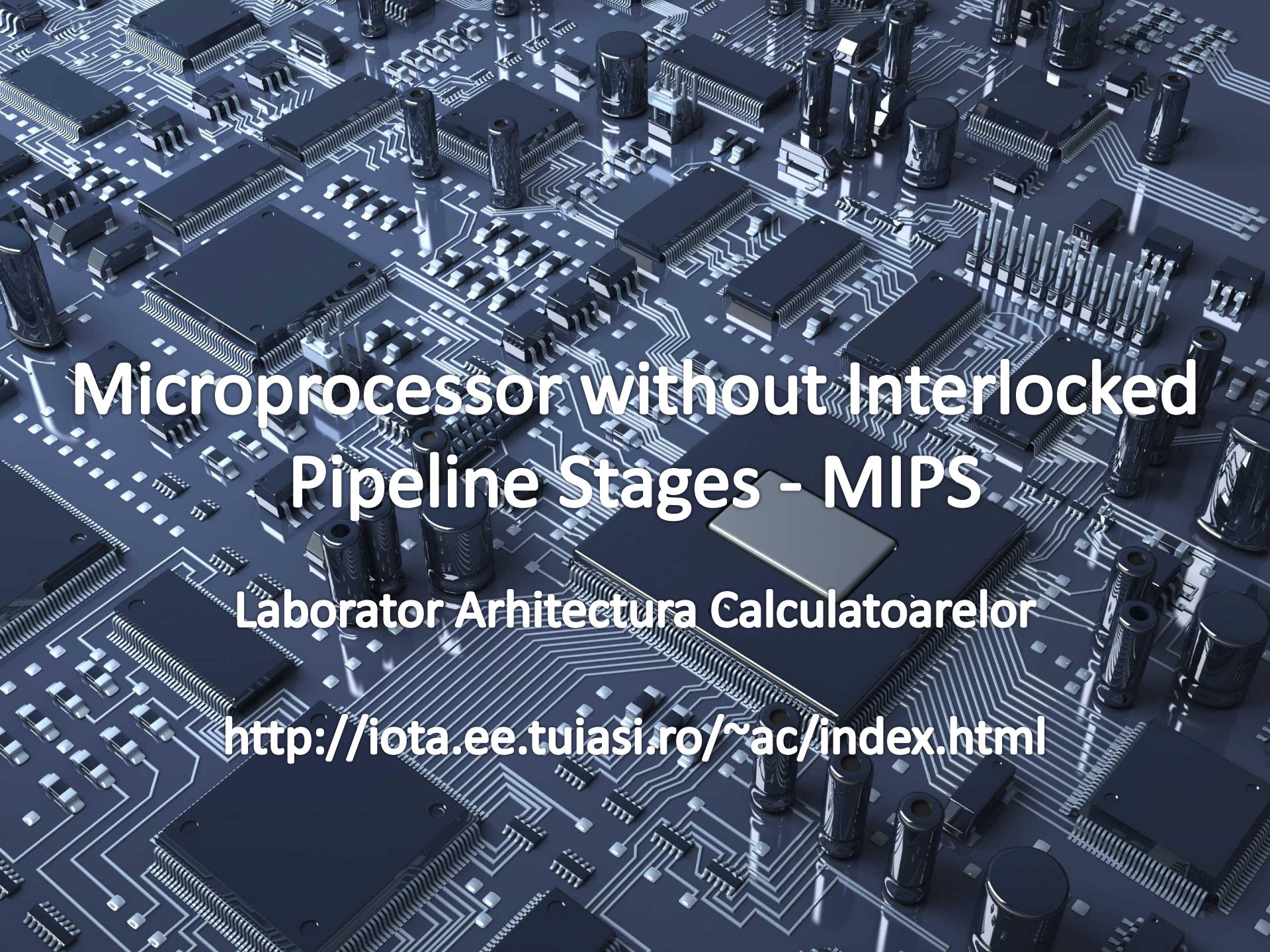


Microprocessor without Interlocked Pipeline Stages - MIPS

