

中国科学技术大学计算机科学与技术学院
《计算机组成原理》实验报告



实验题目：流水线 CPU 设计

学生姓名：郭耸霄_____

学生学号：PB20111712_____

完成日期：2022 年 4 月 11 日

计算机实验教学中心制

2020 年 09 月

1 实验题目

流水线 CPU 设计。

2 实验环境

开发板 Nexy4-DDR xc7a100tcsq324-1。

计算机 Windows Surface Pro 7 Model 1866 i7。

操作系统 Windows 11 Pro Version 22H2 (OS Build 22581.200)。

集成设计环境 Vivado 2019.1。

3 实验目的

- 1、理解流水线 CPU 的结构和工作原理。
- 2、掌握流水线 CPU 的设计和调试方法，特别是流水线中数据相关和控制相关的处理。
- 3、熟练掌握数据通路和控制器的设计和描述方法。

4 实验步骤

- 1、设计无数据和控制相关处理的流水线 CPU。
- 2、设计仅有数据相关处理的流水线 CPU。
- 3、设计完整的有数据和控制相关处理的流水线 CPU。

5 逻辑设计

5.1 数据通路

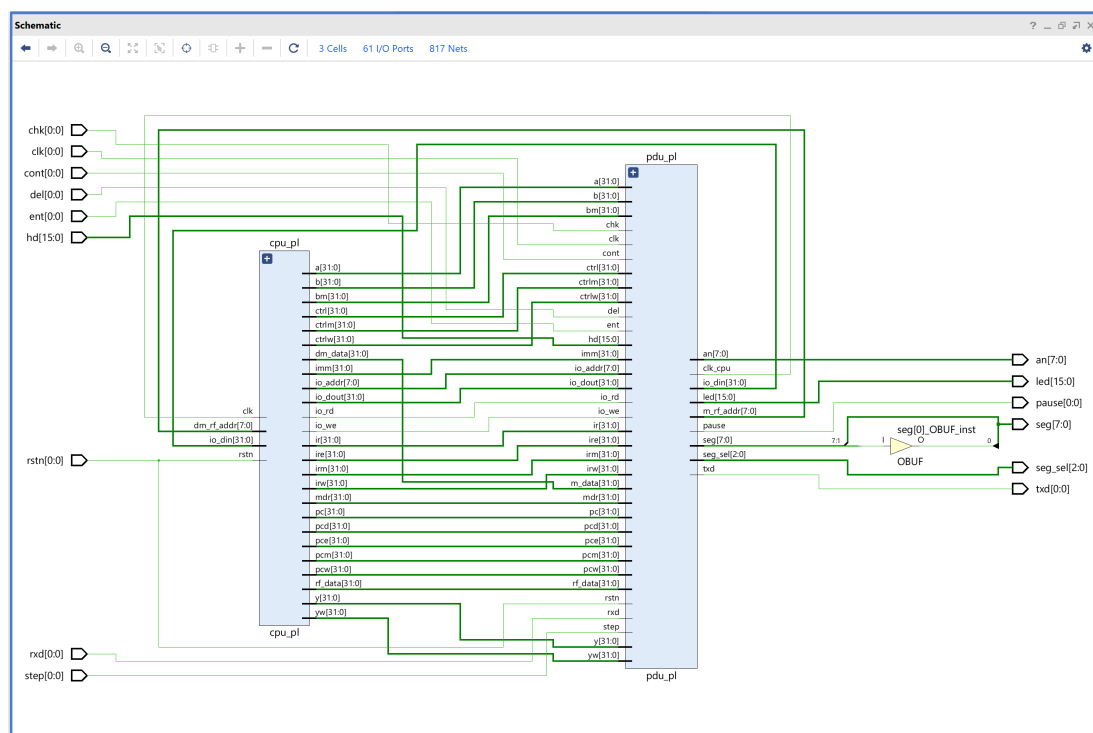


图 1: 顶层模块

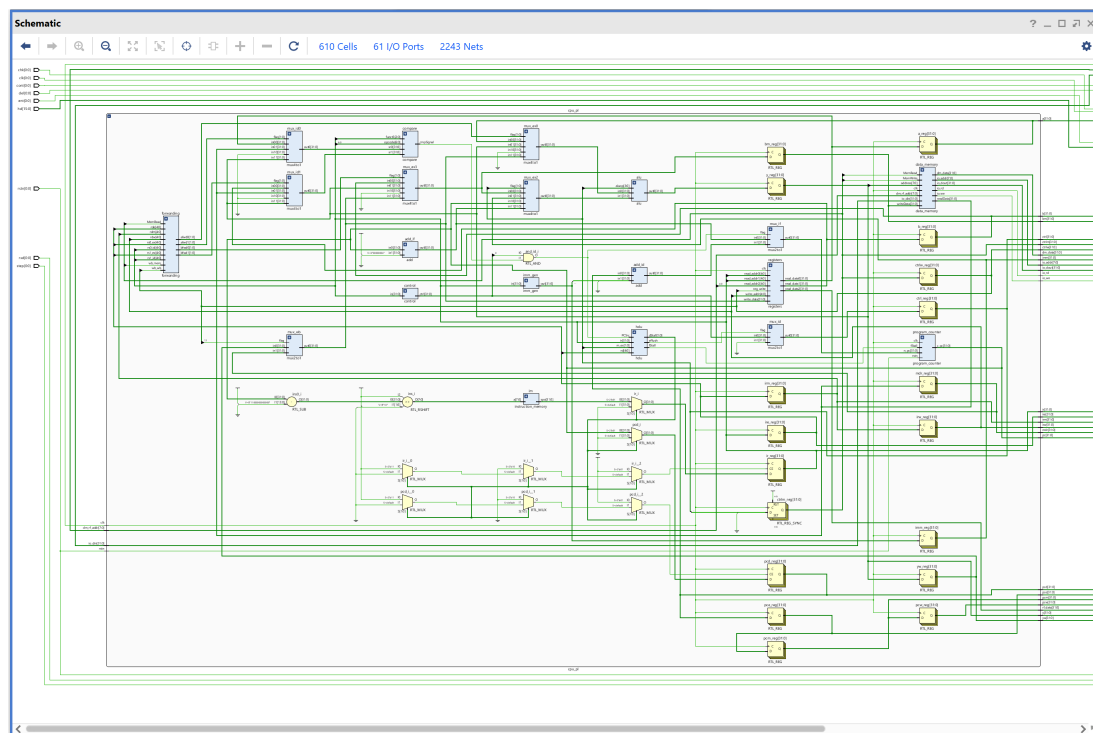


图 2: CPU 模块

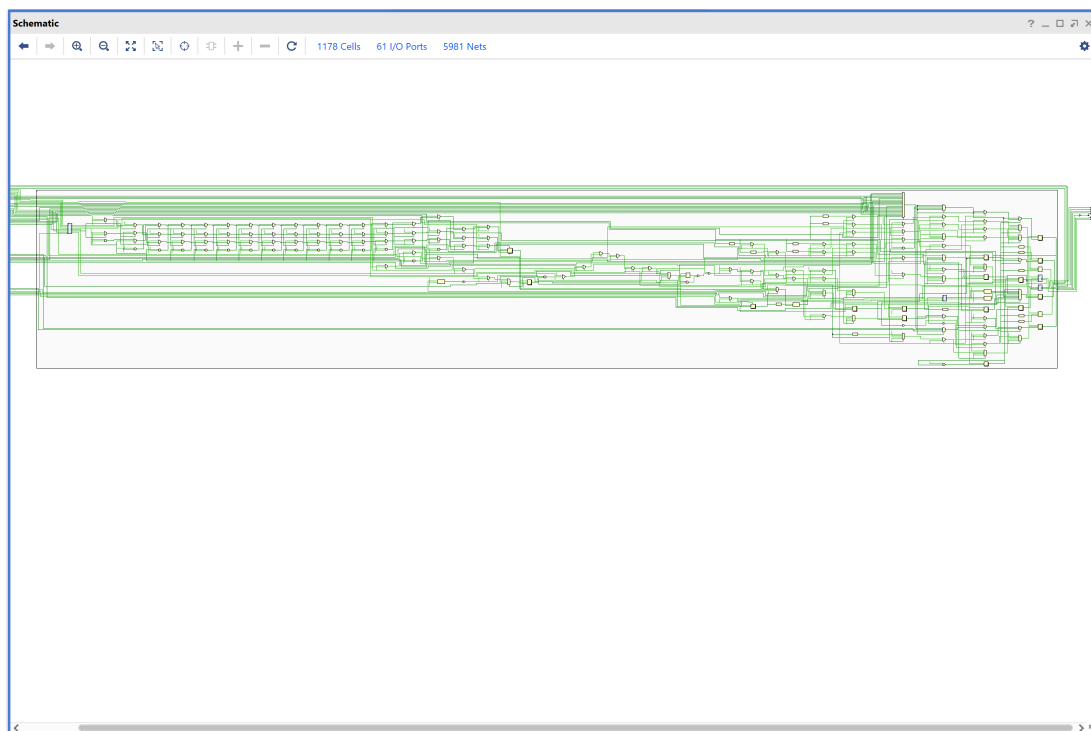


图 3: PDU 模块

5.2 状态机

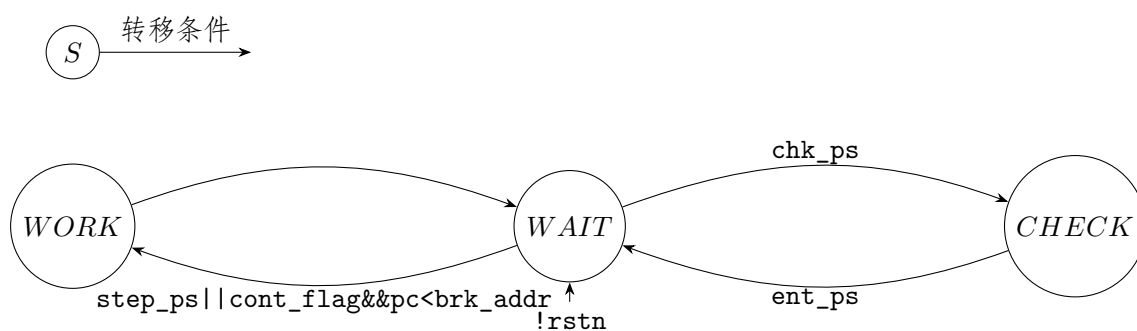


图 4: PDU 状态机

6 核心代码

这里仅放置 CPU 与 PDU 两个模块的代码，其余代码请看附件 0：实验源码。

CPU

```

1 'timescale 1ns / 1ps
2 //////////////////////////////////////
3 // Company: USTC
4 // Engineer: 郭耸霄
5 //
    
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
6 // Create Date: 03/22/2022 08:44:56 AM
7 // Design Name: RISC-V架构5级流水CPU
8 // Module Name: cpu_pl
9 // Project Name: 流水线CPU设计
10 // Target Devices: Nexys4-DDR
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////

21
22
23 module cpu_pl (
