COMP2120 Computer Organisation
24/25 Semester 2
Assignment 1

1. Write down the logical expression $P = f(A,B,C)$ corresponding to the following truth table (simplification of logical expression is not required):

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Solution: Since there are less rows with $P = 0$ than rows with $P = 1$, it is easier to work by Product-of-Sums. First, identify the rows with $P = 0$, which are $(0,0,1)$, $(0,1,0)$ and $(1,0,0)$. For each of the rows, construct a product of all the variables, such that the term equals 1, then apply NOT to the term to invert it to 0. This gives:

$$(0,0,1) \implies \overline{\overline{A} \cdot \overline{B} \cdot C}$$
$$(0,1,0) \implies \overline{\overline{A} \cdot B \cdot \overline{C}}$$
$$(1,0,0) \implies \overline{A \cdot \overline{B} \cdot \overline{C}}$$

Simplify each term by applying De Morgan's Theorem, then combine them with AND, we have:

$$\boxed{P = \left(A + B + \overline{C}\right) \cdot \left(A + \overline{B} + C\right) \cdot \left(\overline{A} + B + C\right)}$$

2. Consider a 16-bit 2's complement representation.
   (a) What is the largest (most positive) number and the smallest (most negative) value in this representation scheme?

   Solution: Recall that in 2's complement representation, the most significant bit (MSB) is the sign bit. The largest number is when the MSB is 0 and all other bits are 1's, i.e., $0111\,1111\,1111\,1111_2 = \boxed{32767_{10}}$.

   Similarly, the smallest number is when the MSB is 1 and all other bits are 0's, i.e., $1000\,0000\,0000\,0000_2 = \boxed{-32768_{10}}$.

   (b) Write down the bit pattern representing 18, -18, 25, and -25, respectively.

   Solution: The bit patterns are:

Decimal	Bit Pattern
18	$0000\,0000\,0001\,0010_2$
-18	$1111\,1111\,1110\,1110_2$
25	$0000\,0000\,0001\,1001_2$
-25	$1111\,1111\,1110\,0111_2$

   (c) What are the values of the above bit patterns if they are treated as unsigned integers?

   Solution: When treated as unsigned integers, the values are:

Bit Pattern	`uint` Value
$0000\,0000\,0001\,0010_2$	18
$1111\,1111\,1110\,1110_2$	65 518
$0000\,0000\,0001\,1001_2$	25
$1111\,1111\,1110\,0111_2$	65 511

(d) Add the bit patterns together for the following:

(1) $18+25$ (2) $18+(-25)$ (3) $(-18)+25$ (4) $(-18)+(-25)$

Solution: The bit patterns required are:

(1) $18+25=43$

$$\begin{array}{rl} 0000\,0000\,0001\,0010 & (+18) \\ +\,0000\,0000\,0001\,1001 & (+25) \\ \hline 0000\,0000\,0010\,1011 & (+43) \end{array}$$

(3) $(-18)+25=7$

$$\begin{array}{rl} 1111\,1111\,1110\,1110 & (-18) \\ +\,0000\,0000\,0001\,1001 & (+25) \\ \hline \not{1}\,0000\,0000\,0000\,0111 & (+7) \end{array}$$

(2) $18+(-25)=-7$

$$\begin{array}{rl} 0000\,0000\,0001\,0010 & (+18) \\ +\,1111\,1111\,1110\,0111 & (-25) \\ \hline 1111\,1111\,1111\,1001 & (-7) \end{array}$$

(4) $(-18)+(-25)=-43$

$$\begin{array}{rl} 1111\,1111\,1110\,1110 & (-18) \\ +\,1111\,1111\,1110\,0111 & (-25) \\ \hline \not{1}\,1111\,1111\,1101\,0101 & (-43) \end{array}$$

3. Prove that the multiplication of an n-bit binary number A and an m-bit binary number B gives a product $A \times B$ of no more than $(n+m)$ bits.

Solution: Note that $A \in [0, 2^n-1]$ and $B \in [0, 2^m-1]$, $m, n \geq 1$, and $m+n \geq 2$.

