



Pipeline d'instructions

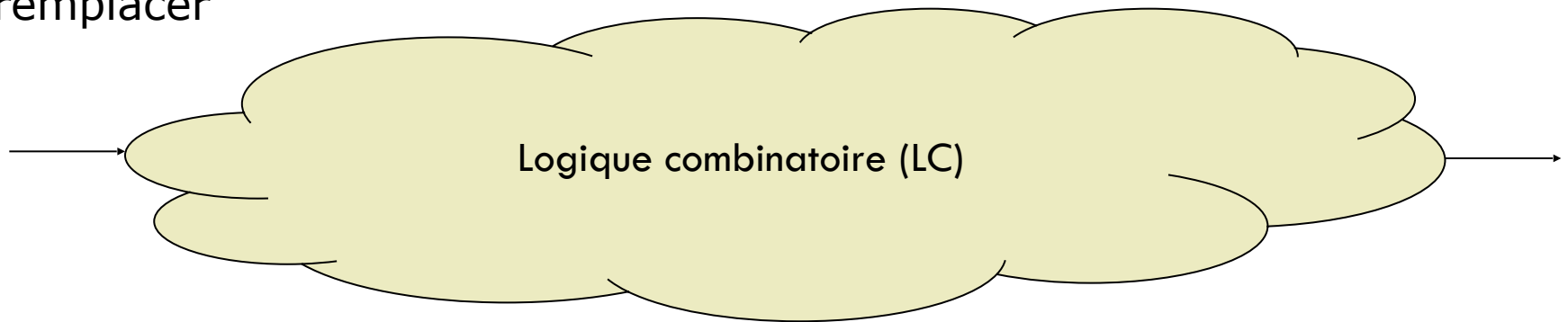
Principes

- ***Pipeline*** : technique d'implantation dans laquelle ***des instructions multiples se recouvrent (overlapping), i.e. s'exécutent partiellement simultanément***

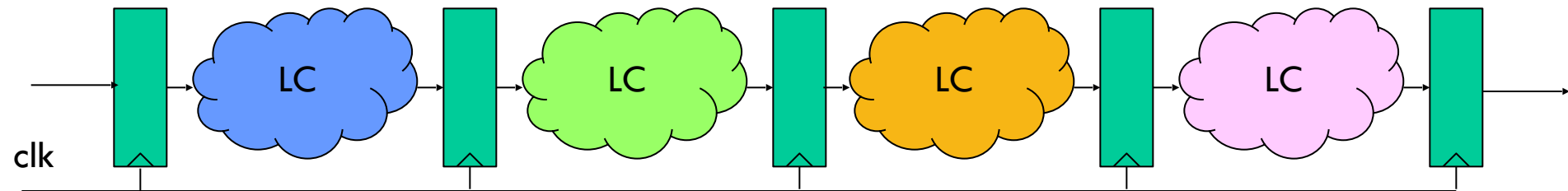
- Différentes étapes importantes de l'exécution des instructions sur les processeurs se dégagent :
 - ▣ Récupération (fetch) de l'instruction à partir de la mémoire
 - ▣ Lecture de registre et décodage instruction
 - ▣ Exécution de l'opération ou calcul d'adresse
 - ▣ Accès à un opérande en mémoire
 - ▣ Ecriture du résultat dans un registre

Principes

remplacer

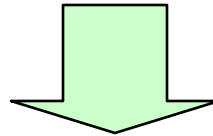


par

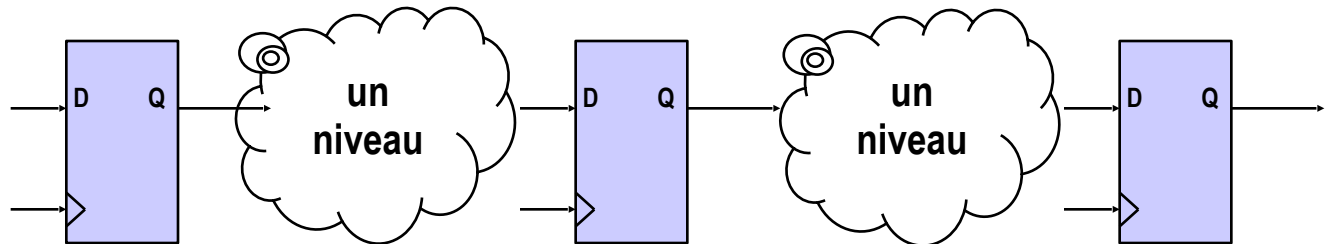


Amélioration : fréquence

$$f_{\text{MAX}} = n \text{ MHz}$$



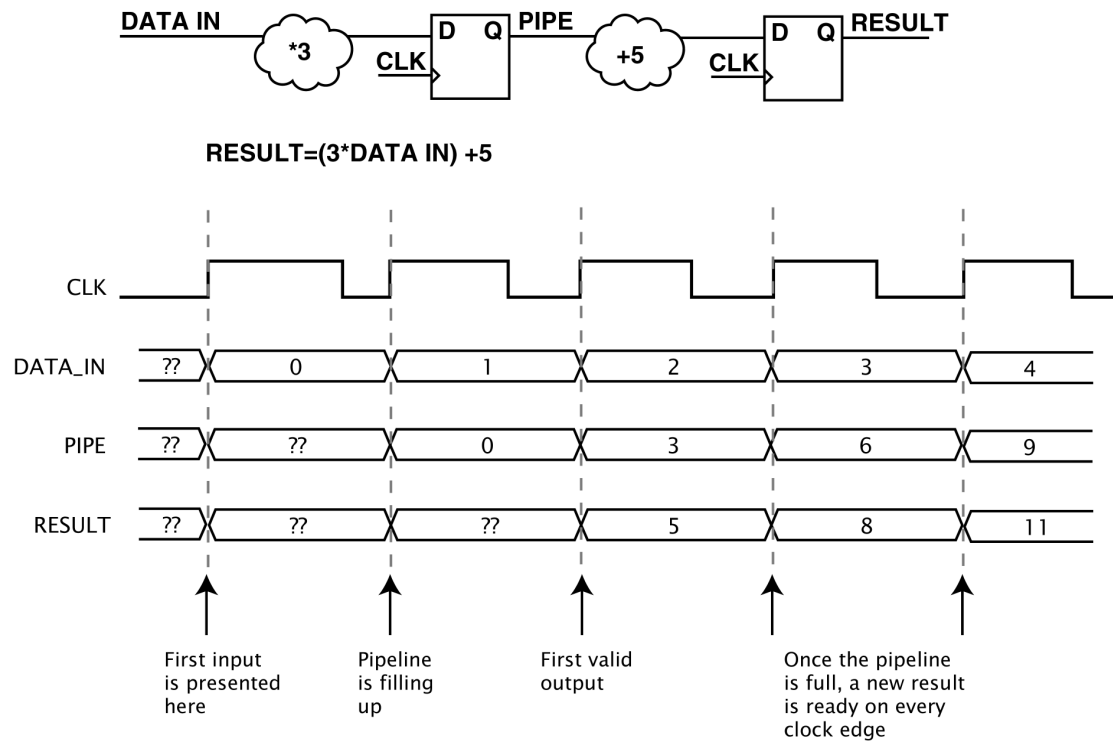
$$f_{\text{MAX}} \approx 2n \text{ MHz}$$



Problèmes de conception de pipeline

- Dispose-t-on des ressources additionnelles de stockage ?
 - ▣ Registres
- Quels sont les niveaux logiques multiples entre les différents étages du pipeline ?
 - ▣ Une partition équilibrée est souhaitable
- Le système à concevoir tolère-t-il la latence (délai d'obtention des résultats) ?
 - ▣ Chaque étage du pipeline introduit de la latence

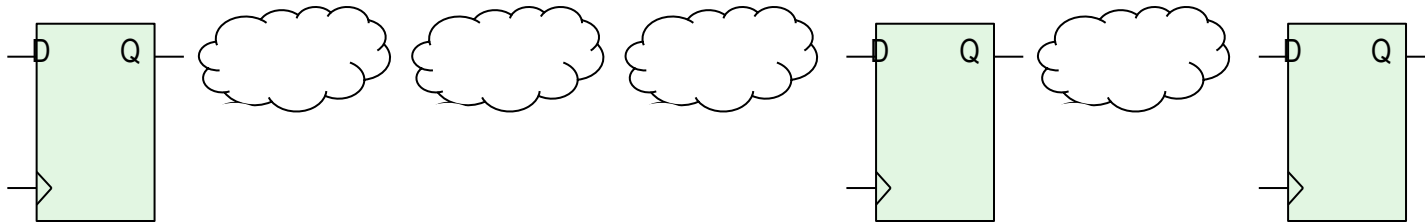
Pipeline et latence



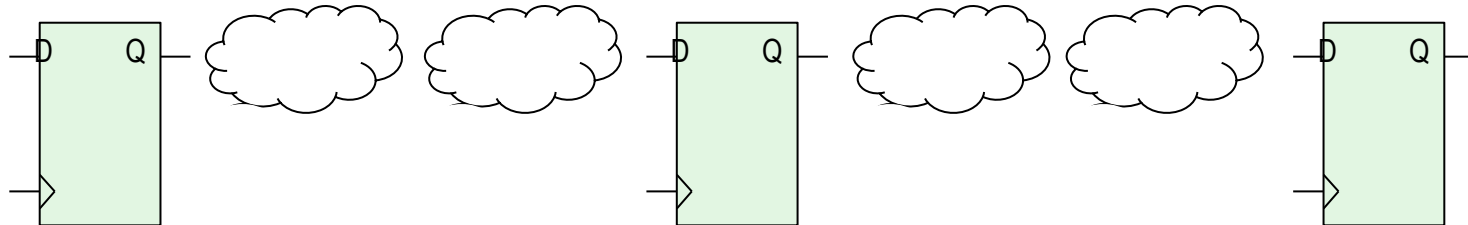
- Chaque étage de pipeline ajoute un cycle d'horloge de délai avant que la première sortie valide ne soit disponible (remplissage du pipeline)
- Une fois que le pipeline est rempli, une nouvelle sortie est disponible à chaque cycle d'horloge

Equilibrage du pipeline : retiming

Avant retiming



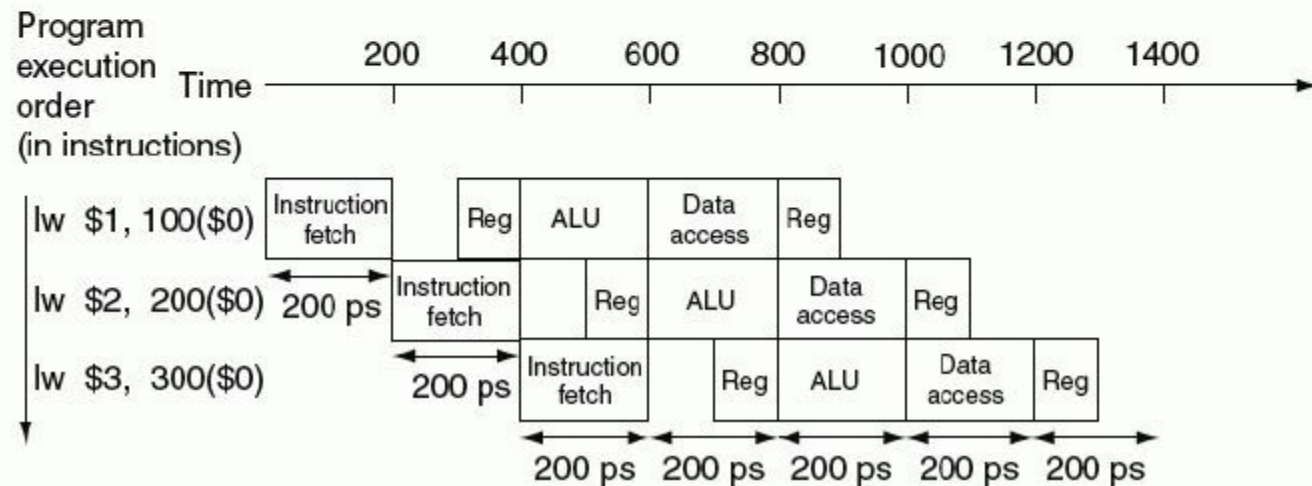
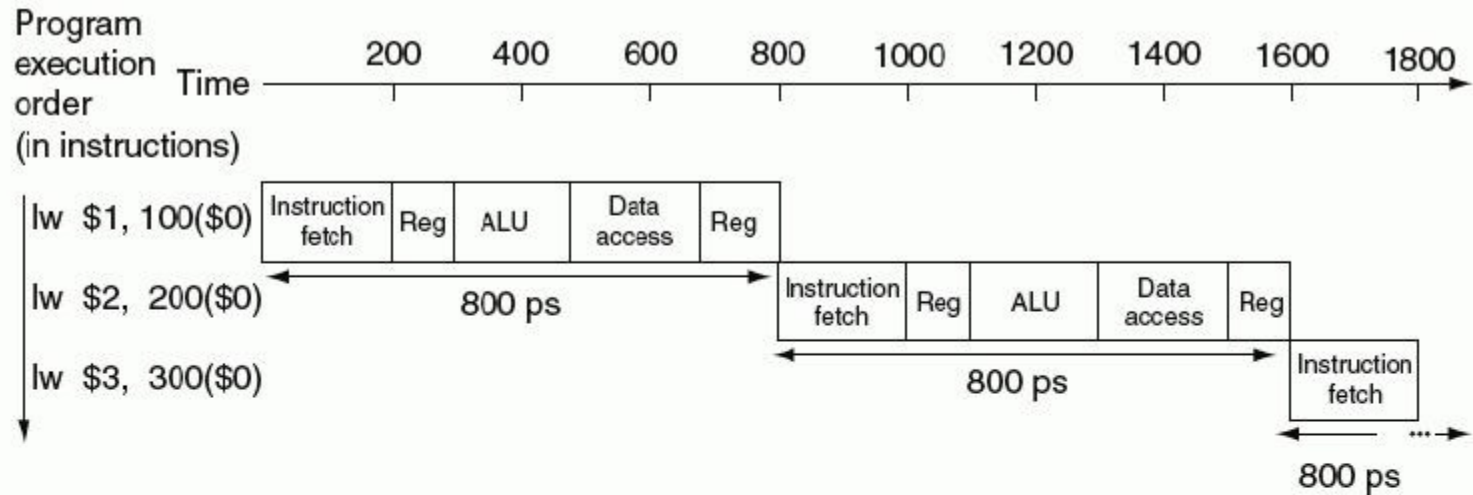
Après retiming



Pipeline dans le processeur

- Rappel : étapes de traitement d'une instruction par un processeur
 - ▣ Récupération de l'instruction à partir de la mémoire (IF, instruction fetch)
 - ▣ Lecture de registre et décodage d'instruction (ID, instruction decoding)
 - ▣ Exécution de l'opération ou calcul d'une adresse (EX, execution)
 - ▣ Accès à un opérande en mémoire de données (MEM)
 - ▣ Ecriture du résultat dans un registre (WB, write back)
- Le pipeline pose des problèmes/défis aux concepteurs matériels en raison des dépendances de données et de contrôle (**pipeline hazards, risques/aléas du pipeline**)

Pipeline dans le processeur



Aléas de pipeline

- Aléa structurel (structural hazard) :
 - ▣ Situation dans laquelle une *instruction planifiée ne peut pas être exécutée dans le cycle d'horloge prévu car le matériel ne tolère pas la combinaison d'instructions induite* dans ce cycle d'horloge (matériel ne supportant pas certaines micro-instructions concurrentes)
- Aléa de données (data hazard) :
 - ▣ Situation dans laquelle une instruction planifiée ne peut pas être exécutée dans le cycle d'horloge prévu *car les données nécessaires à cette exécution ne sont pas encore disponibles*

