

EE312 Lab5 Report

20180155 김준범

20160791 권용빈

I. Introduction

Lab5는 Lab3에서 공부하였던 RISC-V의 uArchitecture의 instruction들을 수행할 수 있는 pipelined cpu를 구현하는 과제였다. CPU를 구성하는 Register, Memory, Clock, ALU, PC, Control signal, forwarding unit, register에 대한 상호작용을 이해해야 하며 각 instruction마다 필요로 하는 hazard detection, ALU/PC 연산에 대한 부분을 implement해야 했다.

II. Design

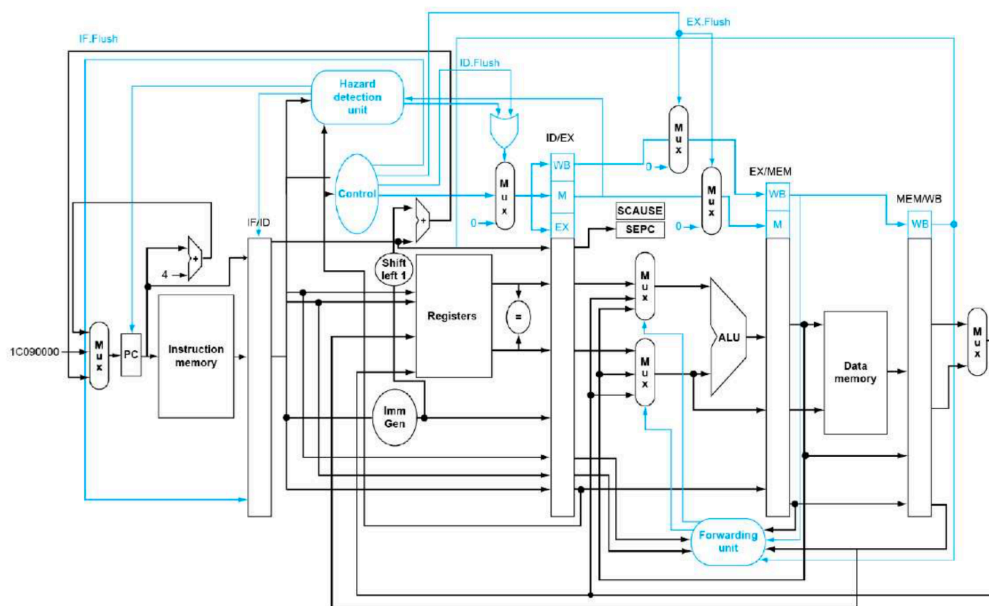


Figure 1. Pipeline CPU

위 회로와 유사하게 RISC_V_TOP module에 IF/ID_REG, ID/EXE_REG, EXE/MEM_REG, MEM_WB/REG, CTRL, FU(forwarding unit) 모듈을 연결시켜 pipelined cpu를 구현하였다.

III. Implementation

매 clk마다 받는 instruction을 한 clock 한에 모든 5 stage(IF-ID-EXE-MEM-WB)를 동시에 실행시켜주는 기능을 한다. IF/ID_Reg->ID/EXE_REG->EXE/MEM_REG->MEM/WB_REG의 관계로 동시다발적으로 update를 시켜주고 CTRL module에서 생성된 control signal들을 필요에 따라 passing 해주었다. 또한 RAW dependency를 해소해주기 위해 forwarding 조건을 생성해주는 FU module을 만들어 dependency를 보유하고있는 data들을 forwarding 해주었다. 또한 hazard가 detect될 경우 nop이라는 signal을 assert시켜 해당 REG module의 update를 멈추어 stall의 기능을 구현하였다.

IV. Evaluation

```
Test # 1 has been passed
Test # 2 has been passed
Test # 3 has been passed
Test # 4 has been passed
Test # 5 has been passed
Test # 6 has been passed
Test # 7 has been passed
Test # 8 has been passed
Test # 9 has been passed
Test # 10 has been passed
Test # 11 has been passed
Test # 12 has been passed
Test # 13 has been passed
Test # 14 has been passed
Test # 15 has been passed
Test # 16 has been passed
Test # 17 has been passed
Test # 18 has been passed
Test # 19 has been passed
Test # 20 has been passed
Finish: 24 cycle
Success.
```

```
Test # 1 has been passed.
Test # 2 has been passed.
Test # 3 has been passed.
Test # 4 has been passed.
Test # 5 has been passed.
Test # 6 has been passed.
Test # 7 has been passed.
Test # 8 has been passed.
Test # 9 has been passed.
Test # 10 has been passed.
Test # 11 has been passed.
Test # 12 has been passed.
Test # 13 has been passed.
Test # 14 has been passed.
Test # 15 has been passed.
Test # 16 has been passed.
Test # 17 has been passed.
Finish: 90 cycle
Success.
```

```
Test # 1 has been passed.
Test # 2 has been passed.
Test # 3 has been passed.
Test # 4 has been passed.
Test # 5 has been passed.
Test # 6 has been passed.
Test # 7 has been passed.
Test # 8 has been passed.
Test # 9 has been passed.
Test # 10 has been passed.
Test # 11 has been passed.
Test # 12 has been passed.
Test # 13 has been passed.
Test # 14 has been passed.
Test # 15 has been passed.
Test # 16 has been passed.
Test # 17 has been passed.
Test # 18 has been passed.
Test # 19 has been failed!
output_port = 0x17c (Ans : 0xe74)
```

위의 각 test bench 실행 결과의 스크린샷에서 볼 수 있듯이 inst와 forloop tesbench의 testcase들은 모두 통과하였고 sort test bench는 test case 18까지 밖에 통과하지 못하였다.

Inst: 24 cycle

Forloop: 90 cycle

V. Discussion

코드의 구조가 다소 복잡하여 예상보다 많은 hazard condition 들이 발생하였고 이들을 일반화시켜 해소시켜지 못하였다.

VI. Conclusion

수업시간에 배우고 책에 나오는 hazard condition을 제외한 모든 case들을 찾지 못하고 BTB를 구현하지 못한점이 너무나도 아쉽다.