24     input clk ,
25     input rstn ,
26
27     //IO_BUS
28     output [7:0] io_addr ,      //外设地址
29     output [31:0] io_dout ,     //向外设输出的数据
30     output io_we ,             //向外设输出数据时的写使能信号
31     output io_rd ,             //从外设输入数据时的读使能信号
32     input [31:0] io_din ,       //来自外设输入的数据
33
34     //Debug_BUS
35     input [7:0] dm_rf_addr ,     //存储器(MEM)或寄存器堆(RF)的调试
        读口地址
36     output [31:0] rf_data ,      //从RF读取的数据
37     output [31:0] dm_data ,     //从MEM读取的数据
38     output [31:0] pc ,
39
40     //ID 流水段寄存器
41     output reg [31:0] pcd ,
42     output reg [31:0] ir ,
43
44     //EX 流水段寄存器
45     output reg [31:0] pce ,
46     output reg [31:0] ire ,
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
47  output reg [31:0] ctrl ,
48  output reg [31:0] a ,
49  output reg [31:0] b ,
50  output reg [31:0] imm ,
51
52  //MEM 流水段寄存器
53  output reg [31:0] pcm ,
54  output reg [31:0] irm ,
55  output reg [31:0] ctrlm ,
56  output reg [31:0] y ,
57  output reg [31:0] bm ,
58
59  //WB 流水段寄存器
60  output reg [31:0] pcw ,
61  output reg [31:0] irw ,
62  output reg [31:0] ctrlw ,
63  output reg [31:0] mdr ,
64  output reg [31:0] yw
65 );
66
67  initial begin
68      ir=0;
69  end
70
71
72  //Control
