DE1-SoC Starting Guide

Written by Electronic Technology Department (University of Vigo) 2015

Authors:
Roberto Fernández Molanes (robertofem@gmail.com)
Filipe Salgado (filipe.salgado@gmail.com)

Index of contents:

1. Introduction

DE1-SoC, depicted in Fig. 1.1 , is a Development Kit designed by Terasic around the Altera System-on-Chip (SoC) FPGA Cyclone V. Cyclone V SoC combines a dual-core Cortex A-9 embedded processor with programmable logic (as a regular FPGA). It includes ARM-based hard processor system (HPS) consisting of processor, peripherals and memory interfaces tied seamlessly with the FPGA fabric using a high-bandwidth interconnect. The DE1-SoC board is equipped with high-speed DDR3 memory, video and audio capabilities, Ethernet networking, OTG USB and much more that promise many applications.

SoC chips allow the development of complete systems (motor drive, industrial PLC CPU, etc.) in a single chip since it includes the advantages of a hard-core, like high clock speed and floating point calculation, with the flexibility of FPGA reconfigurable fabric, which permits critical-timing tasks to be executed in hardware, the implementation of the peripherals strictly required by the application (efficient usage of silicon) and reconfigurability. HPS can be programmed directly in C (bare-metal) or through an application running over an operating system.

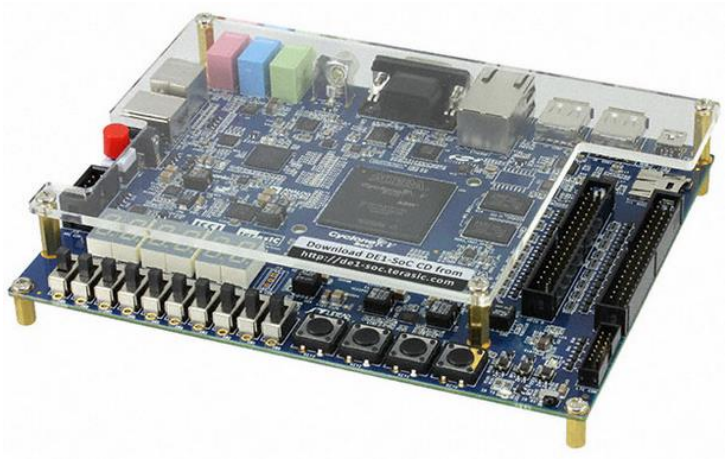

Fig. 1.1. DE-1 SoC board.

This starting guide is intended to be an introductory reference for a student to start with DE1-SoC board.

2. Gathering documentation for DE1-SoC

In this chapter we present main documentation about the DE1-SoC board that a student should manage:

- **DE1-SoC board information**. Specific information regarding DE1-SoC board can be found in Terasic´s website -> SoC Platform -> Cyclone -> DE1-SoC [1]. This website contains:
    - **Documents**: User Manual and Learning Roadmap. These documents are included in CD-ROM so it is not needed that they are downloaded from here.
    - **BSP for Altera SDK OpenCL**: Tools to develop heterogeneous programs (to compute sharing tasks between hardware and software) using OpenCL programming language.
    - **CD-ROM**: It contains all the documentation that Terasic provides for DE1-SoC. Inside CD-ROM we find:
        - Datasheet folder: Datasheet of the DE1-SoC components
        - Demonstrations folder: Code of examples, most of them explained in User Manual PDF. One important example, not explained in User Manual, is Golden Hardware Reference Design (GHRD). Golden System Reference Design (GSRD) is a complete example explained in [3] for Cyclone V SoC Development Board. It is a reference design and modifying this example is a good way to create a custom solution. GHRD is the hardware part of GSRD for DE1-SoC board and permits to adapt all explanations in [3] (from Cyclone V SoC Development Board) to DE1-SoC board.
        - Schematic folder: Schematics of DE1-SoC board.
        - Tools folder: SystemBuilder (permits generating a Quartus II empty project with pin assignments and device selection for DE1-SoC) and rbf_generate_file (generates a file to save FPGA configuration in EQCP device).
        - User Manual folder. Contains:
            - Learning Road Map [8]: Proposed itinerary to learn DE1-SoC and SoC development flow.
            - User Manual [7]: Explanation of the hardware in DE1-SoC board and explanation of most of the Demonstrations folder projects.
            - Tutorials:
                - Getting Started Guide [9]: Installation of tools, powering-up the board and testing Linux.
                - First FPGA project [10]: example project using only the FPGA part of the Cyclone V device.
                - First HPS project [11]: example project using only the HPS subsystem to compile and run an application over Linux.

