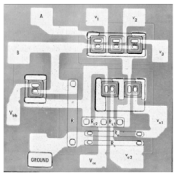
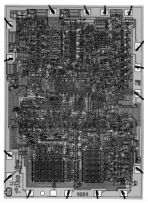
Partie 1 - (Micro)processeurs

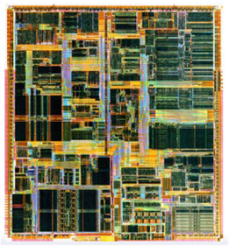
Historique



Premier circuit intégré bipolaire, Motorola, porte 3 entrées, 1966



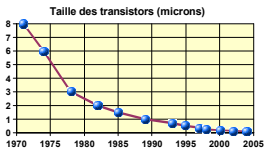
1971, Intel 4004, 1000 transistors 1 MHz NMOS-only tech pour le gain en vitesse

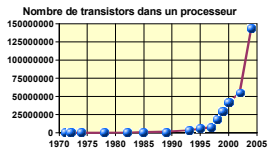


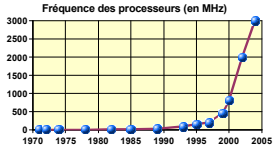
2000, Pentium 4 en CMOS (CMOS depuis '70 afin de réduire la puissance consommée) Multi-million portes

En numérique, le CMOS est aujourd'hui la technologie dominante

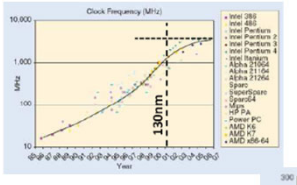
"Loi" de Moore

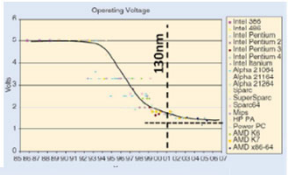




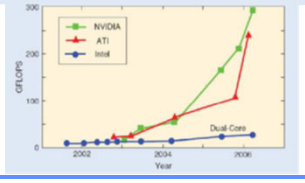


MAIS point d'inflexion !!!





L'évolution suivante est liée aux architectures (parallélisme niveau instruction ILP, multi-coeur ou spécifiques)



Augmentation de la puissance de calcul

Augmentation fréquence ? Limites (conso., ...) => Réduction durée d'un traitement vs. durée d'un ensemble de traitements (séquentiel vs. parallèle), ou les deux …

Instruction-Level Parallelism (ILP)

… + organisation mémoire (von Neumann vs. Harvard, caches …)

Thread-Level Parallelism (TLP)

Data-Level Parallelism (DLP)

Dans un processeur, ou au niveau système ("architectures parallèles")

Co-processeurs … logiciel vs. matériel
FPGA et architectures reconfigurables
…

Puissance de calcul des microprocesseurs

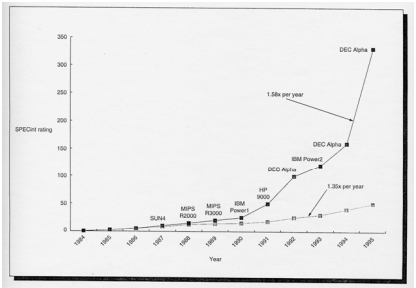
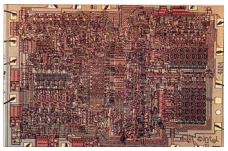


FIGURE 1.1 Growth in microprocessor performance since the mid 1980s has been substantially higher than in earlier years. This chart plots the performance as measured by the SPECint benchmarks. Prior to the mid 1980s, microprocessor performance growth was largely technology driven and averaged about 35% per year. The increase in growth since then is attributable to more advanced architectural ideas. By 1995 this growth leads to more than a factor of five difference in performance. Performance for floating-point-oriented calculations has increased even faster.

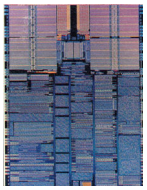
[Hennessy & Patterson 96]

Du CPI à l'IPC



Nov. 1971 : introduction du I4004
2300 transistors, 12mm²

25 ans



1996
IBM présente le P2SC
15M transistors (x6500), 335 mm² (x29),
>x32 instructions/cycle, >x20.000 MIPS

Electronique International Hebdo
31 Oct. 1996

Processeurs et domaines d'application

Contraintes liées au domaine

Systèmes embarqués

- **E/S multiples (standard ou spécialisées)**
- **Faible coût, sûreté, surface maximum, consommation, …**
- **Puissance de calcul adaptée, efficacité énergétique (Perf/W)**

