

Sistemas Digitais I
LESI :: 2º ano

Questões Práticas de Sistemas Sequenciais

António Joaquim Esteves

João Miguel Fernandes

www.di.uminho.pt/~aje

Bibliografia: secções 8.3.1, 8.4 (parte), 8.5 (parte), DDPP, Wakerly



DEP. DE INFORMÁTICA
ESCOLA DE ENGENHARIA
UNIVERSIDADE DO MINHO

7. Questões Práticas de Sistemas Sequenciais

- Sumário -

- Introdução
- Contadores
- Registos de deslocamento (*shift registers*)
- Contadores baseados em registos de deslocamento
- PLDs sequenciais

7. Questões Práticas de Sistemas Sequenciais

- *Introdução (1)* -

- ❑ A maior parte dos sistemas digitais reais é do tipo sequencial.
- ❑ O comportamento sequencial resulta de ciclos de realimentação, de *latches* ou de *flip-flops* que fazem o estado actual do sistema depender do historial das entradas.
- ❑ Para que o número de estados de alguns sistemas digitais reais (como um computador) não atinja um valor extremamente elevado, é habitual decompô-los em partes, como por exemplo: caminhos de dados (*data paths*), bancos de registos e unidades de controlo.
- ❑ Um sistema típico possui várias unidades funcionais com uma interface bem conhecida para elas poderem ser interligadas.
- ❑ Por sua vez, uma unidade funcional pode também ser construída à custa de blocos mais simples passíveis de ser tratados de forma hierárquica e em níveis de abstracção diferentes.

7. Questões Práticas de Sistemas Sequenciais

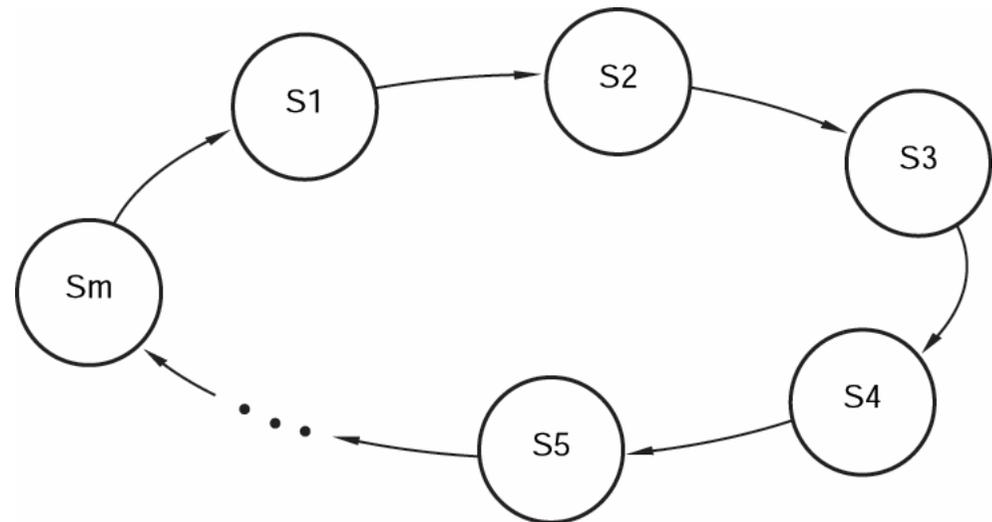
- *Introdução (2)* -

- ❑ Apesar de os sistemas digitais poderem ser muito complexos, a maior parte deles é contruído em torno de uma máquina de estados.
- ❑ Durante a década de 80, o processo utilizado para projectar uma máquina de estados era o que usou no 1º exemplo do módulo anterior. Ou seja, escrevia-se a tabela de estados e saídas que após algumas transformações gerava a tabela de verdade para cada saída combinacional. A síntese manual da máquina terminava com a simplificação das expressões das saídas através de mapas de karnaugh (ou de outro método qualquer).
- ❑ Actualmente, a maior parte dos projectistas descreve a tabela de estados e saídas das máquinas de estados numa HDL (como ABEL, VHDL ou Verilog).
- ❑ Uma ferramenta de síntese para essa HDL processa a descrição da máquina de modo a efectuar uma síntese equivalente à efectuada manualmente e obtém uma implementação numa determinada tecnologia (PLD, CPLD, FPGA ou ASIC) escolhida, desde que seja suportada pela ferramenta.

7. Questões Práticas de Sistemas Sequenciais

- Contadores (1) -

- ❑ Chama-se contador a um circuito sequencial síncrono que é descrito por um diagrama de estados que possui apenas um ciclo.
- ❑ O *modulus* do contador coincide com o número de estados desse ciclo.

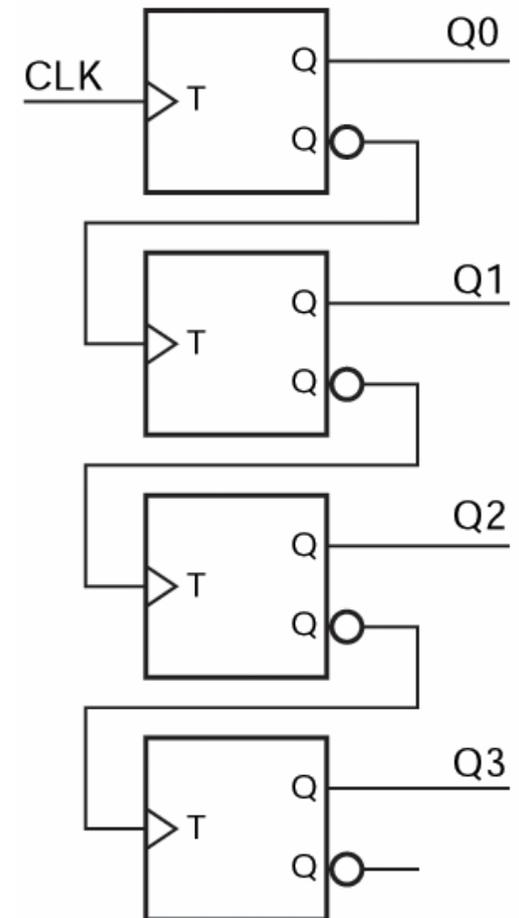


- ❑ A um contador com m estados chama-se um contador módulo-m.
- ❑ Um contador com um modulus que não é potência de 2 possui alguns estados que não são visitados no modo normal de funcionamento.

