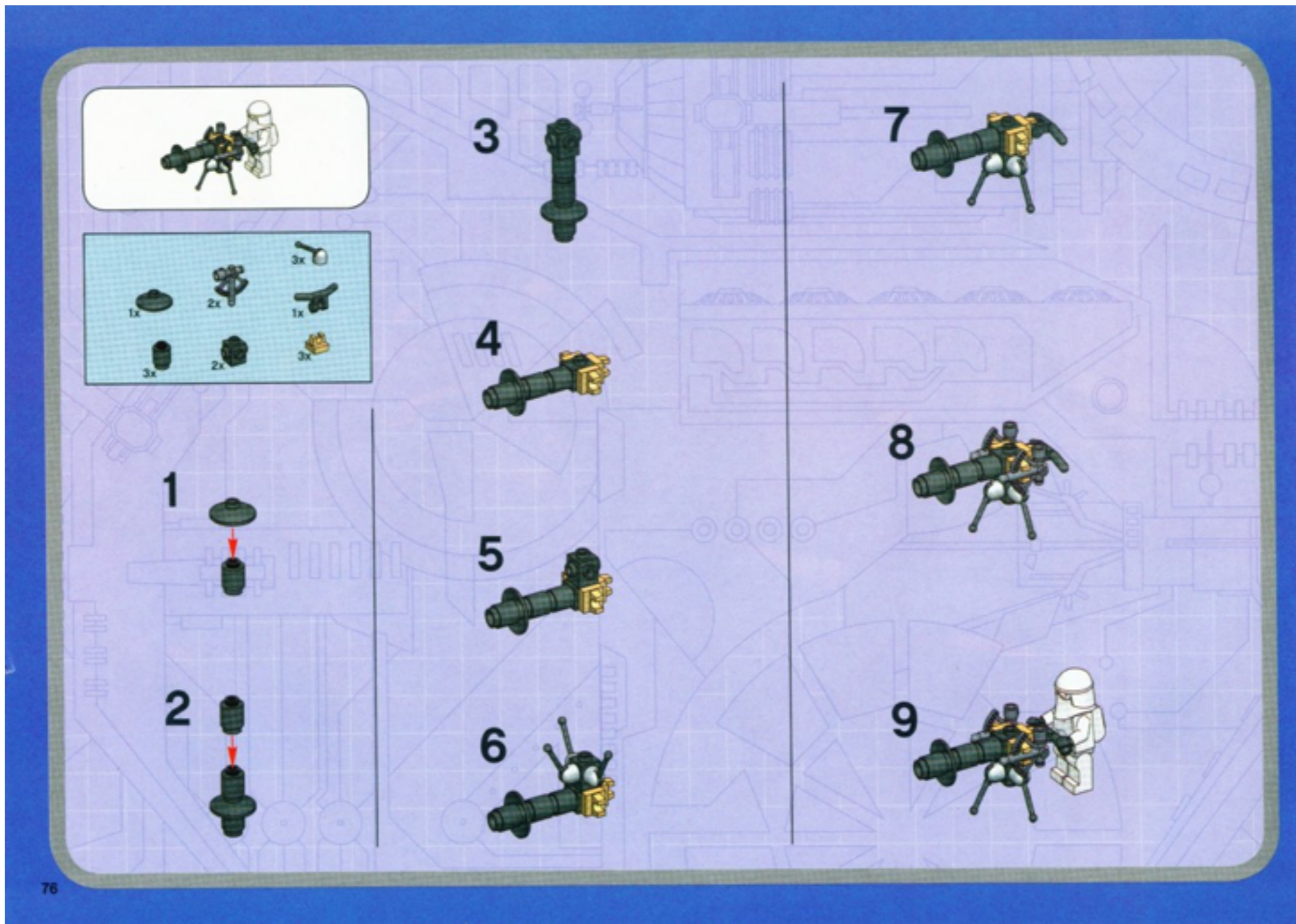


Micro-instructions



Ordinateurs: Structure et Applications, Hiver 2017
Jean-François Lalonde

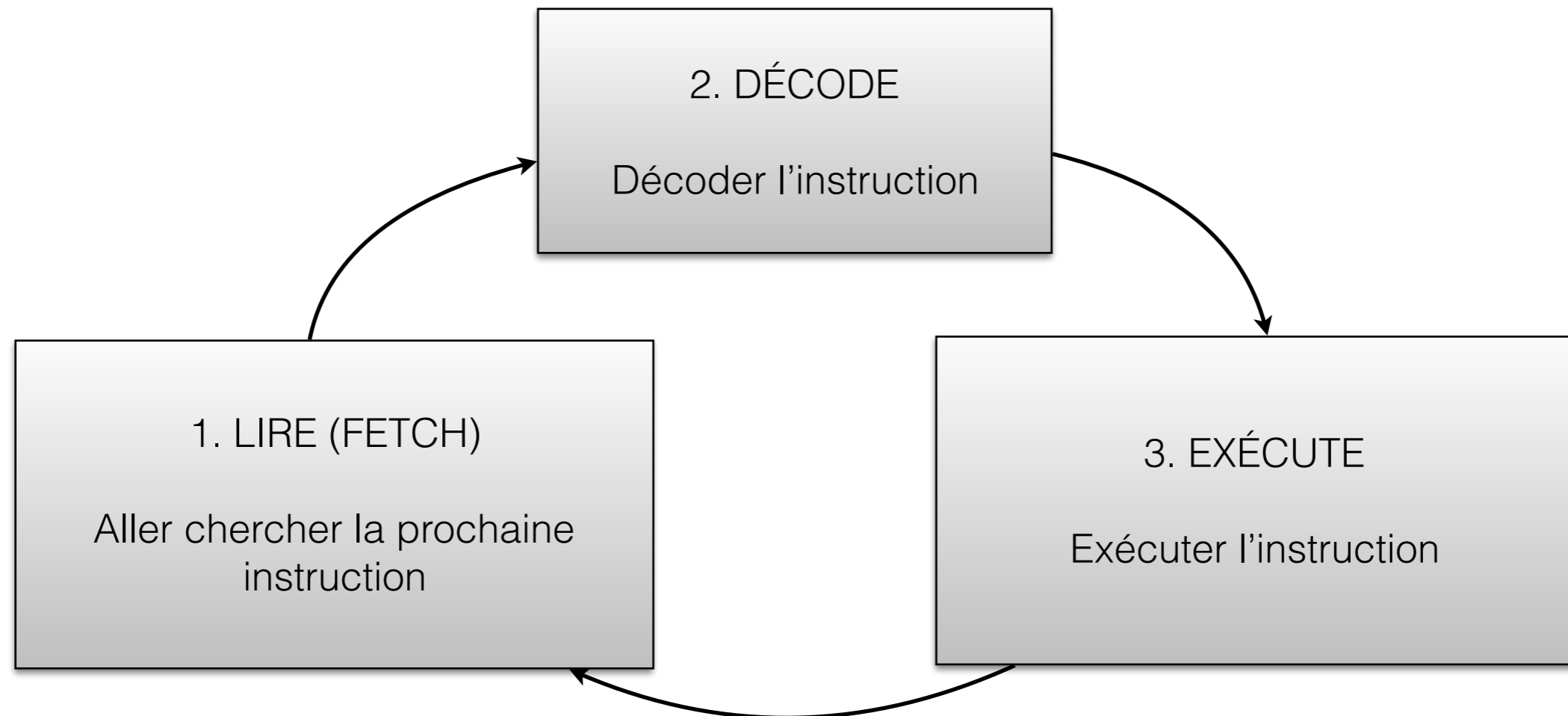
Instructions vs micro-instructions

- Une instruction est une « action » pouvant être exécutée par le microprocesseur.
- Une instruction n'est pas atomique: elle est divisée en une séquence de « micro-instructions »
- Ces micro-instructions dépendent de la structure interne, et du cycle d'instructions du microprocesseur

Cycle d'instructions

Que fait le microprocesseur?

