

# VME READOUT BUFFER / FANOUT

Date: 6/20/97

Mark Bowden, Ted Zmuda, Thinh Pham,  
Hector Gonzalez, Daniel Mendoza\*,  
Marvin Johnson & Ed Barsotti  
Fermilab

\*Universidad de Los Andes/Fermilab

Document # ESE-SVX-950719

General Information .....	4
Description .....	4
Application .....	4
Requirements .....	5
Operating Modes .....	6
Control Logic .....	7
VME Logic .....	7
Processor .....	7
Shared Memory .....	9
Input Data Ports .....	9
Receive Logic .....	10
Event Buffers .....	10
Event Monitor .....	11
Diagnostic/Development Software .....	12
Self-Test Features .....	12
Programming of Non-volatile Memory .....	12
Programming of 68EC030 Processor .....	13
Programming of FPGAs .....	13
Interface Specifications .....	13
System Controller Interface .....	13
Monitor Interface .....	13
VME Interface .....	14
P5/6 Control Bus .....	14
Data Link Interface .....	18
G-Link Data Link .....	18
G-Link Signal Descriptions .....	18
G-Link Synchronization .....	19
TAXI Data Link .....	21
Diagnostic Interface .....	22
Electrical & Mechanical Specifications .....	23
Packaging .....	23
PC Board Construction .....	24
Power Requirements .....	24
System Applications .....	24

Generic SVX Application .....	24
CDF SVX Application .....	28
System Controller (CDF SVX) .....	29
Output Data Format (CDF SVX) .....	30
VME Addressing (CDF SVX) .....	30
DO SVX Application .....	33
System Controller (D0 SVX) .....	33
Output Data Format (D0 SVX) .....	33
VME Addressing (D0 SVX) .....	34
CDF Calorimeter Application .....	35
VRB Fanout .....	36

## **General Information**

This document describes the "VME Readout Buffer" (VRB), a module designed initially for use in the D0 and CDF detector data acquisition systems. Information that is specific to a particular application is provided in System Applications.

In this specification the terminology "READOUT BUFFER" indicates the VRB buffer which will be used to store data received on the input ports (write ports). "SCAN BUFFER" indicates the VRB buffer which will be accessed during a VME read operation (read port).

## **Description**

The VRB is a single-width 9U VME module. The VME interface is used for output of accepted events. The VRB receives data via transition module data links, which are typically serial optical connections, and communicates with a System Controller through the P5/6 connector.

The VRB can accept input data at a combined rate of approximately 500 MBytes/sec on ten (byte-wide) channels. The output rate is limited by VME transfer speeds and by the number of VRB modules sharing the VME bus. To make optimum use of the module, a significant trigger rejection factor between input and output event rates is assumed.

The VRB buffer memory is partitionable, allowing a trade-off between buffer size and number of buffers. The VRB normally does not implement its own buffer management, which is delegated to the System Controller module.

## **Application**

The main function of the VRB is to provide Level 2 buffering for the SVX silicon readout system. A set of VRB modules reside in a VME subrack along with one System Controller or Controller Fanout Module. The System Controller communicates with the VRB through a special J3 backplane, used to send the buffer number for event readout and scan and to monitor operational status.

Data received on each link is stored in the buffer pointed to by the READOUT BUFFER number supplied by the System Controller. The beginning and end of data transfer is application dependent.

For the SVX system, the events stored by the VRB are events which have been accepted by the Level 1 trigger and are waiting for a Level 2 accept or reject. For a reject, the buffer is re-used when the System Controller requests an overwrite (i.e. re-uses the buffer number). Events that are accepted by the Level 2 trigger are accessed via the VME interface with the System Controller supplying the SCAN BUFFER

number.

For non-SVX applications, either of the VRB ports may be programmed to operate in FIFO mode. In this mode the readout and/or scan buffer numbers are incremented automatically. Buffer numbers supplied by the System Controller are ignored.

Use of the System Controller module is optional. Information normally supplied by the System Controller through the P5/6 connector can also be provided through VME if the control traffic will not interfere with data transfer.

## Requirements

The specifications for the VRB module are based on the SVX silicon detector readout rates specified by DO and CDF. For SVX applications, the VRB must input data at the L1 Accept Rate, within the given SVX readout time, and must output data at the L2 Accept Rate. The following table shows typical operating parameters.

L1 Trigger Decision Time	2.5 microseconds
L1 Accept Rate	50 KHz
L1 (SVX) Readout Time	10 microseconds
L2 Accept Rate	1-2 KHz
L3 Accept Rate	100 Hz
Event Size (average per VRB channel)	300 Bytes
Event Size (average per VRB)	3 KBytes

VRB design requirements include:

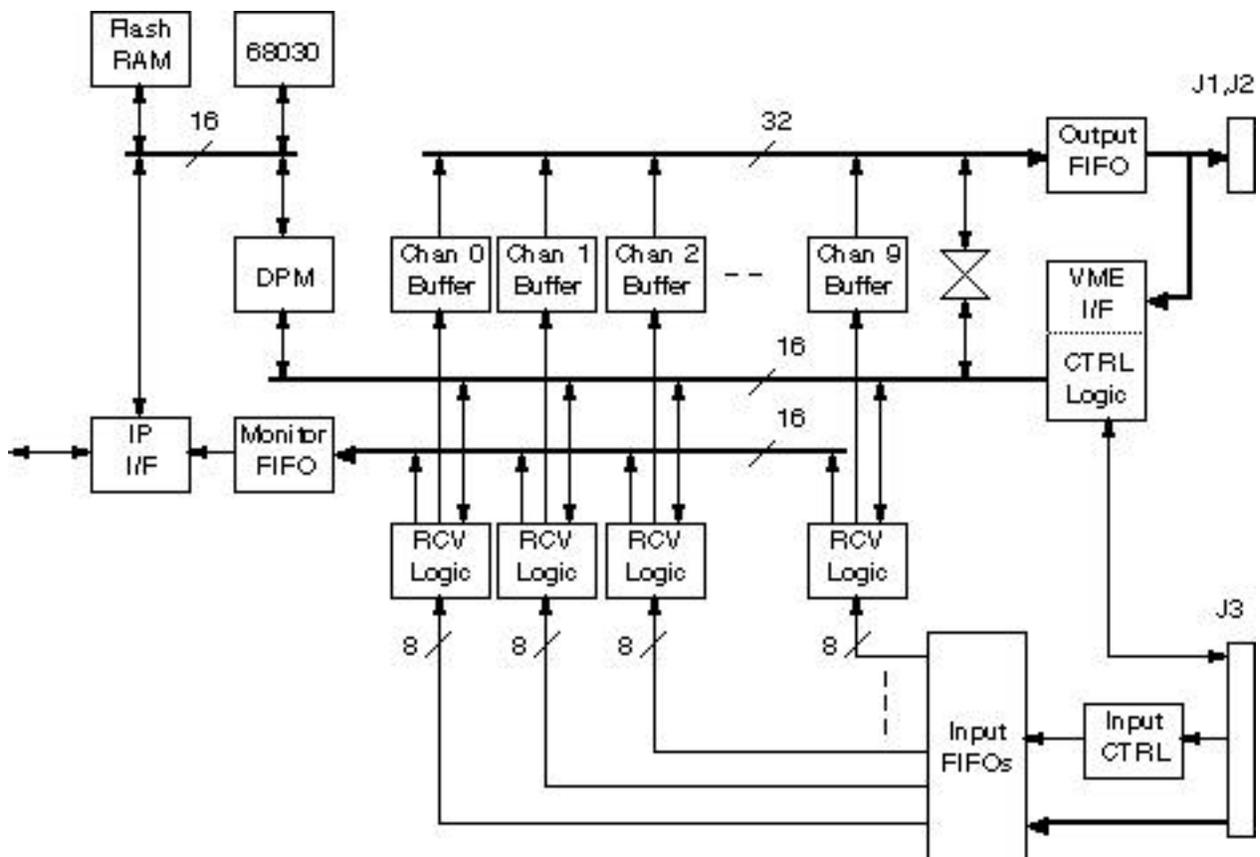
- 9U x 400mm VME board.
- VME64 slave that complies with the standard (ANSI/IEEE STD 1014 and VME 64 Draft Specifications, Rev. 11.1). The block transfer rate must be a minimum of 40 MBytes/sec.
- Application specific P5/6 connector for external buffer management and status monitoring.
- Data links that can accept data from up to ten input channels. Each channel can accept data at 53 MBytes/sec.

