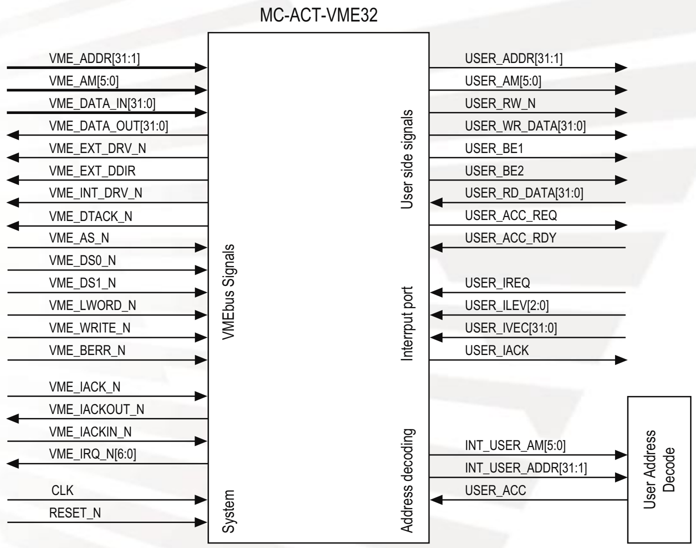
AvnetCore: Datasheet

VME32



MC-ACT-VME32 Pinout

The MC-ACT-VME32 core is used as interface for the VME standard bus. One side contains all VME bus signals and the other side all the user signals. With the defined address and address modifier, the user allows any masters on the VME bus to access the IO, peripherals or memory placed on the user side.

The user has to describe two blocks which are connected to the “address decoding” and to the “user side”. The “address decoding” is used to detect the access and to allow the transfer on the corresponding board. It allows the user to build its own address decoding without changing the code of the MC-ACT-VME32.

The MC-ACT-VME32 core provides a full interrupt controller based on seven interrupt lines connected to the bus. The system release the interrupt on the acknowledge (ROAK). The acknowledge is done on all boards connected on the bus through a daisy-chain.

A complete VHDL test bench verifies every functions and addressing mode and interrupts. These test benches are built as a self testing regression-test suite.

Intended Use:

— Medical systems

— Industrial controls: robotic, factory automation

Features:

— Flexible slave VME controller

— Full interrupt controller (ROAK)

— Control signals for external drivers and drivers on chip

— Synchronous user side interface for registers, peripherals and memories

— User definable waitstates

— Synchronous, reliable design

— Expandable to full set of VME features

— Silicon proven design

Targeted Devices:

— SX-A Family

— Axcelerator® Family

— ProASIC^{PLUS}® Family

Core Deliverables:

— Netlist Version

 > Netlist compatible with the Actel Designer place and route tool

— RTL Version

 > VHDL Source Code

 > Test Bench

— All

 > User Guide

Synthesis and Simulation Support:

— Synthesis: Synplicity®

— Simulation: Model*Sim*®

— Other tools supported upon request

Verification:

— Test Bench

Actel CompanionCore

AVNET® Memec

Functional Description

The falling edge of VME_AS (address strobe) will synchronize all the addresses (VME_ADDR and VME_AM) allowing the controller to decode them in order to define if the present board is addressed or not. Since this moment, all signals on the VME bus have to be stable and the controller will execute the command depending on the control signals (VME_DS0_N, VME_DS1_N, VME_WRITE_N, VME_LWORD_N, VME_IACK_N). The VME master has to release the VME_AS_N signal at the end of a transfer to execute a new command.

WRITE DATA TRANSFER

When the VME_WRITE_N defines a write data transfer, the controller will assign the address (VME_ADDR) on the USER_ADDR bus, the address modifier (VME_AM) on the USER_AM bus and the data (VME_DATA) on the USER_DATA bus. The signals VME_DS0_N and VME_DS1_N select the corresponding data location according the following table:

Data Locations Selected	VME_DS0_N	VME_DS1_N	VME_ADDR01	VME_LWORD_N
VME_DATA_IN(7:0)	Low	High	Low	High
VME_DATA_IN(15:8)	High	Low	Low	High
VME_DATA_IN(7:0)	Low	High	High	High
VME_DATA_IN(15:8)	High	Low	High	High
VME_DATA_IN(31:8)	High	Low	Low	Low
VME_DATA_IN(23:0)	Low	High	Low	Low
VME_DATA_IN(23:16)	Low	Low	High	Low
VME_DATA_IN(15:0)	Low	Low	Low	High
VME_DATA_IN(15:0	Low	Low	High	High
VME_DATA_IN(31:0)	Low	Low	Low	Low

To execute the transfer on the user part, the controller will active a request signal (USER_ACC_REQ) with valid control signals. The data locations selected are enabled with the signals USER_BE1 (bits 7:0), USER_BE2 (bits 15:8), USER_BE3 (bits 23:16) and USER_BE4 (bits 31:24). The transfer will be ended with the acknowledge of the user part (USER_ACC_ACK).

