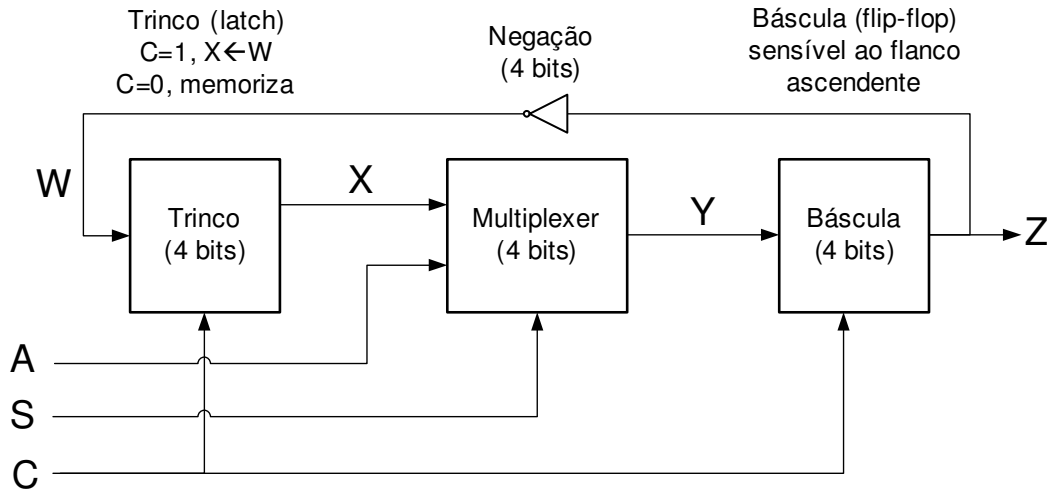
Repescagem - 1.º Teste de Introdução à Arquitetura de Computadores IST – LEIC-T

1.º Semestre 2016/2017 Duração: 60 minutos 4 fevereiro 2017

NOME		NÚMERO	

1. (3 valores) Considere o seguinte circuito, em que os sinais A, W, X, Y e Z são barramentos de 4 bits, C é o *clock* (tanto do trinco como da báscula) e S é o sinal de seleção do *multiplexer* (S=0 seleciona a entrada X). A negação é na realidade um conjunto de 4 negações (negam todos os 4 bits do barramento). Assumindo que os sinais A, C e S evoluem ao longo do tempo da forma indicada na tabela seguinte (valores dos barramentos em hexadecimal), acabe de preencher o resto da tabela (escreva todas as células, mesmo que o valor se mantenha face à anterior).



A	7		B		5	F			4		E		
C	0	1	1	0	0	0	1	0	0	1	1	0	1
S	0	0	0	1	1	1	1	1	1	1	1	0	0
W													
X	8												
Y													
Z	3												

2. (2 valores) Considere o número decimal −2764. Represente-o em notação de complemento para 2, em hexadecimal com 16 e 32 bits.

								H
								H

3. (2 + 2 valores) Considere o número hexadecimal C26H, considerando que está representado em notação de complemento para 2 com 12 bits.

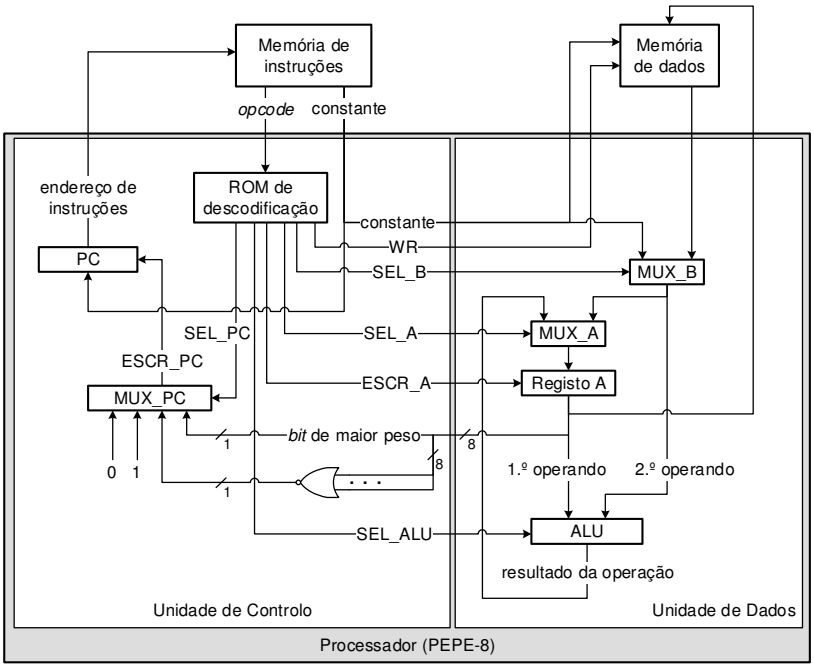
   a) Converta este número para decimal.

	decimal

   b) Indique, em hexadecimal, o menor e o maior números que consegue representar em notação de complemento para 2 com 12 bits.

	H (menor)
	H (maior)

4. (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) 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 momento imediatamente anterior à passagem do relógio de 0 para 1, na execução da instrução ADD [35H], definida como A ← A + M[35H]. 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 (ou seja, não interessa para esta instrução).

Constante	WR	SEL_A	SEL_B	ESCR_A	SEL_ALU	SEL_PC

b) Considere a instrução JAZ [*constante*], definida como (A!=0) : PC ← A + M[*constante*], ou seja, caso o registo A seja diferente de zero o processador salta para o endereço resultante da soma do registo A com o conteúdo da célula de memória com endereço *constante*. O PEPE-8 não suporta esta instrução, mas indique o que é que tinha de alterar no diagrama de blocos do PEPE-8 para passar a suportar (não é preciso indicar os valores dos sinais para a execução da instrução).

5. (2 + 2 + 3 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 ← SP-2 M[SP]←PC PC ← Endereço da Etiqueta
RET	PC ← M[SP] SP ← SP+2

Endereços				
	PLACE	0H		; início da zona de código
	N	EQU	4	
			SP, 2000H	
			R0, 0AE75H	
		MOV	R1, 4321H	
		MOV	R2, Y	
		CALL	RotX	
		MOV	[R2], R0	
	fim:	JMP	fim	
	RotX:		R1	
		MOV	R1, X	
		CALL	RotY	
		RET		
	RotY:	PUSH		; RotY: entradas R0, R1; saída R0
		MOV	R2, [R1]	
	ciclo:	ROL	R0, 1	; rotate left
		SHR	R2, 1	; shift right
		JNZ	ciclo	
	X:	WORD	N	
	Y:	WORD	0	

a) Preencha os endereços de cada instrução no lado esquerdo (preencha apenas as linhas em que tal faça sentido) e os espaços no programa. Considere que cada MOV ocupa apenas uma palavra.

b) Indique qual o valor final dos registos R0, R1 e R2

R0	
R1	
R2	

c) 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 da instrução que faz o acesso	**Endereço acedido**	**L ou E**	**Valor lido ou escrito**