



Master 1 - SEM

Processeurs Embarqués

Cours 3.1

Éléments de conception de processeur

Année 2022-2023

Pr. R. BOUDOUR



Principes de conception de processeur

2

1. Rendre rapide le cas courant

- Loi d'Amdahl : elle définit l'accélération par l'utilisation d'une caractéristique particulière

$$A_{\text{globale}} = 1 / (1 - f_{Am} + (f_{Am} / A_{am}))$$

- Exemple 1 : Soit un dispositif d'amélioration 20 fois plus rapide que la machine de base, mais que l'on ne peut utiliser que 60% du temps. Quelle est l'accélération apportée en introduisant ce dispositif ?

Principes de conception de processeur

3

- Fraction_{Am} = 0.6
- Accélération_{Am} = 20

$$\text{Accélération}_{\text{totale}} = 1 / (0.4 + (0.6 / 20)) = 1 / 0.43 \\ \approx 2.33$$

La loi d'Amdahl :

- sert de guide pour calculer l'impact d'un dispositif sur les performances
- sert à comparer deux alternatives de conception

Principes de conception de processeur

4

- Exemple 2 : Supposons qu'un cache soit 5 fois plus rapide que la mémoire centrale, et que ce cache puisse être utilisé durant 90% du temps. Quel est le gain de vitesse lié à l'utilisation du cache ?
- Il s'agit de la simple utilisation de la loi d'Amdahl :

$$f = \% \text{ temps cache utilisé} = 0.9$$

$$A_{\text{am}} \text{ accélération cache} = 5$$

$$\begin{aligned} A_{\text{globale}} &= 1 / (1 - f_{A_{\text{am}}} + (f_{A_{\text{am}}} / A_{\text{am}})) \\ &= 1 / (1 - 0.9) + (0.9 / 5) \\ &= 3.57 \end{aligned}$$

Le gain de vitesse lié au cache est de 3.57 fois

Principes de conception de processeur

5

2. Optimiser l'équation de performance

- $\text{Temps UC} = \text{NI} \times \text{CPI} \times T$; $\text{Perf} = 1/\text{Temps UC}$
ou $\text{Temps UC} = \text{NI} \times \text{CPI} \times 1/\text{Fréquence d'horloge}$
- *Fréquence d'horloge : technologie matérielle et structure*
- *CPI : Structure et architecture du jeu d'instructions*
- *Nombre d'instructions : Architecture du jeu d'instructions et technologie des compilateurs*

Principes de conception de processeur

6

- **Exemple 1** : Si la machine A exécute un programme en 10 secondes et la machine B exécute le même programme en 15 secondes.

De combien A est-elle plus rapide que B ?

- **Nous savons que A est n fois plus rapide que B si :** $\text{Performances A} / \text{Performances B} = n$
ou $\text{Temps d'exécution B} / \text{temps d'exécution A} = n$

Le rapport est donc de $15/10 = 1.5$

La machine A est 1.5 fois plus rapide que B

Principes de conception de processeur

7

□ **Exemple 2** : Un programme s'exécute en 10 secondes sur A, qui dispose d'une horloge à 100 Mhz. Un concepteur tente de construire une machine B, qui exécutera ce programme en 6 secondes.

Le concepteur établit qu'une augmentation de la fréquence d'horloge est possible, mais que cette augmentation affectera le reste de la conception de l'UC, imposant à la machine B d'utiliser 1,2 fois plus de cycles d'horloge que la machine A pour ce programme.

Quel objectif de la fréquence d'horloge devons-nous donner au concepteur ?

Principes de conception de processeur

8

- Le nombre de cycles d'horloge nécessaires au programme sur A :
 - $\text{Temps}_{UC A} = \text{Nbre cycles}_{UC A} / F_{\text{Horloge A}} \quad (\text{Cycle} = T = 1/F)$
 $10 = \text{Nbre cycles}_{UC A} / 100 \times 10^6 \text{ cycles/s}$
 $\text{Nbre cycles}_{UC A} = 10 \times 100 \times 10^6 = 1000 \times 10^6 \text{ cycles}$
 - Le temps UC pour B peut être obtenu en utilisant l'équation suivante :
 $\text{Temps}_{UC B} = 1.2 \times 1000 \times 10^6 \text{ cycles} / F_{\text{Horloge B}}$
 $F_{\text{Horloge B}} = 1.2 \times 1000 \times 10^6 \text{ cycles} / 6 \text{ s} = 200 \text{ Mhz}$
Ainsi cet objectif devrait être atteint avec $F = 200 \text{ MHz}$

