Embedded Systems

CycloneV & DE0-nano-SoC / DE1-SoC

René Beuchat

Laboratoire d'Architecture des Processeurs

rene.beuchat@epfl.ch

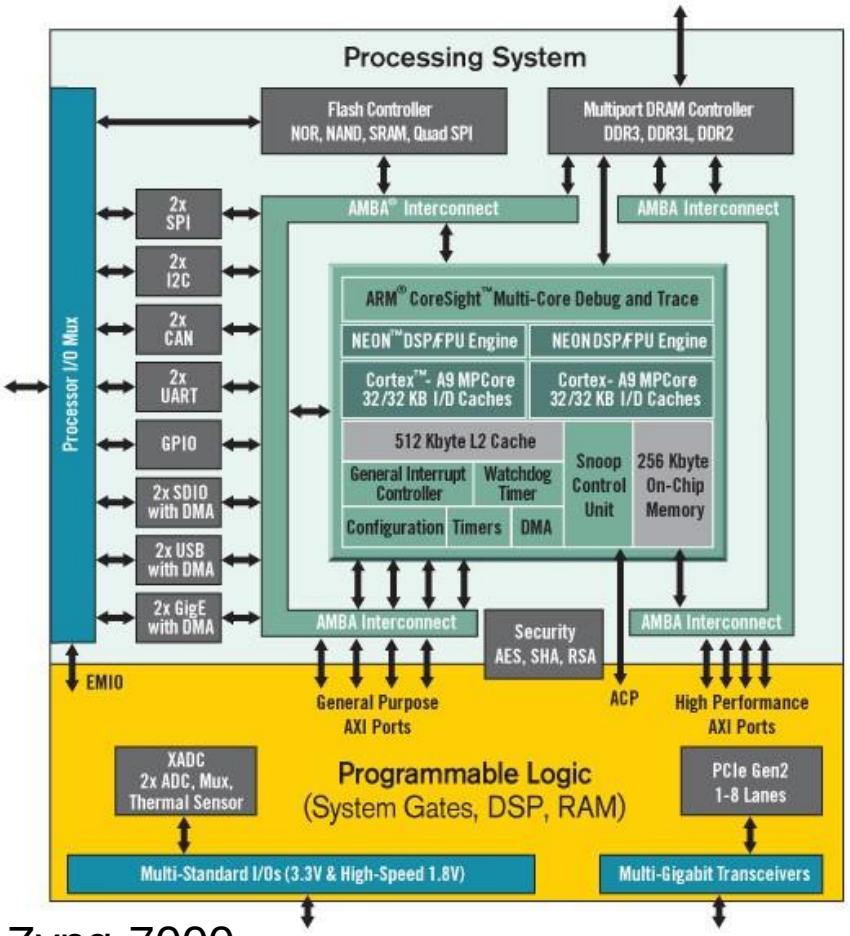
FPGA WITH SOC ARCHITECTURE

2 main actors

- IntelFPGA (Altera (www.altera.com))
  - Cyclone V SOC, Cyclone 10
  - Arria V SOC, Arria 10
  - Stratix 10
- Xilinx (www.xilinx.com):
  - Zynq® 7000 family
  - Zynq UltraScale+ MPSoC

2 main actors, Common Features

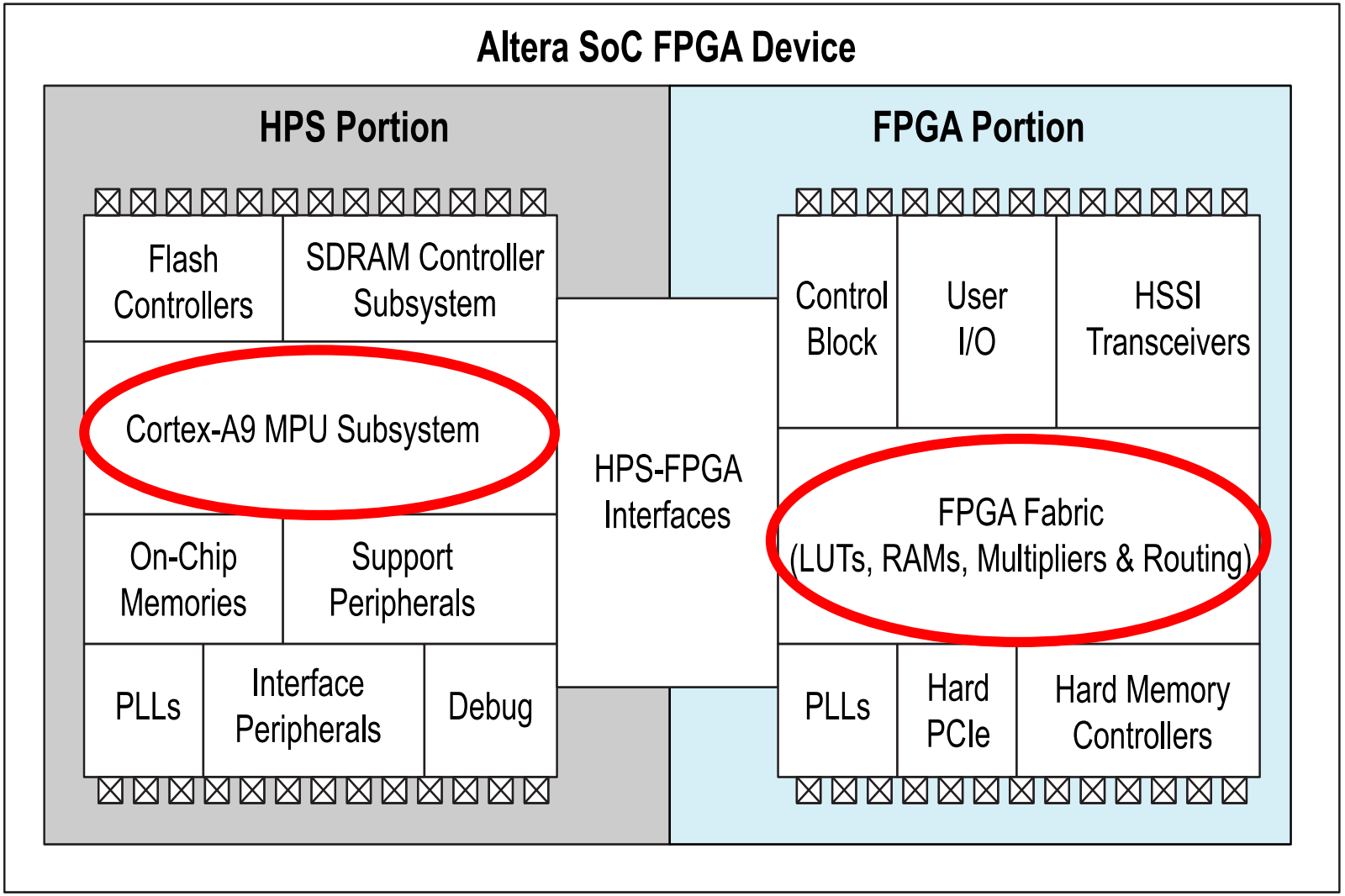
- 2x ARM-Cortex A9 hardcore
  - 2x NEON DSP/FPU
  - Many programmable interface in hardcore
  - Amba interconnect
  - Large FPGA part
  - DDR Controller