1. Lire: aller chercher la prochaine instruction
2. Décode: décode l'instruction (détermine ce qu'il y a à faire)
3. Exécute: exécuter l'instruction

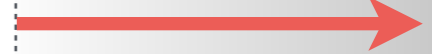


Structure interne

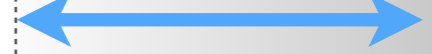
Intérieur du
microprocesseur

Extérieur du
microprocesseur

Bus d'adresses



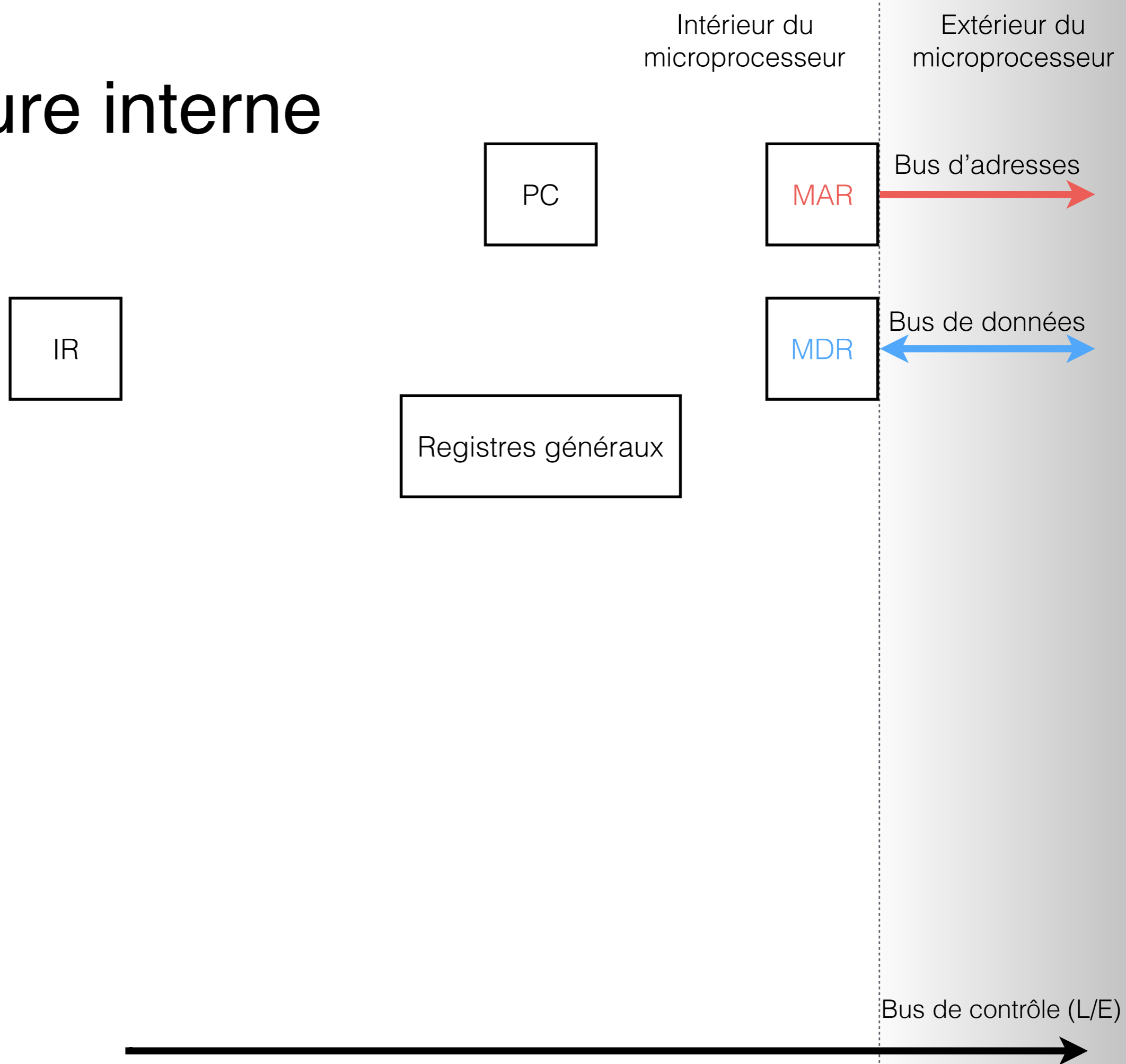
Bus de données



Bus de contrôle (L/E)



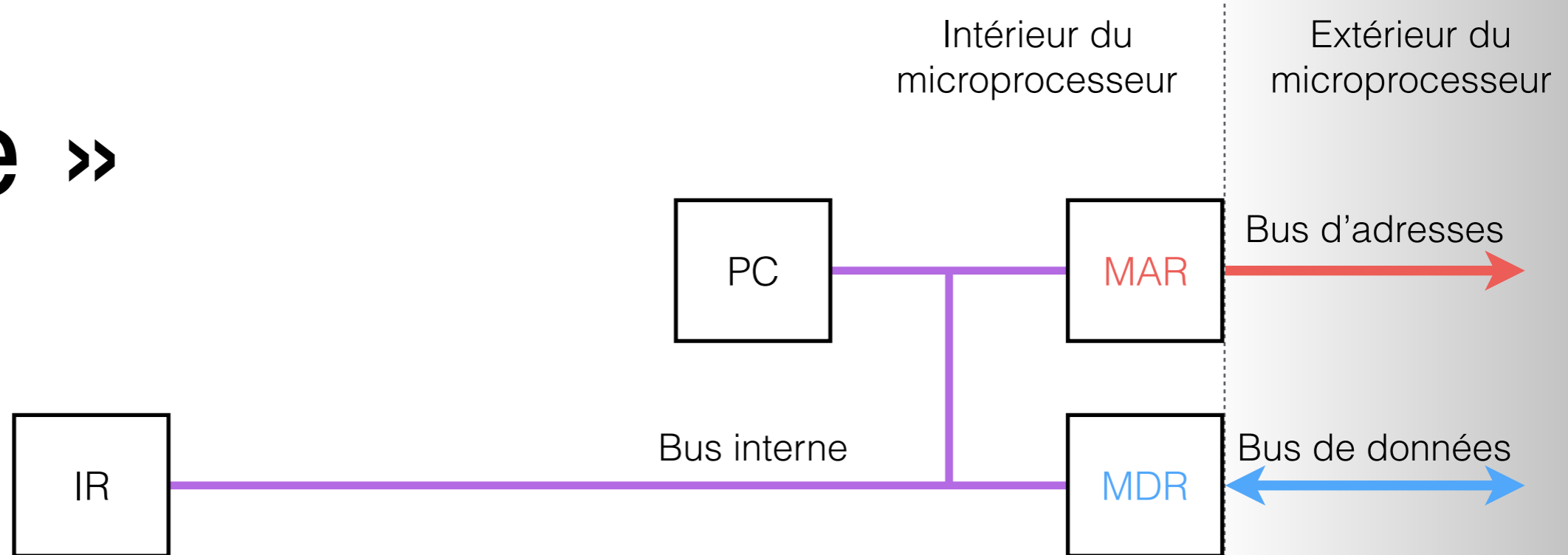
Structure interne



“Fetch” en détails...

- Que se passe-t-il vraiment quand le micro-processeur veut lire la prochaine instruction?
- Où est l'instruction?
 - En mémoire, à l'adresse contenue dans registre PC (“Program Counter”)
- But du « fetch »
 - Placer l'instruction dans le registre IR (« Instruction Register ») pour le décodage

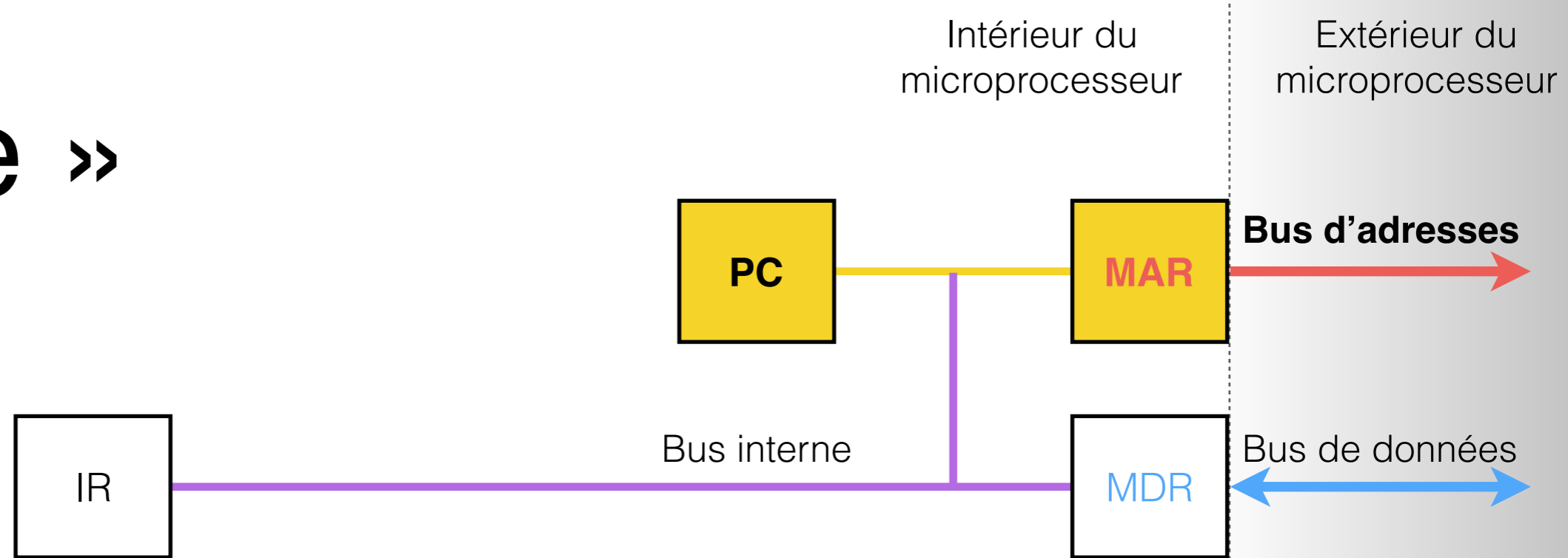
« Lire »



Action	Signification	μ-instructions
Lecture	IR ← instruction	?

