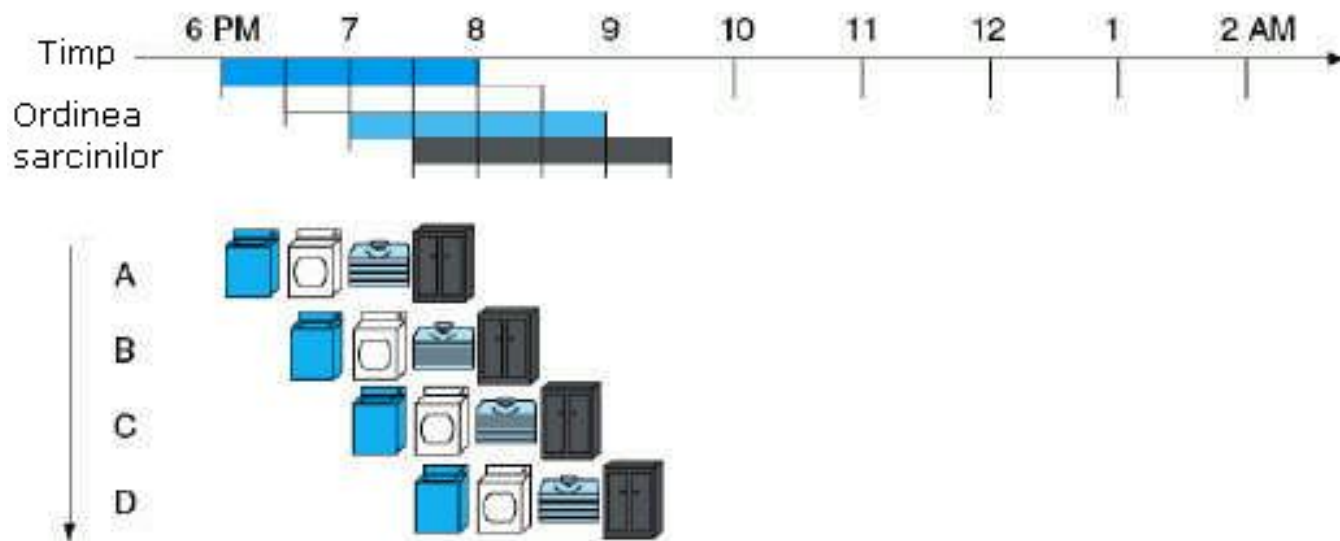
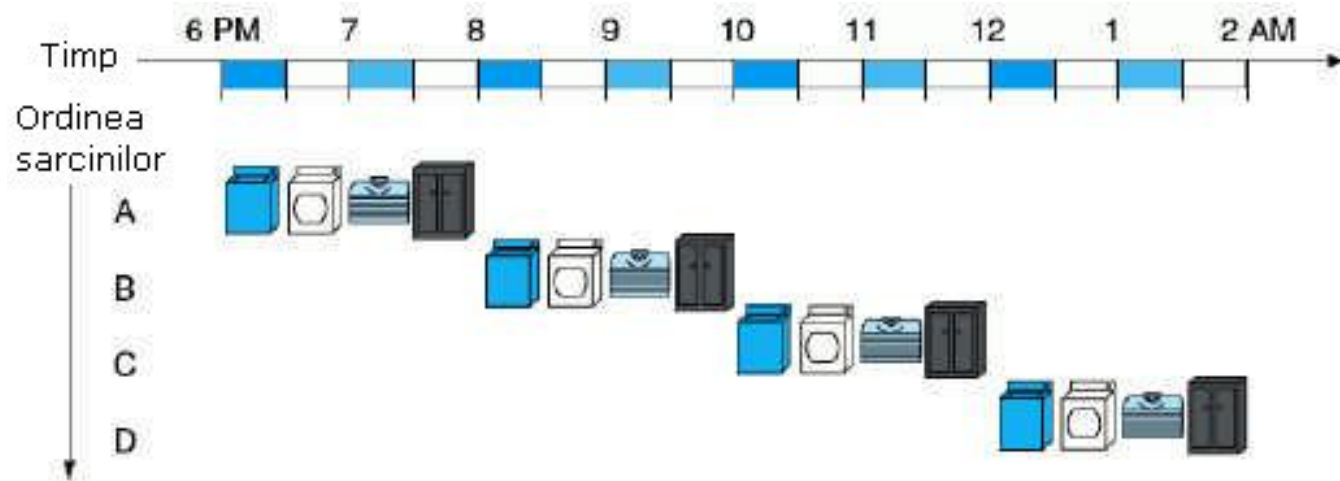


PIPELINE

Introducere

Pipeline – execuția mai multor instrucțiuni se suprapune în timp



Etajul 1 – extragerea instrucțiunii din memorie

Etajul 2 – citirea registrelor în timp ce instrucțiunea este decodificată

Etajul 3 – executarea operației sau calcularea adresei

Etajul 4 – accesul la un operand din memorie

Etajul 5 – scrierea rezultatului într-un registru

Timpul între instrucțiuni $_{\text{pipeline}}$ = Timpul între instrucțiuni $_{\text{fără pipeline}}$ / nr de etaje

Pipeline-ul îmbunătățește performanța prin creșterea productivității instrucțiunilor, nu prin micșorarea timpului de execuție al unei instrucțiuni individuale

OBSERVAȚII

Instrucțiunile au aceeași lungime => extragerea instrucțiunilor și decodificarea lor se poate face în doar două etaje de pipe.

Câpurile registrelor sursă sunt localizate în același loc în cadrul instrucțiunii => în etajul 2 putem începe citirea fișierului de registre în același timp în care hardware-ul determină instrucțiunea ce a fost extrasă.

Operanzii din memorie apar numai în instrucțiunile de încărcare și memorare => putem utiliza etapa de execuție pentru calcularea adresei de memorie, iar accesul la memorie se efectuează în etapa următoare.

Operanzii trebuie să fie aliniați în memorie => data poate fi transferată între procesor și memorie într-o singură etapă pipeline.

HAZARDURI PIPELINE

Evenimentul în urma căruia instrucțiunea următoare nu poate fi executată în următorul ciclu de ceas se numește **hazard**.

HAZARD SRUCTURAL

Hardware-ul nu poate suporta combinația de instrucțiuni pe care dorim să le executăm în același ciclu de ceas.

