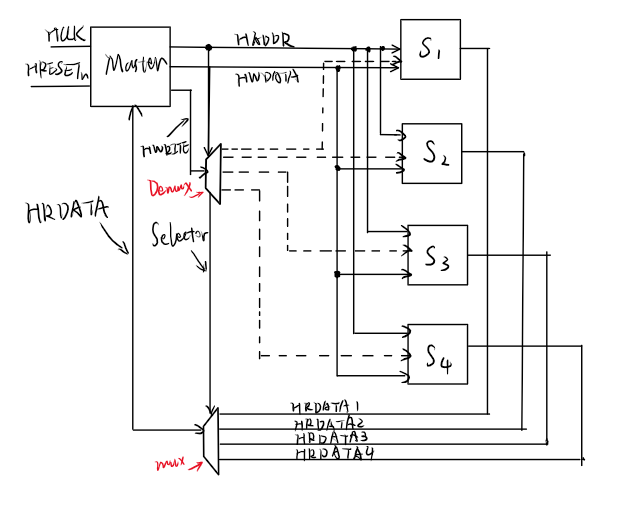
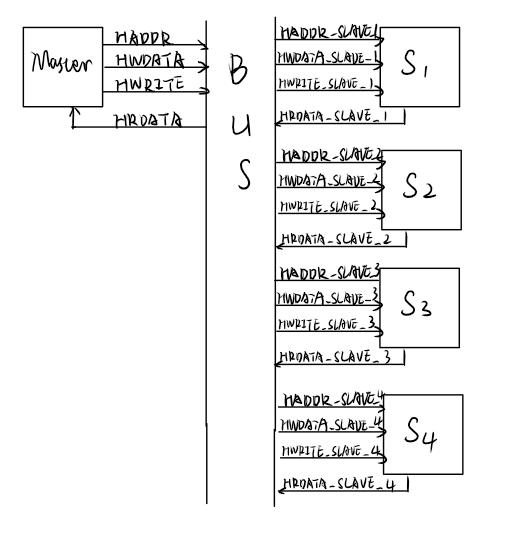
Functional Electronic Circuits Lab4

Student Name: CAO Xinyang

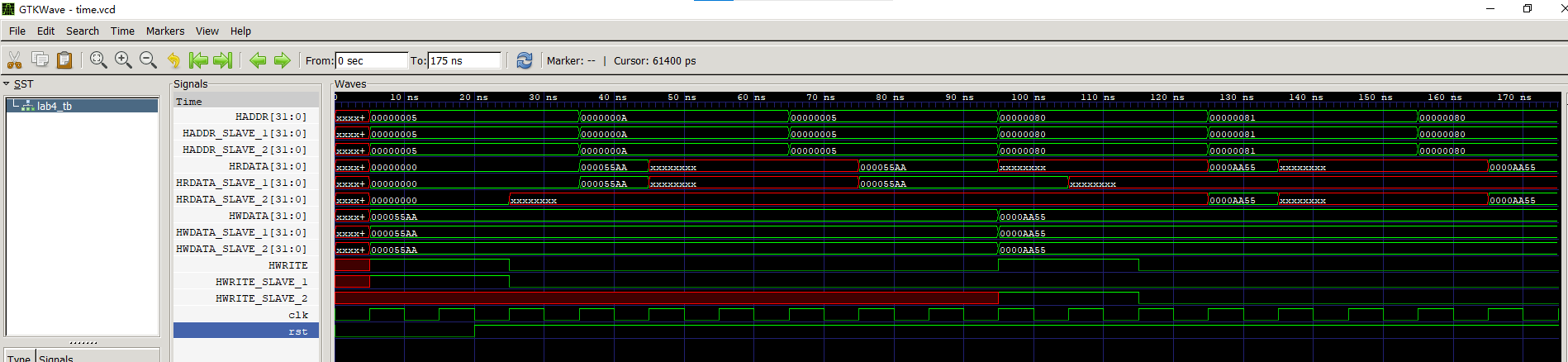
Student ID: 20321308

**1. The picture of the system**

**

**

**2. The timing diagram with simulation results**

****

**3. Code of the testbench and the device**

***busDevice.v***

module bus(

    input [31:0] HADDR\_MASTER\_bi,

    input [31:0] HWDATA\_MASTER\_bi,

    input HWRITE\_MASTER\_i,

    output reg [31:0] HRDATA\_MASTER\_bo,

    output [31:0] HADDR\_SLAVE\_1\_bo,

    output [31:0] HWDATA\_SLAVE\_1\_bo,

    output reg HWRITE\_SLAVE\_1\_o,

    input [31:0] HRDATA\_SLAVE\_1\_bi,

    output [31:0] HADDR\_SLAVE\_2\_bo,

    output [31:0] HWDATA\_SLAVE\_2\_bo,

    output reg HWRITE\_SLAVE\_2\_o,

    input [31:0] HRDATA\_SLAVE\_2\_bi

);

assign HADDR\_SLAVE\_1\_bo = HADDR\_MASTER\_bi;

assign HADDR\_SLAVE\_2\_bo = HADDR\_MASTER\_bi;

assign HWDATA\_SLAVE\_1\_bo = HWDATA\_MASTER\_bi;

assign HWDATA\_SLAVE\_2\_bo = HWDATA\_MASTER\_bi;

always@\*

    if (HADDR\_MASTER\_bi < 32'h80) begin

        HWRITE\_SLAVE\_1\_o <= HWRITE\_MASTER\_i;

