

Driving the Daisy (Chain) or Using the Lattice Compiler & Daisy-chain Download

N.J.Gunton 3/00 02/02/04

Grouping	Individual / Team
Prerequisites	Knowledge of VHDL & EDIF
Courses	CRTS2 CSE EE
Requirements	NT/W2k/XP running the Lattice Tools, Security key, download cable, Target board, Lattice EDIF file & power supply
Summary	Running the Lattice 'fitter' and download software. Programming CPLDs
Objectives	To understand the process of programming CPLDs, the synthesis of digital designs for target devices, To gain an awareness of hardware specific design issues such as critical paths in the circuit. 'In -system testing. An understanding of timing problems.

!!!WARNING : READ THIS BEFORE COMMENCING!!!

The security keys required by the Lattice software are sensitive devices and easily deprogrammed. The easiest way to de-program them is to reboot the system into Linux with the keys still attached. The turnaround time for having the keys reprogrammed is 14 days if we are lucky.

Only fit the security key after the Microsoft operating system is up and running. When the key is plugged into the parallel port you can run the Lattice software. Ensure that the number written on the security key matches the id written on the top of the machine in front of you. The Lattice key is fitted first followed by the download connector (the blue connector with an RJ45 socket in one end and a parallel port connector in the other.

The download cable does not need to be connected until you have synthesised your code and are ready to 'fit' it.

BEFORE connecting the power supply to the target board

- 1) Check the polarity of the power connector. The tip should be set to positive (+ve).
- 2) The voltage should be set to 9 volts.

1. Creating your EDIF file

- If you have already created an EDIF file then you can skip to the next section.

This is a quick summary of the steps that you must undertake in order to create an EDIF file suitable for creating a project in the Lattice Synthesis tool. It assumes that you are logged into the linux system in the lab and uses an example from the `/usr/local/alliance/examples` directory.

- i) Change directory to your VHDL subdirectory and create a new directory (folder) called `edif_example`.

```
mkdir edif_example <enter>
```

- ii) Change into this directory so that it becomes your current working library.

```
cd edif_example <enter>
```

- iii) copy the example into your current directory.

```
cp /usr/local/alliance/examples/t_bird_hand.vbe ./ <enter>
```

- iv) Create the EDIF file. you will use the `sxconv` utility. See `man sxconv` for details. In the meantime

```
sxconv -b t_bird_hand <enter>
```

- v) Check for error messages and check that the resulting file, `t_bird_hand.edf` is about 14kbytes in size. If it is very much smaller than this then ask your lab tutor for help.

```
ls -l t_bird_hand.edf <enter>
```

2. The Lattice Synthesis Tool

The Lattice synthesis tool, or `fitter`, takes an electronic design and maps it into a programmable device from the family of devices provided by the Lattice Semiconductor Corporation. The design can be in a variety of languages. For simplicity we use EDIF as the transfer language.

2.1. Starting the Software

With reference to the comments at the beginning of this worksheet, log into a 'windows' machine that has the software installed. Ensure that the security key is plugged in, along with the blue download connector. Plug the board into the download cable and power up the board. You might find that LEDs light up on the board. This is usually because the board is already programmed with someones design. You will overwrite this with your own design.

Select Programs → Lattice Semiconductor → IspExpert Compiler. This should launch the program.

2.2. Creating a new project

From the menu bar, select Project → New . You should get a pop-up window, shown on the following page. You will need to :

- Select the correct drive and folder for the `.edf` file that you have created. When you have done this, the file(s) will appear in the selection window.
- Select the correct settings for importing an EDIF file. You need to specify the names of the power rails, bus details etc. The correct settings are all shown in the screen shot below.
- You need to select the type of the device that you are going to program. Look at the chip on the board. It is most likely to be a '1016E', but some boards are fitted with '1016's or '2032's which are programmed differently. So check the device fitted to the board and select it from the device list, (screen-shot below). The package is always 'LJ44', the device speed can be read from the chip.
- Select the file that you wish to import to the synthesis tool, this will provide a default project name which you may edit if you wish.

Once you have completed the above steps, click `OK`. If every thing is correct then the EDIF file will be imported and a new project created. If there are any errors in the EDIF file then these will be reported in a window. If there are errors then correct them and reopen the project. If you reopen the project then it will keep the settings that you made in the earlier steps.

