

Programmable Logic Devices

Acknowledgements: Developed by Jesus Casas, Faculty of Austin Community College, Austin, Texas

Lab Summary: This lab introduces CUPL, a programmable logic description language, which is used to program PLDs such as Atmel's ATF16V8, ATF20V8, and ATF22V10.

Lab Goal: In this lab, you will use Atmel's WinCUPL to develop a JEDEC file which contains the Boolean logic equations that can be used to program one of Atmel's ATF16V8, ATF20V8, or ATF22V10 PLDs. The programming is accomplished by using a universal programmer or a PLD programmer that is capable of writing to the target PLD.

Learning Objectives

1. Differentiate between Programmable Logic Devices (PLDs), Complex Programmable Logic Devices (CPLDs), and Field Programmable Gate Arrays (FPGAs).
2. Use a high-level hardware description language to write a PLD test program.
3. Use a universal programmer or a PLD programmer to test a programmable logic device for functionality.

Grading Criteria: The criteria for grading is left to the discretion of the instructor.

Time Required: 3 hours

Special Safety Requirements

Static electricity can damage the PLD device used in this lab. Use appropriate ESD methods to protect the devices. A grounded wrist-strap is provided in the parts kit for this circuit. Be sure to wear it at all times while handling the electronic components in this circuit. The wrist strap need not be worn after the circuit construction is complete.

No serious hazards are involved in this laboratory experiment, but be careful to connect the components with the proper polarity to avoid damage.

Lab Preparation

- Read the WRE PLD Narrative Module.
- Read this document completely before you start on this experiment.
- Gather all components.
- Print out the laboratory experiment procedure that follows.
- The student needs to have Boolean Logic equations which represent the combinational logic circuit the PLD will replace.



Equipment and Materials

Each team of students will need the equipment specified below.

Equipment	Quantity
PC with Atmel's WinCUPL installed.	1
PLD programmer or a Universal Programmer	1
PLD (Atmel's ATF16V8, ATF20V8, or ATF22V10)	1

Introduction

There are many types of programmable logic devices available ranging from the small devices, which can replace a few logic gates, to very large Field Programmable Gate Arrays which can hold an entire processor core. Programmable logic can be placed into three categories: PLDs, CPLDs, and FPGAs.

PLDs are at the low-end of the spectrum and go by the names of Programmable Logic Array (PLA), Programmable Array Logic (PAL), and Generic Array Logic (GAL). These devices usually replace a handful of discrete logic chips. Inside each PLD is a set of fully connected macrocells. These macrocells are typically comprised of some amount of combinational logic and a flip-flop. When a PLD is programmed, these macrocells are configured to represent a Boolean logic equation. PLAs and PALs are generally recognized as one-time-programmable chips while GALs are erasable and re-programmable.

Complex Programmable Logic Devices are the next step up from PLDs. CPLDs allow the implementation of more logic equations, more input/output pins, and more complicated designs. Programmability is achieved by a switch matrix which can be reprogrammed with new equations.

Field Programmable Gate Arrays have three key parts to their structure: logic blocks, interconnects, and Input/Output blocks. Because of this structure, FPGAs are more flexible than CPLDs but are not able to compete with the logic density of CPLDs.

The process of implementing a PLD involves a description of the hardware's structure and behavior written in a high-level hardware description language. The code is then compiled and downloaded to the PLD. The PLD is then ready for use. Another method to describe the hardware is by using schematic representation, but this method becomes less popular as designs become more complex.

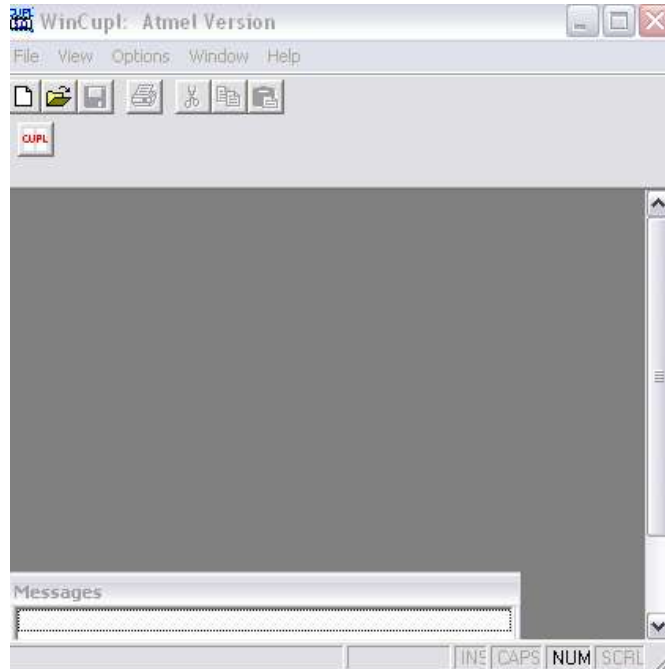
CUPL Description Language

CUPL Programmable Logic Description Language is a compiler that converts logic equations into a Fuse map (Joint Electron Device Engineering Council, or JEDEC, file) for the PLD programmer. Pins can be named, logic equations are simplified, Karnaugh maps can be employed, and state machine techniques can be used. This lab demonstrates the use of WinCUPL which can be downloaded from the Atmel website.



