

Unité : SOCF

Architecture SoC-FPGA

Carte DE1 & device Cyclone-V SoC

heig-**vd**

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

www.heig-vd.ch

Etienne Messerli

février 2021

REDS

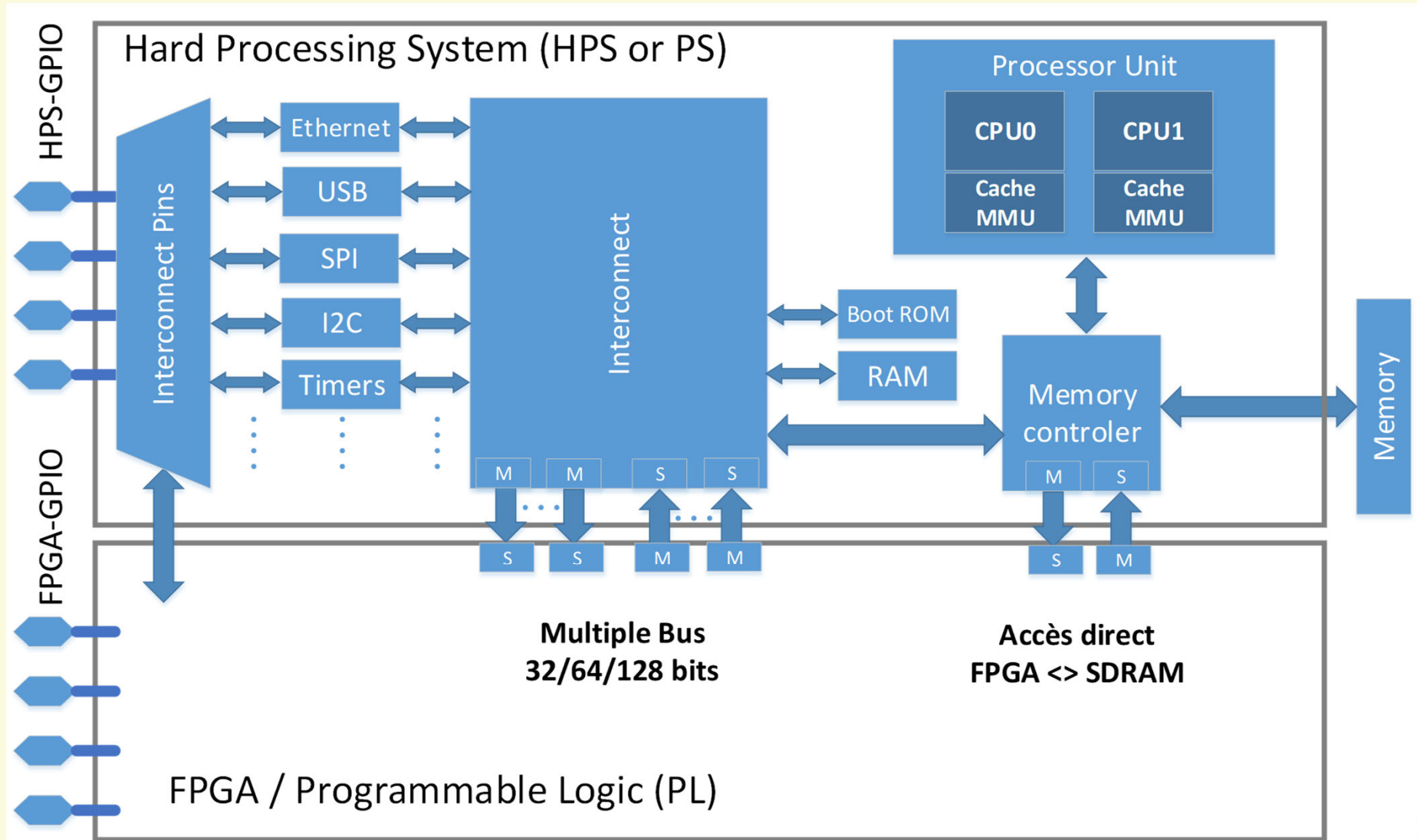


This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Contenu

- Généralité sur les SoC
 - architecture, utilisation, démarrage
- Carte DE1-SoC
- Cyclone-V SoC
 - Architecture
 - Etude de différents blocs et fonctionnalités
- Annexes:
 - Sources des images
 - Lexique des abréviations

Architecture SoC - FPGA



Utilisation SoC-FPGA

- Modes d'utilisation d'un SoC-FPGA :
 - FPGA only
 - HPS only
 - HPS & FPGA
- Configuration du système :
 - HPS
 - boot du processeur, evtl chargement OS
 - configuration des périphériques
 - FPGA
 - programmation avec un "bit stream"

1) Utilisation SoC-FPGA

- Pour le HPS, il y a plusieurs possibilités de fonctionnement, soit :
 - Sans OS (Bare metal)
 - Avec OS
 - OS de différente complexité
 - RTOS pour temps réel
 - Cadre du cours SoCF:
 - 1) Bare metal avec "Altera monitor programm"
 - 2) OS Linux

Démarrage du système

- Mode FPGA only
 - Configuration de la FPGA
 - Utilisation identique à une FPGA classic
 - Chargement du "bit stream" soit par une mémoire (flash memory), un processeur ou JTAG
- Mode HPS only
 - Nécessaire de prévoir une séquence de boot
 - A adapter selon l'OS utilisé et la source du code
 - Configuration des périphériques du HPS
 - via des registres par le programme
 - pas de programmation de la FPGA
- Mode HPS & FPGA
 - idem ci-dessus, plus programmation de la FPGA

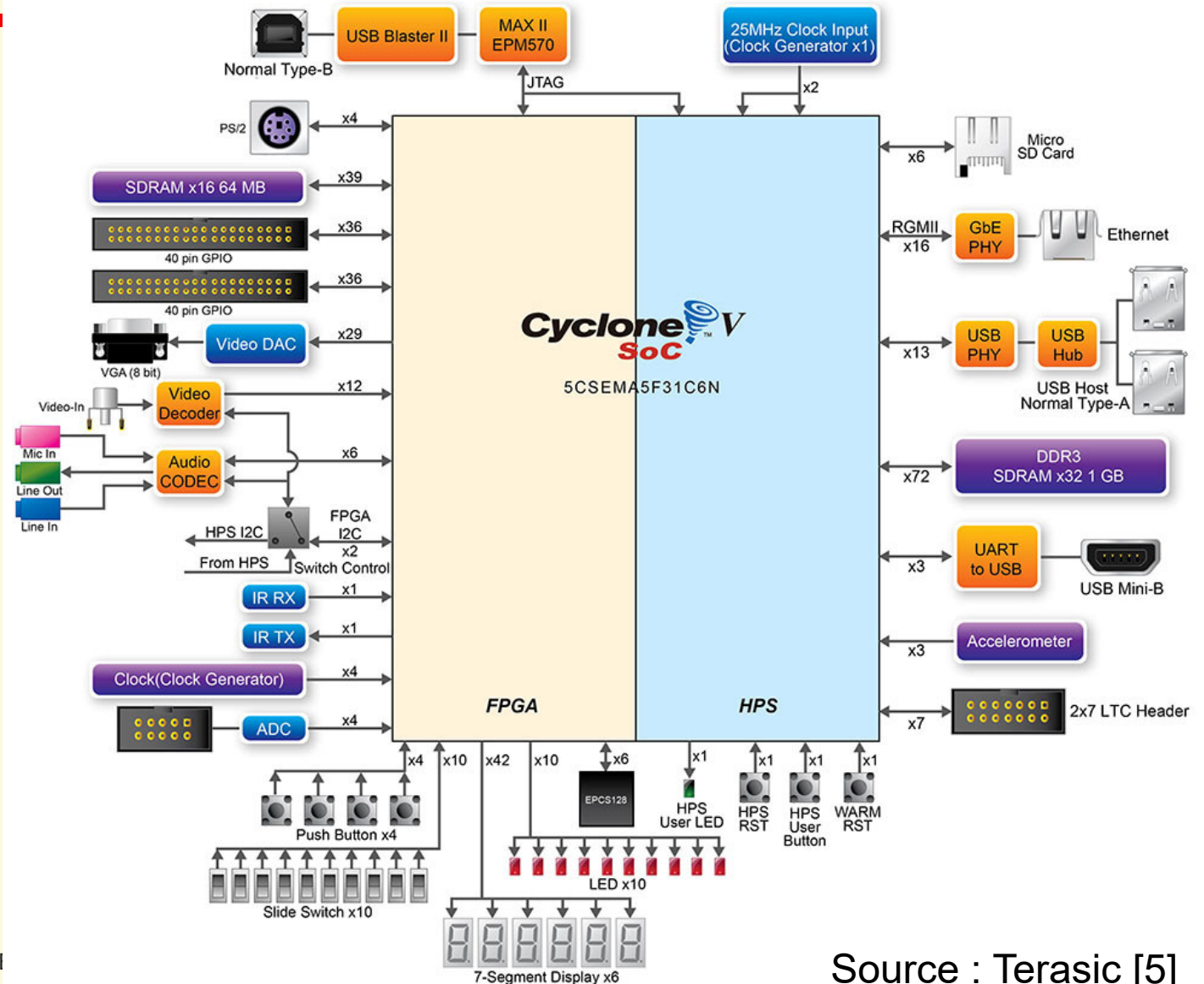
Carte DE1 SoC

- Cadre unité SoCF
 - Utilisation de la carte de développement Terasic DE1-SoC
 - Comprend un circuit Intel-Altera :
Cyclone-V SoC de type 5CSEMA5F31C6N
 - Celui-ci dispose de différents périphériques et IO qui seront utilisés pour les laboratoires

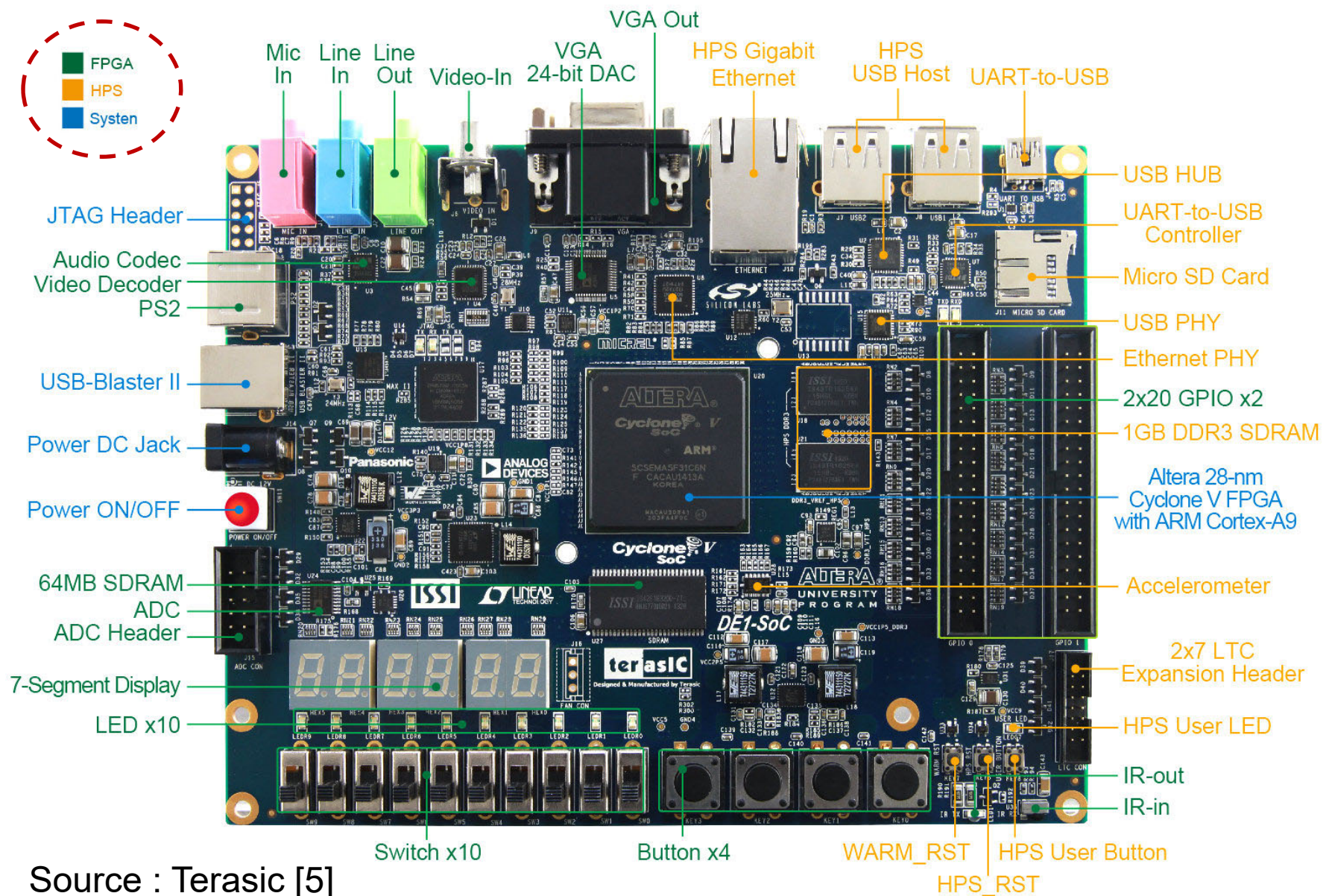
DE1-SoC Board

Block Diagram

A noter :
Il y a des IOs sur la FPGA et d'autres sur le HPS



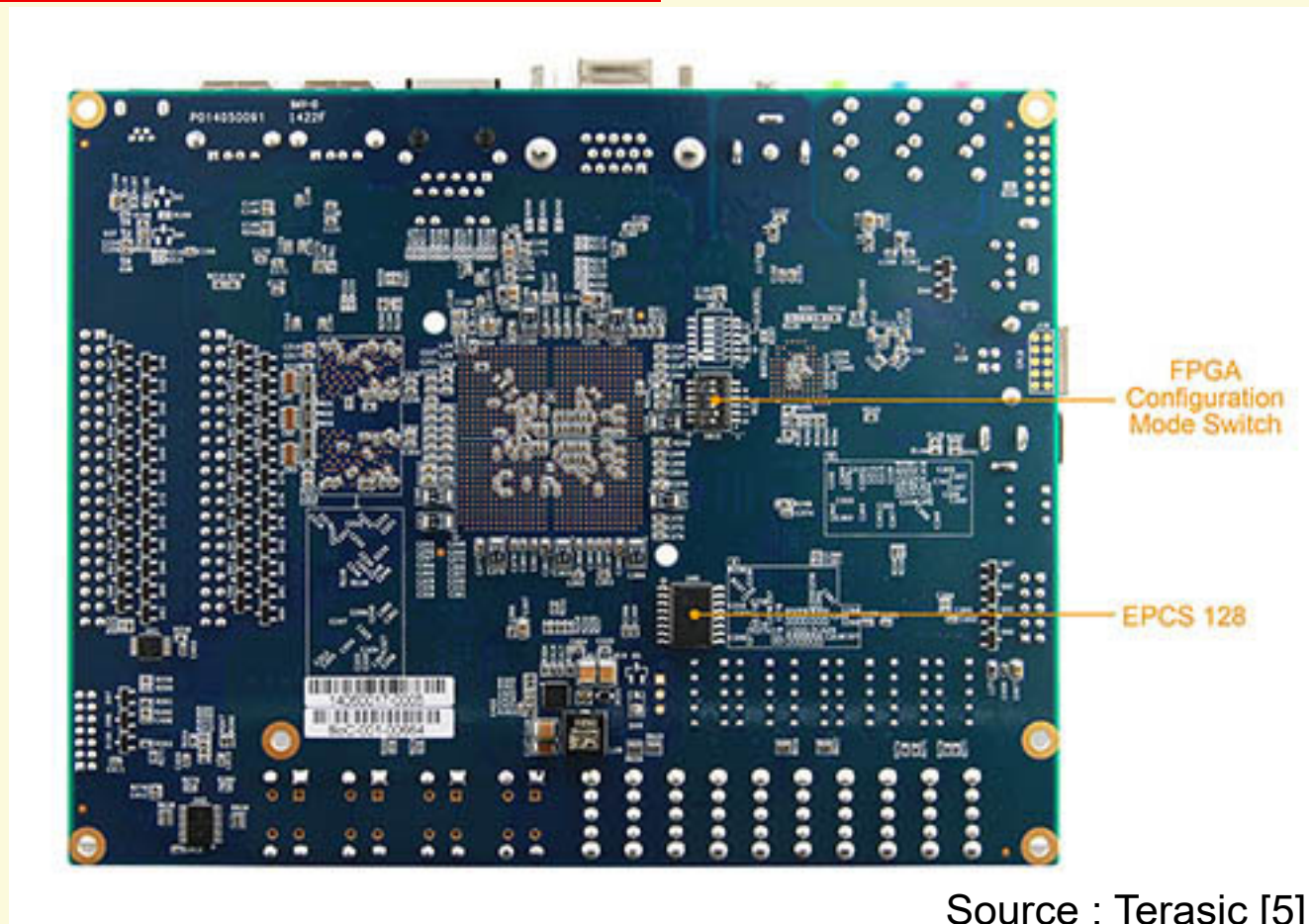
DE1-SoC Board, top



Source : Terasic [5]



