

Architecture des systèmes à processeur

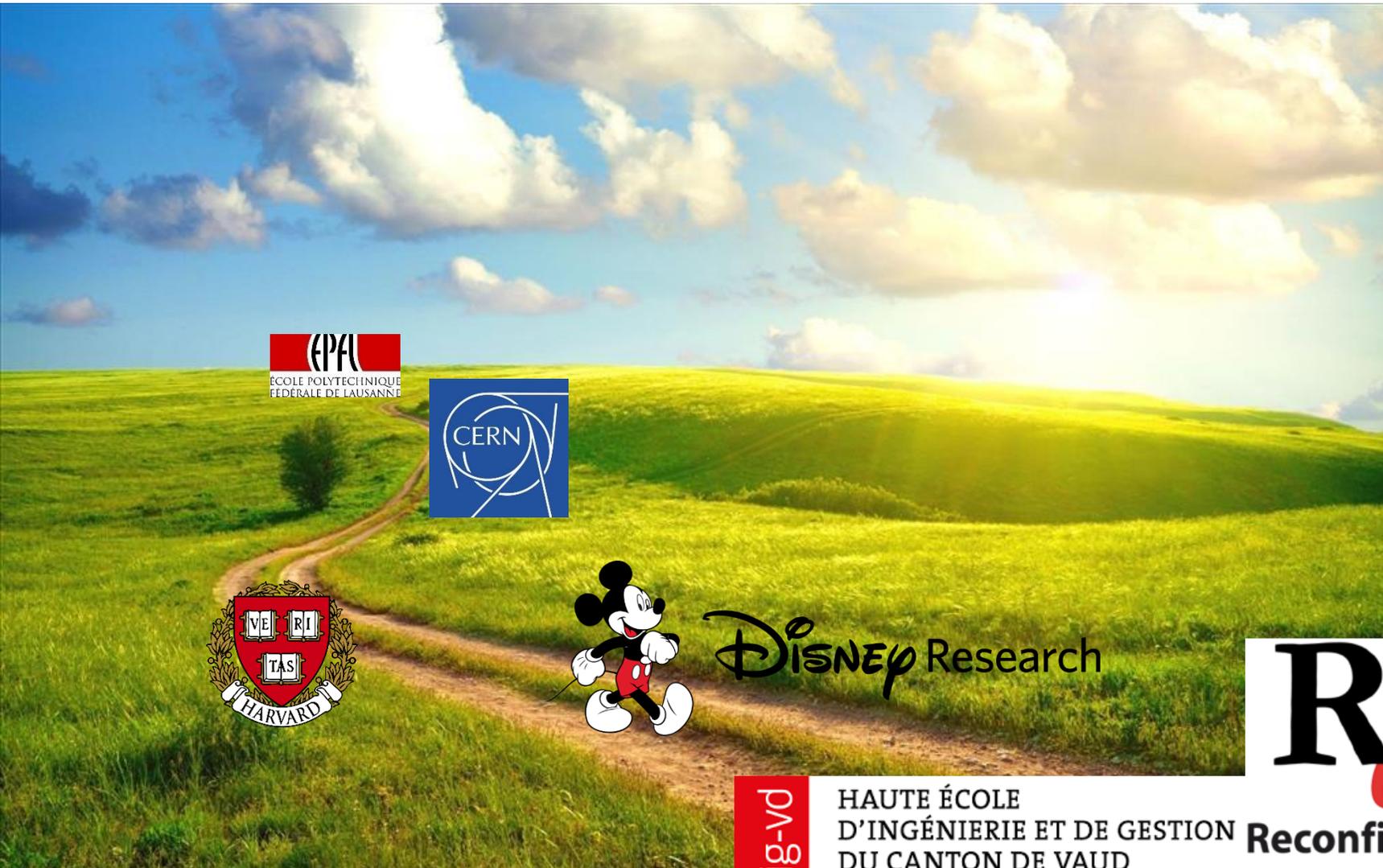
Prof. Géraldine Conti

Basé sur les cours des Profs. Sanchez, Starkier, Mosqueron et Dassatti

Modalités (1)

- Professeure Géraldine Conti, institut REDS
- Bureau A21
- Email : geraldine.conti@heig-vd.ch
- Horaires :
 - Cours : mardis de 14h55-16h30 en A07
 - Laboratoires : jeudis de 13h15-14h45 en A07
- Assistant de laboratoire : David Truan (david.truan@heig-vd.ch)

Mon parcours professionnel...



heig-**vd**

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

ReDS
Reconfigurable & embedded
Digital Systems

Modalités (2)

- Supports de cours :
 - Cyberlearn : **19_HES-SO-VD_Architecture des systèmes à processeur (ASP)**
 - Clé : 2019aspgci
- Notation :
 - Pas d'examens
 - 3 tests de deux périodes durant le semestre
 - 3 laboratoires notés
 - Note finale = moyenne des tests * 0.6 + moyenne laboratoires * 0.4

Objectifs du cours

- 1) Spécifier une **architecture système** matérielle et logicielle
 - Connaître les **grandes familles de processeurs** et les principaux **fabricants** de composants et de cartes
 - Savoir analyser les **besoins** d'un système (puissance de calcul, débits, stockage)
- 2) Analyser et expliquer le principe de fonctionnement des **bus parallèle et série** à partir d'exemples de bus usuels
- 3) Analyser et expliquer la technologie, l'architecture et le fonctionnement de divers types de **mémoires**
- 4) Comprendre le fonctionnement et l'interfaçage de **périphériques** usuels d'entrée/sortie

Bibliographie

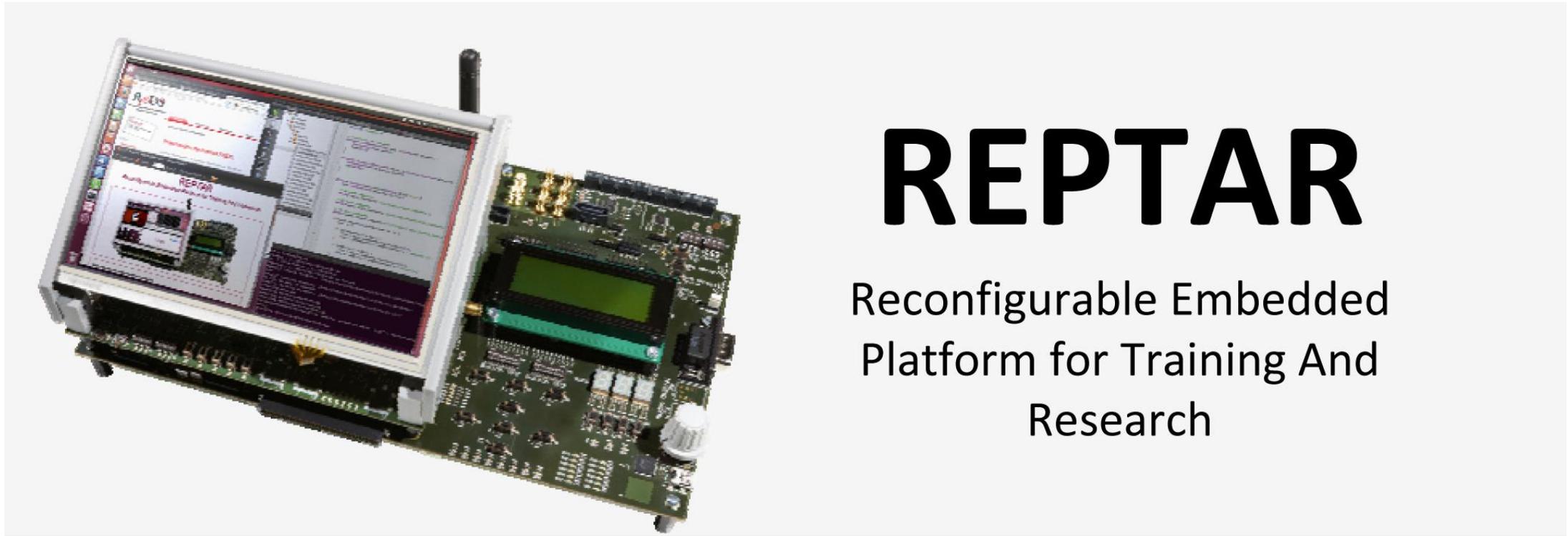
- Computer Architecture, Fifth Edition: A Quantitative Approach / John L. Hennessy , David A. Patterson Morgan Kaufmann Publishers 2011
- High Performance Embedded Computing Handbook: Systems perspective MARTINEZ David R., VAI Michael, BOND Robert A. CRC Press 2008
- Parallel Computer Organization and Design / Michel Dubois, Murali Annavaram, Per Stenström, Cambridge University Press 2012
- Processor architecture: from dataflow to superscalar & beyond / Jurij Silc, Springer 1999
- AM/DM37x Multimedia Device, Technical Reference Manual, TI 2011

Plan du cours

#	dates	cours	labos	Rendus labos (minuit)
1	19.02	Révisions + architecture	Introduction	
2	26.02	Architecture	Périphériques	
3	05.03	Architecture (cache)	Périphériques	
4	12.03	TEST	Périphériques	
5	19.03	Corrigé test, labo	Périphériques	
6	26.03	Parallélisme	Périphériques	
7	02.04	Parallélisme	Périphériques	
8	09.04	Famille processeurs	Sdcard	Périphérique
9	16.04	TEST	Sdcard	
10	30.04	Corrigé test, labo	Sdcard	
11	07.05	Mémoires	Sdcard	
12	14.05	Bus	Sdcard	
13	21.05	Bus	Sdcard	
14	28.05	TEST		
15	04.06	Corrigé test, labo	Picomon	Sdcard
16	11.06	Labo Picomon	Picomon	Picomon

4 laboratoires

Laboratoires



REPTAR

Reconfigurable Embedded
Platform for Training And
Research

1. Architecture



Exercice : Systèmes embarqués

- Définition (mots-clés)
- Exemples (domaines, projets du REDS)

Systemes embarqués



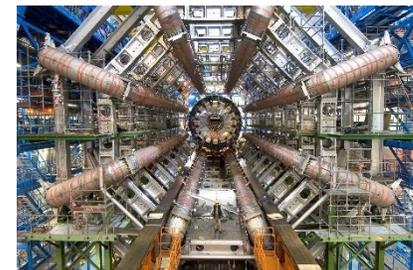
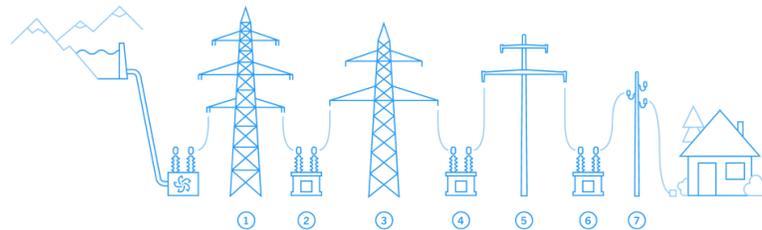
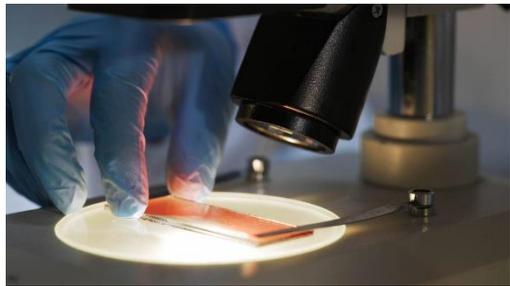
autonome

Ressources limitées
(spatial, énergie)

