Ecole thématique Archi 2019, Lorient, 20-24 mai

D'Euclide aux multi-cœurs

Bernard Goossens

Université de Perpignan Via Domitia, DALI-LIRMM

Sommaire

Section 1

La génèse de l'ordinateur

Subsection 1

Algorithme

L'algorithme d'Euclide (2300 ans)

Les éléments, livre VII sur l'arithmétique ($\sim$ -300)

Anthyphérèse (soustraire alternativement)

https ://mathcs.clarku.edu/ djoyce/java/elements/bookVII/propVII2.html

```
Plus_Grand_Commun_Diviseur(a,b)
    while a ≠ b
        if a > b
            a = a - b
        else
            b = b - a
    return a
```

Al Khwârizmî (780 ?-850 ?)

Abrégé du calcul par la restauration et l'équilibre ($\sim$ 820)

Premiers algorithmes de résolution des équations du second degré

Al jabr (réduction) : $x^2 = 40x - 4x^2$ est réduit à $x^2 + 4x^2 = 40x$.

Al muqabala (équilibre) : $x^2 + 5 = 40x + 4x^2$ devient $5 = 40x + 3x^2$.

Subsection 2

Calcul mécanique

Pascal (1623-1662)

Pascaline (1641)

Machine pour effectuer des additions (la soustraction est obtenue par complément à 9, la multiplication par additions successives et la division par soustractions successives)

Babbage (1791-1871)

Note on the application of machinery to the computation of astronomical and mathematical tables (1824)
Table des logarithmes des entiers naturels de 1 à 108000 (1827)

Difference Engine : premier calculateur mécanique pour calculer des constantes physiques à partir de polynômes
La machine de Babbage se base sur la méthode des différences finies de Newton (calcul de polynômes)

Subsection 3

Logique

Aristote (-384 - -322)

Organon (instrument de travail) - Premiers Analytiques ($\sim$ -350)

Syllogisme, prémisses et conclusion

si B (mortel) est affirmé de tout A (être un homme), et A de tout (ou de quelque) C (Socrate), alors B est nécessairement affirmé de tout C
De même, si B est nié de tout A, et A affirmé de tout (ou quelque) C, B est nié de tout (ou quelque) C

Boole (1815-1864)

The Mathematical Analysis of Logic (1847)

Algèbre de Boole et calcul propositionnel

Sheffer (1882-1964)

A set of five independent postulates for Boolean algebras, with application to logical constants (1913)

Barre de Sheffer
NAND : $a \uparrow b = \neg (a \wedge b)$
$\overline{a} = a \uparrow a$
$a \vee b = ((a \uparrow a) \uparrow (b \uparrow b))$
$a \wedge b = ((a \uparrow b) \uparrow (a \uparrow b))$

Subsection 4

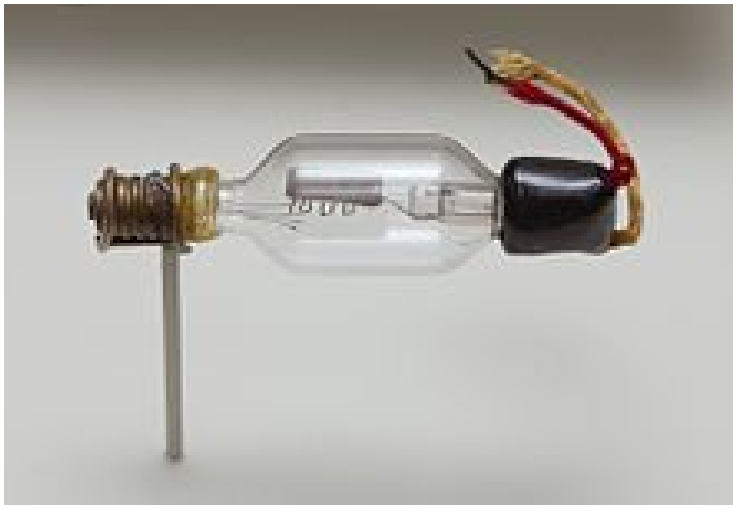
Electronique

Fleming (1849-1945), De Forest (1873-1961)

Fleming : brevet de la diode à vide (1904) (deux électrodes auxquelles on applique une différence de potentiel)

De Forest : triode à vide (1906) (une troisième électrode sert à contrôler le flux d'électrons)

La diode permet le contrôle du sens de déplacement des électrons le long d'un conducteur



Shannon (1916-2001)

A Symbolic Analysis of Relay and Switching Circuits (1937) (Master thesis)

A Mathematical Theory of Communication (1948)

Concepteur des circuits numériques

Lien entre l'algèbre booléenne et les interrupteurs électroniques (TRUE/FALSE ON/OFF)

Père de la théorie de l'information (bit, mot, codage, cryptographie)

Subsection 5

Récapitulatif

De quoi dispose-t-on en 1936 pour imaginer l'ordinateur ?

L'algorithmique (Euclide et Al Khwârizmî)

Le calcul mécanique (Pascal et Babbage)

La logique booléenne (Aristote et Boole)

Le contrôle électronique (Fleming et De Forest)

Les circuits numériques (portes NAND/NOR) (Shannon)

Section 2

La naissance de l'ordinateur

Subsection 1

Le concept

Turing (1912-1954)

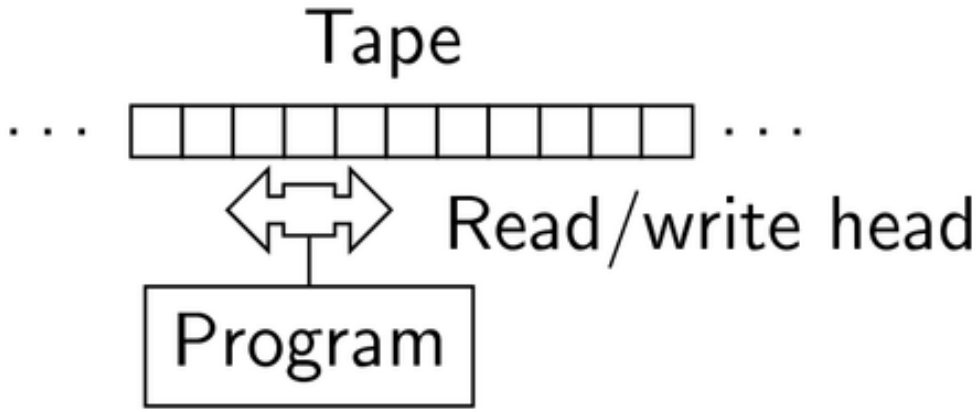
On Computable Numbers, with an Application to the Entscheidungsproblem (1936) (Entscheidungs = décidabilité)

Machine de Turing et machine de Turing universelle

Une machine de Turing calcule toute proposition décidable ou toute fonction récursive

La machine de Turing universelle comprend un second ruban où est écrit le programme chargé de contrôler la machine de Turing

La machine de Turing



Un ruban infini découpé en cellules.

Une tête de lecture/écriture positionnée sur une cellule. La tête peut lire la cellule ou y écrire un mot de l'alphabet. Elle peut aussi avancer (se positionner sur la cellule suivante) ou reculer (cellule précédente).

Un programme pour contrôler la tête de lecture/écriture.

Subsection 2

La réalisation

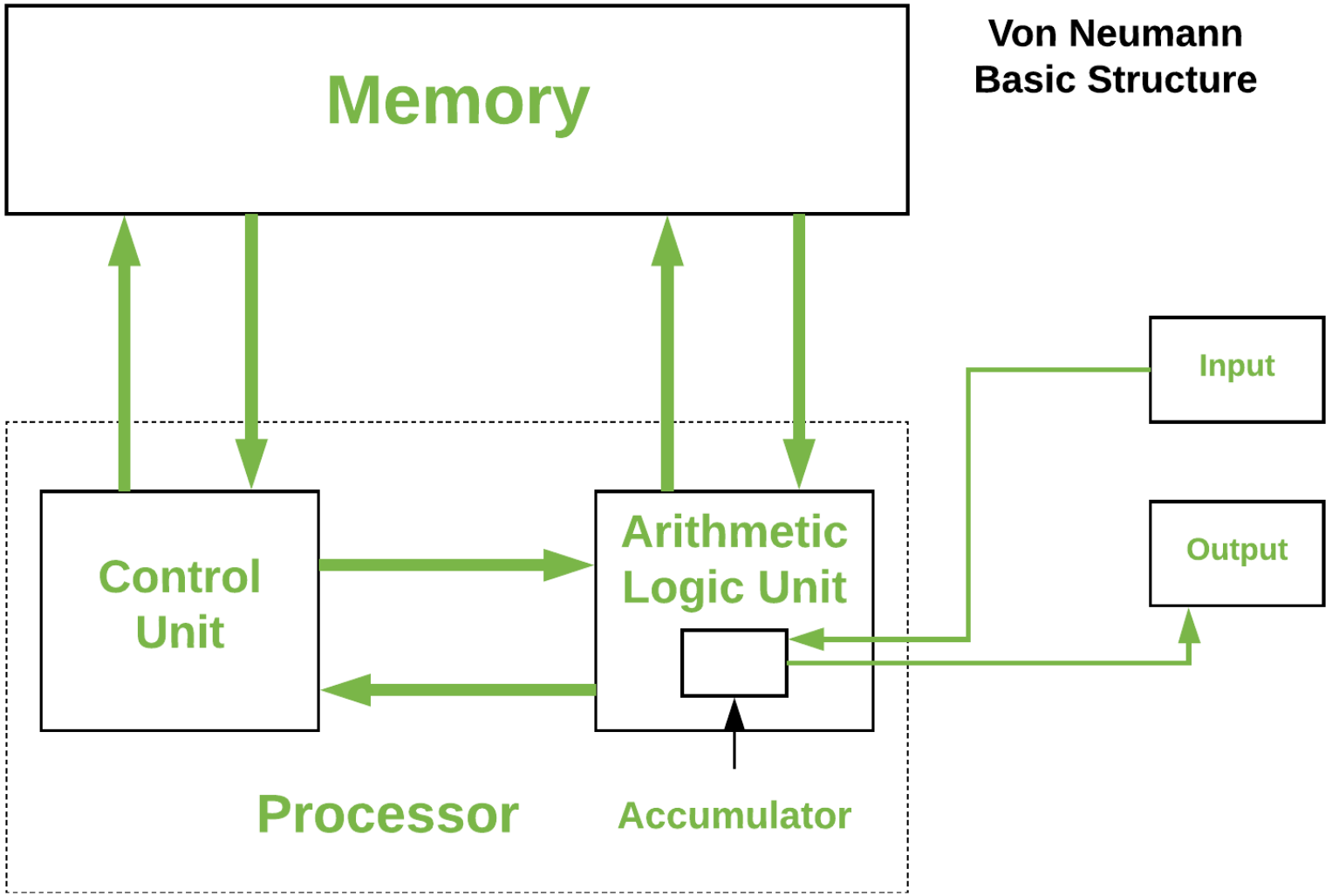
Eckert (1919-1995), Mauchly (1907-1980), Von Neumann (1903-1957)

