**Arhitectura Calculatoarelor**

**MIPS**

Apostu Iulian-Eduard

30222

Numarul de aparitii al unui numar intr-un vector

Cod C:

int n=5, count=0, nr = 4;

int a[]= {1, 2, 4, 4, 5};

for(int i = 0; i<n; i++)

{

if(a[i]==nr)

{

count++;

}

}

Asamblare: Cod C:

lw $4, 0($0) int n = 5;

lw $5, 1($0) int nr = 4;

addi $1, $0, 0 int i = 0;

addi $2, $0, 2 //initializam indexul cu care parcurgem memoria

addi $3, $0, 0 int count = 0

beq $1, $4, 6 for(...;i<n;...)

lw $6, 0($2) a[i] //pargurgem, de la mem(2) se gasesc elementele

NoOp

NoOp

bne $6, $5, 1 if(a[i]==nr)

3x NoOp

addi $3, $3, 1 count++; //$3 va fi count+1

addi $1, $1, 1 for(...;...;i++)

addi $2, $2, 1 //incrementam indexul cu care pargurgem memoria

j 5 //sarim la inceputul loop-ului

addi $3, $3, 0 //$3 + 0 sa afisam la final raspunsul

Cod Masina:

B"010\_000\_100\_0000000", --lw $4<=mem(0)

B"010\_000\_101\_0000001", --lw $5<=mem(1)

B"001\_000\_001\_0000000", --$1=i=0

B"001\_000\_010\_0000010", --$2=2 id mem

B"001\_000\_011\_0000000", --$3=0

B"011\_100\_001\_0001011", -- i == n ($4)

B"010\_010\_110\_0000000", -- $6=mem($2)

B"001\_000\_000\_0000000", -- noop

B"001\_000\_000\_0000000", -- noop

B"101\_101\_110\_0000100", --$6!=$5

B"001\_000\_000\_0000000", -- noop

B"001\_000\_000\_0000000", -- noop

B"001\_000\_000\_0000000", -- noop

B"001\_011\_011\_0000001", --duplicate++ $3

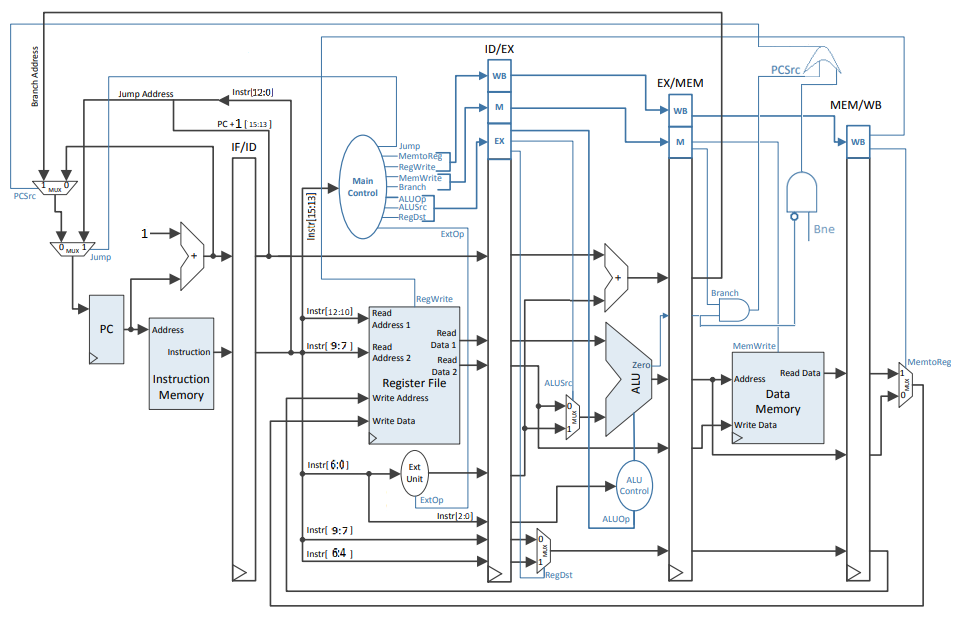
B"001\_010\_010\_0000001", --mem id++

B"001\_001\_001\_0000001", --i++

B"100\_0000000000101", --jump 5

B"001\_011\_011\_0000000", -- raspuns + 0

MIPS PIPE 16 (+ extindere Branch Not Equal):



Pentru a extinde mips-ul cu BNE, avem nevoie de un nou semnal de control, o poarta AND, ce are ca intrare semnalul de control BNE si semnalul de Zero negat de la ALU (se face diferenta si daca semnalul Zero este 0 cele 2 numere sunt diferite si se efectueaza saltul), si o poarta OR, intre Branch si iesirea portii AND de la BNE, a carei iesire este selectia PCSrc.

Diagrama PipeLine:

