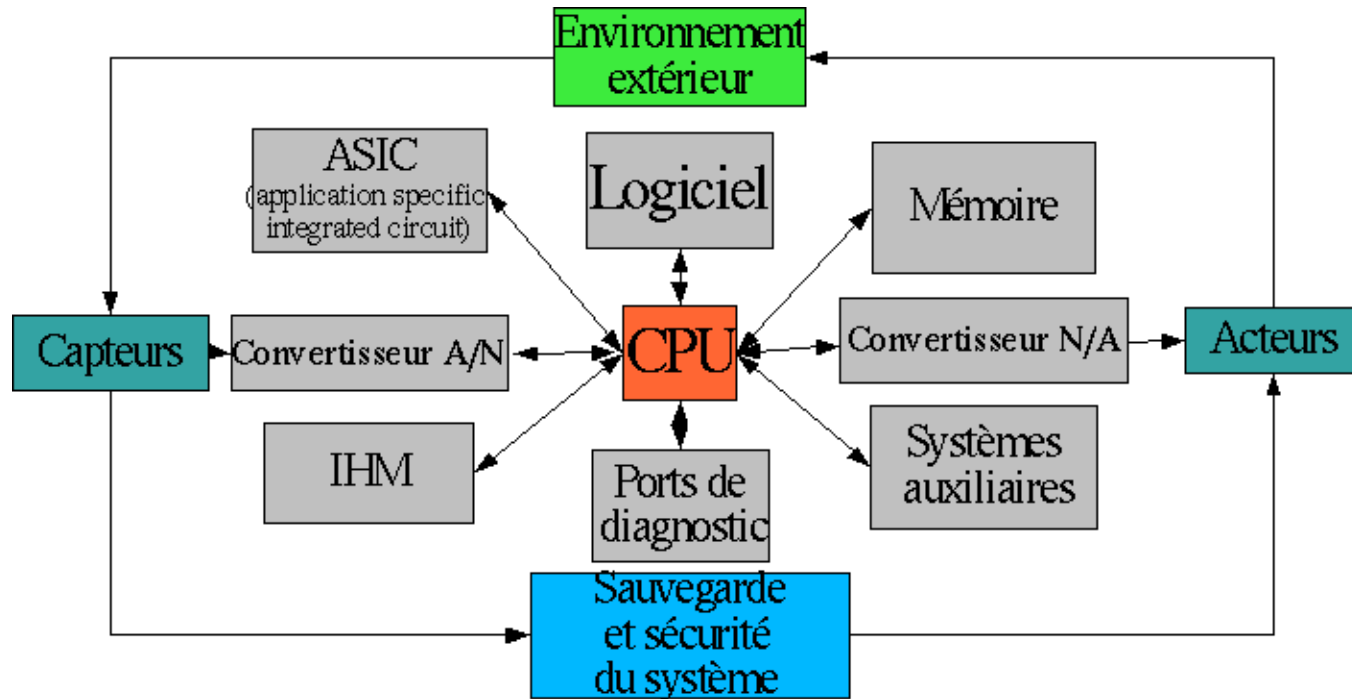


# Conception des SoPC

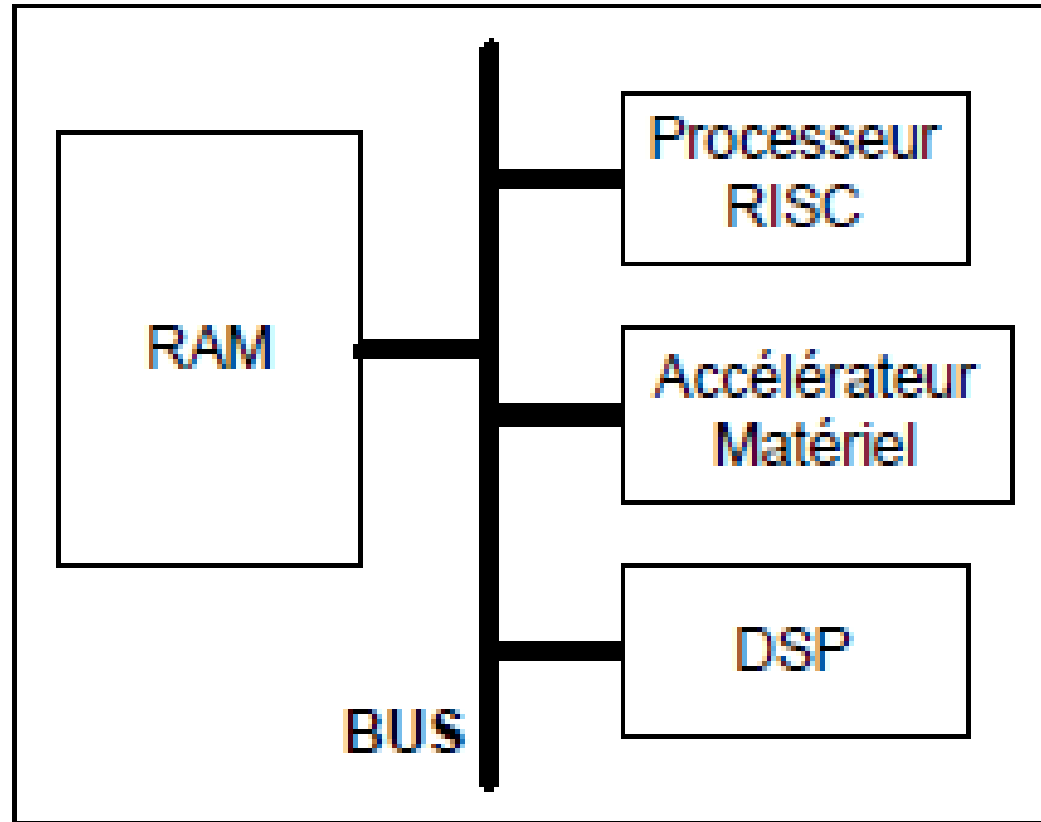
# Architecture d'un système embarqué (1)



Le fonctionnement du système se résume ainsi:

- Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique
- L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC et éventuellement de système externes traite l'information
- Le traitement génère éventuellement une sortie qui est envoyée vers la sortie, les systèmes auxiliaire, les ports de monitoring ou l'IHM

# Conception de SoPC



Système SoPC implanté dans un circuit logique programmable

# HW/SW CODESIGN

- ✚ L'approche SoPC : cohabitation de deux ressources logicielle et matérielle sur une même puce : codesign.
- ✚ Le codesign utilise le matériel et le logiciel pour une fonctionnalité à implanter :
  - ✓ Le logiciel est utilisé pour sa flexibilité.
  - ✓ Le matériel (i.e. FPGA et ASIC) est utilisé pour ses performances.

# Le besoin d'un système d'exploitation

- ✚ La logique programmée incluse dans les systèmes embarqués est de plus en plus complexe.
- ✚ Une approche logicielle de type super boucle (boucle infinie + interruptions) devient trop sommaire pour gérer et maîtriser la complexité.
- ✚ On a donc besoin d'utiliser un système d'exploitation offrant différents services nécessaires pour mieux gérer la complexité algorithmique logicielle.

# LE BESOIN D'UN SYSTÈME D'EXPLOITATION

✚ Un système d'exploitation offre ainsi différents services pour mieux appréhender la complexité :

- Apport du multitâche. Une application monolithique est divisée en une somme de tâches coopératives (système multitâche).
- Maîtrise des contraintes temporelles (système Temps Réel).
- Masquage des spécificités du matériel. On y accède de façon homogène et standard.
- Développement de pilotes de périphérique (driver) simplifié pour pouvoir avoir accès aux accélérateurs matériels.
- Apport d'un système de fichiers.
- Possibilité de communications réseau : pour un contrôle du système à distance par exemple.

## **LES PROCESSEURS POUR LE SOPC**

# Intérêt d'un processeur pour le SOPC

- ✚ Lorsque l'on conçoit un système numérique complexe, on met en oeuvre généralement un processeur embarqué.
- ✚ Ce processeur embarqué est :
  - ✚ Soit un bloc IP : on parle de processeur softcore.
  - ✚ Soit déjà implanté dans le circuit électronique en «dur»: on parle de processeur hardcore. Le processeur de ce type est généralement plus performant que le processeur du type précédent.



# Intérêt d'un processeur pour le SOPC

- ✚ Le processeur embarqué allie la souplesse du logiciel à l'accélération du temps d'exécution du matériel.
- ✚ Une fonctionnalité particulière peut donc être composée d'une partie matérielle couplée à une fonctionnalité logicielle dédiée : on a donc une conception conjointe matérielle-logicielle ou codesign.
- ✚ Le codesign implique donc une conception en même temps du matériel et du logiciel, ce qui est une nouvelle méthodologie par rapport à la méthodologie de conception classique (conception matérielle puis conception logicielle)...

# Choix du processeur:

✚ Le choix d'un processeur pour le SoPC peut se faire sur différents critères :

✚ Processeur hardcore : pour ses performances au détriment de la flexibilité.

✚ Processeur softcore : pour sa flexibilité de mise à jour au détriment de performances moindres que le précédent. La portabilité vers n'importe quel circuit FPGA est assurée en étant donc circuit FPGA indépendant. Il est aussi possible de migrer vers un circuit de type ASIC en cas d'une production en grande série.

✚ Généralement, on privilégie les processeurs softcore pour s'affranchir des problèmes d'obsolescence et pour pouvoir bénéficier facilement des évolutions apportées en refaisant une synthèse.

# Les processeurs softcore pour les SOPC

- ✚ Le processeur softcore peut être libre :
  - ✚ Il est décrit en langage de description de matériel (VHDL, Verilog).
  - ✚ Le code source peut être librement distribué et implanté dans n'importe quel circuit programmable FPGA.
  - ✚ On est alors indépendant du type de circuit FPGA.

# Les processeurs softcore pour les SOPC

- ✚ Le processeur softcore peut être propriétaire :
  - ✚ Il est distribué par exemple sous forme d'une netlist pour être implantée dans un circuit FPGA.
  - ✚ Il est généralement lié à un fondeur de circuit FPGA particulier (comme Altera ou Xilinx).
  - ✚ On ne peut pas l'utiliser dans un circuit FPGA autre que celui pour lequel il est prévu. On a donc ici une boîte noire.

# Les processeurs softcore pour les SOPC

✚ On trouvera principalement au niveau des processeurs softcore libres :

✚ Le processeur Leon <http://www.gaisler.com/index.html>.

✚ Le processeur OpenRisc  
<http://www.opencores.org/projects.cgi/web/or1k/overview>.

✚ Autre processeur : F-CPU <http://www.f-cpu.org>.

