VME64x Slave Interface
IP Core Specifications

Author: Paolo Musico
Paolo.Musico@ge.infn.it

Rev. 0.1
December 1, 2005

Revision History

Rev.	Date	Author	Description
0.1	1/12/05	Paolo Musico	First Draft

Contents

1

Introduction

The VME (Versa Module Extension) Slave Interface core provides functionalities to interface a user logic to the standard cycles of the VME bus.
It implements several functionalities as required by VME64x specifications, with the following features:

- Decodes A32D32 single and block transfer cycles.
- Decodes A32D32 and A32D64 2eVME and 2eSST cycles.
- Implements Configuration ROM and Control/Status Registers.
- Translate the VME cycles into a user friendly non multiplexed bus which can be directly connected to RAMs, FIFOs, … synchronous and asynchronous.
- The user can generate interrupts onto the VME bus in ROAK mode.
- User side selectable data path width between 32 and 64 bit.
- On user side 8 input and 8 output general purpose lines are implemented.

2

I/O ports

2.1 VME side signals

Port	Width	Direction	Description
VME_A	31	Bidir	Muxed address and data lines (in 64 bit mode)
VME_D	32	Bidir	Data lines
VME_AM	6	Input	Address modifiers lines
VME_LWORDb	1	Bidir	Muxed longword (active low) and A0/D32 line (in 64 bit mode)
VME_ASb	1	Input	Address strobe line (active low)
VME_DS0b	1	Input	Data strobe (active low)
VME_DS1b	1	Input	Data strobe (active low)
VME_WRITEb	1	Input	Write line (active low)
VME_IACKb	1	Input	Interrupt acknowledge line (active low)
VME_IACKINb	1	Input	Interrupt acknowledge line in (active low)
VME_IACKOUTb	1	Input	Interrupt acknowledge line out (active low)
VME_IRQ	7	Output	Interrupt lines (active low)
VME_DTACK	1	Output	Data acknowledge line
VME_BERR	1	Output	Bus error line
VME_GAb	5	Input	Geographical address lines (active low)
VME_GAPb	1	Input	Geographical address parity line (active low)

2.2 User side signals

Port	Width	Direction	Description
USER_DATA	64	Bidir	Data lines
USER_ADDR	22	Output	Address lines
USER_WEb	1	Output	Write enable line (active low)
USER_OEb	1	Output	Output enable line (active low)
USER_REb	1	Output	Read enable line (active low)
USER_CEb	8	Output	Chip enable lines (active low)
USER_IRQb	7	Input	Interrupt request lines (active low)
USER_WAITb	1	Input	Wait line (active low)
USER_STATUS	8	Input	User status lines readable through CSR
USER_CTRL	8	Output	User output lines settable through CSR
USER_D64	1	Input	User side data path is 64 bit

2.3 Other signals

Port	Width	Direction	Description
CLK	1	Input	Free running clock
RESETb	1	Input	Asynchronous Reset (active low)
VME_DATA_DIR	1	Output	Direction line for VME_D[] transceiver
VME_DBUF_OEb	1	Output	Enable line for VME_D[] transceiver (active low)
VME_ADDR_DIR	1	Output	Direction line for VME_A[] transceiver
VME_ABUF_OEb	1	Output	Enable line for VME_A[] transceiver (active low)
VME_PROGRESS	1	Output	A VME cycle has been decoded and is in progress

All the above signals terminating with a trailing ‘b’ are active low.

3 Architecture

The core implements a VME64x compliant slave interface. To be connected to the VME backplane dedicated transceivers and drivers are needed.
In Figure 1 the block diagram of the core is represented.