73
74  wire [0:0] PCSrc;
75  wire [0:0] RegWrite;
76
77  //IF
78
79  wire [31:0] n_pc;
80  wire [31:0] pc_plus4;
81  wire [31:0] instruction;
82  wire [7:0] ins;
83  wire [31:0] addSum;
84  wire [0:0] fStall;
85  wire [1:0] dStall;
86  wire [4:0] write_addr;
87
88  assign ins=(pc-32'h3000)>>2;
89
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
90     program_counter program_counter(
91         .clk ( clk ) ,
92         .rstn ( rstn ) ,
93         .fStall ( fStall ) ,
94         .n_pc ( n_pc ) ,
95         .c_pc ( pc )
96     );
97
98     instruction_memory im(
99         .a ( ins ) ,
100        .spo ( instruction )
101    );
102
103    add add_if(
104        .in0 ( pc ) ,
105        .in1 ( 32'd4 ) ,
106        .out0 ( pc_plus4 )
107    );
108
109    mux2to1 mux_if(
110        .in0 ( pc_plus4 ) ,
111        .in1 ( addSum ) ,
112        .flag ( PCSrc ) ,
113        .out0 ( n_pc )
114    );
115
116    always@(posedge clk) begin
117        if ( dStall==1)begin
118            pcd<=pc;
119            ir<=instruction;
120        end else if ( dStall==2)begin
121            pcd<=pcd;
122            ir<=ir;
123        end else if ( dStall==3)begin
124            pcd<=0;
125            ir<=0;
126        end
127    end
