

# Architecture et Système

Stefan Schwoon

Cours L3, 2018/19, ENS Cachan

# Organisation

---

Cours: Lundi 13:45–15:45, C315 (Stefan Schwoon)

TP: Vendredi 13–16, C411 (François Thiré)

Contrôles de connaissances : Projet + Examen (50/50)

Transparents etc : <http://www.lsv.fr/~schwoon/>

E-mail: [schwoon@lsv.fr](mailto:schwoon@lsv.fr)

Bureau: RH-B-113 (LSV)

# Contenu du cours

---

## Contenu

Comment fonctionne un ordinateur ? Que se passe-t-il à l'intérieur ?

Aspects matériel (architecture) et logiciel (système)

**Exemples (matériel)** : Circuits, assembler, représentation de données, mémoire, interruptions, ...

**Exemples (logiciel)** : shell, programmation système, système fichiers, réseau, ...

# Contenu du cours

---

## Objectifs:

Connaissances pratiques pour vos tâches quotidiennes, en expérimentation, programmation, ...

Systeme d'exploitation : Unix (standard POSIX)

Langages utilisés : assembleur (un petit peu) et surtout C

# Littérature

---

## Architecture :

John P. Hayes, *Computer Architecture and Organization*, McGraw Hill (3rd edition)

## Système d'exploitation :

Andrew S. Tanenbaum, *Operating systems*, Prentice Hall

# Contenu d'aujourd'hui

---

Histoire abrégée (et incomplète) des ordinateurs

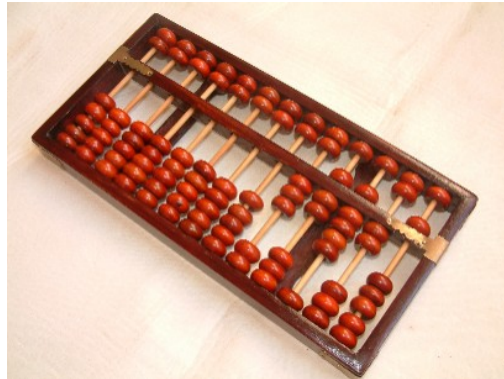
Développement de l'architecture

Un peu de programmation historique

# Les premiers ordinateurs

---

Dans un sens moins strict : **abaque** ou **règle à calcul**



(Source du mot **calcul**: latin *calculus* = caillou)

facilitent le calcul sans le moindre automatisme – c'est l'homme qui exécute un algorithme donné

utilisation des tableaux calculés en avance, p.ex. pour les logarithmes

# Les machines de calcul mécaniques

---

Au 17e siècle : Wilhelm Schickard / Blaise Pascal / Gottfried Leibniz

Machines mécaniques, basées sur le système décimal

Pascaline : addition, subtraction (par complément)

Machine de Leibniz : multiplication en plus (grandes problèmes mécaniques)

motivation : astronomique (S.), calcul financier, impôts (P.), philosophique (L.)

production en masse : à partir du 19e siècle (Thomas de Colmar)



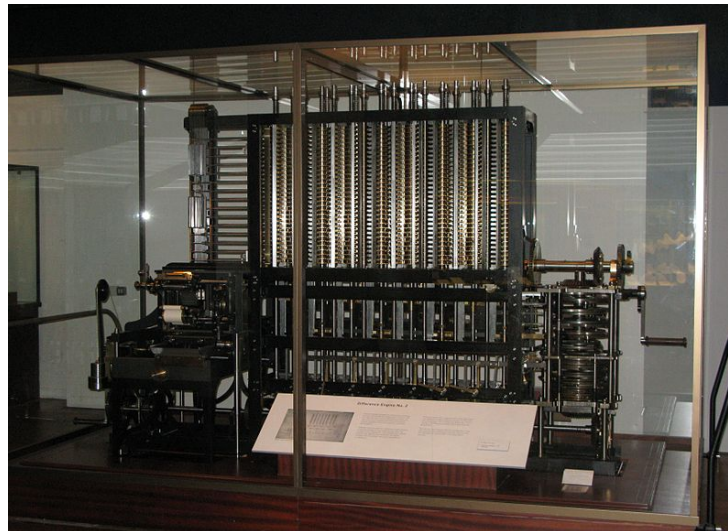
# Première machine dite programmable

---

**Difference Engine** (1822-1832) de Charles Babbage

Une opération (l'addition) appliquée simultanément à plusieurs **registres**

“Programmation” consiste en déterminant les valeurs initiales



# Fonctionnement de la Difference Engine

---

Programmation de la *Difference Engine* pour calculer les carrées :

pour  $n$  registres, fixer les valeurs initiales  $x_1^{(0)}, \dots, x_n^{(0)}$ ;

appliquer la règle  $x_{i+1}^{(k+1)} := x_i^{(k)} + x_{i+1}^{(k)}$ , pour  $1 \leq i < n$  et  $k \geq 0$

La machine réellement construit avait 3 registres à 6 chiffres.

