

Fermi National Accelerator Laboratory

**SVX II Silicon Strip Detector Upgrade Project
Readout Electronics**

General System Test Module (Mother Board)

--PRELIMINARY--

Aug 4, 1997

**Mingshen Gao
Greg Deuerling**

Document # ESE-SVX-960126

DRAFT

1	GENERAL INFORMATION	
1.1	System Introduction	1
1.2	Description Of Components	2
1.3	GSTM for Testing the SVX II Upgrade Readout Electronic System	3
2	THEORY OF OPERATION AND OPERATING MODES	6
2.1	Functional Blocks	6
2.1	GSTM Mother Board to Daughter Cards Interface	11
2.2	Basic Features & Operation	16
3	INTERFACE SPECIFICATIONS	19
3.1	VMEbus Interface	19
3.2	VME Register and Fifos	21
3.2.1	Test Register	22
3.2.2	Control Register	23
3.2.3	Mode Register	26
3.2.4	Status Register	29
3.2.5	T11-FIFO	32
3.2.6	T12-FIFO	33
3.2.7	R11-FIFO	34
3.2.8	R12-FIFO	35
3.2.9	T21-FIFO	36
3.2.10	T22-FIFO	37
3.2.11	R21-FIFO	38
3.2.12	R22-FIFO	39
3.2.13	JTAG Register	40
4	ELECTRICAL & MECHANICAL SPECIFICATIONS	41
4.1	Packaging & Physical Size	41
4.2	Port connectors	41
4.3	Power & Cooling Requirements	43
4.4	Jumpers and Switches	43
4.5	Front Panel	44
5	FPGA EQUATIONS (AHDL FILES)	45

1 GENERAL INFORMATION

1.1 System Introduction

This document describes the “General System Test Module”, hereafter referred as the GSTM. The GSTM is a 9U X 400mm double-width VME module capable of transmitting and receiving multiple streams of test data. This general purpose system test module replaces several hardware test modules (especially where communication links are implemented) in a typical electronic system such as the SVX-II/III readout system. The first application for the GSTM is to test several individual modules and subsystems of the SVX II/III Upgrade Readout Electronics System without an entire functioning system.

A complete GSTM consists of a 9U X 400mm VME mother board and up to four daughter cards (or mezzanine cards). The GSTM mother board performs the common functions such as data storage, flow control and user interface required of a testing module and the daughter cards implements all the specialized functions as well as the system dependent intercommunications with the target system being tested. Replacement of the specialized daughter cards permits the GSTM to be used for testing and debugging many different electronic systems. This document will concentrate only on the GSTM mother board. Daughter cards are project specific and will be given by other documents.

The GSTM mother board has two transmitting ports and two receiving ports. Test system specific daughter cards are attached to these ports. Each transmitting port can be configured to either drive a single 35-bit data stream or drive a 26-bit data stream and receive a 9-bit data stream. Also, each receiving port can be configured to either receive a single 36-bit data stream or receive a 27-bit data stream and drive a 9-bit data stream.

Each transmitting port has two groups of 16K FIFO memories to store test data patterns. The first group is a 27 bit transmit bus that consists of three 16K x 9 FIFO's. The two upper bits of the transmit port (bit25 and bit26) are passed to the LCTL port controller to provide sophisticated logic control. The other group is a 9 bit bi-directional bus that consists of two 16K x 9 FIFO's placed back to back. The 9 bit bi-directional port can be used in conjunction with the 27 bit transmit bus to make a 36 bit transmit bus, or it can be used to receive a return data stream from the module being tested.

Like the transmit port, each receiving port has two groups of 16K FIFO memories to store received data. The first group is a 27 bit receive bus that consists of three 16K x 9 FIFO's. The two upper bits of the receive port (bit25 and bit26) are passed to the LCTL port controller to provide sophisticated logic control. The other group is a 9 bit bi-directional bus that consists of two 16K x 9 FIFO's placed back to back. The 9 bit bi-directional port can be used in conjunction with the 27 bit receive bus to make a 36 bit receive bus, or it can be used to send a data stream to the module being tested.

A 60-pin connector is provided at each of the four ports for the GSTM mother board to communicate with the daughter cards. Hand-shaking (with interlocking control signals) and flow control are done by the port controller on the mother board. The port controllers and other control blocks are all reprogrammable logic devices designed with a high level hardware

description language. They can be reprogrammed to fit nearly any kind of specialized requirement for the target system(s). A second 40-pin connector is also provided for each port to supply the various power, -2V and -5.2V for ECL and +5V for TTL or PECL.

Also, all four ports can transmit and receive data at 53 Mhz rate, and the transmitting and receiving emulation is controlled through the VME interface.

1.2 Description of Components

The GSTM mother board is a standard VIPA 9U X 400mm double-width VME card to be installed in a VME crate such as the VRB crate or the FIB crate of the SVX II/III DAQ Readout Upgrade project. It is designed for operating at 53Mhz. However, the crystal oscillator (socketed) can be replaced for slower operation. The GSTM can also operate with the clock provided by one of its daughter cards in cases where the target system has to operate at certain clock rate and the GSTM has to be synchronized to it.

The GSTM mother board consists of several groups of 16k deep fifo memories to store the data streams to be sent and received. This allows data streams up to approximately 300 microseconds (at 53 MHz) in both the transmitting and the receiving units. If deeper memories are needed, 32k depth fifo memory with the same pinout (currently available) can be used to replace the 16k depth FIFO's.

Four programmable logic devices are used to provide the VME interface and many other control functions. The In-System programmable logic devices used for GSTM mother board are the EEPROM based Altera EPM7128SQC100. All programmable logic devices are designed with the AHDL (a high level logic language) with one AHDL file for each device. The Altera devices can be easily reprogrammed with the AHDL (i.e., modify or rewrite a few logic equations) to accommodate different setup as well as new functionalities required by different applications. Also, since the 7128S supports the in-system programmability via a standard Joint Test Action Group (JTAG) interface, all four logic devices can be reprogrammed in the system with one of Altera's Bit/Byte Blaster cables. One 7128S part is used for the VME interface and the other three 7128S are used for GSTM FIFO control. The VME 7128S part has its own Bit/Byte Blaster port, the other three 7128S's all share a Bit/Byte Blaster port. This was done so that in addition of the Bit/Byte Blaster port the VME interface could be use to program the three GSTM FIFO controllers. Jumper J8 on the GSTM mother board controls which JTAG source is used. If J8 is left OFF the Bit/Byte Blaster port is selected, if J8 jumper is ON the VME controller is selected.

Normal power of +5V is supplied through the power connectors for daughter cards (one connector for each port). DATEL's UWR-5.2/1500-D5 Dc-to-DC converters are used to supply the -5.2V and -2V power needed by ECL circuits that may exist in some daughter cards through the same power connectors.

When the GSTM is used to test different electronic systems, daughter card(s), which carries all the specialized interfacing and protocol for the system, should be designed. Such the GSTM mother board can be reused again and again for different systems or subsystems with different daughter cards. Also, most daughter cards are designed to be used for general purpose and can be reused. For testing the entire SVX II Upgrade Readout Electronics System, seven

daughter cards are expected to be designed. We will devote the next section to a brief outline for the project.

1.3 GSTM for Testing SVX II Upgrade Readout Electronic System

In this section, we will briefly outline the arrangement for using the GSTM to test the SVX II Upgrade Readout Electronics System.

Figure 1 shows a Block Diagram of the control flow (all the control signals related communication links) for the SVX II Upgrade Readout Electronics System. Shown at the top of the figure is the fast control link (TAXI optic link) and the timing link (differential PECL level signals) from the top level system modules, represented by the Trigger Supervisor Interface (TSI) and Master Clock (MC), to the Silicon Readout Controller (the SRC) module. The fast control and timing links from the SRC module to the VME Readout Buffer (VRB) is implemented through daisy-chained differential RS485 twist & flat ribbon cable. and communications from the SRC to the FIB are via the high speed Glink/Finisar optic link). Also, the figure shows the returned control signals from the SRC module to the TSI (via coax cable) and the returned control signals from the VRB crate fanout module to the SRC module.

By the design of 7 system dependent daughter cards, the GSTM should be able to emulate all the communications protocols in the SVX II Upgrade Readout Electronics System. Therefore, all crates or modules in the system can be effectively tested without the presence of other system modules or before the final system is installed. We will only outline the basic functions for these daughter cards and how they can be used for the test in this section.

These seven daughter cards are 1) the optical 1.5-Gb G-link/Finisar Transmitter, 2) the optical 1.5-Gb G-link/Finisar Receiver, 3) the optical AMD TAXI transmitter with miscellaneous other communication links for testing the SRC, 4) the optical AMD TAXI receiver, 5) the parallel differential PECL (or RS-485) drivers and receivers (as well as connector for the daisy-chained twist&flat cable), 6) the reverse of the number 5 daughter card for testing the VRB crates, and 7) the DOIM communication link for the PC (Port Card). Several daughter cards may be required simultaneously for testing and debugging the SVX II Upgrade Readout Electronics System. Details on these daughter cards will be given by separated documents listed below (some documents may be combined):

GSTM daughter cards for SRC testing :

Daughter card (number 3 in the list) for TAXI transmitter, PECL, COAX.

Daughter card (number 2 in the list) for G-Link receiver.

Daughter card (number 5 in the list) for differential PECL receiver.

GSTM daughter card for VRB testing (number 6 in the list, differential PECL transmitter).

GSTM daughter card for FIB testing (number 1 in the list, G-Link transmitter).

GSTM daughter card for TAXI receiver (number 4 in the list).

GSTM daughter card for PC testing (number 7 in the list).

The number 1 daughter card in the list is the optical 1.5-Gb G-link/Finisar Transmitter and the number 2 daughter card is optical 1.5-Gb G-link/Finisar receiver. Both are general purposed G-Link transmitter and receiver daughter cards which could be used for any modules or systems that used the G-Link.

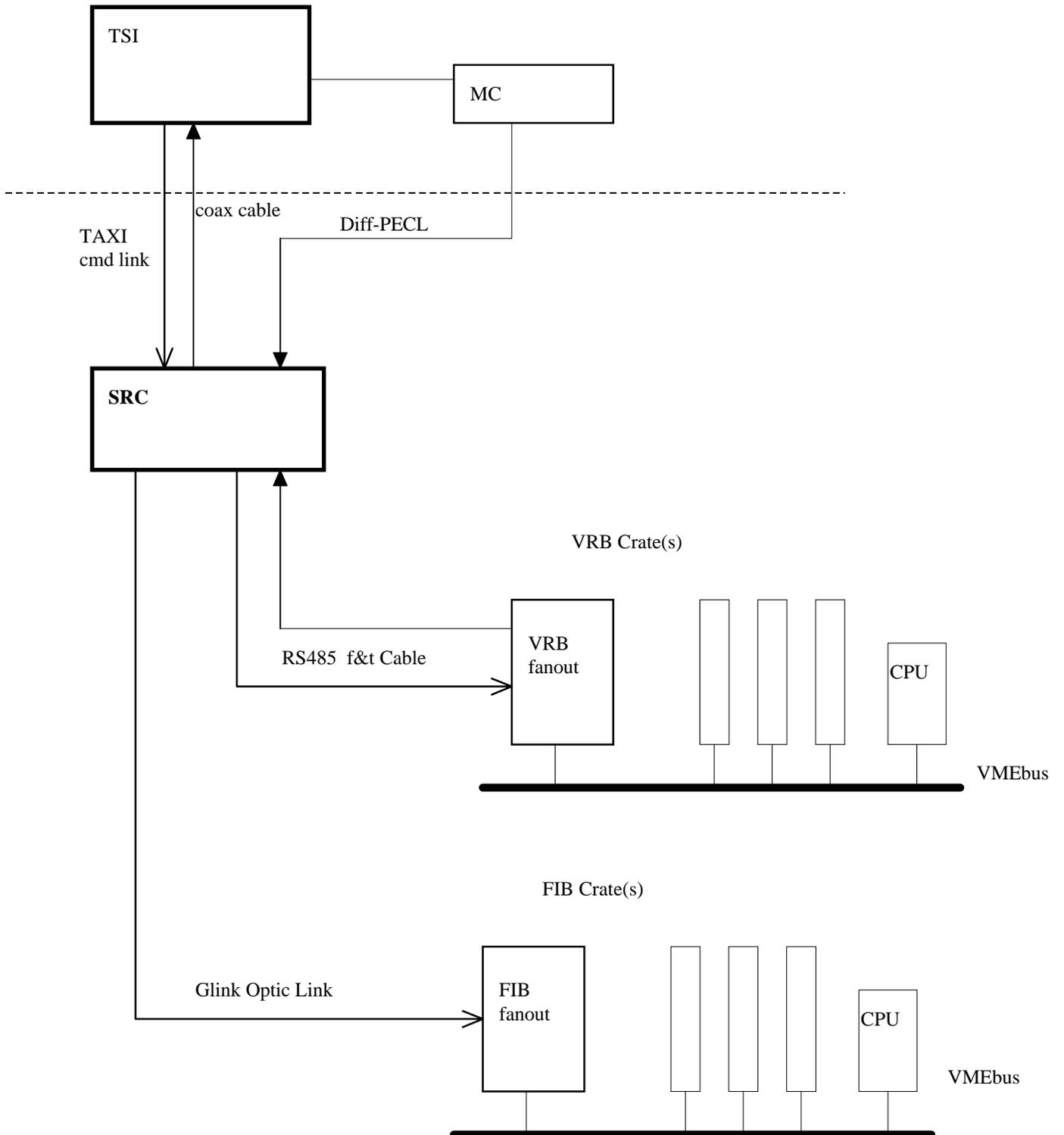


Figure 1: Control Flow (Control/Timing Links) for the SVX II / III Upgrade Readout Electronics System