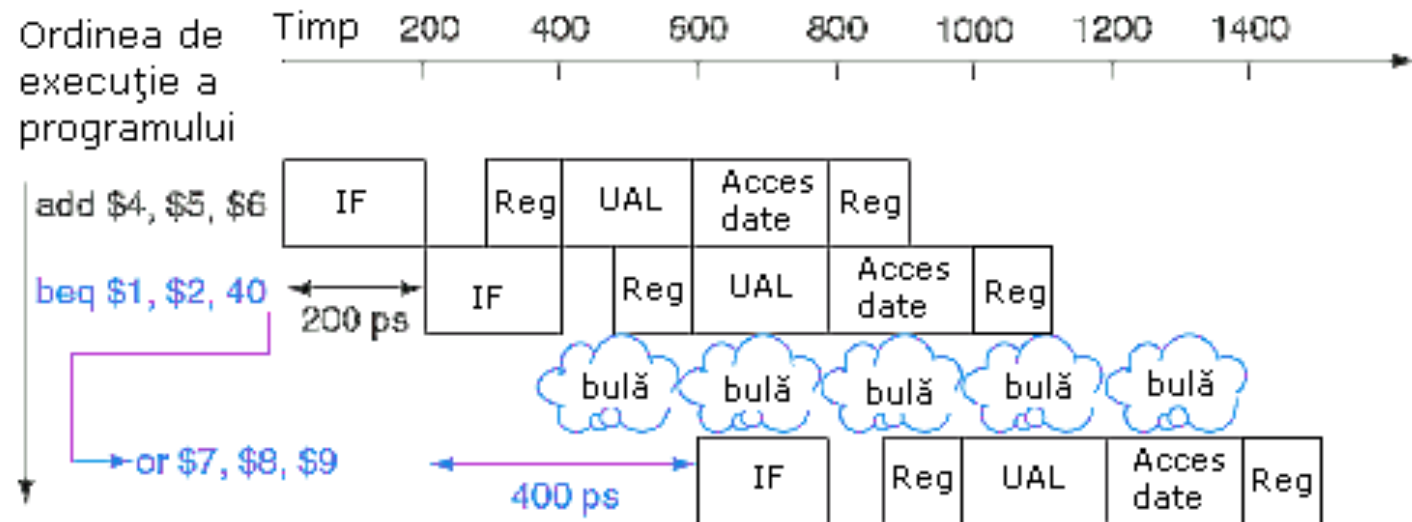
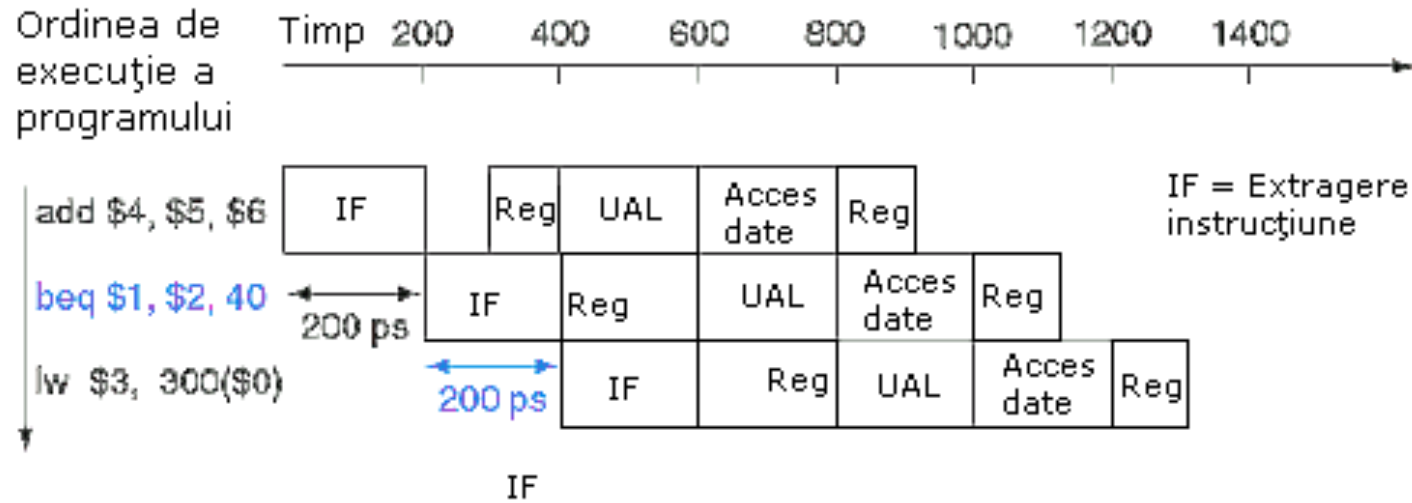
HAZARD DE CONTROL

Apare din necesitatea de a lua o decizie pe baza rezultatelor unei instrucțiuni, în timp ce altele sunt în execuție

SOLUȚII

1. Staționarea – primul ciclu se va executa secvențial și apoi se va trece la execuția pipeline – soluție viabilă în practică dar prea lentă în cazul pipe-urilor lungi

