

Macrocad Development Inc.

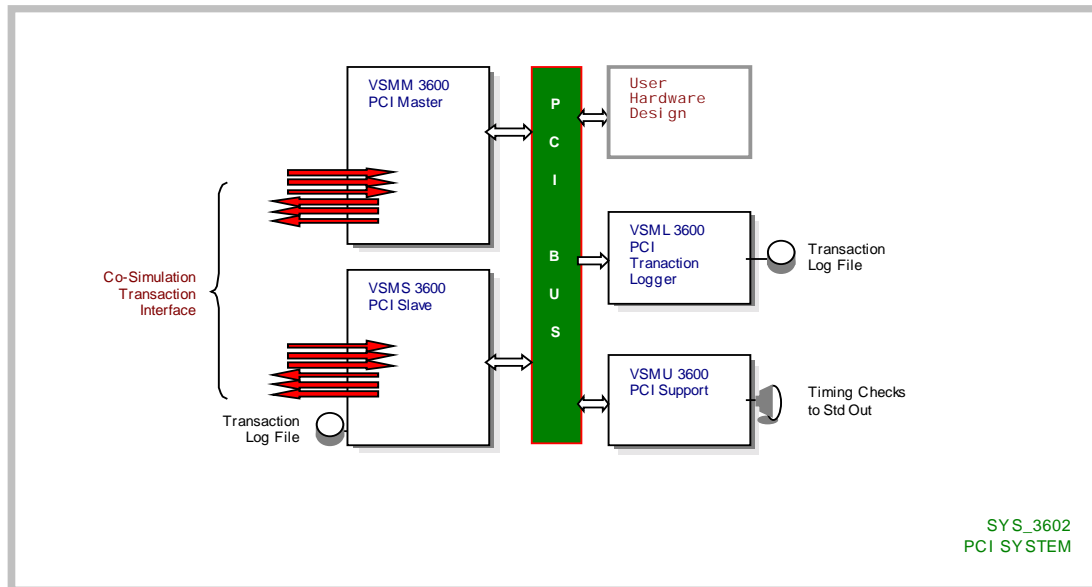
VSM 3602 PCI Bus Functional Models

Description:

Macrocad's VSM 3602 models are behavior models for creating a virtual reality system simulation environment based on the PCI bus specifications. System developers will appreciate the ease of use, and features which allow the designer to manipulate the system at will. Each model is an independent agent - there are no interdependencies to constrain the designer, or to make the system artificial in its operation. Operating parameters are passed to each model to characterize their behavior. Master models execute an instruction list so PCI transaction sequences are easy to control. Simulations run in batch and interactive modes.

Features:

- ◇ Independent agents - realistic system simulation
- ◇ Function and timing checks are automated
- ◇ All PCI transactions are logged
- ◇ Debug features are included
- ◇ Operations are parameterized
- ◇ Interfaces with Co-Simulation tool
- ◇ 64 bit / 66MHz operation supported
- ◇ Designed *by hardware engineers for hardware engineers*
- ◇ Master, Slave, Bus Log, Support, and System Models



VSM 3602



System Architecture

SYS_3602 System Model

This model is the system backbone for PCI based system simulation environments. All PCI devices, structural or behavioral “plug” in to this backbone. A variety of masters and slaves can be assembled on this system backbone, as well as user designs.

VSMU3600 Bus Support Model

This model provides the basic arbitration function for PCI bus devices, generates the PCI clock, checks signal timing during PCI transactions, and generates PCI reset. The user can chose from several arbitration schemes. There is an optional PCI bus activity display, (to Std Out).

VSML3600 Transaction Logger

This model logs all PCI bus transactions to a file. This includes burst, configuration, and special cycles. Bus ownership, parity, etc. are all logged for each transaction. Each transaction is assigned a frame number, which is consistent for all logs files generated by this, or any master or

slave model. This model does not participate in any PCI transaction, but monitors all PCI activity.

VSMS3600 Slave Model

This model acts as a PCI bus slave. It responds to all PCI defined transactions. Operational parameter information can be entered as constants at the system (SYS_3602) level. PCI transactions which this model participates in, can be logged to a file. Configuration information is based on the PCI spec v2.1.

VSMM3600 Master Model

This model acts as a PCI bus master. It will request the bus and, when granted, execute a PCI bus transaction in response to a transaction request which has been deposited in the instruction request cue. As with the slave, operational parameter information can be entered as constants at the system (SYS_3602) level. PCI transactions which this model participates in, can be logged to a file. Configuration information is based on the PCI spec v2.1.

System Test Creation and Execution

In order to construct a test which involves the participation of several models, each model must be properly configured, and 'set up' at the system level. Once the models are configured, the transactions specified must be consistent with the test objective. For example, an access to a memory location outside the memory aperture will result in a master abort PCI transaction. This would not achieve the test objective, if that objective was to transact data with a specific PCI slave

device. It would achieve the test objective, if that objective was to test the PCI device's selection logic, (as a way of verifying the PCI slave did not respond to an address outside its specified aperture). Thus tests can be constructed to verify the ability to handle all normal data transactions to a specific PCI device. Tests can also be constructed to verify that a specific PCI device handles error conditions properly.

