VL-7842

Multifunction Z80 Processor Card
for the STD Bus

Model VL-7842a/b/c/d

Multifunction Z80 Processor Card for the STD Bus

REFERENCE MANUAL

VL-7842a Rev. 0.00
VL-7842b Rev. 0.00
VL-7842c Rev. 0.00
VL-7842d Rev. 0.00

Doc. Rev. 01/04/95

VersaLogic Corporation
3888 Stewart Rd. • Eugene, OR 97402

(503) 485-8575
Fax (503) 485-5712

NOTICE:

Although every effort has been made to ensure this documentation is error-free, VersaLogic makes no representations or warranties with respect to this product and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

VersaLogic reserves the right to revise this product and associated documentation at any time without obligation to notify anyone of such changes.

M7842

Model VL-7842
Multifunction Z80 Processor Card for the STD BUS

REFERENCE MANUAL

CONTENTS

1. Introduction

2. Installation and Configuration

3. Operation

4. Reference

VersaLogic Corporation
3888 Stewart Rd.
Eugene, OR 97402

(503) 485-8575

Section 1
INTRODUCTION

INTRODUCTION

This manual details the installation and operation of VersaLogic's VL-7842 Multifunction Z80 processor card. This card includes a number of important system functions including on-board memory expansion and RS-422 I/O channels. These features simplify system construction, and lower the total system cost.

The VL-7842 card is part of the full line of STD BUS products which are available from VersaLogic. Contact VersaLogic for further information on the compatible STD BUS expansion cards that are currently available for use with the VL-7842 processor cards.

OVERVIEW

The VL-7842 microprocessor card features a Z80 processor, four counter/timer channels, four 28 pin memory sockets, and two high speed communications ports on a single STD BUS card.

It includes flexible memory chip type selection that allows a mixture of on-board memory devices to be accommodated. The on-board sockets can accept 2 to 32K RAMs, 2 to 32K PROMs, EEPROMs, and nonvolatile (Zero Power) RAMs. In addition, the memory expansion line (MEMEX) can be controlled by the board.

The on-board I/O channels provide RS-232, RS-422, and RS-485 multidrop capabilities. The RS-422 interface is ideal for longer distance (4000 feet) communications and communications within electrically noisy environments. Both channels allow high speed (to 750K baud) prioritized, interrupt driven operation.

Normally one or two of the on-board timer channels are used for baud rate generation. The remaining two or three can be used as general purpose timers, external event counters, or as prioritizing interrupt inputs.

The VL-7842 is available in several different speed versions. Currently clock speeds of 2.5 MHz, 3.6864 MHz, 4 MHz, and 6 MHz are available. The card's clock speed affects the overall thruput of the system (processing speed), as well as the communications speed (baud rates) at which the card may be operated.

Section 2
INSTALLATION AND CONFIGURATION

HANDLING

** CAUTION ** The VL-7800 cards use chips which are sensitive to static electricity discharges. Normal precautions, such as discharging yourself, work stations, and tools to ground before touching the board should be taken whenever the board is handled.

The board should also be protected during shipment or storage by placing it in a conductive bag (such as the one it was received in) or by wrapping it in metal foil.

INSTALLATION

The VL-7800 cards can be used alone, as single card microcomputers, or combined with other STD BUS I/O cards in an STD BUS system. When they are used with other STD BUS cards they can usually be inserted into any slot of an STD BUS card cage.

If the interrupt priority chain feature is used (supported by the VL-7842), cards must be inserted into the card cage according to their relative priority (usually left to right). See the Interrupt Priority Chain section for further information.

** CAUTION ** When cards are installed in an STD BUS card cage they must be oriented correctly (usually with the card ejector toward the top of the cage). Refer to the card cage documentation for the correct way to insert the STD BUS cards.

** CAUTION ** Cards should be inserted or removed from the STD BUS card cage only when the system power is off.

OPTION JUMPERS

Various options available on the VL-7842 card are selected using removable jumper plugs (shorting plugs). Features are selected or de-selected by installing or removing the jumper plugs as noted. The terms "IN" or "JUMPED" are used to indicate an installed plug. "OUT" or "OPEN" indicates the absence of a jumper plug.

Figure 2-1 shows the jumper block locations on the VL-7842 board. It also indicates the position of the jumper plugs when the board was shipped from the factory. The function of each jumper block is detailed in Figure 2-2.

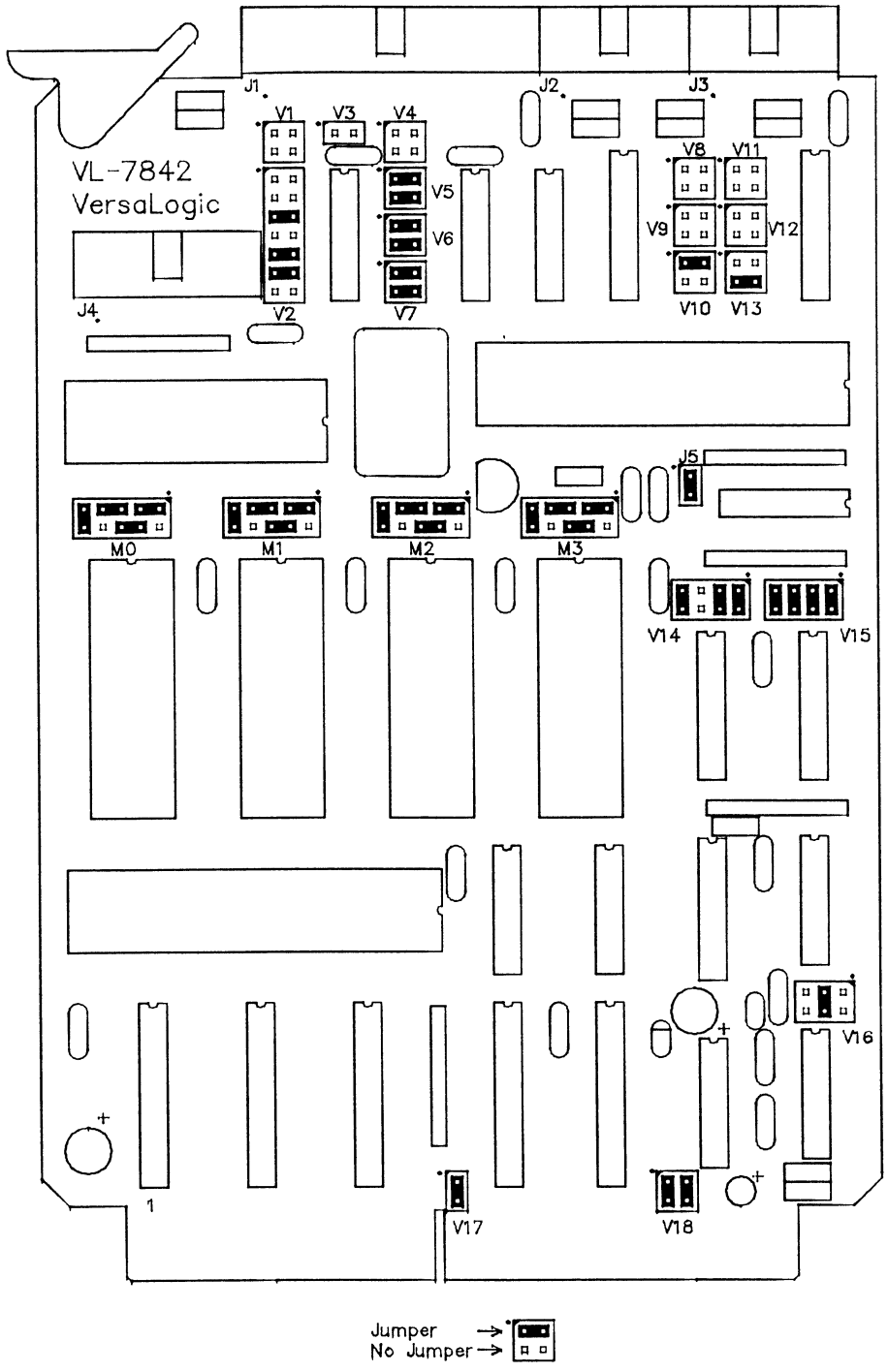


Figure 2-1. Jumper Block Locations

Jumper Block	Description	As Shipped
M0-M3	Memory socket type configuration.	4-8K RAMs/ROMs
J5	Memory map select/ Segment signal connector.	IN
V1	a - SIO chan. A ready/wait line to WAITRQ*. b - SIO chan. B ready/wait line to WAITRQ*.	out out
V2	Baud rate clock options. See Baud Rate Clock. a - Chan. B RECV clock from RS-422 line. b - Chan. B RECV clock from chan. B XMIT. c - Chan. B XMIT clock from CTC chan. 1. d - Chan. B XMIT clock from CTC chan. 0. e - SYSCLK/2 to CTC chan. 1 input. f - Chan. A RECV clock from CTC chan. 0. g - Chan. A RECV clock from extrnl. line.	 a - out b - out c - IN c - out d - IN e - IN g - out
V3-V7	Serial I/O port options. See Serial I/O Ports.	
V8-V9	Channel A multidrop mode	out
V10	a - Channel A RS-232 input. b - Channel A RS-422 input.	a - IN b - out
V11-V12	Channel B multidrop mode	out
V13	a - RS-232 received clock to jumper V2g. b - RS-422 received clock to jumper V2g.	a - out b - IN
V14	Memory map selection. See Memory Map.	Map #0
V15	Memory socket enable/disable. a - Socket M3 enabled. b - Socket M2 enabled. c - Socket M1 enabled. d - Socket M0 enabled.	 a - IN b - IN c - IN d - IN
V16	a - MEMEX is set high at power-up.[1] b - MEMEX is set low at power-up. c - WAIT state enabled.	a - out b - IN c - out[2]
V17	a - MEMEX signal controlled on-board.	a - IN
V18	a - IOEXP connected to ground. b - AUX GND connected to digital ground.	a - IN b - IN

Notes:
1) Also inverts data written to the MEMEX control port.
2) IN on 6 MHz versions.

Figure 2-2. Jumper Functions

EXTERNAL CONNECTIONS

Connections to the RS-232 and RS-422 connectors can be made as noted below. Pinout information for all of the VL-7842 connectors is shown in the following figures.

VL-7842 Connector	Mating Connector or Cable Assembly
J1	VersaLogic #9560 + 9553 (1' DB-25S + 10' RS-232 cable) or VersaLogic # 9560 + 9554 (1' DB-25S + 25' RS-232 cable)
J2 & J3	3M connector #3473-7010 or equiv.
J4	AMP #530554-1 + (2) #530553-5 or equiv. two pin connector or 3M #3473-7014 or equiv. 14 pin female cable connector
J5	AMP #530554-1 + (2) #530553-5 or equiv. two pin connector

Figure 2-3. I/O Connections

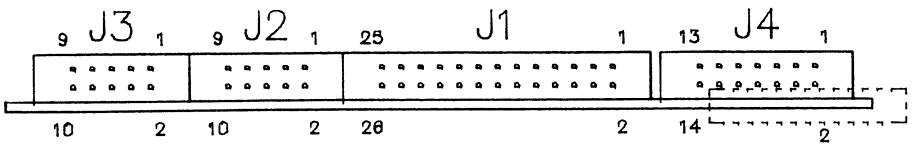


Figure 2-4. I/O Connector Physical Pin Locations

J1 Pin	Signal Name	RS-232-C Pin
1	-	1
2	-	14
3	TD	2
4	(RCLK)	15
5	RD	3
6	-	16
7	RTS	4
8	RCLK	17
9	CTS	5
10	-	18
11	DSR	6
12	-	19
13	GND	7
14	DTR	20
15	DCD	8
16	-	21
17	-	9
18	-	22
19	-	10
20	-	23
21	-	11
22	TCLK	24
23	-	12
24	-	25
25	-	13
26	-	-

Note: The J1 connector can be converted to the RS-232-C pinout using VersaLogic cable #9560.

Figure 2-5. RS-232 Serial Port Connector (J1) Pinout

J2 / J3 Pin	Signal Name	Direction
1	TXC+	OUT
2	TXC-	OUT
3	TXD+	OUT
4	TXD-	OUT
5	SGND	-
6	SGND	-
7	RXD-	IN
8	RXD+	IN
9	RXC-	IN
10	RXC+	IN

Figure 2-6. RS-422 Serial Port Connector (J2 and J3) Pinout

J4 Pin	Signal Name	Input Load (Sink ma)	Output Drive (Sink ma)
1	GND		
2	CLOCK/INT 3	1	
3	GND		
4	CLOCK/INT 2	1	
5	GND		
6	COUNT/TIME 2		2
7	GND		
8	CLOCK/INT 1	1	
9	GND		
10	COUNT/TIME 1		2
11	GND		
12	WAIT/READY* A	0.1	2
13	GND		
14	WAIT/READY* B	0.1	2

Figure 2-7. Counter/Timer Connector J4

J5 Pin	Signal Name	Input Load (Sink ma)
1	GND	
2	SEGMENT	1.3

Figure 2-8. Segment Connector / Jumper J5

MEMORY

The VL-7800 cards have four on-board memory sockets. These sockets can be individually programmed (jumpered) to accept a variety of memory devices.

Installation of the on-board memory chips is accomplished in two separate steps. First, a memory map is selected that will allow each socket to be addressed at the desired memory location. Second, the jumpers for each memory socket are set to accommodate the type of chip that will be used in the socket.

Memory Map

The desired memory map is selected using jumper V14. There are 7 preprogrammed options included in the memory decoder PROM. In addition, there are 9 blank (unprogrammed) options which may be programmed by the user for special applications.

Note that the memory map option only determines the memory space that is reserved for the on-board memory sockets. Whether or not the space is used by each socket, and the type of RAM or RAM which can be plugged into each socket, is determined by the memory socket jumpers (discussed later in this section).

Memory Map Selection

The desired memory map is selected using jumper plug V14 and J5. J5 can also be used as a connector to an external "Segment select" signal for special applications.

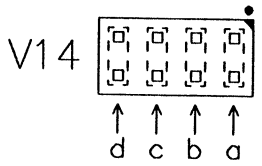
Both the MEMEX (memory expansion) and Segment signals can be used to control the currently selected memory map. However, most applications will not benefit from the use of these signals and normally they are not used.

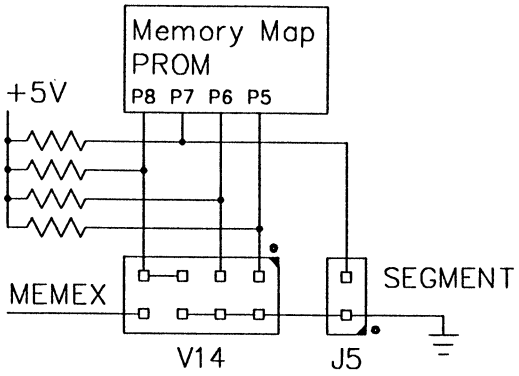
The following chart lists the standard jumper configurations used to select any of the 15 memory map options. These standard configurations ignore the state of the MEMEX and Segment signals.

Map#	Socket 0	Socket 1	Socket 2	Socket 3
0	4K 0000-0FFF	4K 1000-1FFF	4K 2000-2FFF	2K 3000-37FF
1	8K 0000-1FFF	8K 2000-3FFF	8K 4000-5FFF	8K 6000-7FFF
2	16K 0000-3FFF	16K 4000-7FFF	8K 8000-9FFF	8K A000-BFFF
3	32K 0000-7FFF	8K 8000-9FFF	8K A000-BFFF	8K C000-DFFF
4	4K 0000-0FFF	4K 1000-1FFF	4K 2000-2FFF	4K 3000-3FFF
5	16K 0000-3FFF	12K 4000-6FFF	2K 7000-77FF	2K 7800-7FFF
6	2K 0000-07FF	2K 0800-0FFF	2K 1000-17FF	2K 1800-1FFF
7*	32K 0000-7FFF	32K 8000-FFFF	-	-
8*	16K 0000-3FFF	16K 4000-7FFF	16K 8000-BFFF	16K C000-FFFF
9*	16K 0000-3FFF	32K 4000-BFFF	8K C000-DFFF	2K E000-E7FF
10*	16K 0000-3FFF	8K 4000-5FFF	8K 6000-7FFF	8K 8000-9FFF
11*	16K 0000-3FFF	32K 4000-BFFF	8K C000-DFFF	8K E000-FFFF
12*	16K 0000-3FFF	16K 4000-7FFF	32K 8000-FFFF	-
13	unprogrammed (all sockets selected)			
14	unprogrammed (all sockets selected)			
15*	32K 0000-7FFF	-	32K 8000-FFFF	-

* Available only with memory map ROM #I512X4 Vers. F.

