

Macrocad Development Inc.

VCM 1012 DMA Model

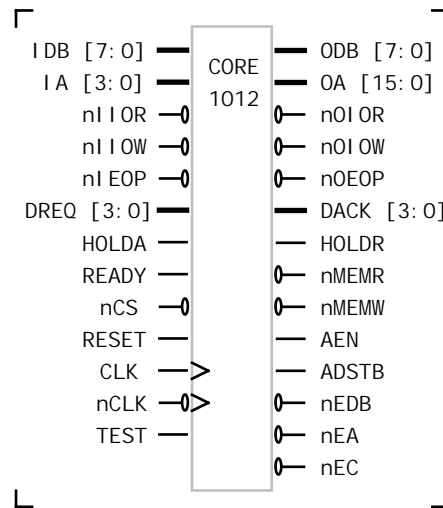
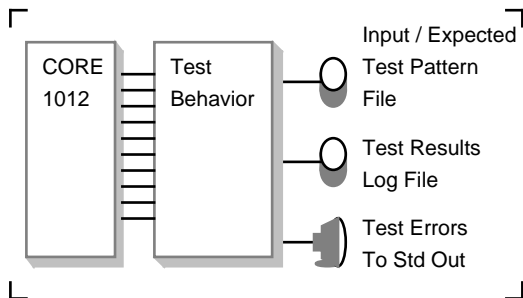
DMA Virtual Component Model
 Synthesizable Behavior Model
 8237 Compatible
 Test Driver Included

Description:

Macrocad's VCM 1012 model is a synthesizable 8237 compatible DMA Controller. The design is synchronous, avoiding silicon based delay chains. Synchronous design and small module size assures worry free synthesis. Well commented code provides insight into operations. Test vectors and a test bench support model are included. The test vectors are applied to the unidirectional pins of the core function. All tristate and bidirectional signals are contained in a shell model. The bidirectional shell contains no functions, other than that of IO buffers. Two clock phases are used to provide compatibility with the 8237 and ISA and EISA specifications. All 16 bits of address are available at the core interconnect, making the external upper 8 bit address latch optional. Separate IO strobe and arbitration state machines make for easy modifications.

Features:

- ◇ Synchronous design
- ◇ Design reuse becomes reality
- ◇ Modular design provides flexibility
- ◇ Synthesizable RTL code
- ◇ Well commented code for clarity
- ◇ Regression tests are made easy
- ◇ Test behavior is included
- ◇ Approx. 4500 (asic) gates
- ◇ Available in Verilog and VHDL versions
- ◇ Functional partitioning
- ◇ Designed *by* hardware engineers *for* hardware engineers
- ◇ Programmable arbitration
- ◇ Registers can be implemented with SRAM



VCM 1012



Architecture

The VCM 1012 DMA controller model is functionally partitioned to provide the most flexibility for various ASIC and FPGA implementations. The IO buffer shell level contains the only bidirectional signals in the model. The functional units are all contained in the core level. At this level, various functional units are connected to provide specific

capabilities for the overall DMA operations. All synchronous functions are clocked on the positive edge of one of two clock phases. (There are two clock phases to accommodate the 8237 specification which describes signals changing on the negative edge of the clock.)

