Model V165-xBx1

Digital Signal Processor

INSTRUCTION MANUAL

March 14, 2001

Contents

FIGURES

TABLES

Warranty
SCK:rem(WP)

Schematic Drawing # 292144-C-6656
Schematic Drawing # 232336-C-6661

Digital Signal Processor

A TI-based DSP with 40 MFLOP processing power

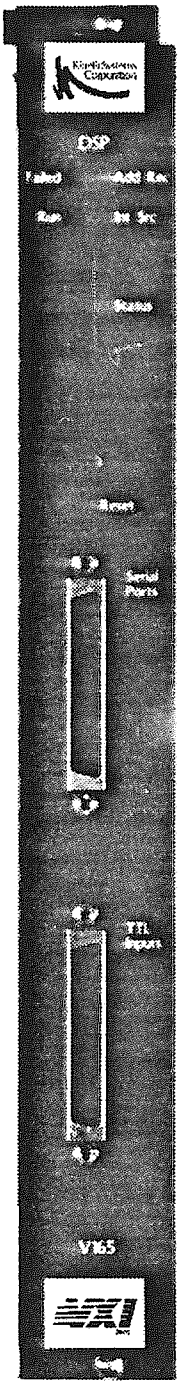
V165

Features

- **Performs FIR & IIR filtering**
- **Computes FFTs**
- **Performs signal averaging**
- **Executes limit checking**
- **Converts to engineering units**
- **Executes at 40 MFLOPS**
- **Performs 32/40-bit floating point operations**
- **Includes 1 to 16 Mbyte DRAM**
- **Has DMA capability**
- **Options for Digi-bus™ input and output data paths**

Typical Applications

- **Wind tunnel data acquisition**
- **Aerospace tests**
- **Automotive tests**
- **Acoustic tests**
- **General-purpose signal analysis**
- **High-speed control loops**



General Description *(Product specifications and descriptions subject to change without notice.)*

The V165 is a single-width, C-size, register-based, VXIbus module that provides realtime processing for data collected from other modules within the VXI chassis. The processing power for this module is provided by the Texas Instruments TMS320C30 ('C30) Digital Signal Processor (DSP). This DSP provides 40 MFLOP operation and its 32/40-bit floating-point arithmetic capabilities simplify the development of applications, eliminating the effects of scaling, normalization, and overflow. Implemented with low power CMOS, the 'C30 also provides 16- and 24-bit integer operations, IEEE floating-point conversion capability, a bit reversal addressing mode, eight 40-bit accumulators, two 32-bit timers, and a 64 x 32-bit instruction cache. It can implement FIR filters at 50 ns per tap and compute a 1024-point complex FFT in 3.04 ms.

The 'C30 has two blocks of 1K x 32 single-cycle, dual-access, on-chip RAM. A range of 1 to 16 Mbytes of dual-ported RAM is also provided on the module. All dual-access memory is addressable from the VXIbus as well as by the DSP. This allows application programs and data to be downloaded from the host computer and to dynamically change as conditions require. Instructions and data can arbitrarily reside in either the on-chip or off-chip memory.

Data to be manipulated by the DSP may be obtained from various sources and stored in various destinations. The data may be read or written by the V165 using Direct Memory Access (DMA). The V165 can become a master on the VXIbus and transfer data to and from other VXIbus modules within the chassis. Data may also be sent into and out of the V165 using the VXI Local Bus. For the V165-Bxy1 option, this module supports a KineticSystems-developed Local Bus interface and protocol called Digi-bus™. Digi-bus provides the mechanism to transfer digital data between adjacent modules at high speed without degrading the performance of the VXI system bus or introducing latency from "slow responding" VXIbus modules within the chassis.

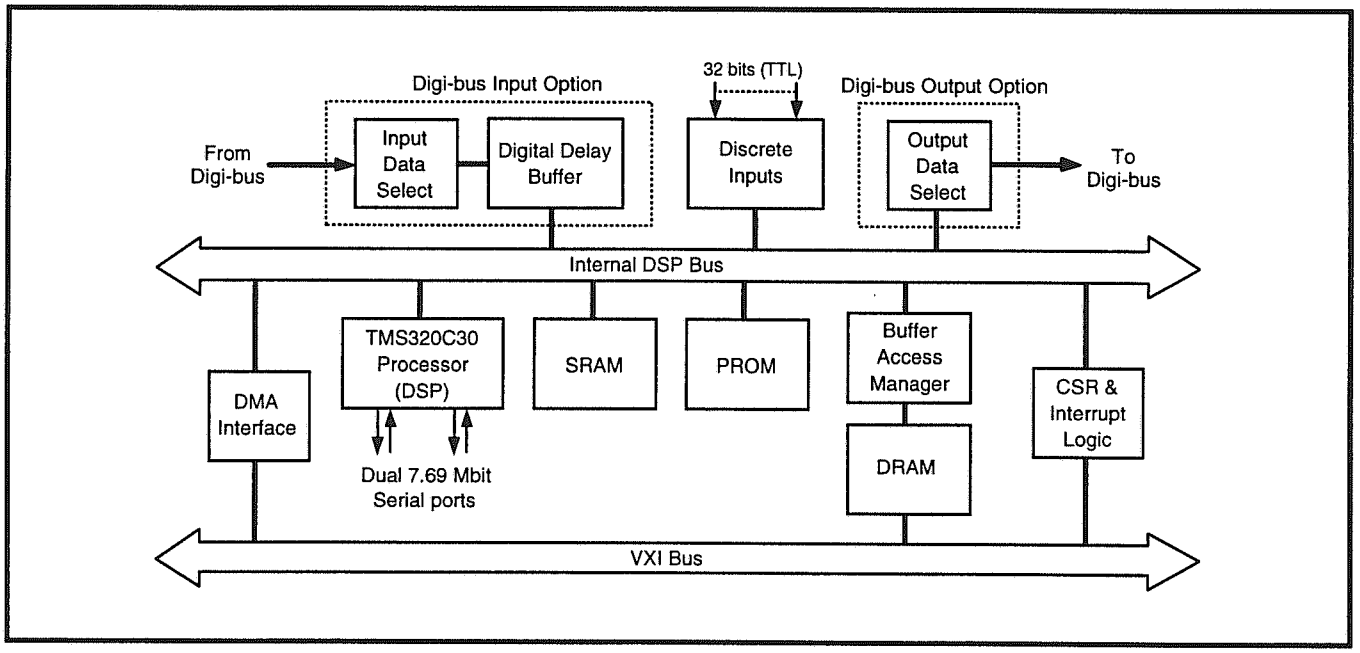
Digi-bus data enters the V165 from the module on its right (the next higher numbered slot) and is passed to a selection mechanism on the V165 Digi-bus mezzanine board. The V165 can be programmed to accept all the incoming data or only selected subsets. The selected data is then available to the DSP through a 32 kword buffer. Circuitry on the mezzanine board allows the DSP to send data to the module on its left (the next lower numbered slot) by writing the word to a register on the mezzanine board.

The V165 can synchronize with other modules in the chassis by the use of its VXIbus triggers and interrupts. The V165 can be programmed to generate any number of simultaneous VXI triggers using the Synchronous Trigger protocol and to generate a VXI interrupt on any one of the eight VXI interrupt sources. The V165 can also be programmed to interrupt the DSP on the occurrence of a selected VXI trigger or when a VXI request is made with a write to a Generate DSP Interrupt register in A32 space on the V165. In addition, the 'C30 DSP provides two internal, 32-bit timers which can be programmed for periodic interrupts.

A 32-bit digital input register, which is directly readable by the DSP, can be written from the front-panel connector. Reduced data and results from the DSP's operation can be sent either to other VXI modules via DMA or to other Digi-bus equipped, KineticSystems VXI modules downstream from the V165. Also available at the front connector are two DSP serial ports capable of 7.69 Mbit/s operation. These ports can be used to transmit or receive serial data between this module and an external source, such as another V165 DSP module, for multiprocessor communication.

The V165 supports both static and dynamic configuration. Access to the DSP and its associated memory is through memory locations selected by the Offset Register within the VXIbus Configuration Register set. Access to these registers and memory is accomplished within A32/A16 address space using D32/D16 transfers.

V165 Block Diagram



Cross-development Software and Utilities

The Texas Instruments TMS320 Floating-Point DSP Optimizing C -compiler, assembler, linker, emulator, and simulator are available to support application programming. This software is available for the PC/MS-DOS, VAX/VMS, and SUN-3/UNIX platforms. We also support Momentum Data Systems' QEDesign 1000 Digital Filter Design package. This package allows the user to design FIR and IIR digital filters for the V165. This software is available for the PC under Windows as well as for SUN workstations.

KineticSystems has developed a support library for the V165 DSP which offers a high-level interface to the module's hardware features. The library functions include DMA read and write, Digi-bus acceptor and source, VXI trigger interrupt service, VXI trigger and interrupt source, and timer interrupt service. Other V165 hardware features, such as the front-panel TTL inputs and the 8-bit LED status register, also have interface functions in the support library. The library distribution also includes a library header file with prototypes for all the functions, working code examples for each hardware group, and utilities which download the user's 'C30 application code to the V165 and execute it.

When used in conjunction with the Texas Instruments TMS320 Floating-Point DSP Optimizing C-Compiler package (Model AB11-DPA1), the support library can simplify the development of DSP applications for the V165. Programmers can establish the data path into and out of the V165 with DMA or Digi-bus function calls, leaving them more time to implement signal processing algorithms. While C should be the language of choice, the compiler package includes a 'C30 assembler (and other programmer's tools such as a linker and archiver) when the application must execute as quickly as possible. Both the compiler and the library support Interrupt Service Routines (ISRs) are written in C. The compiler generates entry and exit code that preserves the machine environment. The library simplifies the use of VXI and timer interrupts by setting up the interrupt vector and the interrupt masks, leaving the programmer to concentrate on the application. The distribution disk contains source and object code for VXI trigger and DSP timer ISR examples to facilitate development.

Please contact KineticSystems or your local representative for additional information about these software products.

Item	Specification
DSP	
Cycle time	50 ns
Operations per second	40 MFLOPS
Instructions per second	20 MIPS
Instruction cache	64 x 32
On-chip RAM	Two blocks of 1K x 32
Timers	Two, 32 bits each
DSP Support Circuitry	
Dual-ported dynamic RAM	1 to 16 Mbyte, 80 ns pipelined-access
Static RAM	8 kbyte to 128 kbyte, 0 wait state; optional 128 kbyte, 1 wait state
ROM	32 kbyte to 128 kbyte, 1 wait state
DMA	Supports 8, 16, and 32-bit data transfers
Digi-bus input port	32 kbyte x 16-bit sample buffer
Digi-bus output port	16-bit samples configurable via DSP
Discrete Digital Inputs	
Number of inputs	32
Input signal level	TTL
Input signal logic level	High true (0 V read by DSP as a logic "0")
Input signal termination	Pulled up to +5 V through 4.7 kΩ resistors
Serial Ports	
Number of ports	2
Port direction	Bi-directional
Bits per serial word	8, 16, 24, or 32
Maximum serial rate	7.69 Mbit/s
Connector Type	50P High Density
Power Requirements	
+5 V	6.8 A
-5.2 V	125 mA
-2 V	125 mA
Environmental and Mechanical	
Temperature range	
Operational	0°C to +50°C
Storage	-25°C to +75°C
Relative humidity	0 to 85%, non-condensing to +40°C
Cooling requirements	10 CFM
Dimensions	340 mm x 233.35 mm x 30.48 mm (C-size VXIbus)
Front-panel potential	Chassis ground

Ordering Information

Model V165-wxy1 Digital Signal Processor

w: Option cards

A = no options installed
B = Digi-bus™ Local Bus option

x: DRAM Size

A = 1 Mbyte DRAM
B = 4 Mbyte DRAM
C = 8 Mbyte DRAM
D = 16 Mbyte DRAM

y: SRAM Size

A = 8 kbyte SRAM
B = 32 kbyte SRAM
C = 128 kbyte SRAM
D = 256 kbyte SRAM

Model V165-0001	Digi-bus Factory Upgrade

Related Products

Model 5819-Bxyz	Cable—50S High Density to Unterminated
Model 5819-Dxyz	Cable—50S High Density to 50S Amphenol Ribbon
Model 5819-Fxyz	Cable—50S High Density to 50P High Density
Model 5819-Gxyz	Cable—50S High Density to 50S High Density
Model AB11-DPA1	V165 TI Cross-development Package for MS-DOS
Model AB12-DPA1	V165 Full Development Package for MS-DOS (Cross-development routines plus KineticSystems Software Support Library)

Please contact the factory for information on software packages for other operating systems.

Model V110	4 to 128 Megabyte Memory
Model V207	16-bit, 500,000 Sample/second ADC Subsystem
Model V208	16-bit, 100,000 Sample/second ADC Subsystem
Model V285	8 or 16-channel, 16-bit, 500 kHz DAC/Waveform Generator
Model V387	128-channel Discrete Input/Output
Model V765	Rack-mount Termination Panel

UNPACKING AND INSTALLATION

The Model V165 is shipped in an anti-static bag within a styrofoam packing container. Carefully remove the module from its static-proof bag and prepare to set the various options to conform to the operating environment.

Module Insertion

The V165 is a C-sized, single width VXI*bus* module. It requires 5700 milliamperes of +5 volt power, 150 milliamperes of -5.2 volt power, 150 milliamperes of -2 volt power, and 10 cubic feet per minute of air flow to maintain stability. Except for Slot 0, it can be mounted in any unoccupied slot in a C-sized VXI*bus* mainframe.

CAUTION: TURN MAINFRAME POWER OFF WHEN INSERTING OR REMOVING A VXI MODULE

WARNING: REMEMBER TO REMOVE THE INTERRUPT ACKNOWLEDGE DAISY-CHAIN JUMPERS AND BUS GRANT-IN TO BUS GRANT-OUT JUMPERS PRIOR TO INSERTING THIS MODULE IN THE VXI BACKPLANE

To insure proper interrupt acknowledge cycles from the V165 module, the daisy-chain Interrupt Acknowledge jumper must be removed before the module is installed in a slot. Conversely, daisy-chain jumpers must be installed in any empty slot between the V165 and the Slot 0 Controller. Since the V165 may execute DMA operations to other modules within the chassis, the Bus Grant jumpers must be removed from the slot in which the V165 is installed. Ensure that the Bus Grant chain is continuous from the slot 0 controller to the V165.

