



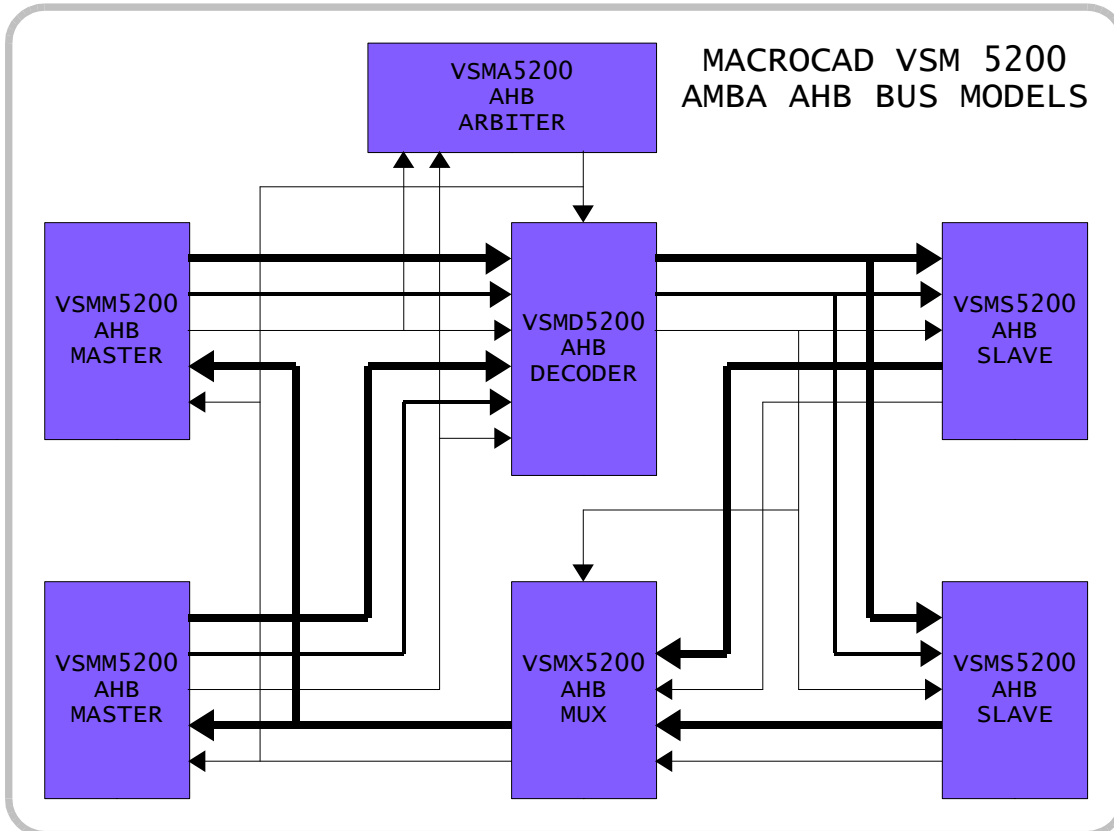
Macrocad Development Inc. VSM 5200 AMBA AHB Bus Functional Models

Description:

Macrocad's VSM 5200 models are synthesizable behavior models for creating a system simulation environment based on the AMBA AHB 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 AMBA AHB transaction sequences are easy to control. Simulations run in batch and interactive modes.

Features:

- ? AMBA AHB 2.0 Compliant
- ? All AMBA AHB transactions are logged
- ? Debug features are included
- ? Independent agents - realistic system simulation
- ? Driver operates from an instruction list
- ? System Backbone and Tests Included
- ? Master, Slave, Decoder, Multiplexer, and Support Models
- ? Models may be synthesized for a hardware accelerator target
- ? Big / Little Endian Support
- ? Supports industry standard simulators



ARM and AMBA are trademarks of ARM Limited.



System Architecture

SYS_5200 System Model

This model is the system backbone for AMBA AHB based system simulation environments. All AHB devices are instantiated from this backbone. A variety of masters and slaves can be instantiated in this model, as well as user designs. The system instantiates VSM_5200 AHB bus agent models, including the VSMM5200 master, VSMS5200 slave, VSMA5200 arbiter, VSMD5200 decoder, VSMX5200 multiplexer, VSMT5200 tracker, VSML5200 logger, and CLK_5200 bus support models. Together, these models provide the user a realistic AHB system simulation environment for design verification.

