

NOME: _____

TURMA _____



FEUP
Universidade do Porto
Faculdade de Engenharia

Departamento de Engenharia Electrotécnica
e de Computadores

Sistemas Digitais (2000/2001)**Correcção****1ª chamada – 8/Janeiro/2001****Duração: 2horas , sem consulta.****Antes de iniciar a prova, tenha em atenção as seguintes recomendações:**

- Leia atentamente toda a prova antes de a iniciar.
- Mostre e justifique adequadamente todos os passos das suas respostas.
- A prova deverá ser resolvida no enunciado. Se necessário, utilize o verso para continuar a sua resolução.
- Assine todas as folhas que entregar, indicando em cada uma o número de páginas/folhas que entregou.

1 - Considere a sequência de dígitos 101**a) Diga qual é o seu valor se essa sequência representar:****i) um número inteiro em base 8.**

Se 101 representar um número inteiro em base 8, então o seu valor é $1 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 = 65$

ii) um número inteiro em base 16 com 9 bits e complemento para dois.

Se 101 representar um número em base 16, então podemos escrever a sua representação em binário substituindo cada dígito hexadecimal pela representação binária correspondente:

$$101_{16} = 000100000001_2$$

Se é dito que o número dado tem 9 bits, então deveremos desprezar os 3 bits (zeros) da esquerda ficando apenas:

$$101_{16} = 100000001_2$$

Como é também dito que 101_{16} representa um número em complemento para dois, então podemos concluir que o número é negativo (o bit mais significativo é 1) e o seu valor absoluto é obtido calculando o seu simétrico (que é complemento para dois do número dado):

$$-100000001_2 = 011111110 + 1 = 01111111 = 255$$

Assim, se 101_{16} representar um número com 9 bits em complemento para dois , o seu valor é -255

b) Determine o número com 6 bits representado em complemento para dois, que adicionado ao número 01001 representado em complemento para dois com 5 bits, dá o resultado -2. Efectue as operações aritméticas em binário que achar convenientes (utilize o verso da folha).

NOME: _____

TURMA _____

Pretendemos calcular um valor X tal que $X+Y=-2$, onde Y é o número binário 01001. Se é dito que X deve ser um número com 6 bits em complemento para dois, deveremos transformar os operandos Y e -2 para esse formato:

$$Y = 01001 = 001001 \text{ (é positivo, a extensão de sinal acrescenta um zero)}$$

$$-2 = -000010 = 111101 + 1 = 111110$$

Assim, o número pedido X pode ser obtido como:

$$X = -2 - Y = 111110 - 001001$$

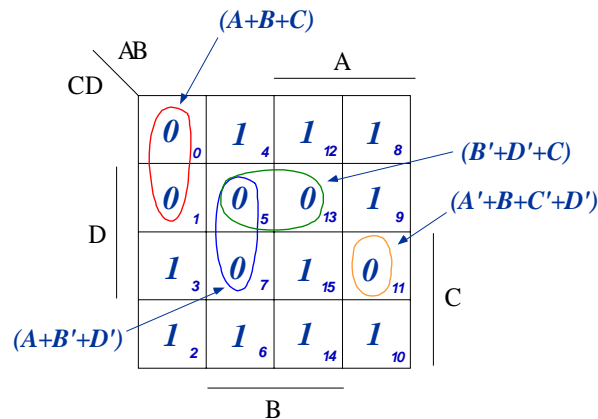
$$\begin{array}{r} 111110 \\ -001001 \\ \hline 110101 \end{array}$$

O valor pedido X é 110101 (representa a quantidade -11)

2

- a) Considere a função booleana $F(A,B,C,D)$ que assume '1' quando ABCD representa valores múltiplos de 2, e a função $G(A,B,C,D)$ que assume '1' quando ABCD representa valores múltiplos de 3 (A é o bit mais significativo e D o bit menos significativo). Apresente uma expressão simplificada na forma produto-de-somas para a função $Z=F+G$ (soma lógica das funções F e G) (se necessário utilize o verso da folha).