2 THEORY OF OPERATION AND OPERATING MODES

The main purpose of the GSTM is to facilitate the testing and debugging of the intercommunication links of an electronic system without the presence of complete system components. It does this by mimicking the communication links between the modules of the system. This main purpose sets the basic requirement for the GSTM module. Users will be able to load designed sequences of control/data patterns into onboard memory and then start/stop/restart data transmission as well as continuous loop sequences to the modules under testing. Also, the GSTM module saves the returned response sequence (control or data signals) from the modules being tested, and allows the user to read and check the result.

To meet the basic requirement and to make the GSTM generally useful for various different kinds of systems at the same time, the GSTM is divided into a mother board and several daughter cards. While the GSTM mother board performs the common functions (such as data storage, flow control and user interface) typically performed by a test module, the GSTM daughter cards (up to four) are used for all the specialized system-dependent functions and for communicating with the target system or subsystem being tested. The GSTM mother board communicates with each daughter card through the corresponding port connector. This chapter is devoted to the organization and the operation of the GSTM mother board.

This chapter is organized into 3 sections. In section 2.1, we will introduce the functional blocks for the GSTM mother board. In section 2.2, we will explain the interface between the GSTM mother board and its daughter card(s). Finally, in section 2.3, we will be back to the basic operations and features for the GSTM. We will also discuss how the GSTM should be used and how it could be used in other applications in the last section.

2.1 Functional Blocks

The GSTM mother board has two transmitting ports (named as the T1-Port and the T2-Port) and two receiving ports (named as the R1-Port and the R2-Port) for up to four daughter cards. **Figure 2** shows all the major functional blocks and especially the data flow for the GSTM mother board.

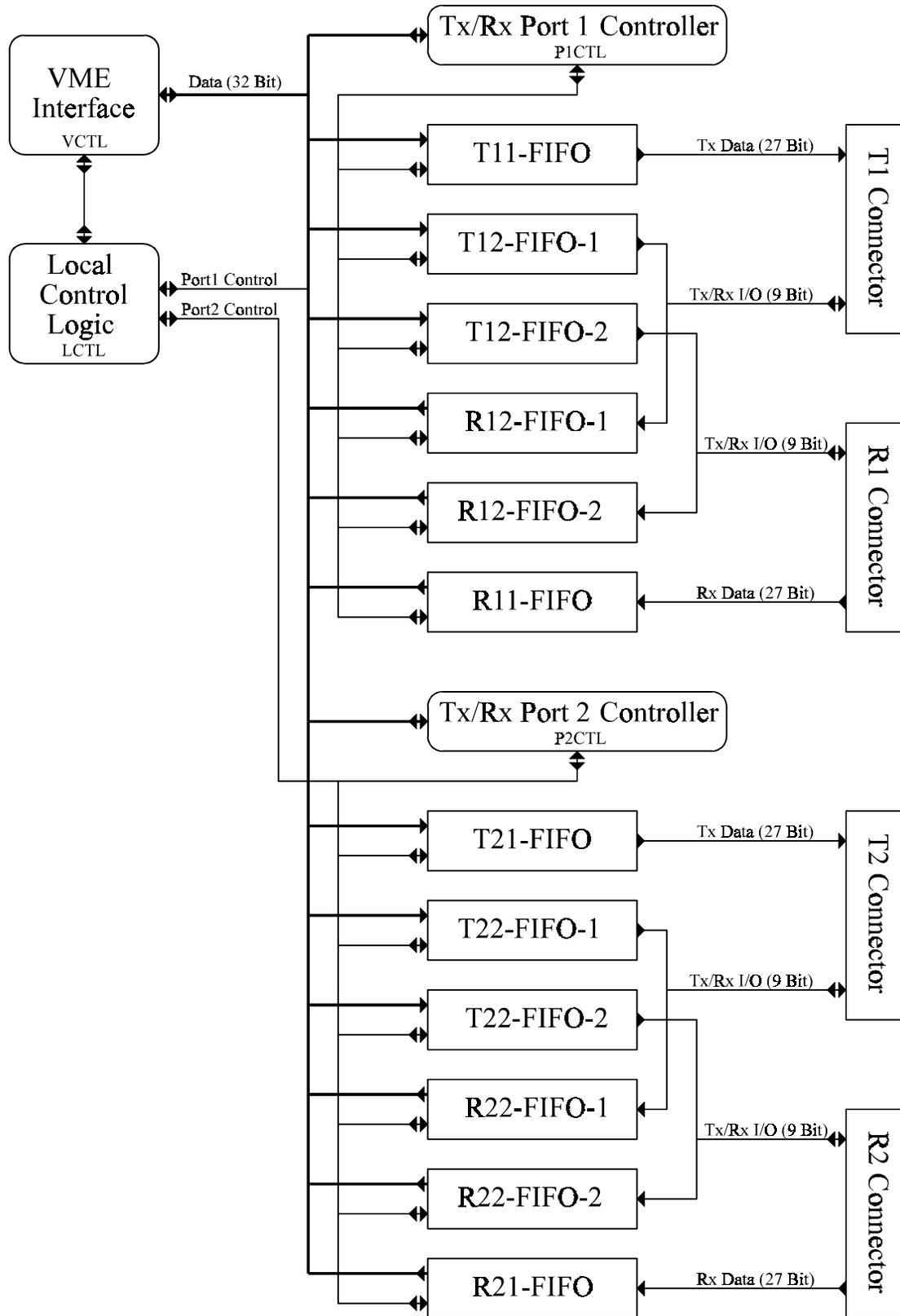


Figure 2: Functional Blocks for the GSTM Mother Board

VME Interface Block and Local Control Logic Block

The VME Interface Block consists of a Control-Register, a Mode-Register and a Status-Register. It is controlled by an Altera EPM7128SQC100 (we will call it the **Vctl** as VME-controller hereafter). It allows the access of all the FIFOs. Next Chapter will give more detailed information on the VME interface for the GSTM.

The Local Control Logic Block decodes and encodes the VME registers and generates the corresponding control signals to the GSTM mother board. The heart of this block is implemented by an Altera EPM7128SQC100 (we will call it the **Lctl** as Local-controller hereafter). The **Vctl** and the two port controllers (discussed below) also decode part of the VME registers and implement some of the functions for this block. Together, these programmable logic devices provide 1) the encoding of the user defined status bits for the SR (Status-Register), 2) the decoding of the control bits for the CR (Control-Register) and mode bits for the MR (Mode-Register) including those user defined bits, 3) and control the operation of the entire GSTM Mother board and its daughter cards. By reprogramming this device, the GSTM can be used more effectively for different testing purposes. Section 3.2 has more information on the bits assignment for the VME registers.

Transmitter T1-Port and Receiver R1-Port

The remainder of the GSTM mother board is divided into 4 functions blocks corresponding to the 4 ports which we will call them the T1-Port and the T2-Port transmitting ports, and the R1-Port and the R2-Port receiver ports.

Note, all FIFO's that begin with a T are transmitting FIFO's while all the FIFO's that begin with a R are receiving FIFO's. The VME interface side of all FIFO's is controlled by the **Vctl**. All FIFO's that relate to the T1-Port or the R1-Port begin with a T1 or a R1 with the daughter card side controlled by the **P1ctl**. Similarly, for the FIFO's that relate to the T2-Port and the R2-Port begin with a T2 or a R2 with the daughter card side controlled by the **P2ctl**. Also, all buffers and switches in the GSTM mother board are set by the **Lctl**.

The T1-Port Block consists of several groups of fifo-memories, a port connector (T1-Connector) and data buffers. The T11-FIFO, the R12-FIFO-1 and the T12-FIFO (which is the combination of the T12-FIFO-1 and the T12-FIFO-2) can only be used when the T1-Port is selected (or enabled). The R1-Port Block also consists of several groups of fifo-memories, a port connector (R1-Connector) and data buffers. The R12-FIFO-2 is a parallel extension to the R11-FIFO due to the 32 bit data size limit on the VME bus interface and the requirement of the port connector.

The T1-Port implements a 60-pin connector for the daughter card with 27 pins for driving data from their transmitting FIFOs on the GSTM mother board. The transmitting port also has 9 I/O pins for either driving or receiving data depending on the configuration of this port. The configuration for a port is set by the corresponding bits in the Mode-Register, as discussed in Section 3.2.3. More information on the transmitting port connector will be given in the next section.

The R1-Port implements a 60-pin connector for the daughter card with 27 pins for the received data routed to input. The receiving port also has 9 I/O pins for either driving or receiving data depending on the configuration of these ports. The configuration for a port is set by the corresponding bits in the Mode-Register, as discussed in Section 3.2.3. More information on the receiving port connector will be given in the next section.

The T1-Port and the R1-Port are controlled by the P1-Controller, which is an Altera EPM7128SQC100 (we will call it **P1ctl** hereafter). The **P1ctl** controls the data flow between the GSTM mother board and daughter cards on these two ports including the operation of the related FIFO memories and data buffers, etc. It also takes care of specialized handshaking types of control between the mother board and the daughter cards by driving 3 control signals and receiving 3 status signals for each port. Since this is a programmable logic device, the 6 control/status signals for each port can be reprogrammed to provide any special interfacing requirement of the daughter card on each port (the two ports are independent). The documents on daughter cards for the SVX II Upgrade Readout Electronic System give good examples for this. Of course, if special controls are not required (See the next subsection), reprogramming the controller for the 6 control signals is unnecessary.

For the Transmitter T1-Port Block, the T11-FIFO (a 16kX27 bits fifo) is used to store data pattern to be transmitted to the daughter card on the T1-Port. The T12-FIFO-1 (a 16kX9 bits fifo) and the T12-FIFO-2 (a 16kX9 bits fifo) are combined to form the T12-FIFO, which is the actual fifo that the user can write to through the VME bus interface. The T12-FIFO-1 drives the T1-Port if the 9 I/O pins for the T1-Port are selected by the Mode-Register. The T12-FIFO-2 can be used to drive the R1-Port when the 9 I/O signals of the R1-Port are configured in the Mode-Register as transmitting pins. The Mode-Register contains two mode bits to configure the T1-Port, the R1-Port and to enable the buffers for the two 9 bit data streams. The R12-FIFO-1 is used to store the response data stream (sequence of status or others) from the of transmitted data stream of T1-Port daughter card. The 9 I/O pins have to be configured as receiving signals in this case.

Note, the T11-FIFO and the T12-FIFO (the combination of T12-FIFO-1 and the T12-FIFO-2) will operate under the same set of control signals during a transmitting process on the fifo's output side. But from the VME interface the T11-FIFO and the T12-FIFO are separated FIFO's. And, they have to be loaded separately.