- Programmable buffer size to support CDF and D0 L2 and L3 readout systems.
- Simultaneous read and write of event data as long as the buffers are different.
- Control and status registers that support debugging, configuration and integration into CDF and D0 DAQ systems.

Each VRB channel receives a stream of event data from the front-end system and buffers that data for VME readout to the Level 3 trigger system. The VME connection is shared by all VRB modules in a subrack.

## Operating Modes

A block diagram of the VRB is shown in the following figure. The individual functional blocks are explained in this section.



## **Control Logic**

The VRB Control Logic performs three basic functions:

- receive and process messages from the System Controller or VME port
- generate status signals to the System Controller
- manage the flow of data to and from the VRB buffers

Messages from the System Controller are received on the P5/6 connector and buffered in a small FIFO where they are processed in the order received. Messages consist of one or more 8 bit words (each with an accompanying 4 bit identifier) and are application dependent. The Control Logic will respond to a READOUT or SCAN message with either "READOUT BUSY" or "SCAN BUSY" within approximately 200 nsec.

When the VRB Control Logic receives a message specifying the next input (readout) buffer, it looks up the buffer starting address and maximum buffer size (in the shared memory), and broadcasts this information to the Receive Logic controllers for all channels. When all event data is received, the VRB Control Logic will read the individual byte counts from each Receive Logic controller and generate a global byte count for the event. The global byte count is available to a VME scan controller so that it can perform a single block read operation to obtain all data for the event. During scan operations, the individual channel byte counts are used by the VRB Control Logic to initialize a DMA controller which concatenates and transfers data blocks to the VME output FIFO.

## **VME Logic**

The VME slave interface logic is incorporated with the VRB Control Logic. A description of the VME address space is found in VME Interface. The VME logic is entirely programmable and can support a variety of existing or future VME modes within the limitation of available FPGA resources.

## **Processor**

The VRB uses a Motorola 68EC030 processor to initialize on-board programmable logic to set the default operating parameters. This information is stored in a non-volatile memory attached to the processor. During operation, the processor functions only as a controller for the Monitor Port interface. It is not involved directly in

the data readout/scan operations. The processor operates in 16 bit mode.

The VRB processor has access to 1 MByte of non-volatile (Flash) memory. The memory is arranged in two banks. Bank 0 contains default processor startup code and programmable logic configurations to allow communication with the VRB through the VME and Monitor ports. This information is normally changed only for VRB version number revisions and should be common to all VRB applications. Bank 1 contains processor code and programmable logic configurations which are application specific, along with VRB parameters which may change occasionally. Both banks may be write protected by on-board configuration switches. The following information is typically stored in non-volatile memory:

### **Bank 0**

- Default processor code
- Drivers for the Monitor Port
- Programmable logic initialization code
- Default Programmable logic configurations

### **Bank 1**

- Application specific processor code
- Application specific programmable logic configurations
- Default Channel enables
- Channel scan sequence
- Status signal enables
- Status register enables
- Buffer start addresses
- Buffer sizes
- Data record identifiers
- Fixed event header information
- Link configuration
- Link emulation data

A VRB configuration switch enables the Bank 1 memory. At startup the VRB will

initialize with the Bank 0 default information. If the configuration switch is ON, the VRB processor will then transfer control to the Bank 1 program which can reinitialize the programmable logic with application specific information. The Bank 0 program contains routines to allow download of Bank 1 information through the VME shared memory interface.

## Shared Memory

A 16 KByte Dual-port Memory (DPM) allows information to be shared between the Control Logic and Processor. During initialization the processor copies operating parameters (buffer sizes, starting addresses, etc) from non-volatile memory to shared memory for use by the Control Logic. During readout and scan processing, the Control Logic uses the shared memory to store detailed status information and byte counts for each event. The shared memory is also used by the processor for local temporary data storage. Access to this memory from VME is arbitrated by the Control Logic.

## Input Data Ports

The input data links are implemented on a transition module to simplify testing and to allow alternate protocols. The following formats are initially supported:

### *G-Link transition module operating in 16 bit mode:*

- In this mode, each G-Link is divided into two independent VRB channels. Each channel is 8 bits of data with no control bits. There are eight channels (0-7) per VRB. Channels 8 and 9 are not used.

### *G-Link transition module operating in 20 bit mode with control information:*

- In this mode, each G-Link is divided into two independent VRB channels. Each channel is 8 bits of data with 2 control bits. There are eight channels (0-7) per VRB. Channels 8 and 9 are not used.

### *G-Link transition module operating in 20 bit mode without control information:*

- In this mode, each G-Link is divided into two and a half VRB channels. Each channel is 8 bits of data with no control bits defined. There are ten channels (0-9) per VRB. Channels 8 and 9 are each formed by combining the high order 4 bits from two adjacent G-Links.

*TAXI Link transition module operating in 8 bit mode:*

- In this mode, each TAXI link drives a single VRB channel. There are ten channels (0-9). Each channel is 8 bits of data with no control bits defined.

*TAXI Link transition module operating in 10 bit mode:*

- In this mode, each TAXI link drives a single VRB channel. There are ten channels (0-9). Each channel is 8 bits of data with 2 control bits.

Control bits may be used to signal start or end-of-record, valid data, error or other information. These control bits are processed by the Receive Logic in each channel, but are not stored with the event data in the VRB buffers. Up to four G-Links or ten TAXI links are supported.

Input FIFOs are used to decouple the data link/VRB data streams. In addition to the data and user defined control bits, the FIFO also contains link status information which indicates first valid data word in a contiguous packet, frame errors and control word reception. The input FIFOs will hold up to 512 bytes per VRB channel. This allows for some skew between the arrival of the first byte of data and receipt of the readout command. If the event data exceeds 512 bytes per channel, the readout command must be received by the VRB within approximately 6-7 microseconds (512 bytes/53 Mbytes sec - setup overhead) of the start of data transmission or the input FIFO may overflow.

## **Receive Logic**

Event data passes through the Receive Logic before it is stored in the VRB buffers. The Receive Logic program is application dependent. It performs event format verification and can also be used for limited data processing.

The beginning of data input is initiated by the System Controller when it sends a READOUT BUFFER message to the VRB. Any data received before the READOUT BUFFER message will be held in the input FIFOs (up to the 512 byte limit).

## **Event Buffers**

The VRB supports a programmable number of event buffers. Buffer starting locations and sizes are also programmable and are aligned on 8 Byte boundaries.

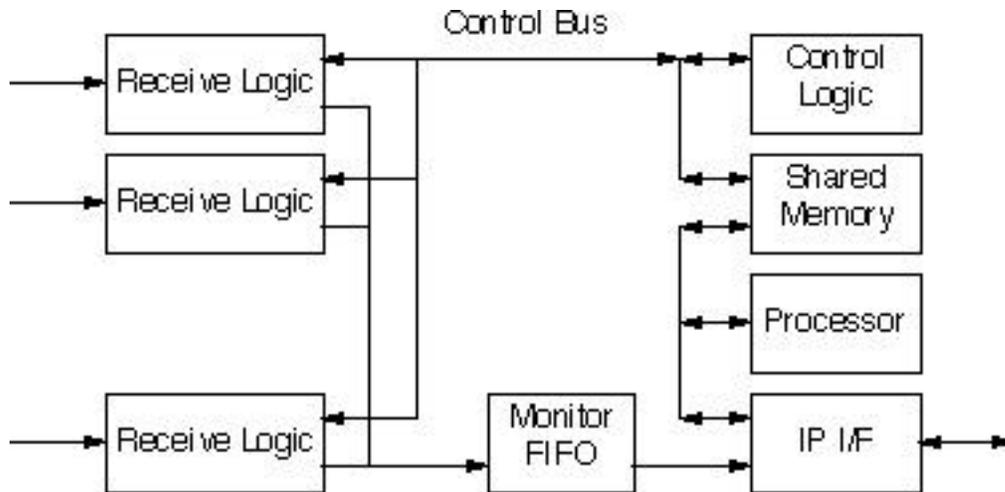
The SARAM memory specified for the VRB provides a total of 32 KBytes per input channel (expandable to 64 KBytes).

