

Gabriel Rodrigues Marques - 5097

Alan Araújo dos Reis - 5096

OBJETIVO

Nesta segunda parte do Trabalho Prático foi solicitado a implementação em FPGA dos módulos anteriormente elaborados: Display e Codificador.

DESENVOLVIMENTO E RESULTADOS

Inicialmente foi preciso configurar todo ambiente para trabalhar com a FPGA. Foi utilizado uma FPGA Cyclone IV da Altera, modelo EP4CE115F29C7, e dois programas: Intel Quartus Prime Design Software e Terasic DE2-115 System Builder. O System Builder foi utilizado para gerar o projeto e os arquivos necessários no Quartus. No Quartus, foi desenvolvido o projeto e realizada a compilação e síntese para a FPGA.

Com o ambiente devidamente configurado, foi feita a ligação das entradas e saídas do módulo com a FPGA (*DE2_115.v*).

Codificador Codificador (

.Input(SW[3:0]),

.Ready(KEY[0]),

.Reset(KEY[1]),

.Output(LED[3:0])

);

Display Display (

.Output(LED[3:0]),

.Display(HEX[6:0])

);

Foi necessário fazer algumas alterações nos módulos Verilog do Codificador e Display para que obtivessem o funcionamento esperado na FPGA. Os módulos Codificador e Display foram colocados em um único arquivo (*Verilog1.v*).

O operador ternário sensível ao valor dos sinais de controle Ready e Reset foi alterado para que funcionasse corretamente nos botões da FPGA. A alteração foi realizada devido ao fato de que os botões operam de uma forma não habitual, ou seja, em um nível lógico diferente. Enquanto não pressionados, eles enviam um sinal de nível lógico alto (1) e, quando pressionados, nível lógico baixo (0), prejudicando as funções de Ready e Reset.

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

foi alterado para

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

Agora, com essa modificação, o módulo do Codificador sempre está em um “estado” de Reset e quando o valor a ser codificado é inserido e o botão Ready pressionado a codificação é realizada.

O módulo Display também foi alterado pelo mesmo motivo anterior. Os segmentos do display são acesos em 0 e não em 1. Desse modo, toda a expressão de codificação do segmento foi complementada.

assign Display [Segmento] = Expressão de Codificação

foi alterado para

assign Display [Segmento] = ~(Expressão de Codificação)

Por último, a ordem de atribuição dos segmentos do display foi alterada, pois houve um erro ao definir qual segmento do display era mais/menos significativo.

Display[6] ... Display[0]

foi alterado para

Display[0] ... Display[6]

Com todas as alterações mencionadas foi possível obter o resultado que estava sendo esperado: a codificação de um número binário em outro e sua exibição tanto em binário por LEDs como em hexadecimal pelo display de 7 segmentos.

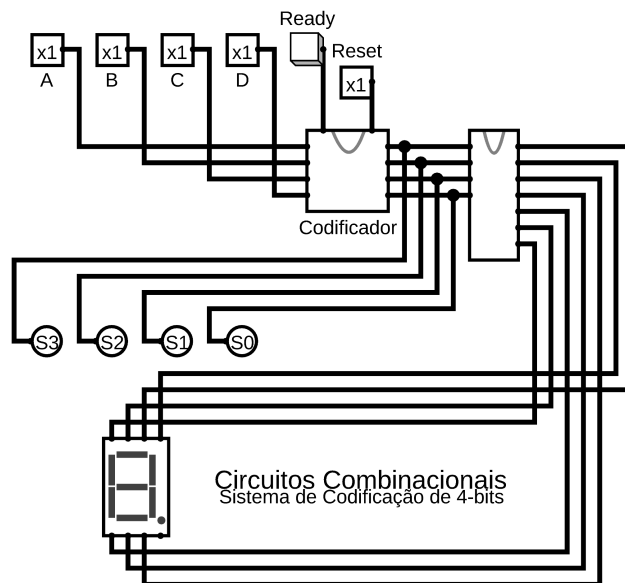


Imagem 1 - Circuito Idealizado

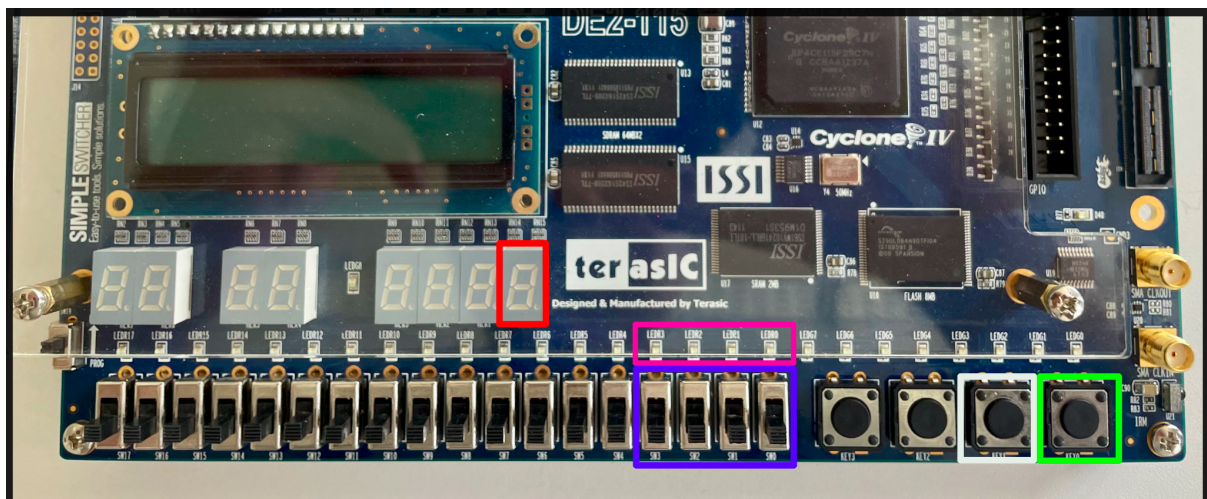


Imagem 2 - FPGA

O circuito idealizado (Imagem 1) é bem similar ao que foi desenvolvido em FPGA (Imagem 2). Na FPGA tem-se a entrada *A B C D* em azul, a saída do número binário *S4 S3 S2 S1* nos LEDs em rosa e no display em vermelho na sua forma hexadecimal. Além disso, de verde e branco são os sinais de controle *Ready* e *Reset*

OBS.: mais detalhes do resultado podem ser vistos no vídeo.

CONCLUSÃO

Por fim, ao final do desenvolvimento da segunda parte do Trabalho Prático, com a utilização da ferramenta Quartus e algumas alterações nos módulos implementados, foi possível obter o resultado idealizado na etapa anterior que atende às especificações solicitadas. O projeto demonstrou a aplicação prática do que é possível implementar em uma FPGA e as etapas que devem ser seguidas.

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] Intel Quartus Prime Lite Edition Design Software;
- [7] Terasic DE2-115 System Builder;