Von Neumann : First draft of a report on the EDVAC (1945)

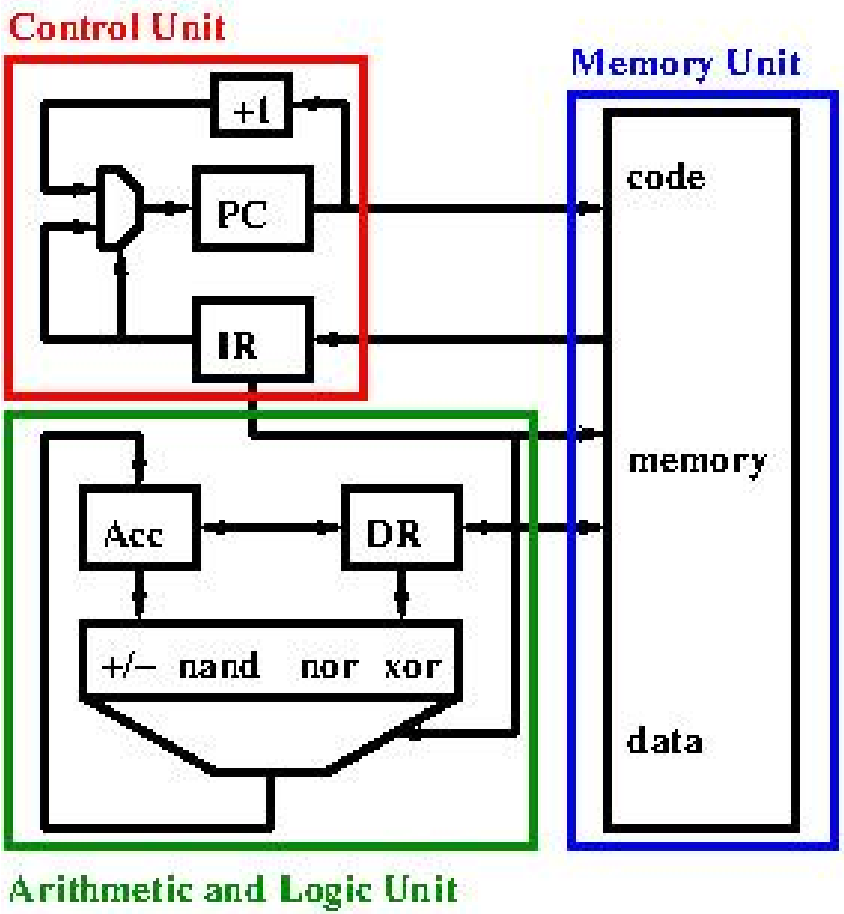
Eckert-Mauchly : ENIAC (1946), UNIVAC (1951)

Concepteurs du premier ordinateur (machine commandée par un programme enregistré en mémoire)

L'architecture de Von Neumann (vue générale)



L'architecture de Von Neumann (vue fonctionnelle)



Program	Run
LD DR, a	a contains A
MOV Acc, DR	b contains B
LD DR, b	Acc contains A
XOR	Acc contains A xor B
MOV DR, Acc	
ST DR, a	a contains A xor B
LD DR, b	
XOR	Acc contains A
MOV DR, Acc	
ST DR, b	b contains A
LD DR, a	DR contains A xor B
XOR	Acc contains B
MOV DR, Acc	
ST DR, a	a contains B

L'architecture de Von Neumann (fonctionnement électrique)

Un courant en entrée pour alimenter chaque circuit, une sortie (GND).

Un flux d'électrons traverse chaque circuit. Une charge négative (présence d'électrons) est interprétée comme la valeur logique FAUX. Une charge positive (absence d'électrons) est interprétée comme la valeur logique VRAI.

La sortie d'un circuit est l'entrée d'un autre.

Un flux sans séparateur (sans verrou ou *latch*) calcule une fonction combinatoire des entrées.

Un verrou fermé sépare la charge (valeur logique) en entrée de la charge en sortie (autre valeur logique).

Les verrous sont ouverts et fermés alternativement par action sur un signal périodique (horloge et cycle).

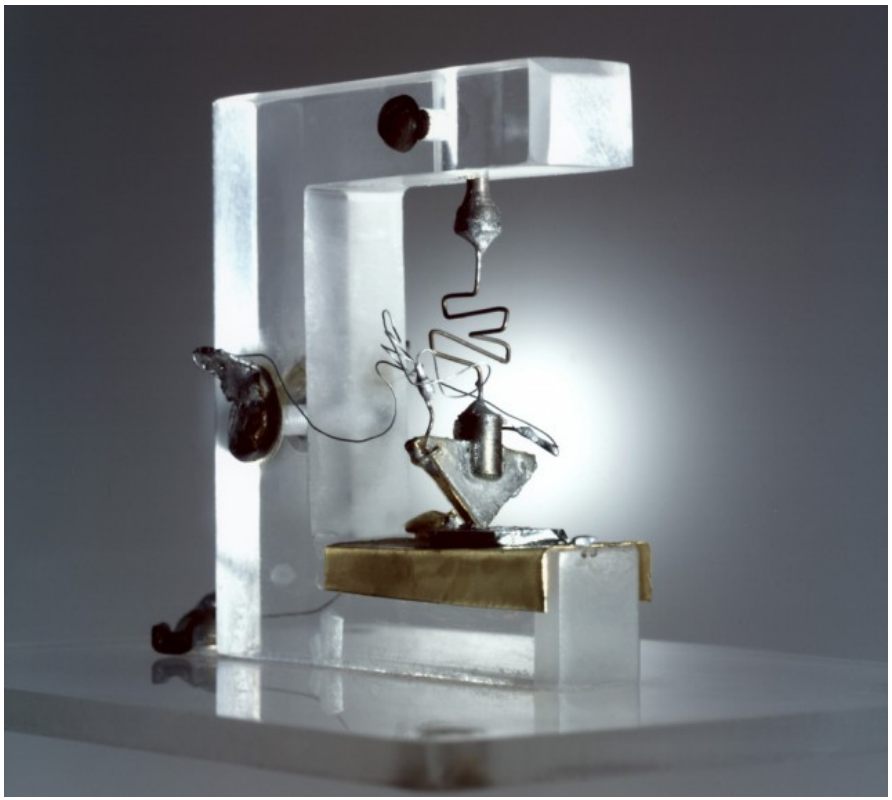
Subsection 3

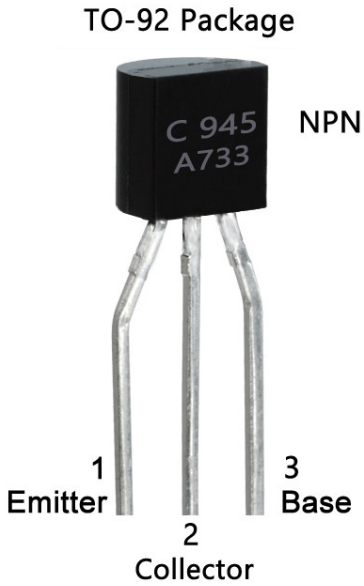
La miniaturisation des composants

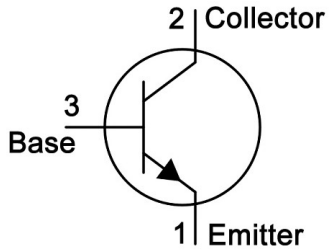
Shockley (1910-1989), Bardeen (1908-1991), Brattain (1902-1987)

Electrons and Holes in Semiconductors (1950)

Inventeurs du transistor au Bell laboratories (1947)



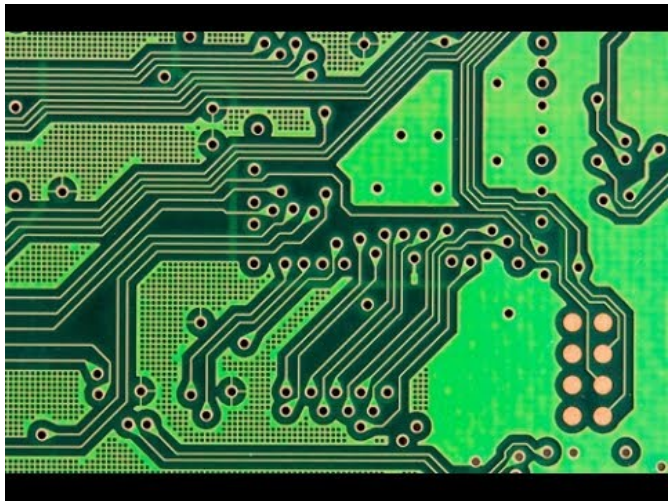


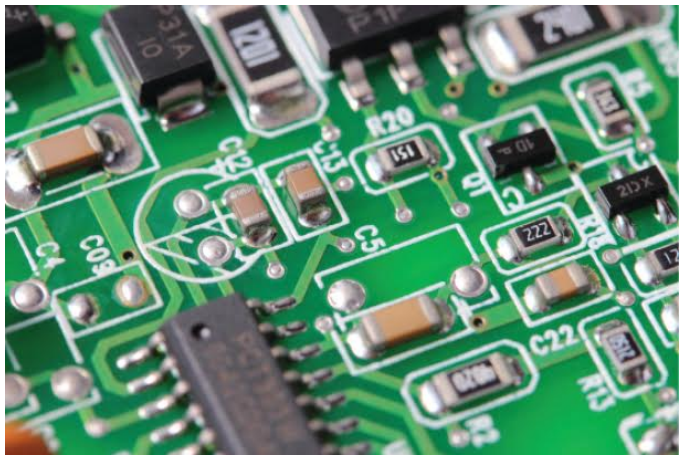


Un transistor est un semi-conducteur (silicium) servant de commutateur (*switch*), en particulier dans les circuits numériques.

