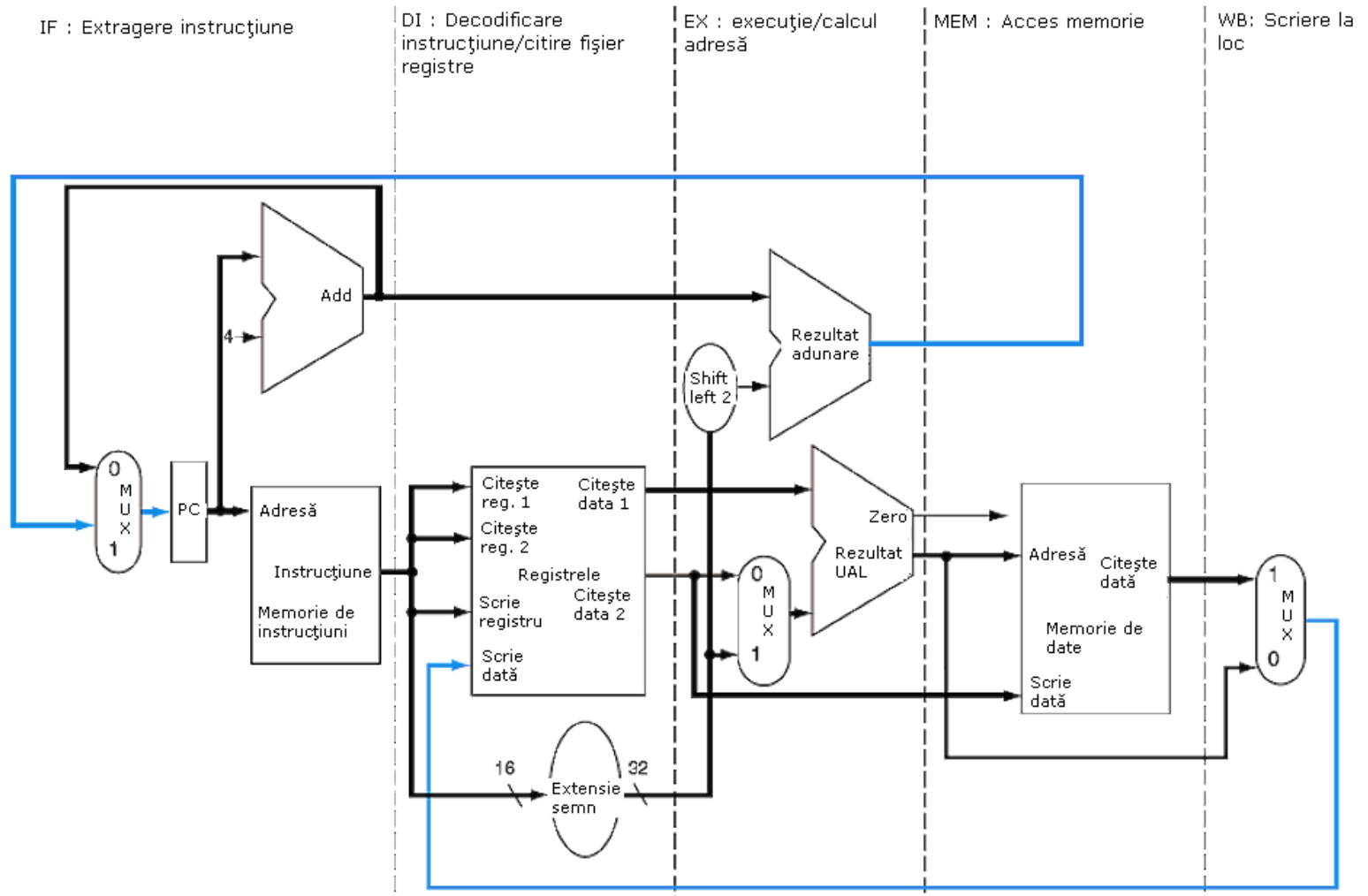
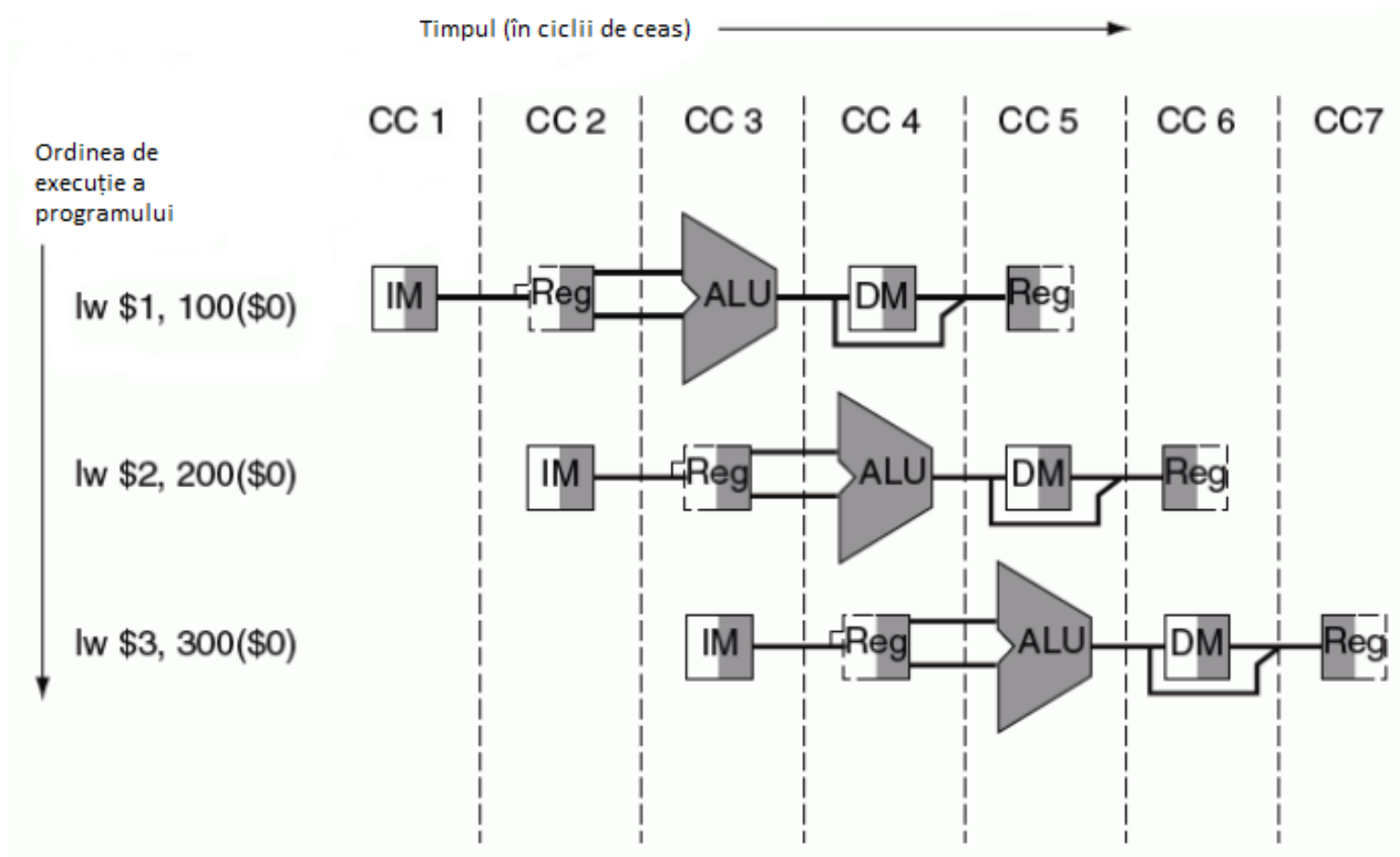


