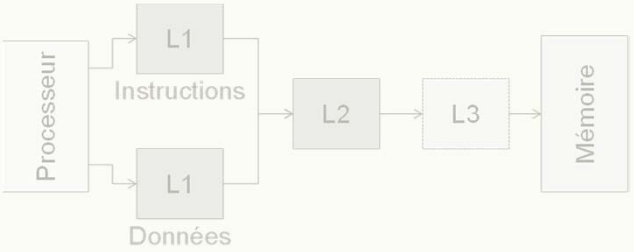


ARO2

Architectures de processeur et performances

Basé sur le cours du prof. E. Sanchez
et le cours ASP du prof. M.Starkier

RMQ/CPN

Cours ARO2

RISC / CISC

Conception des processeurs CISC

- **Pour augmenter les performances des processeurs,**
  => augmenter la complexité des instructions
  - **réduire le nombre d'instruction**
  - **réduire la taille du code**

Architecture CISC (1970 à ...)

- Augmenter le nombre et la complexité des instructions disponibles
- Objectif : réduire le nombre d'instruction utilisées pour un programme et la taille du code

  CISC = Complex Instruction Set Computer
  - Ordinateur à jeu d'instructions complexes

- Microcode ou microprogramme d'instruction
- Instructions de taille variable (code et nb de cycles d'exécution)
- Multiples modes d'adressage supportés (pour toutes les instructions)
- Nombre d'instructions élevé

La révolution du RISC

- Constatation : 80% du code utilise 20% des instructions (CISC)
- => imaginer un jeu d'instruction plus limité, mais plus efficace

- Nouvelle architecture => RISC (1980 - 1985)
- Acorn RISC Machine (UK – Cambridge)) = ARM = Advanced RISK Machine

Architecture RISC

- RISC = Reduced Instruction Set Computer
  - Ordinateur à jeu d'instruction réduit
- Pas de micro-programme
- Instructions de taille fixe
- Limitation des modes d'adressage
  - Registres pour opérations arithmétiques et logiques
  - Adressage indirect pour accès mémoire
- Nombre d'instructions plus faible

Approches CISC / RiSC

- Calcul de la performance d'un processeur par mesure du temps d'exécution d'un programme

Te = Nbr_instr. x Nbr_cycle/instr x Temps_cycle

- Pour diminuer Te (temps d'exécution)
  - Approche CISC => Diminuer le nombre d'instructions pour coder un programme
  - Approche RISC => Diminuer le temps de cycle et le nombre de cycle par instruction

Comparaison CISC / RISC

- **Avantages CISC**
  - Moins d'instructions pour une fonction
  - Possibilité de micro-programmation, donc de correction du jeu d'instructions
  - Permet d'utiliser des instructions très complexes et très rapides

- **Avantages RISC**
  - Architecture globalement moins complexe
  - Consommation d'énergie moindre
  - Interruptions plus rapides
  - Compilateur plus simple et efficace

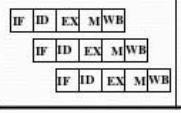
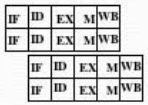
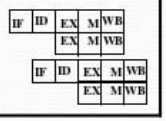
Comparaison CISC / RISC

- Inconvénient CISC
  - Surface importante de silicium
  - Consommation importante
  - Difficulté à réaliser des compilateurs gérant des instructions complexes
  - Temps de développement et de validation importants

- Inconvénient RISC
  - Taille du programme plus importante (20 % à 50%)
  - Instruction à taille fixe pas toujours optimisée
  - Traitements avec accès multiples à la mémoire pénalisés (chaînes,...)

CISC vs RISC

CISC vs RISC vs SS vs VLIW

	CISC	RISC	Superscalar	VLIW
Instruction size	variable size	fixed size	fixed size	fixed size (but large)
Instruction format	variable format	fixed format	fixed format	fixed format
Registers	few, some special	many GP	GP and rename (RUU)	many, many GP
Memory reference	embedded in many instr's	load/store	load/store	load/store
Key Issues	decode complexity	data forwarding, hazards	hardware dependency resolution	code scheduling, (compiler)
Instruction flow				

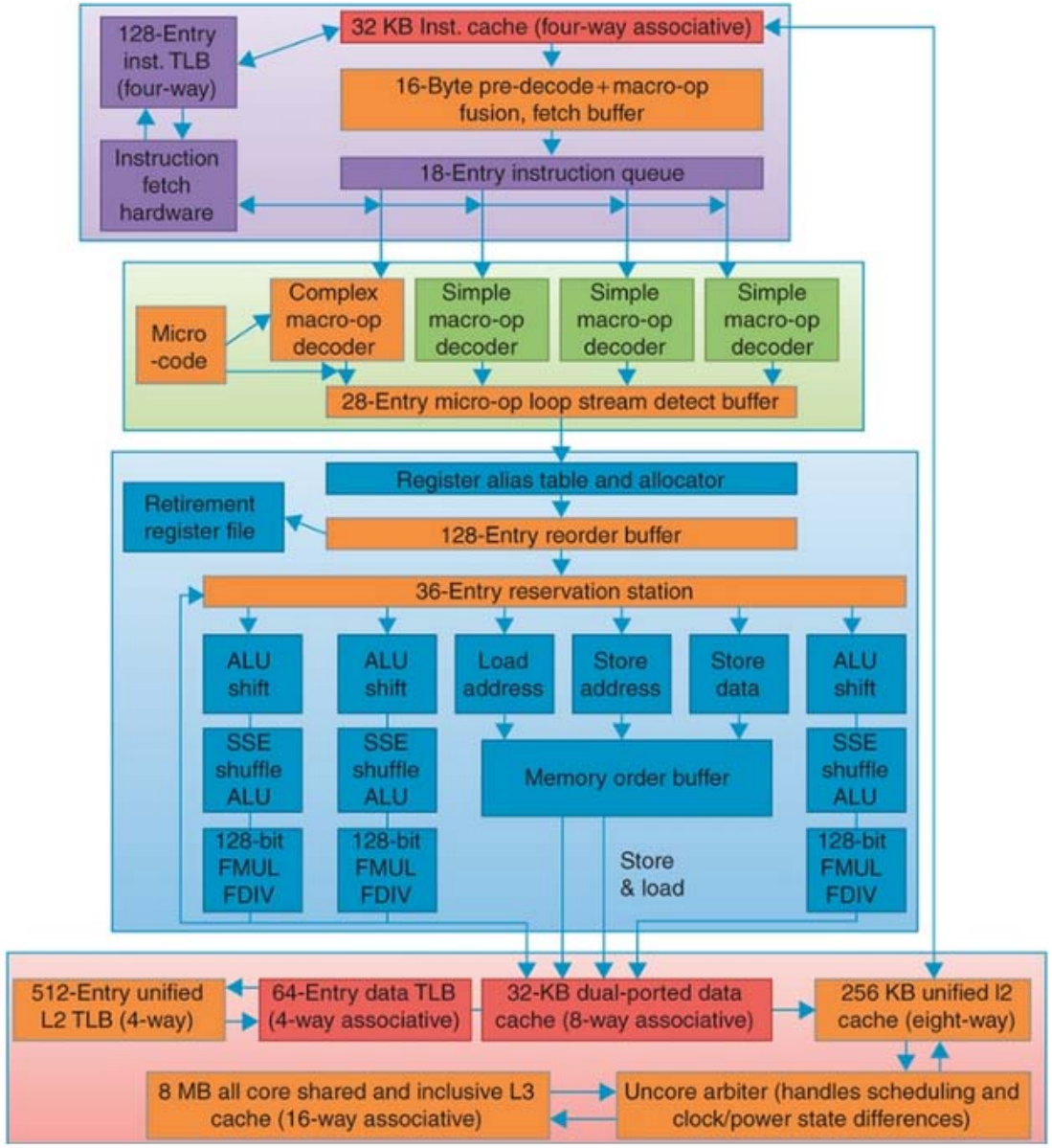
RISC	ARM7	ARM9
CISC	Pentium	SHARC (DSP)
	von Neumann	Harvard

