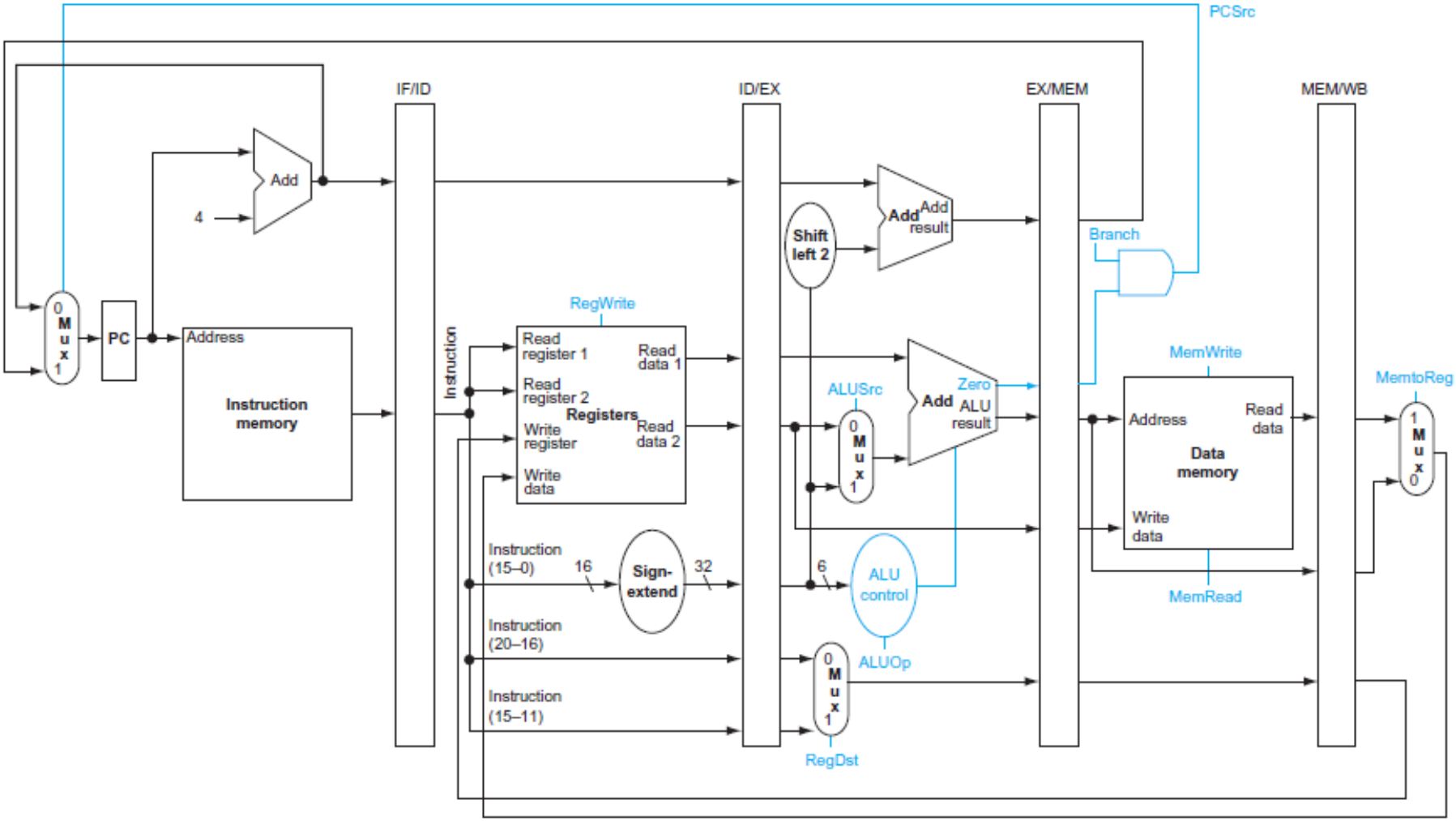
Pipeline - Banda de asamblare (continuare)

– Curs 12–


PCSrc
IF/ID
ID/EX
EX/MEM
MEM/WB
Add
4
M u x
0
1
PC
Address
Instruction memory
Instruction
RegWrite
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
Shift left 2
Add
Add result
Branch
ALUSrc
Zero
ALU result
MemWrite
MemtoReg
Address
Data memory
Read data
Write data
MemRead
Instruction (15–0)
16
Sign-extend
32
6
ALU control
ALUOp
Instruction (20–16)
Instruction (15–11)
RegDst

