SoC-FPGA Design

DE1-SoC Guide

Real Time Embedded Systems Course

LAP – IC – EPFL

Version 0.7 (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

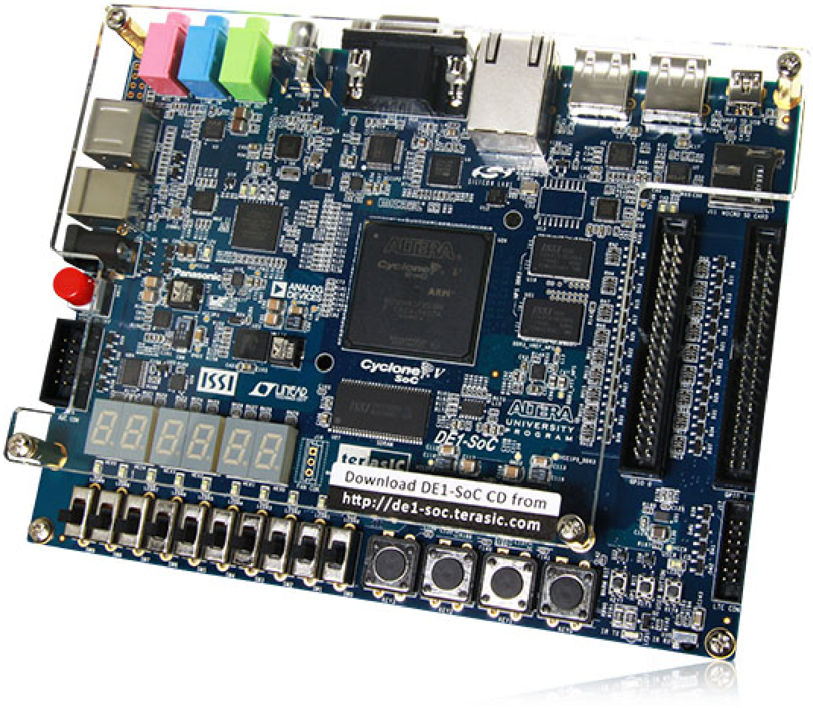


Figure 1. Terasic DE1-SoC Board

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

- *64 MB* (32Mx16) SDRAM on FPGA
- *1 GB* (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

