

# Co-design sur FPGA

## Conception d'un System on Chip (SoC)

Nicolas Gac – Imen Bahri  
IUT Cachan – Département GEII1  
Module complémentaire S3



# Demandez le programme !

- Séance 1 :
  - ✓ Panorama des techniques de conception des systèmes embarqués (très orienté FPGA du fabricant Altera)
  - ✓ Tutoriel de conception d'un SoC sur la DE2 (intégration d'un processeur NIOS) : « helloworld »
  - ✓ Présentation plus détaillée du sujet de « conception hardware/software d'un système de commande d'affichage VGA »
- Séance 2 et 3 :
  - ✓ Conception hardware du module d'affichage VGA (VHDL)
- Séance 4 et 5 :
  - ✓ Conception software de la commande du module VGA (C)
  - ✓ Test de fin maquette

# Planning !

- Mercredi 9 décembre :
  - 8h / 12h
  - 13h30 / 17h
- Jeudi 10 décembre :
  - ✓ 8h / 12h30
- Vendredi 11 décembre :
  - ✓ 8h / 11h30
  - ✓ 13h / 16h30
  - ✓ 16h30 / 17h30 Test de fin maquette

# Systèmes embarqués

système électronique et informatique autonome spécialisé dans une tâche bien précise aux ressources souvent limitées (encombrement spatial, consommation, contrainte temps réel...)

- Télécom/Multimédia : smartphones, tablettes, Box, Console, modem...
- Réseau/Informatique : routeur, stockage de données, impression, photocopieuse...
- Transport : Automobile, Aéronautique (avion), ferroviaire...
- Robotique
- Médical : instrumentation biomédical, scanners...
- Spatial : satellite, fusée, sonde spatiale...
- Militaire : missile...
- Mesures (métrologie) : télémètre...
- Banque : distributeur de billet, machine CB...
- Automate programmable industriel : Ascenseurs, gestion de batiments, logistique...

# Systèmes à base de $\mu\text{C}/\mu\text{P}$

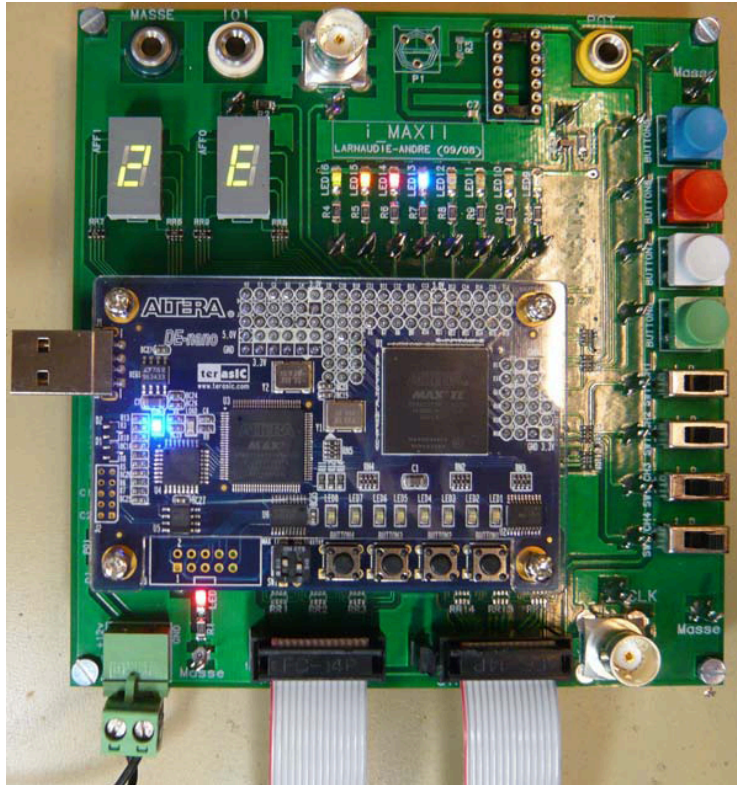
- Familles de  $\mu\text{C}/\mu\text{P}$  couramment utilisées :
  - ✓ ARM : Cortex-A, Cortex-R, Cortex-M
  - ✓ Freescale ColdFire, Freescale 68HC12
  - ✓ Intel 805 ...
- Processeurs spécialisés : DSP (Texas Instrument), graphic (GPU : NVIDIA/AMD)
- **Avantages  $\mu\text{C}/\mu\text{P}$**  : Flexible, facilité d'évolution, bibliothèques disponibles, rapidité de développements
- **Désavantage  $\mu\text{C}/\mu\text{P}$**  : performances limitées

