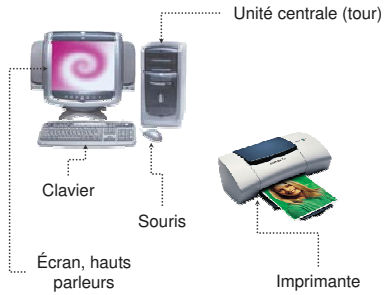
Architecture matérielle

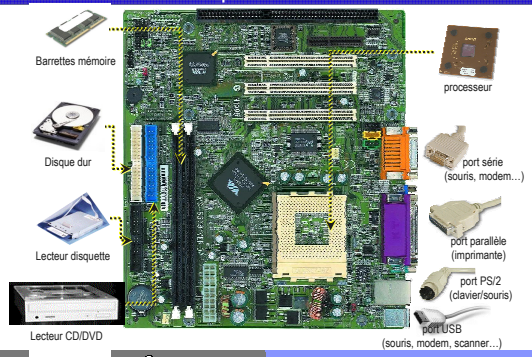
- PCI
- Permis de Conduire Informatique
- L1 1er semestre
- UFR d'Informatique

- Comment fonctionne un ordinateur ?
- Quels sont ses composants ?
- Comment et où l'information est stockée ?
- Comment s'exécute une instruction ?
- Comment une information passe de la mémoire au processeur, et réciproquement ?
- Comment sont gérées les communications avec l'extérieur ?

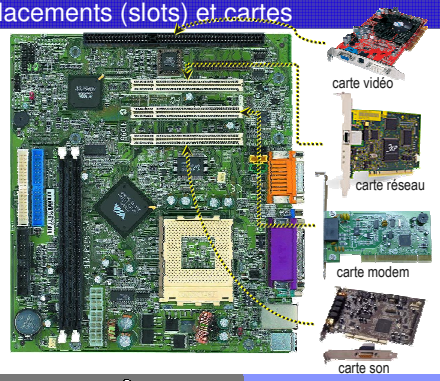
Un ordinateur type (et ses périphériques)



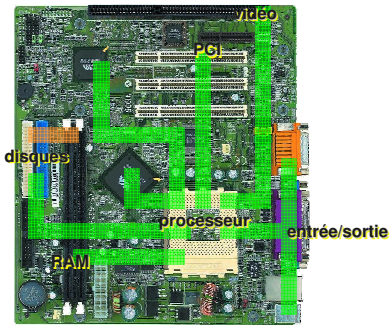
Branchement des composants sur la carte mère

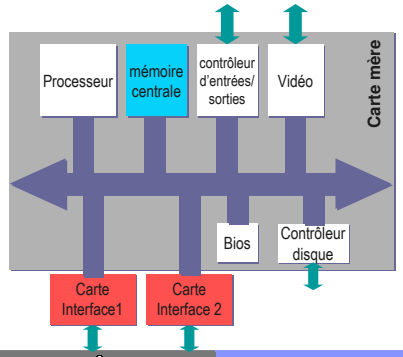


Emplacements (slots) et cartes



Le bus





Rôle

- Stocker les informations
  - données
  - calculs intermédiaires
  - programmes
- Récupérer les informations

Types de mémoire

- Mémoire vive : *RAM* (*random access memory*)
  - accessible en écriture et en lecture
  - volatile
- Mémoire morte : *ROM (read-only memory)*
  - accessible en lecture seule
  - stockée sur composants électroniques (puces)
  - persistante
- Remarque
  - la mémoire centrale est de la mémoire vive
  - le bios est une EPROM *(Erasable Programmable ROM)*

Stockage en mémoire

- Pour stocker l'information, il faut savoir la représenter
- Pour représenter en mémoire des données ou instructions, il faut les coder
  - exemples de codage
    - alphabets, chiffres (romains, arabes), cryptographie…
- Une fois codée, l'information est « stockée » en mémoire centrale

Codage

- La plus petite unité d'information
  - le « bit » (*binary digit*), 0 ou 1 (base 2)
- On groupe les bits par 8
  - octet (*byte*, by eight)
- On représente par une suite d'octets toute information
  - les nombres
  - les caractères
  - les instructions
  - les pixels d'une image
  - …