Principes de conception de processeur

9

3. Localité des références

- ❑ Les programmes ont tendance à réutiliser les données et les instructions qu'ils ont utilisé récemment
- ❑ **Règle empirique** : Un programme passe 90% de son temps d'exécution sur seulement 10% des instructions.
La localité des références s'applique aux accès aux données, mais moins fortement aux accès au code.



Hiérarchie mémoire

Principes de conception de processeur

10

- **Localité spatiale** : le code d'un programme s'exécute toujours à l'intérieur de petites zones répétées de mémoire (des blocs correspondant à des boucles ou/et des sous-programmes)
- **En d'autres termes**, c'est la probabilité d'accès à une adresse voisine : Si un élément est référencé, les éléments dont les adresses sont voisines auront tendance à être référencés bientôt.
- **Localité temporelle** : les blocs s'exécutent en séquences très proches (il y a plus de chances d'accéder à une position de mémoire utilisée il y a 10 cycles qu'à une autre utilisée il y a 10000 cycles)
En d'autres termes, c'est la probabilité d'accéder aux mêmes adresses successivement. Si un élément est référencé, il aura tendance à être référencé bientôt de nouveau.

Principes de conception de processeur

11

● Les données

```
for (i=0; i<N; i++) {  
  for (j=0; j<N; j++) {  
    y[i] = y[i] + a[i][j] * x[j]  
  }  
}
```

- **y[i]**: propriétés de localités temporelle et spatiale.
- **a[i][j]**: propriété de localité spatiale.
- **x[j]**: propriété de localité temporelle et spatiale.

● Le programme

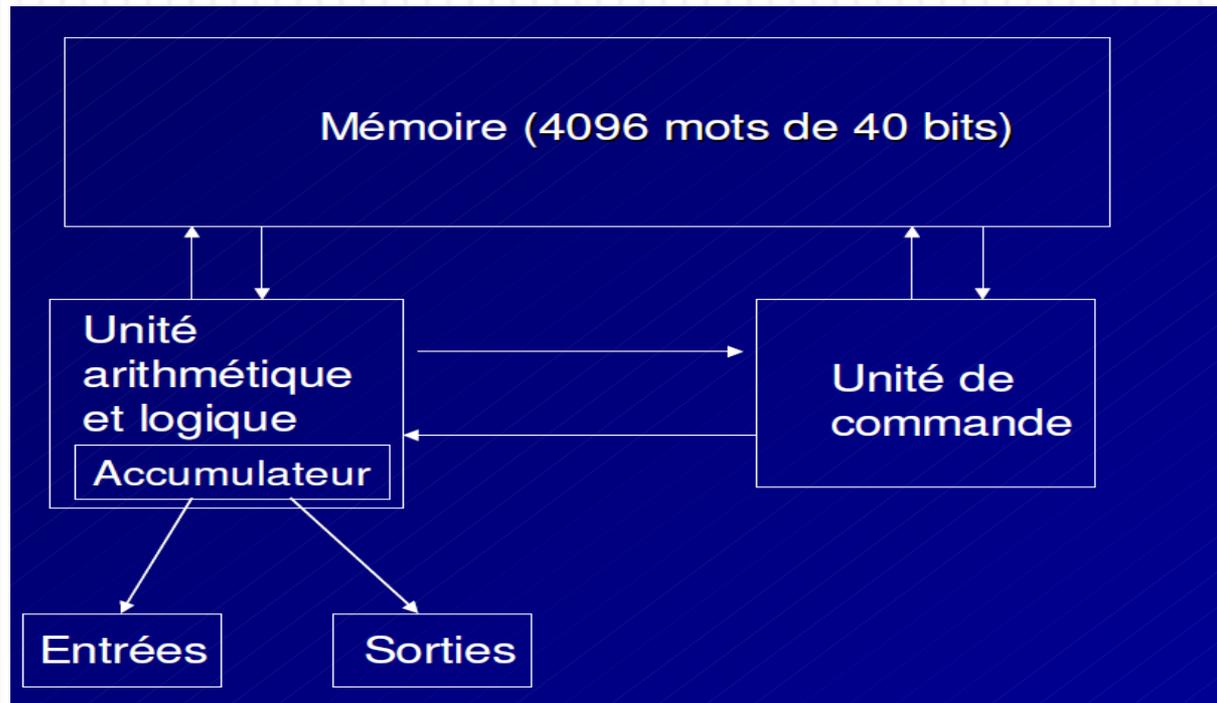
```
...  
05 LOOP      LDR R1, R0, #3  
06           ADD R1, R1, #5  
07           STR R1, R0, #30  
08           ADD R0, R0, #1  
09           ADD R3, R0, R2  
0A           BRn LOOP  
...
```

