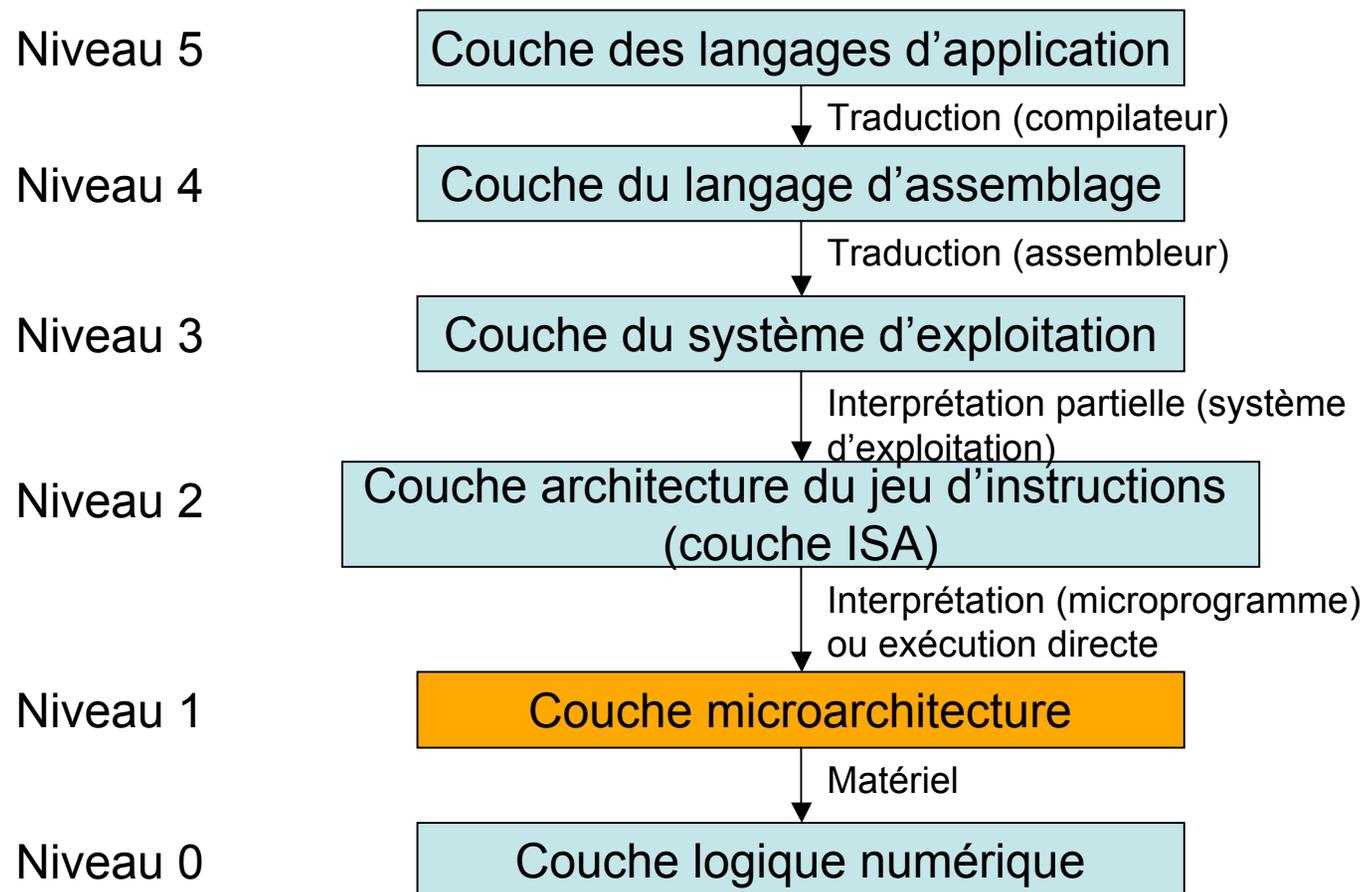

La fonction d'exécution

CPU, Mémoire

La couche « microarchitecture »

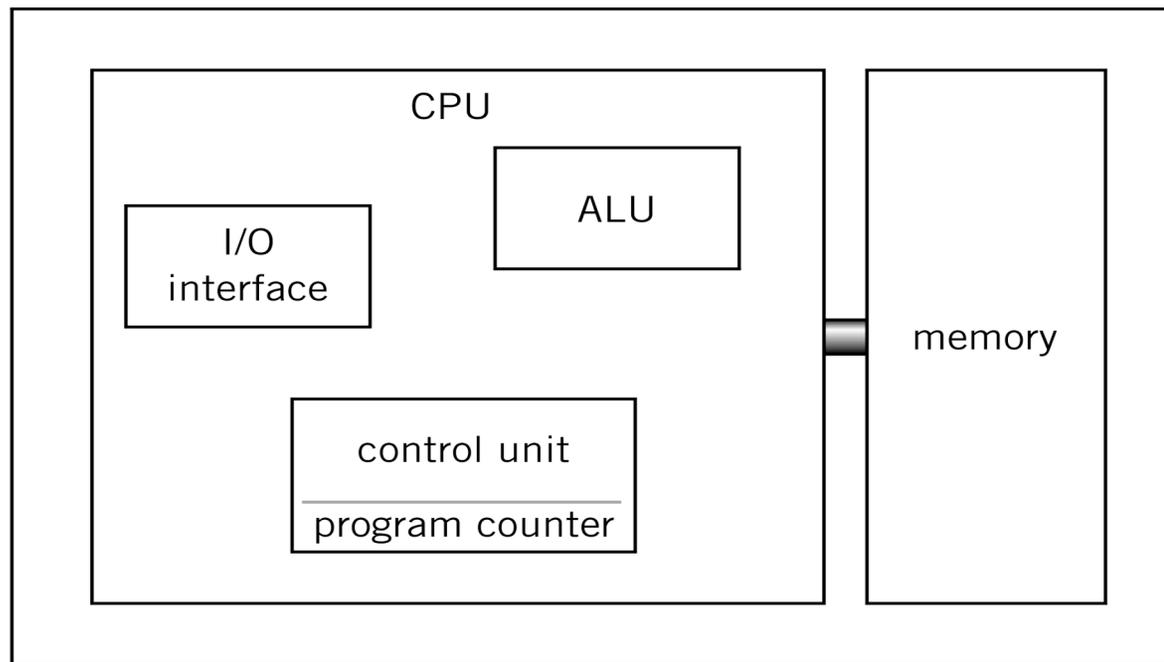


La couche « microarchitecture »

- Des fonctions logiques évoluées sont interconnectées :
 - Registres
 - Contrôleur
 - Unité Arithmétique et logique
 - ...
- Couche supérieures : ISA (jeu d'instructions)
- Couche inférieure : Logique numérique

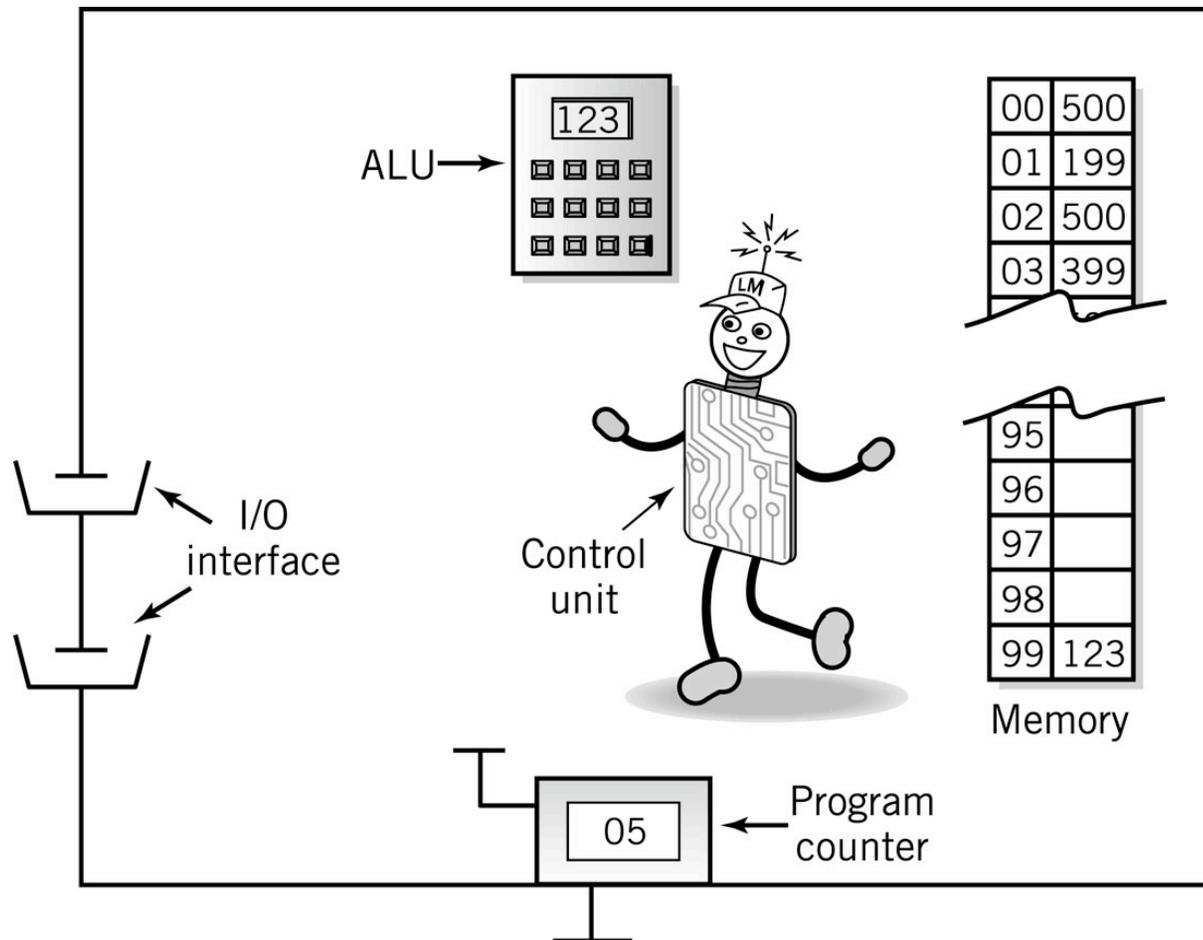
CPU: Structure

- Dans les ordinateurs réels, le jeu d'instructions est encodé en binaire et la logique câblée remplace le LM



- CU (Control Unit) contrôle, interprète les instructions, lit le compteur d'instructions et fait la séquence d'actions correspondantes au cycle *Fetch/Execute*
- ALU + CU = CPU (Central Processing Unit)

Little Man Computer



CPU & Registres

- Exécute les instructions machines placées en mémoire centrale
 - Est constitué de quatre parties
 - ▣ L'unité arithmétique et logique (*UAL*)
 - ▣ Exécution de tous les calculs de microprocesseur
 - ▣ *Les registres*
 - ▣ Zones de mémorisation de l'information internes au microprocesseur
 - ▣ L'unité de commande (*CU* – control unit)
 - ▣ Exécute les instructions machines en utilisant les registres et l'UAL
 - ▣ Le bus de communication interne

Registres LMC

- Zones de mémorisation de l'information internes au microprocesseur
 - Registres spécifiques:
 - 2 registres pour la manipulation des instructions
 - PC (le compteur ordinal)
 - IR (le registre d'instruction)
 - 2 registres permettant la communication avec les autres modules via le bus
 - MAR (le registre d'adresses)
 - MDR (le registre de données)
 - Registres généraux
 - Un seul registre : Acc (accumulateur)

Compteur de programme (PC)

C'est un registre d'adresses.

Contient l'adresse de la prochaine instruction à exécuter.

Incrémenté après l'exécution de chaque instruction.

Peut être changé lors de l'exécution: instruction "jump".

Initialiser à zéro ou « reset » au début.

Le registre d'instruction (IR)

C'est un registre de données. Il contient une instruction à exécuter.

Op Code + Adresse

Type de traitement à réaliser

Localisation des données

ADD 99:

1 99

Registre d'adresses (MAR)

Contient l'adresse du mot mémoire. Cette adresse est placée sur le bus d'adresses et devient la valeur d'entrée du circuit de sélection qui va à partir de cette entrée sélectionner le mot correspondant.

Registre de données (MDR)

Il permet l'échange d'informations (contenu d'un mot mémoire) entre la mémoire centrale et le processeur (registre)

Accumulateur

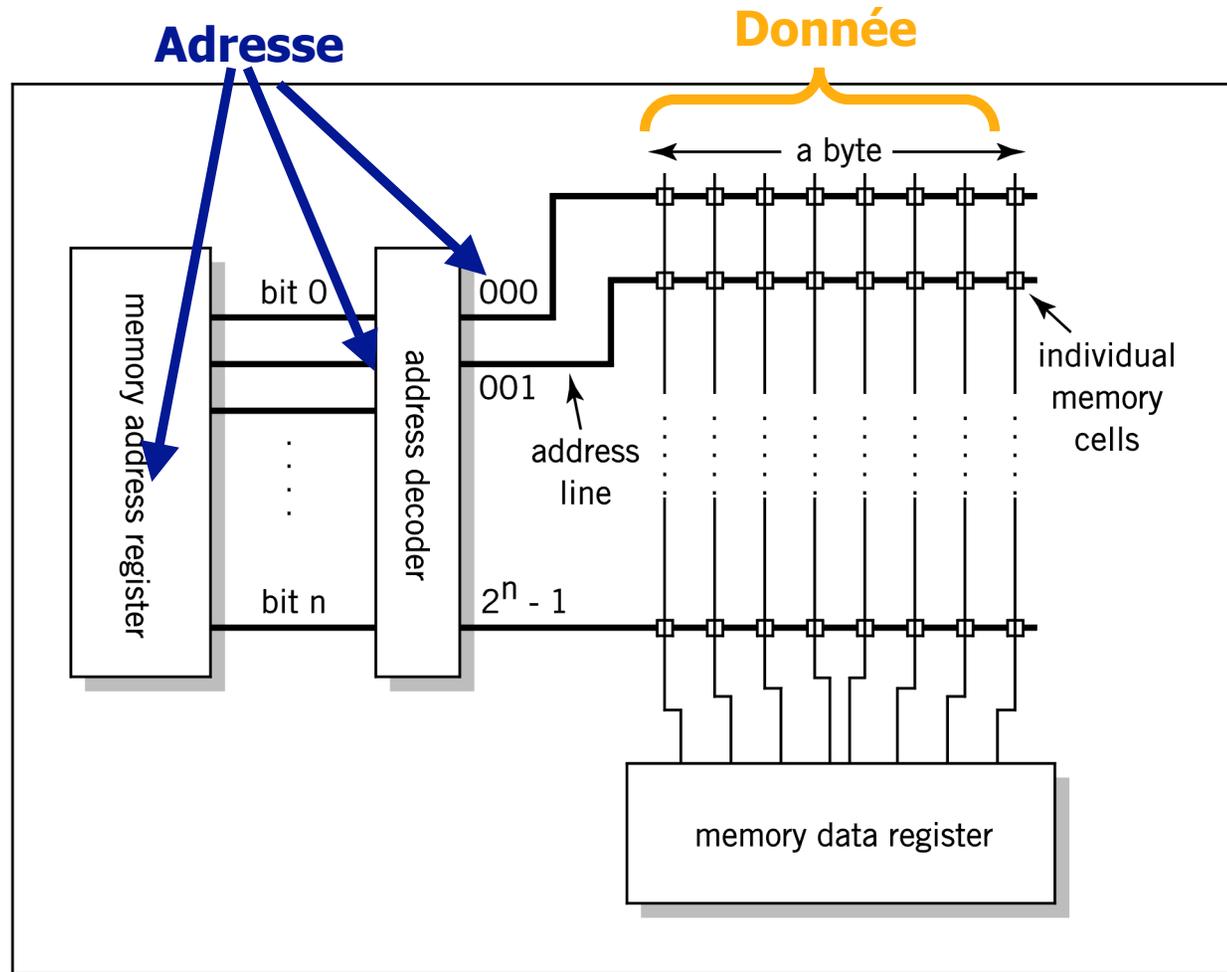
Registre (ou l'ensemble de registres) utilisé pour la manipulation des données.