# Systemes Matériels ASIC/FPGA

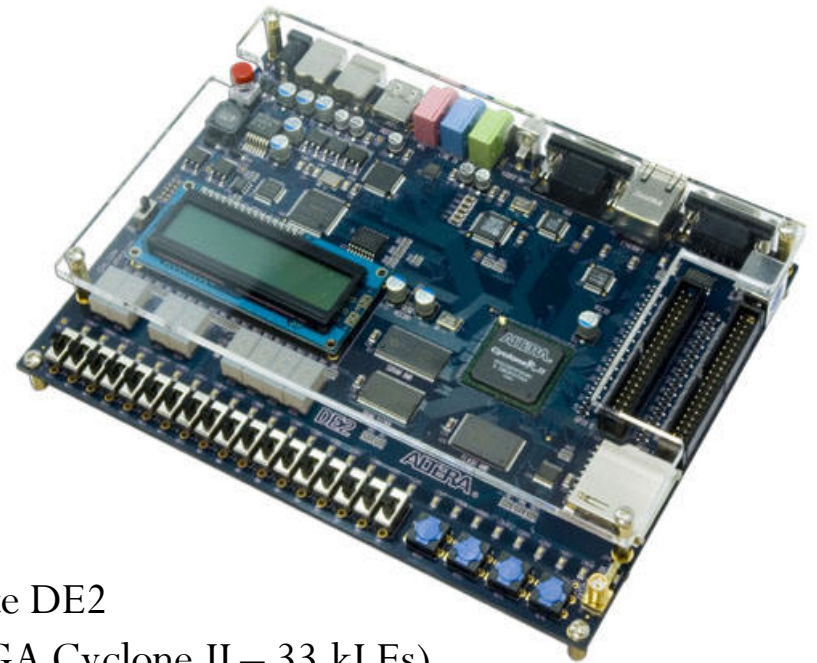
- **ASIC** : Application Specific Integrated Circuit
  - ✓ du sur mesure avec conception au niveau transistors/physique (« Full custom » ou « Standard Cell ») pour gros volume de production  $> 100\,000$  pièces
- **FPGA** : logique programmable
  - ✓ Configuration et placement/routage de cellule logique à partir d'une conception en Hardware Description Language (HDL): VHDL, Verilog, SystemC
  - ✓ Très utile pour le prototypage ou production à faible volume

# FPGA : « mer de portes »

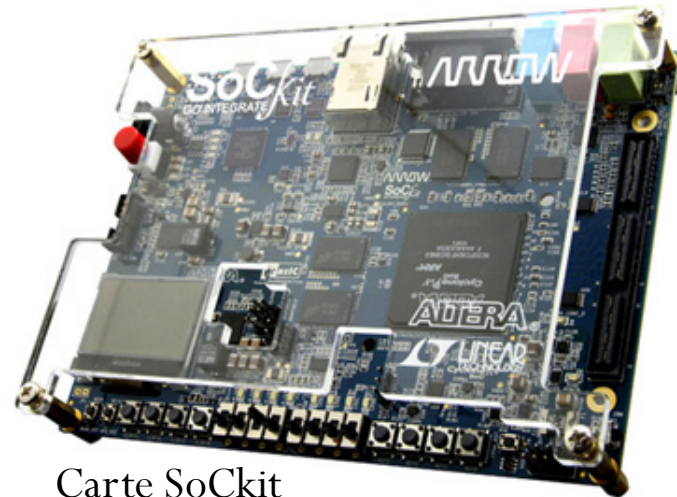
logique programmable ou  
plutôt reconfigurable



iMaxII (CPLD Max II - 2 kLEs)