Codage binaire : puissances de 2

1	00 00 00 00	00 00 00 01
2	00 00 00 00	00 00 00 10
4	00 00 00 00	00 00 01 00
8	00 00 00 00	00 00 10 00
16	00 00 00 00	00 01 00 00
32	00 00 00 00	00 10 00 00
64	00 00 00 00	01 00 00 00
128	00 00 00 00	10 00 00 00
256	00 00 00 01	00 00 00 00
512	00 00 00 10	00 00 00 00
1024	00 00 01 00	00 00 00 00
1143	00 00 01 00	01 11 01 11

Codage ASCII

- Chaque caractère a un « code » unique
  - entier entre 0 et 255
  - Exemple
    - E 69
    - x 120
    - e 101
    - m 109
    - p 112
    - l 108
    - e 101

32		64	@	96	`	
33	!	65	A	97	a	
34	"	66	B	98	b	
35	#	67	C	99	c	
36	$	68	D	100	d	
37	%	69	E	101	e	
38	&	70	F	102	f	
39	'	71	G	103	g	
40	(	72	H	104	h	
41	)	73	I	105	i	
42	*	74	J	106	j	
43	+	75	K	107	k	
44	,	76	L	108	l	
45	-	77	M	109	m	
46	.	78	N	110	n	
47	/	79	O	111	o	
48	0	80	P	112	p	
49	1	81	Q	113	q	
50	2	82	R	114	r	
51	3	83	S	115	s	
52	4	84	T	116	t	
53	5	85	U	117	u	
54	6	86	V	118	v	
55	7	87	W	119	w	
56	8	88	X	120	x	
57	9	89	Y	121	y	
58	:	90	Z	122	z	
59	;	91	[	123	{	
60	<	92	\	124	|	
61	=	93	]	125	}	
62	>	94	^	126	~	
63	?	95	_	127	□	

Codage des caractères

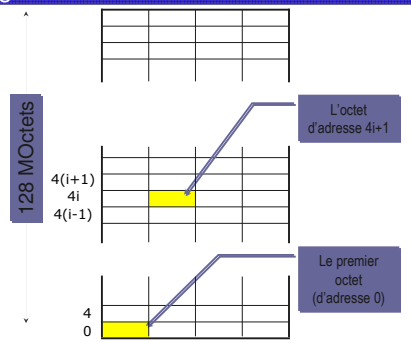
- Pour coder des caractères
  - on code le caractère en ASCII
  - on transforme le code ASCII en code binaire

caractère	Ascii	décomposition de l'Ascii	binaire
E	69	64 + 4 + 1	01 00 01 01
x	120	64 + 32 + 16 + 8	01 11 10 00
e	101	64 + 32 + 4 + 1	01 10 01 01
m	109	64 + 32 + 8 + 4 + 1	01 10 11 01
p	112	64 + 32 + 16	01 11 00 00
l	108	64 + 32 + 8 + 4	01 10 11 00
e	101	64 + 32 + 4 + 1	01 10 01 01
		2^6 2^5 2^4 2^3 2^2 2^1 2^0	

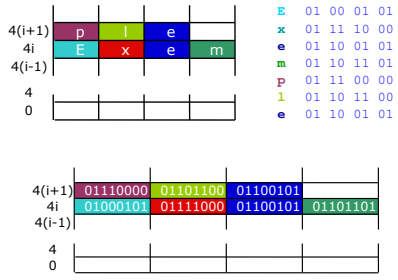
Qu'est-ce que l'adressage ?