128
129    //ID
130
131    wire [31:0] write_data1;
132    wire [31:0] read_data0;
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
133     wire [31:0] read_data1;
134     wire [31:0] ctrl_out;
135     wire [31:0] ctrl_choose;
136     wire [1:0] m_ex;
137     wire [31:0] imm_num;
138     wire [0:0] eFlush;
139     wire [31:0] cmpsr0;
140     wire [31:0] cmpsr1;
141     wire [1:0] afwd1;
142     wire [1:0] bfwd1;
143     wire [31:0] ALUresult;
144     wire [31:0] readData;
145
146
147     registers registers(
148         .clk (clk),
149         .reg_write (RegWrite),
150         .read_addr0 (ir [19:15]),
151         .read_addr1 (ir [24:20]),
152         .read_addr2 (dm_rf_addr [4:0]),
153         .write_addr (write_addr),
154         .write_data (write_data1),
155         .read_data0 (read_data0),
156         .read_data1 (read_data1),
157         .read_data2 (rf_data)
158     );
159
160     imm_gen imm_gen(
161         .in (ir),
162         .out (imm_num));
163
164     control control(
165         .in (ir),
166         .ctrl (ctrl_out)
167     );
168
169     mux2to1 mux_id(
170         .in0 (ctrl_out),
171         .in1 (0),
172         .flag (eFlush),
173         .out0 (ctrl_choose)
174     );
175
```