A função pedida $Z(A,B,C,D)$ pode ser representada directamente no mapa de Karnaugh, preenchendo uns nas posições correspondentes aos minterms cujo número é múltiplo de 2 ou de 3. Tenha em atenção a ordenação dos minterms no mapa, de acordo com a disposição das variáveis representada. Como se pretende obter uma expressão minimizada na forma produto-de-somas, agrupam-se os zeros da forma representada no mapa:



A expressão mínima produto-de-somas será então:

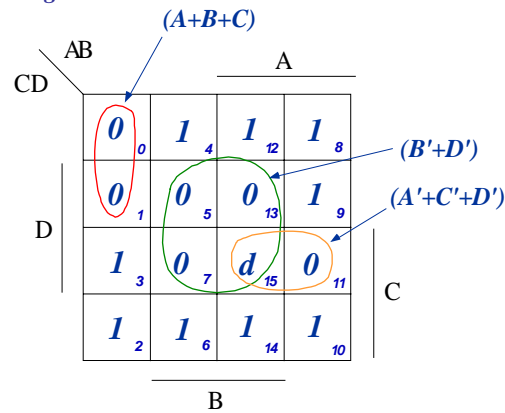
$$Z(A,B,C,D) = (A+B+C).(A+B'+D').(B'+D'+C).(A'+B+C'+D')$$

- b) Considere agora uma função H idêntica a Z , excepto no termo $ABCD=1111$ em que é indiferente (*don't care*). Construa um circuito minimizado utilizando apenas portas lógicas NOR de 2 ou 3 entradas que realize a função $H(A,B,C,D)$ (utilize o verso da folha)

NOME: _____

TURMA _____

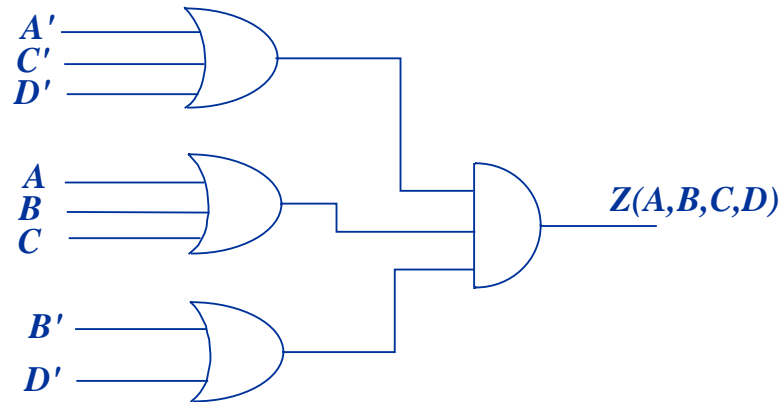
Para construir um circuito que contenha apenas portas lógicas NOR (com 2 ou 3 entradas), é conveniente começar por desenhar o circuito minimizado do tipo OR-AND, obtido da expressão mínima do tipo produto-de-somas. Se o termo 15 ($ABCD=1111$) for indiferente podemos agrupar os zeros no mapa de Karnaugh da forma seguinte:



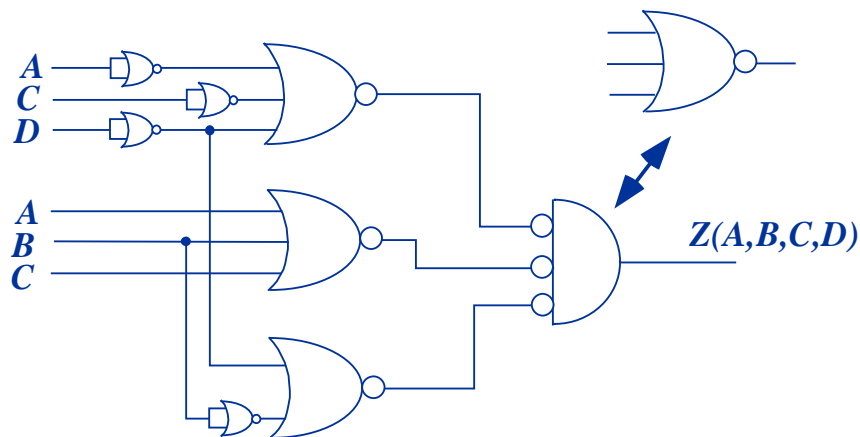
A nova expressão do tipo POS é:

$$Z(A,B,C,D) = (A+B+C).(A'+C'+D').(B'+D')$$

O circuito do tipo OR-AND que realiza esta função é:



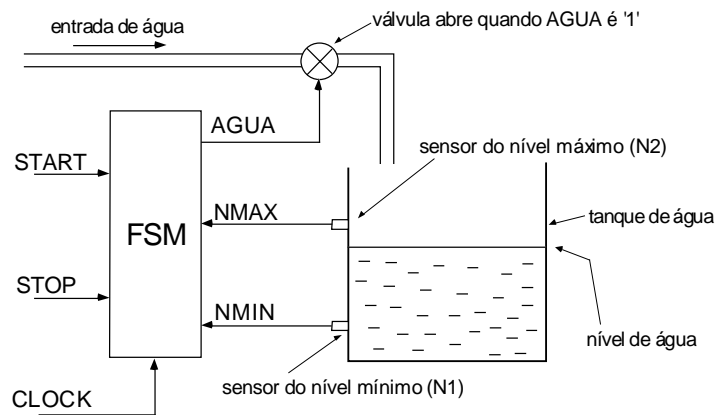
e o circuito equivalente só com portas NOR de duas ou 3 entradas é:



NOME: _____

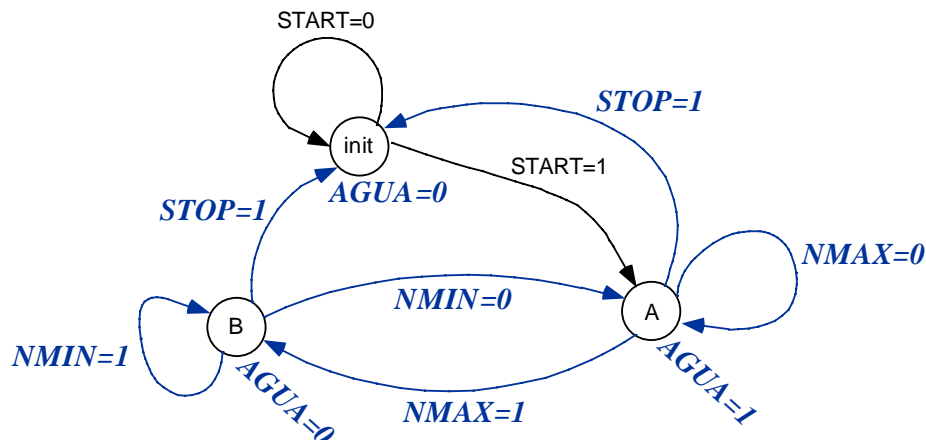
TURMA _____

3 – Pretende-se projectar o sistema de controlo do nível de água de um tanque para abastecimento público. O sistema tem uma saída **AGUA** que quando é activada (nível lógico alto) abre uma válvula de entrada de água. Para controlar o nível de água dispõe-se de duas entradas **NMAX** e **NMIN** provenientes de sensores de nível de água, que são activadas (nível lógico alto) quando o nível no tanque se torna, respectivamente, maior do que o nível máximo N2 ou menor do que o nível mínimo N1. O sistema dispõe ainda de duas entradas ligadas a um botão de arranque (**START**) para iniciar o funcionamento do sistema e a um botão de paragem (**STOP**) para o desligar. Estas entradas tomam o valor lógico alto quando os botões respectivos são pressionados.



Para controlar o nível de água no tanque, mantendo-o entre N1 e N2, deve-se abrir a válvula de entrada de água sempre que o nível de água no tanque for inferior a N1, e fechar quando for ultrapassado o nível N2.

Complete o diagrama de transição de estados do sistema descrito, utilizando apenas os estados já representados e os nomes simbólicos referidos no texto para as entradas e saída.