✚ Autres processeurs : clones de 6800, 68HC11, 68K, PIC :

✚ [http://www.opencores.org/browse.cgi/filter/category\\_microprocessor](http://www.opencores.org/browse.cgi/filter/category_microprocessor)

✚ ...

# Les processeurs softcore pour les SOPC

✚ On trouvera principalement au niveau des processeurs softcore propriétaires :

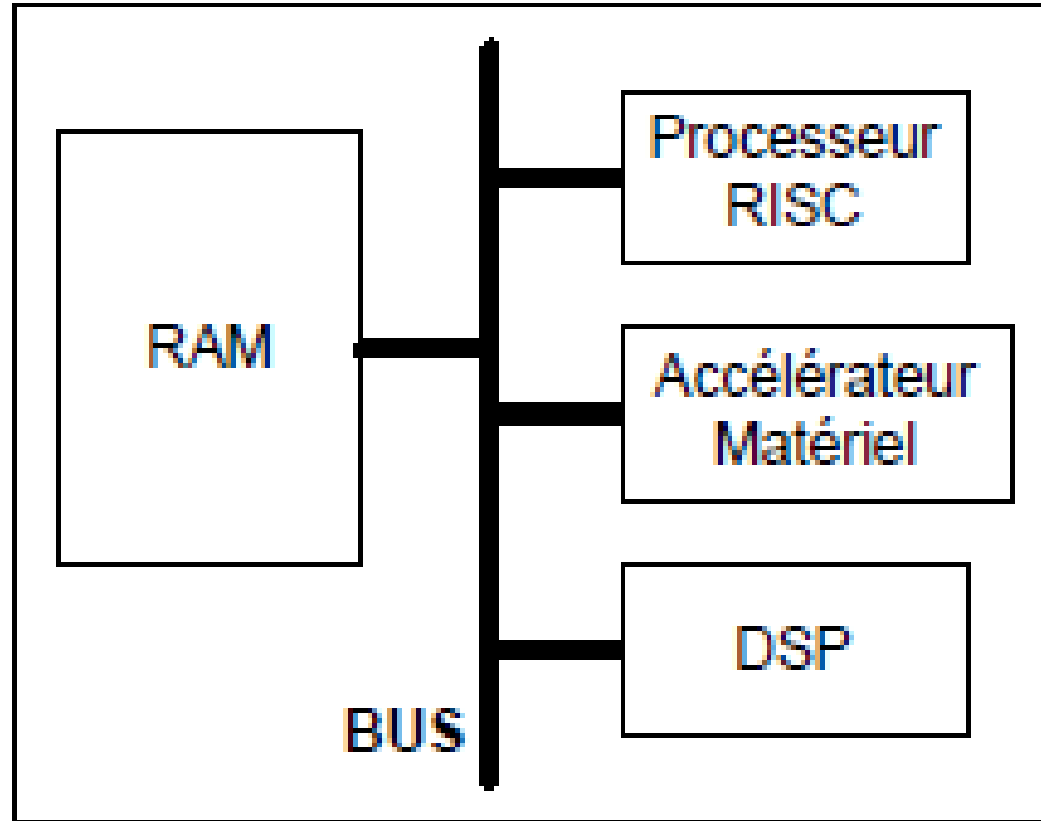
✚ Le processeur NIOS et NIOS II d'Altera  
<http://www.altera.com>.

✚ Le processeur Microblaze de Xilinx  
<http://www.xilinx.com>.

✚ ...

# Conception de SoPC

# Conception de SoPC



Système SoPC implanté dans un circuit logique programmable



# HW/SW CODESIGN

- L'approche SoPC : cohabitation de deux ressources logicielle et matérielle sur une même puce : codesign.
- Le codesign utilise le matériel et le logiciel pour une fonctionnalité à implanter :
  - ✓ Le logiciel est utilisé pour sa flexibilité.
  - ✓ Le matériel (i.e. FPGA et ASIC) est utilisé pour ses performances.

# Le besoin d'un système d'exploitation

- ✓ La logique programmée incluse dans les systèmes embarqués est de plus en plus complexe.
- ✓ Une approche logicielle de type super boucle (boucle infinie + interruptions) devient trop sommaire pour gérer et maîtriser la complexité.
- ✓ On a donc besoin d'utiliser un système d'exploitation offrant différents services nécessaires pour mieux gérer la complexité algorithmique logicielle.

# LE BESOIN D'UN SYSTÈME D'EXPLOITATION

❖ Un système d'exploitation offre ainsi différents services pour mieux appréhender la complexité :

- Apport du multitâche. Une application monolithique est divisée en une somme de tâches coopératives (système multitâche).
- Maîtrise des contraintes temporelles (système Temps Réel).
- Masquage des spécificités du matériel. On y accède de façon homogène et standard.
- Développement de pilotes de périphérique (driver) simplifié pour pouvoir avoir accès aux accélérateurs matériels.
- Apport d'un système de fichiers.
- Possibilité de communications réseau : pour un contrôle du système à distance par exemple.

# **LES PROCESSEURS POUR LE SOPC**

# Intérêt d'un processeur pour le SOPC

- Lorsque l'on conçoit un système numérique complexe, on met en oeuvre généralement un processeur embarqué.
- Ce processeur embarqué est :
  - ❖ Soit un bloc IP : on parle de processeur softcore.
  - ❖ Soit déjà implanté dans le circuit électronique en «dur»: on parle de processeur hardcore. Le processeur de ce type est généralement plus performant que le processeur du type précédent.

# Intérêt d'un processeur pour le SOPC

- Le processeur embarqué allie la souplesse du logiciel à l'accélération du temps d'exécution du matériel.
- Une fonctionnalité particulière peut donc être composée d'une partie matérielle couplée à une fonctionnalité logicielle dédiée : on a donc une conception conjointe matérielle-logicielle ou codesign.
- Le codesign implique donc une conception en même temps du matériel et du logiciel, ce qui est une nouvelle méthodologie par rapport à la méthodologie de conception classique (conception matérielle puis conception logicielle)...

# Choix du processeur:

➤ Le choix d'un processeur pour le SoPC peut se faire sur différents critères :

- ❖ Processeur hardcore : pour ses performances au détriment de la flexibilité.

- ❖ Processeur softcore : pour sa flexibilité de mise à jour au détriment de performances moindres que le précédent. La portabilité vers n'importe quel circuit FPGA est assurée en étant donc circuit FPGA indépendant. Il est aussi possible de migrer vers un circuit de type ASIC en cas d'une production en grande série.

➤ Généralement, on privilégie les processeurs softcore pour s'affranchir des problèmes d'obsolescence et pour pouvoir bénéficier facilement des évolutions apportées en refaisant une synthèse.

# Les processeurs softcore pour les SOPC

- Le processeur softcore peut être libre :
  - ✓ Il est décrit en langage de description de matériel (VHDL, Verilog).
  - ✓ Le code source peut être librement distribué et implanté dans n'importe quel circuit programmable FPGA.
  - ✓ On est alors indépendant du type de circuit FPGA.



# Les processeurs softcore pour les SOPC

- Le processeur softcore peut être propriétaire :
  - ✓ Il est distribué par exemple sous forme d'une netlist pour être implantée dans un circuit FPGA.
  - ✓ Il est généralement lié à un fondeur de circuit FPGA particulier (comme Altera ou Xilinx).
  - ✓ On ne peut pas l'utiliser dans un circuit FPGA autre que celui pour lequel il est prévu. On a donc ici une boîte noire.

# Les processeurs softcore pour les SOPC

➤ On trouvera principalement au niveau des processeurs softcore libres :

- ✓ Le processeur Leon <http://www.gaisler.com/index.html>.

- ✓ Le processeur OpenRisc

  - <http://www.opencores.org/projects.cgi/web/or1k/overview>.

- ✓ Autre processeur : F-CPU <http://www.f-cpu.org>.

- ✓ Autres processeurs : clones de 6800, 68HC11, 68K, PIC :

- ✓ [http://www.opencores.org/browse.cgi/filter/category\\_microprocessor](http://www.opencores.org/browse.cgi/filter/category_microprocessor)

- ✓ ...

# Les processeurs softcore pour les SOPC

➤ On trouvera principalement au niveau des processeurs softcore propriétaires :

✓ Le processeur NIOS et NIOS II d'Altera  
<http://www.altera.com>.

✓ Le processeur Microblaze de Xilinx  
<http://www.xilinx.com>.

✓ ...

