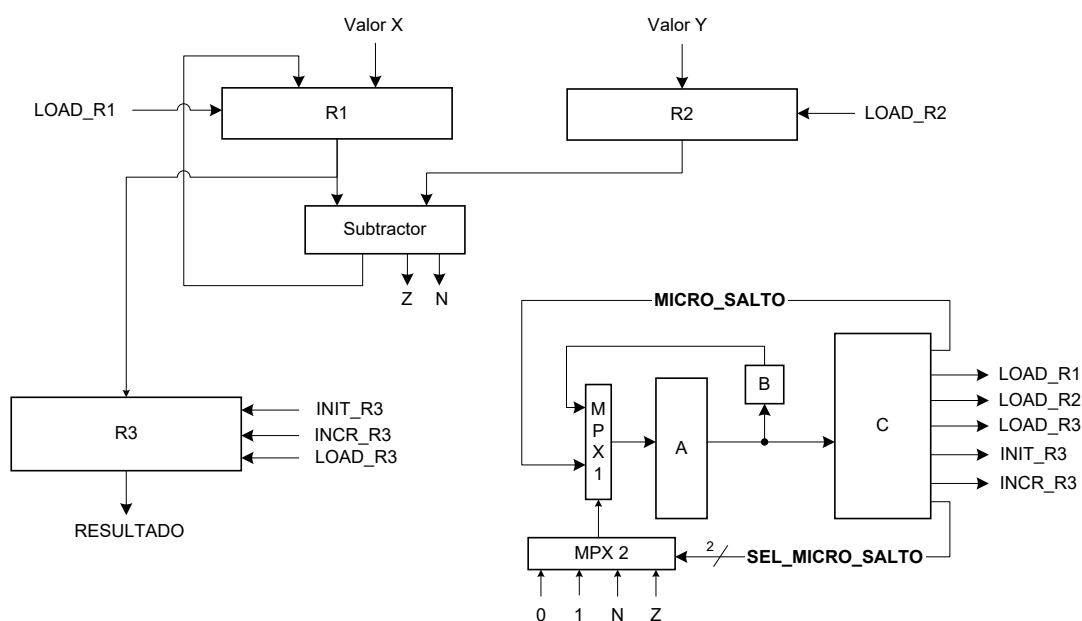


1 - Considere o circuito seguinte, que implementa a operação de divisão inteira usando microprogramação e em que:

- R1, R2 e R3 são registos de 16 bits;
- Para além do resultado R1-R2, o subtractor produz dois bits, Z e N, que ficam a 1 se o resultado da subtracção for zero ou negativo, respectivamente;
- Os sinais INIT_R3, INCR_R3 e LOAD_R3 inicializam a 0000H, incrementam de uma unidade e carregam um valor no registo R3, respectivamente;
- Os sinais LOAD_R1 e LOAD_R2 carregam os valores X e Y nos registos R1 e R2, respectivamente;
- O multiplexer MPX1 selecciona a entrada de baixo se o valor à saída do multiplexer MPX2 for 1.



a) - Indique o tipo (multiplexer, registo, somador, RAM, ROM, processador, decodificador, etc) e a função dos blocos A, B e C nesta figura.

Bloco	Tipo	Para que serve
A	Registo	Micro-Program Counter (MPC). Indica o endereço da microinstrução corrente.
B	Somador	Permite incrementar o MPC de uma unidade
C	ROM	ROM de microprograma. Contém os bits das microinstruções, incluindo os sinais de controlo do circuito

b) - Preencha a tabela seguinte com as microinstruções necessárias para implementar a funcionalidade da divisão inteira. Use “SIM” para indicar os sinais que devem estar activos em cada ciclo de relógio, e deixe em branco as células dos sinais não activos.

Endereço	Microinstrução (RTL)	LOAD_R1	LOAD_R2	INIT_R3	INCR_R3	LOAD_R3	SEL_MICRO_SALTO	MICRO_SALTO
0	$R1 \leftarrow X;$ $R2 \leftarrow Y;$ $R3 \leftarrow 0$	SIM	SIM	SIM				
1	$(R1 < R2) : MPC \leftarrow 5$						Entrada N	5
2	$R1 \leftarrow R1 - R2$	SIM						
3	$R3 \leftarrow R3 + 1$				SIM			
4	$MPC \leftarrow 1$						Entrada 1	1
5	$MPC \leftarrow 5$						Entrada 1	5

c) - Quantos bits deve ter o sinal MICRO_SALTO? Justifique.

3, pois o número de microinstruções usadas é 5 (maior que 4 mas menor que 8)

d) - Imagine que agora quer que o circuito implemente a funcionalidade de “resto da divisão inteira”. Preencha a tabela seguinte com as microinstruções necessárias (mas desta vez preencha também a descrição em RTL). Não é obrigatório usar as linhas todas da tabela.