For the R1-Port Block, if the 9 I/O pins are configured as receiving pins, then the R11-FIFO and the R12-FIFO-2 will work together to store the 36 bits received data stream. if the 9 I/O pins are configured as transmitting pins, then the R11-FIFO itself will be used to store the 27 bits received data stream and the T12-FIFO-2 will be used to transmit the 9 bits data stream.

The R11-FIFO and R12-FIFO-2 will operate under the same set of control signals during a receiving process on the fifo's input side. The R12-FIFO-1 has separated control signals related to the T1-Port. However, the receiving fifo (the combination of the R12-FIFO-1 and the R12-FIFO-2) are actually two independent FIFO's read separately from VME.

Transmitter T2-Port and Receiver R2-Port

The T2-Port Block and the R2-Port Block are identical to the T1-Port Block and the R1-Port Block. All ports are independent except that any transmitting port may be used to control the operation of the receiving ports.

The T2-Port and the R2-Port communicate with their respective daughter cards through their own interface connectors just like the T1-Port and R1-Port.

The T2-Port and the R2-Port are controlled by the P2-controller, an Altera EPM7128SQC100 named as **P2ctl**.

For the T2-Port Block and the R2-Port Block, the T21-FIFO and the T22-FIFO, the R21-FIFO and the R22-FIFO, and others are operating in exactly the same way as the corresponding FIFO's in the T1-Port/R1-Port blocks, except that they are working for the T2-Port and the R2-Port. Also, they are completely independent from those FIFO's in the T1-Port/R1-Port blocks.

2.2 GSTM Mother Board to Daughter Card Interface

While the GSTM mother board performs the common functions, the GSTM daughter cards (up to four) perform all the specialized system-dependent functions to communicate with the target system or subsystem being tested. The GSTM mother board communicates with each daughter card through its corresponding port connector. All four port connectors have 60 pins. The block diagram (**Figure 3**) defines the interface between the GSTM mother board and daughter cards, one block describes the daughter cards on a transmitting port and one block describes the daughter cards on a receiving port. The numbers shown next to the signals are the number of bits for each signal.

Figure 4 shows the pin assignment for both types of port connectors. Only pins that tie to the GND and pins that carry signals exist in the 60-pin interface connector for each daughter card. A second smaller connector is used to provide the power for each daughter card. This second power connector will also provide special supply voltages such as the -5.2V, -2V. All these are generated on the GSTM mother board from its regular +5V power supply. The section on the power connector in chapter 4 will give more information on the power connector.

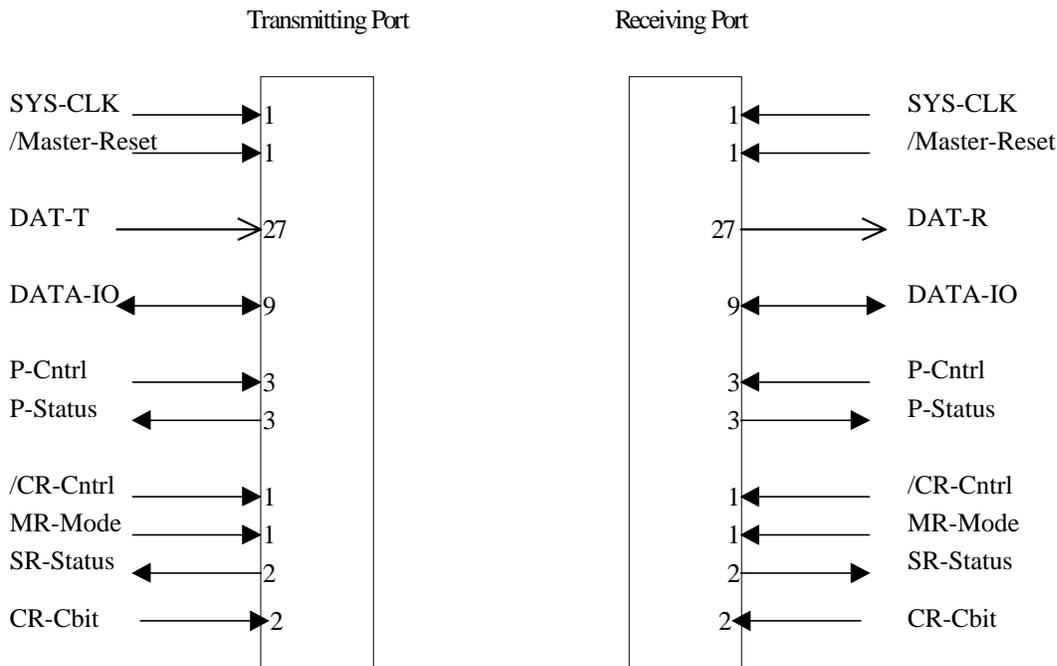


Figure 3. GSTM Mother Board and Daughter Cards Interface

Top View

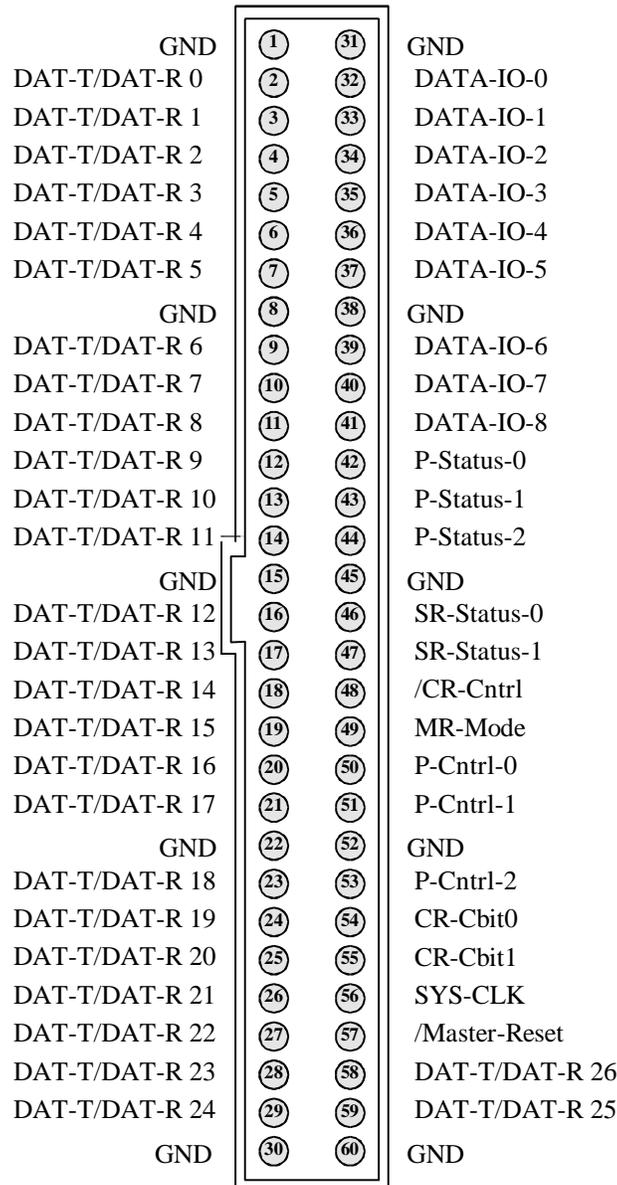


Figure 4. GSTM Port Connectors Pin Assignment

The only difference between a transmitting port and a receiving port is that the transmitting port has 27 bits of data that goes into the port connector while the receiving port has 27 bits of data that comes out of the port. These signals are shown in **Figure 3** and **Figure 4**, as DAT-T (going into the port connector) is used for the transmitting port connector, and DAT-R (coming out of the port connector) is used for the receiving port connector. We will only discuss the other common (to both types) signals in the following paragraphs.

The SYS-CLK signal is the system clock used for the entire GSTM mother board. Almost all the signals on the GSTM mother board are synchronized to this clock. The GSTM daughter cards should use this SYS-CLK as their clock signal if possible. Note, the data received by the daughter card may need to be synchronized to this clock signal to avoid data setup or hold types of timing problems for latches and FIFO's. The GSTM mother board is also provided with a jumper to change the phase (i.e., to invert the phase) of the SYS-CLK signal in order to help to solve this potential problem. Also, the GSTM mother board can use the clock signals generated on one of the daughter cards as its SYS-CLK in some special cases where the GSTM has to be synchronized in frequency with the clock of the target system. An example of this type of application would be the test of a communication link using two GSTM modules where the receiving GSTM module should operate at the same clock rate as the transmitting GSTM module. The daughter card will pass the clock information to the GSTM through the P-Status[2] signal in these cases and the **Lctl** can be programmed to use the correct clock signal.

The /Master-Reset signal (active low) is the logical 'OR' of the power-up reset, push-button reset, and the software generated reset (through the Control-Register). The daughter card should reset all its electronics when the /Master-Reset signal is asserted, thus resetting the entire GSTM (mother board and daughter cards).

The DATA-IO signals are the 9 I/O pins for a port connector. They can be set either as input pins (so an input to the daughter card can transmit data received through them to other modules) or as output pins (so an output from the daughter card can drive them with data from another module), regardless of whether the port is a transmitting port or a receiving port. The direction is controlled by a mode bit in the Mode-Register for each port. The mode bit sets these 9 pins as input (for transmitting) or output (for receiving), as well as configures the data latches (switches) to connect the data to/from the right fifo memories. For example, setting the T1-Port's DATA-IOs as receiving bits will automatically connect the DATA-IOs to the receiving fifo R12 -FIFO-1, and the data buffer for the T12-FIFO-1 will be tri-stated (see block diagram in **Figure 2**).

The GSTM user performs test procedures by controlling the data flow to and from the daughter card(s). The GSTM mother board and daughter card interface provides two groups of control signals for the user to control the daughter card. The port controller uses the 3 P-cntrl signals and the 3 P-status signals to communicate with the daughter card, these are the first group of control signals. The user can also communicate with the daughter card directly by using the CR-Cntrl, MR-Mode and SR_Status(s) through the VME registers, which are the second group of control signals.

The 3 P-cntrl signals and the 3 P-status signals are primarily hand-shaking signals between the port controller on the GSTM mother board and the daughter card. Since these six signals are only between the port controller and the port connector, and the port controller is a reprogrammable logic device, these six signals have the flexibility to implement any type of interface required by the daughter card.

The port controller can also encode the status signals received from the daughter card and pass it to the **Lctl**, which will report the information to the user through the Status Register. Details on these VME registers are given in the next chapter.

Examples of the use of the control and status signals may be found in the documents describing the SVXII upgrade daughter card references.

The /CR-Cntrl signal and MR-Mode signal reflect bits in the controlregister and mode register, and as such may be used to pass these user settable signals to the daughter cards. The two SR-Status signals are used to get status information from the daughter card displayed in the Status-Register (The status information will pass an encoding process in the **Lctl** first, see the section 3.2.4). In general, the SR-Status[0] signal should be used for passing the ready or good status information and the SR-Status[1] signal should be used for passing the bad or error information. However since the GSTM can be reprogrammed for a specific test, the two may be used differently. Whenever possible, these signals should be used for the control of the daughter card(s). More information on these signals are given in Section 3.2 when we discuss the VME registers for the GSTM mother board.

The two CR-Cbit signals pass 2 bits of control information to the daughter card directly through the Control Register. These two control signals are passed to all ports simultaneously while there is one /CR-Cntrl signal for each port. The daughter card can use the two signals to generate WAIT, ERROR, or other exceptional condition. Or they can be used to generate any other control sequence for the daughter card. Not all the daughter cards will need them. A daughter card can ignore them if they are not needed and a daughter card can also combine them with other control signals when they are needed.

For a receiving port, the upper 2 bits of the received data (DAT-R-26, DAT-R-25) and the upper 2 bits of the I/O data (DATA-IO-8, DATA-IO-7) are routed to its port controller. These bits can be used for sophisticated control of the daughter card. Since the port controller is a reprogrammable logic device, their usage can be redefined by the user for special functions of the daughter card. For example, for a daughter card with a TAXI receiver chip, the DAT-R-25 and the DAT-R-26 can be used to pass the CSTRB (command received) and the DSTRB (data received). Also, for a daughter card with a G-Link receiver, the DAT-R-25 and the DAT-R-26 can be used to pass the /CAV (command received) and the /DAV (data received). The /DAV and the /CAV signals combined with the /READY signal indicate that valid data or a valid command has been received. The port controller can store only the valid data or commands to its receiving fifo rather than the way (which is currently used) to save one word on every cycles once the receiving process is started. However, like a logic analyzer, saving one word on every cycle will also save the important timing information associated with the receiving process for the receiving port.

