

ReCO

Reconfigurable Computing

Rappels

27/09/2009

Architecture des systèmes à processeur / MSR

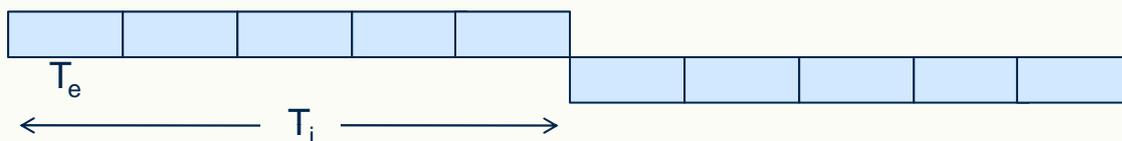
Michel Starkier

1

Traitement d'une instruction

- Le traitement d'une instruction s'effectue généralement en plusieurs étapes
- Exemple sans pipeline

<= traitement instruction 1 => <= traitement instruction 2 =>



- soit n le nombre d'étapes par instruction
- et T_e le temps de traitement d'un étape $T_i = nT_e$
- pour m instructions $T_t = m T_i = mnT_e$

27/09/2009

Reconfigurable Computing / MSR -EMI

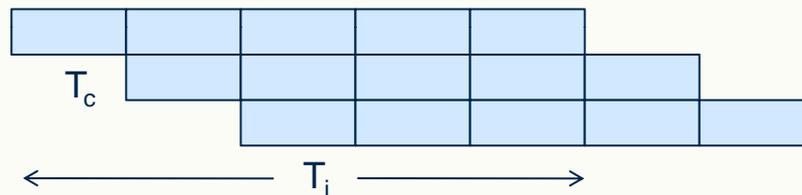
2

Traitement avec pipeline

Exemple avec pipeline

<= traitement instruction 1 =>

<= traitement instruction 2 =>



pour m instructions

$$T_t = T_i + (m-1)T_e = nT_e + (m-1)T_e = (n+m-1)T_e$$

pour un grand nombre d'instructions

$T_t = m T_e$ soit n fois moins que le traitement sans pipeline

Terminologie

Cycle time

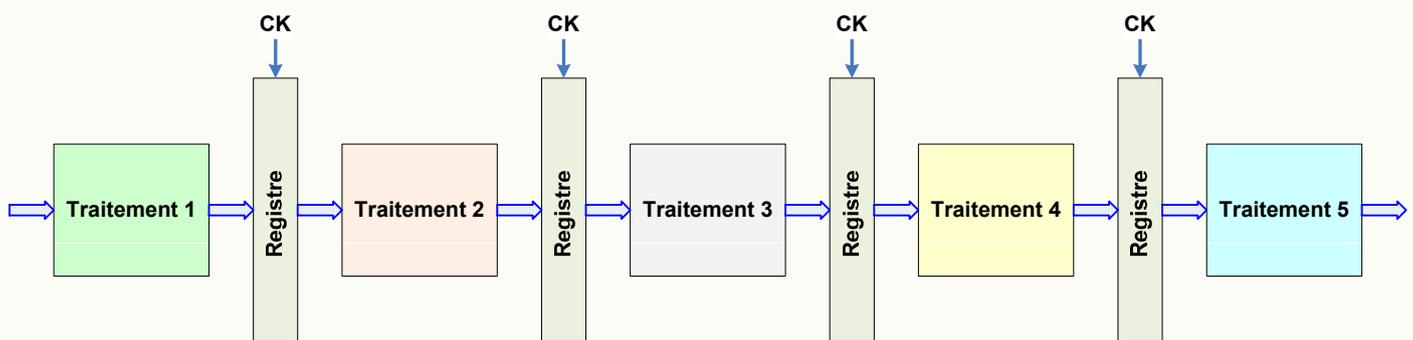
- Temps de traitement de chaque étape
- = une ou plusieurs période d'horloge

Flowthrough (latency) time

- temps de latence, d'attente du premier résultat
- (n cycles)

Architecture de pipeline

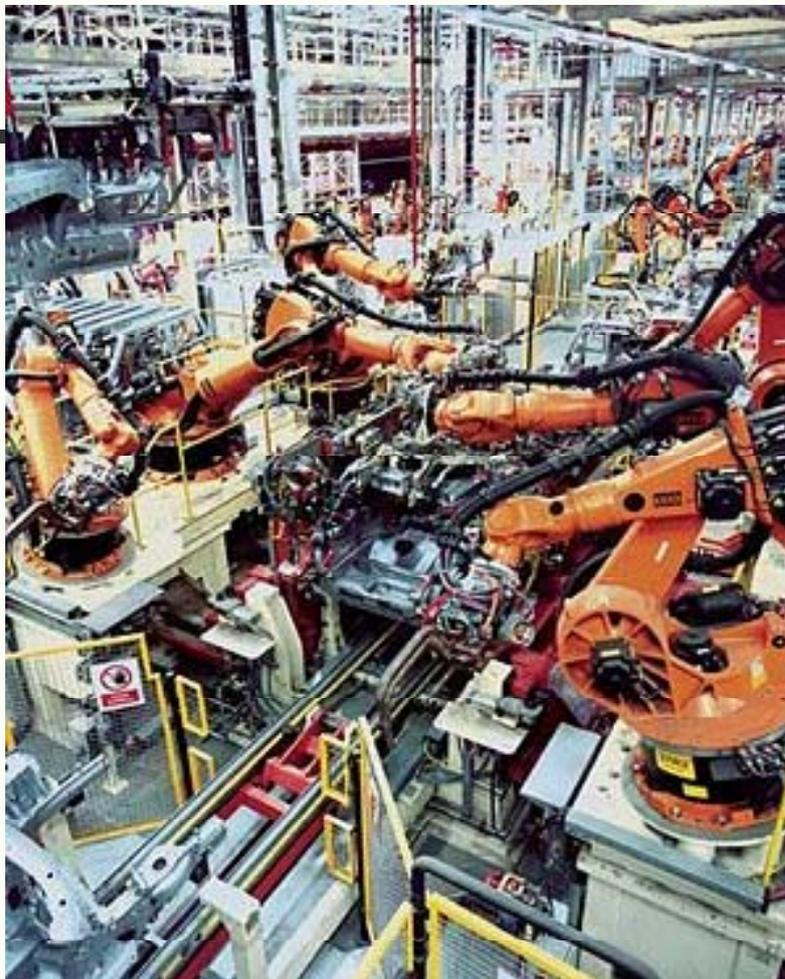
- Exemple de pipeline à 5 étages ou niveaux
- Le traitement s'effectue en 5 étapes (ou 5 cycles)
- Chaque étage est isolé par un registre
- Temps de cycle identique pour chaque étape (ou phase)
- Les traitements 1, 2, 3, 4 et 5 s'effectuent en parallèle