Microprocessors

RISC v.s. CISC

Parameter	RISC	CISC
Instruction types	Simple	Complex
Number of instructions	Reduced (30-40)	Extended (100-200)
Duration of an instruction	One cycle	More cycles (4-130)
Instruction format	Fixed	Variable
Instruction execution	In parallel (pipeline)	Sequential
Addressing modes	Simple	Complex
Instructions accessing the memory	Two: Load and Store	Almost all from the set
Register set	multiple	unique
Complexity	In compiler	In CPU (micro-program)

Intel Core I7



A Quantitative Approach,
[John L. Hennessy, David A. Patterson]

Actual RISC families

- ARM
  - Cortex-A, Cortex-M, Cortex-R
- MIPS
  - Classic, Aptiv, Warrior
- SPARC
- POWER PC

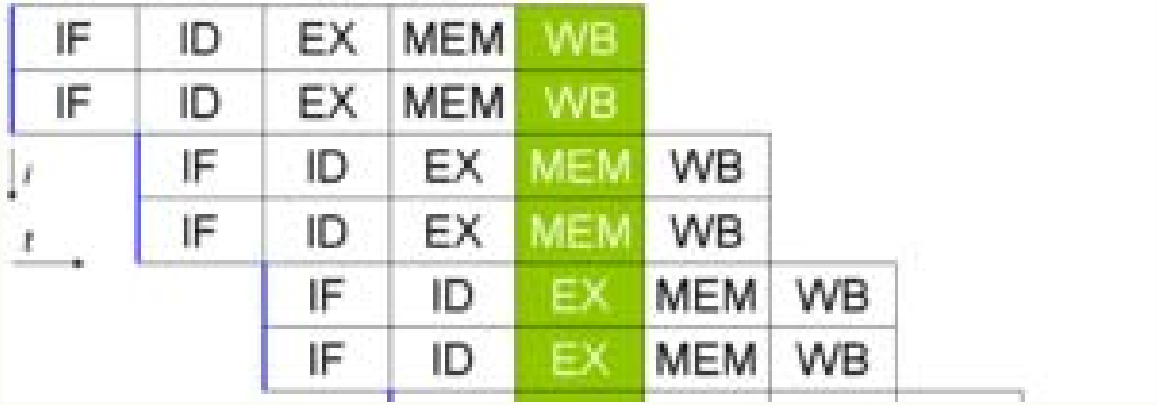
Cours ARO2

AMÉLIORATION DES PERFORMANCES

Processeur superscalaire

- 1885 - Exécution de plusieurs instructions simultanément
- Plusieurs pipelines de traitement en parallèle
- Nb instructions par cycle > 1
- Gestion complexe du parallélisme par hardware
- Depuis 10 ans, les processeurs sont superscalaires