Ex: Zynq-7000

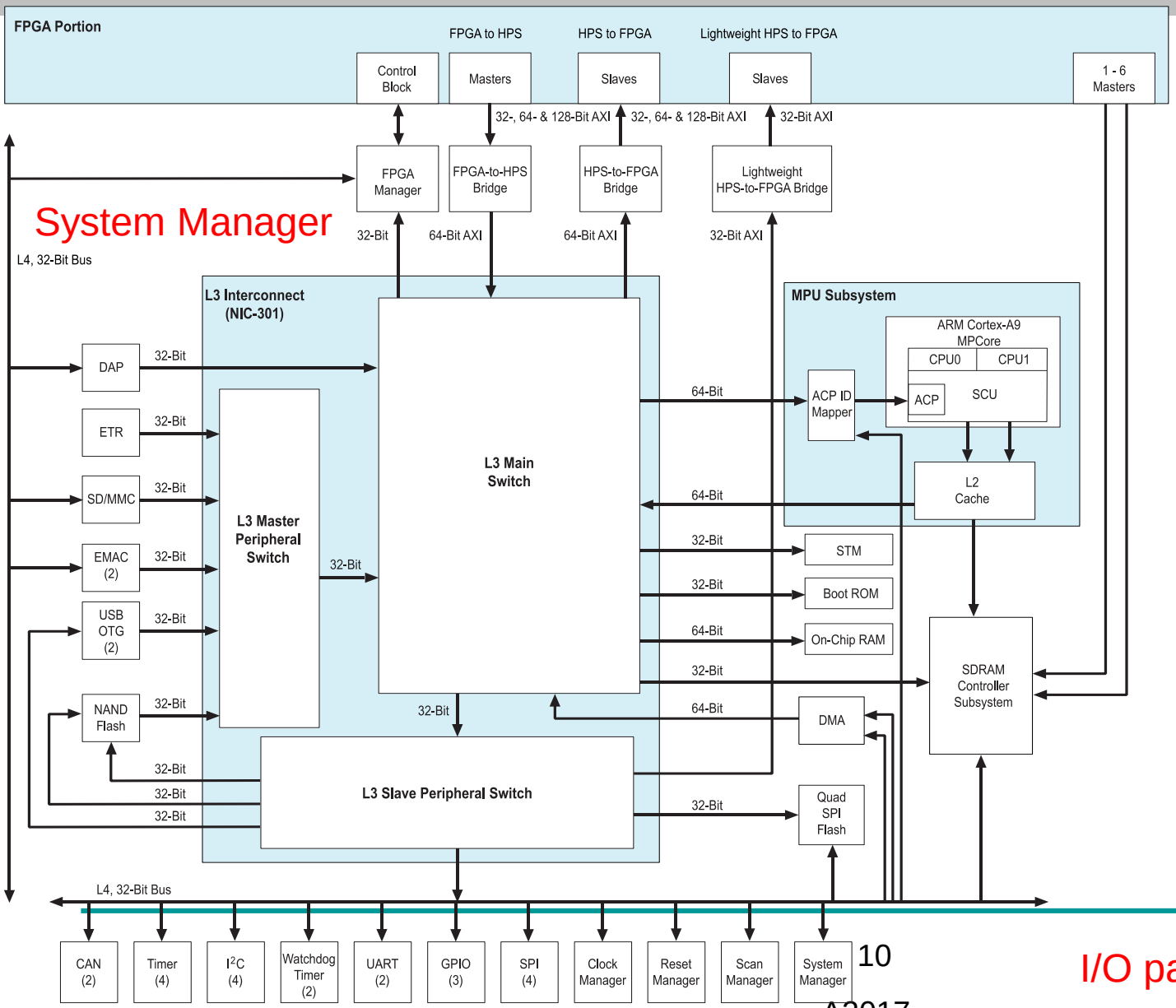
CYCLONE V-SOC ARCHITECTURE (INTELFPGA)

Cyclone V SoC Overview



http://www.altera.com/literature/hb/cyclone-v/cv_5v4.pdf

Cyclone V HPS (Hard Processor System)

