

UG0847

HiFive Unleashed Platform User Guide

Preliminary

October 2018



Contents

1 Revision History	1
1.1 Revision 1.0	1
2 Overview	2
2.1 HiFive Unleashed Platform (MPFS-DEV-KIT)	2
3 Hardware Features	3
3.1 HiFive Unleashed Kit	3
4 System Setup and Prerequisites	6
4.1 Libero SoC PolarFire Design Suite	6
4.2 Programming FlashPro and FlashPro Express	6
4.3 Microsemi PolarFire Linux SDK for the HiFive Unleashed Expansion Board	6
4.3.1 Ubuntu	7
4.3.2 Centos	7
4.4 Firmware Versions	7
5 Board Setup	8
6 Software Installation and Configuration	12
6.1 Programming the FPGA Using FlashPro	12
6.2 Building and Loading the Linux Image	14
6.2.1 Preparing an SD Card and Programming an Image for the First Time	14
6.2.2 Rebuilding the Linux Kernel	15
6.2.3 Linux Boot and Login Credentials	16
7 FPGA Design in Libero	17
7.1 Memory Map	17
7.2 GPIO Implementation	18
8 Reference	19
8.1 Recommended Reading	19
8.2 Reference	19
9 Technical Support	20

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.0

Revision 1.0 is the first publication of this document.

2 Overview

The HiFive Unleashed Platform™ is purpose-built to emulate most of the functionality of the upcoming PolarFire SoC FPGA, which will be the industry's first RISC-V based FPGA SoC.

This guide describes the MPFS-DEV-KIT, board setup, and installation steps to get the HiFive Unleashed platform boot Linux. New IP cores can be ported on the PolarFire FPGA with the Libero SoC PolarFire Design Suite. For more details on the design suite, see section [FPGA Design in Libero \(see page 17\)](#).

2.1

HiFive Unleashed Platform (MPFS-DEV-KIT)

The HiFive Unleashed Platform consists of the SiFive's HiFive Unleashed kit and Microsemi's HiFive Unleashed Expansion kit with their respective accessories. Microsemi's HiFive Unleashed Expansion board enables users to create a Linux system running on a RISC-V core complex, with a large FPGA fabric accessible through the memory map. The expansion board is shipped with a pre-configured bitstream enabling PCIe root port functionality.

HiFive Unleashed Kit

- One SiFive's HiFive Unleashed board
- One power wall adapter 12 V
- One USB-A to micro USB-B cable

HiFive Unleashed Expansion Kit

- One Microsemi's HiFive Unleashed Expansion Board
- One USB-A to micro USB-B cable
- One 12 V, AC power adapter and cord
- One FlashPro4/FlashPro5 programming dongle
- Two Libero Platinum software licenses of duration one year (\$995 value)

3 Hardware Features

This section provides the block diagram and features of the kit hardware.

3.1 HiFive Unleashed Kit

SiFive's HiFive Unleashed development kit is based on the Freedom U540-C000 chip, the first 4+1 multi-core RISC-V Linux-capable SoC.

- SiFive Freedom U540 SoC
- 8 GB DDR4 with ECC
- Gigabit Ethernet port
- 32 MB quad SPI flash from ISSI
- MicroSD card for removable storage
- FMC connector for future expansion

Figure 1 • HiFive Unleashed Board Block Diagram

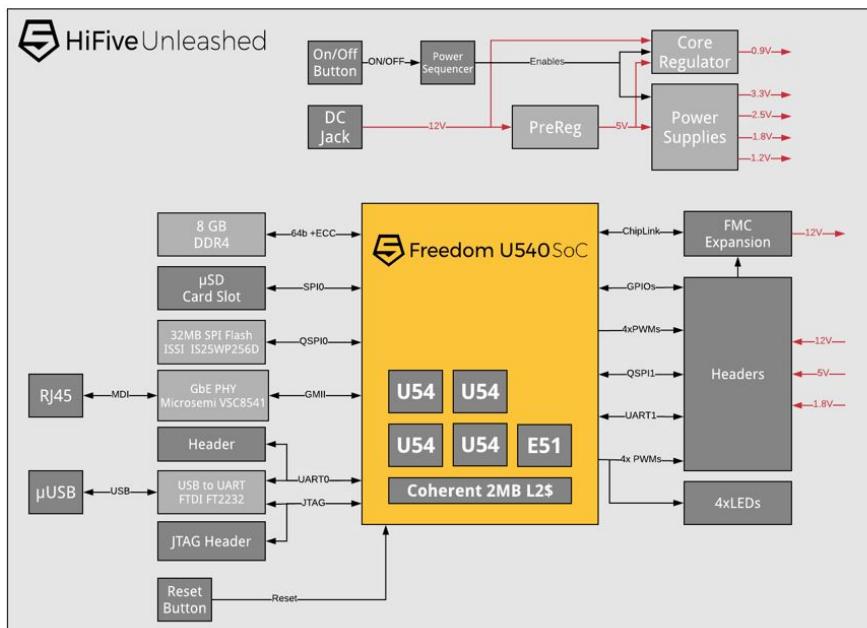
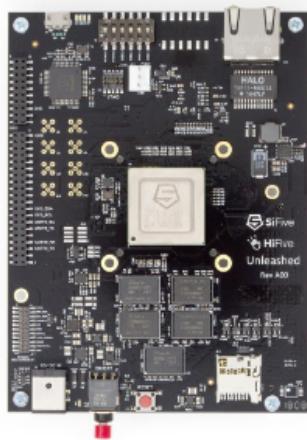


