

7. Hardware programável sequencial .....	7-2
7.1    PAL sequencial .....	7-2
7.2    Construção <i>sequence present</i> .....	7-4
7.3    ATF750C .....	7-5

# 7. HARDWARE PROGRAMÁVEL SEQUENCIAL

Como anteriormente podemos observar na estrutura PAL descrita no capítulo 5, a saída da função que é posta disponível num pino de saída, também é disponibilizada na matriz de programação, permitindo por realimentação positiva constituir elementos de memória. Esta solução, seria consumidora de um elevado número de recursos, razão pela qual, os fabricantes de dispositivos programáveis põem disponíveis PALs que incluem *flip*-flops na estrutura da macro-célula.

## 7.1 PAL sequencial

Na Figura 7-1 está representada a macro-célula da PAL22V10 onde podemos observar as possíveis variantes programáveis da estrutura.

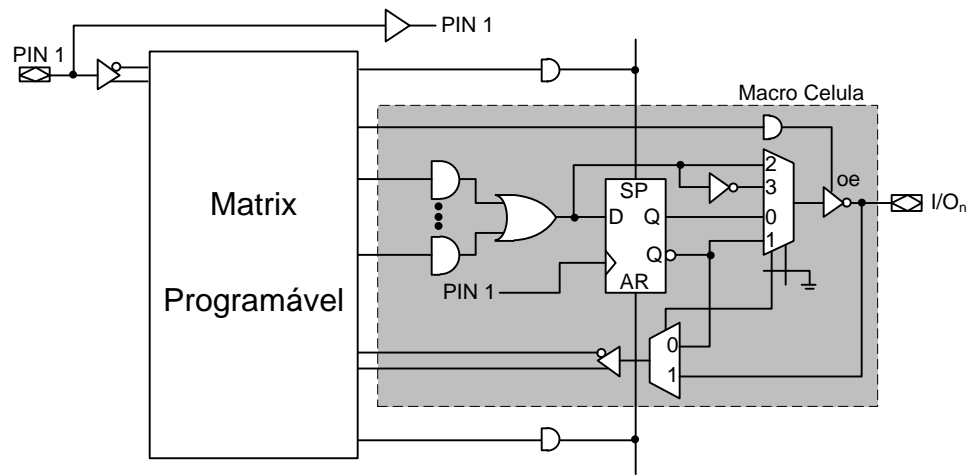


Figura 7-1

Como se pode ver na Figura 7-1, podemos definir se a saída da macro célula é combinatória ou sequencial (*registered*). Fica também disponível na matriz de programação a saída combinatória e a saída *registered*. Poderemos desta forma implementar com eficiência circuitos sequenciais baseados neste tipo de PAL. O acesso a cada um dos pontos da macro-célula faz-se evocando o nome da macro-célula separada por um ponto como mostra a Tabela 7-1.

MC.ar = termo produto	<i>Asynchronous Reset</i>
MC.sp = termo produto	<i>Synchronous Preset</i>
MC.d = união de termos produto	<i>D input</i>
MC.oe = termo produto	<i>Output Enable</i>
MC.io	<i>Input/Output Pin</i>

Tabela 7-1

O exemplo que se segue corresponde à implementação de um registo *edge trigger* de 8 bits com controlo de output.

```
Name      regist8 ;
PartNo    00 ;
Date      27-03-2010 ;
Revision  02 ;
Designer  JParaiso ;
Company   isel ;
Assembly  None ;
Location  ;
Device    p22v10 ;

/* ***** INPUT PINS ***** */
PIN      1 = CLK                ; /* clock de registo */
PIN      [2..9] = [D0..7]       ; /* entrada de dados do registo */
PIN      10 = OC                ; /* controlo de output */

/* ***** OUTPUT PINS ***** */
PIN      [14..21] = [Q0..7]     ; /* 8 flip-flops do registo */

[Q0..7].AR='b'0;
[Q0..7].SP='b'0;
[Q0..7].d=[D0..7];
[Q0..7].oe=oc;
```

As descrições CUPL que se seguem correspondem as várias implementações do ASM do exemplo 3 do capítulo 6. A primeira corresponde à implementação da figura 6.17, ou seja, utilizando para gerador de estado seguinte dois multiplexes. A segunda escrevendo as expressões simplificadas de ES0 e ES1, como acontece na figura 6.18. Como é óbvio, o resultado final será exactamente o mesmo, pois o compilador de CUPL encontrará com certeza as expressões OR-AND mais simples.

```
/* ***** INPUT PINS ***** */
PIN 1 = CLK; /* clock */
PIN 2 = A;
PIN 3 = B;

/* ***** OUTPUT PINS ***** */
PIN [14..15] = [X0..1]; /* memoria de estado */
PIN 16 = W;
PIN 17 = Y;
PIN 18 = Z;

Mux0 = [X1..1]:0;
Mux1 = [X1..1]:1;
Mux2 = [X1..1]:2;
Mux3 = [X1..1]:3;

ES0 = Mux0 # Mux1&A # Mux3&B;
ES1 = Mux1&A # Mux3;

X0.D = ES0;
X1.D = ES1;

W = !X1&X0;
Y = X1&X0;
Z = X1&!X0;
```

```

/* ***** INPUT PINS ***** */
PIN 1 = CLK; /* clock */
PIN 2 = A;
PIN 3 = B;

/* ***** OUTPUT PINS ***** */
PIN [14..15] = [X0..1]; /* memoria de estado */
PIN 16 = W;
PIN 17 = Y;
PIN 18 = Z;

ES0 = !X1&!X0 # !X1&A # X1&X0&B;
ES1 = X1&X0 # X0&A;

X0.D = ES0;
X1.D = ES1;

W = !X1&X0;
Y = X1&X0;
Z = X1&!X0;

