

CIRCUITOS ARITMÉTICOS PARTE 2

Júlia Pessoa Souza

Teoria de Eletrônica Digital-Turma C
Faculdade Gama - Universidade de Brasília
15/0133294
Laboratório 4

João Pedro Vergara Coneglian

Teoria de Eletrônica Digital-Turma C
Faculdade Gama - Universidade de Brasília
14/0145974
Laboratório 4

RESUMO

O experimento foi projetar e simular um circuito que fosse capaz de detectar uma condição de overflow (estouro de capacidade) para que fosse usado com um circuito somador de números com sinal, de três bits, codificados na forma de complemento de 2.

1. INTRODUÇÃO

Na Unidade Lógica e Aritmética de um hardware acontecem as operações lógicas. Essas operações são feitas por circuitos que recebem dados binários e podem somar, subtrair, multiplicar e dividir, de acordo com instruções da Unidade de Controle. Nesta experiência revisamos os circuitos somadores em VHDL.

Circuito meio somador (Half-Adder) tem entradas para dois dígitos binários, uma saída para a soma deles e uma saída para o dígito "carry (vai um)". É dito "meio" por conta da falta de entrada para o dígito "carry", ou seja, ele pode apenas iniciar uma soma, mas não pode continuar. Só pode ser usado para o par de bits menos significativos (mais à direita). Para cada um dos demais pares, deve existir entrada do (Cin), que recebe a saída (Cout) da soma do par anterior.

O somador completo (Full-Adder) possibilita a soma de 2 números binários de 1 bit mais a adição do carry anterior. Possui 3 bits de entrada ($A + B + \text{Cin}$) e 2 bits de saída (Soma + Cout).

2. EXPERIMENTO

2.1. Vivado

Foi utilizado o programa Vivado para escrever o algoritmo em VHDL. Criou-se um novo programa com a opção RTL project selecionada. foi escolhida a board Basys3 para fazer a implementação posteriormente. Então, criou-se o arquivo fonte em VHDL. Foram adicionadas as entradas A, B, Cin, sendo A e B os dois termos que serão somados e Cin o carry in. Foram adicionadas as saídas S e Cout, sendo S a soma dos dois termos e Cout o carry out. Na

arquitetura do programa foram escritos os comandos para definir S e Cout, como mostrado na Figura 1. Foi feita a simulação e forçados os clocks de A, B e Cin. A com 200ns, B com 100ns e Cin com 50ns. A simulação foi colocada para rodar por 400ns e o resultado encontra-se na Figura 3. Em seguida, foi criado outro arquivo para se fazer a soma de dois termos com 3 bits cada. A este arquivo foi dado o nome de som3bit. Foram gerados dois vetores de 3 componentes como entradas (Ai e Bi), que serão os termos da soma, e uma variável Cin para o carry in. Criou-se a saída Sum, também um vetor de 3 componentes e Cout para o carry out. Foram escritas linhas de código, como mostradas na Figura 2, e, foi feita uma função para chamar o código da soma de 1 bit e a partir disso foram escritos os comandos para definir cada bit do vetor Sum. Após os códigos, rodamos a simulação no programa. Forçou-se os clocks das variáveis Ai, Bi e Cin. Para Ai foi definido 200ns, para Bi, 100ns e para Cin, 50ns. A simulação foi configurada para rodar por 400ns, e, assim, foi possível ver todas as combinações de valores.

