

VME64 Slave Controller

Version 2.2

Abstract:

This VME64 slave controller is designed for custom integration using standard FPGA and ASIC technologies. It is fully compliant to the VME specification supporting A16/A24/A32 address mode, D8/D16/D32 data modes (read/write/read-modify-write), D32-BLT, D64-MBLT, as well as interrupt acknowledge cycles. VMEbus timing is guaranteed by using a system clock of 40 Mhz or higher. A synchronous design approach is used to simplify interfacing to the asynchronous VMEbus. The user side interface is full synchronous. Data access is either single cycle or multi-cycle controlled through user wait states.

Deliverables of the core include RTL code, a self verifying system testbench, synthesis information and a user guide.

1 Overview

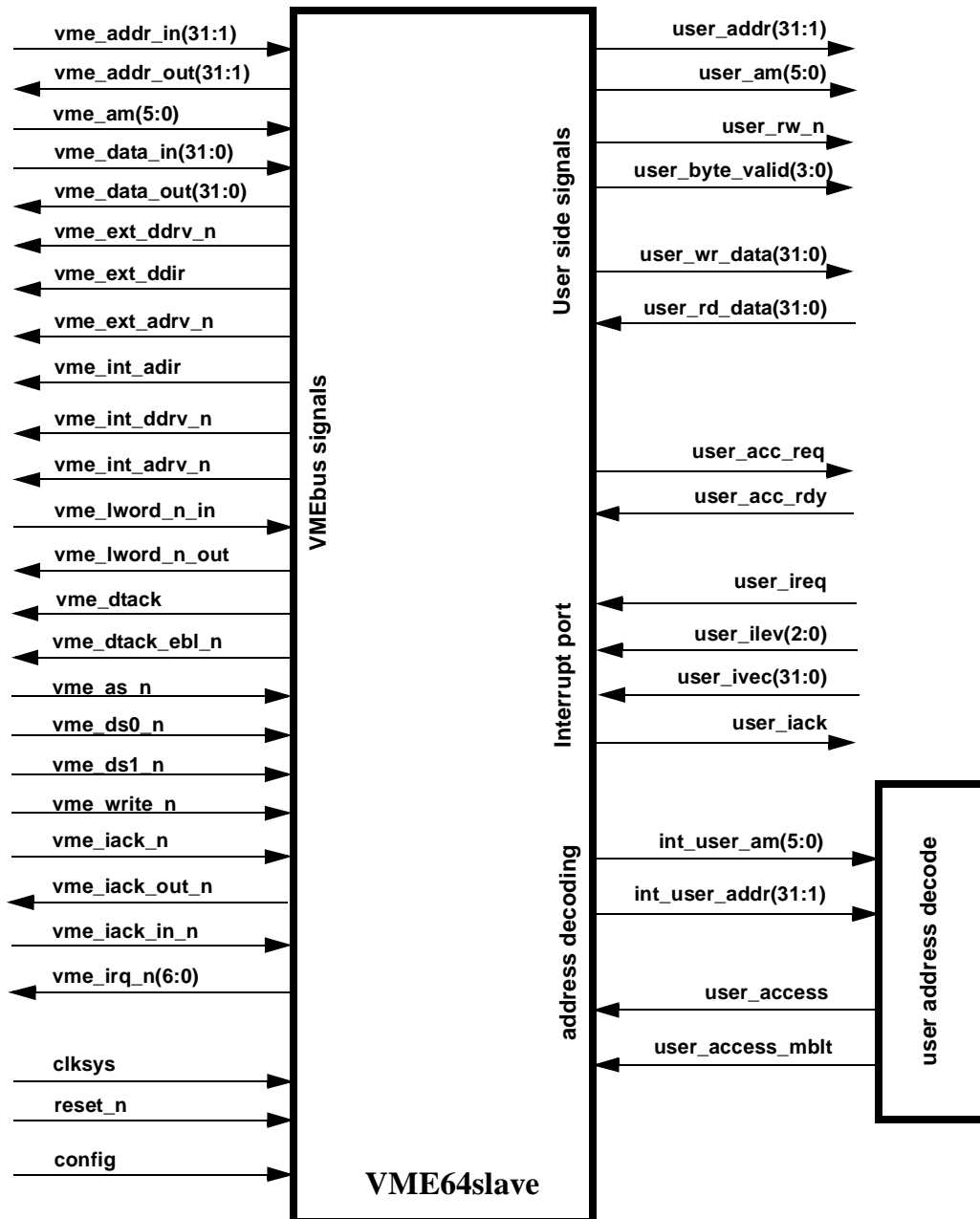
This VME slave controller core provides an easy-to-use, synchronous parallel interface towards custom logic (called user side), including full interrupt support. VME compatible external drivers can be directly connected to the VME. Examples are shown in the application section.