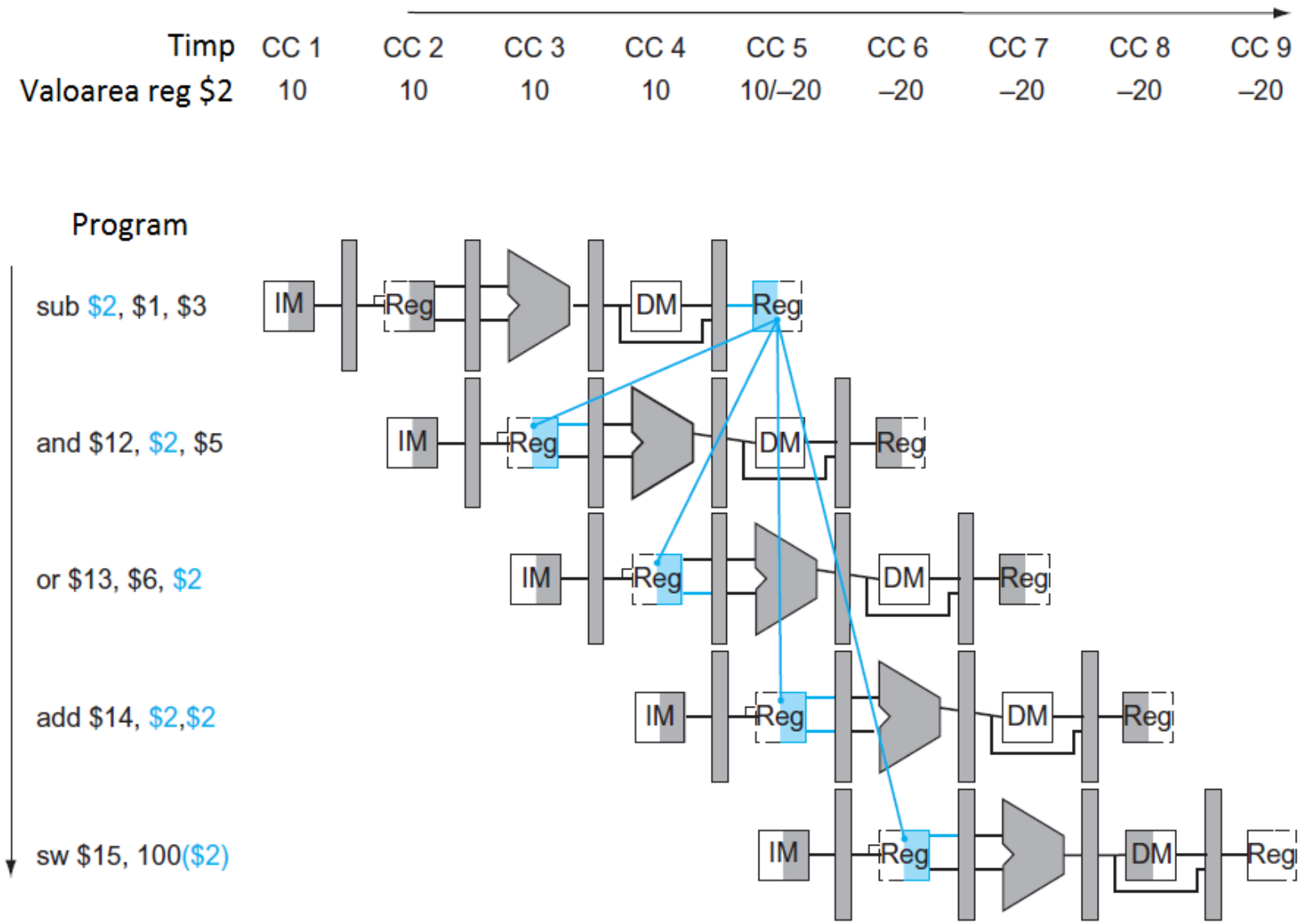
Hazardele de date: Forwarding vs. Stalling

Hazardele sunt obstacole în calea execuției pipeline.

Exemplu:

```
sub $2, $1, $3
and $12, $2, $5
or $13, $6, $2
add $14, $2, $2
sw $15, 100($2)
```

Ultimele 4 instrucțiuni depind de valoarea registrului $2 obținută în prima instrucțiune.



Liniile albastre care se “intorc in timp” sunt hazarde de date.

Forwarding

- ✓ Plecăm spre exemplu de la o operație aflată în stagiul EX – operație ALU sau calculul unei adrese effective.
- ✓ Avem nevoie de valorile de intrare pentru ALU
- ✓ Introducem notația ID/Ex.RegRs pentru a ne referi la nr. registrului a cărei valoare este găsită în registrul pipeline ID/EX. Este val. de la primul port citit al registrelor generale.

