INICORE
System-on-Chip from Spec to Silicon

Optimized for

Actel CompanionCore

VME64 Master Controller

This core is obsolete and not recommended for new designs. Please use the VME System Controller core VMESCmodule instead.

Description

The VME64M core is a VME compliant master controller for use in FPGA and ASIC based implementations. It is designed to provide master capabilities to VME cards where increased data throughput or added features are required, that can't be provided by the VME64S slave core.

With a minimal system clock of 40 MHz, the VME bus timing is guaranteed. A fully synchronous user side interface simplifies system integration by hiding any issues interfacing to the asynchronous VME bus. If necessary, the user logic can delay bus cycles by introducing user wait-states if a peripheral such as a FIFO or a shared memory needs more time to finish the requested operation.

The VME64M core is not a replacement for a VME slot 0 controller, but it is intended to be used on advanced peripheral cards.

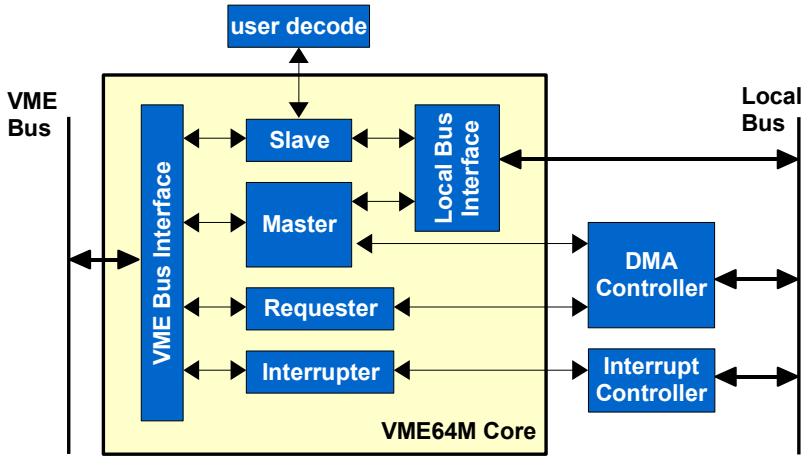

Figure 1: Block Diagram VME64 Master core

The above figure shows the main building block of the VME64M core complemented with some typical user logic modules.

Utilization and Performance Table Optimized for Actel Devices

Family	Device	Utilization				Performance
	- (speed grade)	s-mod	c-mod	RAM	Total	[MHz]
Fusion	AFS1500-2	570	1572		7.00%	80
IGLOO	AGL1000V5	574	1537		10%	52
ProASIC3	A3P1000-2	574	1567		10%	80
ProASIC3L	A3P1000L-2	560	1523		10%	60
Axcelerator	AX500-2	590	1010		20%	121
RTAX	RTAX1000S-1	590	1010		9%	87

Features

ANSI/VITA 1-1994 compliant

Master controller

- Addresses: A16, A24, A32
- Data modes: D8(EO), D16, D32, D32-BLT, D64-MBLT
- Supports data read-ahead and posted-write for increased throughput
- Constant local bus address for DMA transfers to/from FIFOs

Slave controller

- Addresses: A16, A24, A32
- Data types: D8(EO), D16, D32, D32-BLT, D64-MBLT
- Access modes: Read, write, read-modify-write
- Selectable rescinding DTACK
- Provides bid-endian to little-endian conversion option

Interrupter

- D8(O) D16, D32
- Supports RORA and ROAK

Bus Requester

- Supports RWD and ROR arbitration schemes
- FAIR requester
- Supports early withdrawal of bus request

Local Bus interface

- Fully synchronous bus inter-face for user logic
- User selectable wait-states
- Optional big-endian to little endian conversion