Figure 2-9. Memory Map Options





J4	V14 d	V14 c	V14 b	V14 a	Map#
IN	-	IN	IN	IN	0
IN	-	IN	IN	-	1
IN	-	IN	-	IN	2
IN	-	IN	-	-	3
-	-	IN	IN	IN	4
-	-	IN	IN	-	5
-	-	IN	-	IN	6
-	-	IN	-	-	7
IN	-	-	IN	IN	8
IN	-	-	IN	-	9
IN	-	-	-	IN	10
IN	-	-	-	-	11
-	-	-	IN	IN	12
-	-	-	IN	-	13
-	-	-	-	IN	14
-	-	-	-	-	15

Standard Selection Method

P8	P7	P6	P5	Map#
LO	LO	LO	LO	0
LO	LO	LO	HI	1
LO	LO	HI	LO	2
LO	LO	HI	HI	3
LO	HI	LO	LO	4
LO	HI	LO	HI	5
LO	HI	HI	LO	6
LO	HI	HI	HI	7
HI	LO	LO	LO	8
HI	LO	LO	HI	9
HI	LO	HI	LO	10
HI	LO	HI	HI	11
HI	HI	LO	LO	12
HI	HI	LO	HI	13
HI	HI	HI	LO	14
HI	HI	HI	HI	15

Alternate Selection Method

Figure 2-10. Memory Map Selection

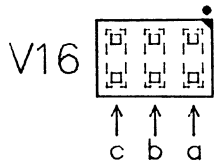
MEMEX Signal

The MEMEX (memory expansion) signal on the STD BUS is normally used to select between two different 64K byte memory banks or maps. It can be used to expand the available memory, or to control a bootstrap PROM.

The MEMEX signal can be controlled either on-board (by the VL-7842 card) or by another card on the bus. The MEMEX signal can also be set to a high (1) or low (0) state during system power-on. Jumpers V16 and V17 control these functions as shown in Figure 2-11.

The standard memory map selection jumpers (Figure 2-9) ignore the state of the MEMEX signal. These memory map selections may be used whether or not the MEMEX signal will be used by other cards in the system.

Alternately, the on-board memory map can change according to the state of the MEMEX signal. The jumper settings, and the resulting memory maps, can be determined using Figure 2-10.



Jumper Block	Description	As Shipped
V16	a - MEMEX is set high at power-up.[1] b - MEMEX is set low at power-up.	a - open b - IN
V17	IN - MEMEX signal controlled on-board. OUT - MEMEX signal controlled externally.	IN

[1] Also inverts data written to MEMEX control port.

Figure 2-11. MEMEX Options

Segment Signal

The Segment signal is similar to MEMEX except that it is not carried by the STD BUS. This signal is generated, usually by an I/O card in the system, and externally connected to the memory (or other) cards to be controlled. The Segment signal is included on the VL-7842 card only for compatibility with older designs. Its use is not recommended.

To use a Segment signal with the VL-7842 the control cable is connected to J5. The memory maps selected, depending on the state of the Segment input, can be determined using Figure 2-10.

Memory Socket Configuration

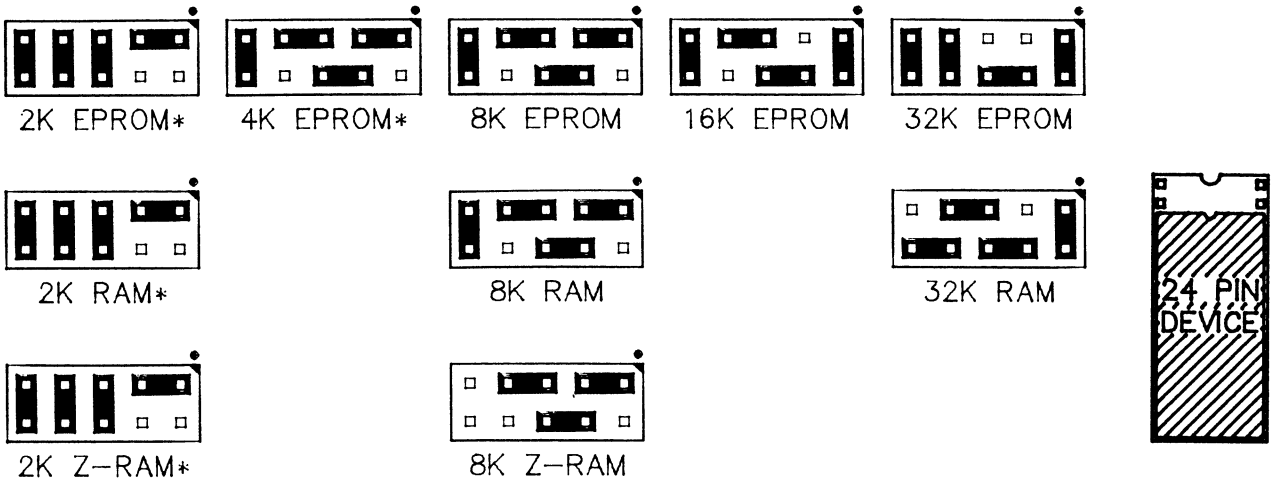
Once a memory map has been selected, each socket must be configured for the type of device that will be used in the socket. The sockets are individually configured using jumper blocks M0-M3 which are located directly above the sockets.

Refer to Figure 2-12 for selection of the appropriate jumpers for each memory device type. Jumpering is shown for EPROMs, RAMs, and Z-RAMs (Zero Power RAMs). The sockets can also accomodate standard ROMs (use EPROM jumpering) and 5V type EEPROMs (contact VersaLogic for jumper configurations).

Note that both 24 and 28 pin devices are plugged into the 28 pin sockets. Care must be taken to locate 24 pin devices at the bottom of the 28 pin socket.

Disabling Unused Sockets

Any unused socket(s) may be disabled, freeing its space in the memory map for off-board memory. Memory sockets are disabled by removing jumpers from the V15 jumper block.



* 24 pin device. Locate in the 28 pin socket as shown.

Figure 2-12. Memory Socket Jumper Configuration

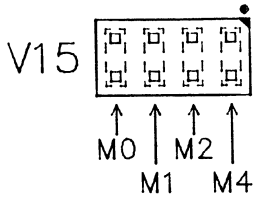


Figure 2-13. Memory Socket Enable Jumpers

I/O MAPPING

Mapping of the on-board I/O devices is controlled by a decoder PROM in location U9. The standard I/O map is shown below.

I/O Address	Port Type	Description
F0-F3	I/O	CTC Channel 0-3
F4-F5	I/O	SIO Channel A
F6-F7	I/O	SIO Channel B
FE	OUT	MEMEX Control

Figure 2-14. I/O Port Mapping

The remaining I/O addresses, 00-EF and F8-FD, are available for use by off-board I/O devices.

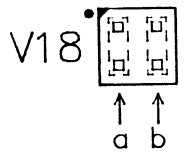
For special applications the I/O map can be altered as desired with a user programmed decoder PROM (contact VersaLogic for detailed programming information).

IOEXP Signal

The IOEXP (I/O expansion) signal on the STD BUS can be used to select between two different 256 port I/O banks or maps. It is used to expand the number of I/O ports that can be accessed by the processor.

The VL-7842 card does not control or decode the IOEXP signal. Since many I/O cards require this signal to be low on the bus, the IOEXP signal is connected to ground on the VL-7842.

For special applications the IOEXP signal can be disconnected from ground, allowing it to be controlled by another card in the system. To "float" the IOEXP signal, remove the jumper from position V18a.



Jumper Block	Description	As Shipped
V18	a - IOEXP connected to ground. b - AUX GND connected to digital ground.	a - IN b - IN

Figure 2-15. IOEXP and AUX GND Jumper Options

INTERRUPT PRIORITY CHAIN

The VL-7842 supports the STD BUS interrupt priority chain signal. This signal prioritizes system interrupts, and makes sure that only one interrupt source sends a vector address to the CPU in interrupt Mode 2. If more than one card in the system will be interrupting with Mode 2 interrupts, the priority chain must be used.

The priority chain establishes interrupt service priority based upon a card's physical position in the card cage. Cards of the highest priority should be placed in the right-most card cage slot. Note: VersaLogic and Pro-Log card cages establish priority from right to left; other card cage manufacturers may connect the priority signal from left to right.

Since the priority chain signal is passed from one slot to the next, ALL THE PRIORITIZED CARDS MUST BE IN ADJACENT SLOTS. Empty slots between cards will break the chain. Cards which do not use the priority chain (i.e. that don't use interrupts) can be placed in any of the remaining bus slots without concern for empty card slots.

SERIAL I/O PORTS

The VL-7842 card includes two high speed serial I/O channels. Both channels can be used for synchronous or asynchronous transmissions. Channel A may be jumpered for either RS-232 or RS-422 operation. Channel B includes an RS-422 interface only. If the RS-232 interface is not used, the VL-7842 card can be operated with a 5 volt supply only (±12 volt supply is not required).

The interface used with channel A is selected with jumpers V10 and V13 as shown in Figure 2-16. Note that the setting of V13 does not matter unless an external receive clock line is used (synchronous mode).

Jumper	Description	As Shipped
V10	a - Channel A RS-232 input. b - Channel A RS-422 input.	a - IN b - out
V13	a - RS-232 received clock to jumper V2g. b - RS-422 received clock to jumper V2g.	a - out b - IN

Figure 2-16. Channel A RS-232 / RS-422 Selection

RS-232 Port

As shipped, the RS-232 interface (Channel A) is jumpered as data communications equipment (DCE) for connection to a video terminal or printer. This interface can also be jumpered as data terminal equipment (DTE) for connection to another computer.

The jumper blocks that control the RS-232 port configuration are shown in Figure 2-17. Two standard configurations are shown along with the resulting RS-232 signal connections. Note that with either of these configurations, RS-232 pin 17 (TCLK) can also be connected to RS-232 pin 15 using jumper V3. For special applications jumpering refer to the VL-7842 schematics.

VL-7842 SIO Signal		DCE to Terminal RS-232 Signal (Pin#)	DTE to Computer
TXD (Xmit. Data)	---->	RD(3)	TD(2)
RXD (Recv. Data)	<----	TD(2)	RD(3)
RTS (Recv. Handshake)	---->	CTS(5)	RTS(4)
CTS (Xmit. Handshake)	<----	RTS(4)	CTS(5)
DTR (Xmit. Eqpt. Stat.)	---->	DSR(6)	DTR(20)
DCD (Recv. Eqpt. Stat.)	<----	DTR(20)	DSR(6)
TCLK (Xmit. Clock)	---->	TCLK(24)	RCLK(17/15)
RCLK (Recv. Clock)	<----	RCLK(17/15)	TCLK(24)

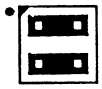
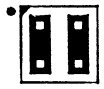
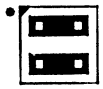
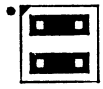
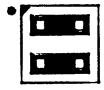
Jumper Setting for the Mode Shown	DCE to Terminal	DTE to Computer
V4		
V5		
V6		
V7		

Figure 2-17. RS-232 Serial Port Configuration Jumpers

RS-422 Ports

The RS-422 high speed serial ports allow data to be dependably transmitted over long distances (to 4000 feet) using twisted pair wire. A typical connection is shown in Figure 2-18. This cable may be optionally shielded if desired. Shielded cable has higher capacitance ratings than unshielded cable and may not be appropriate for longer cable runs.

RS-485 Multidrop

Using the jumpers shown in Figure 2-19 the RS-422 ports can be configured for RS-485 multidrop operation. This allows up to 32 driver/receivers to be connected to a single communications line. When the VL-7842 is used in multidrop applications, the 100 ohm terminating resistors should be removed from all stations except for the two farthest line locations (i.e. the start and end of the line).

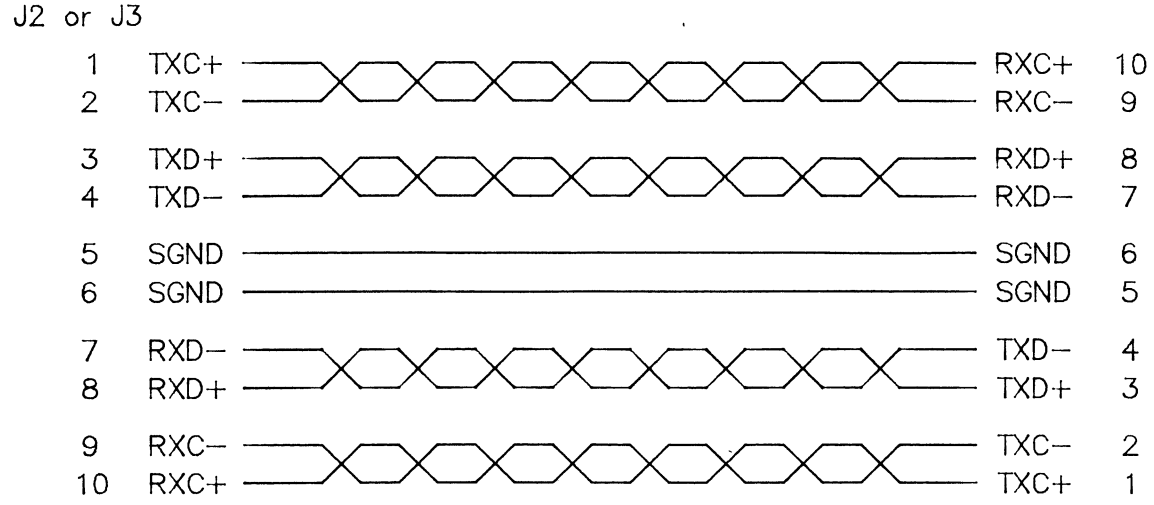


Figure 2-18. Typical RS-422 Connection

Channel	Multidrop Jumpers*	Terminators
A	V8a, V8b, V9a, V9b	R5, R6
B	V11a, V11b, V12a, V12b	R7, R8

* The four jumpers for each channel should be IN for multidrop operation and OUT for RS-422 operation.

Figure 2-19. Multidrop (RS-485) Mode Jumpers

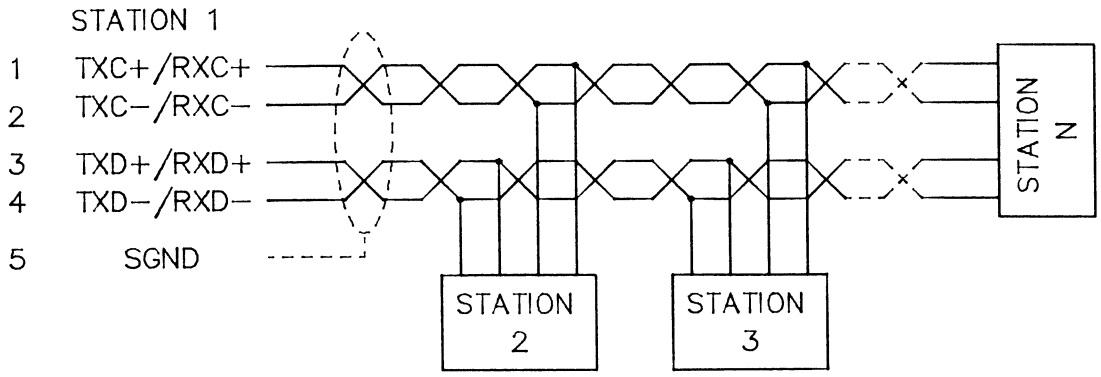


Figure 2-20. Typical Multidrop (RS-485) Connection

Baud Rate Clock

The baud rate inputs to the SIO chip are controlled by jumper block V2. Many configurations are available, depending on the baud rate clock source desired for each channel. The primary source of the baud rate clock is either the CTC chip, or the Received Clock line (RS-422 channels). Note that the channel A XMIT clock input is permanently connected to CTC channel 0. Jumpers a-e affect serial channel B; jumpers f-g affect serial channel A.

As shipped the board is configured with the channel A baud rate determined by CTC channel 0 (async mode), and the channel B transmit rate determined by CTC channel 1 (sync mode).

The baud rate options are shown in Figure 2-21. Several typical configurations are shown below. Refer to the VL-7842 schematic for special applications.

Channel A options:

Channel A async mode (jumper f IN, g OUT). Baud rate is determined by CTC channel 0.

