



SoC sur FPGA

Systèmes programmables sur FPGA

Tarik Graba
18 octobre 2016



Plan

Introduction

Softcore processor

Hardcore processor

Outils et méthodologie

SoC sur FPGA

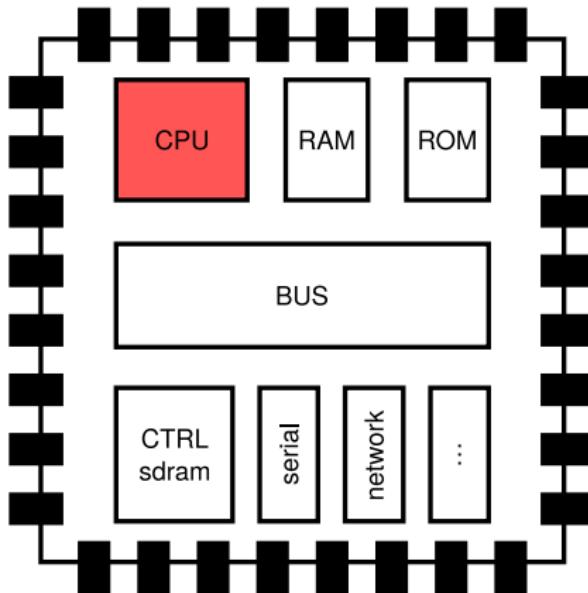
Pourquoi?

- Les FPGAs peuvent être utilisés pour implémenter matériellement des fonctions numériques:
 - des fonctions de traitement du signal,
 - des processeurs,
 - des accélérateurs
 - ...
- Utiliser pour construire des systèmes à base de processeurs

- Développer rapidement une application autour d'un **coeur** de processeur:
 - le développement logiciel est plus simple,
 - plus facile de trouver des développeurs,
 - réutiliser les bibliothèques et les OS existants.
- Ajouter des périphériques spécialisés grâce au FPGA:
 - pour des questions de performance,
 - pour simplifier les cartes électroniques (beaucoup d'IO, glue logic, ...)
- En deçà d'un certain volume, on n'a pas le choix.

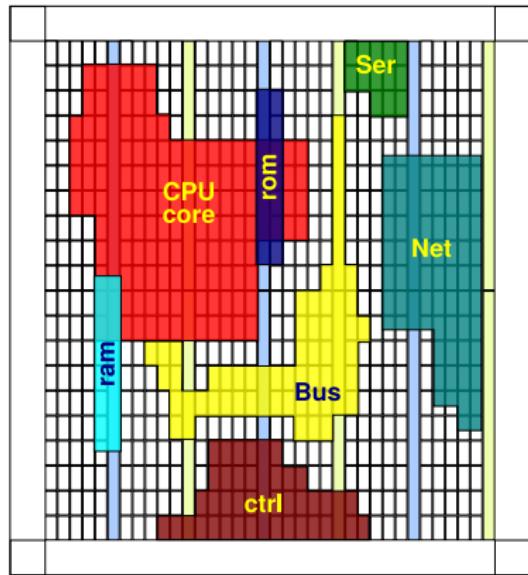
SoC sur FPGA

Avantages?



SoC sur FPGA

Avantages?



SoC sur FPGA

Avantages?

Un système à base de processeur

- un ou plusieurs processeurs,
- la mémoire (ram/rom) (jusqu'à une certaine taille),
- des interfaces de communication (standards, ad hoc, ...)
- les bus internes.

Qui n'est pas figé

- le FPGA reste programmable, on peut faire évoluer le système
- changé sa structure, sa fonctionnalité.