Condițiile de detecție

1a : EX/MEM.RegRD = ID/EX.RegRS

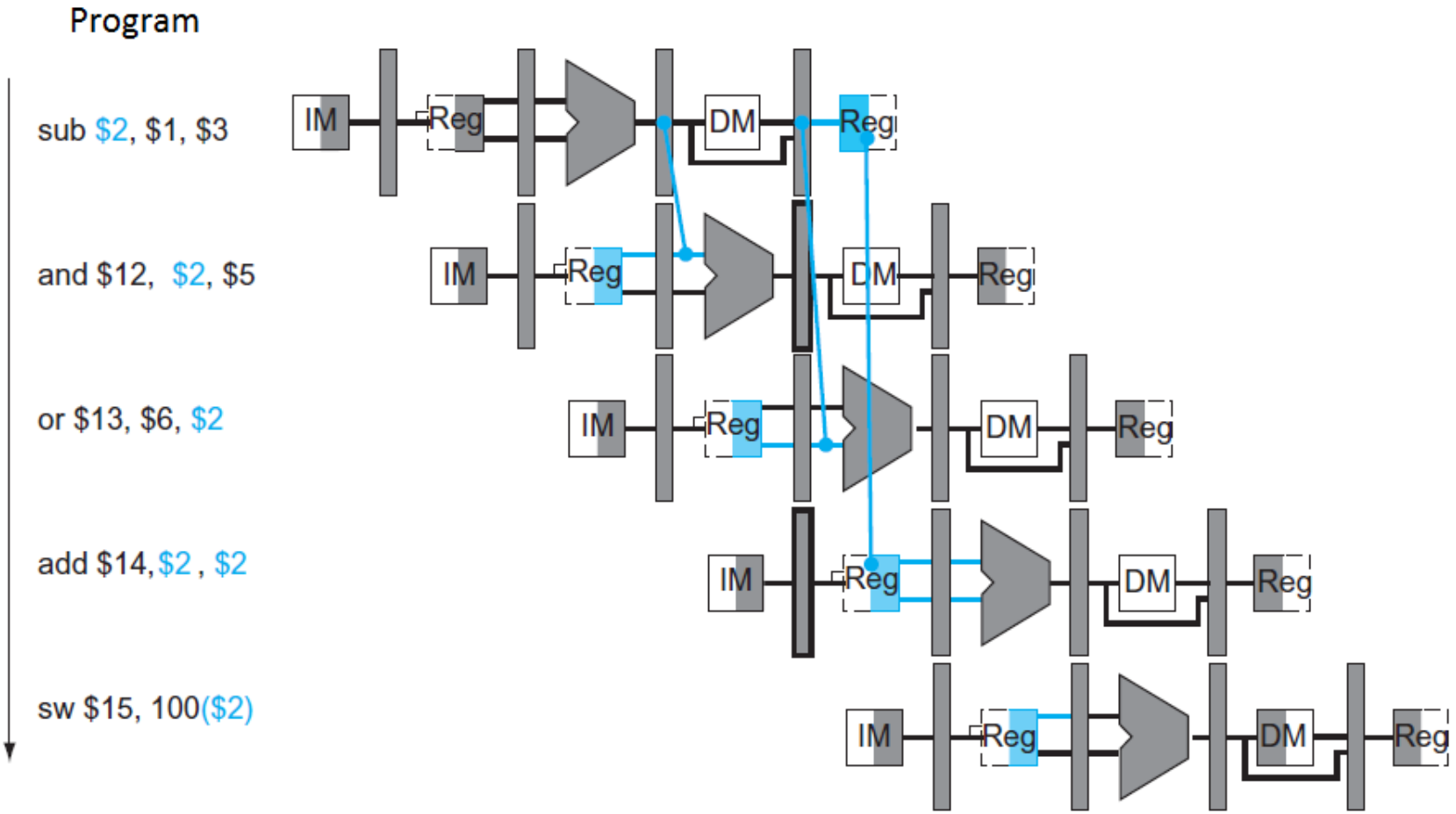
1b : EX/MEM.RegRD = ID/EX.RegRT

2a : MEM/WB.RegRD = ID/EX.RegRS

2b : MEM/WB.RegRD = ID/EX.RegRT

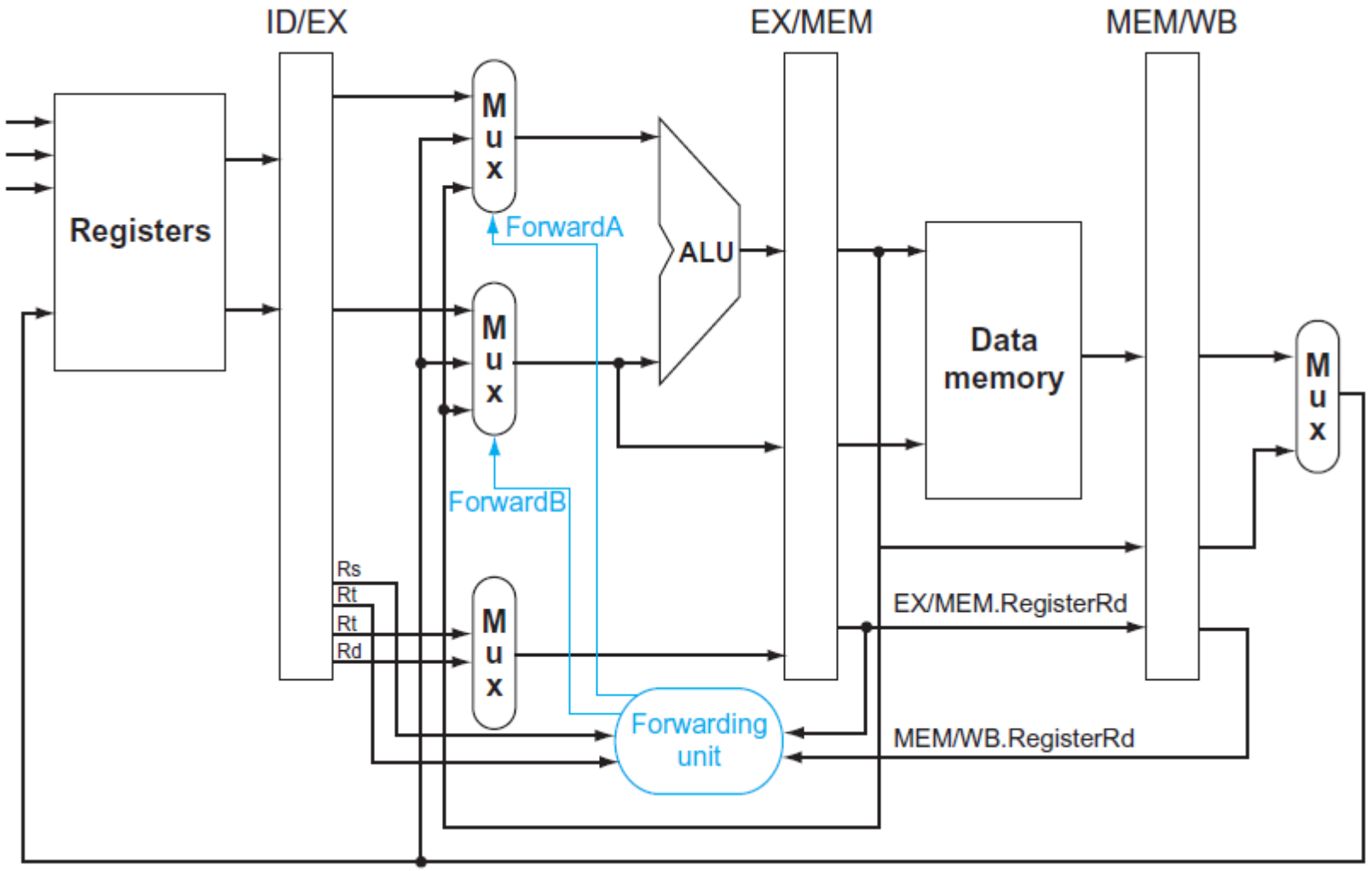
Apar două perechi de condiții de detecție pentru hazarde.

	Timp								
	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7	CC 8	CC 9
Valoare registru $2	10	10	10	10	10/–20	–20	–20	–20	–20
Valoare EX/MEM:	X	X	X	–20	X	X	X	X	X
Valoare MEM/WB:	X	X	X	X	–20	X	X	X	X



Forwarding pentru valorile aflate în registrele din pipeline.

Varianta cu detecție de hazard



Adăugând multiplexoare și activând semnalele de control corespunzătoare, putem rula pipe-ul la viteza maximă, cu dependențele de date semnalate anterior.

