

1)

0000000000010	01011	001	01011	0010011		
0000000000010	01100	001	01100	0010011		
0000000	01011	01010	000	00101	0110011	
0000000000000	00101	010	01101	0000011		
0000000	01100	01010	000	00110	0110011	
0000000000000	00110	010	01110	0000011		
0000000	01101	00110	010	00000	0100011	
0000000	01110	00101	010	00000	0100011	
0000000	00000	00001	000	0000	0	1100111

A)

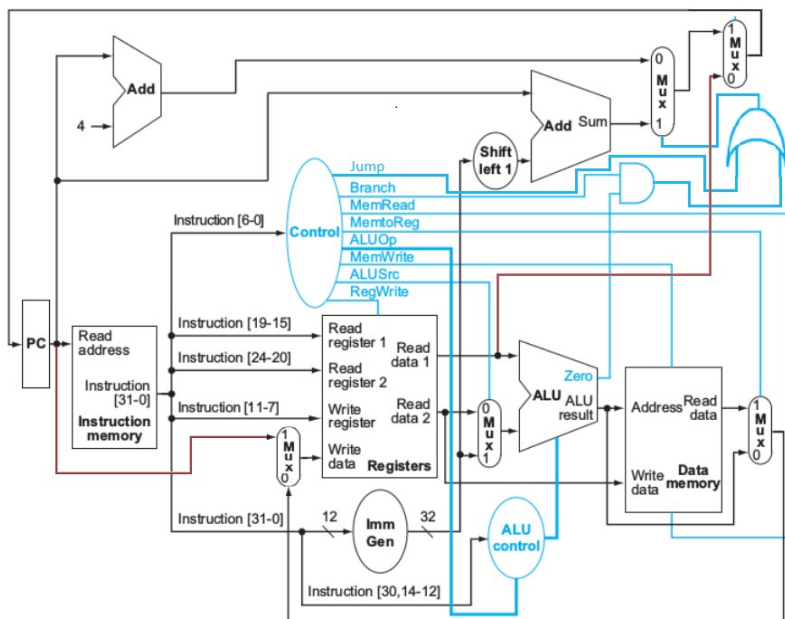
SLLI	RD,RS1,imm	SLLI	x11,x11,2
SLLI	RD,RS1,imm	SLLI	x12,x12,2
ADD	RD,RS1,RS2	ADD	x5,x10,x11
LW	RD,imm(RS1)	LW	x13,0(x5)
ADD	RD,RS1,RS2	ADD	x6,x10,x12
LW	RD,imm(RS1)	LW	x14,0(x6)
SW	RS2,imm(RS1)	SW	x13,0(x6)
SW	RS2,imm(RS1)	SW	x14,0(x5)
JALR	RD,RS1,imm	JALR	x0,x1,0

SLLI	a1,a1,2
SLLI	a2,a2,2
ADD	t0,a0,a1
LW	a3,0(t0)
ADD	t1,a0,a2
LW	a4,0(t1)
SW	a3,0(t1)
SW	a4,0(t0)
RET	

B) A função acima realiza a troca de valores entre dois índices de vetores, sendo que a0 é a posição da memória do vetor, a1 e a2 são os índices que terão os valores trocados, para realizar a troca, o programa grava em registradores temporários o conteúdo do endereço de memória de cada índice passado anteriormente, e posteriormente grava novamente na memória um valor no endereço do outro, por fim, retorna para o endereço da chamada de função.

2)

A)



B)

Foi adicionado uma porta OR para setar se é um desvio condicional(branch) ou se é um desvio incondicional(jump), também foi adicionado um novo multiplexador que controla se o dado vai vir do outro multiplexador, de onde pode ser um desvio condicional ou da próxima instrução(PC+4), ou de um registrador.

Também foi adicionado um multiplexador na entrada de escrita do banco de registradores, onde pode vir o dado do multiplexador da ULA ou direto do valor atual de PC.

c)

Instrução	ALUSrc	MEMtoREG	REGwrite	MEMread	MEMwrite	Branch
R-Type	0	0	1	0	0	0
LW	1	1	1	1	0	0
SW	1	X	0	0	1	0
BEQ	0	X	0	0	0	1
JALR	1	X	1	0	0	0