Endereço	Microinstrução (RTL)	LOAD_R1	LOAD_R2	INIT_R3	INCR_R3	LOAD_R3	SEL_MICRO_SALTO	MICRO_SALTO
0	$R1 \leftarrow X;$ $R2 \leftarrow Y$	SIM	SIM					
1	$(R1 < R2) : MPC \leftarrow 4$						Entrada N	4
2	$R1 \leftarrow R1 - R2$	SIM						
3	$MPC \leftarrow 1$						Entrada 1	1
4	$R3 \leftarrow R1$					SIM		
5	$MPC \leftarrow 5$						Entrada 1	5
6								
7								

- 2 - Considere um processador que inclui uma cache de dados associativa por conjuntos de 2 vias e com um total de 8 blocos de 1 palavra cada. Suponha que inicialmente a cache está vazia e que o processador vai fazendo acessos de dados aos endereços indicados na tabela em baixo. Notas:

- (i) Assuma que o endereçamento é por palavras, não por bytes;
- (ii) A política de substituição em cada conjunto é alternada entre vias.

a) - Preencha a tabela seguinte com:

- (i) os sucessivos conteúdos da cache ao longo dos vários acessos (use a notação [*endereço*] para se referir à informação que a cache guarda sobre um dado valor armazenado em memória principal nesse endereço);
- (ii) a indicação em cada acesso se houve sucesso (*hit*) ou falha (*miss*).

Endereço	Hit / miss	Nº bloco via 0				Nº bloco via 1			
		0	1	2	3	0	1	2	3
361H	miss		[361H]						
123H	miss		[361H]		[123H]				
2A7H	miss		[361H]		[123H]				[2A7H]
C0CH	miss	[C0CH]	[361H]		[123H]				[2A7H]
361H	hit	[C0CH]	[361H]		[123H]				[2A7H]
499H	miss	[C0CH]	[361H]		[123H]		[499H]		[2A7H]
361H	hit	[C0CH]	[361H]		[123H]		[499H]		[2A7H]
123H	hit	[C0CH]	[361H]		[123H]		[499H]		[2A7H]
2A7H	hit	[C0CH]	[361H]		[123H]		[499H]		[2A7H]
100H	miss	[C0CH]	[361H]		[123H]	[100H]	[499H]		[2A7H]
499H	hit	[C0CH]	[361H]		[123H]	[100H]	[499H]		[2A7H]

- b) - Suponha que o tempo de acesso com sucesso (*hit*) desta *cache* é de 5 nanossegundos, enquanto um acesso com insucesso (*miss*), incluindo a respectiva recuperação (leitura da memória principal), demora cerca de 60 ns. Se num dado programa a taxa média de sucesso for de 95%, calcule o tempo médio de acesso à memória nesse programa.

O tempo médio T_m de acesso à memória nestas condições será:

$$T_m = 0,95 * 5 \text{ ns} + 0,05 * 60 \text{ ns} = 4,75 + 3 = 7,75 \text{ ns}$$

- 3 - Suponha que o PEPE, com os seus 16 bits de endereço e endereçamento de byte, suporta memória virtual com páginas de 256 bytes. Assuma que a memória física é de apenas 8 Kbytes e que a TLB é uma cache totalmente associativa de 4 entradas, estando já carregada com os seguintes valores:

Válida	Página alterada	Nº de página virtual	Nº de página física
1	1	73H	3DH
1	0	A6H	22H
1	0	04H	1CH
0	0	FAH	0EH

- a) - Preencha a tabela seguinte:

	Valor	Breve justificação
Dimensão do espaço virtual	64K bytes	2^{16} bits de endereço virtual
nº de páginas virtuais	256	64K bytes / 256 (dimensão da página)
nº de páginas físicas	32	8K bytes / 256 (dimensão da página)

- b) - Acabe de preencher a tabela seguinte:

Endereço virtual	Endereço físico
04FAH	1CFAH
A6A6H	22A6H
733DH	3D3DH
040EH	1C0E H

- c) - Suponha que, para além das páginas virtuais indicadas na TLB em cima, estão ainda carregadas em memória principal as páginas virtuais nºs 5CH e 89H (e mais nenhuma – as restantes estão em disco). Indique genericamente o que aconteceria se o processador acesse em sequência aos endereços virtuais indicados em seguida, tomando como base o estado da TLB indicado em cima. Nota: a solução não é única, pelo que pode tomar as suas opções (mas explicita-as).

Endereço virtual acedido	O que é que acontecerá
FA89H	A página virtual FAH não está mapeada na TLB (o seu bit Válida está a 0), logo começa por dar TLB miss. No entanto, também não está carregada em memória principal. Logo, este acesso provocará uma falta de página (page-fault), cuja rotina de excepção deverá resolver (carregando a página em memória principal e actualizando a tabela de páginas). Seguidamente o acesso será repetido, dando TLB miss, que agora já é mais fácil de resolver (basta carregar na última linha da TLB o mapeamento correspondente numa página física).
5CFAH	A página virtual 5CH não está mapeada na TLB, logo começa por dar TLB miss. No entanto, não dá falta de página, pois ela está carregada em memória principal. Basta actualizar a TLB, mas como esta já encheu é preciso libertar uma das entradas para incluir o mapeamento da página virtual 5CH. Qual entrada é deitada fora depende da política usada (poderá ser a mais antiga, por exemplo). Se calhar a ser a página virtual 73H, não basta substituir a entrada correspondente da TLB. Tem de se actualizar a página em si no disco, uma vez que esta tinha sido alterada.