

FRAMEWORK FOR EMBEDDED CONCURRENT CONTROL (FECC)

USER GUIDE

Victor Gusev Lesau, Edward Chen, William A. Gruver, Dorian Sabaz

Simon Fraser University

iDEA Lab

<http://www.ensc.sfu.ca/idea/>

December 30, 2011

1. Project Versions

Ver.	Date	Author	Description
0.10	Dec 30, 2011	Victor Gusev Lesau, Edward Chen	Delivery contains: - RTL for complete embedded system including PRMs - Generated bitstreams (full and partial) - Linux kernel image - C applications for Linux (CLT, DP) and a standalone app for HA_CPU (HCA) - Standalone test application for HA_CPU The following functions are supported by Linux apps: - Reconfigure PRR_A/PRR_B with Incrementor, Decrementor and Blank PRMs - Request and report status

2. Contents

1.	Project Versions	2
2.	Contents	3
3.	Terminology	4
4.	Requirements	5
5.	Quick Start.....	6
6.	Appendix 1: Tool Installation—Linux VM.....	11
7.	Appendix 2: Tool Installation—Windows.....	12
8.	Appendix 3: Installing Kermit	13
9.	Appendix 4: Linux Bootup Log.....	14
10.	References.....	16

3. Terminology

Abbreviation	Description
CLT	Command line tool that recognizes user commands, creates packets and forwards to DP (Linux).
DP	Daemon process that keeps track of each PRR in a status table and forwards commands between the user and HA/PRR.
HA	Hardware Administrator
HA_CPU	CPU connected to Hardware Administrator (i.e. Microblaze_1)
HCA	HA_CPU application
LINUX_CPU	CPU running Linux (i.e. Microblaze_0)
PRM	Partially reconfigurable module
PRR	Partially reconfigurable region
XMD	Xilinx Microprocessor Debugger

4. Requirements

Hardware

- Xilinx ML505 board
- Xilinx Platform Cable USB II DLC10

Tools

The project has been developed by a team of two on different host machines: Linux and Windows.

Static embedded subsystem development and merged project testing:

- Ubuntu 10.04 development host
- Xilinx ISE: 9.2.04i_PR14 (PR patched), Application version J.40
- Xilinx EDK: 9.2.04, Application version EDK_Jm_SP2.3+0
- Xilinx Partial Reconfiguration patch R14
- Xilinx Impact 9.2.04i, Application version J.40
- PetaLogix PetaLinux 0.40 RC4 (kernel 2.6.20) [1]

DPR system wrapper and PRM development:

- Windows 7 Professional development host
- PlanAhead 10.8
- ISE 9.02.04i_PR14, Application version J.40
- EDK 9.2.02i, Application version EDK_Jm_SP2.3+0
- Xilinx Impact 9.2.04i, Application version J.40

Quick Start:

- Xilinx Impact configuration tool with cable drivers installed
- Ubuntu 10.04 development host or Windows 7 Professional development host with Xilinx Cygwin console installed

5. Quick Start

This section starts from unpacking the project and goes through running a simple reconfiguration sanity test. There are two PR regions. Each can be loaded with the following PR modules: incrementor (free running incrementing counter), decrementor (free running decrementing counter), blank (empty bitstream). Loading the empty bitstream is equivalent to unloading a PRM.

The following steps assume that the Xilinx ML505 board is connected to the host Linux machine via a Platform Cable and its UART—via a serial cable for console printing. Also, it is expected that Xilinx Impact configuration tool and Xilinx cable drivers are installed. The similar procedure can be done in Windows with the help of Xilinx Cygwin console installed.

1. Extract `pr_hardware.tar.gz` into a working directory.

```
tar -xzf pr_hardware.tar.gz
```

2. Open a Linux console (i.e. referred to as Embedded Console from this point) and connect to the embedded system's UART using a com port communication application such as Kermit (please refer to Appendix 3: Installing Kermit).



