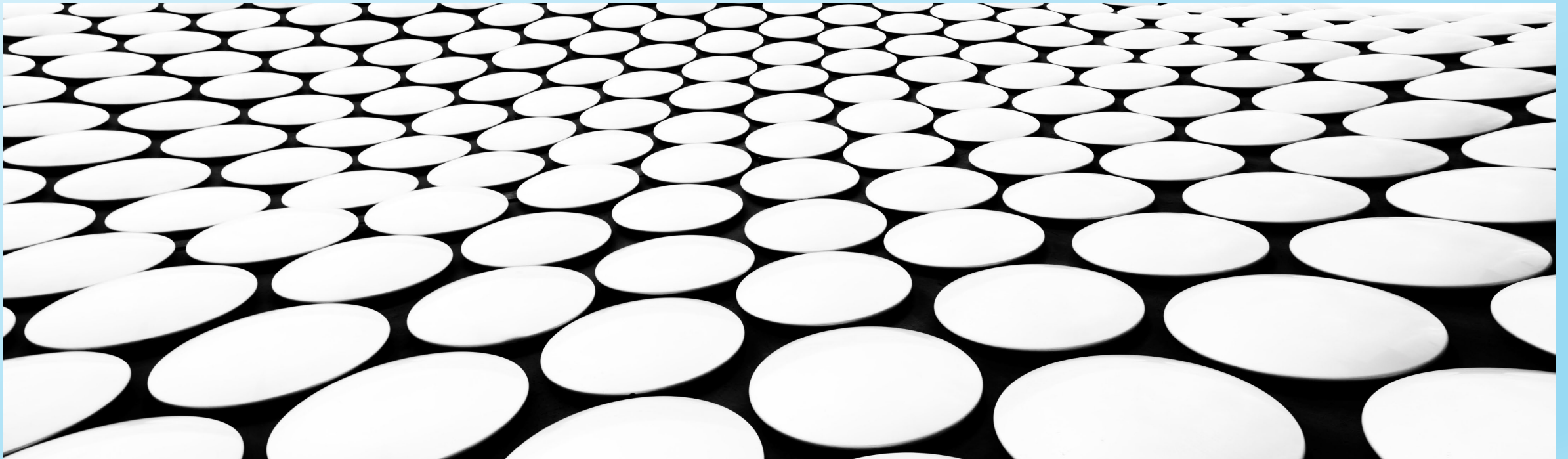

ARHITECTURA SISTEMELOR DE CALCUL

UB, FMI, CTI, ANUL III, 2022-2023



Tipuri de formate de instrucțiuni masina

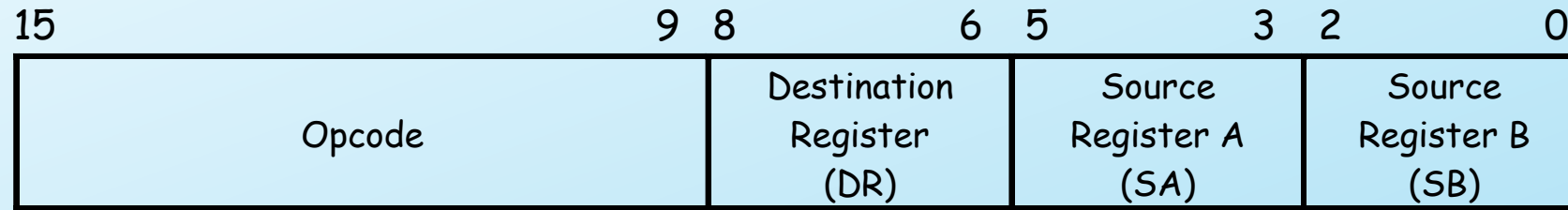
Tipuri de formate de instrucțiuni masina (3 tipuri)

- × Instrucțiuni tip: **Register format**
 - + Necesita doi registri sursa
- × Instrucțiuni tip: **Immediate format**
 - + Necesita un registru sursa si o constanta operand (inclusa in corpul instrucțiunii)
- × Instrucțiuni tip: **Jump and branch format**
 - + Necesita un registru sursa si o constanta adresa (inclusa in corpul instrucțiunii)