Boucle : réutilisation des instructions :
localité temporelle

Instructions consécutives en mémoire :
localité spatiale

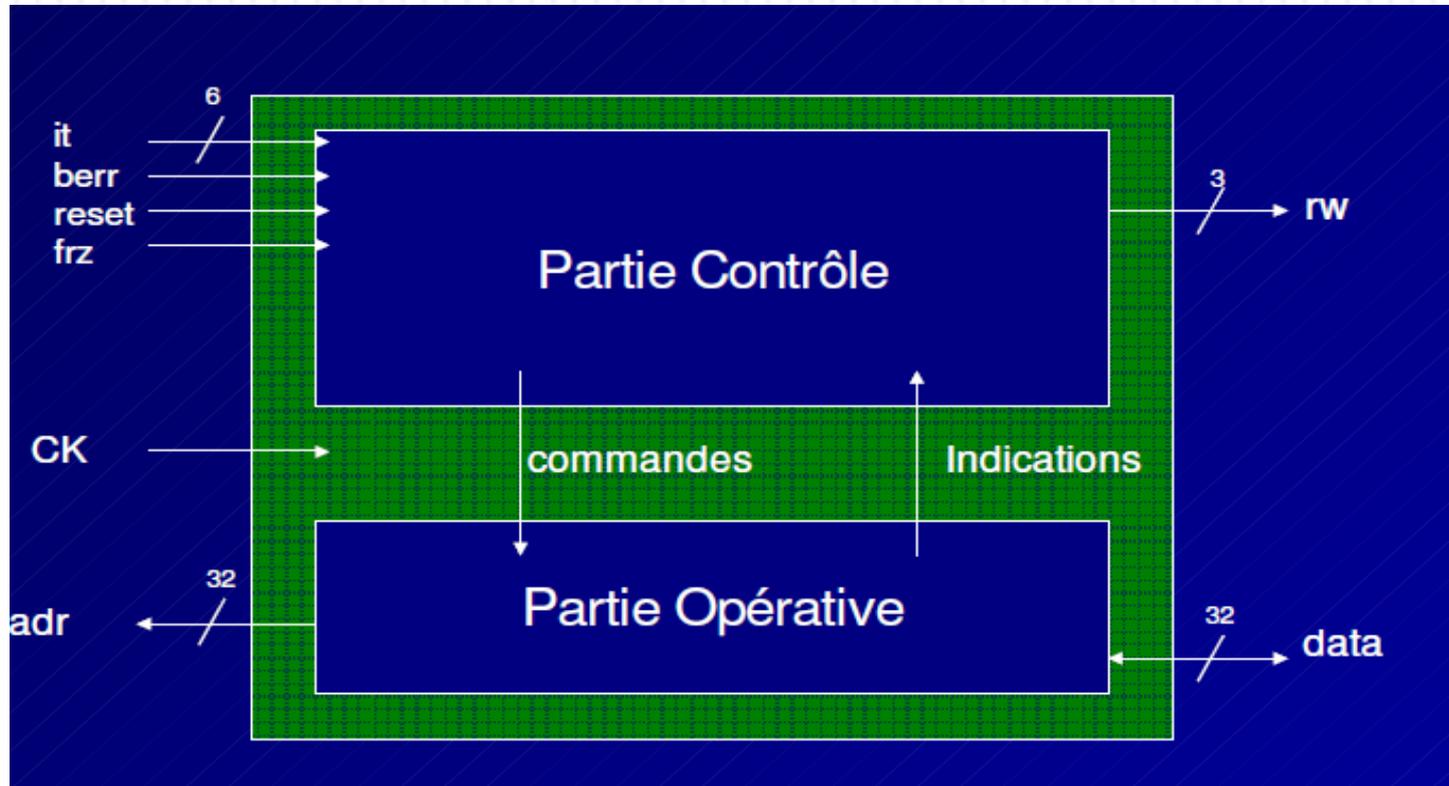
Histoire : Architecture de Von Neumann (1952)

