

RAMP Blue Setup Guide

Alex Krasnov & Greg Gibeling
UC Berkeley
{akrasnov, gdgib}@eecs.berkeley.edu

February 24, 2008

1 Introduction

This guide covers setting up and using the full RAMP Blue build and runtime environment. You may complete all of the steps or skip most of them by downloading the pre-built VM image.

Minimum system requirements:

- 1 x86 CPU
- 1-2 GB RAM
- 10-20 GB HDD
- 2 NICs
- 1 RS232 port (serial cable)
- 1 LPT port (Parallel Cable IV)
- 1 USB port (CF card reader)
- Debian GNU/Linux (host instructions may differ otherwise)

The following parameter conventions are used throughout the system and this guide:

- n number of boards per system, typical 1
- m number of chips per board, typical 4
- o number of nodes per chip, typical 8
- p number of XAUIs per chip, typical 4
- i board index, $0 \leq i < n$
- j chip index, $0 \leq j < m$
- k nodes index, $0 \leq k < o$
- l XAUI index, $0 \leq l < p$
- u 3D mesh height, typical 1
- v 3D mesh width, typical 1
- w 3D mesh length, typical 1

The following color conventions are used throughout this guide:

- Setup step
- Execution location
- General comment
- General action
- Shell command
- General setting
- Literal text
- Possible change
- Usage example

2 VMware Installation

- Install VMware server

```
- On the host
- wget "http://download3.vmware.com/software/vmserver/VMware-server-1.0.4-56528.tar.gz"
- tar xfvzp VMware-server-1.0.4-56528.tar.gz
- rm VMware-server-1.0.4-56528.tar.gz
- cd vmware-server-distrib
- sudo ./vmware-install.pl
  * Enter "/usr/bin" for "In which directory do you want to install the binary files?"
  * Enter "/etc" for "What is the directory that contains the init directories (rc0.d/ to rc6.d/)?"
  * Enter "/etc/init.d" for "What is the directory that contains the init scripts?"
  * Enter "/usr/sbin" for "In which directory do you want to install the daemon files?"
  * Enter "/usr/lib/vmware" for "In which directory do you want to install the library files?"
  * Enter "yes" for "The path "/usr/lib/vmware" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?"
  * Enter "/usr/share/man" for "In which directory do you want to install the manual files?"
  * Enter "/usr/share/doc/vmware" for "In which directory do you want to install the documentation files?"
  * Enter "yes" for "The path "/usr/share/doc/vmware" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?"
  * Enter "yes" for "Before running VMware Server for the first time, you need to configure it by invoking the following command: "/usr/bin/vmware-config.pl". Do you want this program to invoke the command for you now?"
  * Press Enter to display license agreement
  * Press Enter to scroll down
  * Enter "yes" for "Do you accept?"
  * Enter "/usr/share/icons" for "In which directory do you want to install the mime type icons?"
  * Enter "/usr/share/applications" for "What directory contains your desktop menu entry files? These files have a .desktop file extension."
  * Enter "/usr/share/pixmaps" for "In which directory do you want to install the application's icon?"
  * Enter "yes" for "None of the pre-built vmmon modules for VMware Server is suitable for your running kernel. Do you want this program to try to build the vmmon module for your system (you need to have a C compiler installed on your system)?"
  * Enter "/lib/modules/2.6.18/build/include" for "What is the location of the directory of C header files that match your running kernel?"
  * Enter "yes" for "Do you want networking for your virtual machines?"
  * Enter "eth0" for "Your computer has multiple ethernet network interfaces available: dummy0, eth0, eth1, eth2, eth3. Which one do you want to bridge to vmnet0?"
  * Enter "yes" for "Do you wish to configure another bridged network?"
  * Enter "eth1" for "Your computer has multiple ethernet network interfaces available: dummy0, eth1, eth2, eth3. Which one do you want to bridge to vmnet2?"
  * Enter "yes" for "Do you wish to configure another bridged network?"
  * Enter "eth2" for "Your computer has multiple ethernet network interfaces available: dummy0, eth2, eth3. Which one do you want to bridge to vmnet3?"
  * Enter "yes" for "Do you wish to configure another bridged network?"
  * Enter "eth3" for "Your computer has multiple ethernet network interfaces available: dummy0, eth3. Which one do you want to bridge to vmnet4?"
  * Enter "no" for "Do you wish to configure another bridged network?"
  * Enter "yes" for "Do you want to be able to use NAT networking in your virtual machines?"
  * Enter "no" for "Do you want this program to probe for an unused private subnet?"
  * Enter "10.5.0.1" for "What will be the IP address of your host on the private network?"
  * Enter "255.255.0.0" for "What will be the netmask of your private network?"
  * Enter "no" for "Do you wish to configure another NAT network?"
  * Enter "yes" for "Do you want to be able to use host-only networking in your virtual machines?"
  * Enter "no" for "Do you want this program to probe for an unused private subnet?"
  * Enter "10.4.0.1" for "What will be the IP address of your host on the private network?"
  * Enter "255.255.0.0" for "What will be the netmask of your private network?"
  * Enter "no" for "Do you wish to configure another host-only network?"
  * Enter "902" for "Please specify a port for remote console connections to use"
  * Enter "/var/lib/vmware/Virtual Machines" for "In which directory do you want to keep your virtual machine files?"
  * Enter "yes" for "The path "/var/lib/vmware/Virtual Machines" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?"
  * Enter "XXXXX-XXXXX-XXXXX-XXXXX" for "Please enter your 20-character serial number."
```

Skip to Section 7 or 11 if you have downloaded the pre-built VM image and do or do not desire to install commercial tools, respectively.

3 Initial VM Image Setup

- Set up VM image

```
- On the host
- sudo modprobe -r lp
- vmware &
  * Select "Local host" for "Select the VMware host that you want to connect to."
  * Press "Connect"
  * Press "File" → "New" → "Virtual Machine..."
  * Press "Next"
  * Select "Custom" for "Virtual Machine Configuration"
  * Press "Next"
  * Select "Linux" for "Guest Operating System"
  * Select "Other Linux 2.6.x kernel" for "Version"
  * Press "Next"
  * Enter "Debian" for "Name"
```

```

* Enter "/home/proj/blue/Virtual Machines/Debian" for "Location"
* Press "Next"
* Select "One" for "Number of processors"
* Press "Next"
* Deselect "Make this virtual machine private" for "Access Rights"
* Press "Next"
* Enter "2048" for "Memory"
* Press "Next"
* Select "Do not use a network connection" for "Network Connection"
* Press "Next"
* Select "LSI Logic" for "SCSI adapter"
* Press "Next"
* Select "Create a new virtual disk" for "Disk"
* Press "Next"
* Select "SCSI" for "Virtual Disk Type"
* Press "Next"
* Enter "64.0" for "Disk size"
* Deselect "Allocate all disk space now"
* Select "Split disk into 2GB files"
* Press "Next"
* Enter "Debian.vmdk" for "Disk File"
* Press "Finish"
* Press "VM" → "Settings"
* Select "Hardware"
* Press "Add"
* Select "Ethernet Adapter" for "Hardware Type"
* Press "Next"
* Select "Custom: Specific virtual network" for "Network Connection"
* Enter "/dev/vmnet8" for "Custom: Specific virtual network"
* Select "Connect at power on" for "Device Status"
* Press "Finish"
* Press "Add"
* Select "Ethernet Adapter" for "Hardware Type"
* Press "Next"
* Select "Custom: Specific virtual network" for "Network Connection"
* Enter "/dev/vmnet2" for "Custom: Specific virtual network"
* Select "Connect at power on" for "Device Status"
* Press "Finish"
* Press "Add"
* Select "Serial Port" for "Hardware Type"
* Press "Next"
* Select "Use physical serial port on the host" for "Serial Port"
* Press "Next"
* Enter "/dev/ttyUSB0" for "Physical Serial Port"
* Select "Connect at power on" for "Device Status"
* Press "Finish"
* Press "Add"
* Select "Parallel Port" for "Hardware Type"
* Press "Next"
* Select "Use a physical parallel port on the host" for "Parallel Port"
* Press "Next"
* Enter "/dev/parport0" for "Physical Parallel Port"
* Select "Connect at power on" for "Device Status"
* Press "Finish"
* Press "Add"
* Select "USB Controller" for "Hardware Type"
* Press "Next"
* Press "Finish"
* Press "OK"

