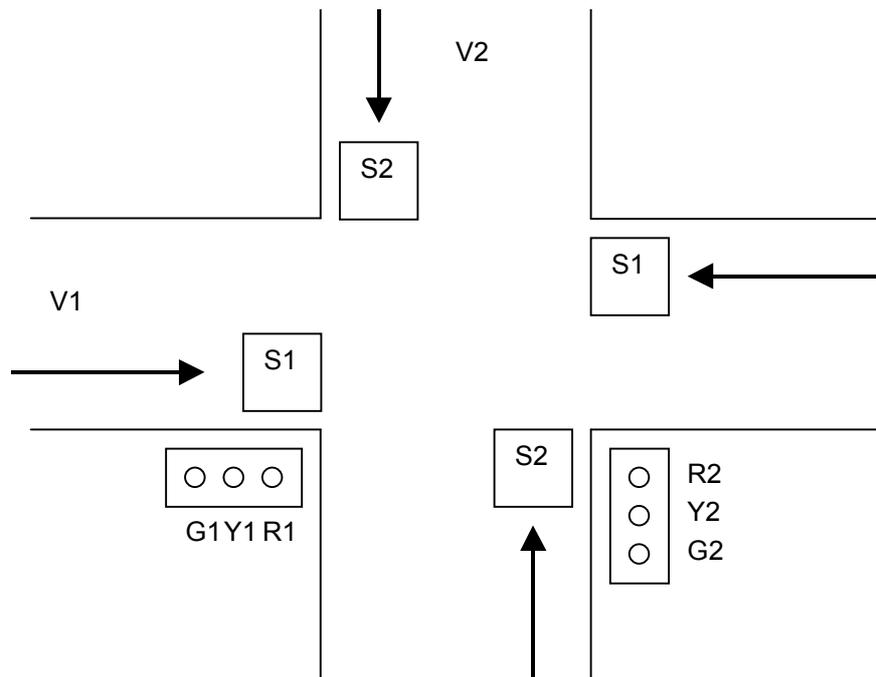


Trabalho Prático Nº 8

Projecto de um controlador de semáforos



Especificações funcionais:

- ◆ Controlador de semáforos de um cruzamento entre duas vias, V1 e V2, cada uma das quais equipadas com sensores, S1 e S2 que detectam a presença de veículos.
- ◆ O tempo de passagem (sinal verde) deve ser uniformemente distribuído quando, simultaneamente, existir ou não tráfego nas duas vias. Nestas situações o circuito segue uma sequência normal controlada apenas pelo sinal de relógio (funcionamento de um semáforo comum).
- ◆ Quando é detectada a presença de um veículo em apenas uma das vias, o estado do sistema deve ser alterado para permitir a passagem desse veículo.
- ◆ Sempre que existir tráfego em apenas uma das vias o sinal verde dessa via deve manter-se.

Variáveis de entrada:

S1, S2 – Entradas correspondentes aos sensores que detectam a presença de veículos nas vias V1 e V2 respectivamente (entradas activas a 1).

Variáveis de saída:

G1, Y1, R1 – Sinais verde, laranja e vermelho, respectivamente, para a via V1 (saídas activas a 1).

G2, Y2, R2 – Sinais verde, laranja e vermelho, respectivamente, para a via V2 (saídas activas a 1).

Diagrama de estados

De acordo com as especificações funcionais, obtém-se o diagrama de estado da figura. Ao contrário do que é habitual, cada transição é acompanhada da expressão lógica (equação de transição) que condiciona a transição para o estado seguinte. Por exemplo:

- do estado E0, mantém-se o mesmo estado se $S1=1$ e $S2=0$ (eq. transição = $S1 S2'$), transita-se para o estado E1 se $S1=S2=0$ ou $S1=S2=1$ (eq. transição = $(S1 \oplus S2)'$), transita-se para o estado E3 se $S1=0$ e $S2=1$ (eq. transição = $S1' S2$);
- do estado E1, transita-se para o estado E2 quaisquer que sejam os valores de S1 e S2.

Na representação do estado, apenas se faz a referência às variáveis de saída que se encontram activas. Por exemplo, no estado E0, estão activas as saídas G1 e R2.

