

Introduction à l'architecture des ordinateurs

CM1-1 : Architecture d'un système informatique

Mickaël Martin Nevot

V1.2.0



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 3.0 non transposé](#).

Introduction à l'architecture des ordinateurs

- I. Présentation
- II. Architecture
- III. Réseaux
- IV. Codage de l'info.

Architecture de référence

Couches
hautes

- Couche des applications
- Couche distribution

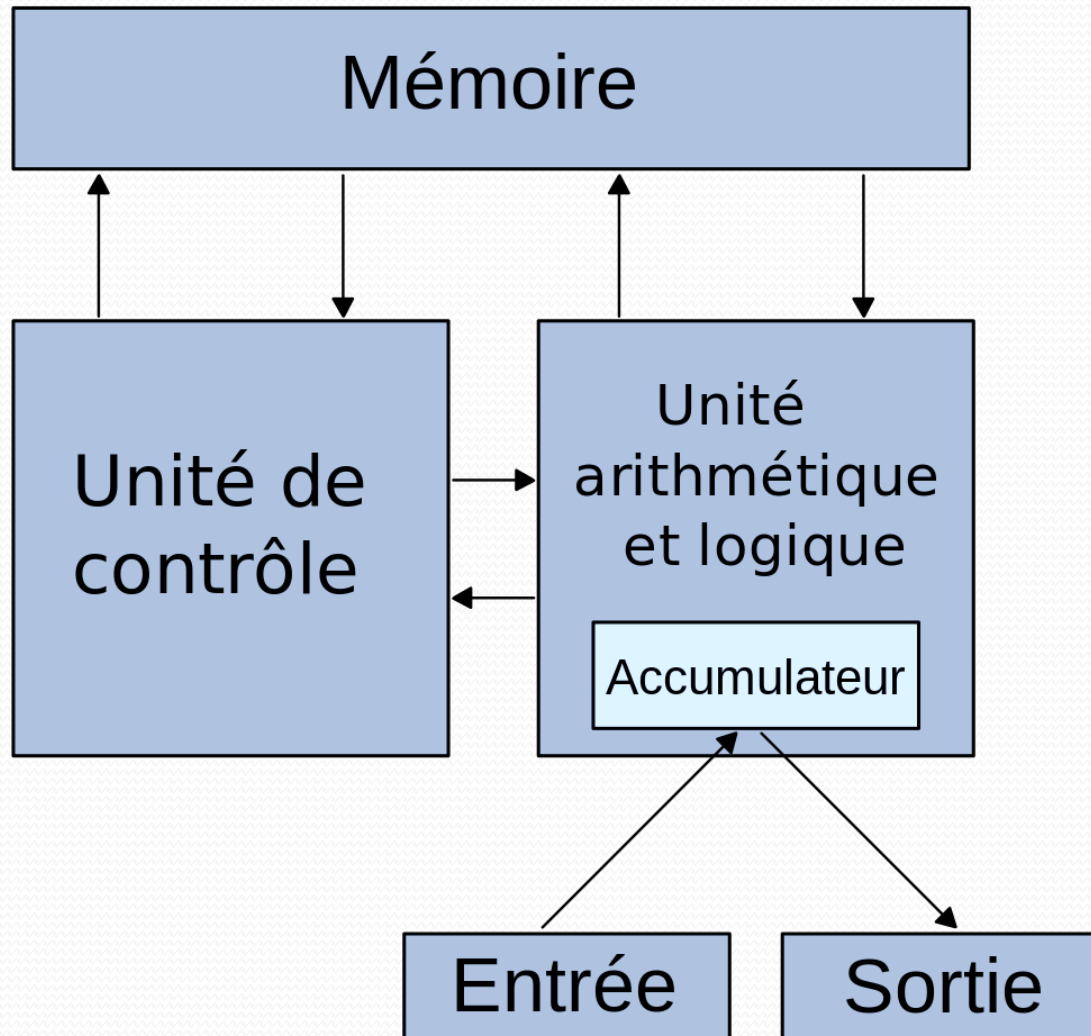
Couche
système/réseau

- Services additionnels
- Noyau

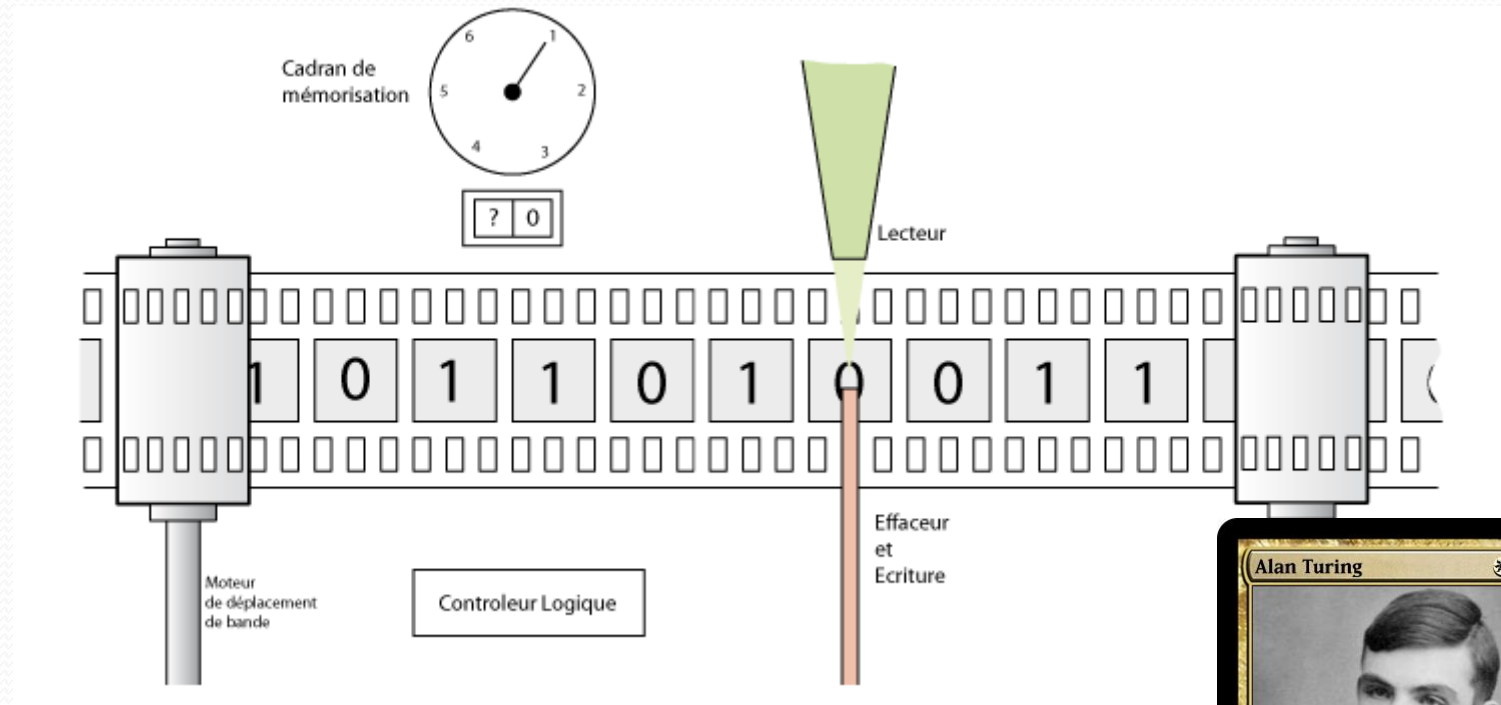
Couches
basses

- Couche du microcode et du langage machine
- Couche du matériel

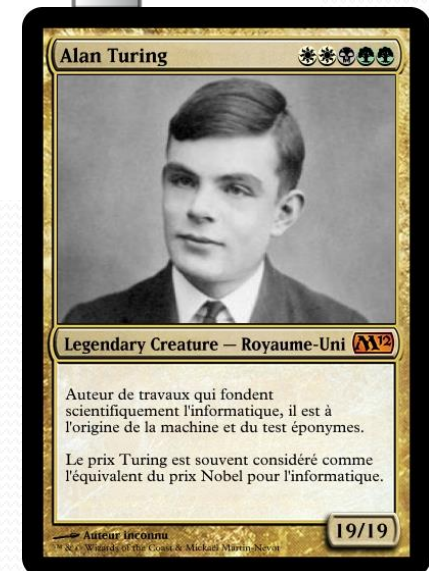
Architecture de von Neumann



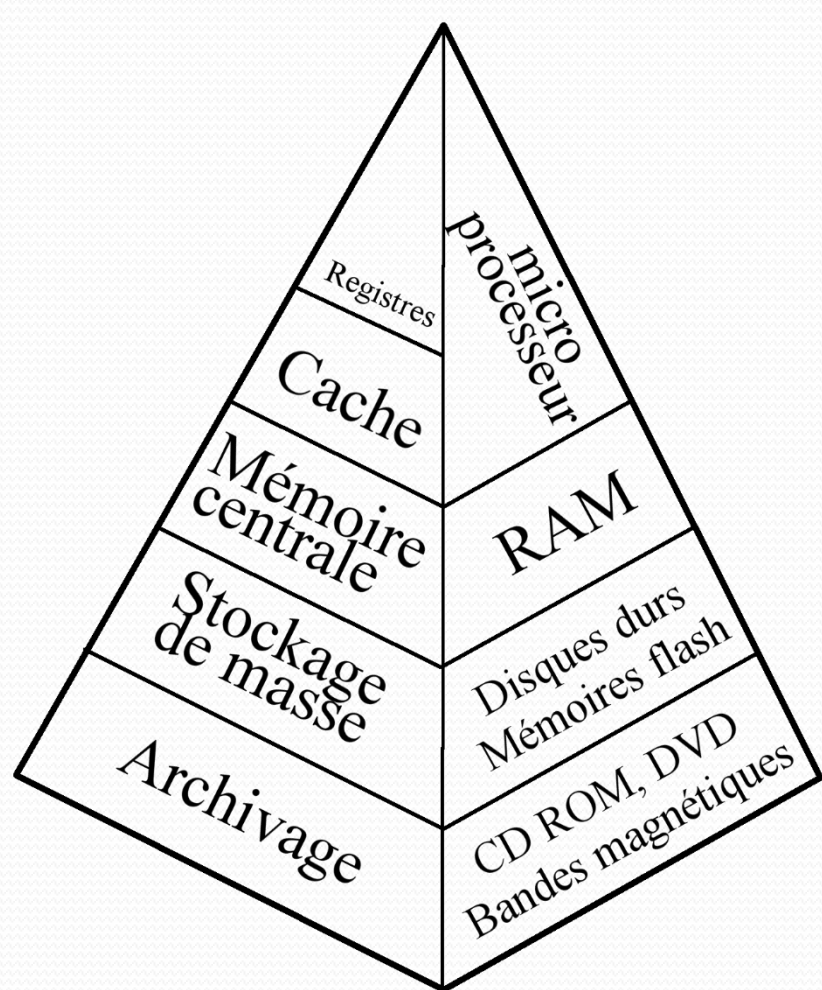
Machine de Turing



Tout ce qui est calculable peut être calculé par une machine de Turing



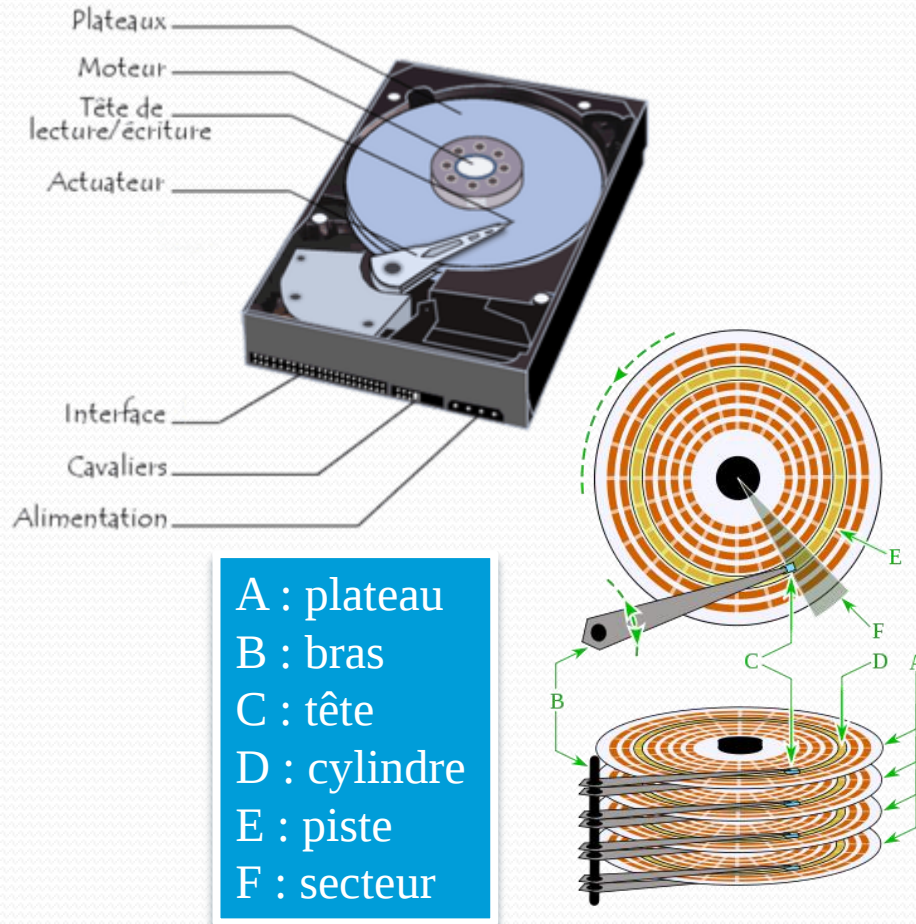