Similarly, for a transmitting port, the upper 2 bits of the transmitted data (DAT-T-26, DAT-T-25) and the upper 2 bits of the I/O data (DATA-IO-8, DATA-IO-7) are routed to its port controller. As discussed in the previous paragraph, these bits can be used for more sophisticated control of the daughter card. However, for the transmitting ports, the highest transmitting data bit, the DAT-T-26, is used to mark the end of a data pattern sequence. When this bit is high, the port controller will stop the transmitting process for the data sequence.

In summary, the transmitting port connectors and the receiving port connectors define the interface between the GSTM mother board and its daughter cards. Each receiving port allows the GSTM mother board to receive up to 27 (or 36 when including I/O bits) bits of data from its daughter card at 53 Mhz and each transmitting port allows the GSTM mother board to transmit up to 26 (or 35 when including I/O bits) bits of data to its daughter card at 53 Mhz. In addition, each port has 9 flexible I/O bits for either transmitting or receiving. The user can communicate with each port and its daughter card through the GSTM VME registers. Control information is sent through the Control Register, mode or configuration information is sent through the Mode Register and the status or error information is received from the Status Register. The user can also send 2 bits of control information to all ports simultaneously through the Control Register.

Also, each port connector has 6 signals to allow the daughter card to exchange control or other types of information with the port controller on the GSTM mother board. These may be used for handling special interface requirements of a daughter card.

Additionally, each daughter card can use the upper 2 transmitting bits of a transmitting port or the upper 2 receiving bits of a receiving port and the upper 2 I/O bits to pass information to its port controller for more sophisticated (e.g., data dependent) control of the daughter card.

2.3 Basic Features & Operation

The GSTM is designed to be a multipurpose test module capable of driving and receiving several data streams to and from a target system or systems simultaneously. It is a 9uX400 VME card designed to operate at clock speeds up to 53Mhz. All data driving and receiving functions may be controlled through the VME interface.

To generalize the GSTM for use in testing in various different projects, it has been functionally divided into a mother board and up to four daughter card(s). The GSTM mother board performs the common functions of a test module such as the user interface, data storage, and controlling data flow. The GSTM daughter card(s) takes care of the specialized functions and system dependent interfaces for the target system(s). The GSTM mother board also contains reprogrammable logic devices so that control of the GSTM daughter card(s) can be changed to fit the special needs of the test project.

Design of daughter card(s)

To use the GSTM for testing a system such as the SVX II Upgrade Readout Electronic System, the user should refer to the documents on the existing individual GSTM daughter card(s) such as those listed in the appendix. They provide examples on how the GSTM works and especially how the GSTM mother board interfaces with its daughter cards. The user may also determine whether the current GSTM daughter cards are appropriate for his/her target system. Many GSTM daughter cards have been designed to accommodate various. For example, the AMD TAXI receiver and/or transmitter daughter cards and the G-Link/Finisar optical receiver and/or transmitter daughter cards may be used as independent testing tools for other (future) modules that require this type of communication link.

New GSTM daughter cards may be designed to accommodate different communication links or to perform different functions. If the new card(s) need a specialized interface or user-defined bits in the GSTM VME registers, the port controller(s) and the **Lctl** may be reprogrammed. This should not be difficult since the programmable logic devices are designed with the AHDL (a high-level hardware description language by Altera), with one AHDL file per device.

3 VME INTERFACE AND VME REGISTERS

3.1 VMEbus Interface

The GSTM mother board is a 9U X 400mm double width VME card. It is a VMEbus slave and typically controlled by a VMEbus CPU module running VxWorks. All functionalities of the GSTM are controlled through the VMEbus interface. The VMEbus interface has the following characteristics:

- The GSTM board is always a VMEbus slave with no VME master capability.
- The GSTM board implements only the A32:D32 addressing scheme.
- The VME interface executes only single data transfers with no block transfer capability.
- The VME interface allows only 32 bit data transfers.
- The VME interface does not generate interrupt.
- The VME interface does not generate VMEbus errors.

The GSTM is addressed in a 32-bit address mode using the VME Address Modifiers either 0x09 or 0x0d. The Address Modifier for the GSTM is set by a DIP switch (marked as SW4 on the GSTM mother board). This DIP switch should be set to 0x09 or 0x0d normally. More information on the DIP switches are given in the section on the GSTM jumpers/switches in chapter 4.

The GSTM implements 16 bits of base address in the upper 16 bits of the 32 bit address. When the upper 16 bits of the VME address (A[31:16]) match the setting of the GSTM base address and the address modifier matches the setting of the SW4 DIP switch (Address Modifier for the GSTM), a select LED on the front panel will indicate that the GSTM is addressed by the VME interface and the address lines A[7:4] will select one of the internal registers or FIFO's. The requested action is then performed and is asserted until the DS* or AS* is deasserted.

The base address for the GSTM is divided into two 8-bit segments. The second 8 bit segment is determined by the setting of a DIP switch (marked as SW2 on the GSTM mother board), which sets the bits A[23:16] for the base address. The first 8 bit segment is determined one of two ways depending on what type of VME crate is used to host the GSTM module. When the 160-pin J1 (the VIPA crate) connector is used, the 5 Geographical Addresses GA4,...GA0 are defined by the VME crate and they will set the bits A[31:27] for the base address (Note, the inverted values on the GA's are used to set the base address as required by the new VME standard). The remaining 3 bits, bits A[26:24], are set to zero in this case. Otherwise, when no GA-lines are available (which is the case when an old VME crate or a VME extension card is used), the top 8 bits of the base address, bits A[31:24], are set by a DIP switch (marked as SW3 on the GSTM mother board) similar to the case for the lower 8 bit segment. More information on jumpers and DIP switches is given in the section on the GSTM jumpers/switches in chapter 4.

Note that the GSTM module will determine how to set its base address by monitoring the 5 GA lines coming from the VME crate. If the VME crate does not drive the GA lines (i.e., all GA lines are high due to pull-up resistors), the setting of the SW3 DIP switch is effective. Otherwise, GA lines are selected. It is required of the software to use the right address in order to access the GSTM module in both cases.

3.2 VME Registers and Fifos

Table 3 shows the VME registers and FIFO's and associated information. The following subsections will give more detailed descriptions for each of these. Note that all bits in the VMEbus registers and FIFO's are active high (i.e., effective when in the high state) unless otherwise specified. Therefore, setting a bit means writing a one to it and clearing a bit means writing a zero to it. All bits marked as Not-Valid in the VMEbus registers and FIFO's mean that the bits are not supported by the current GSTM hardware. All bits marked as Not-Used mean that they are passed to the programmable logic devices but not specified yet for any particular use. If needed, they can be reprogrammed for further expansion as user defined bits.

In all tables about VME registers and FIFO's, the first column defines the bit position, the second column gives a brief description of the bit's use, and the third column provides special notes or shows, as an example, how it is used in testing the SVX II Readout System. Since reprogrammable logic devices are used to decode all of the bits, their usage can be easily changed if necessary when the GSTM is used for different projects. However, the current definition and usage of these bits should be fine for most applications.

REGISTER	OPERATION	ADDRESS
Test Register	read only	Base Address + 0x00
Control Register	write/(read)	Base Address + 0x10
Mode Register	write/(read)	Base Address + 0x20
Status Register	read only	Base Address + 0x30
T11-FIFO	write only	Base Address + 0x40
T12-FIFO	write only	Base Address + 0x50
T21-FIFO	write only	Base Address + 0x60
T22-FIFO	write only	Base Address + 0x70
R11-FIFO	read only	Base Address + 0x80
R12-FIFO	read only	Base Address + 0x90
R21-FIFO	read only	Base Address + 0xa0
R22-FIFO	read only	Base Address + 0xb0
Reserved	write only	Base Address + 0xe0
JTAG Register	write only	Base Address + 0xf0

Table 3. VME Registers in GSTM

3.2.1 Test Register

Table 4 provides the Test Register bit assignment. The Test Register is a read only register which returns an identification number. The identification number is divided into 5 sections. The lowest 8 bits (bits 0 to 7) is hard-wired to be 0x09, which is the ID number assigned to the GSTM and programmed into the configuration ROM.

BIT	MEANING	EXAMPLE/NOTES
26-31	ID for the DC in T1-Port	DC = Daughter Card
20-25	ID for the DC in R1-Port	
14-19	ID for the DC in T2-Port	
8-13	ID for the DC in R2-Port	
0-7	fixed to be 0x09	GSTM is assigned ID 0x09

Table 4. Test Register

The remaining 24 bits are divided into 4 sections of 6 bits each. Each daughter card should return a 6 bit ID number (they are passed from the power connector as will be discussed in chapter 4) such that the different daughter cards for a project can be easily distinguished. The four sections contain the returned identification numbers from the four daughter cards, one for each port. Therefore, the combination of all the ID numbers for the daughter cards will give the user information on what has been installed on a GSTM module. Since different tests need different combinations of daughter cards, this is very useful.

3.2.2 Control Register

Table 5 provides the Control Register bit assignment. The Control Register controls the operation of the entire GSTM module. Bits 0 to 3 are routed to the **Vctl**, bits 4-13 are routed to both port controllers and all others are routed to the **Lctl**. All the unused user-defined bits can be used in a specific test project. All the reprogrammable logic devices (port controllers and the **Vctl/Lctl**) can be changed (including redefining the used control bits) to fit the special needs of a test project if necessary.

BIT	MEANING	EXAMPLE/NOTES
31	/CR-Cntrl for T1-Port	Cntrl-Signal to T1 Port
30	/CR-Cntrl for R1-Port	Cntrl-Signal to R1 Port
29	/CR-Cntrl for T2-Port	Cntrl-Signal to T2 Port
28	/CR-Cntrl for R2-Port	Cntrl-Signal to R2 Port
27	User-Defined Cbits, Not Used	
15-26	Not Valid	
14	Read-only Bit, Not Used	
11-13	User-Defined Cbits, Not Used	Passed to Port-Controllers
10	Start-Receiving	
9	Stop-Receiving	
8	Reset/Clear Selected FIFOs	Selected in mode register
7	LoopOver-Transmitting	
6	Restart-Transmitting	Restart at beginning of FIFO
5	Start-Transmitting	
4	Stop-Transmitting	
3	Clear Error Status Bits	
1-2	Global CR-Cbits	Cntrl-Signals to all Ports
0	Master-Reset	

Table 5. Control Register

Bit 0 is used for the software generated Master-Reset signal. Set, by writing a one to it, the Master-Reset control bit will reset the whole module, leaving all FIFOs empty with all the transmitting and receiving units disabled. This bit is sticky and you have to clear (by write a zero to it) it to deassert this software generated reset.

Bits 1 and 2 (Global CR-Cbits) are used to pass two bits of (global) control information to all four ports. They can also be used to control the daughter card(s) or the daughter card(s) can use them to generate some control sequences. An example would be to use them to generate an ERROR or a WAIT signal on the daughter cards during a transmitting or receiving process with a

test module or modules. The daughter cards that make use of these 2 bits should decode them and decide what to do with them

Bit 3 (Clear Error Status Bits) is used to clear the error message in the Status Register (encoded in the **Lctl**). All error bits should be cleared when this bit is set.

Bits 4-7, the 4 transmitting control bits, (Stop-Transmitting, Start-Transmitting, Restart-Transmitting, LoopOver-Transmitting) are only effective when one or both of the transmitting ports are selected in the mode register.

Setting the Start-Transmitting bit will start the transmission of data from the selected transmitting FIFOs out the port connectors. Also, all selected receiving ports will start to write incoming data (from the port connectors) to the receiving FIFOs. This transmitting process continues until the port controller detects the STOP-BIT (bit 26) in one of the data words loaded into a transmitting FIFO. It transmits two words beyond this stop bit. If there are more sequences inside the FIFO's, this bit has to be set again to start the next sequence.