D'habitude contient les résultats des opérations arithmétiques ou logiques.

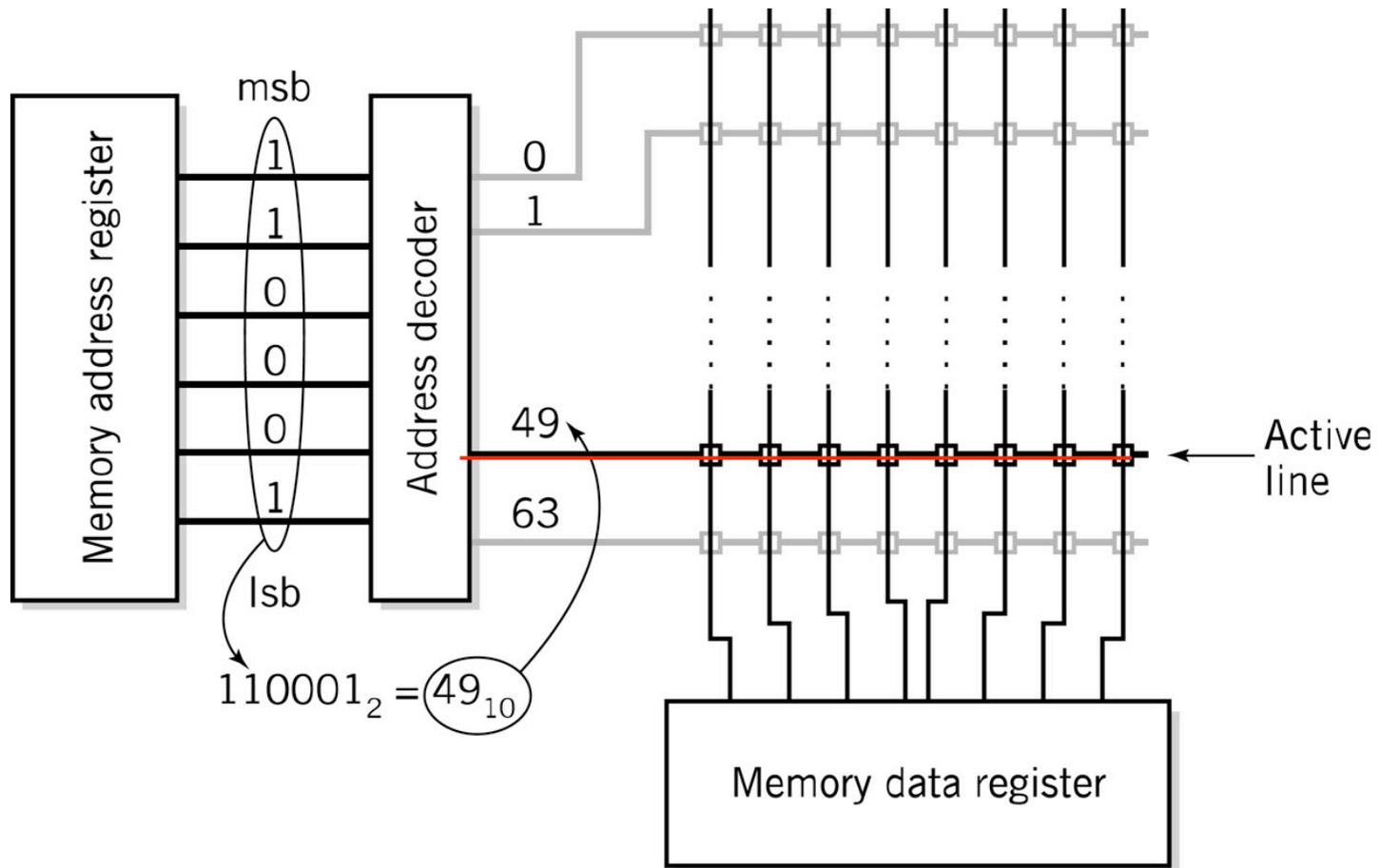
Opération de la mémoire

- Une unité Mémoire consiste en un ensemble de cellules de stockage possédant une adresse propre et pouvant stocker une valeur binaire
- MAR – registre contenant l'adresse de la mémoire où sera stockée la donnée
- MDR – registre contenant la donnée/Instruction à être stockée/lu

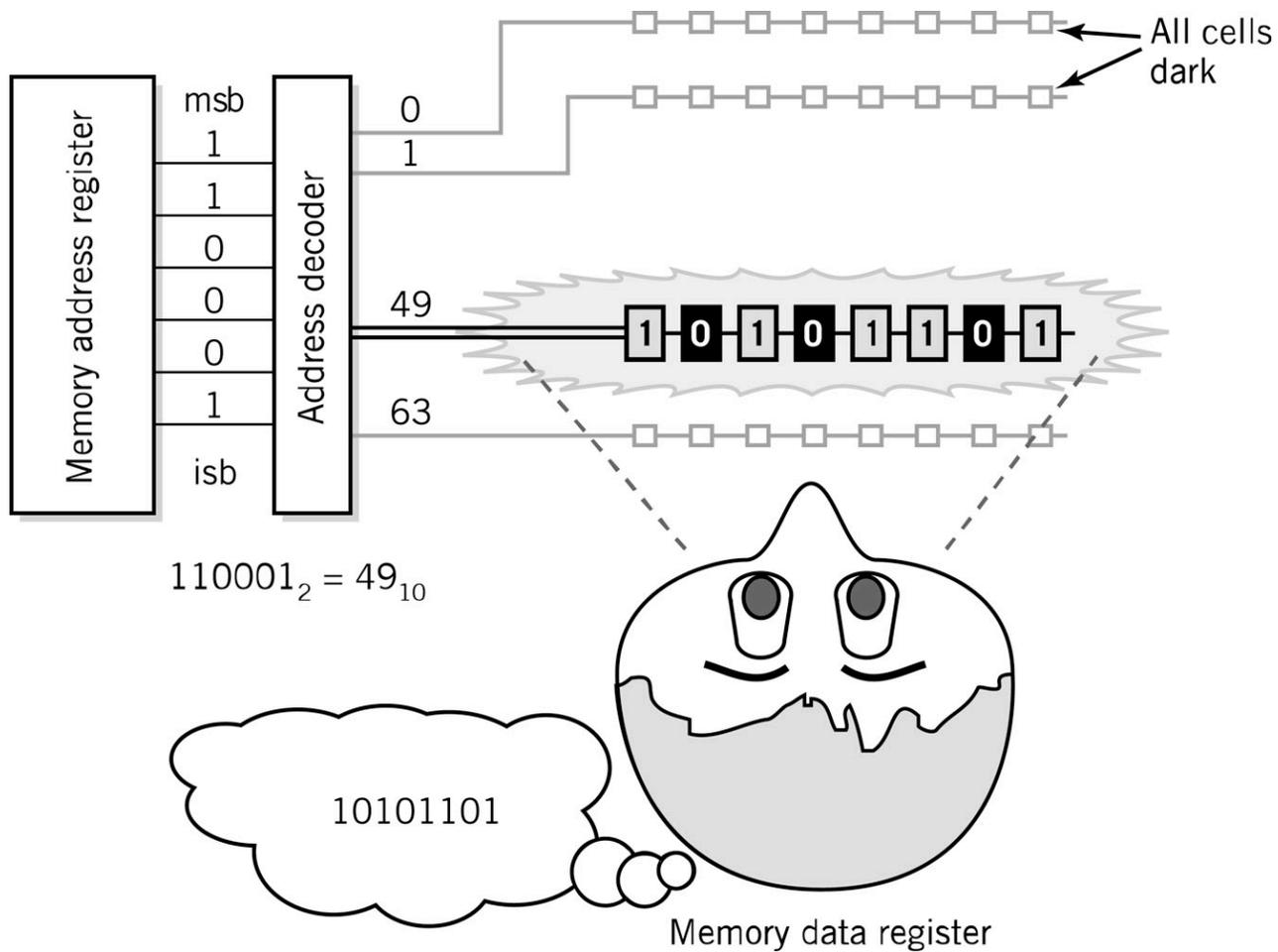
Mémoire et les registres MDR & MAR



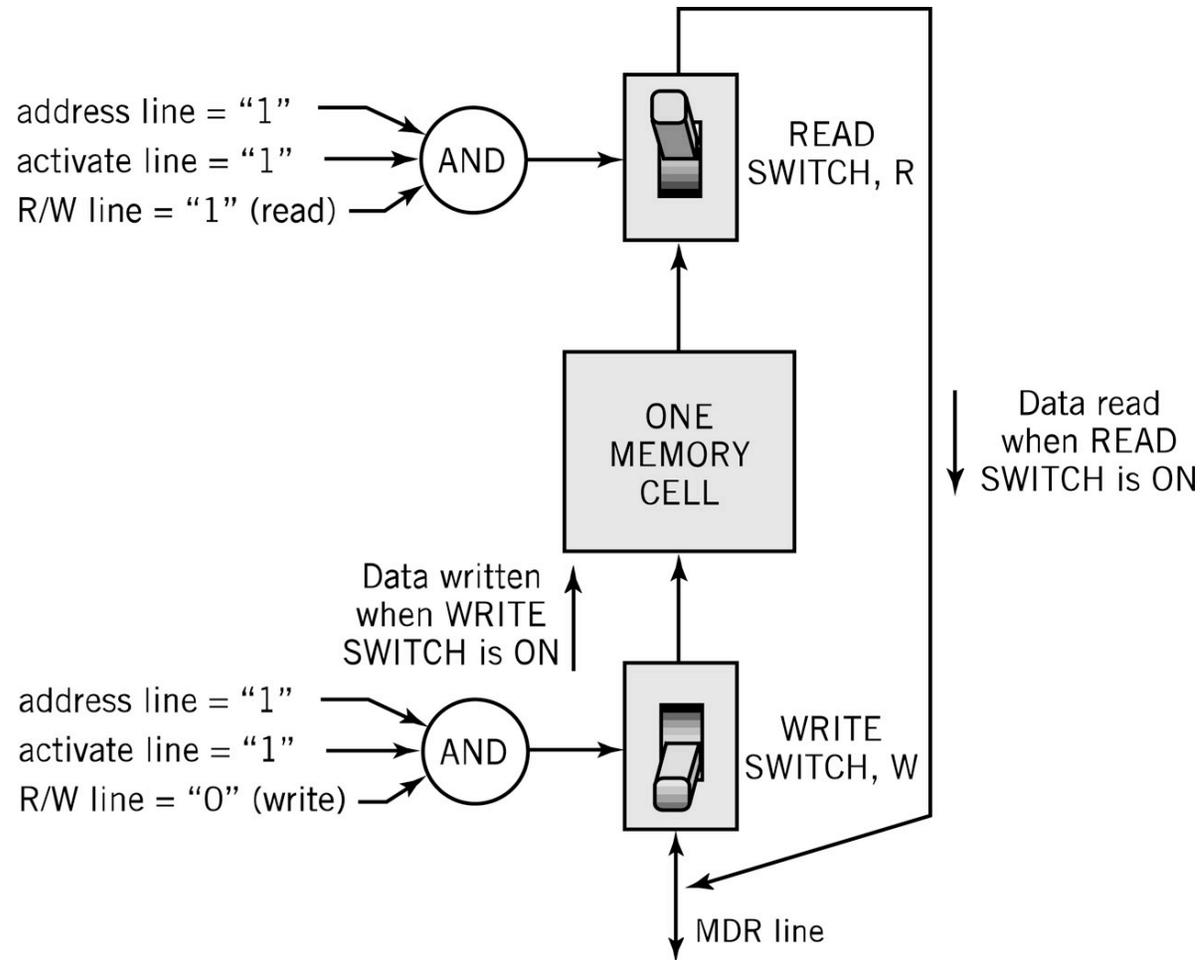
Exemple MAR-MDR



Analogie visuelle



Cellule mémoire



Capacité Mémoire

Déterminée par deux facteurs:

1. Nombre de bits dans MAR

- ▣ LMC = 100 (00 to 99)
- ▣ Nombre d'adresses mémoires pouvant être décodées

2. Taille du champs adresse dans l'instruction

- ▣ 4 bits permet 16 locations
- ▣ 8 bits permet 256 locations
- ▣ 32 bits permet 4,294,967,296 ou 4 GB

L'exécution d'une instruction

- L'exécution d'une instruction implique:
 - Le microprocesseur
 - Registres
 - UAL
 - Unité de commande
 - Bus interne
 - Le bus de communication mémoire/microprocesseur
 - La mémoire centrale

L'exécution d'une instruction

- Exécuter une instruction équivaut à permettre des interactions efficaces entre ces trois composants
 - Microprocesseur
 - Le bus de communication mémoire/microprocesseur
 - La mémoire centrale