7. Questões Práticas de Sistemas Sequenciais

- Contadores (2) -

- ❑ O tipo de contador mais comum é o contador binário de n-bits.
- ❑ Este contador é implementado com n *flip-flops* e possui 2^n estados, os quais são visitados seguindo a sequência 0, 1, 2, ..., 2^n-1 , 0, 1, ...
- ❑ Cada estado é codificado usando o valor binário de n-bits que correspondente à posição desse estado na sequência 0, 1, 2, ..., 2^n-1 . **Exemplo:** se um estado estiver na posição $0 \leq p \leq 2^n-1$, o seu código é o valor binário que representa p em n-bits.
- ❑ Um contador binário de n-bits pode ser construído apenas com n *flip-flops* tipo T, sem qualquer tipo de lógica combinacional. Numa implementação deste tipo, cada bit Q_i do contador muda quando o bit Q_{i-1} imediatamente antes (e menos significativo) mudar de 1 para 0.
- ❑ O contador é designado por contador de ripple porque a informação de transporte é propagada desde o LSB até ao MSB.



7. Questões Práticas de Sistemas Sequenciais

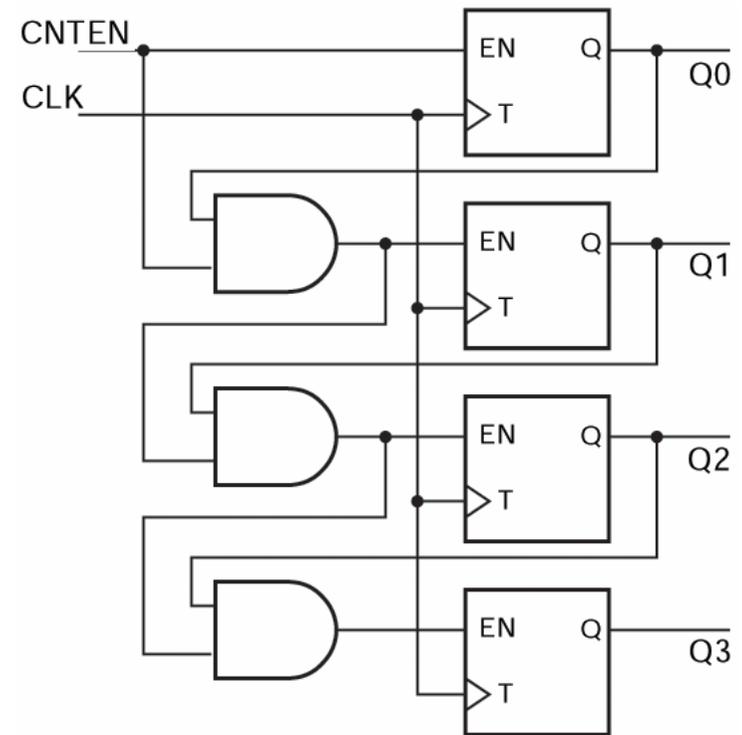
- Contadores (3) -

- ❑ Apesar de exigir menos lógica do que qualquer outro tipo de contador, o contador de *ripple* também é mais lento.
- ❑ O pior cenário, que ocorre quando o MSB tiver que mudar, a saída só é válida passados $n \cdot t_{TQ}$ sobre a transição ascendente do relógio.
(em que t_{TQ} é o tempo de atraso que a entrada demora a afectar a saída dum *flip-flop* tipo T).
- ❑ Num contador síncrono, o sinal de relógio de todos os *flip-flops* está ligado a um sinal **CLK** comum.
- ❑ Deste modo, a saída de todos os *flip-flops* muda de valor no mesmo instante, passado um atraso t_{TQ} após a transição ascendente de **CLK**.

7. Questões Práticas de Sistemas Sequenciais

- Contadores (4) -

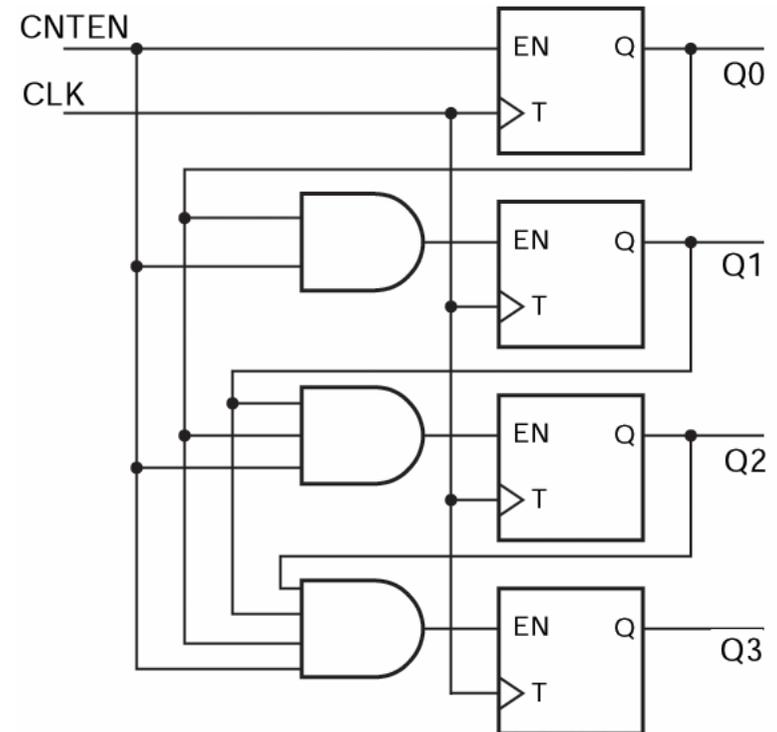
- ❑ Um contador síncrono de ripple exige que se use *flip-flops* T com *enable*.
- ❑ Assim, a saída Q muda de valor na transição ascendente da entrada T mas apenas se a entrada EN estiver activa.
- ❑ A saída Q de cada *flip-flop* T muda de valor se CNTEN estiver activa e todas as saídas Q de ordem inferior forem 1.
- ❑ Tal como o contador de *ripple*, o contador síncrono também pode ser construído com uma quantidade de lógica por bit fixa. No exemplo ao lado, um *flip-flop* T com *enable* e uma porta AND de 2 entradas.
- ❑ A este tipo de contador também se chama contador série síncrono porque o sinal de *enable* se propaga em série desde o LSB até ao MSB.



