

# DOCUMENTATIE MIPS CICLU UNIC

*Arhitectura Calculatoarelor*

NUME STUDENT Jarda Adina-Ionela  
GRUPA 30222

## **CUPRINS**

1. Precizarea elementelor functionale/nefunctionale
2. Descrierea instructiunilor implementate

## **1. Prezentarea elementelor functionale/ nefunctionale**

### **FUNCTIONALE**

- ✓ Unitatea de executie a instructiunilor (Instruction Execute EX)
- ✓ Unitatea de extragere a instructiunilor (Instruction IFetch IF)
- ✓ Unitatea de memorie (MEM)
- ✓ Generatorul de monoimpuls sincron (MPG)
- ✓ Afisorul pe 7 segmente (Seven Segment Display SSD)
- ✓ Unitatea de control (UC)
- ✓ Decodicatorul de Instructiuni (ID)
- ✓ Registru de fisiere (RegFile)

### **NEFUNCTIONALE**

Nu este cazul.

Functionalitatea programului a fost testata pe placa in timpul laboratorului de testare.

## 2. Descrierea instructiunilor implmentate

### 1 Instructiuni de tip Register (R)

- SRA – Shift-Right Arithmetic

- a) Descriere

- Deplasare aritmetică la dreapta pentru un registru, rezultatul este memorat în altul. Se repetă valoarea bitului de semn

- b) Format

- RTL \$d <= \$t >> h;  
PC <= PC+4

- c) Sintaxă sra

- \$d, \$t, h

- d) Cod mașină

- 000000 00000 ttttt ddddd hhhhh 000011

- e) Semnale Control

- RegDst<=1, RegWrite<=1, ALUOp<=10, ALUCtrl<=111(>>a) 2)

- XOR – bitwise eXclusive-OR

- a) Descriere

- SAU-Exclusiv logic între două registre, memorează rezultatul în alt registru

b) Format

RTL  $\$d \leq \$t \wedge \$t;$   
 $PC \leq PC + 4$

c) Sintaxă xor

$\$d, \$s, \$t$

d) Cod mașină

000000 sssss ttttt ddddd 00000 100110

e) Semnale Control

$\text{RegDst} \leq 1, \text{RegWrite} \leq 1, \text{ALUOp} \leq 10, \text{ALUCtrl} \leq 110(^)$

## 2 Instrucțiuni de tip Immediate

- BGTZ – Branch on Greater than Zero

a) Descriere

Salt condiționat dacă un registru este mai mare ca 0

b) Format

If  $\$s > 0$  then  $PC \leq (PC + 4) + (\text{SE}(\text{offset}) \ll 2)$   
else  $PC \leq PC + 4;$

c) Sintaxă bgtz

$\$s, \text{offset}$

d) Cod mașină

000111 sssss 00000 oooooooooooooooooo

e) Semnale Control

ExtOp<=1, Br\_gtz<=1, ALUOp<=01(-), ALUCtrl<=100(-)

- ANDI – And Immediate

a) Descriere

SI logic între un registru și o valoare imediată, memorează rezultatul în alt registru

b) Format

RTL \$t <= \$s | ZE(imm);  
PC <= PC + 4;

c) Sintaxă

andi \$t, \$s, imm

d) Cod mașină

001100 sssss tttt iiiiiiiiiiiiii

e) Semnale Control

ALUSrc<=1, RegWrite<=1, ALUOp<=11, ALUCtrl<=010(&)

Semnalele de control care nu sunt menționate la fiecare instrucțiune se consideră implicit 0.