From … → **MCU 8/16 bits …**

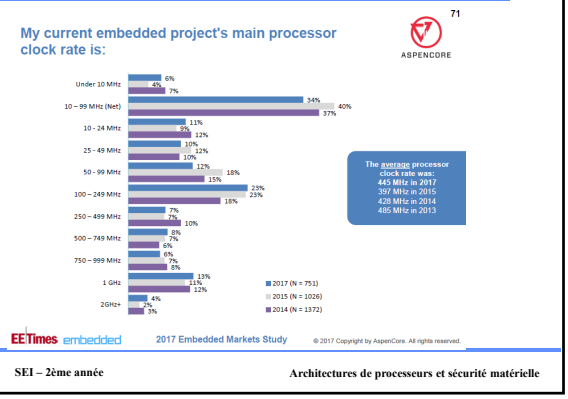
… to → **Many-core CPU 32/64 bits …**

PC/Serveur

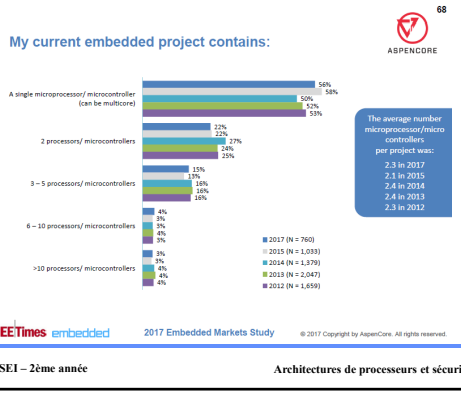
- **Puissance de calcul …**
- **… et autres**

→ **MPU/CPU superscalaire / SMT 32/64 bits …**

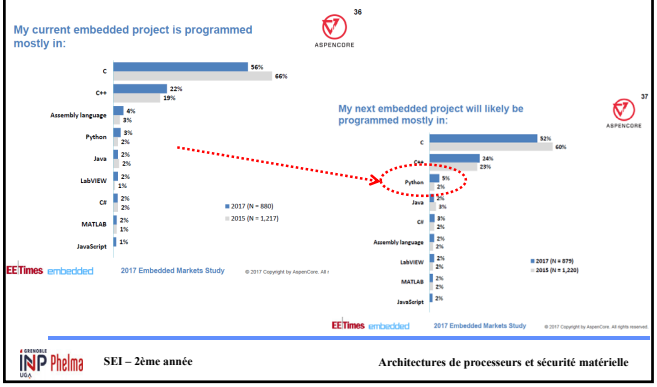
Fréquences d'horloge



Nombre de processeurs

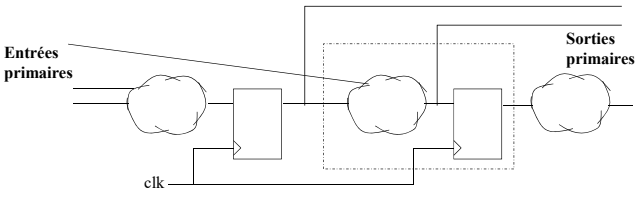


Langages de programmation



Circuit logique synchrone - rappel

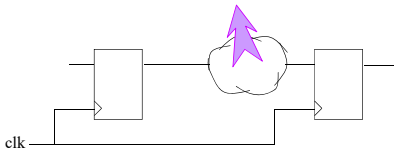
- Hors mémoires : logique combinatoire entre deux "barrières" de registres + connexions à des E/S primaires
- Fréquence globale limitée par le bloc le plus "lent" entre deux barrières + synchronisation des entrées primaires



Un sous bloc (ou étage) interne de base

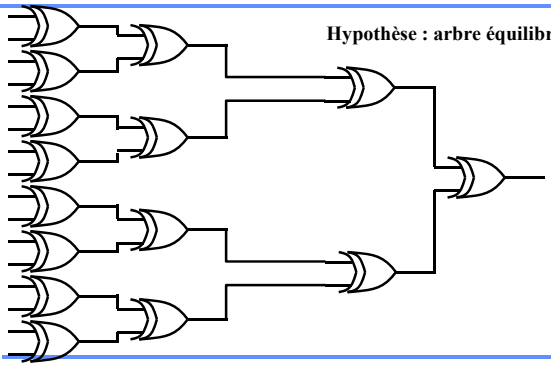
- Logique combinatoire entre deux "barrières" de registres
- Comment augmenter la fréquence d'horloge ??

Exemple : calcul de la parité sur 16 bits
F = a ⊕ b ⊕ c ⊕ d ⊕ e ⊕ f ⊕ g ⊕ h ⊕ i ⊕ j ⊕ k ⊕ l ⊕ m ⊕ n ⊕ o ⊕ p
Implantation sur une bibliothèque avec des portes XOR2

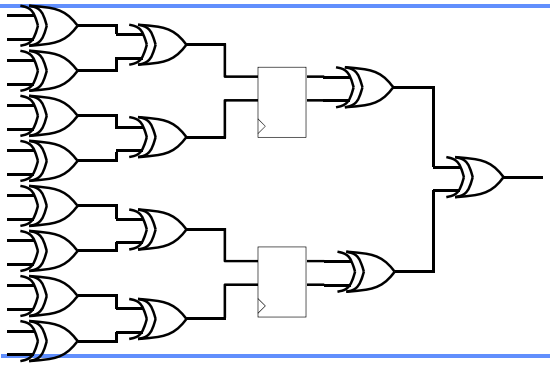


Réalisation de la fonction de base

Hypothèse : arbre équilibré



Ajout d'un pipeline 2 étages

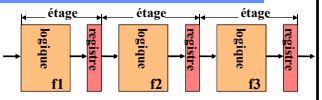


Impact du pipeline

- 2 étages
  - Ajout de 4 bascules
  - Gain en fréquence ?
  - Gain en temps de calcul ?
  - Impact sur la latence ?
  - Gain en débit ?
- 4 étages ?
- Et 3 étages ?

