

3. V6M6 Baseboard

This chapter describes the baseboard for the V6M6. It is organized into the following sections:

3. V6M6 Baseboard

3.1 Overview	3-1
3.2 Primary Components	3-4
3.3 Configuration Microcontroller (CM)	3-5
3.4 PCI Bus	3-8
3.5 TDM Subsystem	3-12
3.6 VME/PCI Bridge	3-22
3.7 Software Support	3-22

3.1 Overview

The V6M6 is a 6U VME carrier for up to six plug-in modules. The V6M6 and its modules together occupy a single VME slot.

A variety of module types are available to perform general purpose processing, digital signal processing, and I/O interface functions. Any mix of module types may be installed on the V6M6, subject to certain physical limitations such as I/O connector accessibility.

The six module sites on the V6M6 are interconnected by two buses. The first is a PCI bus, electrically compatible with the standard PCI bus used in PC applications. The second bus is a proprietary Time Division Multiplexed (TDM) bus typically used to carry isochronous (constant bit rate) data such as digitized audio.

In addition to the six module sites interconnected by PCI and TDM buses, the V6M6 includes:

- DRAM controller and 0, 32MB, or 64MB of DRAM attached to the PCI bus
- VME bridge attached to the PCI bus
- Configuration microcontroller to handle V6M6 initialization
- TDM bus controller to manage TDM bus activity
- Programmable clock synthesizers to generate PCI and TDM clocks
- 7 LEDs to display VMEbus activity and module-specific status
- Board reset and VME enable switches
- TDM bus front-panel expansion connector
- Connector to provide VME backplane P2 connector access to one of the six module sites

In addition, the "S" option of the V6M6 (i.e. V6M6/S) includes an interface between the TDM bus and the standard VITA 6-1994 SCSA bus on the VME P2 connector.

The host (VME) interface provides PIO ports to access the module sites and global memory. Each PIO port is implemented as a set of three registers: a Control / Status Register, an Address Register, and a Data Register. Access to these registers is controlled via the Unix device interface by a special device driver. A library of C-callable functions is provided for application programmers to access and utilize the interface for data transfer and module control operations.

Figure 1, below, is a general outline of the V6M6 showing the location of the module sites, and VME address switches.