Motivation : calcul automatique des tableaux mathématiques

Problèmes mécaniques empêchent une construction efficace

Innovations : mémoire, itération

# Machine théorique conçue par Babbage

---

**Analytical Engine** (jamais terminé)

Opérations multiples (addition/multiplication/. . .)

**Séquence** d'opérations programmable avec branchement conditionnel

# Fin 19e/début 20e siècle

---

Améliorations mécaniques et en utilisabilité

Des machines construites pour un but précis :

*calculatrices* utilisés par les ingénieurs, navigateurs, militaires, ...

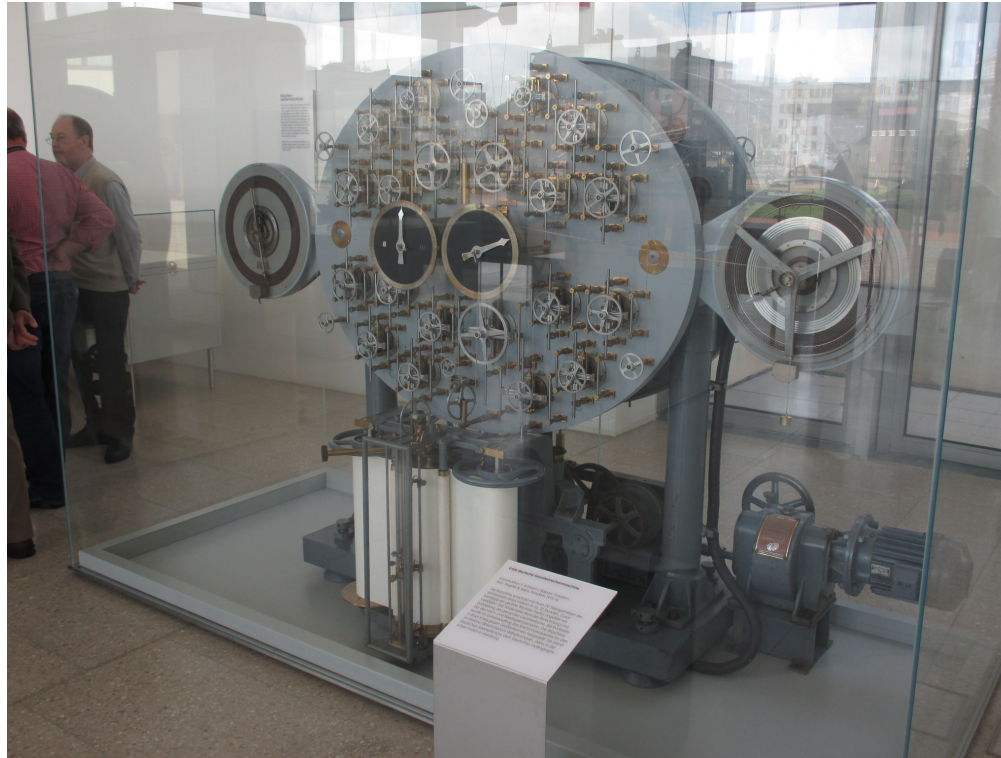
automation dans la production des tableaux résultants

Encore des machines mécaniques basées sur le système décimal.