Structure pipeline - Résumé

- Partition + ajout de registres (composition de fonctions)
- Utile pour améliorer la fréquence maximum



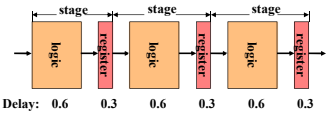
F = f3 o f2 o f1

- Si la décomposition peut être équilibrée et si possibilité au niveau système (maillon faible = étage le plus lent, limitation supplémentaire potentielle par l'horloge système)
- Mais pas dans un rapport N avec N étages ($T_{setup}/T_{H->Q}$ des registres)

- N'améliore pas forcément les performances !
  - Calculs multiples enchaînés possibles ? Sinon latence augmentée sans gain en puissance de calcul
  - Taux d'utilisation des étages (=> cf. aléas dans les processeurs)
  - Energie (registres, arbres d'amplification), surface …

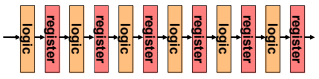
Impact des bascules : pipeline vs. superpipeline

Micro-architecture typique du Pentium 3 :



0,9 ns par étage
=> 1,1 GHz max.

Pentium 4 ? :

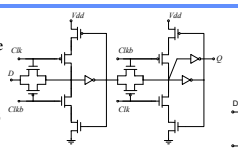


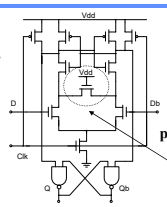
0,6 ns par étage
=> 1,67 GHz max.
(latence doublée, aléas plus nombreux, …)

Augmentation de la fréquence => choix des éléments de mémorisation pour réduction des retards (compromis avec énergie, surface …)

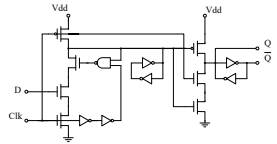
Bascules D utilisées dans des processeurs

Transfer-Gate Master-Slave Latch (Gerosa) PowerPC 603

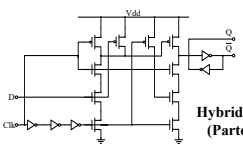




SA110 StrongARM low-power 0,5 W (basé sur structure DEC Alpha 21264, 1 MOS de plus pour opération statique, résistance aux courants de fuite)



SDFF -Semi Dynamic Flip-Flop (Klass) - UltraSparc III



HLFF Hybrid Latch Flip-Flop (Partovi) – AMD K6

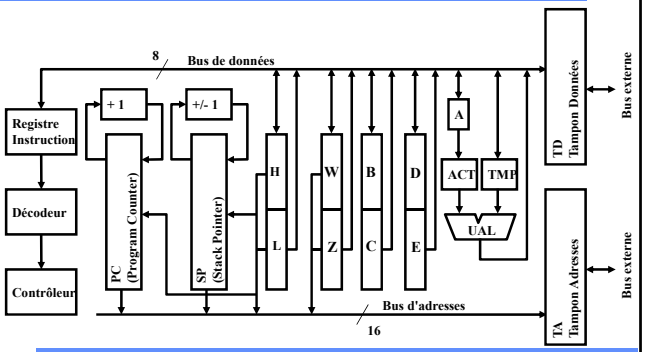
Micro-parallélisme (ILP vers DLP/TLP)

- Evolution progressive

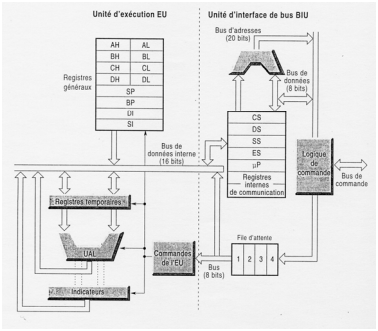
CISC
- 8 bits : limitations technologiques => quasiment aucun parallélisme des opérations internes (exemple : I8080)
- Prefetch (exemple : MC68000)
- Queue d'instructions, séparation de l'unité d'exécution (exemple : I8088)
- RISC (ISA régulier, pipeline) + pipelines dans les CISC
- Superpipeline, superscalaire et VLIW
- Instructions SIMD (DLP)
- MT, SMT, … (TLP)

- Architecture Von Neumann vs. Harvard
  - Séparation données/instructions
  - Augmentation de la bande passante, parallélisation des accès mémoire
  - Contraintes liées au nombre de broches