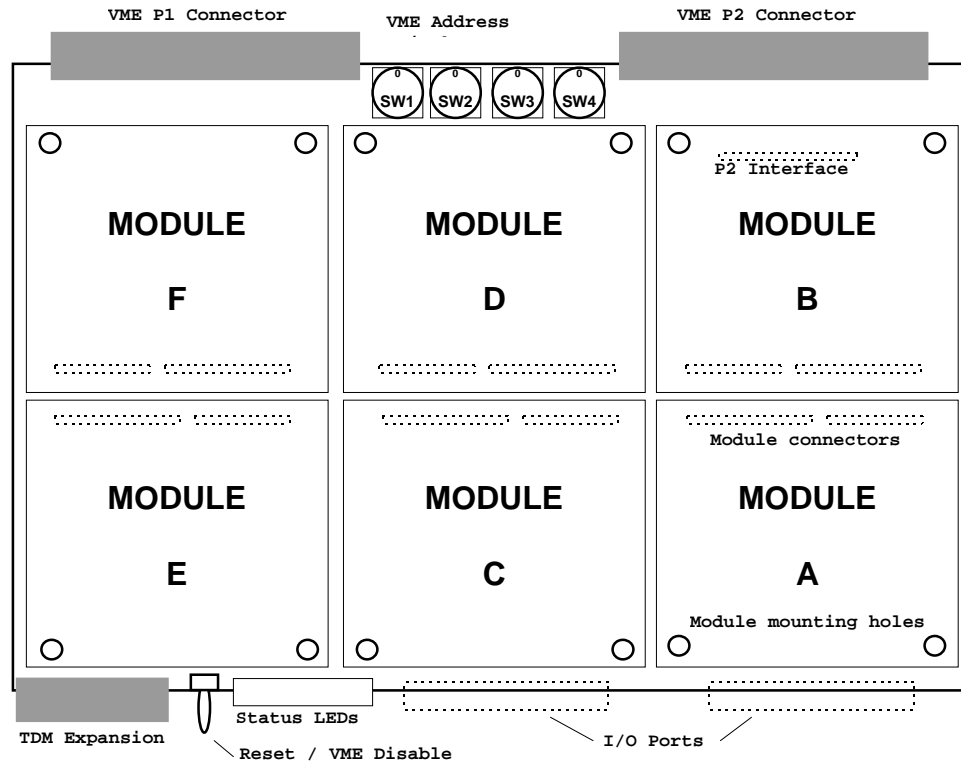


Figure 1: V6M6 Outline

3.2 Primary Components

MODULE A-F	The PCI Module Sites. These are the locations of the modules on the baseboard.
VME Connectors	Connectors P1 & P2 plug into the VME backplane.
SW1 - SW4	These select the VME address for the board. Their settings correspond to the upper 4 hex digits of the address (SW1 is the most significant).
TDM Expansion	This port connects via 20-pin ribbon cable to the TDM expansion ports of other V6M6 or VME6U6 DSP boards to combine TDM subsystems.
Reset / VME	This dual function switch is used to reset and reconfigure the V6M6 and to disable its VME interface. Toggling the switch towards the TDM connector (momentary position) will initiate a reset cycle. Toggling the switch towards the LEDs (stationary position) disables the VME interface.
Status LEDs	One tri-color LED for each module and one for the VME interface. These LEDs may take on three color states, RED, GREEN or AMBER, as well as OFF. The use of the module status LEDs is dependent on the module type installed. The VME status flashes green to indicate slave access and red when the V6M6 is VME master.
I/O Ports	Module sites A and C may have an I/O port accessed from the front panel. For example, RJ45 connectors for T1 or E1, special multi-channel audio connectors or serial ports for MIPS processors. Various front panel inserts are available for the different I/O connectors.

3.3 Configuration Microcontroller (CM)

The logic on the V6M6 baseboard, and on most of its plug-in modules, is implemented in field programmable gate arrays (FPGAs). The FPGAs used are primarily Xilinx and Lucent Technologies devices based on SRAM technology. These devices must be initialized with a configuration bit stream before they become functional.

The primary responsibility for configuring the FPGAs on the baseboard and on installed modules rests with a microcontroller subsystem on the V6M6 baseboard called the Configuration Microcontroller (CM). The CM is built around an 8051-compatible microcontroller from Dallas Semiconductor, the DS80C310. It does not rely on any FPGA logic itself, and so becomes functional immediately after power-up or board reset.

The CM's program is stored in two separate nonvolatile memories. The first memory is a 4 Mbit Flash EPROM, and the second is a standard 27C256 256 Kbit EPROM. These are described in later subsections. In addition, the CM owns a 1024-bit serial EEROM used to hold information such as desired PCI and TDM clock frequencies, baseboard serial number, and burn-in history.

Modules installed on the V6M6 identify themselves to the CM via a 1024-bit serial EEROM on each module. The EEROM contains a module-type indicator and other information. The CM queries the EEROM on each module to determine which FPGA configuration bit stream, if any, is required by the module. The serial EEROM is described in more detail in Section 3.3.3 Serial EEROMs.

The CM cannot be used as a general purpose processing resource on the V6M6. The only non-configuration functions performed by the CM are the generation of a 62.4 KHz timing signal distributed to the six module sites and control of the PCI and TDM clock synthesizer chips described later.

3.3.1 Microcontroller Flash Memory

The microcontroller's main memory is a 4 Mbit Flash EPROM that holds the current version of the CM's program as well as the current version of the configuration bit stream for each FPGA that the CM might need to initialize.

The V6M6 software distribution includes utility programs which allow the VME host computer to update the contents of the Flash memory. This could occur as a result of:

- Adding support for a new module type. If the new module includes an SRAM-based FPGA, a new configuration bit stream must be added to the Flash memory.
- Upgrading an existing FPGA configuration bit stream. This might be to correct a problem in a previous version or to add new functionality.
- Upgrading the CM program itself. This is rare, but is supported in case some new module design requires an enhancement to the configuring software.

3.3.2 Microcontroller EPROM Memory

The 27C256 EPROM is permanently mounted to the V6M6. It is preprogrammed when the V6M6 is manufactured with a basic CM program and with basic operating configurations for the FPGAs on the V6M6 baseboard.

The purpose of the EPROM is to have a way of getting the V6M6 operational enough to allow the Flash memory to be downloaded during initial board test. In addition, the EPROM allows recovery from a failed update of the CM program in Flash memory.

The EPROM includes the CM's reset vector, so the CM always begins operating from the program stored in the EPROM. The EPROM program first checks for a signature in the Flash memory to determine if the Flash has been initialized. If not, then the CM continues executing from EPROM and loads a minimum operating configuration into the V6M6 baseboard FPGAs. Once this configuration is complete, the VME host can communicate with the V6M6 to load current software and configuration bit streams into the Flash memory.

If the CM program in Flash has somehow been corrupted, a shunt can be installed on a header on the V6M6 baseboard to force the CM to execute only from the EPROM.

3.3.3 Serial EEROMs

Each V6M6 module must include a 1024-bit serial EEROM (93C46 style) to hold information about the module. The CM queries the 93C46 on each module to determine the module's need for configuration. It then finds the required configuration data in Flash and sends it to the module serially through the module's JTAG port.

The information held in the 93C46 varies depending on the type of module, but typically includes:

- Module type identifier
- PCI bus address type and size
- Hardware revision number
- Serial number
- Date serialized
- Burn-in hours
- Module options (e.g. memory size, clock speed)

This information is available to VME host software through functions in the V6M6 Host Application Library.