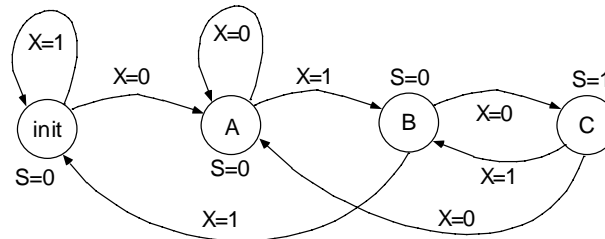


NOME: _____

TURMA _____

4 – O diagrama de transição de estados da figura representa uma máquina de Moore com uma entrada X e uma saída S. A saída S toma o valor 1 quando, em 3 estados consecutivos, é detectada na entrada X a sequência **010**. A sequência a detectar pode ser parcialmente sobreposta da forma que se exemplifica na figura:

Entrada X: 0 0 1 0 1 0 1 1 0 1 0 0
Saída S: 0 0 0 1 0 1 0 0 0 0 1 0



- a) Construa a tabela de transição de estados, atribuindo uma codificação apropriada aos estados.

Utilizando primeiro os nomes simbólicos atribuídos aos estados, a tabela de transição pedida é:

estado actual	próximo estado		saída S
	X=0	X=1	
init	A	init	0
A	A	B	0
B	C	init	0
C	A	B	1

Codificando agora os estados como: init=00, A=01, B=10 e C=11, serão necessários 2 flip-flops do tipo D e a tabela de transição de estados fica:

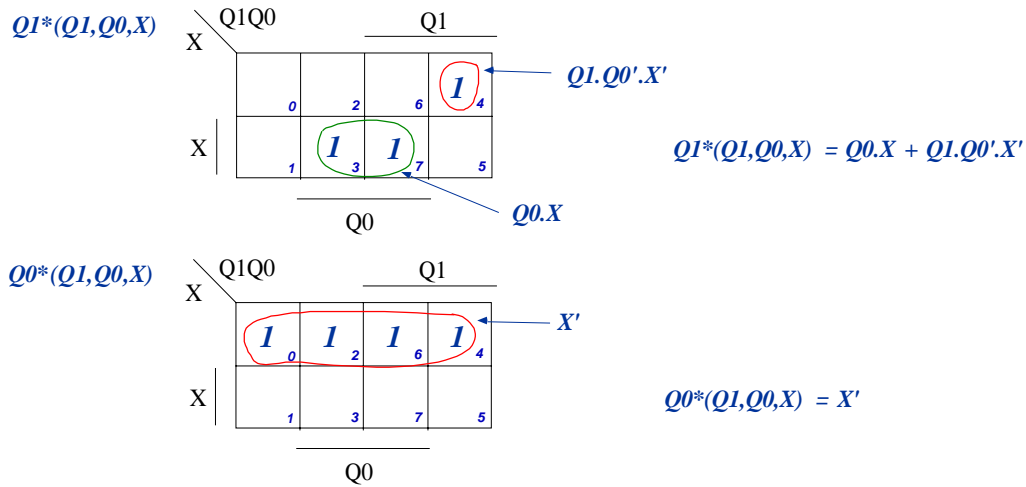
estado actual Q1 Q0	próximo estado Q1* Q0*		saída S
	X=0	X=1	
0 0	0 1	0 0	0
0 1	0 1	1 0	0
1 0	1 1	0 0	0
1 1	0 1	1 0	1

NOME: _____

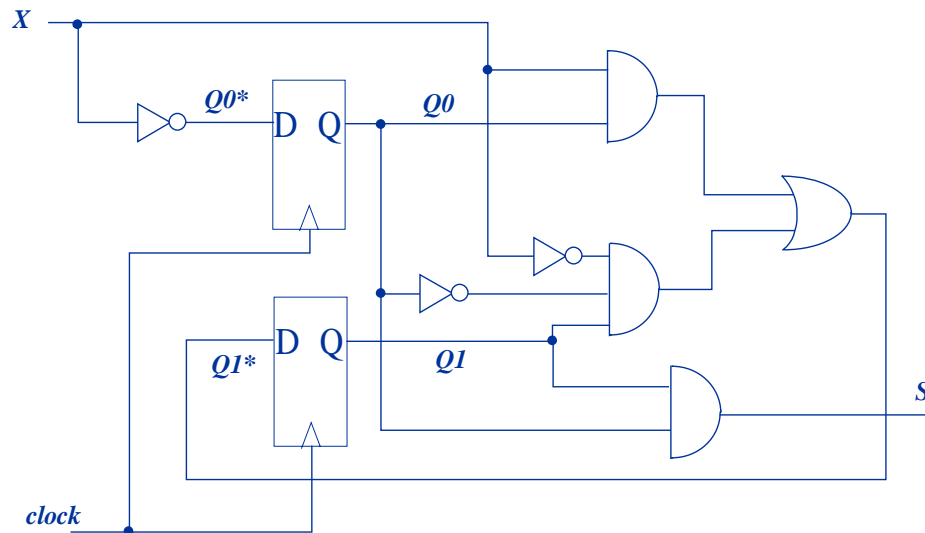
TURMA _____

- b) Desenhe o esquema do circuito lógico que implementa a máquina de estados, utilizando *flip-flops* do tipo D. (utilize o verso da folha)