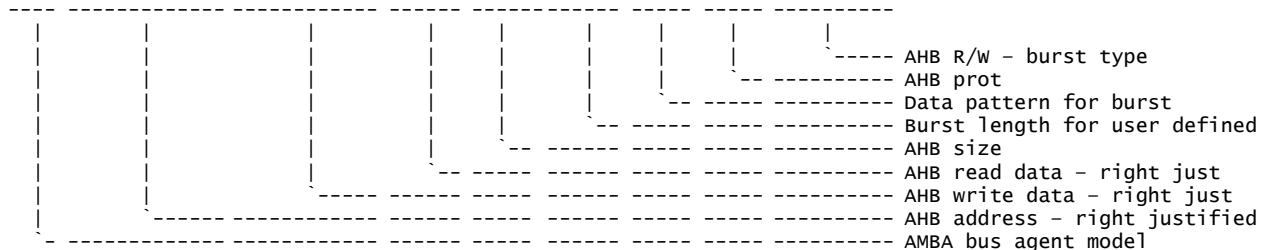
System Test Creation and Execution

In order to construct a test which involves the participation of several models, each model must be properly configured. 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 defined memory range will result in a bus error indication. This would not achieve the test objective, if that objective was to transact data with a specific AMBA AHB slave device. It would achieve the test objective, if that objective was to test the AMBA AHB device's selection logic, (as a way of verifying the AMBA AHB slave did not respond to an address outside its specified range). Thus tests can be constructed to verify the ability to handle all normal data transactions to a specific AMBA AHB device. Tests can also be constructed to verify that a specific AMBA AHB device handles error conditions properly.

Test example

Test can be created by using a series of mac bus transactions. Each transaction contains the information necessary to do one sideband register access to an AHB bus agent model, or an AHB bus frame, if the transaction is to a master model. To simplify this, routines can be created for each device which will generate the proper mac bus request vector, and deal with the tag generation and other handshaking. Use of such a task is shown below. In this case, by passing the task these arguments, an AHB transaction will be executed by the bus master model.

```
dev_0(64'h00000100, 64'h11223344, v_datw, 8'h02, 16'h00, 8'h00, 8'h00, p_rd_ic_4);
```



System Test Configuration for Accelerator Target

Due to requirements for synthesizability with an accelerator target, model configuration must be accomplished through the transaction interface of the master module. Modification of 'r_ctrl_reg', 'r_debug_level', and 'r_mac_mask' is accomplished by using the read/write sideband commands. For more information on sideband commands, see the supported sideband instruction listing. The VSML5200 AHB logger model is not available when a simulation hardware accelerator is used.

System Test Configuration for Simulator Target

Model configuration for a simulator target has parameter options used to control timing and log file configuration. These parameters are in addition to the sideband register configuration available through the transaction interface.



User compile time constants

Several predefined parameters are available for the user when constructing a test. These parameters use bit 4 as the write / read not bit, and use the lower 3 bits to indicate the burst behavior desired. These parameters can be applied to the CMMD field of the control portion of the mac request bus to facilitate an AHB frame.

CONSTANT	VALUE	DESCRIPTION
p_wr_ic16	0x17	Write 16 beats with address increment
p_wr_wp16	0x16	Write 16 beats with address wrap
p_wr_ic8	0x15	Write 8 beats with address increment
p_wr_wp8	0x14	Write 8 beats with address wrap
p_wr_ic4	0x13	Write 4 beats with address increment
p_wr_wp4	0x12	Write 4 beats with address wrap
p_wr_udef	0x11	Write indefinite beats with address increment
p_wr_sngl	0x10	Write Single beat
p_write	0x10	Write Single beat
p_rd_ic16	0x07	Read 16 beats with address increment
p_rd_wp16	0x06	Read 16 beats with address wrap
p_rd_ic8	0x05	Read 8 beats with address increment
p_rd_wp8	0x04	Read 8 beats with address wrap
p_rd_ic4	0x03	Read 4 beats with address increment
p_rd_wp4	0x02	Read 4 beats with address wrap
p_rd_udef	0x01	Read indefinite beats with address increment
p_rd_sngl	0x00	Read Single beat
p_read	0x00	Read Single beat
p_postit	0x08	Post the transaction request, do not wait for results : for pipelining

User compile time constants

The user can specify operating parameters for the simulation. Below are the user parameters which are used to set slave address apertures, log file locations, control register initialization values, test selection and maximum simulation time out trap values..