Formatul instrucțiunilor

Exemplu pe 16 biti

Register format (16 biti)



Un camp de 7 biți pentru codul operațiunii (**opcode**)

Un camp de 3 biți pentru registrul destinație

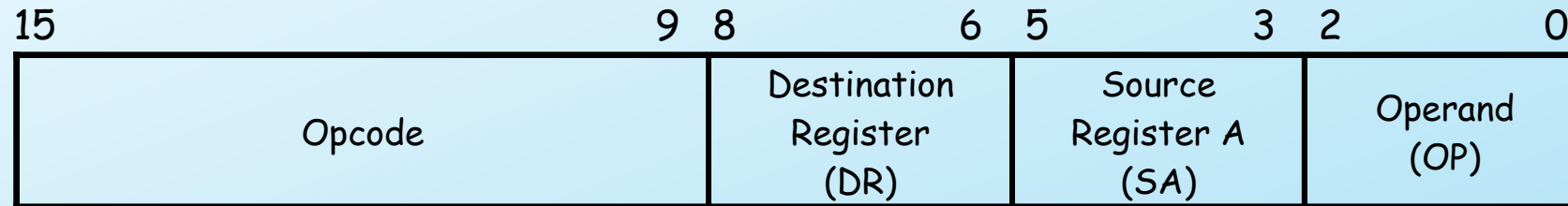
Doua campuri de 3 biți pentru registrele sursa

Aceste tipuri de instrucțiuni sunt valabile pentru
arhitecturi cu pana la 8 registrii

Exemplu

ADD R1, R2, R3

Immediate format



Un camp de 7 biți pentru codul operațiunii

Un camp de 3 biți pentru registrul destinație

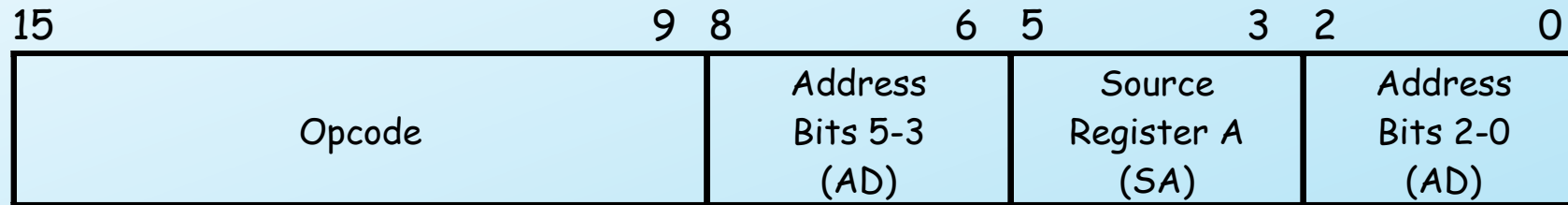
Un camp de 3 biți pentru registrul sursa

Un camp de 3 biți pentru constanta operand

Exemplu

ADD R1, R2, #3

Jump and branch format



Un camp de 7 biți pentru codul operațiunii

Un camp de 3 biți pentru registrul destinație

Un camp (compus din doua segmente) de 6 biți pentru constanta adresa.

Segmentarea acestui camp se face pentru uniformitate.

Exemplu

```
BZ    R3, -24
JMP   18
```

Instruction type
Register-format ALU operation
Register-format shift operation
Memory write (from registers)
Memory read (to registers)
Immediate ALU operation
Immediate shift operation
Conditional branch
Jump

Instruction type	Opcode bits		
	15	14	13
Register-format ALU operation	0	0	0
Register-format shift operation	0	0	1
Memory write (from registers)	0	1	0
Memory read (to registers)	0	1	1
Immediate ALU operation	1	0	0
Immediate shift operation	1	0	1
Conditional branch	1	1	0
Jump	1	1	1