Calea de date pipeline
- continuare -

Calea de date cu un singur ciclu de ceas



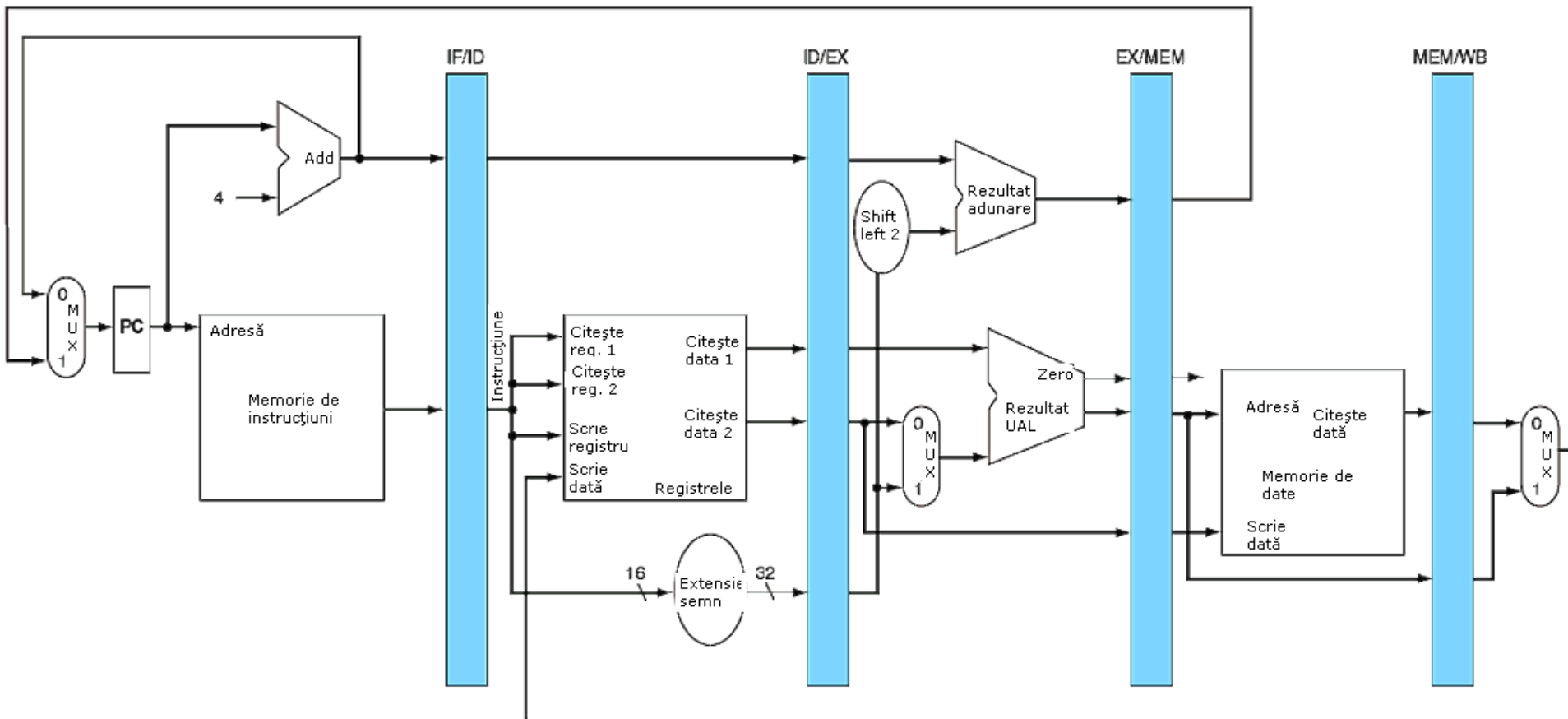
Excepții: actualizarea PC-ului și pasul de rescriere



Fiecare instrucțiune are propria sa cale de date;

MI = memoria de instrucțiuni + PC-ul din etapa extragerii instrucțiunii

REG = fișierul de registre + unitatea de extindere a semnului din etapa DI

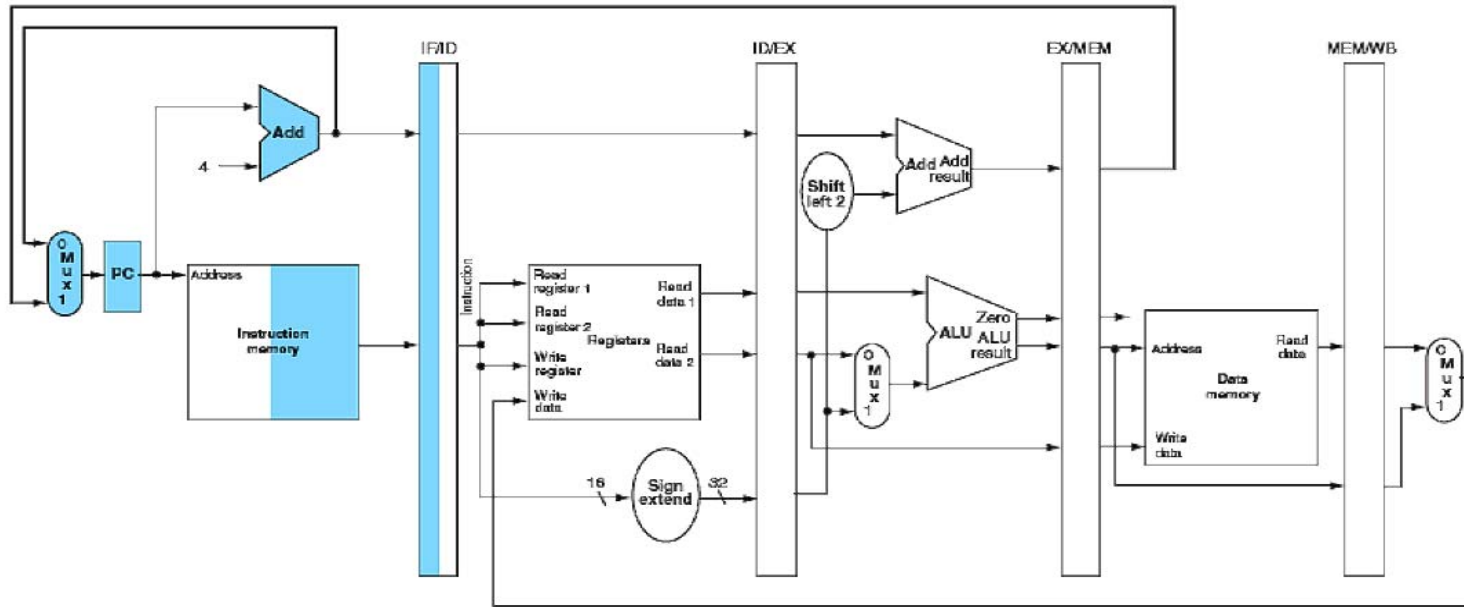


Registre pipeline separă fiecare etapă de pipe

EI/DI – separă etapele de extragere a instrucțiunii și de decodificare a instrucțiunii

NU EXISTĂ REGISTRE LA SFÂRȘITUL ETAPEI DE RESCRIERE

Fig. 1



ÎN CĂR C A R E A

Extragerea instrucțiunii – instrucțiunea este citită din memorie (folosind PC) și este pusă în registrul EI/DI

Adresa lui PC este incrementată cu 4 și rescrisă în PC. Această adresă este salvată în EI/DI în vederea folosirii ei ulterioare – spre exp. urmează instrucțiunea **beq**.

Calculatorul NU știe ce instrucțiune este extrasă.