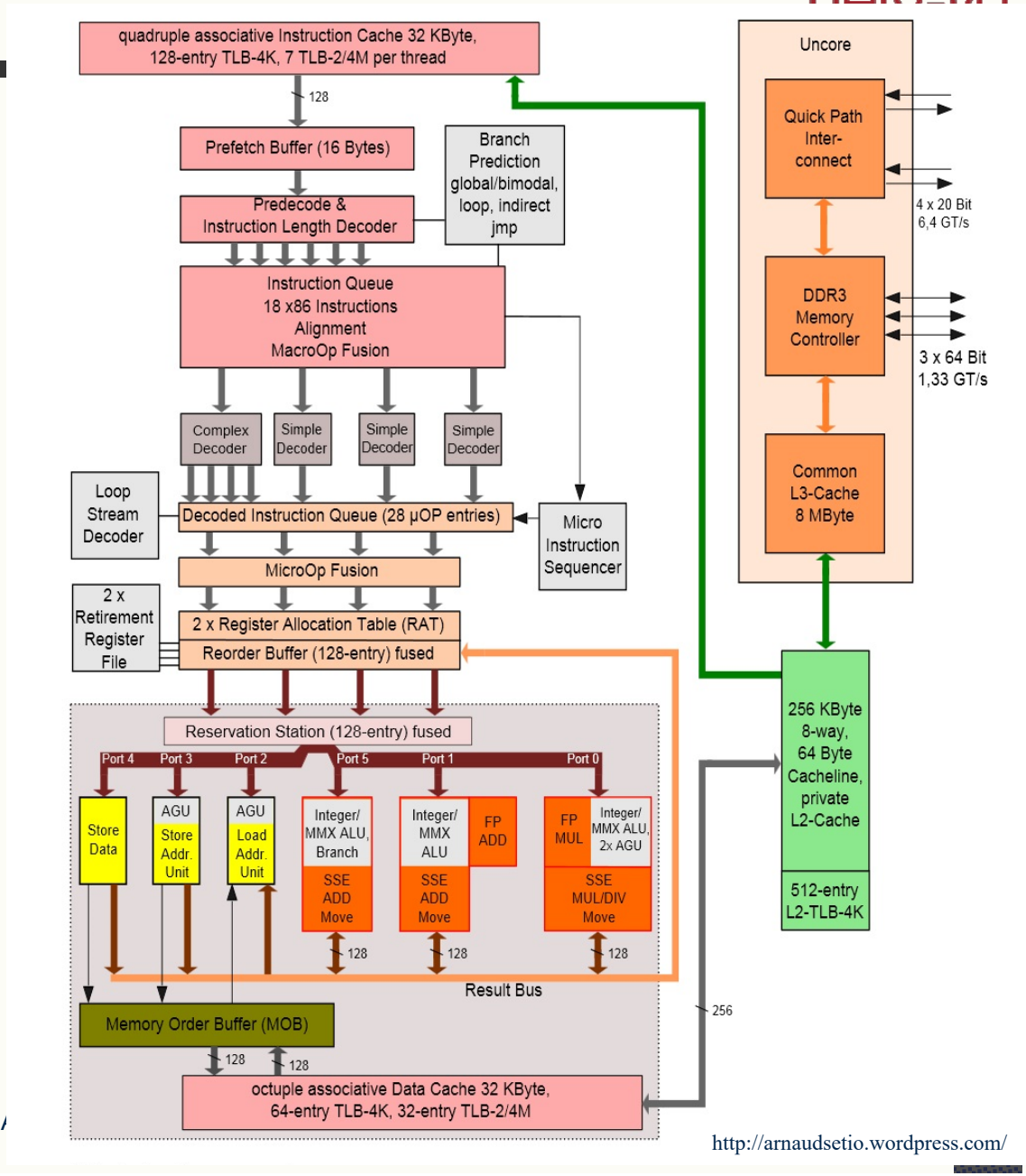
ADD R3, R4, R5
SUB R2, R1, #1
.....



Wikipédia

Processeur superscalaire

- Intel I960
- AMD 29000
- **Intel Nehalem**

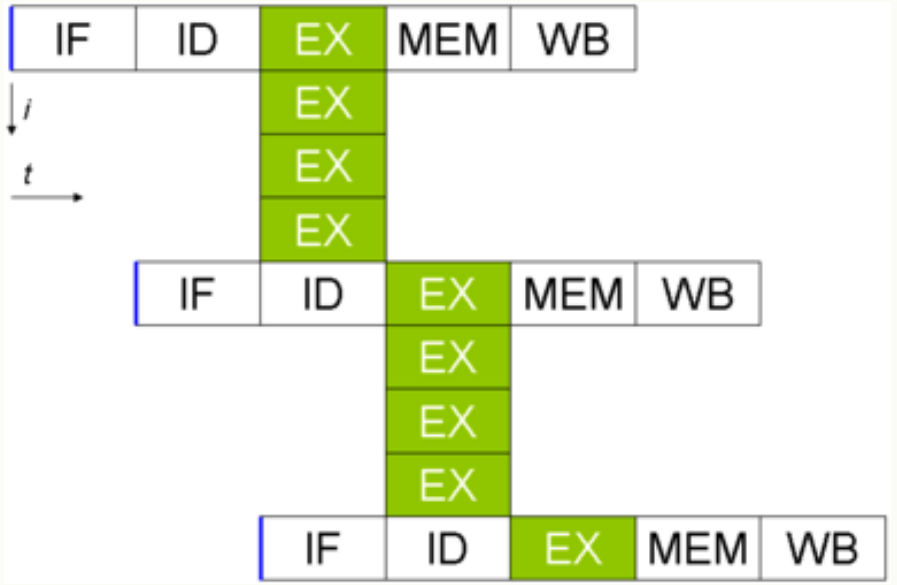


http://arnaudsetio.wordpress.com/

VLIW Very Long Instruction Word

- 1988 - Parallélisme implicite dans le jeu d'instruction
- Code instructions multiples (> 128 bits)
- Parallélisme géré par le compilateur
- Philips TriMedia /AD Sharc DSP / TI 6000 DSP

ADD R3, R4, R5 / SUB R6,R7,R1/.....