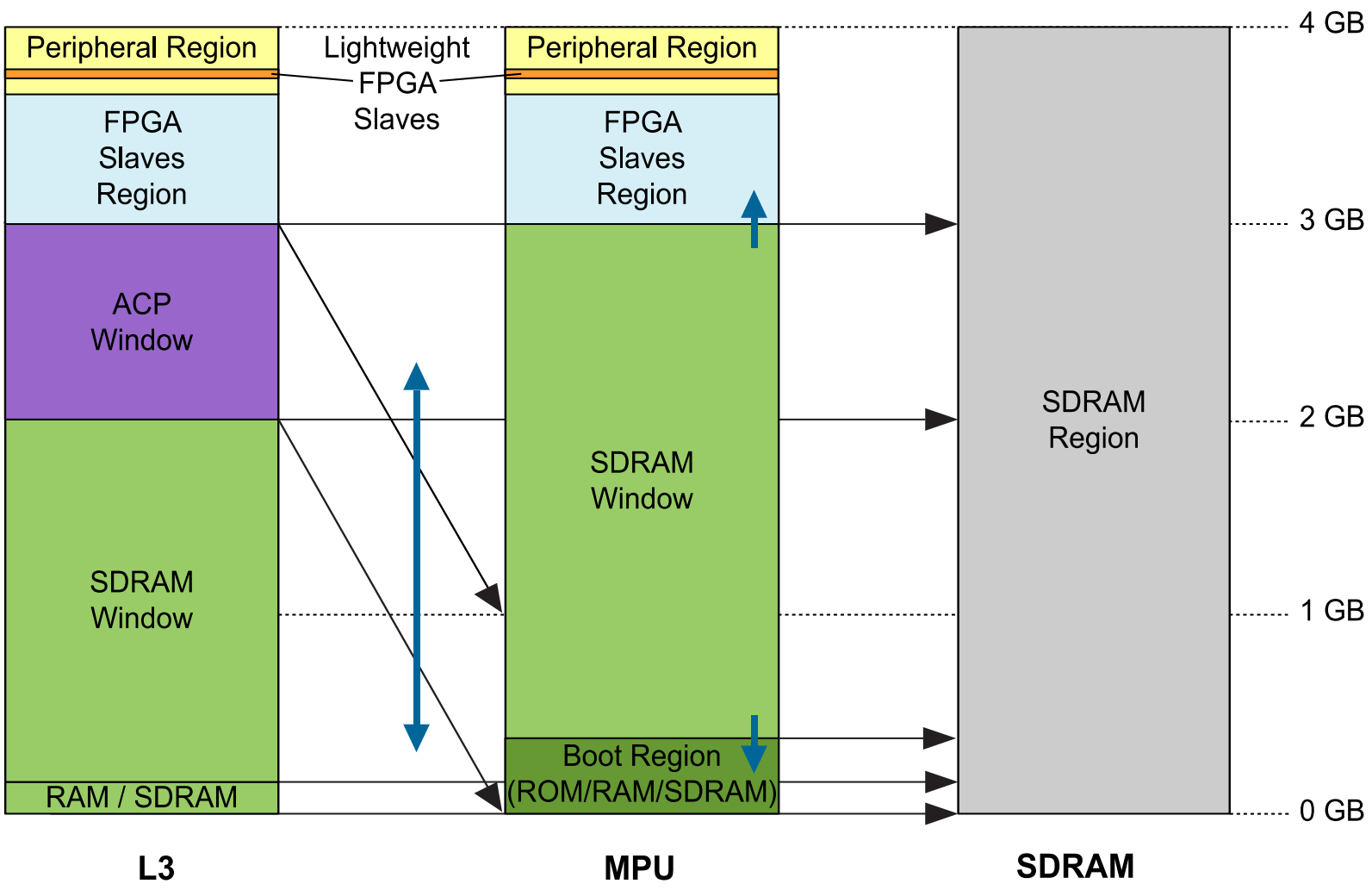


FPGA part

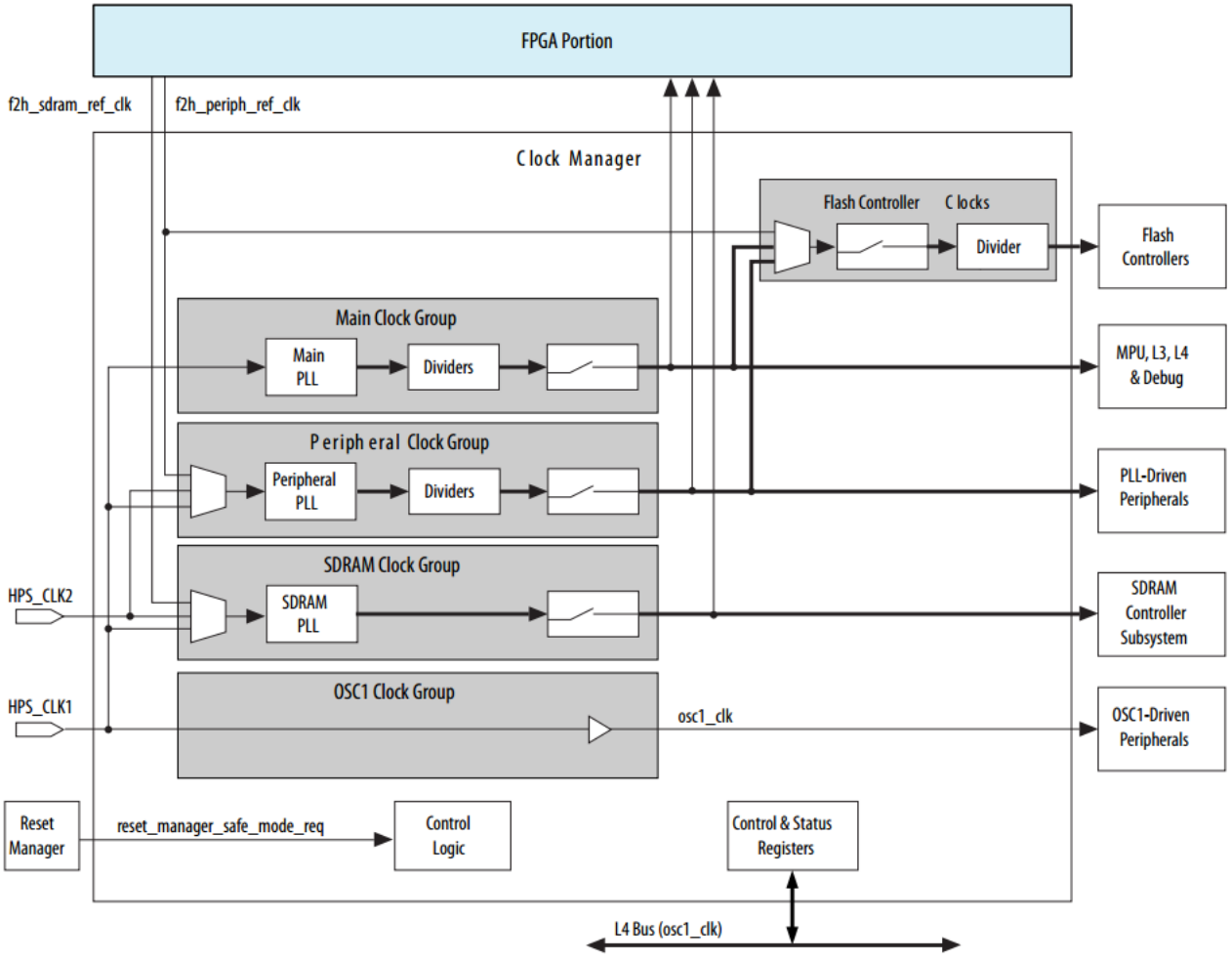
I/O part

A2017

HPS-FPGA Address Space



Clock manager



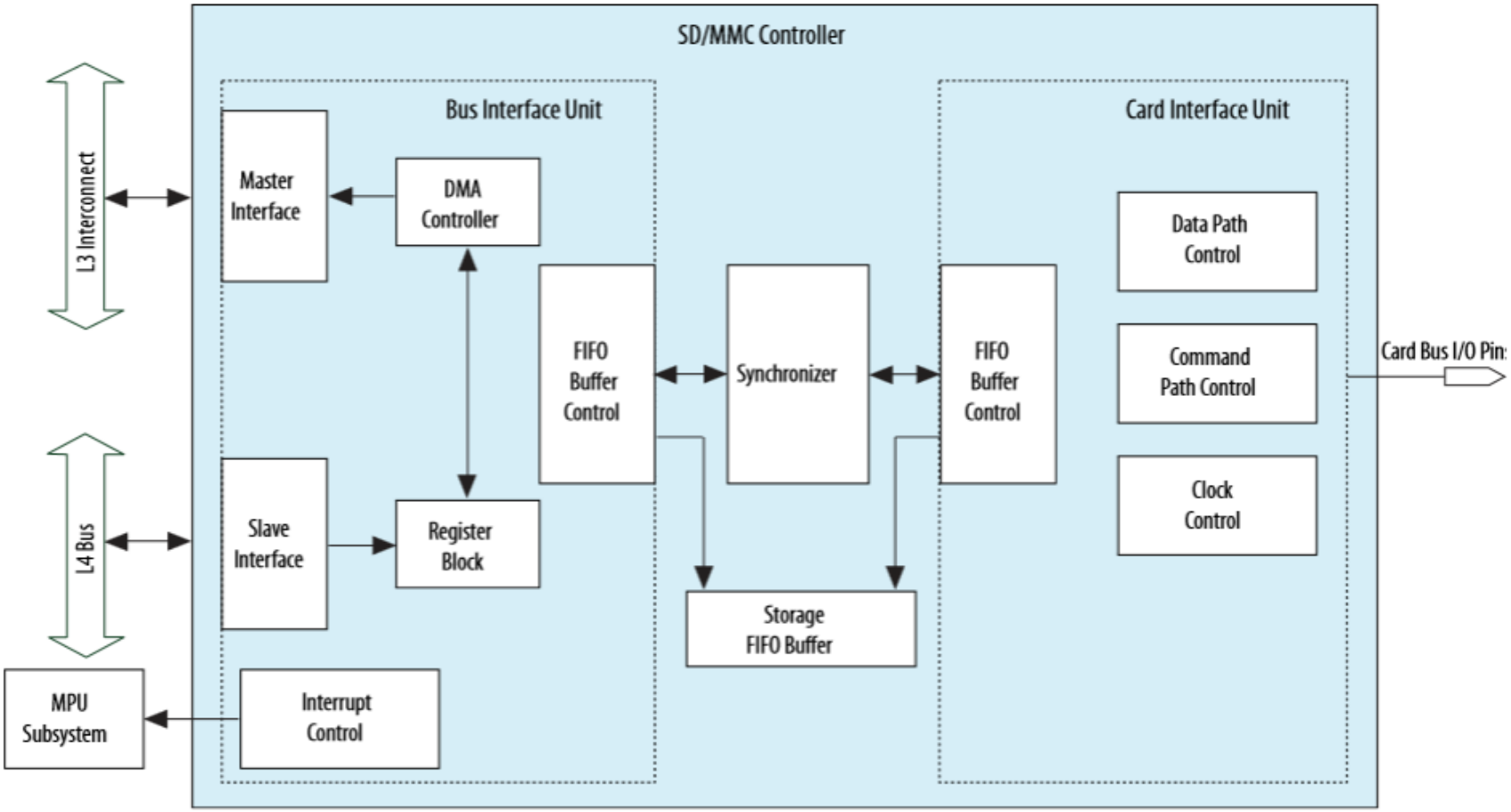
Abbreviation

- STM System Trace Module
- DMA Direct Memory Access
- DAP Debug Access Port
- ETR Embedded Trace Router
- SD/ Supporte: SDSC(SD), SDHC, SDXC, eSD, SDIO, eSDIO
- MMC MMC, RSMMC, MMCPlus, MMCMobile, eMMC
- EMAC Ethernet Media Access Controller

Abbreviation

- ACP Accelerator Coherency Port
- USB Universal Serial Bus
- UART Universal Asynchronous Receiver-Transmitter
- SPI Synchronous Peripheral Interface
- CAN Controller Area Network
- I2C Inter-Integrated Circuit