CONSTANT	RANGE	DESCRIPTION
p_global_control_reg	31:0	Initial value for r_control_reg registers
p_log_dir	string	Log file directory
p_test_mask	15:0	Each bit enables a specific test
p_max_sim_time	Integer	Maximum simulation time allowed, prevents run away simulations
p_lwr_apr_0	31:0	Initial value for lower address aperture for slave 0
p_upr_apr_0	31:0	Initial value for upper address aperture for slave 0
p_lwr_apr_1	31:0	Initial value for lower address aperture for slave 1
p_upr_apr_1	31:0	Initial value for upper address aperture for slave 1
p_lwr_apr_2	31:0	Initial value for lower address aperture for slave 2
p_upr_apr_2	31:0	Initial value for upper address aperture for slave 2
p_lwr_apr_3	31:0	Initial value for lower address aperture for slave 3
p_upr_apr_3	31:0	Initial value for upper address aperture for slave 3



MAC Bus Transaction Interface

AMBA AHB transactions are initiated by an outside source, through the “MAC BUS” transaction interface. The intent is to present a high level of content in the request / return transaction. The bus agent models should strive for the highest level of abstraction for the transaction bus interface. Transaction requests are submitted to the AHB bus devices via the request vector, `mac_req`. Transaction results are returned via the return vector, `mac_ret`. The request and result vectors are comprised of three 64 bit fields; data, address and control. Request and return vector formats are defined below. See the individual model descriptions for the control and status field definitions. The tag field definitions below are the only portion of the request and return definitions which are common to all bus agent models.

NAME	FIELD	RANGE	DESCRIPTION
mac_req		191:0	Request bus
	address	191:128	Address for AHB, register index for sideband
	data	127:64	Data, right aligned
	control	63:0	Control fields 63:56 - model dependent 55:52 - tag request 51:48 - tag acknowledge, <optional> 47:0 - model dependent
mac_ret		191:0	Return response bus
	address	191:128	RESERVED
	data	127:64	Data return, right aligned
	status	63:0	Status fields 63:56 - model dependent 55:52 - tag acknowledge 51:48 - tag return response 47:0 - model dependent

The transaction interface handshaking is a fast and efficient way to create sophisticated system test scenarios. When a AHB or sideband register access is desired, the MAC BUS request vector is asserted to the appropriate AHB bus agent. Below is a sequence for this handshaking.

REQUEST	TAG REQUEST	TAG RESPONSE	DESCRIPTION
Ahb request 1	0x10		Request 1 presented to AHB bus agent model
		0x10	Request 1 accepted by AHB bus agent model
		0x10	AHB master model executes AHB frame for request 1
		0x11	Response 1 presented to requester
Ahb request 2	0x20	0x11	Request 2 presented to AHB bus agent model
		0x20	Request 2 accepted by AHB bus agent model
		0x20	AHB master model executes AHB frame for request 2
		0x22	Response 2 presented to requester

The AHB bus agent will respond when it has completed the request. The way that the handshaking works is through request and response tags. The tag request value of 0x0 is used for resetting the bus agent model. This should be avoided in any circular or modulo tag generation scheme.

Sideband register access

Several of the models are capable of modifying the AHB behavior and functions. This is controlled by sideband registers. These registers are accessed via the mac bus transactions, but do not result in any AHB frames or transfers. The sideband register request and return responses are shown below.



NAME	FIELD	RANGE	DESCRIPTION
mac_req		191:0	bus agent model request
	address	191:128	ADDR
		191:160	RESERVED
		159:128	sideband register index
	data	127:64	DATA
		127:95	RESERVED
		95:64	sideband register data
	control	63:0	CTRL
		63:56	RESERVED
		55:48	TAGS
		47:8	RESERVED
		7:0	CMMD 7:0 = 0x50 - read sideband register 7:0 = 0x60 - write sideband register

NAME	FIELD	RANGE	DESCRIPTION
mac_ret		191:0	bus agent model return response
	address	191:128	RESERVED
	data	95:64	DATA
		127:95	RESERVED
		95:64	sideband register data
	status	63:0	XSTS - transaction status return
		63:56	RESERVED
		55:48	TAGS
		47:0	RESEVED



VSM5200 Master model

The VSMM5200 model acts as an AMBA AHB bus master. It will request the bus and, when granted, execute an AHB bus transaction in response to a transaction request which has been deposited in the instruction request cue. The AHB bus master initiates the AHB transactions.

