Report

一、设计思路

1. 时钟

对于给定的外部时钟，instruction memory和data memory的时钟即为该时钟，CPU的时钟为二分频后的时钟。

具体实现：

sc_computer.v

```
reg clock;
initial
begin
    clock = 0;
end
always @(posedge clk)
begin
    if(~reset)
        clock <= 0;
    clock <= ~clock;
end
```

2. IO端口

根据给定的地址，选定对应的外部设备读写。

具体实现：

sc_io_input.v

```
module io_input(addr,io_clk,io_read_data,in_port0,in_port1);
    input [31:0] addr;
    input io_clk;
    input [31:0] in_port0,in_port1;
    output [31:0] io_read_data;
    reg [31:0] in_reg0;
    reg [31:0] in_reg1;
    io_input_mux io_input_mux2x32(in_reg0,in_reg1,addr[7:2],io_read_data);

    always @ (posedge io_clk)
    begin
        in_reg0 <= in_port0;
        in_reg1 <= in_port1;
    end


endmodule

module io_input_mux(a0,a1,sel_addr,y);
    input [31:0] a0,a1;
    input [5:0] sel_addr;
    output [31:0] y;
    reg [31:0] y;
    always @ *
        case (sel_addr)
            6'b100000: y = a0;
            6'b100001: y = a1;
            // more ports 可根据需要设计更多的端口
            default: y = 32'h0;
        endcase

endmodule
```

sc_io_output.v

```
module io_output(addr,datain,write_io_enable,io_clk,out_port0,out_port1,out_port2);
    input [31:0] addr,datain;
    input write_io_enable,io_clk;
    output [31:0] out_port0,out_port1,out_port2;
    reg [31:0] out_port0;
    reg [31:0] out_port1;
    reg [31:0] out_port2;
    always @ (posedge io_clk)
    begin
        if (write_io_enable == 1)
            case (addr[7:2])
                6'b100000: out_port0 <= datain; // 80h
                6'b100001: out_port1 <= datain; // 84h
                6'b100010: out_port2 <= datain; // 88h
            endcase
    end
endmodule
```

