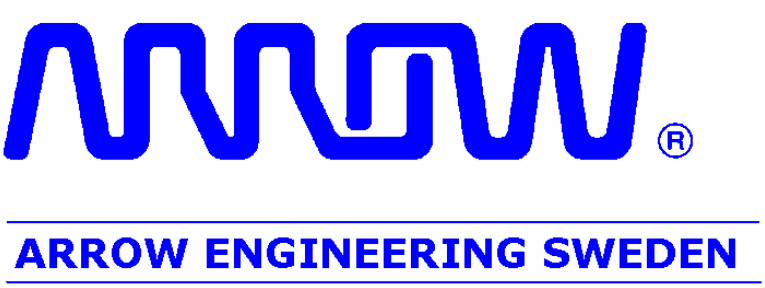
Bitfire Development Kit
FPGA Developers Guide
Version 1.1



This page is intentionally left blank.

Revision History		
Version	**Date**	**Updates**
1.1	051019	Minor updates
1.0	050531	Document public release.

Arrow Engineering Sweden, Malmö, Sweden

1 Overview

The Bitfire development kit comes with a large low-cost Cyclone® FPGA from Altera® (EP1C6T144C6). This makes the kit very flexible and opens possibilities for a wide variety of applications. A large number of the FPGA's I/O pins are available on the boards contact headers; see the Bitfire Hardware Reference Guide for details. This guide explains how to install and setup the required tools for FPGA development. Further it explains how to compile the reference design and configure the FPGA or its PROM. It also contains a brief tour of the Bitfire FPGA reference design.

2 Getting Started

In order to convert design entry to FPGA bit files you need tools to synthesize the RTL files and place-and-route (or fit) the netlists. In case of Altera® design flow; this is all done in a single tool called Quartus®. For Cyclone® designs, the free version called Quartus®II Web Edition can be used.

Another tool that is frequently used when creating and debugging RTL files is a simulator. When the reference design for Bitfire was written the popular simulator ModelSim® was used. There is however no free version of ModelSim® available. With the reference design, a ModelSim® test-bench is supplied, which you can benefit from if you decide to get a ModelSim® license. Contact your local Arrow sales representative for details on licenses.

2.1 Installing and setting up Quartus®II

In the Bitfire CDs `tools` directory, run the file called
quartusii_50_sp1_web_edition_single.exe

Once installed you will need to generate a license for Quartus®. In order to do this go to the website: http://www.altera.com/licensing/ and enter **the Quartus®II Web Edition Software** section and follow the instructions.

2.2 Installing the on-board ByteBlasterII

The Bitfire kit comes with an on-board ByteBlaster that is used for communication between the FPGA on PC. It's used mainly for programming, but can also be used for debugging and monitoring.

To install the driver do the following (for Windows© 2000 and XP);

- Click on the Start menu, and click on Control Panel.
- Click the **Switch to Classic View** link if necessary.
- Double-click the **Add Hardware** icon to start the Add Hardware Wizard and click **Next** to continue.
- Select **Yes, I have already connected the hardware** and click **Next**.
- Select **Add a new hardware device** in the Installed hardware list, and click **Next** to continue.
- Select **Install the hardware that I manually select from a list (Advanced)** and click **Next** to continue.
- Select **Sound, video and game controllers**, and click **Next** to continue.
- Select **Have Disk ....**
- Browse to the win2000.inf file in the \drivers\win2000 directory of your Quartus®II software installation and click **OK**.
- Click **Continue Anyway** when the Software Installation warning appears.
- Select Altera® ByteBlaster and click **Next** to continue.
- Click **Next** to install the driver.
- Click **Continue Anyway** when the Hardware Installation warning appears.
- Click **Finish** in the Completing the Add/Remove Hardware Wizard window.
- Reboot the computer.

And to setup programming hardware in the Quartus®II software;

- Start the Quartus® II software.
- Choose **Programmer** from the Tools menu. The programmer window will open.
- Click the **Hardware** button to open the Hardware Setup window.
    - The selected programming hardware is identified as Currently Selected Hardware.
    - Programming hardware that is already set up appears in the Available hardware items window.
- Click the **Add Hardware** button to open the Add Hardware window if the programming hardware you would like to use is not listed in the Available hardware items window.
    - Select the appropriate programming cable or programming hardware from the Hardware Type list.
    - Select the appropriate port and baud rate if necessary.
    - Click **OK**.

- Select the programming hardware you would like to use by clicking on it in the Available hardware items window then clicking the **Select Hardware** button. Your choice will be listed as the Currently Selected Hardware.
- Click **Close**.

2.3 Installing the Reference Design

In the Bitfire CDs `source` directory, unzip the file called **bitfire_fpga_src.zip**
Suggested install location is `C:\Bitfire\fpga`

2.4 Installing FPGA tools under Linux

Quartus®II Web Edition is not available for Linux; however Linux is supported by the full version of Quartus®. In order to get this license contact your local Arrow sales representative.

