



Microeletrônica I

LAB 2

Projeto I – Decodificador BCD para 7 segmentos

- A figura 1 mostra um decodificador BCD para 7 segmentos, com as saídas ativas em nível lógico baixo. A figura 2 mostra os displays de 7 segmentos do kit nexys 2.

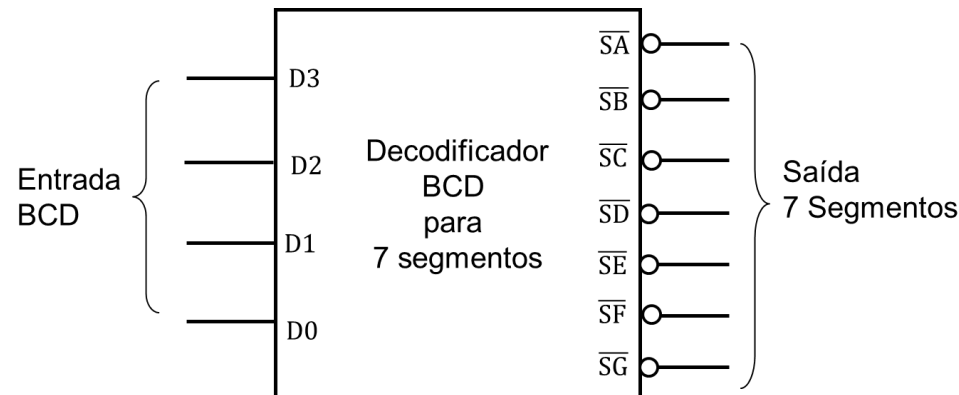


Figura 1 – Decodificador BCD para 7 segmentos.

Projeto I – Decodificador BCD para 7 segmentos

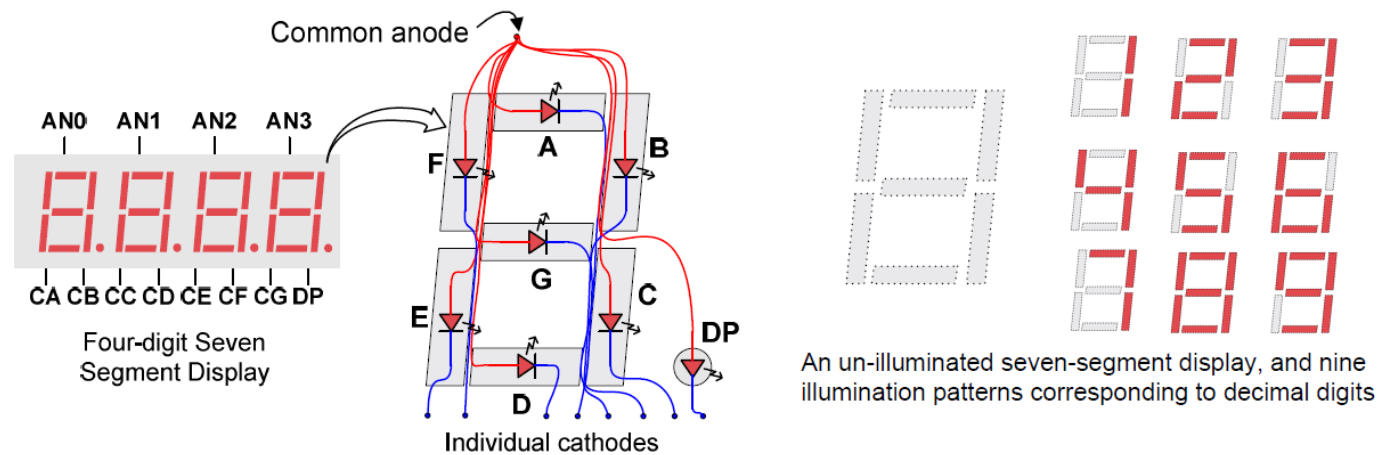


Figura 2 – Displays de 7 segmentos do kit nexys 2.

Projeto I – Decodificador BCD para 7 segmentos

- 1) Implemente em VHDL o decodificador BCD para 7 segmentos. Utilize apenas código concorrente (instruções: WHEN, SELECT e GENERATE);
- 2) Apresente a simulação comportamental para diferentes combinações das entradas;
- 3) Apresente o esquemático RTL obtido após a síntese;
- 4) Para implementação no kit Nexys 2, utilize os switches para as entradas do circuito e 1 display de 7 segmentos para as saídas. Obs: será necessário mapear o anodo do SSD escolhido).