List of features:

- Data modes: D8, D16, D32, D32-BLT, D64-MBLT
- Address modes: A16, A24, A32
- Read, write, read-modify-write cycles
- Selectable rescinding DTACK
- Configurable D8, D16 or D32 interrupter
- Selectable little/big endian conversion
- Full synchronous user side interface for registers, peripherals and memories
- User selectable wait states

1.2 Inputs - Outputs

The following pictures shows all inputs and outputs:



2 IO description

The following part lists the input and output ports of the VME core and provides an overview of their functionality.

2.1 General inputs

These pins are used to clock and initialize the whole VME core. To guarantee VME compliance, the falling edge of `vme_as_n` is used to latch the `vme_addr` and the `vme_am` signals. The falling edge of the system clock '`clk`' is used to guarantee interface timing on some signals. All other registers use the rising edge of '`clk`' as the system clock. For proper operation of the VME interface, it is recommended that the system clock is 40 Mhz or higher.

pin name	type	description
<code>clk</code>	in	System clock
<code>reset_n</code>	in	Asynchronous system reset, active low

2.2 VME Bus

These pins are used to control data transfer through the VME interface.

pin name	type	description
<code>vme_addr_in[31:1]</code>	in	VME address bus input
<code>vme_addr_out[31:1]</code>	out	VME address bus output, used for MBLT
<code>vme_am[5:0]</code>	in	VME address modifier input
<code>vme_data_in[31:0]</code>	in	VME data bus input (from bus driver)
<code>vme_data_out [31:0]</code>	out	VME data bus output (goes to bus driver)
<code>vme_ext_ddrv_n</code>	out	Active low drive enable signal for external bidirectional data bus drivers.
<code>vme_ext_ddir</code>	out	Direction control signal for external bidirectional data bus drivers: '1' to VME bus '0' from VME bus
<code>vme_ext_adrv_n</code>	out	Active low drive enable signal for external bidirectional address/lword drivers.
<code>vme_ext_adir</code>	out	Direction control signal for external bidirectional address/lword drivers: '1' to VME bus '0' from VME bus
<code>vme_int_ddrv_n</code>	in	Active low drive enable signal for internal bidirectional data bus drivers.
<code>vme_int_adrv_n</code>	in	Active low drive enable signal for internal bidirectional address/lword drivers.
<code>vme_lword_n_in</code>	in	VME long word access indicator, low active
<code>vme_lword_n_out</code>	out	VME long word access indicator output, used for MBLT
<code>vme_dtack</code>	out	Data transfer acknowledge. Used to indicate whether the DTACK is drive low or high (for rescinding)

pin name	type	description
vme_dtack_ebl_n	out	Data transfer acknowledge driver output, active low. This is the enable signal of the external DTACK driver.
vme_as_n	in	VME address strobe: clocks with falling edge the internal synchronisation signals like vme_addr and vme_am. vme_as_n is also used as data signal for access start detection.
vme_ds0_n	in	Data strobes 0, active low
vme_ds1_n	in	Data strobes 1, active low
vme_write_n	in	Read/write signal, active low
vme_iack_n	in	Interrupt acknowledge, active low
vme_iack_in_n	in	Interrupt acknowledge daisy chain, active low
vme_iack_out_n	out	Interrupt acknowledge daisy chain, active low
vme_irq_n[6:0]	out	Interrupt, active low. Have to be connected to open collector driver.

2.3 Configuration

The core can be configured and optimized for a particular application. Prior to synthesis, these values should be fixed to the chosen configuration.

pin name	type	description
config.rsc_dtack	in	Rescinding DTACK enable '0': Disabled '1': Enabled
config.interrupter	in	Interrupter selection 8: D08(O) type interrupter 16: D16 type interrupter 32: D32 type interrupter
config.endian	in	Endian selection for user side interface '0': Big endian, transparent '1': Little endian

2.4 User Side Interface

This part describes the user side interface, which is designed for easy register and memory access.

