

Introduction à l'architecture ARM

The ARM logo is displayed in a bold, blue, sans-serif font. The letters 'A', 'R', and 'M' are significantly larger than the 'A' in 'ARM'. A registered trademark symbol (®) is positioned to the upper right of the 'M'.

Petit historique

- Tandis qu'Intel produit des microprocesseurs 8086 et 80286, des employés d'une compagnie anglaise inspirés par les travaux de l'heure travaillent à la conception d'un ordinateur
- En 1985, la compagnie ACORN produit un premier processeur qui résulte de ces travaux: le Acorn RISC Machine (ARM)
- La complexité du ARM est réduite par choix et par faute de moyens et le jeu d'instructions est simplifié.
- Alors que le 8086 gagne en popularité, ACORN et Apple collaborent pour fonder la compagnie ARM Ltd en 1990
- ARM et Intel feront évoluer leurs produits en parallèle dans des marchés différents

ARM

- ARM Holdings:
 - développe des architectures de micro-processeurs et des jeux d'instructions
 - ne construit aucun micro-processeur comme tel! La compagnie licencie la technologie à d'autres qui l'implémentent à leur façon en hardware.
 - clients: Apple, Nvidia, Samsung, Texas Instruments, etc...
- Micro-processeurs ARM
 - Supportent 32 et 64 bits
 - L'architecture **la** plus utilisée au monde
 - 10 milliards produits en 2013
 - 98% de téléphones portables contiennent au moins 1 processeur ARM