Plan

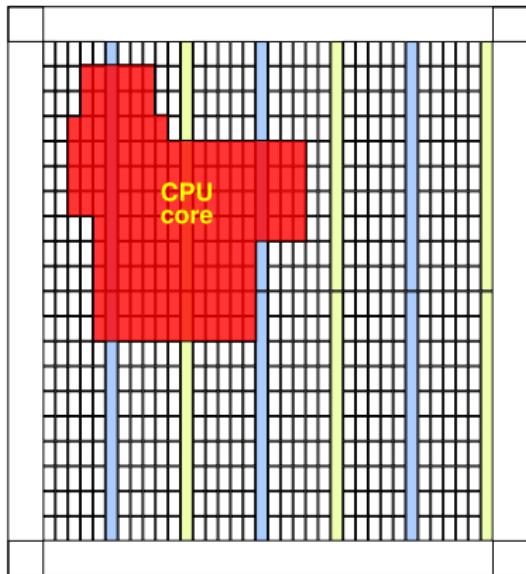
Introduction

Softcore processor

Hardcore processor

Outils et méthodologie

Softcore processor



Utiliser les ressources du FPGA pour le processeur et ses périphériques.



Softcore processor

Utiliser les ressources du FPGA pour le processeur

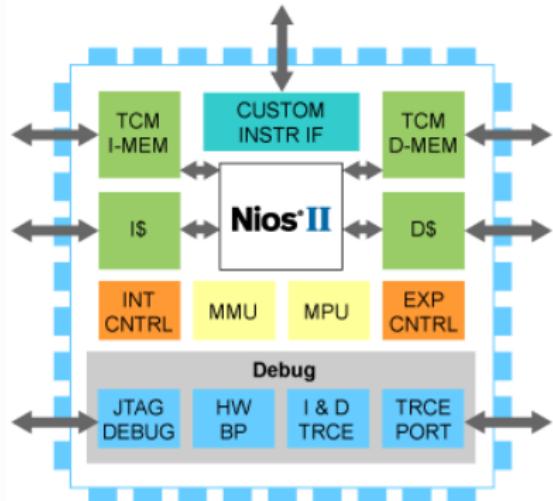
- + faisable sur n'importe quel FPGA
- + le cœur est modifiable
- processeur "spécial" FPGA
- performances limitées
- utilise des ressources dans le FPGA

Softcore processor

Exemple de plateformes

Altera Nios II

- exclusivement sur FPGA Altera
- processeur 32bits
- configurable (pipeline, instructions, cache, ...)
 - supporté par Linux
- les sources ne sont pas distribuées
- payant



Softcore processor

Exemple de plateformes

Xilinx Microblaze

- exclusivement sur FPGA Xilinx
- processeur 32bits
- configurable (pipeline, instructions, cache, ...)
 - supporté par Linux
- les sources ne sont pas distribuées
- payant

MicroBlaze

Softcore processor

Exemple de plateformes

Lattice Mico32

- conçu pour les FPGA lattice
- processeur 32bits
- les sources sont accessible et "libre"
- pas de support Linux

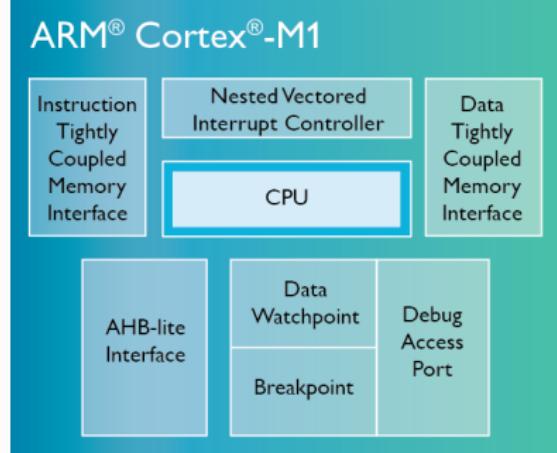


Softcore processor

Exemple de plateformes

ARM Cortex-M1

- Microcontrôleur 32bits (proche des Cortex-M0)
- configurable
- Supporte plusieurs familles de FPGAs
- pas de support Linux
- payant

