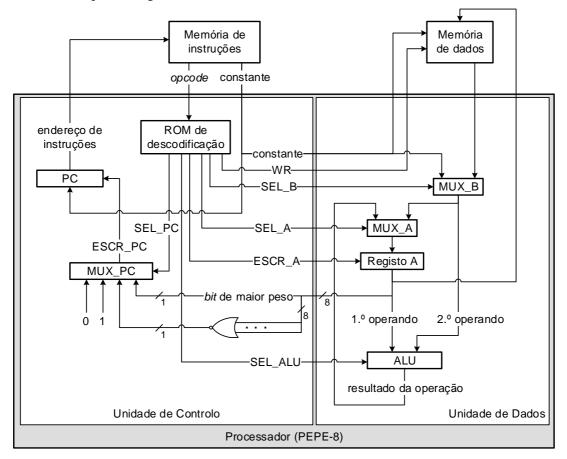
Duração: 60 minutos

3 fevereiro 2020

NOME	NÚMERO	

1. (2+1 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 que a Unidade de Controlo gera para controlar a Unidade de Dados do PEPE-8. Preencha esta tabela, especificando para cada sinal qual a sua utilidade genérica e a indicação concreta no caso de o PEPE-8 estar a executar a instrução AND [6EH], que em notação RTL quer dizer A ← A ^ M[6EH].

Sinal	Objetivo do sinal	Indicação concreta nesta instrução (ou indicação de que não interessa)
Constante		
WR		
SEL_B		
SEL_A		
ESCR_A		
SEL_ALU		

b)	Quantos bytes,	no máximo,	é que o	PEPE-8 supe	orta na memória	de dados?
----	----------------	------------	---------	-------------	-----------------	-----------

2.	(2+	2+1+2+2 v	alores) Cons	idere o seg	guinte prog	grama do l	PEPE-	16.				
		MOV MOV ROR ADD	R1, -2379 R2, 0D7EF R2, 3 R1, R2		right (rota	ção à direi	ta)					
	a)		valor de R1 da primeira i		ndecimal o	com 16 bit		ndo a H	nota	ção de	e cor	mplemento para 2) após a
	b)											para 2) com que R1 e R2 as quatro instruções.
	c)	N	mplo, o resu						CI	\\ \(\)		R1 (após os MOVs) R2 (após os MOVs) R2 final R1 final
!	d)		o <u>simétrico</u> de complemen		te da segu	ında instru	ıção, e	em dec	cimal	e em	hexa	adecimal com 32 bits, em
						Н				ecimal exadec		32 bits
	e)		s valores nur e complemer		n decimal	l) mínimo	e máx	imo q	jue é j	possív	el re	presentar com 15 bits, em
]	Mínimo:			Máxir	no:				

3. (1 + 3 + 4 valores) Considere o seguinte programa em linguagem *assembly* do PEPE-16.

Endereços					
	PLACE	2000H			SP ← SP-2
	AA	EQU	9BH	CALL Etiqueta	M[SP]←PC
	BB	EQU	56EH	•	PC ← Etiqueta
	CC	EQU	0B3ADH	RET	PC ← M[SP]
	pilha:	TABLE	100H		SP ← SP+2
	fim_pilha:				
	X:	WORD	CC		
		,, 0112			
	PLACE	0H			
			SP, fim_pilha		
		MOV	R0, AA		
		MOV	R1, BB		
	1	1110 1	R2, X		
	1	MOV	R1, [R2]		
	1	CALL	Z		
	fim:	JMP	fim		
	Z:	31411	R0		
			R1		
	_	MOV	R0, CC		
		MOV	R1, 5		
		CALL	Y Y		
	_				
	_	MOV	[R2], R0		
	_				
	_	DET			
		RET	6\ D0 \	*4-	
			ft) R0 à esquerda de N b		
		: K1 – Num	ero de bits a deslocar (N	N)	
	Y:				
	_				
	1				
	4				
	_				
	_				
	_				
	_				
	_				

- a) Preencha os <u>endereços que faltam</u> (lado esquerdo, preencha apenas as linhas em que tal faça sentido). Considera-se que cada MOV com uma constante <u>ocupa apenas uma palavra</u>.
- b) Preencha as <u>instruções que faltam</u> (ou partes delas), tendo em atenção os comentários e funcionamento do programa. <u>Use apenas as linhas que necessitar</u>.
- c) Acabe de preencher a tabela da página seguinte com informação sobre os <u>acessos à memória</u> feitos pelo programa, de leitura (L) ou escrita (E). <u>Use apenas as linhas que necessitar</u>.

Endereço em que está a instrução que faz o acesso	Endereço acedido	L ou E	Valor lido ou escrito
instrução que raz o acesso	accurdo		ou escrito