```

4 Debian GNU/Linux Installation

- Mount Debian GNU/Linux 4.0 r2 ISO image:

- On the host

- `wget "http://cdimage.debian.org/debian-cd/4.0_r2/1386/iso-cd/debian-40r2-i386-netinst.iso"`

- In the host VMware

```

* Press "VM" → "Settings"
* Select "Hardware"
* Select "CD-ROM 1 (IDE 1:0)" for "Device"
* Select "Use ISO image" for "Connection"
* Enter "/home/proj/blue/debian-40r2-i386-netinst.iso" for "Use ISO Image"
* Press "OK"

```

- Install Debian GNU/Linux 4.0 r2:

- In the host VMware

```

* Press "VM" → "Power" → "Power On"

```

- On the guest

- Press Enter

```

* Select "English" for "Choose a language"

```

```

* Press Enter
* Select "United States" for "Choose a country, territory or area"
* Press Enter
* Select "American English" for "Keymap to use"
* Press Enter
* Select "eth0" for "Primary network interface"
* Press Enter
* Select "Continue" for "Network autoconfiguration failed"
* Press Enter
* Select "Configure network manually" for "Network configuration method"
* Press Enter
* Enter "10.5.0.2" for "IP address"
* Press Enter
* Enter "255.255.0.0" for "Netmask"
* Press Enter
* Enter "10.5.0.1" for "Gateway"
* Press Enter
* Enter "128.32.62.21 128.32.62.23" for "Name server addresses"
* Press Enter
* Enter "sting" for "Hostname"
* Press Enter
* Enter "eecs.berkeley.edu" for "Domain name"
* Press Enter
* Select "Manual" for "Partitioning method"
* Press Enter
* Select "SCSI1 (0,0,0) (sda) - 68.7 GB VMware, VMWare Virtual S"
* Press Enter
* Select "Yes" for "Create new empty partition table on this device?"
* Press Enter
* Select " pri/log 68.7 GB FREE SPACE "
* Press Enter
* Select "Create a new partition" for "How to use this free space"
* Press Enter
* Enter "4 GB" for "New partition size"
* Press Enter
* Select "Primary" for "Type for the new partition"
* Press Enter
* Select "Beginning" for "Location for the new partition"
* Press Enter
* Select "Use as" for "Partition settings"
* Press Enter
* Select "swap area" for "How to use this partition"
* Press Enter
* Select "Done setting up the partition"
* Press Enter
* Select " pri/log 64.7 GB FREE SPACE "
* Press Enter
* Select "Create a new partition" for "How to use this free space"
* Press Enter
* Enter "64.7 GB" for "New partition size"
* Press Enter
* Select "Primary" for "Type for the new partition"
* Press Enter
* Select "Bootable flag" for "Partition settings"
* Press Enter
* Select "Done setting up the partition"
* Press Enter
* Select "Finish formatting and write changes to disk"
* Press Enter
* Select "Yes" for "Write the changes to disks?"
* Press Enter
* Select "Pacific" for "Select your time zone"
* Press Enter
* Enter "rootsting" for "Root password"
* Press Enter
* Enter "rootsting" for "Re-enter password to verify"
* Press Enter
* Enter "" for "Full name for the new user"
* Press Enter
* Enter "upc" for "Username for your account"
* Press Enter
* Enter "upcsting" for "Choose a password for the new user"
* Press Enter
* Enter "upcsting" for "Re-enter password to verify"
* Press Enter
* Select "Yes" for "Use a network mirror?"
* Press Enter
* Select "United States" for "Debian archive mirror country"
* Press Enter
* Select "mirrors.kernel.org" for "Debian archive mirror"
* Press Enter
* Enter "" for "HTTP proxy information"
* Press Enter
* Select "No" for "Participate in the package usage survey?"
* Press Enter

```

- * Select "Desktop environment" for "Choose software to install"
- * Select "Standard system" for "Choose software to install"
- * Press Enter
- * Select "1280x1024" for "Video modes to be used by the X server"
- * Select "1024x768" for "Video modes to be used by the X server"
- * Select "800x600" for "Video modes to be used by the X server"
- * Select "640x480" for "Video modes to be used by the X server"
- * Press Enter
- * Select "Yes" for "Install the GRUB boot loader to the master boot record?"
- * Press Enter
- * Select "Continue" for "Installation complete"
- * Press Enter

5 Initial Debian GNU/Linux Setup

- Turn off screensaver

- In the guest desktop session
- Press "Desktop" → "Preferences" → "Screensaver"
 - * Deselect "Activate screensave when session is idle"
 - * Press "Close"

- Turn off gnome-volume-manager

- In the guest desktop session
- Press "Desktop" → "Preferences" → "Removable Drives and Media"
 - * Select "Storage"
 - * Deselect "Mount removable drives when hot-plugged" for "Removable Drives"
 - * Deselect "Mount removable media when inserted" for "Removable Drives"
 - * Deselect "Browse removable media when inserted" for "Removable Drives"
 - * Deselect "Burn a CD or DVD when a blank disc is inserted" for "Blank CD and DVD Discs"
 - * Select "Multimedia"
 - * Deselect "Play audio CD discs when inserted" for "Audio CD Discs"
 - * Deselect "Play video DVD disks when inserted" for "Video DVD Discs"
 - * Deselect "Play music files when connected" for "Portable Music Players"
 - * Select "Camera"
 - * Deselect "Import digital photographs when connected" for "Digital Camera"
 - * Deselect "Edit video when connected" for "Digital Video Camera"
 - * Select "PDAs"
 - * Deselect "Sync Palm devices when connected" for "Palm"
 - * Deselect "Sync PocketPC devices when connected" for "PocketPC"
 - * Select "Printers & Scanners"
 - * Deselect "Automatically run a program when a printer is connected" for "Printers"
 - * Deselect "Automatically run a program when a scanner is connected" for "Scanners"
 - * Select "Input Devices"
 - * Deselect "Automatically run a program when a USB mouse is connected" for "Mice"
 - * Deselect "Automatically run a program when a USB keyboard is connected" for "Keyboards"
 - * Deselect "Automatically run a program when a tablet is connected" for "Tablets"
 - * Press "Close"

- Set up shortcut icons

- In the guest desktop session
- Right-click evolution icon on top panel
 - * Press "Remove From Panel"
- Press "Application" → "Accessories" → "Terminal" and drag onto top panel

- Add upc to sudoers

- In the guest desktop session
- su
- export EDITOR=pico
- visudo
 - * Add the following line:
 - upc ALL=(ALL) ALL
- exit

- Remove network-manager

- In the guest desktop session
- sudo apt-get remove network-manager

- Set up DNS servers

- In the guest desktop session
- sudo pico /etc/resolv.conf
 - * Edit to contain only the following lines:
 - nameserver 128.32.62.21
 - nameserver 128.32.62.23
 - domain eecs.berkeley.edu
 - search eecs.berkeley.edu

