

SoC-FPGA Design

DE1-SoC Guide

Real Time Embedded Systems Course

LAP – IC – EPFL

Version 0.5 (Preliminary)

René Beuchat

1 Introduction

The development of embedded systems based on chips containing one or more microprocessors and hard-core peripherals, as well as an FPGA part is becoming more and more important. This technology gives the designer a lot of freedom and powerful abilities. Classical design flows with microcontrollers are emphasized with the full power of FPGAs.

Mixed designs are becoming a reality with. One can now design specific accelerators to greatly improve algorithms, or create specific programmable interfaces with the external world.

Two main HDL (**H**ardware **D**esign **L**anguage) languages are available for the design of the FPGA part: **VHDL** and Verilog. There also exist other tools that perform automatic translations from C to HDL. New emerging technologies like OpenCL allow compatibility between high-level software design, and low-level hardware implementations as:

- Compilation for single or multicore processors
- Compilation for GPUs (Graphical Processing Unit)
- Translation and compilation for FPGAs. The latest models use a PCIe interface or some other way of parameters passing between the main processor and the FPGA

This guide assumes users know how to use QuartusII, NIOSII, Qsys and ModelSim-Altera.

We will be using the Terasic DE1-SoC board: <http://de1-soc.terasic.com>

2 Terasic DE1-SoC Board

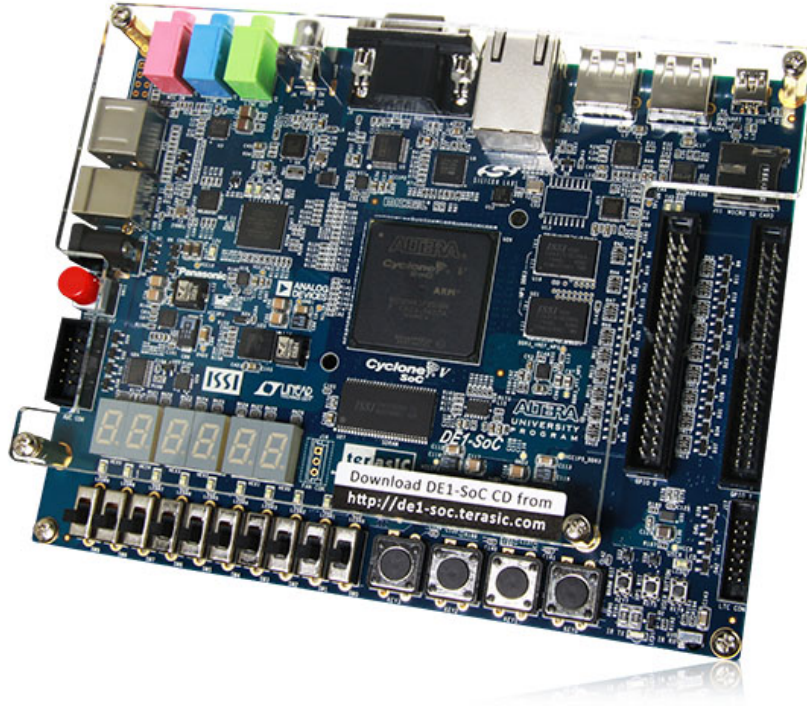


Fig. 1. Terasic DE1-SoC Board

The DE1-SoC board has many features that allow users to implement a wide range of designed circuits. We will discuss some noteworthy features in this guide.

2.1 Specifications

2.1.1 FPGA Device

- Cyclone V SoC 5CSEMA5F31C6 Device
- Dual-core ARM Cortex-A9 (HPS)
- 85K Programmable Logic Elements
- 4'450 Kbits embedded memory
- 6 Fractional PLLs
- 2 Hard Memory Controllers

2.1.2 Configuration and Debug

- Quad Serial Configuration device – EPCQ256 on FPGA
- On-Board USB Blaster II (Normal type B USB connector)

2.1.3 Memory Device

- 64MB (32Mx16) SDRAM on FPGA
- 1GB (2x256Mx16) DDR3 SDRAM on HPS
- Micro SD Card Socket on HPS

2.1.4 Communication

- Two Port USB 2.0 Host (ULPI interface with USB type A connector)
- USB to UART (micro USB type B connector)
- 10/100/1000 Ethernet
- PS/2 mouse/keyboard
- IR Emitter/Receiver

2.1.5 Connectors

- Two 40-pin Expansion Headers
- One 10-pin ADC Input Header
- One LTC connector (One Serial Peripheral Interface (SPI) Master, one I2C and one GPIO interface)

2.1.6 Display

- 24-bit VGA DAC

2.1.7 Audio

- 24-bit CODEC, line-in, line-out, and microphone-in jacks

2.1.8 Video Input

- TV Decoder (NTSC/PAL/SECAM) and TV-in connector

2.1.9 ADC

- Fast throughput rate: 1 MSPS
- Channel number: 8
- Resolution: 12 bits
- Analog input range : 0 ~ 2.5 V or 0 ~ 5V as selected via the RANGE bit in the control register

2.1.10 Switches, Buttons and Indicators

- 4 User Keys (FPGA x4)
- 10 User switches (FPGA x10)
- 11 User LEDs (FPGA x10; HPS x 1)
- 2 HPS Reset Buttons (HPS_RST_n and HPS_WARM_RST_n)
- Six 7-segment displays