Channel A sync mode (jumper f OUT, g IN). XMIT baud rate is determined by CTC channel 0. RECV rate is determined by external input (jumpers V13, V4, and V3)

Channel B options when used at same baud rate as A (or channel A not used):

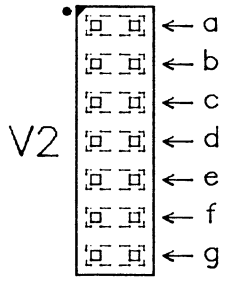
Channel B async mode (jumpers b,d IN, a,c,e OUT). Baud rate for both channels is determined by CTC channel 0.

Channel B sync mode (jumper a,d IN, b,c,e IN). XMIT baud rate is determined by CTC channel 0. RECV rate is determined by external input (RS-422 clock line).

Channels B at a different baud rate than channel A:

Channel B async mode (jumpers b,c,e IN, a,d OUT). Baud rate for channel B is determined by CTC channel 1.

Channel B sync mode (jumper a,d IN, b,d,e IN). XMIT baud rate is determined by CTC channel 0. RECV rate is determined by external input (jumpers V13, V4, and V3).



Jumper	Description	As Shipped
V2a -	Chan. B RECV clock from RS-422 line.	out
V2b -	Chan. B RECV clock from chan. B XMIT.	out
V2c -	Chan. B XMIT clock from CTC chan. 1.	IN
V2d -	Chan. B XMIT clock from CTC chan. 0.	out
V2e -	SYSCLK/2 to CTC chan. 1 input.	IN
V2f -	Chan. A RECV clock from CTC chan. 0.	IN
V2g -	Chan. A RECV clock from extrnl. line.	out

Figure 2-21. Baud Rate Clock Jumpers

COUNTER/TIMER CHANNELS

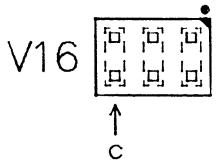
The Counter/Timer Chip (CTC) consists of four timer/counter channels. Normally one or two of these channels are used to generate the baud rate clock for the serial I/O channels. The remaining channels can be used for general purpose timing/counting.

CTC channel 0 is always connected to the SIO channel A XMIT clock input. The CTC channel 1 output may be connected to SIO channel B using jumper V2c. Jumper V2e can connect the system clock, divided by two, to the CTC channel 1 input (for baud rate generation or general purpose timing). The CTC's I/O lines, except for channel 0, are available at connector J4.

WAIT STATE OPTION

The wait state jumper option allows the Z80 op code fetch cycle to be lengthened. This allows use of the VL-7842 card with slower memory devices without significant degradation in system processing speed. It is especially useful with the 6 MHz VL-7842 version.

The jumper option, and the resulting access times, are shown in Figure 2-22. As shown, a 4 MHz card operating without the wait state jumper would require RAM and ROM devices with an access time of 250 ns or better.



Jumper V16c	Minimum Access Speed 2.5 MHz	3.68 MHz	4.0 MHz	6.0 MHz
IN	600	350	350	200
out	450	250	250	150

Figure 2-22. Wait State Option

SYSTEM GROUND

As delivered the VL-7842 card connects the +5V and ±12V power supply grounds together (STD BUS pins 3-4 to 53-54). Interconnection of these grounds is required whenever the RS-232 serial I/O port is in use. They may also be connected together at the system power supply or on the backplane.

Special applications that require separated digital and auxiliary power supplies, and are not using the on-board RS-232 port, may break the common ground by removing jumper V18b (see Figure 2-15).

Section 3
OPERATION

INTRODUCTION

This section includes information on the use (programming) of the various VL-7842 functions. Both background information and software examples are presented to assist you in constructing your own software routines.

MEMEX CONTROL

The MEMEX (memory expansion) signal on the STD BUS is normally used to select between two different 64K byte memory banks or maps. It can be used to expand the available memory, or to control a bootstrap PROM.

Memory cards in the system can choose to operate with MEMEX low (standard map), MEMEX high (secondary map), or ignore MEMEX (both maps).

In order for the MEMEX signal to be controlled by the VL-7842 board, jumper V17 must be installed. Jumper V16 determines the state of MEMEX at power-up and whether data is inverted when written to the MEMEX control port. The effect of the V16 jumper is shown in Figure 3-1.

The MEMEX signal is controlled by bit 0 at I/O port FE. Writing a 0 or 1 to this port will set MEMEX low or high depending on the setting of jumper V16.

Note that the routine that controls (changes) the state of MEMEX must be located in shared or common memory. That is, it must be in a section of memory (RAM or ROM) that is selected both when MEMEX is low and when MEMEX is high. If the switching routine is not resident in both memory maps, the system will crash when MEMEX is changed.

Figure 3-2 shows the typical code used to control the MEMEX signal (when jumper V16a is out and V16b is in).

Jumper V16 Description	MEMEX Port	As Shipped
a - MEMEX is set high at power-up.	Inverted	a - out
b - MEMEX is set low at power-up.	True	b - IN

Figure 3-1. MEMEX Jumper Options

```
                    ;Select secondary memory area routine
0100    3E 01       SELSEC LD       A,01H           ;
0102    D3 FE              OUT      (0FEH),A        ;Output 01H to MEMEX port
0104    C9                 RET
                    ;
                    ;Select primary memory area routine
0105    3E 00       SELPRI LD       A,00H           ;
0107    D3 FE              OUT      (0FEH),A        ;Output 00H to MEMEX port
0109    C9                 RET
```

Figure 3-2. MEMEX Software Example

COUNTER/TIMER CHIP (CTC)

The Counter/Timer Chip (CTC) can be used to count external events, measure pulse widths, and generate timed output pulses. Programmable options allow selection of external or on-board triggering, rising or falling edge triggering, prescaling, and Mode 2 interrupts at countdown completion. The CTC can also be used to generate prioritized vectored interrupts from three external inputs. The CTC supports the priority interrupt chain.

The CTC consists of four timer/counter channels. Normally two of these channels are used to generate baud rate clocks for the serial I/O channels. This leaves at least two CTC channels available for general purpose use. The number of available channels/functions is shown in Figure 3-3. The four channels are numbered 0 through 3.

	---- Baud Clocks Used ----		
	2 Baud Clocks	1 Baud Clock	No Baud Clocks
Function	**--- Channels Available ---**		
Timers	2,3	1,2,3	0,1,2,3
Inputs (for counting or interrupts)	2,3	1,2,3	1,2,3
Output Lines (for output pulses)	2	1,2	1,2

Figure 3-3. CTC Channels available

Internally each CTC channel has three registers- Load (initial count), Current Count, and Control. In addition, there is a vector register which is accessed through channel 0. All registers are 8 bits wide.

The Control and Current Count registers can be accessed directly by the

processor. The Load registers are written after setting the required bit in a Control register. The Vector register is written directly using a special bit in the channel 0 control register. The port addresses for these registers are shown in Figure 3-4.

I/O Port	Write (OUT)	Read (IN)
F0	Chan. 0 Control, Load & Vector	Chan. 0 Current Count
F1	Chan. 1 Control and Load	Chan. 1 Current Count
F2	Chan. 2 Control and Load	Chan. 2 Current Count
F3	Chan. 3 Control and Load	Chan. 3 Current Count

Figure 3-4. CTC Register Addresses

CTC CONTROL REGISTER

D7	D6	D5	D4	D3	D2	D1	D0
ENABLE INTPS.	COUNTER/ TIMER	CLOCK DIVIDER	CLOCK EDGE	TRIGGER	LOAD REG.	CHANNEL RESET	VECTOR LOAD

Bit Description

D7 1=Enable Interrupts

D6 0=Timer Mode
1=Counter Mode

D5# 0=Clock divided by 16
1=Clock divided by 256

D4 0=External trigger is falling edge sensitive
1=External trigger is rising edge sensitive

D3# 0=Timer starts when loaded
1=Timer starts from external trigger

D2 1=Next byte to this channel goes to the LOAD register

D1 1=Reset (stop) counter (until registers are reloaded)

D0 0=Vector register write (illegal except to channel 0)
1=Normal register access (must be 1 except for channel 0)

Timer mode only.

Figure 3-5. CTC Control Register

Control Register

The control register for each of the four channels is identical. Figure 3-5 shows the function of each data bit in the register.

D7 - Enable Interrupts: Allows an interrupt to be generated whenever the counter reaches zero.

D6 - Counter/Timer: Selects the counter mode or the timer mode.

D5 - Clock Divider: Divides the main CTC clock input by 16 or 256. Applies to the timer mode only.

D4 - Clock Edge: Selects rising or falling edge triggering from the external input.

D3 - Trigger: Starts timer from an external pulse, or whenever the Load register is written to. Applies to the timer mode only.

D2 - Load Register Control: Allows the next byte written to be put into this channel's Load register.

D1 - Channel Reset: Stops the counter. Does not clear any registers. Does not clear other control register bits (they must be set as desired when this bit is written). Counter/timer will restart when reloaded, retriggered, etc.

D0 - Vector Load: Writes the current byte (upper 5 bits) to the Vector register (channel 0 only). This bit must be set high at all other times.

Load Register

The Load register for each channel hold the initial value for the counter. The counter automatically reloads itself from the Load register when it reaches zero. This register may be loaded with a new value without disturbing a current countdown in progress. The Load register may hold 01 thru FF(255), and 00 which yields 256 counts.

Current Count Register

The Current Count register contains the current value in the counter. It may be read at any time without affecting the current countdown in progress.

Vector Register

A single Vector register, for the entire CTC chip, exists in channel 0. It is updated by writing to the channel 0 control register with D0 set to zero. The upper part of this byte (D3-D7) is immediately written to the Vector register.

CTC Operation

Operation of the CTC is based around the counters which can be loaded with an initial value, clocked from an internal or external source, and will do something when the count decrements down to zero.

To use a channel simply write the desired bits to the control register (with D2=1) and load a value into the Load register. If interrupts are used, the Vector register must also be initialized. Once started, the Current Count register can be read at any time to determine the current progress of the countdown.

In the timer mode, the counter is decremented by the system clock (divided by 16 or 256). No external clock input is needed. Note that the timer intervals will change if the same routines are used on a CPU card with a different system clock rate.

Since the counters count down only, the value read from the Current Count register should be inverted when counting external events or measuring time intervals. Loading the channel (Load register) with hex FF, and then inverting the value read from the Current Count register (do an Exclusive OR with FF) will yield the actual count.

Output Pulses. The CTC can be used to generate output pulses of a desired frequency. The output line for each channel goes high momentarily each time the counter reaches zero. Using the timer mode, the value of the Load register and clock divider determines the output frequency.

Cascading. Channels can be cascaded together for higher counts or longer time delays. This is done by connecting the output of the first channel (in counter or timer mode), to the external input of the second channel (in counter mode). Each time the first channel reaches zero, it will decrement the count in the second channel.

Software Trigger. Once a channel has been initialized, it can be triggered directly by the CPU if desired. In the timer mode the counter can be started by the processor. In the counter mode, the processor can decrement the counter by one. This is performed by writing to the control register of the desired channel, with the clock edge bit (D4) complemented from the last setting. This acts as a pulse, just as a change in polarity of the external input, and will start the timer or decrement the counter.

Interrupts. The CTC interrupts can only be used in the Z80 interrupt mode 2. The CTC Vector register must be initialized before interrupts are enabled from any CTC channel. The vector value generated by each counter channel is shown in Figure 3-16.

Note that the CTC can also be used as an external interrupt prioritizer. Putting a channel(s) in the counter mode, with a count of one in the Load register, will cause an interrupt every time the desired transition occurs on an external input line.

Care should be taken when disabling a CTC interrupt. It is possible to disable an interrupt while it is in progress (after the interrupt has

occurred and before the vector has been given to the processor). To avoid this possibility the following procedure should be followed whenever a CTC interrupt is disabled: Disable Z80 interrupts, disable CTC interrupt(s), re-enable Z80 interrupts.

The Z80 Return From Interrupt command should always be used at the end of your interrupt service routines. This re-enables the interrupt priority chain.

On-Board Connections. On-board jumpers may be used to connect CTC inputs to the system clock (divided by 2), and/or connect CTC outputs to baud rate inputs on the serial I/O channels.

Various CTC jumper options are available at jumper block V2 as shown below. For additional information see the Option Jumpers and Serial I/O sections.

Jumper	Description	As Shipped
V2a	Chan. B RECV clock from RS-422 line.	out
V2b	Chan. B RECV clock from chan. B XMIT.	out
V2c	Chan. B XMIT clock from CTC chan. 1.	IN
V2d	Chan. B XMIT clock from CTC chan. 0.	out
V2e	SYSCLK/2 to CTC chan. 1 input.	IN
V2f	Chan. A RECV clock from CTC chan. 0.	IN
V2g	Chan. A RECV clock from extrnl. line.	out

Figure 3-6. CTC jumper options.

Timing Notes. Detailed timing for the CTC can be found in the Zilog Z8430A and Mostek MK3882-4 data sheets. Several general points are noted below.

- The zero state does not require an additional clock pulse. With a Load register of "4" the countdown would be 4, 3, 2, 1, 4, 3, ...

- The output line goes high (at zero count) for only about one system clock cycle. It does not stay high even in the counter mode.

- External trigger pulses must not be faster than one half of the system clock rate (2.5 to 6 MHz depending on the CPU version).

- External inputs must remain high or low at least 200 nanoseconds.

- If a Control register is written with the Load bit set, the timer trigger is not enabled until the Load register byte is written.

Software Example. Figure 3-7 shows a routine that uses CTC channel 3 to generate an interrupt every 1 millisecond.

```
                    ;CTC EXAMPLE
                    ;
                    ;FOR A 4.0 MHZ CARD
                    ;
          F0 00     CTC     EQU     0F0H    ;CTC base address
          F0 00     CTC0    EQU     CTC+0   ;CTC channel 0 control
          F3 00     CTC3    EQU     CTC+3   ;CTC channel 3 control
                    ;
0000                        ORG     0000H   ;Start of program
                    ;
                    ;Routine to generate an interrupt every 1ms
                    ;using CTC channel 3.  Uses mode 2 interrupts to
                    ;access interrupt service routine (SVC3)
                    ;
                    ;Load Interrupt Control Vector register with
                    ;upper 8 bits of interrupt table address
0000   3E 01        INITINT LD      A,01H           ;8 hi bits of INTSVC3
0002   ED 47                LD      I,A             ;Load CPU register
                    ;Set Z80 mode 2 interrupts
0004   ED 5E                IM      2               ;
                    ;Load CTC interrupt vector with lower 8 bits
                    ;of interrupt table address
0006   3E 20                LD      A,20H           ;8 lo bits of INTSVC3
0008   D3 F0                OUT     (CTC0),A        ;Load CTC interrupt vector
                    ;Initialize CTC to timer mode, divide system clock
                    ;by 16, with a counter value of 250.
                    ;(4MHZ/16/250 = 1KHZ)
000A   3E 85                LD      A,10000101B     ;Control data
000C   D3 F3                OUT     (CTC3),A        ;
000E   3E FA                LD      A,250           ;Count data
0010   D3 F3                OUT     (CTC3),A        ;1KHZ clock started
                    ;Enable CPU interrupts
0012   FB                   EI                      ;
0013   C9                   RET
                    ;
                    ;
                    ;Interrupt service jump address table
                    ;
0120                        ORG     120H
                    ;
0120   00 00        INTSVC0 JR      0000H   ;SVC0 = interrupt 0 routine
0122   00 00        INTSVC1 JR      0000H   ;SVC1 = interrupt 1 routine
0124   00 00        INTSVC2 JR      0000H   ;SVC2 = interrupt 2 routine
0126   50 01        INTSVC3 JR      SVC3    ;SVC3 = 1ms interrupt routine
                    ;
0150                        ORG     150H
                    ;
                    ;User supplied interrupt service routine for 1ms interrupt
0150   00           SVC3    NOP                     ;USER CODE
0151   FB                   EI                      ;Enable interrupts
0152   ED 4D                RETI                    ;Return from interrupt
                    ;
```

Figure 3-7. CTC Software Example

SIO CHIP

The serial I/O ports on the VL-7842 card are implemented with an SIO (Serial Input/Output) chip (Zilog Z8442). General operation of the SIO chip is discussed here, while its specific use is detailed in later sections.

It should be noted that the SIO is also sold for asynchronous applications as the DART (Dual Asynchronous Receiver/ Transmitter). The DART is identical to the SIO, except that the synchronous functions of the DART may not be fully operational. For asynchronous operation, the same software can be used with both the SIO and the DART chip.

Internal Structure

The SIO chip includes two independent channels, channels A and B. Both channels are identical in structure and operation (except for the interrupt vector which is set in channel B and is common to both channels). Each SIO channel has 11 registers. Four of these can be directly read/written by the processor. The remainder are accessed by setting pointer bits in the control 0 register.