Once the transfer executed, the controller will acknowledge the data transfer on the VME bus with the VME_DTACK_N. When seeing this acknowledge, the master will release the VME_AS_N signal ending the actual data transfer.

READ DATA TRANSFER

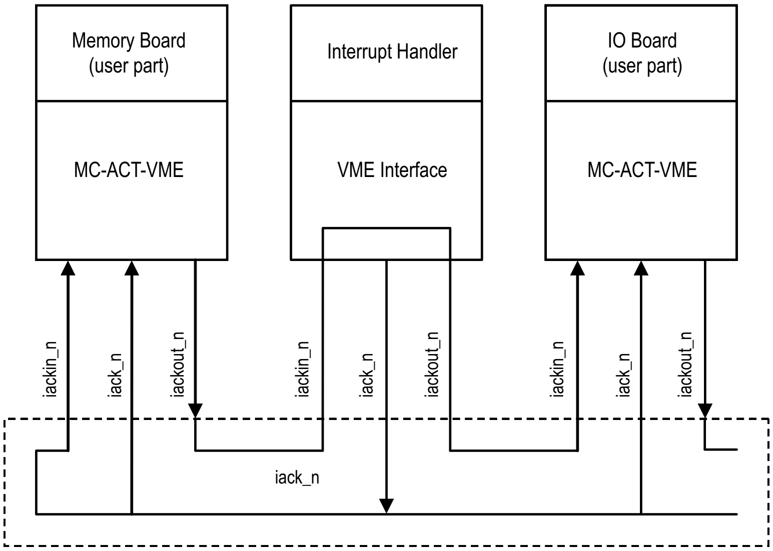
When the VME_WRITE_N defines a read data transfer, the controller will assign the address (VME_ADDR) on the USER_ADDR bus and the address modifier (VME_AM) on the USER_AM bus. As the write data transfer, the signals USER_BE1/2/3/4 are depending on the VME_DS0/1_N, VME_ADDR(1) and VME_LWORD_N.

The controller will active a request signal (USER_ACC_REQ) until the acknowledge (USER_ACC_ACK) coming from the user part. The read data have to be valid during this acknowledge.

Once ready, the data are transferred on the VME_DATA_OUT bus and acknowledged with the signal VME_DTACK_N. When seeing this acknowledge, the master will release the VME_AS_N signal ending the actual data transfer.

INTERRUPT

The interrupts on the VME bus are generated by the different modules connected on the bus and are acknowledged through a daisy-chain interrupt line as shown on the figure below:



Between the user part and the VME controller, the transfer of interrupt information is based on the request/acknowledge protocol.

When an interrupt occurs on the user part, it generates an interrupt request on the VME controller through the signal USER_IREQ. Depending on the value defined by the user part on the vector USER_ILEV, the VME controller activates an interrupt on the respective interrupt line VME_IRQ_N(x) (USER_ILEV = 1 -> VME_IRQ_N(1), USER_ILEV = 7 -> VME_IRQ_N(7)). Interrupt level 7 has the highest priority and interrupt level 1 has the lowest.

Since there, the interrupt handler acknowledges the interrupt by asserting a low state on the signal VME_IACK_N. This signal will be transmitted to all modules and as well in the first module of the daisy-chain. The modules which didn't generate an interrupt just assert the VME_IACKIN_N signal to the VME_IACKOUT_N. A module which generated an interrupt will detect the acknowledge at the falling edge and compare the VME_ADDR(3:1) with the interrupt level register (USER_ILEV) to determine if it has the priority (depending on the level) to execute the interrupt or not. In case of not, it will assign the VME_IACKIN_N to VME_IACKOUT_N. In the other case (the corresponding module has the priority), the high state is asserted to the VME_IACKOUT_N signal in order to break the daisy-chain and avoid other modules taking the interrupt acknowledge. During the acknowledge, the module transfers the interrupt vector register defined by the user (USER_IVEC) on the VME data bus (VME_DATA_OUT). When the data are valid, the VME controller drives the VME_DTACK_N signal low, allowing the master and interrupt handler executing next commands.