# Exemple d'une machine analogue

---

Calculatrice de marées de 1915



Calcule les marées d'un port donnée (dans la mer du nord) pour un an dans 12 heures

# La “machine” de Turing

---

Alan Turing (1936)

Conception théorique d'une machine universelle :

ruban infini pour stocker des données + un état interne

la machine lit un symbole à la fois, se déplace par une position et modifie son état selon des règles fixes

*Machine de Turing universelle*: peut simuler d'autres machines de Turing

⇒ un programme est une donnée

Modèle reconnu comme étant équivalent en pouvoir à un ordinateur quelconque;  
terminaison indécidable

# Les années 1930

---

Les machines de Konrad Zuse : Z1 (1938) and Z3 (1941)

fonctionnement électro-mécanique

calcul **binaire**, virgule flottante

programmation avec des **boucles** (mais pas de branchement conditionnel)



# Pendant la guerre

---

**Colossus** (britannique, 1943, pour briser des codes)

existence connue publiquement depuis les années 1970

construit pour un objectif précis, pas universelle, pas vraiment programmable

calcul binaire

**Mark I** (américaine, 1944, calcul ballistique)

calcul décimal

programmable par bande perforée

Turing-complet



# Les premiers ordinateurs électriques

---

## ENIAC (1946)

construit à l'Université de la Pennsylvanie

Poids : 30 tonnes; 18.000 tubes électroniques

encore décimal (20 registres à 10 chiffres)

programmable en (dé)branchant des câbles

temps requis pour une multiplication: 3 ms

utilisé pour des calculs ballistiques

# Les ordinateurs construits par von Neumann

---

## EDVAC (1951)

Calcul binaire

programme stocké en mémoire

instructions de la forme  $(a_1, a_2, a_3, a_4, op)$ :

appliquer  $op$  sur les données aux adresses  $a_1$  et  $a_2$ , stocker le résultat à  $a_3$ ,  
prochaine instruction à  $a_4$ .

branchement conditionnel: comparer les données à  $a_1, a_2$ , continuer soit à  
 $a_3$  ou à  $a_4$ .

---

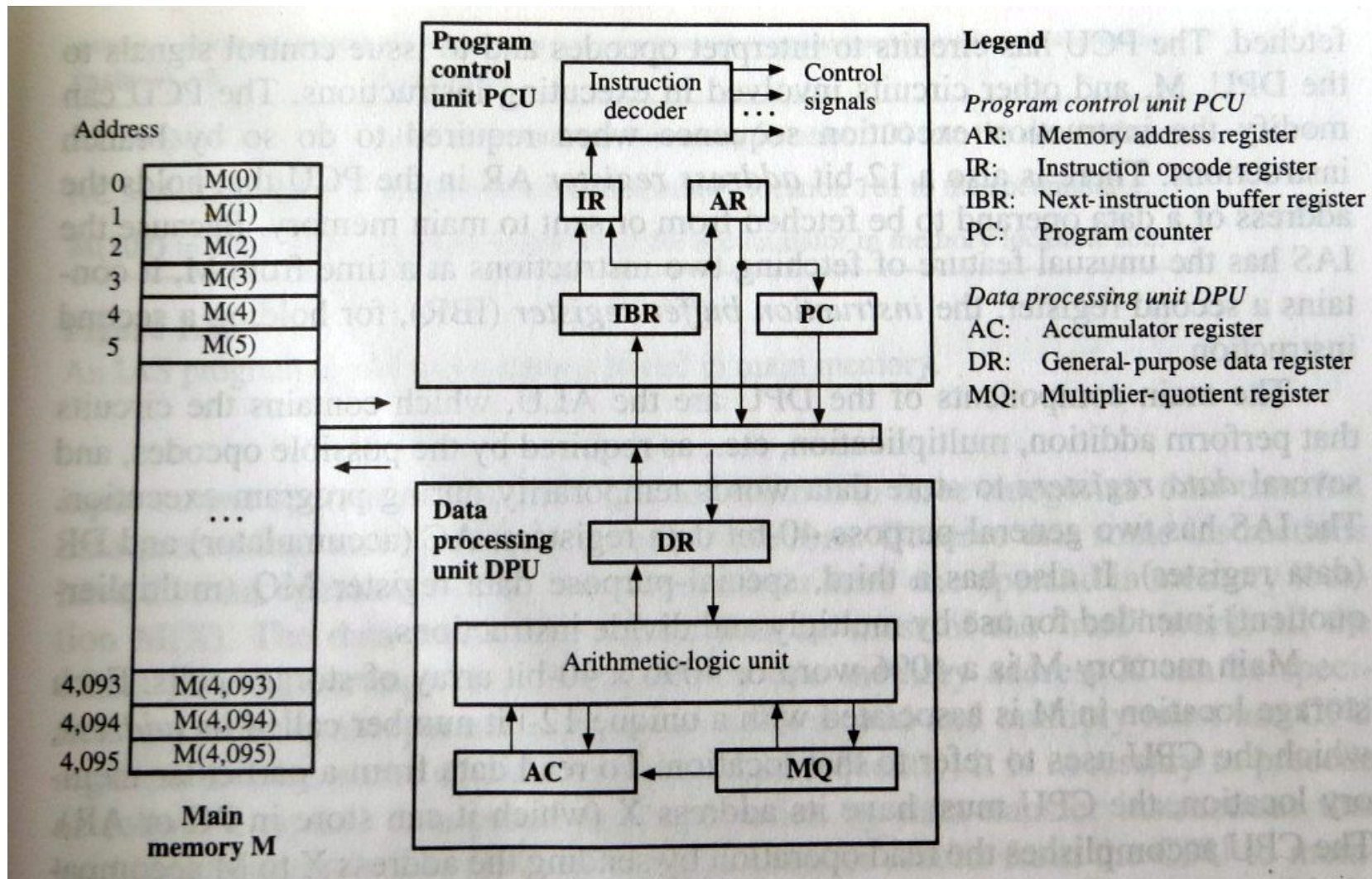
La **machine de l'IAS** (construit à Princeton)

