

III

Considere o *datapath* e a unidade de controlo fornecidos na Figura anexa (ligeiramente alterado em relação à versão das aulas teórico-práticas), sabendo que corresponde a uma implementação multi-ciclo simplificada do MIPS.

1. Preencha a tabela fornecida em anexo com o nome de cada uma das fases de execução da instrução “*addi \$18, \$13, -1*” e com o valor que tomam, em cada uma delas os sinais de controlo aí indicados. Admita que o valor lógico “1” corresponde ao estado activo (considere que os registos no instante em que vai iniciar-se o *instruction fetch*, têm os seguintes valores $\$13=0x10010000$, $\$PC=0x00400054$).
2. Admita que os valores indicados no *datapath* fornecido correspondem à fotografia tirada no decurso de uma dada instrução. Identifique e justifique a sua resposta, observando todos os sinais:
 - A fase de execução em que se encontra;
 - Instrução em causa (apresente a instrução *assembly* completa);
 - o valor do registo $\$PC$ após a execução da instrução.

IV

Considere as tabelas apresentadas na Figura n.º1. Admita que o valor presente no registo $\$PC$ correspondente à 1ª instrução, que nesse instante o conteúdo dos registos é o indicado, e que vai iniciar-se o *instruction fetch* dessa instrução. Considere ainda o *datapath* e a unidade de controlo fornecido na Figura n.º1.

1. Face aos valores presentes no segmento de dados (tabela da esquerda) e ao trecho de código apresentado, determine o número total de ciclos de relógio que demora a execução completa desse trecho de código (desde o instante inicial do *instruction fetch* da primeira instrução presente em “L2:”), bem como os valores finais armazenados no segmento de dados. Justifique.
2. Represente, sob a forma de um diagrama temporal, a evolução do sinal de relógio e dos sinais de controlo “PCSource”, “PCWriteCond”, “Iord”, “IRWrite”, “MemRead”, e “RegWrite” durante a execução (em sequencia e pela ordem apresentada) das 3 primeiras instruções do trecho de código apresentado na Figura n.º1. Apresente a sua resposta como uma sequência contínua. (Nota: represente “don’t care” por //).

\$5	0x00008421
\$6	0x10010070
\$7	0x1001009C
\$8	0x00001E00
\$9	0x10400EA0
\$10	0x00000003
	...
\$PC	0x00400048
	CPU

L1:	lw	\$7,0(\$6)
	and	\$7,\$7,\$5
	beq	\$7,\$0,L2
	sw	\$7,4(\$6)
	addi	\$6,\$6,8
	J	L1
L2:	...	

Endereço	Dados
0x10010070	0xFFFF0003
0x10010074	0X021B581A
0x10010078	0X17AC1248
0x1001007C	0X1854E790
0x10010080	0X48310060
0x10010084	0X5FF38C29

Opcode	Funcnt	Operação
0	0x20	add
0	0x22	sub
0	0x24	and
0	0x25	or
0x02		j
0x04		beq
0x05		bne
0x08		addi
0x0C		andi
0x23		lw
0x28		sw

Figura n.º1 – Tabelas e dados fornecidos.