DE1-SoC Board, bottom



Important : vérifier position des switches pour les laboratoire

Board DE1

Hardware provided on the board:

- Intel Cyclone® V SE 5CSEMA5F31C6N
 - Dual-core ARM Cortex-A9
- On the HPS side:
 - 1GB DDR3 SDRAM (32-bit data bus)
 - 1 Gigabit Ethernet PHY with RJ45 connector
 - 2-port USB Host, normal Type-A USB connector
 - Micro SD card socket
 - Accelerometer (I2C interface + interrupt)
 - UART to USB, USB Mini-B connector
 - Warm reset button and cold reset button
 - One user button and one user LED
 - LTC 2x7 expansion header

Board DE1

Hardware provided on the board:

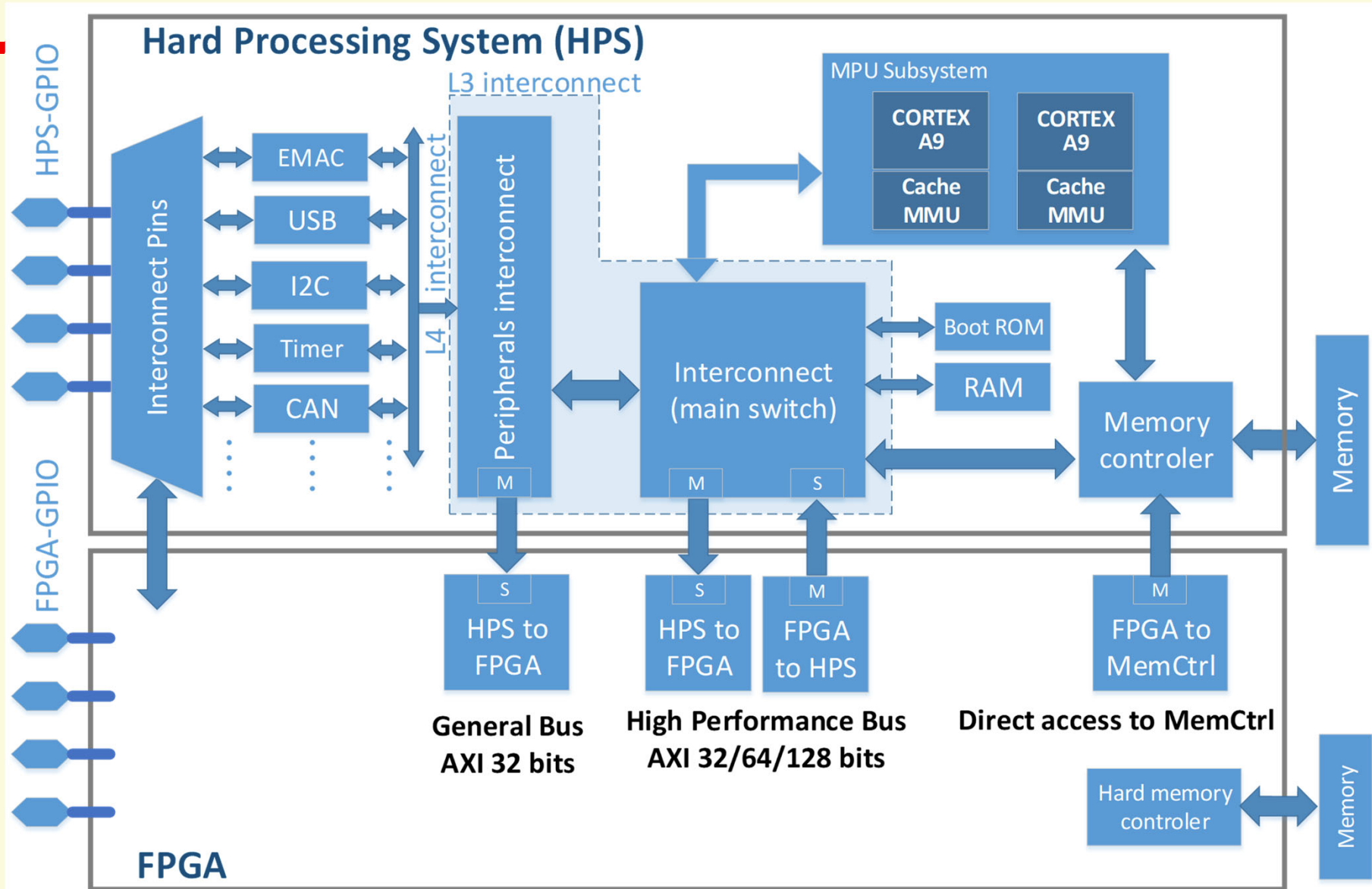
- Intel Cyclone® V SE 5CSEMA5F31C6N
- On the FPGA side:
 - 64MB SDRAM (16-bit data bus)
 - 4 push-buttons, 10 slide switches
 - 10 red user LEDs, Six 7-segment displays
 - Four 50MHz clock sources from the clock generator
 - 24-bit CD-quality audio CODEC (line-in, line-out, microphone-in)
 - VGA DAC (8-bit high-speed triple DACs) with VGA-out connector
 - TV decoder (NTSC/PAL/SECAM) and TV-in connector
 - PS/2 mouse/keyboard connector
 - IR receiver and IR emitter
 - Two 40-pin expansion header with diode protection
 - A/D converter, 4-pin SPI interface

FPGA Cyclone V SoC

Contenu :

- Etude de l'architecture spécifique de cette famille de SoC de Intel-Altera
- Connaitre les ressources disponibles pour chaque partie, soit HPS & FPGA
- Comprendre les interactions entre les deux monde soft <> hard

Architecture Cyclone-V SoC



Features of Cyclone V FPGA

Ressources Cyclone V FPGA, 5CSEMA5F31C6N

- 85K LE (logic elements), 32K ALM (Adaptive logic module)
- 128K flip-flops
- 3'970 Kbits embedded memory
- 87 DSP block
- 174 multiplier 18x18
- GPIO: 288 FPGA, 181 HPS
- PLLs : 6x FPGA, 3x HPS
- Hard memory controllers : 1x FPGA

Features of Cyclone V HPS ...

Le HPS comprend 3 catégories de blocs, soit:

- Main modules
 - Les CPUs et les mémoires
- General peripherals modules
 - Les divers périphériques disponibles
- The system modules
 - Systèmes de contrôle et de gestion pour faire fonctionner le HPS

... features of Cyclone V HPS ...

The main modules of the **HPS** are:

- MPU subsystem: Dual-core ARM Cortex-A9
 - ARM NEON SIMD coprocessor
 - Memory Management Unit (MMU)
 - Cache for Instruction & Data, 32KB each
 - Interrupt controller
- NAND flash controller
- Quad SPI flash controller
- SDRAM controller subsystem
- 64 KB on-chip RAM
- 64 KB on-chip boot ROM

... features of Cyclone V HPS ...

The general peripherals modules of the **HPS** are:

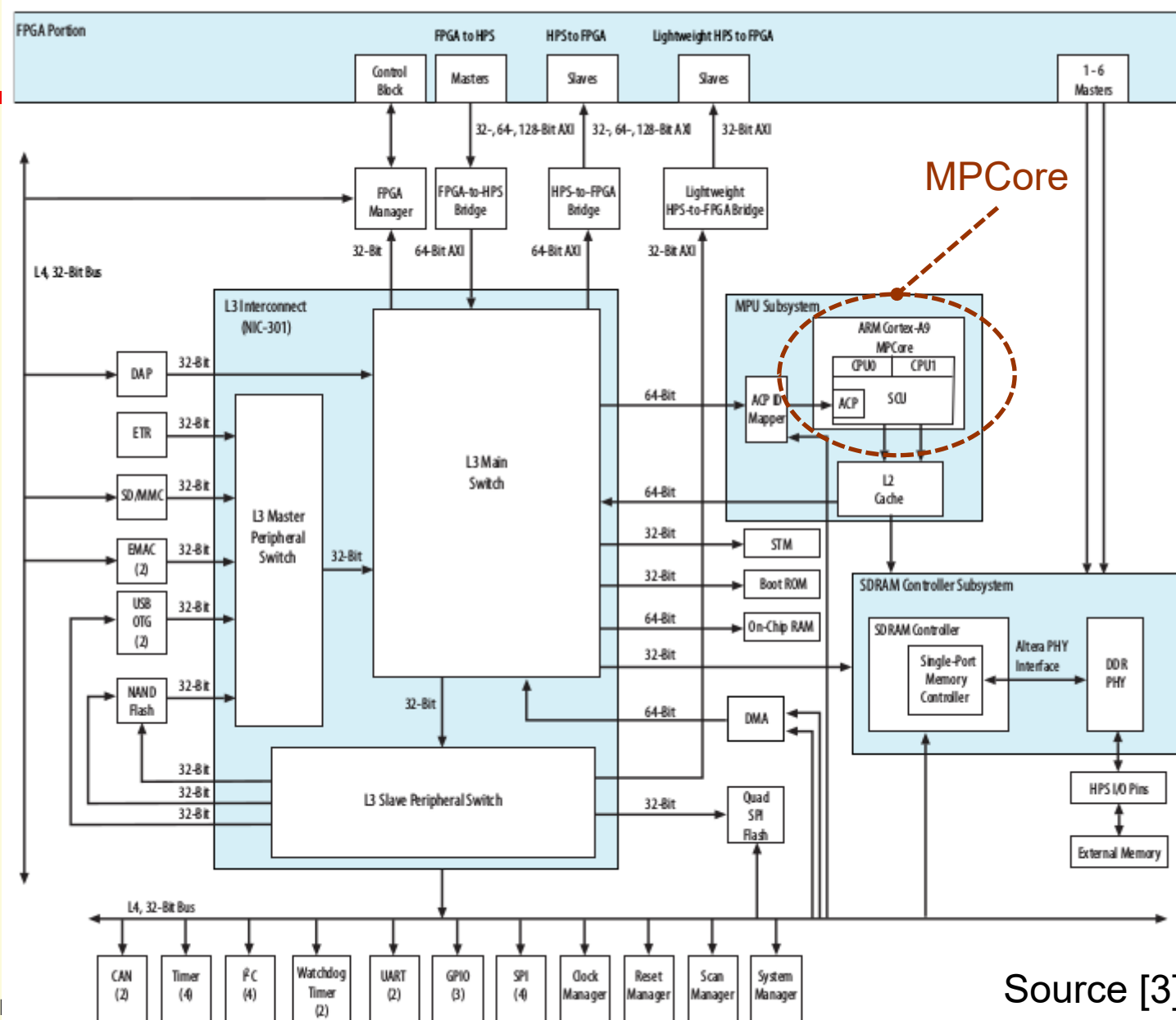
- General-purpose direct memory access (DMA) controller
- Secure digital/multimedia card (SD/MMC) controller
- Two Ethernet media access controllers (EMACs)
- Two USB 2.0 on-the-go (OTG) controllers
- Two controller area network (CAN) controllers
- Two serial peripheral interface (SPI) master controllers
- Two serial peripheral interface (SPI) slave controllers
- Four inter-integrated circuit (I²C) controllers
- Two UARTs
- Four timers
- Three general-purpose I/O (GPIO) interfaces

... features of Cyclone V HPS

The system modules of the **HPS** are:

- System manager
- Clock manager
- Reset manager
- FPGA manager
- Scan manager
- Two watchdog timers
- ARM Coresight™ debug components:
 - Debug access port (DAP)
 - Trace port interface unit (TPIU)
 - System trace macrocell (STM)
 - Program trace macrocell (PTM)
 - Embedded trace router (ETR)
 - Embedded cross trigger (ECT)

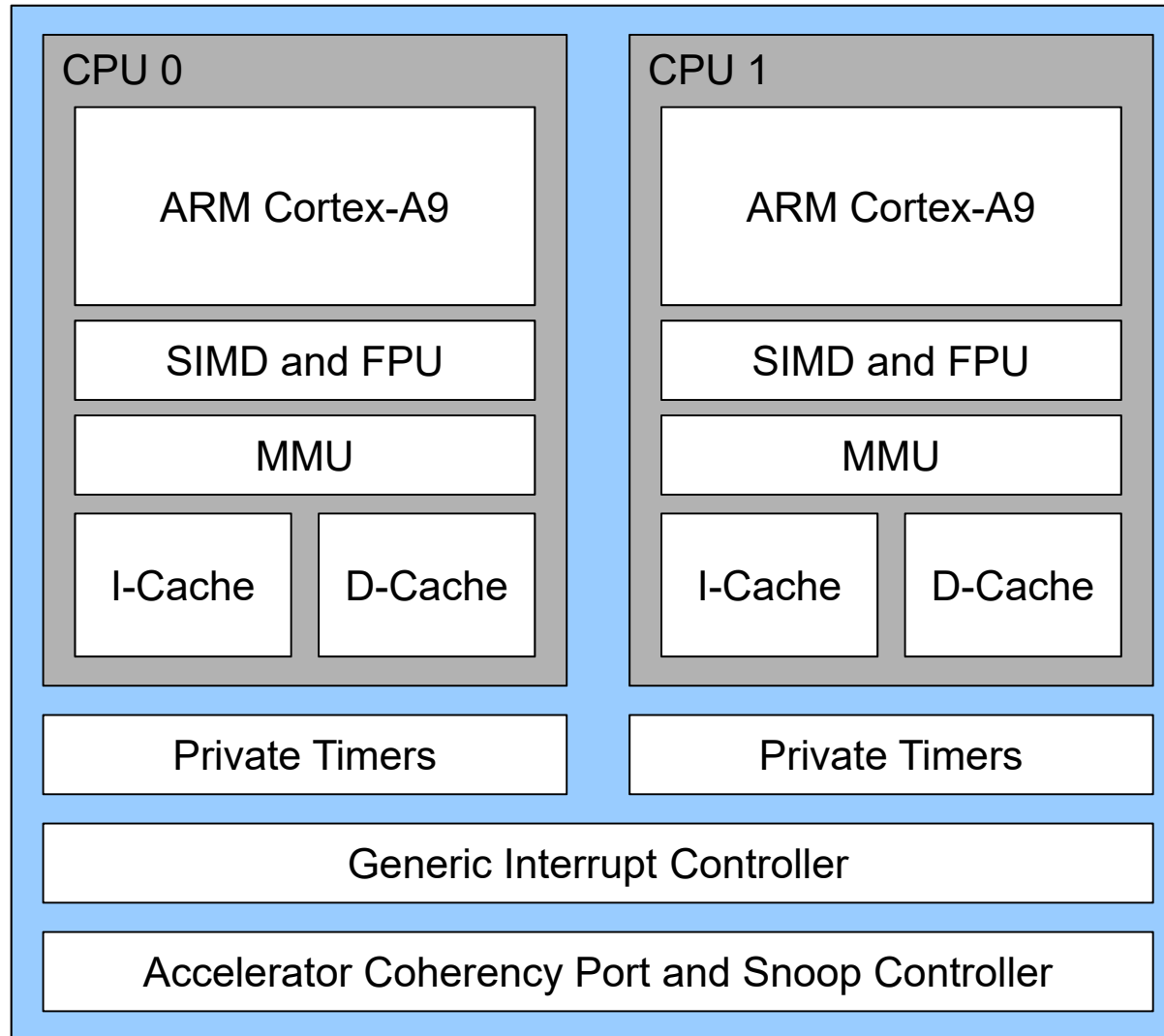
Cyclone V SoC block diagram



MPCore

Source [3]

