VME64x to WB core User Guide

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
2.0	2017-12-12		TG

Contents

1 Introduction **1**

2 Features **2**

2.1 VME interface . . . 2

2.1.1 Supported: . . . 2

2.1.2 Not supported: . . . 2

2.1.3 Deviations . . . 3

2.2 WB interface . . . 3

2.3 CR/CSR space . . . 3

2.4 Interrupt controller . . . 4

3 VME64x Core Instantiation **5**

3.1 Generics . . . 5

3.2 Ports . . . 6

4 Programming the VME64x Core **8**

5 Performance **9**

6 Changes in V2 (compared to V1) **10**

7 Appendix: Implementation of the core **11**

7.1 xvme64x_core.vhd . . . 11

7.2 vme_bus.vhd . . . 11

7.3 vme_cr_csr_space.vhd . . . 11

7.4 vme_func_match.vhd . . . 11

7.5 vme_user_csr.vhd . . . 11

7.6 vme_irq_controller.vhd . . . 12

8 Appendix VME64 VITA-1 rules compliance **13**

9 External References **16**

Chapter 1

Introduction

This core implements a VME64x slave - WB master bridge.

The design can be downloaded from https://www.ohwr.org/projects/vme64x-core

The vme64x core conforms to the standards defined by ANSI/VITA VME64 [1] and VME64x [2]. In particular this core is provided with the "plug and play" capability. It means that in the vme64x core you can find a CR/CSR space whose base address is set automatically with the geographical address lines and does not need to be set by jumpers or switches on the board. Manual configuration is error prone and is thus avoided. Software must map the module memory in the address space by writing the CSR space.

The core supports SINGLE, BLT (D32), MBLT (D64) transfers in A16, A24, and A32 address modes and D08 (OE), D16, D32 data transfers. The core can be configured via the CR/CSR configuration space. A ROACK type IRQ controller with one interrupt input and a programmable interrupt level and Status/ID register are provided optionally and can be enabled at instantiation.

Since the vme64x core acts as a Wishbone (WB) master in the WB side, the WB pipelined single read/write transfer is provided by the core. This functionality conforms the Wishbone B4 standard [3].

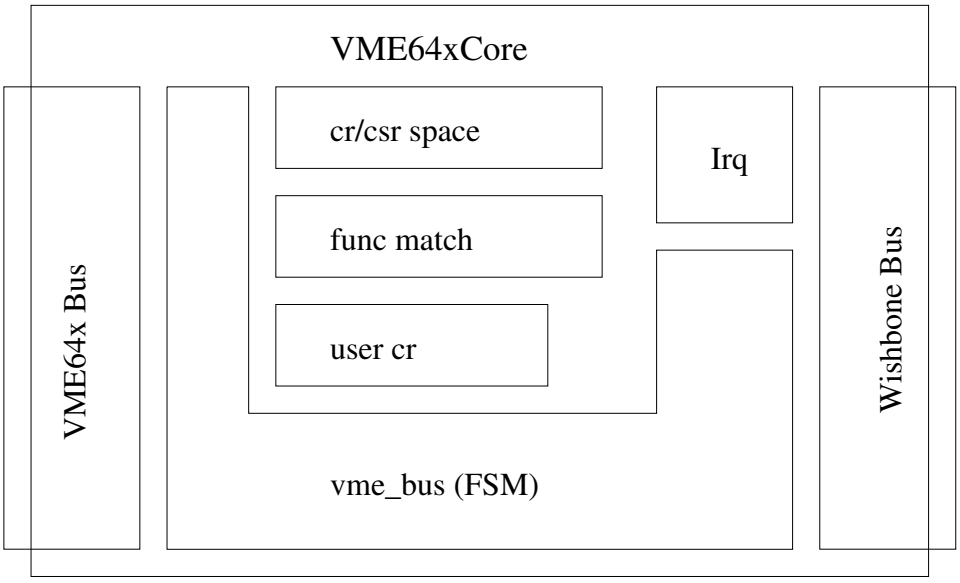

Figure 1.1: Block schema

Chapter 2

Features

2.1 VME interface

This section lists which VME features are supported and which are not supported by the core.

