ProDAQ

User Manual

ProDAQ 3040

VME64x to C-Size VXI Adapter

PUBLICATION NUMBER: 3040-XX-UM-0101

PUBLICATION DATE: 01-JUN-2004

Bustec Production, Ltd.
World Aviation Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

bustec

PROPRIETARY NOTICE

Table of Contents

1. THEORY OF OPERATION 5

1.1 GENERAL 5
1.2 VME-VXI MASTER CHANNEL 5
1.3 VXI-VME SLAVE CHANNEL 5
1.4 VXI TO VME ADDRESS SPACE MAPPING 7
1.5 VXI-TO-VME INTERRUPT FORWARDING 11
1.6 VME-TO-VXI INTERRUPT FORWARDING 11
1.7 UTILITY BUS 12
1.8 AUTO SLOT-0 DETECTION AND MODID/CLK10 12
1.9 AUTO SYSTEM CONTROLLER DETECTION 13

2. INSTALLATION AND CONFIGURATION 15

2.1 PREPARING THE VME BOARD 15
2.2 CONFIGURING THE PRODAQ 3040 ADAPTER 15
2.2.1 Logical Address 16
2.2.2 VXI to VME Address Space Mapping 17

3. REGISTER DESCRIPTION 21

3.1 ADDRESS MAP AND REGISTERS 21
3.2 REGISTER DETAILS 22
3.2.1 ID Register 22
3.2.2 LogAdr 22
3.2.3 DevType 22
3.2.4 Status 23
3.2.5 Control 24
3.2.6 Offset 24
3.2.7 MODID 24
3.2.8 VMEOffset 25
3.2.9 VXIControl 25
3.2.10 VMEControl 26
3.2.11 EEPROMData 26
3.2.12 EEPROMCtrl 27
3.2.13 TrigStatus 27
3.2.14 TrigIntMask 27
3.2.15 TrigControl 29
3.2.16 TrigIntMode 29
3.2.17 IRQDir 30
3.2.18 SerNumHigh 30
3.2.19 SerNumLow 30

APPENDIX A. EEPROM PROGRAMMING FLOW CHARTS 31

1. Theory of Operation

The ProDAQ 3040 6U VME64x to C-Size VXIbus Adapter allows the usage of 6U VMEbus boards in a C-Size VXIbus system. It translates VMEbus cycles into VXIbus cycles and vice versa. In addition it houses the extensions necessary for VXIbus devices, as there are the configuration registers, a trigger and extended interrupt interface, MODID support and the 10 MHz clock generation.
It forwards all VME master cycles transparently to the VXIbus, allowing a VMEbus master the full access to the VXIbus. On the VXIbus it allows the full integration of the module in the VXIbus resource management by providing a set of VXIbus compatible configuration registers and a configurable translation window in the VXIbus A24 or A32 address space. Accesses to this translation window are forwarded to the VMEbus module's A16, A24, A32 or CR/CSR space.
Programmable interrupt, trigger and MODID support (both detection and generation), 10 MHz clock generation and auto-Slot0 detection allow the board to be used as an embedded controller in Slot-0 and non-Slot-0 applications.

1.1 General

The ProDAQ 3040 adapter can be placed in both slot-0 and non-slot-0 positions and accepts both master and slave VME boards. It forwards all VME master cycles transparently to the VXIbus through the VME-VXI master channel, and translates accesses to the configured address range through the VXI-VME slave channel.

1.2 VME-VXI master channel

The cycles generated by the VME master are directly forwarded to the VXI bus. Both address modifier and addresses stay unchanged during the conversion of the cycle. The cycles are forwarded at full speed and the only delay is caused by the propagation delay of the buffers. For the VME-VXI master channel the following cycles are supported:

A16: D08(EO), D16, D32
A24: D08(EO), D08(EO)BLT, D16, D16BLT, D32, D32BLT, MBLT
A32: D08(EO), D08(EO)BLT, D16, D16BLT, D32, D32BLT, MBLT

1.3 VXI-VME slave channel

The VXI-VME slave channel allows the accesses to the on-board register resources and to the resources of the VME board. The cycles performed in the A16 address space are directed to the adapter's registers. The cycles performed in the A24 or A32 are mapped to the selected address space of the VMEbus. The following cycles are supported:

A16: D08(EO), D16, D32
A24: D08(EO), D08(EO)BLT, D16, D16BLT, D32, D32BLT, MBLT
A32: D08(EO), D08(EO)BLT, D16, D16BLT, D32, D32BLT, MBLT

The base address of the VXIbus configuration registers in A16 address space is set by the logical address switch or is assigned dynamically. The base address and size of the window in A24 or A32 is configured as follows:

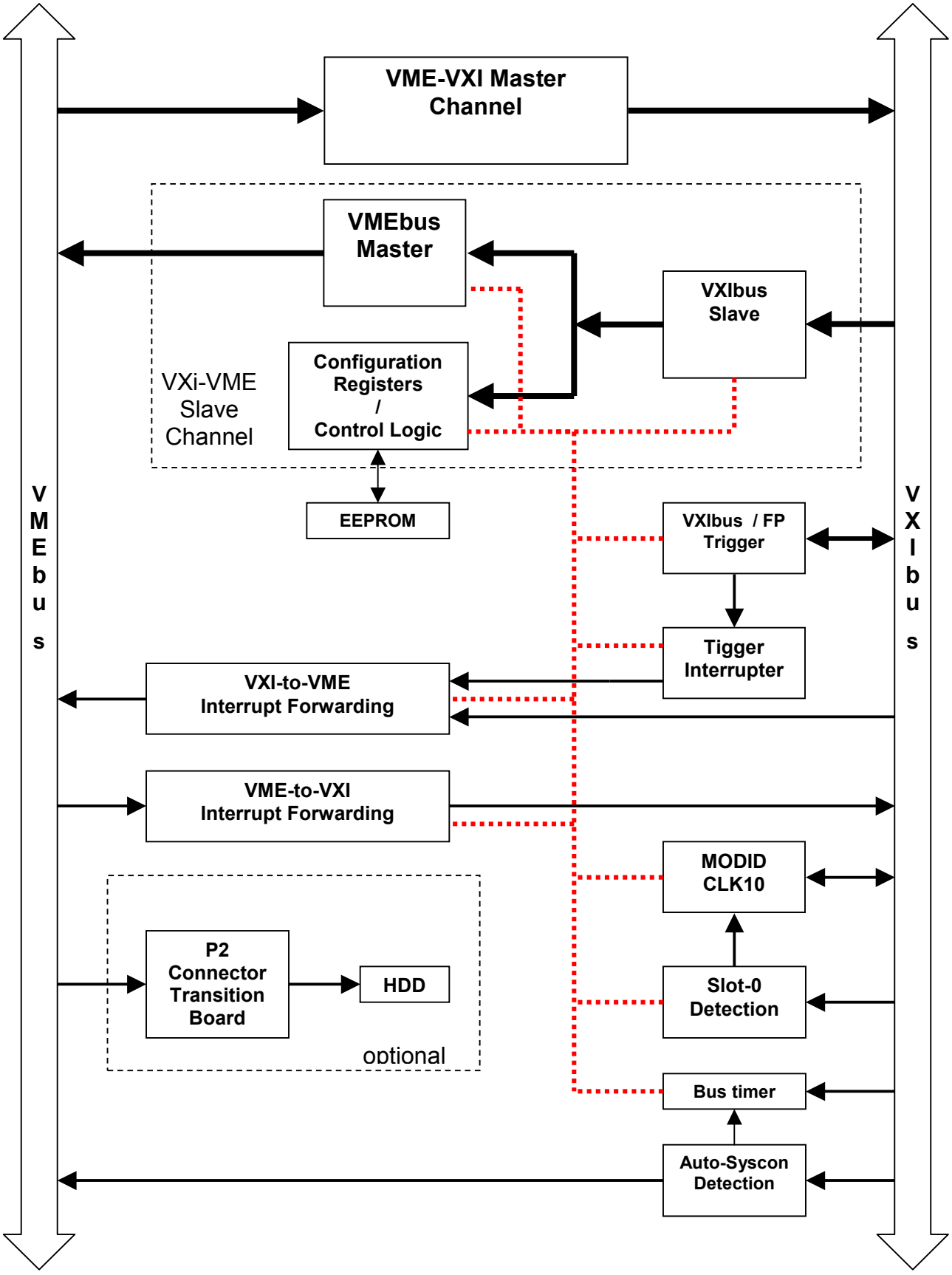


Figure 1 - ProDAQ 3040 Block Diagram

1.4 VXI to VME address space mapping

The ProDAQ 3040 adapter is a register based VXIbus device. It features the standard set of VXIbus configuration registers and can be configured to decode additionally an address range in the VXIbus A24 or A32 address range. Accesses to this address range are translated into a configurable address range on the VMEbus.

