VL-1295
VL-12CT95

High Speed Analog Input
Card for the STD Bus

STD32™

Model VL-1295

High Speed Analog Input Card for the STD Bus

REFERENCE MANUAL

STD32™

VL-1295 Rev. 3.01
VL-12CT95 Rev. 3.01
Doc. Rev. 09/14/94

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.

M1295

Table of Contents

Overview

This manual details the installation and operation of VersaLogic's VL-1295 analog input card. This card interfaces directly with external analog voltages and provides digital readings to the STD Bus with 12-bit accuracy.

Introduction

The VL-1295 analog input card provides high speed, 12-bit A to D conversion for ±5 or ±10 volt inputs. Two input paths are provided: 16 on-board multiplexed single-ended inputs or 1 direct input for externally multiplexed signals. High speed operation is enhanced with a 16-bit data bus mode, DMA handshaking, automatic channel incrementing, and interrupt support.

Each input channel can be read as desired by the system CPU. The board is capable of 333,000 single-channel readings per second, or 200,000 scanned-channel readings per second.

Features

- 16 single-ended input channels.
- 12-Bit resolution (4096 counts).
- 3 microsecond conversion time.
- ±5 Volt or ±10 Volt input ranges.
- DMA and interrupt capabilities.
- Auto channel increment mode.
- IOEXP supported.
- STD 32 compatible (16-bit data path).
- STD 80 and STD Z80 compatible (8-bit data path).
- +5 Volt single supply operation.
- Extended temperature version available.
- Software compatible with Analog Devices RTI-1265.

Specifications

Input Channels: 16 single-ended multiplexed, 1 external

Input Range: ±5V or ±10V

Resolution: 12 Bits (4096 counts)

Conversion Time: 3 μs + settling time

Settling Time: 2 μs, 20 μs, or 100 μs (selectable)

Throughput:

- 333,000 readings/sec (any one channel repeated read)
- 200,000 readings/sec (scanned channel reading)

Input Overvoltage Protection:

- ±35V with power on (channel inputs)
- ±20V with power off (channel inputs)
- ±20V with power on (AIN2 input)
- ±10V with power off (AIN2 input)

Input Impedance: $1 \times 10^8\ \Omega$ min.

Data Format: Binary or two's complement

Linearity: ±½ LSB – over temperature range

Differential Linearity: ±¾ LSB – over temperature range

Temperature Coefficient:

- Gain: ±25 ppm/° C of FSR
- Offset: ±10 ppm/° C of FSR

Addressing: I/O, 8, 10, or 16 bits plus IOEXP

Mapping: 16-byte block on any 16-byte boundary

Size: Meets all STD 32 Bus mechanical specifications

Storage Temperature: −40° to +85° C

Free Air Operating Temperature:

- VL-1295: 0° to +65° C
- VL-12CT95: −25° to +85° C

Power Requirements:

- VL-1295: 5V ±5% at 300 ma typ.
- VL-12CT95: 5V ±5% at 300 ma typ.

Bus Compatibility:

- STD 80: Full compliance, all bus speeds
- STD Z80: Full compliance, all bus speeds
- STD 32: I/O slave, SA16, SA8-I, IXP, SDMA8, SDMA16

Configuration

Jumper Options

Various options available on the VL-1295 card are selected using removable jumper blocks (shorting plugs). Features are selected or deselected by installing or removing the jumpers as noted. The terms "In" or "Jumpered" are used to indicate an installed plug: "Out" or "Open" are used to indicate a removed plug.

Figure 2-1 shows the jumper block locations on the VL-1295 card. It indicates the position of the jumpers as shipped from the factory. The function of each jumper block is detailed in Figure 2-2.

VL-1295 Jumper Block Locations

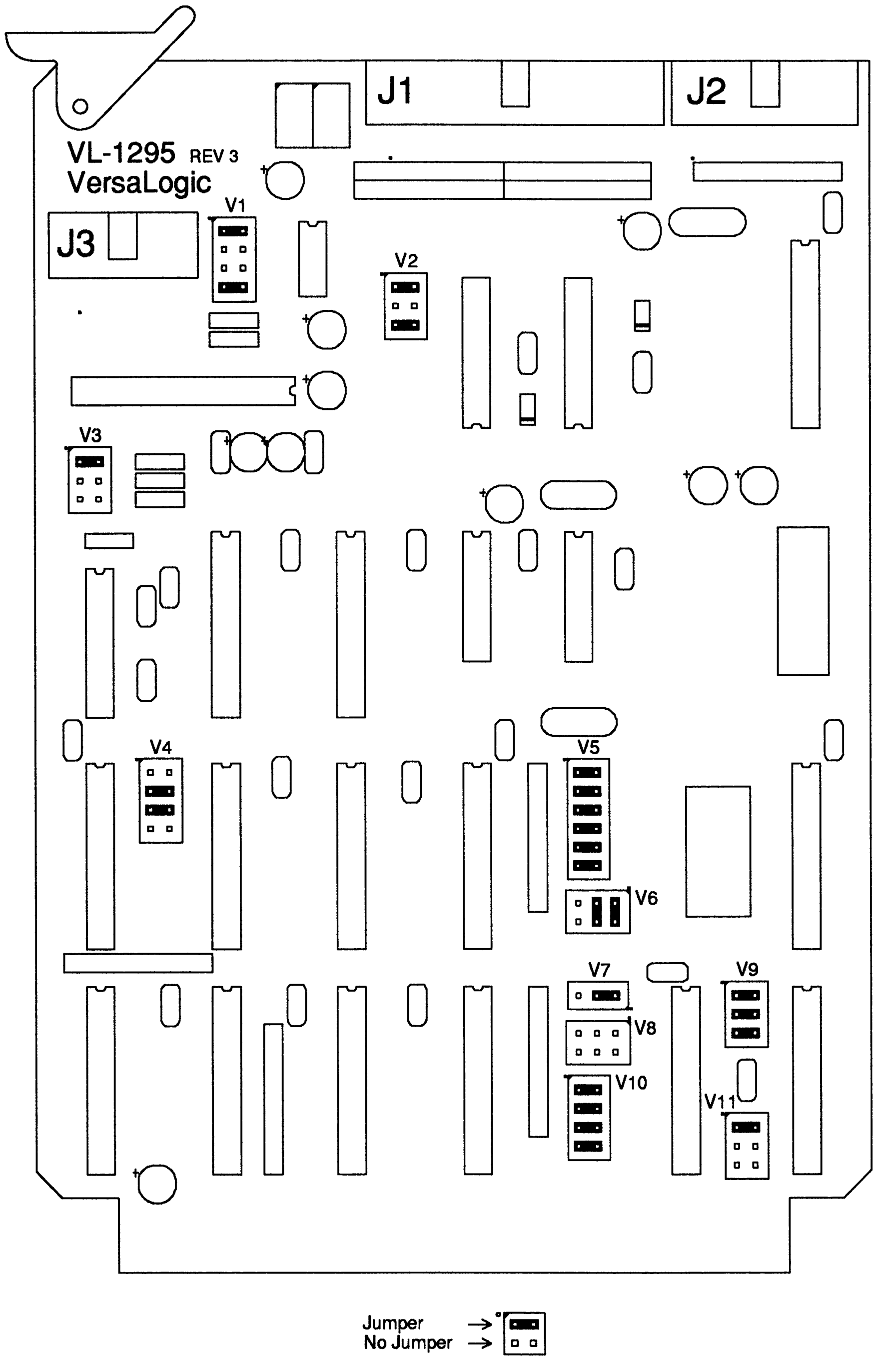


Figure 2-1. Jumper Block Locations for VL-1295

Jumper Options

Jumper Block	Description	As Shipped	Page
$V1_{a-d}$	Input Voltage Range		2-6
	a & d – ± 10 Volt	a & d In	
	b & c – ± 5 Volt	b & c Out	
$V2_a$	Analog signal source	In	2-7
	In – A/D circuitry receives analog voltage from on-board multiplexers		
	Out – A/D circuitry disconnected from on-board multiplexers		
$V2_b$	Analog signal source	Out	2-7
	In – A/D circuitry receives analog voltage from AIN2		
	Out – A/D circuitry disconnected from AIN2		
$V2_c$	Analog ground reference	In	2-7
	In – SEN signal (J1 pin 25) connected to analog ground		
	Out – SEN signal (J1 pin 25) floating		
$V3_{a-c}$	Multiplexer delay time		2-8
	a – 2 Microseconds	a In	
	b – 20 Microseconds	b Out	
	c – 100 Microseconds	c Out	
$V4_{a-d}$	Data Format		2-9
	a & d – Offset Binary	a & d Out	
	b & c – Two's Complement	b & c In	
$V5_{a-f}$	Board address (A10 – A15)		2-4
	a – A15	a In	
	b – A14	b In	
	c – A13	c In } 0300H	
	d – A12	d In	
	e – A11	e In	
	f – A10	f In	
$V6_{a-c}$	Address mode selector (8- or 16-bit decoding)		2-4
	a & c – 8-Bit address decoding	a & b In	
	a & b – 16-Bit address decoding		
V7	IOEXP select		2-6
	a – Board responds to IOEXP high and low (IOEXP ignored)	a In	
	b – Board responds to IOEXP low		
	None – Board responds to IOEXP high		
$V8_{a-b}$	Board address (A8, A9) / 8-Bit mode selector		2-4
	a – A9	a Out } 0300H	
	b – A8	b Out	
$V9_a$	DMA Acknowledge / BUS Interconnect	In	2-10
	In – STD 32 Bus signal DAKx* (E69) connected to on-board circuitry		
	Out – STD 32 Bus signal DAKx* (E69) disconnected		
$V9_b$	DMA Read / BUS Interconnect	In	2-10
	In – STD 32 Bus signal DMAIORD* (E58) connected to on-board circuitry		
	Out – STD 32 Bus signal DMAIORD* (E58) disconnected		
$V9_c$	Data bus width	In	2-8
	In – 8-bit data bus		
	Out – 16-bit data bus		
$V10_{a-d}$	Board address (A4 – A7)		2-4
	a – A7	a In	
	b – A6	b In } 0300H	
	c – A5	c In	
	d – A4	d In	
$V11_{a-c}$	Interrupt Request / BUS Interconnect		2-9
	a – Interrupt request connected to bus signal INTRQ* (P44)	a In	
	b – Interrupt request connected to bus signal INTRQ1* (P37)	b Out	
	c – Interrupt request connected to bus signal INTRQ2* (P50)	c Out	

Figure 2-2. Jumper Functions

Board Addressing

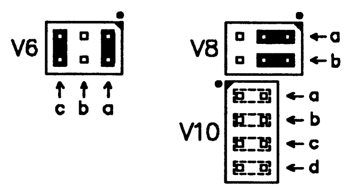
The VL-1295 card supports both 8- and 16-bit I/O addressing. 8-bit addressing is used with most 8-bit processors (Z80, 8085, 6809, etc.) which provide 256 I/O addresses. 16-bit addressing can be used with 16-bit processors (i.e. 8088, 80188, etc.) to decode up to 65536 I/O port addresses.

Both 8- and 16-bit addressing can be extended (capacity doubled) using the IOEXP signal which is decoded on board.

As shipped the board is configured for 16-bit addressing with a board address of hex 0300. The card occupies sixteen consecutive I/O addresses (i.e. 0300H to 030FH). Six of these addresses are mapped to control, data, and status registers, the remaining ten are inaccessible. See the I/O Port Mapping section on page 4-1 for further information.

8-Bit Addressing

To configure the board for an 8-bit I/O address refer to the figure below. Use the table to select the jumpering for the appropriate upper hex digit of the desired starting address (i.e. "3" and "0" = hex address 30). Note: the lower digit is always "0."



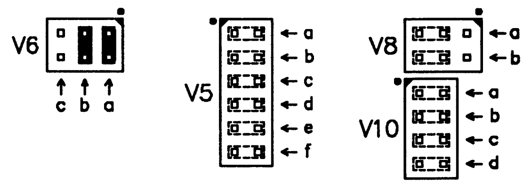
$V10_a$	$V10_b$	$V10_c$	$V10_d$	Upper Digit	Lower Digit
X	X	X	X	0	Always "0"
X	X	X	–	1	
X	X	–	X	2	
X	X	–	–	3	
X	–	X	X	4	
X	–	X	–	5	
X	–	–	X	6	
X	–	–	–	7	
–	X	X	X	8	
–	X	X	–	9	
–	X	–	X	A	
–	X	–	–	B	
–	–	X	X	C	
–	–	X	–	D	
–	–	–	X	E	
–	–	–	–	F	

x = Jumper installed.
– = Jumper removed.

Figure 2-3. 8-Bit Address Jumpers

16-Bit Addressing

To configure the board for a 16-bit I/O address refer to the figure below. Use the table to select the jumpering for the appropriate four hex digits of the desired starting address (i.e. "1" and "2" and "3" and "0" = hex address 1230). Note: the lower digit is always "0."