The System Controller provides a pointer to indicate which buffer is to be written on input and which buffer is to be read on output. To support simultaneous event read and write, the VRB buffer memory is dual-ported. To accelerate the VME scan operation, the VRB calculates a total byte count for each of the stored events. A VME master determines how much data to transfer by reading the total byte count register, and then performs a block transfer from the event data FIFO port. During the transfer the VRB will output buffers in the selected order, skipping buffers for disabled channels. (Note: disabled channels currently generate a few bytes of filler data....this will be deleted in another revision.)

## **Event Monitor**

The VRB provides a 4K X 16 bit Monitor FIFO which allows sampling of event data from a single VRB input channel for diagnostics or statistical evaluation. The information can then be read by the processor and transmitted via the VRB Monitor Port (or indirectly through VME). The Monitor Port consists of a control FPGA and mezzanine connectors for a dual width Industry Pack. In the D0 SVX application, this mezzanine card will likely hold a 1394 serial interface. In other applications it may be used as an Ethernet interface. Monitor FIFO data may be transferred to the Monitor Port at a relatively high rate (using DMA logic in the FPGA). Normal reads and writes to this port from the processor will be slower.

The processor requests an event by writing the selected channel number and an enable bit to the shared memory. The Control Logic will poll this information when initializing the Receive Logic at the start of each event readout. At the completion of event readout, the Control Logic copies the byte counts for all channels to the shared memory, and clears the enable bit to indicate that the requested event data is available to the processor. The Monitor FIFO may be enabled at any time, but does not begin recording data until the start of the next event readout.



## Diagnostic/Development Software

### Self-Test Features

This subsection describes the features and data paths in the VRB that can be used for diagnostic purposes.

If no System Controller module is available, a VME master can emulate the Controller by writing READOUT and SCAN messages directly to the VRB control registers. If no data sources are available, the VRB channels can be individually configured for emulation mode (in this mode the Receive Logic in each channel generates an incrementing data pattern, from 0 to 7F, in response to a READOUT command).

### Programming of Non-volatile Memory

The Flash memory used in the VRB is rated for 10,000 write cycles and can only be written in blocks. If a single word in the memory is to be changed, the entire block (512 bytes) must be rewritten. Delay between writing any two words in a block should not exceed 150  $\mu$ sec. A delay of 10 msec is required for completion of the block write, following the last word transferred.

Because of these restrictions, the Flash memory is not directly accessible through VME, but may be updated by the on-board processor in response to requests from VME.

## **Programming of 68EC030 Processor**

The 68EC030 code is written in C (with a small assembly language segment to define the exception vector table). The output of the C compiler is then merged with the code from the FPGA compiler to form a single object file which can be downloaded to the VRB Flash memory.

(Note: the C compiler currently being used does not allow specification of data sections, so the initial version of the VRB software is written in a way that avoids the need for a linker....by using absolute addressing and inline functions. In the future, it is expected that the VRB embedded software will be ported to the GNU compiler.)

## **Programming of FPGAs**

The Control Logic, Receive Logic and Monitor Interface FPGAs are Altera 8000 series components and are programmed using the Altera MAX PLUS+ development system. The original FPGA code is written entirely in AHDL and the Altera ".tff" files are then converted to assembly language and merged with the 60EC030 code.

## **Interface Specifications**

This subsection provides a detailed description of all VRB interfaces.

### **System Controller Interface**

A System Controller is not required for VRB operation, but it will improve performance in most applications by removing control traffic from the VME bus during data acquisition. It also allows a centralized interface to the trigger system so that the event scan processor can access events in order without knowing the event number.

The System Controller interface is implemented on the P5/6 connector. The following figure shows the interconnection between the System Controller and the VRB subracks/modules. The Controller will drive a low-voltage differential signal (LVDS) cable to multiple VRB Fanout modules which will translate these signals to TTL on the J3 backplane of VRB subracks. Alternatively, a separate System Controller can be placed in each VRB subrack without the need for the VRB Fanout module.

### **Monitor Interface**

The monitoring system interface is intended to provide an auxiliary control and diagnostics port which is independent of the VME bus. This minimizes interference

with VME data transfers for systems which support this interface. All communication with the VRB through this port is handled by the VRB processor.

### VME Interface

The supported VME transfer modes and address space are determined by the application. See System Applications.

### P5/6 Control Bus

The pinout for control/status signals on the VRB P5/6 connector is provided in the following table;

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	.	.	msg0	.	.
2	.	.	msg1	.	.
3	.	.	msg2	.	.
4	.	.	msg3	.	.
5	.	.	msg4	.	.
6	.	.	msg5	.	.
7	.	.	msg6	.	.
8	.	.	msg7	.	.
9	.	.	msg8	.	.
10	.	.	msg9	.	.
11	.	.	msg10	.	.
12	.	.	msg11	.	.
13	.	.	msg_strobe	.	.
14	.	.	status0	.	.
15	.	.	status1	.	.
16	.	.	status2	.	.
17	.	.	status3	.	.
18	.	.	status4	.	.

19	.	.	status5	.	.
20	.	.	status6	.	.
21	.	.	status7	.	.
22	.	.	status8	.	.
23	.	.	status9	.	.
24	.	.	reserved	.	.
25	.	.	reserved	.	.
26	.	.	GND	.	.
27	.	.	dclk+	.	.
28	.	.	dclk-	.	.
29	.	.	GND	.	.
30	.	.	mclk+	.	.
31	.	.	mclk-	.	.
32	.	.	GND	.	.
33	.	.	.	.	.
34	.	.	.	.	.
35	.	.	.	.	.
36	.	.	.	.	.
37	.	.	.	.	.
38	.	.	.	.	.
39	.	.	.	.	.
40	.	.	.	.	.
41	.	.	.	.	.
42	.	.	.	.	.
43	.	.	.	.	.
44	.	.	.	.	.
45	.	.	.	.	.
46	.	.	.	.	.
47	.	.	.	.	.

Terminations for the MSG and STATUS bus signals are handled in the same way as signals on the J1/J2 backplane.

Control signals to the VRBs, as implemented on the J3 backplane:

- MSG[11:0]: TTL signals used to transmit control messages from the System Controller to the VRBs.
- MSG\_STROBE: TTL signal used to indicate that the System Controller has placed a new control message on the backplane.

The message data (MSG[11:0]) is written into the message FIFO on the rising edge of the MSG\_STROBE signal. Message data should be valid for at least 15 ns (setup time) before the strobe.

Status signals from the VRBs, as implemented on the J3 backplane (paranetical signal definitions are for the SVX application). These signals are inverted by the VRB Fanout module:

- STAT0 (READOUT\_BUSY\*): Open collector TTL signal used to indicate that all data from the current input event has been received. READOUT\_BUSY\* is driven low when a READOUT BUFFER message is received by the VRB and is released when all active channels on the VRB have received the event data. This signal should be treated as asynchronous by the System Controller. Note: this protocol does not allow pipelining of READOUT BUFFER messages....the System Controller must wait until READOUT\_BUSY\* is received before sending the next READOUT BUFFER message. READOUT\_BUSY\* will be driven low for a minimum of 300 nsec in response to the READOUT BUFFER message.
- STAT1 (SCAN\_BUSY\*): Open collector TTL signal used to indicate that all data from the current output event has been transmitted. SCAN\_BUSY\* is driven low when a SCAN BUFFER message is received by the VRB and is released when the VRB has been read out through VME. This signal should be treated as asynchronous by the System Controller. Note: this protocol does not allow pipelining of SCAN BUFFER messages....the System Controller must wait until SCAN\_BUSY\* is received before sending the next SCAN BUFFER message. SCAN\_BUSY\* will be driven low for a minimum of 300 nsec in response to the SCAN BUFFER message.