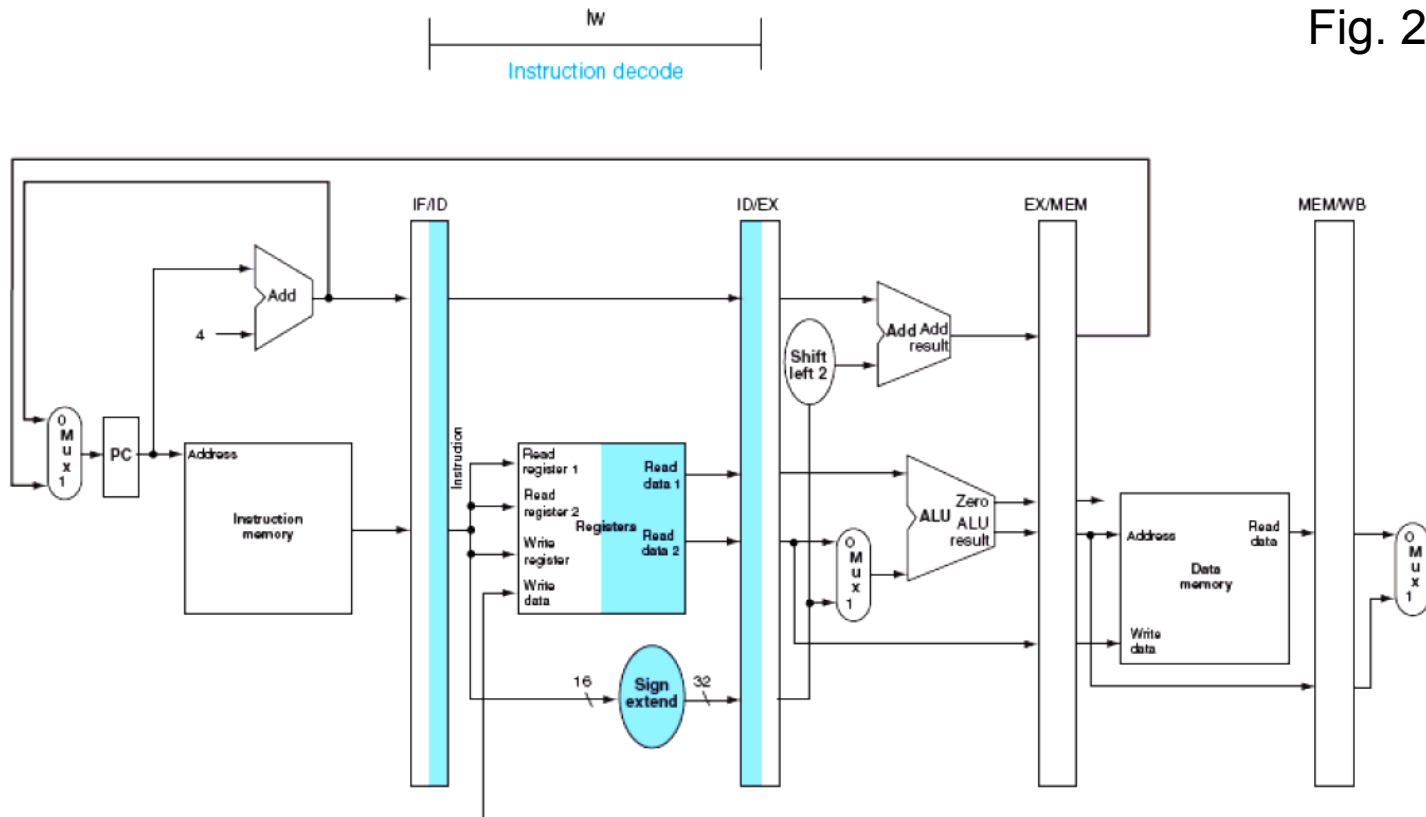


Fig. 2

ÎN CĂRCAREA

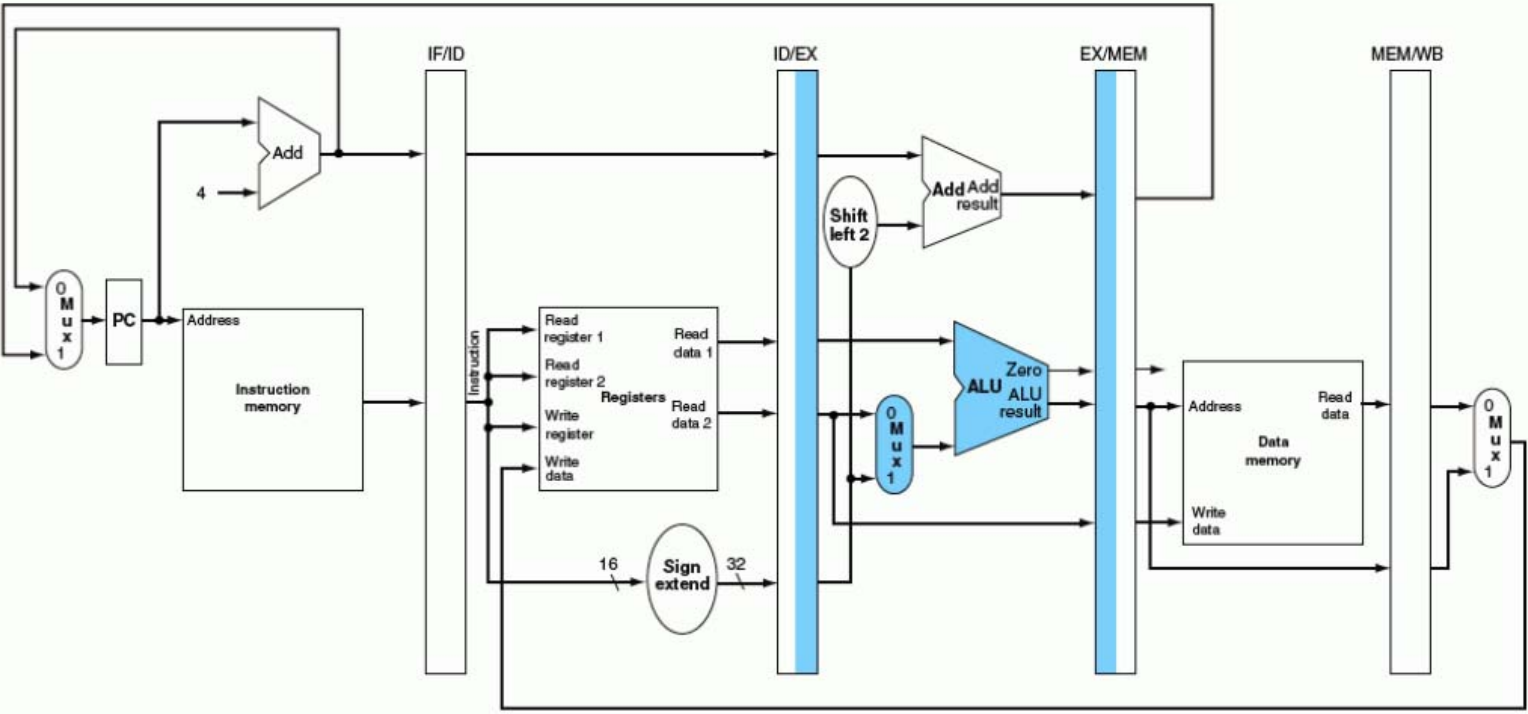
Decodificarea instrucțiunii – registrul EI/DI furnizează câmpul imediat de 16 biți căruia i se extinde semnul până la 32 de biți, și numerele registrelor pentru citirea celor două registre.

Aceste valori împreună cu adresa PC incrementată se regăsesc în DI/EX

Sunt necesari toți cei 3 operanzi ?