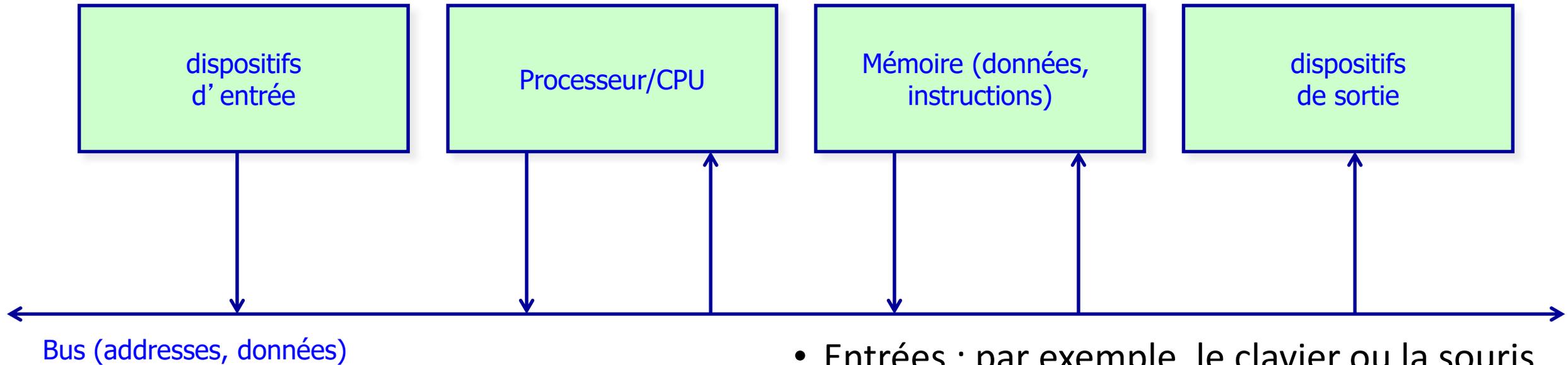
Électronique et
informatique

Temps réel

Spécialisé pour
tache précise



Architecture générale des ordinateurs



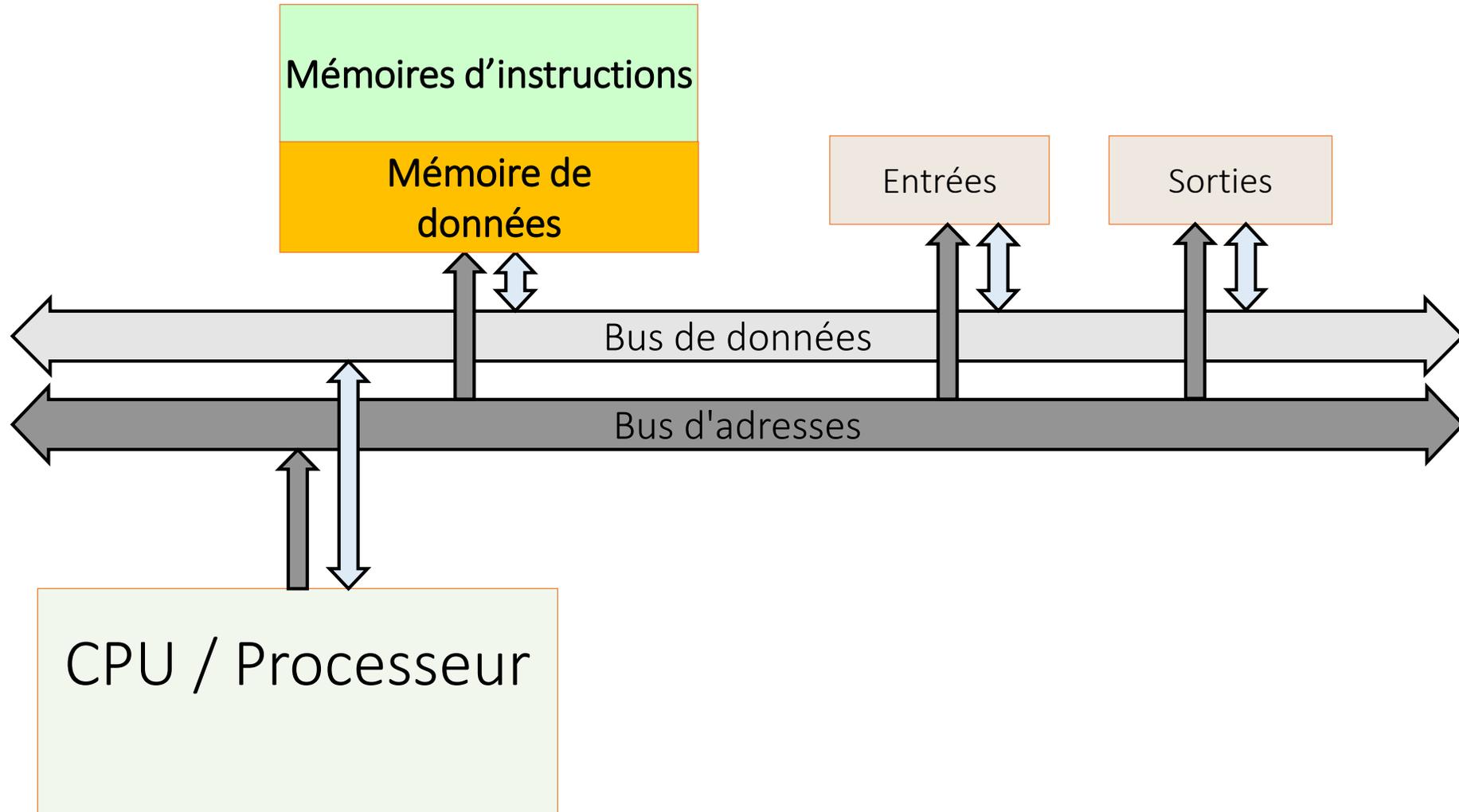
- Entrées : par exemple, le clavier ou la souris
- Sorties : par exemple l'écran ou le son

Exercice :

- *Dessinez les schémas correspondants pour les architectures Von Neumann et Harvard*
- *Citez un avantage de Harvard par rapport à Von Neumann*

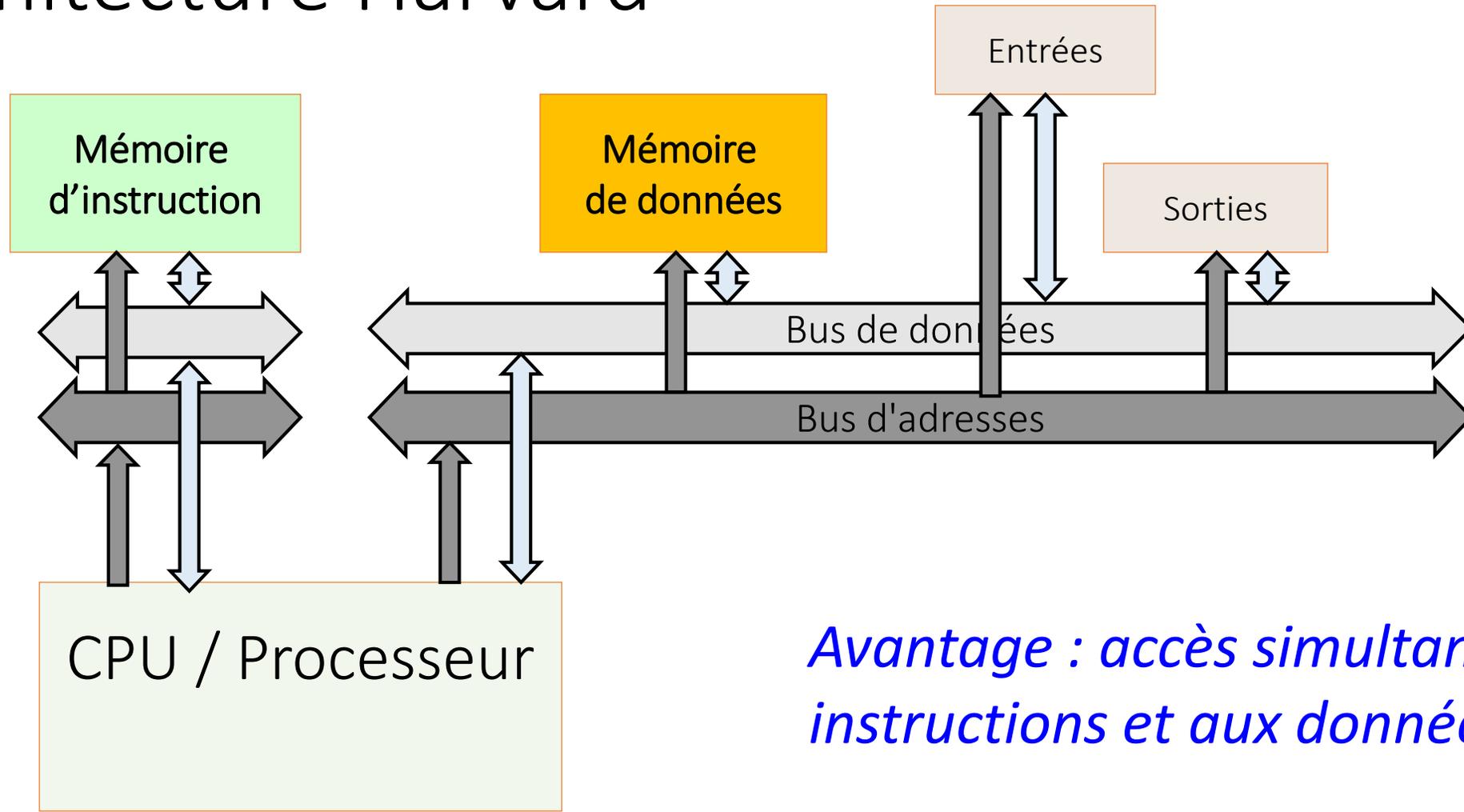
Architecture Von Neumann

Architecture Von Neumann



Architecture Harvard

Architecture Harvard



Avantage : accès simultané aux instructions et aux données

Instructions (1)

RISC v.s. CISC

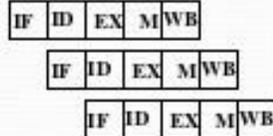
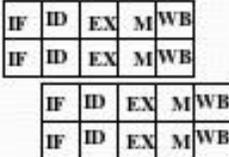
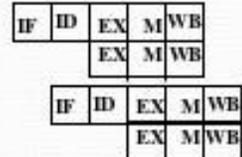
RISC = Reduced Instruction Set Computer

CISC = Complex Instruction Set Computer

Parameter	RISC	CISC
Instruction types	Simple	Complex
Number of instructions	Reduced (30-40)	Extended (100-200)
Duration of an instruction	One cycle	More cycles (4-120)
Instruction format	Fixed	Variable
Instruction execution	In parallel (pipeline)	Sequential
Addressing modes	Simple	Complex
Instructions accessing the memory	Two: Load and Store	Almost all from the set
Register set	multiple	unique
Complexity	In compiler	In CPU (micro-program)

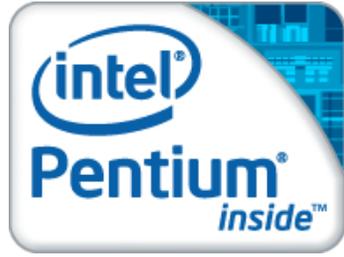
Instructions (2)

CISC vs RISC vs SS vs VLIW

	CISC	RISC	Superscalar	VLIW
Instruction size	variable size	fixed size	fixed size	fixed size (but large)
Instruction format	variable format	fixed format	fixed format	fixed format
Registers	few, some special	many GP	GP and rename (RUU)	many, many GP
Memory reference	embedded in many instr's	load/store	load/store	load/store
Key Issues	decode complexity	data forwarding, hazards	hardware dependency resolution	code scheduling, (compiler)
Instruction flow				

	Acronyme
IF	Instruction Fetch
ID	Instruction Decode Register Fetch
EX	Execute Address Calc.
MEM	Memory Access
WB	Write Back

Types de microprocesseurs



	Von Neumann	Harvard
RISC		
CISC		



ARM 7

ARM 9

Lequel choisir pour les systèmes embarqués ?

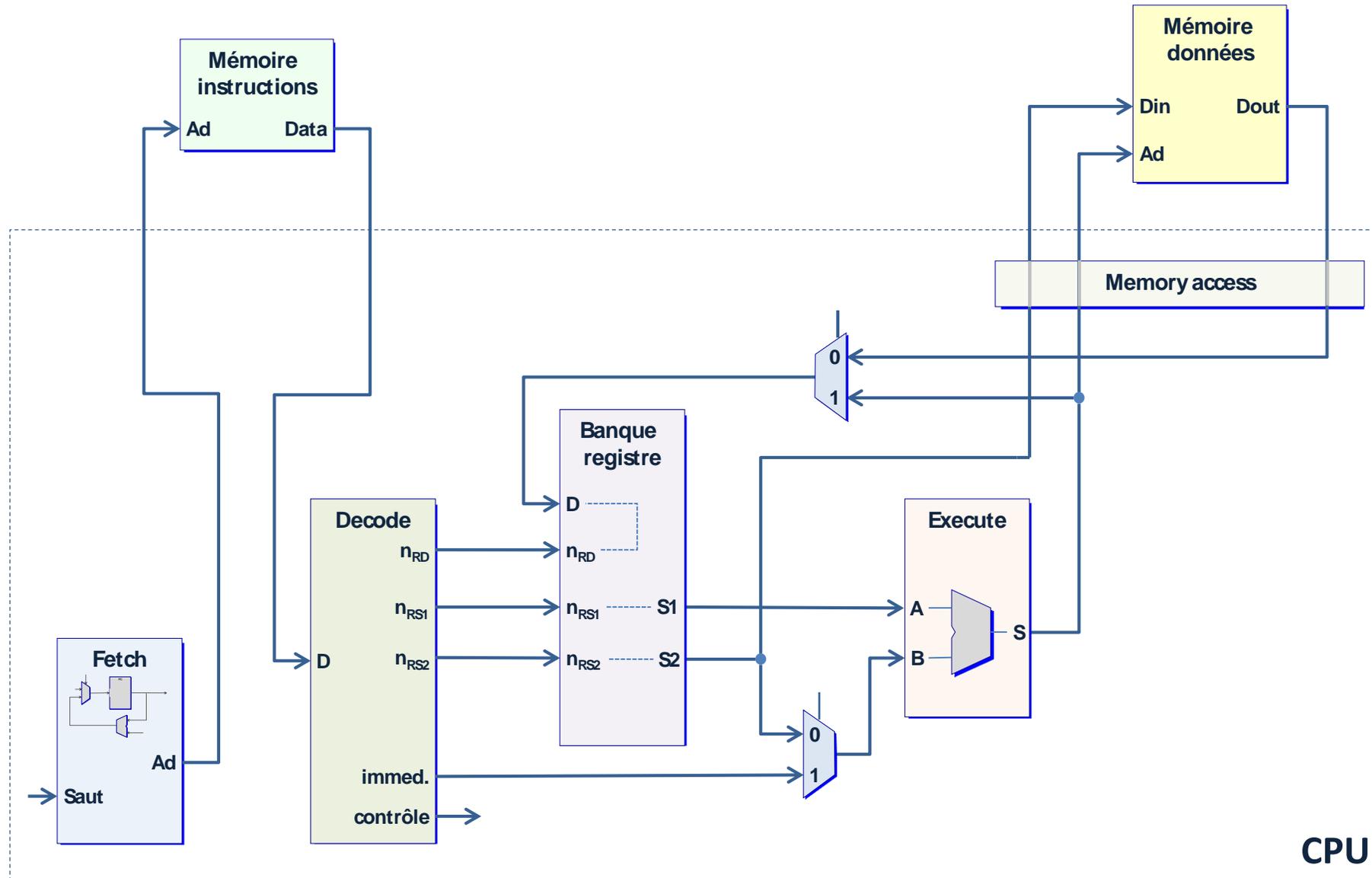
Types de microprocesseurs

	Von Neumann	Harvard
RISC	ARM 7 MIPS TECHNOLOGIES	ARM 9
CISC		

En général RISC (MIPS, très basse consommation d'énergie)