Eisler (1907-1992)

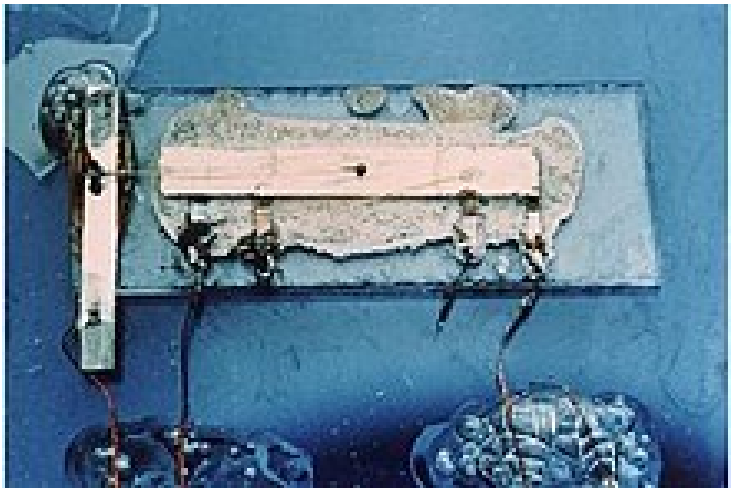
Inventeur du circuit imprimé (Printed Circuit Board ou PCB) (1936)

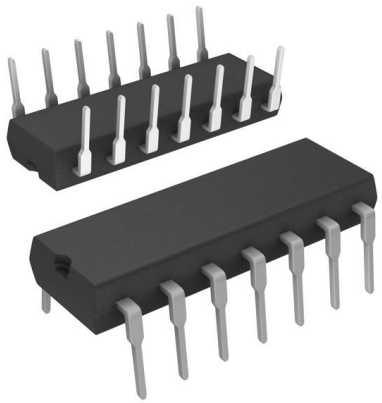


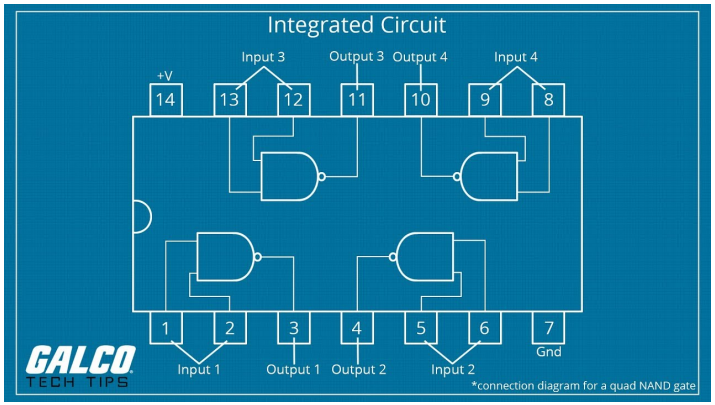


Kilby (1923-2005)

Inventeur du circuit intégré (IC) (1957)







Section 3

L'évolution des processeurs : une course à la performance

L'équation de la performance

$P = 1 / T = F / C$

T est le temps d'exécution (en secondes)

C est le nombre de cycles

F est la fréquence (en Hz ; 1/F est la durée du cycle en secondes)

$P = 1 / T = (I * F) / N$

N est le nombre d'instructions

I est le nombre moyen d'instructions exécutées par cycle (IPC)

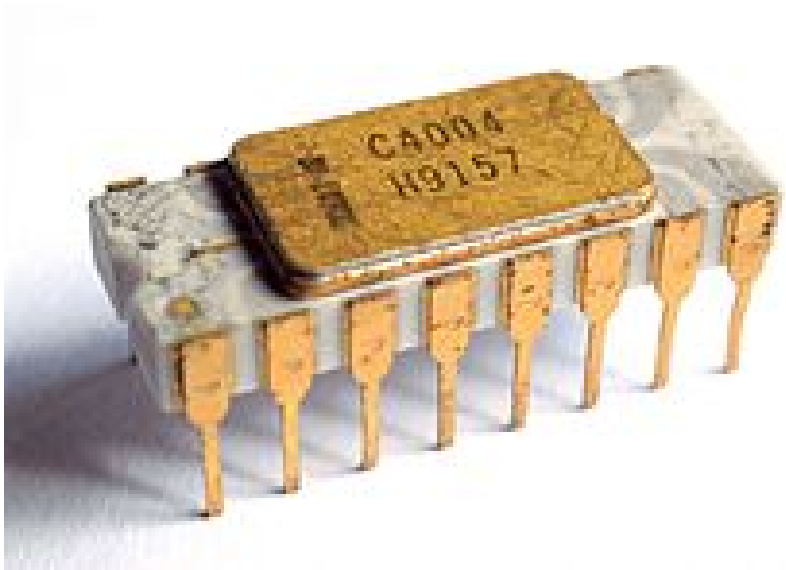
Pour accroître la performance : Augmenter la fréquence (technologie et architecture) ou diminuer le nombre d'instructions exécutées (algorithme et architecture) ou augmenter l'IPC (architecture).

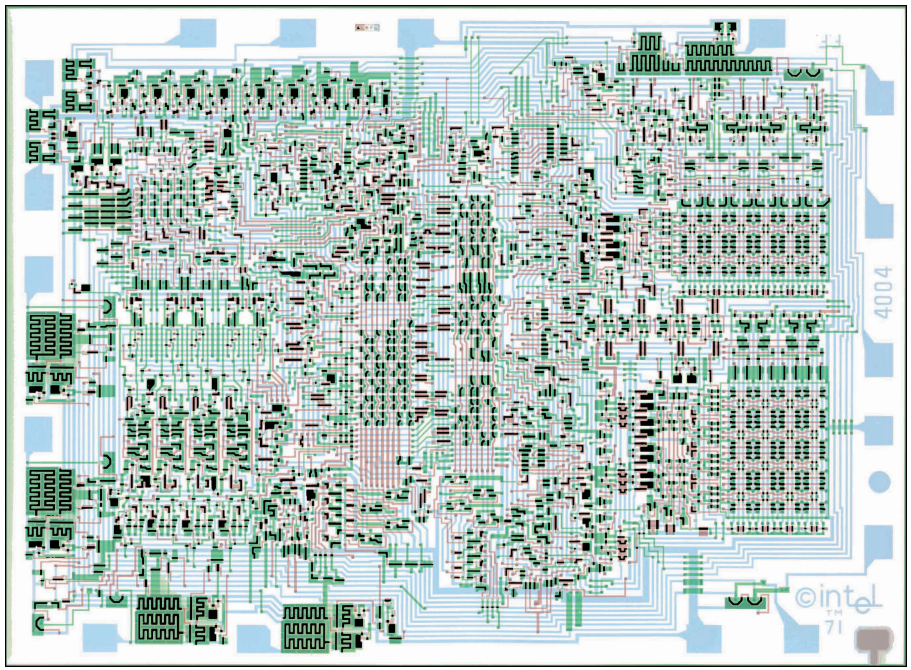
Subsection 1

L'enfance des microprocesseurs (décennie 70) : élargir le chemin des données (diminuer N)

Faggin (1941)

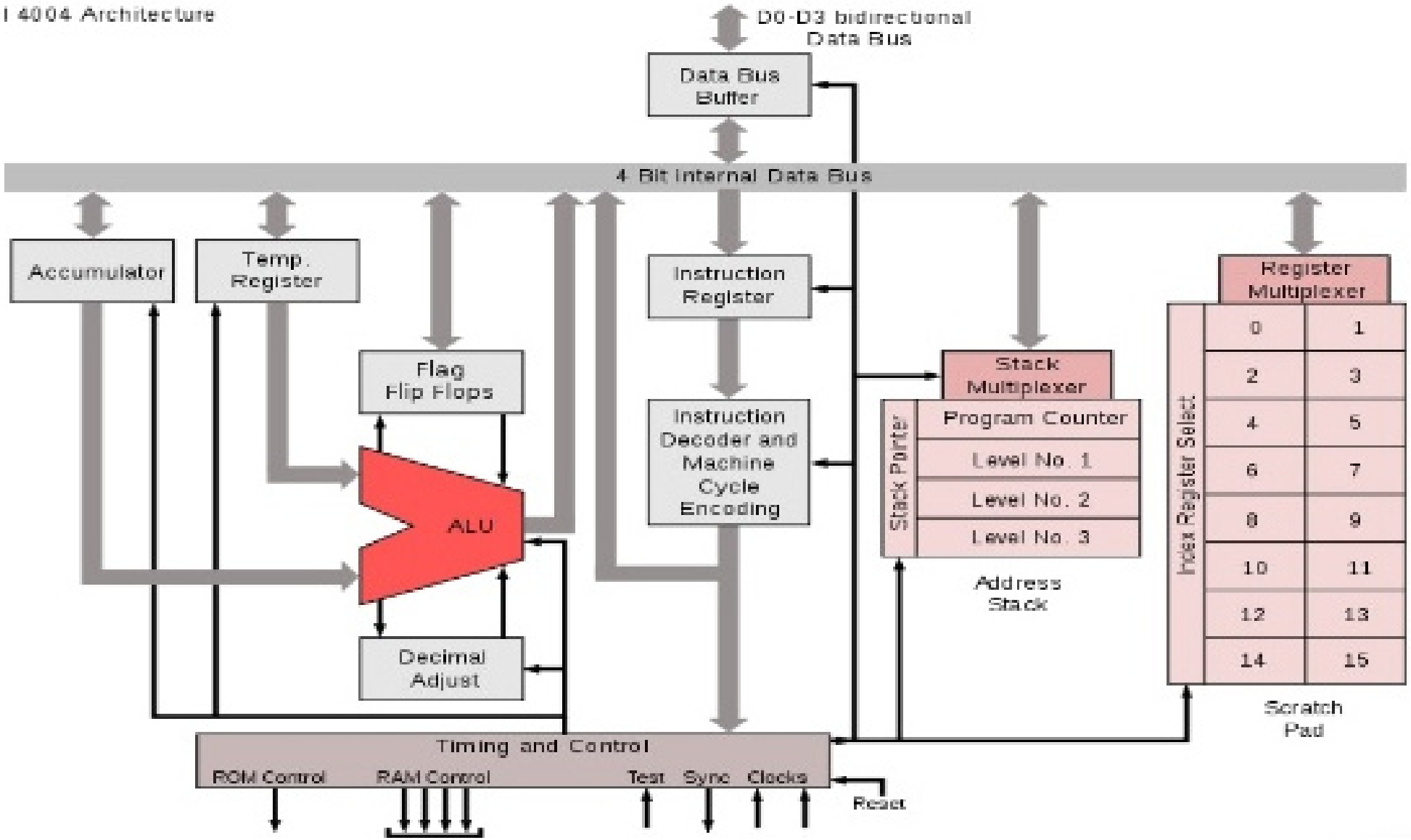
Inventeur du premier microprocesseur : Intel 4004 (1971)





Le 4004 est un microprocesseur 4 bits construit avec 2300 transistors.