7. Questões Práticas de Sistemas Sequenciais

- Contadores (5) -

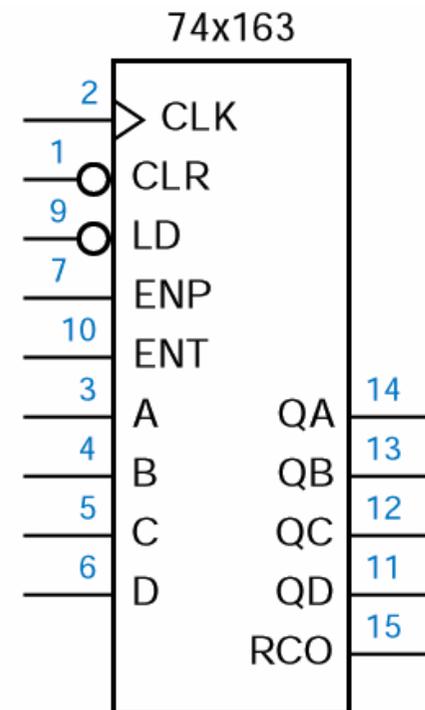
- ❑ Se o período do relógio for demasiado reduzido, poderá não dar tempo para que o sinal de *enable* se propague desde o LSB até ao MSB.
- ❑ Este problema é eliminado se cada entrada EN for gerada por uma porta AND própria sem recurso aos EN's de ordem inferior. →
- ❑ A alteração introduzida resulta no tipo de contador com estrutura mais rápida: o contador síncrono paralelo.



7. Questões Práticas de Sistemas Sequenciais

- Contadores (6) -

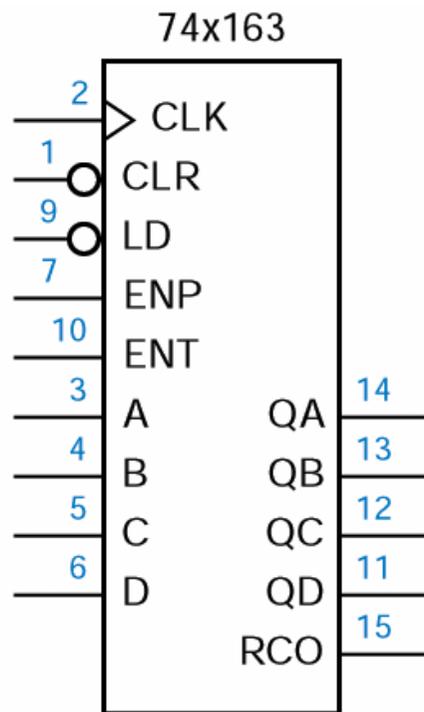
- ❑ O CI 74x163 contém um contador síncrono binário de 4-bits.
- ❑ O contador é construído com *flip-flops* tipo D.
- ❑ A entrada síncrona LD (*load*) permite colocar o valor das entradas D-C-B-A nas saídas Q.
- ❑ A entrada síncrona CLR (*clear*) permite fazer o *reset* das saídas Q.
- ❑ A saída RCO (*ripple carry do MSB*) é 1, durante um ciclo de relógio, quando todas as saídas Q e a entrada ENT estiverem a 1.



7. Questões Práticas de Sistemas Sequenciais

- Contadores (7) -

- Tabela de estados do contador síncrono binário de 4-bits 74x163.



Inputs				Current State				Next State			
CLR_L	LD_L	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	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	0	0	1	0	0	0	1	1
1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	1	0	1	0
1	1	1	1	1	1	0	0	1	1	0	1
1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0

7. Questões Práticas de Sistemas Sequenciais

- Contadores (8) -

- ❑ Descrição comportamental do contador síncrono binário de 4-bits 74x163.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity V74x163 is
    port ( CLK, CLR_L, LD_L, ENP, ENT: in STD_LOGIC;
          D: in UNSIGNED (3 downto 0);
          Q: out UNSIGNED (3 downto 0);
          RCO: out STD_LOGIC );
end V74x163;
```

Define '+' e '-' para o tipo de dados sem sinal → UNSIGNED