27/09/2009

Reconfigurable Computing / MSR -EMI

5



27/09/2009

Reconfigurable Computing / MSR -EMI

Image Wikipédia

6

- Suivant les types de processeurs :
 - Pipeline à 3 niveaux :
 - Fetch, decode , execute
 - Pipeline à 5 niveaux (processeurs ARM 9, MIPS) :
 - IF, ID, EX, MEM, WB
 - Jusqu'à 25 niveaux ! (20 niveaux pour le Pentium 4)

Pipeline à 5 niveaux : phases

- **Phase IF (Instruction Fetch)** : recherche de l'instruction
- **Phase ID (Instruction Decode)**: décodage de l'instruction et lecture des registres opérandes
- **Phase EX (Execution)**: exécution de l'opération ou calcul de l'adresse de mémoire
- **Phase MEM (Memory)**: accès de la mémoire ou écriture dans le PC de l'adresse de saut
- **Phase WB (Write Back)**: écriture dans un registre du résultat de l'opération

Amélioration des performances

● Temps d'exécution d'un programme

$$T_e = \text{Nbr_instr.} \times \text{Nbr_cycle} \times T_{\text{cycle}}$$

- Diminuer Nbr_instr
- Diminuer Nbr_cycle et Tcycle

Architecture des systèmes à processeur / MSR

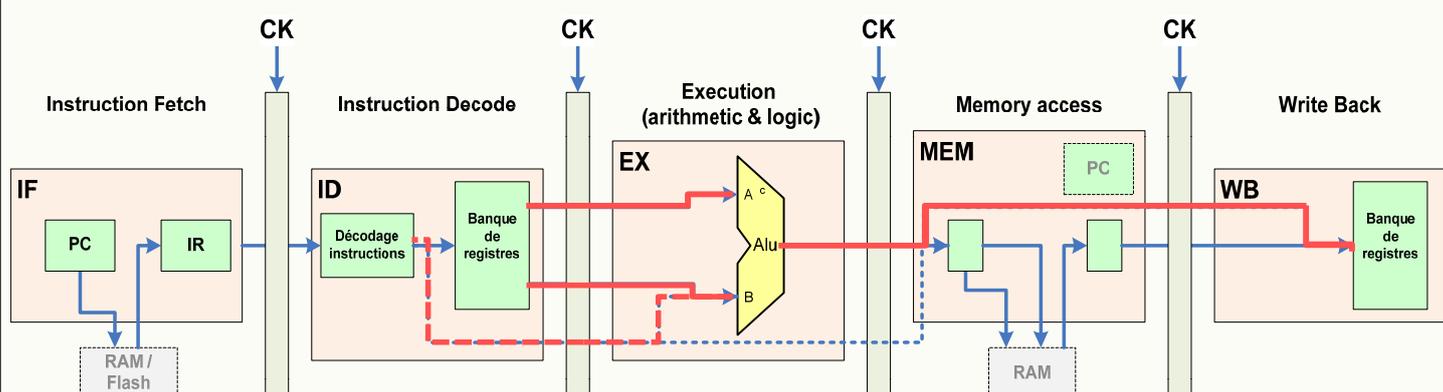
Traitement des instructions (1/3)

● Adressage direct (par registre) :

- ADD Rn, Rm, Rp => Rn = Rm + Rp
- SUB Rn, Rm, Rp => Rn = Rm - Rp

● Adressage direct & immédiat :

