Universidade Federal de Viçosa - Campus Florestal
CCF 251 – Introdução aos Sistemas Lógicos Digitais
Prof. José Augusto Miranda Nacif

Trabalho Prático 1 - Circuitos Combinacionais (Simulação)

Gabriel Rodrigues Marques - 5097
Alan Araújo dos Reis - 5096

OBJETIVO

Neste Trabalho Prático, foi proposto o desafio de criar um sistema que transforma um valor de 4-bits em outro valor de 4-bits equivalente. Em outras palavras, quando fornecido um número binário como entrada, o programa deve ser capaz de realizar uma conversão especial, agindo como uma forma de codificação da entrada. Além disso, era preciso exibir o número binário codificado e sua saída para um display de 7 segmentos.

O primeiro objetivo deste projeto foi o de desenvolver expressões lógicas com base em uma tabela-verdade fornecida e empregar Mapas de Karnaugh para simplificá-las. Isso, consequentemente, resultará em expressões simplificadas que aumentarão a eficiência do sistema de codificação. Com essas expressões lógicas definidas, nosso segundo objetivo é implementá-las usando a linguagem de descrição de hardware Verilog, permitindo que o programa realize a codificação de números no intervalo de 0 a 15. Futuramente, os módulos aqui elaborados e simulados serão utilizados para uma implementação em FPGA.

DESENVOLVIMENTO E RESULTADOS

Inicialmente, a partir da tabela-verdade fornecida, foi elaborada uma segunda tabela (Imagem 1) com INPUT (Entrada 4-bits), OUTPUT (Codificado 4-bits) e DISPLAY (Display 7-bits). Os sinais de controle Ready e Reset não foram diretamente colocados como entrada, pois não interferem na conversão e sim na saída esperada.

INPUT					OUTPUT					DISPLAY							
Entrada	A	B	C	D	S3	S2	S1	S0	Saída	A	B	C	D	E	F	G	Saída
0	0	0	0	0	1	1	1	0	14	1	0	0	1	1	1	1	E
1	0	0	0	1	1	0	1	0	10	1	1	1	0	1	1	1	A
2	0	0	1	0	0	1	0	0	4	0	1	1	0	0	1	1	4
3	0	0	1	1	1	1	0	1	13	0	1	1	1	1	0	1	D
4	0	1	0	0	0	1	1	0	6	1	0	1	1	1	1	1	6
5	0	1	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1
6	0	1	1	0	0	1	1	1	7	1	1	1	0	0	0	0	7
7	0	1	1	1	1	1	1	1	15	1	0	0	0	1	1	1	F
8	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
9	1	0	0	1	1	0	1	1	11	0	0	1	1	1	1	1	B
10	1	0	1	0	0	0	1	0	2	1	1	0	1	1	0	1	2
11	1	0	1	1	1	0	0	0	8	1	1	1	1	1	1	1	8
12	1	1	0	0	1	0	0	1	9	1	1	1	1	0	1	1	9
13	1	1	0	1	0	0	1	1	3	1	1	1	1	0	0	1	3
14	1	1	1	0	1	1	0	0	12	1	0	0	1	1	1	0	C
15	1	1	1	1	0	1	0	1	5	1	0	1	1	0	1	1	5

Imagem 1 - Tabela Verdade do Sistema Codificador

Com a tabela-verdade devidamente elaborada, foi feita a construção das expressões para cada saída do sistema. Para o nosso sistema, foi adotado expressões lógicas na forma de soma de produtos dos mintermos. Para o desenvolvimento das expressões, nas formas canônicas e simplificadas, foi utilizado o software Logisim para facilitar o trabalho. No Logisim, a partir das entradas, saídas e tabela-verdade, é possível desenvolver e simplificar as expressões com Mapas de Karnaugh (Imagem 2 e Imagem 3).

