CHERI DE4 Factory Install Guide

Version 1.0

This interim document is not released for public consumption

Brooks Davis
SRI International and the University of Cambridge[*]

March 16, 2013

Abstract

This document describes the process of bootstrapping or recovering SRI International and the University of Cambridge's demo platform for the research prototype implementation of the Capability Hardware Enhanced RISC Instructions (CHERI) instruction set architecture (ISA). The demo platform uses Terasic's DE4 FPGA board and their MTL touchscreen to implement a (rather bulky) tablet computer. The document covers the process of writing an FPGA image and kernel to a DE4 board configured as shipped by Terasic or in common corruption cases.

The Factory Install Guide is targeted at integrators who will be configuring the demonstration hardware to provide to demonstrators. Instructions related to running the demo can be found in the *CHERI DE4 Getting Started Guide*. Advanced configuration and/or development can be found in the *CHERI User's Guide*. This guide includes an overview of the relevant portions of the physical hardware as well as a review of the features of demo software.

[*] Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

1 Prerequsites

To configure or reset DE4 for use with CHERI you will need an FPGA bitfile, a flash preparation kernel, the Altera Quartus 12 toolchain, and the `cherictl` program. All of these tools will be run on a Linux system or VM.

1.1 Files

There are currently three flash preparation images available. The `mdroot-demo` image writes a kernel with an embedded file system containing the CHERI touchscreen demo platform. The `mdroot-net` image writes a similar kernel, but with network tools include ssh and sshd. It can be temporarily configured for remote access. Configuraiton include ssh keys is lost at boot. The `sdroot` image writes a kernel that expects to find a root filesystem on the SDCard.

The current recommended flash preparation kernels are:

- `20130307-flash-cheribsd-de4-mdroot-demo-kernel.bz2`
- `20130307-flash-cheribsd-de4-mdroot-net-kernel.bz2`
- `20130307-flash-cheribsd-de4-sdroot-kernel.bz2`

Users of the `sdroot` kernel will also wish to obtain `20130307-cheribsd-sdcard.img.xz`, extract it using `unxz` and write it to an SD Card.

Each set of kernels has an associated recommended FPGA bitfile. As a rule, the recommended bitfile should be used due to the co-evolution of CPU and operating system features.

The current recommended bitfile is: `20130125-cheri-de4-bitfile-baz21-2eth.sof.bz2`

Cambridge users can find these files in the `/usr/groups/ctsrd/cheri` directory. A full list of CHERI snapshots, releases, and bitfile may be found at `https://ctsrd-trac.cl.cam.ac.uk/projects/cheri/wiki/CheriBSD`.

External users will find CHERI releases at `XXX:TBD`

1.2 Altera toolchain

You must have the Altera Quartus 12 toolchain installed and configured so that it can see the DE4's JTAG unit over USB. The tools `system-console` and `nios2-terminal` must be in your `PATH`. Cambridge users can accomplish this by sourcing the setup file `/usr/groups/ecad/setup.bash`. A simple way to test this is to run the `jtagconfig` command with the board connected and powered on:

```
$ jtagconfig
1) USB-Blaster [5-1]
  024090DD   EP4SGX230(.|ES)

$
```

If you see a USB-Blaster entry you should be good to go.