- sudo ifdown eth0; sudo ifup eth0
- Update distribution
 - [In the guest desktop session](#)
 - sudo apt-get update
- Install NTP
 - [In the guest desktop session](#)
 - sudo apt-get install ntp
- Install SSH server and VNC client and server
 - [In the guest desktop session](#)
 - sudo apt-get install openssh-server
 - sudo apt-get install xvnc4viewer vnc4server
- Set up SSH and VNC
 - [On the host](#)
 - ssh -L 5901:10.5.0.2:5901 upc@10.5.0.2
 - [In the guest SSH session](#)
 - mkdir /home/upc/.vnc
 - pico /home/upc/.vnc/xstartup
 - * Edit to contain only the following lines:
 - . #!/bin/sh
 - .
 - . # Uncomment the following two lines for normal desktop:
 - . # unset SESSION_MANAGER
 - . # exec /etc/X11/xinit/xinitrc
 - .
 - . [-x /etc/vnc/xstartup] && exec /etc/vnc/xstartup
 - . [-r \$HOME/.Xresources] && xrb \$HOME/.Xresources
 - . xsetroot -solid grey
 - . gnome-session &
 - chmod +x /home/upc/.vnc/xstartup
 - vncserver :1
 - * Enter "upcsting" for "Password"
 - * Enter "upcsting" for "Verify"
 - [On the host](#)
 - vncviewer :1
- Mount physical CD-ROM drive
 - [In the host VMware](#)
 - * Press "VM" → "Settings"
 - * Select "Hardware"
 - * Select "CD-ROM 1 (IDE 1:0)" for "Device"
 - * Select "Use a physical drive" for "Connection"
 - * Select "Host" for "Location"
 - * Enter "/dev/hda" for "Device"
 - * Select "Legacy emulation"
 - * Press "OK"
- Set up CD-ROM fstab entry
 - [In the guest VNC session](#)
 - sudo pico /etc/fstab
 - * Change the following line:
 - . /dev/hdc /media/cdrom0 udf,iso9660 user,noauto 0 0
 - * To the following line:
 - . /dev/hdc /media/cdrom0 udf,iso9660 user,noauto,exec 0 0
- Make tool directories
 - [In the guest VNC session](#)
 - sudo ln -s opt /tools
 - sudo mkdir /opt/commercial
 - sudo mkdir /opt/commercial/xilinx
 - sudo mkdir /opt/commercial/flexlm
 - sudo ln -s commercial/xilinx /opt/xilinx
 - sudo ln -s commercial/flexlm /opt/flexlm
- Make build directories
 - [In the guest VNC session](#)
 - sudo mkdir /home/proj
 - sudo mkdir /home/proj/blue
 - sudo mkdir /home/proj/root
 - sudo mkdir /home/proj/tmp
 - sudo chown upc:upc /home/proj/*
 - chmod 1777 /home/proj/tmp
- Make mount directories
 - [In the guest VNC session](#)
 - sudo mkdir /mnt/1 /mnt/2 /mnt/3

6 Official Package Installation

- Update APT source list

- [In the guest VNC session](#)
- `sudo pico /etc/apt/sources.list`
 - * Edit to contain only the following lines:
 - . `deb http://mirrors.kernel.org/debian/ sarge non-free`
 - . `deb-src http://mirrors.kernel.org/debian/ sarge non-free`
 - .
 - . `deb http://mirrors.kernel.org/debian/ etch main`
 - . `deb-src http://mirrors.kernel.org/debian/ etch main`
 - .
 - . `deb http://www.backports.org/debian/ etch-backports non-free`
 - . `deb-src http://www.backports.org/debian/ etch-backports non-free`
 - .
 - . `deb http://security.debian.org/ sarge/updates non-free`
 - . `deb-src http://security.debian.org/ sarge/updates non-free`
 - .
 - . `deb http://security.debian.org/ etch/updates main`
 - . `deb-src http://security.debian.org/ etch/updates main`
- `gpg --keyserver subkeys.gpg.net --recv-keys 16BA136C`
- `gpg --export 16BA136C | sudo apt-key add -`

- Update and upgrade distribution

- [In the guest VNC session](#)
- `sudo apt-get update`
- `sudo apt-get dist-upgrade`

- Remove old kernel image if one exists

- [In the guest VNC session](#)
- `sudo apt-get remove linux-image-2.6.18-5`
 - * Select "No" for "Do you want to abort removal now?"
 - * Press Enter

- Reboot if new kernel image was installed

- [In the guest VNC session](#)
- `sudo shutdown -r now`

- Install autoconf and automake

- [In the guest VNC session](#)
- `sudo apt-get install autoconf automake1.9 automake`
- `sudo update-alternatives --set automake /usr/bin/automake-1.10`

- Install pkg-config and libtool

- [In the guest VNC session](#)
- `sudo apt-get install pkg-config`
- `sudo apt-get install libtool`

- Install make and GCC

- [In the guest VNC session](#)
- `sudo apt-get install build-essential`
- `sudo ln -s make /usr/bin/gmake`
- `sudo apt-get install gcc-3.4 g++-3.4 libstdc++5`
- `sudo rm /usr/bin/cpp /usr/bin/gcc /usr/bin/g++ /usr/bin/gccbug /usr/bin/gcov`
- `sudo ln -s cpp-3.4 /usr/bin/cpp`
- `sudo ln -s gcc-3.4 /usr/bin/gcc`
- `sudo ln -s g++-3.4 /usr/bin/g++`
- `sudo ln -s gccbug-3.4 /usr/bin/gccbug`
- `sudo ln -s gcov-3.4 /usr/bin/gcov`

- Install kernel-package and fakeroot

- [In the guest VNC session](#)
- `sudo apt-get install kernel-package`
- `sudo apt-get install fakeroot`

- Install Flex and Bison

- [In the guest VNC session](#)
- `sudo apt-get install flex bison`

- Install ncurses library

- [In the guest VNC session](#)
- `sudo apt-get install libncurses5-dev`

- Install Open Motif library
 - [In the guest VNC session](#)
 - `sudo apt-get install libmotif3`
- Install GNOME libraries
 - [In the guest VNC session](#)
 - `sudo apt-get install libglade2-dev libgnomeui-dev`
- Install pcap library
 - [In the guest VNC session](#)
 - `sudo apt-get install libpcap0.8-dev`
- Install Java 1.6
 - [In the guest VNC session](#)
 - `epiphany "http://java.sun.com/javase/downloads" &`
 - * Download "jdk-6-doc.zip" to /tmp
 - `sudo chown root:root /tmp/jdk-6-doc.zip`
 - `sudo apt-get install sun-java6-bin sun-java6-jre sun-java6-jdk sun-java6-doc sun-java6-plugin sun-java6-source sun-java6-demo sun-java6-fonts`
 - * Select "Ok"
 - * Press "Enter"
 - * Select "Yes" for "Do you agree with the DLJ license terms?"
 - * Press "Enter"
 - `sudo rm /tmp/jdk-6-doc.zip`
- Install teTeX
 - [In the guest VNC session](#)
 - `sudo apt-get install tetex-bin tetex-extra`
- Install rsync
 - [In the guest VNC session](#)
 - `sudo apt-get install rsync`
- Install CVS and Subversion
 - [In the guest VNC session](#)
 - `sudo apt-get install cvs`
 - `sudo apt-get install subversion`
- Install minicom
 - [In the guest VNC session](#)
 - `sudo apt-get install minicom`
 - `sudo minicom -s`
 - * Select "Serial port setup"
 - * Press Enter
 - * Enter "A" for "Serial Device" for "Change which setting?"
 - * Enter "/dev/ttyS0" for "Serial Device"
 - * Press Enter
 - * Enter "E" for "Bps/Par/Bits" for "Change which setting?"
 - * Enter "1" for "115200" for "Speed"
 - * Enter "Q" for "8-N-1" for "Parity", "Data", "Stopbits"
 - * Press Enter
 - * Enter "F" for "Hardware Flow Control", toggle to "No"
 - * Enter "G" for "Hardware Flow Control", toggle to "No"
 - * Press Enter
 - * Select "Save setup as df1"
 - * Press Enter
 - * Select "Exit from Minicom"
 - * Press Enter
- Install Expect
 - [In the guest VNC session](#)
 - `sudo apt-get install tcl8.4 expect`
- Remove iptables
 - [In the guest VNC session](#)
 - `sudo apt-get remove iptables`
- Install NFS server
 - [In the guest VNC session](#)
 - `sudo apt-get install nfs-kernel-server`
- Install tcpdump and Wireshark
 - [In the guest VNC session](#)
 - `sudo apt-get install tcpdump`
 - `sudo apt-get install wireshark`
- Clean distribution
 - [In the guest VNC session](#)
 - `sudo apt-get clean`