Figure 2 • HiFive Unleashed Board

The HiFive Unleashed Expansion board contains the following items.

- 300K LE PolarFire FPGA in an FCG1152 package (MPF300TS-1FCG1152EES)
- 24-lane PCIe switch
- eMMC Nand Flash, uSD card slot
- USB to Uart bridges
- 4 Gbit DDR4 x16
- PCI Express x1 card connector
- PCI Express x16 card connector with x4 lane support
- SPI Flash for FPGA remote updates, QSPI Flash connected to GPIO
- FMC connection to HFU540-4A00 kit
- SSD M.2 connector
- SATA connector

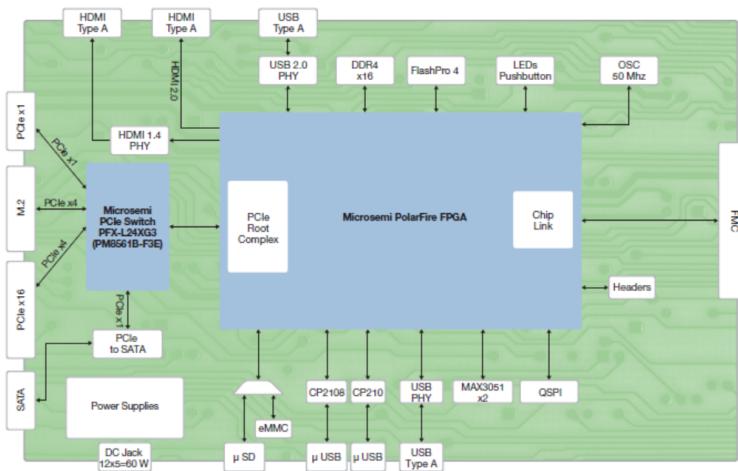
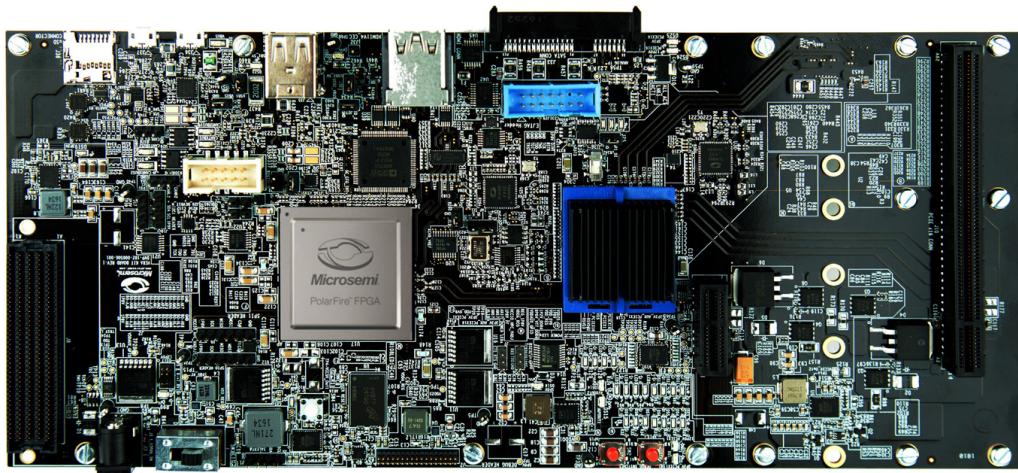
Figure 3 • HiFive Unleashed Expansion Board Block Diagram

Figure 4 • HiFive Unleashed Expansion Board



4 System Setup and Prerequisites

Download and install the following development tools in the PC in order to design, synthesize, simulate, and debug on the HiFive Unleashed Platform (MPFS-DEV-KIT).