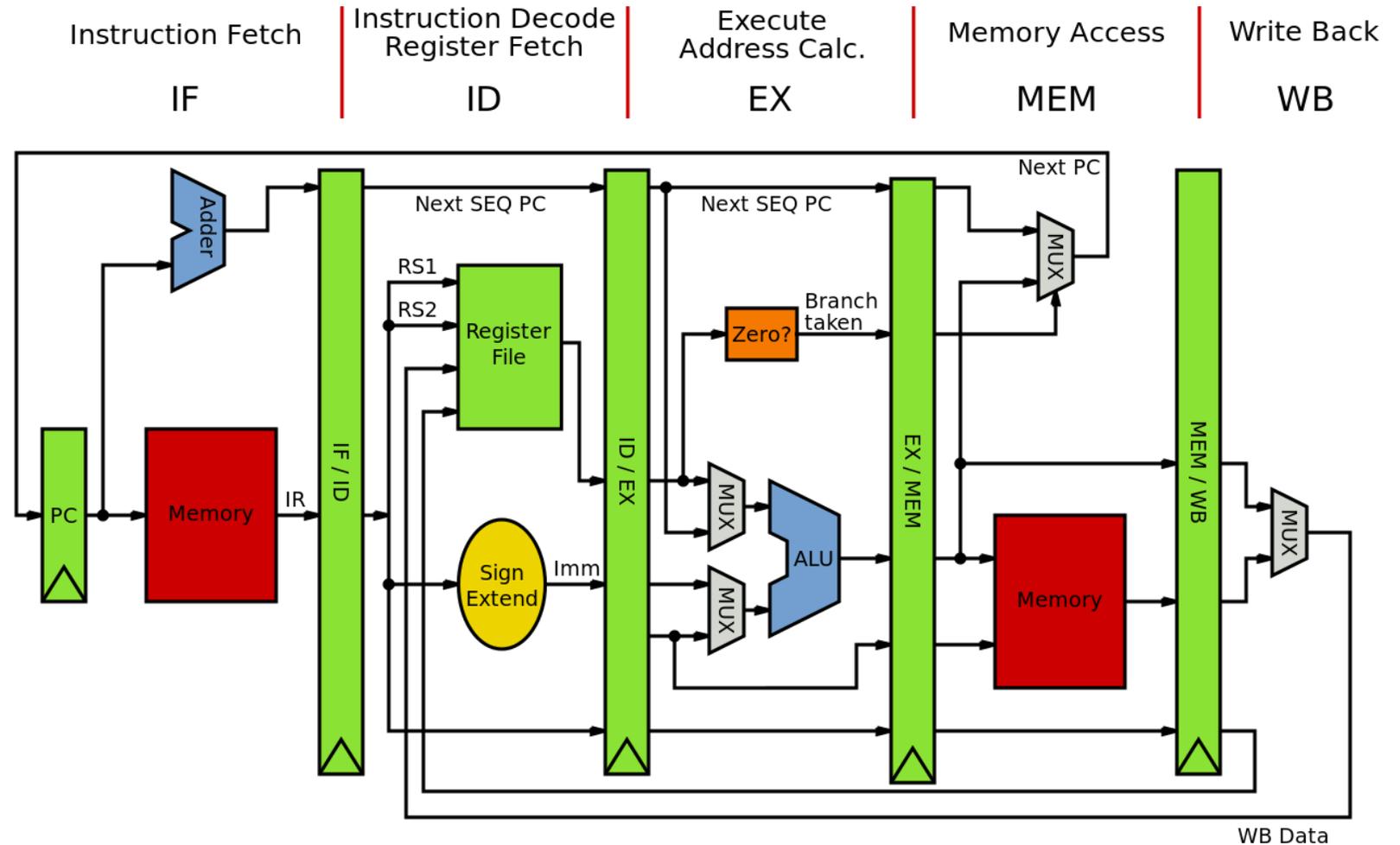
Types de microprocesseurs (2)

Lequel ?

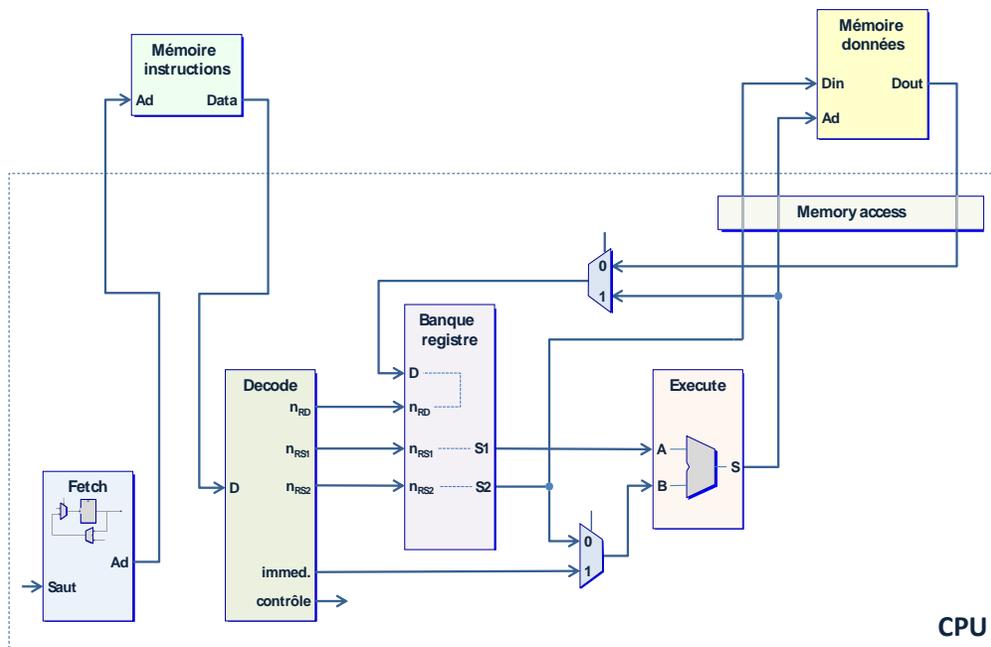


Types de microprocesseurs (3)

- *Est-ce de type von Neumann ou Harvard ?*
- *Reconnaissez les 5 fonctions du processeur*
- *A quoi correspondent les parties vertes ?*



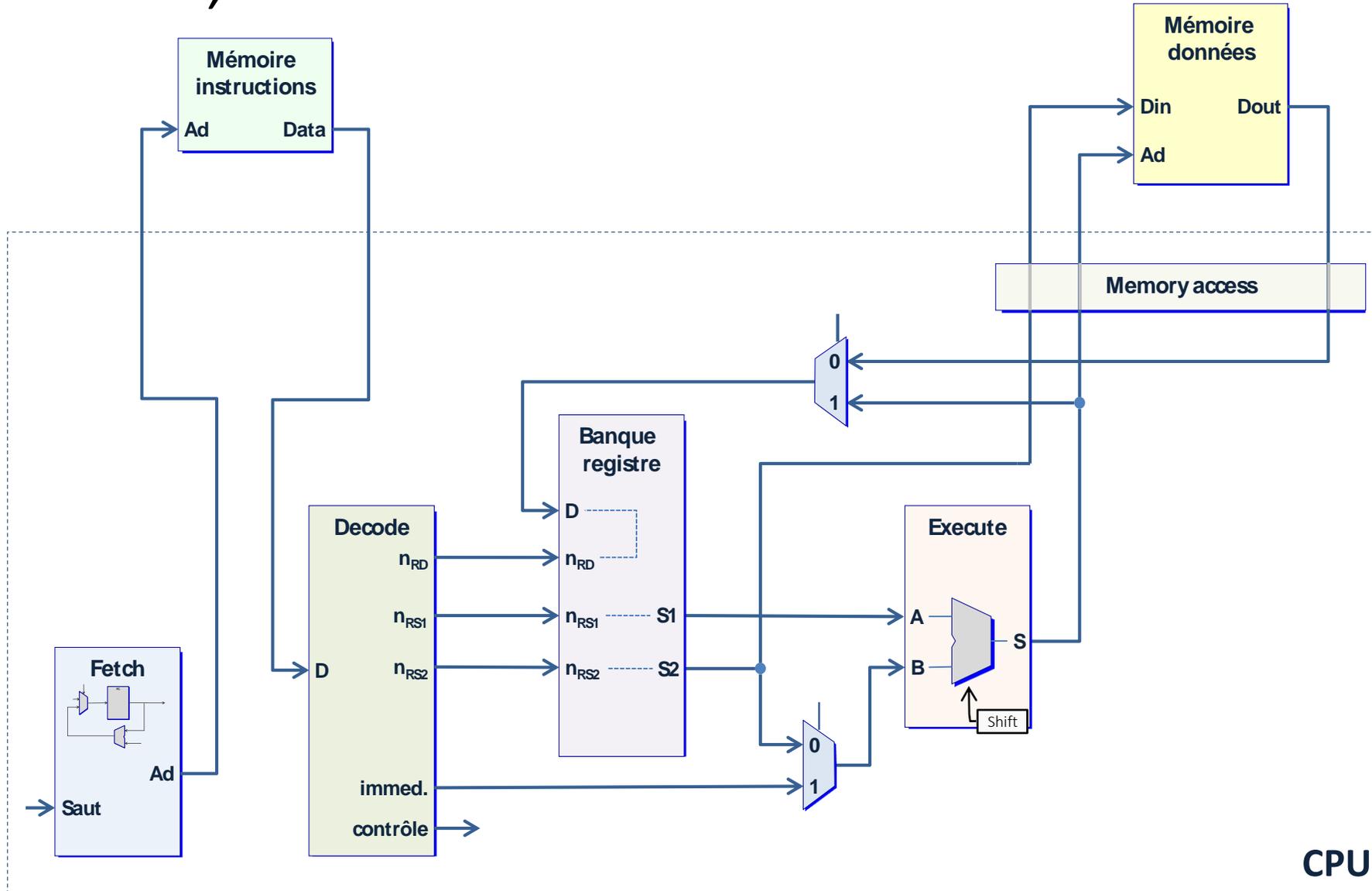
Exercice 1 : Microprocesseur type ARM



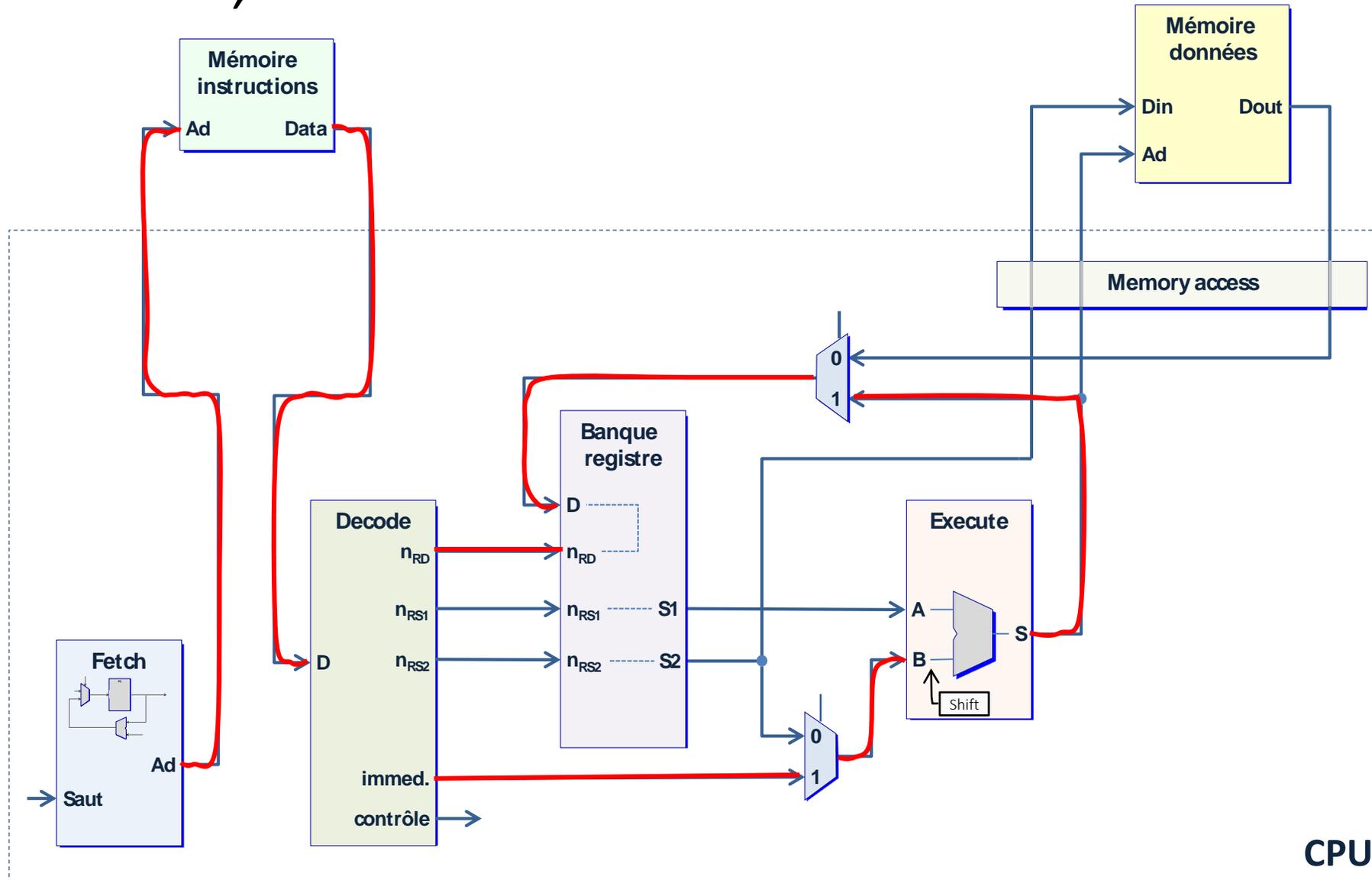
• Compléter les schémas avec les valeurs correspondant aux exemples ci-dessous :

