

[illegible]

Exercice 2 :

1. LW r3, 0(r2) 2. ADD r5, r3, r4 3. SUB r6, r5, r7 4. OR r8, r6, r9 5. SW r8, 4(r2)	En utilisant un pipeline à 5 étages (Fetch, Decode, Read Register, Execute, Write-back) : 1. Identifiez tous les alés de données et indiquez leur type : ○ Lecture Après Écriture (LAE) ○ Écriture Après Écriture (EAE) ○ Lecture Après Lecture (LAR) 2. Tracez le diagramme d'exécution 3. Proposez des solutions pour résoudre ou minimiser les alés.
---	--

Réponse :

1. Aléas de données

[illegible]

2. Diagramme d'exécution

[illegible]

3. Proposez des solutions pour résoudre ou minimiser les aléas.

[illegible]

Corrigé Micro Interrogation

Exercice 1 :

Architecture accumulateur (Variables temporaires: T ₁ , T ₂ , T ₃)	Architecture Pile	Architecture Registre-Registre (Registres de R ₀ à R ₉)
LOAD A # ACC = A ADD B # ACC = ACC + B STORE T1 # T1 = A + B LOAD C # ACC = C SUB D # ACC = ACC - D DIV E # ACC = ACC / E MUL T1 # ACC = ACC × T1 STORE F # F = ACC	PUSH A #Pile : [A] PUSH B #Pile : [B, A] ADD #Pile : [A + B] PUSH C #Pile: [C, A + B] PUSH D #Pile: [D, C, A+B] SUB #Pile:[C-D, A+B] PUSH E #Pile:[E, C-D, A+B] DIV #Pile:[(C-D)/E, A+B] MUL #Pile:[(A+B) × ((C-D)/E)] POP F # F = Pile[Sommet]	LOAD R1, A # R1 = A LOAD R2, B # R2 = B ADD R3, R1, R2 #R3=R1+R2 LOAD R4, C # R4 = C LOAD R5, D # R5 = D SUB R6, R4, R5 #R6=R4-R5 LOAD R7, E # R7 = E DIV R8, R6, R7 #R8=R6/R7 MUL R9, R3, R8 #R9=R3×R8 STORE F, R9 # F = R9

Exercice 2 :

1. Aléas de données

Lecture Après Écriture (LAE)	Écriture Après Écriture (EAE)	Lecture Après Lecture (LAL)
○ ADD - LW (r3) ○ SUB - ADD (r5) ○ OR - SUB (r6)	SW - OR (r8)	 Aucun aléa de ce type....

2. Diagramme d'exécution

Fetch (F) : Lecture de l'instruction.

Decode (D) : Décodage de l'instruction et calcul des adresses.

Read Register (RR) : Lecture des registres nécessaires.

Execute (E) : Exécution de l'opération (ALU ou calcul).

Write-back (W) : Écriture du résultat dans le registre.

	Cycles																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F	lw	add	sub	or	sw															
D		lw	add	sub	or	sw														
R			lw	-	-	add	-	-	sub	-	-	or	sw							
E				lw	-	-	add	-	-	sub	-	-	or	sw						
W					lw	-	-	add	-	-	sub	-	-	or	sw					

3. Proposez des solutions pour résoudre ou minimiser les aléas.

- **Insertion de NOP** : Ajouter des instructions nop pour résoudre les dépendances :
- **Forwarding (si supporté par l'architecture)** : Transmettre directement les résultats entre les étages pour éviter les cycles inutiles.
- **Réorganisation des instructions** : Si possible, déplacer les instructions indépendantes pour maximiser l'utilisation du pipeline.