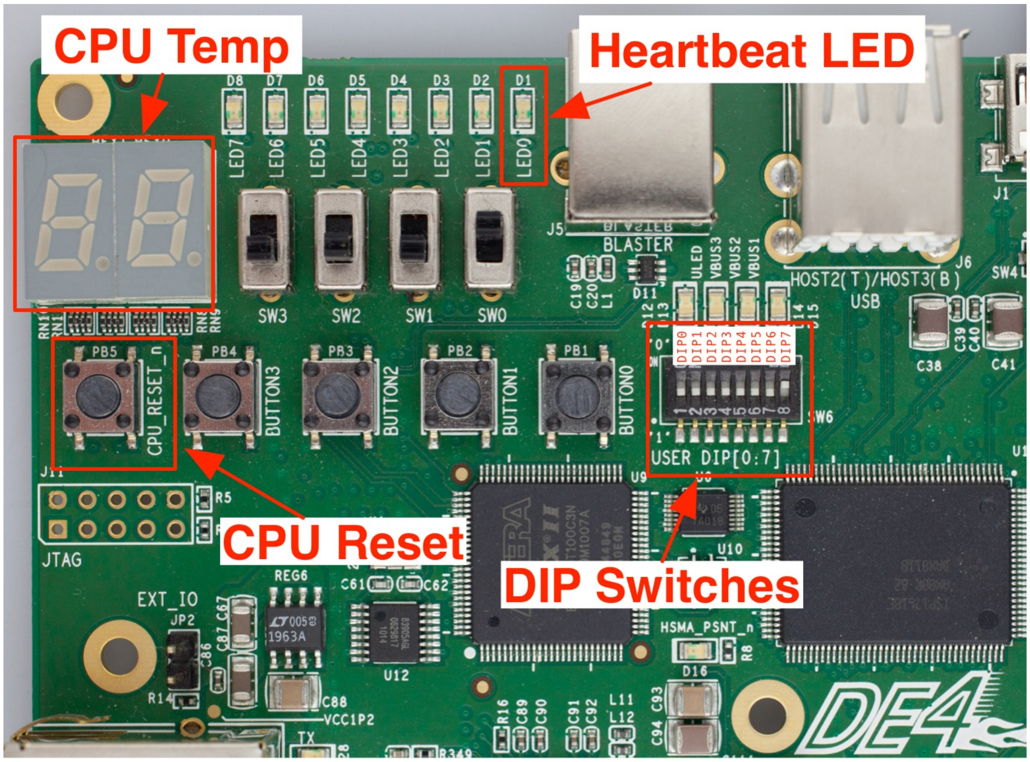


Figure 1: DE4 buttons and switches

1.3 `cherictl`

If you do not already have `cherictl` available, you can build it by checking out `svn+ssh://secsvn@svn-ctsrd.cl.cam.ac.uk/ctsrd/cherilibs/trunk`, changing the `tools/debug` directory, and running `make`. Your version of `cherictl` must support the `loadsof` and `loaddram` commands.

2 Loading the bitfile

First, make sure DIP0 and DIP1 on your board are in the ON (UP) position. The DIP switches can be seen in Figure 1. Note that they are numbered 0-7, but the physical switch is labled 1-8. This causes the bitfile to load with the loader paused. Second, connect the USB cable and turn to board on. The location of the JTAG USB port and power switch can be seen in Figure 2.

In one terminal, run the `system-console` command as follows:

```
$ system-console --server
TCP PORT: 37556
System Console server started on TCP port 37556
```

Take note of the TCP port number. You will need it in the next steps.

In another terminal, run the following commands:

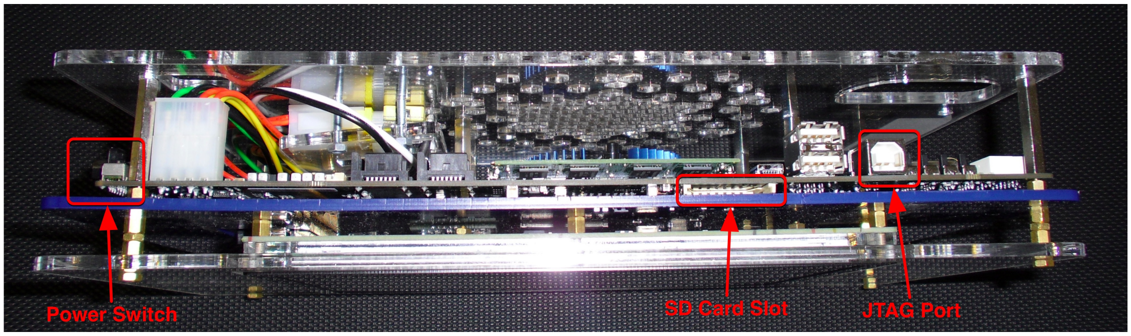


Figure 2: Top of the demo DE4 package

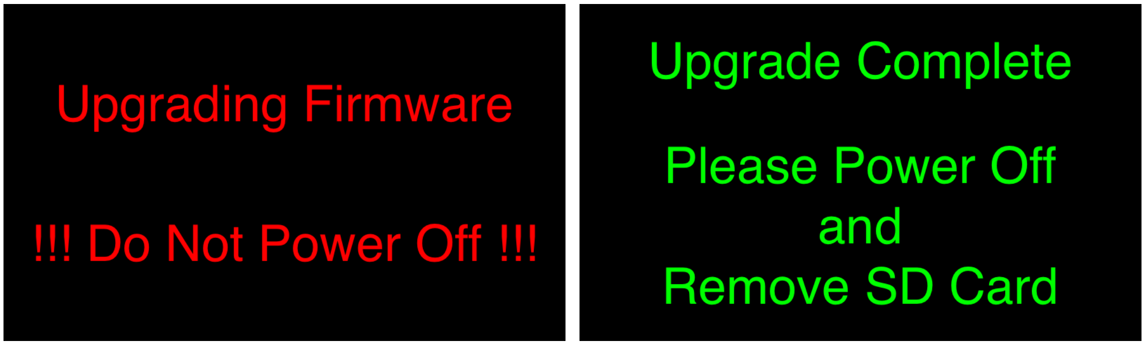
```
$ export SYSTEM_CONSOLE_PORT=37556
$ cherictl -p $SYSTEM_CONSOLE_PORT \
-z -f 20130125-cheri-de4-bitfile-baz21-2eth.sof.bz2 loadsof
1362602351 Command: get_service_paths master
1362602354 Path: {/devices/EP4SGX230(.|ES)@1#5-1/(link)/JTAG/(110:132 v1 #0)
+/phy_0/master}
1362602354 Command: open_service master {/devices/EP4SGX230(.|ES)@1#5-1/(lin
+k)/JTAG/(110:132 v1 #0)/phy_0/master}
1362602354 Command: device_download_sof {/devices/EP4SGX230(.|ES)@1#5-1/(lin
+k)/JTAG/(110:132 v1 #0)/phy_0/master} /var/tmp/cherictl.Y6oQkb.sof
1362602387 Command: close_service master {/devices/EP4SGX230(.|ES)@1#5-1/(li
+nk)/JTAG/(110:132 v1 #0)/phy_0/master}
1362602387 Done
$
```

The `cherictl loadsof` command should take about 40 seconds. The fans will run at full speed part way through the process and the touch screen may may display odd patterns before turning green at the end of the process.

3 Loading a kernel

To load a kernel run the following commands:

```
$ cherictl -p $SYSTEM_CONSOLE_PORT \
-z -f flash-cheribsd-de4-mdroot-demo-kernel.bz2 -a 0x100000 loaddram
1362604696 Command: get_service_paths master
1362604699 Path: {/devices/EP4SGX230(.|ES)@1#5-1/(link)/JTAG/(110:132 v1 #0)
+/phy_0/master}
1362604699 Command: open_service master {/devices/EP4SGX230(.|ES)@1#5-1/(lin
+k)/JTAG/(110:132 v1 #0)/phy_0/master}
1362604699 Command: master_write_from_file {/devices/EP4SGX230(.|ES)@1#5-1/(
+link)/JTAG/(110:132 v1 #0)/phy_0/master} /var/tmp/cherictl.YahVhU 0x100000
1362604784 Command: close_service master {/devices/EP4SGX230(.|ES)@1#5-1/(li
+nk)/JTAG/(110:132 v1 #0)/phy_0/master}
1362604785 Done
$
```



Upgrading Firmware

!!! Do Not Power Off !!!

Upgrade Complete

Please Power Off
and
Remove SD Card

(a) In Progress

(b) Complete

Figure 3: Upgrade Status Screens

A 32MB kernel will take around 90 seconds to load with the standard JTAG interface.