The CM itself has an additional 93C46 which holds the desired PCI and TDM clock frequencies, as well as information similar to that in the modules.

3.3.4 Host-Driven FPGA Configuration

For some possible module types, it is not feasible to store the FPGA configuration entirely in the Flash memory. For example, a module for "Configurable Computing" based on a large FPGA would, by its very nature, be subject to frequent changes to its functionality. It would not be reasonable to require new configurations to be downloaded to the Flash and then have the CM load them to the module.

Instead, the CM supports a download protocol which allows the VME host to send new configuration information through the CM and into the destination module FPGA. This transfer bypasses the Flash and can be repeated without resetting the V6M6.

3.4 PCI Bus

The major bus that ties together the elements of the V6M6 board is the PCI bus. It is electrically identical to the PCI bus version 2.1 standardized by the PCI Special Interest Group, commonly used in PCs. The mechanical provisioning—connectors and form factor—of the V6M6 PCI plug-in module is specific to CAC.

The V6M6 supports up to six plug-in PCI modules plus PCI-connected resources on the baseboard, including:

- Global DRAM controller(s). Rev 2 or lower baseboards may have zero, one or two DRAM controllers attached to the PCI bus, each of which can control 32 MB of DRAM. Rev 3 or higher baseboards have at most one DRAM controller which can control up to 64 MB of DRAM.
- VME bridge. This is the path between PCI-connected resources and the VMEbus.
- TDM controller and SCSA interface (Rev 3 and higher). The TDM controller on Rev 2 or lower is not PCI-connected, but instead is controlled solely through the VMEbus. Rev 3 and higher also supports the "S" option to connect the TDM bus to the SCSA bus on the VME P2 connector. The SCSA bus interface circuit is controlled via the same PCI interface as the TDM controller.

3.4.1 PCI Bus Clocking

The PCI specification places a limit of 10 loads on the PCI bus for 33 MHz operation. Traditionally, PCI electrical loading constraints count each plug-in module as two loads. By this counting method, the V6M6 PCI bus can have as many as 15 loads. The V6M6 limits PCI bus speed to 25 MHz max in order to overcome the excess loading and to ease the design task for FPGA-based PCI interfaces.

The PCI bus clock is generated by an ICS AV9110-01, a high resolution frequency synthesizer, controlled by the configuration microcontroller. The CM stores the desired frequency in its serial EEROM memory. The output of the synthesizer is buffered with a separate low-skew driver for each plug-in module and for each PCI interface on the baseboard. The individual clock lines are balanced for equal delay to each destination to minimize clock skew between all of the modules and on-board resources.

Note that the PCI clock line is NOT bused, as it is in typical PCI systems. A separate clock line is driven to each module. Each clock signal is terminated at its end-point with a series combination of a resistor (between 62 and 75 Ω) and a capacitor (between 100 pF and 220 pF) to ground. Module clock signals are terminated on the module, not on the baseboard.

The point-to-point distribution of PCI clock eliminates the requirement from the PCI specification of having only one load on the PCI clock signal on a plug-in module. Multiple loads are permissible as long as they are clustered within 1" of the clock signal terminator.

Because of the way that the clocks are distributed, there is very little skew between module clock signals at the module connectors. Therefore it is not necessary to maintain the precise 2.5" \pm .1" clock signal length on the plug-in module, as called out in the PCI specification. The clock signal length on the module should be in the range of 1" to 2".

The PCI clock generator will never stop or slow to very low rates. Furthermore, its frequency cannot be changed without a hard reset of the V6M6 board. Therefore, it is permissible for plug-in modules to use phase-locked loops to regenerate or multiply the module PCI clock signal. Most CAC modules do, in fact, multiply up the PCI clock to create clocks on the module that are locked to the PCI clock. This greatly simplifies synchronization circuitry in the PCI interface FPGAs on the modules.

3.4.2 PCI IDSEL Signals

The IDSEL signals are produced separately for each module and for each baseboard PCI resource. During PCI configuration cycles, an active high IDSEL to a PCI resource causes that resource to respond to the configuration cycle.

The V6M6 produces IDSEL signals through resistive connections to specific PCI Address/Data lines. The following table shows the correspondence between PCI AD line and IDSEL for each resource.

PCI Resource	PCI AD Line
Module F	19
Module E	20
Module D	21
Module C	22
Module B	23
Module A	24
Mem2/TDM *	25
MemCtl 1	26
VME Bridge	27

* MemCtl 2 on Rev 1 or 2, TDM and SCSA Interface on Rev 3 and up

3.4.3 PCI Arbiter

The V6M6 includes an arbitration circuit that grants control of the PCI bus to PCI resources that request control. This arbiter accepts seven "Request" inputs: one from each module position and one from the VME bridge. It generates seven "Grant" signals back to the same resources.