7 Commercial Tool Installation

• Install ISE 9.2i

```
- In the guest VNC session
- Insert ISE 9.2i DVD
- mount /dev/hdc
- sudo /media/cdrom0/setup
  * Press "Next"
  * Enter "xxxx-xxxx-xxxx-xxxx" for "Registration ID"
  * Press "Next"
  * Select "I accept the terms of this software license."
  * Press "Next"
  * Select "I accept the terms of this software license."
  * Press "Next"
  * Select "I accept the terms of this software license."
  * Press "Next"
  * Enter "/tools/xilinx/ISE9.2i_lin" for "Select Destination Directory"
  * Press "Next"
  * Select "Design Environment Tools" for "Select Installation Options"
  * Deselect "CPLD" for "Select Installation Options"
  * Deselect "Virtex/VirtexE/Spartan2/Spartan2E" for "Select Installation Options"
  * Select "Virtex2/Virtex2F/Spartan3/Spartan3E" for "Select Installation Options"
  * Deselect "Spartan3A/Spartan3AN/Spartan3ADSP" for "Select Installation Options"
  * Deselect "Virtex4" for "Select Installation Options"
  * Select "Standalone Programming Tools" for "Select Installation Options"
  * Press "Next"
  * Select "LD_LIBRARY_PATH" for "Select Installation Options"
  * Select "LMC_HOME" for "Select Installation Options"
  * Select "NPX_PLUGIN_PATH" for "Select Installation Options"
  * Select "PATH" for "Select Installation Options"
  * Select "XILINX" for "Select Installation Options"
  * Press "Next"
  * Deselect "Launch WebUpdate" for "Select Installation Options"
  * Deselect "Install Cable Drivers" for "Select Installation Options"
  * Press "Next"
  * Press "Install"
  * Press "OK"

- sudo rm -rf /Xilinx
- sudo ln -s ISE9.2i_lin /tools/xilinx/ISE
- umount /dev/hdc
- Remove ISE 9.2i DVD
```

• Install EDK 9.2i

```
- In the guest VNC session
- Insert EDK 9.2i DVD
- mount /dev/hdc
- sudo /media/cdrom0/setup
  * Press "Next"
  * Enter "xxxx-xxxx-xxxx-xxxx" for "Registration ID"
  * Press "Next"
  * Select "I accept the terms of this software license."
  * Press "Next"
  * Select "I accept the terms of this software license."
  * Press "Next"
  * Enter "/tools/xilinx/EDK9.2i_lin" for "Select Destination Directory"
  * Press "Next"
  * Select "Platform Studio Tool and Processor IP" for "Select Installation Options"
  * Press "Next"
  * Select "LD_LIBRARY_PATH" for "Select Installation Options"
  * Select "PATH" for "Select Installation Options"
  * Select "XILINX_EDK" for "Select Installation Options"
  * Press "Next"
  * Press "Install"
  * Press "OK"

- sudo ln -s EDK9.2i_lin /tools/xilinx/EDK
- umount /dev/hdc
- Remove EDK 9.2i DVD
```

• Install ISE service pack

```
- In the guest VNC session
- epiphany "http://www.xilinx.com/support/download/" &
  * Download "http://download.xilinx.com/protected/generics/ise/ise9/92i_sp4/9_2_04i_lin.zip"
- unzip 9_2_04i_lin.zip
- rm 9_2_04i_lin.zip
- cd 9_2_04i_lin
- sudo ./setup
  * Enter "/tools/xilinx/ISE9.2i_lin" for "Select Destination Directory"
  * Press "OK"
```

```

* Select "ISE 9.2i Service Pack 4" for "Update Name"
* Press "OK"
* Press "OK"
* Press "OK"

- sudo rm -rf /tools/xilinx/ISE9.2i_lin/.backup
- cd ..
- sudo rm -rf 9_2_04i_lin

```

• Install ISE IP update

```

- In the guest VNC session
- epiphany "http://www.xilinx.com/support/download/" &
    * Download "http://download.xilinx.com/protected/generics/ip/coregen/ise9/ise_92i_ip_update2_install.zip"

- mkdir ise_92i_ip_update2_install
- mv ise_92i_ip_update2_install.zip ise_92i_ip_update2_install
- cd ise_92i_ip_update2_install
- unzip ise_92i_ip_update2_install.zip
- rm ise_92i_ip_update2_install.zip
- sudo ./setup.lin
    * Enter "/tools/xilinx/ISE9.2i_lin" for "Select Destination Directory"
    * Press "OK"
    * Select "ISE 9.2i IP Update 2" for "Update Name"
    * Press "OK"
    * Press "OK"
    * Press "OK"

- sudo rm -rf /tools/xilinx/ISE9.2i_lin/.backup
- cd ..
- sudo rm -rf ise_92i_ip_update2_install

```

• Install EDK service pack

```

- In the guest VNC session
- epiphany "http://www.xilinx.com/support/download/" &
    * Download "http://download.xilinx.com/protected/generics/edk/EDK_9_2_02i_lin.zip"

- mkdir EDK_9_2_02i_lin
- mv EDK_9_2_02i_lin.zip EDK_9_2_02i_lin
- cd EDK_9_2_02i_lin
- unzip EDK_9_2_02i_lin.zip
- rm EDK_9_2_02i_lin.zip
- sudo ./setup
    * Enter "/tools/xilinx/EDK9.2i_lin" for "Select Destination Directory"
    * Press "OK"
    * Select "EDK 9.2i Service Pack 2" for "Update Name"
    * Press "OK"
    * Press "OK"
    * Press "OK"

- sudo rm -rf /tools/xilinx/EDK9.2i_lin/.backup
- cd ..
- sudo rm -rf EDK_9_2_02i_lin

```

• Install ChipScope Pro

```

- In the guest VNC session
- epiphany "http://www.xilinx.com/support/download/" &
    * Download "http://download.xilinx.com/protected/generics/chipscope/ChipScope_Pro_9_2i_lin.zip"

- mkdir ChipScope_Pro_9_2i_lin
- mv ChipScope_Pro_9_2i_lin.zip ChipScope_Pro_9_2i_lin
- cd ChipScope_Pro_9_2i_lin
- unzip ChipScope_Pro_9_2i_lin.zip
- rm ChipScope_Pro_9_2i_lin.zip
- sudo ./setup
    * Press "Next"
    * Enter "xxxx-xxxx-xxxx-xxxx" for "Registration ID"
    * Press "Next"
    * Select "I accept the terms of this software license."
    * Press "Next"
    * Enter "/tools/xilinx/ISE9.2i_lin/chipscope" for "Select Destination Directory"
    * Press "Next"
    * Select "ChipScope Common Files" for "Select Installation Options"
    * Press "Next"
    * Select "CHIPSCOPE" for "Select Installation Options"
    * Press "Next"
    * Press "Install"
    * Press "OK"

- sudo /tools/xilinx/ISE9.2i_lin/chipscope/bin/linux/cs_register.sh
    * Press "Add License"

```

```

    * Enter "xxxx-xxxx-xxxx-xxxx" for "Enter your 16 digit registration code"
    * Press "OK"
    * Press "OK"
- cd ..
- rm -rf ChipScope_Pro_9_2i_lin

```

- Install ChipScope Pro service pack
 - In the guest VNC session
 - epiphany "http://www.xilinx.com/support/download/" &
 - * Download "http://download.xilinx.com/protected/generics/chipscope/ChipScope_Pro_9_2_04i_lin.zip"
 - mkdir ChipScope_Pro_9_2_04i_lin
 - mv ChipScope_Pro_9_2_04i_lin.zip ChipScope_Pro_9_2_04i_lin
 - cd ChipScope_Pro_9_2_04i_lin
 - unzip ChipScope_Pro_9_2_04i_lin.zip
 - rm ChipScope_Pro_9_2_04i_lin.zip
 - sudo ./setup
 - * Enter "/tools/xilinx/ISE9.2i_lin/chipscope" for "Select Destination Directory"
 - * Press "Next"
 - * Select "ChipScope Pro 9.2i Service Pack 4" for "Update Name"
 - * Press "OK"
 - * Press "OK"
 - * Press "OK"
 - sudo rm -rf /Xilinx
 - sudo rm -rf /tools/xilinx/ISE9.2i_lin/chipscope.backup
 - cd ..
 - sudo rm -rf ChipScope_Pro_9_2_04i_lin

- Set up FLEXlm

```

- In the guest VNC session
- sudo pico /tools/flexlm/flexlm.sh
    * Add appropriate license variables
- sudo pico /tools/flexlm/flexlm.cshrc
    * Add appropriate license variables

```

- Fix home ownerships

```

- In the guest VNC session
- sudo chown -R upc:upc /home/upc/. * /home/upc/*

```

Skip to Section 9 or 11 if you have downloaded the pre-built VM image and do or do not desire to install commercial drivers, respectively.

