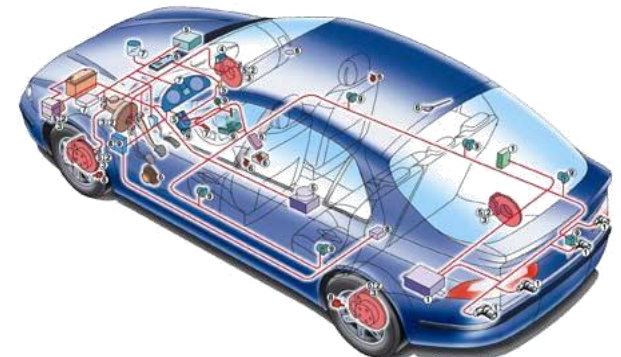
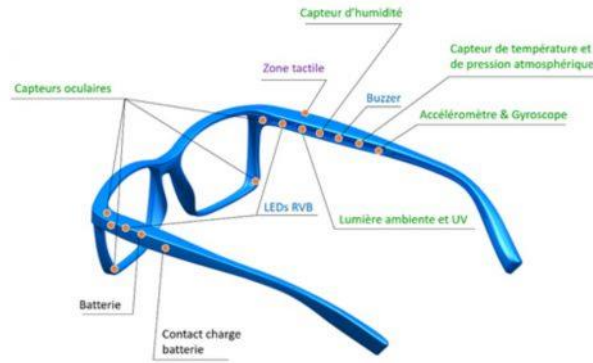


Architecture des systèmes embarqués SI3

B. Miramond

Objectifs du projet

Domaines des systèmes embarqués



B. Miramond

Kits de développement embarqués



BeagleBoard

Architecture

ARMv7l Cortex-A8

Processor

TI OMAP 3530 720MHz

RAM

256MB

Raspberry Pi

Architecture

ARMv6l

Processor

Broadcom BCM2835 700MHz

RAM

512MB



Raspberry Pi 2, 3 puis 4

Architecture

ARMv7l Cortex-A7 -> Cortex®-A72

Processor

Broadcom 900MHz -> 1,5 Ghz

RAM

1 GB -> 4GB

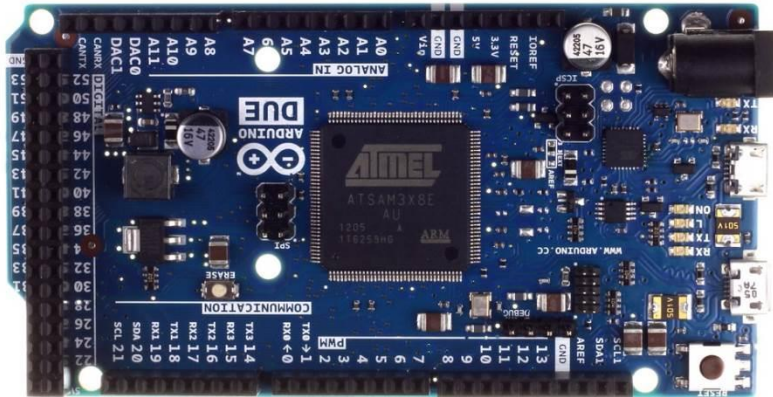
Objectif du module

- Comprendre les mécanismes matériels essentiels de l'informatique.
- Comprendre l'organisation de l'ordinateur autour de l'élément central, le processeur.
- Pour cela, étudier les différentes couches qui structurent la machine et la philosophie de son utilisation.
- En bref, remonter du circuit numérique jusqu'à l'architecture processeur

En pratique

- Etudier le jeu d'instruction ARM v7
- Réaliser un simulateur de processeur ARM à travers le **projet P-ARM** : *Polytech ARM-based embedded processor*

ARM dans les cartes Arduino



Arduino Due

Architecture

ARMv7 Cortex-M3

Processor

Atmel SAM3X8E – 84MHz

RAM

96KB

Arduino Zero

Architecture

ARMv7 Cortex-M0

Processor

Atmel SAMD21 – 48MHz

RAM

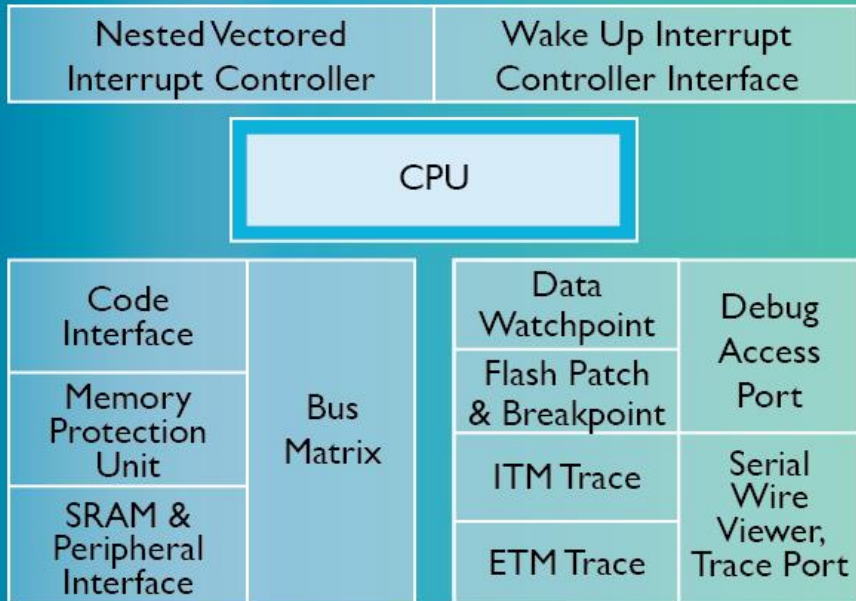
32 KB



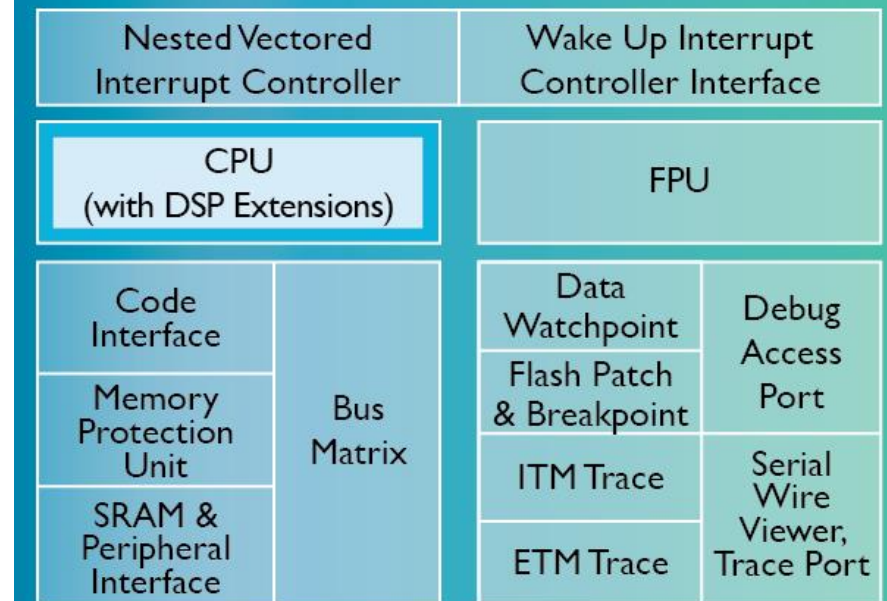
Famille Cortex M3

ISA Support	Thumb® / Thumb-2
Pipeline	3-stage
Performance Efficiency	1.25 / 1.50 / 1.89 DMIPS/MHz**
Memory Protection	Optional 8 region MPU
Interrupts	1 to 240 physical interrupts
Interrupt Priority Levels	8 to 256 priority levels

