2017112002_손보성_SOC_HW#6

2. HW#5 redo

0) AHB-compatible SRAM controller DUT

```
module sram_controller(hclk, reset, htrans, hwrite, haddr, hwdata, hrdata, hready, addr, we, sram_dout, sram_din);

    input hclk, hwrite, reset;
    input [1 : 0] htrans;
    input [3 : 0] haddr;
    input [7 : 0] hwdata;
    input [7 : 0] sram_dout;

    reg we_buf;
    reg [1 : 0] sc;
    reg [3 : 0] addr_buf;

    reg hready, we;
    reg [3 : 0] addr;
    reg [7 : 0] hrdata;
    reg [7 : 0] sram_din;

    output hready, we;
    output [3 : 0] addr;
    output [7 : 0] hrdata;
    output [7 : 0] sram_din;

    always @(posedge hclk or negedge reset) begin
        if (!reset) begin
            sc <= 2'b00;
            hready <= 1'b1;
            we <= 1'b1;
            we_buf <= 1'b0;
            addr <= 4'h0;
            addr_buf <= 4'h0;
            hrdata <= 8'h00;
            sram_din <= 8'h00;
            end

        if (sc == 2'b00) begin
            if (htrans == 2'b00) begin
                we <= 1'b1;
                end
            else if (htrans == 2'b11) begin
                addr_buf <= haddr;
                we_buf <= hwrite;
                if (hwrite == 1'b0) begin
                    hready <= 1'b0;
                #1  addr <= haddr;
                    end
                sc <= 2'b01;
                end
            end
```

2017112002_손보성_SOC_HW#6

```
49
50          else if (sc == 2'b01) begin
51              sc <= 2'b11;
52              if (we_buf == 1'b1) begin
53              #1  addr <= addr_buf;
54                  sram_din <= hwdata;
55                  we <= 1'b0;
56                  end
57              else if (we_buf == 1'b0) begin
58                  hready <= 1'b0;
59                  end
60              end
61
62          else if (sc == 2'b11) begin
63              if (hready == 1'b1) begin
64                  sc <= 2'b00;
65              #1  we <= 1'b1;
66                  end
67              else if (hready == 1'b0) begin
68                  hready <= 1'b1;
69                  hrdata <= sram_dout;
70                  sc <= 2'b10;
71                  end
72              end
73
74          else if (sc == 2'b10) begin
75              sc <= 2'b00;
76              end
77          end
78
79
80  endmodule
```

2017112002_손보성_SOC_HW#6

1) Testbench

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////...
module tb();

    integer fp;
    reg [7 : 0] in [0 : 1];
    reg hclk, hwrite, reset;
    reg [1 : 0] htrans;
    reg [3 : 0] haddr;
    reg [7 : 0] hwdata;

    wire hready, we;
    wire [3 : 0] addr;
    wire [7 : 0] hrdata, sram_din, sram_dout;

    sram_controller sram_controller0(hclk, reset, htrans, hwrite, haddr, hwdata, hrdata, hready, addr, we, sram_dout, sram_din);
    sram_16x8 sram0(addr, hclk, sram_din, sram_dout, we);

    always #5 hclk = ~hclk;
    initial begin
        $readmemb("init_mem.txt", tb.sram0.mem);
        $readmemb("in.txt", tb.in);
        hclk <= 1'b1; reset <= 1'b1; hwrite <= 1'b0; htrans <= 2'b00;
    #1  reset <= 1'b0;
    #1  reset <= 1'b1; haddr <= 4'h0; hwdata <= 8'h00;
    #9 hwrite <= 1'b1; haddr <= 4'h0; htrans <= 2'b11;
    #10 hwrite <= 1'b0; hwdata <= in[0]; htrans <= 2'b00;
    #10
    #10 haddr <= 4'h0; htrans <= 2'b11;
    #10 htrans <= 2'b00;
    #30 hwrite <= 1'b1; haddr <= 4'h1; htrans <= 2'b11;
    #10 hwrite <= 1'b0; hwdata <= in[1]; htrans <= 2'b00;
    #10
    #10 haddr <= 4'h1; htrans <= 2'b11;
    #10 htrans <= 2'b00;
    #30 fp = $fopen("out.txt");
        $fdisplay(fp, "mem[00x] = %b", tb.sram0.mem[4'h0]);
        $fdisplay(fp, "mem[01x] = %b", tb.sram0.mem[4'h1]);
        $fclose(fp);
    #10 $finish;
    end
endmodule
```