2.1.11 Sensors

- G-Sensor on HPS

2.1.12 Power

- 12V DC input

2.1.13 Block Diagram

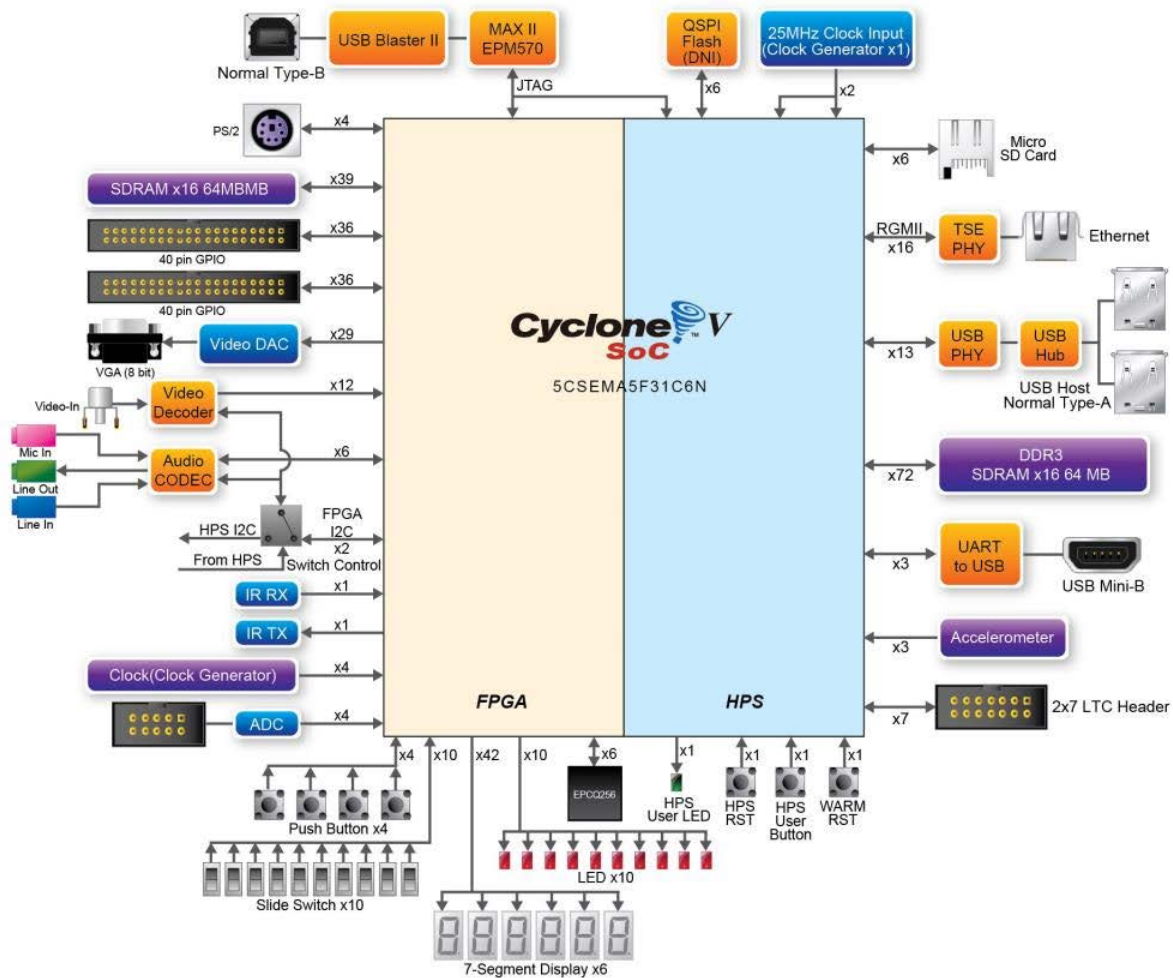


Fig. 2. Block Diagram of the DE1-SoC Board

2.2 Layout

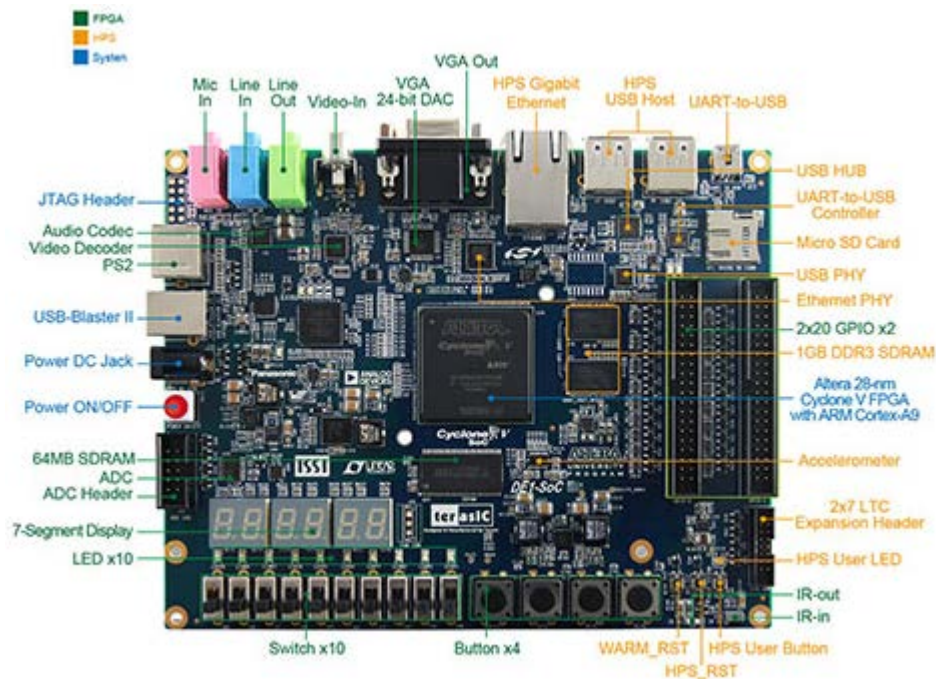


Fig. 3. Front

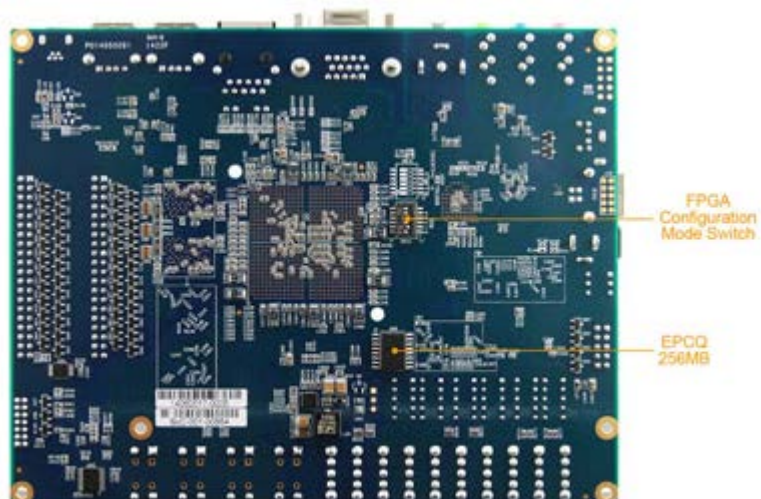


Fig. 4. Back

- Green for peripherals directly connected to the FPGA
- Orange for peripherals directly connected to the HPS
- Blue for board control

Manuals and resources are available on the DE1-SoC [resources](#) page.