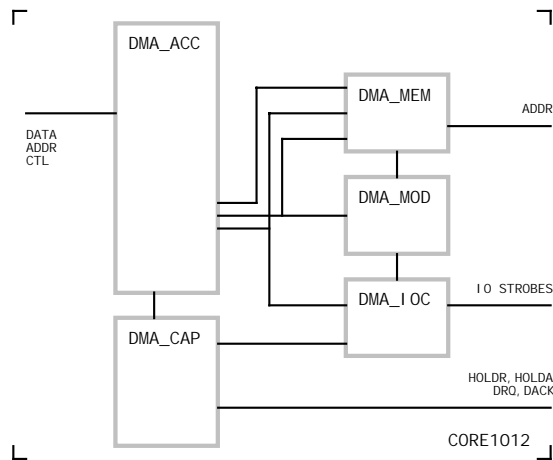


figure 1

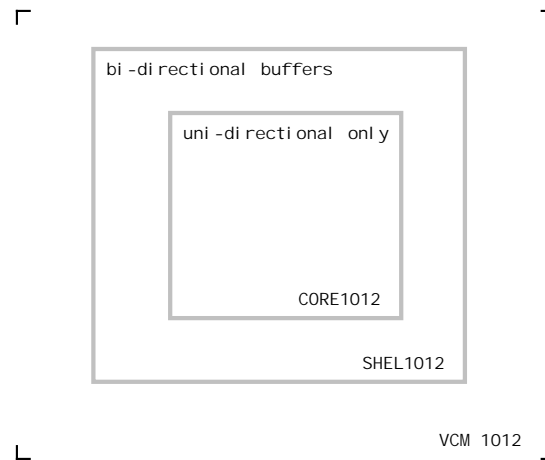


figure 2

VCM 1012



Pin List

Shell	Core	Type	Description
DB7		Bi -di rect	Programmi ng Data bit 7
DB6		Bi -di rect	Programmi ng Data bit 6
DB5		Bi -di rect	Programmi ng Data bit 5
DB4		Bi -di rect	Programmi ng Data bit 4
DB3		Bi -di rect	Programmi ng Data bit 3
DB2		Bi -di rect	Programmi ng Data bit 2
DB1		Bi -di rect	Programmi ng Data bit 1
DB0		Bi -di rect	Programmi ng Data bit 0
	ODB [7: 0]	Out	Programmi ng Data bus
	IDB [7: 0]	In	Programmi ng Data bus
	NEDB	Out	Programmi ng Data bus enable
A7		Bi -di rect	address bit 7
A6		Bi -di rect	address bit 6
A5		Bi -di rect	address bit 5
A4		Bi -di rect	address bit 4
A3		Bi -di rect	address bit 3
A2		Bi -di rect	address bit 2
A1		Bi -di rect	address bit 1
A0		Bi -di rect	address bit 0
	MADD [15: 0]	Out	address bus
	IA [7: 0]	In	address bus
	NEA	Out	address bus enable
NI OR		Bi -di rect	IO read
	NOIOR	Out	IO read output
	NI IOR	In	IO read input
NI OW		Bi -di rect	IO write
	NOIOW	Out	IO write output
	NI IOW	In	IO write input
NEOP		Bi -di rect	End of process
	NOEOP	Out	End of process output
	NI EOP	In	End of process input
	NEC	Out	Control signal output enable
DREQ3		In	DMA request bit 3
DREQ2		In	DMA request bit 2
DREQ1		In	DMA request bit 1
DREQ0		In	DMA request bit 0
	DREQ [3: 0]	In	DMA request bus
NTEST	TEST	In	Test - not used
HOLDA	HOLDA	In	Hold Acknowl edge (from processor)
READY	READY	In	Ready
NCS	NCS	In	Chip select
RESET	RESET	In	Reset
CLK	CLK	In	Clock (posi tive phase)
	NCLK	In	Clock (negati ve phase)
DACK3		Out	DMA acknowl edge bit 3
DACK2		Out	DMA acknowl edge bit 2
DACK1		Out	DMA acknowl edge bit 1
DACK0		Out	DMA acknowl edge bit 0
	DACK [3: 0]	Out	DMA acknowl edge bus
HOLDR	HOLDR	Out	Hold request (to processor)
NMEMR	NMEMR	Out	Memory read
NMEMW	NMEMW	Out	Memory write
AEN	AEN	Out	Address enable (DMA active)
ADSTB	ADSTB	Out	Address strobe

VCM 1012



Register Address Map:

ADDR	REGISTER	OP	CH	DESCRIPTION	@RESET
0000	CMADD	W/R	0	current memory address	00
0000	BMADD	W	0	base memory address	00
0001	CTCNT	W/R	0	current transfer count	00
0001	BTCNT	W	0	base transfer count	00
0002	CMADD	W/R	1	current memory address	00
0002	BMADD	W	1	base memory address	00
0003	CTCNT	W/R	1	current transfer count	00
0003	BTCNT	W	1	base transfer count	00
0004	CMADD	W/R	2	current memory address	00
0004	BMADD	W	2	base memory address	00
0005	CTCNT	W/R	2	current transfer count	00
0005	BTCNT	W	2	base transfer count	00
0006	CMADD	W/R	3	current memory address	00
0006	BMADD	W	3	base memory address	00
0007	CTCNT	W/R	3	current transfer count	00
0007	BTCNT	W	3	base transfer count	00
0008	CMD	W	All	command register	00
0008	STATUS	R	All	status register	00
0009	REQ	W	All	request register	00
0009	- na -	R		reserved	
000A	MASK	W	All	set / clear single ch. mask	FF
000A	- na -	R		reserved	
000B	MODE	W	All	Mode for selected channel	00
000B	- na -	R		reserved	
000C	CBP	W	- na -	Clear byte pointer	na
000C	- na -	R		reserved	
000D	MCLR	W	- na -	Master clear	na
000D	TEMPD	R		Temporary data transfer register	00
000E	MASK	W	All	clears all masks	FF
000E	- na -	R		reserved	
000F	MASK	W	All	set / clear all ch. mask	FF
000F	- na -	R		reserved	

VCM 1012



Register Bit Assignments:

ADDR	REGISTER	OP	DESCRIPTION	@RESET
0008	CMD	W	command register - all channels 7: 1=DACK active hi, 0=DACK active lo 6: 1=DREQ active lo, 0=DREQ active hi 5: 1=early write, 0=late write 4: 1=rotating priority, 0=fixe 3: 1=compressed cycle, 0=normal 2: 1=controller disable, 0=enable 1: 1=CHO addr hold, 0=no hold 0: 1=mem to mem enable, 0=disable	00
0008	STATUS	R	status register 7: CH3 request active 6: CH2 request active 5: CH1 request active 4: CH0 request active 3: CH3 EOP active 2: CH2 EOP active 1: CH1 EOP active 0: CH0 EOP active	00
0009	REQ	W	request register 7-3: reserved 2: 1=set request, 0=clear request 1-0: CH select 3:0	00
000A	MASK	W	set / clear single ch. Mask 7-3: reserved 2: 1=set mask, 0=clear mask 1-0: CH select 3:0	FF
000B	MODE	W	modes for selectd ch. 7-6: 3=cascade 2=block 1=single 0=demand 5: 1=addr decrement, 0=increment 4: 1=autoinit enable, 0=disable 3-2: 2=mem read 1= mem write 0=verify 1-0: CH select 3:0	00
000C	CBP	W	clear the byte pointer 7-0: reserved	
000D	MCLR	W	master clear 7-0: reserved	
000D	TEMPD	R	read temporary data x-fer reg 7-0: data	00
000E	MASK	W	clears all masks 7-0: reserved	FF
000F	MASK	W	set / clear all ch. Mask 7:4 reserve 3: 1=set mask CH3, 0=clear 2: 1=set mask CH2: 0=clear 1: 1=set mask CH1, 0=clear 0: 1=set mask CH0, 0=clear	FF

Functional Description:

The VCM1012 core functionality is subdivided into a hierarchy of sub-functions (modules). They are listed below:

- MODULE DEFINITIONS:
- 1 DMA_ACC Register access logic includes MASK and REQ registers
 - 2 DMA_MEM Transfer and address register array
 - 3 DMA_MOD Mode registers
 - 4 DMA_IOC IO bus controller includes temporary data register
 - 5 DMA_CAP Central arbitration point

1 DMA_ACC

The DMA_ACC module is responsible for access to the register set. In addition, it contains the Mask Register, the Command Register, the Status Register, and the Request Register. These registers are located in this module, because they are global to the overall DMA macro, and not specific to one channel. They also have an effect on the selection of

VCM 1012



the channel specific registers in that they can block the access to these registers during a DMA transfer operation.