The maximum value of $A \times B$ is given by:

$$\begin{aligned} \max(A \times B) &= \max(A) \times \max(B) \\ &= (2^n-1) \times (2^m-1) \\ &= 2^n \times 2^m - 2^n - 2^m + 1 \\ &= 2^{n+m} - 2^n - 2^m + 1 \end{aligned}$$

Note that to represent the term 2^{n+m}, we need at least $(n+m+1)$ bits, and the maximum value representable by $(n+m)$ bits is $2^{n+m}-1$. Therefore, we need to show that:

$$\forall m, n \in [1, +\infty) \cap \mathbb{Z}, \quad 2^{n+m} - 2^n - 2^m + 1 < 2^{n+m} \tag{1}$$

Assume that m is constant, when $n = 1$, we have:

$$\begin{aligned} \text{L.H.S.} &= 2^{1+m} - 2^1 - 2^m + 1 \\ &= 2^{m+1} - 2 - 2^m + 1 \\ &= 2^{m+1} - 2^m - 1 \\ &= 2^m(2-1) - 1 \\ &= 2^m - 1 \end{aligned}$$

and R.H.S. $= 2^{m+1}$, so L.H.S. $<$ R.H.S. clearly holds. Therefore, the proposition holds for $n = 1$.

Now, assume that the proposition holds for $n = k$ where $k \in [1, +\infty) \cap \mathbb{Z}$, i.e.,

$$\forall k \in [1, +\infty) \cap \mathbb{Z}, \quad 2^{k+m} - 2^k - 2^m + 1 < 2^{k+m} \tag{2}$$

We need to show that the proposition holds for $n = k+1$, i.e.,

$$\begin{aligned} \text{L.H.S.} &= 2^{k+1+m} - 2^{k+1} - 2^m + 1 \\ &= 2^{k+m} \cdot 2 - 2^k \cdot 2 - 2^m + 1 \\ &= 2^{k+m} - 2^m - 2^k + 1 + 2^{k+m} - 2^k \end{aligned}$$

and

$$\begin{aligned} \text{R.H.S.} &= 2^{k+1+m} \\ &= 2^{k+m} \cdot 2 \\ &= 2^{k+m} + 2^{k+m} \end{aligned}$$

Subtract 2^{k+m} from both sides, we have:

$$\text{L.H.S.} = \underbrace{2^{k+m} - 2^m - 2^k + 1}_{\text{Induction Hypothesis (Eq. 2)}} - 2^k \qquad \text{and} \qquad \text{R.H.S.} = 2^{k+m}$$

Observe that by the induction hypothesis, we have $2^{k+m} - 2^m - 2^k + 1 < 2^{k+m}$, and since $2^k > 0$, we have:

$$\text{L.H.S.} < 2^{k+m} - 2^k < 2^{k+m}$$

Therefore, the proposition holds for $n = k+1$.

By the principle of mathematical induction, the proposition holds for all $n \in [1, +\infty) \cap \mathbb{Z}$.

Now, proposition 2 is partially proven for all n and constant m. Observe that the proposition is symmetric in m and n, i.e., m and n are dummy variables. Therefore, it is trivial to show that the proposition holds for all $m \in [1, +\infty) \cap \mathbb{Z}$ and constant n. Hence, we have shown that the proposition holds for all $m, n \in [1, +\infty) \cap \mathbb{Z}$.

Now, we can conclude that the $A \times B = 2^{n+m} - 2^n - 2^m + 1 < 2^{n+m}$, which means that the product $A \times B$ can be represented by no more than $(n+m)$ bits.

Q.E.D.

4. Verify the validity of the multiplication of integers (2's complement) procedure in the lecture notes. (Give the prove)

Solution: It may be helpful to recall that for any n-bit 2's complement number $(a_{n-1}a_{n-2}\ldots a_2a_1a_0)_2$, where $a \in [0,1]$, the value of the number is given by

$$A = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

Also, recall that the multiplication procedure in the lecture notes involves sign-extension and negation of the multiplicand and multiplier, it would be helpful to first prove that these two operations are valid.