Bus de contrôle (L/E)

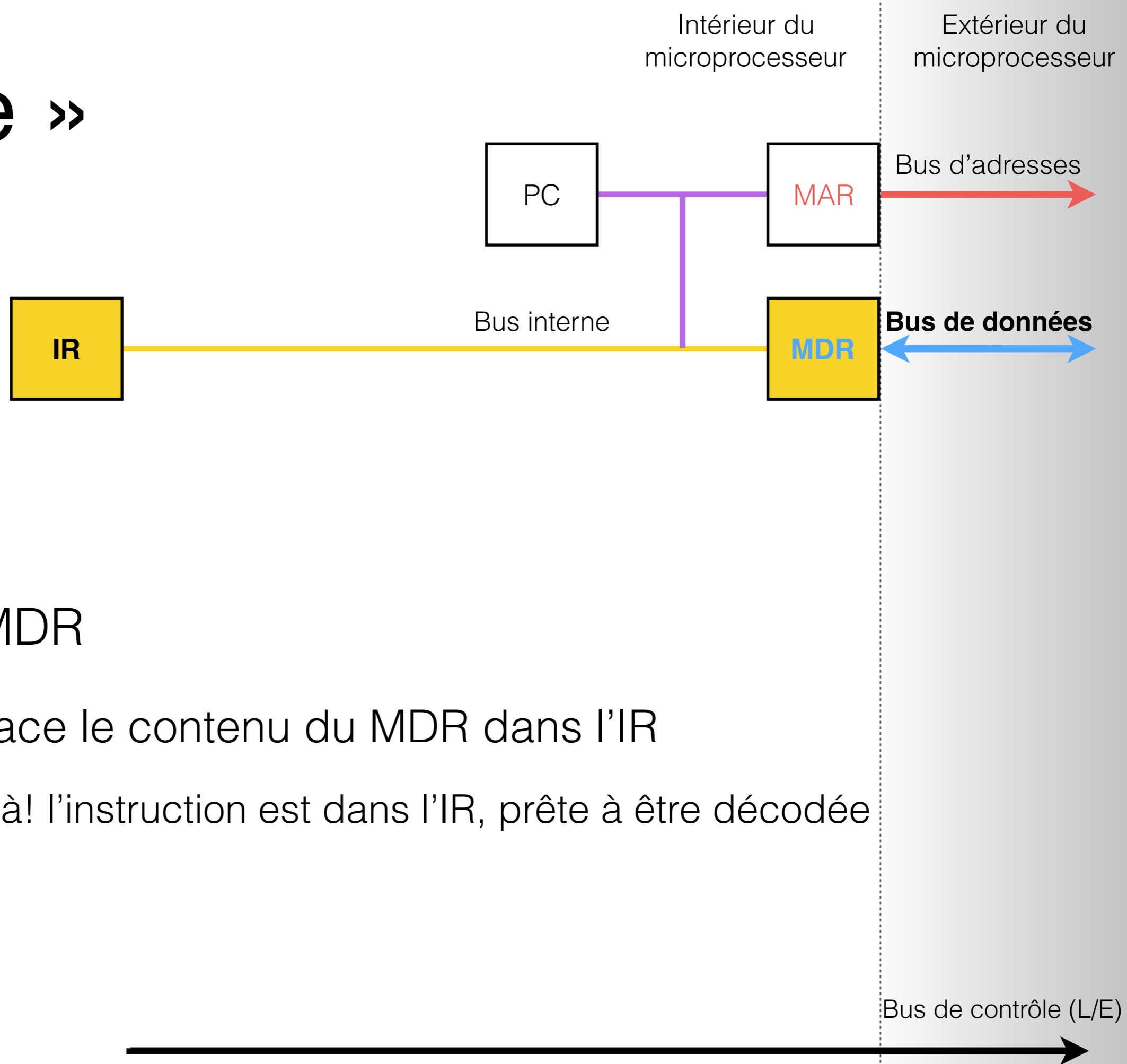
« Lire »



- MAR ← PC
- On place le contenu du PC dans le MAR
 - PC contient l'adresse de la prochaine exécution à exécuter.
 - On active le bus de contrôle en lecture
 - Après cette opération, l'instruction comme telle est disponible... dans le MDR

Bus de contrôle (L/E)

« Lire »



- IR ← MDR
- On place le contenu du MDR dans l'IR
 - Voilà! l'instruction est dans l'IR, prête à être décodée

« Lire » en détails...

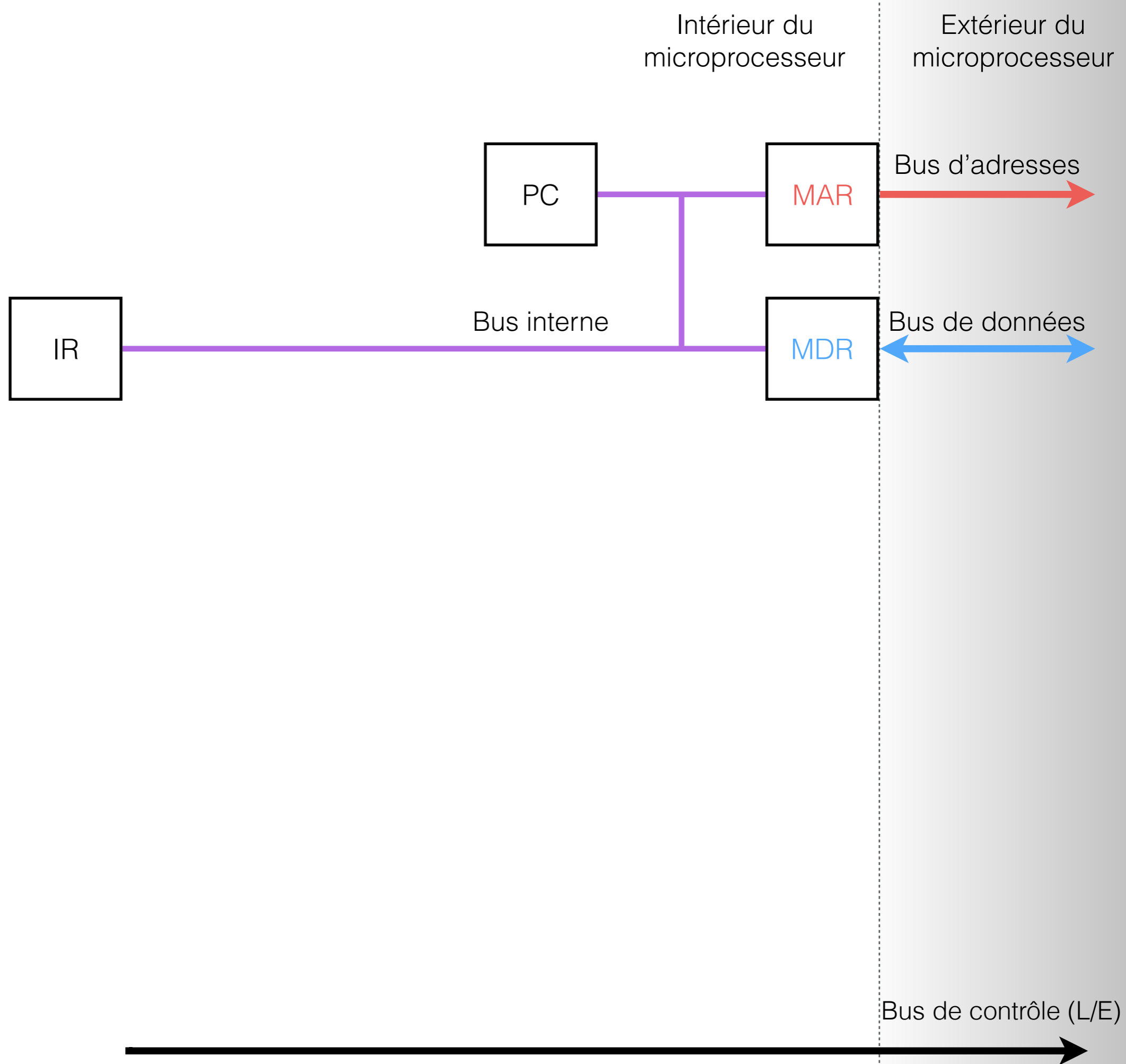
- $MAR \leftarrow PC$
 - On place le contenu du PC dans le MAR
 - PC contient l'adresse de la prochaine exécution à exécuter.

