Roach Server Machine Setup

目录 [隐藏]

Introduction

What's Roach

ROACH (Reconfigurable Open Architecture Computing Hardware) is a standalone FPGA processing board. [1] The FPGA is mainly based on Virtex series produced by Xilinx Corporation.
Supplement - BORPH
Supplement - Official Logo

Roach in Tianlai Project

Its Job

Now, we have several ROACH-2 revison2 boards, and their FPGAs' version is Virtex-6. They will be used to do AD conversion, FFT, quantization and packetization.

Hardware Configuration

[2]

- Virtex-6 SX475T FPGA (XC6VSX475T-1FFG1759C)
- PowerPC 440EPx stand-alone processor to provide control functions
- 2 x Multi-gigabit transceiver break out card slots, supporting up to 8x10Ge links which may be CX4 or SFP+
- 4 x 36 * 2M QDR II+ SRAMs connected to the FPGA
- A single 72-bit DDR3 RDIMM slot connected to the FPGA
- 2 x ZDOKs
- An FTDI FT4232H USB to JTAG, serial and IIC

What's Roach Server

Roach has several methods to boot the Linux system. SOLO boot enables Roach to boot a Linux system stored in flash on Roach board. USB boot or MMC boot enables Roach to boot from a USB/MMC inserted on Roach board. NET boot enables Roach to boot off the network using BOOTP, and that is what a Roach server does. With such a Roach server, it will be very convenient to control the Roach boards, especially when one has more than Roach boards.

Setup a Roach Server

Introduction

The Roach server will be running a Network File Sharing(NFS) service, with which the Roach can boot off the network using BOOTP. Meanwhile, the Roach server will also provide a DHCP service, so when a Roach boards is connected to the LAN, an IP address will be assigned automatically.

In addition, we also let the Roach server be a development environment for Xilinx models. That is what step 3 tells. However, one may also use another computer to develop models, and then transfer the compiled models to the Roach server in order to download to the FPGA. In that way, the following step 3 need not to do on the Roach server, but on another computer.

Hardware Requirement

The Roach server should have at least two Ethernet cards, one for connection to Roach and one for connection to Internet. Obviously, several straight Ethernet cables with a switch are needed. An Ethernet cross-over cable is also mentioned in the Roach official site which connects the server with the Roach directly. You can use that if you only have one Roach board. By the way, the cable needs not to be a cross-over one. As I tested, a straight cable also works. One also needs a USB cable (USB Type A plug to Type B plug, commonly used in printer) for the first to look into what’s the Roach board doing when booting, and modify the boot method of the Roach board.

If the development environment is also installed, the Roach server is expected to have more than 8GB memory because the Xilinx ISE consumes a lot of memory in compilation.

Software Requirement

Operating System

Though Ubuntu 12.04 64bit is taken as an example in the Roach official site, I didn’t make it through when dealing with dnsmasq. The Ubuntu OS has a Network Manager application which also uses dnsmasq. So the problem is the conflicts between the dnsmasq of OS and of user installed. Therefore, I use Centos 6.5 64bit, and I will list the procedures to configure on it and the various problems and their solutions.

Essential Softwares

- dnsmasq

Dnsmasq is a lightweight DNS/DHCP/TFTP server. We’ll use it to assign IP addresses for Roach boards.

- nfs

The BORPH server embedded Linux file system stored in the Roach server will be shared with Roach boards via network.

Development tools

- MSSGE

The MSSGE tool flow (short for Matlab/Simulink/System Generator/EDK) is the platform for FPGA-based CASPER development, which stitches together several design and implementation environments. [3]

- Matlab 2012a/b

Matlab provides a scriptable back-end for Simulink. All mask scripts are written in the Matlab language. [4] Simulink is one part of Matlab. It serves as both a schematic capture tool and a design simulation environment for system models targeted for CASPER FPGA boards. [5]
Here I use Matlab 2012a. I think 2012b also works.

- Xilinx 14.2

Xilinx contains a tool called System Generator. The system generator translates Simulink schematics into Hardware Description Language (HDL) code (either VHDL or Verilog) during design compilation. It also enables design simulation from within the Simulink environment.
Xilinx also provides an Embedded Development Kit(EDK) which compiles the generated HDL code into a bit stream that runs on the targeted FPGA.
Here I use Xilinx 14.2.