FRONT PANEL INFORMATION

LEDs

Add Rec	The Addressed Received LED is illuminated when the V165 registers are being accessed and when the V165 acknowledges an interrupt.
Failed	The Failed LED is illuminated when the V165 has failed its self-test.
Run	The Run LED is illuminated as long as the DSP is running (not in the RESET state).
Int Src	The Interrupt Source LED is illuminated as long as the V165 has an interrupt source pending. The interrupt source indicates that a condition exists for generating an interrupt.
Status	The eight status LEDs are driven from the output of a register accessible by the DSP. These LEDs are used to display error conditions found during the power-on self-test diagnostics. The user also has access to these LEDs under programmed I/O from the DSP.

Switches

The V165 has only one front-panel switch which is used to reset the DSP and its associated circuitry. The switch is recessed into the module to prevent accidental activation. A strap option located on the module may be configured to enable or disable the operation of this switch. Please refer to the Strap Options section of this manual for further information.

Connectors

The V165 has two fifty-pin male connectors on the front panel of the module. The connector towards the bottom of the module, labeled TTL Inputs, routes 32 digital input lines to a register that the DSP may access. These digital inputs have input protection circuitry to prevent any damage to the module in the case an input is shorted to a high voltage source. The second connector on the V165, labeled Serial Ports, contains the serial port signals from the DSP. This connector also contains two general purpose I/O bits of the DSP. Along with these signals are also two timer/counter bits to/from the DSP.

STRAP OPTIONS AND SWITCHES

Before the V165 is installed in the VXI chassis, several user selectable options must be configured. The following indicates the various options and their default configurations as set by the factory.

PARAMETER	DEFAULT CONFIGURATION
Logical Address	255
Bus Request Level	3
DSP Reset Vector Address	810000 Hex
Front Panel Reset	Enabled

There are other straps located on the module, but they MUST remain in the positions as configured at the factory. The only user configurable straps/switches are those listed above.

Logical Address Switches

The V165 represents one of the 255 devices permitted in a VXI*bus* system. (Logical Address 0 is reserved for the Slot 0 controller device.) The module is shipped from the factory with its address set for Logical Address 255. This address can be shared by multiple devices in a system that supports dynamic configuration. If the V165 is to be used in a system that does not support dynamic configuration, or in a system where static configuration of the module is desired, the Logical Address must be manually established. This is accomplished by manipulating eight rocker switches located under the access hole in the module's right-side ground shield.

Model V165-xBx1

The eight switches represent a binary combination of numbers that range from zero to 255. Use a scribe or other appropriate instrument to set the Logical Address to the desired value.

The bit pattern for the A16 base address is shown below:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	LA128	LA64	LA32	LA16	LA8	LA4	LA2	LA1	0	0	0	0	0	0

Bits 15 and 14 are set to one (VXI defined).
Bits 13 through 6 are user selectable via the address switches LA128-LA1.
Bits 5 through zero are set to "0" to indicate a the start of a block of 64 bytes.

The location of these switches is shown in the diagram in Appendix A.

Setting a switch to the on (closed) position causes the corresponding bit to be a logical 0.
Setting a switch to the off (open) position causes the corresponding bit to be a logical 1.

Assume that the user wishes to allow the V165 to be dynamically configured. To allow dynamic configuration, the V165 must be set to logical address 255 (0FF Hex). The following shows the bit pattern for this logical address.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

To set the V165 for logical address 255 (0FF Hex) place the switches LA128 through LA1 in the off (open) position.

As a second example, assume it is desired to statically configure the V165 for a logical address of 103 (67 Hex). The following diagram shows the bit pattern for this logical address.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	0	1	1	0	0	1	1	1	0	0	0	0	0	0

To set the V165 for logical address 103 (67 Hex), place switches LA128, LA16, LA8 in the (closed) position; and place switches LA64, LA32, LA4, LA2 and LA1 in the off (open) position.

Bus Request Level Straps

The V165 can execute Direct Memory Accesses (DMA) under control of the DSP. Since the V165 implements DMA, the user must select at which level the V165 makes its bus request. The V165 is configured at the factory for Bus Request Level 3.

Model V165-xBx1

The diagram in Appendix A shows the location of the Bus Request Level straps. The following chart shows the request levels obtained by the various combination of the straps.

BR1	BR0	Bus Request Level
0	0	Level 0
0	1	Level 1
1	0	Level 2
1	1	Level 3

Make sure that all the Bus Grant jumpers for the slot in which the V165 is installed are removed.

DSP Reset Vector Address Straps

There are three straps on the V165 which allow the user to select the reset vector that the DSP uses when a reset condition occurs. Four of these vectors are located in the address space of the PROM and the other four are located in the address space of the dynamic memory (DRAM). The diagram in Appendix A shows the location of the DSP Start Reset Vector Start Address straps. The following shows the reset vectors obtained by the various combinations of the straps.

BA2	BA1	BA0	DSP Reset Vector Address
0	0	0	810000 Hex
0	0	1	810002 Hex
0	1	0	810004 Hex
0	1	1	810006 Hex
1	0	0	100000 Hex
1	0	1	100002 Hex
1	1	0	100004 Hex
1	1	1	100006 Hex

Note: If it is desired to allow the Reset Vector Address to be altered by the VXI accessible DSP Control Register, this reset vector should be set to 810000 Hex. Indeterminate errors may occur if these straps are set to any other value and the DSP Control Register is written to through VXI.

Front-Panel Reset Strap

The front-panel reset switch may be optionally enabled or disabled. The switch is enabled as configured at the factory. Refer to Appendix A for the location of the RESET Enable/Disable strap.

To disable the front-panel reset switch, place the strap in the D/RST position.
To enable the front-panel reset switch, place the strap in the E/RST position.

PROGRAMMING INFORMATION

VMEbus/VXIbus Addressing

Of the defined VXI*bus* Configuration Registers, the V165 implements those required for extended register-based devices. The V165 also contains a set of Operational Registers to monitor and control the functional aspects of the device. Both register sets are described in this section.

Access to the Configuration Registers for all VXI*bus* modules is available through the VMEbus short address space. The register addresses are located in the upper 16 kilobytes of the A16 address range ($C000_{16}$ to $FFFF_{16}$). The setting of the Logical Address switch, or the contents of the Logical Address Register (see below) are mapped into Address lines A6 through A13, thereby establishing a base address for the module somewhere in the range of $C000_{16}$ to $FFFF_{16}$.

VXIbus Configuration Registers

Configuration Registers are required by the VXI*bus* specification so that the appropriate levels of system configuration can be accomplished. The Configuration Registers in the V165 are offset from the base address. **Note: the V165 only responds to these addresses if the Short Nonpriviledged Access (29_{16}) or Short Supervisory Access ($2D_{16}$) Address Modifier Codes are set for the backplane bus cycle.** Table 1 shows the applicable Configuration Registers present in the V165, their offset from the base (Logical) address, and their Read/Write capabilities.

TABLE 1 - Configuration Registers Configuration (A16) Space

A16 Offset	Read/Write Capability	Register Name
00_{16}	Read/Write	ID/Logical Address Register
02_{16}	Read Only	Device Type Register
04_{16}	Read/Write	Status/Control Register
06_{16}	Read/Write	Offset Register
08_{16}	Read Only	Attribute Register
$0A_{16}$	Read Only	Serial Number High Register
$0C_{16}$	Read Only	Serial Number Low Register
$0E_{16}$	Read Only	Version Number Register
10_{16} - 19_{16}	Read Only	Reserved
$1A_{16}$	Read Only	Interrupt Status Register

A16 Offset	Read/Write Capability	Register Name
$1C_{16}$	Read/Write	Interrupt Control Register
$1E_{16}$	Read Only	Subclass Register
20_{16}	Read Only	Suffix High Register
22_{16}	Read Only	Suffix Low Register
24_{16} - $3F_{16}$	Read Only	Reserved

ID/Logical Address Register

The ID/Logical Address register, which is located at offset 0 from the logical address base, serves two functions, depending on the direction of the VME transfer. When executing a read operation to this register, the data returned indicates the Device Class, the Address Space in which the Operational Registers of the V165 reside, and the Manufacturer's Identification. A write operation to this register is only executed during a dynamic configuration sequence. During the configuration sequence, the Resource Manager assigns a logical address to the V165 by writing the logical address into the lower 8-bits of this register. The format and bit assignments for the ID/Logical Address register are as shown below. Since this register has write-only and read-only bits, two bit patterns are shown.

On READ transactions:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	0	1	1	1	1	1	0	0	1	0	1	0	0	1

Bit(s)	Mnemonic	Meaning
15, 14	Device Class	This is an Extended Register-Based device.
13, 12	Address Space Requirement	This module requires the use of A16/A32 address space.
11 - 00	Manufacturer's ID	3881 ($F29_{16}$) for KineticSystems.

On WRITE transactions:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	LA128	LA64	LA32	LA16	LA8	LA4	LA2	LA1

For WRITE transactions, bits fifteen through eight are not used. A write to these bits has no effect on the V165. In Dynamically configured systems (i.e., the Logical Address switches were set to a value of 255), bits seven through zero are written with the new Logical Address value. This write operation is typically executed by a Resource Manager.

Model V165-xBx1

Device Type Register

The Device Type Register, which is located at offset 2 from the logical base address, is a read-only register which reflects the Model Code of the V165 and the memory requirements for the V165 Operational Registers. The memory requirement field is used by the Resource Manager to allocate physical address space to the devices located in the VXI chassis. A write operation to this register has no effect on the contents of this register.

The format and bit assignments for the Device Type register are as follows:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1

On READ transactions:

Bit(s)	Mnemonic	Meaning
15 - 12	Required Memory	The V165 requires 8 Megabytes of additional memory space. This value varies based on the DRAM option.
11 - 0	Model code	Identifies this device as Model V165 (165_{16}).

Status/Control Register

The Status/Control Register, which is located at offset 4 from the logical base address, contains read-only, write-only, and write/read bits. The following describes the bits defined for write and read operations.

The following shows the register layout and the bit pattern for read accesses to the Status/Control register.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A32 ACT	MODID	1	1	1	1	1	1	1	1	1	1	RDY	PASS	0	RST

Bit(s)	Mnemonic	Meaning
15	A32 Active	This bit is read as a one when the V165 is enabled for accesses in A32 address space. This bit is reset to zero on power-up or the assertion of SYSRESET*
14	MODID*	This bit is set to a one if the module is <u>not</u> selected with the MODID line on P2. A zero indicates that the device is selected by a high state on the P2 MODID line.

Model V165-xBx1

13 - 04	Not used	These bits are not used and read as ones.
03	Ready	A one indicates the successful completion of register initialization.
02	Passed	A one indicates that the V165 has failed or is executing its self-test.
01	Not used	This bit is not used and read as zero.
00	RST	When this bit is reset as a one, the V165 is in a RESET state.

The following shows the register layout and the bit pattern for write accesses to the Status/Control register.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A32 ACT	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	SFINH	RST

Bit(s)	Mnemonic	Meaning
15	A32 Active	This bit must be set to a one to enable the V165 to be accessed in A32 address space. This bit is reset to zero on power-up or the assertion of SYSRESET*
14 - 02	Not used	These bits are not used for write operations.
01	Sysfail Inb.	Setting this bit to a one inhibits the V165 from driving the VME SYSFAIL* signal.
00	Reset	Writing this bit as a one resets the V165. After the bit is set back to a zero, the V165 executes its self-test procedure.

Offset Register

The Offset register, which is located at offset 6 from the logical base address, is used for specifying the base address of the V165 operational registers in A32 space. Since the V165 occupies 8 Megabytes of address space, only the most significant 9 bits of this register are used. This register is a 16-bit write/read register with the following bit assignments:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A31	A30	A29	A28	A27	A26	A25	A24	A23	0	0	0	0	0	0	0

Model V165-xBx1

Bit(s)	Mnemonic	Meaning
15 - 07	A31 - A23	These write/read bits are used for defining the base address of the V165 Operational Registers.
06 - 00	Not used	These bits are not used but should be written as zeros.

After SYSRESET*, and prior to self-test, all bits are set to zero. Writing to this register is executed by the Resource Manager during the configuration process. The user may then examine this register to determine the base address of the V165 operational registers. After this register is written with the desired base address, the A32 ENABLE bit in the Status/Control register must be set to a one before the V165 Operational Registers may be accessed.

Attribute Register

The Attribute register, which is located at offset 8 from the logical base address, is a 16-bit read-only register which defines the interrupting capabilities of the V165. The V165 is an interrupter with interrupt status functionality, but does not implement an interrupt handler.

The format and bit assignments for the Attribute register are as follows:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	1	1	1	1	1	IR*	IH*	IS*

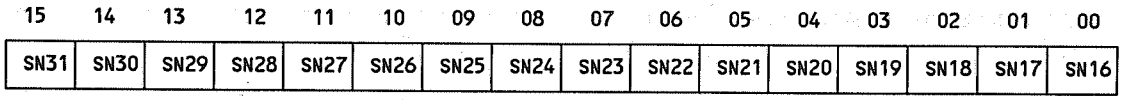
Bit(s)	Mnemonic	Meaning
15 - 03	Not used	A write to these bits has no effect. A read of these bits will return all ones.
02	IR*	The V165 has interrupter control capabilities and returns this bit as a zero.
01	IH*	The V165 does not have interrupt handling capability and returns this bit as a one.
00	IS*	The V165 has interrupt status capability and returns this bit as a zero.

Serial Number High Register

The Serial Number High register, which is located at offset 0A Hex from the logical base address, is used in combination with the Serial Number Low register to define the serial number of the V165 module. These registers are read-only and a write operation to these registers has no effect.