Laborator Arhitectura Calculatoarelor

http://iota.ee.tuiasi.ro/~ac/index.html

Microprocesorul de uz general

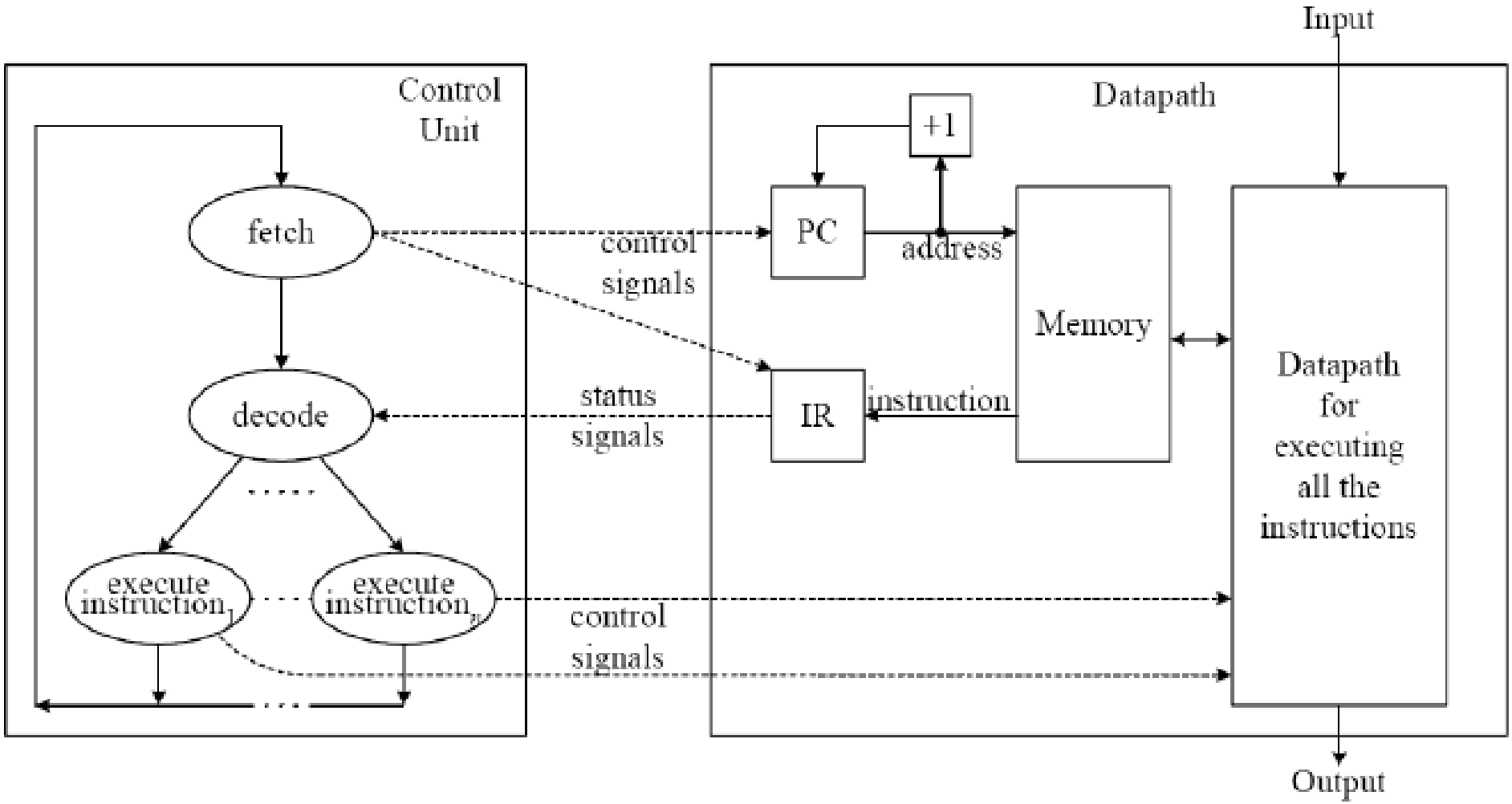
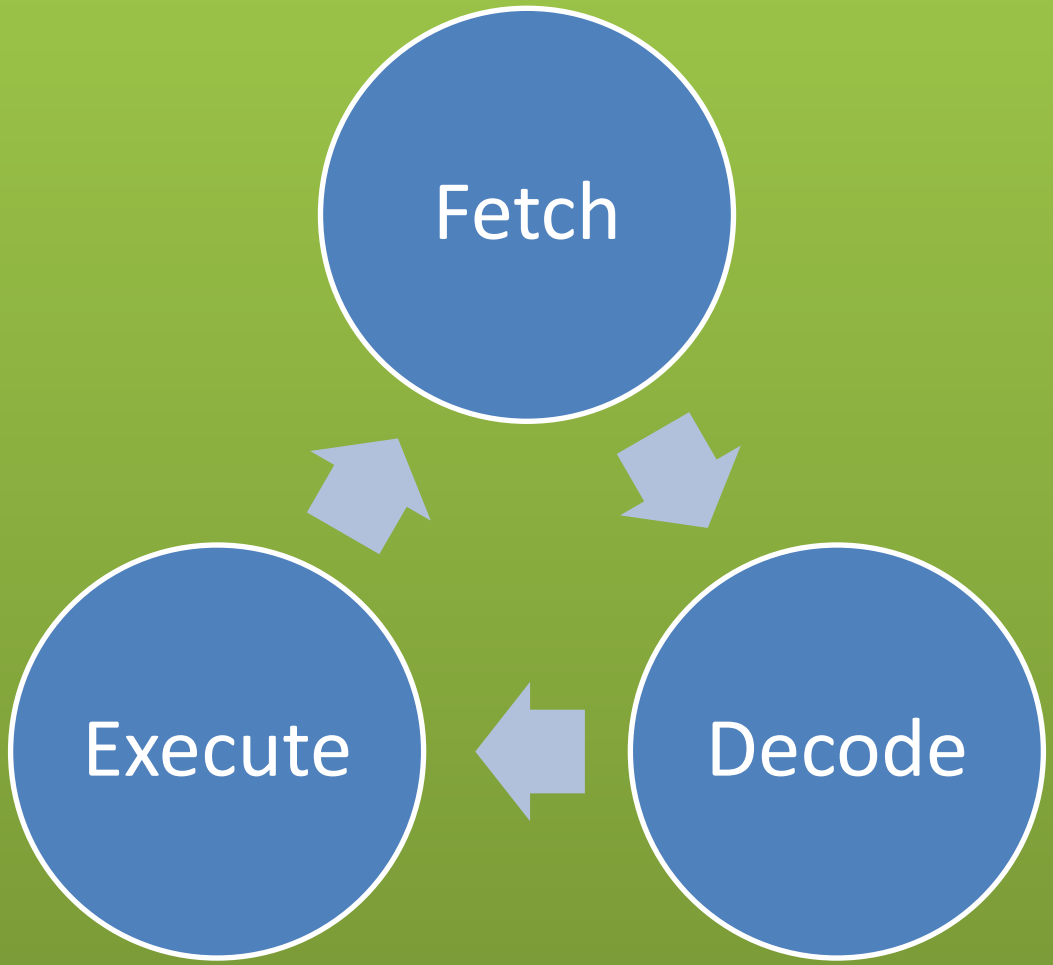


