

**1- (6 pontos)** Implemente uma nova instrução LD RS, RT(RD);  $RS \leftarrow \text{Mem}[RD+RT]$  usando o datapath da single cycle. Você pode adicionar muxes e outros recursos, além de ter que modificar a unidade de controle com saídas extras. Quais serão os valores dos sinais de controle para executar sua instrução (complete no datapath) ?

Field	0	rs	rt	rd	shamt	funct
Bit positions	31:26	25:21	20:16	15:11	10:6	5:0

a. R-type instruction

Field	35 or 43	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

b. Load or store instruction

Field	4	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

c. Branch instruction

**FIGURE 4.14 The three instruction classes (R-type, load and store, and branch) use two different instruction formats.** The jump instructions use another format, which we will discuss shortly.

(a) Instruction format for R-format instructions, which all have an opcode of 0. These instructions have three 2 (6 pontos) Suponha que o registro Ri tem o valor i, a memória(i) tem o valor I (considerando que está organizada em palavras de 32 bits, como foi considerado em aula), por exemplo mem(1) → 0000 0000 0000 0000 0000 0000 0000 0001. Suponha o cálculo de endereço com o div 4 para pular de palavra em palavra como estabelecido no livro. Coloque os valores (0, 1 ou X senão for importante) nas linhas de controle e os valores nas linhas de dados no ciclo 6 na execução do código abaixo no Datapath do pipeline com unidade de Forwarding.