实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
176     hdu hdu(  
177         .ir(ir),  
178         .PCSrc(PCSrc),  
179         .rd(ire[11:7]),  
180         .m_ex(m_ex),  
181         .fStall(fStall),  
182         .dStall(dStall),  
183         .eFlush(eFlush)  
184     );  
185  
186     add add_id(  
187         .in0(pcd),  
188         .in1({imm_num[30:0],1'b0}),  
189         .out0(addSum)  
190     );  
191  
192     and and_id(  
193         PCSrc,  
194         ctrl_out[0],  
195         cmpSignal  
196     );  
197  
198     mux4to1 mux_id0(  
199         .in00(read_data0),  
200         .in01(readData),  
201         .in10(0),  
202         .in11(y),  
203         .flag(afwd1),  
204         .out0(cmpsr0)  
205     );  
206  
207     mux4to1 mux_id1(  
208         .in00(read_data1),  
209         .in01(readData),  
210         .in10(0),  
211         .in11(y),  
212         .flag(bfwd1),  
213         .out0(cmpsr1)  
214     );  
215  
216     compare compare(  
217         .sr0(cmpsr0),  
218         .sr1(cmpsr1),
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
219     .funct3( ir [14:12] ) ,
220     .opcode( ir [6:0] ) ,
221     .cmpSignal( cmpSignal )
222 );
223
224 always @(posedge clk) begin
225     ctrl<=ctrl_choose;
226     pce<=pcd;
227     a<=read_data0;
228     b<=read_data1;
229     imm<=imm_num;
230     ire<=ir;
231 end
232
233 //EX
234
235 wire [31:0] ALUin0;
236 wire [31:0] ALUin1;
237 wire [1:0] ALUSrc;
238 wire [3:0] ALUOp;
239 wire [0:0] wb_ex;
240 wire [6:0] ex;
241 wire [1:0] afwd0;
242 wire [1:0] bfwd0;
243 wire [31:0] bOrImm;
244 wire [0:0] wb_mem;
245 wire [0:0] wb_wb;
246 wire [31:0] b_real;
247
248 assign wb_ex=ctrl [12];
249 assign m_ex=ctrl [9:8];
250 assign ex=ctrl [6:0];
251 assign ALUOp=ex [4:1];
252 assign ALUSrc=ex [6:5];
253
254 mux4to1 mux_ex2(
255     .in00( b_real ) ,
256     .in01( imm ) ,
257     .in10( pc ) ,
258     .in11( pc_plus4 ) ,
259     .flag( ALUSrc ) ,
260     .out0( ALUin1 )
261 );
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
262
263     mux4to1 mux_ex0(
264         .in00(a) ,
265         .in01(y) ,
266         .in10(0) ,
267         .in11(write_data1) ,
268         .flag(afwd0) ,
269         .out0(ALUin0)
270     );
271
272     // mux4to1 mux_ex1(
273     //     .in00(bOrImm) ,
274     //     .in01(y) ,
275     //     .in10(0) ,
276     //     .in11(write_data1) ,
277     //     .flag(bfwd0) ,
278     //     .out0(ALUin1)
279     // );
280
281     mux4to1 mux_ex3(
282         .in00(b) ,
283         .in01(y) ,
284         .in10(0) ,
285         .in11(write_data1) ,
286         .flag(bfwd0) ,
287         .out0(b_real)
288     );
289
290     alu alu(
291         .in0(ALUin0) ,
292         .in1(ALUin1) ,
293         .aluop(ALUOp) ,
294         .out0(ALUresult) ,
295         .zero(Zero)
296     );
297
298     forwarding forwarding(
299         .wb_mem(wb_mem) ,
300         .wb_wb(RegWrite) ,
301         .MemRead(MemRead) ,
302         .rs0_ex(ire[19:15]) ,
303         .rs1_ex(ire[24:20]) ,
304         .rs0_id(ir[19:15]) ,
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
305     .rs1_id( ir [24:20] ) ,
306     .rde( ire [11:7] ) ,
307     .rdm( irm [11:7] ) ,
308     .rdw( irw [11:7] ) ,
309     .afwd0( afwd0 ) ,
310     .bfwd0( bfwd0 ) ,
311     .afwd1( afwd1 ) ,
312     .bfwd1( bfwd1 )
313 );
314
315 always @(posedge clk) begin
316     ctrlm<={ctrl [31:8] ,8'b0};
317     y<=ALUresult;
318     bm<=b_real;
319     irm<=ire;
320     pcm<=pce;
321 end
322
323 //MEM
324
325 assign MemRead=ctrlm [9];
326 assign MemWrite=ctrlm [8];
327 assign wb_mem=ctrlm [12];
328
329 data_memory data_memory(
330     .clk( clk ) ,
331     .MemWrite(MemWrite) ,
332     .MemRead(MemRead) ,
333     .address(y [9:2] ) ,
334     .writeData(bm) ,
335     .readData(readData) ,
336     .dm_rf_addr(dm_rf_addr) ,
337     .dm_data(dm_data) ,
338     .io_addr(io_addr) ,
339     .io_din(io_din) ,
340     .io_dout(io_dout) ,
341     .io_rd(io_rd) ,
342     .io_we(io_we)
343 );
344
345 always @(posedge clk) begin
346     ctrlw<=ctrlm;
347     mdr<=readData;
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
