HW 1 Clay Ramey

problem set A

① 5-bit binary representation of decimal # 21

16	8	4	2	1
×	×	×	×	×
1	0	1	0	1

→ 21 − 16 = 5
5 − 4 = 1 ✓
1 − 1 = 0

answer => $\boxed{10101_2}$

② convert decimal # 219 to hex

128	64	32	16	8	4	2	1
×	×	×	×	×	×	×	×
1	1	0	1	1	0	1	1

219 − 128 = 91 ✓
91 − 64 = 27 ✓
27 − 16 = 11 ✓
11 − 8 = 3 ✓
3 − 2 = 1 ✓
1 − 1 = 0 ✓

→ $\underbrace{1101}_{D}\,\underbrace{1011}_{B}{}_2$ => $\boxed{DB_{16}}$

③ convert hex 9E to decimal

$\overbrace{1001}^{9}\;\overbrace{1110}^{E}$

1	0	0	1	1	1	1	0
×	×	×	×	×	×	×	×
128	64	32	16	8	4	2	1

=> 128 + 16 + 8 + 4 + 2 = 128 + 30

→ $\boxed{158_{10}}$

④ add binary 1101 + 0110 & then convert to hex

$$\begin{array}{r} {\scriptstyle 1\,1} \\ 1101 \\ +\,0110 \\ \hline 10011 \end{array}$$

=> 10011_2 $\underbrace{0001}_{1}\,\underbrace{0011}_{3}$ => $\boxed{13_{16}}$

Problem Set - B

HW 1 Clay Ramey

1) 6 bit 2's complement representation of decimal # -21

6 bits

32	16	8	4	2	1
x	x	x	x	x	x
0	1	0	1	0	1

21 - 16 = 5 ✓

5 - 4 = 1 ✓

1 - 1 = 0 ✓

→ 010101 1's complement => 101010

$$\begin{array}{r} 101010 \\ + \quad\quad 1 \\ \hline 101011 \end{array}$$ 2's complement

ans → $\boxed{101011_2}$

2) convert decimal -51 to binary 8 bit 2's complement to hexadecimal

8-bits

128	64	32	16	8	4	2	1
x	x	x	x	x	x	x	x
0	0	1	1	0	0	1	1

51 - 32 → 19 ✓

19 - 16 → 3 ✓

3 - 2 → 1 ✓

1 - 1 → 0 ✓

$00110011_2 \xrightarrow{\text{1's complement}} 11001100_2$

↳ 2's complement

$$\begin{array}{r} 11001100 \\ + \quad\quad\quad 1 \\ \hline 11001101_2 \end{array}$$

Binary 8-bit 2's complement -51

=> $\underbrace{1100}_{C}\underbrace{1101}_{D}{}_2$

ans → $\boxed{CD_{16}}$

3) convert 8-bit 2's complement converted to hex D6, convert to decimal

$\overbrace{1101}^{D}\ \overbrace{0110}^{6} \xrightarrow{\text{2's complement}}$

$$\begin{array}{r} 00101001 \\ + \quad\quad\quad 1 \\ \hline 00101010 \end{array}$$

0	0	1	0	1	0	1	0
x	x	x	x	x	x	x	x
128	64	32	16	8	4	2	1

32 + 8 + 2 = 42_{10}

ans → $\boxed{42_{10}}$

HW 1 Clay Ramey

Problem set B continued.

④ compute 10101 − 00011 where operands are two's complement numbers. Give answer in decimal

$$00011 \xrightarrow{1's} \begin{array}{r} 11100 \\ +\quad 1 \\ \hline 11101 \end{array} \Big\} 2's$$

$$\begin{array}{r} 16\ 8\ 4\ 2\ 1 \\ 10101 \\ 00011 \end{array} \qquad \begin{array}{r} 21 \\ -3 \\ \hline 18 \end{array}$$

$$\Rightarrow \begin{array}{r} 1\,1\;\;1\;\; \\ 10101 \\ +\ 11101 \\ \hline \cancel{1}\,10010 \end{array} \Rightarrow \begin{array}{r} 16\ 8\ 4\ 2\ 1 \\ 10010 \end{array}$$

$$\xrightarrow{2's} \begin{array}{r} 01101 \\ +\quad 1 \\ \hline 01110 \end{array}$$

$$\begin{array}{r} 8\,4\,2\;\; \\ 8+4+2 \end{array} = 14 \rightarrow \boxed{-14_{10}}$$

HW 1 Clay Ramey

Problem Set C

1)

op	rs	rt	rd	shamt	funct
0	9	8	10	0	34

op = 0 & funct = 34 → subtract (sub)

rs = 9 = $t1   rt = 8 = $t0   rd = 10 = $t2

rs - rt → $\boxed{\text{sub \$t2 \$t1 \$t0}}$

2) var h is associated w/ register $s2, array A is in $s3, MIPS assembly code for: A[12] = h + A[9]

```
lw  $t0, 36($s3)
add $t0, $s2, $t0
sw  $t0, 48($s3)
```

3) variable f, g, h, i, j equals $s0, $s1, $s2, $s3, $s4 whats the MIPS assembly code for: f = (g+h) - (i+j)

```
add $t0, $s1, $s2
add $t1, $s3, $s4
sub $s0, $t0, $t1
```

1 2 4 8 16 32 64 128

HW 1 Clay Ramey

Problem Set C continued!

(4) \$s1 has the base arr A, \$s3 corresponds to h, the statement A[200] = h + A[200] compiles to:

```
lw  $t0, 800($t1)
add $t0, $s3, $t0
sw  $t0, 800($t1)
```

→

op	rs	rt	rd	adr	func
35	9	8	800		
0	20	8	8	0	32
43	9	8	800		

machine lang code:

op	rs	rt	rd	adr	func
100011	01001	01000	0011 0010 0000		
000000	10100	01000	01000	00000	100000
101011	01001	01000	0011 0010 0000		

Bonus problem - overflow

(1) $64+64 \rightarrow$ 01000000 + 01000000 = 10000000 → $10000000_2 = 128_{10}$, CORRECT

(2) $-127+30 \rightarrow 30-127 \rightarrow$ 00011110 − 01111111 $\xrightarrow{2's}$ 10000000 + 1 = 10000001

00011110 + 10000001 = $\boxed{1}$0011111 → negative #

$\xrightarrow{2's}$ 01100000 + 1 = 01100001

128 64 32 16 8 4 2 1

64+32+1

96+1

→ $01100001_2 = -97_{10}$ CORRECT

(3) $-1-127 \rightarrow$ 00000001 − 01111111

00000001 $\xrightarrow{2's}$ 11111110 + 1 = 11111111

01111111 $\xrightarrow{2's}$ 10000000 + 1 = 10000001

11111111 + 10000001 = 1 10000000

*ignore 9th bit

negative # bc msb = 1

10000000 $\xrightarrow{2's}$ 01111111 + 1 = 10000000

→ $10000000_2 \Rightarrow -128$ CORRECT

(4) $38-40 \rightarrow$ 00100110 − 00101000 $\xrightarrow{2's}$ 11010111 + 1 = 11011000

00100110 + 11011000 = 11111110 $\xrightarrow{2's}$ 00000001 + 1 = 00000010

*MSB = 1 → neg #

→ $00000010_2 \Rightarrow -2$ CORRECT