

Architecture et Système

Stefan Schwoon

Cours L3, 2016/17, ENS Cachan

Contenu du cours

Cours orienté vers la pratique :

Comment fonctionne un ordinateur ?

Comprendre ce qui se passe à l'intérieur

Mieux s'en servir pour ses projets

Exemples : Shell, système d'exploitation, interfacier avec d'autres services, codage des données

Aspects matériel et logiciel – architecture / système

Relation entre architecture et système

L'utilisateur travaille avec des **applications** (navigateur, traitement de texte, ...).

Pour fonctionner, les applications font appel à un ensemble de logiciels qui forment le **système d'exploitation**.

C'est le système qui donne des bonnes instructions au **matériel** pour faire fonctionner les applications.

⇒ pour créer ses propres applications, il faut connaître le système

⇒ pour comprendre comment le système peut fonctionner, il faut une certaine connaissance du matériel (partie **architecture**)

Contenu du cours

Objectifs:

Meilleure compréhension du comportement de l'ordinateur

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

Sujets:

Couche physique, opérations sur les bits/mots, représentation des données

Organisation d'un processeur, composants, périphériques

Composants d'un système d'exploitation (basé sur le standard POSIX): processus, fichiers, mémoire, ordonnancement, ...

Langages utilisés : assembleur (un peu) et C

Littérature

Architecture :

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

Système d'exploitation :

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

Organisation

Cours: Lundi 13:45–15:45, C321 (Stefan Schwoon)

TP: Vendredi 13–16, C411, (Francis Hulin-Hubard)

Contrôles de connaissances : Projet + Examen (50/50), détails à confirmer

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

E-mail: schwoon@lsv.fr

Bureau: RH-B-114 (LSV)

Aujourd'hui

Histoire abrégée 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 premiers ordinateurs automatiques

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.)

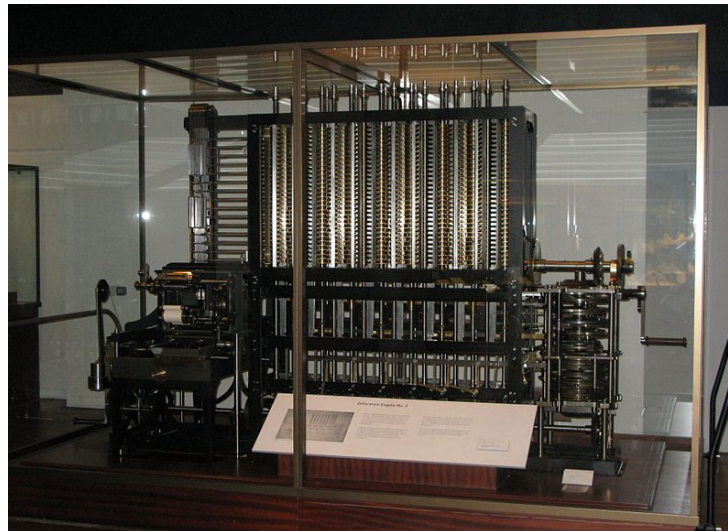
production en masse : à partir du 19e siècle (Thomas de Colmar)

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 :

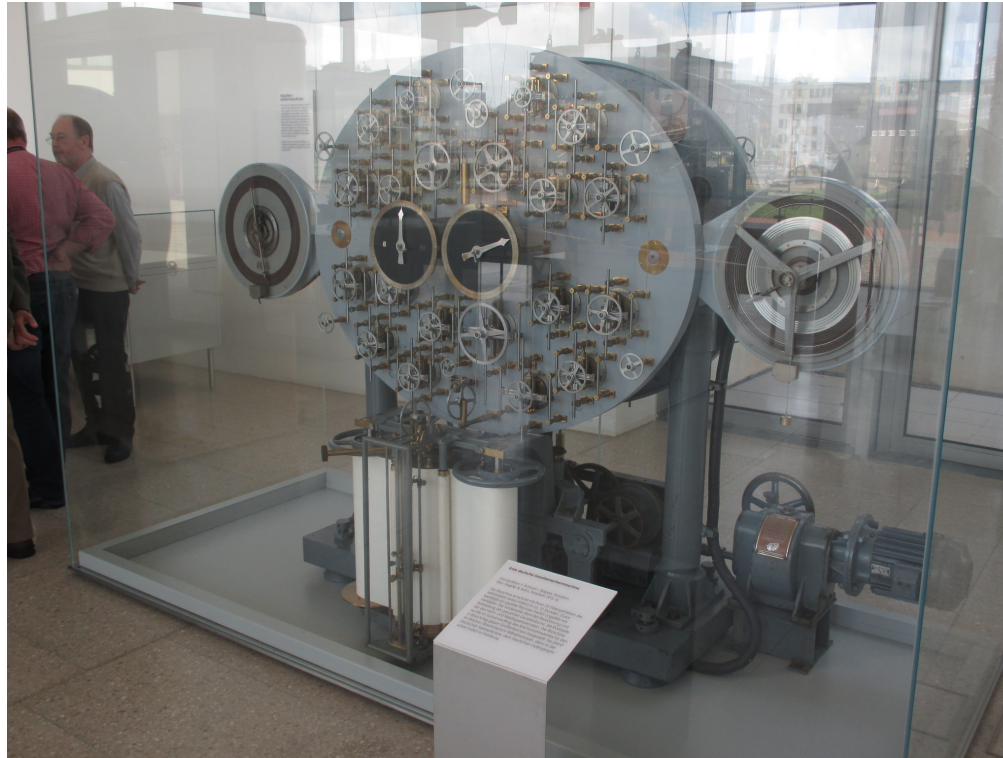
calculatrices utilisés par les ingénieurs, navigateurs, militaires, ...