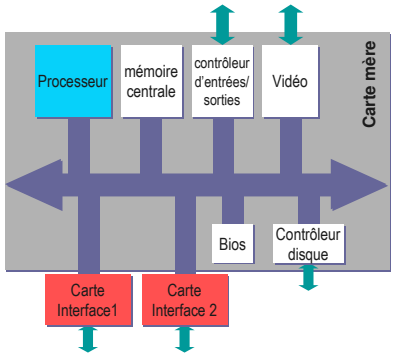
- Stocker l'information, c'est
  1. la coder
  2. la placer en mémoire : à quel endroit ?
- Récupérer l'information, c'est
  1. la prendre en mémoire : à quel endroit ?
  2. la décoder
- Comment désigner un emplacement en mémoire ?
  - il faut un système d'« adressage »

Adressage en mémoire

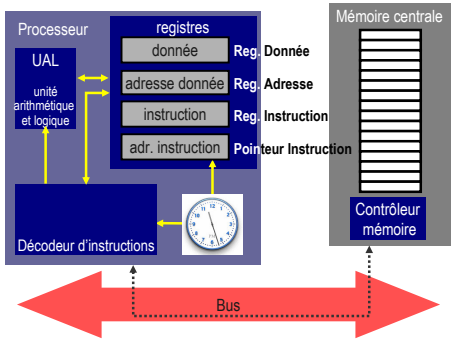


Exemple





Le processeur

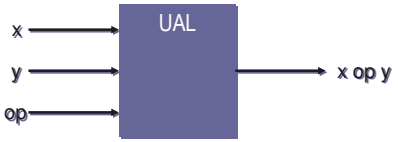


Le processeur

- Il se compose
  - de l'horloge
  - du décodeur d'instructions
  - de l'Unité Arithmétique et Logique
  - des registres
- Un cycle du processeur correspond à
  - charger la prochaine instruction
  - décoder cette instruction
  - exécuter cette instruction
- Un cycle est lancé par l'horloge
- Le processeur effectue un certain nombre de cycles par seconde (fréquence)

L'unité arithmétique et logique (UAL)

- Elle réalise les opérations arithmétiques (+ - × /) et logiques (et ou non) de base
- Elle dispose de
  - trois entrées (opération, opérande 1 et opérande 2)
  - une sortie (le résultat)
- Elle s'active quand on lui donne un ordre



Codage des programmes (1/2)

- Programme
  - suite d'instructions élémentaires
  - en mémoire, les instructions d'un programme sont stockées consécutivement
- Instruction
  - opération élémentaire réalisée par le processeur
  - codée sur n octets
    - code de l'opération
    - code des opérandes
  - le processeur sait décoder l'opération et les opérandes

Codage des programmes (2/2)

- Exemples
  - ajouter 32 au contenu du registre RD

    RD ← RD + 32

    ADD RD, 32

    0010 1100 00100000
  - stocker, dans le registre RD, le contenu de la mémoire, à l'adresse 71

    RD ← Mem [71]

    MV< RD , Mem [71]

    0011 1100 01000111
  - stocker en mémoire, à l'adresse 84, le contenu du registre RD

    RD → Mem [84]

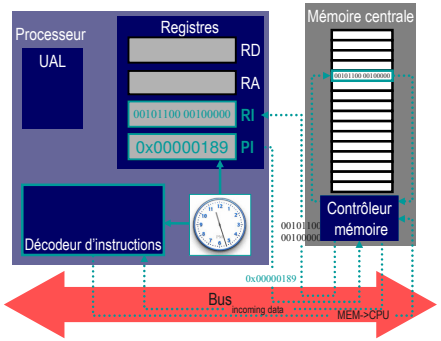
    MV> RD , Mem [84]

    0110 1100 01010100

Chargement d'une instruction (1/2)

- Rappel : en mémoire, il y a
  - des programmes (un ensemble d'instructions)
  - des données
- Avant le chargement d'une instruction
  - la « prochaine » instruction est en mémoire à l'adresse définie par le pointeur d'instruction
- Après le chargement
  - le code de l'instruction à exécuter est stocké dans le registre d'instruction
- Principe
  - demander au contrôleur mémoire de récupérer la bonne valeur et de la mettre au bon endroit

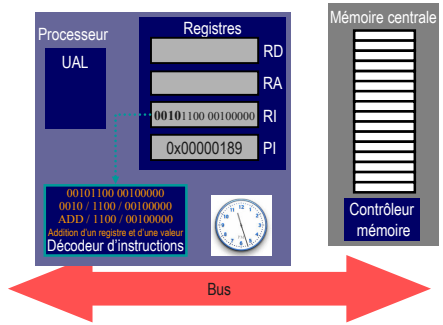
Chargement d'une instruction (2/2)