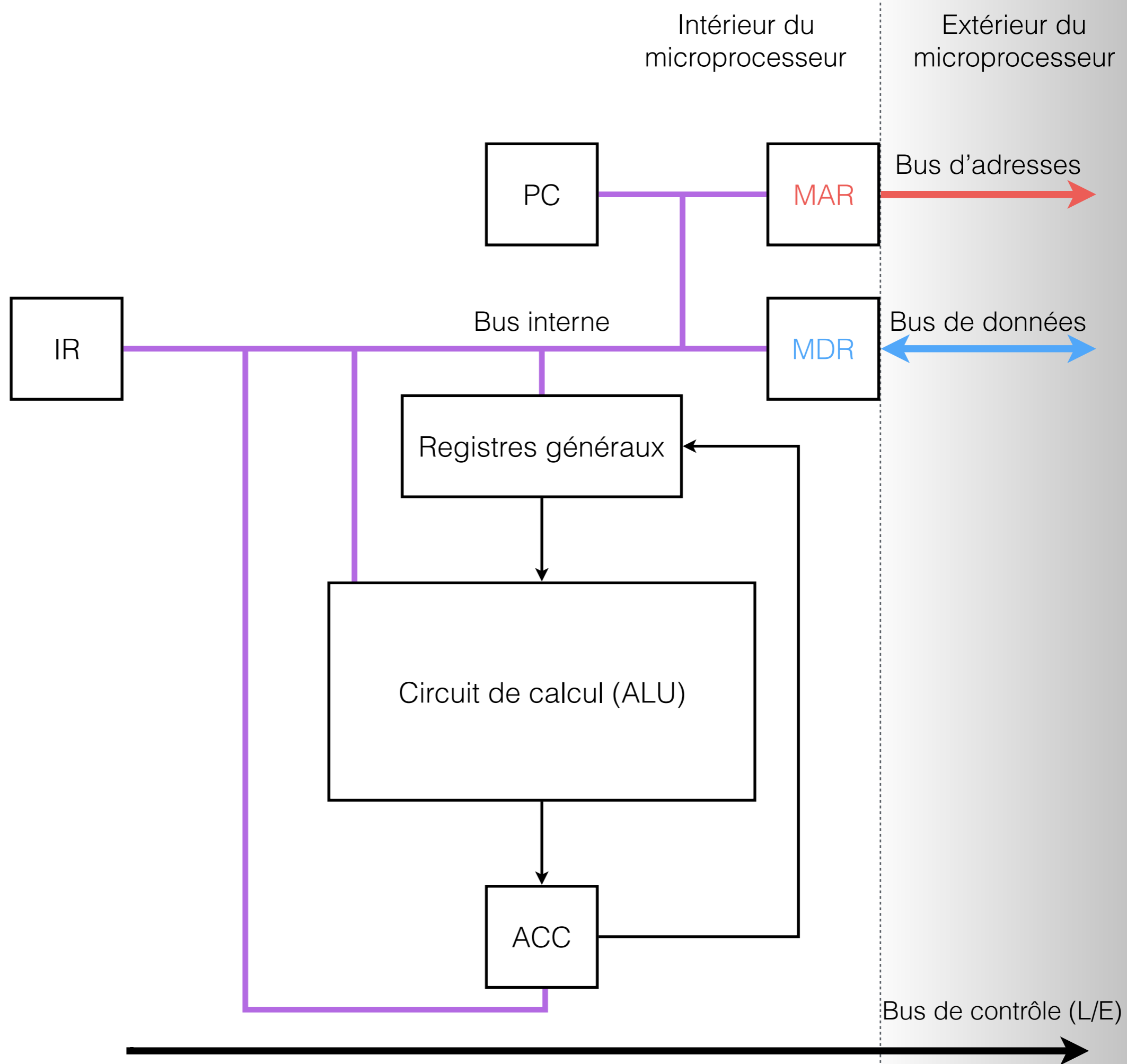
Action	Signification	μ -instructions
Lecture	$IR \leftarrow instruction$	$MAR \leftarrow PC$ $IR \leftarrow MDR$

- $IR \leftarrow MDR$
 - On place le contenu du MDR dans l'IR
 - Voilà! l'instruction est dans l'IR, prête à être décodée et ensuite exécutée

« Exécuter » en détails

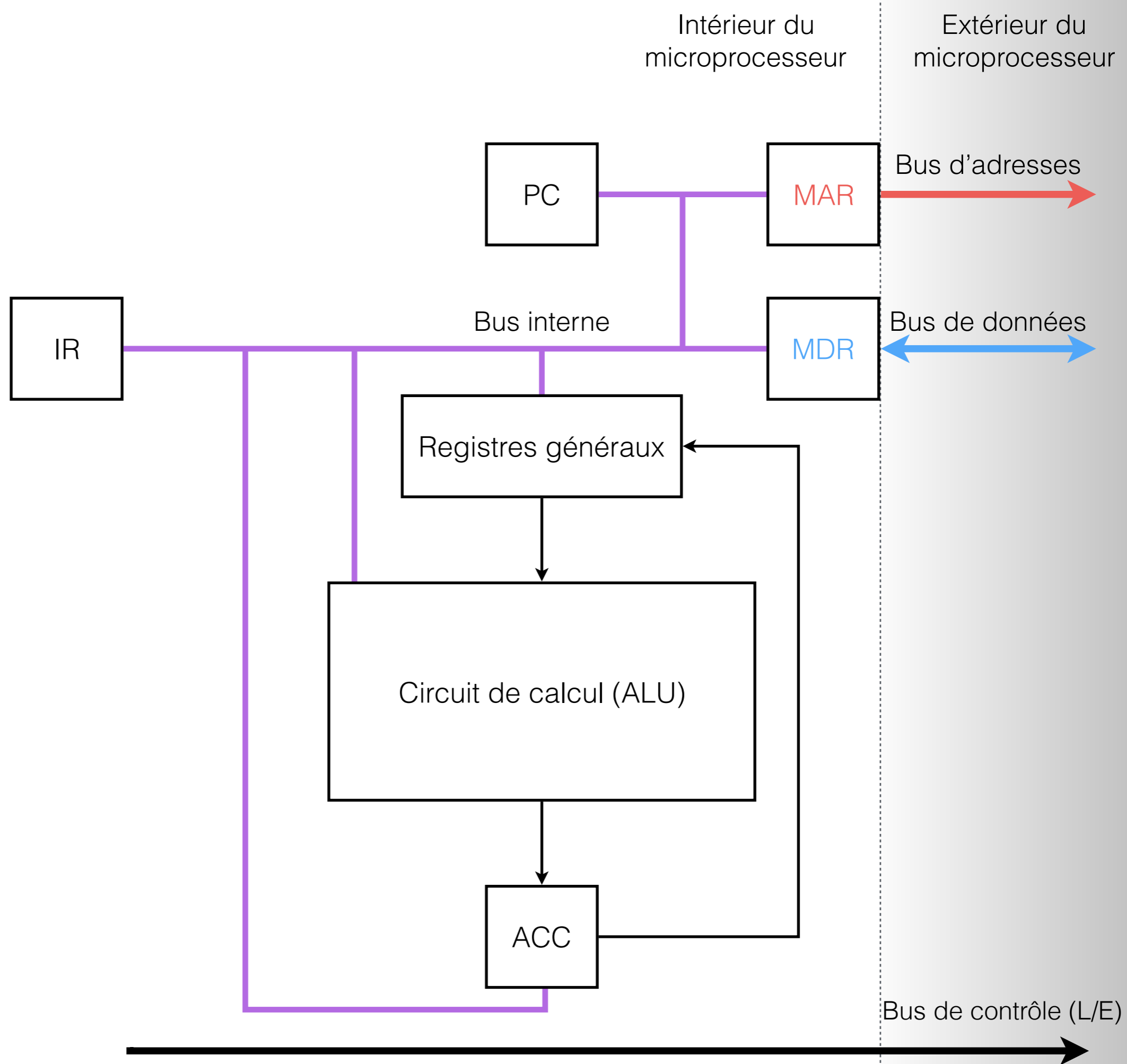
- Admettons un moment que l'instruction dans IR ait été décodée. Comment faire pour exécuter l'instruction?
- Il nous faut connecter un circuit de calcul sur le bus interne