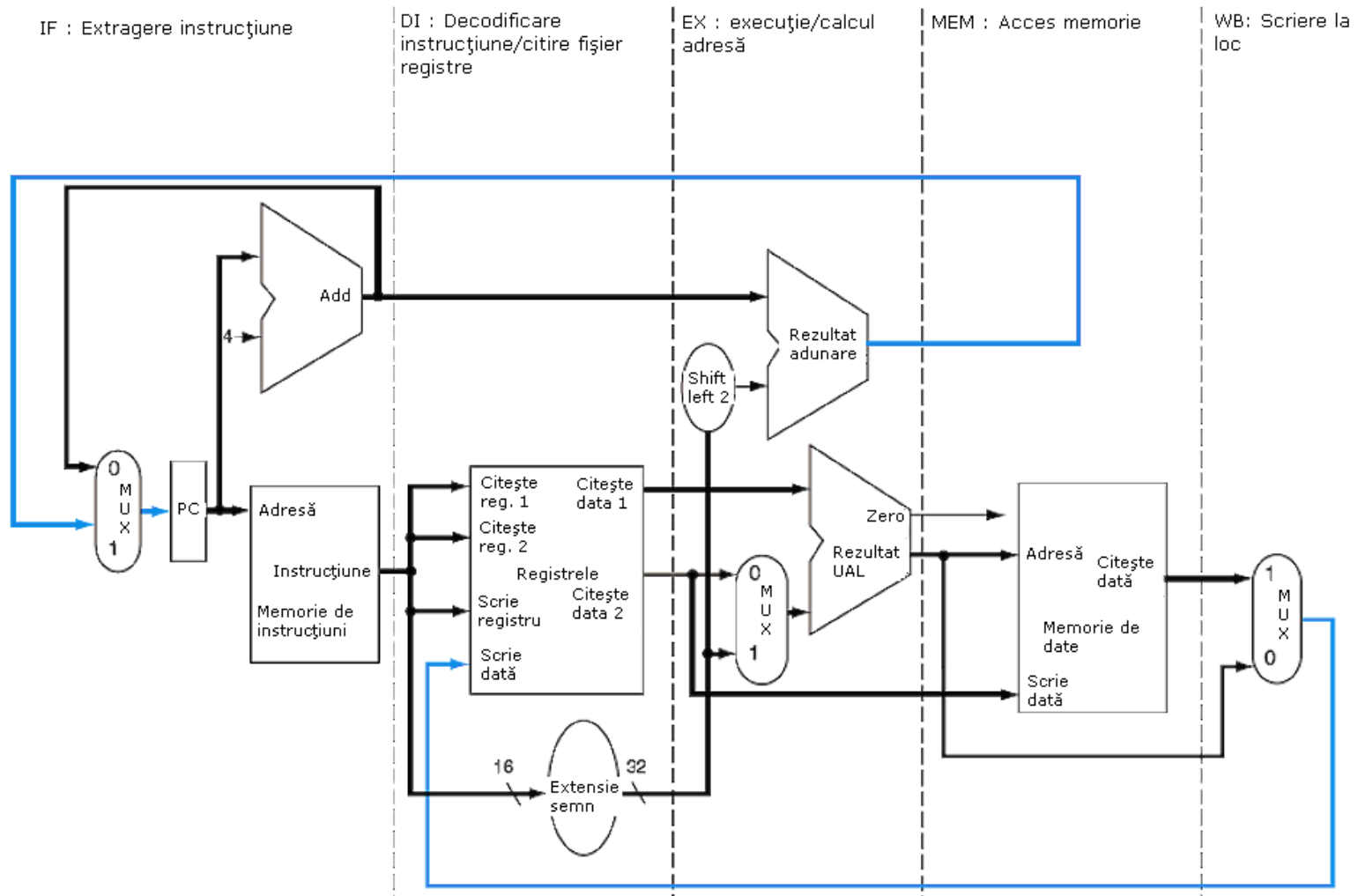
2. Predicția – se presupune că ramificațiile nu vor reuși => că doar în cazul reușitelor pipe-ul va staționa.



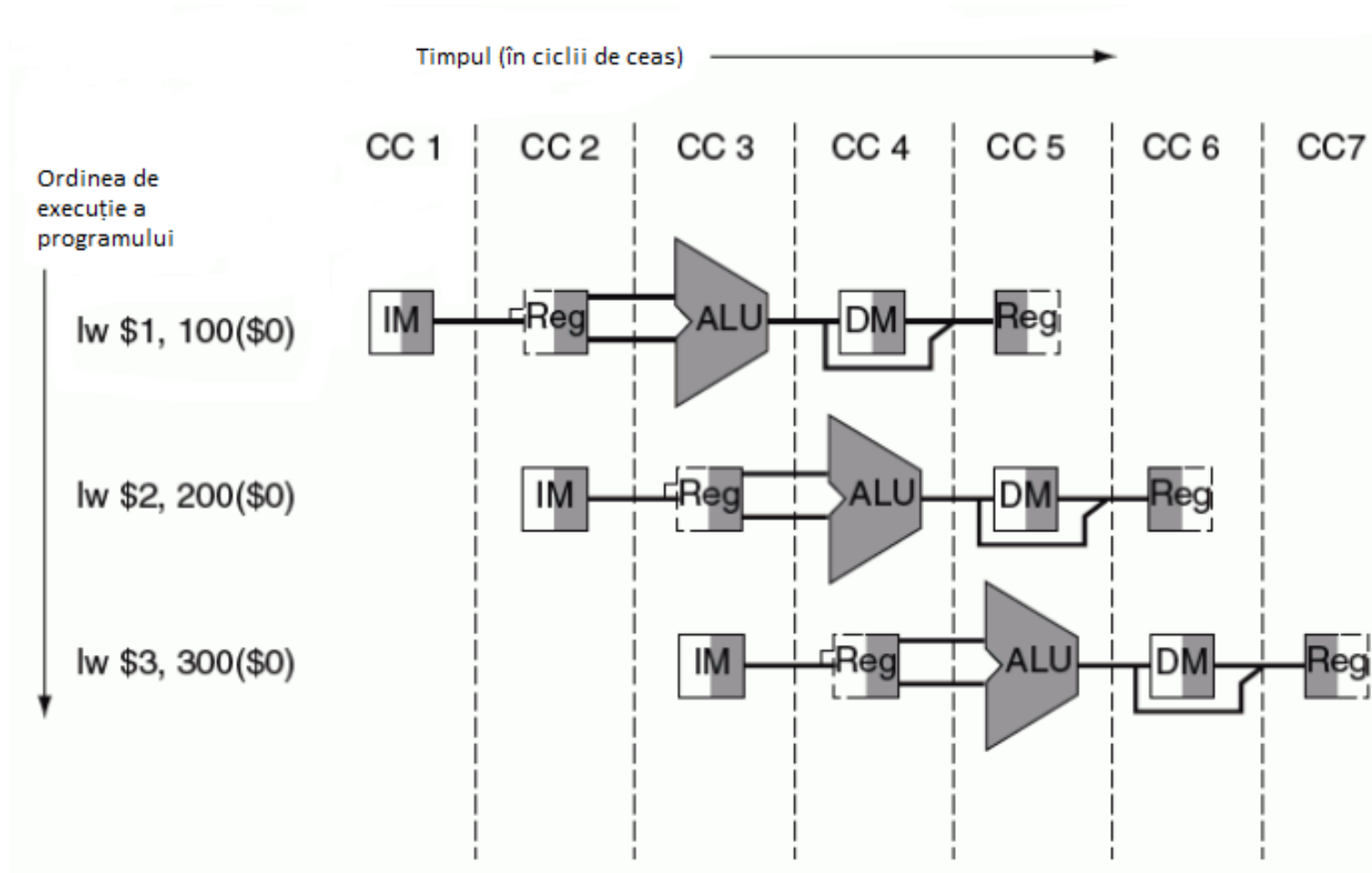
CONCLUZII

1. Pipeline-ul crește numărul instrucțiunilor executate simultan și viteza cu care instrucțiunile sunt începute și terminate
2. Execuția pipeline nu reduce timpul necesar derulării unei instrucțiuni individuale

