Compsci 210 S1 2013 - Assignment One

Ksenia Kovaleva
ID: 4716583

1. 101 ten

$$\begin{array}{rl} & 101 \\ - & 64 \;\; (2^6) \\ \hline & 37 \\ - & 32 \;\; (2^5) \\ \hline & 5 \end{array}$$

5 = 4 (2^2) + 1 (2^0)

0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									4	2			4		1

4 + 2 = 6; 4 + 1 = 5 → 6, 5

1 A) 16 bit 2's comp: 0000 0000 01100101$_2$
Hex: 65_{16}
16 bit S.M.: 0000 0000 0110 0101$_2$
Hex: 65_{16} (Number is positive so signed magnitude rep. does not differ from 2's comp & hex rep also stays the same)

B) -101_{ten}

101 is 0000 0000 0110 0101 (from part A)

1's comp: 1111 1111 1001 1010
+ 1
2's comp: 1111 | 1111 | 1001 | 1011
F F 9 B (11 = B)

2's comp working

1000 | 0000 | 0110 | 0101 (from part A, w/ most sig. bit changed to rep. negative sign)
8 0 6 5

signed-magnitude working

1B) 16-bit 2's comp: 1111 1111 10011011$_2$
Hex: FF9B$_{16}$
16-bit S.M: 1000 0000 0110 0101$_2$
Hex: 8065$_{16}$

c) -32,000

Range = $[-2^{15}, 2^{15}-1]$
$[-32,768, 32,767]$

∴ -32,000 can be represented in 16 bits

32,000 in binary is:

2^8	256
2^9	512
2^{10}	1024
2^{11}	2,048
2^{12}	4,096
2^{13}	8,192
2^{14}	16,384
2^{15}	32,768

32,000
- 16,384 — 2^{14}
15 616
- 8,192 — 2^{13}
7,424
- 4,096 — 2^{12}
3,328
- 2,048 — 2^{11}
1,280
- 1 024 — 2^{10}
256 — 2^8

0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

1's comp: 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1
+ 1

1 1 1 1 1 1 1 1 0 C

2's comp (-32,000) → 1 0 0 0 | 0 0 1 1 | 0 0 0 0 | 0 0 0 0

8 2 3

(8) (3) (0) (0)

SM. (Change sign of 32,000 binary rep)

→ 1 1 1 1 | 1 1 0 1 | 0 0 0 0 | 0 0 0 0$_2$

A = 10
B = 11
C = 12

1C) 16 bit 2's comp: 1000 0011 0000 0000_2
Hex: 8300_{16}
16-bit S.M: 1111 1101 0000 0000_2
Hex: $FD00_{16}$

2.
```
  1111 1011
+ 0100 0000
  1000 0000   (Carries Line)
 10011 1011   (Result)
```

unsigned 8-bit

128 64 32 16 8 2 1
1111 1011 → $128+64+32+16+8+2+1 = 251_{10}$
0100 0000 → 64
64

251 + 64 = 315 (Answer out of range of 8 bits - overflow is occuring)
255 < 315
also evident from carry out into 9th bit in the Result. (left most bit)

128 64
10011 1011 (Result)
256 32 16 8 2 1
$= 256 + 32 + 16 + 8 + 2 + 1$
$= 315_{10}$

2A) Unsigned 8-bit binary
Answer: 315_{ten}
Overflows occurs.

B) 2's complement

```
  1111 1011
+ 0100 0000
  1000 0000   (Carries line)
 1 0011 1011  (Result)
```

2's comp ignores carry out;
Answer is $0011\ 1011_2$ → 59_{10}
32 16 8 2 1
48 + 11

Check with 0100 0000 = 64_{10}
2^6

Can check w/

```
  00000101  (5)
+ 11111011
  11111110  C
 100000000  R
```

5 + -5 = 0
∴ 11111011 is -5

(Operand is -5 in 2's comp)

So $64 + -5 = 59$

→ consistent with binary interpretation

No overflow is occuring because the leading bits of operands are different (0 & 1) and we know 59 can be represented in 8 bits.

2B 2's complement 8-bit binary

Answer: 59_{ten}.

<u>No</u> overflow.

c) 1's complement

1111 1011

0000 0100 = 4 (1's comp)

∴ 1111 1011 rep. -4_{ten}

0100 0000 → rep 64_{ten}

Sum will be 60_{ten}.

addition with 2's complement

4 = 00000100

−4 = 11111011

+1

11111100 ← −4 in 2's complement

So
```
  11111100
+ 01000000
 10000000  C
 1|00111100  R
     32 16 8 4
```

Result = $32 + 16 + 8 + 4 = 60_{ten}$ (Consistent with base 10 addition)

No overflow because most sig. bit of operands are different & 60 can be represented in 8 bits.

Another way to check/perform the addition is using the 1's complement method:

```
   1111 1011      (-4)
+  0100 0000      (64)
  ----------
  1 0000 0000     Carry Line
  ----------
 (1)0011 1011     <- Intermediate Result (incorrect)
+ - - - - ->(1)   <- add end-carry
  ----------
          110     Carry Line
  ----------
   0011 1100      Result
     | | | |
    32 16 8 4
```

Result: $32 + 16 + 8 + 4 = 60_{ten}$

2C. 1's complement 8-bit binary
Answer: 60_{ten}
No overflow.