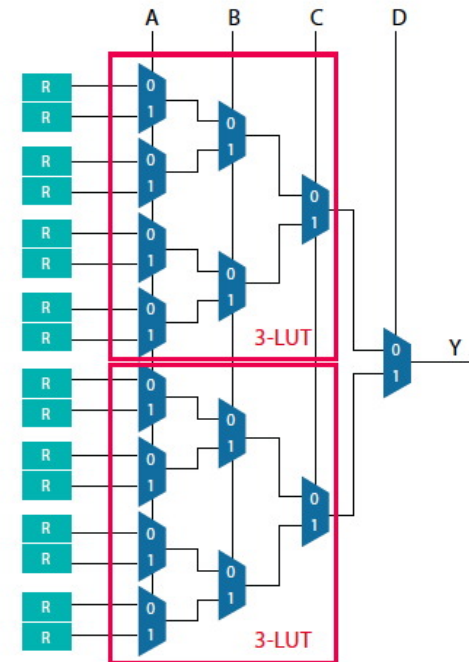
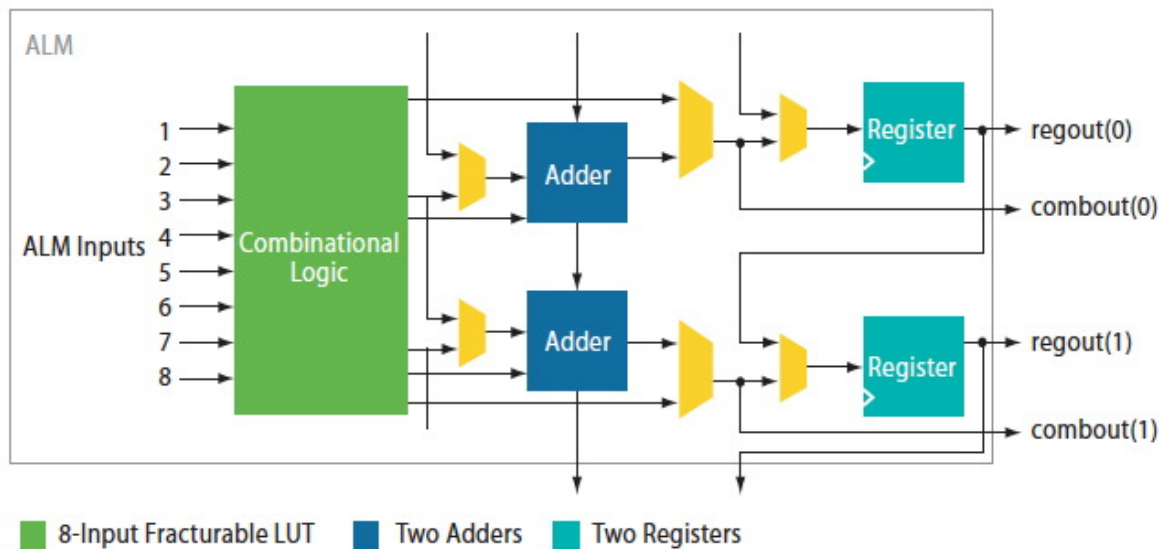
Carte DE2  
(FPGA Cyclone II – 33 kLEs)



Carte SoCkit  
(Cyclone V SoC – 110 kLEs + ARM A9 dual core)