1. MOV R3, #34
2. R4 = 26, MOV R2, R6
3. R3 = 12, ADD R2, R3, #3
4. R4 = 14, R6 = 7, ADD R1, R4, R6
5. R2 = 0x200, LDRH R3, [R2, #3 * 2]
6. R6 = 37, R2 = 0x 220, STRH R6, [R5,#4 * 2]
7. PC = 0x140, Label = 0x150, B Label

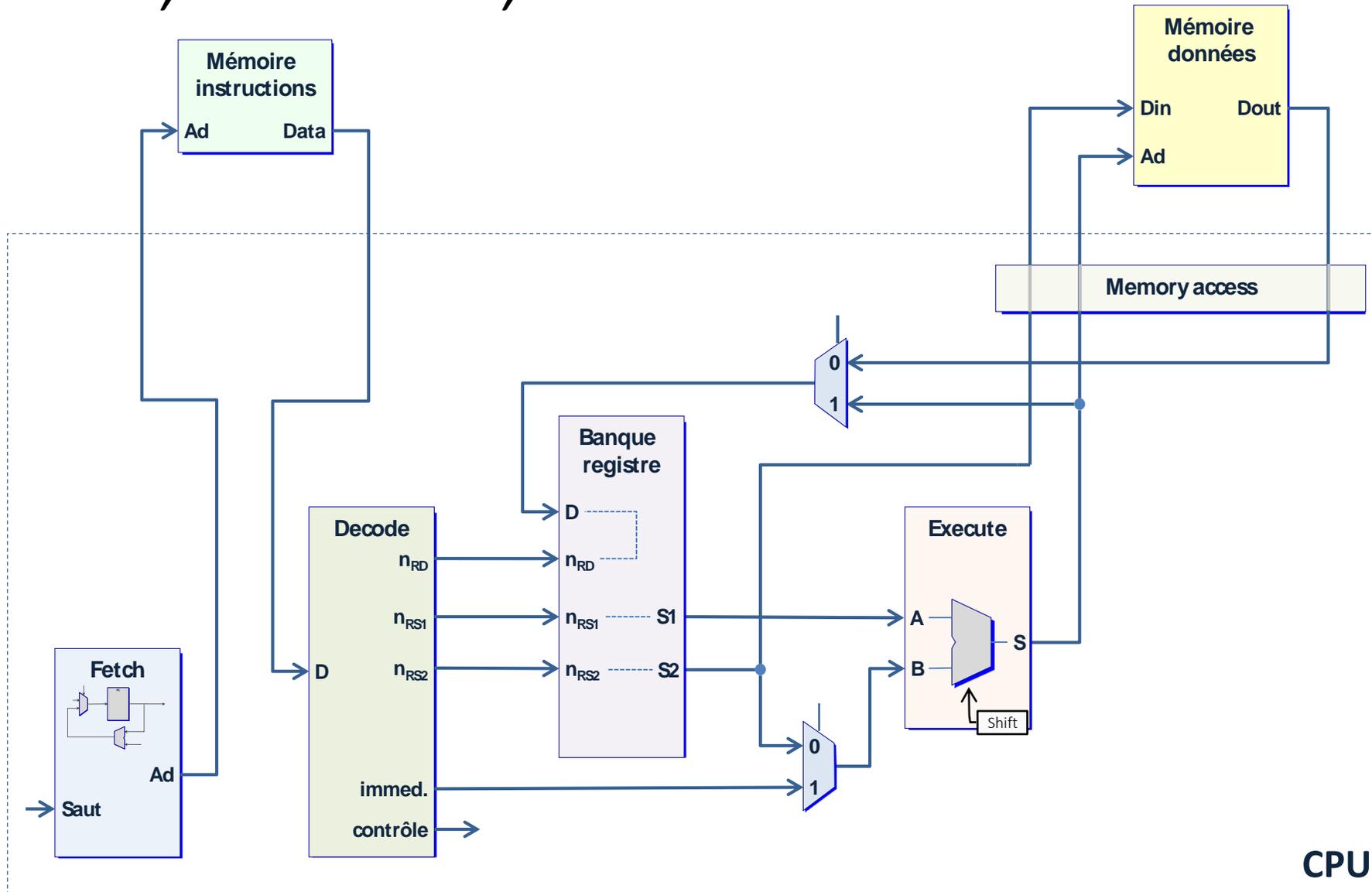
MOV R3, #34



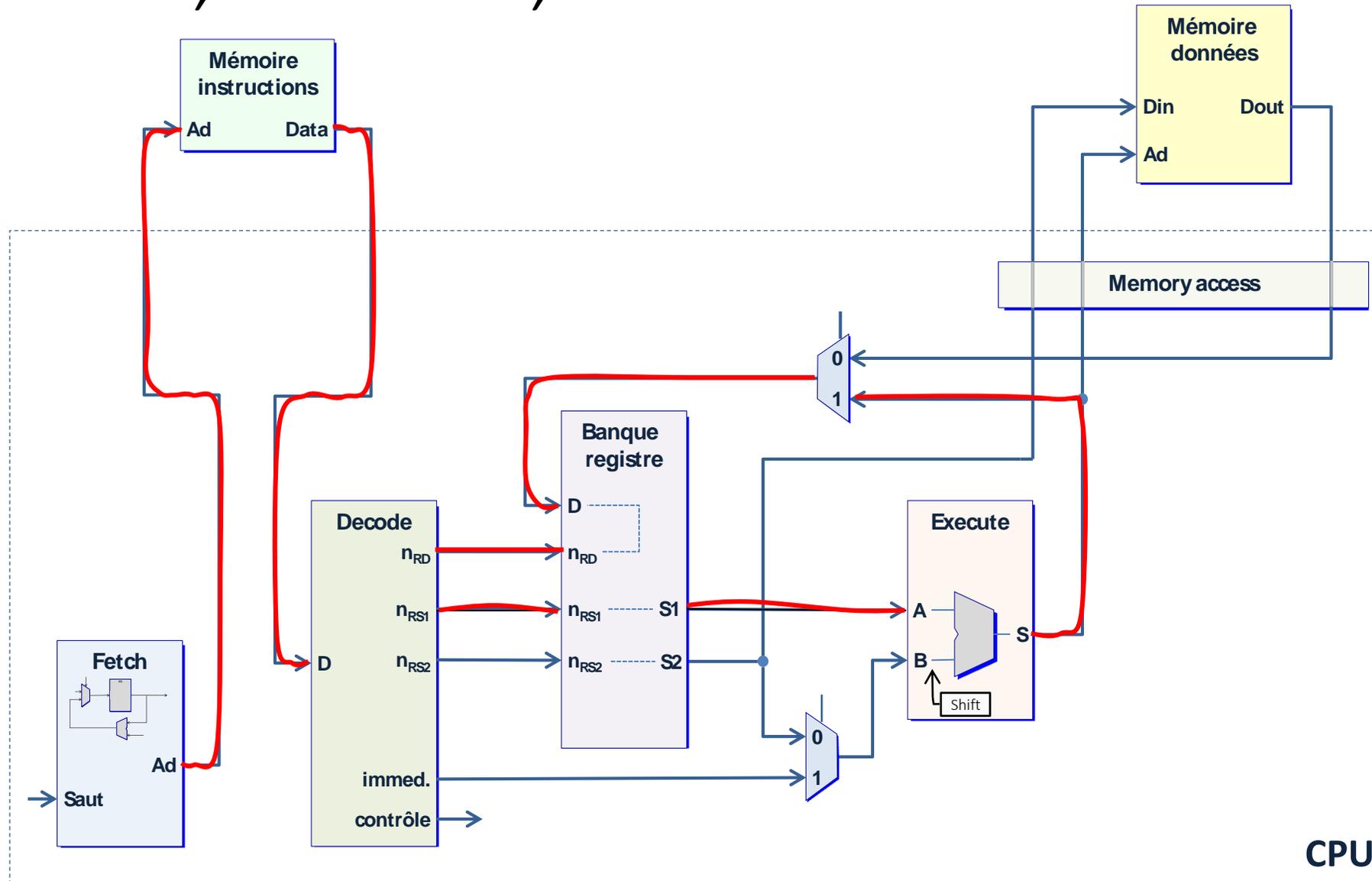