Architecture du microprocesseur I8080



Architecture du microprocesseur I8088



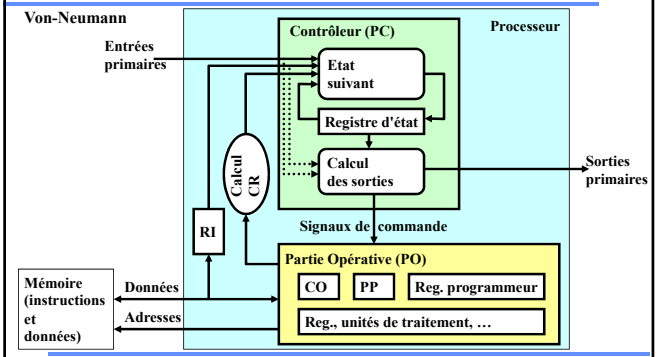
[Lilen 95]

Modèles d'exécution

Modèle d'exécution	mémoire-mémoire	mémoire-accumulateur	mémoire-registre	registre-registre	pile
Génération (exemples)	ordinateurs (VAX 11)	µp 8 bits (I8080, M6800)	µp 16 bits (I8086, M68000)	µp RISC (I80860,M88000)	(Transputer)
Opérandes (mem. ext., spécifiées)	(3,3)	(1,1)	(1,2)	(0,3)	(0,0)
Inst. calcul	Mem op Mem -> Mem	ACC op Mem -> ACC	Rx op Mem -> Rx	Rx op Ry -> Rz	Pile1 op Pile2 -> Pile1
Exécution M1 op M2 -> M3	1 instruction Op M1, M2, M3	3 instructions Load M1, ACC Op M2 Store ACC, M3	3 instructions Load M1, Rx Op Rx, M2 Store Rx, M3	4 instructions Load M1, Rx Load M2, Ry Op Rx, Ry, Rz Store Rz, M3	4 instructions Push M1 Push M2 Op Pop M3
Cas particulier			(0,2) : Rx op Ry -> Rx		

Note : le modèle registre-registre est aussi appelé "chargement-rangement"
(seules les instructions Load et Store accèdent à la mémoire externe de données)

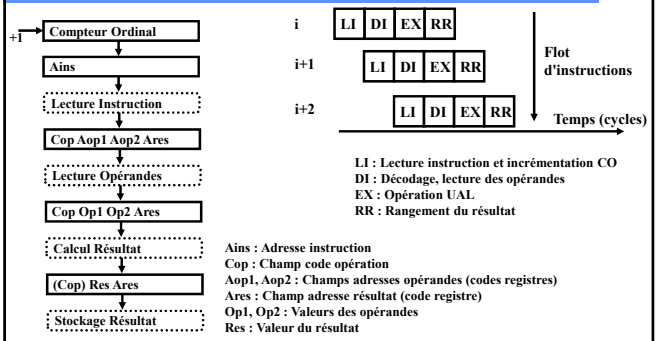
Architecture de base d'un (micro-)processeur



Interprétation d'une (macro-)instruction

- Principales phases :
  - Recherche du code de l'instruction (fetch), incrémentation CO
  - Décodage
  - Calcul d'adresses (selon mode d'adressage)
  - Transfert des opérandes
  - Exécution (traitement)
  - Stockage du résultat
- Phases utiles dépendantes de l'instruction et du mode d'adressage (donc initialement du modèle d'exécution puis de la définition de l'ISA)

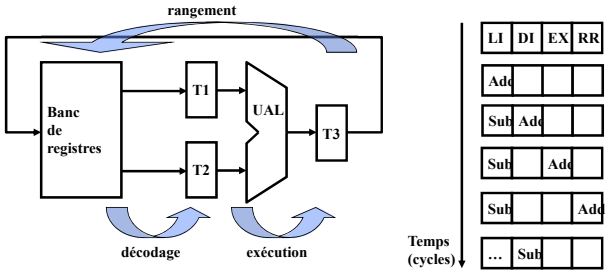
Architecture pipeline : principe de base (Op calcul)



LI : Lecture instruction et incrémentation CO
DI : Décodage, lecture des opérandes
EX : Opération UAL
RR : Rangement du résultat

Ains : Adresse instruction
Cop : Champ code opération
Aop1, Aop2 : Champs adresses opérandes (codes registres)
Ares : Champ adresse résultat (code registre)
Op1, Op2 : Valeurs des opérandes
Res : Valeur du résultat