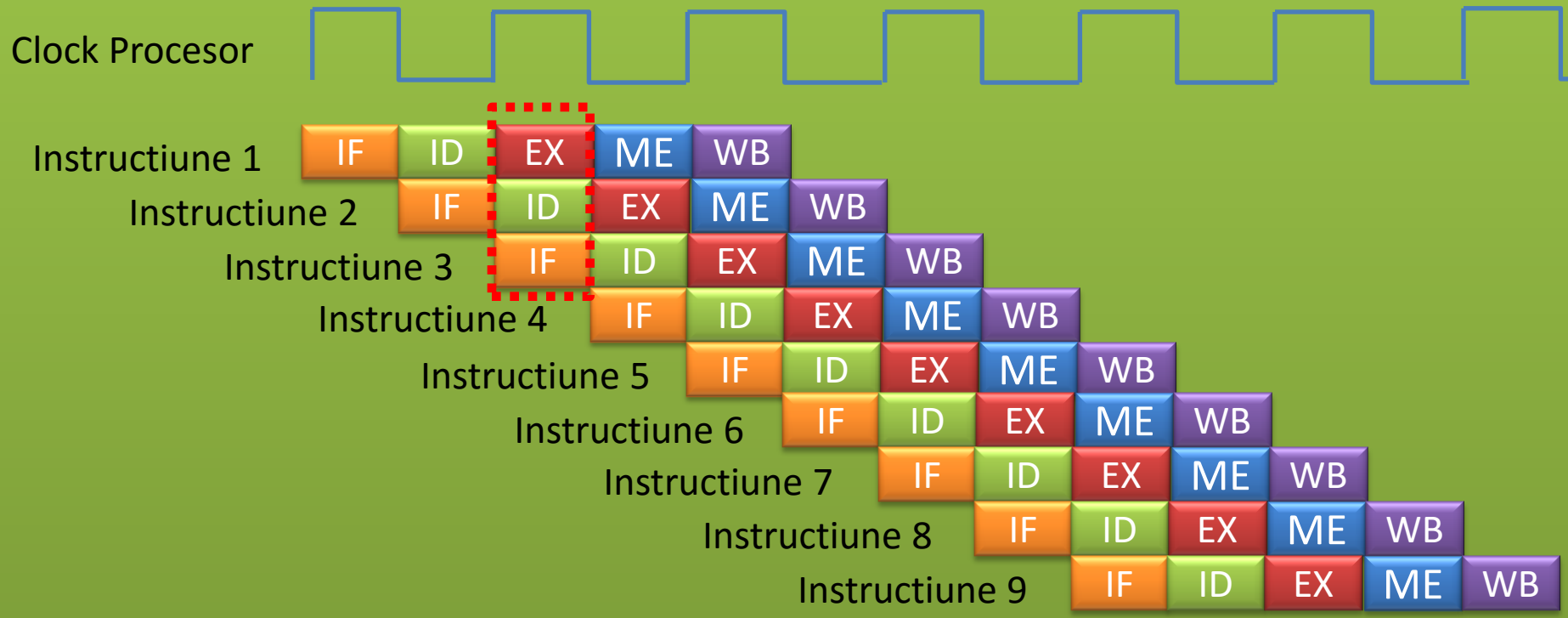
Figura 1. Organigrama de funcționare a unui microprocesor de uz general