Partindo da tabela de transição de estados construída na alínea anterior, vamos obter expressões minimizadas para as funções $Q1^$ e $Q0^*$ na forma soma-de-produtos. Note que pela tabela de transição de estados pode ver-se facilmente que a saída S é apenas a função lógica AND das variáveis de estado $Q1$ e $Q0$:*



um circuito lógico que realiza esta máquina de estados é:

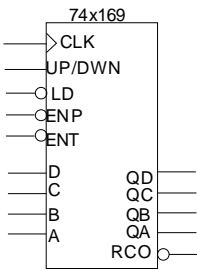


NOME: _____ TURMA _____

5 –

- a) Construa um circuito síncrono baseado num contador do tipo *up/down* (74x169) e em circuitos lógicos adicionais capaz de gerar, nas saídas Q_D, Q_C, Q_B, Q_A do contador, a seguinte sequência de valores (admitindo que o estado inicial é igual a $Q_D, Q_C, Q_B, Q_A=0000$):

0, 1, 2, 3, 4, 5, 10, 9, 8, 3, 4, 5, 10, 9,



74x169				estado presente				próximo estado			
UP/DWN	/LD	/ENT	/ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*
x	0	x	x	x	x	x	x	D	C	B	A
x	1	1	x	x	x	x	x	QD	QC	QB	QA
x	1	x	1	x	x	x	x	QD	QC	QB	QA
1	1	0	0	N (se $N < 15$)				N + 1			
0	1	0	0	N (se $N > 0$)				N - 1			
1	1	0	0	1	1	1	1	0	0	0	0
0	1	0	0	0	0	0	0	1	1	1	1

NOME:_____

__TURMA

Analisando a sequência pretendida, representada em binário:

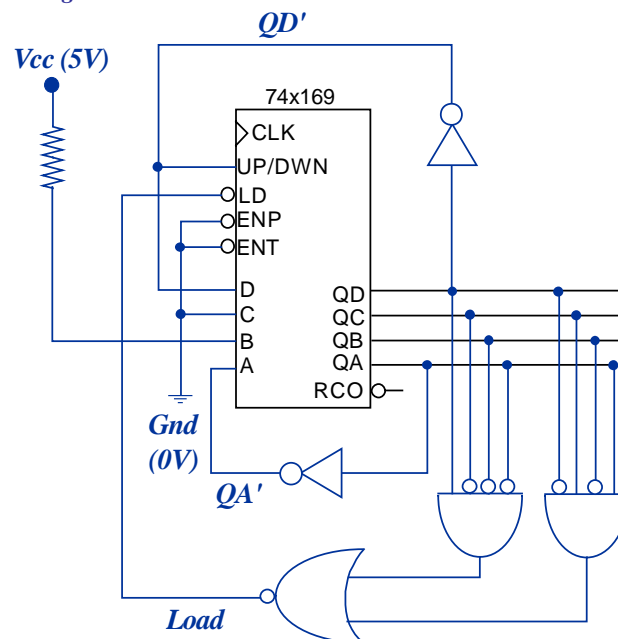
0	0 0 0 0	
1	0 0 0 1	
2	0 0 1 0	
3	0 0 1 1	conta para cima
4	0 1 0 0	
5	0 1 0 1	
10	1 0 1 0	load 1010 quando saída é 0101
9	1 0 0 1	conta para baixo
8	1 0 0 0	
3	0 0 1 1	load 0011 quando saída é 1000
4	0 1 0 0	conta para cima
5	0 1 0 1	
10	1 0 1 0	load 1010 quando saída é 0101
9	1 0 0 1	

podemos concluir o seguinte:

1 – o bit mais significativo (QD) define o sentido da contagem: quando QD=0 conta para cima e quando QD=1 conta para baixo. Podemos por isso controlar a entrada UP/DWN com o sinal QD negado.