        - First HPS-FPGA project [12]: complete example using FPGA and HPS. A program running over linux accesses to a FPGA peripheral.
    - **Linux BSP MicroSD Card Image**: Some images of Operating Systems (OSs) provided by Terasic.
- **Cyclone V documentation**: Website containing all documentation updated for Cyclone V is the Altera Support Website for Cyclone V devices [2]. Most important documents are:
    - **Cyclone V Device Handbook** [13]:
        - Vol.1: Device Interfaces and Integration: it includes, among other information, explanation of logic array blocks, PLLs and DSP blocks.
        - Vol.2: Transceivers: it includes information about the configurable fast transceivers in Cyclone V devices.
        - Vol.3: Hard Processor System (HPS): it includes description of the HPS blocks including Cortex A9 Microprocessor system, HPS peripherals (USB and Ethernet Controllers, Timers, On-Chip Memory, etc.), SDRAM controller for external memory and booting process after power on.
    - **Cyclone V Datasheet** : Includes electrical and timing characteristics of Cyclone V devices.
- **Programming resources**. There are three ways of configuring this kind of devices:
    - **Using Quartus II (for hardware) and Altera SoC EDS (for software)**. This is the programming flow used in this document. Most important documents to program Cyclone devices are:
        - Quartus II Handbook [15] : Quartus II is the tool to design, simulate, compile and load hardware for Altera FPGAs, CPLDs and SoCs. This handbook is a complete documentation of this software, including Qsys (tool to easily develop hardware connecting blocks graphically), compilation and simulation options, etc.
        - Embedded Peripheral IP User Guide: documentation of the IP (Intellectual Property) cores available in Qsys.
        - Making Qsys components [17]: an explanation about how Qsys components are created is exposed in [15]. However this document is interesting because it contains a detailed explanation from HDL source code to the component, making emphasis in relating signals in the component with signals in the predefined Avalon Interfaces (Avalon is the bus specification to connect different components in Qsys).
        - Avalon Specification [18]: complete specification of Avalon Interfaces.
        - Altera SoC Embedded Design Suite (EDS) User Guide [19]: It explains how to install and license Altera SoC EDS. Altera SoC EDS includes DS-5 Altera Edition software (program to compile and debug ARM devices with some modifications from Altera to adapt it to their devices), preloader and uboot generator (software running before OS), devicetree generator (files describing hardware inside Cyclone V device) and other useful scripts. This document also explains some

interesting examples about programming and debugging using DS-5 Altera Edition tool.
        - Howard Mao Website [6]: Website of a PhD student including tutorials and projects using Arrow SoCKit board.
        - RocketBoards website[3]: Very important website containing documentation, projects and examples for some Cyclone V SoC boards, including DE1-SoC from Terasic. Golden System Reference Design (GSRD) is explained for Cyclone V SoC Development Board. Hardware and OS SD image is explained here. Using GHRD including in DE1-SoC CDROM this explanations are ported to DE1-SoC board.
    - **DSP Builder for Matlab Flow**: Algorithms are programmed using only Matlab, without knowledge of the device architecture. Using Simulink Altera Blocks and regular Simulink blocks it is possible generate a hardware block for Qsys. Later, Quartus II compiles the created Qsys system. Using C coder from Matlab it is possible to generate code for the ARM processors in HPS from Matlab language sources. This way, having only Matlab programming skills it is possible to program ARM processor and configure FPGA part.
    - **BSP (Board Support Package) for Altera SDK OpenCL**: OpenCL language, born to do heterogeneous computing with x86 processor and GPU connected by PCI bus, can be now used to program Cyclone V devices. OpenCL code is traduced into C code for ARM processor and to HDL for the FPGA part. This kind of programming is not treated in this document.
- **Other resources:**
    - More documentation in PDF format can be found at Altera´s documentation web-site [4].
    - Training courses, some with video-tutorials, are available at Altera Training Website [5]. Most of courses are free.

