

取指单元仿真时序

SEU-09019204-曹邹颖

1. 设计的取指单元的仿真波形图

在实验中，我在选择指令存储器（ROM）的 IP 核时并没有选择 **Primitives Output Register** 这一选项，从而取指单元的仿真波形图如图 1；如果选择，由于寄存器需要时钟触发，会使得存储器的输出延迟一个时钟，经验证 MOOC 教程中便是如此，从而取指单元的仿真波形图如图 2 与 MOOC 演示一致。



图 1 取指单元的仿真波形图(未选择 Primitives Output Register)

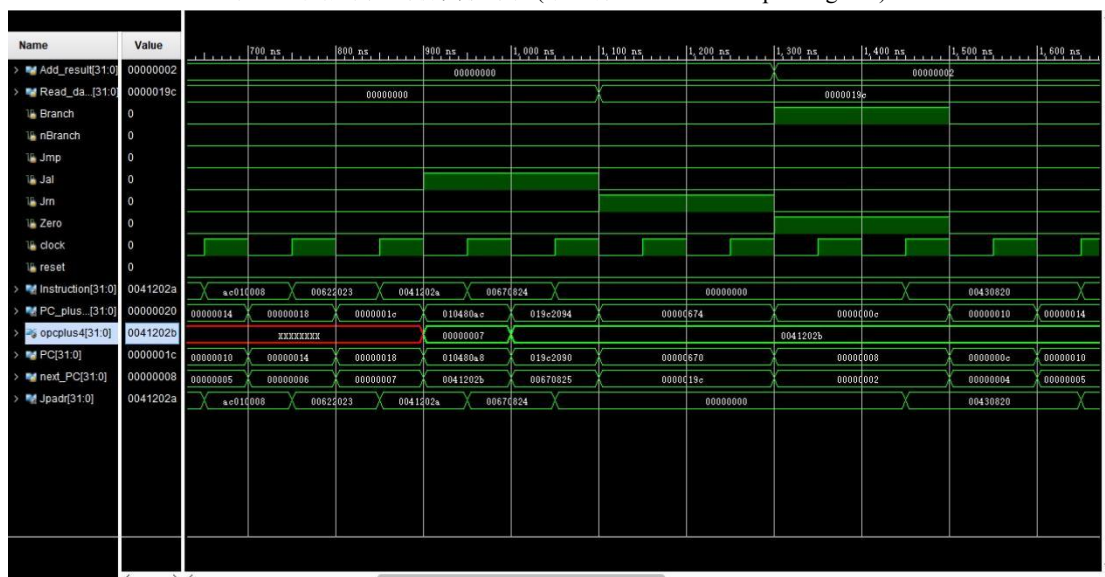


图 2 取指单元的仿真波形图(选择 Primitives Output Register)

2. ifect32.v

```
`timescale 1ns / 1ps
module Ifetc32(Instruction,PC_plus_4_out,Add_result,Read_data_1,Branch,
nBranch,Jump,Jal,Jrn,Zero,clock,reset,opcplus4);
    output[31:0] Instruction;           // 输出指令
    output[31:0] PC_plus_4_out;         // PC+4 的结果输出,送入加法器
    input[31:0]  Add_result;            // 加法器中算出的跳转地址,beq,bne
```

```

input[31:0]  Read_data_1;      // 寄存器组中读出, jr:(PC)←(rs)
input       Branch;           // beq
input       nBranch;          // bne
input       Jmp;              // j
input       Jal;              // jal
input       Jrn;              // jr
input       Zero;             // 是否相等
input       clock,reset;
output[31:0] opcplus4;        // jal 指令专用的 PC+4,送入数据选择器-7

wire[31:0]  PC_plus_4;        // PC+4
reg[31:0]   PC;               // PC 寄存器内容
reg[31:0]   next_PC;          // 下一条指令的 PC
wire[31:0]  Jpadr;
reg[31:0]   opcplus4;

//分配 64KB ROM, 编译器实际只用 64KB ROM
prgrom instmem(
    .clka(clock),              // input wire clka
    .addra(PC[15:2]),           // input wire [13 : 0] addra
    .douta(Jpadr)              // output wire [31 : 0] douta
);

assign Instruction = Jpadr;     // 取出指令
assign PC_plus_4 = {PC[31:2] + 1,2'b00}; // PC+4
assign PC_plus_4_out = PC_plus_4;

always @* begin
    if(Jrn) next_PC = Read_data_1; // jr
    else if((Branch&&Zero)||(nBranch&&!Zero)) // beq || bne
        next_PC = Add_result;
    else next_PC = {2'b00,PC_plus_4[31:2]}; // 一般情况
end

always @(negedge clock) begin // 时钟下降沿更改 PC
    if(reset) PC = 32'h00000000;
    else if(Jmp||Jal) begin // j || jal
        if(Jal)opcplus4 = {2'b00,PC_plus_4[31:2]}; // PC+4, 用于 jal,
        $31=PC+4, 右移两位以存入寄存器
        PC = {4'b0000,Instruction[27:0]<<2}; // 先左移再零扩展
    end
    else PC = next_PC<<2;
end
endmodule

```