9. 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 põe disponíveis PALs incluindo *flip*-flops na estrutura da macro-célula. Na Figura 9-1 está representada a macro-célula da PAL22V10 onde podemos observar as possíveis variantes programáveis da estrutura.

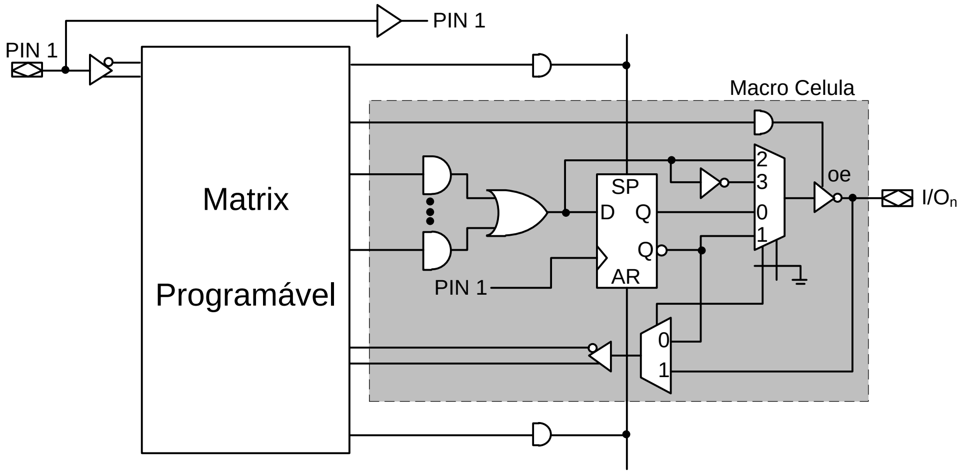

Figura 9-1

Como se pode ver na Figura 9-1, 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 9-1.

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

Tabela 9-1

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;
```

Para implementação de ASM, o CUPL põe disponível uma construção que permite de uma forma simples a sua descrição. Na Figura 9-2 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;

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;
}
```

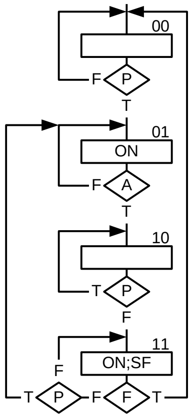

Figura 9-2

Se utilizarmos a implementação com variáveis de saída função de estado e entrada obteremos o programa descrito na Figura 9-3.

```
Name     ControloPorta ;
PartNo   00 ;
Date     11-06-2008 ;
Revision 02 ;
Designer JParaiso ;
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;
}
```

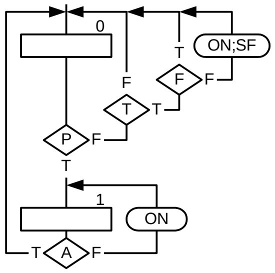

Figura 9-3

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 9-4 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 9-4 a saída do flip-flop associado ao pino também está disponível na matriz através de $\mathbf{Q_0}$, 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 9-2.