Proof of sign-extension:

Suppose A is sign-extended to A' of m bits, where $m > n$. Then, A' can be expressed as:

$$A' = \underbrace{(a_{n-1}a_{n-1}\ldots a_{n-1}a_{n-2}\ldots a_2a_1a_0)_2}_{m \text{ bits}}$$

and its value is given by:

$$A' = -2^{m-1}a_{n-1} + \sum_{i=0}^{m-2} 2^i a_j, \text{ where } j = \begin{cases} i & \text{if } i \in [0, n-2] \\ n-1 & \text{if } i \geq n-1 \end{cases}$$

By splitting the summation, we have:

$$\begin{aligned}
A' &= -2^{m-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i + \sum_{i=n-1}^{m-2} 2^i a_{n-1} \\
&= -2^{m-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i + a_{n-1} \cdot \sum_{i=n-1}^{m-2} 2^i \\
&= -2^{m-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i + a_{n-1} \cdot \left(\frac{2^{n-1}\left(2^{m-2-(n-1)+1} - 1\right)}{2-1} \right) \\
&= -2^{m-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i + a_{n-1} \cdot \left(2^{m-1} - 2^{n-1}\right) \\
&= \cancel{-2^{m-1}a_{n-1}} + \sum_{i=0}^{n-2} 2^i a_i + \cancel{2^{m-1}a_{n-1}} - 2^{n-1}a_{n-1} \\
&= -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i \\
&= A
\end{aligned}$$

We can conclude that the value of A' is the same as the value of A, therefore, sign-extension is valid.

Q.E.D.

Proof of negation: Recall that the negation of a number A is given by taking its 2's complement, i.e. apply bitwise NOT to the number, then add 1 to the result.

Consider a number B of n bits, which is given by $(\overline{a_{n-1}}\,\overline{a_{n-2}}\ldots\overline{a_2}\,\overline{a_1}\,\overline{a_0})_2+1$. Then, the value of B is given by:

$$B=-2^{n-1}\overline{a_{n-1}}+\sum_{i=0}^{n-2}2^i\overline{a_i}+1$$

Also recall that for any bit a_i, we have $\overline{a_i}=1-a_i$.

Now, consider $A+B$, we have:

$$\begin{aligned}
A+B &= -2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^ia_i-2^{n-1}\overline{a_{n-1}}+\sum_{i=0}^{n-2}2^i\overline{a_i}+1\\
&= -2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^ia_i-2^{n-1}(1-a_{n-1})+\sum_{i=0}^{n-2}2^i(1-a_i)+1\\
&= -2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^ia_i-2^{n-1}+2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^i-\sum_{i=0}^{n-2}2^ia_i+1\\
&= \cancel{-2^{n-1}a_{n-1}}+\sum_{i=0}^{n-2}\cancel{2^ia_i}-2^{n-1}+\cancel{2^{n-1}a_{n-1}}+\sum_{i=0}^{n-2}2^i-\sum_{i=0}^{n-2}\cancel{2^ia_i}+1\\
&= -2^{n-1}+\frac{2^0\cdot\left(2^{n-2-0+1}-1\right)}{2-1}+1\\
&= -2^{n-1}+2^{n-1}-1+1\\
A+B &= 0\\
\text{Therefore, } B &= -A
\end{aligned}$$

We can conclude that the negation of a number A is indeed valid, since it gives the negative of the number, i.e., $-A$.

Q.E.D.

Now, we can verify the multiplication procedure in the lecture notes.

Proof of multiplication procedure:

First, sign-extension is performed on the multiplicand and multiplier to ensure that they are both of the same bit length, which is proven to be valid above.