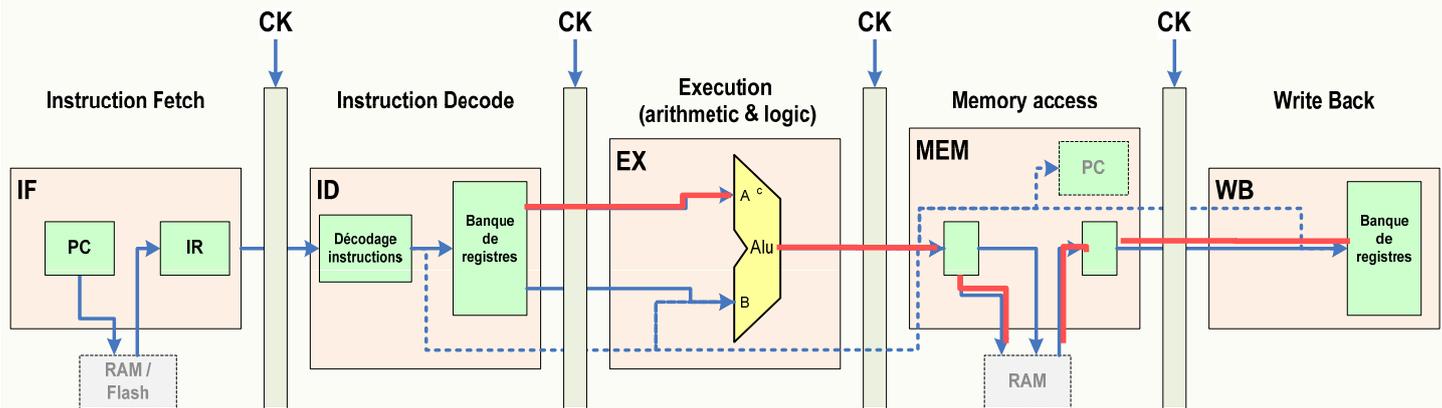
- ADD Rn, Rm, #P => Rn = Rm + P



Traitement des instructions (2/3)

● Transferts mémoire / registre:

- LDR Rn, [Rp] Rn ← mem [Rp]
- STR Rn, [Rp] mem [Rp] ← Rn



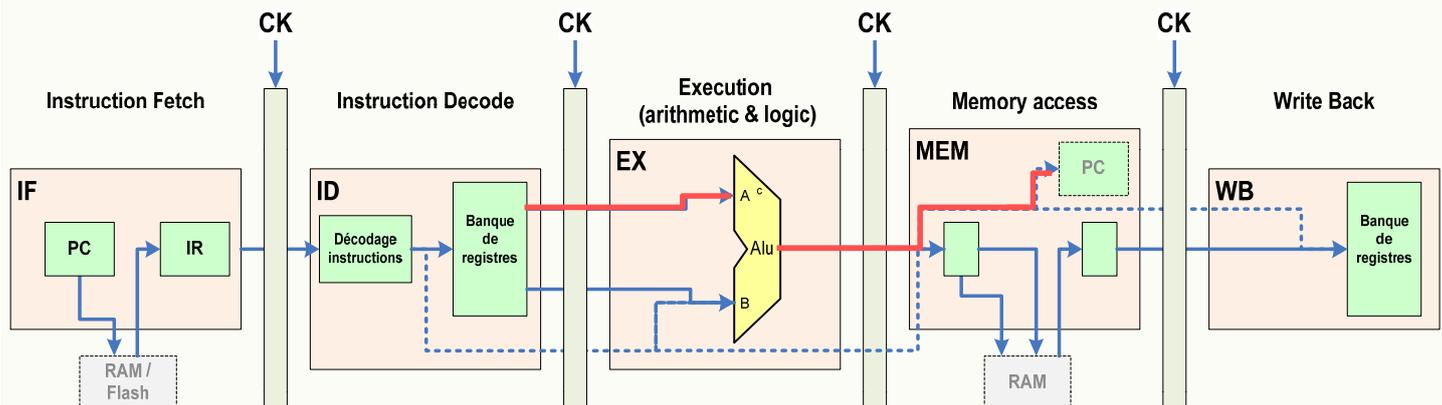
27/09/2009

Reconfigurable Computing / MSR -EMI

Traitement des instructions (3/3)

● Instructions de contrôle:

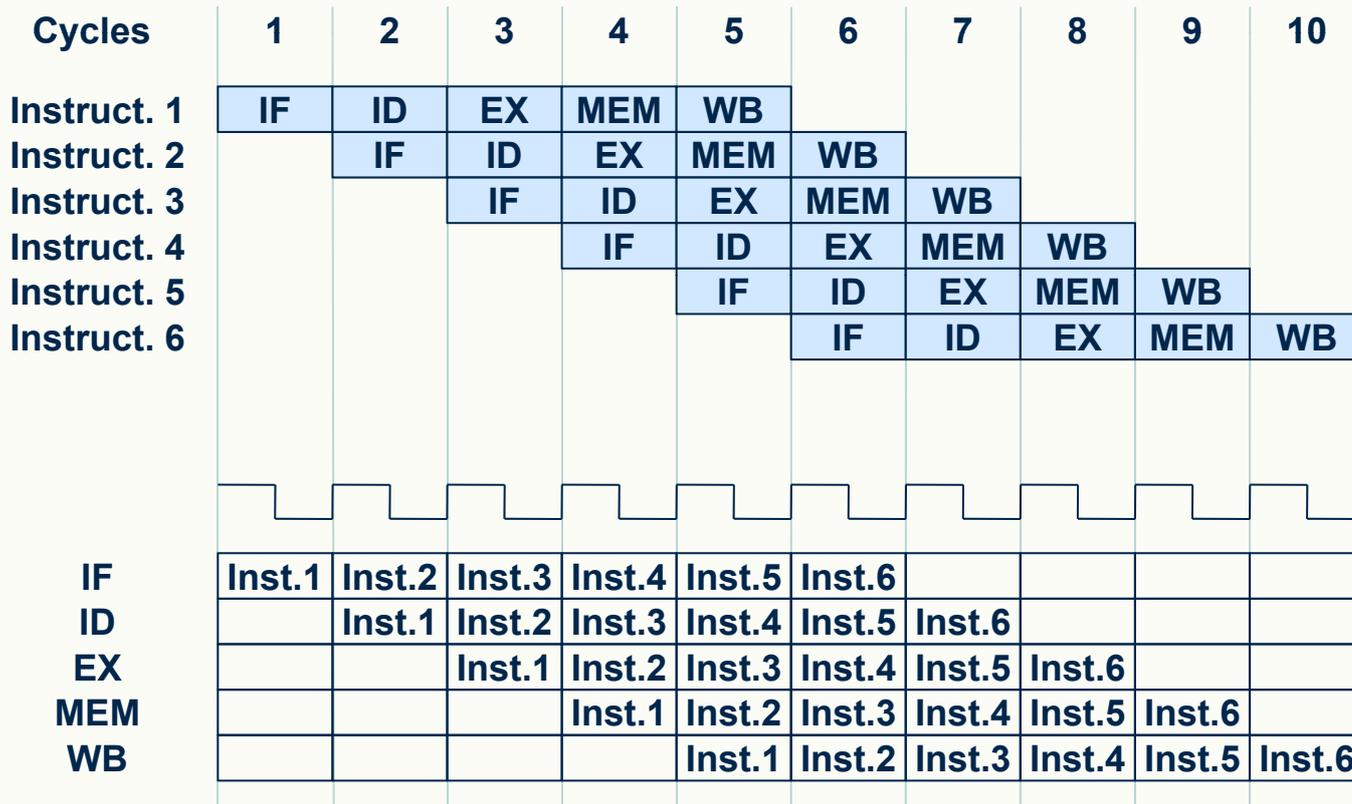
- BEQ LABEL if résultat = 0 jump to LABEL