Master model ports

NAME	I/O	RANGE	DESCRIPTION
haddr	0	31:0	AHB address
hwdata	0	31:0	AHB write data
hrdata	I	31:0	AHB read data
htrans	0	1:0	AHB transfer type
hsize	0	2:0	AHB size
hburst	0	2:0	AHB burst
hprot	0	3:0	AHB protect
hresp	I	1:0	AHB response
hwrite	0		AHB write
hlock	0		AHB lock
hbusreq	0		AHB bus request
hready	I		AHB ready
hgrant	I		AHB bus grant
hresetn	I		AHB reset
hclk	I		AHB clock
mac_req	I	191:0	MAC TRANSACTION BUS REQUEST
mac_ret	0	191:0	MAC TRANSACTION BUS RESPONSE

MASTER REQUEST

This AHB master model is controlled through the mac transaction interface. The specific definitions of the transaction fields are shown below, for both the request and return response vectors.

NAME	FIELD	RANGE	DESCRIPTION
mac_req		191:0	bus agent model request
	address	191:128	ADDR
		191:160	RESERVED
		159:128	address for AHB or sideband index
	data	127:64	DATA
		127:95	RESERVED
		95:64	AHB or sideband data, right aligned
	control	63:0	CTRL
		63:56	SIZE field 63 - no expected data supplied, disable data checks 62:59 - RESERVED 58:56 - AHB hsize
		55:48	TAGS
		47:40	RESEVED



NAME	FIELD	RANGE	DESCRIPTION
		39:32	PATT 7 - loop on access till data compare leng field defined as loop interval and loop limit 6:4 - RESERVED 3:0 - if 0f, mutated CRC 0e, upper half word XORed with lower half word on both half words 0d, upper nibbles XORed with lower nibbles 0c, byte lanes 0 and 2 inverted 0b, end wrap right shift data by 1 bit 0a, end wrap left shift data by 1 bit 09, end wrap right shift data by 1 byte 08, end wrap left shift data by 1 byte 07, decrement by 0x01010101 06, increment by 0x01010101 05, decrement by 0x00000001 04, increment by 0x00000001 03, invert data 02, unmodified data 01, data is all 1's 00, get data from XACT data bus
		31:24	PROT 31:28 - RESERVED 27:24 - AHB prot
		23:8	LENG 23:19 - RESERVED 18:8 - AHB burst length in beats
		7:0	CMMD 7:5 - if not = 000 7:0 = 0x50 - read sideband 7:0 = 0x60 - write sideband 7:5 - if = 000 4 = 1 - write AHB frame = 0 - read AHB frame 3 = Lock request for this AHB frame 2:0 = BRST = 111 - increment 16 beats = 110 - wrap 16 beats = 101 - increment 8 beats = 100 - wrap 8 beats = 011 - increment 4 beats = 010 - wrap 4 beats = 001 - increment indefinite beats = 000 - single beat

MASTER RETURN RESPONSE

NAME	FIELD	RANGE	DESCRIPTION
mac_ret		191:0	bus agent model return response
	address	191:128	RESERVED
	data	127:64	DATA
		127:95	RESERVED
		95:64	AHB or sideband data, right aligned
	status	63:0	XSTS - transaction status return
		63:56	RESERVED
		55:48	TAGS
		47:8	STAT 47 - 1 = global error return 0 = normal transaction return 46:40 - RESERVED 39:10 - RESERVED 9 - data miscompare 8 - error response returned by AHB slave
		7:0	RESERVED

Master Sideband Registers

The master has sideband registers which are used to control the master's AHB frame transactions.



SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_control_reg	31:0	0x00	Control register 31:5 - RESERVED 4 - Allow address increment past 1 K boundary 3:2 - RESERVED 1 - AHB signal timing pessimism enable 0 - model enable
r_debug_switch	31:0	0x01	Display and logging switches 15:8 - RESERVED 7:5 - RESERVED 4 - display detailed info to STD OUT 3 - display AHB info to STD OUT 2 - display transaction to STD OUT 1 - display errors to STD OUT 0 - stop on error
r_data_mask	31:0	0x02	AHB data mask on reads
r_mac_req_cnt	31:0	0x03	Mac bus request counter This counter increments each time the bus agent detects a mac bus request. It is reset by asserting a request with a tag request of 0x0.
r_pace_0	31:0	0x04	Each nibble corresponds to the data availability for the first set of data beats. Thus the first nibble is the data availability delay for the first beat, and so on.
r_pace_1	31:0	0x05	Each nibble corresponds to the data availability for the next set of data beats. Thus the first nibble is the data availability delay for the ninth beat, and so on.
VERSION	31:0	0xfe	Model version This is a read only register which indicates the version of this AHB bus agent model.