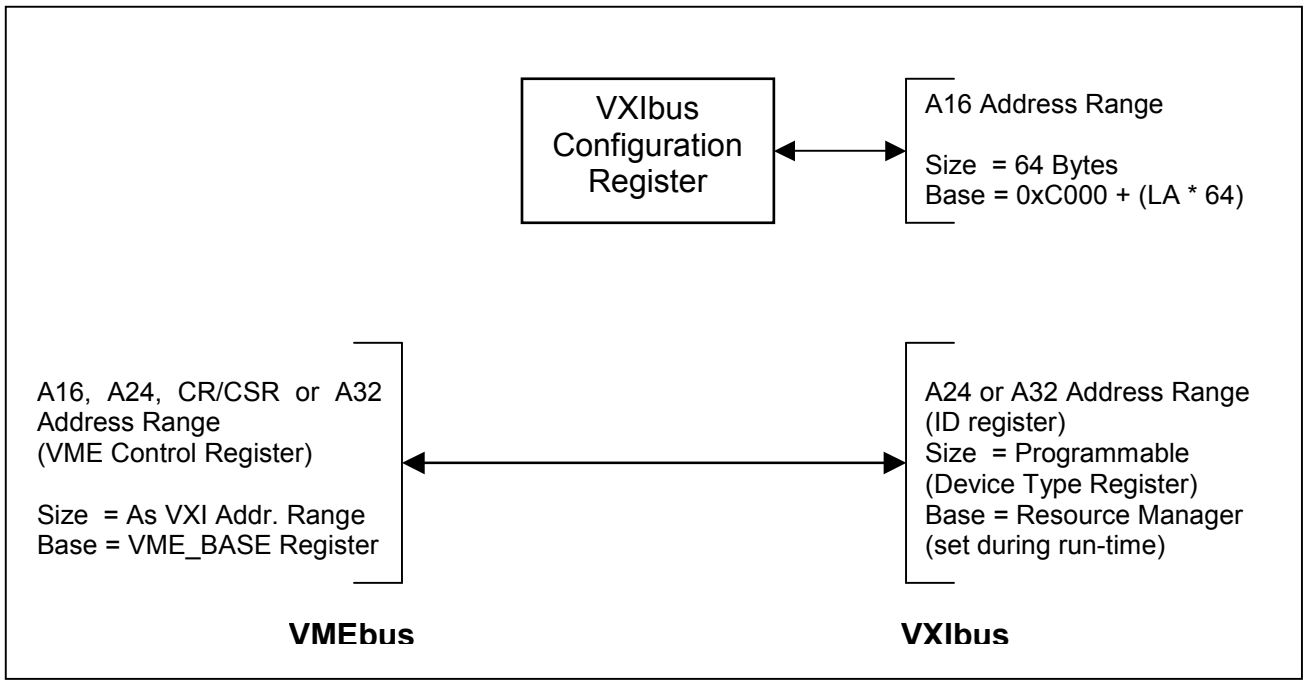


Figure 2 - Available Address Ranges

The address range for the access to the VXIbus configuration registers is 64 bytes in size and its base address is either set by statically the logical address switch or dynamically by the resource manager.
The configuration registers contain the information about the address range (on the VXIbus) of the module to be used by the resource manager. On the ProDAQ 3040 this information is programmable and stored in an EEPROM and is also used during hardware initialization to configure the VXI-to-VME address mapping.
The **ID** register at offset 0 contains the information, whether the board is set for A16 only, A16/A24 or A16/A32 address space. If the board is programmed for A16 only, the VXI-to-VME address mapping is turned off and only the access to the configuration registers in the A16 address space is possible. If it is programmed for A16/A24 address space, the address mapping is enabled for mapping a range in the A24 VXIbus address space to the specified VME address space. If the board is set for A16/A32 address space, the address mapping is enabled for mapping a range in the A32 VXIbus address space to the specified VME address space.
The **Device Type** register at offset 2 in the VXIbus configuration register contains the information of the memory size a VXIbus module requires. This information is used by the VXIbus resource manager to arrange the different memory ranges of the VXIbus boards in a system in the available space. It also determines the size of the range that is mapped to the VME address space.
The **Offset** register determines the base address of the decoded VXIbus A24 or A32 address range and is set during run-time by the resource manager.

The **VME Control** register contains the information, which address range on the VMEbus the accesses to the configured VXIbus address range should be mapped to. The **VME Offset** register determines the base address of the VMEbus address range.

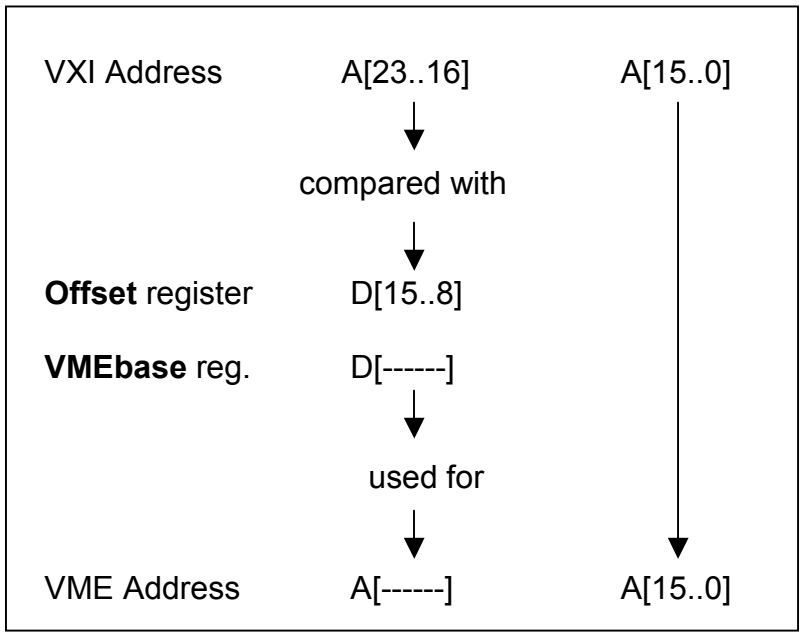


Figure 3 - VXIbus A24 to VMEbus A16 mapping

Figure 3 shows the VXIbus A24 to VME A16 mapping. The only usable size of the mapped range is 64k due to the minimum memory space size in A24 as defined by the VXIbus standard.

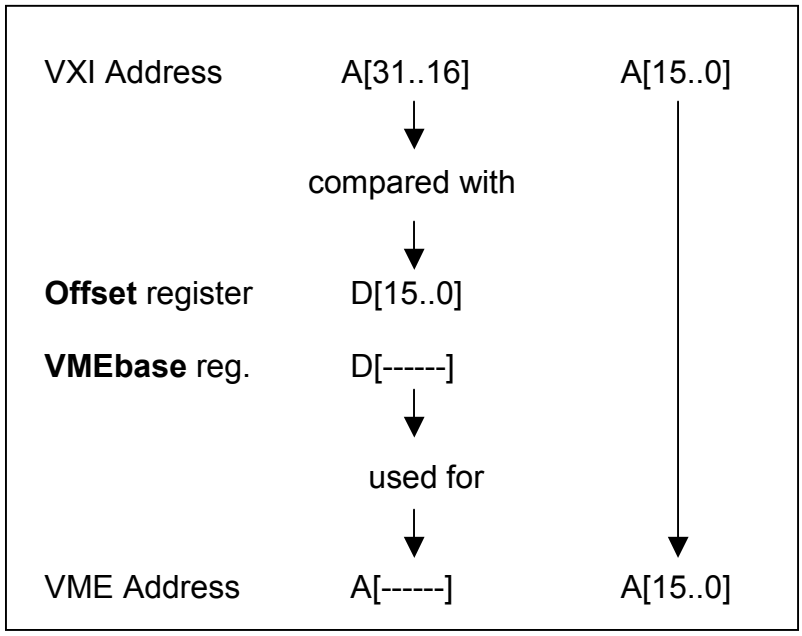


Figure 4 - VXIbus A32 to VMEbus A16 mapping

Figure 4 shows the VXIbus A32 to VMEbus A16 mapping. As before, the only usable size of the mapped range is 64k due to the minimum memory space size as defined by the VXIbus standard.