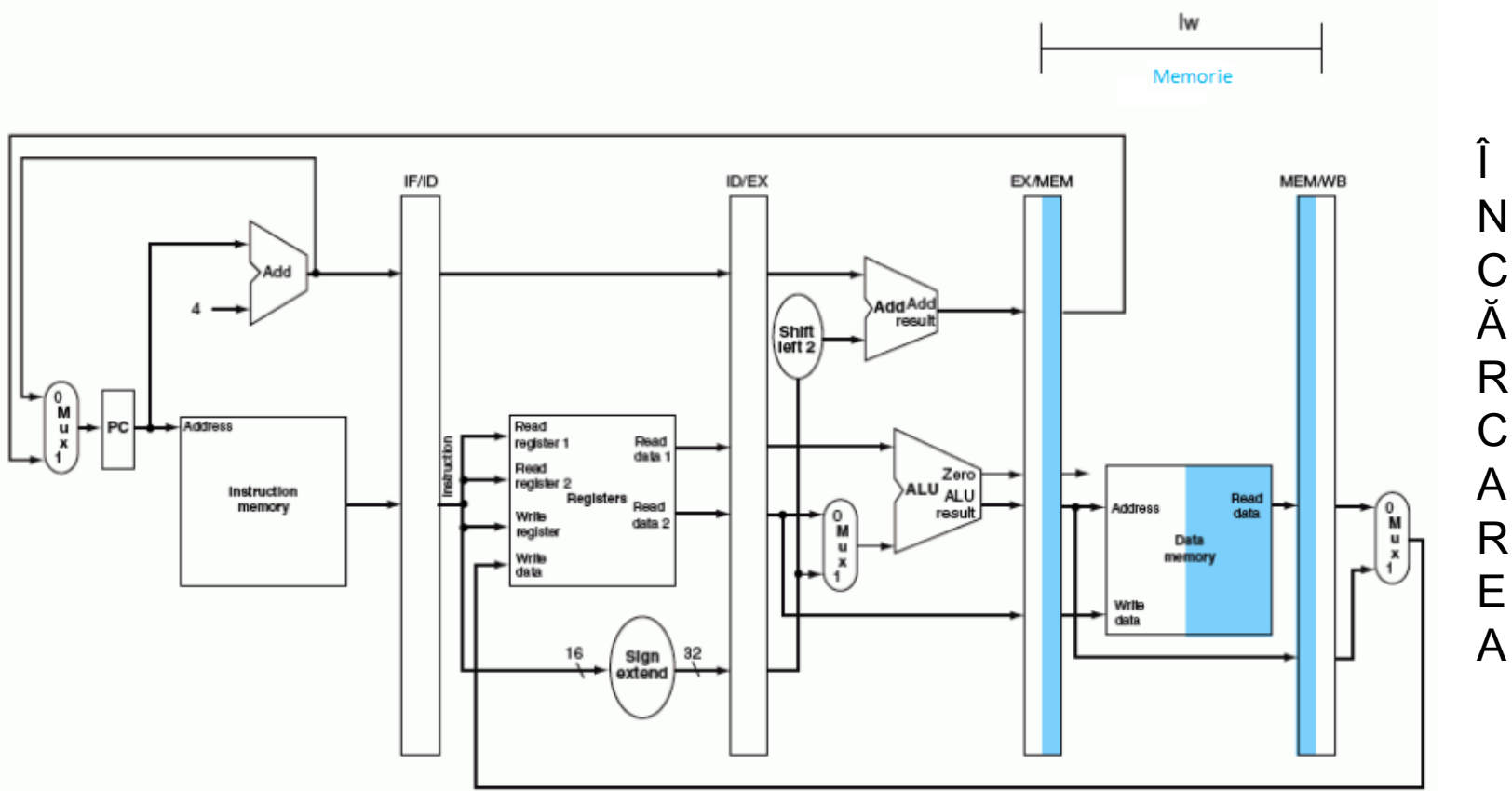
lw
Execuție

ÎN CĂRCAREA

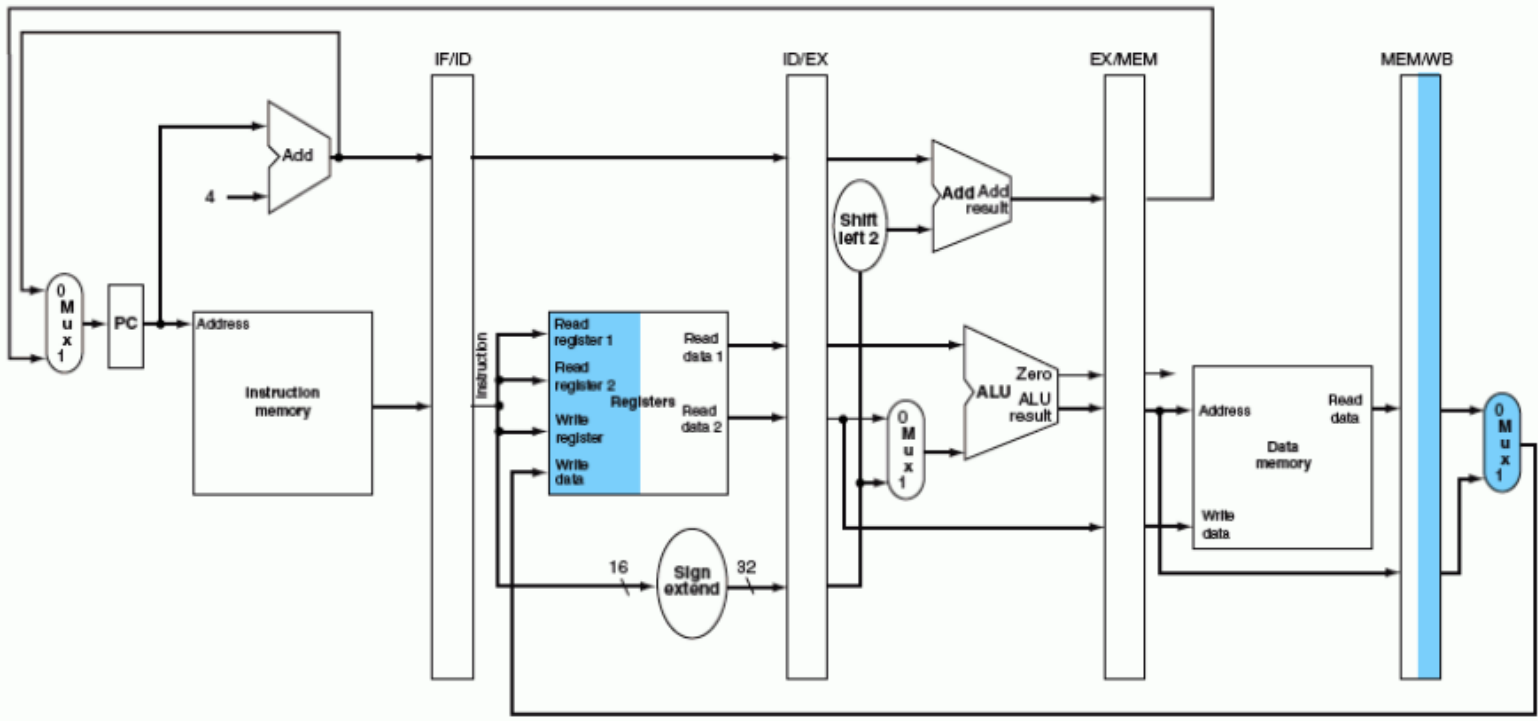


Execuția instrucțiunii – instrucțiunea de încărcare citește din registrul pipeline EI/DI conținutul registrului \$t1 și câmpul imediat cu semnul extins și le adună cu ajutorul UAL-ului.

Suma se va pune în registrul pipeline EX/MEM.



Accesul la memorie – instrucțiunea de încărcare citește memoria de date folosind adresa din registrul pipeline EX/MEM și încarcă data în registrul pipeline MEM/RS.

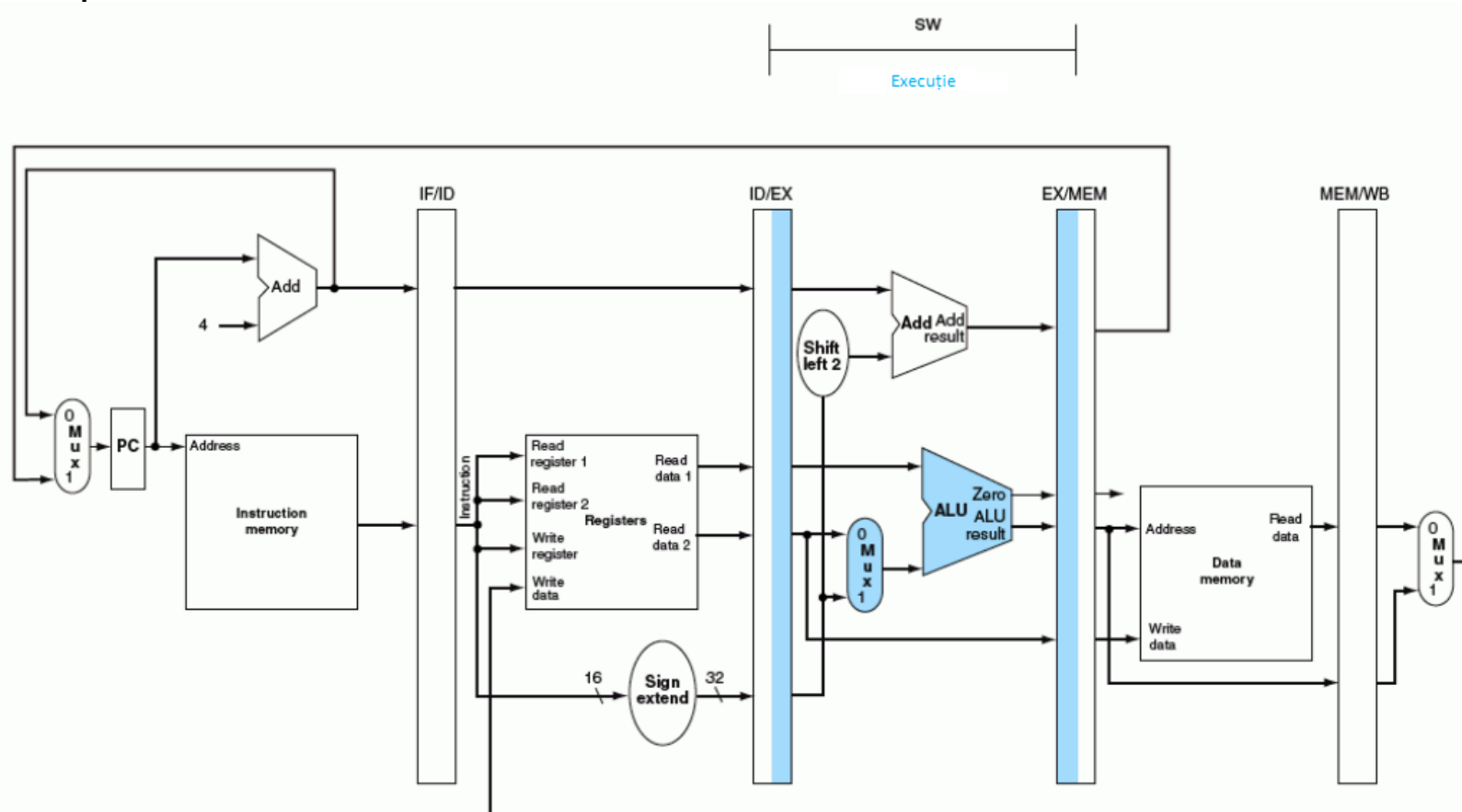


Rescrierea fișierului de registre – citirea datei din registrul pipeline MEM/RS pe care o scrie în fișierul de registre.