```
architecture V74x163_arch of V74x163 is
    signal IQ: UNSIGNED (3 downto 0);
begin
    process (CLK, ENT, IQ)
    begin
        if (CLK'event and CLK='1') then
            if CLR_L='0' then IQ <= (others => '0');
            elsif LD_L='0' then IQ <= D;
            elsif (ENT and ENP)='1' then IQ <= IQ + 1;
            end if;
        end if;
        if (IQ=15) and (ENT='1') then RCO <= '1';
        else RCO <= '0';
        end if;
        Q <= IQ;
    end process;
end V74x163_arch;
```

Define um vector de zeros com a largura do vector alvo IQ

Soma sem sinal

7. Questões Práticas de Sistemas Sequenciais

- Contadores (9) -

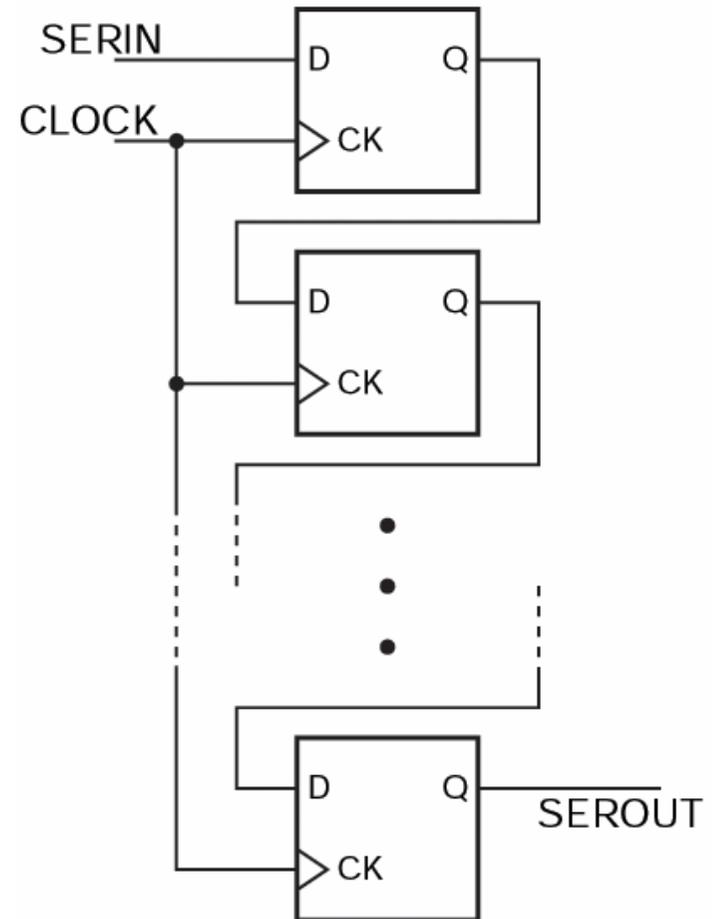
- Contador semelhante ao 74x163 mas que conta em excesso de 3 (3,4,...,11,12,3,...).

