

NOME: _____ CORRECÇÃO _____ TURMA _____



FEUP
Universidade do Porto
Faculdade de Engenharia

Departamento de Engenharia Electrotécnica
e de Computadores

Sistemas Digitais (2000/2001)

2ª chamada – 11/Julho/2001

Duração: 2 horas , 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 -a) Dado o número hexadecimal $1A0_{16}$, diga qual é o seu valor se representar:

i) um número inteiro em complemento para dois com 9 bits.

O número dado tem a seguinte representação binária em 12 bits (3 dígitos hexadecimais):

$$1A0_{16} = 000110100000_2$$

Como é dito que $1A0_{16}$ representa um número inteiro em complemento para dois com 9 bits (os 9 bits da direita), então podemos concluir que representa um número negativo porque o bit de sinal (MSB) é 1. Para obter o seu valor absoluto temos de calcular o seu simétrico, i.e. o complemento para dois desse número:

$$-(110100000) = 001011111 + 1 = 001100000_2 = 2^6 + 2^5 = 64 + 32 = 96_{10}.$$

Assim, o valor do número $1A0_{16}$ representando um número inteiro em complemento para dois com 9 bits é -96 .

ii) um número inteiro em complemento para dois com 10 bits

Se $1A0_{16}$ representar agora um número inteiro em complemento para dois com 10 bits (os 10 bits da direita), então podemos concluir que o seu valor é uma quantidade positiva porque o bit de sinal (MSB) é 0, sendo o seu valor igual a:

$$0110100000_2 = 2^8 + 2^7 + 2^5 = 256 + 128 + 32 = 416_{10}$$

b) Obtenha o resultado da adição, efectuada em complemento para dois com 9 bits, entre o número dado em a) e -27_{10} .

Primeiro vamos representar -27 em complemento para dois com 9 bits. Para isso comecemos por calcular a representação de 27 em binário:

$$\begin{array}{r} 27 \overline{) 2} \\ \text{LSB} \rightarrow 1 \quad 13 \overline{) 2} \\ \quad 1 \quad 6 \overline{) 2} \\ \quad \quad 0 \quad 3 \overline{) 2} \\ \quad \quad \quad 1 \quad 1 \leftarrow \text{MSB} \end{array}$$

Assim, $27_{10} = 11011_2 = 000011011_2$. A representação de -27 em complemento para dois com 9 bits obtém-se calculando o complemento para dois de $+27$:

NOME: _____

CORRECÇÃO _____

TURMA _____

$$-27 = -(000011011) = 111100100 + 1 = 111100101_2$$

Efectuando agora a adição binária entre o número dado em a) e -27 :

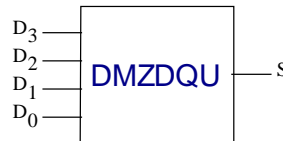
$$\begin{array}{r} 110100000 \\ +111100101 \\ \hline 111000101 \end{array}$$

Como não ocorre overflow (porque os bits de sinal dos operandos são iguais), podemos concluir que o resultado da adição pode ser representado em 9 bits por 110000101 (representa a quantidade $-123 = (-27) + (-96)$).

- c) Determine qual é a constante inteira positiva de 6 bits que é necessário adicionar a valores representados em complemento para dois com 6 bits, para que passem a representar números inteiros positivos (sem sinal) com 6 bits no intervalo $[0, +63]$.

Com uma representação binária em complemento para dois com 6 bits podem-se representar números inteiros com sinal no intervalo $[-2^{(6-1)}, +2^{(6-1)}-1] = [-32, +31]$. Somando a constante inteira $+32$ a números representados neste intervalo passamos a ter valores que pertencem a $[0, +63]$. A representação binária de $+32$ em 6 bits é 100000 (note que, em complemento para dois com 6 bits o número 100000₂ representa o valor -32).

2 — Considere um circuito combinacional com 4 entradas $D_3D_2D_1D_0$, e uma saída S que é 1 sempre que no conjunto das entradas o número de zeros seja maior ou igual ao número de uns.