Setting the Stop-Transmitting bit will stop any transmitting process in progress. Otherwise, it has no effect.

Setting the Restart-Transmitting bit will generate a FIFO-retransmit signal to the transmitting FIFOs to reset the pointer to the beginning of the FIFOs, then restarting the transmission process. Therefore, the effect is a combination of FIFO read-pointer reset and Start-Transmitting. If any receiving unit is enabled, the receiver FIFOs will also get a FIFO-reset signal to reset the pointer to its beginning.

Setting the LoopOver-Transmitting bit has the same effect as setting the Restart-Transmitting bit with the exception that while Restart-Transmitting will retransmit the loaded sequence only once, the LoopOver-Transmitting will retransmit the loaded sequence again and again. At the end of the first loaded sequence, as indicated by the Stop Bit, the LoopOver-Transmitting bit will cause the fifo pointer to reset and then start to transmit the loaded sequence all over again. The process has to be stopped by either setting the Stop-Transmitting bit or clearing the LoopOver-Transmitting bit.

Bit 8 (Reset/Clear Selected FIFOs) is used to reset all the transmitting or receiving FIFOs for the selected ports. The sequences loaded to the FIFO's will be lost. To reset or clear the FIFO's associated with one specific port, the other ports can be temporarily deselected by clearing the port enable bits in the Mode Registers, setting and clearing this bit, then restoring the original Mode Register settings.

Bits 9-10, the 2 receiving control (Start-Receiving, Stop-Receiving) bits, are only effective when no transmitting ports are enabled. The current GSTM firmware dictates that when either transmitting port is selected, it will control the operation of the receiver ports. (In other words, as long as one of the transmitter ports is transmitting data, the receiver ports will clock data into the receive FIFOs. This design feature was to allow the user to receive a defined number of data words by loading a random sequence with the desired length into a transmitting FIFO and running the corresponding transmitting port.) The Start-Receiving and Stop-Receiving bits are used to control the operations of all the enabled receiving ports when the transmitting ports are deselected.

Setting the Start-Receiving bit will put the enabled receiver unit into a ready to receive state. The first valid incoming data will start the recording of the incoming data stream on every clock cycle. This process continues until the receiving FIFOs are full or the Stop-Receiving bit is set. If programmed, the process can also be stopped when a “receive error” is encountered if the Stop-On-Error mode bit is set (see the next section).

Setting the Stop-Receiving bit will stop the FIFO receiving process in progress, but leave the FIFO pointers where they are. The Start-Receiving bit can be set again to start the receiving process from this point.

Bits 11 -13 (3 bits) in the Control Register are user-defined control bits which are not used for anything yet. They are passed directly to both port controllers which are reprogrammable logic devices. Therefore, they can be function as user defined control bits for future expansion or project specific control. Also, since the port controllers have the 3 P-cntrl and 3 P-status signals for each port connector, these bits can easily be used for passing more control information to daughter cards.

Bit 14 is a read-only bit. It is used to pass a bit of information from the Lctl to the user through the Control Register. It is not used for anything yet, but can be reprogrammed for future usage.

Bits 15 - 26 are not valid bits. They have no effect since there is no hardware support for them.

Bit 27 in the Control Register is an additional user-defined control bit which is not used for anything yet. It is passed directly to the **Lctl**. It can be used for anything by reprogramming the **Lctl**. The **Lctl** also has a few signals that are connected to all the port controllers and the **Vctl** so that it can also be used by the port controllers and the **Vctl**.

Bits 28-31, (one for each port), are used to pass one control signal to each daughter card. The bits are inverted first to pass the inverted control signals to the daughter cards. This should be very useful for passing specialized signals such as a reset, a resynch or others which are normally active low. Unlike the transmitting or receiving control bits, these bits are sticky. Therefore the user can control the length of time they are set. Also, since the **Lctl** decodes and passes them to the port connectors, they can be changed to active high very easily by reprogramming the **Lctl**.

Even though the control register can be read, the result can only reflect what was written into the Control Register during the last write. Many control bits in this register are not sticky and they cause the specified action to be executed only once. However, observing the last operation can be useful for tracking down what has been done with the GSTM module. The action specified by the control bits have been executed in most cases by the time the register is read back.

3.2.3 Mode Register

Table 6 provides the Mode Register bit assignment.

BIT	MEANING	EXAMPLE/NOTES
31	Set T1 DATA-IO as Receiving	
30	Set R1 DATA-IO as Receiving	
29	Set T2 DATA-IO as Receiving	
28	Set R2 DATA-IO as Receiving	
27	User-Defined Mbit, Not-Used	Passed to the Lctl
26	Read-only Bit, Not Used	from theLctl
14-25	Not-Valid	
10-13	User-Defined Mbits, Not-Used	Passed to all Port-Controllers
9	User-Defined Mbit, for G-Link	G-Link flag select
8	Stop-on-Error	
7	T1-Mode	TAXI 8/9 transfer mode
6	R1-Mode	TAXI 8/9 transfer mode
5	T2-Mode	G-Link 16/20 transfer mode
4	R2-Mode	G-Link 16/20 transfer mode
3	Select/Enable T1-Port	Also the FIFO's in the block
2	Select/Enable R1-Port	
1	Select/Enable T2-Port	Also the FIFO's in the block
0	Select/Enable R2-Port	

Table 6. Mode Register

The Mode Register sets the operating mode of the entire GSTM (mother board and daughter cards) and also sets the configuration for all ports. Bits 0 to 3 are passed to the **Vctl**, bits 4-13 are sent to both port controllers and bits 27-31 are passed to **Lctl** (all bits are routed as in the Control Register). All the unused user-defined bits can be used in a specific test project. All the reprogrammable logic devices (port controllers and the **Vctl/Lctl**) can be changed (including redefine those used mode bits) to fit the special needs of the test project if it is necessary.

The lowest 4 bits are used to enable the selected ports. All four ports are independent. For those ports to be used for a test, the user should set the corresponding bits before loading test data sequences. The user can change the setting later if the user want to used more or less number of data streams. Also, all the active bits (bits 4-10) in the Control Register are only effective to

the selected ports. Additionally, a transmitting fifo can only be used if the port is selected. This should not be a problem if the port is not used for a daughter card.

Bits 4 to 7 are passed to the four port connectors by the port controllers with one bit for each port. These are the MR-Mode signals for each port. Right now, they are passed as it is by assuming that they are active high. If negative logic is needed, the corresponding port controller has to be changed!

For the SVX II Upgrade Readout Electronic System, the T1-Port and R1-Port are mainly used for the TAXI transmitter and receiver (the transmitter card has some other electronics too). The T1-Mode and R1-Mode (passed to the daughter cards from the **P1ctl**) are used to set the TAXI transmitter and receiver chips into either a 8 bit or a 9 bit data transfer mode respectively.

For the SVX II Upgrade Readout Electronic System, the T2-Port and R2-Port are used for the G-Link transmitter and receiver daughter cards. The T2-Mode and R2-Mode (passed to the daughter cards from the **P2ctl**) are used to set the G-Link transmitter and receiver chips into either a 16 bit or a 20 bit data transfer mode respectively.

Bit 8 is used as a stop-on-error signal for all ports, i.e., while this bit is high, each port controller will stop its transmitting or receiving process when an error is encountered. Otherwise, the error is reported to the **Lctl** and the process continues.

Bits 9 to 13 (5 bits) in the Mode Register are additional user-defined mode bits. They are not used for anything yet except that the **P2ctl** (for the G-Link receiver/transmitter daughter cards in the SVX project) uses the bit 9 as shown in the next example. But they are routed directly to both port controllers which are reprogrammable logic devices. Such, they can be used as user defined mode (or configuration) bits for future expansion or project specific control. Also, since the port controllers also have the 3 P-cntrl and 3 P-status signals for each port connector, they can be easily used for passing more configuration or operating mode information to the daughter cards that need them.

For example, the two daughter cards controlled by the P2-Controller are the G-Link transmitter and receiver for the SVX II Upgrade Readout Electronic System. Bit 9 is passed to both daughter cards through the P-Cntrl-2 signal. It is the FLAG-SEL (flag bit selection) signal to the G-Link which will decide whether the flag bit for the G-Link will be used as an extra data bit for both the G-Link transmitter and receiver. If so, the actual data transfer size for both G-Link will be either 17 bit or 21 bits instead of 16 or 20 bits.

Bits 14 to 25 are not valid bits. They have no effect since there is no hardware support for them at all.

Bit 26 is a read-only bit. It is used to pass a bit of information from the Lctl to user through the Mode Register. It is not used for anything yet. But it can be reprogrammed for future usage.

Bit 27 in the Mode Register is an additional user-defined mode bit which is not used for anything yet. But it is routed directly to the **Lctl**. It can be used for anything by reprogramming the **Lctl**. The **Lctl** also has a few signals that are connected to all the port controllers and the **Vctl** so that it can also be used by the port controllers and the **Vctl**.

Bits 28-31 are used for configure the 9 bits DATA-IO signals for each port. Set any bit in this group will configure the DATA-IO pins of the corresponding port as receiving pins. Then, data buffers or switches are automatically set according to the I/O requirement. Data in the transmitting FIFO's that connected to the DATA-IOs are passed to the ports where the DATA-IO are configured as transmitting. Related receiving FIFO's will be connected to the ports where the DATA-IO are configured as receiving. The **Lctl** is used to decode the 4 mode (configuration) bits and generate the corresponding control signals for the data buffers (switches).

Let us emphasize, the Mode Register has to be set correctly before the GSTM can work properly! You may set it only once for each test. You may need more tests for each project.

All the mode bits in this register are sticky. They will stay active until the bits are cleared. The Mode Register can be read back.

Also, if a daughter card does not need mode or configuration information and the Control Register does not have unused control bits left, then many of the unused mode bits in this Mode Register can be used for passing control signals to the daughter card. This is especially true for the bits 4 to 7, which are passed to each port directly. User can easily take advantage of these bits to send control signals to daughter cards without reprogramming or any other effort.

3.2.4 Status Register

Table 7 provides the Status Register bit assignment.

BIT	MEANING	EXAMPLE/NOTES
31	General(user-defined)Status bit 15	G-Link Receiver Error
30	General(user-defined)Status bit 14	TAXI Receiver VLTN
29	General(user-defined)Status bit 13	TAXI Overwrite
28	General(user-defined)Status bit 12	
27	General(user-defined)Status bit 11	
26	General(user-defined)Status bit 10	
25	General(user-defined)Status bit 9	
24	General(user-defined)Status bit 8	
23	General(user-defined)Status bit 7	
22	General(user-defined)Status bit 6	
21	General(user-defined)Status bit 5	
20	General(user-defined)Status bit 4	
19	General(user-defined)Status bit 3	
18	General(user-defined)Status bit 2	
17	General(user-defined)Status bit 1	G-Link Transmitter Ready
16	General(user-defined)Status bit 0	G-Link Receiver Ready
15	R12-FIFO-1 full (full=0)	
14	R12-FIFO-1 empty (empty=0)	
13	R22-FIFO-1 full (full=0)	
12	R22-FIFO-1 empty (empty=0)	
11	JTAG <i>TDO</i> bit (through VME)	
10	T11-FIFO full (full=0)	also for T12-FIFO-1,2
9	T11-FIFO empty (empty=0)	also for T12-FIFO-1,2
8	R1-Port in receiving (=0)	
7	R11-FIFO full (full=0)	also for R12-FIFO-2
6	R11-FIFO empty (empty=0)	also for R12-FIFO-2
5	T1 or/and T2-Port in transmitting	
4	T21-FIFO full (full=0)	also for T22-FIFO-1,2
3	T21-FIFO empty (empty=0)	also for T22-FIFO-1,2
2	R2-Port in receiving (=0)	
1	R21-FIFO full (full=0)	also for R22-FIFO-2
0	R21-FIFO empty (empty=0)	also for R22-FIFO-2

Table 7. Status Register

The Status Register contains the status of the entire GSTM including messages passed from daughter cards. Unlike other registers where nearly all bits are active high, 14 of the lower 16 bits here are active low (the active low bits are indicated by the (=0) at the end). They are coming from the corresponding fifo chips and they are the true status when they are zero. The other 16 bits are user defined (project specific) which are encoded and passed by the **Lctl** (a FPGA, reprogrammable logic device).