ARM Cortex-A9 MPCore



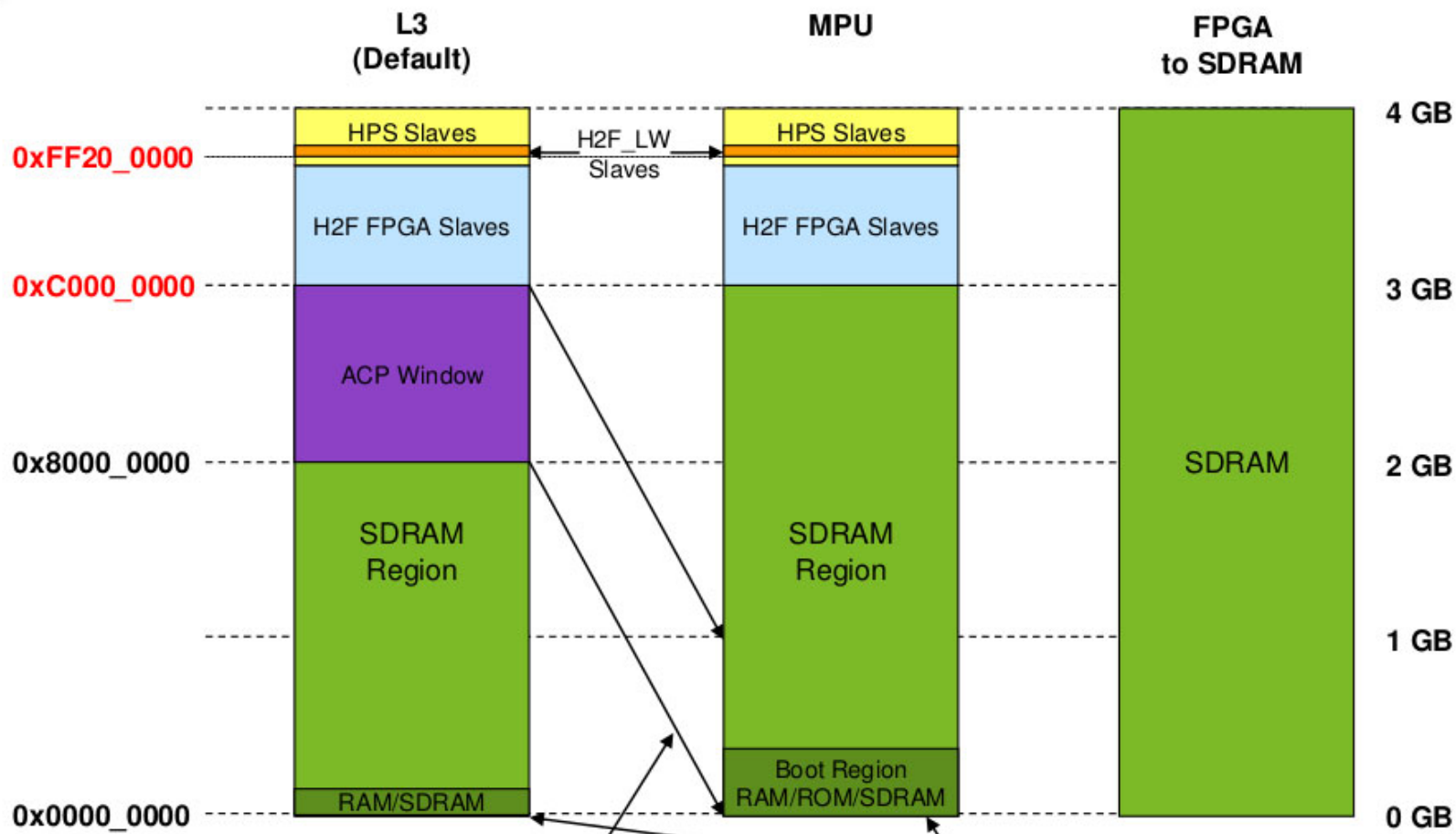
Questions

1. Comparaison entre microcontrôleur / circuit SoC-FPGA
 - a) Points commun?
 - b) Différences?
2. Composant circuit SoC-FPGA ?
 - a) Avantages
 - b) Inconvénients
3. Différences entre un processeur soft core intégré dans une FPGA et un composant SoC-FPGA (hard core) ?

Memory map

- Visibilité du plan d'adressage différenciée selon le bus utilisé
 - Les zones mémoires sont différentes
 - Il s'agit d'adresse physique
 - L'accès est différent depuis le HPS ou la FPGA
- La figure suivante donne un aperçu de 3 plans d'adressage.
 - Nous allons étudier/ utiliser différents cas durant le cours

Cyclone V & Arria V SoC HPS Physical Memory Map



H2F: HPS to FPGA

ACP : Accelerator Coherency Port

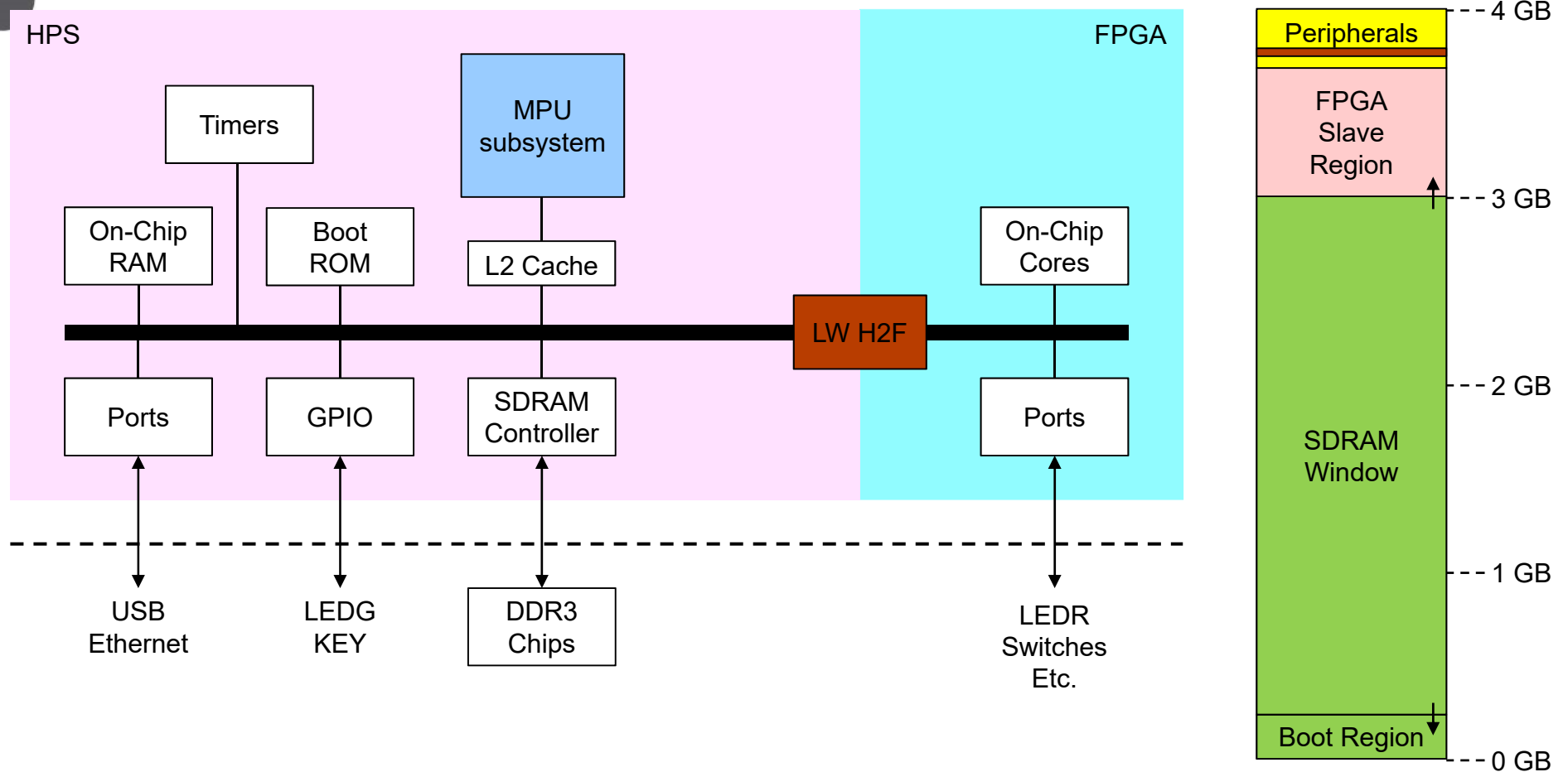
Default remap to 0x0

Remaps as RAM & ROM or SDRAM

Source [2]



Lightweight HPS-to-FPGA Bridge



H2F: HPS to FPGA

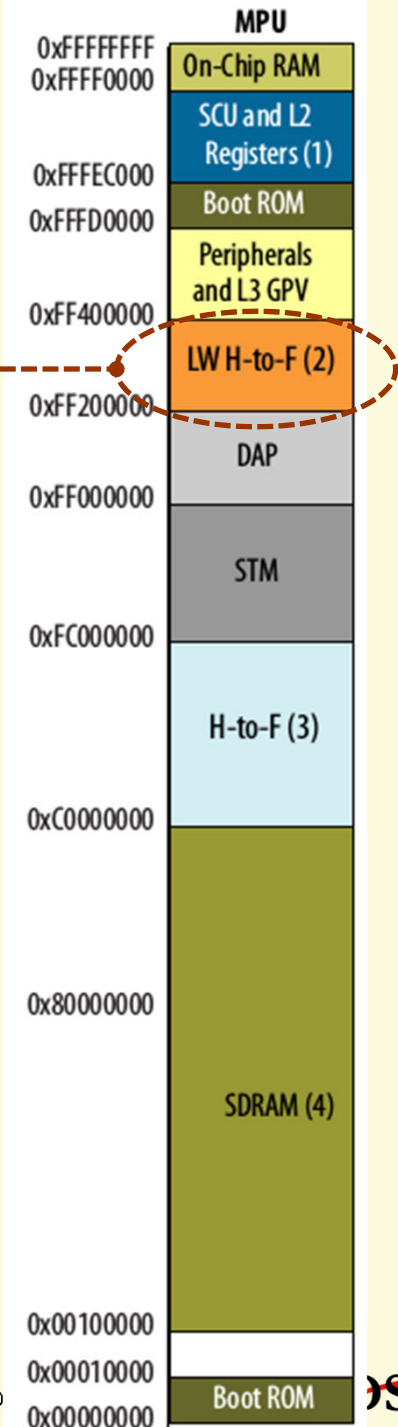
MPU Microprocessor Unit Subsystem

Memory map Lightweight H2F

- Adresses du bridge Lightweight HPS-to-FPGA
 - 2 MB
 - 0xFF20 0000 to 0xFF3FFFFFF

Autres zones du plan mémoire CPU :

- On-chip RAM
 - 64 KB
 - 0xFFFF0000 to 0xFFFFFFFF
- Adresse des périphériques
 - 12096 KB
 - 0xFF400000 to 0xFFFCFFFF
- ...

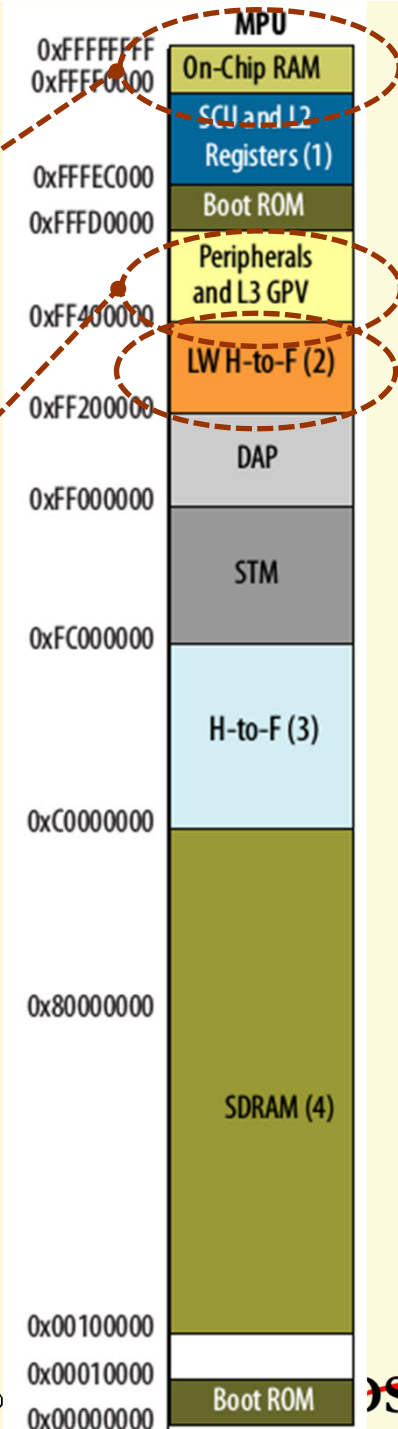


Memory map Lightweight H2F

- Adresses du bridge Lightweight HPS-to-FPGA
 - 2 MB
 - 0xFF20 0000 to 0xFF3FFFFFF

Autres zones du plan mémoire CPU :

- On-chip RAM
 - 64 KB
 - 0xFFFF0000 to 0xFFFFFFFF
- Adresse des périphériques
 - 12096 KB
 - 0xFF400000 to 0xFFFCFFFF
- ...



Interconnections HPS<>FPGA

- Il y a trois bridge entre le HPS et la FPGA :
 - FPGA-to-HPS Bridge
 - HPS-to-FPGA Bridge
 - Lightweight HPS-to-FPGA Bridge
- Il y a un interface SDRAM entre la FPGA et le memory-controller du HPS:
 - SDRAM controller FPGA-to-HPS

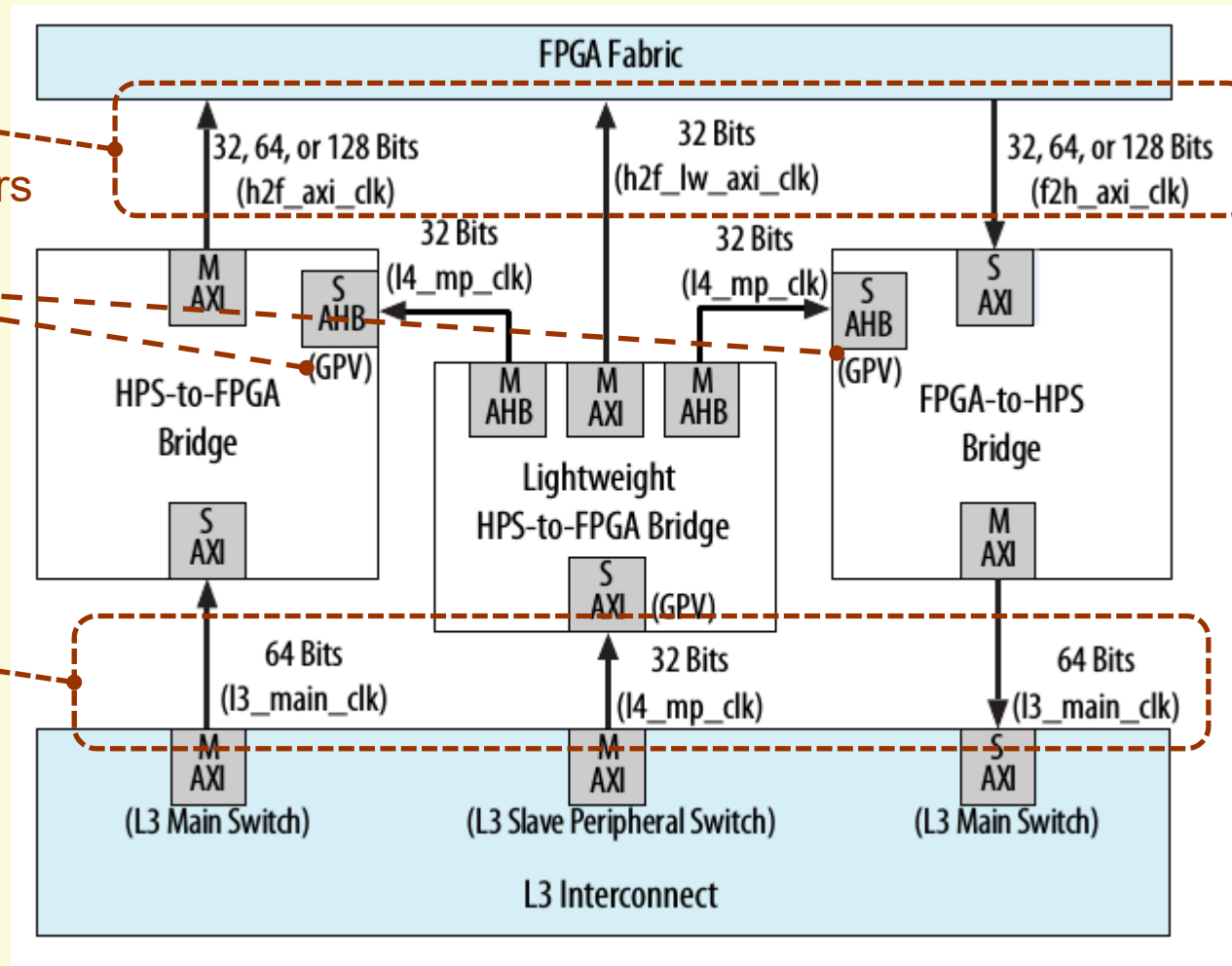
Source [3] : HPS-FPGA Bridges, chapitre 8
SDRAM Controller Subsystem, chapitre 12