Master compile time constants

CONSTANT	RANGE	DESCRIPTION
p_control_reg	31:0	Initial value for r_control_reg register
p_debug_local	31:0	Initial value for r_debug_switch register
p_data_pacing	63:0	Initial value for r_data_pacing, data availability register



VSMS5200 Slave model

This model acts as a AMBA AHB bus slave. It responds to all AHB transactions defined in the release. Address ranges for each slave are configured in the user define file. Currently the slave models a dual port RAM, maximum 1Kb (4x256 bytes).

Slave model ports

NAME	I/O	RANGE	DESCRIPTION
haddr	I	31:0	AHB address
hwdata	I	31:0	AHB write data
hrdata	O	31:0	AHB read data
hsplit	O	15:0	AHB split
htrans	I	1:0	AHB transfer type
hsize	I	2:0	AHB size
hburst	I	2:0	AHB burst
hprot	I	3:0	AHB protect
hresp	O	1:0	AHB response
hmast	I	3:0	AHB master active
hwrite	I		AHB write
hmastlock	I		AHB master lock
hready_out	O		AHB ready, this slave's output
hready	I		AHB ready, any slave, multiplexed
hsel	I		AHB slave select
hresetn	I		AHB reset
hclk	I		AHB clock
xact_s	O		Send XACT data to AHB
xact_r	O		AHB data received
xact_data_s	I	31:0	XXACT data to send to AHB, aligned
xact_data_r	O	31:0	AHB data received and aligned
mac_req	I	191:0	MAC TRANSACTION BUS REQUEST
mac_ret	O	191:0	MAC TRANSACTION BUS RESPONSE

Slave Sideband Registers

The slave has sideband registers which are used to control the slave's AHB frame transactions.

SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_control_reg	31:0	0x00	Control register 31:2 - RESERVED 1 - AHB signal timing pessimism enable 0 - model enable
r_debug_switch	31:0	0x01	Display and logging switches 15:8 - RESERVED 7:5 - RESERVED 4 - display detailed info to STD OUT 3 - display AHB info to STD OUT 2 - display transaction to STD OUT 1 - display errors to STD OUT 0 - stop on error
r_mac_req_cnt	31:0	0x03	Mac bus request counter This counter increments each time the bus agent detects a mac bus request. It is reset by asserting a request with a tag request of 0x0.
r_pace_0	31:0	0x04	Each nibble corresponds to the data availability for the first set of data beats. Thus the first nibble is the data availability delay for the first beat, and so on.



SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_pace_1	31:0	0x05	Each nibble corresponds to the data availability for the next set of data beats. Thus the first nibble is the data availability delay for the ninth beat, and so on.
r_access_limit	31:0	0x10	AHB access limits 31:16 - RESERVED 15:8 - sets the limit for retry response if this limit is exceeded in the r_data_pacing register, then a retry response will be issued 7:0 - sets the limit for split response if this limit is exceeded in the r_data_pacing register, then a split response will be issued
r_send_split	15:0	0x11	Send a split request with the value in this register. This register will clear when selected by the master indicated by the split.
VERSION	31:0	0xfe	Model version This is a read only register which indicates the version of this AHB bus agent model.

Slave compile time constants

CONSTANT	RANGE	DESCRIPTION
p_debug_local	31:0	Initial value for r_debug_switch register
p_control_reg	31:0	Initial value for r_control_reg register
p_data_pacing	63:0	Initial value for r_data_pacing, data availability register



VSMD5200 Decoder model

This model acts as an AMBA AHB decoder. It currently drives select signals for two slaves based on a combinatorial decode of the address signals. Address ranges for the select signals are configured in the user define file. Default slave transactions (transactions outside of the defined slave memory space) are also handled by this module.