2.1.1 Supported:

- D08(EO), D16, D32
- Addressing mode: A16, A24, A32
- BLT, MBLT
- D08(O), I(7-1), ROAK interrupts
- CR/CSR space with extensions from VME64x
- Geographic Address (GA), dynamic configuration (ader).

2.1.2 Not supported:

- 2eVME
- 2eSST
- Dynamic size (DFSR, DFS)
- Fixed address (FAF)
- Extra Function Mask (EFM)
- XAM (no 2e)
- A40, A64
- MD32 (multiplexed data cycle, only for A40)
- LCK (bus lock)
- UAT (unaligned accesses)
- RMW cycles (but should work)
- ADO, ADOH (address only cycle)
- D08, D16 for BLT
- RETRY (cf rule 2.91 - incompatibility with WB)

2.1.3 Deviations

- Not compatible with non-VME64x crates (doesn't support jumpers to set the GA).
- The reset bit in the Bit Set Register is automatically cleared at the next access.
- Automatically repeat interrupts every 1 ms if the source is always active.

2.2 WB interface

This section corresponds to the datasheet required by the WB specification [3].

1. Compliant to Wishbone B4 specifications
2. Slave
3. Signals name follows the specification
4. err_i is forwarded to VME as BERR*
5. rty_i is not supported
6. no TAGs
7. Port size is 32 bit
8. Port granularity is 8 bit
9. Maximum operand size is 8 bit (TBC)
10. Data transfer ordering is BIG ENDIAN
11. Sequence of data transfer is defined by the VME side
12. No CLK_I signal, clock is provided separately.
    - Non pipeline behaviour (but compatible with pipeline). The core doesn't take any advantage of the pipeline behaviour, as the WB bus is much faster than the VME bus.

2.3 CR/CSR space

To provide a "plug and play" capability, CR/CSR space is implemented as defined by ANSI/VITA Standards for VME64 Extensions.

A dedicated "Configuration ROM / Control & Status Register" (CR/CSR) address space is provided by the core. It consists of ROM and RAM regions with a set of well-defined registers. It is addressed with the address modifier 0x2F in the A24 address space.

Every VME module occupies a 512 kB page in this address space. The location of this page in the A24 space is defined by geographical address lines on the backplane: each slot is provided with a unique geographical five bit address at the J1 connector (row d). From these bits A23...A19 of the CR/CSR page are derived.

If the geographical address is not correct (GA parity bit does not match), the base address is set to 0x00, which indicates a faulty condition. An odd parity is used.

If the board is plugged into an old crate that doesn't provide the GA lines, the CR/CSR space cannot be accessed and therefore the core remains disabled.

The CR/CSR space can be accessed with the data width D08(EO), D16 byte(2-3) and D32. Please note that in compliance with the CR/CSR definition, only every fourth location in the CR/CSR space is used. If the master tries to write another location the write will not take effect and if the master reads the byte(0) or byte(1) or byte(2) locations the value returned is 0.

As you can see in the table below, not all this space of memory is defined yet, and the designer can add additional CR and CSR spaces called User Csr and User CR which are not implemented in the vme64x core.

The location of the User CR and User CSR regions as well as the CRAM region are programmable. For each of these, six bytes defining the start and the end address (with respect to the start of the configuration space) are reserved in the CR region. Designers are free to use these regions for module specific purposes.

By default the size of the CRAM space is 0, which means it is disabled and doesn't use any resources. User can define the address range of CRAM in order to generate a programmable area.

All the registers in the CSR space have been implemented as defined by the VME64 Extensions.

