Architecture des ordinateurs II

Les architectures RISC et CISC (1)

Actuellement l'architecture des microprocesseurs se composent de deux grandes familles :

- L' architecture CISC

  (Complex Instruction Set Computer)

  (Ordinateur à jeu d'instructions complexe)

- L'architecture RISC

  (Reduced Instruction Set Computer)

  (Ordinateur à jeu d'instructions réduit)

Le traitement des instructions

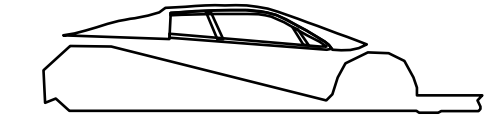
Les architectures RISC et CISC (2)

Architecture RISC	Architecture CISC
instructions simples ne prenant qu'un seul cycle	instructions complexes prenant plusieurs cycles
instructions au format fixe	instructions au format variable
décodeur simple (câblé)	décodeur complexe (microcode)
beaucoup de registres	peu de registres
peu de modes d'adressage	beaucoup de modes d'adressage
compilateur complexe	compilateur simple

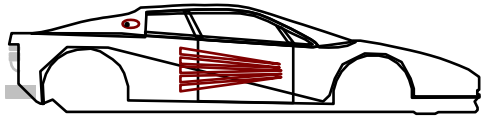
Chapitre 2 : Le pipeline

- 2.1 Définition d'un pipeline
- 2.2 Les étages d'un pipeline
- 2.3 Les aléas dans le pipeline

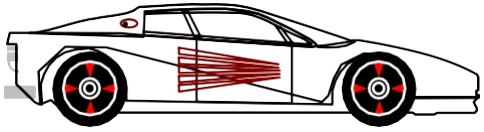
Comparaison (1)



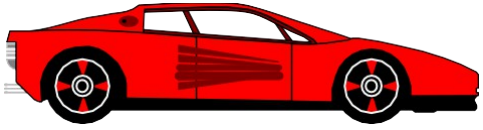
1ère étape de conception



2ème étape de conception 3ème étape



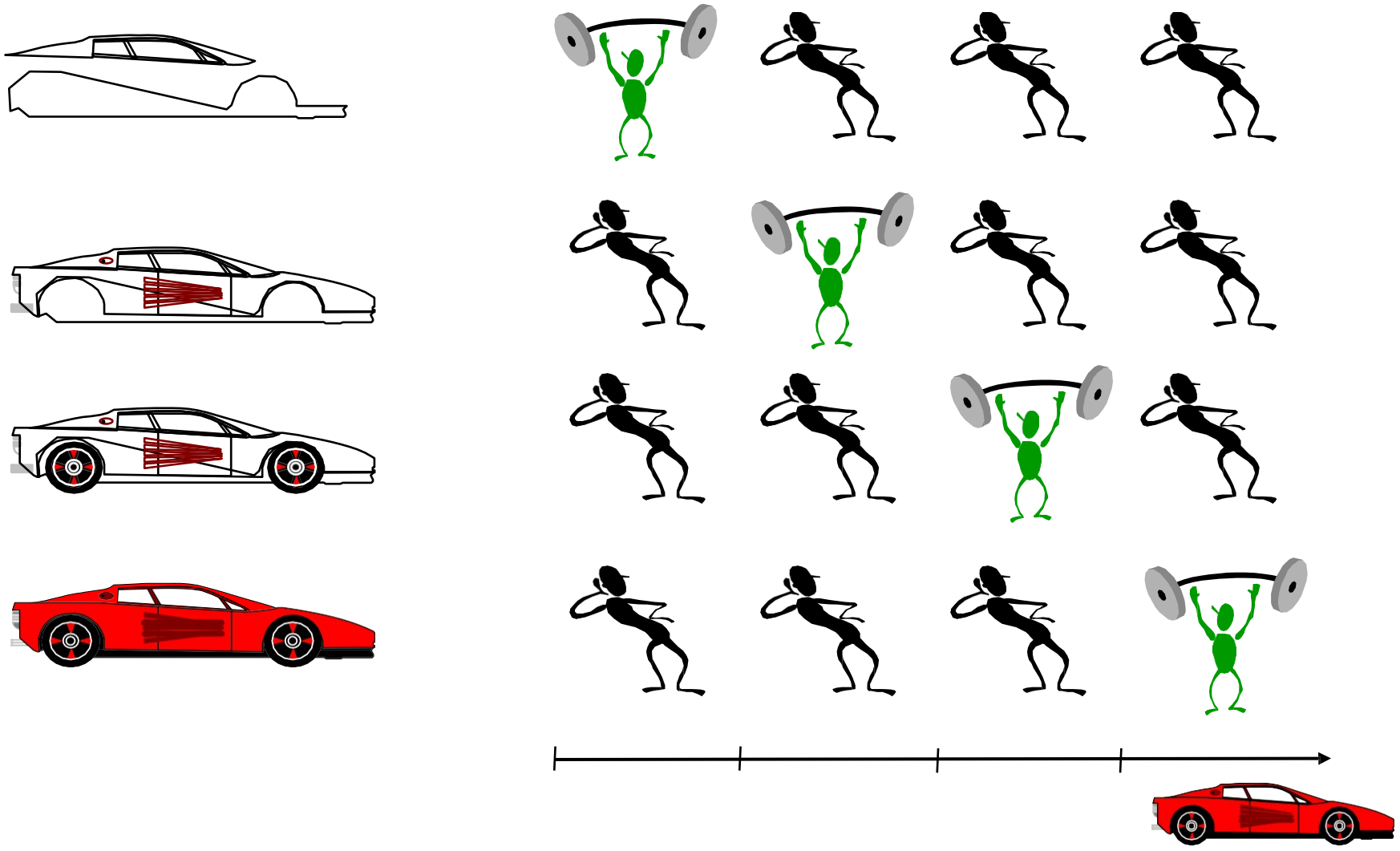
de conception



4ème étape de conception

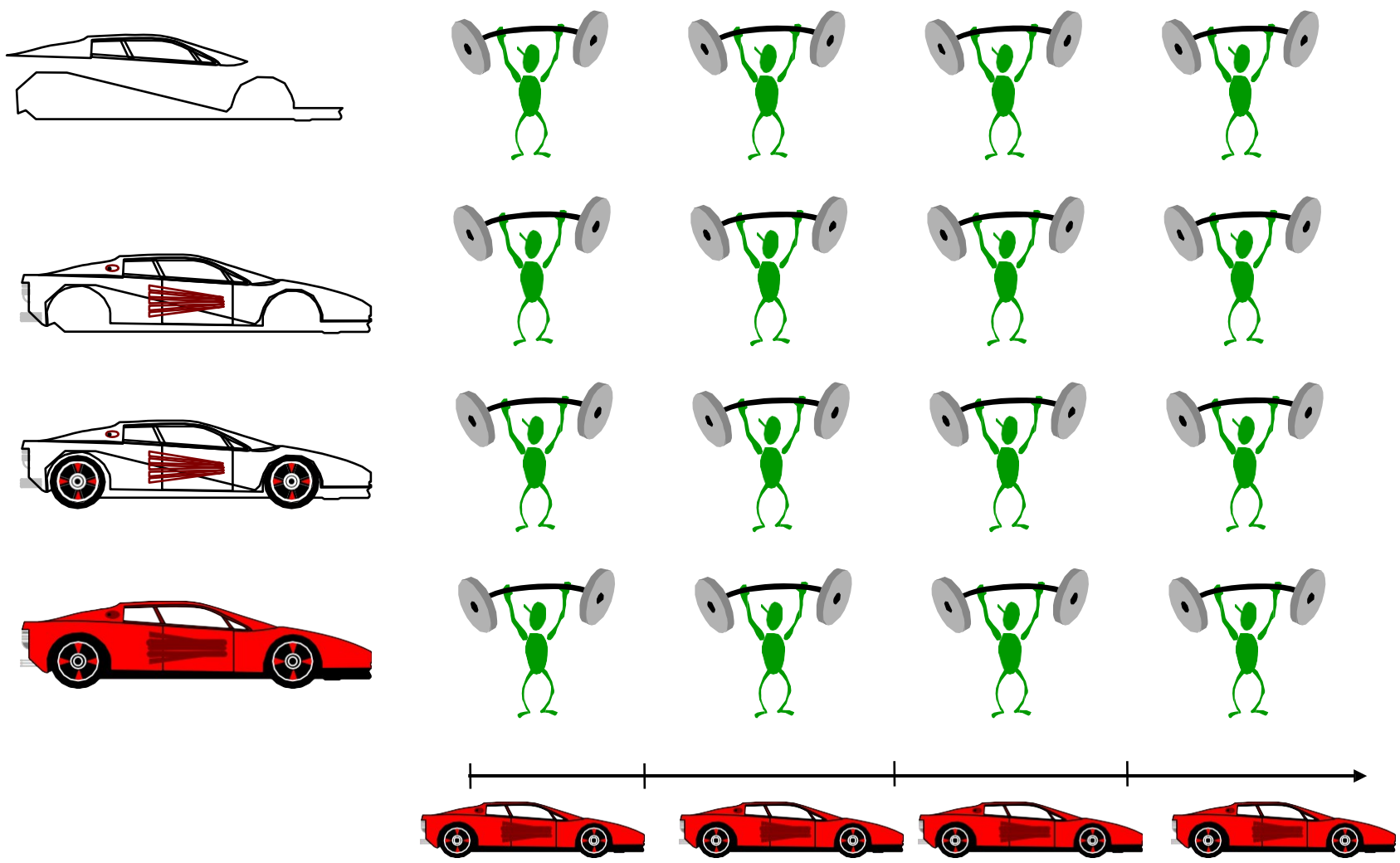
Définition d'un pipeline

Comparaison (2)



Définition d'un pipeline

Comparaison (3)



Définition d'un pipeline

- La technique du pipeline est une technique de mise en oeuvre qui permet à plusieurs instructions de se chevaucher pendant l'exécution.

