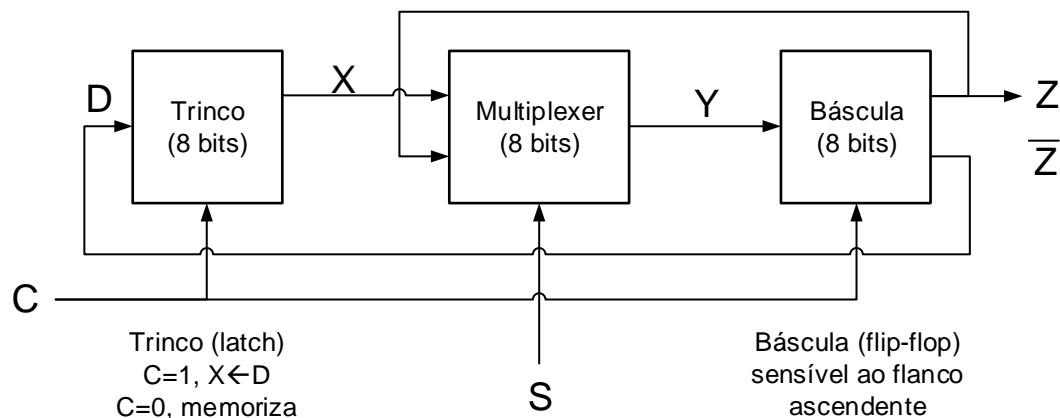


NOME		NÚMERO	
------	--	--------	--

1. (3 valores) Considere o seguinte circuito, com barramentos de 8 bits. C é o *clock* (tanto do trinco como da básica) e S é o sinal de seleção do *multiplexer* ($S=0$ seleciona a entrada X). Assumindo que os sinais C e S evoluem ao longo do tempo da forma indicada na tabela seguinte, preencha os valores estáveis no resto da tabela (escreva todas as células, mesmo que o valor se mantenha). Todos os valores de 8 bits estão representados em hexadecimal (não é preciso colocar o H).



C	1	0	1	0	0	1	1	0	1	1
S	1	0	0	1	0	0	1	1	1	0
X										
Y	31									
Z										
\bar{Z}										

2. (2 + 1 + 3 valores) Considere o seguinte programa no PEPE-16:

```
MOV R1, 2473      ; constante em decimal
MOV R2, 0F47DH    ; constante em hexadecimal
ADD R1, R2
```

- a) Indique cada um dos 16 bits do R1, após o primeiro MOV.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

R1, em binário

- b) Indique cada um dos 16 bits do R2, após o segundo MOV.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

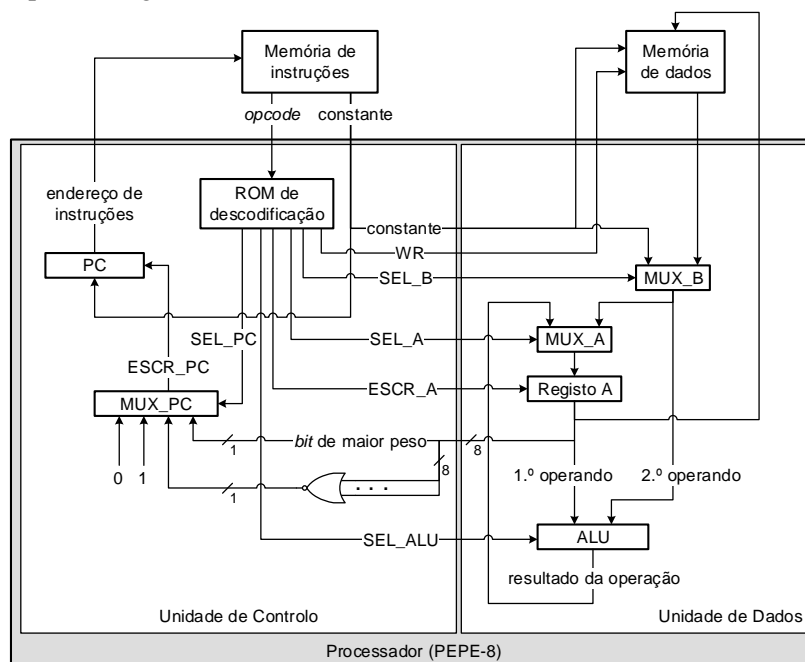
R2, em binário

- c) Indique cada um dos 16 bits do R1 e o respetivo valor em decimal, após o ADD.

R1, em binário

R1, em decimal

3. (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.



Suponha que o PEPE-8 suporta instruções de deslocamento, em que a ALU tem duas operações, DeslocaDta e DeslocaEsq, sendo o 1.º operando o valor a deslocar e o 2.º operando o número de bits a deslocar. 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 este dá no caso de o PEPE-8 estar a executar a instrução de deslocamento à direita SHR 3 ($A \leftarrow A \gg 3$), assumindo que esta instrução existia. Para cada sinal, use a indicação que for mais conveniente:

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

Constante	WR	SEL_A	SEL_B	ESCR_A	SEL_ALU	SEL_PC

4. (1 + 1 valores) Considere que o PEPE-16 (processador de 16 bits, endereçamento de byte) está ligado a uma RAM com capacidade de 2 KBytes, localizada a partir do endereço 0000H.

a) A quantos endereços diferentes é possível o PEPE-16 aceder (valor em decimal)?

b) Qual é o maior endereço (em hexadecimal) da RAM a que o PEPE consegue aceder com MOVB? H

5. (3 valores) Complete o programa do lado direito, preenchendo os retângulos com os valores corretos.

Registo – registo onde a instrução dessa linha armazena o resultado
Valor – valor desse registo após a execução da instrução
 SHL – Deslocamento à esquerda

Instrução	Registo e valor
MOV <input type="text"/> 34EDH	R7 <input type="text"/>
<input type="text"/> R7, 5	<input type="text"/> 34F2H
SHL <input type="text"/> 3	R7 <input type="text"/>
<input type="text"/>	R9 522AH
OR R7, R9	<input type="text"/>

6. (1 + 3 valores) Considere o seguinte programa em linguagem *assembly* do PEPE-16.

Endereços			
	PLACE	5000H	
	FUNDO	EQU 8000H	; valor em complemento para 2
	N	EQU 5	; número de elementos da lista
	lista:	WORD 11	
		WORD 9	
		WORD 13	
		WORD 3	
		WORD 8	
	variavel:	WORD FUNDO	
	PLACE	0000H	
		MOV R1, lista	
		MOV R2, N	
		MOV R3, variavel	
	ciclo:	MOV R4, [R1]	
		MOV R5, [R3]	
		CMP R4, R5	
		JLE prox	; jump if less than or equal
		MOV [R3], R4	; atualiza a “variavel”
	prox:	ADD R1, 2	
		SUB R2, 1	
		JNZ ciclo	
	fim:	JMP fim	

- a) 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.
- b) Acabe de preencher a tabela com informação sobre os acessos de dados 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