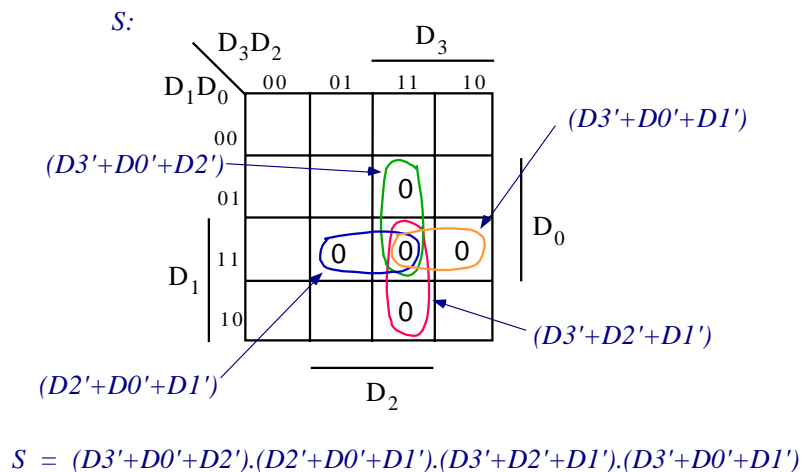


- a) Projecte o circuito utilizando apenas portas NOR de 3 entradas (utilize o verso da folha).

Como pretendemos construir um circuito que apenas contenha portas NOR, é conveniente obter uma função minimizada na forma produto-de-somas:

Tabela de verdade:

D_3	D_2	D_1	D_0	S
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

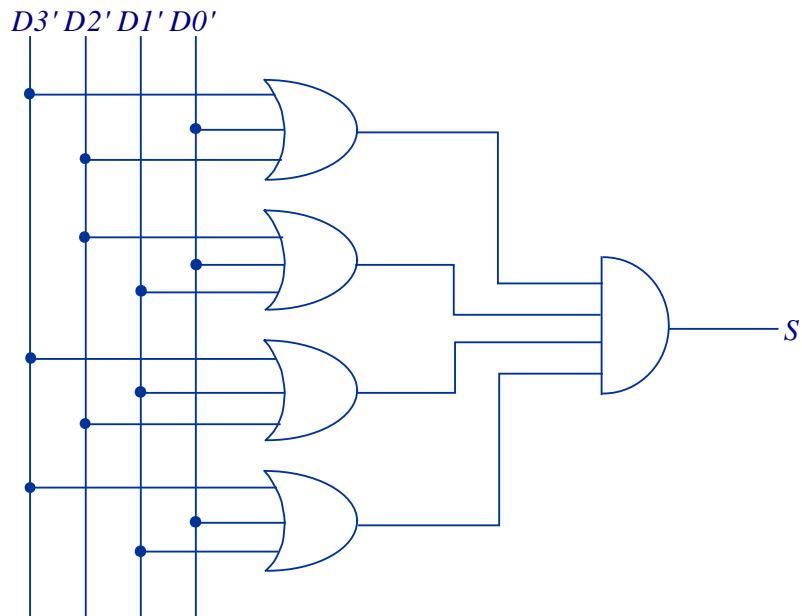


NOME: _____

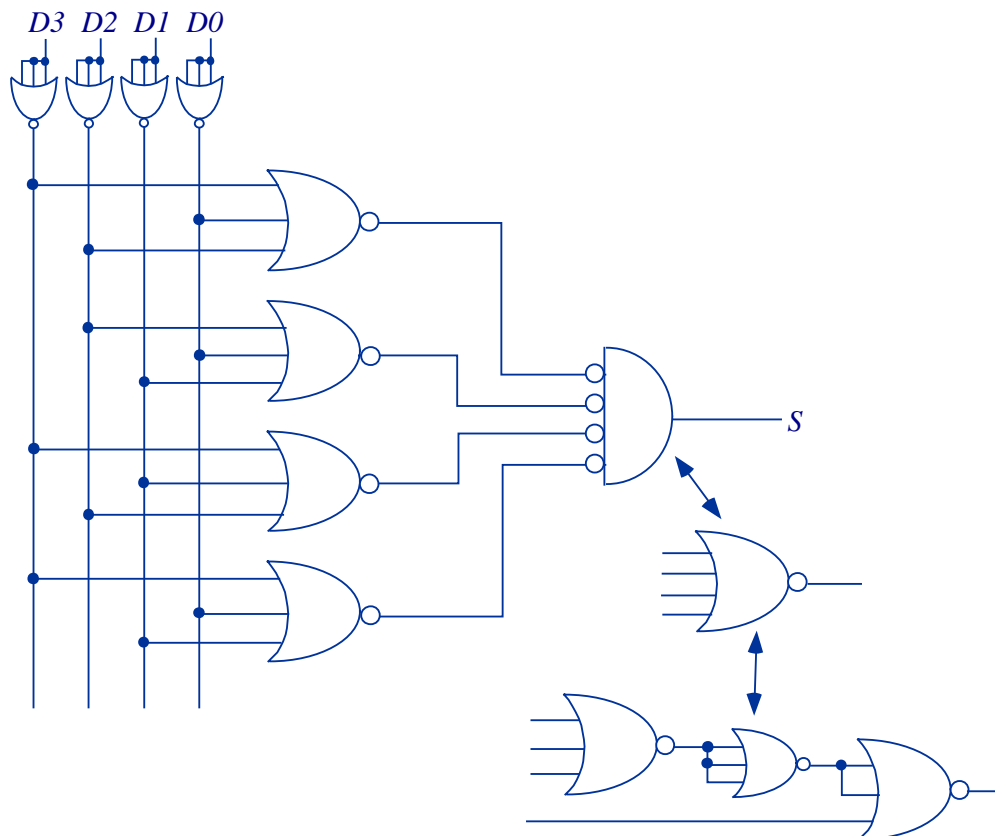
CORRECÇÃO _____

TURMA _____

Circuito OR-AND (tradução directa da expressão minimizada produto-de-somas):



E utilizando apenas portas NOR com 3 entradas:

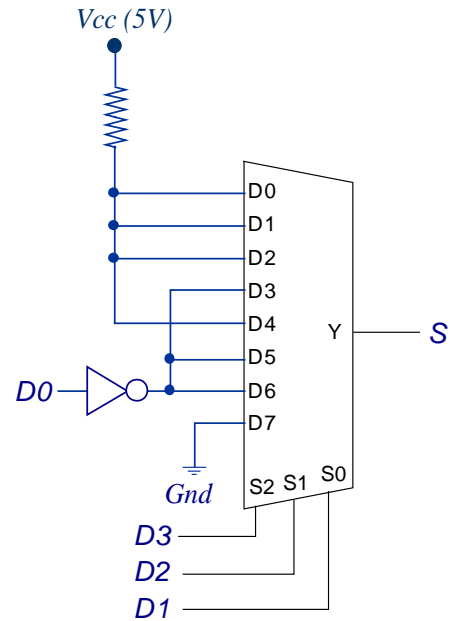


NOME: _____ CORRECÇÃO _____ TURMA _____

- b) Projecte o mesmo circuito utilizando um multiplexer 8÷1 (3 linhas de selecção).

Vamos ligar às linhas de selecção do multiplexer as variáveis D3~D1. Desta forma, as entradas D0~D7 do multiplexer serão ligadas a 0, 1, D0 ou D0', de acordo com a tabela de verdade da função a implementar:

D ₃	D ₂	D ₁	D ₀	S
0	0	0	0	1 → 1
0	0	0	1	1 → 1
0	0	1	0	1 → 1
0	0	1	1	1 → 1
0	1	0	0	1 → 1
0	1	0	1	1 → 1
0	1	1	0	1 → D0'
0	1	1	1	0 → D0'
1	0	0	0	1 → 1
1	0	0	1	1 → 1
1	0	1	0	1 → D0'
1	0	1	1	0 → D0'
1	1	0	0	1 → D0'
1	1	0	1	0 → D0'
1	1	1	0	0 → 0
1	1	1	1	0 → 0