4.1 Libero SoC PolarFire Design Suite

The Libero SoC PolarFire Design Suite offers high productivity with its comprehensive, easy-to-learn, easy-to-adopt development tools for designing with Microsemi's [PolarFire FPGA Family](#). The suite integrates industry standard Synopsys [Synplify Pro ME®](#) synthesis and Mentor Graphics [ModelSim ME](#) simulation with best-in-class constraints management, [Debug Tools](#) capabilities, and [Secure Production Programming](#) support.

Download Libero SoC PolarFire v2.2 for Windows environment [here](#).

Download Libero SoC PolarFire v2.2 for Linux environment [here](#).

Along with the purchase of the MPFS-DEV-KIT, customers are eligible for two Platinum Floating licenses for the Libero SoC PolarFire Design Suite. Write to mi-v-embeddedpartner@microchip.com with the subject "License Request <your organization name>" and include the 12-digit MAC ID of the two linux machines/PCs in your email.

4.2 Programming FlashPro and FlashPro Express

The Microsemi FlashPro programming system is a combination of Microsemi's FlashPro software and a hardware programmer. Together, they provide in-system programming (ISP) for all FPGA families. The required programming and debug software is integrated with the Libero SoC PolarFire software. This software is also available as a standalone programmer for production programming. Visit [Microsemi's FlashPro](#) page to download the standalone programmer (if needed).

Table 1 • FlashPro Software and Hardware Support

Software	Description	Hardware	Description
FP Express	Software for Windows and Linux	FlashPro5	Hardware programmer for Windows and Linux
FlashPro	Software for Windows	FlashPro4	Hardware programmer for Windows

4.3 Microsemi PolarFire Linux SDK for the HiFive Unleashed Expansion Board

The Microsemi PolarFire Linux SDK is based on the SiFive freedom-u-sdk with modifications to the device tree to support the HiFive Unleashed Expansion board. It also includes drivers for Microsemi PCIe, I²C, SPI, MMUART, and GPIO peripherals. See [Firmware Versions \(see page 7\)](#) to download the Microsemi PolarFire Linux SDK.

The build procedure follows that of the freedom-u-sdk as described in [HiFive Unleashed Getting Started Guide](#).

Before building the Linux image, the following packages must be installed depending on the Linux distribution in your machine.

4.3.1 Ubuntu

(tested on Ubuntu 16.04)

```
apt-get update

sudo apt-get install autoconf automake autotools-dev bc bison build-essential curl
flex gawk gdisk git gperf libgmp-dev libmpc-dev libmpfr-dev libncurses-dev libssl-dev
libtool patchutils python screen texinfo unzip zlib1g-dev patch device-tree-compiler
openssl-devel wget
```

4.3.2 Centos

(tested on CentOS7)

```
yum update

sudo yum install autoconf automake autotools-dev bc bison build-essential gcc-c++
curl flex gawk gdisk git gperf gmp-devel libmpc-dev libmpfr-dev ncurses-devel libssl-
dev libtool patchutils python screen texinfo unzip zlib1g-dev zlib-devel patch dtc
openssl-devel wget vim-common
```

4.4 Firmware Versions

The following table contains links to the Libero Project, .stp file, .job file, and the Linux SDK for each release.

Table 2 • Software Versions Links

Revision	Libero Project	.stp	.job	MPFS-Linux-SDK
Initial release	Libero Initial Release	.stp Initial Release	.job Initial Release	MPFS-Linux-SDK

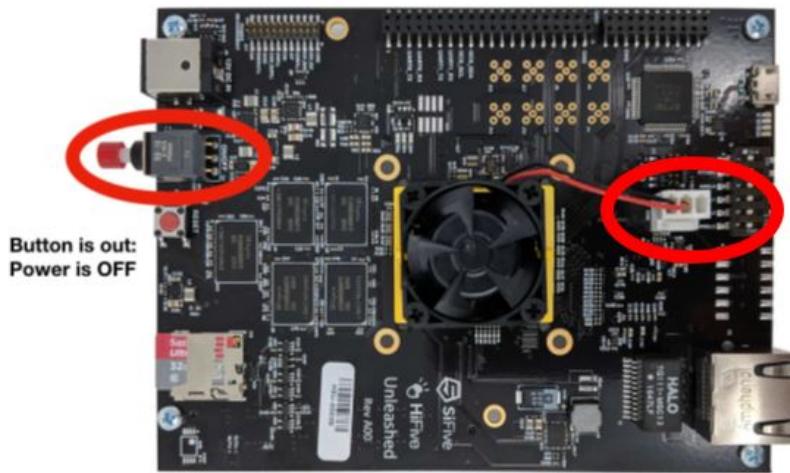
For more documentation, visit the [extranet page](#).