Bits 0, 3, 6, 9 and bits 1, 4, 7, 10 are the empty and full flags for the two main receiving FIFO's and the two main transmitting fifo. When they are zero, they indicate that the corresponding FIFO's are empty or full. Since the R22-FIFO-2 will always work together with the R21-FIFO, bits 0 and 1 should be the true flags for the R22-FIFO-2 too. Similarly, the R12-FIFO-2 and the R11-FIFO use the same flags. Since both the T22-FIFO-1 and the T22-FIFO-2 will work together with the T21-FIFO, they will use the same flags. So do the T12-FIFO-1 and the T12-FIFO-2 with the T11-FIFO.

Bits 2 and 8 are the write enable signals for the two receiving ports. When they are zero, they indicate that the GSTM receiving port are recording data to their FIFO's, i.e., the GSTM is in an active receiving process for the corresponding FIFO's.

Bit 5 is the combination of the read enable signals for the two transmitting ports masked by their port select bits in the Mode Register. If one or both transmitting ports are selected, it will be used for reflecting the activity of the transmitting FIFO's. When it is zero, it indicates that the GSTM transmitting port(s) is (are) reading data from their FIFO's, i.e., the GSTM is in an active transmitting process for the corresponding FIFO's.

Bit 11 will be used as the TDO signal returned from the JTAG chain, when the **Lctl** and the two port controllers are programmed through the VME-Interface via the JTAG chain. The JTAG Register will be used to write to the chain in this case.

Bits 12 and 13 are the empty and full flags for the R22-FIFO-1, and bits 14 and 15 are the empty and full flags for the R12-FIFO-1. When they are zero, they indicate that the corresponding FIFO's are empty or full.

The top 16 bits are the general purposed status bits. They will be defined by the user for his or her specific test project. They are reprogrammable and can be used for reporting any kind of messages to the user. There are two extreme ways of using them. When there are less than 16 (status or error or others) messages to be reported, the user can use one bit for each. When there are more messages, the 16 bits can be used to encoding up to 65536 possible messages. The user can set priorities for reporting errors if there are many of them. In a real application, the user can mix the two extreme ways, i.e., using a few for direct indication and using the others for encoding. The 16 bits data can also be divided into several groups.

These 16 bits are encoded in the **Lctl**. The **Lctl** receives 5 bits of status or error information from each port controller and three bits from the **Vctl**. These are the source for the 16 status bits. Also, when the bit 29 (Clear Error Status bits) in the Control Register is set, all bits representing error messages should be cleared!

For the project in testing the SVX II Upgrade Readout Electronic System, bits 16 and 17 are used for reporting the G-Link transmitter ready to transfer (i.e., the RFD signal from the transmitter) and the G-Link receiver ready to receive (i.e., the inverse of the LINKRDY* signal from the receiver). All others are used for reporting errors. See daughter card documents list in Section 1.2 for more details. These error bits are sticky, even though their source may be back to normal. These error bits can be cleared by the bit 29 (Clear Error Status bits) in the Control Register.

Here is a brief list for some of them.

The TAXI Receiver VLTN bit is caused by the VLTN signal from the TAXI receiver chip. It means that a data transfer error is found by the TAXI receiver.

The G-Link Receiver Error bit is caused by the ERROR signal from the G-Link receiver chip. It means that a data transfer error is found by the G-Link receiver.

When the TAXI Overwrite bit is high, it means that the TAXI transmitter has been overwritten. This can only happen if the two input register of the TAXI transmitter is loaded and a third writing is issued. The third data will be dropped by the TAXI transmitter in this case. Such the transfer rate has to be reduced in order for the right data pattern to be passed.

The status Register is read only!

3.2.5 T11-FIFO

Table 8 provides the T11-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
27-31	Not-Valid	
26	STOP-BIT	end a testing data sequence
25	T11-DAT-25	also sent to the P1ctl
0-24	T11-DAT-(00..24)	drive the DAT-T of the T1-Port

Table 8. T11-FIFO

The T11-FIFO is used to store the testing data sequences. It is a write only fifo to the VME interface. The lower 26 bits (bits 0-25) of the data in this fifo are sent to the data buffer that drive the 26 DAT-T pins of the T1-Port. Bit 26 is sent to the port controller (**P1ctl**) for the T1-Port. Also the bit 25 is sent to the **P1ctl**.

T1-Port has to be enabled before this fifo can be used!

The lower 26 bits of the T11-FIFO are used for the testing data pattern. Normally one or more sequences of testing data pattern will be loaded into this fifo. Bit 26 is used to mark the end of a data sequence. When it is set, it stops the process of reading and transmitting of the current data sequence. However, due to the delay of the port controller, bit 26 has to be set one cycle (or one word) early than the real end of the sequence.

Bit 25 is also passed to the **P1ctl** (a reprogrammable logic device) for more sophisticated control in the future. By reprogramming the **P1ctl**, this bit can be used for anything (e.g., when it is set, the P1ctl will generate a control signal to the daughter card). But it is not used for anything right now.

The T11-FIFO will operate according to the four transmitting control bits in the Control Register if the T1-Port is selected (enabled) in the Mode Register. Once the transmitting process started, the **P1ctl** will read one word per cycle from this FIFO and send it to the data buffer that drive the DAT-T pins of the T1-Port connector.

Remember, some of the data bits can also be used for passing data-related control information to the daughter card. For example, 2 of the 26 available data bits could be used as the data-strobe and the command-strobe for the transmitted data sequence, so that the daughter card could use them to send the right control signal(s) to its target.

3.2.6 T12-FIFO

Table 9 provides the T12-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
23-31	T12-DAT-(09-17)	drive the DATA-IO for the T1-Port
14-22	T12-DAT-(00..08)	drive the DATA-IO for the R1-Port
0-13	Not-Valid	

Table 9. T12-FIFO

The T12-FIFO is the combination of two FIFO's, the T12-FIFO-1 and the T12-FIFO-2. They are used to store the sequences of testing data pattern for the DATA-IO pins of the T1-Port connector and the R1-Port connector. The upper 9 bits (bits 23-31) of this T12-FIFO is the T12-FIFO-1 and the data are sent to the data buffer that drive the 9 DATA-IO pins of the T1-Port. The next 9 bits (bits 14-22) is the T12-FIFO-2 and they are sent to the data buffer that drive the 9 DATA-IO pins of the R1-Port. These data buffers may or may not be enabled depending on the configuration of the T1-Port and the R1-Port. The T12-FIFO is a write only fifo to the VME interface.

The upper 2 bits for both 9-bit groups are passed to the **P1ctl** (a reprogrammable logic device) for more sophisticated control in the future (project specific) if the data buffers are enabled (actually, the top 2 DATA-IO pins are routed to the **P1ctl**).

The input side (the VME bus interface side) of the T12-FIFO is independent from the T11-FIFO. But the output side (the transmitting side) of the T12-FIFO is a parallel extension to the T11-FIFO, such that the T12-FIFO and T11-FIFO will always work together in a transmitting process regardless whether the T12-FIFO's are used or not. Of course, part of the bits in both the T12-FIFO's and the T11-FIFO may not be used in a real application.

3.2.7 R11-FIFO

Table 10 provides the R11-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
27-31	Not-Valid	
25-26	R11-DAT-(25,26)	as others, but also to P1ctl
0-24	R11-DAT-(00..24)	from the DAT-R of the R1-Port

Table 10. R11-FIFO

The R11-FIFO is used to store the received data sequences from the R1-Port. It is a read only fifo to the VME interface.

The input to the R11-FIFO is driven by a (flip-flop) register connected to the DAT-R pins of the R1-Connector. It is always enabled and clocked by a clock signal generated by the P1-Controller (**P1ctl**). By default, this clock signal is the inverse of the system clock SYS-CLK (which will be about the same as the SYS-CLK for a 53 Mhz clock and advanced by a few ns). But this clock signal could be reprogrammed to be driven by the data receiving clock of the daughter card passed from the P-Status signal. This will ensure the synchronization of the received data and avoid the problem that may arise due to the synchronized receiving of the daughter card. It is suggested that the daughter card itself should try to synchronize the data to the GSTM's SYS-CLK before using the final resources for the received data.

If no transmitting port is selected and the R1-Port is selected, the receiving FIFO's will start the recording process when a programmed trigger signal (first valid data or other signal from the daughter card) is set after the START-RECEIVING bit in the Control Register is set. Once the process started, the port controller will read one word from the receiving port connector and write it to this receiving fifo on every cycle. The process will be stopped only when the fifo is full or the STOP-RECEIVING bit is set in the Control Register. The port controller can be reprogrammed to stop the process on error or whatever.

If one or more transmitting ports are selected, then the recording process will be controlled by the selected transmitting port. However, the data may be delay by a few clocks due to the delay in the port controller and the delay in the receiving daughter card. When the transmitting FIFO's get a retransmitt signal, the receiving FIFO's get a reset signal.

Pins for DAT-R-25 and DAT-R-26 (corresponding to bits 25 and 26 for the R11-FIFO, but before the flip-flop data register) of the R1-Connector are also passed to the **P1ctl** (a reprogrammable logic device) for more sophisticated control in the future. Right now, the two pins are programmed for passing the /data-strobe and the /command-strobe for the R1-Port, which are used to start the recording of the incoming received data. In the case of the G-Link receiver daughter card, the two strobe signals (DAV* and CAV*) should be combined with the LINK-RDY* signal passed from one of the P-Status.

If it is necessary, the **P1ctl** can be reprogrammed to store only valid data into the R11-FIFO as discussed in the section 2.3. Or in some cases, it can be reprogrammed to allow the daughter card on R1-Port to control the receiving fifo directly.

3.2.8 R12-FIFO

Table 11 provides the R12-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
23-31	R12-DAT-(09-17)	from the DATA-IO for the T1-Port
14-22	R12-DAT-(00..08)	from the DATA-IO for the R1-Port
0-13	Not-Valid	

Table 11. R12-FIFO

The R12-FIFO is the combination of the two FIFO's, the R12-FIFO-1 and the R12-FIFO-2. They are used to store the received data sequences from the DATA-IO pins of the T1-Port connector and the R1-Port connector. The upper 9 bits (bits 23-31) of this fifo (actually a fifo group) is the R12-FIFO-1 and the data for them come from the data buffer (flip-flop register) that is driven by the 9 DATA-IO pins of the T1-Port. The next 9 bits (bits 14-22) is the R12-FIFO-2 and the data for these 9 bits come from the data buffer (flip-flop register) that is driven by the 9 DATA-IO pins of the R1-Port. These data buffers are always enabled. The R12-FIFO is a read only fifo to the VME interface.

Even though the two FIFO's in this fifo group is read together from the VME interface, the two FIFO's work differently (empty fifo is still readable). The R12-FIFO-2 is a parallel extension to the R11-FIFO and it always works together with the R11-FIFO regardless how the DATA-IO pin is configured. Such, all the discussion on the R11-FIFO are applied to the R12-FIFO-2.

But the R12-FIFO-1 is used to save the received data stream from the T1-Port even when R1-Port is not selected. Such the control of the input to the R12-FIFO-1 follows the control of the output of the T11-FIFO exactly except that one is for reading and the other is for writing. Also, a retransmit signal for the T11-FIFO will result a reset signal to the R12-FIFO-1.

Pins for the DATA-IO-7 and the DATA-IO-8 (corresponding to the upper two bits for the R12-FIFO-1, or bits 30 and 31 for the R12-FIFO, but before the flip-flop data register) of the T1-Connector and pins for the DATA-IO-7 and the DATA-IO-8 (corresponding to the upper two bits of the R12-FIFO-2, or bits 21 and 22 for the R12-FIFO, but before the flip-flop data register) of the R1-Connector are also passed to the P1-Controller (a reprogrammable logic device) for more sophisticated control in the future. Right now, they are not used for anything yet.

3.2.9 T21-FIFO

Table 12 provides the T21-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
27-31	Not-Valid	
26	STOP-BIT	end a testing data sequence
25	T21-DAT-25	also sent to the P2ctl
0-24	T21-DAT-(00..24)	drive the DAT-T of the T2-Port

Table 12. T21-FIFO

The T21-FIFO is used to store the testing data sequences. It is a write only fifo to the VME interface. The lower 26 bits (bits 0-25) of the data in this fifo are sent to the data buffer that drive the 26 DAT-T pins of the T2-Port. Bit 26 is sent to the port controller (**P2ctl**) for the T2-Port. Also the bit 25 is sent to the **P2ctl**.

T2-Port has to be enabled before this fifo can be used!

The lower 26 bits of the T21-FIFO are used for the testing data pattern. Normally one or more sequences of testing data pattern will be loaded into this fifo. Bit 26 is used to mark the end of a data sequence. When it is set, it stops the process of reading and transmitting of the current data sequence. However, due to the delay of the port controller, bit 26 has to be set one cycle (or one word) early than the real end of the sequence.

Bit 25 is also passed to the **P2ctl** (a reprogrammable logic device) for more sophisticated control in the future. By reprogramming the **P2ctl**, this bit can be used for anything (e.g., when it is set, the P1ctl will generate a control signal to the daughter card). But it is not used for anything right now.

The T21-FIFO will operate according to the four transmitting control bits in the Control Register if the T1-Port is selected (enabled) in the Mode Register. Once the transmitting process started, the **P2ctl** will read one word per cycle from this FIFO and send it to the data buffer that drive the DAT-T pins of the T2-Port connector.

Remember, some of the data bits can also be used for passing data-related control information to the daughter card. For example, 2 of the 26 available data bits could be used as the data-strobe and the command-strobe for the transmitted data sequence, so that the daughter card could use them to send the right control signal(s) to its target.

3.2.10 T22-FIFO

Table 13 provides the T22-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
23-31	T22-DAT-(09-17)	drive the DATA-IO for the T2-Port
14-22	T22-DAT-(00..08)	drive the DATA-IO for the R2-Port
0-13	Not-Valid	

Table 13. T22-FIFO

The T22-FIFO is the combination of two FIFO's, the T22-FIFO-1 and the T22-FIFO-2. They are used to store the sequences of testing data pattern for the DATA-IO pins of the T2-Port connector and the R2-Port connector. The upper 9 bits (bits 23-31) of this T22-FIFO is the T22-FIFO-1 and the data are sent to the data buffer that drive the 9 DATA-IO pins of the T2-Port. The next 9 bits (bits 14-22) is the T22-FIFO-2 and they are sent to the data buffer that drive the 9 DATA-IO pins of the R2-Port. These data buffers may or may not be enabled depending on the configuration of the T2-Port and the R2-Port. The T22-FIFO is a write only fifo to the VME interface.

The upper 2 bits for both 9-bit groups are passed to the **P2ctl** (a reprogrammable logic device) for more sophisticated control in the future (project specific) if the data buffers are enabled (actually, the top 2 DATA-IO pins are routed to the **P2ctl**).

The input side (the VME bus interface side) of the T22-FIFO is independent from the T21-FIFO. But the output side (the transmitting side) of the T22-FIFO is a parallel extension to the T21-FIFO, such that the T22-FIFO and T21-FIFO will always work together in a transmitting process regardless whether the T22-FIFO's are used or not. Of course, part of the bits in both the T22-FIFO's and the T21-FIFO may not be used in a real application.

3.2.11 R21-FIFO

Table 14 provides the R21-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
27-31	Not-Valid	
25-26	R21-DAT-(25,26)	as others, but also to P2ctl
0-24	R21-DAT-(00..24)	from the DAT-R of the R2-Port

Table 14. R21-FIFO

The R21-FIFO is used to store the received data sequences from the R2-Port. It is a read only fifo to the VME interface.

The input to the R21-FIFO is driven by a (flip-flop) register connected to the DAT-R pins of the R2-Connector. It is always enabled and clocked by a clock signal generated by the P2-Controller (**P2ctl**). By default, this clock signal is the inverse of the system clock SYS-CLK (which will be about the same as the SYS-CLK for a 53 Mhz clock and advanced by a few ns). But this clock signal could be reprogrammed to be driven by the data receiving clock of the daughter card passed from the P-Status signal. This will ensure the synchronization of the received data and avoid the problem that may arise due to the synchronized receiving of the daughter card. It is suggested that the daughter card itself should try to synchronize the data to the GSTM's SYS-CLK before using the final resources for the received data.

If no transmitting port is selected and the R2-Port is selected, the receiving FIFO's will start the recording process when a programmed trigger signal (first valid data or other signal from the daughter card) is set after the START-RECEIVING bit in the Control Register is set. Once the process started, the port controller will read one word from the receiving port connector and write it to this receiving fifo on every cycle. The process will be stopped only when the fifo is full or the STOP-RECEIVING bit is set in the Control Register. The port controller can be reprogrammed to stop the process on error or whatever.

If one or more transmitting ports are selected, then the recording process will be controlled by the selected transmitting port. However, the data may be delay by a few clocks due to the delay in the port controller and the delay in the receiving daughter card. When the transmitting FIFO's get a retransmitt signal, the receiving FIFO's get a reset signal.

Pins for DAT-R-25 and DAT-R-26 (corresponding to bits 25 and 26 for the R21-FIFO, but before the flip-flop data register) of the R2-Connector are also passed to the **P2ctl** (a reprogrammable logic device) for more sophisticated control in the future. Right now, the two pins are programmed for passing the /data-strobe and the /command-strobe for the R2-Port, which are used to start the recording of the incoming received data. In the case of the G-Link receiver daughter card, the two strobe signals (DAV* and CAV*) should be combined with the LINK-RDY* signal passed from one of the P-Status.

If it is necessary, the **P2ctl** can be reprogrammed to store only valid data into the R21-FIFO as discussed in the section 2.3. Or in some cases, it can be reprogrammed to allow the daughter card on R2-Port to control the receiving fifo directly.

3.2.12 R22-FIFO

Table 15 provides the R22-FIFO bit assignment.

BIT	MEANING	EXAMPLE/NOTES
23-31	R22-DAT-(09-17)	from the DATA-IO for the T2-Port
14-22	R22-DAT-(00..08)	from the DATA-IO for the R2-Port
0-13	Not-Valid	

Table 15. R22-FIFO

The R22-FIFO is the combination of the two FIFO's, the R22-FIFO-1 and the R22-FIFO-2. They are used to store the received data sequences from the DATA-IO pins of the T2-Port connector and the R2-Port connector. The upper 9 bits (bits 23-31) of this fifo (actually a fifo group) is the R22-FIFO-1 and the data for them come from the data buffer (flip-flop register) that is driven by the 9 DATA-IO pins of the T2-Port. The next 9 bits (bits 14-22) is the R22-FIFO-2 and the data for these 9 bits come from the data buffer (flip-flop register) that is driven by the 9 DATA-IO pins of the R2-Port. These data buffers are always enabled. The R22-FIFO is a read only fifo to the VME interface.

Even though the two FIFO's in this fifo group is read together from the VME interface, the two FIFO's work differently (empty fifo is still readable). The R22-FIFO-2 is a parallel extension to the R21-FIFO and it always works together with the R21-FIFO regardless how the DATA-IO pin is configured. Such, all the discussion on the R21-FIFO are applied to the R22-FIFO-2.

But the R22-FIFO-1 is used to save the received data stream from the T2-Port even when R2-Port is not selected. Such the control of the input to the R22-FIFO-1 follows the control of the output of the T21-FIFO exactly except that one is for reading and the other is for writing. Also, a retransmit signal for the T21-FIFO will result a reset signal to the R22-FIFO-1.

Pins for the DATA-IO-7 and the DATA-IO-8 (corresponding to the upper two bits for the R22-FIFO-1, or bits 30 and 31 for the R12-FIFO, but before the flip-flop data register) of the T1-Connector and pins for the DATA-IO-7 and the DATA-IO-8 (corresponding to the upper two bits of the R12-FIFO-2, or bits 21 and 22 for the R12-FIFO, but before the flip-flop data register) of the R1-Connector are also passed to the P2-Controller (a reprogrammable logic device) for more sophisticated control in the future. Right now, they are not used for anything yet.

3.2.13 Reserved Register,

The Reserved Register will generate a signal on the **Vctl** and pass it to the **Lctl**. It is currently not used and is reserved for future upgrades.

3.2.14 JTAG Register

When the Altera EPM7128SQC100's are installed in the final GSTM mother board, the **Lctl** and the two port controllers could be reprogrammed through the VME interface by using the JTAG Register. This is a more convenient way of implementing new logics on these logic devices than the typical way of reprogramming through the JTAG cable supplied by Altera. The **Vctl** (VME controller for GSTM) is not expected to be changed very often. Such the **Vctl** is designed with the capability of driving the on board JTAG chain for reprogramming the three programmable logic devices. A jumper in the GSTM front panel will select whether the JTAG cable connector (also in the front panel) or the **Vctl** should control the JTAG chain.

BIT	MEANING
3	Generate and Drive the TCK line
2	Generate and Drive the TMS line
1	Generate and Drive the TDI line
0	Not-Used

Table 16. R12-FIFO

Table 16 provides the JTAG Register bit assignment. It is a write-only register. The JTAG Register has only 4 bits and 3 bits are used for generating the TDI, TMS and the TCK signals. If the **Vctl** is selected to drive the JTAG chain, these signals will be driven to the chain. Otherwise, no effects will be generated outside the **Vctl**. The Status Register (SR) should be used to get the TDO signal (driven by the last device in the chain) back.

A software tool should be used for reprogramming the three logic devices (the **Lctl** and the two port controllers) by accessing the JTAG Register once the reprogramming data files for the devices are completed. This register should not be accessed for any other purpose or by any other means.

4 ELECTRICAL & MECHANICAL SPECIFICATIONS

4.1 Packaging & Physical Size

The GSTM will be a standard double width 9U X 400mm VME card. When the GSTM daughter card is mounted to the GSTM mother board, the GSTM mother board/daughter card combination should not be any higher than a double width VME module.

The GSTM mother board can accommodate three different daughter card sizes, 3.3" x 6", 3.3" x 8", and 3.3" x 13" as show in **Figure 5**. Components may be mounted on both sides of the daughter card, but a component mounted on the solder side of the daughter card may be no higher than .3". The daughter cards will be secured to the GSTM module with from any where from four to eight 4-40 screws.

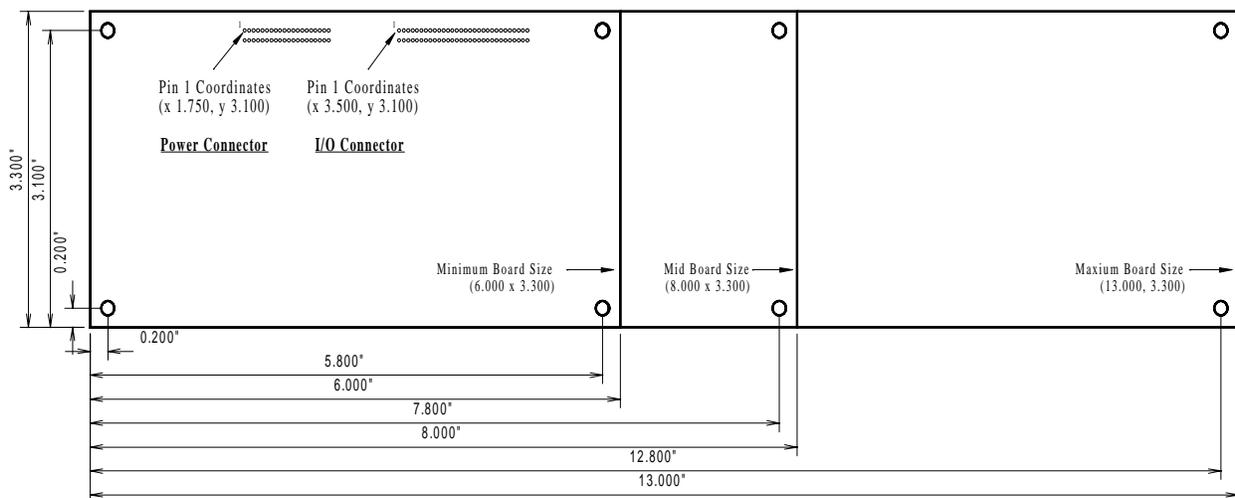


Figure 5: GSTM Daughter Card Dimensions

4.2 Port Connectors

Each daughter card will mate with the GSTM mother board through two connectors, a 40 pin power/ID connector, and a 60 pin signals connector. The connectors will be keyed so a transmitting daughter card can not be plugged into a receiving daughter card port and vice versa. The GSTM mother board can have up to 4 daughter cards, two transmitting daughter cards and two receiving daughter cards.