Ex. Programmable Interface SD/MMC Unit

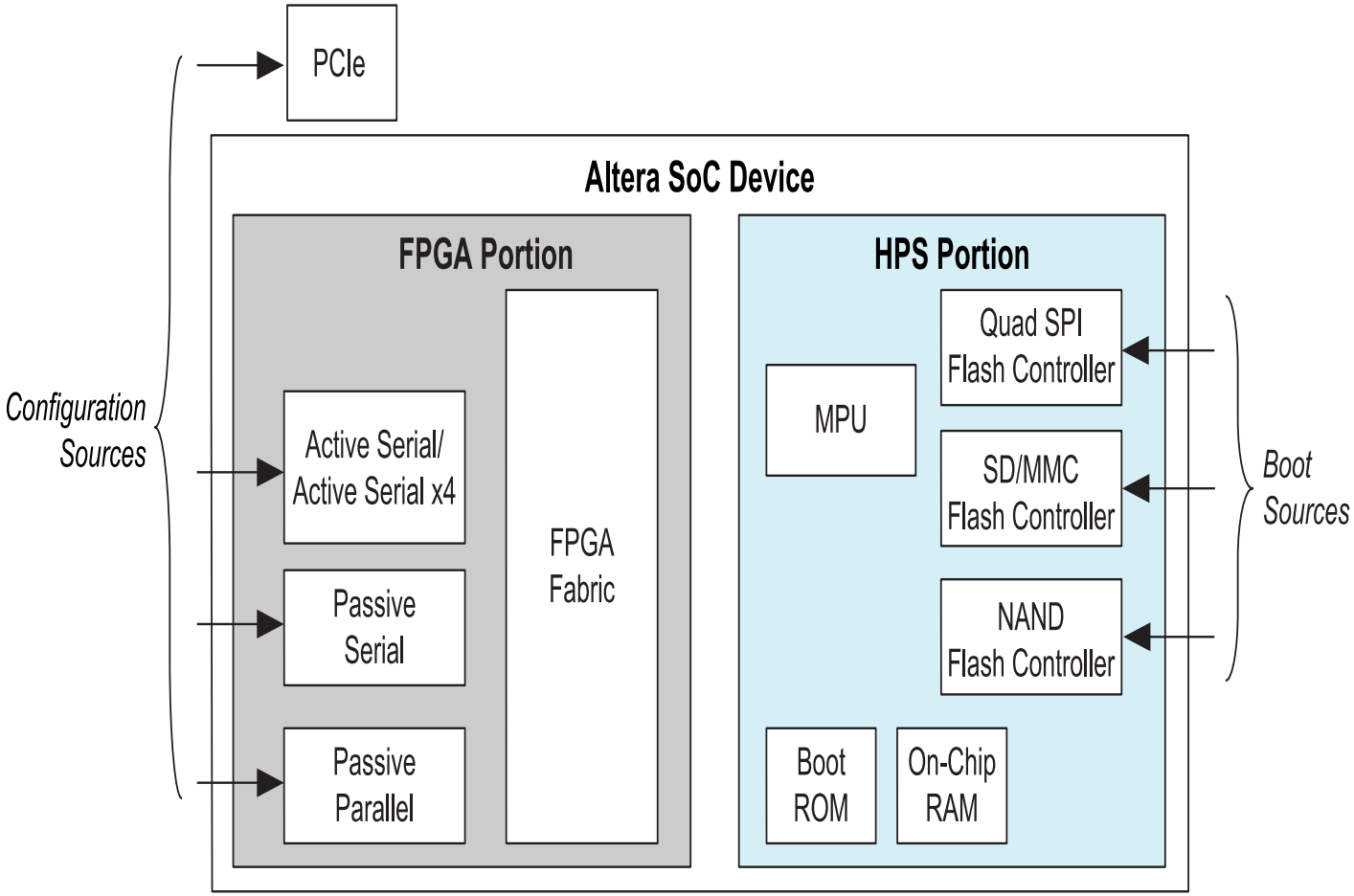


Boot process

- It is possible to use the Cyclone V SoC in 3 different configurations:
  - FPGA-only
  - HPS-only
  - HPS & FPGA
- The configurations using the HPS are more difficult to set up than the *FPGA-only* one.

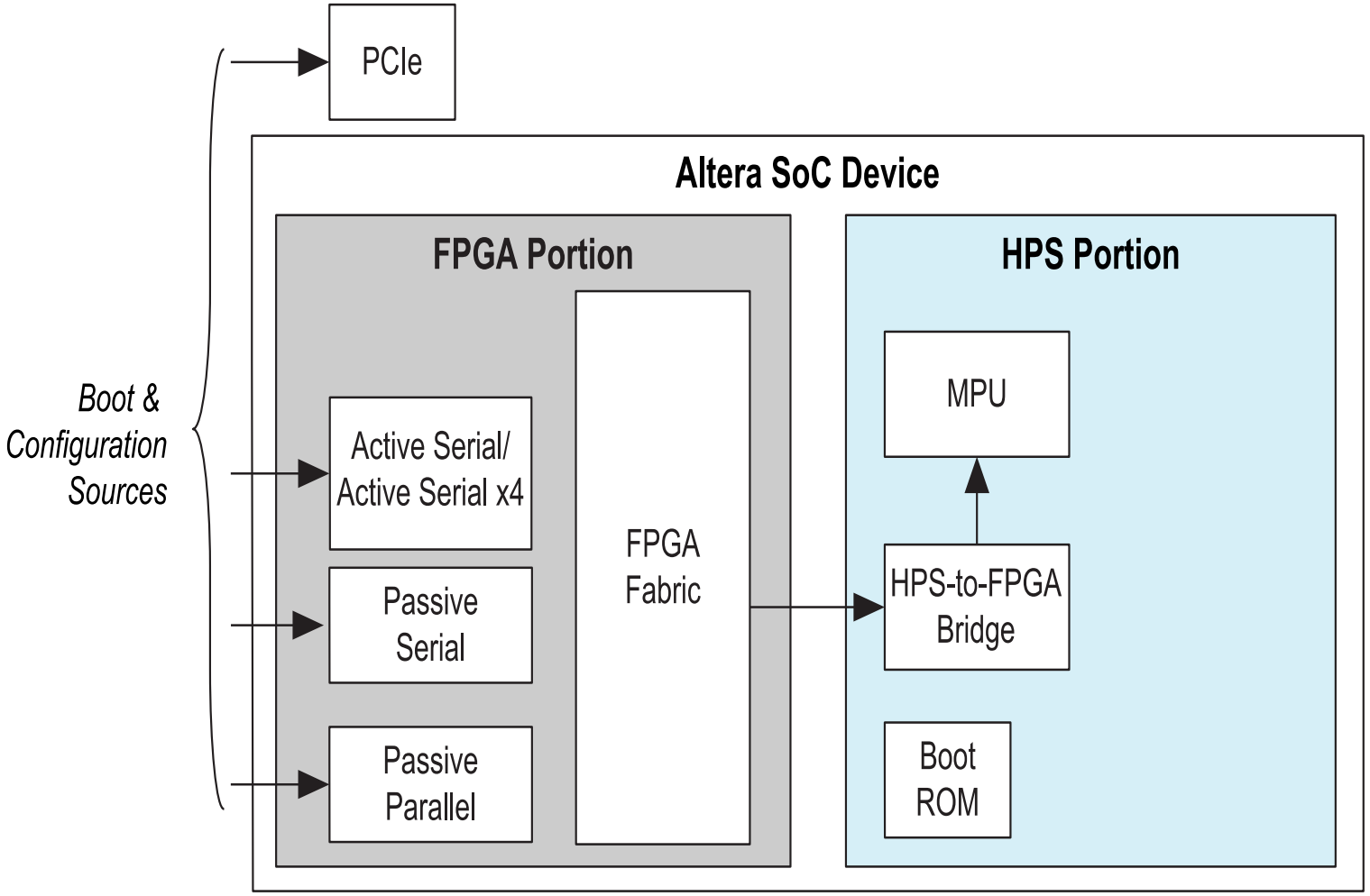
HPS/FPGA Boot (1)