5 Board Setup

Follow the instructions to set up the HiFive Unleashed board.

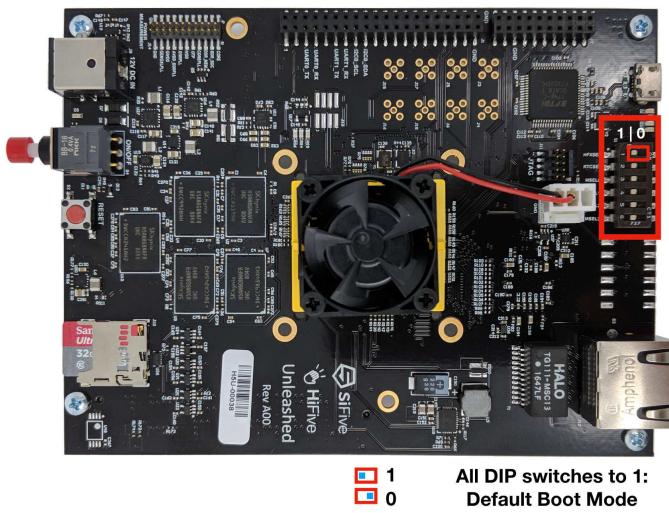
1. Switch off the power button (red button in the following figure) on the HiFive Unleashed board. Ensure the fan is plugged in.

Figure 5 • Power Button and Fan Connection

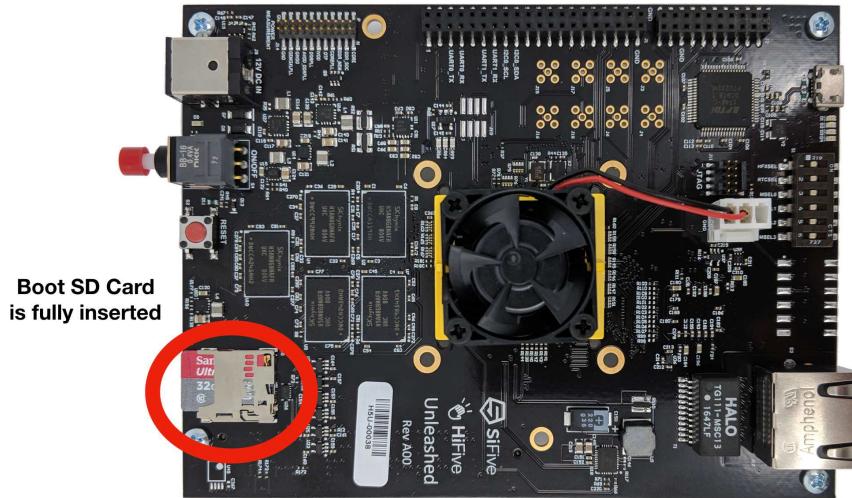


2. Set all pins in the DIP-switch block to the LEFT. The ON position=0; therefore, this sets MSEL to mode 1111. See the boot modes table in Section 4 of the [HiFive Unleashed Getting Started Guide](#) for more information on MSEL.

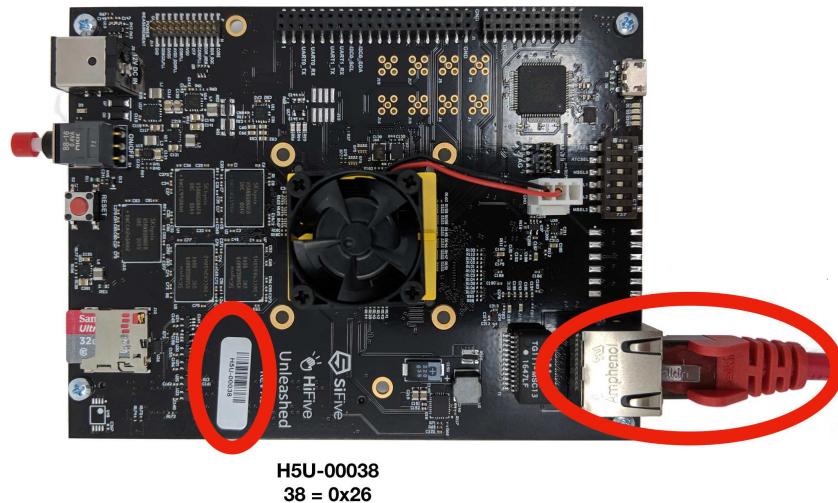
Figure 6 • DIP-Switch Setting



3. Insert an SD-card programmed with the bootloader and Linux. See the [Building the Linux Image \(see page 14\)](#) section of this document for details on creating the content of this SD card.