La microarchitecture du 4004



Moore (1929)

Co-fondateur d'Intel (1968)

Loi de Moore initiale (1965) : The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.

Loi de Moore révisée (1975) : Semiconductor complexity will continue to double annually until about 1980 after which it will decrease to a rate of doubling approximately every two years.

Facteur 1000 en 20 ans, 1M en 40 ans. 1971 : 2300 transistors dans le 4004. 1993 : 3.1M transistors dans le Pentium (facteur 1300). 2011 : 2.3G transistors dans le Core i7-39xx.

Des microprocesseurs 4 bits aux microprocesseurs 64 bits

4-bit : Intel 4004 (1971) (2.3KT) (740Khz) (10μm)

8-bit : Intel 8008 (1972) (3.5KT) (500Khz) (10μm)

16-bit : Intel 8086 (1978) (29KT) (10Mhz) (3μm)

32-bit : Intel 80386 (1985) (275KT) (16Mhz) (1μm)

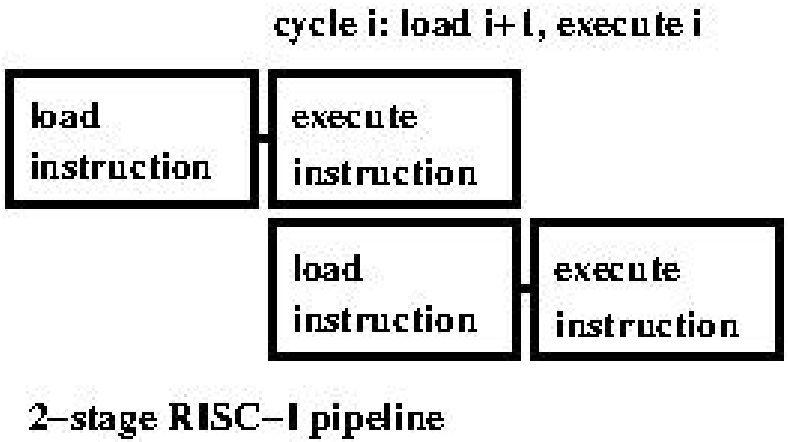
64-bit : MIPS R4000 (1991) (1.2MT) (100Mhz) (0.8μm)

Subsection 2

L'adolescence des microprocesseurs (décennie 80) : augmenter la fréquence

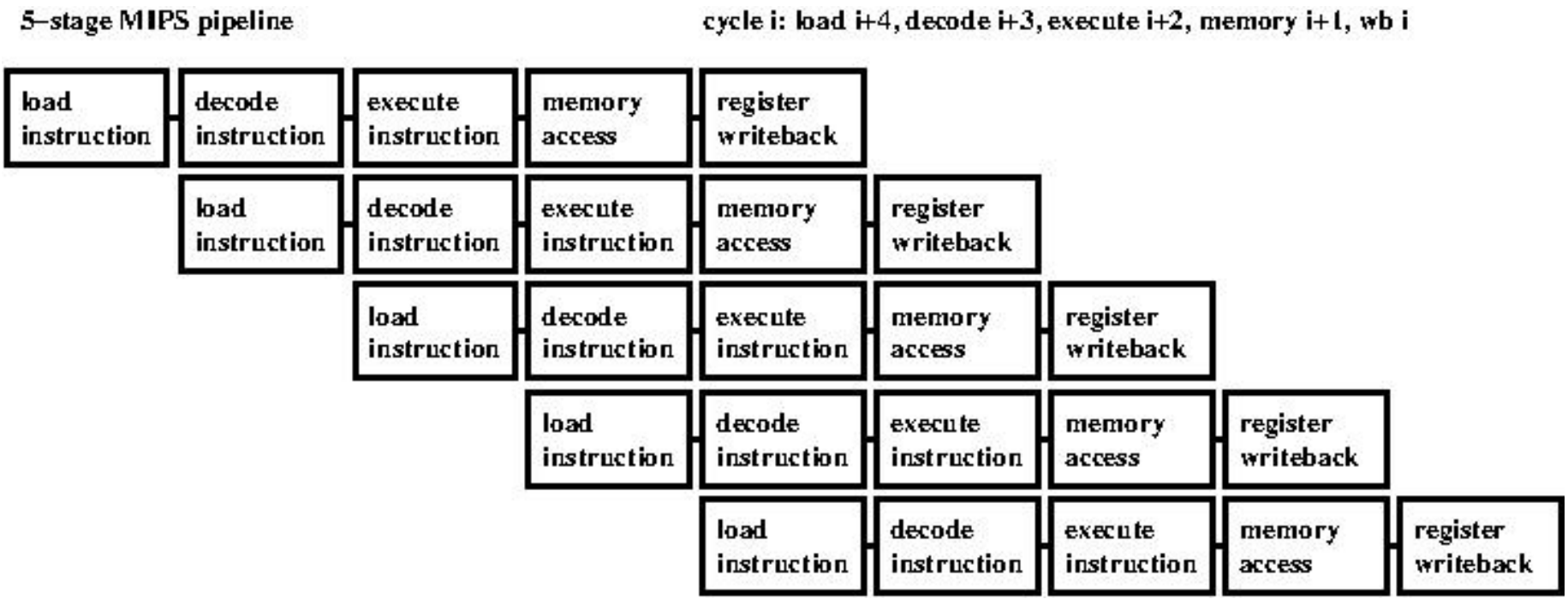
Un pipeline à deux étages : RISC-I (Patterson)

Pipeliner : augmenter F par la microarchitecture, augmenter I par la régularité (architecture chargement-rangement ou RISC)



Un pipeline à 5 étages : MIPS (Hennessy)

Plus d'étages = étages plus simples = horloge plus rapide



Patterson (1947), Séquin (1941), Hennessy (1952)

Reduced Instruction Set Computer : augmenter un peu N pour augmenter beaucoup I

RISC I : A Reduced Instruction Set VLSI Computer (Berkeley 1981)

RISC-I (1981) (44.5KT) (1Mhz) (5μm) (pipeline à 2 étages)

RISC-II (1983) (40.8KT) (3Mhz) (3μm) (pipeline à 3 étages)

MIPS R2000 (1986) (110KT) (15Mhz) (2μm) (pipeline à 5 étages)

MIPS R4000 (1991) (1.2MT) (100Mhz) (0.8μm) (pipeline à 8 étages)

Les freins architecturaux à la performance des pipelines

Les aléas empêchent de bien remplir le pipeline : le I crête est 1, le I réel est 0.5

Attente du PC suivant (sauts)

Attente d'instructions multi-cycles (multiplication, division)

Attente de chargement mémoire

Plus le pipeline est profond, plus les aléas sont nombreux. D'un côté, un pipeline plus profond augmente F, D'un autre côté, cela diminue I.

Subsection 3

L'âge adulte des microprocesseurs (décennie 90) : le parallélisme d'instructions

Dupliquer et vectoriser l'UAL, dupliquer le processeur

Principe du processeur superscalaire :
Avec deux UAL, on peut calculer deux opérations par cycle (on augmente I au-delà de 1)

Principe du processeur vectoriel :
Avec une UAL double, on peut calculer une opération sur deux données par cycle (on diminue N)

Principe du multi-processeur :
Avec deux processeurs, on peut exécuter deux instructions par cycle (on augmente I au-delà de 1)

Cray (1925-1996)

CDC 6600 : premier ordinateur superscalaire (1966)

Cray-I : premier ordinateur vectoriel (1976) (ancêtre des GPU)

Burroughs D825 : premier multi-processeur (1/4 processeurs) (1962)
Cray X/MP : multi-processeur vectoriel (2/4 processeurs) (1983)