Aléas de pipeline

- Aléa de données : exemple 1

add \$s0,\$t0,\$t1

sub \$t2,\$s0,\$t3

l'étape ID de la deuxième instruction ne dispose pas de la bonne valeur de \$s0 avant la fin de l'étape WB de la première

- Aléa de données : exemple 2

lw \$s0,20(\$t1)

sub \$t2,\$s0,\$t3

même problème, mais solution différente (car la mémoire est impliquée)

Aléas de pipeline

- Aléa de données : exemple 1

add \$s0,\$t0,\$t1

sub \$t2,\$s0,\$t3

solution : transmission anticipée

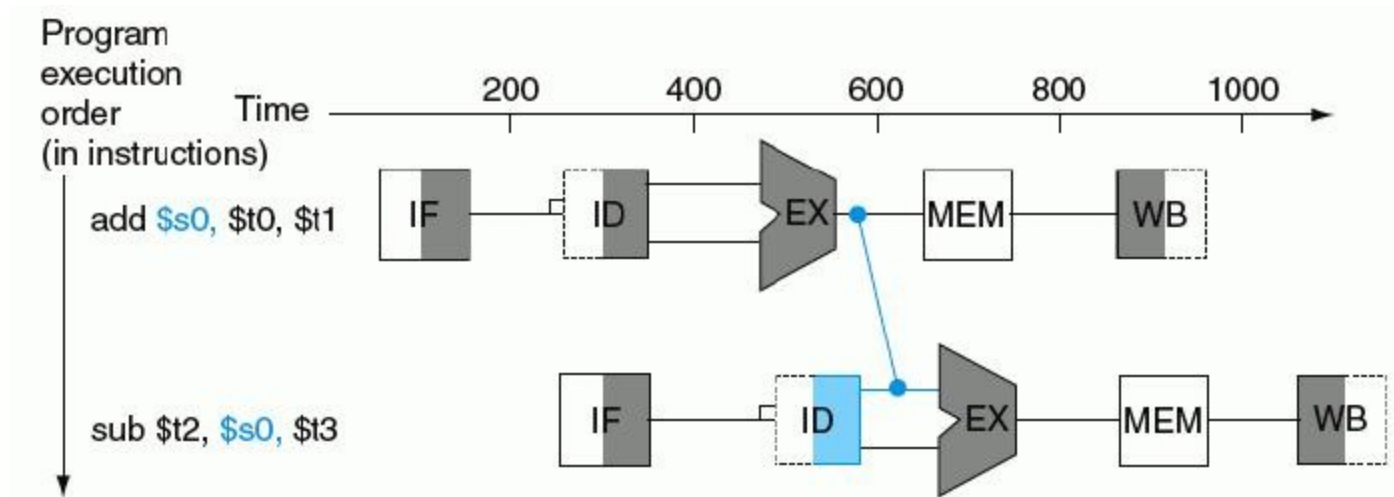
- Aléa de données : exemple 2

lw \$s0,20(\$t1)

sub \$t2,\$s0,\$t3

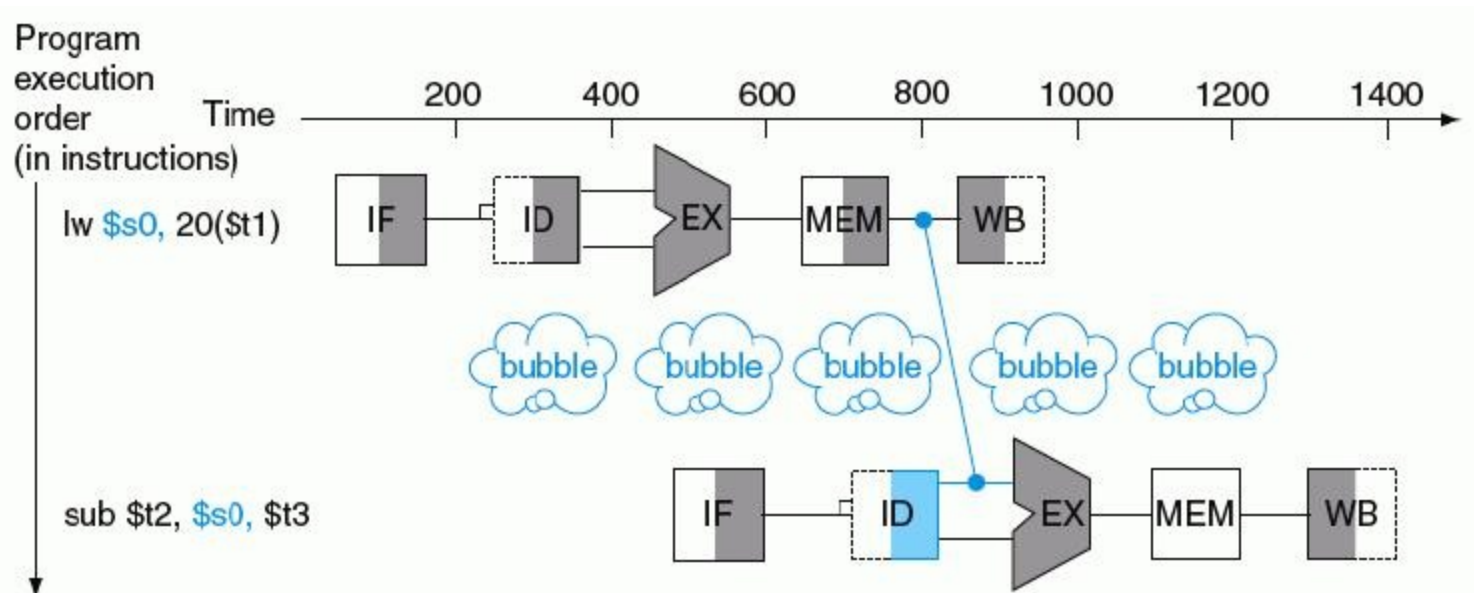
solution partielle : cycle de « stalled pipeline » (pipeline bloqué)

Aléa de données / transmission (forwarding)



Forwarding : résultat connu dès la fin de l'étape EX, sans attendre WB

Aléa de données / pipeline bloqué (stalled)



Stalled pipeline + forwarding : contenu mémoire connu dès l'étape MEM, sans attendre WB

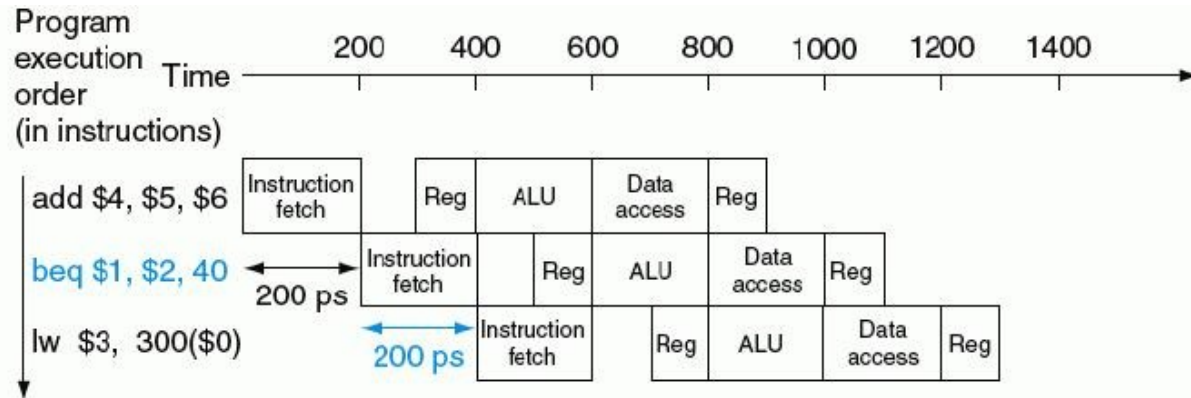
Aléas de pipeline

- Aléa de contrôle :
 - ▣ Situation dans laquelle une instruction ne peut pas être exécutée dans le cycle d'horloge prévu ***car l'instruction qui a été récupérée en mémoire n'est pas celle qui doit être exécutée***; autrement dit, la suite des adresses des instructions à exécuter n'est pas celle prévue (rupture de pipeline par branchement)
 - ▣ Deux solutions possibles :
 - **Blocage (stalled pipeline)** : le pipeline est interrompu à chaque branchement conditionnel (jusqu'à être certain de l'instruction à exécuter)
 - **Prédiction** : prédit que le branchement ne va pas être pris/suivi, et poursuit alors le pipeline à rythme normal; ou alors prédit que le branchement va être pris, et bloque alors le pipeline

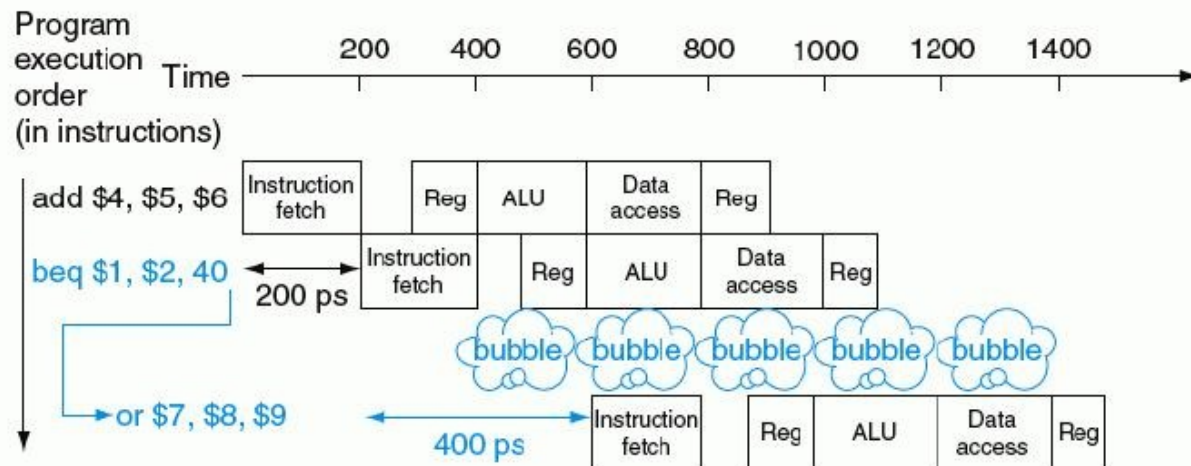
Prédiction de branchement

- Méthode de résolution d'aléa qui suppose un certain résultat pour la décision de branchement et procède en fonction de ce résultat supposé plutôt que d'attendre l'issue véritable de la décision
 - ▣ Les ***prédicteurs dynamiques*** établissent leur prédiction en fonction du comportement de chaque branchement et peuvent changer ces prédictions au cours du déroulement du programme

Pipeline et prédiction de branchement



Prédiction : branchement non pris



Prédiction : branchement pris (cycle pour préparer l'adresse de branchement)

Pipeline : résumé

- Le pipeline d'instructions augmente le nombre d'instructions exécutées simultanément et le rythme auquel ces instructions sont démarrées et achevées
- Le pipeline d'instructions ne réduit pas le temps total mis pour exécuter totalement chaque instruction individuelle (latence)
 - ▣ Le pipeline améliore le débit des instructions et non la latence
 - ▣ La latence est sensiblement proportionnelle au nombre d'étages du pipeline