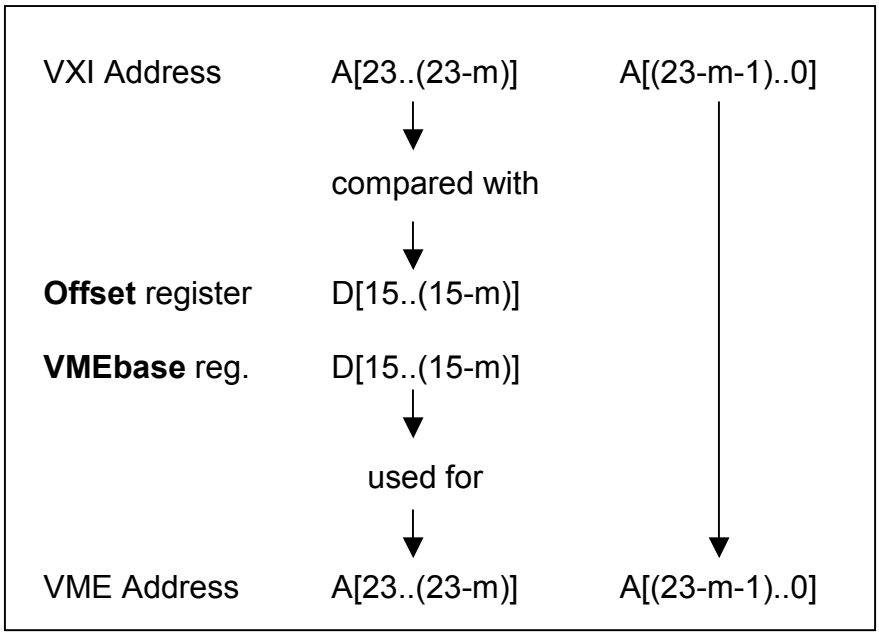


Figure 5 - VXIbus A24 to VMEbus A24 mapping

Figure 5 shows the VXIbus A24 to VMEbus A24 mapping. The size of the mapped range can be 64 kByte to 8 MByte (m = 7..0).

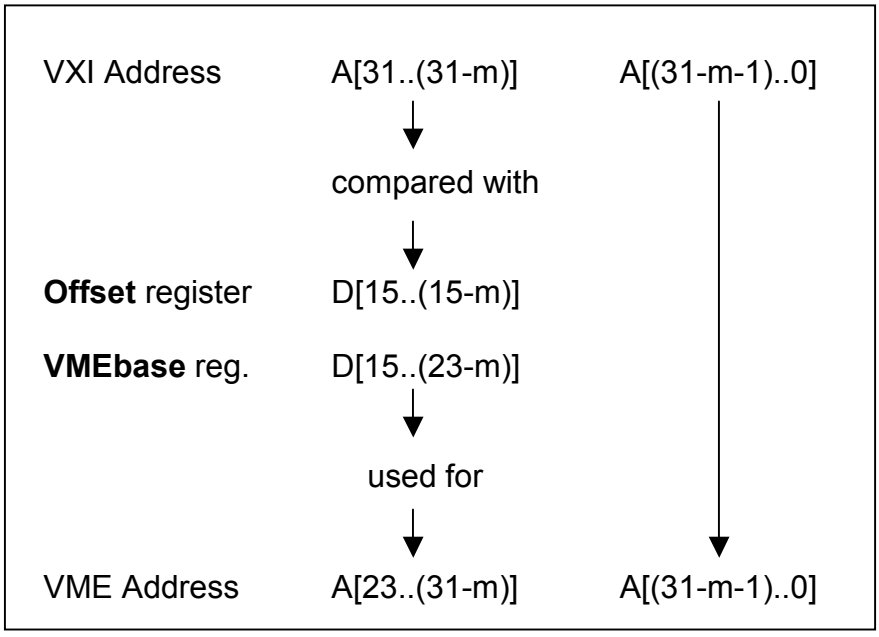


Figure 6 - VXIbus A32 to VMEbus A24 mapping

Figure 6 shows the VXIbus A32 to VMEbus A24 mapping. The size of the mapped range can be 64 kByte to 16 MByte (m = 15..7). If it is 16 MByte, the content of the VMEbase register is not used, because the whole A24 address range is mapped.

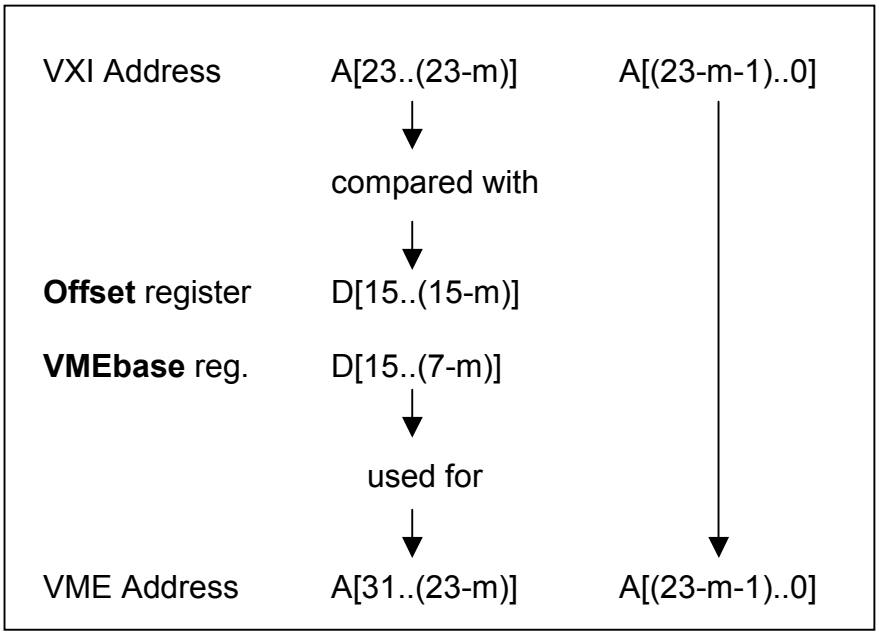


Figure 7 - VXIbus A24 to VMEbus A32 mapping

Figure 7 shows the VXIbus A24 to VMEbus A32 mapping. The size of the mapped range can be 64 kByte to 8 MByte (m = 7..0).

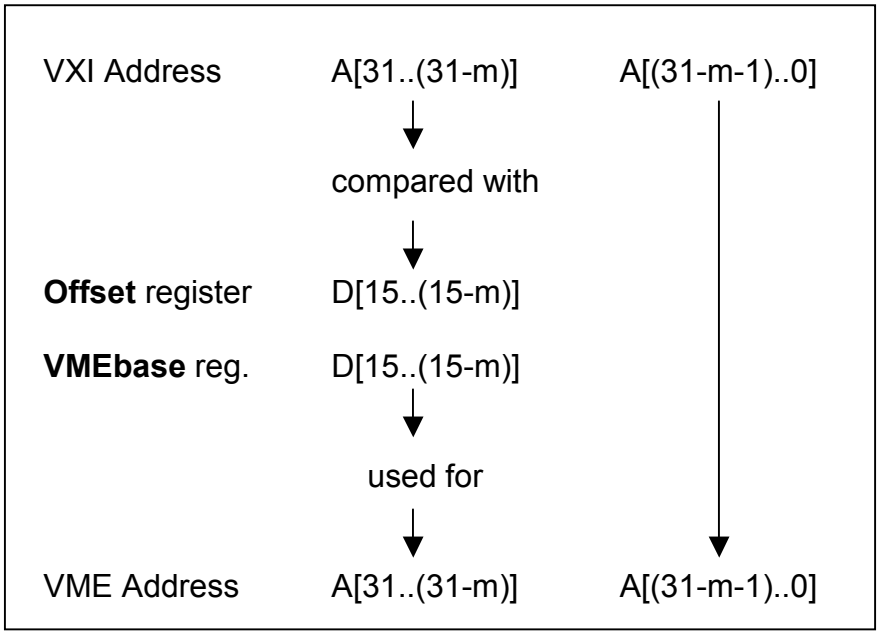


Figure 8 - VXIbus A32 to VMEbus A32 mapping

Figure 8 shows the VXIbus A32 to VMEbus A32 mapping. The size of the mapped range can be 64 kByte to 2 GByte (m = 15..0).