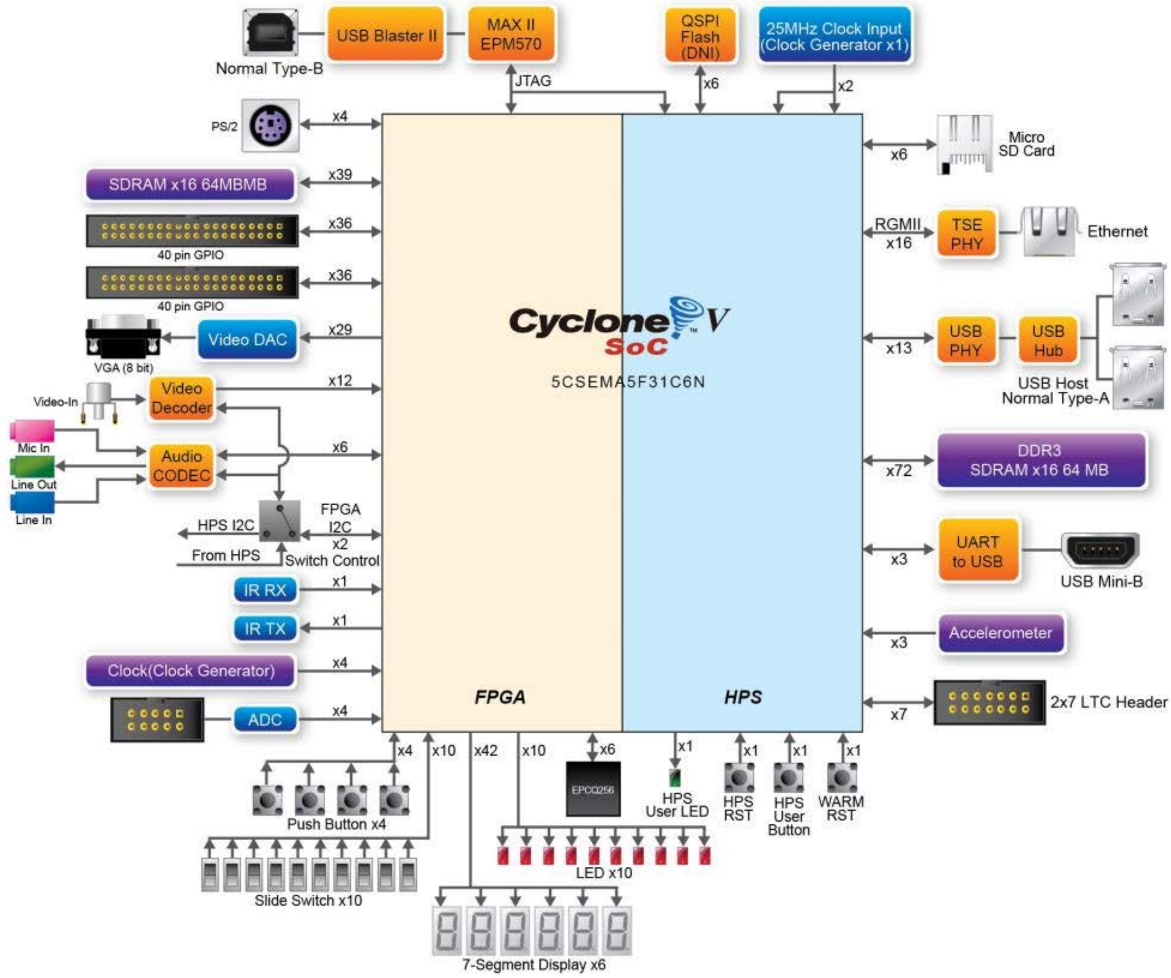


Figure 2. Block Diagram of the DE1-SoC Board

2.2 Layout

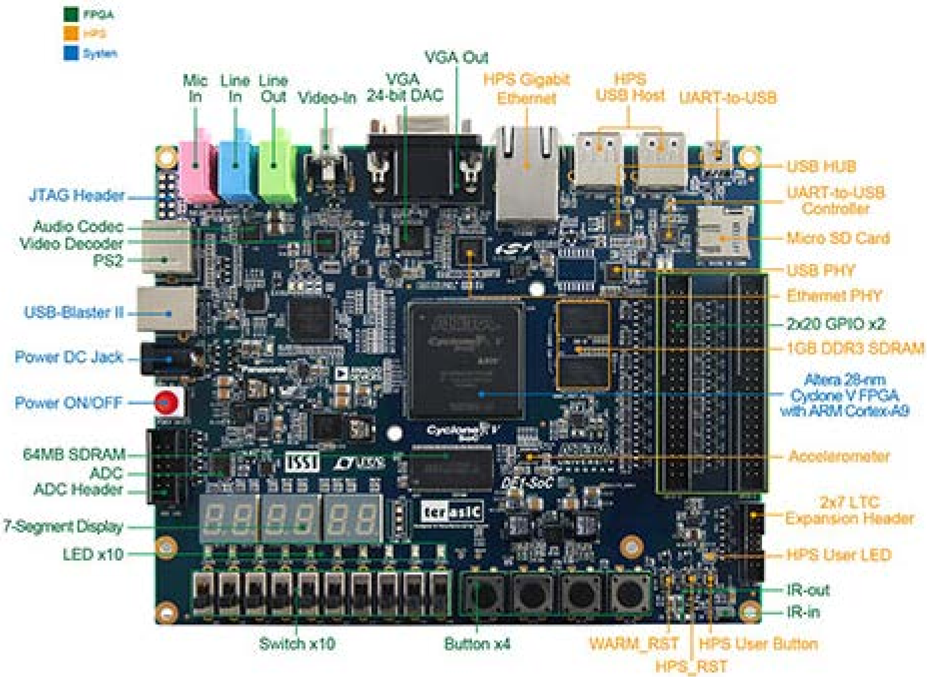


Figure 3. Front

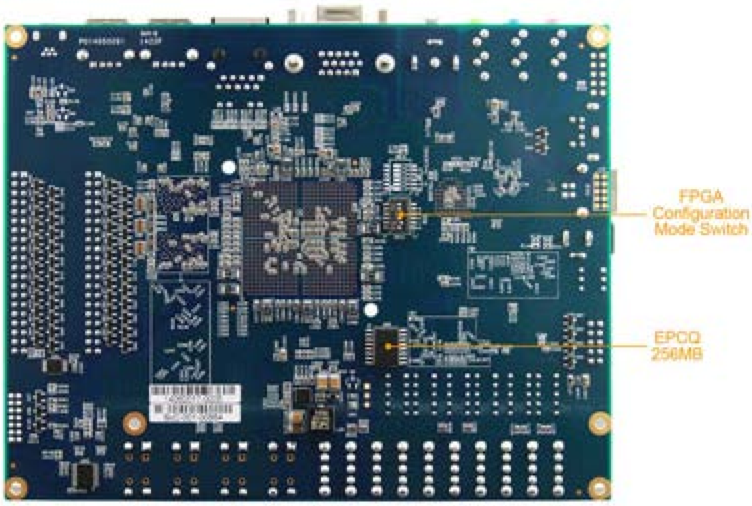


Figure 4. Back

- 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.