Figure 7 • SD Card

4. If available, connect the board to a network switch. The board will run DHCP on boot and start an ssh server. The MAC address is 70:b3:d5:92:fX:XX, where X:XX is replaced by the board number converted to hexadecimal. For example, if the board is H5U-00063, then the last digits of the MAC address are 0:3f.

Figure 8 • Ethernet

5. Connect the board through USB to a developer machine. The USB connector has two serial interfaces: the first contains the Linux console running at 115200 baud and the second provides JTAG suitable for use with OpenOCD.
6. On the HiFive Unleashed Expansion board, update the PolarFire FPGA with the FPGA bitstream provided. See section [Software Installation and Configuration \(see page 12\)](#) for steps to program the FPGA.
7. The HiFive Unleashed platform is now configured as seen in [section 7 \(FPGA Design in Libero\) \(see page 17\)](#).

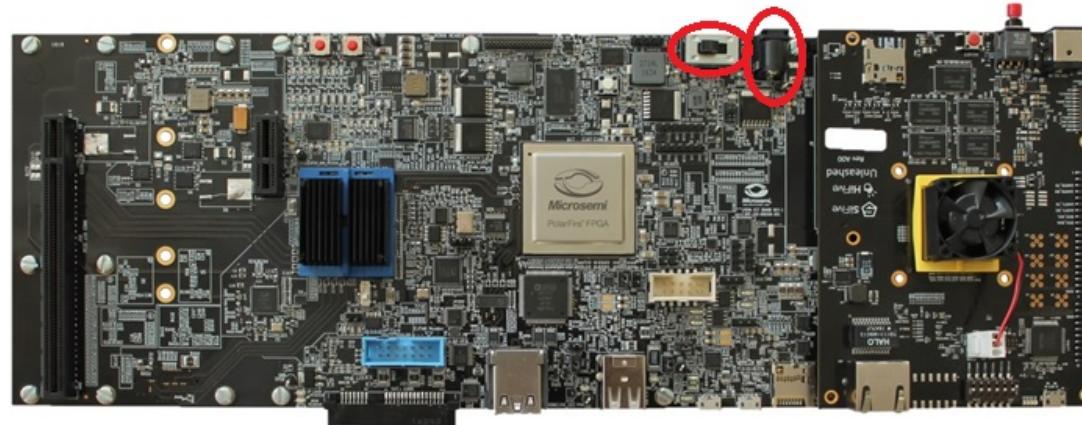
8. Plug-in the HiFive Unleashed board to the HiFive Unleashed Expansion board on the FMC connector.
Push the power button ON on the HiFive Unleashed board.
Note: Do not connect the power supply to the HiFive Unleashed board.

Figure 9 • Power Button



9. Connect the power supply to the HiFive Unleashed Expansion board in order to power the HiFive Unleashed board.
10. Slide the HiFive Unleashed Expansion board's power switch (SW3) on. Boot messages should now appear on the console in the developer's machine.

Figure 10 • Power Connections



11. After 30 seconds, an LED should begin to regularly blink a heartbeat on the HiFive Unleashed board.

Figure 11 • LED Indicators



6 Software Installation and Configuration

The following steps explain the procedure to download the FPGA bitstream onto the PolarFire FPGA.

6.1 Programming the FPGA Using FlashPro

Windows Environment

To program the PolarFire device with the .stp programming file (using FlashPro in Windows environment), perform the following steps. The link to the .stp file is given in [Firmware Versions \(see page 7\)](#).

1. Ensure that the jumper settings on the board are the same as those listed in [Jumper Settings \(see page 13\)](#).
Note: The power supply switch must be switched off while making the jumper connections.
2. Connect the power supply cable to the J3 connector on the board.
3. Connect the FlashPro4 to a PC USB port and to the connector J24 (FP4 header) of the HiFive Unleashed Expansion board.
4. Power on the board using the SW3 slide switch.
5. On the host PC, launch the FlashPro software.
6. Click **New Project** to create a new project. In the New Project window, enter a project name.
7. Click **Browse** and navigate to the location where you want to save the project.
8. Select **Single Device** as the programming mode and click **OK** to save the project.
9. Click **Configure Device**.
10. Click **Browse**, and navigate to the location where the HFU540_EXP_Bitstream_r20101.stp file is located and select the file.
11. Click **Open**. The required programming file is selected and ready to be programmed in the device.
12. Click **PROGRAM** to program the device. When the device is programmed successfully, a Run PASSED status is displayed.

See the [FlashPro User Guide](#) for more information.

Linux Environment

To program the PolarFire device with the .job programming file (using FlashPro5 programmer in Linux environment), perform the following steps. The link to the .job file can be found in [Firmware Versions \(see page 7\)](#).