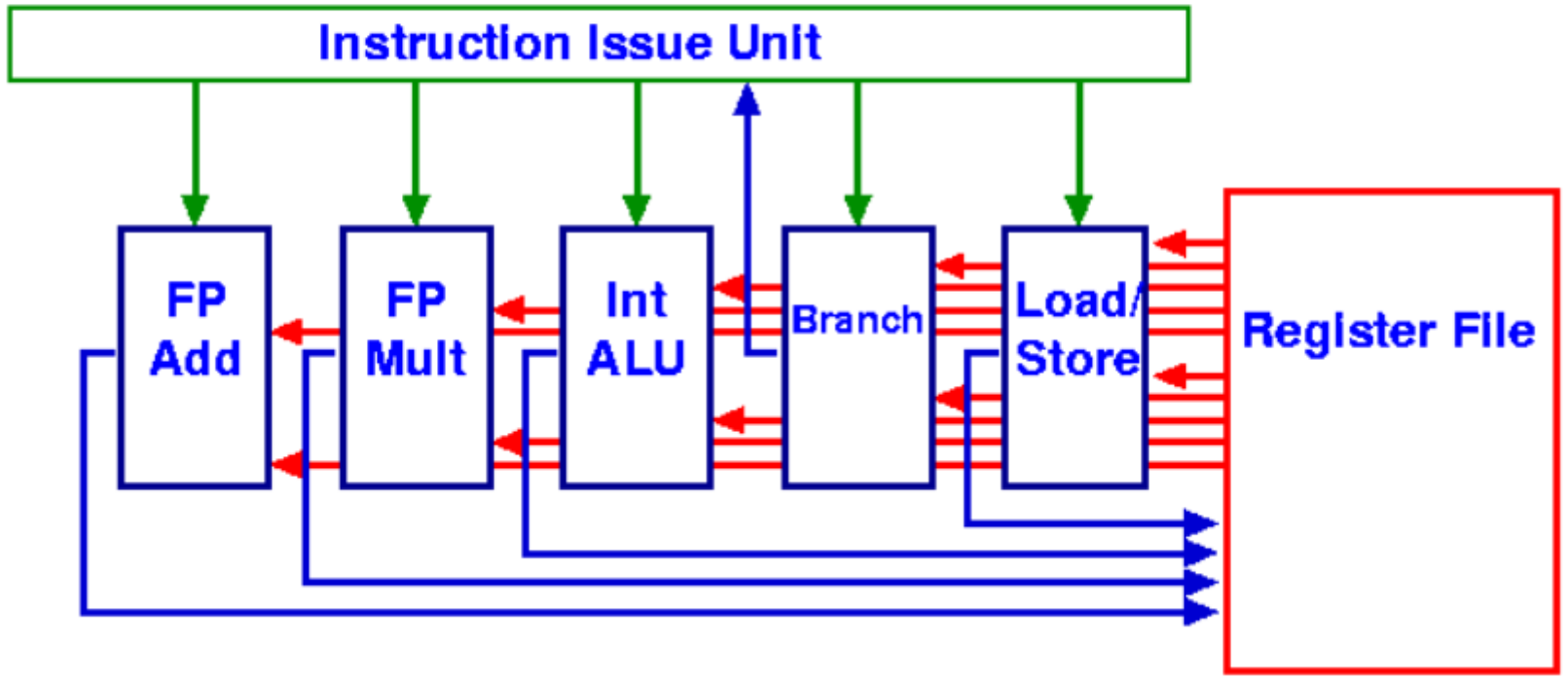


Wikipédia

VLIW architecture

Instruction Format

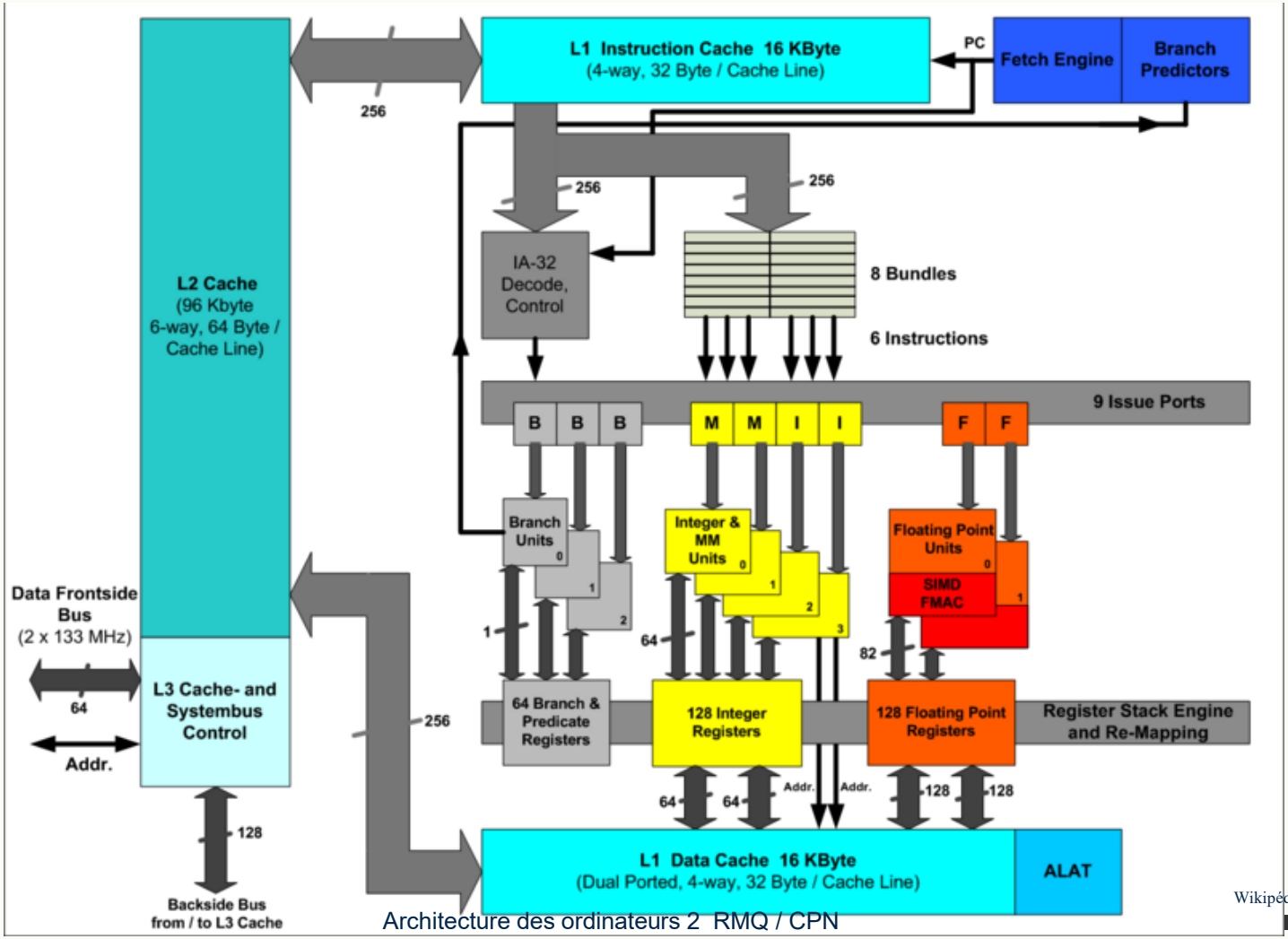
FP Add	FP Mult	Int ALU	Branch	Load/Store