For installing the onboard Byteblaster under Linux, please consult the Altera® documentation.

To get the reference design unzip the file **bitfire_fpga_src.zip** located in the Bitfire CDs `source` directory.

3 Compiling the Reference Design

Open Quartus®II and go to the menu **File->Open Project** select the file `fpga/quartus/cpuledctrl/cpuledctrl.qpf`

This will open the reference design project. To compile go to the menu **Processing->Start Compilation** and wait for it to finish.

4 Configuring the FPGA

Once you have compiled your design successfully you should have a .sof and .pof file in your project directory. The sof (SRAM object file) is used for programming the FPGA's SRAM cells directly (contents will be lost a next reset). The pof (Programmer object file) is used to program the EPCS1 PROM device (in active serial mode).

4.1 With a SOF file

Make sure the Bitfire is setup for **FPGA Config** and connected to your PC correctly; see Bitfire Hardware Reference Guide for details.

Open Quartus® and go to the menu **Tools->Programmer** to open the programmer. Make sure the ByteBlasterII is selected, see 2.2. Press the **Add File** button and select a sof file, for example `fgpa/quartus/cpuledctrl/cpuledctrl_top.sof`

Press the **Start** button and wait for it to finish.

4.2 Configuring the EPCS1 PROM device

4.2.1 In Active Serial Mode

In this case you will put the configuration in the PROM device (in active serial mode). This configuration will automatically be loaded into the FPGA at reset.

Make sure the Bitfire is setup for **FPGA Config** and connected to your PC correctly; see Bitfire Hardware Reference Guide for details.

Open Quartus® and go to the menu **Tools->Programmer** to open the programmer. Make sure the ByteBlasterII is selected, see 2.2. In the mode selector, select **Active Serial Programming**. Press the **Add File** button and select a pof file, for example `fgpa/quartus/cpuledctrl/cpuledctrl_top.pof`

Press the **Start** button and wait for it to finish.

4.2.2 In JTAG Mode

It is also possible to program the PROM via the FPGA's JTAG port (the EPCS1 PROM doesn't have its own JTAG port). In order to do this you have to create a JIC (JTAG Indirect Configuration File) file.

- Go to the menu **File->Convert Programming Files**
- Change programming file type to **JTAG Indirect Configuration File (jic)**
- Change the configuration device to **EPCS1**
- Alter the filename if you wish
- Highlight the **Flash Loader** line in the input files list and click the **Add Device** button
- Select the Cyclone, EP1C6T144 device in the list and press **OK**
- Highlight the **SOF Data** line and press **Add File** button
- Select the SOF file you want to convert and press **OK**
- Highlight the new line with the name of SOF file you just added and click **Properties**
- Tick the Compression checkbox and click **OK**
- Now press the **OK** button in the Convert Programming Files window to create the JIC file

Make sure the Bitfire is setup for **FPGA JTAG** and connected to your PC correctly; see Bitfire Hardware Reference Guide for details.

Open Quartus® and go to the menu **Tools->Programmer** to open the programmer. Make sure the ByteblasterII is selected, see 2.2. Press the **Add File** button and select a jic file, for example `fgpa/quartus/cpuledctrl/output_file.jic`

Press the **Start** button and wait for it to finish.