A list of the SIO's registers is shown in Figure 3-8. The detailed functions of these registers are shown in Figure 3-9. Each register is discussed in detail below.

I/O Port	Write (OUT)	Read (IN)
F4	Chan. A XMT data	Chan. A RCV data
F5	Chan. A Control Reg. 0	Chan. A Status Reg. 0
F6	Chan. B XMT data	Chan. B RCV data
F7	Chan. B Control Reg. 0	Chan. B Status Reg. 0

Figure 3-8. SIO I/O Port Locations

Name	Use	Access
XMT	Data to transmit	I/O port
Control 0	Commands and other register access	I/O port
Control 1	Interrupt control	Via Control 0
Control 2	Interrupt vector set (chan B only)	Via Control 0
Control 3	Receiver control	Via Control 0
Control 4	Clock and word control	Via Control 0
Control 5	Transmitter control	Via Control 0
Control 6	Sync character or SDLC address	Via Control 0
Control 7	Sync character or SDLC flag	Via Control 0
RCV	Data received	I/O port
Status 0	Ready/busy bits	I/O port
Status 1	Receive error flags	Via Control 0
Status 2	Interrupt vector read (chan B only)	Via Control 0

Figure 3-9. SIO Registers

REGISTER	D7	D6	D5	D4	D3	D2	D1	D0
CNTL 0	--CRC RESET[s]--			------COMMANDS[e]----			---REGISTER SELECT---	
CNTL 1	WT/RDY ENABLE	±READY SELECT	WT/RDY ON RCV	RCV INT MODE[p]		VECTOR MODE[b,q]	XMT INT	HNDSHK INT
CNTL 2	----------INTERRUPT VECTOR (WRITE)[b]----------							
CNTL 3	---RCV CHAR--- LENGTH[k]		AUTO HNDSHK	START HUNT	RCV CRC ENABLE	ADDR MATCH	SYNC STRIP	RCV ENABLE
CNTL 4	CLOCK DIVIDER[m]		--SYNC MODE[t]--		-STOP BITS[c]		EVEN PARITY	PARITY ENABLE
CNTL 5	DTR CNTL	---XMT CHAR--- LENGTH[k]		SEND BREAK	XMTR ENABLE	SDLC/ CRC-16	RTS CNTL	XMT CRC ENABLE
CNTL 6	----------SYNC BIT 7 - 0 (OR SDLC ADDRESS FIELD)----------							
CNTL 7	----------SYNC BIT 15 - 8 (OR SDLC FLAG CHARACTER)----------							

[a] Channel A only.
[b] Channel B only.

[e] 000 No operation
001 Send abort (SDLC)
010 Reset handshake INT
011 Channel reset
100 Enable INT on next char RCVD
101 Reset XMT INT
110 Error reset
111 Return from INT[b]

[s] 00 No operation
01 Reset RCV CRC checker
10 Reset XMT CRC generator
11 Reset XMT Underrun/EOM latch

[p] 00 RCVR INT disable
01 RCVR INT on first char.
10 RCVR INT on every char., parity affects vector
11 RCVR INT on every char., parity doesn't affect vector

[k] 00 5 bits/char.
01 7 bits/char.
10 6 bits/char.
11 8 bits/char.

[m] 00 Clock divided by 1
01 Clock divided by 16
10 Clock divided by 32
11 Clock divided by 64

[t] 00 8 bit programmed sync
01 16 bit programmed sync
10 SDLC mode (01111110 flag)
11 External sync mode (not used)

[c] 00 Sync mode
01 1 stop bit
10 1 1/2 stop bits
11 2 stop bits

[q] INT vector will be modified as follows if D2 in CNTL 1 is set.

D3	D2	D1[b]	
0	0	0	Chan B XMT buffer empty
0	0	1	Chan B Handshake change
0	1	0	Chan B RCVD char. ready
0	1	1	Chan B Special RCV condition
1	0	0	Chan A XMT buffer empty
1	0	1	Chan A Handshake change
1	1	0	Chan A RCVD char. ready
1	1	1	Chan A Special RCV condition

Figure 3-10. SIO Control Register Functions

REGISTER	D7	D6	D5	D4	D3	D2	D1	D0
STAT 0	LINE[r] BREAK	XMIT EOM	CTS[r]	SYNC/ HUNT	DCD[r]	XMT BUF. EMPTY	INT[a] PEND.	CHAR RCVD
STAT 1	FRAME END	CRC/FRM ERROR	OVERRUN ERROR	PARITY ERROR	--SDLC RESIDUE CODE[s]--			XMTR DONE
STAT 2	----------INTERRUPT VECTOR (READ)[b,q]----------							

[a] Channel A only.
[b] Channel B only.

[r] A "handshake interrupt" signal

[q] INT vector will be modified as follows if D2 in CNTL 1 is set.

D3	D2	D1[b]	
0	0	0	Chan B XMT buffer empty
0	0	1	Chan B Handshake change
0	1	0	Chan B RCVD char. ready
0	1	1	Chan B Special RCV condition
1	0	0	Chan A XMT buffer empty
1	0	1	Chan A Handshake change
1	1	0	Chan A RCVD char. ready
1	1	1	Chan A Special RCV condition

[s] The SDLC residue codes are:

D3	D2	D1	I field in prev. byte	I field in 2nd prev. byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

Figure 3-11. SIO Status Register Functions

RCV Register

The RCV register holds the data that has been received at the serial port. It can be read directly by the processor.

The RCV register is triple buffered so that several characters can be received without the processor reading them immediately. Note however that the Status 1 register is always valid for the data currently in the RCV register. Once the RCV register is read, the next byte of received data (if any) is made available, and the Status 1 register is updated for the new byte in the RCV register. The Status 1 register should always be read first, before the data in the RCV register is read, or it will be lost.

In the receiver interrupt mode (see the Control 1 register) an error (in Stat. 1 register) will freeze both data and status registers until an error reset is executed (see Cntl. 0 register).

Note that if less than 8 data bits are being received the unused upper bit(s) will contain garbage. The unused bit positions should be cleared (with an AND command) before the data is processed.

XMT Register

The XMT register holds data to be transmitted by the channel. It is directly addressable by the processor. The XMT register is buffered, allowing the next character to be written while the current one is being transmitted.

An interrupt and/or flag (in the Status 0 register) indicates when the buffer is available to accept another character. The Status 1 register contains a flag that indicates when the actual transmitting has been completed.

Control 0 Register

The Control 0 register serves two functions. It executes special commands (as noted below), and its register pointer bits (D2-D0) allow other registers (Cntl. 1-7 and Stat. 1-2) to be accessed. It is directly addressable by the processor.

Access to most SIO registers requires two writes to the SIO chip. The first write sets the register pointer bits in the Cntl. 0 register. The second write contains data which is directed to the selected register. The second write is replaced by a read in the case of the status registers.

Several commands can be written to the Control 0 register and are executed immediately. Register pointer bits can be included with most commands, in which case the next data read or write will be directed to the selected register. The commands are described below.

In addition, bits D7-D6 of the Control 0 register allow resetting of the

CRC generator and the transmitter underrun/EOM latch. Refer to Figure 3-10 for the reset codes available.

No Operation. This command is used when the pointer register is being set and no other effect is desired.

Send Abort (SDLC). Causes a sequence of 8 to 13 ones to be transmitted. Used only with the SDLC mode.

Reset Handshake INT. If an incoming handshake line changes state, or a break or end of break is detected, bits 3, 4, 5, and 7 in Stat. 0 become frozen (and an interrupt is generated if enabled by Cntl. 1, bit 0). Until the Reset Handshake INT command is executed these bits will not change, and no further interrupts can be caused by them. If the command is executed and the status of external lines has changed, the bits are updated (with an interrupt if enabled) and then frozen once again.

Channel Reset. This command resets a single channel of the SIO, as if a system reset had occurred. On Channel A it also resets the prioritization logic. All control registers must be reinitialized before the channel is used again. A delay of four clock states (i.e. one Z80 NOP instruction) must occur before the channel is written to again. Since this command clears the Control 0 register, the pointer bits cannot be set along with this command.

Enable INT On Next Character RCVD. If an "Interrupt on Received Character" mode is selected (Control 1, D4-D3) it must be reenabled after each interrupt (by executing this command).

Reset XMT INT. This command resets a transmitter interrupt that occurs after the last character in a group has been sent (when you have no more characters to transmit). When additional characters are sent the XMT Buffer Empty interrupt (vector) will occur again (assuming the interrupt is enabled).

Error Reset. The Parity Error and Overrun flags are latching to allow their examination at the end of a block transfer. This command clears these flags.

Return From INT. This command has the same effect as a return-from-interrupt command performed by the processor. This command can only be issued to Channel A, but affects both SIO channels by resetting the interrupt-under-service latch, allowing lower priority devices to interrupt.

Control 1 Register

This register is used for control of interrupts, and for controlling use (if any) of the Wait/Ready lines (channel A and channel B) on the SIO. The Wait/Ready lines can be used to control block transfers of data from/to the SIO. The Wait/Ready signals can jumpered to the CPU wait line (WAITRQ*) using jumper block V1. These signals are also available on connector J4 for external connection, such as to a DMA controller.

The Control 1 register is accessed via the Channel 0 register.

D7 - Wait/Ready Enable: The SIO Wait/Ready line remains high (in the Ready mode) or floating (in the Wait mode) until it is enabled by setting this bit high.

D6 - ±Ready Select: Setting this bit high puts the Wait/Ready line into the Ready mode, while setting it low keeps it in the Wait mode. In the Ready mode, the Wait/Ready line (for the applicable channel) goes high whenever the receiver or transmitter needs service. In the Wait mode, the line goes high whenever the processor attempts to read or write data to the SIO and it is not ready to accept or supply the requested data.

D5 - Wait/Ready on RCV: Set this bit high to use the Wait/Ready line for RCVR control, low for XMIT control.

D4-D3 - RCVR Interrupt Mode: Selects the Receiver Interrupt mode. See Figure 3-10 for the options available. Note that the Parity Affects Vector mode will cause the Special RCV Condition vector to be used when a parity error occurs (if D2 in this register is also set high).

D2 - Vector Mode: If this bit is set (high), the interrupt vector is altered depending on the interrupt condition. Refer to Figure 3-10 for the interrupts generated. This bit can be set only in Channel B but affects the whole SIO.

D1 - XMT Interrupt Enable: Setting this bit high causes interrupts (Transmit vector) to occur when the XMT data register is empty.

D0 - Handshake Interrupt Enable: Enables interrupts to occur when there are changes on the handshake lines, or the start or end of a break condition. The handshake signals are DCD, RI, and CTS (Status 0 register).

Control 2 Register

The Control 2 register is used to program the interrupt vector. It can be written only to Channel B, but affects the whole SIO. Three bits of the vector are replaced by the SIO during interrupts if the Vector Mode bit (Control 1, D2) is set (the original vector written to this register is not changed). Note that in order to be compatible with the Z80 interrupt scheme the lowest bit of this register must be a zero. This register is accessed via the Control 0 register.

Control 3 Register

This register contains receiver control bits, and controls the auto handshake mode. It is accessed via the Control 0 register.

D7-D6 - RCVR Character Length: Sets the data word size for receiving (selected length should not include parity bit). Refer to

Figure 3-10 for the lengths available. Note that if the data word length is less than 8 bits, the upper bits will contain garbage that should be discarded (masked) by the receive routine.

D5 - Auto Handshake Enable: Setting this bit high enables the auto handshake mode. When enabled the DCD* line must be active to enable the receiver. The CTS* line must be active before the transmitter is enabled. These signals appear in the Status 0 register and can be monitored directly if desired.

D4 - Start Hunt Mode: Setting this bit high causes the hunt mode to be re-started. This can be done when synchronization is lost, or when the incoming message is not needed (SDLC mode).

D3 - Receiver CRC Enable: Setting this bit causes CRC calculation to begin with the next received character.

D2 - Address Match Mode: When this bit is set, and the SDLC mode selected, messages are rejected (no interrupts are generated) unless its address matches the programmed (in Control reg. 6) or global (11111111) address.

D1 - Sync Character Strip: Setting this bit causes Sync characters preceding the message (leading sync characters) to not appear in the receiver buffer. Since CRC calculations are not stopped in this mode, it should only be used at the start of a message.

D0 - RCVR Enable: Setting this bit high enables the receiver. It is normally left high.

Control 4 Register

This register controls baud rate clock division, parity, and stop bit selection. It is accessed via the Control 0 register.

D7-D6 - Clock Divider: These bits select the number by which the incoming clock (usually from the CTC) is divided. This sets the baud rate for both the receive and transmit channels (they are always the same). The X1 mode must be selected for Sync use. The baud rate must never be more than five times the system clock rate (CPU clock).

D5-D4 - Sync Mode: These bits select the character synchronization options. Note: The external sync mode is not available on the VL-7842.

D3-D2 - Stop Bits: Selects the number of stop bits to be transmitted after each character (for asynchronous modes). Note: The receiver always operates properly with one or more stop bits.

D1 - Even Parity Select: This bit selects even parity (high) or odd parity (low). It affects both transmitter and receiver.

D0 - Parity Enable: Setting this bit high adds a parity bit to the character length. It affects both the transmitter and receiver.

Control 5 Register

This register contains transmitter and handshake controls. It is accessed via the Control 0 register. Note that the DTR and RTS handshake lines may actually be jumpered to other signal lines (DSR and CTS respectively) depending on whether the channel is in the terminal or modem configuration. The descriptions below refer to the signals as they appear at the jumper block and the card edge, not as they come out of the SIO chip (inverted).

D7 - DTR Control: The DTR signal goes high or low according to the setting of this bit.

D6-D5 - XMT Character Length: Sets the data word size for the transmitter (selected length does not include parity bit).

D4 - Send Break: Setting this bit forces the transmit data pin to go low (line break condition) regardless of any data being transmitted.

D3 - XMIT Enable: Setting this bit high enables the transmitter. Setting it low disables the transmitter (after any character being sent is completed). Note: If it is disabled during the transmission of a CRC character, sync or flag characters are sent in place of the CRC.

D2 - SDLC / CRC-16 select: Setting this bit high causes CRC-16 polynomial to be used (by both the transmitter and receiver). Setting it low causes the SCLC polynomial to be used.

D1 - RTS Control: Setting this bit high causes the RTS signal to go high. Setting this bit low causes RTS to go low (only after the transmitter is empty). This line is used only with the RS-232 interface on channel A.

D0 - Transmit CRC Enable: Setting this bit causes the CRC to be generated on the next character transferred from the internal transmit buffer into the transmitter. The CRC is not automatically transmitted (on a transmit buffer empty condition) unless this bit is set.

Control 6 Register

This register contains transmitter sync character in the Monosync mode, or the first 8 bits of a Bisync sync character. In SDLC mode it can contain the address to compare against the incoming SDLC frame address field.

Control 7 Register

This register can be programmed with the receive sync character in the Monosync mode, the second 8 bits of a Bisync sync character, or a flag character (01111110) in SDLC mode.

Status 0 Register

The Status 0 register contains handshake and SIO status flags. It is directly accessible by the processor. Note that the CTS, RI, and DCD handshake lines may actually be jumpered to other signal lines depending on the configuration of the on-board jumpers. The descriptions below refer to the signals as they appear at the jumper block and the card edge, not as they come out of the SIO chip (inverted).

D7 - Line Break: This bit indicates that a break condition has been detected on the receive data pin (continuous zeros in asynchronous mode). This bit is also set if an abort sequence (7 or more ones) is received in the SDLC mode.

D6 - XMIT Underrun/ End of Message: This bit is set following a reset. It is reset only with a Reset XMIT Underrun/EOM Latch command (in Control 0). It is set whenever a transmitter underrun (buffer and transmit register empty) occurs (which also causes an External/Status interrupt).

D5 - CTS Status: This bit indicates the state of the CTS input line. This bit can become latched (see the Reset Handshake Interrupt command in the Control 0 register). This bit is used only with the RS-232 interface operating on channel A.

D4 - Sync/ Hunt Status: This bit is low if synchronization has occurred, or high if currently in the Hunt mode.

D3 - DCD Status: This bit indicates the status of the DCD line. It can be latched like the CTS bit. This bit is used only with the RS-232 interface operating on channel A.

D2 - XMT Buffer Empty: This bit is set whenever the XMT Data register is empty (except when a CRC character is being automatically sent in sync or SDLC mode).

D1 - Interrupt Pending: This bit is high if an interrupt is pending for either channel. It can be read only from Channel A.