MOV R3, #34



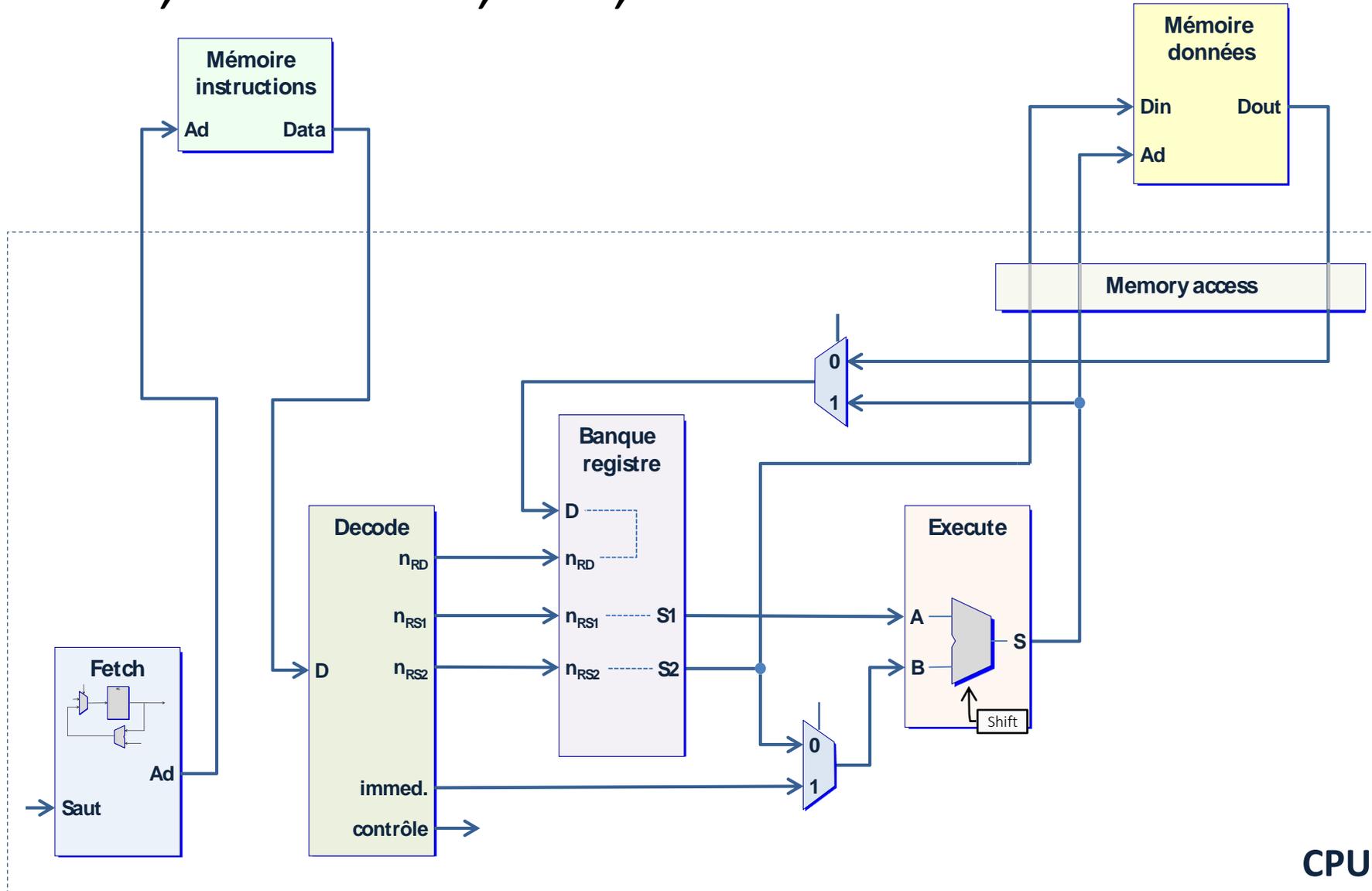
R4 = 26, MOV R2, R6



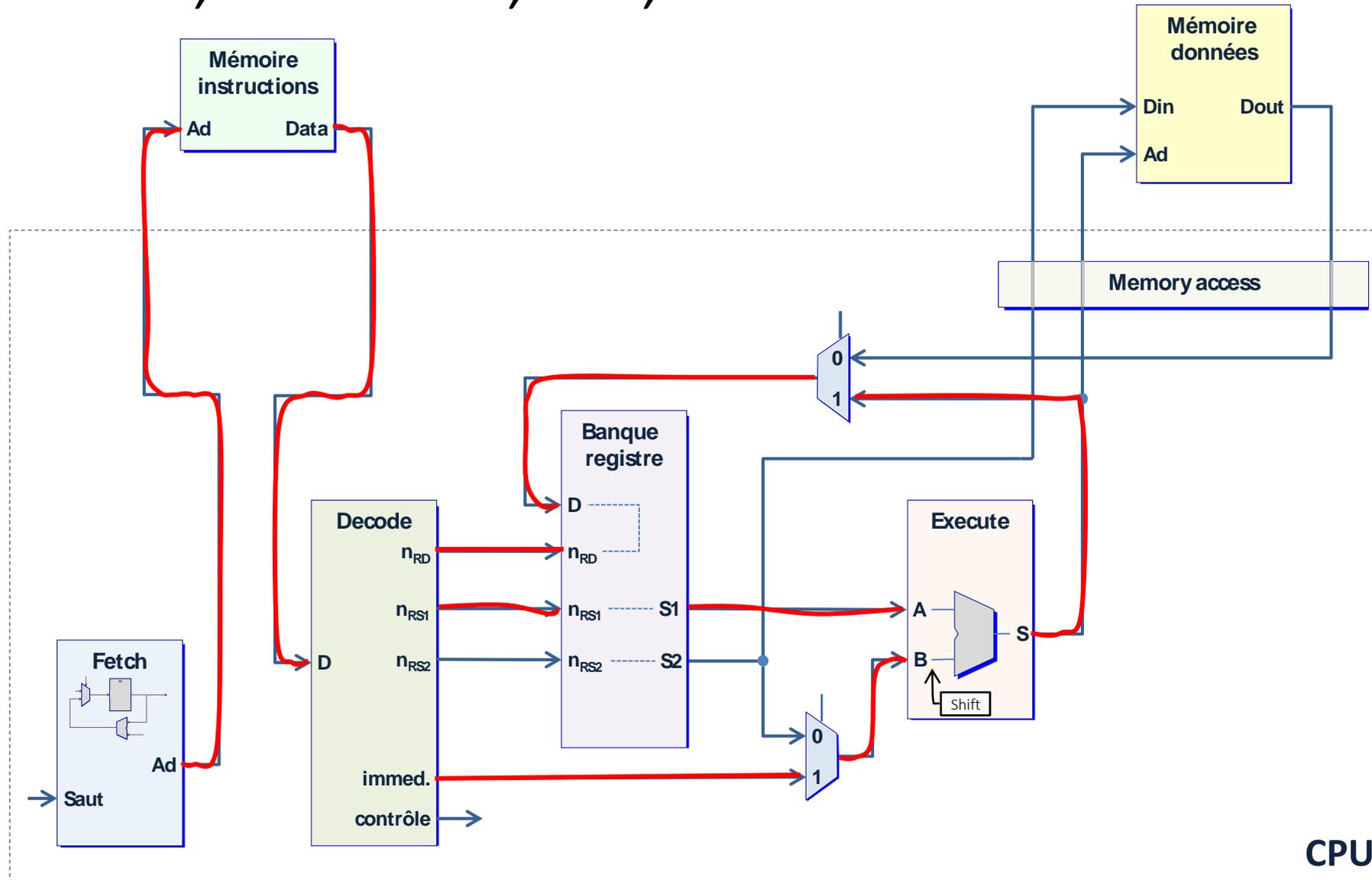
R4 = 26, MOV R2, R6



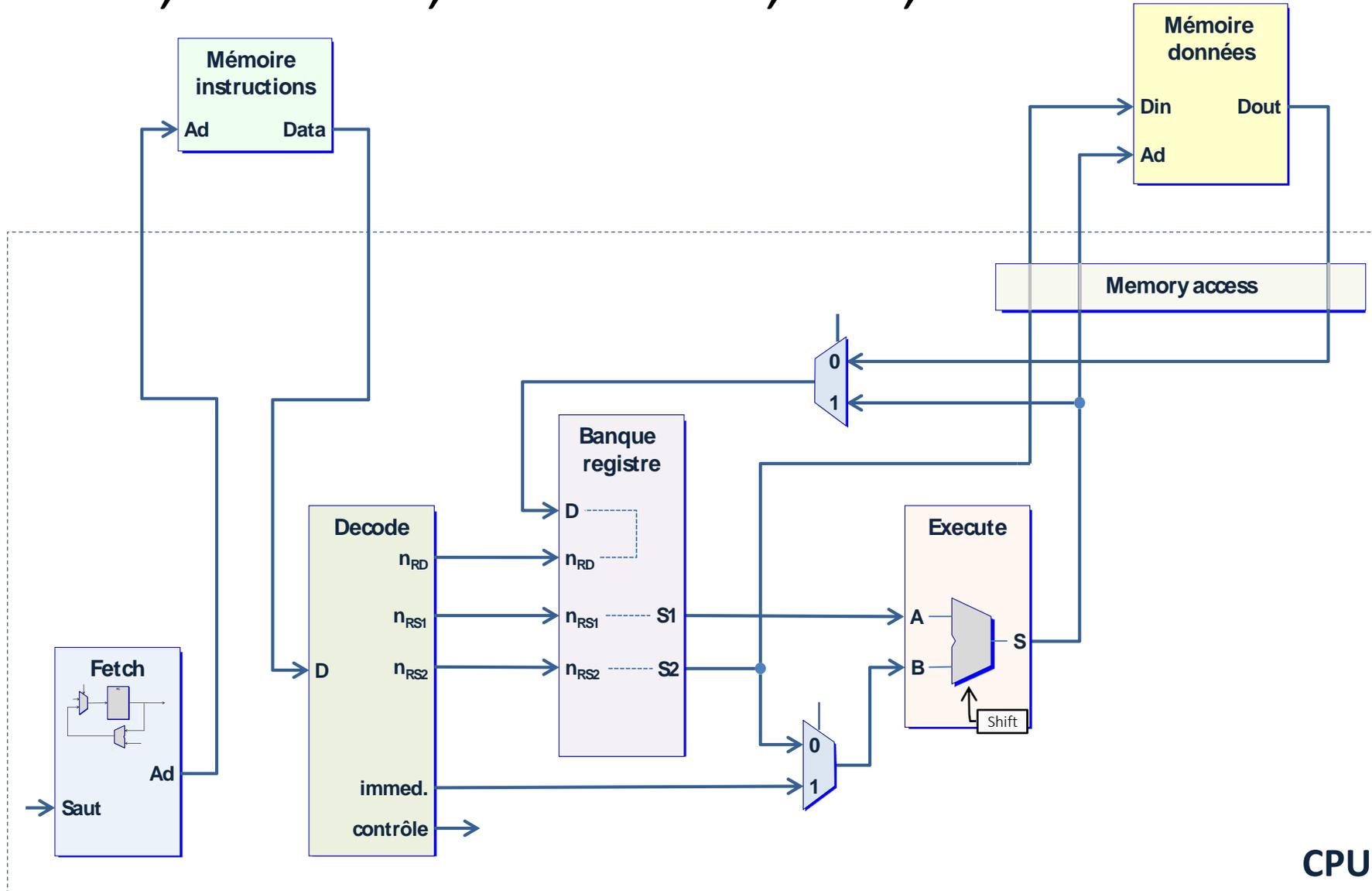
R3 = 12, ADD R2, R3, #3



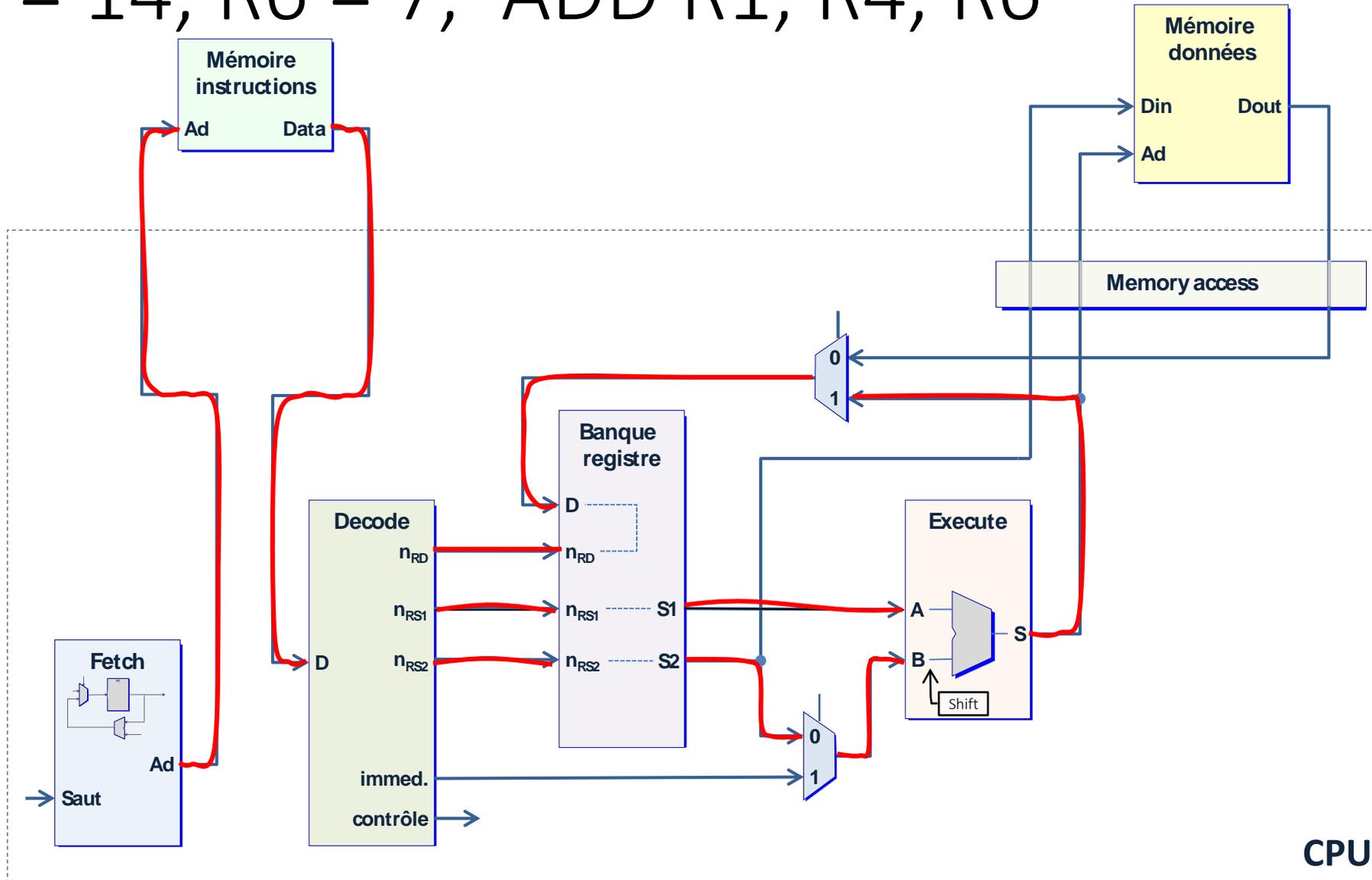
R3 = 12, ADD R2, R3, #3