Model V165-xBx1

The format and bit assignments for the Serial Number High register are as follows:

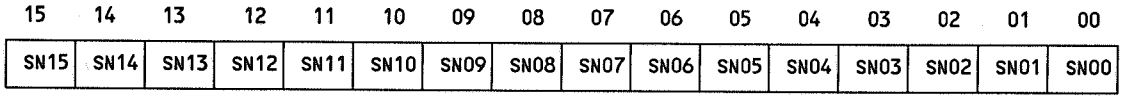


Bit(s)	Mnemonic	Meaning
15 - 00	SN31 - SN16	These are the upper 16-bits of the 32-bit module serial number. These bits are used in conjunction with bits SN15-SN00 located in the Serial Number Low register.

Serial Number Low Register

The Serial Number Low register, which is located at offset 0C Hex from the logical base address, is used in combination with the Serial Number High register to define the serial number of the V165 module. These registers are read-only and a write operation to these registers has no effect.

The format and bit assignments for the Serial Number Low register are as follows:

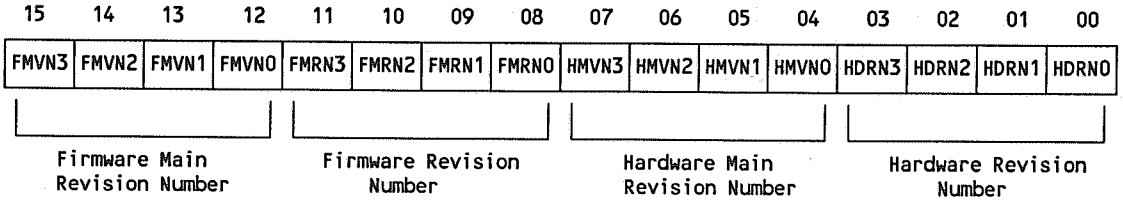


Bit(s)	Mnemonic	Meaning
15 - 00	SN15 - SN00	These are the lower 16-bits of the 32-bit module serial number. These bits are used in conjunction with bits SN31-SN16 located in the Serial Number High register.

Version Number Register

The Version Number register, which is located at offset 0E Hex from the logical base address, reflects the current revision level of the hardware and firmware residing on the V165. This is a read-only register and any write operations to this register are ignored.

The format and bit assignments for the Version Number register are as follows:



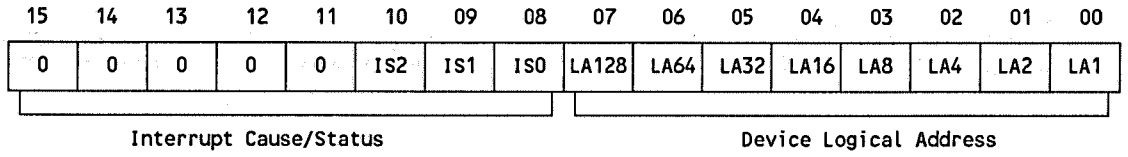
Model V165-xBx1

Bit(s)	Mnemonic	Meaning
15 - 12	FMVN3-FMVN0	These bits reflect the modules firmware main revision level.
11 - 08	FMRN3-FMRN0	These bits reflect the modules firmware revision number.
07 - 04	HMVN3-HMVN0	These bits reflect the modules hardware main revision level.
03 - 00	HDRN3-HDRN0	These bits reflect the modules hardware revision number.

Interrupt Status Register

The Interrupt Status register is a 16-bit read-only register which is used for determining the cause of an interrupt generated by the V165. The DSP is the only mechanism on the V165 that may generate an interrupt to VME. The DSP writes to the Source VXI Interrupt register which asserts an interrupt. A 3-bit data pattern that is written to the Source VXI Interrupt register by the DSP is then stored as the cause for the VME interrupt. When the Interrupt Status register is read by a programmed I/O, or by an interrupt acknowledge cycle, the 3-bit pattern is enabled onto data lines 10 through 08. Bits 7 through 0 reflect the logical address of the V165.

The format and bit assignments for the Interrupt Status register are as follows:



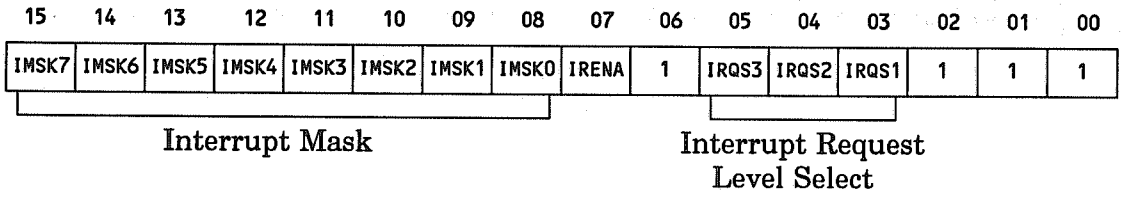
Bit(s)	Mnemonic	Meaning
15 - 08	Cause/Status	This field reflects the data pattern written by the DSP to the Source VXI Interrupt register to cause the interrupt.
07 - 00	Logical Address	Logical address of the V165.

Interrupt Control Register

The Interrupt Control register, which is located at offset 1C Hex from the logical base address, is a 16-bit write/read register which is used to configure the V165 for interrupt generation. The Interrupt Request Level, Interrupt Mask field, and Interrupt Enable bit are located in this register.

Model V165-xBx1

The format and bit assignments for the Interrupt Control register are as follows:



Bit(s)	Mnemonic	Meaning
15 - 10	IMSK7-IMSK0*	These 8 Interrupt Mask bits are used to enable/disable the generation of an interrupt from the 8 possible sources. These bits enable the various data patterns available for interrupt generation by the DSP writing to the Source VXI Interrupt register. A zero in each location enables the interrupt source and a one disables it.
07	IR ENA*	This Interrupt Request Enable bit is used to disable and enable the generation of an interrupt by the V165. Setting this bit to a one disables interrupt generation and clearing this bit to a zero enables interrupts.
06	Not Used	This bit is not used and should be written to a one.
05 - 03	IRQS3-IRQS1	The Interrupt Request Select bits are used to select the desired Interrupt Request Level that the V165 asserts when an interrupt is sourced. The following shows the binary combination that yield the Interrupt Request Level selections: **IRQS3 - IRQS2 - IRQS1** 111- disconnected 110 - Interrupt Request Level 1 101 - Interrupt Request Level 2 100 - Interrupt Request Level 3 011 - Interrupt Request Level 4 010 - Interrupt Request Level 5 001 - Interrupt Request Level 6 000 - Interrupt Request Level 7
02 - 00	Not Used	These bits are not used and should be written as ones.

Model V165-xBx1

Subclass Register

The Subclass register, which is located at offset 1E Hex from the logical base address, is a 16-bit read-only register that reflects the subclass of the V165. The V165 is an Extended Register Based Device as the following bit pattern indicates:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit(s)	Meaning
15	This bit is set to one indicating that this is an VXIbus Extended Device.
14 - 0	These bits are set to 7FFE Hex which indicates that the V165 is an Extended Register Based Device.

Suffix High Register

The Suffix High register, which is located at offset 20 Hex from the logical base address, is a 16-bit read-only register used in combination with the Suffix Low register to determine the module suffix. The Suffix High register contains the first two ASCII characters of the module suffix and the Suffix Low register contains the second two characters. The suffix shown is for a V165-ZBD1 module.

The format and bit assignments for the Suffix High Register are as follows:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1

This read only register contains the first two characters of the module's suffix ("ZB" = 5A42 Hex)

Suffix Low Register

The Suffix Low register, which is located at offset 22 Hex from the logical base address, is a 16-bit read-only register used in combination with the Suffix High register to determine the module suffix. The Suffix Low register contains the second two ASCII characters of the module suffix and the Suffix High register contains the first two characters. The suffix shown is for a V165-ZBD1 module.

The format and bit assignments for the Suffix Low Register are as follows:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1

This read only register contains the second two characters of the modules suffix ("D1" = 4431 Hex)

Model V165-xBx1

VXI Operational Registers

The Operational registers are the addresses through which the various functions of the V165 are controlled. Since the V165 requires 8 Megabytes of VME address space for the Operational registers, they are positioned in VMEbus Extended Address (A32) space. The base address for these registers is defined by the contents of the Offset register within the Configuration Register set. This offset is typically loaded by the Resource Manager during configuration. The physical address of the Operational registers is calculated by shifting the data found in the Offset register 16-bit locations to the left. For example, if the value 2000 Hex is loaded in the Offset register, the physical address for the base of the Operational registers is at 20000000 Hex.

All registers in A32 address space can be accessed as 32-bit longwords and 16-bit shortwords. These registers cannot be accessed as bytes.

Prior to gaining access to the Operational Registers, the A32 Enable bit (bit 15) must be set in the Status/Control Register. **Note: The V165 will only respond to these addresses if the Extended Address Modifier Codes (0F - 0D Hex or 0B - 09 Hex) are set for the bus cycles.**

The following chart shows the registers accessible through VXI A32 address space and their offsets from the operational base address.

	A32 Address Offset	Access	Function
	00000000	Read/Write	Semaphore Flag #1
	00000004	Read/Write	Semaphore Flag #2
	00000008	Read/Write	Semaphore Flag #3
	0000000C	Read/Write	Semaphore Flag #4
	00000010	Write-only	DSP Control Register
*	00000014	Write-only	Frame Interval Counter
	00000018	Write-only	Generate DSP Interrupt
	0000001C		
	•	Reserved	
	0000003C		
*	00000040	Read/Write	Frame Data Selection Word #1
*	00000044	Read/Write	Frame Data Selection Word #2
*	00000048	Read/Write	Frame Data Selection Word #3
	•		
	•		

Model V165-xBx1

	A32 Address Offset	Access	Function
*	0000007C	Read/Write	Frame Data Selection Memory Word #16
	●		
	3FFFFF		
	400000		
	●	Read/Write	4 Megabytes Dynamic Memory (DRAM)
	●		
	7FFFFF		
NOTE:	* Indicates registers that are only available on a V165 with the V165-Bxx1 DIGIBUS option.		

Semaphore Flags

The V165 provides four individual Semaphore flags to establish a communication scheme between the DSP and the host computer, via the VME bus. There are four addresses within the VME address space and also four addresses within the DSP address space. To obtain a flag, the requesting processor executes a read to the Semaphore flag desired. If a value of zero is obtained by the processor, the flag is considered transferred to that processor. If a value of 80008000 Hex is received, the flag has already been obtained by the other processor. When the processor that has "won" a flag desires to relinquish it, it simply executes a write operation to the Semaphore Flag address which it wishes to release. Any data may be used to free a flag. Flags are obtained on a first-come first-serve basis. All requests for the flags go through a arbitration sequence to prevent a simultaneous granting of a flag.

The following shows the bit pattern for the 32-bit Semaphore Flags. When accessing these flags as 16-bit quantities, address each flag by its defined offset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FLG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31 and 15 are the actual flag bits. Reading these bits as zero indicates that the flag has been transferred. If these bits are read as ones, the flags are currently unavailable. Bits 30 through 16 and 14 through 0 are not used and read as zeros.

As an example of Semaphore Flag use, assume an application program running on the DSP uses a section of the dual-access DRAM for data storage in an array and the host processor also needs access to the same addresses within the DRAM. This dual-access DRAM can be accessed by either the DSP or by VME operations. An arbitration mechanism prevents simultaneous access to the DRAM.

In the example, the host processor cannot alter the contents of the data array while the DSP is in the process of updating the array. Similarly, the DSP cannot alter the contents of the data array while the host processor is in the process of updating the array. Therefore, a communication protocol is required for determining "ownership" of the array.

If either the host processor or DSP need to access the array, it must verify that the array is not "owned" by the other processor. When the DSP desires to update the array, it must execute a read command to the appropriate Semaphore Flag to establish "ownership". If the read command resulted in read data of 80008000 Hex, it must continue to execute the read command until data of zero is received. When data of zero is received, it may then access the array and make any necessary changes to the data within the array. Once the DSP has completed the update, it then releases the flag by executing a write operation to the Semaphore Flag that it obtained earlier. This frees the flag and the arbitration process continues.

DSP Control Register

The DSP Control register is a write-only register that is used to start/stop DSP operation and to configure the starting address for DSP code execution after a RESET operation.

The RESET bit in this register is used to assert and clear the RESET line to the DSP. Setting this bit to a zero stops the DSP from executing code and places the DSP into a reset state. To enable the DSP for code execution and exit the reset state, set this bit to a one.

After the RESET has been removed from the DSP, the DSP reads the data at address location zero in physical address space to determine where the next instruction resides. The V165 allows for 8 starting address selections. Four of these reside in the PROM and the other four reside in the slow-access DRAM. For further information regarding DSP activity after a RESET condition, please refer to the section entitled DSP Reset.

The following shows the bit pattern for the DSP Control register along with a description of the bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	BA2	BA1	BA0	RESET

Bit(s)	Mnemonic	Meaning
31-04	Not used	These bits are not used.
03-01	BA2-BA0	The binary combination of these bits determine the starting address of DSP code execution following a RESET condition as follows :

BA2	BA1	BA0	Starting Address
0	0	0	810000
0	0	1	810002
0	1	0	810004
0	1	1	810006
1	0	0	100000
1	0	1	100002
1	1	0	100004
1	1	1	100006

00 Reset

This bit is used to assert/clear the RESET line of the DSP. Setting this bit to a one enables the DSP to execute code. Setting this bit to a zero places the DSP in the RESET state.

Frame Interval Register

The Frame Interval register is only available when the V165 is equipped with the DIGIBUS option. This 32-bit write/read register is used to specify the interval at which the V165 stores DIGIBUS data frames. The Capture bit located in this register enables/disables the capturing of DIGIBUS data by the V165-Bxx1.

The FC7 through FC0 bits define the interval at which frames are captured from the DIGIBUS. Setting this field to zero instructs the DIGIBUS capture circuitry to store every frame of data that it sees on the DIGIBUS. To capture every other frame of data, set this field to one. The maximum value for this field is 255. This frame interval selection can ease the burden on the DSP when receiving data from the DIGIBUS when not every frame of data is required for the application.