2.4.1 Signal description

The entire user side interface is part of the clk system-clock domain. The application dependent address modifier logic is not integrated in the core. This allows the user to build his own address decoding logic without changing the code of the VME. For convenience, more signals are provided than a minimal approach would require. This simplifies the design of the user side logic. Timing information can be found in paragraph 3 *Timing*.

pin name	type	description
int_user_addr(31:1)	out	Registered VME address bus (by falling edge of vme_as_n)

pin name	type	description
int_user_am(5:0)	out	Registered VME address bus modifier (by falling edge of vme_as_n)
user_access.ebl	in	User access signal. The user has 50 ns time to decode the address and asserting the user_access signal when addressed.
user_access.blr	in	BLT user access signal. The user has 50 ns time for decoding of the address and address modifier and to assert the user_access.blr signal when BLT is addressed.
user_access.mblr	in	MBLT user access signal. The user has 50 ns time for decoding of the address and address modifier and to assert the user_access_mblr signal when MBLT is addressed.
user_acc_req	out	User access request: Active high until user_acc_rdy acknowledges the request (or VME bus error occurs)
user_acc_rdy	in	User side acknowledgement signal. Active one event which finishes user side access.
user_addr(31:1)	out	Registered VME address bus
user_am(5:0)	out	Registered VME address bus modifier
user_wr_data(31:0)	out	Write data bus.
user_rd_data(31:0)	in	Read data bus. Must be valid when user_acc_rdy is 1.
user_rw_n	out	Data read/write_not signal. '1': read data '0': write data
user_byte_valid(3:0)	out	User data byte valid signal. Indicates which byte of the user_wr_data/user_rd_data bus are valid or requested. Bit (0): user_wr_data(7:0) is valid Bit (1): user_wr_data(15:8) is valid Bit (2): user_wr_data(23:16) is valid Bit (3): user_wr_data(31:24) is valid
user_ireq	in	Interrupt request. Active one indicates that an interrupt is pending and a VME interrupt will be generated. Must return to zero with user_iack = 1.
user_iack	out	Interrupt acknowledgement. An active one event indicates the end of a valid interrupt acknowledge cycle.
user_ilev(2:0)	in	Interrupt level
user_ivec(31:0)	in	Interrupt vector. Depending on the interrupter configuration, the core responds as D08(O), D16 or D32 interrupter.

2.4.2 User address decoding

To decide if an access is for the actual card or not, a decode logic has to be added externally. The signals `int_user_addr` and `int_user_am` allow a standard memory mapped address decoding scheme. The address modifiers may be used to differentiate between user/system access, io or memory etc. It is not mandatory to use the address modifiers.

The address modifier codes (`vme_am`) are used in the user address decoding module to detect a BLT or MBLT cycle. If a valid MBLT cycle is detected, this module has to assert `user_access.mblt` together with `user_access.ebl` to indicate a valid cycle. For a BLT cycle, `user_access.blit` has to be asserted.

The `int_user_addr` and `int_user_am` signals are latched with the falling edge of `vme_as_n` and are immediately valid. The user address decode logic has to be designed for a delay time shorter than 2 clock cycles. This gives more flexibility for the user and lowers the constraints for this part of logic.

3 Timing

This paragraph describes the timing of the VME slave core.