# FPGA : (1) Logic Fabric

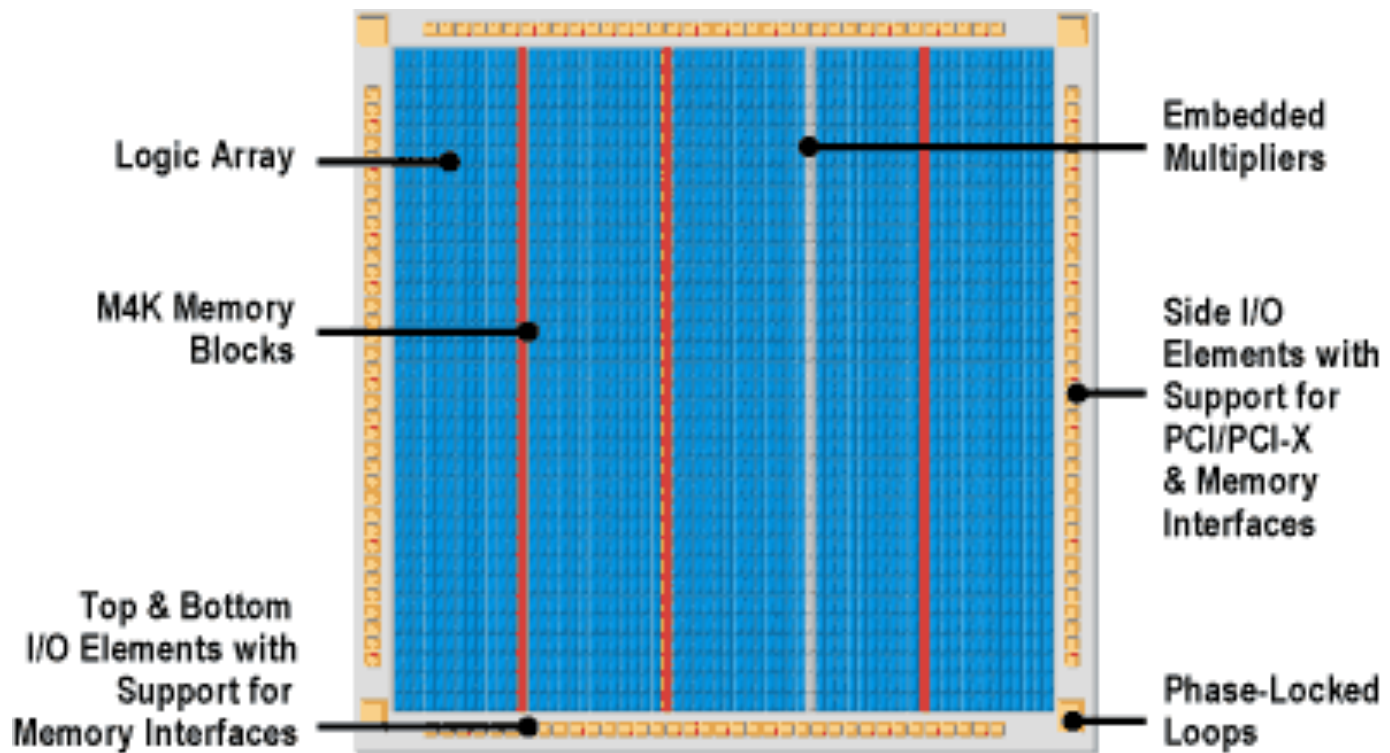
- Milliers de cellules logiques élémentaires configurables, les Logic Elements (LE) :
  - ✓ 4-input LUT (Look-Up Table) + Adder + Register
  - ✓ Configuration des LUTs par écriture dans les  $2^4$  SRAM bits





# FPGA : (2) Routing Architecture

- Interconnexion entre les différents LEs (regroupés en LAB de 10 LEs)



Cyclone II

# Systemes Matériels ASIC/FPGA

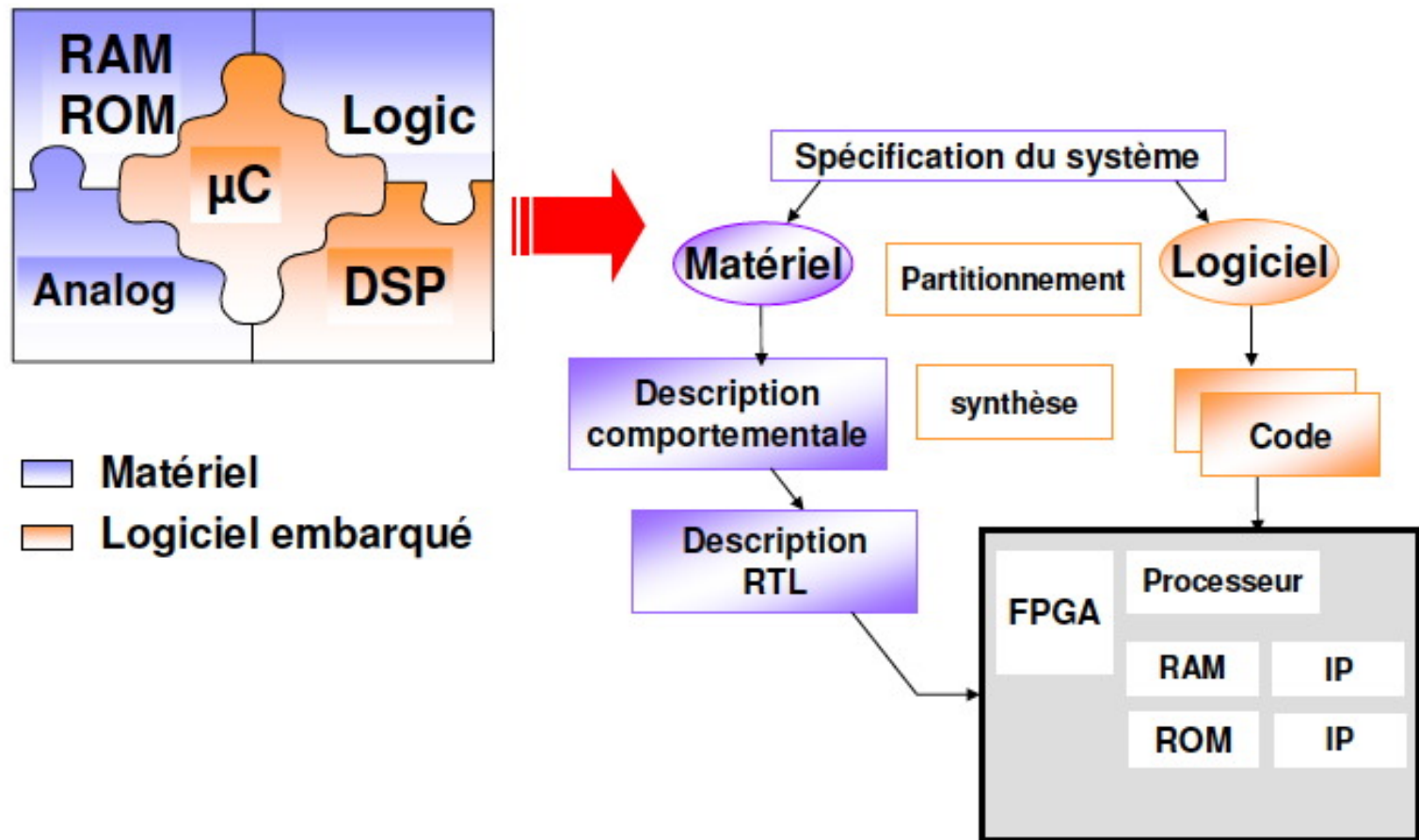
- **Avantages ASIC** : redoutablement précis et efficace (rapidité, faible consommation, déterminisme...)
- **Désavantages ASIC** : temps de développement très long (cycle 18 mois)
  