12



Parties principales d'un processeur

13



MIPS R3000

Parties principales d'un processeur

14

Architecture composée de 2 parties

■ Le chemin de donnée réalise les traitements

1. Unité de transfert mémoire

- Registre d'adresse de données
- Registre d'adresse d'instructions ou Compteur Ordinal (CO) ou PC

2. Registres de travail (architecture à Accumulateur, à Pile, à banc de registres)

3. L'ALU

■ Le contrôleur

1. Décodeur

- Registre de transfert des instructions
- Registre d'instruction

2. Séquenceur

- Positionne les commandes à chaque cycle d'horloge

04/09/07

Rappel : Architecture externe

15

- ❑ Une **architecture externe de processeur** ou **architecture de jeu d'instructions (ISA, de l'anglais *Instruction Set Architecture*)**, ou tout simplement **architecture de processeur**
- ❑ C'est la spécification fonctionnelle d'un processeur, du point de vue du programmeur en langage machine.
- ❑ Elle s'intéresse à la conception de ce jeu d'instructions, notamment :
 - ❑ le modèle de programmation : registres généraux,
 - ❑ le nombre de registres utilisés ou visibles par le programmeur,
 - ❑ les méthodes d'accès aux données,
 - ❑ etc.

Notion de microarchitecture

16

Les choix d'organisation du chemin de donnée forment ce que l'on appelle la **microarchitecture** du processeur :

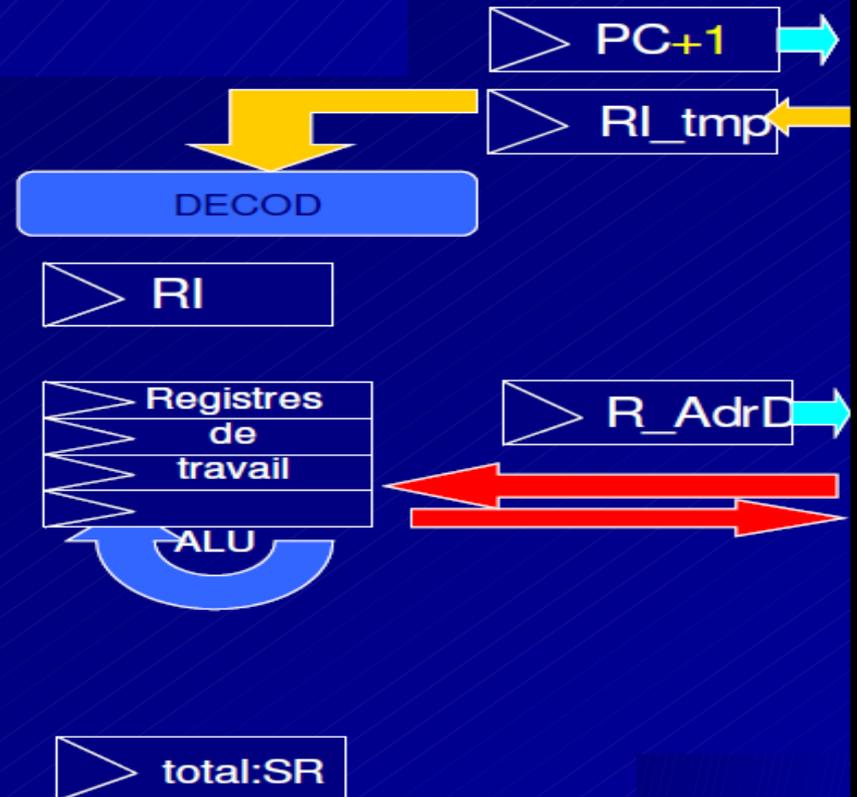
- Le CdD dispose de 2 ou 3 bus
- Le nombre de registres
- L'architecture dispose d'un pipeline
- Elle est superscalaire
- Son contrôle est microprogrammé ou câblé
- ...

