Architecture des Ordinateurs
Partie II : Microprocesseur

1. Mémoire et Entrées/Sorties

David Simplot
simplot@fil.univ-lille1.fr

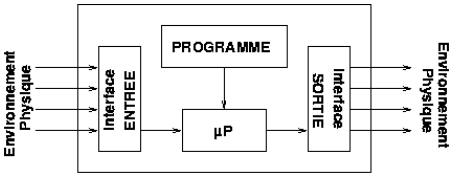
Objectifs

- Comprendre les mécanismes de fonctionnement du microprocesseur
  - comment fonctionne la mémoire,
  - comment fonctionnent les entrées/sorties

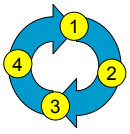
Au sommaire...

- **Machine à base de µP**
- Mémoire
- Entrées/Sorties

Machine à µP (1/2)



Machine à µP (2/2)



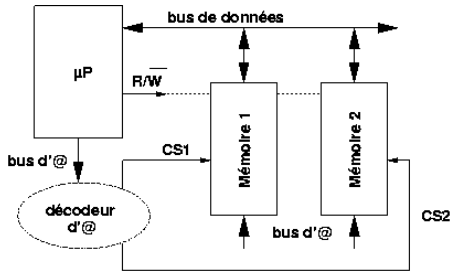
- 1 – Lire l'instruction
- 2 – Décoder l'instruction
- 3 – Exécuter l'instruction
- 4 – Préparer l'instruction suivante

- Performance du µP dépend
  - Nombre de cycle d'horloge moyen pour faire 1 à 4
  - Vitesse de l'horloge
  - Compacité du code
- Deux approches :
  - CISC ou RISC

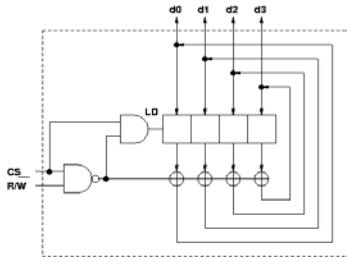
Au sommaire...

- Machine à base de µP
- **Mémoire**
- Entrées/Sorties

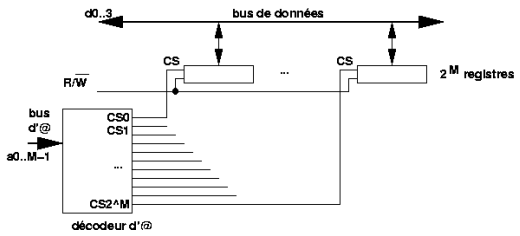
Mémoire (1/7)
Principe de fonctionnement



Mémoire (2/7)
Pour une case mémoire



Mémoire (3/7)
Cas mémoire 2 mots et 2^M mots



Mémoire (4/7)
Assemblage de mémoire

- Espace d'adressage du µP
  - Déterminé par la taille du bus d'adresses
    - 10 bits → 2^{10} mots = 1024 mots = 1 **Kilo**mot
    - 16 bits → 2^{16} mots = 65536 mots = 64 **Kilo**mot
    - 20 bits → 2^{20} mots = 1024x1Kmot = 1 **Méga**mot
    - 30 bits → 2^{30} mots = 1024x1Mmot = 1 **Giga**mot
    - 32 bits → 2^{32} mots = 4x1Gmot = 4 **Giga**mot
  - On découpe l'espace d'adressage en « plages mémoire ».

Mémoire (5/7)
Assemblages de mémoire (suite)

- Ex. µP 8 bits (bus de données) et 23 bits d'adressage
  - → soit 8 Mo
  - On peut découper en plages de 1 Mo

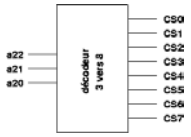
7FFFFF 700000
…
3FFFFF 300000
2FFFFF 200000
1FFFFF 100000
0FFFFF 000000

23 bits : **010** 0010 1101 1001 0110 0001

a22 … a0

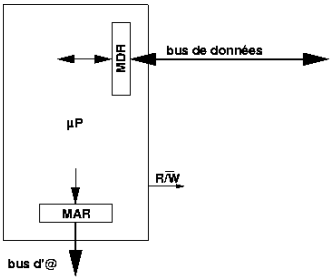
Mémoire (6/7)
Assemblage de mémoire (suite)

- On utilise un décodeur 3 vers 8 avec a20..22



- Ex. 2 mémoires 1 Mo, 1 mémoire de 2 Mo, 1 mémoire de 512 Ko

Mémoire (7/7)
Côté µP

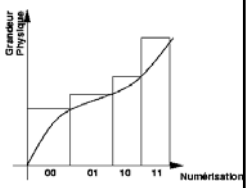


Au sommaire...

