컴퓨터 구조 중간고사 (2019/10/23) 학번: 이름:

시험에서는 MIPS 어셈블리어를 사용하며 유사명령어는 사용하지 않는다. MIPS opcode 표는 다음을 참조하라.

0x00 sll (function field : 0x00)
0x00 srl (function field : 0x02)
0x00 mult (function field : 0x18)
0x00 div (function field : 0x1a)
0x00 add (function field : 0x20)
0x00 addu (function field : 0x21)
0x00 sub (function field : 0x22)
0x00 and (function field : 0x24)
0x00 or (function field : 0x25)
0x00 nor (function field : 0x27)
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

1.컴퓨터 프로세서에서 사용되는 명령어들의 집합 및 그 정의를 나타내는 용어를 쓰시오.

[10] Instruction Set Architecture

2. 프로세서 내부에 위치하는 작고 빠른 임시의 메모리를 지칭하는 용어를 쓰시오.

[10] Register

3. 5번 레지스터가 -2 라는 정수를 가지도록 하는 MIPS 어셈블리 명령어를 한 개 쓰시오.

[10] addi $5, $0, -2

4. 위의 명령어를 이진수로 표현하고 다시 그것을 16진수로 표현하시오.

1) 2진수로
[5] 001000 00000 00101 1111 1111 1111 1110

2) 16진수로
[5] 0x2005 FFFE

5. 위의 명령어를 수행한 결과 5번 레지스터의 값을 16진수로 표현하시오.

[10] 0xFFFF FFFE

6. MIPS 어셈블리 언어를 사용하여 8번 레지스터가 0x7FFFFFFF 의 값을 가지도록 하는 명령어를 쓰시오. 명령어는 한 개 이상일 수 있음.

[10] lui $8, 0x7FFF
ori $8, $8, 0xFFFF

7. 4번 레지스터의 값이 0x00000001 일 때, add $8, $4, $5 명령어를 수행한 결과 overflow exception 이 일어나도록 하는 5번 레지스터의 값을 16진수로 쓰시오.

[10] 0x7FFF FFFF

8. 다음의 2진수를 MIPS assembly instruction 으로 해석해라.

0000 0000 1010 0100 0001 1000 0010 0000

[10] add $3, $5, $4

9. '목적 프로그램을 메인 메모리에 적재해서 실행할 수 있게 하는 시스템 프로그램'을 나타내는 용어를 쓰시오.

[10] Loader

10. '실행 시에 프로그램과 링크되는 라이브러리 루틴' 을 나타내는 용어를 쓰시오.

[10] Dynamic Link Library

11. Little endian byte-order 를 사용하는 컴퓨터에서 다음과 같은 MIPS 프로그램을 수행했을 때 $t0 의 값을 16진수로 써라.

```
.data 0x10008004
.byte 0x11
.byte 0x33
.byte 0x55
.byte 0x77

.text
main:   lw  $t0, 4($gp)
```

[10] 0x77553311

다음의 MIPS 프로그램을 수행할 때, 각 명령어에 주어진 (a) ~ (n) 기호를 사용하여 답하시오. $sp 의 초기값은 0x7fffff84 이다. PC 의 값은 현재 수행되고 있는 명령어의 주소를 가지고 있다.

```
.data 0x10000000
.word  16
.word  5
.text 0x00400200
main :  addi $sp, $sp, _-4_[3]___      # (a)
        sw   $ra, __0($sp) [3]_____          # (b)
        lui   $t1, 0x1000             # (c)
        lw    $a0, 0($t1)             # (d)
        lw    $a1, 4($t1)             # (e)
        jal    func                   # (f)
        lw   $ra, _0($sp) [3]_____           # (g)
        addi $sp, $sp, _4 [3]____            # (h)
        jr      ___$ra [3]__                  #(i)
func:   div   $a0, $a1                #(j)
        mflo $a0                      #(k)
        bne  $a0, $0, func            #(l)
        mfhi  $v0                     #(m)
        jr      ___$ra [3]__                  #(n)
```

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

[10] 14

13. 이 코드는 메모리의 text segment 내에서 몇 bytes 를 차지하고 있나?

