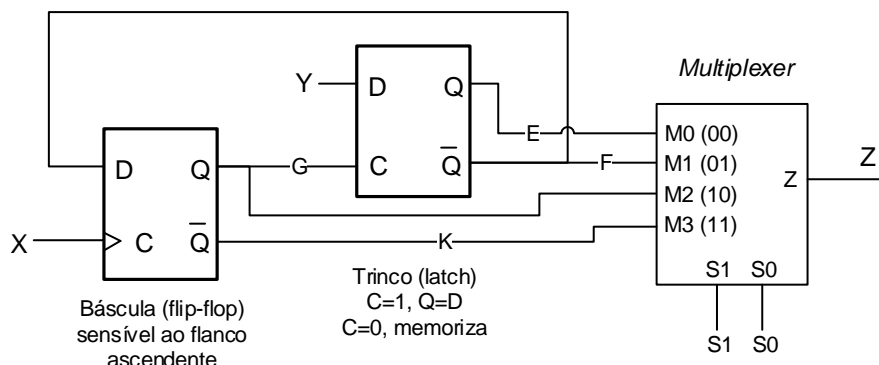


NOME		NÚMERO	
------	--	--------	--

1. (2 valores) Considere o seguinte circuito. Assumindo que os sinais X, Y, S0 e S1 evoluem ao longo do tempo da forma indicada na tabela seguinte, acabe de preencher o resto da tabela.



X	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	
Y	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
S0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1
S1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
G	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

2. (2+2+2 valores) Considere o seguinte programa, a executar no PEPE (processador de 16 bits, endereçamento de byte). Todas as constantes estão em decimal.

```
MOV R1, -2301
MOV R2, 381
SHR R2, 2 ; shift right (deslocamento para a direita)
OR R1, R2
```

- a) Indique o valor de R1 (em hexadecimal com 16 bits, usando a notação de complemento para 2) após a execução da primeira instrução.

F 7 0 3 H

- b) Indique os valores (em binário com 16 bits, usando a notação de complemento para 2) com que R1 e R2 são inicializados, bem como os valores finais destes registos, após a execução destas instruções.

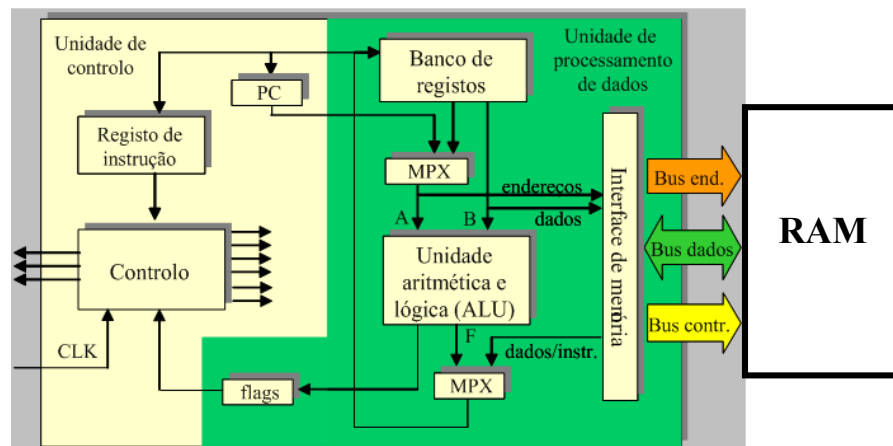
1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	1	R1 (após os MOVs)
0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	R2 (após os MOVs)
1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	R1 final
0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	R2 final

- c) Indique (em decimal) o mínimo e o máximo valor numérico que é possível representar com 9 bits, em notação de complemento para 2.

Mínimo: **- 256**

Máximo: **+ 255**

3. (2+2+3 valores) A figura seguinte representa o diagrama de blocos básico do PEPE (processador de 16 bits, endereçamento de byte), a uma memória RAM onde estão armazenados tanto os dados como as instruções dos programas.



- a) Assuma que a memória tem 13 bits de endereço e que está acessível a partir do endereço 0000H. Qual a capacidade em bytes da RAM?

8192 bytes (em decimal)

- b) Qual é o endereço (em hexadecimal) imediatamente a seguir à última célula da RAM?

2000 H

- c) Considere o programa seguinte e preencha a seguinte tabela (use apenas as linhas necessárias), indicando **todos os acessos à memória** (em leitura ou escrita) efetuados pelo processador durante a execução do programa. Assuma que os MOVs ocupam apenas uma palavra.

PLACE 0

I1: MOV R1, D1
 I2: MOV R2, [R1]
 I3: ADD R2, R1
 I4: MOV [R2], R1
 I5: MOV R2, 7AH
 D1: WORD 7C40H

Registo que indica o endereço da memória	Valor deste registo	Etiqueta da instrução em que ocorre	Leitura ou escrita	Valor lido ou escrito (só nos acessos de dados)
PC	0000H	I1	L	
PC	0002H	I2	L	
R1	000AH	I2	L	7C40H
PC	0004H	I3	L	
PC	0006H	I4	L	
R2	7C4AH	I4	E	000AH
PC	0008H	I5	L	

4. (2+3 valores) Considere o seguinte programa em linguagem *assembly* do PEPE. Para facilitar, forneça-se a descrição interna das instruções CALL e RET.

PLACE 0H		
00H	MOV	SP, 2000H
02H	MOV	R1, 7EH
04H	MOV	R2, 30FH
06H	MOV	R3, C
08H	CALL	B
0AH	fim:	JMP fim
0CH	A:	PUSH R2
0EH	ciclo:	SHR R1, 1
10H		SUB R2, 1
12H		JNZ ciclo
14H	POP	R2
16H	RET	
18H	B:	PUSH R1
1AH		PUSH R2
1CH		MOV R1, 9B73H
1EH		MOV R2, 5
20H		CALL A
22H		MOV [R3], R1
24H		POP R2
26H		POP R1
28H		RET
2AH	C:	WORD 5A2CH

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

- a) Preencha os endereços de cada instrução (lado esquerdo) e os espaços no programa. Considere que os MOVs ocupam apenas uma palavra.
- b) Acabe de preencher a tabela com informação sobre os acessos à memória (só contam acessos de dados, buscas de instruções ignoram-se) feitos pelo programa, de leitura (L) ou escrita (E).

Endereço da instrução que faz o acesso	Endereço acedido	L ou E	Valor lido ou escrito
08H	1FFE H	E	000AH
18H	1FFCH	E	007EH
1AH	1FFAH	E	030FH
20H	1FF8H	E	0022H
0CH	1FF6H	E	0005H
14H	1FF6H	L	0005H
16H	1FF8H	L	0022H
22H	002AH	E	04DBH
24H	1FFAH	L	030FH
26H	1FFCH	L	007EH
28H	1FFE H	L	000AH