

cours d'architecture

but : répondre à la question

“comment fonctionne un ordinateur?”

40 heures de cours

15 heures de TD

cours disponible à

<http://www.blois.univ-tours.fr/~marcel/>

plan

1. introduction
2. pré-requis : codage et algèbre de Boole
3. arithmétique et circuits de calcul
4. mémoire
5. bus
6. jeux d'instructions
7. CPU
8. unité de commande

cours 1

introduction

un premier aperçu du programme

plan de ce cours

- introduction

plan de ce cours

- introduction
- historique

plan de ce cours

- introduction
- historique
- machine Von Neumann

plan de ce cours

- introduction
- historique
- machine Von Neumann
- aperçu

plan de ce cours

- introduction
- historique
- machine Von Neumann
- aperçu
- conclusion

la machine perçue par l'utilisateur

interactions avec la machine au travers des périphériques

- clavier,
- souris,
- écran,
- imprimante,
- disquette, etc...

la machine perçue par le programmeur

architecture = attributs visibles par le programmeur

- mécanisme d'entrées-sorties
- nombre de bits utilisés pour un type de donnée
- ...

exemple : un instruction de multiplication est elle disponible ?

la machine invisible

organisation = comment ces attributs sont implantés

- technologie de la mémoire
- signaux de contrôle
- ...

exemple : la multiplication est elle implantée directement ou par une suite d'addition ?

architecture et organisation

toute la famille x86 d'Intel partage la même architecture de base

toute la famille System/370 d'IBM partage la même architecture de base

mais l'organisation est différente d'une version sur l'autre

la machine invisible

l'unité centrale contient 3 unités fonctionnelles

l'automate donne les ordres

la partie calcul exécute les ordres

la mémoire stocke les ordres et les données

la machine invisible

l'unité centrale contient 3 unités fonctionnelles

l'automate donne les ordres

la partie calcul exécute les ordres

la mémoire stocke les ordres et les données

une interaction se traduit par l'exécution d'une séquence d'opérations par ces unités

fonctions de base

4 fonctions de base

- traitement des données
- mémorisation des données
- transfert des données
- contrôle

historique

- avant le 19^e siècle : automates spécialisés
 - boîte à musique
 - métiers à tisser
 - ...

historique

- avant le 19^e siècle : automates spécialisés
 - boîte à musique
 - métiers à tisser
 - ...
- 19^e siècle : automate et machine à calcul
 - Charles Babbage décrit un calculateur pouvant
 - répéter des séquences d'opérations
 - choisir en fonction de l'état du calcul

historique

20ème siècle

J. Von Neumann (1946) décrit un modèle de machines universelles

les machines actuelles

- s'appuient sur ce modèle
- sont classées en 5 générations

générations

| génération | date approximative | technologie | vitesse (opérations/s) |
|------------|-----------------------|-------------|---------------------------|
| 1 | 1946-1957 | tube à vide | 40 000 |
| 2 | 1958-1964 | transistor | 200 000 |
| 3 | 1965-1971 | SSI/MSI | 1 000 000 |
| 4 | 1972-1977 | LSI | 10 000 000 |
| 5 | 1978- | VLSI | 100 000 000 |

1945-1958

- ordinateurs spécialisés, exemplaires uniques
- machines volumineuses et peu fiables
- technologie à lampes, relais, résistances
- 10^4 éléments logiques
- programmation par cartes perforées

représentant : ENIAC (1946) pour l'étude de faisabilité de la bombe H

1958-1964

- usage général, machines fiables
- technologie à transistors
- 10^5 éléments logiques
- premiers langages de programmation évolués (COBOL, FORTRAN, LISP)

représentant : mini ordinateur DEC PDP-1 (1961)