Biții 12-9

Corespund codului de selecție a
funcției ALU

FS	Operation
00000	$F = A$
00001	$F = A + 1$
00010	$F = A + B$
00011	$F = A + B + 1$
00100	$F = A + B'$
00101	$F = A + B' + 1$
00110	$F = A - 1$
00111	$F = A$
01000	$F = A \wedge B$ (AND)
01010	$F = A \vee B$ (OR)
01100	$F = A \oplus B$
01110	$F = A'$
10000	$F = B$
10100	$F = sr\ B$ (shift right)
11000	$F = sl\ B$ (shift left)

Instruction	Bits 15-9 (Opcode)	Bits 8-6	Bits 5-3	Bits 2-0	Format
LD R1 , (R0)	011xxxx	001	000	xxx	Immediate
BZ R1 , +4	110x011	000	001	100	Jump/branch
SUB R5 , R5 , #1	1000100	101	101	001	Immediate
ADD R1 , R0 , R5	0000010	001	000	101	Register
JMP -3	111xxxx	111	xxx	101	Jump/branch

Conversia si executia instructiunilor de catre Unitatea de Control

Unitatea de control convertește o instrucțiune într-un set de semnale recunoscute de:

ALU, registri si alte blocuri funcționale ale unui CPU.

- IR: Instruction Register

- PSR: Processor Status Register

N (negative)

The result of the last ALU operation is negative (MSB = 1)

Z (zero)

The result of the last ALU operation is zero

C (carry)

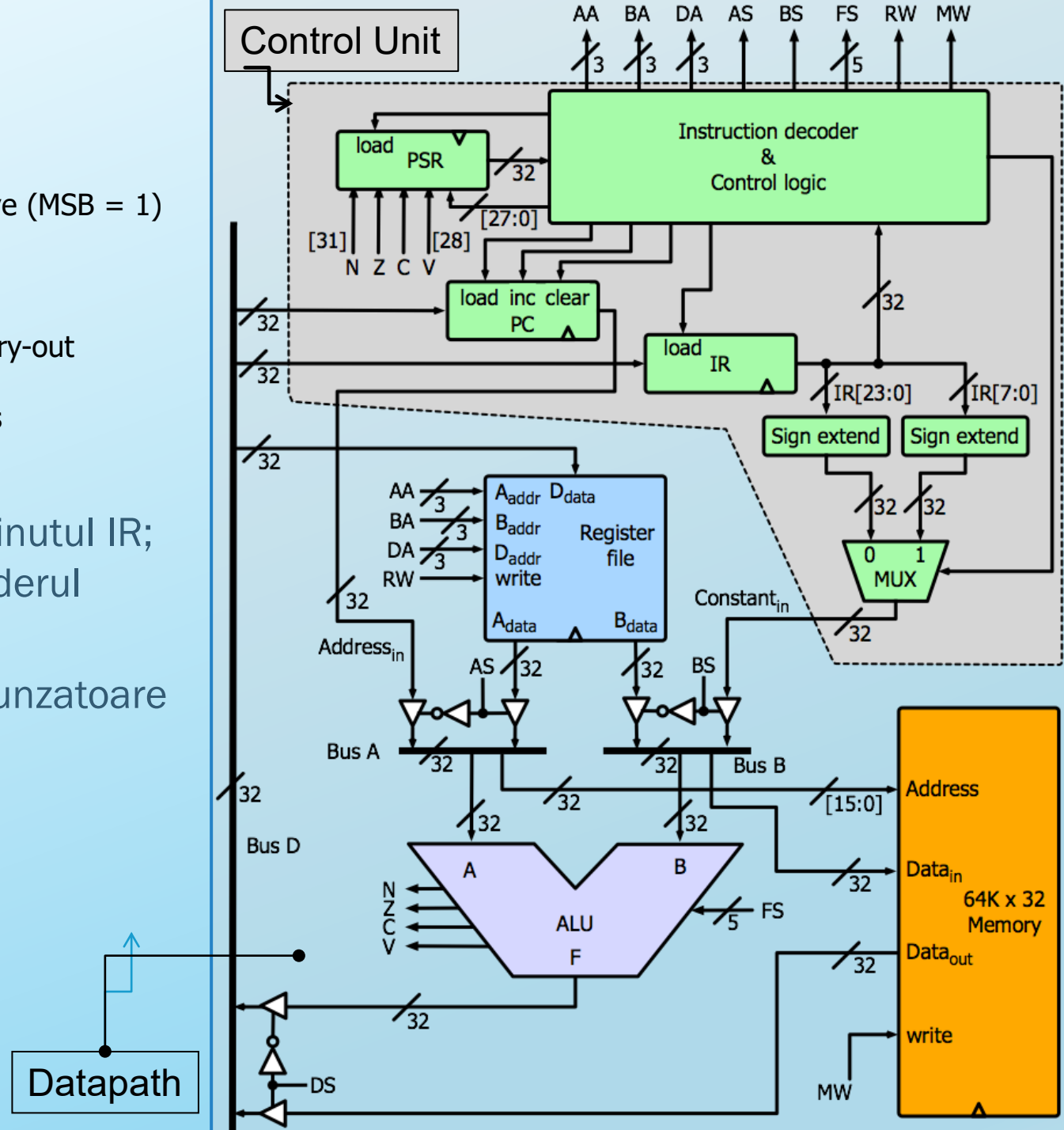
The result of the last ALU operation has a carry-out