ARM® Cortex®-M3



ARM® Cortex®-M4



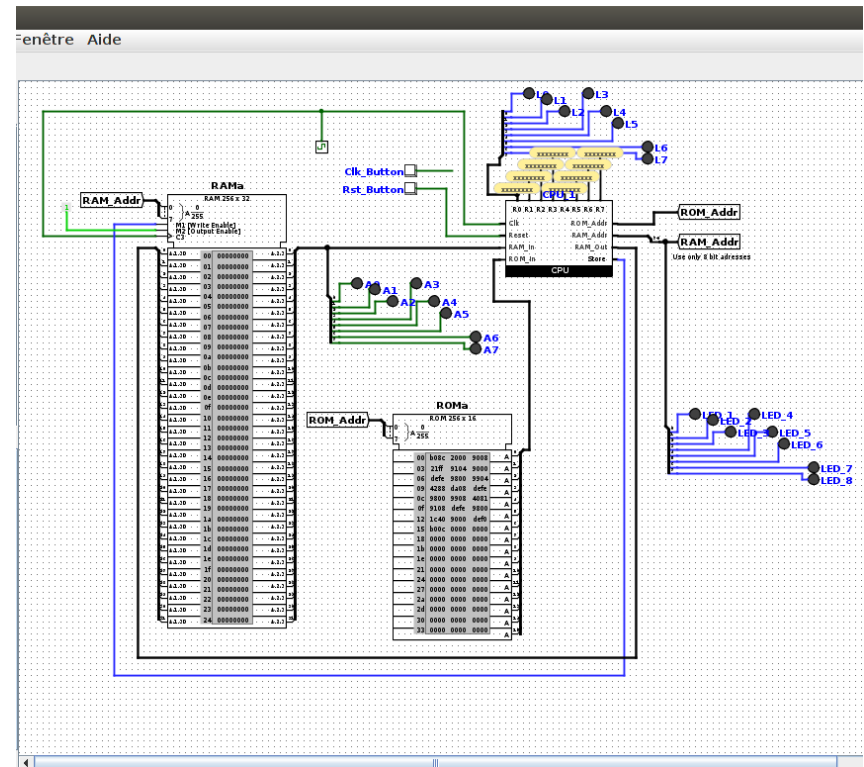
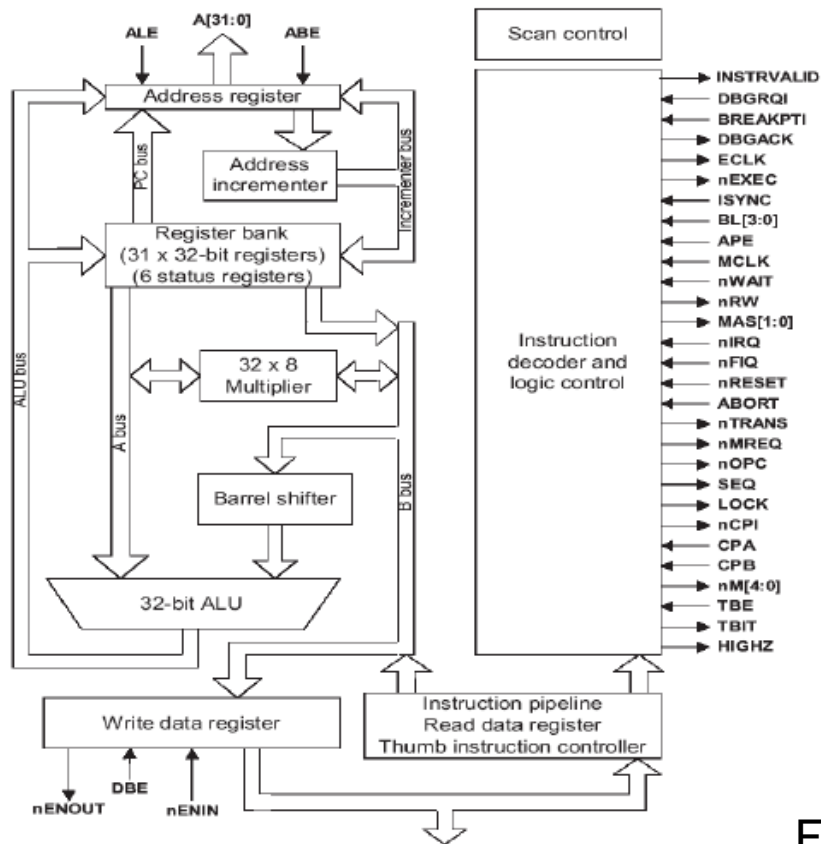
Objectifs du projet P-ARM

Polytech ARM-based embedded processor

Concevoir une version allégée du processeur
ARM Cortex-M0

1. Utilisation d'un éditeur de circuits numériques
2. Conception matérielle
3. Génération de code binaire depuis l'assembleur,
voire depuis du code C
4. Validation par simulation
5. Test de déploiement sur FPGA

Architecture du processeur P-ARM



Fichiers de démarrage du projet disponibles sur : lms.univ-cotedazur.fr

Logiciel

Assembleur / C

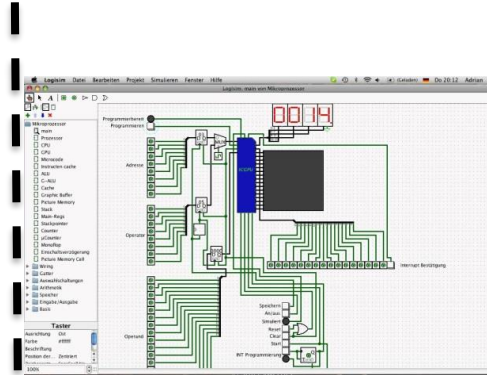
```
pgcd:
while: cmp R1,R2
      bne do
      b endwhile
do:
if:    cmp R1,R2
      bhi then
      b else
then:  sub R1,R1,R2
      b endif
else:  sub R2,R2,R1
endif: b while
endwhile:
      mov pc,R14
```

0x2345 4569
0x0124 4828
0x2839 4018
0x3282 0294
...