- La microarchitecture s'intéresse à l'implémentation pratique du comportement spécifié par une architecture externe (conception).
- Une architecture externe (une conception) donnée peut être réalisée sous forme de plusieurs microarchitectures (plusieurs implémentations).
exemple : PC 8086 et 8088 de Intel.

Cycle d'exécution machine

17

- Charger l'instruction
- Incrémenter PC
- Décoder l'instruction
- Charger les données
- Exécuter l'opération
- Ranger le résultat
- Retour



Résumé

18

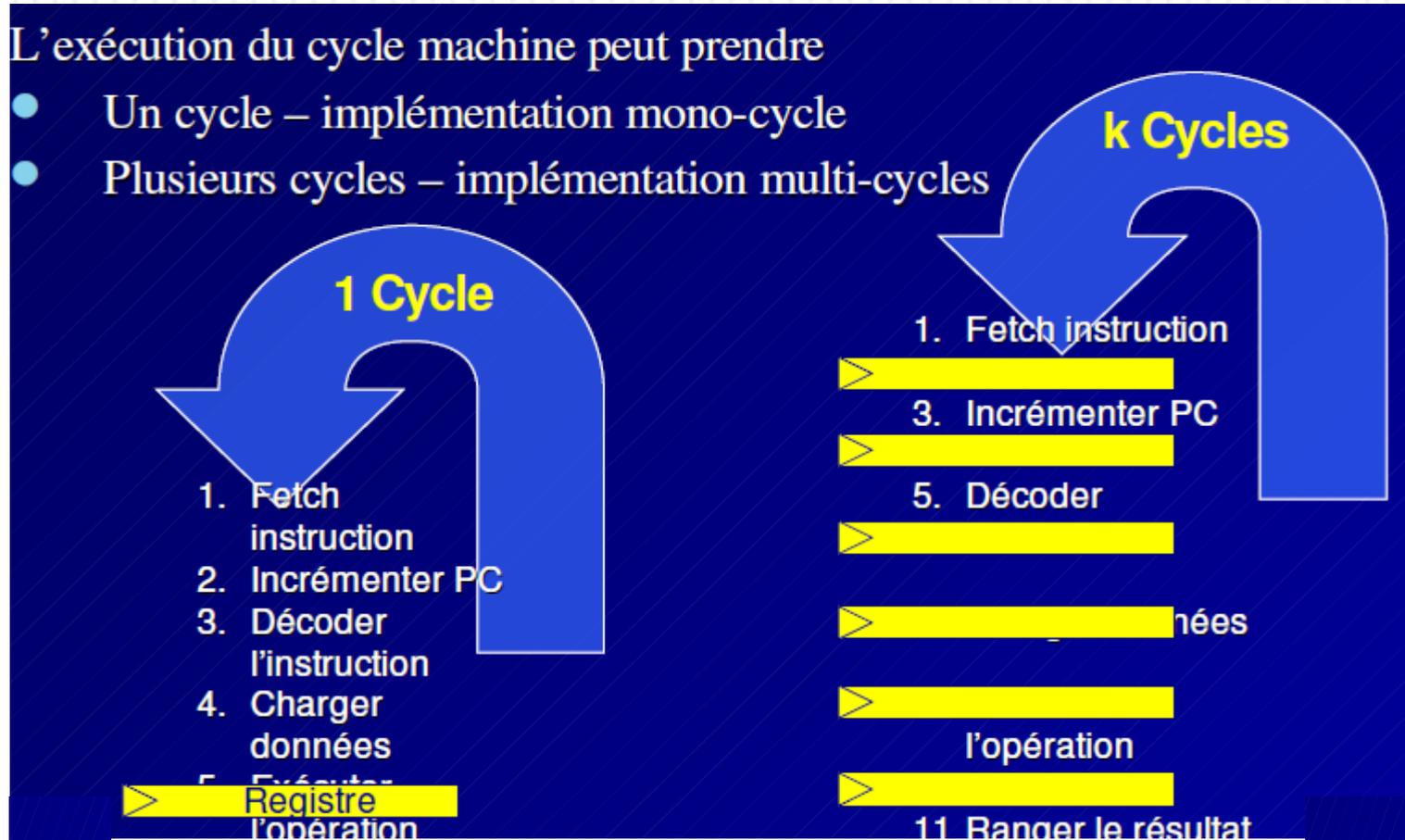
- Une architecture de Von Neumann est constituée de 5 unités:
 1. de Calcul
 2. de Mémoire
 3. de Registres
 4. Séquenceur
 5. Décodeur
- Elle utilise les 2 registres principaux suivants :
 - PC
 - RI
- Elle s'exécute en 7 étapes :
 1. Fetch instruction
 2. Incrémenter PC
 3. Décoder l'instruction
 4. Charger données
 5. Exécuter l'opération
 6. Ranger le résultat
 7. Retour
- Les échanges avec la mémoire sont de 3 sortes :
 - Données
 - Instructions
 - Adresses