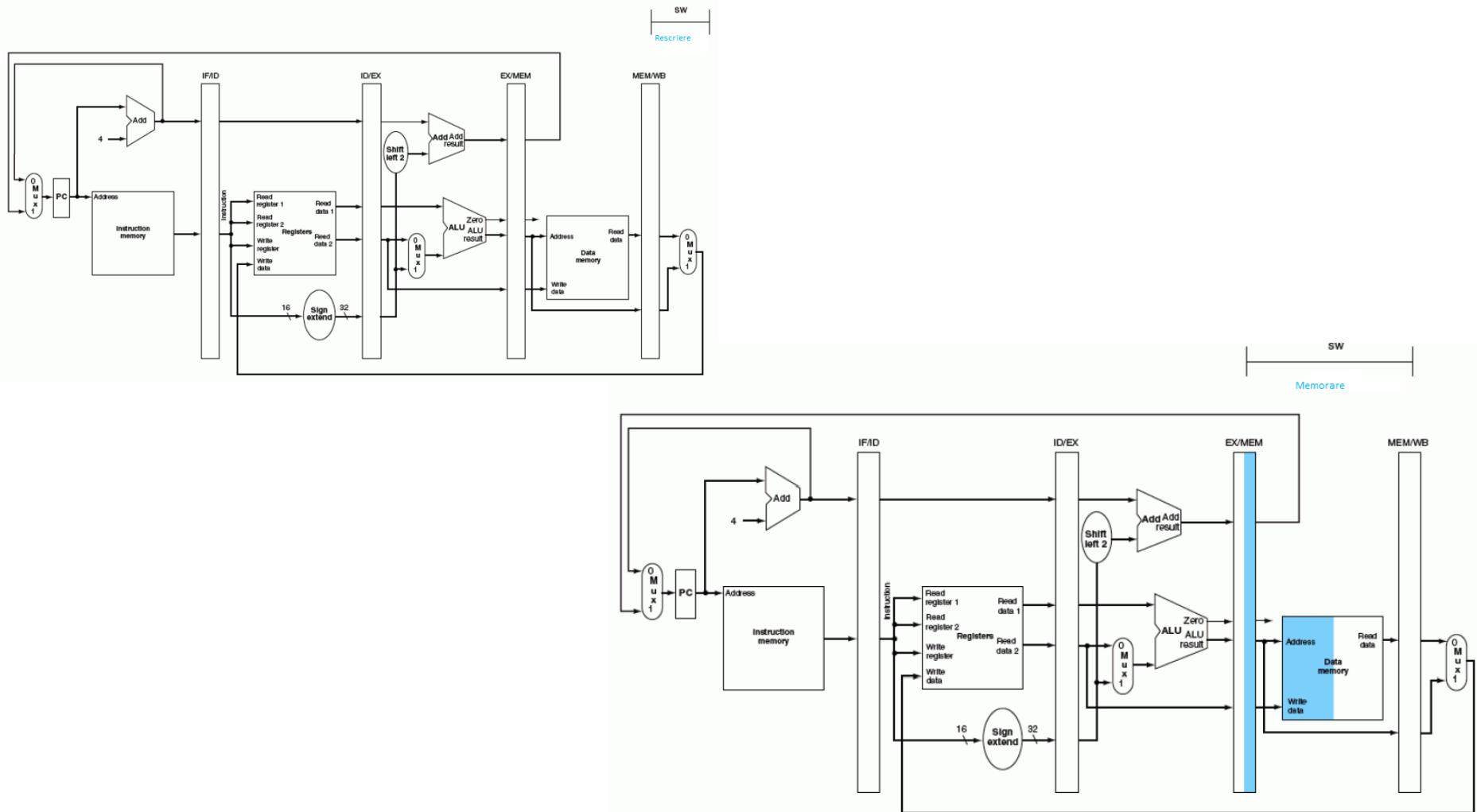
CONCLUZIE – orice informație necesară într-o etapă ulterioară de pipeline trebuie transmisă către etapa respectivă prin registrele pipeline.

INSTRUCȚIUNEA DE MEMORARE

1. EXTRAGEREA INSTRUCȚIUNII – instrucțiunea este citită din memorie folosindu-se adresa din PC, apoi este pusă în registrul pipeline EI/DI
2. DECODIFICAREA INSTRUCȚIUNII – respectă Fig. 2
3. EXECUȚIA INSTRUCȚIUNII – adună ce se citește din registrul EI/DI, conținutul lui \$t1, și câmpul imediat cu semnul extins



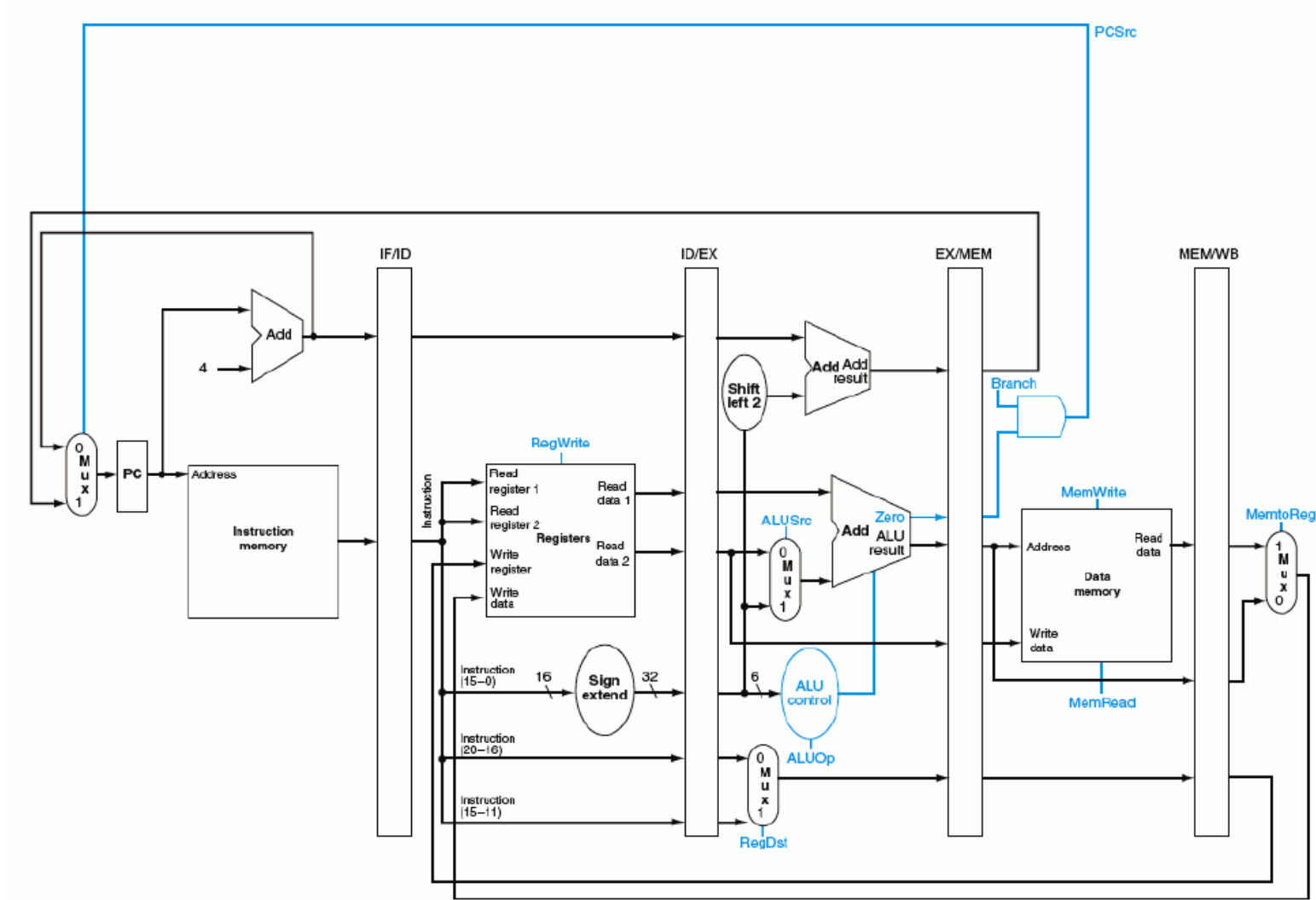
4. ACCESUL LA MEMORIE – citește memoria de date folosind adresa din registrul pipeline EX/MEM și încarcă data în registrul pipeline MEM/RS