$V5_a$	$V5_b$	$V5_c$	$V5_d$	Top Digit	$V5_e$	$V5_f$	$V8_a$	$V8_b$	Second Digit	$V10_a$	$V10_b$	$V10_c$	$V10_d$	Third Digit	Lower Digit
X	X	X	X	0	X	X	X	X	0	X	X	X	X	0	Always "0"
X	X	X	–	1	X	X	X	–	1	X	X	X	–	1	
X	X	–	X	2	X	X	–	X	2	X	X	–	X	2	
X	X	–	–	3	X	X	–	–	3	X	X	–	–	3	
X	–	X	X	4	X	–	X	X	4	X	–	X	X	4	
X	–	X	–	5	X	–	X	–	5	X	–	X	–	5	
X	–	–	X	6	X	–	–	X	6	X	–	–	X	6	
X	–	–	–	7	X	–	–	–	7	X	–	–	–	7	
–	X	X	X	8	–	X	X	X	8	–	X	X	X	8	
–	X	X	–	9	–	X	X	–	9	–	X	X	–	9	
–	X	–	X	A	–	X	–	X	A	–	X	–	X	A	
–	X	–	–	B	–	X	–	–	B	–	X	–	–	B	
–	–	X	X	C	–	–	X	X	C	–	–	X	X	C	
–	–	X	–	D	–	–	X	–	D	–	–	X	–	D	
–	–	–	X	E	–	–	–	X	E	–	–	–	X	E	
–	–	–	–	F	–	–	–	–	F	–	–	–	–	F	

x = Jumper installed.
– = Jumper removed.

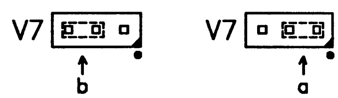
Figure 2-4. 16-Bit Address Jumpers

IOEXP Signal

The IOEXP (I/O expansion) signal on the STD Bus is normally used to select between two different I/O banks or maps. It can be used to double the number of available I/O addresses in the system (by selecting between two banks of I/O boards). The IOEXP signal is usually controlled by (or jumpered to ground on) the system CPU card.

A low IOEXP signal usually selects the standard or normal I/O map. A high IOEXP signal usually selects the secondary or alternate I/O map. Boards that ignore (or do not decode IOEXP) will appear in both I/O maps.

As shipped, the IOEXP jumper is configured to ignore the IOEXP signal. The board will be addressed whether the IOEXP signal is high or low. It can be jumpered for two other modes as shown in the figure below.



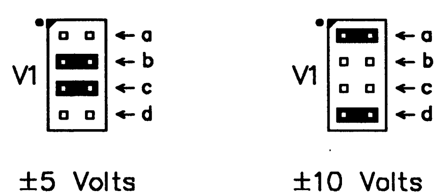
Jumper Block	Description	As Shipped
$V7_a$	Ignore IOEXP (enable high or low)	Ignore IOEXP
$V7_b$	Enable on IOEXP low	
None	Enable on IOEXP high (no jumpers)	

Figure 2-5. IOEXP Jumper

Input Range

The board may be operated with an input range of ±5 volts or ±10 volts. The ±5 volt range is preferred for signals which do not exceed ± 5 volts since the per volt resolution is twice that of the ±10 volt range.

As shipped the board is configured for ±10 volt operation. It can be jumpered for another range as shown below.

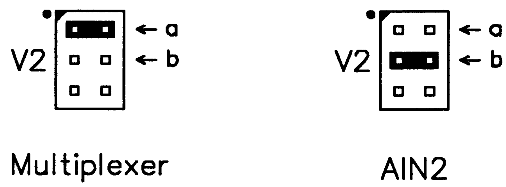


Jumper Block	Description	As Shipped
$V1_{bc}$	±5 volt input range	±10 Volts
$V1_{ad}$	±10 volt input range	

Figure 2-6. Input Range Jumper

Input Source

Select $V2_a$ for normal operation. In this configuration, analog signals are applied to connector J1 where they are selected by on-board multiplexers. $V2_b$ configures the analog input signal to originate from the AIN2 input (J2 pin 2) for applications which incorporate external multiplexer circuitry.



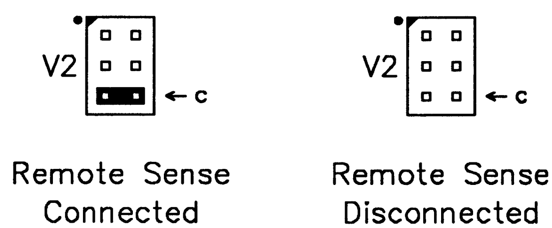
Jumper Block	Description	As Shipped
$V2_a$	Input sourced from on-board multiplexers	Multiplexer
$V2_b$	Input sourced from AIN2 input (J2 pin 2)	

Figure 2-7. Remote Sense Jumper

Remote Sense

The analog ground local to the on-board A/D converter can be brought out to connector J1 pin 25 for compatibility with 5B01 signal conditioning racks.

The SEN signal (J1 pin 25) is connected to analog ground when the jumper is in. It is left floating when the jumper is removed.



Jumper Block	Description	As Shipped
$V2_c$	Analog ground remote sense	Connected

Figure 2-8. Remote Sense Jumper

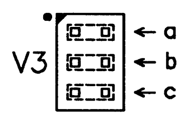
Settling Time

The VL-1295 board inserts a short delay between the time a channel is selected and time the A/D conversion begins. The delay allows the multiplexer and associated circuitry to properly settle for a more accurate reading.

The settling delay starts when the Channel Select register is written to by software, or when the Data High register is read (in auto increment mode). After channel selection (during the settling time), the A/D conversion cycle can be started by writing to the Convert register; however, the A/D converter is not triggered until the delay time elapses. The time delay is not inserted when one channel is read repeatedly.

In most cases, the standard 2 µS delay (as shipped) is appropriate. When using external signal conditioning circuitry, or external multiplexers attached to connector J2, it may be necessary to increase the settling time.

One of three delay times can be selected as indicated in the figure below.



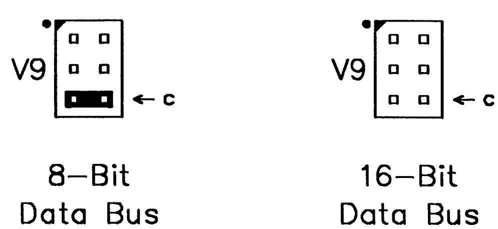
Jumper Block	Description	As Shipped
$V3_a$	2 Microseconds	2 Microseconds
$V3_b$	20 Microseconds	
$V3_c$	100 Microseconds	

Figure 2-9. Settling Time Jumper

Data Bus Width

The VL-1295 provides both 8-bit and 16-bit data paths to the bus. The 8-bit mode is compatible with STD 80, STD Z80, and STD 32 bus specifications. The 16-bit mode is used only when operating the card in an STD 32 card cage.

In 8-bit mode, the register map for the VL-1295 is a superset of the Analog Devices RTI-1265 board. In 16-bit mode, the registers are remapped to even addresses for efficient I/O access. This allows the full 12-bit A/D data word to be read in a single bus cycle. See the I/O Port Mapping section on page 4-1 for further information.



Jumper Block	Description	As Shipped
$V9_c$	Data Bus Width	8-Bit

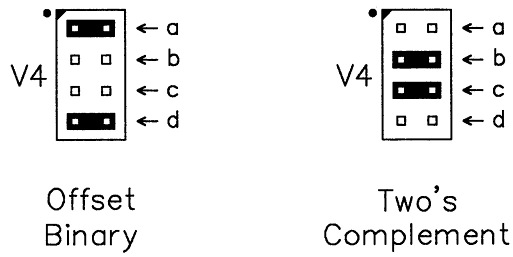
Figure 2-10. Data Bus Width Jumper

Data Format

The data format for the 12-bit A/D value can be configured as either offset binary or two's complement code. The appropriate format can be selected by positioning the jumpers as indicated in the figure below.

Note: When using binary coding in conjunction with a bipolar input signal, the result is effectively offset binary.

See the Data Representation section on page 4-8 for a description of these data formats.



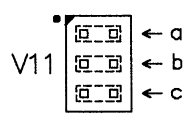
Jumper Block	Description	As Shipped
$V4_{ad}$	Offset binary	Two's Complement
$V4_{bc}$	Two's complement	

Figure 2-11. Data Format Jumper

Interrupt Request Enable

Jumper V11 connects the interrupt request signal from the VL-1295 card to one of three STD Bus interrupt request lines. The choice of which jumper position to choose depends upon the capabilities of the CPU or interrupt controller used in the system.

If an STD 32 Slot X interrupt controller is used, interrupts are requested on the dedicated slot specific signal IRQx (E47). In this case, all three jumpers in V11 should be removed.



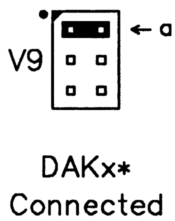
Jumper Block	Description	As Shipped
$V11_a$	Interrupt to INTRQ* (P44)	INTRQ*
$V11_b$	Interrupt to INTRQ1* (P37)	
$V11_c$	Interrupt to INTRQ2* (P50)	

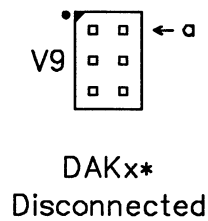
Figure 2-12. Interrupt Request Jumper

DMA Acknowledge

Jumper $V9_a$ connects the slot specific DAKx* (E69) signal from the STD 32 Bus to the VL-1295. The jumper should be inserted when using DMA with an STD 32 motherboard.

Note that the DMA acknowledge signal is always available on connector J3 pin 4 for use with non-STD 32 DMA controllers.





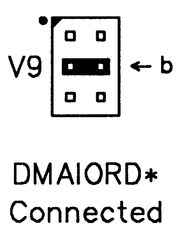
Jumper Block	Description	As Shipped
$V9_a$	DMA Acknowledge	DAKx* Connected

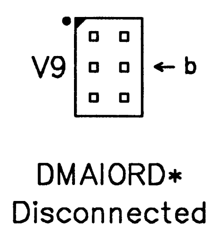
Figure 2-13. DMA Acknowledge Jumper

DMA Read

Jumper $V9_b$ connects the DMAIORD* (E58) signal from the STD 32 Bus to the VL-1295. The jumper should be inserted when using DMA with an STD 32 motherboard.

Note that the DMA Read signal is always available on connector J3 pin 4 for use with non STD 32 DMA controllers.





Jumper Block	Description	As Shipped
$V9_b$	DMA Read	DMAIORD* Connected

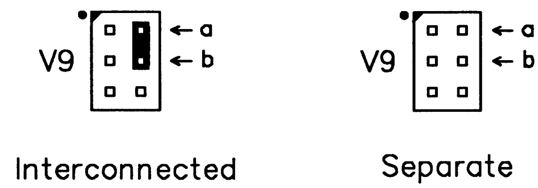
Figure 2-14. DMA Read Jumper

DMA Acknowledge / DMA Read Interconnect

The DMA acknowledge (DAK* J3 pin 4) and DMA read (DRD* J3 pin 8) signals can be tied together by inserting jumper $V9_{ab}$ as shown below.

Insert this jumper when using a DMA controller connected to J3 which does not provide a separate DMA read signal.

The jumper should be removed when using DMA with an STD 32 motherboard.



Jumper Block	Description	As Shipped
$V9_{ab}$	DMA Acknowledge / DMA Read Interconnect	Separate

Figure 2-15. DMA Acknowledge / DMA Read Interconnect Jumper

5B01 Analog Signal Conditioning Rack

When using a 5B01 series analog signal conditioning rack connected to J1, the following jumpers should be installed.

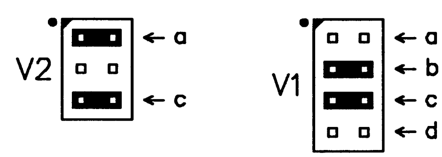


Figure 2-16. Signal Conditioning Rack Jumpering

Installation

Handling

CAUTION: The VL-1295 card uses 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-1295 card can be installed in any slot of an STD Bus card cage, excluding Slot-X in STD 32 cages, and should only be used with other standard (TTL level bus) STD Bus boards. When using CMOS STD Bus CPU boards use the VL-12CT95.

Signal Levels

The analog circuits on connector J1 are single ended analog inputs. All analog signals must be referenced to analog ground on connector J1. The maximum non-destructive input voltage applied to any of the inputs is ±35V with power on (±20V with power off). Each analog channel presents a minimum input impedance of $1 \times 10^8\ \Omega$.

Connector J2 contains a mixture of analog and digital signals. The digital signals are standard 5 Volt TTL logic levels referenced to digital ground on the same connector. The signals on connector J3 are all TTL.

Priority Chain

The VL-1295 card does not use the STD Bus priority interrupt chain signals PCO and PCI; however, because PCI is connected to PCO on board, the card can be installed between cards using the chain.

External Connections

J1, J2 and J3 are unlatched header type connectors. External connections to the VL-1295 can be made with standard cable assemblies, or with the following mating connectors:

Connector	Cable Assembly
J1	26-pin socket type connectors such as 3M #3399-7026
J2	14-pin socket type connectors such as 3M #3385-6614
J3	10-pin socket type connectors such as 3M #3473-6610

Figure 3-1. Cable Assemblies

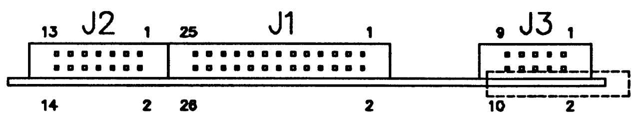
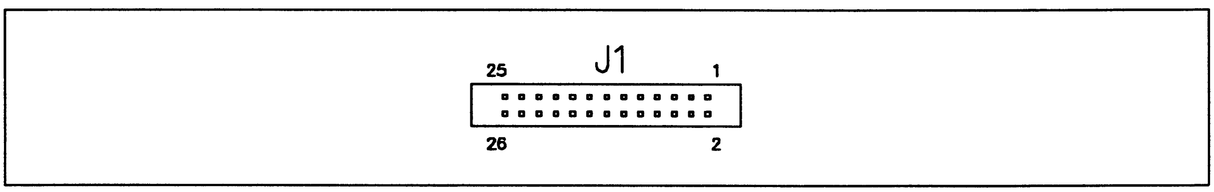


Figure 3-2. I/O Connector Physical Pin Locations

J1—Analog Input Connector



J1 Pin	Signal Name
1	Channel 0
2	Channel 8
3	Analog ground
4	Channel 9
5	Channel 1
6	Analog ground
7	Channel 2
8	Channel 10
9	Analog ground
10	Channel 11
11	Channel 3
12	Analog ground
13	Channel 4
14	Channel 12
15	Analog ground
16	Channel 13
17	Channel 5
18	Analog ground
19	Channel 6
20	Channel 14
21	Analog ground
22	Channel 15
23	Channel 7
24	Analog ground
25	Remote sense
26	N/C

Figure 3-3. J1 – Analog Input Connector Pinout

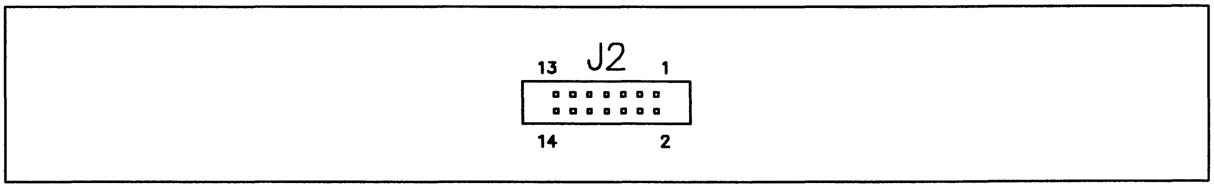
Channel 0 to 15. Analog voltages are applied to these single-ended inputs for A/D conversion.

Analog Ground. Analog voltages are referenced to these pins. The use of multiple ground connections is recommended to maintain a high degree of signal integrity.

Remote Sense. This signal is connected to analog ground at a point very close to the on-board A/D converter when jumper $V2_c$ is installed. It is used by the 5B01 signal conditioning rack. Remote sense is left floating when $V2_c$ is removed.

N/C — No Connection. This signal is not connected to on-board circuitry. It has no function on the VL-1295.

J2—External Multiplexer Connector



J2 Pin	Signal Name
1	Analog ground
2	AIN2
3	Analog ground
4	MA0
5	MA1
6	MA2
7	MA3
8	MA4
9	MUX EN*
10	EXT TR*
11	Digital ground
12	Digital ground
13	+5 Volts
14	+5 Volts

Figure 3-4. J2 – External Multiplexer Connector Pinout

AIN2 — Auxiliary Analog Input. This Auxiliary analog input bypasses the on-board multiplexers and is connected directly to the A/D input circuitry when jumper $V2_b$ is inserted. It is used for customer supplied external multiplexer circuitry. The maximum non-destructive input voltage applied to AIN2 is ±20V with power on (±10V with power off).

Analog Ground. The analog voltage applied to AIN2 is referenced to this pin.

MA0 to MA4 — Off-board Multiplexer Address Lines. These TTL output signals constantly reflect the value stored in the Channel Select register. They can be used for customer supplied external multiplexer circuitry.

MUX EN* — Multiplexer Enable. This TTL output signal goes active low when the address lines MA0 to MA4 are stable.

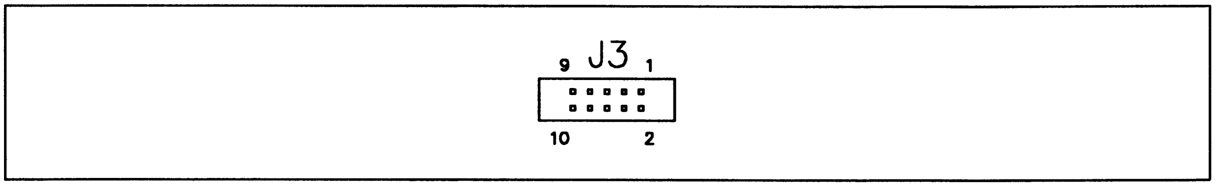
EXT TR* — External Trigger. A low going pulse on this TTL input triggers an A/D conversion. It is used to trigger A/D conversions synchronously with external events. The overrun condition is checked and latched into the status register on the high-to-low transition, the low-to-high transition triggers the A/D conversion.

An overrun condition will result if an external trigger is applied while a conversion is currently underway, or before reading the data from the previous conversion.

Digital Ground. All digital signals on connector J2 are referenced to these two digital ground connections. The use of both ground connections is recommended to maintain a high degree of signal integrity.

+5 Volts. This power supply output is available for external use. A maximum of 250 mA can be drawn from both of these pins. The return current path should be connected to digital ground.

J3—DMA Control Connector



J3 Pin	Signal Name
1	Digital ground
2	DRQ*
3	Digital ground
4	DAK*
5	Digital ground
6	N/C
7	Digital ground
8	DRD*
9	Digital ground
10	N/C

Figure 3-5. J3 – DMA Control Connector Pinout

DRQ* — DMA Request. When DMA mode is enabled this TTL output goes active (low) when the Done bit in the Status register is set to "1." It indicates to the system DMA controller that an A/D conversion is complete and data is ready to be transferred from the VL-1295 to memory. DRQ* goes inactive (high) on the high-to-low transition of DAK*.

DAK* — DMA Acknowledge. This TTL input from the system DMA controller signals the VL-1295 that a DMA request is now being honored. An active (low) level causes DRQ* to go inactive (high), also it enables the VL-1295 to respond to DRD*. An inactive (high) on DAK* causes DRD* to be ignored.

DRD* — DMA Read. This TTL input from the system DMA controller forces the VL-1295 to drive data onto the bus (assuming DAK* is active). When operating in 8-bit mode, two DRD* pulses are required to complete a full data transfer. In 16-bit mode, a single pulse transfers both the Data High and Data Low registers in one DMA cycle.

N/C — No Connection. These signals are not connected to on-board circuitry. They have no function on the VL-1295.

Digital ground. All signals on connector J3 are referenced to these digital ground connections. The use of all ground connections is recommended to maintain a high degree of signal integrity.

Registers

Introduction

This section includes information about registers, control and status bits, and data formats. It focuses primarily on the individual registers, the bits contained within them, and their functional descriptions.

I/O Port Mapping

The VL-1295 uses one of two I/O port maps depending upon whether the data bus is configured for 8-bit or 16-bit operation. In 8-bit mode, registers are mapped to correspond with Analog Device's RTI-1265 board. In 16-bit mode, the data registers are moved to an even address boundary to facilitate efficient single-cycle reading of the A/D data.

8-Bit Mode Input Port	8-Bit Mode Output Port	16-Bit Mode Input Port	16-Bit Mode Output Port	Port Address	As Shipped Address
—	—	—	—	Board Address + 15	030FH
—	—	—	—	Board Address + 14	030EH
—	—	—	—	Board Address + 13	030DH
—	—	—	—	Board Address + 12	030CH
—	—	—	—	Board Address + 11	030BH
—	—	—	—	Board Address + 10	030AH
—	Clear Flags	—	Clear Flags	Board Address + 9	0309H
—	—	—	—	Board Address + 8	0308H
—	—	—	—	Board Address + 7	0307H
—	—	—	—	Board Address + 6	0306H
—	—	Data High	—	Board Address + 5	0305H
Data High	—	Data Low	—	Board Address + 4	0304H
Data Low	—	—	—	Board Address + 3	0303H
—	Convert	—	Convert	Board Address + 2	0302H
—	Chan. Select	—	Chan. Select	Board Address + 1	0301H
Status	Control	Status	Control	Board Address + 0	0300H

Figure 4-1. I/O Port Addresses

Register Functions

The following table lists the functions assigned to each read and write register.

Write Registers	Functions	Page
Control	Interrupt, DMA, Auto-trigger, Auto increment, Scan range limit	4-2
Channel Select	Selects channel to convert	4-4
Convert[1]	Triggers A/D conversion	4-5
Clear Flags[1]	Clears Overrun, and Done bits	4-5

Read Registers	Functions	Page
Status	Busy, Done, Overrun, and Mux delay status bits	4-6
Data Low	Least significant byte of A/D data	4-7
Data High	Most significant byte of A/D data	4-7

[1] Data insensitive. Any value may be written to this port.

Figure 4-2. Read and Write Register Functions

Write Registers

Control Register

Control Register — 0300H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Figure 4-3. Control Register

The Control register is a write register used to configure the operating mode of the VL-1295.

D7, D6, D5 — Scan Range Limit. These three bits define and restrict the number of channels scanned in auto-increment mode. This allows for faster throughput when only eight or four analog signals are connected. A "1" bit in D7, D6, or D5 forces a "0" on the multiplexer address lines MA4, MA3, and MA2 respectively. Values other than those listed will cause channels to be skipped in groups. A board reset selects (000) so that applications which do not use auto-increment mode are not restricted to a limited set of channels.

D7	D6	D5	Scan Range	Comment
0	0	0	0 to 31	16 channels on-board, 16 off-board
1	0	0	0 to 15	All 16 on-board channels
1	1	0	0 to 7	Restricted range, on-board channels
1	1	1	0 to 3	Restricted range, on-board channels

Figure 4-4. Scan Range Limit

(000) Channels 0 to 31. This selection does not restrict the number of channels accessed in auto-increment mode. However, its use is limited to applications where external multiplexer circuitry is connected to J2. In this mode, on-board channels 0 through 15 are accessed in sequence, followed by off-board channels 16 through 31. Off-board multiplexer address lines MA0 through MA4 on connector J2 increment through the entire 0-31 address range. (000) should be selected for applications which do not use auto-increment mode.

(100) Channels 0 to 15. This selection causes all 16 on-board channels to be accessed in auto-increment mode. Channels 0 through 15 are accessed in sequence, and then repeated (modulo 16 restriction of the value contained in the Channel Select register). Off-board multiplexer address lines MA0 through MA4 on connector J2 increment from 0 to 15. MA4 will always be low.

(110) Channels 0 to 7. This selection causes the first eight channels to be accessed in auto-increment mode. Channels 0 through 7 are accessed in sequence, and then repeated (modulo 8 restriction of the value contained in the Channel Select register). Off-board multiplexer address lines MA0 through MA4 on connector J2 increment from 0 to 7. MA4 and MA3 will always be low.

(111) Channels 0 to 3. This selection causes the first four channels to be accessed in auto-increment mode. Channels 0 through 3 are accessed in sequence, and then repeated (modulo 4 restriction of the value contained in the Channel Select register). Off-board multiplexer address lines MA0 through MA4 on connector J2 increment from 0 to 3. MA4, MA3, and MA2 will always be low.

D4 — Auto Increment Enable. Setting this bit to "1" places the VL-1295 in auto-increment mode. In this mode the Channel Select register increments by one after the Data High register is read, allowing the next channel in sequence to be converted. The Channel Select register will increment to a maximum value set by the Scan Range Limit (bits D7, D6, and D5 of this register) and then repeat starting again with channel 0. A settling delay set by jumper V3 is inserted after each increment. Resetting this bit to "0" disables auto-increment mode, allowing the Channel Select register to retain its value. Auto-increment is disabled upon board reset. Auto-increment is compatible with manual and auto-trigger modes.

Before selecting auto-increment mode, the initial channel to be converted (usually channel 0) should be selected by writing to the Channel Select register. Also, in most cases (100) is written to the Scan Range Limit bits at the same time auto-increment is enabled.

D3 — Auto Trigger Enable. Setting this bit to "1" places the VL-1295 in auto-trigger mode. In this mode a new A/D conversion is triggered after the Data High register is read, eliminating the need to trigger a conversion by writing to the Convert register. Auto-trigger is typically used in DMA applications.

There are three possible ways to trigger an A/D conversion: writing to the Convert register; reading the Data High register (if auto-trigger is enabled); or a high-to-low transition on the external trigger input (EXT TR*, J2 pin 10).

To use auto-triggering, set this bit to "1", start the first A/D cycle using one of the three methods mentioned above, wait until Done, then read the Data registers. From this point on, just wait until Done and read data.

D2 — Direct Memory Access Enable. Setting this bit to "1" enables DMA mode. In this mode the system DMA controller is signaled to transfer data when the A/D conversion is complete. When using DMA, auto-trigger mode is usually enabled, allowing the VL-1295 to start the next conversion immediately after the previous cycle's data is transferred to memory. Reset this bit to "0" to disable DMA requests. See page 5-2 for further information about operating the VL-1295 with DMA. A DMA software example is shown on page 6-2.

D1 — Not Used. This bit has no function in the VL-1295.

D0 — Interrupt Enable. Setting this bit to "1" enables interrupts. In this mode an interrupt request is sent to the CPU when the A/D conversion is complete. Reset this bit to "0" to disable interrupt requests. See page 5-3 for further information about operating the VL-1295 with interrupts. An interrupt software example is shown on page 6-5.

Channel Select Register

Channel Select Register — 0301H							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Figure 4-5. Channel Select Register

The Channel Select register is a write register used to select the analog channel number to read. A word-wide output instruction to this register (out dx,ax) also writes into the Convert register causing channel addressing and triggering with one CPU instruction.

D7, D6, D5 — Not Used. These bits have no function in the VL-1295.

D4, D3, D2, D1, D0 — Channel Address. These bits select the analog channel to use for A/D conversion. In auto-increment mode, the channel address changes after each A/D conversion, in all other cases the value remains static.

Note: The Scan Range Limit bits in the Control register affect the number stored in this register. See page 4-2 for further information.

The off-board multiplexer address lines, MA0 through MA4 on connector J2 always reflect the value stored in this register. If not using an external multiplexer, D4 should always be reset to "0."

