Review Session Packet 1

Binary Definitions:

1. How many 0s/1s are in the following?

Bit: _____

Nibble: _____

Byte: _____

Half Word: _____

Word: _____

2. What is the largest number that you can store in an unsigned 8-bit binary number? What is this in decimal form?

3. What is the largest number that you can store in a signed 8-bit binary number? What is this value in decimal form?

4. How can you distinguish between a positive and a negative signed binary number? What does this differentiating factor represent?

Conversions:

5. Fill in the following power of two values.

2^0 = _____ 2^1 = _____ 2^2 = _____ 2^3 = _____

2^4 = _____ 2^5 = _____ 2^6 = _____ 2^7 = _____

2^8 = _____ 2^9 = _____ 2^{10} = _____ 2^{11} = _____

2^{12} = _____

6. Convert the decimal number "76" into binary using one of the following methods.

Method 1: Divide by 2 and Reverse

76 / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

_____ / 2 = _____ (Remainder: _____)

A) *What are all of the remainders from your calculations, put together?*

B) *Now, reverse the remainders.*

C) *Which of the above two is the correct conversion to 76?*

D) *What is your final answer?*

_____ $_2$

Method 2: Utilize Pre-Determined Slots

A) *What's the highest power of two that can fit in the number 76?*

B) Place a 1 in that spot.

2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

C) Subtract that number from 76.

D) Now, what's the highest power of two that can fit in the new number?

E) Place a 1 in that spot.

2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

F) Subtract that number from the previous.

G) Repeat this algorithm until all positions have been filled.

2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

H) Now, put zeroes in all of the remaining positions.

2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

I) *What is your final answer?*

_____ 2

7. Convert the binary number "01110010" into decimal.

A) Write in the values for the powers of two underneath this binary statement.

0	1	1	1	0	0	1	0
____	____	____	____	____	____	____	____

B) Now, add up the numbers that you wrote in for any slot with a 1 over them.

C) What's the final answer that you get?

_____ 10

(You can double check your answer by applying one of the algorithms described in problem 2.)

8. Convert the binary number "01001" into octal.

A) Sign extend the number so that the number of bits is a factor of 3.

B) Divide your sign-extended number into sections with 3 bits each.

C) Calculate the total of the binary statement within each of these sections, in decimal form.

D) What is the final answer that you get?

_____ 8

9. Now convert "11011" into hexadecimal.

A) Sign extend the number so that the number of bits is a factor of 4.

B) Divide your sign-extended number into sections with 4 bits each.

C) Calculate the total of the binary statement within each of these sections, in decimal form.

D) Are there any summations that are greater than 9? What's another way that we can represent these numbers so that we don't confuse them with our other numbers?

E) *What is the final answer that you get?*

0x _____

Addition & Subtraction in Binary:

10. What's the difference between carry out and overflow?

11. What's the difference between logical and bitwise operators? What are these operators?

12. What are the two steps required for 2s complement?

13. Calculate the following sums in binary.

Carry-In	*1*	*1*	*1*	*1*	*0*	*0*	*0*	*0*
a	*1*	*1*	*0*	*0*	*1*	*1*	*0*	*0*
b	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*0*
Total								
Carry-Out								

14. Calculate a - b:
a = 0011 1100 0010 1111
b = 0111 1000 0110 0001

Bit Shifting:

15. Write a MIPS instruction that will multiply the value stored in `$t0` by 16.

16. Write two MIPS instructions that will multiply the value stored in `$t0` by 20.

17. Write a MIPS instruction that will divide the value stored in `$t0` by 8.

18. What's the difference between `srl` and `sra`? Why doesn't `sll` have an `sla` counterpart?

Masking:

19. Say we have the 8 bit binary number: *0011 1011*. *Assume that you're reading bits right to left, and indexing at 1. Calculate the masks and logical operators you would want to use if you were going to:*

A) Isolate the bits at positions 4, 6, and 7?

B) Set the bit values to 1 at positions 3, 4, 5, 6, and 8?

C) Flip the bits at positions 1, 2, and 5?

MIPS:

20. What's the difference between an assembly register and a variable in a higher level language?

21. What's the difference between `add`/`slt`/`or`/`and`/`xor`/`addu` and `addi`/`slti`/`ori`/`andi`/`xori`/`addiu`?

22. Why don't we need a `subi` instruction?

23. What's the difference between the register order of `sw` and `lw`?

24. When would you use `.data`, `.text`, and `.asciiz`?

25. When would you use a label and why would it be important? What MIPS instructions would you use to populate a value stored in "`MYSTERY_LABEL`" into `$t0`?

26. What are the `syscall` values for printing integers, characters, and strings?

27. How are the different data structures represented in assembly? More specifically, how does an assembly language represent data structures like integers, chars, Strings, and booleans? Are they all the same?

28. Using two instructions each, how would you represent the following branching statements?

a) `if (t0 < t1), branch to AFTER`

b) `if (t0 > t1), branch to AFTER`

c) `if (t0 <= t1), branch to AFTER`

d) `if (t0 >= t1), branch to AFTER`