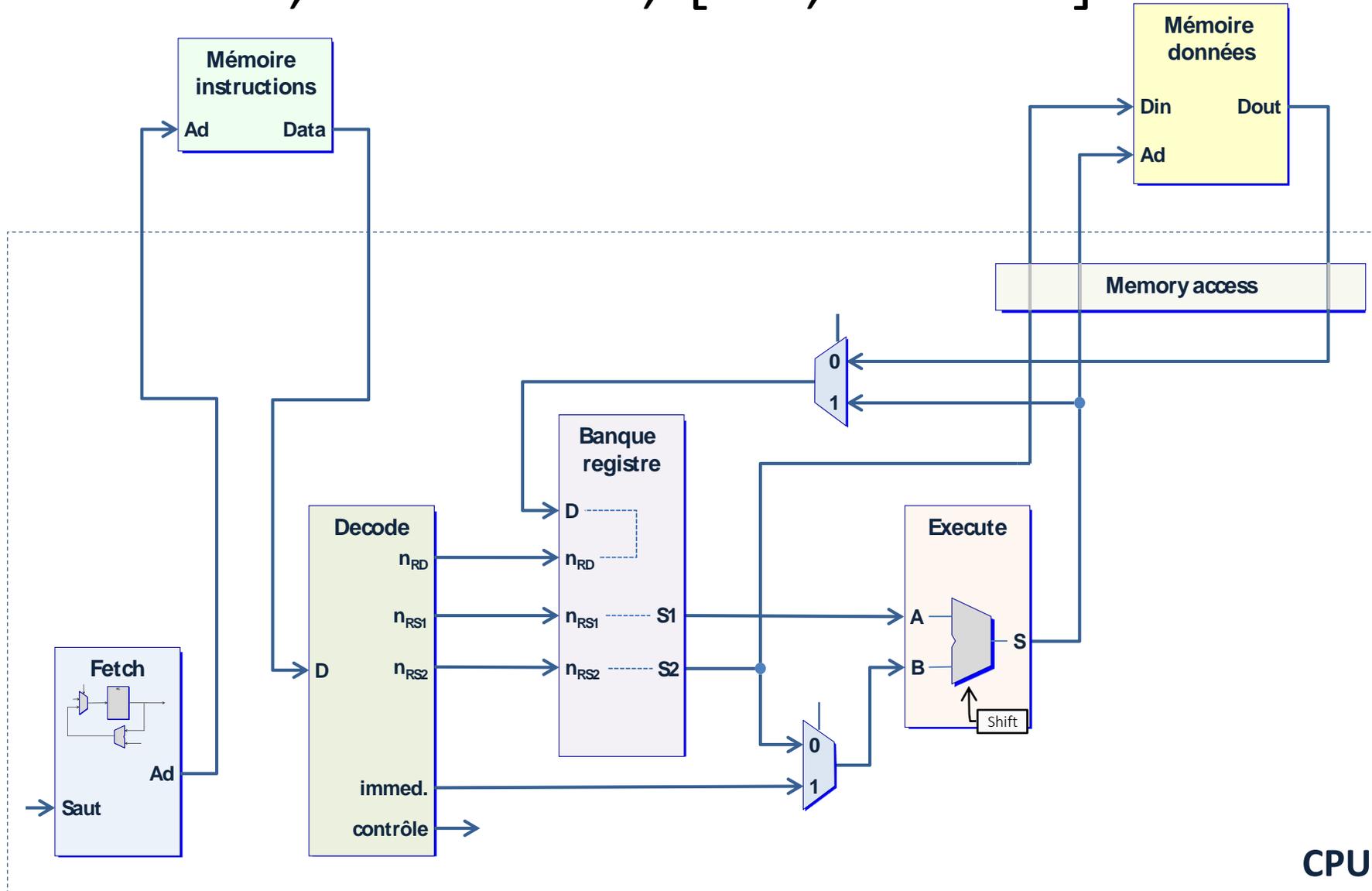
R4 = 14, R6 = 7, ADD R1, R4, R6



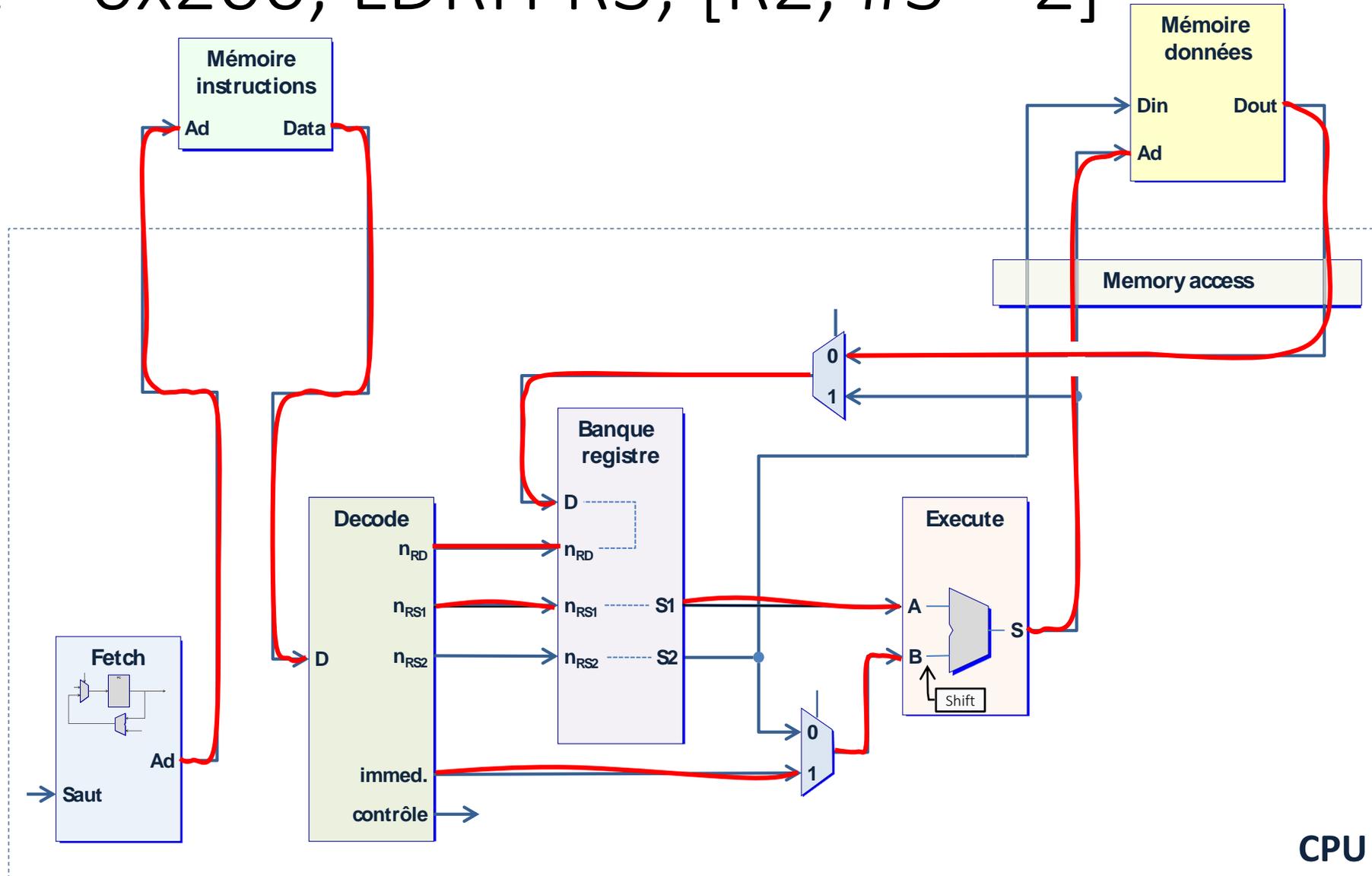
R4 = 14, R6 = 7, ADD R1, R4, R6



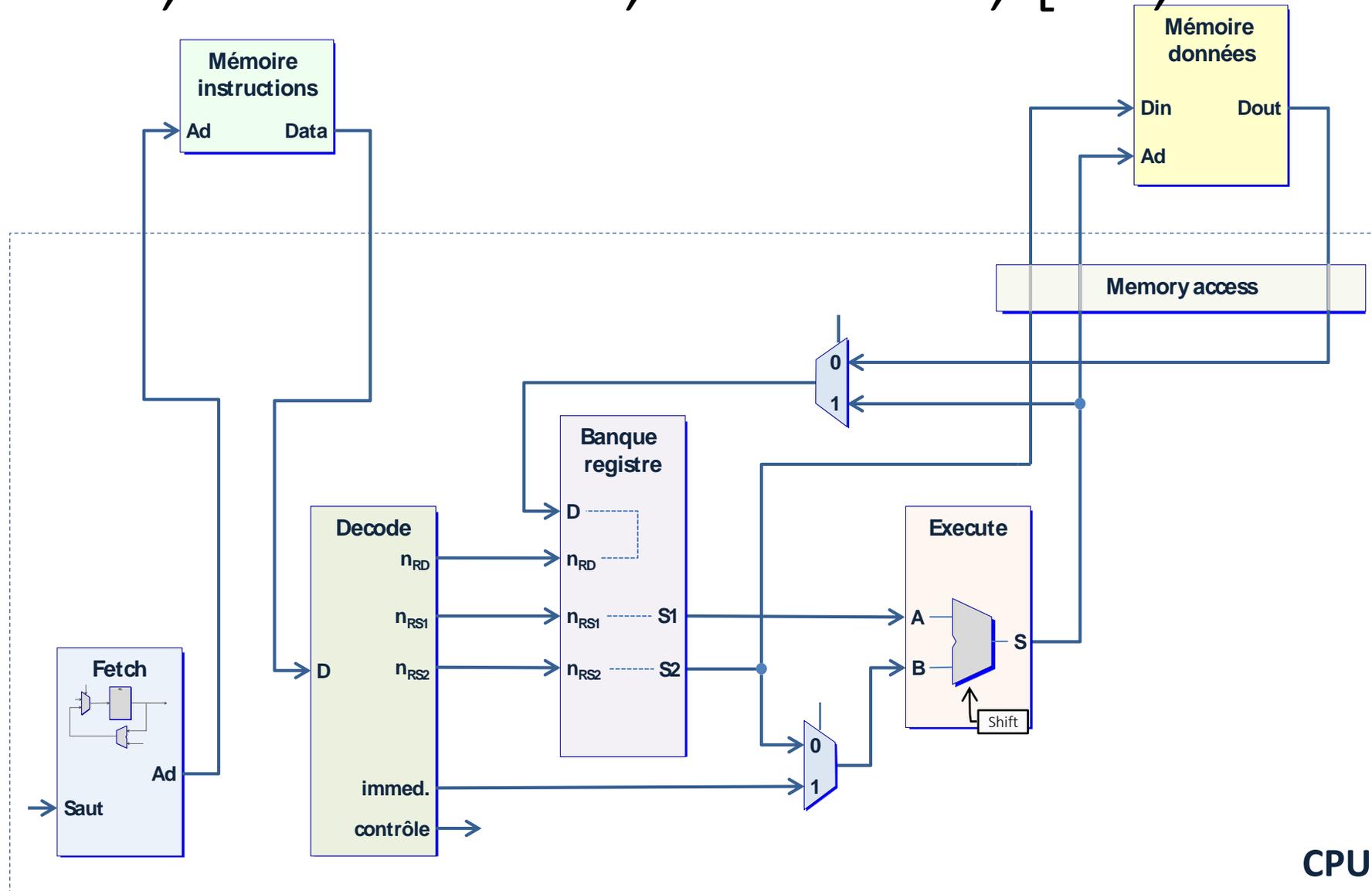
R2 = 0x200, LDRH R3, [R2, #3 * 2]



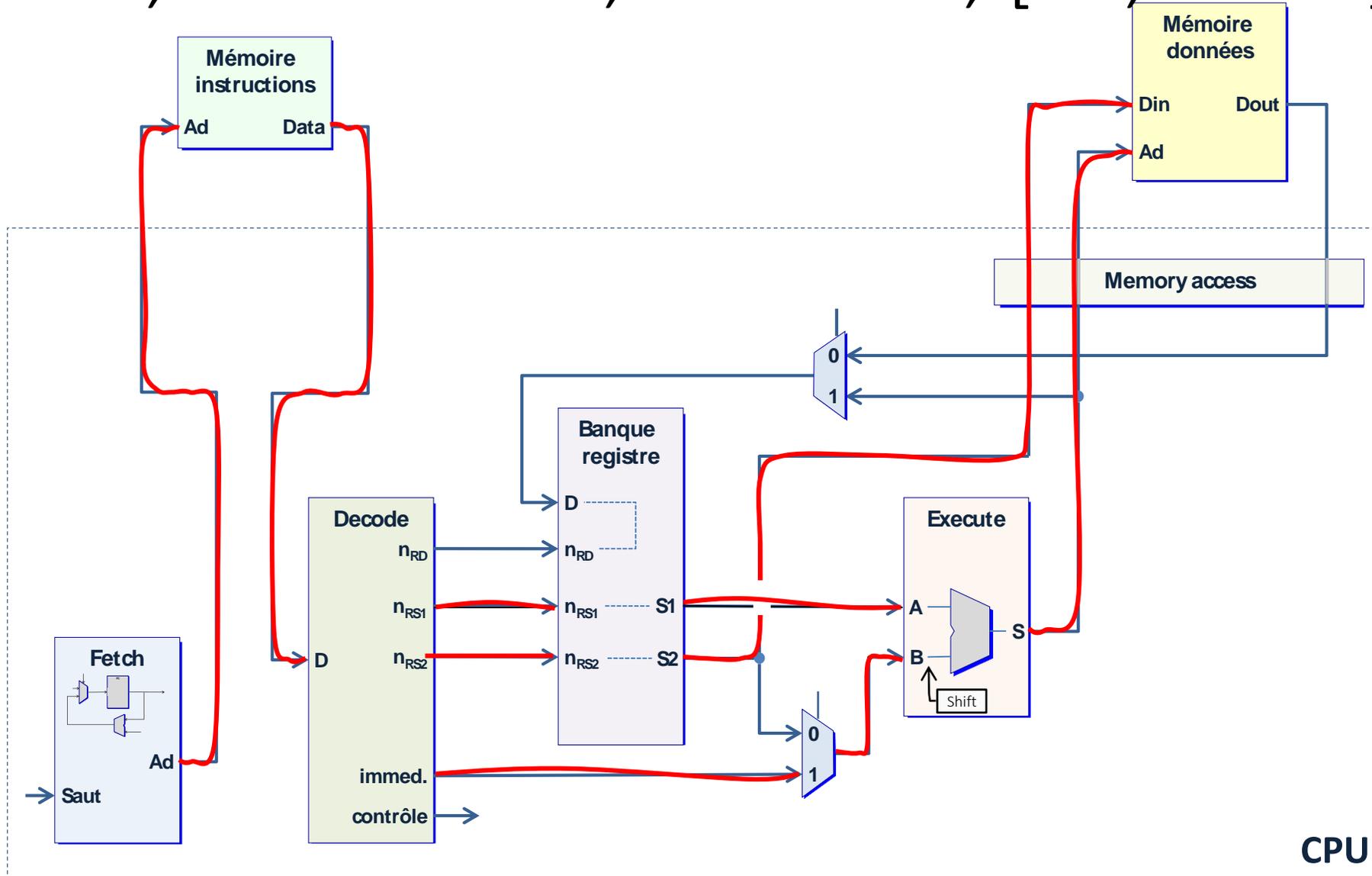
R2 = 0x200, LDRH R3, [R2, #3 * 2]