Hazard EX

•If (EX/MEM.RegWrite

and (EX/MEM.RegisterRd <>0)

and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
ForwardA = 10

•If (EX/MEM.RegWrite

and (EX/MEM.RegisterRd <>0)

and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
ForwardB = 10

Hazard MEM

•If (MEM/WB.RegWrite

and (MEM/WB.RegisterRd <>0)

and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
ForwardA = 01

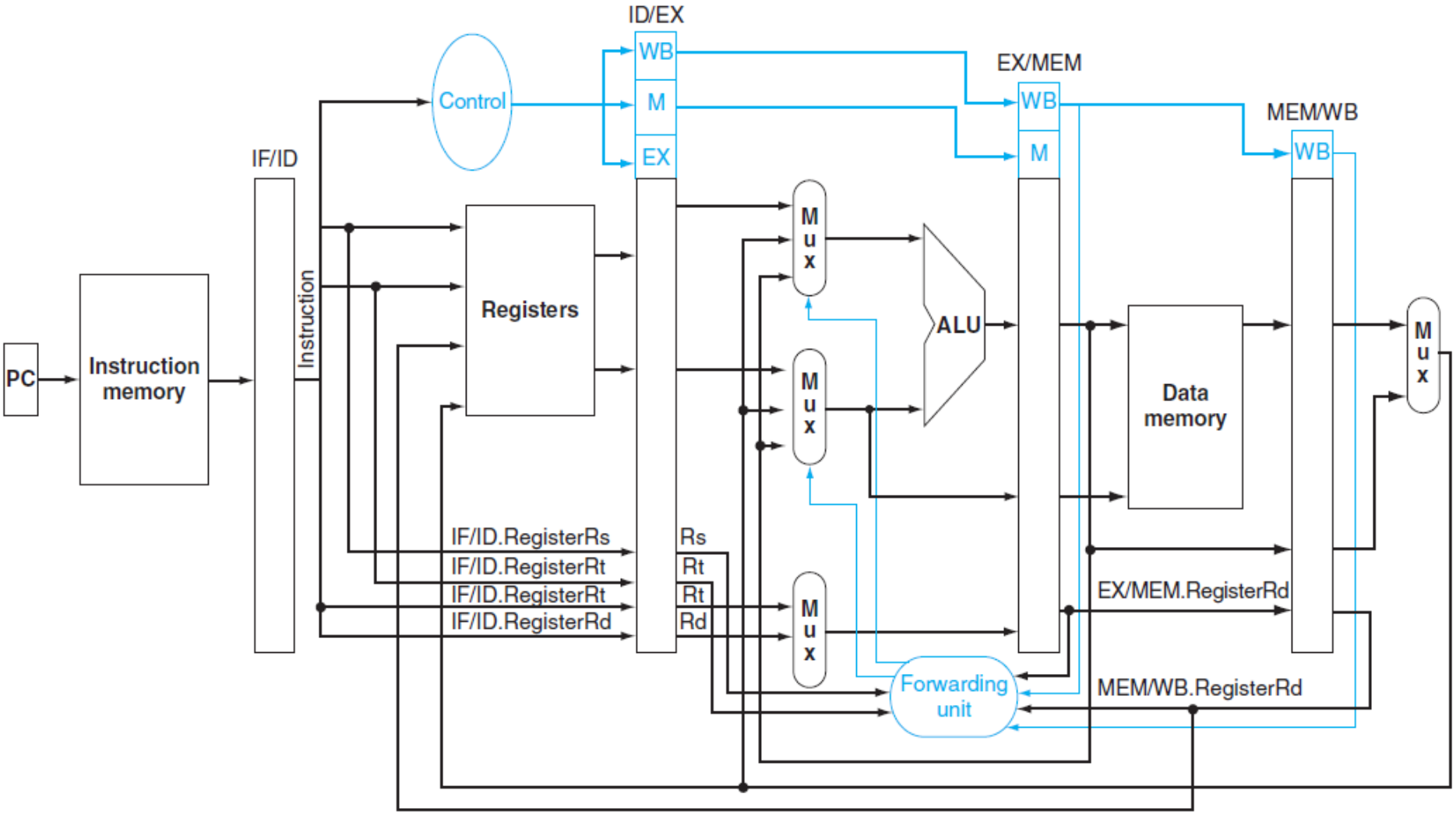
•If (MEM/WB.RegWrite

and (MEM/WB.RegisterRd <>0)

