Documentatie procesor MIPS pe 32 de biti

CIOBANU RADU-RARES
GRUPA 30222

Elemente functionale:

Monopulse Generator(MPG):

- Ajuta la identificarea momentului in care butonul pentru afisajul Seven Segment Decoder, este apasat, pentru a trece la urmatoarele instructiuni.

Seven Segment Display(SSD):

- Cu ajutorul SSD–ului suntem capabili sa vedem rezultatele executiei procesorului, prin cele 8 afisaore disponibile pe Nexys.

Procesor:

- Este entitatea in care toate componentele sunt conectate in mod corespunzator, si responsabila cu executia programului.

ROM (Read – Only Memory):

- Memoria ROM, este folosita, exact asa cum este numita, adica permite programului doar sa citeasca date, acele date fiind instructiunile pe care procesorul trebuie sa fie capabil sa le execute fara probleme. Este pozitionat in componenta IFetch.

Register File (RegFile):

- Register File, adica blocul de registii, este folosit pentru a stoca informatiile rezultate din executia programului, si ulterior folosite mai departe in executie.

IFecth(Instruction Fetch):

- Instruction Fetch, primeste adresa curenta PC, cu instructiunile respective de la acea adresa, si genereaza adresa urmatoare, adica PC + 4, si transmite mai departe instructiunile generate.

- Adresa umratoare este data de urmatoarele conditii:
  - Dacă Jump = 1, atunci PC ; ← Jump Address;
  - Dacă Jump = 0, atunci:
    - Dacă PCSrc = 1, atunci PC← Branch Address;
    - Dacă PCSrc = 0, atunci PC ← PC+4.

Instruction Decoder (ID):

- Componenta Instruction Decoder, primeste ca intrari 3 flag-uri din Main Control Unit, RegWrite, RegDst, ExtOp, cu instructiunile respective, un semnal de Enable, cat si datele care trebuie scrise la adresa data de un multiplexor, care specifica daca se scrie in registrul destinatie, sau in registrul target.

- Datele rezultate sunt transmise mai departe catre Unitatea de Executie, adica, datele citite de la adresele respective, functia ce trebuie efectuata, valoarea de shift(daca exista), si imediatul extins.

Unitatea principala de control (Main Control-Unit):

- Componenta Main Control-Unit, este responsabila pentru detectarea tipului de registru, R, I sau J, si pentru generarea semnalelor, RegDst, ExtOp, ALUSrc, Branch, Jump, ALUOp (pe 3 biti), MemWrite, RegWrite, MemtoReg, corespunzator functiilor pe care procesorul trebuie sa le execute.

CIOBANU RADU-RARES
GRUPA 30222

Unitatea de executie (EX):

- Realizeaza opratii aritmetice si logice in functie de datele primite de la Instruction Decoder.
- Datele de intrare sunt date de continutul citit din Register File, Imediatul extins, ALUSrc care decide daca imediatul extins sau RD2, participa la operatia cate trebuie efectuata, funtia, cu shift amount, si se calculeaza adresa de branch si rezultatul din ALU, care sunt prezentate pe iesiri.

