UNIVERSITY OF CALIFORNIA, RIVERSIDE

Department of Computer Science and Engineering

CS61 – Machine Organization and Assembly Language

Midterm 1

August 8, 2001

22

Name: Solution Key **Student ID#:** _____

Please print legibly

(Numbers in parenthesis denote total possible points for question.)

1. If the ASCII code for the character "E" is 69 in decimal, what is the ASCII code for the character "H" in decimal? (1)

Answer

72.

2. Convert the decimal number $76\frac{21}{32}$ into a 32-bit floating point number representation using the method described in the textbook. (4)

Answer

$76_{10} = 1001100_2$

$\frac{21}{32} = \frac{16}{32} + \frac{4}{32} + \frac{1}{32} = \frac{1}{2} + \frac{1}{8} + \frac{1}{32} = 1\times2^{-1} + 1\times2^{-3} + 1\times2^{-5} = 0.10101$

Therefore, $76\frac{21}{32}$ in binary is 1001100.10101

Normalizing this binary number gives 1.00110010101×2^6

The exponent representation is $6 + 127 = 133_{10} = 10000101$.

Thus, the floating point representation is:

0 10000101 00110010101000000000000

1 point for the fraction
1 point for the exponent
1 point for the normalized number.
1 point for the final 32 bit string

3. Fill in the following table by performing the necessary conversions. The binary numbers are in 2's complement. (4)

Binary	Octal	Decimal	Hexadecimal
1001101			
	536		
		735	
			6E7

Answer:

Binary	Octal	Decimal	Hexadecimal
1001101	715 (115)	-51	CD (4D)
101011110	**536**	-162	F5E (15E)
01011011111 (1011011111)	1337	**735**	2DF
011011100111 (11011100111)	3347	1767	**6E7**

4. Perform the following binary arithmetic:

   a) 11010010111 ÷ 1010 using unsigned numbers (positive integers). (2)
   b) 010111 × 110111 using signed numbers (i.e. 2's complement). (2)

Answer:

a)

```
          10101000
1010 ) 11010010111
       1010
         1100
         1010
         001010
           1010
           0000111
```

b)

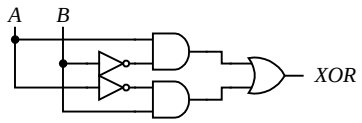
```
          010111
×         110111
000000010111
00000010111
0000010111
000000000
00010111
+ 110100100000
111100110001
```

5. The truth table for the 2-input exclusive-or (XOR) gate is as follows:

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Draw the circuit that implements the XOR gate using AND, OR, and NOT gates. (4)

Answer



6. The LDR (Load) and JMPR (Jump) instruction formats and operations are described as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	×	×	×	×	×	×	×	×	×	×	×	×
LDR				Destination register			Base register			Offset					

The LDR instruction add the contents of the Base register to the Offset value to form the address of a memory location. It then loads the contents stored in that memory location into the Destination register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	×	×	×	×	×	×	×	×	×
JMPR							Base register			Offset					

The JMPR instruction add the contents of the Base register to the Offset value to form the address of a memory location. It then loads that address into the PC.

Assume that the current memory and register contents are as follows:

Registers	Register contents (16 bits)
PC	0000000010011000
R1	0000000000000001
R2	0000000001011001
R3	0000000000000010

Memory address	Memory contents (16 bits)
0000000010010100	0110010001000001
0000000010010110	1100000101000111
0000000010011000	0110010011000001
0000000010011010	1100000010000010

Using the values from the table above, trace through the execution of the instructions until reaching an unspecified memory location. After reaching this point, write the contents of the four registers in the table below. Assume that the PC is incremented by 2 at the end of each Fetch cycle.

Registers	Register contents (16 bits)
PC	
R1	
R2	
R3	

Answer

Registers	Register contents (16 bits)
PC	0000000010011010
R1	0000000000000001
R2	content of memory location 3
R3	0000000000000010