VSM 3602



Model Configuration and Operating Parameters

Operational parameter information can be entered as constants at the system (SYS_3602) level. These constants include the file names and paths for log files, etc. Each instantiated model which can participate in a PCI transaction, (VSMM3600 and VSMS3600), will have configuration information loaded from a file at the start of

simulation time. This configuration information can also be loaded if specifically called as part of a test routine, or the registers can be accessed via the normal configuration cycle, thus putting it under program control. Register assignments in the configuration space are based on the PCI spec v2.1.

Model	Name	Range	Type	Definition
VSMM3600	p_dev	integer	parameter	PCI device slot
	p_clk_period	integer	parameter	PCI clock speed, 30ns, or 15ns
	p_log_enable	integer	parameter	Not used, should be set to 0
	p_cfg_file_name	string	parameter	Specifies the PCI device configuration file name and path
	p_pest	integer	parameter	Timing pessimism enabled This drives PCI signals to an 'x' value in between the specified min and max delays with respect to the PCI clock
	p_narcology	integer	parameter	Number of clocks to sleep
VSMS3600	p_mabort	integer	parameter	Number of PCI clocks without response before a master abort is executed
	p_data_size	integer	parameter	data width 32, 64
	p_dev	integer	parameter	PCI device slot
	p_log_enable	integer	parameter	Enables the log file
	p_clk_period	integer	parameter	PCI clock speed, 30ns, or 15ns
	p_log_enable	integer	parameter	slave transaction log enable
	p_data_type	integer	parameter	- 1= 1M register - 2= internal dynamically allocated memory resource - 3= external data
	p_data_size	integer	parameter	data width 32, 64
	p_cfg_file_name	string	parameter	Specifies the PCI device configuration file name and path
	p_log_file_name	string	parameter	Specifies the log file name and path
VSML3600	p_pest	integer	parameter	Timing pessimism enabled This drives PCI signals to an 'x' value in between the specified min and max delays with respect to the PCI clock
	p_io_dword_range	integer	parameter	select delay if byte range
	p_log_enable	integer	parameter	Enables the log file
	p_clk_period	integer	parameter	PCI clock speed, 30ns, or 15ns
VSMU3600	p_log_file_name	string	parameter	Specifies the log file name and path
	p_clk_period	integer	parameter	PCI clock period
	p_pdelay	integer	Parameter	Propagation delay for non-specified signal delays
	p_seed_limit	integer	Parameter	Random number generator seed limit
	p_timestr	integer	Parameter	Time reporting format
	p_pagecount	integer	Parameter	Number of pages before a banner is printed
	p_dsp	integer	Parameter	Display information to Std. Out
	p_arb_type	integer	parameter	Type of arbitration used 0 : Disables arbitration 1 : Simple quasi-random 2 : Highest bit value granted (slot weighted) 3 : Round robin

VSM 3602



SYS_3602	p_test_mask	63: 0	parameter	Switches to enable individual tests
	p_clk_period	integer	parameter	PCI clock speed, 30ns, or 15ns
	p_pest	integer	parameter	Timing pessimism enabled This parameter is sent to the master and slave models at compile time. In this way all models can be controlled by this one parameter at the top level.
all	p_debug_level	15: 0	parameter	Initial value for the r_debug_level register.

These parameters can be set to unique values for each instantiated model. These are to be considered constants, but some are used to set variables which can be changed dynamically during simulation time. One such example is p_debug_level. This is only used to load an initial value, (at the

start of simulation time), into the r_debug_level register. This register can be dynamically changed during simulation, so display options can be switched on and off based on simulation time, sequential test instructions, PCI status, error conditions, or other events.

Test Flow

The test is determined by the list of transaction instructions deposited in the "requested instruction" cue in the PCI master model, (VSMM3600). This model interprets the request (which has a generic format), and translates the instruction into a

PCI transaction. This master model manages all the PCI protocol handshaking, and timing constraints. It executes the PCI transfer, and at the completion, will respond by returning the results and status of the PCI transfer to the Co-Simulation driver.

Test Field Definitions

The instructions and results are comprised of three 64 bit fields, data, address and control.

Request Field Definitions:

Field	Range	Definition
		REQUESTED TRANSACTIONS:
DATA	63: 0	Data Field
	63: 0	This data is octal-byte <64 bit little endian> aligned. Thus the 8 byte enables in the control field determine which bytes are to be transferred.
ADDRESS	63: 0	Address Field
	63: 32	The upper 32 bits of this field provide the upper 32 bit address range for 64 bit PCI devices. This field can also be used to provide a 32 bit data bit mask for the data transaction. This is selected with an option bit in the control field. 64 bit addressing and the data mask option are mutually exclusive operations.
	31: 0	The lower 32 bits of this field provide the entire PCI address range

