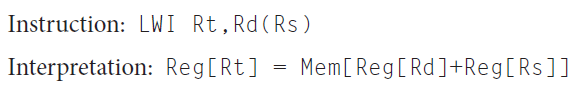
Chapter 4 hw

2017069598 박상지

4.2



4.2.1

해당 instruction은 target address가 없기 때문에 PC + 4, instruction memory을 사용한다.

해당 instruction을 메모리에 접근하기 때문에 Memory unit을 사용하고, 레지스터를 사용하고 읽고 쓰고 ALU 연산을 사용하기 때문에 Register와 ALU unit을 사용한다.

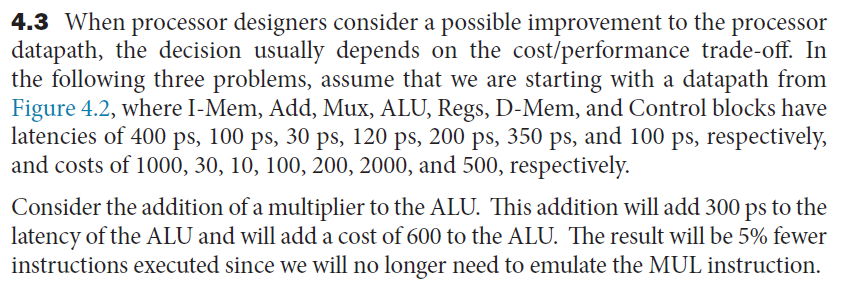
4.2.2

이미 있는 unit으로 충분히 instruction을 수행할 수 있기 때문에 새로운 functional block을 만들 필요가 없다.

4.2.3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | RegDst | ALUSrc | MemtoReg | RegWrite | MemRead | MemWrite | Branch |
| LWI | X | 0 | 1 | 1 | 1 | 0 | 0 |

이미 있는 control signal로 충분히 instruction을 실행할 수 있기 때문에 새로운 signal이 필요 없다.



4.3.1

ALU 연산에서는 instruction memory에서 data memory까지 총 두 개의 MUX를 지나기 때문에 1330ps의 clock cycle time이 걸린다. improvement까지 따지면 300ps를 더하기 때문에 총 1630ps의 clock cycle time이 걸린다.

4.3.2

95%의 starting instruction이 필요하기 때문에 전체 cct에 0.95를 곱하면 1548.5ps를 갖는다. without improvement와 비교하면 0.86이 나온다. 새로운 프로세서는 전 프로세서보다 14% 느려 진다.

4.3.3

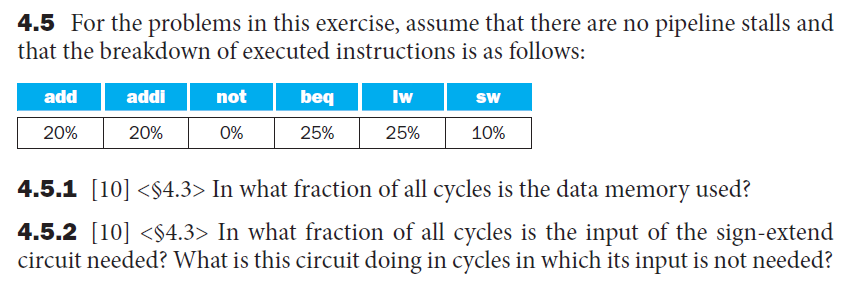
비용을 비교하면 원래 프로세서에서는 3890이 나온다. 새로운 프로세서에서는 600을 더하라 했으니 4490 비용이 나온다.

원래 프로세서 비용 = 0.866 \* 새로운 프로세서 비용이 된다.

원래 프로세서 퍼포먼스 = 1.16 \* 새로운 프로세서 퍼포먼스

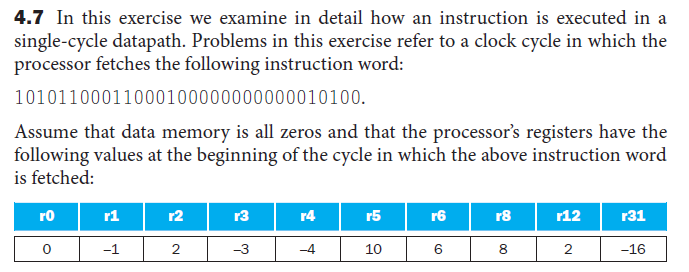
원래 프로세서 비용/원래 프로세서 퍼포먼스 = 0.75 \* 새로운 프로세서 비용 / 새로운 프로세서 퍼포먼스

새로운 프로세서가 15% 작다.



4.5.1 data memory는 load와 store에서 사용한다. 그러므로 전체에서 35% clock cycle을 사용한다.

4.5.2 sign-extend는 immediate 값을 사용하는 addi, load, store instruction과 beq instruction에서 사용된다. 그러므로 전체에서 80% clock cycle이 사용된다.



4.7.1

shift left by 2 unit을 하면 ~ 0001100010000000000001010000 이 된다.

4.7.2

해당 opcode는 101011이기 때문에 43이다. 이것은 store instruction을 가리키며 해당 ALUop = 00이다 즉 signal on ALU은 0010이 된다.

4.7.3

PC + 4

4.7.4

data memory mux의 결과값이 정해지지 않았고 ALU mux는 0000 0000 0000 0000 0000 0000 0001 0100이고 Branch mux는 PC + 4이다.

4.7.5

ALU는 -3 과 20이며 첫번째 add에서 pc+4가 일어나고 두번째에서는 branch가 일어나므로 pc+4그리고 16 \* 5

4.7.6

read register 1 , read register 2 , write register, write data, RegWrite