|  |  |
| --- | --- |
| logo[1] | 杭州电子科技大学 |

实验报告

Experimental report

|  |  |
| --- | --- |
| **实验课程名称** | **数字系统与处理器** |
| **实验序号** | 4 |
| **实验内容** | 程序计数器设计 |
| **班级** | 18184111 |
| **姓名** | 杨臻 |
| **学号** | 18052032 |
| **指导教师** | 黄继业 |

二○一九 年 十二月 七日

## 实验目的与要求

实验任务1：设计一个PC寄存器，位宽为32位，在不同的操作类型（PC\_src\_sel信号）控制下，执行下表操作（除了PC\_src\_sel外，均为32位）

|  |  |  |
| --- | --- | --- |
| 操作类型PC\_src\_sel | Base | Offset |
| PC\_JAL\_TARGET | PC\_DX | jal\_offset |
| PC\_JALR\_TARGET | rs1\_data | jalr\_offset |
| PC\_BRANCH\_TARGET | PC\_DX | imm\_b |
| PC\_REPLAY | PC\_IF | 0 |
| PC\_HANDLER | handler\_PC | 0 |
| PC\_EP | epc | 0 |
| 其他 | PC\_IF | 4 |

其中PC\_PIF = base + offset

每个时钟上升沿寄存一次PC\_PIF的值，请写出PC的Verilog代码，要求带异步复位

实验任务2：写上PC的TechBench，在ModelSim中仿真

## 实验原理

寄存器的功能是存储二进制代码，它是由具有存储功能的触发器组合起来构成的。一个触发器可以存储1位二进制代码，故存放n位二进制代码的寄存器，需用n个触发器来构成。

按照功能的不同，可将寄存器分为基本寄存器和移位寄存器两大类。基本寄存器只能并行送入数据，也只能并行输出。移位寄存器中的数据可以在移位脉冲作用下依次逐位右移或左移，数据既可以并行输入、并行输出，也可以串行输入、串行输出，还可以并行输入、串行输出，或串行输入、并行输出，十分灵活，用途也很广。

## 实验内容

实验步骤：

1. 建立Vivado工程文件，配置好环境
2. 编写主程序代码
3. 编写TestBench仿真代码
4. 执行仿真

关键源程序：

主程序：

module pc(

clk,rst,PC\_src\_sel,PC\_PIF,PC\_DX,jal\_offset,rs1\_data,jalr\_offset,imm\_b,PC\_IF,handler\_PC,epc

);

parameter PC\_JAL\_TARGET=0,PC\_JALR\_TARGET=1,PC\_BRANCH\_TARGET=2,PC\_REPLAY=3,PC\_HANDLER=4,PC\_EP=5;

input clk;

input [2:0] PC\_src\_sel;

input rst;

input [31:0]PC\_DX;

input [31:0]jal\_offset;

input [31:0]rs1\_data;

input [31:0]jalr\_offset;

input [31:0]imm\_b;

input [31:0]PC\_IF;

input [31:0]handler\_PC;

input [31:0]epc;

reg [31:0] temp;

output [31:0] PC\_PIF;

always @ (posedge clk,posedge rst)begin

if(rst)begin

temp<=0;

end

else

begin

case(PC\_src\_sel)

PC\_JAL\_TARGET:temp<=PC\_DX+jal\_offset;

PC\_JALR\_TARGET:temp<=rs1\_data+jalr\_offset;

PC\_BRANCH\_TARGET:temp<=PC\_DX+imm\_b;

PC\_REPLAY:temp<=PC\_IF;

PC\_HANDLER:temp<=handler\_PC;