Processeurs superscalaires

Pour augmenter I, plutôt que de chercher à remplir le pipeline, le dédoubler

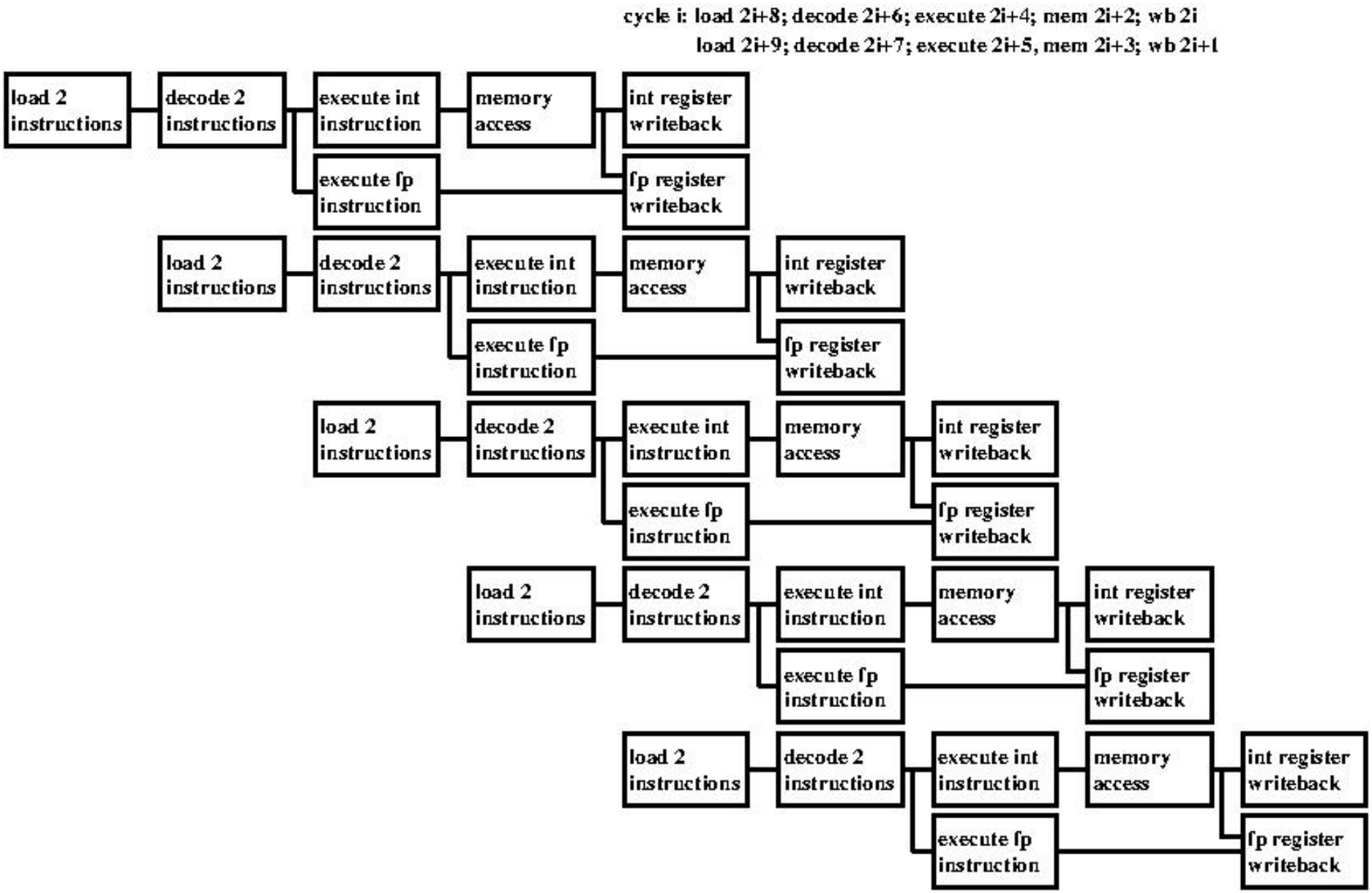
Première étape : deux pipelines (pipeline entier, pipeline flottant)

Motorola 88100 (1988), Intel i960 (1989) et AMD 29050 (1990) : premiers microprocesseurs superscalaires

Etape suivante : plusieurs pipelines entiers et flottants

Intel Pentium P5 (1993) : deux IU et deux FPU

Pipeline superscalaire



Les freins architecturaux à la performance des pipelines superscalaires

Un code purement entier n'utilise pas les pipelines flottants

Chargement des instructions complexe (plusieurs instructions, détection des sauts)

File de registres complexe (ports de lecture/écriture pour alimenter tous les pipelines)

Limite de l'ILP proche très basse (ILP=1.8) sans réordonnancement des instructions

Fréquence élevée des sauts (20%)

Fisher (1946)

Very Long Instruction Word architectures and the ELI-512 (1983)

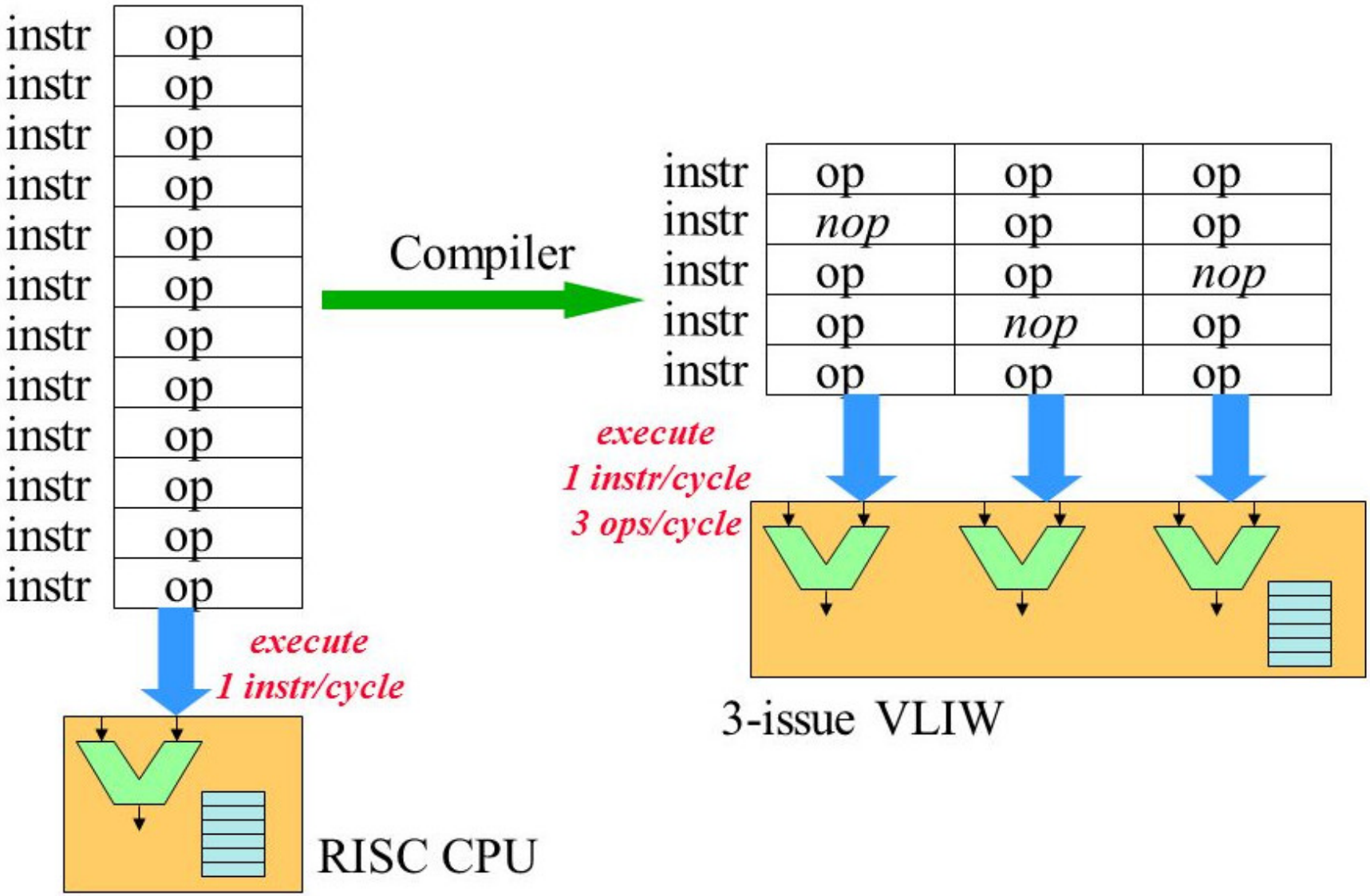
Inventeur des processeurs VLIW (1979)

Paquets de 7 instructions : 4 IU, 2 FPU, 1 branchement

Défi : remplir les paquets (trace scheduling compiler)

Obstacle : ILP proche bas même avec réordonnancement des instructions (ILP=6)

VLIW vs RISC



Le VLIW nécessite un assemblage statique des paquets.

Les registres vectoriels

Pour diminuer N, on vectorise les unités de calculs.

Registres XMM du Pentium III (1999) : 128 bits (4*32) (AVX)
Registres YMM du Sandy Bridge (2008) : 256 bits (8*32) (AVX2)
Registres ZMM du Knighs Landing (2016) : 512 bits (16*32) (AVX-512)

Tomasulo (1934-2008)

Algorithme de Tomasulo (1967)

Exécution en ordre partiel (*Out-of-Order*) implémentée dans l'IBM 360/91

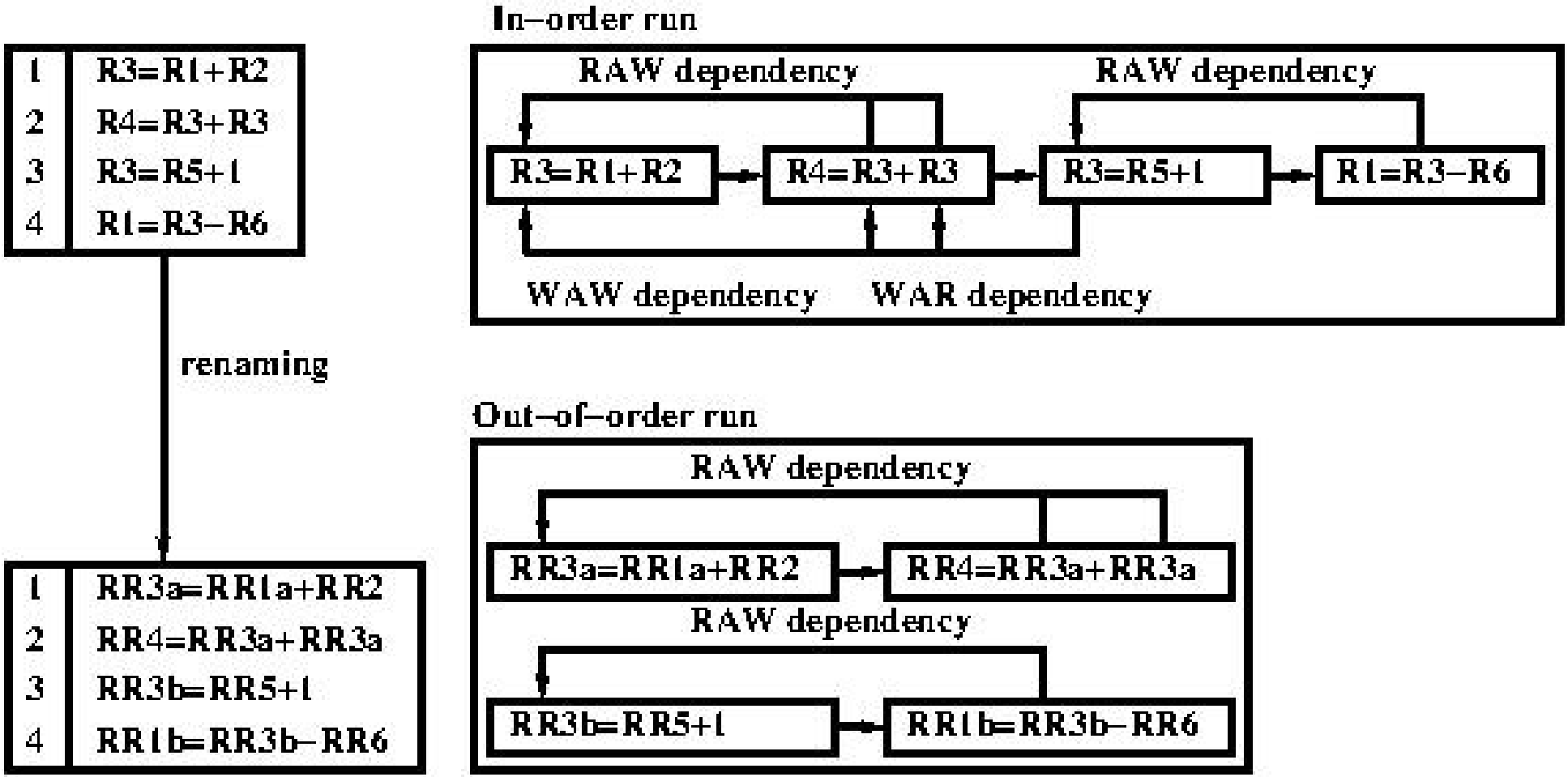
A l'origine, l'algorithme de Tomasulo servait à permettre l'exécution d'une opération pendant que la précédente était encore en cours