The CAPTURE bit in the Frame Interval register is used to enable and disable the capturing of data from the DIGIBUS. Setting this bit to a one enables the capturing of data from the DIGIBUS. After the bit is set, the capture mechanism waits for the start of a frame before storing data. The storage of data will not start in the middle of a frame.

The following shows the bit pattern for the Frame Interval register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	CAP	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0

Model V165-xBx1

Bit(s)	Mnemonic	Meaning
31-09	Not used	These bits are not used and read as zeros.
08	CAP	This bit is used to enable/disable DIGIBUS data capturing. Setting this bit to a one enables capture and a zero disables capture.
07-00	FC7-FC0	Frame interval bits 7 through 0 control the interval at which DIGIBUS frames are store.

Frame Data Selection Words

The Frame Data Selection Words are only used with the V165-Bxx1 DIGIBUS option for the V165. These 16 16-bit words control which samples within each frame are stored. The selection words have a bit location for each of the 256 samples within a frame. The maximum number of samples in any given frame is 256.

Setting a bit to a one enables the corresponding data sample to be stored from the DIGIBUS. A zero in any given location indicates that the sample is to be discarded. The following chart shows the bit layout for the selection words. Bits 31 through 16 of these longwords are not used.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
SMP 15	SMP 14	SMP 13	SMP 12	SMP 11	SMP 10	SMP 9	SMP 8	SMP 7	SMP 6	SMP 5	SMP 4	SMP 3	SMP 2	SMP 1	SMP 0	:40
SMP 31	SMP 30	SMP 29	SMP 28	SMP 27	SMP 26	SMP 25	SMP 24	SMP 23	SMP 22	SMP 21	SMP 20	SMP 19	SMP 18	SMP 17	SMP 16	:44
•																
•																
•																
•																
SMP 255	SMP 254	SMP 253	SMP 252	SMP 251	SMP 250	SMP 249	SMP 248	SMP 247	SMP 246	SMP 245	SMP 244	SMP 243	SMP 242	SMP 241	SMP 240	:7C

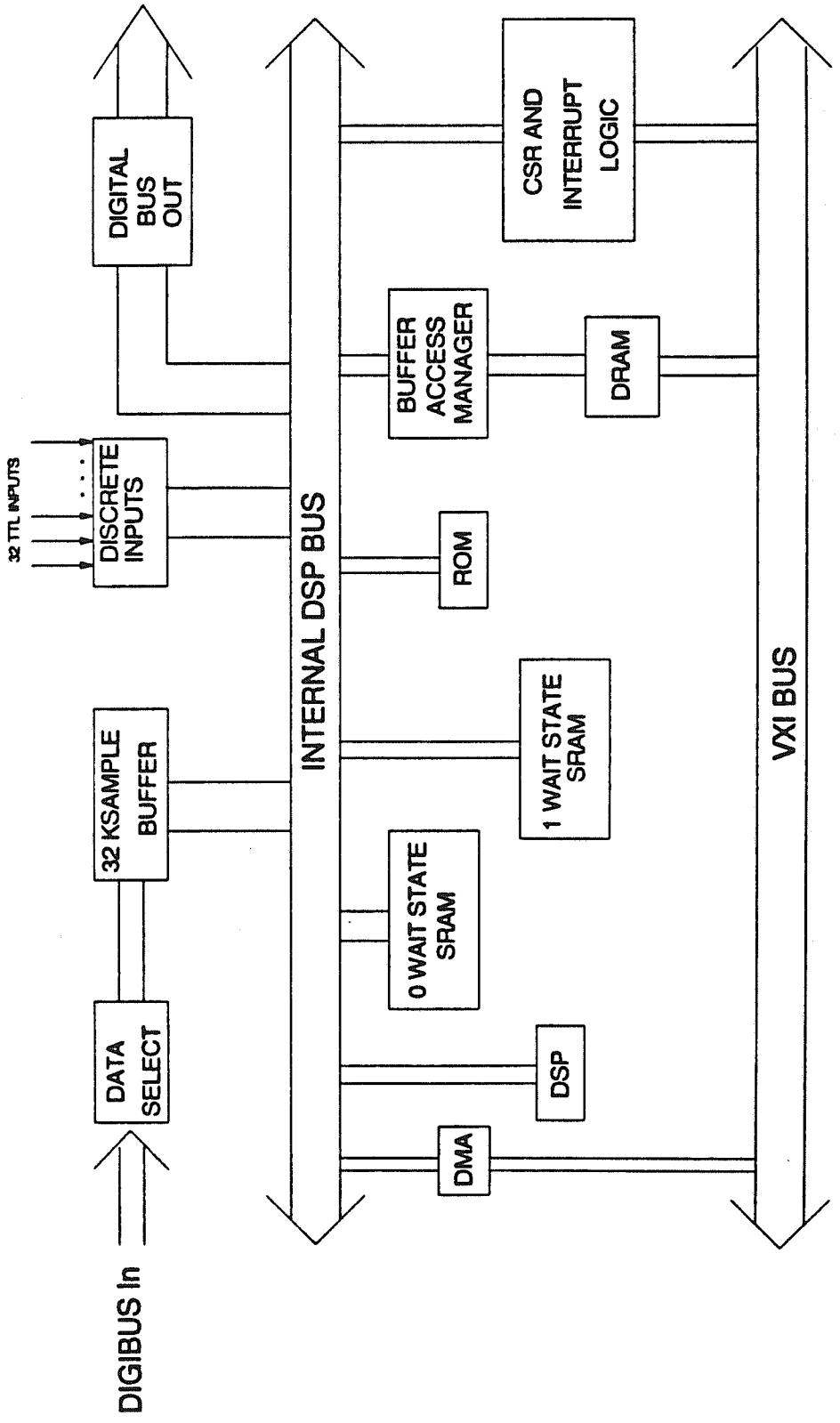
DRAM

The V165 has 4 Megabytes of Dynamic Random Access Memory (DRAM). This memory is accessible by both the DSP and the host processor. The starting address of this memory is at an offset of 400000 Hex from the operational registers base address and continues to 7FFFFF Hex. This memory is configured as 32-bit longwords, but may be accessed as 16-bit shortwords.

Model V165-xBx1

DSP Architecture

The basic architecture of the DSP and its support circuitry are shown in the following block diagram.



The DSP is a Texas Instruments (TI) TMS320C30 40 Megahertz component with a 50 nanosecond (ns) cycle time. This means that the DSP may accesses data/code from any zero wait-state external component in 50 nanoseconds. External devices that require one wait-state access cause the DSP to add one additional cycle, increasing the access time to 100 nanoseconds.

The DSP has two busses, a primary bus and an expansion bus. The primary bus contains 32-bits of data and 24-bits of addressing. The expansion bus contains 32-bits of data and 13-bits of addressing. In the V165 implementation, all the memory components (DRAM, PROM,..) are located on the primary bus. The expansion bus is used for other peripheral communications such as semaphore flags, Direct Memory Access (DMA) interface to VME, and VXI Trigger Line access.

ALL DSP ADDRESSING IS 32 BITS WIDE.

The primary bus is used to communicate with the following :

128 Kilobyte zero wait-state Static RAM
128 Kilobyte one wait-state Static RAM
128 Kilobyte one wait-state PROM
1 Megabyte dual access Dynamic RAM
DRAM Buffer Access Manager
32 Ksample DIGIBUS Static RAM (V165-Bxx1 option only)

The expansion bus is used to communicate with the following peripherals :

Digital Input Register
DMA Control Registers
VXI Interrupt Registers
VXI Trigger Line Access Registers
Front Panel LED Display Register
SYSFAIL Register
DIGIBUS Control Registers (V165-Bxx1 option only)

DSP Memory Map

The following diagram shows the memory layout for the DSP.

0 Wait State SRAM 128 Kbytes	000000h 007FFFh
1 Wait State SRAM 128 Kbytes	008000h 00FFFFh
Reserved	10000h 0FFFFFh
Slow Access DRAM 4 Mbytes	100000h 1FFFFFh
64Kbyte DIGIBUS Memory	500000h 507FFFh
Reserved	508000h 5FFFFFh
DRAM Prefetch Address	600000h 600003h
Reserved	600004h 7FFFFFh
Semaphore Flags	800000h 800003h
Reserved	800004h 803FFFh
Expansion Bus Peripherals	804000h 804010h
Reserved	804011h 807FFFh
DSP Internal Memory Mapped Registers	808000h 8097FFh
DSP Internal RAM Block 0	809800h 809BFFh
DSP Internal RAM Block 1	809C00h 809FFFh
Reserved	80A000h 80FFFFh
128 Kbyte PROM	810000h 817FFFh
Reserved	818000h 8FFFFFh
Buffer Access Manager DRAM 4 Mbytes	900000h 9FFFFFh
Reserved	D00000h FFFFFFh

Model V165-xBx1

The following chart indicates the address assignments of the peripheral components connected to the DSP and the bus to which it is connected.

00000-0007FFF	128 Kbyte zero wait-state SRAM	Primary
008000-00FFFF	128 Kbyte one wait-state SRAM	Primary
010000-0FFFFF	Reserved	- - -
100000-13FFFF	1 Mbyte Slow Access DRAM	Primary
500000-507FFF	64 Kbyte DIGIBUS SRAM (V165-Bxx1 only)	Primary
508000-5FFFFF	Reserved	- - -
600000-600000	Buffer Manager Prefetch Address	Primary
600001-7FFFFF	Reserved	- - -
800000-800003	Semaphore Flags	Expansion
800004-803FFF	Reserved	- - -
804000-804010	I/O Mapped Peripherals	Expansion
804011-807FFF	Reserved	- - -
808000-8097FF	DSP Internal Peripherals	Internal
809800-809BFF	DSP Internal 4 Kbyte SRAM	Internal
809C00-809FFF	DSP Internal 4 Kbyte SRAM	Internal
80A000-80FFFF	Reserved	- - -
810000-817FFF	128 Kbyte one wait-state PROM	Primary
818000-8FFFFF	Reserved	- - -
900000-93FFFF	1 Mbyte Buffer Manager Access DRAM	Primary
D00000-FFFFFF	Reserved	- - -

DSP Memory Accesses

The DSP has access to 256 Kilobytes of SRAM, 128 Kilobytes of PROM and 4 Megabytes of DRAM. The DSP also has 8 Kilobytes of internal zero wait-state memory. The 256 Kilobytes of SRAM is divided into two banks. The first bank has zero wait-state (50 nanosecond) access and the second has one wait-state (100 nanosecond) access. The 128 Kilobytes of PROM is also has a one wait-state (100 nanosecond) access time.

The V165 has 4 Megabytes of DRAM. This memory is mapped into two locations in the DSP's address space. The previous memory map shows that the location of the Slow Access DRAM at address locations 100000 - 1FFFFF Hex. The same physical memory also resides at address locations 900000 - 9FFFFF Hex. The difference in between these two address assignments is the manner in which the DRAM is accessed. The addresses that correspond to the Slow Access DRAM are slower than the accesses to the Buffer Access Manager DRAM.
When executing DRAM write or read operations to the Slow Access DRAM, the DSP must wait for the DRAM write or read operation to complete before it may continue processing. If there is no contention from the VMEbus accessing the DRAM, the access time is typically 350 to 400 nanoseconds. Many DSP applications use the DRAM to store results of DSP computations. Since these transfers do not typically occur as back-to-back transfers, a mechanism has been incorporated into the V165 to reduce the time it takes to access the DRAM. In order to make these transfers more efficient, a Buffer Access Manager is included to speed up the transfer of data to/from the DRAM.

The Buffer Access Manager can speed up both write and read transfers to DRAM. The transfers only become more efficient if the time between accesses to the DRAM is about 500 nanoseconds, or 10 bus cycles of the DSP. For writes to DRAM using the Buffer Access

Manager, the DSP simply writes to the desired memory location in the range of addresses from 900000 to 9FFFFF Hex. The write data and address for the write operation are stored in latches so that the Buffer Access Manager can cycle the write data to DRAM while the DSP continues processing. The DSP can execute this write while only imposing one wait-state, yielding a 100 nanosecond access time. If a subsequent DRAM write/read access occurs within 400 nanoseconds after completing the one-wait state write, that operation will take the full 350 to 400 nanoseconds.

To execute reads from the DRAM using the Buffer Access Manager, the Prefetch Memory Address register, located at address 600000 Hex, must first be loaded with the first address to be accessed during the block read. After loading the prefetch address, the DSP may then read the DRAM address specified when loading the prefetch address. After the DSP reads the DRAM for the first time after loading the prefetch address, the data is returned to the DSP in the 350 to 400 nanosecond period. After the data has been read by the DSP, the Buffer Access Manager initiates a DRAM read operation to the next sequential address in the DRAM. If the DSP does not read the DRAM for the 350 to 400 nanoseconds, the prefetched data can be returned to the processor with only one wait-state (100 nanoseconds).

If it is necessary to access the DRAM faster than the default 350 to 400 nanosecond interval, use the Buffer Access Manager. Remember that this mechanism increases memory access efficiency when accesses to the DRAM are separated by the 350 to 400 nanosecond time periods.

Note: The DSP *MUST NOT* execute code from the DRAM using the Buffer Access Manager. The DSP may execute code out of DRAM, but it must be in the address range of 100000 to 1FFFFF.

DSP Expansion Bus I/O Mapped Registers

The following section describes the registers connected to the Expansion Bus of the DSP. All these registers are memory mapped and are accessed in one wait-state (100 nanoseconds). Since the DSP does not allow for byte or shortword addressing, all registers shown here are 32-bits wide.

The following chart shows the registers accessible as I/O mapped locations and their corresponding addresses.