Calea de date cu un singur ciclu de ceas



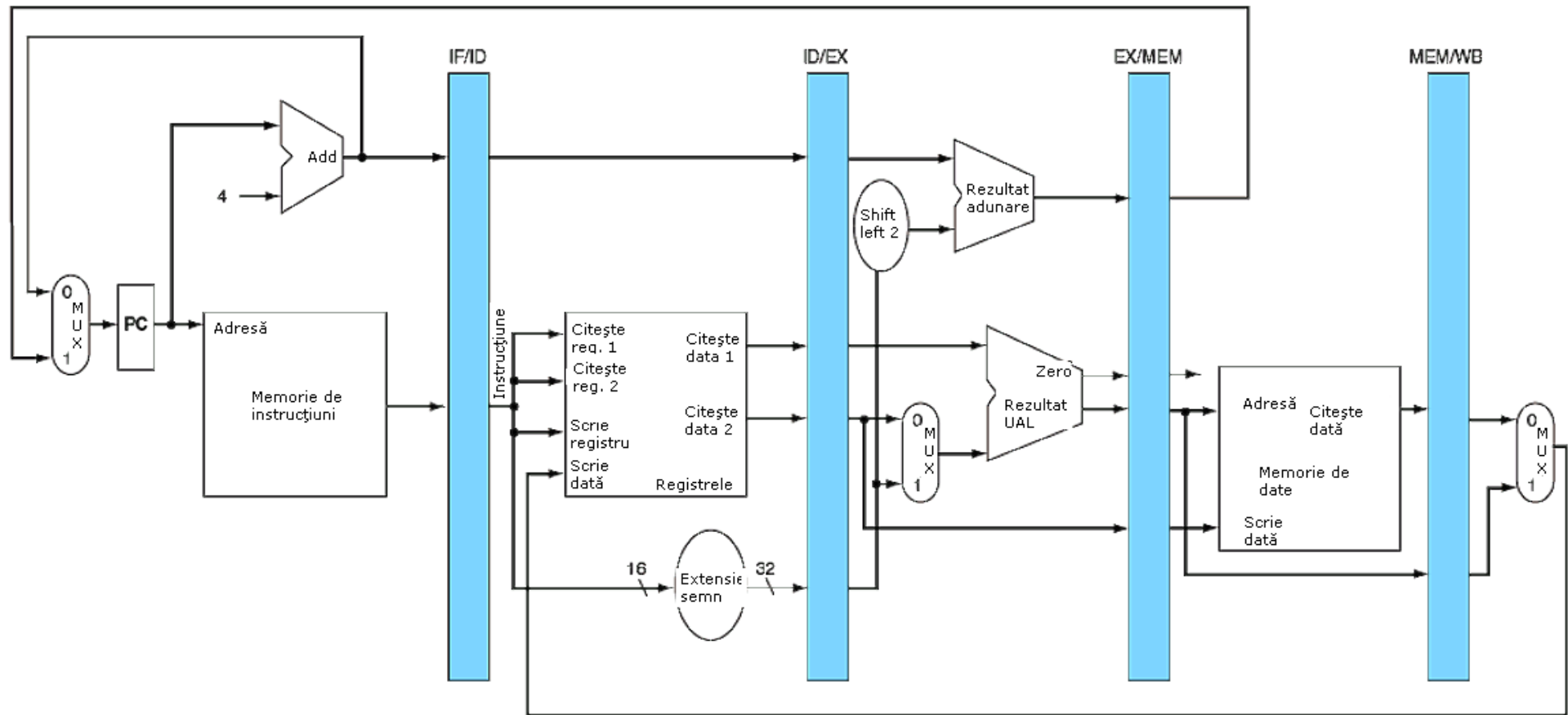
Excepții: actualizarea PC-ului și pasul de rescriere