Pipeline d'instructions : cas du processeur MIPS

Chemin de données et étages du pipeline

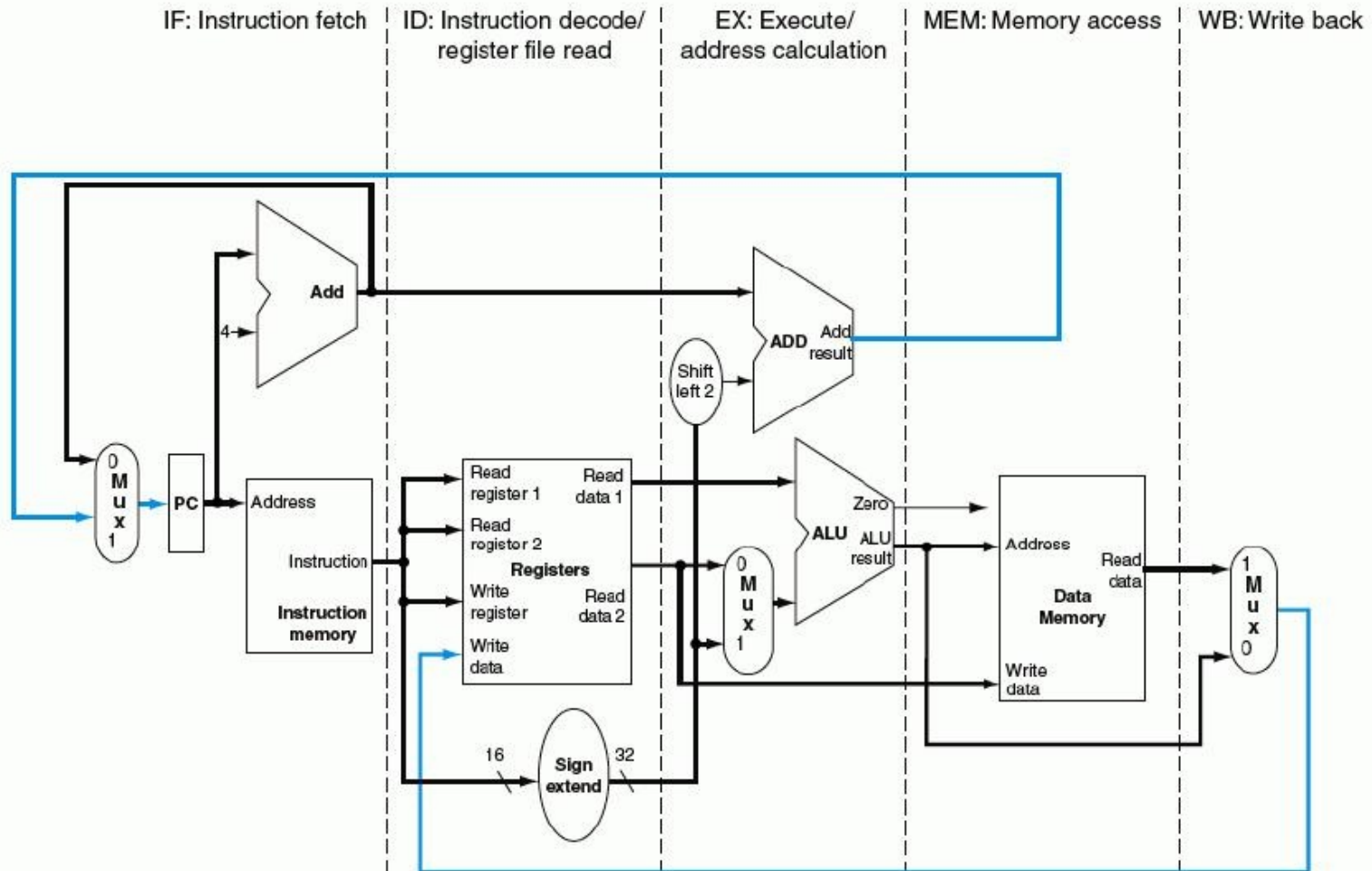
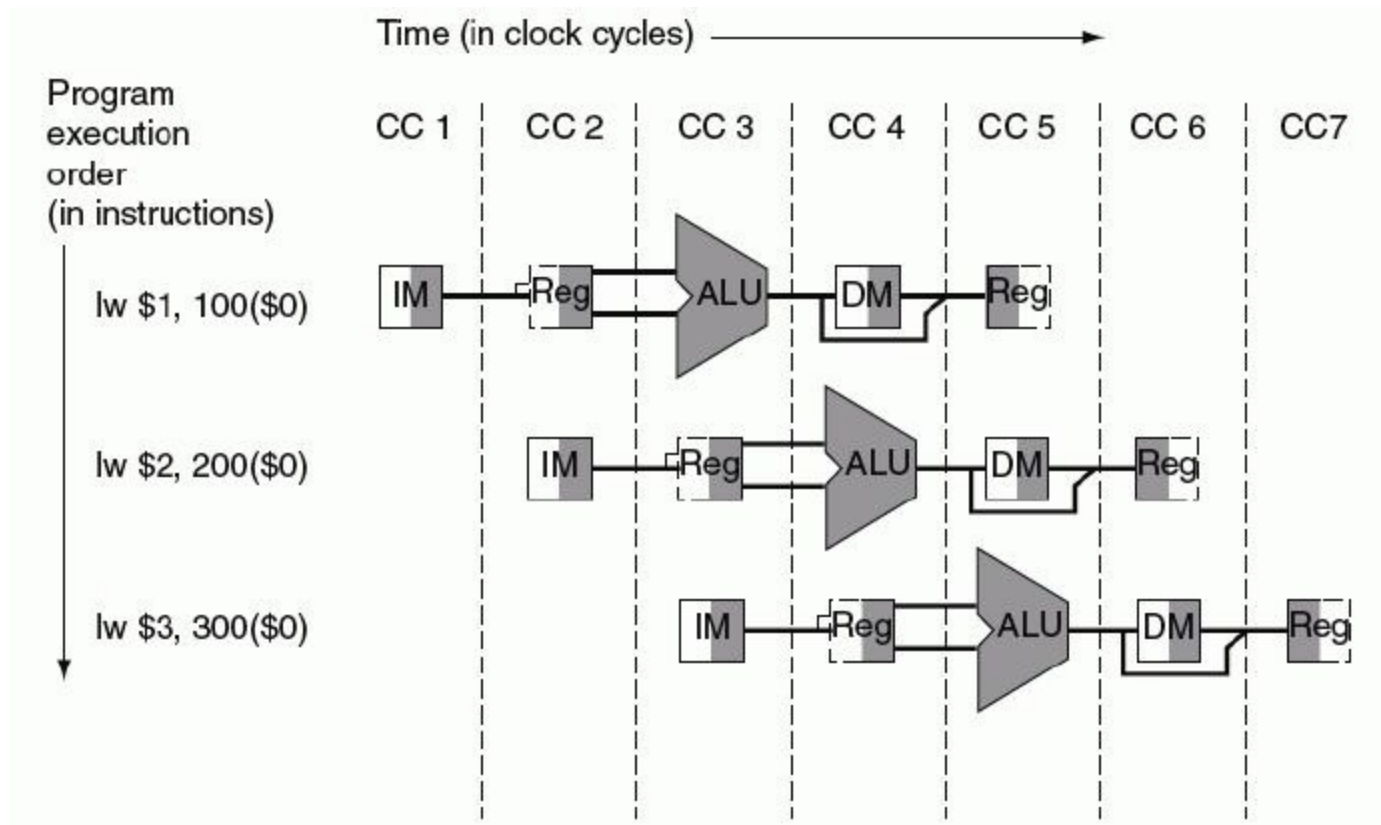
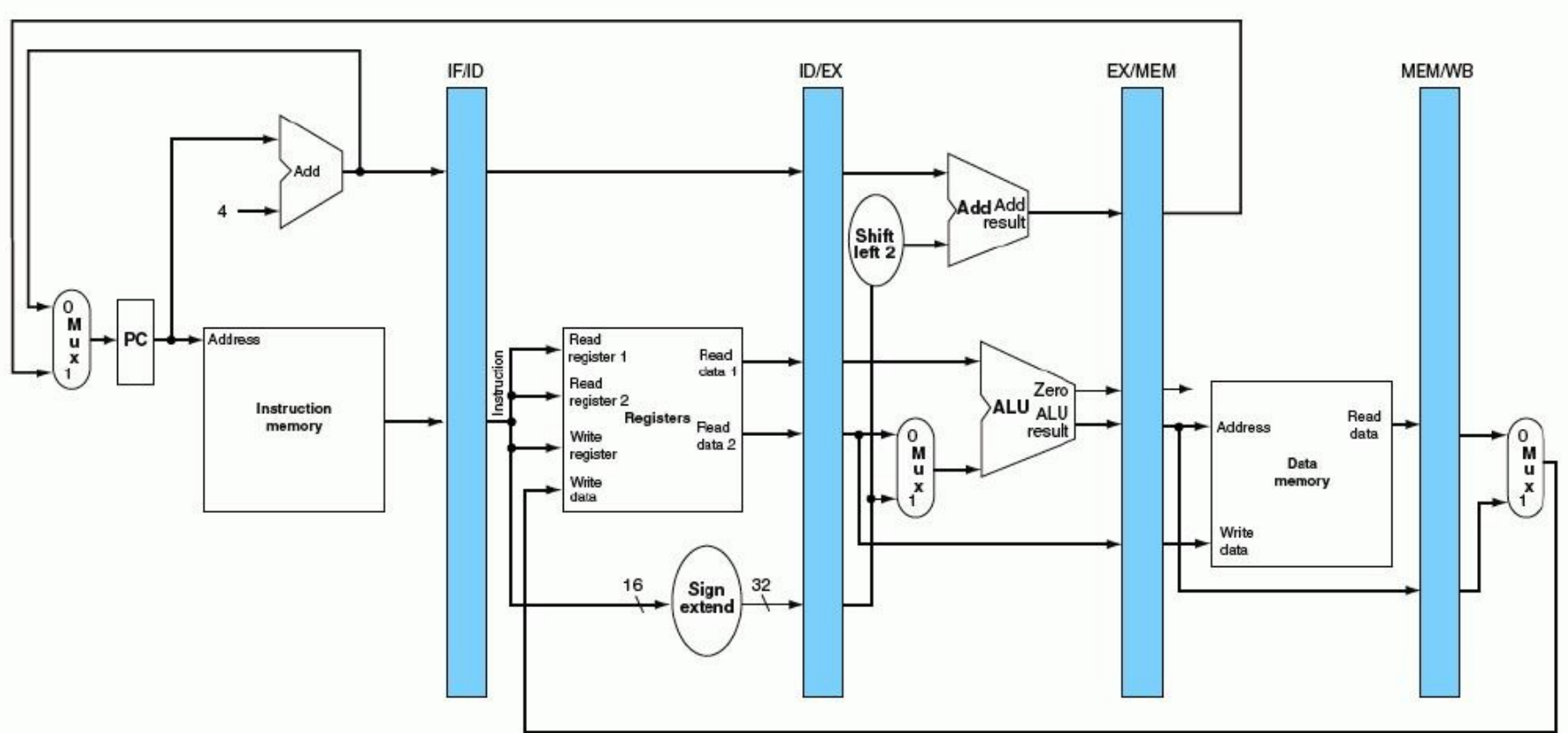


Diagramme de pipeline



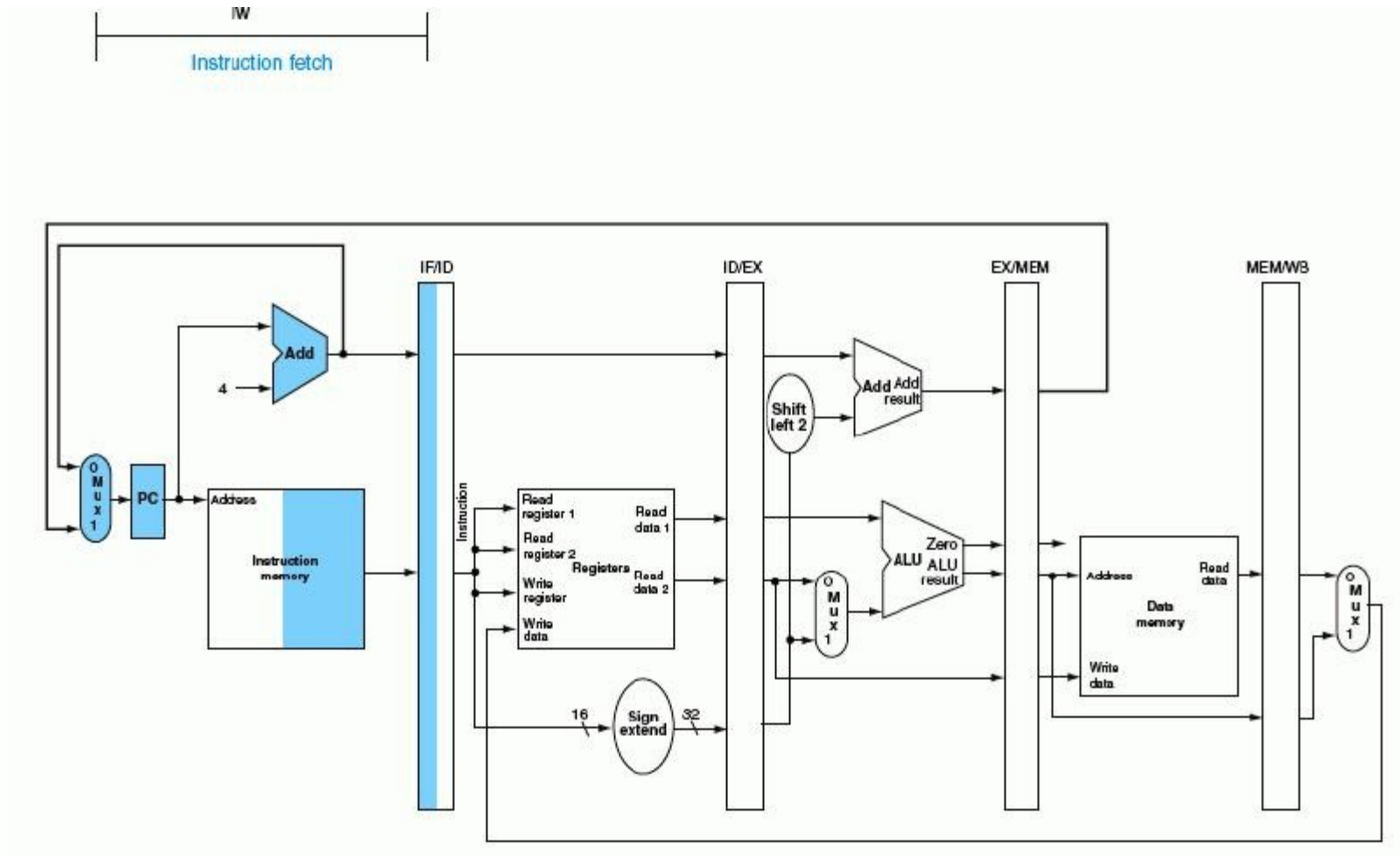
IM : instruction memory – Reg : registers – DM : data memory

Chemin de données et ressources du pipeline



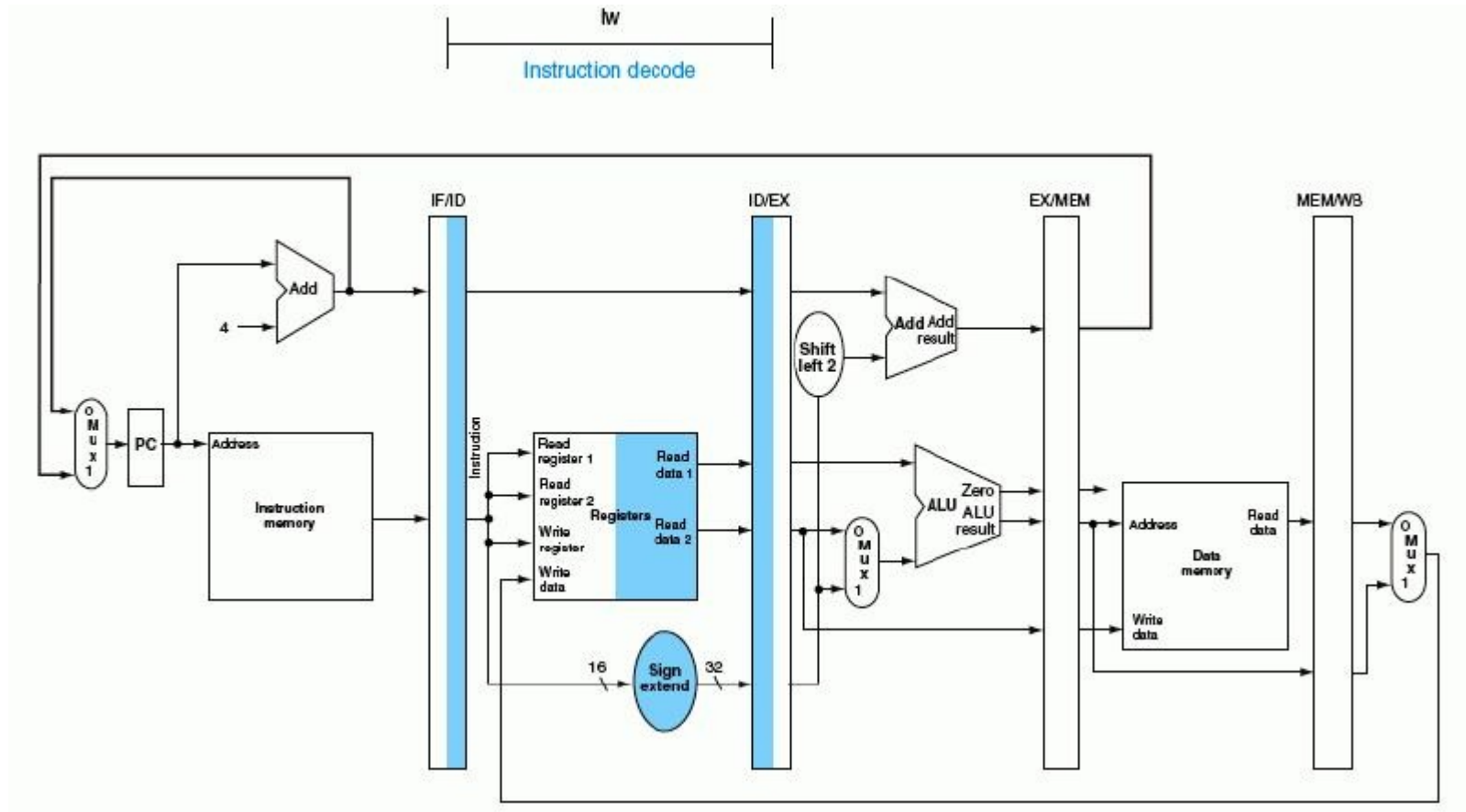
Introduction des registres entre les étages de pipeline

Pipeline / lw IF (instruction fetch)



