UTM UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

SECR1013-08 - Digital Logic

Mini Project : Programmable Logic Device (PLD)
Photocopying (Xerox) Machine

NO.	NAMA	STUDENT ID
3	NUR NABILAH BT YUSMAN	A19EC0136
5	NOR FARAHZIBA BINTI HAMADUN	A19EC0123

SECTION	:	08
COURSE CODE	:	1 SECR
LECTURER’S NAME	:	Ms. Rashidah bin Kadir
DATE OF SUBMISSION	:	1st January 2020

DEDICATION & ACKNOWLEDGEMENT

We have taken steps in this project. However, this would not have been possible without the support and assistance of many individuals and UTM. We would like to express our sincere thanks to all of them.We are indebted to our lecturer, Ms. Rashidah for her constant guidance and supervision and to provide the necessary information on this project as well as their support in completing the project.We would also like to thank our classmates who willingly helped to assist us in their ability to complete this project.

Table of contents

1.0 The Background

This mini project will implement 2 different components on single GAL device, those components.

- 2-bit up counter
- 3-bit up counter
- Clock Disabler

2.0 The Problem

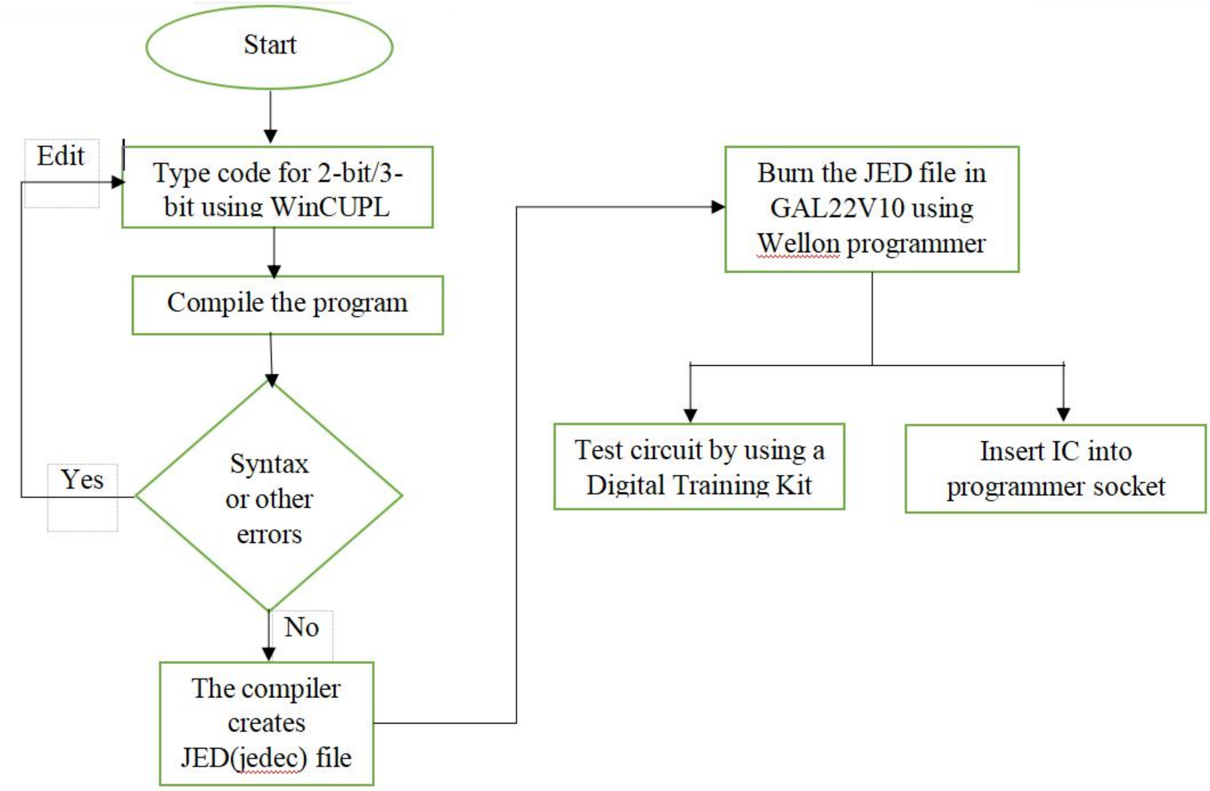
Firstly, the user will enter the number of copies needed. Then, the counter will start to count the amount of the copies that had been photocopied. Lastly,once the required amount of copies produced, the machine will stop.

3.0 Objectives

The objective of the laboratory :

- To introduce us about the development of a PLD device.
- To introduce us a simple Hardware Description Language.

4.0 The flowchart


Start
Edit
Type code for 2-bit/3-bit using WinCUPL
Compile the program
Yes
Syntax or other errors
No
The compiler creates JED(jedec) file
Burn the JED file in GAL22V10 using Wellon programmer
Test circuit by using a Digital Training Kit
Insert IC into programmer socket

5.0 Components

Component used in the laboratory :

- Switch
- Comparator
- Counter
- Clock Disabler

6.0 Materials and Software's Used\

- **Breadboard**
- **ETS-5000 Digital Training Kit**

6.1 GAL 22V10 (ATMEL PLD)

A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits. Unlike integrated circuits (IC) which consist of logic and have a fixed function, a PLD has an undefine function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed (reconfigured) by using a specialized program.

6.2 Wellon Universal Programmer & Tester

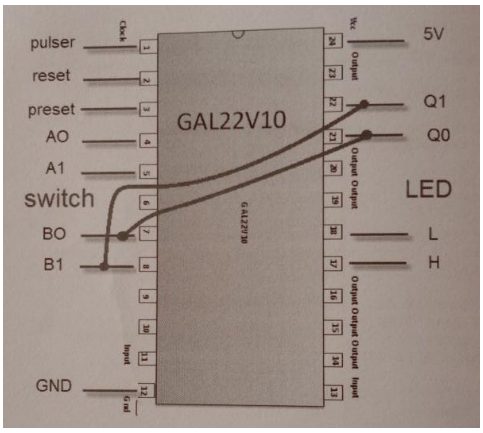
Burn the .JED file to GAL 22V10 using Wellon Pogrammer. Under ***Edit Auto*** field, check ***Erase/Blank*** and ***Program/Verify***. This option will blank check the IC to determine whether it is blank, if not it will erase it, then program the IC and verify that it's content is the same as in the programmer buffer after the programming process. The entire tep will be executed in sequence automatically. To execute the sequence selected in ***Edit Auto*** field, click button Prog. All the sequence will be executed and the result will be displayed.

6.3 WinCUPL 5.0 Software

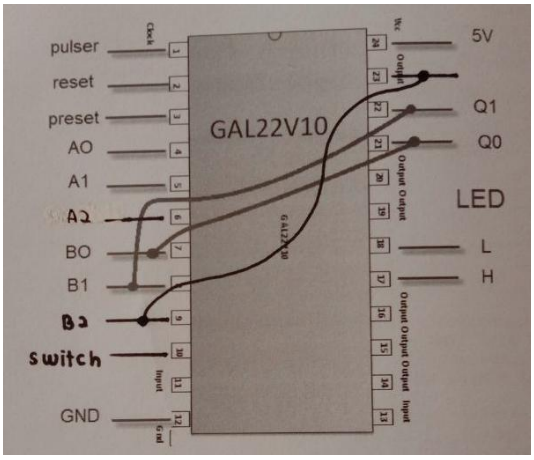
Use WInCUPL to generated the code. After a successful compilation, a jedec (.JED) file will be created. This file wil be needed to program the PLD device by using a universal programmer such as Wellon Programmer.

7.0 Circuit Implementation

2-Bit Circuit : A0 and B0 is LSB while A1 and B1 become MSB.



3-Bit Circuit : Difference between this two circuit is, for 3-Bit add another input A2 and output B2, it is MSB. The others pin remain same like 2-Bit.



8.0 Physical System Implementation

Steps:

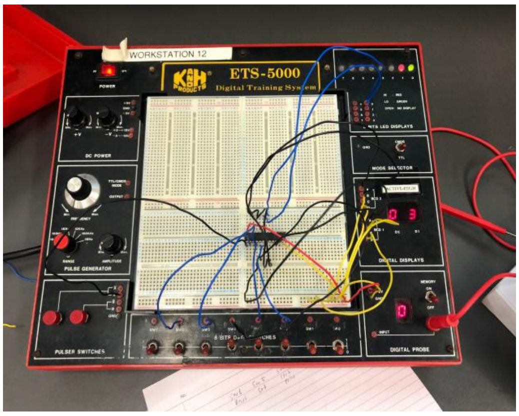
i. Set the Asynchoronous pin switch input to the correct setting. Make sure the initial output of the counter in ZERO.

ii. Set the number of required copies by flipping the switch (A1A0 = 10; 2 copies)

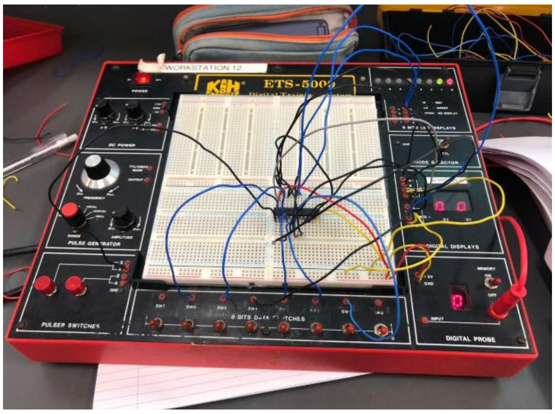
iii. Press the pulser switch, it will increment the counter (00 → 01 → 10)

iv. When the counter output equals to 2 (B1B0 = 10) the output of the comparator will change.