8 RAMP Blue Build

- Set up shadow map

```

- In the guest VNC session
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/shadow_map"

```

- Set up scripts

```

- In the guest VNC session
- cd /home/proj/blue
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Scripts"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/Scripts" Shadow/Scripts
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Scripts.stat"
- chmod +x Scripts/file_distribute_shadows.sh
- chmod +x Scripts/file_set_status.sh
- Scripts/file_distribute_shadows.sh Scripts\ Scripts\
- Scripts/file_set_status.sh 0 0 1 Scripts < Scripts.stat
- rm -rf Shadow
- rm Scripts.stat
- rm -rf Scripts/.svn
- mv Scripts/. * Scripts/* .
- rmdir Scripts
- pico /home/upc/.bashrc
    * Add the following lines:
      . source /home/proj/blue/.bashrc
      .
      . cd /home/proj/blue

```

The guest working directory is assumed to be `/home/proj/blue` in the remainder of this guide.

• Build x86 Linux kernel

```
- In the guest VNC session
- Original source: sudo apt-get install linux-source-2.6.18
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/linux-source-2.6.18" linux-source-2.6.18-1
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/linux-source-2.6.18" Shadow/linux-source-2.6.18
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/linux-source-2.6.18.stat"
- ./file_distribute_shadows.sh linux-source-2.6.18\ linux-source-2.6.18-1\
- ./file_set_status.sh 0 0 1 linux-source-2.6.18-1 < linux-source-2.6.18.stat
- rm -rf Shadow
- rm linux-source-2.6.18.stat
- cd linux-source-2.6.18
- make oldconfig
- make menuconfig
  * Select "Exit"
  * Press Enter
  * Select "Yes" for "Do you wish to save your new kernel configuration?"
  * Press Enter
- make-kpkg clean
- fakeroot make-kpkg --revision=custom.1.0 --initrd kernel_image
- cd ..
- sudo dpkg -i linux-image-2.6.17_custom.1.0_i386.deb
- sudo pico /boot/grub/menu.lst
  * Move the following options to the top of the list:
    . title Debian GNU/Linux, kernel 2.6.18
    . root (hd0,1)
    . kernel /boot/vmlinuz-2.6.18 root=/dev/md0 ro
    . initrd /boot/initrd.img-2.6.18
    . savedefault
    .
    . title Debian GNU/Linux, kernel 2.6.18 (single-user mode)
    . root (hd0,1)
    . kernel /boot/vmlinuz-2.6.18 root=/dev/md0 ro single
    . initrd /boot/initrd.img-2.6.18
    . savedefault
- sudo shutdown -r now
```

• Install VMware Tools

```
- In the host VMware
  * Press "VM" → "Install VMware Tools..."
  * Press "Install"
- In the guest desktop session
- mount /dev/hdc
- tar xfvzp /media/cdrom0/VMwareTools-1.0.4-56528.tar.gz
- umount /dev/hdc
- cd vmware-tools-distrib
- sudo ./vmware-install.pl
  * Enter "/usr/bin" for "In which directory do you want to install the binary files?"
  * Enter "/etc" for "What is the directory that contains the init directories (rc0.d/ to rc6.d/)?"
  * Enter "/etc/init.d" for "What is the directory that contains the init scripts?"
  * Enter "/usr/sbin" for "In which directory do you want to install the daemon files?"
  * Enter "/usr/lib/vmware-tools" for "In which directory do you want to install the library files?"
  * Enter "yes" for "The path "/usr/lib/vmware-tools" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?"
  * Enter "/usr/share/doc/vmware-tools" for "In which directory do you want to install the documentation files?"
  * Enter "yes" for "The path "/usr/share/doc/vmware-tools" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?"
  * Enter "yes" for "Before running VMware Tools for the first time, you need to configure it by invoking the following command: "/usr/bin/vmware-config-tools.pl". Do you want this program to invoke the command for you now?"
  * Enter "yes" for "None of the pre-built vmhgfs modules for VMware Tools is suitable for your running kernel. Do you want this program to try to build the vmhgfs module for your system (you need to have a C compiler installed on your system)?"
  * Enter "/lib/modules/2.6.18/build/include" for "What is the location of the directory of C header files that match your running kernel?"
  * Enter "7" for "1280x1024" for "Please choose one of the following display sizes (1 - 13)"
- cd ..
```

• Build EtherApe

```
- In the guest VNC session
- Original source: wget "http://downloads.sourceforge.net/etherape/etherape-0.9.7.tar.gz"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/etherape-0.9.7" etherape-0.9.7-2
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/etherape-0.9.7" Shadow/etherape-0.9.7
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/etherape-0.9.7.stat"
- ./file_distribute_shadows.sh etherape-0.9.7\ etherape-0.9.7-2\
- ./file_set_status.sh 0 0 1 etherape-0.9.7-2 < etherape-0.9.7.stat
```

```

- rm -rf Shadow
- rm etherape-0.9.7.stat
- find etherape-0.9.7-2 -exec touch -d "`date`" {} \;
- cd etherape-0.9.7-2
- ./configure
- make prefix=/usr
- sudo make install prefix=/usr
- cd ..

```

• Build PowerPC cross-compiler

```

- In the guest VNC session
- Original source: wget "http://www.kegel.com/crosstool/crosstool-0.42.tar.gz"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/crosstool-0.42"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/crosstool-0.42" Shadow/crosstool-0.42
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/crosstool-0.42.stat"
- ./file_distribute_shadows.sh crosstool-0.42\ / crosstool-0.42\
- ./file_set_status.sh 1 0 1 crosstool-0.42 < crosstool-0.42.stat
- rm -rf Shadow
- rm crosstool-0.42.stat
- cd crosstool-0.42
- export TARGET=powerpc-405-linux
- export PREFIX=/home/proj/blue/$TARGET-0
- unset LD_LIBRARY_PATH
- sudo mkdir /opt/crosstool
- sudo chown upc:upc /opt/crosstool
- ./demo-ppc405.sh
- rm -rf /home/upc/downloads
- sudo rmdir /opt/crosstool
- cd ..
- ln -s powerpc-405-linux-0 powerpc-405-linux

```

• Set up control FPGA hardware

```

- In the guest VNC session
- This system can be built only with ISE 9.1.03i (XST 8.2.03i) and EDK 9.1.02i
- mkdir uclinux-bee2-ctrl-1
- svn checkout --revision 857 "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/BEE2/applications/reference/MPI_demo/IPS_Ctrlfpga"
uclinux-bee2-ctrl-1/reference
- cd uclinux-bee2-ctrl-1/reference
- mkdir implementation
- cd ../../
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Binaries"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/Binaries" Shadow/Binaries
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Binaries.stat"
- ./file_distribute_shadows.sh Binaries\ / Binaries\
- ./file_set_status.sh 0 0 1 Binaries < Binaries.stat
- rm -rf Shadow
- rm Binaries.stat
- mv Binaries/download-ctrl.bit uclinux-bee2-ctrl-1/reference/implementation/download.bit
- rm -rf Binaries

```

• Build PowerPC Linux kernel

```

- In the guest VNC session
- Original source: rsync -avzP "rsync://source.mvista.com/linuxppc-2.4" .
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/linuxppc-2.4" linuxppc-2.4-1
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/linuxppc-2.4" Shadow/linuxppc-2.4
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/linuxppc-2.4.stat"
- ./file_distribute_shadows.sh linuxppc-2.4\ / linuxppc-2.4-1\
- ./file_set_status.sh 0 0 1 linuxppc-2.4-1 < linuxppc-2.4.stat
- rm -rf Shadow
- rm linuxppc-2.4.stat
- cd linuxppc-2.4
- make oldconfig
- make menuconfig
  * Select "Exit"
  * Press Enter
  * Select "Yes" for "Do you wish to save your new kernel configuration?"
  * Press Enter
- make clean; make dep; make zImage
- cd ..

