



## RELATÓRIOS - EXERCÍCIOS

### SD112 - Introdução a Verilog

DOCENTE Felipe Gustavo de Freitas **Rocha**

DISCENTE André Francisco Ribeiro **Bezerra**

DATA DE ENTREGA **07 de novembro de 2025 (prazo máximo)**

A-001	A-002	A-003
A-004	A-005	A-006 20 de outubro
A-007	A-008	A-009
A-010	A-011	A-012
A-013	A-014	

Formulário para envio das Atividades

## SUMÁRIO

REFERÊNCIAS	1
ANOTAÇÕES	2
A-001: Álgebra Booleana	3
A-002: Mintermos, Maxtermos e Mapas de Karnaugh	11
<b>A-003: O inversor</b>	<b>16</b>
A-004: Half Adder	17
A-005: Full Bit Adder	18
>> A-006: Declarações Processuais e Contínuas	19
A-007: Circuito simples de debounce	20
A-008: Reset Síncrono e Assíncrono	21
A-009: Estilos de Código	22
A-010: Descrição RTL	23
A-011: Descrição Comportamental	24
A-012: Descrição Estrutural	25
A-013: Primitivas	26
A-014: Codificação de Síntese vs Simulação	27

## REFERÊNCIAS

- [1] Digital Systems ; Authors, Ronald Tocci, Neal Widmer, Greg Moss ; Edition, 12 ; Publisher, Pearson Education, 2016 ; ISBN, 0134220145, 9780134220147;
- [2] Frank Vahid. 2010. Digital Design with RTL Design, Verilog and VHDL (2nd. ed.). Wiley Publishing.;



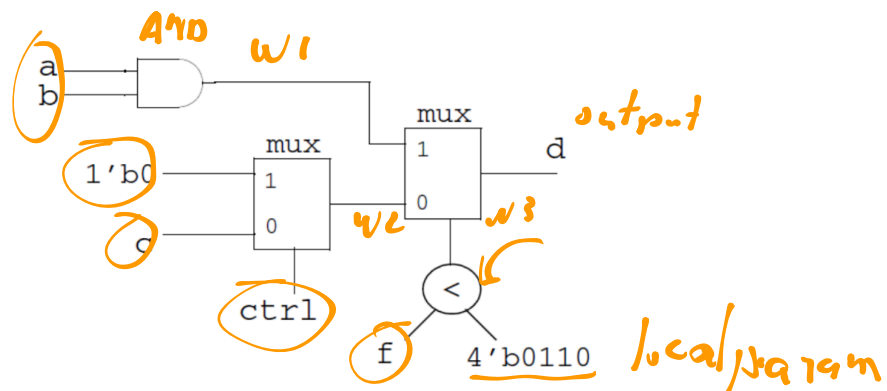


## A-006: Declarações Processuais e Contínuas

pts

### Conceitual

- 1 Se mais de uma condição de uma instrução *if...else* for verdadeira, qual condição tem prioridade?
- 2 As instruções *if*, *case* e *for* são contínuas ou processuais?
- 3 Codifique a seguinte lógica condicional:



### Prático

- 1 Implemente e simule um codificador binário que atenda aos seguintes requisitos:
  - > I. possui uma entrada de 4 bits codificada em binário que representará valores de 0x0 a 0xF,
  - > II. possui uma entrada de 1 bit para o segmento DP,
  - > III. possui uma saída de 8 bits que deve ser organizada da forma  $out[7:0] = [DP, A, B, C, D, E, F, G]$ . Os valores codificados no display serão apenas os valores com representação hexadecimal (números 0... 9, letras A... F). Qualquer outro valor deve mostrar o display inteiramente apagado.
- 2 Implemente um bloco que conte de 0x0 a 0xF para alimentar o bloco do exercício anterior. Sempre que a contagem atingir o valor 0xF, o sinal para o segmento DP deve ser invertido (de 0 para 1, ou de 1 para 0).

Handwritten notes for exercise 2:  $DP = DP \oplus H$  where  $H$  is the high bit of the 4-bit input. A truth table for the 4-bit input is shown: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. The high bit (H) is 0 for the first 8 values and 1 for the last 8 values.

A-006: Declarações Processuais e Contínuas

1- Em uma instrução *if...else (...)*, as condições são avaliadas sequencialmente top-down (de cima para baixo). A avaliação é baseada na confirmação positiva das condições. Em síntese de hardware tem's priority encoder



Onde a primeira condição na sequência tem a prioridade mais alta.

2- As instruções if, case e for são estritamente procedurais.

Só existem e ser utilizadas dentro de blocos procedurais: always, initial ou function.

Construções que controlam o fluxo de execução sequencial de um algoritmo. →  $\neq$  assign (concurrentes?)

3- Com base na estrutura de multiplexadores e portas lógicas do diagrama em Verilog corresponde a:

↖ ~~##~~ reference: 4'b0110

```
module conditionalLogic(
```

```
    input a, b, c, ctrl,
```

```
    input [3:0] f,
```

```
    output d
```

```
);
```

ternário

```
    assign d = (f < 4'b0110) ? (a & b) : (ctrl ? c : 1'b0);
```

```
endmodule
```

↖ reference

Practical notes

A-006: Declarações Processuais e Contínuas (prático) - código nos arquivos de suporte enviados no formulário

REPOSITÓRIO DO GITHUB

[github.com/ci-digital-inatel/SD112-INTRO-VERILOG](https://github.com/ci-digital-inatel/SD112-INTRO-VERILOG)

(QR Code scan)