D) Signed-magnitude

```
 1   1   1   1     1   0   1   1
 |   |   |   |     |       |   |
-ve  64  32  16  + 8   +   2 + 1
```

$\ominus_{ve}$ $64 + 32 + 16 + 8 + 2 + 1 \rightarrow -123_{ten}$

0100 0000 $\rightarrow$ 64_{ten}

$-123 + 64 = -59$ (expected result)

(Using 2's complement to perform addition)

$-123 \rightarrow$ 123 in 2's comp:

```
  123
- 64  — 2^6
  ---
   59
- 32  — 2^5
  ---
   27
- 16  — 2^4
  ---
   11
   8  — 2^3
```

$123 = \underset{7}{0}\ \underset{6}{1}\ \underset{5}{1}\ \underset{4}{1}\ \underset{3}{1}\ \underset{2}{0}\ \underset{1}{1}\ \underset{0}{1}$

123 0 1 1 1 1 0 1 1

1's comp : 1 0 0 0 0 1 0 0

+ 1

2's comp : 1 0 0 0 0 1 0 1 (-123)

```
    1 00 00 1 01
+   0 10 00 0 00
    ------------
    0 00 00 0 00   C
    ------------
    1 10 00 1 01   R
```

Interpreting the (negative) result of 2's comp addition

```
         11000 101
~~1's comp~~ 00 111 010   ← (1's comp)
               + 1
         00 111 011   ← (2's comp)
            32 16 8  2 1
```

(-) 59
− 32
= 27
− 16
= 11
− 8
= 3 → 2, 1

(10111011) in signed magnitude

32 + 16 + 8 + 2 + 1 = 59_{ten}

∴ Original number represented by result is -59_{ten}.

No overflow is occuring - operands have opposite signs (0 & 1 in most sig bit) & -59 can be represented in 8 bits in S.M.

→ Overflow is indicated if signs of operands are the same and the sign of the result is different (in 2's complement method)

2D. Signed-magnitude 8-bit binary

Answer: -59_{ten}

No overflow.

3.
```
  1111 1011
- 0100 0000
  0000 0000   Borrows Line
  1011 1011
```
128, 32, 16, 8, 2, 1

→ 1111 1011 (128 64 32 16 8 2 1)
0100 0000 (64)

(251)
− (64)
187 (expected)

128 + 32 + 16 + 8 + 2 + 1 = 187_{ten} (187 can be represented in 8 bits no overflow)

A) Unsigned binary
Answer: 187_{ten}
No overflow.

B) 1111 1011
→ 0000 0100
+ 1

0 000 0101 (2's comp) 5

$\underbrace{0101}_{5}$

Original Number (1111 1011) is −5 in 2's comp

in decimal = −5 − 64

0100 0000 (64)
1011 1111 (1's comp)
+ 1

0111 1110 Carries Line
1100 0000 (−64) 2's comp.

So −5 + −64 →
```
   1111 1011
 + 1100 0000
 1 1100 0000   Carries Line
 1 1011 1011   Result
```

1011 1011
0100 0100 1's comp
+ 1

0100 0101 (69) 2's comp = original binary number is −69
64 + 4 + 1

B) 2's complement 8-bit binary
Answer: -69_{ten}
No overflow. (Signs of operands are the same as the result and final answer can be expressed in 8-bit 2's complement).

c) 1111 1011 ← 0000 0100 (1st operand is −4 in 1's complement)
 4
− 0100 0000 (64)

−4 − 64 same as −4 + −64
−64 = 1011 1111 (in 1's comp)

Using 1's complement method of addition:

```
   1111 1011
 + 1011 1111
 -----------
 1 1111 1110   Carries Line
 1 1011 1010   (intermediate result)
          +1   add back end carry
 -----------
   1011 1011   Result
```

↳ (1's comp) 0100 0100 = 68
 64 + 4

Original result is ⊖ve
So Answer is −68.
(Consistent with decimal arithmetic of −4 + −64 or −4 − 64 = −68)

1's complement 8-bit binary
Answer: -68_{ten}
No overflow.
→ (Signs of operands are the same and the final answer is the same sign. The result can also be represented in 8-bit 1's complement).

