

Enunț

Să se proiecteze și implementeze un procesor de tip RISC pe 8 biți în conformitate cu noțiunile prezentate la curs - procesorul RISC pe 32 de biți cu execuția instrucțiunilor într-un singur ciclu de ceas.

Specificații de proiectare

1. Un ciclu de ceas este de dorit să conțină 4 cicluri de ceas. Această cerință presupune implementarea unui divizor de ceas (exemplu: lucrarea de laborator TRANSMISIA - RECEPȚIA SERIALĂ A INFORMAȚIEI). Frecvența ceasului plăcii FPGA se va considera de 50MHz.
2. Procesorul va avea un port input/output de 8 biți conectat direct la registrul 3 REG3.
3. Magistralele de date se vor considera pe 16 biți.
4. OPCODE-ul va avea dimensiunea de 5 biți (cei mai semnificativi 5 biți). Excepție fac instrucțiunile MOVE care au OPCODE-ul pe 2 biți, iar bitul MSB este întodeauna 1.
5. Magistrala de adrese are 10 biți și permite adresarea a 1024 de linii de program aflate într-o memorie ROM externă.
6. Toate instrucțiunile JUMP trebuie efectuate într-un singur ciclu de ceas
7. Va exista o stivă pe patru nivele (de 10 biți) folosită pentru memorarea conținutului PC-ului pe durata unei întreruperi => că putem avea până la 4 întreruperi.
8. Vor exista 3 tipuri de întreruperi:
 - a. întrerupere externă nemascabilă
 - b. întrerupere mascabilă primită prin port - după îndeplinirea unor condiții ca de exemplu configurarea direcției portului.
 - c. întrerupere software pentru chemarea unui subprogram
9. Setul de instrucțiuni al procesorului

Instrucțiuni	Cuvântul de date	Explicații
SBC Ri	0000xxxxxxRi4Ri3Ri2Ri1Ri0	Scădere cu transp și memorare în R0
ADC Ri	00001xxxxxxRi4Ri3Ri2Ri1Ri0	Adunare cu transp și memorare în R0
AND Ri	00010xxxxxxRi4Ri3Ri2Ri1Ri0	AND logic și memorare în R0
OR Ri	00011xxxxxxRi4Ri3Ri2Ri1Ri0	OR logic și memorare în R0
XOR Ri	00100xxxxxxRi4Ri3Ri2Ri1Ri0	XOR logic și memorare în R0
BZ Addr zero este activ	00101xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0	Încarcă PC dacă flag-ul
BNZ Addr zero nu este activ	00110xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0	Încarcă PC dacă flag-ul
BC Addr carry este activ	00111xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0	Încarcă PC dacă flag-ul
BNC Addr BZ Addr carry nu este activ	01000xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0	Încarcă PC dacă flag-ul

BN Addr 01111xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0 negativ este activ	Încarcă PC dacă flag-ul negativ este activ
BNN Addr 01001xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0 negativ nu este activ	Încarcă PC dacă flag-ul negativ nu este activ
SHL Ri 01010xxxxxxRi4Ri3Ri2Ri1Ri0 poziție	Deplasare stanga 1
SHR Ri 01011xxxxxxRi4Ri3Ri2Ri1Ri0 poziție	Deplasare dreapta 1
RTS 01100xxxxxxxxxxxxx valoare din stivă	Încarcă PC cu ultima valoare din stivă
CALL Addr 01101xPC9PC8PC7PC6PC5PC4PC3PC2PC1PC0	Încarcă direct PC
STOP 01110xxxxxxxxxxxxx până se execută întreruperea	STOP toate procesele până se execută întreruperea
MOV Ri, Const 10Ri4Ri3Ri2Ri1Ri0xC7C6C5C4C3C2C1C0 într-un registru	Încarcă o constantă într-un registru
MOV Ri, Rj 11Ri4Ri3Ri2Ri1Ri0xxxxRj4Rj3Rj2Rj1Rj0 într-un registru	Copiează un registru într-un registru

10. Blocul de registre conține 32 de registre. Primele 7 registre sunt registrele de sistem, restul registrelor sunt registre generale.

REG0 este registrul acumulator

REG1 este registrul pointer adrese - memorează în cei 5 biți LSB adresa registrului către care pointăm

REG2 - registru peek - are aceeași valoare cu cea a registrului a cărui adresă este memorată în REG1

REG3 - portul input/output sau în anumite condiții primește o întrerupere mascabilă

REG4 - configurează direcția portului REG3

REG5 - registru folosit de întreruperile care sosesc prin portul A (REG3)

REG6 - CCR (conditional code register)

REG6[0] - mask interrupt flag

REG6[1] - flag-ul ZERO

REG6[2] - carry bit

REG6[3] - flag-ul negativ

Cerințe:

1. Realizarea schemei de principiu a procesorului în care apar ca blocuri toate modulele componente. Trebuie explicat ce rol are fiecare bloc. (0.25p)

2. Realizarea schemei de proiectare în variantă TOP-DOWN (0.25p)

3. Realizarea codului VERILOG pentru fiecare modul prezentat la punctul 2. Se va prezenta și simularea modulului respectiv având grijă să se prezinte toate cazurile. (0.5p)

4. Realizarea simulării în Verilog a întregului procesor. (0.5p)

5. Realizarea simulării a unui program impus având în vedere plăcile de dezvoltare prezente în laborator. Memoria ROM trebuie să fie o memorie externă. Programul va fi introdus de la tastatură și afișarea se va face pe un monitor extern. (1.5p)