Microprocesorul de uz general

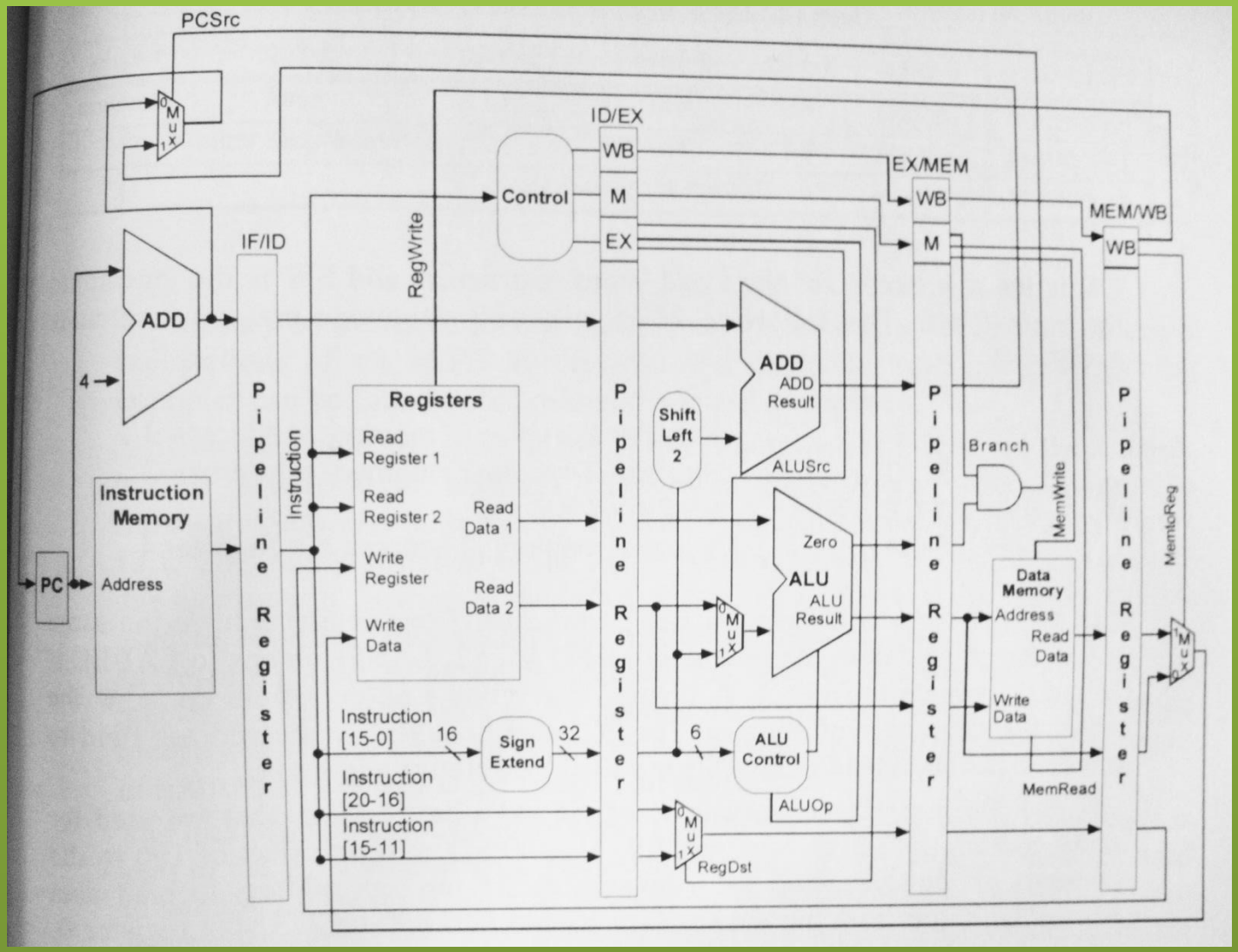


Procesor cu mai multe nivele pipeline

Executarea instructiuni specifica procesorului cu pipeline pe 5 nivele



Arhitectura MIPS
Pipeline pe 5 nivele



Setul de instructiuni

Mnemonic	Format	Opcode Field	Function Field	Instruction
Add	R	0	32	Add
Addi	I	8	-	Add Immediate
Addu	R	0	33	Add Unsigned
Sub	R	0	34	Subtract
Subu	R	0	35	Subtract Unsigned
And	R	0	36	Bitwise And
Or	R	0	37	Bitwise OR
Sll	R	0	0	Shift Left Logical
Srl	R	0	2	Shift Right Logical
Slt	R	0	42	Set if Less Than
Lui	I	15	-	Load Upper Immediate
Lw	I	35	-	Load Word
Sw	I	43	-	Store Word
Beq	I	4	-	Branch on Equal
Bne	I	5	-	Branch on Not Equal
J	J	2	-	Jump
Jal	J	3	-	Jump and Link (used for Call)
Jr	R	0	8	Jump Register (used for Return)

Memoria de Program

Continutul memoriei de program se regaseste in fisierul program.mif

```
-- MIPS Instruction Memory Initialization File
Depth = 256;
Width = 32;
Address_radix = HEX;
Data_radix = HEX;
Content
Begin
-- Use NOPS for default instruction memory values
        [00..FF]: 00000000; -- nop (add r0,r0,r0)
-- Place MIPS Instructions here
-- Note: memory addresses are in words and not bytes
-- i.e. next location is +1 and not +4
        00: 8C020000;       -- lw $2,0 ;memory(00)=55
        01: 8C030001;       -- lw $3,1 ;memory(01)=AA
        02: 00430820;       -- add $1,$2,$3
        03: AC010003;       -- sw $1,3 ;memory(03)=FF
        04: 1022FFFF;       -- beq $1,$2,-4
        05: 1021FFFA;       -- beq $1,$1,-24
End;
```

Memoria de Date

Continutul memoriei de date se regaseste in fisierul dmemory.mif

```
--  MIPS Data Memory Initialization File
depth=256;
width=8;
Content
Begin
-- default value for memory
 [00..FF] : 00;
-- initial values for test program
 00 : 55;
 01 : AA;
End;
```

Simulare functionala