Imagem 2 - Mapas de Karnaugh com Mintermos (Soma de Produtos)

Imagem 3 - Mapas de Karnaugh com Maxtemos (Produto de Somas)

Equações de S3, S2, S1 e S0 nas formas canônicas e simplificada de **mintermos**:

$$S3(A,\ B,\ C,\ D) = \Sigma m(0, 1, 3, 7, 9, 11, 12, 14)$$
$$S3(A,\ B,\ C,\ D) = A'\,B'\,C' \ +\ B'\,D \ +\ A'\,C\,D \ +\ A\,B\,D'$$

$$S2(A,\ B,\ C,\ D) = \Sigma m(0, 2, 3, 4, 6, 7, 14, 15)$$
$$S2(A,\ B,\ C,\ D) = A'\,D' \ +\ A'\,C \ +\ B\,C$$

$$S1(A,\ B,\ C,\ D) = \Sigma m(0, 1, 4, 6, 7, 9, 10, 13)$$
$$S1(A,\ B,\ C,\ D) = A'\,B'\,C' \ +\ A'\,C'\,D' \ +\ A'\,B\,C \ +\ A\,C'\,D \ +\ A\,B'\,C\,D'$$

$$S0(A,\ B,\ C,\ D) = \Sigma m(3, 5, 6, 7, 9, 12, 13, 15)$$
$$S0(A,\ B,\ C,\ D) = A'\,C\,D \ +\ B\,D \ +\ A'\,B\,C \ +\ A\,C'\,D \ +\ A\,B\,C'$$

Equações de S3, S2, S1 e S0 nas formas canônicas e simplificada de **maxtermos**:

$$S3(A,\ B,\ C,\ D) = \Pi M(2, 4, 5, 6, 8, 10, 13, 15)$$
$$S3(A,\ B,\ C,\ D) = (A\ +\ C'\ +\ D)\ (A\ +\ B'\ +\ C)\ (A'\ +\ B\ +\ D)\ (A'\ +\ B'\ +\ D')$$

$$S2(A,\ B,\ C,\ D) = \Pi M(1, 5, 8, 9, 10, 11, 12, 13)$$
$$S2(A,\ B,\ C,\ D) = (C\ +\ D')\ (A'\ +\ B)\ (A'\ +\ C)$$

$$S1(A,\ B,\ C,\ D) = \Pi M(2, 3, 5, 8, 11, 12, 14, 15)$$
$$S1(A,\ B,\ C,\ D) = (A\ +\ B\ +\ C')\ (B\ +\ C'\ +\ D')\ (A\ +\ B'\ +\ C\ +\ D')\ (A'\ +\ C\ +\ D)\ (A'\ +\ B'\ +\ C')$$

$$S0(A,\ B,\ C,\ D) = \Pi M(0, 1, 2, 4, 8, 10, 11, 14)$$
$$S0(A,\ B,\ C,\ D) = (A\ +\ B\ +\ C)\ (A\ +\ C\ +\ D)\ (B\ +\ D)\ (A'\ +\ B\ +\ C')\ (A'\ +\ C'\ +\ D)$$

O mesmo processo adotado anteriormente para as expressões de codificação também foi adotado para desenvolver as equações do display de 7 segmentos, sendo utilizado soma de produtos dos mintermos. Os detalhes podem ser encontrados no módulo Verilog e no circuito do Logisim.

Em sequência, com as equações simplificadas, foi elaborado o circuito simplificado com portas lógicas através do Logisim. Os circuitos Codificador (Imagem 4) e Display (Imagem 5) foram gerados.