Choix d'implémentation temporelle

19

L'exécution du cycle machine peut prendre

- Un cycle – implémentation mono-cycle
- Plusieurs cycles – implémentation multi-cycles



Quelles différences ?

20

Supposons les informations suivantes :

Étape d'exécution	Temps d'exécution
Fetch instruction	2ns
Incrémenter PC	1ns
Décoder l'instruction	1,5ns
Charger données	2ns
Exécuter l'opération	1,8ns
Ranger le résultat	1,7ns
Retour	-

Quelles différences monocycle – multicycle ?

21

- La version mono-cycle est-elle plus rapide de la version multi-cycles ?
- Quel est le temps de cycle de l'horloge pour la version mono-cycle ? Sa fréquence ?
- Quel est le temps de cycle de l'horloge pour la version multi-cycle ? Sa fréquence ?
- Quel est le temps d'exécution de l'instruction en mono-cycle?
- Quel est le temps d'exécution de l'instruction en multi-cycle ?

Quelles différences monocycle – multicycle ?

22

- La version mono-cycle est-elle plus rapide de la version multi-cycles ?
 - ...
- Quel est le temps de cycle de l'horloge pour la version mono-cycle ? Sa fréquence ?
 - $P = 10 \text{ ns}$, $F = 100 \text{ MHz}$
- Quel est le temps de cycle de l'horloge pour la version multi-cycle ? Sa fréquence ?
 - $P = 2 \text{ ns}$, $F = 500 \text{ MHz}$
- Quel est le temps d'exécution de l'instruction en mono-cycle ?
 - $T = 10 \text{ ns}$
- Quel est le temps d'exécution de l'instruction en multi-cycle ?
 - $T = 2 * 6 = 12 \text{ ns}$

Contrôle des signaux de commandes

23

- Les composants de l'architecture sont contrôlés par un ensemble de signaux de commande
- Une micro-instruction représente une configuration complète de tous les signaux de commande de l'architecture à un instant donné
- Une instruction est donc une séquence de micro-instructions en version multi-cycles
- Le **séquenceur** définit l'état de chaque signal de commande à tout instant de l'exécution en fonction de l'instruction reçue en entrée

Le séquenceur fonction des instructions

24

- Les micro-instructions peuvent être séparées en plusieurs étapes appartenant au cycle d'exécution machine
 - Fetch
 - Decode
 - Execute
 - ...
- La phase de Fetch est toujours la même quelque soit l'instruction
- Dès que l'on connaît l'instruction à exécuter, les micro-instructions sont différentes

Le séquenceur fonction de l'architecture

25

- L'architecture définit le nombre et la nature des micro-commandes
- Elle a donc un impact direct sur la structure des micro-instructions et sur leur séquencement