The arbiter uses a rotating priority method of granting control to requesters. This assures that all PCI requesters are treated equally, and that each will be given control of the PCI bus within a predictable maximum time. The rotating priority scheme views the seven potential requesters in a fixed order:

VME	Mod A	Mod B	Mod C	Mod D	Mod E	Mod F
-----	-------	-------	-------	-------	-------	-------

At any instant, one of these is considered the lowest priority potential requester and the others are considered incrementally higher priority in a circular fashion. The resource most recently in control of the PCI bus is considered the lowest priority. For example, if Module B most recently controlled the bus, then the priority assigned to each resource would be (with 0 lowest priority and 6 highest):

VME	Mod A	Mod B	Mod C	Mod D	Mod E	Mod F
5	6	0	1	2	3	4

The highest priority requester is given a Grant signal and will take control of the PCI bus immediately if no PCI transaction is in progress, or at the end of the current transaction. If no requests are present, the arbiter leaves the bus ownership "parked" on the last resource that controlled the bus.

3.4.4 PCI Interrupts

Each V6M6 module site can source two discrete interrupt request lines. These correspond to INTA/ and INTB/ interrupt signals in the PCI specification. Note that INTC/ and INTD/ are not supported on the V6M6.

The INTA/ and INTB/ signals are not common or bused as in many PC applications. Each is separately wired to an interrupt routing circuit. This circuit can selectively enable the twelve interrupt requests to generate a VMEbus interrupt or to generate interrupt requests back to the on-board PCI modules.

The VMEbus interrupt circuit has a mask register, described in Section 3.6 VME/PCI Bridge, to define which of the twelve PCI interrupt request signals will propagate through to the VMEbus. This interrupt path is supported in all revisions of the V6M6.

Revision 2 V6M6 boards contain another six bit mask register which allows selected PCI INTA signals from each module to propagate onto a single, global interrupt signal, "MODINT/", which is supplied in common to all six module sites. This signal is separate from the normal PCI bus signals and can be used on a module to generate an interrupt to a processor or DSP on that module. This is intended to allow I/O modules to generate interrupts directly to processor or DSP modules on the same V6M6.

Revision 3 V6M6 boards extend this on-board interrupt router by separating the MODINT/ signals to each module and by allowing both INTA and INTB from each module to participate in the routing. Additional control registers are included to allow selected interrupt requests to propagate to specific module MODINT/ signals. This is provided to better support interruptibility of multiple processor or DSP modules on a V6M6.

3.5 TDM Subsystem

The TDM bus on the V6M6 is a communication channel completely separate from the PCI bus.

The TDM bus is intended to carry low latency isochronous (constant bit rate) data efficiently, without burdening the PCI bus. Typical usage for the TDM bus is to carry digitized audio from one module to another or between off-board audio sinks/sources and modules on the V6M6.

Data passing between resources on a V6M6 assembly is routed through four serial time-division multiplexed (TDM) buses. They are called buses A, B, C, and D. These TDM buses run synchronously, each with the same bit clock, time slot definition, frame duration and frame sync pulse.

A connection established through the TDM system appears to be a dedicated link between the resources participating in the connection. Connections may be point-to-point or point-to-multipoint. That is, a single resource sources the data in a connection and any desired number of resources can receive the data.

The TDM buses interconnect the six module sites, a front-panel TDM expansion connector labeled "J4", and, on V6M6 Rev 3 and higher, a baseboard interface to an SCSA bus on the VME P2 connector.

The SCSA bus interface is used to connect TDM data streams between VME boards via an SCSA backplane on the P2 connector. This bus is a standardized method of exchanging telephony data between boards from various vendors. It does not require front-panel cables, and so is appropriate for uses where straightforward board replacement is important. Up to twenty one boards in a VME backplane may be connected together through this bus. For Rev 2 V6M6 boards, the SCSA bus is accessible through a plug-in module, the IMSCSA, which may be installed in Module position B.

J4 is a 20-pin right angle header intended to mate with a multi-connector ribbon cable assembly. It may be used to directly connect the TDM buses of a group of CAC V6M6 and/or VME6U6 boards. The number of boards that can be tied together through a J4 ribbon depends on the TDM clock rate. Assemblies of two V6M6 and/or VME6U6s tied together via the J4 connector will operate at up to 16 MHz, three or four boards at up to 8 MHz and five or more boards at up to 4 MHz. Note that some module designs cannot support data rates higher than 8 MHz.

3.5.1 TDM Bus Logical Structure

Consecutive TDM bit clocks are grouped together into a timeslot. All timeslots on all four TDM buses must be the same length. This length is programmably selectable via the TDM controller as 8 bits, 16 bits or 32 bits. Normally, a timeslot carries a single data item across each TDM data bus. However, the TDM system allows the use of multiple timeslots on a TDM bus to carry larger data items. Four 8-bit timeslots, for instance, can carry a 32-bit word between resources. CAC refers to the technique of using multiple small timeslots to carry a larger data item as "slot continuation".