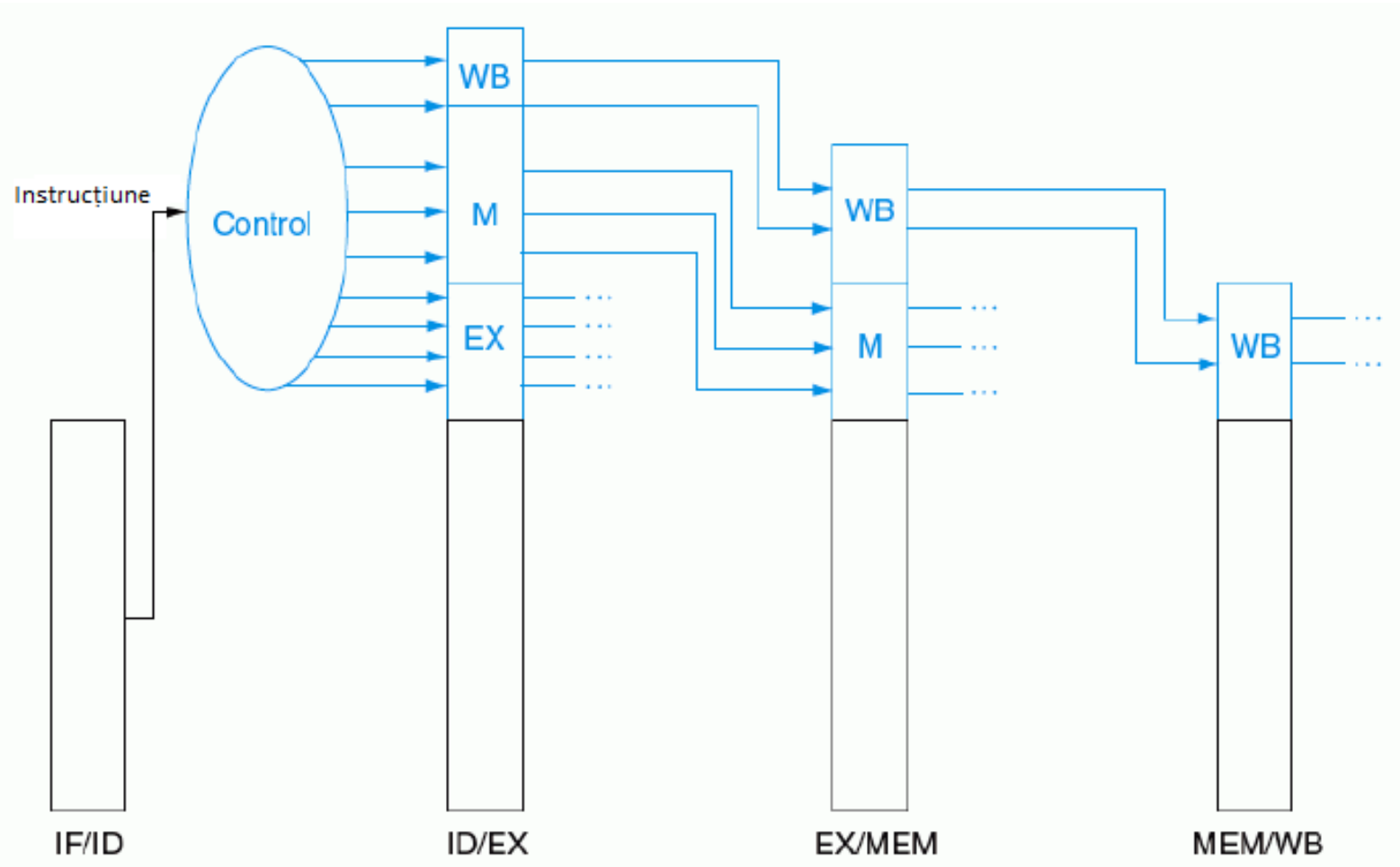
5. RESCRIEREA FIȘIERULUI DE REGISTRE – citirea datei din registrul pipeline MEM/RS pe care o scrie în fișierul de registre.

CONTROLUL PENTRU PIPELINE

Fiecare linie de control este asociată cu o componentă care este activă doar într-o Singură etapă pipeline.



Implementarea controlului



TEMĂ

Să se arate trecerea prin pipeline a următoarelor 5 instrucțiuni:

lw	\$10, 20 (\$1)
sub	\$11, \$2, \$3
and	\$12, \$4, \$5
or	\$13, \$6, \$7
add	\$14, \$8, \$9

Să se eticheteze instrucțiunile din pipeline care precedă instrucțiunea lw sub forma

inainte <1>, inainte <2>,

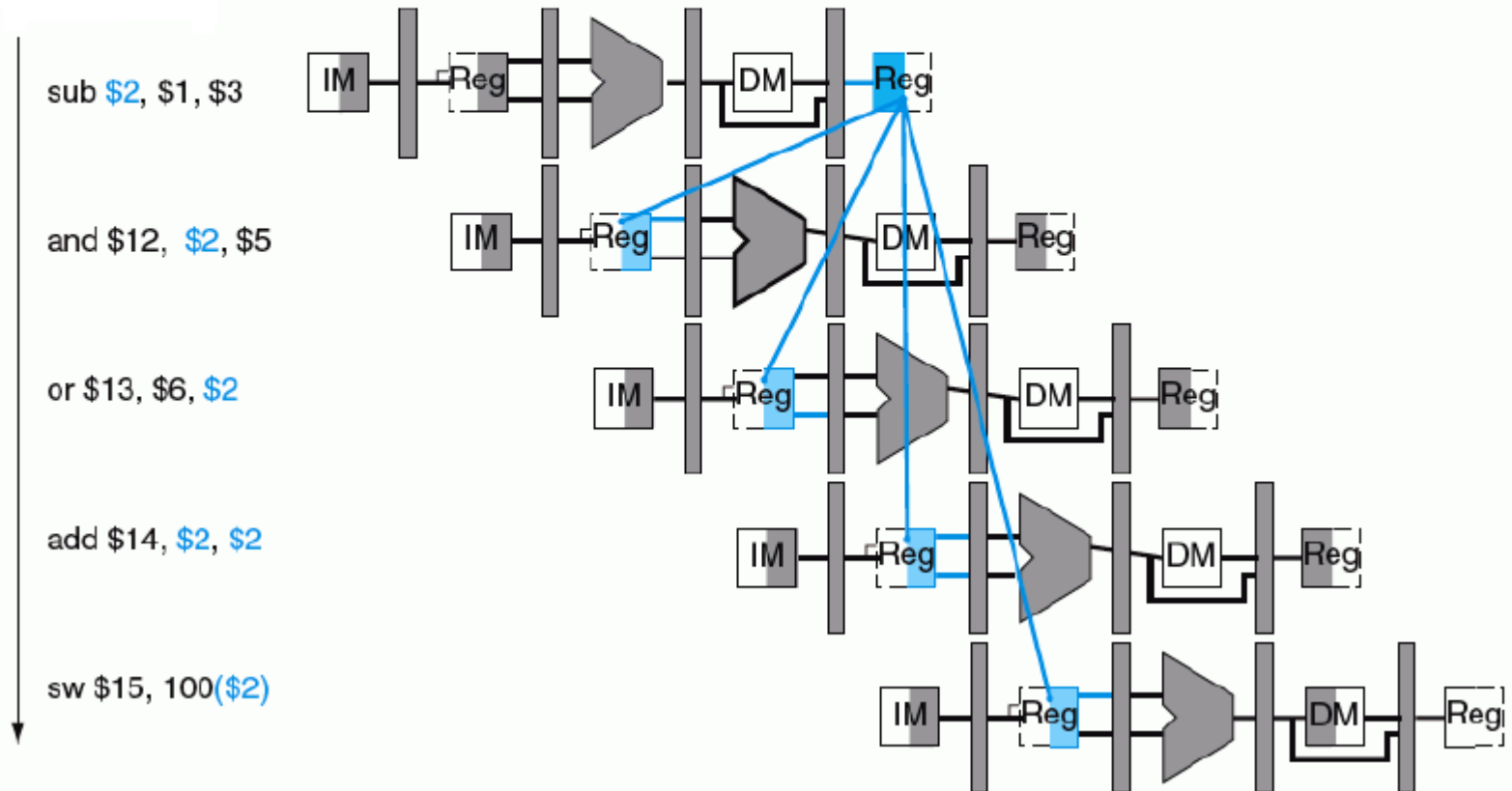
și instrucțiunile care urmează instrucțiunii add sub forma

dupa <1>, dupa <2>

HAZARDURILE DE DATE ȘI AVANSAREA

Valoarea registrului \$2:	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7	CC 8	CC 9
10	10	10	10	10	10/-20	-20	-20	-20	-20