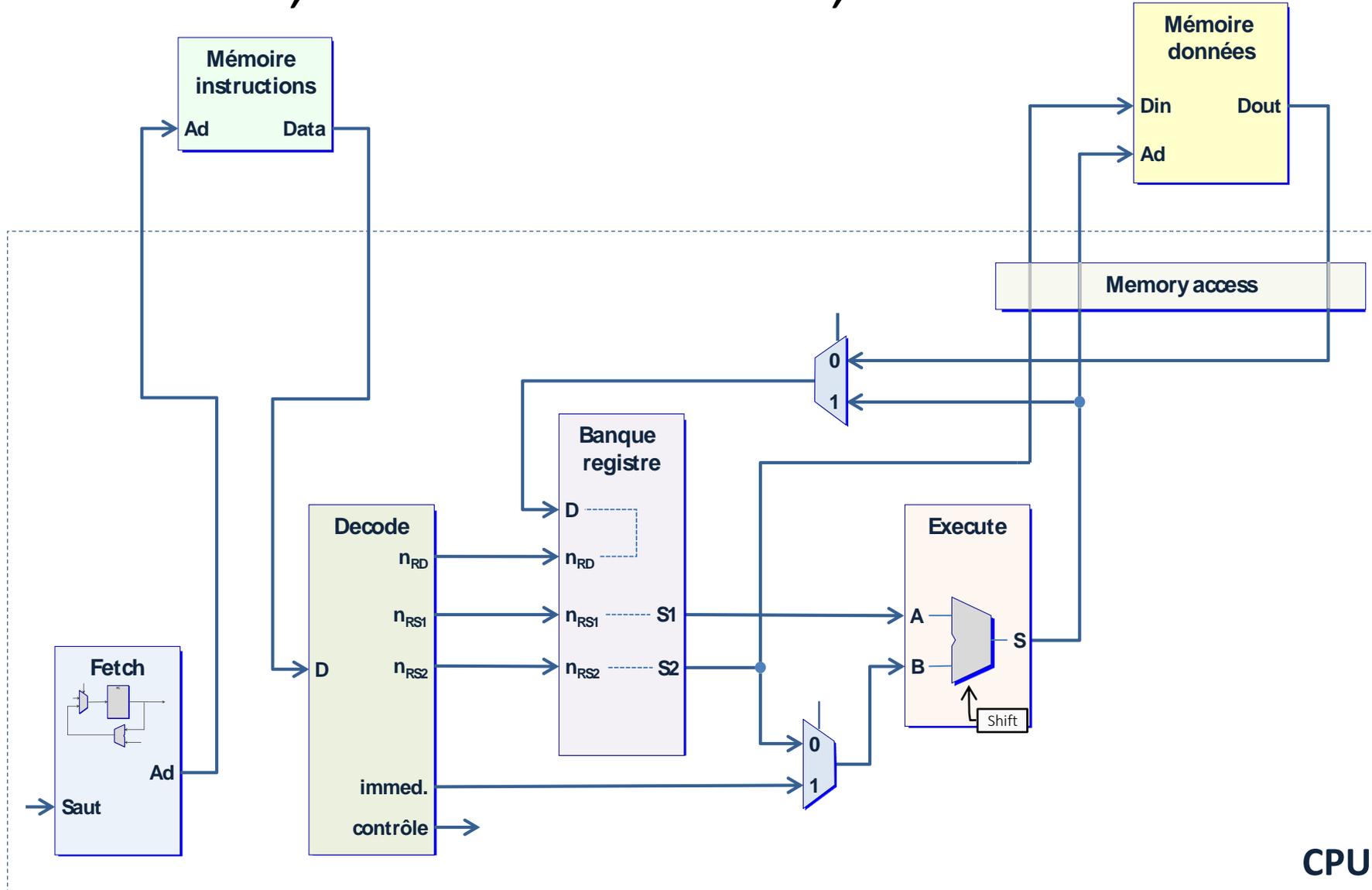
R6 = 37, R2 = 0x 220, STRH R6, [R5,#4 * 2]



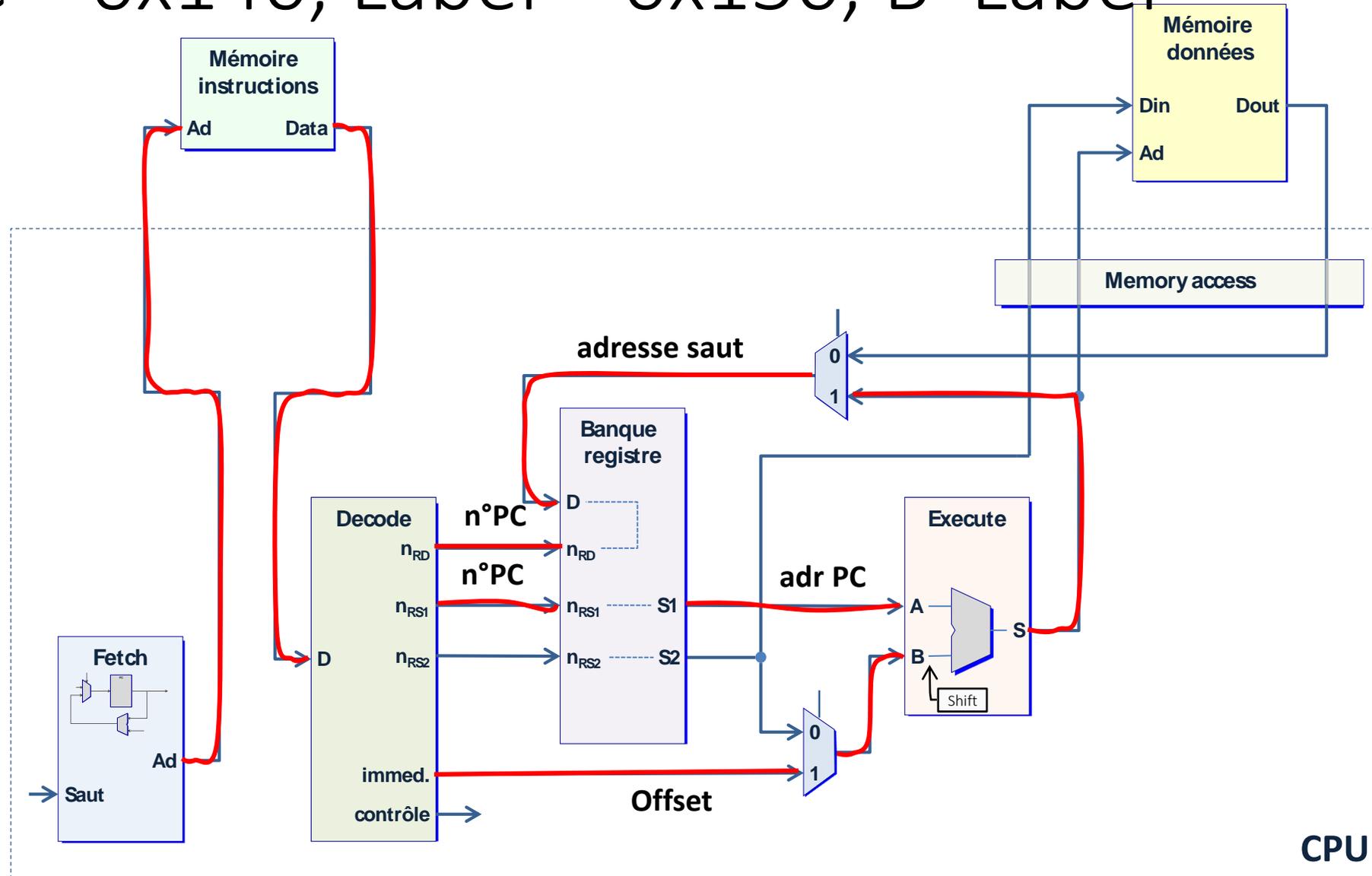
R6 = 37, R2 = 0x 220, STRH R6, [R5,#4 * 2]



PC = 0x140, Label = 0x150, B Label



PC = 0x140, Label = 0x150, B Label

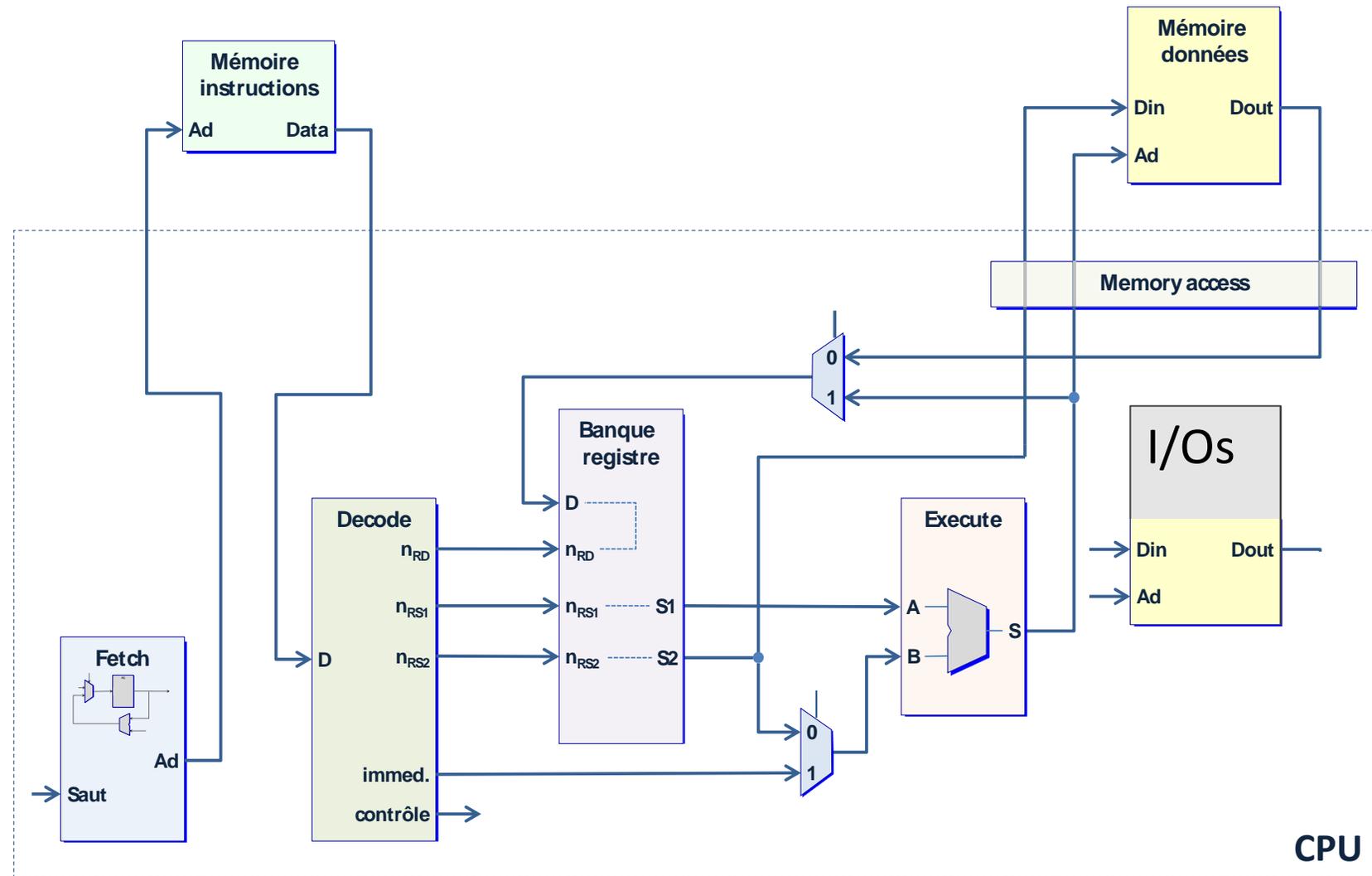


Exercice 2 : Microprocesseur type ARM

- Comment interfacer les périphériques I/O ?
- Un périphérique a pour adresse de base 0x40001000 et comprend 5 registres 32 bits :

1. 0x40001000
2. 0x40001004
3. 0x40001008
4. 0x4000100C
5. 0x40001010

Quelle est l'adresse du périphérique suivant au minimum ?