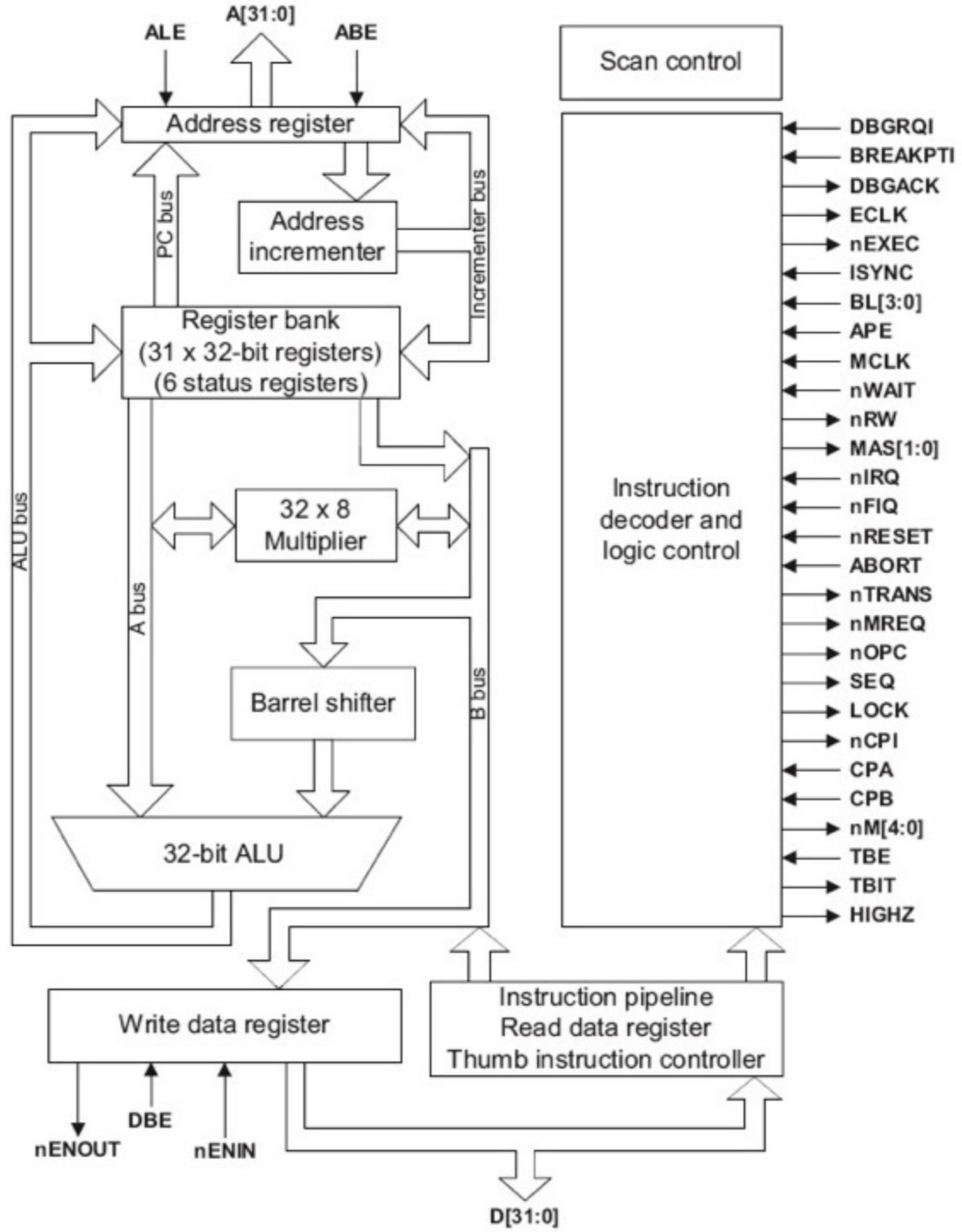
Processeurs ARM

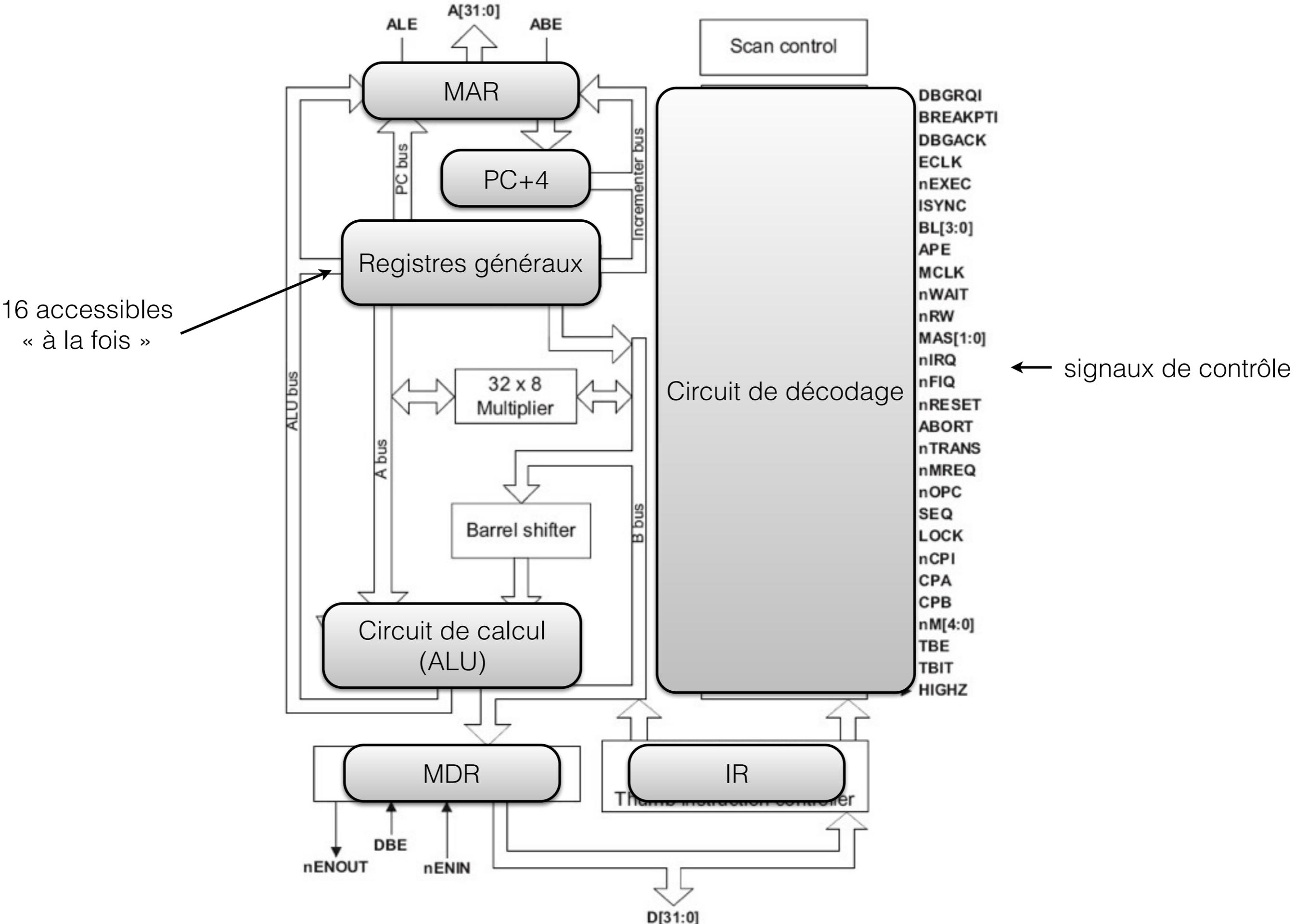
- Plusieurs versions de processeurs, utilisées partout!
 - ARM7TDMI(-S): Nintendo DS, Lego NXT
 - ARM946E-S: Canon 5D Mark ii (caméra)
 - ARM1176JZ(F)-S: Raspberry Pi
 - Cortex-A9: Apple iPhone 4S, iPad2
 - Cortex-A15: Nexus 10
- Beaucoup d'autres!
 - http://en.wikipedia.org/wiki/List_of_applications_of_ARM_cores