1. Ensure that the jumper settings on the board are the same as those listed in [Jumper Settings \(see page 13\)](#).
Note: The power supply switch must be switched off while making the jumper connections.
2. Connect the power supply cable to the J3 connector on the board.
3. Connect the FlashPro5 to a PC USB port and to the connector J24 (FP4 header) of the HiFive Unleashed Expansion board.
4. Power on the board using the SW3 slide switch.
5. On the host PC, launch the FlashPro Express (FP Express) software.
6. From the Project menu, choose **Create Job Project** from Programming Job.
7. Click **Browse** to load the Programming Job File HFU540_EXP_Bitstream_r20102.job, and specify your FlashPro Express job project location. Click **OK** to continue.
8. Save the FlashPro Express job project.
9. Set the Programming Action in the dropdown menu to PROGRAM.
10. Click **RUN**. Detailed individual programmer and device status information appears in the Programmer List. Your programmer status (PASSED or FAILED) appears in the Programmer Status Bar.

See the [FlashPro Express User Guide](#) for more information.

Table 3 • HiFive Unleashed Expansion Board Jumper Settings

Switch	Ref	Section	Description	Definition	Default
1	SW3	Power switch	Sliding switch for 12 V input	Open: DC adaptor 12 V is not connected to the board power Close: DC adaptor 12 V is connected to board power supplies	Open
2	SW5	PCIe switch (Switchtec)	Selection of Normal/Boot Recovery mode for PCIe switch	0: Boot recovery mode 1: Normal mode	1–2: Normal mode
3	J25	SATA controller	Between SPI flash output to input of PCIe to SATA controller	Open: SPI flash o/p is not connected to PCIe to SATA controller Close: SPI flash o/p is connected to PCIe to SATA controller	Close
5	J31	USB instance 1	VBUS source selection (on-board 5 V/USB connector)	Open: VBUS sourced by USB connector (Device mode) Close: On-board 5 V to USB CONN (Host mode)	Close
6	J27	USB instance 2	Configuring USB PHY (U60) as host/device	Open: USB3340 acts as device Close: USB3340 acts as host	Close
7	J30	USB instance 2	VBUS source selection (on-board 5 V/USB connector)	Open: VBUS sourced by USB connector (Device mode) Close: On-board 5 V to USB CONN (Host mode)	Close
8	J21	SC-SPI	SC-SPI I/O configuration interface	Open: SC SPI acts as master Close: SC SPI acts as slave	Close
9	J22	JTAG	Weak pull down to JTAG TRSTB	Open: 1K pull down connected to TRSTB Close: 1K 100K pull down connected to TRSTB	Open

Switch	Ref	Section	Description	Definition	Default
11	J4	FMC bank IO voltage	FMC VADJ voltage selection	1–2: VADJ is set to 3.3 V 3–4: VADJ is set to 2.5 V 5–6: VADJ is set to 1.8 V 7–8: VADJ is set to 1.5 V 9–10: VADJ is set to 1.2 V	5–6: VADJ is set to 1.8 V

6.2

Building and Loading the Linux Image

To build and checkout the code, use the following set of commands.

```
$ tar -zxf mpfs-linux-sdk-20180906.tar.gz
$ cd mpfs-linux-sdk
$ unset RISCV
$ make all
```

This will build the system to a work/sub-directory.

Note: It can take awhile to build the first time.

The first time the build is run, it also builds the RISC-V cross compiler toolchain. The output file work/bbl.bin contains the bootloader (RISC-V pk/bbl), the Linux kernel, and the device tree blob.

6.2.1

Preparing an SD Card and Programming an Image for the First Time

Add an SD card to boot your system (16 GB or 32 GB).

If the SD card is auto-mounted, first unmount it manually. Check if your SD card is mounted and unmount it using the following commands, where XN are replaced with the SD card's specific values found from the mount command:

```
$ mount | grep sd
$ sudo umount /dev/sdXN
```

The SD card should have a GUID Partition Table (GPT) rather than Master Boot Record (MBR). It must also have the following partitions on it:

- The first partition should be for the kernel image. It will be 32 MB in size.
- The second partition is reserved for a Linux root partition. It will fill most of the free space on the card.
- The third partition should be for the HiFive First Stage Boot Loader (FSBL) binary. It will be 1 MB in size.

To automatically partition and format your SD card, in the top level of mpfs-linux-sdk, type:

```
$ sudo make DISK=/dev/path-to-sdcard-device format-boot-loader
```

This populates the SD card with the required partition types, and it copies across the Linux kernel image to the appropriate partition. If an fsbl.bin file is present in the top-level directory, it copies this across to the correct partition.

