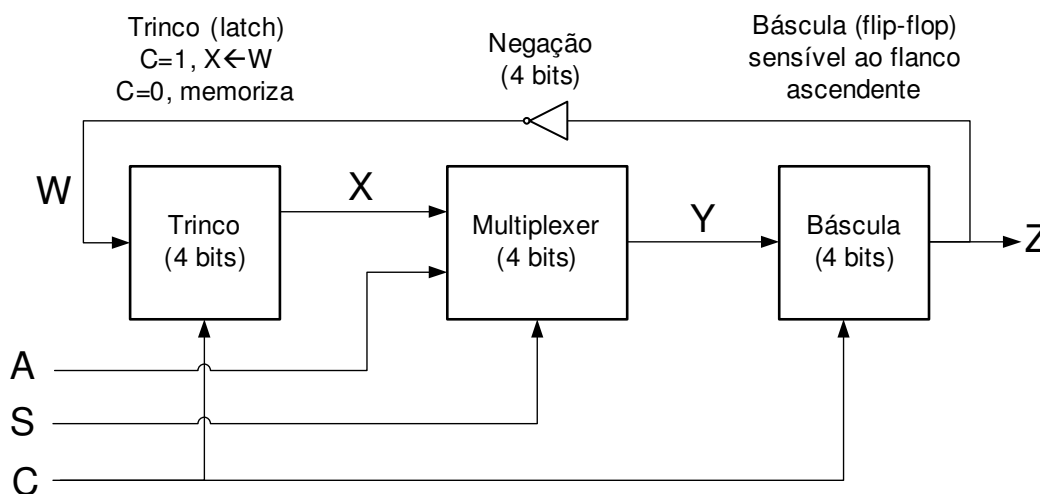


NOME		NÚMERO	
------	--	--------	--

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 básica) 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	C	7	7	7	7	7	0	0	0	B	B	B	4
X	8	7	7	7	7	7	0	0	0	B	B	B	4
Y	8	7	7	B	5	F	F	F	4	4	E	B	4
Z	3	8	8	8	8	8	F	F	F	4	4	4	B

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

					F	5	3	4	H
F	F	F	F	F	F	5	3	4	H

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

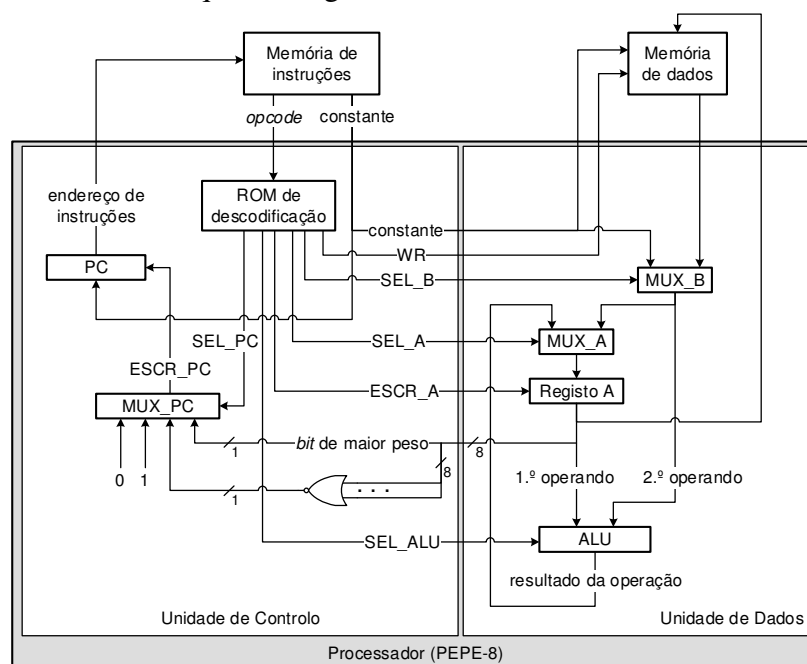
a) Converta este número para decimal.

-986 decimal

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

800 H (menor)
7FF 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 Controle. 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 $A \leftarrow A + M[35H]$. 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
35H	Não ativo	Esquerda	Direita	Ativo	Soma	Não salta

- b) Considere a instrução JAZ [*constante*], definida como $(A \neq 0) : PC \leftarrow A + M[\text{constante}]$, 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).

A entrada do registo PC tinha de ligar a um multiplexer de duas entradas, uma vinda da memória de instruções (sinal Constante) e outra vinda da saída da ALU.

A ROM de descodificação tinha de gerar mais um sinal, para controlar a entrada de seleção do novo multiplexer.

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

Endereços

	PLACE	0H		; início da zona de código
	N	EQU	4	
0000H		MOV	SP, 2000H	
0002H		MOV	R0, 0AE75H	
0004H		MOV	R1, 4321H	
0006H		MOV	R2, Y	
0008H		CALL	RotX	
000AH		MOV	[R2], R0	
000CH	fim:	JMP	fim	
000EH	RotX:	PUSH	R1	
0010H		MOV	R1, X	
0012H		CALL	RotY	
0014H		POP	R1	
0016H		RET		
0018H	RotY:	PUSH	R2	; RotY: entradas R0, R1; saída R0
001AH		MOV	R2, [R1]	
001CH	ciclo:	ROL	R0, 1	; rotate left
001EH		SHR	R2, 1	; shift right
0020H		JNZ	ciclo	
0022H		POP	R2	
0024H		RET		
0026H	X:	WORD	N	
0028H	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	73ADH
R1	4321H
R2	0028H

- 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
0008H	1FFE H	E	000AH
000EH	1FFCH	E	4321H
0012H	1FFAH	E	0014H
0018H	1FF8H	E	0028H
001AH	0026H	L	0004H
0022H	1FF8H	L	0028H
0024H	1FFAH	L	0014H
0014H	1FFCH	L	4321H
0016H	1FFE H	L	000AH
000AH	0028H	E	73ADH