FPGA Bridges Block Diagram

Size and clock on the FPGA side

GPV : Global programmers view access to registers

Size and clock on the HPS side

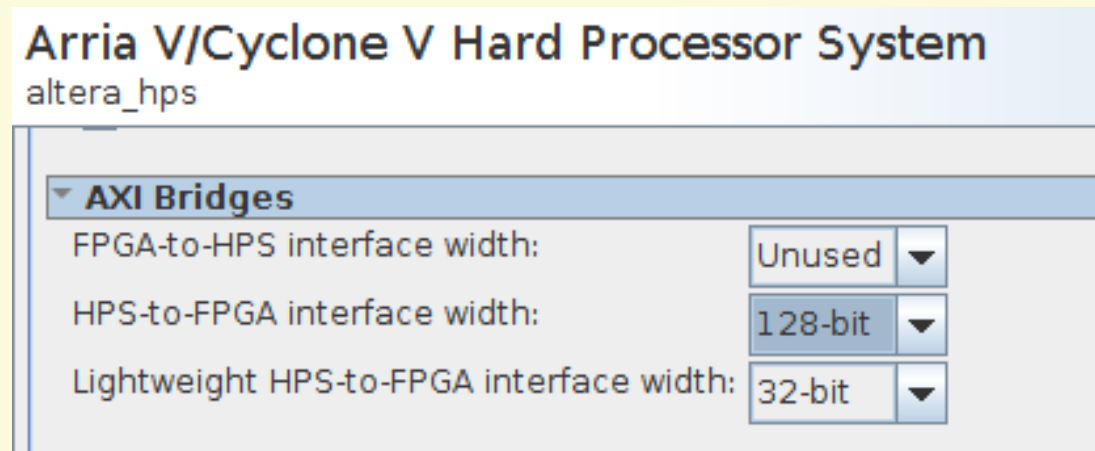


AHB :
Advanced High-performance Bus
AXI :
Advanced eXtensible Interface

Source [3]

FPGA Bridges configuration

- Configuration depuis la page paramètres du HPS dans Qsys.
 - Voici la vue de la partie correspond aux bridges :



Source [3]

Interconnections HPS<>FPGA

- **FPGA-to-HPS Bridge**
 - Bus haute performance FPGA (master) -> HPS (slave)
 - Côté HPS: connecté sur le L3 Main switch
 - Plusieurs largeur de données

Bridge Property	FPGA Slave Interface	L3 Master Interface
Connected to	FPGA fabric	L3 Main switch
Data width	32, 64, 128 bits	64 bits
Byte address width	32 bits	32 bits
Read/Write acceptance	16 transactions	16 transactions

Interconnections HPS<>FPGA

- HPS-to-FPGA Bridge

- Bus haute performance HPS (master) -> FPGA (slave)
- Côté HPS: connecté sur le L3 Main switch
- Plusieurs largeur de données

Bridge Property	L3 Slave Interface	FPGA Master Interface
Connected to	L3 Main switch	FPGA fabric
Data width	64 bits	32, 64, 128 bits
Address width	32 bits (bytes)	30 bits (word)
Read/Write acceptance	16 transactions	16 transactions

Pourquoi 30 bits ?

Interconnections HPS<>FPGA

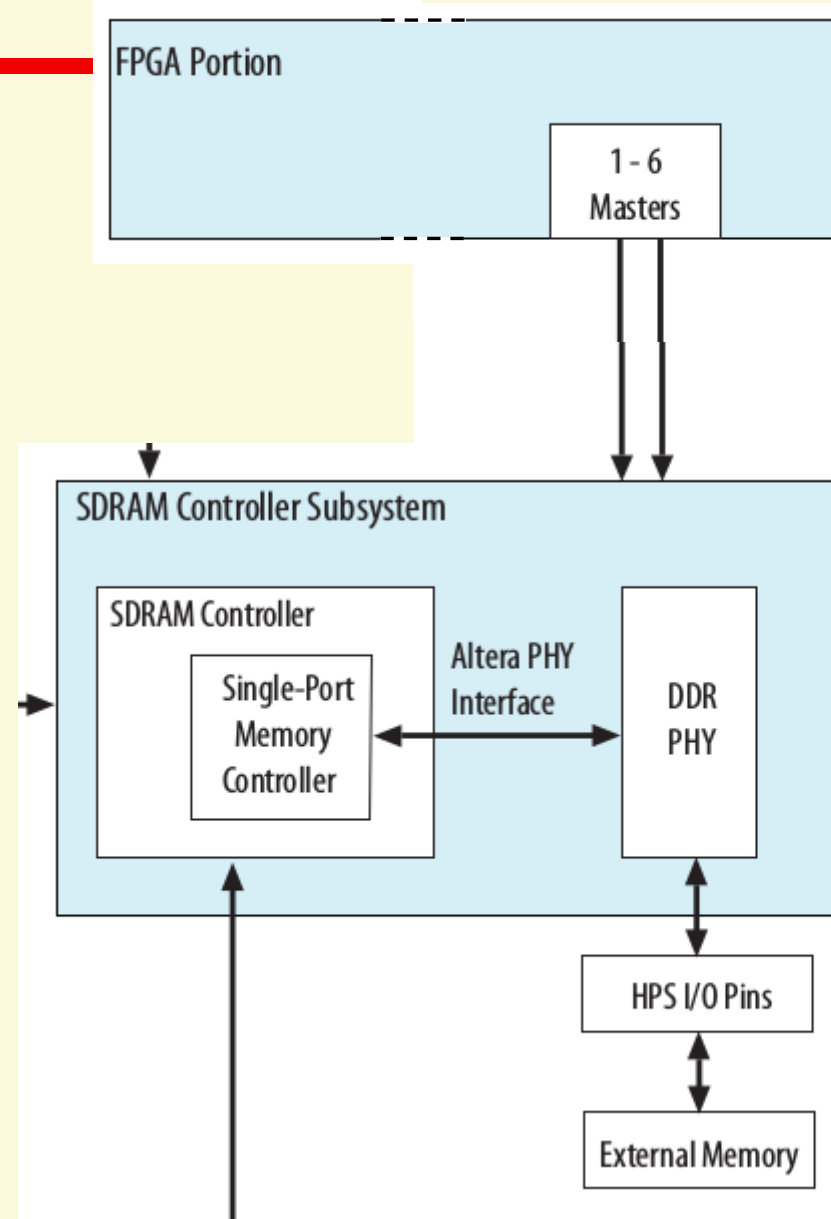
- Lightweight HPS-to-FPGA Bridge
 - Bus 32 bits
 - Bus pour l'accès aux périphériques HPS et registres de configuration

Bridge Property	L3 Slave Interface	FPGA Master Interface
Connected to	L3 Slave peripheral switch	FPGA fabric
Data width	32 bits	32 bits
Byte address width	32 bits	21 bits
Read/Write acceptance	16 transactions	16 transactions

Pourquoi 21 bits ?

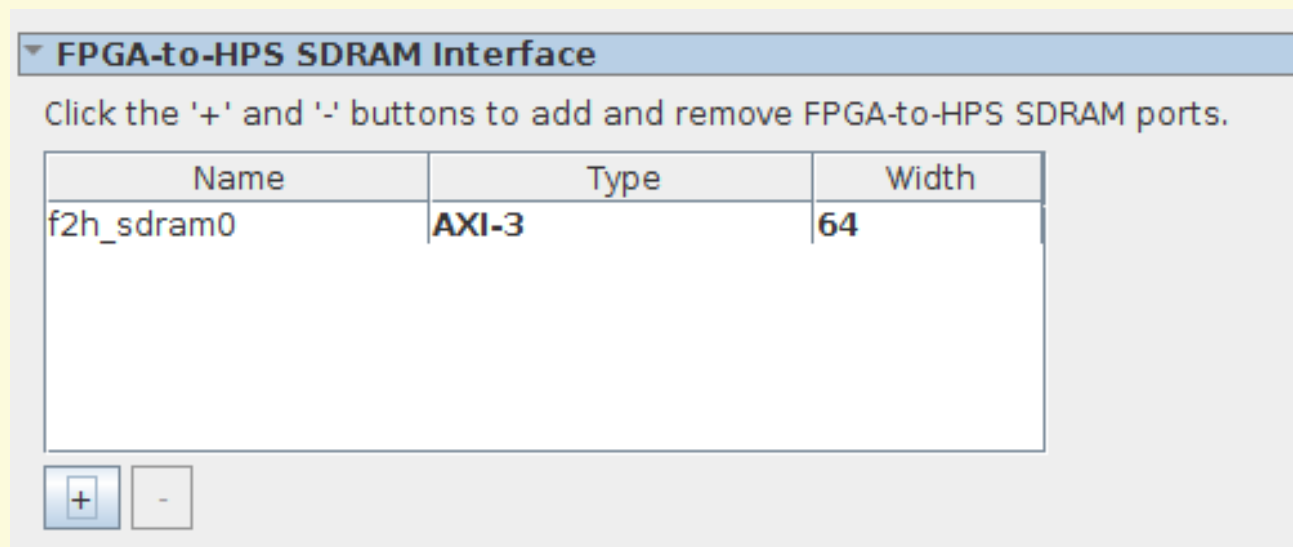
Interface FPGA - SDRAM Ctrl

- Schéma bloc de l'interface



Interface FPGA - SDRAM Ctrl

- Configuration depuis la page paramètres du HPS dans Qsys.
 - Voici la vue de la partie correspond à cet interface :



Voir : Source [3], SDRAM Controller Subsystem, chapitre 12

Interface FPGA - SDRAM Ctrl

- Configuration depuis la page paramètres du HPS dans Qsys.
 - Paramètres :

Paramètres	Options
Type de bus	AXI3, Avalon-MM bidirectionnal Avalon-MM write only, Avalon-MM read only
Data width	32, 64, 128, 256 bits

Questions

4. Le circuit Cyclone-V SoC dispose de plusieurs niveaux de bus:
 - a) Pourquoi ?
 - b) Combien?

5. Interconnections HPS<>FPGA
 - a) Pourquoi plusieurs bridges entre HPS – FPGA ?
 - b) Pourquoi différents Master ?
 - c) Quel est l'utilisation du bus LW-HPS-to-FPGA ?
 - d) Utilisation de l'interface SDRAM FPGA-to-HPS ?
 - e) Existe-t-il des signaux connectés entre HPS & FPGA ?
Si oui pourquoi ?

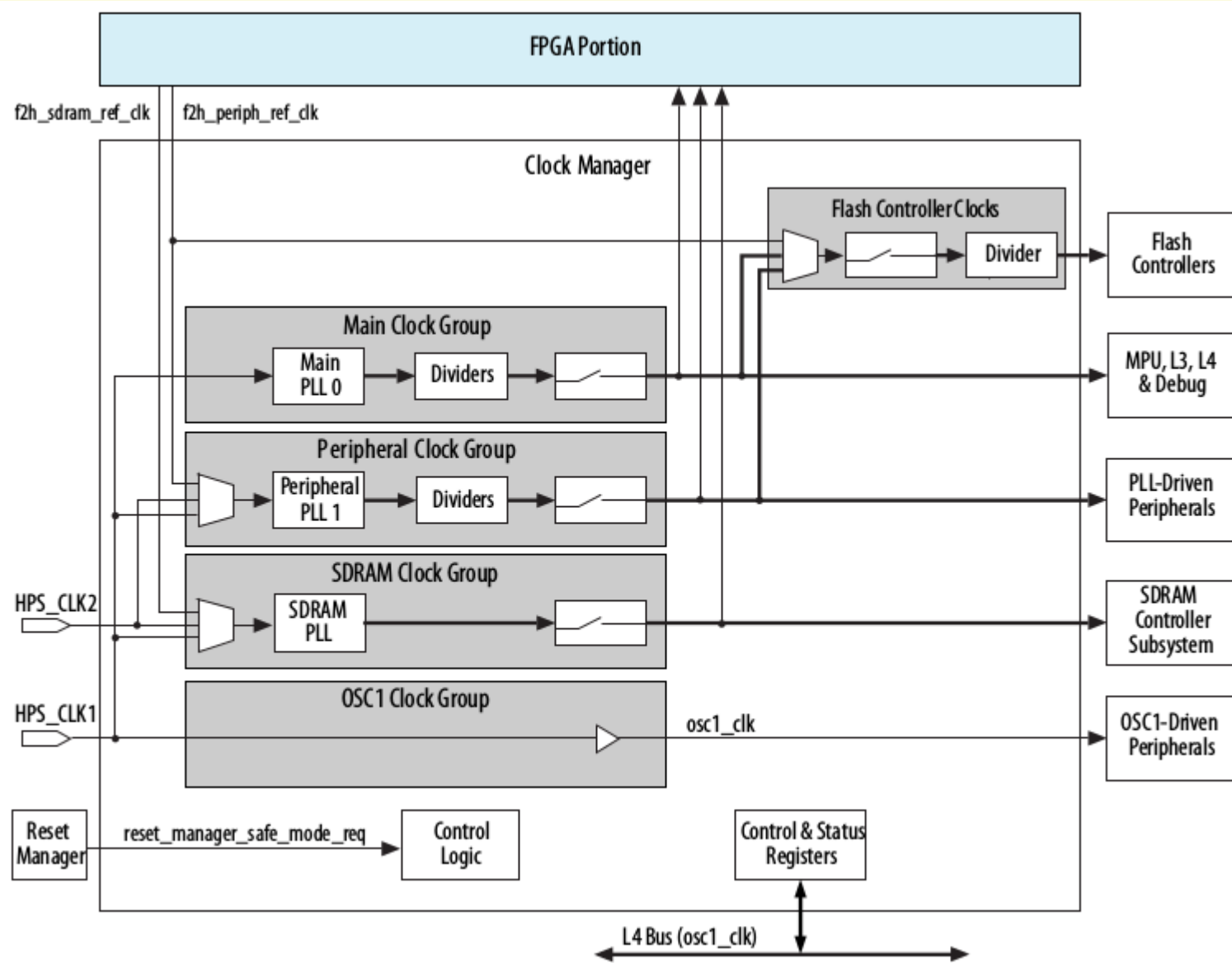
Questions

6. Est-ce que la FPGA du Cyclone V dispose de hard blocs?
 - a) Si oui, lesquels?
 - b) Quelles sont les avantages?
7. Le Cyclone-V dispose d'un "Clock manager"
A votre avis quel est son rôle ?
8. Le Cyclone-V dispose d'un "Reset manager"
A votre avis quel est son rôle ?