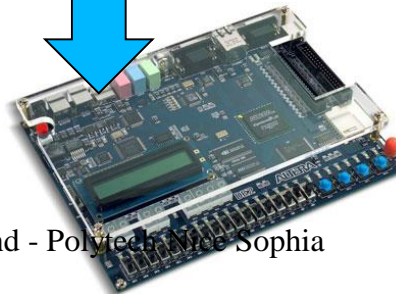
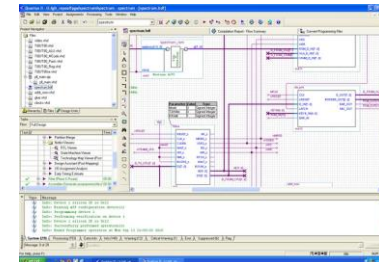
Initialisation
de la
mémoire
dans Logism

Matériel

Architecture / Simulation



Simulation

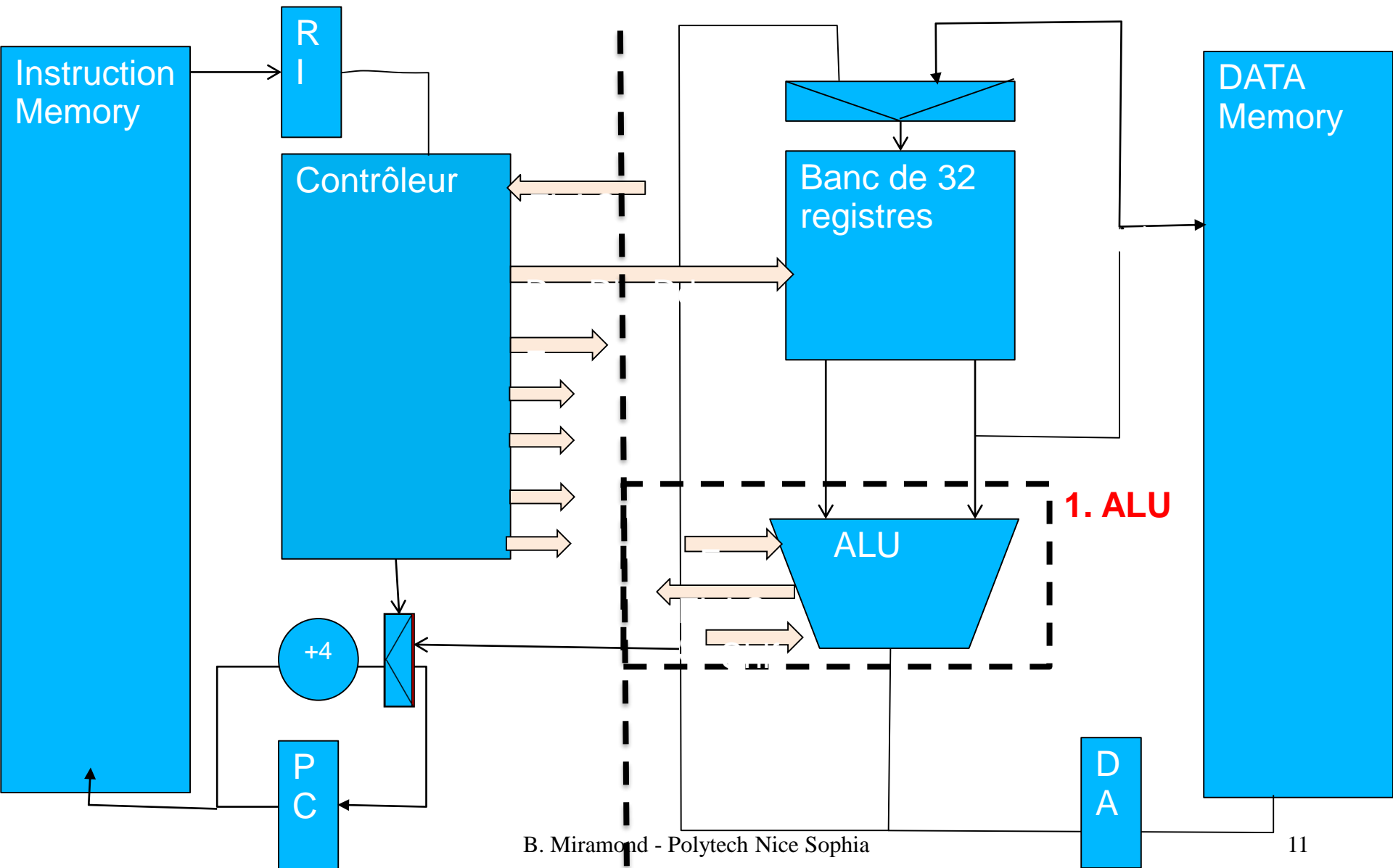


FPGA (possible mais hors
sujet)