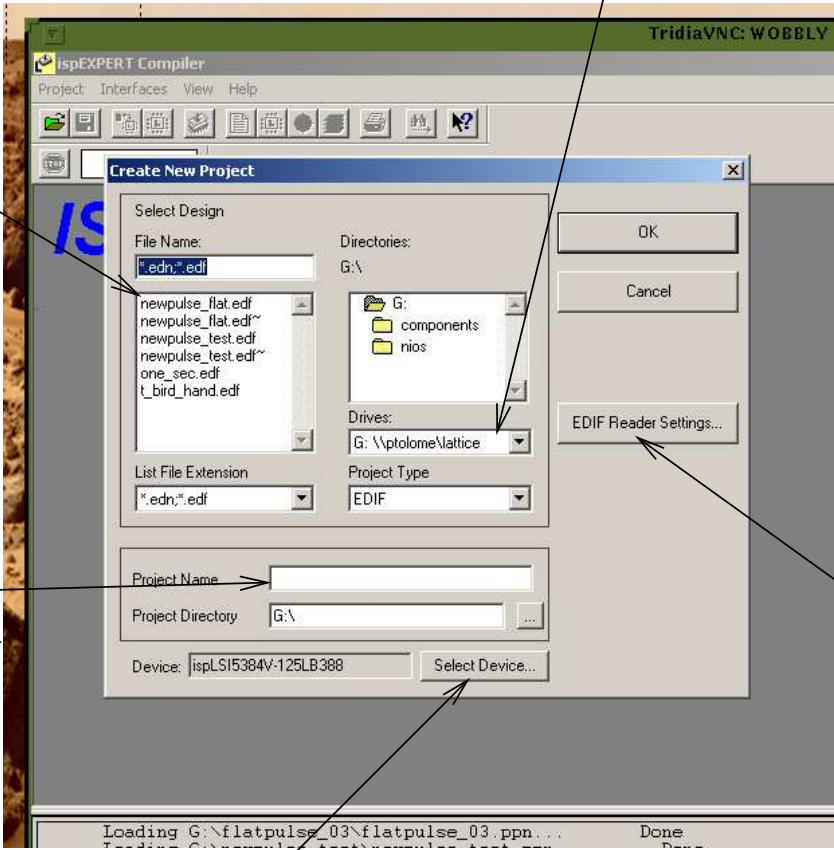
1) Select the drive and folder that contains the .edf file you want to use for the project.

2) Select the correct settings. See below.

3) Select the correct device. See below.

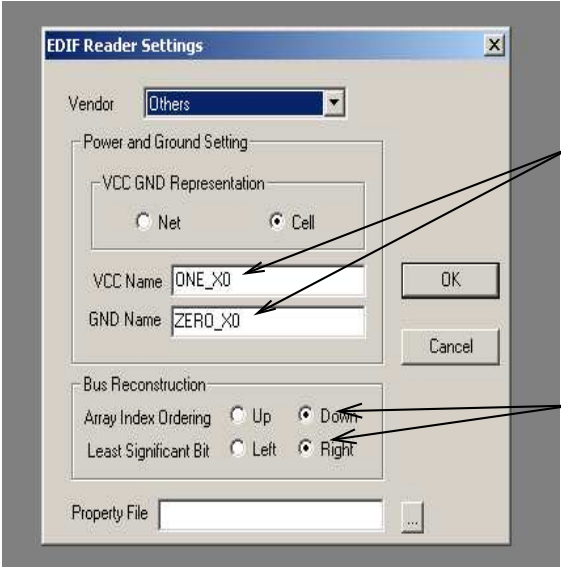
4) Select the .edf file that holds your design