Both LINUX_CPU's and HA_CPU's stdin/stdout are mapped to the same UART on FPGA. Thus, messages from both applications are printed on the same embedded console and compete with each other. Some messages might get partially overwritten.

3. Configure the FPGA on the ML505 board with the following bitstream using Impact GUI.
`pr_hardware/hardware/Bitstreams/download_CF.bit`.

The provided bitstream has its block RAMs initialized with a demo application for HA_CPU that can run without Linux. The following will be printed in embedded console.

```
System ACE Controller Initialized
After HWICAP LookupConfig
HWICAP Initialized

DPR Framework Testing
Press a to Query PRR_A Status
Press b to Set PRR_A to Increment
... (etc)
```

4. Open a second Linux console (i.e. referred to as Host Console from this point) and create a link pointing to the location of directory that contains the HCA app.

```
cd pr_hardware/software/petalinux/software/petalinux-dist/scripts
ln -sf ../../../../../../hardware/edk/sw sw
```

5. Download and boot Linux kernel and HCA app on LINUX_CPU and HA_CPU respectively using XMD via processors' debugging ports. From the host console run `xmd.cmd` which in turn reads its commands from `xmd.opt`.

```
./xmd.cmd
```

The embedded console will print Linux bootup messages and stop at login prompt. See Appendix 4: Linux Bootup Log.

6. Log into Linux using 'root' as both user name and password.
7. Load Mailbox kernel driver. In embedded console type:

```
# mknod /tmp/mailbox_driver c 32 0
# insmod /lib/modules/kernel/drivers/misc/mailbox_driver.ko
```

Linux will print the following:

```
Initializing interrupts..... Done

Allocating MBox tx/rx buffers of 1024 bytes each..... Done

Registering IOs..... Done

Requesting IO ports.....
- Real address 0 : 0x71010000
.....DONE.....

Remapping IO ports.....
- Virtual address 0: 0x71010000
.....DONE.....

Registering driver to kernel..... Done

Creating netlink socket for communication with user-space..... Done
```

8. Load Daemon Process application. In embedded console type:

```
# /bin/pr_config &
```

Linux will print the following:

```
-----
PRR Control Table (2 records)

PRR Management Table record:
dest_ha      = 0
dest_prr     = 1
bitfilename  = na
inst_name    = na
is_loaded    = 0

PRR Management Table record:
dest_ha      = 0
dest_prr     = 2
bitfilename  = na
inst_name    = na
is_loaded    = 0
-----
```

9. Reconfigure PRR_A with a blank bitstream blka.bit. In embedded console type:

```
# /bin/prctl -bit blka.bit -dest 0 1
```

Linux will print the following:

```
Reconfigure (HA, PRR) (0, 1) with blka.bit
-----
PRR Control Table (2 records)

PRR Management Table record:
dest_ha      = 0
dest_prr     = 1
bitfilename  = blka.bit
inst_name    = na
is_loaded    = 0
```

Framework for Embedded Concurrent Control User Guide

```
PRR Management Table record:
dest_ha          = 0
dest_prr         = 2
bitfilename      = na
inst_name        = na
is_loaded        = 0
-----
Execute: PRR Reconfiguration
init_sysace: System ACE Controller Initialized
init_hwicap: running HWICAP LookupConfig
init_hwicap: running HWICAP initialization
init_hwicap: HWICAP initialized
download_bitstream: calling XHwIcap_CF2Icap()
HA_CPUreceive_packet_from_ha: Received a short packet from HA
Packet:
0x:  0 10  5  1 (0x 100501) (*HEADER*)
-----
PRR Control Table (2 records)

PRR Management Table record:
dest_ha          = 0
dest_prr         = 1
bitfilename      = blka.bit
inst_name        = na
is_loaded        = 1

PRR Management Table record:
dest_ha          = 0
dest_prr         = 2
bitfilename      = na
inst_name        = na
is_loaded        = 0
-----
```



Both Daemon Process and HCA have several sleep functions used for debugging purpose while messages are printed in verbose mode. For this demo the verbose mode has been removed, but the sleeping functions remain. It will take ~20-30 seconds before messages are printed in UART.

is_loaded status bit reports 1 once a bitstream has been loaded and PRR replied to a status query.