- Une instruction est découpée dans un pipeline en petits morceaux appelés étage de pipeline.

- La technique du pipeline améliore le débit des instructions plutôt que le temps d'exécution de chaque instruction.

- La technique du pipeline exploite le parallélisme entre instructions d'un flot séquentiel d'instructions. Elle présente l'avantage de pouvoir, contrairement à d'autres techniques d'accélération, être rendue invisible du programmeur.

Chapitre 2 : Le pipeline

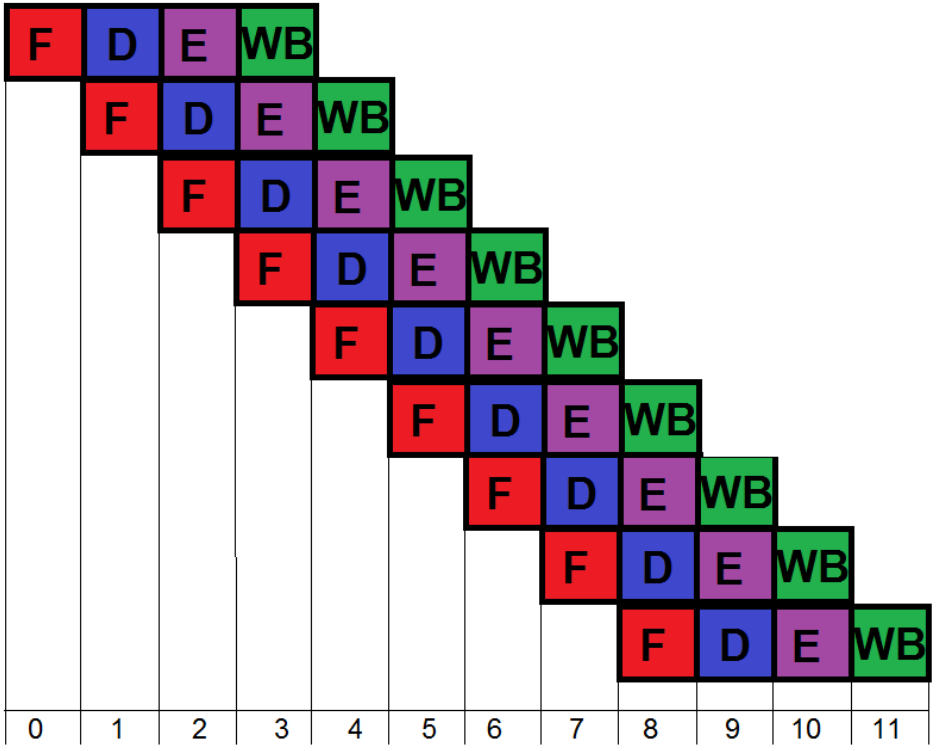
- 2.1 Définition d'un pipeline
- 2.2 Les étages d'un pipeline
- 2.3 Les aléas dans le pipeline

Les étages d’un pipeline

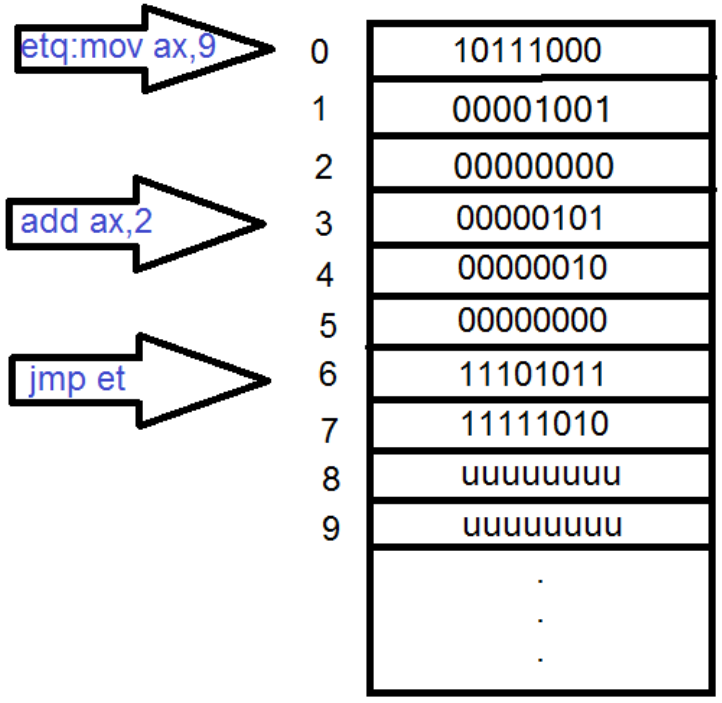
F (Fetch) Lit l'instruction à exécuter de la memoire.
D (Decode) décode l'instruction.
E (Execute) exécute l'instruction (par l'UAL).
WB (Write Back) stocke le résultat dans un registre ou en mémoire.

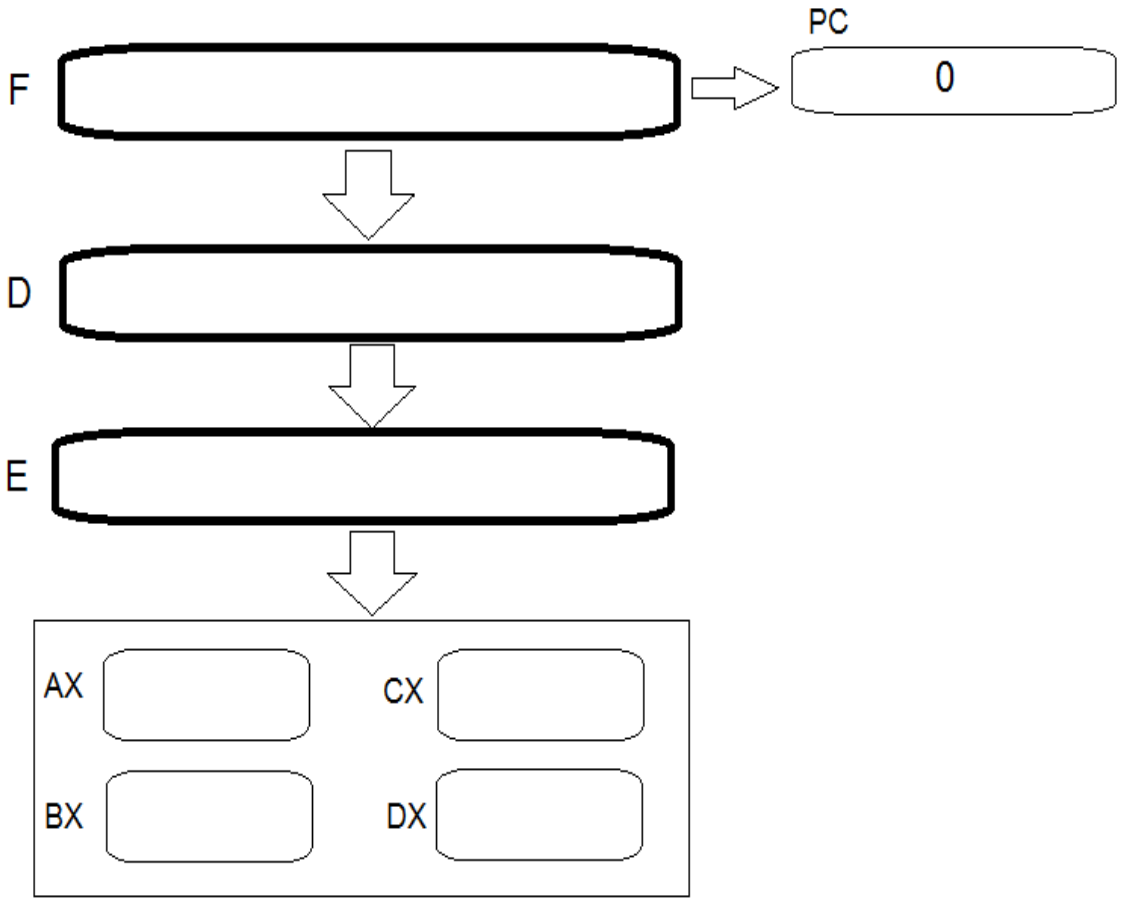
Les étages d’un pipeline

Clock	Execution
0	• Four instructions are waiting to be executed
1	• The green instruction is fetched from memory
2	• The green instruction is decoded • The purple instruction is fetched from memory
3	• The green instruction is executed (actual operation is performed) • The purple instruction is decoded • The blue instruction is fetched
4	• The green instruction's results are written back to the register file or memory • The purple instruction is executed • The blue instruction is decoded • The red instruction is fetched
5	• The execution of green instruction is completed • The purple instruction is written back • The blue instruction is executed • The red instruction is decoded
6	• The execution of purple instruction is completed • The blue instruction is written back • The red instruction is executed
7	• The execution of blue instruction is completed • The red instruction is written back
8	• The execution of red instruction is completed
9	• The execution of all four instructions is completed