1965-1971

- technologie des circuits intégrés
(S/MSI small/medium scale integration)
- 10^6 éléments logiques
- avènement du système d'exploitation complexe
- UNIX, Pascal, Basic,
- CISC

loi de Moore : le nombre de transistors intégrables sur une seule puce double chaque année

loi de Moore

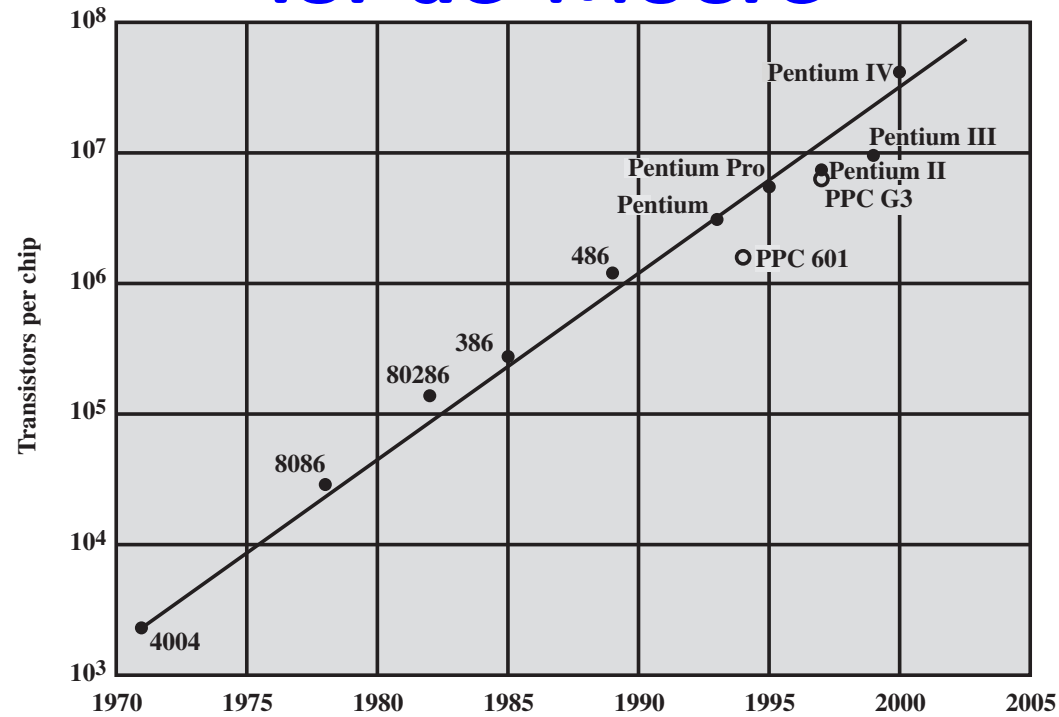


Figure 2.8 Growth in CPU Transistor Count

1972-1977

- technologie LSI (large SI)
- 10^7 éléments logiques
- avènement de réseaux de machines
- traitement distribué/réparti

1971 : premier microprocesseur 4004 de INTEL

toutes les composantes de la CPU sont réunies sur une même puce

après

- technologie VL/WSI (very large, wafer)
- $> 10^8$ éléments logiques
- systèmes distribués interactif
- multimédia, traitement de données non numériques (textes, images, paroles)
- parallélisme massif, client-serveur
- RISC

Evolution

quelques caractéristiques des séries IBM700/7000,
PDP-8, et INTEL

Machine Von Neumann

modèle de machine universelle possédant

une **mémoire**

contient instructions
et données

une **unité arithmétique
et logique (ALU)**

effectue les calculs

une **unité d'entrées/sorties
(I/O)**

échange d'informations
avec les périphériques

**unité de commande
(UC)**

contrôle

fonctionnement schématique

l'UC

1. extrait une instruction de la mémoire
2. analyse l'instruction
3. recherche dans la mémoire les données concernées
4. déclenche l'opération adéquate sur l'ALU ou l'E/S
5. range le résultat dans la mémoire

