NAME: ______________________ STUDENT #: ______________

EECE 259: Introduction to Microcomputers **QUIZ 1 – 201** **Jan 21, 2011abcd**

1. Fill in the rest of the table.

6

Decimal (unsigned)	Binary (%)	Hexadecimal ($)
42	%00101010	$2A
105 201 155 092	%0110 1001 %1100 1001 %1001 1011 %0101 1100	$69 $C9 $9B $5C
250	%11111010	$FA

2. Perform each of the operations in the table below. Express the answer in binary, decimal, and hexadecimal, and compute the flags Z, N, Vsub, Vadd, B and C. The grayed-out cells **will not be marked**.

4

1 mark for whole row correct

Operation on 8-bit values	Binary	Decimal	Hex	Z	N	V_{sub}	V_{add}	B	C
a. $FA + $08	%		$02	0	0				1
b. $FA + $06	%		$00	1	0				1
c. $FA + $07	%		$01	0	0				1
d. $FA + $09	%		$03	0	0				1
e. $69 + $3C	%		$A5	0	1		1		
f. $59 + $3C	%		$95	0	1		1		
g. $49 + $3C	%		$85	0	1		1		
h. $39 + $4C	%		$85	0	1		1		
i. $89 – $3D	%		$4C	0	0	1			
j. $90 – $39	%		$57	0	0	1			
k. $89 – $0D	%		$7C	0	0	1			
l. $90 – $0D	%		$83	0	1	0			
m. $22 – $23	%		$FF	0	1			1	
n. $23 – $32	%		$F1	0	1			1	
o. $22 – $33	%		$EF	0	1			1	
p. $32 – $32	%		$00	1	0			0	

3

3. Suppose you are using 8-bit signed values to compute something important, and –128 is subtracted from 0. Yes
Suppose you are using 8-bit signed values to compute something important, and –128 is added to 0. No
Suppose you are using 8-bit signed values to compute something important, and –127 is subtracted from –1. No
Suppose you are using 8-bit signed values to compute something important, and –128 is added to 0. No
Does this produce an overflow to be concerned about? Explain in plain English, not logic equations.

1: Yes or No

2: For explanation – see section 202

3

4. With 4-bit arithmetic, the equations for C and V_{sub} are C= $a_3b_3 + a_3\overline{r_3} + b_3\overline{r_3}$ and V_{sub} = $a_7 \cdot \overline{b_7} \cdot \overline{r_7} + \overline{a_7} \cdot b_7 \cdot r_7$.
Give an example where adding two 4-bit values results in both C=0 (ALSO: C=1)and V_{sub}=1. What is the correct way to interpret the V_{sub}=1 result (keep in mind that your example uses addition)?

2: example (1 for only a_3, b_3, r_3; 1 for 4-bit values)

1: interpretation of V_{sub}: during an addition, V_{sub} is meaningless and should be ignored.

7 (2+5)

5. Use the **Computational Datapath** worksheet on the back to perform each of the operations below. For each operation, write out the sequence of μ-ops required in RTN. Treat each operation as if it is a stand-alone operation that does not depend upon the previous one. Do not rely upon the registers having any particular initial values. Change only the register or memory location specified. There may be multiple solutions.

a. Mem[1] ← R2	c. R2 ← Mem[1]	e. Mem[1] ← R1	g. R1 ← Mem[1]
b. R2 ← R1 + Mem[2]	d. R3 ← 2 + Mem[R1]	f. R3 ← 2 + Mem[R1]	h. R1 ← R3 + Mem[2]

NAME: ______________________________ STUDENT #: ____________________

EECE 259: Introduction to Microcomputers **QUIZ 1 – 202** **Jan 21, 2011abcd**

6

1. Fill in the rest of the table.

Decimal (unsigned)	Binary (%)	Hexadecimal ($)
42	%0010 1010	$2A
181 201 155 092	%1011 0101 %1100 1001 %1001 1011 %0101 1100	$B5 $C9 $9B $5C
250	%1111 1010	$FA

2. Perform each of the operations in the table below. Express the answer in binary, decimal, and hexadecimal, and compute the flags Z, N, Vsub, Vadd, B and C. The grayed-out cells **will not be marked**.