Table 2.1: CSR Address Space

Start Address	**End Address**	**Content**
0x7ff60	0x7ffff	CSR (Control Status Registers)
0x7fc00	0x7ff5f	Reserved for CSR
BEG_USER_CSR	END_USER_CSR	User defined CSR (option)
BEG_CRAM	END_CRAM	User defined CRAM (option)
BEG_USER_CR	END_USER_CR	User defined CR (option)
0x00000	0x00fff	CR (Configuration ROM)

In addition to the standard registers in the CSR space and for compatibility with existing drivers for previous version of the core, the VME64x defines by default a user CSR space (within the CSR space reserved by the VME64x standard) with the following registers:

Table 2.2: Default User CSR Space

Address	**Content**	**Reset value**
0x7ff5f	IRQ vector	0x00
0x7ff5b	IRQ level	0x00

2.4 Interrupt controller

The interrupt controller implemented is a ROAK (Release On Acknowledge) type controller. It means that the Interrupter releases the interrupt request lines when it acknowledges the interrupt cycle. Upon synchronously detecting a rising edge on the interrupt request signal input on the WB bus, the VME64x core drives the IRQ request line on the VME bus low thus issuing an interrupt request. The VME master acknowledges the interrupt in the form of an IACK cycle. During the IACK cycle the vme64x core sends the IRQ_Vector to the master. After the interrupt is acknowledged, the VME IRQ line is released.

There are seven VME IRQ lines but only one interrupt request input. For the purpose of configuring which of the seven IRQ lines the VME64x core will drive (in response to a rising edge on the IRQ input), an IRQ Level register is available in the user CSR space. The value of this register corresponds to the number of the IRQ line on the VME bus that is to be used (note that on the VME master side priorities are taken into account, IRQ7 having the highest priority and IRQ1 the lowest). If the IRQ level register is set to 0x00, interrupts are disabled. In the default power-up and reset configuration the interrupts are disabled.

There is a non-standard mechanism to retrigger unhandled interrupts. Once an interrupt is asserted by the WB slave, the interrupt is marked as pending and the interrupt request that it is relayed on the VME bus. The OS and the driver has to acknowledge the interrupt and to act on the hardware so that the WB slave doesn't request anymore OS attention. If the OS acknowledge the interrupt but doesn't acknowledge the request, the VME64x Core will relay again the interrupt on the VME bus after 1ms.

Chapter 3

VME64x Core Instantiation

There are two top-level entities:

- The `xvme64x_core` that is the main top-level entity. It uses records for the `g_DECODER` generic, VME and WB buses in order to simplify the connections.
- The `vme64x_core` that is a wrapper of `xvme64x_core` which allows interfacing with verilog code.

3.1 Generics

Generic `g_CLOCK_PERIOD` defines the clock priod in ns. This generic must be set by the user and is used for synchronization of the VME DS signal.

Generic `g_DECODE_AM` enables/disables the AM field of ADER when decoding address. When it is set to false, behavior of this core is consistent with its previous versions. In particlular, when false, the AM field of ADER is not used when decoding address, so the core will recognize any access allowed by the corresponding AMCAP. New designs should set this generic to true.

Generic `g_USER_CSR_EXT` enables/disables user-defined CSR. The interface with the user CSR is a very simple synchronous SRAM (signals `user_csr_addr_o`, `user_csr_data_i`, `user_csr_data_o` and `user_csr_we_o`). In addition, if user-defined CSR is enabled, the input port `irq_level_i` and `irq_vector_i` are used by the interrupt controller to define the irq level and vector (otherwise they are read from the default user CSR registers).

Generic `g_WB_GRANULARITY` specifies the address granularity of the wishbone bus. The value is one of:

- `WORD`: addresses represent words, so VME address 4, 5, 6 and 7 are translated to WB address 1.
- `BYTE`: addresses represents bytes, so VME addresses 4, 5, 6 and 7 are translated to WB address 4. The two LSB of WB address are always 0.

