



## Application Note

---

# Using Impulse C with BlueCat Linux 5.4.2 on MicroBlaze via FSL

**Mei A. Xu**

Impulse Accelerated Technologies, Inc.

Copyright© 2008 Impulse Accelerated Technologies, Inc.

## Overview

This application note describes how to incorporate an Impulse CoDeveloper-generated hardware module with a MicroBlaze microprocessor running the LynuxWorks BlueCat Linux 5.4.2 operating system, using the Xilinx FSL interface for data communication. A complex FIR filter application is used as an example to demonstrate the design method.

BlueCat Linux package version 5.4.2 from LynuxWorks provides a Linux kernel and the development tools needed to compile Linux for the MicroBlaze V7 embedded processor. Impulse CoDeveloper allows you to create a hardware accelerator module that is connected via FSL interfaces to the MicroBlaze.

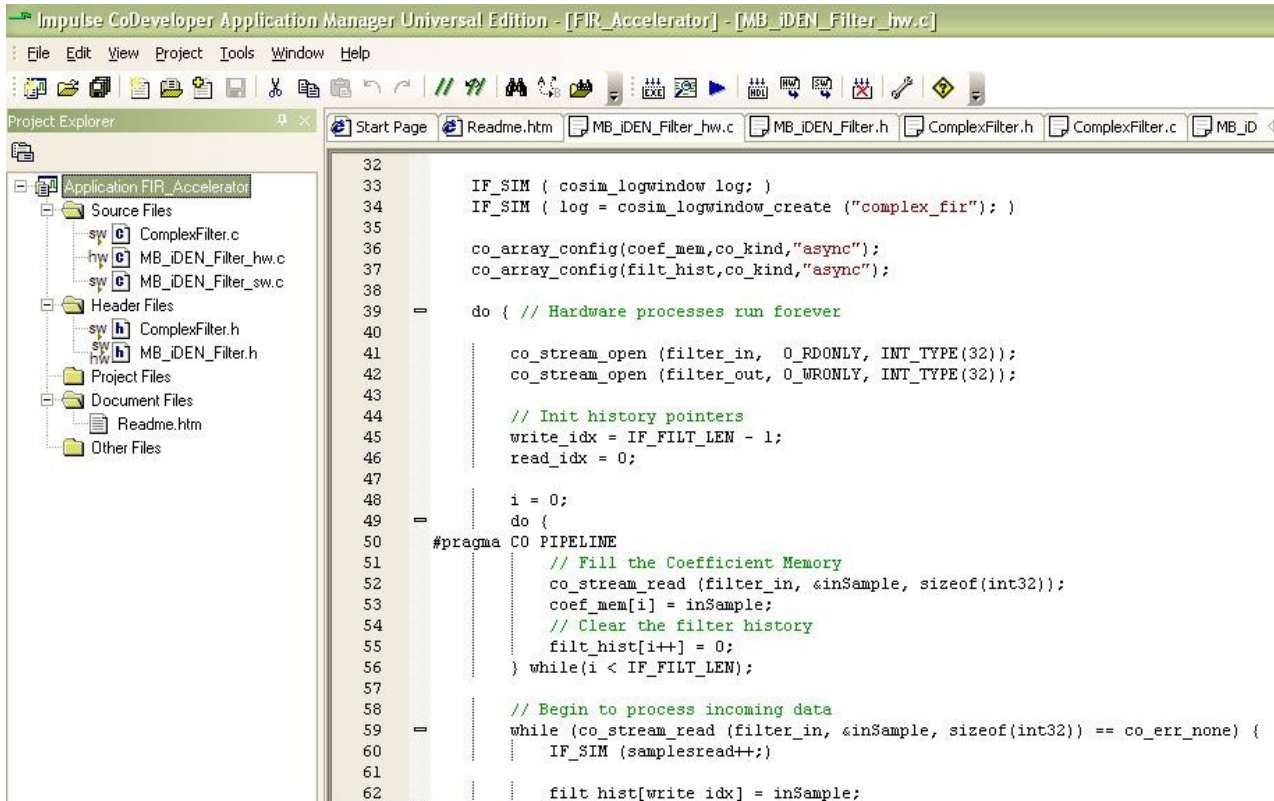
This example makes use of the Spartan-3A DSP 1800 Board, which features a Spartan-3A FPGA with a MicroBlaze soft processor. Other FPGA development boards can be targeted using the same basic steps described in the application note.

