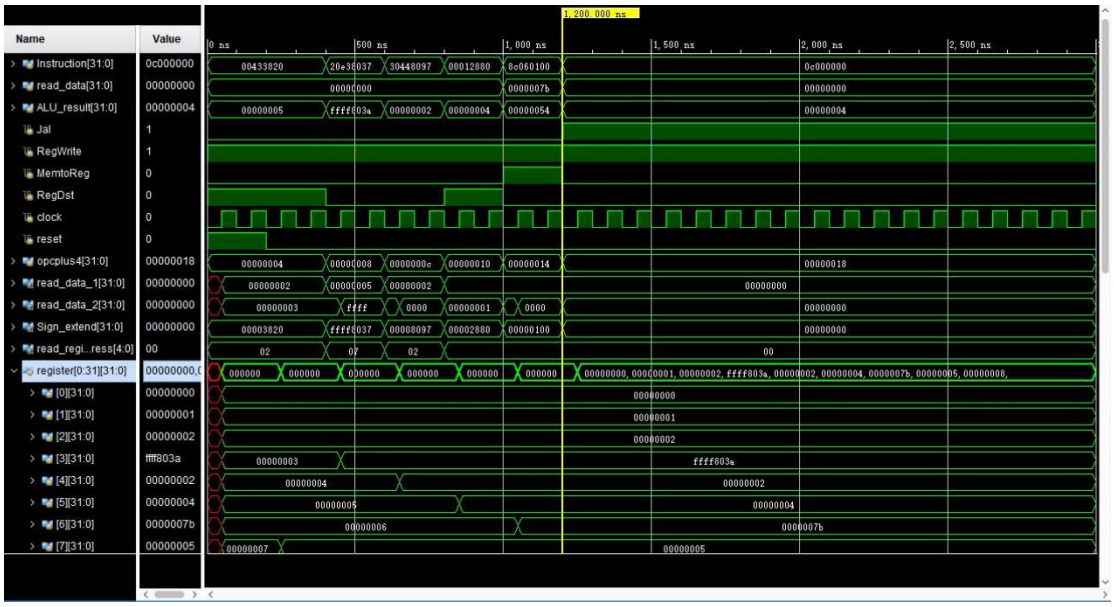
译码单元仿真时序

SEU-09019204-曹邹颖

1. 设计的译码单元的仿真波形图



2. idecode32.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
module Idecode32(read_data_1,read_data_2,Instruction,read_data,ALU_result,
                 Jal,RegWrite,MemtoReg,RegDst,Sign_extend,clock,reset,
                 opcplus4,read_register_1_address);
    output[31:0] read_data_1;
    output[31:0] read_data_2;
    input[31:0]  Instruction;
    input[31:0]  read_data;          // 从 DATA RAM or I/O port 取出的数据
    input[31:0]  ALU_result;         // 需要扩展立即数到 32 位
    input        Jal;
    input        RegWrite;
    input        MemtoReg;
    input        RegDst;
    output[31:0] Sign_extend;
    input        clock,reset;
    input[31:0]  opcplus4;                  // 来自取指单元，JAL 中用
    output[4:0]  read_register_1_address;   // rs

    wire[31:0] read_data_1;
    wire[31:0] read_data_2;
    reg[31:0] register[0:31];               // 寄存器组共 32 个 32 位寄存器
```

```
    reg[4:0] write_register_address;
    reg[31:0] write_data;
    wire[4:0] read_register_2_address;                  // rt
    wire[4:0] write_register_address_1;                 // rd(r-form)
    wire[4:0] write_register_address_0;                 // rt(i-form)
    wire[15:0] Instruction_immediate_value;             // immediate
    wire[5:0] opcode;                                   // op

    wire sign;
    assign opcode = Instruction[31:26];
    assign read_register_1_address = Instruction[25:21];        // rs
    assign read_register_2_address = Instruction[20:16];        // rt
    assign write_register_address_1 = Instruction[15:11];       // rd
    assign write_register_address_0 = Instruction[20:16];       // rt(i-form)
    assign Instruction_immediate_value = Instruction[15:0];     // immediate

    assign read_data_1 = register[read_register_1_address];
    assign read_data_2 = register[read_register_2_address];

    assign sign = Instruction[15];
    // andi,ori,xori,sltui 零扩展, 其余符号扩展
    assign Sign_extend =
(opcode==6'b001100||opcode==6'b001101||opcode==6'b001110||opcode==6'b0010
11)?{16'd0,Instruction_immediate_value}:{{16{sign}},Instruction_immediate_val
ue};

    always @* begin          // 这个进程指定不同指令下的目标寄存器
        if(Jal)
            write_register_address = 5'd31;
        else if(RegDst)
            write_register_address = write_register_address_1;
        else
            write_register_address = write_register_address_0;
    end

    always @* begin          // 实现结构图中多路选择器,准备要写的数据
        if(Jal)
            write_data = opcplus4;
        else if(MemtoReg)
            write_data = read_data;
        else
            write_data = ALU_result;
     end
```

```
    integer i;
    always @(posedge clock) begin                  // 本进程写目标寄存器
        if(reset==1) begin                         // 初始化寄存器组
            for(i=0;i<32;i=i+1) register[i] <= i;
        end else if(RegWrite==1) begin             // 注意寄存器 0 恒等于 0
            if(write_register_address != 5'b00000)
                register[write_register_address] = write_data;
        end
    end
endmodule
```