This command may fail with slower USB SD card devices but *should* succeed if repeated.

At this point, your system should be bootable using your new SD card. You can remove it from your PC and insert it into the SD card slot on the HiFive Unleashed board, and then power-on the HiFive Unleashed Expansion board.

6.2.2 Rebuilding the Linux Kernel

To rebuild your kernel, type the following from the top level of mpfs-linux-sdk:

```
$ rm work/linux/vmlinux  
$ make
```

Copy this newly built image to the SD card using:

```
$ sudo dd if=work/bbl.bin of=/dev/path-to-sdcard-device1
```

Note the '1' at the end of the SD Card device path to signify the first partition.

The source for the device tree for HiFive Unleashed Expansion board is in the conf/riscvpc.dts directory. The configuration options used for the Linux kernel are in conf/linux_defconfig.

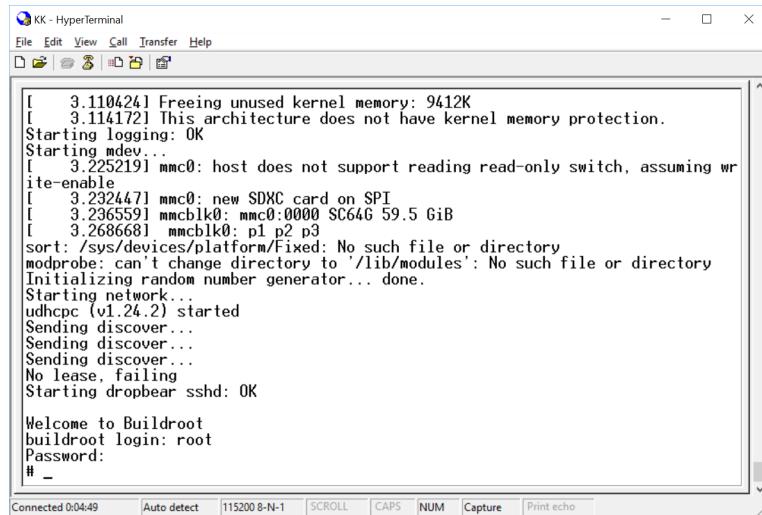
Currently, the Microsemi PolarFire Linux SDK for the HiFive Unleashed platform uses a modification to the RISC-V Bootloader startup code to pass in the device tree blob (see riscv-pk/machine/mentry.S for the modification.)

6.2.3 Linux Boot and Login Credentials

The Linux boot process can be observed by connecting a serial terminal to the USB port on the HiFive Unleashed board. Settings are 115200 baud, 8 data bits, 1 stop bit, no parity, and no flow control.

The root password is “microchip”. The console should look similar to the following figure.

Figure 12 • Console Image for Boot



KK - HyperTerminal

```
[ 3.110424] Freeing unused kernel memory: 9412K
[ 3.114172] This architecture does not have kernel memory protection.
Starting logging: OK
Starting mddev...
[ 3.225219] mmc0: host does not support reading read-only switch, assuming write-enable
[ 3.232447] mmc0: new SDXC card on SPI
[ 3.236559] mmcblk0: mmc0:0000 SC64G 59.5 GiB
[ 3.268668] mmcblk0: p1 p2 p3
sort: /sys/devices/platform/Fixed: No such file or directory
modprobe: can't change directory to '/lib/modules': No such file or directory
Initializing random number generator... done.
Starting network...
udhcpc (v1.24.2) started
Sending discover...
Sending discover...
Sending discover...
No lease, failing
Starting dropbear sshd: OK

Welcome to Buildroot
buildroot login: root
Password:
```

Connected 0:04:49 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

7 FPGA Design in Libero

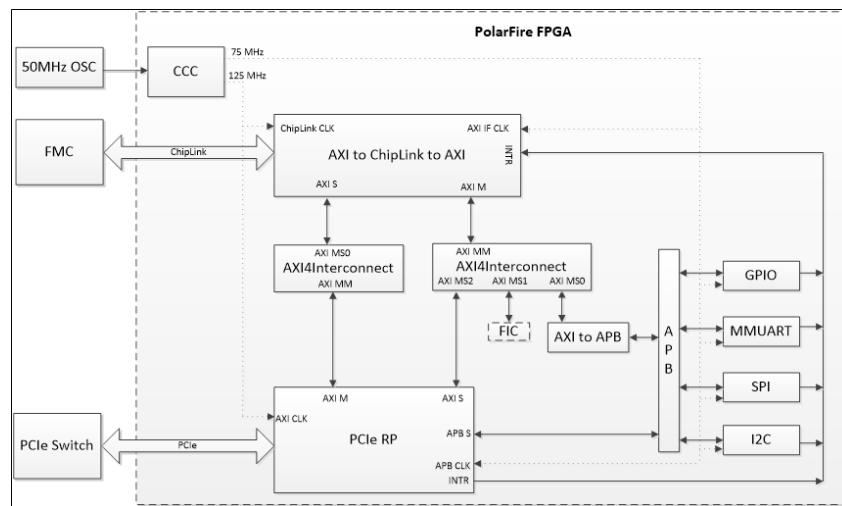
Libero SoC Design suite provides a comprehensive design flow including traditional FPGA design flow, embedded design flow, and graphical configurators. The suite provides a comprehensive development environment to build embedded solutions using hard core and soft core processors.