Hiérarchie des mémoires



+ ROM (*read only memory*) : mémoire persistante dont le contenu est fixé initialement (BIOS, *firmware*, etc.)

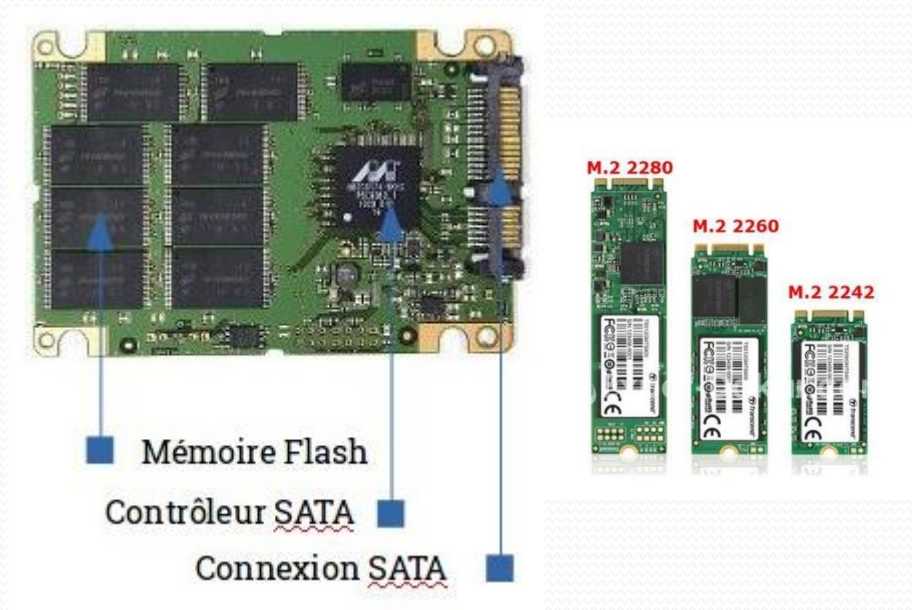
Stockage de masse

Disques magnétiques



SSD

- Mémoire vive persistante



Portes et circuits logiques

- Rappel : algèbre de Boole

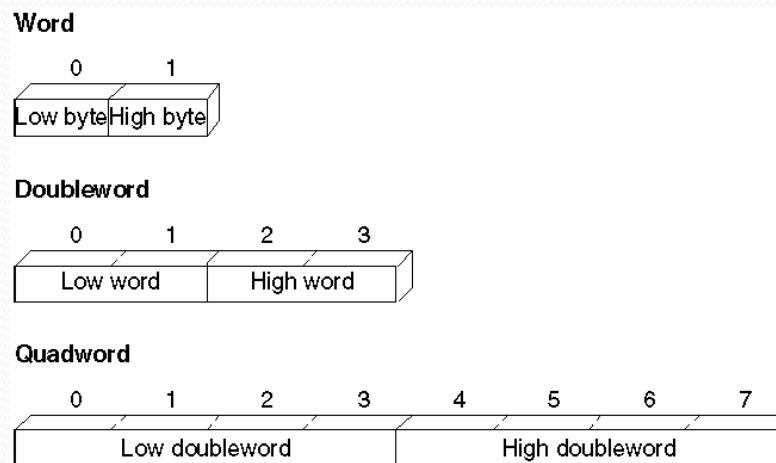
FONCTION	SYMBÔLES		TABLES DE VERITE		
	International	Français	a	b	S
NON			a	b	S
			0	1	1
			1	0	0
ET			a	b	S
			0	0	0
			0	1	0
			1	0	0
			1	1	1
NAND			a	b	S
			0	0	1
			0	1	1
			1	0	1
			1	1	0
OU			a	b	S
			0	0	0
			0	1	1
			1	0	1
			1	1	1
NOR			a	b	S
			0	0	1
			0	1	0
			1	0	0
			1	1	0
OU Exclusif			a	b	S
			0	0	0
			0	1	1
			1	0	1
			1	1	0
NOR Exclusif			a	b	S
			0	0	1
			0	1	0
			1	0	0
			1	1	1



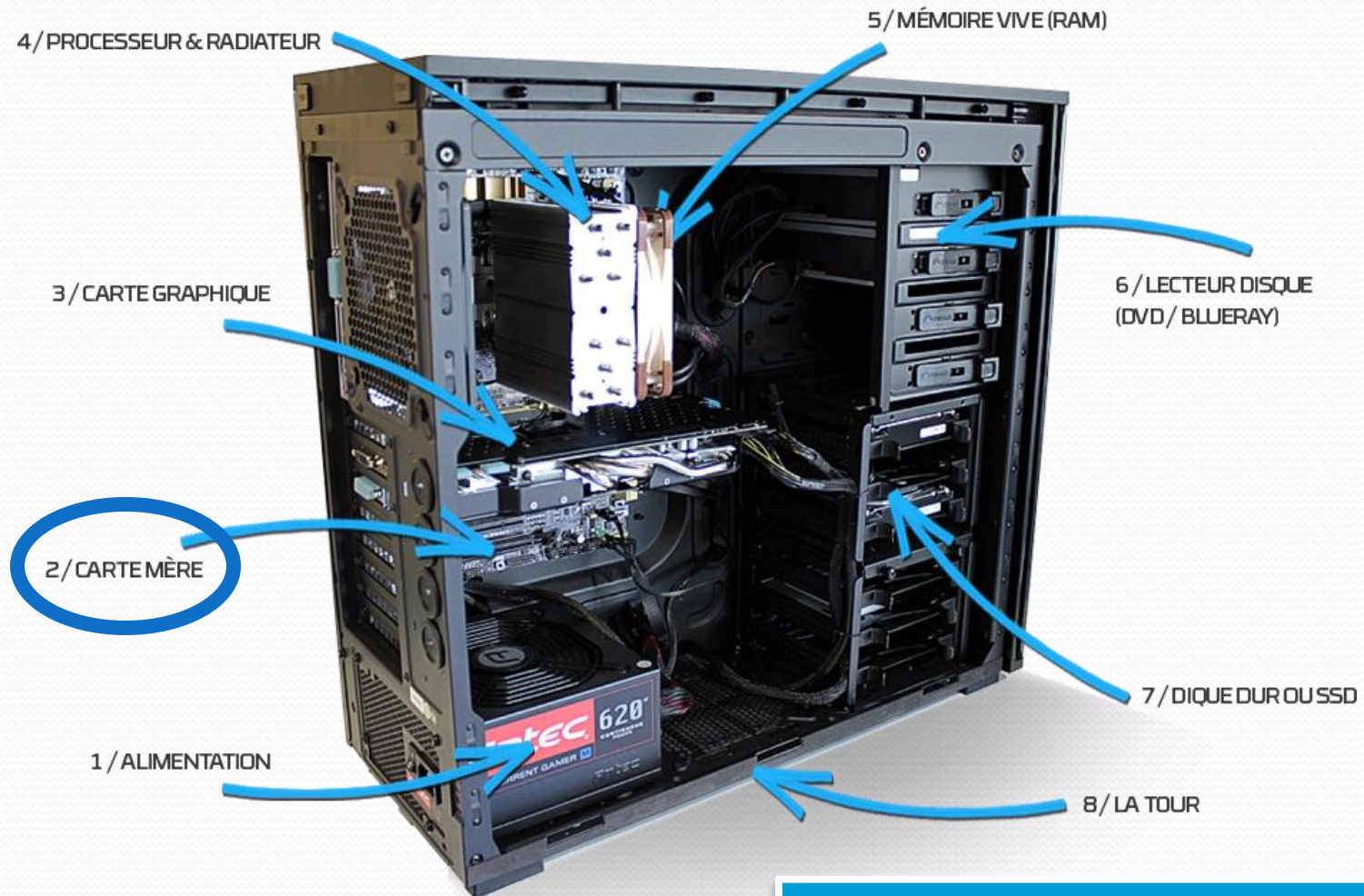
Les circuits logiques peuvent être combinatoires ou séquentiels