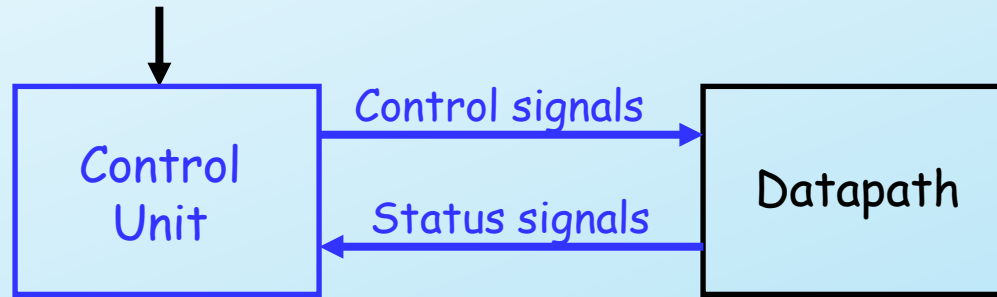
V (oVerflow)

The result of the last ALU operation overflows

Decodorul de instructiuni citeste continutul IR;
sunt decodificati: **opcode** (din headerul
instructiunii) si **operandii**

Apoi sunt generate semnalele corespunzatoare
operatiei specificate de opcode

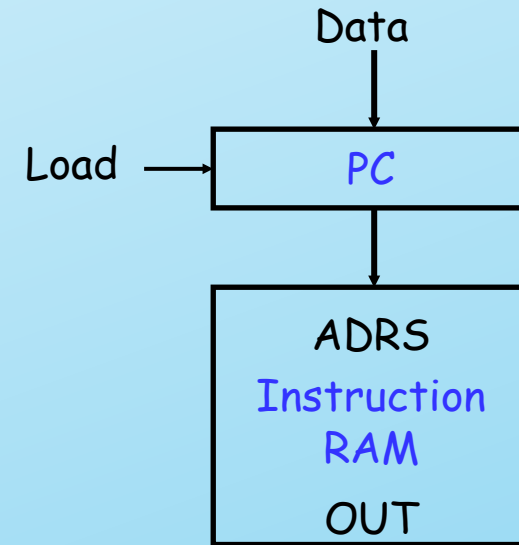




- × Unitatea de control conecteaza programul cu blocul generic "Datapath", care contine ALU, registrii si alte blocuri.
- × Genereaza semnalele: WR, DA, AA, BA, MB, FS, MW, MD.
- × Recepționeaza semnale dinspre Datapath cum ar fi: biții de stare ALU: V, C, N, Z.