Registres CPU LMC

Compteur de programme (PC)

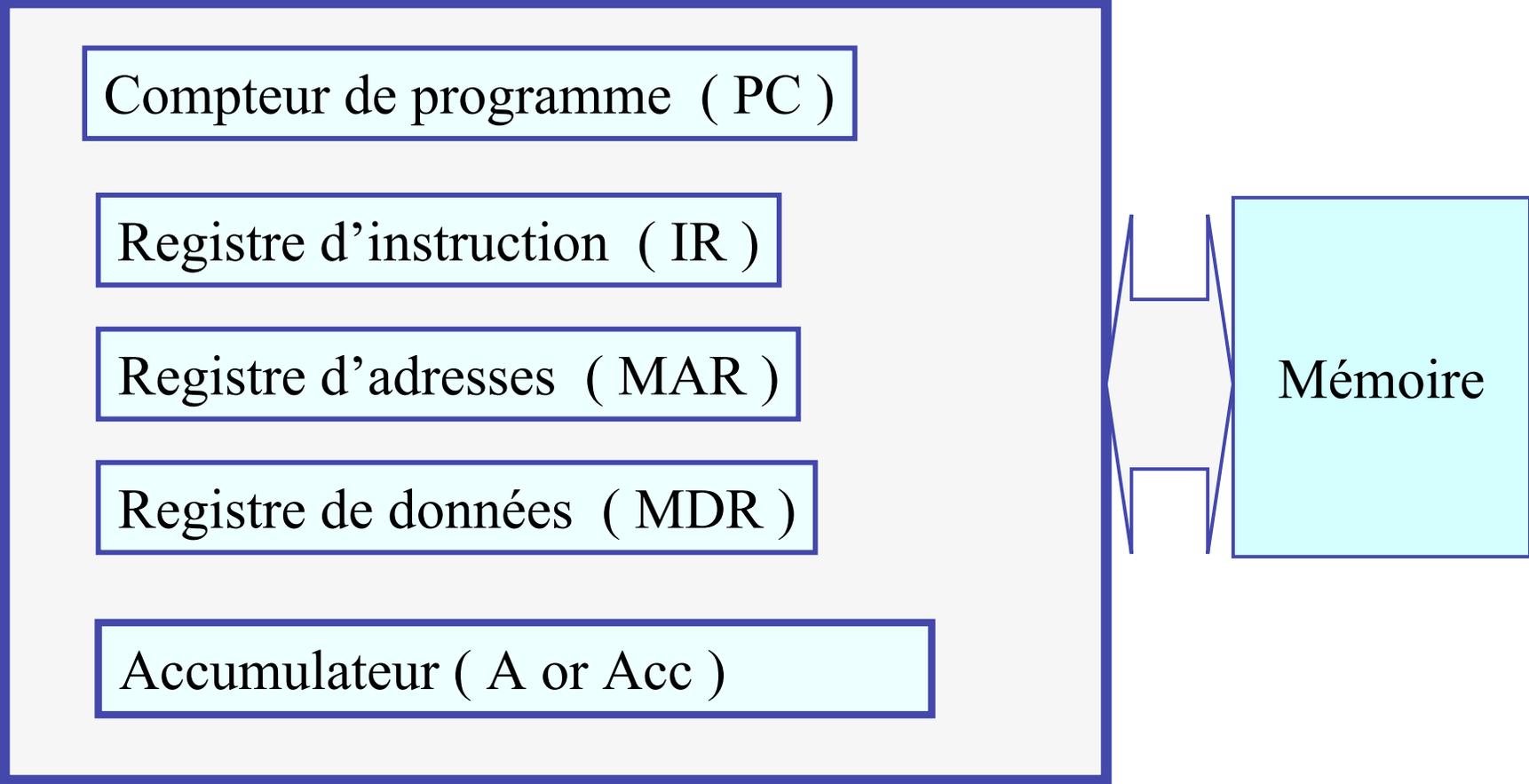
Registre d'instruction (IR)

Registre d'adresses (MAR)

Registre de données (MDR)

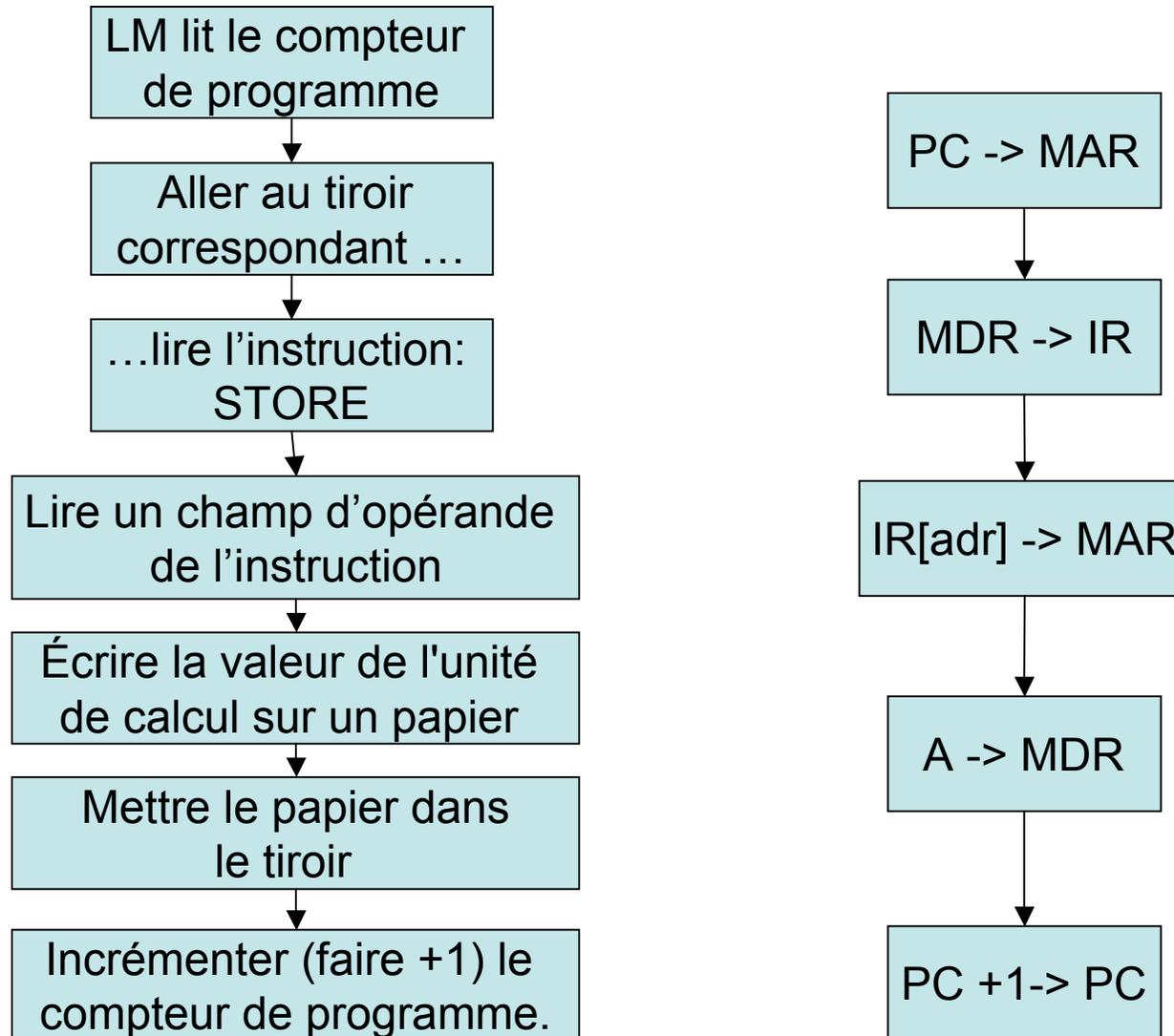
Accumulateur (A or Acc)

Mémoire

The diagram shows a large light blue box on the left containing five smaller boxes, each representing a CPU register: 'Compteur de programme (PC)', 'Registre d'instruction (IR)', 'Registre d'adresses (MAR)', 'Registre de données (MDR)', and 'Accumulateur (A or Acc)'. To the right of this box is a light blue box labeled 'Mémoire'. Two horizontal lines connect the CPU box to the memory box, with a zigzag line on each side, representing a bidirectional data path.

Little Man vs CPU

Cycle Fetch-Execute



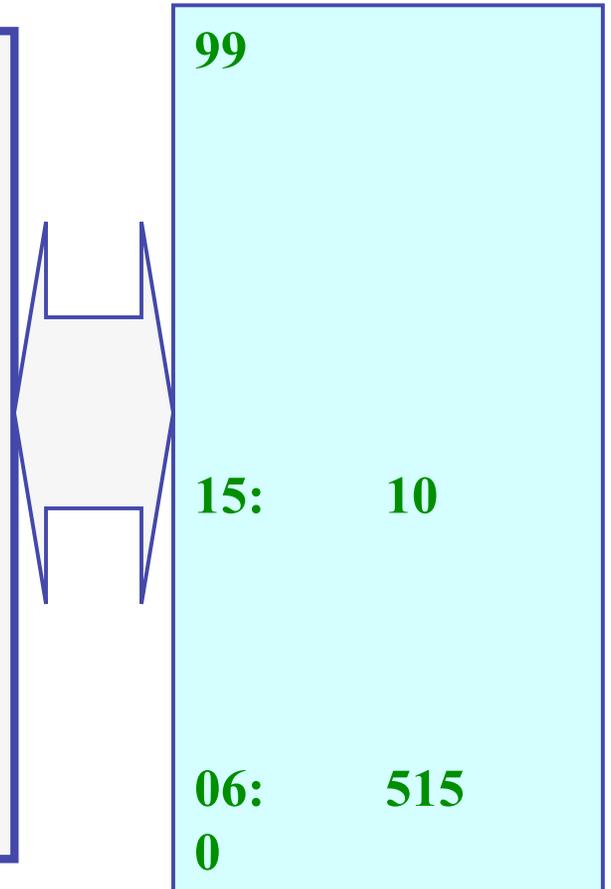
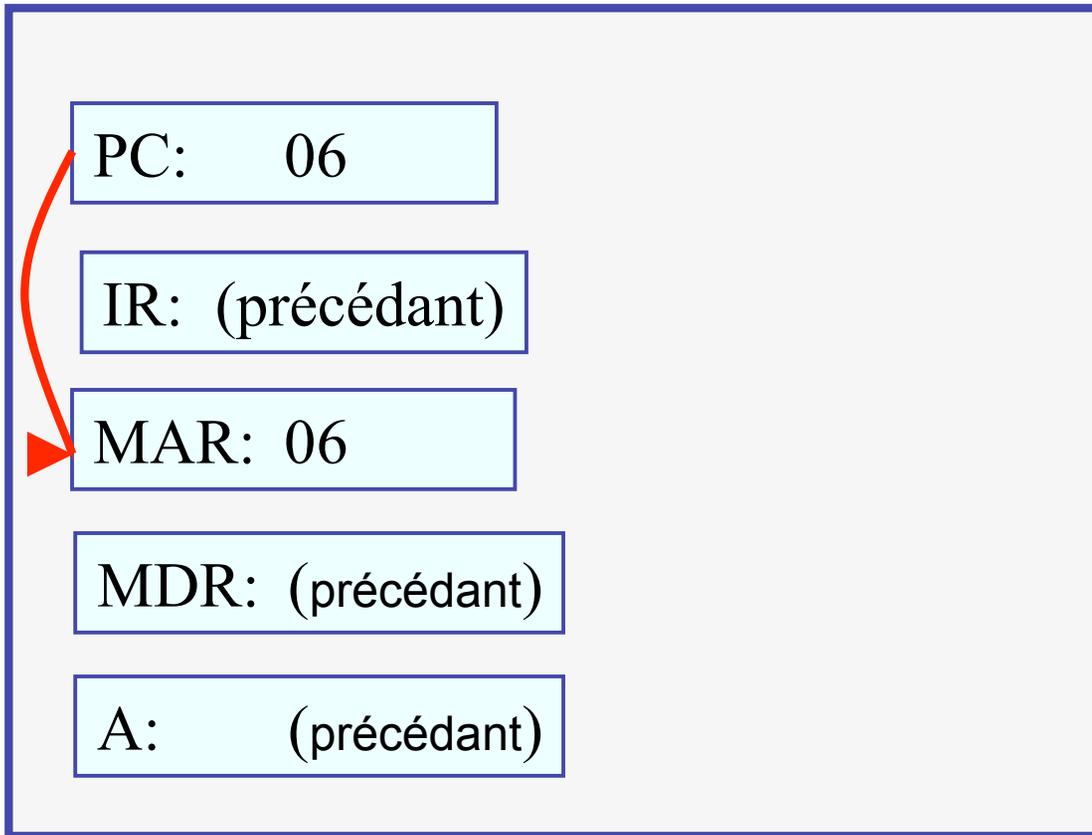
Cycle d'instruction: Fetch-Execute

06	LDA	15	$A \leftarrow M[15]$
		515	

«Fetch»