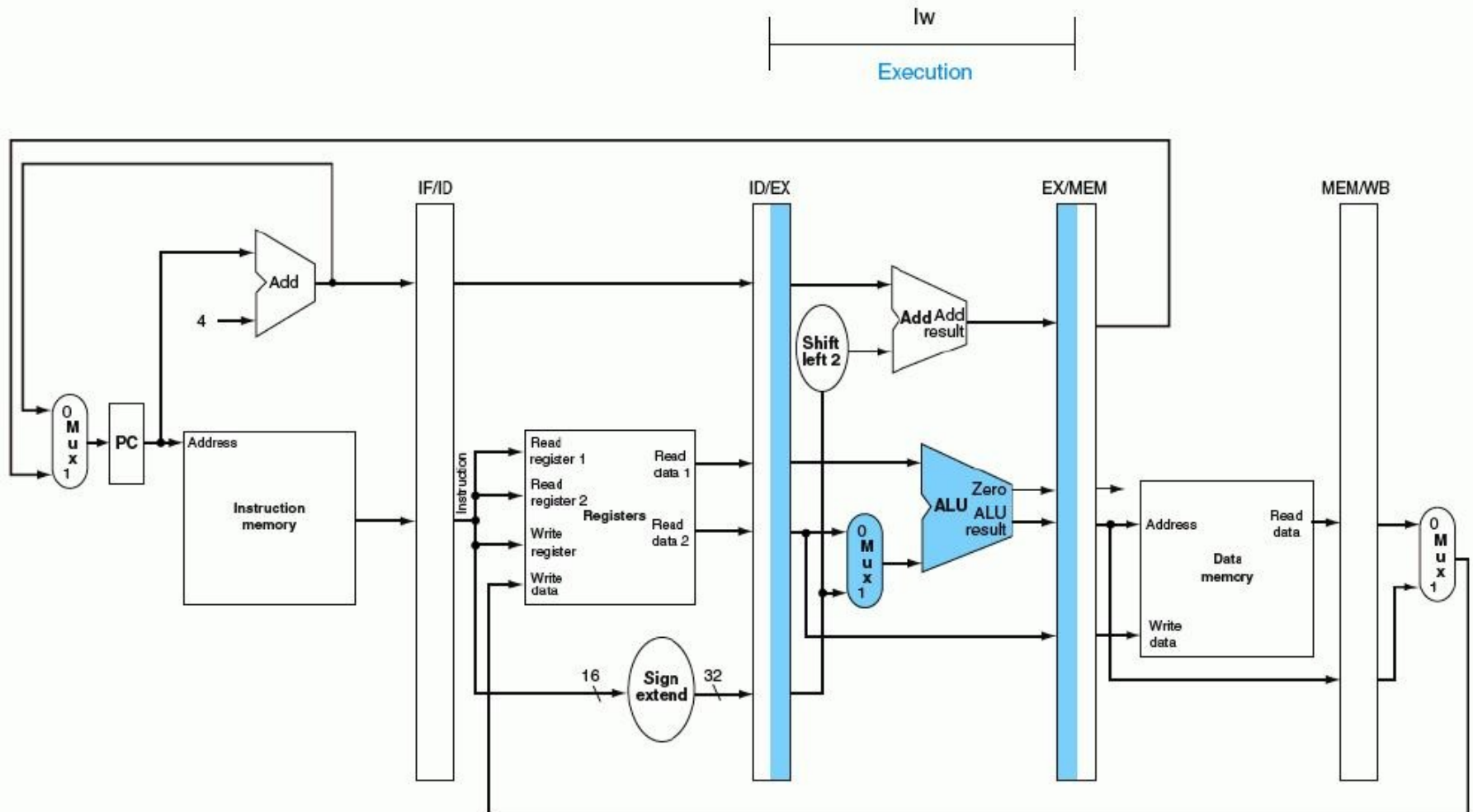
IF : instruction lue en mémoire et stockée dans le registre IF/ID

Pipeline / lw ID (instruction decode)



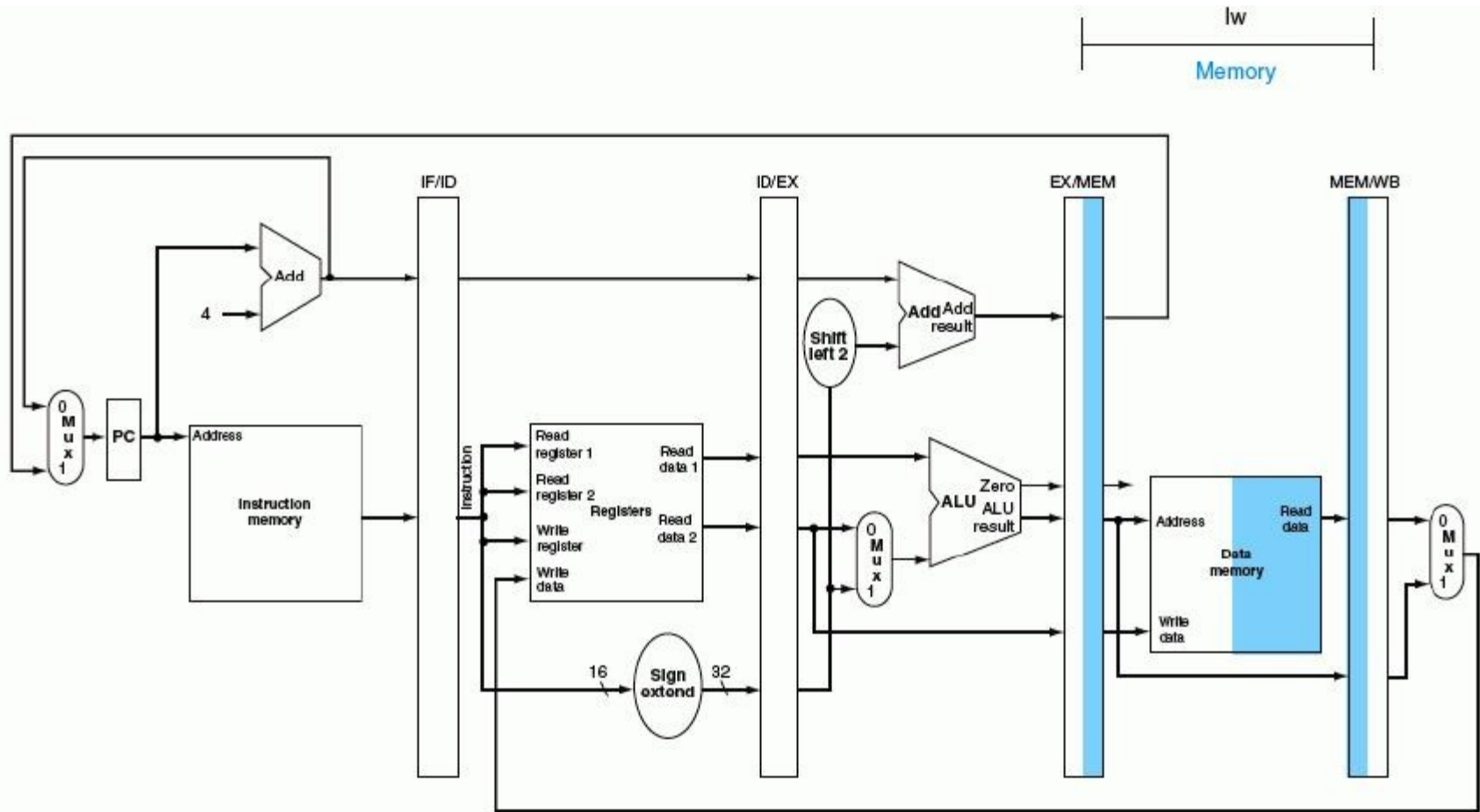
ID : instruction décodée, contenus registres utiles (et éventuelle valeur immédiate) stockés dans le registre ID/EX

Pipeline / lw EX (execution)



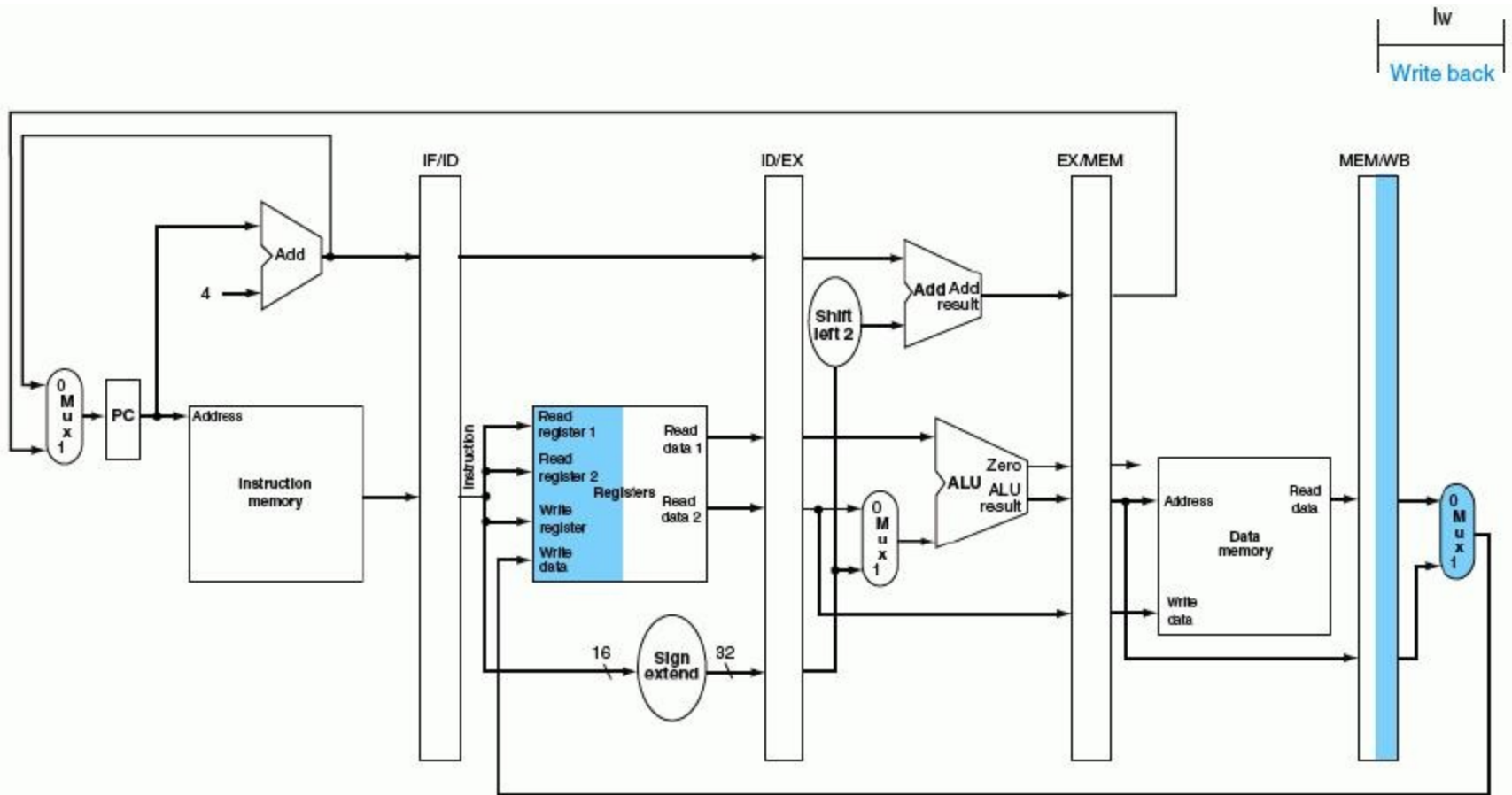
EX : résultat ou adresse mémoire calculé(e) et stocké(e) dans registre EX/MEM

Pipeline / lw MEM (memory)



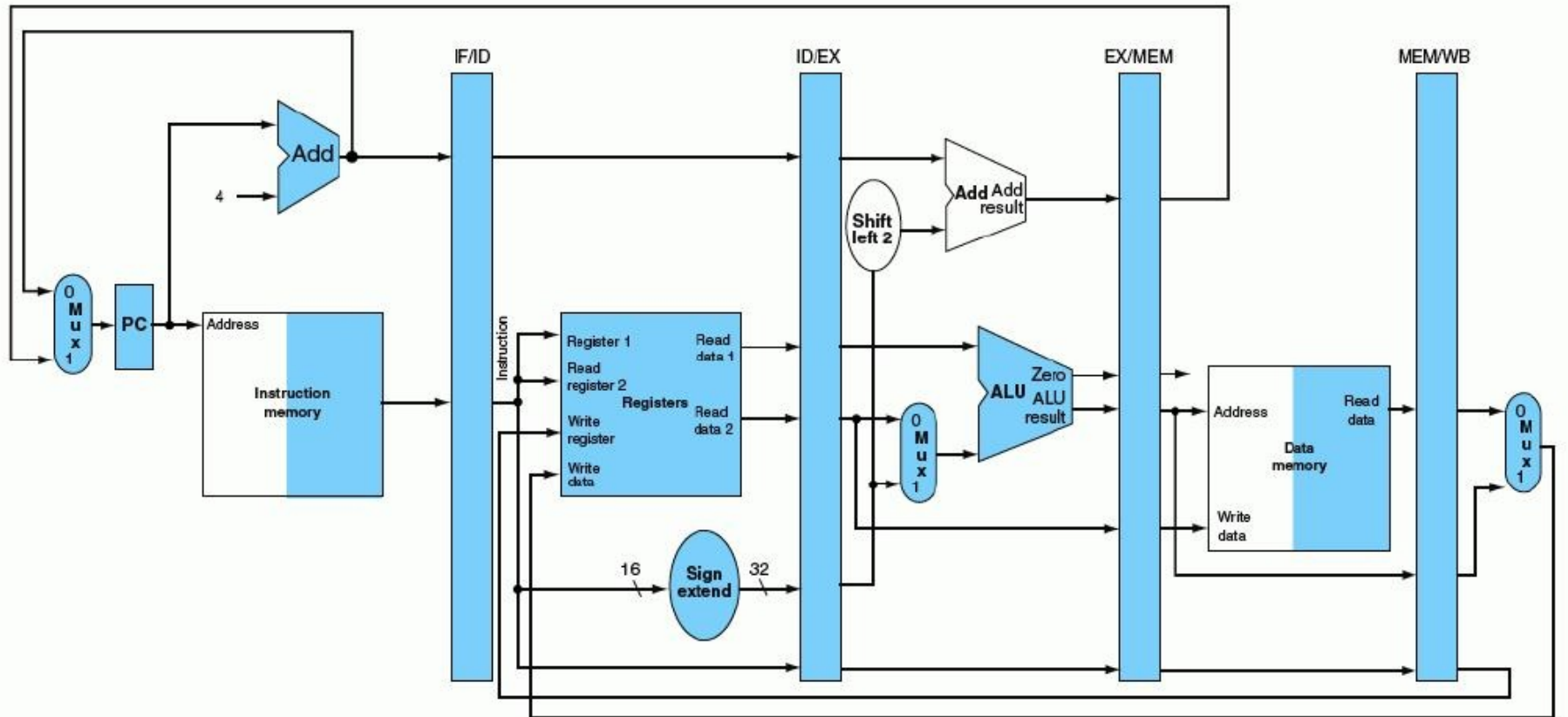
MEM : donnée éventuellement lue/écrite en mémoire et stockée si besoin dans le registre MEM/WB

Pipeline / lw WB (write back)



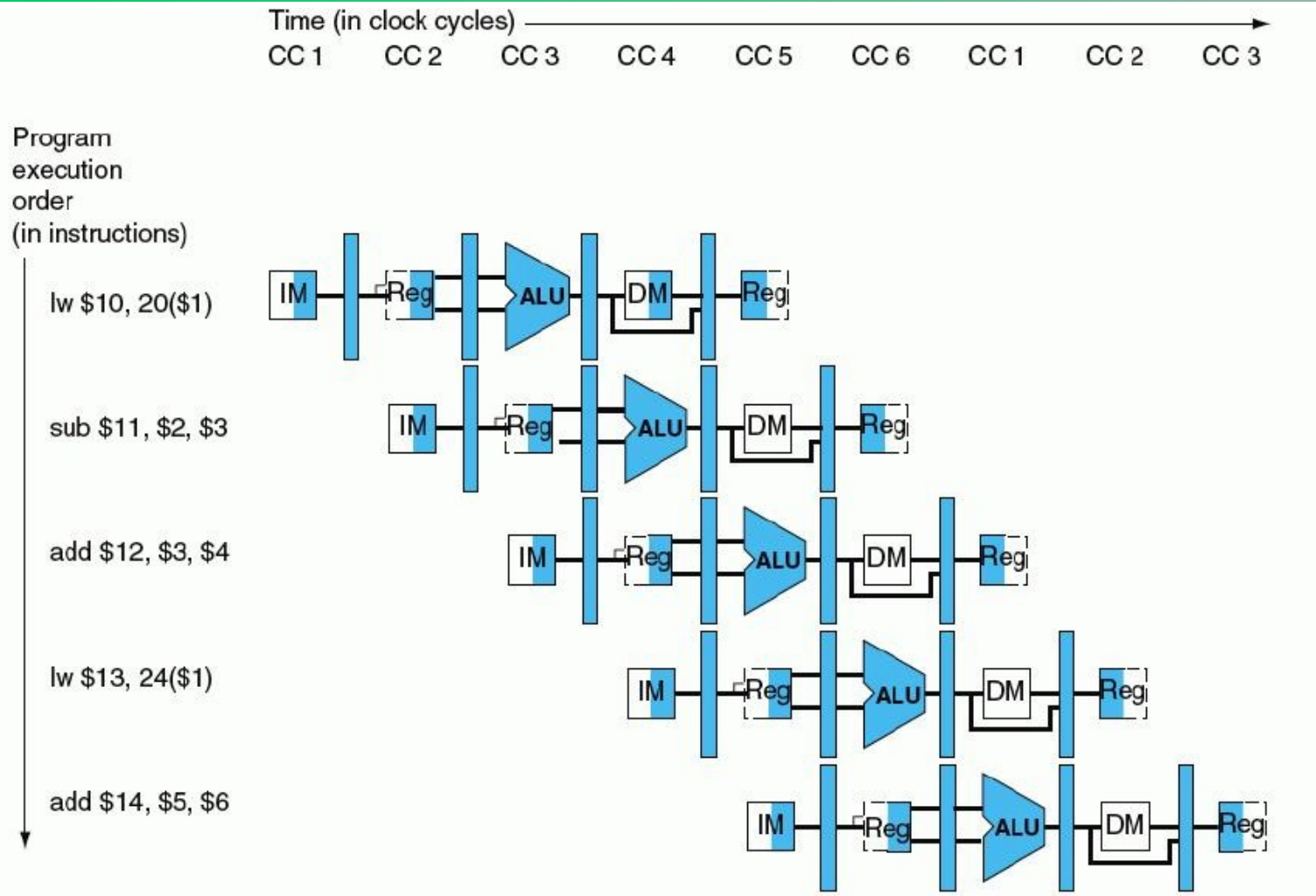
WB : donnée calculée ou lue écrite dans la banque de registres