3. Data Memory

在data memory中根据给定地址确定输入/输出数据来自内存还是外部设备。

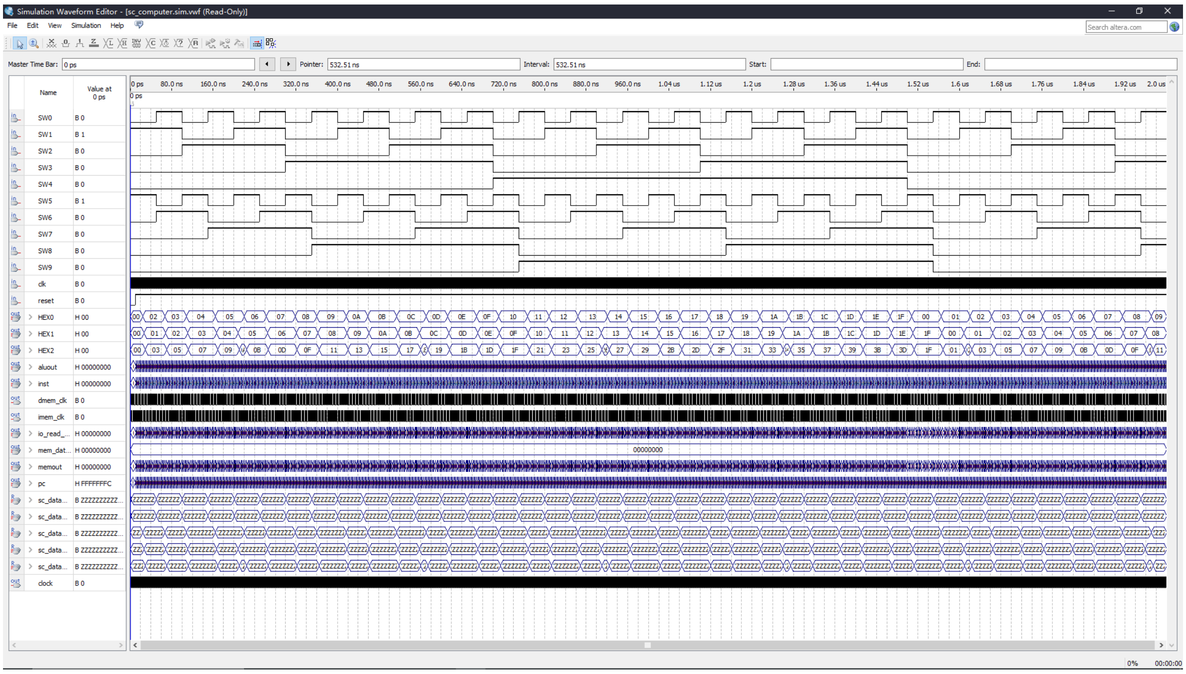
具体实现：

sc_datamem.v

```
module sc_datamem (addr,datain,dataout,we,clock,mem_clk,dmem_clk,out_port0,out_port1,out_port2,in_port0,in_port1,mem_dataout,io_read_data);

   input  [31:0]  addr;
   input  [31:0]  datain;

   input          we, clock,mem_clk;
    input [31:0] in_port0,in_port1;
   output [31:0]  dataout;
   output         dmem_clk;
    output [31:0] out_port0,out_port1,out_port2;
    output [31:0] mem_dataout;
    output [31:0] io_read_data;

    wire            dmem_clk;
    wire            write_enable;
     wire [31:0] dataout;
     wire [31:0] mem_dataout;
     wire write_data_enable;
     wire write_io_enable;
    assign          write_enable = we & ~clock;

    assign          dmem_clk = mem_clk & ( ~ clock) ;
     assign write_data_enable = write_enable & (~addr[7]);//注意
     assign write_io_enable = write_enable & addr[7]; //注意

    lpm_ram_dq_dram  dram(addr[6:2],dmem_clk,datain,write_data_enable,mem_dataout );

     mux2x32 io_data_mux(mem_dataout,io_read_data,addr[7],dataout);

     io_output io_output_reg (addr, datain, write_io_enable, dmem_clk, out_port0,out_port1,out_port2);
     io_input io_input_reg(addr, dmem_clk, io_read_data, in_port0, in_port1);



endmodule
```

二、仿真结果

总览:

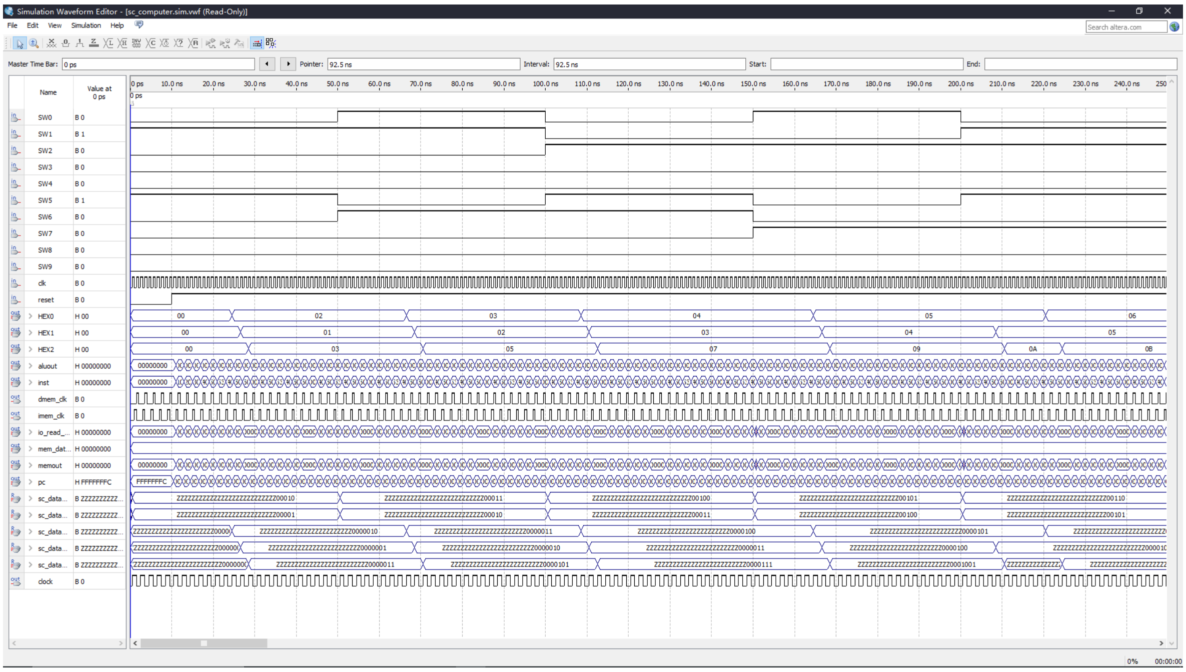


Simulation Waveform Editor - [sc_computer.sim.vwf (Read-Only)]

Master Time Bar: 0 ps   Pointer: 532.51 ns   Interval: 532.51 ns   Start: End:

Name	Value at 0 ps
SW0	B 0
SW1	B 1
SW2	B 0
SW3	B 0
SW4	B 0
SW5	B 1
SW6	B 0
SW7	B 0
SW8	B 0
SW9	B 0
clk	B 0
reset	B 0
HEX0	H 00
HEX1	H 00
HEX2	H 00
aluout	H 00000000
inst	H 00000000
dmem_clk	B 0
imem_clk	B 0
io_read_...	H 00000000
mem_dat...	H 00000000
memout	H 00000000
pc	H FFFFFFFC
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
clock	B 0

具体细节:



Simulation Waveform Editor - [sc_computer.sim.vwf (Read-Only)]

Master Time Bar: 0 ps

Pointer: 92.5 ns

Interval: 92.5 ns

Start:

End:

Name	Value at 0 ps
SW0	B 0
SW1	B 1
SW2	B 0
SW3	B 0
SW4	B 0
SW5	B 1
SW6	B 0
SW7	B 0
SW8	B 0
SW9	B 0
clk	B 0
reset	B 0
HEX0	H 00
HEX1	H 00
HEX2	H 00
aluout	H 00000000
inst	H 00000000
dmem_clk	B 0
imem_clk	B 0
io_read_...	H 00000000
mem_dat...	H 00000000
memout	H 00000000
pc	H FFFFFFFC
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
sc_data...	B ZZZZZZZZZZ...
clock	B 0