

1 – Na sequência de código, abaixo, considere que o valor inicial do registo **%ebx** é **10**:

<pre> .pos 0x00000F00 I1: irmovl \$8, %ecx I2: mrmovl \$22(%ebx), %eax I3: subl %eax, %ecx I4: irmovl \$100, %esi I5: jg I8 I6: andl %esi, %ecx I7: jmp I9 I8: xorl %esi, %ecx I9: halt </pre>	<pre> .pos 0x00000020 .byte 0x0A .byte 0x00 .byte 0x00 .byte 0x00 </pre>
---	--

1.1 – Preencha a tabela 1, indicando a ocupação dos estágios da versão **PIPE**- do Y86.

<i>Justifique brevemente as suas opções</i>

2 – Preencha a tabela 2, indicando a ocupação dos estágios da versão **PIPE** do Y86.

<i>Justifique brevemente as suas opções referindo a Instrução e ciclo</i>

2.1 – Para as condições da alínea anterior indique, justificando, qual o valor do PC no ciclo 7.

--

2.2 – Reordene as instruções de forma a minimizar o número de ciclos necessários para executar este programa (versão **PIPE**), sem alterar a sua funcionalidade. **Justifique as suas opções!**

<i>Preencha e justifique usando o formato, I_i – I_(antigo) .</i>
I ₁ --- I ; I ₂ --- I ; I ₄ --- I ; I ₃ --- I ; I ₄ --- I ; I ₅ --- I ; I ₆ --- I ; I ₇ --- I ; I ₈ --- I ; I ₉ --- I

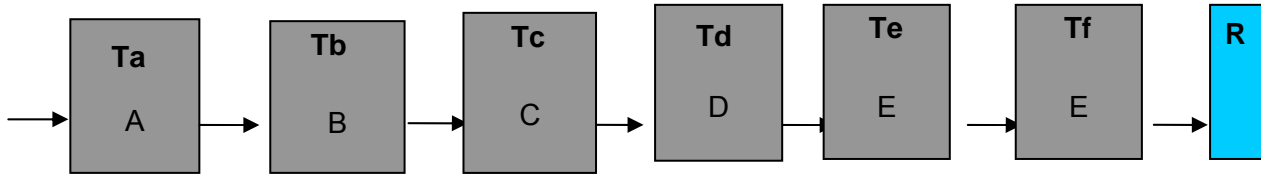
3. – Considere a seguinte sequência de código:

1	0x000: 308209000000		irmovl	\$9,	%edx
2	0x006: 308315000000		irmovl	\$21,	%ebx
3	0x00c: 6123		subl	%edx,	%ebx
4	0x00e: 308480000000		irmovl	\$128,	%esp
5	0x014: 404364000000		rmmovl	%esp,	100(%ebx)
6	0x01a: a028		pushl	%edx	
7	0x01c: b008		popl	%eax	
8	0x01e: 7328000000		je	fim	
9	0x023: 8029000000		call	proc	
10	0x028:		fim:		
11	0x028: 10		halt		
12	0x029:		proc:		
13	0x029: 90		ret		

3.1 – Para a instrução assinalada como específica, preencha a tabela com os valores relevantes dos sinais de controlo, na versão **sequencial** do processador **Y86**

Estágios	Genéricos	Específica
	Instrução	call proc
Extrai	icode:ifun \leftarrow M ₁ [PC] rA:rB \leftarrow M... ValC \leftarrow M.. valP \leftarrow PC...	
Descodifica	ValA \leftarrow R[rA] ValB \leftarrow ...	
Executa	valE \leftarrow ... ajusta CC Bch \leftarrow Cond(CC, ifun)	
Memória	M _? \leftarrow ValM \leftarrow	
Actualiza	R[?] \leftarrow ...	
PC	PC \leftarrow valP PC \leftarrow Cch: valC: ValP ValM	

4. – A lógica combinacional de um processador com uma organização encadeada pode ser decomposta nos blocos **Ta..Tf - R**, visíveis na figura, com os tempos de processamento abaixo:



Ta=80, Tb=50, Tc=30, Td=70, Te=60 ps, Tf= 10 ps; R= 15 ps -- Justifique as respostas!

4.1 Se considerarmos 4 estágios: **a)** como devem ser agrupados os blocos para maximizar o débito?
b) Qual o débito e tempo de execução de cada instrução?

a)

b)

4.2 Qual **a)** o máximo débito que pode ser obtido? e **b)** a quantos estágios corresponde?

a)

b)

[illegible][illegible]