and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
ForwardB = 01

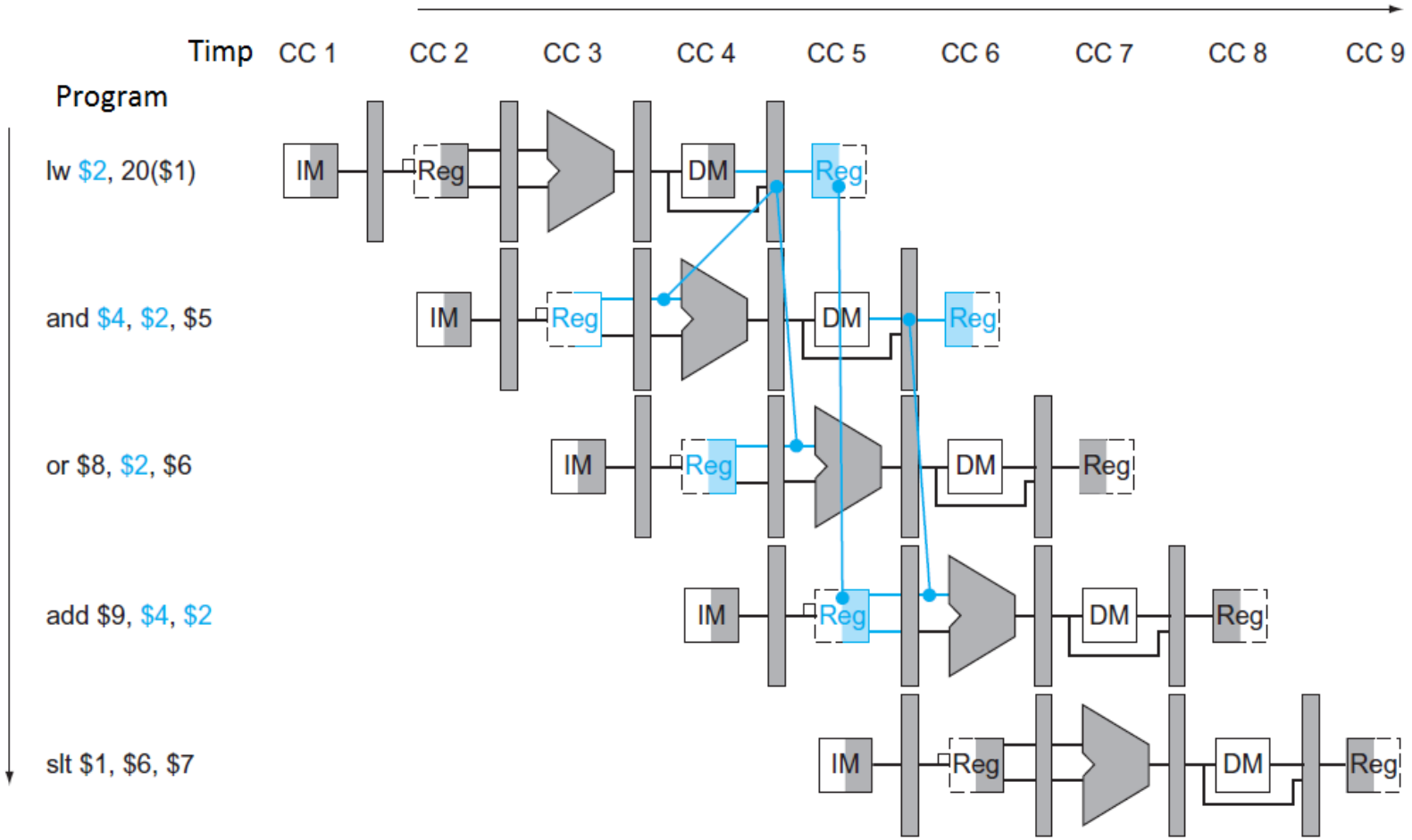
Controlul multiplexoarelor de forwarding

Control MUX	Sursa	Explicație
ForwardA = 00	ID/EX	Primul operand ALU vine din fișierul de registre
ForwardA = 01	MEM/WB	Primul operand ALU este forward-at din memoria de date sau de la un rezultat ALU anterior.
ForwardA = 10	Ex/MEM	Primul operand ALU este forward-at de la rezultatul ALU calculat anterior.
ForwardB = 00	ID/EX	Al doilea operand ALU vine din fișierul de registre
ForwardB = 01	MEM/WB	Al doilea operand ALU este forward-at din memoria de date sau de la un rezultat ALU anterior.
ForwardB = 10	Ex/MEM	Al doilea operand ALU este forward-at de la rezultatul ALU calculat anterior.


ID/EX
EX/MEM
MEM/WB
WB
M
EX
Control
IF/ID
PC
Instruction memory
Instruction
Registers
ALU
Data memory
Mux
IF/ID.RegisterRs
IF/ID.RegisterRt
IF/ID.RegisterRt
IF/ID.RegisterRd
Rs
Rt
Rt
Rd
EX/MEM.RegisterRd
MEM/WB.RegisterRd
Forwarding unit

Introducerea stall-urilor (nop-urilor)

Dacă o instrucțiune încearcă să citească un registru imediat după o instrucțiune de încărcare ce scrie în același registru, pipe-ul trebuie să rămână într-o stare stall (bubble), apelarea la forwarding nerezolvând complet problema.