Chemin de données pipeliné (lw)



instruction complète (lw) : ressources utilisées

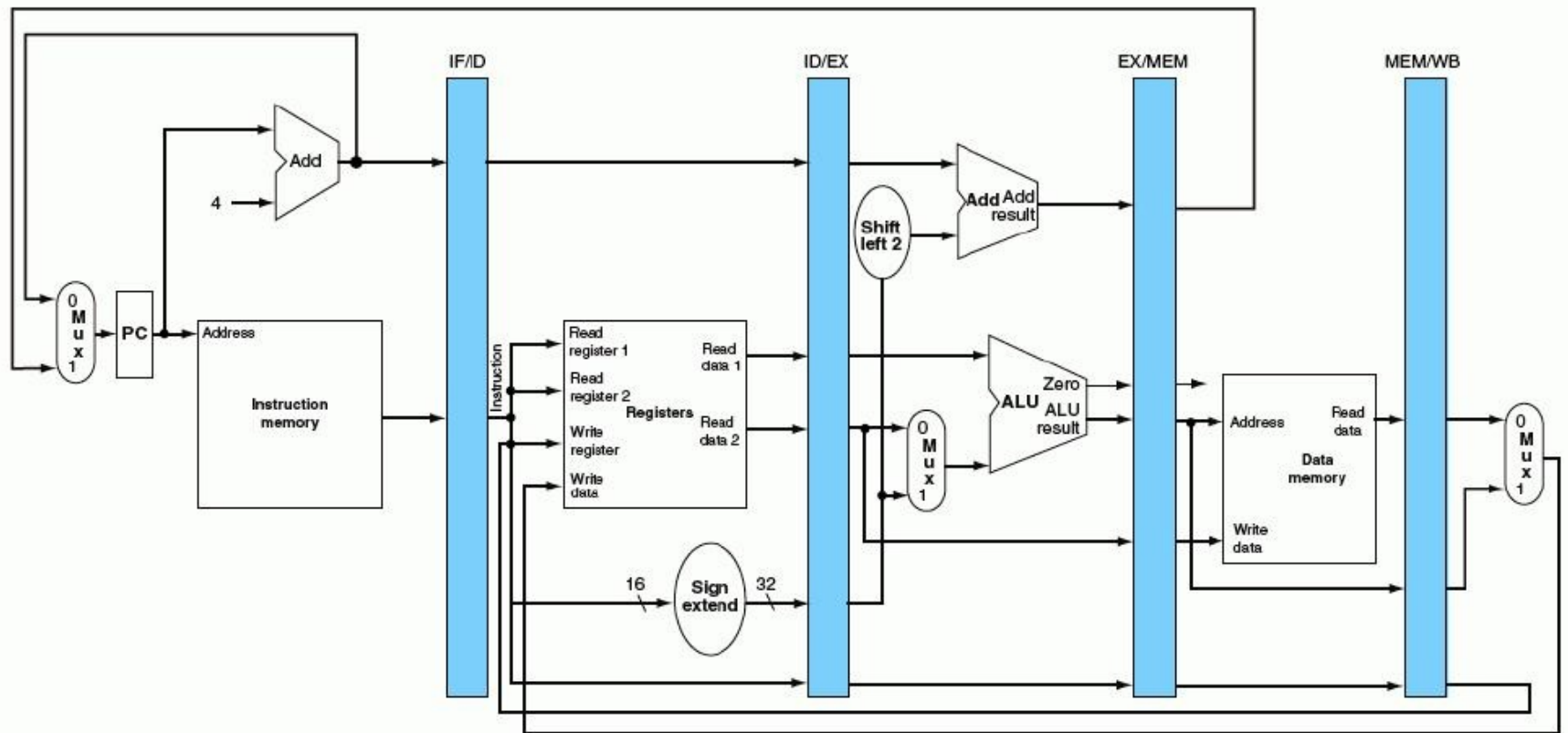
Pipeline : séquence d'instructions



pipeline : à chaque cycle, chaque ressource n'est utilisée au plus que par une instruction

Pipeline : cycle d'horloge simple (CC5)

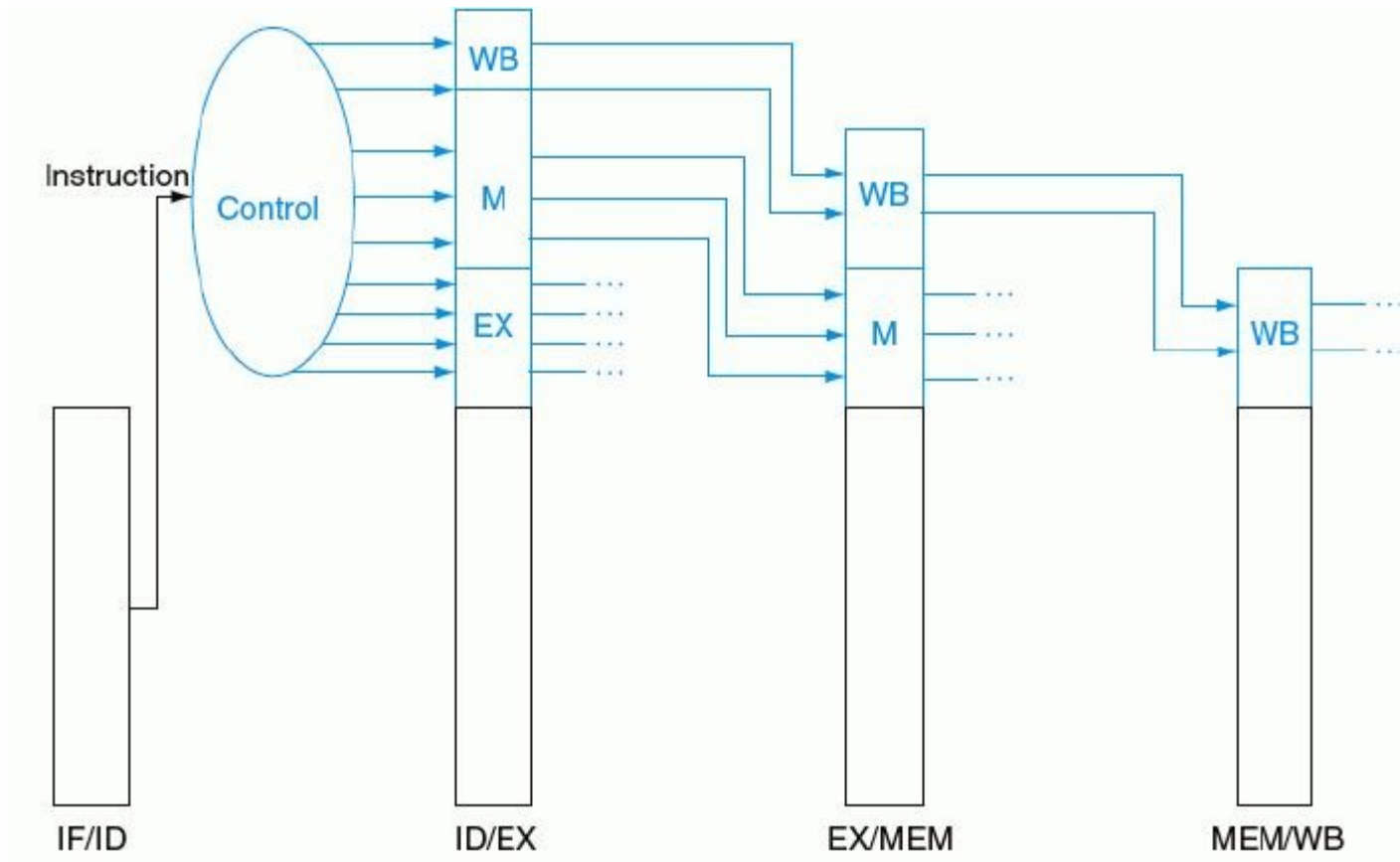
add \$14, \$5, \$6	lw \$13, 24 (\$1)	add \$12, \$3, \$4, \$11	sub \$11, \$2, \$3	lw\$10, 20(\$1)
Instruction fetch	Instruction decode	Execution	Memory	Write back



Affectation des ressources du processeur aux différentes instructions pendant le cycle 5

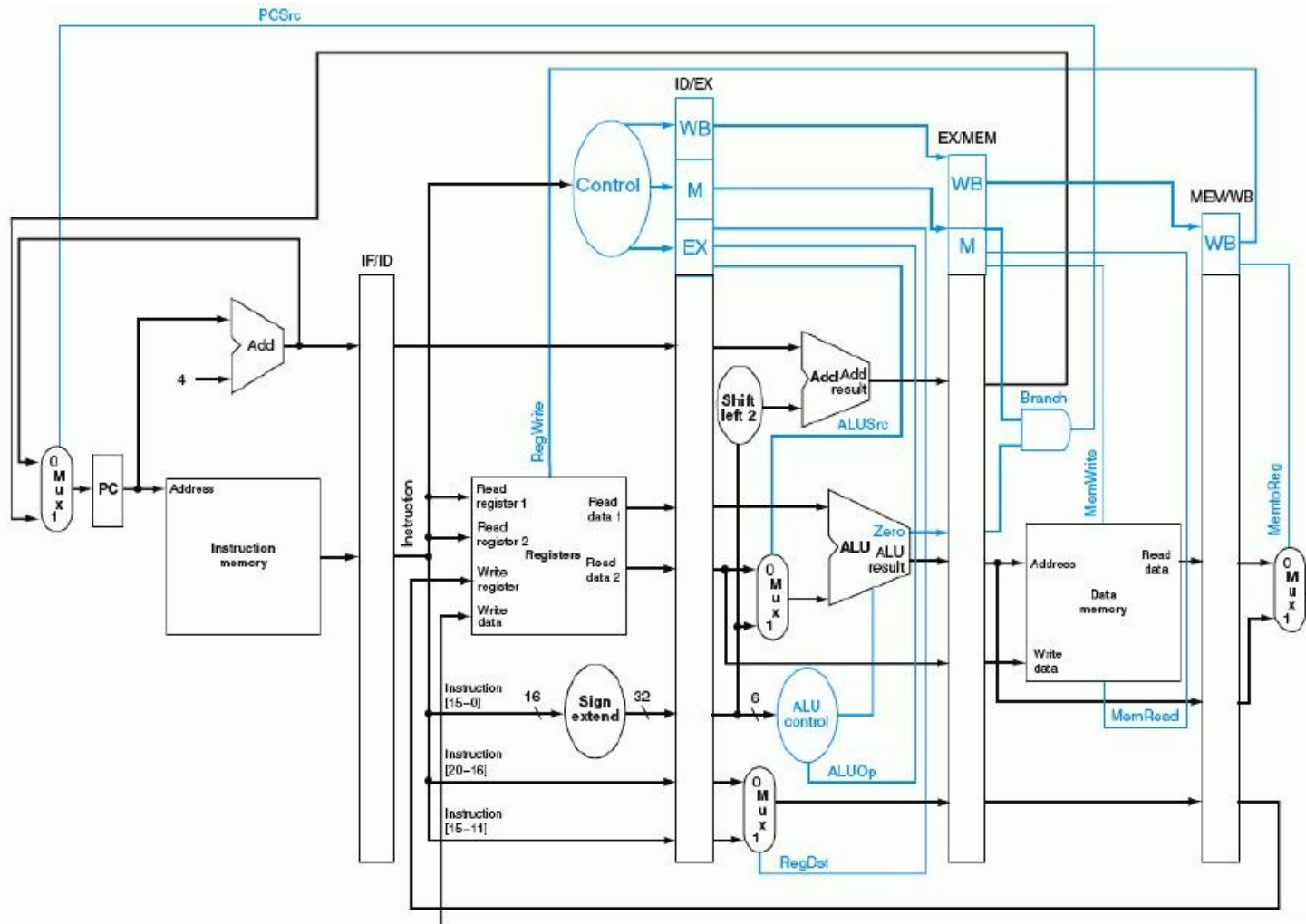
Year	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100																																																																																																													
Population	7,811,000	7,860,000	7,909,000	7,958,000	8,007,000	8,056,000	8,105,000	8,154,000	8,203,000	8,252,000	8,301,000	8,350,000	8,399,000	8,448,000	8,497,000	8,546,000	8,595,000	8,644,000	8,693,000	8,742,000	8,791,000	8,840,000	8,889,000	8,938,000	8,987,000	9,036,000	9,085,000	9,134,000	9,183,000	9,232,000	9,281,000	9,330,000	9,379,000	9,428,000	9,477,000	9,526,000	9,575,000	9,624,000	9,673,000	9,722,000	9,771,000	9,820,000	9,869,000	9,918,000	9,967,000	10,016,000	10,065,000	10,114,000	10,163,000	10,212,000	10,261,000	10,310,000	10,359,000	10,408,000	10,457,000	10,506,000	10,555,000	10,604,000	10,653,000	10,702,000	10,751,000	10,800,000	10,849,000	10,898,000	10,947,000	10,996,000	11,045,000	11,094,000	11,143,000	11,192,000	11,241,000	11,290,000	11,339,000	11,388,000	11,437,000	11,486,000	11,535,000	11,584,000	11,633,000	11,682,000	11,731,000	11,780,000	11,829,000	11,878,000	11,927,000	11,976,000	12,025,000	12,074,000	12,123,000	12,172,000	12,221,000	12,270,000	12,319,000	12,368,000	12,417,000	12,466,000	12,515,000	12,564,000	12,613,000	12,662,000	12,711,000	12,760,000	12,809,000	12,858,000	12,907,000	12,956,000	13,005,000	13,054,000	13,103,000	13,152,000	13,201,000	13,250,000	13,299,000	13,348,000	13,397,000	13,446,000	13,495,000	13,544,000	13,593,000	13,642,000	13,691,000	13,740,000	13,789,000	13,838,000	13,887,000	13,936,000	13,985,000	14,034,000	14,083,000	14,132,000	14,181,000	14,230,000	14,279,000	14,328,000	14,377,000	14,426,000	14,475,000	14,524,000	14,573,000	14,622,000	14,671,000	14,720,000	14,769,000	14,818,000	14,867,000	14,916,000	14,965,000	15,014,000	15,063,000	15,112,000	15,161,000	15,210,000	15,259,000	15,308,000	15,357,000	15,406,000	15,455,000	15,504,000	15,553,000	15,602,000	15,651,000	15,700,000	15,749,000	15,798,000	15,847,000	15,896,000	15,945,000	15,994,000	16,043,000	16,092,000	16,141,000	16,190,000	16,239,000	16,288,000	16,337,000	16,386,000	16,435,000	16,484,000	16,533,000	16,582,000	16,631,000	16,680,000	16,729,000	16,778,000	16,827,000	16,876,000	16,925,000	16,974,000	17,023,000	17,072,000	17,121,000	17,170,000	17,219,000	17,268,000	17,317,000	17,366,000	17,415,000	17,464,000	17,513,000	17,56