3 Cyclone V Overview

This section describes some features of the Cyclone V family of devices. We do not list all features, but only the ones most important to us. All this information, along with the most complete documentation regarding this family can be found on the [Cyclone V Device Handbook](#), more specifically [Volume 3: Hard Processor System Technical Reference Manual](#).

3.1 Introduction to the Cyclone V Hard Processor System

The Cyclone V device is a single-die system on a chip (SoC) that consists of two distinct parts – a hard processor system (HPS) portion and an FPGA portion.

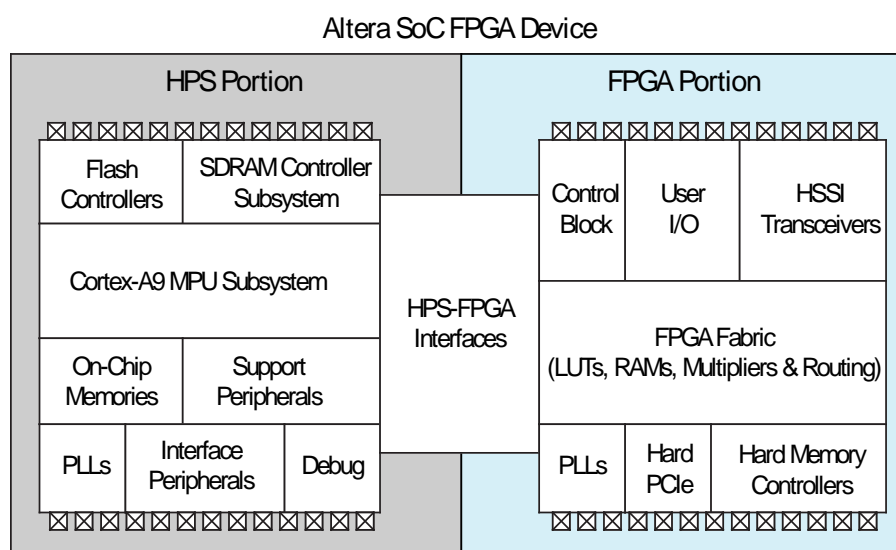


Fig. 5. Altera SoC FPGA Device Block Diagram

The HPS contains a microprocessor unit (MPU) subsystem with single or dual ARM Cortex-A9 MPCore processors, flash memory controllers, SDRAM L3 Interconnect, on-chip memories, support peripherals, interface peripherals, debug capabilities, and phase-locked loops (PLLs). The dual-processor HPS supports symmetric (SMP) and asymmetric (AMP) multiprocessing.

*The DE1-SoC has a **dual**-processor HPS.*

The FPGA portion of the device contains the FPGA fabric, a control block (CB), phase-locked loops (PLLs), and depending on the device variant, high-speed serial interface (HSSI) transceivers, hard PCI Express (PCIe) controllers, and hard memory controllers.

*The DE1-SoC does **not** contain any HSSI transceivers, or hard PCIe controllers.*

The HPS and FPGA portions of the device are distinctly different. The HPS can boot from multiple sources, including the FPGA fabric and external flash. In contrast, the FPGA must be configured through either the HPS or an externally supported device.

The MPU subsystem can boot from flash devices connected to the HPS pins. Or, when the FPGA portion is configured by an external source, the MPU subsystem can boot from memory available on the FPGA portion of the device.

The HPS and FPGA portions of the device each have their own pins. Pins are not freely shared between the HPS and the FPGA fabric. The FPGA I/O pins are configured by an FPGA configuration image through the HPS

or any external source supported by the device. The HPS I/O pins are configured by *software* executing in the HPS. Software executing on the HPS accesses control registers in the system manager to assign HPS I/O pins to the available HPS modules.

