



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

Mini Project (Group)

SECR1013 DIGITAL LOGIC

SEMESTER I, SESSION 2019/2020

Lab 4: Part III & IV - The Practical Work

Lecturer: Ms. Rashidah binti Kadir

Name	Student ID
SEE WEN XIANG	A19EC0206
BRYNARD BRYAN EDMUND	A19EC0029

Section: 05

PROGRAMME: BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING)

DEDICATION AND ACKNOWLEDGEMENT

This final project has been completed successfully with the dedication to our Digital Logic Lecturer, Ms Rashidah. I appreciate that she has been teaching us throughout the semester with passion and kindness. She will answer our doubt without hesitation although she was busy with something else. Without her, we would not have much knowledge of this subject and might not be able to complete this project. Furthermore, Ms. Rashidah helped us a lot when conducting the project. As we are fairly new to the components and steps to conduct the project, she was willing to assist us group by group without any hesitation.

Next, we would like to thank our friends and course mate who supports us when conducting this project. We helped one another to make sure everyone was not left behind.

Finally, we would like to thank our parents for giving moral support and financial support for us to continue our study in Universiti Teknologi Malaysia.

Table of Contents

1.0	BACKGROUND	1
2.0	THE PROBLEM	1
3.0	OBJECTIVES	1
4.0	THE FLOWCHART	1
5.0	COMPONENT	2
6.0	MATERIALS AND SOFTWARE USED	2
7.0	CIRCUIT IMPLEMENTATION	3
8.0	PHYSICAL SYSTEM IMPLEMENTATION	5
9.0	DISCUSSION	6
10.0	CONCLUSION	6
11.0	REFERENCES	7
12.0	APPENDIX	7

1.0 BACKGROUND

In this project, we will implement 3 different components on a single GAL device. These components including:

1. 3-bit Count up Counter
2. 3-bit Comparator
3. Clock disabler

2.0 THE PROBLEM

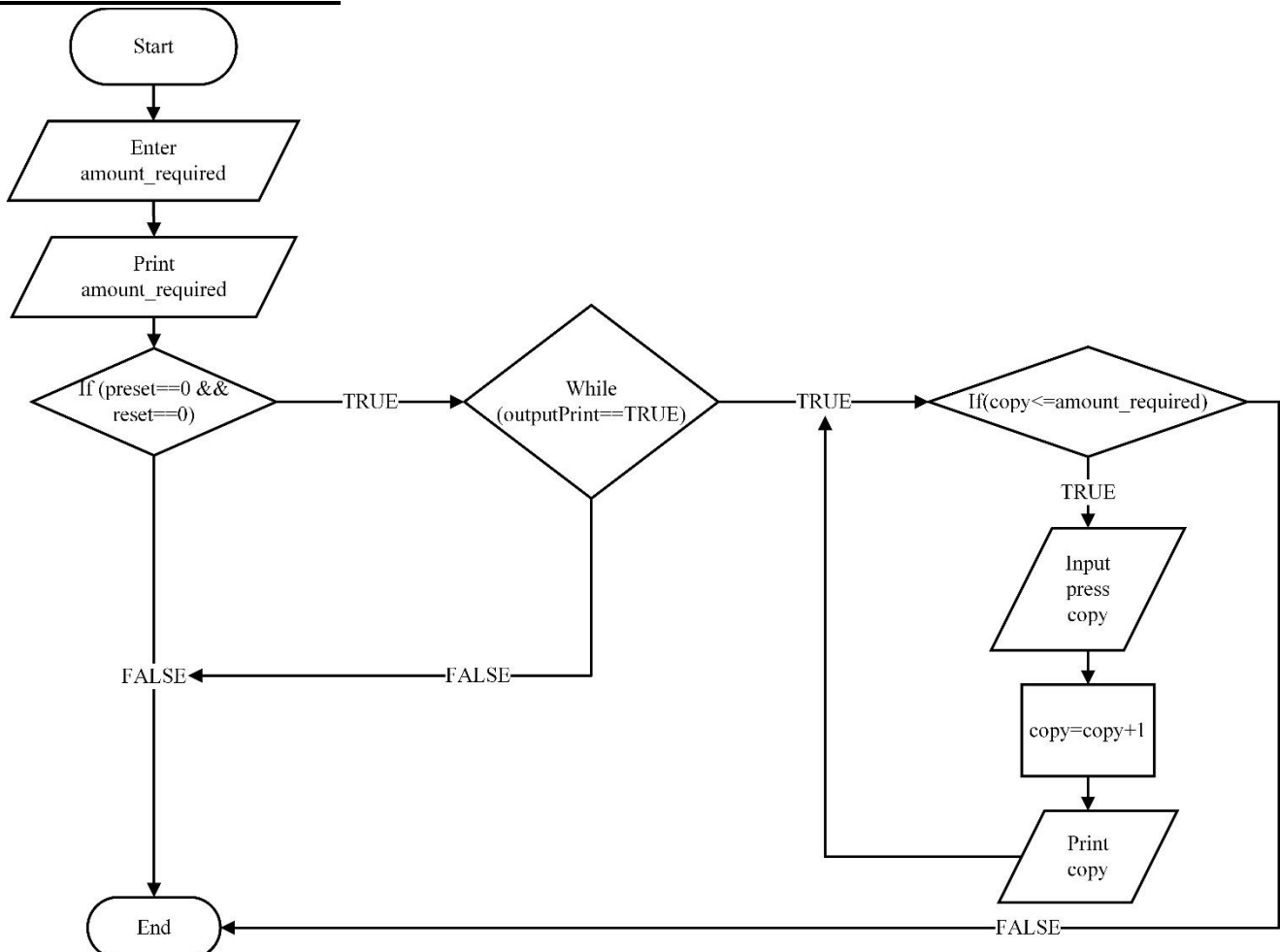
The user can enter the number of copies required. There is two led display, one is for the amount of copies required, and another one is for the amount of printed copies. The counter will count when user press a button. The comparator will switch between high and low according to the required copies and amount of printed copies. After the amount of copies reached the required amount, a clock disabler will stop the counter.

3.0 OBJECTIVES

The objectives of this laboratory are to introduce the students to:

1. Improve the development of a PLD device
2. Know how to write a simple Hardware Description Language

4.0 THE FLOWCHART



5.0 COMPONENT

1. Input Switches: to set the required no of copies
2. 3-bit up Counter: to count the amount of copies that has been printed
3. 3-bit Comparator: to compare the amount of copies required and the amount of copies printed
4. Clock Disabler: to stop the counter
5. 7-Segment Display: To display the number of copies required and printed

6.0 MATERIALS AND SOFTWARE USED

1. Breadboard
2. ATMEL 22V10 - 1 unit
3. ETS-5000 Digital Training kit
4. Wellon or Hi-Lo ALL-11 Universal Programmer & Tester
5. WinCUPL 5.0 Software
6. Handouts:
 - a) "WinCUPL user manual"
 - b) "ATMEL22V10 Data Sheet"
 - c) "How to use Wellon Programmer"
 - d) "How to use Win CUPL 5"
6. Input Switches: to set the required no of copies
7. 3-bit up Counter: to count the amount of copies that has been printed
8. 3-bit Comparator: to compare the amount of copies required and the amount of copies printed
9. Clock Disabler: to stop the counter
10. 7-Segment Display: To display the number of copies required and printed

