Introduction to Computer Science-102

Quiz_2

1. Show how the computer finds the result of 33.1875 − 0.4375 = (+32.75). (10%)

$33.1875 - 0.4375 = (100001.0011)_2 - (0.0111)_2 = 2^5 \times (1.000010011)_2 - 2^{-2} \times (1.11)_2$. These two numbers are stored in floating-point format as shown, but we need to remember that each number has a hidden 1 (which is not stored, but assumed). $E_1 = 127 + 5 = 132 = (10000100)_2$ and $E_2 = 127 + (-2) = 125 = (01111101)_2$

	S	E	M
A	0	10000100	00001001100000000000000
B	0	01111101	11000000000000000000000

The first two steps in UML diagram is not needed. Since the operation is subtraction, we change the sing of the second number.

	S	E	M
A	0	10000100	00001001100000000000000
B	1	01111101	11000000000000000000000

We denormalize the numbers by adding the hidden 1's to the mantissa and incrementing the exponent. Now both denormalized mantissas are 24 bits and include the hidden 1's. They should store in a location to hold all 24 bits. Each exponent is incremented.

	S	E	Denormalized M
A	0	10000101	**1**00001001100000000000000
B	1	01111110	**1**11000000000000000000000

We align the mantissas. We increment the second exponent by 7 and shift its mantissa to the right seven times.

	S	E	Denormalized M
A	0	10000101	**1**00001001100000000000000
B	1	10000101	0000000**1**11000000000000000

Now we do sign-and-magnitude addition treating the sign and the mantissa of each number as one integer stored in sign-and-magnitude representation.

	S	E	Denormalized M
R	0	10000101	10000011000000000000000

There is no overflow in mantissa, so we normalized.

	S	E	M
R	0	10000100	00000110000000000000000

The mantissa is only 23 bits because there is no overflow, no rounding is needed.

$$E = (10000100)_2 = 132, M = 0000011$$

The result is

$$(1.0000011)_2 \times 2^{132-127} = (100000.11)_2 = 32.75$$

2. Show the result of the following operations. (10%)

   a. $(99)_{16}$ OR $(99)_{16}$ $(99)_{16}$

   b. $(99)_{16}$ OR $(00)_{16}$ $(99)_{16}$

   c. $(99)_{16}$ AND $(FF)_{16}$ $(99)_{16}$

   d. $(FF)_{16}$ AND $(FF)_{16}$ $(FF)_{16}$

3. The CPU and memory are normally connected by three groups of connections, each called a bus. What are the three kinds of them? And what are their own function? (10%) 1. data bus 2. address bus 3. control bus

   The databus (connections between and within the CPU, memory, andperipherals) used to carry data.

   The addressbus:connections between the CPU and memory which carry theaddres s from/to which the CPU wishes to read or write

   The controlbus:In a digitalcomputer, the signal paths that carry commands fromth e instruction decode logic to various different functional units suchas the ALU, me mory address register, memory data register andother buffers.

4. Show the result of the following operations assuming that the numbers are stored in 16-bit two's complement representation. Show the result in hexadecimal notation. (10%)

   a. $(012A)_{16} + (0E27)_{16}$ $(0F51)_{16}$

   b. $(712A)_{16} + (9E00)_{16}$ $(0F2A)_{16}$

5. What are the two methods for handling the addressing of I/O devices?What is the diffenence between them?(10%)

   The only difference is the instruction. If the instruction refers to a word in main memory, data transfer is between main memory and the CPU. If the instruction identifies an I/O device, data transfer is between the I/O device and the CPU.
   There are two methods for handling the addressing of I/O devices: isolated I/O and memory-mapped I/O.

6. Compare and contrast CISC architecture with RISC architecture. (10%)

   CISC (Complex Instruction Set Computer) has a large set of instructions to execute commands at the machine level. This makes the circuitry of the CPU and the control unit very complicated. RISC (Reduced Instruction Set Computer) uses a small set of instructions. Complex operations are accomplished using a set of simple commands.

7. How many bytes of memory are needed to store a full screen of data if the screen is made of 30 lines with 70 characters in each line? The system uses ASCII code, with each ASCII character stored as a byte. (10%)

   We need 30 × 70 = 2100 bytes.

8. An imaginary computer has sixteen data register (R0 to R15), 1024 words in memory, and 16 different instructions (add, subtract, and so on). If a typical instruction uses the following format: **instruction M R2.** If the computer uses the same size of word for data and instructions, what is the size of each data register? (10%) 4+10+4=18bits

9. Each computer instruction consists of two parts, what are they? And how many bits are contained in the first field and the second field? (10%)

   operation code (opcode) and the operand (s)

   Each instruction consists of sixteen bits divided into four 4-bit fields. The leftmost field contains the opcode and the other three fields contains the operand or address of operand

10. Using the instruction set of the simple computer in the following table, write the code for a program that performs the following calculation:

    $$B \leftarrow A - 2$$

    A and 2 are integers in two's complement format. The user types the value of A and the value of B is displayed on the monitor. The keyboard is assumed to be memory location $(FE)_{16}$, and the monitor is assumed to be $(FF)_{16}$.(10%)

Instruction	Code	Operands			Action
	d_1	d_2	d_3	d_4	
HALT	0				Stops the execution of the program
LOAD	1	R_D	M_S		$R_D \leftarrow M_S$
STORE	2	M_D		R_S	$M_D \leftarrow R_S$
ADDI	3	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} + R_{S2}$
ADDF	4	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} + R_{S2}$
MOVE	5	R_D	R_S		$R_D \leftarrow R_S$
NOT	6	R_D	R_S		$R_D \leftarrow \overline{R_S}$
AND	7	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ AND R_{S2}
OR	8	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ OR R_{S2}
XOR	9	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ XOR R_{S2}
INC	A	R			R $\leftarrow$ R + 1
DEC	B	R			R $\leftarrow$ R – 1
ROTATE	C	R	n	0 or 1	Rot_n R
JUMP	D	R	n		IF $R_0 \neq$ R then PC = *n*, otherwise continue

Key: R_S, R_{S1}, R_{S2}: Hexadecimal address of source registers
R_D: Hexadecimal address of destination register
M_S: Hexadecimal address of source memory location
M_D: Hexadecimal address of destination memory location
n: hexadecimal number
d_1, d_2, d_3, d_4: First, second, third, and fourth hexadecimal digits

Step	Code(hexadecimal)	Description
1	1FFE	// $R_F \leftarrow M_{FE}$, Input A from keyboard to R_F
2	240F	// $M_{40} \leftarrow R_F$, Store A in M_{40}
3	1040	// $M_{40} \leftarrow R_0$, Load A from M_{40} to R_0
4	B000	// $R_0 \leftarrow R_0 - 1$, Decrement A
5	B000	// $R_0 \leftarrow R_0 - 1$, Decrement A
6	2410	// $M_{41} \leftarrow R_0$, Store The result in M_{41}
7	1F41	// $R_F \leftarrow M_{41}$, Load the result to R_F
8	2FFF	// $M_{FF} \leftarrow R_F$, Send the result to the monitor
9	0000	// Halt