1.5 VXI-to-VME Interrupt Forwarding

The VXI-to-VME interrupt forwarding concerns the situation when the interrupt handler is located on the VME board and interrupters are located on the VXIbus (including the adapter's own interrupter). In this situation there is a need to send an interrupt that happened on the VXIbus to the VMEbus. The adapter maps directly the VXIbus interrupts to the VMEbus interrupts (VXI IRQ1 to VME IRQ1, VXI IRQ2 to VME IRQ2, etc).
Before the interrupt signal can be forwarded from the VXIbus to the VMEbus, it has to be enabled by using the IRQDIR[7..1] bits from **IRQDir** register. Only the interrupts, which are handled by the VME Interrupt Handler, have to be forwarded and only these interrupts should be enabled using IRQDIR bits.
Most of the VME Interrupt Handlers generate D08(O) cycles during interrupt acknowledge cycle but D16 and D32 are allowed by the VME64 standard as well. On the other side the VXI standard specifies the status/ID of up to 32-bit width and every interrupter is forced to respond to any of D08, D16 or D32 cycles with the appropriate status/ID width.
To fulfill the VXI and VME standard requirements, the following scheme has been implemented in the VXI-to-VME interrupt forwarding channel: the VME interrupt handler can initiate any of the three cycles D08(O), D16 or D32. In response to any of these cycles the interrupt forwarder will always generate a D32 cycle on the VXIbus. When the interrupters status/ID is ready, only the part will be forwarded to the VMEbus, which is requested by the handler (byte, word or long word).

The status/ID forwarded to the VME side is defined in the following way:

Status/ID bits forwarded from VXI to VME bus	If VME Interrupt Handler is
D[7..0]	D08(O)
D[15..0]	D16
D[31..0]	D32

Independently of this the whole 32-bit status/ID is latched in the adapter's memory and can be read by the ISR. There is a separate location for every interrupt level to store the status/ID. Forwarding the interrupt for the given level becomes disabled after the status/ID is latched during interrupt acknowledge cycle. It can be only enabled by reading out the latched status/ID.

1.6 VME-to-VXI Interrupt Forwarding

The VME-to-VXI interrupt forwarding concerns the situation when the interrupt handler is located on the VXIbus and the interrupter is located on the VMEbus. In this situation there is a need to forward an interrupt that happened on the VMEbus to the VXIbus. The adapter maps directly the VMEbus interrupts to the VXIbus interrupts (VME IRQ1 to VXI IRQ1, VME IRQ2 to VXI IRQ2, etc).
Before the interrupt signal can be forwarded from the VMEbus to the VXIbus it has to be enabled using IRQDIR[7..1] bits from **IRQDir** register. The VXI Interrupt Handler generates D16 or D32 cycles during interrupt acknowledge cycle. On the other side the VME standard specifies that every interrupter (D08, D16 or D32) is forced to respond to D32 cycle with the appropriate status/ID width.

To fulfill the requirements, the following scheme has been implemented for the VME-to-VXI interrupt forwarding channel: the VXI interrupt handler can initiate either D16 or D32. In the response to any of these cycles the interrupt forwarder will always generate D32 cycle on the VMEbus. The status forwarded to the VXI side will be 16-bit always and will be composed of the adapter's logical address and the eight lower bits of the VME status/ID.

The status/ID forwarded to the VXI side will always be 16-bit and is defined in the following way:

Forwarded 16-bit VXIbus Status/ID	Is composed of	Description
D[7..0]	LA[7..0]	Adapter's Logical Address
D[15..8]	D[7..0] (VME)	Lower 8 bits of VME Status/ID

The VME-to-VXI interrupt forwarding will not become disabled after the status/ID is latched during interrupt acknowledge cycle.

1.7 Utility Bus

The ProDAQ 3040 Adapter allows generating either a soft reset or a VXIbus SYSRESET.
The soft reset is generated using the RESET bit in the **Control** register. It restarts the adapters self test and disables the MODID drivers.
The VXIbus SYSRESET is generated by using the SYSRESET bit in the **VXIcontrol** register and is send to the VXIbus only to allow to reset the VXIbus under software control.

The SYSCLK isignal s taken from the VME board if the module is system controller (see below). The SYSCLK signal is not generated on the adapter. If the module is not the system controller, then the SYSCLK is propagated from the VXIbus to the VMEbus.

The SYSFAIL driver is implemented as a bi-directional signal driver that allows the VME board to monitor and drive this line. The ACFAIL signal is forwarded from the VXIbus to the VMEbus so that the VME board can monitor this signal.

1.8 Auto Slot-0 Detection and ModID/CLK10

The adapter automatically detects whether he is placed in slot 0 (the leftmost slot in a VXIbus mainframe) and enables or disables the CLK10 and MODID lines accordingly. The CLK10 signal can be configured to be either generated from the on-board oscillator or to be sourced from an external signal via the front panel connector. The front-panel input for the CLK10 signal accepts a standard TTL signal.

1.9 Auto System Controller Detection

The adapter supports auto system controller detection on the VXIbus. The result is then carried forward to the VMEbus to allow auto system controller detection on the VME board. Except of the bus timer the adapter does not implement any functionality of a system controller on its own. It only carries the system controller functionality from the VME board to the VXIbus, if the adapter is placed in the system controller slot.

2. Installation and Configuration

2.1 Preparing the VME Board

To use a VME board in a VXIbus system, simply connect the board to the adapter using the P1 and P2 connector of the VME board and the female connectors J300 and J301 of the ProDAQ 3040 Adapter.

The adapter supports automatic system controller detection on the VME side. None the less this feature might not work with some boards, so that it could be necessary to configure the VME board to be fixed either system controller or not, depending on the usage in the VXIbus system.

If the VME board features a VME slave for registers or local memory, the adapter must be set up to support a VXI to VME address translation window. Please refer to section 2.2.2 how to configure the adapter for this. The adapter supports, in accordance to the VXI standard, one window, which can translate a A24 or A32 access on the VXI side to a A16 or A24 or A32 access on the VME side (see section 1.3 and 1.4).

2.2 Configuring the ProDAQ 3040 Adapter

The ProDAQ 3040 Adapter can be configured by writing to the VXIbus configuration registers located in A16 address space of the VXIbus(see section 3). Do be able to write to those registers, the adapter must be configured either to use a static logical address or the resource manager must be used to assign dynamically a logical address to the adapter (see section 2.2.1 for details). Any master on the VXIbus, as well as a VME master located on the adapter (because of the transparent forwarding of VME master cycles to the VXIbus done by the adapter), can then access the configuration registers.

If the system supports the VISA standard, a session can be opened to the adapter using the standard viOpen() call, as for example (using logical address 2):

```
ViSession rm_session;
ViSession session;
ViStatus status;

status = viOpenDefaultRM (&rm_session);
status = viOpen (rm_session, "VXI0::2::INSTR", VI_NULL, VI_NULL, &session);
```

The register offsets as specified in Table 1 can then directly be used in the viIn() and viOut() calls to access the configuration registers:


```
ViBusAddress offset;
ViUInt16 value;

offset = (ViBusAddress) 0x30; /* TrigStatus register */
status = viIn16 (session, VI_A16_SPACE, offset, &value);
```

If the system supports only bus addresses, the address of a register must be calculated using:

```
address = 49152 + (LA * 64) + offset
```

where 49152 is the base address for the configuration register space according to the VXIbus standard, `LA` is the logical address the adapter is set to, and `offset` is a register offset as specified in Table 1. The calculated address is in the A16 address space and the registers can be accessed using either user or supervisory accesses as well as either program or data accesses.