# **LE PROCESSEUR SOFTCORE MICROBLAZE**

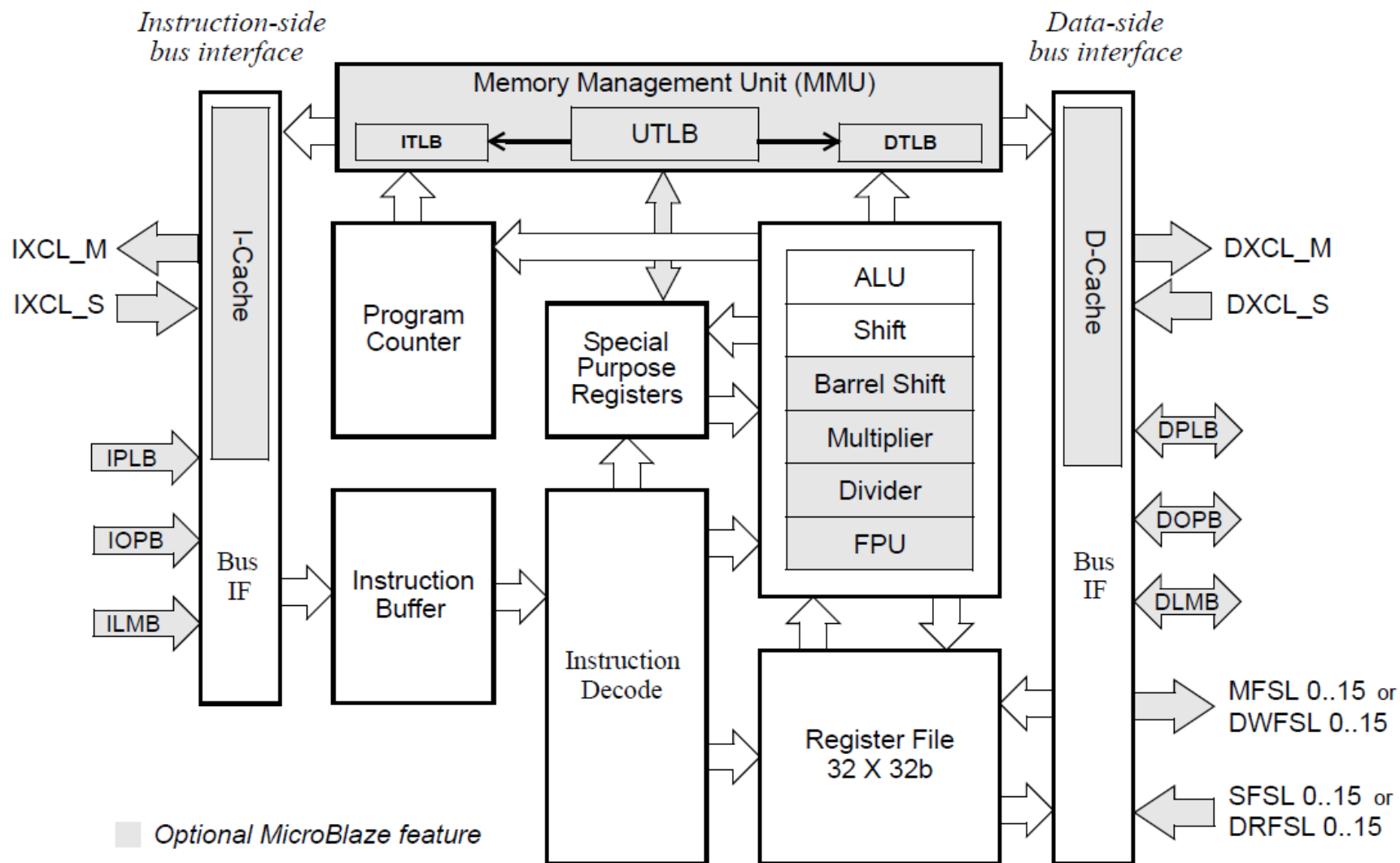


Figure 1-1: MicroBlaze Core Block Diagram

# Le processeur softcore MicroBlaze

## *MicroBlaze : ARCHITECTURE*

- Jeu d'instructions 32 bits.
- 32 registres de données.
- Pipeline à 3 niveaux.
- Cache d'instructions et de données configurable.
- Bus interne LMB et CoreConnect IBM.
- Pas de MMU.
- Produit commercial de Xilinx.

# Le processeur softcore MicroBlaze

## *MicroBlaze : LOGICIELS*

- Plateforme de développement Xilinx XPS.
- Chaîne d'outils GNU (compilation croisée).
- Simulateur XMD.
- OS supportés :
- $\mu$ Clinux. Portage GPL

<http://www.itee.uq.edu.au/~jwilliams/mblaze-uclinux/>.

- microC/OS II. Produit commercial.
- Noyau ATI. Produit commercial.

## **LE PROCESSEUR SOFTCORE NIOS II**



# Le processeur softcore NIOS II

## *NIOS II : ARCHITECTURE*

- ✚ 2 versions : NIOS I et NIOS II. Seule NIOS II présentée.
- ✚ 3 cores possibles : fast, economy, standard.
- ✚ Jeu d'instructions 32 bits.
- ✚ 32 registres dont 6 de contrôle.
- ✚ Pipeline à 6 niveaux (fast).
- ✚ Cache d'instructions et de données configurable.
- ✚ Bus interne Avalon.
- ✚ Pas de MMU.
- ✚ Produit commercial d'Altera.

# Le processeur softcore NIOS II

## *NIOS II : LOGICIELS*

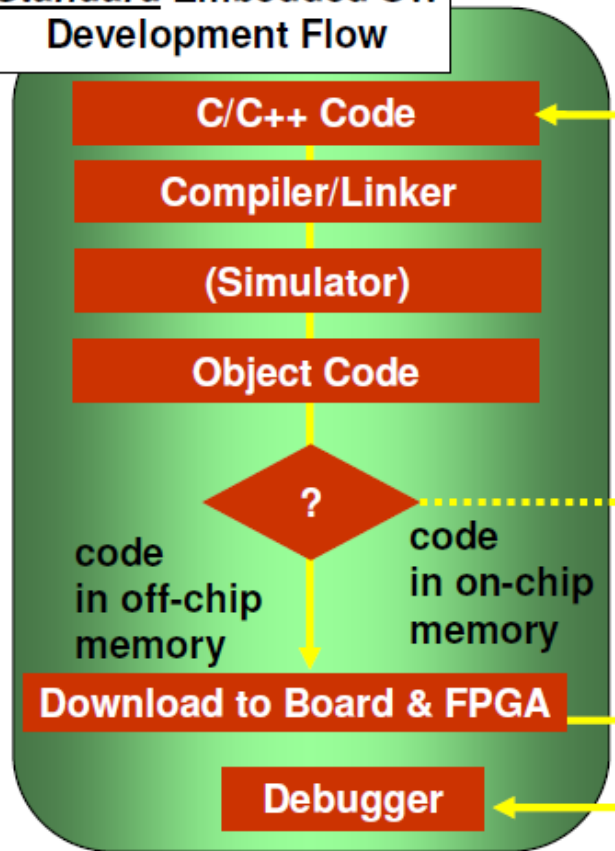
- ✚ Plateforme de développement Quartus II.
- ✚ Chaîne d'outils GNU (compilation croisée).
- ✚ IDE Eclipse
- ✚ Simulateur ModelSim.
- ✚ OS supportés :
  - ✚  $\mu$ Clinux. Portage GPL
    - ✚ Version intégré à l'IDE : <http://www.niosforum.com/>
    - ✚ Version sous Linux : <http://nioswiki.jot.com/WikiHome>
  - ✚ microC/OS II. Produit commercial.
  - ✚ Noyau Nucleus. Produit commercial.

# **PLATEFORME POUR LA MISE EN OEUVRE DU CODESIGN**

# EDK System Design

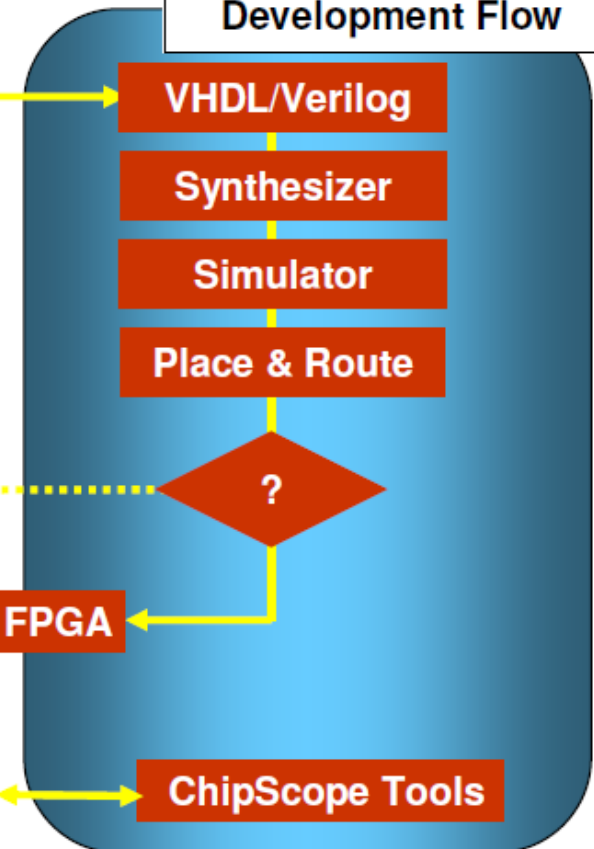
## Comprehensive Tool Chain

### Standard Embedded SW Development Flow



Xilinx Platform Studio (XPS)