4

1 mark for whole row correct

Operation on 8-bit values	Results								
	Binary	Decimal	Hex	Z	N	V_{sub}	V_{add}	B	C
a. $FB + $06	%		$01	0	0				1
b. $FA + $06	%		$00	1	0				1
c. $FB + $05	%		$00	1	0				1
d. $FA + $05	%		$FF	0	1				0
e. $69 + $8C	%		$F5	0	1		0		
f. $59 + $8C	%		$E5	0	1		0		
g. $39 + $3C	%		$75	0	0		0		
h. $39 + $2C	%		$65	0	0		0		
i. $09 – $0D	%		$FC	0	1	0			
j. $0D – $09	%		$04	0	0	0			
k. $F9 – $0D	%		$EC	0	1	0			
l. $09 – $FD	%		$0C	0	0	0			
m. $23 – $32	%		$F1	0	1			1	
n. $33 – $22	%		$11	0	0			0	
o. $22 – $33	%		$EF	0	1			1	
p. $32 – $23	%		$0F	0	0			0	

3

3. Suppose you are using 8-bit signed values to compute something important, and –128 is subtracted from 0. Yes
Suppose you are using 8-bit signed values to compute something important, and –128 is added to 0. No
Suppose you are using 8-bit signed values to compute something important, and –127 is subtracted from –1. No
Suppose you are using 8-bit signed values to compute something important, and –127 is added to –1. No
Does this produce an overflow to be concerned about? Explain in plain English, not logic equations.

1: Yes or No
2: For explanation
- Yes: +127 is the largest 8-bit signed value, so +128 overflows to –128
- No: 8-bit signed values range from –128 to +127

3

4. With 4-bit arithmetic, the logic equations for C and B are $C = a_3b_3 + a_3\bar{r}_3 + b_3\bar{r}_3$ and $B = \bar{a}_3b_3 + b_3r_3 + \bar{a}_3r_3$. Give an example where adding two 4-bit values results in both C=1 and B=1. What is the correct way to interpret the B=1 result (keep in mind that your example uses addition)?

A_3	B_3	R_3
0	1	0
1	1	1

2: example (1 for only a_3, b_3, r_3; 1 for 4-bit values)
1: interpretation of ~~Vsub~~ B: during an addition, B is meaningless and should be ignored

7 (2+5)

5. Use the **Computational Datapath** worksheet on the back to perform each of the operations below. For each operation, write out the sequence of μ-ops required in RTN. Treat each operation as if it is a stand-alone operation that does not depend upon the previous one. Do not rely upon the registers having any particular initial values. Change only the register or memory location specified. There may be multiple solutions.

a. Mem[1] ← R2
b. R2 ← R1 + Mem[2]

c. R2 ← Mem[1]
d. R3 ← 2 + Mem[R1]

e. Mem[1] ← R1
f. R3 ← 2 + Mem[R1]

g. R1 ← Mem[1]
h. R1 ← R3 + Mem[2]

Section 201

4/ $C = a_3 b_3 + a_3 \overline{r_3} + b_3 \overline{r_3}$

$V_{sub} = a_3 \overline{b_3}\,\overline{r_3} + \overline{a_3} b_3 r_3$

find example where $C=0$ and $V_{sub}=1$ by writing out a truth table for a_3, b_3, r_3

a_3	b_3	r_3	C	V_{sub}
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

← $C=0$ and $V_{sub}=1 \Rightarrow \overline{a_3} b_3 r_3$

← $C=1$ and $V_{sub}=1 \Rightarrow a_3 \overline{b_3}\,\overline{r_3}$

Example for $C=0$, $V_{sub}=1$:

```
  0 000
+ 1 000
  -----
  1 000
```

rest of 4-bit values must agree with column 3 result

$C=1$, $V_{sub}=1$:

```
  1 100
+ 0 100
  -----
  0 000
```

Section 202

4/

a_3	b_3	r_3	C	$B = \overline{a_3} b_3 + b_3 r_3 + \overline{a_3} r_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

either $\overline{a_3} b_3 \overline{r_3}$ or $a_3 b_3 r_3$

```
  0100        1100
+ 1100      + 1100
  ----        ----
  0000        1000
```