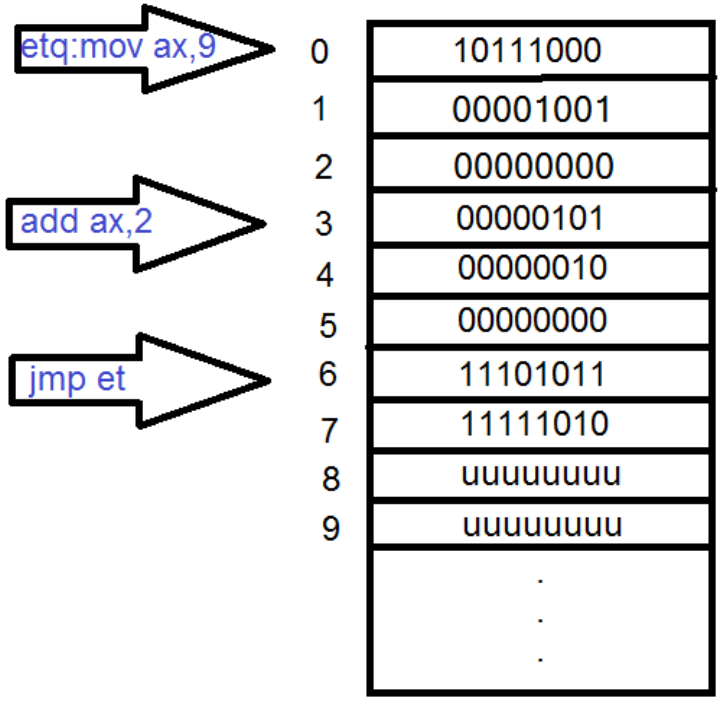


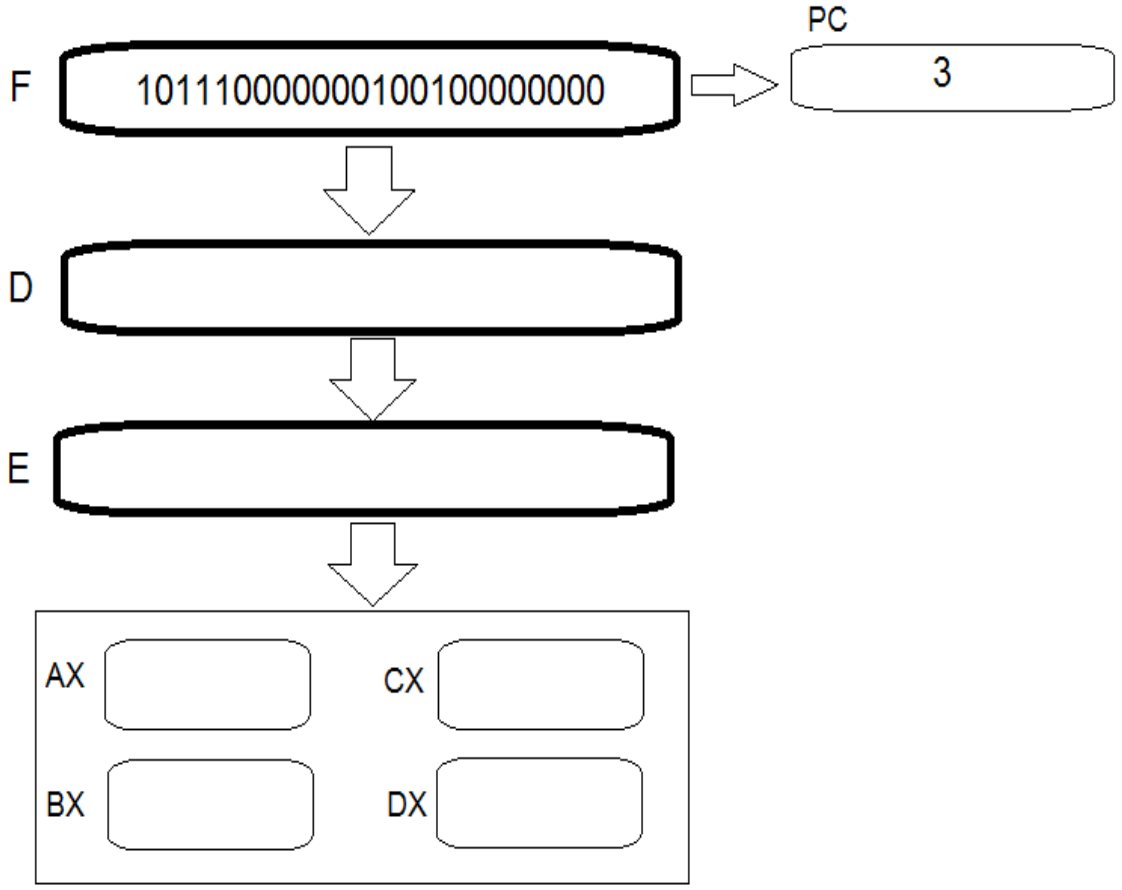
Les étages d’un pipeline



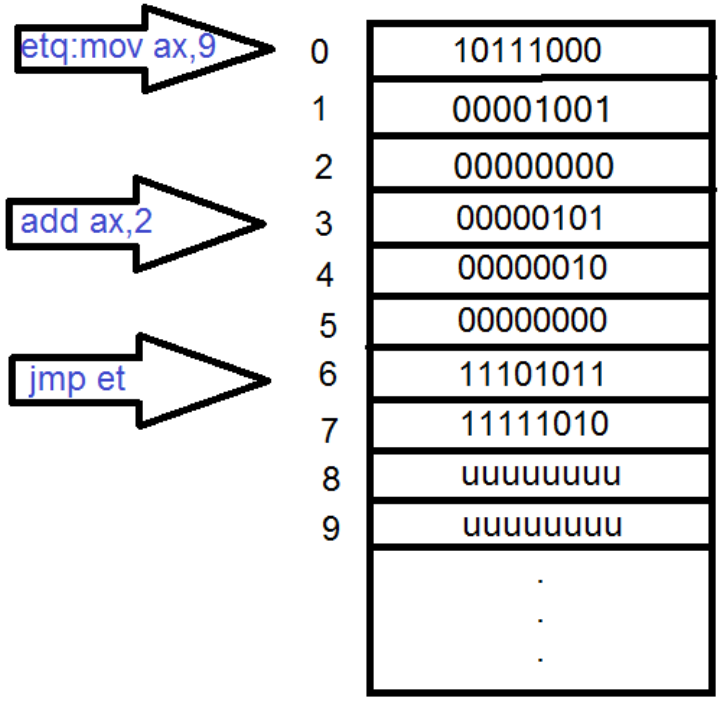


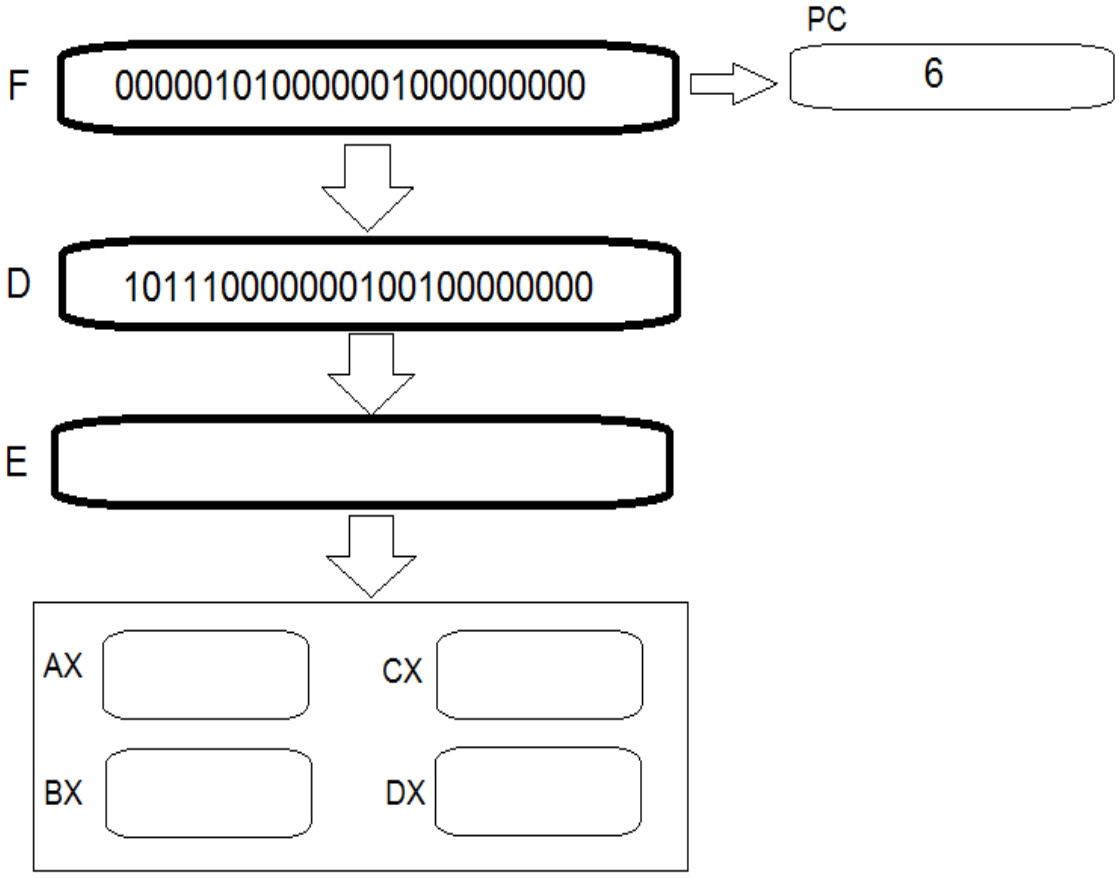
Les étages d’un pipeline