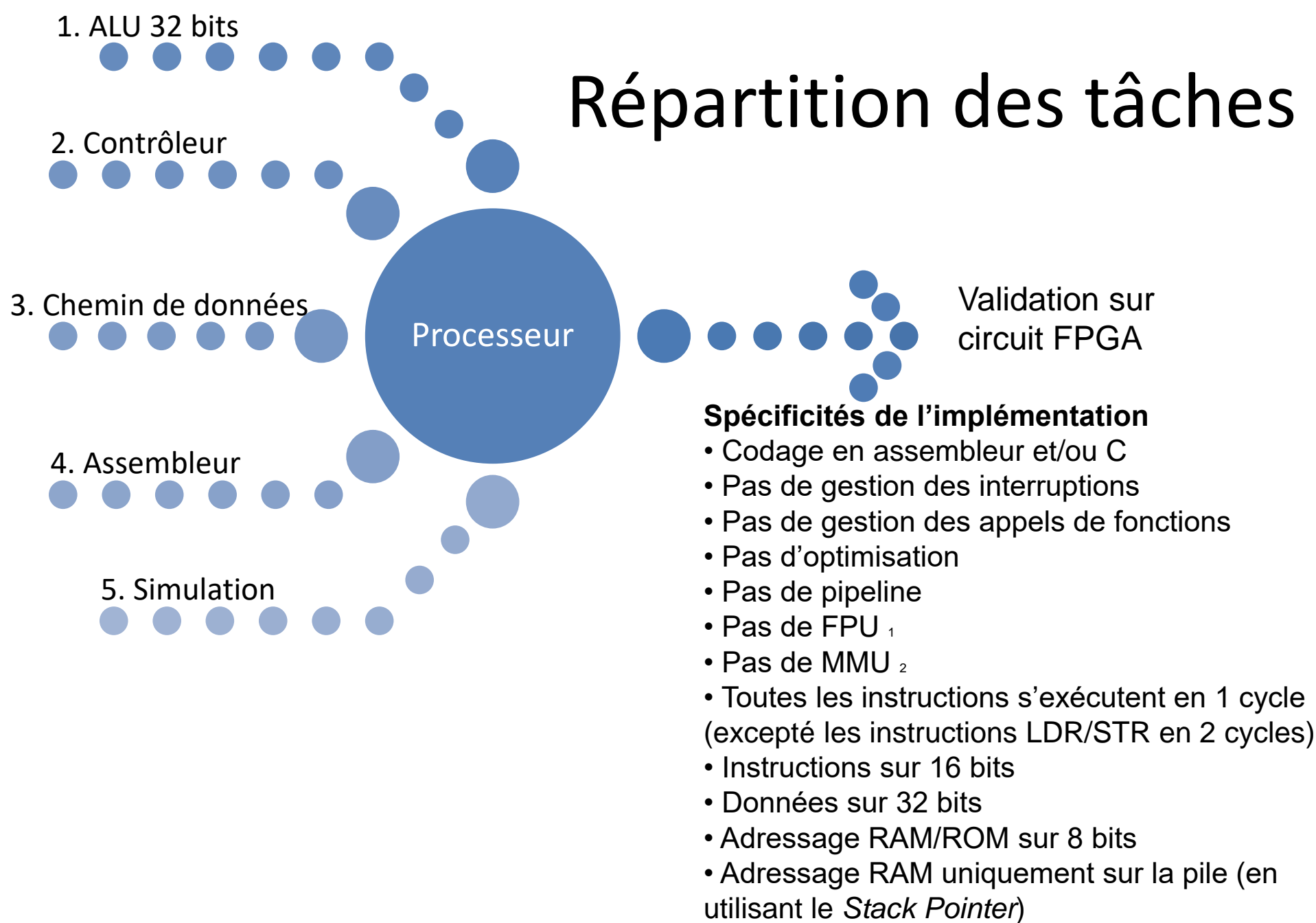
Architecture générale

2. Contrôleur

3. Chemin de données



Répartition des tâches



Rôles de chaque partenaire

1. ALU (Hardware)	2. Contrôleur (Hardware)	3. Chemin de données (Hardware)	4. Assembleur (Software)	Tous. Validation
Réaliser les blocs d'opérateurs arithmétiques et logiques	Lecture instruction depuis la mémoire de programme	Mouvement de registre à registre	Parser un fichier assembleur	Intégration de tous les composants
Générer les Flags	Décoder les instructions et générer les signaux de commande du chemin de données	Lecture mémoire de données Ecriture en mémoire de données	Générer le fichier binaire à charger dans la mémoire d'instruction de logisim	Validation par simulation
	Calcul d'adresse de la prochaine instruction	Envoie d'adresse pour les lectures / écritures	Générer le fichier binaire à charger dans la mémoire de données de logisim	Tester le déploiement du processeur sur FPGA

Constituer les groupes !

4 personnes par groupe, dont 1 chef d'équipe.

Ré-organisation des efforts du groupe en fonction de l'avancement.

Inscriptions des équipes projets :

<https://lms.univ-cotedazur.fr>

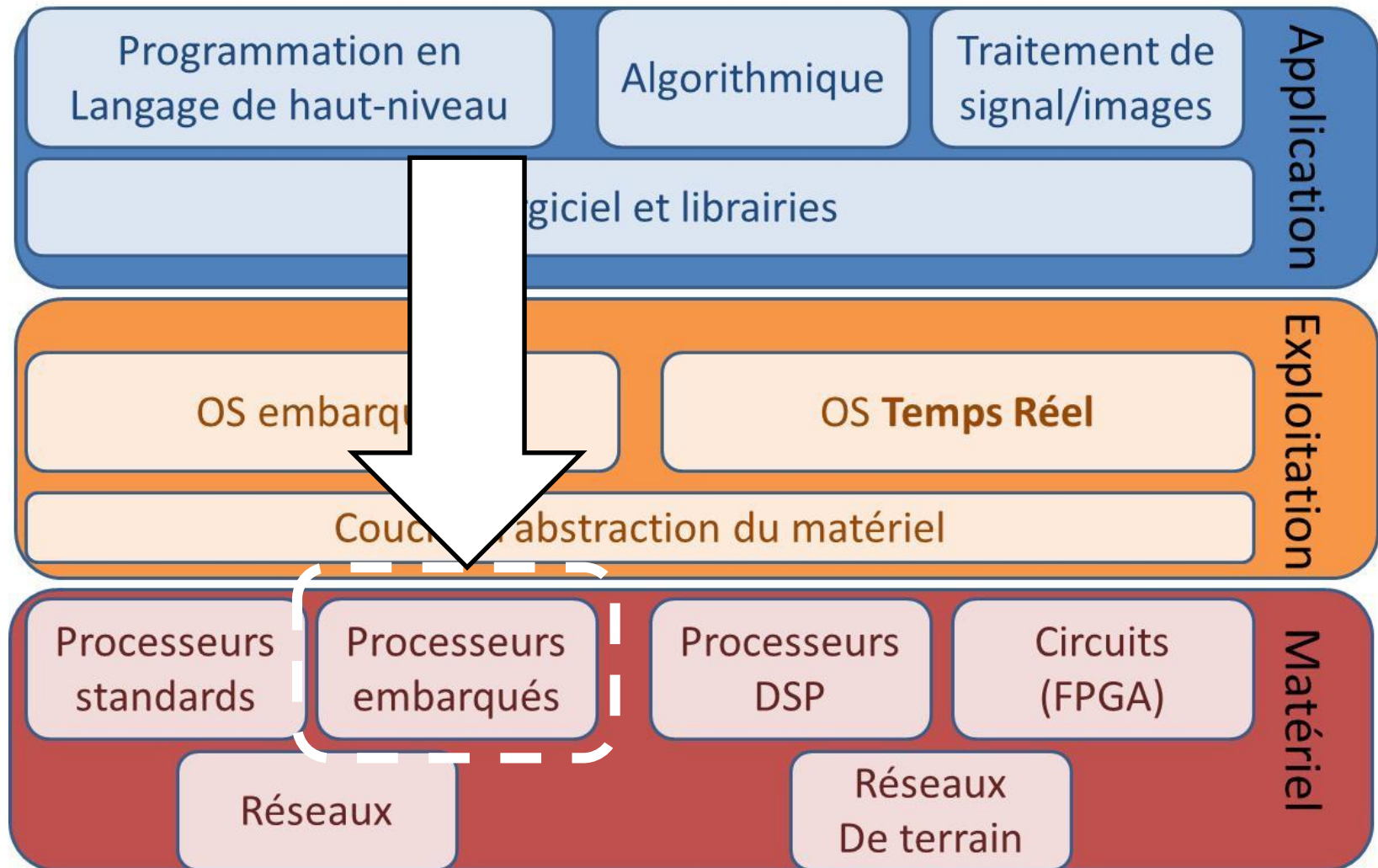
Soutenance la première semaine de janvier:

- 15 minutes par groupe (10 min. de prés. / 5 min. de questions)
- 15 planches maximum
- Présentation des parties du projet
- Présentation des résultats avec des exemples
- Répétitions obligatoires

Organisation du module

- 7 séances de cours de 1 heure
- 7 séances de TD/projet de 3h
- 1^{er} semaine de janvier : soutenance des projets par groupes
- 1ere semaine de janvier : contrôle
- 2 notes : Projet, Contrôle final

Organisation du système embarqué



Progression des notions d'architecture embarquée

- SI3 – Architecture & Réseaux
- SI4 – Capteurs/actionneurs
- SI5 – Conception conjointe
logicielle/matérielle des systèmes embarqués

Bibliographie

- Learning Center

- **Organisation et architecture de l'ordinateur**, W. Stallings
- **Architecture des machines et systèmes informatiques**, A. Cazes, Dunod
- **Architecture de l'ordinateur**,
A. Tanenbaum
Learning Centre SophiaTech (Cote : 004.22 TAN)
- **Structured computer organization**,
Andrew Tanenbaum
Learning Centre SophiaTech Salle de lecture (Cote : 004.22 TAN)

- Autres

- **Architecture des machines et des systèmes informatiques : cours et exercices corrigés**
Alain Cazes, Joëlle Delacroix BU Sciences
Valrose RDC (Cote : 004.2 CAZ)
- **Architecture des ordinateurs : une approche quantitative**
John L Hennessy David A Patterson
BU Sciences - Valrose RDC (Cote : 004.2 HEN)
- **Les DSP - Famille ADSP218X : principes et applications**
Michel Pinard Paris : Dunod DL 2000
Learning Centre SophiaTech Salle de lecture (Cote : 621.382 2 PIN)
- **Architecture des ordinateurs, une approche quantitative**, Hennessy & Patterson

