

MIPS Single Cycle 32 bits

Problema proiectului:

Sa se determine daca un sir de N elemente este sortat crescator sau nu.

Componente functionale:

IFETCH:

Memoria ROM este initializata cu codificarea binara a instructiunilor ce rezolva problema. IFETCH reprezinta unitatea de preluare a instructiunilor pentru procesor.

Problema intampinata de mine a fost in cadrul instructiunilor de salt, insa am reusit sa o remediez.

MPG:

MPG este mecanismul de generare a unui semnal bazat pe un semnal de ceas si pe starea unui buton

SSD:

Utilizat pentru afisarea cifrelor pe un display cu sapte segmente.

UC:

UC implementeaza un decodificator care are intrare *Instruction* si semnalele de control ca iesiri. Sunt actualizate doar semnalele de control care trebuie sa fie diferite de 0 pentru instructiunea respectiva.

ID:

ID decodifica instructiunile primite si pregrateste datele necesare pentru executia lor. Sunt generate semnale pentru citire si scriere in registre, extindere a imediatelor si gestionarea altor operatii.

EX:

Componenta responsabila pentru executia operatiilor aritmetice si logice in cazul instructiunilor, precum si calcularea adresei de branch.

MEM:

MEM este componenta responsabila pentru gestionarea accesului la memoria de date a sistemului. Aceasta primeste datele de la EX, inclusiv rezultatele operatiilor ALU si datele citite din registre, urmand ca apoi sa efectueze operatii de scriere in memorie daca este necesar.

O problema cu care am avut de lucru a fost scrierea la o adresa gresita. Aceasta problema a fost rezolvata ulterior.

CODUL A FOST TESTAT PE PLACUTA

Instructiuni suplimentare:

TIP I:

ORI:

Sau logic intre un registru si o valoare imediata. Rezultatul se memoreaza in alt registru

- RTL: $\$t \leq \$s \mid ZE(imm); PC \leq PC + 4$
- sintaxa: ori \$t, \$s, imm
- format: 001101 sssss tttt iiiiiiiiiiiiii

ADDI:

Aduna un registru cu o valoare imediata si memoreaza rezultatul in alt registru

- RTL: $\$t \leq \$s + SE(imm); PC \leq PC + 4$
- sintaxa: addi \$t, \$s, imm
- format: 001000 sssss tttt iiiiiiiiiiiiii

BNE:

Salt conditionat daca doua registre sunt diferite

- RTL: if $\$s \neq \t then $PC \leq (PC + 4) + (SE(offset) \ll 2)$ else $PC \leq PC + 4$
- sintaxa: bne \$s, \$t, offset
- format: 000101 sssss tttt oooooooooooooo

TIP R:

SLT:

Daca $\$s < \t , atunci \$d este initializat cu 1 altfel cu 0

- RTL: $PC \leq PC + 4; \text{if } \$s < \$t \text{ then } \$d \leq 1 \text{ else } \$d \leq 0;$
- sintaxa: slt \$d, \$s, \$t
- format: 000000 sssss tttt dddd 00000 101010

Observatii:

AluCtrl:

- 000 (+) → in ALU are loc o adunare
- 001 (|) → in ALU are loc o operatie de sau-logic
- 111 (<) → in ALU are loc o operatie de comparare (pentru instructiunea SLT)