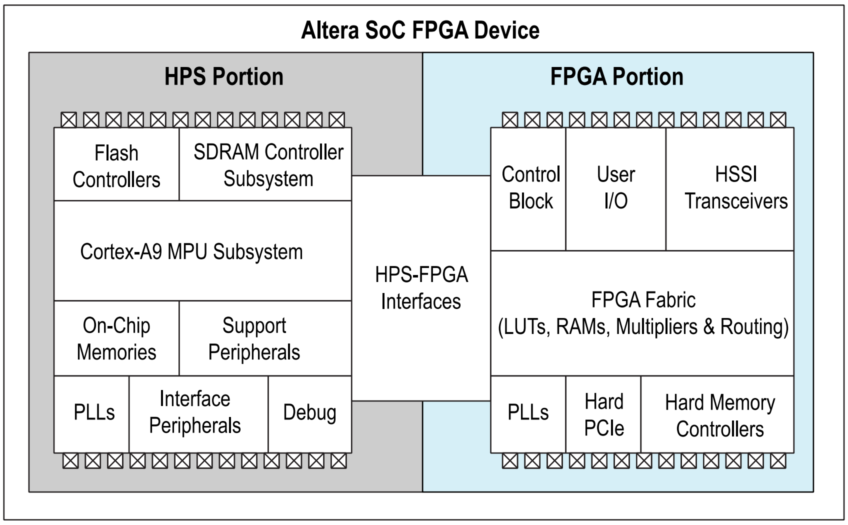


Figure 5. Altera SoC FPGA Device Block Diagram

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

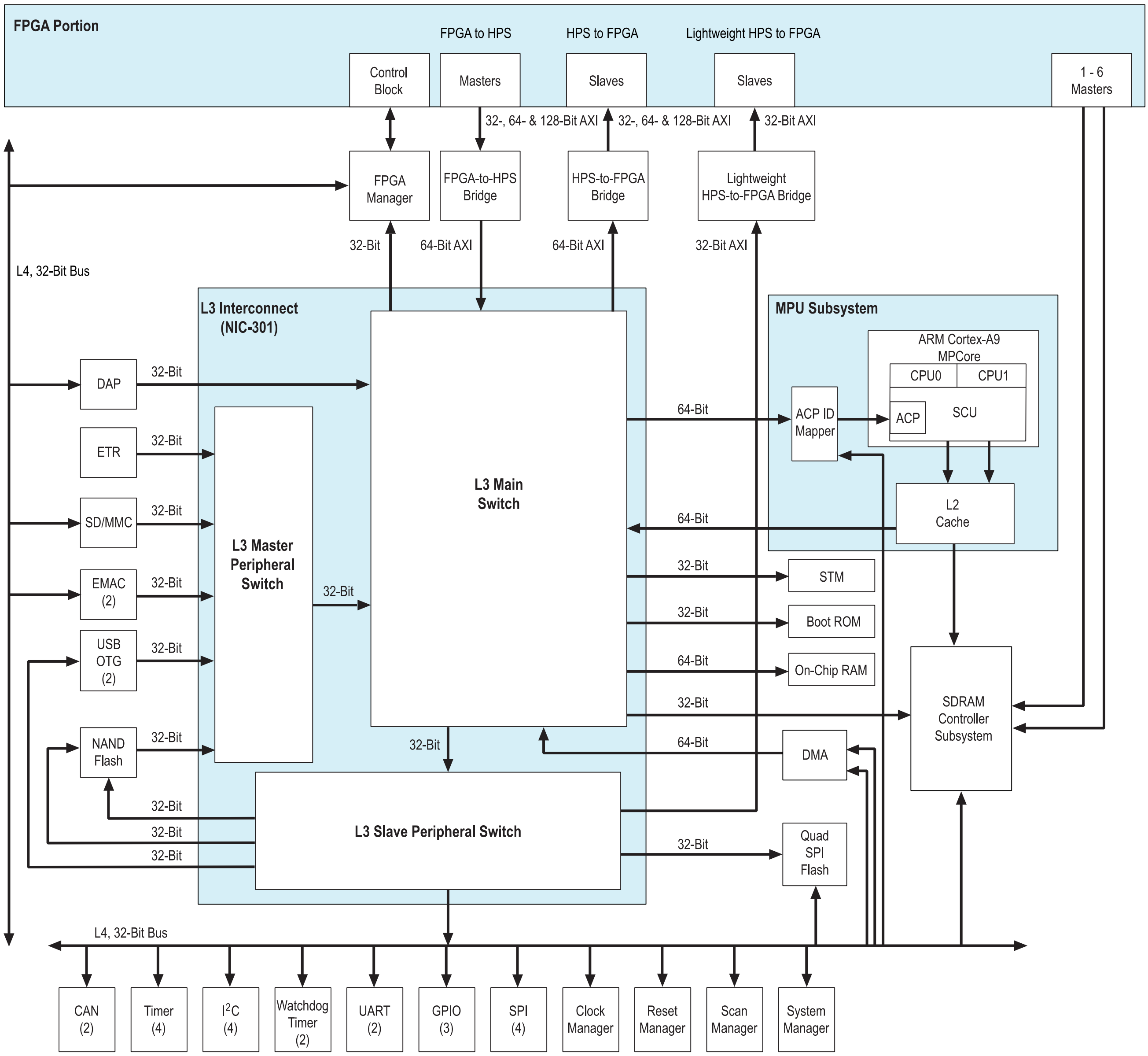


Figure 6. HPS Block Diagram

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 (I^2C) 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 System Integration Overview

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

3.3.1 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.

3.3.2 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

3.3.3 Support Peripherals

3.3.3.1 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 and should not be used in code.

3.3.3.2 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

3.3.4 Interface Peripherals

3.3.4.1 GPIO Interfaces

The HPS provides three GPIO interfaces and offer the following features:

- Supports digital de-bounce
- Configurable interrupt mode
- Supports up to 71 I/O pins and 14 input-only pins, based on device variant
- Supports up to 67 I/O pins and 14 input-only pins

The DE1-SoC has 67 I/O pins and 14 input-only pins

3.3.5 On-Chip Memory

The following on-chip memories are different from any on-chip memories located in the FPGA fabric.

3.3.5.1 On-Chip RAM

The on-chip RAM offers the following features:

- 64 KB size
- High performance for all burst lengths

3.3.5.2 Boot ROM

The boot ROM offers the following features:

- 64 KB size
- Contains the code required to support HPS boot from cold or warm reset
- Used *exclusively* for booting the HPS