PC\_EP:temp<=epc;

default:temp<=PC\_IF+4;

endcase

end

end

assign PC\_PIF=temp;

endmodule

仿真程序：

`timescale 10ns / 100ps

module pc\_tb();

reg clk;

reg rst;

reg [2:0]PC\_src\_sel;

wire [31:0]PC\_PIF;

reg [31:0]PC\_DX;

reg [31:0]jal\_offset;

reg [31:0]rs1\_data;

reg [31:0]jalr\_offset;

reg [31:0]imm\_b;

reg [31:0]PC\_IF;

reg [31:0]handler\_PC;

reg [31:0]epc;

pc pctb(clk,rst,PC\_src\_sel,PC\_PIF,PC\_DX,jal\_offset,rs1\_data,jalr\_offset,imm\_b,PC\_IF,handler\_PC,epc);

initial

begin

clk<=0;

rst<=0;

#2

rst<=1;

#3

rst<=0;

repeat(10)

begin

PC\_src\_sel<=$random;

PC\_DX<=$random;

jal\_offset<=$random;

rs1\_data<=$random;

jalr\_offset<=$random;

imm\_b<=$random;

PC\_IF<=$random;

handler\_PC<=$random;

epc<=$random;

#5;

end

#10000

$stop;

end

always #2

begin

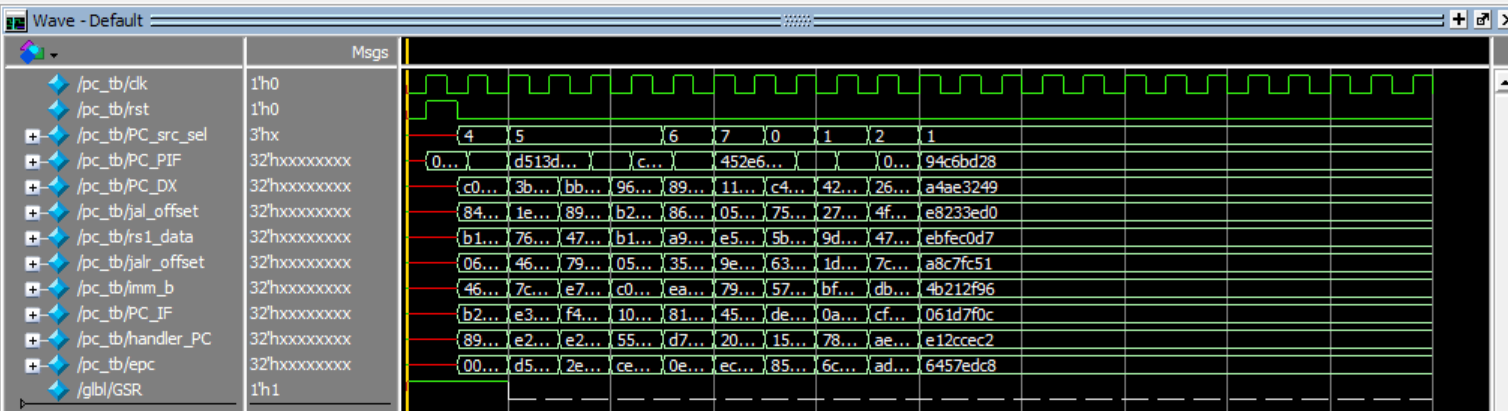
clk<=~clk;

end

endmodule

设计框图：

仿真波形：



## 总结与体会

这一次实验较为简单，我学会了编写PC寄存器。

注意：（不要对project目录直接打包，上交文件不符合要求视为未交）

实验报告（doc或docx，不能为pdf或wps文件）与实验文件：

|  |  |
| --- | --- |
| .v | Verilog设计文件 |
| \*\_tb.v | TestBench文件 |
| .xpr | Vivado工程文件 |
| .xdc | 约束文件 |
| .xci | IP定制文件 |
| .do | Modelsim仿真脚本记录文件 |
|  | 其他必要文件 |

以RAR格式打包，rar文件名格式为

学号姓名DSCPU实验序号.rar

如：14081866张山DSCPU2.rar，

然后，登录 网络教学平台

在作业栏目中上传附件