348     yw<=y;
349     irw<=irm;
350     pcw<=pcm;
351     end
352
353 //WB
354
355     assign write_addr=irw[11:7];
356     assign RegWrite=ctrlw[12];
357     assign MemtoReg=ctrlw[13];
358
359     mux2to1 mux_wb(
360         .in0(yw),
361         .in1(mdr),
362         .flag(MemtoReg),
363         .out0(write_data1)
364     );
365
366 endmodule
```

PDU

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
3  // Company: USTC
4  // Engineer: 郭耸霄
5  //
6  // Create Date: 03/31/2022 11:23:01 AM
7  // Design Name: 流水线CPU的调试单元
8  // Module Name: pdu_pl
9  // Project Name: 流水线CPU设计
10 // Target Devices: Nexys4-DDR
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 – File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
21
22
23 module pdu_pl (
24
25     input clk ,                //clk100mhz
26     input rstn ,               //cpu_reseten
27     input step ,              //btneu
28     input cont ,              //btnd
29     input chk ,               //btnr
30     input ent ,               //btnc
31     input del ,               //btnl
32     input [15:0] hd ,         //sw15-0
33     output reg clk_cpu ,
34     output reg pause ,        //led16r
35     output reg [15:0] led ,   //led15-0
36     output [7:0] an ,         //an7-0
37     output [7:0] seg ,        //ca-cg
38     output reg [2:0] seg_sel , //led17
39     input rxd ,               //串口接收数据
40     output txd ,              //串口发送数据
41
42     //IO_BUS
43     input [7:0] io_addr ,
44     input [31:0] io_dout ,
45     input io_we ,
46     input io_rd ,
47     output reg [31:0] io_din ,
48
49     //Debug_BUS
50     output reg [7:0] m_rf_addr ,
51     input [31:0] rf_data ,
52     input [31:0] m_data ,
53
54     //IF流水段寄存器
55     input [31:0] pc ,
56
57     //ID 流水段寄存器
58     input [31:0] pcd ,
59     input [31:0] ir ,
60
61     //EX 流水段寄存器
62     input [31:0] pce ,
63     input [31:0] ire ,
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
64     input [31:0] ctrl ,
65     input [31:0] a ,
66     input [31:0] b ,
67     input [31:0] imm ,
68     //MEM 流水段寄存器
69     input [31:0] pcm ,
70     input [31:0] irm ,
71     input [31:0] ctrlm ,
72     input [31:0] y ,
73     input [31:0] bm ,
74
75     //WB 流水段寄存器
76     input [31:0] pcw ,
77     input [31:0] irw ,
78     input [31:0] ctrlw ,
79     input [31:0] mdr ,
80     input [31:0] yw
81
82 );
83
84     parameter WAIT=0;
85     parameter CHK=1;
86     parameter WORK=2;
87
88     integer i ;
89
90     wire [0:0] step_ps ;
91     wire [0:0] cont_ps ;
92     wire [0:0] chk_ps ;
93     wire [0:0] ent_ps ;
94     wire [0:0] del_ps ;
95     wire [15:0] hd_ps ;
96
97     reg [2:0] state ;
98     reg [2:0] n_state ;
99     reg [31:0] addr ;
100    reg [31:0] brk_addr ;
101    reg [31:0] show ;
102    reg [0:0] cont_flag ;
103    reg [0:0] chk_flag ;
104
105    reg [15:0] led_dout ;
106    reg [31:0] sw_din ;
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
107     reg [31:0] pol_dout;
108     reg [31:0] pol_dout_vld;
109     reg [31:0] pol_din;
110     reg [31:0] pol_din_vld;
111     reg [31:0] tx_data;
112     reg [31:0] tx_rdy;
113     reg [31:0] rx_data;
114     reg [31:0] rx_vld;
115     reg [31:0] tm_data;
116     reg [31:0] tx_rd;
117     wire [0:0] vld_temp;
118     wire [0:0] rd_temp;
119     wire [7:0] data_temp;
120
121     dis dis(
122         .clk ( clk ) ,
123         .rst ( rstn ) ,
124         .in ( show ) ,
125         .an ( an ) ,
126         .seg ( seg )
127     );
128
129     db_ps db_ps(
130         .clk ( clk ) ,
131         .step ( step ) ,
132         .cont ( cont ) ,
133         .chk ( chk ) ,
134         .ent ( ent ) ,
135         .del ( del ) ,
136         .hd ( hd ) ,
137         .step_ps ( step_ps ) ,
138         .cont_ps ( cont_ps ) ,
139         .chk_ps ( chk_ps ) ,
140         .ent_ps ( ent_ps ) ,
141         .del_ps ( del_ps ) ,
142         .hd_ps ( hd_ps )
143     );
144
145     tx u_tx (
146         .clk                ( clk                ) ,
147         .rst                ( rstn                ) ,
148         .tx_ready           ( tx_rdy [0]          ) ,
149         .tx_data            ( tx_data [7:0] ) ,
```


