Lab 6 report

PB22051022王嘉宁

实验目的与内容

> 结合前递模块与段间寄存器控制模块，得到一个能正常运行的完整流水线 CPU。

逻辑设计

SegCtrl

```
module SegCtrl(···
);
always @(*) begin
    if (rf_we_ex == 1 && rf_wa_ex != 0 && rf_wd_sel_ex == 2'b10 || npc_sel_ex ==
2'b01) begin
        if(npc_sel_ex == 2'b01) begin
            stall_pc = 0;
            stall_if_id = 0;
            flush_if_id = 1;
            flush_id_ex = 1;
        end
        else if (rf_ra0_id == rf_wa_ex || rf_ra1_id == rf_wa_ex) begin
            stall_pc = 1;
            stall_if_id = 1;
            flush_if_id = 0;
            flush_id_ex = 1;
        end
        else begin
            stall_pc = 0;
            stall_if_id = 0;
            flush_if_id = 0;
            flush_id_ex = 0;
        end
    end
    else begin
        stall_pc = 0;
        stall_if_id = 0;
        flush_if_id = 0;
        flush_id_ex = 0;
    end
end
endmodule
```

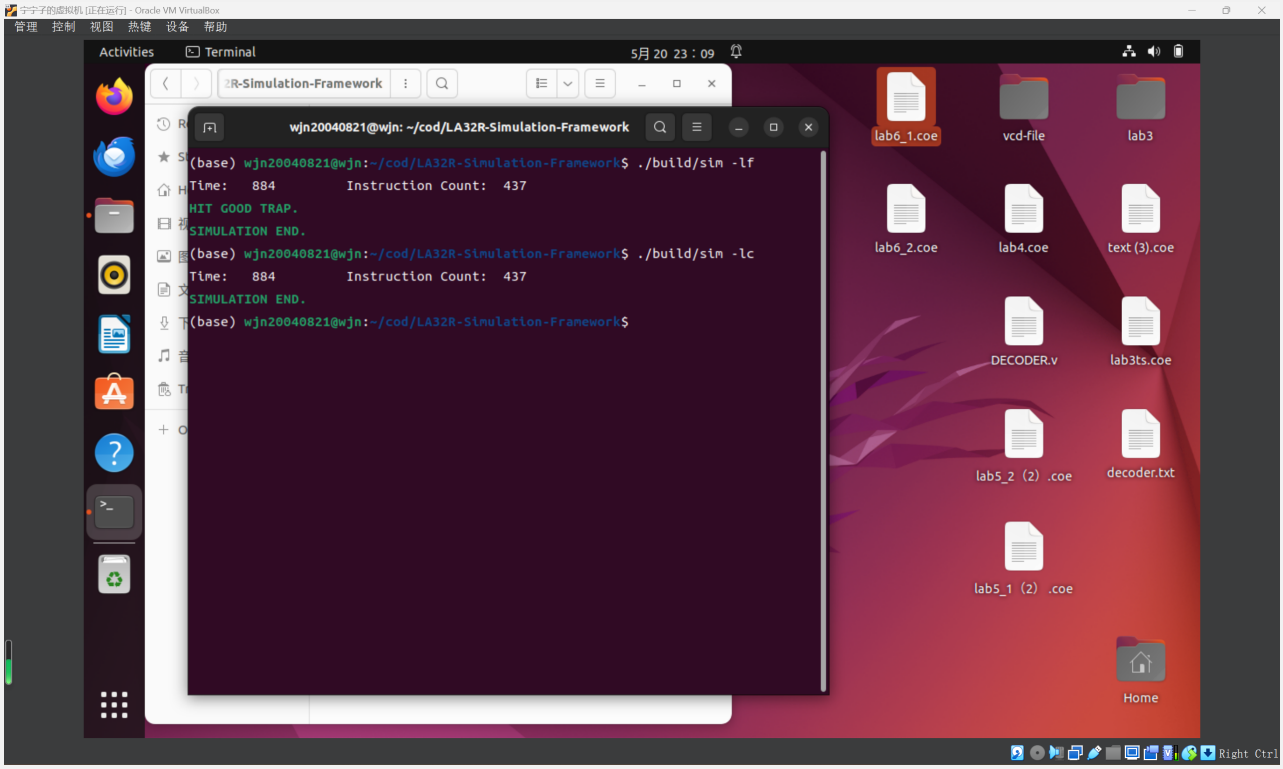
Forwarding

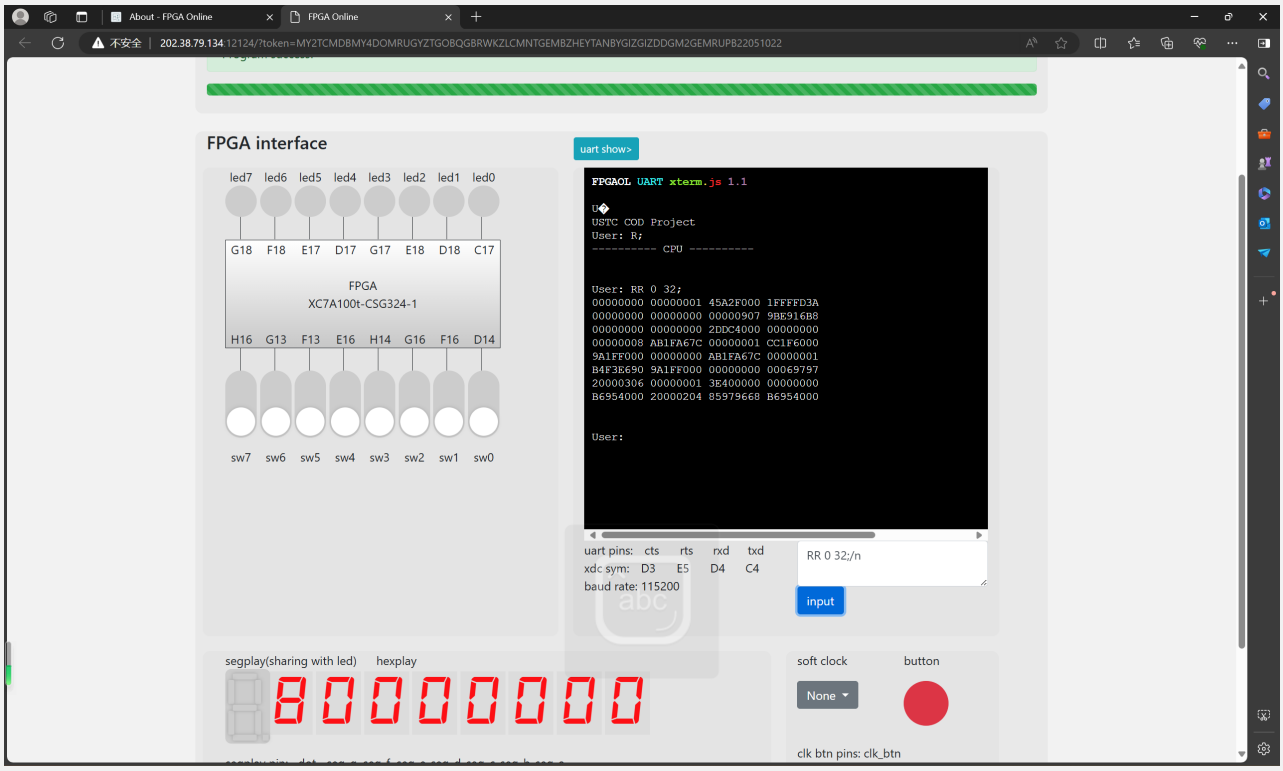
```
module Forwarding(···
    );
always @(*) begin
    if (rf_we_mem == 1 && rf_wa_mem != 0 && rf_we_wb == 1 && rf_wa_wb != 0) begin
        if (rf_wa_mem == rf_ra0_ex && rf_wa_wb == rf_ra1_ex) begin
            rf_rd0_fe = 1;
