

Midterm Examination #1 - Solution

시험시간: 10월 5일(화) 오후 7:00-9:00

주의사항:

- 답안지에 이름, 학번을 반드시 기입, 답안지는 앞면, 뒷면을 모두 사용할 것
- 답이 도출되는 과정을 명확하고 알아보기 쉽게 적을 것
- 문제에 애매한 부분이 있다고 판단되는 경우 합리적인 가정을 한 후 문제를 풀어도 무방함
- 문제들마다 난이도가 다르므로 먼저 전체를 한번 훑어 본 후 시간을 잘 조절하여 문제를 풀 것

1. (15 points) 서로 다른 instruction set을 갖는 2대의 컴퓨터 A, B가 있다. 컴퓨터 A는 1.3의 평균 CPI를 갖고 600MHz의 clock rate으로 동작한다. 컴퓨터 B는 2.5의 평균 CPI를 가지며 750MHz의 clock rate으로 동작한다. 어떤 프로그램을 컴퓨터 A에서 실행하기 위하여 컴파일 했을 때 100000개의 명령어로 구성된다고 하자. 이 프로그램이 컴퓨터 B에서 실행될 때, 컴퓨터 A에서 실행될 때와 동일한 실행 시간을 갖기 위해서는 컴퓨터 B의 instruction set으로 몇 개의 명령어들로 구성되어야 하는가?

$$\text{CPUTime}_A = (\text{Instruction count})_A * (\text{CPI})_A * (\text{Clock cycle Time})_A = (100,000) * (1.3) / (600 * 10^6) \text{ ns}$$

$$\text{CPUTime}_B = (\text{Instruction count})_B * (\text{CPI})_B * (\text{Clock cycle Time})_B = I_B * (2.5) / (750 * 10^6) \text{ ns}$$

Since $(\text{CPUTime})_A = (\text{CPUTime})_B$, we have to solve for I_B and get 65000

2. (10 points) 어떤 새로운 컴퓨터에서 곱셈명령어의 수행성능이 기존의 시스템보다 15배가 빨라졌다고 가정해 보자. 여기에서 어떤 프로그램을 실행할 경우 프로그램의 전체 실행 성능이 4배가 향상되기 위해서는 이 프로그램에서 곱셈명령어가 차지하는 비율이 얼마나 되어야 하는가?

We will use Amdahl's Law again for this question.

Let x be percentage of floating point instructions. Since the speedup is 4, if the original program executed in 100 cycles, the new program runs in $100/4 = 25$ cycles.

$$(100)/4 = (x)/15 + (100 - x)$$

Solving for x, we get:

$$x = 80.36$$

The percent of floating point instructions need to be 80.36.

3. (10 points) 아래의 MIPS 어셈블리 코드에 대해 다음의 질문에 대답하십시오.
- A. 명령어들을 읽기 위해서는 몇 번의 메모리 접근이 필요한가?
 - B. 명령어가 아닌 데이터들을 접근하기 위해서는 몇 번의 메모리 접근이 필요한가? (레지스터 접근이 아닌 메모리 접근 횟수만 카운트 할 것)

```
lw      $v1, 0($a0)
addi    $v0, $v0, 1
sw      $v1, 0($a1)
addi    $a0, $a0, 1
```

The instruction memory is accessed four times (as there are four instructions) and the data memory is accessed twice (once for the lw instruction and another time for the sw instruction).

4. (15 points) 아래의 레지스터 및 메모리 값들을 이용하여 다음의 질문들에 답하십시오. 32-bit 컴퓨터이고, 각 문제를 풀 때는 레지스터와 메모리들이 아래의 테이블의 값들로 채워져 있는 상태에서 시작한다고 가정한다.

| Register | Value | Memory Location | Value |
|----------|-------|-----------------|-------|
| R1 | 12 | 12 | 16 |
| R2 | 16 | 16 | 20 |
| R3 | 20 | 20 | 24 |
| R4 | 24 | 24 | 28 |

- a) 명령어 "add R3, R2, R1"을 수행한 후의 R1, R2, R3의 값은 어떻게 되는가?