4 Flashing the device

Now we need to boot the kernel. We trigger booting with the command:

```
$ cherictl boot
Set $13 to 0000000000000000
$
```

The system will now boot and install the OS and flash images. If you have a touchscreen you will see images indicating that the system is being flashed as in Figure 3a and another indicating that it is done as seen in Figure 3b. If you do not have a touchscreen you can monitor progress using `nios2-terminal --instance 1` (you may wish to start this in another window prior to running the setreg command above). *WARNING: you must not have nios2-terminal running when running* `cherictl ... loadsof` *or* `cherictl ... loaddram`. The process of flashing board takes about 15 minutes.

```
$ nios2-terminal --instance 1
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-Blaster [5-1]", device 1, instance 1
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

Mini Bootloader Run
Test I2C on HDMI chip
Reset HDMI chipSetting clock_scale to 0x00000000000004E2
clock scale = 0x00000000000004E2 - passed
Correct vendor ID
Correct device ID
entry: platform_start()
cmd line:
envp:
memsize = 3eeffc00
Cache info:
  picache_stride    = 4096
  picache_loopcount = 4
  pdcache_stride    = 4096
  pdcache_loopcount = 4
cpu0: Unknown cid 0 processor v0.4
  MMU: Standard TLB, 40 entries
  L1 i-cache: direct-mapped with 512 sets, 32 bytes per line
  L1 d-cache: direct-mapped with 512 sets, 32 bytes per line
  Config1=0xcee07040<COP2>
Physical memory chunk(s):
0x1f95000 - 0x3eefffff, 1022799872 bytes (249707 pages)
Maxmem is 0x3ef00000
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 10.0-CURRENT #0 222633+edit: Wed Mar  6 19:22:48 UTC 2013
    bed22@zenith.cl.cam.ac.uk:/home/bed22/obj/mips.mips64/home/bed22/p4/beribs
+d/src/sys/BERI_DE4_MDROOT mips
Preloaded elf kernel "kernel" at 0xffffffff81f874f0.
...
start_init: trying /sbin/init
Beginning upgrade at Wed Mar  6 19:23:01 UTC 2013
/usr/sbin/flashit kernel /upgrades/beribsd-de4-mdroot-net-kernel.bz2
Validating /upgrades/beribsd-de4-mdroot-net-kernel.bz2
 Extracting to /tmp/flashit.QZ9vM2Mq/beribsd-de4-mdroot-net-kernel
Writing /tmp/flashit.QZ9vM2Mq/cheribsd-de4-mdroot-demo-kernel to cfid0 @ 0x020
+00000
dd if=/tmp/flashit.QZ9vM2Mq/cheribsd-de4-mdroot-demo-kernel of=/dev/cfid0 isee
+k=0 oseek=65536 conv=osync
64202+1 records in
64203+0 records out
32871936 bytes transferred in 168.259715 secs (195364 bytes/sec)
/usr/sbin/flashit fpga /upgrades/20130125-cheri-de4-bitfile-baz21-2eth.bin.bz2
Validating /upgrades/20130125-cheri-de4-bitfile-baz21-2eth.bin.bz2
Extracting to /tmp/flashit.HvvWnsCL/20130125-cheri-de4-bitfile-baz21-2eth.bin
Writing /tmp/flashit.HvvWnsCL/20130125-cheri-de4-bitfile-baz21-2eth.bin to cfi
+d0 @ 0x00020000
dd if=/tmp/flashit.HvvWnsCL/20130125-cheri-de4-bitfile-baz21-2eth.bin of=/dev/
+cfid0 iseek=256 oseek=256 conv=osync
23190+1 records in
23191+0 records out
11873792 bytes transferred in 62.522217 secs (189913 bytes/sec)
Upgrade complete at Wed Mar  6 19:30:54 UTC 2013

===> Installation complete.

===> you may now reboot or power off the system
```

5 Running the demo

Restore DIP0 and DIP1 to the OFF (DOWN) position, remove the SD Card, and power the system on. It should boot into the demo in about 60 seconds as described in the *CHERI DE4 Getting Started Guide*.