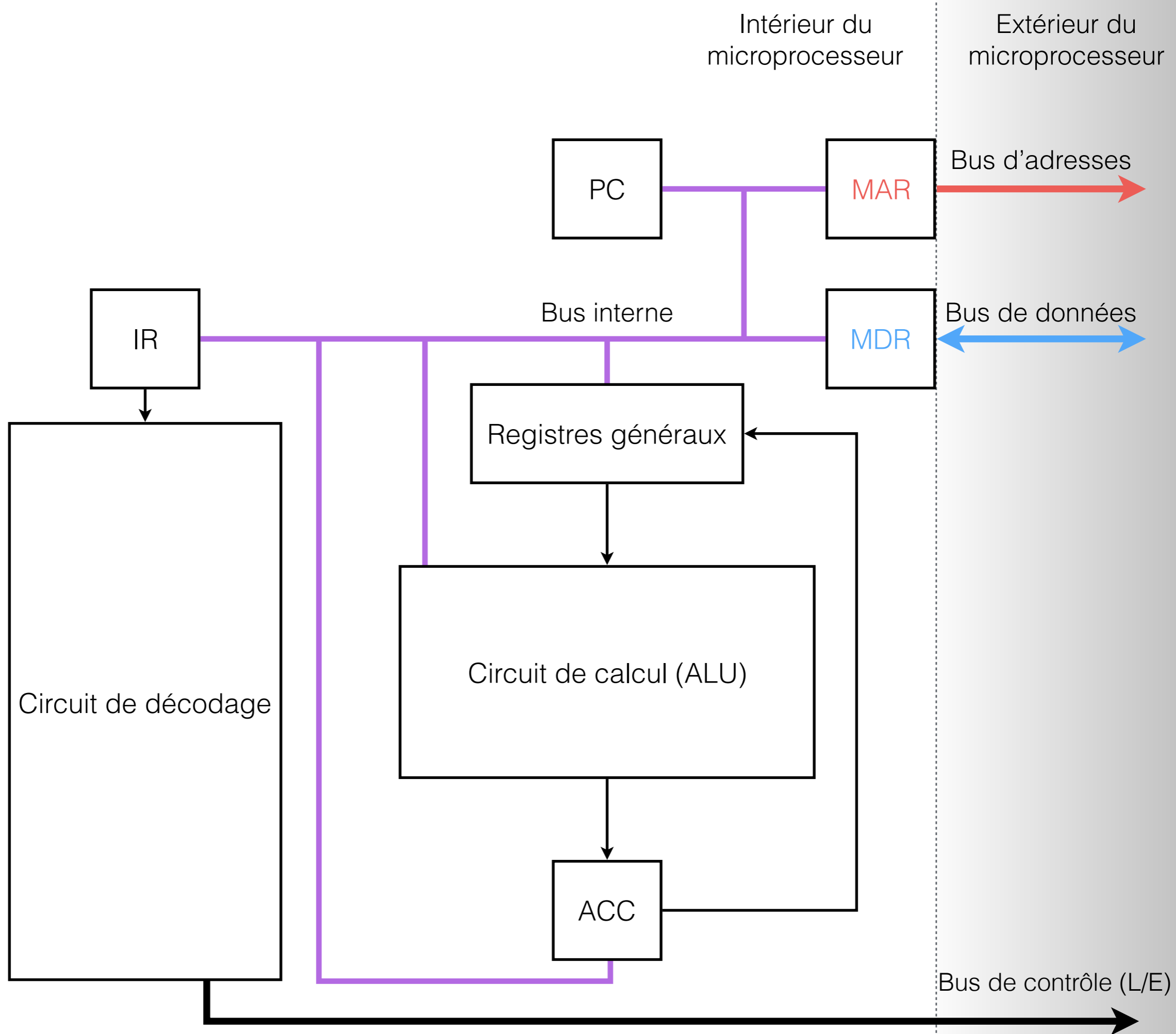


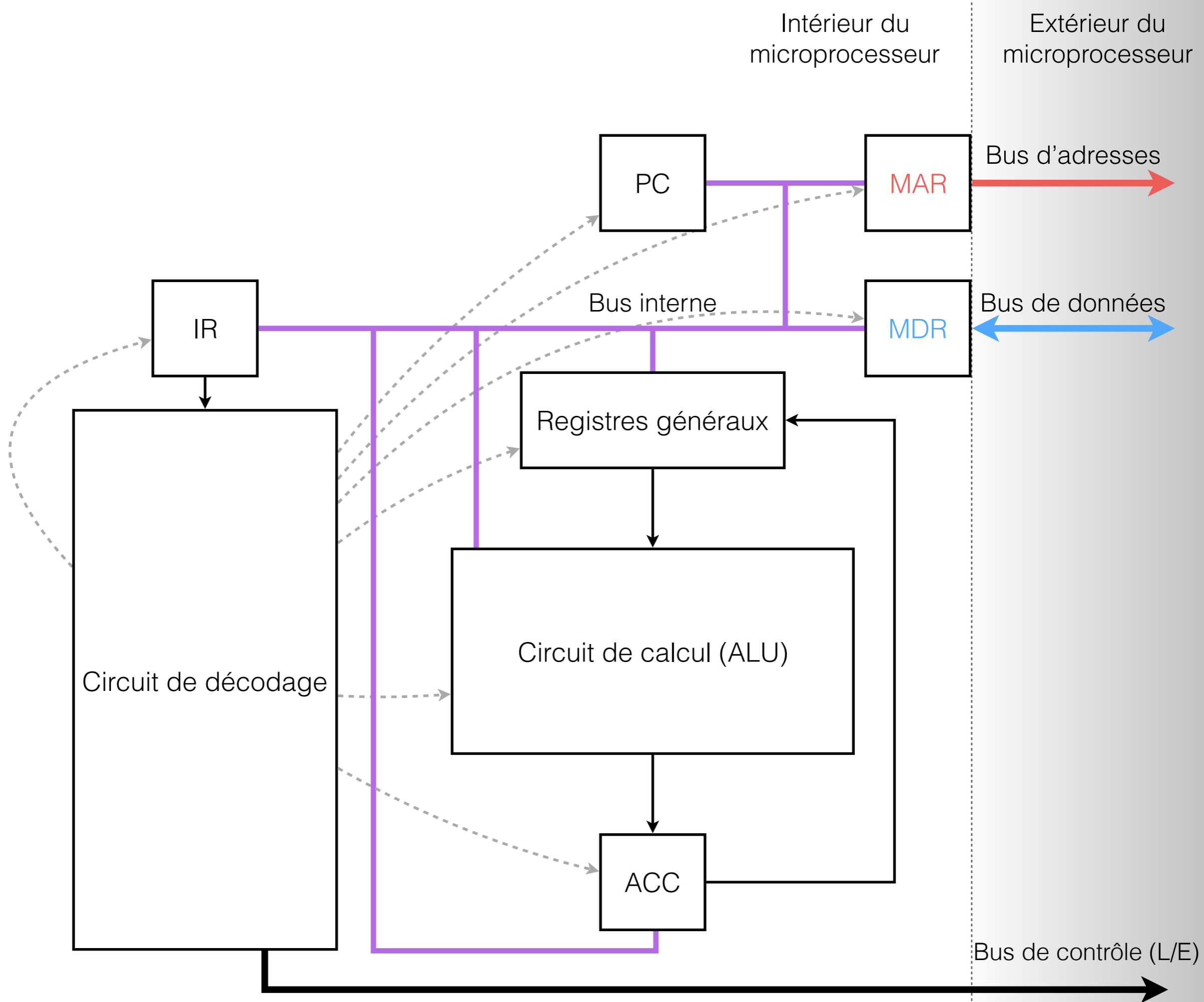


« Décoder » en détails

- Comment faire pour savoir quelles opérations effectuer pour exécuter une instruction?
- Il nous faut un circuit de décodage sur le bus interne







MOV R0 #0x71

Instruction	Signification	μ -instructions
MOV R0 #0x71	R0 \leftarrow 0x71	?

MOV R0 #0x71

- Tout d'abord: quelle est la représentation binaire (sur 16 bits, comme dans le TP1) de cette instruction?