            rf_rd1_fe = 1;
            rf_rd0_fd = rf_wd_mem;
            rf_rd1_fd = rf_wd_wb;
        end
        else if (rf_wa_mem == rf_ra1_ex && rf_wa_wb == rf_ra0_ex) begin
            rf_rd0_fe = 1;
            rf_rd1_fe = 1;
            rf_rd0_fd = rf_wd_wb;
            rf_rd1_fd = rf_wd_mem;
        end
        else if (rf_wa_mem == rf_ra0_ex) begin
            rf_rd0_fe = 1;
            rf_rd1_fe = 0;
            rf_rd0_fd = rf_wd_mem;
            rf_rd1_fd = 0;
        end
        else if (rf_wa_mem == rf_ra1_ex) begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 1;
            rf_rd0_fd = 0;
            rf_rd1_fd = rf_wd_mem;
        end
        else if (rf_wa_wb == rf_ra0_ex) begin
            rf_rd0_fe = 1;
            rf_rd1_fe = 0;
            rf_rd0_fd = rf_wd_wb;
            rf_rd1_fd = 0;
        end
        else if (rf_wa_wb == rf_ra1_ex) begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 1;
            rf_rd0_fd = 0;
            rf_rd1_fd = rf_wd_wb;
        end
        else begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 0;
            rf_rd0_fd = 0;
            rf_rd1_fd = 0;
        end
    end
    else if (rf_we_mem == 1 && rf_wa_mem != 0) begin
