数字电路实验

作业 1：PC 及 AL 的仿真

选课编号：K0166210.02

姓名（周子涵）选课序号（53）

电子科技大学

通信与信息工程学院

2019 年 11 月

一、verilog 代码（请在代码后附加注解）

PC：
```
//-------------------------------------------------------------------
//--SAYEH (Simple Architecture Yet Enough Hardware) CPU
//-------------------------------------------------------------------
//Program Counter

`timescale 1 ns /1 ns

module ProgramCounter (in, enable, clk, out);
input [15:0] in;
input enable, clk;
output [15:0] out;
reg [15:0] out;

   always @(negedge clk)
            //code
      begin
         if(enable==1)//只有使能端有效时，输入才有效
             out=in;

      end

endmodule
```

AL：
```
//-------------------------------------------------------------------
//--SAYEH (Simple Architecture Yet Enough Hardware) CPU
//-------------------------------------------------------------------
// Address Logic

`timescale 1 ns /1 ns
`define ResetPCH 5'b10000//进行宏定义
`define PCplusIH 5'b01000
`define PCplus1H 5'b00100
`define RplusIH 5'b00010
`define Rplus0H 5'b00001
module AddressLogic (
      PCside, Rside, Iside, ALout, ResetPC, PCplusI, PCplus1, RplusI, Rplus0
);
input [15:0] PCside, Rside;
input [7:0] Iside;
input ResetPC, PCplusI, PCplus1, RplusI, Rplus0;
output [15:0] ALout;
reg [15:0] ALout;
```

```
    always @(PCside or Rside or Iside or ResetPC or PCplusI or PCplus1 or RplusI or Rplus0)
    begin
        case ({ResetPC, PCplusI, PCplus1, RplusI, Rplus0})
            //code
            `ResetPCH:ALout=16'b0;//复位且使输出为 0
            `PCplusIH:ALout=PCside+Iside;
            `PCplus1H:ALout=PCside+1;
            `RplusIH:ALout={{8'b0},Iside};//将 Iside 扩展至 16 位
            `Rplus0H:ALout=Rside;
            default: ALout = PCside;//AL 无效时，PCside 直接传到输出
        endcase
    end

endmodule

AdressingUnit：//----------------------------------------------------------------------
//--SAYEH (Simple Architecture Yet Enough Hardware) CPU
//----------------------------------------------------------------------
// Addressing Unit

`timescale 1 ns /1 ns

module AddressingUnit (
input [15:0] Rside,
input [7:0] Iside,
output [15:0] Address,
input clk,ResetPC, PCplusI, PCplus1, RplusI, Rplus0, PCenable
);
wire [15:0] PCout;

    ProgramCounter PC (Address, PCenable, clk, PCout);
    AddressLogic AL (PCout, Rside, Iside, Address, ResetPC, PCplusI, PCplus1, RplusI,
Rplus0);//进行调用

endmodule
```