Architecture des ARM (32 bits)

- Dans le cours: ARM7TDMI (architecture 32 bits populaire)
 - Registres de 32 bits
 - Instructions de 32 bits
 - Adresses de 32 bits
- Architecture RISC pour laquelle tout passe par des registres
- Accès à combien de mémoire?
 - 2^{32} octets (4GB) de mémoire





Dans la série «Vous n'avez rien à faire ce weekend?»»

<http://www.megaprocessor.com>

3 Affaires à Retenir Mentalement™ (ARM)

- Organisation de la mémoire
- Instructions et PC
- Registres



ARM™ #1: Organisation de la mémoire

TP1

- Taille des emplacements en mémoire: **16** bits
- Chaque emplacement de 16 bits possède une adresse
- Taille du bus de données et des registres: **16** bits
- Taille d'une instruction: **16** bits
- Pas de « endian-ness »

ARM

- Taille des emplacements en mémoire: **8** bits
- Chaque emplacement de 8 bits possède une adresse
- Taille du bus de données et des registres: **32** bits
- Taille d'une instruction: **32** bits
- Little endian (le plus souvent)



Big vs Little Endian

comment stocker un mot de 32 bits (4 octets) à l'adresse 0x0?
exemple: 0x12345678?

Adresse (A)	A+0	A+1	A+2	A+3
0x0	78	56	34	12
0x4				

OU

Adresse (A)	A+0	A+1	A+2	A+3
0x0	12	34	56	78
0x4				

Big vs Little Endian

- La taille minimum d'une donnée stockée en mémoire est habituellement 1 octet
- Les données ayant plusieurs octets peuvent être stockées de 2 façons:
 - "Little endian": l'octet le moins significatif est placé à la **plus petite** adresse dans la mémoire (ex: Intel x86)
 - "Big endian": l'octet le moins significatif est placé à la **plus haute** adresse dans la mémoire (ex: Motorola 68000)



Big vs Little Endian

comment stocker un mot de 32 bits (4 octets) à l'adresse 0x0?
exemple: 0x12345678?

Little Endian

Adresse (A)	A+0	A+1	A+2	A+3
0x0	78	56	34	12
0x4				

Big Endian

Adresse (A)	A+0	A+1	A+2	A+3
0x0	12	34	56	78
0x4				

ARM™ #2: Instructions et PC

- Combien d'adresses mémoire a-t-on besoin pour stocker une instruction?
 - chaque octet possède une adresse
 - une instruction possède 4 octets (32 bits)
 - donc une instruction nécessite 4 adresses mémoire
- 2 considérations importantes sur PC:
 - après chaque exécution, de combien doit-on incrémenter PC?
 - $PC = PC + 4$
 - qu'est-ce que contient PC?
 - PC = adresse de deux instructions plus loin que l'instruction exécutée
 - PC = adresse de l'instruction exécutée **+ 8**