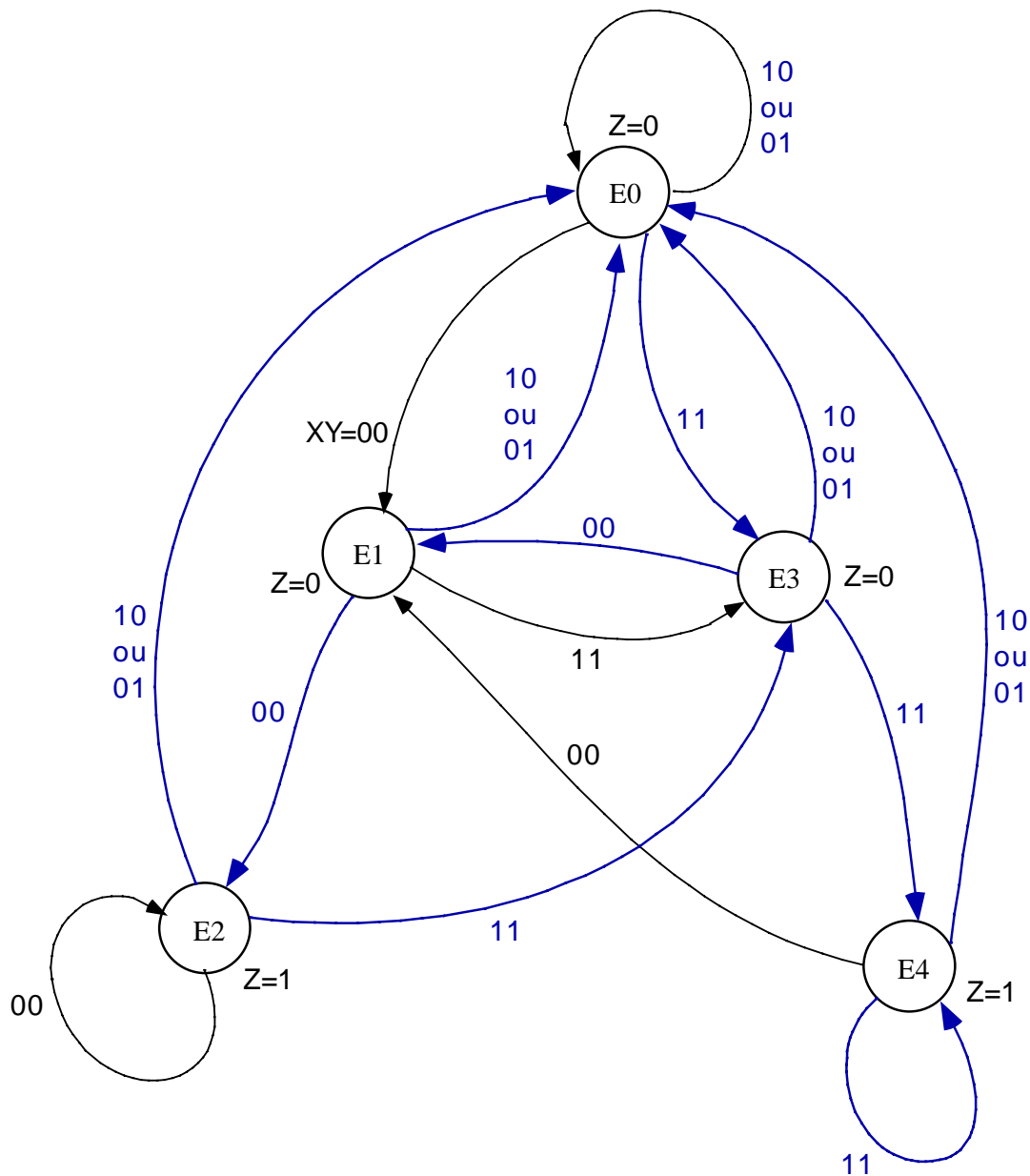


NOME: _____ CORRECÇÃO _____ TURMA _____

3 – Considere uma máquina de estados com duas entradas X,Y e uma saída Z. A saída Z deve tomar o valor lógico 1 sempre que, em duas transições de relógio consecutivas, as entradas X e Y mantenham o mesmo valor e sejam iguais entre si (ver figura).

Entrada X: 0 1 0 1 1 0 0 0 0 1 1 0 1
 Entrada Y: 0 0 0 1 1 1 1 0 0 1 1 1 1
 Saída Z: 0 0 0 0 1 0 0 0 1 1 0 1 0 0

Complete o diagrama de estados apresentado na figura que descreve o funcionamento desta máquina de estados. As condições de transições de estado apresentadas representam os valores das entradas X e Y, nesta ordem.



NOME: _____ CORRECÇÃO _____ TURMA _____

4 — Pretende-se projectar um circuito sequencial síncrono que produza nas saídas $Q_3Q_2Q_1Q_0$ a seguinte sequência:

0, 1, 3, 6, 10, 15, 0, 1, ...

