Chips-2.0 Demo for SP605 Development Card

Author: Jonathan P Dawson
Date: 2013-10-17
email: chips@jondawson.org.uk

This project implements a TCP/IP stack. The TCP/IP stack acts as a server, and can accept a single connection to a TCP port. The connection is provided as a bidirectional stream of data to the application. The following protocols are supported:

- ARP request/response (with 16 level cache)
- ICMP echo request/response (ping)
- TCP/IP socket

Synthesis Estimate

The TCP/IP server consumes around 800 LUTs and 300 Flip-Flops in a Xilinx Spartan 6 device.

Dependencies

The stack is implemented in C, and needs Chips-2.0 to compile it into a Verilog module.

Source Files

The TCP/IP stack is provided by two source files:

- source/server.h
- source/server.c

Configuration

The following parameters can be configured at compile time within source/server.h:

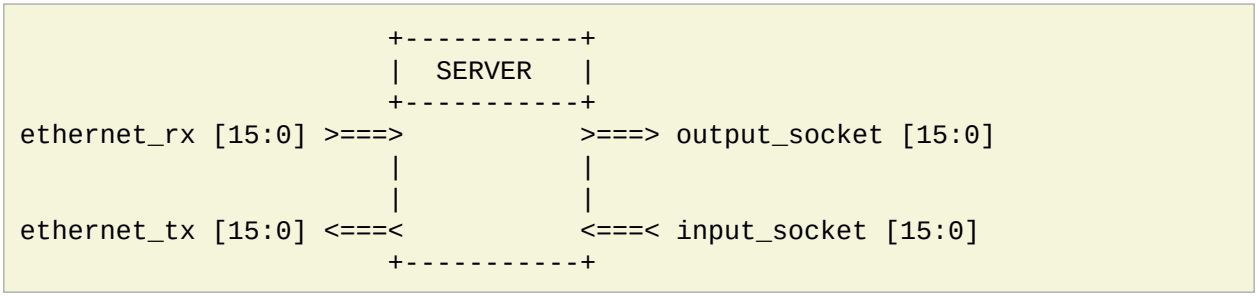
- Local Ethernet MAC address (default: 0x000102030405)
- Local IP Address (default: 192.168.1.1)
- Local TCP Port number (default: 80 HTTP)

Compile

Compile into a Verilog module (server.v) using the following command:

```
$ chip2/c2verilog source/server.v
```

Interface



Ethernet Interface

The Ethernet interface consists of two streams of data:

- An input, input_eth_rx.
- An output, output_eth_tx.

Both streams are 16 bits wide, and use the following protocol:

word	designation
0	length in bytes
n	data

Socket Interface

The socket interface consists of two streams of data:

- An input, input_socket.
- An output, output_socket.

Both streams are 16 bits wide, and use the following protocol:

word	designation
0	length in bytes
n	data

Stream Interconnect Conventions

The interfaces are based on the Chips Physical Interface Convetions which are described in the Chips-2.0 reference manual.