3. Available OS

No OS: The board can be programmed withot OS, it is called baremetal application. Instad of using OS calls your code uses the Hardware Libraries from Altera to access hardware. For more info check [19].

Terasic provides four OS images, available in [1]:

- Linux Console: In fact is a linux distribution compiled using Yocto tools. It is the most basic available Linux distribution. You work through serial console. To install some package or driver you should add it to the OS and recompile it. Although the first time you compile a Yocto-based system takes several hours (4 to 10 hours depending on the machine), recompilation takes lttle time because most files are not recompiled.
- Linux Console with Framebuffer: It is a linux kernel with a Framebuffer console. Framebuffer is an improved standar linux console. Some features of the framebuffer is hardware and driver independence and colorful graphics that can even display video.
- Linux LXDE desktop: a full desktop distribution with LXDE as desktop environment and package manager (so software is easily installed into the system).
- Linux UBUNTU desktop: porting of the famous UBUNTU distribution to Cyclone V. The exact name of the distribution is Xillinux (by Xillibus). Xillinux is UBUNTU plus some drivers to control an FPGA peripheral designed by Xillibus. This peripheral contains some streaming 1 bit ports in the FPGA so any other peripheral in the FPGA can easily communicate to CPU part through it. To read/write from/to a port in the xillibus peripheral from an application you open a file (like open a serial port) and start listening or writing information. Using this Xillibus solution it is very easy to pass information between a peripheral in the FPGA and an application. However you lose control because of the processing done in Xillibus driver and it is slow because you are using serial communication while having 32, 64 and 128 bit bridges available between the processor and the FPGA.

GSRD distribution:

- Angstrom: it is a distribution used in the Golden System Reference Design (GSRD) from Altera. GSRD is a basic example from where a person can start. Adding hardware to the GSRD is a good way to develop any custom system because you start from a system already working and all the configuration for the processor (very complex) is already done. Angstrom is a famous distribution (used in boards like Beagleboard). It contains a package manager so adding software to the OS is easy. However repositories for Angstrom apps in Cyclone V are not ready at this moment a the way to install something is compile it and manually add it with the package manager (opkg) or with driver/module tools (insmod). It has no desktop environment and board communication is done thanks to the serial port or internet (SSH).

4. Starting with DE1-SoC

4.1. Mandatory first steps

To start with DE1-SoC board, authors think that following steps are mandatory at the beginning of any project using DE1-SoC board:

1. Read User Manual [7], Chapters 1 to 4 and 8. General explanation of board resources and its possibilities.
2. Following learning road [8] map proposed tutorials:
    a. Do Getting Started [9] tutorial. In this step explanation software installation is explained
    b. Do My First FPGA tutorial [10]
    c. Do My First HPS tutorial [11]
    d. Do My First FPGA-to-HPS tutorial[12]

4.2. Other optional steps

If you want to go deeper you can check example project chapters in the User Manual:

1. Read available example projects in User Manual [7], Chapters 5 to 7. Most important example here is DE1-SoC Control Panel. Explanations here show how to program an Graphical Application running on DE1-SoC.