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 .

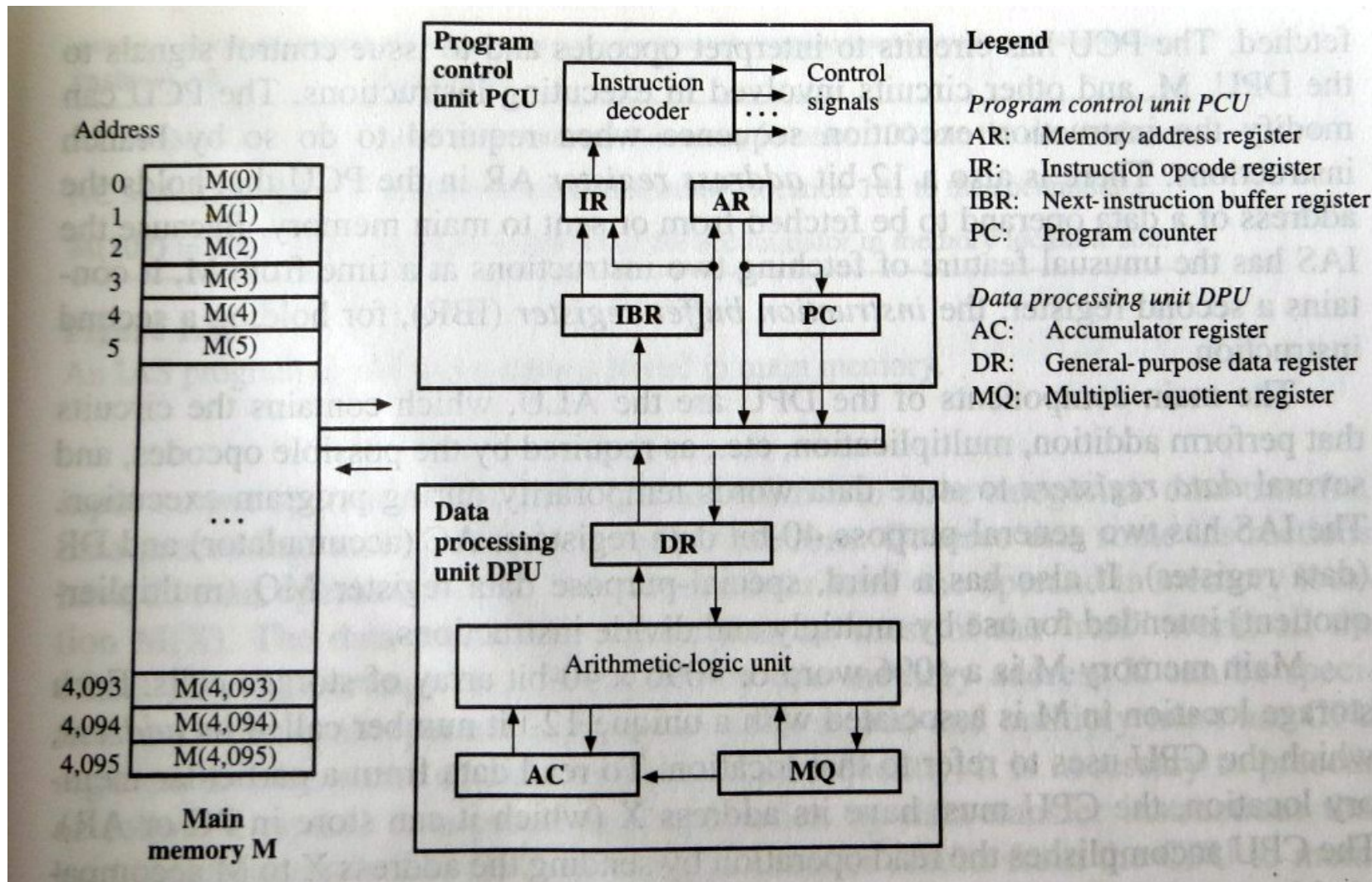
L'IAS (construit à Princeton)