- MSSGE libraries

This is the Simulink libraries developed by Casper for Casper hardware.
Caution: For Xilinx 14.2, the mlib_devel repository from https://github.com/casper-astro/mlib_devel/ is compatible. Do not use the one from https://github.com/ska-sa/mlib_devel which is only compatible with Xilinx 14.7.

Install Centos 6.5 OS

Install Centos

Install a Centos 6.5 64bit Operating System.
From now on, almost all of the following commands need root privilege. I wouldn't mention any longer. As long as an error contains something like "Permission denied", please change to root user to try again. In Centos, change to root by running:

```
su
```

And then input the password for root.

Update source

For a fresh Cenos OS, if one runs yum install, the OS cannot find any packages. An rpm package called epel-release-6-8.noarch.rpm is suggested to install first. It's easily to find and download. For future convenience, I will post one here.
(1) Visit https://fedoraproject.org/wiki/EPEL and find "how can I use these extra packages?"
(2) Find an EL version matching your system (for CentOS 6.5, you can choose EL6).
(3) Get the link of epel release noarch rpm.
(4) In terminal run:

```
rpm –Uvh the-rpm-link-you-get-above
```

Now yum install should work, and you can check it by installing something.

For the First Time

See What's Roach Doing

If this is the first time to use Roach, one has to know what the Roach is doing when it is powered on. Besides, one has to know the MAC address of the Roach Ethernet card according to which the Roach server assigns IP address.
Though the Roach official site says that the default boot option of the Roach is netboot, our Roach boards actually boot with soloboot. So I have to change the boot method of the Roach from soloboot to netboot.

To achieve the above demands, we have to use a USB cable to connect the Roach with the server. And in the server, we'll use minicom to operate.

Install Minicom

In Centos, run the following command in root user:

```
yum install minicom
```

Operate with minicom

Now, use the USB type-A to type-B cable to connect the server with Roach board. Your server may have many USB slots. Once choose one, remember its position and always use it without change. In Roach board, insert the type-B plug to the FTDI-USB slot.
Open a terminal on server, type:

```
minicom –s
```

The option -s means to configure the settings of the minicom, since this is the first time to use the minicom.
In the list, choose "Serial port setup".
Here, you can see several options. To modify certain option, input the letter in the beginning. To take effect your choice, press Enter.
Firstly, set Bps/Par/Bits option to 115200 8N1 and Hardware Flow Control option to No.
Then, let's tell minicom which Serial device we are using. To find out how many Serial Devices in the server, open another terminal and run:

```
ls /dev/ttyUSB*
```

Since here we are using USB cable instead of traditional serial cable, we only list USB devices. You should have seen several ttyUSB devices, all ending with a number. Of course, we do not know which device we have inserted into the type-A plug. So one has to try them one by one. However, almost certainly it would be ttyUSB2. Try this one first. Change to minicom terminal, and in Serial device option, input /dev/ttyUSB2 for example. Press Enter until you return to the configuration list. Choose Save setup as dfl and then Exit.
Now, you've returned to the terminal. Run:

```
minicom
```

You should have seen the welcome message by minicom. Then, power on the Roach. If immediately you see many messages appears in the minicom screen, it means that the ttyUSB device you choose is right. If nothing appears, you have to try another ttyUSB until you can see many messages.
Once the right ttyUSB device is found, you don't need to configure minicom any more as long as the USB type-A plug is inserted in the same slot.

Change to netboot

When powering on the Roach, minicom shows many messages on screen. These messages are output automatically by the PowerPC on Roach, including basic hardware information. If you do not press any button in minicom, the Roach will boot from the default option. For Roach-2 boards, the default one is soloboot. Then Roach will load Busybox from flash, and you can log in as “root” without password.

Since we want to use netboot method, you should press any button to interrupt the boot process.

Here, you should take down the MAC address in the display.

To change the boot method to netboot, run the following command in minicom:

```
setenv bootcmd run netboot
```

This change is volatile. Roach will only boot this one time, next time it will again boot from default soloboot. To permanently use this option, run:

```
saveenv
```

Now, the Roach will always boot from network. We’ll set up the NFS later.