10. Reconfigure PRR_A with Incrementor bitstream inca.bit. In embedded console type:

```
# /bin/prctl -bit inca.bit -dest 0 1
```

Linux will print the following:

```
Reconfigure (HA, PRR) (0, 1) with inca.bit
-----
PRR Control Table (2 records)

PRR Management Table record:
dest_ha          = 0
dest_prr         = 1
bitfilename      = inca.bit
inst_name        = na
is_loaded        = 0

PRR Management Table record:
dest_ha          = 0
dest_prr         = 2
bitfilename      = na
inst_name        = na
is_loaded        = 0
```

Framework for Embedded Concurrent Control User Guide

```
-----
Execute: PRR Reconfiguration
init_sysace: System ACE Controller has already been initialized
init_hwicap: HWICAP has already been initialized
download_bitstream: calling XHwIcap_CF2Icap()
HA_CPUreceive_packet_from_ha: Received a short packet from HA
Packet:
0x: 10 0 5 1 (0x10000501) (*HEADER*)
-----
```

PRR Control Table (2 records)

PRR Management Table record:

dest_ha	= 0
dest_prr	= 1
bitfilename	= inca.bit
inst_name	= na
is_loaded	= 1

PRR Management Table record:

dest_ha	= 0
dest_prr	= 2
bitfilename	= na
inst_name	= na
is_loaded	= 0

11. Reconfigure PRR_B with Decrementor bitstream decb.bit. In embedded console type:

```
# /bin/prctl -bit decb.bit -dest 0 2
```

Linux will print the following:

Reconfigure (HA, PRR) (0, 2) with decb.bit

PRR Control Table (2 records)

PRR Management Table record:

dest_ha	= 0
dest_prr	= 1
bitfilename	= inca.bit
inst_name	= na
is_loaded	= 1

PRR Management Table record:

dest_ha	= 0
dest_prr	= 2
bitfilename	= decb.bit
inst_name	= na
is_loaded	= 0

```
Execute: PRR Reconfiguration
init_sysace: System ACE Controller has already been initialized
init_hwicap: HWICAP has already been initialized
download_bitstream: calling XHwIcap_CF2Icap()
HA_CPUreceive_packet_from_ha: Received a short packet from HA
Packet:
0x: 20 1 9 1 (0x20010901) (*HEADER*)
-----
```

PRR Control Table (2 records)

PRR Management Table record:

dest_ha	= 0
dest_prr	= 1
bitfilename	= inca.bit
inst_name	= na
is_loaded	= 1

PRR Management Table record:

dest_ha	= 0
dest_prr	= 2
bitfilename	= decb.bit
inst_name	= na

Framework for Embedded Concurrent Control User Guide

```
is_loaded      = 1
-----
```

12. Query status of HA 0 PRR 1. In embedded console type:

```
# /bin/prctl -ctlid 1 -dest 0 1
```

Linux will print the following:

```
Sending command 1 to (HA, PRR) (0, 1)
HA_CPUreceive_packet_from_ha: Received a short packet from HA
Packet:
0x: 10  0  5  1 (0x10000501) (*HEADER*)
-----
PRR Control Table (2 records)

PRR Management Table record:
dest_ha      = 0
dest_prr     = 1
bitfilename   = inca.bit
inst_name    = na
is_loaded    = 1

PRR Management Table record:
dest_ha      = 0
dest_prr     = 2
bitfilename   = decb.bit
inst_name    = na
is_loaded    = 1
-----
```

6. Appendix 1: Tool Installation—Linux VM

This section describes the steps taken to install the following Xilinx and PetaLogix tools on Ubuntu virtual machine development host: ISE, EDK, Platform Cable drivers and PetaLinux. It also goes over some caveats a user should be aware of when using the cable drivers in Virtual Machine (i.e. cable locking and resetting).

Xilinx

TBD—for the time being please refer to installation instructions of each tool vendor.