*The *software* that configures the HPS I/O pins is called the *Preloader*.*

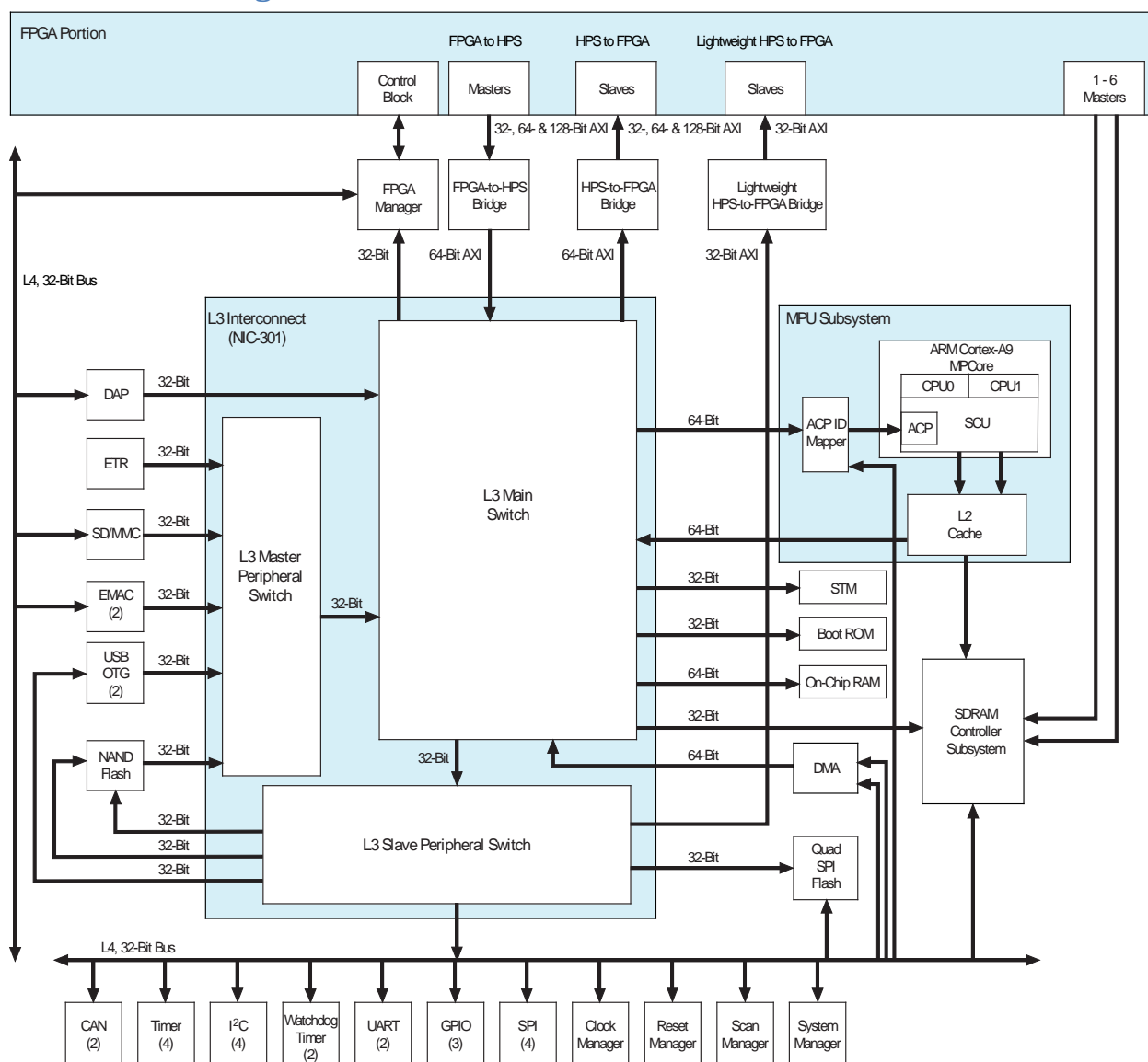
The HPS and FPGA portions of the device have separate external power supplies and independently power on. You can power on the HPS without powering on the FPGA portion of the device. However, to power on the FPGA portion, the HPS must already be on or powered on at the same time as the FPGA portion. You can also turn off the FPGA portion of the device while leaving the HPS power on.

3.2 Features of the HPS

The following list contains the main modules of the HPS:

- MPU subsystem featuring dual ARM Cortex-A9 MPCore processors
- General-purpose Direct Memory Access (DMA) controller
- Two Ethernet media access controllers (EMACs)
- Two USB 2.0 On-The-Go (OTG) controllers
- NAND flash controller
- Quad SPI flash controller
- Secure Digital (SD) / MultiMediaCard (MMC) controller
- Two serial peripheral interface (SPI) master controllers
- Two SPI slave controllers
- Four inter-integrated circuit (I²C) controllers
- 64 KB on-chip RAM
- 64 KB on-chip *boot* ROM
- Two UARTs
- Four timers
- Two watchdog timers
- Three general-purpose I/O (GPIO) interfaces
- Two controller area network (CAN) controllers
- ARM CoreSight debug components
- System manager
- Clock manager
- Reset manager
- Scan manager
- FPGA manager

3.3 HPS Block Diagram



3.4 System Integration Overview

In this part, we briefly go through some features provided by the most important HPS components.

3.4.1 MPU Subsystem

Here are a few important features of the MPU subsystem:

- Interrupt controller
- One general-purpose timer and one watchdog timer per processor
- One Memory management unit (MMU) per processor

The HPS masters the L3 interconnect and the SDRAM controller subsystem.

3.4.2 SDRAM Controller Subsystem

The SDRAM controller subsystem is *mastered by HPS masters and FPGA fabric masters*. It supports DDR2, DDR3, and LPDDR2 devices. It is composed of 2 parts:

- SDRAM controller
- DDR PHY (interfaces the single port memory controller to the HPS I/O)

The DE1-SoC contains DDR3 SDRAM on the HPS

3.4.3 System Manager

This is one of the most *essential* HPS components. It offers a few important features:

- *Pin multiplexing* (term used for the *software* configuration of the HPS I/O pins by the *Preloader*)
- Freeze controller that places I/O elements into a safe state for configuration
- Low-level control of peripheral features not accessible through the control and status registers (CSRs)

*The low-level control of some peripheral features that are not accessible through the CSRs is **not** externally documented.*

