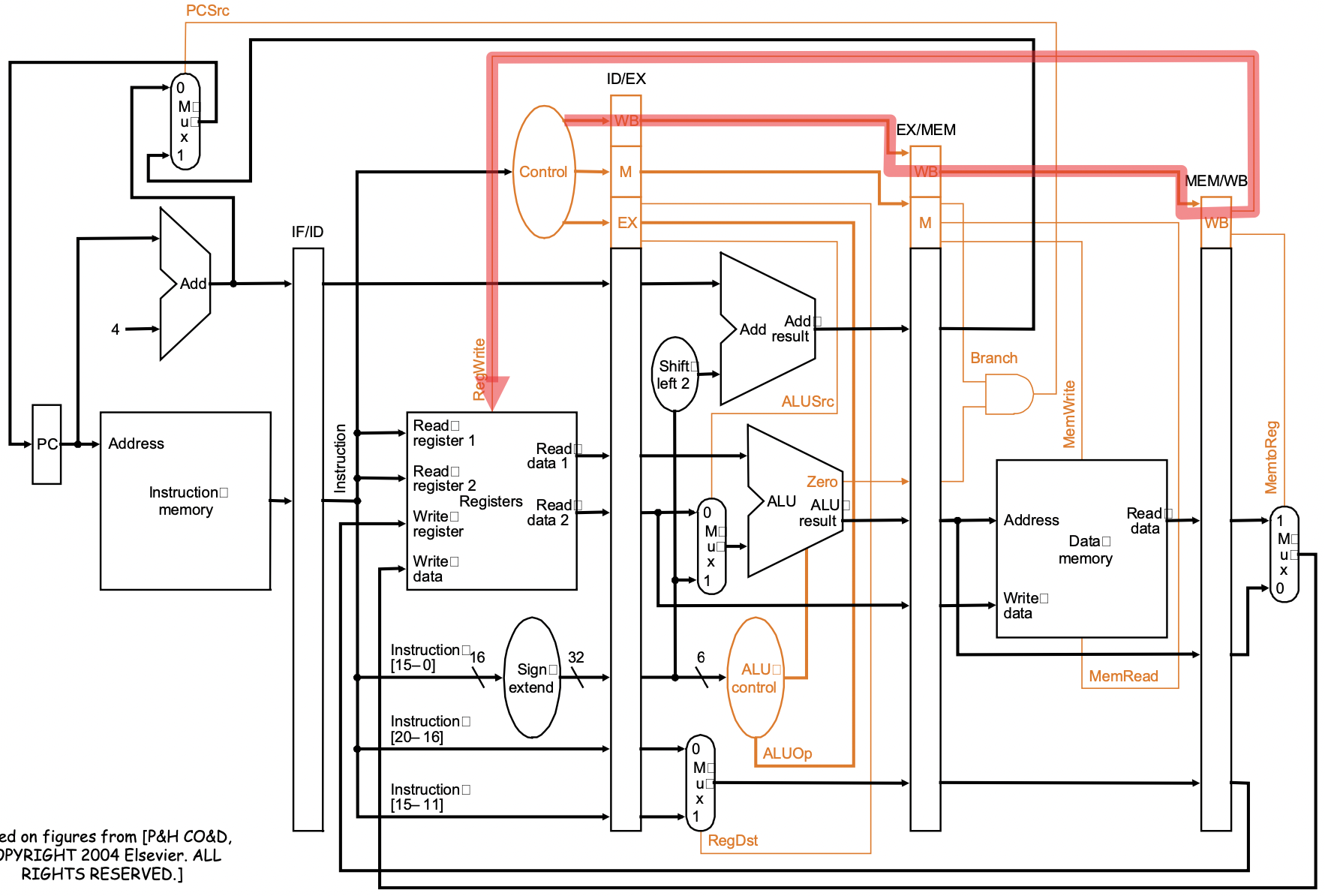
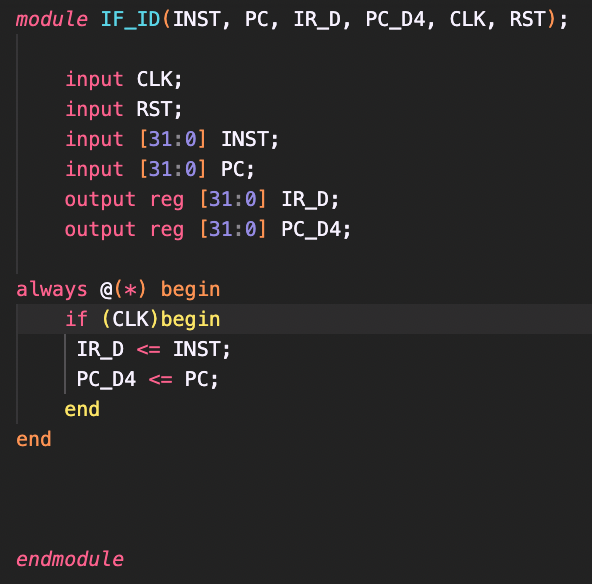
Assignment 6