7.0 CIRCUIT IMPLEMENTATION

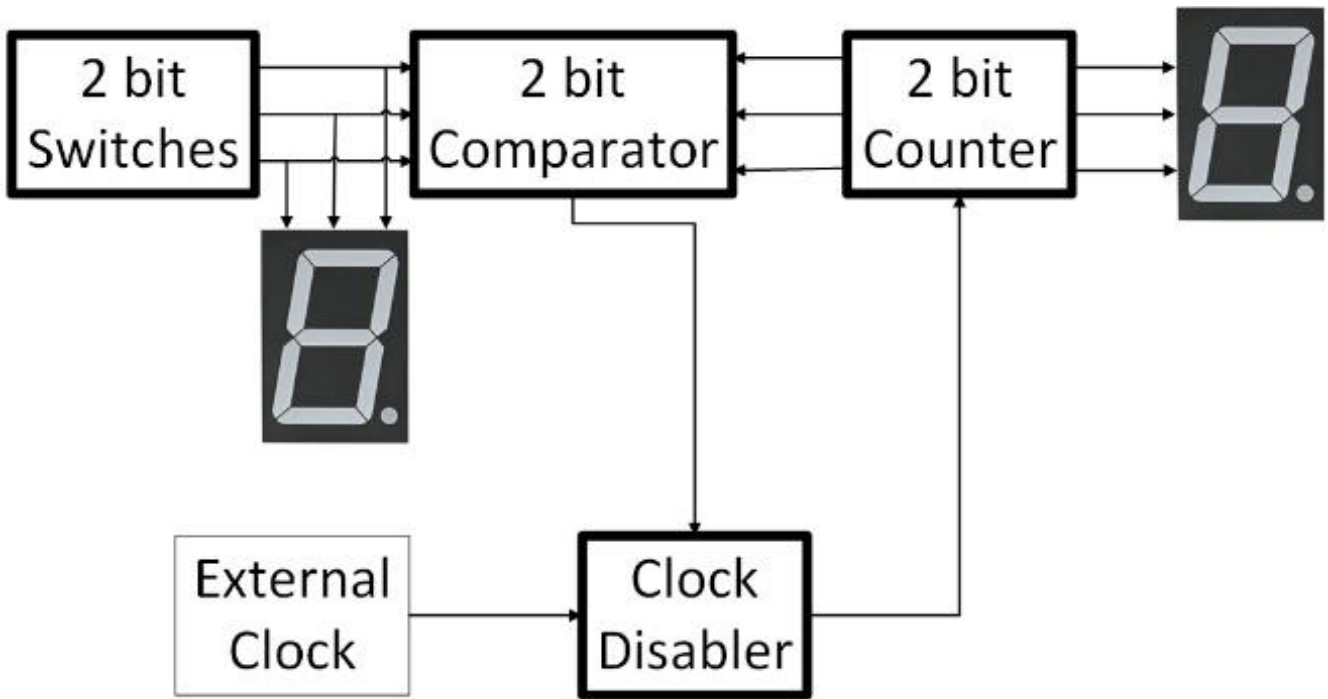


Figure 7.1 A block diagram of a complete 2-bit system

The pattern of our design is as shown in Figure 7.1 above. The main components are the 2-bit counter, 2-bit comparator, and clock disabler while the sub-component is some basic gates. The 2-bit counter is to count the amount of copies printed by the machine. The 2-bit comparator will compare the number entered by the user to the number count by the counter. If the number became the same, the clock will be disabled, and the counter will stop counting. The required amount of copies and the printed amount of copies will be displayed on two 7-Segment LED display.

To implement this, the user has to input the number of required copies to be printed by using two input switches. The number can be entered are 0 to 3, which is 2 bits. After that, the user can start the printing machine by on the switch. The counter will start counting the printed amount of copies, and at the same time, the comparator will compare the value of the input switch and the value of the counter. If both values are different, it will continue counting, and if both values are the same, a signal will be sent to the clock disabler and it will stop counting. The LED "STOP" will stop flashing and keep its light on.