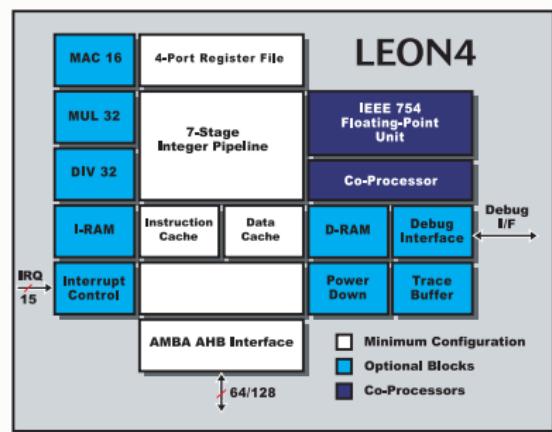


Softcore processor

Exemple de plateformes

Cobham/Gaisler Leon

- SPARC 32bits développé initialement pour l'ESA
 - supporte plusieurs familles de FPGAs
 - support Linux
 - opensource jusqu'à la version 3
 - existe en version durcie pour le spatial et l'avionique
 - existe aussi en ASIC



Softcore processor

Exemple de plateformes

OpenRisc, RiscV, ...

- des cœurs de processeurs opensources
- dans toutes les catégories (μ -contrôleur, high-end,...)
- l'environnement (logiciel/matériel) n'est pas toujours de qualité
- support de la communauté





Plan

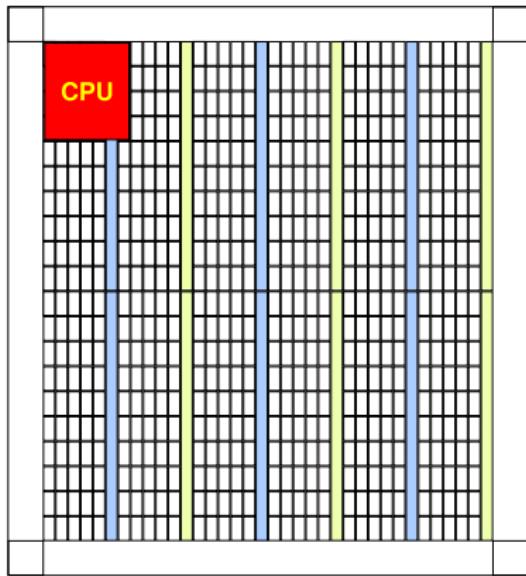
Introduction

Softcore processor

Hardcore processor

Outils et méthodologie

Hardcore processor



Avoir un vrai processeur dans la même puce que le FPGA.



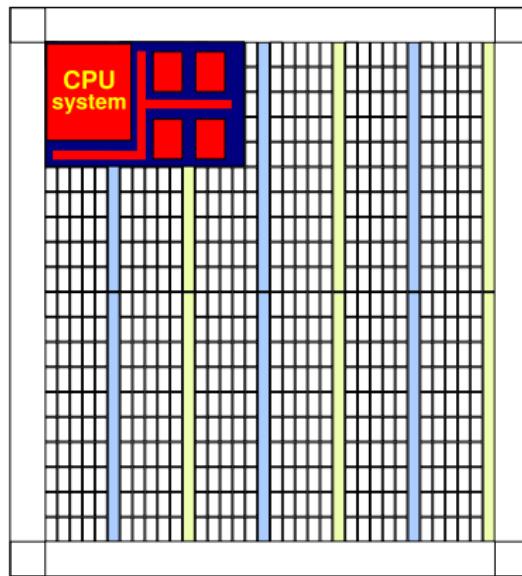
Hardcore processor

hard core: un bloc en dur dans la puce du FPGA

- restreint à certaines familles de FPGAs
- le cœur est figé
- + performances élevée
- + processeurs “standard”
- + la FPGA peut servir à autre chose

Hardcore processor

Hard System



Avoir un vrai SoC de base dans la même puce que le FPGA.

Un système sur puce de base dans la puce du FPGA

- + un soc de base facilite le développement logiciel:
 - + support des systèmes d'exploitation et bibliothèques
 - + mise en commun et réutilisation plus simple
- + on se concentre sur les ajouts
- apprentissage un peu plus complexe