Clock manager

- Le bloc "clock manager" assure la génération centralisée des horloges. Il dispose des fonctionnalités suivantes:
 - Dispose de 3 PLL pour les principales horloges (voir schéma)
 - Permet d'ajuster les horloges sans désactiver les périphériques et la mémoire SDRAM
 - Contrôle l'activation et la désactivation pour la majorité des horloges
 - Permet l'initialisation des horloges lors d'un reset "cold" ou "warm"
 - Permet la configuration des paramètres des horloges via le programme
- Il y a différentes sources possibles d'horloge pour le "clock manager"
 - Horloges du HPS: HPS_CLK1 ou HPS_CLK2
 - Horloges de la FPGA : f2h_sdram_ref_clk ou f2h_periph_ref_clk

Clock manager



Reset manager

- Système permettant de gérer différents types de reset dans le SoC-FPGA
- Trois domaines de reset

Domain	Domain Logic	Source [3]
TAP	JTAG test access port (TAP) controller, which is used by the debug access port (DAP) (TAP fait partie du Scan Manager)	
Debug	All debug logic including most of the DAP, CoreSight TM components	
System	All HPS logic except what is in the TAP and debug reset domains. Includes non-debug logic in the FPGA fabric connected to the HPS reset signal	

Reset manager

- Le SoC supporte trois types de reset:
 - System cold reset
 - relance le boot complet
 - reset l'ensemble des composants et de la logique
 - affecte tous les domaines de reset
 - System warm reset
 - intervient après un "cold reset"
 - récupérer le système après une non-réponse
 - affecte seulement le "system domain"
 - Debug reset
 - utilisé pour récupérer la logique de debug après une non-réponse
 - affecte seulement le "debug reset domain"

Source [3] : Reset Manager, chapitre 3

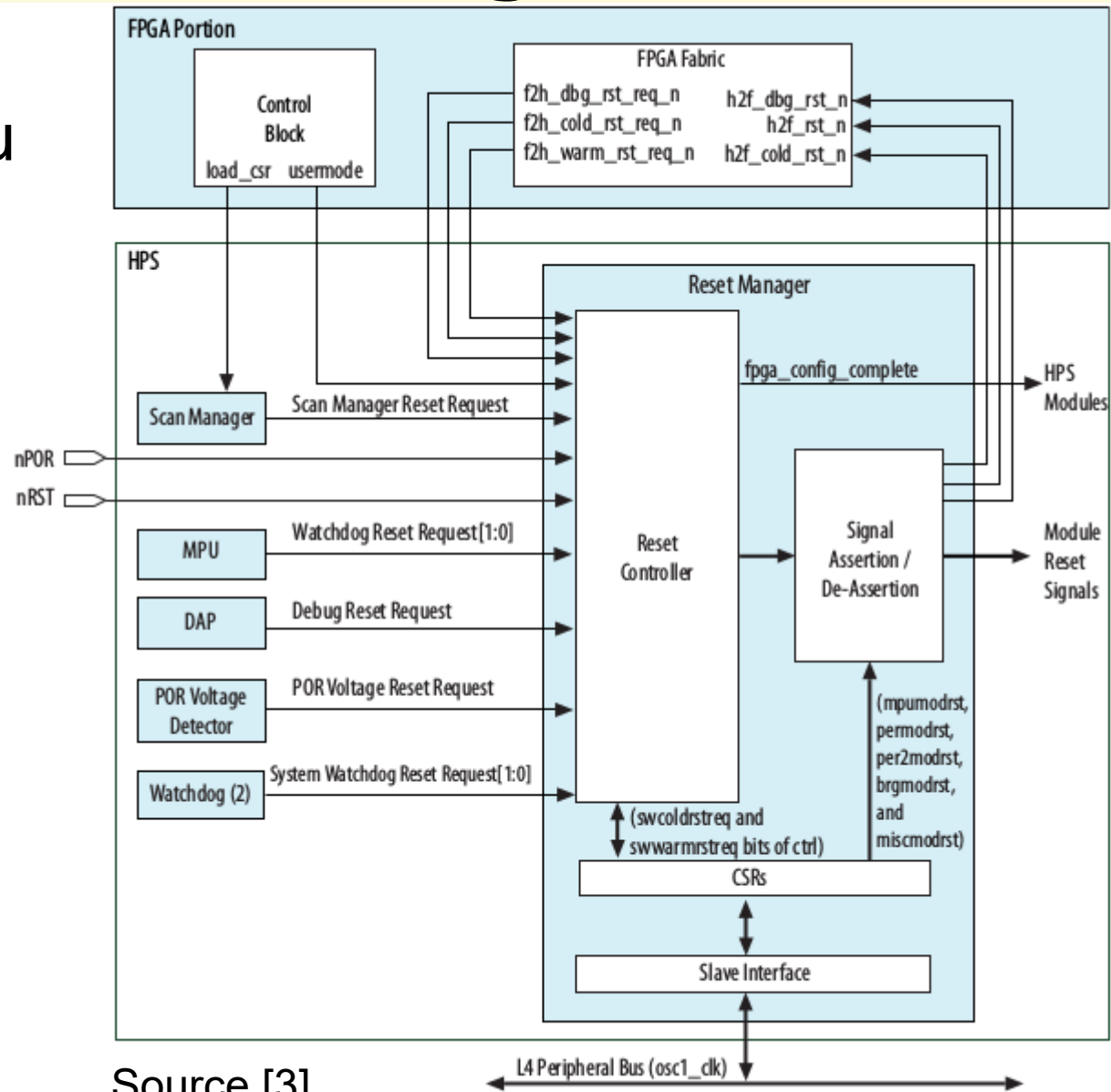
Reset manager

- Les sources de reset sont multiples:
 - Power-on reset
 - circuit de détection sur l'alimentation
 - Bouton reset sur le HPS
 - reset "cold" ou "warm"
 - Depuis la FPGA
 - reset "cold", "warm", "debug"
 - Watchdog
 - Système de debug (DAP)
 - ...

Source [3] : Reset Manager, chapitre 3

Reset manager

- Schéma bloc du gestionnaire de reset



Reset manager

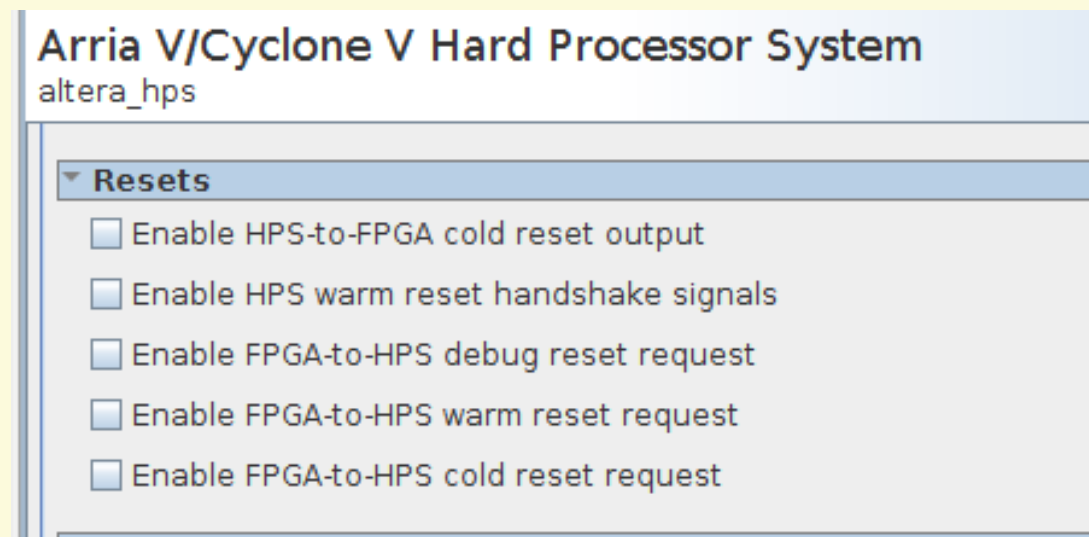
- Table 3-2: HPS External Reset Sources

Source	Description
f2h_cold_rst_req_n	Cold reset request from FPGA fabric (active low)
f2h_warm_rst_req_n	Warm reset request from FPGA fabric (active low)
f2h_dbg_rst_req_n	Debug reset request from FPGA fabric (active low)
h2f_cold_rst_n	Cold-only reset to FPGA fabric (active low)
h2f_rst_n	Cold or warm reset to FPGA fabric (active low)
h2f_dbg_rst_n	Debug reset (<code>dbg_rst_n</code>) to FPGA fabric (active low)
load_csr	Cold-only reset from FPGA control block (CB) and scan manager
nPOR	Power-on reset pin (active low)
nRST	Warm reset pin (active low)

Source [3]

Reset manager configuration

- Configuration depuis la page paramètres du HPS dans Qsys
 - Voici la vue de la partie correspond aux reset :



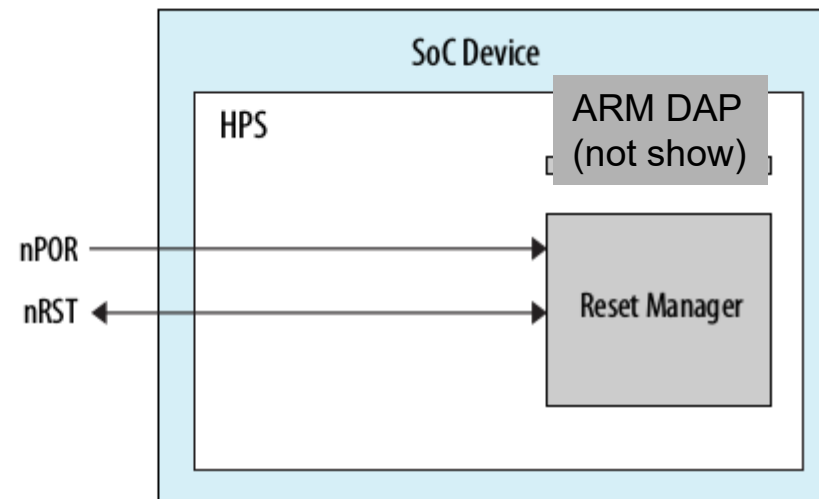
Source [3]

Reset manager

- Reset pins for the Reset manager:
 - The nPOR and nRST pins are used to request cold and warm resets respectively.
 - The nRST pin is an open drain output as well.

Figure 3-5: Reset Pins

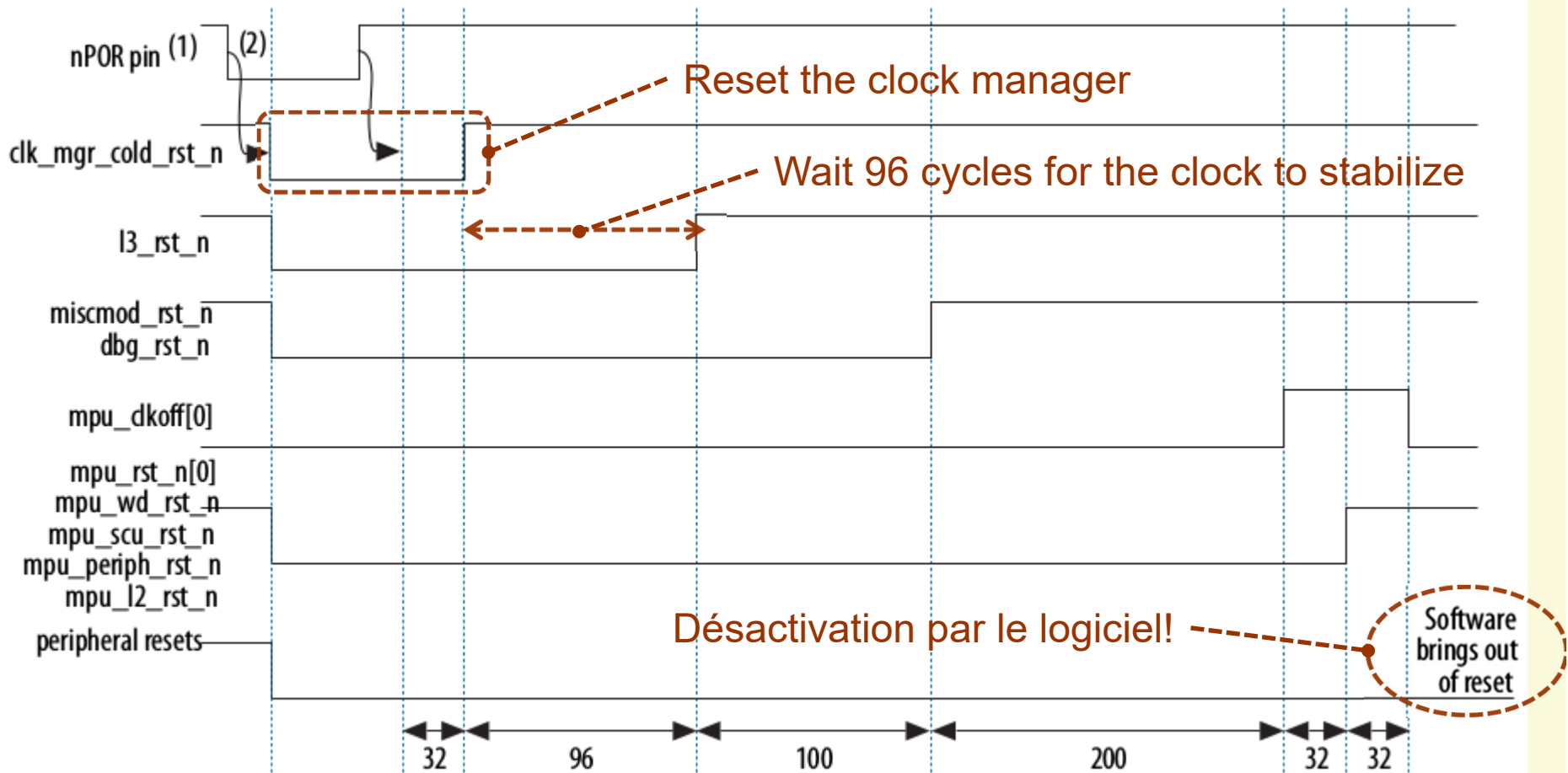
Source [3]



Reset manager

Figure 3-3: Cold Reset Timing Diagram

Source [3]



(1) Cold reset can be initiated from several other sources: FPGA CB, FPGA fabric, modules in the HPS, and reset pins.

(2) This dependency applies to all the reset signals.

Reset manager

Cold Reset Assertion Sequence

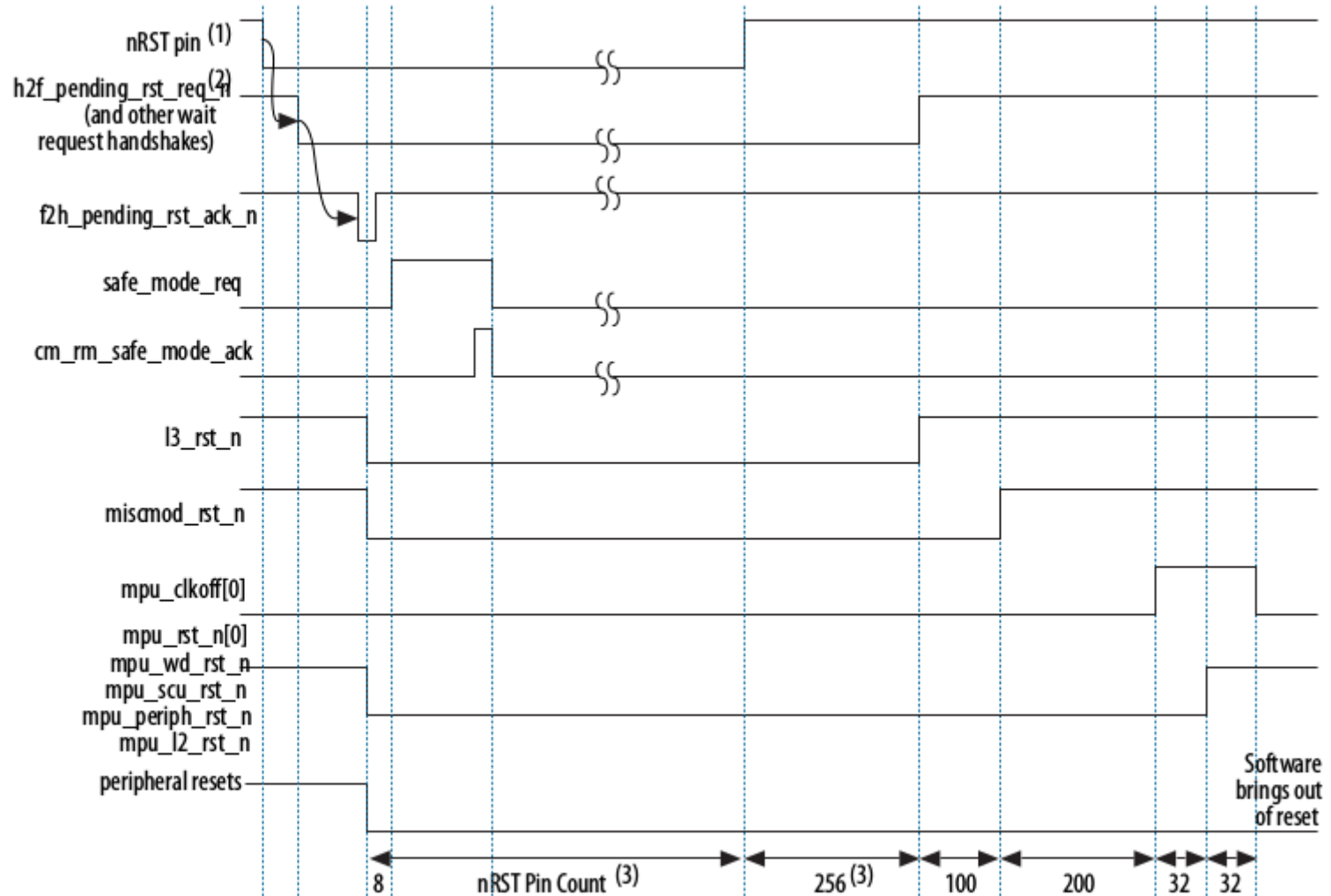
The following list describes the assertion steps for cold reset shown in the Cold Reset timing diagram:

1. Assert module resets
2. Wait for 32 cycles. Deassert clock manager cold reset.
3. Wait for 96 cycles (so clocks can stabilize).
4. Proceed to the “Cold and Warm Reset Deassertion Sequence” => voir à la suite

Reset manager

Figure 3-4: Warm Reset Timing Diagram

Source [3]



(1) Cold reset can be initiated from several other sources: FPGA CB, FPGA fabric, modules in the HPS, and reset pins.