https://www.cs.auckland.ac.nz

VLIW Very Long Instruction Word

Itanium / HP-Intel alliance / EPIC Explicitly Parallel Instruction Computing



Wikipédia

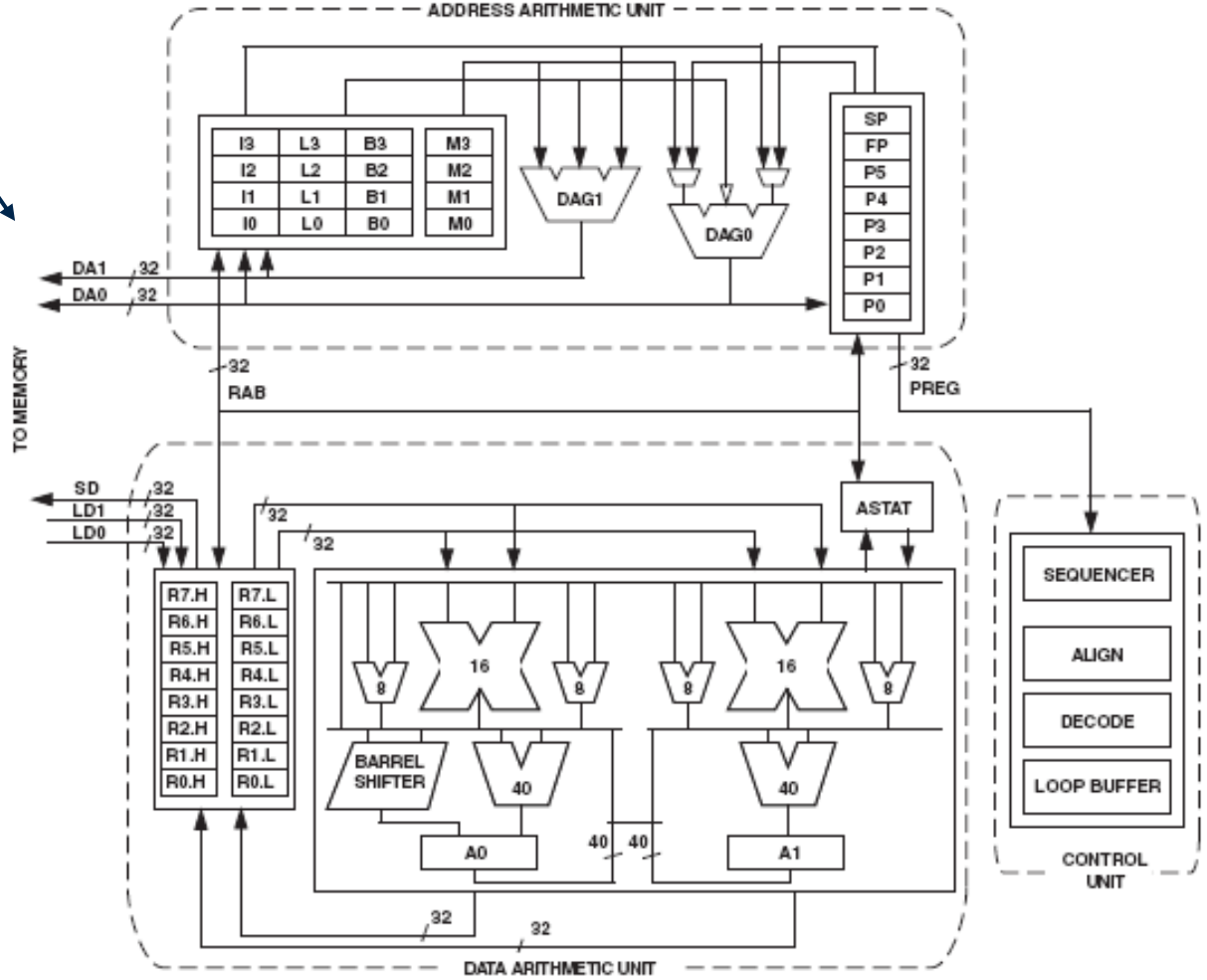
DSP

- Processeur optimisé pour le traitement de signal
- Architecture Harvard
- Une (ou plus) multiplication-accumulation par cycle
- En général, format de données : entiers
- Bus d'I/O à haut débits
  - Interfaces pour convertisseurs AD/DA
  - Ou convertisseurs incorporés
- Traitement vectoriel
- Extension DSP de processeurs standards :
  - Par exemple extensions DSP ARM

DSP

Analog Devices

Sharc / Blackfin



Traitement des threads

- Multithreading
  - exécutions des threads non simultanées
  - «illusion de parallélisme »
- Time/slice or temporal multithreading
  = superthreading
  - exécutions des threads (d'un même process) simultanément – avec restriction
- Silmutaneous multithreading (SMT)
  = hyperthreading
  - exécution simultanée des threads

Multi-core

- Plusieurs CPU en parallèle dans un même processeur
- Traitement d'un ou plusieurs threads par CPU
- 2/4/8 coeurs (> 64 ?)

- Multi-core homogène ou hétérogène

Multi-processeurs / multi-core

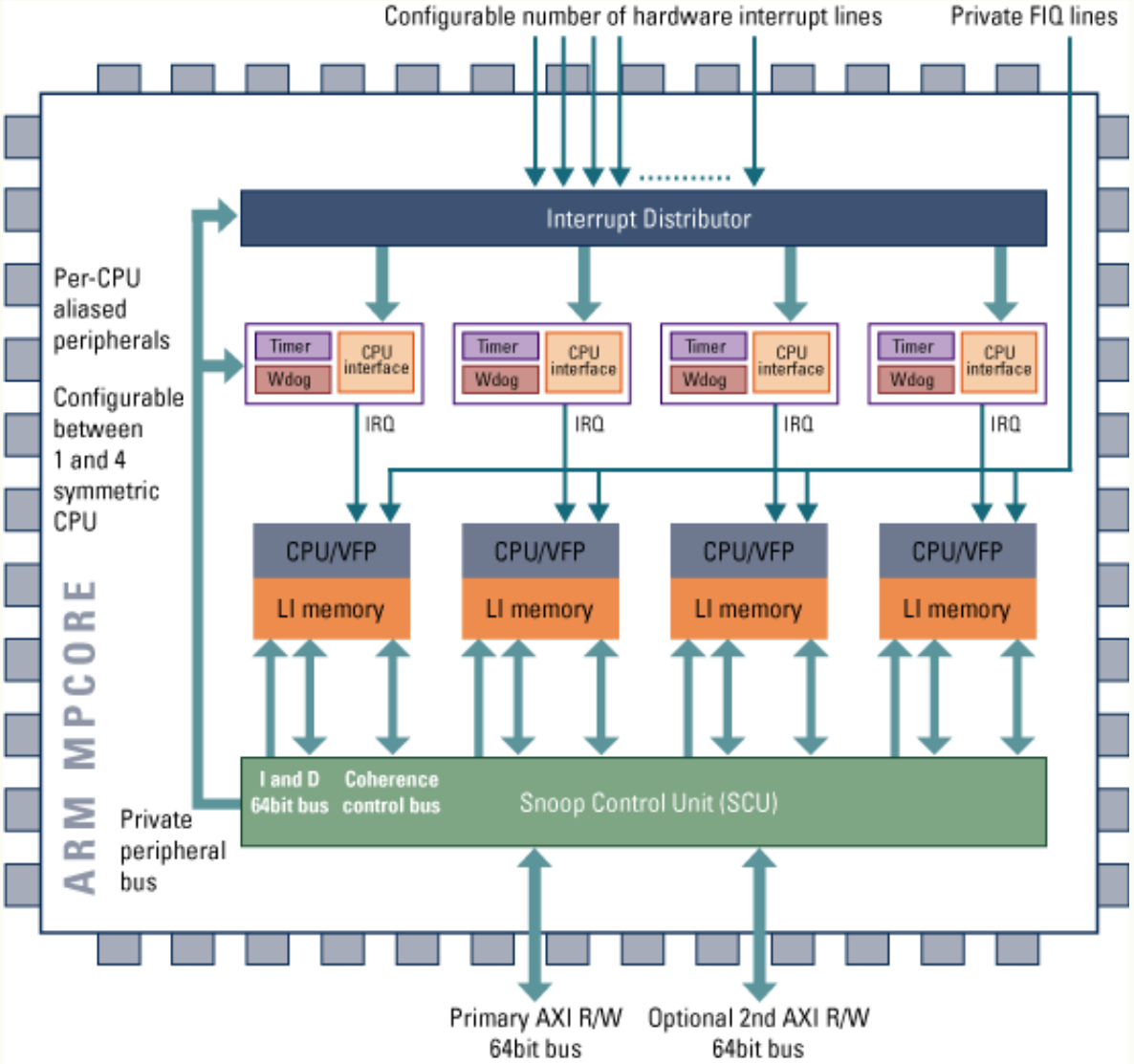
- Massively Parallel Processing (MPP)
  - plusieurs processeurs communiquants
  - chaque processeur => mémoire, OS, et application
- Symetric Multi Processing (SMP) :
  - plusieurs coeurs + une mémoire partagée (cache)
  - une seule instance de l'OS et de l'application
- Non-Uniform Memory Access (NUMA)
  - plusieurs coeurs + mémoire locale
  - mémoire partagée (cache)