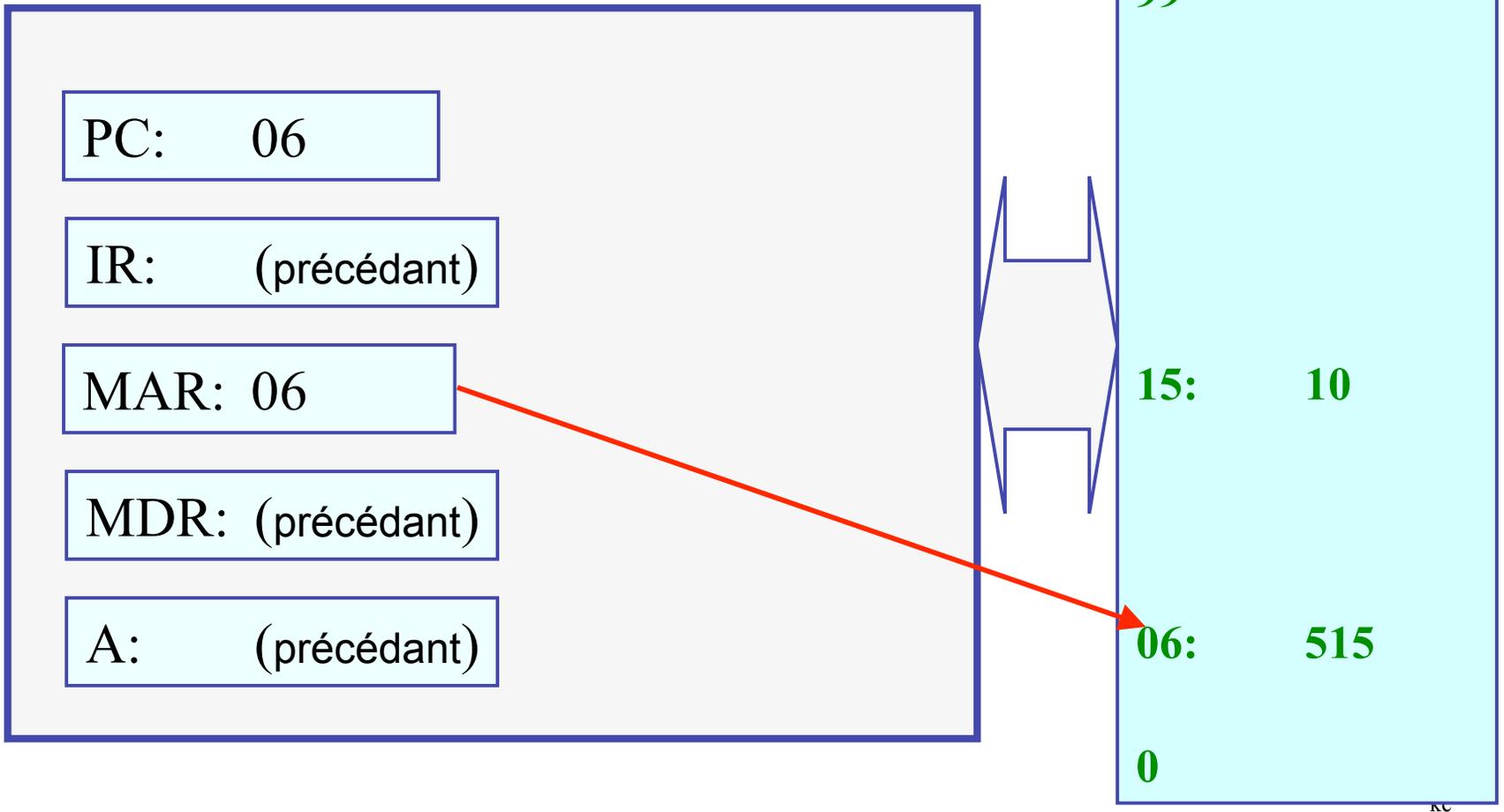
- C'est la phase de recherche et de chargement de l'instruction, pointée par le compteur de programme PC dans le registre instruction IR
 - Le contenu du PC est placé dans le registre d'adresses MAR
 - ▣ L'accès mémoire
 - Placer le contenu du registre MDR dans le registre IR

PC -> MAR

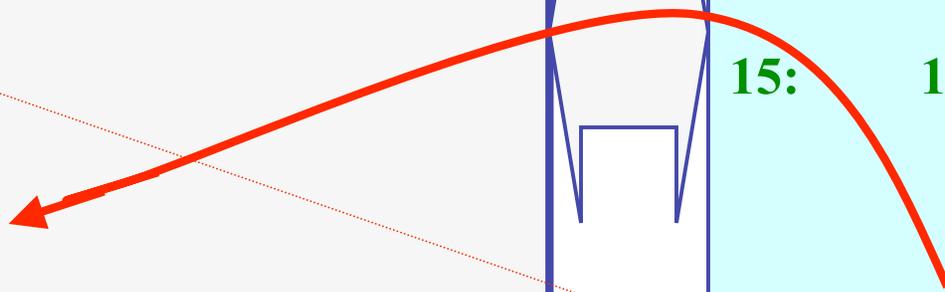
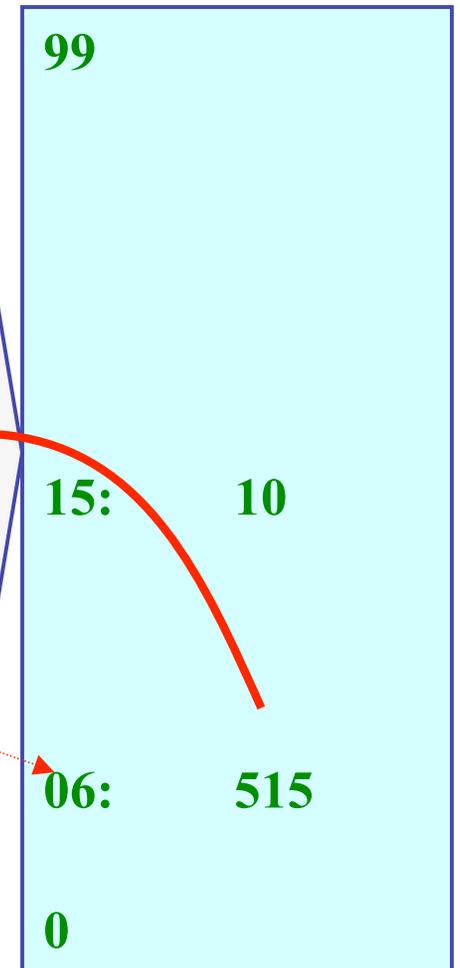
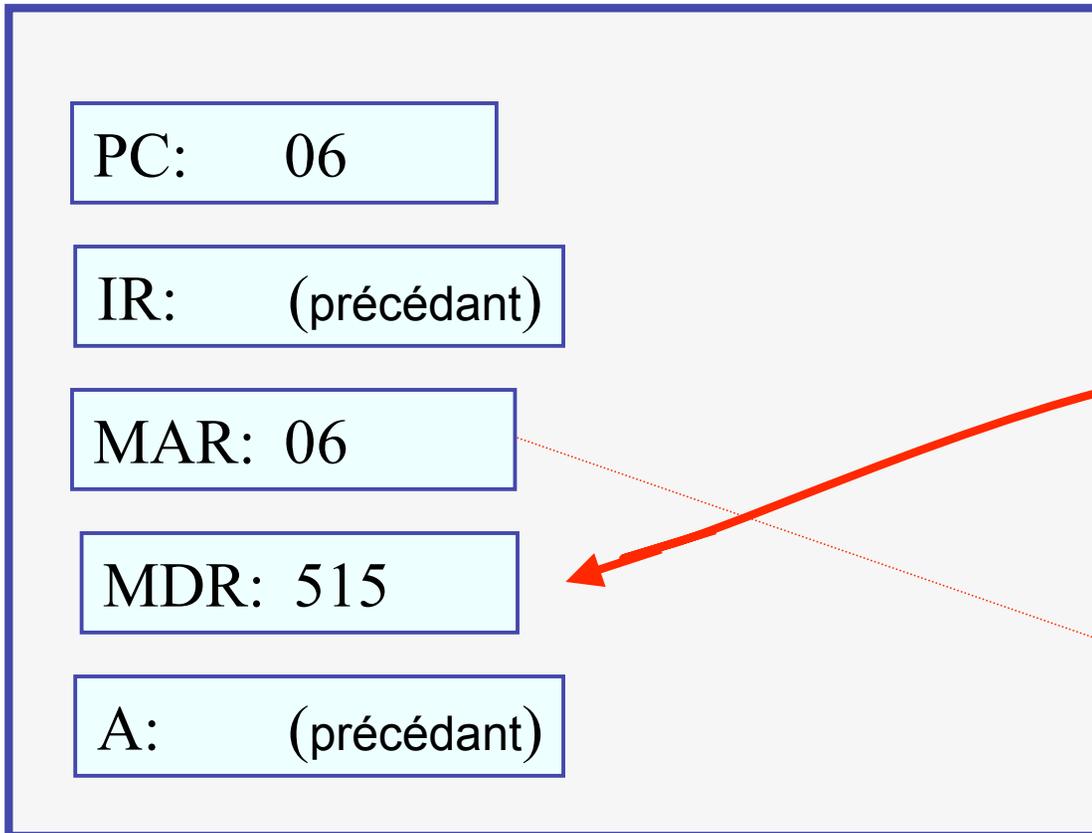


kc

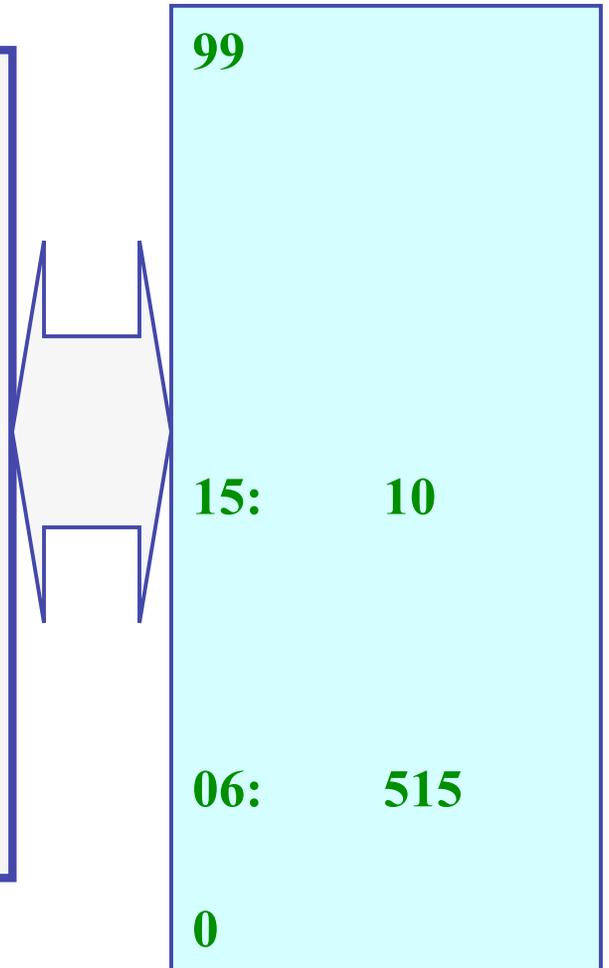
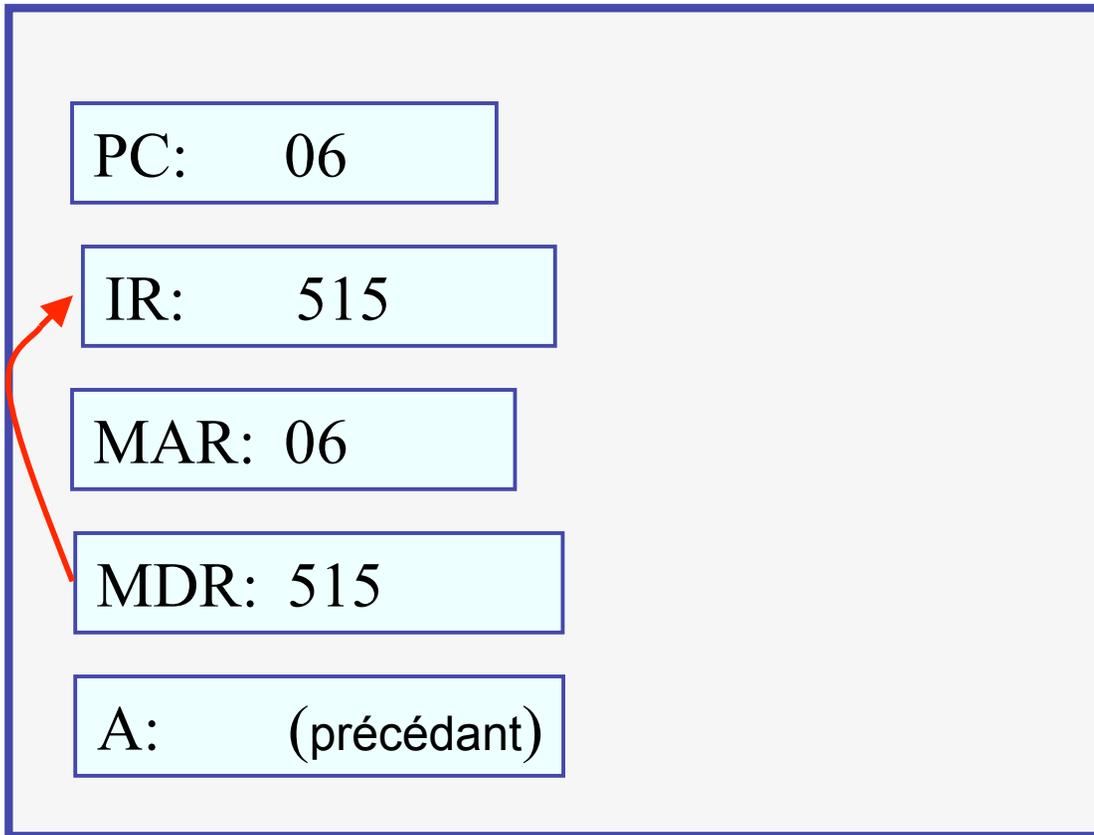
L'accès mémoire à l'adresse 06



Le contenu M[06] dans MDR:



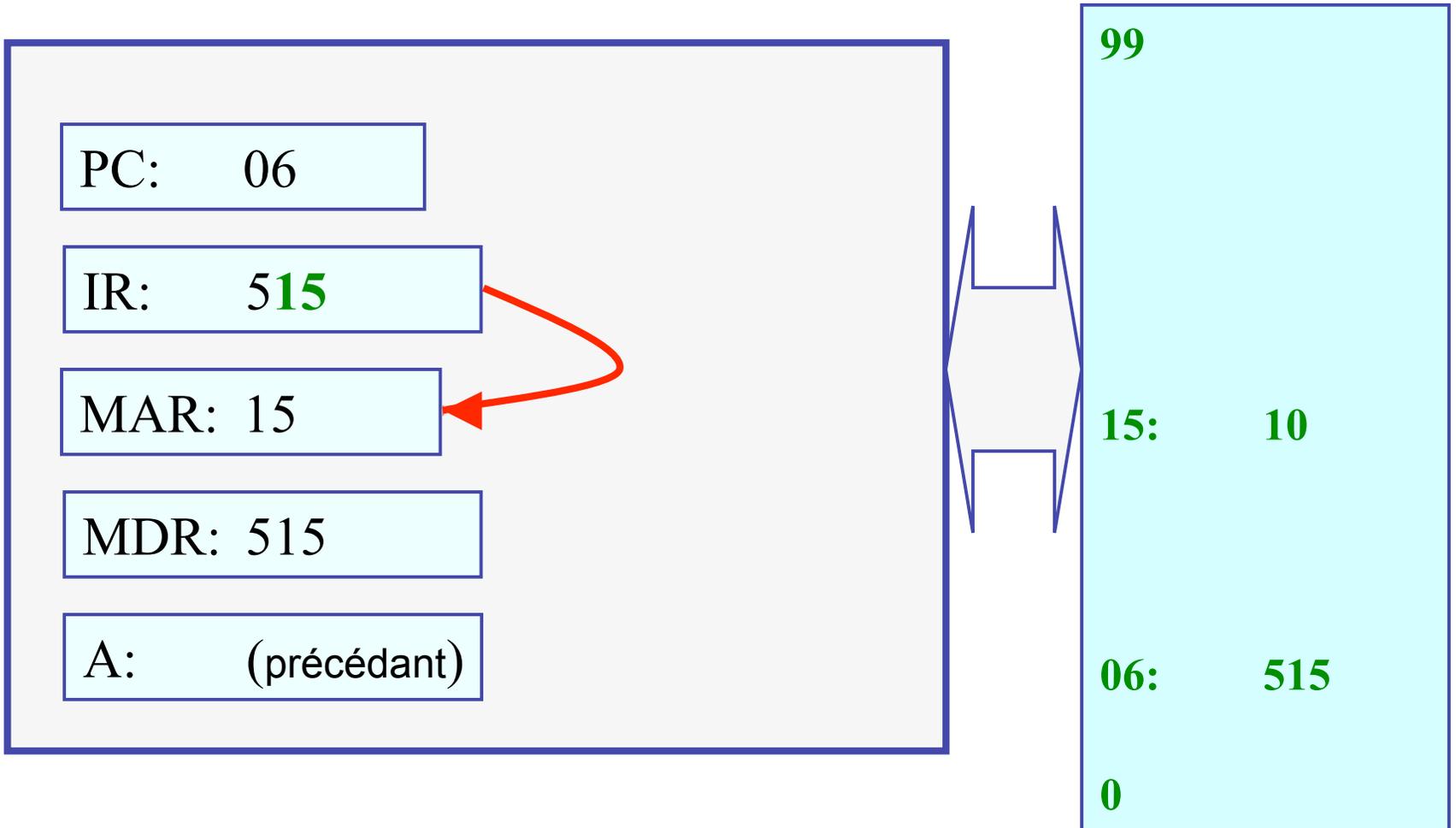
MDR -> IR



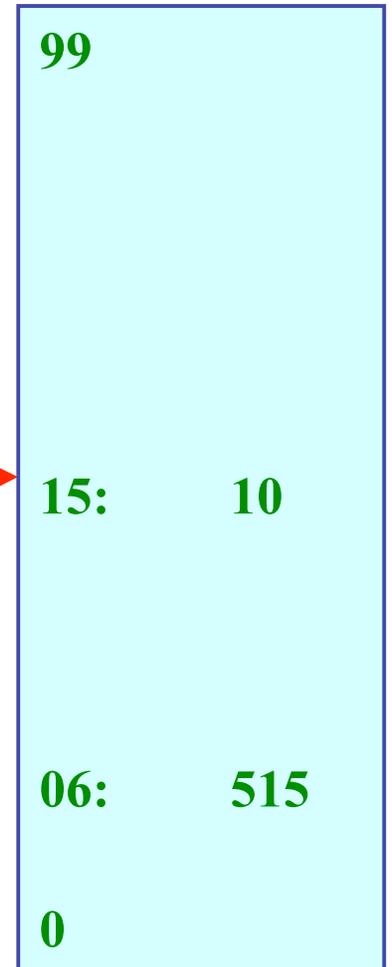
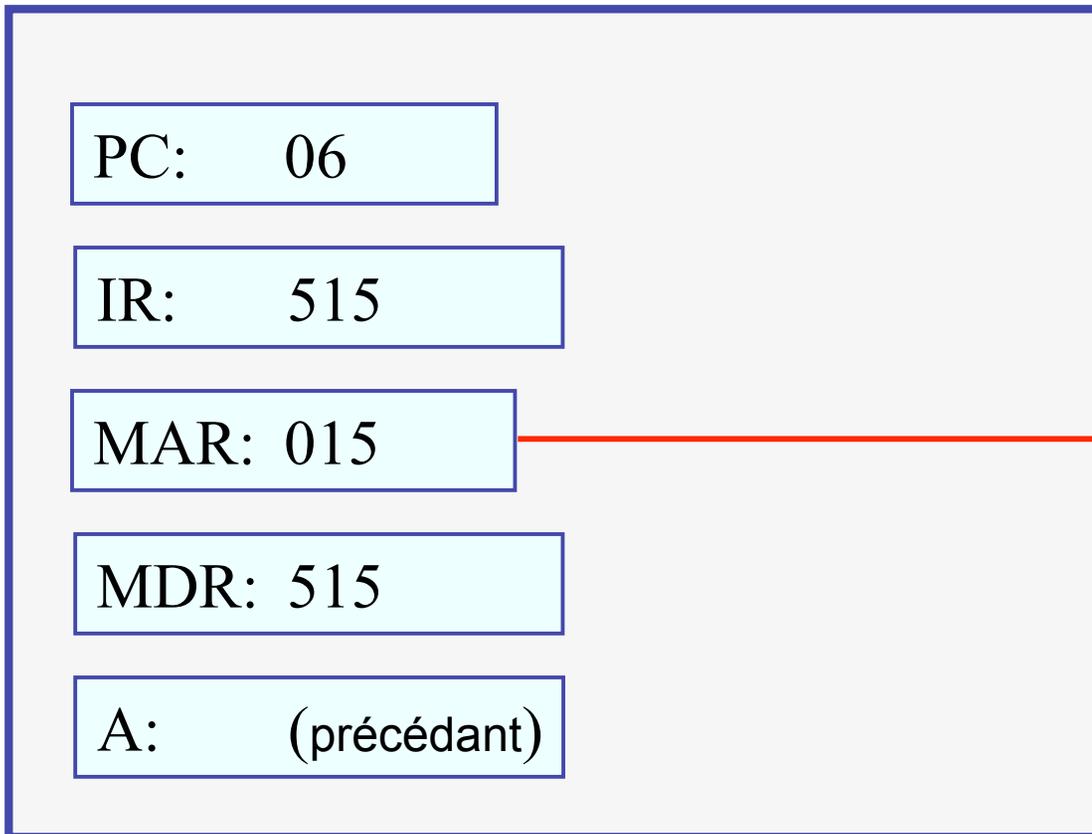
«Execute»

- Analyse de l'instruction
- L'unité de commande déclenche la séquence de micro-instructions nécessaires à la réalisation de l'instruction
- Incrémenter le compteur de programme
- Dépend de l'instruction
 - On continue l'exemple : LDA 15

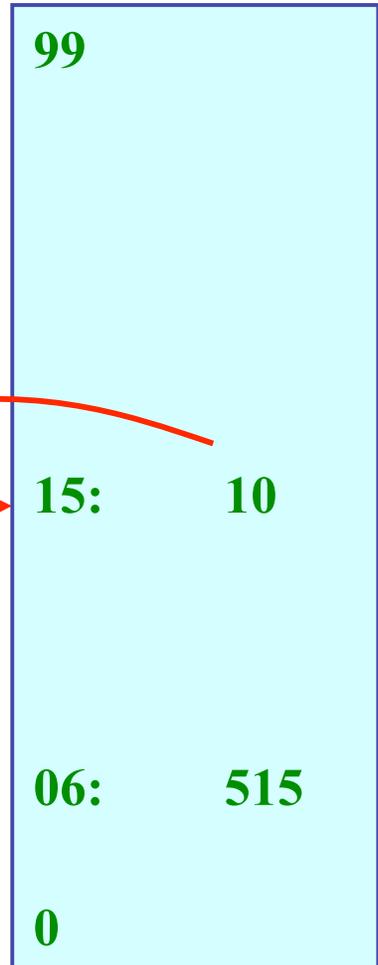
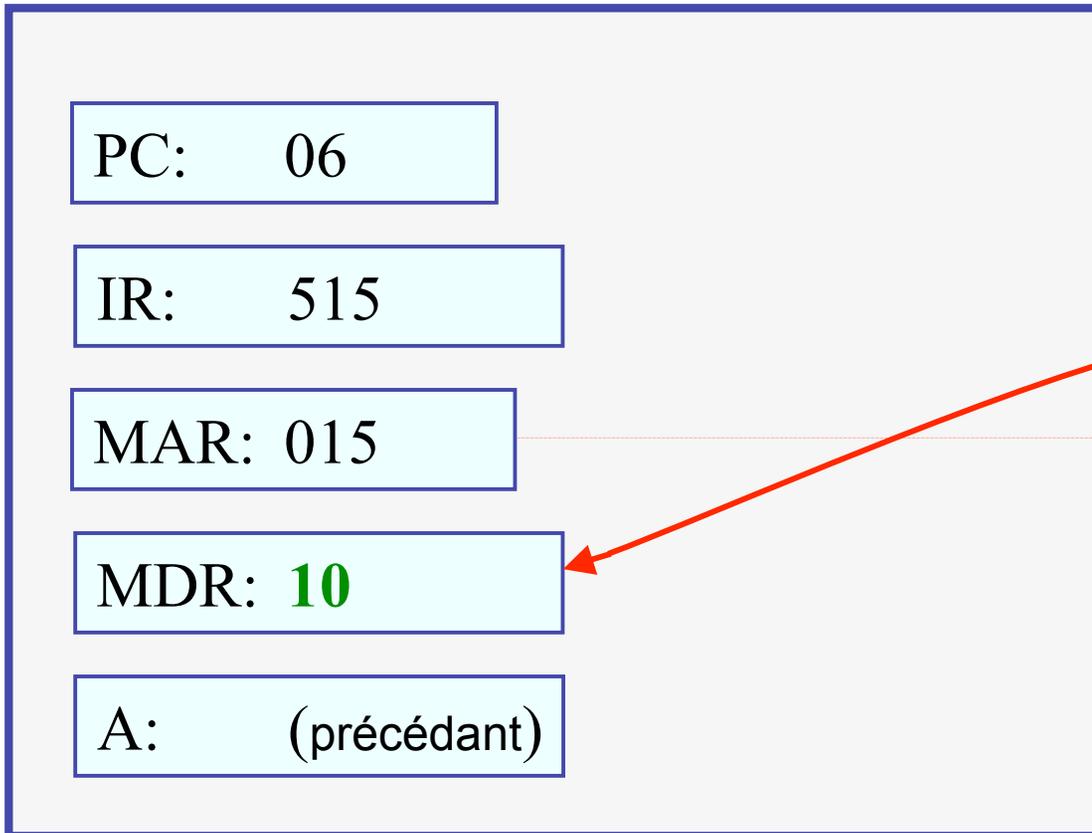
IR [partie adresse] -> MAR



