

Architecture des ordinateurs I

Frédéric Boulanger

CentraleSupélec



Objectifs et programme

Pourquoi un cours d'architecture

- ▶ Démystifier le fonctionnement des ordinateurs
- ▶ Connaître leur principe de fonctionnement
- ▶ Savoir ce qui est mis en œuvre dans un système informatique



Les ordinateurs sont partout

- ▶ Pourquoi ?
- ▶ Comment ?
- ▶ Numérisation des données et des traitements



Les fondements

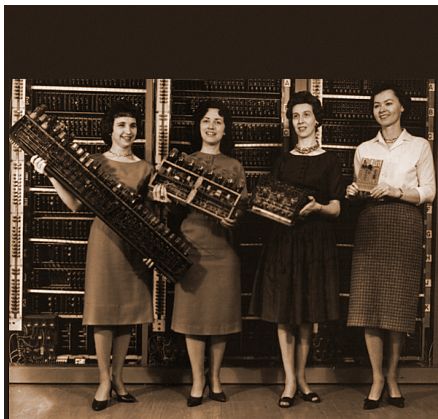
- ▶ De l'électronique à la logique et à l'arithmétique
- ▶ Logique séquentielle



Automate d'Henri Maillardet

Structure d'un ordinateur

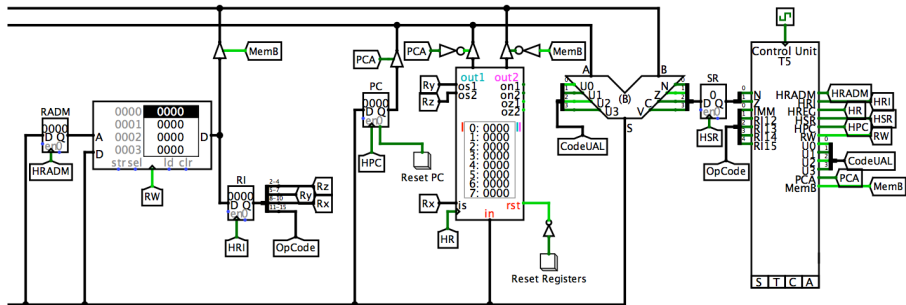
- ▶ Modèle de von Neumann
- ▶ Composants essentiels
- ▶ Instructions et séquençement des instructions
- ▶ Présentation de la machine de cours



ENIAC, 1940

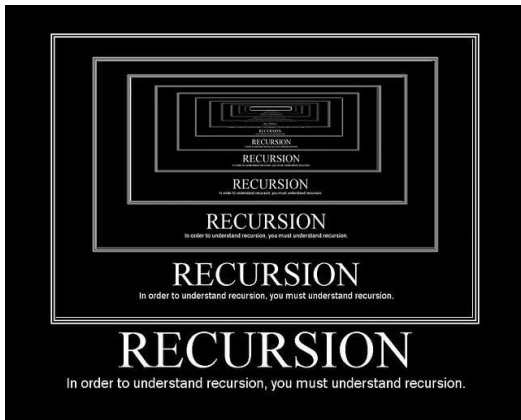
Premier BE

- ▶ Mise en œuvre de portes logiques
- ▶ Utilisation de bascules et registres
- ▶ Réalisation d'une ébauche de chemin de données



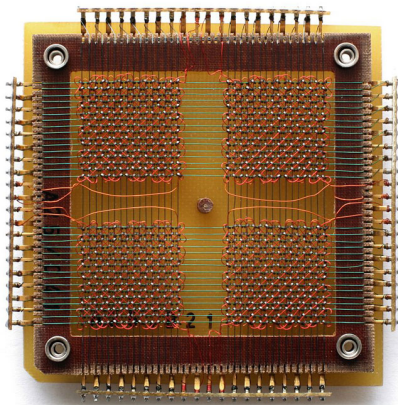
Traduction des langages de programmation

- ▶ Représentation des données
- ▶ Alternative et boucle
- ▶ Appels de fonction, récursion