        if (rf_wa_mem == rf_ra0_ex) begin
            rf_rd0_fe = 1;
```

```
            rf_rd1_fe = 0;
            rf_rd0_fd = rf_wd_mem;
            rf_rd1_fd = 0;
        end
        else if (rf_wa_mem == rf_ra1_ex) begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 1;
            rf_rd0_fd = 0;
            rf_rd1_fd = rf_wd_mem;
        end
        else begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 0;
            rf_rd0_fd = 0;
            rf_rd1_fd = 0;
        end
    end
    else if (rf_we_wb == 1 && rf_wa_wb != 0) begin
        if (rf_wa_wb == rf_ra0_ex) begin
            rf_rd0_fe = 1;
            rf_rd1_fe = 0;
            rf_rd0_fd = rf_wd_wb;
            rf_rd1_fd = 0;
        end
        else if (rf_wa_wb == rf_ra1_ex) begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 1;
            rf_rd0_fd = 0;
            rf_rd1_fd = rf_wd_wb;
        end
        else begin
            rf_rd0_fe = 0;
            rf_rd1_fe = 0;
            rf_rd0_fd = 0;
            rf_rd1_fd = 0;
        end
    end
    else begin
        rf_rd0_fe = 0;
        rf_rd1_fe = 0;
        rf_rd0_fd = 0;
        rf_rd1_fd = 0;
    end
end
endmodule
```

