EC 2.101 - Digital Systems and Microcontrollers

Practice Sheet 1 (Lec 1 – Lec 9)

Q1. Base Conversions

a. $(5641)_7 = (5 \times 7^3) + (6 \times 7^2) + (4 \times 7^1) + (1 \times 7^0)$

$= 1715 + 294 + 28 + 1$

$= (2038)_{10}$

$2038 = (679 \times 3) + \mathbf{1}$

$679 = (226 \times 3) + \mathbf{1}$

$226 = (75 \times 3) + \mathbf{1}$

$75 = (25 \times 3) + \mathbf{0}$

$25 = (8 \times 3) + \mathbf{1}$

$8 = (2 \times 3) + \mathbf{2}$

$2 = (0 \times 3) + \mathbf{2}$

(The division should be continued till zero quotient)

Therefore, $(5641)_7 = (\mathbf{2210111})_{\mathbf{3}}$.

b. $(F3E612)_{16} = 1111|0011|1110|0110|0001|0010$

$= (\mathbf{111100111110011000010010})_{\mathbf{2}}$

$= 11|11|00|11|11|10|01|10|00|01|00|10$

$= (\mathbf{330332120102})_{\mathbf{4}}$

$= 111|100|111|110|011|000|010|010$

$= (\mathbf{74763022})_{\mathbf{8}}$

c. $(A4389)_{16} = (11 \times 16^4) + (4 \times 16^3) + (3 \times 16^2) + (8 \times 16^1) + (9 \times 16^0)$

$= 720896 + 16384 + 768 + 128 + 9$

$= (738185)_{10}$

$= 0111|0011|1000|0001|1000|0101$ (Converting each decimal to BCD)

$= (011100111000000110000101)_{BCD}$

$(100000101011000111000111)_{BCD} = 1000|0010|1011|0001|1100|0111$

The highlighted segment is not a valid BCD code. Hence, this number cannot be converted into decimal.

d. Refer Morris Mano Problem 1.9

Q2. Complements

a. $(+7634)_8 = (111110011100)_2$.
For the negative number, you just take the 2's complement.
However, this is supposed to be a 16-bit number. So, we add zeros first.
Hence, $(+7634)_8 = (0000111110011100)_2$ in 16-bit 2's complement representation.
We want $(-7634)_8$ – We take 2's complement of $(0000111110011100)_2$ which is $(1111000001100100)_2$. (One trick to easily calculate 2's complement is to flip the bits only to the *left* of the last '1' in the binary number).

Similarly, we convert the remaining numbers to 16-bit binary numbers in 2's complement representation.
$(-512)_{10} = (1111111000000000)_2$.
$(+4AF)_{16} = (0000010010101111)_2$.

Since $(011001100110)_2$ is positive (left-most bit is zero), we can just add zeros.
$(011001100110)_2 = (0000011001100110)_2$.

Now that we have all four numbers in 16-bit 2's complement representation, adding them is also a little tricky. The key here is to note the difference between carry out and overflow.

- When two positive numbers are added resulting in a carry out, it is overflow.
- Similarly, when two negative numbers are added resulting in a carry out, it is overflow.
- However, when a positive and a negative number are added, the carryout can be discarded and there is **no** overflow (Why?).

Keeping this in mind and adding numbers from the right,

$$\begin{array}{r} 0000010010101111 \\ \underline{0000011001100110} \\ 0000101100010101 \end{array}$$

Adding this to $(-512)_{10}$,

$$\begin{array}{r} 0000101100010101 \\ \underline{1111111000000000} \\ \mathbf{1}0000100100010101 \end{array}$$

We can discard this carry. A noteworthy point here is that the resulting number is positive. Hence, when we add this to $(-7634)_8$, the third rule applies again, if there is a carry.

$$\begin{array}{r} 0000100100010101 \\ \underline{1111000001100100} \\ 1111100101111001 \end{array}$$

As we can see, there is no carry, and the result is also a negative number, $(1111100101111001)_2$ or $(-1671)_{10}$.

b. Similar to the above question, we first convert all given numbers to 12-bit binary numbers in 2's complement representation.
$(+657)_9 = (001000011010)_2$,
$(-565)_7 = (111011011100)_2$,
$(100001000101)_{BCD} = (001101001101)_2$, and
$(1101010110)_2 = (111101011000)_2$.

