1. 20202023002 박용준

① load/store의 경우

- PC 값을 통해서 Instruction을 fetch하여 그에 맞는 번호의 register을 읽는다. 읽은 register의 data를 ALU에 넘겨주는데 만약 immediate를 사용하면 sign-extend하여 ALU에 넘겨준다. ALU는 ALU Operation signal을 통해서 ALU의 연산이 결정된다.

이때 ALU는 address를 계산한다. load면 memory를 읽고 register에 값을 업데이트한다. store도 마찬가지로 ALU를 거쳐 memory에 register 값을 저장한다. 이런 과정 중에서 ALU 연산 과정과 같은 시점에 다음 PC 값도 계산이 되고, 그 값이 PC에 저장된다.

② ALU 연산의 경우

- PC 값을 통해서 Instruction을 fetch하여 그에 맞는 번호의 register을 읽고, 그 값을 ALU에 넣어 ALU Operation signal에 따라 연산을 한다. 그리고 그 결과를 register에 저장한다. (ALU에는 immediate 값이 들어갈 수 있다. 그래서 immediate 값을 넣으려면 sign-extend를 해야한다.

③ branch의 경우

- PC 값을 통해서 Instruction을 fetch하여 그에 맞는 번호의 register을 읽는다. ALU를 통해서 0 인지를 체크한다. 그래서 PC+4로 갈 것인지, 다른어떤 target address 인 (PC+4 + ?x4)로 이동할 것인지 결정하여 PC에 그 값을 저장한다.

2.

- k stage pipeline의 경우, 첫 instruction은 k cycle의 끝에 도달한다. 이후, 나머지 n-1의 instruction들은 n-1 cycle동안 한 사이클에 한 instruction이 실행되므로 n개의 instructions를 k stage pipeline에서 실행하려면 k+n-1 cycles가 필요하다.

3.

```
addi $s0, $s1, 5
NOP
NOP
add $s2, $s0, $s1
addi $s3, $s0, 15
NOP
addi $s4, $s2, $s1
```

4.

① EX/MEM stage에 있는 Rd를 위한 register number값이 ID/EX stage에 있는 Rs or Rt를 위한 register number 값과 같을 때

② MEM/WB stage에 있는 Rd를 위한 register number 값이 ID/EX stage에 있는 Rs or Rt를 위한 register number 값과 같을 때

5.

다음 장에 그렸습니다.

6.

④ ID/EX stage에서 Memory Read를 하고, ID/EX stage에 있는 Rt를 위한 Register number값이 IF/ID stage에 있는 Rs or Rt를 위한 Register number 값과 같을 때

5. 20202003002 박상훈