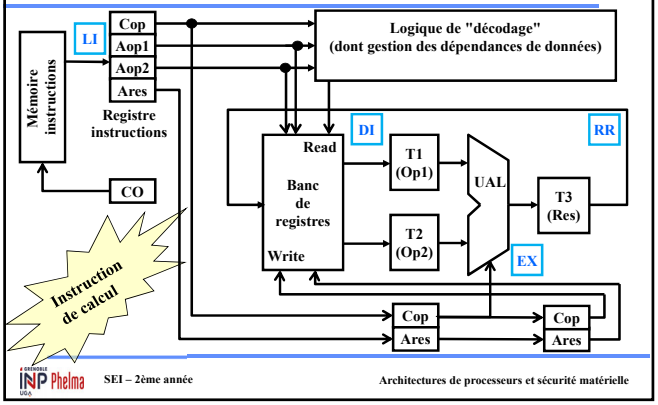
Partie opérative pipelinée (principe)



Exemple d'aléa d'exécution :
Add R1, R2, R3
Sub R3, R4, R5

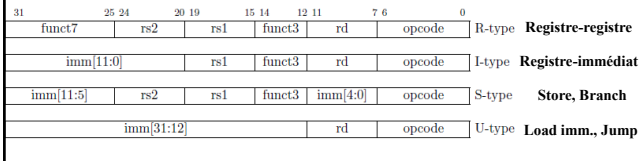
=> 2 "bulles" propagées

Structure pipeline pour calculs - PO + Contrôle



Régularité du codage des instructions (ISA)

- Exemple : RISC V, instructions 32 bits "entier" de base



The RISC-V ISA keeps the source (rs1 and rs2) and destination (rd) registers at the same position in all formats to simplify decoding. Except for the 5-bit immediates used in CSR instructions, immediates are always sign-extended, and are generally packed towards the leftmost available bits in the instruction and have been allocated to reduce hardware complexity. In particular, the sign bit for all immediates is always in bit 31 of the instruction to speed sign-extension circuitry.

https://riscv.org/specifications/ riscv-spec-v2.2

Aléas dans un pipeline (RISC ou CISC !)

- Aléas structurels
  - Accès simultanés à une même ressource par plusieurs étages
  - Exemple : accès à un cache unique données/instructions …
- Aléas d'exécution
  - Dépendances de données
  - Branchements
  - Exceptions (interruptions)
- Réduction du CPI apparent => nécessité de techniques de gestion des aléas pour limiter le nombre de "bulles"

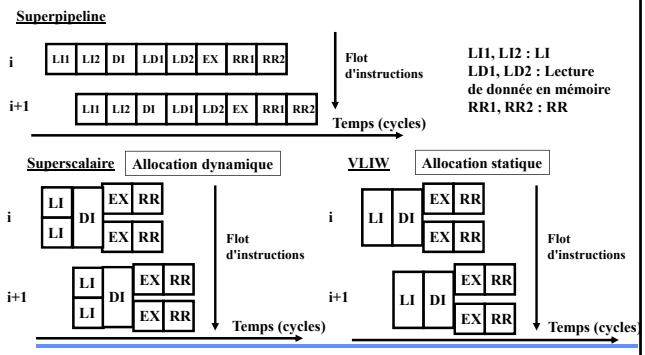
Principales caractéristiques RISC

- Modèle d'exécution chargement-rangement
- Format d'instruction de longueur fixe (1 mot) ou en tous cas régulier (pas "réduit" aujourd'hui !!) ,
- avec modes d'adressage adaptés à la régularité du codage et à la limitation du nombre de bits pour coder une instruction => peu de modes d'adressage (surtout 2 : immédiat, base + déplacement M[Ri+dep]) avec possibilité de Load/Store par demi mots ou par octets, éventuellement avec décalages
- exemple MIPS : "LUI" Load Upper Immediate => chargement immédiat des 16 bits de poids fort d'un registre, avec mise à 0 des 16 bits de poids faible)
- "Grand" nombre de registres banalisés (32 …)

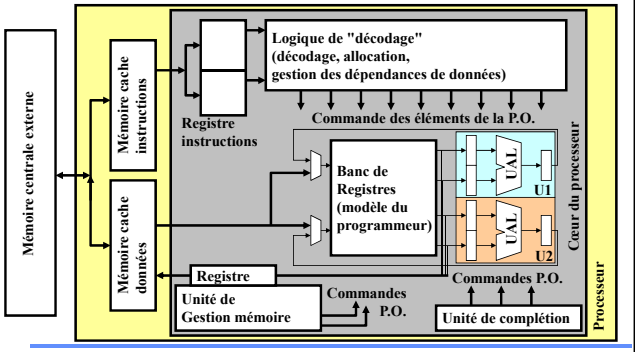
Apparu RISC, fréquent dans les CISC récents

- "Grand" nombre de registres banalisés (32 …)
- Pipeline interne / techniques de gestion des aléas
- Contrôleur (FSM) remplacé par les registres de contrôle (commandes internes) du pipeline
- Utilisation de caches pour réduire les temps d'accès mémoire
- CPI (apparent) de l'ordre de 1 ou inférieur

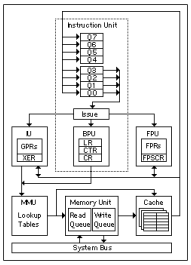
Architecture pipeline : extensions (principes)