	Registers	Access	Function
*	804000	Write-only	Decrement Frame Counter
*	804000	Read-only	Frame Counter
	804001	Read-only	Digital Input Register
	804002	Write/Read	DMA Data Register
	804003	Write/Read	DMA Control Register
	804004	Write-only	DMA Address Register
	804005	Write-only	VXI Interrupt Source Register

*	804005	Read-only	Frame Address Register
	804006	Write-only	VXI Trigger Source Register
	804007	Write/Read	VXI Trigger Select Register
	804008	Reserved	
*	804009	Write-only	Generate DIGIBUS Output Cell with Data
*	80400A	Write-only	Generate DIGIBUS Output Cell without Data
*	80400B	Write-only	Start DIGIBUS Output Frame
*	80400C	Write-only	Terminate DIGIBUS Output Frame
*	80400D	Write-only	Enable DIGIBUS Data FIFO to Memory Transfers
*	80400E	Write-only	Disable DIGIBUS Data FIFO to Memory Transfers
	80400F	Write-only	LED Display Register
	804010	Write-only	SYSFAIL Register

NOTE: * Indicates that these registers are only supported on a V165 equipped with a V165-Bxx1 DIGIBUS option. These registers are described under the section entitled DIGIBUS Acceptor Control Registers and DIGIBUS Source Control Registers.

Digital Input Register

The Digital Input register is a 32-bit read-only register which is used to read the state of the 32 digital inputs that enter the V165 through the "DIGITAL INPUTS" connector on the front-panel of the module. These inputs are TTL compatible and have input protection to guard against overvoltage applied at the connections.

These TTL inputs are pulled up through a 4700 ohm resistor to +5 volts. Any input to the module that is a TTL one (high voltage level) is read by the DSP as a logical 1. Any input to the module that is a TTL zero (low voltage level) is read by the DSP as a logical 0.

The chart in Appendix A shows the allocation of the digital inputs on the front-panel connector. INP 0 on the connector corresponds to INP 0 in the register; INP 1 on the connector corresponds to INP 1 in the register and so on. The following diagram shows the bit layout for the Digital Input register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INP31	INP30	INP29	INP28	INP27	INP26	INP25	INP24	INP23	INP22	INP21	INP20	INP19	INP18	INP17	INP16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
INP15	INP14	INP13	INP12	INP11	INP10	INP09	INP08	INP07	INP06	INP05	INP04	INP03	INP02	INP01	INP00

Model V165-xBx1

Bit(s)	Mnemonic	Meaning
31-00	INP31-INP00	TTL digital inputs 31 through 0. A high voltage on the input is read as a one and a zero voltage on the input as read as a zero.

DMA Data Register

The DMA Data register is used for holding DMA write data or receiving DMA read data. This write/read register is 32-bits wide. During DMA write operations, the data preloaded in this register is placed on the VME data lines during the bus transfer. After a DMA write operation completes, this register contains the data obtained during the bus transfer.

When the V165 executes 32-bit DMA operations, the entire DMA Data register is used for either holding write data or receiving read data. During 16-bit DMA transfers, only the lower 16-bits of this register are used. The upper 16-bits should be masked out after executing a 16-bit DMA read operation since they do not have any significance. If 8-bit DMA read transfers are used, the upper 24-bits of the obtained read data should be ignored since these bits do not have any significance.

The following diagram shows the bit layout of the DMA Data register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRD31	WRD30	WRS29	WRD28	WRD27	WRD26	WRD25	WRD24	WRD23	WRD22	WRD21	WRD20	WRD19	WRD18	WRD17	WRD16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WRD15	WRD14	WRD13	WRD12	WRD11	WRD10	WRD09	WRD08	WRD07	WRD06	WRD05	WRD04	WRD03	WRD02	WRD01	WRD00

Bit(s)	Mnemonic	Meaning
31-00	WRD31-WRD00	This data is either the data obtained from a DMA read operation or the data to be used for a DMA write operation.

DMA Control/Status Register

The DMA Control register is used to setup, initiate, and monitor the status of a DMA operation. This write/read register also contains the Address Modifier bits that are to be used during the DMA operation. The DONE bit is set to a one when a DMA operation completes. The TIMEOUT (TMO) bit is set when the BERR signal is asserted by the addressed location or if an addressed module does not respond with DTACK in 200 microseconds. The size bits SIZ1 and SIZ0 are used to specify the size of the data word to transfer during the DMA operation. The DIRECTION (DIR) bit specifies the direction of the DMA transfer. The GO bit is used to initiate the transfer.

The following shows the bit layout of the DMA Control/Status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	AM5	AM4	AM3	AM2	AM1	AM0	DONE	TMO	0	0	SIZ1	SIZ0	DIR	GO

Bits	Mnemonic	Meaning
31-14	Not Used	These bits are not used and read as zeros.
13-08	AM5-AM0	These write-only bits are set prior to a DMA operation to specify the Address Modifier used during the bus transfer. These bits are read as zeros. The following chart is a subset of the VME address modifiers.

Address Modifier Data (Hex)	**Function**
3F	Standard (A24) Supervisory Block Transfer
3E	Standard (A24) Supervisory Program Access
3D	Standard (A24) Supervisory Data Access
3A	Standard (A24) Nonprivleged Program Access
39	Standard (A24) Nonprivleged Data Access
2D	Short (A16) Supervisory Access
29	Short (A16) Nonprivleged Access
0E	Extended (A32) Supervisory Program Access
0D	Extended (A32) Supervisory Data Access
0A	Extended (A32) Nonprivleged Program Access
09	Extended (A32) Nonprivleged Data Access

07	DONE	This read-only bit is set when a DMA operation has completed. After setting the GO bit to a one, the DONE bit goes to zero until the bus transfer completes.
06	TMO	This read-only bit is set when a DMA operation results in a timeout condition. A timeout is indicated when the addressed location asserts BERR (Bus Error) or an addressed location does not respond within 200 microseconds.

05-04 Not used — These bits are not used and read as zeros.

03-02 SIZ1-SIZ0 — These write-only bits are used to specify the size of the DMA data word. The binary combinations of these bits yield the DMA data word sizes as follows:

SIZ1	SI0	DMA Data Size
0	0	32-Bit
0	1	16-Bit
1	0	8-Bit
1	1	Reserved

These bits are read as zeros.

01	DIR	This write-only bit is used to specify the direction of the DMA transfer. Setting this bit to a one indicates that the data transfer direction is into the V165 (DMA read). A zero in this location specifies that the data transfer direction it out of the V165 (DMA write). This bit is read as zero.
00	GO	This write-only bit is used to initiate the DMA transfer. Once the GO bit is set to a one, the DONE bit is set to zero until the operation is complete. The GO bit is read as a zero.

DMA Address Register

The DMA Address register is used for specifying the physical address that is to be accessed during the DMA operation. The contents of this register are output onto the VME A31-A1 address lines during the DMA transfer. All bits in this register are write-only.

The following shows the bit layout for the DMA Address register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	N/U

Bit(s)	Mnemonic	Meaning
31-01	A31-A01	These write-only bits specify the physical address to be accessed during the DMA operation. The contents of this register are placed on the VME address lines A31 though A01.
00	Not used	This bit is not used.

VXI Interrupt Source Register

The VXI Interrupt Source register is a write-only register used to generate a VXI interrupt. Before a VXI interrupt may be generated, the Interrupt Control configuration register must be initialized. The Interrupt Control register must be loaded with the desired Interrupt Request Level, the required Interrupt Mask field, and the Interrupt Request bit must be enabled. Please refer to the Interrupt Control Register section of this manual for additional information.

The VXI Interrupt Source register is written with a 3-bit encoded pattern which in turn generates a VXI interrupt. This 3-bit encoded pattern maps to the 8 interrupt sources available as follows:

Model V165-xBx1

IS2	IS1	IS0	Interrupt Source
0	0	0	Interrupt Source 0
0	0	1	Interrupt Source 1
0	1	0	Interrupt Source 2
0	1	1	Interrupt Source 3
1	0	0	Interrupt Source 4
1	0	1	Interrupt Source 5
1	1	0	Interrupt Source 6
1	1	1	Interrupt Source 7

The following diagram shows the bit layout of the VXI Interrupt Source register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	ENA	IS2	IS1	IS0

Bit(s)	Mnemonic	Meaning
31-04	Not used	These bits are not used.
03	ENA	This write-only bit is used to enable the generation of a VXI interrupt. When setting the IS2-IS0 bits to the desired pattern, the ENABLE bit must also be set to a one.
02-00	IS2-IS0	These write-only bits are used to specify the interrupt source of the VXI interrupt.

To enable the individual interrupt sources, the corresponding mask bit in the Interrupt Control register must be set to a one. Once the interrupt source, interrupt enable, and interrupt request level have been setup in the Interrupt Control register, the DSP may then generate a VXI interrupt by writing to the VXI Interrupt Source register.

When a VXI interrupt is acknowledged by an Interrupt Handler, a Status\ID field is passed to the handler on the lower 16 data lines. The lower 8-bits of this data contains the logical address of the module generating the interrupt. The upper 8 bits of this data represent the cause of the interrupt. The V165 sets the lower 3 bits of this 8 bit field to the IS2 through IS0 data written by the DSP. The following bit pattern shows the format of the 16-bit Status/ID word sent to the Interrupt Handler during the Interrupt Acknowledge cycle.

Model V165-xBx1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	IS2	IS1	IS0	LA128	LA64	LA32	LA16	LA8	LA4	LA2	LA1

Bit(s)	Mnemonic	Meaning
15-11	Not used	These bits are not used and returned as zeros.
10-08	IS2-IS0	These bits reflect the 3-bit encoded pattern written into the VXI Interrupt Source register by the DSP.
07-00	LA128-LA1	These bits represent the current Logical Address of the V165.

VXI Trigger Source Register

The VXI Trigger Source register is used to generate VXI triggers source by the DSP. The V165 can source both the ECL and TTL trigger lines. There are two ECL trigger lines and eight TTL trigger lines. The V165 can source these triggers individually or in any combination. The only trigger protocol supported by the V165 is the Synchronous Trigger Protocol. This protocol is a single line broadcast trigger that does not require an acknowledgement from any acceptor.

When this register is written, all the bits set to a one in the register generate a 260ns pulse on the corresponding trigger line. The following chart shows the bit layout of the VXI Trigger Source register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	ECLT1	ECLT0	TTLT7	TTLT6	TTLT5	TTLT4	TTLT3	TTLT2	TTLT1	TTLT0

Bit(s)	Mnemonic	Meaning
31-10	Not used	These bits are not used.
09-08	ECLT1-ECLT0	These bits correspond to the VXI ECL trigger lines 1 and 0.
07-00	TTLT7-TTLT0	These bits correspond to the VXI TTL trigger lines 7 through 0.

VXI Trigger Select Register

The VXI Trigger Select register is used to enable and select a VXI trigger line that can generate an interrupt to the DSP. The interrupt to the DSP is at interrupt level 0 (INT0).

Model V165-xBx1

The VXI Trigger Select register is written with an encoded pattern which routes a particular VXI trigger line to the interrupt signal of the DSP. The selection mechanism allows for only one trigger line to be used at any given time.

The following shows the bit layout for the VXI Trigger Select register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	ENA	SELT3	SELT2	SELT1	SELT0

Bit(s)	Mnemonic	Meaning
31-05	Not used	These bits are not used and read as zeros.
04	ENA	The ENABLE bit is used to enable/disable the generation of a DSP interrupt on the occurrence of a selected VXI trigger line. Setting this bit to a one enable the DSP interrupt and disabled by setting this bit to a zero.
03-00	SELT3-SELT0	These write/read bits are used to select the VXI trigger line that is routed to the DSP interrupt line. The binary combination of these bits determine which VXI trigger line is routed as follows:

SELT3	SELT2	SELT1	SELT0	Trig Line
0	0	0	0	TTL Trig 0
0	0	0	1	TTL Trig 1
0	0	1	0	TTL Trig 2
0	0	1	1	TTL Trig 3
0	1	0	0	TTL Trig 4
0	1	0	1	TTL Trig 5
0	1	1	0	TTL Trig 6
0	1	1	1	TTL Trig 7
1	0	0	0	ECL Trig 0
1	0	0	1	ECL Trig 1
1	0	1	0	Reserved
.	.	.	.	.
1	1	1	1	Reserved

Model V165-xBx1

LED Display Register

The LED Display register is an 8-bit write-only register which is used to present diagnostic and other visual information to the operator through the front panel LEDs. The LEDs are initially used by the power-on self-test to indicate whether the test passed or failed. If the test fails, a code is written to the LED Display register reflecting the cause of the error.

The format of the LED Display register is shown in the following diagram. LED7 refers to the topmost LED on the front panel Status array, and LED0 refers to the bottommost LED.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0

Bit(s)	Mnemonic	Meaning
31-08	Not used	These bits are not used.
07-00	LED7-LED0	These write-only bits are used to illuminate the front panel Status LEDs. Setting a bit to a one illuminates the LED and a zero turns the LED off.

SYSFAIL Register

The SYSFAIL register is a write-only register which is used by the V165 to assert the VME SYSFAIL signal. In order for the SYSFAIL line to be asserted, the SYSFAIL Inhibit bit in the VXI Control register in configuration space must be set to a zero. This register is used by the V165 during the power-on self-test to assert SYSFAIL until the self-test successfully completes. If an error occurs during the self-test, the SYSFAIL line is left asserted, indicating to the slot zero controller that a fault has been detected.

The following shows the bit pattern for the SYSFAIL register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	N/U	SFAIL

Bit(s)	Mnemonic	Meaning
31-01	Not used	These bits are not used.

00	SFAIL	This write-only bit is used to set and clear the VME SYSFAIL signal. Setting this bit to a one asserts the SYSFAIL line and clearing this bit to a zero negates the V165's assertion of the SYSFAIL line. The SYSFAIL Inhibit bit in the VXI Control register must be set to zero to enable the V165's assertion of the SYSFAIL line.

DSP Semaphore Flags

The V165 provides four individual Semaphore flags to establish a communication scheme between the DSP and the host computer, via the VME bus. There are four addresses within the VME address space and also four addresses within the DSP address space. To obtain a flag, the requesting processor executes a read to the Semaphore flag desired. If a value of zero is obtained by the processor, the flag is considered transferred to that processor. If a value of 80000000 Hex is received, the flag has already been obtained by the other processor. When the processor that has "won" a flag desires to relinquish it, it simply executes a write operation to the Semaphore Flag address which it wishes to release. Any data may be used to free a flag. Flags are obtained on a first-come first-serve basis. All requests for the flags go through a arbitration sequence to prevent a simultaneous granting of a flag.