Toutes les instructions du microprocesseur sont sur 16 bits et se décomposent comme suit :

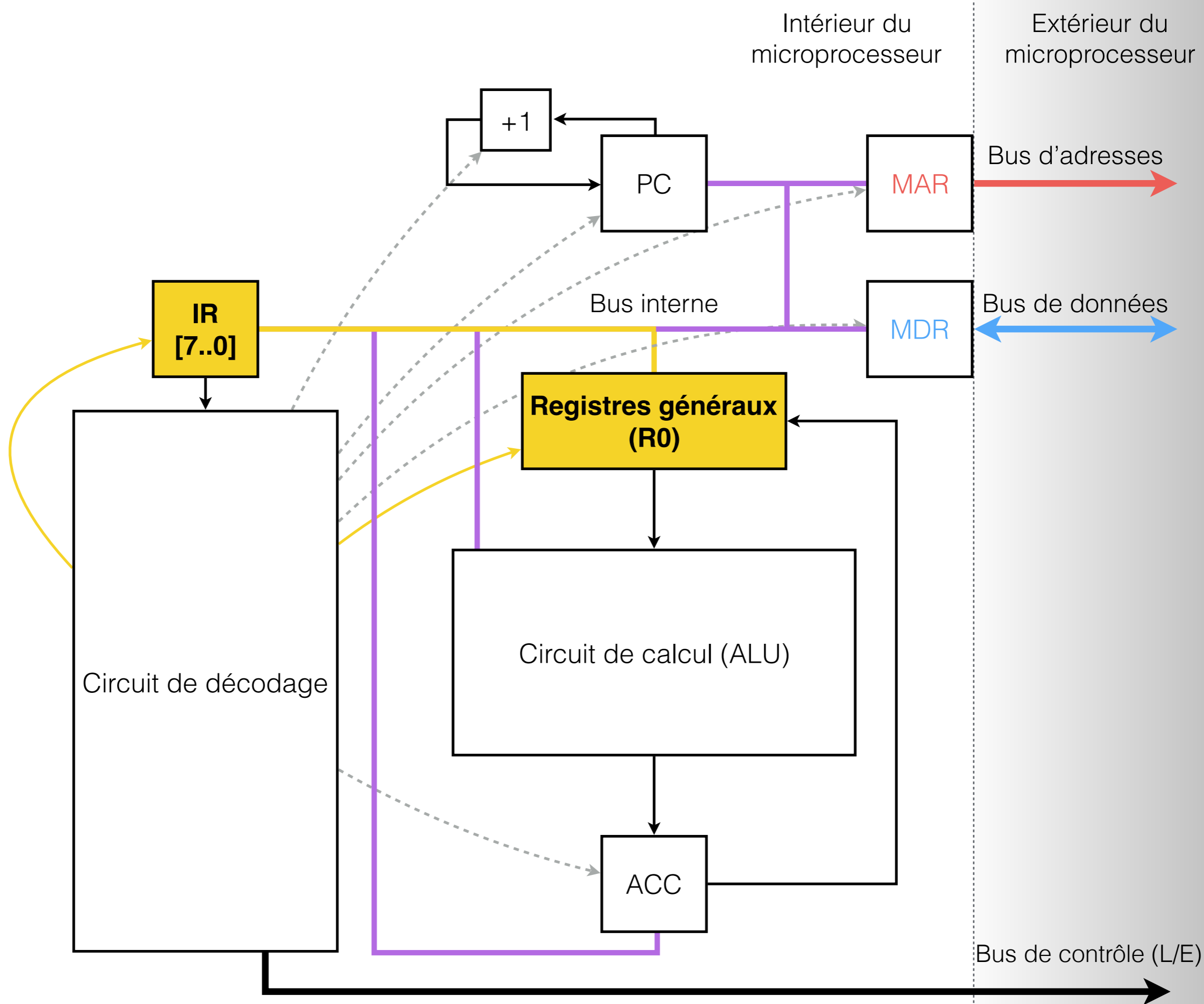
Bits 15 à 12 : Opcode de l'instruction

Bits 11 à 8 : Registre utilisé comme premier paramètre.

Bits 7 à 0 : Registre ou constante utilisés comme deuxième paramètre

Mnémonique	Opcode	Description
MOV Rd Rs	0000	Écriture de la valeur du registre Rs dans le registre Rd
MOV Rd Const	0100	Écriture d'une constante dans le registre Rd

Instruction	Signification	Binaire (16 bits)
MOV R0 #0x71	$R0 \leftarrow 0x71$	0x4071