- a) Construa a tabela de transição de estados, admitindo um critério de custo mínimo para os estados não especificados, e fazendo coincidir as saídas $Q_3Q_2Q_1Q_0$ com as variáveis de estado.

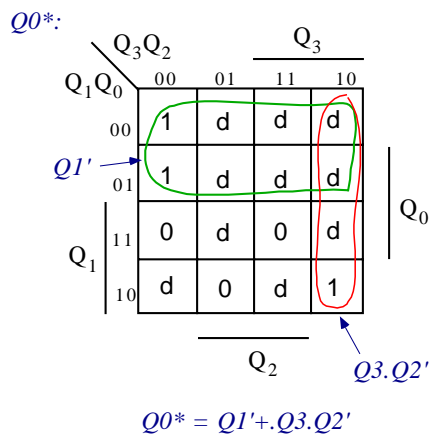
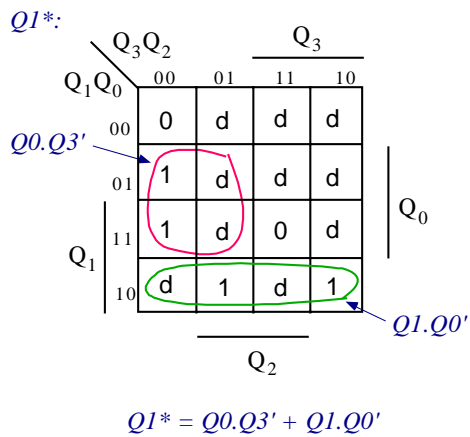
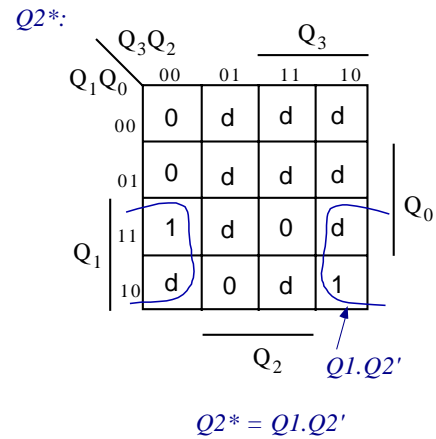
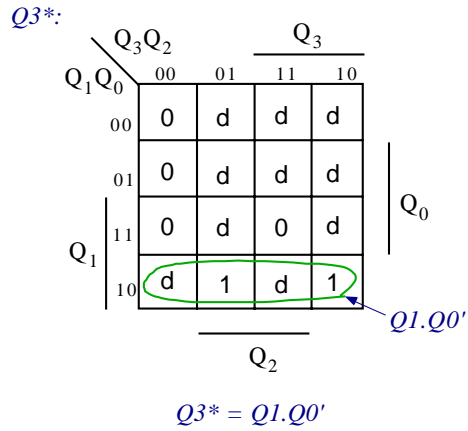
Vamos começar por construir a tabela de transição de estados do circuito pretendido. Note que esta máquina de estados não tem entradas e repete ciclicamente uma sequência de estados determinada. Como é pedido no enunciado, vamos fazer coincidir as saídas da máquina de estados com as variáveis $Q_3Q_2Q_1Q_0$ que representam o estado presente (na realidade seriam suficientes 3 flip-flops para representar o estado presente, já que só temos 6 estados; no entanto isso obrigaria a construir circuitos que produzissem as saídas pretendidas $Q_3Q_2Q_1Q_0$ em função do estado presente codificado em 3 bits).

Estado presente					Próximo estado				
	Q_3	Q_2	Q_1	Q_0	Q_3^*	Q_2^*	Q_1^*	Q_0^*	
0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1	1	3
	0	0	1	0	d	d	d	d	
3	0	0	1	1	0	1	1	0	6
	0	1	0	0	d	d	d	d	
	0	1	0	1	d	d	d	d	
6	0	1	1	0	1	0	1	0	10
	0	1	1	1	d	d	d	d	
	1	0	0	0	d	d	d	d	
	1	0	0	1	d	d	d	d	
10	1	0	1	0	1	1	1	1	15
	1	0	1	1	d	d	d	d	
	1	1	0	0	d	d	d	d	
	1	1	0	1	d	d	d	d	
	1	1	1	0	d	d	d	d	
15	1	1	1	1	0	0	0	0	0