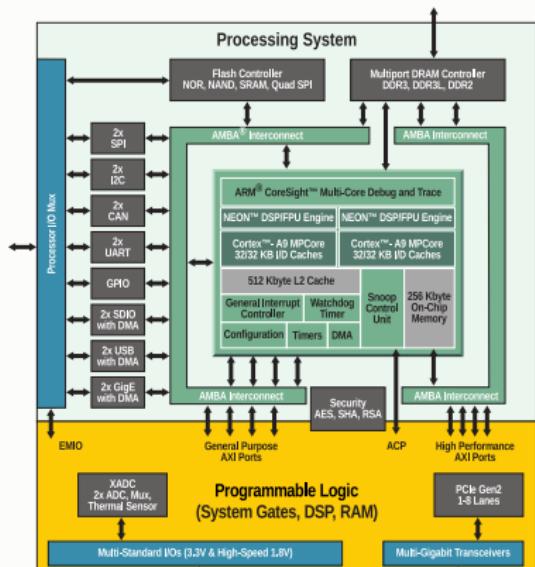
Exemple de plateformes

Xilinx Zynq SoC

Système sur puce complet:

- processeur dual core ARM Cortex-A9 (32bits)
- contrôleur de mémoire DDR, flash
- interfaces série, réseau, ...

Zone FPGA connectée au bus du processeur



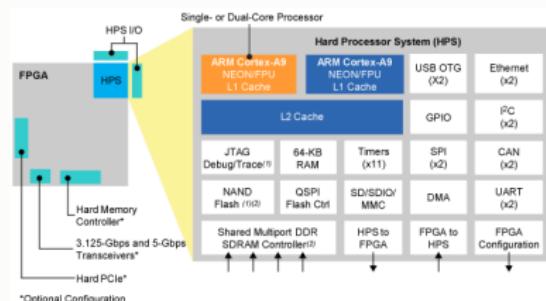
Exemple de plateformes

Altera CycloneV SoC

Système sur puce complet:

- processeur dual core ARM Cortex-A9 (32bits)
- contrôleur de mémoire DDR, flash
- interfaces série, réseau, ...

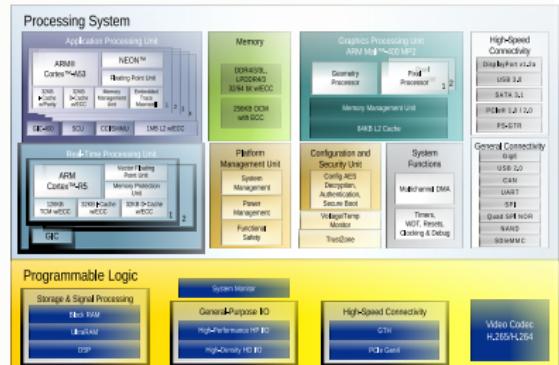
Zone FPGA connectée au bus du processeur



Exemple de plateformes

Xilinx Zynq Ultrascale+ MPSoC

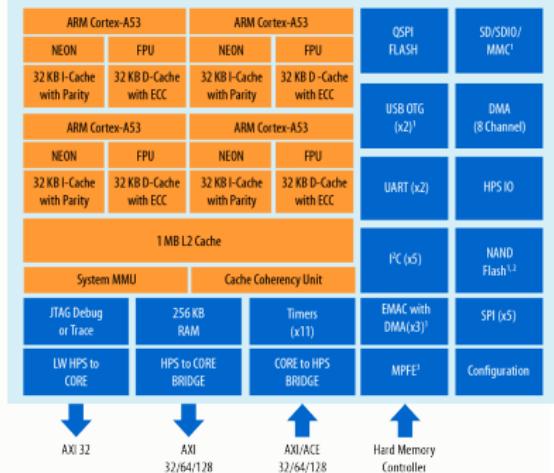
- processeur dual core ARM Cortex-A53 (64bits) et processeur temps réel dual core ARM Cortex-R5 (32bits)
- GPU Mali-400, processeur vidéo (optionnels)
- contrôleur de mémoire DDR, flash
- PCIe, SATA, DisplayPort, ...
- interfaces série, réseau, ...



