

Lab installation guide (IN4073 2005-2006)

Tom Janssen

March 2006

1 Installing Xilinx ISE Webpack on linux

To download the free ISE Webpack 8.1i go to http://www.xilinx.com/ise/logic_design_prod/webpack.htm

and click on 'download free'. If you already have a Xilinx account, you can use it here to download the webpack or else create an account first. (Those with a Xilinx account can access the webpack download site directly at <http://www.xilinx.com/webpack>)

After having downloaded it, run setup to install it on your computer. It is not possible to do a non-graphical install, for some reason the supported `-nox11` option still needs an active X-server.

Start the ISE project navigator with `[path-to-xilinx]/bin/linux/ise`.

Also make sure the Impact device programmer works, by creating or opening a project, building it and choose 'configure device' from the process tree in the project navigator. Alternatively you can directly start it with `[path-to-xilinx]/bin/linux/impact`.

When the standard Xilinx iMPACT device programmer cannot detect the device chain using the parallel JTAG cable, the following sections will help fix this. Two modules are required to make this work. One is `parport_pc`, the other one is called `windrvr6`.

1.1 Enabling parport_pc

The `parport_pc` module should be included in every kernel from 2.2 and above. In the 2.4 versions and later it can be enabled by loading the module `parport`.

`insmod -k parport_pc` should add modules `parport`, `parport_pc` and either `ppdev` or `vmpuser`. Check this with `lsmod`. The latter module is used by the older kernels. If the device entry `/dev/parport0` does not exist after this, it can be added manually with `mknod /dev/parport0 c 99 0 0`. To use port 1 or 2, instead of standard 0, change the last value to 16 or 32. Also change permissions to be able to use it as normal user (`chmod 666 /dev/parport0`).

1.2 Enabling windriver

Windriver should be downloaded from: <http://www.jungo.com/download/WD702LN.tgz>

Untar it, go to the dir `WinDriver/redist/`, run `./configure`, `make` and `make install` to correctly install it. Now run `modprobe windrvr6` and this module is also loaded. If `/etc/windrvr6` does not exist after this, add this manually with `mknod /dev/windrvr6 c 253 0` and change file permissions.

1.3 Loading modules at startup

To save you the trouble of manually loading these modules each time you start your computer, it is recommended to load these modules automatically at startup. To do this, load them by changing

your shell startup script (inserting the above `modprobe` commands), or by updating your module configuration file (`/etc/modules` or `/etc/modprobe.preload`, depending on your kernel)

1.4 Updating WebPack with SP2

If things are working, you can update your ISE by going to http://www.xilinx.com/xlnx/xil_ans_display.jsp?getPagePath=22629

All fixes are stated there. These include timing issues for the spartan and a lot of project navigator errors.

2 Using non-graphical Xilinx tools

At the lab we prefer using the command line tools with a makefile. The makefile included in the example projects on the lab web page can be modified to compile your own projects. To make this work, first some environment variables need to be set:

```
export XILINX=[path-to-xilinx]
export LD_LIBRARY_PATH=[path-to-xilinx]/bin/lin
export PATH=$PATH:[path-to-xilinx]/bin/lin
export XIL_NGDBUILD_OLDFLOW=1
```

To program the device, a 3rd party tool is used, in stead of iMPACT. It can be downloaded from <http://www.rogerstech.force9.co.uk/xc3sprog/>

Before making it, first change the file `xc3sprog.cpp`. Replace the line `#define DEVICEDB "devlist.txt"` with a reference to the absolute path of the file. Also add the path of the executable to the `PATH` environment variable so it can run from everywhere. When `xc3sprog` gives a parallel port error, see the above section on how to enable the `parport` module.

Note that the example bit file included with `xc3sprog` is either invalid or it just doesn't work on the spartan3 xc3s400.

3 RS232 through USB using Sweex usb2rs232

To use the Sweex `usb2rs232`, the kernel module `p12303` is needed. This is a USB Serial Driver, which makes sure that the proprietary reader protocol is translated to a standard protocol. It can be found under **Device Drivers** → **USB-Support** → **USB Serial Converter Support** → **USB Prolitic 2303** in the 2.6 kernel configuration, make sure it is enabled there. The USB subsystem should automatically insert this module when the cable is plugged into a USB port. The port can then be accessed under `/dev/ttyUSB0` (if no other USB devices are attached, otherwise another number on the end).

If it is enabled in your kernel configuration, but is not listed in `lsmod`, use `modprobe usbserial` to insert it. If `/dev/ttyUSB0` does not exist, use `mknod /dev/ttyUSB0 c 188 0` to create the device node.

It could be the case that the **generic usbserial** driver conflicts with the `p12303` module. Make sure that only `usbserial` and `p12303` are loaded to fix this.

Acknowledgements

Thanks to Michel Wilson for providing the installation procedure for the Sweex `usb2rs232`.