A settling delay set by jumper V3 is inserted whenever this register changes.

D4	D3	D2	D1	D0	Selected Channel
0	0	0	0	0	Channel 0
0	0	0	0	1	Channel 1
0	0	0	1	0	Channel 2
0	0	0	1	1	Channel 3
0	0	1	0	0	Channel 4
0	0	1	0	1	Channel 5
0	0	1	1	0	Channel 6
0	0	1	1	1	Channel 7
0	1	0	0	0	Channel 8
0	1	0	0	1	Channel 9
0	1	0	1	0	Channel 10
0	1	0	1	1	Channel 11
0	1	1	0	0	Channel 12
0	1	1	0	1	Channel 13
0	1	1	1	0	Channel 14
0	1	1	1	1	Channel 15
1	0	0	0	0	Channel 16[1]
1	0	0	0	1	Channel 17[1]
1	0	0	1	0	Channel 18[1]
1	0	0	1	1	Channel 19[1]
1	0	1	0	0	Channel 20[1]
1	0	1	0	1	Channel 21[1]
1	0	1	1	0	Channel 22[1]
1	0	1	1	1	Channel 23[1]
1	1	0	0	0	Channel 24[1]
1	1	0	0	1	Channel 25[1]
1	1	0	1	0	Channel 26[1]
1	1	0	1	1	Channel 27[1]
1	1	1	0	0	Channel 28[1]
1	1	1	0	1	Channel 29[1]
1	1	1	1	0	Channel 30[1]
1	1	1	1	1	Channel 31[1]

[1] Off-board channels using external multiplexer.

Figure 4-6. Channel Selection Codes

Convert Register

Convert Register — 0302H

D7	D6	D5	D4	D3	D2	D1	D0

Figure 4-7. Convert Register

The Convert register is a write register which, when written to, triggers (starts) an A/D conversion. It is data insensitive; any value written to the Convert register will produce a trigger.

There are three possible ways to trigger an A/D conversion: writing to the Convert register; reading the Data High register (if auto-trigger is enabled); or a high-low-high pulse on the external trigger input (EXT TR*, J2 pin 10).

A word-wide output instruction to the Channel Select register (out dx,ax) also writes into the Convert register causing channel addressing and triggering with one CPU instruction.

D7-D0 — Not Used. These bits have no function in the VL-1295. Any value written triggers an A/D conversion.

Clear Flags Register

Clear Flags Register — 0309H

D7	D6	D5	D4	D3	D2	D1	D0

Figure 4-8. Clear Flags Register

The Clear Flags register is a write register which, when written to, clears the Done bit and the Overrun bit in the Status register. It is data insensitive; any value written to the Clear Flags register will clear the flags.

D7-D0 — Not Used. These bits have no function in the VL-1295. Any value written will cause the Done and Overrun bits in the Status register to be cleared.

Read Registers

Status Register

Status Register — 0300H

D7	D6	D5	D4	D3	D2	D1	D0

Figure 4-9. Status Register

The Status register is a read register which contains the status of the VL-1295. It can be read at any time to determine if an A/D conversion is complete, or if any errors have occurred.

D7 — Busy. This bit is set to "1" when an A/D conversion is currently in progress. This bit comes directly from the A/D converter BUSY signal and automatically resets to "0" when the A/D conversion is complete.

D6 — Done. This bit is set to "1" when an A/D conversion has completed. It indicates that data is available to be read from the Data Low and Data High registers. When DMA mode is enabled, the DMA request signal goes active when Done is set. When interrupts are enabled, the interrupt request signal goes active when Done is set.

The Done bit is reset to "0" when the Data High register is read (or it may be reset by writing to the Clear Flags register).

D5 — Overrun. This bit is set to "1" whenever an attempt to trigger an A/D conversion occurs while a conversion is currently underway, or if an attempt is made to trigger an A/D conversion before reading the data from the previous conversion. The VL-1295 locks out all A/D triggering while the overrun condition is in force.

The Overrun bit is reset to "0" by writing to the Clear Flags register.

D4 — Settling Delay. This bit is reset to "0" while the settling delay set by jumper V3 is in progress. The bit is set to "1" at all other times. It is intended for factory use only.

D3, D2, D1, D0 — Not Used. These bits have no function in the VL-1295. They will always read as "1."

Data Low Register

Data Low Register — 0303H (8-Bit mode)
— 0304H (16-Bit mode)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Figure 4-10. Data Low Register

The Data Low register is a read register containing the lower 8 bits of data from the A/D conversion results. It is used in conjunction with the Data High register to read the complete 12-bit A/D data word.

After a conversion is complete (as reported by the Done bit in the Status register) the Data Low register should be read first, followed by the Data High register. A word-wide input instruction from the Data Low register (in ax,dx) will fetch data from both the Data Low and Data High registers in the proper sequence. This is true for both 8-bit and 16-bit modes.

In 8-bit mode, the Data High and Data Low registers are mapped to correspond with Analog Device's RTI-1265 board. In 16-bit mode, the data registers are moved to an even address boundary to facilitate efficient single-cycle reading of the A/D data.

D7-D0 — A/D Data (Least Significant Byte). These bits contain data bits D7 through D0 of the conversion results. See the Data Representation section on page 4-8 for a discussion of data formats.

Data High Register

Data High Register — 0304H (8-Bit mode)
— 0305H (16-Bit mode)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Figure 4-11. Data High Register

The Data High register is a read register containing the upper 8 bits of data from the A/D conversion results. It is used in conjunction with the Data Low register to read the complete 12-bit A/D data word. In two's complement mode bit D_3 is duplicated (sign extended) into bits D_4 through D_7.

When reading data the Data Low register should be read first, followed by the Data High register. See the Data Low Register section for further information on data register access.

When the Data High register is read the following events occur:

- A new A/D conversion is triggered if auto-trigger mode is enabled.
- The next channel in sequence is selected if auto-increment mode enabled.
- The DMA request signal goes inactive if DMA mode is enabled.
- The Done bit in the Status register is reset to "0."

D7-D0 — A/D Data (Most Significant Byte). These bits contain data bits D7 through D0 of the conversion results.

Data Representation

The format of the data read from the VL-1295 varies depending upon the data format chosen with jumper V4. The two formats are discussed below.

Natural Binary Format

Natural binary format divides the full applied analog input range into 4096 steps. The output code zero (0000H) is associated with the most negative voltage, i.e. −10 Volts. The largest output code (0FFFH) describes the most positive voltage, i.e., +9.9951 Volts. An analog value of 0 Volts (ground) will read as 0800H. The upper four bits of the Data High register are all zeros when using natural binary.

The formulas for calculating analog or natural binary digital values are given by:

$$Digital = \left[\frac{Analog + Span}{Step}\right] + 1 \qquad Analog = Step \times (Digital - 1) - Span$$

Where:

Analog	=	Applied voltage	
Digital	=	A/D Conversion Data	
Span	=	9.995117188	(±10V Range)
		4.997558594	(±5V Range)
Step	=	0.0048828125	(±10V Range)
		0.00244140625	(±5V Range)

Sample natural binary values are shown in the table below:

±5V Input Voltage	±10V Input Voltage	Output Data (Hex)	(Decimal)	Comment
+5.0000	+10.000	—	—	Out of range
+4.9976	+9.9951	0FFF	4095	Maximum positive voltage
+2.5000	+5.0000	0C00	3072	Positive half scale
+1.2500	+2.5000	0A00	2560	Positive quarter scale
+1.00098	—	099A	2458	+1 Volt (±5V range)
—	+1.00098	08CD	2253	+1 Volt (±10V range)
+0.00488	+0.00977	0802	2050	Positive 2 LSBs
+0.00244	+0.00488	0801	2049	Positive 1 LSB
0.0000	0.0000	0800	2048	Zero (ground input)
−0.00244	−0.00488	07FF	2047	Negative 1 LSB
−0.00488	−0.00977	07FE	2046	Negative 2 LSBs
—	−1.00098	0733	1843	−1 Volt (±10V range)
−1.00098	—	0666	1638	−1 Volt (±5V range)
−1.2500	−2.5000	0600	1536	Negative quarter scale
−2.5000	−5.0000	0400	1024	Negative half scale
−5.0000	−10.000	0000	0	Maximum negative voltage

Figure 4-12. Natural Binary Data Format

Two's Complement Format

Two's complement format, like natural binary format, divides the full applied analog input range into 4096 steps. The output code (0000H), however, is associated with an analog value of 0 Volts (ground). Positive analog values are represented by positive binary numbers, whereas negative analog values are represented by negative binary numbers, i.e., −1 = FFFFH. Bit D_3 of the Data High register is duplicated (sign extended) into bits D_4 through D_7.

The formulas for calculating analog or two's complement digital values are given by:

$$Digital = \left[\frac{Analog}{Step}\right] \qquad Analog = Step \times Digital$$

Where:

Analog = Applied voltage

Digital = A/D Conversion Data

Step = 0.0048828125 (±10V Range)
0.00244140625 (±5V Range)

Sample two's complement values are shown in the table below:

±5V Input Voltage	±10V Input Voltage	Output Data (Hex)	(Decimal)	Comment
+5.0000	+10.000	—	—	Out of range
+4.9976	+9.9951	07FF	2047	Maximum positive voltage
+2.5000	+5.0000	0400	1024	Positive half scale
+1.2500	+2.5000	0200	512	Positive quarter scale
+1.00098	—	019A	410	+1 Volt (±5V range)
—	+1.00098	00CD	205	+1 Volt (±10V range)
+0.00488	+0.00977	0002	2	Positive 2 LSBs
+0.00244	+0.00488	0001	1	Positive 1 LSB
0.0000	0.0000	0000	0	Zero (ground input)
− 0.00244	− 0.00488	FFFF	− 1	Negative 1 LSB
− 0.00488	− 0.00977	FFFE	− 2	Negative 2 LSBs
—	− 1.00098	FF33	− 205	− 1 Volt (±10V range)
− 1.00098	—	FE66	− 410	− 1 Volt (±5V range)
− 1.2500	− 2.5000	FE00	− 512	Negative quarter scale
− 2.5000	− 5.0000	FC00	− 1024	Negative half scale
− 5.0000	− 10.000	F800	− 2048	Maximum negative voltage

Figure 4-13. Two's Complement Data Format

Operation

This section describes how to operate the VL-1295. Three typical modes of operation, polled mode, DMA, and interrupt mode are discussed. Code examples written in 80188 assembly language are included in the next section.

Polled Mode

Polled mode operation is the simplest method of operating the VL-1295. In this mode, software is in control of the card at all times. It is the responsibility of the CPU to start each new A/D conversion as desired, and to read the digital results upon completion.

Polled Mode Steps

- Channel selection
- Trigger A/D conversion
- Wait until done
- Read data

Channel selection

Output the desired channel number to the Channel Select register. See page 4-4.

After the first channel selection, this step can be skipped for multiple conversions of the same channel. This will eliminate the settling delay which is inserted every time the Channel Select register is written to.

A word-wide output instruction to this register (out dx,ax) also writes into the Convert register causing channel addressing and triggering with one CPU instruction.

Trigger A/D conversion

Output to the Convert register. The register is data insensitive, any value written to it will start an A/D conversion. See page 4-5.

Wait until done

Read the Status register repeatedly until the Done bit is set to "1." See page 4-6.

Read data

Input from the Low Data register first, followed by the High Data register. A word-wide input instruction from the Data Low register (in ax,dx) will fetch data from both the Data Low and Data High registers in the proper sequence.

DMA Mode

DMA mode is used for high performance analog data acquisition. It is useful for moving large blocks of data (multiple A/D conversions) from the VL-1295 directly to memory without the software overhead associated with polled or interrupt modes.

DMA Mode Steps

- Initialize DMA controller
- Initialize VL-1295 for DMA mode
- Initialize VL-1295 for Auto-trigger (if desired)
- Channel selection
- Initialize VL-1295 for Auto-increment (if desired)
- Trigger A/D conversion (kick start)
- Wait until transfer complete
- Access data

Initialize DMA controller

The system DMA controller requires initialization of the destination memory address, number of transfers to perform, terminal count characteristics, etc. See your DMA controller instruction manual for further information. Setting the source address is irrelevant because the VL-1295 will provide data during a DMA read cycle regardless of the address being driven on the bus.

Initialize VL-1295 for DMA mode

Set bit D2 in the Control register. See page 4-3 for further information.

Initialize VL-1295 for Auto-trigger (if desired)

Set bit D3 in the Control register. See page 4-3 for further information.

Auto-trigger mode is usually enabled when using DMA. If it is left disabled, the VL-1295 depends upon external triggering to start each A/D conversion.

Channel selection

Output the desired channel number to the Channel Select register. See page 4-4.

This sets the starting channel number when using auto-increment mode.

Initialize VL-1295 for Auto-increment (if desired)

Set bit D4 in the Control register. See page 4-3 for further information.

Auto-increment mode is optional when using DMA. Use it when you want to transfer a series of analog input channels to memory. When disabled, the VL-1295 will convert the same channel repeatedly, without the penalty of the settling delay.

Trigger A/D conversion (kick start)

The DMA process is started when the VL-1295 completes an A/D conversion; however, it is necessary to start the first conversion by writing to the Convert register or by applying a high-to-low transition on the external trigger input (EXT TR*, J2 pin 10).

If auto-trigger mode is used, the VL-1295 will automatically continue converting and transferring data to memory until the DMA controller reaches the terminal count.

Wait until transfer complete

This is determined by the DMA controller. Usually the DMA transfer is considered complete when the terminal count is reached. Often the CPU is notified of this event by an interrupt. Alternatively many DMA controllers have status registers which can be read to determine a terminal count condition.