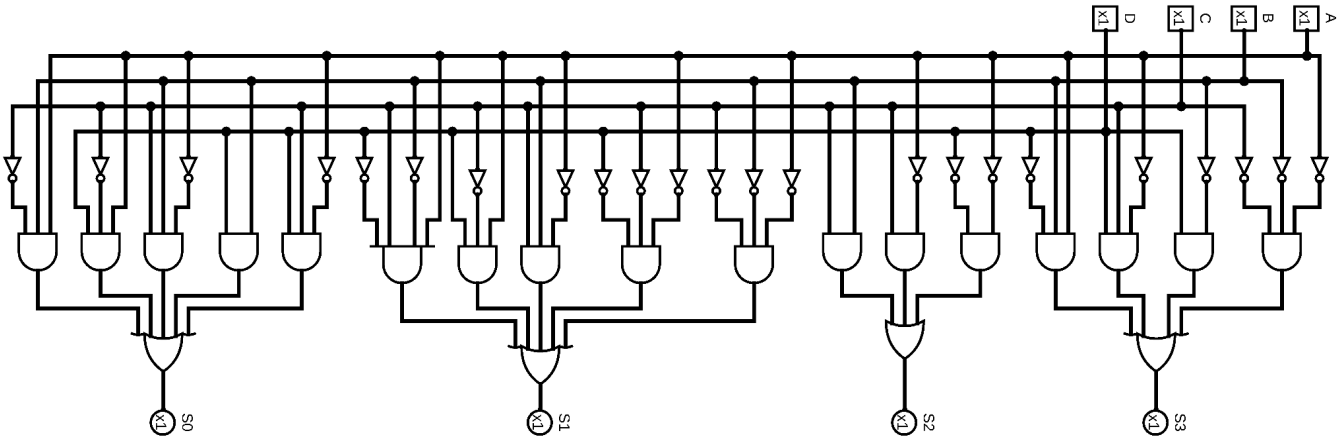

Imagem 4 - Circuito do Codificador

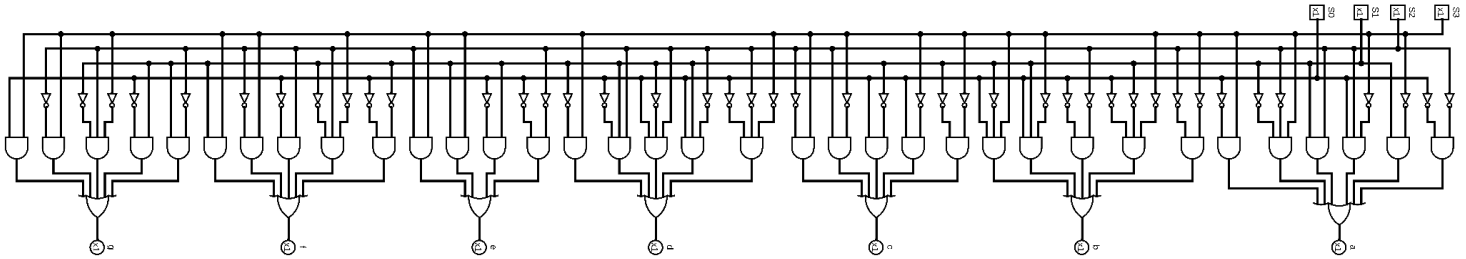

Imagem 5 - Circuito do Display

Além dos circuitos Codificador e Display, foi implementado um circuito Simulação do Codificador (Imagem 6) para verificar se as entradas e saídas eram compatíveis com o que foi elaborado na tabela-verdade, verificando também se as expressões desenvolvidas estavam corretas.

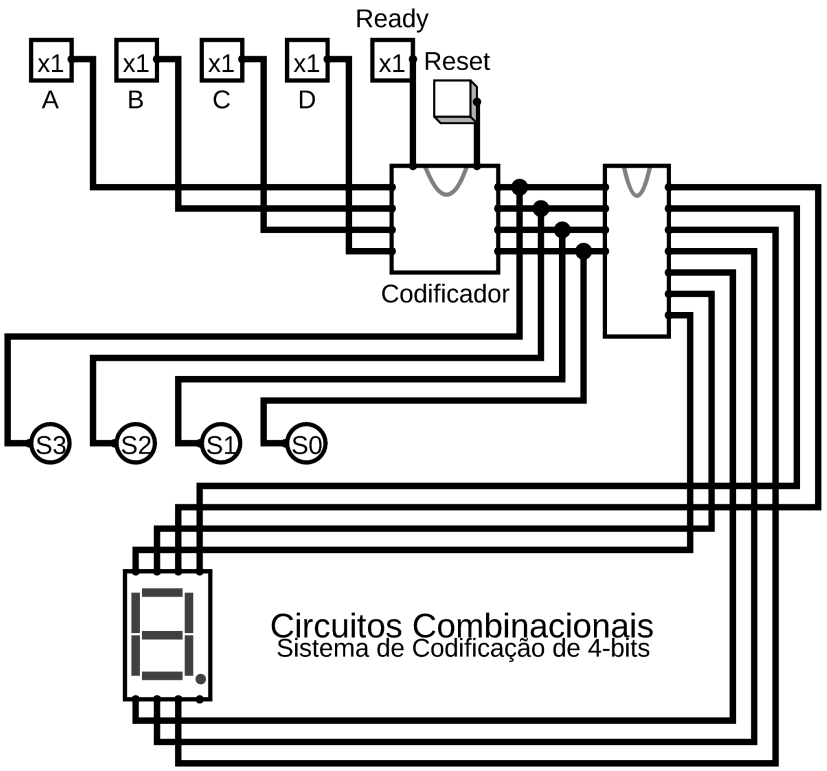

Imagem 6 - Simulação do Codificador

Após a elaboração dos circuitos no Logisim e verificado que tudo estava funcionando corretamente, foi utilizada a linguagem Verilog, através da ferramenta Icarus Verilog, para elaboração e simulação dos módulos do projeto. A implementação foi dividida nos módulos Codificador e Display, além do módulo Testbench para simulações.