27/09/2009

Reconfigurable Computing / MSR -EMI

Pipeline à 5 niveaux

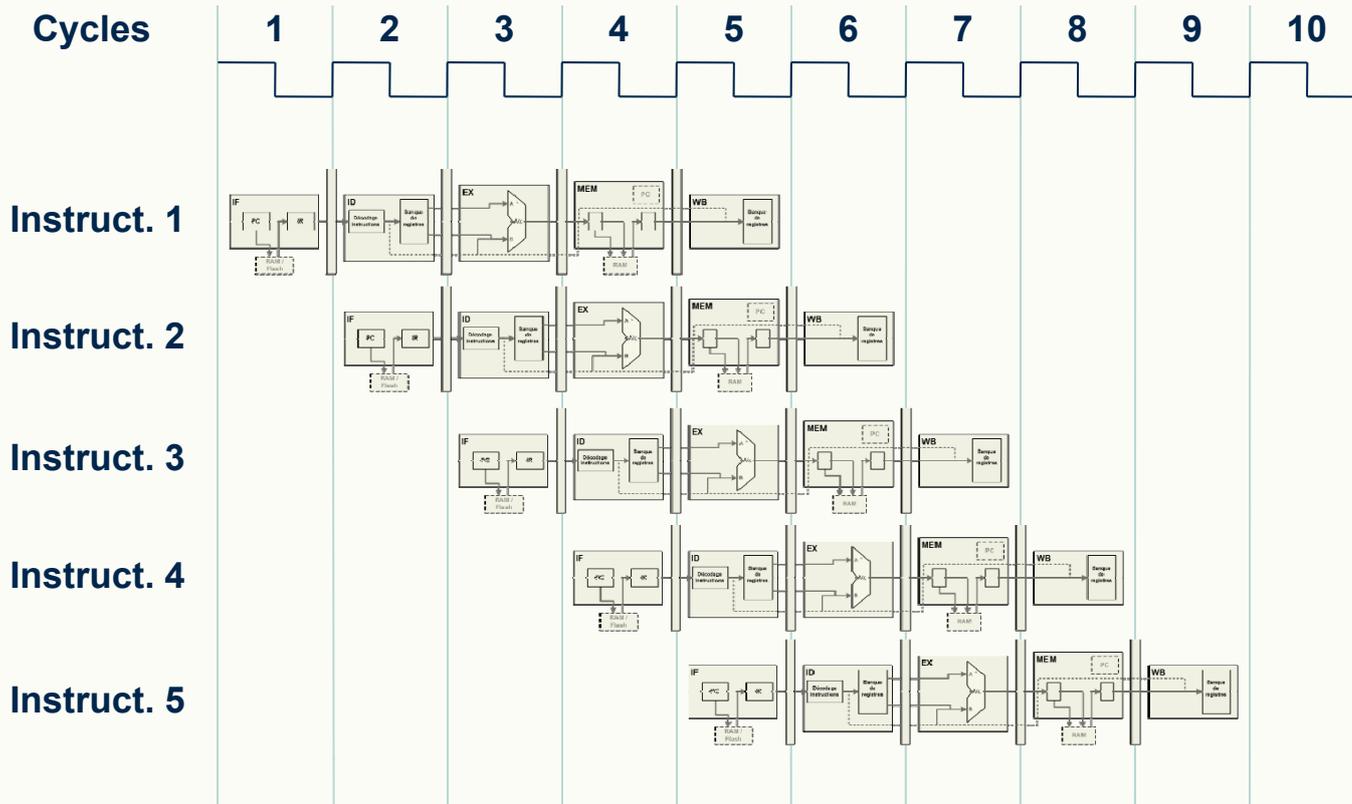


27/09/2009

Reconfigurable Computing / MSR -EMI

13

Pipeline à 5 niveaux

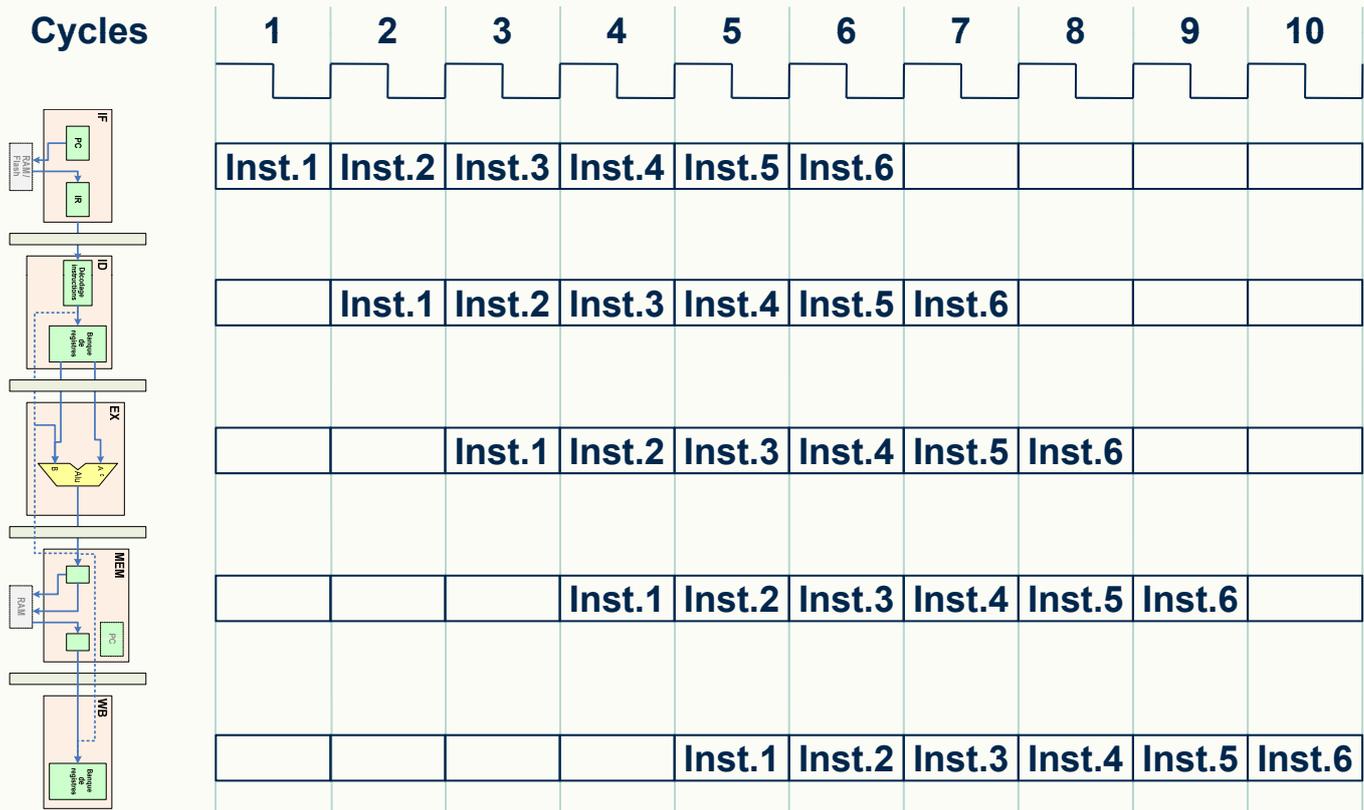


27/09/2009

Reconfigurable Computing / MSR -EMI

14

Pipeline à 5 niveaux



Cours ReCo

ALEAS DE PIPELINE

- **Les aléas sont inhérents au parallélisme**
- ***Aléas structurels***
 - **conflits d'accès aux ressources**
- ***Aléas de données***
 - **modification de l'ordre d'accès aux opérandes**
- ***Aléas de contrôle***
 - **décision de branchement**

Aléas de données

- **Dépendances entre instructions = même opérandes**
- **Si dépendance, trois types d'aléa de données possibles**
 - ***RAW (read after write):***
L'instruction n+x risque de lire une source avant que l'instruction n ne l'ait modifiée.
 - ***WAR (write after read):***
L'instruction n+x risque d'écrire dans une destination avant que l'instruction n ne l'ait utilisée comme source
 - ***WAW (write after write):***
L'instruction n+x risque d'écrire dans une destination avant que l'instruction n n'ait écrit dans cette destination : l'ordre normal d'écriture est modifié