Access data

Data is accessed by reading memory. The high and low bytes are transferred to memory in reversed pairs. The table below shows the results of 4 readings.

Memory address	Contents
DMA destination address + 0	Low byte of conversion 1
DMA destination address + 1	High byte of conversion 1
DMA destination address + 2	Low byte of conversion 2
DMA destination address + 3	High byte of conversion 2
DMA destination address + 4	Low byte of conversion 3
DMA destination address + 5	High byte of conversion 3
DMA destination address + 6	Low byte of conversion 4
DMA destination address + 7	High byte of conversion 4

Terminate DMA

The DMA process can be terminated by the DMA controller, or by resetting the DMA enable bit in the Control register to "0."

Interrupt Mode

Interrupt mode eliminates the need to repeatedly poll the Status register for Done status. This frees up the CPU to execute unrelated code while the VL-1295 is busy with an A/D conversion. This is especially useful when using long settling delays. Another use for interrupts is for handling externally triggered applications.

Interrupt Mode Steps

- Interrupt Service Routine
- Initialize interrupt controller
- Initialize CPU to receive interrupt
- Initialize VL-1295 for interrupt mode
- Initialize VL-1295 for Auto-trigger mode (if desired)
- Channel selection
- Initialize VL-1295 for Auto-increment (if desired)
- Trigger A/D conversion

Interrupt Service Routine

The interrupt service routine reads A/D conversion results from the VL-1295 and stores the data somewhere. Data is input from the Low Data register first, followed by the High Data register. A word-wide input instruction from the Data Low register (in ax,dx) will fetch data from both the Data Low and Data High registers in the proper sequence.

The interrupt service routine can be written to select a different channel or trigger a new conversion.

Initialize interrupt controller

This involves setting up interrupt vector registers, priority, and unmasking. See your interrupt controller instruction manual for further information.

Initialize CPU to receive interrupts

This involves preparing the interrupt vector table, and enabling interrupts. See your CPU instruction manual for further information.

Initialize VL-1295 for interrupt mode

Set bit D0 in the Control register. See page 4-3 for further information.

Initialize VL-1295 for Auto-trigger mode (if desired)

Set bit D3 in the Control register. See page 4-3 for further information.

Channel selection
Output the desired channel number to the Channel Select register. See page 4-4.

This sets the starting channel number when using auto-increment mode.

Initialize VL-1295 for Auto-increment mode (if desired)
Set bit D4 in the Control register. See page 4-3 for further information.

Auto-increment mode is optional when using interrupts. Use it when you want to convert a series of analog input channels. When disabled, the VL-1295 will convert the same channel repeatedly, without the penalty of the settling delay.

Trigger A/D conversion
An interrupt is generated when the VL-1295 completes an A/D conversion. This can be accomplished by writing to the Convert register or by applying a high-to-low transition on the external trigger input (EXT TR*, J2 pin 10).

Software Examples

This section shows some software examples written in Microsoft MASM 5.0 assembly language to assist you in constructing your own software routines. The DMA and interrupt code examples are written specifically for use with VersaLogic's 80188 CPU card, VL-188.

Polled Mode Code Example

The following example reads channel 0 into the AX register. It assumes that the board is addressed at location 0300H in 8-bit mode.

The key program sections are:

READ : Reads A/D channel 0 into AX register.

BUSY : Location where program loops waiting for A/D conversion to complete.

```
                                                ;VL-1295 I/O PORT ADDRESSES
= 0300          control equ     0300h           ;Control Register
= 0300          status  equ     0300h           ;Status Register
= 0301          select  equ     0301h           ;Channel Select Register
= 0302          convert equ     0302h           ;Convert Register
= 0303          d_low   equ     0303h           ;Data Low Register
= 0304          d_high  equ     0304h           ;Data High Register
= 0309          clrflg  equ     0309h           ;Clear Flags Register


0000            code    segment para public 'CODE'
                        assume  cs:code

0000            read:                           ;READ CHANNEL 0 INTO AX REGISTER

0000  B0 00             mov     al,00h          ;Select channel 0
0002  BA 0301           mov     dx,select
0005  EE                out     dx,al

0006  BA 0302           mov     dx,convert      ;Trigger conversion
0009  EE                out     dx,al

000A  BA 0300           mov     dx,status       ;Standby until done
000D  EC        busy:   in      al,dx
000E  A8 40             test    al,01000000b
0010  74 FB             jz      busy

0012  BA 0303           mov     dx,d_low        ;Read data into AX register
0015  ED                in      ax,dx

0016            code    ends
                        end     read
```

DMA Code Example

The following code example shows how to operate the VL-1295 using DMA mode. This example performs 256 readings of channel 0, placing the results into a RAM buffer.

The key program sections are:

MAIN : Execution begins here. Installation and setup are coordinated here.

INIT_188 : Initializes the VL-188 DMA controller.

INIT_1295 : Initializes the VL-1295 to generate DMA requests upon conversion complete. Auto trigger mode is also enabled here.

CLEAR_MEM : Clears the DMA destination RAM area.

DMA_START : Starts the DMA block transfer by enabling DMA mode and manually triggering one A/D conversion.

```
                                                ;VL-1295 I/O PORT ADDRESSES
= 0300          control equ     0300h           ;Control Register
= 0300          status  equ     0300h           ;Status Register
= 0301          select  equ     0301h           ;Channel Select Register
= 0302          convert equ     0302h           ;Convert Register
= 0303          d_low   equ     0303h           ;Data Low Register
= 0304          d_high  equ     0304h           ;Data High Register
= 0309          clrflg  equ     0309h           ;Clear Flags Register

                                                ;VL-188 I/O PORT ADDRESSES
= FF30          istatus equ     0FF30h          ;Interrupt Status Register
= FFDA          dcw     equ     0FFDAh          ;DMA Control Word
= FFD8          tc      equ     0FFD8h          ;DMA Transfer Count
= FFD6          d_hi    equ     0FFD6h          ;DMA Destination Pointer (upper 4 bits)
= FFD4          d_lo    equ     0FFD4h          ;DMA Destination Pointer (lower 8 bits)
= FFD2          s_hi    equ     0FFD2h          ;DMA Source Pointer      (upper 4 bits)
= FFD0          s_lo    equ     0FFD0h          ;DMA Source Pointer      (lower 8 bits)

                                                ;Reserve space for stack
0000            stack   segment page public 'STACK'
0000  0200[ ?? ]        db      200h dup (?)

0200            stack   ends
                                                ;Reserve space for DMA data buffer
0000            data    segment page public 'DATA'
0000  0200[ ?? ] buffer db      200h dup (?)

0200            data    ends

0000            code    segment para public 'CODE'
                        assume  cs:code,ds:data
```

```
0000                main:                          ;MAINLINE CODE

0000  B8 ---- R             mov    ax,data         ;Set data segment register
0003  8E D8                 mov    ds,ax

0005  E8 0057 R             call   clear_mem       ;Zero destination buffer
0008  E8 0013 R             call   init_188        ;Initialize VL-188 DMA hardware
000B  E8 0050 R             call   init_1295       ;Enable VL-1295 DMA mode w/ autotrigger
000E  E8 0068 R             call   dma_start       ;Start DMA transfers

0011  EB FE         stop:   jmp    stop

0013                init_188:                      ;VL-188 DMA CONTROLLER INITIALIZATION

0013  BA FFD2               mov    dx,s_hi         ;DMA Source pointer set to any address in
0016  B8 0000               mov    ax,0000h        ;the range 200h - 2FFh in order to produce
0019  EF                    out    dx,ax           ;a DMA Acknowledge signal using Peripheral
001A  BA FFD0               mov    dx,s_lo         ;Chip Select 4 (PCS4* -- J1 pin 11).
001D  B8 0200               mov    ax,0200h        ;Upper 16 bits = 0000h
0020  EF                    out    dx,ax           ;Lower 16 bits = 0200h

0021  B8 ---- R             mov    ax,seg buffer   ;Upper 4 bits of destination address
0024  B1 0C                 mov    cl,12
0026  D3 E8                 shr    ax,cl
0028  BA FFD6               mov    dx,d_hi
002B  EF                    out    dx,ax
002C  B8 ---- R             mov    ax,seg buffer   ;Lower 8 bits of destination address
002F  B1 04                 mov    cl,4
0031  D3 E0                 shl    ax,cl
0033  05 0000 R             add    ax,offset buffer
0036  BA FFD4               mov    dx,d_lo
0039  EF                    out    dx,ax

003A  BA FFD8               mov    dx,tc           ;Transfer Count = 512 bytes
003D  B8 0200               mov    ax,0200h
0040  EF                    out    dx,ax

0041  BA FF30               mov    dx,istatus      ;Clear DMA Halt Transfer Bit
0044  B8 0000               mov    ax,0000h
0047  EF                    out    dx,ax

0048  BA FFDA               mov    dx,dcw          ;VL-188 DMA CONTROL WORD
004B  B8 A264               mov    ax,0A264h       ;D15 1 DMIO  = Destination in memory space
004E  EF                    out    dx,ax           ;D14 0 DDEC  = Increment destination address
004F  C3                    ret                    ;D13 1 DINC  = Increment destination address
                                                   ;D12 0 SMIO  = Source in I/O space
                                                   ;D11 0 SDEC  = Static source address
                                                   ;D10 0 SINC  = Static source address
                                                   ;D9  1 TC    = Stop DMA when done
                                                   ;D8  0 INT   = No interrupt upon DMA stop
                                                   ;D7  0 SYN1  = Source sync
                                                   ;D6  1 SYN0  = Source sync
                                                   ;D5  1 P     = High priority
                                                   ;D4  0 TDRQ  = No Timer 2 DMA
                                                   ;D3  0 ---   = Non Functional Bit
                                                   ;D2  1 CHG   = Allow mod. of ST Bit
                                                   ;D1  0 ST    = Disarm DMA channel
                                                   ;D0  0 ---   = Non Functional Bit
```

```
0050                 init_1295:                          ;VL-1295 INTERRUPT INITIALIZATION

0050  B0 0C                  mov     al,0Ch              ;CONTROL REGISTER
0052  BA 0300                mov     dx,control          ;D7   0 Scan Range Limit   = No limit
0055  EE                     out     dx,al               ;D6   0 Scan Range Limit   = No limit
0056  C3                     ret                         ;D5   0 Scan Range Limit   = No limit
                                                         ;D4   0 Auto Increment     = Off
                                                         ;D3   1 Auto Trigger       = On
                                                         ;D2   1 DMA Enable         = On
                                                         ;D1   0 Non Functional Bit = 0
                                                         ;D0   0 Interrupt Enable   = Off

0057                 clear_mem:                          ;ZERO DESTINATION BUFFER

0057  B8 ---- R              mov     ax,data
005A  8E C0                  mov     es,ax
005C  BF 0000 R              mov     di,offset buffer
005F  B9 0200                mov     cx,0200h
0062  B0 00                  mov     al,0
0064  FC                     cld
0065  F3/ AA             rep stosb
0067  C3                     ret

0068                 dma_start:                          ;START DMA TRANSFER

0068  BA FFDA                mov     dx,dcw
006B  ED                     in      ax,dx               ;Fetch existing value of DMA Control Word
006C  0D 0006                or      ax,0006h            ;ST = 1, CHG = 1   : Arm DMA channel
006F  EF                     out     dx,ax               ;Update DMA Control Word

0070  B0 00                  mov     al,00h              ;Select analog channel 0
0072  BA 0301                mov     dx,select
0075  EE                     out     dx,al

0076  BA 0302                mov     dx,convert          ;Trigger A/D conversion
0079  EE                     out     dx,al

007A  C3                     ret

007B                 code    ends
                             end     main
```

Interrupt Code Example

The following code example shows how to operate the VL-1295 using interrupts. This specific example uses the external trigger input to start each new A/D conversion. When the data is ready, an interrupt is generated.