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

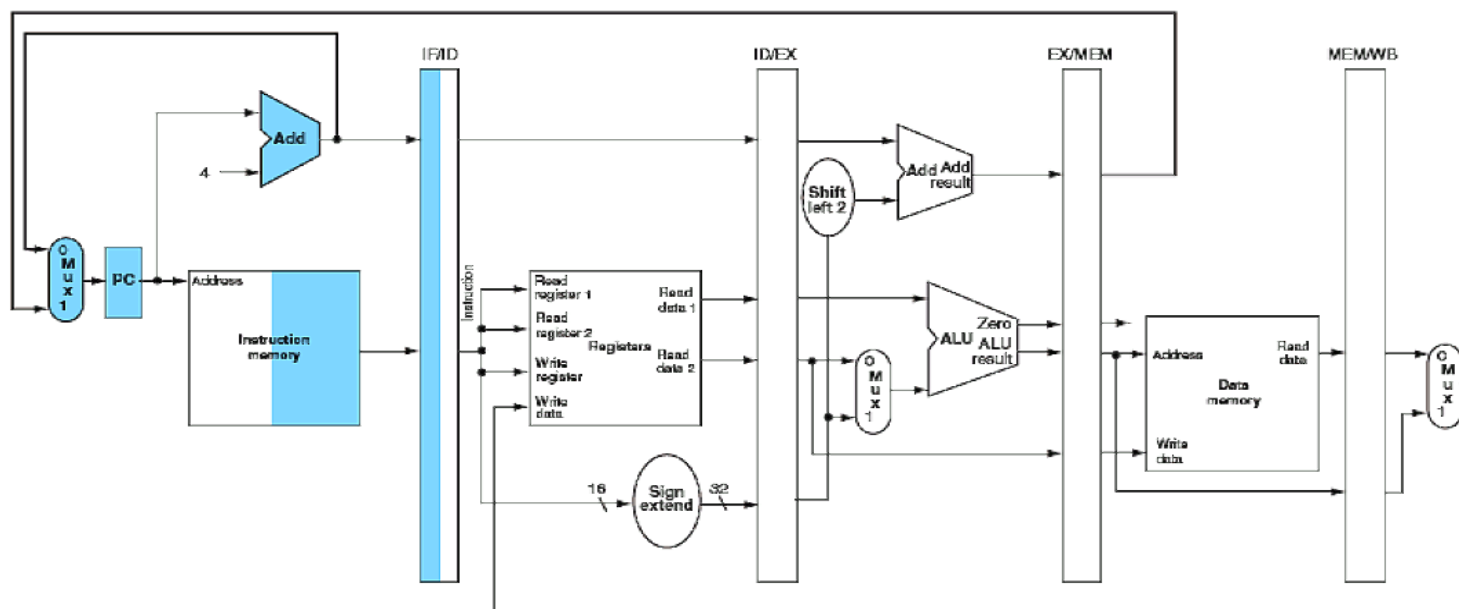


Registre pipeline separă fiecare etapă de pipe

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

NU EXISTĂ REGISTRE LA SFÂRȘITUL ETAPEI DE RESCRIERE

Fig. 1



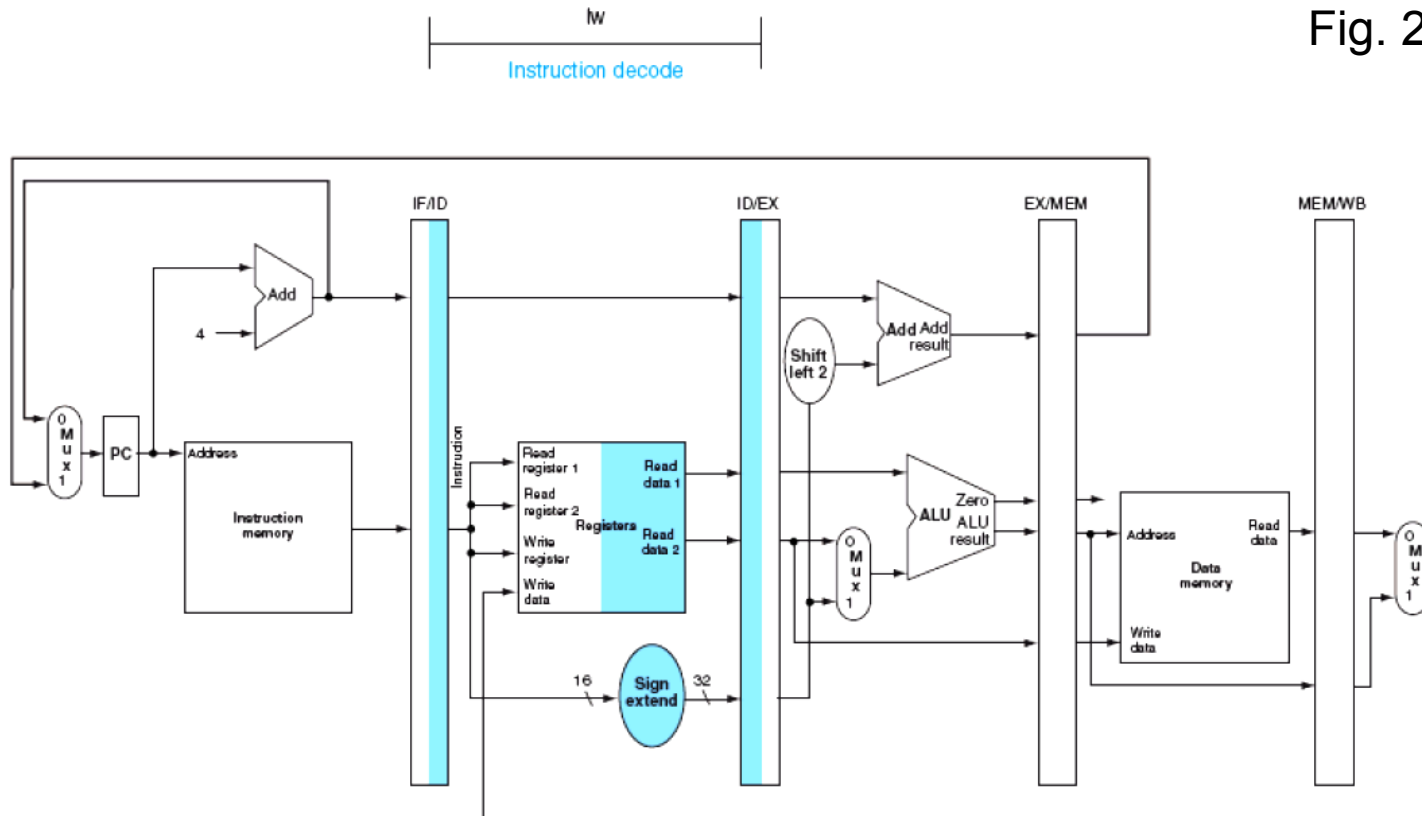
ÎN CĂR C A R E A

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

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

Calculatorul NU știe ce instrucțiune este extrasă.