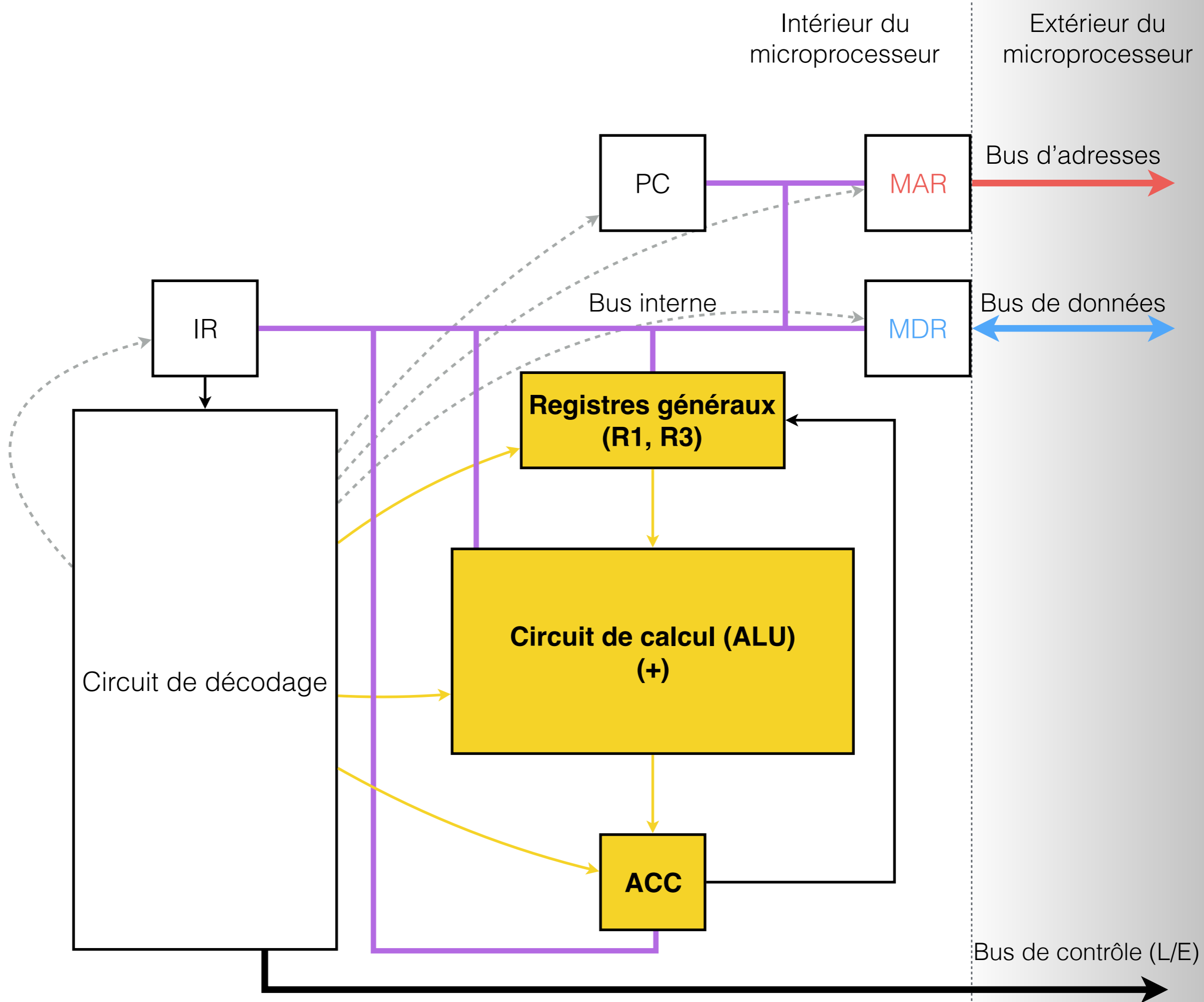
MOV R0 #0x71

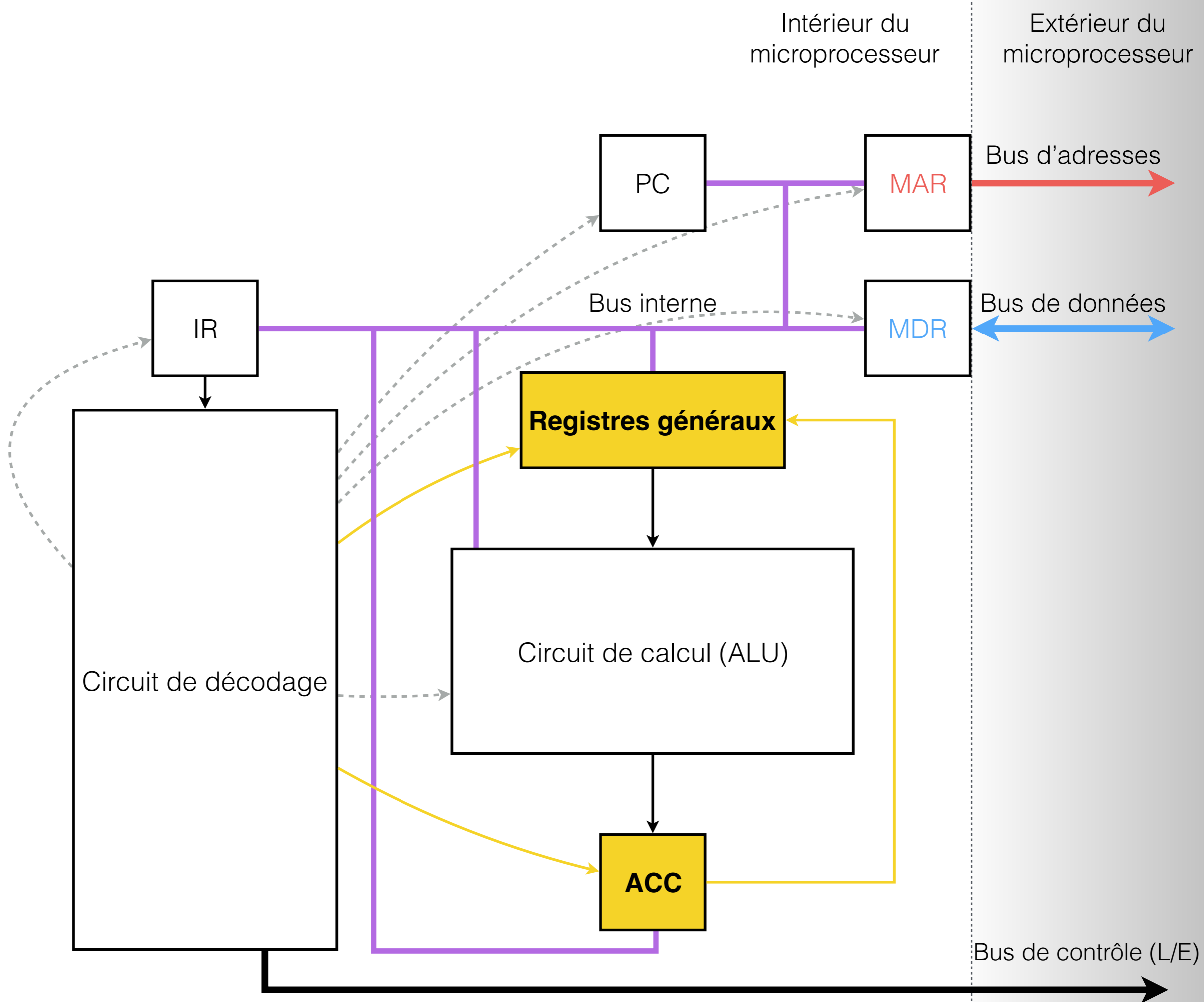
- $R0 \leftarrow IR[7\dots 0]$
 - On place les 8 bits les moins significatifs dans R0 via le bus interne

Instruction	Signification	μ -instructions
MOV R0 #0x71	$R0 \leftarrow 0x71$	$R0 \leftarrow IR[7\dots 0]$

ADD R1 R3

Instruction	Signification	μ -instructions
ADD R1 R3	$R1 \leftarrow R1 + R3$?