PetaLinux

Please refer to the following guide to download and install PetaLinux toolchain: [1].

7. Appendix 2: Tool Installation—Windows

This section describes the steps taken to install the following Xilinx tools on Windows development host: ISE, EDK, PlanAhead.

TBD—for the time being please refer to installation instructions of each tool vendor.

8. Appendix 3: Installing Kermit

Kermit is a console program used for communicating with embedded target. Please refer to [1] and [2] for its installation and setup instructions. The setup used for this project used a USB to serial converter since the development host did not have a COM port. Thus, `~/.kermrc` has been modified as the following:

```
set line /dev/ttyUSB0
set speed 115200
set carrier-watch off
set handshake none
set flow-control none
robust
set file type bin
set file name lit
set rec pack 1000
set send pack 1000
set key \127 \8
set key \8 \127
set window 5
```

9. Appendix 4: Linux Bootup Log

```

Found romfs @ 0x201b108c (0x001fa000)
#### klimit 201c9000 ####
Moving 0x001fa000 bytes from 0x201b108c to 0x201c887c
-----
Launching HA_CPU
-----
Enabling Bus Macros
New klimit: 0x203c3000
Linux version 2.6.20-uc0 (victor@ubuntu) (gcc version 3.4.1 ( PetaLinux 0.20 Build -rc1 050607 ))
#39 Fri Mar 18 02:10:23 PDT 2011
setup_cpuinfo: initialising
setup_cpuinfo: Using full CPU PVR support
setup_memory: max_mapnr: 0x2ffff
setup_memory: min_low_pfn: 0x20000
setup_memory: max_low_pfn: 0x10000
On node 0 totalpages: 65536
  DMA zone: 512 pages used for memmap
  DMA zone: 0 pages reserved
  DMA zone: 65024 pages, LIFO batch:15
  Normal zone: 0 pages used for memmap
Built 1 zonelists. Total pages: 65024
Kernel command line:
OPB INTC #0 at 0x71100000
PID hash table entries: 1024 (order: 10, 4096 bytes)
TIMER at 0x71C00000
disabling early console
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 255872k/262144k available
Calibrating delay loop... 61.84 BogoMIPS (lpj=309248)
Mount-cache hash table entries: 512
NET: Registered protocol family 16
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 8192 bind 4096)
TCP reno registered
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
uartlite.0: ttyUL0 at MMIO 0x71000000 (irq = 3) is a uartlite
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
physmap platform flash device: 01000000 at 89000000
CFI: Found no physmap-flash.0 device at location zero
physmap-flash physmap-flash.0: map_probe failed
uclinux[mtd]: RAM probe address=0x201c887c size=0x1fa000
Creating 1 MTD partitions on "RAM":
0x00000000-0x001fa000 : "ROMfs"
uclinux[mtd]: set ROMfs to be root filesystem index=0
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
VFS: Mounted root (romfs filesystem) readonly.
Freeing unused kernel memory: 76k freed
Mounting proc:
Mounting var:
Populating /var:
Running local start scripts.
Mounting /etc/config:
Populating /etc/config:
flatfsd: Nonexistent or bad flatfs (-48), creating new one...
flatfsd: Failed to write flatfs (-48): No such device
flatfsd: Created 5 configuration files (185 bytes)
Mounting sysfs:
Setting hostname:
Setting up interface lo:
Setting up interface eth0:
SIOCSIFADDR: No such device
eth0: unknown interface: No such device

```

Framework for Embedded Concurrent Control User Guide

```
Starting portmap:  
Starting thttpd:  
  
uclinux login: root  
Password:  
#
```

10. References

- [1] PetaLogix. (Dec 28, 2011). *PetaLinux Installation Procedure*. [Online]. Available: <http://www.petalogix.com/resources/documentation/petalinux/petalinux-installation-guide/petalinux-installation-procedure>.
- [2] Panix. (Sep 29, 2011). *The Kermit Project*. [Online]. Available: <http://www.kermitproject.org/>.