On the user part, the interrupt is acknowledged with the signal USER_IACK which clears the pending interrupt request.

Device Requirements

Family	Device	Utilization			Performance
		COMB	**SEQ**	**Total**	
SX-A	A54SX72A-STD	220 (6%)	196 (10%)	416 (7%)	62 MHz
ProASIC3	A3PE600-STD	n/a	n/a	536 (4%)	82 MHz
ProASIC^{PLUS}	APA600-STD	n/a	n/a	672 (3%)	58 MHz
Axcelerator	AX500-STD	216 (4%)	196 (8%)	412 (5%)	86 MHz

Table 1: Device Utilization and Performance

Verification and Compliance

Complete functional and timing simulation has been performed on the VME using ModelSim 5.5e. This core has also been used successfully in customer designs.

Signal Descriptions

The following signal descriptions define the IO signals.

Signal	Direction	Description
CLK	Input	**Clock:** System clock. This clock is provided on the board and doesn't coming from the VME bus
RESET_N	Input	**System reset:** Asynchronous system reset, active low.
VME_ADDR[31:1]	Input	**VME Address Bus:** The smallest addressable unit is the byte location. Masters use address lines to select the data which has to be accessed.
VME_AM[5:0]	Input	**VME Address Modifier Bus:** Allow the master to pass additional binary information to the slave during data transfer cycles.
VME_DATA_IN[31:0]	Input	**VME Data Bus In:** 32 write data lines are available. Depending on the control signals, only one or two byte(s) can be used for the transfer on the user part.
VME_DATA_OUT[31:0]	Output	**VME Data Bus Out:** 32 read data lines. For each access, the 32 data bits are read.
VME_EXT_DRV_N	Output	**VME External Data Drive:** Active low drive enable signal for external bidirectional data bus drivers.
VME_INT_DRV_N	Output	**VME Internal Data Drive:** Active low drive enable signal for internal bidirectional data bus drivers.
VME_EXT_DDIR	Output	**VME External Data Direction:** Direction control signal for internal bidirectional data bus drivers. High indicates data to VME bus and low from VME bus.
VME_LWORD_N	Input	**VME Long Word:** Active low signal indicating long word access.
VME_DTACK_N	Output	**VME Data Acknowledge:** This active low signal acknowledges the data transfer. It has to be connected to an open collector driver.
VME_AS_N	Input	**VME Address Strobe:** clocks with falling edge the internal synchronization signals like VME_ADDR and VME_AM. It is also used as data signal for access start detection.
VME_DS0_N	Input	**VME Data Strobe 0:** Active low signal used for the selected location part of the data.