3.4.4 FPGA Manager

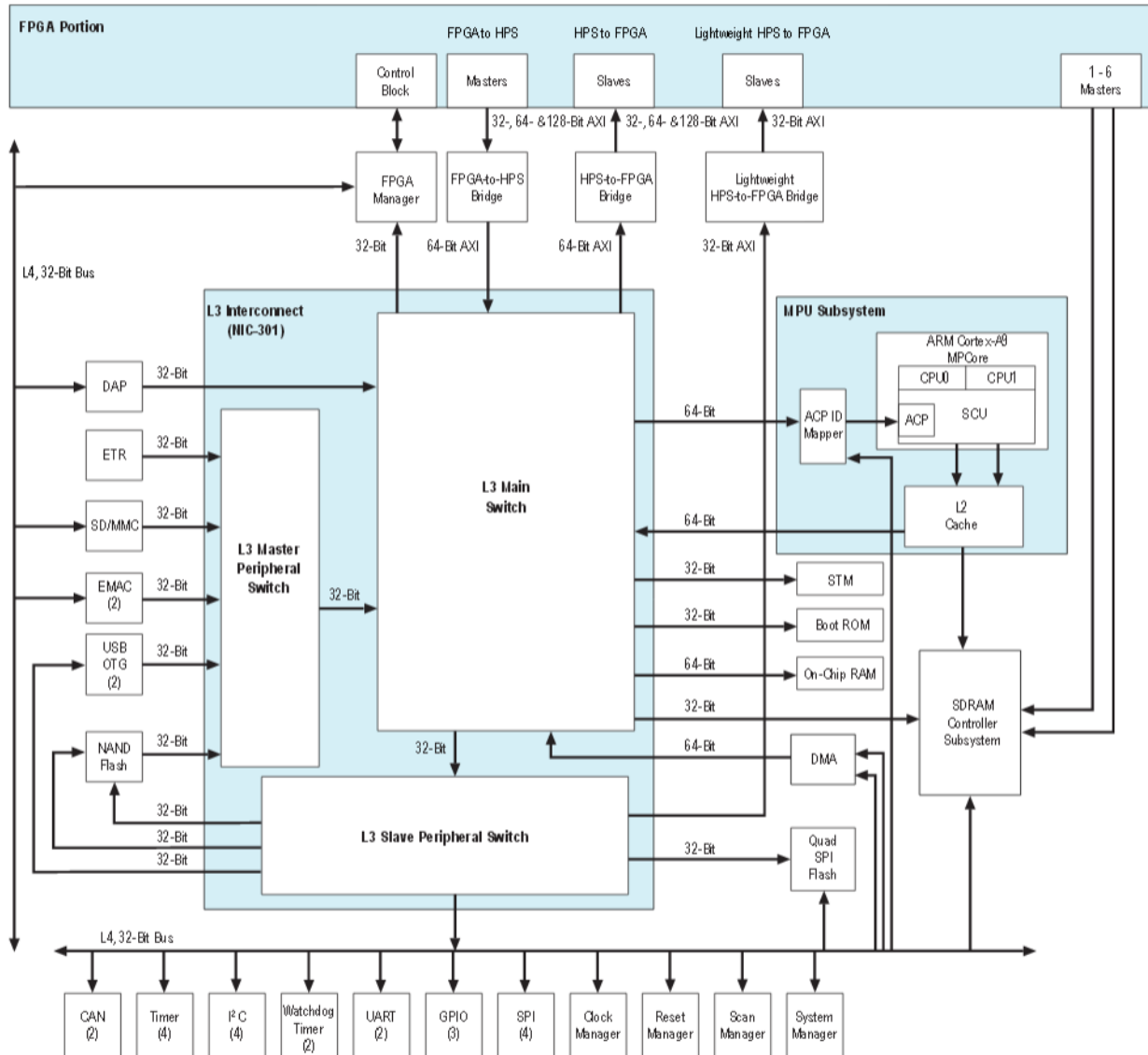
The FPGA manager offers the following features:

- Manages configuration of the FPGA portion of the device
- Monitors configuration-related signals in the FPGA
- Provides 32 general-purpose inputs and 32 general-purpose outputs to the FPGA fabric

4 SoC part test

4.1 HPS Architecture

To be able to program the ARM9's processors it is almost necessary to have the global view of the HPS architecture.



4.2 Hardware development

4.2.1 Qsys integration

Starting with **QuartusII** and after creating a project, select **Tools → Qsys**

In **Qsys**, open **Library → Embedded Processors → Hard Processor System** the window with description of the parameters for the HPS is open.

The **FPGA Interface** tab allows the access from to the FPGA part with the HPS part.

Hard Processor System
altera_hps

Block Diagram

hps_0

h2f_mpu_gp conduit conduit memory
f2h_sdr0m0_clock clock reset h2f_reset
f2h_sdr0m0_data axi axi h2f_axi_master
h2f_axi_clock clock axi h2f_lw_axi_master
f2h_axi_slave axi
h2f_lw_axi_clock clock

altera_hps

FPGA Interfaces | Peripheral Pin Multiplexing | HPS Clocks | SDRAM

General

- ☐ Enable MPU standby and event signals
- ☒ Enable MPU general purpose signals
- ☐ Enable Debug APB interface
- ☐ Enable System Trace Macrocell hardware events
- ☐ Enable FPGA Cross Trigger Interface
- ☐ Enable FPGA Trace Port Interface Unit
- ☐ Enable boot from fpga signals
- ☐ Enable HLGPI Interface

AXI Bridges

FPGA-to-HPS interface width: 64-bit
HPS-to-FPGA interface width: 64-bit
Lightweight HPS-to-FPGA interface width: 32-bit

FPGA-to-HPS SDRAM Interface

Click the '+' and '-' buttons to add and remove FPGA-to-HPS SDRAM ports.

Name	Type	Width
f2h_sdr0m0	SDRAM	128

Resets

- ☐ Enable HPS-to-FPGA cold reset output
- ☐ Enable HPS warm reset handshake signals
- ☐ Enable FPGA-to-HPS debug reset request
- ☐ Enable FPGA-to-HPS warm reset request
- ☐ Enable FPGA-to-HPS cold reset request