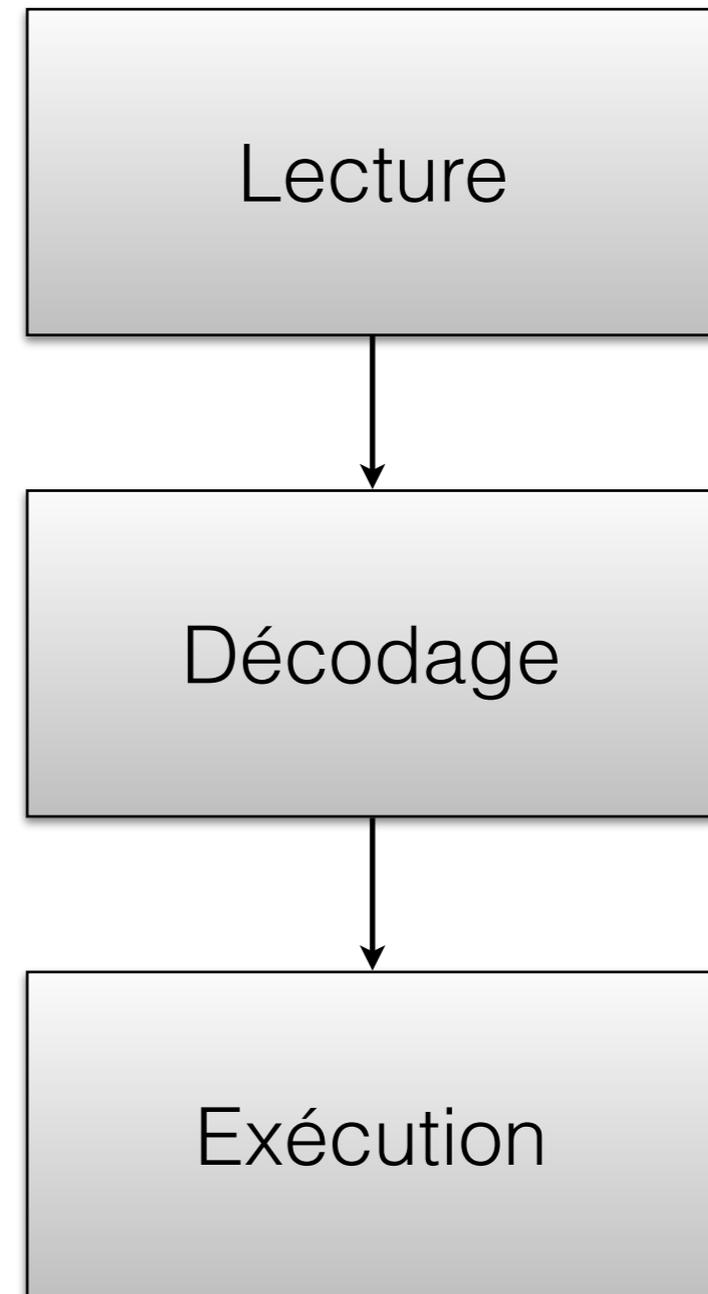


Pipelines

- Début des années 1990
- Idée: séparer l'architecture en deux unités séparées, pouvant être exécutées simultanément
- Comme dans un restaurant:
 - “Fetch-decode”: les serveurs vont prendre les commandes des clients aux tables
 - “Execute”: les cuisiniers préparent les repas

Pipeline ARM

- Le pipeline ARM est divisé en 3
- Que contient PC?
 - L'adresse de la prochaine instruction **à être lue**
- Donc, lors de l'exécution d'une instruction, PC indique l'adresse de **deux instructions** plus loin!



Pipeline ARM

- Le pipeline ARM est divisé en 3
- Que contient PC?
 - L'adresse de la prochaine instruction **à être lue**
- Donc, lors de l'exécution d'une instruction, PC indique l'adresse de **deux instructions** plus loin!

	Adresse	Instruction
exécution	0x0	MOV R0, #0x40
décodage	0x4	LDR R0, [R0]
lecture	0x8	ADD R0, R1, R2
	0xC	...

