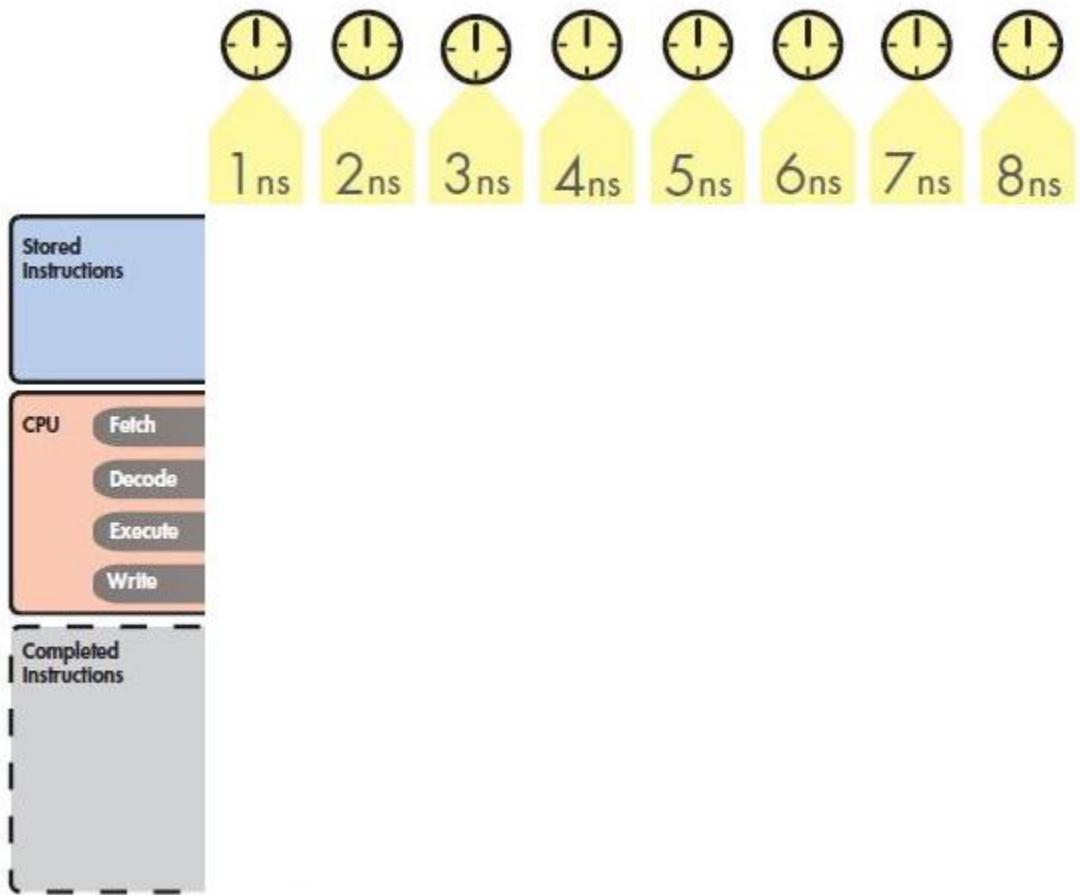


HENALLUX	TECHNOLOGIE DE L'INFORMATIQUE ARCHITECTURE DES ORDINATEURS	NOM : Gr. :	Date :	
----------	---	----------------	--------	--

1. Complétez le schéma de ce *pipeline* pour un ensemble de 4 instructions, chaque étage du *pipeline* prenant 1 ns (la partie *Stored Instructions* n'est pas prise en compte pour la correction).



- Donnez le *completion rate* théorique maximum de ce *pipeline*.
- Donnez le *completion rate* réel moyen de ce *pipeline* après 20ns si le celui-ci n'était pas préalablement chargé.
- Donnez le *program execution time* d'un programme comprenant 15 instructions (sur base du *completion rate* théorique calculé ci-dessus).

2. Illustrez graphique le phénomène de décrochage du *pipeline*.

Citez une des causes principales de ces décrochages.

