

Fluxo de Projetos com FPGAs

Disciplina:
Laboratório de Sistemas Digitais

Professor:
Jhonattan Córdoba Ramírez

Alunos:
João Pedro Copelli - 2021014414
João Victor Gomes - 2020072690
Marcelle Christine Aquino Silva - 2021014546

1 de outubro de 2022

Sumário

1	Introdução	3
2	Parte Teórica	3
2.1	O que é um FPGA?	3
2.2	Para que serve uma FPGA?	3
2.3	Quais partes compõem um FPGA?	3
2.4	Qual é o objetivo de cada parte de um FPGA?	3
2.5	Como funciona cada parte de um FPGA?	4
2.6	Como programar um FPGA?	4
2.7	Quais partes compõem o kit Altera DE2 – Therasic?	5
2.8	Quais partes do kit Altera DE2 – Therasic são usadas para gravação e depuração de sistemas Digitais?	5
3	Parte Prática	5
4	Conclusão	7

1 Introdução

Neste relatório serão discutidos os primeiros conceitos sobre FPGAs, para que serve um FPGA, quais as suas partes e objetivos . Mais adiante discutiremos sobre como gravar e testar as aplicações em uma FPGA.

2 Parte Teórica

2.1 O que é um FPGA?

É um dispositivo lógico programável que suporta a implementação de circuitos digitais. Sendo um hardware programável, pode alterar as ligações dos circuitos para determinada aplicação e, para uma outra aplicação, alterar essas ligações para atingir outros objetivos.

2.2 Para que serve uma FPGA?

Serve para implementar circuitos digitais, como processadores, interfaces, controladores e decodificadores, enviados ao hardware em uma linguagem de descrição de hardware ou esquemas de portas lógicas

2.3 Quais partes compõem um FPGA?

O chip é dividido em 3 componentes principais:

- Blocos lógicos configuráveis e de funções fixas;
- Blocos de E/S;
- Blocos de Comutação

2.4 Qual é o objetivo de cada parte de um FPGA?

- Blocos de Entrada e Saída: Entrada e saída de dados
- Blocos Lógicos: Unidades que guardam pequenas funções lógicas

- Switches: Controle de dados de acordo com a implementação

2.5 Como funciona cada parte de um FPGA?

- Blocos de E/S
 - São os módulos no qual são feitas as interconexões dos dados de entrada como os resultados dos blocos lógicos configuráveis. Podemos assimilar sua definição com as funcionalidades de um buffer de teclado por exemplo no qual transmite os dados oriundo do teclado mas que também além de enviar, poderia receber dados.
- Blocos lógicos configuráveis e de funções fixas
 - São fabricados de forma bidimensional criando pequenos 'módulos funcionais' de circuitos. São organizados com conjuntos de flip-flop com a lógica combinacional. Sobre uma definição mais geral, eles comportam-se como uma tabela verdade no qual é possível, com determinada estimulação, produzir determinada saída. Este conceito será descrito mais detalhado no decorrer deste documento.
- Blocos de Comutação
 - São as trilhas que formam a conexão entre os blocos de entrada e saída com os blocos lógicos configuráveis. Células de memória estáticas são configuradas pelo próprio compilador da placa determinando as funções lógicas que serão utilizadas e as conexões internas. Além da função de conectar os pinos externos aos blocos lógicos, elas também fazem o papel de interconexão de blocos.

2.6 Como programar um FPGA?

Para programar uma FPGA, normalmente os fabricantes disponibilizam softwares específicos com as arquiteturas de cada FPGA. Nesses softwares, são utilizadas linguagens de descrição de hardware (HDL) para descrever quais ligações deveram ser feitas no módulo e quais lógicas devem ser executadas com as entradas e sinais recebidos pela FPGA.

Exemplos de linguagens de descrição de hardware: Verilog, SystemC, Lola, Lava, MyHDL. Antes das HDL's a melhor representação era o esquemático;