NOME: _____ CORRECÇÃO _____ TURMA _____

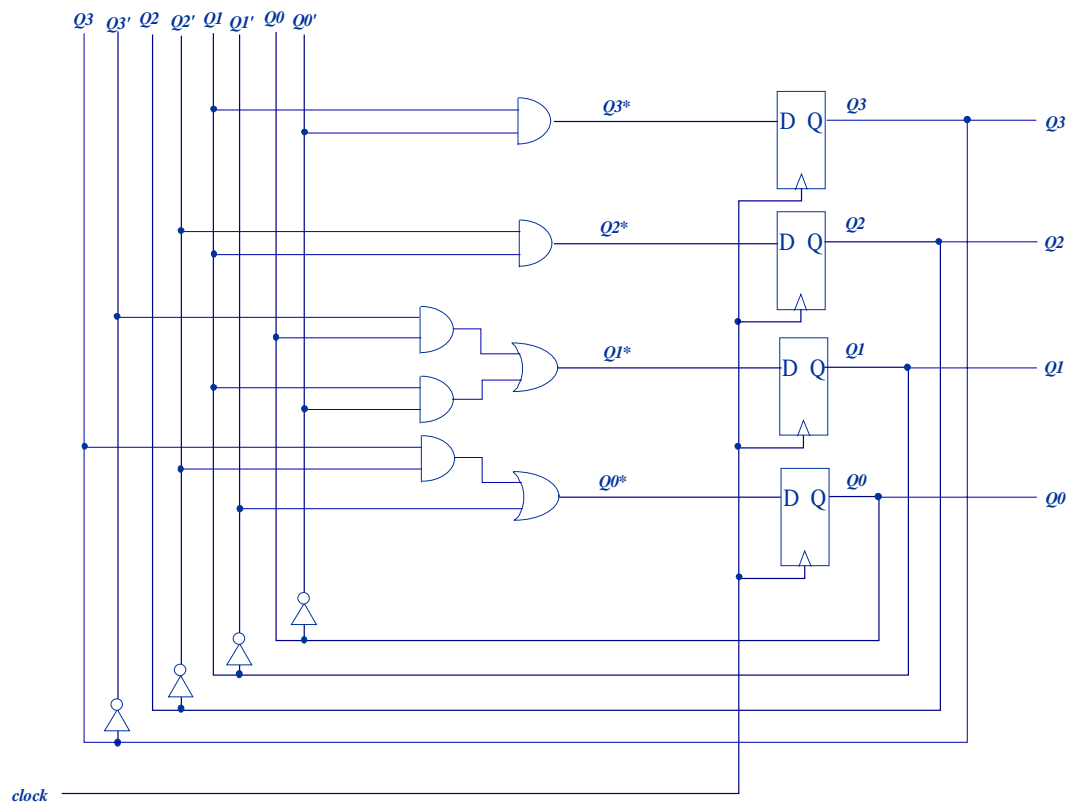
b) Projecte o circuito utilizando *flip-flops* D e circuitos lógicos adicionais (se necessário utilize o verso da folha).

Vamos obter funções minimizadas para as funções que produzem o próximo estado: $Q3^$, $Q2^*$, $Q1^*$ e $Q0^*$*