Quelques liens sur l'embarqué

- Embedded linux
 - <http://elinux.org/>
- ARM micro-contrôleurs
 - <https://www.arm.com/markets/internet-of-things-iot.php>
- Magazine en ligne l'embarqué
 - <http://www.lembarque.com/>

Introduction

Retour sur l'historique

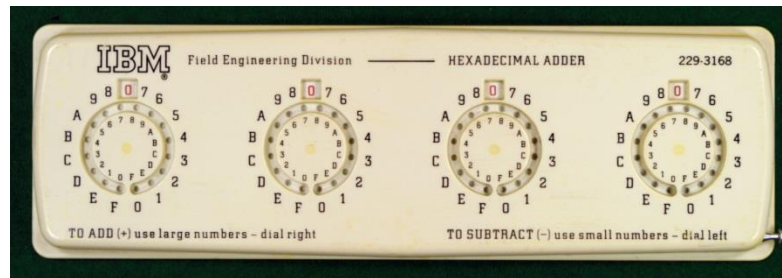
(à lire par vous-même)

« Le passé est immuable, l'avenir est incertain. »

Cicéron, Du destin, VII - env. 44 av. J.-C

I. Les premiers calculateurs mécaniques

- Blaise Pascal (1623-1662)
 - 1642 Première calculatrice mécanique
 - Addition, soustraction en base 10



- Leibnitz (1646-1716)
 - 1670 Ajout de l'opération de multiplication, et division

- Charles Babbage (1792-1871)



- « *Machine à différence* » :

- un seul algorithme (méthode des différences finies)
 - pour les tables numériques pour la navigation en mer
 - Résultats gravés sur un plateau de cuivre

- Machine analytique (1840)

- Lecture sur des cartes perforées
 - Différents algorithmes exécutables
 - 4 parties



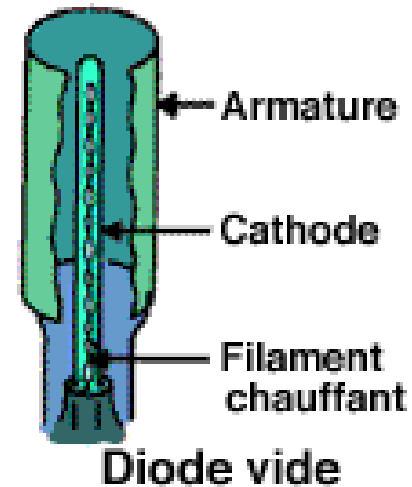
- Le magasin(la mémoire) = 1000 mots de 50 chiffres = 1,25Mo
 - Le moulin (l'unité de calcul)
 - L'entrée (lecteur de cartes perforées)
 - La sortie (perforation ou impression)

- 1843 Premier programme informatique

- Ada Lovelace (langage d'assemblage)

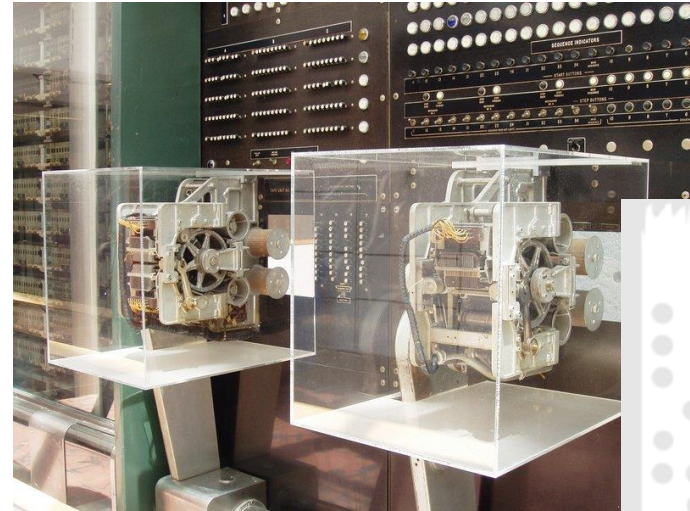
II. les tubes à vides (1945-1955)

- Technologiques
 - 1904 John Flemming invente le tube à vide :
 - Interrupteur électronique (pannes fréquentes, encombrant)
- Réalisations : Alan Turing
 - 1943 Premier calculateur numérique
 - Resté sous silence pendant 30ans



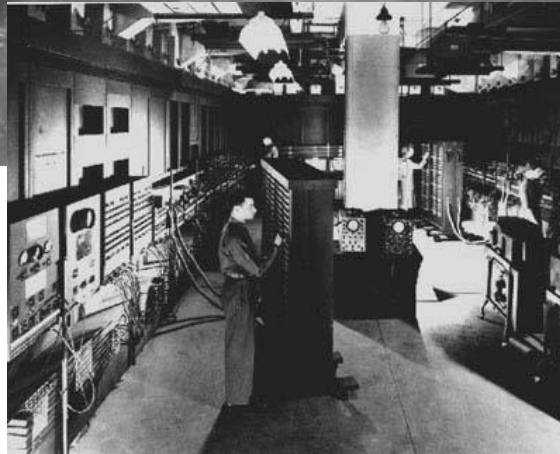
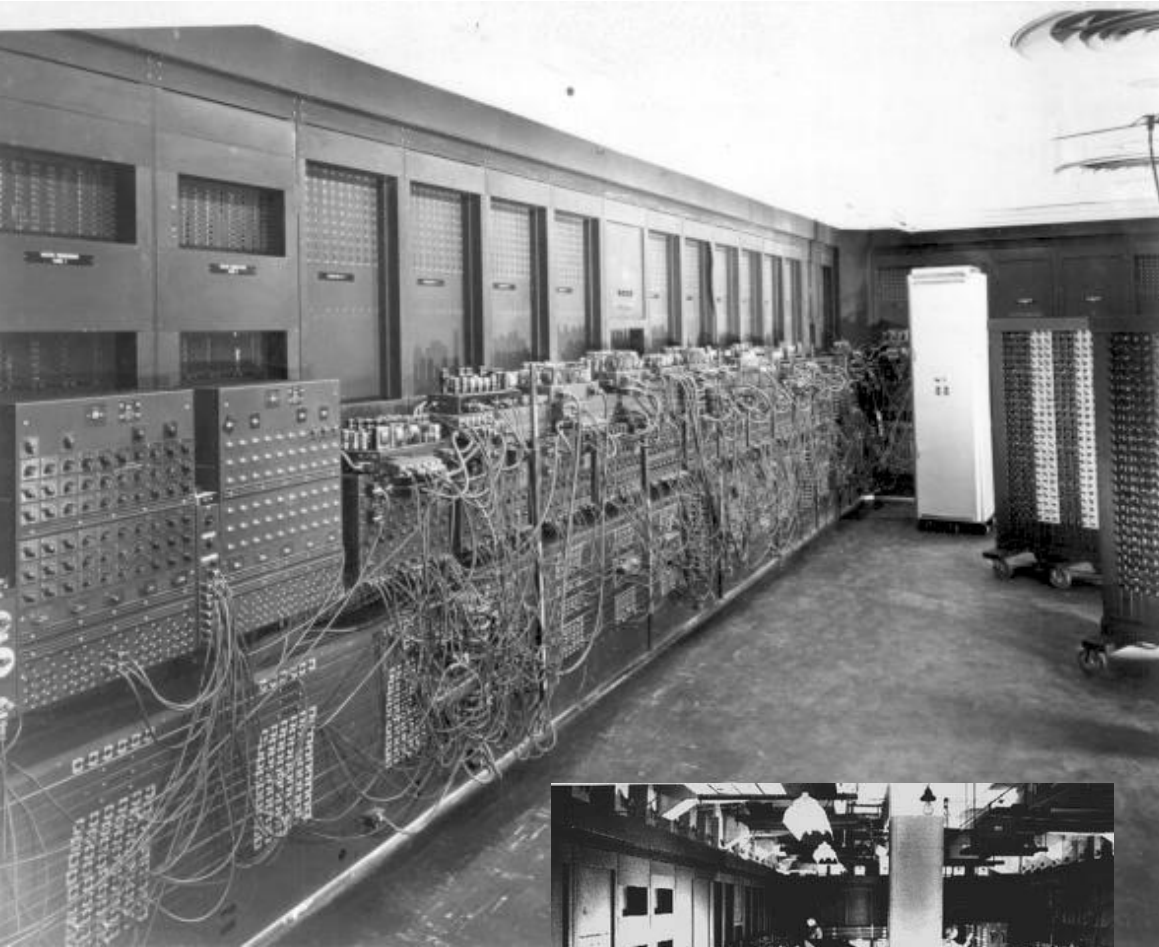
Machine de Harvard