Aléas de données

- Exemples de dépendances
- Ces dépendances n'entraînent pas forcément des aléas

- **RAW (read after write):**

ADD R1, R2, R3
SUB R5, R1, #2

$$R1 = R2 + R3$$

$$R5 = R1 - 2$$

- **WAR (write after read):**

ADD R1, R2, R3
SUB R2, R4, #2

$$R1 = R2 + R3$$

$$R2 = R4 - 2$$

- **WAW (write after write):**

ADD R1, R2, R3
AND R5, R1, R2
SUB R1, R4, #2

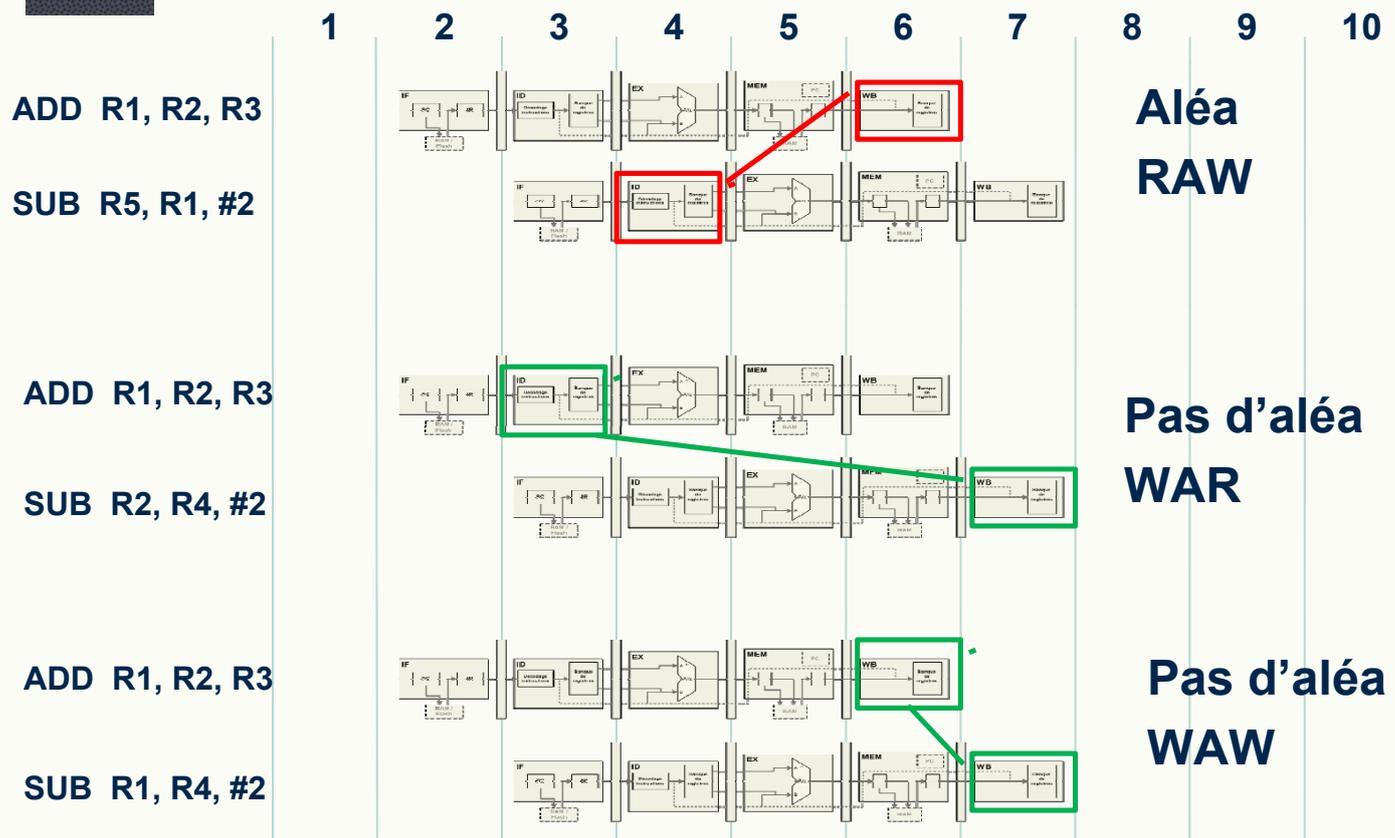
$$R1 = R2 + R3$$

$$R1 = R4 - 2$$

Question

- Quels aléas sont possibles avec l'architecture de pipeline à 5 niveaux présentée précédemment ?
- Tous ? Aucun ? Certains ?

Aléas de données : exemples



27/09/2009

Reconfigurable Computing / MSR -EMI

21

Exercice : dépendances et aléas de pipeline

- Indiquez les dépendances de types RAW, WAR et WAW

- 1 LDR R1, [R0]
- 2 LDR R2, [R1]
- 3 ADD R6, R5, R4
- 4 ADD R3, R1, R2
- 5 LDR R4, [R6]
- 6 SUB R2, R0, R4
- 7 ADD R7, R1, #4
- 8 ADD R4, R1, R3
- 9 SUB R6, R7, R4

Cours ReCo

RESOLUTION DES ALEAS

27/09/2009

Reconfigurable Computing / MSR -EMI

23

Résolution des problèmes d'aléas

- **Méthodes simples (et pénalisantes) :**
 - **Hardware:** arrêter le pipeline (stall / break)
 - **Software :** insérer des NOPs (no opération)

- **Calcul de la pénalité et de l'IPC (nombre d'instruction par cycle)**

- **Exemple :**
 - Arrêt de 3 cycles pour 20% des instructions
 - $IPC = 1 / (0.8 \times 1 + 0.2 \times 4) = 0.625$ instructions par cycle

