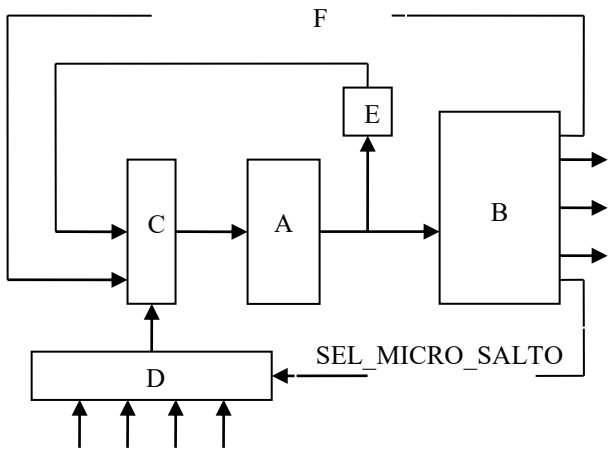
1) Considere um sistema de descodificação de endereços utilizando um descodificador de 1-8 com um processador com bus de **dados de 8 bits** e bus de endereços de 16 bits. Os endereços **A13-A11** estão ligados à entrada do descodificador.
a) Preencha o campo Início e Fim com o primeiro e último endereço do respectivo dispositivo.
b) Indique quais os bits de endereços que são necessários ligar a cada dispositivo (exemplo A0-A4).

Dispositivo	Início	Fim	Capacidade	Bits de Endereço	Saída do Descodificador
ROM1	**0000H**	**07FFH**	2Kbytes	**A0-A10**	S0
Periférico1	**1800H**	**180BH**	12bytes	**A0-A3**	S3
RAM1	**2000H**	**23FFH**	1KBytes	**A0-A9**	S4
RAM2	**3800H**	**39FFH**	512Bytes	**A0-A8**	S7

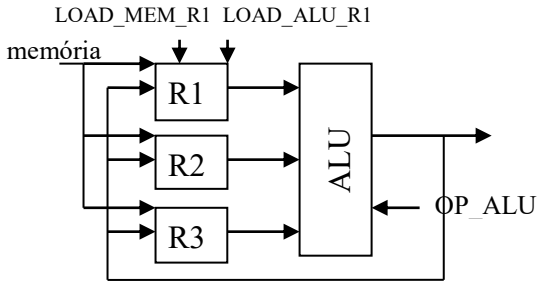
c) Se em vez de um descodificador utilizar uma PROM para realizar a descodificação de endereços, e sabendo que tem um **NOVO** Periférico2 com (Início=1000H, Fim=10FFH) e que é o seu dispositivo com endereçamento mais fino (ocupa apenas uma palavra da PROM), indique quais os endereços necessários para endereçar a PROM. _____ **A8-A15**_____

2) A figura seguinte representa uma unidade microprogramada simples. Faça a legenda da figura, preenchendo a seguinte tabela, e indique o que é e para que é que serve cada um dos blocos.



	O Que é para que serve
A	**Micro Program Counter (MPC). Indica o endereço da microinstrução corrente na ROM do microprograma.**
B	**ROM de microprograma. Contém as microinstruções**
C	**Multiplexador. Permite seleccionar de onde vem o próximo valor do Micro Program Counter (MPC): ou de MPC+1 ou do valor indicado pela própria microinstrução (quando há um salto em microcódigo)**
D	**Multiplexador. Permite escolher qual a condição que influencia o próximo valor de MPC.**
E	**Somador com 1. Permite gerar MPC+1 a partir de MPC**
F	**Micro-Salto. Endereço da próxima microinstrução para onde saltar, se for esse o caso.**

3) Considere o circuito seguinte que implementa o resto de uma divisão entre dois números inteiros. O circuito é controlado por uma unidade microprogramada. Os sinais do tipo LOAD_MEM_Rn memorizam no Registo Rn (em que n=1,2,3,4) os dados provenientes da memória. Os sinais LOAD_ALU_Rn memorizam os dados vindos da ALU (**a figura exemplifica apenas os sinais para R1)** Os sinais disponíveis para SEL_MICRO_SALTO são (0, 1, 2=R1_Menor_R2, 3=R2_UM).
As opções disponíveis para OP_ALU são (A=SUB_R1_R2, B=PASS_R1, C=ZERO_R3, etc.)



Endereço na ROM	Operações	LOAD_ALU_R1	LOAD_MEM_R1	LOAD_ALU_R2	LOAD_MEM_R2	LOAD_ALU_R3	LOAD_MEM_R3	OP_ALU	SEL_MICRO_SALTO	MICRO_SALTO
0	Inicializa R1 com X, que vem da memória.		x							
1	Inicializa R2 com Y, que vem da memória.				x					
3	Inicializa R3 a 0					x		C		
4	Se R1 < R2 salta para 6								2	6
5	Subtrai a R1 o R2 coloca o resultado em R2			x				A		
6	Move R1 para R3					x		B		
7	Salta para 7								1	7

PASS_R1 = Operação da ALU que permite colocar na saida da ALU o valor de R1 sem ser modificado.