

	Agend	da
<ul> <li>What is BFM</li> <li>Usage of BFM</li> <li>Task-based BFM</li> <li>File-driven BFM</li> <li>Native code-driven BFM</li> </ul>	rgone	
Copyright © 2013-2017 by Ando Ki	Introduction to BFM	M (2)



















bfm.v	(1/2)
module ahb_bfm ( ); parameter START_ADDR=0; parameter DEPTH_ADDR=32h100; parameter END_ADDR=START_ADDR+DEPTH_ADDR-1; input HRESETn; wire HRESETn; input HRESETn; wire HCLK; output HBUSREQ; reg (BUUSREQ; input HGRANT; wire HGRANT; output[31:0] HADDR; reg [31:0] HADDR; output[31:0] HADDR; reg [31:0] HADDR; output [31:0] HROT; reg [31:0] HPROT; output HLOCK; reg HLOCK; output HLOCK; reg HLOCK; output [2:0] HSIZE; reg [2:0] HSIZE; output [2:0] HBURST; reg [21:0] HWRATA; input [31:0] HRDATA; reg [31:0] HWDATA; input [31:0] HRDATA; wire [31:0] HWDATA; input [31:0] HRDATA; wire [31:0] HRESP; input HREADY; wire HREADY; input IRQn; wire IRQn; input IRQn; wire FIQn;	<pre>initial begin HBUSREQ = 0; HADDR = 0; HPROT = 0; HLOCK = 0; HTRANS = 0; HWRITE = 0; HSIZE = 0; HBURST = 0; HWDATA = 0; while (HRESETn===1'bx) @ (posedge HCLK); while (HRESETn===1'b1) @ (posedge HCLK); repeat (3) @ (posedge HCLK); memory_test(START_ADDR, END_ADDR, 1); memory_test(START_ADDR, END_ADDR, 1); memory_test(START_ADDR, END_ADDR, 2); memory_test(START_ADDR, END_ADDR, 4); repeat (5) @ (posedge HCLK); \$finish(2); end</pre>
Copyright © 2013-2017 by Ando Ki Introduction to	BFM (12)



ahb_tasks.v: A	HB read tasks	
<pre>task ahb_read; input [31:0] address; input [2:0] size; output [31:0] data; begin @ (posedge HCLK); HBUSREQ &lt;= #1 1'b1; @ (posedge HCLK); while ((HGRANT!==1'b1)) ((HREADY!==1'b1)) @ (posedge HCLK); HADDR &lt;= #1 address; HPROT &lt;= #1 4'b0001; // HPROT_DATA HTRANS &lt;= #1 2'b10; // HTRANS_NONSEQ; HBURST &lt;= #1 3'b000; // HBURST_SINGLE; HWRITE &lt;= #1 3'b000; // HBURST_SINGLE; HWRITE &lt;= #1 3'b000; // HSIZE_BYTE; 2: HSIZE &lt;= #1 3'b001; // HSIZE_HWORD; 4: HSIZE &lt;= #1 3'b010; // HSIZE_WORD; default: \$display(\$time,, "ERROR: unsupported transfer size; %d-byte", size); endcase</pre>	<pre>@ (posedge HCLK); while (HREADY!==1'b1) @ (posedge HCLK); HADDR &lt;= #1 32'b0; HPROT &lt;= #1 4'b0000; // HPROT_OPCODE HTRANS &lt;= #1 2'b0; HBURST &lt;= #1 3'b0; @ (posedge HCLK); while (HREADY===0) @ (posedge HCLK); data = HRDATA; // must be blocking if (HRESP!=2'b00) //if (HRESP!='HRESP_OKAY) \$display(\$time,, "ERROR: non OK response for read"); @ (posedge HCLK); end endtask</pre>	
Copyright © 2013-2017 by Ando Ki Introduction to BFM (14)		









Reference
В 기안도, 시스템 집적 반도체 설계검증 환경과 기법, 제4장 BFM을 이용한 설계 자산의 기능검증, 홍릉과학출판사, 2008.

Copyright © 2013-2017 by Ando Ki

Introduction to BFM (19)