Mot

- Unité
- Cellule mémoire (de traitement ou de transition)
- Dépend du microprocesseur :
 - 32 bits (x86) Appelé *dword* pour « double mot »
 - 64 bits Appelé *qword* pour « quadruple mot »
- Au plus mot long, au plus microprocesseur rapide

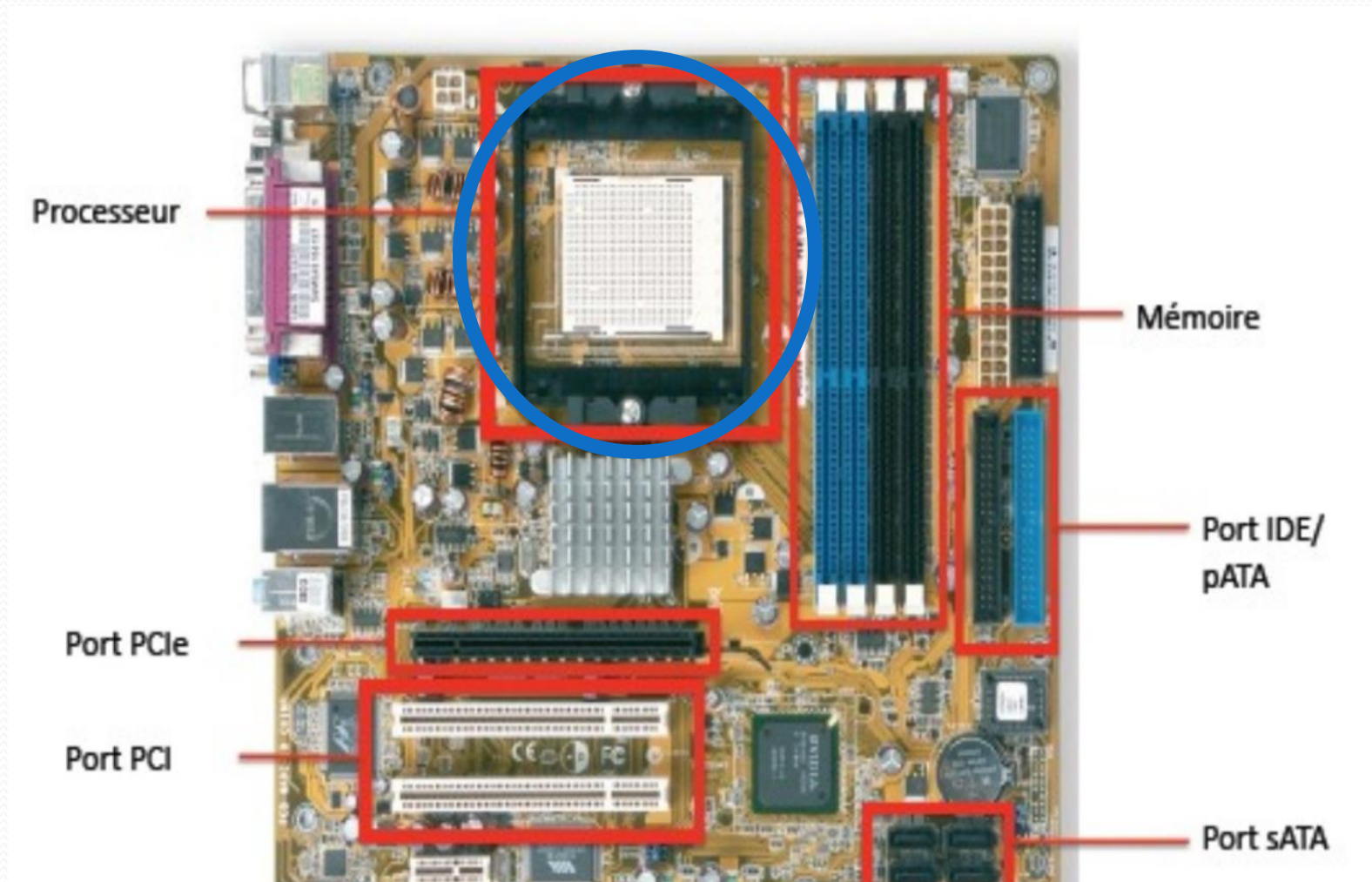


Ordinateur

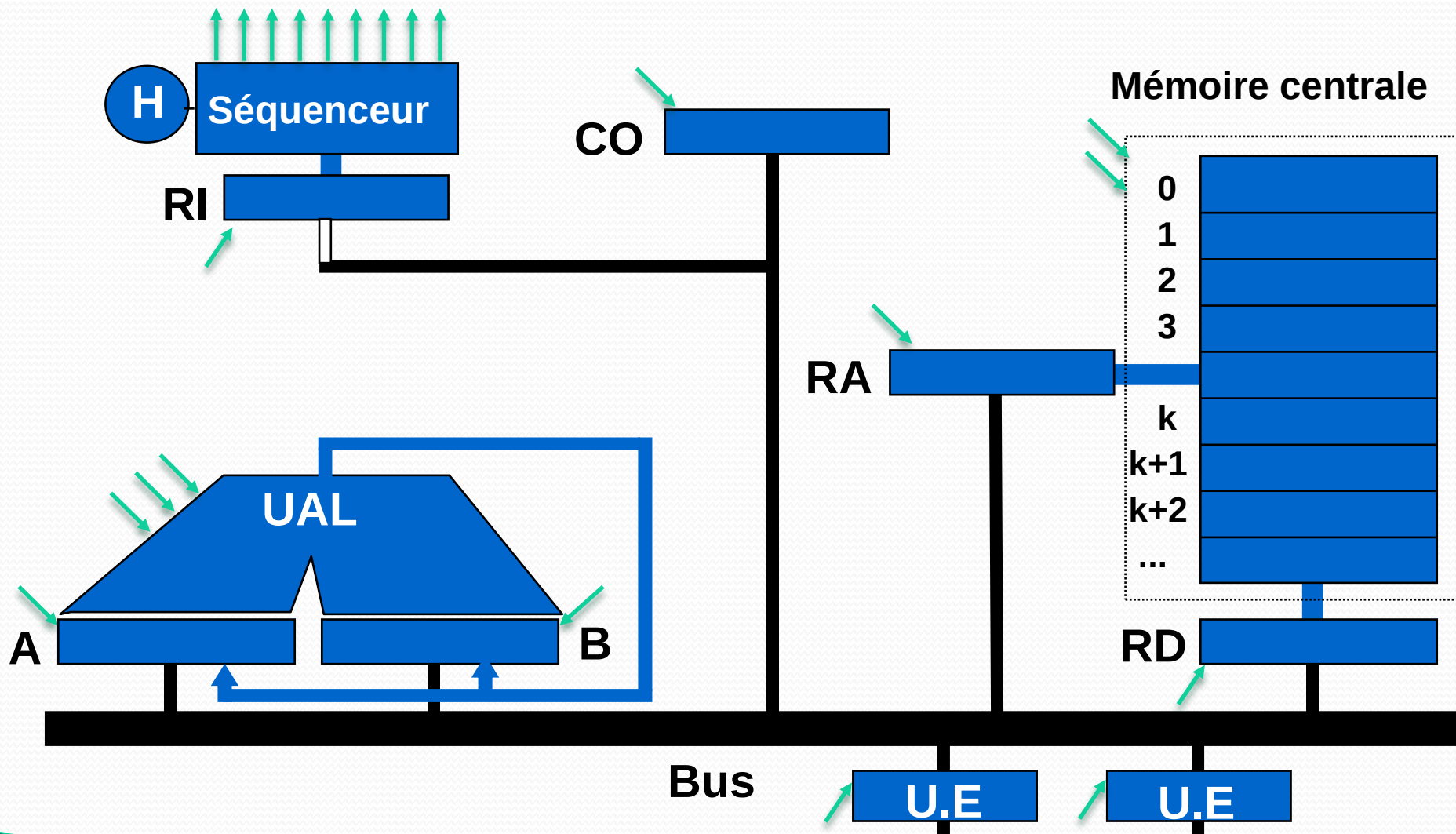


Idem pour les ordinateurs portables

Carte mère

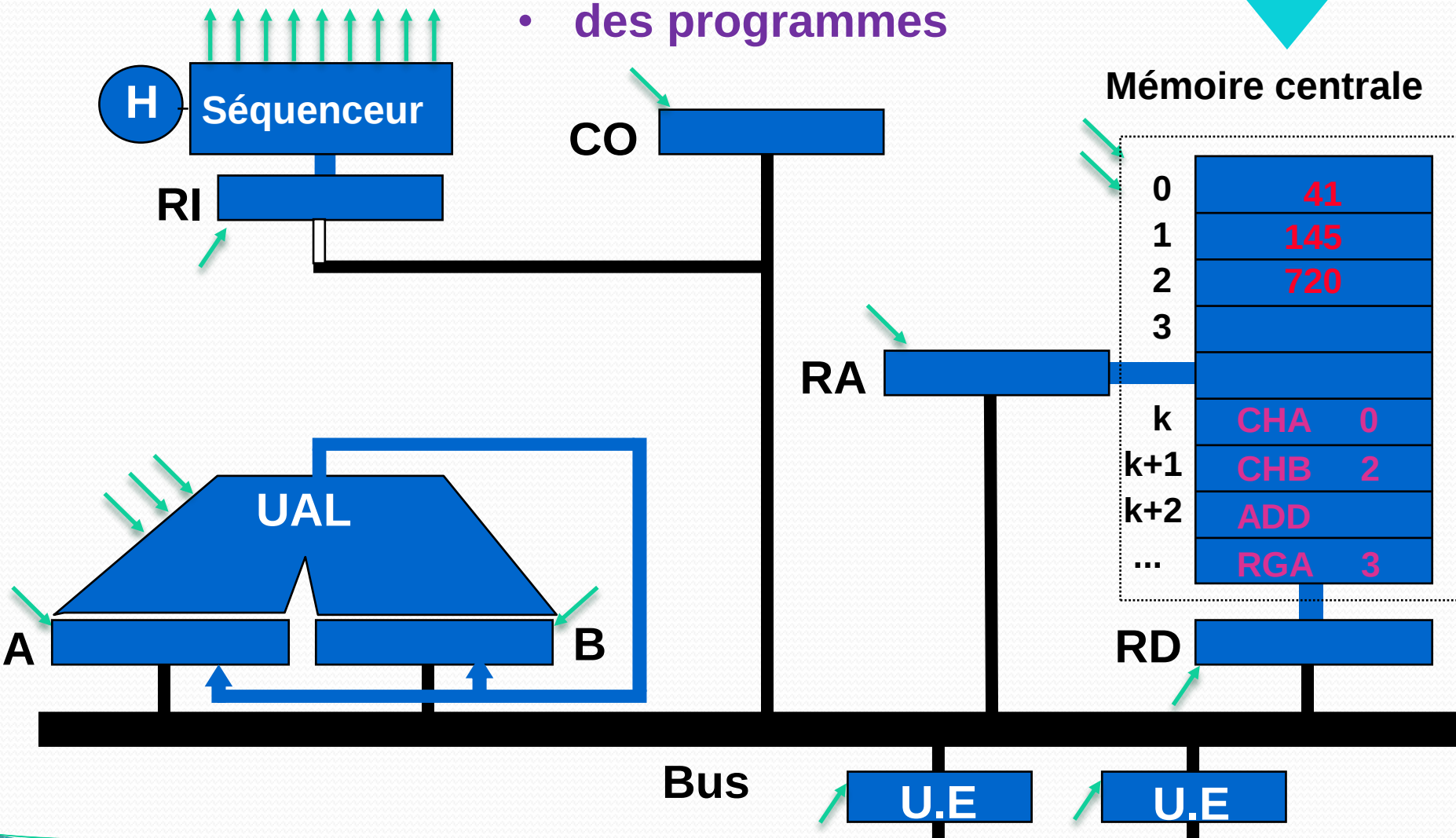


Unité centrale

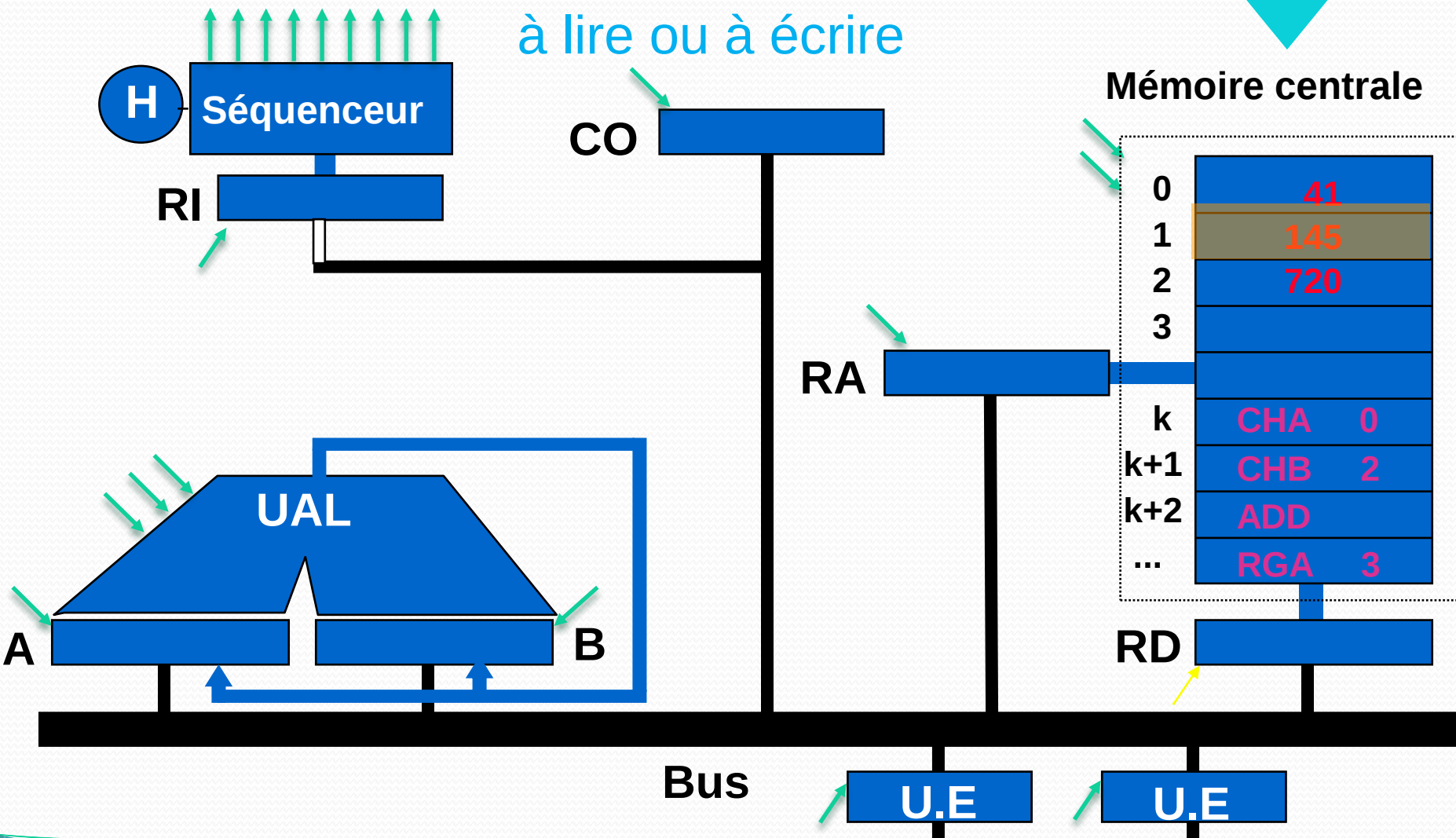


La mémoire centrale contient :