Processeur superscalaire élémentaire



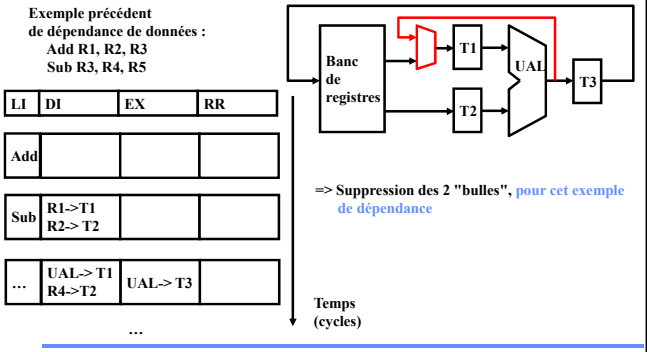
PowerPC 601



- Fetch : jusqu'à 8 instructions chargées en queue => allocation en désordre
- BPU : unité de prédiction de branchements, 2 étages de pipeline
- IU/FPU : traitements entiers/flottants, 4 et 6 étages de pipeline
- Accès mémoire (Load/Store) : 5 étages de pipeline
- Cache : associatif 8 voies, unifié instructions/données
- MMU : adressage jusqu'à 4 Gb de mémoire physique et 4 Peta-octets de mémoire virtuelle (2^{52} octets … ou 6,2 millions de CD-ROM !)

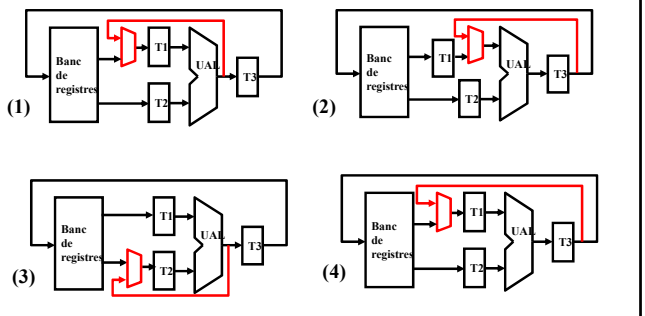
Traitement d'une dépendance de données

Exemple précédent de dépendance de données :
Add R1, R2, R3
Sub R3, R4, R5



=> Suppression des 2 "bulles", pour cet exemple de dépendance

Modifications équivalentes ??

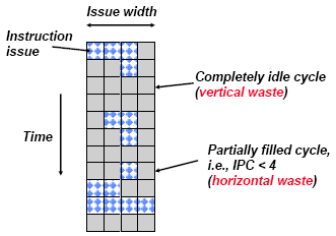


Traitement d'un aléa d'exécution (branchement)

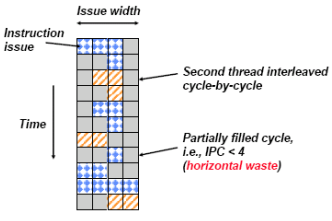
- **Trois grandes approches**
  - **Branchements retardés (jeu d'instructions des premiers RISC) :** exécution de l'instruction suivant le Br quelque soit la valeur de la condition – "Delay slot"
  - **Anticipation de branchement (ex. PowerPC 601) :** si la condition est connue, remplacement de l'instruction Br par l'instruction destination du saut, dans la file d'attente d'instructions, avant le décodage effectif ("issue") => pas de pénalité dans le pipeline, et la lecture d'instructions continue en séquence avec l'instruction destination. Possibilité aussi de 2 queues d'instructions avec sélection lorsque la condition est connue (SuperSparc).
  - **Prédiction de branchement : exécution spéculative.**
    - **Prédiction statique :** règle prédéfinie
    - **Prédiction dynamique :** mémorisation des adresses des instructions de branchement exécutées et des adresses de destination calculées pour chacune d'elles
- **Remarque : possibilité en compilation de "dérouler" les boucles avec condition d'arrêt fixe (boucles for) pour supprimer le branchement … Ou utilisation de "boucles matérielles"**

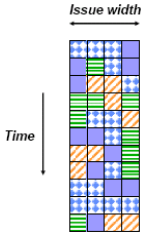
Parallélisme niveau tâches : motivation

- **Exécution d'une tâche unique : utilisation limitée des ressources disponibles en fonction**
  - **Des types d'instructions à exécuter par rapport aux unités disponibles**
  - **Des dépendances et autres aléas (dont accès mémoire)**



Parallélisme niveau tâches : gains



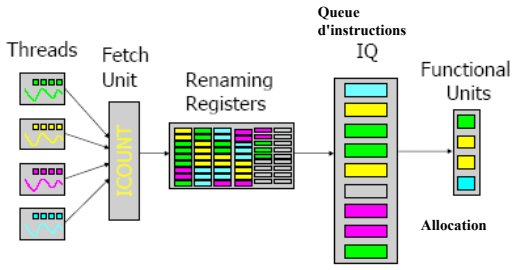


MT ("vertical")
=> Gain sur les cycles perdus

SMT "idéal"
=> Utilisation maximisée

Schémas issus de [www.cs.ucr.edu/~bhuyan/cs162]