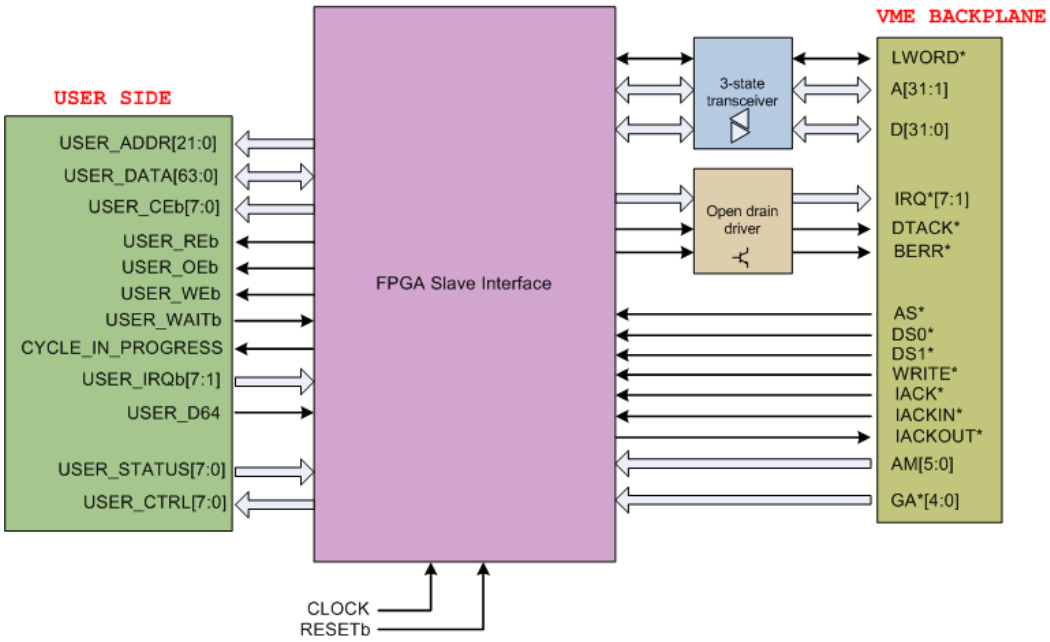

Figure 1: The core block diagram

In particular Data and Address lines must be connected using transceivers with high current driving capability, low ground bounce and +5V tolerance.
IRQ, DTACK and BERR lines can be connected to the backplane using open drain drivers. If 2eVME and/or 2eSST functionalities have to be used, the DTACK must provide fast rising edges and it must be connected to the backplane with a proper driver.
All other signals can be directly connected to the backplane paying attention only to the +5V compatibility between VME backplane and FPGA.

3.1 Data cycles

At the falling edge of Address Strobe (ASb) the following process is executed:

- ✓ The Address Modifier lines are decoded to identify various cycles:
  - o Single A24 cycles: non privileged and supervisor program and data space, CR/CSR space. They are all used to access configuration ROM and Control/Status registers.
  - o Single A32 cycles: non privileged and supervisor program and data space
  - o Block transfer A32 cycles: non privileged and supervisor
  - o 2eVME cycles: A32 3U and 6U, D32 and D64, master terminated only
  - o 2eSST cycles: A32 3U and 6U, D32 and D64, master terminated only

- ✓ The incoming VME address is loaded into a 32 bit up counter and its output is decoded:
  - o if an A24 cycle is detected the 5 MSB are compared with the 5 MSB stored in the BAR (see CR/CSR section);
  - o if a user cycle (A32) is detected the 8 MSB are compared against the 8 ADER registers (see CR/CSR section). In case of a positive match the corresponding USER_CEb line is activated.

- ✓ If a data cycle has been decoded, both A24 or A32, a state machine starts the process to handle the translation of the protocol, between VME and user side.

In Figure 2 a single read transaction is represented. In Figure 3 a single write transaction is represented. In Figure 4 a block read transaction is represented.

The width of both USER_REb and USER_WEb is one clock cycle, with the rising edge of the clock (not shown) at the end of the active low pulse.