Signal	Direction	Description
VME_DS1_N	Input	**VME Data Strobe 1:** Active low signal used for the selected location part of the data.
VME_WRITE_N	Input	**VME Read/Write:** Active low signal which is used by the master to indicate the data direction.
VME_BERR_N	Input	**VME Bus Error:** Active low signal driven by other modules indicating that the data transfer was unsuccessful.
VME_IACK_N	Input	**VME Interrupt Acknowledge:** When driven low, the VME_IACKIN_N causes the IACK daisy-chain driver, located in slot 1, to propagate a falling edge down the interrupt acknowledge daisy-chain.
VME_IACKIN_N	Input	**VME Interrupt Acknowledge Input Daisy-Chain:** This active low signal is used as input of the module for the daisy-chain interrupt acknowledge.
VME_IACKOUT_N	Output	**VME Interrupt Acknowledge Output Daisy-Chain:** This active low signal is used as output of the module for the daisy-chain interrupt acknowledge.
VME_IRQ_N[6:0]	Output	**VME Interrupt Request Lines:** Interrupters request interrupts by driving an interrupt request line low. VME_IRQ_N[7] has the highest priority. These signals have to be connected to open collector drivers.
INT_USER_ADDR[31:1]	Output	**Registered VME Address Bus:** Synchronized on the falling edge of VME_AS_N. This bus is used to decode the address and to active an access signal on the user part.
INT_USER_AM[5:0]	Output	**Registered VME Address Modifier Bus:** Synchronized on the falling edge of VME_AS_N. This bus is used to decode the address modifier and to active an access signal on the user part.
USER_ACCESS	Input	**User Access Signal:** The user has 50 ns time to decode the address and asserting the USER_ACCESS signal when addressed.
USER_ACC_REQ	Output	**User Access Request:** Active high signal which requests for an access on the VME bus. Valid until USER_ACC_RDY acknowledges the request (or VME bus error occurs).
USER_ACC_RDY	Input	**User Side Acknowledgement Signal:** Acknowledge the request of the access on the VME bus of the user part.
USER_ADDR[31:1]	Output	**User Address:** Address used for the access on the user part.
USER_AM[5:0]	Output	**User Address Modifier:** Address modifier used for the access on the user part.
USER_WR_DATA[31:0]	Output	**User Write Data:** The data[31:24] are valid while USER_BE4 is high, data[23:16] while USER_BE3, data[15:8] while USER_BE2 and data[7:0] while USER_BE1.
USER_RD_DATA[31:0]	Input	**User Read Data:** The read data has to be valid when USER_ACC_RDY is high.
USER_RW_N	Output	**User Read/Write Signal:** A low signal indicates that the data are written in the user part and a low signal, that the data are read.
USER_BE1	Output	**User Byte 1 Enable:** Active high signal enabling the low byte [7:0] of the data.
USER_BE2	Output	**User Byte 2 Enable:** Active high signal enabling the high byte [15:8] of the data.
USER_BE3	Output	**User Byte 3 Enable:** Active high signal enabling the high byte [23:16] of the data.
USER_BE4	Output	**User Byte 4 Enable:** Active high signal enabling the high byte [31:24] of the data.
USER_IREQ	Input	**User Interrupt request:** This active high signal indicates that an interrupt is pending on a VME interrupt will be generated. It will return to zero with USER_IACK active.
USER_IACK	Output	**User Interrupt Acknowledge:** An active one event which indicates the end of a valid interrupt acknowledge cycle.
USER_ILEV[2:0]	Input	**User Interrupt Level:** This bus indicates the level of priority of the pending interrupt. It will generate the corresponding VME_IRQ.
USER_IVEC[31:0]	Input	**User Interrupt Vector:** The interrupt vector will be transmitted on the VME data bus during the acknowledgement of the interrupt.

Table 2: VME Core Signal List

Recommended Design Experience

For the source version, users should be familiar with HDL entry and Actel design flows. Users should be familiar with Actel Libero v2.2 Integrated Design Environment (IDE) and preferably with Synplify and Model*Sim*.

Ordering Information

The CORE is provided under license from Avnet Memec for use in Actel programmable logic devices. Please contact Avnet Memec for pricing and more information.

Information furnished by Avnet Memec is believed to be accurate and reliable. Avnet Memec reserves the right to change specifications detailed in this data sheet at any time without notice, in order to improve reliability, function or design, and assumes no responsibility for any errors within this document. Avnet Memec does not make any commitment to update this information.

Avnet Memec assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction, if such be made, nor does the Company assume responsibility for the functioning of undescribed features or parameters. Avnet Memec will not assume any liability for the accuracy or correctness of any support or assistance provided to a user.

Avnet Memec does not represent that products described herein are free from patent infringement or from any other third-party right. No license is granted by implication or otherwise under any patent or patent rights of Avnet Memec.

AvnetCore products are not intended for use in life support appliances, devices, or systems. Use of a AvnetCore product in such application without the written consent of the appropriate Avnet Design officer is prohibited.

All trademarks, registered trademarks, or service marks are property of their respective owners.

Contact Information:

North America
10805 Rancho Bernardo Road
Suite 100
San Diego, California 92127
United States of America
TEL: +1 858 385 7500
FAX: +1 858 385 7770

Europe, Middle East & Africa
Mattenstrasse 6a
CH-2555 Brügg BE
Switzerland
TEL: +41 0 32 374 32 00
FAX: +41 0 32 374 32 01

Ordering Information:

Part Number	Hardware	Resale
MC-ACT-VME32-NET	Actel Core Netlist	Contact for pricing
MC-ACT-VME32-VHD	Actel Core VHDL	Contact for pricing

Actel CompanionCore

AVNET Memec

www.em.avnet.com/actel