If you want to **create a device in the FPGA and use it from an application on top of OS** (very common situation) you have few options. As it was said before you can use Quartus II and Altera EDS flow, DSP builder for Matlab flow and Altera SDK for OpenCL. Using Quartus II and Altera EDS flow you will have more control of data flow and systems are going to be much more efficient in terms of resource usage, timing and energy consumption. However it will take a longer time to develop the software. To communicate FPGA peripheral and an app running of top of OS you can:

1. Using mmap(): mmap makes a tunnel through the virtual memory management of the OS and gives you access to the physical memory so you can directly use hardware addresses to access peripherals. This way you can access from an application to the hardware registers of the peripheral in the FPGA. However if application some wait for the peripheral to finish some task, the application must do polling on some peripheral register. To use interrupts next method should be used. This method is used in My First FPGA-to-HPS tutorial[12]. In Qsys you can edit the physical address space occupied by your FPGA peripheral inside the HPS-FPGA bridge you are using. HPS-FPGA bridges address space is located in [13].
2. Writing a driver for the FPGA peripheral. You can write and add a driver into the OS so you can use interrupts and all the OS features to stop the application and wait for the peripheral to interrupt the CPU. This method is more difficult than the previous one. [6] explains a complete example about writing a driver. [20] is a complete reference book about writing drivers for linux. Workshop 3 in [3] shows how to design drivers for

Linux. Author has done modules for Angstrom and Workshop 3 has been of little usefulness because it explains procedures for a different OS. However it is always good to take a look to it. After writing a driver it should be added to the system either while OS is being compiled (process that totally depends on the OS distribution and compilation tools) or after compilation (using insmod). IN Internet there is a lot of information on this process.

When booting OS it reads the available hardware and drivers it should load from DTB (Device Tree Blob) files, data structures defining the hardware of the system. In this DTB, among other things, the driver OS should load for this piece of hardware is specified. So writing DTBs is needed to completely integrate a driver in the OS.

Using insmod to add a driver into the OS is a problem because every time you boot the board it should be added again. You can write a script to automatically load it after booting. However the best way to proceed is to include the driver in the OS, recompile the kernel and edit a DTB in order the driver to be automatically loaded if your peripheral is included in this hardware version.

5. References

Websites:

[1] DE1-SoC Resources: Terasic Website / SoC Platform / Cyclone/ DE1-SoC: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=836&PartNo=4

[2] Altera Website / Products / FPGA / Cyclone V Series / Support: https://www.altera.com/products/fpga/cyclone-series/cyclone-v/support.html

[3] Rocket Boards Website: http://www.rocketboards.org/

[4] Main Altera Documentation Website: https://www.altera.com/support/literature/lit-index.html

[5] Altera Training Courses: https://www.altera.com/support/training/catalog.html

[6] Howard Mao Website: https://zhehaomao.com

Documents:

[7] DE1-SoC User Manual.pdf (in DE1-SoC CD-ROM/UserManual)

[8] learning_roadmap.pdf (in DE1-SoC CD-ROM/UserManual)

[9] DE1-SoC_Getting_Started_Guide.pdf (in DE1-SoC CD-ROM/UserManual)

[10] My_First_Fpga.pdf (in DE1-SoC CD-ROM/UserManual)

[11] My_First_HPS.pdf (in DE1-SoC CD-ROM/UserManual)

[12] My_First_HPS-Fpga.pdf (in DE1-SoC CD-ROM/UserManual)

[13] Cyclone V Device Handbook: (Altera Website / Products / FPGAs / Cyclone Series / Cyclone V)

[14] Cyclone V Datasheet (Altera Website / Products / FPGAs / Cyclone Series / Cyclone V)

[15] Quartus II Handbook: cyclone5_handbook.pdf (Update d version in Altera Website / Quartus II)

[16] Embedded Peripheral IP User Guide: ug_embedded_ip.pdf (In Support / Literature/ Documentation:User Guides)

[17] making_qsys_components.pdf

[18] Avalon Interface Specifications: mnl_avalon_spec.pdf (In Support / Literature/ Documentation:Functional Specifications)

[19] Altera SoC Embedded Design Suite User Guide: (In Support / Literature/ Documentation: User Guides)

[20] Linux Device Drivers Book (Open to the public by its authors it is available in the following websites: http://www.etnassoft.com/biblioteca/linux-device-drivers/ and http://free-electrons.com/doc/books/ldd3.pdf)