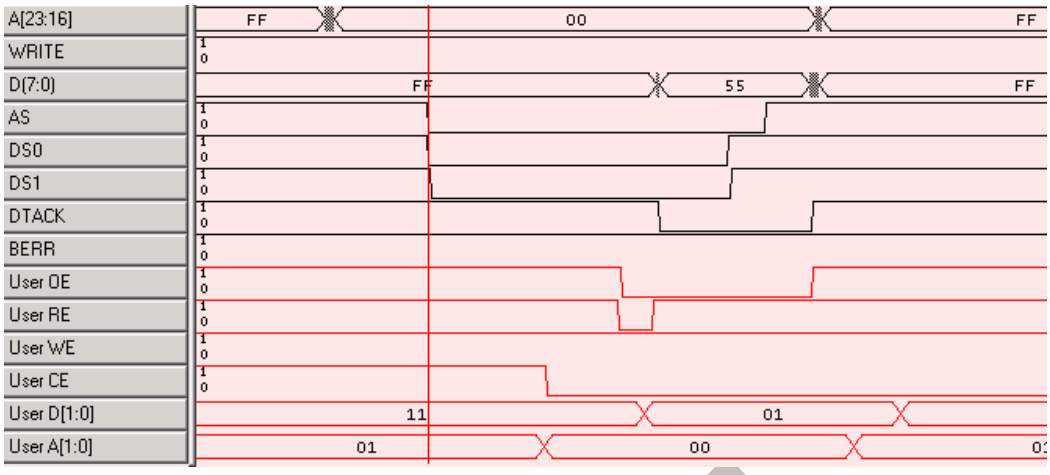

Figure 2: A32 D32 single read operation

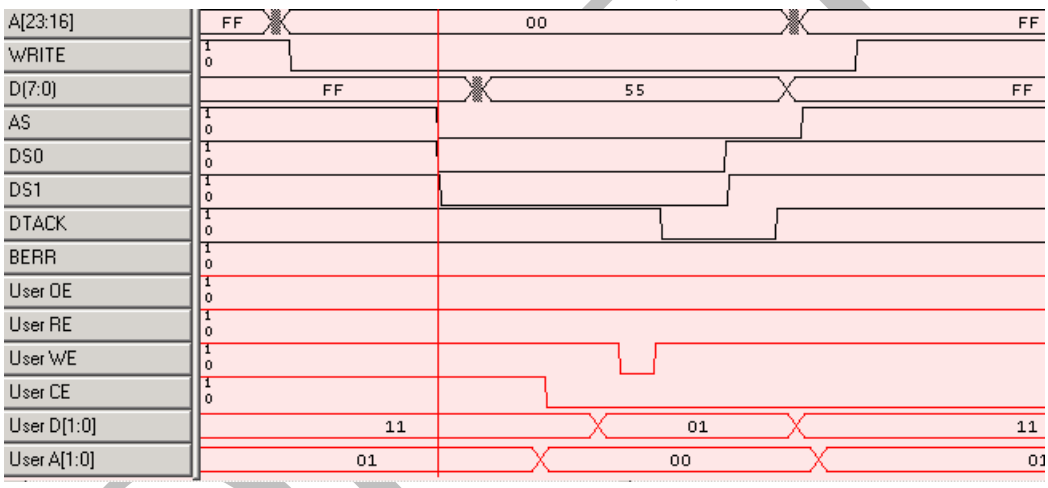

Figure 3: A32 D32 single write operation

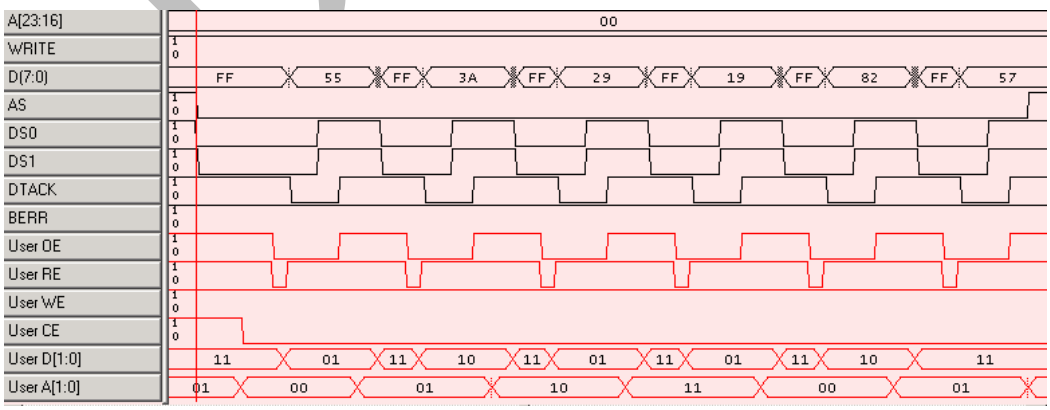

Figure 4: A32 D32 block read operation

The USER_OEb line is held active until the cycle is closed (rising edge of DTACK).