Redécouvert dans les années 90 par Yale Patt pour aider à remplir le pipeline

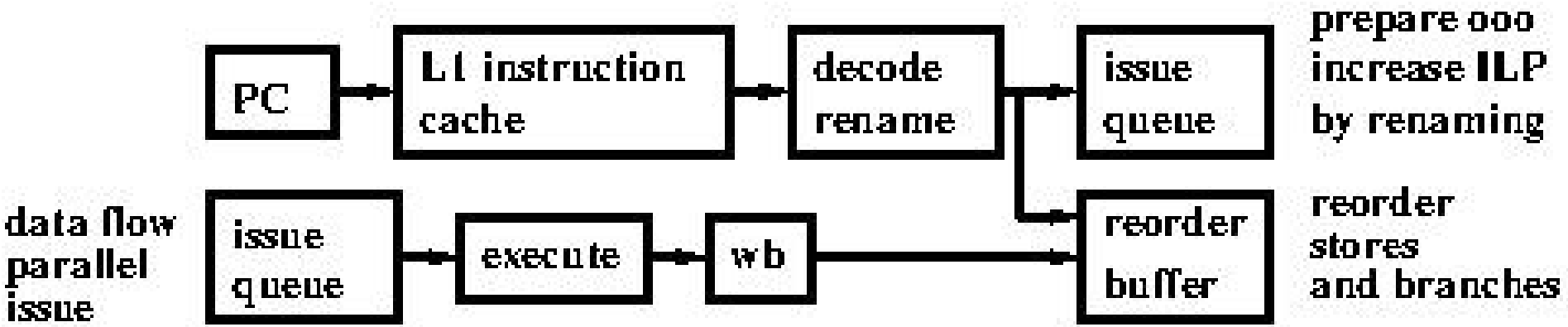
L'algorithme de Tomasulo est basé sur le renommage de registres qui élimine dynamiquement les fausses dépendances et augmente l'ILP local

L'exécution en ordre partiel est un contrôle par flot de donnée (Data flow)

Renommage et exécution en ordre partiel



Pipeline *Out-Of-Order*

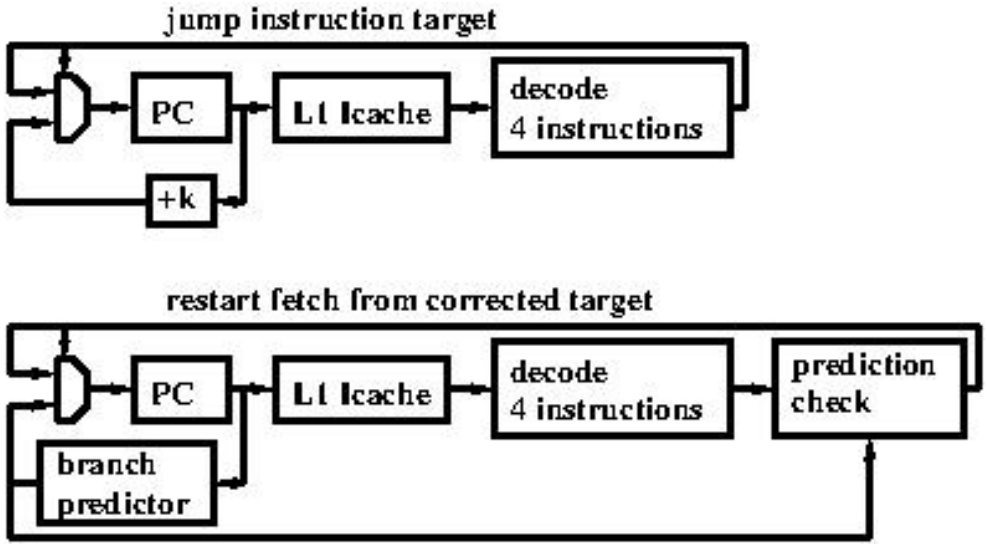


Patt (1939), Seznec (1959)

Patt : HPS, a new microarchitecture : rationale and introduction (1985) : exécution spéculative

Patt : Two-Level Adaptive Training Branch Prediction (1991) : prédicteurs de branchements

Seznec : TAGE-SC-L Branch Predictors (2014) : champion du monde des prédicteurs (2.5 MPKI *Miss Per Kilo Instructions*)

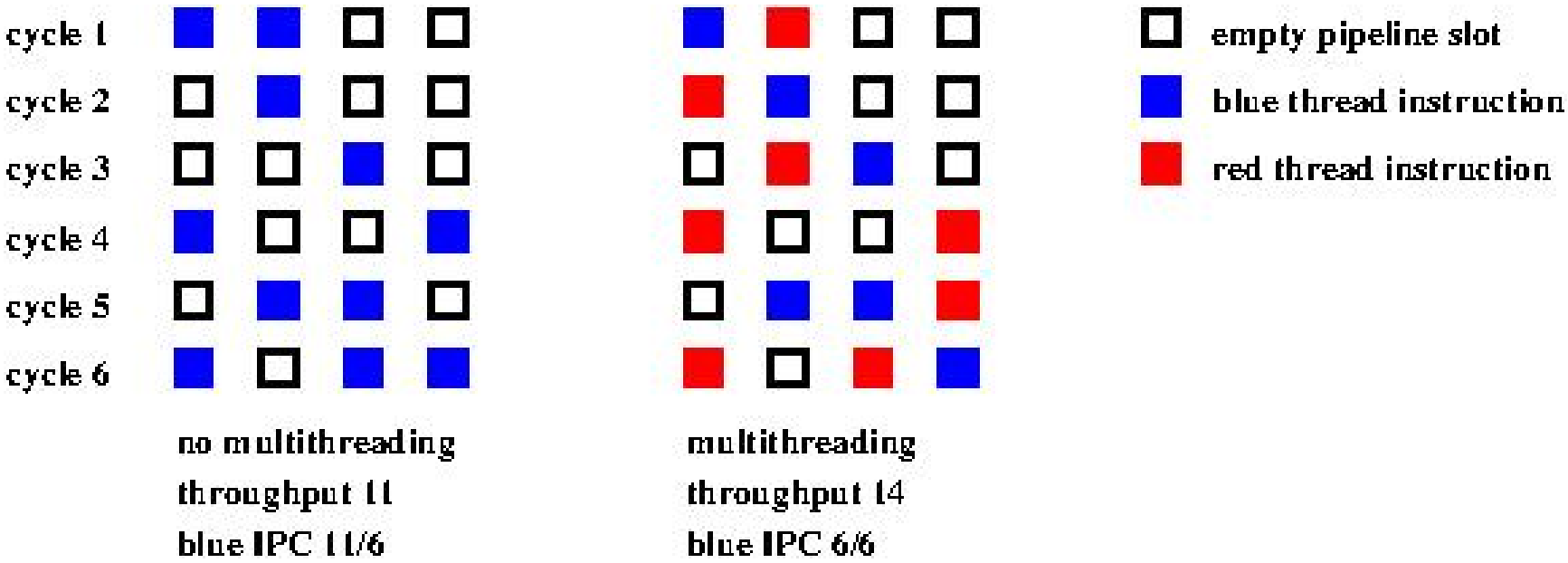


Tullsen (1961)

Simultaneous Multithreading : Maximizing On-chip Parallelism (1995) : processeurs *multithread*

Burton Smith HEP machine (1982)

Pour remplir le pipeline, entrelacer plusieurs *threads*. Cela augmente le débit mais aussi la latence de chaque *thread*.