```
architecture V74xs3_arch of V74x163 is
signal IQ: UNSIGNED (3 downto 0);
begin
process (CLK, ENT, IQ)
begin
if CLK'event and CLK='1' then
if CLR_L='0' then IQ <= (others => '0');
elsif LD_L='0' then IQ <= D;
elsif (ENT and ENP)='1' and (IQ=12) then IQ <= ('0','0','1','1');
elsif (ENT and ENP)='1' then IQ <= IQ + 1;
end if;
end if;
if (IQ=12) and (ENT='1') then RCO <= '1';
else RCO <= '0';
end if;
Q <= IQ;
end process;
end V74xs3_arch;
```

7. Questões Práticas de Sistemas Sequenciais

- Registos de Deslocamento (1) -

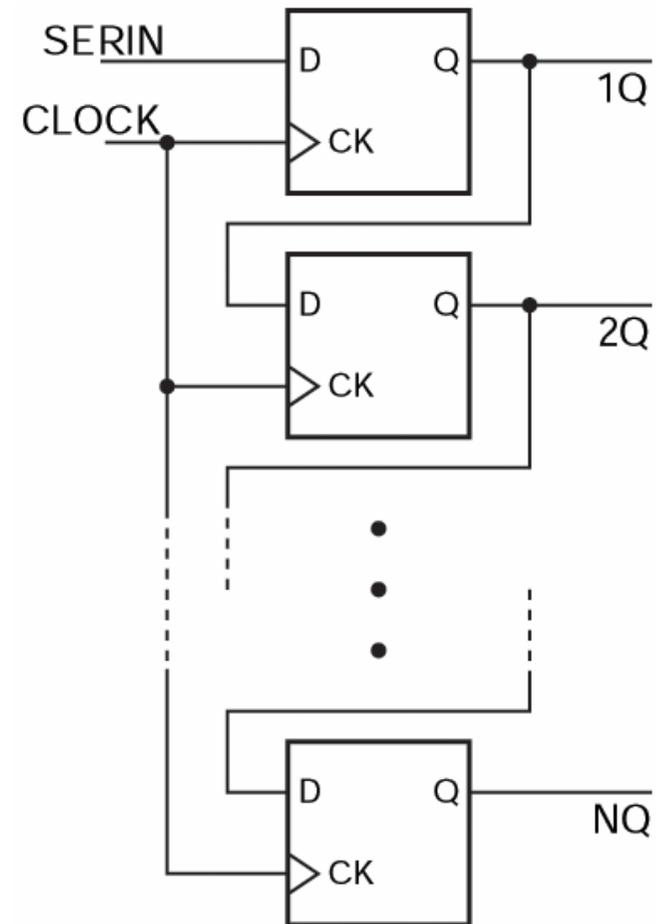
- ❑ Um registo de deslocamento é um registo de n-bits com a capacidade de deslocar em uma posição o valor armazenado, em cada ciclo de relógio.
- ❑ Um registo de deslocamento com entrada e saída série possui apenas uma entrada (SERIN) e uma saída (SEROUT).
- ❑ Na entrada SERIN aplica-se um novo bit a deslocar até à saída da cadeia de F/Fs, um F/F em cada ciclo do relógio.
- ❑ O bit aplicado na entrada surge na saída SEROUT passados n ciclos do relógio e sai do registo no ciclo seguinte (n+1).
- ❑ Um registo de deslocamento com entrada e saída série, e com n-bits, permite atrasar um sinal em n ciclos do relógio.



7. Questões Práticas de Sistemas Sequenciais

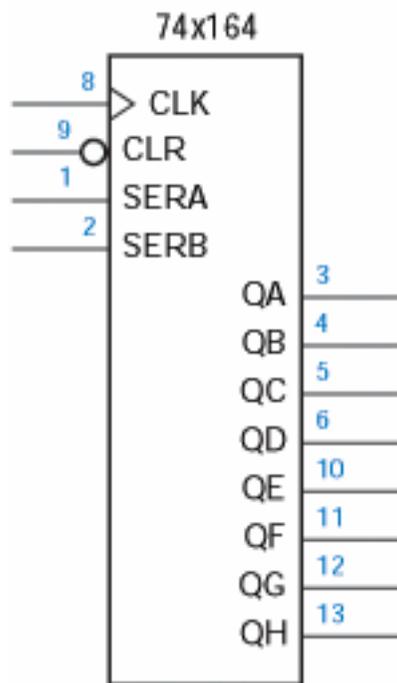
- Registos de Deslocamento (2) -

- ❑ Um registo de deslocamento com entrada série e saída paralela possui uma saída para cada bit armazenado, colocando-os disponíveis em 1Q a NQ.
- ❑ Os registos deste tipo são utilizados para efectuar conversões de série para paralelo.
- ❑ Pode ainda construir-se um registo de deslocamento com entrada paralela e saída série ou com entrada e saída paralela.
- ❑ O registo de deslocamento com entrada e saída paralela é útil na implementação das operações de SHIFT e ROTATE.