Ordinea de execuție a programului (în instrucțiuni)



Soluția software – introducerea nop-urilor => cicluri de ceas în care nu se face nimic

Detecția hazardului

1a. EX/MEM.RegistruRd = DI/EX.RegistruRs

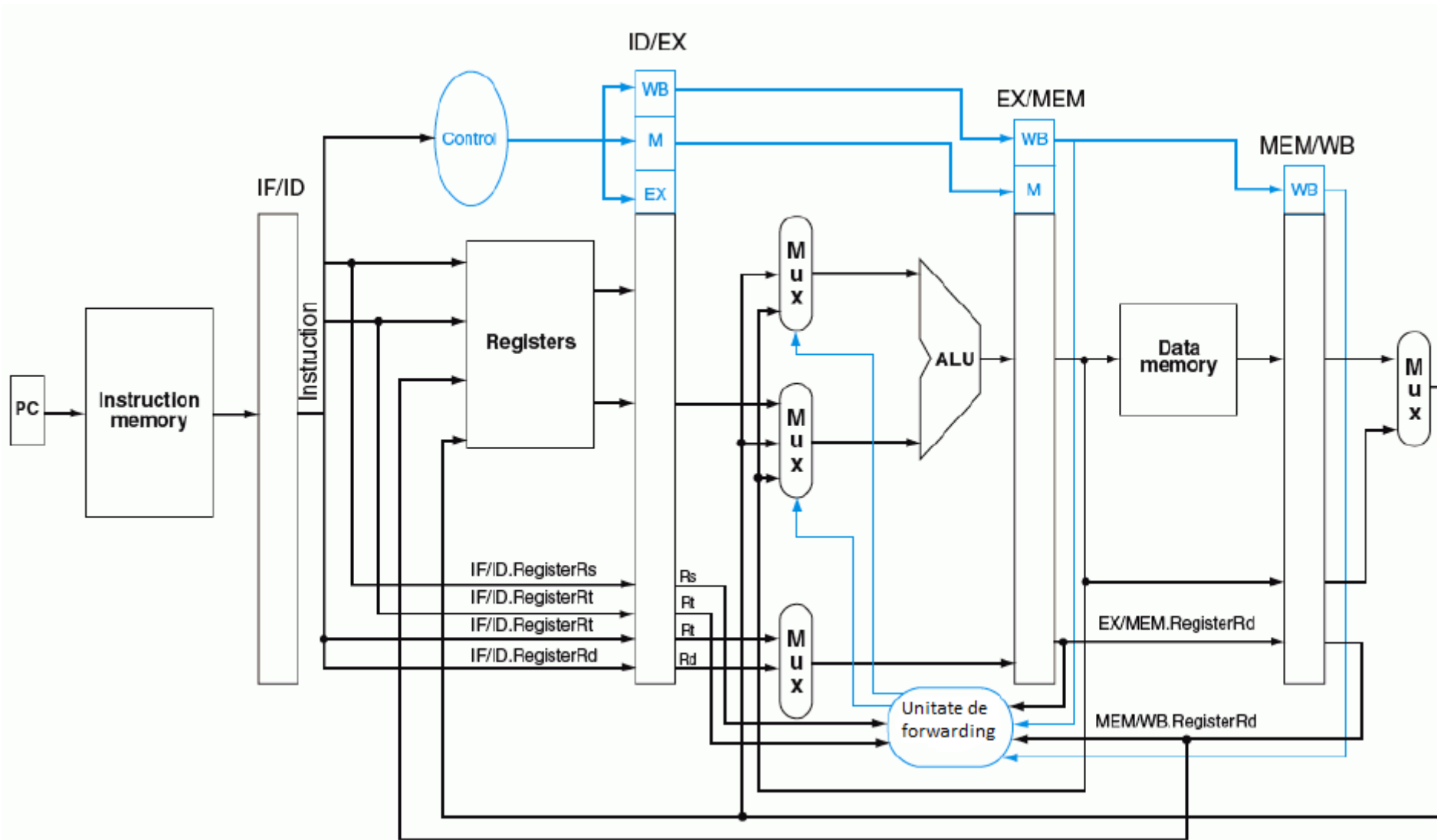
1b. EX/MEM.RegistruRd = DI/EX.RegistruRt

2a. MEM/RS.RegistruRd = DI/EX.RegistruRs

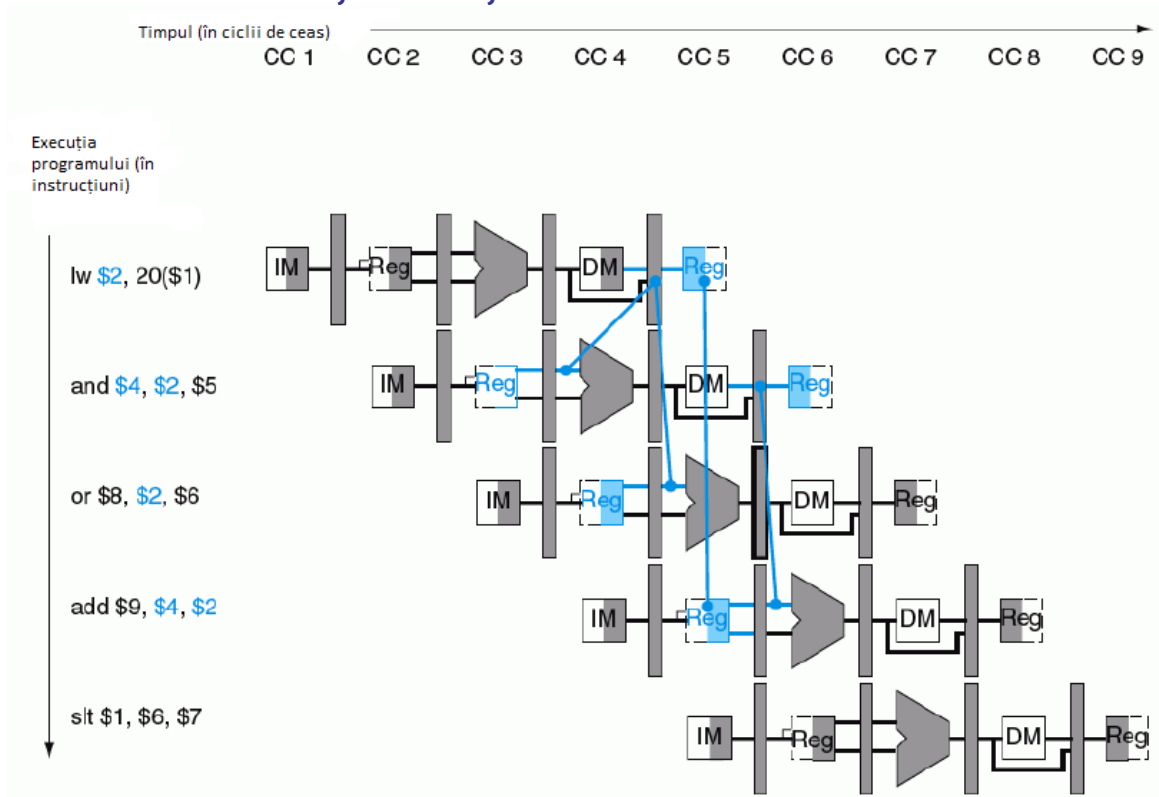
2b. MEM/RS.RegistruRd = DI/EX.RegistruRt

Dacă intrările UAL pot fi luate de la orice registru pipeline, nu numai de la EI/DI, atunci avansarea ar fi corectă.