- **Avantages FPGA** : Bon compromis entre temps de développement, flexibilité et performance + possibilité d'utiliser des librairies (blocs IPs : Intellectual Property)
- **Désavantages FPGA** : nécessite des compétences hardware (VHDL...), flot logiciel pas aussi simple qu'un compilateur logiciel

# Nécessité d'une nouvelle méthode de conception face à la complexité croissante

- Les progrès technologiques en semi-conducteur permet d'intégrer de + en + de complexité dans une puce
  - ✓ 2004 : Cyclone II en 90 nm offre 33 kLEs (Logic Elements)
  - ✓ 2013 : Cyclone V en 28 nm offre 110 kLEs + ARM A9 dual core
- Accélération du « time to market » : évolution rapide des produits, concurrence accrue...

Difficultés accrues pour les concepteurs de suivre le rythme...

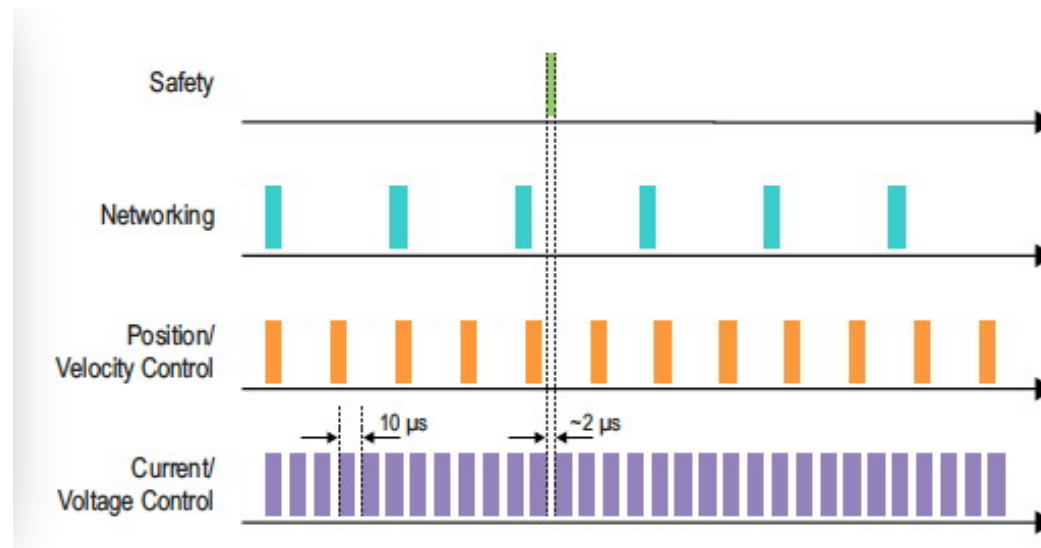
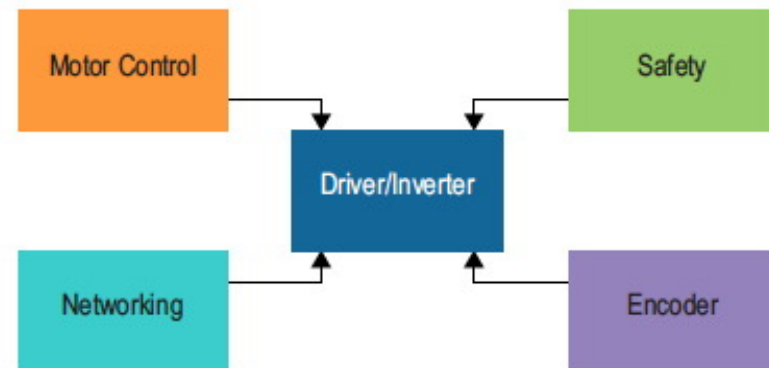
# Vers une approche système avec la conception d'une architecture hybride