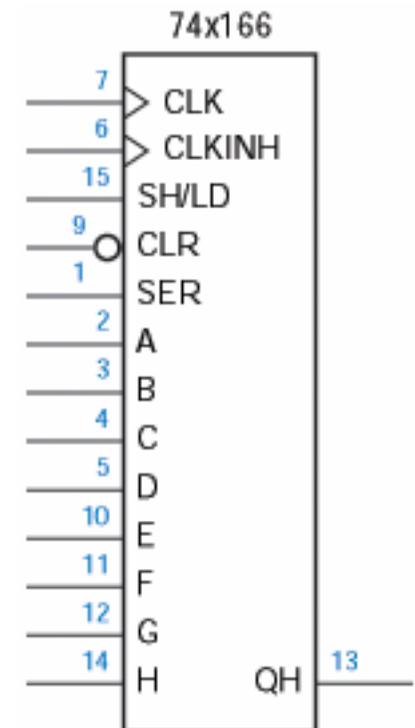


7. Questões Práticas de Sistemas Sequenciais

- Registos de Deslocamento (3) -



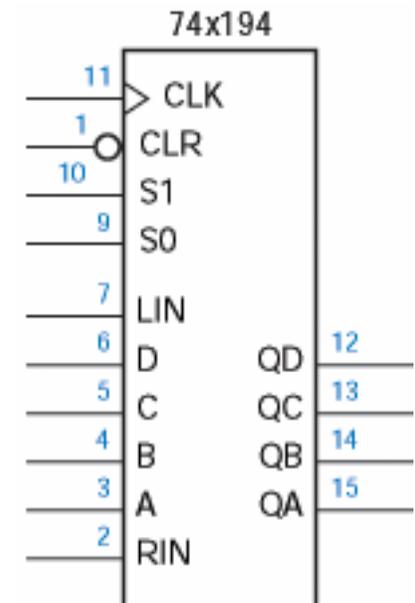
- ❑ O registo de deslocamento com 8-bits dos CIs 74x164 e 74x166 são unidireccionais dado que só efectuam o deslocamento dos bits numa direcção.
- ❑ O CI 74x164 é do tipo entrada série e saída paralela e possui uma entrada de *reset* CLR assíncrona.
- ❑ O CI 74x166 é do tipo entrada paralela e saída série, também com uma entrada de *reset* CLR assíncrona. O 74x166 regista a entrada quando SH/LD é 0 e desloca quando for 1.



7. Questões Práticas de Sistemas Sequenciais

- Registos de Deslocamento (4) -

- ❑ O CI MSI 74x194 é um registo de deslocamento com 4-bits e bi-direccional, do tipo entrada e saída paralela.
- ❑ O registo de deslocamento é bi-direccional porque desloca os bits armazenados em ambas as direcções (direita/esquerda), de acordo com as entradas de controlo S1 e S0.
- ❑ O CI 7x194 é considerado um registo de deslocamento universal, dado poder funcionar como qualquer um dos tipos de registo de deslocamento menos genérico apresentado.

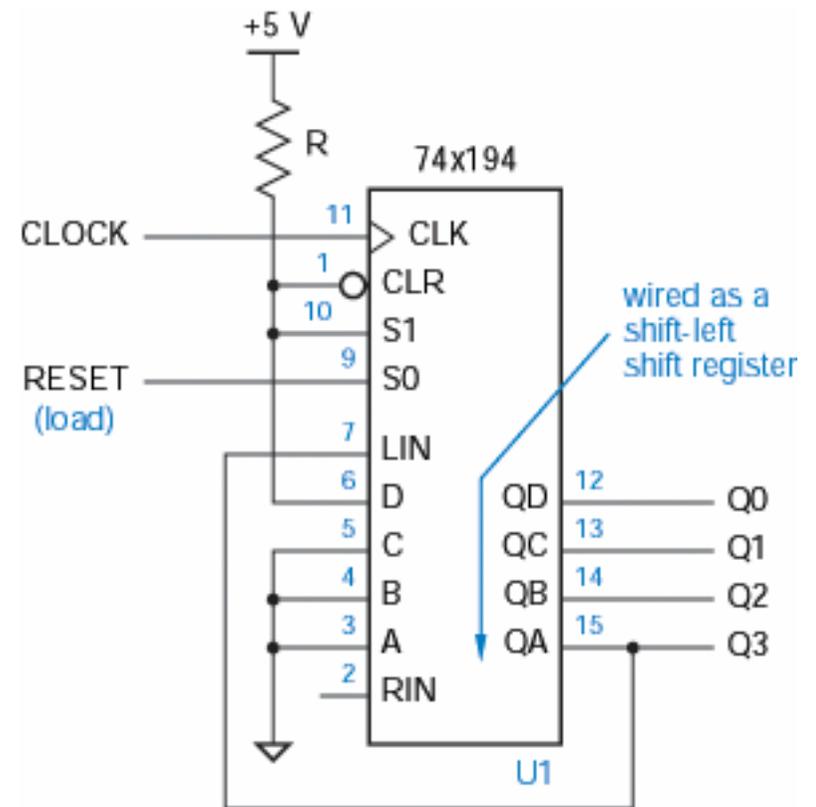


Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D

7. Questões Práticas de Sistemas Sequenciais

- Contadores baseados em Registos de Deslocamento (1) -

- ❑ Pode combinar-se um registo de deslocamento com lógica combinacional para dar origem a uma máquina de estados com um diagrama de estados cíclico → este tipo de circuito chama-se contador baseado num registo de deslocamento (*shift-register counter*)