Multi-core

- **Multi-coeurs homogènes**
  - **Intel Core2** — **2 /4 coeurs**
  - **Sun UltraSPARCT1** — **8 coeurs, 32 threads**
  - **ARM11 MPcore** — **2 /4 coeurs**
- **Multi-coeurs hétérogènes**
  - **IBM CellBe** — **1 PowerPC + 8 CPU spécialisés**
  - **AMD Fusion**

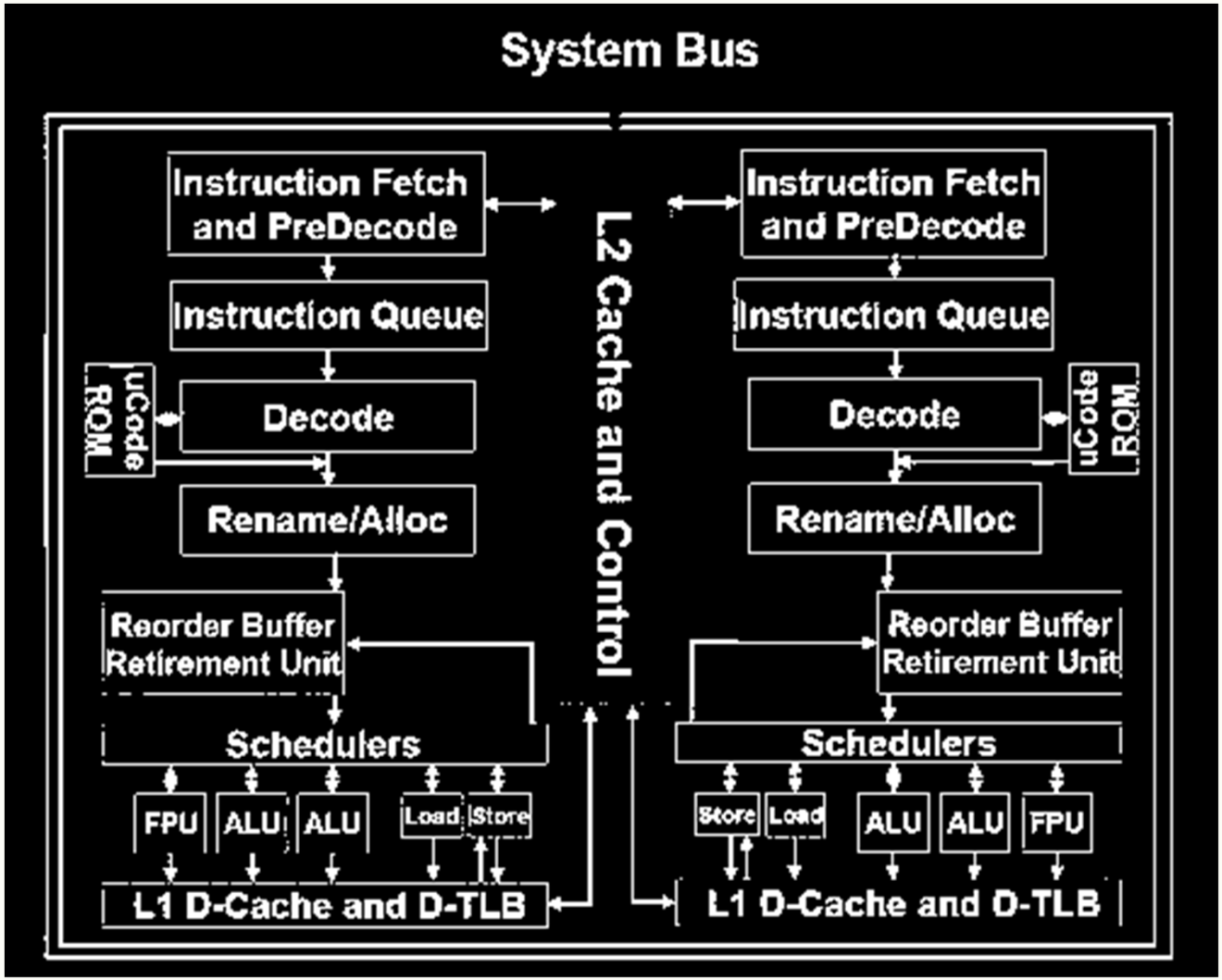
Multi-core

ARM11 MPCore



Multi-coeurs

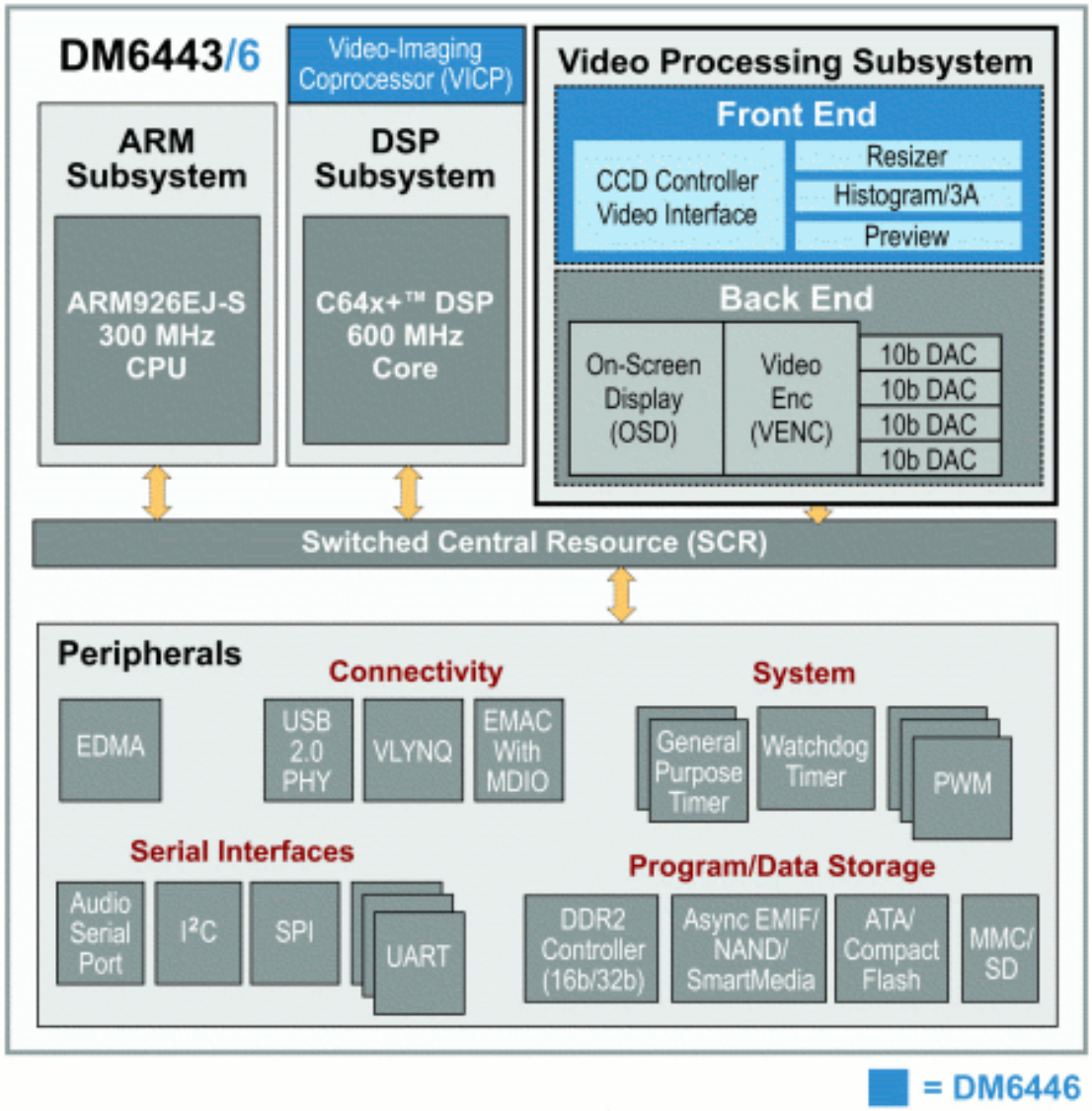
Core 2 Duo



www.hardwarezone.com

Multi-core