Now consider the multiplication of two n-bit 2's complement numbers A and B, where A is the multiplicand and B is the multiplier. We have:

$$A\times B=\left(-2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^ia_i\right)\times\left(-2^{n-1}b_{n-1}+\sum_{j=0}^{n-2}2^jb_j\right)$$

Case 1: $A,B\geq 0$ (i.e., $a_{n-1}=b_{n-1}=0$), our desired expression of the product is:

$$A\times B=\left(\sum_{i=0}^{n-2}2^ia_i\right)\times\left(\sum_{j=0}^{n-2}2^jb_j\right) \tag{3}$$

Consider the procedure given by the lecture notes, $\forall b_j\in B=1$, A is shifted to the left by j bits, which is equivalent to $A\times 2^j$, and then added to the product. Therefore,

$$\begin{aligned}
A\times B &= \left(\sum_{i=0}^{n-2}2^ia_i\right)\times 2^0\times b_0+\left(\sum_{i=0}^{n-2}2^ia_i\right)\times 2^1\times b_1+\cdots+\left(\sum_{i=0}^{n-2}2^ia_i\right)\times 2^{n-1}\times b_{n-1}\\
&= \sum_{i=0}^{n-2}2^ia_i\cdot\left(2^0b_0+2^1b_1+\cdots+2^{n-1}b_{n-1}\right)\\
&= \sum_{i=0}^{n-2}2^ia_i\cdot\left(\sum_{j=0}^{n-2}2^jb_j+2^{n-1}b_{n-1}\right)\\
&= A\times B \qquad (\text{since } b_{n-1}=0)
\end{aligned}$$

Therefore, the multiplication procedure holds for two non-negative numbers.

Case 2: $A \geq 0, B < 0$ (i.e., $a_{n-1} = 0, b_{n-1} = 1$), our desired expression of the product is:

$$A \times B = \left(\sum_{i=0}^{n-2} 2^i a_i \right) \times \left(-2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \right) \tag{4}$$

The procedure states:

1. For each $j \in [0, n-2]$ where $b_j = 1$, shift A to the left by j bits, which is equivalent to $A \times 2^j$.
2. Sum all terms given by step 1, which gives $\sum_{j=0}^{n-2} A \times 2^j \times b_j$.
3. For the MSB of B, if $b_{n-1} = 1$, then negate A and shift it to the left by $(n-1)$ bits, which is equivalent to $-A \times 2^{n-1} \times b_{n-1}$.
4. Add the results of step 2 and step 3 together, which gives:

$$\begin{aligned} A \times B &= \sum_{j=0}^{n-2} A \times 2^j \times b_j - A \times 2^{n-1} \times b_{n-1} \\ &= A \times \left(-2^{n-1} b_{n-1} + \sum_{j=0}^{n-2} 2_j b_j \right) \\ &= \left(\sum_{i=0}^{n-2} 2^i a_i \right) \times \left(-2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \right) \end{aligned}$$

This matches our desired expression of the product, therefore, the multiplication procedure holds for $A \geq 0, B < 0$.

Case 3: $A < 0, B \geq 0$ (i.e., $a_{n-1} = 1, b_{n-1} = 0$). By considering the commutativity of multiplication, and the fact that A and B are dummy variables, it is trivial to show that the multiplication procedure holds for this case as well.

Case 4: $A < 0, B < 0$ (i.e., $a_{n-1} = 1, b_{n-1} = 1$), our desired expression of the product is:

$$A \times B = \left(-2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i \right) \times \left(-2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \right) \tag{5}$$

The multiplication procedure for this case is identical to that of Case 2. Notice how A was not expressed in its full form (i.e. $-2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$) in the proof of Case 2. This implies that the calculation procedure is independent of the sign of A, and therefore, the multiplication procedure holds for this case as well.

By now, we have shown that the multiplication procedure in the lecture notes holds for all cases of A and B, therefore, the multiplication procedure is valid.

Q.E.D.

5. Any floating-point representation used in a computer can represent only certain real numbers exactly; all others must be approximated. If A' is the stored value approximating the real value A, then the relative error, r, is expressed as

$$r = \frac{A - A'}{A}$$

Represent the decimal quantity $+0.4$ in the following floating-point format: base $= 2$; exponent: biased, 4 bits; significand, 7 bits. What is the relative error?