3 D) Signed magnitude

1111 1011 → ⊖ (64+32+16+8+2+1) ⇒ −123

0100 0000 → 64

−123 − 64 = −187$_{ten}$ ← out of range

Same as −123 + −64

(Using 2's complement addition to perform arithmetic)
/show overflow

−64 = 1100 0000 (in 2's complement from part B)

−123 = 123 − 64 (2^6) = 59 − 32 (2^5) = 27 − 16 (2^4) = 11 − 8 (2^3) = 3 → 2 (2^1), 1 (2^0)

0 1 1 1 1 0 1 1 (123)
1 0 0 0 0 1 0 0 (1's comp)
\+ 1
1 0 0 0 0 1 0 1 (−123 2's comp)

```
  10000101
+ 11000000
 100000000   Carries line
 101000101
```

The 2's complement arithmetic indicates overflow – because the sign of the result is not the same as the sign of the operands. The answer (as calculated by base 10 arithmetic) is also clearly out of range of 8-bit signed-magnitude binary. The range is [−128, 127] and so −187 cannot be expressed in 8 bits. It requires a minimum of 9 bits (11011 1011 – in ~~unsigned binar~~ signed magnitude binary)

4 A.

$$\begin{array}{lr} & 1111\ \ 0000 \\ \text{AND} & 0101\ \ 0110 \\ \hline & \underbrace{0101}_{5}\ \ \underbrace{0000}_{0} \end{array}$$

4 A. Binary rep: $0101\ 0000_2$
Hex: 50_{16}

10 = A
11 = B
12 = C
13 = D
14 = E
15 = F

B)

$$\begin{array}{lr} & 1101\ \ 1110 \\ \text{OR} & 0101\ \ 0111 \\ \hline & \underbrace{\overset{8}{1}\,\overset{4}{1}\,0\,\overset{1}{1}}_{13\,=\,D}\ \ \underbrace{1111}_{F} \end{array}$$

4B) Binary rep: 1101 1111
Hex: DF

c)

$$\begin{array}{lr} & 1101\ \ 0010 \\ \text{XOR} & 0101\ \ 0111 \\ \hline & \underbrace{1000}_{8}\ \ \underbrace{0101}_{5} \end{array}$$

4C) Binary rep: 1000 0101
Hex: 85

D) NOT

$$\begin{array}{r} 1101\ \ 0010 \\ \hline \underbrace{0010}_{2}\ \ \underbrace{\overset{8}{1}\,\overset{4}{1}\,0\,\overset{1}{1}}_{13\,=\,D} \end{array}$$

4D) Binary rep: 0010 1101
Hex: 2D

5. -31_{ten}

A) 31 in 2 comp →

$$\begin{array}{r} 31 \\ -16 \quad 2^4 \\ \hline 15 \\ -8 \quad 2^3 \\ \hline 7 \\ -4 \quad 2^2 \\ \hline 3 \end{array}$$

$3 \to 2 \;(2^1),\; 1 \;(2^0)$

$\underset{5}{0}\ \underset{4}{1}\ \underset{3}{1}\ \underset{2}{1}\ \underset{1}{1}\ \underset{0}{1}$ = 31 in 2's comp

100 000 ← 1's comp
+ 1

100001 ← 2's comp (-31)

5A) 100001

B) 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1

c)

$\underbrace{1\,1\,1\,1}_{F}\ \underbrace{1\,1\,1\,1}_{F}\ \underbrace{\overset{}{\underset{8}{1}}\,\underset{4}{1}\,\underset{2}{1}\,0}_{E\,(14)}\ \underbrace{0\,0\,0\,1}_{1}$

c) Hex : FFE1

6.A) $\ldots a_3 \cdot 5^3 + a_2 \cdot 5^2 + a_1 \times 5^1 + a_0 \times 5^0$

To represent 50 → smallest power of 5 less than 50
→ is 5^2

$$\begin{array}{r} 50 \\ -25 \quad 5^2 \\ \hline 25 \quad 5^2 \end{array}$$

so there are 2 lots of 5^2 in 50.

Therefore 50 is represented as $\underset{(5^2)}{2}\ \underset{(5^1)}{0}\ \underset{(5^0)}{0}$.

$50_{ten} = 200_{five}$

6B) 44444444 (4 is the largest 'figit' available)

6 C) $4\times5^7 + 4\times5^6 + 4\times5^5 + 4\times5^4 + 4\times5^3 + 4\times5^2 + 4\times5^1 + 4\times5^0$

$= 390,624_{ten}$

Equivalent to the max value of range of figits in 8 bits so (5^8-1) which represents maximum possible representations of unsigned integers in 8 figits ~~w~~ minus 1 = taking away the one representation of 0.

7A) $1\cdot5^2 + 2\cdot5^1 + 3\cdot5^0$

$= 25 + 10 + 3$

$= 38_{ten}$

A 1) $123_{five} = 38_{ten}$

8-bit unsigned binary:

$$38 - 32 = 6 \quad (32 = 2^5)$$

$$6 = 4 + 2 \quad (4 = 2^2,\ 2 = 2^1)$$

0	0	1	0	0	1	1	0
7	6	5	4	3	2	1	0

2 6

A2) 8 bit - unsigned binary: 00100110_2

A3) Hex: 26_{16}

7 B) 4_{five}

$\rightarrow 4\times5^0 = 4_{ten}$

00 00|0100

4

B1) $4_{five} = 4_{ten}$

2) 8-bit unsigned rep: 00000100_2

3) Hex: 4_{16}

7c) 1444_{five}

$= 1 \times 5^3 + 4 \times 5^2 + 4 \times 5^1 + 4 \times 5^0$

$125 + 100 + 20 + 4$

$225 \quad 245 \quad 249$

$= 249_{ten}$

$$
\begin{array}{r l}
 & 249 \\
- & 128 \; — \; 2^7 \\ \hline
 & 121 \\
- & 64 \; — \; 2^6 \\ \hline
 & 57 \\
- & 32 \; — \; 2^5 \\ \hline
 & 25 \\
- & 16 \; — \; 2^4 \\ \hline
 & 9 \\
- & 8 \; — \; 2^3 \\ \hline
 & 1 \; — \; 2^0
\end{array}
$$