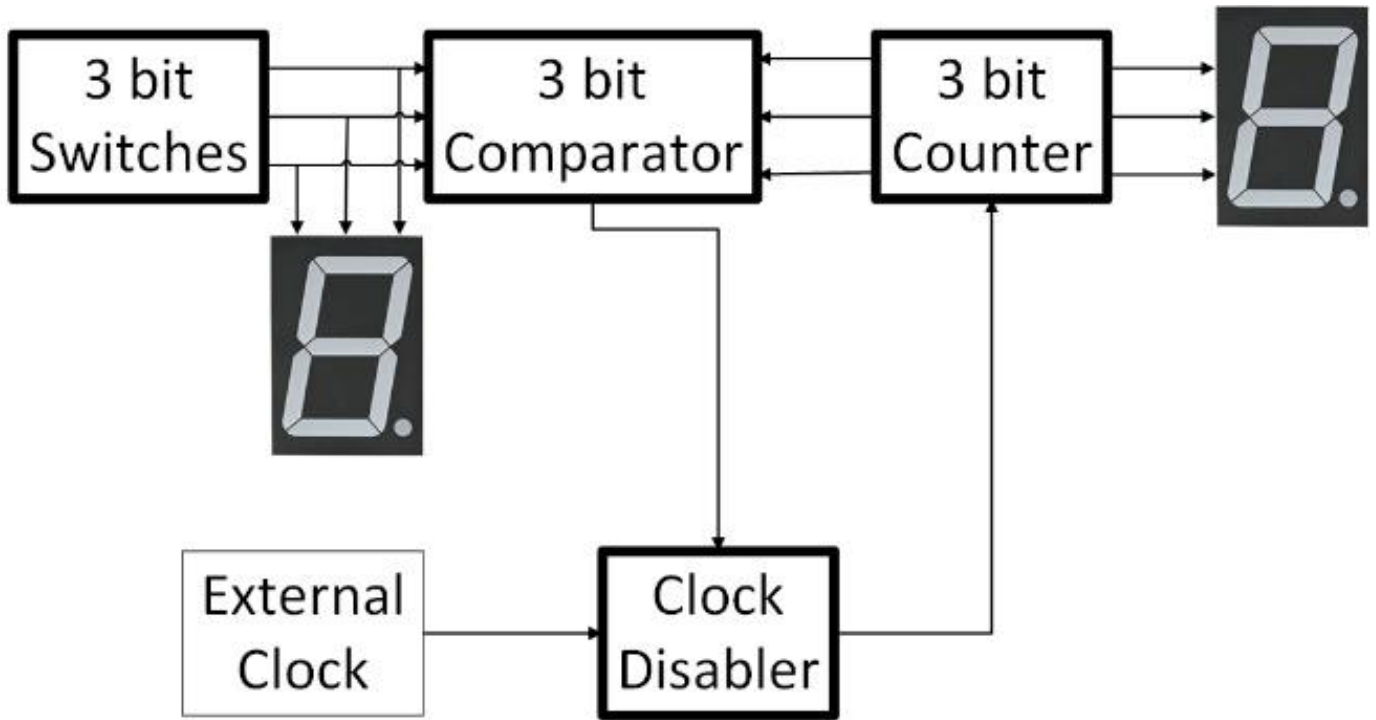


Figure 7.2 A block diagram of a complete 3-bit system

The pattern of our design is as shown in Figure 7.2 above. The main components are the 3-bit counter, 3-bit comparator, and clock disabler while the sub-component is some basic gates. The 3-bit counter is to count the amount of copies printed by the machine. The 3-bit comparator will compare the number entered by the user to the number counted by the counter. If the number becomes the same, the clock will be disabled, and the counter will stop counting. The required amount of copies and the printed amount of copies will be displayed on two 7-Segment LED displays.

To implement this, the user has to input the number of required copies to be printed by using three input switches. The number can be entered as 0 to 7, which is 3 bits. After that, the user can start the printing machine by turning on the switch. The counter will start counting the printed amount of copies, and at the same time, the comparator will compare the value of the input switch and the value of the counter. If both values are different, it will continue counting, and if both values are the same, a signal will be sent to the clock disabler and it will stop counting. The LED "STOP" will stop flashing and keep its light on.