D0 - Character RCVD: This bit indicates that a character is available in the RCV Data register. It is reset automatically when the RCV Data register is read.

Status 1 Register

The Status 1 register contains received data error and transmitter status flags. It is accessed via the Control 0 register.

D7 - Frame End (SDLC): This bit is used only in the SDLC mode to indicate that a valid ending flag has been received, and that the CRC Error and Residue codes are valid. It is reset by the Error Reset command.

D6 - CRC/Framing Error: This bit indicates that a framing error (improper location of the stop bit) occurred in a received character (async mode). In sync. mode it indicates that a CRC error has occurred, but it is valid only at the end of a message.

D5 - Overrun Error: This bit indicates that an overrun error has occurred. This is caused when characters received by the SIO are not read soon enough by the system processor. On error, this bit latches and remains frozen until an Error Reset command is issued to the Control 0 register.

D4 - Parity Error: This bit indicates that a parity error occurred in the received character. On error, this bit latches and remains frozen until an Error Reset command is issued to the Control 0 register.

D3-D1 SDLC Residue Codes: The residue codes indicate the length of the I-field for SDLC messages where the I-field is not an integral multiple of the character length. It is valid only when the End Of Frame bit is set.

D0 - XMTR Done: In async. mode this bit indicates that the transmitter itself (not the XMT buffer) is completely done transmitting. In sync. mode it is always set (high).

Status 2 Register

The Status 2 register allows the current interrupt vector to be read. It is available only on Channel B, and is accessed via the Control 0 register. If the Status Affects Interrupt bit (CNTL 1, D2) is not set, the vector, exactly as written into the Control 2 register, is read. Otherwise the vector is modified according to the interrupt pending (see Vector Mode in the Control 1 register). If no interrupt is pending, the vector is modified with D3=0, D2=1, and D1=1.

SERIAL PORTS

This section discusses actual operation of the serial channels on the VL-7842 card. It encompasses information presented in the last several sections. Software examples are included for typical operating modes.

Initialization

Initialization of the serial channels should occur in the following order: CTC baud rate clock start-up, SIO registers 2 and 4, other SIO registers.

Baud Rate Clock

The baud rate clock for the SIO channels can be supplied by the on-board CTC chip, or the clock signals from the RS-422 lines in several different configurations. As delivered, the baud rates for channel A and B are controlled by CTC channel 0 and 1 respectively. Other baud rate jumper options are shown below.

Jumper	Description	As Shipped
V2a	- Chan. B RECV clock from RS-422 line.	out
V2b	- Chan. B RECV clock from chan. B XMIT.	out
V2c	- Chan. B XMIT clock from CTC chan. 1.	IN
V2d	- Chan. B XMIT clock from CTC chan. 0.	out
V2e	- SYSCLK/2 to CTC chan. 1 input.	IN
V2f	- Chan. A RECV clock from CTC chan. 0.	IN
V2g	- Chan. A RECV clock from extrnl. line.	out

Figure 3-12. Baud Rate Clock Jumpers

Initializing the baud rate clock(s) consists of writing two bytes to the CTC. The first byte sets the counter/timer mode, the clock divider rate, and the load bit (that signals that the next byte written to the CTC will go the the Load register). The second byte written is the data hat determines the beginning value of each countdown sequence.

The actual values for these parameters depend on the baud rate desired, and the system clock speed. Charts detailing the values for many common baud rates are included in the Reference section at the end of this manual. Note that some standard baud rates are not available with certain system clock rates (e.g. 9,600 baud with a 4 MHz clock) due to these clocks not being divisible closely enough to the desired rate. The maximum allowable baud rates available are shown below.

	System Clock Speed			
	2.5 MHz	**3.6864 MHz**	**4.0 MHz**	**6.0 MHz**
Async. Mode	38.4K	57.6K	62.5K	93.7K
Sync. Mode	312.5K	460.8K	500K	750K

Fig. 3-13. Maximum Allowable Baud Rates

Examples of CTC initialization for baud rate generation are shown in Figure 3-14. Note that the VL-7842 normally uses two CTC timers, one for each serial channel. This leaves two timers of the four timers available for general use.

SIO Initialization

Initialization of the SIO channel(s) is most easily accomplished with parameter tables. The tables can be written to the SIO with the Z80 block move instruction.

There are a number of modes in which to operate the SIO. The examples show in the listings below initialize it for polled (non-interrupt) operation. The data in the initialization table can be changed to initialize the SIO for any other type of operation desired. If channel A and B will be operated in the same mode then a single table can be used for both of them.

```
                ; SIO INITIALIZATION EXAMPLE FOR THE VL-7842 CARD
                ;
                ; THIS EXAMPLE IS FOR A 4.0 MHZ VERSION CARD
                ;
      F0 00     CTC     EQU     0F0H    ;CTC base address
      F0 00     CTC0    EQU     CTC+0   ;CTC channel 0 control
      F1 00     CTC1    EQU     CTC+1   ;CTC channel 1 control
                ;
      F4 00     SIO     EQU     0F4H    ;SIO base address
      F4 00     CHADATA EQU     SIO+0   ;SIO CHA data
      F5 00     CHACTRL EQU     SIO+1   ;SIO CHA control
      F6 00     CHBDATA EQU     SIO+2   ;SIO CHB data
      F7 00     CHBCTRL EQU     SIO+3   ;SIO CHB control
                ;
0000                    ORG     0000H   ;PROGRAM LOCATION
                ;
```

```
                    ;Initialize CTC output 0 (SIO channel A) for 9600 Baud
0000   3E 45              LD      A,01000101B   ;(45H) Control data
0002   D3 F0              OUT     (CTC0),A      ;Write control data
0004   3E 0D              LD      A,0DH         ;Count data for 9600 Baud
0006   D3 F0              OUT     (CTC0),A      ;Write count data - DONE
                    ;
                    ;Initialize CTC output 1 (SIO channel B) for 100K Baud
0008   3E 45              LD      A,01000101B   ;(45H) Control data
000A   D3 F1              OUT     (CTC1),A      ;Write control data
000C   3E 0A              LD      A,0AH         ;Count data for 100K Baud
000E   D3 F1              OUT     (CTC1),A      ;Write count data - DONE
                    ;
                    ;Initialize SIO channel A for async
0010   0E F5              LD      C,CHACTRL     ;C = Channel A control port
0012   2A 23 00           LD      HL,(SIOTBLA)  ;HL => SIO initialize data table
0015   06 0B              LD      B,11          ;B = Number of bytes in table
0017   ED B3              OTIR                  ;Write data table to Channel A
                    ;
                    ;Initialize SIO channel B for SDLC
0019   0E F7              LD      C,CHBCTRL     ;C = Channel B control port
001B   2A 2E 00           LD      HL,(SIOTBLB)  ;HL => SIO initialize data table
001E   06 0C              LD      B,12          ;B = Number of bytes in table
0020   ED B3              OTIR                  ;Write data table to Channel A
0022   C9                 RET
                    ;
0023   10          SIOTBLA DEFB    10H     ;Reset handshake command
0024   30                  DEFB    30H     ;Reset error flags command
0025   18                  DEFB    18H     ;Reset channel command
0026   04                  DEFB    04H     ;Select register 4
0027   47                  DEFB    47H     ;Even par. 1 stop, /16 clk
0028   05                  DEFB    05H     ;Select register 5
0029   AA                  DEFB    0AAH    ;DTR & RTS on, xmit 7 data, on
002A   03                  DEFB    03H     ;Select register 3
002B   41                  DEFB    41H     ;Rcv 7 data bits, on
002C   01                  DEFB    01H     ;Select register 1
002D   00                  DEFB    00H     ;No interrupts
                    ;
002E   30          SIOTBLB DEFB    30H     ;Reset error flags command
002F   18                  DEFB    18H     ;Reset channel command
0030   04                  DEFB    04H     ;Select register 4
0031   20                  DEFB    20H     ;/1 clock, SDLC
0032   03                  DEFB    03H     ;Select register 3
0033   C0                  DEFB    0C0H    ;Rcv 8 data bits, rcv disabled
0034   05                  DEFB    05H     ;Select register 5
0035   E0                  DEFB    0E0H    ;Disable xmit and drivers,
                                            ;xmit 8 data bits
0036   07                  DEFB    07H     ;Select register 7
0037   7E                  DEFB    7EH     ;SDLC flag
0038   01                  DEFB    01H     ;Select register 1
0039   00                  DEFB    00H     ;No interrupts
```

Figure 3-14. SIO Initialization

Operating Modes

The SIO chip can be operated in three modes: polled, interrupt, and block transfer. These modes are generally applicable to low, medium, and high baud rates respectively. However, the most efficient way to use the SIO depends on the particulars of an application. This includes the speed and volume of the data being transmitted, the speed of response that is required, and the type of other processing that must be carried on while communications is occurring.

It should also be remembered that the operating mode of the SIO chip can be changed at any time. It is possible to use the interrupt mode to detect the first received character, receive a block of data in polled or block transfer mode, and return to interrupt mode to await the next message block.

The three operating modes are detailed below.

Polled Operation

In the polled (non-interrupt) mode, the SIO status registers are read by the processor to determine when a character has been received, or if the next character can be transmitted. It is the simplest mode of SIO operation since all of the I/O routine is "in line". Program operation does not depend on the correct initialization of interrupt vectors, or the proper placement of routines at those vectors. This method does require the input routine to be called frequently enough that incoming characters don't overflow the triple buffered receive data register. For this reason polled operation is usually more appropriate for lower baud rates.

Polled operation is appropriate at higher baud rates when the data being transmitted / received is in groups or blocks (rather than occasional unrelated single characters). Depending on the baud rate, the overhead of doing an interrupt for each character can be more of an overhead than simply transmitting / receiving the entire message and then returning to the mainline program. Usually the SIO would be programmed to interrupt on the first received character, process the message in polled mode, and then return to the caller.

Figure 3-15 shows three subroutines for polled operation of channel A in async. RS-232 use. INSTAT (input status) returns a flag indicating whether there is currently a received character waiting to be read from the SIO. CHARIN (character input) returns a received character or waits until one is received. CHAROUT (character output) transmits a character.

Figure 3-16 includes routines for polled operation of channel B in sync. RS-422 use. Transmit and receive routines are listed for handling messages (blocks of data) in SDLC format.

```
                        ;I/O routines for channel A in Async. operation.
                        ;
        F4 00           SIO     EQU     0F4H    ;SIO base address
        F4 00           CHADATA EQU     SIO+0   ;SIO CHA data
        F5 00           CHACTRL EQU     SIO+1   ;SIO CHA control
        F6 00           CHBDATA EQU     SIO+2   ;SIO CHB data
        F7 00           CHBCTRL EQU     SIO+3   ;SIO CHB control
                        ;
                        ;Input status routine, checks SIO to see if a character is
                        ;available.  If available, returns A = FFH (Z = 0).  If not
                        ;available, returns A = 00H (Z = 1).
0200    DB F5           INSTAT  IN      A,(CHACTRL)     ;Read status
0202    CB 47                   BIT     0,A             ;Bit 0 = data available
0204    28 04                   JR      Z,INSTAT1       ;No data available
0206    3E FF                   LD      A,0FFH          ;Data available, set A = FFH
0208    A7                      AND     A               ;Z = 0
0209    C9                      RET
020A    AF              INSTAT1 XOR     A               ;A = 0, Z = 1
020B    C9                      RET
                        ;
                        ;
                        ;Character input routine, waits for data and returns
                        ;character in A.
020C    DB F5           CHARIN  IN      A,(CHACTRL)     ;Read status
020E    CB 47                   BIT     0,A             ;Bit 0 = data available
0210    28 FA                   JR      Z,CHARIN        ;No data, wait
0212    DB F4                   IN      A,(CHADATA)     ;Read data
0214    E6 7F                   AND     7FH             ;Mask bit 7 (Junk)
0216    C9                      RET
                        ;
                        ;
                        ;Character output routine, checks CTS line and xmits
                        ;character in C when CTS is active (RS232 > +3 Volts)
0217    3E 10           CHAROUT LD      A,10H           ;SIO handshake reset data
0219    D3 F5                   OUT     (CHACTRL),A     ;Update handshake register
021B    DB F5                   IN      A,(CHACTRL)     ;Read status
021D    CB 6F                   BIT     5,A             ;Check CTS bit
021F    28 F6                   JR      Z,CHAROUT       ;Wait until CTS is active
0221    DB F5           COUT1   IN      A,(CHACTRL)     ;Read status
0223    CB 57                   BIT     2,A             ;Xmit buffer empty?
0225    28 FA                   JR      Z,COUT1         ;Wait until buffer is empty
0227    79                      LD      A,C             ;Character to A
0228    D3 F4                   OUT     (CHADATA),A     ;Output data
022A    C9                      RET
```

Figure 3-15. Polled Async. RS-232 Example

```
                        ;I/O routines for channel B in SDLC (sync.) operation.
                        ;
        F4 00           SIO     EQU     0F4H    ;SIO base address
        F4 00           CHADATA EQU     SIO+0   ;SIO CHA data
        F5 00           CHACTRL EQU     SIO+1   ;SIO CHA control
        F6 00           CHBDATA EQU     SIO+2   ;SIO CHB data
        F7 00           CHBCTRL EQU     SIO+3   ;SIO CHB control
                        ;
                        ;
                        ;SDLC receive routine, puts data in block pointed to by HL,
                        ;number of bytes (including CRC) is returned in register B,
                        ;returns Z = 1 if data good or Z = 0 for a CRC error
022B    3E 40           SDLCIN  LD      A,40H           ;Reset receive CRC
022D    D3 F7                   OUT     (CHBCTRL),A     ;
022F    3E 03                   LD      A,03H           ;Point to register 3
0231    D3 F7                   OUT     (CHBCTRL),A     ;
0233    3E D9                   LD      A,0D9H          ;Reg 3 data to enable receive
0235    D3 F7                   OUT     (CHBCTRL),A     ;and enter hunt mode
0237    06 00                   LD      B,00H           ;Zero byte counter
0239    DB F7           SDLCIN1 IN      A,(CHBCTRL)     ;Read status
023B    CB 47                   BIT     0,A             ;Bit 0 = data available
023D    28 FA                   JR      Z,SDLCIN1       ;No data, wait
023F    DB F6                   IN      A,(CHBDATA)     ;Read data
0241    77                      LD      (HL),A          ;Write to data block
0242    04                      INC     B               ;Increment counter
0243    23                      INC     HL              ;Update pointer
0244    3E 01                   LD      A,01H           ;Point to status 1
0246    D3 F7                   OUT     (CHBCTRL),A     ;
0248    DB F7                   IN      A,(CHBCTRL)     ;Read status 1
024A    CB 7F                   BIT     7,A             ;Check for frame end
024C    28 EB                   JR      Z,SDLCIN1       ;Loop if not
024E    CB 77                   BIT     6,A             ;Z=0=CRC error, Z=1=Data good
0250    3E 03                   LD      A,03H           ;Point to register 3
0252    D3 F7                   OUT     (CHBCTRL),A     ;
0254    3E C0                   LD      A,0C0H          ;Reg 3 data to disable receive
0256    D3 F7                   OUT     (CHBCTRL),A     ;
0258    C9                      RET
                        ;
                        ;
```

(continued)

```
                        ;SDLC xmit routine, transmits block of data pointed to by HL,
                        ;number of bytes enters in register B
0259  3E 05   SDLCOUT LD      A,05H           ;Point to register 5
025B  D3 F7           OUT     (CHBCTRL),A     ;
025D  3E 69           LD      A,69H           ;Reg 5 data to enable xmit
025F  D3 F7           OUT     (CHBCTRL),A     ;and drivers
0261  3E 80           LD      A,80H           ;Reset xmit CRC data
0263  D3 F7           OUT     (CHBCTRL),A     ;
0265  DB F7   SOUT1   IN      A,(CHBCTRL)     ;Read status
0267  CB 57           BIT     2,A             ;Xmit buffer empty?
0269  28 FA           JR      Z,SOUT1         ;Wait until buffer is empty
026B  7E              LD      A,(HL)          ;Get char
026C  D3 F6           OUT     (CHBDATA),A     ;Xmit first data
026E  3E C0           LD      A,0C0H          ;Reset xmit underrun/EOM latch
0270  D3 F7           OUT     (CHBCTRL),A     ;
0272  23      SOUT2   INC     HL              ;Update pointer
0273  05              DEC     B               ;And counter
0274  28 0B           JR      Z,SOUT4         ;Done, exit
0276  DB F7   SOUT3   IN      A,(CHBCTRL)     ;Read status
0278  CB 57           BIT     2,A             ;Xmit buffer empty?
027A  28 FA           JR      Z,SOUT3         ;Wait until buffer is empty
027C  7E              LD      A,(HL)          ;Get char
027D  D3 F6           OUT     (CHBDATA),A     ;Xmit data
027F  18 F1           JR      SOUT2           ;Loop
0281  3E 05   SOUT4   LD      A,05H           ;Point to register 5
0283  D3 F7           OUT     (CHBCTRL),A     ;
0285  3E E0           LD      A,0E0H          ;Disable xmit and drivers
0287  D3 F7           OUT     (CHBCTRL),A     ;
0289  C9              RET
```

Figure 3-16. Polled Sync. RS-422 Example

Interrupt Operation

The most common mode of SIO interrupt operation is with the Z80 interrupt mode 2. In this mode the vector (supplied by the interrupt source) is combined with the contents of the Z80 interrupt register to form a 16 bit vector address of the interrupt service routine.