mémoire:  $4096 = 2^{12}$  cellule ('mots') de 40 bit

notion d'un *processeur central* et *unité arithmétique-logique* (CPU et ALU)

considérée comme le prototype de l'architecture moderne

# Architecture de l'IAS



# Instructions de l'IAS

---

Un mot dans la mémoire peut être interprété soit comme une instruction, soit comme une valeur numérique.

Interpretation numérique : entier ou virgule fixe entre -1/+1

Instruction : un mot = 2 instructions de 20 bits

format  $(op, a)$  pour manipuler mémoire et registres

– op: code d'opération (8 bit) / a = adresse de 12 bit

transfert de données entre mémoire et registres/entre registres

addition/multiplication

opérations de contrôle (branchement conditionnel, code automodifiant)

# Ordinateurs de “deuxième génération” (50s-60s)

---

Couche physique :

remplacement de tubes par des transistors

plus petit, moins cher, plus rapide, plus fiable

Architecture:

instructions plus puissantes : adressage indirect, registres d'index

registres et opérations pour les virgules flottantes

réursion (opération sur pile)

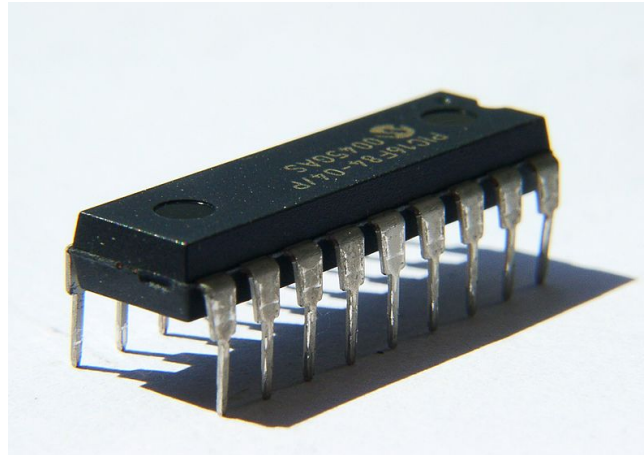
# Ordinateurs de “troisième génération” (1960s-70s)

---

Arrivée des **circuits intégrés** (IC)

beaucoup de transistors sur un espace énormément réduit

→ opérations plus rapides



Mémoires à vitesses différentes (cache); intégrée soit sur IC, soit ailleurs

---

Interface :

Langages de programmation et compilateurs

Ordinateurs centraux avec *terminaux* multiple  
(connection par ligne de téléphone possible)

Les premiers systèmes d'exploitation

Applications :

au-delà du calcul pur : p.ex., système de réservation de vol



# Les premiers systèmes d'exploitation :

---

organiser le travail de plusieurs utilisateurs

facilitation des tâches répétitives

système de fichiers

applications interactives

côte matériel :

ordonnancement, interruptions

processeur bascule entre modes superviseur/utilisateur

# Dès les 1970s

---

Ordinateurs toujours plus petits et rapides

Production en masse (desktops, laptops, smartphones, ...)

Développement d'une couche "micro-architecture" :

Interface abstraite proposée au programmeur toujours style "von Neumann"  
(registres, mémoire linéaire)

Microarchitecture optimise l'exécution des programmes en assembleur :  
découpage en micro-instructions, exécution parallèle, mémoire cache,  
mémoire virtuelle, ...