The following registers can be configured permanently using the on-board EEPROM. Using the algorithms as shown in Appendix A, the values stored in the EEPROM can be changed and will be used to initialize the registers after a power-on or hard reset.

Register	EEPROM offset
ID	0
DevType (for LA 0)	1
DevType (LAs 1..255)	2
VMEOffset	3
VMEControl	4

Attention: After changing the values in the EEPROM, a hard reset is necessary for the new values to become effective !

2.2.1 Logical Address

The ProDAQ 3040 Adapter can operate as either a Slot-0 controller or as a standard VXIbus device (non-Slot-0). When the ProDAQ 3040 is operated as the Slot-0 controller, it must be located in the left-most slot (slot “0“) of a VXIbus System Specification Rev 1.3 (or higher) compatible VXIbus mainframe and be set for logical address zero (0). If the ProDAQ 3040 is operated as a non-Slot-0, it may be located in any other slot and the logical address can be set to any value between 1 and 255.

Figure 9 shows the location of the logical address switch on the ProDAQ 3040. Set each switch to ‘Off’ for a logical one (1) and to ‘On’ for a logical zero (0). The picture shows the address switch set to logical address zero (0).

If the ProDAQ 3040 is used in a non-slot-0 position, it can be either statically or dynamically configured. To configure it statically, the logical address switch must be set to a value between 1 and 254. This determines the logical address of the module permanently and can only be altered by changing the setting of the logical address switch.

To configure the ProDAQ 3040 dynamically, the logical address switch must be set to 255. The resource manager will use the VXIbus MODID lines to access and configure the board, and assigns a logical address during run-time.

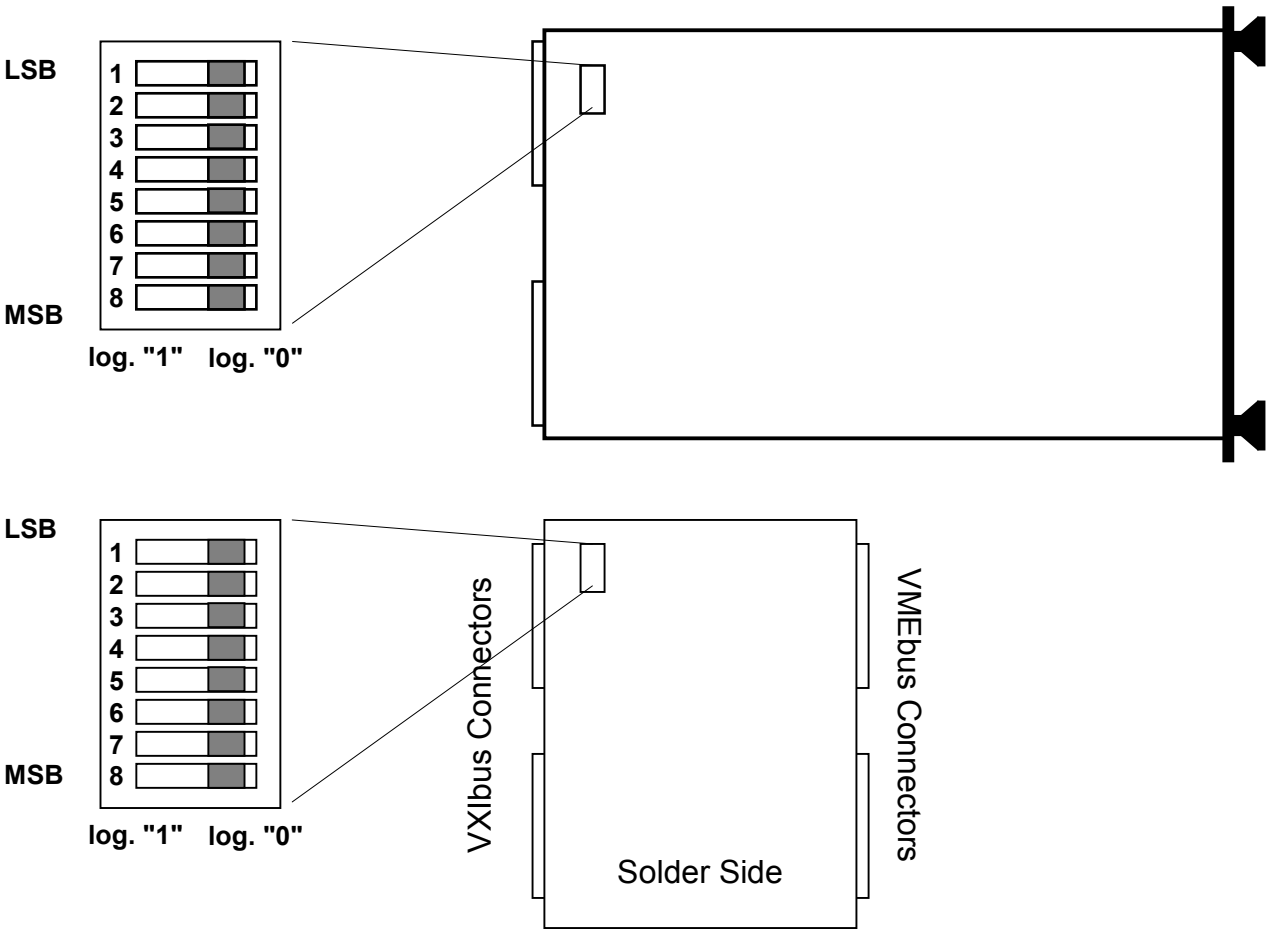


Figure 9 - Logical Address Switch Location

2.2.2 VXI to VME Address Space Mapping

The ProDAQ 3040 Adapter can be configured to forward slave accesses to an address range in VXIbus A24 or A32 to an address range in A16 or A24 or A32 in the VME space. This is done by writing register settings to the on-board EEPROM, from where they will be used after power-on or a hard reset to initialize the configuration registers.

To set up the address space mapping, use the following steps:

1. Select an address space in VXI

The mapped address range can be located in the VXIbus A24 address space or the VXIbus A32 address space. This is configured by setting the adapter to be either a

"A16/A24" device, a "A16/A32" device or a "A16 only" device. This information is located in the **ID** register (see 3.2.1 ID Register). Because the ID register content is initialized from the value stored in the EEPROM at address 0, the information stored there must be altered by using the algorithms shown in Appendix A.

To change the value, read first the EEPROM contents at address 0. To map an address range in the VXIbus A32 address space, set the bits 13 and 12 of the 16-bit word read to zero ("0") and one ("1"). To map an address range in the VXIbus A24 address space, set bits 13 and 12 both to zero ("0"). To disable any address mapping, set bits 13 and 12 both to one ("1"). Store the changed value at EEPROM address zero.

2. Specify the size of the mapped range

The size of the mapped range is determined by the *ReqMemory* field in the **DevType** register. Valid values for this field are:

VXI Address Space	VME Address Space	Size	ReqMemory Bits
A24	A16, A24, A32	64 kB	0111
A24	A24, A32	128 kB	0110
A24	A24, A32	256 kB	0101
A24	A24, A32	512 kB	0100
A24	A24, A32	1024 kB	0011
A24	A24, A32	2048 kB	0010
A24	A24, A32	4096 kB	0001
A24	A24, A32	8192 kB	0000
A32	A16, A24, A32	64 kB	1111
A32	A24, A32	128 kB	1110
A32	A24, A32	256 kB	1101
A32	A24, A32	512 kB	1100
A32	A24, A32	1024 kB	1011
A32	A24, A32	2048 kB	1010
A32	A24, A32	4096 kB	1001
A32	A24, A32	8192 kB	1000
A32	A24, A32	16384 kB	0111
A32	A32	32 MB	0110
A32	A32	64 MB	0101
A32	A32	128 MB	0100
A32	A32	256 MB	0011
A32	A32	512 MB	0010
A32	A32	1024 MB	0001
A32	A32	2048 MB	0000

As before, read the corresponding value from the EEPROM (**DevType** is address 1 and 2), modify the bits for the *ReqMemory* field and store the value back into the EEPROM. In the case of the **DevType** register this must be done twice, because the EEPROM contains a different ID for the case the adapter is used as slot-0 and as non-slot-0.