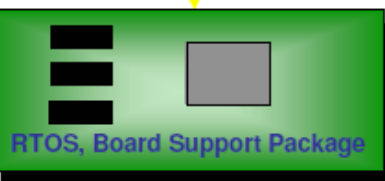
### Standard FPGA HW Development Flow



Data2BlockRAM

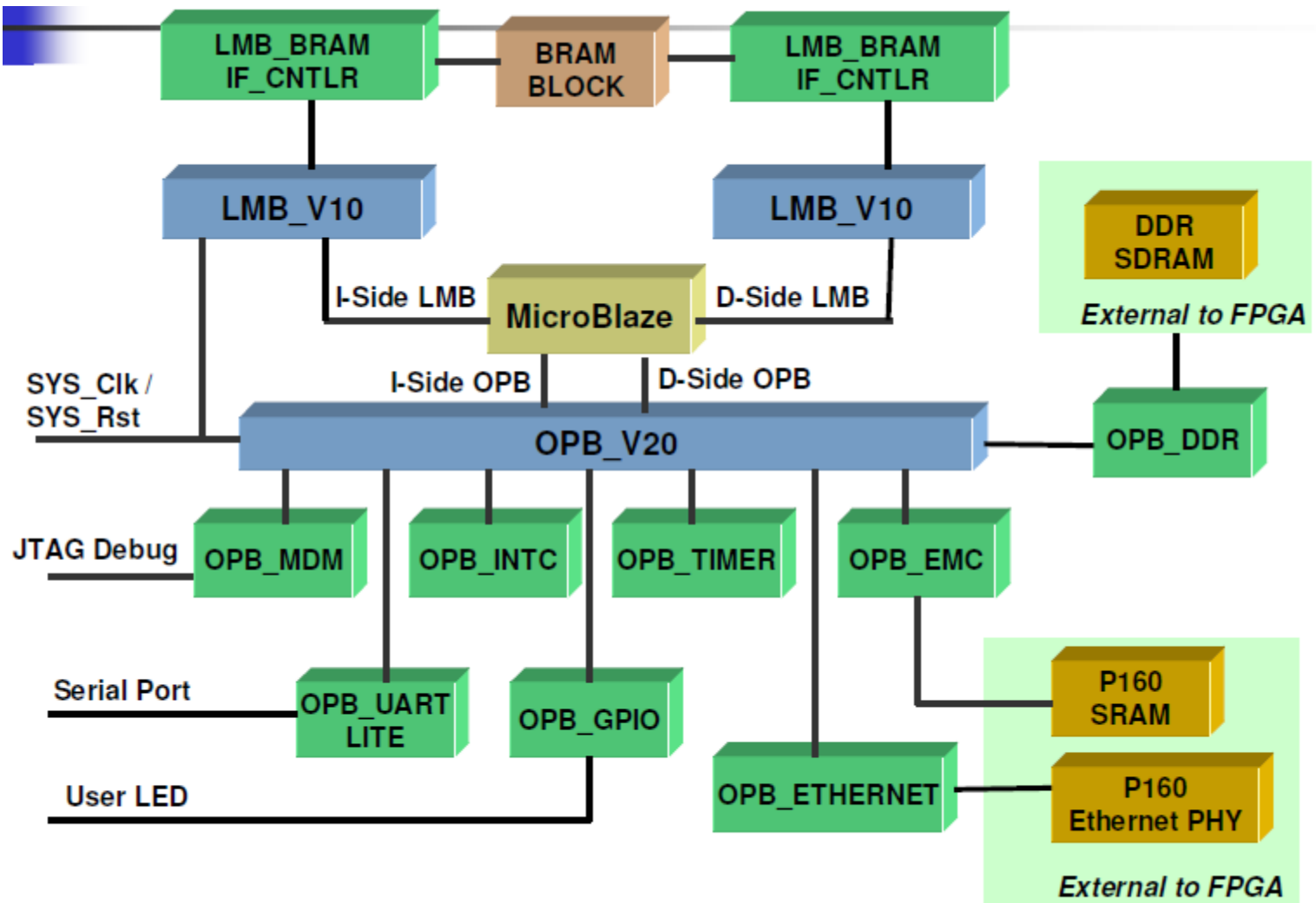
Bitstream

Download to FPGA

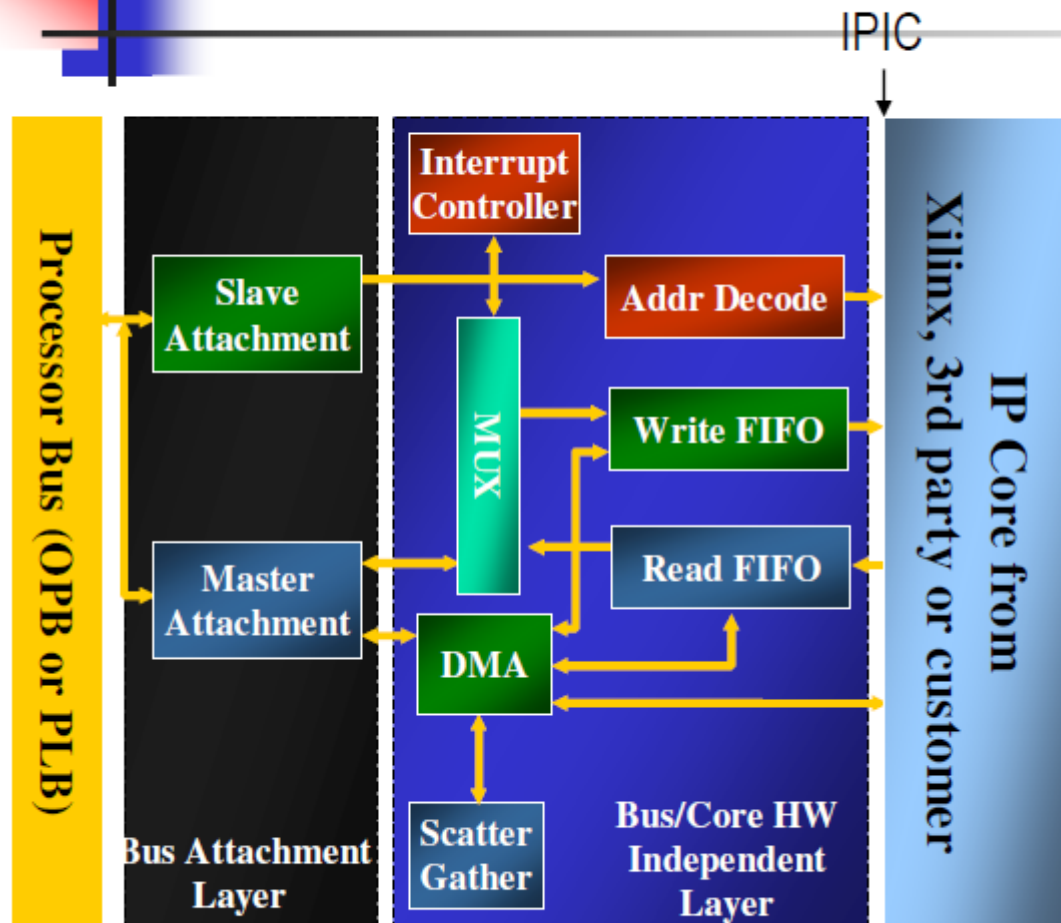


ChipScope Tools

# Exemple Microblaze System



# Full IP Interface (IPIF)



- Consists of 8 modules
  - Each module is selectable and parameterizable
- Automatically configures a core to the processor bus
  - Xilinx IP Cores
  - 3rd Party IP Cores
  - Customer proprietary cores and external devices
- OPB & PLB supported
  - Bus independent IP Cores and associated Device Drivers
- IPIF will be added to other LogiCOREs

# La famille Zynq

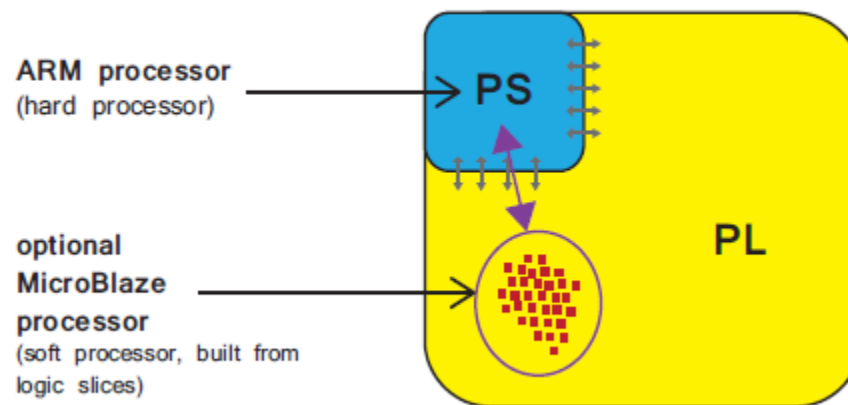
- Dans ce paragraphe, nous allons expliquer ce que Xilinx a modifié (par rapport à la Spartan 3 qui date de 2003) dans un FPGA plus récent : la Zynq qui fait partie de la famille des FPGA de la série 7.

	ARTIX <sup>7</sup>	KINTEX <sup>7</sup>	VIRTEX <sup>7</sup>
Maximum Capability	Lowest Power and Cost	Industry's Best Price/Performance	Industry's Highest System Performance
Logic Cells	20K – 355K	70K – 480K	285K – 2,000K
Block RAM	12 Mb	34 Mb	65 Mb
DSP Slices	40 – 700	240 – 1,920	700 – 3,960
Peak DSP Perf.	504 GMACS	2,450 GMACs	5,053 GMACS
Transceivers	4	32	88
Transceiver Performance	3.75Gbps	6.6Gbps and 12.5Gbps	12.5Gbps, 13.1Gbps and 28Gbps
Memory Performance	1066Mbps	1866Mbps	1866Mbps
I/O Pins	450	500	1,200
I/O Voltages	3.3V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below

Nous trouvons dans la Zynq deux parties principales :

- la logique programmable (PL)
- et un système à microprocesseur ARM complet avec ses périphériques (PS).

Du point de vue de la logique programmable, la famille Zynq est à cheval entre la famille Artix et la famille Kintex.

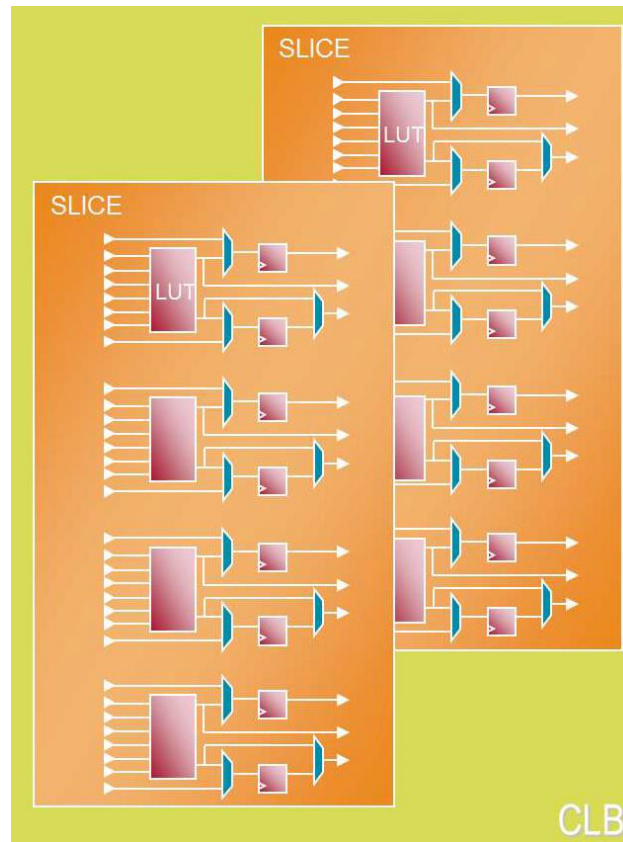




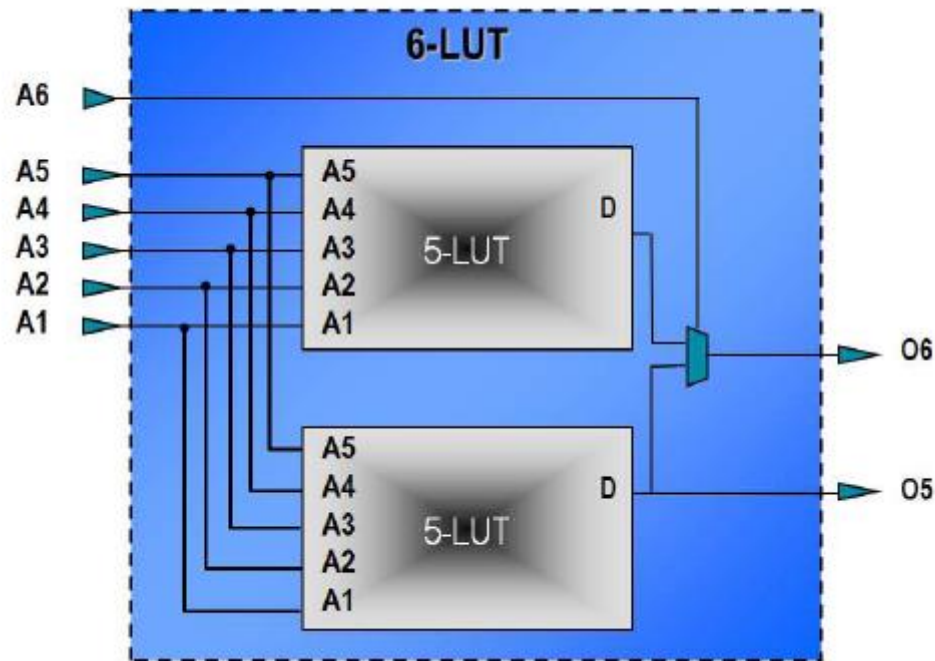
# Logique programmable (PL)

- **CLB/Slices**

- Il y a maintenant deux slices par CLB : les slice\_M (ROM, logique, RAM et registres à décalage), et les slice\_L (ROM et logique). On a 4 LUT et 8 bascules D par slice.