No módulo Codificador foi utilizado um operador ternário sensível ao valor dos sinais de controle Ready e Reset para realizar a operação de codificação.

assign Saída = (Ready & ~Reset) ? Expressão de Codificação : 1'b0;

Já o módulo Display, utiliza a saída do Codificador para definir quais segmentos do display deverão ser acesos. O Testbench foi implementado com casos de teste que vão de 0 a 15, definindo os sinais de Ready em 1 e Reset em 0. Como saída tem-se a entrada, o número codificado e os segmentos do display que serão acesos.

Entrada : A B C D

Saída : S3 S2 S1 S0

Display : a b c e f g

É gerado um arquivo para visualização das formas de onda resultantes através da ferramenta GTKWave (Imagem 7). As formas de onda permitem uma representação gráfica detalhada das variações dos de entrada e saída do circuito ao longo do tempo de simulação. Essa representação facilita visualizar o comportamento do projeto.

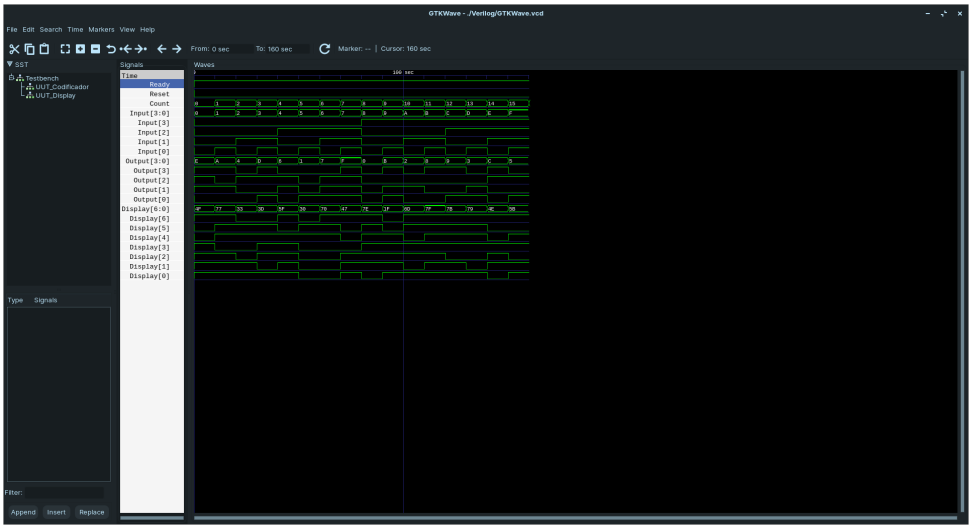

Imagem 7 - Diagrama GTKWave

CONCLUSÃO

Por fim, com a utilização de ferramentas disponibilizadas, este Trabalho Prático foi bem-sucedido na criação de um sistema de codificação de 4-bits, utilizando Mapas de Karnaugh para simplificar as expressões lógicas. O projeto não apenas atingiu seus objetivos, mas também demonstrou a aplicação prática dos conceitos de lógica digital e elaboração e simulação de módulos em Verilog. Além disso, o terreno também foi preparado para futuras implementações no hardware da FPGA.

REFERÊNCIAS

[1] R. Katz, G. Borriello, Contemporary Logic Design, 2ª edição, Prentice Hall, 2004;
[2] TANENBAUM, A.S. Organização Estruturada de Computadores. 5. ed. Editora Pearson Prentice Hall, 2007;
[3] HDLBits — Verilog Practice. Disponível em: <https://hdlbits.01xz.net/wiki/Main_Page>
[4] Github. Disponível em: <https://github.com/gabridulol/CircuitosCombinacionais>;
[5] Icarus Verilog;
[6] GTKWave;
[7] Logisim;