Subiect : Procesor

Timp de lucru : 120min

Descriere: Circuitul Din figura este un processor simplu, asemanator (dar semnificativ simplificat) cu cele studiate la amp/microcontrollere. El este alcatuit dintru Unitate aritmetica si logica (alu) si 2 memorii, una pentru datele ce sunt prelucrate si una pentru instructiuni. Memoria de instructiuni este parcursa folosind un counter (program counter), iar rezultatele finale ale calculului sunt afisate pe un display cu 7 segmente.

Cerinta: Implementati circuitul descris mai sus folosind diagrama de mai jos, unde :

Data RAM = Memorie de tip RAM cu citire asincrona pe ambele porturi de citire.

Instruction RAM = Memorie de tip RAM cu citire sincrona pe portul de citire. Avand o singura intrare pentru adresa, aceasta se foloseste atat pentru scriere cat si pentru citire.

Mux = Multiplexor cu 2 intrari pe 4 biti.

Counter = Numarator folosit ca program counter. Dorim sa se treaca la urmatoarea instructiune la aproximativ 2 secunde. Pentru a obtine asta se conecteaza bitii [29:27] ai registrului de numarare catre iesirea pe 3 biti.

Alu = Unitatea aritmetica si logica ce are rolul de a efectua propriu zis operatia dorita. Urmatoarele instructiuni sunt valide:
00 => in0 + in1
01 => in0 – in1
10 => in0 & in1
11 => iesirea ia valoarea 0;

Transcoder = transcodor cu rolul de a transforma sistemul de reprezentare binar in reprezentarea specifica display-ului cu 7segmente.

Pentru simulare:

clock : se schimba la 2 unitati de timp

rst_n : (logica negativa)valoare initiala 0; dupa 100 de unitati de timp devine 1;

write_enable_data_mem_n (logica negativa)

write_enable_instr_mem_n : (logica negativa)

data_write (pentru ambele memorii)

Simularea se va opri dupa 1milion de unitati de timp

Operatiile dorite:
Memoria de date :

1) dupa 1000 de unitati de timp: scriu in memoria de date la adresa 0 valoarea 7.
2) dupa inca 100 de unitati de timp: scriu in memoria de date la adresa 1 valoarea 8.
3) setez adresele de citire din memoria de date la valoarea 0 si 1.
4) dupa inca 1000 de unitati de timp: dau valoarea 1 lui mux_select si setez adresa de scriere in memoria de date la adresa 5.
5) dupa inca 100 de unitati de timp activez write_enable_n pentru un ciclu de ceas.

Memoria de instructiuni :
1) la 1500 de unitati de timp de la inceperea simularii se scrie in memoria de instructiuni (la adresa 0 ca acolo e program counter-ul) valoarea 0.

Constrangeri:

clock : 50 MHz

rst_n (logica negativa) : Button 0

write_enable_data_mem_n (logica negativa) : Button 1

addr_read_a (data mem) : Switches [1:0]

addr_read_b (data_mem) : Switches [3:2]

addr_write (data mem) : Switches [5:4]

mux_select : Button 3

write_enable_instr_mem_n (logica negativa) : Button 2

data_write (both memories) : Switches [9:6]

counter_value : Leds Red [3:0]

transcoder_out : Display cu 7 segmente numarul 0

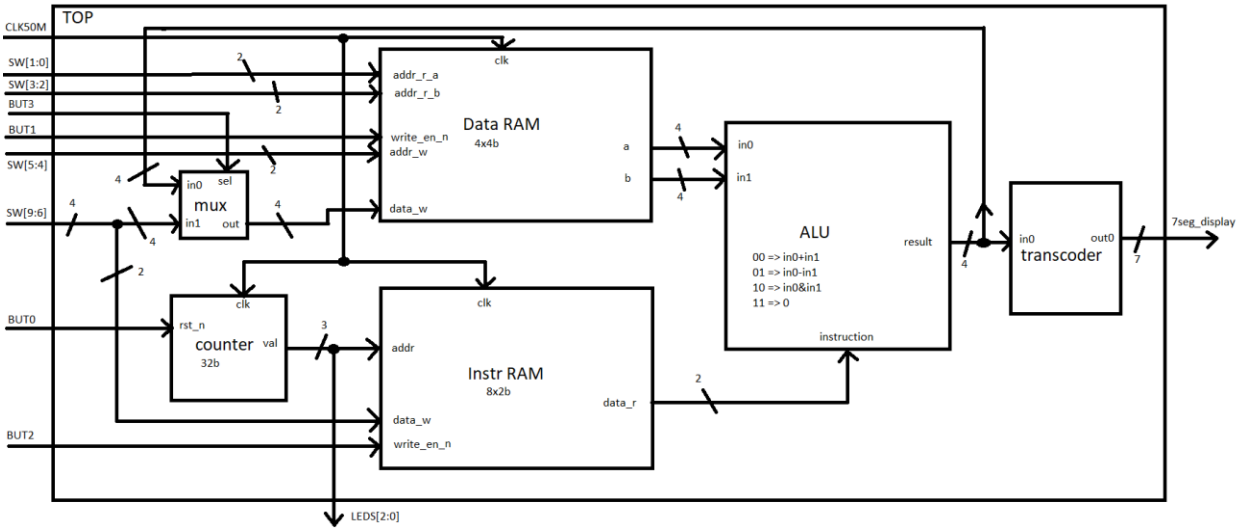
Mentiuni :

Intrarea de date este comuna celor doua memorii (daca ignoram muxul si bucla de reactive). Ca sa scriem lucruri diferite in ele intai scriem in prima memorie, apoi in a doua.

Cum cele 2 memorii au interfete diferite, trebuie obligatoriu scrise 2 module distincte.

Interfetele trebuie sa corespunda cu figura data.

Denumirile sa fie logice si sugestive (fisiere, module, fire, instante, interfete).



Barem: (total 30)

Top – 6p

Data RAM – 2p

Instruction RAM – 2p

ALU – 2p

Counter – 1p

Mux - 1p

Transcoder – 0p (se da)

Testbench – 4p

Simulare – 4p

Constrangeri – 3p

Functionalitate finala, demonstratie pe FPGA – 5p