$\overbrace{1 \; 1 \; 1 \; 1}^{15} \; | \; \overbrace{1 \; 0 \; 0 \; 1}^{8 \quad 1}$

7	6	5	4	3	2	1	0
1	1	1	1	1	0	0	1

F 9

7C 1) $1444_{five} = 249_{ten}$
2) $1111\ 1001_2$ (unsigned 8-bit rep)
3) Hex: $F9_{16}$

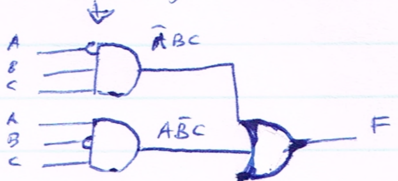
8) F is 1 when

A	B	C	F
0	1	1	1
1	0	1	1
1	1	0	1

8A) ((NOT A) AND B AND C) OR (A AND (NOT B) AND C) OR (A AND B AND (NOT C))

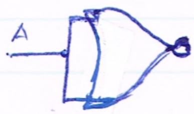
$\rightarrow \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} = F$

(working - NOT final answer!!) –

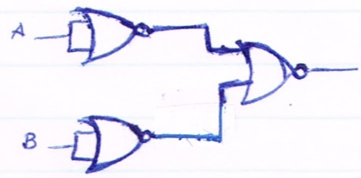


(Using this basic structure to convert the gates - more working on next page)

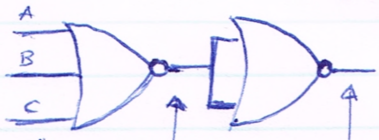
NOT w NOR is



AND w NOR



OR w NOR

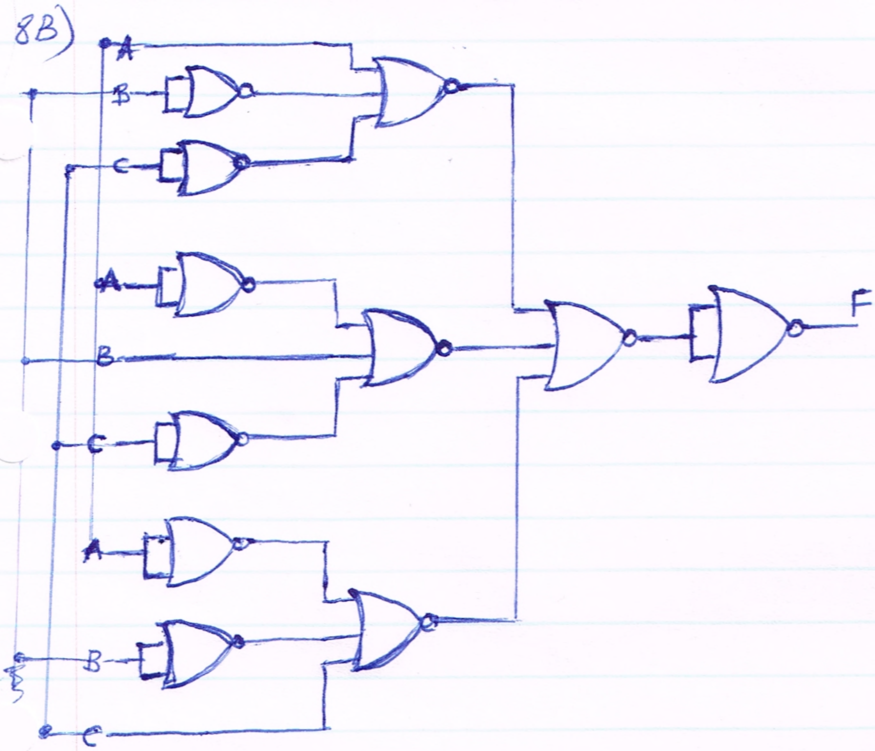


if any true ⇒ returns 0 ⟶ will invert to 1.

* NOR gate only outputs 1 if <u>all</u> inputs are 0.