Texas Instrument Davinci



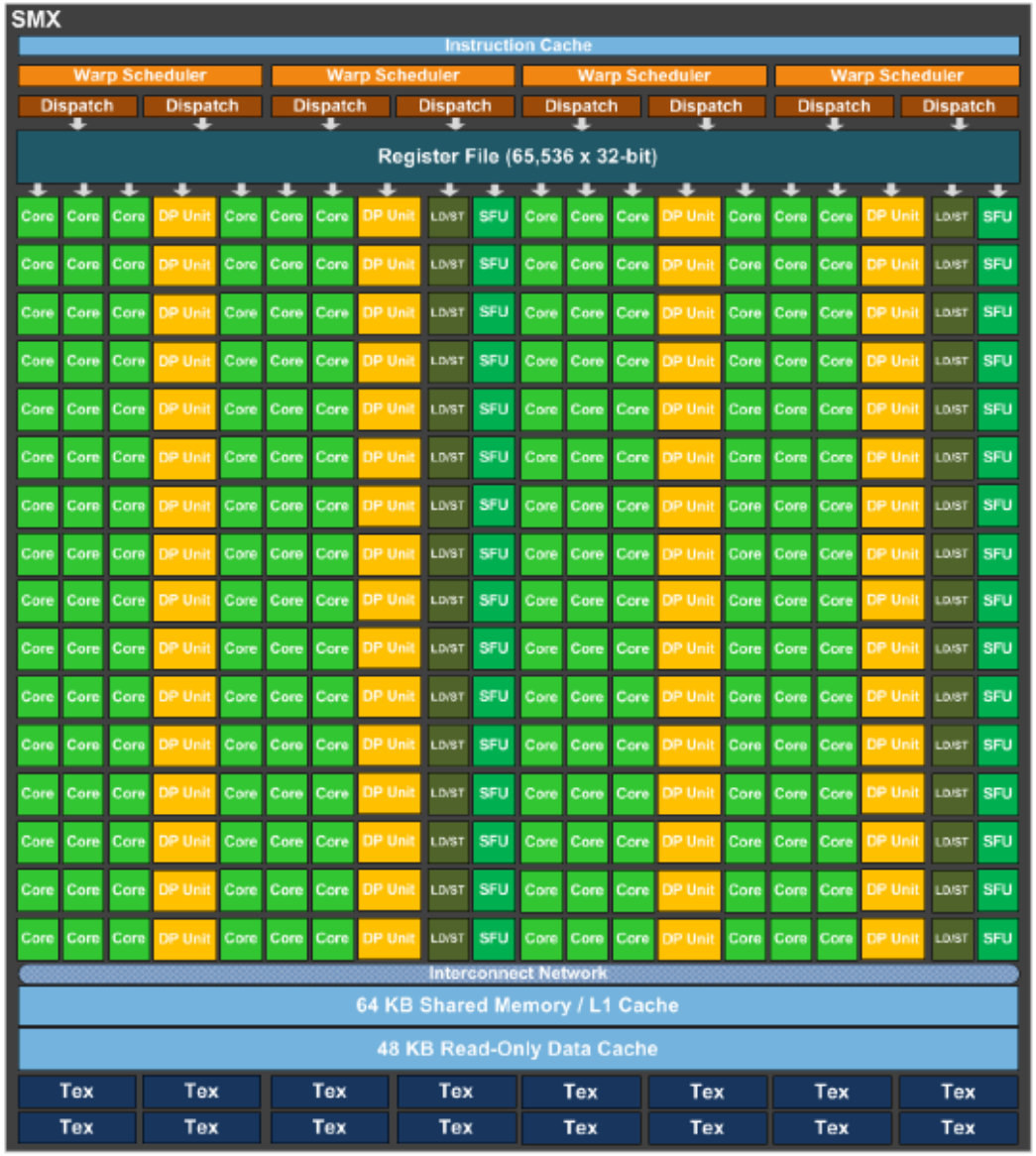
Histoire des GPUs

- Une idée ancienne: ajouter un coprocesseur pour les opérations graphiques
- 2000 : accélérateurs de calculs => instructions spécifiques graphiques pipelinées et vectorisées
- Réflexion : implémenter des instructions générales pour traiter le graphique ?

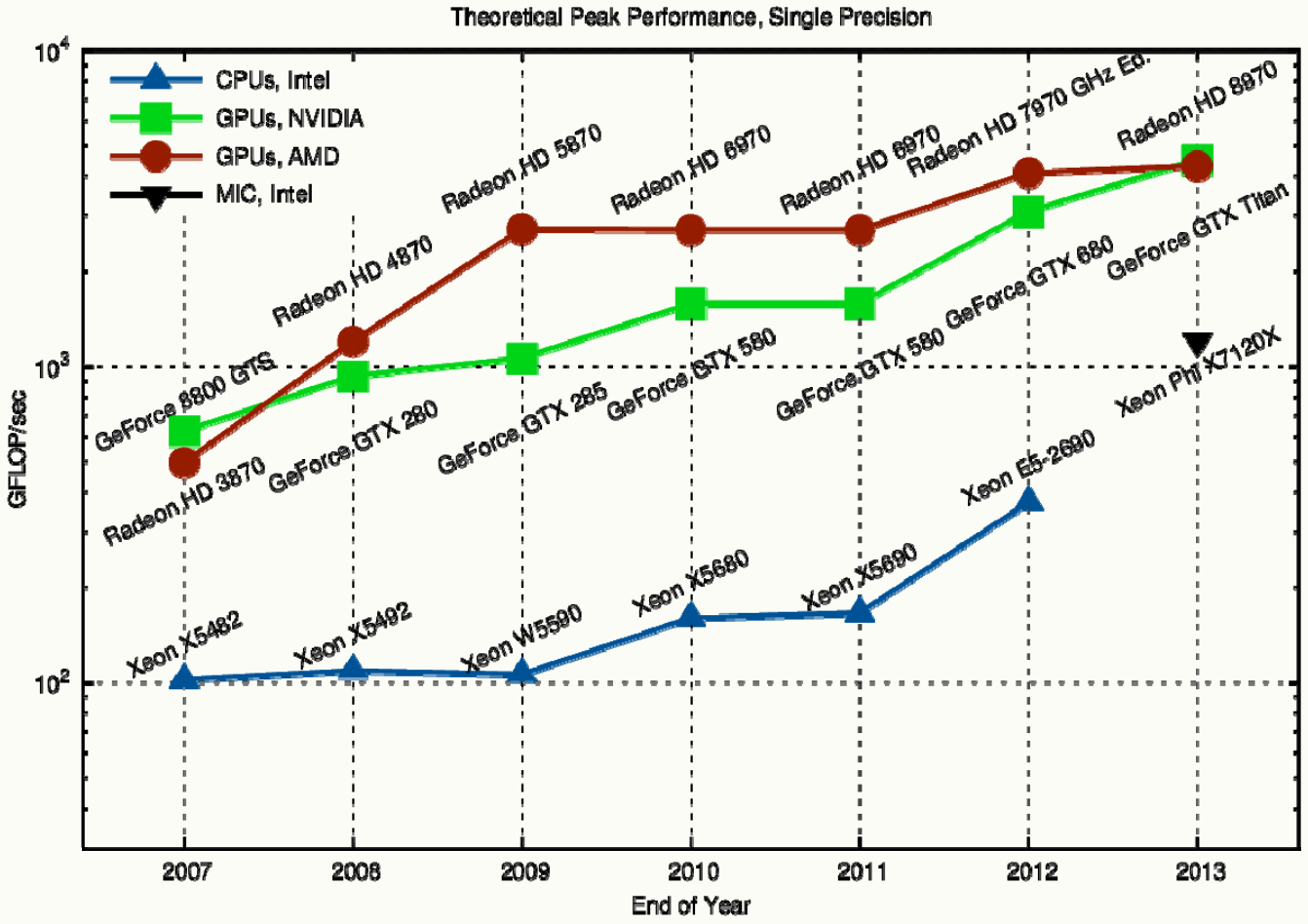
- 2006 : Premier GPU utilisable pour le traitement graphique et le traitement générale haute performance : NVIDIA GT 80 chip/GeForce 8800

GPU multi-core:
NVidia GK110 Kepler Architecture

Inside the NVidia GK110 Kepler : SMX



Evolution des GPUS



FPGA

- Field Programmable gates array
  - Cours Syslog1

- Xilinx
- Altera => Intel PSG
- Atmel
- Lattice
- Actel => Microsemi

Zynq-7000 SoPC

- FPGA avec processeur Arm dual-core Cortex-A9 MPCore (hard core)
- Environnement de développement hardware et software intégré