2.7 Quais partes compõem o kit Altera DE2 – Therasic?

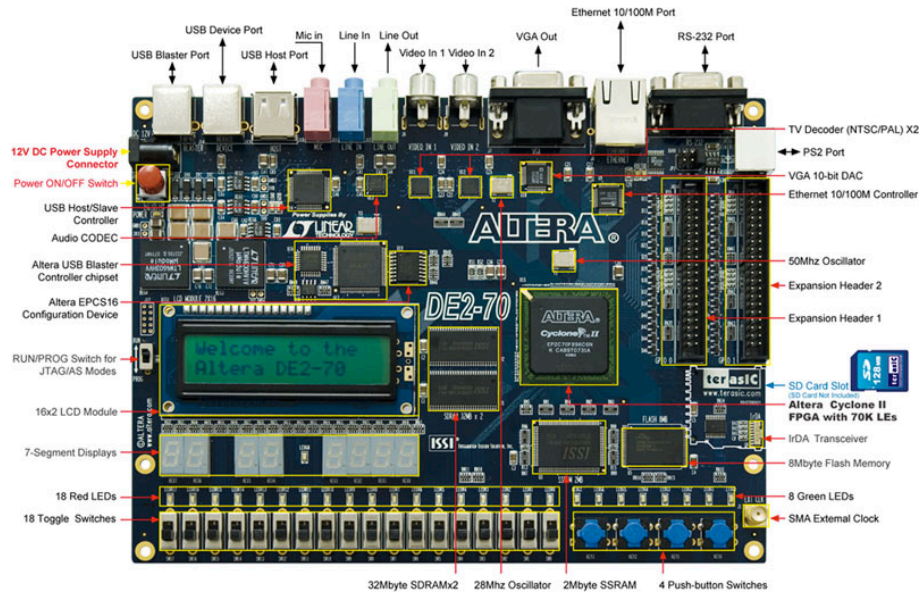


Figura 1: Componentes da FPGA Altera DE2

2.8 Quais partes do kit Altera DE2 – Therasic são usadas para gravação e depuração de sistemas Digitais?

USB Blaster na placa para gravação e JTAG para depuração do sistema

3 Parte Prática

Criamos um novo projeto no Quartus II e adicionamos o arquivo *somador.vhd* ao projeto e realizamos a compilação. Com a compilação finalizada, temos a visualização RTL abaixo:

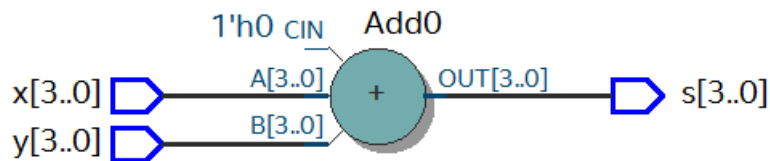


Figura 2: Visualização RTL do Somador

Com tudo verificado, fizemos a configuração do *testbench* e executamos a simulação pelo ModelSim. Com a simulação também funcional, podemos fazer o planejamento de quais pinos serão utilizados na placa real. Definimos que os pinos utilizados serão:

- Entradas X:
 - Chave SW0 - PIN_C10
 - Chave SW1 - PIN_C11
 - Chave SW2 - PIN_D12
 - Chave SW3 - PIN_C12
- Entradas Y:
 - Chave SW4 - PIN_A12
 - Chave SW5 - PIN_B12
 - Chave SW6 - PIN_A13
 - Chave SW7 - PIN_A14
- Saída S:
 - LED0 - PIN_A8
 - LED1 - PIN_A9
 - LED2 - PIN_A10
 - LED3 - PIN_B10

Com isso, realizamos a gravação da placa para os testes.

Com a placa gravada, realizamos o seguinte teste:

- Teste 1
 - Inserimos o valor '0011' na entrada X
 - Inserimos o valor '0001' na entrada Y
 - Esperamos que o valor na saída S seja '0100'

O resultado obtido foi esse:

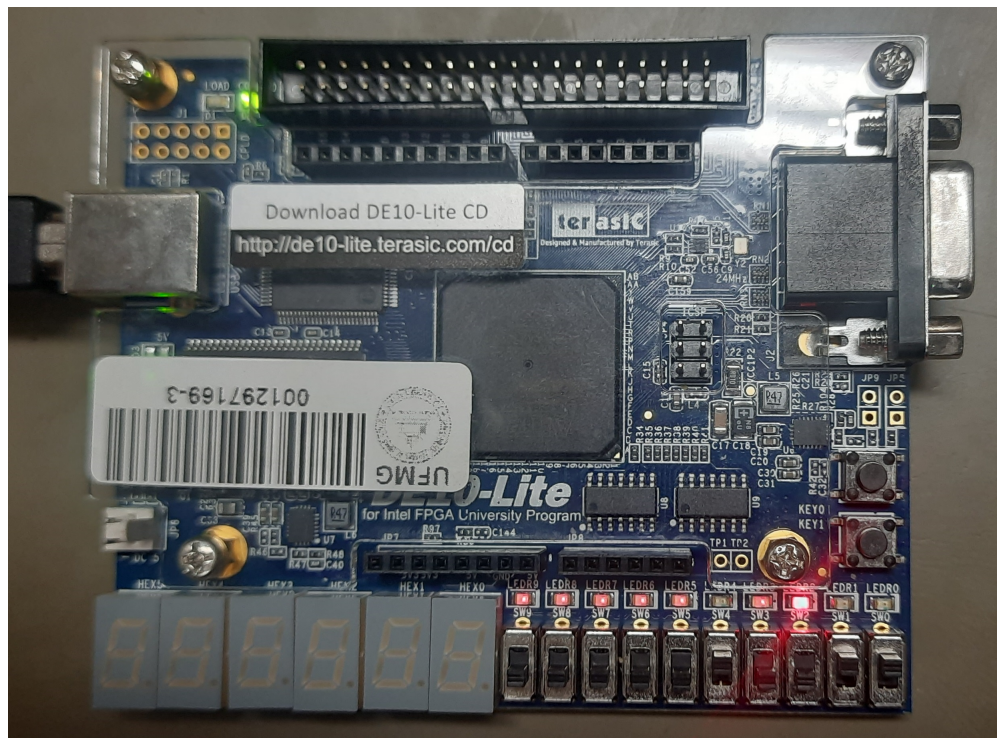


Figura 3: Resultado da implementação do somador na placa

4 Conclusão

Como podemos observar pela Figura 3, o resultado obtido foi igual ao resultado esperado, isto é, ao adicionarmos '3' e '1' obtivemos '4' na saída.