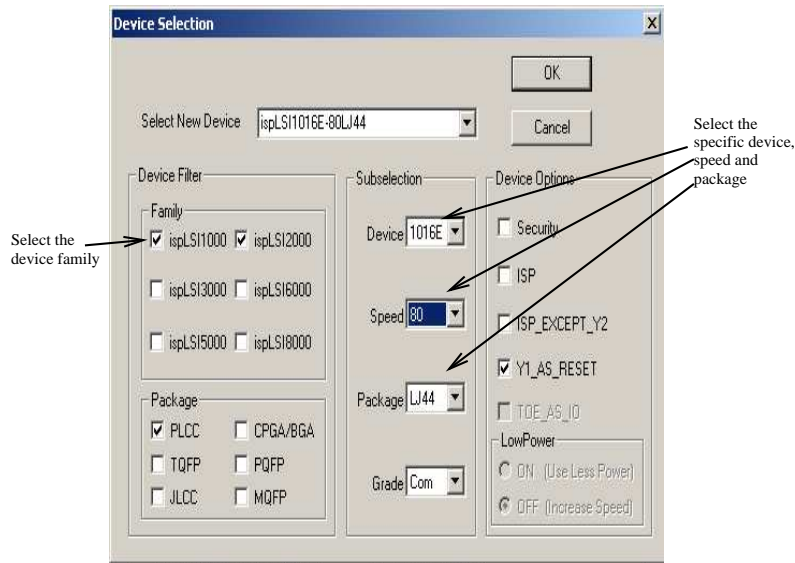
5) Project is named after your .edf file. Can be changed.



note that the last character is a zero, not the letter 'o'

You must set these as shown. Why?





2.3. Compiling your design

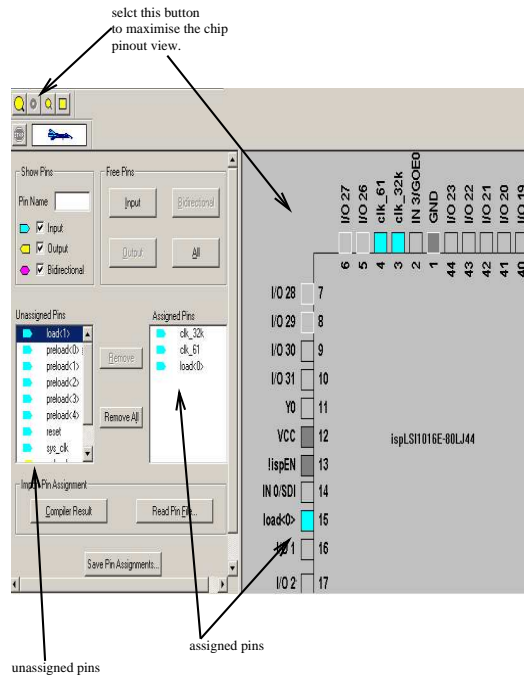
Before you compile your design, you must specify the relationship between the physical pins on the device and the inputs and outputs of your design. The majority of the boards have 8 inputs from dip switches and 8 outputs (to LEDs) available. Two of the boards have been modified to provide 2 additional clock inputs and 6 general purpose (unconnected) pins. The boards also have either one or two seven-segment displays on them.

The dip switches are connected to I/O pins 0 - 7 and the LEDs are connected to I/O pins 8 -15. The seven-segment displays are connected to I/O pins 16 - 23 and 24 -31 except for the modified boards which have had the last seven-segment display removed. These have a 60hz clock on I/O pin 24, a 32khz clock on I/O pin 25 and the remainder are unconnected.

A screen-shot of the pin assignment window is shown below.

If you wish to use the 1Mhz board clock then you MUST assign the clock input of your design to pin 'Y0'. If your design needs a slow or very slow clock then you must use one of the modified boards and assign your clock input to either I/O pin 24 or 25 as appropriate.

- i) Select **Assign** → **Pin locations** from the menu bar. Maximise the pop-up window and select the icon that looks a bit like a yellow circular target to maximise the view of the device. Use the scroll bars to view all of the device.
- ii) Drag an input from the list on the left and drop it on the pin that you wish to assign the input or output to. Repeat this until you have assigned all of your inputs and outputs. This can be a very frustrating process as, although the software recognises when you are over a pin, there is only a very small 'drop zone'.
- iii) When you have assigned all your pins then click on **Save Pin Assignments** and select either of the choices, at this stage it doesn't matter which.