<u>Logic diagram using only NOR gates:</u>

8B)



9 A) unsigned integer

$0x45 \rightarrow 45_{16} \Rightarrow \underbrace{0100}_{4}\,\underbrace{0101}_{5}$

7 bits = 1000101

$64 + 4 + 1$ $64 + 4 + 1 = 69_{ten}$

8 bits = 0100 0101 = 69_{ten}

64 4 1

A) 7 bits = 69_{ten} } only 69_{ten} is represented for unsigned

9B) 2's complement

7 bits = 1000101

1's comp = 0111010

$+ \quad 1$

2's comp 0111011 $= 59_{ten}$

32 16 8 2 1

7 bits in 2's comp $= -59_{ten}$

8 bits in 2's comp: 01000101_{two}

64 4 1

$= 69_{ten}$

1B) 0x45 could represent -59_{ten} in 7 bits 2's complement or 69_{ten} in 8-bits 2's complement.

c) signed magnitude

$1\underbrace{000101}_{5} = -5_{ten}$ (7 bits)

-ve

$01000101 = 69_{ten}$ (8 bits)

9C) 7 bits signed magnitude $= -5_{ten}$

8 bits signed magnitude $= 69_{ten}$

9D $45_{hex} = E$ (using ASCII table)

10C) $42C8\ 0000_{hex}$ $\quad C = 12$

$\boxed{0}\boxed{100,0010,1}100,1000,0000,0000,0000,0000$

4 2 C 8 0 0 0 0

sign = 0 = +ve

2^7

exponent = 10000101 − bias

$(128 + \underbrace{5}) - (127$

$= 133 - 127$

$= 6$

$= 2^6$

$$\text{Fraction} = .\underset{2^{-1}}{1}00\underset{2^{-4}}{1}000\ldots$$

(.5, .25, .125, .0625)

same as $\frac{1 \times 8}{2 \times 8} + \frac{1}{16} = \frac{8}{16} + \frac{1}{16} = \frac{9}{16}$

or $0.5 + 0.0625$

$= 0.5625$ (equivalent to $\frac{9}{16}$)

$\Rightarrow$ add implicit 1 $= 1.5625$

$\Rightarrow 1.5625 \times 2^6$

$= 100_{ten}$

D) $3F40\ 0000_{hex}$

$$0\,|\boxed{011,1111,0}|100,0000,0000,0000,0000,0000$$

(3) (F) (4) (0) (0) (0) (0) (0)

+ve sign

exponent $= \underset{64\ 32\ 16\ 8\ 4\ 2}{01111110}$ $-$ bias (127)

$\Rightarrow 64 + 32 + 16 + 8 + 4 + 2 - (127)$

$\Rightarrow 126 - 127 = -1$

exponent: 2^{-1}

Fraction part $= .100\ldots\ldots$

2^{-1} or .5

$\rightarrow 1.5$

$\Rightarrow 1.5 \times 2^{-1}$ (same as $1.5 \times .5$ or $1.5 \div 2$)

$= 0.75_{ten}$

E) $4049\ 0FDB_{hex}$

$$0\,|\boxed{100\ 0000\ 0}|100\ 1001\ 0000\ 1111\ 1101\ 1011$$

(4) (0) (4) (9) (0) (F) (D) (B)

(13) (11)

sign = +ve

exponent $= \underset{2^7}{1}00\ 000\ 00$ $-$ bias

$= 128 - 127$

Fraction part

100 1001 0000 1111 1101 1011

-1 -4 -7 -12 -13 -14 -15 -16 -17 -19 -20 -22 -23

$$= 2^{-1} + 2^{-4} + 2^{-7} + 2^{-12} + 2^{-13} + 2^{-14} + 2^{-15} + 2^{-16} + 2^{-17} + 2^{-19} + 2^{-20} + 2^{-22} + 2^{-23}$$

Answer = 0.57079637

Total fraction is 1.57079637

$\Rightarrow 1.57079637 \times 2^{1}$

$= 3.14159274_{ten}$

11. A) 3.5_{ten} is 011.100

01.1100×2^{1}

sign = 0 (positive)

exponent = 1 + 127 = 128

0100 0000 0110 0000 0000 0000 0000 0000

4 0 6 0 0 0 0 0

Hex = 40600000_{hex}

$\rightarrow 20_{ten}$ (rep) $\rightarrow$ (sign is ⊕ve)

B) 010100_{two} $\rightarrow$ 010100.

$\rightarrow 01.0100 \times 2^{4}$

sign = 0 (⊕ve)

exponent = 4 + 127 = 131

131 to binary $\rightarrow$ 131 − 128 (2^7) = 3 = 2 (2^1) + 1 (2^0)

10000011

Fraction part is 0100...

$\rightarrow$ 0100 0001 1010 0000 0000 0000 0000 000

11c) 1021_{ten}

$$\begin{array}{rl} 1021 & \\ -512 & 2^9 \\ \hline 509 & \\ -256 & 2^8 \\ \hline 253 & \\ -128 & 2^7 \\ \hline 125 & \\ -64 & 2^6 \\ \hline 61 & \\ -32 & 2^5 \\ \hline 29 & \\ -16 & 2^4 \\ \hline 13 & \\ -8 & 2^3 \\ \hline 5 & \end{array}$$

$5 \to 4 \ (2^2),\ 1 \ (2^0)$

Binary $\underset{9}{1} \,|\, \underset{8}{1}\ \underset{7}{1}\ \underset{6}{1}\ \underset{5}{1}\ \underset{4}{1}\ \underset{3}{1}\ \underset{2}{1}\ \underset{1}{0}\ \underset{0}{1}.$

exponent = 2^{9}

$9 + 127 = 136 \longrightarrow$

$$\begin{array}{rl} 136 & \\ -128 & 2^7 \\ \hline 8 & 2^3 \end{array}$$

sign = +ve (0)