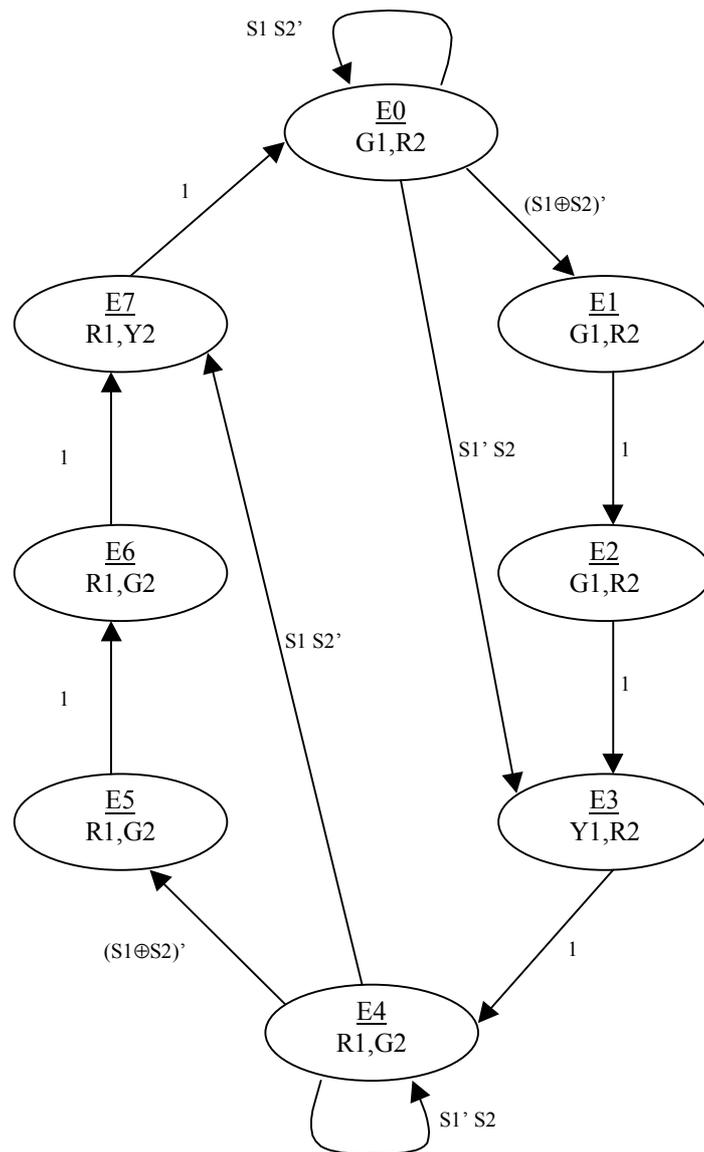


Tabela e codificação de estados

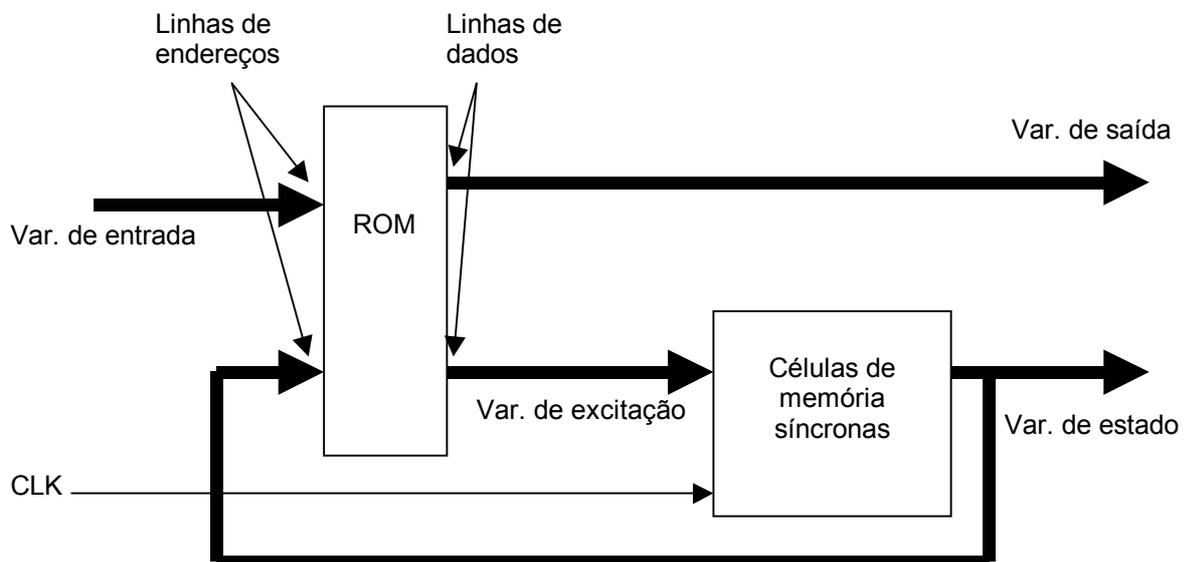
Para codificar 8 estados (E0, ..., E7) são necessários 3 bits (Q_C, Q_B, Q_A). Dado o tipo de implementação a utilizar, a codificação dos estados não obedece a nenhuma regra específica, tendo-se utilizado a seguinte codificação: E0=000; E1=001; ...; E7=111.