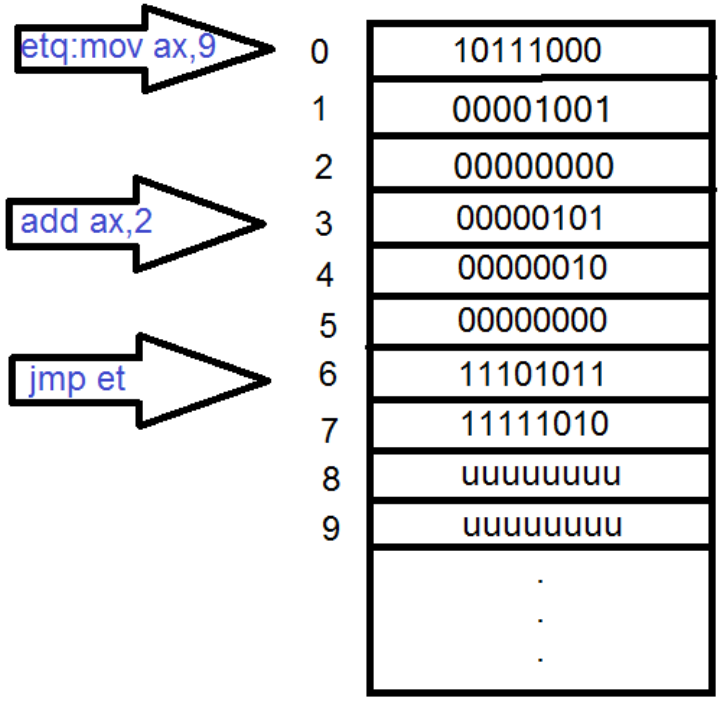


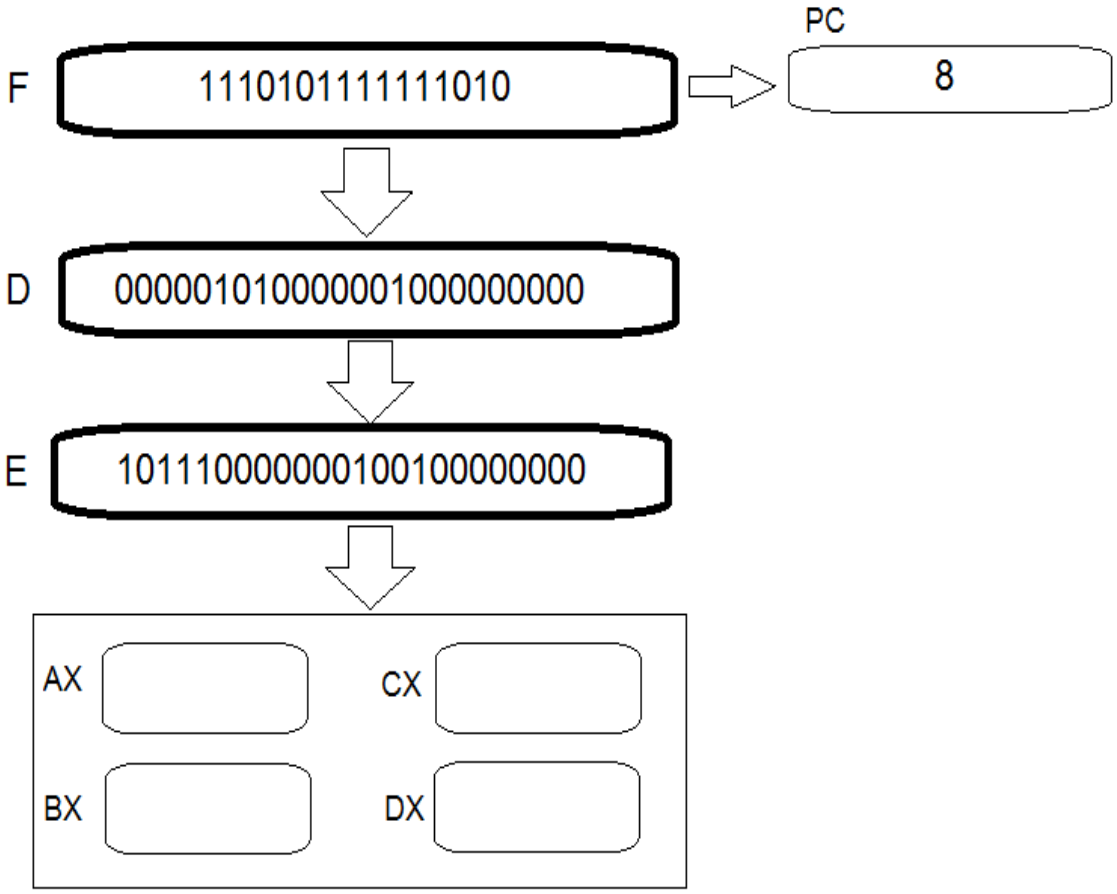
Les étages d’un pipeline



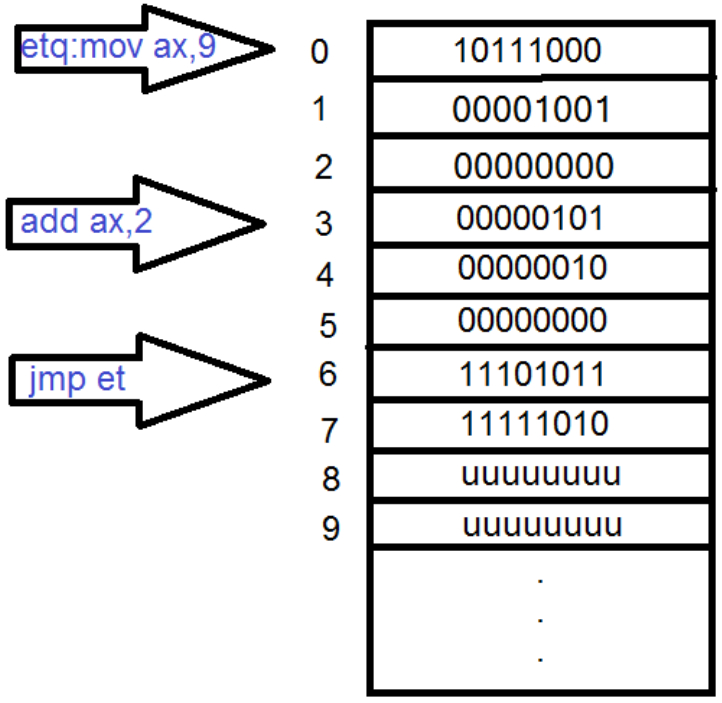


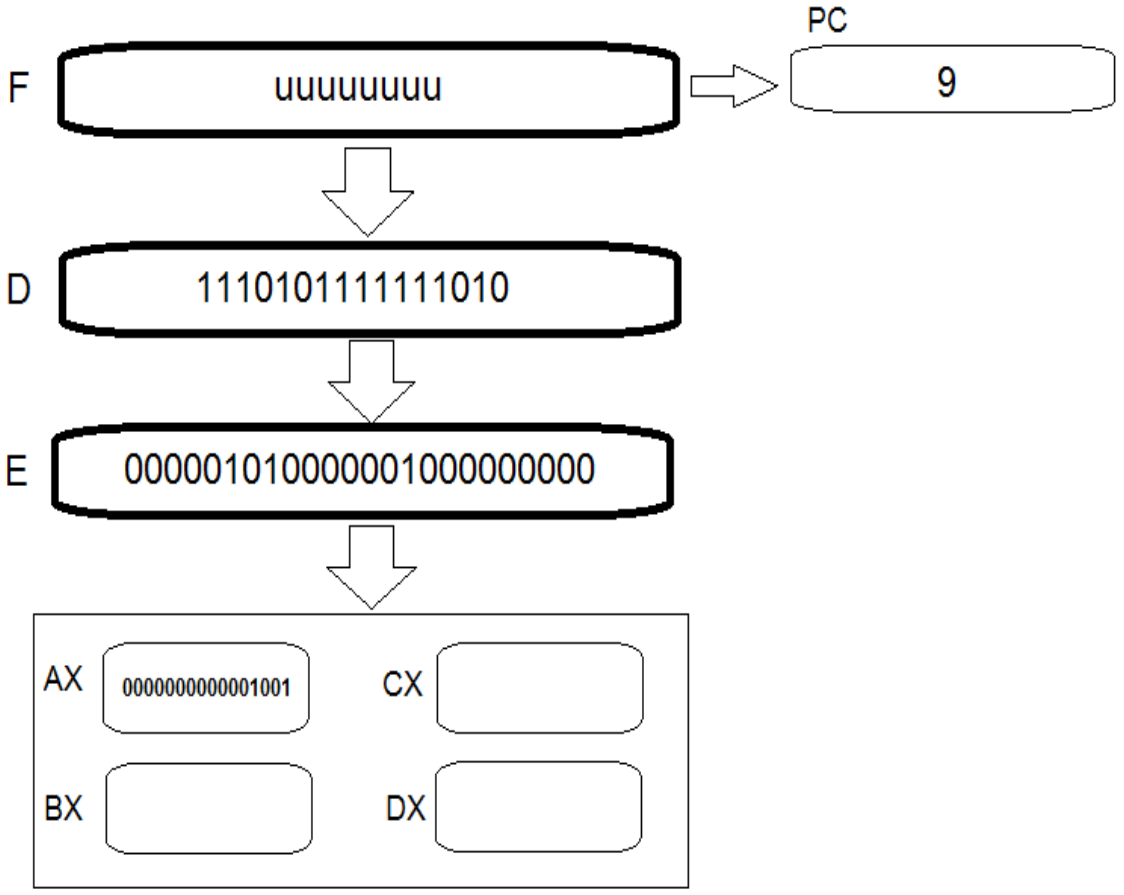
Les étages d’un pipeline