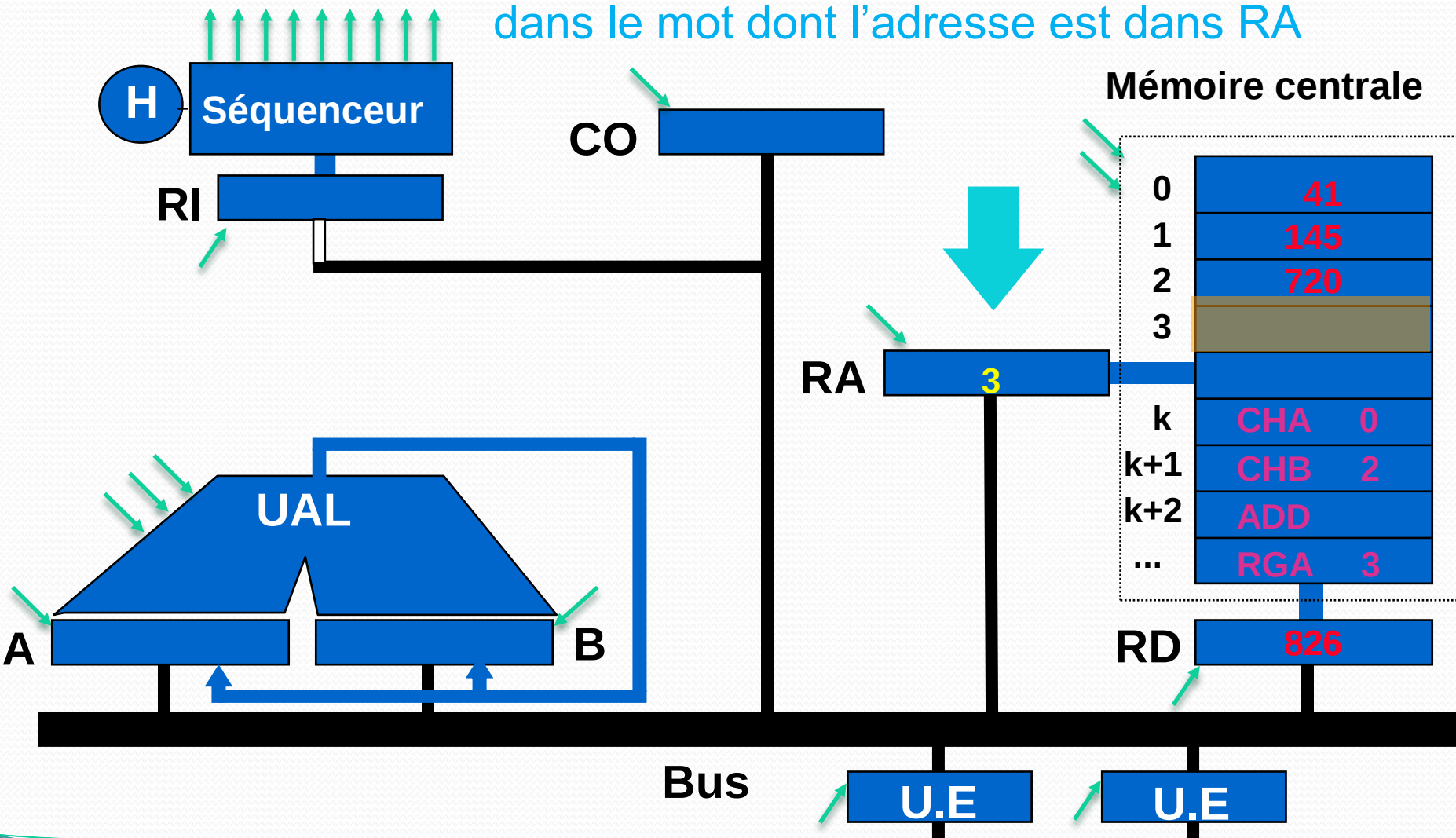
- des données
- des programmes



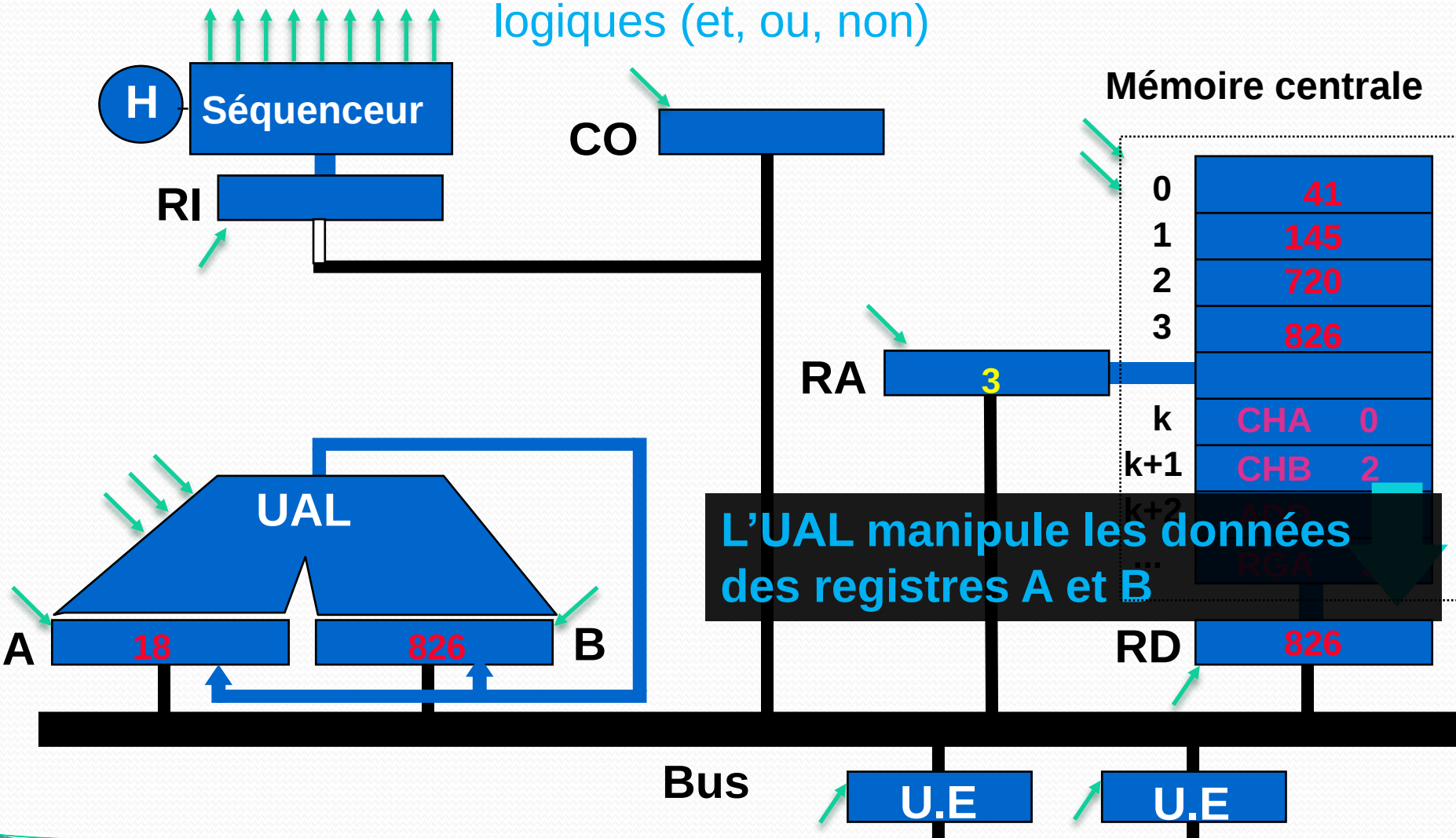
Le registre d'adresse (RA)
contient l'adresse du mot mémoire
à lire ou à écrire



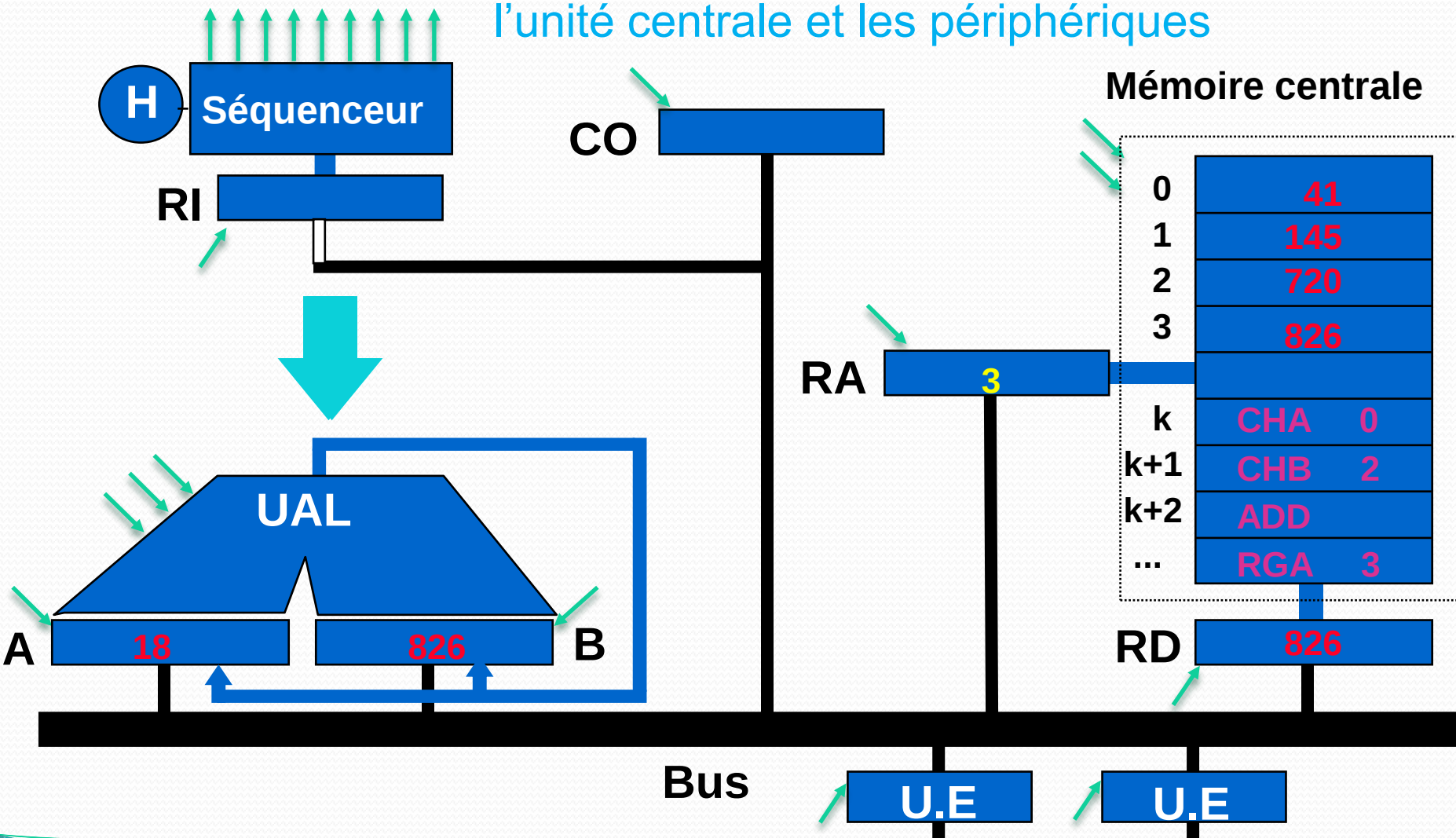
Le registre donnée (RD)
contient l'information lue ou à écrire
dans le mot dont l'adresse est dans RA



L'unité arithmétique et logique effectue les calculs (+, -, *, /) et les opérations logiques (et, ou, non)

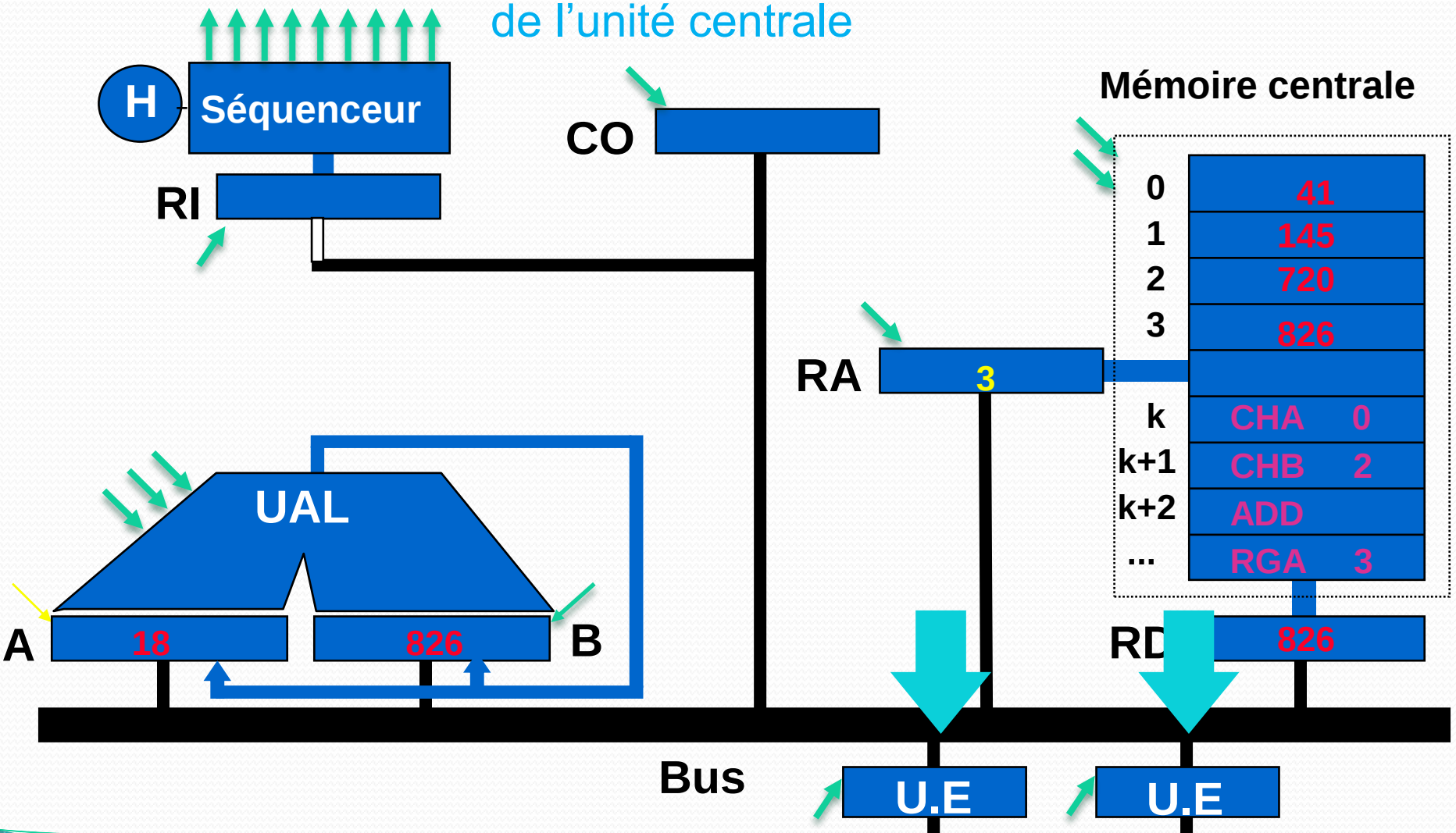


Les unités d'échange assurent la communication entre l'unité centrale et les périphériques