Fig. 2

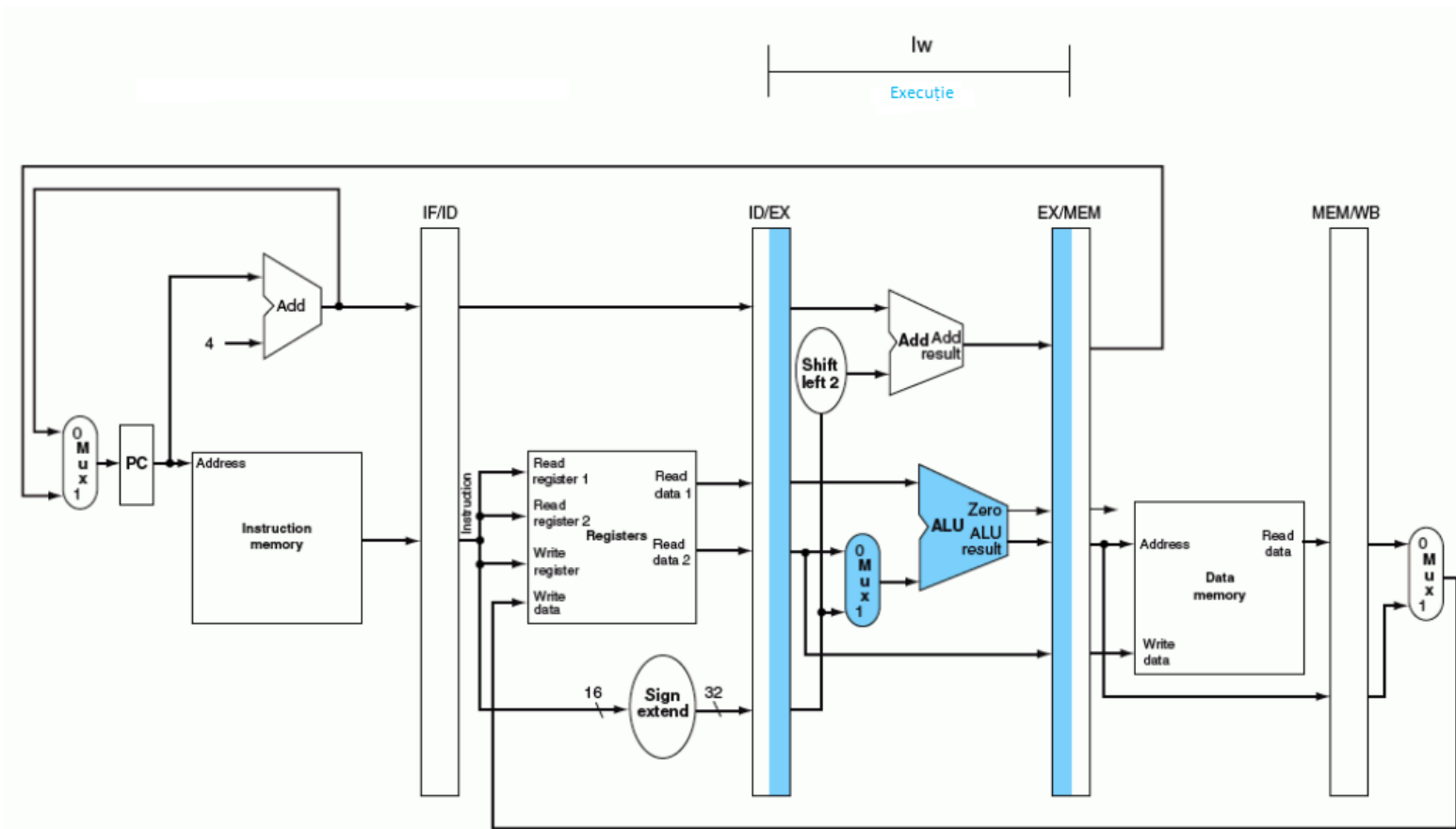


Decodificarea instrucțiunii – registrul EI/DI furnizează câmpul imediat de 16 biți căruia i se extinde semnul până la 32 de biți, și numerele registrelor pentru citirea celor două registre.

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

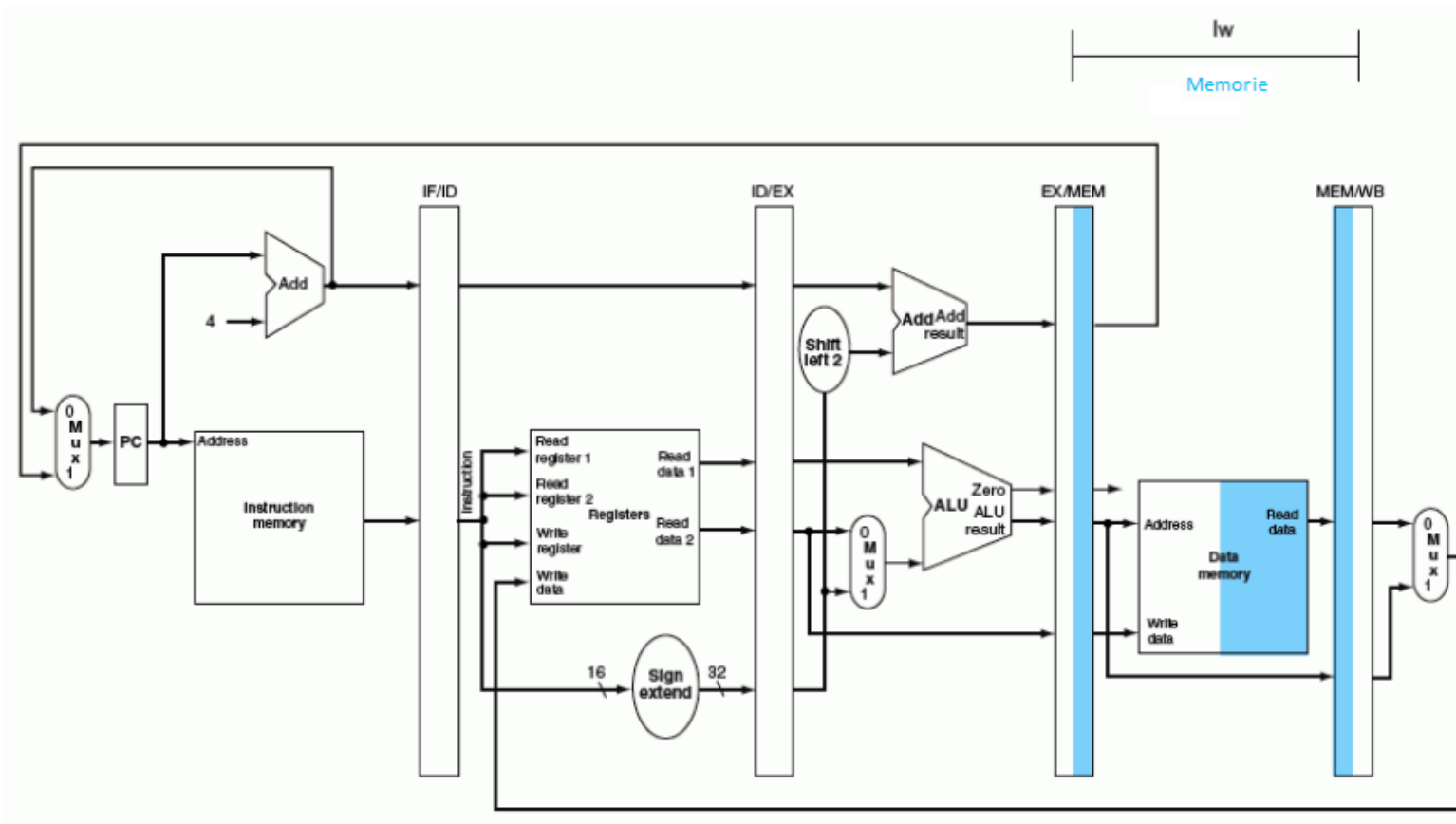
Sunt necesari toți cei 3 operanzi ?