实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
150
151         .tx                                ( txd          ),
152         .tx_rd                             ( rd_temp       )
153     );
154
155     rx u_rx (
156         .clk                                ( clk           ),
157         .rst                                ( rstn           ),
158         .rx                                  ( rxd           ),
159
160         .rx_vld                             ( vld_temp       ),
161         .rx_data                            ( data_temp      )
162     );
163
164     initial begin
165         addr=0;
166         pol_dout_vld=0;
167     end
168
169     always @(posedge clk) begin
170         tm_data<=tm_data+1;
171     end
172
173     always @(*) begin
174         rx_vld[0]=vld_temp;
175         rx_data[7:0]=data_temp;
176         tx_rd[0]=rd_temp;
177     end
178
179     always @(posedge clk) begin
180
181         if (del_ps) begin
182             addr<={4'b0, addr[31:4]};
183         end else begin
184             for (i=0; i<16; i=i+1) begin
185                 if (hd_ps[i]) begin
186                     addr<={addr[27:0], 4'b0}+i;
187                 end
188             end
189         end
190         // if (del) begin
191         //     addr<={4'b0, addr[31:4]};
192         // end else begin
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
193         //      for ( i=0;i <16;i=i+1)begin
194         //          if (hd[ i ]) begin
195         //              addr<={addr[27:0] ,4'b0}+i ;
196         //          end
197         //      end
198         //  end
199         if (!rstn)begin
200             state<=WAIT;
201         end else begin
202             state<=n_state;
203         end
204
205     end
206
207     always @(posedge clk) begin
208
209         case ( n_state )
210
211         WAIT:begin
212
213             if ( pol_dout_vld )begin
214                 seg_sel <=4;
215             end else if ( pol_din_vld )begin
216                 seg_sel <=2;
217             end
218
219             show<=pol_dout_vld?pol_dout:addr;
220             clk_cpu <=0;
221             pause <=1;
222             led<=led_dout;
223             sw_din<=hd_ps;
224
225             if ( tx_rd )begin
226                 tx_rdy <=0;
227             end else
228                 if ( ent_ps )begin
229                     //  if ( ent )begin
230                         pol_dout_vld <=0;
231                         pol_din_vld <=0;
232
233                     end else if ( io_we )begin
234                         case ( io_addr )
235                             6:begin
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
236             tx_rdy<=1;
237             tx_data<=io_dout;
238         end
239     2: begin
240         pol_dout_vld<=1;
241         pol_dout<=io_dout;
242     end
243     0: begin
244         led_dout<=io_dout[15:0];
245     end
246     default: ;
247 endcase
248 end else if (io_rd) begin
249     case (io_addr)
250     10: begin
251         io_din<=tm_data;
252     end
253     9: begin
254         io_din<=rx_vld;
255     end
256     8: begin
257         io_din<=rx_data;
258     end
259     7: begin
260         io_din<=tx_rdy;
261     end
262     5: begin
263         io_din<=pol_din_vld;
264     end
265     4: begin
266         //pol_din_vld<=1;
267         //pol_din<=addr;
268         //io_din<=pol_din;
269         io_din<=addr;
270     end
271     3: begin
272         io_din<=pol_dout_vld;
273     end
274     1: begin
275         io_din<=sw_din;
276     end
277     default: ;
278 endcase
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
279         end else begin
280             if (chk_ps) begin
281                 m_rf_addr<=addr [7:0];
282             end else if (cont_ps) begin
283                 cont_flag<=1;
284                 brk_addr<=addr [15:0];
285             end else if (pc==brk_addr) begin
286                 cont_flag<=0;
287             end
288             // if (chk)
289             //     m_rf_addr<=addr [7:0];
290             // else if (cont) begin
291             //     cont_flag<=1;
292             //     brk_addr<=addr [15:0];
293             // end else if (pc==brk_addr) begin
294             //     cont_flag<=0;
295             // end
296         end
297
298     end
299
300     CHK: begin
301
302         led<=m_rf_addr;
303
304         if (step_ps) begin
305             m_rf_addr<=addr [7:0];
306         end
307
308         if (chk_ps) begin
309
310             seg_sel<=1;
311
312             case (addr [15:12])
313             4: begin
314                 case (addr [7:0])
315                     8'h00: show<=pc;
316                     8'h10: show<=pcd;
317                     8'h11: show<=ir;
318                     8'h20: show<=pce;
319                     8'h21: show<=ire;
320                     8'h22: show<=ctrl;
321                     8'h23: show<=a;
```

实 验 报 告

11 系 20 级 3 班

郭耸霄 PB20111712

2022 年 4 月 11 日