The DMA_ACC module decodes the address presented to it from the IO bus. It then generates a signal to access the particular register requested. It does this in a straight-forward way for some registers, (a specific register has a specific address decode), and is more complex for other, (they might use the Byte Pointer). Since this section handles access to the internal DMA registers, it also manages the channel selection for these registers during DMA transfer operations. The status of the transfers on the various channels is part of this section, as well as the global operational parameters.

With the exception of some storage functions optionally described as to transparent latches, this module is synchronous in nature. Even in the cases where transparent latches are used, they are controlled by synchronous signals, and not by external pin connections. Thus with adequate set up and hold time guaranteed, they behave as synchronous functions external to the module. The data bus is synchronized before being presented to the other modules. In addition, the data bus is set to 00hex when reset is active. Thus, circuitry in the DMA_MEM module can use the data bus to reset the Register Array to a known value at reset without incurring a large gate count penalty simply by writing each byte in the Register Array at reset.

1.1 The Address Decodes

The address decodes take the address on the IO pins and perform a straight binary decode of the address

for all 16 decodes. These decodes are used to access some registers directly, and some indirectly. The register map is a good reference for the address decode as it relates to registers and functions.

1.2 The Byte Pointer

The byte pointer is used to access 16 bit registers one byte at a time. This is done without using extra address decodes. Since there are more than 16 registers in the macro function, the byte pointer is used to access sequentially the lower and upper bytes of these 16 bit registers. Registers controlled by the byte pointer are the Memory Address Registers, and the Transfer Count Registers.

1.3 The Mask Register

The Mask Register masks requests for DMA service. This happens at the beginning of a DMA transfer. Typically the last thing done in a programming sequence for a channel is that the mask bit for that channel is cleared. This allows any request to immediately be serviced by the DMA macro. At the end of a transfer, if auto-initialize is not active for that channel, the mask bit will be set automatically. The Mask Register does not affect the software Request Register.

1.4 The Software Channel Request Register

A DMA service request can be initiated by "software", or program control. This is typically used for memory to memory transfers since the memory subsystem typically does not have any mechanism to request DMA service.

VCM 1012



1.5 The Status Register

The Status Register provides information regarding all channels of the DMA controller. It shows the request, and end of transfer information for each of the 4 channels.

1.6 The Command Register

The Command Register provides operating parameters which affect all channels. The exception to this is the memory to memory transfer and channel 1 address increment bits.

1.7 The Channel Select Logic

The Register Array Channel Selects are generated by the DMA_ACC module because the mask register is located here. The DMA_CAP module presents this module with the Channel Select bits, and these are gated with the Channel Selects from the programming interface, since both paths need access to the Register Array.

1.8 The Register Read Funnel

The Register Read Funnel selects the appropriate register for presentation to the programming interface (IO pins). The read of the registers is an asynchronous select, but there are no memory elements in the path for this operation, just an address decode and the gating of the selected register on to the output data bus. The address decode to read these registers is shown in the Address Map section. The Byte Pointer works with both the reading and writing of the Memory Address and Transfer Count Registers.

1.9 The IO Read / Write Control Strobe Processing

The IO interface pins are assumed to be asynchronous. These pins are therefore processed to present a synchronous strobe and data to the registers for programming.

