

Pins marked IN are inputs; pins marked I/O can be either input or output pins. Pin 1 can also serve as the clock if the clocking feature is used in some of the equations.

**Wincupl .pld files.**

The format of a pld file is as follows:

* Header. Key points are:
  + Each line starts in column 1 with a wincupl keyword. The line terminates with a semicolon.
  + It is essential that the header include a “Device” line that specifies the generic device family. In EECS 346 this will always be g22v10.
* Declarations. This section specifies names for the pins.
  + Comments are used to specify which pins will be input and which output.
  + The format of a pin declaration is:
    - the keyword “PIN” starting in column 1
    - one or more spaces
    - the pin number
    - one or more spaces
    - =
    - one or more spaces
    - the symbolic name
    - one or more spaces
    - semicolon
    - optional descriptive comment
* Equations. The equations use ! (NOT), & (AND), # (OR), and $ (XOR).

(NOTE: wincupl equations can be used to define clocked behavior of the outputs, in which case output symbolic names have extensions. For example, one could write latch.d to represent the d input for an output pin called latch. We will not use that capability in EECS 346.)

**Compiling and Testing a pld File.**

The compiler produces several output files, and the simulator requires one or more files. I usually make a new folder for each project so that all the files – pld, jed, listing, etc. – can be together and easily identifiable.

To compile and test your pld file:

* Start wincupl.
* Go to the Options/Device and select the appropriate device. For EECS 346 this will be DIP package type, ATMEL manufacturer, and ATF22v10B for Device.
* Load your pld file.
* In Options/Compiler check JDEC in the Output Files tab. You can experiment with options in the other tabs; I usually check the Simulate option in the General tab.
* Click “Device Dependent Compile” on the menu bar. If there are errors, fix them and try again; errors should be listed in the window at the bottom.
  + Note that if you have prepared a simulation file (see next paragraph) you may get errors because the simulation produced different outputs than your simulation file specified. These will be flagged as CSIM errors. So be sure to read the error messages carefully.
  + The compilation process produces several new files. The one used to program an actual 22V10 circuit is the jed file. The others contain summary information about the project. It is useful to look at these at least once so you know what they contain.

To test your equations:

One of the icons on the menu bar is WinSim.

* The vertical axis is labeled at the left by pin names.
* The horizontal axis is divided into time units.
* Simulation requires two additional files, a sim file and a si file, so if you click before you create these file you will get an error message and only see an empty simulation window.
  + The compiler produces the sim file when your pld file is successfully compiled.
  + You need to create the si file.
    - The si file starts with header information that matches the header section of the pld file.
    - Next is a section specifying the order that you want to see the signals listed in the simulation screen. The signals you list in the ORDER line will be displayed in the left-most column.
    - Finally, you provide vectors of values for the selected input and output pins, using the same order as specified in the ORDER statement.
    - For simple circuits, like we use in 346, each vector will be displayed in one time slice on the simulation screen. The symbols that we will use in 346 are:
* 0 or 1 – specifies a value for an input.
* X – specifies a don’t care for either input or output
* L or H – the value that the user expects for an output
* \* - simulator determines what the output value is

One way to test your equations is to provide a set of 2n vectors, where n is the number of input signals and use \* for each of the output signals. Then the simulator will simply generate the output signals, which you can check visually. Another way is to use H and L. In this case, the simulator will compute the correct value and mark the graph with red if the user-provided expected value does not match. After you clicked the “Device Dependent Simulate” icon from wincupl and dismiss the error box, the simulator window will appear. In that window click the “simulate” icon to generate the test results.

After programming a 22V10 circuit, you can test again using a breadboard wired just for the 22V10 circuit, using wires to set the inputs and a meter or scope to check the outputs.

**Loading the file to the 22V10.**

Once you have specified the circuit the way you want and compiled and tested it in wincupl, use the .jed file produced by the compiler to program a 22V10 circuit with the device programmer. Perform the following steps:

1. Turn on the device programmer.
2. Start the programmer software.
3. Select the device (manufacturer and device type) to match the actual 22V10 circuit that you have. For example, if the 22V10 is marked ATF22V10C, the manufacturer is ATMEL, and the device should be set to ATF22V10C.
4. insert your 22V10 into the zif socket.
5. Load the .jed file.
6. You may need to blank check and erase the 22V10
7. Click on the program command in the menu bar.

You should see the standard progress bar for programming and verifying the device. Assuming the operation was successful, you can now remove the 22V10 from the programmer and insert into your project.