Lab Procedure

1. If you do not have the WinCUPL software loaded on your PC, access the Atmel website at <http://www.atmel.com/>. Look up WinCUPL and click on the sentence for the free download.
2. Start the WinCUPL software. You should see the following screen.



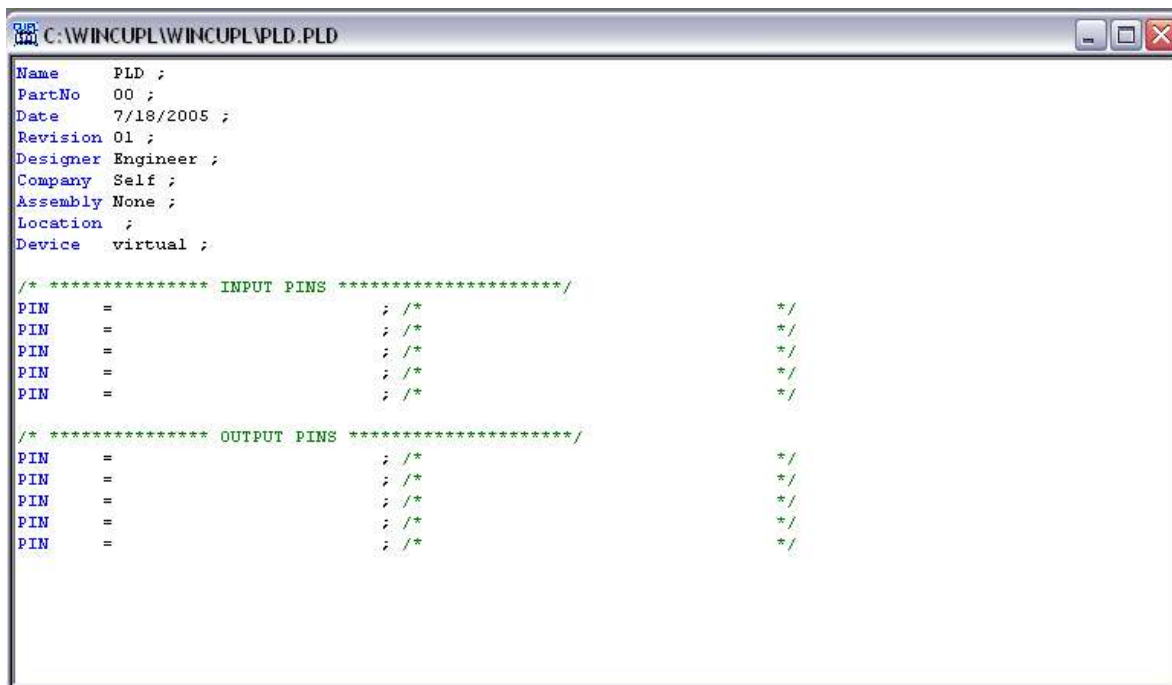
3. Select “File” from the menu bar, go to “New”, then click on “Design File”. The following Design Properties screen should be displayed.

Design Properties	
Name:	Name
PartNo:	00
Date:	7/18/2005
Revision:	01
Designer:	Engineer
Company:	Self
Assembly:	None
Location:	
Device:	virtual

4. In the Design Properties, input a name for this design. Note that this will also become your filename.



5. Change the following parameters:
 - a. Enter your name as the Engineer
 - b. Enter your instructor's name as the Company
 - c. Enter your class number as the Location
6. When you are finished, click OK.
7. Next, a series of windows requiring input will “pop” up.
 - a. Input the number of input pins needed by your logic circuit
 - b. Input the number of output pins needed by your logic circuit.
 - c. Leave the number of pinnodes at 0.
8. The following window should be displayed at this point. (Note: Your screen will be slightly different because it will reflect your inputs.)



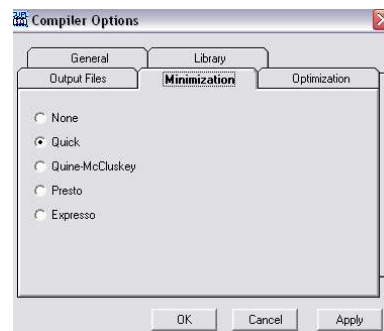
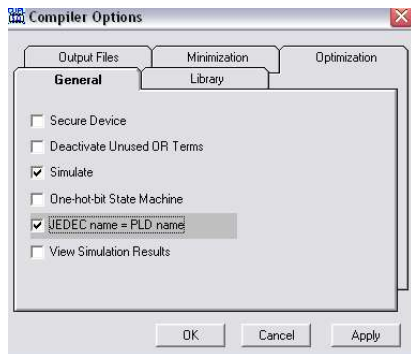
```
Name      PLD ;
PartNo    00 ;
Date      7/18/2005 ;
Revision  01 ;
Designer  Engineer ;
Company   Self ;
Assembly  None ;
Location  ;
Device    virtual ;

/* ***** INPUT PINS ***** */
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*

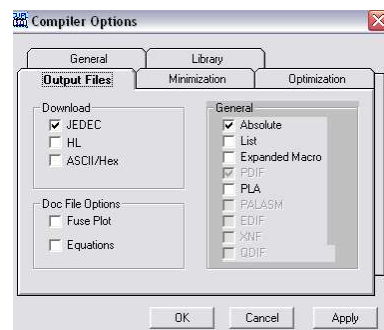
/* ***** OUTPUT PINS ***** */
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*
PIN      =          ; /*
```