Machine Von Neumann

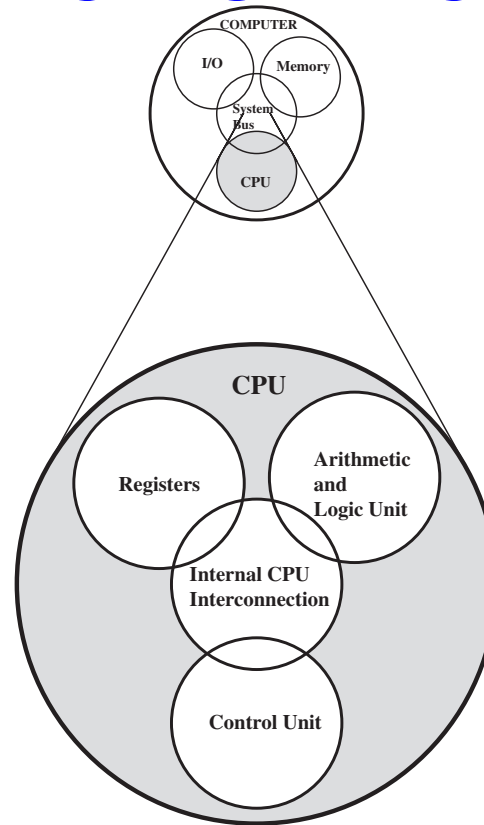


Figure 1.5 The Central Processing Unit (CPU)

Aperçu

dispositifs de base

- signal et chronogramme
- horloge
- registres
- bus

unités fonctionnelles

- mémoire
- ALU
- E/S
- UC

signal et chronogramme

un **signal** est une grandeur discrète appartenant à $[0,1]$

signal et chronogramme

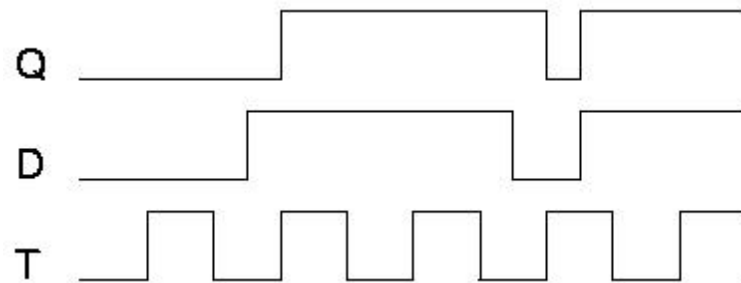
un **signal** est une grandeur discrète appartenant à $[0,1]$

un **chronogramme** est la représentation graphique d'un signal évoluant dans le temps

signal et chronogramme

un **signal** est une grandeur discrète appartenant à $[0,1]$

un **chronogramme** est la représentation graphique d'un signal évoluant dans le temps



Horloge

signal de séquences régulières de 0 et de 1

une séquence est appelée **cycle**

exemple : une fréquence d'horloge de 500MHz donne des cycles de 2 nanosecondes

synchronise l'ensemble des dispositifs

Registres

éléments de mémoire rapide internes à la CPU

un signal commande la mémorisation

- chargement sur niveau
- chargement sur front

Bus

ensemble de fils électriques sur lesquels transitent les signaux

relie les unités entre elles

largeur du bus

- nombre de fils constituant le chemin
- nombre de signaux pouvant être envoyés en même temps

Mémoire

vecteur dont chaque composante est accessible par une
adresse

opérations permises

- lecture
- écriture

mot = unité d'information accessible en une seule
opération de lecture

fonctionnement

1. l'UC inscrit l'adresse d'une cellule dans un registre d'adresse (RA)
2. l'UC demande une opération
3. les échanges se font via un registre de mot (RM)