```

• Build control FPGA SystemACE file

- In the guest VNC session
- cp linuxppc-2.4-1/arch/ppc/boot/images/zImage.embedded uclinux-bee2-ctrl-1/reference
- cd uclinux-bee2-ctrl-1/reference
- ./mkace.sh
- cd ../../

- Build test suite CF card

- In the guest VNC session
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-test-suite"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/bee2-test-suite" Shadow/bee2-test-suite
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-test-suite.stat"
- sudo ./file_distribute_shadows.sh bee2-test-suite/ bee2-test-suite/
- sudo ./file_set_status.sh 0 2 1 bee2-test-suite < bee2-test-suite.stat
- rm -rf Shadow
- rm bee2-test-suite.stat
- sudo find bee2-test-suite/test_suite -type d -name .svn -exec rm -rf {} \;
- cd bee2-test-suite
- In the host VMware
 - * Select "VM" → "Removable Devices" → "USB Devices" → "SanDisk ImageMate CF Reader/Writer"
- In the guest VNC session
- Insert CF card
- pico run.sh
 - * Set CF device:
 - ./format-1.sh <dev> ``echo -n test_suite/*``
 - ./format-1.sh /dev/sdb ``echo -n test_suite/*``
- sudo ./run.sh
- Remove CF card
- In the host VMware
 - * Deselect "VM" → "Removable Devices" → "USB Devices" → "SanDisk ImageMate CF Reader/Writer"
- In the guest VNC session
- cd ..

- Set up Linux CF card

- In the guest VNC session
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-linux"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/bee2-linux" Shadow/bee2-linux
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-linux.stat"
- sudo ./file_distribute_shadows.sh bee2-linux/ bee2-linux/
- sudo ./file_set_status.sh 0 2 1 bee2-linux < bee2-linux.stat
- rm -rf Shadow
- rm bee2-linux.stat
- sudo find bee2-linux/linux -type d -name .svn -exec rm -rf {} \;

- Build custom CF card

- In the guest VNC session
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-custom"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/bee2-custom" Shadow/bee2-custom
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/bee2-custom.stat"
- sudo ./file_distribute_shadows.sh bee2-custom/ bee2-custom/
- sudo ./file_set_status.sh 0 2 1 bee2-custom < bee2-custom.stat
- rm -rf Shadow
- rm bee2-custom.stat
- sudo find bee2-custom/custom -type d -name .svn -exec rm -rf {} \;
- sudo cp uclinux-bee2-ctrl-1/reference/implementation/system.ace bee2-custom/custom
- cd bee2-custom
- In the host VMware
 - * Select "VM" → "Removable Devices" → "USB Devices" → "SanDisk ImageMate CF Reader/Writer"
- In the guest VNC session
- Insert CF card
- pico run.sh
 - * Set CF device:
 - ./format-1.sh <dev> ``echo -n custom/*``
 - ./format-1.sh /dev/sdb ``echo -n custom/*``
- sudo ./run.sh
- Remove CF card
- Repeat for remaining boards
- In the host VMware
 - * Deselect "VM" → "Removable Devices" → "USB Devices" → "SanDisk ImageMate CF Reader/Writer"
- In the guest VNC session

```
- cd ..
```

• Set up PowerPC root file system

- In the guest VNC session

```
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/linuxppc-root-min" linuxppc-root-min-2
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/linuxppc-root-min" Shadow/linuxppc-root-min
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/iptables-1.3.7.stat"
- sudo ./file_distribute_shadows.sh linuxppc-root-min\ linuxppc-root-min-2\
- sudo ./file_set_status.sh 1 1 1 linuxppc-root-min-2 < linuxppc-root-min.stat
- rm -rf Shadow
- rm linuxppc-root-min.stat
- sudo find linuxppc-root-min-2 -type d -name .svn -exec rm -rf {} \;
- ssh-keygen -t rsa
- Enter "/home/upc/.ssh/id_rsa" for "Enter file in which to save the key"
- Enter "" for "Enter passphrase"
- Enter "" for "Enter same passphrase again"
- sudo cp /home/proj/upc/.ssh/id_rsa.pub linuxppc-root-min-2/root/.ssh/authorized_keys
- i=0; while [ $i -lt 32 ]; do echo "board$i `cat linuxppc-root-min-2/etc/ssh/ssh_host_rsa_key.pub`" >> /home/upc/.ssh/known_hosts; i=`expr $i + 1`; done
- ssh-keygen -H
- rm /home/upc/.ssh/known_hosts.old
```

• Build iptables

- In the guest VNC session

- Original source: `wget "http://www.netfilter.org/projects/iptables/files/iptables-1.3.7.tar.bz2"`

```
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/iptables-1.3.7"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/iptables-1.3.7" Shadow/iptables-1.3.7
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/iptables-1.3.7.stat"
- ./file_distribute_shadows.sh iptables-1.3.7\ iptables-1.3.7\
- ./file_set_status.sh 0 0 1 iptables-1.3.7 < iptables-1.3.7.stat
- rm -rf Shadow
- rm iptables-1.3.7.stat
- cp -r iptables-1.3.7 iptables-1.3.7-1
- cp -r iptables-1.3.7 iptables-1.3.7-2
- rm -rf iptables-1.3.7
- cd iptables-1.3.7-1
- make CC=gcc KERNEL_DIR=/home/proj/blue/linux-source-2.6.18-1 PREFIX=/usr
- sudo make install KERNEL_DIR=/home/proj/blue/linux-source-2.6.18-1 PREFIX=/usr
- cd ..
- cd iptables-1.3.7-2
- make CC=powerpc-405-linux-gnu-gcc KERNEL_DIR=/home/proj/blue/linuxppc-2.4-1 PREFIX=/usr
- sudo make install KERNEL_DIR=/home/proj/blue/linuxppc-2.4-1 PREFIX=/home/proj/blue/linuxppc-root-min-2/usr
- cd ..
```

• Set up sting root file system

- In the guest VNC session

```
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/sting-root"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/sting-root" Shadow/sting-root
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/sting-root.stat"
- sudo ./file_distribute_shadows.sh sting-root\ sting-root\
- sudo ./file_set_status.sh 1 2 1 sting-root < sting-root.stat
- rm -rf Shadow
- rm sting-root.stat
- sudo find sting-root -type d -name .svn -exec rm -rf {} \;
- sudo mv sting-root/etc/network/interfaces .
- sudo cp -r sting-root/* /
- sudo mv interfaces sting-root/etc/network
- sudo sh -c "cat sting-root/etc/network/interfaces >> /etc/network/interfaces"
- sudo pico /etc/network/interfaces
    * Merge lines from sting-root/etc/network/interfaces
- sudo shutdown -r now
```

• Build MicroBlaze cross-compiler

- In the guest VNC session

- Original source: `wget "http://www.itee.uq.edu.au/~jwilliams/mblaze-uclinux/Toolchain/_files/microblaze-gcc-sources-20051121.tar.gz"`

```
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/microblaze-gcc-sources" microblaze-gcc-sources-1
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/microblaze-gcc-sources" Shadow/microblaze-gcc-sources
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/microblaze-gcc-sources.stat"
- ./file_distribute_shadows.sh microblaze-gcc-sources\ microblaze-gcc-sources-1\
- ./file_set_status.sh 0 0 1 microblaze-gcc-sources-1 < microblaze-gcc-sources.stat
- rm -rf Shadow
```

```

- rm microblaze-gcc-sources.stat
- cp -r microblaze-gcc-sources-1 microblaze-gcc-sources-2
- find microblaze-gcc-sources-1 -exec touch -d "`date`" {} \;
- find microblaze-gcc-sources-2 -exec touch -d "`date`" {} \;
- cd microblaze-gcc-sources-1
- ./do_everything.csh
- cd ..
- cd microblaze-gcc-sources-2
- pico srcs/gcc/gcc/Makefile.in
  * Change the following line:
  . LIBGCC2_CFLAGS = -O2 $(LIBGCC2_INCLUDES) $(GCC_CFLAGS) $(TARGET_LIBGCC2_CFLAGS) $(LIBGCC2_DEBUG_CFLAGS) $(GTHREAD_FLAGS) -DIN_LIBGCC2
  -D_GCC_FLOAT_NOT_NEEDED @inhibit_libc@
  * To the following line:
  . LIBGCC2_CFLAGS = -O2 $(LIBGCC2_INCLUDES) $(GCC_CFLAGS) $(TARGET_LIBGCC2_CFLAGS) $(LIBGCC2_DEBUG_CFLAGS) $(GTHREAD_FLAGS) -DIN_LIBGCC2
  -D_GCC_FLOAT_NOT_NEEDED -DMB_DP_FPU @inhibit_libc@
- ./do_everything.csh
- cd ..
- ln -s microblaze-gcc-sources-1 microblaze-gcc-sources

```

• Install RDLC2

```

- In the guest VNC session
- wget "http://ramp.eecs.berkeley.edu/Downloads/RDLC_2.2007.8.13_Base_JAR_and_Examples.zip"
- mkdir RDLC-2.2007.8.13
- mv "RDLC_2.2007.8.13_Base_JAR_and_Examples.zip" RDLC-2.2007.8.13
- cd RDLC-2.2007.8.13
- unzip "RDLC_2.2007.8.13_Base_JAR_and_Examples.zip"
- rm "RDLC_2.2007.8.13_Base_JAR_and_Examples.zip"
- cd ..