27/09/2009

Reconfigurable Computing / MSR -EMI

24

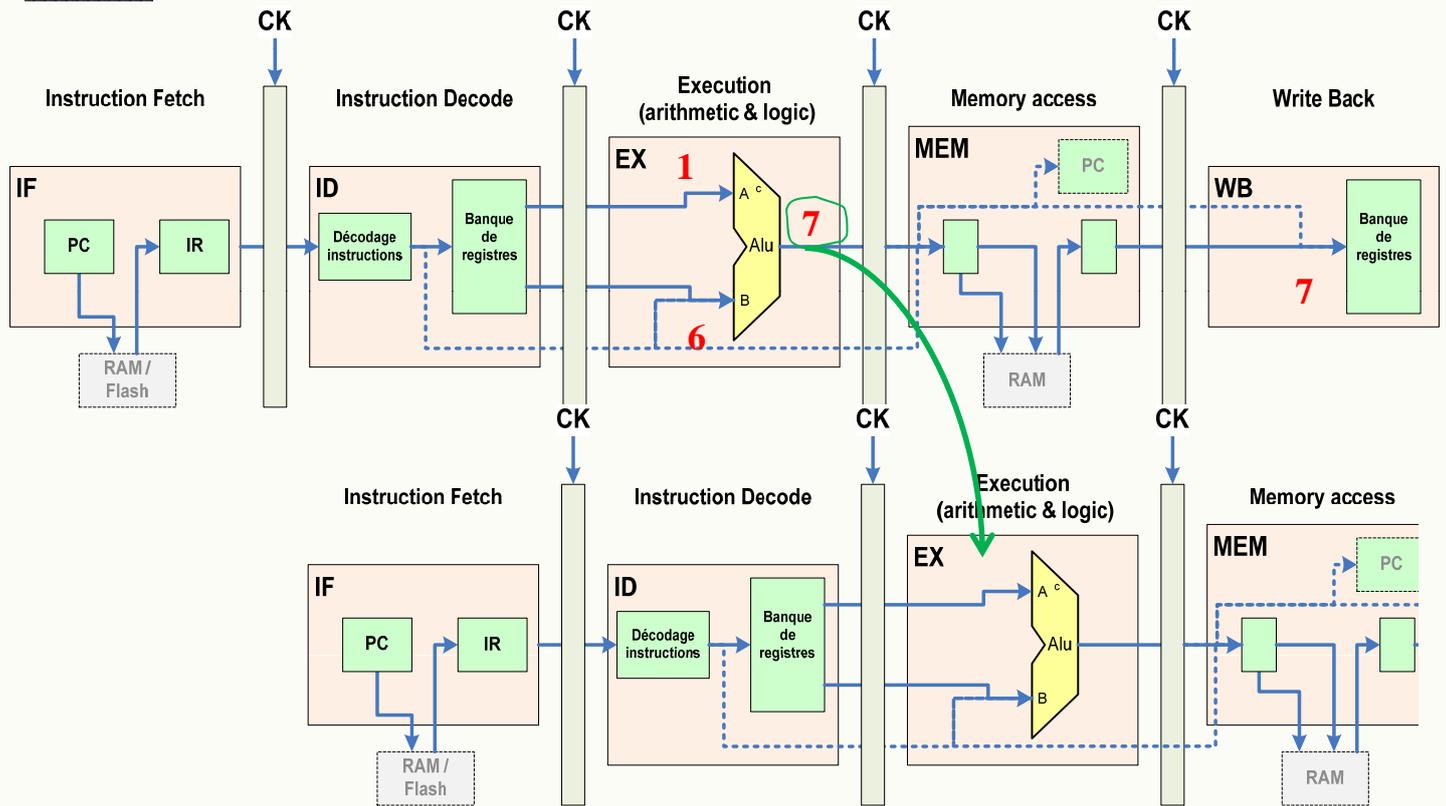
Arrêt de pipeline ou insertion de NOPs

	1	2	3	4	5	6	7	8	9	10
ADD R1, R2, R3	IF	ID	EX	MEM	WB					
SUB R5, R1, #2		IF	ID	ID	ID	ID	EX	MEM	WB	
ADD R1, R2, R3	IF	ID	EX	MEM	WB					
NOP		IF	ID	EX	MEM	WB				
NOP			IF	ID	EX	MEM	WB			
NOP				IF	ID	EX	MEM	WB		
SUB R5, R1, #2					IF	ID	EX	MEM	WB	