3. Citez les 2 limites du *pipeline* (pourquoi ne pas avoir des profondeurs énormes ?). Expliquez brièvement.

4. Expliquez les commandes suivantes (soyez précis, utilisez les termes corrects) :

- Load #13,D
- Load #C, B
- Store B, #(D+17)
- Add B, C, C

5. Soit le mapping et le format d'instructions suivant :

Mnemonic	Opcode
add	000
sub	001
load	010
store	011

Register	Binary Code
A	00
B	01
C	10
D	11

0	1	2	3	4	5	6	7
mode	opcode			source		destination	

Byte 1

8	9	10	11	12	13	14	15
8-bit immediate value							

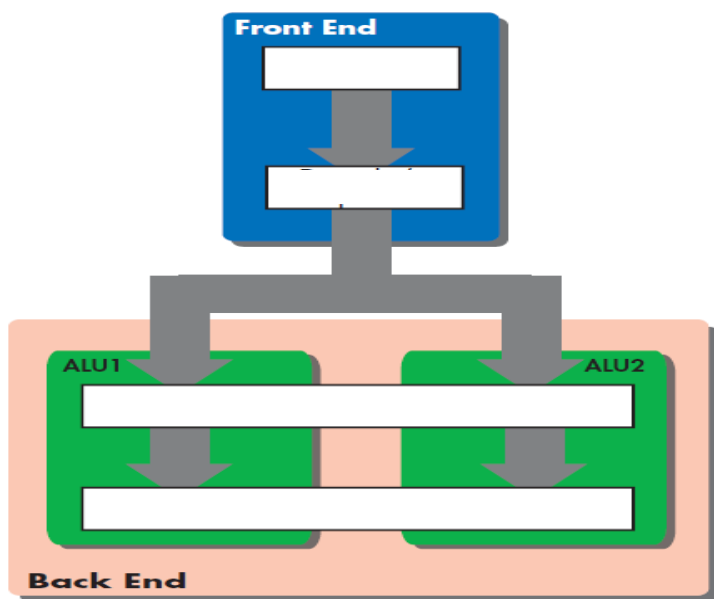
Byte 2

- Traduire en langage machine l'instruction suivante : sub C, A, A
- Traduire en langage assembleur l'instruction suivante : 1000110100110101

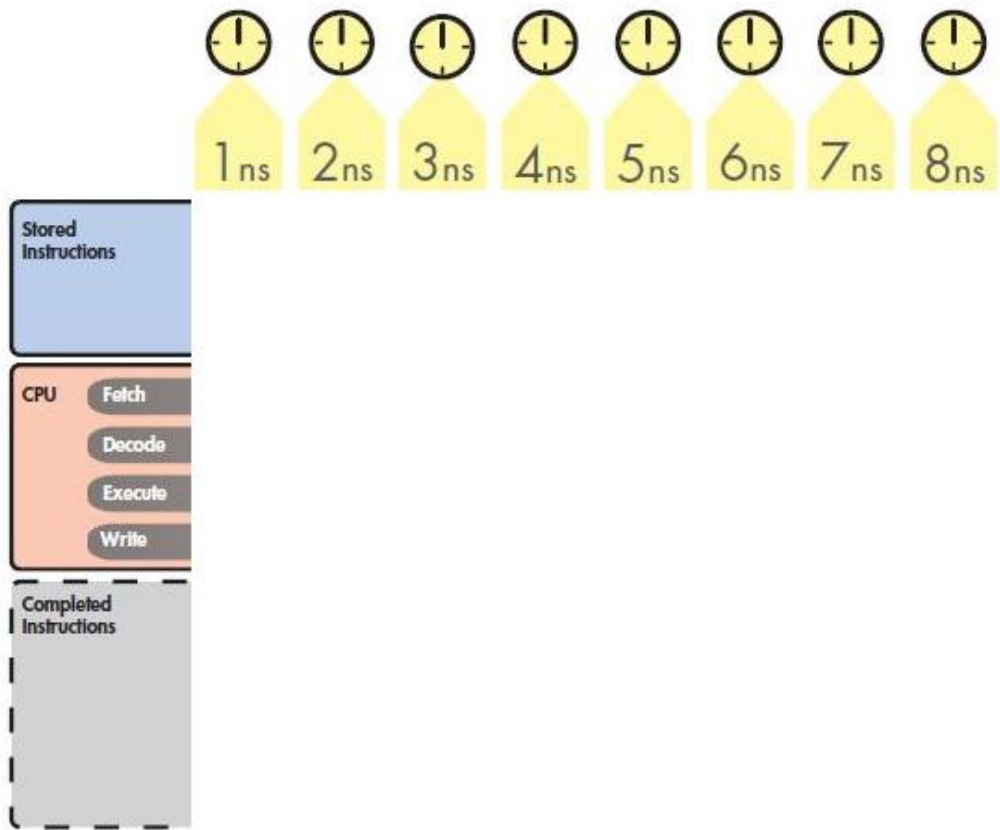
6. Définir en quelques mots:

- Un registre,
- Le *program counter*,
- Le *programming model*,
- Un bus,
- l'instruction register*,
- Le PSW,
- L'assembleur.

7. Complétez les 4 cases du schéma d'architecture superscalaire ci-dessous



8. Complétez le schéma de ce *pipeline superscalaire* pour un ensemble de 8 instructions, chaque étage de *pipeline* prenant 1 ns, 2 ALUs en parallèle (la partie *Stored Instructions* n'est pas prise en compte pour la correction).



- Donnez le *completion rate* théorique maximum de cette architecture.
- Donnez le *completion rate* réel moyen de ce *pipeline* après 20ns si le celui-ci n'était pas préalablement chargé.
- Donnez le *program execution time* d'un programme comprenant 15 instructions (sur base du *completion rate* théorique calculé ci-dessus)

9. Expliquez le problème lié aux dépendances des données dans l'architecture superscalaire. Donner un exemple d'instructions problématiques. Expliquez brièvement comment le *Forwarding* peut solutionner ce problème.

#### Pondération des questions

- Q1: /10
  - Schéma: 4 points (0,25/colonne)
  - Valeurs clés 6 points (3x2)
- Q2: /5
  - Schéma: 2 points
  - Cause: 3 points
- Q3: /5
  - Deux limites: 2 points
  - Explication: 3 points
- Q4: /8 (4x2)
- Q5: /4 (2x2)
- Q6: /14 (7x2)
- Q7: /4 (4x1)
- Q8: /10 (Idem Q1)
- Q9: /5
  - Explication du problème (2,5 points)
  - Solution (2,5 points)

Total: /65 (20 points pour le pipeline, 26 pour les généralités et 19 pour le superscalaire).