The pin assignment for the transmitting port connector is given in **Figure 4**, where the DAT-T's are used instead of the DAT-R's. The connector is a 60 pin AMP AMPMODU System 50 connector, AMP part number 104078-6. The Transmitter daughter cards will use a **female** style connector. The Receiver daughter cards use a male connector, this will provide the keying

necessary so that a Transmitter daughter card can not be plugged into a Receiver daughter card connector.

Similarly, the pin assignment for the receiving port connector is also given in **Figure 4**, where the DAT-R's are used instead of the DAT-T's. The connector is a 60 pin AMP AMPMODU System 50 connector, AMP part number 104068-6. The Receive daughter cards will use a **male** style connector.

Figure 6 shows the pinout of the Power/ID connector. The Power/ID connector is used to supply power to the daughter cards, and the daughter card should use the 6 ID bits to tell the GSTM mother board what type of daughter card is being plugged in. The connector is a 40 pin AMP AMPMODU System 50 **female** connector, AMP part number 104078-2. ID's for the daughter cards will be issued by the ESE group.

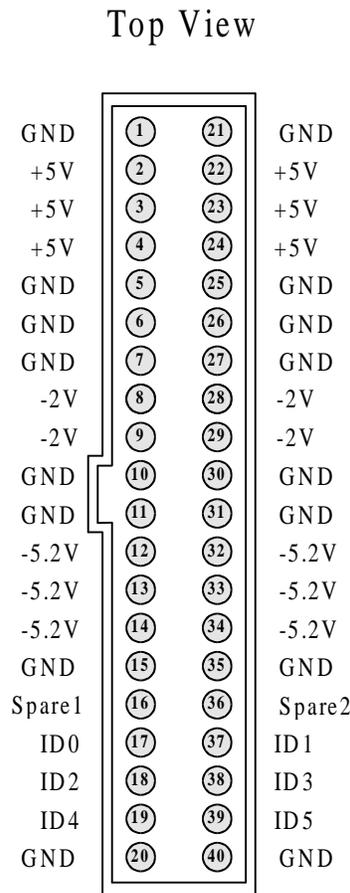


Figure 6: The GSTM Power/ID Connector

4.3 Power & Cooling Requirements

The GSTM mother board consumes less than 20 watts of power.

The GSTM daughter cards will be limited to 20 watts of power consumption per daughter card. The power supplied for the -5.2V is limited to be 7.5 watts.

The entire GSTM module will require no special cooling otherwise supplied by a standard VME crate.

4.4 Jumpers and Switches

Jumpers:

There are several jumpers on the GSTM mother board. Most of them are spares, and they can be used special situations. Jumpers J5, J7, J6 and J8 (near the 4 ALTERA chips on the GSTM mother board) , also marked as SPR1, SPR2, SPR3, and SPR4 are spare jumpers that are used to the special functional pins for the ALTERA reprogrammable devices. They are the global CLK and OE pins.

These jumpers can be used either for send a special signals to the reprogrammable devices or for special applications which may need some hardware modifications. One example for the first case is that a jumper is used to select between two different internal configurations that generate different control sequence. One example for the second case is that a few pins of the device can be activated or tri-stated by the global OE function of the pin connected to the jumper which may be hardwired to a control signal generated by other reprogrammable devices.

Jumper J8, also marked as VME BitBlaster, is used to select the source for reprogram the **P1ctl**, **P2ctl** and **Lctl**. If it is open, the normal ALTERA BitBlaster port is used to reprogram the three reprogrammable devices. Otherwise, the **Vctl** is used (through the VME interface by using a special program) to reprogram the three chips.

Switches:

There are 4 DIP There are 4 DIP switches on the GSTM mother board. They are named as SW1, SW2, SW3 and SW4. The SW1 is located near the **Lctl** at the middle of the board and the other 3 are located near the VME Interface connector.

The SW4 is a 6 bits DIP switch. It is used to set the VME address modifier for GSTM module. The address modifier on the VME bus has to match the setting of this DIP switch before the GSTM module response to the VME bus. Bits 1, 2, 3, 4, 5, and 6 are corresponding to the AM0, AM1, AM2, AM3, AM4 and AM5 respectively. When a bit is at the on position, the corresponding AM bit is set to zero. Otherwise, at the off position, the corresponding AM bit is set to one. Although the SW4 can be set to anything, it is recommended to set it to 0x09 only, unless a special situation is needed.

The SW3 and SW2 are 8 bits DIP switches that are used to set the VME base address for the GSTM module. The 8 bits at the SW3 set the bits A24 to A31 and the 8 bits at the SW2 set the bits A16 to A23. They are marked clearly on the GSTM mother board. The SW2 is always enabled. But the SW3 is only effective when the GSTM module is in a old VME crate or with a extension card (i.e., without the active GA's lines). When a new VIPA crate is used, where the GA's lines are driven by the VME bus, the SW3 ignored. See the discussion in Section 3.1.

The SW1 is a 3 bits DIP Switch. The 3 bits are marked as ck_0, ck_1 and ck_2. Only the ck_0 is currently used. The SW1 is used to send 3 signals to the **Lctl** to help reconfigure the **Lctl**'s functionalities. The **Lctl** is reprogrammable to accommodate the different configurations for various applications. For example, the ck_0 right now is used as the system clock phase inverter. When it is switched from off to on or from on to off, the GSTM system clock will invert its phase, i.e., the high becomes low and low becomes high. What actually happened is that the system clock will start with the different phases of the incoming clock source depends on the setting of the ck_0 bit. This is useful in the case that the incoming data from a daughter card (which is not synchronized with the mother board) may violate the setup or hold time for the input Flip-Flop register. Inverting the clock will remove this problem.

4.5 Front Panel

There are 3 LEDs (2 yellow and 1 red) near the top of the front panel of the GSTM mother board and there are 2 LEDs (blue) and one reset push-button near the bottom of the front panel of the GSTM mother board.

Of the 2 LEDs (blue color) at the bottom, the up LED is a power indicator. It is turned on when the GSTM gets the 5V power. The lower LED is the JTAG power indicator.

The push button is a master reset button. It will reset the whole module, mother board and all the daughter cards on the module.

The top yellow LED is the VME selection indicator. It is turned on whenever the GSTM is successfully accessed by the VME bus. The middle yellow LED is turned on only when the GSTM is actively transmitting data.

The red LED near the top (but below the 2 yellow LEDs) is a user defined and reprogrammable LED. It is controlled by the Lctl. It can be reprogrammed by the user for the specific application. For example, in the case of G-Link receiver draught card, it is used to indicate the transition for the system clock from GSTM crystal oscillator to the received clock of the G-Link. It is off first. Then it is turned on when the G-link receiver is locked into the transmitter and received the right clock.

Appendix A)

Typical operation for a test:

First, for each test (set up), the Mode Register should be set properly. This includes to select/enable the ports that will be used for the test, to configure the selected port with its I/O bits, and to set the daughter cards to the right mode(s). Note, additional mode bits may be optionally set depending on the daughter card design. For example, the G-Link transmitter or receiver daughter card actually use one of the three P-cntrl signals to receive additional mode or configuration information from one of the user-defined mode bits in the Mode Register.

Second, the testing data pattern (data sequences) for all the transmitting FIFO's should be loaded through the VME bus. Depending on how the GSTM will be used, the user may need to load not only the data to be transmitted by the transmitting ports, but also the data streams for driving the DATA-IO for all ports. Remember, any transmitting port may be used to control a receiving port for recording an incoming data stream with a fixed length. Therefore, dummy driving sequence for this purpose should also be loaded.

The GSTM uses the preloaded data pattern (designed and generated by the user) in the transmitting FIFOs and transmits them to the testing modules (target system) as a command (control signals) sequence. Also, the GSTM will save the received sequence of incoming (or returned) data pattern into the receiving FIFOs, which will be readout and checked.

Then, the Control Register is used to control the emulation or simulation sequence of the targeted system. The Status Register is used to monitor the status of the emulation sequence. Also, the received data streams from all selected receiving ports or receiving DATA-IO channels may be readout and verified.

After data verification, a different control or data sequence can be sent if more sequences are in the transmitting fifo, or new sequences can be loaded and sent, and the emulation procedure stated above can be repeated.

Other features:

The GSTM allows several sequences of data pattern to be loaded into the transmitting FIFOs and single-stepped through one by one manually. This is done by setting the STOP-BIT after each data sequence. When the Start-Transmitting bit in the Control Register is set, a sequence is read and transmitted. The STOP-BIT at the end of each sequence terminates the transmitting process for the sequence. The Status Register can be read to determine whether the sequence is finished or not. The transmit led in the front panel can also be used to indicate the starting and ending of the transmitting process if timing is not a problem since the led could be delayed by 100 ms for the human eyes.

In cases where a transmitted data pattern requires no action on the receiver side, the STOP-BIT between the sequences can be removed and one or more of the NO-OPERATION

data patterns can be inserted. In this way, multiple data sequences will run continuously for high speed testing.

The GSTM also allows a loopover mode to repeat a data sequence continuously, i.e., the sequence can be transmitted over and over again until it is forced to stop. This should be useful for debugging and testing a target system.

A Stop-On-Error mode can also be set to force the stop of the transmitting process when an error is encountered, or the daughter card can stop the process by issue a faked error. If the mode bit is not set, the process will continue and errors are ignored.

In normal situations, the receiving ports are controlled by the transmitting ports. If no transmitting ports are enabled, the receiving port will continue recording the received data until it is forced to stop or the fifo is full (The port controller can be reprogrammed if the user do not want to do it in this way) once the recording process started. However, if one or more transmitting ports are selected, the receiving port will start and stop in concert with the transmitting ports. Therefore, the GSTM can be used to receive a data sequence of certain length determined by the transmitting port. This could be done by loading a transmitting port with a data sequence of the length desired, which may contain only a STOP-BIT at the end. Start up the transmitting process causing the receiving port to record the incoming data sequence until the STOP-BIT occurs. This trick can be extended to allow the GSTM to receive multiple data sequences, in which each sequence has a known length.

Use the GSTM for other applications:

As mentioned earlier, the GSTM can be used to emulate the intercommunication links between modules in a communication intensive electronic system by driving and receiving multiple streams of data. Therefore, it is possible for many system modules to be tested in this manner without the existence of the entire system. This is the main purpose for the GSTM.

However, the GSTM can actually be used for many other applications limited only by the user's imagination. There are two main reasons for this. First, the GSTM provides a flexible interface between the mother board and its daughter cards. Remember the 3 P-cntrl and the 3 P-status signals? They make the project specific control achievable. The protocol can be easily customized to fit the specialized requirement of the system. The port controller which controls the interface can be reprogrammed and customized information can be passed between the GSTM mother board (therefore to the user) and its daughter cards. Also, the GSTM VME registers (the Control Register, the Mode Register, the Status Register as well as many of the FIFOs) have many user-defined bits which can be changed to meet the requirement of the test project.

Second, no limit has been placed on the functionality of the daughter card. In fact, the daughter cards can contain sufficient logic to accomplish complex test. For example, two transmitting ports can be used to test the reliability (bit error rate) of a high speed communication link. This could be a G-Link or a TAXI or many others. One of the transmitting ports would implement the transmitter daughter card and the other would implement the receiver daughter card. The receiver daughter card would have a programmable logic device (an Altera or a Xilinx chip) and a timer and would be designed to compare two data streams; one from the receiver and

the other from the transmitting port of the mother board. Both ports would be loaded with the same data sequence with the GSTM in a loopover mode. The transmitter daughter card will send the data stream to the receiver card. The receiver card will compare the received data stream with the data stream from the mother board on the fly so that the data are checked instantaneously. A counter would be used to record the number of errors detected within a certain period of time. The counter (< 9bits) value could be sent to the GSTM mother board at the end of each time period. In this way, the error rate would be a time ordered sequence during long time periods. Therefore, the error rate as a function of time can be displayed!