2.2. Figuras

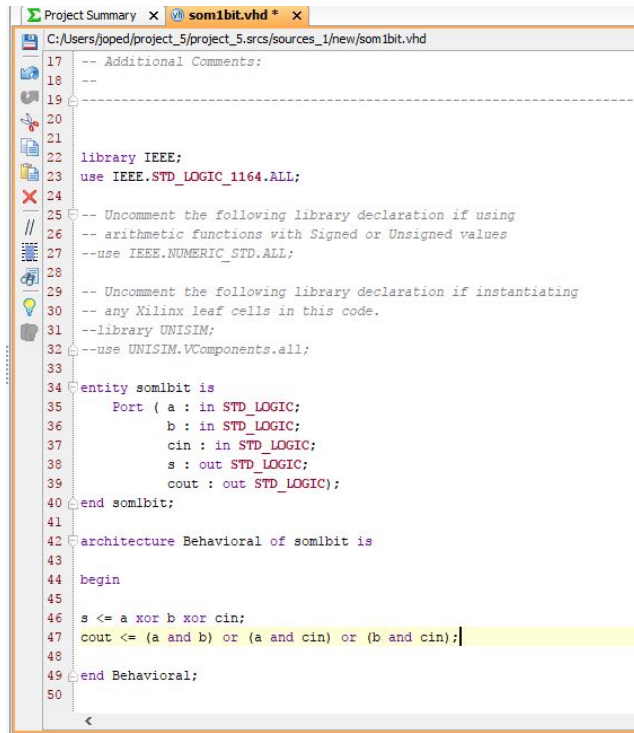


Figura 1: código som1bit

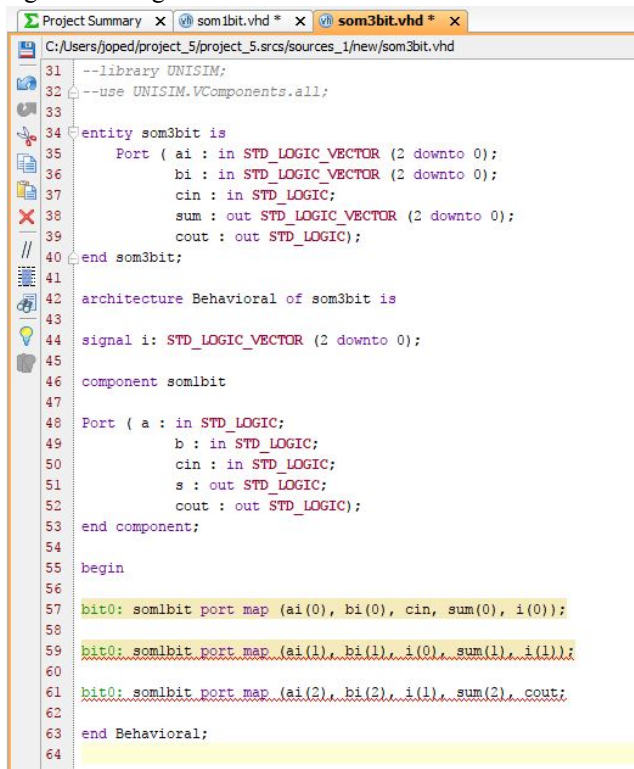


Figura 2: código som3bit.

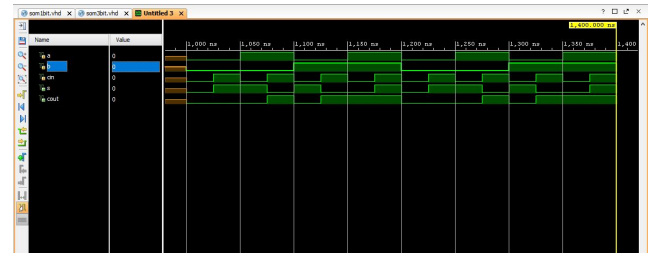


Figura 3: Simulação som1bit.

3. RESULTADOS

Na figura 1 foi feito o código para somar dois termos de 1 bit, o qual é necessário para o algoritmo que fará a soma de 2 termos de 3 bits, que está representado na figura 2. De acordo com a tabela verdade vista em sala de aula, é possível perceber as expressões booleanas de S e de Cout para somador de 1 bit, estas são escritas no código, o qual também é utilizado para somador de 3 bits. Na figura 3 pode-se observar os resultados da soma e do carry out na simulação da soma de 1 bit. Quando tem-se a soma de dois termos “1”, obtém-se resultado “0” e carry out “1”. Quando é feita a soma de “1” com “0”, obtém-se resultado “1” e carry out “0”. Já na soma de dois termos “0”, obtém-se resultado “0” e carry out “0”.

4. DISCUSSÃO E CONCLUSÕES

O experimento não obteve total sucesso ao mostrar os resultados esperados na simulação. A soma de 1 bit ocorreu como o esperado, com os valores teóricos e práticos sendo os mesmos, de acordo com o que foi visto em sala de aula e na tabela verdade. Porém, a soma de 3 bits não foi simulada corretamente. Os valores calculados teoricamente na soma de 3 bits não foram os mesmos do resultado gráfico proveniente da simulação. Não foi descoberto onde estava o erro, porque o VIVADO não apresentava nenhum alerta e não demonstrou sinais de erro.

5. REFERÊNCIAS

- [1] I. Idoeta e I. Valeije, “Elementos de Eletrônica Digital”, Érica Ltda, São Paulo, 2008.
- [2] R. Tocci, N.Widmer e G. Moss, “Sistemas Digitais: Princípios e aplicações” Pearson.
- [3] G. Bezerra, “Apostila Prática de Eletrônica Digital I”, 2017
- [4] G. Bezerra, “Tutorial de Simulação e Implementação no

Vivado”, 2017