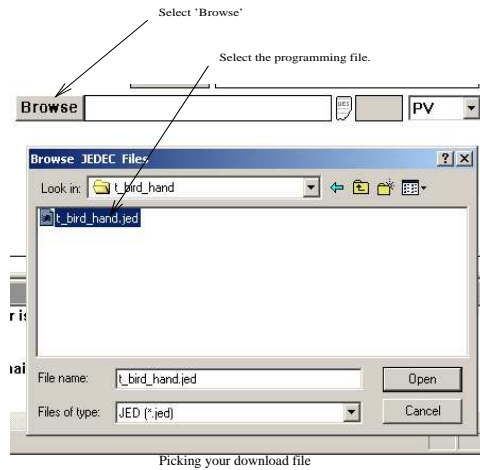
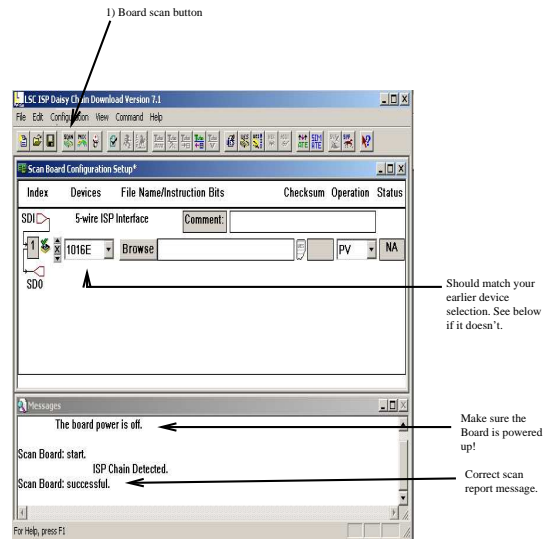


- iv) From the menu bar, select **Tools** → **Compile**. You can amuse yourself by watching the little aeroplane taking off. Seriously though, The compiler will generate messages in the bottom window and change the colour of the flow diagram on the left as it completes each stage. If there are any errors then the stage at which the error occurred will be coloured red and the details will be in the report below. If there were errors then they must be resolved and your design recompiled. If compilation was successful then you can proceed to the download.

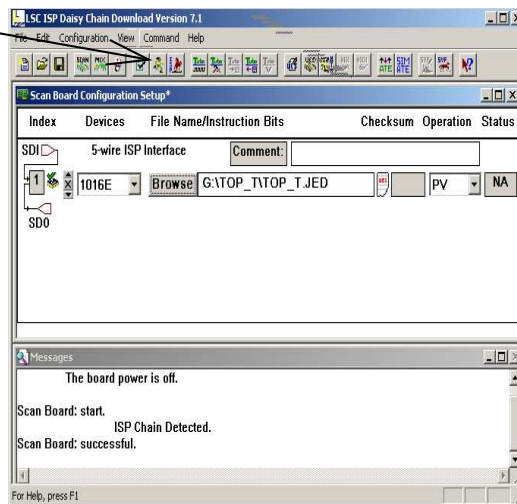
3. Downloading your Design

From the menu bar select **Tools** → **ispDCD**. This stands for Daisy-Chain Download as multiple devices can be chained together and programmed in one session. This will launch a new tool which will be superimposed on the compiler. A screen-shot is given below.

- i) Select 'scan' from the set of icons, the large LED on the board should flash briefly and the download tool should report the device correctly. If the reported device does not match the one that you selected earlier, then you will have to go back to the compiler, assign the correct device, reassign the pins and recompile. The download tool will also complain if the board is not powered up.
- ii) If the board has reported the device correctly then click on **Browse** and select what should be the only choice offered.
- iii) Select the icon of the little person running. The download should commence and the large LED on the board should flash for a few seconds. When the download is completed, you can begin testing.



Programming the device



4. Board Familiarisation

You should make sure that you can identify the various components on the target board. Identify the following.

- i) The block of 8 DIP switches. These are your inputs. By implication your design should not have more than 8 inputs.
- ii) The LED bar-graph. This is used to represent the outputs from your design. Note that only 8 of the LEDs are used. The leftmost 2 are unconnected.

Study the circuit diagram on the next page. It shows the part of the circuitry for the target board that we are concerned with. You will need this information when making pin assignments for your design.

Note that the LEDs will be ON when the output is LOW or 0 and OFF when the output is HIGH or 1.

Note that an input will be HIGH, or 1, when the switch is open circuit (OFF) and that the input will be LOW, or 0, when the switch is closed (ON).