2 – são efectuadas duas operações load: quando a saída é 0101 é carregado o valor 1010 e quando é 1000 é carregado 0011. Analisando a relação entre os valores que provocam o load e os valores carregados para o contador podemos concluir que em ambos os casos $QD^*=QD'$, $QA^*=QA'$, $QC=0$ e $QB=1$.

O circuito pretendido é o seguinte:



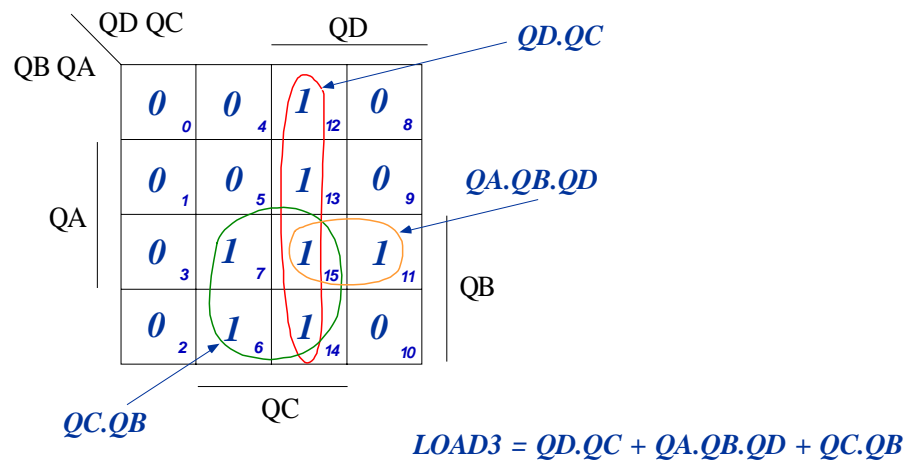
NOME: _____

TURMA _____

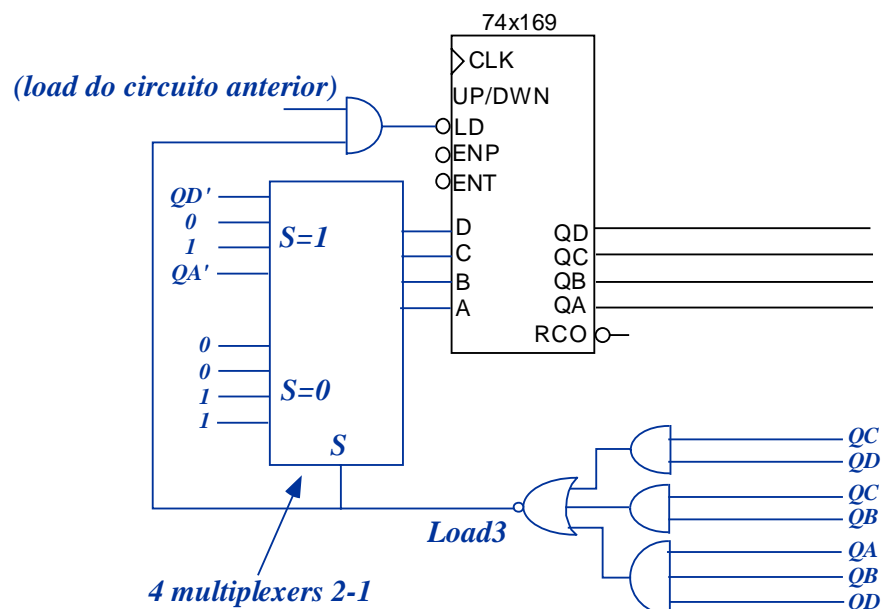
- b) Modifique o circuito que construiu por forma a que as saídas do contador sejam reiniciadas com o valor 3 sempre que for atingido um estado não pertencente à sequência especificada em a) (utilize o verso da folha).

Os estados não pertencentes à sequência de contagem são: 6, 7, 11, 12, 13, 14, 15. Quando aparecer na saída do contador um desses estados, deve ser feito um load com o valor 3, caso contrário deve ser seguida a operação descrita acima.

