Tema bonus

1. Sa se implementeze in limbaj de asamblare compatibil RV32I un program ce realizeaza sortarea in ordine crescatoare a cinci numere folosind tehnica de sortare bubble sort.

   Cele cinci numere care vor fi incarcate in memoria de date incepand cu adresa 0x0 sunt: 5, 2, 9, 1, 4. Pentru incarcarea numerelor in memorie se poate folosi combinatia de instructiuni:

   addi x5, x0, <a[1]> #incarca in resgistrul x5 elementul de pe pozitia 1 din array
   sw x5, 4(x0) #salveaza valoarea lui x5 la pozitia 1 din array

   Pentru parcurgerea elementelor vectorului, trebuie mereu realizata adresarea multipla cu 4 intru-cat fiecare cuvant din memeorie se afla la 4 octeti de elementul anterior.

   Pentru saltul la diferitele etichete definite in program se poate folosi instructiunea de branch cu urmatoarea sintaxa: beq x0,x0,<ETICHETA>. Acest lucru face posibil folosirea instructiunii de branch pe post de intructiune de jump.

   Pentru a testa corectitudinea programului inainte de simulare, puteti folosi interpretatorul RISC-V de aici.

2. Sa se scrie codul masina in format hexazecimal, echivalent codului de asamblare de la punctul anterior.

3. Sa se simuleze codul hexazecimal de la punctul anterior pe procesorul RISC-V realizat la laborator.

   Inainte de simulare, trebuie sa se asigure faptul ca toate instructiunile folosite la punctele anterioare pot fi executate corect de catre procesor.

   Pentru aceasta, modulele Verilog descrise, precum imm_gen, ALU, si ALUcontrol trebuie sa poata parsa corect aceste instructiuni, iar unde este cazul aceste module trebuiesc imbunatatite pentru a suporta instructiunile necesare.

   In momentul in care incarcati datele initiale ale vectorului in memoria de date, simularea din Vivado ar trebui sa contina urmatoarea regiune de memorie:

Memory Address	Decimal	Hex
0x00000000	5	0x00000005
0x00000004	2	0x00000002
0x00000008	9	0x00000009
0x0000000c	1	0x00000001
0x00000010	4	0x00000004

Dupa simularea intregului program, aceeasi regiune de memorie ar trebui sa contina elementele vectorului sortate in ordine crescatoare ca in imaginea de mai jos:

Memory Address	Decimal	Hex
0x00000000	1	0x00000001
0x00000004	2	0x00000002
0x00000008	4	0x00000004
0x0000000c	5	0x00000005
0x00000010	9	0x00000009