- STAT2 (SYNC\_ERROR\*): Open collector TTL signal used by the VRBs to flag a link synchronization error on one of the data links. This signal remains asserted only while the sync error exists. It should be treated as asynchronous by the System Controller.
- STAT3 (FRAME\_ERROR\*): Open collector TTL signal used by the VRBs to flag a non-valid data word on one of the data links. This signal remains asserted while the current event is being processed. It should be treated as asynchronous by the System Controller.
- STAT4 (IDENTIFIER\_ERROR\*): Open collector TTL signal used by the VRBs to flag an invalid event identifier (crossing number, capacitor number, etc.) in the data stream. Event identifiers are application dependent. This signal remains asserted while the current event is being processed. It should be treated as asynchronous by the System Controller.
- STAT5 (FORMAT\_ERROR\*): Open collector TTL signal used by the VRBs to flag that a data format error has occurred. Data format errors indicate markers (channel numbers, chip IDs, etc.) which are outside the valid range, or not in the expected location in the data stream. Format information is application dependent. This signal remains asserted while the current event is being processed. It should be treated as asynchronous by the System Controller.
- STAT6 (CONTROLLER\_ERROR\*): Open collector TTL signal used by the VRBs to flag unrecognizable or invalid System Controller messages. This signal remains asserted while the current event is being processed. It should be treated as asynchronous by the System Controller.
- STAT7 (VRB\_ERROR\*): Open collector TTL signal used by the VRBs to flag all other errors, including buffer overflow. This signal remains asserted while the current event is being processed. It should be treated as asynchronous by the System Controller.
- STAT8 (reserved): Open collector TTL signal. It should be treated as asynchronous by the System Controller.
- STAT9 (reserved): Open collector TTL signal. It should be treated as asynchronous by the System Controller.

The DCLK and MCLK signals are present to allow J3 backplane compatibility with another subsystem in the CDF SVX application. It is possible to drive the VRB internal logic with MCLK instead of the on-board oscillator for applications which might require synchronous system clocking. In addition, DCLK is routed to the VRB for use as a fast trigger input if needed. These signals are otherwise not required for VRB operation.

## Data Link Interface

This subsection describes the front-end electronics interface. This interface is provided by a separate transition module.

### G-Link Data Link

For some applications the data link transition module will use the Finisar optical receiver and HP G-Link serial/parallel converter. A single G-Link receiver may supply data for 2 or 3 VRB channels. These channels are meant to be logically independent, but because of the common G-Link control signals there is some interaction which must be considered in the Receive Logic. The MODID (module ID) pin is left unconnected to identify a G-Link transition module.

### G-Link Signal Descriptions

Name	Signal
STRBOUT*	Recovered Frame-rate Data Clock: This output is the PLL recovered frame rate clock. The G-link signals: D0-D19, DAV*, CAV*, LNKRDY* and ERROR are valid at the rising edge of this clock.
LNKRDY*	Link Ready Indicator: This active-low signal indicates that the startup sequence is completed and that the data and control signals are valid. TA high level on this signal indicates loss of data link synchronization.
DAV*	Data Available: This active-low signal indicates that the G-Link has received a data frame.
CAV*	Control Frame Available: This active-low signal indicates that the Rx chip has received a control frame.
ERROR	Received Data Error: Asserted when a frame is received that does not correspond to either Data, Control or fill frame. Two consecutive ERROR frames with will cause the G-link to deassert LNKRDY.
RESET*	GLink Reset. Connects to the G-Link state machine reset signal and is common to all Glink receivers. It is driven low during VRB power-up or reset.

## G-Link Synchronization

The data links are configured in simplex mode with a system supplied feedback path that informs the G-link transmitters of the synchronization state of the G-link receivers. If SYNC\_ERROR is asserted by the VRB, the System Controller sends a message to the data sources requesting transmission of fill frames. The System Controller then waits until SYNC\_ERROR is deasserted. The specific link causing the SYNC\_ERROR may be located by reading VME registers in the VRBs [to be added]. In some cases it may be necessary to reset the VRB to force G-Link receiver synchronization.

The P5/6 connector pin definitions for a G-Link transition module are listed in the following table.

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	L0_D0	GND	.	GND	L2_D0
2	L0_D1	L0_D2	.	L2_D2	L2_D1
3	GND	L0_D3	.	L2_D3	GND
4	L0_D5	L0_D4	.	L2_D4	L2_D5
5	L0_D6	GND	.	GND	L2_D6
6	L0_D7	L0_D8	.	L2_D8	L2_D7
7	GND	L0_D9	.	L2_D9	GND
8	L0_D11	L0_D10	.	L2_D10	L2_D11
9	L0_D12	GND	.	GND	L2_D12
10	L0_D13	L0_D14	.	L2_D14	L2_D13
11	GND	L0_D15	.	L2_D15	GND
12	L0_D16	L0_D16	.	L2_D16	L2_D16
13	L0_D17	GND	.	GND	L2_D17
14	L0_D17	L0_D18	.	L2_D18	L2_D17
15	GND	L0_D18	.	L2_D18	GND
16	L0_D19	L0_D19	.	L2_D19	L2_D19
17	L0_CAV*	GND	.	GND	L2_CAV*
18	L0_DAV*	L0_LNKRDY*	.	L2_LNKRDY*	L2_DAV*

19	GND	L0_LNKRDY*	.	L2_LNKRDY*	GND
20	L0_STRBOUT	GND	.	GND	L2_STRBOUT
21	L0_STRBOUT*	L0_ERROR	.	L2_ERROR	L2_STRBOUT*
22	GND	L0_ERROR	.	L2_ERROR	GND
23	GND	GND	.	GND	GND
24	GND	GND	.	GND	GND
25	GND	GND	.	GND	GND
26	L1_CAV*	GND	.	GND	L3_CAV*
27	L1_DAV*	L1_LNKRDY*	.	L3_LNKRDY*	L3_DAV*
28	GND	L1_LNKRDY*	.	L3_LNKRDY*	GND
29	L1_STRBOUT	GND	.	GND	L3_STRBOUT
30	L1_STRBOUT*	L1_ERROR	.	L3_ERROR	L3_STRBOUT*
31	GND	L1_ERROR	.	L3_ERROR	GND
32	L1_D0	GND	.	GND	L3_D0
33	L1_D1	L1_D2	.	L3_D2	L3_D1
34	GND	L1_D3	.	L3_D3	GND
35	L1_D5	L1_D4	.	L3_D4	L3_D5
36	L1_D6	GND	GND	GND	L3_D6
37	L1_D7	L1_D8	GND	L3_D8	L3_D7
38	GND	L1_D9	GND	L3_D9	GND
39	L1_D11	L1_D10	GND	L3_D10	L3_D11
40	L1_D12	GND	GND	GND	L3_D12
41	L1_D13	L1_D14	.	L3_D14	L3_D13
42	GND	L1_D15	.	L3_D15	GND
43	L1_D16	L1_D16	.	L3_D16	L3_D16
44	L1_D17	GND	.	GND	L3_D17
45	L1_D17	L1_D18	.	L3_D18	L3_D17
46	GND	L1_D18	RESET	L3_D18	GND
47	L1_D19	L1_D19	MODID	L3_D19	L3_D19

## TAXI Data Link

For some applications the data link transition module will use the AMD TAXI serial/parallel converter. The TAXI links operate at a maximum rate of approximately 15 Mbytes/sec. The MODID (module ID) pin is tied to GND to identify a TAXI transition module. P5/6 connector pin definitions for a TAXI transition module are listed in the following table.

