Lógica Digital (1001351)

Experimento NR.6

Somador de 3 bits com *overflow* [1]

---

1 Instruções Gerais

- Grupos definidos no AVA, só incluir os nomes de quem efetivamente participou;
- Ler atentamente todo o procedimento desta experiência antes de realizá-la;

2 Objetivos da Prática

- Teste de um somador de 3 bits que sinalize *overflow*.
- Simulação que comprove seu correto funcionamento.
- Composição da hierarquia de módulos de modo a mostrar os resultados na placa.

3 Materiais, Equipamentos e Arquivos

- Altera Quartus II 13.0sp1;
- Kit Terasic DE0-CV.

4 Fundamentos teóricos

4.1 Módulo somador completo *(full-adder)*

Para realizar uma soma binária de múltiplos bits é necessário que cada bit seja computado por um somador completo, capaz de receber o *vem um* da soma anterior e de propagar o *vai um* para o próximo bit. Na Figura 1 é apresentado o circuito do módulo somador completo. Sua implementação em Verilog é apresentada no Código 1.

É possível construir um somador binário de um número arbitrário de bits encadeando vários somadores de um bit, conforme o esquema da Figura 2. Neste esquema, o *carry* do LSB é recebido como entrada e os demais propagados para os bits seguintes. Também é necessário calcular o *overflow*, quando o resultado da soma excede o campo de representação (no nosso caso, 3 bits). Esta saída pode ser apenas o último *carry* para números

[1] Revisão em 14 de maio de 2024: Prof. Mauricio Figueiredo e Prof. Ricardo Menotti.