The code in the boot ROM cannot be changed.

3.4 HPS-FPGA Interfaces

The HPS-FPGA interfaces provide a variety of communication channels between the HPS and the FPGA fabric. The HPS-FPGA interfaces include:

- FPGA-to-HPS bridge – a high performance bus with a configurable data width of 32, 64, or 128 bits. It allows the FPGA fabric to master transactions to slaves in the HPS. This interface allows the FPGA fabric to have full visibility into the HPS address space.
- HPS-to-FPGA bridge – a high performance bus with a configurable data width of 32, 64, or 128 bits. It allows the HPS to master transactions to slaves in the FPGA fabric.
- Lightweight HPS-to-FPGA bridge – a bus with a 32-bit fixed data width. It allows the HPS to master transactions to slaves in the FPGA fabric.

- FPGA manager interface – signals that communicate with FPGA fabric for boot and configuration.
- Interrupts – allow soft IP to supply interrupts directly to the MPU interrupt controller.
- HPS debug interface – an interface that allows the HPS debug control domain to extend into the FPGA.

3.5 HPS Address Map

3.5.1 HPS Address Spaces

The HPS address map specifies the address of slaves, such as memory and peripherals, as viewed by the HPS masters. The HPS has 3 address spaces:

Name	Description	Size
MPU	MPU subsystem	4 GB
L3	L3 interconnect	4 GB
SDRAM	SDRAM controller subsystem	4 GB

Table 1. HPS Address Spaces

The following figure shows the relationships between the different HPS address spaces. The figure is *not* to scale.