Décodage d'une instruction (1/2)

- Avant le décodage
  - le code de l'instruction à exécuter est stocké dans le registre d'instruction
- Après le décodage
  - le décodeur
    - connaît l'opération à réaliser
    - sait où se situent les opérandes
- Principe
  - dans le décodeur, le contenu du registre d'instruction est traité par un ensemble de composants électroniques (portes logiques)

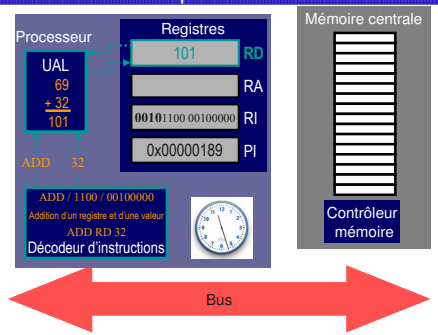
Décodage d'une instruction (2/2)



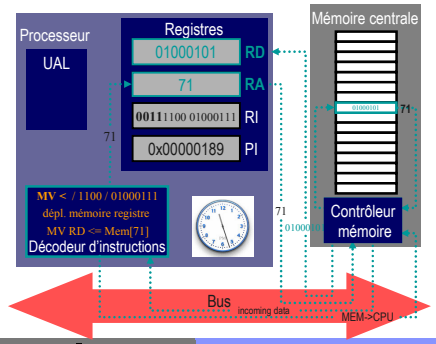
1. Les grandes lignes

- Avant l'exécution
  - le décodeur
    - connaît l'opération à réaliser
    - sait où se situent les opérandes
- Après l'exécution
  - les calculs/transferts sont réalisés sur les bons opérandes
  - le résultat a été récupéré au bon endroit
- Principe
  - récupérer les opérandes
  - transmettre les opérandes au composant qui va réaliser l'opération
  - indiquer à ce composant quoi faire

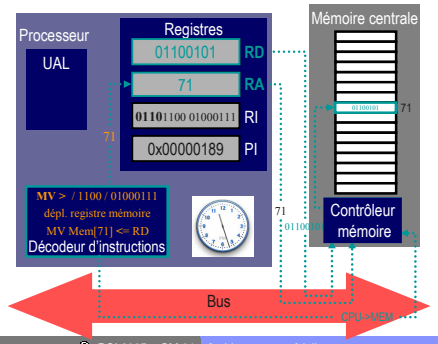
2. Opération arithmétique



3. Lecture depuis la mémoire



4. Écriture dans la mémoire



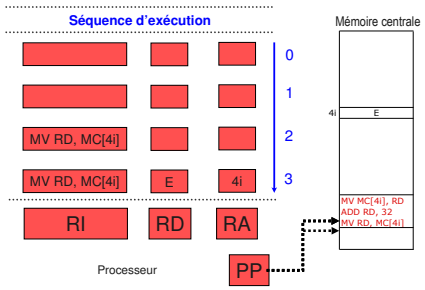
Contexte

- Ce transparent est en mémoire
  - il contient le mot « Exemple »
  - son premier caractère (E) est stocké à l'adresse 4i
- Programme
  - mettre la première lettre du mot « Exemple » en minuscule
- Rappel
  - code ascii pour E : 69
  - code ascii pour e : 101
- Passage de E à e : + 32

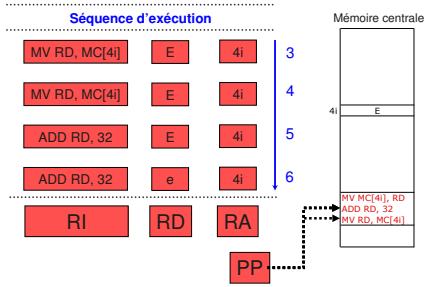
Programme à exécuter

- Récupérer la première lettre du mot (à l'adresse 4i)
- La convertir en minuscule
- Mettre la nouvelle « valeur » en mémoire au même endroit