二、 testbench 代码

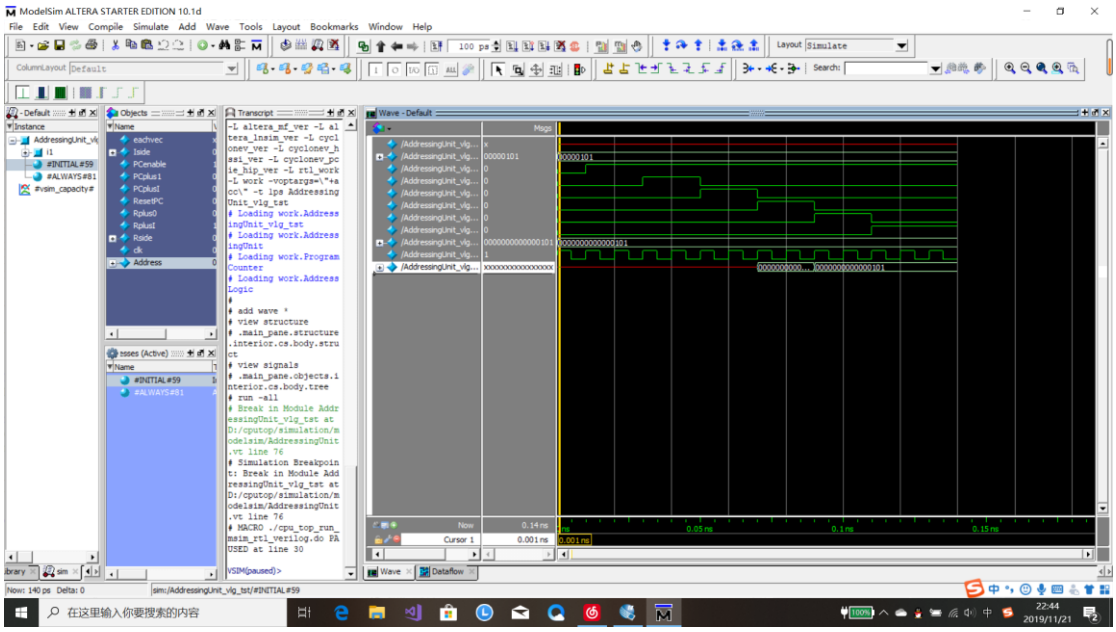
```
// Copyright (C) 1991-2013 Altera Corporation
// Your use of Altera Corporation's design tools, logic functions
```

```
// and other software and tools, and its AMPP partner logic
// functions, and any output files from any of the foregoing
// (including device programming or simulation files), and any
// associated documentation or information are expressly subject
// to the terms and conditions of the Altera Program License
// Subscription Agreement, Altera MegaCore Function License
// Agreement, or other applicable license agreement, including,
// without limitation, that your use is for the sole purpose of
// programming logic devices manufactured by Altera and sold by
// Altera or its authorized distributors.  Please refer to the
// applicable agreement for further details.

// ********************************************************************************
// This file contains a Verilog test bench template that is freely editable to
// suit user's needs .Comments are provided in each section to help the user
// fill out necessary details.
// ********************************************************************************
// Generated on "11/21/2019 21:54:08"

// Verilog Test Bench template for design : AddressingUnit
//
// Simulation tool : ModelSim-Altera (Verilog)
//

`timescale 1 ps/ 1 ps
module AddressingUnit_vlg_tst();
// constants
// general purpose registers
reg eachvec;
// test vector input registers
reg [7:0] Iside;
reg PCenable;
reg PCplus1;
reg PCplusI;
reg ResetPC;
reg Rplus0;
reg RplusI;
reg [15:0] Rside;
reg clk;
// wires
wire [15:0]  Address;
```

```
// assign statements (if any)
AddressingUnit i1 (
// port map - connection between master ports and signals/registers
	.Address(Address),
	.Iside(Iside),
	.PCenable(PCenable),
	.PCplus1(PCplus1),
	.PCplusI(PCplusI),
	.ResetPC(ResetPC),
	.Rplus0(Rplus0),
	.RplusI(RplusI),
	.Rside(Rside),
	.clk(clk)
);
initial
begin
// code that executes only once
// insert code here --> begin
clk=1;
PCenable=0;
Rside=16'b0000_0000_0000_0101;
Iside=8'b0000_0101;
PCplus1=0;PCplusI=0;ResetPC=0;Rplus0=0;RplusI=0;

#10 PCenable=1;
#20 PCplus1=1;PCplusI=0;ResetPC=0;Rplus0=0;RplusI=0;
#20 PCplus1=0;PCplusI=1;ResetPC=0;Rplus0=0;RplusI=0;
#20 PCplus1=0;PCplusI=0;ResetPC=1;Rplus0=0;RplusI=0;
#20 PCplus1=0;PCplusI=0;ResetPC=0;Rplus0=1;RplusI=0;
#20 PCplus1=0;PCplusI=0;ResetPC=0;Rplus0=0;RplusI=1;

#30 $stop;

// --> end

end
always
// optional sensitivity list
// @(event1 or event2 or .... eventn)
begin
```

```
// code executes for every event on sensitivity list
// insert code here --> begin

#5 clk=~clk;
// --> end
end
endmodule
```

三、仿真波形（请对仿真波形进行简要分析）



在时钟进入下降沿时，就会进入 always，从而检查变量是否发生改变，PC 使能端有效，AL 无效时，输出一直等于输入；当 PC 和 AL 同时有效时，就可以产生相应功能的输出。