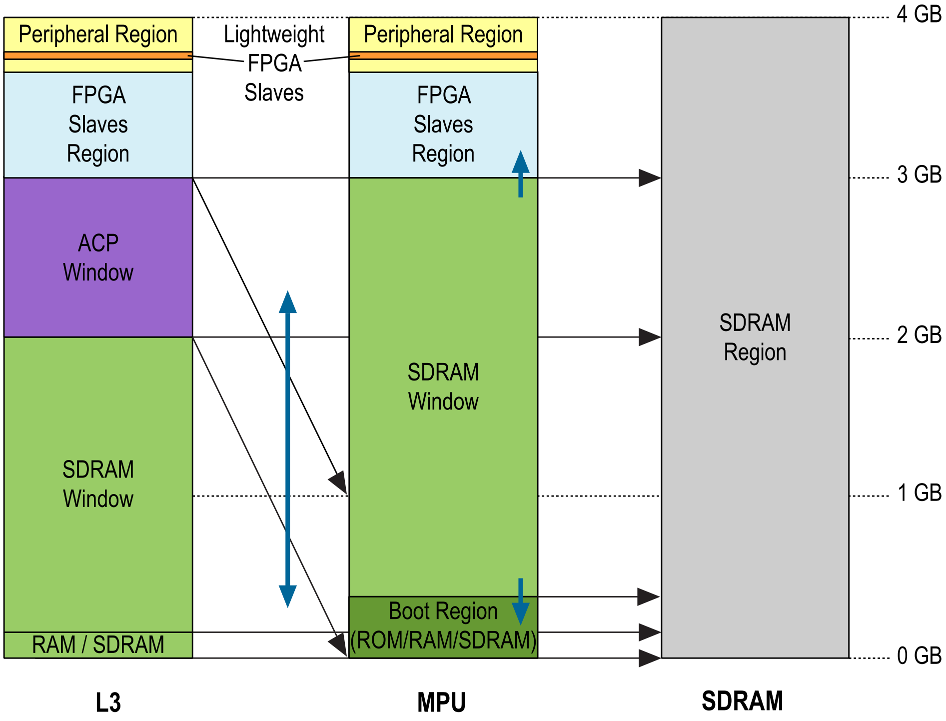


Figure 7. HPS Address Space Relations

The window regions provide access to other address spaces. The thin black arrows indicate which address space is accessed by a window region (arrows point to accessed address space).

The SDRAM window in the MPU can grow and shrink at the top and bottom (short blue vertical arrows) at the expense of the FPGA slaves and boot regions.

The ACP window can be mapped to any 1 GB region in the MPU address space (blue vertical bidirectional arrow), on gigabyte-aligned boundaries.

The following table shows the base address and size of each region that is common to the L3 and MPU address spaces.

Region Name	Description	Base Address	Size
FPGA slaves	FPGA slaves connected to the HPS-to-FPGA bridge	0xC0000000	960 MB
HPS peripherals	Slaves directly connected to the HPS (corresponds to all orange colored elements on Figure 3 and Figure 4)	0xFC000000	64 MB
Lightweight FPGA slaves	FPGA slaves connected to the Lightweight HPS-to-FPGA bridge	0xFF200000	2 MB

Table 2. Common Address Space Regions

3.5.2 HPS Peripheral Region Address Map

The following table lists the slave identifier, slave title, base address, and size of each slave in the HPS peripheral region. The *Slave Identifier* column lists the names used in the HPS register map file.

Slave Identifier	Slave Title	Base Address	Size
STM	STM		
DAP	DAP		
LWFPGASLAVES	FPGA slaves accessed with lightweight HPS-to-FPGA bridge		
LWHPS2FPGAREGS	Lightweight HPS-to-FPGA bridge GPV		
HPS2FPGAREGS	HPS-to-FPGA bridge GPV		
FPGA2HPSREGS	FPGA-to-HPS bridge GPV		
EMAC0	EMAC0		
EMAC1	EMAC1		
SDMMC	SD/MMC		
QSPIREGS	Quad SPI flash controller registers		
FPGAMGRREGS	FPGA manager registers		
ACPIDMAP	ACP ID mapper registers		
GPIO0	GPIO0		
GPIO1	GPIO1		
GPIO2	GPIO2		
L3REGS	L3 interconnect GPV		
NANDDATA	NAND controller data		
QSPIDATA	Quad SPI flash data		
USB0	USB0 OTG controller registers		
USB1	USB1 OTG controller registers		
NANDREGS	NAND controller registers		
FPGAMGRDATA	FPGA manager configuration data		
CAN0	CAN0 controller registers		
CAN1	CAN1 controller registers		
UART0	UART0		
UART1	UART1		
I2C0	I2C0		
I2C1	I2C1		
I2C2	I2C2		
I2C3	I2C3		