**ADD R1,R2,R2**

**ADD R2,R1,R1**

**SUB R3,R2,R1**

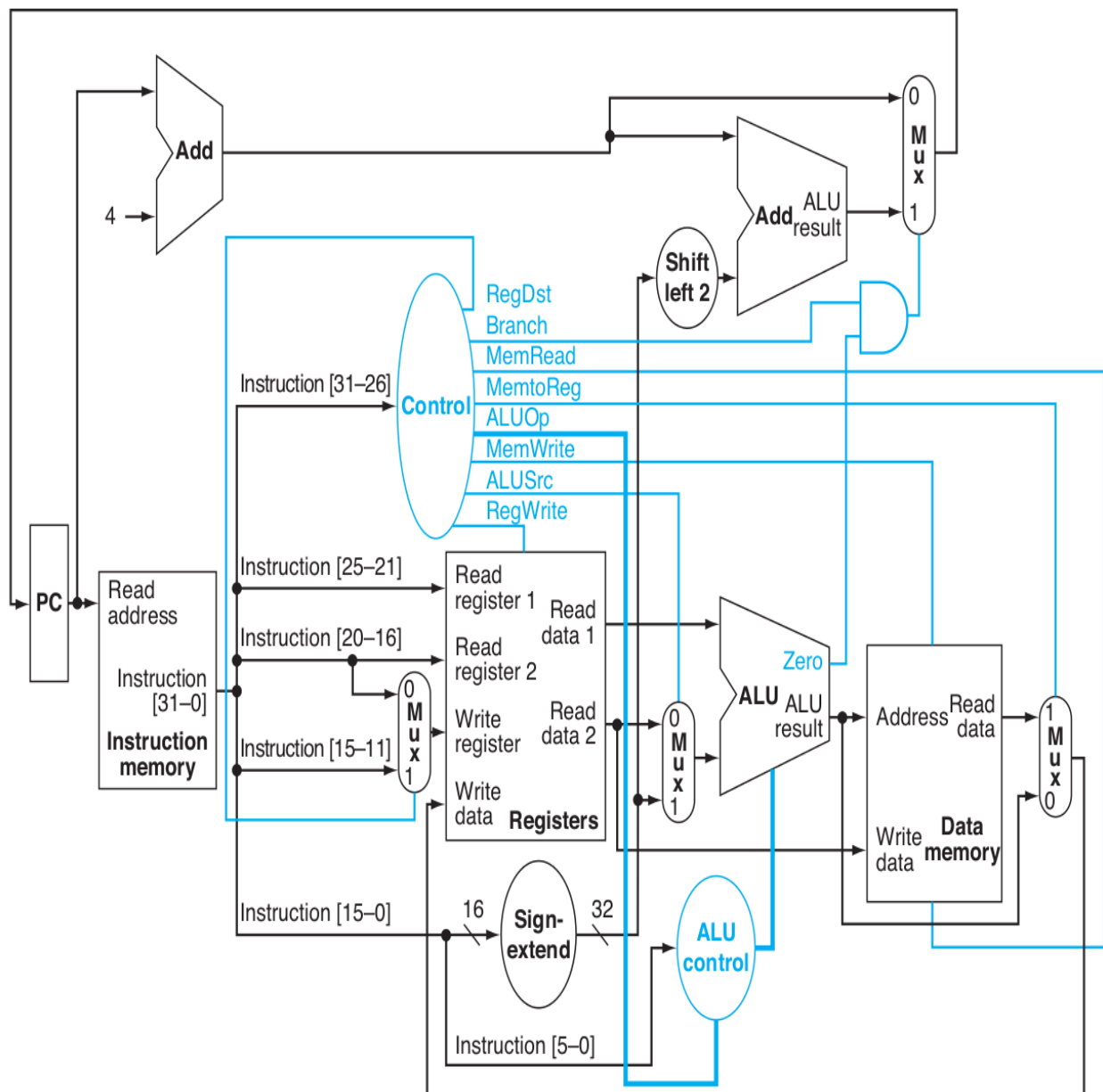
**BEQ R3,R4,-2**

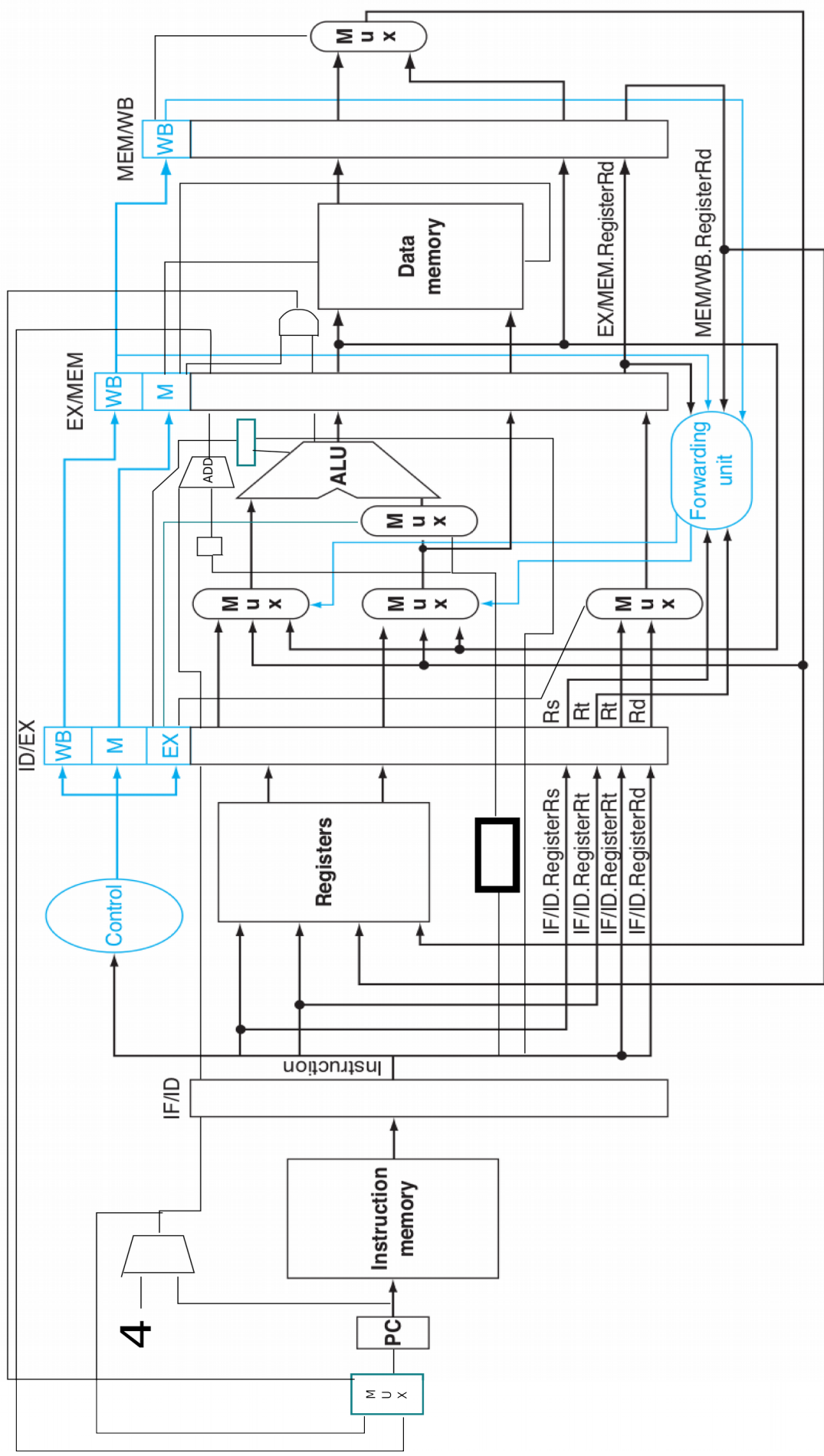
**ADD R5,R3,R3**

**ADD R6,R5,R2**

3. (10 pontos). Faça uma máquina de estados usando portas lógicas e flipflop que gera as seguintes sequencias: c=0, a sequencia é **0,1,7,9,0,7**, 0,1,7,9,0,7, ..., se c=1 a sequencia será **0,7,9,0,7,9**,...