Note : For 3-Bit have another input A2 and output B2 (eg: A2A1A0 = 110; 6 copies)



2-Bit



3-Bit

9.0 Discussion

9.1 Summarizes of the whole project

In this project, we have to solve the problem of photocopy which was to make sure the number of copies entered must be exactly same with the number of copies produced. We use counter to determine the amount of copies that had been entered and the comparator will compare the amount of copies entered with the amount of copies produced.

For part one we do the 2-bits up counter so firstly we type the coding by using WinCUPL. Then compile the program and it will generate to .JED (jedec) file. After that, by using Wellon programmer in the lab , we burn the .JED file to program the GAL22V10D. Next, we test the circuit by using a Digital Training Kit by connecting the IC as instructed in the lab manual and comparator output to LEDs. Once finished, to test it , we begin by setting up the Asynchronous pin switch input to the correct setting and must make sure the initial output is zero. Then, set the required amount of copies by flipping the switch and press the pulser switch, we can see the increment of the counter. Once the counter output is equal to the comparator, it will change the LED bulb colour as a signal to stop the counting. For ‘L’ , it turn from red to green while ‘H’ turn from green to red.

For the second part , we just have to change the 2-bits up counter to 3-bit up counter. Using the design in 2-bits up counter as guidance, make design for the 3 bits comparator. So overwrite the generated code turn it into 3-bits up counter.

For this second part just either one of the comparator output must be connected to the clock disabler.

9.2 Strength/achievements

In this project, although we had faced many problems but in the end we successfully complete this project. We might had some problems when implementing the circuit on ETS5000 Training kit since the wire connected look so messy but we didn't gave up and collaborate with each other until we got the correct result. At first, our IC not functioned but none of us notice it earlier so we took so much time with the defective IC. But keep going on doing the project with patient. For us, it is a great achievement when the got the result we wanted regarding to this project since we are the last group to succeed and put so much effort on it. Our strength can be described when we working with each other as one team and faced the problem together without losing our hope in order to accomplish our goal.

9.3 Problem faced

During this project, we faced so much problem. First, our IC was defective. So we just wasting time thinking the error when implementing the circuit. Second, we forgot the basic when putting the IC on the Training kit. We connect the IC in the wrong ways. Third, we lack of knowledge, we took time to understand how this Xerox Machine work. We ask help from others to detect our problems and teach us until we completely understand the project.

9.4 New function suggestions for improvement and future works

As the improvement and future work of our system, we suggest to use Deeds Software for the implementation. The design is the same like the previous one but just extend the design by adding more component with different feature like Encoder, Decoder, Mux, and Demux. We can add authentication or security such as password by using Decoder or Encoder for the pattern recognition. Other ideas are add more indicator for count-up and count-down, different source of input paper or output printing stacks can be selected (Mux or Demux), online monitoring and buzzer.

10.0 Conclusion

From the project , we learn how the photocopy work and see how each components like counter,comparator , clock disabler and switch performing their role. We also gain knowledge about the development of a PLD device. PLD must be programmed or reconfigured first using a specialized program before it can be used. Furthermore, we also learn a simple Hardware Description Language (HDL). HDL is the language used as source code for the logic compiler. Other than that, the soft skill we gain is the way to interact and collaborate to each other. Everybody have different idea and personalities , so in this project we have to accept each other opinions and try to adapt the team member way of working. We also divided our task to make the project more easier to carry on and to make the working more systematic.

Digital logic is the basic of digital computing so we can gain better understanding about the communication between circuit and hardware within a computer. Video games, computer and even calculator were electronic devices that embedded by digital logic. Through the simple input or algorithm to build computer hardware , the hand-on experience can be gain. We learn how the data on computer like document and movie can be store just by simple input of one and zeros. Moreover, digital logic is important for electronic and computer engineer in order to perform their job since it is the key for many career in engineering. This show how important we as a Computer Science students to take Digital Logic subject as advancement and future learning.

11.0 Reference

Yusoff, A. B., Salleh, M., Rohani, M. F., & Isnin, I. F. (2018). *Digital Logic.* Johor Bahru: School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia.

12.0 Appendix

12.1 2-bit XEROX System

```
Name Lab 4 (Mini Project ;
PartNo 00 ;
Date 18/12/2019 ;
Revision 01 ;
Designer Engineer ;
Company UTM ;
Assembly None ;
```

```
Location Digital Logic Lab ;
Device ATF22V10CQZ ;

/* *************** 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;
}
```

12.2 3-bit XEROX System

```
Name Lab 4 (Mini Project) ;
PartNo 00 ;
Date 18/12/2017 ;
Revision 01 ;
Designer Engineer ;
Company UTM ;
Assembly None ;
Location Digital Logic Lab ;
Device ATF22V10CQZ;

/* *************** 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 ;
PIN 23 = q2 ;       /* output counter */

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

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

/*** Function Counter 2 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; }
```