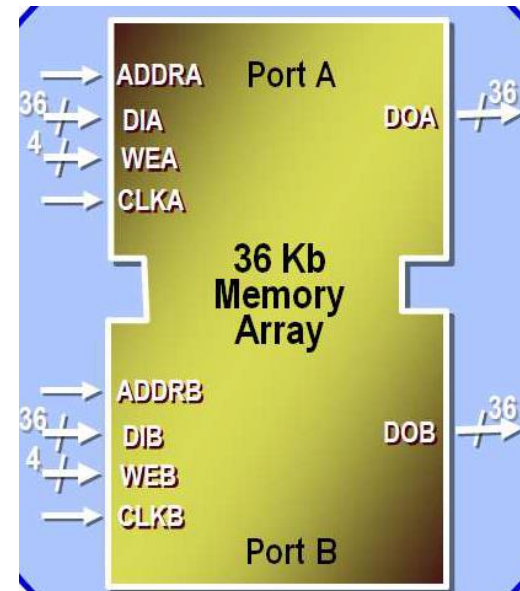


Les LUT sont passées à 6 entrées. On peut les utiliser en une fonction combinatoire à 6 entrées ou en 2 fonctions combinatoires indépendantes à 5 entrées.



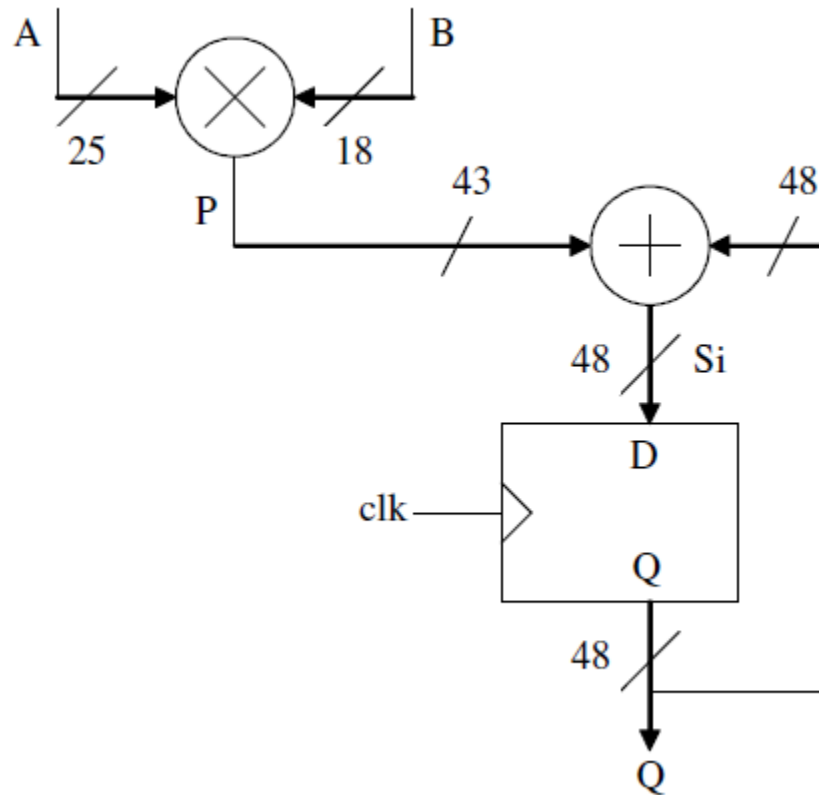
# Block RAM

- Les blocs RAM sont passés à 36 kbits mais peuvent être utilisés en deux blocs 18 kbits indépendants. Ils ont les caractéristiques suivantes :
  - De 32kx1 à 512x72 dans un bloc 36k.
  - Mémoire ROM simple port à mémoire RAM double port.
  - Configurable en FIFO sans logique externe.
  - Code correcteur d'erreur possible.
  - 2 blocs 36k adjacents forment un bloc 64k sans logique supplémentaire.



# DSP Slice

- Le multiplieur de la Spartan 3 a évolué en un bloc MAC (Multiply and ACcumulate) qui est la brique de base d'un Digital Signal Processor (DSP) :



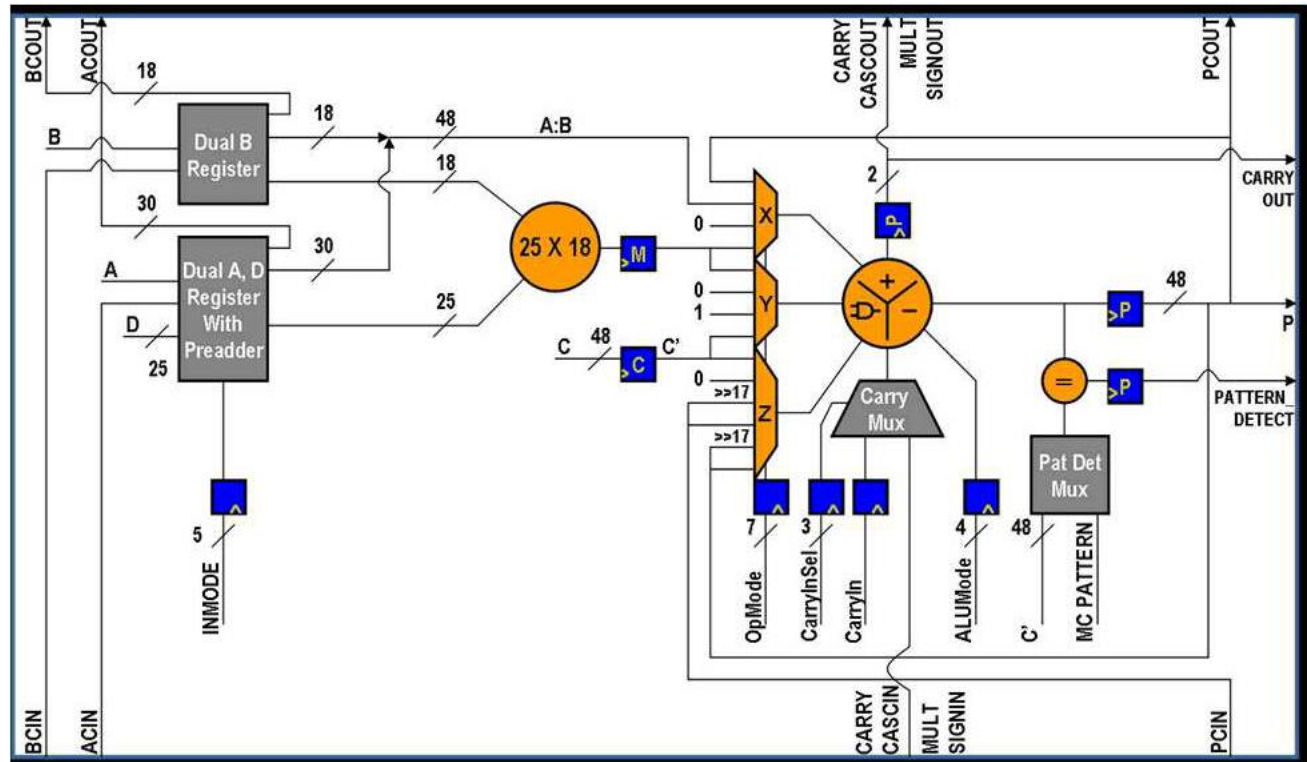
**Equation réalisée :**

$$Q = \sum_{i=1}^L A_i \cdot B_i$$

**Exemple d'utilisation : la multiplication-accumulation permet de calculer un produit de convolution. C'est l'opération de base du filtrage numérique.**

Dans la série 7, le MAC est hautement polyvalent d'où sa complexité apparente.

Il est conçu pour être cascadié avec d'autres DSP slices afin de réaliser des filtres numériques rapides (500 MHz à comparer avec les 250 MHz de la logique programmable usuelle) :



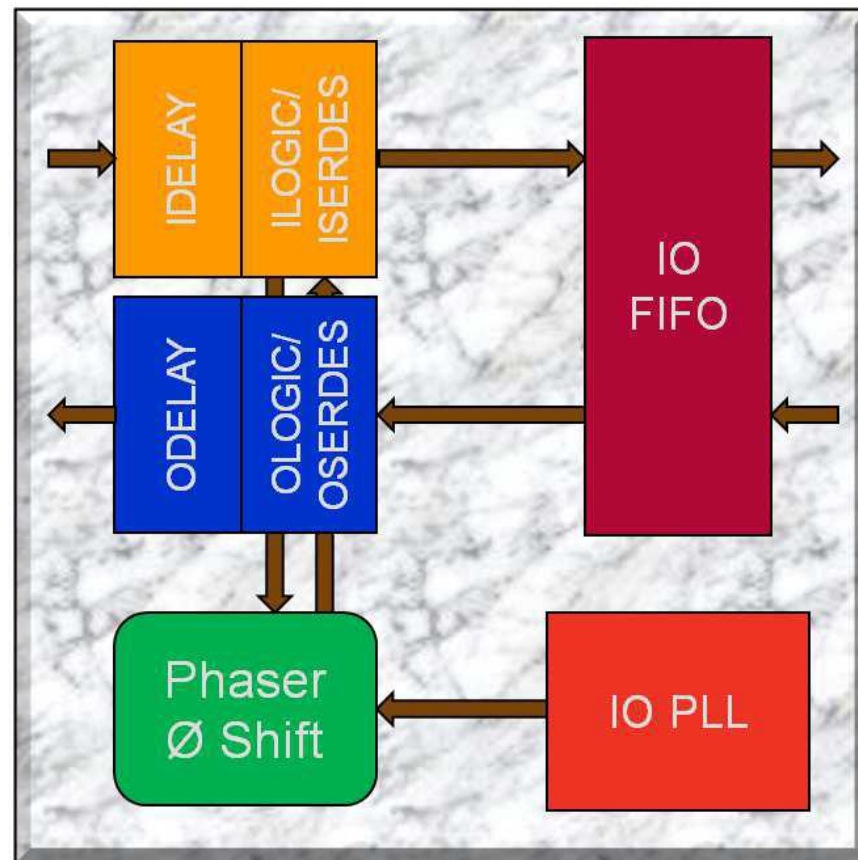
# Input/Output block (IOB)

- Les IOB peuvent être de deux types : High range ou High performance :

I/O Types	Artix-7 Family	Kintex-7 Family	Virtex-7 Family	Virtex-7 XT/HT Family
High Range	All	Most	Some	
High Performance		Some	Most	All

