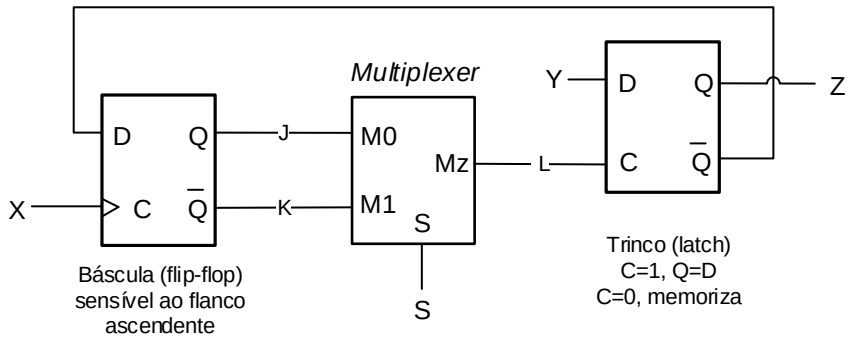
1.º Teste de Introdução à Arquitetura de Computadores — IST – LEIC-Taguspark

1.º Semestre 2013/2014 — Duração: 60 minutos — 31 outubro 2013

NOME		NÚMERO	

1. (2 valores) Considere o seguinte circuito. Assumindo que os sinais X, Y e S evoluem ao longo do tempo da forma indicada na tabela seguinte, acabe de preencher o resto da tabela (o sombreado é apenas para melhor visualização).



X	1	0	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
Y	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	1	1
S	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
J	1	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
K	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
L	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
Z	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**

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, -3728
MOV   R2, 231
ADD   R1, R2
SHL   R2, 2        ; shift left (deslocamento para a esquerda)
```

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	**1**	**7**	**0**	H

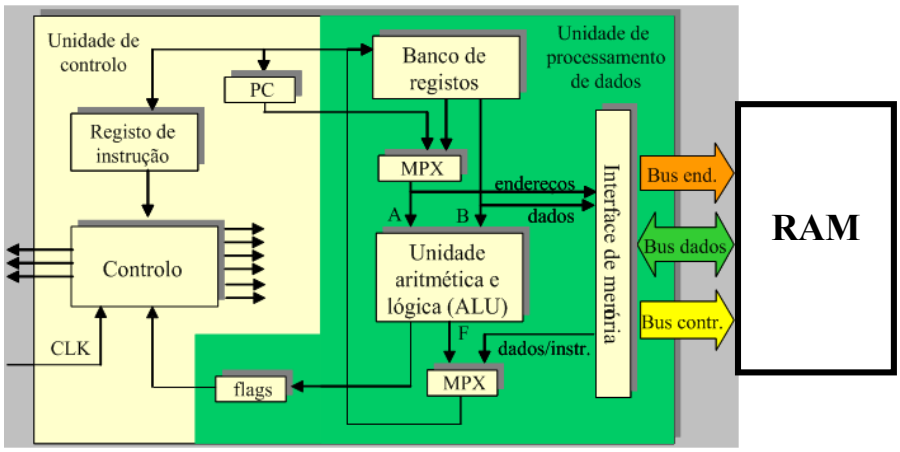
b) Indique os valores (em binário com 16 bits, usando a notação de complemento para 2) com que R1 e R2 são inicializados, bem como os valores finais destes registos, após a execução destas instruções.

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

c) Se em vez de 16 bits quisesse representar o valor inicial de R1 com menos bits, mas ainda em notação de complemento para 2 e com o mesmo valor numérico, qual o mínimo número de bits necessário?

13	bits

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 11 bits de endereço e que está acessível a partir do endereço 0000H. Qual a capacidade em bytes da RAM (valor em decimal)?

2048 bytes

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

07FF H

c) Assuma que o programa seguinte se situa a partir do endereço 0000H e que o PC já foi previamente inicializado com esse valor. 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.

```
I1:    MOV   R2, 6AH
I2:    MOV   R1, X
I3:    MOV   R2, [R1]
I4:    ADD   R2, R1
I5:    MOV   [R2], R1
PLACE 1000H
X:     WORD  1234H
```

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**	
PC	**0004H**	**I3**	**L**	
R1	**1000H**	**I3**	**L**	**1234H**
PC	**0006H**	**I4**	**L**	
PC	**0008H**	**I5**	**L**	
R2	**2234H**	**I5**	**E**	**1000H**

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

```
        PLACE   0H
00H             MOV     SP, 2000H
02H             MOV     R1, 78H
04H             MOV     R2, 14CH
06H             MOV     R3, 8FAH
08H             CALL    Y
0AH     fim:    JMP     fim
0CH     X:      PUSH    R2
0EH     ciclo:  SHR     R1, 1       ; deslocamento à direita
10H             SUB     R2, 1
12H             JNZ     ciclo
14H             POP     R2
16H             RET
18H     Y:      PUSH    R1
1AH             PUSH    R2
1CH             MOV     R1, 7FA3H
1EH             MOV     R2, 5
20H             CALL    X
22H             MOV     [R3], R1
24H             POP     R2
26H             POP     R1
28H             RET
```

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 todos 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	1FFEH	E	000AH
18H	1FFCH	E	0078H
1AH	1FFAH	E	014CH
20H	1FF8H	E	0022H
0CH	1FF6H	E	0005H
14H	1FF6H	L	0005H
16H	1FF8H	L	0022H
22H	08FAH	E	03FDH
24H	1FFAH	L	014CH
26H	1FFCH	L	0078H
28H	1FFEH	L	000AH