Timeslots are organized into frames. A frame consists of a programmably selectable, or externally determined, number of consecutive timeslots. The maximum frame size on CAC VME boards is 128 slots, and so the maximum number of actual data timeslots is 128 per bus times 4 buses, or 512 slots. The connection pattern used on the TDM buses is recycled with each frame; that is, each connection established over a TDM bus recurs at the frame rate.

A frame sync signal is included in the TDM subsystem to indicate the boundary between consecutive frames. This sync pulse can be generated by the TDM controller on a V6M6 board, by a plug-in module, by external circuits connected to the J4 expansion cable, or by the SCSA interface.

The V6M6 J4 expansion connector directly links together the TDM subsystems of multiple VME boards. When boards are linked, they share the same TDM clock, timeslot definition and framing. One board must be programmed as clock master to generate the TDM clock; the remaining boards are programmed as clock slaves. The selection of which board generates frame sync is independent of the clock master selection. That is, TDM clock and TDM sync can come from the same board or different boards, depending on the application.

3.5.2 TDM Data Validity

The V6M6 TDM system includes a "Validity" signal associated with each TDM data bus. That is, TDM bus "A" has an associated "A Valid" signal, bus "B" has a "B Valid", etc. These lines indicate whether the associated TDM data bus lines carry valid data during each timeslot. This information is used to implement conditional TDM transfers on some modules.

Connections established through the TDM system provide the opportunity to transmit a data item from source to destination(s) each time its assigned timeslot occurs. If the source, for example, is assigned to drive

TDM bus A and has data ready to transmit when its timeslot comes by, it drives the TDM A Valid line high at the beginning of the time slot. This indicates to the destination(s) that bus A should be read. If the source has no data to transmit, it drives A Valid low, indicating that the destination(s) for the connection should ignore the data on bus A.

The SCSA bus does not include the concept of timeslot validity. Therefore, any TDM connections involving the SCSA bus are assumed to always have valid data.

In addition, some module types (the DM2C31, for example) are not capable of generating conditional transfers. In such modules, the data source (a TI TMS320C31 DSP in this case) does not indicate whether its serial port is ready to transmit. Thus the TDM interface of the data source must always assume that it is ready, and always drives data on its assigned timeslot(s).

3.5.3 TDM Clocking

The number of bits in a timeslot and the duration of a frame are programmable in the TDM controller. The TDM clock itself may be derived from:

- A plug-in module, such as a telecom network interface device.
- The J4 TDM expansion connector
- The SCSA bus, when the V6M6 operates as an SCSA timing slave.
- A programmable division of a frequency synthesizer on the V6M6 baseboard. This synthesizer is factory set to 16.384 MHz; a utility program allows this frequency to be set according to the needs of the application.

Typical telephony applications set the TDM clock to 8.192 MHz in order to provide the maximum number of 8-bit timeslots (128 per TDM bus, times 4 buses, yielding 512 slots). This clock is typically derived from a telecom network interface such as a T1 or E1 interface module or from another board via the J4 or SCSA interfaces.

In most telephony applications, the frame length is fixed at 125 μ S, corresponding to an 8 KHz audio sampling rate. The number of timeslots per frame thus depends on the TDM clock rate chosen. For 2.048 MHz, the frame is 32 slots (128 timeslots total), for 4.096 MHz it is 64 slots (256 timeslots total), and for 8.192 MHz it is 128 slots (512 timeslots total).

The V6M6 generates a separately buffered TDM clock signal to each module and to baseboard resources concerned with the TDM system. These signals are supplied by low-skew drivers, with delay-balanced circuit board traces to each destination. Thus, the TDM clock signals at the six module sites have very little skew. Each clock must be terminated at its end-point with a series combination of a 62 ohm to 75 ohm resistor and a 100 pF to 220 pF capacitor to ground. The TDM clock trace on a module must be three inches or less in length, and the terminator must be within one inch of the end of the trace.

If the V6M6 is connected to the SCSA bus via an IMSCSA module or via the "S" option on Rev 3 and higher, a phase-locked loop associated with the SCSA interface forces the V6M6 TDMCLK signals to be properly phase aligned with the Local Bus Clock signal at the SCSA bus interface circuit (a VLSI SC4000 chip). This assures that setup and hold times associated with the SC4000 are satisfied.

The SC4000 may be programmed to become SCSA bus clock master. In this case, the TDM clock signal on the V6M6 board is derived from the frequency synthesizer on the baseboard or from a module with an external timing source, such as a T1 or an E1 interface. The SC4000 uses the output from its associated phase locked loop to produce the SCSA Bus Clock and a Local Bus Clock properly phase aligned to the TDM clock.

3.5.4 TDM Connection Control - The TDM Map RAM

Connectivity of the TDM buses on the V6M6 is directed by the content of a dual-port memory, called the TDMRAM. One port of the TDMRAM is loaded via the VME interface and, on V6M6 Rev 3 or higher, the PCI bus.

