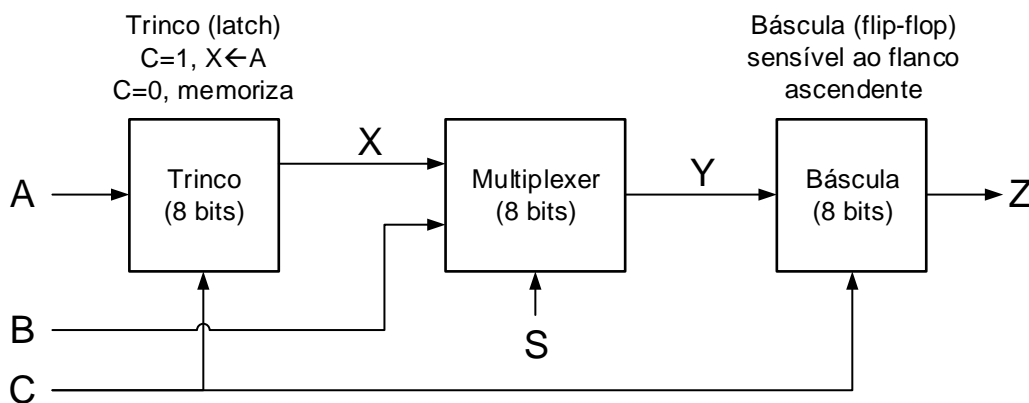


NOME		NÚMERO	
------	--	--------	--

1. (3 valores) Considere o seguinte circuito, em que os sinais A, B, X, Y e Z são 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 A, B, C e S evoluem ao longo do tempo da forma indicada na tabela seguinte, acabe de preencher o resto da tabela (escreva todas as células, mesmo que o valor se mantenha).



A	21H	4AH	73H	9CH	A4H	58H
B	E3H	29H	67H	C8H	85H	B1H
C	1	0	0	1	1	0
S	0	0	0	1	1	0
X	21H	21H	21H	4AH	73H	9CH
Y	21H	21H	21H	4AH	67H	C8H
Z	25H	25H	25H	21H	21H	C8H

2. (2 valores) Considere o número decimal -2574. Represente-o em notação de complemento para 2, em hexadecimal com 16 e 32 bits.

F	5	F	2	H
F	F	F	F	F

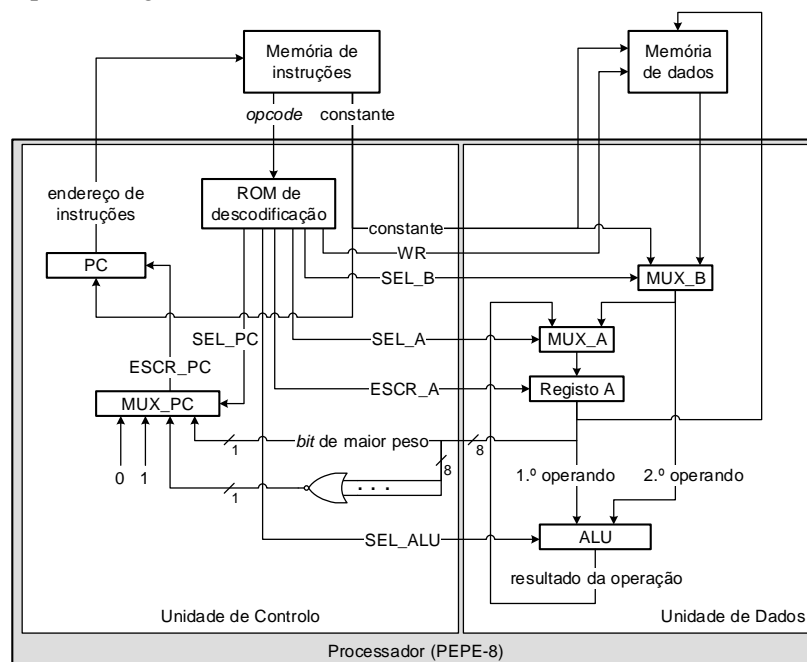
3. (2 valores) Converta para decimal o número A75H, considerando que este número está representado em notação de complemento para 2 com 12 bits.