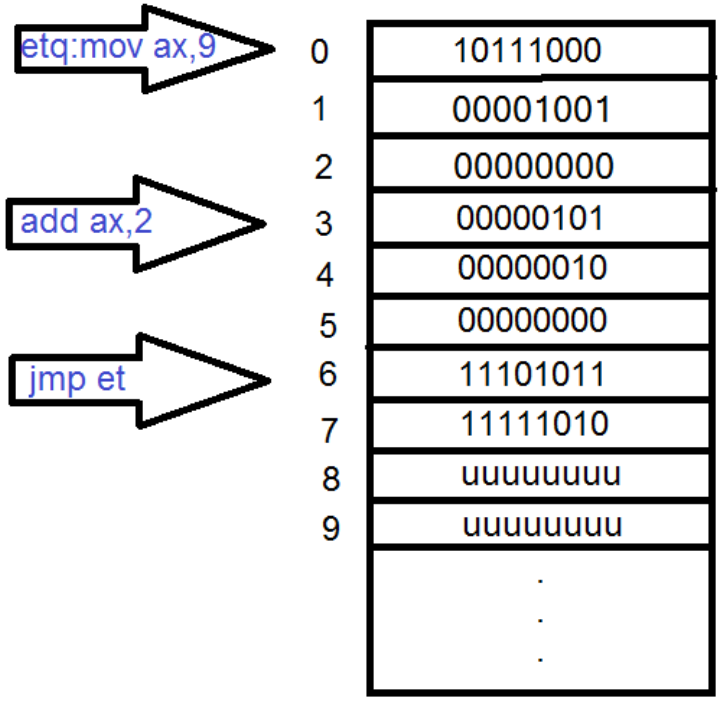


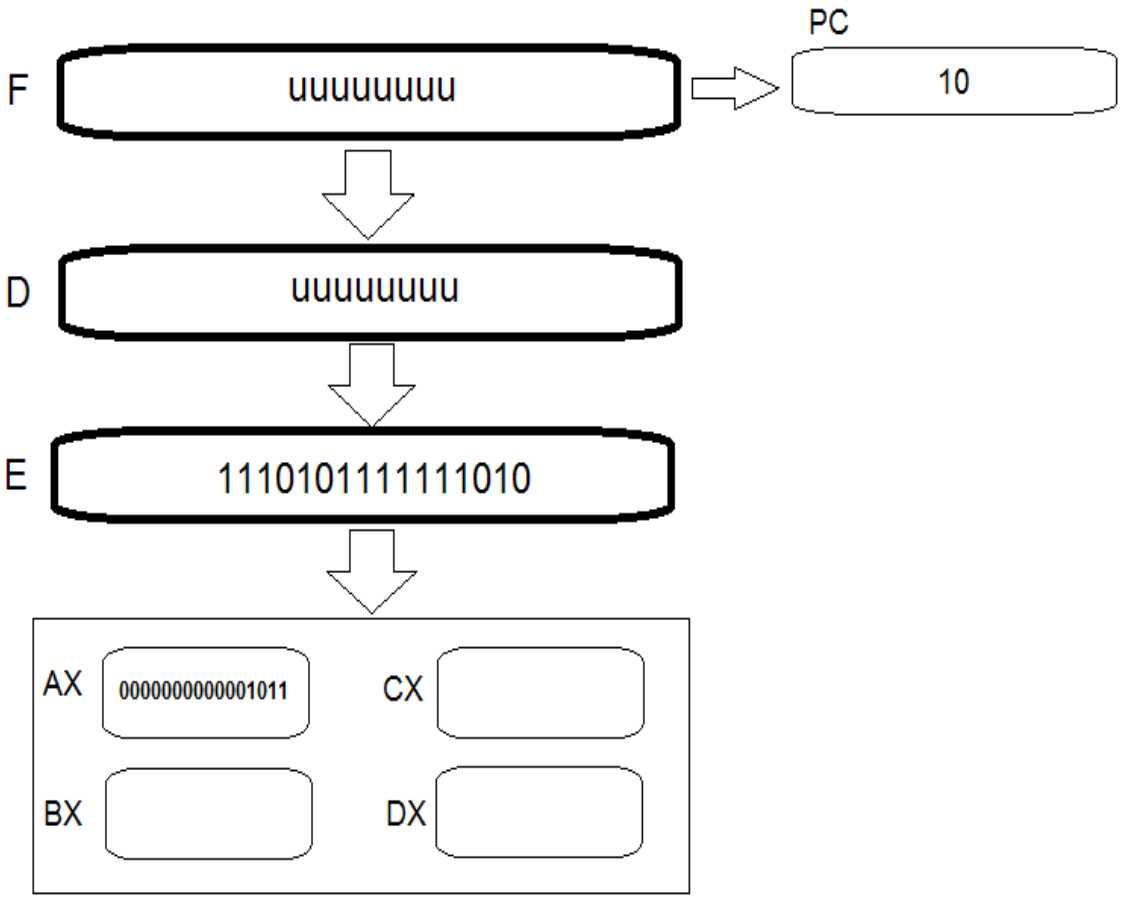
Les étages d’un pipeline



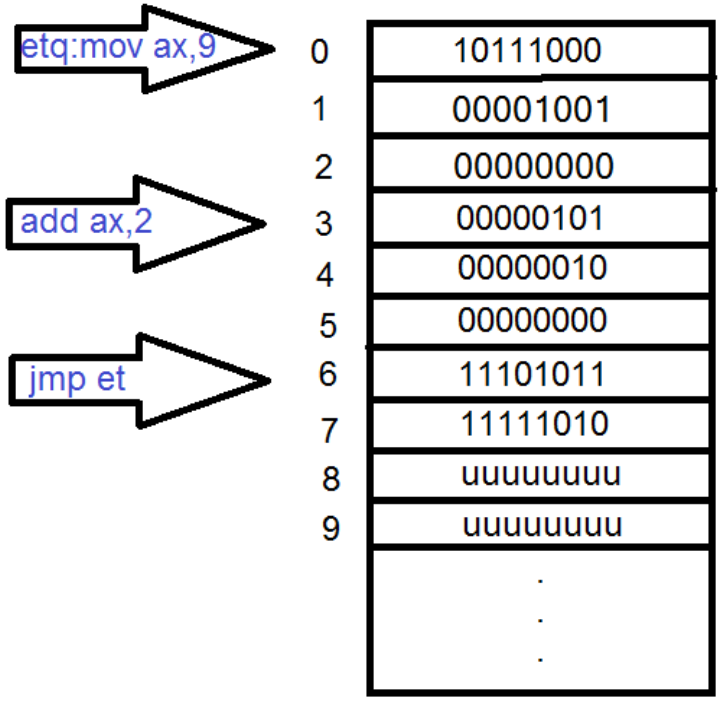


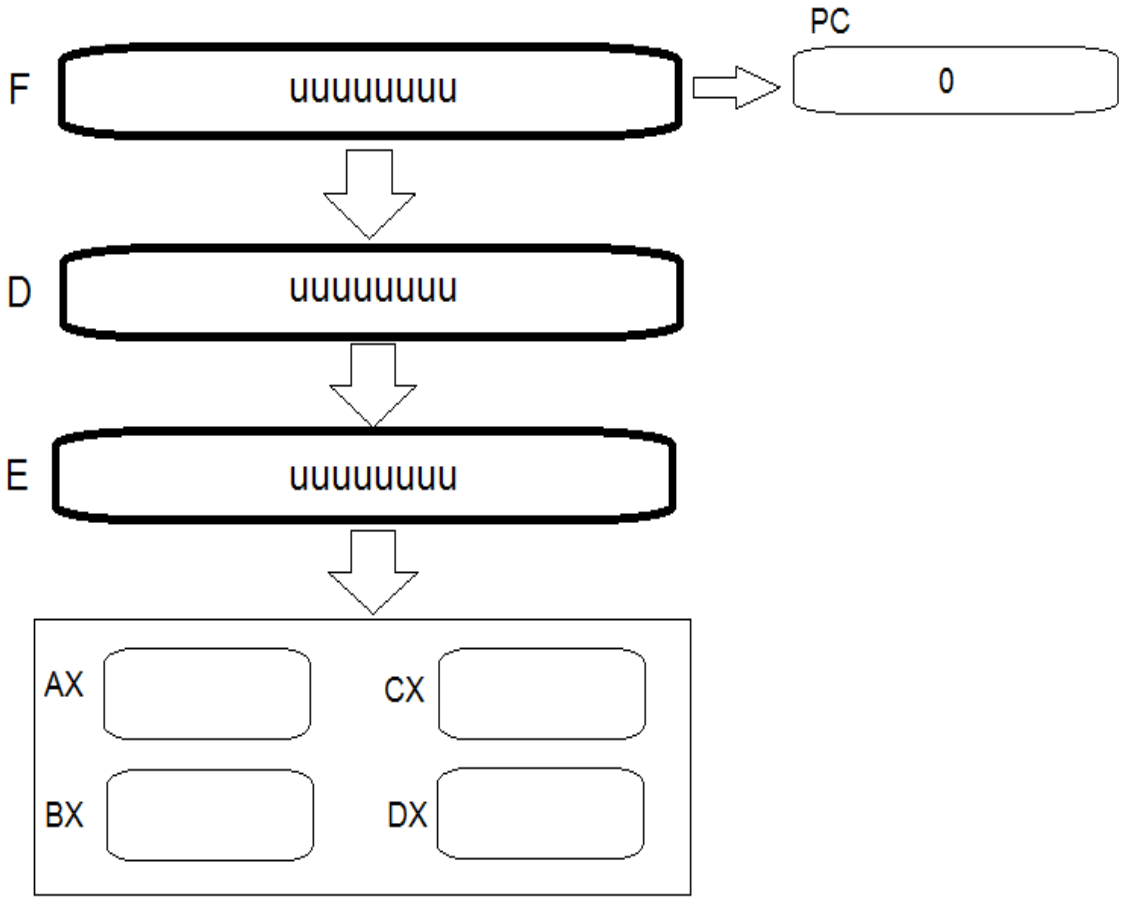
Les étages d’un pipeline