Pipeline : contrôle pipeliné



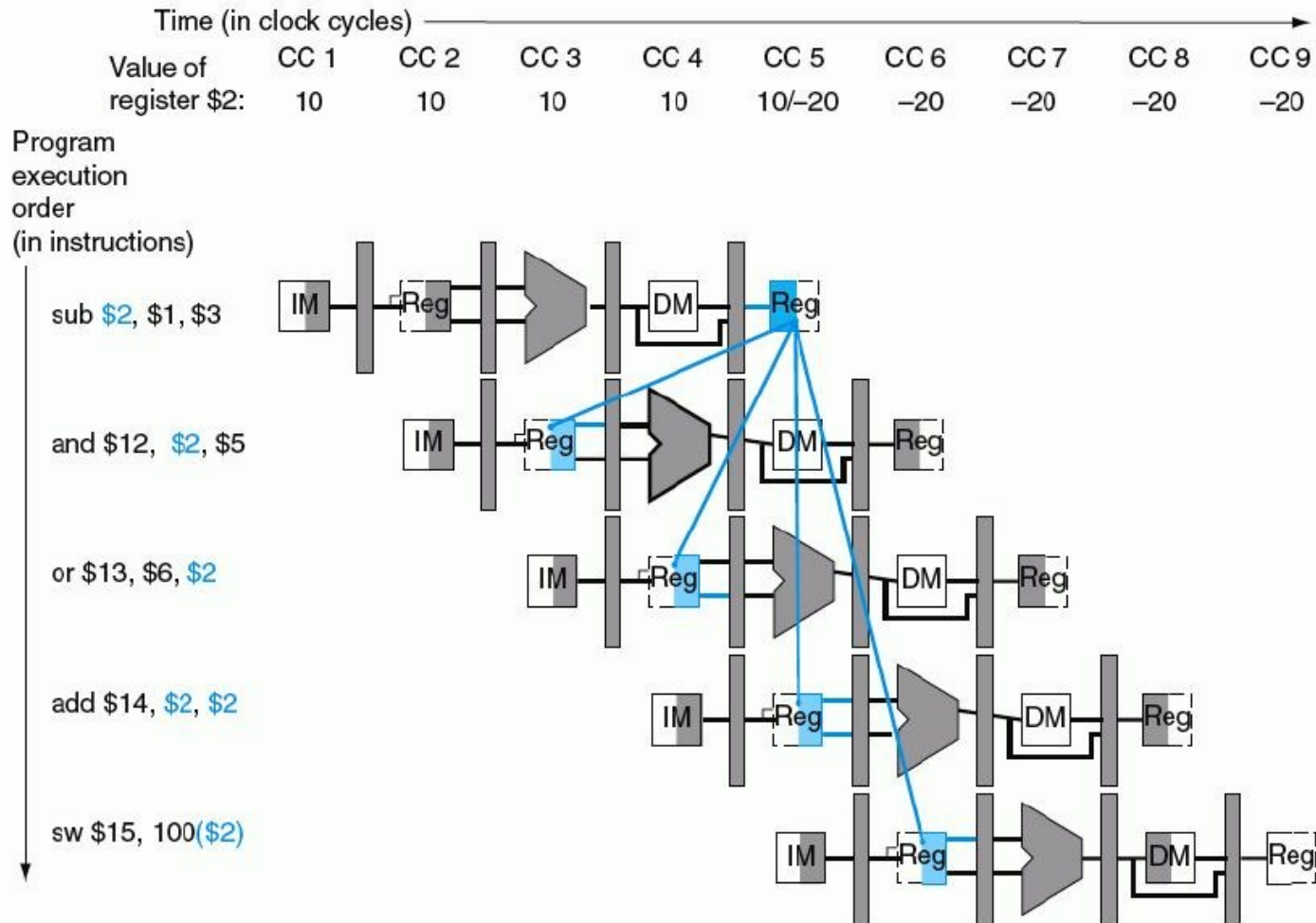
Pipeline de contrôle : ajout de registres de transmission/décalage des signaux de contrôle (issus de ID)

Pipeline : contrôle et chemin de données



Pipeline de contrôle : acheminement des signaux de contrôle pipelinés

Dépendances de pipeline



Aléa de données : a priori les instructions 2 à 5 doivent attendre le WB de l'instruction 1

Pipeline : transmission anticipée

Program
execution
order
(in instructions)

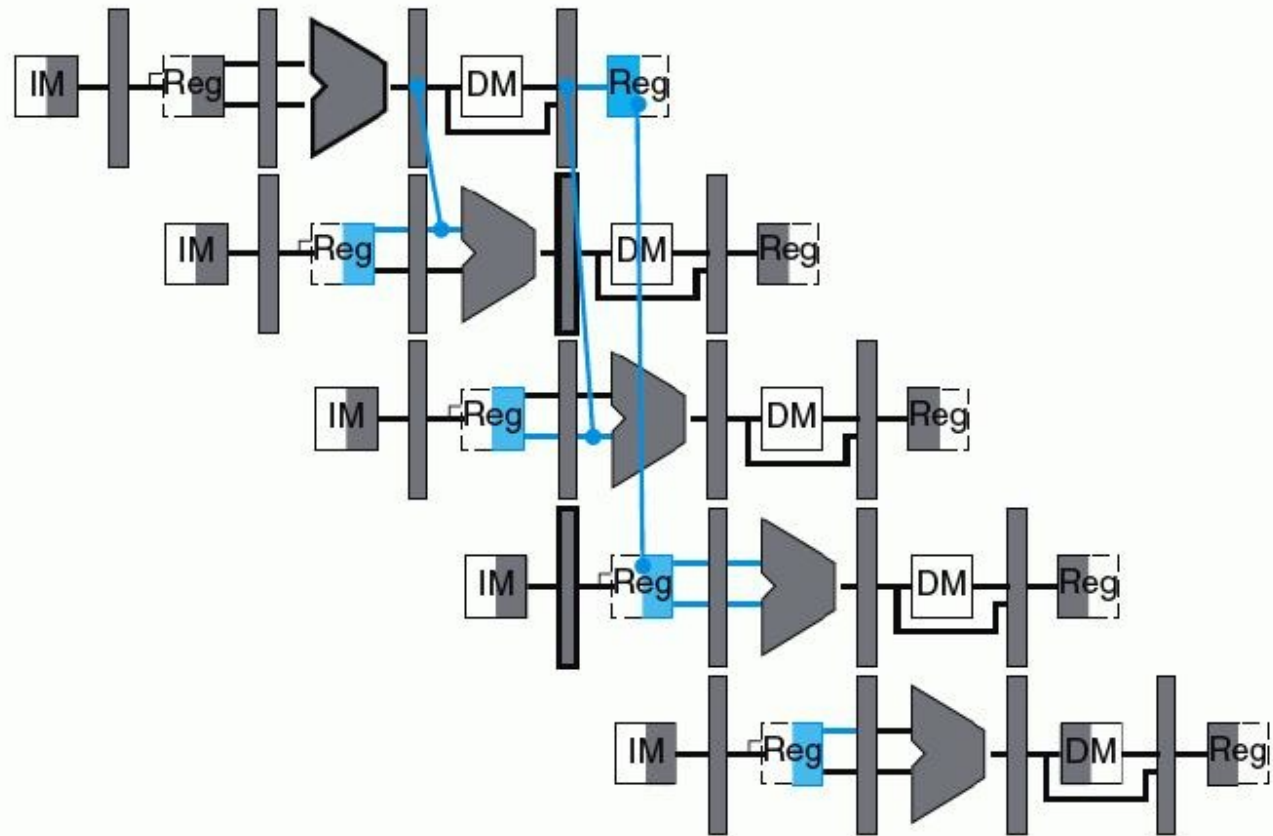
sub \$2, \$1, \$3

and \$12, \$2, \$5

or \$13, \$6, \$2

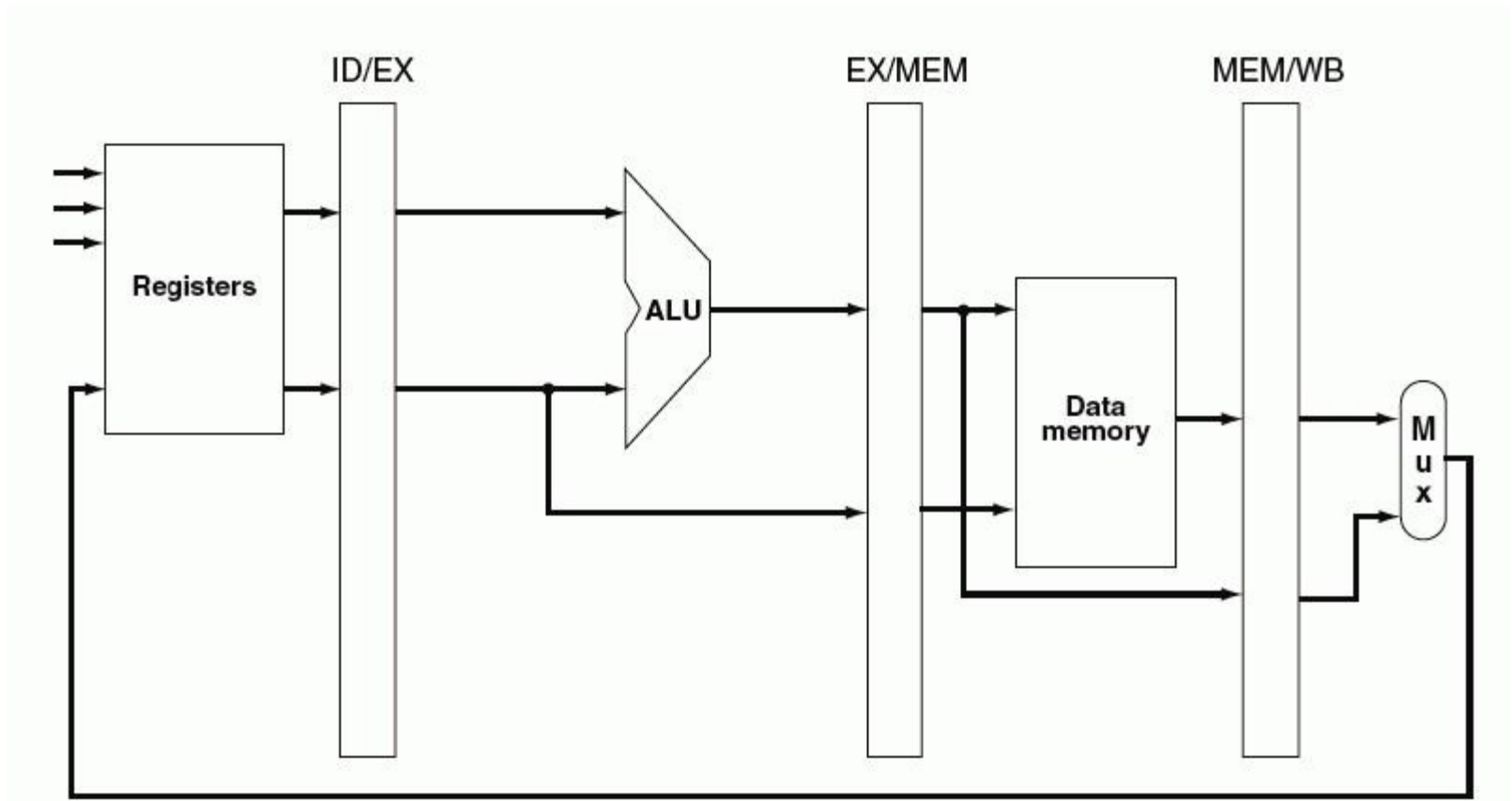
add \$14, \$2, \$2

sw \$15, 100(\$2)

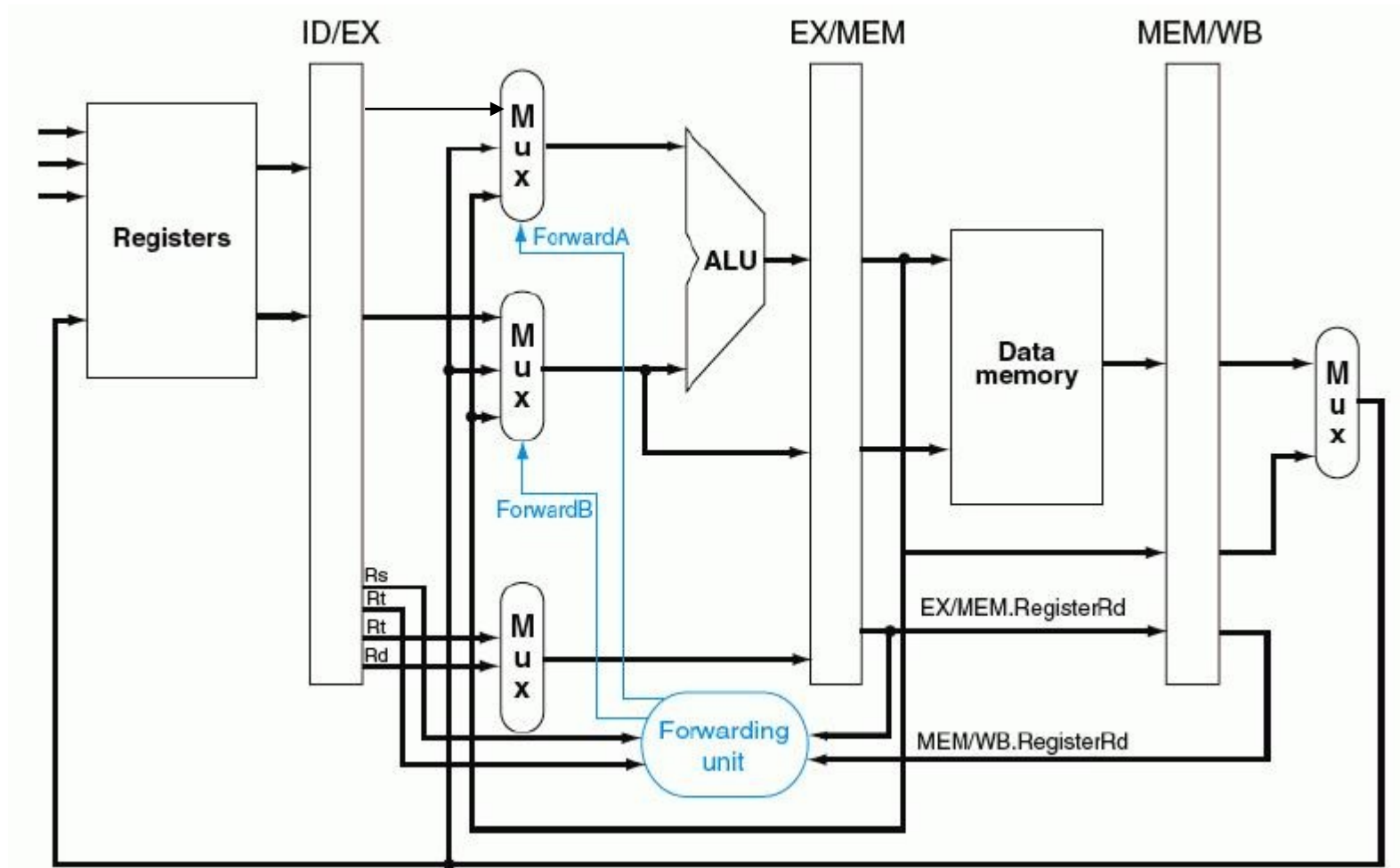


Transmission anticipée : à partir des étages EX, MEM ou en court-circuitant la banque de registres