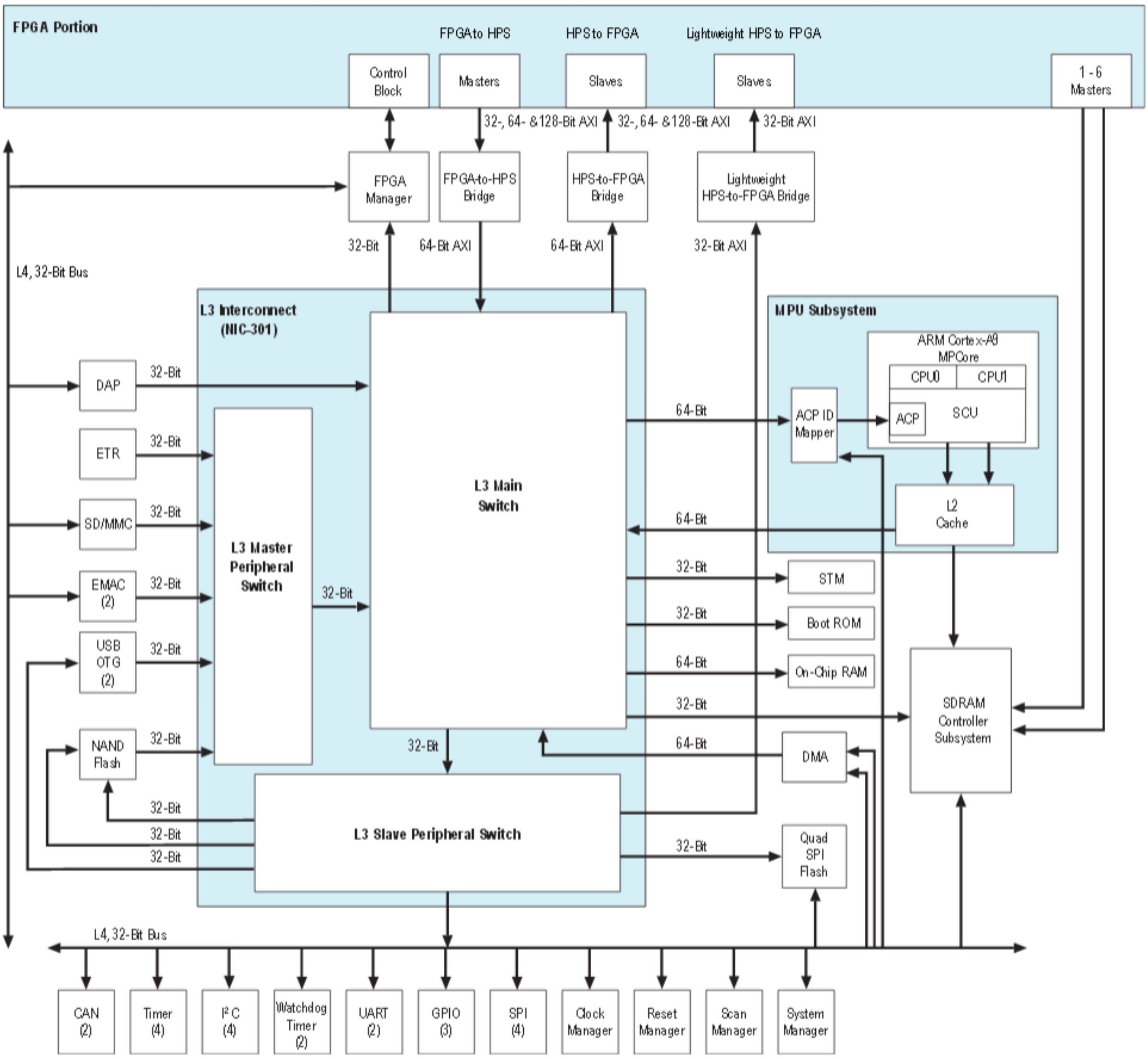
SPTIMER0	SP Timer0		
SPTIMER1	SP Timer1		
SDRREGS	SDRAM controller sub-system registers		
OSC1TIMER0	OSC1 Timer0		
OSC1TIMER1	OSC1 Timer1		
L4WD0			
L4WD1			
CLKMGR			
RSTMGR			
SYSMGR			
DMANONSECURE			
DMASECURE			
SPIS0			
SPIS1			
SPIM0			
SPIM1			
SCANMGR			
ROM			
MPUSCU			
MPUL2			
OCRAM			