fonctionnement

lecture

$RA \leftarrow \text{adresse}$

$RM \leftarrow \text{mémoire}[RA]$

fonctionnement

lecture

$RA \leftarrow \text{adresse}$

$RM \leftarrow \text{mémoire}[RA]$

écriture

$RM \leftarrow \text{valeur}$

$RA \leftarrow \text{adresse}$

$\text{mémoire}[RA] \leftarrow RM$

ALU

fonction à 3 paramètres

- 1 opération
- 2 arguments
- 1 résultat

nécessite un/des registre(s) de mémorisation

- entrées, sortie
- résultats intermédiaires

E/S

sert d'interface avec les périphériques

les opérations associées sont fonction du périphérique

fonctionnement similaire à la mémoire

– registre mémorisant l'adresse du périphérique

Registre de Sélection du Périphérique

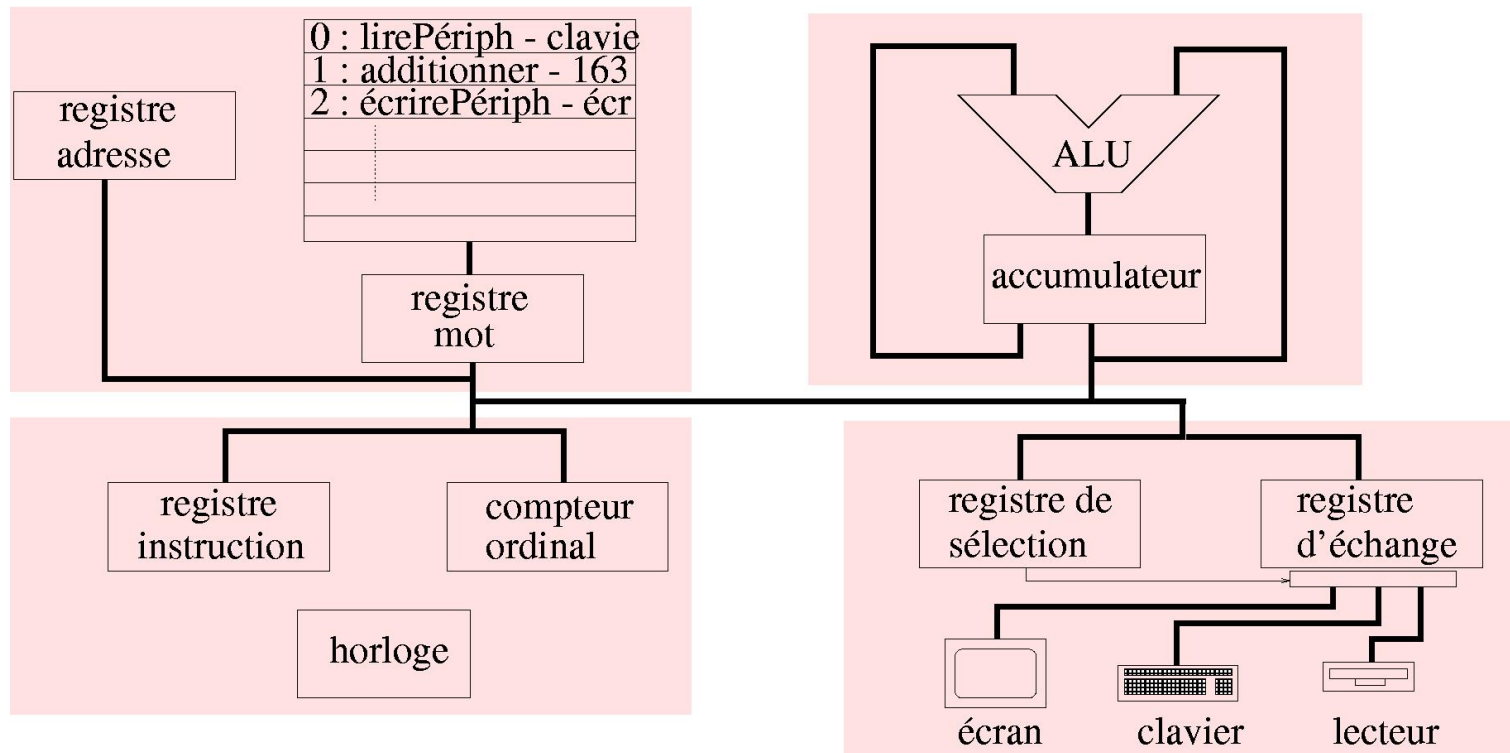
– un Registre d'Echange de données

Unité de commandes

son **fonctionnement** est celui du modèle Von Neumann associée aux registres suivants :