-1419

4. (2 valores) Indique, em binário e em decimal, qual o maior número que consegue representar em notação de complemento para 2, com 10 bits.

0	1	1	1	1	1	1	1	1	1	binário
									+511	decimal

5. (2+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.



- a) Justifique a necessidade/utilidade da existência do *multiplexer A*.

Permite que no registo A seja escrito quer o resultado de uma operação na ALU, quer um valor vindo ou da memória de dados ou da própria instrução.

- b) 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 ADD [40H] (ou, em notação RTL, $A \leftarrow A + M[40H]$).

Sinal	Valor numérico, ou Qual indicação seleccionada, ou Não interessa (sinal inativo)
Constante	40H
WR	Inativo (lê da memória)
SEL_B	Entrada da direita
SEL_A	Entrada da esquerda
ESCR_A	Ativo (escreve no registo A)
SEL_ALU	Soma
SEL_PC	0 (não salta)

6. (3+4 valores) Considere o seguinte programa em linguagem *assembly* do PEPE-16. Para facilitar, fornece-se a descrição interna das instruções CALL e RET.

CALL Etiqueta	$SP \leftarrow SP-2$ $M[SP] \leftarrow PC$ $PC \leftarrow \text{Endereço da Etiqueta}$
RET	$PC \leftarrow M[SP]$ $SP \leftarrow SP+2$

Endereços			
	PLACE	2000H	; início da zona de dados
	INDICE	EQU 05H	
2000H	tabela:	TABLE 20H	
2040H	pos_bit:	WORD 0	; variável para guardar o resultado
	PLACE	0H	; início da zona de código
0000H		MOV SP, 1000H	
...	; <u>inicializa todos os elementos da tabela a 78H</u>
0010H		MOV R2, INDICE	
0012H		CALL elemento	; trata o elemento n.º INDICE
0014H		MOV R1, pos_bit	
0016H		MOV [R1], R0	; guarda resultado na variável
0018H	fim:	JMP fim	
	; Obtém em R0 a posição do bit a 1 com maior peso no elemento da tabela		
	; com índice dado por R2 (exemplo: se o elemento for 37H, R0 ficará com 5)		
001AH	elemento:	PUSH R1	
001CH		PUSH R2	
001EH		PUSH R3	
0020H		MOV R3, tabela	; base da tabela
0022H		SHL R2, 1	
0024H		MOV R1, [R3+R2]	; obtém elemento da tabela
0026H		CALL posicao	; obtém posição do bit de maior peso a 1
0028H		POP R3	
002AH		POP R2	
002CH		POP R1	
002EH		RET	
	; Obtém em R0 a posição do bit a 1 com maior peso no valor de R1		
0030H	posicao:	PUSH R1	
0032H		MOV R0, -1	; inicializa a -1 porque soma logo a seguir
0034H	maisUm:	ADD R0, 1	; conta mais uma posição
0036H		SHR R1, 1	; desloca R1 à direita
0038H		JNZ maisUm	
003AH		POP R1	
003CH		RET	

- a) Preencha os endereços que faltam (lado esquerdo, preencha apenas as linhas em que tal faça sentido) e os espaços no programa.

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.

Considere que todos os registos estão a 0000H antes de o programa começar.

Note que no início o programa inicializa todos os elementos da tabela a 78H.

Endereço da instrução que faz o acesso	Endereço acedido	L ou E	Valor lido ou escrito
0012H	0FFEH	E	0014H
001AH	0FFCH	E	0000H
001CH	0FFAH	E	0005H
001EH	0FF8H	E	0000H
0024H	200AH	L	0078H
0026H	0FF6H	E	0028H
0030H	0FF4H	E	0078H
003AH	0FF4H	L	0078H
003CH	0FF6H	L	0028H
0028H	0FF8H	L	0000H
002AH	0FFAH	L	0005H
002CH	0FFCH	L	0000H
002EH	0FFEH	L	0014H
0016H	2040H	E	0006H