Mémoires

- ▶ Les différents types de mémoire
- ▶ Stratégies pour leur utilisation
- ▶ Principes des hiérarchies de mémoires



Mémoire à tores de ferrite, 1955–1975



Deuxième BE

- ▶ Traduction d'instructions Python en langage machine
- ▶ Appels de fonction et récursion
- ▶ Programmation de l'algorithme d'Euclide pour calculer le PGCD

```
1      mov r1, #5      % charge la valeur 5 dans le registre r1
2      mov r2, #2      % charge la valeur 2 dans le registre r2
3      cmp r1, r2      % compare la valeur de r1 à celle de r2 (donc compare 5 et 2)
4      beq egaux       % saute à l'étiquette "egaux" si r1 égal r2
5      blt inferieur   % saute à l'étiquette "inférieuré si r1 est inférieur à r2
6      @superieur     b superieur   % le programme boucle ici si r1 est supérieur à r2
7      @egaux         b egaux       % le programme boucle ici si r1 et r2 sont égaux
8      @inferieur     b inferieur   % le programme boucle ici si r1 est inférieur à r2
```



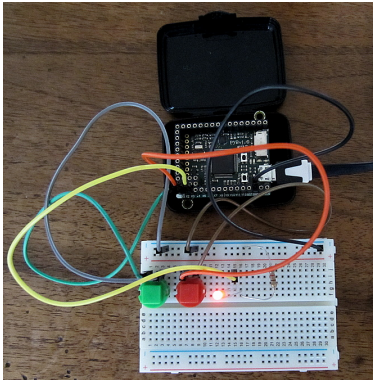
Entrées-sorties

- ▶ Comment un processeur communique avec son environnement
- ▶ Bus, périphériques, contrôleurs, adressage
- ▶ Interruptions, accès direct à la mémoire



Troisième BE

- ▶ Entrées-sorties simples
- ▶ Gestion des interruptions
- ▶ Timers



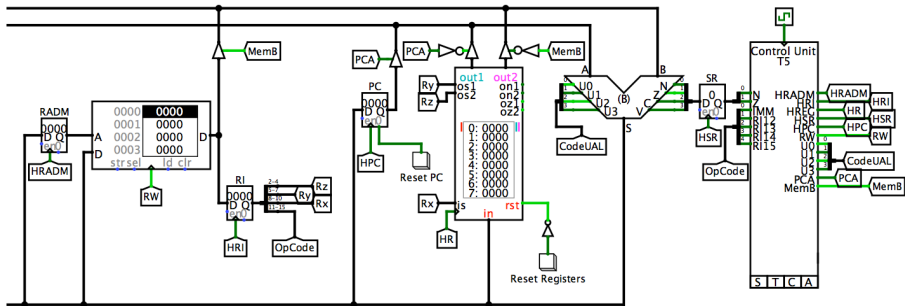
Pour aller plus loin

- ▶ Instructions privilégiées, appels système
- ▶ Multicœurs et pipeline
- ▶ Réponse à vos questions



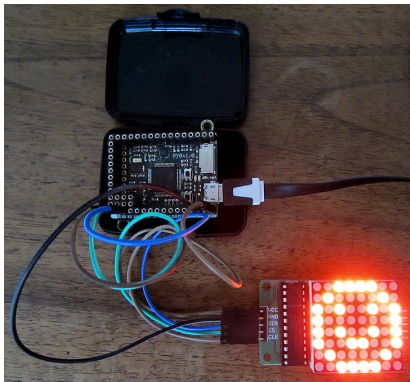
Première EL

- ▶ Conception d'un microprocesseur
- ▶ Ajout de l'adressage indexé à la machine de cours
- ▶ Ajout de l'instruction **b1** pour les appels de fonction



Deuxième EL

- ▶ Entrées-sorties
- ▶ Utilisation d'un bus série
- ▶ Configuration d'un contrôleur de périphérique
- ▶ Affichage sur une matrice de LED 8x8



Suite...

Architecture