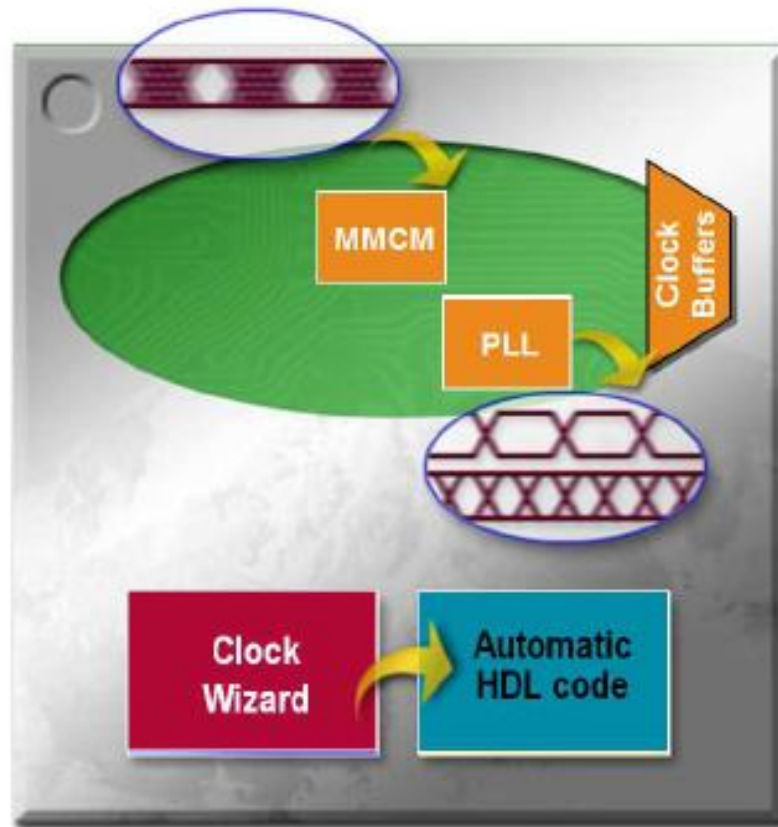
- Les IOB « high range » supportent des tensions jusqu'à 3.3 V (unipolaire ou différentiel, sans digitally controlled impedance DCI).
- Les IOB « high performance » supportent les standards jusqu'à 1.8 V (unipolaire ou différentiel avec DCI).

- Il y a des SERDES (sérialiseur/désérialiseur) et des fifo dans chaque IOB et les retards sont beaucoup plus précis (32 pas, le retard élémentaire étant calibré par une horloge).
- Les broches ont des ressources spécifiques (IO PLL et IO phaser shift) pour s'interfacer avec des mémoires de type DDR3.



# Ressources d'horloges

- Il y a beaucoup plus de ressources d'horloge dans le FPGA avec des horloges globales et régionales. Il n'y a quasiment plus de restriction sur le nombre d'horloges.



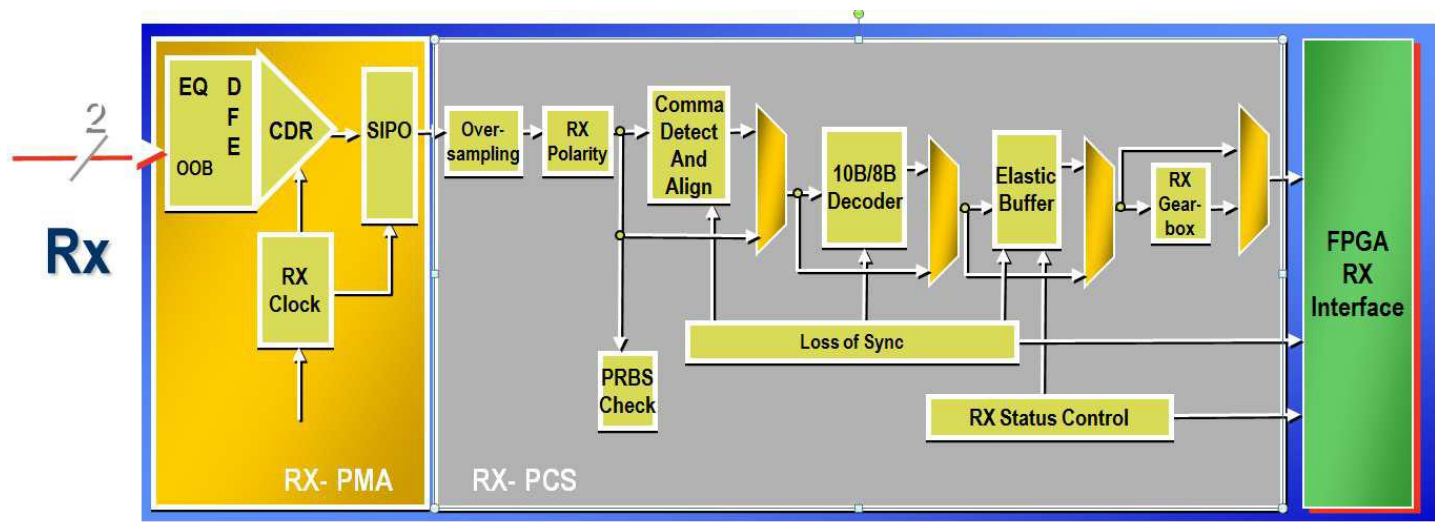


- Les Mixed-Mode Clock Managers (MMCM) sont les évolutions des DCM de la Spartan 3.
- Les performances en synthèse d'horloge et en déphasage se sont considérablement améliorées.
- Les Phase Locked Loop (PLL) sont conçues pour travailler avec les IO phaser pour contrôler de la mémoire.

# High-Speed Serial I/O transmitter/receiver (Transceivers)

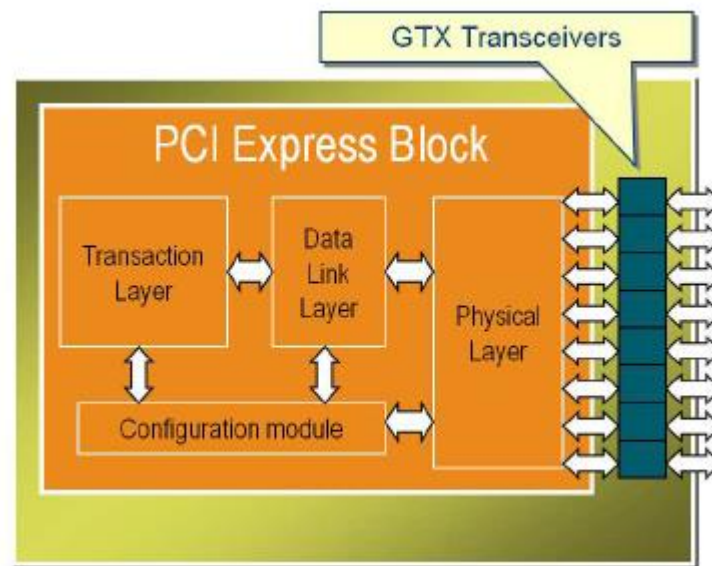
- Les liaisons séries rapides ( $\gg 1\text{Gbits/s}$ ) sont de plus en plus utilisées en électronique car la plupart des interfaces sont passées de bus parallèles à interface série (USB, IEEE-1394 « Firewire », SATA, PCI Express, ...). De plus, les interfaces réseau Ethernet ne cessent de monter en débit (100 Gbit/s avec IEEE-802.3ba-2010). Il ne s'agit plus de broche d'entrée/sortie traditionnelle, mais d'une liaison de type télécom avec une reconstruction de l'horloge indépendante pour chaque broche.

- Dans la série 7, il existe 4 familles de transceivers, mais la Zynq n'utilise que les 3 premières :
  - GTP transceivers jusqu'à 3.75 Gbps (PCI Express 1, Gigabit Ethernet, ...),
  - GTX transceivers jusqu'à 12.5 Gbps (PCI Express 1/2/3, 10 Gigabit Ethernet, ...),
  - GTH transceivers jusqu'à 13.1 Gbps (PCI Express 1/2/3, 10 Gigabit Ethernet avec correction d'erreurs, ...),
  - GTZ transceivers jusqu'à 28 Gbps (100 Gigabit Ethernet, ...).



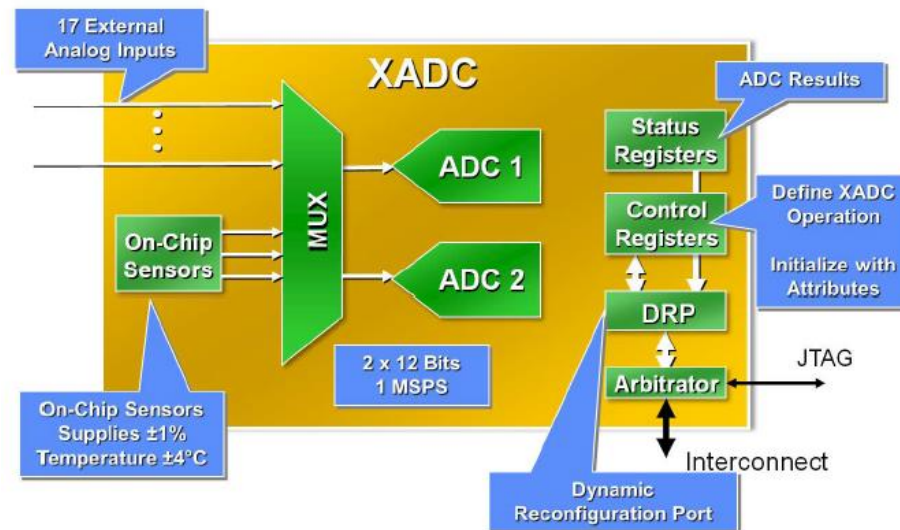
# Interface PCI Express

- La norme PCI Express est universellement implantée depuis 2005 dans le domaine de l'informatique. C'est une liaison série qui a remplacé le bus parallèle PCI. Sa bande passante s'adapte aux besoins :
  - Gen1 (2.5Gbps), Gen2, (5.0Gbps), Gen3 (8Gbps).
  - Configurations multi-voies (x1, x2, x4, x8, x16, x32).