(2) When the nRSTpin count is zero, the 256 cycle stretch count is skipped and the start of the deassertion sequence is determined by the safe mode acknowledge signal or the user releasing the warm reset button, whichever occurs later.

Reset manager

Warm Reset Assertion Sequence

The following list describes the assertion steps for warm reset shown in the Warm Reset Timing Diagram:

1. Optionally, handshake with the embedded trace router (ETR) and wait for acknowledge.
2. Optionally, handshake with the FPGA fabric and wait for acknowledge.
3. Optionally, handshake with the SDRAM controller, scan manager, and FPGA manager, and wait for acknowledges.
4. Assert module resets (except the MPU watchdog timer resets when the MPU watchdog timers are the only request sources).
5. Wait for 8 cycles and send a safe mode request to the clock manager.
6. Wait for the greater of the nRST pin count + 256 stretch count, or the warm reset counter, or the clock manager safe mode acknowledge, then deassert all handshakes except warm reset ETR handshake (which is deasserted by software).
7. Proceed to the “Cold and Warm Reset Deassertion Sequence” => voir à la suite

Reset manager

Cold and Warm Reset Deassertion Sequence

The following list describes the deassertion steps for both cold and warm reset shown in the Cold Reset Timing Diagram and Warm Reset Timing Diagram:

1. Deassert L3 reset.
2. Wait for 100 cycles. Deassert resets for miscellaneous-type and debug (cold only) modules.
3. Wait for 200 cycles. Assert mpu_clkoff for CPU0 and CPU1.
4. Wait for 32 cycles. Deassert resets for MPU modules.
5. Wait for 32 cycles. Deassert mpu_clkoff for CPU0 and CPU1.
6. Peripherals remain held in reset until software brings them out of reset.

Reset Manager Module Address Map

- Base address: 0xFFD05000

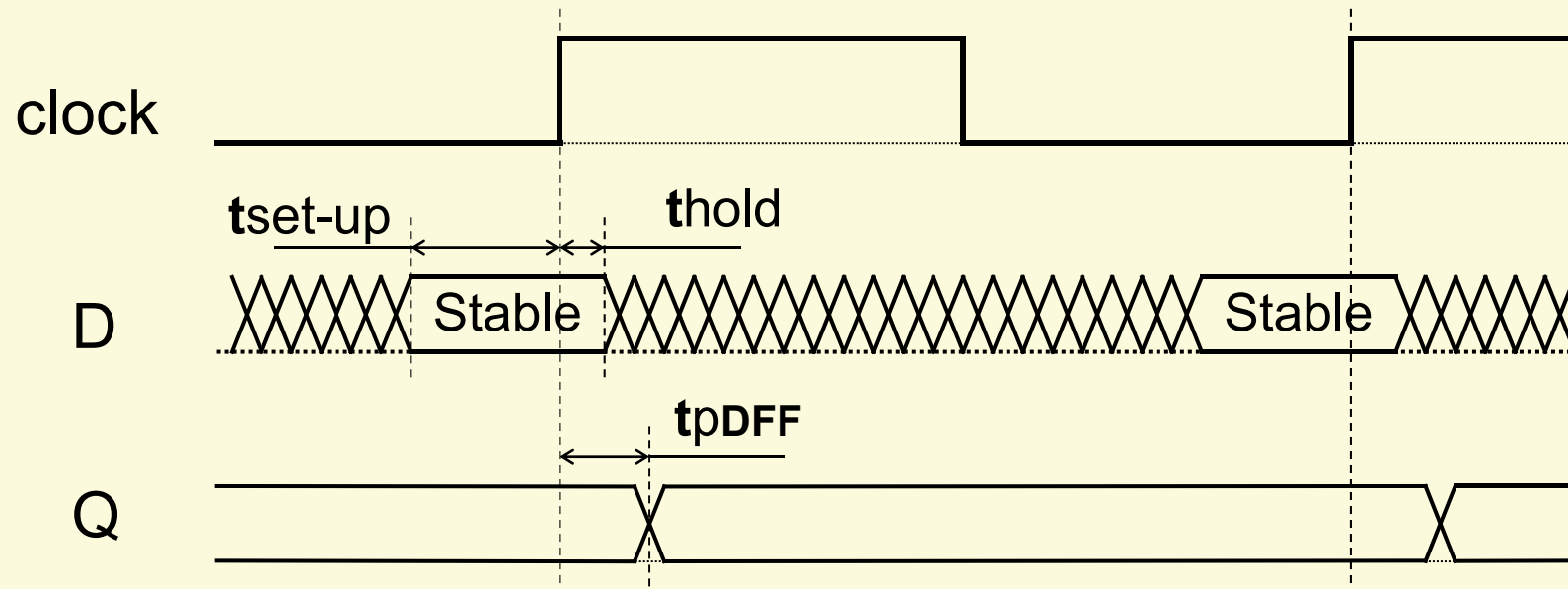
Register	Offset	Width	Access	Reset Value	Description
<code>stat</code> on page 3-17	0x0	32	RW	0x0	Status Register
<code>ctrl</code> on page 3-19	0x4	32	RW	0x100000	Control Register
<code>counts</code> on page 3-23	0x8	32	RW	0x80080	Reset Cycles Count Register
<code>mpumodrst</code> on page 3-24	0x10	32	RW	0x2	MPU Module Reset Register
<code>permodrst</code> on page 3-25	0x14	32	RW	0x3FFFFFFF	Peripheral Module Reset Register
<code>per2modrst</code> on page 3-27	0x18	32	RW	0xFF	Peripheral 2 Module Reset Register
<code>brgmodrst</code> on page 3-29	0x1C	32	RW	0x7	Bridge Module Reset Register
<code>miscmodrst</code> on page 3-29	0x20	32	RW	0x0	Miscellaneous Module Reset Register
<code>tsstscratch</code> on page 3-31	0x54	32	RW	0x0	Test Scratch Register

Action du reset

- Est-t-il possible que le reset crée un cas de dysfonctionnement d'un flip-flop?
 - Si non: justifier qu'il n'y a pas de cas problématique ?
 - Si oui: dans quel cas y a-t-il un risque ?

Fonctionnement d'un flip-flop

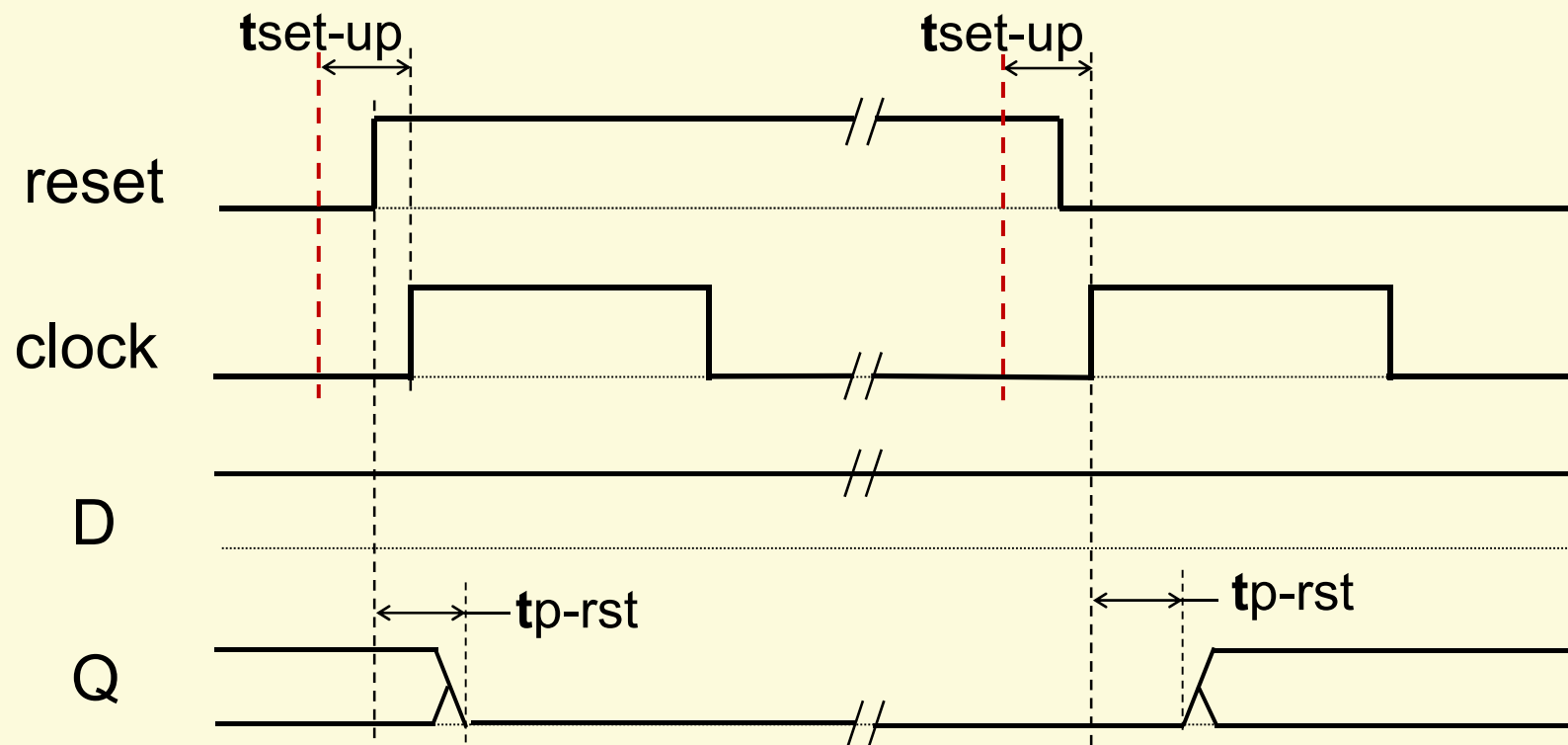
- Fonctionnement dynamique d'un flip-flop:



Contrainte : D doit être stable durant le temps set-up
En pratique thold = 0 (pas de contrainte)

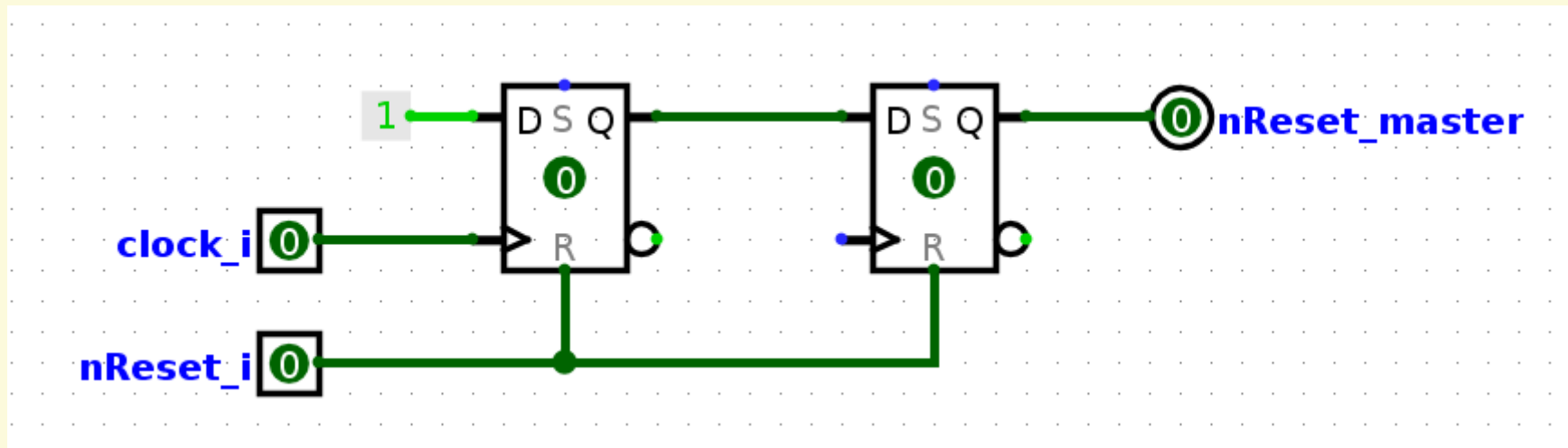
Action du reset

- Voici la situation où le reset peut créer un dysfonctionnement du flip-flop: $T_{hold} = 0 \text{ ns}$



Désactivation d'un reset

- Solution :
 - Génère une désactivation synchrone du signal reset.



- Option outil EDA:
OUTPUT_RESET_SYNC_EDGES => "deassert",

Questions : Reset manager

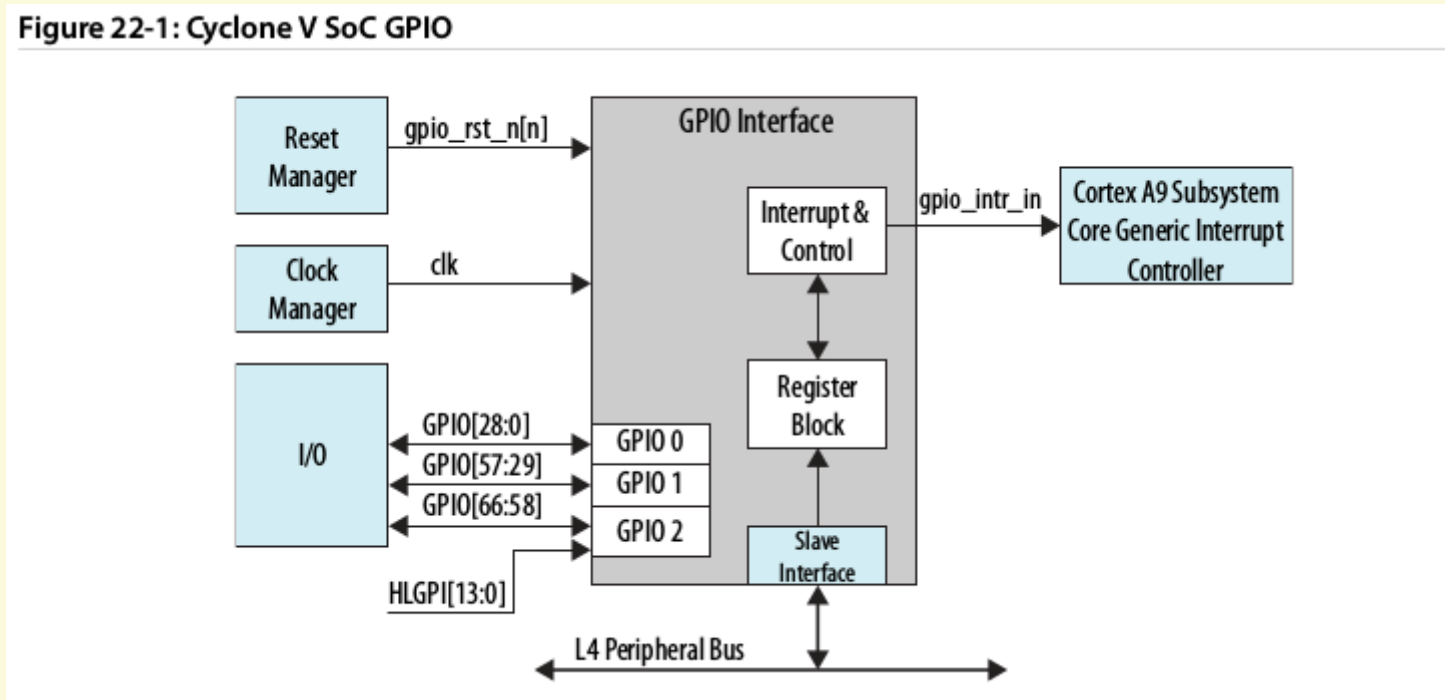
1. Quel est le rôle du Reset manager ?
2. Pourquoi parle-t-on d'un reset synchrone ?
3. Quels sont les sources possible de Reset ?
4. Pourquoi existe-t-il différents types de Reset ?
 - Cold - Warm - Debug