The key program sections are:

MAIN :	Execution begins here. Installation and setup are coordinated here.
INIT_188 :	Initializes the VL-188 interrupt controller to accept interrupts from the STD Bus. Installs interrupt vector into low RAM.
INIT_1295 :	Initializes the VL-1295 to generate interrupts upon conversion complete.
ISR :	Interrupt Service Routine. This subroutine is responsible for reading, processing, and/or storing the A/D results from the VL-1295.

```
                                                                ;VL-1295 I/O PORT ADDRESSES
= 0300                          control equ     0300h           ;Control Register
= 0300                          status  equ     0300h           ;Status Register
= 0301                          select  equ     0301h           ;Channel Select Register
= 0302                          convert equ     0302h           ;Convert Register
= 0303                          d_low   equ     0303h           ;Data Low Register
= 0304                          d_high  equ     0304h           ;Data High Register
= 0309                          clrflg  equ     0309h           ;Clear Flags Register

                                                                ;VL-188 I/O PORT ADDRESSES
= FF22                          eoi     equ     0FF22h          ;80188 EOI Register
= FF3A                          intl    equ     0FF3Ah          ;80188 INT1 Control Register

0000                            stack   segment page public 'STACK'
0000  0100[ ?? ]                        db      100h dup (?)

0100                            stack   ends

0000                            data    segment page public 'DATA'
0000  ????                      value   dw      ?
0002                            data    ends

0000                            vector  segment at 0 public 'VECTOR'


0034                                    org     0034h
0034  ????????????????          vec     dq      ?
003C                            vector  ends

0000                            code    segment para public 'CODE'
                                        assume  cs:code,ds:data

0000                            main:                           ;MAINLINE CODE

0000  B8 ---- R                         mov     ax,data         ;Set data segment register
0003  8E D8                             mov     ds,ax

0005  C7 06 0000 R 0000                 mov     value,0         ;Initialize variable

000B  E8 0014 R                         call    init_188        ;Initialize VL-188 interrupts
000E  E8 002D R                         call    init_1295       ;Initialize VL-1295 interrupts
0011  FB                                sti                     ;Enable CPU interrupt flag

0012  EB FE                     stop:   jmp     stop            ;Rest of mainline goes here
```

```
0014                     init_188:                        ;VL-188 INTERRUPT INITIALIZATION

0014  BA FF3A                     mov    dx,int1          ;INT1 CONTROL REGISTER
0017  B8 0017                     mov    ax,0017h         ;D15 0 0 --- = Non Functional Bit
001A  EF                          out    dx,ax            ;D14 0 0 --- = Non Functional Bit
                                                          ;D13 0 0 --- = Non Functional Bit
                                                          ;D12 0 0 --- = Non Functional Bit
                                                          ;D11 0 0 --- = Non Functional Bit
                                                          ;D10 0 0 --- = Non Functional Bit
                                                          ;D9  0 0 --- = Non Functional Bit
                                                          ;D8  0 0 --- = Non Functional Bit
                                                          ;D7  0 0 --- = Non Functional Bit
                                                          ;D6  0 SFNM  = Normal
                                                          ;D5  0 C     = Non Cascade
                                                          ;D4  1 LTM   = Level Trigger
                                                          ;D3  0 MSK   = Non masked
                                                          ;D2  1 PR2   = Priority 7
                                                          ;D1  1 PR1   = Priority 7
                                                          ;D0  1 PR0   = Priority 7

                                                          ;Un-mask STD Bus INTRQ* interrupts
                                                          ;and set to non-cascade mode because
                                                          ;VL-1295 does not provide interrupt
                                                          ;vector.  CPU will internally
                                                          ;generate type code 13.

001B  B8 ---- R                   mov    ax,vector        ;Install service routine address
001E  8E D8                       mov    ds,ax            ;into CPU Interrupt Vector Table

                                  assume ds:vector
0020  C7 06 0034 R 0034 R         mov    word ptr vec[0],offset isr
0026  C7 06 0036 R ---- R         mov    word ptr vec[2],seg    isr
                                  assume ds:data
002C  C3                          ret

002D                     init_1295:                       ;VL-1295 INTERRUPT INITIALIZATION

002D  B0 01                       mov    al,01h           ;CONTROL REGISTER
002F  BA 0300                     mov    dx,control       ;D7   0 Scan Range Limit   = No limit
0032  EE                          out    dx,al            ;D6   0 Scan Range Limit   = No limit
0033  C3                          ret                     ;D5   0 Scan Range Limit   = No limit
                                                          ;D4   0 Auto Increment     = Off
                                                          ;D3   0 Auto Trigger       = Off
                                                          ;D2   0 DMA Enable         = Off
                                                          ;D1   0 Non Functional Bit = 0
                                                          ;D0   1 Interrupt Enable   = On


0034                     isr:                             ;INTERRUPT SERVICE ROUTINE

0034  50                          push   ax               ;Save CPU registers
0035  52                          push   dx
0036  1E                          push   ds

0037  B8 ---- R                   mov    ax,data          ;Set data segment register
003A  8E D8                       mov    ds,ax

003C  BA 0303                     mov    dx,d_low         ;Read A/D results
003F  ED                          in     ax,dx

0040  A3 0000 R                   mov    value,ax         ;Store results into variable
                                                          ;
                                                          ;Additional processing code is
                                                          ;inserted here if desired.
                                                          ;This could include mathematic
                                                          ;manipulation, data storage,
                                                          ;limit checks, etc.
```

```
0043                      isr_exit:
0043  BA FF22                     mov     dx,eoi          ;Issue a Non-Specific End-Of-Interrupt
0046  B8 8000                     mov     ax,8000h        ;command to 80188 interrupt controller
0049  EF                          out     dx,ax

004A  1F                          pop     ds              ;Restore CPU registers
004B  5A                          pop     dx
004C  58                          pop     ax

004D  CF                          iret                    ;Return to interrupted program

004E                      code    ends
                                  end     main
```

Reference

Specifications

Input Channels: 16 single-ended multiplexed, 1 external

Input Range: ±5V or ±10V

Resolution: 12 Bits (4096 counts)

Conversion Time: 3 µs + settling time

Settling Time: 2 µs, 20 µs, or 100 µs (selectable)

Throughput:

- 333,000 readings/sec (any one channel repeated read)
- 200,000 readings/sec (scanned channel reading)

Input Overvoltage Protection:

- ±35V with power on (channel inputs)
- ±20V with power off (channel inputs)
- ±20V with power on (AIN2 input)
- ±10V with power off (AIN2 input)

Input Impedance: $1 \times 10^8\ \Omega$ min.

Data Format: Binary or two's complement

Linearity: ±½ LSB – over temperature range

Differential Linearity: ±¾ LSB – over temperature range

Temperature Coefficient:

- Gain: ±25 ppm/° C of FSR
- Offset: ±10 ppm/° C of FSR

Addressing: I/O, 8, 10, or 16 bits plus IOEXP

Mapping: 16-byte block on any 16-byte boundary

Size: Meets all STD 32 Bus mechanical specifications

Storage Temperature: −40° to +85° C

Free Air Operating Temperature:

- VL-1295: 0° to +65° C
- VL-12CT95: −25° to +85° C

Power Requirements:

- VL-1295: 5V ±5% at 300 ma typ.
- VL-12CT95: 5V ±5% at 300 ma typ.

Bus Compatibility:

- STD 80: Full compliance, all bus speeds
- STD Z80: Full compliance, all bus speeds
- STD 32: I/O slave, SA16, SA8-I, IXP, SDMA8, SDMA16

Jumper Block Locations

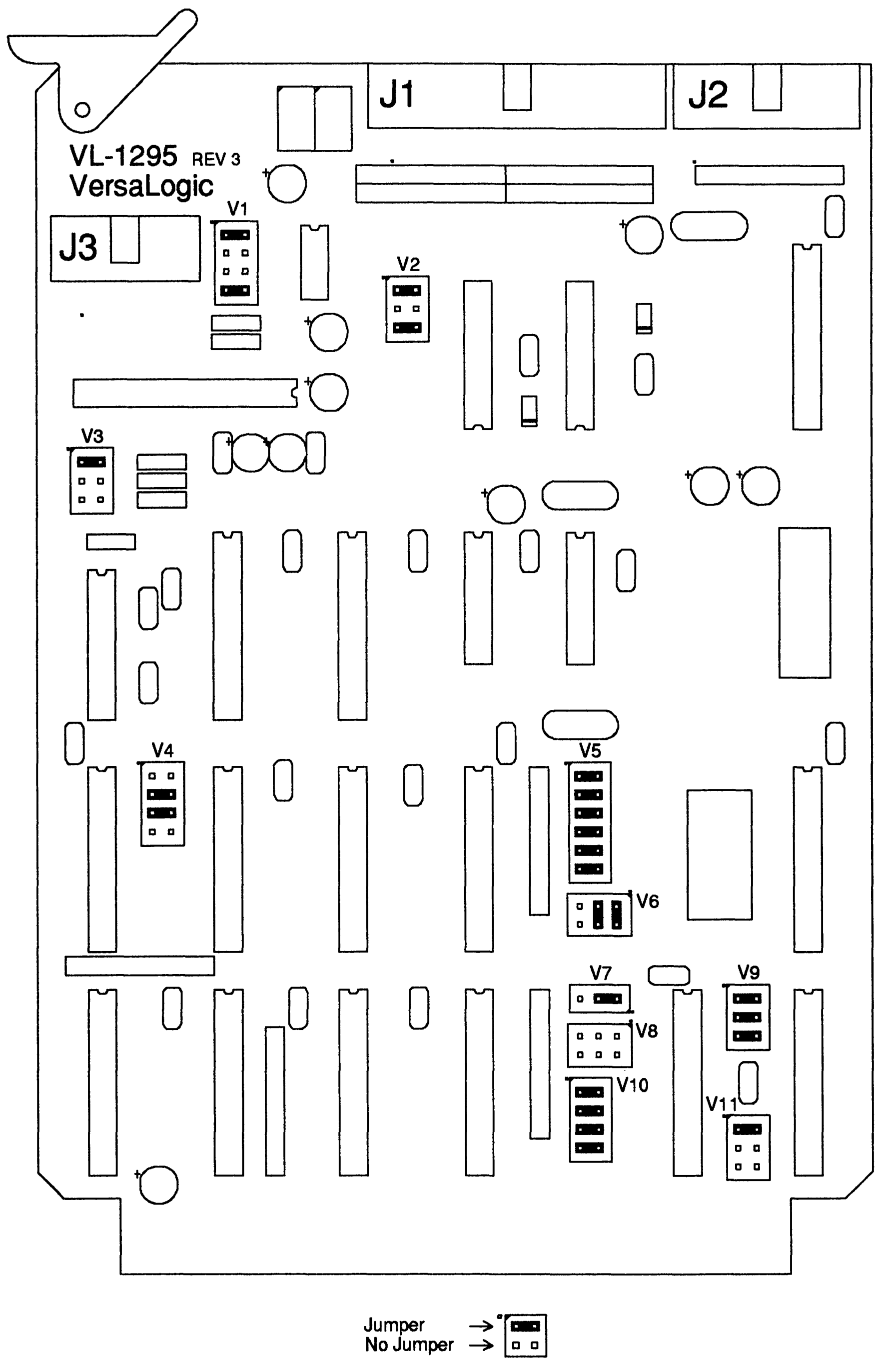


Figure 7-1. Jumper Block Locations for VL-1295

Jumper Options

Jumper Block	Description	As Shipped	Page
V1$_{a\text{-}d}$	Input Voltage Range		2-6
	a & d – ± 10 Volt	a & d In	
	b & c – ± 5 Volt	b & c Out	
V2$_{a}$	Analog signal source	In	2-7
	In – A/D circuitry receives analog voltage from on-board multiplexers		
	Out – A/D circuitry disconnected from on-board multiplexers		
V2$_{b}$	Analog signal source	Out	2-7
	In – A/D circuitry receives analog voltage from AIN2		
	Out – A/D circuitry disconnected from AIN2		
V2$_{c}$	Analog ground reference	In	2-7
	In – SEN signal (J1 pin 25) connected to analog ground		
	Out – SEN signal (J1 pin 25) floating		
V3$_{a\text{-}c}$	Multiplexer delay time		2-8
	a – 2 Microseconds	a In	
	b – 20 Microseconds	b Out	
	c – 100 Microseconds	c Out	
V4$_{a\text{-}d}$	Data Format		2-9
	a & d – Offset Binary	a & d Out	
	b & c – Two's Complement	b & c In	
V5$_{a\text{-}f}$	Board address (A10 – A15)		2-4
	a – A15	a In	
	b – A14	b In	
	c – A13	c In	
	d – A12	d In } 0300H	
	e – A11	e In	
	f – A10	f In	
V6$_{a\text{-}c}$	Address mode selector (8- or 16-bit decoding)		2-4
	a & c – 8-Bit address decoding	a & b In	
	a & b – 16-Bit address decoding		
V7	IOEXP select		2-6
	a – Board responds to IOEXP high and low (IOEXP ignored)	a In	
	b – Board responds to IOEXP low		
	None – Board responds to IOEXP high		
V8$_{a\text{-}b}$	Board address (A8, A9) / 8-Bit mode selector		2-4
	a – A9	a Out } 0300H	
	b – A8	b Out	
V9$_{a}$	DMA Acknowledge / BUS Interconnect	In	2-10
	In – STD 32 Bus signal DAKx* (E69) connected to on-board circuitry		
	Out – STD 32 Bus signal DAKx* (E69) disconnected		
V9$_{b}$	DMA Read / BUS Interconnect	In	2-10
	In – STD 32 Bus signal DMAIORD* (E58) connected to on-board circuitry		
	Out – STD 32 Bus signal DMAIORD* (E58) disconnected		
V9$_{c}$	Data bus width	In	2-8
	In – 8-bit data bus		
	Out – 16-bit data bus		
V10$_{a\text{-}d}$	Board address (A4 – A7)		2-4
	a – A7	a In	
	b – A6	b In } 0300H	
	c – A5	c In	
	d – A4	d In	
V11$_{a\text{-}c}$	Interrupt Request / BUS Interconnect		2-9
	a – Interrupt request connected to bus signal INTRQ* (P44)	a In	
	b – Interrupt request connected to bus signal INTRQ1* (P37)	b Out	
	c – Interrupt request connected to bus signal INTRQ2* (P50)	c Out	

Figure 7-2. Jumper Functions

I/O Port Mapping

All communications with the VL-1295 take place through the following I/O ports:

8-Bit Mode Input Port	8-Bit Mode Output Port	16-Bit Mode Input Port	16-Bit Mode Output Port	Port Address	As Shipped Address
—	—	—	—	Board Address + 15	030FH
—	—	—	—	Board Address + 14	030EH
—	—	—	—	Board Address + 13	030DH
—	—	—	—	Board Address + 12	030CH
—	—	—	—	Board Address + 11	030BH
—	—	—	—	Board Address + 10	030AH
—	Clear Flags	—	Clear Flags	Board Address + 9	0309H
—	—	—	—	Board Address + 8	0308H
—	—	—	—	Board Address + 7	0307H
—	—	—	—	Board Address + 6	0306H
—	—	Data High	—	Board Address + 5	0305H
Data High	—	Data Low	—	Board Address + 4	0304H
Data Low	—	—	—	Board Address + 3	0303H
—	Convert	—	Convert	Board Address + 2	0302H
—	Chan. Select	—	Chan. Select	Board Address + 1	0301H
Status	Control	Status	Control	Board Address + 0	0300H

Figure 7-3. I/O Port Addresses

Calibration

The VL-1295 is calibrated before shipment. However, it may be desirable to recalibrate the card after installation, and approximately once per year (depending upon the accuracy requirements of the application).