This application note assumes you are already familiar with the design flow from Impulse CoDeveloper into the Xilinx EDK tools. If you are not familiar with this design flow, please read the MicroBlaze tutorials provided with Impulse CoDeveloper, in the Help and Support section of the CoDeveloper Start Page.

This tutorial also assumes you are using the Xilinx EDK 9.2 (or later) development tools.

## Complex FIR Example in CoDeveloper

In CoDeveloper, open the ComplexFIR project as shown below. For details about this example, please refer to Impulse C Xilinx MicroBlaze Tutorial #2.



In the Project Options Dialogue, Generate Tab, choose “Xilinx BlueCat Linux MicroBlaze FSL” as the Platform Support Package (PSP).

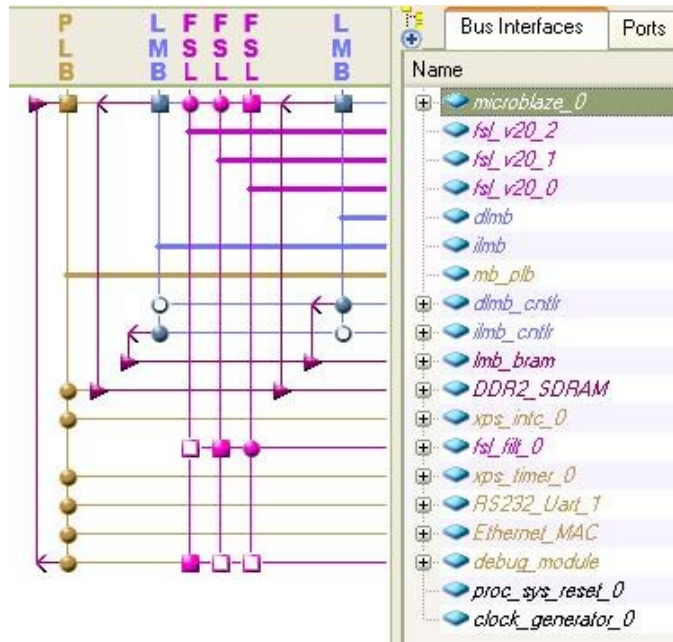


Now generate HDL and export the generated hardware and software using the “Generate HDL”, “Export Hardware” and “Export Software” toolbar buttons.

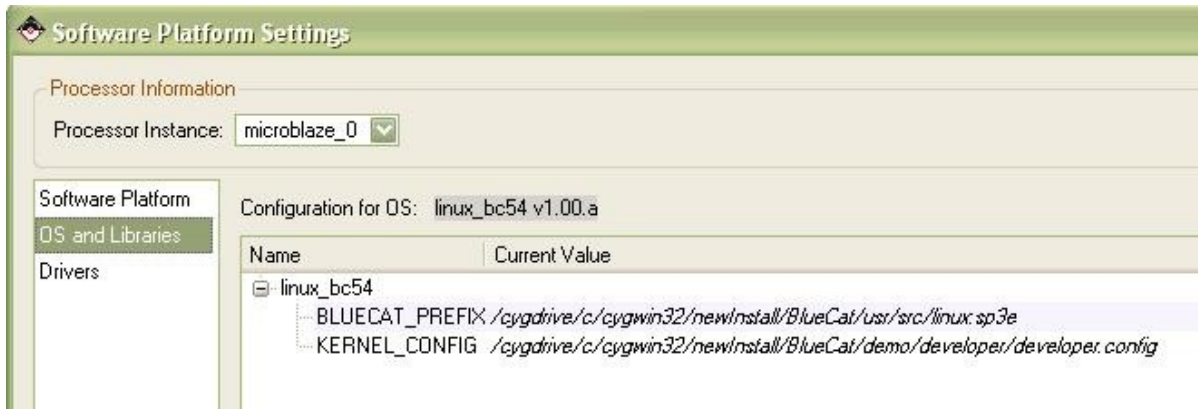
## Building Hardware in Xilinx EDK

In Xilinx EDK (Platform Studio), created a new project called ComplexFIR. Refer to Xilinx MicroBlaze Tutorial #2 for details of how to set up the MicroBlaze and its peripherals.