General-Purpose I/O Interface

- The GPIO interface offers the following features:
 - Supports digital debounce
 - Configurable interrupt mode
 - Supports up to 67 I/O pins and 14 input-only pins

General-Purpose I/O Interface

- Les GPIO sont répartis dans 4 bank, soit :



Source [3]

General-Purpose I/O Interface

- Voici la répartition des GPIO entre les différentes bank :

Source [3]

Table 22-1: GPIO Interface pin table

Pin Name	Mapped to GPIO Signal Name	Comments
GPIO [28:0]	GPIO 0 [28:0]	Input / Output
GPIO [57:29]	GPIO 1 [28:0]	Input / Output
GPIO [66:58]	GPIO 2 [8:0]	Input / Output
HLGPI [13:0]	GPIO 2 [26:13]	Input only

Table 22-2: GPIO Interface pin table

Pin Name	Mapped to GPIO Signal Name	Comments
HPS_DEDICATED_Q1 [12:1]	GPIO 0 [11:0]	Input / Output
HPS_DEDICATED_Q2 [12:1]	GPIO 0 [23:12]	Input / Output
HPS_DEDICATED_Q3 [12:1]	GPIO 1 [11:0]	Input / Output
HPS_DEDICATED_Q4 [12:1]	GPIO 1 [23:12]	Input / Output

General-Purpose I/O Interface

- Les GPIO sont partagés avec les pins dédiées du HPS.
- La configuration se fait dans Qsys :
 - HPS parameters, menu Peripheral pins

Arria V/Cyclone V Hard Processor System
altera_hps

SPI Controllers

SPI0 pin:	HPS I/O Set 0
SPI0 mode:	Dual Slave Selects
SPI1 pin:	Unused
SPI1 mode:	N/A
SPI2 pin:	Unused
SPI2 mode:	N/A
SPI3 pin:	Unused
SPI3 mode:	N/A

UART Controllers

UART0 pin:	HPS I/O Set 2
UART0 mode:	Flow Control
UART1 pin:	Unused
UART1 mode:	N/A

I2C Controllers

I2C0 pin:	HPS I/O Set 1
I2C0 mode:	I2C
I2C1 pin:	Unused
I2C1 mode:	N/A
I2C2 pin:	Unused
I2C2 mode:	N/A
I2C3 pin:	Unused
I2C3 mode:	N/A

CAN Controllers

CAN0 pin:	Unused
CAN0 mode:	N/A
CAN1 pin:	Unused
CAN1 mode:	N/A

Source [3]

General-Purpose I/O Interface

- Configuration depuis la page paramètres du HPS dans Qsys.
 - Voici la vue de la table de multiplexage des GPIO avec les fonctions HPS :

SDMMC_D6		USB0.D6 (Set0)	SDIO.D6 (Set0)	GPIO42	LOANIO42
SDMMC_D7		USB0.D7 (Set0)	SDIO.D7 (Set0)	GPIO43	LOANIO43
HPS_GPIO44		USB0.CLK (Set0)		GPIO44	LOANIO44
SDMMC_CCLK_OUT		USB0.STP (Set0)	SDIO.CLK (Set0)	GPIO45	LOANIO45
SDMMC_D2		USB0.DIR (Set0)	SDIO.D2 (Set0)	GPIO46	LOANIO46
SDMMC_D3		USB0.NXT (Set0)	SDIO.D3 (Set0)	GPIO47	LOANIO47
TRACE_CLK			TRACE.CLK (Set0)	GPIO48	LOANIO48
TRACE_D0	UART0.RX (Set0)	SPIS0.CLK (Set0)	TRACE.D0 (Set0)	GPIO49	LOANIO49
TRACE_D1	UART0.TX (Set0)	SPIS0.MOSI (Set0)	TRACE.D1 (Set0)	GPIO50	LOANIO50
TRACE_D2	I2C1.SDA (Set0)	SPIS0.MISO (Set0)	TRACE.D2 (Set0)	GPIO51	LOANIO51
TRACE_D3	I2C1.SCL (Set0)	SPIS0.SS0 (Set0)	TRACE.D3 (Set0)	GPIO52	LOANIO52
TRACE_D4	CAN1.RX (Set0)	SPIS1.CLK (Set0)	TRACE.D4 (Set0)	GPIO53	LOANIO53
TRACE_D5	CAN1.TX (Set0)	SPIS1.MOSI (Set0)	TRACE.D5 (Set0)	GPIO54	LOANIO54
TRACE_D6	I2C0.SDA (Set0)	SPIS1.SS0 (Set0)	TRACE.D6 (Set0)	GPIO55	LOANIO55
TRACE_D7	I2C0.SCL (Set0)	SPIS1.MISO (Set0)	TRACE.D7 (Set0)	GPIO56	LOANIO56
SPIM0_CLK	UART0.CTS (Set2) (Set1) (Set0)	I2C1.SDA (Set1)	SPIM0.CLK (Set0)	GPIO57	LOANIO57
SPIM0_MOSI	UART0.RTS (Set2) (Set1) (Set0)	I2C1.SCL (Set1)	SPIM0.MOSI (Set0)	GPIO58	LOANIO58
SPIM0_MISO	UART1.CTS (Set0)	CAN1.RX (Set1)	SPIM0.MISO (Set0)	GPIO59	LOANIO59
SPIM0_SS0	UART1.RTS (Set0)	CAN1.TX (Set1)	SPIM0.SS0 (Set0)	GPIO60	LOANIO60
UART0_RX	SPIM0.SS1 (Set0)	CAN0.RX (Set0)	UART0.RX (Set1)	GPIO61	LOANIO61
UART0_TX	SPIM1.SS1 (Set0)	CAN0.TX (Set0)	UART0.TX (Set1)	GPIO62	LOANIO62
I2C0_SDA	SPIM1.CLK (Set0)	UART1.RX (Set0)	I2C0.SDA (Set1)	GPIO63	LOANIO63
I2C0_SCL	SPIM1.MOSI (Set0)	UART1.TX (Set0)	I2C0.SCL (Set1)	GPIO64	LOANIO64
CAN0_RX	SPIM1.MISO (Set0)	UART0.RX (Set2)	CAN0.RX (Set1)	GPIO65	LOANIO65
CAN0_TX	SPIM1.SS0 (Set0)	UART0.TX (Set2)	CAN0.TX (Set1)	GPIO66	LOANIO66

Source [3]

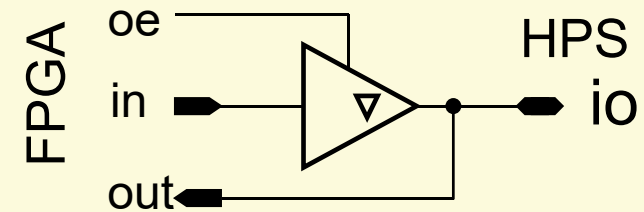
GPIO LoanIO

- Using Unassigned IO as LoanIO

- You can utilize unused HPS IOs as LoanIO, which is directly driven by the FPGA and can be used as input, output, or bi-directional IO
- Each LoanIO port has an input, output, and output enable, which directly controls the HPS IO functions. The LoanIO only operates when the HPS registers have been set up in the pre-loader to allow their operation. The LoanIO are asynchronous, thus no clocking is required

Remarque:

- Chaque pin LoanIO configurée dans le HPS va impliquer trois signaux vers la FPGA, soit
 - h2f_laon_io_in, h2f_laon_io_out, h2f_laon_io_oe



Loan = prêt, donc LoanIO signifie "prêt d'IO" !

GPIO LoanIO

- LoanIO : signaux entre HPS et FPGA

Table 27-11: Generated Conduit Signal Interface

Conduit Name	Direction	Declarations
._hps_io_gpio_inst_LOANIOXX	Bi-direction	User must declare as a top-level pin; pin assignment is hardcoded following the HPS IO location.
._h2f_loan_io_in	Out	HPS IO data input signal, output to FPGA user logic.
._h2f_loan_io_out	In	HPS IO data output signal, input from FPGA user logic.
._h2f_loan_io_oe	In	HPS IO data output enable signal, input from FPGA user logic.

Qsys will generate a full signal array for `h2f_loan_io_in`, `h2f_loan_io_out`, and `h2f_loan_io_oe`. You must assign user logic to the specific signal array. For example, you have triggered LoanIO 40, so its respective signal array is `h2f_loan_io_in[40]`, `h2f_loan_io_out[40]`, and `h2f_loan_io_oe[40]`.

Source [3]

General-Purpose I/O Interface

- GPIO Module address map :

Module Instance	Base Address
gpio0	0xFF708000
gpio1	0xFF709000
gpio2	0xFF70A000

Source [3]

General-Purpose I/O Interface

- Principaux registres

Source [3]

Register	Offset	Width	Accesses	Reset Value	Description
gpio_swporta_dr on page 22-5	0x0	32	RW	0x0	Port A Data Register
gpio_swporta_ddr on page 22-6	0x4	32	RW	0x0	Port A Data Direction Register
gpio_inten on page 22-7	0x30	32	RW	0x0	Interrupt Enable Register
gpio_intmask on page 22-8	0x34	32	RW	0x0	Interrupt Mask Register
gpio_inttype_level on page 22-9	0x38	32	RW	0x0	Interrupt Level Register
gpio_int_polarity on page 22-10	0x3C	32	RW	0x0	Interrupt Polarity Register
gpio_intstatus on page 22-11	0x40	32	RO	0x0	Interrupt Status Register
gpio_debounce on page 22-13	0x48	32	RW	0x0	Debounce Enable Register
gpio_porta_eoi on page 22-14	0x4C	32	WO	0x0	Clear Interrupt Register
gpio_ext_porta on page 22-15	0x50	32	RO	0x0	External Port A Register

Questions : GPIO

1. Est-il possible de connecter chaque périphérique sur toutes les GPIOs du HPS ?
2. Est-il garanti de pouvoir utiliser tous les périphériques du HPS quelque soit le boitier choisi ?

Voici les boitiers possible pour Cyclone V SE A5

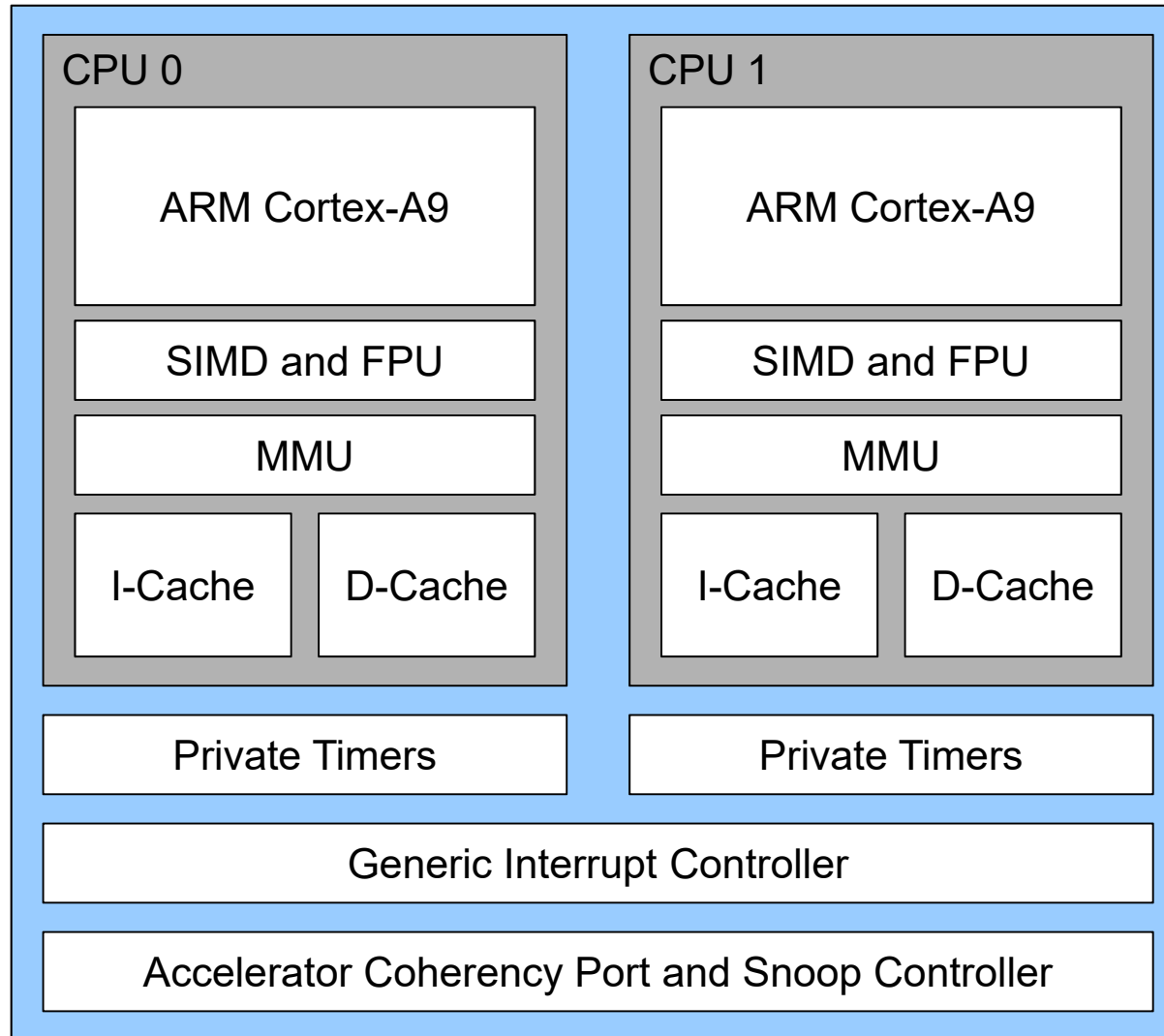
Member Code	U484 (19 mm)		U672 (23 mm)		F896 (31 mm)	
	FPGA GPIO	HPS I/O	FPGA GPIO	HPS I/O	FPGA GPIO	HPS I/O
A4	66	151	145	181	—	—
A5	66	151	145	181	288	181
A6	66	151	145	181	288	181

Questions : GPIO

3. Pourquoi l'utilisation des GPIO LoanIO nécessite 3 signaux connectés à la FPGA ?
4. Est-il possible d'interconnecter un périphérique du HPS sur la partie FPGA ?
Si oui: quelle page de configuration le permet-elle ?