Decoder model ports

NAME	I/O	RANGE	DESCRIPTION
hwdata	0	31:0	AHB write data field
haddr	0	31:0	AHB address field
htrans	0	1:0	AHB transfer type field
hsize	0	2:0	AHB size field
hburst	0	2:0	AHB burst field
hprot	0	3:0	AHB protection field
hwrite	0		AHB write
hwdata_0	I	31:0	AHB device 0 write data field
Haddr_0	I	31:0	AHB device 0 address field
htrans_0	I	1:0	AHB device 0 transfer type field
hsize_0	I	2:0	AHB device 0 size field
hburst_0	I	2:0	AHB device 0 burst field
hprot_0	I	3:0	AHB device 0 protection field
hwrite_0	I		AHB device 0 write
hwdata_1	I	31:0	AHB device 1 write data field
Haddr_1	I	31:0	AHB device 1 address field
htrans_1	I	1:0	AHB device 1 transfer type field
hsize_1	I	2:0	AHB device 1 size field
Hburst_1	I	2:0	AHB device 1 burst field
hprot_1	I	3:0	AHB device 1 protection field
hwrite_1	I		AHB device 1 write
hwdata_2	I	31:0	AHB device 2 write data field
Haddr_2	I	31:0	AHB device 2 address field
htrans_2	I	1:0	AHB device 2 transfer type field
hsize_2	I	2:0	AHB device 2 size field
hburst_2	I	2:0	AHB device 2 burst field
hprot_2	I	3:0	AHB device 2 protection field
hwrite_2	I		AHB device 2 write
hwdata_3	I	31:0	AHB device 3 write data field
Haddr_3	I	31:0	AHB device 3 address field
htrans_3	I	1:0	AHB device 3 transfer type field
hsize_3	I	2:0	AHB device 3 size field
hburst_3	I	2:0	AHB device 3 burst field
hprot_3	I	3:0	AHB device 3 protection field
hwrite_3	I		AHB device 3 write
hready	I		AHB ready
hsel	0	15:0	AHB slave select field
Hmaster	I	3:0	AHB master selected
hclk	I		AHB clock
hresetn	I		AHB reset
mac_req	I	191:0	MAC TRANSACTION BUS REQUEST



NAME	I/O	RANGE	DESCRIPTION
mac_ret	0	191:0	MAC TRANSACTION BUS RESPONSE

Decoder sideband registers

SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_control_reg	31:0	0x00	Control register 31:2 - RESERVED 1 - AHB signal timing pessimism enable 0 - model enable
r_debug_switch	31:0	0x01	Display and logging switches 31:0 - RESERVED
r_mac_req_cnt	31:0	0x03	Mac bus request counter This counter increments each time the bus agent detects a mac bus request. It is reset by asserting a request with a tag request of 0x0.
r_lwr_apr_0	31:0	0x10	Value for lower address aperture for slave 0
r_upr_apr_0	31:0	0x11	Value for upper address aperture for slave 0
r_lwr_apr_1	31:0	0x12	Value for lower address aperture for slave 1
r_upr_apr_1	31:0	0x13	Value for upper address aperture for slave 1
r_lwr_apr_2	31:0	0x14	Value for lower address aperture for slave 2
r_upr_apr_2	31:0	0x15	Value for upper address aperture for slave 2
r_lwr_apr_3	31:0	0x16	Value for lower address aperture for slave 3
r_upr_apr_3	31:0	0x17	Value for upper address aperture for slave 3
VERSION	31:0	0xfe	Model version This is a read only register which indicates the version of this AHB bus agent model.

Decoder compile time constants

CONSTANT	RANGE	DESCRIPTION
p_debug_local	31:0	Initial value for r_debug_switch register
p_control_reg	31:0	Initial value for r_control_reg
p_lwr_apr_0	31:0	Initial value for lower address aperture for slave 0
p_upr_apr_0	31:0	Initial value for upper address aperture for slave 0
p_lwr_apr_1	31:0	Initial value for lower address aperture for slave 1
p_upr_apr_1	31:0	Initial value for upper address aperture for slave 1
p_lwr_apr_2	31:0	Initial value for lower address aperture for slave 2
p_upr_apr_2	31:0	Initial value for upper address aperture for slave 2
p_lwr_apr_3	31:0	Initial value for lower address aperture for slave 3
p_upr_apr_3	31:0	Initial value for upper address aperture for slave 3



VSMX5200 Multiplexer model

This model acts as an AMBA AHB Multiplexer. The multiplexer selects slave return data and sends the appropriate return signals to the master module.