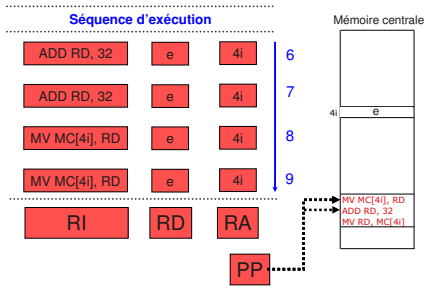
1. Chercher E en MC[4i]

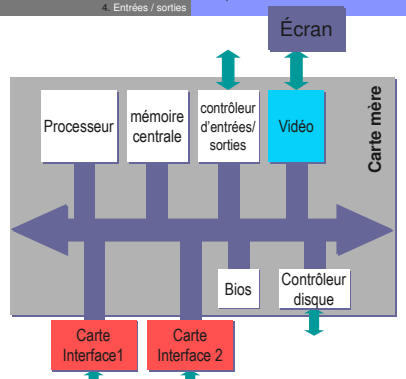


2. Modifier E en e



3. Écrire e en MC[4i]



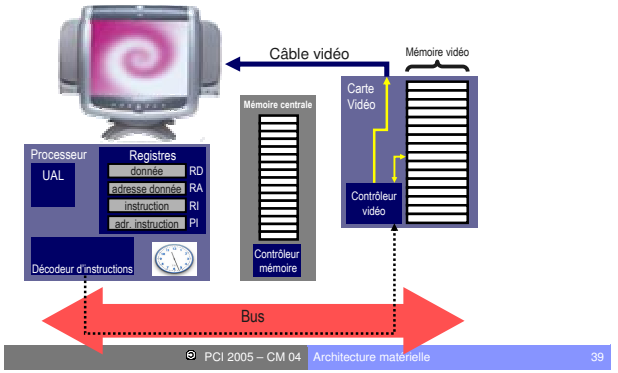


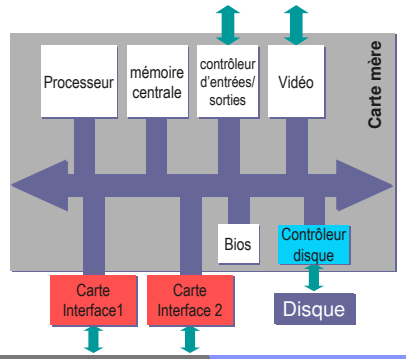
Résolution, couleur

- L'écran se décompose en lignes de pixels
  - *picture elements* : état colorimétrique d'un point
- Résolution d'affichage
  - **H** lignes
  - **W** colonnes
- La couleur d'un pixel
  - **D** octets (R,V,B)
  - 3 octets = 256^3 = 16,3 M de couleurs
- Les résolutions utilisables dépendent
  - de la quantité de mémoire vidéo
  - des capacités du moniteur

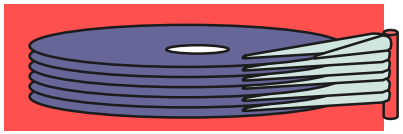
Exemple:
1(R) + 1(V) + 1(B) = 3 octets
R = 255 V = 000 B = 000 rouge
R = 000 V = 255 B = 000 vert
R = 255 V = 255 B = 000 jaune