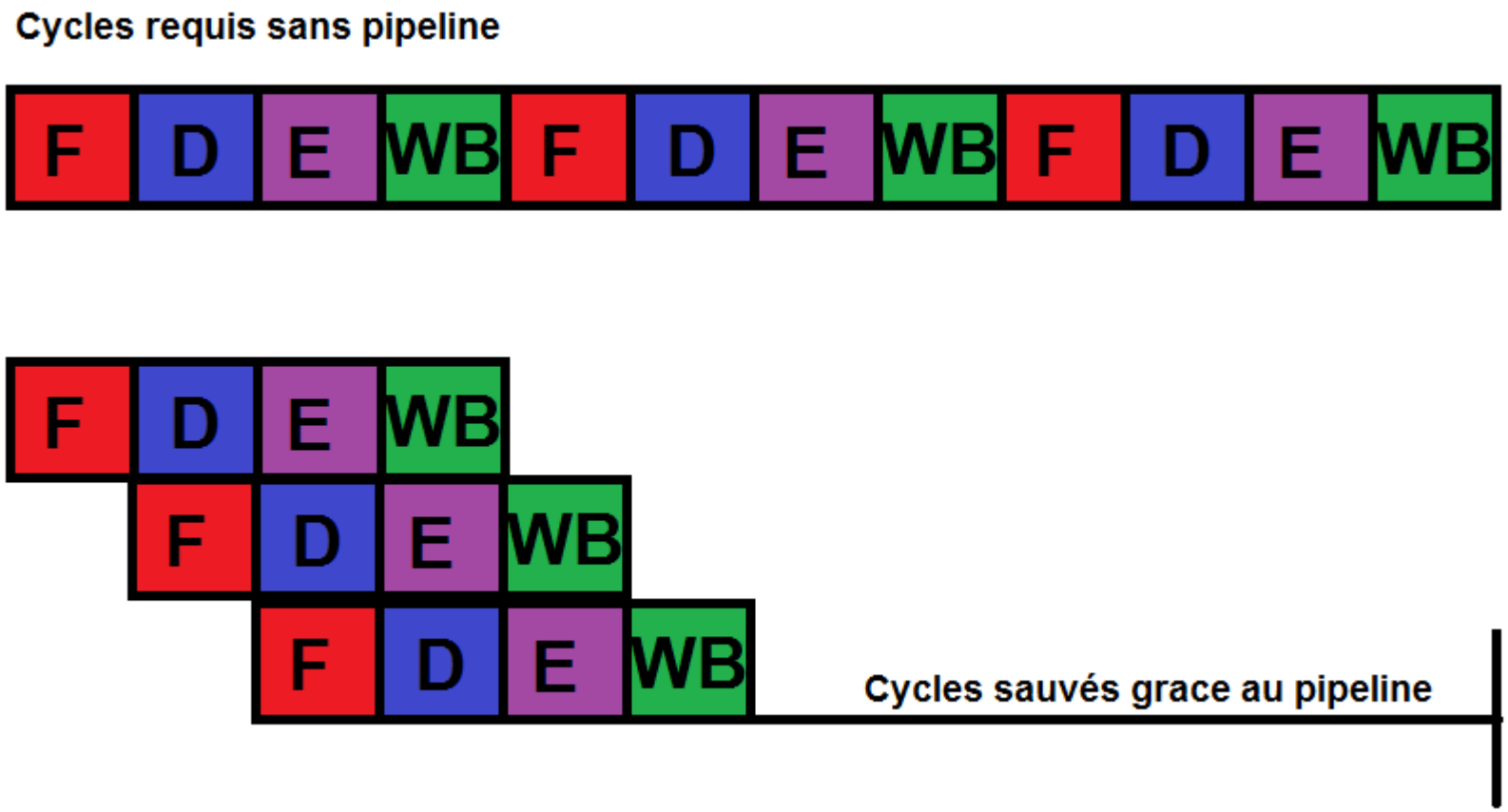
Les étages d’un pipeline





Les étages d'un pipeline

Comparaison avec et sans pipeline



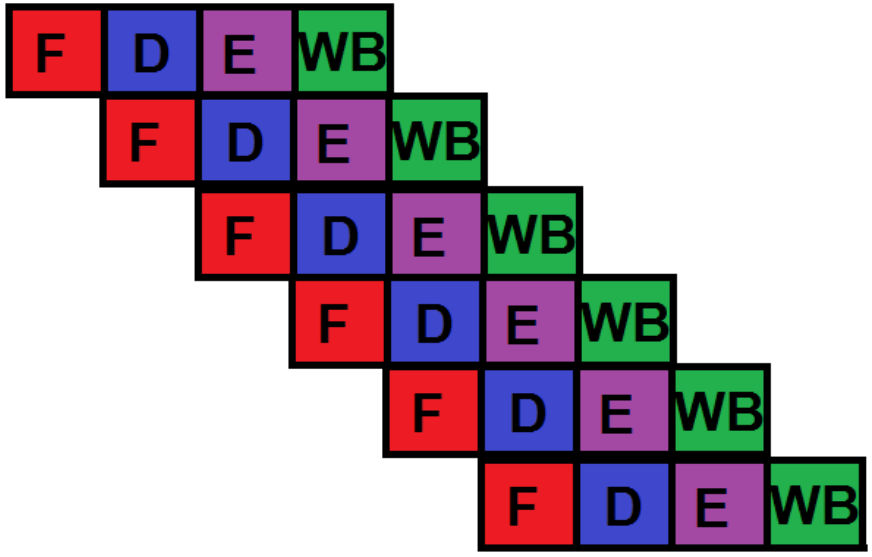
- Moins de cycles par instruction
- Consommation réduite