Independent FPGA Configuration and HPS Booting



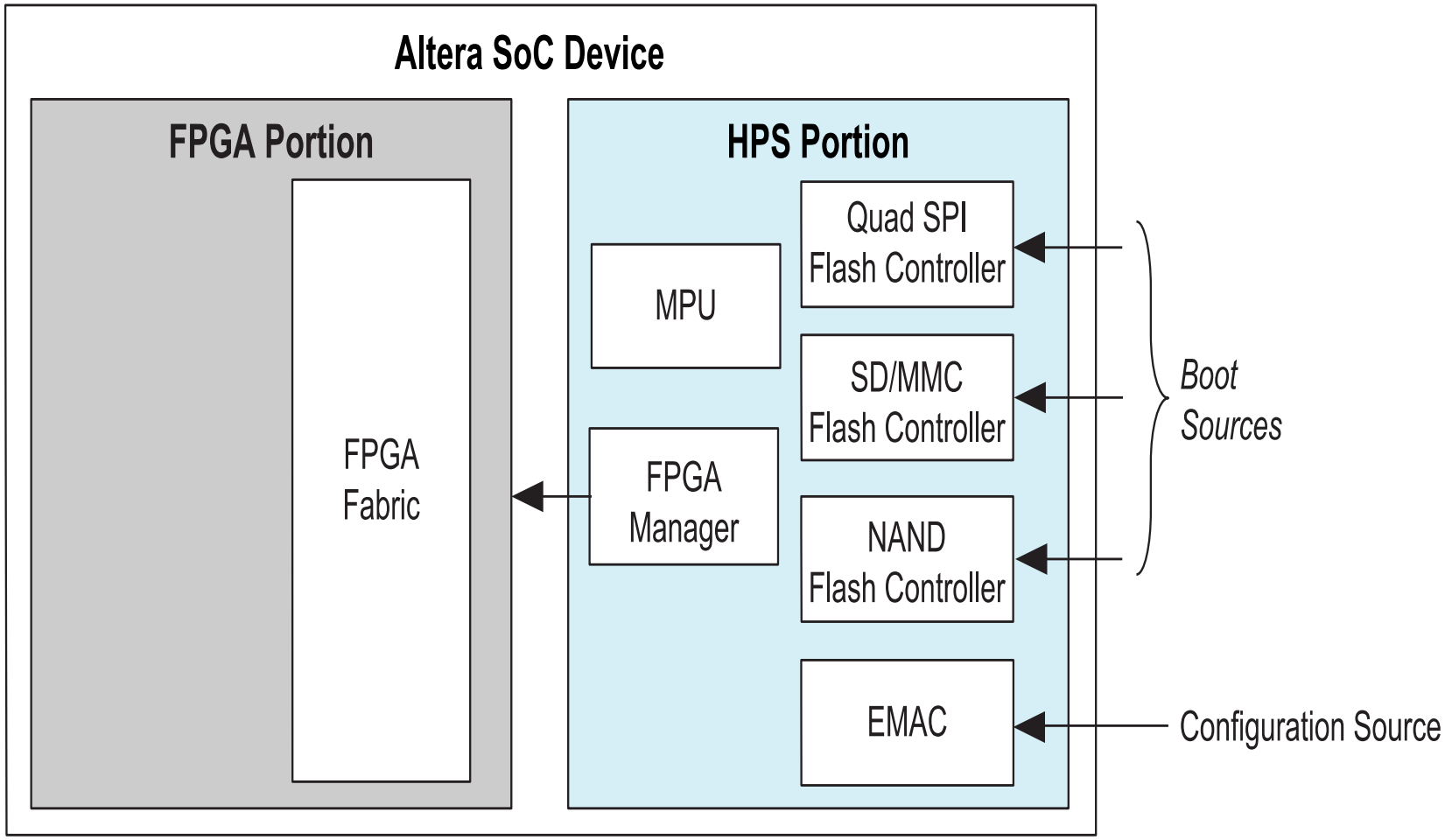
HPS/FPGA Boot (2)

FPGA Configuration before HPS Booting
(HPS boots from FPGA)



HPS/FPGA Boot (3)

HPS Boots and Performs FPGA Configuration



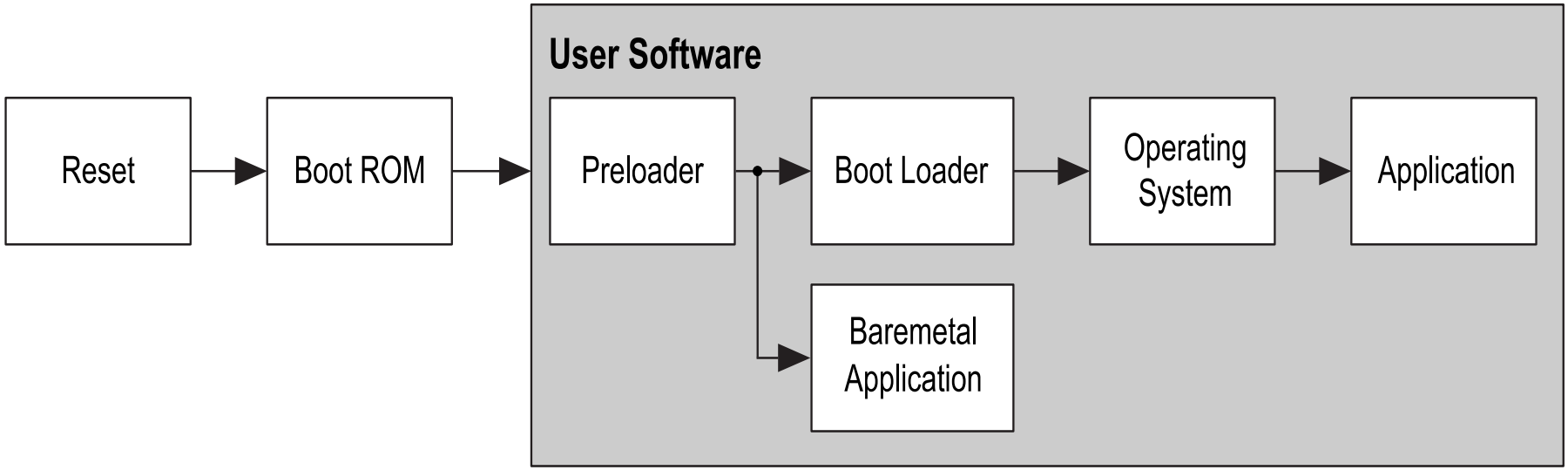
FPGA only case

- Exclusively using the FPGA part of the Cyclone V is easy, as the design process is identical to any other Altera FPGA.
- We can build a complete design in *Quartus II & Qsys*, simulate it in *ModelSim-Altera*, then program the FPGA through the *Quartus II Programmer*.
- We can instantiate a Nios II processor in *Qsys,* we can use the *Nios II SBT* IDE to develop software for the processor.

Type of Application

- OS based (ie: Linux)
- Bare-metal (No OS)

HPS Boot Flows



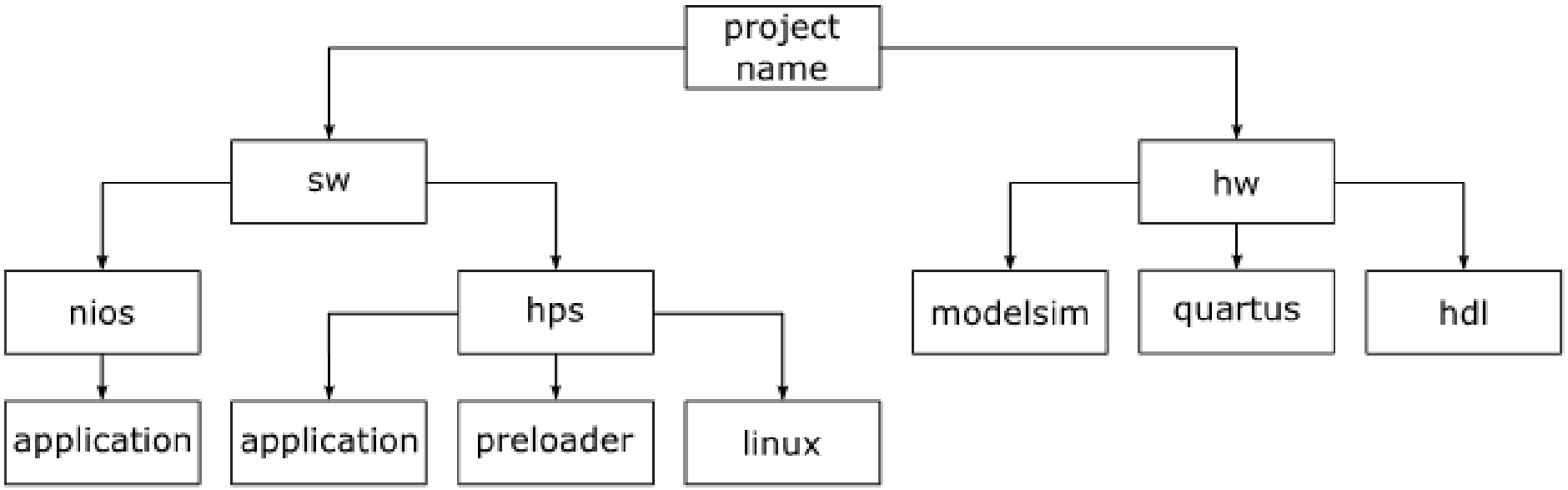
*Although the HPS has a **DUAL**-processor,*
CPU1 is under reset, and the boot flow only executes on CPU0.