Roach can also boot from mmc or USB, but the related development work is halted, because ROACH2 has sufficient onboard flash to keep a number of (optionally compressed) boffiles there without having to use external storage, as written in Roach2 official site. [6]

All commands in minicom

- boot from network

  ```
  run netboot
  ```
- boot from USB

  ```
  run usbboot
  ```
- boot from MMC

  ```
  run mmcboot
  ```
- boot from on board flash

  ```
  run soloboot
  ```
- Change default boot option

  ```
  setenv bootcmd run XXXboot
  ```
- Set MAC address

  ```
  setenv ethaddr xx:xx:xx:xx:xx:xx
  ```
- Save changes

  ```
  saveenv
  ```

Any changes are volatile and requires a saveenv command to store them on flash.

- Reset UBOOT environment and boot settings

  ```
  run clearenv
  reset
  ```

Stop autoboot and then,

```
run init_eeprom
```

- Set MAC address

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

- Save the environment

```
saveenv
```

Setup Process in Detail

File system preparation

Firstly, we’ll prepare several directories to contain the file system.:

```
mkdir /srv
mkdir /srv/roach_boot
mkdir /srv/roach_boot/boot
```

The kernel

For Roach-2 revision2, download the kernel from https://github.com/ska-sa/roach2_nfs_uboot/tree/master/boot and choose the newest one “uImage-r2borph3”.
Then move it to /srv/roach_boot/boot directory with name changed to “uImage”. If you want to copy it, you should add “--preserve=all” option to preserve the security context. For example:

```
cp --preserve=all uImage-r2borph3 /srv/roach_boot/boot/uImage
```

Always remember add “--preserve=all” option when copying.
Then give boot directory and uImage full rights with:

```
chmod -R 777 /srv/roach_boot/boot
```

The file system

For Roach-2 revision2, download the file system from https://github.com/ska-sa/roach2_nfs_uboot and choose “roach2-debian-fs-snampshot-224-10-2012.tar.gz. Extract it inside the /srv/roach_boot directory by:

```
tar -zxf the-downloaded-filesystem.tar.gz -C /srv/roach_boot
```

Rename the extracted folder with “etch” with “mv” command:

```
mv extracted-folder /srv/roach_boot/etch
```

Within the etch folder, there is a “boffiles” folder. For future convenience, make a link in “/” directory:

```
ln -s /srv/roach_boot/etch/boffiles /boffiles
```

IP Address Assignment

We will build a small LAN with IP addresses 192.168.100.X. The IPs will be assigned as below:

Roach board 1: 192.168.100.1
Roach board 2: 192.168.100.2

......	
Roach board 99:	192.168.100.99
Roach server:	192.168.100.100
Other devices:	192.168.100.125-254
Roach switch:	192.168.100.125

Install DHCP service

1. Install dnsmasq

```
yum install dnsmasq
```

2. Configuration

```
vim /etc/dnsmasq.conf
```

and paste the following contents into the file:

```
# Configuration file for dnsmasq
# Edited for ROACH boot server

# We don't want dnsmasq to read /etc/resolv.conf or anything else
no-resolv

# Assign the ROACH an IP address manually, based on its MAC
dhcp-host=02:44:01:02:02:08,192.168.100.1
dhcp-host=02:44:01:02:02:13,192.168.100.2

# Have a DHCP address range for other things
dhcp-range=192.168.100.128,192.168.100.254,12h

# Set the location of the ROACH's root filesystem on the NFS server.
dhcp-option=17,192.168.100.100:/srv/roach_boot/etch

# Set the boot filename for BOOTP, which is what the ROACH boots over
dhcp-boot=uImage

# Enable dnsmasq's built-in TFTP server. Required for BOOTP.
enable-tftp

# Set the root directory for files availble via FTP.
tftp-root=/srv/roach_boot/boot

# Set the DHCP server to authoritative mode (then keep away from other networks!)
dhcp-authoritative

#Specify which ethernet interface you use to connect to the ROACH (eth0, eth1, eth2 ...)
```

```
interface=eth0

#May be useful if you have several ethernet interfaces
bind-interfaces
```

Some changes are needed according to your devices, mainly on MAC address in“dhcp-host”option and Ethernet card in "interface" option. Then save and exit, and restart dnsmasq by:

```
service dnsmasq restart
```

If an error occurs something like:

```
clnt_create: RPC: Unknown host
```

You should edit /etc/hosts file and add a line:

```
127.0.0.1       server's-name
```

Here,“server's-name”is the Roach server's host name. It's usually shown in the title of a terminal.