Les retards

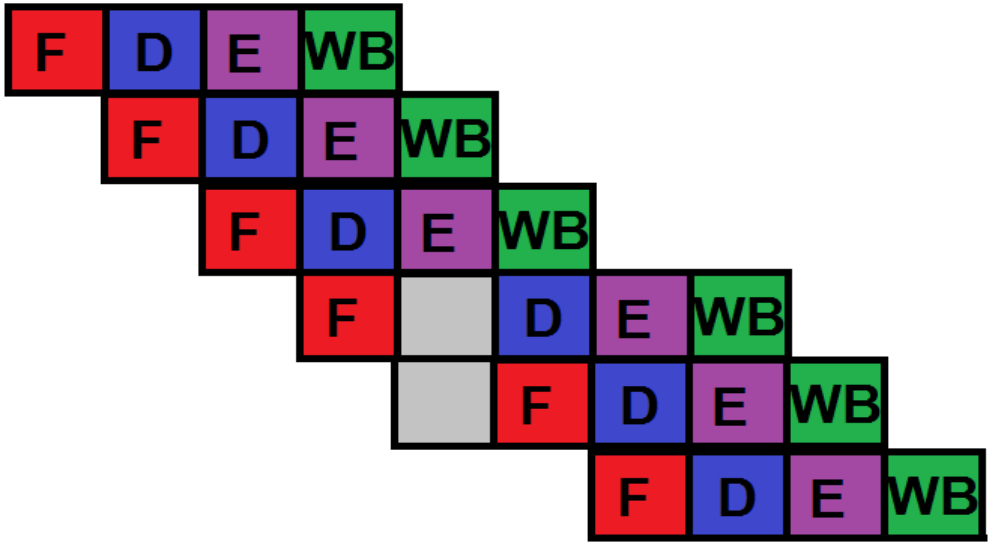
- Le pipeline atteint son plein rendement une fois qu'il est "rempli"
- Un retard peut se produire
  - S'il existe un conflit de ressources (retard ponctuel)
    - accès à la mémoire
    - utilisation des bus
  - En cas de rupture de séquence (vidange du pipeline)
    - branchement non prévu
    - appel de sous-programme
    - interruption

Les étages d'un pipeline

Exemple de rupture



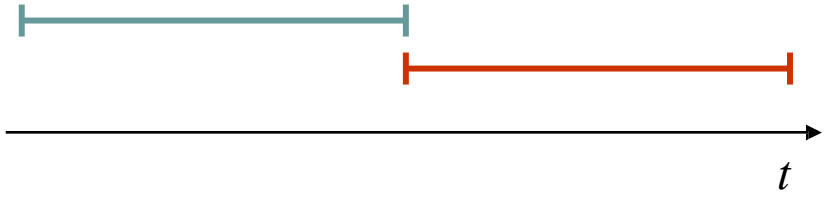
Sans rupture



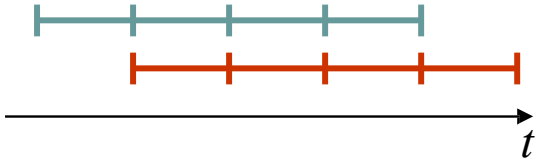
Rupture

Types de pipelining

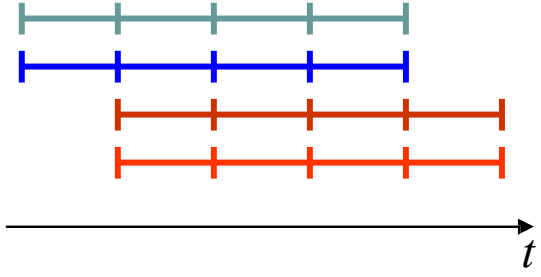
Séquentiel :
Pas de pipeline
(ex: Motorola 56000)



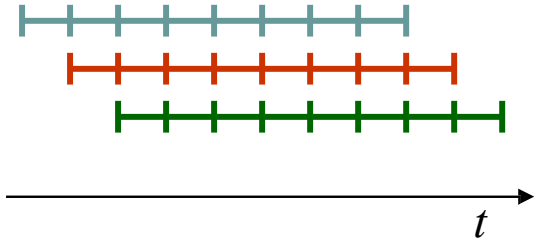
Pipeline simple
(plupart des DSP)



Double pipeline
(ex: Pentium)



Superpipeliné :
Nombre d'étages plus élevé
(ex: TMS320C6000)



Remarques sur les performances

Certaines phases sont inutiles pour certaines instructions (p.ex. un LOAD ne nécessite pas d'exécution), mais toutes les instructions doivent traverser tout le pipeline. Ce "gaspillage" est nécessaire pour simplifier le contrôle.

Exemple de profondeur

Processeur	***Profondeur du pipeline***
Intel Pentium 4 Prescott	31
Intel Pentium 4	20
AMD K10	16
Intel Pentium III	10
AMD Athlon	12
PowerPC G4 (PPC 7450)	7
IBM POWER5	16
IBM PowerPC 970	16
Intel Itanium	10

Chapitre 2 : Le pipeline

- 2.1 Définition d'un pipeline
- 2.2 Les étages d'un pipeline
- 2.3 Les aléas dans le pipeline

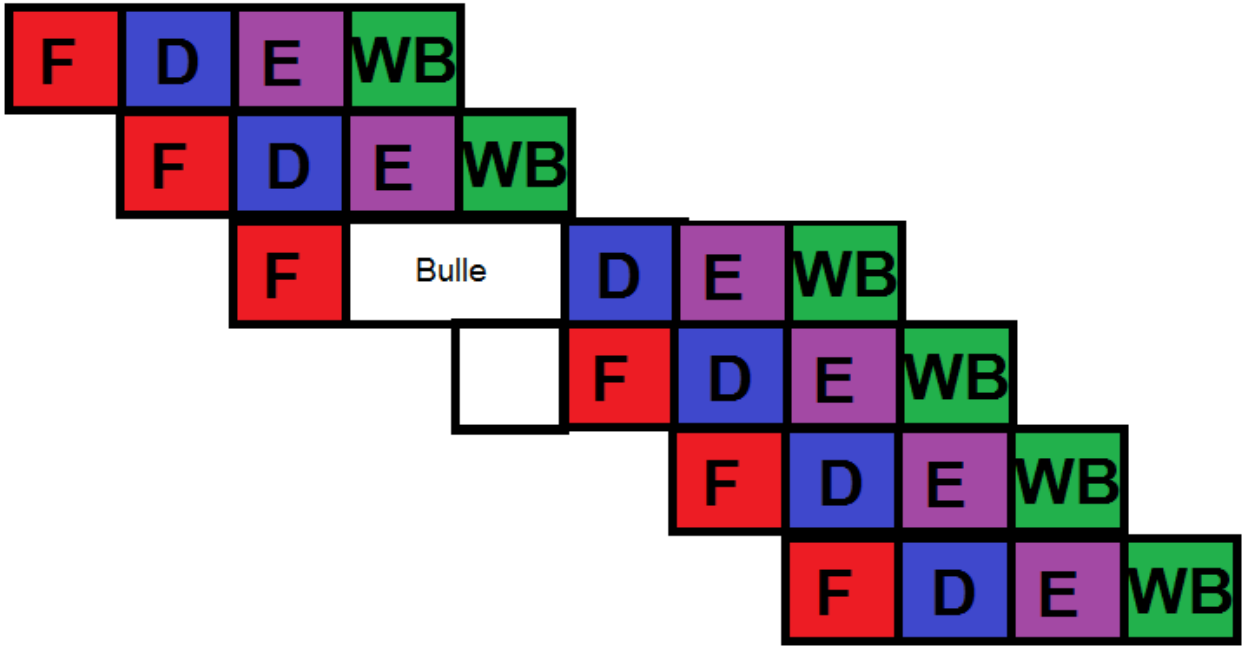
Aléas d'un pipeline

Les types d'aléas

- La présence d'un pipeline (et donc le partage de l'exécution d'une instruction en plusieurs étages) introduit des aléas :
  - Aléas de structure : L'implémentation empêche une certaine combinaison d'opérations (lorsque des ressources matériels sont accédées par plusieurs étages).
  - Aléas de données : Le résultat d'une opération dépend de celui d'une opération précédente qui n'est pas encore terminée.
  - Aléas de contrôle : L'exécution d'un saut conditionnel ne permet pas de savoir quelle instruction il faut charger dans le pipeline puisque deux choix sont possibles.