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	L0_D0	GND	.	GND	L4_D0
2	L0_D1	L0_D2	.	L4_D2	L4_D1
3	GND	L0_D3	.	L4_D3	GND
4	L0_D5	L0_D4	.	L4_D4	L4_D5
5	L0_D6	GND	.	GND	L4_D6
6	L0_D7	L1_D0	.	L5_D0	L4_D7
7	GND	L1_D1	.	L5_D1	GND
8	L1_D3	L1_D2	.	L5_D2	L5_D3
9	L1_D4	GND	.	GND	L5_D4
10	L1_D5	L1_D6	.	L5_D6	L5_D5
11	GND	L1_D7	.	L5_D7	GND
12	L0_D8	L8_D0	.	L9_D0	L4_D8
13	L0_D9	GND	.	GND	L4_D9
14	L8_D1	L1_D8	.	L5_D8	L9_D1
15	GND	L8_D2	.	L9_D2	GND
16	L1_D9	L8_D3	.	L9_D3	L5_D9
17	L0_DSTRB	GND	.	GND	L4_DSTRB
18	L1_DSTRB	L0_BUSY	.	L4_BUSY	L5_DSTRB
19	GND	L1_BUSY	.	L5_BUSY	GND
20	L0_CLOCK	GND	.	GND	L4_CLOCK

21	L1_CLOCK	L0_VLTN	.	L4_VLTN	L5_CLOCK
22	GND	L1_VLTN	.	L5_VLTN	GND
23	L8_CLOCK	GND	.	GND	L9_CLOCK
24	L8_DSTRB	L8_VLTN	.	L9_VLTN	L9_DSTRB
25	GND	L8_BUSY	.	L9_BUSY	GND
26	L2_DSTRB	GND	.	GND	L6_DSTRB
27	L3_DSTRB	L2_BUSY	.	L6_BUSY	L7_DSTRB
28	GND	L3_BUSY	.	L7_BUSY	GND
29	L2_CLOCK	GND	.	GND	L6_CLOCK
30	L3_CLOCK	L2_VLTN	.	L6_VLTN	L7_CLOCK
31	GND	L3_VLTN	.	L7_VLTN	GND
32	L2_D0	GND	.	GND	L6_D0
33	L2_D1	L2_D2	.	L6_D2	L6_D1
34	GND	L2_D3	.	L6_D3	GND
35	L2_D5	L2_D4	.	L6_D4	L6_D5
36	L2_D6	GND	L8_D8	GND	L6_D6
37	L2_D7	L3_D0	L8_D9	L7_D0	L6_D7
38	GND	L3_D1	GND	L7_D1	GND
39	L3_D3	L3_D2	L9_D8	L7_D2	L7_D3
40	L3_D4	GND	L9_D9	GND	L7_D4
41	L3_D5	L3_D6	.	L7_D6	L7_D5
42	GND	L3_D7	.	L7_D7	GND
43	L8_D4	L2_D8	.	L6_D8	L9_D4
44	L2_D9	GND	.	GND	L6_D9
45	L8_D5	L3_D8	.	L7_D8	L9_D5
46	GND	L8_D6	RESET	L9_D6	GND
47	L3_D9	L8_D7	MODID	L9_D7	L7_D9

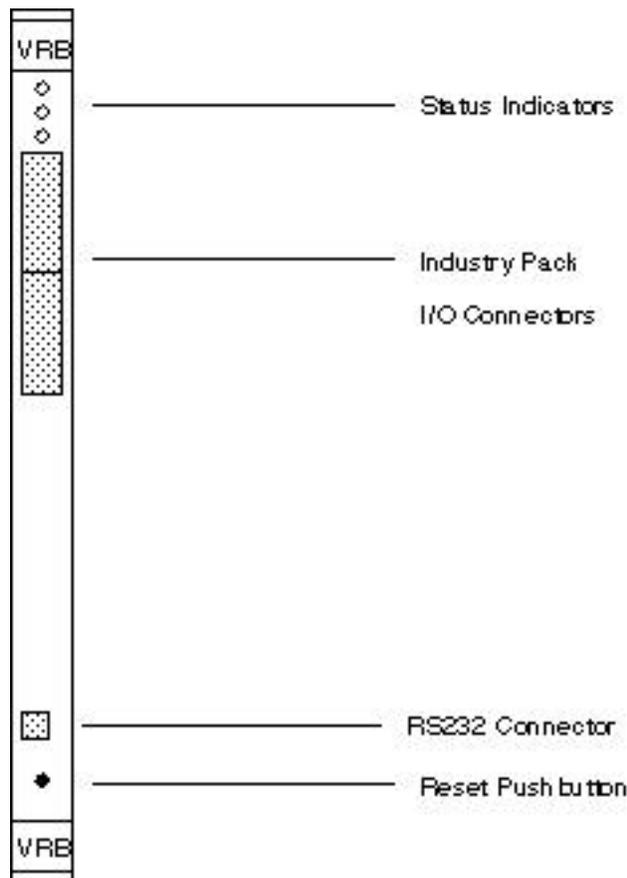
## Diagnostic Interface

A simple 9600 Baud, RS232 compatible terminal port is present on the front panel of the VRB. This port is driven directly by the processor, using a software emulation of a UART. It is used to display startup and configuration information about the VRB along with any malfunctions detectable by the processor.

## Electrical & Mechanical Specifications

### Packaging

The VRB is a 9U x 400mm VME module. The front panel of the VRB is shown in the following figure.



## PC Board Construction

The VRB printed circuit board is FR4 with a thickness of approximately .092". The top and bottom card edges are milled to .063". The board is 10 layers with 50 ohm controlled impedance.

## Power Requirements

Power consumption is estimated at 50-60 watts for the VRB and 20 watts for the transition module. This is highly dependent on operating frequency. In an idle state, the VRB power consumption may be less than 15 watts. The module is fused at +5V, 15A; +12V,2A; -12V,2A. The 12 volt supplies are used only by the IndustryPack mezzanine card and will rarely be needed.

## System Applications

Configuration information for the applications listed in this section is resident in the VRB Flash memory. Application selection is made by setting DIP switches on the VRB printed circuit board. The setting of these switches may be read back through VME for remote verification (register 0x0002). The following applications are currently defined;

Application	S3-8	S3-7	S3-6	S3-5	Readback Value
D0 SVX	ON	ON	ON	ON	xx0x
CDF SVX	ON	ON	ON	OFF	xx1x
CDF Cal	ON	ON	OFF	ON	xx2x

## Generic SVX Application

The silicon readout system described in this document is designed to perform an event data readout of the silicon detectors being designed for CDF and D0. The front-end electronics is based on the SVX IC currently under development.

The SVX IC implements 32 storage cells per channel and 128 channels per IC. The performance characterization of this IC is done with a 132 nsec crossing rate in anticipation of accelerator upgrades. The per channel storage capacity of the SVX IC allows the L1 trigger to make a decision without stopping the analog acquisition of the silicon detector. At the front-end, SVX ICs are mounted on a hybrid and behave like a

single source of data for readout purposes. Upon a Level 1 accept (L1A), the readout control electronics instructs the SVX ICs to digitize the appropriate analog cell, and transmit any samples above a pre-programmed threshold.

The data acquisition control of the SVX IC and the VRB modules is provided by a System Controller module. The controller also provides the upper level functions to interface the silicon detector data acquisition system with the trigger system, including the level 3 readout. Differences in implementation of the trigger distribution, buffer management and readout systems lead to the use of different controllers. The Controller is in charge of a set of VRBs and the control signals are implemented on an application specific J3 backplane. On a Level 1 trigger the Controller provides the VRB with the buffer number to store the event data. The readout electronics will route data from the front-end via the data links to the VRB module. The events that are accepted by the Level 2 trigger are transmitted to Level 3 by a VME Scan Controller.

The first bytes of each data stream contains a header inserted by the front-end electronics to identify the data source. It's meaning is experiment dependent. SVX event data is logically organized by words (2 bytes). For an example of SVX IC ID, status, channel number and data, refer to the following table.

IHDR0
IHDR1
IHDR2
IHDR3
Chip ID
Status
Channel #
Data
Channel #
Data
:
Chip ID
Status
Channel #
Data

:
EOR
EOR
EOR
EOR

Because the data streams in each link are independent, the transferred number of bytes can be different and the two streams sharing the same link are required to provide separate End-of-record (EOR) signals. The VRB input FIFO will continue to accept data from a G-link while the G-Link Data Valid signal is asserted. The Receive Logic will continue to process the data until it recognizes EOR. It will then discard data until the next Start of Record. Start of Record may be identified by a rising transition on the G-Link Data Valid signal or by a data word containing a recognizable start of record symbol. Once all the channels transmitting data to a VRB are done, the VRB will inform the Controller. This is accomplished by releasing the READOUT\_BUSY\* signal on the J3 backplane. READOUT\_BUSY\* is an open collector signal that can be driven low by any of the VRBs in the subrack. The event readout at the subrack level is finished when all VRBs have released this line. When readout is complete, it takes a few microseconds to save the event byte count and status information before the VRB releases it's READOUT\_BUSY\* signal and the event is available for scan.

The VRB Receive Logic checks the event data as it is being received. The following is a list of the checks performed for SVX data [not yet implemented].