```
322             8'h24:show<=b;
323             8'h25:show<=imm;
324             8'h30:show<=pcm;
325             8'h31:show<=irm;
326             8'h32:show<=ctrlm;
327             8'h33:show<=y;
328             8'h34:show<=bm;
329             8'h40:show<=pcw;
330             8'h41:show<=irw;
331             8'h42:show<=ctrlw;
332             8'h43:show<=mdr;
333             8'h44:show<=yw;
334             default: ;
335             endcase
336         end
337         8:begin
338             show<=rf_data;
339             m_rf_addr<=m_rf_addr+1;
340         end
341         12:begin
342             show<=m_data;
343             m_rf_addr<=m_rf_addr+1;
344         end
345         default:;
346         endcase
347     end else begin
348         seg_sel<=2;
349     end
350
351 end
352
353 WORK:begin
354     clk_cpu<=1;
355     pause<=0;
356 end
357
358 endcase
359
360 end
361
362 always @(*) begin
363
364     case (state)
```

```
365
366     WAIT: begin
367         if (chk_ps) begin
368             n_state=CHK;
369         end else if (step_ps || cont_flag && pc < brk_addr) begin
370             n_state=WORK;
371         end else begin
372             n_state=WAIT;
373         end
374     end
375
376     CHK: begin
377         if (ent_ps) begin
378             n_state=WAIT;
379         end else begin
380             n_state=CHK;
381         end
382     end
383
384     WORK: begin
385         n_state=WAIT;
386     end
387
388     default: n_state=n_state;
389
390     endcase
391
392 end
393
394 endmodule
```

7 下载结果

请看附件 1：下载结果视频。

8 结果分析

结果符合设计预期。视频中原数据寄存器中存有的数组经过排序变成升序排列。进入检查模式，PC、寄存器堆、数据寄存器的数据均符合预期。

9 实验总结

9.1 实验过程

在实验中，我遇到了许多较难解决的错误，现列举如下：

现象 0 同时读写寄存器堆时读取的值没有更新。

问题 0 没有设计写优先方案。

解决方案 0 寄存器堆模块根据读写地址选择读数据为寄存器内容或写数据。

现象 1 立即数生成器生成的立即数异常。

问题 1 立即数生成器接入了 IF 阶段的指令，导致生成的立即数提前一个周期。

解决方案 1 将立即数生成器的输入连接为 ID 段的指令。

现象 2 执行分支指令时，跳转写入新的 PC 失败与不跳转 PC 被强制更新无法同时避免。

问题 2 控制 IF/ID 指令寄存器的 fStall 仅有 1 位，却需要有正常写入、清零与不写入 3 种功能。

解决方案 2 将 fStall 改为 2 位。

现象 3 执行 SW 指令同时 Forwarding 时，ALUin1 进入的不是立即数 imm，而是前移的 ALUresult 或 write_data1。

问题 3 b fwd0 信号一方面需要选择前移的数据，另一方面需要选择前一级生成的立即数 imm。

解决方案 3 交换分别以 b fwd0 与 ALUSrc 两个信号为控制信号的 MUX 的顺序，即先选择前移数据，后选择立即数。

现象 4 轮询输出时，输出有效标志位置位不及时。

问题 4 输出有效标志位是 PDU 模块根据输出数据控制的，无法通过 CPU 的 Forwarding 模块进行数据前移。

解决方案 4 更改汇编程序逻辑，先用一个循环等待输出成功，再用一个循环检测输出是否有效。

9.2 实验收获

本次实验在第 2 次实验基础上，仍然用了 26 个小时，可以说是十分困难。不过在实验过程中遇到并解决了很多在理论课上没有想到的问题，一方面使我的硬件设计与调试能力有了进一步的提高，尤其我在本次实验中第一次实现了有关串口的部分。另一方面通过这次实验，我初步地了解了流水线 CPU 的工作原理，对计算机有了更深的认识。稍有遗憾的是设计的电路性能较为低劣，只能在比较低的时钟频率下运行。或许学到计算机体系结构课程，就有机会设计出性能更好的 CPU 吧。

10 意见建议

可以考虑每次实验结束后下发班级内的优秀代码以便互相学习。