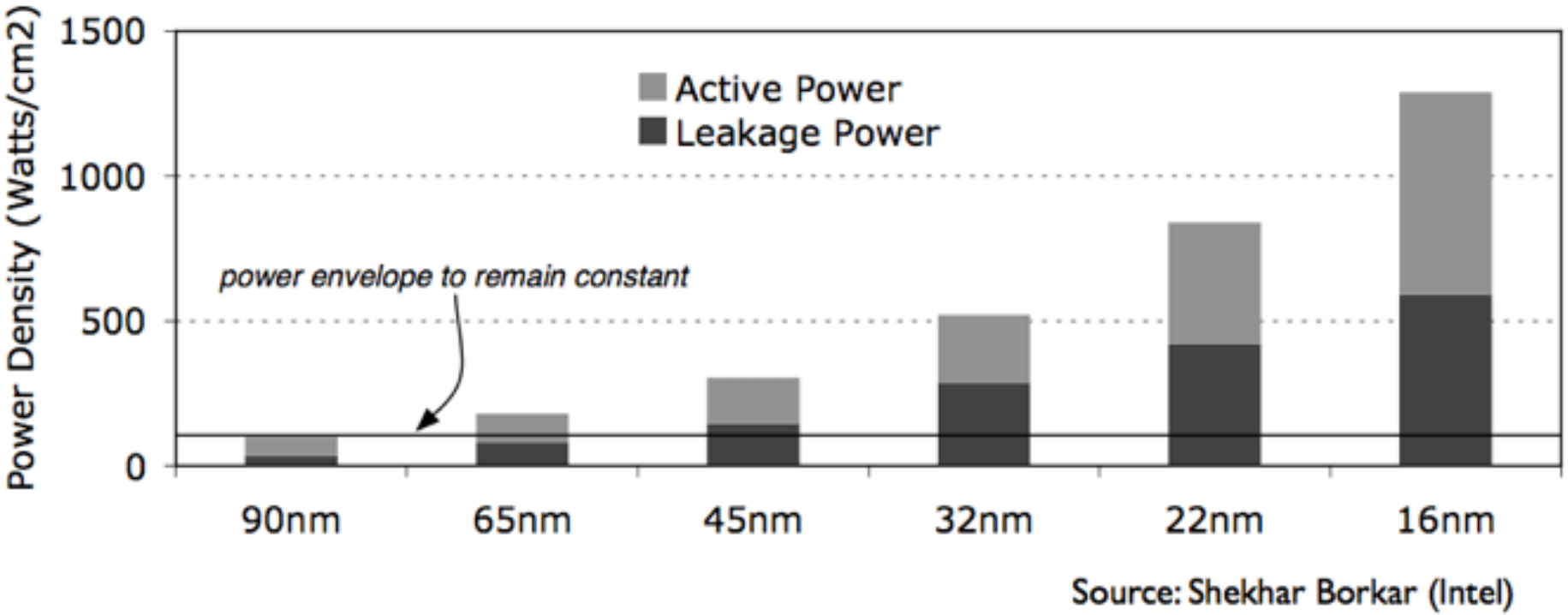
Subsection 4

La vieillesse des microprocesseurs (décennie 00) : augmenter le nombre de cœurs

La fin de l'accroissement de la fréquence

F n'augmente plus après 2002 (Pentium 4)

La taille des transistors diminue, mais la puissance nécessaire pour les alimenter tous augmente, ce qui pose des problèmes de refroidissement



$$P_d = V_{dd} * I_{leak} + \alpha * \sum C_i * V_{dd}^2 * F$$

La fin de l'accroissement de l'IPC

Amélioration très lente depuis 1990 (de 2 IPC crête à 4 IPC crête)

Plus d'amélioration de I après 2011 (Sandy Bridge) (4 IPC crête)

L'ILP croît très lentement avec la taille de la fenêtre

Un processeur avec un seul PC et un prédicteur de sauts emmagasine au plus 400 instructions, donc l'ILP reste faible

Que faire des transistors ?

La loi de Moore continue : 2012 - 5GT (Xeon Phi), 2015 - 10GT (Sparc M7), 2017 - 19GT (AMD EPYC)

Depuis 2001 (IBM Power4), on augmente N artificiellement en dupliquant les cœurs.

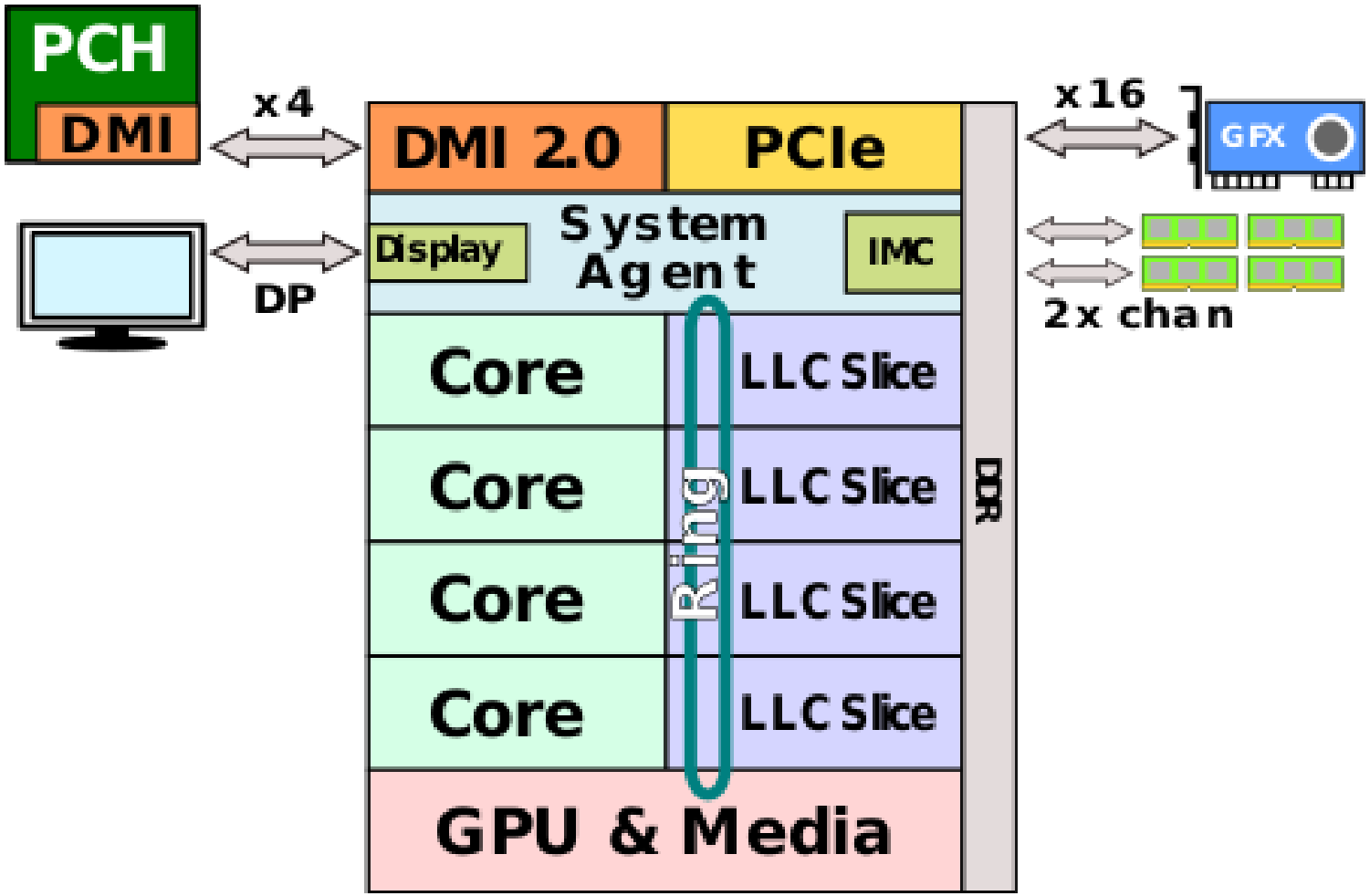
Augmenter le nombre de cœurs augmente le débit sans améliorer la latence des applications, sauf si elles sont parallélisées

Intel Pentium D (2005) : 2 cœurs
Intel Xeon Clovertown (2006) : 4 cœurs
Intel Xeon 7000 (2010) : 8 cœurs
Intel Xeon E7 (2011) : 10 cœurs
Intel Xeon E5-v2 (2013) : 12 cœurs
Intel Xeon E5-v3 (2014) : 18 cœurs
Intel Xeon E5-v4 (2016) : 24 cœurs
AMD EPYC (2017) : 32 cœurs

Loi de Moore : 2 cœurs en 2001 = 256 cœurs en 2015 et 1024 cœurs en 2019

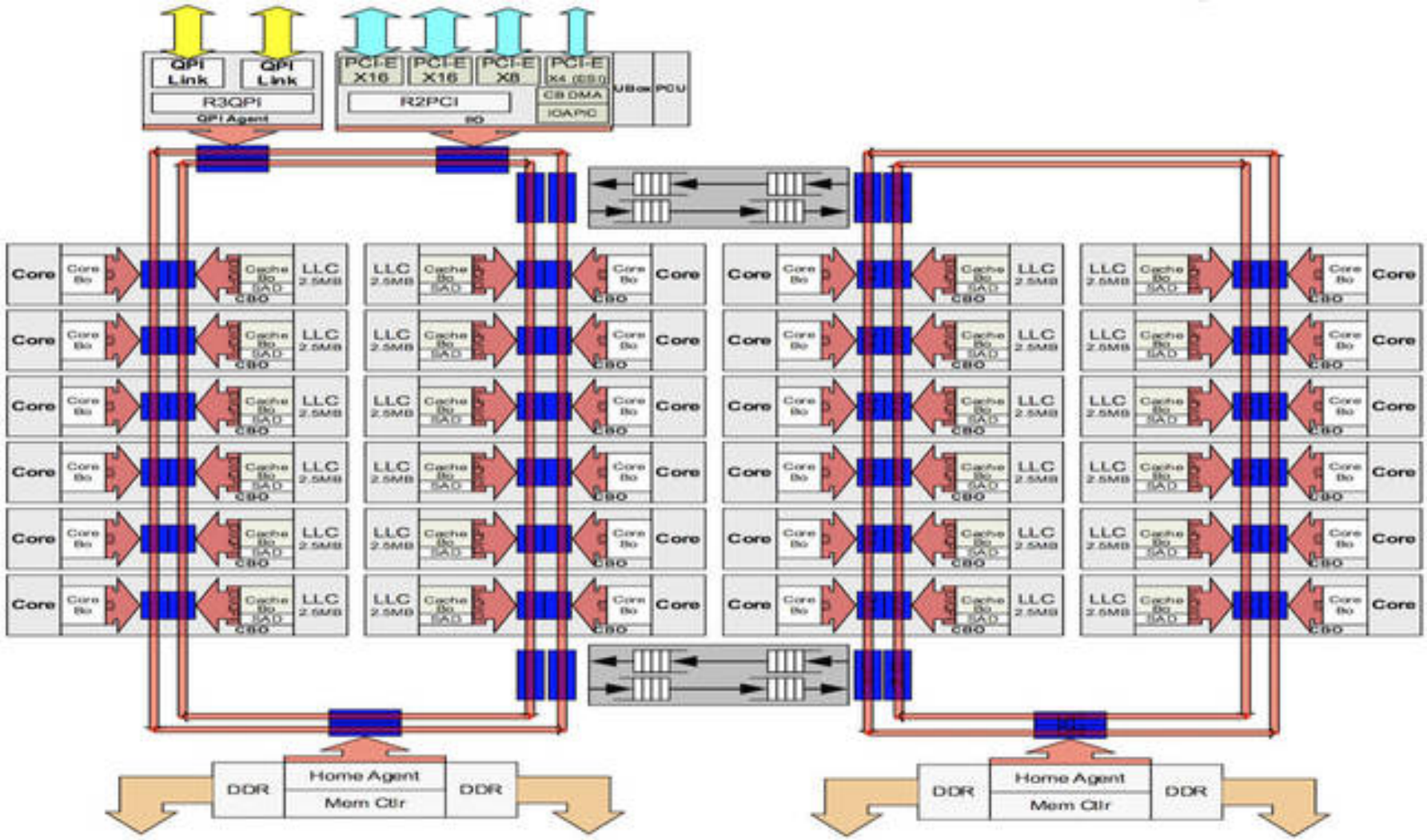
Interconnecter les cœurs

Sandy-bridge (jusqu'à 6 cœurs)