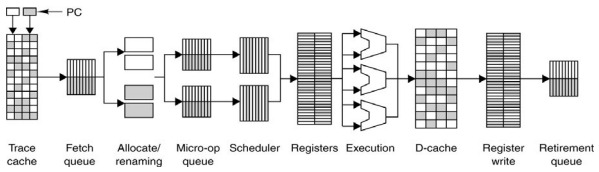
Du superscalaire au processeur SMT



Renommage de registres : utilisé en superscalaire pour supprimer les fausses dépendances, exploité aussi pour les fausses dépendances entre tâches (utilisation de registres supplémentaires, non visibles dans l'ISA)

Microarchitecture Pentium 4 NetBurst

- **Partage de ressources (unités de traitement, gestion des aléas, mémoires …)**
- **Augmentation du nombre d'étages de pipeline mais différent du superpipeline (sous-tâches distinctes entre étages)**



Parallélisme niveau tâches : coûts

- **Gestion de l'état de chaque tâche et maintien de la cohérence globale**
  - **Compteur de programme par tâche**
  - **Pile (et pointeur de pile) par tâche**
  - **Gestion des identificateurs de tâche sur l'ensemble des étapes de traitement (incluant les tables d'allocation des caches)**
  - **Gestion des queues d'instructions par tâche**
  - **Gestion des complétions par tâche**
- **Pertes de performances dues aux changements de contexte**

Parallélisme niveau tâches - vue d'ensemble

- **MT : multi-threading**
  - **"Entrelacement" des recherches d'instructions sur plusieurs tâches**
  - **Gestion en parallèle de plusieurs environnements d'exécution**
  - **Changement possible de contexte entre tâches : grain fin (chaque cycle), sur un aléas d'accès au cache niveau 1, sur un aléas d'accès au cache niveau 2, sur attente d'une requête externe …**
- **SMT : simultaneous multi-threading**
  - **Extension du MT => chargement simultané dans la queue d'instructions, à chaque cycle, d'instructions provenant de plusieurs process ou tâches**
  - **Permet d'augmenter l'utilisation des différentes ressources de traitement (ex. : unités de calcul entier et de calcul flottant)**
- **HT : hyper-threading technology (Intel) => Pentium 4 à 3,06 GHz**
  - **SMT + techniques de prédiction au niveau tâche**
  - **+ techniques de parallélisation des accès dans la hiérarchie mémoire (MLP – Memory Level Parallelism)**

=> Lien très étroit avec les systèmes d'exploitation !!!!

Comparaison avec multi-coeur (débit)

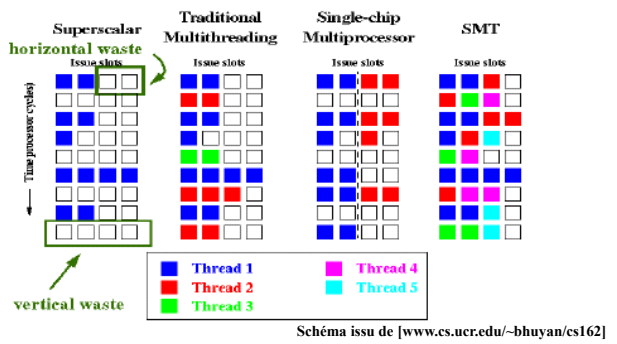


Schéma issu de [www.cs.ucr.edu/~bhuyan/cs162]

Parallélisme niveau tâches : exemples

- MIPS32 – Famille 34K
  - Processeurs uniscalaires (1 résultat max / cycle), 9 niveaux de pipeline
  - Mais SMT (au niveau de l'étage de lecture), jusqu'à 5 tâches (micro-tâches, ou programmes distincts) traitées simultanément dans le pipeline
  - Réduction du coût des changements de contexte, possibilité de masquer les attentes dues aux opérations lentes (ex. accès mémoire données)
  - Conserve la compatibilité avec les logiciels à tâche unique ("single-threaded")
- Sun : processeur Niagara 2, destiné aux serveurs
  - 64 tâches traitées simultanément …
  - Possibilité de coupler 2 processeurs pour traiter 128 tâches simultanément de manière cohérente.
  - Problème de gestion par l'OS, contraintes de synchronisations entre tâches, limitation de la bande passante interne et externe => quelle est le niveau limite d'efficacité du parallélisme niveau tâche ??