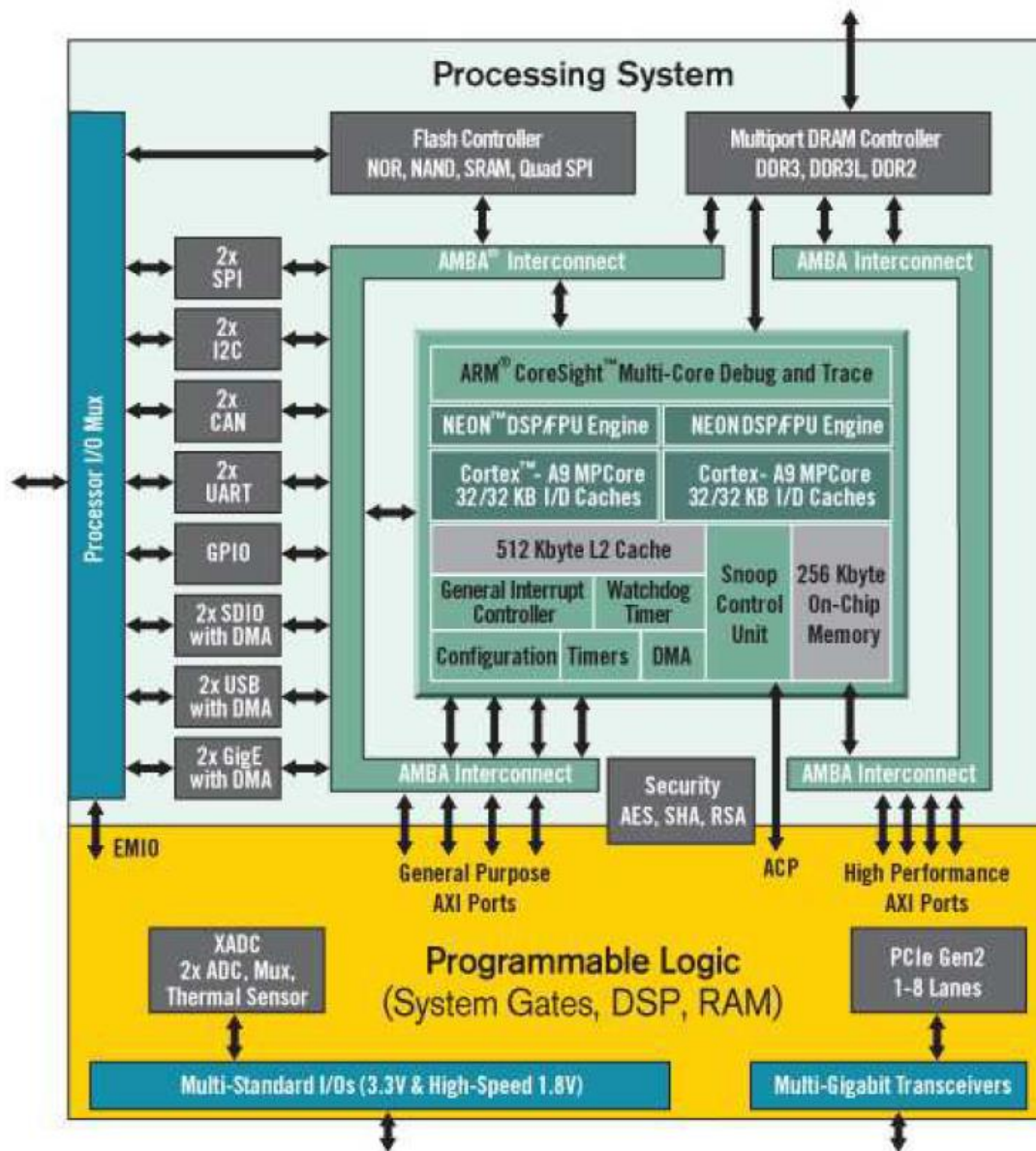
# XADC: Dual 12-Bit 1-MSPS ADCs

- Le bloc XADC permet soit de surveiller les alimentations et la température du FPGA via des capteurs sur la puce, soit d'acquérir jusqu'à 17 signaux analogiques externes via un multiplexeur. Il comporte deux convertisseurs analogique/numérique (ADC) 1 Msp/s 12 bits.
- On peut l'utiliser directement en VHDL ou avec le processeur ARM via le bus AXI.



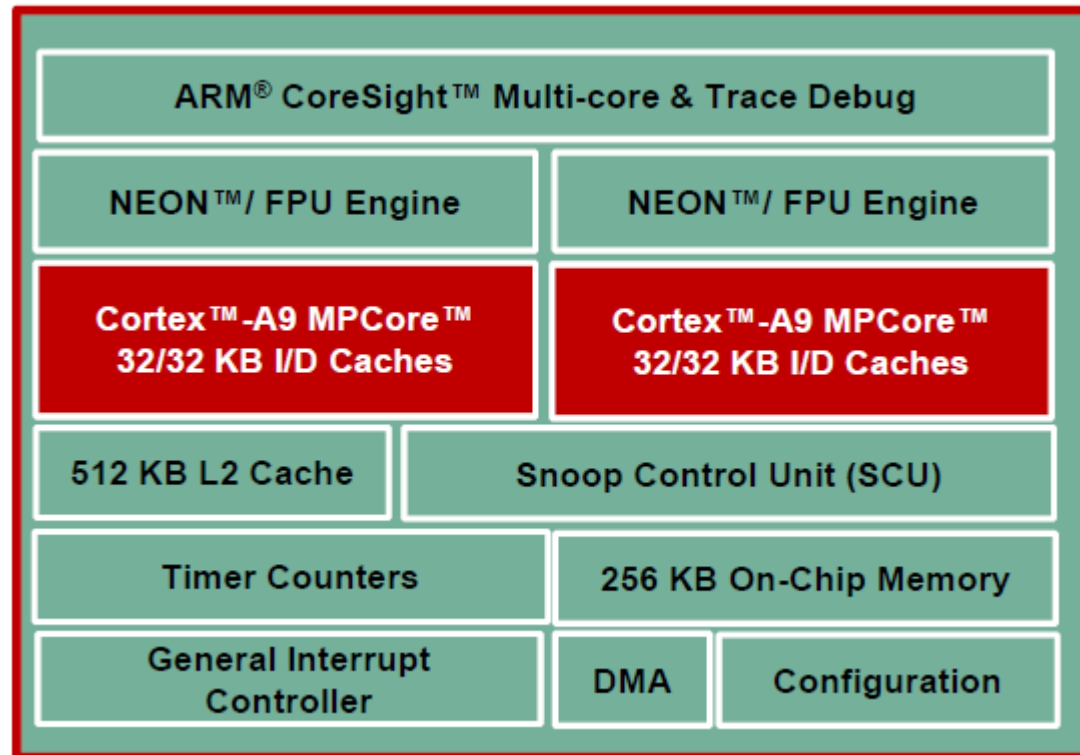
# Système microprocesseur (PS)

- La Zynq est une famille de FPGA qui contient en plus de la partie logique programmable (Programmable Logic PL) un système microprocesseur ARM double cœur complet avec ses contrôleurs mémoire et ses périphériques (Processing System PS).
- On dit que son architecture est centrée sur le PS car le processeur démarre en premier, puis charge le fichier de configuration de la PL.
- La logique programmable est alors utilisée pour réaliser des accélérateurs matériels connectés au PS. Ceci dit, on peut ne pas utiliser le PS et travailler uniquement avec la PL mais le PS sera nécessaire pour configurer la logique programmable sauf si on n'utilise que le JTAG.



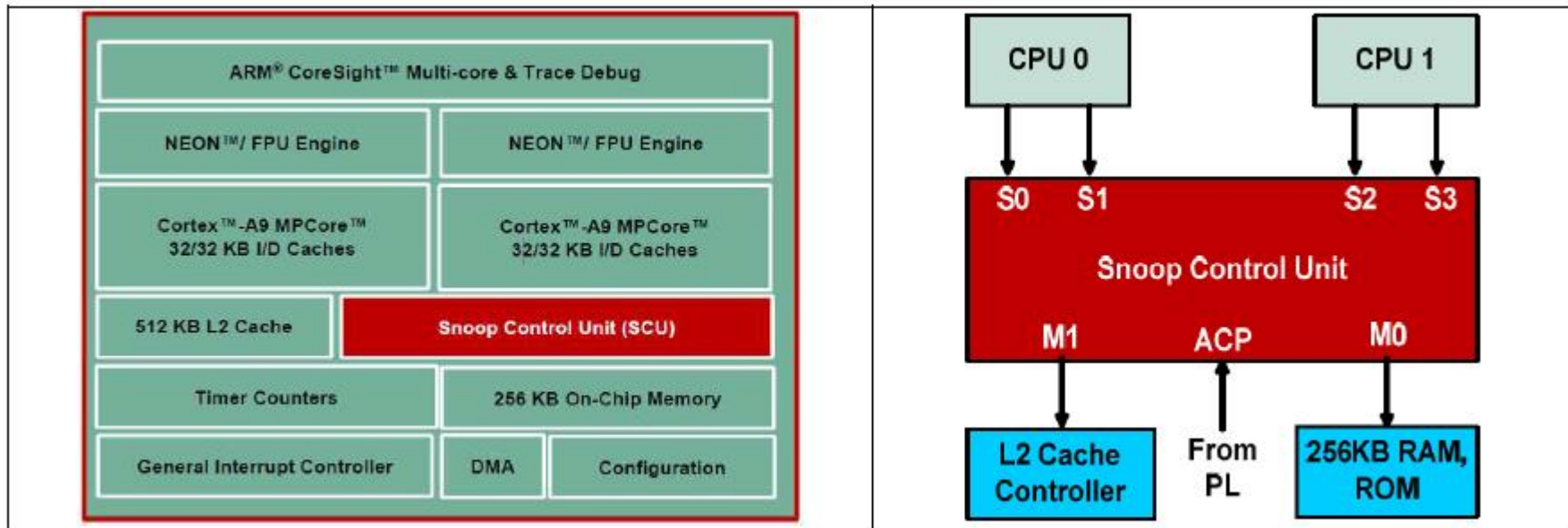
# Le système double cœur

- L'ARM Cortex-A9 est un processeur performant de type tablette fonctionnant jusqu'à 800 MHz. Il possède un cache L1 de 32 kilo octets sur les données et sur les adresses. Il y a deux coeurs A9 dans la Zynq.