If an error occurs something like:

```
dnsmasq: tftp /srv/roach_boot/boot permission denied
```

Firstly make sure that you have given“boot”directory full right. Secondly, the error is mostly caused by SELinux. Solving it by editing the /etc/sysconfig/selinux file, and change:

```
SELIUX=permissive
```

Now a system reboot is needed to take effect. If you don't want to reboot, do:

```
setenforce 0
```

3. Configure Ethernet card

In Centos OS, the Ethernet card configuration files are in the following directory:

```
/etc/sysconfig/network-scripts/
```

Edit the one you use to connect to Roach, and edit or add the following lines:

```
IPADDR=192.168.100.100
NETMASK=255.255.255.0
BOOTPROTO=static
ONBOOT=yes
```

Then, re-active the Ethernet card by:

```
Ifconfig eth0 down
Ifconfig eth0 up
```

Then run“ifconfig”to see if the settings have taken effect.

4. Edit hosts

For convenience, add some items in the /etc/hosts file. Such as:

```
192.168.100.1       roach1
192.168.100.2       roach2
```

5. Others

Turn off the firewalls:

```
service iptables stop
```

Let firewalls not start when booting OS:

```
chkconfig iptables off
```

Let dnsmasq service start automatically when booting OS:

```
chkconfig dnsmasq on
```

Install NFS service

1. Install NFS

Centos 6.5 already has NFS service. If not, install it by:

```
yum install nfs-utils nfs-utils-lib
```

Start the nfs service by:

```
service nfs start
```

Now, if you run:

```
rpcinfo -p
```

you'll see nfs running, also some other services like portmapper and mounted.

2. Share directory

Now let's tell nfs which directory to share by:

```
vim /etc/exports
```

And add the following lines:

```
# Share 'roach_boot' directory
/srv/roach_boot
192.168.100.0/24(rw,subtree_check,no_root_squash,insecure)
```

Here 192.168.100.0 is the subnet of out small LAN, which contains all of the 192.168.100.X IP addresses. Save and exit, then export the file system by:

```
exportfs -a
```

We can check what have been shared by nfs with the command:

```
showmount -e
```

The output would be:

```
Export list for roachserver:
/srv/roach_boot 192.168.100.0/24
```

3. Others:

Let NFS start automatically when booting OS:

```
chkconfig nfs on
```

Test Roach Server

Monitor with minicom

Connect the Roach server and Roach with USB cable, and start up the minicom in terminal. We will see what's Roach doing via minicom.

Get Server Prepared

Check that the file system has been put in the correct directory with correct rights.

Check that dnsmasq.conf has been correctly modified.

Check that dnsmasq and nfs services are running.

Check that the firewall is turned off.

Connect the Roach server and Roach with an Ethernet cable.
Check that you are using the correct Ethernet card on Roach server and Roach (PPC NET).

Start

Now it's time to power on the Roach. You can see what's Roach doing via minicom.
If finally the minicom shows a prompt something like:

```
Login as:
```

It means that the Roach has successfully booted up from the network.
If a warning occurs saying that:

??????

Just ignore it.
From now on, you do not need the USB cable any more.

Install Development Tools

Install Matlab and Xilinx

Install Matlab 2012a/b.
Install Xilinx 14.2.
Here, we install them in /opt/ directory.

Tweaks for compilation

[7]
Xilinx removed support for several hardware pcores we use for ROACH1/2 from the 14 versions. So the current solution is to add the following pcores from the Xilinx 11 install to the XPS_ROACH_BASE/pcores folder or to the Xilinx14 install directory at Xilinx/14.2/ISE_DS/EDK/hw/XilinxProcessorIPLib/pcore.
Download the 11 opb pcores from HERE.
Change to root user, and unzip the downloaded ZIP file:

```
unzip the_file_you_downloaded.zip
```

You will get the following files:

- bram_if_cntlr_v1_00_a
- bram_if_cntlr_v1_00_b
- ipif_common_v1_00_c
- opb_arbiter_v1_02_e
- opb_bram_if_cntlr_v1_00_a
- opb_ipif_v3_00_a
- opb_opb_lite_v1_00_a
- opb_v20_v1_10_c
- proc_common_v1_00_a

Move these files to the XPS_ROACH_BASE/pcores folder. Probably it's in '/opt/mlib_devel/xps_base/XPS_ROACH_base/pcores'.