VSM 3602



		for 32 bit devices, and the lower 32 bit address range for 64 bit PCI devices. This address is assumed to be 64 bit aligned. Thus the lower 3 bits will be ignored.
CONTROL	63:0	Control Field
BEN	63:56	Byte Enable Field
Byte Enables	63:56	These 8 active low byte enables determine which data byte lanes are active during the PCI transaction. Since the requested address field is assumed to be 64 bit aligned, the lower PCI address bits, (2:0) will be generated (if required) from the byte enable information. This might result in multiple PCI accesses for a particular request.
TAG	55:48	Tag Field
Request Tag	55:52	This tag is used to determine the master's instruction cue location for the instruction.
Return Tag	51:48	This tag should be 0 for requests. In the status field returned by the master model, it indicated the instruction which was requested, and is used to match the returned results with the requested instruction.
BURST	47:40	Burst Field
Burst Length	47:40	This indicates the requested burst length, (in bytes). The burst length can be greater than 8 bytes, which will indicate to the master that the PCI transaction should not be completed, but the subsequent instruction cue location contains a subsequent transaction which is a continuation of a PCI burst transaction. Thus specific PCI transaction parameters and options can be specified for each cycle of a PCI burst transaction.
RSVD	39:32	RESERVED
STPAC	31:24	Address Stepping / Data Pacing Field
Address Stepping	31:28	<ul style="list-style-type: none"> - MASTER ONLY : This 4 bit field specifies the number of PCI clocks to step the address prior to entering the data phase. - SLAVE ONLY : This field determines the number of clocks to wait prior to responding with device select.
Data Pacing	27:24	This 4 bit field specifies the number of PCI clocks to inactivate the PCI initiator ready signal, thus temporarily stalling the data transfer cycle.
DEBUG	23:16	Debug switches Field
Debug	23	enable debug feature
Debug	22	display operational errors
Debug	21	display operational info
Debug	20	display details
Debug	19	display time
Debug	18	display task entry - exit
Debug	17	error detect enable
Debug	16	stop on error enable
OPTION	15:8	PCI Transaction Options Field
Options	15	reserved
Option	14	reserved
Option	13	Respond with retry or disconnect
Option	12	Lock the target on this transaction
Option	11	Fast back to back transaction
Option	10	reserved
Option	9	Generate address phase parity error
Option	8	Generate data phase parity error
PCICMD	7:0	PCI Command Field
PCI Command	7:0	PCI command as defined by the PCI spec, v2.1.
reserved	7:4	reserved
	3:0	This field will result in the PCI transaction specified, namely memory reads and writes, IO reads and writes, etc.

VSM 3602



Return Field Definitions:

Field	Range	Definition
		PCI TRANSACTION RESULTS:
DATA	63: 0	Data Field
	63: 0	This data is octal -byte <64 bit little endian> aligned. It reflects the data which was transacted on the PCI bus for both reads and writes.
ADDRESS	63: 0	Address Field
	63: 0	This field reflects the address of the PCI transaction.
CONTROL	63: 0	Control Field
BEN	63: 56	Byte Enable Field
Byte Enables	63: 56	These 8 active low byte enables indicate which data byte lanes were active during the PCI transaction.
TAG	55: 48	Tag Field
Request Tag	55: 52	This tag is used to determine the master's instruction cue location for the instruction. It will be 0 for the returned value.
Return Tag	51: 48	This is the return tag which should be matched with the request tag by the requesting driver.
BURST	47: 40	Burst Field
Burst Length	47: 40	This indicates the number of bursts remaining in the request, (in bytes). The burst field is decremented (with the number of executed bytes) each time a PCI cycle is executed.
RSVD	39: 32	RESERVED
RSVD	31: 24	RESERVED
STATUS	23: 16	Debug switches Field
status	23: 16	This field returns specific information on the PCI transfer executed in reaction to the requested instruction. A normal transaction completion will result in this field being 0.
System Error	23	This indicates that a serious system error has occurred on the PCI bus, (such as a parity error on the address phase of a transaction).
Parity Error	22	This indicates that a parity error has occurred on the PCI bus during the data phase of a PCI transaction. This should normally result in the PCI initiator attempting a retry of the PCI transaction. This is not automatically executed by the VSMM3600 PCI master, but is the responsibility of the driver model.
Reserved	21	TBD
Target Error	20: 19	This indicates that the target has responded with the following status on the PCI transaction: 11 : Target Abort 10 : Target Disconnect 01 : Target Retry 00 : Transaction Complete, No Error
Master Abort	18	This indicates that a master abort has occurred, probably due to no device responding in the address range of the PCI transaction.
Warning Error	17	TBD
Error	16	TBD
Option	15: 8	Option field
Options	15	- SLAVE ONLY : stalls due to external memory resource delay. When this bit is returned active to the slave, the current memory resource read does not have valid data yet. In the case of the write, the write to the memory resource has not completed, and thus subsequent memory resource writes should be delayed until this bit is returned as a '0'. - MASTER ONLY : reserved
Option	14: 8	Reserved
PCICMD	7: 0	PCI Command Field
PCI Command	7: 0	PCI command as defined by the PCI spec, v2.1.
reserved	7: 4	reserved
Cmd	3: 0	This field indicates the PCI transaction which has been completed as a result of the request, namely memory reads and writes, IO reads and writes, etc.

VSM 3602