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

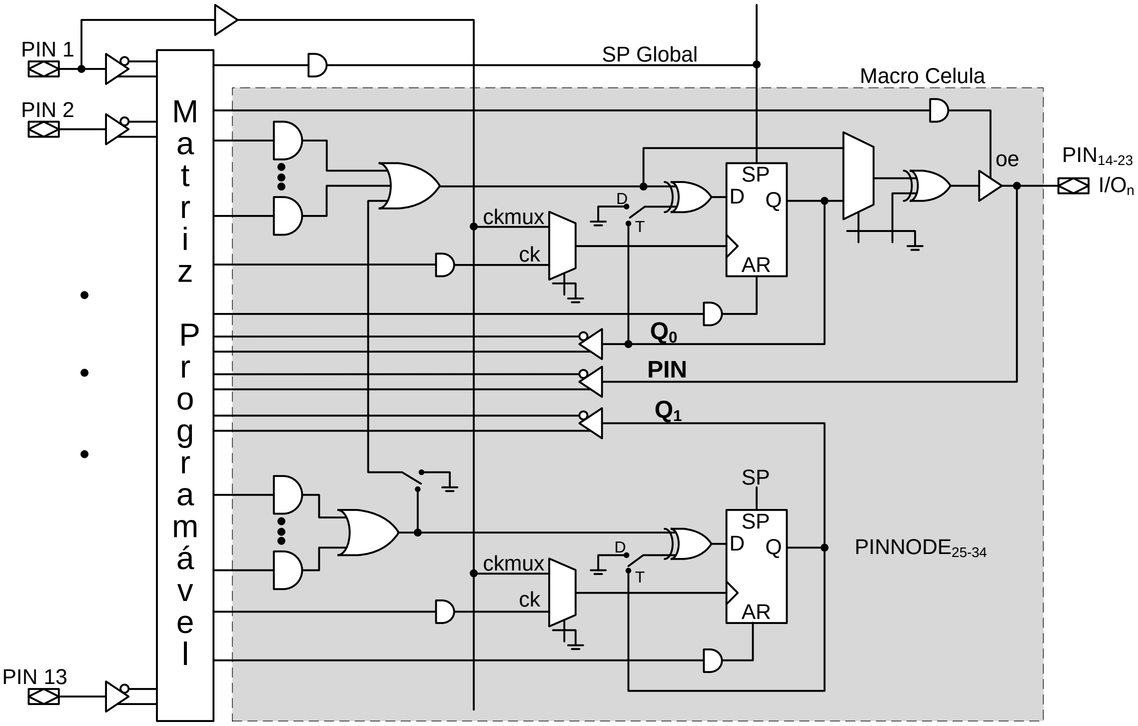

Figura 9-4

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

PIN (Fisico)	Q_0 (Flip-FLop)	Q_1 (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 9-2

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 9-2.

Exemplo:

O exemplo que se segue corresponde à implementação de um contador de 4 bits com entrada de CE (*Count Enable*) e saída Max (*Maxim*) e TC (*Terminal Count*) registado. A implementação do contador utiliza como célula de memória o *flip-flops* tipo T.

É utilizado um pino de I/O como entrada e é reutilizada a macro-célula que lhe está associado para registar a passagem do contador por Max.

É de notar que devido à estrutura da PAL, o contador tem implementação paralela, ou seja, a entrada T de cada célula toma simultaneamente o estado de todas as células que lhe estão a montante.

```
Name     Cont8 ;
PartNo   00 ;
Date     27-03-2010 ;
Revision 02 ;
Designer JParaiso ;
Company  isel ;
Assembly None ;
Location  ;
Device   v750c ;

/**************** INPUT PINS ********************************************/
PIN    1 = CLK;        /* clock de contagem                             */
PIN    2 = CE;         /* entrada de Count Enable                       */
PIN    15= TCR;        /* Terminal Count Reset (utiliza um pino de I/O) */

/***************** OUTPUT PINS ****************************************************************/
PIN 14 = Max;                /* saída que indica que o contador atingiu o valor maximo        */
PINNODE  [25..28] = [Q0..3]; /* 4 flip-flops do registo                                       */
PIN 36 = TC;                 /* registo da passagem por Max. Utiliza a macro celula do PIN 15 */

TC.AR=TCR;
TC.SP='b'0;
TC.CK= !CLK;
TC.D= !Max&TC # Max;

/* A solucao que a seguir se descreve e que corresponde ao controlo do sinal de clock, nao deve
ser utilizada, pois o sinal Max pode surgir na passagem do valor 7 para o valor 8.

TC.CK= Max;
TC.D= 'b'1;
*/

[Q0..3].AR='b'0;
[Q0..3].SP='b'0;
[Q0..3].CK= CLK;

Q0.t=CE;
Q1.t=Q0&CE;
Q2.t=Q0&Q1&CE;
Q3.t=Q0&Q1&Q2&CE;

Max=[Q0..3]:&;
```