Multiplexer model ports

NAME	I/O	RANGE	DESCRIPTION
hrdata	0	31:0	AHB read data
hresp	0	1:0	AHB response
hready	0		AHB ready
hrdata_0	I	31:0	AHB slave 0 read data
hrdata_1	I	31:0	AHB slave 1 read data
hrdata_2	I	31:0	AHB slave 2 read data
hrdata_3	I	31:0	AHB slave 3 read data
hresp_0	I	1:0	AHB slave 0 response
hresp_1	I	1:0	AHB slave 1 response
hresp_2	I	1:0	AHB slave 2 response
hresp_3	I	1:0	AHB slave 3 response
hready_0	I		AHB slave 0 ready
hready_1	I		AHB slave 1 ready
hready_2	I		AHB slave 2 ready
hready_3	I		AHB slave 3 ready
Htrans	I	1:0	AHB transaction type
hsel	I	15:0	AHB slave selects
hclk	0		AHB clock
hresetn	0		AHB reset



VSMA5200 Arbiter model

The AMBA AHB Arbiter model provides the arbitration resolution for the AHB bus.

Arbiter model ports

NAME	I/O	RANGE	DESCRIPTION
hgrant	o	15:0	AHB grant
hmaster	o	3:0	AHB master granted
hmastlock	o		AHB master lock
hbusreq	I	15:0	AHB bus request
hlock	I	15:0	AHB lock
haddr	I	31:0	AHB address
hsplit_0	I	15:0	AHB split device 0
hsplit_1	I	15:0	AHB split device 1
hsplit_2	I	15:0	AHB split device 2
hsplit_3	I	15:0	AHB split device 3
htrans	I	1:0	AHB transfer type
hburst	I	2:0	AHB burst
hresp	I	1:0	AHB slave response
hready	I		AHB slave ready
hclk	o		AHB clock
hresetn	o		AHB reset
mac_req	I	191:0	MAC TRANSACTION BUS REQUEST
mac_ret	o	191:0	MAC TRANSACTION BUS RESPONSE

Arbiter sideband registers

SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_control_reg	31:0	0x00	Control register 31:2 - RESERVED 1 - AHB signal timing pessimism enable 0 - model enable
r_debug_switch	31:0	0x01	Display and logging switches 15:8 - RESERVED 7:5 - RESERVED 4 - display detailed info to STD OUT 3 - display AHB info to STD OUT 2 - display transaction to STD OUT 1 - display errors to STD OUT 0 - stop on error
r_mac_req_cnt	31:0	0x03	Mac bus request counter This counter increments each time the bus agent detects a mac bus request. It is reset by asserting a request with a tag request of 0x0.
VERSION	31:0	0xfe	Model version This is a read only register which indicates the version of this AHB bus agent model.

Arbiter compile time constants

CONSTANT	RANGE	DESCRIPTION
p_debug_local	31:0	Initial value for r_debug_switch register
p_control_reg	31:0	Initial value for r_control_reg register



CLK_5200 Bus support model

This model provides the clocking and initial reset function for AHB devices.

Support model ports

NAME	I/O	RANGE	DESCRIPTION
reset_req	I		AHB reset request
hclk	o		AHB clock
hresetn	o		clock reset



VSM5200 Logger model

The AMBA AHB Transaction Logger provides the user with the ability to print each data transaction for each AHB frame. This transaction logger model is only used in a pure software simulation environment. If a simulation accelerator is used, this model will not be used, because it is not synthesizable.

Logger model ports

NAME	I/O	RANGE	DESCRIPTION
track_o	o	127:0	tracking vector
hclk	o		AHB clock
hresetn	o		AHB reset
mac_req	I	191:0	Mac transaction request
mac_ret	o	191:0	Mac transaction return response

The bus logger monitors the bus tracker's output vector, "track_o". This 128 bit vector provides the frame, address, data, and status information required to construct a log file in a readable format. Examples of log file output segments are shown below.

FORMAT 0:

```
#
# == => : -----
#         : DEVICE OP      FRAME ADDRESS  DATA      SIZE  BEAT  RESP  TIME    UTILIZED
#         : -----
# --i-- : 0 0 Read      168 00000200 54321678  word  1 okay 80370 ns    73 %
# --i-- : 0 0 Read      168 00000204 54321679  word  2 okay 80430 ns    73 %
# --i-- : 0 0 Read      168 00000208 5432167a  word  3 okay 80490 ns    73 %
# --i-- : 0 0 Read      169 00000200 54321678  word  1 okay 80550 ns    73 %
# --i-- : 0 0 Read      170 00000204 54321679  word  1 okay 80610 ns    73 %
# --i-- : 0 0 Read      171 00000208 5432167a  word  1 okay 80670 ns    73 %
# --i-- : 1 0 Write     172 00000240 00000000  word  1 okay 80850 ns    73 %
# --i-- : 1 0 Write     172 00000244 00000000  word  2 okay 80910 ns    73 %
# --i-- : 1 0 Write     172 00000248 00000000  word  3 okay 80970 ns    73 %
# --i-- : 0 0 Write     173 00000200 21654378  word  1 okay 81210 ns    73 %
# --i-- : 0 0 Write     174 00000220 32167845  word  1 okay 81270 ns    73 %
```

The banner will be printed at intervals of 30. The device field indicates the master and slave which are participating in this transaction. The size, time and bus utilization statistics are added as an option.

