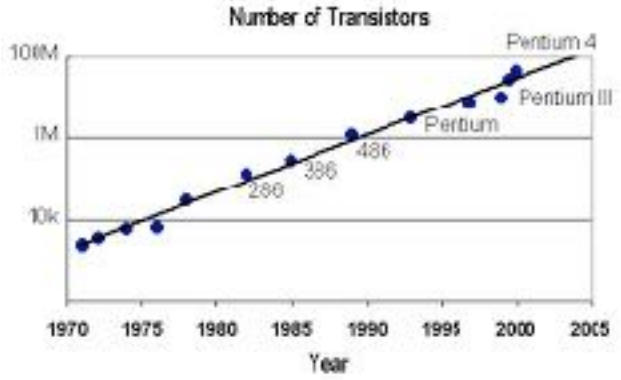
컴퓨터 구조 중간고사 (2018/10/16) 학번: 이름:

*모든 문제에 틀린 경우 감점이 있음.

1. 다음의 그림이 나타내는 현상을 '무슨 법칙' 이라고 하는지 쓰고, 그 법칙의 내용을 한 문장으로 쓰시오.



2. Interface between hardware and low-level software 를 지칭하는 말은?

3. Interface between hardware and low-level software에서 정의되지 않는 것을 모두 골라라.

a. 어떤 명령어가 있는가
b. 명령어를 어떻게 표현하는가
c. 명령어의 동작 결과
d. 명령어를 하드웨어로 구현하는 방법

4. Register 란 무엇인가? (다른 종류와 구별되는 4가지 특징을 포함할 것)

5. 다음 중 CISC 와 비교되는 RISC architecture 의 특징이 아닌 것을 모두 골라라.

a. 명령어의 종류가 다양하다.
b. 명령어의 길이가 일정하다.
c. 산술 논리 연산 명령어의 operand 로 메모리 주소를 쓸 수 있다.
d. high-level programming language 의 명령문들이 기계어 명령어로 1:1 매핑(mapping) 이 된다.

6. -4를 2의 보수로 표현되는 32-bit 정수로 나타낼 때 이진수로 표현하고 이것을 다시 16진수로 표현하라.

MIPS opcode 표

0x00 add (function field : 0x20)
0x00 sub (function field : 0x22)
0x00 slt (function field : 0x2a)
0x00 jr (function field : 0x08)
0x02 j
0x03 jal
0x04 beq
0x05 bne
0x08 addi
0x0a slti
0x23 lw
0x2b sw

7. 이진수 0010 1000 1010 1000 0000 0000 0000 0100 를 MIPS assembly instruction 으로 해석해라.

8. MIPS 명령어 sub $5, $4, $3 를 기계어(2진수)로 표현하라.

9. MIPS 명령어 addi $3, $4, -1 를 기계어(2진수)로 표현하라.

10. 다음과 같은 프로그램을 수행했을 때 $t0 의 값이 0x11 이 되었다면 이 프로세서의 byte order 는 little endian 인가? big endian 인가?

```
.data 0x10000004
.word  0x11223344

.text
main: lui $s0, 0x1000
        lb  $t0, 4($t0)
```

11. 다음의 C 코드를 MIPS assembly codes 로 표현하라.

```
void bar(){
   foo();
}
void foo(){
  return;
}
```

12. 유사명령어(pseudoinstruction) 가 아닌 div 명령어는 2개의 operand 를 가진다. 그런데 유사명령어로 3개의 operands 를 가지는 div 명령어가 있다. div $10, $4, $5 를 실행하면 $4의 값을 $5의 값으로 나눈 몫이 $10에 저장된다. 이 명령어를 assemble 할 때 실제 MIPS 명령어들로 구현하려면 (divide by 0 인 경우가 없다고 가정하고) 두 개의 명령어를 연속으로 사용하여 구현할 수 있다. 유사 명령어 div $10, $4, $5 를 (유사 명령어가 아닌) 두 개의 MIPS assembly 명령어로 표현하라.

13. 다음의 MIPS 프로그램을 수행할 때, 각 명령어에 주어진 (a) ~ (q) 기호를 사용하여 답하시오. $sp 의 초기값은 0x7fff0010 이다. Program Counter 의 값은 명령어를 수행할 때 이미 4만큼 증가되어 있다고 가정하라.

```
.data 0x10004000
.word  2
.word  1
.text 0x00400080 # 다음의 코드가 메모리 주소
                 # 0x00400080부터 로드된다는 뜻임
main :  addi $sp, $sp, -4        # (a)
        sw   $ra, 0($sp)         # (b)
        lui   $t0, 0x1000        # (c)
        ori   $t0, $t0, 0x4000   # (d)
        lw    $v0, 0($t0)        # (e)
L:      add   $a0, $v0, $0       # (f)
        jal    f1                # (g)
        bne   $v0, $0, L         # (h)
        lw    $ra, 0($sp)        # (i)
end:    addi $sp, $sp, 4         # (j)
        jr     $ra               # (k)
f1:     addi $sp, $sp, -4        # (l)
        sw   $s0, 0($sp)         #(m)
        lw    $s0, 4($t0)        # (n)
        sub  $v0, $a0, $s0       # (o)
        lw    $s0, 0($sp)        # (p)
        j      end               #(q)
```

메모리에 저장된 이 프로그램의 코드는 몇 개의 MIPS 명령어로 구성되어 있나?

그렇다면 이 코드는 메모리 내에서 몇 bytes 를 차지하고 있나?

명령어 (a) 의 메모리 주소를 16진수로 써라.

명령어 (b) 의 메모리 주소를 16진수로 써라.

명령어 (f) 의 메모리 주소를 16진수로 써라.

명령어 (h) 의 메모리 주소를 16진수로 써라.

명령어 (j) 의 메모리 주소를 16진수로 써라.
이 프로그램이 수행되어서 끝날 때까지 명령어가 수행되는 순서를 (a)~(q) 기호를 사용하여 나열하라. (예를 들어 a-b-a-c-d-k 이런 식으로)

a-

명령어 (g) 를 첫번째로 수행할 때 다음의 값들을 16진수로 써라.
$sp 의 값 :

$t0 의 값 :

$a0 의 값 :

$pc 의 값 :

명령어 (p) 를 첫번째로 수행할 때 다음의 값들을 16진수로 써라.
$sp 의 값 :

$t0 의 값 :

$a0 의 값 :

$v0 의 값 :

$pc 의 값 :

$ra 의 값 :

명령어 (n) lw $16, 4($8)
를 2진수로 표현하라.

명령어 (h) bne $2, $0, L
를 2진수로 표현하라.

명령어 (q) j end
를 2진수로 표현하라.