Î
N
C
Ă
R
C
A
R
E
A



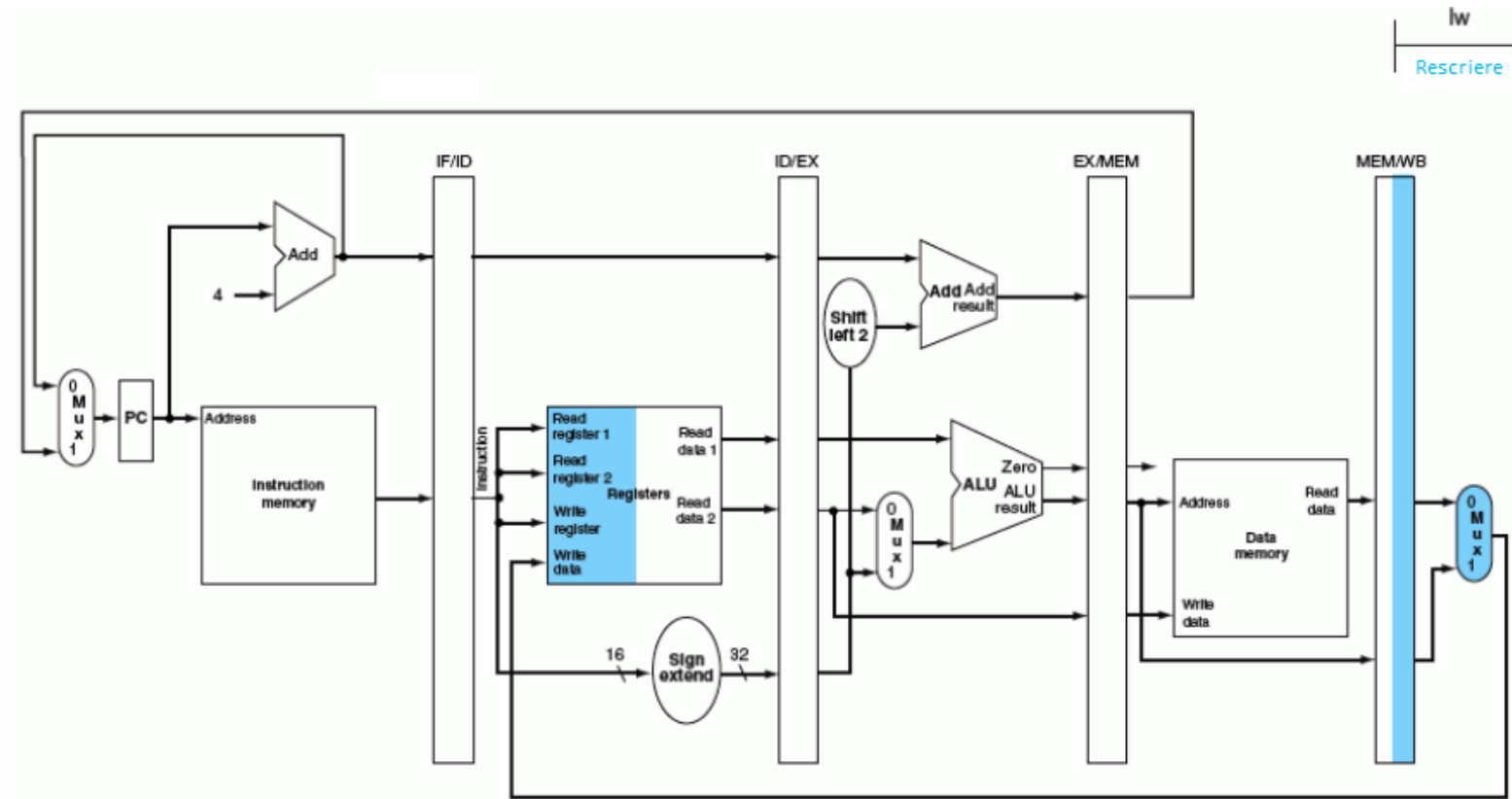
Execuția instrucțiunii – instrucțiunea de încărcare citește din registrul pipeline EI/DI conținutul registrului \$t1 și câmpul imediat cu semnul extins și le adună cu ajutorul UAL-ului.

Suma se va pune în registrul pipeline EX/MEM.



ÎN CĂR C A R E A

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.