Estado presente (Q_C, Q_B, Q_A)	S1	S2	Var. excitação/ Estado seguinte (D_C, D_B, D_A) /(Q_C, Q_B, Q_A)	G1	Y1	R1	G2	Y2	R2
E0(000)	0 0 1 1	0 1 0 1	E1(001) E3(011) E0(000) E1(001)	1	0	0	0	0	1
E1(001)	X	X	E2(010)	1	0	0	0	0	1
E2(010)	X	X	E3(011)	1	0	0	0	0	1
E3(011)	X	X	E4(100)	0	1	0	0	0	1
E4(100)	0 0 1 1	0 1 0 1	E5(101) E4(100) E7(111) E5(101)	0	0	1	1	0	0
E5(101)	X	X	E6(110)	0	0	1	1	0	0
E6(110)	X	X	E7(111)	0	0	1	1	0	0
E7(111)	X	X	E0(000)	0	0	1	0	1	0

Implementação

Para a implementação deste circuito serão utilizados flip-flops D para o módulo de células de memória e uma memória EPROM 2764 (8Kx8bits) para o descodificador de estado seguinte e descodificador de saída.

A utilização de uma ROM na implementação da lógica combinacional dos circuitos sequenciais síncronos segue a estrutura geral da figura.



As linhas de endereços são alimentadas com as variáveis de estado e as variáveis de entrada. As linhas de dados correspondem às variáveis de excitação (entradas do módulo de células de memória) e às variáveis de saída.

A EPROM 2764 disponibiliza 8K x 8 bits (8KBytes), possuindo 13 linhas de endereço (A12..A0) ($2^{13}=8K$) e 8 linhas de dados (Q7..Q0). O circuito a implementar possui 3 variáveis de estado (Q_C, Q_B, Q_A) e duas variáveis de entrada (S1, S2), ou seja, 5 variáveis no total. Assim sendo, serão apenas utilizados os primeiros $2^5=32$ endereços de memória. As linhas de endereço A12..A5 serão ligadas a 0, efectuando-se a associação das restantes linhas de endereço de acordo com a tabela:

Linhas de endereço	A12..A5	A4	A3	A2	A1	A0
Var. de estado/entrada	0..0	Q_C	Q_B	Q_A	S1	S2

Relativamente às variáveis de saída e variáveis de excitação, o circuito possui 6 variáveis de saída (G1, Y1, R1, G2, Y2 e R2) e 3 variáveis de excitação (entrada dos flip-flops= D_C, D_B, D_A), num total de 9 variáveis. Tendo a

EPROM 2764 apenas 8 bits de dados, uma das variáveis de saída/excitação deveria ser implementada externamente à EPROM. No entanto, neste circuito, esta tarefa encontra-se simplificada, dado que a saída R1 coincide com a variável de estado Q_C , conforme se pode verificar na tabela de estados.

A tabela seguinte define uma solução possível para a associação entre as linhas de dados da EPROM e as variáveis de excitação/saída:

Linhas de dados	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
Var. excitação/saída	D_C	D_B	D_A	G1	Y1	G2	Y2	R2

Gravação do conteúdo da EPROM

A gravação/programação do conteúdo da EPROM será realizada através de um programador universal disponível no laboratório. A introdução dos dados neste dispositivo é, normalmente, realizada através de valores hexadecimais, quer para os endereços, quer para o conteúdo da respectiva célula de memória. Assim sendo, sugere-se o preenchimento da tabela de estados apresentada em anexo, onde se define o endereço e respectivo conteúdo da célula de memória.