FORMAT 1:

```
--i-- : Write  F= 83 A= 00000100 D= 11223344 S= 10201083 beat= 1 resp= okay
--i-- : Write  F= 84 A= 00000104 D= 01020304 S= 10201083 beat= 1 resp= okay
--i-- : Write  F= 85 A= 00000108 D= 01020304 S= 10201083 beat= 1 resp= okay
--i-- : Write  F= 86 A= 0000010c D= 01020304 S= 10201083 beat= 1 resp= okay
--i-- : Read   F= 87 A= 00000100 D= 11223344 S= 10231081 beat= 1 resp= okay
--i-- : Read   F= 87 A= 00000104 D= 01020304 S= 102320c1 beat= 2 resp= okay
--i-- : Read   F= 87 A= 00000108 D= 01020304 S= 102330c1 beat= 3 resp= okay
--i-- : Read   F= 87 A= 0000010c D= 01020304 S= 102340c1 beat= 4 resp= okay
```

In this segment, the user can see the write and read accesses, the frame count, address and data. In addition, the status field is followed by the beat count, and the response from the slave.

Logger Sideband Registers



SIDEBAND REGISTER	RANGE	INDEX	DESCRIPTION
r_debug_switch	31:0	0x01	Display and logging switches 31:17 - RESERVED 16 - Log format type, (default 0) 15:8 - RESERVED 7:11 - RESERVED 10 - log transaction to file 9 - log errors to file 8:5 - RESERVED 4 - display detailed info to STD OUT 3 - display AHB info to STD OUT 2 - display transaction to STD OUT 1 - display errors to STD OUT 0 - stop on error
VERSION	31:0	0xfe	Model version This is a read only register which indicates the version of this AHB bus agent model.

Logger compile time constants

CONSTANT	RANGE	DESCRIPTION
p_debug_local	31:0	Initial value for r_debug_switch register
p_log_file_dir	string	Initial value for log file directory path
p_log_file_name	string	Initial value for log file name



VSMT5200 Tracker model

The AMBA AHB Bus tracker provides current information on the state of the AHB bus frames. This information can be used by the bus logger which will format the information, and display it to STD OUT, or to a log file. The status information is presented on a 128 bit bus with the following fields defined:

Tracker model ports

NAME	I/O	RANGE	DESCRIPTION
track_o	0	127:0	tracking vector
haddr	I	31:0	AHB address
hwdata	I	31:0	AHB write data
hrdata	I	31:0	AHB read data
hsel	I	15:0	AHB slave select
hbusreq	I	15:0	AHB bus request
hgrant	I	15:0	AHB bus grant
hlock	I	15:0	AHB lock
hsplit	I	15:0	AHB split
htrans	I	1:0	AHB transfer type
hsize	I	2:0	AHB size
hburst	I	2:0	AHB burst
hprot	I	3:0	AHB protect
hresp	I	1:0	AHB response
hmast	I	3:0	AHB master active
hmastlock	I		AHB master lock
hwrite	I		AHB write
hready	I		AHB ready
hclk	I		AHB clock
hresetn	I		AHB reset

Tracking Vector Description

The AHB tracking vector is available on each AHB clock. However, a transfer on the AHB bus during an AHB frame will be indicated by bit 0 being active, "data beat detected".

NAME	FIELD	RANGE	DESCRIPTION
track_o		127:0	AHB tracking vector
	fram	127:96	127:124 - split 123:120 - lock 119:116 - track_err This register indicates an AHB sequence error 115:112 - RESERVED 111:96 - frame count This register increments each time an AHB frame starts
	addr	95:64	AHB burst address
	data	63:32	AHB burst data
	stat	31:0	31:28 - AHB hmast 27:24 - slave selected 23 - RESERVED 22:20 - AHB hsize 19 - RESERVED 18:16 - AHB hburst 15:12 - beat count 11:8 - AHB hprot 7:6 - AHB htran 5:4 - AHB hresp 3 - RESERVED 2 - pipelined start detected 1 - AHB write 0 - data beat detected