DMA Peripheral Request

Peripheral Request ID	Enabled
0	No
1	No
2	No
3	No
4	No

Presets

Project Library

- ELPIDA EDJ1108BASE-8C
- ELPIDA EDJ308BASE-8C
- JEDEC DDR2-1066 256MB X8
- JEDEC DDR2-1066 512MB X8
- JEDEC DDR2-400 256MB X8
- JEDEC DDR2-400 512MB X8
- JEDEC DDR2-533 256MB X8
- JEDEC DDR2-533 512MB X8
- JEDEC DDR2-667 256MB X8
- JEDEC DDR2-667 512MB X8
- JEDEC DDR2-800 256MB X8
- JEDEC DDR2-800 512MB X8
- JEDEC DDR3-1066E 1GB X8
- JEDEC DDR3-1066E 2GB X8
- JEDEC DDR3-1066E 512MB X8
- JEDEC DDR3-1066F 1GB X8
- JEDEC DDR3-1066F 2GB X8
- JEDEC DDR3-1066F 512MB X8
- JEDEC DDR3-1066G 1GB X8
- JEDEC DDR3-1066G 2GB X8
- JEDEC DDR3-1066G 512MB X8
- JEDEC DDR3-1G4 1GB X8
- JEDEC DDR3-1G4 2GB X8
- JEDEC DDR3-1G6 1GB X8
- JEDEC DDR3-1G6 2GB X8
- JEDEC DDR3-800D 1GB X8
- JEDEC DDR3-800D 2GB X8
- JEDEC DDR3-800D 512MB X8
- JEDEC DDR3-800E 1GB X8
- JEDEC DDR3-800E 2GB X8
- JEDEC DDR3-800E 512MB X8
- JEDEC DDR3L-1066E 1GB X8
- JEDEC DDR3L-1066E 2GB X8
- JEDEC DDR3L-1066E 512MB X8
- JEDEC DDR3L-1066F 1GB X8
- JEDEC DDR3L-1066F 2GB X8
- JEDEC DDR3L-1066F 512MB X8
- JEDEC DDR3L-1066G 1GB X8
- JEDEC DDR3L-1066G 2GB X8
- JEDEC DDR3L-1066G 512MB X8
- JEDEC DDR3L-1G4 2GB X8
- JEDEC DDR3L-1G6 2GB X8
- JEDEC DDR3L-800D 1GB X8
- JEDEC DDR3L-800D 2GB X8
- JEDEC DDR3L-800D 512MB X8
- JEDEC DDR3L-800E 1GB X8
- JEDEC DDR3L-800E 2GB X8
- JEDEC DDR3L-800E 512MB X8
- MICRON MT41J128M16HA-15E

Hard Processor System
altera_hps

Block Diagram

hps_0

h2f_mpu_gp conduit conduit memory
f2h_sdr0m0_clock clock reset h2f_reset
f2h_sdr0m0_data axi axi h2f_axi_master
h2f_axi_clock clock axi h2f_lw_axi_master
f2h_axi_slave axi
h2f_lw_axi_clock clock

altera_hps

Resets

- ☐ Enable HPS-to-FPGA cold reset output
- ☐ Enable HPS warm reset handshake signals
- ☐ Enable FPGA-to-HPS debug reset request
- ☐ Enable FPGA-to-HPS warm reset request
- ☐ Enable FPGA-to-HPS cold reset request

DMA Peripheral Request

Peripheral Request ID	Enabled
0	No
1	No
2	No
3	No
4	No

Interrupts

- ☐ Enable FPGA-to-HPS Interrupts

HPS-to-FPGA

- ☐ Enable CAN interrupts
- ☐ Enable dock peripheral interrupts
- ☐ Enable CTI interrupts
- ☐ Enable DMA interrupts
- ☐ Enable EMAC interrupts
- ☐ Enable FPGA manager interrupt
- ☐ Enable GPIO interrupts
- ☐ Enable I2C-EMAC interrupts
- ☐ Enable I2C peripheral interrupts
- ☐ Enable L4 timer interrupts
- ☐ Enable NAND interrupt
- ☐ Enable OSC timer interrupts
- ☐ Enable QSPI interrupt
- ☐ Enable SD/MMC interrupt
- ☐ Enable SPI master interrupts
- ☐ Enable SPI slave interrupts
- ☐ Enable UART interrupts
- ☐ Enable USB interrupts
- ☐ Enable watchdog interrupts

Presets

Project Library

- ELPIDA EDJ1108BASE-8C
- ELPIDA EDJ308BASE-8C
- JEDEC DDR2-1066 256MB X8
- JEDEC DDR2-1066 512MB X8
- JEDEC DDR2-400 256MB X8
- JEDEC DDR2-400 512MB X8
- JEDEC DDR2-533 256MB X8
- JEDEC DDR2-533 512MB X8
- JEDEC DDR2-667 256MB X8
- JEDEC DDR2-667 512MB X8
- JEDEC DDR2-800 256MB X8
- JEDEC DDR2-800 512MB X8
- JEDEC DDR3-1066E 1GB X8
- JEDEC DDR3-1066E 2GB X8
- JEDEC DDR3-1066E 512MB X8
- JEDEC DDR3-1066F 1GB X8
- JEDEC DDR3-1066F 2GB X8
- JEDEC DDR3-1066F 512MB X8
- JEDEC DDR3-1066G 1GB X8
- JEDEC DDR3-1066G 2GB X8
- JEDEC DDR3-1066G 512MB X8
- JEDEC DDR3-1G4 1GB X8
- JEDEC DDR3-1G4 2GB X8
- JEDEC DDR3-1G6 1GB X8
- JEDEC DDR3-1G6 2GB X8
- JEDEC DDR3-800D 1GB X8
- JEDEC DDR3-800D 2GB X8
- JEDEC DDR3-800D 512MB X8
- JEDEC DDR3-800E 1GB X8
- JEDEC DDR3-800E 2GB X8
- JEDEC DDR3-800E 512MB X8
- JEDEC DDR3L-1066E 1GB X8
- JEDEC DDR3L-1066E 2GB X8
- JEDEC DDR3L-1066E 512MB X8
- JEDEC DDR3L-1066F 1GB X8
- JEDEC DDR3L-1066F 2GB X8
- JEDEC DDR3L-1066F 512MB X8
- JEDEC DDR3L-1066G 1GB X8
- JEDEC DDR3L-1066G 2GB X8
- JEDEC DDR3L-1066G 512MB X8
- JEDEC DDR3L-1G4 2GB X8
- JEDEC DDR3L-1G6 2GB X8
- JEDEC DDR3L-800D 1GB X8
- JEDEC DDR3L-800D 2GB X8
- JEDEC DDR3L-800D 512MB X8
- JEDEC DDR3L-800E 1GB X8
- JEDEC DDR3L-800E 2GB X8
- JEDEC DDR3L-800E 512MB X8
- MICRON MT41J128M16HA-15E
- MICRON MT41J128M16HA-187E
- MICRON MT41J128M16HA-187F

Apply Update... Delete

With the **PeripheralPin Multiplexing**, some I/O interface can be used by the HPS part or the FPGA part. The selection is done here.

4.3 Software development

4.3.1 ARM DS-5 tools

There are some differences between the versions of DS-5.