Obs: Os pinos de entrada e saída devem ser do tipo `std_logic`

Projeto 2 - Peso de Hamming

O peso de Hamming de um vetor é o número de '1's neste vetor. Por exemplo, o peso de Hamming do vetor "1001011" é 4.

Projeto 2 - Peso de Hamming

- 1) Projete um circuito que determine o peso de Hamming de um vetor de comprimento genérico, utilizando apenas código concorrente (instruções: WHEN, SELECT e GENERATE);
- 2) Apresente a simulação para diferentes combinações das entradas;
- 3) Apresente o esquemático RTL obtido após a síntese;
- 4) Para implementação no kit Nexys 2 defina $N=8$. Utilize os switches para as entradas do circuito e um display de 7 segmentos mostrar o valor da saída (para isso reutilize o decodificador BCD implementado no projeto 1).

Obs: Todas as portas de entrada e saída devem ser especificadas como STD_LOGIC(_VECTOR).

Projeto 3 – Ordenador binário com GENERATE

- 1) Utilizando apenas código concorrente (instruções: WHEN, SELECT e GENERATE), projete um circuito capaz de ordenar os bits de um vetor de comprimento genérico. A ordem deve ser da esquerda para a direita, com todos os '1's vindo primeiro (por exemplo, "00011001" se tornará "11100000").
Sugestão: primeiro conte o número de '1's (isto é, determine o peso de Hamming do vetor) e depois realize a ordenação.
- 2) Apresente a simulação para diferentes combinações das entradas.
- 3) Para implementação no kit Nexys 2, defina N=8. Utilize os switches para o vetor de entrada e LEDs para o vetor de saída.

Obs: Os pinos de entrada e saída devem ser do tipo `std_logic` ou `std_logic_vector`

Projeto 4 - Circuito Aritmético com STD_LOGIC

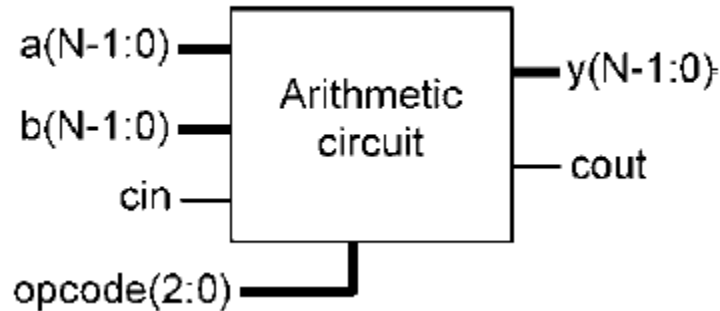


Figura 1(a)

| Tipo | Operação | Opcode |
|----------|-------------------|--------|
| Unsigned | $y = a + b$ | 000 |
| | $y = a - b$ | 001 |
| | $y = -a + b$ | 010 |
| | $y = a + b + cin$ | 011 |
| Signed | $y = a + b$ | 100 |
| | $y = a - b$ | 101 |
| | $y = -a + b$ | 110 |
| | $y = a + b + cin$ | 111 |

Figura 1(b)

Figura 3 - Mini ULA.

Projeto 4 - Circuito Aritmético com STD_LOGIC

- 1) Projete o circuito aritmético mostrado na figura 3(a) que realiza as operações especificadas na tabela da figura 3(b) (que é uma mini ULA, com apenas a unidade aritmética). Utilize apenas código concorrente.
- 2) No código VHDL, o número de bits das entradas a e b e da saída y deve ser genérico (N), para isso use o parâmetro GENERIC.
- 3) Apresente a simulação para diferentes combinações das entradas; Para simulação defina $N=4$.
- 4) Apresente o esquemático RTL obtido após a síntese;
- 5) Para apresentação no kit Nexys2 defina $N=2$. Utilize os switches para as entradas do circuito e os LEDs para as saídas;

Obs: Todas as portas de entrada e saída devem ser especificadas como STD_LOGIC(_VECTOR).

Apresentação dos Resultados

- Implementar os projetos na ferramenta ISE e apresentar o funcionamento de cada projeto no kit Nexys2 (arquivo .BIT) durante a aula de laboratório.
- Elaborar um relatório técnico incluindo a descrição e o desenvolvimento de cada projeto, assim como os esquemáticos RTL obtidos após a síntese. As simulações para diferentes combinações das entradas também devem ser apresentadas (todos os resultados mostrados devem ser comentados e justificados). Lembre-se de incluir no relatório os códigos do design e do testbench e resumo dos recursos lógicos utilizados.
- O trabalho deverá ser entregue pela UFPR Virtual (faça um arquivo compactado incluindo os projetos no ISE e o relatório).