Controlul pt. detecția hazardului în cazul instrucțiunii de încărcare:

if (ID/EX.MemRead and

((ID/EX.RegisterRt = IF/ID.RegisterRs) or

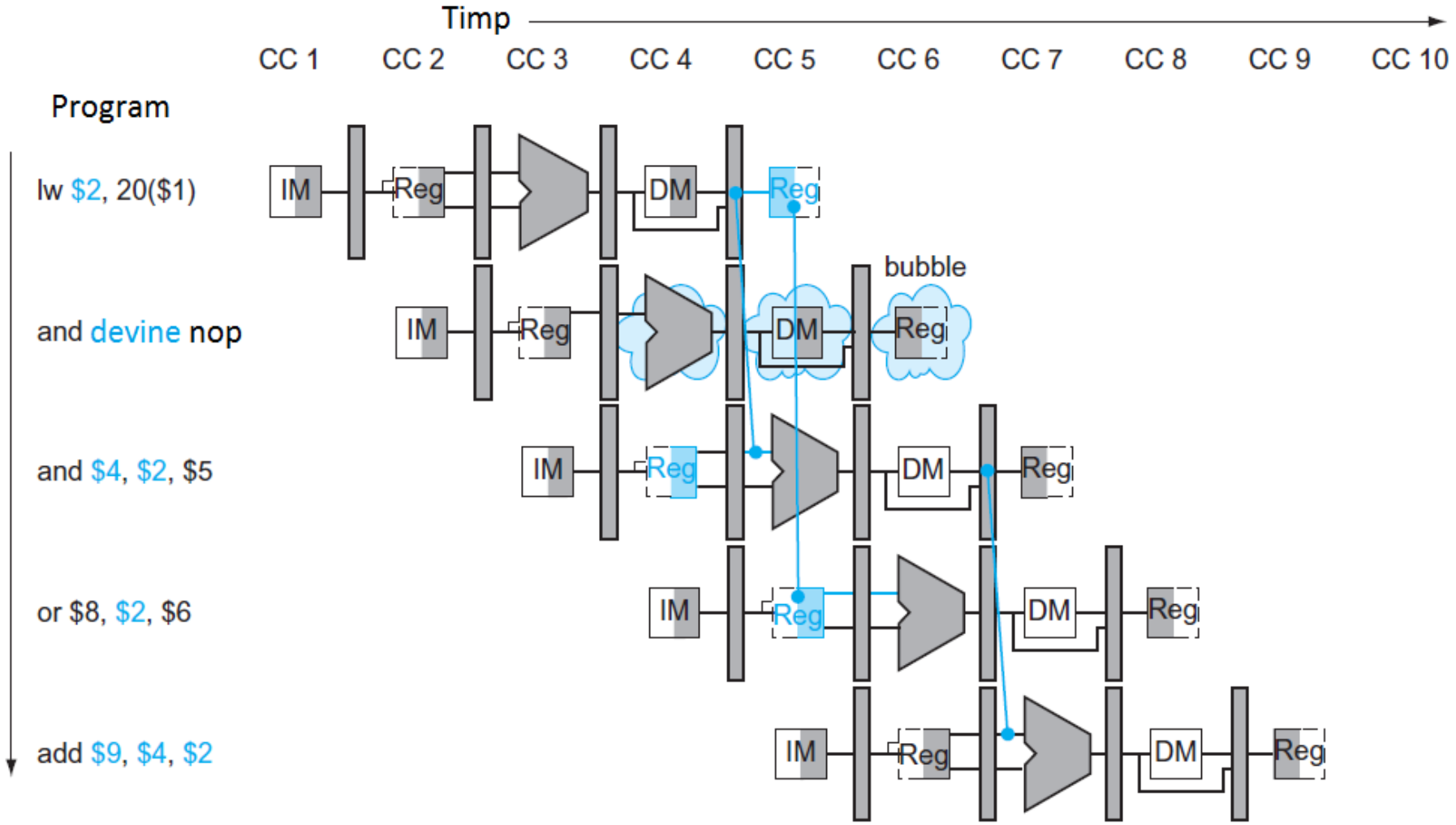
(ID/EX.RegisterRt = IF/ID.RegisterRt)))