Install mlib_devel

Download the package from https://github.com/casper-astro/mlib_devel/.
Extract it and move it to /opt/ directory, too.
Edit the “startsg” file to tell where Matlab and Xilinx are installed.
For “startup.m” file, you can add the following line at last:

```
simulink
```

So that Simulink will be started up automatically when Matlab is started.
All of the users in Roach server only need the two files to start up the development tools. They can copy from here.

Xilinx License Problem

Normal user in Roach server will encounter an error reporting Xilinx license invalid when compiling Simulink models. Solve the error by adding the following line to the end of the file /etc/bashrc:

```
export XILINXD_LICENSE_FILE="/opt/xilinx_lic/xilinx_ise.lic"
```

Test

In normal user mode, run the file “startsg” to see if the Simulink can start up.
What’s more, you can try some tutorials on Roach official site:
https://casper.berkeley.edu/wiki/Tutorials

Interact with Roach

Introduction

The file system running on Roach has a telnet service, through which we can interact with Roach. One can directly use telnet to communicate with Roach, or as suggested by Roach official site, use corr modules developed for python to communicate with Roach.
The python module corr uses Karoo Array Telescope Control Protocol (KATCP). It is a general purpose protocol for board interaction. Then, a standalone KATCP server is implemented which runs under BORPH. When this server is running on a board, a user can interact with the the board's FPGA resources by querying it over the network with the corr python module.

Install Interaction Tools

Install Telnet

```
yum install telnet
```

Install iPython

```
yum install ipython
```

Install numpy h5py

```
yum install numpy h5py
```

Install corr and related packages

Download the following packages from their website:

- Construct: https://pypi.python.org/pypi/construct
- Spead: https://pypi.python.org/pypi/spead
- Katcp 0.3.5: https://pypi.python.org/pypi/katcp/0.3.5 (Do not use version newer than 3)
- Corr: https://pypi.python.org/pypi/corr

For each package downloaded, use:

```
tar –zxf xxxxxxx.tar.gz
```

to extract. Then enter the extracted folder, run:

```
python setup.py install
```

to install them one by one.

Test

Open iPython, and try to import the module corr. If only two Runtime Warnings called “numpy.dtype size changed” occur, it means that the packages have been correctly installed. Just ignore the two warnings.

Interact with Roach

Interact via Telnet

The telnet service running on Roach is listening the port 7147. In terminal, start up telnet with:

```
telnet roach1 7147
```

Some messages should be returned. If so, the connection is OK.

Now we interact with Roach via telnet. Remember that your command should always begin with a question mark “?”. The messages returned from the Roach all begin with a pound sign “#” except the last line which begins with an exclamatory mark “!”.

Here are several commands you can use:

- List the available bof files: ?listbof
- Program the FPGA: ?progdev

- List the devices: ?listdev()
- Write value to device: ?wordwrite device_name bit_offset value
- Read value from device: ?wordread device_name

To exit from telnet, press “Ctrl +]”, then press “Enter” to see the prompt at “telnet>”, then input “close” and confirm.

Interact via Python

Start up the ipython.

Firstly, import the module:

```
from corr import katcp_wrapper as kw
```

Secondly, build a Roach object:

```
my_roach = kw.FpgaClient(‘roach1’, 7147)
```

Then, you can input “my_roach.” and press “Tab” to get the various methods available. Such as:

- Check if roach is connected: my_roach.is_connected()
- Start the connection: my_roach.start()
- List the available boff files: my_roach.listbof()
- Program the FPGA: my_roach.progdev()
- Write value to device: my_roach.write_int(‘device_name’, value)
- Read value from device: my_roach.read_int(‘device_name’)
- Stop the connection: my_roach.stop()

An Example

Build a Simulink Model

Build a Simulink model following the intructions in the tutorial 1 website:
https://casper.berkeley.edu/wiki/Introduction_to_Simulink

Compilation with Casper_xps

In Matlab prompt, run “casper_xps” to start up the compilation.

Program the FPGA

Copy you compiled *.bof file to “/boffiles” directory.
Use telnet or ipython to program the FPGA.

Check Results

For tutorial 1, write an arbitrary value to software register “a” and “b”, and then read out the value from “sum_a_b” to see if it’s right.