Since the CTC can only provide Mode 2 type interrupts, you must use interrupt Mode 2 if interrupts are needed from any of the timer/counters.

When using Mode 2 interrupts you must: Set the value of the Z80 interrupt vector (for the upper byte of the interrupt vector), set the interrupt vector in the SIO (for the lower byte of the interrupt vector- the lowest bit of this vector must be 0), and always use the Z80 Return From Interrupt command at the end of the interrupt service routine (to re-enable the priority chain in the SIO).

Normally the vectors for the CTC and SIO are selected so that a single continuous jump table can be used. Programming the SIO with a vector of

XO hex, and the CTC with a vector of (X+1)0 hex will result in a jump table that is 23 bytes long (40 to 46, 50 to 56, etc.).

The on-board priorities of the CTC and SIO interrupts are preset and cannot be altered. They are listed, along with the vectors they generate, in Figure 3-17 below.

Priority	Description	Vector
1	CTC Channel 0	XXXX X000
2	CTC Channel 1	XXXX X010
3	CTC Channel 2	XXXX X100
4	CTC Channel 3	XXXX X110
	SIO Chan. A	
5	RCV char.	XXXX 110X
5	Special RCV cond.	XXXX 111X
6	XMT buffer empty	XXXX 100X
7	Handshake transition	XXXX 101X
	SIO Chan. B	
8	RCV char.	XXXX 010X
8	Special RCV cond.	XXXX 011X
9	XMT buffer empty	XXXX 000X
10 (lowest)	Handshake transition	XXXX 001X

Figure 3-17. On-Board Interrupt Vector Priorities

Other Interrupt Modes

Other interrupt modes can be used in special situations as noted below.

- Other cards in the system generate interrupts with 8080 type restart vectors only.

Use Z80 interrupt mode 0 and program the SIO with a restart instruction for the vector. Do not use the Status Affects Vector mode. Use the Z80 Return From Interrupt command after servicing the SIO.

- Other cards in the system generate interrupts but do not provide vectors.

Use Z80 interrupt mode 1. Do not use the SIO's Status Affects Vector mode. Upon interrupt check the possible sources of the interrupt. Check the Interrupt Pending bit in Status 0 of the SIO. If set, check the other flags in Status 0 to determine the reason. Use the Z80 Return From Interrupt command (or the Return From Interrupt command in Control 0) after servicing the SIO. This re-enables the SIO's priority chain.

- Interrupts are not used at all.

The SIO interrupt vectors can be used if desired, even if interrupts are not actually generated to the bus. Enable the SIO's Status Affects Vector mode and monitor the Interrupt Pending bit in Status 0. When it is set, read the vector (from Status 2) and use the vector to determine what service is needed. Use the Return From Interrupt command in Control 0 after servicing the SIO. This re-enables the priority chain in the SIO (necessary only in the Status Affect Vector mode).

Block Transfer Mode

The SIO can also be used in a block transfer mode. This mode is normally used where large blocks of data must be received/transmitted at very high speeds.

Block transfers can occur between the SIO and the CPU, or the SIO and a DMA controller. The transfers are controlled by the Wait/Ready lines (for channels A and B). These lines are accessible at connector J4 (along with the CTC signals), or they can be jumpered to the system WAITRQ* bus signal with jumper block V1.

For CPU block transfers the Wait/Ready line(s) must be connected to the STD BUS WAITRQ* line with jumper V1a or V1b. The Wait/Ready line is programmed for the Wait mode (via control register 1) and left enabled while block transfers may occur. The Wait line becomes active during certain reads or writes to the SIO and causes the CPU to wait until the SIO can supply or accept the requested data. The Z80 block I/O transfer instructions are used to transfer blocks of up to 256 bytes of data to/from the SIO. The Interrupt On First Character mode should be used when receiving in the block mode.

For DMA block transfers, a Wait/Ready line can be connected directly to a DMA controller. After this line is initialized for the Ready mode it will go active whenever the SIO is ready to transfer another byte.

RS-232 Handshake Signals

When SIO channel A is configured for RS-232 operation, there are several handshake lines which are used to control the flow of data from/to the serial port. Figure 3-18 lists these handshake lines.

No handshake lines are used with RS-422 operation.

Signal	Direction	Use
CTS	<-- (IN)	Transmit handshake.
DTR	--> (OUT)	Transmit equipment status.
RTS	--> (OUT)	Receiver handshake.
DCD	<-- (IN)	Receiver equipment status.

Figure 3-18. RS-232 Handshake Lines

Normally the data handshake lines (CTS and RTS) are used to signal readiness to accept data. The equipment status signals (DTR and DCD) are used to indicate whether an operating (powered on) device is connected to a port.

The SIO can be programmed for the automatic handshake mode using control register 3, bit D5. In this mode the transmitter and receiver are automatically enabled/disabled by the CTS and DCD signals.

If the automatic handshake mode is not used, the CTS and DCD pins should be monitored by the processor in the Status 0 register. They can also generate interrupts if handshake interrupts are enabled.

The DTR and RTS output lines are controlled by writing to the Control 5 register.

Section 4
REFERENCE

SPECIFICATIONS

VL-7842 Multifunction Z80 CPU Card

Size: Meets all STD BUS mechanical specifications.

Storage Temperature: -40° to +75° C.
Free Air Operating Temperature: 0° to +65° C.

Memory Sockets: Four 24/28 pin JEDEC compatible.

Power Requirements:
5V ±5% at 750 ma typ. (without on-board memory).
±12V ±10% at 18 ma typ. (if RS-232 used).

Wait State Control: Jumper selectable, op code cycle only.

Reset Pulse: 35 ms typ., op code sychronized.

Memory Map: 16 selectable maps (7 standard, 9 user programmable).

I/O Map: Standard (F0-F7,and FE) or custom I/O mapping.

Clock Speed Selection Chart

CPU Clock Speed	RAM/ROM Access Speed (ns)*	Standard Async. Baud Rates	Max. Async. Baud Rate	Max. Sync. Baud Rate
2.5 MHz	600/450	75-38,400	38.4K	312.5K
3.6864 MHz	350/250	75-38,400	57.6K	460.8K
4.0 MHz	350/250	75-4,800	62.5K	500K
6.0 MHz	200/150	75-19,200	93.7K	750K

* shown with/without wait cycle.

JUMPER OPTIONS

Jumper Block	Description	As Shipped
M0-M3	Memory socket type configuration.	4-8K RAMs/ROMs
J5	Memory map select/ Segment signal connector.	IN
V1	a - SIO chan. A ready/wait line to WAITRQ*. b - SIO chan. B ready/wait line to WAITRQ*.	out out
V2	Baud rate clock options. See Baud Rate Clock. a - Chan. B RECV clock from RS-422 line. b - Chan. B RECV clock from chan. B XMIT. c - Chan. B XMIT clock from CTC chan. 1. d - Chan. B XMIT clock from CTC chan. 0. e - SYSCLK/2 to CTC chan. 1 input. f - Chan. A RECV clock from CTC chan. 0. g - Chan. A RECV clock from extrnl. line.	 a - out b - out c - IN c - out d - IN e - IN g - out
V3-V7	Serial I/O port options. See Serial I/O Ports.	
V8-V9	Channel A multidrop mode	out
V10	a - Channel A RS-232 input. b - Channel A RS-422 input.	a - IN b - out
V11-V12	Channel B multidrop mode	out
V13	a - RS-232 received clock to jumper V2g. b - RS-422 received clock to jumper V2g.	a - out b - IN
V14	Memory map selection. See Memory Map.	Map #0
V15	Memory socket enable/disable. a - Socket M3 enabled. b - Socket M2 enabled. c - Socket M1 enabled. d - Socket M0 enabled.	 a - IN b - IN c - IN d - IN
V16	a - MEMEX is set high at power-up.[1] b - MEMEX is set low at power-up. c - WAIT state enabled.	a - out b - IN c - out[2]
V17	a - MEMEX signal controlled on-board.	a - IN
V18	a - IOEXP connected to ground. b - AUX GND connected to digital ground.	a - IN b - IN

Notes:
1) Also inverts data written to the MEMEX control port.
2) IN on 6 MHz versions.

Jumper Functions

I/O MAP

I/O Address	Port Type	Description
F0-F3	I/O	CTC Channel 0-3
F4-F5	I/O	SIO Channel A
F6-F7	I/O	SIO Channel B
FE	OUT	MEMEX Control

I/O Port Mapping

INTERRUPT VECTORS

Priority	Description	Vector
1	CTC Channel 0	XXXX X000
2	CTC Channel 1	XXXX X010
3	CTC Channel 2	XXXX X100
4	CTC Channel 3	XXXX X110
	SIO Chan. A	
5	RCV char.	XXXX 110X
5	Special RCV cond.	XXXX 111X
6	XMT buffer empty	XXXX 100X
7	Handshake transition	XXXX 101X
	SIO Chan. B	
8	RCV char.	XXXX 010X
8	Special RCV cond.	XXXX 011X
9	XMT buffer empty	XXXX 000X
10 (lowest)	Handshake transition	XXXX 001X

On-Board Interrupt Vector Priorities

CTC INFORMATION

I/O Port	Write (OUT)	Read (IN)
F0	Chan. 0 Control, Load & Vector	Chan. 0 Current Count
F1	Chan. 1 Control and Load	Chan. 1 Current Count
F2	Chan. 2 Control and Load	Chan. 2 Current Count
F3	Chan. 3 Control and Load	Chan. 3 Current Count

CTC Register Addresses

CTC CONTROL REGISTER

D7	D6	D5	D4	D3	D2	D1	D0
ENABLE INTPS.	COUNTER/ TIMER	CLOCK DIVIDER	CLOCK EDGE	TRIGGER	LOAD REG.	CHANNEL RESET	VECTOR LOAD

Bit	Description
D7	1=Enable Interrupts
D6	0=Timer Mode 1=Counter Mode
D5#	0=Clock divided by 16 1=Clock divided by 256
D4	0=External trigger is falling edge sensitive 1=External trigger is rising edge sensitive
D3#	0=Timer starts when loaded 1=Timer starts from external trigger
D2	1=Next byte to this channel goes to the LOAD register
D1	1=Reset (stop) counter (until registers are reloaded)
D0	0=Vector register write (illegal except to channel 0) 1=Normal register access (must be 1 except for channel 0)

Timer mode only.

CTC Control Register

SIO INFORMATION

I/O Port	Write (OUT)	Read (IN)
F4	Chan. A XMT data	Chan. A RCV data
F5	Chan. A Control Reg. 0	Chan. A Status Reg. 0
F6	Chan. B XMT data	Chan. B RCV data
F7	Chan. B Control Reg. 0	Chan. B Status Reg. 0

SIO I/O Port Locations

Name	Use	Access
XMT	Data to transmit	I/O port
Control 0	Commands and other register access	I/O port
Control 1	Interrupt control	Via Control 0
Control 2	Interrupt vector set (chan B only)	Via Control 0
Control 3	Receiver control	Via Control 0
Control 4	Clock and word control	Via Control 0
Control 5	Transmitter control	Via Control 0
Control 6	Sync character or SDLC address	Via Control 0
Control 7	Sync character or SDLC flag	Via Control 0
RCV	Data received	I/O port
Status 0	Ready/busy bits	I/O port
Status 1	Receive error flags	Via Control 0
Status 2	Interrupt vector read (chan B only)	Via Control 0

SIO Registers

REGISTER	D7	D6	D5	D4	D3	D2	D1	D0
CNTL 0	--CRC RESET[s]--			------COMMANDS[e]----			---REGISTER SELECT---	
CNTL 1	WT/RDY ENABLE	±READY SELECT	WT/RDY ON RCV	RCV INT MODE[p]		VECTOR MODE[b,q]	XMT INT	HNDSHK INT
CNTL 2	----------INTERRUPT VECTOR (WRITE)[b]----------							
CNTL 3	---RCV CHAR--- LENGTH[k]		AUTO HNDSHK	START HUNT	RCV CRC ENABLE	ADDR MATCH	SYNC STRIP	RCV ENABLE
CNTL 4	CLOCK DIVIDER[m]		--SYNC MODE[t]--		-STOP BITS[c]		EVEN PARITY	PARITY ENABLE
CNTL 5	DTR CNTL	---XMT CHAR--- LENGTH[k]		SEND BREAK	XMTR ENABLE	SDLC/ CRC-16	RTS CNTL	XMT CRC ENABLE
CNTL 6	----------SYNC BIT 7 - 0 (OR SDLC ADDRESS FIELD)----------							
CNTL 7	----------SYNC BIT 15 - 8 (OR SDLC FLAG CHARACTER)----------							

[a] Channel A only.
[b] Channel B only.

[e] 000 No operation
001 Send abort (SDLC)
010 Reset handshake INT
011 Channel reset
100 Enable INT on next char RCVD
101 Reset XMT INT
110 Error reset
111 Return from INT[b]

[s] 00 No operation
01 Reset RCV CRC checker
10 Reset XMT CRC generator
11 Reset XMT Underrun/EOM latch

[p] 00 RCVR INT disable
01 RCVR INT on first char.
10 RCVR INT on every char., parity affects vector
11 RCVR INT on every char., parity doesn't affect vector

[k] 00 5 bits/char.
01 7 bits/char.
10 6 bits/char.
11 8 bits/char.

[m] 00 Clock divided by 1
01 Clock divided by 16
10 Clock divided by 32
11 Clock divided by 64

[t] 00 8 bit programmed sync
01 16 bit programmed sync
10 SDLC mode (01111110 flag)
11 External sync mode (not used)

[c] 00 Sync mode
01 1 stop bit
10 1 1/2 stop bits
11 2 stop bits

[q] INT vector will be modified as follows if D2 in CNTL 1 is set.

D3	D2	D1[b]	
0	0	0	Chan B XMT buffer empty
0	0	1	Chan B Handshake change
0	1	0	Chan B RCVD char. ready
0	1	1	Chan B Special RCV condition
1	0	0	Chan A XMT buffer empty
1	0	1	Chan A Handshake change
1	1	0	Chan A RCVD char. ready
1	1	1	Chan A Special RCV condition

SIO Control Register Functions

REGISTER	D7	D6	D5	D4	D3	D2	D1	D0
STAT 0	LINE[r] BREAK	XMIT EOM	CTS[r]	SYNC/ HUNT	DCD[r]	XMT BUF. EMPTY	INT[a] PEND.	CHAR RCVD
STAT 1	FRAME END	CRC/FRM ERROR	OVERRUN ERROR	PARITY ERROR	--SDLC RESIDUE CODE[s]--			XMTR DONE
STAT 2	----------INTERRUPT VECTOR (READ)[b,q]----------							

[a] Channel A only.
[b] Channel B only.

[r] A "handshake interrupt" signal

[q] INT vector will be modified as follows if D2 in CNTL 1 is set.

