

# 实验报告

2021 年 5 月 21 日

成绩: \_\_\_\_\_

姓名	*****	学号	*****	班级	*****
专业	计算机科学与技术		课程名称	计算机组成原理课程设计	
任课老师	曾虹	指导老师	曾虹	机位号	31
实验序号	7	实验名称	取指令与指令译码操作		
实验时间	2021.5.21	实验地点	一教 225	实验设备号	31

## 一、实验程序源代码

```
`timescale 1ns / 1ps
module GetInst(
    input clk,
    input rst,
    output reg [31:0]PC,
    output [31:0]PC_new,
    output [31:0]Inst_code,
    output [5:0]opcode,func,
    output [4:0]rs,rt,rd,shamt,
    output [15:0]imm,
    output [25:0]address
);
    initial PC = 32'h00000000;

    assign PC_new = PC + 4;
    assign opcode = Inst_code[31:26];
    assign rs = Inst_code[25:21];
    assign rt = Inst_code[20:16];
    assign rd= Inst_code[15:11];
    assign shamt = Inst_code[10:6];
    assign func = Inst_code[5:0];
    assign imm = Inst_code[15:0];
    assign address = Inst_code[25:0];

    Inst_addr ROM1 (
        .clka(clk), // input clka
        .addra(PC[7:2]), // input [5 : 0] addra
        .douta(Inst_code) // output [31 : 0] douta
    );
```

```

always @(posedge clk or posedge rst)
begin
    if (rst)
        PC = 32'h00000000;
    else
        PC = PC_new;
end
endmodule

```

## 二、仿真测试代码

```

`timescale 1ns / 1ps
module TestInst;

    // Inputs
    reg clk;
    reg rst;

    // Outputs
    wire [31:0] PC;
    wire [31:0] PC_new;
    wire [31:0] Inst_code;
    wire [5:0] opcode;
    wire [5:0] func;
    wire [4:0] rs;
    wire [4:0] rt;
    wire [4:0] rd;
    wire [4:0] shamt;
    wire [15:0] imm;
    wire [25:0] address;

    // Instantiate the Unit Under Test (UUT)
    GetInst uut (
        .clk(clk),
        .rst(rst),
        .PC(PC),
        .PC_new(PC_new),
        .Inst_code(Inst_code),
        .opcode(opcode),
        .func(func),
        .rs(rs),
        .rt(rt),
        .rd(rd),
        .shamt(shamt),
        .imm(imm),

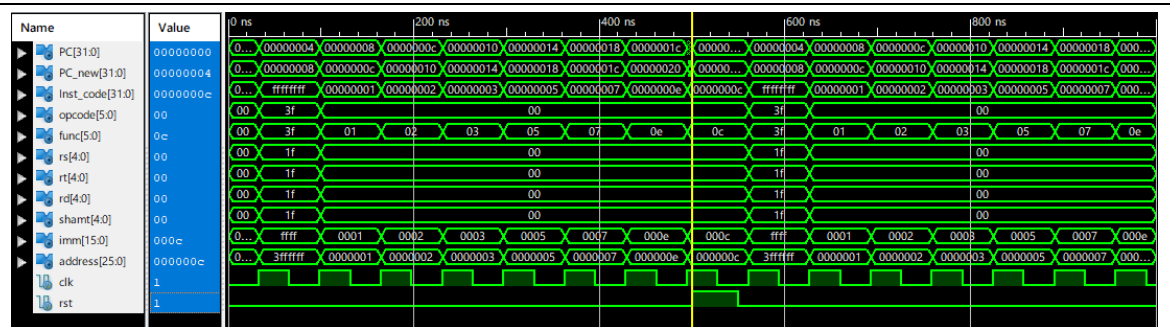
```

```
        .address(address)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        rst = 0;
        // Add stimulus here
        #500 rst = 1;
        #50 rst=0;
    end

    always #33 clk = ~clk;
endmodule
```

### 三、仿真波形



### 四、思考与探索

这次实验有了上一次存储器的底子，做起来无论是写调用还是创建 IP 核的 ROM 堆都熟练了很多，这次也完成了取指令和指令分段的任务。

复位后第一个时钟信号上升沿来临的时候读出的是地址 0x0（0 号单元）的指令。

整个实验流程还是很顺利的，只是因为没有做实验 6，获取实验 6 得到的指令的代码还是比较麻烦的。另外，这个实验其实好像不是真正的指令译码，只是取指令后给它分块了。具体怎么实现还要看后面的继续学习。其实做这个实验相当于还去熟悉了一下 PCSpim 软件，收获很大，知道计算机内部如何获取与分块指令了。