Solution: First, convert 0.4 to binary and normalise it:

$$\begin{aligned} 0.4_{10} &= 0.0110011001100\ldots_2 \\ &= 1.\underbrace{1001100}_{\text{significand}}1100\ldots_2 \times 2^{-2} \end{aligned}$$

Then, find the biased exponent:

$$\text{Biased exponent} = -2 + (2^3 - 1) = 5_10 = 0101_2$$

Therefore, the floating-point representation of 0.4 is:

$$A' = \underbrace{0}_{\text{sign}}\ \underbrace{0101}_{\text{exponent}}\ \underbrace{1001100}_{\text{significand}}$$

Now, we find the stored value of A':

$$\begin{aligned} A' &= (1+2^{-1}+2^{-4}+2^{-5})\times 2^{-2} \\ &= 2^{-2}+2^{-3}+2^{-6}+2^{-7} \\ &= 0.3984375_{10} \end{aligned}$$

Now, we can calculate the relative error:

$$\begin{aligned} r &= \frac{A-A'}{A} \\ &= \frac{0.4-0.3984375}{0.4} \\ &= \frac{0.0015625}{0.4} \\ &= \boxed{0.00391} \text{(corr. to 3 sig. figs.)} \end{aligned}$$

6. Consider a 40-bit floating point representation with a sign bit S, an exponent E (biased, 11 bits), and a significand f (28 bits). The value is

$$V = (-1)^S \cdot 1.f \cdot 2^{E-1023}$$

Here, $E = 11\ldots111_2$ and $f = 00\ldots000_2$ do not have special meanings.

(a) Write down the largest positive number that can be represented.

Solution:

$$\begin{aligned} \text{Largest value} &= \left(1+\sum_{i=-28}^{-1} 2^i\right)\cdot 2^{2^{11}-1-1023} \\ &= \left(1+2^{-28}\cdot(2^{28}-1)\right)\cdot 2^{1024} \\ &= (1+1-2^{-28})\cdot 2^{1024} \\ &= \boxed{2^{1025}-2^{996}} \end{aligned}$$

(b) Write down the smallest positive number that can be represented.

Solution:

$$\begin{aligned} \text{Smallest value} &= 1\cdot 2^{-1023} \\ &= \boxed{2^{-1023}} \end{aligned}$$

(c) Write down the bit pattern representing the value 15.3125.

Solution: First, convert 15.3125 to binary:

$$\begin{aligned} 15.3125_{10} &= 1111.0101_2 \\ &= 1.1110101_2 \times 2^3 \end{aligned}$$

Now, find the biased exponent:

$$\begin{aligned} \text{Biased exponent} &= 3+(2^{11-1}-1) \\ &= 3+1023 = 1026_{10} = 10000000010_2 \end{aligned}$$

Combining the sign bit, exponent, and significand, we have:

$$\text{Bit pattern} = \underbrace{0}_{\text{sign}}\ \underbrace{10000000}_{\text{exponent}}\underbrace{10111010000000000000000}_{\text{significand}} = \boxed{\texttt{0x402EA00000}}$$

(d) Write down the value represented by the bit pattern `0xC06F800000`.

Solution: Convert the hexadecimal to binary:

$$\begin{aligned}\texttt{0xC06F800000} &= \texttt{1100 0000 0110 1111 1000 0000 0000 0000 0000 0000}\\ &= \underbrace{1}_{\text{sign}}\ \underbrace{10000001}_{\text{exponent}}\underbrace{10111100000000000000000}_{\text{significand}}\end{aligned}$$

Now, calculate the value:

$$\begin{aligned}V &= (-1)^1 \cdot 1.1111000_2 \cdot 2^{1030-1023}\\ &= -1.11111000_2 \cdot 2^7\\ &= -11111100_2\\ &= \boxed{-252_{10}}\end{aligned}$$

(e) If we assign 16 bits and 23 bits for exponent E and significand f , respectively. What is the largest positive number that can be represented ? Discuss what is the relation between range and precision in floating point number representation?

Solution: *Omitted.*