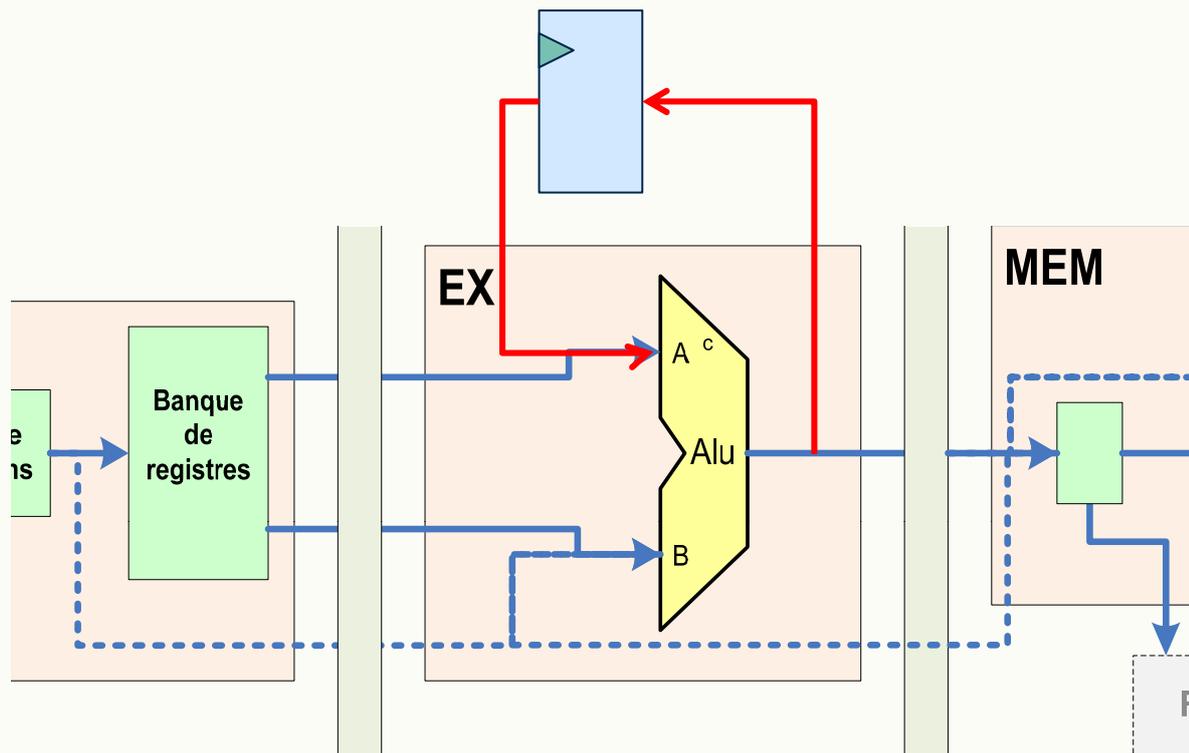
Méthode de résolution des aléas de données

- Forwarding ou bypassing
- Utilisation des résultats à la sortie de l'ALU sans attendre le cycle WB
- Les opérandes sont stockés dans des registres
- Il faut stocker 3 résultats pour un pipeline à 5 étages

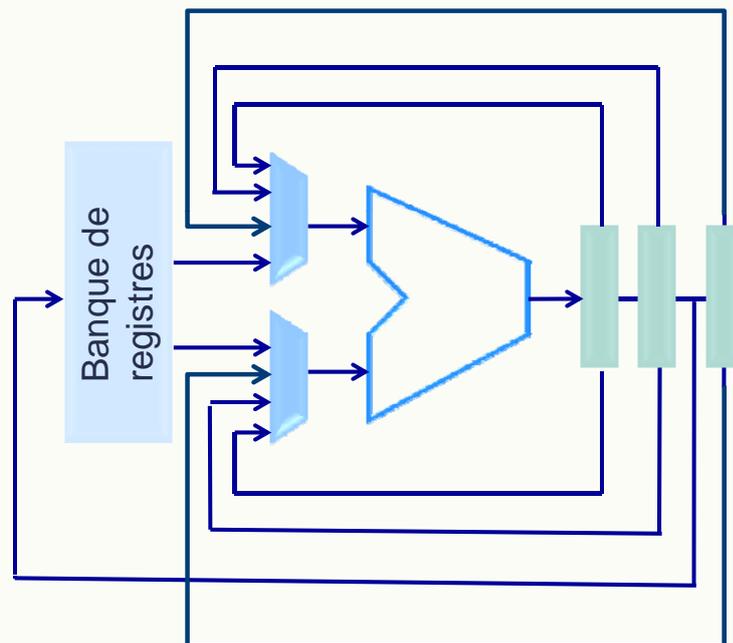
Forwarding (ou bypassing) (1/3)



Forwarding (ou bypassing) (2/3)



Forwarding



Cours ReCo

EXERCICES

Exercice : dépendances et aléas de pipeline

- Dessinez le chronogramme de l'exécution du code avec résolution des aléas par arrêt (hardware) du pipeline. Quel est l'IPC (nombre moyen d'instruction par cycle) pour ces 9 instructions ?

```
1  LDR   R1, [R0]
2  LDR   R2, [R1]
3  ADD   R6, R5, R4
4  ADD   R3, R1, R2
5  LDR   R4, [R6]
6  SUB   R2, R0, R4
7  ADD   R7, R1, #4
8  ADD   R4, R1, R3
9  SUB   R6, R7, R4
```

Eduardo Sanchez
HEIG-VD

Reconfigurable Computing / MSR -EMI

- RAW (read after write):

```
1  LDR   R1, [R0]
2  LDR   R2, [R1]
3  ADD   R6, R5, R4
4  ADD   R3, R1, R2
5  LDR   R4, [R6]
6  SUB   R2, R0, R4
7  ADD   R7, R1, 4
8  ADD   R4, R1, R3
9  SUB   R6, R7, R4
```

Reconfigurable Computing / MSR -EMI

• *WAR(write after read):*

1	LDR	R1, [R0]
2	LDR	R2, [R1]
3	ADD	R6, R5, R4
4	ADD	R3, R1, R2
5	LDR	R4, [R6]
6	SUB	R2, R0, R4
7	ADD	R7, R1, 4
8	ADD	R4, R1, R3
9	SUB	R6, R7, R4

• *WAW(write after write):*

1	LDR	R1, [R0]
2	LDR	R2, [R1]
3	ADD	R6, R5, R4
4	ADD	R3, R1, R2
5	LDR	R4, [R6]
6	SUB	R2, R0, R4
7	ADD	R7, R1, 4
8	ADD	R4, R1, R3
9	SUB	R6, R7, R4