D3	D2	D1[b]	
0	0	0	Chan B XMT buffer empty
0	0	1	Chan B Handshake change
0	1	0	Chan B RCVD char. ready
0	1	1	Chan B Special RCV condition
1	0	0	Chan A XMT buffer empty
1	0	1	Chan A Handshake change
1	1	0	Chan A RCVD char. ready
1	1	1	Chan A Special RCV condition

[s] The SDLC residue codes are:

D3	D2	D1	I field in prev. byte	I field in 2nd prev. byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

SIO Status Register Functions

BAUD RATE TABLES

The data required to program the CTC and SIO for common baud rates is shown below. The CTC count for any baud rate can also be determined with the following formulas (7842 card only):

CTC in Counter Mode: Baud rate = system clock/4/SIO divider/count
CTC in Timer Mode: Baud rate = system clock/32/SIO divider/count

2.5 MHz Clock

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
312.5K	1	02	02	Counter	1	0
125.0K	1	05	05	Counter	1	0
62.5K	1	0A	10	Counter	1	0
38.4K	1	10	16	Counter	1	+1.72
19.2K	1	21	33	Counter	1	-1.36
9600	1	41	65	Counter	1	+0.16
4800	1	82	130	Counter	1	+0.16

Synchronous Baud Rate Programming for 2.5 MHz 7842 Boards

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
38.4K	1	01	1	Counter	16	+1.72
19.2K	1	02	2	Counter	16	+1.72
9600	1	04	4	Counter	16	+1.72
4800	1	08	8	Counter	16	+1.72
2400	1	10	16	Counter	16	+1.72
1800	1	16	22	Counter	16	-1.36
1200	1	21	33	Counter	16	-1.36
600	1	41	65	Counter	16	+0.16
300	16	10	16	Timer	16	+1.72
150	16	21	33	Timer	16	+1.36
110	16	2C	44	Timer	16	+0.38
75	16	41	65	Timer	16	+0.16

Asynchronous Baud Rate Programming for 2.5 MHz 7842 Boards

3.6864 MHz Clock

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
460.8K	1	02	2	Counter	1	0
307.2K	1	03	3	Counter	1	0
153.6K	1	06	6	Counter	1	0
76.8K	1	0C	12	Counter	1	0
57.6K	1	10	16	Counter	1	0
38.4K	1	03	3	Counter	1	0
19.2K	1	06	6	Counter	1	0
9600	1	0C	12	Counter	1	0
4800	1	18	24	Counter	1	0

Synchronous Baud Rate Programming for 3.6864 MHz 7842 Boards

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
19.2K	1	03	0	Counter	16	0
9600	1	06	6	Counter	16	0
4800	1	0C	12	Counter	16	0
2400	1	18	24	Counter	16	0
1800	1	20	32	Counter	16	0
1200	1	30	48	Counter	16	0
600	1	60	96	Counter	16	0
300	16	18	24	Timer	16	0
150	16	30	48	Timer	16	0
110	16	41	65	Timer	16	+0.07
75	16	60	96	Timer	16	0

Asynchronous Baud Rate Programming for 3.6864 MHz 7842 Boards

4.0 MHz Clock

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
500K	1	02	2	Counter	1	0
200K	1	05	5	Counter	1	0
100K	1	0A	10	Counter	1	0
62.5K	1	10	16	Counter	1	0
50K	1	14	20	Counter	1	0
25K	1	28	40	Counter	1	0
20K	1	32	50	Counter	1	0
19.2K	1	34	52	Counter	1	+0.16
9600	1	68	104	Counter	1	+0.16
4800	1	D0	208	Counter	1	+0.16

Synchronous Baud Rate Programming for 4.0 MHz 7842 Boards

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
4800	1	0D	13	Counter	16	+0.16
2400	1	1A	26	Counter	16	+0.16
1800	1	23	35	Counter	16	-0.79
1200	1	34	52	Counter	16	+0.16
600	1	68	104	Counter	16	+0.16
300	16	1A	26	Timer	16	+0.16
150	16	34	52	Timer	16	+0.16
110	16	47	71	Timer	16	+0.03
75	16	68	104	Timer	16	+0.16

Asynchronous Baud Rate Programming for 4.0 MHz 7842 Boards

6.0 MHz Clock

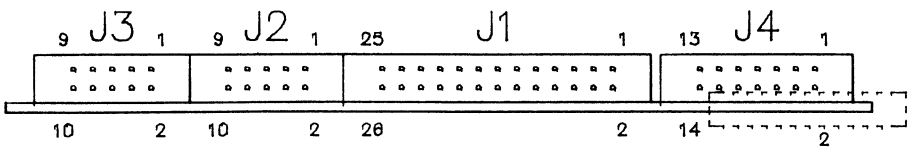
Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
750K	1	02	2	Counter	1	0
500K	1	03	3	Counter	1	0
250K	1	06	6	Counter	1	0
100K	1	0F	15	Counter	1	0
50K	1	1E	30	Counter	1	0
25K	1	3C	60	Counter	1	0
20K	1	4B	75	Counter	1	0
19.2K	1	4E	78	Counter	1	+0.16
9600	1	9C	156	Counter	1	+0.16

Synchronous Baud Rate Programming for 6 MHz 7842 Boards

Baud Rate	CTC Divider	--CTC Count-- Hex	Decimal	CTC Mode	SIO Divider	Baud Rate Error (%)
19200	1	05	5	Counter	16	-2.34
9600	1	0A	10	Counter	16	-2.34
4800	1	14	20	Counter	16	-2.34
2400	1	27	39	Counter	16	0.16
1800	1	34	52	Counter	16	0.16
1200	1	4E	78	Counter	16	0.16
600	1	9C	156	Counter	16	0.16
3009	16	27	39	Timer	16	0.16
150	16	4E	78	Timer	16	0.16
110	16	6B	107	Timer	16	-0.44
75	16	9C	156	Timer	16	0.16

Asynchronous Baud Rate Programming for 6 MHz 7842 Boards

CONNECTOR PINOUTS



I/O Connector Physical Pin Locations

J1 Pin	Signal Name	RS-232-C Pin
1	-	1
2	-	14
3	TD	2
4	(RCLK)	15
5	RD	3
6	-	16
7	RTS	4
8	RCLK	17
9	CTS	5
10	-	18
11	DSR	6
12	-	19
13	GND	7
14	DTR	20
15	DCD	8
16	-	21
17	-	9
18	-	22
19	-	10
20	-	23
21	-	11
22	TCLK	24
23	-	12
24	-	25
25	-	13
26	-	-

Note: The J1 connector can be converted to the RS-232-C pinout using VersaLogic cable #9560.

RS-232 Serial Port Connector (J1) Pinout

J2 / J3 Pin	Signal Name	Direction
1	TXC+	OUT
2	TXC-	OUT
3	TXD+	OUT
4	TXD-	OUT
5	SGND	-
6	SGND	-
7	RXD-	IN
8	RXD+	IN
9	RXC-	IN
10	RXC+	IN

RS-422 Serial Port Connector (J2 and J3) Pinout

J4 Pin	Signal Name	Input Load (Sink ma)	Output Drive (Sink ma)
1	GND		
2	CLOCK/INT 3	1	
3	GND		
4	CLOCK/INT 2	1	
5	GND		
6	COUNT/TIME 2		2
7	GND		
8	CLOCK/INT 1	1	
9	GND		
10	COUNT/TIME 1		2
11	GND		
12	WAIT/READY* A	0.1	2
13	GND		
14	WAIT/READY* B	0.1	2

Counter/Timer Connector J4

J5 Pin	Signal Name	Input Load (Sink ma)
1	GND	
2	SEGMENT	1.3

Segment Connector / Jumper J5

STD BUS Pinout

Connections from the VL-7806/7 to the STD BUS are shown below. Pins 1 and 2 are at the top (card ejector) edge of the board. As noted below the odd numbered pins are on the component side of the board while the even numbered pins are on the solder side. Direction of signal flow is referenced to the VL-7806/7.

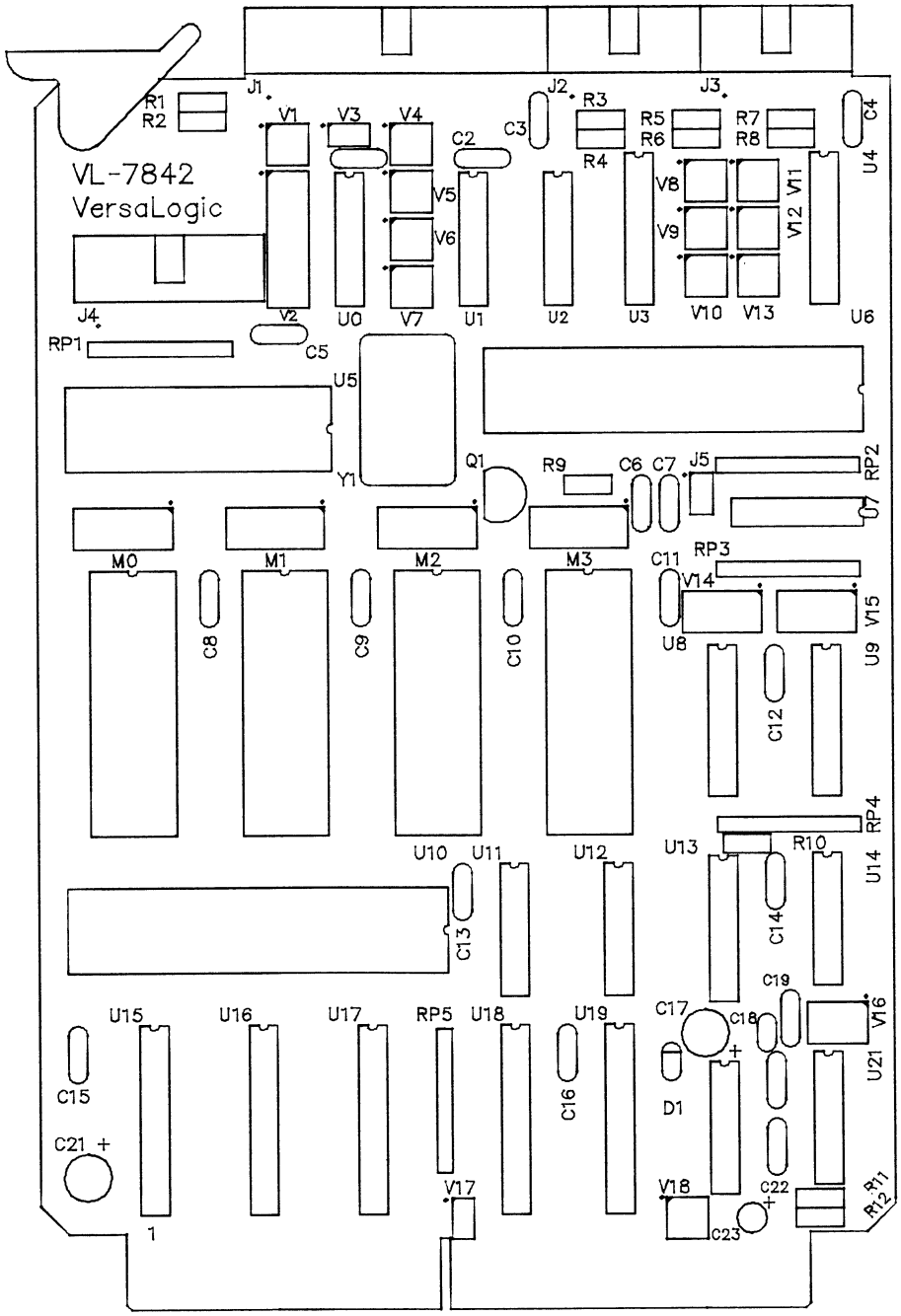
COMPONENT SIDE				SOLDER SIDE			
PIN	SIGNAL	FLOW	DESCRIPTION	PIN	SIGNAL	FLOW	DESCRIPTION
1	+5V	In	+5 volt power	2	+5V	In	+5 volt power
3	GND	In	Digital ground	4	GND	In	Digital ground
5	VBB/VBAT	-	-5V or bat. backup	6	-5V	-	-5V power
7	D3	I/O	Data bus	8	D7	I/O	Data bus
9	D2	I/O	Data bus	10	D6	I/O	Data bus
11	D1	I/O	Data bus	12	D5	I/O	Data bus
13	D0	I/O	Data bus	14	D4	I/O	Data bus
17	A6	Out	Address bus	18	A14	Out	Address bus
19	A5	Out	Address bus	20	A13	Out	Address bus
21	A4	Out	Address bus	22	A12	Out	Address bus
23	A3	Out	Address bus	24	A11	Out	Address bus
25	A2	Out	Address bus	26	A10	Out	Address bus
27	A1	Out	Address bus	28	A9	Out	Address bus
29	A0	Out	Address bus	30	A8	Out	Address bus
31	WR*	Out	Write strobe	32	RD*	Out	Read strobe
33	IORQ*	Out	I/O addr. select	34	MEMRQ*	Out	Memory addr. select
35	IOEXP*	(1)	I/O expansion	36	MEMEX*	(2)	Memory expansion
37	REFRESH*	Out	Refresh timing	38	MCSYNC*	Out	Machine cycle sync.
39	STATUS1*	Out	Z80 M1*	40	STATUSO*	-	CPU status
41	BUSAK*	Out	Bus acknowledge	42	BUSRQ*	In	Bus request
43	INTAK*	Out	Interrupt acknowl.	44	INTRQ*	In	Interrupt request
45	WAITRQ*	In	Wait request	46	NMIRQ*	In	Non-maskable int.
47	SYSRESET*	In	System reset	48	PBRESET*	In	Push button reset
49	CLOCK*	Out	CPU clock	50	CNTRL*	-	AUX timing
51	PCO	Out	Priority chain out	52	PCI	In	Priority chain in
53	AUXGND	(3)	±12 volt ground	54	AUXGND	(3)	±12 volt ground
55	AUX+V	In	+12 volt input	56	AUX-V	In	-12 volt input

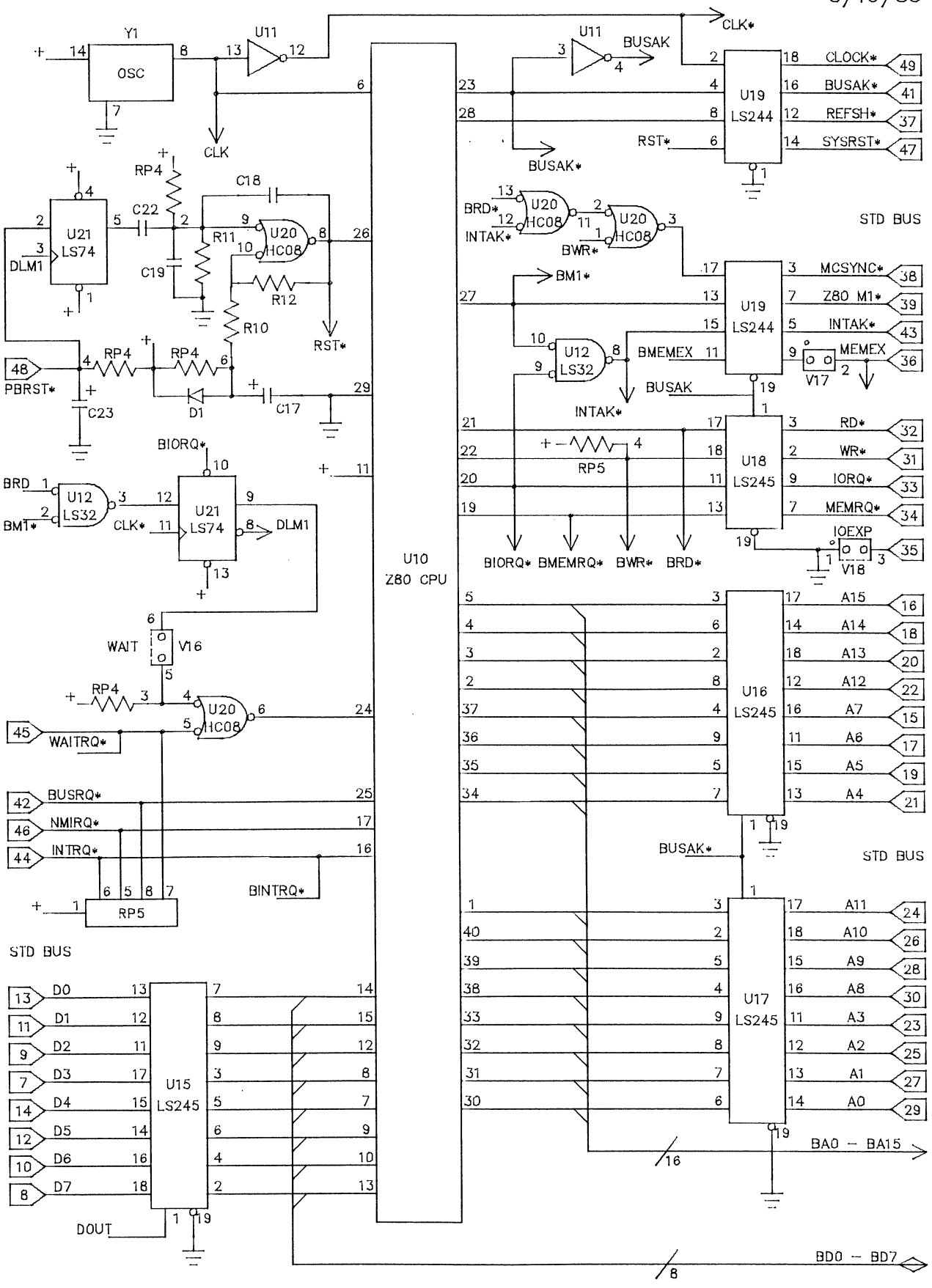
Notes:
* Denotes an active low signal.
1) Jumper option. Normally strapped to ground.
2) Jumper option. Normally an output.
3) Jumper option. Normally strapped to digital ground (pins 3-4).
-) All address, data, and status lines (except 37, 41, 47 and 49) can be driven by another source while BUSAK* is active. DMA can be performed to on-board memory devices.

