

PROJETO 5: DECODIFICADOR BCD PARA SETE SEGMENTOS USANDO TABELA LÓGICA.

O desenvolvimento no modo AHDL permite utilizar a declaração de tabelas lógicas para a criação de circuitos combinacionais, tal como o de um circuito decodificador. A listagem da figura 5.1 exemplifica o projeto de um decodificador BCD para sete segmentos.

```

Title " Decodificador BCD para 7 segmentos "

Subdesign proj5

(   i[3..0]           :INPUT;

    a,b,c,d,e,f,g     :OUTPUT; )

BEGIN
    TABLE
        i[3..0]        => a,b,c,d,e,f,g;

        H"0"           => 1,1,1,1,1,0,0;
        H"1"           => 0,1,1,0,0,0,0;
        H"2"           => 1,1,0,1,1,0,1;
        H"3"           => 1,1,1,1,0,0,1;
        H"4"           => 0,1,1,0,0,1,1;
        H"5"           => 1,0,1,1,0,1,1;
        H"6"           => 1,0,1,1,1,1,1;
        H"7"           => 1,1,1,0,0,0,0;
        H"8"           => 1,1,1,1,1,1,1;
        H"9"           => 1,1,1,1,0,1,1;
        H"A"           => 0,0,0,0,0,0,0;
        H"B"           => 0,0,0,0,0,0,0;
        H"C"           => 0,0,0,0,0,0,0;
        H"D"           => 0,0,0,0,0,0,0;
        H"E"           => 0,0,0,0,0,0,0;
        H"F"           => 0,0,0,0,0,0,0;
    END TABLE;
END;

```

Fig.5.1 Decodificador BCD - 7 segmentos em AHDL

Para a implementação do projeto execute os seguintes passos:

- 1) Execute o software MAX + PLUS II e inicialize o projeto com o nome **proj5**, dentro do subdiretório max2work.

Para tal, selecione a opção Project Name dentro do menu File.

Habilite o subdiretório max2work no campo Directories.

Entre com o nome proj5 no campo Project Name.

Selecione OK.

Verifique que na parte superior da tela está o caminho de desenvolvimento do projeto, por exemplo, c:\max2work\proj5.

- 2) Abra a área de desenvolvimento do projeto usando a opção New do menu File.

Habilite a opção Text Editor File que terá extensão .Tdf.

Selecione OK.

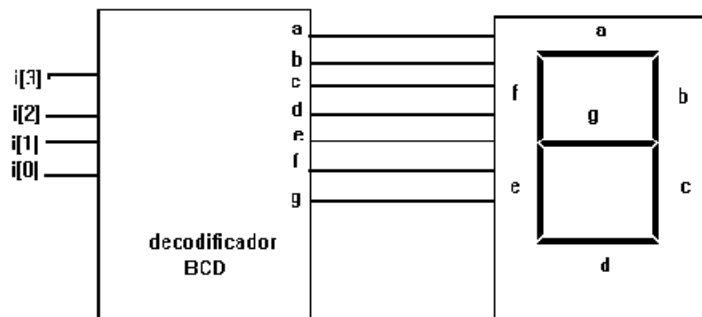
- 3) Entre com o texto em AHDL correspondente para o projeto, conforme apresentado na figura 5.1.

- 4) Comentário sobre o projeto

As declarações Title, Subdesign, Begin e End fazem parte da estrutura de um projeto no modo AHDL e já foram comentadas no projeto 4

A declaração **Table** é usada para definir uma lógica combinatorial através de tabelas lógicas relacionando entradas e saídas.

No projeto o circuito possui as entradas i[3], i[2], i[1], i[0] e como saídas os sinais para os correspondentes sete segmentos do display, ou seja:



As entradas (4 bits) assumirão valores no código hexadecimal, ou seja de 0 até F. Entretanto, como o projeto visa implementar um decodificador BCD, ou seja decimal, para os códigos maiores que 9, optou-se para que as saídas permaneçam inativas, ou seja display apagado, poderia ter-se optado por uma indicação qualquer de ultrapassagem do código.

Este é um caso de decodificador onde todas as combinações de entradas são possíveis, denominado de decodificador não parcial, neste caso pode-se solucionar o projeto usando também a declaração **DEFAULTS**.

A declaração **DEFAULTS** pode ser usada para especificar a saída de um decodificador, quando não houver valores definidos para certas combinações de entradas e, nestes casos, a saída assumiria um valor especificado (valor default).

A figura 5.2 apresenta a solução do projeto com o uso da declaração DEFAULTS.

Para efeitos didáticos, nesta figura as entradas foram definidas usando o código binário.

```

Title " Decodificador BCD para 7 segmentos usando a declaração defaults ";

Subdesign proj5

(   i[3..0]                :INPUT;

    a,b,c,d,e,f,g          :OUTPUT; )

BEGIN
  DEFAULTS
    a   = B"0";           % valores das saídas para      %
    b   = B"0";           % códigos não especificados    %
    c   = B"0";
    d   = B"0";
    e   = B"0";
    f   = B"0";
    g   = B"0";

  END DEFAULTS;

  TABLE
    i[3..0]      => a,b,c,d,e,f,g;

    B"0000"      => 1,1,1,1,1,1,0;
    B"0001"      => 0,1,1,0,0,0,0;
    B"0010"      => 1,1,0,1,1,0,1;
    B"0011"      => 1,1,1,1,0,0,1;
    B"0100"      => 0,1,1,0,0,1,1;
    B"0101"      => 1,0,1,1,0,1,1;
    B"0110"      => 1,0,1,1,1,1,1;
    B"0111"      => 1,1,1,0,0,0,0;
    B"1000"      => 1,1,1,1,1,1,1;
    B"1001"      => 1,1,1,1,0,1,1;

  END TABLE;
END;

```

Fig.5.2 Decodificador BCD - 7 segmentos em AHDL, usando a declaração Defaults

- 5) Salvar o projeto usando a opção **Save As** no **menu File**.

Verifique que o projeto esteja com o mesmo nome no campo **File Name**, que o diretório esteja no caminho correto, por exemplo **c:\MAX2WORK** e que a extensão seja **.tdf**.

Selecione OK.

- 6) Siga os passos seguintes descritos no Projeto1 para compilar, programar e testar o dispositivo.

Se não houver erro, serão criados os arquivos:

proj5.cnf - que contem informações da lógica e conexões do projeto.

proj5.rpt - que contem informações gerais de implementação.

proj5.snf - que contem base de dados para simulações funcionais.

proj5.pof - que contem as informações para programação do dispositivo.

- 7) Para testar o projeto efetue as conexões necessárias de entradas e saídas, conforme mostrado e complete a tabela seguinte. Use o arquivo proj5.rpt para verificar as pinagens correspondentes.

ENTRADAS/CHAVES				SAÍDAS/LED'S						
i[3]	i[2]	i[1]	i[0]	a	b	c	d	e	f	g
A	B	C	D	L6	L5	L4	L3	L2	L1	L0
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
Demais códigos										

8) Verifique os resultados.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.