The following shows the bit pattern for the 32-bit Semaphore Flags.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31 is the flag bit. Reading this bit as zero indicates that the flag has been transferred. If the flag is read as one, the flag is currently unavailable. Bits 30 through 0 are not used and read as zeros.

As an example of Semaphore Flag use, assume an application program running on the DSP uses a section of the dual-access DRAM for data storage in an array and the host processor also needs access to the same addresses within the DRAM. This dual-access DRAM can be accessed by either the DSP or by VME operations. An arbitration mechanism prevents simultaneous access to the DRAM.

In the example, the host processor cannot alter the contents of the data array while the DSP is in the process of updating the array. Similarly, the DSP cannot alter the contents of the data array while the host processor is in the process of updating the array. Therefore, a communication protocol is required for determining "ownership" of the array.

If either the host processor or DSP need to access the array, it must verify that the array is not "owned" by the other processor. When the DSP desires to update the array, it must execute a read command to the appropriate Semaphore Flag to establish "ownership". If the read command resulted is read data of 80000000 Hex, it must continue to execute the read command until data of zero is received. When data of zero is received, it may then access the

array. Once the DSP has completed the update, it then releases the flag by executing a write operation to the Semaphore Flag that it obtained earlier. This frees the flag and the arbitration process continues.

DSP Reset Vector

The DSP is reset on power-up, by depressing the RESET switch on the front-panel, or by setting the RESET bit in the DSP Control register to a zero. Once the DSP exits the reset state, it examines physical address location zero to determine the address of the first instruction to be executed.

When the RESET signal is applied to the DSP, the first block of SRAM is mapped out and is replaced with a buffer. This buffer contains the initial address (vector) that the DSP uses to start executing code. After the DSP reads the vector, the first bank of SRAM is mapped back in and the buffer is mapped out.

The reset vector can be set through the VME bus using programmed I/O or through hardware via 3 straps located on the V165. Either of these options allows for 8 possible reset vector addresses. Four of these addresses are located in PROM address space and the other four are located in slow access DRAM address space. The following chart shows the various reset vectors obtainable based on the setting of the reset vector straps/register bits.

BA2	BA1	BA0	Reset Vector Address
0	0	0	810000
0	0	1	810002
0	1	0	810004
0	1	1	810006
1	0	0	100000
1	0	1	100002
1	1	0	100004
1	1	1	100006

If the application requires that the reset vector by alterable through programmed I/O, the strap selectable positions **MUST** be set to provide a vector address of 810000. Please refer to the DSP Control Register and DSP Reset Vector Address Straps sections of this manual for additional information on configuring the reset vector address.

DSP Self-Test

The V165 has an integral self-test feature as the initial stage of its PROM-based DSP firmware. This self-test executes immediately after the 'C30 DSP is reset, which will occur after any of the following events:

- power is turned on,
- the VME SYSRESET signal is asserted,
- the Soft Reset bit in the V165's Control configuration register (in VXI A16 space) is asserted, or
- the V165's front panel RESET switch is activated.

Once the self-test starts, the DSP firmware first asserts the SYSFAIL line on the VMEbus, which turns on the front panel FAILED LED, and then the firmware writes the value 80 hex to the Status LEDs. The SYSFAIL signal will be negated once the critical memories on the V165 have passed self-test. If the FAILED LED never comes on after the V165 is reset (by one of the four events listed above), the V165 is not functioning properly and must be returned for repair.

The DSP firmware then executes the following tests, and as each test passes, a new value is written to the Status LEDs:

TEST	PASS	FAIL
Checksum of the PROM itself	81 hex	80 hex
Internal 'C30 DSP RAM read/write	82 hex	81 hex
V165 minimum SRAM read/write	83 hex	82 hex
V165 minimum DRAM read/write	84 hex	83 hex

Therefore, if the next test fails, the Status LEDs will still show the value of the last test. If a module does not consistently reach the end of self-test (does not display a value of 84 hex on the Satus LEDs and/or does not turn off the FAILED LED), then it must be returned to KineticSystems for repair. Please record the FAILED and Status LED values and mention the eror code when requesting a Return Authorization (RA) number. This is particularly true if the value on the Status LEDs is not in the pass/fail table above.

NOTE: The 80 hex bit on the Status LED display is the one closest to the FAILED and RUN LEDs, while the 01 hex bit is closest to the Reset switch and Serial Ports connector.

DSP Interrupts

The DSP can be interrupted from three sources. These sources include a preselected VXI Trigger line, a VXI request to interrupt the DSP, and a DIGIBUS FIFO overflow. The following chart shows the allocation of the interrupt sources to the DSP's interrupt lines.

INT0 Selected VXI Trigger
INT1 VXI Request
INT2 DIGIBUS FIFO Overflow
INT3 Reserved

The DSP can be interrupted at level 0 when a selected VXI trigger is received by the V165. The VXI Trigger Select register is used to select which of the 8 TTL or 2 ECL VXI Trigger lines is to generate a DSP interrupt. Only one trigger line may be enabled at any given time.

The DSP is interrupted at level 1 when a VXI request is made via programmed I/O. The host processor can execute a write to the Generate DSP Interrupt register, which in turn interrupts the DSP.

The level 2 DSP interrupt is caused by an error condition occurring on the V165-Bxx1 DIGIBUS option. This error can be caused by two conditions. The first cause may be that the DSP has disabled the DIGIBUS FIFO-To-Buffer transfers for too long of a time. This causes DIGIBUS samples to back up into the FIFO until it eventually overflows. Another cause of this may be that the DSP is continually reading the circular buffer and not allowing enough time for the DIGIBUS to store its samples in the buffer. It may be necessary in these cases to reduce the number of samples that the V165-Bxx1 is storing.

Model V165-xBx1

DSP Serial Ports

The DSP on the V165 has two independent serial ports that support bidirectional transfers. These ports can be configured to transfer 8, 16, 24 or 32-bits of data per word. The clock for each serial port can be generated internally be the DSP or externally through the front-panel connector. Please refer to the Texas Instruments TMS320C30 User's Guide for additional information.

Each serial port has a transmit clock, receive clock, transmit data, receive data, transmit synchronization, and receive synchronization pin located on the front panel connector of the V165. The following chart shows the signal allocation on the Serial Port front-panel connector for these control signals.

CLKX0	Serial Port 0 Transmit Clock	Pin 20
DX0	Serial Port 0 Transmit Data	Pin 22
FSX0	Serial Port 0 Transmit Synchronization	Pin 24
CLKR0	Serial Port 0 Receive Clock	Pin 28
DR0	Serial Port 0 Receive Data	Pin 30
FSR0	Serial Port 0 Receive Synchronization	Pin 32
CLKX1	Serial Port 1 Transmit Clock	Pin 19
DX1	Serial Port 1 Transmit Data	Pin 21
FSX1	Serial Port 1 Transmit Synchronization	Pin 23
CLKR1	Serial Port 1 Receive Clock	Pin 27
DR1	Serial Port 1 Receive Data	Pin 29
FSR1	Serial Port 1 Receive Synchronization	Pin 31

DSP Timer/Counter

The DSP on the V165 has two timer/counter signals that are routed to the front panel connector of the module. These signals are found within the connector labeled Serial Ports. These two signals can be independently configured as inputs or outputs to the DSP. As inputs to the DSP, these signal can be used to increment 32-bit counters within the DSP. As outputs from the DSP, these signals can be outputs of a timer within the DSP. Please refer to the Texas Instruments TMS320C30 User's Guide for additional information.

The following shows the allocation of the Timer/Counter bits on the front panel Serial Port connector.

TCLK0	Timer/Counter 0	Pin 8
TCLK1	Timer/Counter 1	Pin 7

DSP Flags

The DSP has two general purpose I/O bits which are routed to the front panel through the Serial Port connector. These signals may be configured independently as inputs or outputs. Please refer to the Texas Instruments TMS320C30 User's Guide for additional information.

Model V165-xBx1

The following shows the allocation of the general purpose bits on the front panel Serial Port connector.

XF0 General Purpose I/O Flag 0 Pin 4
XF1 General Purpose I/O Flag 1 Pin 3

DIGIBUS Overview

KineticSystems has developed a digital bus protocol for transferring data from one module to another using the VXI Local Bus. This bus is designed to transfer digital data from data source modules such as ADCs and Discrete Inputs to other modules for data processing and/or buffering. Since the bus is implemented on the VXI Local Bus, it does not degrade the performance of the VXI system bus, nor is it subject to the latencies that can result from addressing slow responding VXI modules.

The Local Bus connects the outer two rows of the P2 connector together on adjacent modules. This bus consists of 12 user-definable lines. The following diagram shows the Signals and their allocations on the P2 connector.

DIGIBUS	OUT	DIGIBUS	IN
D7	A15	D7	C15
D6	A14	D6	C14
D5	A12	D5	C12
D4	A11	D4	C11
D3	A9	D3	C9
D2	A8	D2	C8
D1	A6	D1	C6
D0	A5	D0	C5
BYTE CELL	A17	BYTE CELL	C17
FRAME	A18	FRAME	C18
SCLOCK	A21	SCLOCK	C21

The DIGIBUS is capable of transferring data at a rate of 10 Megabytes per second. Our DIGIBUS implementation transfers data one byte at a time. Two bytes are then combined to form one 16-bit sample. The transfer protocol consists of a number of bytes transferred during a particular frame time. All data byte transfers occur under a frame time. A frame time can contain from 1 to 256 samples (512 bytes).

The DIGIBUS contains a strobe signal, Byte Cell, from which data is latched from the local bus into the receiving module. The Byte Cell strobe indicates when valid data is present on the local bus. Data is strobed from both the leading and trailing edges of the Byte Cell signal.

Model V165-xBx1

Multiple local bus sources and acceptors may reside on a single segment of the local bus. Also, several local bus segments may coexist in the same chassis. When multiple sources are present on the local bus, they are preassigned a starting address within the frame to output their data. Once the source has filled its section of the frame, it ceases driving the local bus and allows the other sources to place their data onto the local bus.

If multiple acceptors are on the local bus, each may accept the entire frame of data or a subset of that data. A selection mechanism is provided on the acceptor modules to individually select the samples of interest. Some acceptor modules may also implement circuitry that allows the module to accept every frame, every other frame, every tenth frame, and so on.

V165 DIGIBUS Option

The V165 may be purchased with the DIGIBUS daughter card, the V165-0001. This add-on card gives the V165 the additional functionality of the DIGIBUS as described above. The DIGIBUS implementation on the V165-0001 includes both a local bus source and a local bus acceptor. The acceptor circuitry can receive local bus data from ADCs and Discrete I/O modules that implement the DIGIBUS protocol. The local bus source module can be used to send data to the Discrete I/O module and any other device that accepts data from the DIGIBUS.

DIGIBUS Acceptor Functions

The V165-0001 implements the full DIGIBUS acceptor protocol as established by KineticSystems. This includes the full input selection circuitry, frame counter, frame address register, and a 32 Ksample circular buffer for holding the received data. The following discusses the acceptor functions implemented by the V165-0001.

Before the V165-0001 can accept any data from the DIGIBUS, the source module and the V165-0001 must be configured. Before the V165-0001 is configured, it must be determined which samples the DSP is interested in accessing. The Frame Selection Words, accessible through VME only, are loaded with the required data to enable only the samples that the DSP needs to access. Please refer to the Frame Selection Words section of this manual for further information. For example, if the DSP only requires samples 1 through 16 in a 256 sample frame, set the first Frame Selection Word to FFFF Hex and the rest of the words to zero. Only the 16 samples are then accepted by the V165-0001.

If the DSP must receive all 256 samples from a frame and do intense computations on that data, it may be necessary to reduce the number of frames that are stored. This can be accomplished by loading the Frame Interval register with the required value. A value of zero in this register instructs the selection mechanism to store every frame. If the frame interval is set to one, every other frame of data is stored. The valid range of data for this selection is from 0 to 255.

Once the selection criteria has been loaded into the Frame Selection Words and the Frame Interval register, the DSP may then enable the capturing of data from the DIGIBUS. The ENABLE bit in the Frame Interval register may then be set to a one to start capturing data from the DIGIBUS. Data storage may not occur at the moment the ENABLE bit is set. The V165-0001 must first wait until the beginning of a frame is detected to ensure that the capture

circuitry is synchronized with the DIGIBUS. Storage of data always starts at address location zero of the 32Ksample circular buffer following the enable of the acceptor.

After the raw DIGIBUS data passes through the selection filters, it is written into a FIFO buffer. The FIFO provides a buffering mechanism to "hold" DIGIBUS data samples while the 32Ksample circular buffer is being read by the DSP. Data acquired is then transferred to the 32Ksample circular buffer for readout by the DSP.

When the beginning of a frame is detected by the acceptor circuitry, a flag is inserted into the circular buffer. This flag appears as the most significant bit (bit 31) in the buffer. This flag allows the DSP to keep track of frame boundaries within the buffer. A frame address register latches the address at which the frame boundary was detected. This allows the DSP to read the start address of the most current frame of data in the circular buffer.

A frame counter is provided to keep track of the number of frames that have been stored in the circular buffer. This counter is only incremented after the entire frame of data has been stored in the buffer. The DSP must manually decrement this counter as frames of data are read from the buffer. This counter also provides a benchmark concerning the DSP ability to keep up with the incoming data stream. If the counter rolls over during the processing of data, the DSP is not able to keep up with the incoming stream. This results in the overwriting of data in the circular buffer before it could be processed.