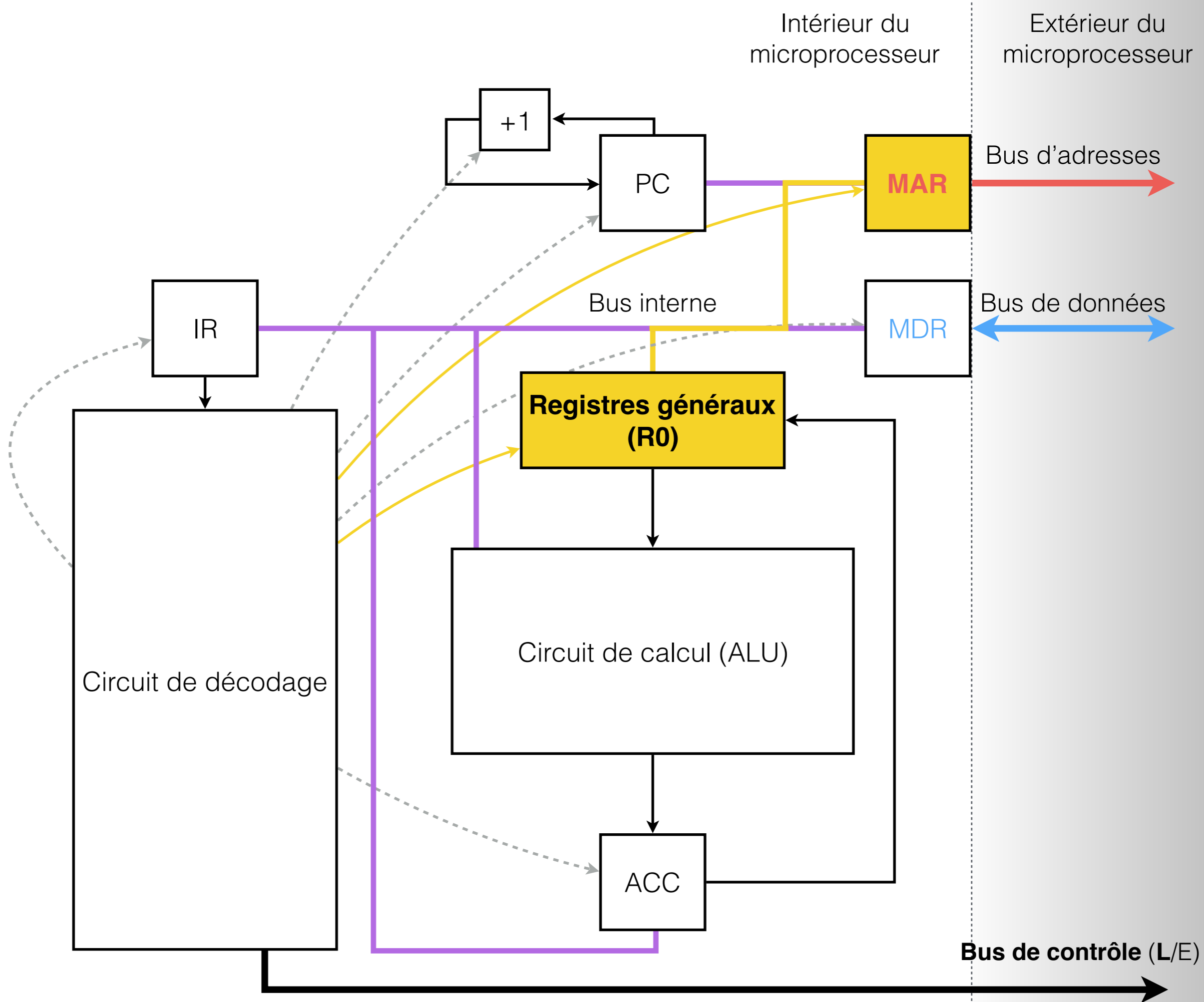
ADD R1 R3

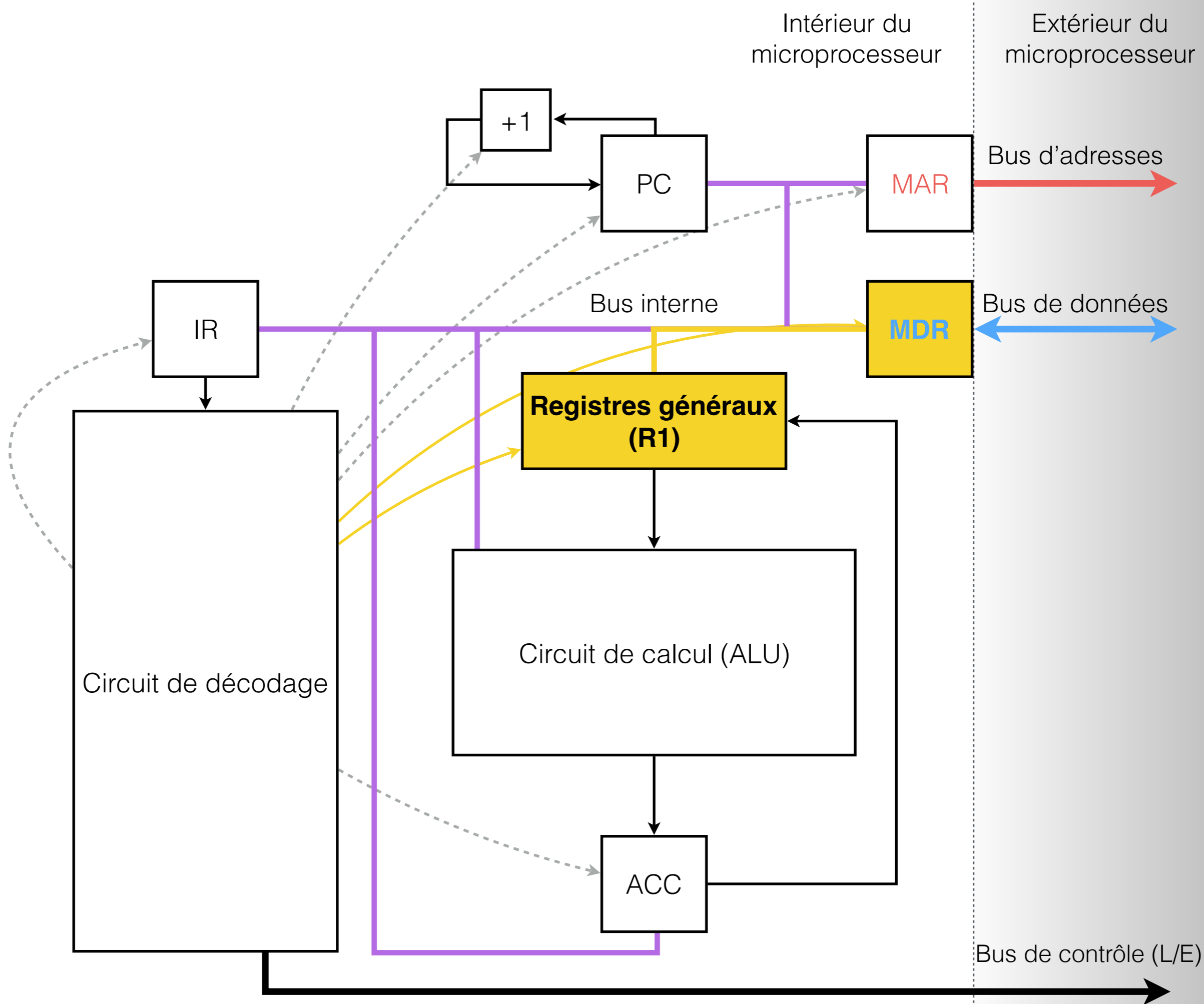
- $ACC \leftarrow R1 + R3$
 - On fait l'addition de R1 et R3, le résultat est placé dans l'accumulateur
- $R1 \leftarrow ACC$
 - On transfère le contenu de l'accumulateur dans le registre R1

Instruction	Signification	μ -instructions
ADD R1 R3	$R1 \leftarrow R1 + R3$	$ACC \leftarrow R1 + R3$ $R1 \leftarrow ACC$

LDR R1 [R0]

Instruction	Signification	μ -instructions
LDR R1 [R0]	$R1 \leftarrow \text{Memoire}[R0]$?





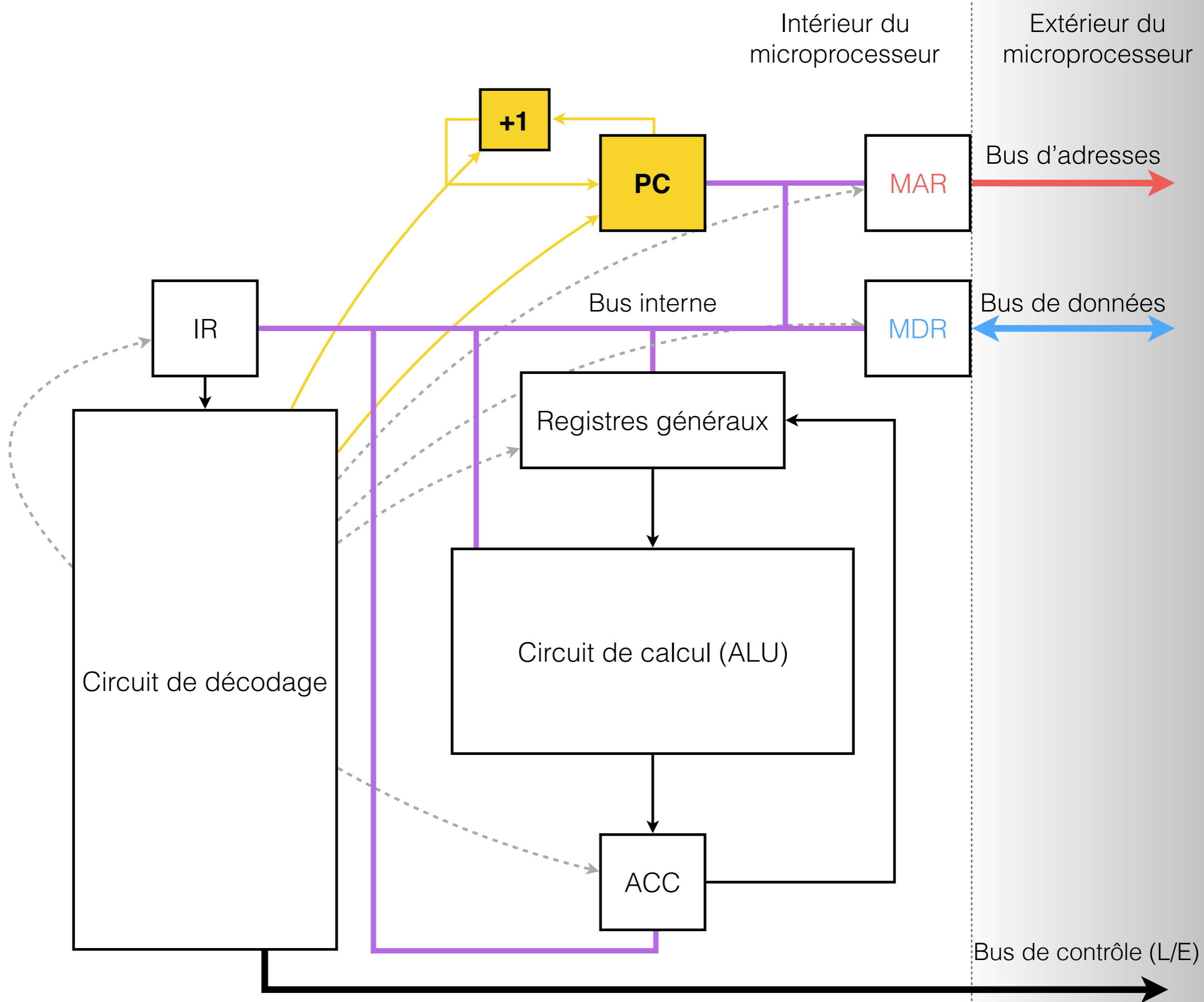
LDR R1 [R0]

- $MAR \leftarrow R0$
 - Le contenu de R0 est placé sur le bus d'adresse via le MAR
 - Le bus de contrôle est activé en lecture
- $R1 \leftarrow MDR$
 - Le contenu du bus de données, accessible via le MDR, est placé dans R1.

Instruction	Signification	μ -instructions
LDR R1 [R0]	$R1 \leftarrow \text{Memoire}[R0]$	$MAR \leftarrow R0$ $R1 \leftarrow MDR$

Après chaque lecture...

Action	μ -instructions
Incrémenter PC	$PC \leftarrow PC + 1$



Résumé (lecture & exécution)

Instruction	Étapes	μ-instructions
ADD R1 R3	lecture	MAR ← PC IR ← MDR
	incrémente PC	PC ← PC+1
	exécution	ACC ← R1 + R3 R1 ← ACC
MOV R0 #0x71	lecture	MAR ← PC IR ← MDR
	incrémente PC	PC ← PC+1
	exécution	R0 ← IR[7..0]

Programmes, instructions, et micro-instructions