Pipeline sans transmission

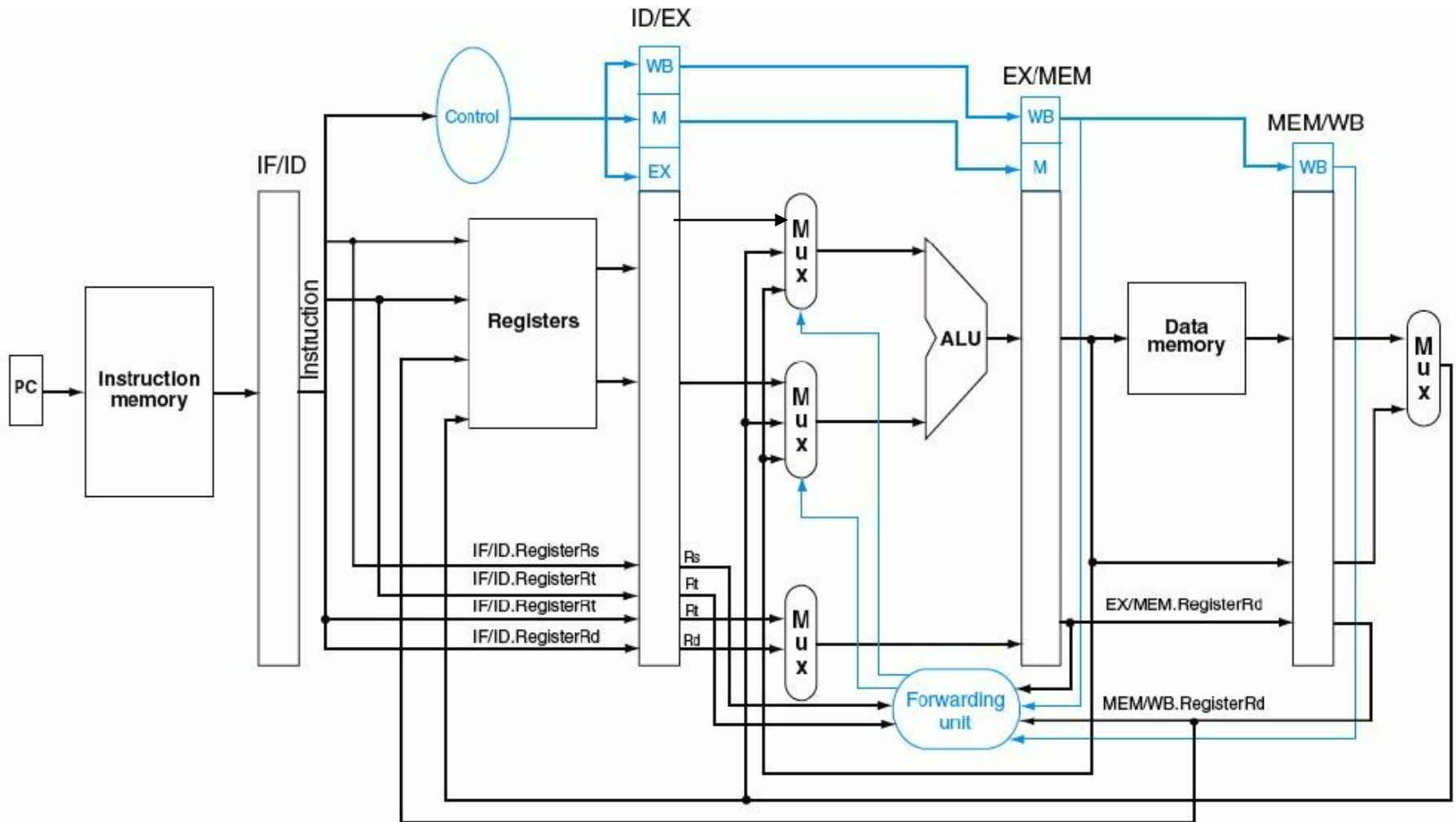


Pipeline avec unité de transmission

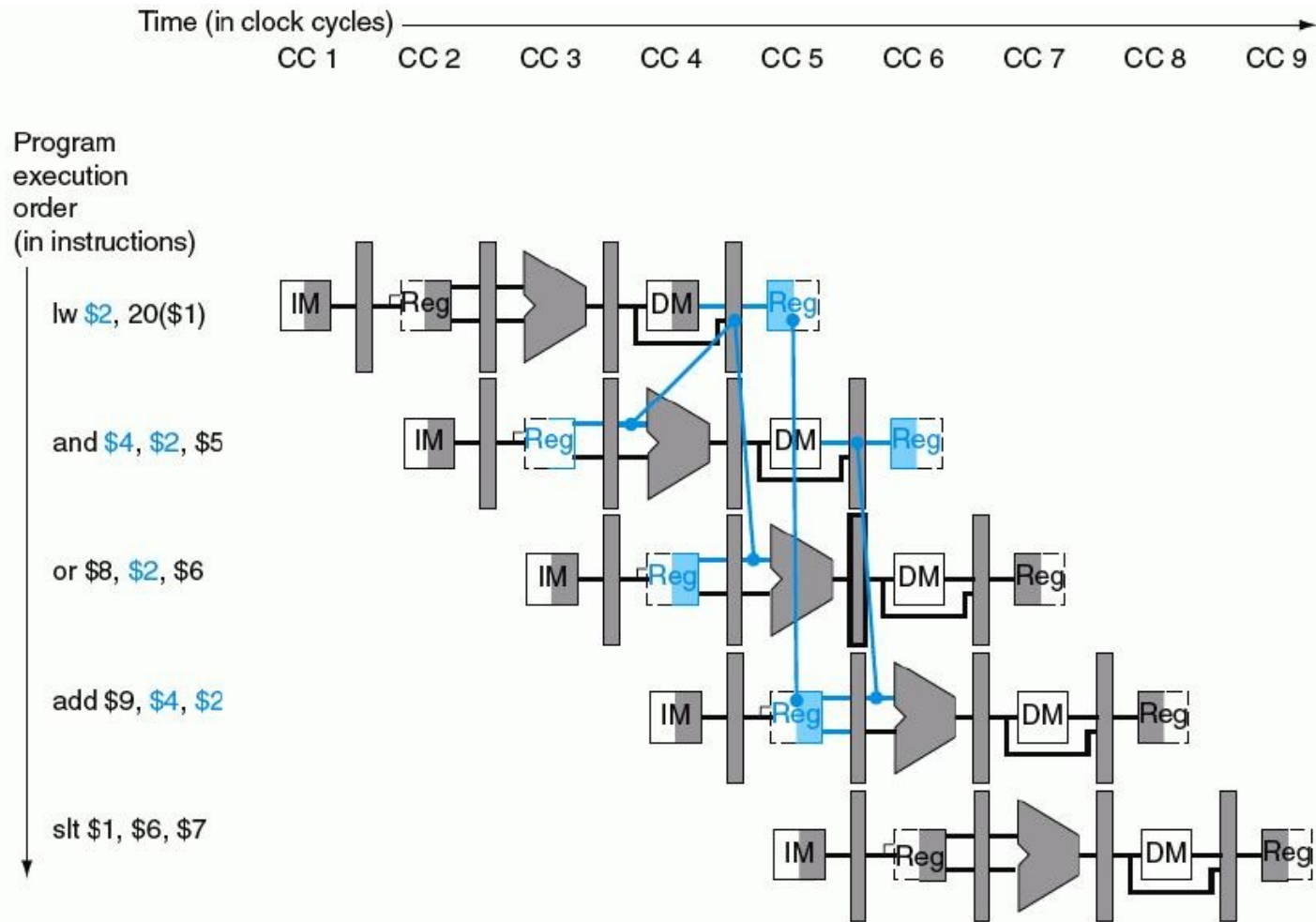


Unité de transmission anticipée : contrôle en fonction des numéros de registres

Pipeline : chemin de données complété

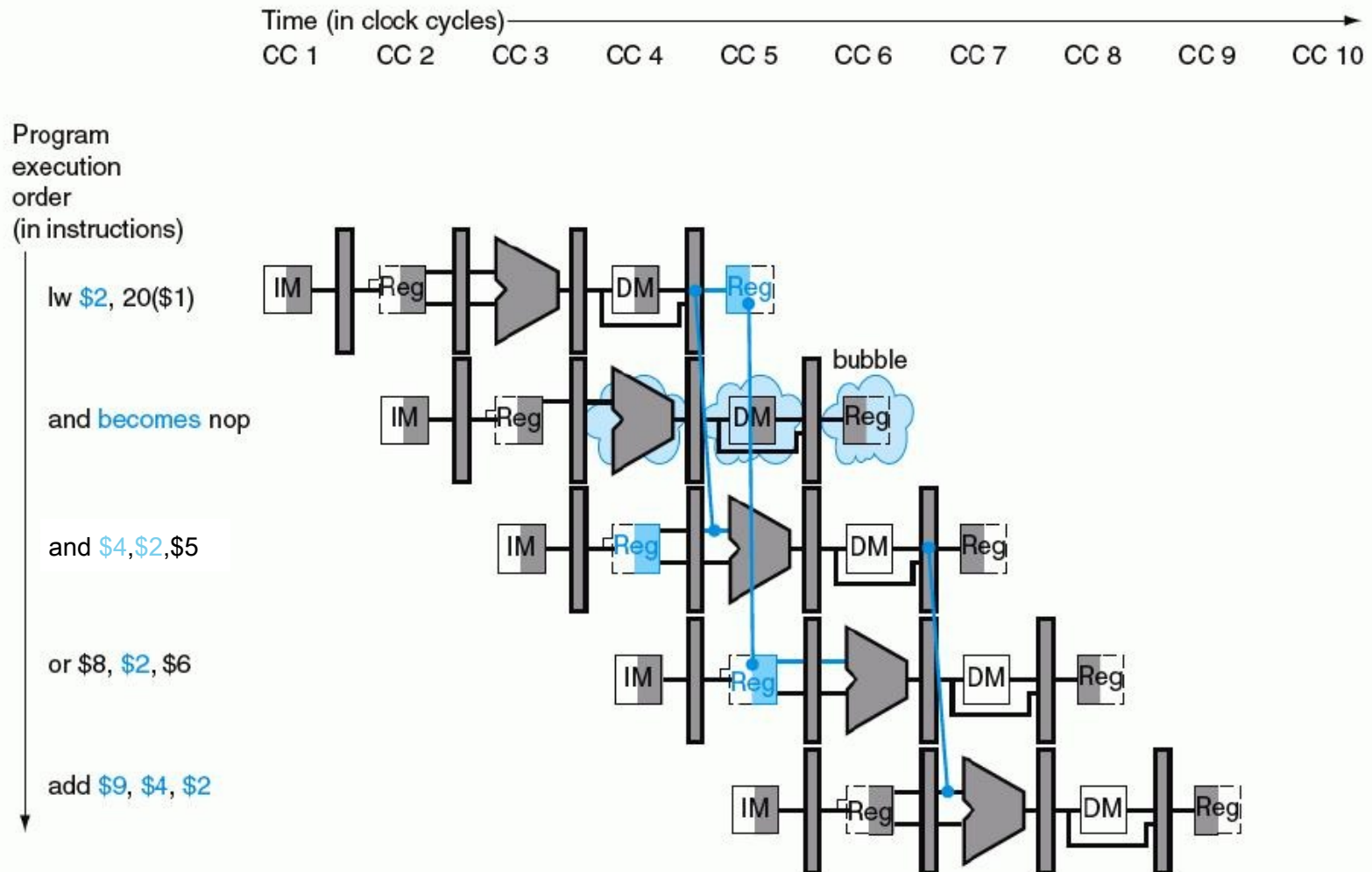


Aléas de données et rupture de pipeline

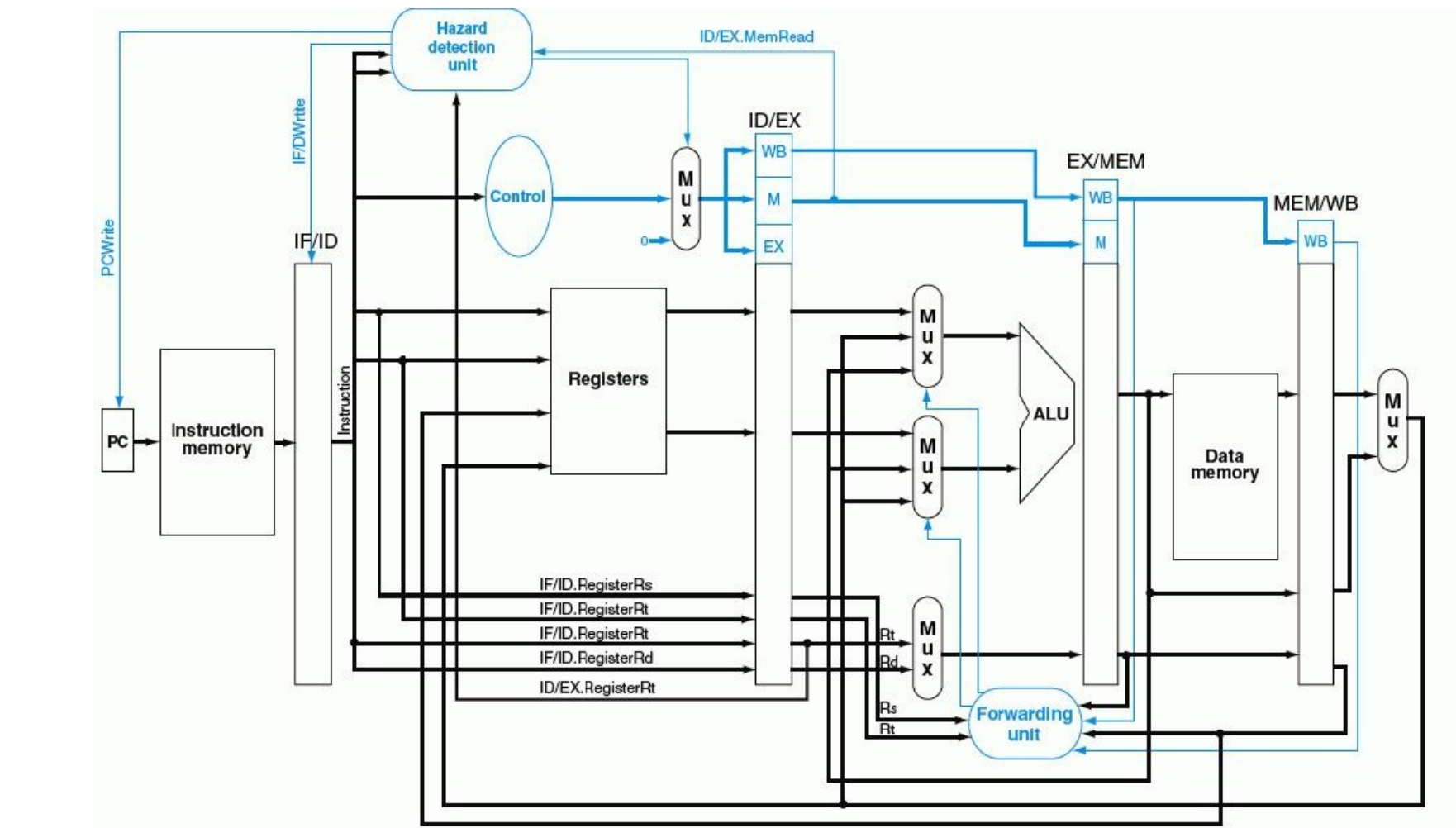


Transmission anticipée insuffisante en cas d'aléa de données dû à une lecture mémoire