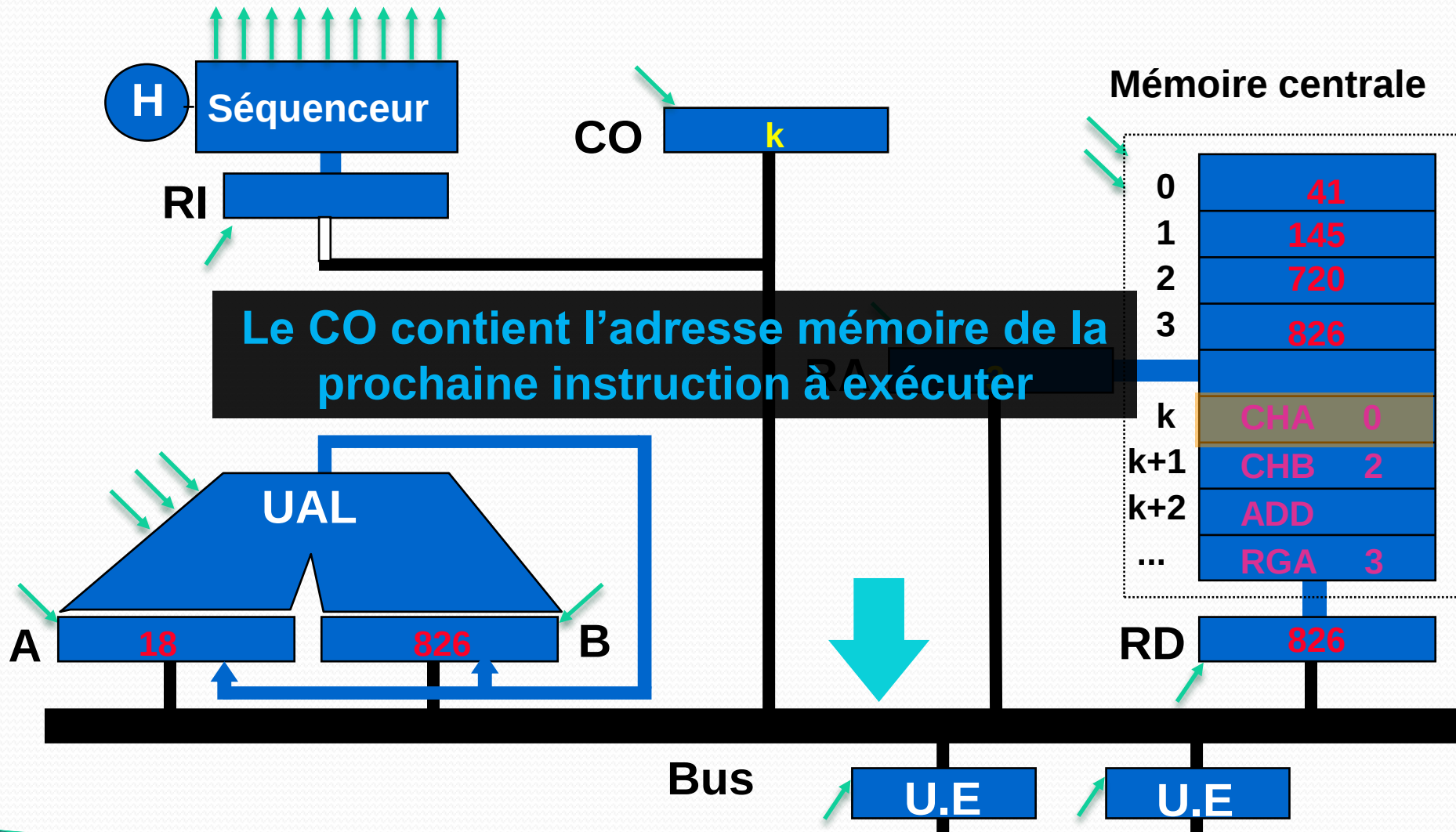


Le Bus

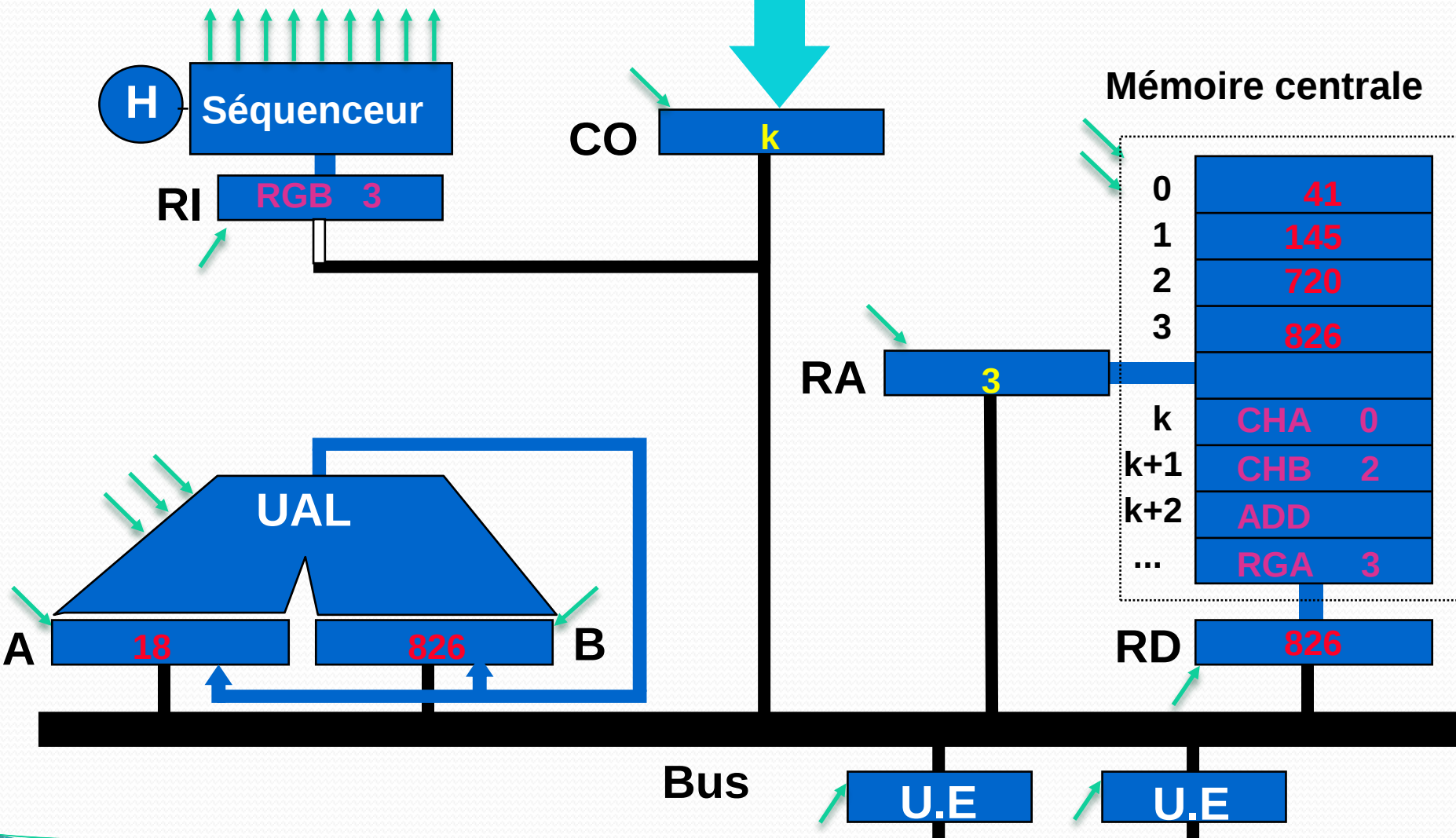
assure la communication entre les composants de l'unité centrale