Adding,

$$\begin{array}{r} 001000011010 \\ \underline{111011011100} \\ \mathbf{1}000011110110 \end{array}$$

Discarding the carry and adding,

$$\begin{array}{r} 000011110110 \\ \underline{001101001101} \\ 010001000011 \end{array}$$

Adding to the last number,

$$\begin{array}{r} 010001000011 \\ \underline{111101011000} \\ \mathbf{1}001110011011 \end{array}$$

Again, we can discard the carry. Hence the answer is $(001110011011)_2$, or $(923)_{10}$.

c. Converting the given numbers into 16-bit 2's complement binary representation,
$(1000100101010111)_{BCD} = (0010001011111101)_2$

$(7345)_8 = (0000111011100101)_2$

$(4FB)_{16} = (0000010011111011)_2$

All of these are positive numbers. Hence, we simply add them and report overflow if there's a final carry.

Carries

$$\begin{array}{r} 1\qquad\quad \\ 111111\ \ \ 11111\ \ \\ 0010001011111101 \\ 0000111011100101 \\ \underline{0000010011111011} \\ 0011011011011101 \end{array}$$

Hence, the answer is $(0011011011011101)_2$ or $(14045)_{10}$.

Q3. Boolean Expressions

a. The dual of $(xy' + x'y)$ is $(x + y') \cdot (x' + y)$.

Now, $(xy' + x'y)' = (xy')' \cdot (x'y)'$

$$= (x' + y) \cdot (x + y')$$

$$= (x + y') \cdot (x' + y)$$

Hence, the complement and dual of the given expression (XOR gate) are the same.

b. Simplifying the given expressions,

i. $x'y + yz + xz' + x'y' + xyz' = xz'(1 + y) + x'(y + y') + yz = xz' + x' + yz$

Applying distributive property,

$$xz' + x' + yz = (x + x') \cdot (z' + x') + yz$$

$$= z' + x' + yz$$

$$= (z' + y) \cdot (z' + z) + x'$$

$$= z' + y + x'$$

ii. $xyz + x'y'z + xy'z + xz + xzy' = xyz + y'z(x' + 1) + xz(1 + y')$

$$= xyz + y'z + xz$$

$$= xz(1 + y) + y'z$$

$$= xz + y'z$$

iii. $xy + xy' + x'y'z + xy'z' + xyz = xy(1 + z) + xy'(1 + z') + x'y'z$

$$= xy + xy' + x'y'z$$

$$= x(y + y') + x'y'z = x'y'z + x$$

Applying distributive property,

$$x'y'z + x = (x' + x) \cdot (y'z + x) = x + y'z$$

iv. $x'y + x'y'z + xyz' + xy + xy'z' = x'(y + y'z) + xy(z' + 1) + xy'z'$

$$= x'[(y + y') \cdot (y + z)] + xy + xy'z'$$

$$= x'y + x'z + xy + xy'z'$$

$$= y(x' + x) + x'z + xy'z' = y + x'z + xy'z'$$

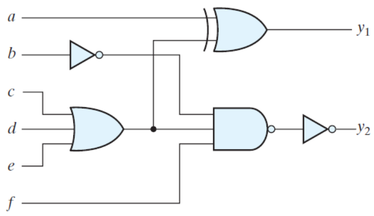
Applying distributive property,

$$
\begin{aligned}
y + xy'z' + x'z &= (y + x) \cdot (y + y'z') + x'z \\
&= (y + x) \cdot (y + y') \cdot (y + z') + x'z \\
&= (y + x) \cdot (y + z') + x'z \\
&= y + yz' + xy + xz' + x'z \\
&= y(1 + x + z') + xz' + x'z \\
&= y + x'z + xz'
\end{aligned}
$$

Note: This sheet was given before k-maps were taught. The idea behind was to make you understand how far expressions can be simplified by hand, and some more by applying laws cleverly. K-maps make this process that much easier.

Q4. Logic Gates

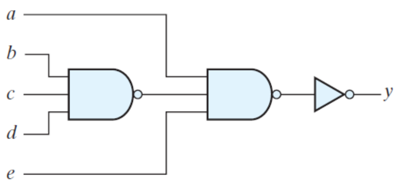
a.



$$
\begin{aligned}
y_1 &= a \oplus (c + d + e) \\
&= a'(c + d + e) + (c + d + e)'a \\
&= a'c + a'd + a'e + ac'd'e'
\end{aligned}
$$

$$
\begin{aligned}
y_2 &= ((b'(c + d + e)f)')' \\
&= b'cf + b'df + b'ef
\end{aligned}
$$

b.



$$
\begin{aligned}
y &= ((a(bcd)'e)')' \\
&= ae(b' + c' + d') = ab'e + ac'e + ad'e
\end{aligned}
$$