```

## 7.2 Construção *sequence present*

O CUPL põe disponível uma construção que permite de uma forma simples a implementação de máquinas de estado. Esta construção denominada por *sequence present* consiste em indicar quais os *flip-flops* que constituem a memória de estado e em cada estado *present* indicar quais as condições de evolução para o próximo (*next*) estado e quais as saídas que ficam activas nesse estado. Na Figura 7-2 está descrito em linguagem CUPL a implementação do módulo de controlo do exercício 3 capítulo 6. O ASM descrito, apresenta exclusivamente saídas função de estado.

```

/* ***** INPUT PINS ***** */
PIN 1 = CLK ; /* clock da maquina de estados */
PIN 2 = A ; /* sensor de porta aberta */
PIN 3 = F ; /* sensor de porta fechada */
PIN 4 = P ; /* sensor de presenca */
/* ***** OUTPUT PINS ***** */
PIN 14 = ON ; /* Activacao do motor */
PIN 15 = SF ; /* sentido de rotacao do motor a fechar */
PIN [16..17] = [Q0..1] ; /* flip-flops para implementacao do ASM */

[Q0..1].AR='b'0;
[Q0..1].SP='b'0;
[Q0..1].CK=CLK;

sequence[Q1,Q0] {
  present 0
    if P next 1;
    if !P next 0;
  present 1
    out ON;
    if A next 2;
    if !A next 1;
  present 2
    if P next 2;
    if !P next 3;
  present 3
    out ON,SF;
    if F next 0;
    if !F&P next 1;
    if !F&!P next 3;
}

```

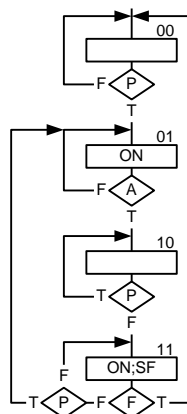


Figura 7-2

Se utilizarmos a implementação com variáveis de saída função de estado e entrada obteremos o programa descrito na Figura 7-3.

```

Name      ControloPorta ;
PartNo    00 ;
Date      11-06-2008 ;
Revision  02 ;
Designer  JParaíso ;
Company   isel ;
Assembly  None ;
Location  ;
Device    p22v10 ;

/* ***** INPUT PINS ***** */
PIN      1 = CLK                ; /* clock da maquina de estados */
PIN      2 = A                  ; /* sensor de porta aberta */
PIN      3 = F                  ; /* sensor de porta fechada */
PIN      4 = P                  ; /* sensor de presenca */
PIN      4 = T                  ; /* indicador de tempo */
/* ***** OUTPUT PINS ***** */
PIN      14 = ON                ; /* Activacao do motor */
PIN      15 = SF                ; /* sentido de rotacao do motor a fechar */
PIN      16 = Q0                ; /* flip-flop para implementacao do ASM */

Q0.AR='b'0;
Q0.SP='b'0;

sequence[Q0] {
  present 0
    if (!P & T & !F) out ON,SF;
    if P next 1;
    if !P next 0;
  present 1
    if !A out ON
    if A next 0;
    if !A next 1;
}