Exemple de plateformes

Altera Stratix10-SoC

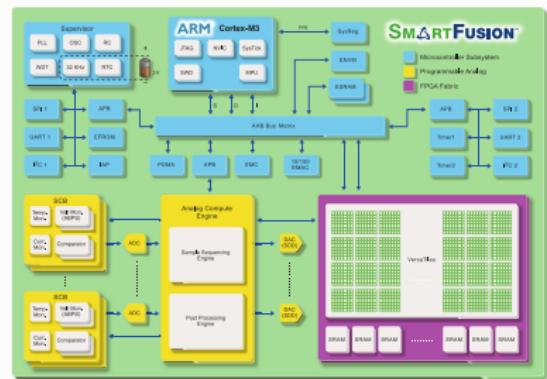
- processeur quad core ARM Cortex-A53 (64bis)
- contrôleur de mémoire DDR, flash
- interfaces série, réseau, ...



Exemple de plateformes

Microsemi Smartfusion/2

- Microcontrôleur 32bits ARM Cortex-M3
- interfaces série, réseau, ...
- ADC, DAC
- fonctions analogiques





Plan

Introduction

Softcore processor

Hardcore processor

Outils et méthodologie

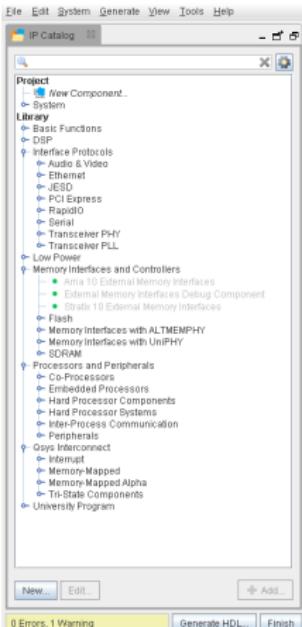
Outils et méthodologie

Développement orienté IPs

- Utilisation d'un standard de communication sur puce
 - AXI, AMBA, Avalon, Wishbone...
- Bibliothèque d'IPs fournies avec les outils
 - fonctions courantes (interfaces, contrôleurs mémoire...)
 - payantes ou gratuites

Outils et méthodologie

Développement orienté IPs



Altera Qsys

The screenshot shows the Xilinx Vivado IP Catalog window. The left pane displays a hierarchical list of cores and interfaces, including Vivado Repository, Alliance Partners, Automotive & Industrial, Base Infrastructure, BaseIP, Basic Elements, Communications & Networking, Debug & Verification, Digital Signal Processing, Embedded Processing, AUI Infrastructure, AUI Peripheral, Debug & Logic, Debug & Verification, Interprocessor Communication, Memory and Memory Controller, Microblaze, Microblaze MCS, UltraScale+ Feature and Design, Math Functions, Memories & Storage Elements, ECC, FIFOs, Memory Interface Generators, RAMs & ROMs, RAMs & ROMs & BRAM, Parallel Bus Configuration, SDAccel OpenCL Infrastructure, SDAccel OpenCL Programmable..., Standard Bus Interfaces, Video & Image Processing, and Video Connectivity. The right pane shows a detailed view of the selected 'ECC' core, with fields for Name (ECC), Version (2.0 (Rev. 11)), Description (ECC IP implements Hamming or HMAX encoder decoder function for varying data width), Status (Production), License (Included), Change Log (View Change Log), Vendor (Xilinx Inc), VLMN (xilinx.com/ip/ecc:2.0), and Repository (/home/elt/softs/opt/Xilinx/Vivado/2016.1/dlina/ip).

Xilinx Vivado

Outils et méthodologie

Développement orienté IPs

Pour l'utilisateur:

- Utiliser les IPs existantes
 - a-t-elle la bonne fonctionnalité?
 - comprendre comment elle fonctionne
- Développer ses propres IPs
 - maîtriser le protocole de bus
 - utiliser la méthodologie prévue pour s'intégrer à l'outil
 - utiliser des canevas et des assistants dans l'outil

Outils et méthodologie

Support logiciel

- Drivers pour interagir avec le matériel
 - avec un système d'exploitation (OS)
 - sans système d'exploitation (bare-metal)
- Bibliothèques logicielles
- systèmes d'exploitation
 - temps réel (rtOS)
 - Linux



Outils et méthodologie

Support logiciel

Comme le matériel peut changer:

- Des outils pour adapter/régénérer les drivers
 - BSP: Board Support Package
- Intégration dans des environnement connus des développeurs de logiciels
 - IDE: Integrated Development Environment
 - généralement basés sur Eclipse