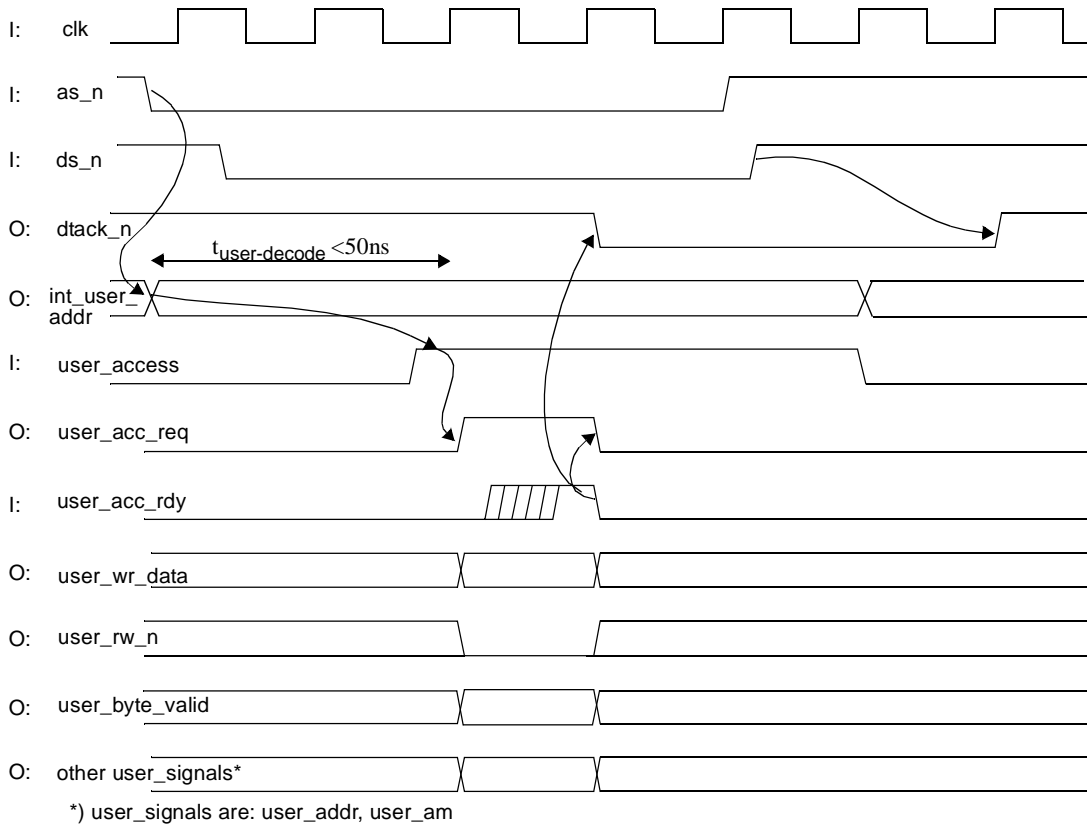
3.1 General Information

All VMEbus inputs are synchronized to the local clock domain. A frequency of at least 40 MHz is required to detect all level changes on the VME bus correctly.

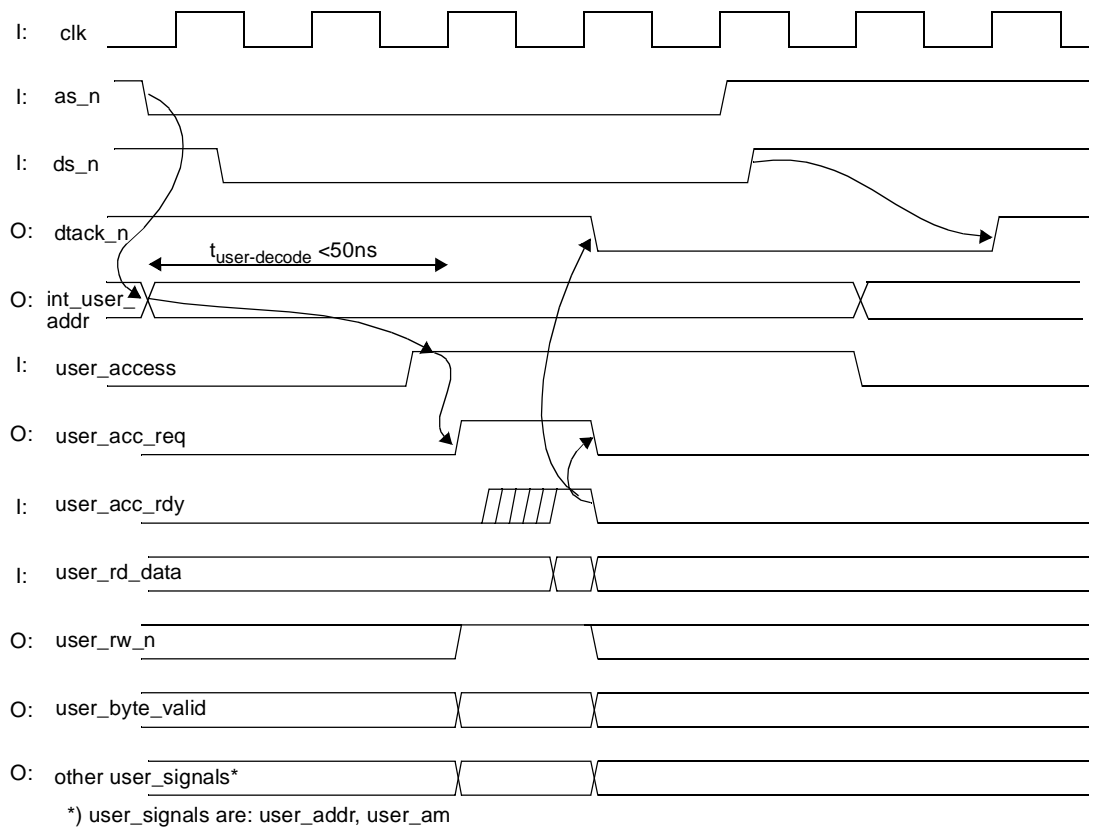
Outputs are all synchronous to the VME clock domain. Only `int_user_addr` and `int_user_am` are registered by the falling edge of `vme_as_n` signal.