# Les System On Chip (SoC)

- Systèmes sur Puces regroupant sur une même puce :
  - ✓ Microcontrôleur et ses périphériques
  - ✓ Mémoire interne
  - ✓ Contrôleur mémoire (mémoire externe SDRAM)
  - ✓ Bus
  - ✓ Des blocs logiques (ASIC ou FPGA) : « glue logic »
  - ✓ ...
- Deux types de SoC à base de FPGA :
  - ✓ 100 % FPGA :  $\mu$ C sur FPGA (**soft-core**) + glue logic sur FPGA
  - ✓ 50 % Hard + 50 % FPGA :  $\mu$ C en dur sur silicium (**hard-core**) + glue logic sur FPGA

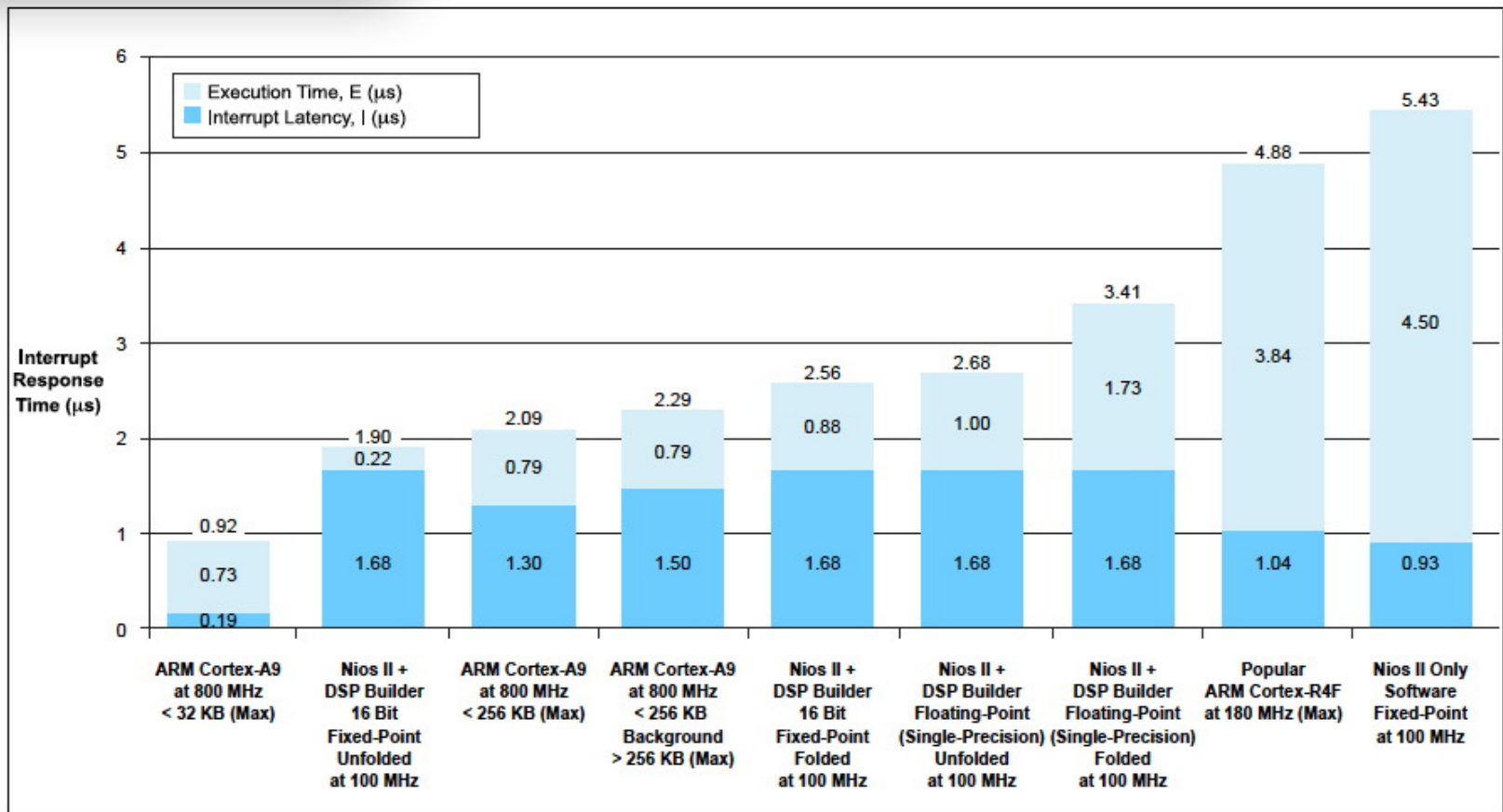
# Une Etude de cas Altera dans le domaine automobile (white paper « Real Time Challenge » on SoCs)