```

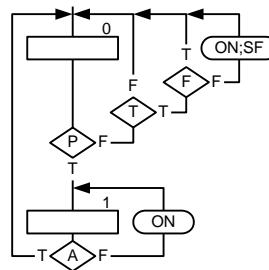


Figura 7-3

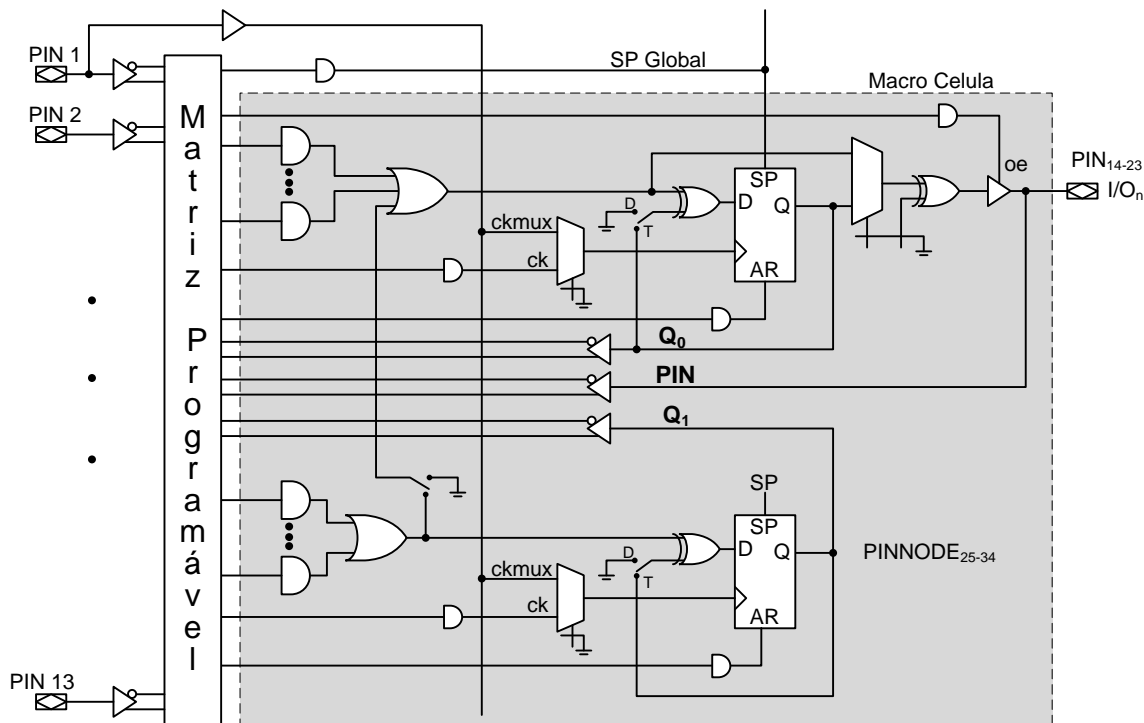
### 7.3 ATF750C

Dada a popularidade da PAL 22v10, o fabricante ATMEL pôs disponível no mercado uma PAL denominada ATF750, que mantendo compatibilidade com a 22v10 apresenta uma série de novas funcionalidades que a tornam mais versátil. Uma das características mais relevantes é o facto de conter o dobro dos *flip-flops* e permitir definir para cada um dos 20 *flip-flops* um termo produto gerador de CK (*clock*) e de AR (*Asynchronous Reset*), permitindo desta forma, a realização de vários módulos sequenciais independentes dentro de um mesmo dispositivo. Na Figura 7-4 é apresentado o diagrama de blocos da macro-célula da PAL ATF750C. Como se pode observar, cada macro-célula contém dois *flip-flops*, podendo cada um deles ser configurado para assumir comportamento tipo D ou T. Em CUPL, o primeiro *flip-flop* da macro-célula é referido como PIN e o segundo *flip-flop* é referido como PINNODE. O número de termos produto disponíveis em cada macro-célula é igual ao da 22V10, sendo estes distribuídos em partes iguais pelos dois *flip-flops*. No entanto, se numa qualquer aplicação o número de termos produto associado ao pino não for suficiente, são utilizados os termos produtos associados ao PINNODE, inviabilizando claro está, a utilização do PINNODE. Como se pode observar na Figura 7-4 a saída do *flip-flop* associado ao pino também está disponível na matriz através de **Q<sub>0</sub>**, tornando possível a utilização do *flip-flop* mesmo que o pino que lhe está associado seja utilizado como entrada.

A entrada de sincronismo de cada um dos flip-flops pode ter origem no pino 1 (CKMUX), ou num termo produto com entradas provenientes da matriz (CK). O sinal de *clock* proveniente do pino 1 através de CKMUX, não disponibiliza o complemento, mas apresenta a vantagem de suportar maior frequência e menor tempo de propagação que o gerado pelo termo produto.

A forma disponibilizada em CUPL para especificar qual o PINNODE ou NODE a utilizar, consiste na atribuição de números de pinos para além dos pinos físicos como mostra a Tabela 7-2.

Como se pode observar, o produto termo que realiza SP (*Synchronous Preset*) continua global.



**Figura 7-4**

Caso se pretenda utilizar o flip-flop tipo T na construção *sequence/present* acrescenta-se a letra T à instrução *sequence* (*sequenceT*).

PIN (Físico)	Q <sub>0</sub> (Flip-Flop)	Q <sub>1</sub> (PINNODE)	Número de Produtos
14	35	25	4+4
15	36	26	5+5
16	37	27	6+6
17	38	28	7+7
18	39	29	8+8
19	40	30	8+8
20	41	31	7+7
21	42	32	6+6
22	43	33	5+5
23	44	34	4+4

**Tabela 7-2**

Na PAL ATF750, os flip-flops associados aos pinos de saída, têm simultaneamente as suas saídas disponíveis na matriz de programação. A razão pela qual estas saídas estão disponíveis na matriz de programação deve-se ao facto dos pinos de saída poderem ser programados como pinos de entrada. No caso de tal acontecer, o flip-flop que lhe está associado pode continuar a ser utilizado, em estruturas que não necessitem de saída em pino físico (máquinas de estado, contadores, etc.). A referência a estas saídas, denominadas por  $Q_i$ , faz-se atribuindo na declaração pinos valores entre 35 e 44 como mostra a Tabela 7-2.