Access to the 32Ksample circular buffer can be as short as 150 nanoseconds (2 wait states) or as long as 300 nanoseconds (5 wait states). If the DSP accesses the circular buffer before a request is made by the DIGIBUS acceptor circuitry to store data, the cycle completes with two wait states. If the DSP accesses the circular buffer a moment later than the DIGIBUS acceptor circuitry, the DSP cycle completes with 5 wait states. In some applications it may be desirable to allow the DIGIBUS data to backup in the FIFO and allow the DSP to have exclusive access to the circular buffer, resulting in 2 wait state accesses. The only potential problem with this approach is the overflowing of the FIFO buffer. The FIFO buffer can "hold" 512 samples. Since the DIGIBUS may operate at 10 Megabytes per second, it does not make sense to stop the transfer of data from the FIFO to the circular buffer because the FIFO would overflow quickly. In most applications, the DIGIBUS only operates for small periods of time at the 10 Megabyte per second rate. In these instances, the user may disable the FIFO to circular buffer transfers for a short period of time while the DSP has exclusive access to the buffer.

The DSP may disable/enable the FIFO to circular buffer by executing write operations to the Enable FIFO-To-Buffer and Disable FIFO-To-Buffer addresses. If a FIFO overflow condition does occur, an interrupt is generated to the DSP at interrupt level 2 (INT2). If this interrupt is generated, do not disable the FIFO-To-Buffer transfers or reduce the number of frames or samples that are stored in the circular buffer.

DIGIBUS Source Functions

The V165-0001 implement a very simple DIGIBUS source protocol. The entire source protocol is controlled by the DSP. Various register addresses are used to start a frame, terminate a frame, and output a byte cell. This source function is primarily used for sourcing digital outputs from the Discrete I/O module. The following lists the various commands available to the DSP to cycle the DIGIBUS.

Start DIGIBUS Frame
Generate DIGIBUS Byte Cell With Data
Generate DIGIBUS Byte Cell Without Data
Stop DIGIBUS Frame

The Start DIGIBUS Frame command merely asserts the DIGIBUS Frame signal, signifying the start of a frame. The Generate DIGIBUS Byte Cell With Data command sources the 16-bits of data associated with the command on the DIGIBUS and asserts the Byte Cell strobe signal. The Generate DIGIBUS Byte Call without data asserts the Byte Cell strobe signal without supplying and data. This command accommodates other DIGIBUS source modules that may be required to place their data on the bus for a given frame. The Stop DIGIBUS Frame command terminates the frame on the DIGIBUS.

DIGIBUS Acceptor Control Registers

This section describes the registers used to control the acceptor circuitry on the V165-0001. The following chart shows the registers and the addresses used to control the DIGIBUS acceptor circuitry.

804000	Read-only	Read Frame Counter
804000	Write-only	Decrement Frame Counter
804005	Read-only	Read Frame Address
80400D	Write-only	Enable DIGIBUS FIFO-To-Buffer Transfers
80400E	Write-only	Disable DIGIBUS FIFO-To-Buffer Transfers

Read Frame Counter

The Read Frame Counter address is used to return the number of frames that have been stored in the 32 Ksample circular buffer. This counter is held cleared as long as the acceptor capture enable bit is cleared in the Frame Interval register. When capturing is enabled, this counter is incremented when a frame of data has been stored in the circular buffer. This counter is decremented when the DSP executes a write operation to the Decrement Frame Counter address. The following chart shows the bit layout of the Frame Counter.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCT15	FCT14	FCT13	FCT12	FCT11	FCT10	FCT09	FCN08	FCT07	FCT06	FCT05	FCT04	FCT03	FCT02	FCT01	FCT00

Bit(s)	Mnemonic	Meaning
31-16	Not used	These bits are not used and read as zeros.

15-00	FCT15-FCT00	These read-only bits reflect the current number of DIGIBUS frames that are remaining in the 32 Ksample circular buffer.

Decrement Frame Counter

The Decrement Frame Counter address is used to decrement the Frame Counter after a frame of data has been read from the 32 Ksample circular buffer. Any data pattern may be used when writing to this address.

Read Frame Address

The Read Frame Address register is used to hold the start address of the last frame of data entered into the circular buffer. This latch is updated whenever the start of a new frame is detected. The address represented in this register is not the physical address of the entry. This value is the offset within the circular buffer that the entry exists. The following shows the bit layout for this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	SPA14	SPA13	SPA12	SPA11	SPA10	SPA09	SPA08	SPA07	SPA06	SPA05	SPA04	SPA03	SPA02	SPA01	SPA00

Bit(s)	Mnemonic	Meaning
31-15	Not used	These bits are not used and read as zeros.
14-00	SPA14-SPA00	These read-only bits reflect the start address of the most recent frame of data entered into the circular buffer.

Enable DIGIBUS FIFO-To Buffer Transfers

A DSP write operation to address 80400D Hex enables the acceptor circuitry to transfer data received from the DIGIBUS FIFO to the circular buffer. Any data pattern may be used when writing to this address.

After power-up of the V165, the FIFO-To-Buffer transfers are disabled.

Disable DIGIBUS FIFO-To-Buffer Transfers

A DSP write operation to address 80400E Hex disables the acceptor circuitry from transferring data from the DIGIBUS FIFO to the circular buffer. Any data pattern may be used when writing to this address.

Model V165-xBx1

DIGIBUS Source Control Registers

This section describes the registers used to control the source circuitry on the V165-Bxx1. The following chart shows the registers and the addresses used to control the DIGIBUS source circuitry.

804009	Write-only	Generate DIGIBUS Byte Cell With Data
80400A	Write-only	Generate DIGIBUS Byte Cell Without Data
80400B	Write-only	Start DIGIBUS Frame
80400C	Write-only	Stop DIGIBUS Frame

Generate DIGIBUS Byte Cell With Data

A DSP write operation to address 804009 Hex initiates a data transfer on the DIGIBUS with the 16-bits of data associated with this command.

Generate DIGIBUS Byte Cell Without Data

A DSP write operation to address 80400A Hex initiates a data transfer on the DIGIBUS, but the V165-Bxx1 does not supply the data. This command is used to supply space for other DIGIBUS source modules present on the segment. Any data pattern can be used with this write operation.

Start DIGIBUS Frame

A DSP write operation to address 80400B Hex causes the DIGIBUS source circuitry to assert the Frame signal on the local bus. This indicates to other devices on the local bus segment that a new frame is starting.

Stop DIGIBUS Frame

A DSP write operation to address 80400C Hex causes the DIGIBUS source circuitry to negate the Frame signal on the local bus.

TTL Input Connector Signal Allocation		
Pin 01 -- Ground		Pin 26 -- Ground
Pin 02 -- Ground		Pin 27 -- INP 15
Pin 03 -- INP 31		Pin 28 -- INP 14
Pin 04 -- INP 30		Pin 29 -- INP 13
Pin 05 -- INP 29		Pin 30 -- INP 12
Pin 06 -- INP 28		Pin 31 -- Ground
Pin 07 -- Ground		Pin 32 -- Ground
Pin 08 -- Ground		Pin 33 -- INP 11
Pin 09 -- INP 27		Pin 34 -- INP 10
Pin 10 -- INP 26		Pin 35 -- INP 09
Pin 11 -- INP 25		Pin 36 -- INP 08
Pin 12 -- INP 24		Pin 37 -- Ground
Pin 13 -- Ground		Pin 38 -- Ground
Pin 14 -- Ground		Pin 39 -- INP 07
Pin 15 -- INP 23		Pin 40 -- INP 06
Pin 16 -- INP 22		Pin 41 -- INP 05
Pin 17 -- INP 21		Pin 42 -- INP 04
Pin 18 -- Ground		Pin 43 -- Ground
Pin 19 -- Ground		Pin 44 -- Ground
Pin 20 -- Ground		Pin 45 -- INP 03
Pin 21 -- INP 19		Pin 46 -- INP 02
Pin 22 -- INP 18		Pin 47 -- INP 01
Pin 23 -- INP 17		Pin 48 -- INP 00
Pin 24 -- INP 16		Pin 49 -- Ground
Pin 25 -- Ground		Pin 50 -- Ground

Model V165-xBx1

Serial Port Connector Signal Allocation		
Pin 01 -- Ground		Pin 26 -- Ground
Pin 02 -- Ground		Pin 27 -- CLKR1
Pin 03 -- XF1		Pin 28 -- CLKR0
Pin 04 -- XF0		Pin 29 -- DR1
Pin 05 -- Ground		Pin 30 -- DR0
Pin 06 -- Ground		Pin 31 -- FSR1
Pin 07 -- TCLK1		Pin 32 -- FSR0
Pin 08 -- TCLK0		Pin 33 -- Ground
Pin 09 -- Ground		Pin 34 -- Ground
Pin 10 -- Ground		Pin 35 -- Reserved
Pin 11 -- Reserved		Pin 36 -- Reserved
Pin 12 -- Reserved		Pin 37 -- Reserved
Pin 13 -- Reserved		Pin 38 -- Reserved
Pin 14 -- Reserved		Pin 39 -- Reserved
Pin 15 -- Reserved		Pin 40 -- Reserved
Pin 16 -- Reserved		Pin 41 -- Reserved
Pin 17 -- Reserved		Pin 42 -- Reserved
Pin 18 -- Reserved		Pin 43 -- Reserved
Pin 19 -- CLKX1		Pin 44 -- Reserved
Pin 20 -- CLKX0		Pin 45 -- Reserved
Pin 21 -- DX1		Pin 46 -- Reserved
Pin 22 -- DX0		Pin 47 -- Reserved
Pin 23 -- FSX1		Pin 48 -- Reserved
Pin 24 -- FSX0		Pin 49 -- Reserved
Pin 25 -- Ground		Pin 50 -- Reserved

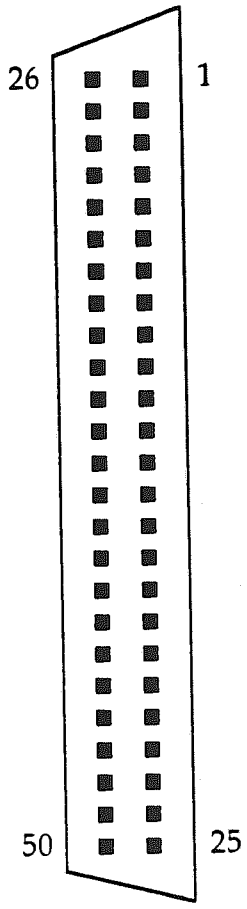


Figure 1 - V165 50-Position SCSI-II Plug

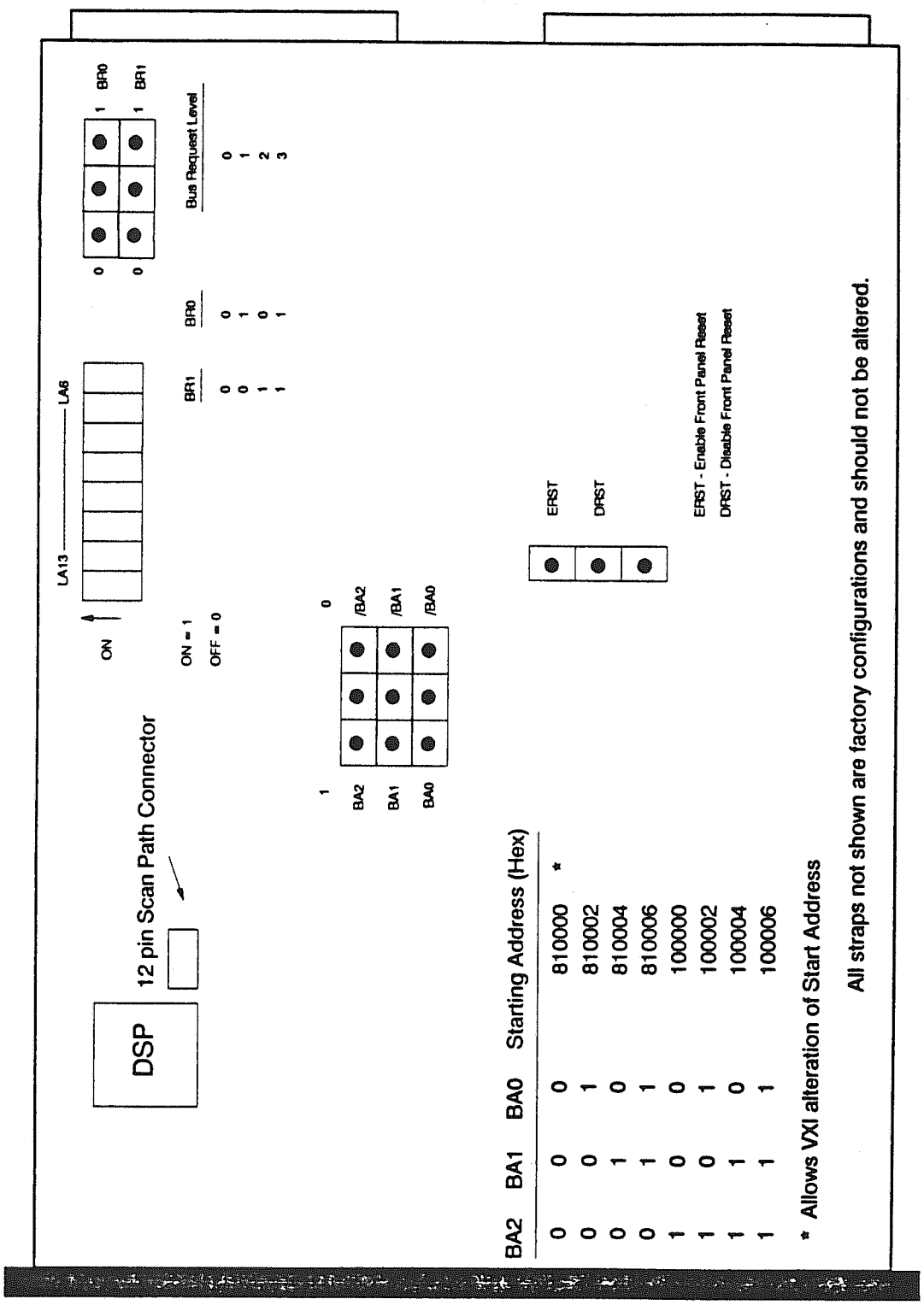


FIGURE 2 - V165 STRAP LOCATIONS

APPENDIX

BOOTROM8: THE V165 DSP ROM MONITOR

This Appendix provides the information users will need to interface with the ROM monitor program on KSC's V165 DSP module. This firmware uses a 1,028 word area of DRAM to receive commands and return status and data. This DRAM can also be accessed in A32 space by any VXIbus module. Generally, the Slot 0 controller will be communicating with the V165 using this DRAM, although it is possible for other modules to do so.