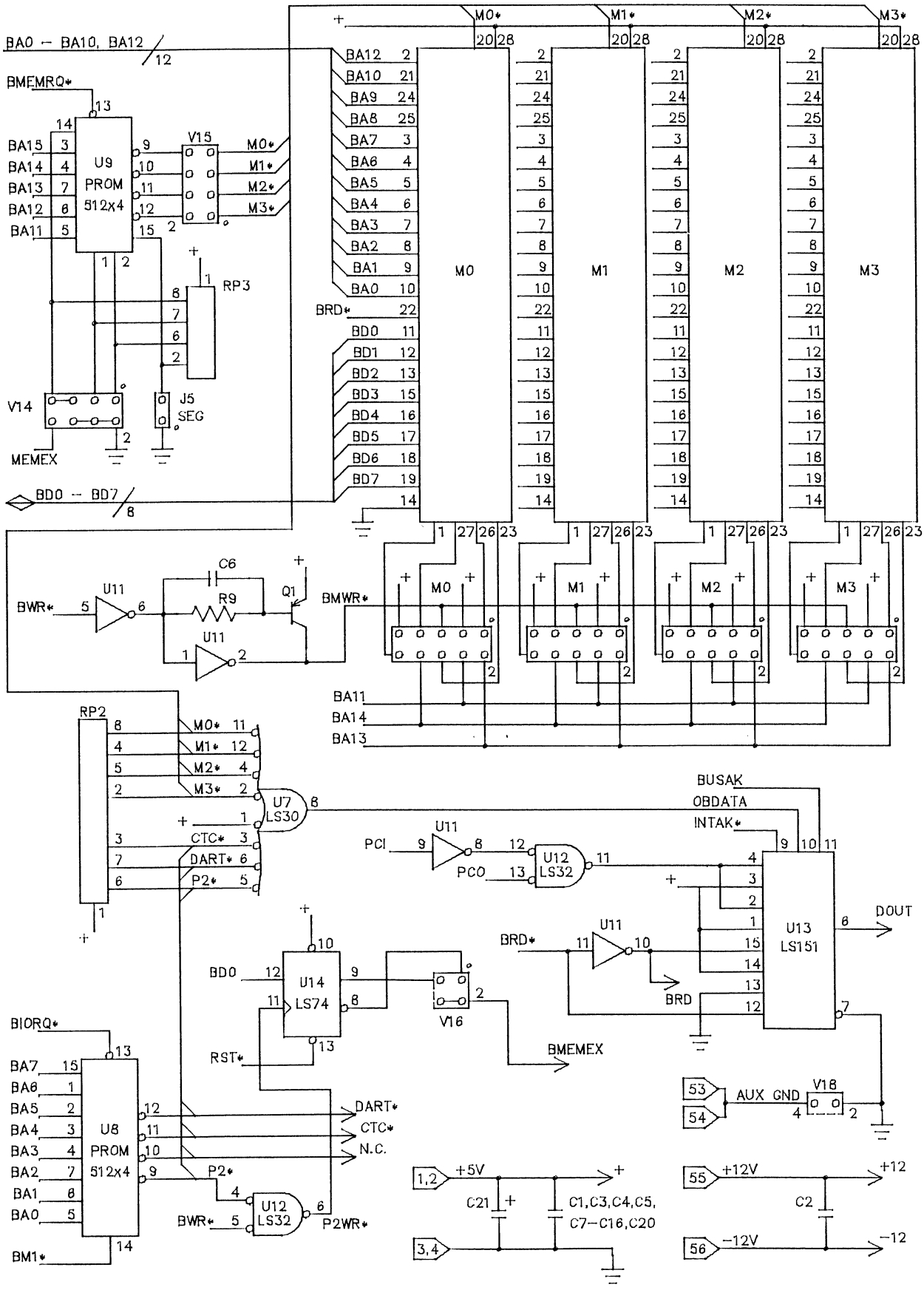
DECIMAL / HEX / ASCII CONVERSION CHART

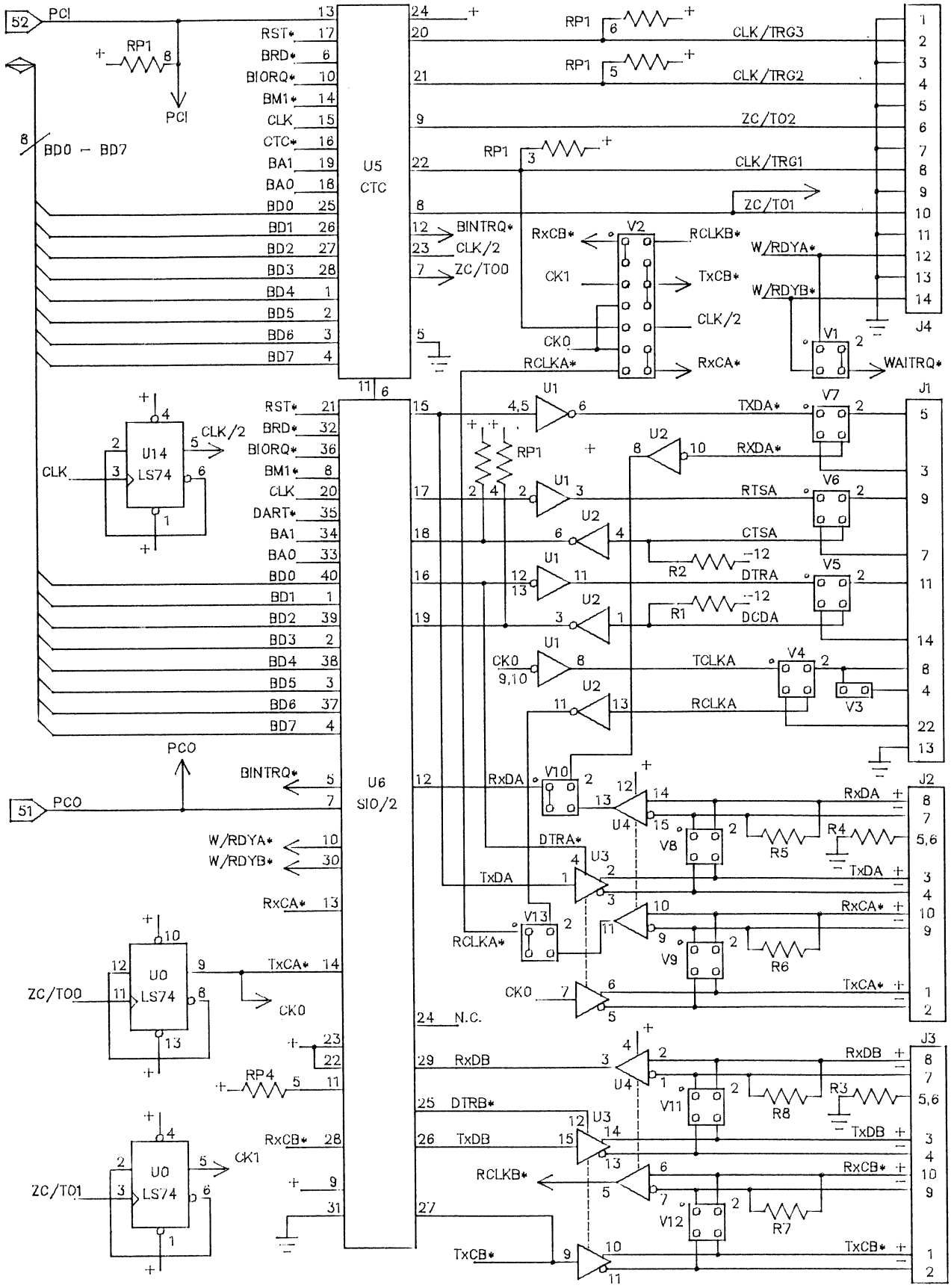
The chart below is useful for both ASCII and decimal/hex conversion. The "^" symbol is used to denote control characters. "^A" represents control A, etc.

Dec.	Hex	ASCII	Dec.	Hex	ASCII	Dec.	Hex	ASCII	Dec.	Hex	ASCII
0	00	NUL	32	20		64	40	@	96	60	`
1	01	^A SOH	33	21	!	65	41	A	97	61	a
2	02	^B STX	34	22	"	66	42	B	98	62	b
3	03	^C ETX	35	23	#	67	43	C	99	63	c
4	04	^D EOT	36	24	$	68	44	D	100	64	d
5	05	^E ENQ	37	25	%	69	45	E	101	65	e
6	06	^F ACK	38	26	&	70	46	F	102	66	f
7	07	^G BEL	39	27	'	71	47	G	103	67	g
8	08	^H BS	40	28	(	72	48	H	104	68	h
9	09	^I HT	41	29	)	73	49	I	105	69	i
10	0A	^J LF	42	2A	*	74	4A	J	106	6A	j
11	0B	^K VT	43	2B	+	75	4B	K	107	6B	k
12	0C	^L FF	44	2C	,	76	4C	L	108	6C	l
13	0D	^M CR	45	2D	-	77	4D	M	109	6D	m
14	0E	^N SO	46	2E	.	78	4E	N	110	6E	n
15	0F	^O SI	47	2F	/	79	4F	O	111	6F	o
16	10	^P DLE	48	30	0	80	50	P	112	70	p
17	11	^Q DC1	49	31	1	81	51	Q	113	71	q
18	12	^R DC2	50	32	2	82	52	R	114	72	r
19	13	^S DC3	51	33	3	83	53	S	115	73	s
20	14	^T DC4	52	34	4	84	54	T	116	74	t
21	15	^U NAK	53	35	5	85	55	U	117	75	u
22	16	^V SYN	54	36	6	86	56	V	118	76	v
23	17	^W ETB	55	37	7	87	57	W	119	77	w
24	18	^X CAN	56	38	8	88	58	X	120	78	x
25	19	^Y EM	57	39	9	89	59	Y	121	79	y
26	1A	^Z SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL


VL-7842
VersaLogic
J1
J2
J3
J4
R1
R2
R3
R4
R5
R6
R7
R8
C1
C2
C3
C4
C5
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
U0
U1
U2
U3
U4
U5
U6
RP1
Y1
Q1
R9
C6
C7
J5
RP2
M0
M1
M2
M3
C8
C9
C10
C11
RP3
V14
V15
U8
U9
C12
RP4
U10
U11
U12
U13
R10
U14
C13
C14
C19
V16
U15
U16
U17
RP5
U18
U19
C17
C18
U21
C15
C16
D1
C21
V17
V18
C22
C23
R12
R11
1


Y1
OSC
U11
CLK
CLK*
RP4
C18
C22
C19
R11
R12
R10
U21
LS74
DLM1
U20
HC08
RST*
PBRST*
C23
D1
C17
BIORQ*
BRD*
BM1*
U12
LS32
WAIT
V16
WAITRQ*
BUSRQ*
NMIRQ*
INTRQ*
BINTRQ*
RP5
U10
Z80 CPU
BUSAK
BUSAK*
RST*
U19
LS244
CLOCK*
BUSAK*
REFSH*
SYSRST*
STD BUS
BRD*
INTAK*
BWR*
BM1*
BMEMEX
MCSYNC*
Z80 M1*
INTAK*
MEMEX
V17
RP5
U18
LS245
RD*
WR*
IORQ*
MEMRQ*
IOEXP
V18
BIORQ*
BMEMRQ*
BWR*
BRD*
U16
LS245
A15
A14
A13
A12
A7
A6
A5
A4
BUSAK*
STD BUS
U17
LS245
A11
A10
A9
A8
A3
A2
A1
A0
BA0 – BA15
16
U15
LS245
D0
D1
D2
D3
D4
D5
D6
D7
DOUT
BD0 – BD7
8


BA0 – BA10, BA12
12
BMEMRQ*
U9
PROM
512x4
BA15
BA14
BA13
BA12
BA11
V15
M0*
M1*
M2*
M3*
RP3
V14
J5
SEG
MEMEX
BD0 – BD7
8
M0
M1
M2
M3
BA12
BA10
BA9
BA8
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BA0
BRD*
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
C6
R9
Q1
U11
BWR*
BMWR*
BA11
BA14
BA13
RP2
U7
LS30
CTC*
DART*
P2*
BUSAK
OBDATA
INTAK*
PCI
PCO
U12
LS32
U13
LS151
DOUT
BRD
U14
LS74
BD0
RST*
V16
BMEMEX
BIORQ*
U8
PROM
512x4
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BA0
BM1*
DART*
CTC*
N.C.
P2*
BWR*
P2WR*
V18
AUX GND
+5V
C21
C1,C3,C4,C5,
C7–C16,C20
+12V
C2
–12V


52 PCI
RP1
PCI
8 BD0 – BD7
U5 CTC
RST* 13
BRD* 17
BIORQ* 6
BM1* 10
CLK 14
CTC* 15
BA1 16
BA0 19
BD0 18
BD1 25
BD2 26
BD3 27
BD4 28
BD5 1
BD6 2
BD7 3
BINTRQ* 12
CLK/2 23
ZC/TO0 7
CLK/TRG3 20
CLK/TRG2 21
ZC/TO2 9
CLK/TRG1 22
ZC/TO1 8
RxCB* V2 RCLKB*
CK1 TxCB*
CLK/2
CK0
RCLKA* RxCA*
W/RDYA*
W/RDYB*
V1 WAITRQ*
J4
U14 LS74
CLK/2
CLK
U6 SIO/2
RST* 21
BRD* 32
BIORQ* 36
BM1* 8
CLK 20
DART* 35
BA1 34
BA0 33
BD0 40
BD1 1
BD2 39
BD3 2
BD4 38
BD5 3
BD6 37
BD7 4
PCO
51 PCO
BINTRQ* 5
7
W/RDYA* 10
W/RDYB* 30
RxCA* 13
U0 LS74
ZC/TO0
TxCA* 14
CK0
RP4
RxCB* 28
CK1
U0 LS74
ZC/TO1
24 N.C.
29 RxDB
25 DTRB*
26 TxDB
27
TxCB*
RCLKB*
U1
U2
RP1
TXDA* V7
RXDA* V6
RTSA
CTSA
R2 12
DTRA V5
R1 12
DCDA
CK0 9,10
TCLKA V4
V3
RCLKA
J1
RxDA V10
U4
V8
R5
R4
RxDA
TxDA
DTRA*
U3
RCLKA* V13
V9
R6
RxCA*
TxCA*
J2
RxDB
U4
V11
R8
R3
TxDB
U3
RCLKB*
V12
R7
RxCB*
TxCB*
J3

VL-7842 PARTS LIST

Capacitors

C1, C2, C3, C4, C5, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C20, C22	.01 uf ceramic disk
C6, C19	680 pf NPO ceramic
C17, C21	22 uf, 25V, elect. radial
C18	270 pf NPO ceramic
C23	2.2 uf 50V elect. radial

Integrated Circuits

U0, U14, U21	LS074
U1	I488
U2	1489A
U3	75174N
U4	75175N
U5	Z80-A CTC, 8430
U6	Z80-A SIO/2, 8442
U7	74LS030
U8	Bipolar PROM "B" (I/O map)
U9	Bipolar PROM "A" (memory map)
U10	Z80-A CPU, 8400
U11	74LS04
U12	74LS32
U13	74LS151
U15, U16, U17, U18	74LS245
U19	74LS244

U20	74HC08

Resistors

R1, R2	22K ohm, 5%, 1/4W
R3, R4, R5, R6, R7, R8	100 ohm, 5%, 1/4W
R9, R12	10K ohm, 5%, 1/4W
R10	1K ohm, 5%, 1/4W
R11	12K ohm, 5%, 1/4W
RP1, RP3, RP4, RP5	4K7 ohm, 7 resistor SIP
RP2	1K ohm, 7 resistor SIP

Semiconductors

D1	1N4148
Q1	2N2907
Y1	Crystal osc. (frequency depends on version)

Miscellaneous

J1	26 pin R/A header
J2, J3	10 pin R/A header
J4	14 pin R/A header