



**Date handed out: 5 March, Monday, 2018**

**Date submission due: 18 March, Sunday, 2018**

### **Programming Assignment 1: Full Adder**

#### **Purpose:**

The main purpose of this programming assignment is to revise the topics covered in the first three weeks including fundamentals of C programming, conditional statements and repetitive statements. In this assignment, you will also practice using character datatype.

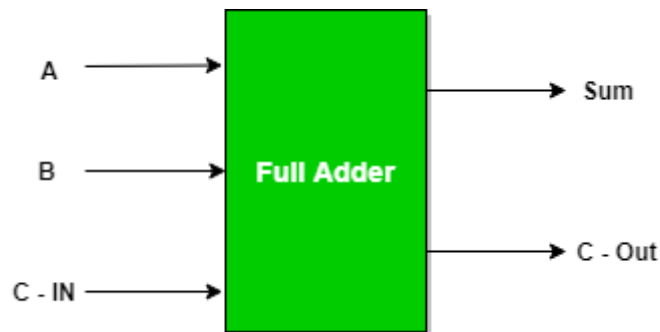
#### **Description:**

An adder is a digital circuit that performs addition of numbers. Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

#### **Programming Requirements:**

A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.

The figure below shows the block diagram of a full adder:



The truth table for the full adder is:

Inputs			Outputs	
A	B	C – IN	Sum	C – Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

When the user first runs the program, you will display the following menu:

Welcome to Full-Adder!

(1) Compute and display the outputs

(2) Quit

Depending on the option given by the user, your program will work as follows:

**Option 1:** When the user chooses this option, your program will first ask you to enter 3-digit binary inputs. If more than 3 bits is entered, then your program should print an error message and ask the user to reenter the input. Three digits will be used as three inputs required by the full adder. The first digit is A, the second digit is B and the third digit is for C\_IN.

The data lines can be entered in different bases, in base 2, base 8 or base 16. If the user is entering the text in base 2, then you will not need to make base conversions. However, if the user is entering in base 16 or base 8, then you will need to convert it to binary. Assuming that the user is entering more than 3 bits for inputs, then your program should print an error message and ask the user to enter the values again. If the data lines are entered in base 8 or 16, your program will convert it to 3- digit binary number. If a value > 7 is entered, then your program will print a message "not possible to convert it to 3-digit binary number" and will ask the user to enter the value again.

Which base will you use to enter input (base 16/8/2)? 2

Please enter input: 011

In order to compute the values of Sum and C\_OUT, you need to use the following logical expressions. For calculating Sum, you will need to mainly use XOR operations and for calculating the value of C\_OUT, you will need to use AND, OR and XOR operations.

**SUM = C-IN XOR (A XOR B)**

**C\_OUT = (A AND B) OR (C\_IN AND (A XOR B))**

For your reference, the truth table for bitwise AND is:

X (where X is a single bit)	Y (where Y is a single bit)	X AND Y
1	1	1
1	0	0
0	1	0
0	0	0

Similarly, bitwise OR and XOR (exclusive or) are given by the table below:

X (where X is a single bit)	Y (where Y is a single bit)	X OR Y	X XOR Y
1	1	1	0
1	0	1	1
0	1	1	1
0	0	0	0

If the user select **option 2** (quit), then your program will stop.

### **Full Sample Run:**

```
Welcome to Full Adder!
(1) Compute and Display the Outputs
(2) Quit
You choose: 1
You have chosen option 1
Which base will you use to enter data lines (base 16/8/2)? 2
Please enter input: 101
Sum is 0 C_out is 1

Welcome to Full Adder!
(1) Compute and Display the Outputs
(2) Quit
You choose: 1
You have chosen option 1
Which base will you use to enter data lines (base 16/8/2)? 8
Please enter input: 10
Octal 10 cannot be represented with 3 bits! Please try again!
Please enter input: 1
Sum is 1 C_out is 0
```

```

Welcome to Full Adder!
(1) Compute and Display the Outputs
(2) Quit
You choose: 1
You have chosen option 1
Which base will you use to enter data lines (base 16/8/2)? 16
Please enter input: A
Hexadecimal A can not be represented with 3 bits!Please try again!
Please enter input: 5
Sum is 0 C_out is 1

```

```

Welcome to 4 to 1 multiplexer!
(1) Compute and Display the Output
(2) Quit
You choose: 2
You have chosen option 2
Byee!!

```

### Grading Schema:

Your program will be graded as follows:

Grading Point	Mark (100)
The menu (keeping the users in a loop until exit is chosen)	10
Reading text in binary	10
Converting to correct base	25
Displaying error messages	10
Finding the Output	25
Code quality (e.g., variable names, formulation of selection statements and loops, etc)	20

### Rules:

Please make sure that you follow the restrictions for the assignment as follows.

- **Strictly obey the input output format. Do not print extra things.**
- **You are not allowed to use data structures such as arrays to store values for the conversion operation.**
- **You are not allowed to use bitwise operators.**
- **You are not allowed to define your own functions.**
- **Add your name/surname and ID at the top of your code as comments and name your source file "CNG140-P1.c"**
- **Upload only source file. Do not compress it (zip, rar, ...)**