



[Home](#) [Archive](#) [ARM](#) [FPGA](#) [How to](#) [Projects](#) [Imprint](#)

Altera DE1-SoC GHRD

[Introduction](#)
[Hardware](#)
[Architecture](#)
[HPS-to-FPGA Addr. Map](#)
[Lightweight HPS-to-FPGA Addr. Map](#)
[SDRAM assignments](#)
[Programming the EPCQ Device](#)
[How to debug?](#)
[Download](#)

(2) J-Link Debugger (C5SoC)

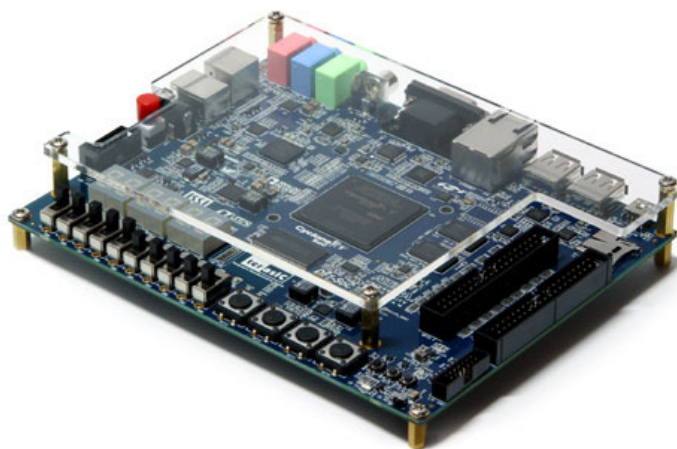
Introduction

Inspired by the Golden Hardware Reference Design from [RocketBoards](#) I have created my own design for the Altera DE1-SoC Board. It is based on the Terasic Verilog example with some modifications and ported to VHDL. The design was created with the Quartus II Web Edition [15.0.2.153](#)

[^ top](#)

Hardware

Here an [Altera DE1-SoC Board](#) was used:



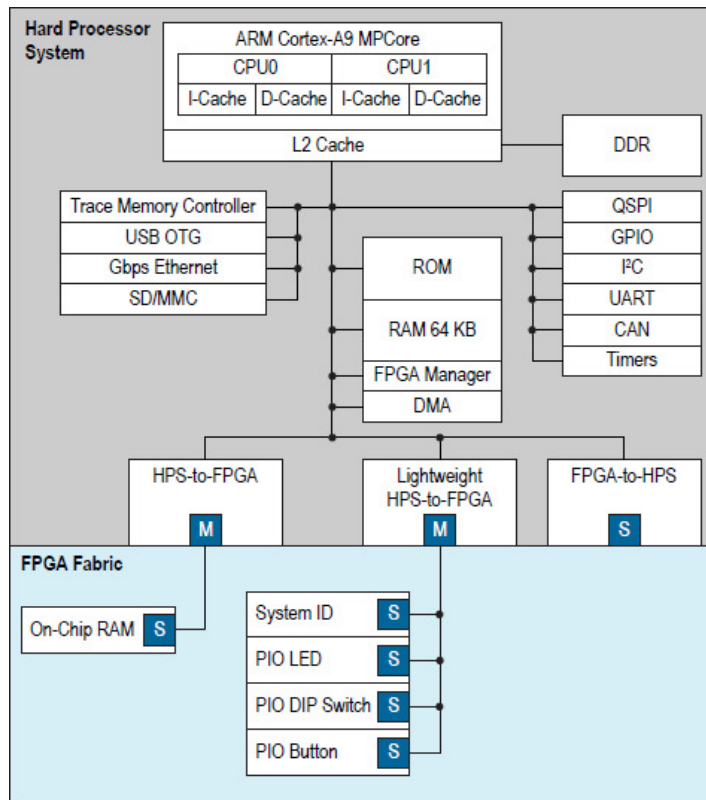
The DE1-SoC board provides a lot of functionality. But only the following subset was used:

- Cyclone V SoC (5SCEMA5F31C6)
- ARM Cortex-A9 (HPS)
- 1GB (2x256Mx16) DDR3 SDRAM on HPS
- USB to UART (micro USB type B connector)
- 4 User Keys (FPGA x4)
- 10 User switches (FPGA x10)
- 11 User LEDs (FPGA x10, HPS x 1)

[^ top](#)

Architecture

The architecture of the hardware looks like:



(Picture source: [RocketBoards](#))

The architecture looks very similar like the design from RocketBoards. But here I have removed the two JTAG Masters which was needed for the System Console.

[^] top

HPS-to-FPGA Address Map

From the perspective of the MPU (Cortex-A9) the HPS-to-FPGA address starts at **0xC0000000** and the map looks like:

Peripheral	Address Offset	Size (bytes)	Attribute
onchip_memory2_0	0x0	64K	On-chip RAM as scratch pad

[^] top

Lightweight HPS-to-FPGA Address Map

From the perspective of the MPU (Cortex-A9) the Lightweight HPS-to-FPGA address starts at **0xFF200000** and the map looks like:

Peripheral	Address Offset	Size (bytes)	Attribute
sysid_qsys	0x10000	8	Unique system ID
led_pio	0x10040	32	LED output
dipsw_pio	0x10080	32	DIP switch input
button_pio	0x100C0	32	Push button input

[^] top

SDRAM assignments

After the **very first run** of Qsys "Generate HDL..." the SDRAM assignments must be executed. After the Qsys code generation start "Analysis & Synthesis". It takes a few minutes (depending on host machine speed) before the "Analysis & Synthesis" will finish.

Now the SDRAM assignments can be created. In Quartus, go to "Tools > Tcl Scripts..." and select the **hps_sdram_p0_pin_assignments.tcl** file. Click "Run" to execute. A notification window will appear when the script is completed. Now the full "Compile Design" process can be used.

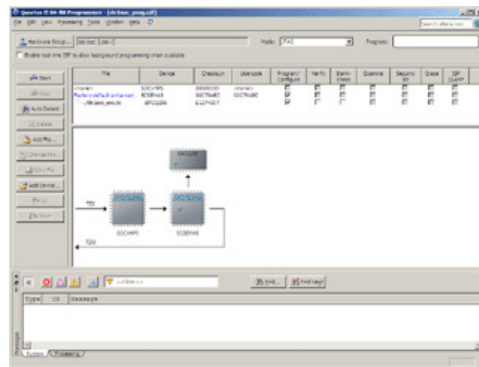
For more information please take a look here at [RocketBoards](https://www.emb4fun.de/fpga/de1socghrd/index.html).

[^ top](#)

Programming the EPCQ Device

To program the quad serial configuration (EPCQ) device a JTAG indirect configuration (.jic) file is needed. The information for how to create this file and program EPCQ device can be found in the [User Manual](#) of the DE1-SoC Board.

Alternative the files from the download can be used. After downloading and unpacking the "de1soc-ghrd.zip" file a release folder with all needed files are available. Open the Quartus II Programmer and use "File > Open..." and select the **de1soc_prog.cdf** file from the release folder. The window should now look like:



(Click inside the picture to expand)

Press the "Start" button to program the device. If the device was programmed successful, close the programmer and power the board off and on. Now the LEDR9 should flashing. KEY3 can be used to reset the board.

Note: "_clean.bat" will clean the release folder too. The files inside the release folder can be created after a successful "Compilation" of the design. After the compilation use "_create_jic.bat" for creating the files.

[^ top](#)

How to debug?

How to debug the Cortex-A9 inside the Cyclone V SoC with a J-Link?
This is another story, which can be find [here](#).

[^ top](#)

Download

 Altera DE1-SoC design [de1soc-ghrd_20161216](#) (3.51 MB, for Quartus II 15.0.2)

[^ top](#)