Solution

La solution générale pour résoudre un aléa est de bloquer l'instruction qui pose problème et toutes celles qui suivent dans le pipeline jusqu'à ce que le problème se résolve. On voit alors apparaître des bulles dans le pipeline. De manière pratique, la bulle correspond à l'exécution de l'instruction NOP qui ne fait rien.



Les aléas dans le pipeline

Aléas de structures

Les aléas de structure peuvent être éliminés en agissant sur l'architecture du processeur lors de sa conception.

Les aléas dans le pipeline

Aléas de données

- Prenons par exemple la séquence suivante. Cette suite d'instruction possède une dépendance directe simple. En effet A ne peut pas être disponible pour la partie droite de la seconde instruction, puisqu'elle n'est pas encore exécuter lorsque les opérandes de la seconde instruction sont chargés dans le pipeline.
  1. A = B + C
  2. D = A + C
  3. E = F + B

- Une solution consiste à réarranger les instructions. Dans cet exemple, l'opération de la ligne 3 n'a aucune interdépendance avec les deux précédentes. Le code modifié sera :
  1. A = B + C
  2. E = F + B
  3. D = A + C

Les aléas dans le pipeline

Aléas de données

- Si nécessaire, les instructions intercalées peuvent être des NOP.
  1. A = B + C
  2. NOP
  3. D = A + C
  4. E = F + B

- Remarques :
  - Le compilateur n'est pas toujours en mesure de détecter les aléas (par exemple, si les aléas concernent des pointeurs).
  - Le nombre d'instructions à intercaler dépend de la structure (nombre d'étages) du pipeline.
  - La complexité du compilateur en est fortement augmentée.

Les aléas dans le pipeline

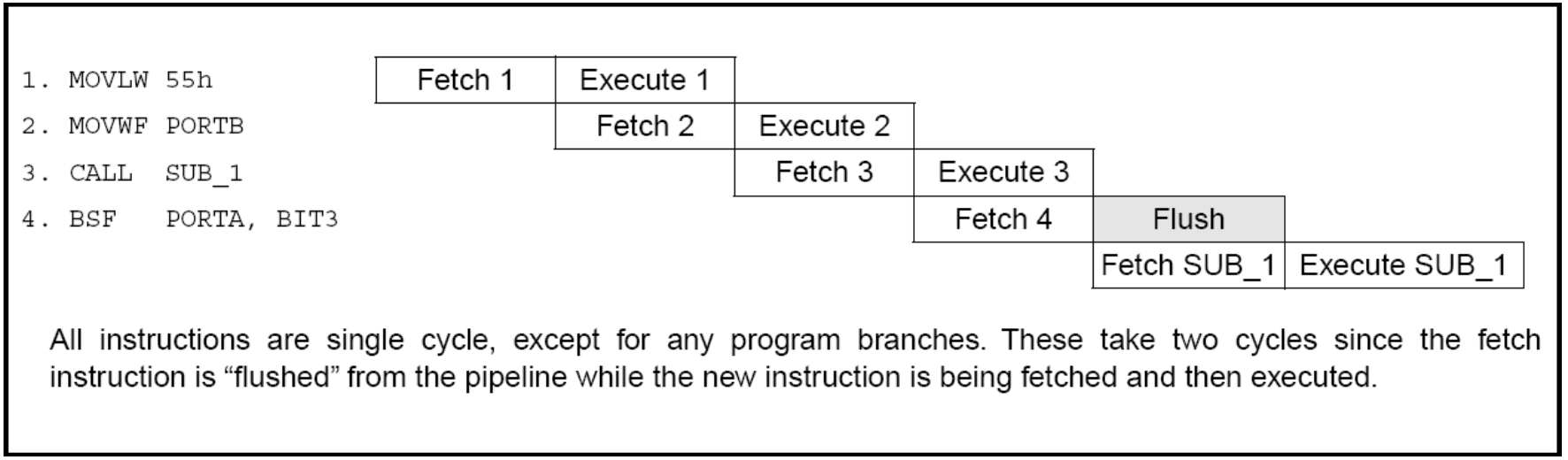
Aléas de données

La fréquence élevée d'aléas de données peut justifier l'introduction de matériel supplémentaire. On peut par exemple introduire une connexion directe entre la sortie de l'étage d'exécution et l'étage de chargement des opérandes. Ceci permet au résultat d'une instruction d'être un opérande de l'instruction suivante.

Les aléas dans le pipeline

Aléas de contrôle

La présence d'un pipeline introduit des complications lors de l'exécution d'un saut ou d'un saut conditionnel. L'étage de décodage de l'instruction n'est pas en mesure de calculer l'adresse de l'instruction suivante avant de connaître le résultat de l'instruction précédente.



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

Les aléas dans le pipeline

Aléas de contrôle

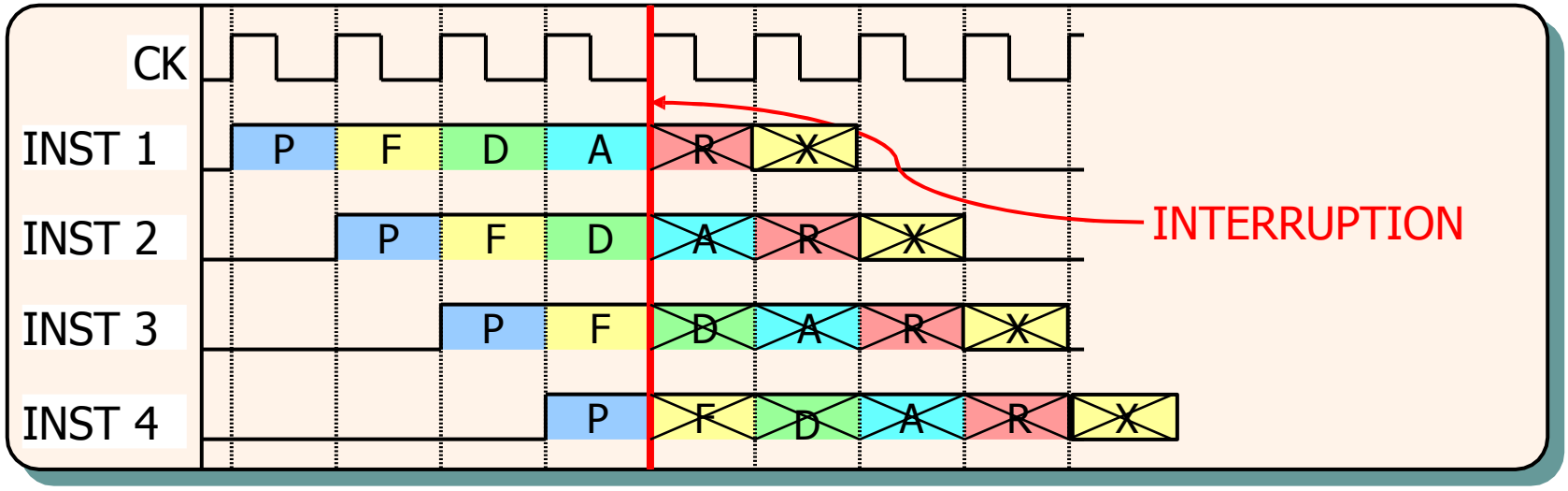
- Une solution possible est de faire en sorte que le processeur devine si le branchement sera pris ou pas pris (branch prediction) et commencer à exécuter les instructions correspondant à cette décision.
  - Si le choix se révèle correct, la pénalité de branchement est éliminée.
  - Si le choix se révèle incorrect, il faudra vider le pipeline et charger l'instruction correcte.

- Pour faire de la prédiction de branchement il y a deux possibilités :
  - Solution statique : La direction du branchement est fixe, définie en matériel au moment de la conception du processeur.
  - Solution dynamique : La direction du branchement est définie au moment de l'exécution du programme, sur la base d'une analyse du code.

Les aléas dans le pipeline

Gestion des interruptions

La présence d'un pipeline complique le traitement des interruptions: lors du déclenchement d'une interruption non-masquable, la routine de traitement doit parfois être lancée immédiatement. Le pipeline contiendra alors des instructions partiellement exécutées.



Les aléas dans le pipeline

Résumé

- Le pipeline améliore le débit mais pas le temps par instruction : il faut toujours 4 cycles à une instruction d'un pipeline à 4 étages pour s'exécuter.

- Les dépendances de données et de contrôle dans les programmes imposent une limite supérieure au gain que peut générer le pipeline car le processeur doit parfois attendre la fin d'une instruction pour que les dépendances soit résolues.

- On peut élever cette limite, mais pas l'éliminer, en réduisant les aléas de contrôle par des optimisations, et les aléas de données par un ordonnancement des instructions par le compilateur.