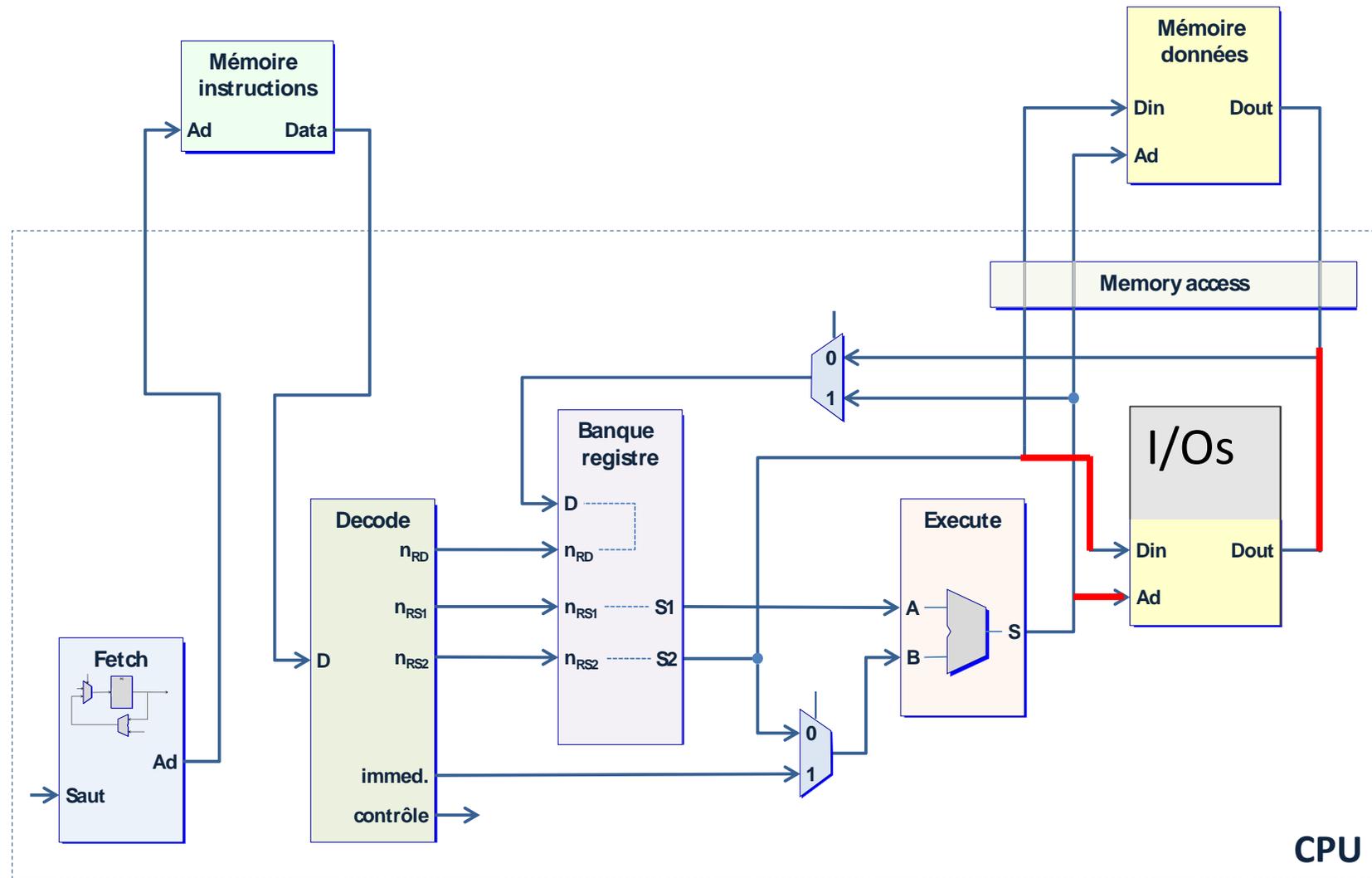


Exercice 2 : Microprocesseur type ARM

- Comment interfacer les périphériques I/O ?
- Un périphérique a pour adresse de base 0x40001000 et comprend 5 registres 32 bits :

1. 0x40001000
2. 0x40001004
3. 0x40001008
4. 0x4000100C
5. 0x40001010

Quelle est l'adresse du périphérique suivant au minimum ? 0x40001020



Le processeur embarqué

Le processeur embarqué

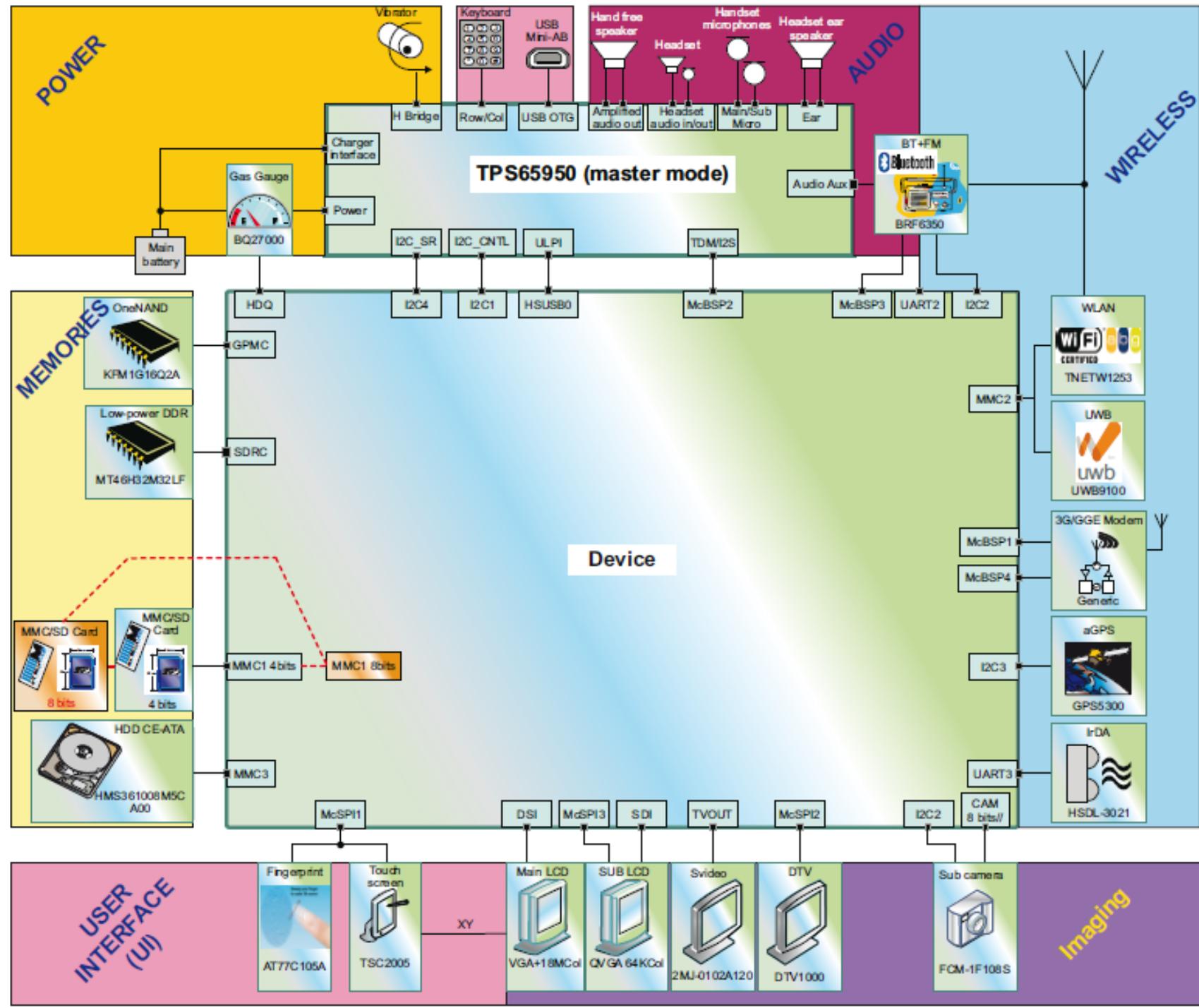
- Aussi nommé microcontrôleur
- Constitué de :
 - un cœur de processeur (ou plusieurs)
 - des périphériques (entrées /sorties ou IOs)
 - Contrôleur écran
 - Contrôleur USB
 - de la mémoire

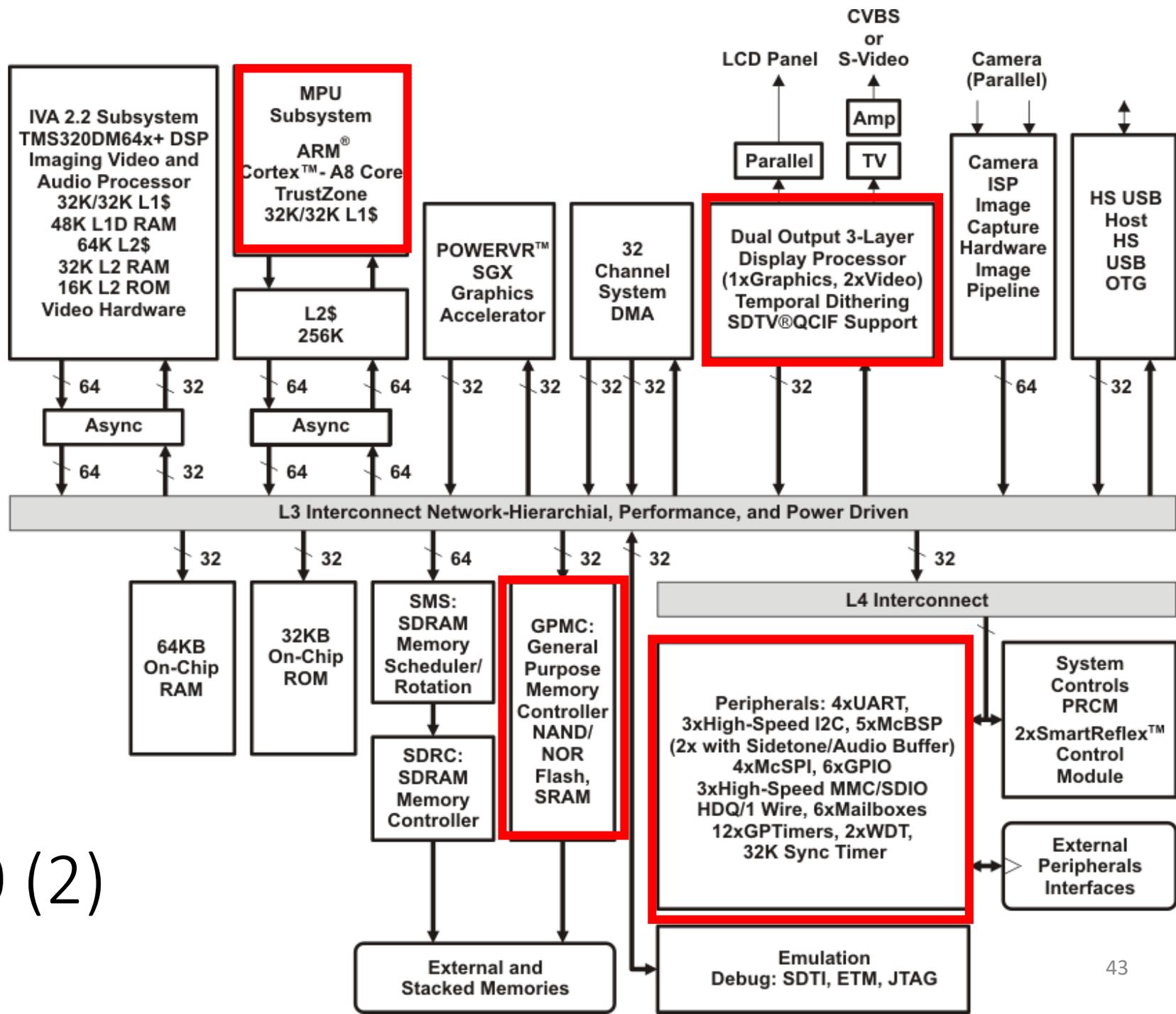


- Contrainte de performance, notion de temps réel
- Problème de la densité de code : taille du code applicatif (mémoire), la mémoire tend à occuper la majeure partie d'une puce, impact de la taille du coeur dans le choix du processeur

Le processeur OMAP/DM3730 (1)

- OMAP = Open Multimedia Applications Platform





Le processeur
OMAP/DM3730 (2)

Entrées/Sorties (IOS)

- Accessibles comme la mémoire en lecture et écriture (mêmes instructions)
- Un périphérique d'entrées /sorties est vu comme un ensemble de registres avec une adresse mémoire pour chaque registre => memory mapped IOs
- Exemple : GPIOs de l'OMAP

Exemple: GPIOs de l'OMAP

- Périphérique le plus courant: le GPIO (General Purpose IO)
- Adresses registres GPIO4:

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
GPIO_REVISION	R	32	0x000	0x4905 4000
GPIO_SYSCONFIG	RW	32	0x010	0x4905 4010
.....				
GPIO_CLEARDATAOUT	RW	32	0x090	0x4905 4090
GPIO_SETDATAOUT	RW	32	0x094	0x4905 4094

- 26 registres
- Adresse = Adresse_GPIO4(0x49054000) + Offset
- Adresse GPIO5

GPIO_REVISION	R	32	0x000	0x4905 6000
GPIO_SYSCONFIG	RW	32	0x010	0x4905 6010