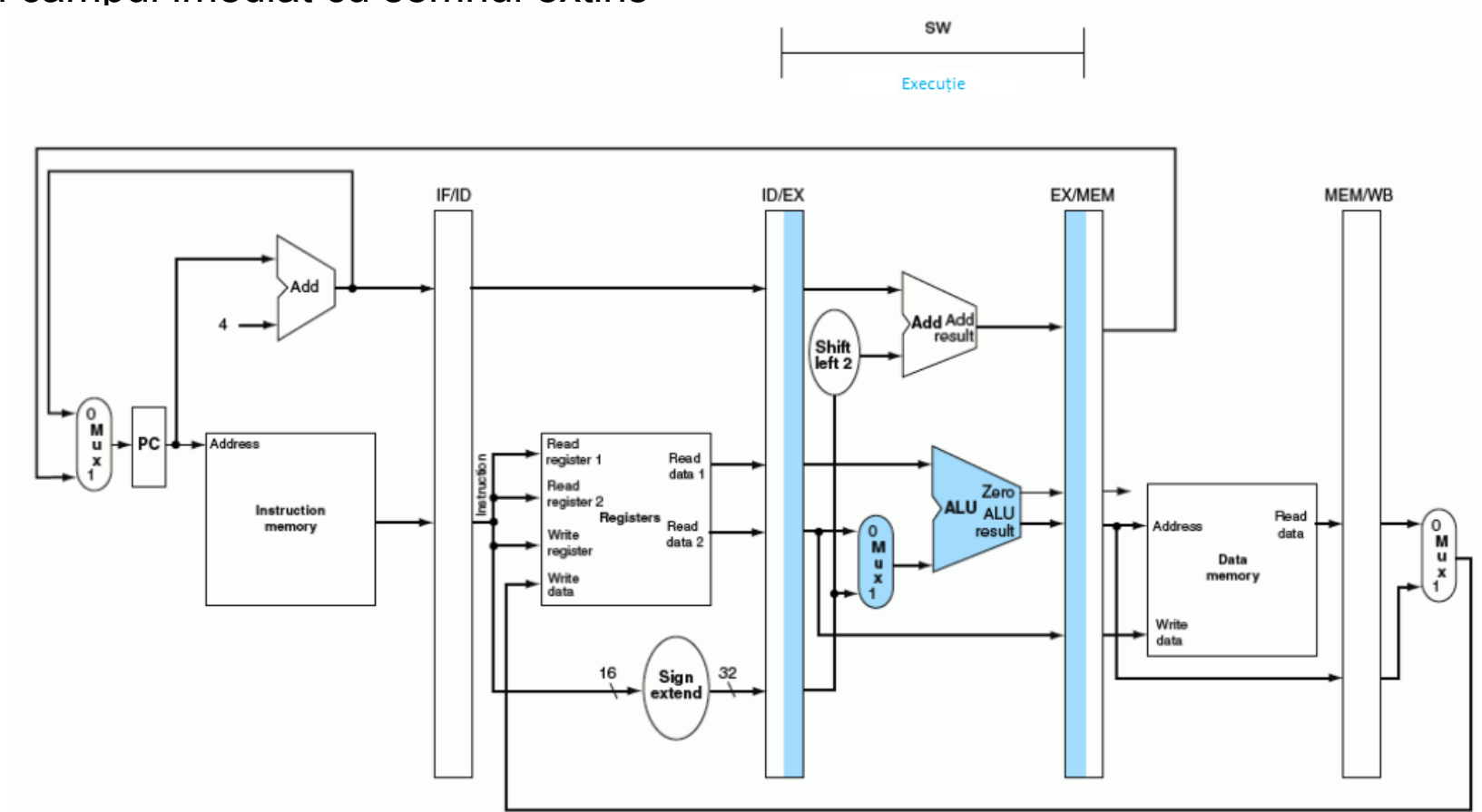
ÎN CĂR C A R E A

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

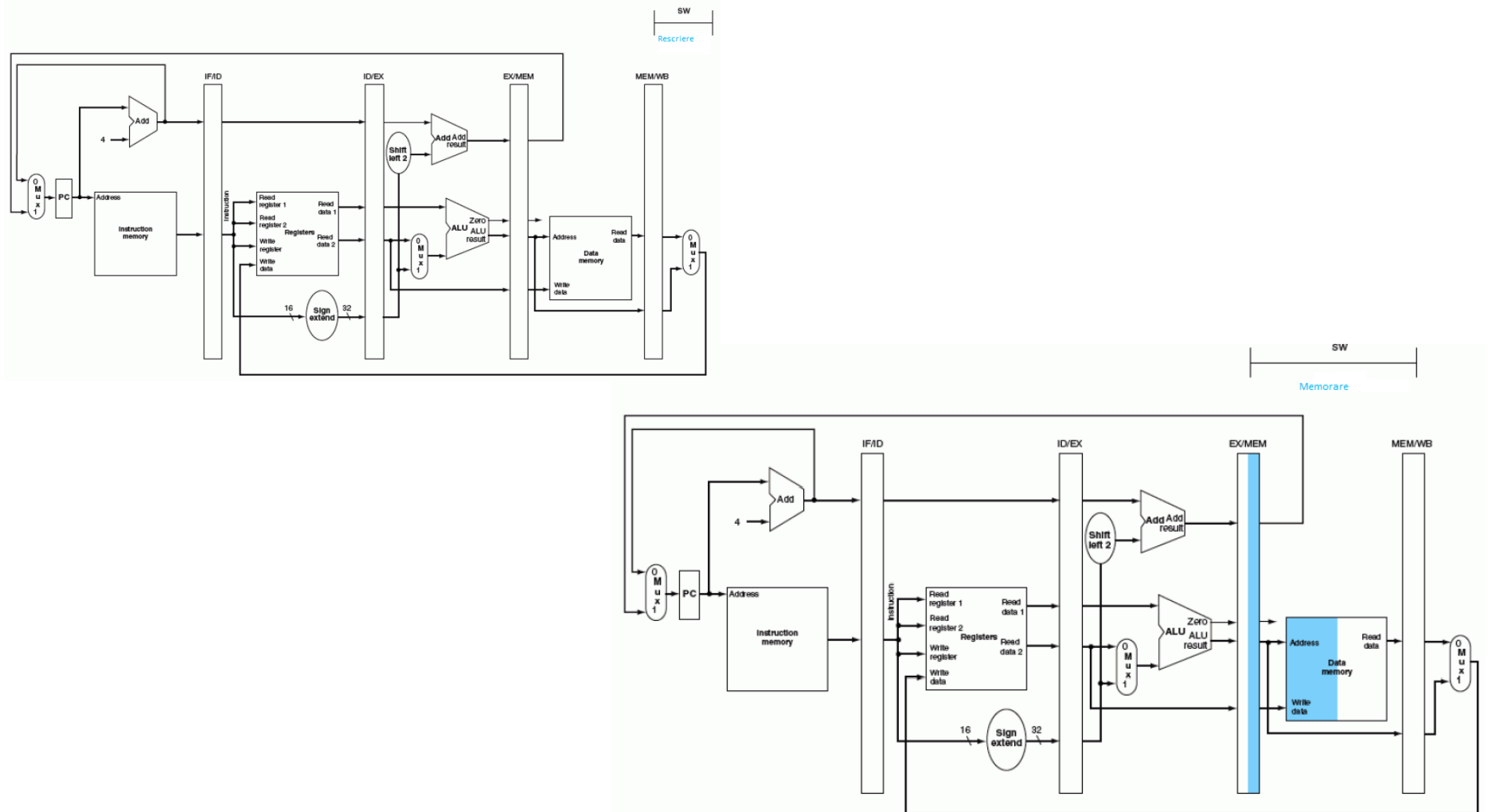
CONCLUZIE – orice informație necesară într-o etapă ulterioară de pipeline trebuie transmisă către etapa respectivă prin registrele pipeline.

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.