[Microprocessor Report, février 2006]

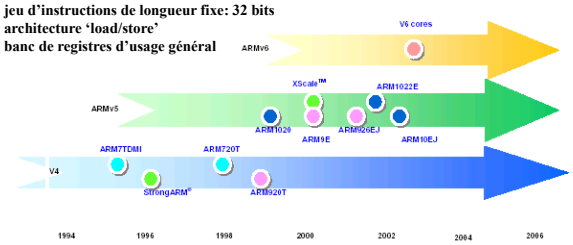
Caractéristiques de processeurs du commerce

- Exemples (de 1993 …)
  - Pentium
    32 bits, CISC superscalaire, 3,1 MTransistors, 294 mm² CMOS 0,8µ
    60-66 MHz – Conso 12-16W – SpecInt92 64,5 – SpecFp92 56,9
    $900 par 1000 pièces
  - PowerPC 601
    32 bits, RISC superscalaire, 2,8 MTransistors, 120 mm² CMOS 0,65µ
    66-80 MHz – Conso 7W (à 66 MHz) – SpecInt92 80 – SpecFp92 105
    $550 par 1000 pièces

Où est l'erreur ?

Evolution de l'architecture ARM

Architecture RISC :
- jeu d'instructions de longueur fixe: 32 bits
- architecture 'load/store'
- banc de registres d'usage général



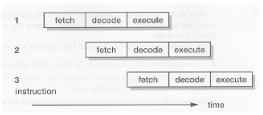
Quelques spécificités :
- jeu d'instructions Thumb sur 16 bits (puis Thumb2)
- technologie Jazelle : multiplie la vitesse d'exécution de code Java par 8
- extensions DSP (instructions supplémentaires)

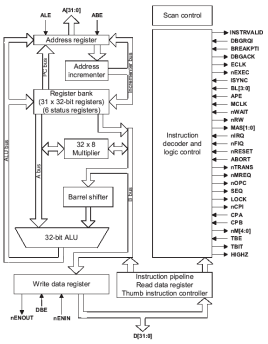
Architecture ARM7TDMI

Basé sur l'architecture v4T

Très utilisé (téléphones portables)

Pipeline 3 étages :



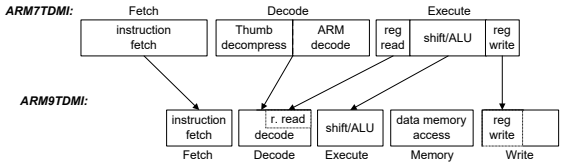


Architecture ARM9TDMI

Amélioration des performances : augmentation de la fréquence d'horloge, diminution du CPI

Architecture Harvard

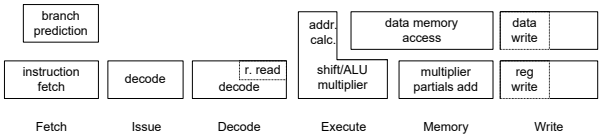
Pipeline à 5 étages :



Architecture ARM10TDMI

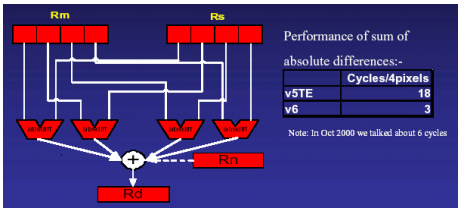
Architecture Harvard

Pipeline évolué à 6 étages :



Ajouts de l'architecture v6

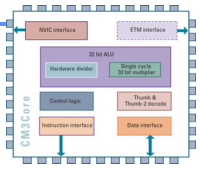
- **Destinée aux applications vidéo/audio**
- **Extension SIMD**

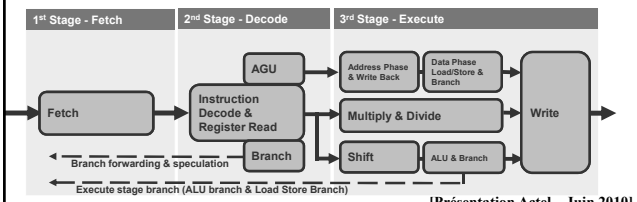


- **Gestion mixte little/big endian**
- **Amélioration de la gestion de la mémoire (caches niveau 1 et 2)**

ARM Cortex-M3 (Central Core)

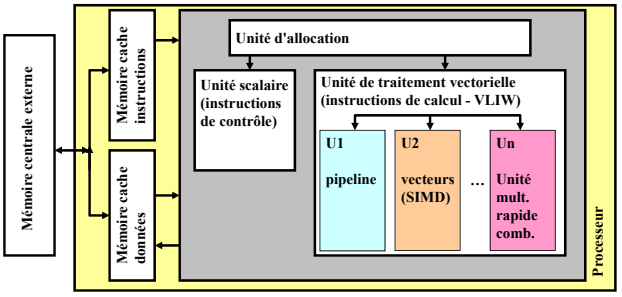
- **Harvard architecture**
  - Separate Instruction & Data buses enable parallel fetch & store
- **Advanced 3-Stage Pipeline**
  - Includes Branch Forwarding & Speculation
- **Additional Write-Back via Bus Matrix**





[Présentation Actel – Juin 2010]

Un processeur DSP haute performance …



Niveaux multiples de parallélisme : combinaison MIMD / VLIW / SIMD / Pipeline …