



**School of Computer Science
Faculty of Science**

**COMP-2650: Computer Architecture I: Digital Design
Winter 2021**

Lab#	Date	Title	Due Date	Grade Release Date
Lab 10	Week 10	Canonical Sum of Products	March 16, 2021 Tuesday Midnight AoE Wednesday 7 AM EDT	March 22, 2021

This lab's objectives will be to master the topics in logic circuit design by implementing the algorithms with a programming language, herein, C/C++.

Step 1. Environment Setup

Our programming environment is the same as the first lab (Lab 01). In this lab, we want to continue the new series of labs about designing a logic circuit. Particularly, in this lab, we want to write the boolean function (expression) for the output binary variables based on the standard form of the product of MAXTERMs. Product of MAXTERMs is also called Canonical Products of Sums (PoS) since each MAXTERM is an OR between the input binary variables (either in normal form X or in complement form X'), e.g., $Z+Y+X'$, followed by an AND on the MAXTERMs, e.g., $F(Z,Y,X) = M_0+M_2+M_3 = (Z+Y+X)(Z+Y'+X)(Z+Y'+X')$.

In the previous Lab 09, we wrote a program that printed out the Boolean function in the form of a sum of minterms (Canonical Sum of Products):

```

01 #include <stdio.h>
02 #include <math.h>
03
04 #define INPUT_VARIABLE_COUNT 3
05 #define OUTPUT_VARIABLE_COUNT 1
06
07 void build_right_side(int truth_table[][INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT]){...}
08 void build_left_side(int truth_table[][INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT]){...}
09 void to_minterm(int truth_table[][INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT]){...}
10 int main(void) {
11     setbuf(stdout, NULL);
12
13     int TRUTH_TABLE_ROW_COUNT = (int)pow(2, INPUT_VARIABLE_COUNT);
14     int truth_table[TRUTH_TABLE_ROW_COUNT][INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT] = {0};
15     const char variables[INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT] = {'Z', 'Y', 'X', 'F'};
16
17     build_left_side(truth_table);
18     build_right_side(truth_table);
19
20     //printing the header for input variables
21     for(int i = 0; i < INPUT_VARIABLE_COUNT; i = i + 1){
22         printf("%c", variables[i]);
23     }
24     printf(" : ");
25
26     //printing the header for output variables
27     for(int i = INPUT_VARIABLE_COUNT; i < INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT; i = i + 1){
28         printf("%c", variables[i]);
29     }
30     printf("\n");
31
32     //printing the content of each row
33     for(int i = 0; i < TRUTH_TABLE_ROW_COUNT; i = i + 1){
34
35         //printing the content of each row regarding the input variables

```

Commented [HF1]: Depending on the C/C++ compiler, some allows us to initialize all the elements of the matrix to 0 by this. However, some does not allow this. So, you have to write for loops to initialize the elements. Please look at the discussion board for a possible solution, proposed by students.

```

36         for(int j = 0; j < INPUT_VARIABLE_COUNT; j = j + 1){
37             printf("%d, ", truth_table[i][j]);
38         }
39         printf(" : ");
40
41         //printing the content of each row regarding the output variables
42         for(int j = INPUT_VARIABLE_COUNT; j < INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT; j = j + 1){
43             printf("%d", truth_table[i][j]);
44         }
45         printf("\n");
46     }
47     to_minterm(truth_table);
48     return 0;

```

```

output value for row# 0 of F1 output variable:1
output value for row# 1 of F1 output variable:0
output value for row# 2 of F1 output variable:0
output value for row# 3 of F1 output variable:0
output value for row# 4 of F1 output variable:1
output value for row# 5 of F1 output variable:1
output value for row# 6 of F1 output variable:0
output value for row# 7 of F1 output variable:0
Z, Y, X,  : F
0, 0, 0,  : 1
0, 0, 1,  : 0
0, 1, 0,  : 0
0, 1, 1,  : 0
1, 0, 0,  : 1
1, 0, 1,  : 1
1, 1, 0,  : 0
1, 1, 1,  : 0
output variable F1 = Z'Y'X'+ZY'X'+ZY'X+

```

Given Lab 09, it's very easy to extend it to MAXTERMs by writing a new function called `to_MAXTERM()` and replace it with `to_minterm()` at line#47 above. Otherwise, the only essential parts are 1) asking the user about the values of output binary functions ('F'), and 2) print out the AND of MAXTERMs whenever a 0 is received from the user.