The one installed for the test is:

ARM DS-5 (DS-5 Altera Edition (Evaluation))
Version: 5.18.0
Build number: 5180018

4.3.2 Hello World on ARM HPS part

Copy the directory from Altera examples:

C:\altera\13.1\embedded\examples\software

And un-gz the file: Altera-SoCFPGA-HelloWorld-Baremetal-ARMCC.tar.gz

Then un-tar it.

The directory **Altera-SoCFPGA-HelloWorld-Baremetal-ARMCC** can then be copied in the Eclipse WorkSpace and Imported as a new project. The files inside are:

- .cproject used by Eclipse
- .project used by Eclipse
- ****.launch ??
- Makefile for the Compiler/Assembler/Linker
An important info is the flag for the cpu: --cpu=Cortex-A9.no_neon.no_vfp
- scatter.scat Info for the compiler for the Code, Data, Stack and Heap addresses
in this case in the internal SRAM

4.3.2.1 Scatter.scat

```

*****
;
; Copyright (c) 2013 Altera All Rights Reserved.
*****
;

; Scatter-file for OnChip RAM based example
; This scatter-file places application code, data, stack and heap at suitable addresses in the memory map.

; Altera SoC-FPGA has 64kB of internal OnChip RAM

OCRAM 0xFFFF0000 0x10000
{
    APP_CODE +0

```

```

{
    * (+RO, +RW, +ZI)
}

ARM_LIB_STACKHEAP 0xFFFF8000 EMPTY 0x8000 ; Application heap and stack
{}
}

```

4.3.2.2 Makefile

Makefile for the ARM compiler

```

# Copyright (C) ARM Limited, 2011. All rights reserved.
#
# This example is intended to be built with the ARM Compiler armcc

TARGET=Altera-SoCFPGA-HelloWorld-Baremetal-ARMCC.axf

CC=armcc
AS=armasm
LD=armlink
AR=armar

# Select build rules based on Windows or Unix
ifdef WINDIR
DONE=@if exist $(1) echo Build completed.
RM=if exist $(1) del /q $(1)
SHELL=$(WINDIR)\system32\cmd.exe
else
ifdef windir
DONE=@if exist $(1) echo Build completed.
RM=if exist $(1) del /q $(1)
SHELL=$(windir)\system32\cmd.exe
else
DONE=@if [ -f $(1) ]; then echo Build completed.; fi
RM=rm -f $(1)
endif
endif

all: $(TARGET)
    $(call DONE,$(TARGET))

rebuild: clean all

clean:
    $(call RM,*.o)
    $(call RM,$(TARGET))

hello.o: hello.c
    $(CC) -c -g --cpu=Cortex-A9.no_neon.no_vfp -O0 hello.c

$(TARGET): hello.o scatter.scatscat
    $(LD) hello.o -o $(TARGET) --cpu=Cortex-A9.no_neon.no_vfp --scatter=scatter.scatscat

```

4.3.3 GPIO access

The references for gpio are:

- http://www.altera.com/literature/hb/cyclone-v/cv_54022.pdf
- <http://www.altera.com/literature/hb/cyclone-v/hps.html>
- Supports up to 71 I/O pins and 14 input-only pins depend on device variant

On de1-soc:

- Only 1 Button for HPS GPIO 1
- Only 1 LED for HPS GPIO 1

Pin Name	HPS GPIO	Register [bit]	Function	Address	Dir
HPS_KEY	GPIO54	GPIO1[25]	I/O	0xFF20 9000	In
HPS_LED	GPIO53	GPIO1[24]	I/O	0xFF20 9000	Out

HPS peripherals are mapped to HPS base address space 0xFC00 0000 with 64KB size.

Registers of GPIO0 controller are mapped to the base address 0xFF20 8000 - 0xFF20 8FFF (4KB size)

Registers of GPIO1 controller are mapped to the base address 0xFF20 9000 - 0xFF20 9FFF (4KB size)

Registers of GPIO2 controller are mapped to the base address 0xFF20 A000 - 0xFF20 8FFF (4KB size)

		http://www.altera.com/literature/hb/cyclone-v/cv_5v4.pdf		
GPIO0	0xFF20 8000 - 0xFF20 8FFF	0xFF70 8000		
GPIO1	0xFF20 9000 - 0xFF20 9FFF	0xFF70 9000		
GPIO2	0xFF20 A000 - 0xFF20 8FFF	0xFF70 A000		
LWFGASLAVES		0xFF20 0000		

gpio0	0xFF70 8000	HPS_GPIO0_ADDRESS	HPS_GPIO0_OFFSET		
gpio_swporta_dr	0	HPS_GPIO0_GPIO_SWPORTA_DR_ADDRESS	GPIO_GPIO_SWPORTA_DR_OFFSET		
gpio_swporta_ddr	0x04	HPS_GPIO0_GPIO_SWPORTA_DDR_ADDRESS	GPIO_GPIO_SWPORTA_DDR_OFFSET		
gpio_inten	0x30	HPS_GPIO0_GPIO_INTEN_ADDRESS	GPIO_GPIO_INTEN_OFFSET		
gpio_intmask	0x34	HPS_GPIO0_GPIO_INTMASK_ADDRESS	GPIO_GPIO_INTMASK_OFFSET		
gpio_inttype_level	0x38	HPS_GPIO0_GPIO_INTTYPE_LEVEL_ADDRESS	GPIO_GPIO_INTTYPE_LEVEL_OFFSET		
gpio_int_polarity	0x3c	HPS_GPIO0_GPIO_INT_POLARITY_ADDRESS	GPIO_GPIO_INT_POLARITY_OFFSET		
gpio_intstatus	0x40	HPS_GPIO0_GPIO_INTSTATUS_ADDRESS	GPIO_GPIO_INTSTATUS_OFFSET		
gpio_raw_intstatus	0x44	HPS_GPIO0_GPIO_RAW_INTSTATUS_ADDRESS	GPIO_GPIO_RAW_INTSTATUS_OFFSET		
gpio_debounce	0x48	HPS_GPIO0_GPIO_DEBOUNCE_ADDRESS	GPIO_GPIO_DEBOUNCE_OFFSET		
gpio_porta_eoi	0x4c	HPS_GPIO0_GPIO_PORTA_EOI_ADDRESS	GPIO_GPIO_PORTA_EOI_OFFSET		
gpio_ext_porta	0x50	HPS_GPIO0_GPIO_EXT_PORTA_ADDRESS	GPIO_GPIO_EXT_PORTA_OFFSET		
gpio_ls_sync	0x60	HPS_GPIO0_GPIO_LS_SYNC_ADDRESS	GPIO_GPIO_LS_SYNC_OFFSET		
gpio_id_code	0x64	HPS_GPIO0_GPIO_ID_CODE_ADDRESS	GPIO_GPIO_ID_CODE_OFFSET		
gpio_ver_id_code	0x6c	HPS_GPIO0_GPIO_VER_ID_CODE_ADDRESS	GPIO_GPIO_VER_ID_CODE_OFFSET		
gpio_config_reg2	0x70	HPS_GPIO0_GPIO_CONFIG_REG2_ADDRESS	GPIO_GPIO_CONFIG_REG2_OFFSET		
gpio_config_reg1	0x74	HPS_GPIO0_GPIO_CONFIG_REG1_ADDRESS	GPIO_GPIO_CONFIG_REG1_OFFSET		