5 The Reference Design

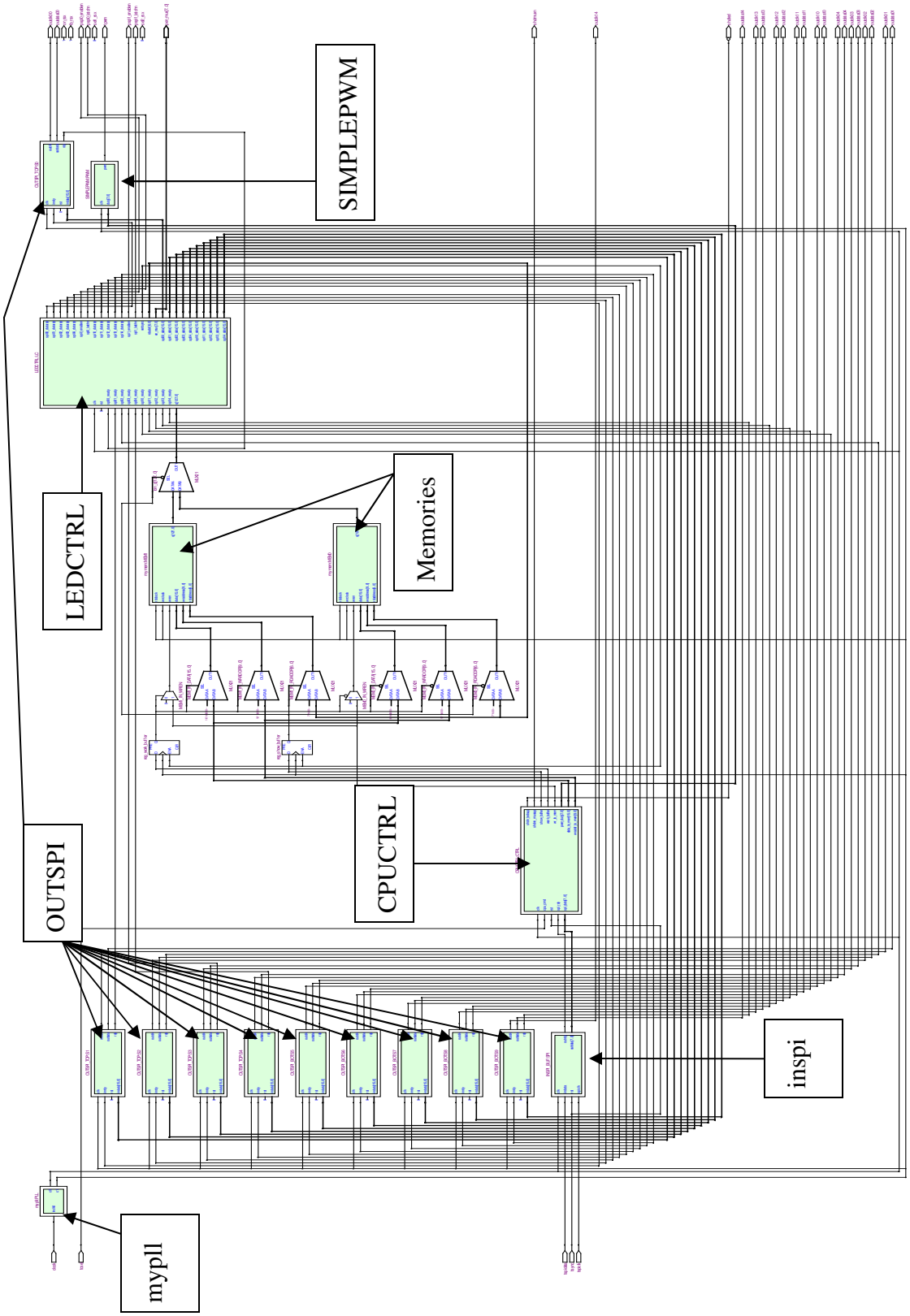
Before diving into the details on the FPGA reference design you should read the Bitfire Hardware Reference Guide to understand how the board and specifically how the LED display works. The purpose of the reference design is to drive the display and control some of the pins used for controlling the UART operation and board power management.

The Reference Design is written in 100% VHDL and can be found in **`fpga/src`**. The Quartus® project that is used for compiling the design can be found in **`fpga/quartus/cpuledctrl`**. Finally, the ModelSim® test-bench is located in **`fpga/modelsim/cpuledctrl`**.

The Bitfire FPGA reference design consists of 9 major blocks:

- INSPI — Receives SPI data/commands from CPU and gives them to the CPUCTRL.
- CPUCTRL — Converts data to memory writes and commands to different pin selections.
- mymem — 2 instances (double buffering) of the altmem megafunction. These memories are double ported (one read and one write port) and hold the display memory.
- mypll — PLL instance of the altpll megafunction. Generates 2 clocks internally in the FPGA. On at double oscillator frequency (used by the OUTSPI and memory blocks) and one at o scillator frequency that are used by all other blocks.
- LEDCTRL — Reads data from one of the memory blocks and drives PWMs on the OUTSPI channels.
- OUTSPI_TOP — 5 instances that drives the pixels of the top 5 8x8 LED matrixes.
- OUTSPI_BOT — 5 instances that drives the pixels of the bottom 5 8x8 LED matrixes.
- SIMPLEPWM — Controls the global dimmer for all LED modules.
- Top — Top entity that instantiates all blocks above. Top also defines the memory muxes that controls the double buffering feature.

See the following picture for an overview of how there blocks is connected.


SIMPLEPWM
LEDCTRL
Memories
CPUCTRL
OUTSPI
inspi
mypll

6 Creating your own design

The reference design takes about 3200 logical elements and 20400 bit of memory in the FPGA, so it's roughly half-full. This makes it possible for you to add a lot of your own logic on top of the reference design (if you want to keep LED support). If you don't need the LED matrix at all, you can of course begin from scratch with your own design.

The simplest way to add your own logic is to edit the **`fpga/src/cpuledctrl_top.vhd`** file and instantiate your own logic.

If you start from scratch you will benefit from the pin assignment done in the reference design. These assignments can be found in the file
`fpga/quartus/cpuledctrl/cpuledctrl_top.qsf`

Be sure to read the Bitfire Hardware Reference Guide carefully to get the UART control and power management pins setup correctly.

