CS 220 | 328034 ID | Isaac Sullivan

BRADLEY University

CS 220 Computer Architecture

HW 07 - MIPS 2

Fall 2023

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Part 0: Reading

- Study the following Tutorials on MIPS Assembly Programming
    1. **Bitwise Logic with Immediate Operands [This is a link]**
    2. **Shift Instructions and Logic Instructions [This is a link]**
    3. **Memory Access Instructions [This is a link]**

Part 1: Questions on MIPS Assembly Programming

Note that, for each of the questions, copy and paste to the following table and provide your **answer** with **Justification.**

For Example:

Question 0

Q0	In the following instruction sub $25, $16, $17 What register holds the result? A. $25 B. $16 C. $17 D. $0
	Answer: A. $25
	Justification: sub $d, $s, $t ⇒ $d = $s - $t $25 is the destination register.

A. Questions on Bitwise Logic with Immediate Operands

Question 1

Q1	What is it called when a logical operation is performed between the bits of each column of the operands to produce a result bit for each column? **A.** logic operation **B.** column operation **C.** bitwise operation **D.** immediate operation
	Answer: C
	Justification: A bitwise operation is where a logical operation is performed on the bits of each column of the operands

Question 2

Q2	When an operation is performed, what is true of the operands in the ALU? **A.** At least one operand must be 32 bits wide. **B.** Each operand can be any size. **C.** Each operand can be any size up to 64 bits. **D.** Each operand must be 32 bits wide.
	Answer: D
	Justification: Because they are adding two 16 bit operations together, and they need another 16 bit operation to catch overflow

Question 3

Q3	What is it called with a bit pattern is copied into a register? **A.** The register is **loaded** with the pattern. **B.** The register is **stuffed** with a pattern. **C.** The register is **stored** with the pattern.

	D. The register is **cached** with the pattern.
	Answer: A
	Justification: Copying a bit pattern into a register is usually called loading the register

QUESTION 4

Q4	Which of the following instructions **sets** all the bits in the low order byte of register $8 and leaves all other bits unchanged? **A.** ori $8,$8,0xFF **B.** ori $8,$0,0x00FF **C.** xori $8,$8,0xFF **D.** andi $8,$8,0xFF
	Answer: A
	Justification: The ori sets the bits in $8, and it can't be B because you do not need to call the $0 register

QUESTION 5

Q5	Which of the following instructions **sets** bit 0 of register $8 and clears all other bits? **A.** ori $8, $8, 0x1 **B.** ori $8, $0, 0x0001 **C.** andi $8, $0, 0x01 **D.** andi $8, $8, 0xFF
	Answer: D
	Justification: We set the bit again by comparing it with itself and clears all the other bits

QUESTION 6

Q6	What is the result of performing an exclusive or of a bit pattern with itself? **A.** The result is the same as the original. **B.** The result is the reverse of the original. **C.** The result is all zero bits. **D.** The result is all one bits.
	Answer: C
	Justification: Because the two operands are identical, only two XOR operations are involved: 0 XOR 0 = 0 and 1 XOR 1 = 0

QUESTION 7

Q7	Do all machine instructions follow the same format? **A.** No. There are several basic formats for machine instructions. **B.** No. The format for each opcode is completely different than for any other opcode. **C.** Yes. The only difference between machine instructions is the opcode. **D.** Yes. The 32 bits are divided into 4 equal parts.
	Answer: A
	Justification: Every processor or processor family has its own instruction set.

QUESTION 8

Q8	When a machine instruction has an immediate operand, how many bits does that operand have? **A.** 8 **B.** 16

	C. depends on the opcode. **D.** as many bits as needed for the data
	Answer: B
	Justification: Some machine instructions use 16 of their 32 bits to hold one of the two operands. An operand that is directly encoded as part of a machine instruction is called an immediate operand.

