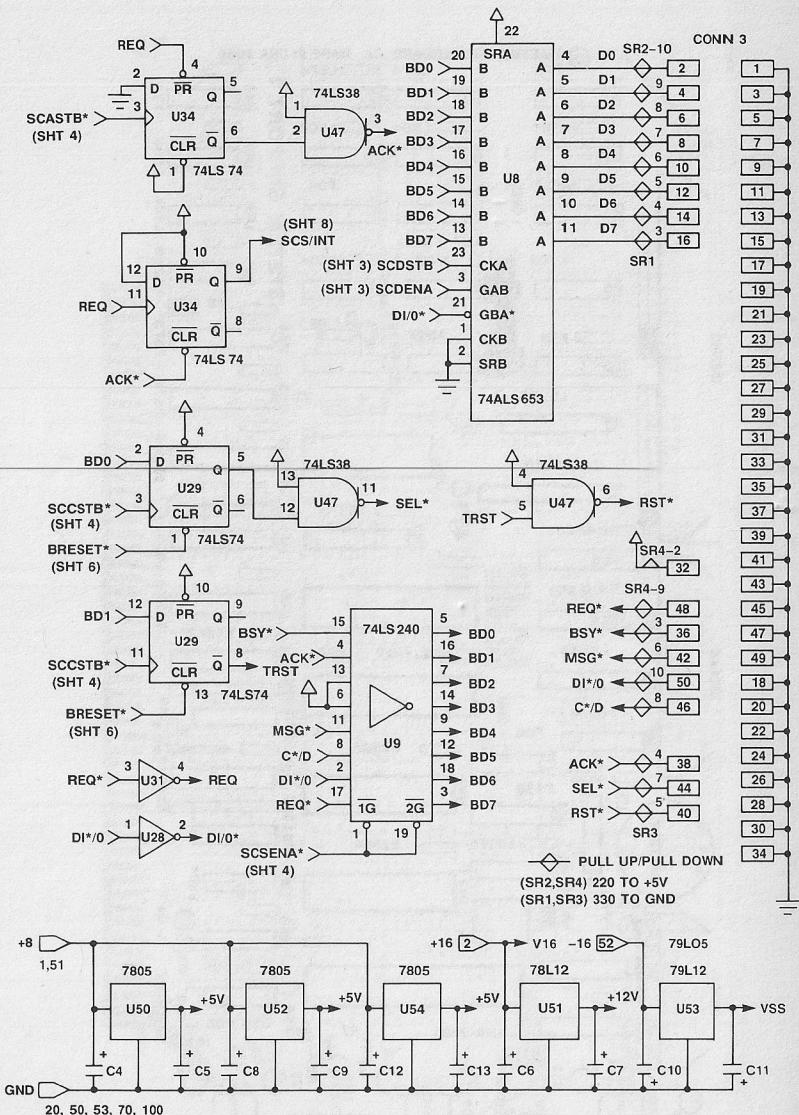
Appendix C



Appendix C

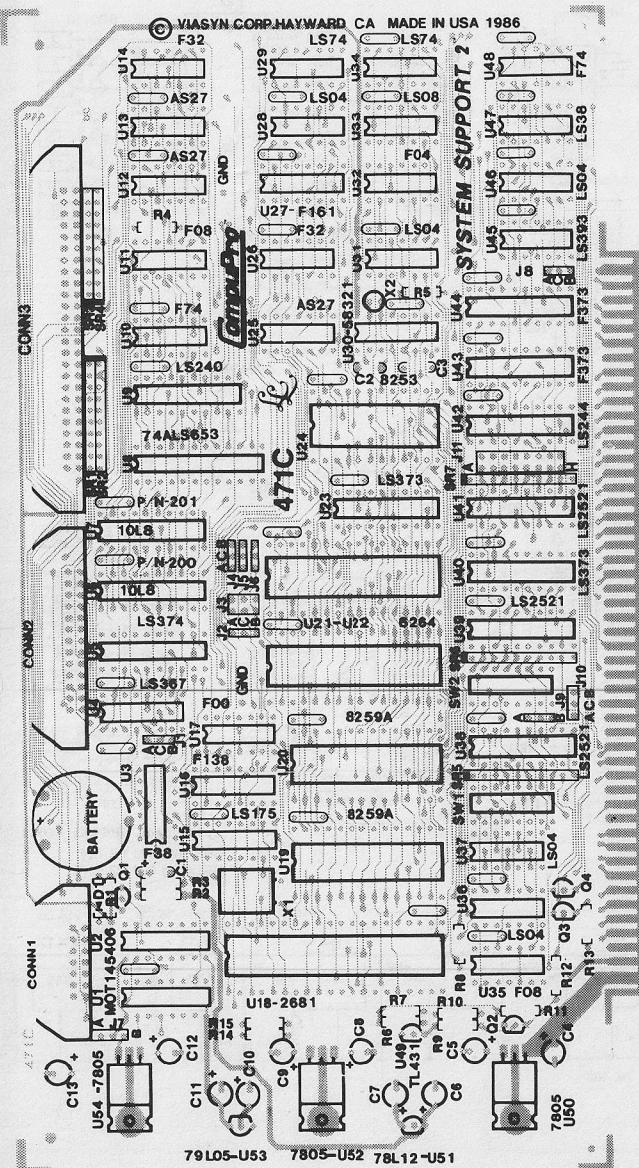


Appendix C



Appendix D

Component Layout



Viasyn Corporation

26538 Danti Court, Hayward, CA 94545-3999 (415) 786-0909

\$30 .00

8261-00848