Donc, PC contient l'adresse de l'instruction courante (exécutée) **+ 8!**

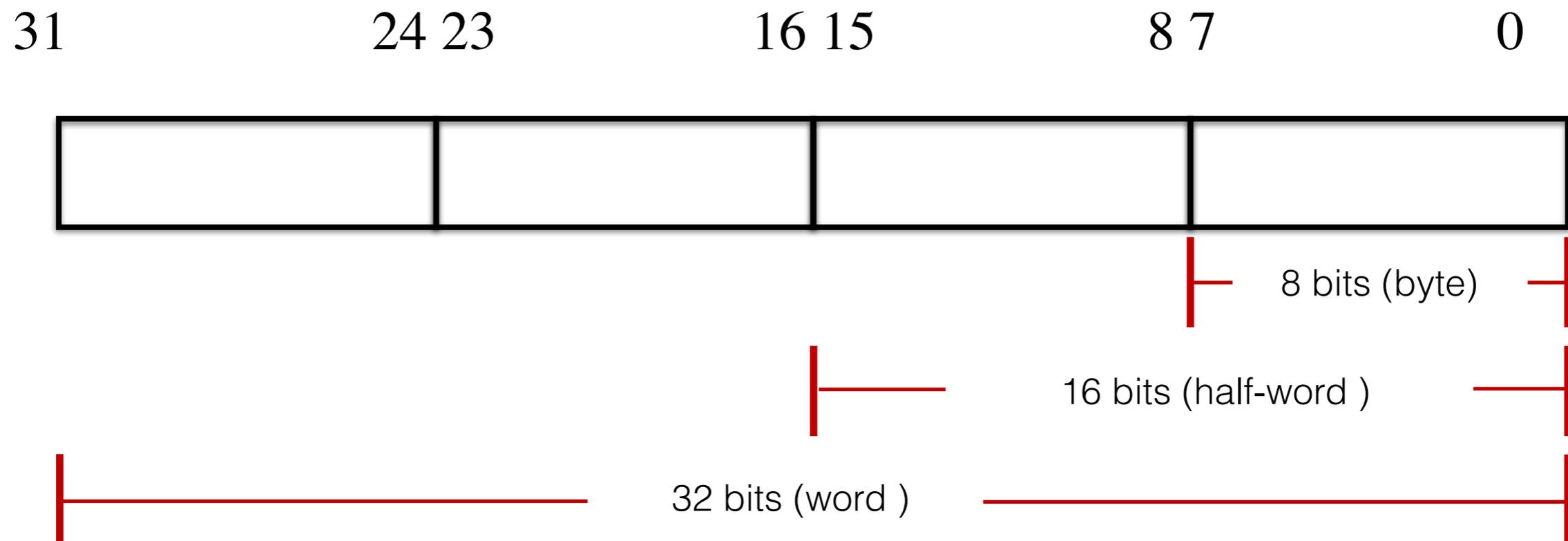
ARM™ #3: Registres

- 16 registres de 32 bits sont disponibles:
 - R0 à R12: usage général
 - R13 à R15: registres “spéciaux”
 - R13: Pointeur de pile (Stack Pointer ou SP)
 - R14: Registre de liens (Link Register ou LR)
 - R15: Compteur de programme (Program Counter ou PC)
- Le “Current Program Status Register” (CPSR) est utilisé pour mémoriser les résultats d’opération



Registres d'usage général

- Gestion des nombres signés ou non à 8, 16 ou 32 bits
- Les calculs sont faits à 32 bits et la gestion des valeurs plus petites est logicielle