The FPGA design has provision to interface the PolarFire FPGA (in the HiFive Unleashed Expansion board) with the HiFive Unleashed board using the ChipLink interface. The FPGA fabric is instantiated with the ChipLink to AXI bridge, while peripherals—GPIO, MMUART, SPI, and I²C—are connected to it using the CoreAXIInterconnect, AXI to AHB, AHBLite, and CoreAPB IPs. The PolarFire PCIe AXI Slave is connected to the AXI2CL2AXI bridge through CoreAXIInterconnect. One AXI port is exposed and marked as unused and can connect to user logic.

The ChipLink interface uses 125 MHz clock and AXI interface uses 75 MHz clock.

The high-level block diagram for the Libero project implemented on the PolarFire FPGA is as seen in the following figure.

Figure 13 • Libero Project Block Diagram



More IPs are being ported on the HiFive Unleashed Platform and will be made available on request.

7.1 Memory Map

The IPs ported on the PolarFire FPGA are accessible from the RISC-V U540 memory map as listed in the following table. The interrupt number 42 is used for all the peripherals.

Table 4 • Memory Map

Peripherals	Mem Requirement	Start Address	End Address	Mem Allocation	Comments
PCIe					
PCIe Config Space	256 MB	0x2030000000	0x203fffffff	256 MB	
PCIe Mem Space 32	512 MB	0x0040000000	0x005fffffff	512 MB	
PCIe Mem Space 64	2 GB	0x2080000000	0x20fffffff	2048 MB	

Peripherals	Mem Requirement	Start Address	End Address	Mem Allocation	Comments
PCIe APB Space	1 MB	0x2000000000	0x2000ffff	1 MB	
Peripherals					
I2C_0	4 KB	0x2000100000	0x2000100fff	4095 KB	I2C0 header
Reserved	4 KB	0x2000101000	0x2000101fff	4095 KB	Reserved
GPIO	4 KB	0x2000103000	0x2000103fff	4095 KB	GPIO Implementation (see page 18)
MMUART_0	4 KB	0x2000104000	0x2000104fff	4095 KB	J36
Reserved	4 KB	0x2000105000	0x2000105fff	4095 KB	Reserved
Reserved	4 KB	0x2000106000	0x2000106fff	4095 KB	Reserved
SPI 0	4 KB	0x2000107000	0x2000107fff	4095 KB	SPI0 header
Reserved	4 KB	0x2000108000	0x2000108fff	4095 KB	Reserved
FIC (AXI)	256 MB	0x2010000000	0x201fffffff	256 MB	To connect with user logic
free space		0x2020000000	0x202fffffff	256 MB	

7.2

GPIO Implementation

The GPIO implemented in the design is pinned out as a starting point for your custom design implementation. The details of the GPIO pinout is listed in the following table.

Table 5 • GPIO Pinout

GPIO	Function
0	led4
1	led5
2	J2-pin13
3	J2-pin14
4	sw9
5	sw10
6	J2-pin9
7	USB1 reset

8 Reference

Visit the following links for further reference reading materials.

8.1 Recommended Reading

- [RISC-V User-level ISA Specification](#)
- [RISC-V Draft Privileged ISA Specification](#)
- [SiFive FU540-C000 User Manual](#)
- [TU0844 Libero SoC PolarFire v2.2 Design Flow Tutorial](#)
- [HiFive Unleashed Getting Started Guide](#)

8.2 Reference

- [PolarFire FPGA Documentation](#)
- [Libero SoC PolarFire Documentation](#)
- [FlashPro User Guide for PolarFire](#)
- [FlashPro Express User Guide for PolarFire](#)

9 Technical Support

For technical queries, email mi-v-embeddedpartner@microchip.com. Microsemi's technical support team will create a ticket, address the query, and track it to completion.

**Microsemi Headquarters**

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

© 2018 Microsemi. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions; setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200847