```

• Build user FPGA hardware

```

- In the guest VNC session
- mkdir uclinux-bee2-user-rdl-1
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/UserFPGA" uclinux-bee2-user-rdl-1/reference
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/UserFPGA" Shadow/UserFPGA
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/UserFPGA.stat"
- ./file_distribute_shadows.sh UserFPGA/ uclinux-bee2-user-rdl-1/reference/
- ./file_set_status.sh 0 0 1 uclinux-bee2-user-rdl-1/reference < UserFPGA.stat
- rm -rf Shadow
- rm UserFPGA.stat
- cd uclinux-bee2-user-rdl-1/reference
- cd doc-source
- pdflatex instructions.tex
- pdflatex instructions.tex
- evince instructions.pdf &
  * Follow instructions as below
- cd ..
- cd ../../
- java rdlc2.Main &
  * Select "Generate Mapped Design" for "Command"
  * Enter "upc" for "Author"
  * Enter "Generated by RDLC v2.2007.8.13 at xx:xx:xx xx-xx-xxxx" for "Description"
  * Enter "/home/proj/blue/uclinux-bee2-user-rdl-1/reference/RAMPBlue.rdl" for "Input RDL File"
  * Enter "":Maps::BEE2" for "Dynamic Root ID"
  * Select "Auto Shells"
  * Select "Run Plugins"
  * Select "Run Back End Tools"
  * Enter "/home/proj/blue/uclinux-bee2-user-rdl-1/reference-out" for "Output Path"
  * Press "Generate Mapped Design"
- cd uclinux-bee2-user-rdl-1/reference-out
- cd Maps/BEE2/RDLC_ISE_BEE2
- ise RDLC_ISE_BEE2.npl &
  * Press "Yes" for "Do you want to update the project?"
  * Press "Generate Programming File"
- cd ../../..
- cd ../../
- cd uclinux-bee2-user-rdl-1/reference
- cd BitInit
- ./init_bram.sh ../../reference-out
- cd ..
- cd ../../

```

• Build MicroBlaze uClinix kernel


```

- In the guest VNC session
- mkdir uClinux-1
- Original source: cvs -d :pserver:anonymous@cvs.uclinux.org:/var/cvs -z 3 checkout -P uClinux-2.4.x
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/uClinux-2.4.x" uClinux-1/uClinux-2.4.x
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/uClinux-2.4.x" Shadow/uClinux-2.4.x
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/uClinux-2.4.x.stat"
- ./file_distribute_shadows.sh uClinux-2.4.x/ uClinux-1/uClinux-2.4.x/
- ./file_set_status.sh 0 0 1 uClinux-1/uClinux-2.4.x < uClinux-2.4.x.stat
- rm -rf Shadow
- rm uClinux-2.4.x.stat
- Original source: cvs -d :pserver:anonymous@cvs.uclinux.org:/var/cvs -z 3 checkout -P uClinux-dist
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/uClinux-dist" uClinux-1/uClinux-dist
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/uClinux-dist" Shadow/uClinux-dist
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/uClinux-dist.stat"
- ./file_distribute_shadows.sh uClinux-dist/ uClinux-1/uClinux-dist/
- ./file_set_status.sh 0 0 1 uClinux-1/uClinux-dist < uClinux-dist.stat
- rm -rf Shadow
- rm uClinux-dist.stat
- find uClinux-1/uClinux-dist -exec touch -d "`date`" {} \;
- cd uClinux-1/uClinux-dist
- ln -s ../uClinux-2.4.x linux-2.4.x
- pico vendors/Xilinx/bee2/etc/rc/ifconfig.eth
  * Set system parameters: n m o p u v w
- make menuconfig
  * Select "Kernel/Library/Defaults Selection"
  * Select "Select"
  * Press Enter
  * Select "Customize Kernel Settings"
  * Press y
  * Select "Customize Vendor/User Settings"
  * Press y
  * Select "Exit"
  * Press Enter
  * Select "Exit"
  * Press Enter
  * Select "Yes" for "Do you wish to save your new kernel configuration?"
  * Press Enter
  * Select "Exit"
  * Press Enter
  * Select "Yes" for "Do you wish to save your new kernel configuration?"
  * Press Enter
  * Select "Exit"
  * Press Enter
  * Select "Yes" for "Do you wish to save your new kernel configuration?"
  * Press Enter
- make clean; make dep; make
- cd ../..
- ln -s uClinux-1 uClinux

```

• Build LED utilities

```

- In the guest VNC session
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/led_util"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/led_util" Shadow/led_util
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/led_util.stat"
- ./file_distribute_shadows.sh led_util/ led_util/
- ./file_set_status.sh 0 0 1 led_util < led_util.stat
- rm -rf Shadow
- rm led_util.stat
- cd led_util
- ./make.sh
- sudo cp `find -mindepth 1 -maxdepth 1 -name "led_*" ! -name "*.*)" ../linuxppc-root-min-2/usr/bin
- cd ..

```

• Build FSL Ethernet utilities

```

- In the guest VNC session
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/fsl_enet_util"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/fsl_enet_util" Shadow/fsl_enet_util
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/fsl_enet_util.stat"
- ./file_distribute_shadows.sh fsl_enet_util/ fsl_enet_util/
- ./file_set_status.sh 0 0 1 fsl_enet_util < fsl_enet_util.stat
- rm -rf Shadow
- rm fsl_enet_util.stat
- cd fsl_enet_util

```

```

- mkdir x86 ppc mb
- ./make.sh
- sudo cp `find ppc -mindepth 1 -maxdepth 1 -name "fsl_*" ! -name "*.*)"` ../linuxppc-root-min-2/usr/bin
- cp `find mb -mindepth 1 -maxdepth 1 -name "fsl_*" ! -name "*.*)"` ../uClinux-1/uClinux-dist/vendors/Xilinx/bee2/bin
- x86/fsl_user_gen_map <n> <m> <o> <p> <u> <v> <w> > ../network-1.map
- x86/fsl_user_gen_map 1 4 8 4 1 1 1 > ../network-1.map
- id0=0; while [ $id0 -lt <id1> ]; do x86/fsl_user_gen_routes <n> <m> <o> $id0 < ../network-1.map >> ../network-1.routes; id0=`expr $id0 + 1`; done
- id0=0; while [ $id0 -lt 32 ]; do x86/fsl_user_gen_routes 1 4 8 $id0 < ../network-1.map >> ../network-1.routes; id0=`expr $id0 + 1`; done
- cd ..