- « the beginning of the era of the modern computer »
- Ordinateur à **relais**
- 1944 Howard Aiken
 - Mark 1 à **Harvard**
 - Entièrement automatique
 - Electro-mécanique
 - 16m x 2,5m – 4500 Kg
 - Démarré par un moteur 5 chevaux
 - Utilise un arbre de synchronisation
 - 72 mots de 23 chiffres
 - Temps de cycle : 6 secondes
 - Soit 0.16 Hz



Addition	1s
Multiplication	6s
Division	15,3s
Logarithme	1min
trigo	1min

- ENIAC (1946)
 - 30m de long
 - 2.5m de haut
 - 18 000 tubes à vide
 - 1500 relais
 - 30 Tonnes
 - 140 KW
 - 20 registres de 10 chiffres décimaux
 - 6000 commutateurs pour la programmation
- Participant :
Von Neumann



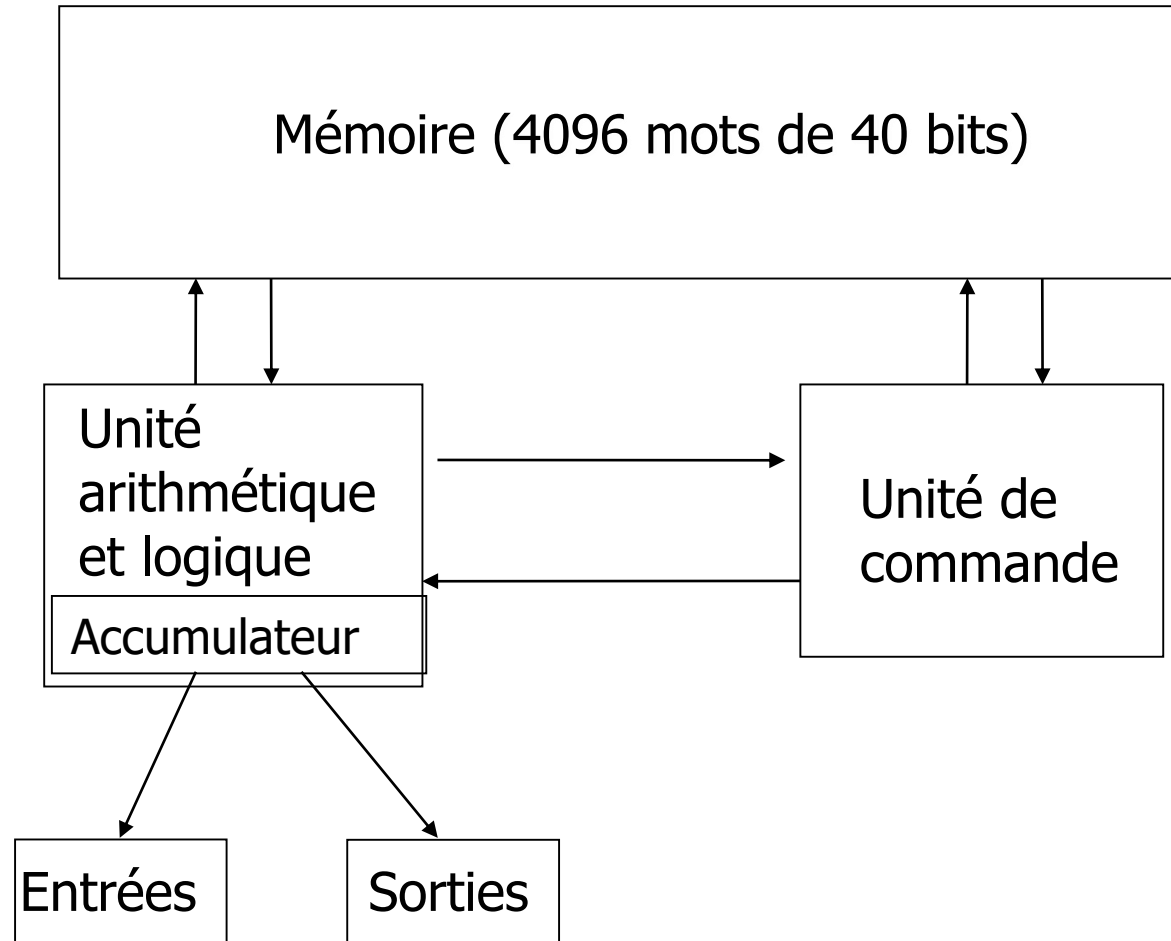
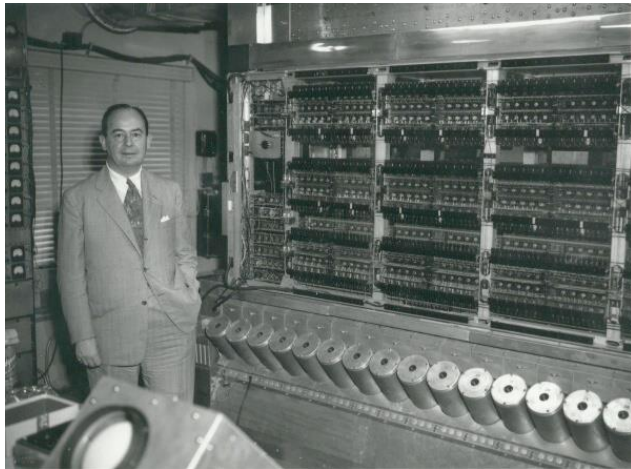
La machine de Von Neumann

- Von Neumann (1903,1957)
 - Il se greffe au projet ENIAC après 1946
- Idées
 - On peut représenter le programme sous une forme numérique et le ranger en mémoire comme les données (plus de programmation des commutateurs)
 - Introduction de l'arithmétique **binaire**!
 - Ancêtre à la base des architectures actuelles
- Résultat : l'IAS (1952)
 - Premier ordinateur à programme enregistré
 - Mémoire séparée du CPU