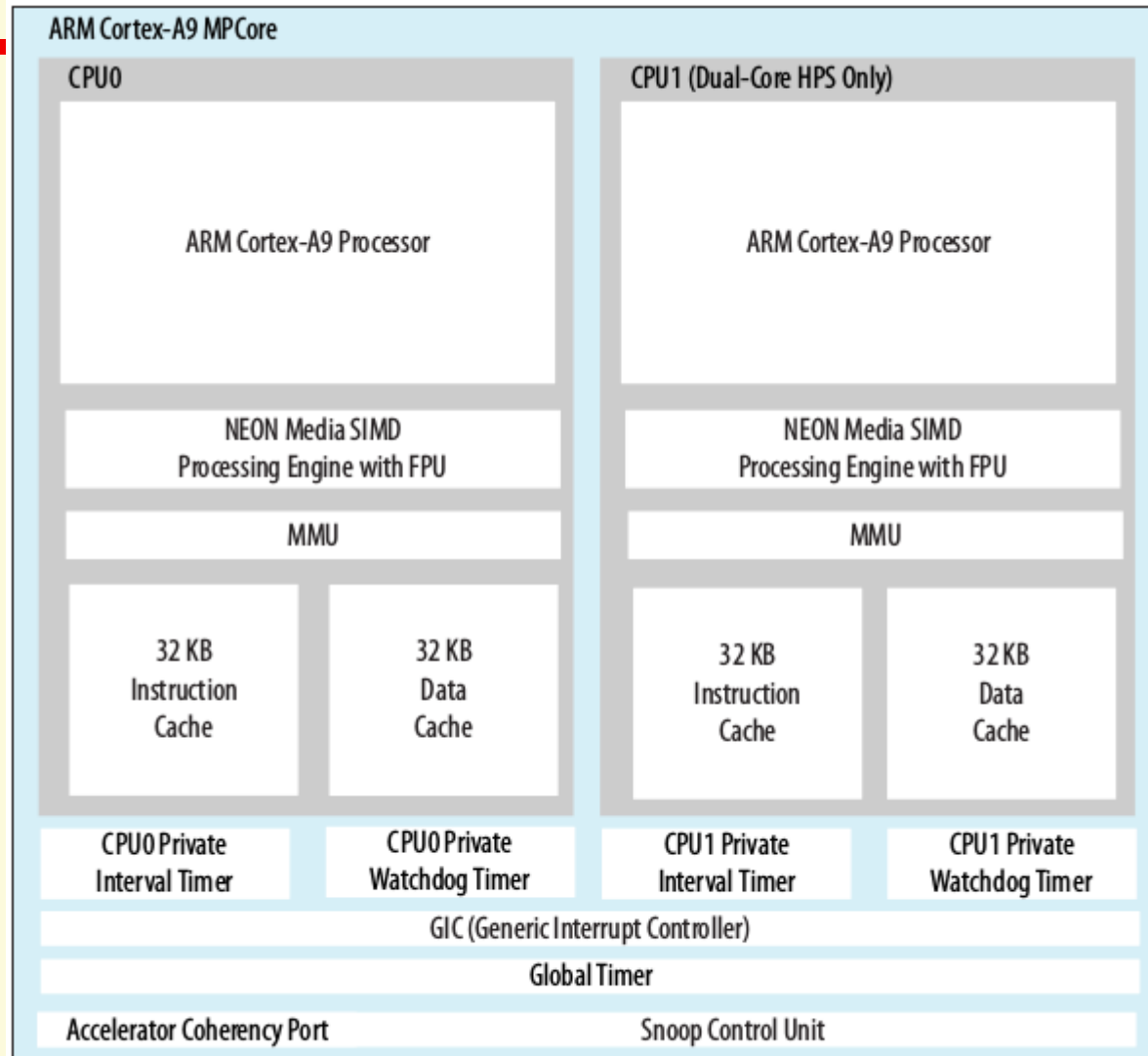
dia laissé vide volontairement

ARM Cortex-A9 MPCore



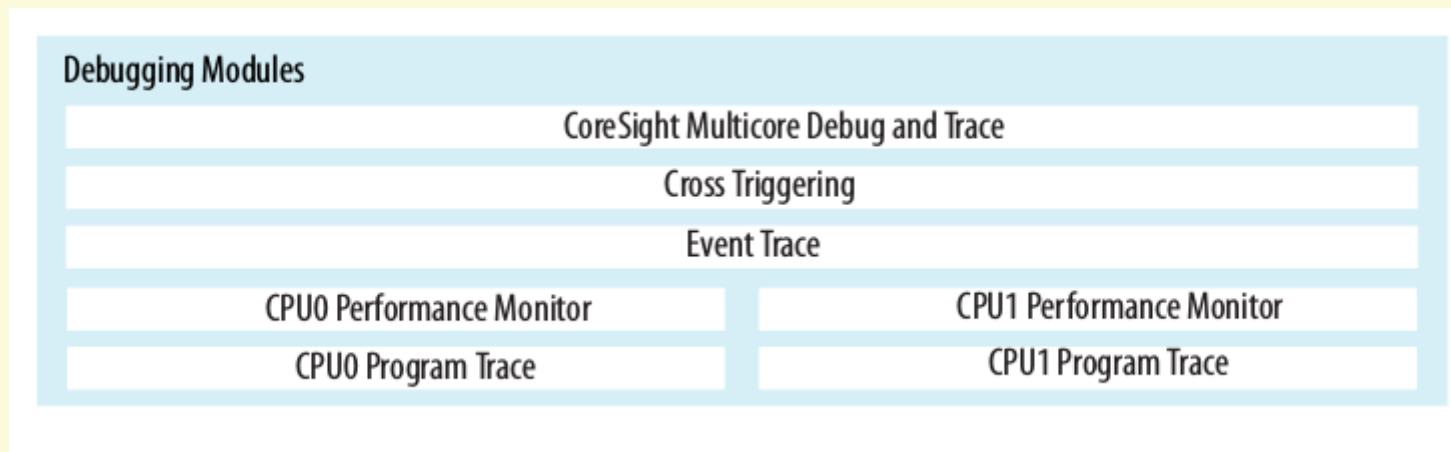
Cortex-A9 MPU Subsystem

- Detailed view



Cortex-A9 MPU Subsystem

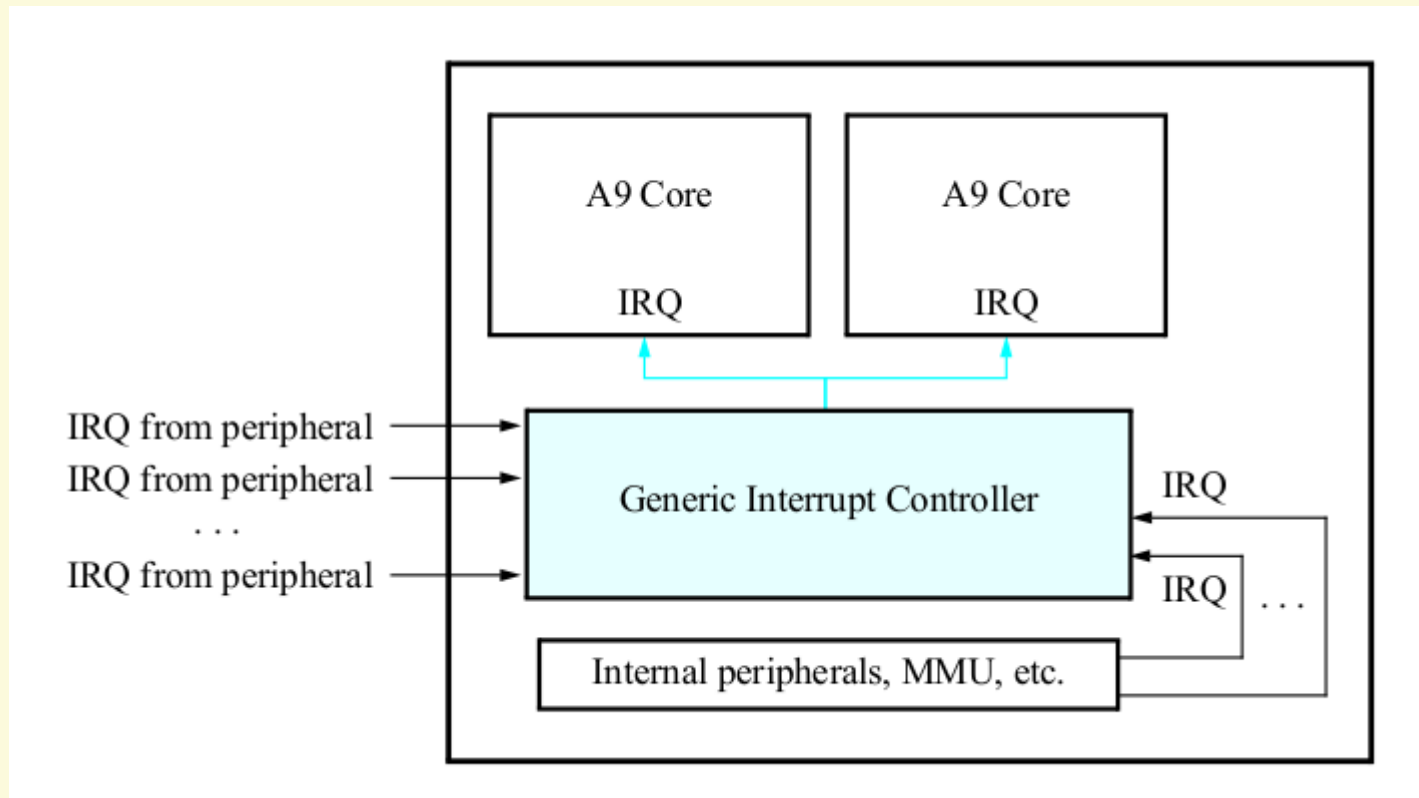
- Detailed view:
 - Debugging modules in the MPU Subsystem



Source [3]

Structure des interruptions

- Vue générale :



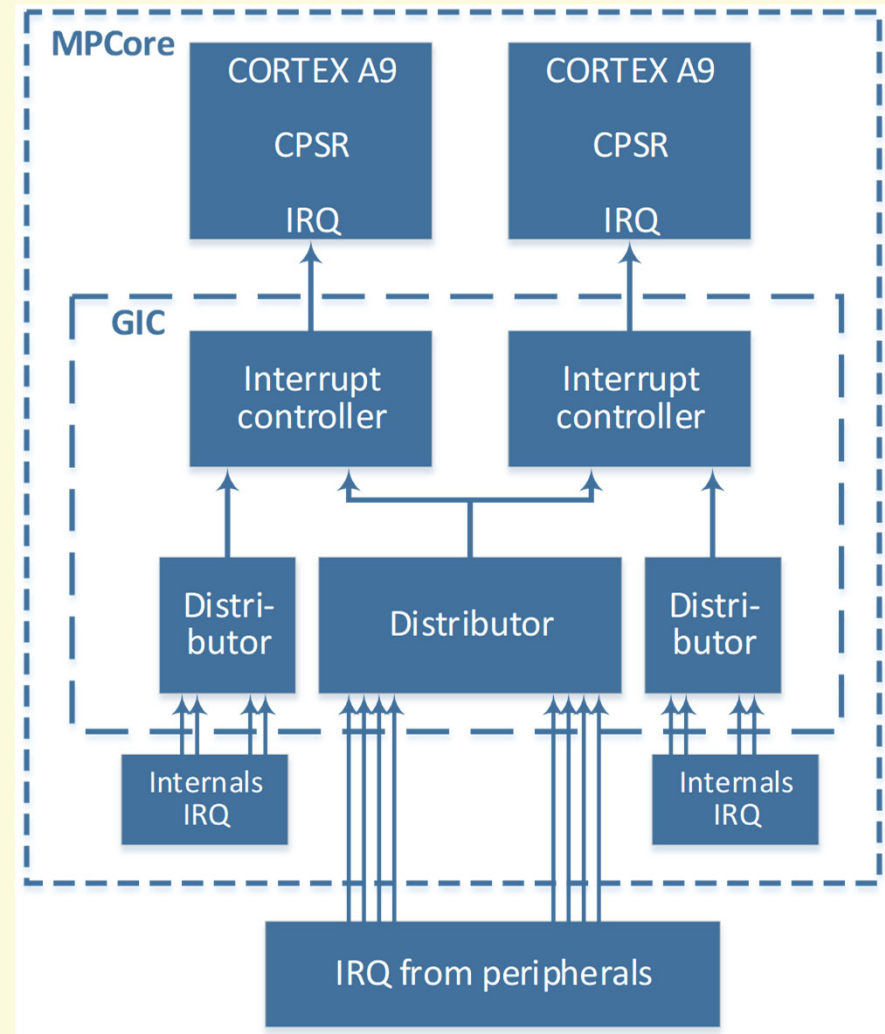
Source [6]

Types d'interruptions

- Le système comprend 3 sources d'interruptions :
 - Software Generated Interrupt (SGI)
 - Private Peripheral Interrupt (PPI)
 - Internal peripherals, MMU, etc.
 - Shared Peripheral Interrupt (SPI)
 - Peripheral from HPS
 - Ethernet, USB, Timer, ..
 - Peripheral from FPGA
 - IP : PIO, DMA, ...
 - User IP
 -

GIC, Generic Interrupt Controller

- Chaine propagation IRQ
 - CPU: CPSR
 - Activation générale
 - Priorité autorisée
 - Statuts général
 - Distributor
 - Activation
 - Priorité
 - Masque
 - Statuts
 - Source => ligne IRQ
 - Périphérique
 - Activation, masque
 - Niveau/flanc



GIC, Generic Interrupt Controller

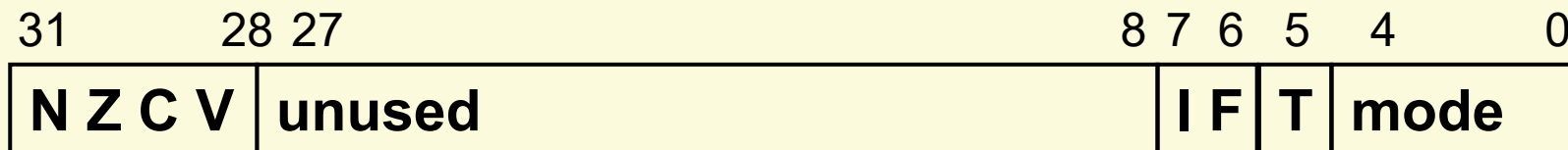
- Bloc générique développé par ARM
 - Version implémentée gère 256 interruptions
 - plan d'adressage permet d'augmenter ce nombre
 - Composer de 2 blocs:
 - Distributor
 - Interrupt controller
 - Gestion priorité des demande d'interruption
 - Transmet IRQ au CPU
 - Permet de gérer jusqu'à 8 cœurs CPU
 - Documentation ARM complète:
 - PrimeCell Generic Interrupt Controller (PL390), v r0p0, Technical Reference Manual

Gestion des interruptions

- La gestion des interruptions nécessite les étapes suivantes:
 - Configuration des interruptions dans les composants suivants
 - CPU
 - GIC (Generic Interrupt Controller)
 - Périphériques (PIO)
 - Configurer la table des vecteurs d'interruption
 - Lancement de la routine d'interruption
 - Routine d'interruption
 - quittance de la source de interruption
 - action à faire

Gestion interruption : CPU

- CPSR : Current Program Status Register
 - Bits de condition (N Z C V)
 - Bits de masquage des interruptions (I F)
 - I=1 => IRQ désactivée
 - F=1 => FIQ désactivée
 - Jeu d'instructions (T)
 - Mode du processeur (mode)



Gestion interruption : CPU

- CPSR : Current Program Status Register
 - Mode du processeur (mode) indiqué par M[4:0]

Code	Mode CPU
10000	USER
10001	FIQ
10010	IRQ
10011	Supervisor
10111	Abort
11011	Undefined
11111	System

Chaine d'une interruption

Exercice:

- Etablir la chaine de transmission d'une interruption
 - source: une IO du HPS (GPIO 54)
 - Destination: CPU0

Donner le schéma de la chaine qui doit comprendre les éléments suivants:

- CPU0
- GIC, Generic Interrupt Controller
- Bank GPIO
- pin GPIO 54

Chaine d'une interruption

Exercice:

- Chaine pour interruption depuis pin HPS-GPIO 54:
 - GPIO 54 => Bank GPIO 1
 - Bank GPIO 1 => gpio1_IRQ => interruption 197 du GIC

196	GPIO0	gpio0_IRQ	—	Level
197	GPIO1	gpio1_IRQ	—	Level
198	GPIO2	gpio2_IRQ	—	Level

correspond à une des interruption SPIs

- Configuration du GIC pour IRQ 197
 - Document Altera: "Using the ARM Generic Interrupt Controller"
 - voir page suivante
- CPU0 : activer le bit I du CPSR

Chaine d'une interruption

Configuration du GIC pour IRQ 197 :

- Distributor Control Register (ICDDCR)
 - Active le distributor, il faut activer E = '1')
- Interrupt Set Enable Registers(ICDISERn)
 - calcul de l'adresse du register:
 $address = 0xFFED100 + (N \div 32) \times 4$
 $(197 \div 32) \times 4 = 24 = 0x18$ address reg = 0xFFED118
- Interrupt Priority Registers(ICDIPRn)
 - reset value : 0x00000000 => highest priority
 - calcul de l'adresse du register:
 $address = 0xFFED400 + (N \div 4) \times 4$
 $(197 \div 4) \times 4 = 196 = 0xC4$ address reg = 0xFFED4C4
- Interrupt Processor Targets Registers(ICDIPTRn)
 - Configure le CPU concerné, address offset same ICDIPRn
address reg = 0xFFED8C4

Chaine d'une interruption

Utilisation IRQ manipulation de laboratoire:

- Etablir la chaine de transmission d'une interruption
 - source: une IO de la FPGA
 - Destination: CPU0

Donner le schéma de la chaine qui doit comprendre les éléments suivants:

- CPU0
- GIC, Generic Interrupt Controller
- Périphérique PIO
- pin IO FPGA

Liens sur les sources

- [1] Slides Altera-Intel: Presentation for board DE1
 - Altera (Intel) University Program, Cyclone V SoC Tutorial
- [2] Slides Altera-Intel: WS_1_Intro_To_SoC_SW_Workshop.pdf
 - SW Development for Altera SoC, Devices Workshop
- [3] Intel-Altera, fichier: hps_5v4.pdf
 - Cyclone V Hard Processor System Technical Reference Manual
- [4] Intel-Altera, fichier: cv_51001.pdf
 - Cyclone V Device Overview
- [5] Terasic, fichier: DE1-SoC_User_manual_ref.pdf
 - DE1-SoC User Manual
- [6] Altera, fichier: gic_altera_manual_short.pdf
 - Manuel Altera sur l'utilisation du "Generic Interrupt Controller ARM"

FIN présentation

Questions