- *Compteur ordinal (PC)*
adresse mémoire de l'instruction à exécuter
- *Registre d'instruction (RI)*
instruction découpée en différentes parties

la machine complète



jeu d'instructions

ensemble des instructions exécutables sur une machine
différents **formats d'instruction** suivant le nombre de
parties réservées aux opérandes (ou adresses)

| | | | |
|-----------|------------|------------|--------------------|
| opération | opérande | | (format 1 adresse) |
| opération | opérande 1 | opérande 2 | (format 2 adresse) |
| ... | | | |

exemple d'instructions au format 1 adresse

| | |
|--------------|-----------|
| lirePériph | nomPériph |
| écrirePériph | nomPériph |
| chargerAcc | adMémoire |
| mémoAcc | adMémoire |
| additionner | adMémoire |
| soustraire | adMémoire |
| multiplier | adMémoire |
| diviser | adMémoire |
| testZéro | adMémoire |
| stop | |

exemple

horloge à 4 phases (1 par cycle de l'UC)

- acquisition au clavier lirePériph clavier
- addition de la valeur lue
 avec une donnée en mémoire additionner 163
- affichage du résultat à l'écran écrirePériph écran

$$H_0 : RA \leftarrow PC$$

$$H_1 : RI \leftarrow RM, PC \leftarrow PC + 1$$

$$H_2 : RSP \leftarrow RI_p$$

$$H_3 : ACC \leftarrow RE$$

$$H_0 : RA \leftarrow PC$$

$$H_1 : RI \leftarrow RM, PC \leftarrow PC + 1$$

$$H_2 : RA \leftarrow RI_p$$

$$H_3 : ACC \leftarrow ACC + RM$$

$$H_0 : RA \leftarrow PC$$

$$H_1 : RI \leftarrow RM, PC \leftarrow PC + 1$$

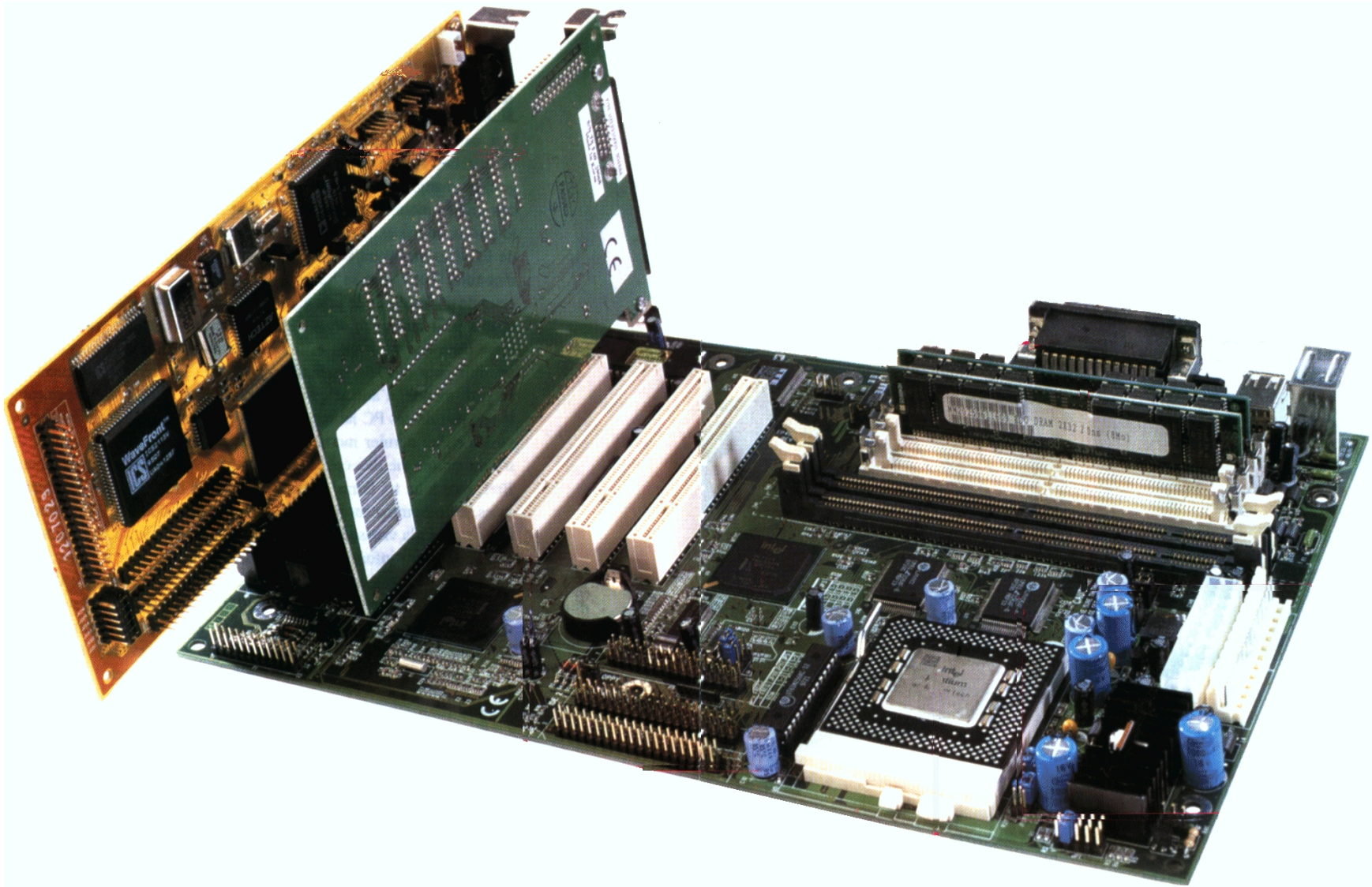
$$H_2 : RSP \leftarrow RI_p$$

$$H_3 : RE \leftarrow ACC$$

chargement

chargeur = programme qui lit un programme et l'écrit en mémoire

le chargeur est chargé par un dispositif matériel appelé microchargeur