Interfaces

Pin Name	Type	Description
Global Signals		
clk_sys	in	System clock, min. 40 MHz
reset_n	in	Asynchronous system reset
VME Bus Interface		
vme_addr_int_in[31:1]	in	VME address bus input
vme_addr_int_out[31:1]	out	VME address bus output
vme_lword_int_in_n	in	Long word access indicator
vme_lword_int_out_n	out	Long word access indicator
vme_addr_drv_n	out	Enable signal for vme_addr and vme_lword_n drivers
vme_addr_dir	out	Address direction control
vme_addr_int_drv_n	in	Enable signal for internal addr and lword_n drivers
vme_data_int_in[31:0]	in	VME Data bus input
vme_data_int_out [31:0]	out	VME Data bus out
vme_data_drv_n	out	Data bus driver enable
vme_data_dir	out	Data direction control
vme_data_int_drv_n	out	Internal data driver enable
vme_am_int_in[5:0]	in	VME address modifier in
vme_am_int_out[5:0]	out	VME address modifier out
vme_write_int_in_n	in	Read/write signal input
vme_write_int_out_n	out	Read/write signal output
vme_am_dir	out	External direction control
vme_am_int_drv_n	out	Internal am drive enable
vme_dtack_int_in_n	in	Data transfer acknowledge
vme_dtack_out_in_n	out	Data transfer acknowledge
vme_dtack_dir	out	Dtack direction control
vme_dtack_drv_n	out	External dtack drive enable
vme_dtack_int_drv_n	out	Internal dtack driver enable
vme_as_int_in_n	in	VME address strobe in
vme_as_int_out_n	out	VME address strobe out
vme_as_dir	out	External direction control
vme_as_int_drv_n	out	Internal drive enable
vme_ds0_int_in_n	in	Data strobe 0 input
vme_ds0_int_out_n	out	Data strobe 0 output
vme_ds1_int_in_n	in	Data strobe 1 input
vme_ds1_int_out_n	out	Data strobe 1 output
vme_ds_dir	out	Data strobe direction control
vme_ds_int_drv_n	out	Internal ds drive enable
vme_br_out_n[3:0]	out	VME bus request out
vme_br_in_n[3:0]	in	VME bus request in
vme_bg_in_n[3:0]	in	Bus grant daisy chain in
vme_bg_out_n[3:0]	out	Bus grant daisy chain out
vme_bbsy_n	in	VME bus busy
vme_bclr_n	in	VME bus clear
vme_retry_n	in	VME retry indicator
vme_berr_n	in	VME bus error indicator
vme_iack_n	in	Interrupt acknowledge
vme_iack_in_n	in	Interrupt acknowledge daisy chain in
vme_iack_out_n	out	Interrupt acknowledge daisy chain out
vme_irq_n[6:0]	out	Interrupt request
User Side Interface		
user_acc_req	out	Data access request
user_acc_ack	in	User side acknowledgment
user_addr[31:2]	out	Registered address bus
user_am[5:0]	out	Address bus modifier
user_data_out[31:0]	out	Local data write bus
user_data_in[31:0]	in	Local data read bus
user_rwn	out	Data read/write access
user_byte_valid [3:0]	out	Data data byte valid
user_slv_mstm	out	User Slave – Master access
Slave Access Decoder		
int_user_addr[31:1]	out	Registered address bus
int_user_am[5:0]	out	Address bus modifier
user_access_ebl	in	User access indicator
user_access_blt	in	BLT user access indicator
user_access_mblt	in	MBLT user access indicator
user_access_addr_inc	in	Address increment indicator
Interrupter		
user_ireq	in	Interrupt request
user_iack	out	Interrupt acknowledgment
user_ilev[2:0]	in	Interrupt level
user_ivec[7/15/31:0]	in	Interrupt vector
Bus Requester		
dtb_req	in	Data transfer bus request
dtb_ack	out	Data transfer bus acknowledgment
dtb_pri[1:0]	in	Data transfer bus priority
dtb_clr_req	out	Data transfer bus clear req
dtb_config_fair	in	FAIR bus request observer
VME Master		
dt_request	in	Data transfer request
dt_ack	out	dt acknowledgment
dt_nack	out	dt not-acknowledgment
dt_status error[1:0]	out	Data transfer error status
dt_status_beat_count[9:0]	out	dt status beat count
dt_cmd_addr[31:0]	in	Data transfer address
dt_cmd_am[5:0]	in	dt address modifier
dt_cmd_rwn	in	Data transfer read/write
dt_cmd_size[9:0]	in	Data size (0 – 1024 beats)
dt_cmd_width[2:0]	in	Data width
dt_cmd_lbai_ebl	in	Local bus address incremental enable
dt_cmd_lbpw_ebl	in	Local bus posted write enable
dt_cmd_lbra_ebl	in	Local bus read ahead enable

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.

Deliverables

The core is available as Actel optimized netlist or as RTL version.

Actel Optimized Netlist:

- Netlist for target FPGA, EDIF, Verilog and VHDL format
- User Guide

RTL Source Code:

- VHDL source code
- Self-verifying system-level testbench
- Synthesis script
- User guide

51420.71.01 Oct/2008

INICORE INC. 5600 Mowry School Road Suite 180 Newark, CA 94560 Tel: 510 445 1529 Fax: 510 656 0995 www.inicore.com