Blocages insérés dans le pipeline

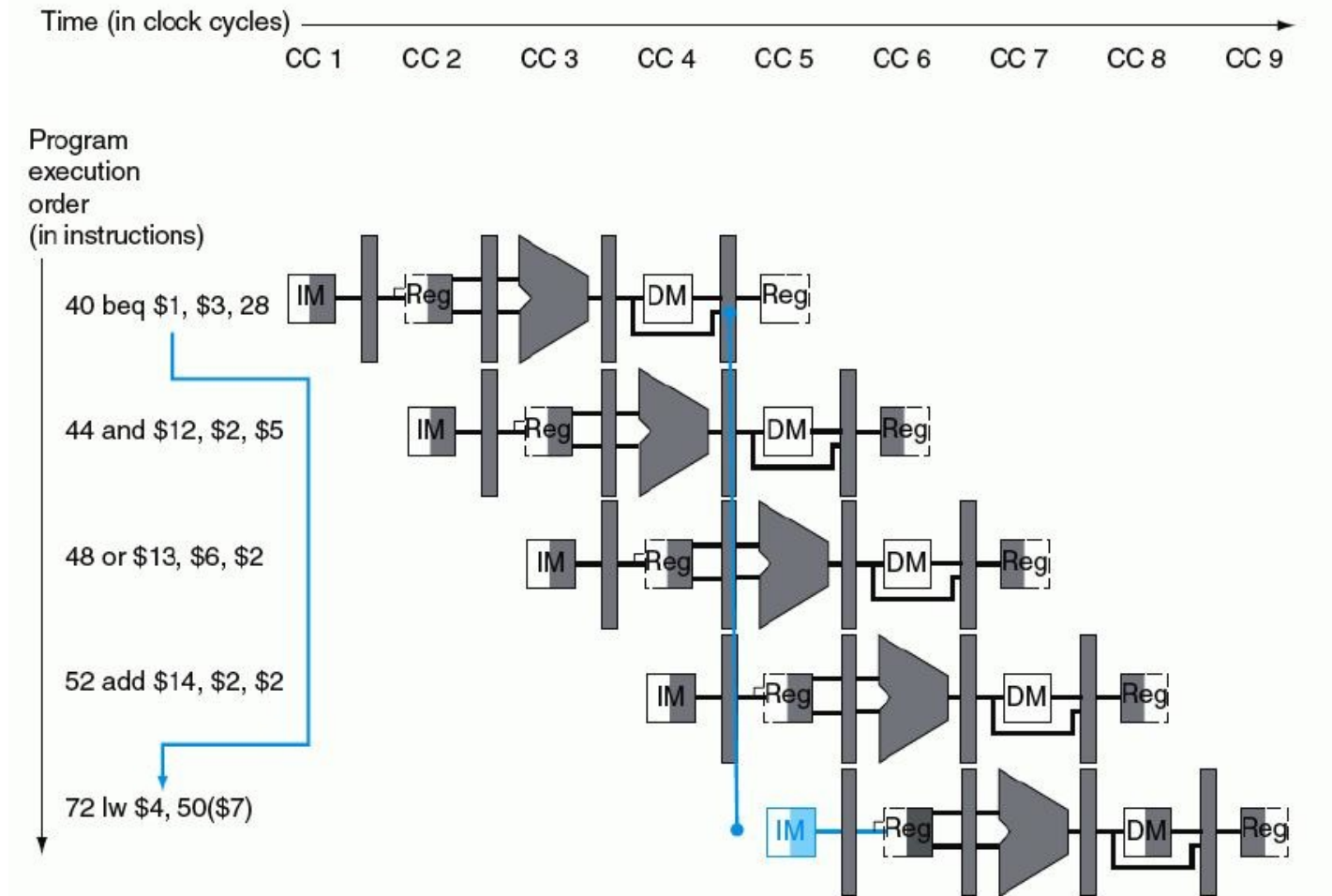


© 2013 Pearson Education, Inc. or its affiliate(s). All rights reserved. 10



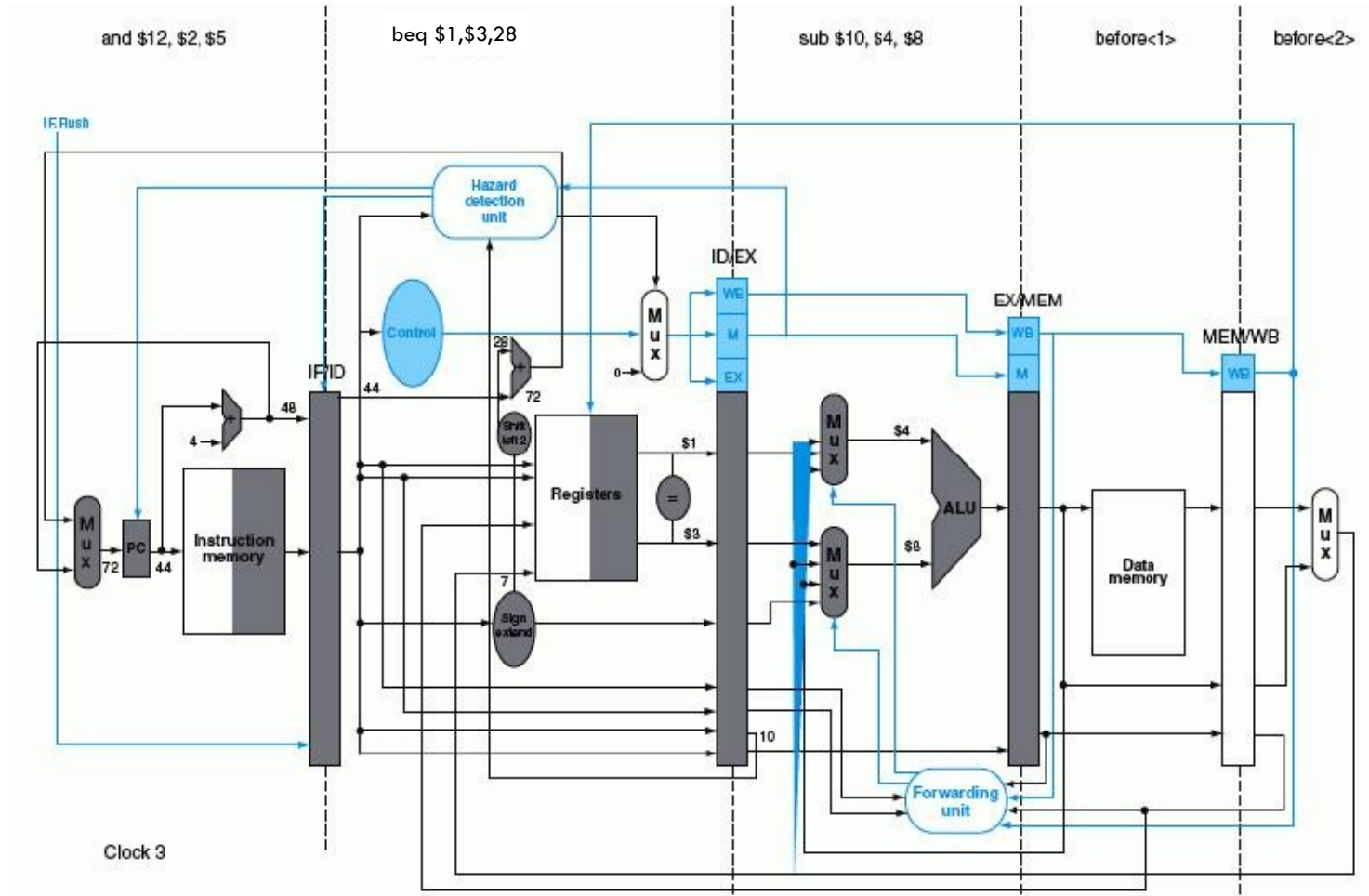
Pipeline bloqué : unité de détection d'aléa empêche l'incrémentation de PC et l'écriture sur IF/ID, et met à zéro tous les signaux de contrôle à propager

Aléas de branchement



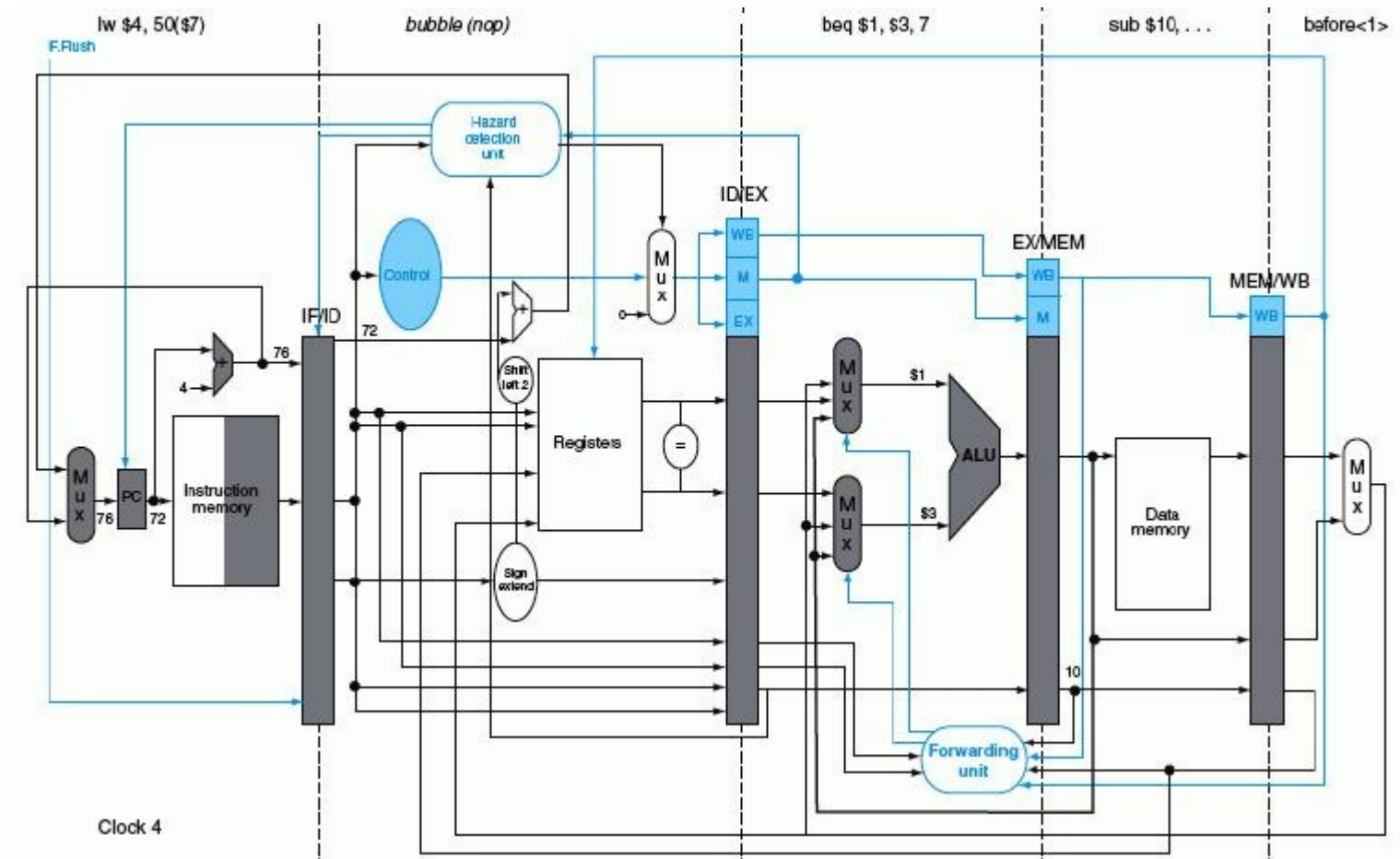
Branchement pris non prédit : trois instructions démarrent « à tort » leur exécution

Pipelined : chemin de données



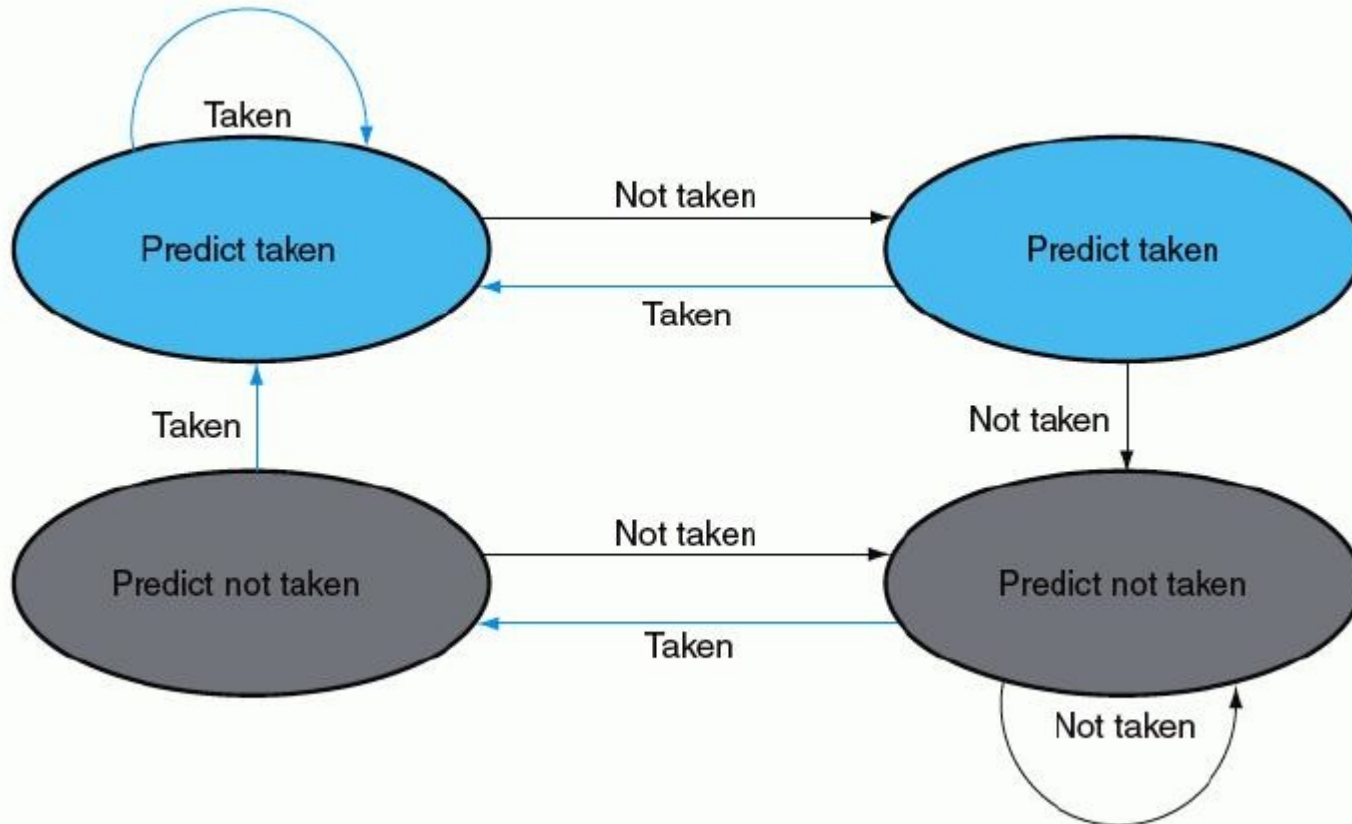
Branchement pris pr dit : l' tage ID bloque le pipeline ...

Pipelined : chemin de données



Branchement pris prédit : ... et l'adresse de branchement est préparée en un cycle

Pipeline : prédiction dynamique



Exemple de prédicteur dynamique (4 états) : prédiction forte, faible, inversion de prédiction après deux contradictions

En cas de mauvaise prédiction : détection erreur après étape EX, pipeline bloqué de façon à éviter les étapes MEM et WB sur les instructions exécutées à tort