The user addresses are incremented every rising edge of the Data Strobes (DS0b, DS1b).
The user can delay the cycle activating the USER_WAITb signal (not shown): in this case the VME cycle lasts until the signal is deactivated. In this situation the VME bus go in a hang state and if it is held busy too long a timeout can occour and the master frees the bus with an error.
The USER_WAITb signal is sampled every cycle by the control state machine before activating USER_REb or USER_WEb.

Other cycles not shown, behave in the same way:

- a user space is selected with USER_CEb and the user addesses are stabilized;
- if a read operation is in progress USER_REb and USER_OEb are activated; User Addresses are stable 1 clock cycle before USER_REb pulse.
- if a write operation is in progress USER_WEb is activated. Both User Data and User Addresses are stable 1 clock cycle before USER_WEb pulse.

If 32 bit cycle are executed only the 32 LSB of the user data bus are used.
If the user deactivates the USER_D64 line (which must be considered a configuration static signal), the 64 bit cycles are not recognized and again only the 32 LSB of the user data bus are used.
Several devices can be seamlessy easily connected to these signals: static RAMs, FIFOs, registers, …
All the user logic connected to this core must be synchronous with the clock.
The user logic can use the VME_PROGRESS line to be informed when a data cycle is in progress.

The user cannot interrupt/suspend the 2e cycles: they are implemented as master terminated only.

3.2 Interrupts

The user can activate an IRQ line on the VME bus.
The USER_IRQb lines are directly mapped to the corresponding VME_IRQb lines.
The IRQ_ENABLE register (in the CSR space) enable the activation of the relative VME_IRQb line. Bit 0 of IRQ_ENABLE register is not used.
If any USER_IRQb line is sampled active for 3 consecutive clock cycles, the corresponding bit in te IRQ_IDENT register is set. If it has been enabeld the corresponding VME_IRQb line is also activated.
The IRQ_IDENT bit can be cleared under software control, writing it to ‘0’. Bit 0 of IRQ_IDENT register is not used.
Once an interrupt has been acknowledged this module puts the contents of the corresponding IRQ_ID register on the 8 LSB of the VME data bus and then propagartes the IACK line.
The interrupt controller acts as a ROAK, releasing the acknowledged interrupt line after issuing the corresponding IRQ_ID.

4

Memory Map & Registers

The core decodes various spaces: 512 kB standard CR/CSR and 8 user spaces (or, as indicated by the specifications, functions).

The CR/CSR space must be accessed with A24-D32 cycles. Any A24 single cycle is allowed, not only the dedicated one (AM=0x2F).

The base address of the CR/CSR space is stored in the Base Address Register (BAR), which has a default value, loaded after a RESETb, equal to the geographical address. The base address of the CR/CSR is the following:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAR[7:3]					**0**																		

BAR[7:3] = GA[4:0] after a RESETb pulse, and can be loaded with any value under software control, thus changing the address mapping of the CR/CSR space.

Eigth general purpose output lines are implemented as output of the USER_BSET register.
Eigth general purpose input lines are implemented and can be read through the USER_STATUS register.

4.1 Control and Status Registers list

Inside the CSR space the following 8 bit registers are implemented:

Name	Address	Access	Description
IRQ_IDENT	0x7FBD8	RW	Interrupt identification register
IRQ_ENABLE	0x7FBDC	RW	Interrupt enable register
IRQ_ID1	0x7FBE0	RW	VME interrupt vector for VME_IRQ[1]
IRQ_ID2	0x7FBE4	RW	VME interrupt vector for VME_IRQ[2]
IRQ_ID3	0x7FBE8	RW	VME interrupt vector for VME_IRQ[3]
IRQ_ID4	0x7FBEC	RW	VME interrupt vector for VME_IRQ[4]
IRQ_ID5	0x7FBF0	RW	VME interrupt vector for VME_IRQ[5]
IRQ_ID6	0x7FBF4	RW	VME interrupt vector for VME_IRQ[6]
IRQ_ID7	0x7FBF8	RW	VME interrupt vector for VME_IRQ[7]
USER_STATUS	0x7FBFC	R	Value of the USER_STATUS lines
ADER0	0x7FF60	RW	Adders DEcoder compaRe register 0
ADER1	0x7FF70	RW	Adders DEcoder compaRe register 1
ADER2	0x7FF80	RW	Adders DEcoder compaRe register 2

ADER3	0x7FF90	RW	Adders DEcoder compaRe register 3
ADER4	0x7FFA0	RW	Adders DEcoder compaRe register 4
ADER5	0x7FFB0	RW	Adders DEcoder compaRe register 5
ADER6	0x7FFC0	RW	Adders DEcoder compaRe register 6
ADER7	0x7FFD0	RW	Adders DEcoder compaRe register 7
USER_BCLR	0x7FFE8	RW	User Bit Clear (not used)
USER_BSET	0x7FFEC	RW	Set values on USER_CTRL lines
BCLR	0x7FFF4	RW	Bit Clear (not used)
BSET	0x7FFF8	RW	Bit Set (not used)
BAR	0x7FFFC	RW	CR/CSR Base address register

4.2 Configuration ROM contents

The Configuration Rom contents is the following:

Address	Len (bytes)	Value	Description
0x0000	1		Checksum
0x0004÷0x000C	3	0x001000	Length of ROM to be check-summed
0x0010	1	0x84	CR data access width
0x0014	1	0x84	CSR data access width
0x0018	1	0x02	VME64x CR/CSR space
0x001C	1	0x43	ASCII ‘C’
0x0020	1	0x53	ASCII ‘R’
0x0024÷0x002C	3	UserDef	Manufacturer ID
0x0030÷0x003C	4	UserDef	Board ID
0x0040÷0x004C	4	UserDef	Revision ID
0x007C	1	0x01	Program ID code
0x00B0÷0x00B8	3	0x07FBDB	Begin address offset of User CSR area
0x00BC÷0x00C4	3	0x07FBFF	End address offset of User CSR area
0x00E0	1	0xFC	Slave characteristics
0x00F4	1	0xFE	Interrupter capabilities
0x0100	1	0x84	Function 0 data access width
0x0104	1	0x84	Function 1 data access width
0x0108	1	0x84	Function 2 data access width
0x010C	1	0x84	Function 3 data access width
0x0110	1	0x84	Function 4 data access width
0x0114	1	0x84	Function 5 data access width
0x0118	1	0x84	Function 6 data access width
0x011C	1	0x84	Function 7 data access width
0x0120÷0x013C	8	0x00000003 0000EE00	Function 0 AM code mask
0x0140÷0x015C	8	0x0000003 0000EE00	Function 1 AM code mask
0x0160÷0x017C	8	0x0000003 0000EE00	Function 2 AM code mask

0x0180÷0x019C	8	0x0000003 0000EE00	Function 3 AM code mask
0x01A0÷0x01BC	8	0x0000003 0000EE00	Function 4 AM code mask
0x01C0÷0x01DC	8	0x0000003 0000EE00	Function 5 AM code mask
0x01E0÷0x01FC	8	0x0000003 0000EE00	Function 6 AM code mask
0x0200÷0x021C	8	0x0000003 0000EE00	Function 7 AM code mask
0x0220÷0x029C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 0 XAM code mask
0x02A0÷0x031C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 1 XAM code mask
0x0320÷0x039C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 2 XAM code mask
0x03A0-0x041C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 3 XAM code mask
0x0420÷0x049C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	Function 4 XAM code mask