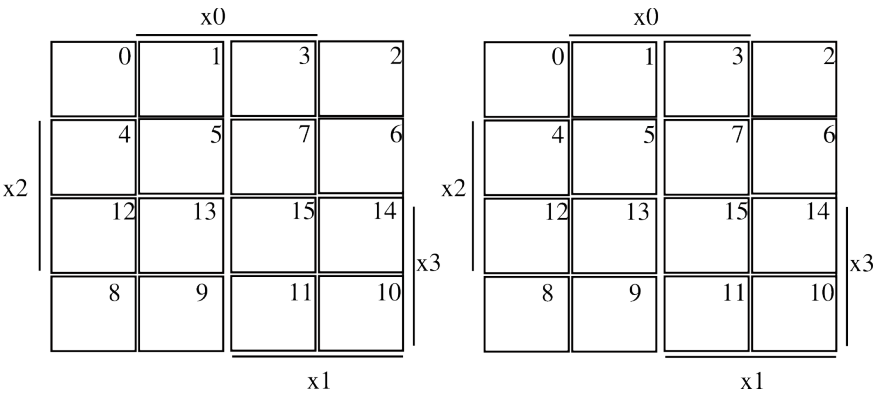
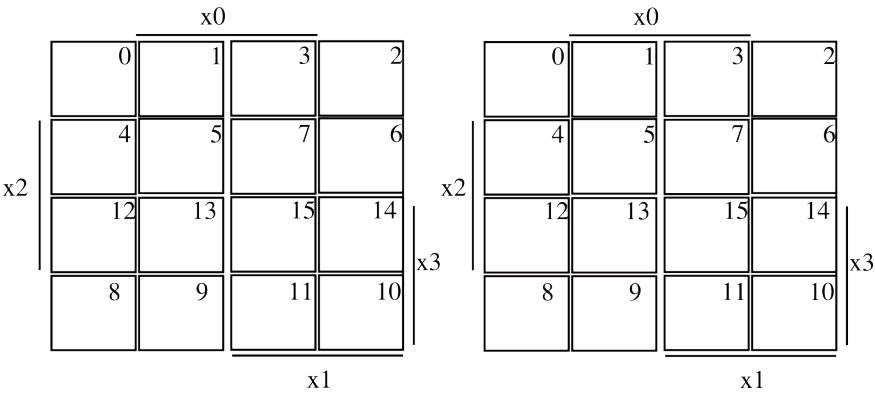
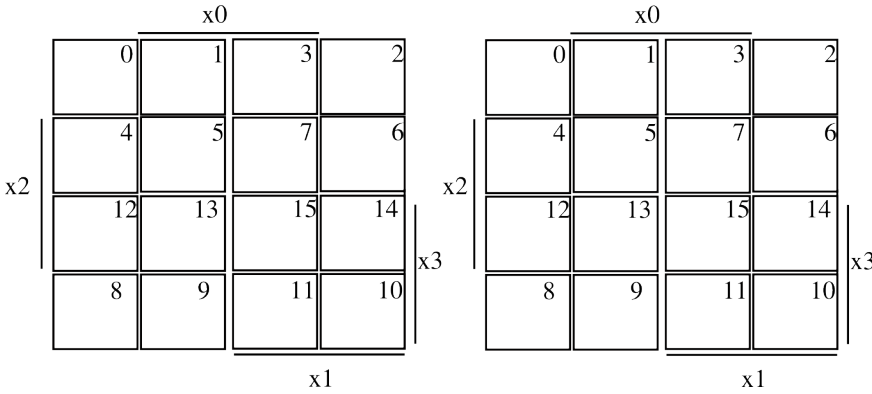
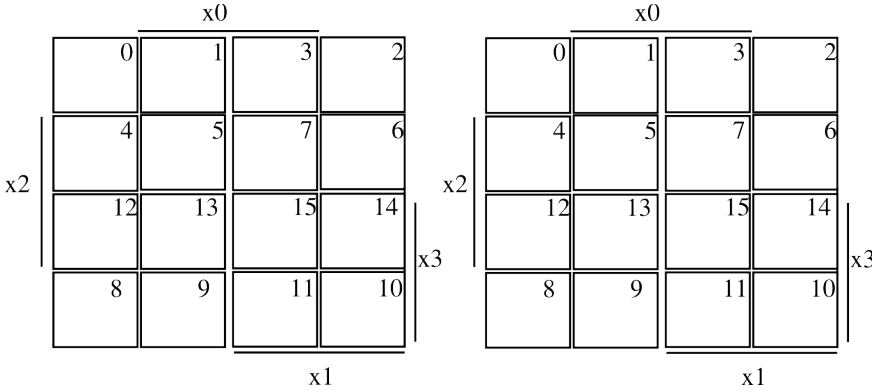
4. (10 pontos) Projetar a unidade de controle para o single cycle considerando os opcodes mais simples onde 0000xx para Load, 0001xx para Store, 0011xx para Beq, 1111xx Tipo R, 1110xx ADDI. A saída aluop 00 para soma, 01 para subtração e 10 para Func. Implemente a unidade com portas lógicas.





**FIGURE 4.56 The datapath modified to resolve hazards via forwarding.** Compared with the datapath in Figure 4.51, the additions are the multiplexers to the inputs to the ALU. This figure is a more stylized drawing, however, leaving out details from the full datapath, such as the branch hardware and the sign extension hardware.

1	X	X	X	X											
	3	2	1	0											
0	0	0	0	0											
1	0	0	0	1											
2	0	0	1	0											
3	0	0	1	1											
4	0	1	0	0											
5	0	1	0	1											
6	0	1	1	0											
7	0	1	1	1											
8	1	0	0	0											
9	1	0	0	1											
1	1	0	1	0											
0															
1	1	0	1	1											
1															
1	1	1	0	0											
2															
1	1	1	0	1											
3															
1	1	1	1	0											
4															
1	1	1	1	1											
5															



1	x3	x2	x1	x0														
0	0	0	0	0														
1	0	0	0	1														
2	0	0	1	0														
3	0	0	1	1														
4	0	1	0	0														
5	0	1	0	1														
6	0	1	1	0														
7	0	1	1	1														
8	1	0	0	0														
9	1	0	0	1														
10	1	0	1	0														
11	1	0	1	1														
12	1	1	0	0														
13	1	1	0	1														
14	1	1	1	0														
15	1	1	1	1														