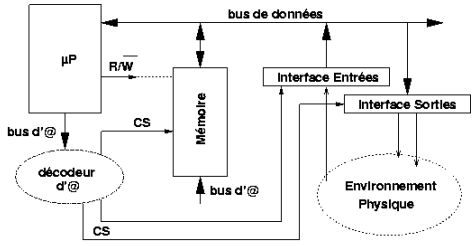
- Machine à base de µP
- Mémoire
- **Entrées/Sorties**

Entrées/Sorties (1/2)

- Sont accédées via l'adresse mémoire :
  - Certaines adresses sont réservées (à la conception de l'ordinateur ou de manière dynamique) aux entrées/sorties
  - Digital-Digital
    - Série ou parallèle
    - Entrées ou sorties
  - Digital-analogue
    - Sorties
  - Analogue-digital
    - entrées



Entrées/Sorties (2/2)



Architecture des Ordinateurs
Partie II : Microprocesseur

2. Instructions machines

David Simplot
simplot@fil.univ-lille1.fr

Objectifs

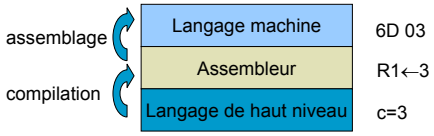
- Voir les instructions élémentaires du microprocesseur
- Comment on les réalise à l'intérieur du µP…

Au sommaire...

- **Instructions**
- Sous-routines
- INT/DMA
- Microprogrammation

Instructions (1/14)

- Programmes = suite d'instructions



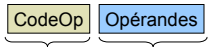
- Trois types d'instructions
  - De rangement
  - De calcul
  - De branchement

Instructions (2/14)
Exemples d'instructions « machine »

R1←4	MOV R1,4 LD R1,4	Mettre la valeur 4 dans le registre R1
R1←R2	MOV R1,R2 LD R1,R2	Mettre la valeur de R2 dans le registre R1
R1←MEM(1515)	MOV R1,[1515]	Copie la valeur stockée en mémoire à l'@ 1515 dans R1
R1←R1+2	ADD R1,2	Additionne 2 à la valeur de R1 et range dans R1
R1←R1+1	INC R1	Incrémente de 1 la valeur de R1
R2←R2-R4	SUB R2,R4	Soustraie à R2 la valeur de R4 et range dans R2

Instructions (3/14)
Codage des instructions

- Structure :



1 mot 0 ou plusieurs mots

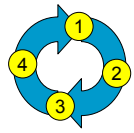
- La plupart du temps, certaines opérandes sont implicitement données dans le code opération. Par exemple :
  - MOV R1, 4 2 mots = 89 04
  - ADD R2, 3 2 mots = B2 03

Instructions (4/14)
Codage des instructions (suite)

- MOV R1, 4
  - 89 04
  - 1000 1001 0000 0100
  - 10001 001 0000 0100
  - MOV R1 4
- ADD R2, 3
  - B2 03
  - 1011 0010 0000 0011
  - 10110 010 0000 0011
  - ADD R2 3

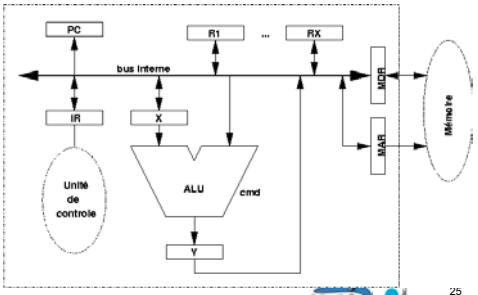
Instructions (5/14)
Exemple d'exécution

- Mémoire :
  - 1515 = 89
  - 1516 = 04
  - 1517 = B2
  - 1518 = 03



1-Lire l'instruction
2-Décoder
3-Exécuter
4-Préparer l'instruction suivante

Instructions (6/14)
Exemple d'exécution (suite)



Instructions (7/14)
Exemple d'exécution (suite)

- Phase 1 – Lire l'instruction
  - PC vaut 1515
  - Mettre PC dans MAR et donner l'ordre de lecture
    - PCout – LDMAR
    - Read – WaitMemory
  - Placer la valeur de MDR dans IR pour que l'instruction soit décodée
    - MDRout – LDIR
  - Pb. Lors de la phase 3, on va avoir besoin de récupérer les paramètres qui sont à PC+1
    - → on anticipe

Instructions (8/14)
Exemple d'exécution (suite)

- Phase 1 (suite)
  - On profite du fait que la lecture en mémoire est lente pour incrémenter PC
  - Dès que l'on est en phase 3, PC pointe vers le premier argument
  - Nb. S'il n'y a pas d'arguments, PC pointe vers l'instruction suivante.
  - « code » réel de la phase 1 :
    - PCout – LDMAR – LDX
    - Read – INCX – LDY
    - Yout – LDPC – WaitMemory
    - MDRout - LDIR

Instructions (9/14)
Exemple d'exécution (suite)

- Phase 2 – décodage de l'instruction
  - Pris en charge par l'UC (unité de contrôle)
- Phase 3 – exécution
  - Deux sous-phases :
    - 3.1 Récupérer les arguments éventuellement
    - 3.2 Exécution
  - 3.1 lecture argument + incrémentation PC
    - PCout – LDMAR – LDX
    - Read – INCX – LDY
    - Yout – LDPC - WaitMemory

Instructions (10/14)
Exemple d'exécution (suite)

- Phase 3 (suite)
  - 3.1 lecture argument + incrémentation PC
    - PCout – LDMAR – LDX
    - Read – INCX – LDY
    - Yout – LDPC - WaitMemory
  - 3.2 exécuter l'instruction (ici MOV R1,4)
    - R1out – LDX
    - MDRout – ADD – LDY
    - Yout – LDR1
- Phase 4 (préparer l'instruction suivante)
  - Rien pour l'instant ☺