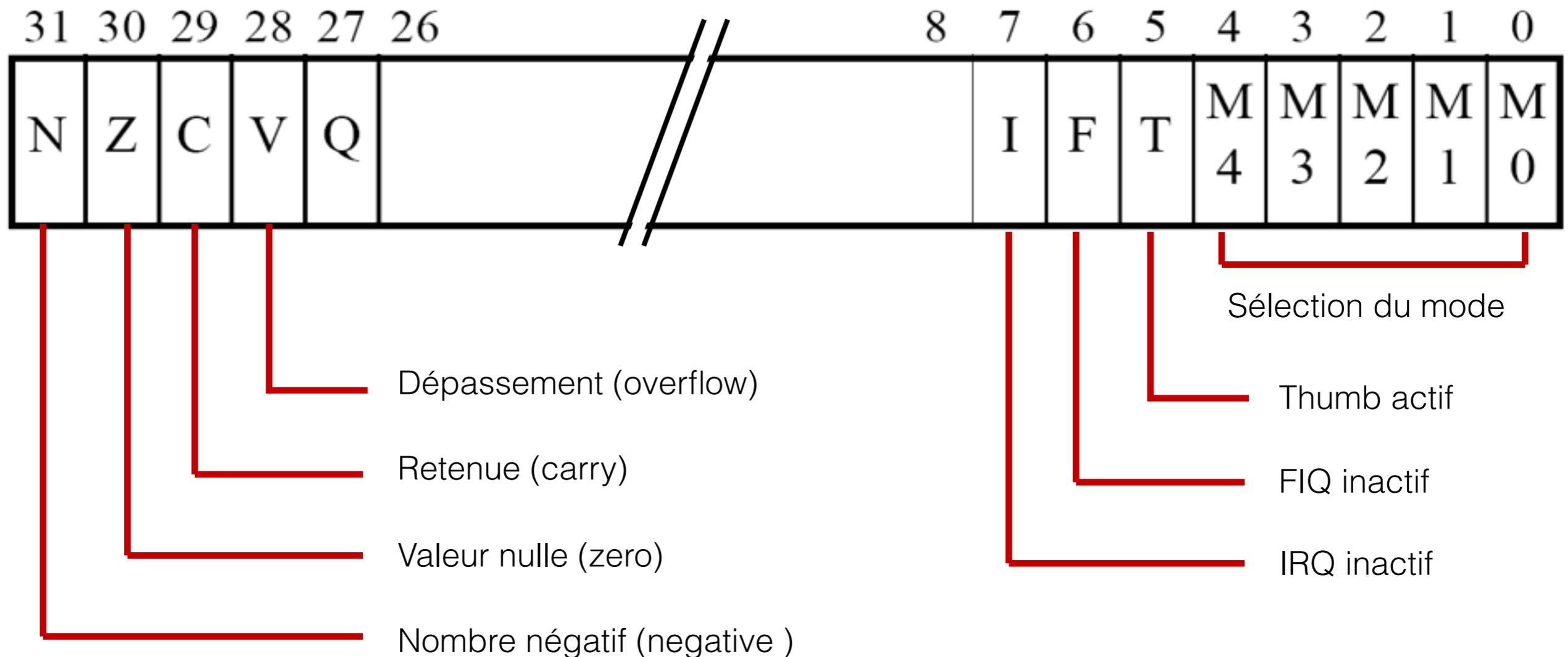


Le registre “Program Counter” R15

- Le contenu du PC indique à quel endroit de la mémoire se trouve la prochaine exécution que le processeur doit *lire*
- PC identifie des octets tandis que la mémoire est composée de mots de 4 octets
- Il y a 2^{32} octets qui peuvent être adressés donc $2^{32}/4 = 2^{30}$ mots de 4 octets à représenter avec le registre PC
 - C’est dire que les deux bits de poids le moins significatif du registre PC ne servent pas
- 2 considérations importantes:
 - après chaque exécution, de combien doit-on incrémenter PC?
 - **PC = PC + 4**
 - qu’est-ce que contient PC?
 - PC = adresse de deux instructions plus loin que l’instruction exécutée
 - PC = adresse de l’instruction exécutée **+ 8**

Registre de statut (CPSR)

- Un registre de statut (dans l'ALU) qui décrit l'état du processeur



CPSR: Détection de conditions

- N: Détection de signe négatif
 - 1 si résultat < 0 , 0 autrement
- Z: Détection de zéro
 - 1 si résultat = 0, 0 autrement
 - Souvent utilisé pour détecter les égalités
- C: Détection de retenue (“carry”)
 - 1 si l’opération a impliqué une retenue
 - Ex. retenue d’addition de nombres positifs
- V: Détection de dépassements (overflow)
 - 1 si l’opération a impliqué un dépassement
 - Ex. dépassement signé lors d’une addition

Retenue (« carry ») avec nombres *non-signés*

- Nombres *non-signés* sur 4 bits
- $10 + 8 = ?$
- $0b1010 + 0b1000 = 0b1\ 0010$
 - Nous avons besoin d'un 5e bit pour que le résultat soit valide, c'est le bit « carry »!

Retenue (« carry ») avec nombres *signés*

- Nombres *signés* (complément-2) sur 4 bits
- $-2 + 2 = ?$
 - $0b1110 + 0b0010 = 0b1\ 0000$
 - Le résultat est valide sur 4 bits, mais un 5e bit est activé: il y a « carry »!
- Est-ce qu'il y a débordement?
 - Non car les deux bits de signes étaient différents initialement.

3 Affaires à Retenir Mentalement™ (ARM)

- Organisation de la mémoire
 - Chaque octet possède une adresse
 - Mots stockés en little endian
- Instructions et PC
 - $PC = PC + 4$
 - PC indique l'adresse de l'instruction courante **+8**
- Registres
 - 16 registres généraux
 - 3 registres spécifiques: SP, LR, PC
 - 1 registre d'état (CPSR)