# Différentes solutions hardware/logicielles possibles

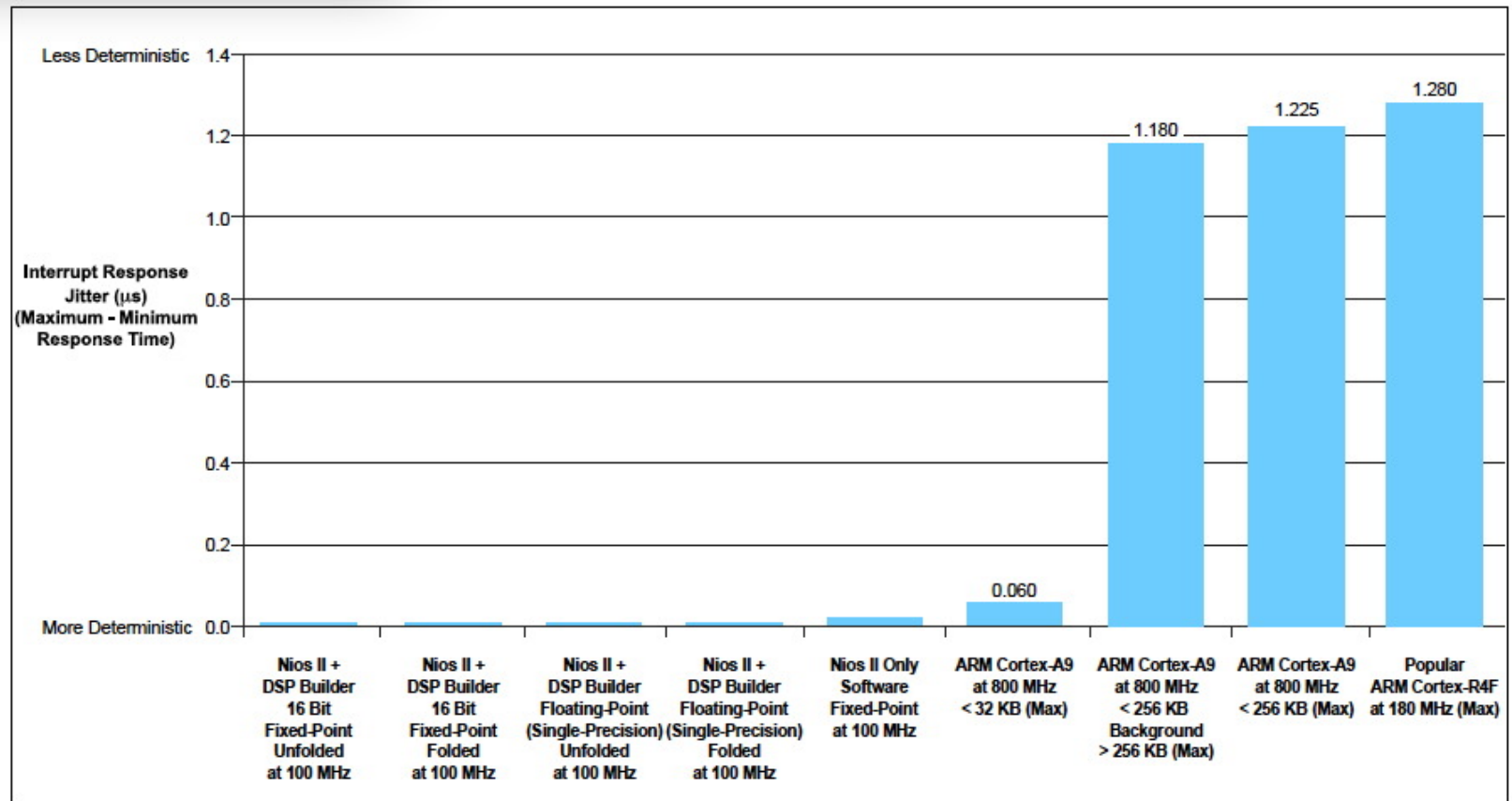
<b>Solution</b>	<b>Interrupt Latency</b>	<b>Execution Speed</b>	<b>Data Sets</b>	<b>Determinism</b>	<b>Design Method</b>
ARM Cortex-A9 Processor	Moderate	High	Very large	Moderate	C
Nios II Soft Processor	Low (vectored interrupt controller)	Moderate	Large	High	C
DSP Builder + Intellectual Property (IP)	Low	High to very high	Limited	Very high (no jitter)	MATLAB/ Simulink
Hardware-Based State Machines	Very low	Extremely high	Small	Very high (no jitter)	FPGA design, HDL tools