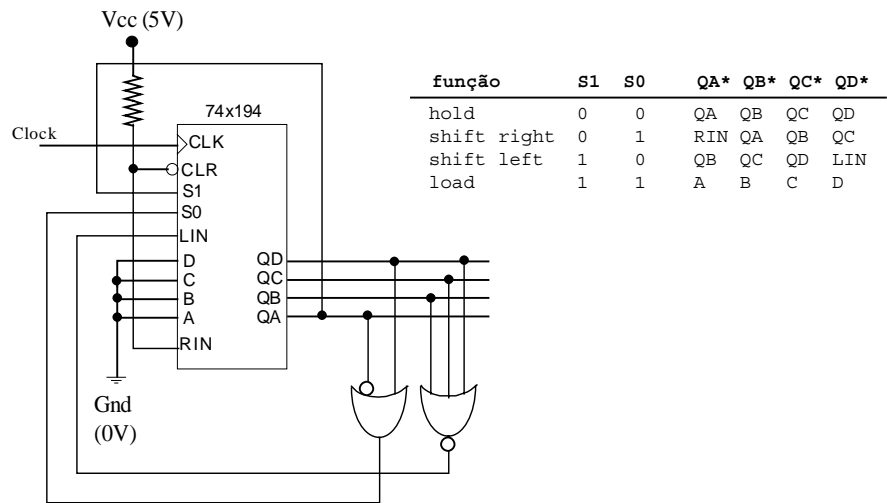


NOME: _____ CORRECÇÃO _____ TURMA _____

O circuito que realiza a máquina de estados pretendida é:



5 – Considere o seguinte circuito síncrono baseado num *shift-register* 74x194 (ver tabela):



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 que o estado inicial é $Q_A Q_B Q_C Q_D = 0000$, determine a sequência de valores nas saídas do circuito.

NOME: _____ CORRECÇÃO _____ TURMA _____

Q_A	Q_B	Q_C	Q_D	S_1	S_0	LIN	RIN	
0	0	0	0	0	1	1	1	← shift-right
1	0	0	0	1	0	1	1	← shift-left
0	0	0	1	0	1	0	1	← shift-right
1	0	0	0	1	0	1	1	← shift-left
...								

$$S1 = Q_A$$

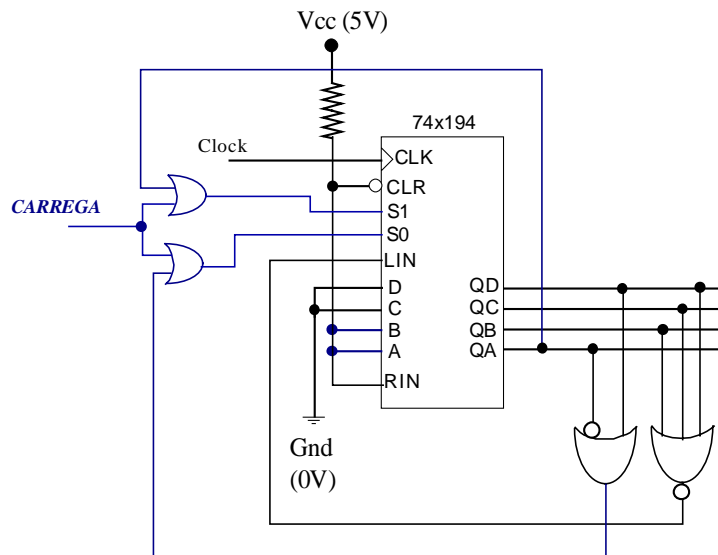
$$S0 = Q_A' + Q_D$$

$$LIN = (Q_D + Q_C + Q_B)'$$

$$RIN = 1$$

- b) Acrescente uma entrada CARREGA que, quando activada com 1, inicialize o shift-register com $Q_A Q_B Q_C Q_D = 1100$. Quando CARREGA=0 deve manter-se o comportamento do circuito inicial (utilize o verso da folha).

Para forçar o carregamento do shift-register temos de colocar com 1 as entradas S1 e S0 quando a entrada CARREGA tiver o nível lógico 1, e manter o funcionamento do circuito anterior quando CARREGA=0. Além disso é necessário ligar as entradas A e B ao nível lógico 1 para que o valor carregado seja 1100:



6 — Pretende-se construir um circuito síncrono baseado em dois contadores 74x163, que apresente, nas duas saídas de 4 bits, respectivamente o dígito das unidades e o dígito das dezenas da sequência de valores 00, 01, 02... 58, 59, 00... .

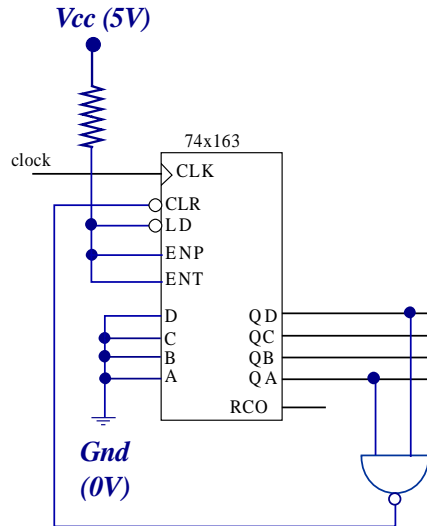
- a) Utilizando um contador 74x163, construa o circuito síncrono que produz o dígito das unidades (conta ciclicamente de 0 a 9).