If we want to use both processors,
*then **USER SOFTWARE** executing on CPU0 is responsible*
for releasing CPU1 from reset

Preloader

- The preloader is one of the most important boot stages. It is actually what one would call the boot "*source*", as **all stages before it are unmodifiable**. The preloader can be stored on external flash-based memory, or in the FPGA fabric.
- The preloader typically performs the following actions:
  - Initialize the SDRAM interface
  - Configure the HPS I/O through the scan manager
  - Configure pin multiplexing through the system manager
  - Configure HPS clocks through the clock manager
  - Initialize the flash controller (NAND, SD/MMC, QSPI) that contains the next stage boot software
  - Load the next boot software into the SDRAM and pass control to it
- The preloader does ***NOT*** release CPU1 from reset. The subsequent stages of the boot process are responsible for it if they want to use the extra processor.

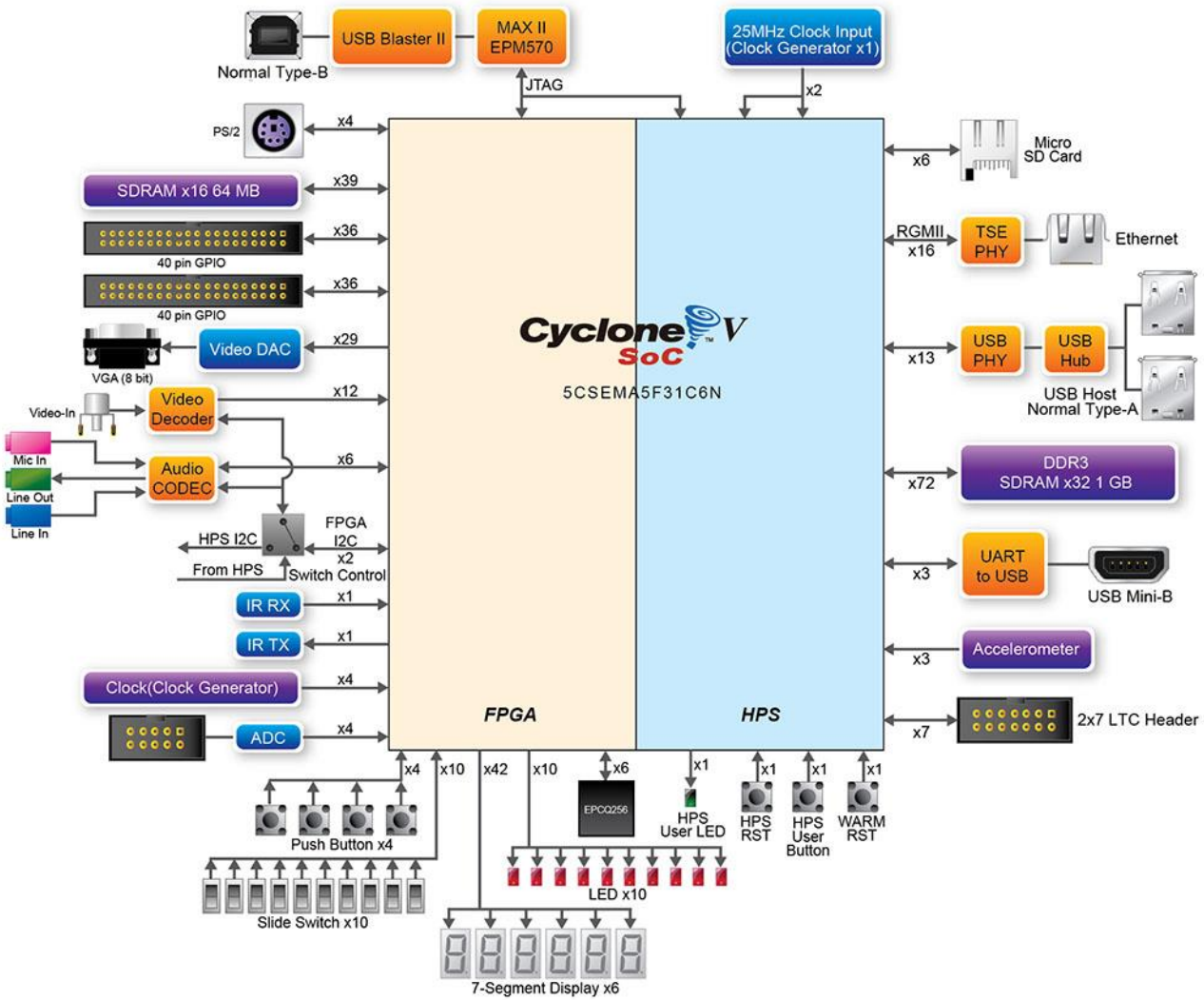
Project structure



DE1-SOC BOARD (TERASIC)