The other generics define values in the CSR. The package `vme64x_pkg` defines default constants for these values, see the VME64x specification for details about these values:

- `g_MANUFACTURER_ID` provides the manufacturer ID,
- `g_BOARD_ID` provides the board ID,
- `g_REVISION_ID` provides the revision ID,
- `g_PROGRAM_ID` provides the type of code in CR,
- `g_ASCII_PTR` provides the pointer to the user defined ASCII string in CR,

- g_BEG_USER_CR and g_END_USER_CR provides the range of the user defined CR area. If the range is not null, ports user_cr_addr_o and user_cr_data_i must be connected to a ROM.
- g_BEG_CRAM and g_END_CRAM provide the range of user CRAM. If the range is not null, the core instantiates an SRAM.
- g_BEG_USER_CSR and g_END_USER_CSR provide the range of the user defined CSR. See above the description of g_USER_CSR_EXT.
- g_BEG_SN and g_END_SN provides the area in CR of the serial number.
- g_DECODER describes the 8 function decoder. Each decoder is described by the following bits (see VME64x specification for details):
- adem bits 8 to 31: address mask
- adem bits 0 to 7: must be set to 0
- amcap: address modifier supported by the decoder. Only bits 0x08 to 0x0f and 0x38 to 0x3f can be set to 1.
- dawpr: data access width (ignored by the decoder). Note that setting adem to 0 disable the decoder. If disabled decoders, they don't use any hardware resources.

3.2 Ports

- clk_i is the clock signal, and the clock of the WB bus. Note that the g_CLOCK_PERIOD generic must be set according to the clk_i frequency to get correct timing for the VME DS signals. The VME DTACK and BERR signals are supposed to be released at most 30ns after DS is released; as the design needs 4 closk to release them (due to synchronizer), this means the minimal frequency is supposed to be 133Mhz. In practice, VME masters are much more tolerant.
- rst_n_i is the reset signal. It could be considered as a power-on reset and is synchronous.
- rst_n_o is the reset signal to the wishbone core. It is asserted in case of reset on the VME bus, or if the module reset bit is set in the CSR, or if the rst_n_i signal is asserted.
- vme_i and vme_o are signals for the VME bus. Refer to the VME64 standard for details.
- wb_i and wb_o are the signals for the WB bus. Refer to the WB specification for details. Note that the WB rty (retry) signal cannot be used, as the VME BLT transactions can only be retried during the address phase and this restriction is not exposed to the WB side. The WB err signal is forwarded to the VME bus as BERR. The address on the WB bus corresponds to the lower bits of the address on the VME bus (bits used to decode the address are cleared on the WB bus).
- irq_ack_o signal is asserted during one cycle when the VME64x Core acknowledge the interrupt on the VME bus. This signal could be used by the slave interrupt controller.

The following signals are used only when the g_USER_CSR_EXT generic is set to true:

- irq_level_i defines which VME IRQ signal is asserted by the VME64x Core to send an interrupt to the VME bus. If set to 0, interrupts are never sent. The level corresponds to the interrupt priority on the VME bus, 7 is the highest priority and 1 the lowest. The value shouldn't change while an interrupt is pending.
- irq_vector_i is the vector sent on the VME bus by the core during an acknowledge cycle.
- user_csr_addr_o, user_csr_data_i, user_csr_data_o, user_csr_we_o define an interface to an external SRAM containing the user CSR values. For read cycles, the data value must be stable on the next cycle.

The following signals are used when a user CR area is defined (i.e. the range defined by g_BEG_USER_CR and g_END_USER_CR is not null):

- user_cr_addr_o, user_cr_data_i define an interface to an external ROM containing the user CR values. Data must be stable on the next cycle.

Note that the `vme` ports are designed to be connected to VME bus transceivers like SN74VMEH2250. In the particular case of the CERN SVEC card, the signals `berr` and `irq` are inverted by the transceivers, so a `not` gate must be inserted in the FPGA. You can refer to the `svec_vmecore_test_top.vhd` file in the svec repository (https://www.ohwr.org/projects/svec) for an example.