[10] 56

14. 명령어 (a) 의 메모리 주소를 16진수로 써라.

[3] 0x0040 0200

15. 명령어 (b) 의 메모리 주소를 16진수로 써라.

[3] 0x0040 0204

16. 명령어 (g) 의 메모리 주소를 16진수로 써라.

[3] 0x0040 0218

17. 명령어 (j) 의 메모리 주소를 16진수로 써라.

[3] 0x0040 0224

18~23. 코드의 빈칸 6개를 채워라.

24. 이 프로그램이 수행되어서 끝날 때까지 명령어가 수행되는 순서를 (a)~(n) 기호를 사용하여 나열하라. (예를 들어 a-b-a-c-d-k 이런 식으로)

[10]

a-b-c-d-e-f-j-k-l-j-k-l-m-n-g-h-i

25~31. 명령어 (m) 를 수행할 때 다음의 값들을 16진수로 써라.

$sp 의 값 : [3] 0x7FFF FF80

$t1 의 값 : [3] 0x1000 0000

$a0 의 값 : [3] 0x0000 0000

$a1 의 값: [3] 0x0000 0005

$v0 의 값 : [3] 0x0000 0003 (unknown is ok)

$pc 의 값 : [3] 0x0040 0230

$ra 의 값 : [3] 0x0040 0218

32. 명령어 (e) lw $5, 4($9) 를 2진수로 표현하라.
($5는 $a1이고 $9 는 $t1 이다.)

[10] 100011 01001 00101 0000 0000 0000 0100

33. 명령어 (f) jal func 를 2진수로 표현하라.

[10] 000011 0000 0100 0000 0000 0010 0010 01

34. 명령어 (l) bne $4, $0, func 를 2진수로 표현하라.
($4 는 $a0 이다. Branch Target Address 를 계산할 때 PC 는 현재 수행 중인 명령어의 주소를 가지고 있다.)

[10] 000101 00100 00000 1111 1111 1111 1110

35~45. 다음 그림의 회로에서 ALU 는 3-bit ALU operation control 의 값이 000 이면 AND, 001이면 OR, 010 이면 덧셈, 011 이면 뺄셈, 100 이면 slt 연산을 수행한다.
명령어 (e) lw $5, 4($9) 를 이 회로에서 수행하는 클락 사이클이 끝나기 직전에, 표시된 부분의 값들을 다음과 같이 써라.

35. 32-bit 16진수로 [3] 0x0040 0210

36. 32-bit 16진수로 [3] 0x0000 0004

37. 32-bit 16진수로 [3] 0x0040 0214

38. 5-bit 2진수로 [3] 01001

39. 5-bit 2진수로 [3] 00101

40. 16-bit 16진수로 [3] 0x0004

41. 32-bit 16진수로 [3] 0x0000 0004

42. 32-bit 16진수로 [3] 0x1000 0000

43. 32-bit 16진수로 [3] 0x1000 0004

44. 32-bit 16진수로 [3] 0x0000 0005

45. 32-bit 16진수로 [3] 0x0000 0005

46~50. 이 때 다음의 control signal 의 값들을 2진수로 써라.
MUX 의 selection control 의 값이 0 이면 MUX 의 출력은 위쪽 입력과 같고 1이면 MUX 의 출력은 아래쪽 입력과 같다.

46. RegWrite (1 bit) [3] 1

47. ALUSrc (1 bit) [3] 1

48. ALU operation (3 bits) [3] 010

49. MemWrite (1 bit) [3] 0

50. MemRead (1 bit) [3] 1

51. MemtoReg (1 bit) [3] 0

100011 01001 00101 0000 0000 0000 0100