2 DMA_MEM

2.1 The Register Array is 4 ranks of 64 bit wide memory elements which are based either on transparent latches or edge flip flops. The 64 bit wide data is assigned as follows:

	REGISTER NAME
	REGISTER ARRAY BITS
=====	
	CURRENT TRANSFER ADDRESS
[15: 0]	[63: 48]
	BASE TRANSFER ADDRESS
[15: 0]	[47: 32]
	CURRENT MEMORY ADDRESS
[15: 0]	[31: 16]
	BASE MEMORY ADDRESS
[15: 0]	[15: 0]

Each channel has a 64 bit wide "slice" of this Register Array. Thus there are 4 ranks, and when a channel is selected for programming or a DMA transfer operation, all of the above registers associated with that channel are selected for reading and writing. The Register Array is contained in a separate sub module, so that specific synthesis requirements can be met, such as static ram modules, etc.

2.2 The Increment Loop

Since the Register Array can be implemented with transparent latches, the output of the Register Array goes immediately to 4 16 bit wide "edge triggered" registers. The Memory Address and Transfer Count outputs of these registers are presented to a set of 'ALU's. There are two of these simple increment / decrement 'ALU's. One is for the Current Address Register, and one is for the Current Transfer Register. They are both 16 bits wide. The output of all 64 bits of the Edge

VCM 1012



Registers, the 16 "ALUed" Next Address Bits, the 16 "Next" Transfer Bits, the 8 bits from the programming interface, are presented to a mux at the input to the Register Array. Thus by selecting from the "next address" (for increment or decrement), the Base Address, the programming interface data byte, or the unchanged output of the Edge Registered version of the Current Address, the Memory Address Register can be accessed for programming, reset, DMA transfers, and End of Transfer Auto initialize operations.

3 DMA_MOD

3.1 The Mode Registers

The Mode Registers are special 6 bit registers which are dedicated to a channel. Thus there are 4 Mode Registers. The definition of these registers is shown below in the Register Map section. The difference between these channel specific registers and the Memory and Transfer Registers is that they are selected for programming by the two lower data bits. This makes the access to these registers a bit different, fortunately there is no need to update the Mode Registers or change their value during a DMA transfer operation.

4 DMA_IOC

4.1 The IO Bus State Machine

The IO Bus Controller is run by a program counter. The numeric sequence of this program counter varies depending on the type of operation. Thus the sequence for memory to memory transfers will be different from the overlapped memory to IO operation. Likewise, it will be different if the cycle is

compressed, extended, or a verify operation.

4.2 The Temporary Register

The Temporary Data Register is used to hold the value for a memory to memory transfer which is by definition a non overlapped type of transfer. Since there is no "on the fly" transfer of data, the data read needs to be stored for the subsequent data write. There is not one register per channel, thus each operation should end with a write to complete the atomic operation since the data in the Temporary Data Register is not guaranteed to be there if the channel is relinquished.

5 DMA_CAP

5.1 Priority

Both Rotating and Fixed Priority are supported and are selectable.

5.2 Cascade Mode

Cascade mode is unique because when a "Cascaded" channel is granted the IO bus, the DMA_CAP does not request an IO transfer operation, but keeps the IO Bus Control state machine in a dormant state so that another Bus Master can access the IO channel. This could be another DMA controller, or another Bus Master mounted on the IO channel.

5.3 IO Bus Control Handshake

When the DMA_CAP detects that a non-cascaded channel has been granted, a signal gets sent to the IO Bus Controller, unless this IO Bus Controller has been disabled by the bit in the Command Register. This module will wait until the requisite acknowledge signal is returned to indicate completion of the transfer

VCM 1012



operation. If the transfer operation is at the end of transfer, (the Transfer Count = 0000hex) or if Single Mode transfers are requested in the Mode Register for that channel, or in Demand Mode, if there is no longer a request on the DREQ input for that channel, then no subsequent request for an IO Bus transaction is issued to the IO Bus Controller.

5.4 System Handshake

The System Handshake operates such that if there is a DMA transfer request, then a Hold Request is issued. When a Hold Acknowledge is received, a DMA transfer operation is started. The priority of the "System Host" is the lowest priority, so that service to the IO channels is more timely. This prevents IO overruns.

VCM 1012