Given the truth table we built in Lab 08, in a loop on rows, wherever we see 0 in the last column of the truth table, we print out the OR of the input variables based on whether they are 0 or 1:

```

void to_MAXTERM(int truth_table[][INPUT_VARIABLE_COUNT + OUTPUT_VARIABLE_COUNT]){
    for(int j = 0; j < OUTPUT_VARIABLE_COUNT; j = j + 1){
        printf("output variable F%d = ", j+1);
        for(int i = 0; i < TRUTH_TABLE_ROW_COUNT; i = i + 1){
            //to be completed!
        }
        printf("\n");
    }
}

```

A sample run would be:

```

output value for row# 0 of F1 output variable:1
output value for row# 1 of F1 output variable:0
output value for row# 2 of F1 output variable:0
output value for row# 3 of F1 output variable:0
output value for row# 4 of F1 output variable:1
output value for row# 5 of F1 output variable:1
output value for row# 6 of F1 output variable:0

```



output value for row# 7 of F1 output variable:0

```
Z, Y, X, : F
0, 0, 0, : 1
0, 0, 1, : 0
0, 1, 0, : 0
0, 1, 1, : 0
1, 0, 0, : 1
1, 0, 1, : 1
1, 1, 0, : 0
1, 1, 1, : 0
```

output variable F1 = $(Z+Y+X')(Z+Y'+X)(Z+Y'+X')(Z'+Y'+X)(Z'+Y'+X')$

As seen, the Boolean function for the only output variable F1 is printed out in the form of the Canonical Product of Sums (Product of MAXTERMs). We can *optionally* print out the MAXTERM numbers, e.g., we could print out:

output variable F1 = $\Pi(1,2,3,6,7) = (Z+Y+X')(Z+Y'+X)(Z+Y'+X')(Z'+Y'+X)(Z'+Y'+X')$

Lab Assignment

You should complete the above program that asks for the value of output variable F1 as follows:

```
output value for row# 0 of F1 output variable:1
output value for row# 1 of F1 output variable:0
output value for row# 2 of F1 output variable:0
output value for row# 3 of F1 output variable:0
output value for row# 4 of F1 output variable:1
output value for row# 5 of F1 output variable:1
output value for row# 6 of F1 output variable:0
output value for row# 7 of F1 output variable:0
```

When the user enters the values, the program can *optionally* print out the truth as shown below:

```
Z, Y, X, : F
0, 0, 0, : 1
0, 0, 1, : 0
0, 1, 0, : 0
0, 1, 1, : 0
1, 0, 0, : 1
1, 0, 1, : 1
1, 1, 0, : 0
1, 1, 1, : 0
```

Then it should output a menu of commands as follows:

Enter the command number:

0) Exit

1) Canonical SoP => Optional! From Lab 09.

2) Canonical PoS

If a user selects (2), the program should print out the Boolean function for F1 in the form of a Product of MAXTERMs (Canonical PoS) as shown below:

output variable F1 = $(Z+Y+X')(Z+Y'+X)(Z+Y'+X')(Z'+Y'+X)(Z'+Y'+X')$

If the user selects (1), the program outputs the Canonical Sum of Products as we did in Lab 09. *This is optional.* If the user selects (0), the program ends. Please restrict the user to enter inputs within the range



{0,1} for the value of the output variable. For instance, if the user enters 2, -1, ..., print out an error message and come back to ask for correct inputs.

It is required to write a *modular* program. Please put the part of the code that outputs a MAXTERM based on the value of input variables in a new function called `to_MAXTERM()` inside the `main.c` file.

Deliverables

You will prepare and submit the program in one single zip file `Lab10_UWinID.zip` containing the following two items:

1. The entire project folder `Project` including the code file (`main.c` or `main.cpp`) and executable file (`main.exe` in windows or `main` in mac/linux)
2. The result of the commands in the file `Results.pdf`. Simply take screenshots of the results and save (print) them **into a single pdf**.
2. [Optional and if necessary] A readme document in a txt file `ReadMe.txt`. It explains how to build and run the program as well as any prerequisites that are needed. ***Please note that if your program cannot be built and run on our computer systems, you will lose marks.***

In sum, your final `Lab10_UWinID.zip` file for the submission includes 1 folder (entire project folder), 1 image (results snapshot) and 1 txt (report). ***Please follow the naming convention as you lose marks otherwise.*** Instead of `UWinID`, use your own UWindsor account name, e.g., mine is `hfani@uwindsor.ca`, so,

`Lab10_hfani.zip`

- (75%) `Project` => Printing the Sum of Products
 - o [any required library, header or source files]
 - o `main.c` or `main.cpp` => Must be compiled and built with no error!
 - o `main.exe` or `main`
- (10%) `Results.pdf`
- (Optional) `ReadMe.txt`

(10%) Modular Programming (using separate functions)

(5%) Files Naming and Formats, and Folder Structure