Chapter 4

Programming the VME64x Core

After power-up or reset, the VME core is disabled (as the `module_enable` bit is cleared) and therefore only the CS/CSR space can be accessed. Software must then first map the module memory in the address space by setting the Address Decoder Compare (ADER) registers in CSR, which, together with Address Decoder Mask (ADEM) registers in the CR relocate the module memory to the desired address range. ADER for each function must also contains the AM code to which it responds. After the module has been placed in the desired address space, it can be enabled by writing 0x10 (`module_enable`) to the Bit Set Register in the CSR.

If the master needs to access to the slave using different address space (e.g. both A32 and A24), or different transaction (e.g. both single and BLT), several function decoders must be used.

The base address is of the CR/CSR space is set by the geographical lines. If the core is used in an old VME system without GA lines, the core reads GA as "11111" which is invalid. As a consequence, the CS/CSR is not accessible and the card cannot be enabled.

It is possible to reset the card in software by setting the `reset` bit in the Bit Set Register. Contrary to the VME64 specification (and for backward compatibility with drivers and previous versions of the core), this bit is automatically cleared during the next CSR write access.

PSEUDO CODE

1. Determine the geographical address of the card (e.g. bus scan)
2. Optionally reset the card (through the Bit Set and Clear registers)
3. Set ADERs
4. Enable the card: Write 0x10 to the Bit Set Register

Chapter 5

Performance

The performance was measured with the `test_vme` program, available in the svec repository. In the measurement setup, the master was the MEN A20 board (https://www.men.de/products/discontinued-products/a20/) and the VME64x Core frequency was 125Mhz.

A24 SCT DMA:

- Read Rate: 15.7 MB/sec
- Write Rate: 17.1 MB/sec

A24 BLT DMA:

- Read Rate: 12.4 MB/sec
- Write Rate: 12.9 MB/sec

A24 MBLT DMA:

- Read Rate: 25.9 MB/sec
- Write Rate: 26.2 MB/sec

According to the simulation, the bad performances of BLT transfer is due to the master.

Chapter 6

Changes in V2 (compared to V1)

- Core is smaller (number of slices is less than 1000)
- No retry
- No endianess convertion
- WB data bus is 32 bit
- Internal component declarations removed.
- Async inputs registered with gc_sync_register.

Chapter 7

Appendix: Implementation of the core

This section describes the internal implementation of the VME64x core.

7.1 xvme64x_core.vhd

The top module `xvme64x_core` instantiates the sub-modules and also synchronizes the asynchronous VME signals (that need to be) to avoid metastability problems.

This module also handles the `g_USER_CSR_EXT` generic and instantiates a default user CSR if the generic is set to false.

7.2 vme_bus.vhd

This is the main module. It implements an FSM to handle the VME protocol, and acts as the interface between the VME bus and either the WB bus or the CR/CSR memory.

The module also handles the interrupt acknowledge. If IACK is asserted on a falling edge of AS, the cycle is considered as an acknowledge cycle. The FSM then waits until IACKIN is asserted (or until AS is deasserted). If an interrupt was pending at the right level when IACKIN is asserted, the VME64x Core responds to the acknowledge cycle with the interrupt vector; otherwise it asserts IACKOUT.

7.3 vme_cr_csr_space.vhd

This module implements the CR and CSR spaces. It builds (during elaboration) the CR memory from the generics value, handle accesses from the VME bus to these memories, interfaces with CRAM (if present), user CR (if present) and user CSR (if present).

7.4 vme_func_match.vhd

This module checks if the VME address+AM has to be handled by this VME slave according to ADER and decoder values. Gives back the corresponding WB address.

7.5 vme_user_csr.vhd

This module implements a default user CSR with the irq_level and irq_vector registers.