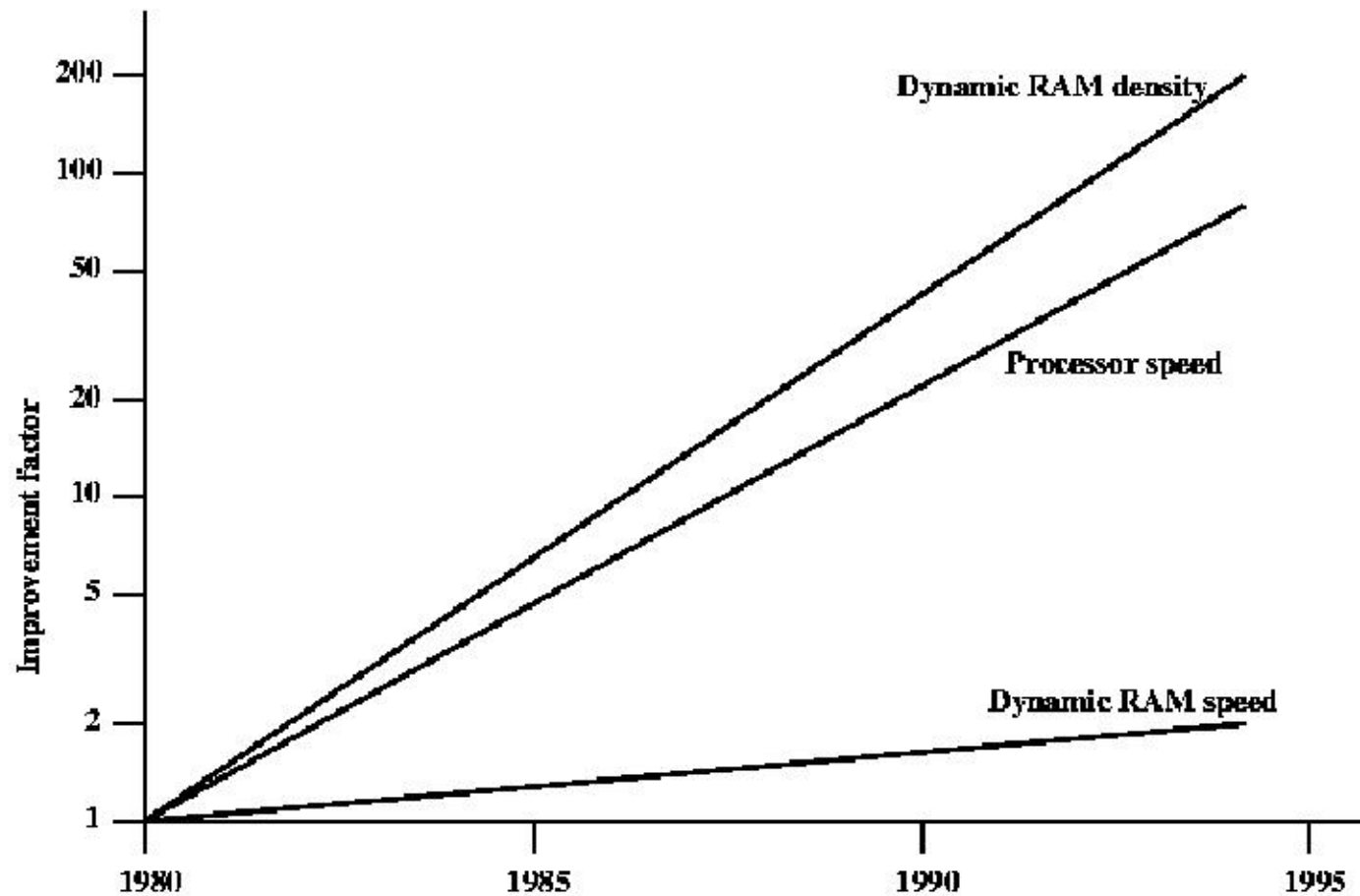
Conclusion

l'évolution des ordinateurs est liée à

- l'évolution des besoins (i.e., des application)
- l'évolution des performances de chaque composant

le principe de conception d'une machine est lié à la performance

évolution des composants



solutions

- augmenter le nombre de bits manipulés simultanément
- changer la structure de la mémoire
- réduire la fréquence des accès mémoire
- augmenter la bande passante

mesures de performance

la performance dépend de l'architecture

mesurer

la fréquence d'horloge?

objectif pour une même architecture

nombre d'instruction /s?
(MIPS, MFLOPS)

objectif pour un même jeu d'instructions

temps d'exécution des programmes (benchmarks)?

le plus objectif

prochain chapitre

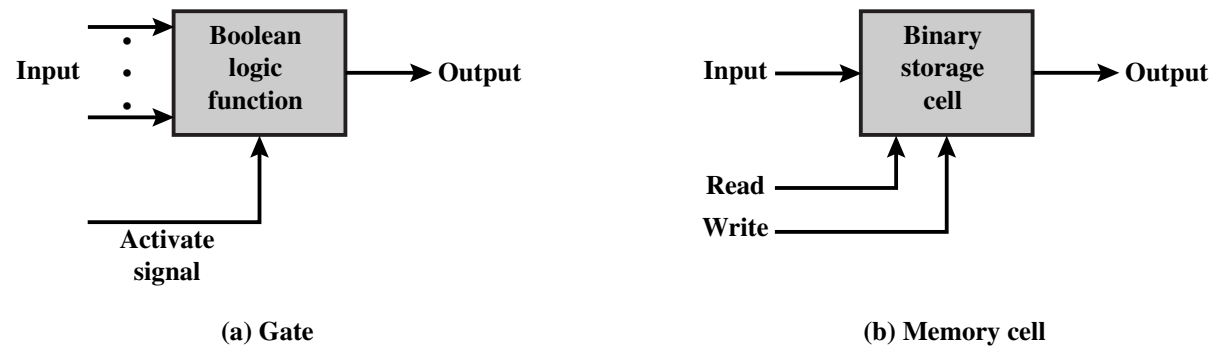


Figure 2.6 Fundamental Computer Elements