- check that the header bytes are received before the data.
- check that the SVX IC control bytes, Chip ID, Channel # and first EOR are at the correct (even byte) locations.
- (optional) check that values for Chip ID and Channel # are ordered
- check that the pipeline capacitor number (if used) and the bunch crossing number match values received from the System Controller in the READOUT message
- on receipt of EOR, append EOR bytes to the event until the event size per channel is a multiple of 8 bytes. This condition is enforced to simplify VME block transfers.
- if the readout is truncated due to buffer overflow or format error, a special EOR code is appended.

Most of the VRB VME address space is common to both D0 and CDF SVX applications. All registers except those in the range 0x0010-0x003E are implemented in the VRB shared memory. All registers are d16 unless otherwise indicated. The following registers are currently defined. Additional status registers will be added later.

<b>VME Address</b>	<b>Function</b>
0x0000	Module ID = 3
0x0002	Configuration Switch setting
0x0004	Date code
0x0006-0x000E	(reserved)
0x0010-0x0016 (d16,d32,d64)	Output data FIFO (non-pipelined mode)
0x0018-0x001E (d16,d32,d64)	Output data FIFO (pipelined mode)
0x0020	(reserved)
0x0022	Readout buffer number
0x0024	Readout pipeline capacitor number (if implemented)
0x0026	Readout bunch crossing number
0x0028	Scan buffer number
0x002A	Scan event number
0x002C-0x002E	(reserved)
0x0030	Scan byte count
0x0032-0x003A	(reserved)
0x003C	VRB reset
0x003E	(reserved)
0x0040	Monitor channel
0x0042	Monitor event byte count
0x0044-0x006E	(reserved)
0x0070	Channel enables
0x0072	Emulation mode enables

0x0080-0x0092	Channel 0-9 configuration words
0x0094-0x009E	(Reserved)
0x00A0-0x00B2	Channel 0-9 Status0 enables
0x00B4-0x00BE	(reserved)
0x00C0-0x00D2	Channel 0-9 Status1 enables
0x00D4-0x02FE	(reserved)
0x0300-0x037E	Buffer 0-63 Start Address
0x0380-0x03FE	Buffer 0-63 Size (bytes)
0x0400-0x07FE	General VRB/VME communication
0x0800-0x1FFE	(Reserved)
0x2000-0x3FFE	VRB working memory

### CDF SVX Application

The local control of the SVX ICs is provided by the Port Card (PC) and the connection between PC and the hybrid is provided by the High Density Interconnect (HDI). The HDI is used to supply the SVX ICs with power and control signals. It also provides a data readout bus that is common to the control bus.

In this application, the G-links are used in 20 bit mode and each transfer packs data from two and a half SVX HDIs. A total of ten HDI channels are supported. Each channel of Receive Logic processes 8 bits of data. The mapping between HDI data channels at the FIB and data channels in the VRB is shown in the following table.

FIB Channel	VRB Channel
A0	0
A1	1
A2	2
A3	3
A4	8
B0	4

B1	5
B2	6
B3	7
B4	9

The reason for this mapping is compatibility with the D0 application of the VRB. The D0 system does not have detector layers for channels A4 and B4, and does not require installation of the Receive Logic/Buffer for channels 8/9. The VRB can be programmed to output buffers in any order, so the internal VRB buffer numbering scheme should be transparent.

### **System Controller (CDF SVX)**

The System Controller for CDF SVX is the SRC. Message types received from the SRC include the following;

- Message type 1: READOUT BUFFER NUMBER.....received after a Level 1 Accept to designate the next buffer number for VRB input.
- Message type 2: PIPELINE CAPACITOR NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 3: BUNCH CROSSING NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 4: SCAN BUFFER.....received after a Level 2 Accept to designate the next buffer number for VRB output.
- Message type 5: EVENT NUMBER.....received after a Level 2 Accept and inserted into the output event record to label the Level 2 event.
- Message type 6-13: (not used)....available for expansion.
- Message type 14: RESET.....causes reset of the input FIFOs and resynchronization of the Receive Logic to the start of the next event. RESET is necessary when changing VRB channel enable or emulation mode registers.

The 12 bit message is interpreted as a four bit field (11:8) identifying the message type and an eight bit field (7:0) containing the message value. The VRB Control Logic may check for proper ordering of message types. The

CONTROLLER\_ERROR signal is asserted if a READOUT BUFFER message is received while the VRB is still asserting READOUT\_BUSY or if a SCAN BUFFER message is received while the VRB is still asserting SCAN\_BUSY.

CDF is expected to use the SVX III IC. The main functional difference between the SVX III and SVX II is that the former implements additional buffers with the required management for concurrent operation of the front-end analog acquisition and digitize/readout of L1 trigger event data.

IDENTIFIER\_ERROR is set when a Pipeline Capacitor Number in the status field of an SVX III IC does not match the expected value, as provided by the SRC. Detection of this error can be individually enabled for each channel and for each SVX IC connected to that channel. Other errors in the received event data will normally generate FORMAT\_ERRORS. [Note: not yet programmed]

### **Output Data Format (CDF SVX)**

The VRB is currently programmed to add the following header information to CDF SVX events:

- 1) two bytes indicating total event byte count
- 2) two bytes indicating the Scan event number
- 3) 10 words of two bytes each indicating individual byte counts from each of the 10 data channels

Zero words are added to the end of each event. If the output data FIFO is empty, subsequent read operations will return this last zero value. The VME Scan Controller can poll the output data FIFO until it returns a non-zero value. This will be the first word (total byte count and event number) of the next scanned event. By polling the output data FIFO, rather than the total byte count register, the Scan Controller can operate independently of the System Controller. Scanned events can be queued in the output data FIFO and read at the maximum rate of the Scan Controller. The Scan Controller does not need to know the event number or scan buffer number.

### **VME Addressing (CDF SVX)**

The CDF SVX application uses A32 VME addressing with the following address modifiers;

- 0x09 or 0D (single word accesses)
- 0x0B or 0F (D32 block transfers)
- 0x08 or 0C (D64 block transfers)

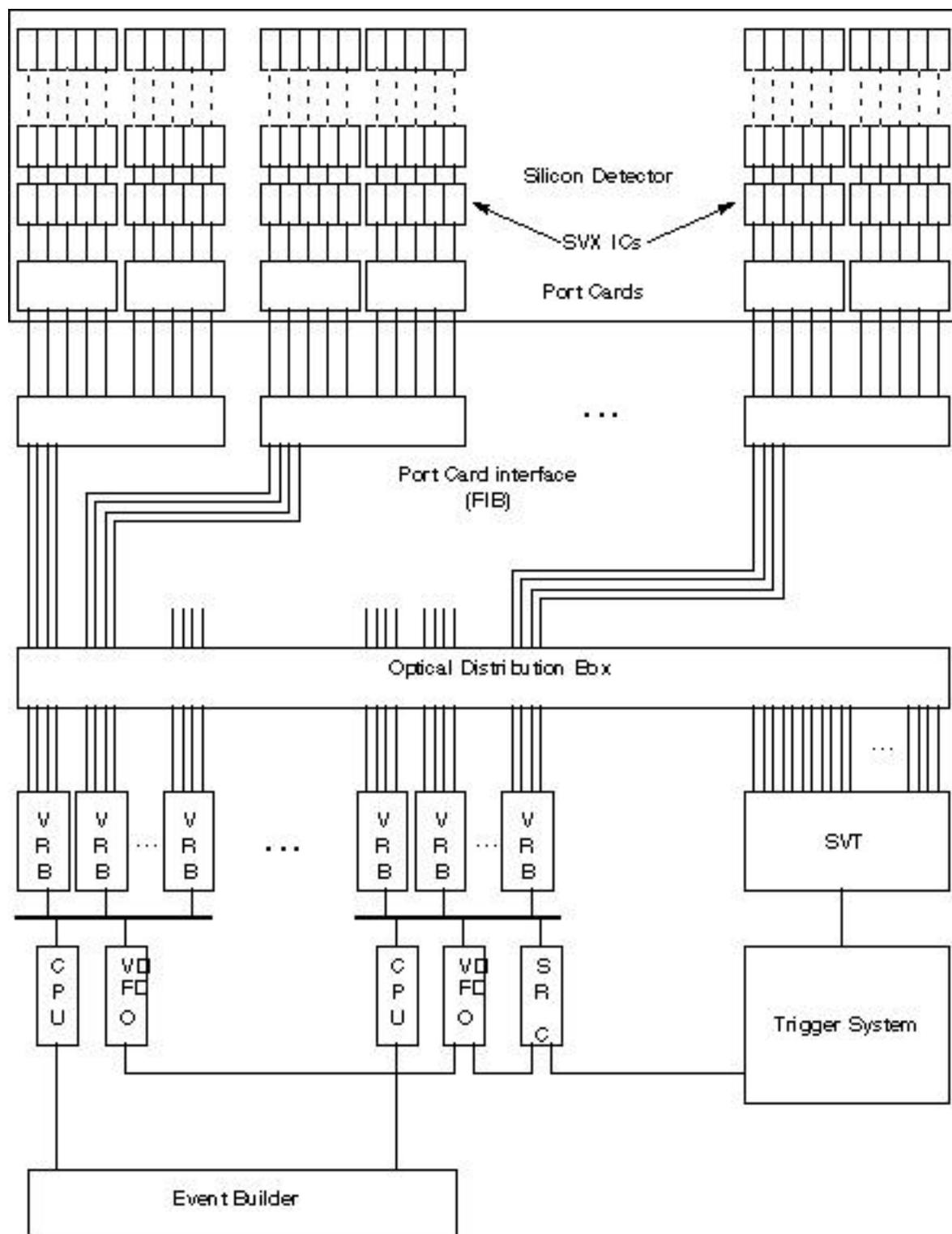
The only VRB address which responds to block transfers is the output data FIFO, which is READ only.