        HRDATA\_MASTER\_bo <= HRDATA\_SLAVE\_1\_bi;

    end else begin

        HWRITE\_SLAVE\_2\_o <= HWRITE\_MASTER\_i;

        HRDATA\_MASTER\_bo <= HRDATA\_SLAVE\_2\_bi;

    end

endmodule

***masterDevice.v***

module master(

    input HCLK\_i,

    input HRESETn\_i,

    input [31:0] HRDATA\_bi,

    output reg [31:0] HADDR\_bo,

    output reg [31:0] HWDATA\_bo,

    output reg HWRITE\_o

);

task write(

    input [31:0] addr,

    input [31:0] data

);

    begin

        @(posedge HCLK\_i);

        HADDR\_bo <= addr;

        HWRITE\_o <= 1;

        HWDATA\_bo <= data;

        @(posedge HCLK\_i);

        @(posedge HCLK\_i);

        HWRITE\_o <= 0;

    end

endtask

task read(

    input [31:0] addr

);

    begin

        @(posedge HCLK\_i);

        HADDR\_bo <= addr;

        HWRITE\_o <= 0;

        @(posedge HCLK\_i);

        @(posedge HCLK\_i);

        $display("READ DATA | addr: %h, data: %h", addr, HRDATA\_bi);

    end

endtask

initial begin

    write(32'h5, 32'h55AA);

    read(32'hA);

    read(32'h5);

    write(32'h80, 32'hAA55);

    read(32'h81);

    read(32'h80);

    $finish;

end

endmodule

***slaveDevice.v***

module slave(

    input HCLK\_i,

    input HRESETn\_i,

    output reg [31:0] HRDATA\_bo,

    input [31:0] HADDR\_bi,

    input [31:0] HWDATA\_bi,

    input HWRITE\_i

);

reg [31:0] mem [255:0];

always@(posedge HCLK\_i)

    if (HRESETn\_i == 0)

        HRDATA\_bo <= 0;

    else begin

        if (HWRITE\_i)

            mem[HADDR\_bi] <= HWDATA\_bi;

        else

            HRDATA\_bo <= mem[HADDR\_bi];

    end

endmodule

***testBench.v***

`timescale 1ns/1ps

module lab4\_tb;

reg clk, rst;

wire [31:0] HRDATA, HADDR, HWDATA;

wire HWRITE;

wire [31:0] HRDATA\_SLAVE\_1, HADDR\_SLAVE\_1, HWDATA\_SLAVE\_1;

wire HWRITE\_SLAVE\_1;

wire [31:0] HRDATA\_SLAVE\_2, HADDR\_SLAVE\_2, HWDATA\_SLAVE\_2;

wire HWRITE\_SLAVE\_2;

bus buut (

    .HADDR\_MASTER\_bi(HADDR),

    .HWDATA\_MASTER\_bi(HWDATA),

    .HRDATA\_MASTER\_bo(HRDATA),

    .HWRITE\_MASTER\_i(HWRITE),

    .HADDR\_SLAVE\_1\_bo(HADDR\_SLAVE\_1),

    .HWDATA\_SLAVE\_1\_bo(HWDATA\_SLAVE\_1),

    .HRDATA\_SLAVE\_1\_bi(HRDATA\_SLAVE\_1),

    .HWRITE\_SLAVE\_1\_o(HWRITE\_SLAVE\_1),

    .HADDR\_SLAVE\_2\_bo(HADDR\_SLAVE\_2),

    .HWDATA\_SLAVE\_2\_bo(HWDATA\_SLAVE\_2),

    .HRDATA\_SLAVE\_2\_bi(HRDATA\_SLAVE\_2),

    .HWRITE\_SLAVE\_2\_o(HWRITE\_SLAVE\_2)

);

master muut(

    .HCLK\_i(clk),

    .HRESETn\_i(rst),

    .HRDATA\_bi(HRDATA),

    .HADDR\_bo(HADDR),

    .HWDATA\_bo(HWDATA),

    .HWRITE\_o(HWRITE)

);

slave suut1(

    .HCLK\_i(clk),

    .HRESETn\_i(rst),

    .HRDATA\_bo(HRDATA\_SLAVE\_1),

    .HADDR\_bi(HADDR\_SLAVE\_1),

    .HWDATA\_bi(HWDATA\_SLAVE\_1),

    .HWRITE\_i(HWRITE\_SLAVE\_1)

);

slave suut2(

    .HCLK\_i(clk),

    .HRESETn\_i(rst),

    .HRDATA\_bo(HRDATA\_SLAVE\_2),

    .HADDR\_bi(HADDR\_SLAVE\_2),

    .HWDATA\_bi(HWDATA\_SLAVE\_2),

    .HWRITE\_i(HWRITE\_SLAVE\_2)

);

always #5 clk = ~clk;

initial begin

    $dumpfile("time.vcd");

    $dumpvars(1, lab4\_tb);

    clk = 0;

    rst = 0;

    #20

    rst = 1;

end

endmodule