- ❑ Um contador baseado num registo de deslocamento é diferente dum contador binário porque não conta por ordem (ascendente ou descendente) e é utilizado em aplicações de controlo.
- ❑ O contador baseado num registo de deslocamento mais simples de todos emprega um registo de deslocamento de n-bits para obter um contador com n estados, e dá pelo nome de contador em anel.

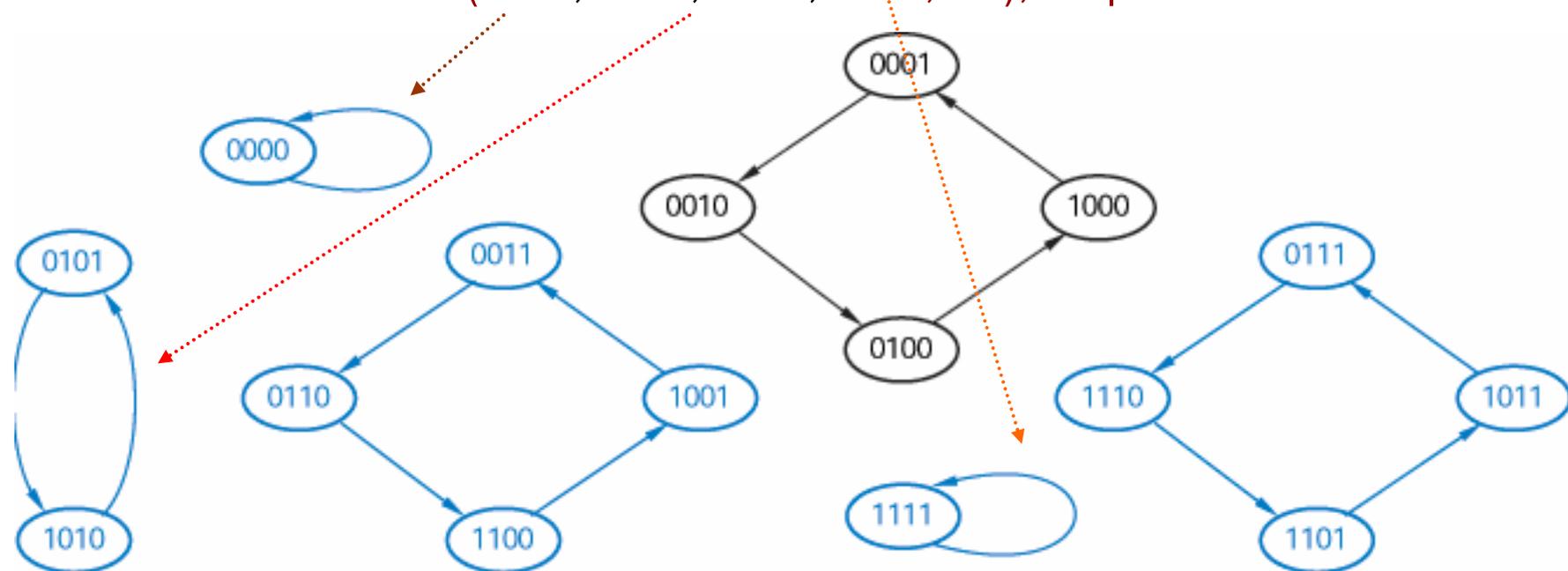


- ❑ Pode obter-se um contador em anel de 4-bits a partir dum CI 74x194 (reg. de deslocamento).

7. Questões Práticas de Sistemas Sequenciais

- Contadores baseados em Registos de Deslocamento (2) -

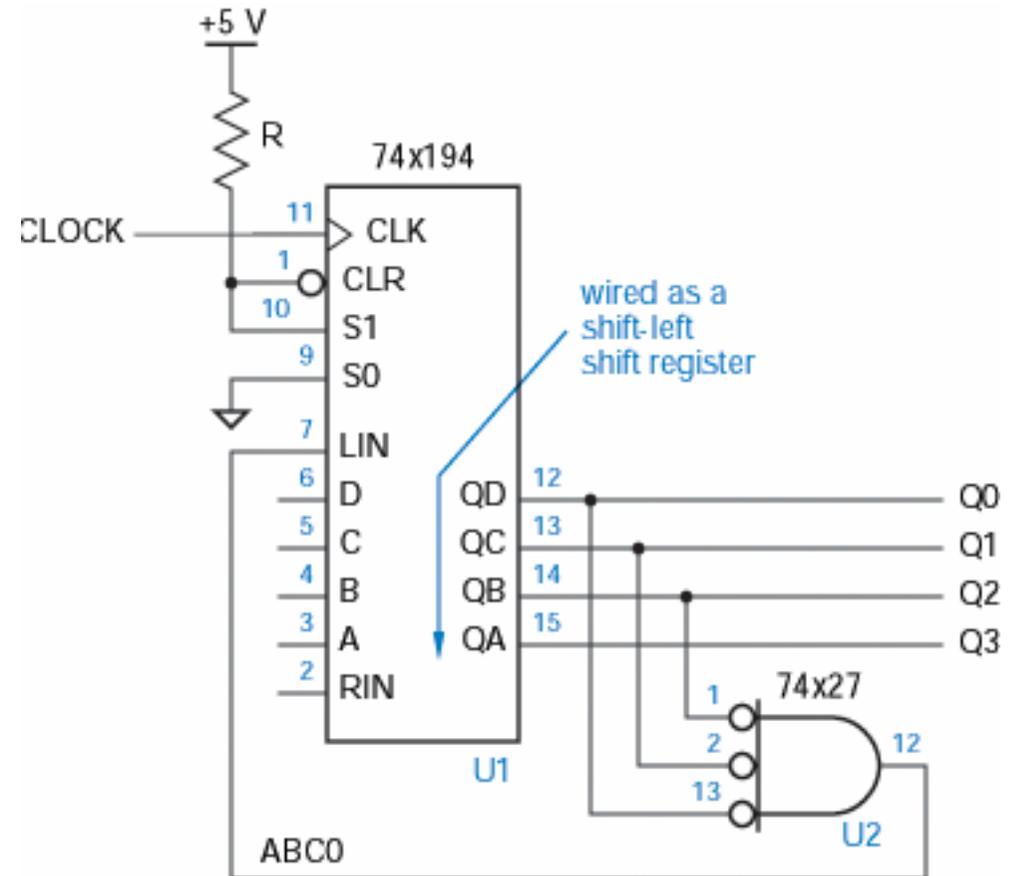
- ❑ Quando o **RESET** está a 1, o contador em anel coloca ABCD=0001 na saída Q3Q2Q1Q0 (estado inicial).
- ❑ No modo normal de funcionamento, com RESET=0, o contador efectua o deslocamento à esquerda do valor das saídas.
- ❑ O circuito não é robusto porque se o contador atingir um estado fora do ciclo normal de 4 estados (0000, 0101, 1010, 1111, etc), vai permanecer fora do ciclo.



7. Questões Práticas de Sistemas Sequenciais

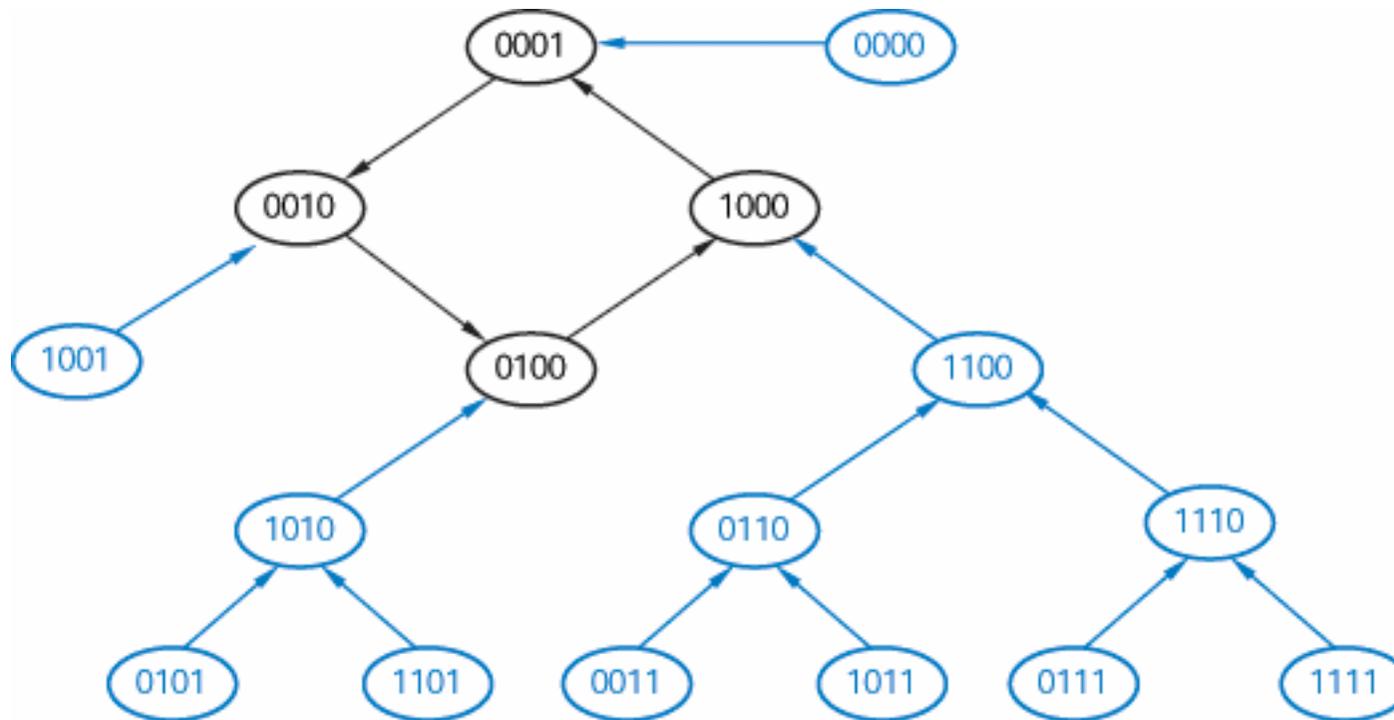
- Contadores baseados em Registos de Deslocamento (3) -

- ❑ Pode projectar-se um contador em anel com mecanismo de auto-correcção de modo a que todos os estados fora do ciclo normal possuam transições para estados dentro desse ciclo.
- ❑ Um exemplo é o contador de 4-bits que se obtém com um CI 74x194 e uma porta NOR. →
- ❑ A porta NOR é usada para aplicar um 1 em LIN, apenas quando os 3 LSBs (Q2Q1Q0) forem 0.



7. Questões Práticas de Sistemas Sequenciais

- Contadores baseados em Registos de Deslocamento (4) -

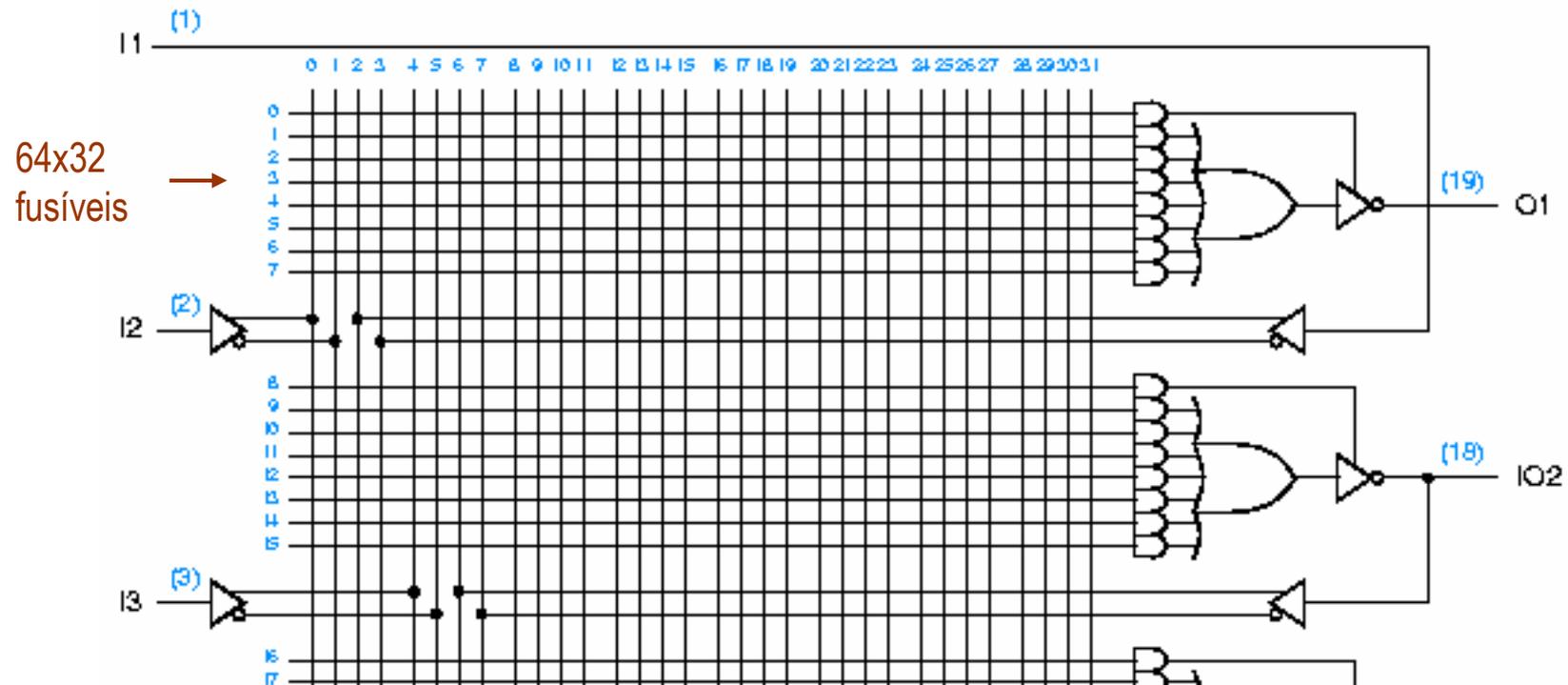


- ❑ Deste modo, todos os estados fora do ciclo normal conduzem a um estado dentro desse ciclo.