Implantation du séquenceur

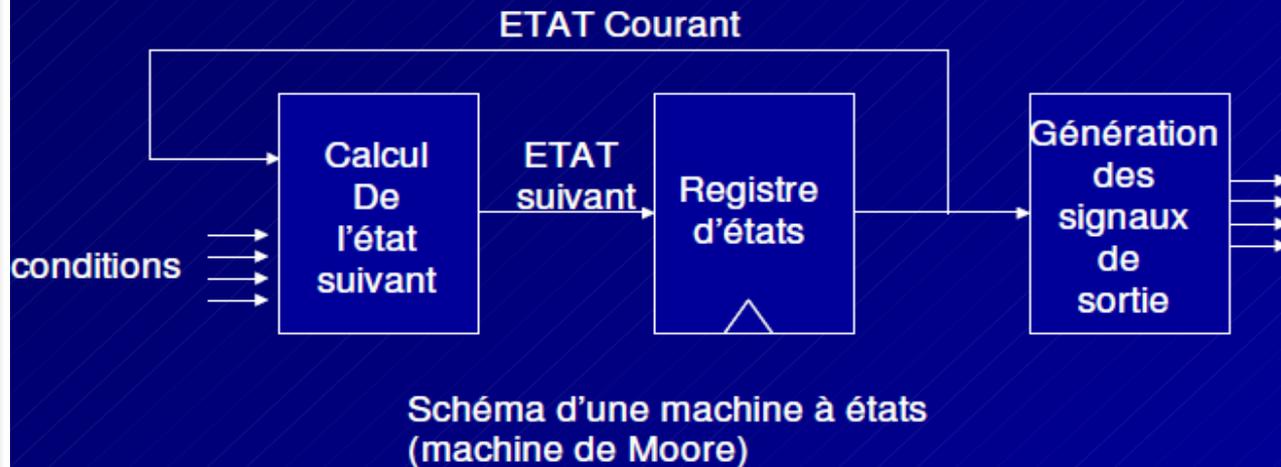
26

- (En multicycles) Le séquenceur est un **automate** distribuant, selon un chronogramme précis, les signaux de commande aux diverses unités participant à l'exécution d'une instruction.
- Deux implantations possibles
 - Séquenceur câblé
 - Séquenceur micro-programmé

Le séquenceur (contrôleur câblé)

27

Un séquenceur câblé est un circuit séquentiel complexe comprenant un sous-circuit pour chacune des instructions à commander. Ce sous-circuit est activé par le décodeur.



Choix d'architecture

28

Contrôleur câblé

■ Avantages

- Exécution rapide
- Instructions simples

■ Inconvénients

- Jeu d'instruction fixé
- Pas d'évolutivité
- Complexe à réaliser

Séquenceur microprogrammé

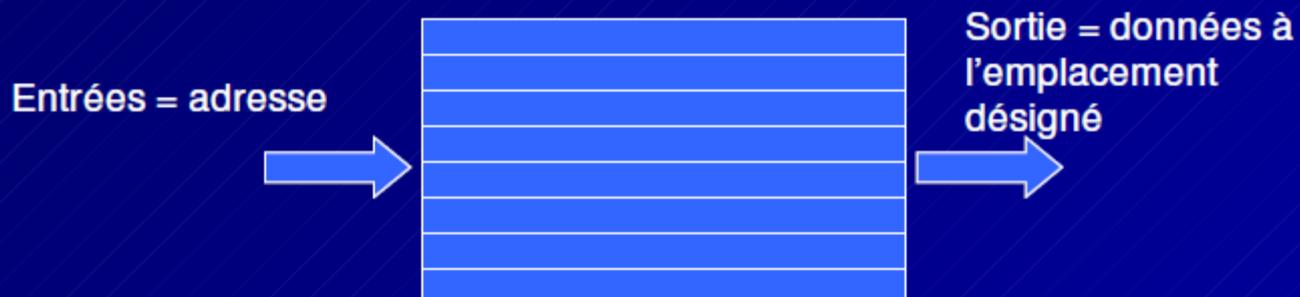
29

- Concept de microprogrammation introduit par M. Wilkes en 1951
- Premier processeur à avoir utilisé un micro-programme : le motorola 68000
- Principe :
 - Remplacer le circuit câblé par une mémoire
 - Chaque configuration des commandes est définie comme une micro-instruction
 - L'exécution d'une instruction machine nécessite un micro-programme

Réalisations de circuits logiques combinatoires

30

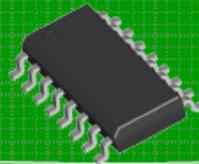
- Un système logique combinatoire peut être réalisé
 - à base de portes logiques discrètes,
 - d'un circuit PLA (Programable Logic Array)
 - ou encore, puisque l'état des sorties ne dépend que de l'état des entrées, d'une mémoire morte (ROM)