Vamos começar por construir um circuito que detecte um dos estados fora da sequência normal (função LOAD3):

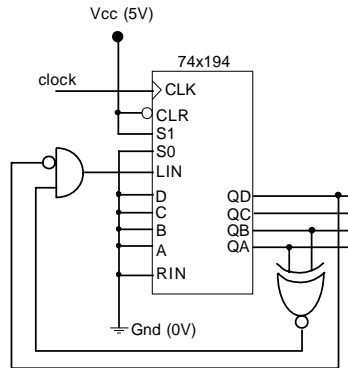


Como a operação normal do circuito também utiliza a operação de load, é necessário utilizar um multiplexer para escolher o valor a carregar no contador: quando é um load “normal”, é carregado o valor definido pelo circuito construído na alínea anterior; quando acontece o load devido a estado inválido deve ser escolhido o valor 3. Um circuito que realiza esta funcionalidade é (apenas se mostram as alterações ao circuito apresentado na alínea anterior).



NOME: _____

TURMA _____

6 – Considere o circuito da figura, baseado num *shift-register* 74x194.

Universal Shift-register 74x194							
função	S1	S0	QA*	QB*	QC*	QD*	
hold	0	0	QA	QB	QC	QD	
shift right	0	1	RIN	QA	QB	QC	
shift left	1	0	QB	QC	QD	LIN	
load	1	1	A	B	C	D	

- a) Admitindo o estado inicial $Q_A, Q_B, Q_C, Q_D = 0000$, determine a sequência (em binário) produzida nas saídas Q_A, Q_B, Q_C, Q_D .

Pela configuração das entradas S1 e S0 do shift-register podemos concluir que é sempre efectuado um deslocamento para a esquerda (shift-left), sendo os bits que entram na entrada LIN definidos pela equação:

$$LIN = QD' \cdot (QA \oplus QB)'$$

A sequência produzida nas saídas é:

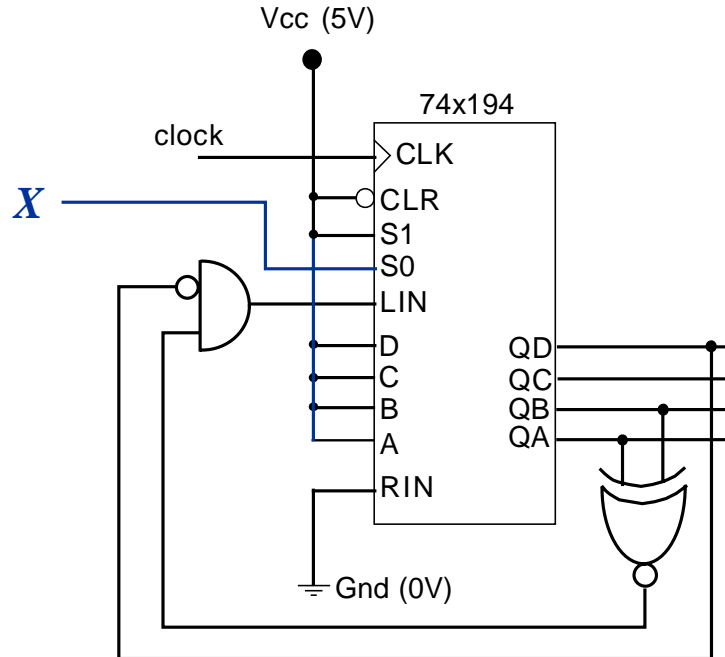
(decimal)	QA	QB	QC	QD	LIN
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
5	0	1	0	1	0
10	1	0	1	0	0
6	0	1	0	0	0
8	1	0	0	0	0
0	0	0	0	0	1

NOME: _____

TURMA: _____

- b) Modifique o circuito apresentado de forma a acrescentar-lhe uma entrada X, activa no nível lógico alto, que permita reinicializar as saídas com o valor $Q_A, Q_B, Q_C, Q_D = 1111$.

Para iniciar as saídas com 1111, basta colocar as entradas Q_A, Q_B, Q_C, Q_D com 1111 e ligar a entrada X à entrada S0 (note que a operação Load é seleccionada quando $S1=1$ e $S0=1$). O circuito resultante é:



- FIM -