Required Equipment

- A voltmeter with resolution and accuracy to ½ LSB of the input range being used (2.44mV for ±10 Volt range, 1.22mV for ±5 Volt range).
- A low noise voltage source adjustable over the input range.

Calibration Procedure

- Disconnect all inputs to the card and connect the voltage source to any channel.
- Adjust the voltage source (per the table below) to the "ZERO" calibration voltage.
- Using a program to continuously read the input channel, adjust the "ZERO" pot until the reading toggles alternately between 07FFH and 0800H (FFFFH and 0000H in two's complement mode).
- Adjust the voltage source (per the table below) to the "SPAN" calibration voltage.
- Using a program to continuously read the input channel, adjust the "SPAN" pot until the reading toggles alternately between 0FFEH and 0FFFH (07FEH and 07FFH in two's complement mode).

Calibration Voltage	±10V Range	±5V Range	Comments
"ZERO"	–2.44mV	–1.22mV	0V – ½LSB
"SPAN"	+9.9927V	+4.9963V	MAX V – ½LSB

Figure 7-4. Calibration Voltages

External Connections

J1, J2 and J3 are unlatched header type connectors. External connections to the VL-1295 can be made with standard cable assemblies, or with the following mating connectors:

Connector	Cable Assembly
J1	26-pin socket type connectors such as 3M #3399-7026
J2	14-pin socket type connectors such as 3M #3385-6614
J3	10-pin socket type connectors such as 3M #3473-6610

Figure 7-5. Cable Assemblies

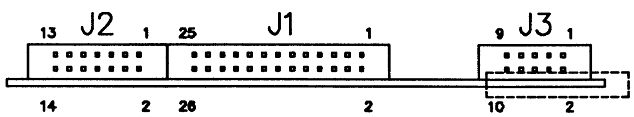
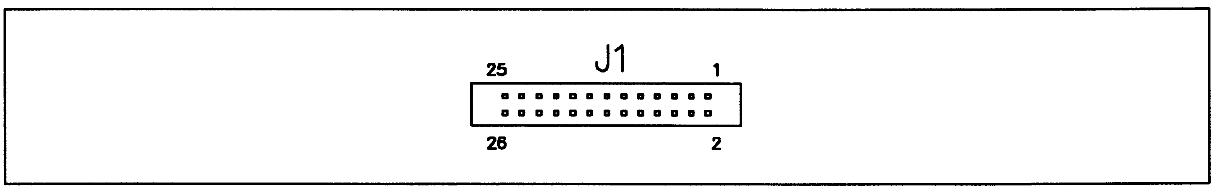


Figure 7-6. I/O Connector Physical Pin Locations

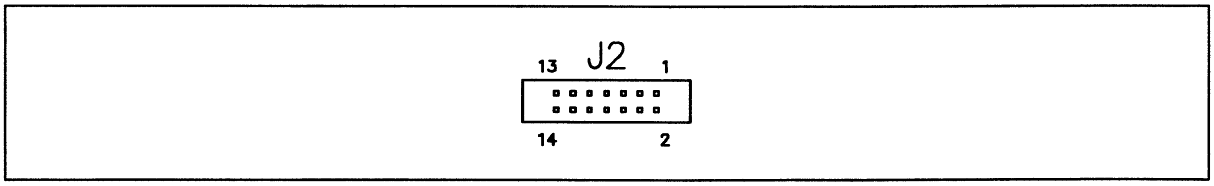
J1—Analog Input Connector



J1 Pin	Signal Name
1	Channel 0
2	Channel 8
3	Analog ground
4	Channel 9
5	Channel 1
6	Analog ground
7	Channel 2
8	Channel 10
9	Analog ground
10	Channel 11
11	Channel 3
12	Analog ground
13	Channel 4
14	Channel 12
15	Analog ground
16	Channel 13
17	Channel 5
18	Analog ground
19	Channel 6
20	Channel 14
21	Analog ground
22	Channel 15
23	Channel 7
24	Analog ground
25	Remote sense
26	N/C

Figure 7-7. Analog Input Connector Pinout

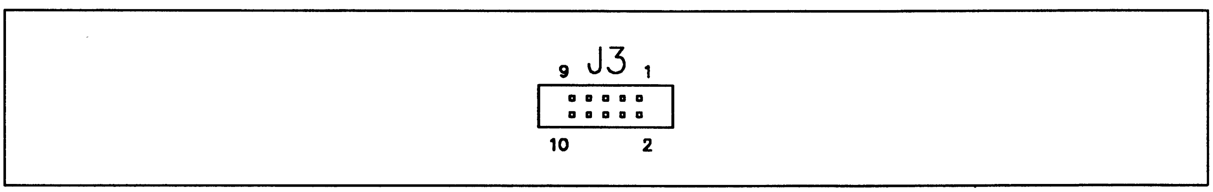
J2—External Multiplexer Connector



J2 Pin	Signal Name
1	Analog ground
2	AIN2
3	Analog ground
4	MA0
5	MA1
6	MA2
7	MA3
8	MA4
9	MUX EN*
10	EXT TR*
11	Digital ground
12	Digital ground
13	+5 Volts
14	+5 Volts

Figure 7-8. External Multiplexer Connector Pinout

J3—DMA Control Connector



J3 Pin	Signal Name
1	Digital ground
2	DRQ*
3	Digital ground
4	DAK*
5	Digital ground
6	N/C
7	Digital ground
8	DRD*
9	Digital ground
10	N/C

Figure 7-9. DMA Control Connector Pinout

Decimal / Hex / ASCII Conversion Chart

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

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

Figure 7-10. Decimal / Hex / ASCII Conversion Chart

STD 80 Bus Pinout

COMPONENT SIDE				SOLDER SIDE			
Pin	**Signal**	**Flow**	**Description**	**Pin**	**Signal**	**Flow**	**Description**
P01	+5VDC	In	Logic Power	P02	+5VDC	In	Logic Power
P03	GND	In	Logic Ground	P04	GND	In	Logic Ground
P05	VBAT	—	Battery Power	P06	DCPDN*	—	DC Power Down
P07	A19/D3	I/O	Address/Data	P08	A23/D7	I/O	Address/Data
P09	A18/D2	I/O	Address/Data	P10	A22/D6	I/O	Address/Data
P11	A17/D1	I/O	Address/Data	P12	A21/D5	I/O	Address/Data
P13	A16/D0	I/O	Address/Data	P14	A20/D4	I/O	Address/Data
P15	A07	In	Address	P16	A15	In	Address
P17	A06	In	Address	P18	A14	In	Address
P19	A05	In	Address	P20	A13	In	Address
P21	A04	In	Address	P22	A12	In	Address
P23	A03	In	Address	P24	A11	In	Address
P25	A02	In	Address	P26	A10	In	Address
P27	A01	In	Address	P28	A09	In	Address
P29	A00	In	Address	P30	A08	In	Address
P31	WR*	In	Write Mem or I/O	P32	RD*	In	Read Mem or I/O
P33	IORQ*	In	I/O Address Select	P34	MEMRQ*	—	Memory Address Select
P35	IOEXP	In	I/O Expansion	P36	BHE*	In	Byte High Enable
P37	INTRQ1*	Out	Interrupt Request 1	P38	ALE*	—	Address Latch Enable
P39	STATUS1*	—	CPU Status 1	P40	STATUS0*	—	CPU Status 0
P41	BUSAK*	—	Bus Acknowledge	P42	BUSRQ*	—	Bus Request
P43	INTAK*	—	Interrupt Acknowledge	P44	INTRQ*	Out	Interrupt Request
P45	WAITRQ*	—	Wait Request	P46	NMIRQ*	—	Non-maskable Interrupt Request
P47	SYSRESET*	In	System Reset	P48	PBRESET*	—	Push-Button Reset
P49	CLOCK*	In	Clock	P50	CNTRL* (INTRQ2*)	Out	Aux Timing
P51	PCO	Out	Priority Chain Out	P52	PCI	In	Priority Chain In
P53	AUX GND	—	AUX Ground	P54	AUX GND	—	AUX Ground
P55	AUX +V	—	AUX Positive (+12VDC)	P56	AUX –V	—	AUX Negative (–12VDC)

* Denotes an active low signal.
– Denotes signal not used on this board.

Figure 7-11. STD 80 Bus Pinout

STD 32 Bus Pinout Extension

COMPONENT SIDE				SOLDER SIDE			
Pin	**Signal**	**Flow**	**Description**	**Pin**	**Signal**	**Flow**	**Description**
E01	GND	—	Logic Ground	E02	RSVD	—	Reserved
E03	XA19	—	Address	E04	XA23	—	Address
E05	XA18	—	Address	E06	XA22	—	Address
E07	XA17	—	Address	E08	XA21	—	Address
E09	XA16	—	Address	E10	XA20	—	Address
E11	NOWS*	—	No Wait States	E12	RSVD	—	Reserved
E13	+5VDC	In	Logic Power	E14	+5VDC	In	Logic Power
E15	DAKx*	—	DMA Acknowledge	E16	DREQx*	—	DMA Request
E17	GND	In	Logic Ground	E18	GND	In	Logic Ground
E19	D27	—	Data	E20	D31	—	Data
E21	D26	—	Data	E22	D30	—	Data
E23	D25	—	Data	E24	D29	—	Data
E25	D24	—	Data	E26	D28	—	Data
E27	D23	—	Data	E28	GND	—	Logic Ground
E29	D22	—	Data	E30	D15	I/O	Data
E31	D21	—	Data	E32	D14	I/O	Data
E33	D20	—	Data	E34	D13	I/O	Data
E35	GND	—	Logic Ground	E36	D12	I/O	Data
E37	D19	—	Data	E38	D11	I/O	Data
E39	D18	—	Data	E40	D10	I/O	Data
E41	D17	—	Data	E42	D09	I/O	Data
E43	D16	—	Data	E44	D08	I/O	Data
E45	GND	—	Logic Ground	E46	MASTER 16*	—	Master 16-Bit
E47	IRQx	Out	Interrupt Request	E48	AENx*	—	Address Enable
E49	BE1*	—	Byte Enable 1	E50	BE3*	—	Byte Enable 3
E51	BE0*	—	Byte Enable 2	E52	BE2*	—	Byte Enable 2
E53	MEM16*	—	Memory 16-Bit	E54	GND	—	Logic Ground
E55	M–IO	—	Memory or I/O	E56	W–R	—	Write or Read
E57	DMAIOW*	—	DMA I/O Write	E58	DMAIOR*	In	DMA I/O Read
E59	IO16*	Out	I/O 16-Bit	E60	EX8*	—	Exchange 8-Bit
E61	CMD*	—	Command	E62	START*	—	Start
E63	EX16*	—	Exchange 16-Bit	E64	EX32*	—	Exchange 32-Bit
E65	EXRDY	—	Exchange Ready	E66	T–C	—	Terminate or Count
E67	LOCK*	—	Lock	E68	+5VDC	—	Logic Power
E69	MAKx*	In	Master Acknowledge	E70	MREQx*	Out	Master Request
E71	SLBURST*	—	Slave Burst	E72	MSBURST*	—	Master Burst
E73	XA27*	—	Address	E74	XA31*	—	Address
E75	XA26*	—	Address	E76	XA30*	—	Address
E77	XA25*	—	Address	E78	XA29*	—	Address
E79	XA24*	—	Address	E80	XA28*	—	Address

* Denotes an active low signal.
– Denotes signal not used on this board.