Schéma de la machine de Von Neumann

5 parties

- La mémoire
- L'UAL
- L'unité de commande
- Les entrées
- Les sorties



L'apport de Von Neumann

- Le premier ordinateur à programme enregistré
- Le programme est traduit de manière numérique et écrit en mémoire.
- Pour cela, création d'un jeu d'instructions machine, et transformation des calculs en séries d'instructions
- La machine devient plus flexible parce que programmable

Von Neumann's Bottleneck

- La séparation entre le CPU et la mémoire a amené à la notion de *Von Neumann's Bottleneck*
- (goulot d'étranglement qu'est la communication entre processeur et mémoire).
- " Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the von Neumann bottleneck. Not only is this tube a literal bottleneck for the data traffic of a problem, but, more importantly, it is an **intellectual bottleneck** that has kept us tied to word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand. Thus programming is basically planning and detailing the enormous traffic of words through the von Neumann bottleneck, and much of that traffic concerns not significant data itself, but where to find it."
- [John Backus](#) in his 1977 ACM [Turing award](#) lecture

Machine de Von Neumann

- Structure des mots mémoire mots de 40 bits :
 - Soit 2 instructions de 20 bits
 - Soit 1 entier signé de 40 bits
- Instructions composées de 2 champs
 - 8 bits codent le type d'instruction
 - 12 bits pour spécifier (adresser) un des 4096 mots



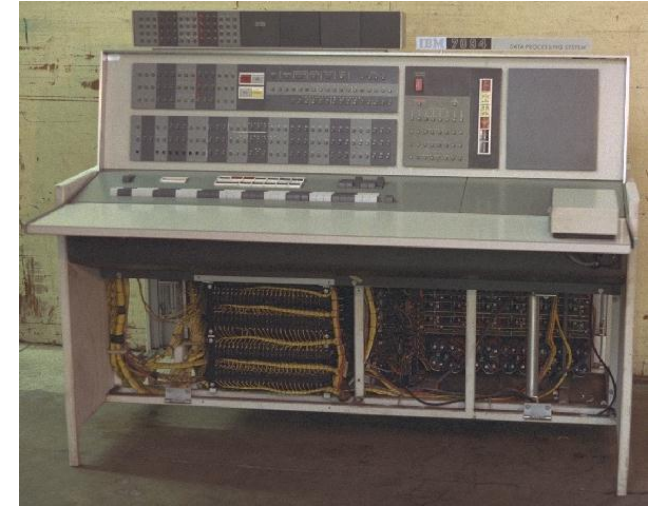
- Ex :
 - PUSH \$1024
 - ACC \$0015

les transistors (1955-1965)

- Innovation technologique
 - 1948 : Invention du **transistor** aux Bell Labs (Bardeen, Shockley)
 - Interrupteur commandé électroniquement
 - Beaucoup moins encombrant que le tube à vide
- Concepts architecturaux
 - Programmation en langage machine
 - Microprogrammation
 - Représentation des nombres en virgule flottante
- Logicielles
 - Apparition des systèmes d'exploitation
 - Programmation en assembleur
 - Premier langage de haut niveau : le FORTRAN

Le début des grandes firmes

- IBM
 - (7090, 7094, 7030...)
- Bull
 - (PDP-1, PDP-8...)
- DEC
 - (TX-0, TX-2...)
- CDC
 - (6600, 6700, Cray-1...)



IBM 7094



PDP-8

III. les circuits intégrés (1965-1980)

- 1961 Premiers circuits intégrés (sur silicium R. Noyce)
 - Innovations technologiques
 - Moindre encombrement
 - Moindre consommation
 - Gain en performances
- Innovations architecturales
 - Compatibilité des machines d'une même gamme (IBM: System/360)
 - Introduction du parallélisme (multi-programmation)
 - Systèmes à multi-processeurs
- Logiciels
 - Langages de programmation
 - Basic, Pascal, C, ...



IBM 360



PDP-11

Figure 1. IBM System/360 Model 40 Data Processing System

Naissance d'Intel, le premier microprocesseur

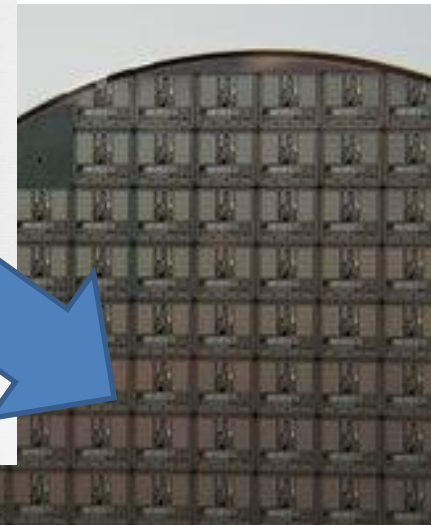
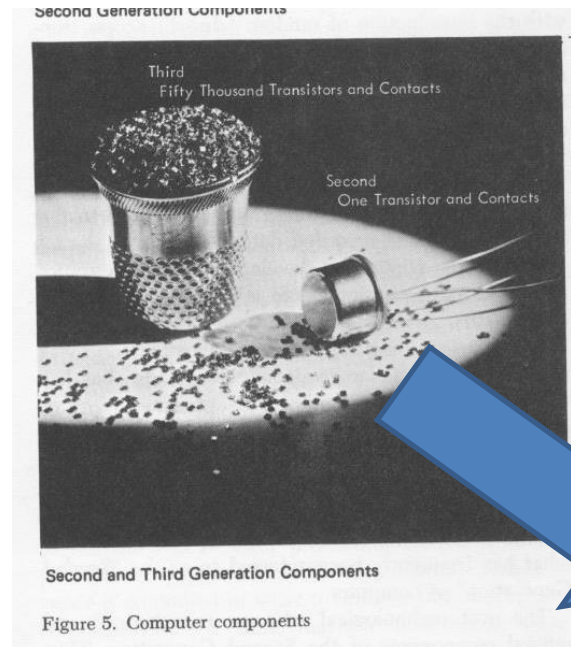
- 1968 R. Noyce, G. Moore et A. Rock fondent Intel Corporation
- 1970 première UC composée de 2300 transistors (LSI) : le 4004 processeur 4bits
- 1973 le 8008 processeur 8bits
- 1978 le 8086 processeurs 16 bits