When using BlueCat Linux, besides the two FSLs for input and output streams, a third FSL is used to connect MicroBlaze and the debugger module XMD for fast downloading of the Linux image.



In Software Platform Settings, choose “linux\_bc54” as OS. In the OS and Libraries Tab, set the parameters as follows:



Now you must generate libraries and drivers as described in the tutorial. This action will update the configuration files for the Linux kernel.

Generate a bitstream for the design. Choose “bootloop” as the software application for BRAM initialization.



Now you can download the bitstream to the FPGA.

## Setting Up the BlueCat Linux Environment

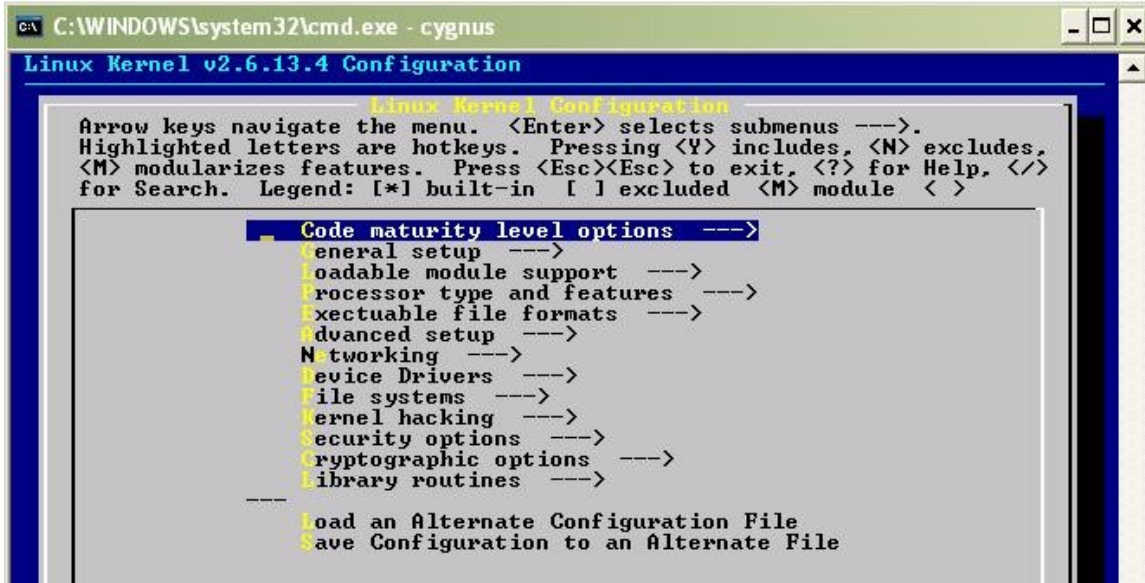
Install BlueCat Linux 5.4.2 according to the instruction provided by LynuxWorks. In the following example, the installation directory is: C:\cygwin32.

*Note: the LynuxWorks patch "5p4p2PATCH" is needed to install the FSL driver.*

The following command lines are used to set up the environment:

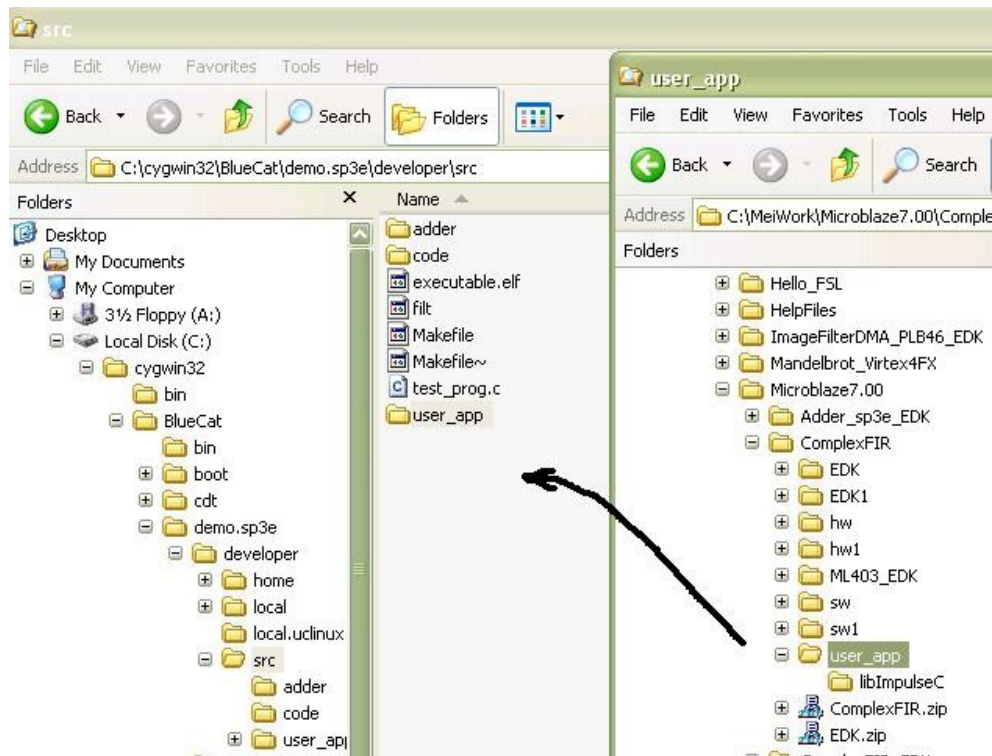
```
BlueCat:bash-3.00$ exit
exit
C:\cygwin32>cygnus
"TARGET_PATH is c:\cygwin32"
Warning: no CDRM found!
bash-3.00$ cd BlueCat/
bash-3.00$ ./SETUP.sh
bash-3.00$ ./SETUP.sh sp3e
bash-3.00$
```

Command "make menuconfig" can be used to modify the Linux kernel configuration. This is optional; we will just use the default configuration here.



## Building the Complex FIR Software into the Linux File System

Copy the software code and libraries from the Impulse C project directory to the BlueCat Linux developer source directory as follows:



Next, modify the Makefile in \$BlueCat/demo/developer directory. Add the following lines:

```
cd src/user_app/libImpulseC; make all
```

```
cd src/user_app; make all
```

Add the following line to the developer.SPEC:

```
cp ./src/user_app/user_app tmp
```

Next, build the kernel by running:

```
make clean  
make all
```

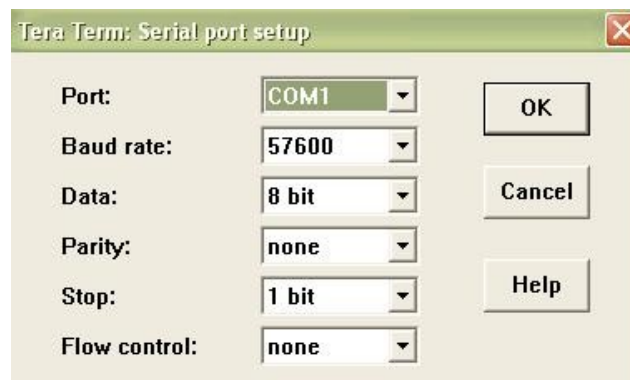
This will build the Linux kernel image with the example application. The current PSP supports up to 4 FSLs. Minor modifications to the source code can be done to accommodate more FSLs (up to 8). The following Linux commands are used to add FSLs as devices:

```
mknod /dev/fsl0 c 10 230  
mknod /dev/fsl1 c 10 231
```

The major number is 10; minor numbers can be 230 and up.

## Download the Linux Image to the Target Board

First, open the Tera Term or other terminal application. Set the serial port to 57600, 8-N-1.