Figure 7-12. STD 32 Bus Pinout Extension

VL-1295 Parts Placement Diagram

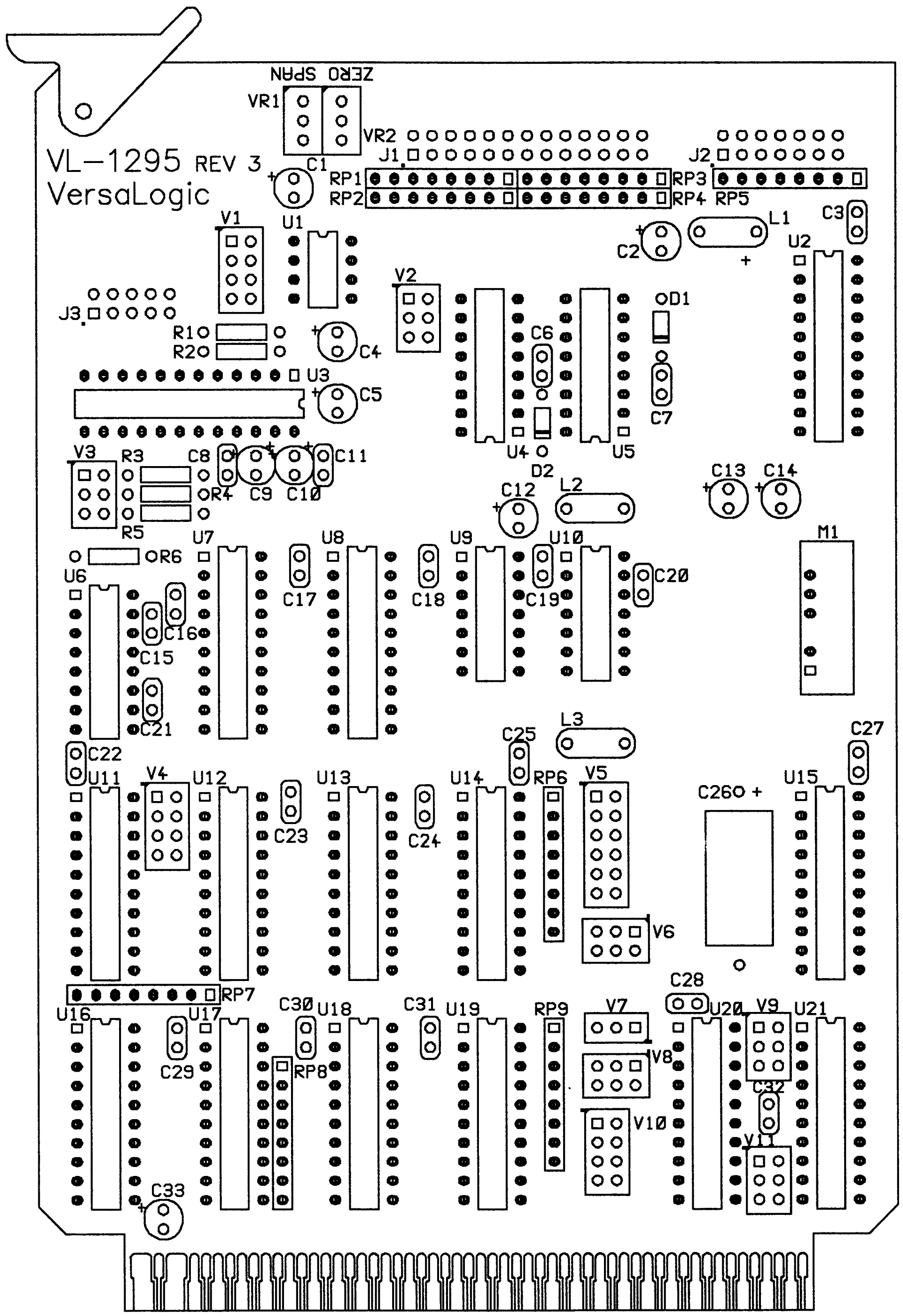
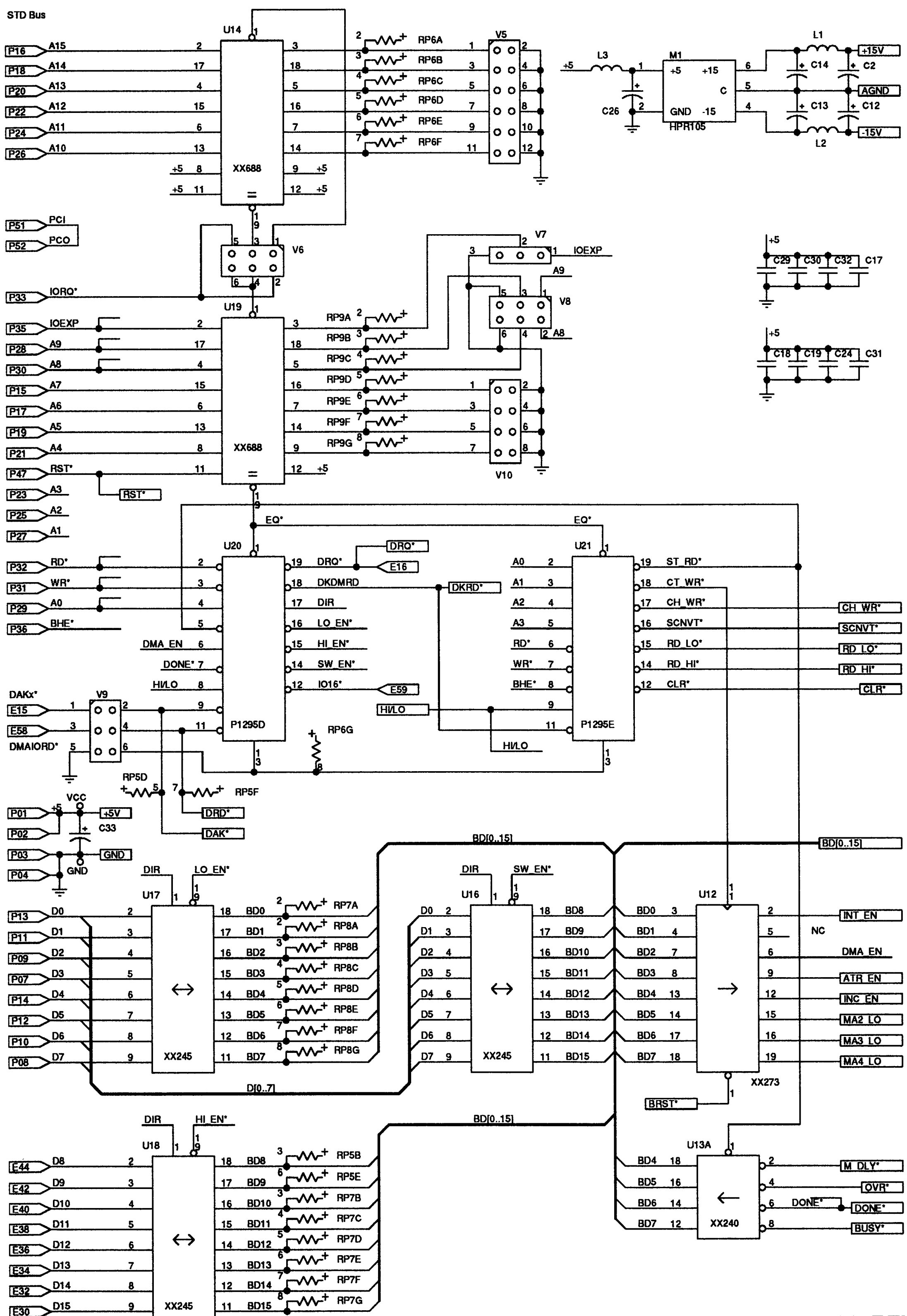


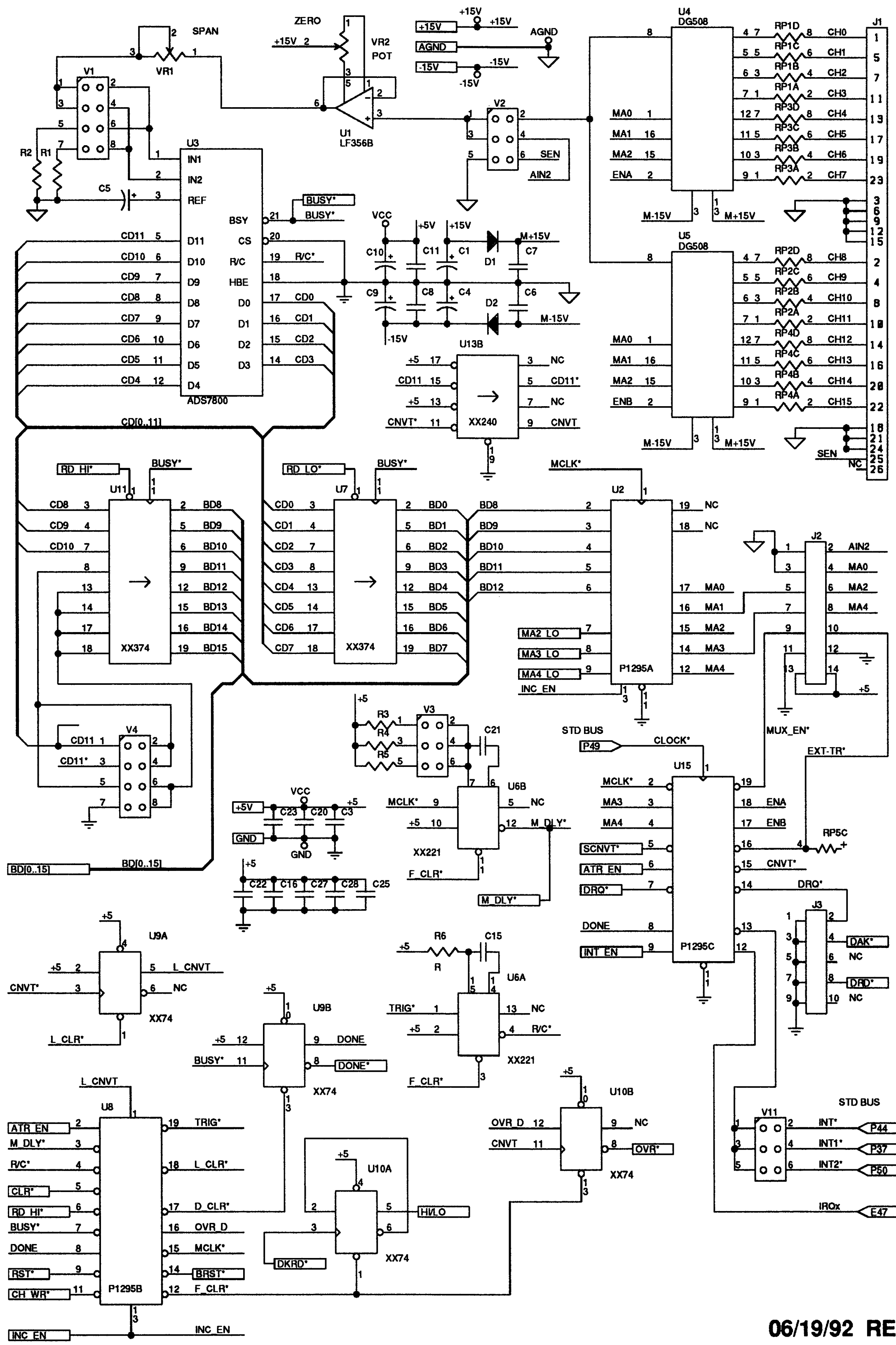
Figure 7-13. VL-1295 Parts Placement Diagram

VL-1295 Schematic



06/19/92 REV 3

VL-1295 Schematic



06/19/92 REV 3

VL-1295 Parts List

Rev. 3.01

Capacitors

C1,C2,C4,C12-C14	1 µf tantalum
C5,C9,C10	10 µf 16V tantalum
C3,C6,C7,C8,C11,C16-C20,C22-C25,C27-C32	.01 µf ceramic
C26	220 µf electrolytic
C15	22 pf NPO ceramic
C21	1500 pf NPO ceramic
C33	22 µf electrolytic

Inductors

L1,L2,L3	10 µH, 250 ma

Integrated Circuits

M1	HPR105
U1	LF356B
U2	16V8-25QP IP1295A Rev. 0.00
U3	ADS7800KP
U4,U5	DG408DJ
U6	74HCT221
U7,U11	74HCT374
U8	16V8-25QP IP1295B Rev. 4.01
U9,U10	74HCT74
U12	74HCT273
U13	74AHCT240
U14,U19	74HCT688
U15	16V8-25QP IP1295C Rev. 3.01
U16,U17,U18	74ACT245
U20	16V8-25QP IP1295D Rev. 3.00
U21	16V8-25QP IP1295E Rev. 2.00

Resistors

R1	49.9 Ω, 1%, ¼W
R2	249 Ω, 1%, ¼W
R3	2.2K Ω, 5% ¼W
R4	20K Ω 5% ¼W
R5	100K Ω, 5%, ¼W
R6	3.3K Ω, 5% ¼W
RP1-RP4	1K Ω, 4 res., SIP
RP5-RP9	100K Ω, 7 res., SIP
VR1	100 Ω, 15 turn pot
VR2	100K Ω, 15 turn pot

Semiconductors

D1,D2	1N4148 diode
D3,D4	1N5817 diode

Miscellaneous

J1	26 pin right angle header
J2	14 pin right angle header
J3	10 pin right angle header

VL-12CT95 Parts List

Rev. 3.01

Capacitors

C1,C2,C4,C12-C14	1 µf tantalum
C5,C9,C10	10 µf 16V tantalum
C3,C6,C7,C8,C11,C16-C20,C22-C25,C27-C32	.01 µf ceramic
C26	220 µf electrolytic
C15	22 pf NPO ceramic
C21	1500 pf NPO ceramic
C33	22 µf electrolytic

Inductors

L1,L2,L3	10 µH, 250 ma

Integrated Circuits

M1	HPR105
U1	OPA627AP
U2	16V8-25QPI IT1295A Rev. 0.00
U3	ADS7800BH
U4,U5	DG408DJ
U6	74HCT221
U7,U11	74HCT374
U8	16V8-25QPI IT1295B Rev. 4.01
U9,U10	74HCT74
U12	74HCT273
U13	74AHCT240
U14,U19	74HCT688
U15	16V8-25QPI IT1295C Rev. 3.01
U16,U17,U18	74ACT245
U20	16V8-25QPI IT1295D Rev. 3.00
U21	16V8-25QPI IT1295E Rev. 2.00

Resistors

R1	49.9 Ω, 1%, ¼W
R2	249 Ω, 1%, ¼W
R3	2.2K Ω, 5% ¼W
R4	20K Ω 5% ¼W
R5	100K Ω, 5%, ¼W
R6	3.3K Ω, 5% ¼W
RP1-RP4	1K Ω, 4 res., SIP
RP5-RP9	100K Ω, 7 res., SIP
VR1	100 Ω, 15 turn pot
VR2	100K Ω, 15 turn pot

Semiconductors

D1,D2	1N4148 diode
D3,D4	1N5817 diode

Miscellaneous

J1	26 pin right angle header
J2	14 pin right angle header
J3	10 pin right angle header