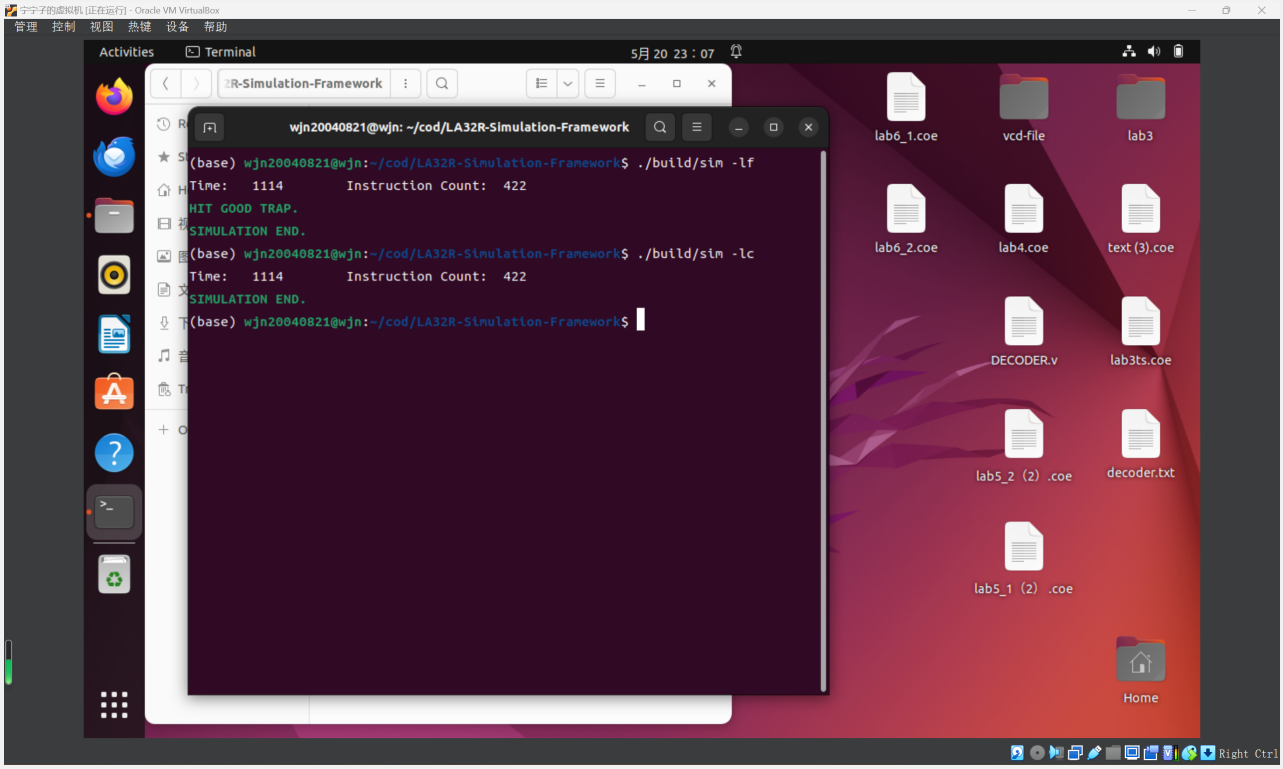
测试结果与分析

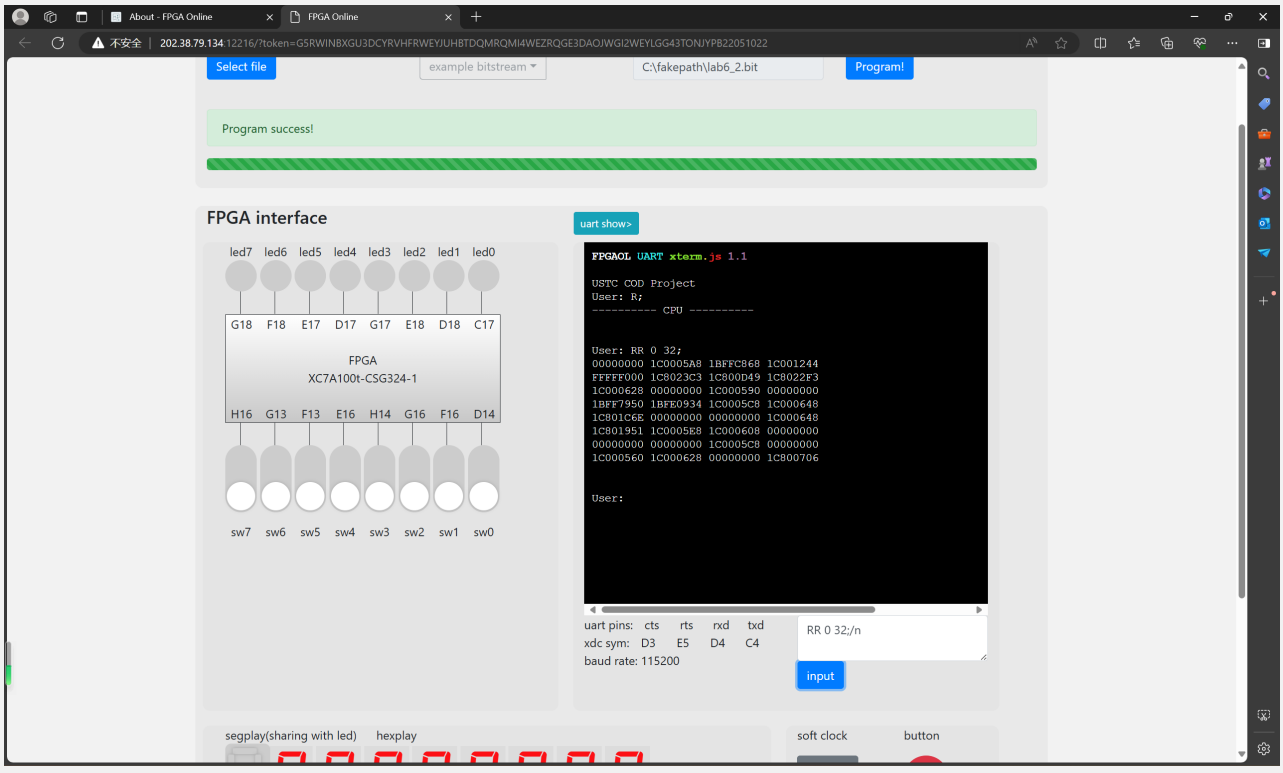
测试一


Activities
Terminal
5月 20 23：09
wjn20040821@wjn: ~/cod/LA32R-Simulation-Framework
(base) wjn20040821@wjn:~/cod/LA32R-Simulation-Framework$ ./build/sim -lf
Time: 884 Instruction Count: 437
HIT GOOD TRAP.
SIMULATION END.
(base) wjn20040821@wjn:~/cod/LA32R-Simulation-Framework$ ./build/sim -lc
Time: 884 Instruction Count: 437
SIMULATION END.
(base) wjn20040821@wjn:~/cod/LA32R-Simulation-Framework$
lab6_1.coe
vcd-file
lab3
lab6_2.coe
lab4.coe
text (3).coe
DECODER.v
lab3ts.coe
lab5_2 (2) .coe
decoder.txt
lab5_1 (2) .coe
Home


FPGA interface
uart show>
led7 led6 led5 led4 led3 led2 led1 led0
G18 F18 E17 D17 G17 E18 D18 C17
FPGA
XC7A100t-CSG324-1
H16 G13 F13 E16 H14 G16 F16 D14
sw7 sw6 sw5 sw4 sw3 sw2 sw1 sw0
FPGAOL UART xterm.js 1.1
USTC COD Project
User: R;
---------- CPU ----------
User: RR 0 32;
00000000 00000001 45A2F000 1FFFFD3A
00000000 00000000 00000907 9BE916B8
00000000 00000000 2DDC4000 00000000
00000008 AB1FA67C 00000001 CC1F6000
9A1FF000 00000000 AB1FA67C 00000001
B4F3E690 9A1FF000 00000000 00069797
20000306 00000001 3E400000 00000000
B6954000 20000204 85979668 B6954000
User:
uart pins: cts rts rxd txd
xdc sym: D3 E5 D4 C4
baud rate: 115200
RR 0 32;/n
input
segplay(sharing with led)
hexplay
soft clock
button
None
clk btn pins: clk_btn

测试二





总结

结合前递模块与段间寄存器控制模块，得到了一个能正常运行的完整流水线 CPU。