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

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

considéré 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)

Interface humain :

Langages de programmation et compilateurs

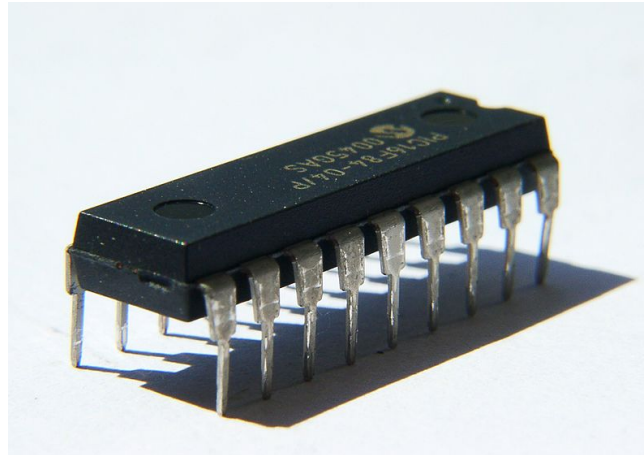
Les premiers systèmes d'exploitation

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

Les premiers systèmes d'exploitation :

organiser le travail de plusieurs utilisateurs

applications interactives

ordonnancement, interruptions

CPU en mode “superviseur” ou “utilisateur”

système de fichiers

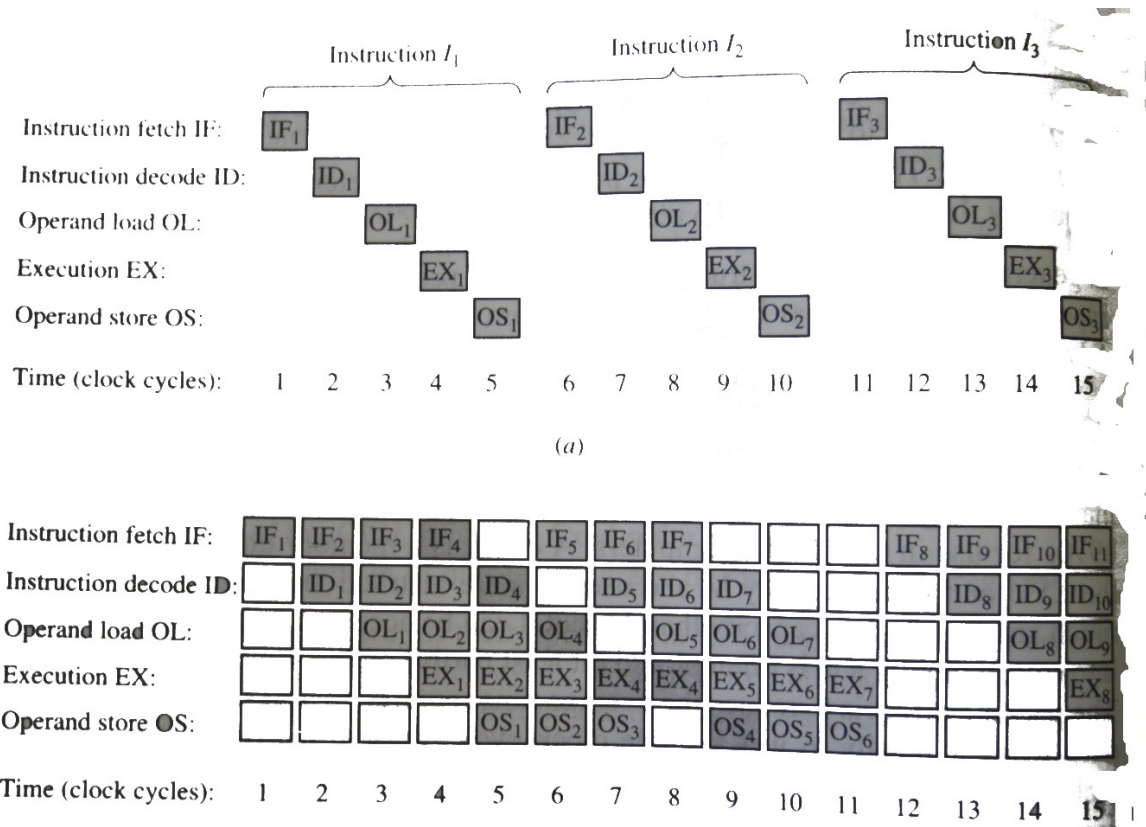
développement d'Unix à partir des 1970s

Dès les 1970s

Ordinateurs toujours plus petits et rapides

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

Architecture superscalaire



La concurrence

Le concurrence devient plus important :

composants spécialisés travaillant en parallèle (GPU, FPU)

processeurs multi-cœurs

réseaux

→ étude d'algorithmes concurrents