201820873 윤현찬

이번 설계에서는 Pipelined Cpu를 설계하였다. Pipelined Cpu란 Single Cpu에서 Latch를 활용하여 Cpu의 성능을 향상시킨 Cpu이다. 방법으로는 Cpu의 동작은 5가지로 나뉘는데 이 5가지 동작의 결과를 Latch에 저장하여 명령어 하나의 연산이 끝나기 전에 다음 연산을 하도록 한다.

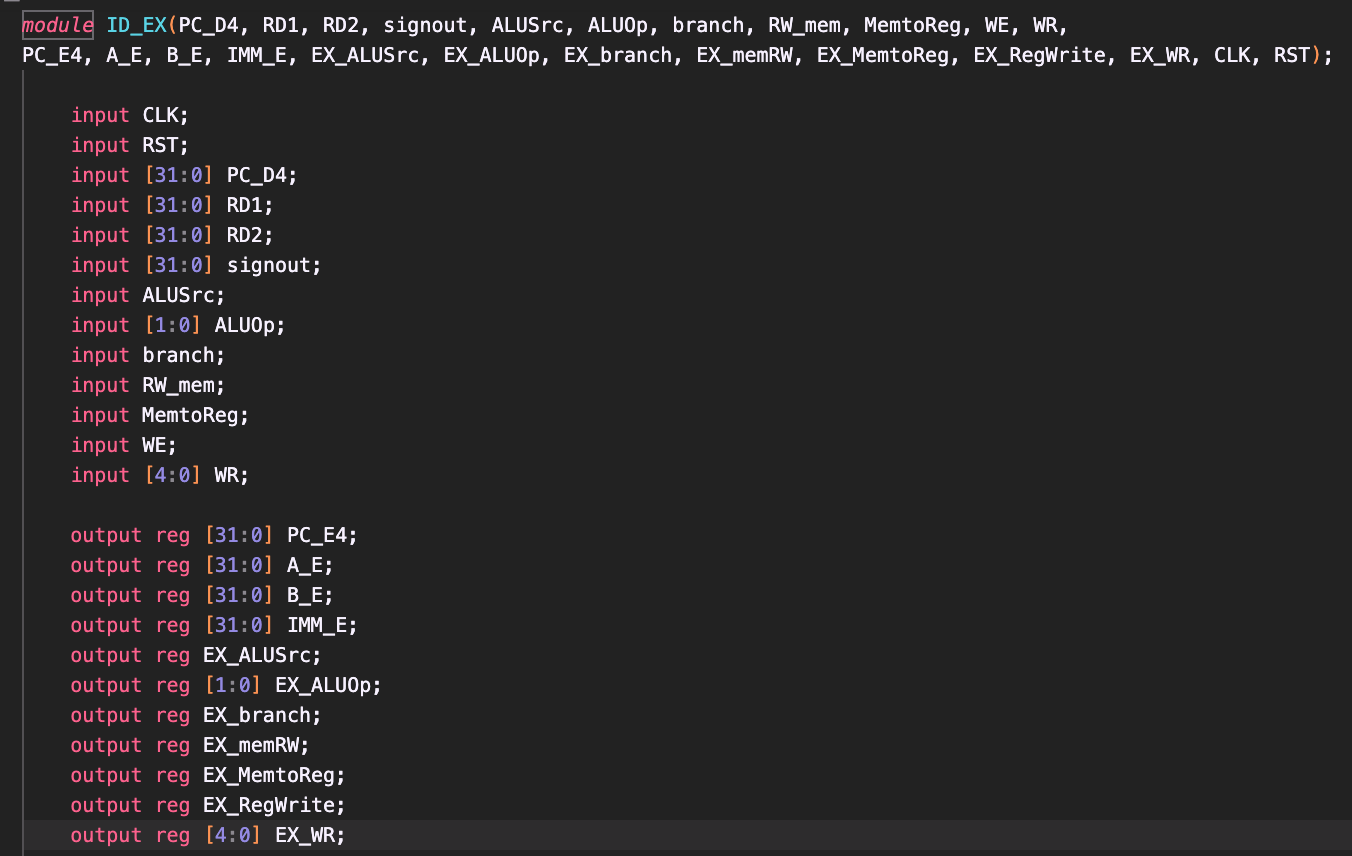


1. IF-ID Latch



첫번째 Latch에서는 명령어 Register의 출력과 PC + 4의 연산을 저장하였다. Latch는 D를 입력으로 하였고, G는 CLK로 설정하였다.