The TDM subsystem accesses the TDM RAM from the other port. It is addressed by a counter which resets to zero at the beginning of each new frame and increments with each new bit clock. An 8-bit control word is fetched from the TDMRAM each bit time. These words are not used on the V6M6 baseboard, but are broadcast to all six module sites. The modules, in turn, interpret these control words to determine their required TDM actions during the next TDM timeslot. Note that some modules (such as the IMSCSA and the DM4C51) promote their own TDM connection control, which is programmed separately.

Each timeslot has a sequence of 8 8-bit TDMRAM words (64 bits total) available to control the TDM subsystem. This 64-bit control word is staged in a holding register in each module and becomes active at the beginning of the next timeslot. That is, the control words fetched from

TDMRAM during timeslot "x" are used to control the TDM buses during timeslot "x+1" (modulo the number of timeslots in a frame).

Only 8 control words are required from the TDMRAM for each timeslot. If a slot size longer than 8 bits is selected, the TDMRAM addressing counter stops incrementing after 8 clocks in each timeslot, and resumes incrementing at the beginning of the next timeslot. Thus, exactly 8 locations in TDMRAM are used to control each timeslot regardless of the specified slot size.

The TDMRAM is divided into two banks. One bank actively controls the TDM subsystem, while the other is available for updating by the host processor through the VME interface (or through the PCI bus on Rev 3 and higher). Typically, each bank would contain a copy of the desired TDM interconnection map. When changes are made to the "Update" bank, they do not take effect until a "Bank Switch" command is issued to the TDMRAM controller. This approach is required in order to ensure that changes to the TDM interconnection map made in the TDMRAM take place in a consistent way, without the possibility of executing partially updated interconnection commands.

3.5.5 TDM Control Word

The table below shows the interpretation of the 64 TDM control bits associated with a TDM system time slot. It is divided into three main portions.

- The first portion (words 0 and 1) defines which module resources drive data on each TDM bus during the following time slot.
- The second portion (words 5, 6 and 7) define TDM destination control, such as from which TDM bus the module is to receive data
- The third portion (words 2, 3, and 4) are used to define some control action related to TDM source or destination activity.

Together these portions describe up to four connections or control actions to be made during the following time slot. Note that a connection can have only one source, but may have multiple destinations.

Word	Bits 7:4	Bits 3:0
0	Bus A Source	Bus B Source
1	Bus C Source	Bus D Source
2	Mod E Control	Mod F Control
3	Mod C Control	Mod D Control
4	Mod A Control	Mod B Control
5	Mod E Dest Control	Mod F Dest Control
6	Mod C Dest Control	Mod D Dest Control
7	Mod A Dest Control	Mod B Dest Control

Table x.a - TDM Control Words

3.5.6 TDM Source Control

Each of the six module sites interprets all four TDM bus source nibbles and two module-specific control nibbles, labeled "Control" and "Dest Control". Some module types only need the Dest Control nibble, while others use both.

The TDM bus source nibbles are shown below. Note that each module may interpret two different source codes, primary and secondary.

4-bit Code	Source Connection
0x0 - 0x5	Mod <i>N</i> (<i>Pri</i>) Drives TDM
0x6 - 0xB	Mod <i>N-6</i> (<i>Sec</i>) Drives TDM
0xC	Src <i>CONTINUATION</i>
0xD - 0xE	Reserved
0xF	No Src Specified

Table x.b - TDM Bus Source Codes

When a module is assigned as a source on a TDM bus time slot, its TDM interface logic takes data serially from whatever source is appropriate for the module and drives it onto the designated TDM bus. Definitions of the TDM source functions for specific module types are provided in reference manual sections describing those modules.

As an example, the DM2C31 module contains two TI TMS320C31 DSP chips, numbered 0 and 1. The TDM primary source code refers to DSP 0 and the secondary source code refers to DSP 1. When a DSP is directed to be the source on a TDM time slot, interface logic on the DM2C31 generates an *FSX* pulse to that DSP's serial port, together with a burst of clock pulses to the *CLKX* pin. This causes the DSP to send serial output data onto its *DX* pin, which the TDM interface logic then drives onto the designated TDM bus.

The "Continuation" code causes the previously selected TDM source to continue transmitting data onto the designated TDM bus without generating a new "Start" command in the associated resource. This allows for data words larger than the basic TDM slot size to be transferred. Most module types do not use the Continuation code. A

DM2C31 module, for instance, has separate logic in its TDM interface to allow multiple, possibly noncontiguous, timeslots to be aggregated into larger word sizes, and does not need the Continuation code. The PM3081 and PM4600 modules do implement Continuation codes, as described in their respective reference manual sections.