Principe

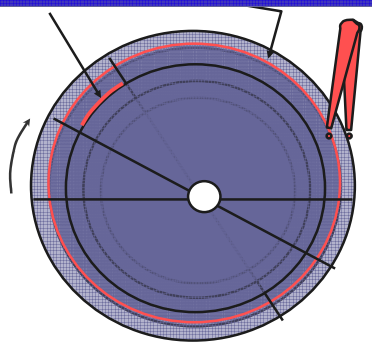




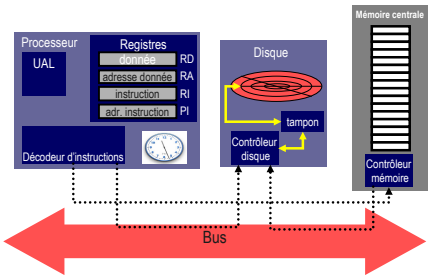
Anatomie



Secteurs, pistes

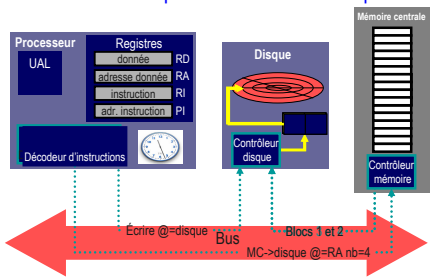


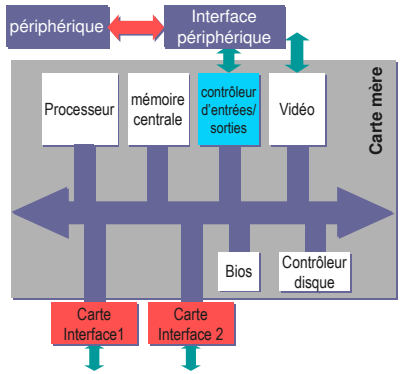
Échange de blocs de *n* secteurs



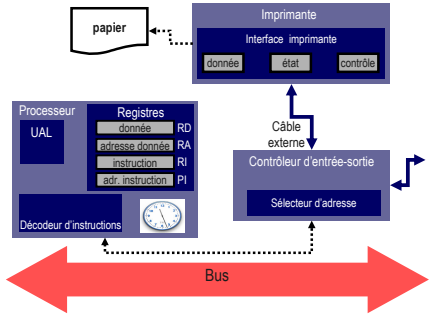
Échange de blocs de *n* secteurs

- Transfert de 4 blocs depuis la MC vers le disque

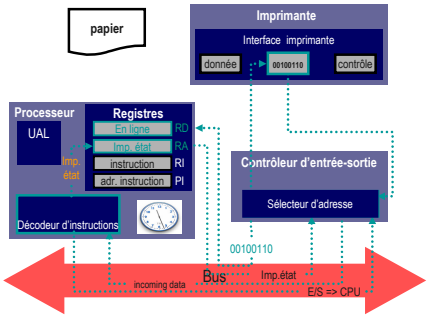




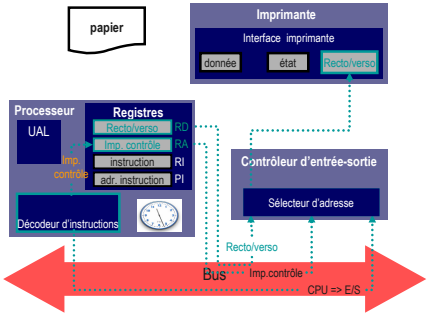
Communication avec le processeur



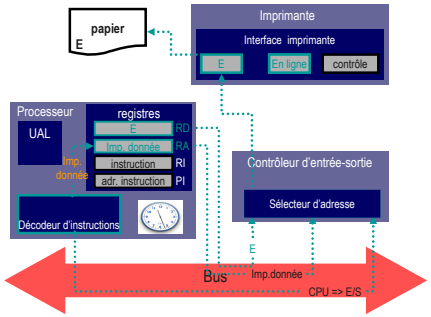
1. Lire l'état de l'imprimante



2. Configurer l'imprimante



3. Transmettre des données



Remarques

- Nous venons de voir un modèle simplifié du fonctionnement matériel d'un ordinateur.
- Au delà du modèle, il convient de se reporter aux descriptions techniques pour comprendre la technologie précise de votre ordinateur.

Quelques sites

- Une sélection qui vieillira vite...
  - http://www.01hardware.com
    pour l'assistance sur le matériel
  - http://www.teaser.fr/~spineau/acrodict/
    pour les acronymes informatiques
  - http://www.aideonline.com
    un site d'aide en ligne gratuit..
  - http://perso.wanadoo.fr/ameliorer.son.pc/informatique.htm
    une visite guidée par un « néophyte »