- ❑ Independentemente do estado anormal atingido, a correcção para o estado normal 0001 faz-se num máximo de 4 ciclos de relógio, dispensando a utilização do RESET.

7. Questões Práticas de Sistemas Sequenciais

- PLDs Sequenciais (1) -

- Relembrar a PAL combinacional 16L8 (vista parcial do seu diagrama lógico):

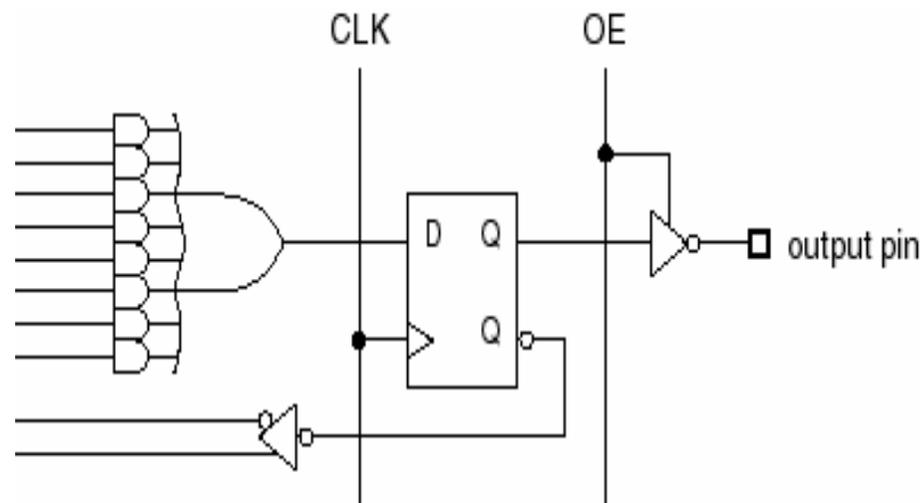


- 10 entradas primárias (I1..I10), 0 saídas registadas, 2 saídas combinacionais (O1 e O8), 6 entradas/saídas (IO2..IO7), cada OR recebe 7 termos de produto e cada AND recebe 16 entradas.

7. Questões Práticas de Sistemas Sequenciais

- PLDs Sequenciais (2) -

- ❑ A PAL sequencial 16R8 possui um *flip-flop* tipo D entre a saída de cada OR do array AND-OR e uma saída da PAL.
- ❑ O array AND-OR da PAL 16R8 é igual ao da PAL 16L8.
- ❑ Todos os *flip-flops* partilham o mesmo sinal de relógio **CLK**.
- ❑ Cada *flip-flop* liga a um *buffer tri-state*, que por sua vez liga ao exterior da PAL através dum pino. Todos os *buffers* partilham o mesmo sinal de *output enable* **OE**.



7. Questões Práticas de Sistemas Sequenciais

- PLDs Sequenciais (3) -

- ❑ As entradas do array AND-OR da PAL 16R8 são as entradas primárias I1 a I8 da PAL (na forma directa e complementada) mais as saídas dos 8 *flip-flops* (na forma directa e complementada).
- ❑ O facto de as saídas dos *flip-flops* alimentarem o array AND-OR facilita a implementação do bloco que gera as transições de estado nas máquinas de estados.
- ❑ Características da PAL 16R8: 8 entradas primárias, 8 saídas registadas, 0 saídas combinacionais, 0 entradas/saídas, cada OR recebe 8 termos de produto e cada AND recebe 16 entradas.
- ❑ A variante 16R6 de PAL dispõe de 2 entradas/saídas (sendo as saídas combinacionais) e 6 saídas registadas.