Documentație MIPS Pipeline

Arhitectura Calculatoarelor

Student: Clonța Ștefania-Elena

Grupa 30222

Profesor coordonator: Cireap Dragoș Gabriel

**Cuprins**

1. Precizarea elementelor funcționale și nefuncționale
2. Descrierea tuturor instrucțiunilor implementate
3. **Precizarea elementelor funcționale și nefuncționale**

Elementele funcționale:

* Unitatea de extragere a instrucțiunilor (IFetch)
* Unitatea de control (UD)
* Unitatea de decodificare a instrucțiunilor (ID)
* Unitatea de execuție a instrucțiunilor (EX)
* Unitatea de memorie (MEM)
* Generatorul monopuls sincron (MPG)
* Afișorul pe 7 segmente (SSD)

Elemente nefuncționale

* Nu există

Programul a fost testat în cadrul laboratorului pe o placuță Nexys dar și pe una Basys în timpul liber.

Acest microprocesor nu a suferit modificari majore fata de MIPS ciclu unic, componentele ID si ED au fost modificate putin, iar in test\_env s-au declarat semnale suplimentare pentru etajele aferente.

1. **Descrierea tuturor instrucțiunilor implementate**
2. ***Instrucțiunile de tipul Register***
3. Shift Right Arithmetic (SRA)

Această instrucțiune realizează o deplasare aritmetică spre dreapta pentru un registru, rezultatul fiind stocat în altul.

**Format RTL**

$d <= $t >> h

PC <= PC + 4

**Sintaxă**

Sra $d, $t, h

**Codul mașină**

000000 00000 ttttt ddddd hhhhh 000011

**Semnale de control**

RegDst <= 1

RegWrite <= 1

ALUOp <= 10

ALUCtrl <= 111

1. Bitwise eXclusive-OR (XOR)

Această instrucțiune realizează sau-exclusiv între două registre și memorează rezultatul în alt registru.

**Format RTL**

$d <= $t ^ $t

PC<=PC+4

**Sintaxă**

xor $d, $s, $t

**Codul mașină**

000000 sssss ttttt ddddd 00000 100110

**Semnale de control**

RegDst<=1

RegWrite<=1

ALUOp<=10

ALUCtrl<=110

1. ***Instrucțiuni de tip Immediate***

a) BGEZ – Branch on Greater than or Equal to Zero

**Descriere**

Salt condiționat dacă un registru este mai mare sau egal cu 0

**Format RTL**

if $s ≥ 0 then

PC <= (PC + 4) + (SE(offset) << 2)

else

PC <= PC + 4;

**Sintaxă**

bgez $s, offset

**Cod mașină**

000001 sssss 00000 oooooooooooooooo

**Semnale Control**

ExtOp<=1

Br\_gez<=1

ALUOp<=01(-)

ALUCtrl<=100(-)

1. ORI – bitwise OR Immediate

**Descriere**

Această instrucțiune realizează SAU logic între un registru și o valoare imediată, iar rezultatul se memorează în alt registru.

**Format RTL**

$t <= $s | ZE(imm)

PC <= PC + 4;

**Sintaxă**

ori $t, $s, imm

**Cod mașină**

001101 sssss ttttt iiiiiiiiiiiiiiii

**Semnale Control**

ALUSrc<=1

RegWrite<=1

ALUOp<=11

ALUCtrl<=010(|)

Restul semnalelor care nu au fost menționate, primesc implicit valoarea 0.