8.0 PHYSICAL SYSTEM IMPLEMENTATION



Figure 8.1

XEROX 2-bit system on the ETS5000 Training Kits



Figure 8.2

XEROX 3-bit system on the ETS5000 Training Kits

1. Input Switch

There are two switches (2-bit) / three switches (3-bits) for the input number of copies required to print, represented in binary form. The user can input value from 0-3 (2-bit) / 0-7 (3-bit) only.

2. 2/3-bit up Counter

This counter will count based on the clock. It will start counting once the switch is on and the password is correct. It will stop counting when the comparator sends a signal to it. By using WINCUPPL code, we program the condition on it. Then the PLD program is burned into PLD ATMEL22V10 by using Wellon Universal Programmer.

3. 2/3-bit Comparator

This comparator uses three 2-input XOR gate to compare value from 2 sources, the input switch, and the counter. If the input from 2 sources are the same, the output will be 0 and followed by a NOT gate, the output will be 1. The same principle goes to the other 2 XOR gates. If the output of the 3 XOR gates is one after complemented, the output of the 3-input NAND gate which connected with the 3 XOR gate will be 0. So, this will active the clock disabler and clock signal will stop sending to the counter.

4. Clock Disabler

This clock disabler is set up using a 2-input AND gates. The input sources are the clock source and output of the comparator. If one of the sources is 0, the clock signal will be blocked.

5. 7-Segment Display

This component has three input from the input switch, and another input is set to 0. It will convert the binary code and display in the decimal form.

9.0 DISCUSSION

In summary, we are able to develop a PLD device after doing this mini project. Besides that, we are able to write a simple Hardware Description Language. Although we faced some obstacle such as wrong coding, as software engineering students, we have some basics in programming language. Hence, when we faced the problem of new programming language in this mini project which is the software coding language-WinCUPL, we also can easily understand and solve the programming coding problem. Furthermore, we proceed the project with care, we handle the apparatus such as PLD and wires with care. With a single mistake, the PLD would be damaged. We also double check the wire every time we implement it into the XEROX machine, it is to prevent any mistake that make the system not working as the debugging is tedious and time consuming. We learned that we must always prepare for the worst. The moment we found an error in our programme or wiring, we must stay calm and try to find the solution instead of arguing with each other.

Everything sure has its both sides. Hence, we think that the skill which we learned from this project can improve our future working knowledge. We should always keep updated the latest things in this world to confirm we are not be eliminated by this world under a highly competitive pressure.

10.0 CONCLUSION

In conclusion, this project was completed with success. Every component we used was working and function well. The whole project was conducted smoothly, despite there is some problem which was handled well by us.

From this project, we learned lots of experience and knowledge in term of practical and soft skills. Practice makes perfect, the application of our knowledge in real life is a boost to our experience. We are glad having a chance to carry out this project. We learned many skills included but not limited to using WinCUPL software as well as Wellon Universal Programmer & Tester software and uses and application of logic gates and DEEDS. Moreover, we learned team managing skill as well as teamwork. The project was carried out by a group, thus it is necessary for us to divide the task given to each of the member to make sure the project can be performed in an effective way.

As software engineering students, we have some basics in coding. Therefore, it is not a difficult task for us to code using WinCUPL although it is a new language that we did not learn in the past. However, one of our weaknesses is handling the components due to lacking of experience in such field. However, with the help of our lecturer and our friends, we were able to complete our task on time. It greatly helped us in our future learning as we had the chance to hand on designing a counting machine. As we know, counting machine is a basic component in our daily lifestyle. Hence, we had been exposed to the basic of real-life experience.

11.0 REFERENCES

Digital Logic Lab Manual. (2018). UTMJB: School of Computing, Faculty of Engineering.

Floyd, T. L. (2014). *Digital Fundamental, 11th Edition.* USA: Prentice Hall.

