

## Arquitetura de Computadores

### Aula Prática #1 (3 de outubro de 2019)

Considere a Figura 1, que apresenta a microarquitetura do processador a 4 bits desenvolvido nas aulas teóricas, e que se pretende alterar para realizar duas novas instruções, conforme indicado na Tabela 1.

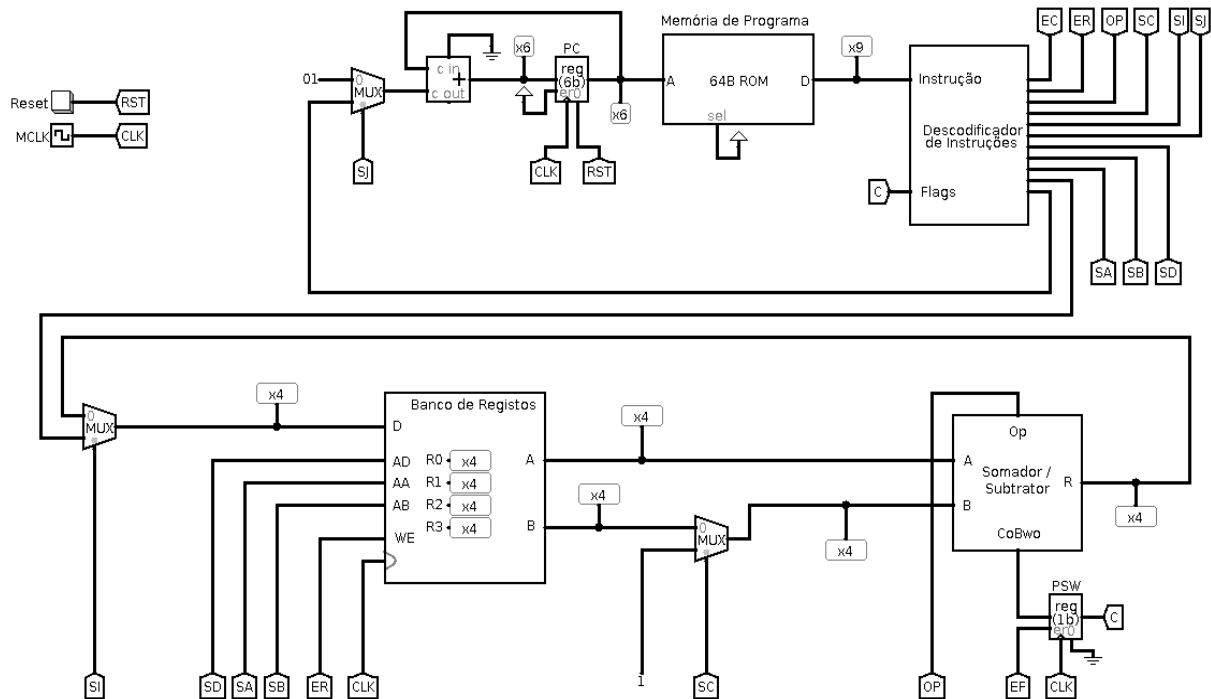


Figura 1 - Diagrama de blocos da microarquitetura do processador.

Instrução	Descrição	
add rd, ra, rb	Adiciona <b>rb</b> a <b>ra</b> e coloca o resultado em <b>rd</b> .	$rd \leftarrow ra + rb$
sub rd, ra, rb	Subtrai <b>rb</b> a <b>ra</b> e coloca o resultado em <b>rd</b> .	$rd \leftarrow ra - rb$
inc rd, ra	Adiciona 1 a <b>ra</b> e coloca o resultado em <b>rd</b> .	$rd \leftarrow ra + 1$
cmp ra, rb	Subtrai <b>rb</b> a <b>ra</b> e atualiza a <i>flag</i> C em conformidade com o resultado, que é descartado.	$ra - rb$
mov rd, #imm4	Carrega o valor da constante <b>const4</b> no registo <b>rd</b> .	$rd \leftarrow const4$
mov rd, ra	Copia para o registo <b>rd</b> o valor do registo <b>ra</b> .	$rd \leftarrow ra$
b offset6	Muda a execução para o endereço resultante da adição ao PC de <b>offset6</b> .	$PC \leftarrow PC + offset6$
bcc offset6	Quando a <i>flag</i> C apresenta o valor 0, muda a execução para o endereço resultante da adição ao PC de <b>offset6</b> .	$PC \leftarrow PC + offset6$ se $C == 0$
bcs offset6	Quando a <i>flag</i> C apresenta o valor 1, muda a execução para o endereço resultante da adição ao PC de <b>offset6</b> .	$PC \leftarrow PC + offset6$ se $C == 1$

Tabela 1 – Conjunto de instruções do processador.

## Arquitetura de Computadores

### Aula Prática #1

(3 de outubro de 2019)

1. Sabendo que as instruções originais foram recodificadas usando 10 bits no formato apresentado na Tabela 2, determine o valor lógico das saídas do subcircuito Descodificador de Instruções do processador para cada uma das instruções do novo conjunto de instruções. Explícite os casos de indiferença (*don't care*) e as saídas obtidas diretamente do código da instrução.

Instrução	Codificação			
	b9 b8	b7 b6	b5 b4	b3 ... b0
<b>add rd, ra, rb</b>	ra	rb	rd	0000
<b>inc rd, ra</b>	ra	--	rd	0010
<b>sub rd, ra, rb</b>	ra	rb	rd	0001
<b>cmp ra, rb</b>	ra	rb	--	0011
<b>mov rd, #imm4</b>	imm4		rd	0110
<b>mov rd, ra</b>	ra	--	rd	0111
<b>bcs offset6</b>	offset6			1000
<b>bcc offset6</b>	offset6			1001
<b>b offset6</b>	offset6			1010

Tabela 2 – Codificação das instruções.

2. Usando a aplicação Logisim, altere o subcircuito Descodificador de Instruções descrito no ficheiro *processador4b\_implementação\_base.circ*<sup>1</sup> para concretizar as definições do ponto anterior.
3. Usando o conjunto de instruções indicado na Tabela 1, implemente um programa que realiza a divisão de D por d usando o algoritmo das subtrações sucessivas. Considere que D e d são dois números naturais, codificados com 4 bits, que tomam dois valores à sua escolha guardados nos registos r0 e r1, respetivamente.
4. Complete a seguinte tabela, relativa à memória de código do processador, com a codificação em linguagem máquina do programa desenvolvido no ponto anterior.

Endereço	Instrução <i>assembly</i>	Instrução máquina									
1	mov r0, D										
2	mov r1, d										
3	...										

5. Na aplicação Logisim, carregue o programa na memória de código do processador e execute-o para verificar o seu funcionamento.
6. Pretende-se substituir a instrução **inc rd, ra** pela instrução **add rd, ra, #imm2**, que realiza a operação  $rd = ra + imm2$ , em que *imm2* representa um número natural codificado com dois bits. Altere a microarquitetura do processador para que este passe a suportar a execução desta nova instrução.

<sup>1</sup> O ficheiro está disponível para *download* na página da turma na plataforma Moodle.