Para construir um contador decimal (conta de 0 a 9) basta ligar à entrada CLR do contador uma saída que apresente o valor lógico 0 sempre que as saídas do contador estejam no estado 1001 (valor 9):

NOME: _____

CORRECÇÃO _____

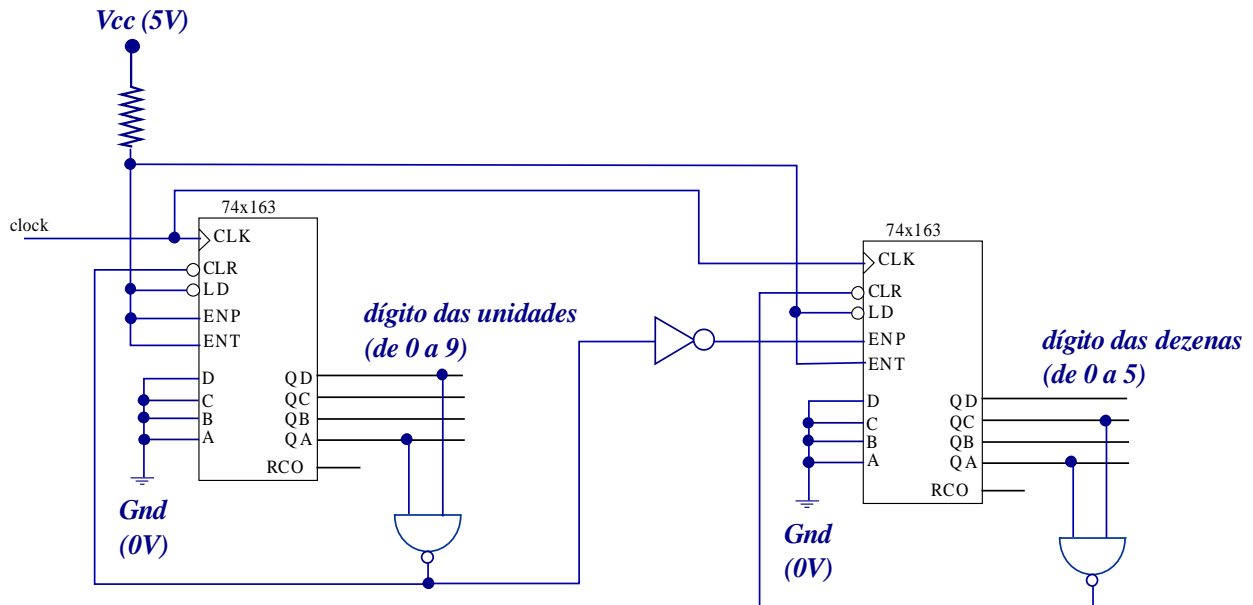
TURMA _____



74x163					estado presente				próximo estado			
/CLR	/LD	ENT	ENP		QD	QC	QB	QA	QD*	QC*	QB*	QA*
0	x	x	x		x	x	x	x	0	0	0	0
1	0	x	x		x	x	x	x	D	C	B	A
1	1	0	x		x	x	x	x	QD	QC	QB	QA
1	1	x	0		x	x	x	x	QD	QC	QB	QA
1	1	1	1		N (se N<15)				N + 1			
1	1	1	1		1	1	1	1	0	0	0	0

- b) Acrescentando o segundo 74x163, complete o circuito anterior por forma a realizar o contador pretendido (contador decimal de 00 a 59).

O circuito pretendido pode ser construído ligando o contador anterior que produz o dígito das unidades a outro contador com uma configuração semelhante que conte de 0 a 5 (dígito das dezenas). Este segundo contador apenas deve contar quando as unidades atingem o valor 9. Para isso podemos ligar o sinal que activa o CLR do primeiro contador à entrada ENP (enable) do segundo contador (note que este sinal tem que ser negado já que ENP é activo com o nível lógico 1 e CLR é activo com o nível lógico 0).



- FIM -