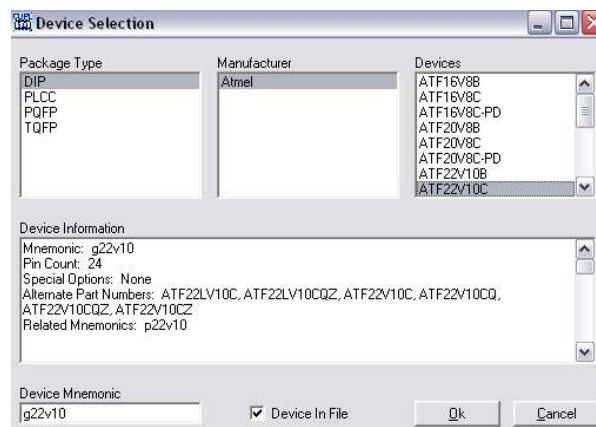
9. Select “Options” from the menu bar and then click on “Compiler”.
10. Ensure the settings reflect the following.
 - a. In the General tab: JEDEC name = PLD name should be checked.
 - b. In the Minimization tab: Quick should be on.



- c. In the Output tab: JEDEC, Fuse Plot, Equations, and Absolute should be checked.



11. Close the Compiler Options and open the Devices Options from the Options menu.
 - a. Select your device from the list at the right.
 - b. Ensure the Package Type and Manufacturer match your device.





12. At this point, the chip pin numbers that will be used need to be associated. Also, the logic equations need to be inputted. The most common logic operators are shown in the following table. An example of inputted equations is shown at the end of this procedure.

Operator	Function	Example	Precedence
!	NOT	!A	1
&	AND	A&B	2
#	OR	A#B	3
\$	XOR	A\$B	4

13. After the information has been inputted, save the file with a “.pld” extension.
14. At this point, your CUPL program is ready to be compiled.
- Select “Run” from the menubar, then go to “Device Dependent Compile”.
 - CUPL will now try to compile your program. If errors are noted, check the “Message” window to see what the errors are. Once the errors are corrected, your program will compile.
15. Once your program has compiled, a list of files will be displayed. One of the files will have a “.doc” extension, and the other will have a “.jed” extension.
- The “.doc” file contains the Fuse Plot and the chip pin designation.
 - The “.jed” file will be used by the programmer to program the PLD.
16. Attach a Universal Programmer or a PLD programmer to the PC. Follow the programmer’s instructions to select the correct PLD and to program it with the “.jed” file provided by CUPL.
17. Test your PLD for functionality.



```

C:\WINCUPL\WINCUPL_DECIMAL_DECODER.PLD

Name      decoder ;
PartNo    00 ;
Date      7/16/2005 ;
Revision  01 ;
Designer  Engineer ;
Company   self ;
Assembly  None ;
Location  ;
Device    22V10 ;

/* ***** INPUT PINS ***** */
PIN 2 = D0; /* LSB */
PIN 3 = D1; /*
PIN 4 = D2; /*
PIN 5 = D3; /* MSB */

/* ***** OUTPUT PINS ***** */
PIN 14 = a1; /* a segment of one's place seven-segment display */
PIN 15 = b1; /* b segment of one's place seven-segment display */
PIN 16 = c1; /* c segment of one's place seven-segment display */
PIN 17 = d1; /* d segment of one's place seven-segment display */
PIN 18 = e1; /* e segment of one's place seven-segment display */
PIN 19 = f1; /* f segment of one's place seven-segment display */
PIN 20 = g1; /* g segment of one's place seven-segment display */
PIN 21 = b2; /* b and c segment of ten's place seven-segment display */

/* ***** Logic Equations ***** */
a1 = (D3&D2&D1&D0)#{D3&D2&D1&D0}#{D3&D2&D1&D0}#{D3&D2&D1&D0};
b1 = (D3&D2&D1&D0)#{D3&D2&D1&D0}#{D3&D2&D1&D0}#{D3&D2&D1&D0};
c1 = (D3&D2&D1&D0)#{D3&D2&D1&D0};
d1 = (D3&D2&D1&D0)#{D3&D2&D1&D0}#{D3&D2&D1&D0}#{D3&D2&D1&D0};
e1 = D0#{D3&D2&D1};
f1 = (D3&D2&D1)#{D2&D1&D0}#{D3&D1&D0}#{D3&D2&D1}#{D3&D2&D0};
g1 = (D3&D2&D1)#{D3&D2&D1}#{D3&D2&D1}#{D3&D2&D1&D0};
b2 = D3#{D2&D1};

```