3.2 User access timing

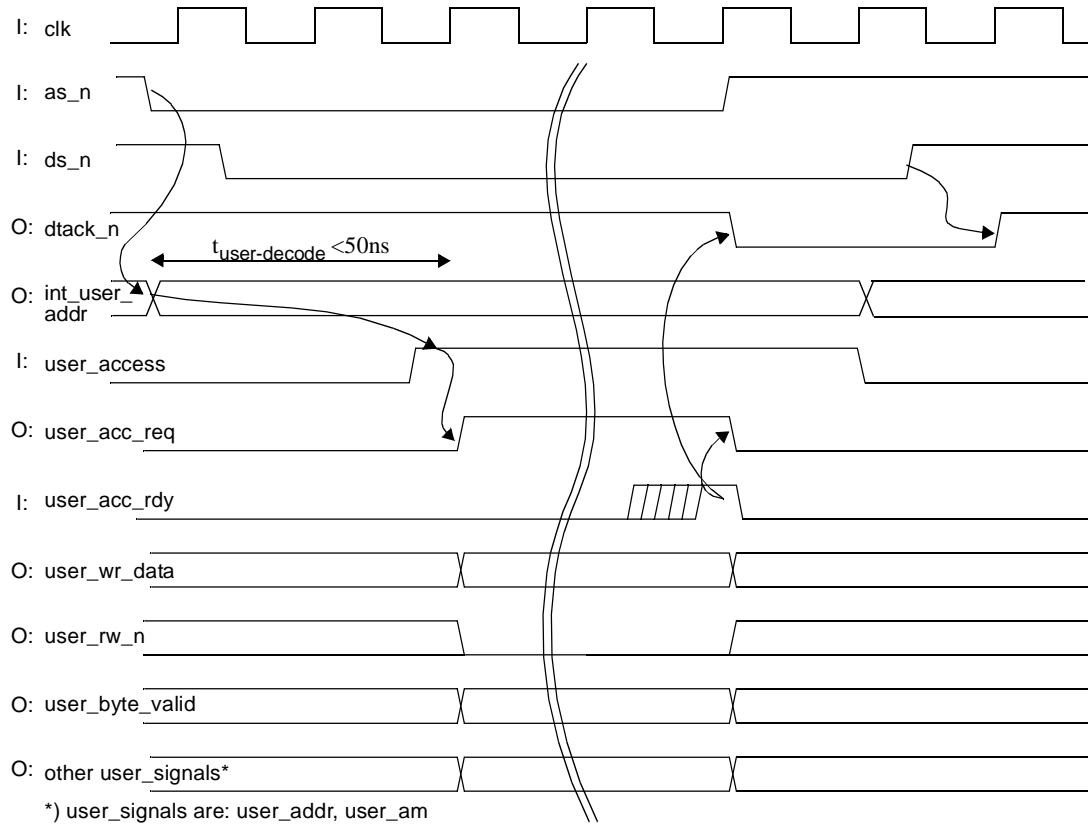
3.2.1 Write access, no user wait states



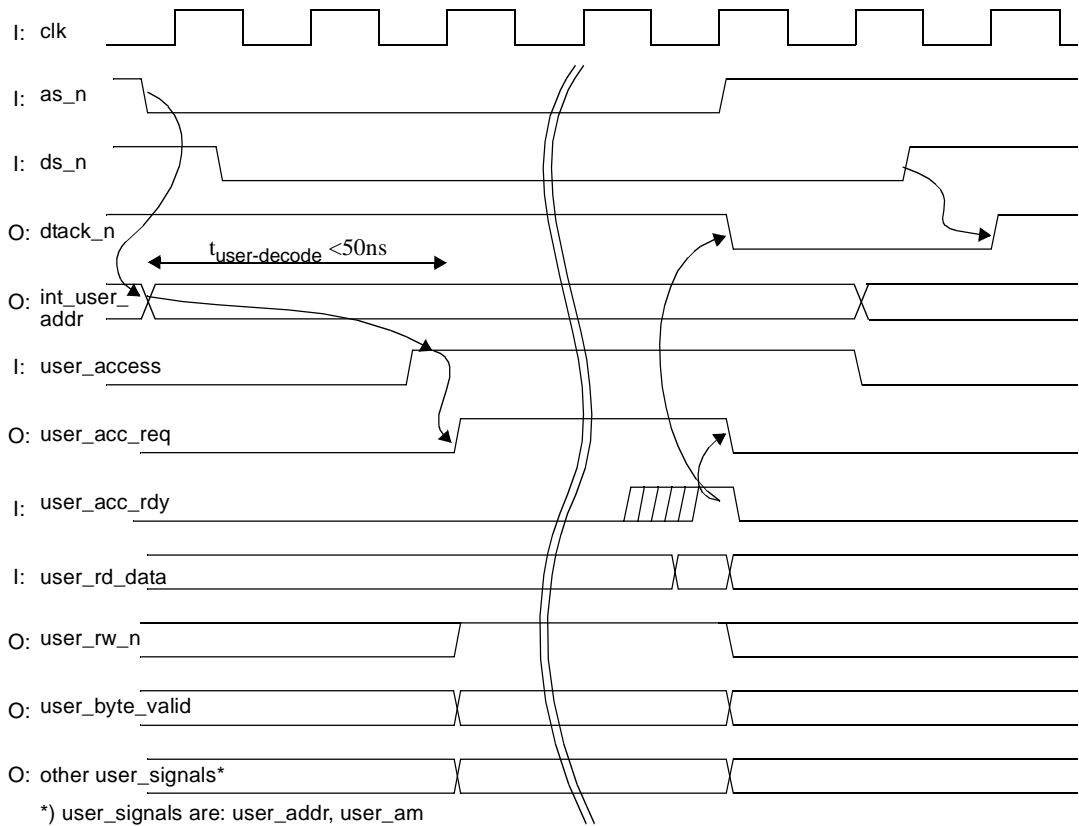
3.2.2 Read access, no user wait states



3.2.3 Write access, user wait states



3.2.4 Read access, user wait states



3.2.5 Configurable Interrupter

Interrupt Acknowledge Cycles

Through a constand definition in the setup file, it is possible to define to what kind of interrupt cycles this VME core responds. Following options are available

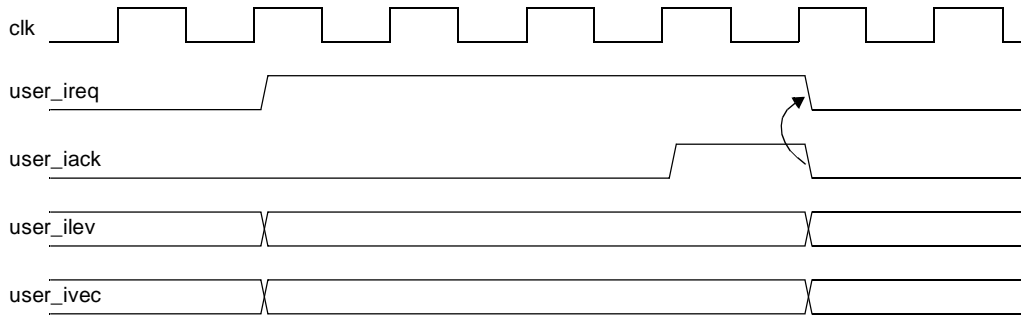
- D08(O) Interrupter: Responds to D08(O), D16 and D32 interrupt cycles
- D16 Interrupter: Responds to D16 and D32 interrupt cycles