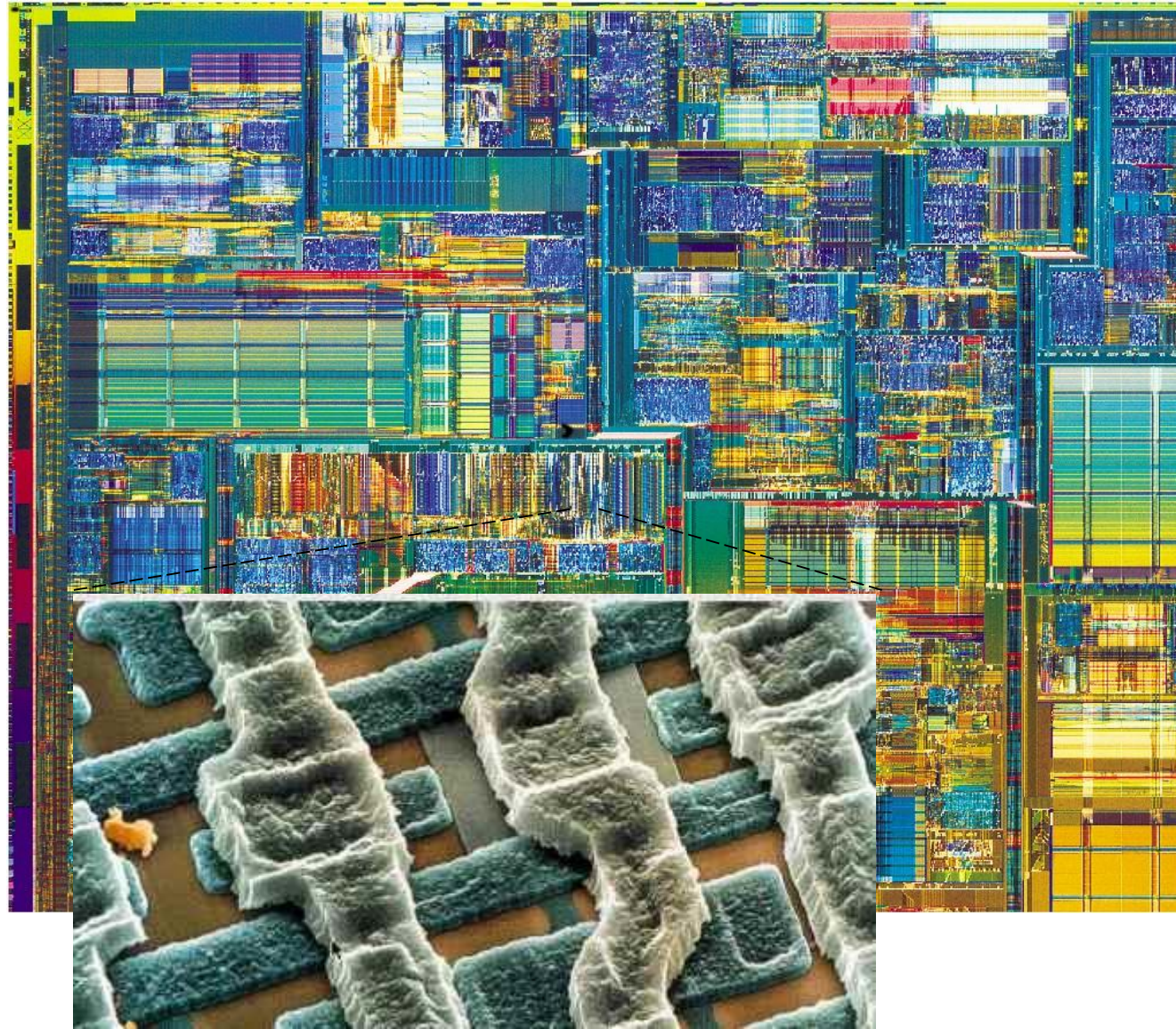
Microprocesseur 4 bits
108 KHz
60000 Instructions par seconde
2300 transistors

IV. Les VLSI (depuis 1980)

- Densité d'intégration augmente : VLSI
 - Dizaine de milliers -> millions de transistors sur une puce
- Passage du centre de calcul à l'ordinateur personnel
- Explosion du logiciel
 - Apple II vs. IBM PC

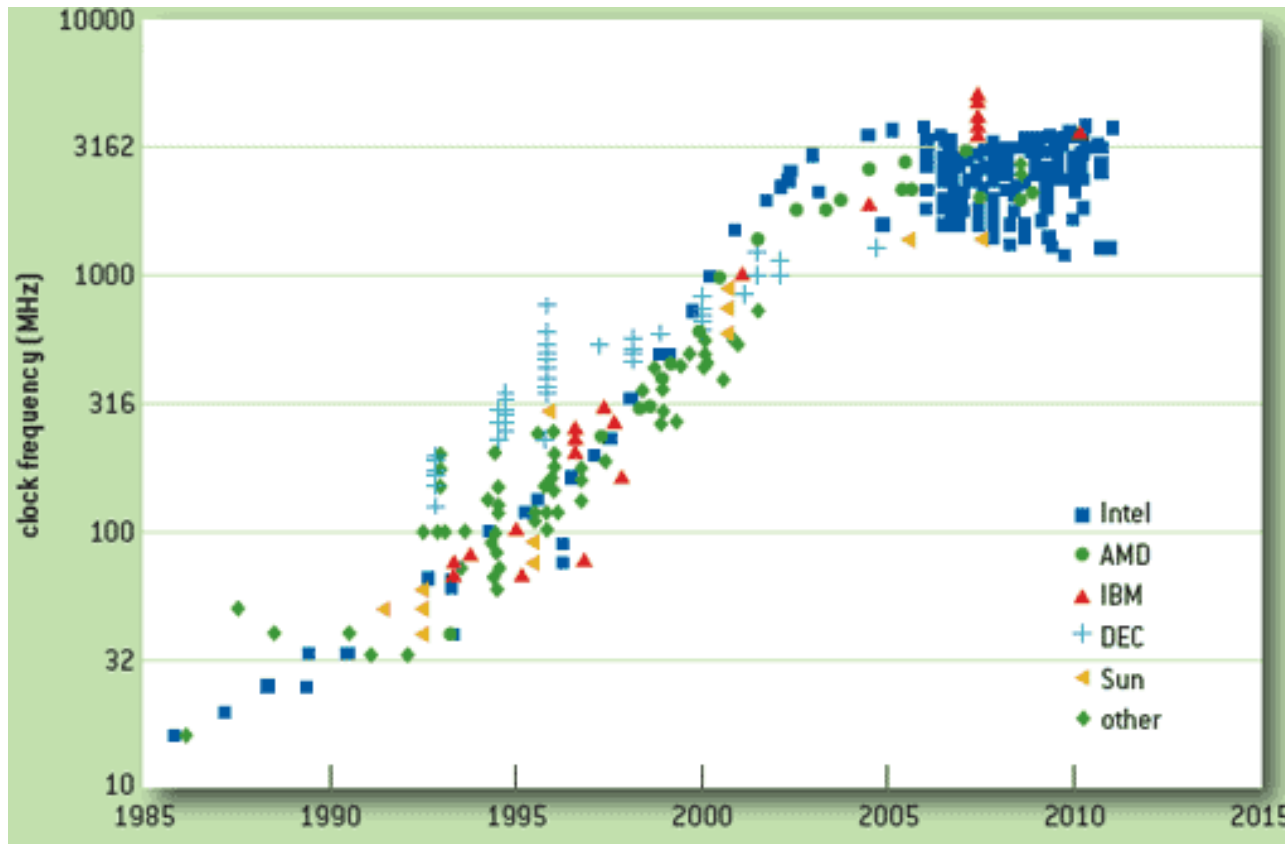


Intérieur du Pentium IV (Intel)



- année 2003
- ~40 millions de transistors
- Gravure à 0,09 μm
- 2 GHz

Augmentation en fréquence, une des formulations de la loi de Moore



[CPU DB: Recording Microprocessor History, A. Danowitz et al., 2012]

Résumé

**Mécanisme
de l'arithmétique**

Pascal

Leibnitz

Automates

Falcon

Jacquard

**Logique
mathématique**

Leibnitz

Boole

Turing

Technologie

Mécanique

Electromécanique

Electronique

Babbage

ENIAC

Von Neumann

Wilkes

Industrie Informatique