		00000801	
0x04A0÷0x051C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 5 XAM code mask
0x0520÷0x059C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 6 XAM code mask
0x05A0÷0x061C	32	0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000801	Function 7 XAM code mask
0x0620÷0x062C	4	0xFF000000	Function 0 address decoder mask
0x0630÷0x063C	4	0xFF000000	Function 1 address decoder mask
0x0640÷0x064C	4	0xFF000000	Function 2 address decoder mask
0x0650÷0x065C	4	0xFF000000	Function 3 address decoder mask
0x0660÷0x066C	4	0xFF000000	Function 4 address decoder mask
0x0670÷0x067C	4	0xFF000000	Function 5 address decoder mask
0x0680÷0x068C	4	0xFF000000	Function 6 address decoder mask
0x0690÷0x069C	4	0xFF000000	Function 7 address decoder mask

All non listed values are read as '0', as required by the specifications.
The meaning of the various values can be found in the references.
The values defined as 'UserDef' (Manufacturer ID, Board ID, Revision ID) can be easily changed in the synthesis phase modifying 3 `defines placed at the beginning of the source file.

4.3 User functions

Up to 8 independent user address spaces (functions) can be configured.
Each function is identified by a USER_CEb signal, which is activated when the 8 MSB of the 32 bit address loaded into the address counter are matched against the corresponding Adders DEcoder compaRe (ADER) register. Each function can span up to 2^{22} = 4 Mwords of 32 or 64 bit, depending of the transfer size.

The size of the transfer can be only 32 bit and 64 bit. Transfer of 16 bit and 8 bit are not allowed: in this case the module issue a BUS ERROR.
Single 32 bit and block transfer are allowed. Are also allowed 2eVME and 2eSST transfer in 32 or 64 bit mode.
The user can decide to allow (and accept) only 32 bit transfers setting to '0' the USER_D64 line.

After a RESETb pulse the value of the ADER registers is the following:

ADER0

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**0**	**0**	**0**

ADER1

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**0**	**0**	**1**

ADER2

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**0**	**1**	**0**

ADER3

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**0**	**1**	**1**

ADER4

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**1**	**0**	**0**

ADER5

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**1**	**0**	**1**

ADER6

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**1**	**1**	**0**

ADER7

7	6	5	4	3	2	1	0
GA4	**GA3**	**GA2**	**GA1**	**GA0**	**1**	**1**	**1**

If an ADER register is set to 0x00, the corresponding USER_CEb line will be kept inactive. This implies that addresses from 0x00000000 up to 0x00FFFFFF will be never decoded and cannot be used.

5 Operation

TBD

6 Implementation

The model does not contain any architecture specific functionality, so in principle it can be synthetized for any technology, both ASIC or FPGA.
The core has been synthetized, implemented and tested in a Xilinx Spartan-3 device XC3S400-FG456 and it is running with a 40 MHz clock. About 550 slices (≈15 % of the device) have been used, including a monostable used to turn on a LED connected to the DTACK signal. The tool used was ISE 7.1. The board used was only D32.

Only a little problem has been experienced. The core needs a RESETb pulse after power up or configuration, because the power up reset puts all flip-flops to logic '0' state, while the RESETb pulse puts them to the user defined state. Both BAR and ADERs registers must be initialized with values different from 0.

Pay attention at the polarity of the following signals: VME_BERR, VME_DTACK and VME_IRQ[7:1]. They are generated active high intending to drive an inverter bus driver.
The polarity of VME_DATA_DIR and VME_ADDR_DIR siganls must also be checked to match what is needed by the transceivers.

At this stage the interface is not completely debugged. Fully simulation bot functional and post implementation was performed for all the functionalities described herein. Single transfers A24-D32, A32-D32 and block transfers A32-D32 are well debugged. The 2eVME read cycle has been tested and timing of the control signals was checked. Other 2e cycles are still not tested since a controller which handle these transaction is not available. The interrupt controller has still not been tested too.

7

References

- ANSI/VITA 1-1994 “American National Standard for VME64”
- ANSI/VITA 1.1-1997 “American National Standard for VME64 Extensions”
- ANSI/VITA 23-1998 “American National Standard for VME64 Extensions for Physics and Other Applications”
- VITA 1.5-1999 “Draft Standard for Trial Use Approved bi the VITA Standards Organization for the 2eSST”
- XILINX “Spartan-3 1.2V FPGA Family: Complete Data Sheet”