[Note: single word access is currently D16 only. This will be upgraded to include D8 access to comply with VME requirements and possibly D32 access. One exception is the output data FIFO which may be read in single word D32 mode in the current version. The D64 block transfer mode has not yet been programmed.]

The base address is determined by the subrack geographical address which is compared against the VME address (A[31:27]). The geographical address is determined by the P1 connector GA pins defined in the VME64 extensions specification.

<b>Address</b>	<b>A32 Decoding</b>
A31	GA4
A30	GA3
A29	GA2
A28	GA1
A27	GA0
A26-A0	x

A block diagram for the CDF SVX readout system is shown here.



## **DO SVX Application**

For the D0 SVX application, the G-Links operate in 16 bit mode and 8 input channels are used. Each channel processes 8 bits of data. A control bit is also available and is set if the byte is part of a received G-Link control word.

## **System Controller (D0 SVX)**

The System Controller is the SARC. Message types received from the SARC include the following;

- Message type 1: READOUT BUFFER NUMBER.....received after a Level 1 Accept to designate the next buffer number for VRB input.
- Message type 2: (not used).....this is the pipeline capacitor number for SVX III.
- Message type 3: BUNCH CROSSING NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 4: SCAN BUFFER.....received after a Level 2 Accept to designate the next buffer number for VRB output.
- Message type 5: EVENT NUMBER.....received after a Level 2 Accept and inserted into the output event record to label the Level 2 event.
- Message type 6-13: (not used)....available for expansion.
- Message type 14: RESET.....causes reset of the input FIFOs and resynchronization of the Receive Logic to the start of the next event. RESET is necessary when changing VRB channel enable or emulation mode registers.

The 12 bit message input (on the P5/6 connector) is decoded as a four bit field (11:8) identifying the message type and an eight bit field (7:0) containing the message value. The CONTROLLER\_ERROR signal is asserted if a READOUT BUFFER message is received while the VRB is still asserting READOUT\_BUSY or if a SCAN BUFFER message is received while the VRB is still asserting SCAN\_BUSY.

The D0 SVX system is currently based on the SVX II IC. The main difference, as it affects the VRB, is in the recognition of EOR in the data stream.

## **Output Data Format (D0 SVX)**

As currently programmed, the VRB does not add any header information to the event for D0 SVX data. Header information is added by the VBD.

## VME Addressing (D0 SVX)

The D0 SVX application uses either A24 or A32 VME addressing with the following address modifiers;

0X39 or 3D (A24 single word accesses)

0x3B or 3F (A24:D32 block transfers)

0x38 or 3C (A24:D64 block transfers)

0x09 or 0D (A32 single word accesses)

0x0B or 0F (A32:D32 block transfers)

0X08 or 0C (A32:D64 block transfers)

The only VRB address which responds to block transfers is the output data FIFO, which is READ only.

[Note: single word access is currently D16 only. This will be upgraded to include D8 access to comply with VME requirements. One exception is the output data FIFO which may be read in single word D32 mode in the current version. The D64 block transfer mode has not yet been programmed.]

The base address is determined by the subrack geographical address which is compared against the VME address (A[23:19]). The geographical address is determined by the P1 connector GA pins defined in the VME64 extensions specification or by the SGA(4:0) dip switches (if the J1 backplane does not provide geographical addressing).

Address	A24 Decoding	A32 Decoding
A31	x	x
A30	x	x
A29	x	x
A28	x	x
A27	x	x
A26	x	x
A25	x	x
A24	x	x
A23	GA4	GA4
A22	GA3	GA3

A21	GA2	GA2
A20	GA1	GA1
A19	GA0	GA0
A18-A0	x	x

[Note: the A32 geographical address decoding shown here is not consistent with the VME specification and may change. Address decoding for the D0 SVX application is restricted by backward compatibility requirements.]

### **CDF Calorimeter Application**

The VRB is used as a Level 3 buffer in this application. It serves mainly as a data multiplexer. The buffers operate in FIFO mode and no data is discarded in the VRB.

TAXI data links are used. The Receive Logic for each channel generates a BUSY signal when the VRB buffer is “almost full”. The definition of “almost full” and the polarity of the BUSY signal are programmable. This signal may be used to inhibit data transmission on the TAXI link.

## VRB Fanout

The VRB Fanout is a front mounted transition module which connects a System Controller in the same VRB subrack or in a different subrack to the VRB modules. This connection is made using front panel ribbon cable between the System Controller and the VRB Fanout Module. The VRB Fanout Module then repeats the signals on a special J3 backplane in the VRB subracks.

The VRB Fanout Module is a single-width 9U VME module. It resides in slot 15 of an SVX VRB subrack and translates low voltage differential signals (LVDS) on the cable to TTL signals on the J3 backplane. There are three groups of signals; a control bus which is used by the System Controller to send commands to the VRBs, a status bus used by the VRBs to signal completion of event readout and various error conditions, and clock drivers used to operate the VRBs synchronously in some applications.

The VRB Fanout Module and link are designed to allow use of a single System Controller with multiple VRB subracks. In applications where a separate controller is used in each VRB subrack, the VRB Fanout Module is not required.

The VRB Fanout Module translate LVDS cable signals to TTL backplane signals. The Fanout Module must provide termination of each link cable segment.

Although the average data rate for control messages on the Fanout cable is only 200 KHz, transceivers must support a rate of approximately 7 MHz since the System Controller normally provides no message queuing.

The pinout for the VRB Fanout J3 (slot 15) connection is provided in the following table. Pin assignments are selected for compatibility with the CDF FIB J3 backplane. Backplane terminations are handled in the same way as signals on the J1/J2 backplane.