# Temps de réaction (Temps de prise en compte de l'interruption + Temps d'exécution de l'ISR)





# Fluctuation du temps de réaction (jitter)



# Conception d'un SoC sur la DE2 :

## « commande de l'affichage sur écran VGA »

- FPGA Cyclone II
  - ✓ Glue Logic : Module VGA qui va générer les signaux vers le port VGA relié à l'écran (pixels + synchronisation ligne/image)
  - ✓ Processeur : soft-core NIOS qui va commander via une liaison série relié au PC le module VGA pour changer les couleurs sur l'écran (programmation en C)
- Flot logiciel
  - I. Quartus :
    - a) Génération du NIOS
    - b) Conception VHDL du module VGA
    - c) Intégration NIOS +VGA dans un schématic top du circuit
    - d) *Synthèse logique* puis configuration du FPGA
  - II. Eclipse :
    - a) Génération du code (drivers « BSP ») et écriture de votre programme (main.c)
    - b) *Compilation logicielle*
    - c) Exécution du programme sur le FPGA configuré comme un SoC

# Tutoriel « Helloword »

- But : créer un premier SoC avec la configuration de votre premier processeur NIOS. Ce NIOS sera relié au switch et au LEDs, aucun bloc hardware (VHDL ou IP) ne sera créer.
- Documents disponibles dans le cours Dokeos
  - S3 Module complémentaire Co-design (C/VHDL)
  - UTCACHANGEII1S3MCCOD – nicolas.gac
  - Tutorial Altera (parties utiles surlignées en jaune fluo)
  - Documents Altera pour l'utilisation de l'UART du NIOS
  - DE2.qsf
  - Sujets 1 et 2 du projet VGA

# Préparation du terrain

## REPERTOIRE DE TRAVAIL

- Créer un répertoire personnel en local dans C:/Travail  
Ex : C/Travail/toto\_and\_titi
- Puis un répertoire pour le tutoriel  
C/Travail/toto\_and\_titi/Myfirst\_codesign
- Aller rechercher dans le cours Dokeos « S3 Module complémentaire Co-design (C/VHDL) », le fichier de configuration des pins « DE2.qsf », placer le dans le répertoire Myfirst\_codesign

## QUARTUS

- Créer sous Quartus dans le répertoire « Myfirst\_codesign » un projet  
Nom du projet : Myfirt\_codesign  
FPGA : Cyclone 2 EP2C35F67C6
- Créer un fichier schématique (new -> Block Diagram/Schematic File), enregistrez le (save as) comme « top.bdf » et définissez comme le top level entity (onglet Files dans le project navigator, clic droit sur le top.bdf, « Set as top level entity »)