Note: Can access the FPGA components by taking their offsets from the h2f interfaces.
Note: All macros for these peripherals can be found in hps.h, except for the heavyweight hps 2 fpga bridge.
Note: The lw hps2fpga bridge acts as a HPS peripheral, so it is accessible from the HPS peripheral region (coincidence that it is 32-bits, like the processor?).

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.



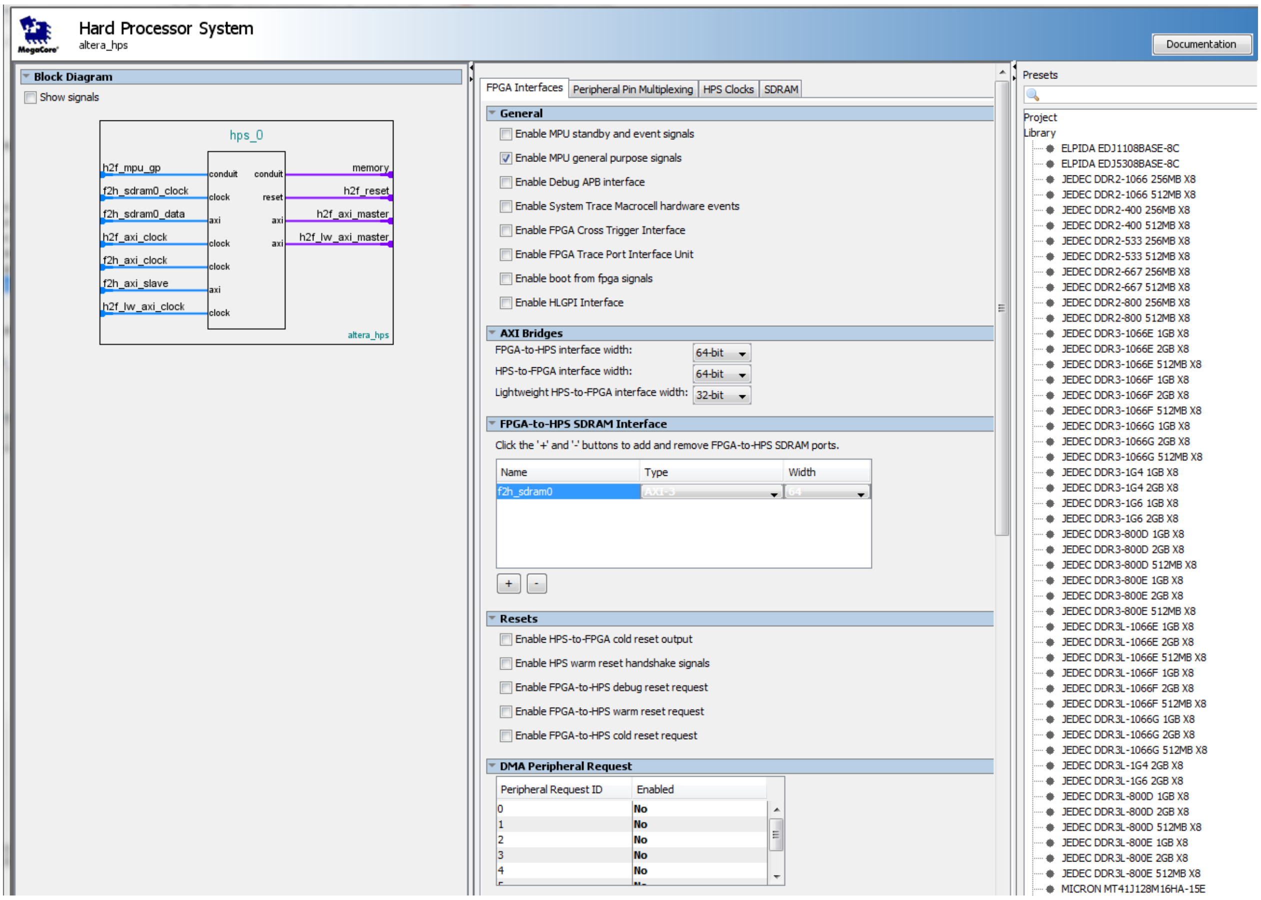
4.2 Hardware development

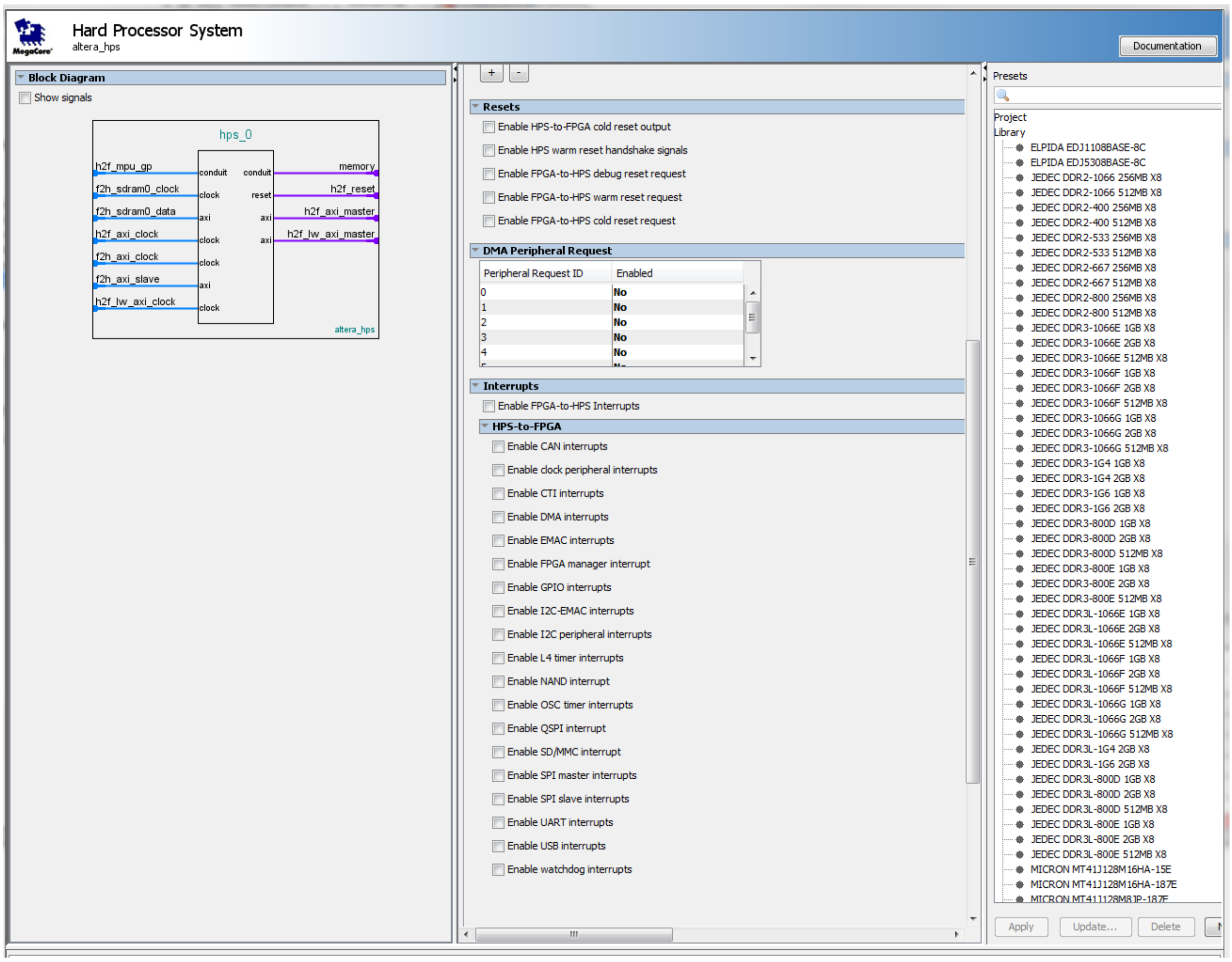
4.2.1 Qsys integration

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.





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

They 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.scat
	$(LD) hello.o -o $(TARGET) --cpu=Cortex-A9.no_neon.no_vfp --scatter=scatter.scat
```

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		
LWFPGASLAVES		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
- Cylone 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=2NBcUv2TxbI
- 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=cWIaqt2RU84 (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

List of Figures