1 0 0 0 1 0 0 0 (136)

(Putting it all together into IEEE format:)

$$\underbrace{0\,1\,0\,0}_{4}\ \underbrace{0\,1\,0\,0}_{4}\ \underbrace{0\,1\,1\,1}_{7}\ \underbrace{1\,1\,1\,1}_{F}\ \underbrace{0\,1\,0\,0}_{4}\ \underbrace{0\,0\,0\,0}_{0}\ \underbrace{0\,0\,0\,0}_{0}\ \underbrace{0\,0\,0\,0}_{0}$$

Hex = 447F4000

11D) -0.40625 (same as $\frac{13}{32}$)

sign = −ve (1)

$$\frac{13}{32} - \underbrace{\frac{1}{32}}_{2^{-5}} = \frac{12}{32} = \frac{3}{8} - \underbrace{\frac{1}{8}}_{2^{-3}} = \frac{2}{8} = \underbrace{\frac{1}{4}}_{2^{-2}}$$

OR equivalently

$$\begin{array}{rl} .40625 & \\ -.25000 & 2^{-2} \\ \hline .15625 & \\ -.12500 & 2^{-3} \\ \hline .03125 & 2^{-5} \end{array}$$

2^{-1}	0.5
2^{-2}	0.25
2^{-3}	0.125
2^{-4}	0.0625
2^{-5}	0.03125

Binary form (fraction part) $.\underset{1}{0}\ \underset{2}{1}\ \underset{3}{1}\ \underset{4}{0}\ \underset{5}{1}$ (normalize) (1.101)

moves 2 places right

so exponent = 2^{-2}

$\rightarrow -2 + 127 = 125$

125 in binary = 125
$-\,64 \leftarrow 2^6$
61
$-\,32 \leftarrow 2^5$
29
$-\,16 \leftarrow 2^4$
13
$-\,8 \leftarrow 2^3$
5
$-\,4 \leftarrow 2^2$
$1 \leftarrow 2^0$

0	1	1	1	1	1	0	1
7	6	5	4	3	2	1	0

IEEE Floating Point Notation

1 011 | 1110 | 1 101 | 0000 | 0000 | 0000 | 0000 | 0000

B E D 0 0 0 0 0

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Hex: $BED00000_{16}$

12. A) $2^n = 2^5$ unique combinations = 32
B) 2^n where n is all possible combinations for 5 variables
$\rightarrow 2^{32} = 4294967296$ unique functions exist.

13. Majority gate for 5 variables / inputs = will only be true if 3 or more inputs are true. Therefore minimum it must detect is 3 inputs true.

Possible combinations of at least 3 inputs = 1.