Throughout this document, the DRAM area used by BOOTROM8 is referred to as the command record. Regardless of the amount of DRAM on the V165 it will be at the same offset from the start of V165 DRAM in A32 space: 20 hex. (The DRAM base address in A32 space varies with the DRAM size. On a V165 with 1 Megabyte of DRAM installed, the base address is 0010_0000 hex, while it is 0100_0000 hex when 16 Megabytes are installed.) The command record is 4,112 bytes long, but because the TMS320C30 DSP on the V165 can only address 32-bit long words, the record should be considered as 1,028 long words. Its layout varies somewhat depending upon the particular command, but the first two words are always the command code word and the "done flag" status word. Following these are one or more long words of data, in which most commands have an address word and at least one data word.

The format used by each command is described in detail later in this document, using excerpts from the V165RCRD.H header file. The complete file is found at the end of this Appendix. Users writing C code to interface to BOOTROM8 should include that file and make use of the manifest constants #define-d there. This will simplify code maintenance in the event that KineticSystems upgrades the V165 BOOTROM monitor.

TYPE DEFINITIONS

The command records are built using three types: an enumeration type called CMD_TYPE, another called FLAG, and the unsigned long integer type which is a 4-byte quantity #define-d as ULONG:

```
#define ULONG       unsigned long
```

The first long word of the command record may be given any of the following values to indicate which command must be performed by the V165 DSP:

NO_COMMAND
WRITE_LEDS
WRITE_ADDRESS
READ_ADDRESS
FILL_MEMORY
SELF_TEST
V165_001_TEST
RECEIVE_DATA
SET_NEW_PC
GET_VERSION

Each of these commands, and the unique command record entries, are detailed later in this document. (The only exception is NO_COMMAND, which is a "placeholder" value.) If a value greater than the last enumeration value (in this case, GET_VERSION) is stored in the command word, the BOOTROM8 monitor will return INVALID_COMMAND in the Done Flag word.

The second long word of the command record is located at an offset of 24 hex from the base of V165 DRAM. It serves a dual purpose: the user code writes the value CLEAR to this word, the DONE_FLAG, to indicate that a new command record is available for the V165. Upon completion of the command, BOOTROM8 writes one of the remaining values back to this word to indicate the status of the operation:

CLEAR
SET
INVALID_COMMAND
INVALID_ADDRESS
INVALID_COUNT
TEST_FAILED

"Set" indicates that the command is done and was successful; any data from the V165 has been written to the designated portion of the command record. The remaining enumerations in this FLAG type indicate an error in processing the command. If a command parameter value from the user's code is invalid, the operation itself is not performed.

Proper execution of the commands will result when the user code fills in all IN fields EXCEPT the done_flag first, and only then writes the value CLEAR to the done_flag. User code should not attempt to write any fields to the command record until the done_flag contains a value other than CLEAR. Other recommendations may be found in the Power-On Self-Test section, which follows the Commands listing.

COMMANDS

This section describes each command accepted by BOOTROM8 and details the record structure unique to that command. The HEX offsets listed for each long word of the particular record should be added to the DRAM base address to obtain the VXIbus address for the record. The direction of data transfer for each word follows the name assigned to that word. IN means parameters passed into the V165; OUT means results from the V165 operation.

In general, commands which require a DSP memory address (i.e., not a VXIbus memory address) provide the 24-bit value in the THIRD long word of the command record, at an offset of 28 hex from the base of V165 DRAM. The V165 will reject any address which has any bits set in the most significant byte of the long word (bits 31 to 24). Next comes any data to be passed to BOOTROM8, followed by addresses and/or data to be passed back to the user code, if any.

FILL_DSP_MEMORY_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	data	IN	
	30	ULONG	count	IN	

This command stores the given data at all V165 memory locations from address to address + count - 1, inclusive. It is equivalent to, and more efficient than, a sequence of "count" WRITE_ADDRESS_RECORD commands with incremented addresses and the same fill data. The count parameter must meet the same restriction imposed upon addresses: it may not have any bits set in the most significant byte of the long word (bits 31 to 24). Values which violate this restriction will be flagged with the INVALID_COUNT return code in the done_flag status word of the command record.

READ_ADDRESS_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	data		OUT

This command returns the data found at the specified address in V165 memory. It is possible, therefore, for the Slot 0 controller to have indirect access to all internal V165 and DSP registers and memory. User code on the V165 can perform these memory accesses much more efficiently, but this command allows users to debug their applications on the V165 in the absence of such code.

RECEIVE_DATA_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	word_count	IN	
	30+	ULONG	data[]	IN	

This command copies the array of data into V165 memory locations from address to address + count - 1, inclusive. It is equivalent to, and more efficient than, a sequence of "count" WRITE_ADDRESS_RECORD commands with incremented addresses and the successive data words from the array. The size of the data[] array, and therefore the upper limit on the word_count, is #define-d by the manifest constant MAX_DATA_LEN, which is 1024 long words for BOOTROM8. If word_count exceeds this, the Done flag will be set to INVALID_COUNT.

This command supports code downloading, particularly by the COFF loader available from Texas Instruments. KineticSystems has ported the TI loader to run on a National Instruments VXIcpu-030 (TM) running VxWorks, and can provide the object code on request. See the Downloading section below for more information.

SELF_TEST_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	data_expected		OUT
	30	ULONG	data_received		OUT
	34	ULONG	sram_end_address		OUT
	38	ULONG	dram_end_address		OUT

This command re-runs the Power-On Self-Test of three areas of RAM present on all versions of the V165 (i.e., regardless of memory configuration): Internal 'C30 DSP RAM (2K long words); External Zero Wait State Static RAM (2K long words); and External Slow Dynamic RAM (256K long words). The algorithm used simply verifies the ability to write an arbitrary pattern to each location, and does not perform exhaustive testing of the memory circuits. The advantage of the read-complement-write-compare algorithm used is that the original data is restored, so user programs need not be downloaded again after self-test.

If all the tests pass, the end addresses reflect the amount of external SRAM and DRAM installed on the V165 module. (The DSP RAM is always 2K long.) If a location fails, the expected and actual data are stored in the record, and the Done flag is set to TEST_FAILED.

SET_NEW_PC_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	

This command calls the subroutine at the given address, and provides the mechanism for executing one or more user code modules on the V165's 'C30 DSP. Ideally, the called code should end with a Return from Subroutine (RETS) instruction in assembly language or optionally a return statement in C. See the Downloading and Multiple Tasks sections below for more information.

V165_001_TEST_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	data_expected		OUT
	30	ULONG	data_received		OUT

This command supports the V165-0001 Digibus option daughterboard and runs the memory test for the 32K Sample RAM. It uses the same non-destructive algorithm described above for the SELF_TEST_RECORD command.

If the test passes, the Done flag is "SET." If a location fails, the expected and actual data are stored in the record, and the Done flag is set to TEST_FAILED.

VERSION_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	rom_version		OUT

This command was added in version 8 so that user code can determine what commands the monitor on a V165 will accept. Future versions of BOOTROM will only add commands to the end of the enumeration, and will preserve every command described in this document. Contact KineticSystems for more information on V165 ROM upgrades.

WRITE_ADDRESS_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	address	IN	
	2C	ULONG	data	IN	

This command stores the data at the specified address in V165 memory. It is possible, therefore, for the Slot 0 controller to have indirect access to all internal V165 and DSP registers and memory. User code on the V165 can perform these memory accesses much more efficiently, but this command allows users to debug their applications on the V165 in the absence of such code. Related commands are FILL_DSP_MEMORY_RECORD and RECEIVE_DATA_RECORD.

WRITE_LED_RECORD

offset	20	CMD_TYPE	command	IN	
	24	FLAG	done_flag	IN	OUT
	28	ULONG	data	IN	

This command writes the least-significant byte of the third long word to the V165's Status LED register. This provides a convenient visual aid when establishing communications between the Slot 0 controller and the V165. It also can be used to show the progress of the user's code and aid in debug.

POWER-ON SELF-TEST

As stated above, user code should not attempt to write any fields to the command record until the done_flag contains a value other than CLEAR. At power-up, however, the state of this location is indeterminate and thus user code must be prepared for the distinct possibility that the done_flag will not be initialized properly until after the Power-On Self-Test has completed. KineticSystems recommends that user code not attempt to send any commands within 5 seconds of power or reset being applied to the V165.

DOWNLOADING DSP CODE TO V165 RAM

Any of the three variants of the Write command may be used to download user code into V165 RAM. The most efficient and flexible is the RECEIVE_DATA_RECORD command described above. The user code on the Slot 0 controller will take the 'C30 object code section by section and extract the section header and data. The section data are broken into suitable blocks if necessary, and sent to the V165 command record. Once all the memory sections are downloaded, the SET_NEW_PC_RECORD command is issued with the entry point for the object code as the start address.

The V165 support tool which runs on the VXIcpu-030 splits the download and the execute portions into separate menu options so that the user can verify that the code is stored in memory properly, and can make any changes to constants before the code executes. The tool uses the TI COFF loader, with some V165-specific functions which interface the generic loader to the command record scheme. This tool is available upon request.

INTERRUPT SERVICE & MULTIPLE TASKS

The SET_NEW_PC_RECORD command allows users to run small initialization modules which set up Interrupt Service Routines (ISRs) for the DSP and V165 interrupt-producing hardware. The modules can program the device control registers, write the address of the ISR to the appropriate vector, enable interrupts, and return to BOOTROM8 so other modules can be executed. Timer, VXI, and Digibus interrupts are the most likely candidates for this treatment.

KineticSystems provides an example module for the DSP Timer hardware which users may modify for their own needs. As delivered, the timer example sets up Timer 1 for interrupts every 50 milliseconds, and the ISR simply updates a "racing" pattern on the Status LED. This example also may be used to verify that user code on the Slot 0 controller is properly interfaced to the V165 command record, since it provides a visual indication that the ISR is running.

With one or two timer interrupts, users could execute periodic functions on the DSP, although users might consider a commercial real-time executive like SPOX if the tasks to be performed are in any way interdependent. Interested users should contact KineticSystems to determine the availability of such executives customized for the V165.

COMMAND RECORD HEADER FILE V165RCRD.H

```
#define ROM_VERSION  8L

#define TRUE  1
#define FALSE 0

#define USHORT      unsigned short
#define ULONG       unsigned long
#define ulong       unsigned long
#define ushort      unsigned short

#define MAX_DATA_LEN 1024

typedef enum {
   CLEAR = 0,
   SET,
   INVALID_COMMAND,
   INVALID_ADDRESS,
   INVALID_COUNT,
   TEST_FAILED
} FLAG;

typedef enum {
   NO_COMMAND = 0,
   WRITE_LEDS,
   WRITE_ADDRESS,
   READ_ADDRESS,
   FILL_MEMORY,
   SELF_TEST,
   V165_001_TEST,
   RECEIVE_DATA,
   SET_NEW_PC,
   GET_VERSION
} CMD_TYPE;

typedef struct {
   CMD_TYPE command;
   FLAG     done_flag;
} GENERIC_RECORD;
```

```
typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   data;
} WRITE_LED_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   data;
} WRITE_ADDRESS_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   data;
} READ_ADDRESS_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   data;
  ULONG   count;
} FILL_DSP_MEMORY_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   data_expected;
  ULONG   data_received;
  ULONG   sram_end_address;
  ULONG   dram_end_address;
} SELF_TEST_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   data_expected;
  ULONG   data_received;
} V165_001_TEST_RECORD;
```

```
typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
  ULONG   word_count;
  ULONG   data[ MAX_DATA_LEN ];
} RECEIVE_DATA_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   address;
} SET_NEW_PC_RECORD;

typedef struct {
  CMD_TYPE command;
  FLAG    done_flag;
  ULONG   rom_version;
} VERSION_RECORD;

typedef union {
  GENERIC_RECORD          generic;
  WRITE_LED_RECORD        write_led;
  WRITE_ADDRESS_RECORD    write_address;
  READ_ADDRESS_RECORD     read_address;
  FILL_DSP_MEMORY_RECORD  fill_memory;
  SELF_TEST_RECORD        self_test;
  V165_001_TEST_RECORD    digibus;
  RECEIVE_DATA_RECORD     receive_data;
  SET_NEW_PC_RECORD       set_new_pc;
  VERSION_RECORD          get_version;
} RECORD;

extern void self_test( RECORD * );
extern ULONG ram_test( RECORD *, ULONG *, ULONG *, ULONG );
```

V165 MEMORY MAP

This supplements the hardware memory map found in the V165 instruction manual, and describes the ROM monitor's use of RAM and ROM. All addresses are expressed in hexadecimal notation, and are 24 bits long. Users should note the reserved area from 10041C to 100FFF, which holds the variables used by BOOTROM8. Software downloaded to the V165 should NOT modify the values in any location in this region.

START	END	USE
0	BF	Interrupt Vectors
C0	7FF	Minimum 0-wait-state SRAM (2K longwords)
800	7FFF	MAXIMUM 0-wait-state SRAM (32K longwords)
8000	FFFF	MAXIMUM 1-wait-state SRAM (32K longwords)
100000	100007	4 DRAM Reset Locations (see hardware manual)
100008	10041B	BOOTROM8 Command Record
10041C	100FFF	BOOTROM8 Variables. **DO NOT USE!**
101000	13FFFF	Minimum Slow DRAM (256K longwords)
140000	4FFFFF	Maximum Slow DRAM (4Meg longwords)
809800	809EFF	DSP Internal 0-wait-state SRAM
809F00	809FFF	BOOTROM8 Stack
810000	810007	4 EPROM Reset Locations (see hardware manual)
810008		EPROM Checksum Start Pointer
810009		EPROM Checksum End Pointer
81000A	81023C	BOOTROM8 Code & Constants
81023D		EPROM Checksum
81023E	817FFF	Maximum EPROM (32K longwords)