QUESTION 9

Q9	Perform an **AND** operation between these two operands: 0110 1100 **A.** 1111 **B.** 0101 **C.** 0110 **D.** 0100
	Answer: D
	Justification: 0 & 1 = 0, 1 & 1 = 1, 1 & 0 = 0, 0 & 0 = 0

QUESTION 10

Q10	Perform an **XOR** operation between these two operands: 0110 1100 **A.** 1110 **B.** 1101 **C.** 1010 **D.** 0100

	Answer: C
	Justification: 0 & 1 = 1, 1 + 1 = 2 so it defaults to 0, 1 + 0 = 1, 0 + 0 = 0

B. Questions on Shift Instructions and Logic Instructions

Note: for each of the Quiz questions, provide your answer with Justification.

Question 1

Q1	Shift left logical the following bit pattern by one position: 0011 1111 **A.** 0011 1111 **B.** 0001 1111 **C.** 0111 1110 **D.** 0111 111
	Answer: C
	Justification: You drop the zero from the front, the add a zero to the LSB side to keep the amount of bits consistent

Question 2

Q2	Shift left logical the following bit pattern by two positions: 0011 1111 **A.** 0001 1111 **B.** 1111 1100 **C.** 1111 1111 **D.** 0111 1110
	Answer: B
	Justification: You drop both zeros as you shift the two out, then add two new zeros to the rightmost point

QUESTION 3

Q3	Here is a program that loads register $5 with a bit pattern. Complete the program so the pattern in register $5 shifted left logical by 4 positions.
	`ori $5, $0, 0x900F # put a bit pattern into register $5`
	`sll ___, ___, ___ # shift left logical by four, put`
	`# pattern remains in register $5`
	A. sll $5, $5, $4
	B. sll $4, $5, $5
	C. sll $5, $0, 4
	D. sll $5, $5, 4

	Answer: D
	Justification: The operation takes register 5, tells to put the result in register 5, then shifts the program by 4

QUESTION 4

Q4	Which of the following instructions has the effect of multiplying the unsigned integer represented by the bits in register eight by sixteen? **A.** sll $8, $0, 16 **B.** sll $8, $8, 4 **C.** sll $8, $8, 2 **D.** sll 4, $8, $8
	Answer: B
	Justification: 2^4 = 16, and you place the exponent on the right

QUESTION 5

Q5	Which of the following instructions has the effect of dividing the unsigned integer represented by the bits in register eight by sixteen? **A.** sll $8, $0, 16 **B.** srl $8, $8, 4 **C.** sll $8, $8, 2

	D. srl 4, $8, $8
	Answer: B
	Justification: 2^4 = 16, and you place the exponent to the right

Question 6

Q6	Write the assembly language statement that will reverse the values of each bit in register $5 and put the result in register $8. **A.** nori $8,$5,$0 **B.** nor $8,$5,$0 **C.** xor $8,$5,$0 **D.** nor $5,$8,$0
	Answer: B
	Justification: The collection register always goes on the right, and nor reverses the values in the register

Question 7

Q7	Write the assembly language statement that will reverse the values of each bit in register $7 and put the result in register $12. **A.** or $7,$0,$12 **B.** nor $7,$7,$12 **C.** not $7,$7,$12 **D.** nor $12,$7,$0
	Answer: C
	Justification: not will reverse each bit in register $7 after taking both in and deposit it in $12

QUESTION 8

Q 8	Write the assembly language statement that computes the bit-wise OR between \$7 and \$8 puts the result in register \$9. **A.** or \$9,\$8,\$7 **B.** or \$7,\$8,\$9 **C.** or \$0,\$8,\$7 **D.** or \$8,\$9,\$7
	Answer: A
	Justification: The operation gets register \$9 and then computes \$8 and \$7, then puts the result into \$9

QUESTION 9

Q9	Write the assembly language statement that computes the bit-wise XOR between \$7 and \$8 puts the result in register \$9. **A.** xori \$9,\$8,0x7 **B.** xor \$7,\$8,\$9 **C.** xor \$9,\$8,\$7 **D.** xor \$8,\$9,\$7
	Answer: C
	Justification: You declare the incoming register first, then the calculation comparing \$8 and \$7 are run

QUESTION 10

Q10	Write the assembly language instruction that copies the bit pattern in register \$13 to register \$15. **A.** ori \$15,\$13,\$0 **B.** or \$15,\$13,\$0 **C.** or \$15,\$13,\$15