7.6 vme_irq_controller.vhd

This module implements the interrupt controller. The interrupt cycle is:

1. The wishbone slave generates an pulse on the `int` line when it has to interrupts the master
2. If no interrupt is pending and the retry mechanism is not started, this module asserts (to 0) the corresponding VME IRQ line (as defined by `irq_level`).
3. When ack'ed, the interrupt is marked as not pending anymore.
4. If the interrupt request stays active for more than 1 cycle (therefore it isn't a pulse), a retry mechanism is started. The interrupt will be re-sent on the VME bus every 1ms as long as it is active.

Chapter 8

Appendix VME64 VITA-1 rules compliance

This appendix lists all rules in the VME64 (in the textual appearance order), and specifies how it is followed. When the rule doesn't concern this core, the reason is brievly indicated: *Master* when the rule applies only to master modules, *D64* or *A64* for unsupported features.

- 2.1a: Master
- 2.69: Master
- 2.2: Followed
- 2.3: Followed, excluded from c_AMCAP_ALLOWED. [no TB]
- 2.70: D64
- 2.7: Followed
- 2.8: Followed
- 2.9: Followed
- 2.71: A64
- 2.10: Master
- 2.11: Followed
- 2.72: A64
- 2.73: A40
- 2.74: Followed (A32, A24, A16 supported)
- 2.75: Followed (likewise)
- 2.76: Followed (D16 and D08(EO) supported)
- 2.77: Followed (likewise)
- 2.4: Followed (D32 supported)
- 2.5: Followed (D16 supported)
- 2.12a: Master
- 2.78: Master
- 2.66: Followed

- 2.79: Master
- 2.80: Master
- 2.6: Followed [no TB]
- 2.68: Followed [no TB]
- 2.81: Followed (LOCK not accepted)
- 2.82: Followed (likewise)
- 2.83: Master
- 2.84: Followed (lock)
- 2.85: Followed (CR/CSR layout)
- 2.86: Followed
- 2.87: Followed (D08(O) is the data access supported)
- 2.93: Master
- 2.18: Followed (A[] and LWORD lines are registered)
- 2.19: Master
- 2.20: Master
- 2.21: Master
- 2.22: Master
- 2.23: Master
- 2.24: Master
- 2.25: Followed (DATA lines are all driven for read, MBLT not supported)
- 2.26: Followed (Likewise)
- 2.27: Master
- 2.28: Master
- 2.29: Master
- 2.30: Master
- 2.31: Master
- 2.32: Master
- 2.33a: Master
- 2.34a: Master
- 2.35: Master
- 2.36: Master
- 2.37: Master
- 2.38: Master
- 2.39: Master
- 2.40: Master

- 2.41: Master
- 2.42: Master
- 2.43: Master
- 2.44a: Master
- 2.94: Master
- 2.45: Master
- 2.46a: Master
- 2.47a: Master
- 2.48a: Master
- 2.49: Master
- 2.50: Master
- 2.51: Master
- 2.52: Master
- 2.95: Master
- 2.96: Master
- 2.53a: Followed (VME_DATA_DIR is set only once DSA goes low)
- 2.54a: Followed
- 2.55: Followed (number of states in the main FSM + synch FF)
- 2.28a: Followed (likewise)
- 2.56a: Followed
- 2.57: Followed
- 2.98: TBC
- 2.58a: Followed (released at the same time)
- 2.99: TBC (retry)
- 2.100: TBC (retry)
- 2.101: TBC (retry)
- 2.102: TBC (retry)
- 2.103: TBC (retry)
- 2.104: TBC (retry)
- 2.105: TBC (retry)
- 2.59: Bus timer
- 2.60: Bus timer
- 3.x: Arbitration
- 4.1: Backplace
- 4.50: Followed (slave can generate interrupt)
- 4.2 Followed

Chapter 9

External References

Specifications used for this core

- [1] VME64 ANSI/VITA 1-1994 (Stabilized Maintenance 2011)
- [2] VME64 Extensions ANSI/VITA 1.1-1997 (Stabilized Maintenance 2011)
- [3] Wishbone System-on-chip (SoC) Interconnection Architecture for Portable IP Cores, Revision B4