- D32 Interrupter: Responds to D32 interrupt cycles

Interrupt Scheme

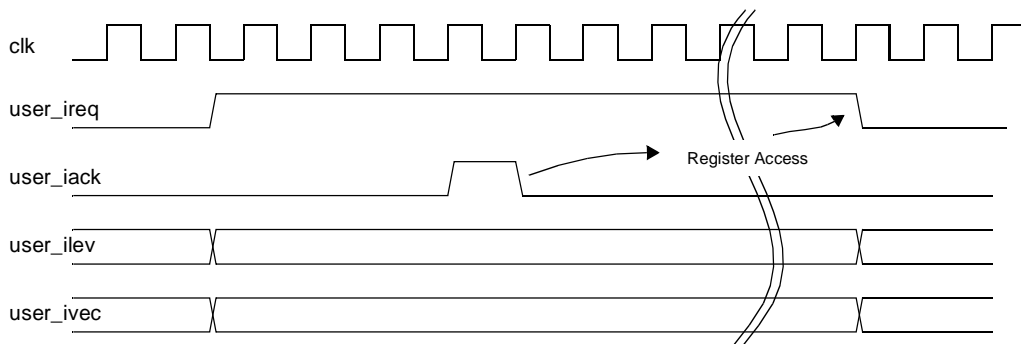
Interrupt requests to the VME bus are signalled by the active high user_ireq. Depending on the user_ilev (interrupt level), the vme_irq_n(x) will be asserted. As soon as the interrupt is acknowledged by the VME bus, the user_iack event is asserted. If the interrupter uses the ROAK (Release On AcKnowledge) scheme, then the user_ireq has to be released immediately. If it uses the RORA (Release On Register Access) scheme then the user_ireq has to be released when the interrupt source is acknowledged.

Note: user_ilev and user_ivec have to be stable for the whole time period where user_ireq is high.