3. Specify the address range in VME

The address space on the VMEbus, where the VXI address space is mapped to, is specified by setting bits 1 and 0 in the **VMEControl** register (see 3.2.10). Because this register contains only this setting, a chosen value can be written directly to the EEPROM. The address of the **VMEControl** register value in the EEPROM is 4.

4. Specify the VME address offset

The base address of the VME target range in the chosen address space is programmable via the **VMEOffset** register. The initial value for the register is stored in the EEPROM at address 3. Depending on the setting of the *ReqMemory* field in the **DevType** register, only the upper bits may be valid (see 1.4). This value can be as well written directly to the correct location in the EEPROM.

5. Reset the System

To check whether the configuration was successful, reset the system and run the resource manager. The ProDAQ 3040 must be shown as a register based board with the required memory size as set in step 2. After the resource manager has configured the VXIbus offset and has enabled the board for A24 or A32, the resources on the VMEbus shall be accessible via the boards VXI address range.

3. Register Description

3.1 Address Map and Registers

All addresses are given in hexadecimal notation. Offset value is an offset in relation to the base address in A16 address space defined by Logical Address.

Offset	Name	Access	Description
0x00	ID	RO	ID Register
	LogAdr	WO	Logical Address Register
0x02	DevType	RO	Device Type Register
0x04	Status	RO	Status Register
	Control	WO	Control Register
0x06	Offset	RW	Offset Register
0x08	MODID	RW	MODID Register
0x0A	VMEOffset	RW	VME target image base address
0x0C	IRQStatusID1	RO	Latched Interrupt Status/ID – upper word
0x0E	IRQStatusID1	RO	Latched Interrupt Status/ID – lower word
0x10	IRQStatusID2	RO	Latched Interrupt Status/ID – upper word
0x12	IRQStatusID2	RO	Latched Interrupt Status/ID – lower word
0x14	IRQStatusID3	RO	Latched Interrupt Status/ID – upper word
0x16	IRQStatusID3	RO	Latched Interrupt Status/ID – lower word
0x18	IRQStatusID4	RO	Latched Interrupt Status/ID – upper word
0x1A	IRQStatusID4	RO	Latched Interrupt Status/ID – lower word
0x1C	IRQStatusID5	RO	Latched Interrupt Status/ID – upper word
0x1E	IRQStatusID5	RO	Latched Interrupt Status/ID – lower word
0x20	IRQStatusID6	RO	Latched Interrupt Status/ID – upper word
0x22	IRQStatusID6	RO	Latched Interrupt Status/ID – lower word
0x24	IRQStatusID7	RO	Latched Interrupt Status/ID – upper word
0x26	IRQStatusID7	RO	Latched Interrupt Status/ID – lower word
0x28	VXIControl	RW	VXI Control Register
0x2A	VMEControl	RW	Controls several VME parameters
0x2C	EEPROMData	RW	EEPROM Data Register
0x2E	EEPROMCtrl	RW	EEPROM Control Register
0x30	TrigStatus	RO	Actual Trigger Status
0x32	TrigIntMask	RW	Trigger Interrupt Mask / Latch state
0x34	TrigControl	WO	Trigger Line Control
0x36	TrigIntMode	RW	Trigger Interrupt Mode register
0x38	Reserved		
0x3A	IRQDir	RW	Interrupt Direction register
0x3C	SerNumHigh	RO	Serial Number upper word
0x3E	SerNumLow	RO	Serial Number lower word

Table 1 - Configuration Register Offsets

3.2 Register Details

3.2.1 ID Register

The ID register provides information about the device's manufacturer and configuration.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	EW0		EW0		1	1	1	0	0	1	1	1	0	0	0	0
Content	Device Class		Address Space		ManufacturerID											

Device Class This field indicates the module as a Register Based VXIbus device (value 0x3).

Address Space This field determines the addressing mode of the device's operational registers.
A16/A24 – 0x0
A16/A32 – 0x1
Reserved – 0x2
A16 Only – 0x3
The value of this field will be initialized during hardware initialization from the on-board EEPROM.

Manufacturer ID The Manufacturer ID is **0xE70** (3696) and has been assigned by the VXIbus Consortium. This number uniquely identifies the manufacturer of the device as Bustec Production Ltd.

3.2.2 LogAdr

The Logical Address register is a write-only register used by the VXIbus resource manager to assign the modules logical address during the dynamic configuration.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	-	-	-	-	WO	WO	WO	WO	WO	WO	WO	WO
Initial	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0
Content	Not used								LogicalAddr[7:0]							

3.2.3 DevType

The Device Type register contains a device dependent type identifier and the information about the memory space required by this device.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	EW1 or EW2*				EW1 or EW2*											
Content	ReqMemory[3:0]				ModelCode[11:0]											

ReqMemory[3:0] The required memory as defined in the VXIbus standard. The value of this field will be initialized during hardware initialization from the on-board EEPROM.

ModelCode[11:0] This field contains a unique card identifier. The adapter module has got two different codes depending on the slot position (slot0 or non-slot0).

3.2.4 Status

The Status register provides information about the device's status.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	h	h	h	h	h	h	h	h	h	h	h	h	h	h	h
Content	A24/A32 active	MODID*	Slot0	VMERead [2]	Logical Address								Ready	Passed	VMERead [1]	VMERead [0]

A24/A32 active A one (1) indicates that the A24/A32 address range is enabled.

MODID* A one (1) indicates that the device is not selected via the P2 MODID line. A zero (0) indicates that the device is selected by a high state on the MODID line.

Slot0 A one (1) indicates that the module is in the leftmost slot of a VXIbus system.

VME Read[2:0] A pattern '100' in this field indicates that the current read access was initiated by the VMEbus master.
A pattern '011' in this field indicates that the current read access was initiated by a VXIbus master.

Logical Address Contains the logical address the adapter is configured for. This may be defined by either the Logical Address Switch or the value written to the Logical Address register during the dynamic configuration.

Ready A zero (0) means the device is executing its self-test.

Passed After completing the self-test (signaled by a one (1) in the Ready bit), the Passed bit indicates the state of the self-test. A one (1) indicates that the self-test has successfully completed. A zero (0) means that the device has failed its self-test.

3.2.5 Control

The Control register contains bits that cause specific action to be executed by the device.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	WO	-	-	-	-	-	-	-	-	-	-	-	-	-	WO	WO
Initial	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0
Content	A24/A32	Not Used													Sysfail Inhibit	Reset

A24/A32 enable Writing a one to this bits enables the decoding of the A24/A32 address range.

Sysfail Inhibit A one (1) written to this bit disables the device from driving the SYSFAIL* line.

Reset A one written to this field forces the device into a reset state. This means the MODID driver will be disabled, if device is Slot 0 controller.

3.2.6 Offset

The Offset register sets the devices base address in A24/A32.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	Offset[15:0]															

Offset[15:0] Offset defines the base address of the A24 or A32 operational registers of a device in the VXI address space

3.2.7 MODID

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x
Content	Not used		Output Enable	MODID[12:0]												

Output Enable Writing a one to this bit enables the Slot 0 MODID driver. Writing a zero disables the MODID driver. This bit is cleared (zero) by device resets. When read, this bit indicates the state of the MODID drivers. A one means the drivers are enabled, a zero indicates that the drivers are disabled.

MODID[12:0] Writing a one to any of these bits drives the corresponding MODID line high. Writing a zero drives the corresponding line low. Writing to these bits has only effect, if the Output Enable bit is set. When read, each of these bits indicates the actual level of the corresponding MODID line.

3.2.8 VMEOffset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	EW3															
Content	VMEBase[15:0]															

VMEBase[15:0] The VMEBase defines the base of the target image in the VME A16, A24 or A32 address space. The value of this register is initialised from the EEPROM, but can be changed during runtime.

