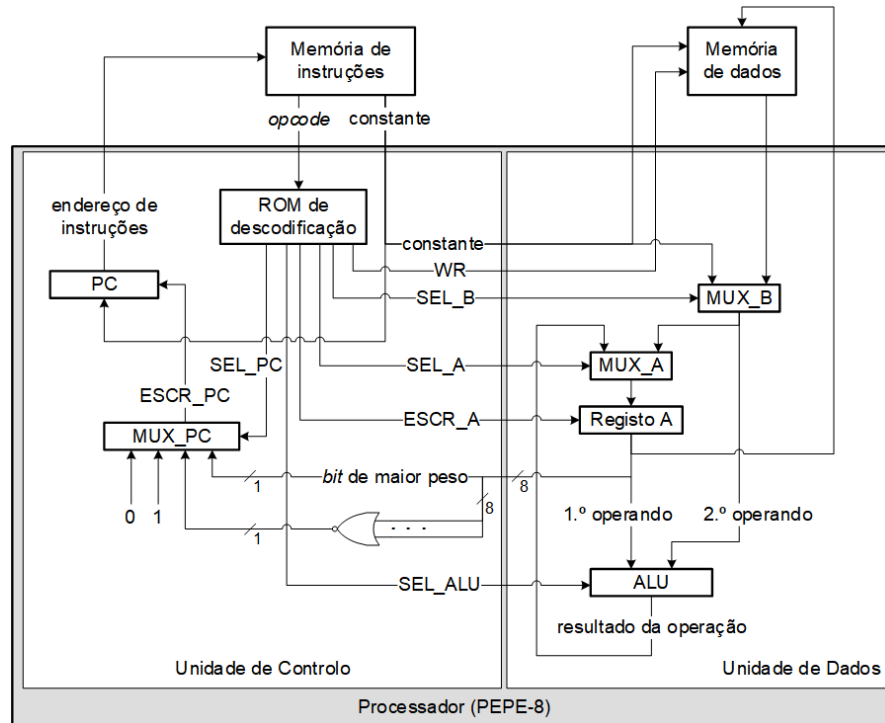


NOME	NÚMERO
------	--------

1. (2 valores) A figura seguinte representa o diagrama de blocos básico do PEPE-8, processador de 8 bits, bem como as memórias a que está ligado.



Na tabela seguinte estão referidos os sinais usados para comandar quer a Unidade de Dados, quer a Unidade de Controlo. Preencha esta tabela, especificando para cada sinal qual a indicação concreta que fornece no caso de o PEPE-8 estar a executar a instrução SUB [5EH] ($A \leftarrow A - M[5EH]$). Para cada sinal, use a indicação que for mais conveniente, como por exemplo:

- Ativo / Não ativo;
- Um valor numérico;
- Uma indicação simples que especifique a opção a selecionar (ex: esquerda / direita);
- Um simples traço horizontal, ou uma cruz (se não interessar para esta instrução).

Constante	WR	SEL_A	SEL_B	ESCR_A	SEL_ALU	SEL_PC
5EH	Não ativo	Esquerda	Direita	Ativo	SUB	0

SUB [5EH] ($A \leftarrow A - M[5EH]$)

2. (1 valor) Indique quantos bits precisa, no mínimo, para contar desde zero até 200 Mi.

28

 bits

3. (1 + 2 + 1 + 3 valores) Num processador com 32 bits de dados, um programa somou as constantes FFFF8AC7H e 00000F3BH (em notação de complemento para 2).

- a) Escreva um programa em *assembly* do PEPE-16 que obtenha o mesmo valor matemático como resultado no R3, colocando o primeiro valor no registo R2 e o segundo no registo R3.

MOV R2, 8AC7H	; primeiro valor, em hexadecimal
MOV R3, 0F3BH	; segundo valor, em hexadecimal
ADD R3, R2	; efetua a operação

- b) Indique os dois valores e o resultado em binário, tal como o PEPE-16 os processa.

1	0	0	0	1	0	1	0	1	1	0	0	0	1	1	1	primeiro valor
0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	segundo valor
1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	resultado

- c) Indique qual o valor, em hexadecimal, que o processador de 32 bits obteve, após a soma.

FFFF9A02	H	resultado, em hexadecimal
-----------------	---	---------------------------

- d) Obtenha o simétrico do segundo valor, em decimal e em hexadecimal (16 bits, em notação de complemento para 2).

-3899	simétrico, em decimal
F0C5	H simétrico, em hexadecimal

4. (3 valores) Considere o seguinte programa em linguagem *assembly* do PEPE-16 (SHL = desloca N bits à esquerda). Responda às seguintes questões:

	MOV	R1, 0DCCDH
	MOV	R2, 7233H
	MOV	R3, 0
ciclo:	AND	R1, R2
	JZ	fim
	ADD	R3, 1
	SHL	R2, 1
	JMP	ciclo
fim:	JMP	fim

Quantas vezes o programa passa por “ciclo”?

4

Qual o valor final de R2?

9198 H

Qual o valor final de R3?

3 H

5. (1 + 2 + 4 valores) Considere o seguinte programa em linguagem *assembly* do PEPE-16.

Endereços				
	PLACE	1000H		
	AA	EQU	7EH	
	BB	EQU	3BFH	
	CC	EQU	0EC4DH	
1000H	pilha:	TABLE	100H	
1200H	fim_pilha:			
1200H	Z:	WORD	CC	
	PLACE	0H		
0000H		MOV	SP, fim_pilha	
0002H		MOV	R0, AA	
0004H		MOV	R1, BB	
0006H		MOV	R2, Z	
0008H		CALL	X	
000AH		MOV	R1, [R2]	
000CH	fim:	JMP	fim	
000EH	X:	PUSH	R0	
0010H		PUSH	R1	
0012H		MOV	R0, CC	
0014H		MOV	R1, 5	
0016H		CALL	Y	
0018H		MOV	[R2], R0	
001AH		POP	R1	
001CH		POP	R0	
001EH		RET		
0020H	Y:	PUSH	R1	
0022H	ciclo:	SHL	R0, 1	; deslocamento à esquerda
0024H		SUB	R1, 1	
0026H		JNZ	ciclo	
0028H		POP	R1	
002AH		RET		

CALL Etiqueta	SP ← SP-2 M[SP] ← PC PC ← Etiqueta
RET	PC ← M[SP] SP ← SP+2

- Preencha os endereços que faltam (lado esquerdo, preencha apenas as linhas em que tal faça sentido). Considera-se que cada MOV com uma constante ocupa apenas uma palavra.
- Preencha as instruções que faltam (ou partes delas), tendo em atenção os comentários e funcionamento do programa.
- Acabe de preencher a tabela da página seguinte com informação sobre os acessos à memória feitos pelo programa, de leitura (L) ou escrita (E). Use apenas as linhas que necessitar.

Endereço em que está a instrução que faz o acesso	Endereço acedido	L ou E	Valor lido ou escrito
0008H	11FEH	E	000AH
000EH	11FCH	E	007EH
0010H	11FAH	E	03BFH
0016H	11F8H	E	0018H
0020H	11F6H	E	5
0028H	11F6H	L	5
002AH	11F8H	L	0018H
0018H	1200H	E	89A0H
001AH	11FAH	L	03BFH
001CH	11FCH	L	007EH
001EH	11FEH	L	000AH
000AH	1200H	L	89A0H