Timing using ROAK scheme:



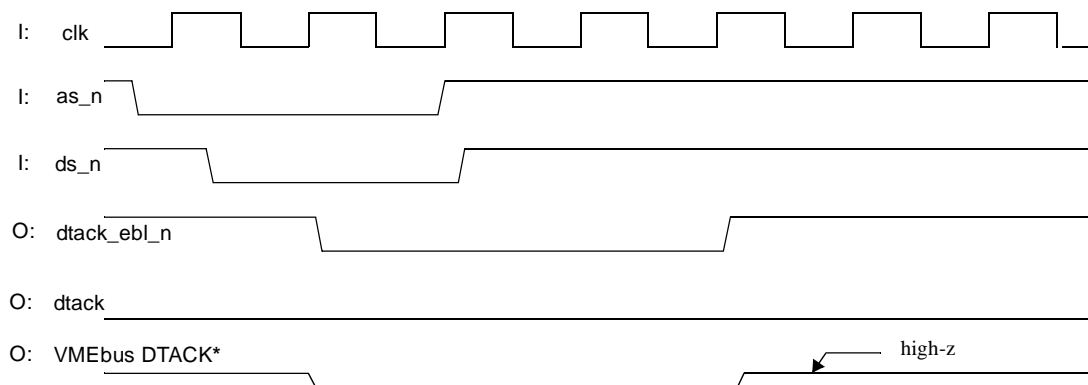
Timing using RORA scheme:



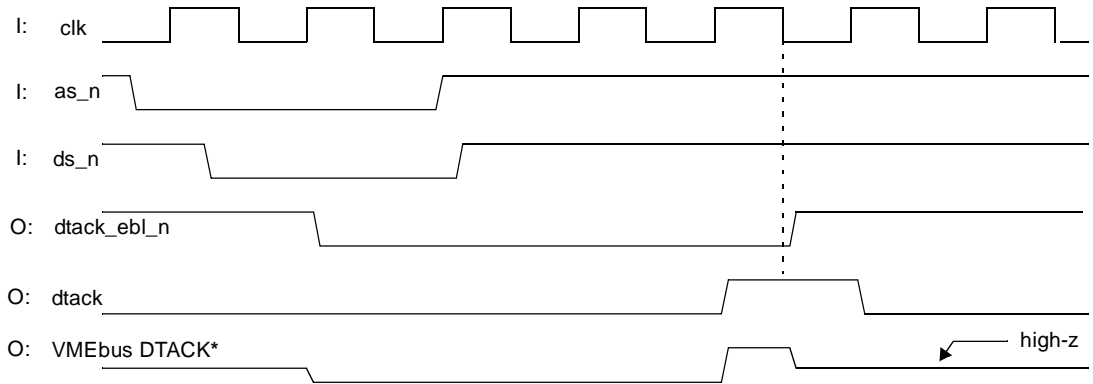
3.2.6 Rescinding DTACK

The VME64 specification allows DTACK to be operated as a rescinding signal instead of an open-collector class signal. This results in an accelerated bus cycle. This feature can be selected through config.rsc_dtack = '1'.

Timing diagram with open-collector DTACK:



Timing diagram with rescinding DTACK:

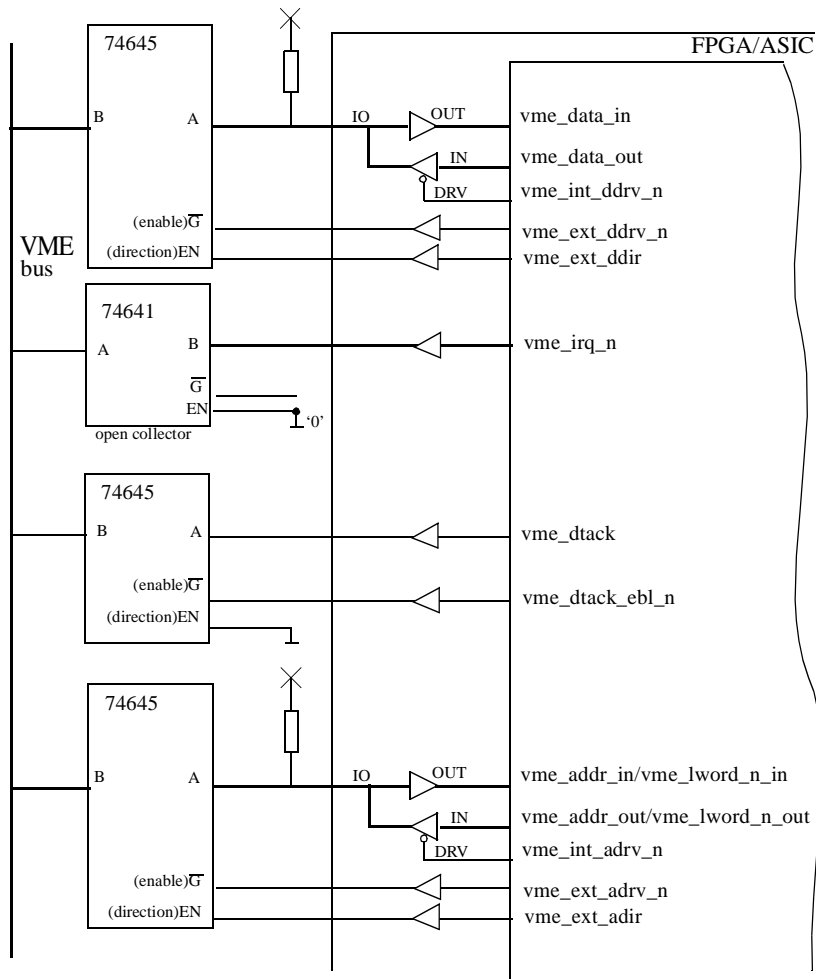


Note: One register clocked with the *falling* edge of clk is used to generate the by 1/2 clock cycle expanded dtack_ebl_n signal. Therefore, the rescinding DTACK is active HIGH for 1/2 clock period before being try-stated.

4 Applications

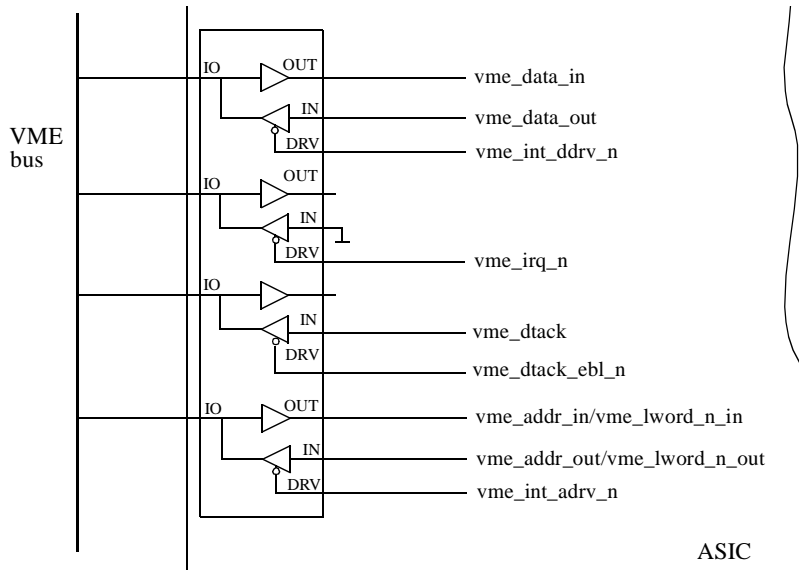
The VME specification requires 64 mA bus drivers. The bus drivers are not integrated in the VME core. Since the FPGA are not able to drive this load, external drivers might be used. The driver should have the following functions:

This diagram shows how external standard parts can be used to interface with the VME bus. The used 74xx parts have to fulfill VME specifications. Please note that outputs like vme_irq_n etc. are push-pull outputs and require an external open collector driver like the 74641.



If rescinding DTACK is not required then DTACK can be driven by an open collector driver as is vme_irq_n.

This diagram shows how internal buffers can be used to directly interface with the VME bus. These drivers have to fulfill VME specifications.



5 References

- The VMEbus Specification, ANSI/IEEE STD1014-1987
- American National Standard for VME64, ANSI/VITA 1-1994



About Inicore

- FPGA and ASIC Design
- Easy-to-use IP Cores
- System-on-Chip Solutions
- Consulting Services
- ASIC to FPGA Migration
- Obsolete ASIC Replacements

Inicore is an experienced system design house providing FPGA / ASIC and SoC design services. The company's expertise in architecture, intellectual property, methodology and tool handling provides a complete design environment that helps customers shorten their design cycle and speed time to market. Our offering covers feasibility study, concept analysis, architecture definition, code generation and implementation. When ready, we deliver you a FPGA or take your design to an ASIC provider, whatever is more suitable for your unique solution.

Customer Advantages

We offer one-stop shopping for everything from the specifications to the chip or module solution. Our experience and fast turnaround time reduces your development costs and increases your returns from the market. Your system is not limited by the level of expertise and standard chip solutions you happen to have in-house. Achieve market success by differentiating and optimizing your product. Reusability builds the basis for further developments in the ever-decreasing product life cycle.

Visit us @ www.inicore.com

INICORE, INC. has made every attempt to ensure that the information in this document is accurate and complete. However, INICORE, INC. assumes no responsibility for any errors, omissions, or for any consequences resulting from the information included in this document or the equipments it accompanies. INICORE, INC. reserves the right to make changes in its products and specifications at any time without notice.

© 2003 INICORE, INC. All rights reserved.

Inicore, Inc.
5600 Mowry School Road, Suite 180
Newark, CA 94560
tel: 510 445 1529 fax: 510 656 0995 email: info@inicore.com