Soluția hardware



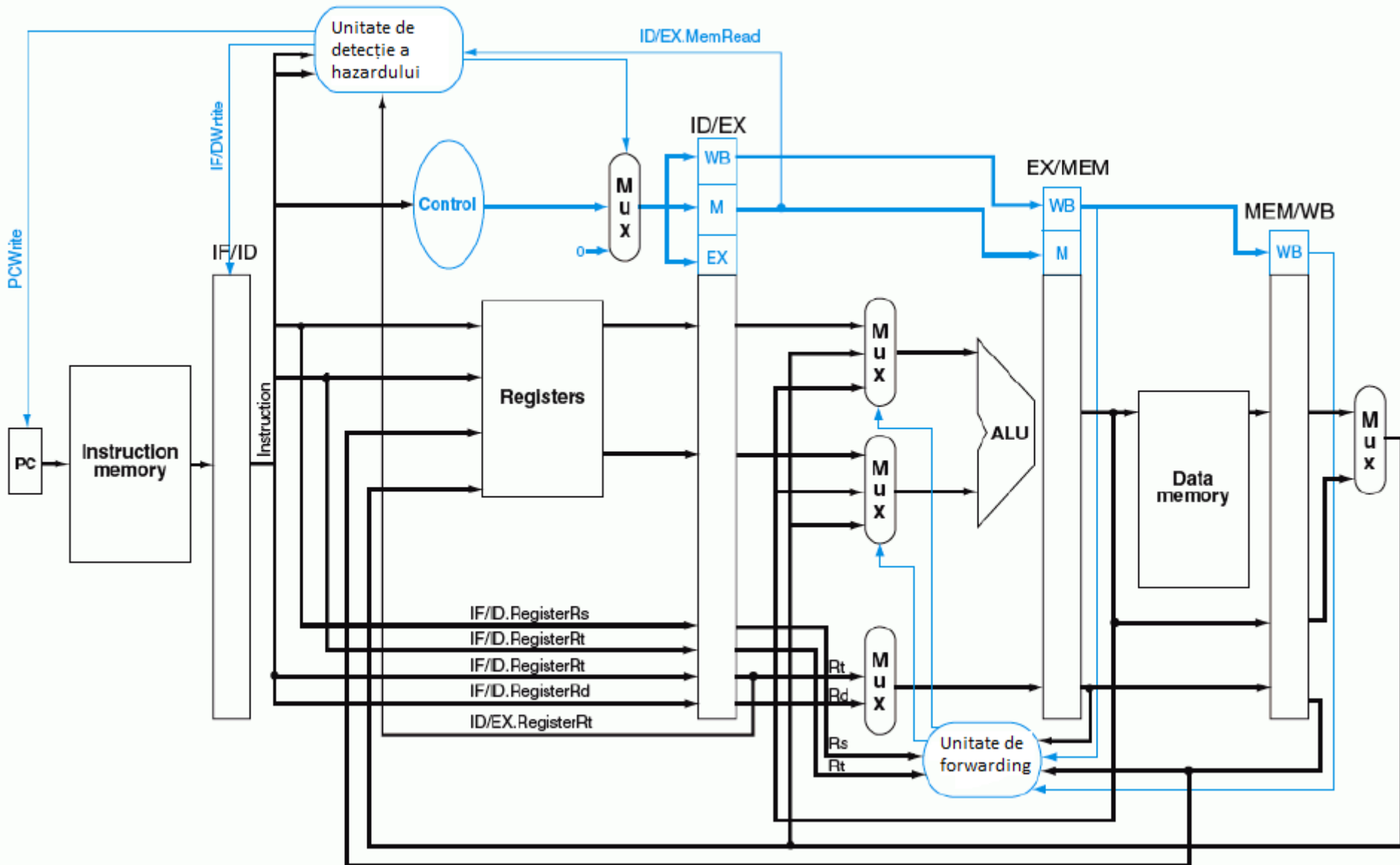
HAZARDURILE DE DATE ȘI STAȚIONĂRILE



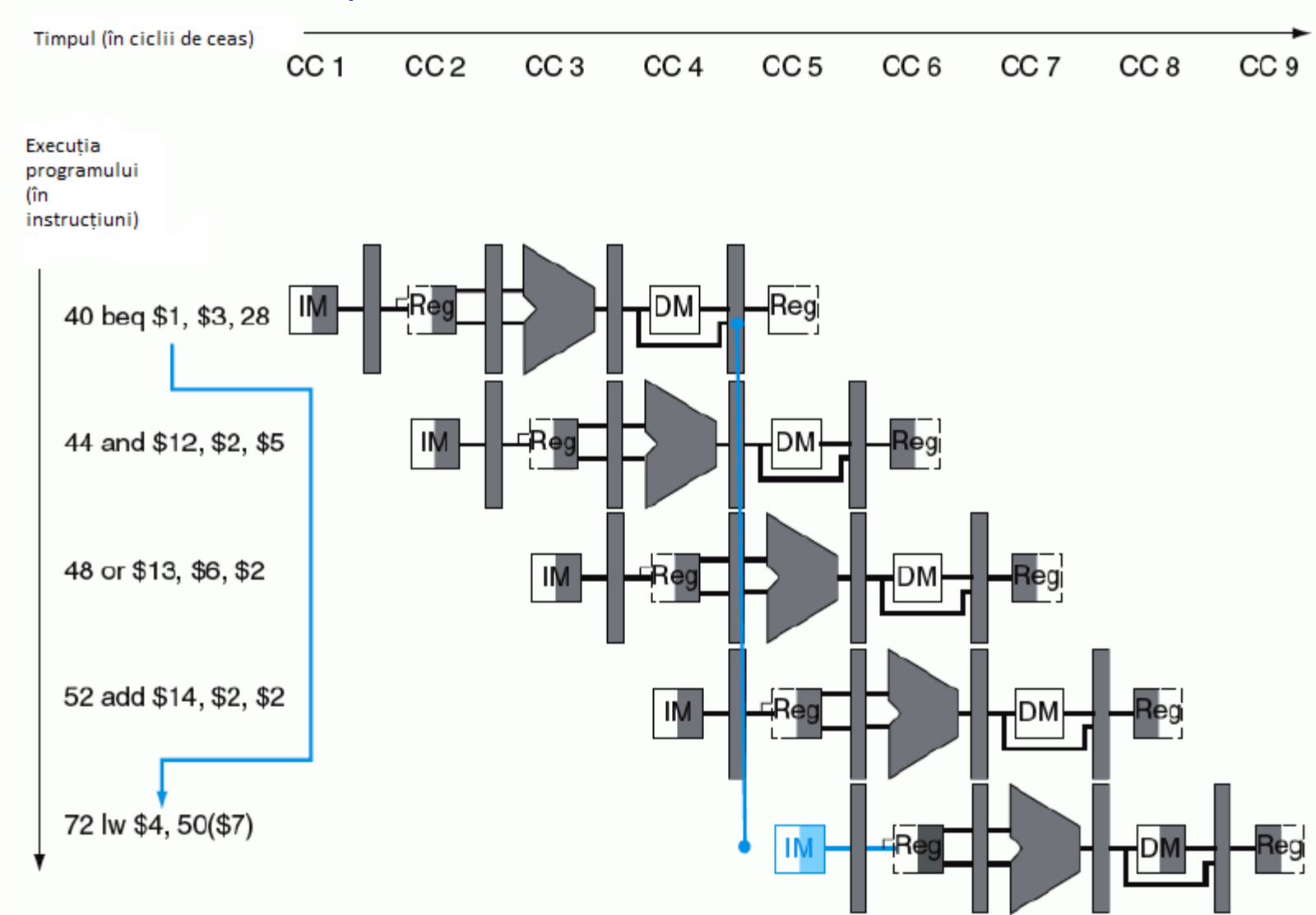
Pe lângă unitatea de avansare trebuie să existe o unitate de detectare a hazardului.

If (ID/EX.CiteșteMem and ((ID/EX.RegistruRt = IE/ID.RegistruRs) or
 (ID/EX.RegistruRt = IE/ID.RegistruR)))
 staționare pipeline

Rezultatul final



Hazarduri de ramificație



Predicția dinamică a ramificației

Presupunem că ramificația nu va fi acceptată

O altă schemă de predicție se bazează pe observarea adresei instrucțiunii

Deficiență

Chiar dacă ramificația va fi acceptată aproape întotdeauna, predicția va fi incorectă de două ori atunci când ramificația nu este acceptată.

Exemplu:

Se consideră o ramificație de buclă care ramifică de 9 ori consecutiv și apoi nu este acceptată o dată.

Soluție: introducerea schemelor de predicție cu 2 biți – o predicție trebuie să greșească de 2 ori înainte de a fi schimbată.

Pipeline superscalar și pipeline dinamic

Superpipeline – presupune pipeline-uri mai lungi

O altă îmbunătățire prevede repetarea componentelor hardware astfel încât să crească numărul de instrucțiuni prelucrate în fiecare etapă de pipeline.

Pipeline dinamic – realizat de hardware pentru evitarea hazardurilor pipeline. De regulă se adaugă resurse hardware suplimentare pentru executarea instrucțiunilor următoare în paralel.

Deficiență

Creșterea complexității controlului.

Lw	\$t0, 20(\$s2)
Addu	\$t1, \$t0, \$t2
Sub	\$s4, \$s4, \$t3
Slti	\$t5, \$s4, 20