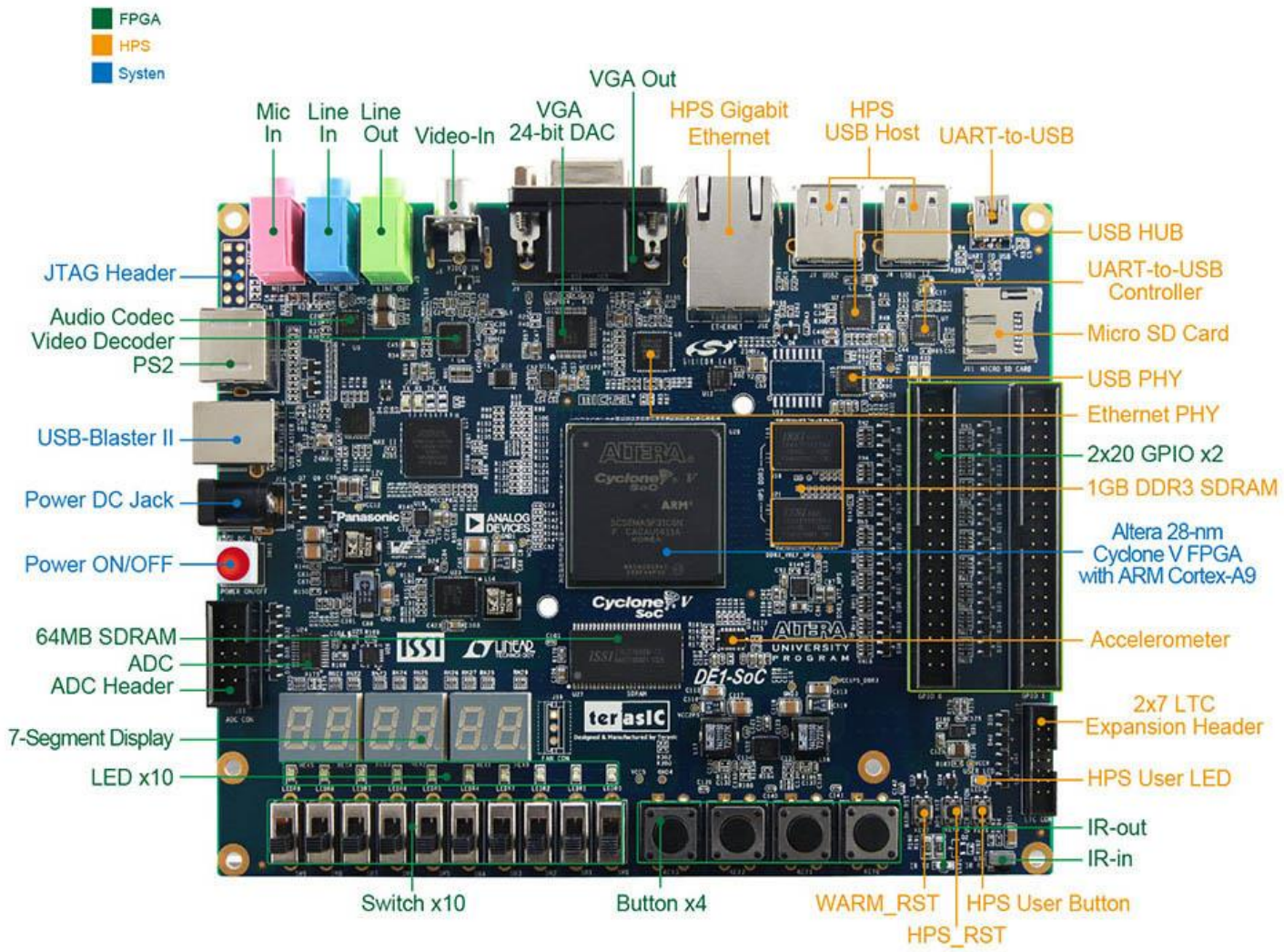
www.terasic.com.tw

DE1-SOC Bloc Diagramm

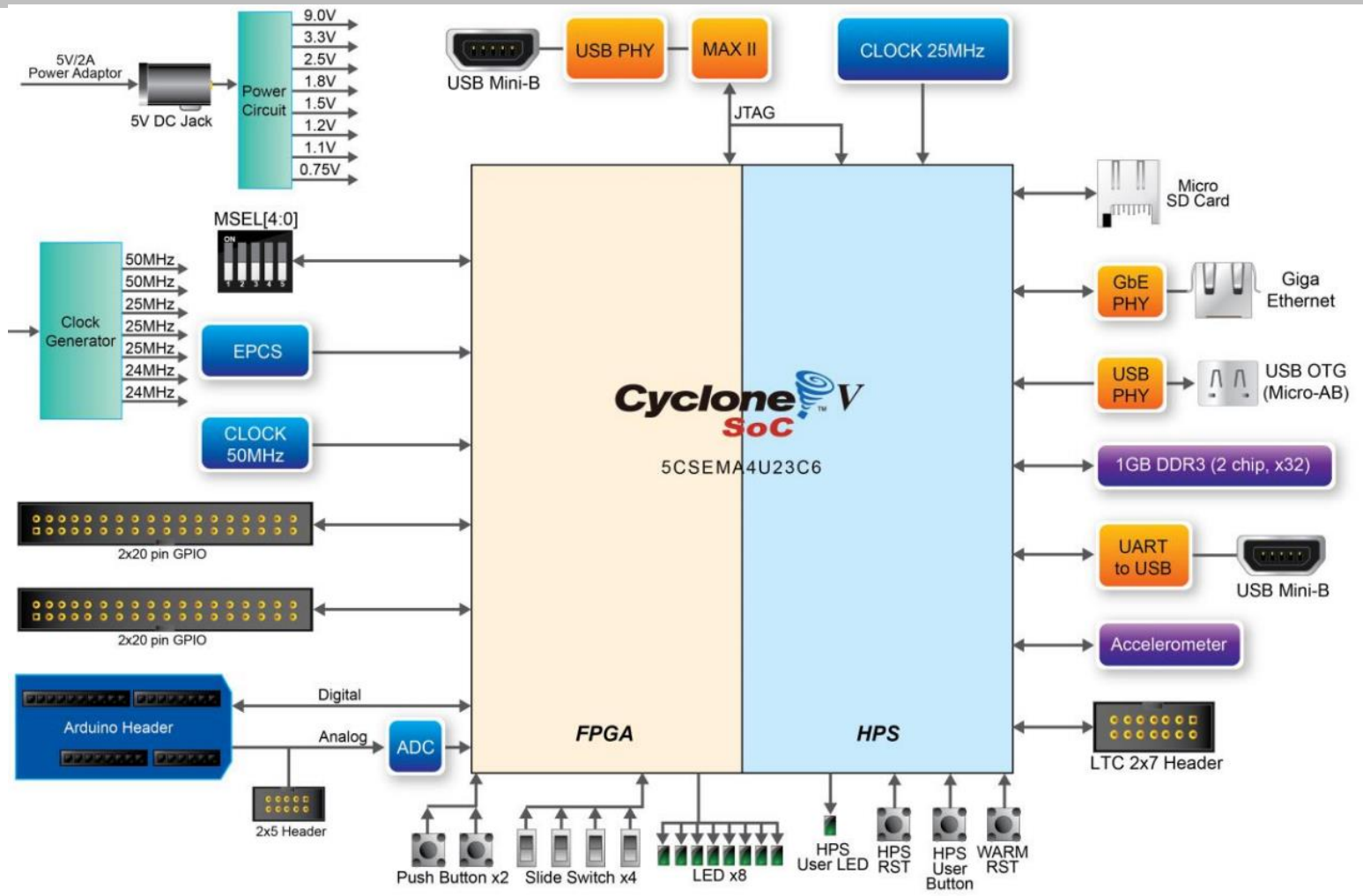


DE1-SoC

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

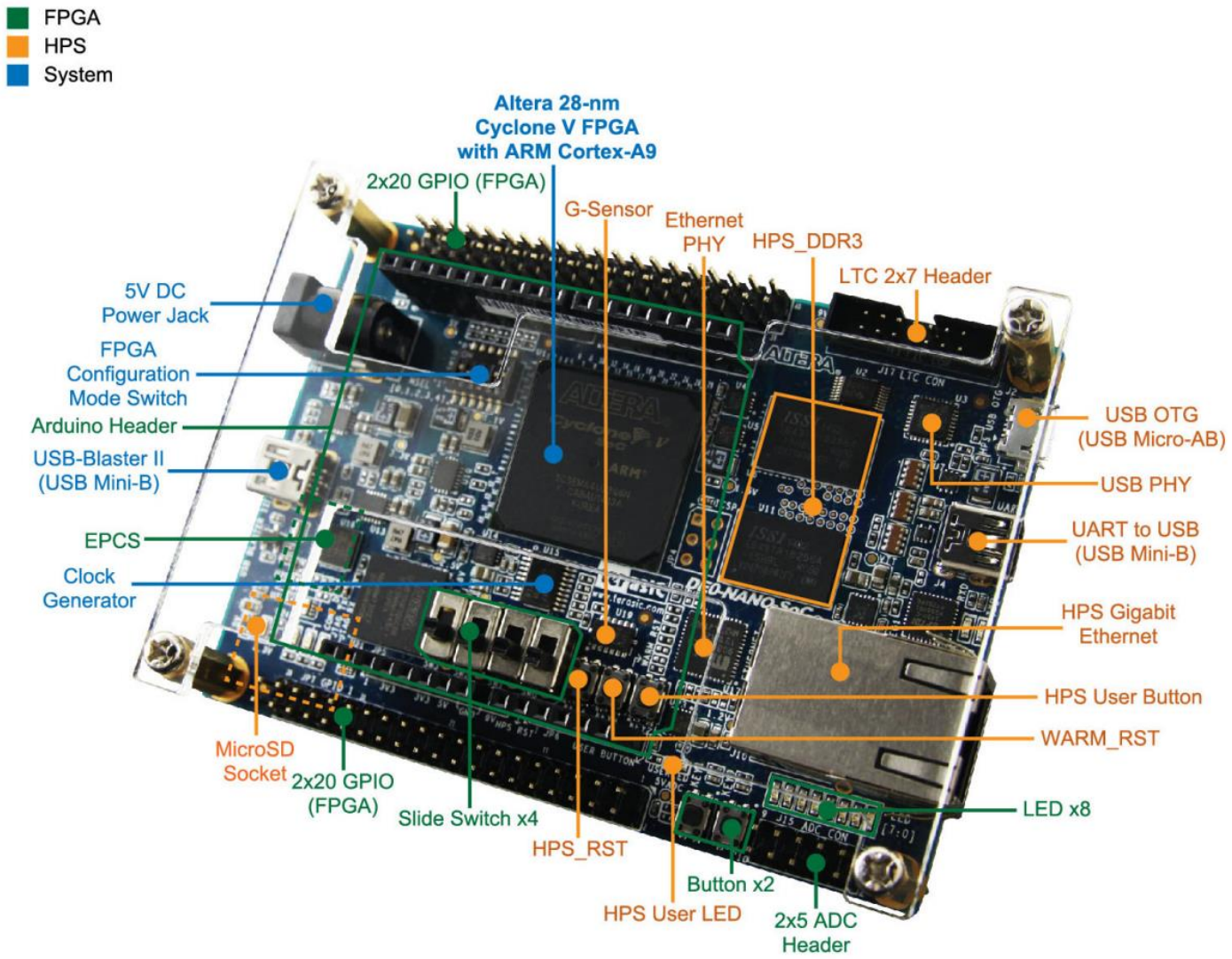


DE0-nano-SoC

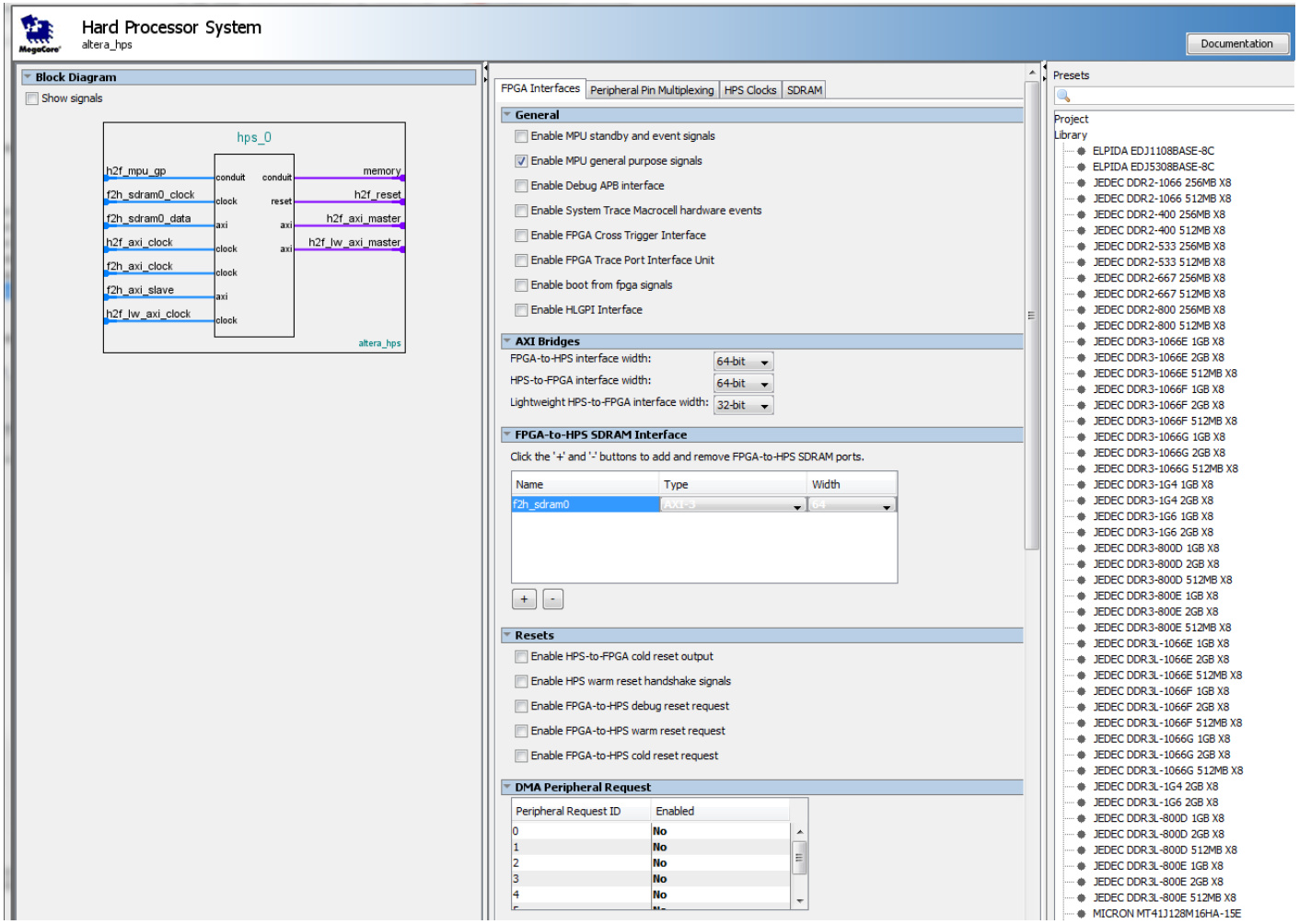


http://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual_rev.C1.pdf

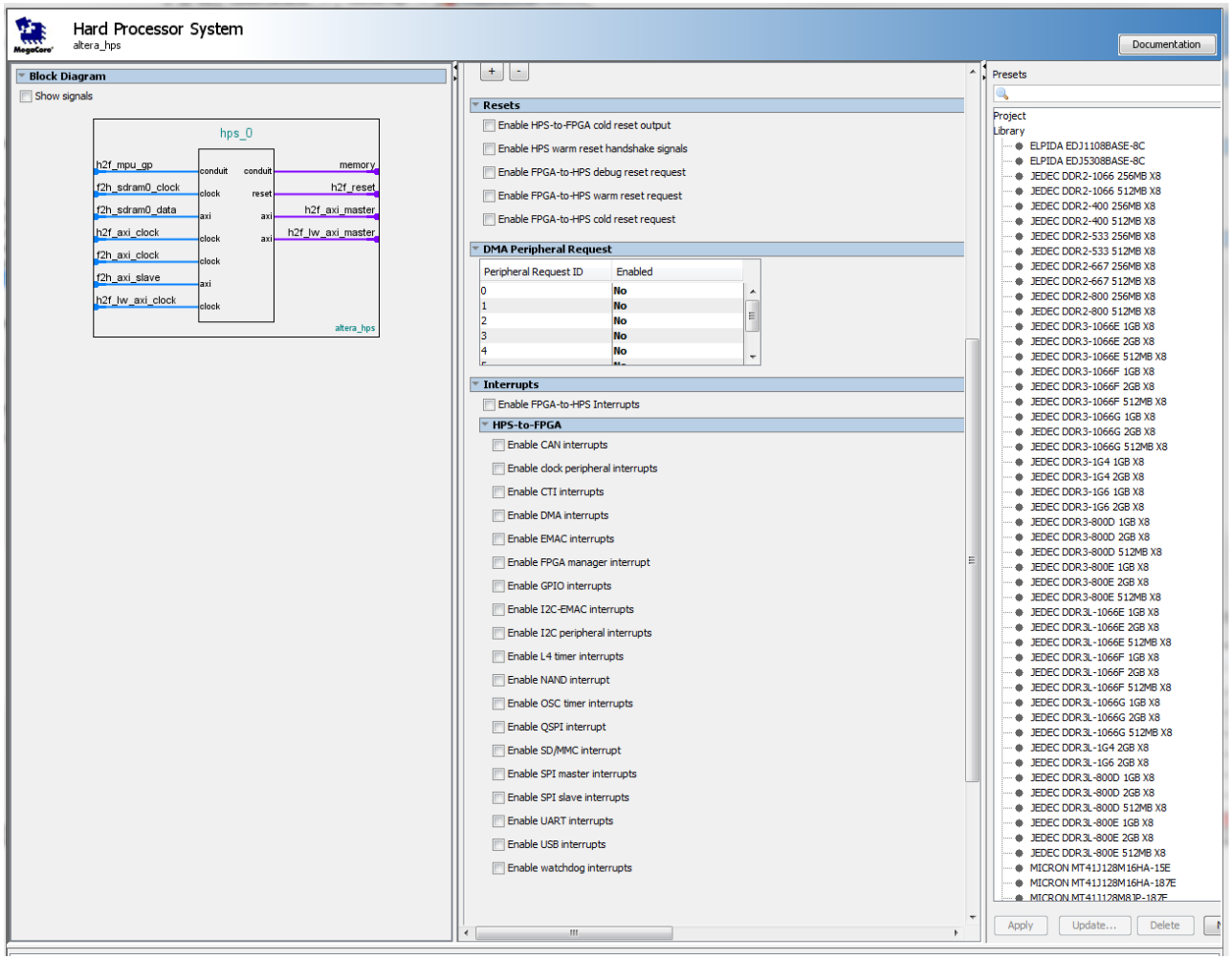
DE0-nano-SoC



Qsys, hps definition (1)



Qsys, hps definition (2)



IO PIN in HPS

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

Exercises / Mini Project

1. Use the DE1-SOC/DE0-nano-SoC without the ARM-A9
   - NIOS design to access the Switches and LEDs
   - Adapt the LCD/camera controller for the NIOSII

2. Use the ARM-A9 with ARM DS-5 software
   - Access through the AXI bridge the Avalon part of the FPGA
   - Control the LCD/camera from the ARM

- Try the Linux access of the FPGA…to control LCD and Camera

- In option !