After add R3, R2, R1:
R1 = 12, R2 = 16 and R3 = 28

- b) 명령어 "lw R3, 12(R1)"을 수행한 후 R1과 R3의 값은 어떻게 달라지는가? 위의 테이블에서 memory location은 word 하나를 의미한다고 가정하십시오.

After load R3, 12(R1):
R3 = 28 and R1 = 12

- c) 명령어 "addi R2, R3, #16"을 수행한 후 R2, R3의 값이 어떻게 달라지는가?

After addi R2, R3, 16:
R2 = 36 and R3 = 20

5. (20 points) 아래의 C 코드를 MIPS 어셈블리 코드로 변환하여 작성하십시오.

```
int A[100], B[100];
for (i=1; i < 100; i++) {
    A[i] = A[i-1] + B[i];
}
```

코드의 시작에서 \$a0와 \$a1 레지스터들은 배열 A와 B의 시작주소를 담고 있다. 곱셈명령어를 사용하지 말 것.

The MIPS assembly sequence is as follows:

```
li $t0, 1          # Starting index of i
li $t5, 100        # Loop bound

loop:
lw $t1, 0($a1)     # Load A[i-1]
lw $t2, 4($a2)     # Load B[i]
add $t3, $t1, $t2  # A[i-1] + B[i]
sw $t3, 4($a1)     # A[i] = A[i-1] + B[i]
addi $a1, 4        # Go to i+1
addi $a2, 4        # Go to i+1
addi $t0, 1        # Increment index variable
bne $t0, $t5, loop # Compare with Loop Bound

halt:
nop
```

6. (30 points) 새로운 MIPS 명령어 bcp가 있다고 가정해 보자. 이 명령어는 메모리 주소상에서 연속적

으로 저장된 여러 개의 32-bit word들을 어떤 곳(source block)에서 다른 곳(destination block)으로 복사하기 위해 만들어졌다. 이를 block copy라고 한다. 이 명령어를 사용하기 위해서는 source block의 시작주소인 source address가 \$t1 레지스터에 저장되고 destination block의 시작주소 destination address는 \$t2 레지스터에 저장되어야 한다. 그리고 복사할 word의 개수는 \$t3 레지스터에 저장된다. 또한 \$t4 레지스터를 이 명령어를 수행하기 위해 임시로 사용할 수 있다고 가정해보자.

- a) Block copy를 수행하기 위한 MIPS 어셈블리 코드를 bcp 명령어를 사용하지 않고 작성해 보시오. (즉 기존의 MIPS 명령어들만을 이용해서 block copy를 수행하도록)

The MIPS code to implement block copy without the bcp instruction is as follows:

loop:

```
lw    $t4, 0($t1)
sw    $t4, 0($t2)
addi  $t1, $t1, 4
addi  $t2, $t2, 4
subi  $t3, $t3, 1
bne   $t3, $zero, loop
```

- b) bcp 명령어를 이용해서 block copy를 수행하는 MIPS 어셈블리 코드를 작성하십시오. Source block의 시작주소는 \$a0, destination block의 시작 주소는 \$a1, 복사할 word의 개수는 \$a2에 저장되어 있다고 가정한다.

To implement block copy with this instruction:

```
add    $t1, $a0, $zero
add    $t2, $a1, $zero
add    $t3, $a2, $zero
bcp
```

- c) 100 word를 전송할 때 각 구현방법에서 걸리는 시간은 얼마인가? bcp를 제외한 모든 명령어들은 1 사이클의 수행시간을 갖지만, bcp 명령어는 전송할 word 개수만큼의 실행시간이 필요하다고 가정한다.

Assuming each instruction in the MIPS code of 'loop' takes 1 cycle, for doing a 100-word copy the total number of cycles taken is $6 \times 100 = 600$ cycles.