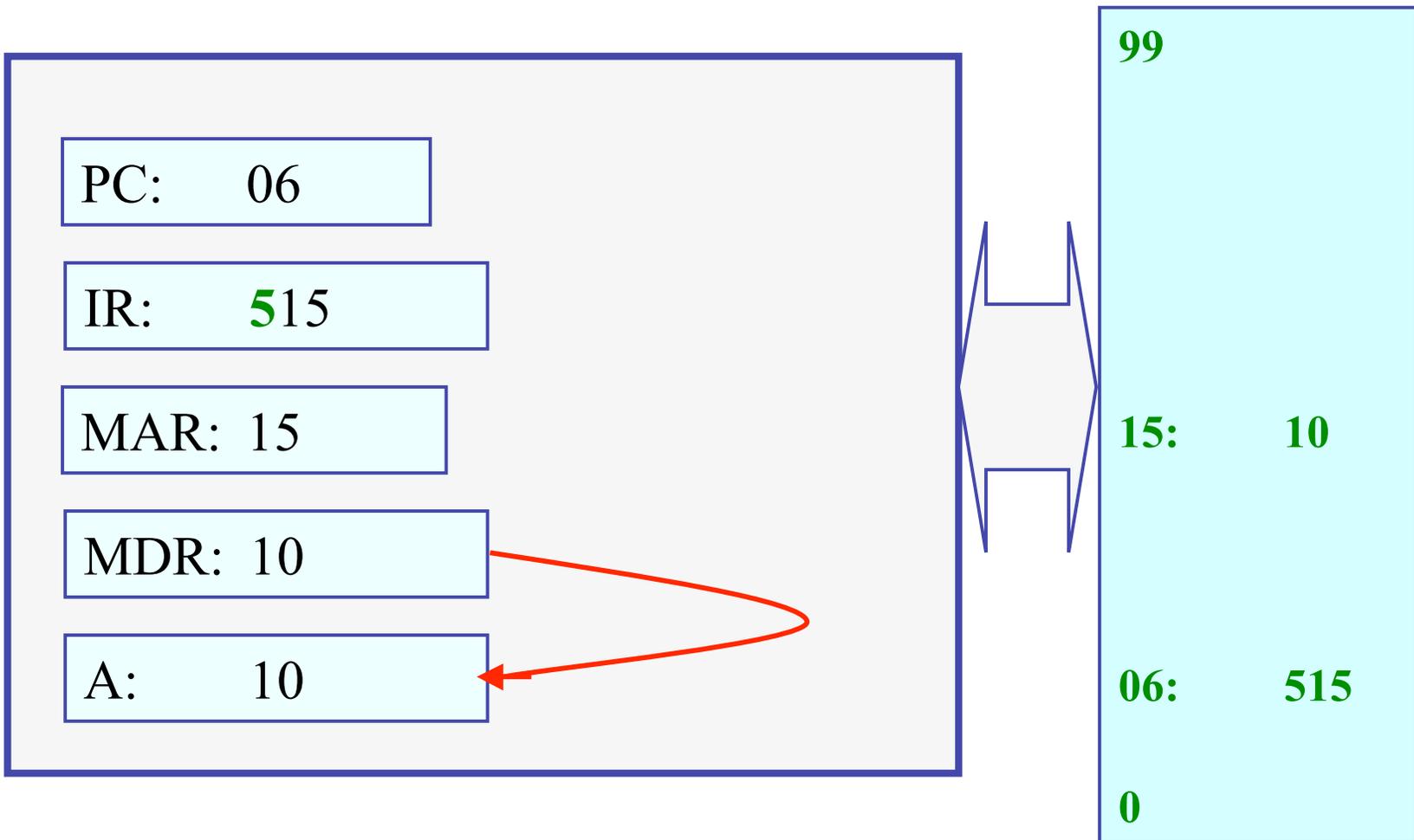
LOAD =>M[15] = 10 =>MDR



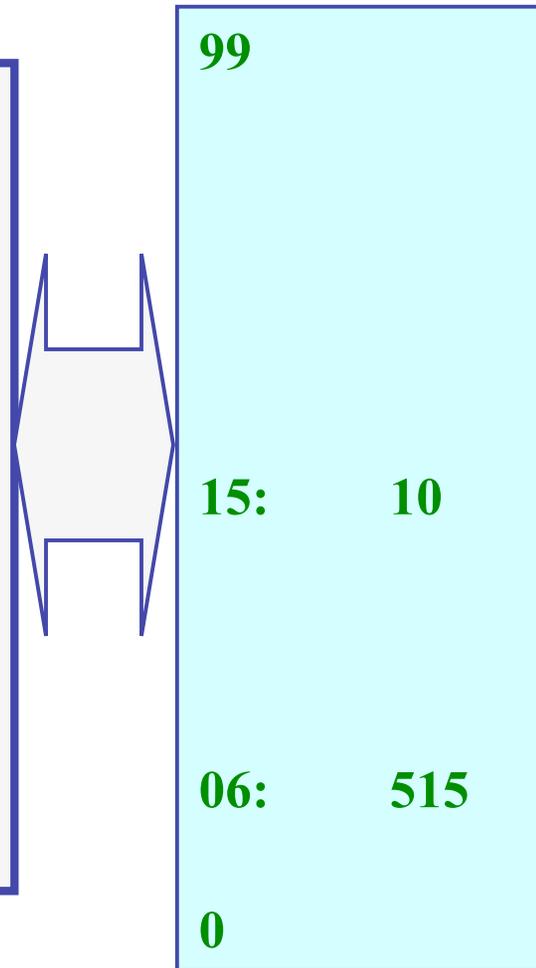
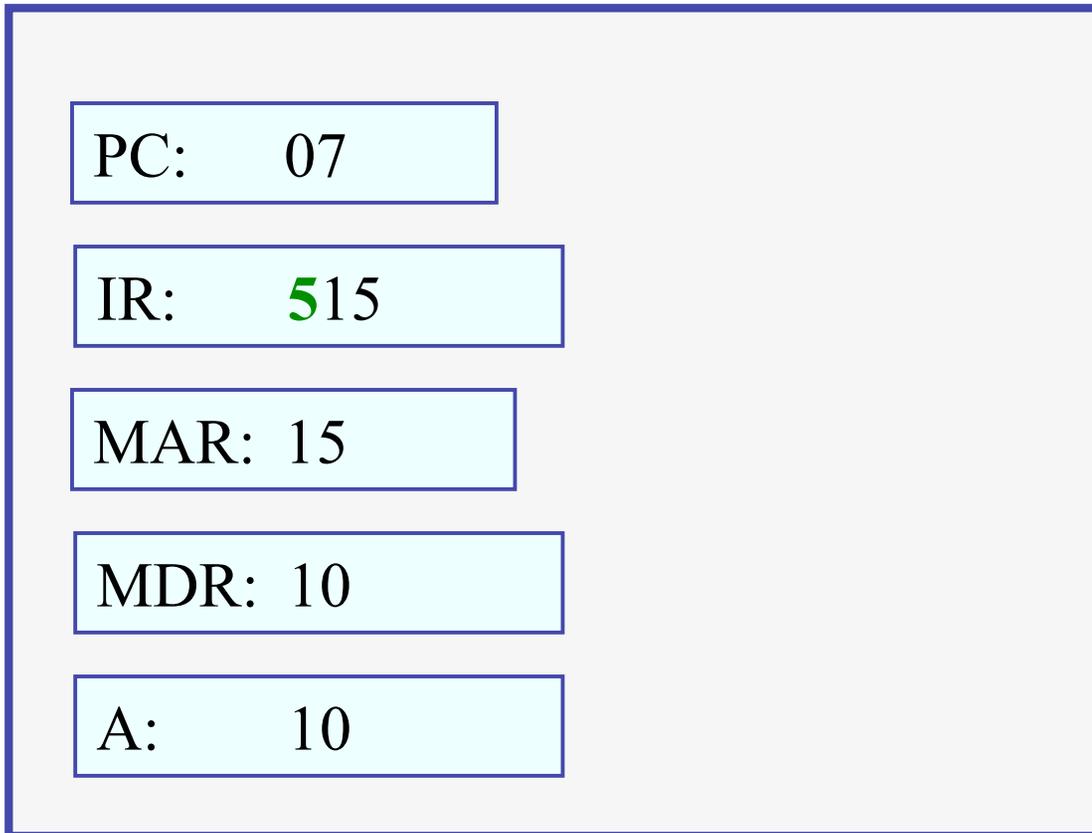
MDR = 10



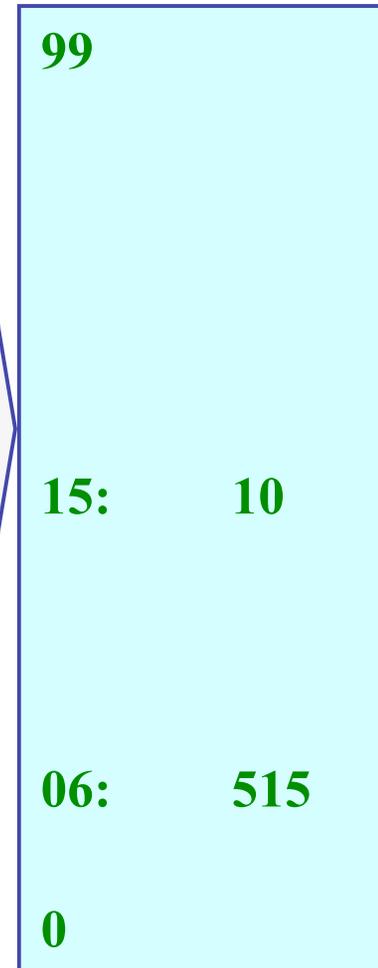
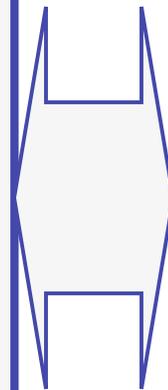
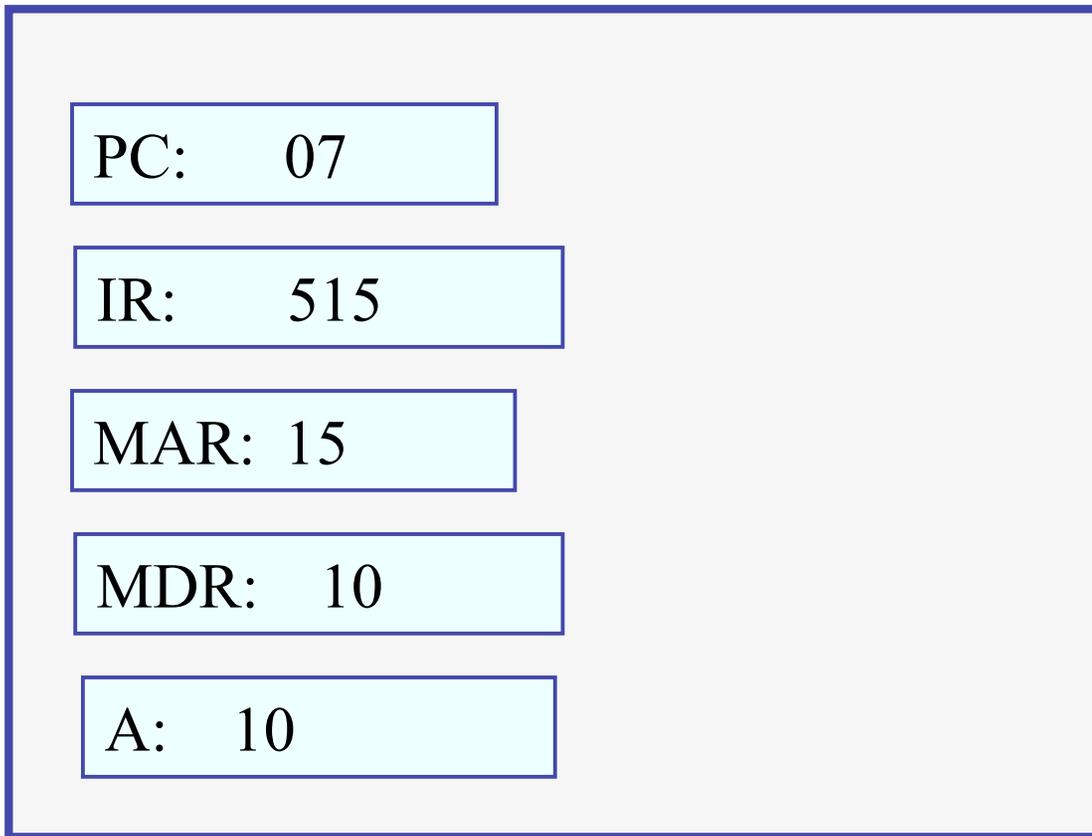
IR [op code] s'exécute: MDR -> A



PC + 1 -> PC



Fin !



Fetch-Execute Cycle de l'instruction Load

 PC -> MAR

 MDR -> IR

 IR(adresse) -> MAR

 MDR -> A

 PC + 1 -> PC

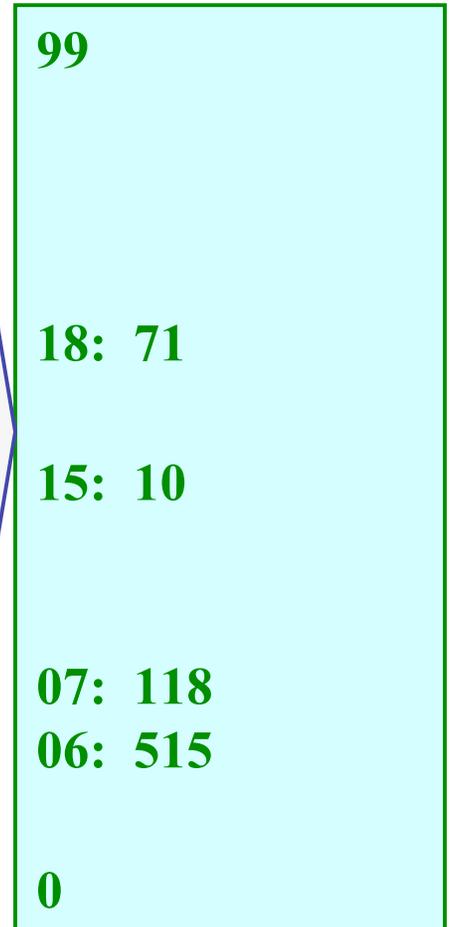
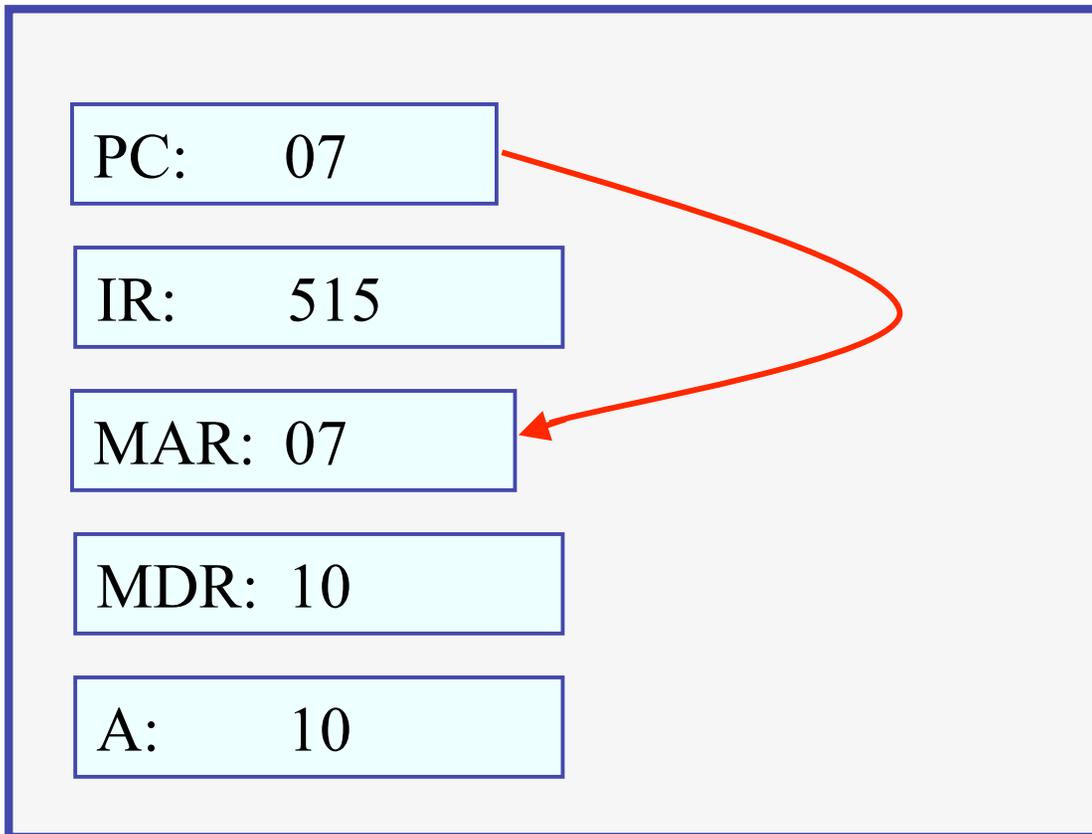
Fetch-Execute Cycle de l'instruction ADD

Maintenant:

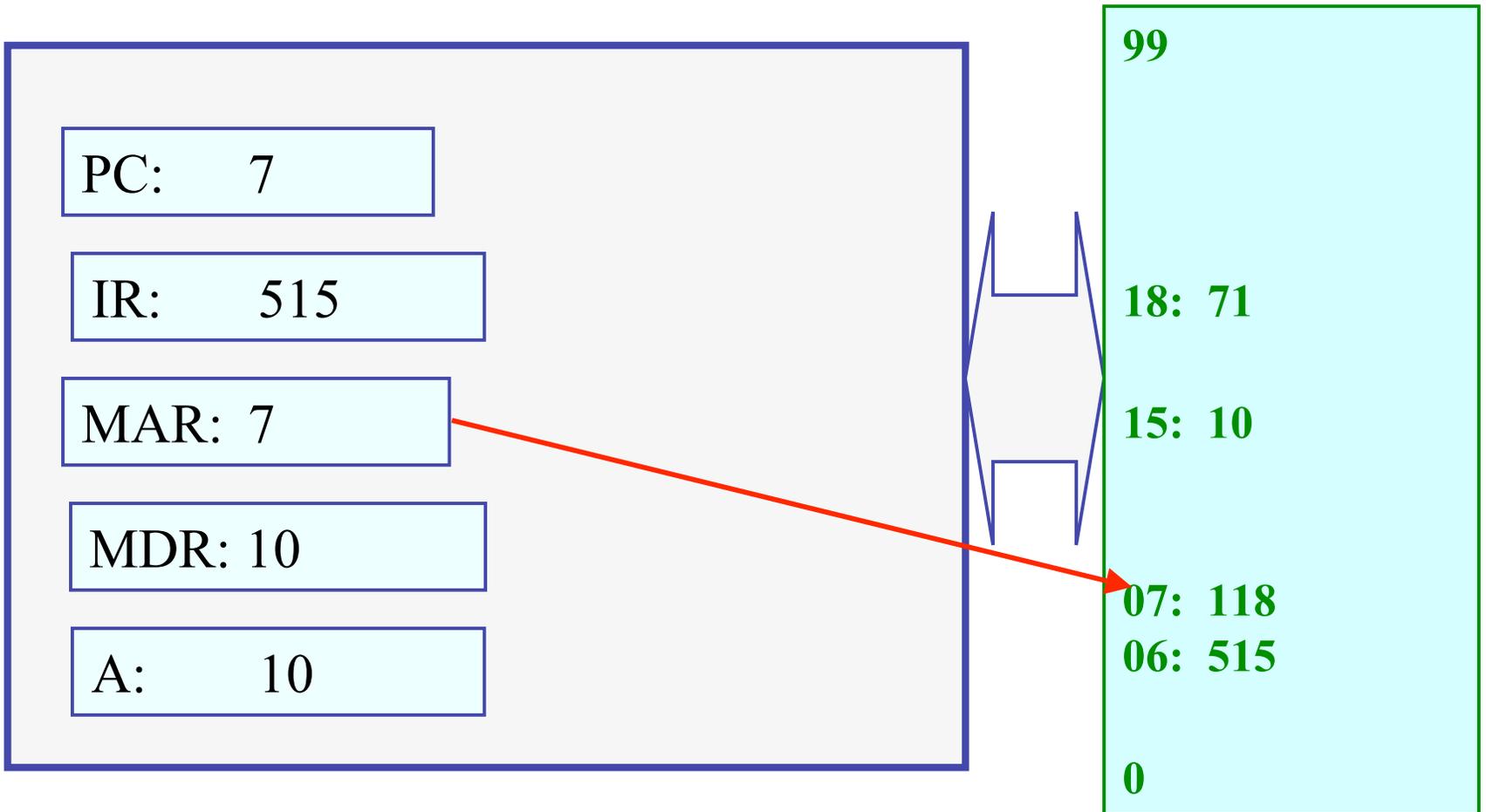
07 ADD 18

$M[18] = 71$, le reste est le même

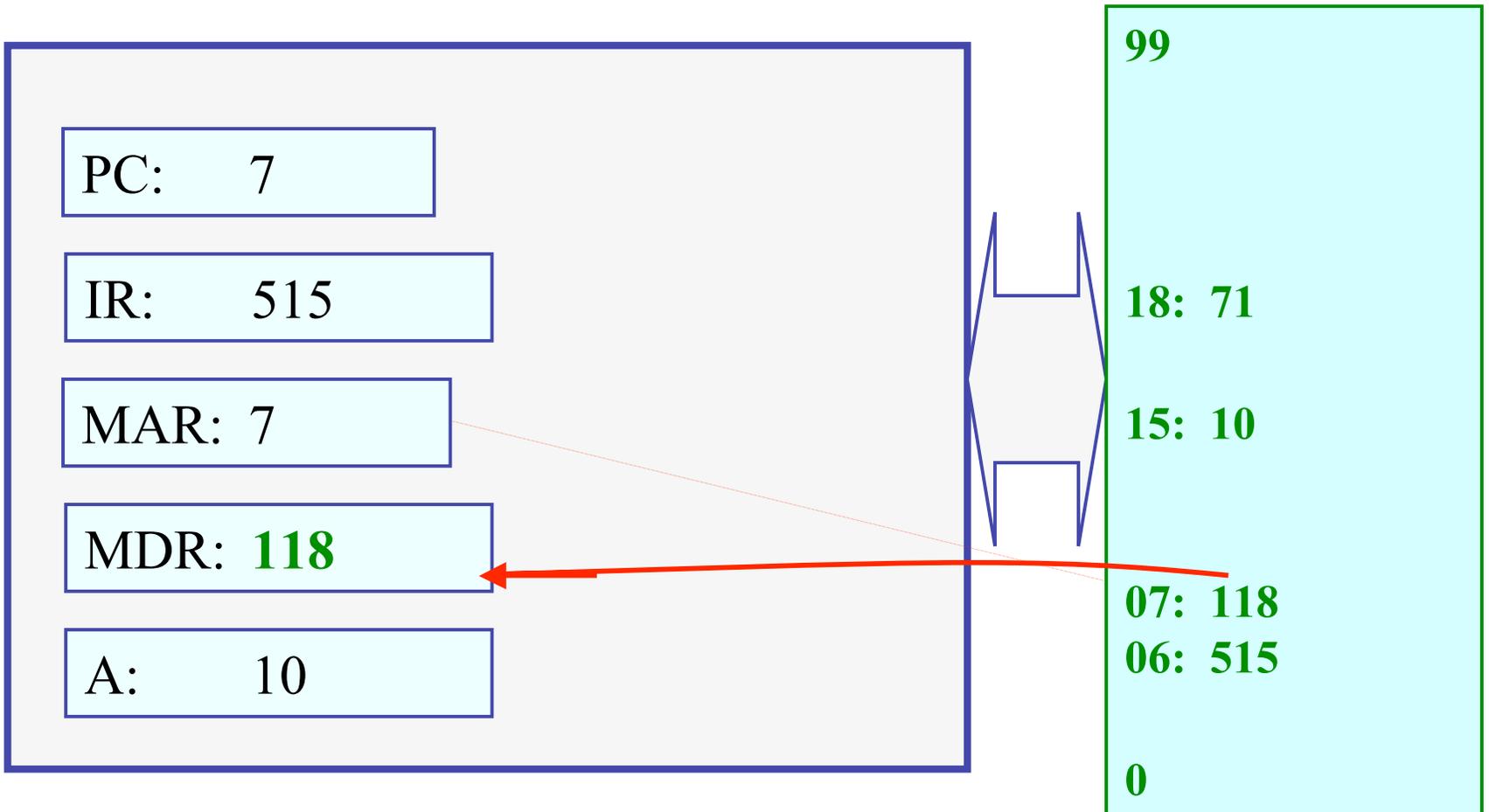
PC -> MAR



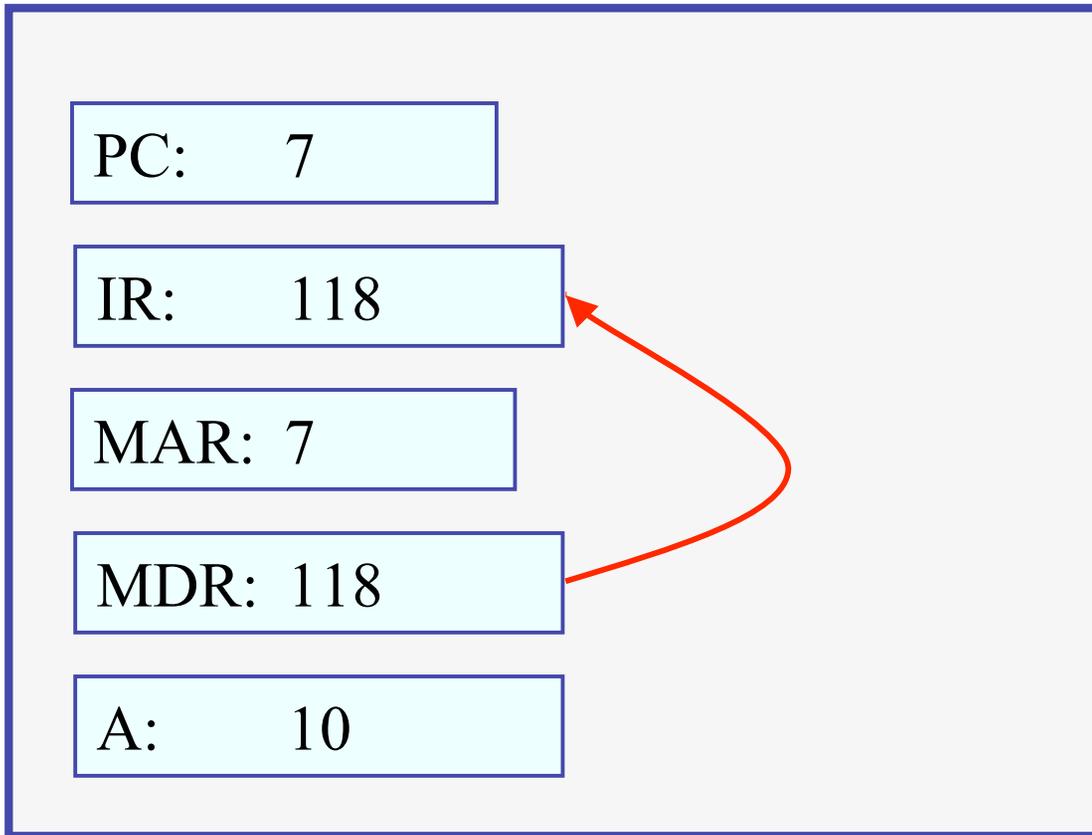
MAR accède l'adresse 7



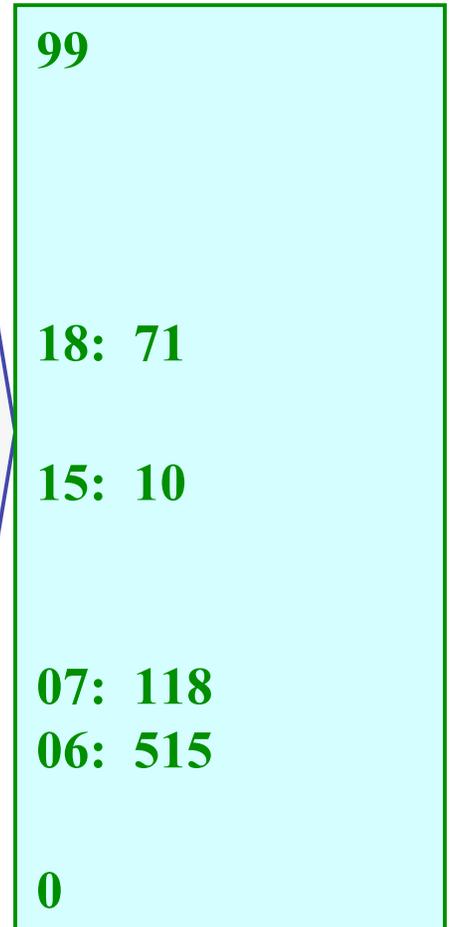
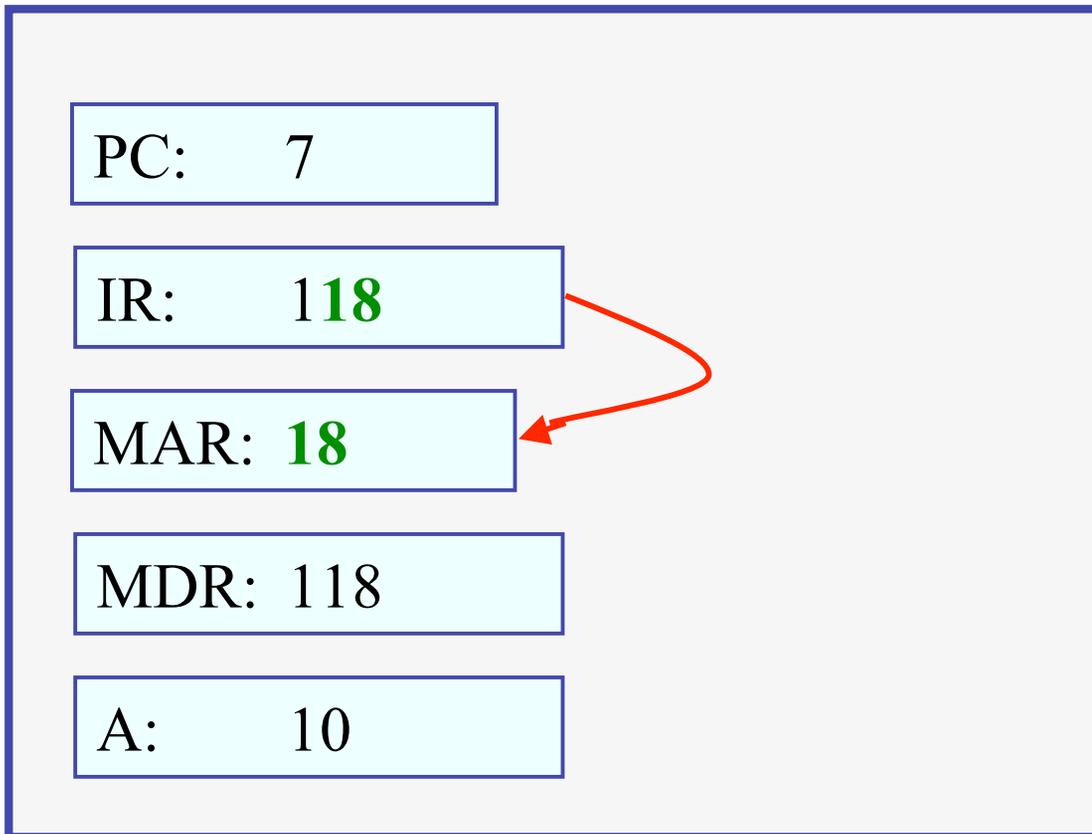
Contenu de M[7] -> MDR