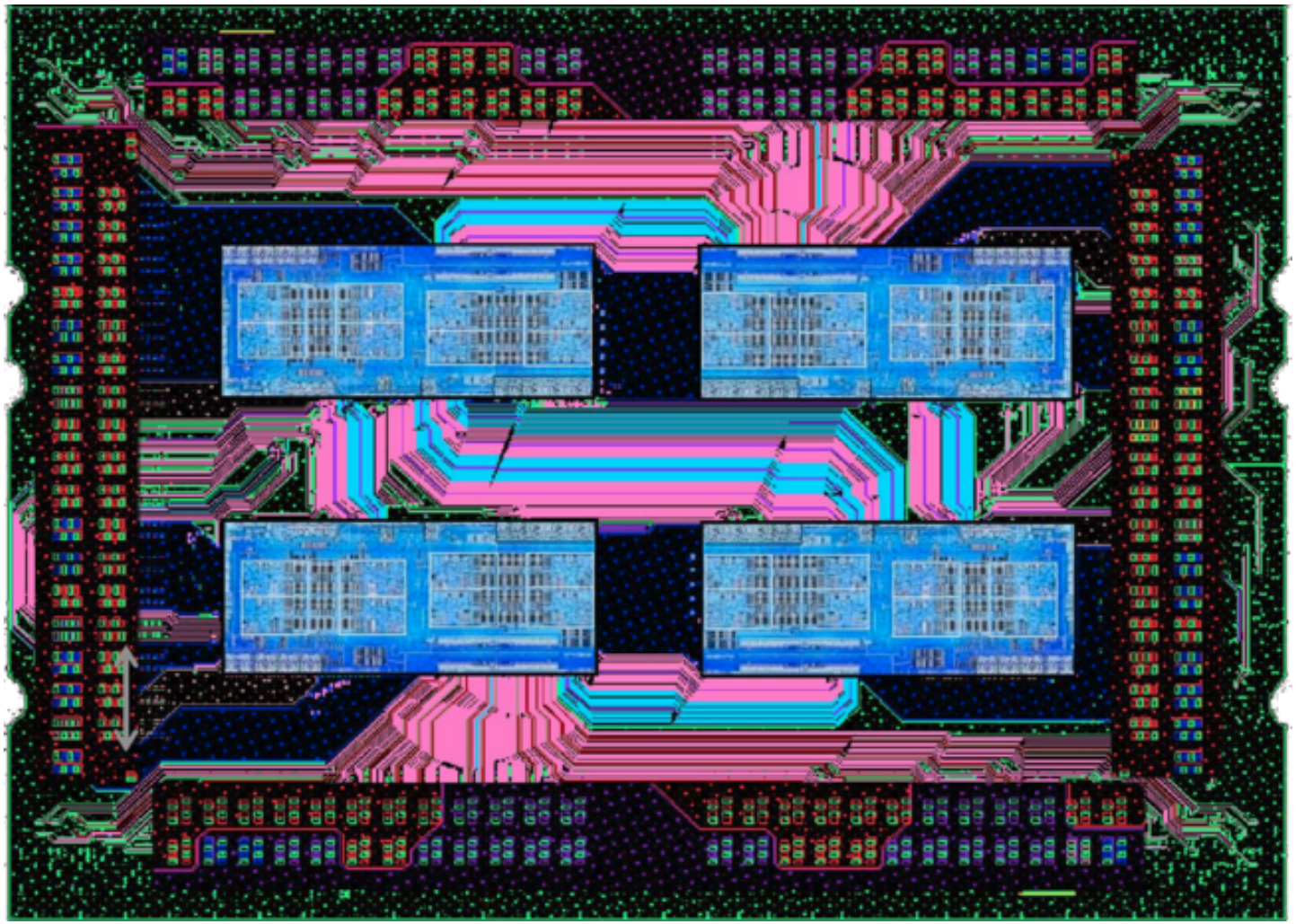
Interconnecter les cœurs

Broadwell Xeon E5-v4 (jusqu'à 24 cœurs)



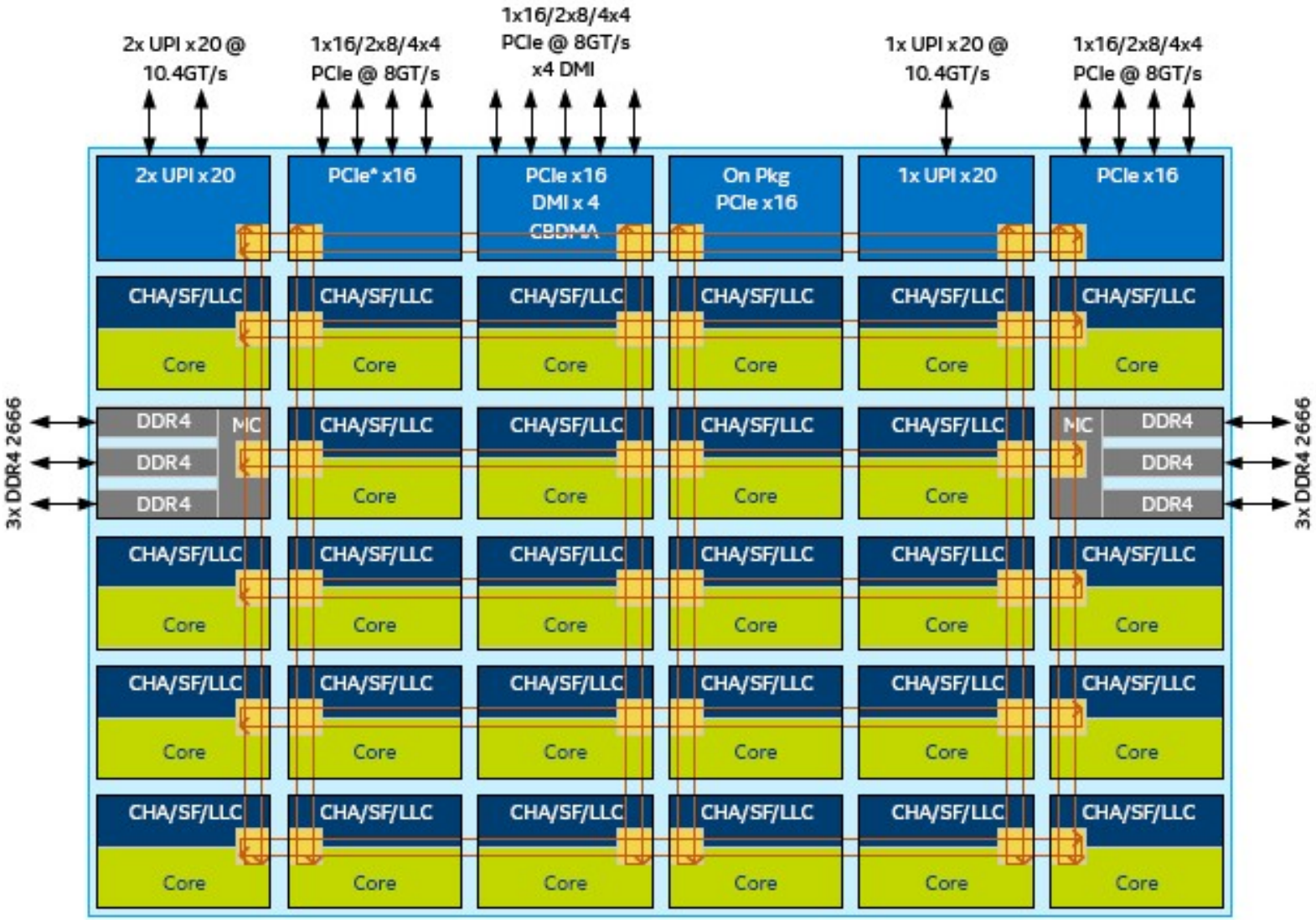
Interconnecter les cœurs

AMD Epyc (jusqu'à 32 cœurs)



Interconnecter les cœurs

Xeon Phi2 (jusqu'à 72 cœurs)



Section 4

Conclusion

Conclusion pour les économistes

L'industrie des ordinateurs a eu une croissance exponentielle prédite par la loi de Moore

Qu'adviendra-t-il aux constructeurs quand la croissance sera linéaire ?

Conclusion pour les physiciens

Un ordinateur est un dispositif électronique basé sur le concept de commutateur : contrôler le flot d'électrons

Le commutateur a été implémenté avec une triode puis un transistor

En réduisant la taille des transistors, on a réduit progressivement le nombre d'électrons impliqués dans chaque commutation

Que fera-t-on quand il n'y aura plus que quelques électrons transitant par chaque transistor ?

Conclusion pour les mathématiciens

Un ordinateur est une machine pour calculer un algorithme

Les ordinateurs peuvent changer dans leur conception mais le calcul d'un algorithme reste le même, c-à-d la machine de Turing

Conclusion pour les informaticiens

Avec la multiplication des cœurs, le défi est de faire progresser la performance (pour un programme) de façon proportionnelle

Il y a deux objectifs contradictoires : augmenter le nombre de cœurs et diminuer la latence des communications inter-cœurs

La seule solution est de s'appuyer sur des algorithmes réduisant le nombre de communications et favorisant la proximité des consommateurs et de leurs producteurs