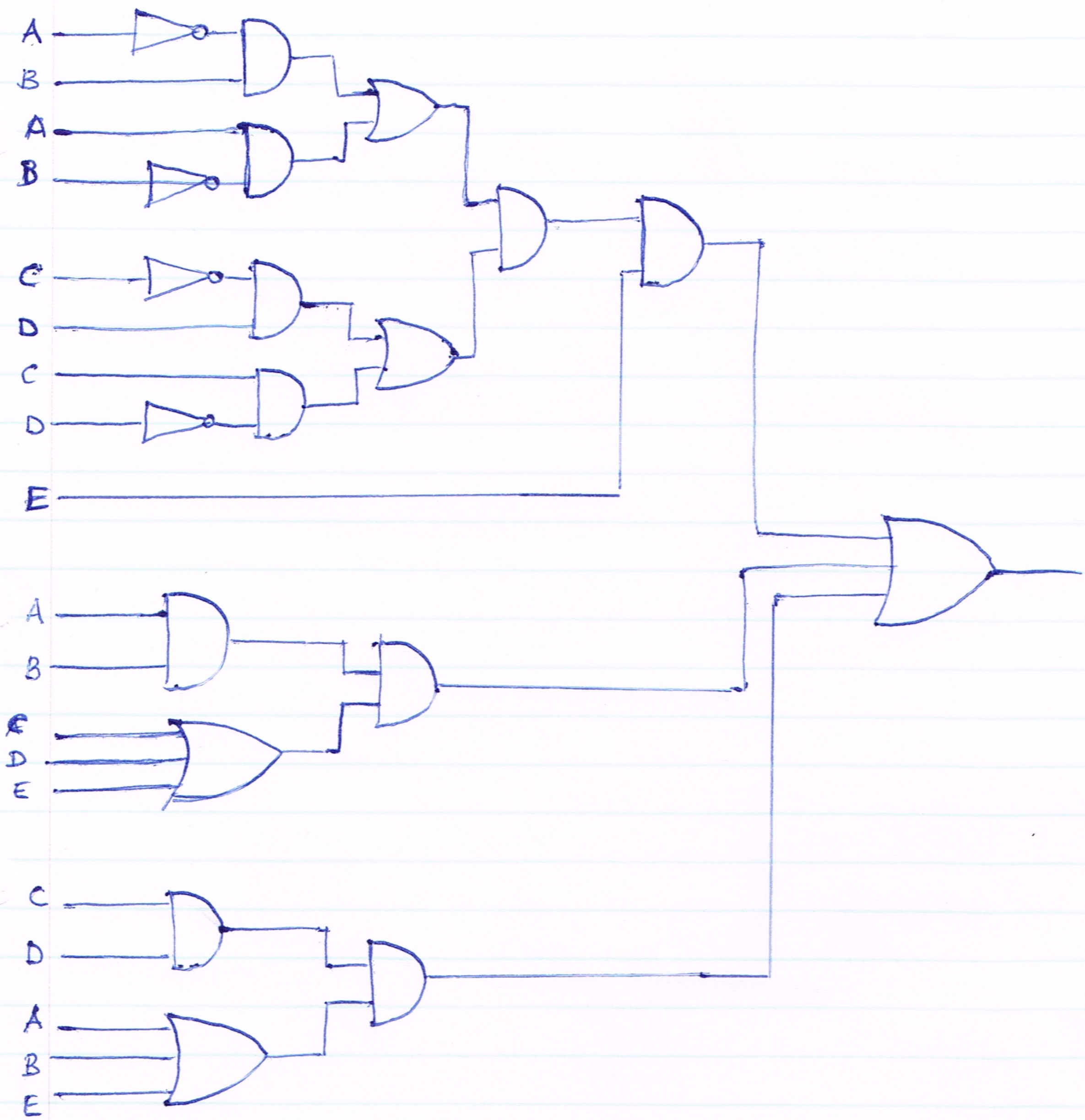
ABC
ABD
ABE
ACD
ACE
ADE
BCD
BCE
BDE

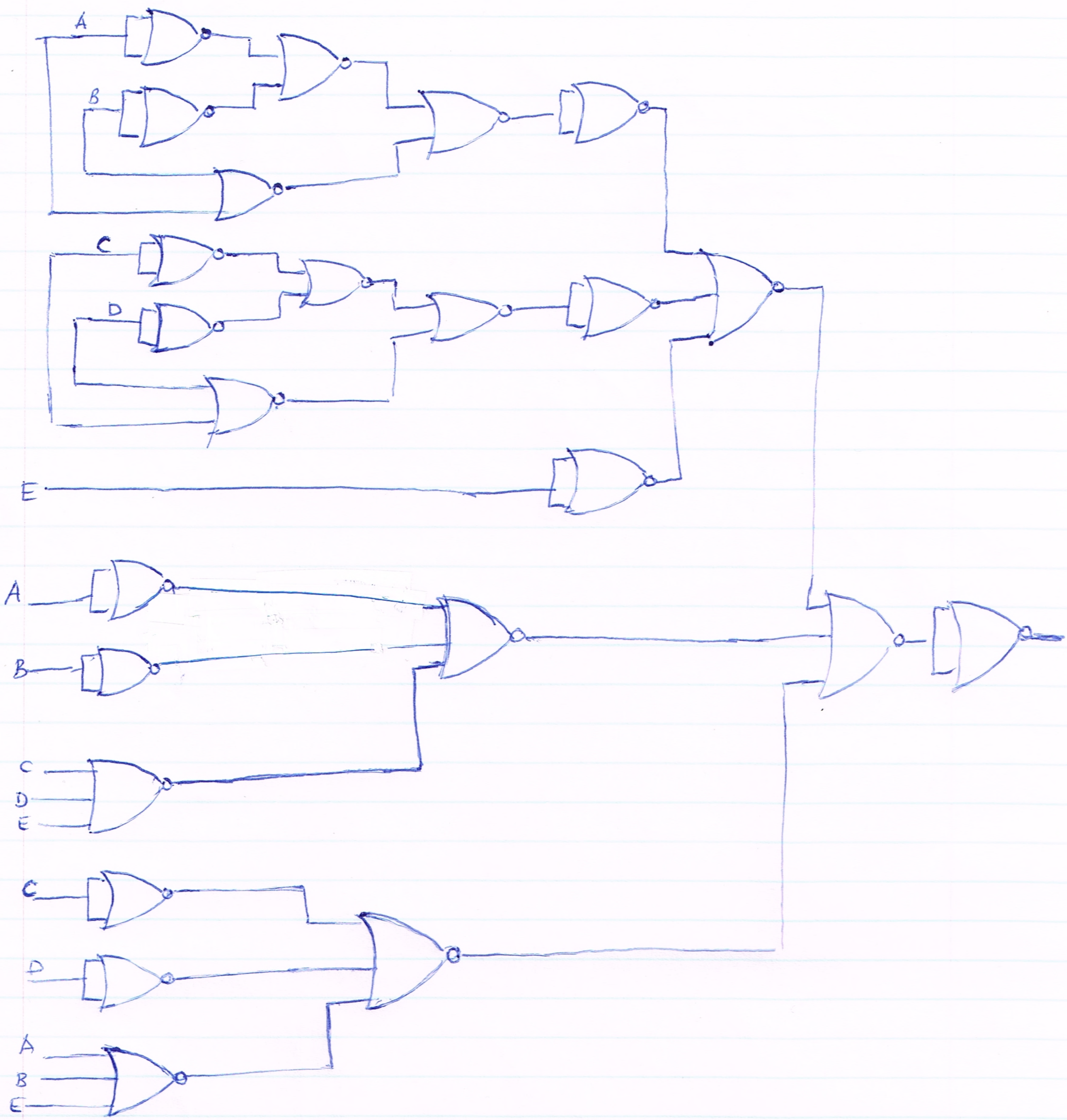
(A AND B) AND (C OR D OR E)

