CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

UNIVERSITY OF WISCONSIN—MADISON

Prof. Mark D. Hill

TAs: Sujith Surendran, Pradip Vallathol

Midterm Examination 1

In Class (50 minutes)

Wednesday, October 2, 2013

Weight: 17.5%

NO: BOOK(S), NOTE(S), OR CALCULATORS OF ANY SORT.

The exam has 8 pages. **Circle your final answers**. Plan your time carefully since some problems are longer than others. You **must turn in the pages 1-7**. Use the blank sides of the exam for scratch work.

LAST NAME: _____

FIRST NAME: _____

ID# _____

Problem	Maximum Points	Points Earned
1	6	
2	2	
3	4	
4	2	
5	6	
6	3	
7	2	
8	3	
9	2	
Total	30	

Problem 1 **(6 Points)**

For the following problems, circle the **best** answer. **Choose only one answer per question.**

i How many unique bit patterns can be represented using n bits?

a n
b 2n
c 2^n
d $2^{(n-1)}$

ii How many Instruction Set Architectures (ISAs) are usually implemented by a given microarchitecture?

a None
b 1
c 2
d There is no limit

iii What is the binary representation of the hexadecimal value **0x6D**

a 0110 1100
b 0110 1101
c 0011 1100
d 0011 1101

iv In how many ways can **0** be represented in Signed Magnitude form?

a 1
b 2
c None
d Cannot be determined

v Which of the following are specified by an Instruction Set Architecture (ISA)?

a Addressing modes
b Instructions
c Data types
d All of the above

vi Digital designs are preferred over analog designs because it is hard to increase the accuracy of analog designs.

a True
b False

Problem 2 **(2 Points)**

Perform binary arithmetic for the following pairs of 6-bit 2's complement numbers.

a
```
  011001
+ 101101
----------
  000110
```

b
```
  010111
- 111110
----------
  011001
```

Problem 3 **(4 Points)**

Perform the specified logical operations on the following binary numbers. Express your result in **hexadecimal**.

a `10111110 AND (NOT(10010001))` (2 Points)

`= 10111110 AND 01101110 = 00101110 = 0x2E`

b `(NOT(0011)) OR (1101 AND 0101)` (2 Points)

`= 1100 OR 0101 = 1101 = 0xD`

Problem 4 **(2 Points)**

Convert the ASCII string “Fall_13” to its **hexadecimal** representation. Only represent the characters within the quotation marks and assume it is null terminated. **Hint: See ASCII to hexadecimal table on the last page of the exam.**

0x 46 61 6C 6C 5F 31 33 00

Problem 5 **(6 Points)**

What are the smallest and largest numbers that can be represented using 5-bits in the following formats?

Format	**Smallest number**	**Largest number**
Unsigned integer	0	31
Signed Magnitude	-15	15
2’s complement	-16	15

Problem 6 **(3 Points)**

Answer the following questions briefly. **(1 or 2 sentences)**

a Why cannot natural language be used as programming languages? (1 Point)

Natural languages are too ambiguous.

b What is the advantage of using fixed point representation of fractions? What is the advantage of using the IEEE floating point representation? (2 Points)

Fixed point representation can be used to directly do addition and subtraction operation and do not waste the space of the exponent. IEEE floating point can be used to represent a large range of numbers.

Problem 7 **(2 Points)**

Arrange the following in order, from highest to lowest level of abstraction. The first element in the sequence has been filled for you.

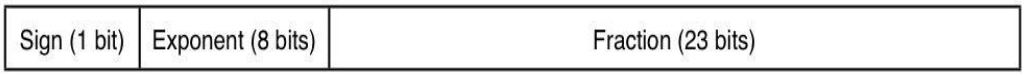
i. Binary/Machine code
ii. Transistors/Devices
iii. Programming language
iv. Microarchitecture
v. Problem Statement

1	**v. Problem Statement**
2	iii. Programming language
3	i. Binary/Machine code
4	iv. Microarchitecture
5	ii. Transistors/Devices

Problem 8 **(3 Points)**

Convert the decimal value **2.5** into its single-precision floating point representation. Write your answer in **hexadecimal**.

Recall that the bits for the IEEE floating point number are as follows:



where $N = (-1)^{sign} * 2^{exponent-127} * 1.fraction$

2.5 = 10.1 = 1.01 * 2^1 = 1.01 * $2^{(128-127)}$
= 0 10000000 01000000000000000000000
= 0x40200000

Problem 9 **(2 Points)**

Does the following binary arithmetic operation result in an overflow for the following 4-bit 2's complement numbers?

`1100 + 1001`

Illustrate your answer by converting the operands and result to decimal. Assume the result is also represented using 4-bits.

```
  1100     (-4)
+ 1001     (-7)
--------
  0101     (5)
```

(-4) + (-7) = (-11) [not equal to 5]

So overflow!

ASCII Table

Character	Hex	Character	Hex	Character	Hex	Character	Hex
nul	00	sp	20	@	40	`	60
soh	01	!	21	A	41	a	61
stx	02	“	22	B	42	b	62
etx	03	#	23	C	43	c	63
eot	04	$	24	D	44	d	64
enq	05	%	25	E	45	e	65
ack	06	&	26	F	46	f	66
bel	07	‘ *(Apostr.)*	27	G	47	g	67
bs	08	(	28	H	48	h	68
ht	09	)	29	I	49	i	69
lf	0A	*	2A	J	4A	j	6A
vt	0B	+	2B	K	4B	k	6B
ff	0C	, *(Comma)*	2C	L	4C	l	6C
cr	0D	-	2D	M	4D	m	6D
so	0E	. *(Period)*	2E	N	4E	n	6E
si	0F	/	2F	O	4F	o	6F
dle	10	0	30	P	50	p	70
dc1	11	1	31	Q	51	q	71
dc2	12	2	32	R	52	r	72
dc3	13	3	33	S	53	s	73
dc4	14	4	34	T	54	t	74
nak	15	5	35	U	55	u	75
syn	16	6	36	V	56	v	76
etb	17	7	37	W	57	w	77
can	18	8	38	X	58	x	78
em	19	9	39	Y	59	y	79
sub	1A	:	3A	Z	5A	z	7A
esc	1B	;	3B	[	5B	{	7B
fs	1C	<	3C	\	5C	|	7C
gs	1D	=	3D	]	5D	}	7D
rs	1E	>	3E	^	5E	~	7E
us	1F	?	3F	_	5F	del	7F