3.2.9 VXIControl

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	-	RW
Initial	x	x	x	x	x	x	x	x	0	0	1	1	0	1	x	0
Content	Not Used								BTO[3:0]				CLK10_FP_OE	CLK10_nFP_OSC		SYSRESET

SYSRESET Writing a one to this bit starting the generation of the SYSRESET. The bit will be cleared after the SYSRESET is done. The pulse will have the width of 250ms. The SYSRESET line will be asserted after the current register access is finished.
The SYSRESET will reset VXIbus only and will not be forwarded to VME side

CLK10_nFP_OSC The bit is used to switch between CLK10 source: when zero CLK10 comes from the front panel connector, when one comes from the on board oscillator.

CLK10_FP_OE The bit controls the output of the CLK10 front panel driver: when zero driver is in high impedance state, when one the output of the driver is enabled.

BTO[3..0] These bits are used to set the adapter's bus timer time-out value. The following values can be set:

0000 – disabled
0001 – 16us
0010 – 32us
0011 – 64us (default)
0100 – 128us
0101 – 256us
0110 – 512us
0111 – 1024us
1xxx – reserved (timer is disabled)

3.2.10 VMEControl

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW	RW
Initial	x	x	x	x	x	x	x	x	x	x	x	x	x	x	EW4	
Content	Not Used														VME Addr Space[1:0]	

VME Addr Space[1:0] Selects the VME address space the accesses to the VXIbus slave image are forwarded to. Depending on this setting the upper three bits of the address modifier code used in the VXI bus transfer are replaced before forwarding it to the VME bus. These bits are initialised during power-up or reset from the EEPROM, but can be changed during runtime to allow access to VME boards implementing different address spaces.

BITS	ADDR SPACE
00	A16
01	A24
02	A32
03	CR/CSR

3.2.11 EEPROMData

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Content	16-bit EEPROM Read/Write Data															

3.2.12 EEPROMCtrl

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	RO	WO	WO	WO	-	RW	RW	RW	RW	RW	RW	RW
Content	Not Used				READY	READ	START	RESET	Not Used	OFFSET[6..0]						

RESET Resets the EEPROM Control Logic

START Writing a "1" to this bit starts the EEPROM access

READ Setting this bit to "1" together with the START bit will cause a read access to the EEPROM, setting this bit to "0" will cause a write access.

READY This bit will be set to "1" by the EEPROM control logic after finishing an access cycle.

OFFSET[6..0] Address offset of the data in the EEPROM to be read/written.

3.2.13 TrigStatus

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	x	x	x	x	h	h	h	h	h	h	h	h	h	h	h	h
Content	Not Used				TRGSTS_FP[1:0]		TRGSTS_ECL[1:0]		TRGSTS_TTL[7:0]							

TRGSTS_TTL[7:0] Show the status of the VXI TTL trigger lines. “0” means trigger line is in inactive state. “1” means trigger line is in active state.

TRGSTS_ECL[1:0] Show the status of the VXI ECL trigger lines. “0” means trigger line is in inactive state. “1” means trigger line is in active state.

TRGSTS_FP[1:0] Show the status of the FP trigger lines. “0” means trigger line is in inactive state. “1” means trigger line is in active state.

3.2.14 TrigIntMask

When writing the Trigger Interrupt Mask register defines which trigger will cause an interrupt.

When reading this register shows the awaiting lines for an interrupt service. During the interrupt acknowledge cycle the information about the events awaiting for a service is latched in IRQStatusID register and then cleared in TrigIntMask.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	WO	WO	WO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	N.U.	VME_IRQ[2:0]			TRGMASK_FP[1:0]		TRGMASK_ECL[1:0]		TRGMASK_TTL[7:0]							

TRGMASK_TTL[7:0] Writing a one (1) to one of these bits enables a VME interrupt to be generated when a selected edge is detected on one of the corresponding VXIbus TTL trigger lines.

Reading (1) from these bits means that the selected edge of the trigger lines happened and caused the interrupt. After latching the bits during interrupt acknowledge cycle these bits which were set are cleared.

TRGMASK_ECL[1:0] Writing a one (1) to one of these bits enables a VME interrupt to be generated when a selected edge is detected on one of the corresponding VXIbus ECL trigger lines.

Reading (1) from these bits means that the selected edge of the trigger lines happened and caused the interrupt. After latching the bits during interrupt acknowledge cycle these bits which were set are cleared.

TRGMASK_FP[1:0] Writing a one (1) to one of these bits enables a VME interrupt to be generated when a selected edge is detected on one of the corresponding FP trigger lines.

Reading (1) from these bits means that the selected edge of the trigger lines happened and caused the interrupt. After latching the bits during interrupt acknowledge cycle these bits which were set are cleared.

VME_IRQ[2:0] Defines the VME interrupt level, which is used for an interrupt from VXIbus trigger. Zero (0x000) written here disables trigger interrupter

Selection of the active edge of the trigger lines which will cause the interrupt (if enabled) is done in the TrigIntMode register.

During the interrupt acknowledge cycle the information about the awaiting events (trigger edges) is latched in IRQStatusID register. Once latched the awaiting bit is cleared in TRIGIntMask register allowing for the next event to come.
More than one trigger event can be serviced during the single interrupt cycle.

32-bit status/ID returned when acknowledging trigger interrupter:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial					0	0	0	0	0	0	0	0	0	0	0	0
Contents	Not Used				TRGMASK_FP[7:0]		TRGMASK_ECL[7:0]		TRGMASK_TTL[7:0]							

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial	0	0	0	0	0	0	0	0	h	h	h	h	h	h	h	h
Contents	If all zero then this is trigger source								Logical Address							

3.2.15 TrigControl

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	CMD[3:0]				TRGEN_ FP[1:0]		TRGEN_ ECL[1:0]		TRGEN_TTL[7:0]							

TRGEN_TTL[7:0] When any bit is set the corresponding VXITTL trigger line will be affected by the command set on the bits CMD[3:0]

TRGEN_ECL[1:0] When any bit is set the corresponding VXIECL trigger line will be affected by the command set on the bits CMD[3:0]

TRGEN_FP[1:0] When any bit is set the corresponding FP trigger line will be affected by the command set on the bits CMD[3:0]

CMD[3:0] The command specifies the action to perform on the selected trigger lines. The action will start immediately after the write access to this register. When pulse generation is in progress then the new command performed on the same trigger line will overcome the previous one.
0001 – deassert
0010 – assert
0011 – negate
1000 – pulse 100ns
1001 – pulse 200ns
1010 – pulse 1us
1011 – pulse 10us
others – reserved, no action performed

3.2.16 TrigIntMode

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0
Content	Not Used				TRGEDGE_ FP[1:0]		TRGEDGE_ ECL[1:0]		TRGEDGE_TTL[7:0]							

TRGEDGE_TTL[7:0] Writing a one (1) to one of these bits selects the rising edge as an active edge, which will generate the interrupt. Writing a zero (0) selects the falling edge as an active edge.
Readout shows current setting of the bits.

TRGEDGE_ECL[1:0] Writing a one (1) to one of these bits selects the rising edge as an active edge, which will generate the interrupt. Writing a zero (0) selects the falling edge as an active edge.
Readout shows current setting of the bits.

TRGEDGE_FP[1:0] Writing a one (1) to one of these bits selects the rising edge as an active edge, which will generate the interrupt. Writing a zero (0) selects the falling edge as an active edge.
Readout shows current setting of the bits.

3.2.17 IRQDir

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	-	-	-	-	-	-	-	-	RW	RW	RW	RW	RW	RW	RW	-
Initial	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	x
Content	Not Used								IRQDIR[7:1]							N.u.

IRQDIR[7:1] When set to one (1) these bits enables forwarding corresponding interrupts from VXIbus to VMEbus. Forwarding interrupts from VMEbus to VXIbus is then disabled.
When the bits are cleared (0) the corresponding VXIbus interrupt will not be forwarded to the VMEbus. It automatically enables VMEbus interrupts to be forwarded to VXIbus for the given level.