2. ID/EX



두번째 Latch에서는 ID의 출력을 저장하는 Latch이다. PC+4연산은 Control Module에서 명령어를 해석하고, 명령어의 Immediate를 해석 후 Mux를 이용하여 다음 PC를 연산한다. 따라서 PC의 연산은 ID에서 행해지지 않으므로 다음 Latch로 전달해야하므로 Latch에 저장한다. 또한 받은 명령어를 해석하여 Register와 Control module에 전달한 후 출력을 저장한다. 또한 명령어의 Opcode를 해석하여 Control Module에서 다음 동작하는 Module에 Signal을 전달하므로 Signal들을 Latch에 저장하였다. 또한 Register에 값을 저장할 때 저장할 값은 WB 동작 이후에 결정되므로 값을 저장할 Register의 주소를 바로 Register에 입력하지 않고 Latch에 저장하여 순서에 맞게 동작하게 한다.

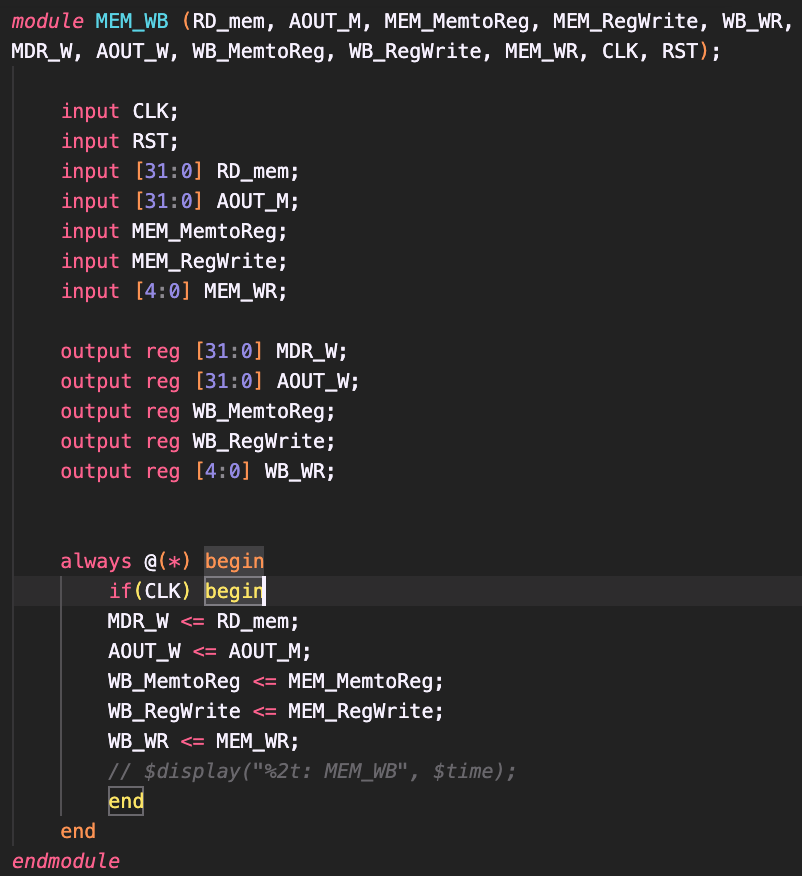
3. EX/MEM

텍스트이(가) 표시된 사진

자동 생성된 설명

EX는 Execute의 줄임말로 연산 동작을 한다. Immediate 값을 PC\_ALU와 CLU에 전달하고, Register의 출력을 ALU에 전달 후 연산을 한다. 따라서 연산의 결과를 저장한다. Control Module에서 나온 Signal들을 EX에서 필요한 Signal들은 이용하고 다음 동작에서 활용하는 Signal들은 다시 Latch에 저장한다.

4. MEM/WB



WB는 Register에 저장할 값을 제어하는 동작이다. 따라서 Immediate 혹은 ALU의 연산결과를 Latch에 저장하고 둘 중 하나를 선택하는 Mux의 SEL 변수인 MEM\_memtoreg 변수를 저장하고 명령어를 해석 후 얻은 값을 저장할 Register의 주소를 저장한다.