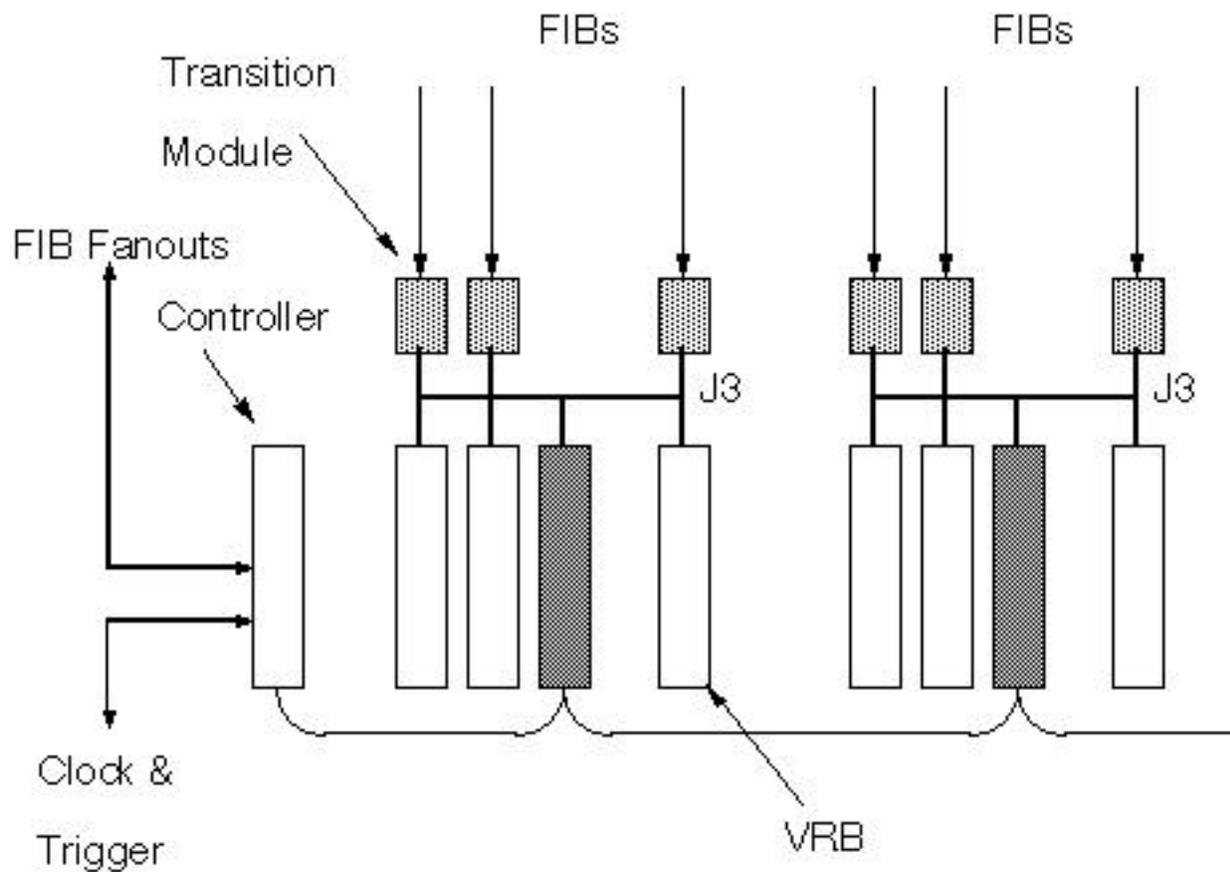
PIN NUMBER	ROW A	ROW B	ROW C	ROW D	ROW E
1	.	mclk9+	msg0	dclk9+	.
2	.	mclk9-	msg1	dclk9-	.
3	.	mclk10+	msg2	dclk10+	.
4	.	mclk10-	msg3	dclk10-	.
5	.	mclk11+	msg4	dclk11+	.
6	.	mclk11-	msg5	dclk11-	.
7	.	mclk12+	msg6	dclk12+	.

8	.	mclk12-	msg7	dclk12-	.
9	.	mclk13+	msg8	dclk13+	.
10	.	mclk13-	msg9	dclk13-	.
11	.	mclk14+	msg10	dclk14+	.
12	.	mclk14-	msg11	dclk14-	.
13	.	mclk16+	msg_strobe	dclk16+	.
14	.	mclk16-	status0	dclk16-	.
15	.	mclk17+	status1	dclk17+	.
16	.	mclk17-	status2	dclk17-	.
17	.	mclk18+	status3	dclk18+	.
18	.	mclk18-	status4	dclk18-	.
19	.	mclk19+	status5	dclk19+	.
20	.	mclk19-	status6	dclk19-	.
21	.	mclk20+	status7	dclk20+	.
22	.	mclk20-	status8	dclk20-	.
23	.	mclk21+	status9	dclk21+	.
24	.	mclk21-	reserved	dclk21-	.
25	.	.	reserved	.	.
26	.	.	GND	.	.
27	.	.	n/c	.	.
28	.	.	n/c	.	.
29	.	.	GND	.	.
30	.	.	n/c	.	.
31	.	.	n/c	.	.
32	.	.	+5V	.	.
33	.	.	.	.	.
34	.	.	.	.	.
35	.	.	.	.	.
36	.	.	.	.	.

37	.	.	.	.	.
38	.	.	.	.	.
39	.	.	.	.	.
40	.	.	.	.	.
41	.	.	.	.	.
42	.	.	.	.	.
43	.	.	.	.	.
44	.	.	.	.	.
45	.	.	.	.	.
46	.	.	.	.	.
47	.	.	.	.	.

The +5V connection supplies power from the VRB Fanout to the J3 backplane termination resistors.

The following figure shows the interconnection between the System Controller and the VRB subracks/modules.



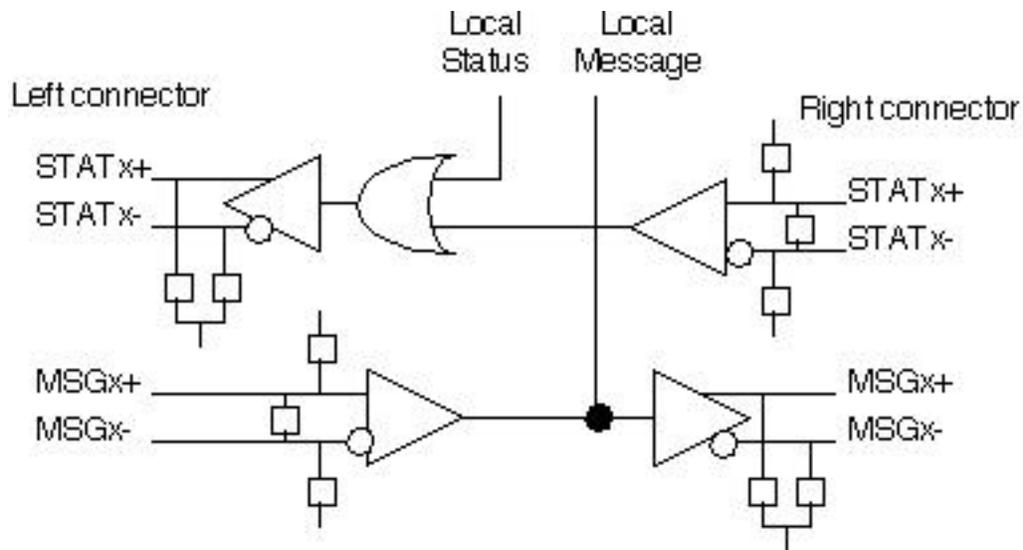
The following signals appear on the front-panel connectors as LVDS levels. The pin assignment for the front-panel connectors is listed in the following table. Both ribbon cable connectors use the same pin assignment. All signals are point-to-point and are terminated on the receiving end at each Fanout module or Controller. A ribbon cable connects the System Controller to the top connector of the first VRB Fanout module. Each additional VRB Fanout module is linked with a separate ribbon cable from the bottom connector of the previous VRB Fanout module to the top connector of the next VRB Fanout. Status signals are inverted (low true on the backplane, high true on the cable). Pins 47-50 are spare.

PIN NUMBER	SIGNAL
1	MSG0+
2	MSG0-
3	MSG1+
4	MSG1-

5	MSG2+
6	MSG2-
7	MSG3+
8	MSG3-
9	MSG4+
10	MSG4-
11	MSG5+
12	MSG5-
13	MSG6+
14	MSG6-
15	MSG7+
16	MSG7-
17	MSG8+
18	MSG8-
19	MSG9+
20	MSG9-
21	MSG10+
22	MSG10-
23	MSG11+
24	MSG11-
25	STROBE+
26	STROBE-
27	STAT0+
28	STAT0-
29	STAT1+
30	STAT1-
31	STAT2+
32	STAT2-
33	STAT3+

34	STAT3-
35	STAT4+
36	STAT4-
37	STAT5+
38	STAT5-
39	STAT6+
40	STAT6-
41	STAT7+
42	STAT7-
43	STAT8+
44	STAT8-
45	STAT9+
46	STAT9-
47	.
48	.
49	GND
50	GND

The generic circuit for each Fanout message and status signal is shown in the following figure.



The VRB Fanout Module is a 9U x 400mm module. It connects to the VME backplane for power only. The board is made of FR4 with a thickness of .092". Top and bottom card edges are milled to .063". Power consumption is estimated at 10 watts and there are no special cooling requirements.