

Creating and Running Circuits with the TME Ports Package

Dave Galloway

*Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto*

January 2014

Communicating with your Circuit

The TME ports package can be used to communicate with your circuit. See *The TME Ports Package* document for details. If you want to use the ports package, create a port description file, giving it the same name as the top level netlist, but with a `.ports` extension. You must also instantiate a component in your netlist called `pcie_portmux`, passing it the signals that you want to connect to the ports package.

For a DE-4 circuit, run the `tme` command like this:

```
tme toplevelname
```

This will run the Altera CAD software, creating a new directory called `pcie_portmux` containing the Verilog source for the `pcie_portmux` wrapper component. It will also copy your ports file to a file called `fpgaN.ports`.

For a DE-5 circuit, run the command like this:

```
tme -de5 toplevelname
```

Compiling your Circuit

Add these lines to your `.qsf` file, so that Altera's Quartus software can find the source for the `pcie_portmux` component:

```
set_global_assignment -name SEARCH_PATH pcie_portmux  
set_global_assignment -name QIP_FILE pcie_portmux/qsys/qsys/synthesis/qsys.qip
```

Compile your circuit using Quartus.

Using the demonstration board

In order to use the demonstration board, the `tmemon` daemon must be running on the machine named `skynet.eecg`. This daemon is normally started when the machine is booted, and will continue running. If the daemon stops for some reason, you can re-start it by signing on to that machine and typing:

tmemon &

The `tm` command can then be used to communicate with the `tmemon` daemon. See *The tmemon Program* document for details.

You can check the status of the demonstration board remotely by running the `tmstatus` program. It will use X-windows to create a popup window on your screen that shows the current status of the board.

The `tmget` command will attempt to reserve the board for a specified number of minutes (default 10). If it succeeds, your name will be shown on everyone's `tmstatus` display. When you are finished, run the `tmrelease` command so that others may use the board.

Once you have ensured that no one else is using the board, you can program your circuit into the board. To do this, follow these steps (note that steps 2 and 3 must be executed as root on the machine that is connected to the demonstration board):

- 1) download your circuit to the board using the `tmrun` command
- 2) `echo 1 > /sys/bus/pci/devices/0000:03:00.0/remove`
- 3) `echo 1 > /sys/bus/pci/rescan`

To communicate with the circuit, run your program that uses the `ports` library in the directory that contains the `fpgaN.ports` file. You do not have to be signed on to the machine that has the demonstration board, as your program will communicate with `tmemon` (and the board) over the network.