3.5.7 TDM Destination Control

The encoding of TDM destinations is highly dependent on the specific module design. Generally, a V6M6 module has two 4-bit nibbles available to it to specify TDM destinations and, for some module types, additional TDM-related controls. These are labeled as *Mod n Dest Control* and *Mod n Control* in Table x.a above. Available TDM destination codes and control actions are defined in the reference manual sections describing each module.

As an example, the PM3081 and PM4600 modules implement the following *Mod n Dest Control* codes. They do not use the *Mod n Control* nibble:

<i>Dest Ctl Code</i>	Dest Connection	New Item
0 - 3	Receives TDM <i>N</i>	Yes
4 - 7	<i>CONTINUATION</i> Recv TDM <i>N-4</i>	No
8 - 11	<i>CONDITIONAL</i> Recv TDM <i>N-8</i>	if TDM VALID
0xF	No Dest Connection	No

Table x.c - PM3081/PM4600 Destination Codes

3.5.8 TDM Bus Timing

Overall timing of the TDM subsystem is controlled by a clock (TDMCLK) and two synchronizing signals (TDMSLOT and TDMFRAME). TDMSLOT and TDMFRAME are synchronous with TDMCLK, are sampled on the rising edge of TDMCLK, are active high, and last for exactly one TDMCLK period. TDMSLOT indicates the beginning of a new timeslot. TDMFRAME indicates the beginning of a new frame of timeslots. An additional synchronizing signal (TDMSF) is available to the TDM resources to indicate "superframe" alignment.

TDMFRAME and TDMSF are directly available on the J4 expansion connector. TDMSLOT is locally generated by the TDM controller on each V6M6 board. TDMCLK is a distinct signal on each V6M6 board. For V6M6 boards that are programmed to participate on the J4 expansion ribbon, TDMCLK is a buffered version of the J4 signal EXPANSION_CLK. The delay from EXPANSION_CLK to TDMCLK is a maximum of 5 nsec

The TDM subsystem relies heavily on pipelining. This means that the TDMSLOT and TDMFRAME signals do not line up directly with the actual data bits carried on the TDM data buses, but rather appear somewhat earlier in time. Specifically, TDMSLOT appears three clocks before the first bit of a new timeslot on the TDM data buses. TDMFRAME appears one full timeslot plus 5 clocks before the first bit of a new frame on the TDM data bus.

The effect of the pipelining for external devices which generate TDMFRAME on the J4 expansion cable is that TDMFRAME must be asserted 13 clocks before the first bit of frame data if the slot length is 8 bits, 21 clocks for 16-bit slots and 37 clocks for 32-bit slots.

On an IMSCSA module or on V6M6 "S" option boards, logic adapts the SC4000 chip's SCSA and Local Bus timing to proper TDM bus timing. The SCSA bus has no signals corresponding to TDMSLOT or TDMSF, and its SYNC signal is aligned differently from the V6M6's.

3.5.9 Superframes and TDMSF

Superframes are commonly encountered in T1 applications, where two standards are used. The first, and older, is "D4" or simply "Superframe" or SF. In SF, a sequence of 12 T1 frames, lasting 125 μ S each, is considered a superframe. Superframe alignment identifies "robbed bit" signaling frames (every 6th and 12th frame); the actual alignment is determined by the specific pattern of the framing bits in consecutive T1 frames. A newer standard, called Extended Superframe or ESF, groups 24 consecutive T1 frames per superframe and allows every other framing bit to carry a maintenance channel called the Facility Data Link (FDL).

E1 signals may also have superframes of two types, called CAS and CRC4. These are both 16-frame superframes, and either or both may be active at the same time, not necessarily aligned with each other. For more discussion of E1 frames and superframes, see the IM2E1 Mini-PCI Module section.

A signal called TDMSF in the TDM subsystem identifies the superframe repetition rate of the TDM data. TDMSF is a pulse which lasts for one full

frame. It may be selected as an interrupt source to all processor and DSP modules on the V6M6. On DSP modules, it is also connected to a directly testable input pin on each DSP.

TDMSF cannot, in general, identify which TDM frame is the boundary of a superframe, because multiple T1 or E1 signals may be present on the TDM buses. The T1 or E1 signals are forced into alignment with TDMFRAME by the network interface module or by the SCSA bus connection, but forced superframe alignment on received signals is not possible without unacceptable delays in those input signals.

The actual superframe alignment of a specific T1 or E1 signal is not normally important to V6M6 modules. Indeed, it is generally not possible even to identify the alignment of T1/E1 signals received over the SCSA bus.

TDMSF may be generated by the TDM controller on a V6M6, by a module board or by external logic through the J4 interface. If a TDM controller is programmed to source TDMSF, it generates it by a division of the TDM frame rate. The divisor is specified as an integer between 2 and 31, so superframe lengths of 2 to 31 frames can be programmably specified. When sourced by the TDM controller, the bit-level timing of TDMSF's transitions are positioned in the middle of the first time slot of the TDM frame.