Program counter (PC)

- × Executa doua sarcini
- × Load=0
 - + Adresa din PC este incrementata cu 1 (la fiecare ciclu de ceas)
- × Load=1
 - + Adresa din PC este actualizata cu valoarea Data

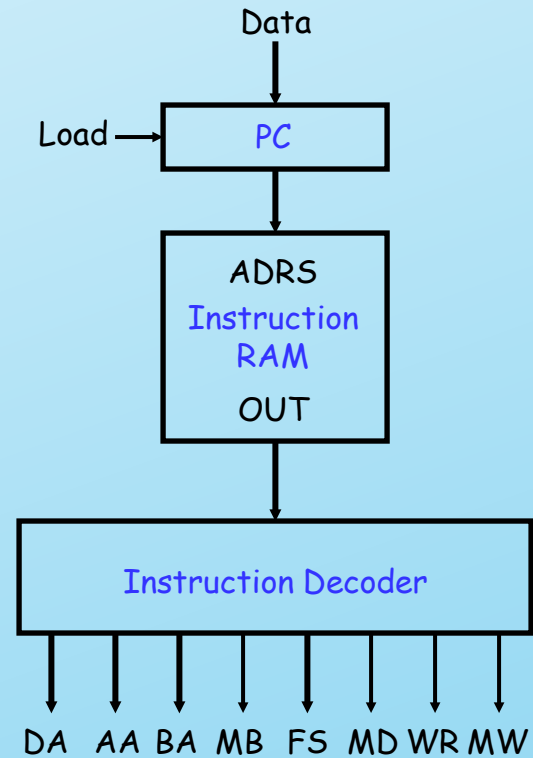


Instruction decoder

circuit combinațional

Genereaza semnale corespunzatoare adresei

Aceste semnale stabilesc ce registri, ce locatie de memorie sau ce operație ALU trebuie efectuată.

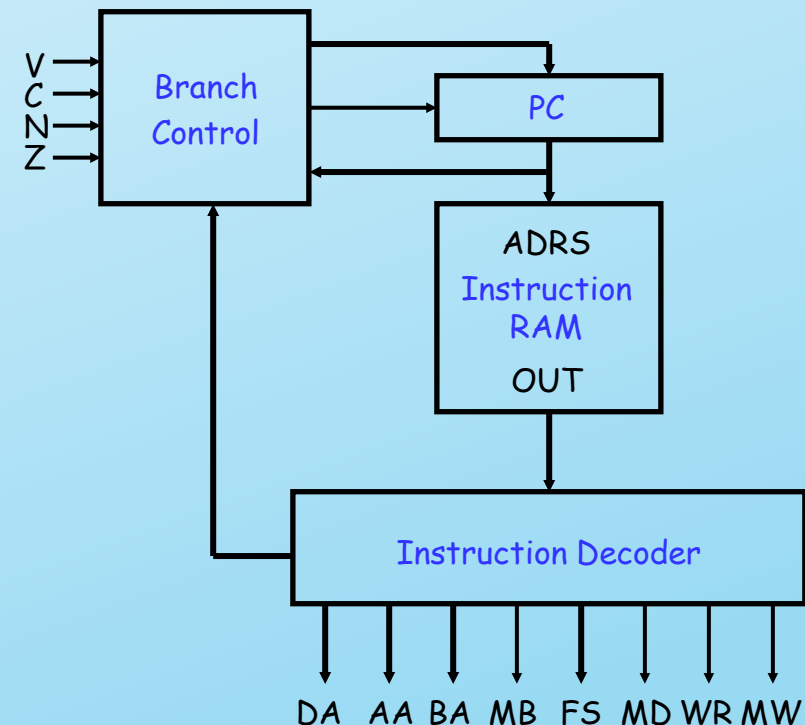


(to the datapath)



Jumps and branches

- ✗ branch control unit
- ✗ Decide ce valoare va fi incarcata in PC
 - + Jumps
 - ✗ PC va fi incarcata cu adresa tinta specificata in instructiune
 - + branch instructions
 - ✗ PC va fi incarcata cu adresa tinta numai daca bitul de stare este TRUE
 - + Pentru alte instructiuni
 - ✗ Este incrementat





Intregul processor