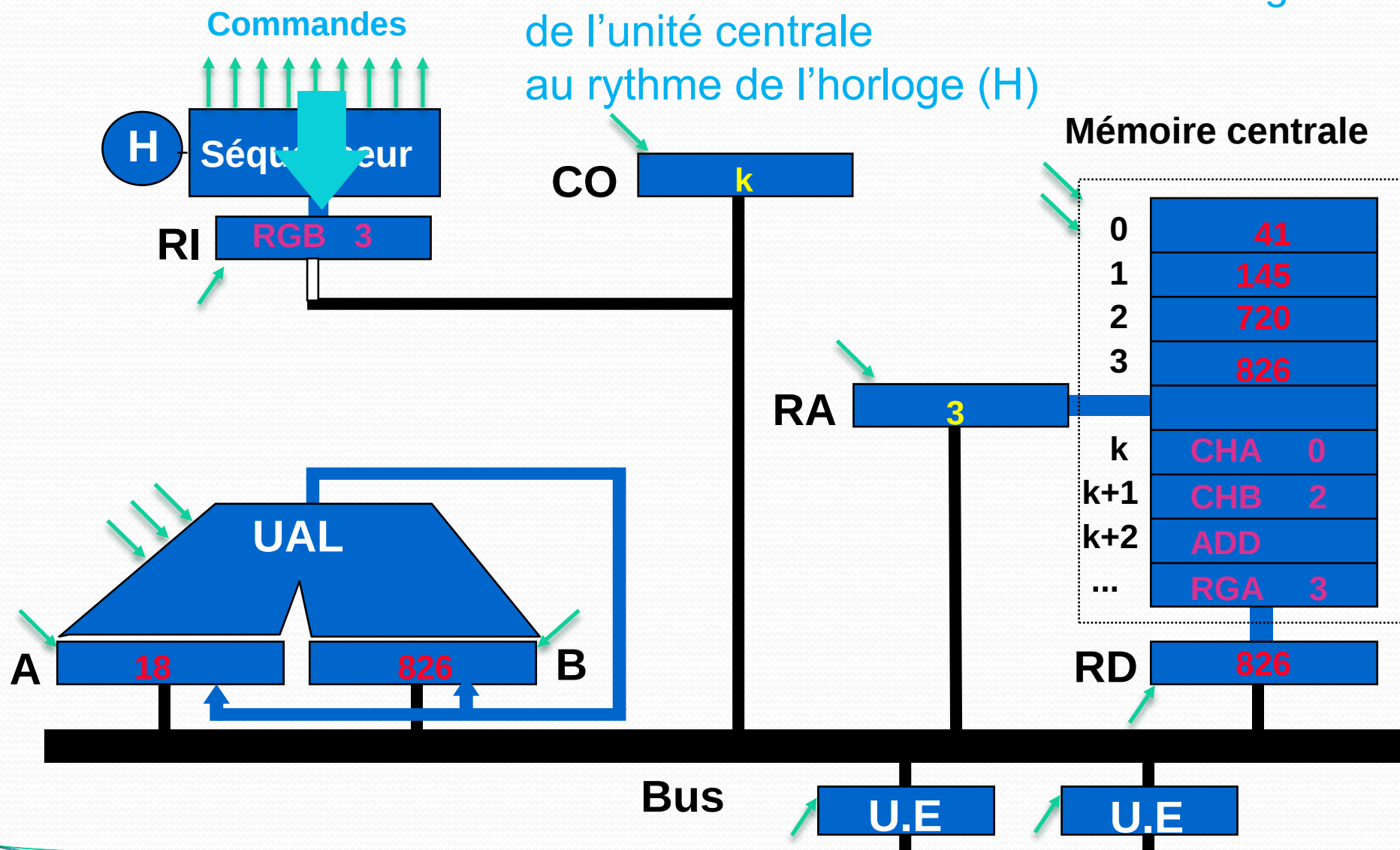
Le compteur ordinal (CO)

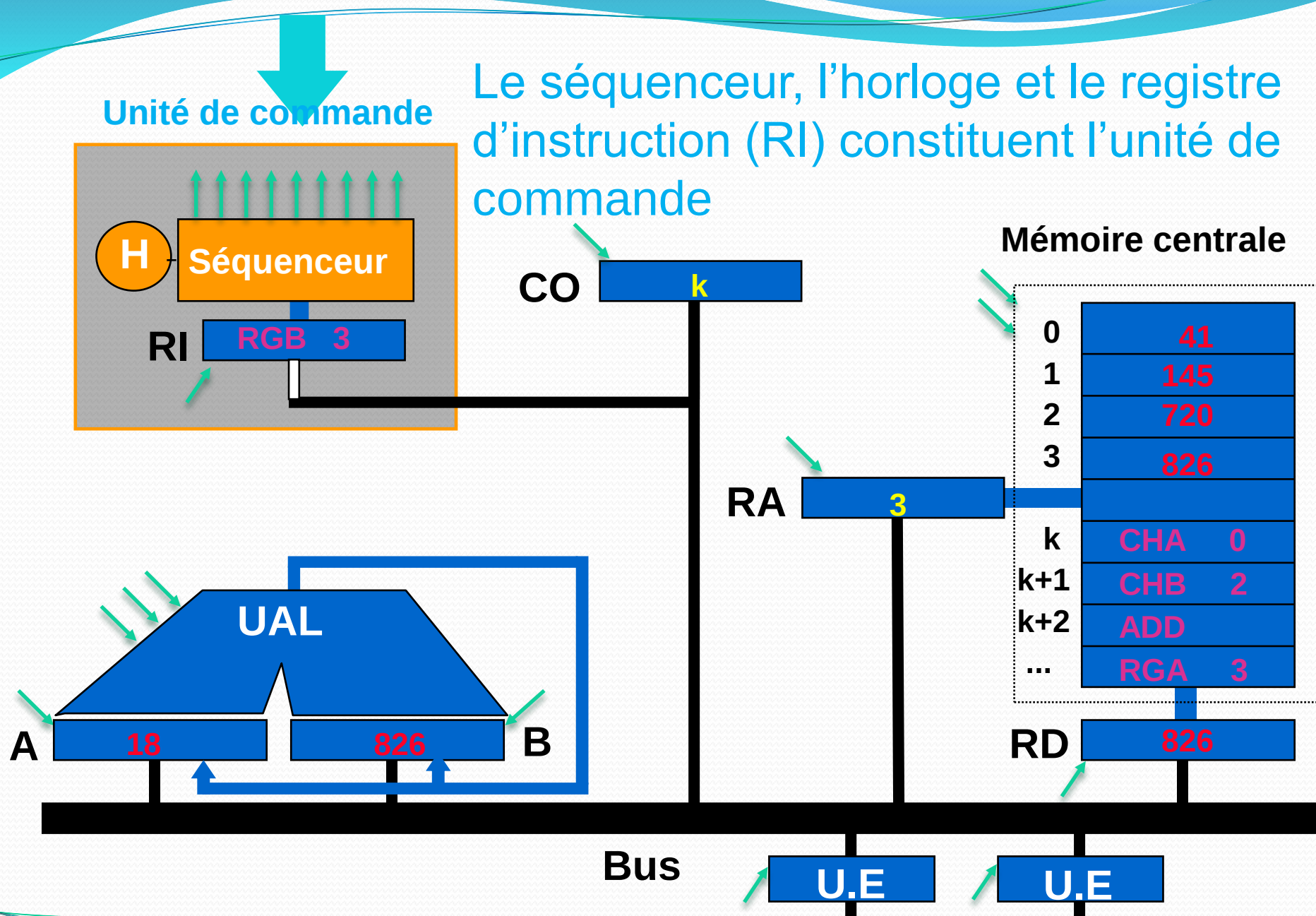


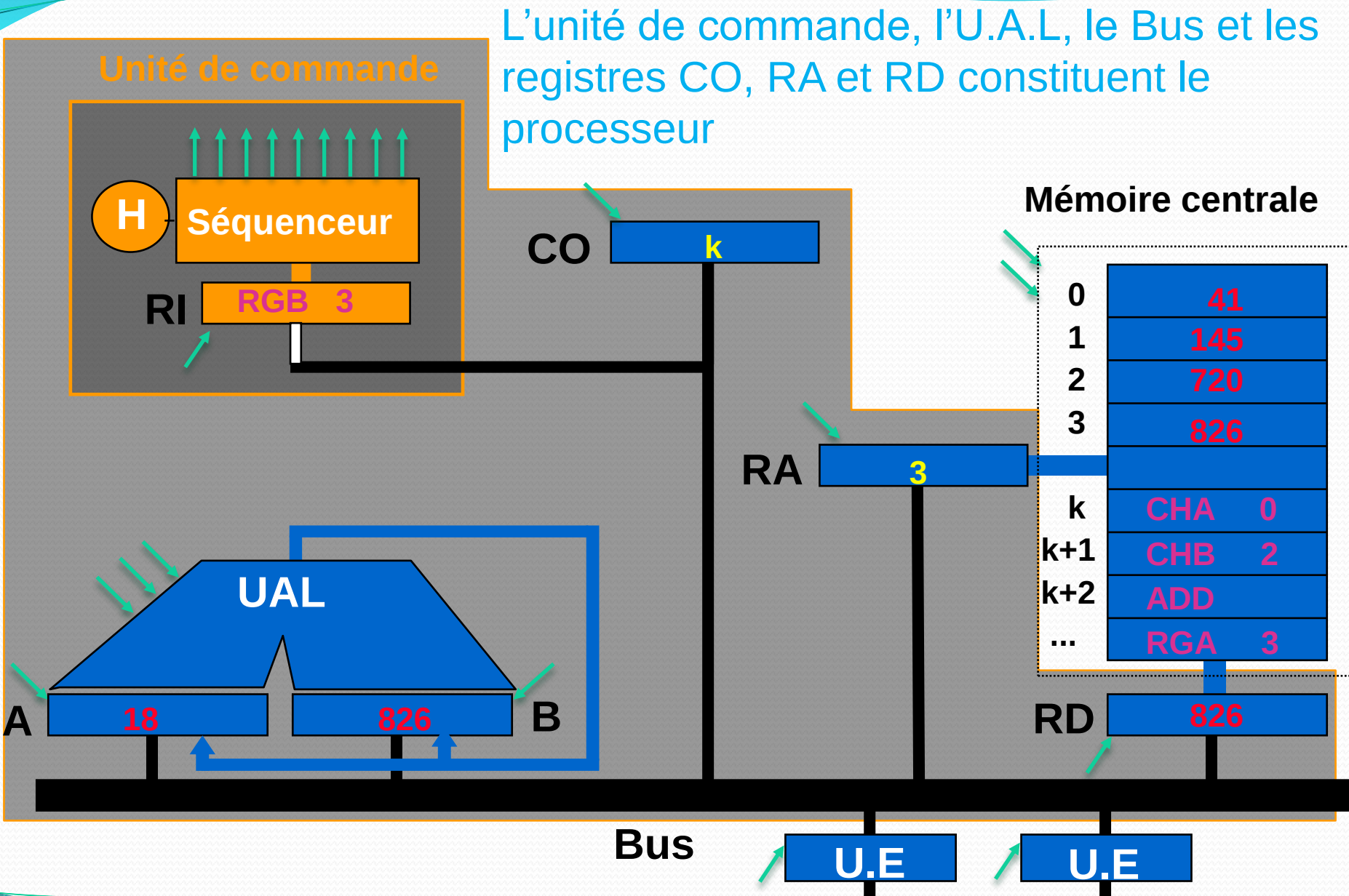
Le registre d'instruction (RI)
contient l'instruction en cours d'exécution



