# 基于FPGA的Web Server使用说明

2011011262 赵一开

June 9, 2014

## 1 总述

该实验由以下几个部分组成：

- 一个使用VHDL编写的MIPS32指令集五段流水线CPU
- 一个可以在该CPU上运行的监控程序
- 一个可以在监控程序上运行的Web Server

## 2 使用说明

### 2.1 系统需求

本实验除了编程FPGA/CPLD需要使用iMPACT，必须在Windows下进行外，其余过程等均可在Mac OS X/Linux/Windows下进行。

与实验板的串口通信使用Python脚本（仅在Python2.7下测试过）进行，Mac OS X以及大多数Linux发行版自带Python，Windows请参考这里安装Python。另外，程序依赖**pySerial**库，需要通过**pip install pyserial**安装，具体参见pySerial安装说明。

在运行串口通信脚本时需要指定串口设备号，在Windows下为形如**COM1**的形式（可以在设备管理器中查看），在Linux/Mac OS X中为形如**/dev/ttyAMA0**的形式（通过ls查看）。下文中以后者为例。另外，Linux中访问串口设备可能需要root权限。

### 2.2 实现步骤

1. 使用iMPACT将**bit/pass.jed**编程至**CPLD**中，其作用为将串口信号线与FPGA相连
2. 将监控程序(kernel)写入SRAM，包括以下几个步骤：

    1. 使用iMPACT将bit/uart.bit编程至FPGA中，其作用为可以读取串口数据写入至SRAM
    2. 运行python tools/sram.py write kernel/kernel.bin /dev/ttyAMA0，将kernel代码写入至内存0x00000000地址
    3. 运行python tools/sram.py write kernel/kernel-int.bin 0x8000 /dev/ttyAMA0，将kernel中断处理代码写入至内存0x00008000地址
3. 使用iMPACT将bit/cpu.bit编程至FPGA中，CPU即可正常运行监控程序代码
4. 将web server代码通过监控程序写入SRAM并运行，包括以下几个步骤：
    1. 运行python tools/term.py writeihex program/main.hex /dev/ttyAMA0，写入程序
    2. 运行python tools/term.py writebin html/out.bin 0x80300000 /dev/ttyAMA0，将数据写入至0x80300000地址（该步骤会花若干分钟时间）
    3. 运行python tools/term.py execute 0x80400000运行代码
5. 测试：
    1. 将计算机和实验板连接至同一个交换机（路由器），网段为192.168.1.0/24，实验板默认IP地址为192.168.1.233。（若局域网不是192.168.1.0/24网段，则可以在计算机上加一条路由表）
    2. 运行ping 192.168.1.233应该可以看到回应
    3. 在浏览器（请使用chrome/safari/firefox）中打开192.168.1.233可以看到网页

# 3 源代码和调试说明

## 3.1 源代码说明

- ./*.vhdl：为CPU的硬件描述代码，顶层模块为TOP.vhdl
- ./CPLD_pass：CPLD硬件描述代码，仅仅将串口与FPGA相连
- ./FPGA_uart：FPGA的串口写入的硬件描述代码，用于在监控程序运行之前将数据通过串口写入SRAM
- ./html：网页前端代码
- ./kernel：监控程序代码
- ./program：Web Server程序代码
- ./raw_prog：测试用
- ./tools：串口通信等python脚本
- ./win-term：Windows版本的term，不确定是否能使用

## 3.2 CPU代码的修改和调试

一些CPU硬件模块包含了相应的Test bench以及makefile，可以通过ghdl仿真，比如在Mac OS X下：

- 安装ghdl和gtkwave
- make -f Registers.makefile （仿真RegistersTestbench，或者其他几个makefile)
- open Registers.vcd （波形文件）

## 3.3 监控程序、Web Server程序的修改和调试

首先需要根据相应的平台下载mips-gcc放在当前目录下，在./kernel和./program和./raw_prog中均可以直接运行make进行编译。具体参见相应的makefile。