TDMSF is an especially important signal for DSP applications that send or receive more than one time slot of data per frame in any given DSP. In those situations, the DSP program needs somehow to align itself with the TDM framing so that it knows which time slot is associated with each inbound or outbound data item. TDMSF provides a mechanism to achieve this alignment. Its transition from low to high can be used to establish a reference point in the inbound and/or outbound TDM data streams within each DSP. The DSP programs are responsible for maintaining alignment between TDMSF transitions. In general, the alignment need only be determined once, prior to beginning data transfers.

Processor modules, such as the PM3081 and PM4600, do not need to use this mechanism. In these modules, each item in the TDM data stream is explicitly tagged with the timeslot number corresponding to that item. See the PM3081 and PM4600 Mini-PCI Module Reference Manuals for more detail.

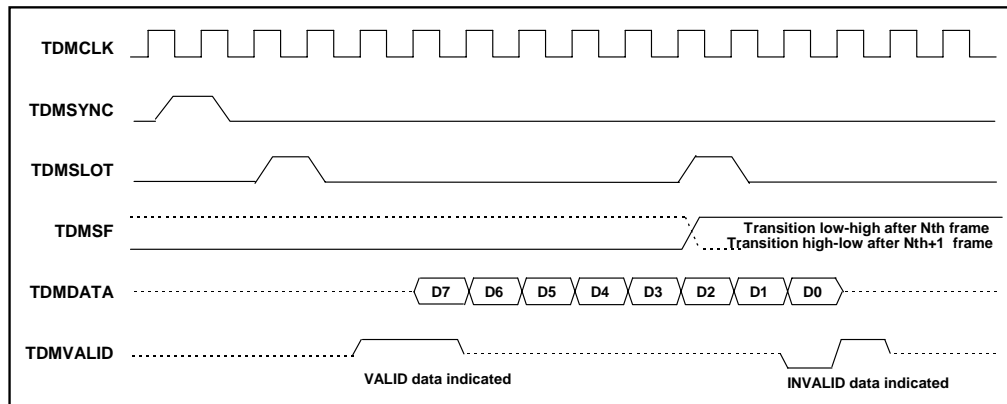


Figure 3.4 - TDM Subsystem Timing

3.5.10 TDM Subsystem Controller

(documentation not yet available)

3.6 VME/PCI Bridge

(documentation not yet available)

3.7 Software Support

The software for the V6M6 consists of the following categories:

- Unix Device Driver
- Host Application Library
- Host Support Programs and Diagnostics
- MIPS Kernel and Application Library
- MIPS Kernel Host Server
- pSOS Support

The software currently supports use on SunOS 4.1.x (sometimes referred to as Solaris 1.x) and Solaris 2.x (sometimes referred to as SunOS 5.x).

Except for the device drivers, the software is platform independent at the source code level. The device drivers and, in some instances, special "stub" modules in the host library handle platform and operating system differences.

3.7.1 Unix Device Driver

Versions of the device driver currently exist for SunOS 4.1.x and Solaris 2.x. The device driver provides access to the resources for host applications, configures the VME interface and handles interrupts from the board (passing software signals to the appropriate host processes).

Access to the board's resources is provided via entries in the /dev directory named `pciNX`, where N is an integer board number specifying a particular board and X is a lower case letter specifying the resources on the board as follows:

<code>pci0a-f</code>	refer to individual module resources.
<code>pci0g,h</code>	provide additional host access to global memory.
<code>pci0q</code>	is the VME Interface resource, which is used for board initialization, and special control operations.
<code>pci0t</code>	is the TDM subsystem resource used to control the TDM serial I/O configuration and operation. This resource is also used to access the microcontroller.

The primary data transfer method is through memory-mapped I/O. Some operations are handled by the device driver using the `ioctl` interface. Although MIPS processor modules can access other boards on the VMEbus, the V6M6 does not support DMA transfer with the host system.

3.7.2 V6M6 Host Application Library

The V6M6 Host Application C Library provides most of the functionality required to access the module resources and perform I/O such as loading and running programs, uploading and downloading data, and controlling the TDM interface.

The following sections discuss general purpose and baseboard-specific functions. Functions to control operations for specific modules are discussed in chapters 4 through 8 which describe the Mini-PCI Modules.

3.7.2.1 Host Device Access and Data Transfer Functions

(documentation not yet available)

3.7.2.2 TDM Subsystem Control Functions

(documentation not yet available)

3.7.3 Host Support Programs

The V6M6 software package includes several utilities run on the host. These perform important operations such as board initialization and updating flash memory. These utilities include:

- pciinit - V6M6 initialization
- pciinfo - Obtains information about V6M6 boards
- pciflashup - Updates contents of flash memory.
- pcidmexec - Loads and executes programs on DSP modules
- pcitdmcfg- Configures TDM subsystem and connection map

To access the utilities and their online manuals, see Section 2.3.2 Programmer and User Setup p 2-16.