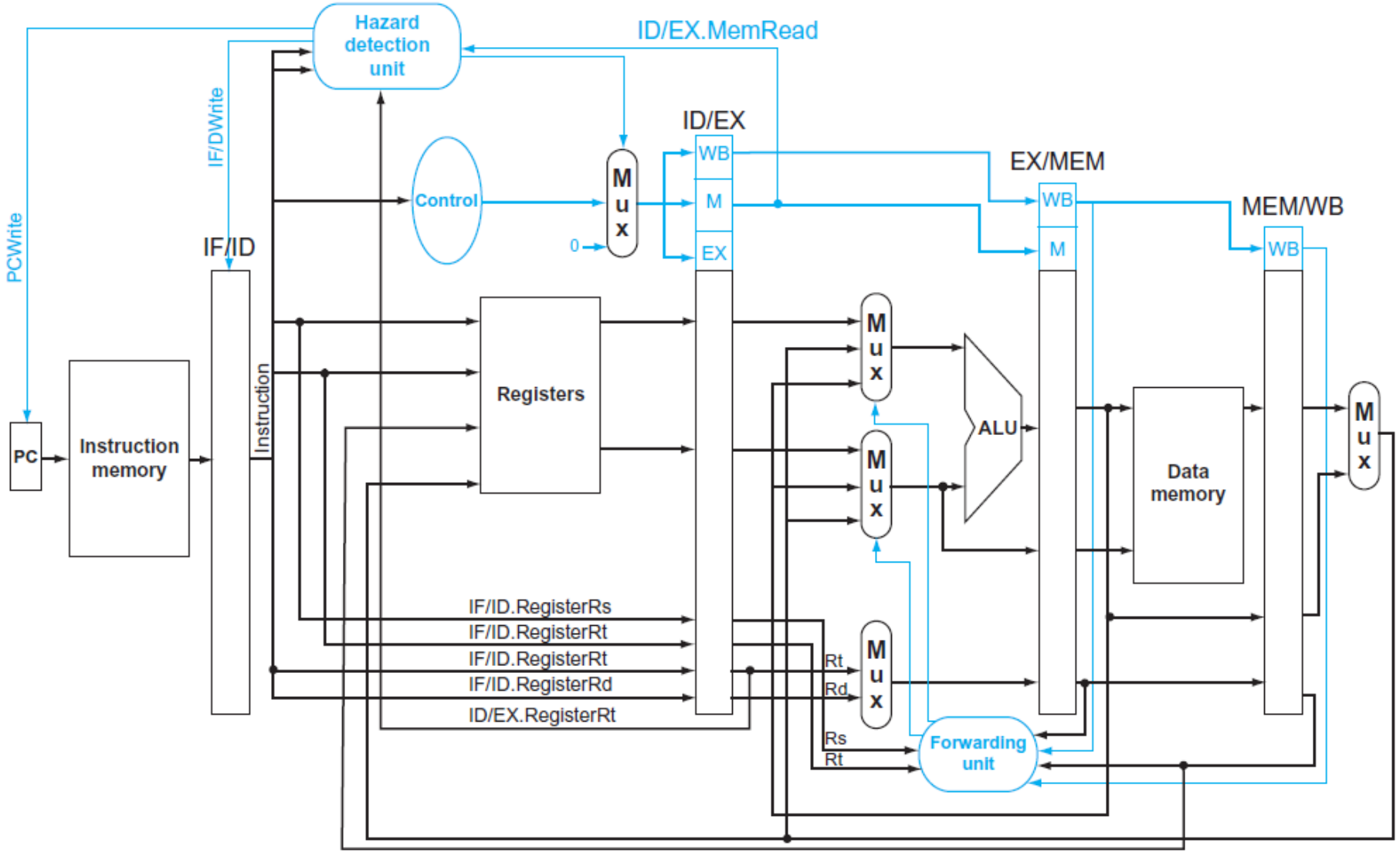
stall pipeline

Testează dacă intrucțiunea este de încărcare – singura care citește memoria de date

Verifică dacă valoarea câmpului registrului destinație pentru încărcare din stagiul EX este vreun registru sursă al instrucțiunii din stagiul ID

Dacă o parte a pipe-ului, începând cu stagiul EX, execută acțiuni care nu au efect asupra instrucțiunii curente, acest lucru se numește ***nop***. Ele se comportă la fel ca bubbles.





Adăugarea unității pentru detectarea hazardelor

Probleme

1. Specificați dacă existe hazarduri în următoarele secvențe de cod:

```
i1:  lw   r1, 0(r2)
i2:  sub  r4, r1, r5
i3:  and  r6, r1, r7
i4:  or   r8, r1, r9
```

2.

Eliminați hazardurile din următoarele secvențe de cod:

P11: lw $1, 40($6)
P12: add $6, $2, $2
P13: sw $6, 50($1)

P21: lw $5, -16($5)
P22: sw $5, -16($5)
P23: add $5, $5, $5

3.

Se consideră următoarea secvență de cod:

```
lw      $1, 40($6)
add     $5, $5, $5
```

a) În cadrul execuției acestor instrucțiuni ce se menține în fiecare registru de pipe?

b) Care registre sunt necesare citirii și care sunt citite efectiv ?

c) Ce se realizează în cadrul stagiilor EX și MEM ?