--	--	--	--	--	--

4.3.3.1 Library installation

C:\altera\13.1\embedded\ip\altera\hps\altera_hps\hwlib

HERE

4.3.3.2 Reference files

hps.h		

4.3.3.2.1 Titre5

4.3.3.2.1.1 Titre6

4.3.3.2.1.1.1 Titre7

4.3.3.2.1.1.1.1 Titre8

4.3.3.2.1.1.1.1.1 Titre9

References

- Altera, Cyclone V Devices documentation,
http://www.altera.com/literature/lit-cyclone-v.jsp?ln=devices_fpga&l3=Low-Cost%20FPGAs-Cyclone%20V%20%28E,%20GX,%20GT,%20SE,%20SX,%20ST%29&l4=Documentation
- Cyclone V Device Handbook Volume 3: Hard Processor System Technical Reference Manual
http://www.altera.com/literature/hb/cyclone-v/cv_5v4.pdf
- Cyclone V Hard Processor System User Guide
http://www.altera.com/literature/hb/cyclone-v/cv_5v4_08.pdf
- Cyclone V, Device Datasheet
http://www.altera.com/literature/hb/cyclone-v/cv_51002.pdf
- Cyclone V HPS addresses
<http://www.altera.com/literature/hb/cyclone-v/hps.html>
- Cyclone V Device Handbook Volume 1: Device Interfaces and Integration
http://www.altera.com/literature/hb/cyclone-v/cyclone5_handbook.pdf

- Cyclone V, Device Overview
http://www.altera.com/literature/hb/cyclone-v/cv_51001.pdf
- SoCAL documentation (html), The Altera SoC Abstraction Layer (SoCAL) API Reference Manual
file:///C:/altera/13.1/embedded/ip/altera/hps/altera_hps/doc/socal/html/index.html
- Altera HWLIB, The Altera HW Manager API Reference Manual
file:///C:/altera/13.1/embedded/ip/altera/hps/altera_hps/doc/hwmgr/html/index.html
- Cyclone V, A Bare-Metal Debugging using ARM DS-5 Altera Edition
<http://www.youtube.com/watch?v=CJ0EHJ9oQ7Y>
- Linux Kernel Debug using ARM DS-5 Altera Edition
<http://www.youtube.com/watch?v=QcA39O6ofGw>
- FPGA-adaptive debug on the Altera SoC using ARM DS-5
<http://www.youtube.com/watch?v=2NBcUv2Txbl>
- A Look Inside: SoC FPGAs Introduction (Part 1 of 5)
<http://www.youtube.com/watch?v=RVM-ESUMOMU> (Part 1 of 5)
<http://www.youtube.com/watch?v=Ssxf8ggmQk4> (Part 2 of 5)
<http://www.youtube.com/watch?v=cWlaqt2RU84> (Part 3 of 5)
<http://www.youtube.com/watch?v=gUE669XKhUY> (Part 4 of 5)
<http://www.youtube.com/watch?v=NxZznvf5EKc> (Part 5 of 5)
- DS-5 Altera Edition: Bare-metal Debug and Trace
http://www.youtube.com/watch?v=u_xKybPhcHI
- OpenCL on FPGAs Accelerating Performance and Design Productivity — Altera
http://www.youtube.com/watch?v=M6vpq6s1h_A
-

Summary

1	Introduction	2
2	Terasic DE1-SoC Board	3
2.1	Specifications	3
2.1.1	FPGA Device	3
2.1.2	Configuration and Debug	3
2.1.3	Memory Device	3
2.1.4	Communication	3
2.1.5	Connectors	4
2.1.6	Display	4
2.1.7	Audio	4
2.1.8	Video Input.....	4
2.1.9	ADC.....	4
2.1.10	Switches, Buttons and Indicators.....	4
2.1.11	Sensors	4
2.1.12	Power	4
2.1.13	Block Diagram	5
2.2	Layout.....	5
3	Cyclone V Overview	7
3.1	Introduction to the Cyclone V Hard Processor System.....	7
3.2	Features of the HPS.....	8
3.3	HPS Block Diagram	9
3.4	System Integration Overview.....	9
3.4.1	MPU Subsystem	9
3.4.2	SDRAM Controller Subsystem	9
3.4.3	System Manager	10
3.4.4	FPGA Manager	10
4	SoC part test.....	11

4.1	HPS Architecture	11
4.2	Hardware development	11
4.2.1	Qsys integration	11
4.3	Software development.....	13
4.3.1	ARM DS-5 tools	13
4.3.2	Hello World on ARM HPS part.....	13
4.3.3	GPIO access	15

List of Figures

Fig. 1. Fig style **Error! Bookmark not defined.**

Fig. 2. Test program for specific parallel port..... **Error! Bookmark not defined.**