



Prof. Dr. Stefan Michael Blawid  
Departamento de Engenharia de Computação  
Centro de Informática

Tel. (81) 2126-8430 r: 4328

[sblawid@cin.ufpe.br](mailto:sblawid@cin.ufpe.br)

<https://sites.google.com/a/cin.ufpe.br/if817/>

## LE3: Síntese

### Fluxo do Projeto Digital & HDL (Aula 4-1 & 4-2)

1. **(1p)** Escreva o código Verilog para (a) um subtrator completo usando equações lógicas; (b) um subtrator de 4 bits usando o módulo definido em (a) como componente.
2. **(1p)** No código Verilog a seguir, A, B, C e D são 0 no tempo 10 ns. Se D mudar para 1 a 20 ns, especifique os horários em que A, B e C serão alterados e os valores que serão executados.

```
(a)
always @(D)
begin
    #5 A <= 1;
    B <= A + 1;
    #10 C <= B;
end
```

```
(b)
always @(D)
begin
    A <= #5 1;
    B <= A + 1;
    C <= #10 B;
end
```

3. **(1p)** O que há de errado com o código a seguir (a) para um meio somador que é ativado se o sinal "add" for igual a 1 e (b) para um MUX 4-para-1. Justifique sua resposta.

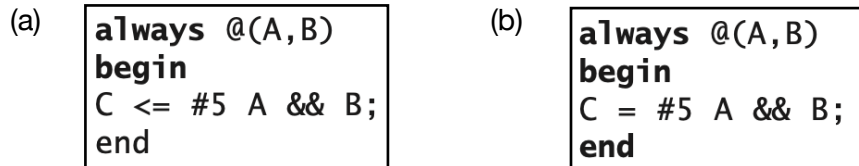
- (i) Ele será compilado, mas não será simulado corretamente.
- (ii) Compilará e simulará corretamente, mas não será sintetizado corretamente.
- (iii) Funcionará corretamente na simulação e na síntese.
- (iv) Não será compilado.

```
always @(x)
begin
    if (add == 1)
    begin
        sum = x ^ y;
        carry = x & y;
    end
    else
    begin
        sum = 0;
        carry = 0;
    end
end
```

```
reg [1:0]sel;
always @(A,B,I0,I1,I2,I3)
begin
    sel = 0;
    if(A == 1'b1)
        sel <= sel + 1;
    else
    begin
        end
    if(B == 1'b1)
        sel <= sel + 2;
    else
    begin
        end
    case(sel)
        0:begin
            F = I0;
        end
        1:begin
            F = I1;
        end
        2:begin
            F = I2;
        end
        3:begin
            F = I3;
        end
    endcase
end
```

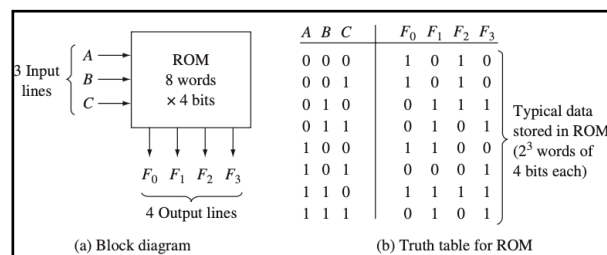
4. **(1p)** Supondo que B seja guiado pelo comando simulador:  
**force B 0 0, 1 4, 0 10, 1 15, 0 20, 1 30, 0 40**

Desenhe um diagrama de tempo ilustrando A, B e C se as seguintes instruções simultâneas forem executadas:



## Síntese e Otimização (Aula 4-3)

Uma memória somente leitura (ROM) consiste em uma matriz de dispositivos semicondutores que são interconectados para armazenar uma matriz de dados binários. Uma vez que os dados binários são armazenados na ROM, eles podem ser lidos sempre que desejado, mas os dados armazenados não podem ser alterados em condições normais de operação. A figura abaixo mostra uma ROM que possui três linhas de entrada e quatro linhas de saída.



Uma ROM pode implementar qualquer circuito combinacional. Essencialmente, se as saídas de todas as combinações de entradas são armazenadas na ROM, as saídas podem ser “consultadas” na tabela armazenada na ROM. O método ROM também é chamado de método Look-Up Table (LUT) por esse motivo.

5. **(1p)** Qual é o tamanho da menor ROM necessária para implementar o seguinte?

- (a) Um somador completo de 8 bits (assumir *carry-in* e *carry-out*); Um MUX 4-para-1; Um decodificador de 3-a-8  
 (b) Um conversor de BCD para binário (2 dígitos BCD); Um somador de 32 bits (adiciona dois números de 32 bits para obter uma soma de 33 bits); Um codificador de prioridade de 8-a-3

6. **(2p)** (a) Escreva um módulo Verilog para uma LUT com quatro entradas e três saídas. A saída de 3 bits deve ser um número binário igual ao número de 1s na entrada da LUT. Escreva uma bancada de teste para verificar o funcionamento correto do módulo e mostrar o resultado; (b) Escreva um módulo Verilog para um circuito que conte o número de 1s em um número de 12 bits. Use três dos módulos de (a) junto com operadores de adição sobrecarregados. Simule seu código e teste-o para as seguintes entradas de dados: 111111111111, 010110101101, 100001011100.

7. **(1p)** Escreva um módulo Verilog completo que realize as funções F e G usando uma ROM de palavras de 8 bits  $\times$  2. O LUT deve obrigatoriamente ser armazenado em um vetor de elementos de dois bits. As funções são dadas pelas equações:  
 (a)  $F = \bar{A}\bar{B} + B\bar{C}$  e  $G = AC + \bar{B}$ ; (b)  $(\bar{A}+\bar{B})(B+\bar{C})$  e  $G = (A+C)\bar{B}$

8. **(1p)** (a) Uma empresa está projetando um produto experimental que está na versão 1 agora. Espera-se que o produto passe por várias revisões. O plano da empresa é usar um FPGA para o design real. Que tipo de FPGA (SRAM ou antifuso) deve ser usado?  
 (b) Uma empresa está projetando um produto. Ele espera vender 100 milhões de cópias do produto. A empresa deve usar um MPGA ou um FPGA para este produto?

9. **(1p)** (a) Implemente a função  $F_1 = \bar{A}BC + \bar{B}C + ABC$  usando um FPGA com blocos lógicos programáveis que consistem em multiplexadores de 4-para-1. Suponha que as entradas e seus complementos estejam disponíveis, como na figura ao lado.

(b) Implemente a função  $F_1 = \bar{A}B + A\bar{B} + A\bar{C} + \bar{A}C$  usando um multiplexador. Qual é o tamanho do menor multiplexador necessário, assumindo que as entradas e seus complementos estão disponíveis?

