



과목명	컴퓨터구조
담당교수	조경산 교수님
학과	소프트웨어학과
학번	32153180
이름	이상민
제출일자	2018.12.7

1. Instruction size : 3bytes, byte addressable, Integer : 8bit two's complement representation. R1 = 70, R2 = 80, M[70] = -70, M[71] = -71, M[72] = -72

300 : MOV R2, #70 ; R2 <= 70

303 : ADD R2, 1@R1 ; R2 <= R2 + M[R1+1]

306 : JP \$-9 ; branch if positive (branch to -9@PC)

1) After executing MOV and ADD instructions, what value is stored in register R2?

-> MOV R2, #70을 하면 R2에 70이라는 값이 들어간다. ADD R2, 1@R1을 하면 R2에 있는 값 70과 1@R1 즉, M[R1+1] 값을 더해 R2에 저장한다.

R1 = 70이기 때문에 M[R1+1] = M[71] = -71이고, 두 수를 ADD하면 -1이 R2에 저장된다.

2) Show the values of flags(C, Z, S, OV) after ADD instruction

-> 70은 2진수로 01000110이고, -71은 2진수로 10111001이다. 두 수를 더하면 11111111로 2의 보수를 해서 확인하면 -1이라는 것을 알 수 있다.

carry나 overflow가 발생하지 않았고, 연산 결과가 0이 아니므로 C : 0, Z : 0, OV : 0이고, 결과가 음수이기 때문에 S : 1이다.

3) What is the address of the instruction executed after JP instruction?

-> JP instruction은 양수인 경우에 branch하라고 되어있다. ADD의 결과가 양수가 아니기 때문에 다음 실행할 instruction address는 순차적으로 봤을 때 309이다.

4) Considering above program, design instruction format with 14 different

operations and 15 registers. Determine instruction format, size of OP-code, number and size of operands, etc.

-> instruction size가 3bytes = 24bits이기 때문에 ARM processor가 아니다. 따라서 CISC라는 것을 알 수 있다. register가 15개이기 때문에 register를 나타내는 bit는 4bit이다. operation도 14개이므로 4bit이다. 위의 instruction에서 addressing mode는 각각 순서대로 immediate, register indirect, displacement 이렇게 3개를 사용하기 때문에 2bit이다. instruction은 opcode와 operand로 구성되어 있는데, 위의 instruction들은 operand가 2개까지 있다. 따라서 opcode + operand1 + operand2로 구성된 instruction이다. operation 개수를 통하여 opcode가 4bit라는 것을 알 수 있고, operand는 register 4bit와 addressing mode 2bit의 합, 6bit이다.

instruction format : opcode(4bit) + operand1(6bit) + operand2(6bit) = 16bit

2. A processor does not provide indirect addressing. Assume that the address of and operand is in main memory. How would you access the operand?

-> register indirect addressing과 displacement addressing이 있다.

register indirect addressing : register에 있는 내용이 memory address인 경우를 뜻한다. register addressing의 단점인 제한된 주소 공간을 해결해주고, indirect addressing보다 메모리 참조를 한 번 더 적게 한다.

displacement addressing : direct addressing과 register indirect addressing을 합친 addressing mode이다. instruction은 2개의 address field를 갖는데 하나는 직접 사용하고, 다른 하나는 register를 참조한다.

3. A pipelined processor has a clock rate of 1GHz and executes a program with one million instructions. The pipeline has four stages and instructions are issued at a rate of one per clock

a) What is the speedup of this processor for this program compared to a non-pipelined processor

-> non-pipelined processor에서 수행하면  $4 \times 10^6 \times (1/10^9)$ 만큼 걸린다.

pipelined processor에서 수행하면  $(4 \times 10^6 - 1) \times (1/10^9)$ 만큼 걸린다.

speedup =  $4 \times 10^6 \times (1/10^9) / (4 \times 10^6 - 1) \times (1/10^9) = 4 \times 10^6 / 3 \times 10^6$ 이므로 약 4이다.

b) What is the MIPS rate for the pipelined and non-pipelined processor

-> MIPS =  $I_c / T \times 10^6$

non-pipelined processor :  $10^6 / 4 \times 10^6 \times (1/10^9) \times 10^6 = 10^3/4 = 262.5$

pipelined processor :  $10^6 / (4 \times 10^6 - 1) \times (1/10^9) \times 10^6 = 10^9/(3 \times 10^6)$ 이므로 약 1000이다.

c) List three pipeline hazards and briefly explain how these hazards affect the performance

-> pipeline hazard는 크게 resource hazard, data hazard, control hazard로 나뉜다.

① resource hazard : 자원이 부족하다는 것을 뜻한다. 예를 들어 instruction fetch와 operand fetch는 동시 수행 시 delay가 발생하는데 둘 다 메모리에 접근해야하기 때문이다. 이러한 resource hazard의 해결 방법은 L1 cache와 같은 split cache를 사용하면 된다.

② data hazard : data 사이 dependency가 있을 때 발생한다. 해결 방법은 컴파일러가 수행 순서를 바꿔주면 된다.

③ control hazard : 수행 순서가 순차적이지 않을 때 발생한다. 여러 해결 방법들이 있는데 가장 많이 사용하는 것은 branch prediction 즉, 분기가 일어날 것인지를 예측하는 것이다.

4. Explain the typical distinguishing characteristics common to RISC organization

-> instruction size가 32bit로 고정되어 있고 상당히 많은 register를 갖고 있다. 그래서 load/store 경우에만 메모리에 접근하고 그 외 모든 경우에는 register에서 데이터를 가져온다. 또한 conditional instruction을 사용해 conditional branch instruction을 최소화시켜 궁극적으로 pipeline 효율을 증가시킨다.

5. What is the difference between the superscalar and superpipelined approaches?

-> superpipelined : pipeline 단계를 더 나눈 것으로 clock cycle 당 2개의 pipeline 단계를 수행할 수 있다. simple pipeline보다 성능이 배가 되지만 hazard가 많이 발생한다.

superscalar : 서로 다른 pipeline에서 독립적으로, 동시에 instruction을 수행하는 것을 뜻한다. superscalar의 궁극적인 목적은 CPI를 1보다 작게, IPC를 1보다 크게 하는 것이다.

6. Explain the purpose and contents of the exception and interrupt vector table

-> interrupt를 굳이 두 종류로 나눈다면 그냥 interrupt와 exception으로 나눌 수 있다. interrupt는 hardware에 의해 발생하는 것을 뜻하고, exception은 software, program에 의해 발생하는 것을 뜻한다.

모든 interrupt는 지정된 번호가 있다. 이 번호가 interrupt vector table에서 index로 사용 되는데, IVT는 이 번호에 대한 ISR(interrupt service routine)의 주소인 256개의 32bit interrupt vector를 포함한다.

7. Show the size of each storage to store

1) character '1'

-> char : 1byte

2) string "1"

-> char : 1byte

3) integer 1

-> int : 4byte

4) real number(single precision) 1.0

-> float : 4byte