Ligação da EPROM

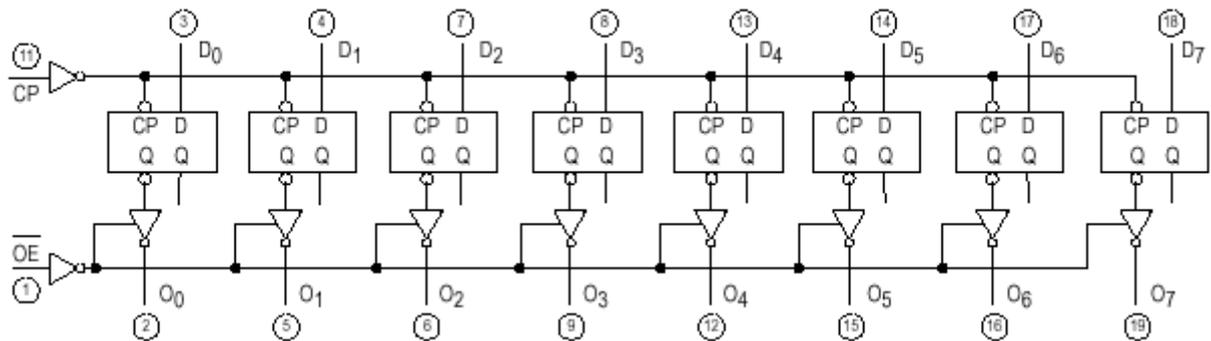
Depois de programada, a EPROM é colocada no circuito em modo de leitura. De acordo com as especificações apresentadas em anexo, as entradas $CE\#$ e $OE\#$ são colocadas a V_{IL} (0V) e as entradas V_{PP} e $PGM\#$ são colocadas a V_{IH} (+5V).

Trabalho a executar

- Preencher a tabela de estados em anexo por forma a definir o conteúdo das primeiras 32 posições de memória da EPROM;
- Proceder à gravação da EPROM;
- Construir o diagrama lógico, utilizando, para as células de memória, os flip-flops D do registo de dados disponível no IC 74LS374 (8 flip-flops D com saídas em três estados);
- Implementar e testar o circuito, utilizando os comutadores sem ruído para simular os sensores S1 e S2 e LED's em configuração de cátodo comum para visualizar as saídas.

74LS374

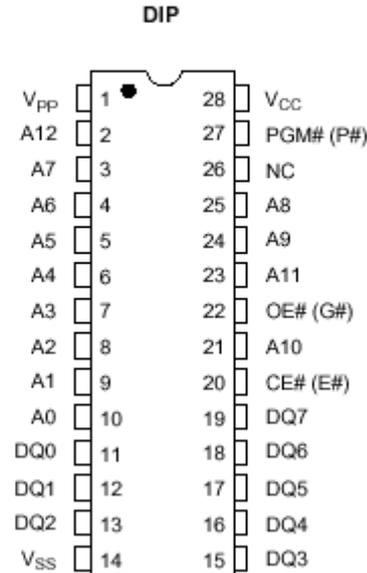
Octal D flip-flop with tri-state output

SN54LS/74LS374**LS374**

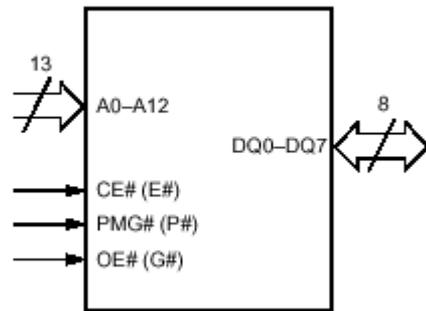
D_n	CP	OE	O_n
H		L	H
L		L	L
X	X	H	Z*

EPROM 2764

8 K x 8bits EPROM

CONNECTION DIAGRAMS**Top View****PIN DESIGNATIONS**

A0–A12	=	Address Inputs
CE# (E#)	=	Chip Enable Input
DQ0–DQ7	=	Data Input/Outputs
OE# (G#)	=	Output Enable Input
PGM# (P#)	=	Program Enable Input
V _{CC}	=	V _{CC} Supply Voltage
V _{PP}	=	Program Voltage Input
V _{SS}	=	Ground
NC	=	No Internal Connection

LOGIC SYMBOL

11419E-4

MODE SELECT TABLE

Mode	CE#	OE#	PGM#	A0	A9	V _{pp}	Outputs	
Read	V _{IL}	V _{IL}	X	X	X	X	D _{OUT}	
Output Disable	X	V _{IH}	X	X	X	X	High Z	
Standby (TTL)	V _{IH}	X	X	X	X	X	High Z	
Standby (CMOS)	V _{CC} ± 0.3 V	X	X	X	X	X	High Z	
Program	V _{IL}	X	V _{IL}	X	X	V _{pp}	D _{IN}	
Program Verify	V _{IL}	V _{IL}	V _{IH}	X	X	V _{pp}	D _{OUT}	
Program Inhibit	V _{IH}	X	X	X	X	V _{pp}	High Z	
Autoselect (Note 3)	Manufacturer Code	V _{IL}	V _{IL}	X	V _{IL}	V _H	X	01h
	Device Code	V _{IL}	V _{IL}	X	V _{IH}	V _H	X	15h

Notes:

1. V_H = 12.0 V ± 0.5 V.
2. X = Either V_{IH} or V_{IL}.
3. A1–A8 and A10–12 = V_{IL}.
4. See DC Programming Characteristics for V_{pp} voltage during programming.