	D. xor $15,$13,$13
	Answer: A
	Justification: The $0 address is just a placeholder, and the copy statement is placed to the left

C. Questions on Memory Access Instructions

1. How many addresses does MIPS main storage have?

   Up to 32 address lines

2. What is the name of the operation that copies data from main memory into a register?

   The store operation

3. Which of the following addresses are word-aligned? Justify your answer.

   0x000AE430 Y
   0x00014433 N
   0x000B0737 N
   0x0E0D8848 Y

   The first and the last addresses both end in a multiple of four. The middle two do not. If the rightmost two (or more) bit positions are zero, the integer is a multiple of four.

4. Here is a bit pattern, with the most significant bits written on the left (as is usual in print): 0xFFBBAACC. Copy the bytes to memory using big-endian and little-endian orders:

Little Endian	
Address	**Contents**
4003	FF
4002	BB
4001	AA
4000	CC

Big Endian	
Address	**Contents**
4003	CC
4002	AA
4001	BB
4000	FF

5. Write the instruction that loads the word at address 0x00400080 into register $8. Assume that the register $10 contains **0x00400000**.
lw $8, 0x80($10)

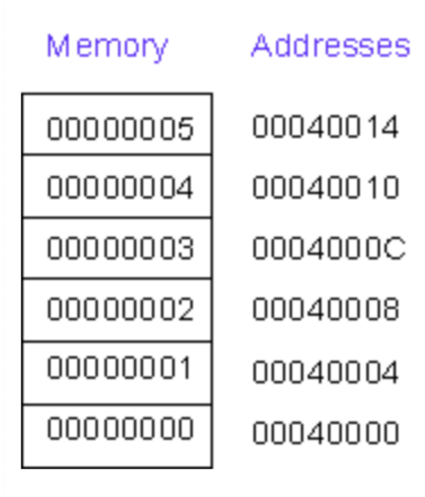
Hints: Here is the load word instruction in assembly language:

```
lw   d,off(b)          # $d <-- Word from memory address b+off
                       # b is a register. off is 16-bit two's complement.
```

At execution time two things happen: (1) an address is calculated using the base register b and the offset off, and (2) data is fetched from memory at that address.

6. Given memory and registers $12 and $13 as below.

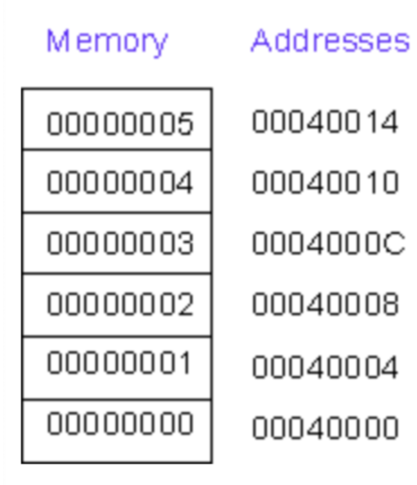
   A. Write the instruction that puts the value **0x00000002** into register $12.
      lw $12 ,0x8($13)
   B. Write the instruction that puts the value **0x00000004** into register $12.
      lw $12 ,0x10($13)

Register $12 contains 0xFFFFFFFF Register $13 contains 0x00040000	

Memory	Addresses
00000005	00040014
00000004	00040010
00000003	0004000C
00000002	00040008
00000001	00040004
00000000	00040000

7. Given registers $12 and $13 and memory.
   A. Write the instruction that puts the word **0xFFFFFFFF** into memory location 0x00040010.
   sw $12 ,-4($13)
   B. Write the instruction that puts the word **0xFFFFFFFF** into memory location 0x0004000C.
   sw $12 ,-8($13)
   C. Write the instruction that puts the word **0xFFFFFFFF** into memory location 0x00040004.
   sw $12 ,-20 ($13)

Hint: it is OK to specify the 16-bit offset as a signed decimal integer.

Register $12 contains 0xFFFFFFFF Register $13 contains 0x00040014	

Memory	Addresses
00000005	00040014
00000004	00040010
00000003	0004000C
00000002	00040008
00000001	00040004
00000000	00040000