12.0 APPENDIX

Program source code of 2-bit XEROX System:

```
Name XeroxMachine2BIT ;
PartNo 03 ;
Date 19/12/2019 ;
Revision 01 ;
Designer Engineer ;
Company UTM ;
Assembly None ;
Location ;
Device G22V10 ;

/* ***** INPUT PINS ***** */
PIN 1 = clk ;          /* clock */
PIN 2 = reset ;       /* reset */
PIN 3 = preset ;      /* preset */
PIN 4 = a0 ;          /* Comparator A */
PIN 5 = a1 ;          /* */
PIN 7 = b0 ;          /* Comparator B */
PIN 8 = b1 ;          /* */
PIN 10 = startPrt ;   /* Start Printing */

/* ***** OUTPUT PINS ***** */
PIN 17 = diffCmp ;    /* XOR (A B not equal HIGH) */
PIN 18 = sameCmp ;    /* XNOR (A B equal HIGH) */
PIN 21 = q0 ;         /* output counter */
PIN 22 = q1 ;         /* output counter */

/***** Function Comparator***** */
sameCmp = !(a0$b0)&!(a1$b1);
diffCmp = !sameCmp ;

/**** Function Clock Enabler ***** */
clkEn=startPrt & diffCmp;

/**** Function Counter 2 Bit UP ***** */
field count =[q1..0];
$define s0 'b' 00
$define s1 'b' 01
$define s2 'b' 10
$define s3 'b' 11

count.ar=reset;      /* connect reg AR to reset (Asyn Mode) */
count.sp=preset;     /* connect reg AR to preset (Syn Mode) */

sequence count{
    present s0 if clkEn next s1;
        default next s0;
    present s1 if clkEn next s2;
        default next s1;
    present s2 if clkEn next s3;
        default next s2;
    present s3 if clkEn next s3;
        default next s3;
}
```

Program source code of 3-bit XEROX System:

```
Name XeroxMachine3BIT ;
PartNo 04 ;
Date 19/12/2019 ;
Revision 01 ;
Designer Engineer ;
Company UTM ;
Assembly None ;
Location ;
Device G22V10 ;

/* ***** INPUT PINS ***** */
PIN 1 = clk ;          /* clock */
PIN 2 = reset ;       /* reset */
PIN 3 = preset ;      /* preset */
PIN 4 = a0 ;          /* Comparator A */
PIN 5 = a1 ;          /* */
PIN 6 = a2 ;          /* */
PIN 7 = b0 ;          /* Comparator B */
PIN 8 = b1 ;          /* */
PIN 9 = b2 ;          /* */
PIN 10 = startPrt ;  /* Start Printing */

/* ***** OUTPUT PINS ***** */
PIN 17 = diffCmp ;   /* XOR (A B not equal HIGH) */
PIN 18 = sameCmp ;   /* XNOR (A B equal HIGH) */
PIN 21 = q0 ;        /* output counter */
PIN 22 = q1 ;        /* output counter */
PIN 23 = q2 ;        /* output counter */

/***** Function Comparator *****/
sameCmp = !(a0$b0)&!(a1$b1)&!(a2$b2);
diffCmp = !sameCmp ;

/**** Function Clock Enabler *****/
clkEn=startPrt & diffCmp;

/**** Function Counter 3 Bit UP *****/
field count =[q2..0];
$define s0 'b' 000
$define s1 'b' 001
$define s2 'b' 010
$define s3 'b' 011
$define s4 'b' 100
$define s5 'b' 101
$define s6 'b' 110
$define s7 'b' 111

count.ar=reset;      /* connect reg AR to reset (Asyn Mode) */
count.sp=preset;     /* connect reg AR to preset (Syn Mode) */

sequence count{
  present s0 if clkEn next s1;
  default next s0;
  present s1 if clkEn next s2;
  default next s1;
  present s2 if clkEn next s3;
  default next s2;
  present s3 if clkEn next s4;
  default next s3;
  present s4 if clkEn next s5;
  default next s4;
  present s5 if clkEn next s6;
  default next s5;
  present s6 if clkEn next s7;
  default next s6;
  present s7 if clkEn next s7;
  default next s7;
}
```