Next, open the XMD from EDK. Download the kernel image “developer.kdi” to the DDR SDRAM starting at address 0x88000000. This will take a few minutes to complete. Then start the execution.

```
dow -data ../developer.kdi 0x88000000  
con 0x88000000
```

```

C:\> C:\EDK9.2i\bin\nt\bash.exe
Instruction Cache Base Address.....0x88000000
Instruction Cache High Address.....0x8fffffff
Data Cache Support.....on
Data Cache Base Address.....0x88000000
Data Cache High Address.....0x8fffffff
Exceptions Support.....on
FPU Support.....off
Hard Divider Support.....on
Hard Multiplier Support.....on - <Mul132>
Barrel Shifter Support.....on
MSR clr/set Instruction Support...on
Compare Instruction Support.....on
Number of FSL ports.....3

Connected to MDM UART Target
Connected to "mb" target. id = 0
Starting GDB server for "mb" target (id = 0) at TCP port no 1234
XMD% dow -data .././developer.kdi 0x88000000
System Reset .... DONE
Downloading Data File -- .././developer.kdi at 0x88000000

XMD% con 0x88000000
Info:Processor started. Type "stop" to stop processor

RUNNING> XMD%

```

Watch the Tera Term window for linux booting messages.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help

myhostname login: Linux version 2.6.13.4 (Mei Xu@impulse-lab-008) (gcc version 4
.1.1) #41 Tue May 13 09:48:08 PDT 2008
On node 0 totalpages: 32768
DMA zone: 32768 pages, LIFO batch:15
Normal zone: 0 pages, LIFO batch:1
HighMem zone: 0 pages, LIFO batch:1
Built 1 zonelists
Kernel command line: ramdisk_size=28472 hda=bswap hdb=bswap hdc=bswap hdd=bswap
root=101
xps_intc_1.00.a INTC at 0x81800000 mapped to 0xFDFFF000
PID hash table entries: 1024 (order: 10, 16384 bytes)
xps_timer_1.00.a TIMER at 0x83C00000 mapped to 0xFDFFE000
Console: Xilinx UART Lite
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 124032k available
Calibrating delay loop... 30.92 BogoMIPS (lpj=154624)
Mount-cache hash table entries: 512
NET: Registered protocol family 16
Registering MicroBlaze FSL FIFO driver.
3 FSL channel(s) registered.
ttyS0 at MMIO 0x84000000 (irq = 4) is a Xilinx UART Lite
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
RAMDISK driver initialized: 16 RAM disks of 28472K size 1024 blocksize
eth0: using fifo mode.
eth0: No PHY detected. Assuming a PHY at address 0.
eth0: Xilinx EMACLite #0 at 0x81000000 mapped to 0xC8020000, irq=5
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP reno registered
TCP bic registered
NET: Registered protocol family 1
RAMDISK: Compressed image found at block 2229588
Freeing BlueCat RFS memory: 4234k freed
UFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 73k freed
INIT: version 2.85 booting

```

Log in as “root”.

The ComplexFIR application is in the tmp directory.

```
cd tmp
./user_app
```

The ComplexFIR filter will execute as follows:

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

myhostname login: root
login(pam_unix)[231]: session opened for user root by (uid=0)
-- root[231]: ROOT LOGIN ON console
-bash-3.00# cd tmp
-bash-3.00# ./user_app_complexFIR

-----
Impulse C
-----
C-toFPGA Tools
for Xilinx FPGA
Platforms

-----
Complex FIR Filter Acceleration demonstration, featuring the Xilinx
MicroBlaze, Spartan-3 FPGA and Impulse C-to-FPGA tools.
Running LynuxWorks BlueCat Linux 5.4.2 on Microblaze.

=====Running the Software-Only Version=====
--> Begin filtering two slots
*****
--> Done filtering two slots
starting at 63 second, 50 milli-second
ending at 91 second, 640 milli-second
...total software execution time: 28590 milli-seconds

=====Running the Accelerated Version=====
--> Begin filtering two slots
*****
--> Done filtering two slots
starting at 91 second, 720 milli-second
ending at 92 second, 0 milli-second
...total FPGA execution time: 280 milli-seconds
-----> Visit www.ImpulseC.com to learn more!COMPLETE APPLICATION
Press Enter to continue...

-bash-3.00# █
```

## Conclusion

The execution time with hardware acceleration is 102 times faster than the software only version.

Compared to the execution on standalone OS on MicroBlaze, which runs only 163 ms, the BlueCat Linux does introduce latency in terms of execution time. But this latency is more than made up for by the hardware acceleration achieved using Impulse C.