Le séquenceur
envoie ses commandes aux autres organes
de l'unité centrale
au rythme de l'horloge (H)

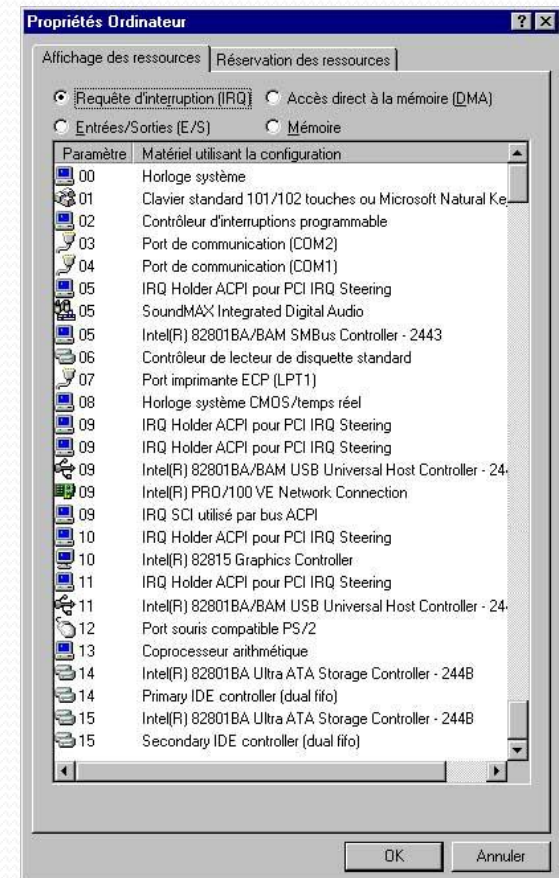






Les interruptions

- **Arrêt temporaire** de l'exécution d'un programme par le microprocesseur pour l'exécution d'un autre plus prioritaire
- **Trois types d'interruption** :
 - Interruption processeur (erreur de calcul)
 - Interruption matérielle (IRQ)
 - Interruption logicielle
- **Principales utilisations** :
 - Communications non bloquantes avec des **périphériques** plus lents que le microprocesseur
 - Communication entre **processus** par l'ordonnanceur



Langages : niveaux d'abstraction

- Application → Langage naturel
- Algorithme → Langage dépendant de l'app.
- Langage de haut niveau → Langage de programmation
- Système d'exploitation → Appels du système
- Architecture de la machine → Langage assembleur
- Microarchitecture → Langage machine
- Circuits logiques → Langage de trans. de registres
- Dispositifs électroniques → Algèbre de Boole

Langage machine

- Binaire : 0 et 1
- Assembleur (langage)

Tout « fil » (Bus) dans un ordinateur a un voltage haut ou bas : la valeur réelle n'intéresse pas, c'est seulement un 1 ou un 0

The screenshot shows the OllyDbg interface for winzip32.exe. The main window displays assembly code with addresses, hex dumps, and mnemonics. The registers window on the right shows the current state of the CPU registers. The memory window at the bottom shows the address, hex dump, and ASCII representation of the code.

1) CODE ASSEMBLEUR

Address	Hex dump	ASCII
00400D42	6A 60	PUSH 60
00400D44	68 F8634C00	PUSH winzip32.004C63F8
00400D49	E8 622D0000	CALL winzip32.004A3AB0
00400D4E	BF 94000000	MOV EDI, 94
00400D53	8BC7	MOV EAX, EDI
00400D55	E8 A60A0000	CALL winzip32.004A1800
00400D5A	8965 E8	MOV DWORD PTR SS:[EBP-18], ESP
00400D5D	8BF4	MOV ESI, ESP
00400D5F	893E	MOV DWORD PTR DS:[ESI], EDI
00400D61	53	POP EBX
00400D62	FF15 55C43AB0	CALL winzip32.0045547E8
00400D68	8B4E 10	MOV ECX, DWORD PTR DS:[ESI+10]
00400D6B	890D C0814E00	MOV DWORD PTR DS:[4E81C0], ECX
00400D71	8B46 04	MOV EAX, DWORD PTR DS:[ESI+4]
00400D74	A3 CC814E00	MOV DWORD PTR DS:[4E81CC], EAX
00400D79	8B56 08	MOV EDX, DWORD PTR DS:[ESI+8]
00400D7C	8915 D0814E00	MOV DWORD PTR DS:[4E81D0], EDX
00400D82	8B76 0C	MOV ESI, DWORD PTR DS:[ESI+C]
00400D85	81E6 FF7F0000	AND ESI, 7FFF
00400D88	8935 C4814E00	MOV DWORD PTR DS:[4E81C4], ESI
00400D91	83F9 02	CMP ECX, 2
00400D94	74 0C	IF SHORT winzip32.00400DD2

2) REGISTRES

Register	Value
EAX	00000000
ECX	0012FFB0
EDX	7FFE0304
EBX	7FFDF000
ESP	0012FFC4
EBP	0012FFF0
ESI	005547E8 winzip32.005547E8
EDI	005547E8 winzip32.005547E8
EIP	00400D42 winzip32

3) ADRESSES MEMOIRE

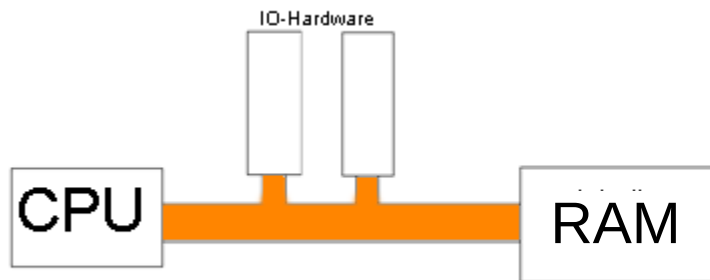
Address	Hex dump	ASCII
0040D700	00 00 00 00 31 5C 4A 00	1NJ.
0040D708	1B 8D 4B 00 31 8D 4B 00	+iK.1iK.
0040D710	47 8D 4B 00 50 8D 4B 00	GiK.1iK.
0040D718	A0 8F 4B 00 B6 8F 4B 00	âAK.âAK.
0040D720	D1 8F 4B 00 EC 8F 4B 00	âAK.âAK.
0040D728	00 00 00 00 00 00 00 00	
0040D738	AA 90 4B 00 1C 91 4B 00	TEK.âEK.
0040D740	FA 91 4B 00 10 92 4B 00	âEK.âEK.
0040D748	26 92 4B 00 3C 92 4B 00	âEK.<âEK.
0040D750	54 92 4B 00 6A 92 4B 00	TiEK.âEK.
0040D758	82 92 4B 00 9A 92 4B 00	âEK.âEK.
0040D760	B2 92 4B 00 CA 92 4B 00	âEK.âEK.

4) EA FILE

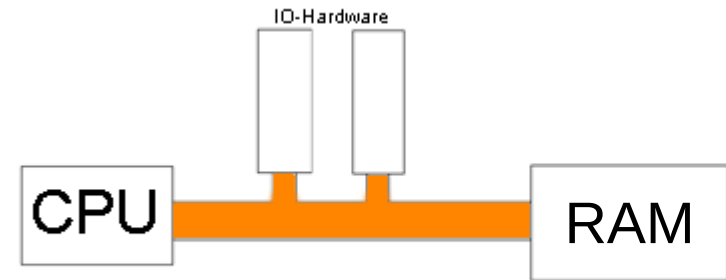
Address	Hex dump	ASCII
0012FFC4	77E614C7	RETURN to kernel32.77E614C7
0012FFC8	005547EC	winzip32.005547EC
0012FFCC	005547E8	winzip32.005547E8
0012FFD0	7FFDF000	
0012FFD4	EF2EBCF0	
0012FFD8	0012FFC8	
0012FFDC	0053C89F	
0012FFE0	47FE0304	EBX from SEH chain
0012FFE4	77E74809	SE handler
0012FFE8	77E71210	kernel32.77E71210
0012FFEC	00000000	
0012FFF0	00000000	
0012FFF4	00000000	
0012FFF8	004A0D42	ASCII "j'h'cL"

DMA

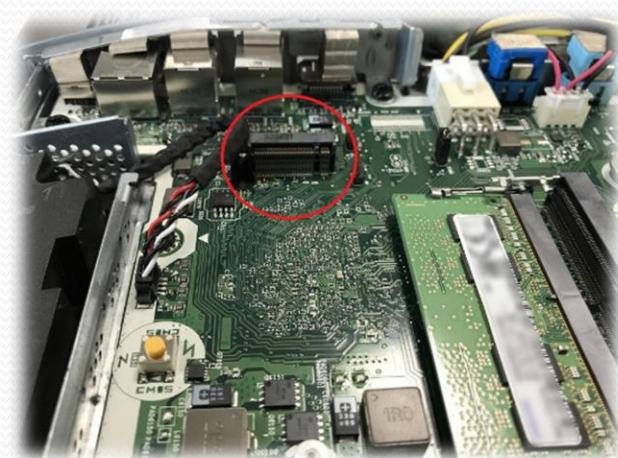
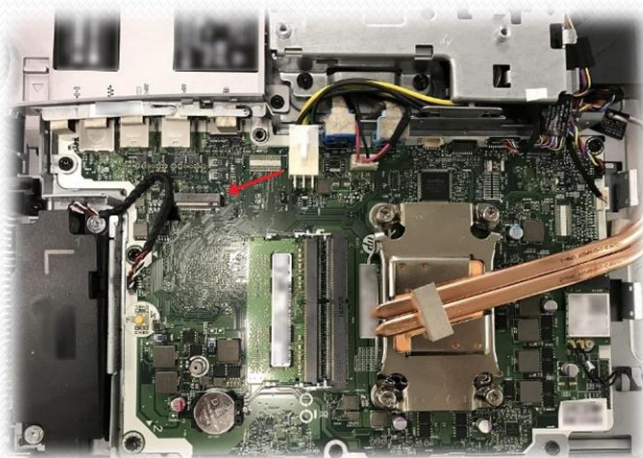
- *Direct memory access*



Entrée-sortie sans DMA



Entrée-sortie avec DMA



Crédits

Auteur

Mickaël Martin Nevot

mmartin.nevot@gmail.com



Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com