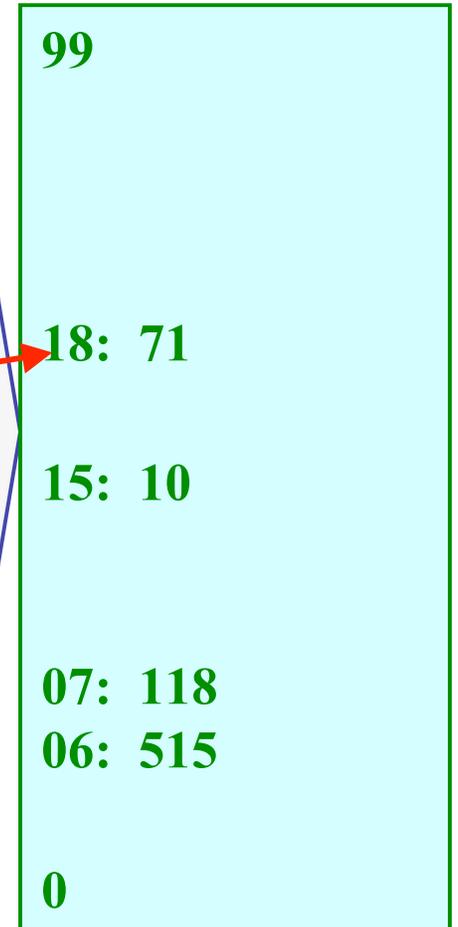
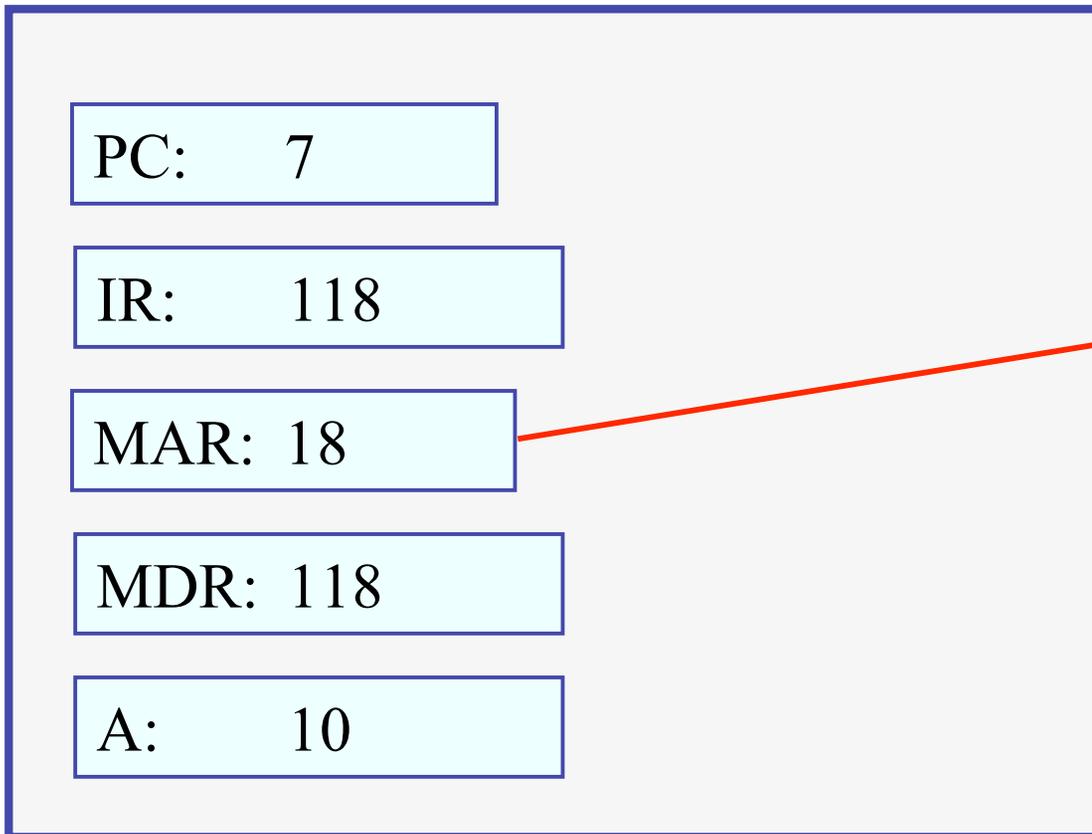
MDR -> IR



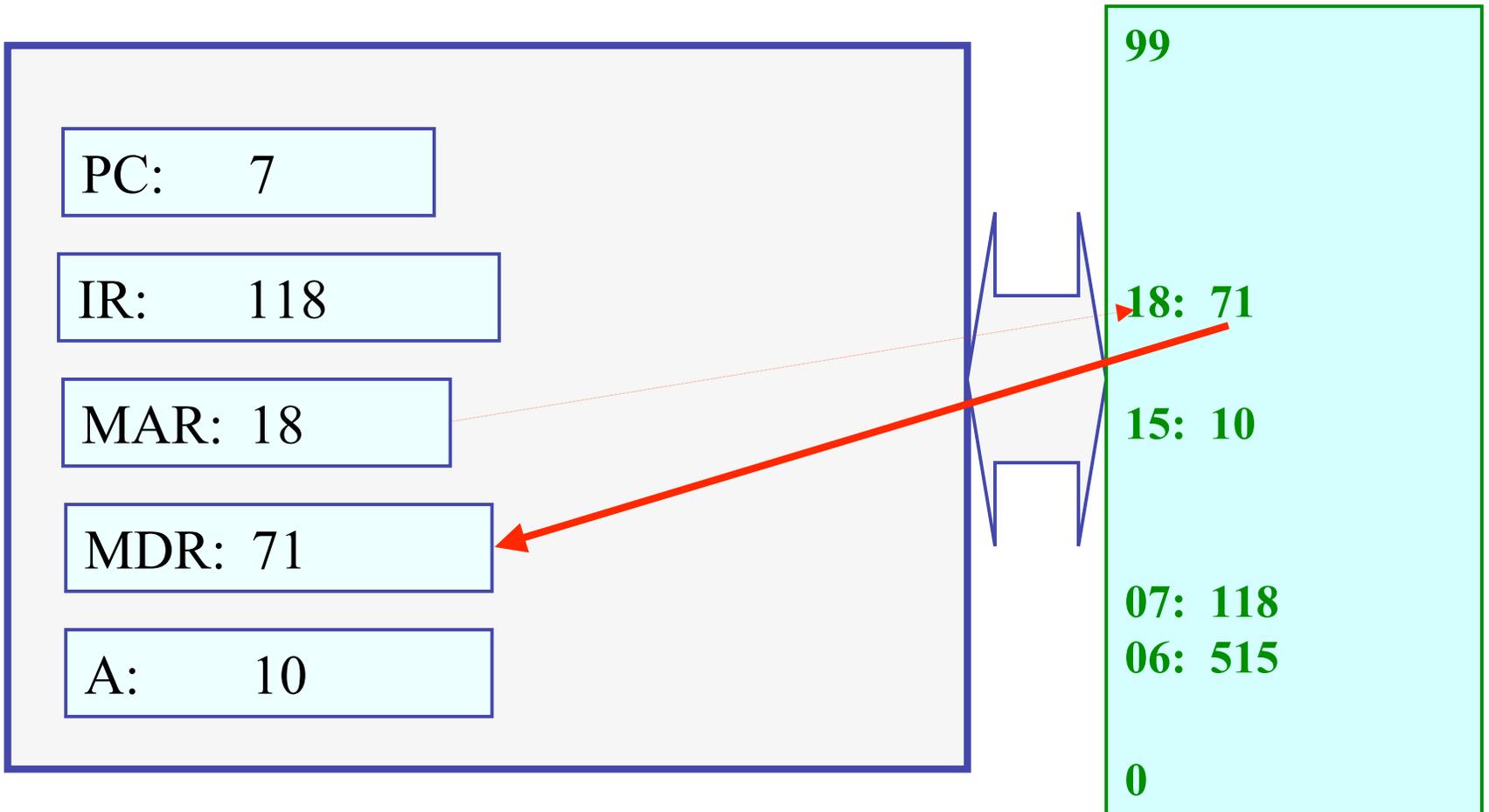
IR [adresse] -> MAR



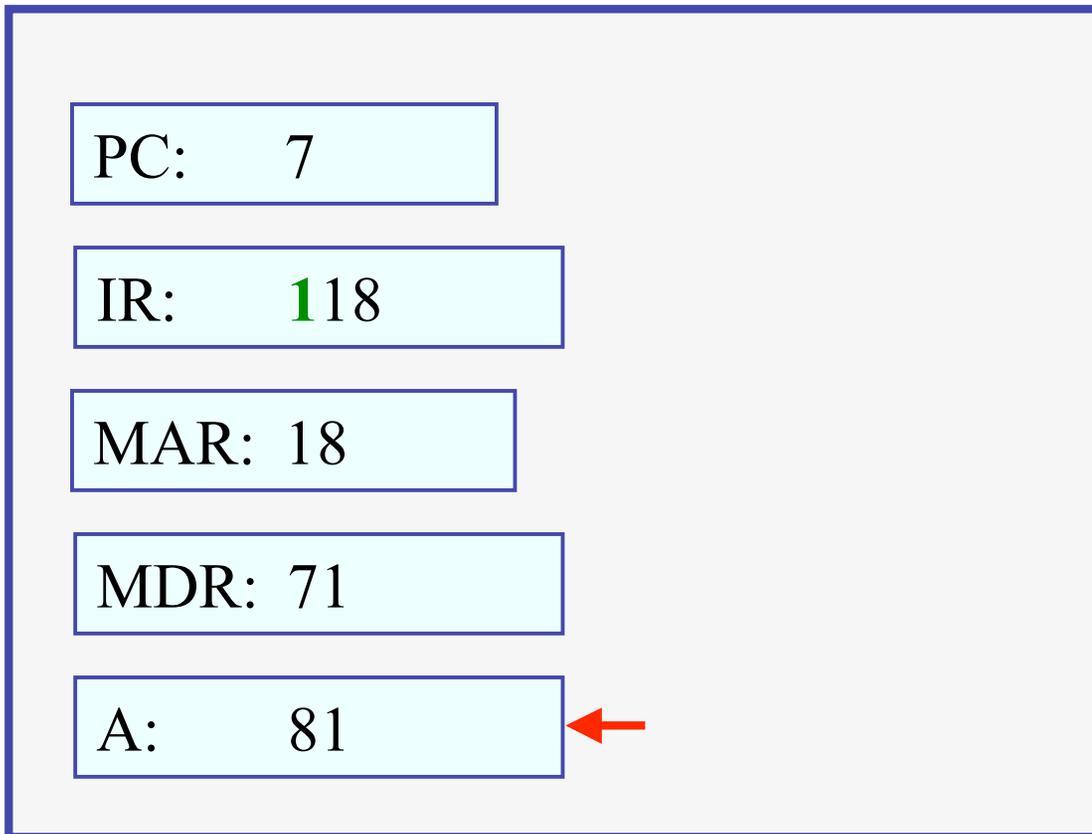
M[MAR] accédé



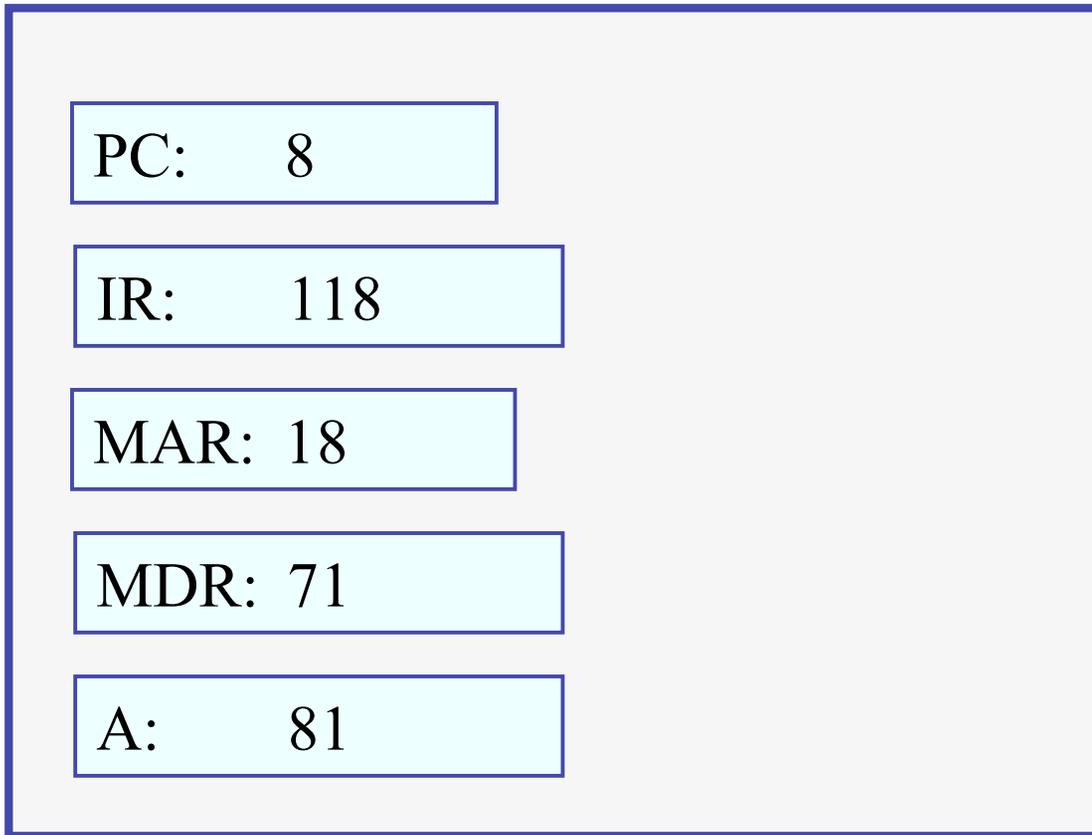
M [18] -> MDR



IR [opcode] exécuté: $A = A + \text{MDR}$



$$PC = PC + 1$$



Fetch-Execute Cycle de l'instruction ADD

 PC -> MAR

 MDR -> IR

 IR(address) -> MAR

 A + MDR -> A

 PC + 1 -> PC

Fetch-Execute Cycle de l'instruction STORE

 PC -> MAR

 MDR -> IR

 IR(address) -> MAR

 A -> MDR

 PC + 1 -> PC

LMC Fetch/Execute

SUB

PC → MAR

MDR → IR

IR[adr] → MAR

A – MDR → A

PC + 1 → PC

IN

PC → MAR

MDR → IR

IOR → A

PC + 1 → PC

OUT

PC → MAR

MDR → IR

A → IOR

PC + 1 → PC

HLT

PC → MAR

MDR → IR

BR

PC → MAR

MDR → IR

IR[adr] → PC

Branchement conditionnel

PC → MAR

MDR → IR

Si la condition est faux: PC + 1 → PC

Si la condition est vrai: IR[adr] → PC

LMC, la couche microarchitecture