Réalisation de circuit

31

Le circuit se construit autour d'une bibliothèque de portes/composants



Portes

■ INV_

■ OR_

■ AND_

■ NAND_

■ NOR_

■ XOR_

■ latchD

MSI

■ MAJ_

■ MUX_

■ DEC_

■ COMP_

■ HalfAdder_

■ ADDER

■ Registre_

LSI

■ PLA_

■ ALU_

■ RAM_

Séquenceur microprogrammé

32

Séquenceur microprogrammé

■ Avantages

- Simplicité de réalisation
- Flexibilité
- Jeu d'instruction plus complexe
- Moins d'accès mémoire pour le transfert des instructions

■ Inconvénients

- Lenteur d'exécution
- 1 micro-programme / instruction (uPC)

Exemples de séquenceurs

33

- Les architectures RISC, séquenceur câblé
 - Exemple de la machine MIPS R3000
- Les architectures CISC, séquenceur micro-programmé
 - Exemple de la machine PicoJava

Résumé

34

- L'exécution d'un programme se fait par **traductions** (interprétation) successives depuis le langage de haut niveau jusqu'au niveau physique
- Les ordinateurs actuels (embarqués ou non) sont des **machines multi-couches**

Résumé

35

- L'architecture d'un processeur est composé de plusieurs **composants** électroniques combinatoires ou séquentiels
 - L'ALU
 - Le décodeur
 - Le Contrôleur
- Le programme utilise la ressource du processeur et le pilote par l'intermédiaire de **registres** (PC, RI, Adr, ...)

Questions

36

Q1. Qu'est ce qu'une ISA ?

Q2. En quoi diffère les processeurs Intel 8086 et 8088 ?

Q3. Qu'est-ce-qu'une microinstruction ?

Q4. Comparer les séquenceurs câblé et microprogrammé

Q5. Citer deux processeurs avec contrôle câblé et contrôle microprogrammé.

Réponse 1

37

- Le **jeu d'instructions** est l'ensemble des instructions machines qu'un processeur d'ordinateur peut exécuter. Ces instructions machines permettent d'effectuer des opérations élémentaires (addition, ET logique ...) ou plus complexes (division, passage en mode basse consommation...). Le jeu d'instructions définit quelles sont les instructions supportées par le processeur. Le jeu d'instructions précise aussi quels sont les registres du processeur manipulables par le programmeur (les registres architecturaux).

Réponse 1

38

- An **instruction set architecture (ISA)** is an abstract model of a computer. It is also referred to as **architecture** or **computer architecture**. A realization of an ISA is called an *implementation*. An ISA permits multiple implementations that may vary in performance, physical size, and monetary cost (among other things); because the ISA serves as the interface between software and hardware. Software that has been written for an ISA can run on different implementations of the same ISA. This has enabled binary compatibility between different generations of computers to be easily achieved, and the development of computer families. Both of these developments have helped to lower the cost of computers and to increase their applicability. For these reasons, the ISA is one of the most important abstractions in computing today.

Réponse 2

39

❑ Différences : processeurs Intel 8086 - 8088

Microprocesseur 8086	Microprocesseur 8088
8086 est un microprocesseur 16 bits.	8088 est un microprocesseur 16 bits.
Le bus de données est de 16 bits.	Le bus de données est de 8 bits.
Il a une ALU de 16 bits.	Il a une ALU de 16 bits.
8086 nécessite de mémoire(memory banking) pour transférer des données 16 bits à la fois.	8088 ne nécessite pas de mémoire (memory banking) car il possède un bus de données 8 bits.
8086 effectue des opérations mémoire plus rapides car il peut transférer 16 bits en un cycle.	8088 effectue des opérations mémoire plus lentes car il ne peut transférer que 8 bits en un cycle.
8086 prend en charge l'architecture de pipeline.	8088 prend en charge l'architecture de pipeline.
8086 dispose d'une file d'attente de prélecture de 6 octets pour le traitement en pipeline.	8088 a une file d'attente de prélecture de 4 octets pour le traitement en pipeline.
8086 a 9 drapeaux/flags.	8088 a 9 drapeaux/flags.
Vitesses d'horloge: 5MHz, 8MHz, 10MHz	Vitesses d'horloge: 5MHz ,8MHz

Thanks!