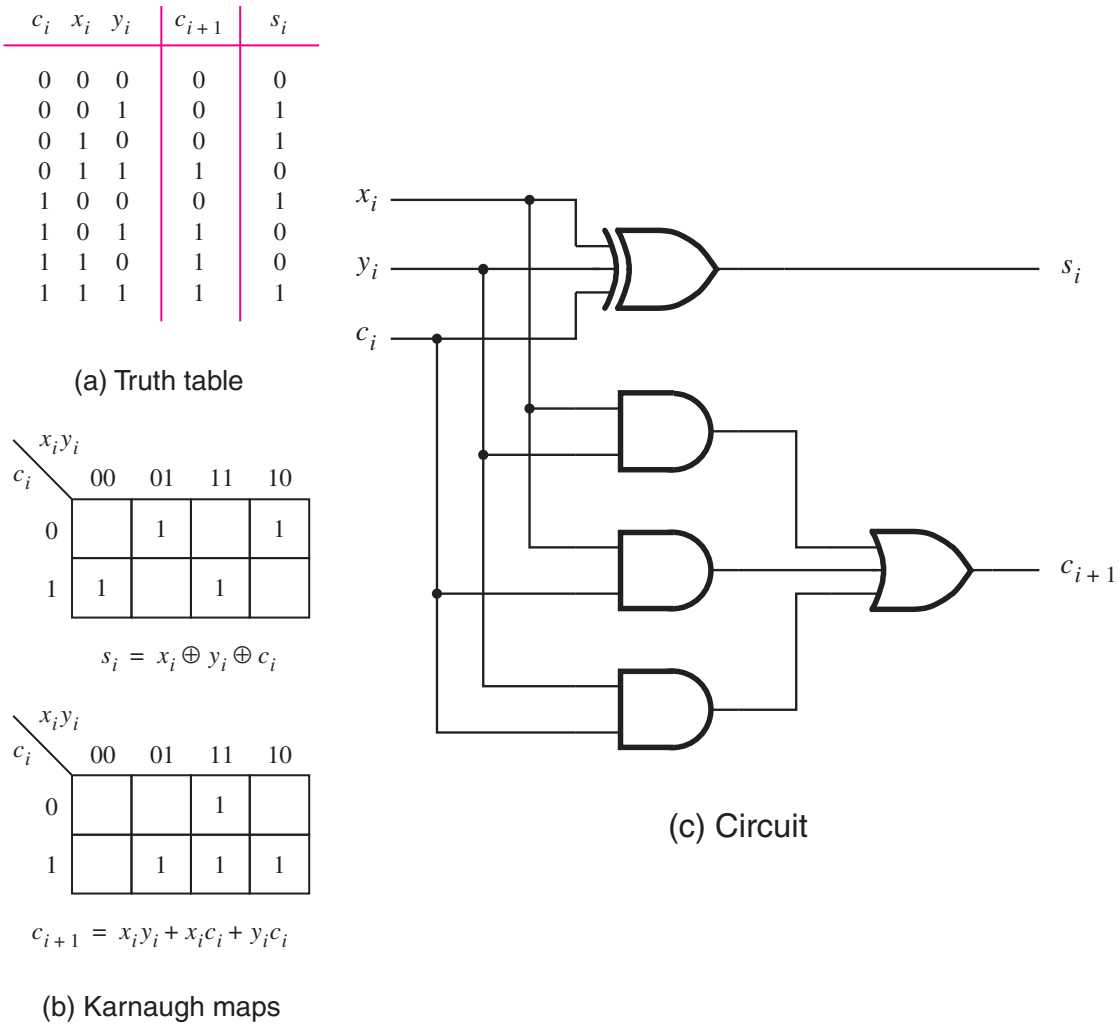

Figura 1: Módulo somador completo (*full-adder*).

```
1 module full_adder(
2     input Cin, X, Y,
3     output S, Cout);
4     assign S = X ^ Y ^ Cin;
5     assign Cout = (X & Y) | (Cin & X) | (Cin & Y);
6 endmodule
```

Código 1: Descrição em Verilog de um somador completo (*full-adder*).

sem sinal ou calculada por uma função XOR entre os dois sinais *carry* mais significativos, ou seja, $C_n \oplus C_{n-1}$, no caso de números com sinal em complemento de 2.

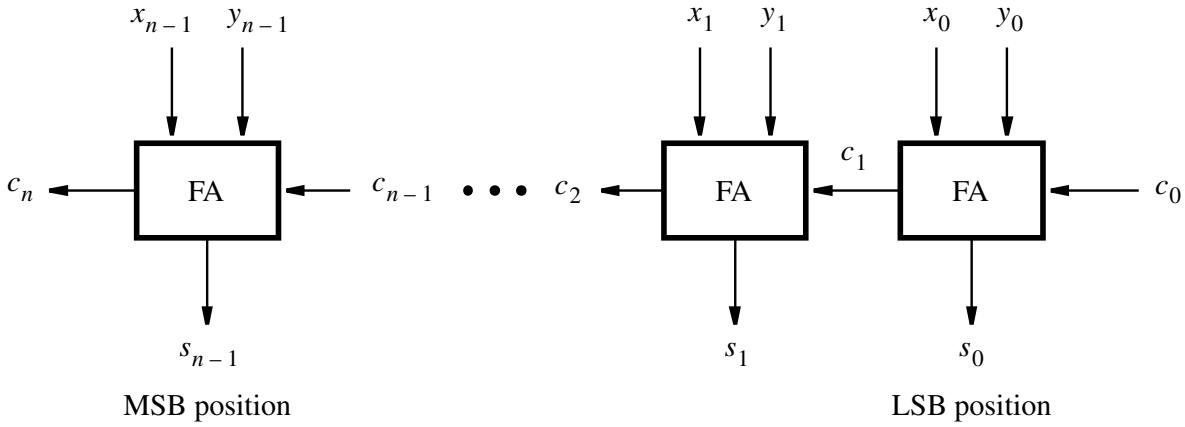

Figure 3.5 An n-bit ripple-carry adder.

Figura 2: Somandor de N bits

4.2 Display de 7 segmentos

O display de 7 segmentos, já conhecido, será utilizado para apresentador o resultado da soma na placa. Ele foi atualizado para decodificar em hexadecimal de 0 até F. Nosso campo de representação em 3 bits vai de 0 até 7 (sem sinal), então qualquer valor acima disso deve ser sinalizado como erro.

5 Procedimentos Experimentais

A partir do somador de 3 bits fornecido, testar seu funcionamento por simulação e depois na placa. Se a soma ficar dentro do campo de representação (de 0 até 7) ela deve ser mostrada no display 0. Se der *overflow*, no display 1 deve aparecer E, sinalizando o Erro.

Para isso, siga os seguintes passos:

1. Crie um projeto novo no Quartus utilizando a placa DE0-CV.
2. Importe os arquivos fornecidos e complete o que for necessário.
3. Faça a simulação do somador acrescentando outros casos de teste diferentes com/sem *carry* e com/sem geração de *overflow*. Note que até aqui o somador é top-level do projeto e que já existem casos de teste disponíveis.
4. Crie um novo arquivo para ser o top-level e fazer a interface com a placa.
5. Adicione o arquivo de restrição `DE0_CV.qsf` e faça as configurações necessárias para:
   (a) Cin seja informado no botão (KEY[0]);
   (b) A entrada X seja informada nos switches SW[2], SW[1], SW[0];

(c) A entrada Y seja informada nos switches SW[9], SW[8], SW[7];

6. Gere o arquivo de bitstream e grave no kit de desenvolvimento DE0-CV.
7. Envie apenas o arquivo *top-level* no AVA, incluindo os nomes dos participantes como comentário no início dele.

Etapa a): Se terminou durante a aula, chame o professor para corrigir;

Etapa b): Senão, envie também – como comentário no código – link para um vídeo demonstrando o correto funcionamento.