AB	C	CD	A	A	C	E
AB	D	CD	B	A	D	E
AB	E	CD	E	B	C	E
				B	D	E

(C AND D) AND (A OR B OR E)

(A XOR B) AND (C XOR D) AND E


A
B
A
B
C
D
C
D
E
A
B
C
D
E
C
D
A
B
E


A
B
C
D
E
A
B
C
D
E
C
D
A
B
E

14 A) All off = 0000
display '0' = 0001
display '1' = 0010
display '2' = 0011
display '3' = 0100
display '4' = 0101
display '5' = 0110
display '6' = 0111
display '7' = 1000
display '8' = 1001
display '9' = 1010

Input	Current State				Next State			
On	S_3	S_2	S_1	S_0	S_3'	S_2'	S_1'	S_0'
0	X	X	X	X	0	0	0	0
1	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	0	0	1	0	1
1	0	1	0	1	0	1	1	0
1	0	1	1	0	0	1	1	1
1	0	1	1	1	1	0	0	0
1	1	0	0	0	1	0	0	1
1	1	0	0	1	1	0	1	0
1	1	0	1	0	0	0	0	0

B)

Current State				Output						
S_3	S_2	S_1	S_0	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	0
0	0	1	0	0	1	1	0	0	0	0
0	0	1	1	1	1	0	1	1	0	1
0	1	0	0	1	1	1	1	0	0	1
0	1	0	1	0	1	1	0	0	1	1
0	1	1	0	1	0	1	1	0	1	1
0	1	1	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	0	1	1

C) 11

15A) Location 3 hex: 0000_{hex}

Location 6 hex: $FED3_{hex}$

$$\underbrace{1111}_{F}\quad \underbrace{\overset{8\,4\,2}{1110}}_{E}\quad \underbrace{1101}_{D}\quad \underbrace{0011}_{3}$$

B) Location 0: $0001\ 1110\ 0100\ 0011$

$$\begin{array}{cccc} 0001 & 1110 & 0100 & 0011 \\ 2^{12} & 2^{11}\ 2^{10}\ 2^{9} & 64 & 3 \end{array}$$

1)

$2^{12}+2^{11}+2^{10}+2^{9}+2^{6}+2^{1}+2^{0}$

$4096 + 2048 + 1024 + 512 + 64 + 2 + 1$

$= 7,747_{ten}$ (location 0)

Location 1: 1111 0000 0010 0101

0000 1111 1101 1010 (1's comp)

\+ 1

0000 1111 1101 1011 (2's comp)

2048 1024 512 256 128 64 16 2 2 1

$(-)\left[2048 + 1024 + 512 + 256 + 128 + 64 + 16 + 8 + 2 + 1\right]$

$= -4059_{ten}$ (location 1)

B2) Location 4 as ASCII

$$0000\quad 0000\quad \underbrace{0110}_{6}\quad \underbrace{0101}_{5}$$

65_{hex} = e (in ASCII)

Location 7 Location 6

B3) 0000 0110 1101 1001 | 1111 1110 1101 0011

2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5} 2^{-6} 2^{-7} 2^{-16} 2^{-17} 2^{-19} 2^{-22} 2^{-23}

sign +ve exponent

exponent = 00001101

8 4 1

= 13 − 127

= −114 ⇒ 2^{-114}

fraction = 101 1001 1111 1110 1101 0011

1 3 4 7 8 9 10 11 12 13 14 16 17 19 22 23

$= 2^{-1} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12} + 2^{-13} + 2^{-14} +$
$+ 2^{-16} + 2^{-17} + 2^{-19} + 2^{-22} + 2^{-23}$

⇒ .703089118

⇒ $1.703089118 \times 2^{-114}$

B 4) location 0: 0001 1110 0100 0011

From part B1) ⇒ same as 16 bit 2's complement

unsigned integer = 7749_{ten}

location 1: 1111 0000 0010 0101

2^{15} 2^{14} 2^{13} 2^{12} 32 4 1

= $61,477_{ten}$ (unsigned integer, location 1)

15 C) 0001 | 1110 | 0 0100 0011

op-code (ADD) store (R7) op 1 (R1) op 2 (R3)

→ "Add the contents of register 1 (R1) to the contents of register 3 (R3) and store the result back into register 7 (R7).

15 D) Location 2 refers to the address 0100 or location 4. Location 4 contains the value 0000 0000 0110 0101

(4+32 + 5)

or 101_{ten}.