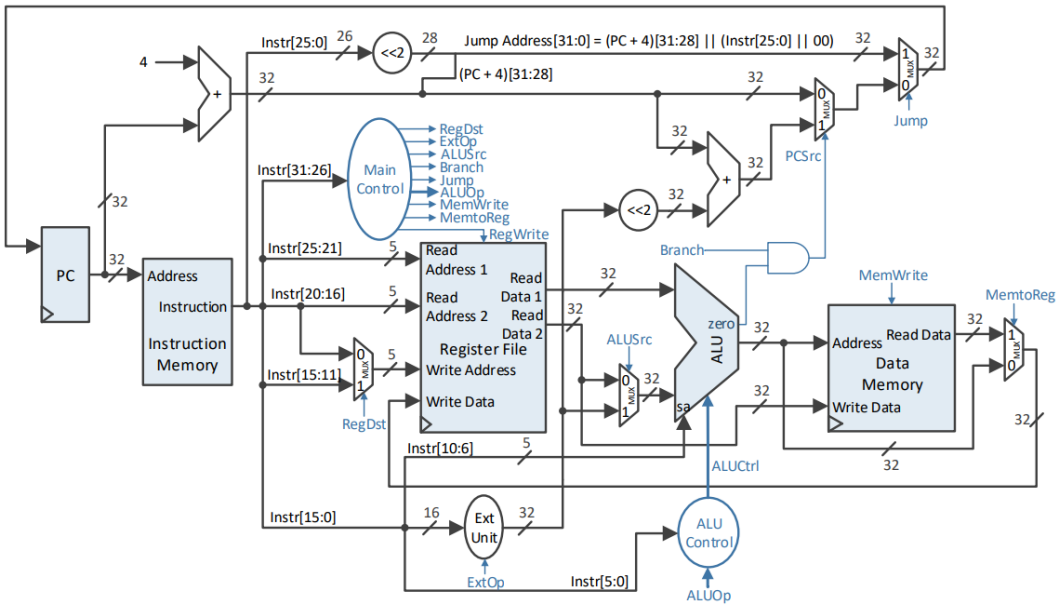
Unitatea de memorie:

- Unitatea de memorie, se prezinta ca o memorie RAM, in care se scriu datele citite(RD2), cand MemWrite este ‘1’ , la adresa data de ALURes, iar ca iesiri are rezultatul din ALU, si datele citite, numite MemData.

Write-Back (WB):

- Write Back se refera la selectarea continutului pe care vrem sa il scriem in Reg File, astfel cand MemToReg = ‘1’ se vor scrie datele MemData, iar cand este MemToReg = ‘0’ se vor sc vor scrie ALURes.

Schema logica:



CIOBANU RADU-RARES
GRUPA 30222

Instructiuni alese:

XOR – bitwise eXclusive-OR

- SAU-Exclusiv logic între două registre (sursa si target), memorează rezultatul in registrul destinație si incrementează program counter-ul
- $d ← $s ^ $t; PC ← PC + 4;
- xor $d, $s, $t
- 000000 sssss ttttt ddddd 00000 100110

SLT – Set on Less Than (signed)

- Instrucțiunea SLT compară două registre (sursa și target) și setează un al treilea registru (destinatie) la valoarea 1 dacă $s este mai mic decât $t și la valoarea 0 în caz contrar
- PC←PC + 4; if $s < $t then $d ← 1 else $d ← 0;
- slt $d, $s, $t
- 000000 sssss ttttt ddddd 00000 101010

ORI – bitwise OR Immediate

- Valoarea din registrul sursa este combinată folosind operația logică SAU cu o valoare imediată iar rezultatul este stocat în registrul $t, în timp ce program counter este incrementat pentru a trece la următoarea instrucțiune din cod în alt registru
- $t ←$s | ZE(imm); PC ←PC + 4;
- ori $t, $s, imm
- 001101 sssss ttttt iiiiiiiiiiiiiiii

SLTI – Set on Less Than Imediate (signed)

- Dacă $s este mai mic decât un imediat, $t este inițializat cu 1, altfel cu 0
- PC←PC + 4; if $s < SE(imm) then $t ← 1 else $t ← 0;
- slti $t, $s, imm
- 001010 sssss ttttt iiiiiiiiiiiiiiii

CIOBANU RADU-RARES
GRUPA 30222

Precizari:

Toate componentele au fost testate pe placuta, si au fost functionale. Totusi exista posibilitatea unei neregului legate de programul ales, datele probabil fiind eronate in unele cazuri. Tabelele, au fost puse toate in pdf-ul Tabele.pdf cu trasarea executiei si instructiunile alese.

Probleme legate de executie:

Probabil exista probleme mici in modul in care am scris cod, care sunt greu de gasit, si de rezolvat, dar se vor rezolva si acelea.

CIOBANU RADU-RARES
GRUPA 30222