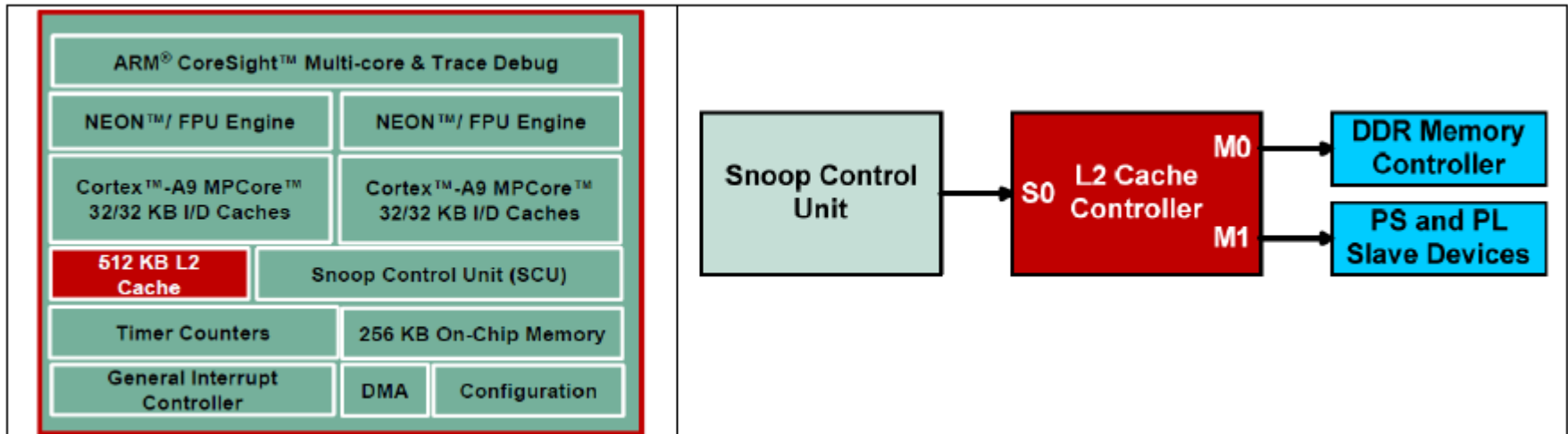




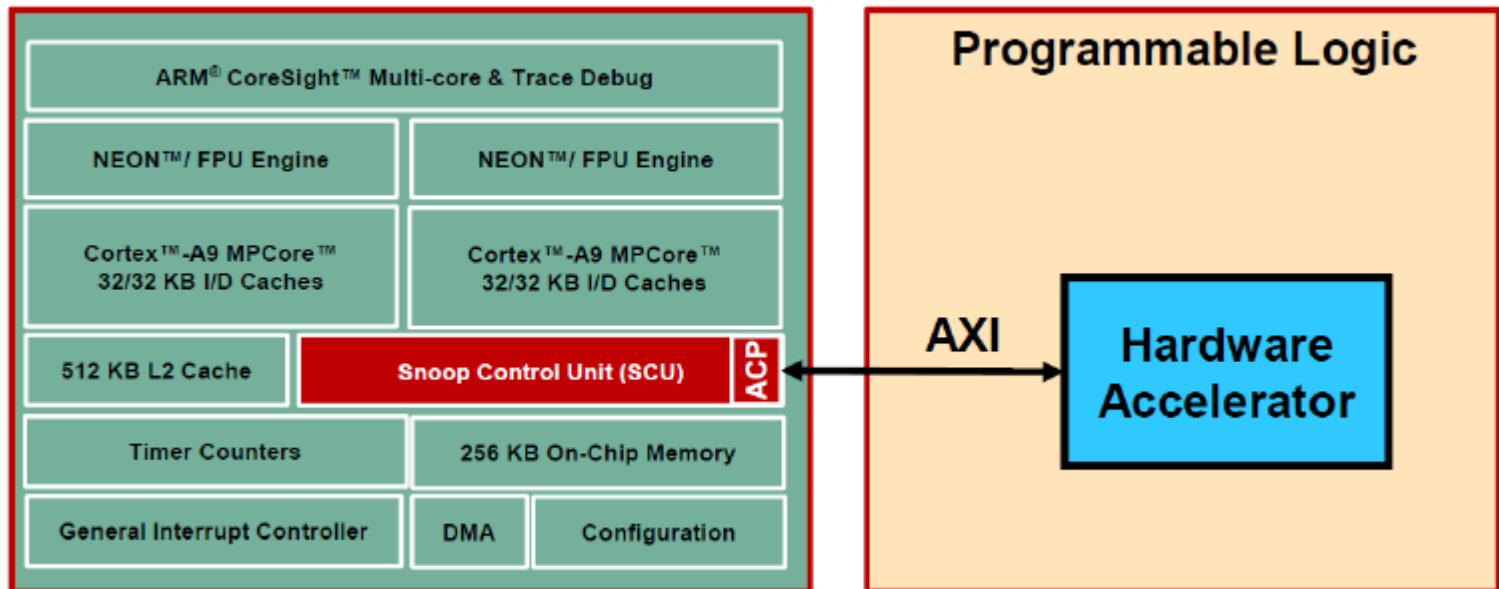
- La Snoop Control Unit (SCU) entreprend plusieurs tâches relatives à l'interfaçage entre les processeurs et les mémoires cache de niveau 1 et 2:
  - La cohérence des données entre les ressources de cache partagées.
  - Il arbitre entre les deux processeurs pour l'accès au cache L2. Il fournit un accès à la ROM/RAM interne.
  - Il fournit aussi un accès à l'ACP (Accelerator Coherence Port) pour la logique programmable.



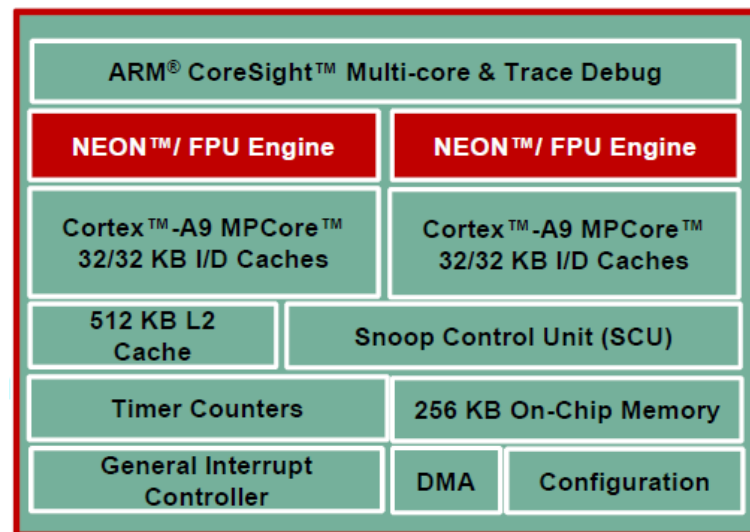
- Le cache L2 de 512 kilo octets est unifié et performant (8 voies associatives, write-back et write-through). Il cache la mémoire DDR ainsi que les périphériques PS et PL.



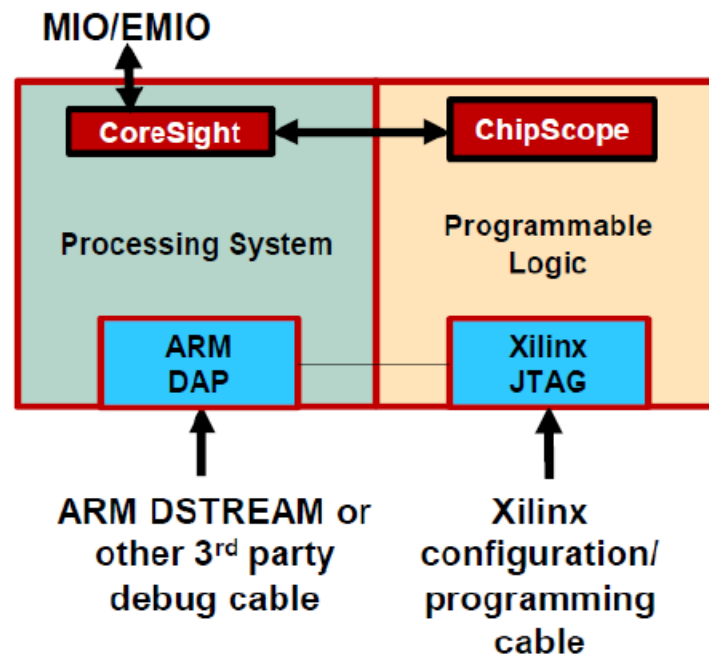
- L'ACP est un port AXI 64 bits reliant la PL avec la SCU. Il permet à l'accélérateur matériel dans la PL d'accéder au contenu des caches L1 et L2 tout en maintenant la cohérence des données.
- Il est utilisé pour partager les données entre PS et PL avec une faible latence.



- Le coprocesseur NEON est dédié aux instructions audio, vidéo et 3D.
  - Comme un DSP, il supporte des instructions SIMD (Single Instruction Multiple Data) sur 8, 16 32 et 64 bits.
  - C'est un accélérateur de calcul dédié au multimédia comme le jeu d'instructions SSE pour les processeurs Intel.
  - La FPU (Floating Point Unit) est un coprocesseur arithmétique dédié aux opérations sur les nombres réels.
  - Elle respecte la norme IEEE-754 en simple et double précision. Ses registres sont partagés avec l'unité NEON.

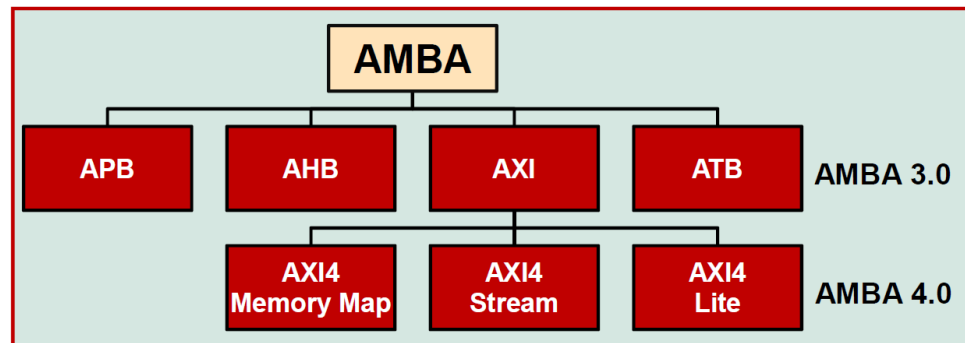


- Il existe plusieurs solutions pour debugger le programme dans la partie ARM ainsi que la logique programmable avec Vivado.
  - Nous utiliserons la solution mono câble JTAG avec le ARM Debug Access Port (DAP) placé en tête dans la chaine JTAG Xilinx.
  - Cela nous permettra aussi de télécharger le bitstream dans la PL.



- **Les interconnexions**

- L'Advanced Microcontroller Bus Architecture (AMBA) est utilisé comme bus interne dans les system-on-a-chip (SoC) basés sur les processeurs ARM depuis 1996.
- L'Advanced eXtensible Interface (AXI) est une extension de l'AMBA (version 4.0) pour Xilinx.



**AXI** - Advanced eXtensible Interface

**AHB** - Advanced High-performance Bus

**APB** - Advanced Peripheral Bus

**ATB** - Advanced Trace Bus (CoreSight on-chip debug and trace)