3.2.18 SerNumHigh

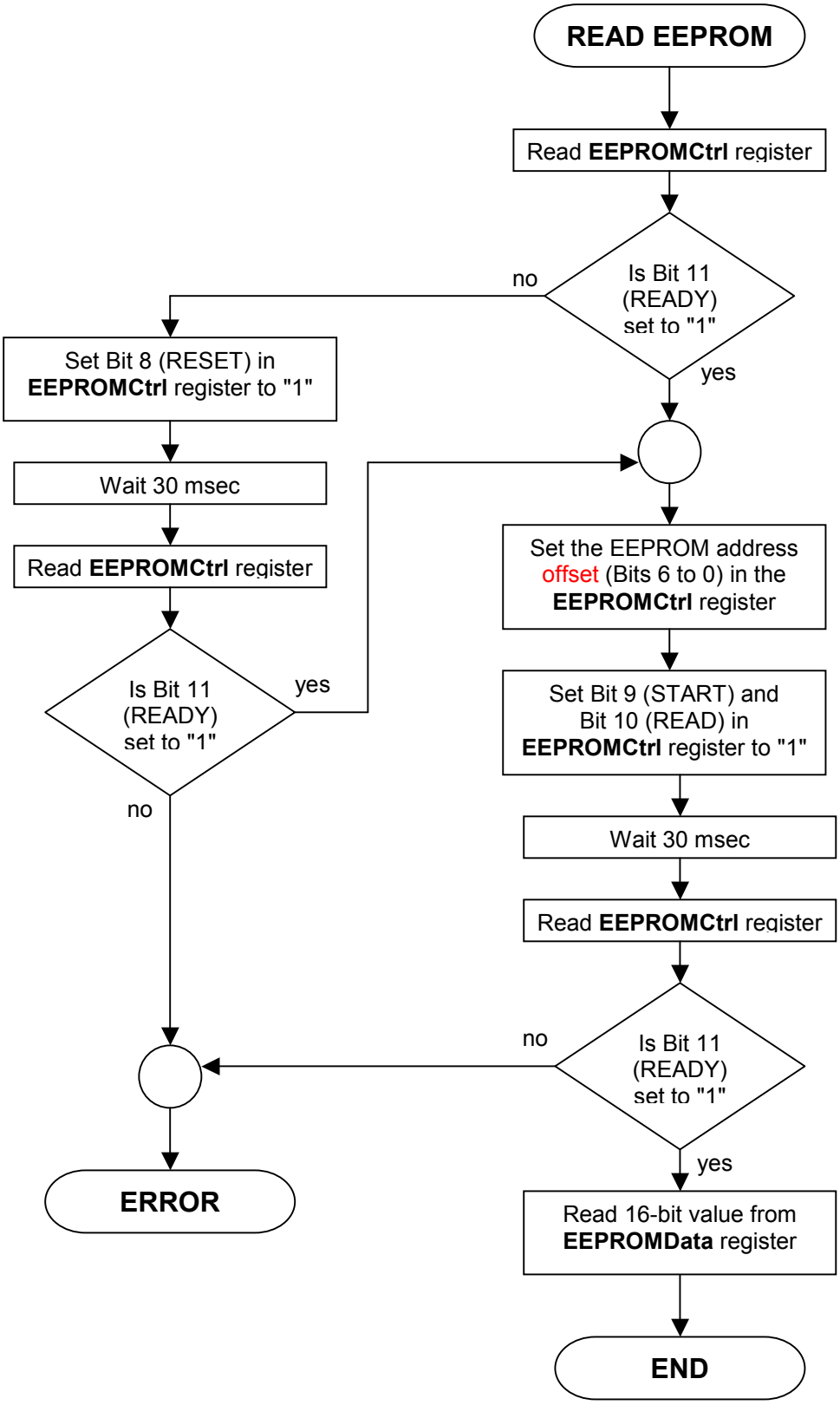
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	EW5															
Content	SN[31:16]															

3.2.19 SerNumLow

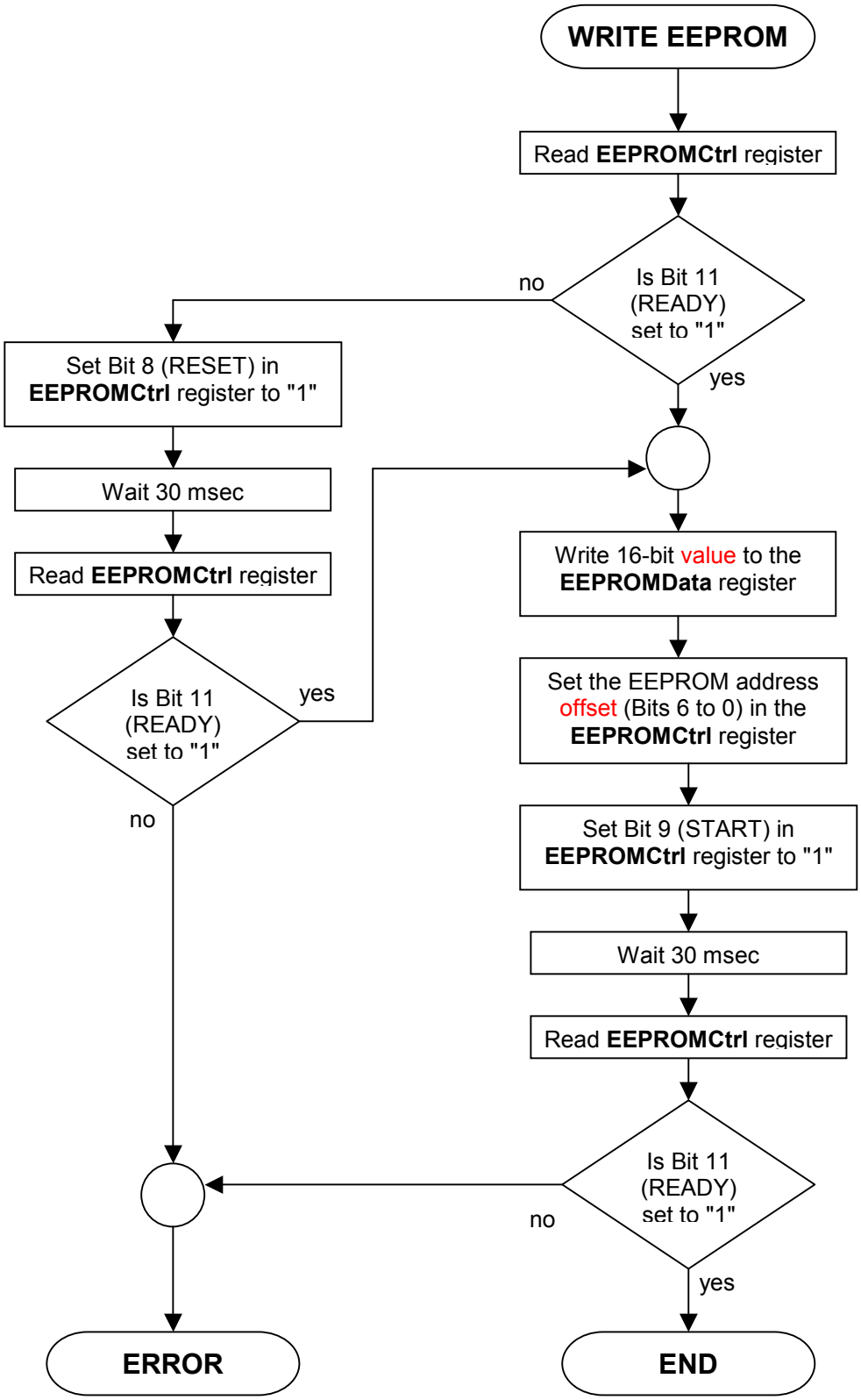
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	EW6															
Content	SN[15:0]															

Appendix A. EEPROM Programming

The following flowchart shows the algorithm for reading a 16-bit value from the EEPROM:



The following flowchart shows the algorithm for writing a 16-bit value to the EEPROM:



<>

Bustec Production, Ltd.
World Aviation Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

Bustec, Inc.
35486 Lorain Road, North Ridgeville, OH 44039, U.S.A
Tel. +1 440 327 8930, Fax: +1 440 327 8982