```

- Build Berkeley UPC compiler

```

- In the guest VNC session
- mkdir berkeley_upc-2
- Original source: wget "http://upc.lbl.gov/download/release/berkeley_upc-2.6.0.tar.gz"
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/berkeley_upc-2.6.0" berkeley_upc-2/berkeley_upc-2.6.0
- svn checkout "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/Shadow/berkeley_upc-2.6.0" Shadow/berkeley_upc-2.6.0
- svn export "svn+ssh://rampsvnpub@wright.eecs.berkeley.edu/public/trunk/Development/Blue/berkeley_upc-2.6.0.stat"
- ./file_distribute_shadows.sh berkeley_upc-2.6.0\ berkeley_upc-2\berkeley_upc-2.6.0\
- ./file_set_status.sh 0 0 1 berkeley_upc-2/berkeley_upc-2.6.0 < berkeley_upc-2.6.0.stat
- rm -rf Shadow
- rm berkeley_upc-2.6.0.stat
- find berkeley_upc-2/berkeley_upc-2.6.0 -exec touch -d "`date`" {} \;
- mv npb_make.sh berkeley_upc-2
- cd berkeley_upc-2
- cd berkeley_upc-2.6.0
- ln -s gasnet/other/contrib/cross-configure-uclinux-microblaze cross-configure-uclinux-microblaze
- cd ..
- mkdir berkeley_upc-build-x86 berkeley_upc-build-mb
- unset GASNET_SPAWNFN
- unset GASNET_CSPAWN_CMD
- cd berkeley_upc-build-x86
- ../berkeley_upc-2.6.0/configure --disable-pthreads
- make prefix=/usr
- cd ..
- cd berkeley_upc-build-mb
- ../berkeley_upc-2.6.0/cross-configure-uclinux-microblaze --disable-pthreads
- make prefix=/usr
- cd ..
- mkdir NPB2.4-mb
- ./npb_make.sh mb "S" <id1> <id2> <did>
- ./npb_make.sh mb "S" 32 32 32
- cd ..
- ln -s berkeley_upc-2 berkeley_upc

```

- Remove shadow map

```

- In the guest VNC session
- rm shadow_map

```

- Fix build permissions

```

- In the guest VNC session
- chmod 755 /home/proj/blue

```

9 Commercial Driver Installation

- Build XPC4 driver

```

- In the guest VNC session
- wget "ftp://ftp.xilinx.com/pub/utilities/M1_workstation/linuxdrivers.2.6.tar.gz"
- tar xfvzp linuxdrivers.2.6.tar.gz
- chmod -R go-w linuxdrivers.2.6
- mv linuxdrivers.2.6 linuxdrivers.2.6-1
- rm linuxdrivers.2.6.tar.gz
- cd linuxdrivers.2.6-1/xpc4drv
- ./configure
- make
- sudo make install
- cd ../../

```

- Build WinDriver driver

```

- In the guest VNC session
- wget "http://www.jungo.com/st/download/WD811LN.tgz"
- tar xfvzp WD811LN.tgz
- chmod -R go-w WinDriver
- mv WinDriver WinDriver-1
- rm WD811LN.tgz
- cd WinDriver-1/redist
- ./configure
- make
- sudo make install
- cd ../../

```

Skip to Section 11 if you have downloaded the pre-built VM image.

10 Final VM Image Setup

- Shrink and defragment VM disk

```

- In the guest desktop session
- sudo vmware-toolbox
  * Select "Shrink"
  * Select "/" for "Select partitions you wish to shrink."
  * Press "Shrink"
  * Press "Yes" for "Do you want to shrink the disk(s)?"
  * Press "OK" for "The shrink process has completed."
  * Press "Close"
- sudo shutdown -h now
- In the host VMWare
  * Press "VM" → "Settings"
  * Select "Hardware"
  * Select "Hard Disk (SCSI 0:0)" for "Device"
  * Press "Defragment" for "Capacity"
  * Press "OK"

```

- Snapshot VM disk

```

- In the host VMWare
  * Press "Snapshot"
  * Press "VM" → "Settings"
  * Select "Options"
  * Select "Snapshot" for "Setting"
  * Select "Lock this snapshot" for "Current snapshot"
  * Select "Revert to snapshot" for "When Powering Off"
  * Press "OK"

```

11 RAMP Blue Demonstration

- Set up board configuration

```

- Set up according to default configuration
- Jumper pins 15 and 16 on panel header J31
- Populate DIMM 1 on control FPGA
- Populate DIMMs 1 and 3 on desired user FPGAs
- Repeat for remaining boards

```

- Set up board EEPROM

```

- In the guest VNC session
- minicom
- Insert test suite CF card
- Power on board
- In the minicom session
- set_eeeprom
  * Enter "console=ttty0 console=tttyS0,115200 nfsaddrs=10.1.0.<i + 1>:10.0.0.2:10.0.0.2:255.252.0.0:board<i>:eth0:none
  nfsroot=/home/proj/root/board<i> root=/dev/nfs rw mem=512M" for "Enter cmd line"
  * Enter "console=ttty0 console=tttyS0,115200 nfsaddrs=10.1.0.1:10.0.0.2:10.0.0.2:255.252.0.0:board0:eth0:none
  nfsroot=/home/proj/root/board0 root=/dev/nfs rw mem=512M" for "Enter cmd line"
  * Enter "006d12b2xyzz" for "Enter MAC"
  * Enter "006d12b21222" for "Enter MAC"
  * Enter "115200" for "Enter baud rate"
  * Enter "x.y.z" for "Enter serial number"
  * Enter "1.2.34" for "Enter serial number"
- Power off board
- Remove test suite CF card

```

- Repeat for remaining boards
- Run demonstration
 - In the guest VNC session
 - cp uclinux-bee2-user-rdl-1/reference/BitInit/download.bit .
 - cp uClinux-1/uClinux-dist/images/image.srec .
 - cp berkeley_upc-2/berkeley_upc-build-x86/opt/gasnet/other/amudp/amudprun .
 - cp network-1.map network.map
 - cp network-2.map network.routes
 - sudo ./ppc_make_root.sh <n> linuxppc-root-min-2
 - sudo ./ppc_make_root.sh 1 linuxppc-root-min-2
 - pico mb_demo_a.sh
 - * Set system parameters: n m o u v w
 - ./ppc_exec.sh <n> "cd /home/proj/blue; ./mb_down_3d_mesh.sh <n> <m> <o> <u> <v> <w>; ./mb_up_3d_mesh.sh <n> <m> <o> <u> <v> <w>"
 - ./ppc_exec.sh 1 "cd /home/proj/blue; ./mb_down_3d_mesh.sh 1 4 8 1 1 1; ./mb_up_3d_mesh.sh 1 4 8 1 1 1"
 - pico mb_demo_b.sh
 - * Set system parameters: n m o
 - ./mb_ping.sh <n> <m> <o>
 - ./mb_ping.sh 1 4 8
 - pico mb_demo_c.sh
 - * Set system parameters: id1
 - ./mb_spawn.sh <id1> / "echo ``hostname; dmesg``"
 - ./mb_spawn.sh 32 / "echo ``hostname; dmesg``"
 - pico mb_demo_d.sh
 - * Set system parameters: id1
 - ./amudprun -np <id1> berkeley_upc/NPB2.4-mb/<id1>/ep-S
 - ./amudprun -np 32 berkeley_upc/NPB2.4-mb/32/ep-S
 - pico mb_demo_e.sh
 - * Set system parameters: n m o
 - etherape -i dummy0 -d -t -B <n> -C <m> -D <o> &
 - etherape -i dummy0 -d -t -B 1 -C 4 -D 8 &
 - etherape -i dummy0 -d -t -B <n> -C <m> -D <o> -R network.routes &
 - etherape -i dummy0 -d -t -B 1 -C 4 -D 8 -R network.routes &
 - minicom
 - Insert custom CF cards
 - Power on boards
 - Wait for boot sequence to complete
 - In the guest VNC session
 - ./mb_demo_a.sh
 - tail -f board0.log
 - * Wait for boot sequence to complete
 - ./mb_demo_b.sh
 - * Make sure all nodes respond to ping
 - ./mb_demo_c.sh > nodes.log
 - wc -l nodes.log
 - * Make sure all nodes respond to telnet
 - grep -v reloc nodes.log | wc -l
 - * Make sure no nodes report errors
 - ./mb_demo_d.sh
 - In the guest VNC session
 - sudo ./mb_demo_e.sh
 - Power off boards
 - Remove custom CF cards
 - sudo shutdown -h now

12 Conclusion

Congratulations, you have completed the RAMP Blue setup! For additional support:

RDL and RDLc	Greg Gibeling	gdgib@eecs.berkeley.edu
Everything else, including this guide	Alex Krasnov	akrasnov@eecs.berkeley.edu