



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
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
A-003: O inversor	16
A-004: Half Adder	20
A-005: Full Bit Adder	22
A-006: Declarações Processuais e Contínuas	25
A-007: Circuito simples de debounce	27
A-008: Reset Síncrono e Assíncrono	29
A-009: Estilos de Código	31
> A-010: Descrição RTL	33
A-011: Descrição Comportamental	35
A-012: Descrição Estrutural	37
A-013: Primitivas	39
A-014: Codificação de Síntese vs Simulação	41

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-010: Descrição RTL

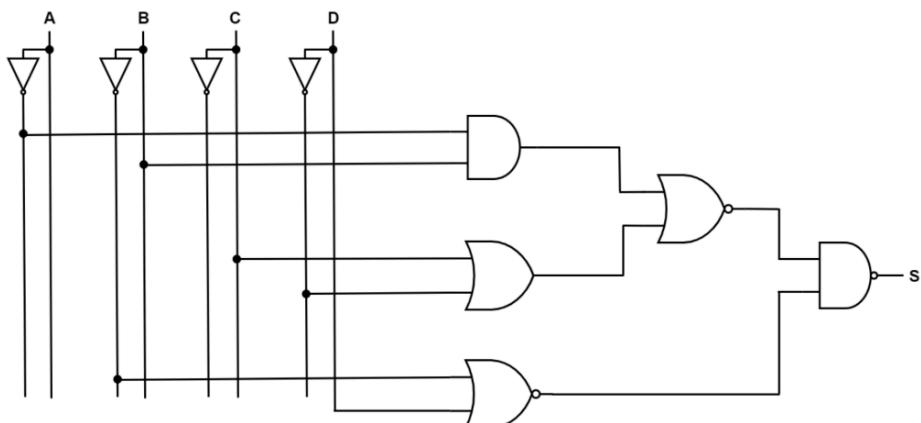
Page 221 / 332

Conceitual

- 1 O que é o RTL (Register Transfer Level) em Verilog?
- 2 Quais são os três principais níveis de abstração usados na descrição de circuitos?
- 3 Qual é o papel dos blocos *assign* na descrição RTL?
- 4 No exemplo do MUX 2x1, como o seletor (select) determina a saída (out)?

Prático

- 1 Implemente uma testbench capaz de avaliar o comportamento dos códigos fornecidos anteriormente.
- 2 Faça uma pesquisa e escreva como implementar o mesmo circuito através de outro operador da linguagem Verilog.
- 3 Escreva um código Verilog que represente o funcionamento do circuito combinacional da figura abaixo utilizando descrição RTL.



A-010: Descrição RTL (Conceitual)

1- RTL (Register Transfer Level) é um nível de abstração no design de hardware, e o principal estilo



de codificações em Verilog para estruturação de circuitos digitais síncronos.

As relações na programação baseada no fluxo de dados (TRANSFER) entre elementos de armazenamento (REGISTERS): flip-flops, e na lógica combinacional que processa os dados.

Este é o nível de descrição que as ferramentas de síntese utilizam para traduzir o comportamento do código em um circuito físico real.

2 - Os três principais níveis de abstração, do mais alto (abstrato) para o mais baixo (concreto) são:

Comportamental (behavioral): Descreve o que o circuito faz, ao tocar no algoritmo e na função, de forma similar a linguagem C. A vantagem é a não preocupação com a estrutura de hardware.

Fluxo de dados (RTL): Descreve a transferência de dados entre registradores e a lógica que os processa, geralmente sincronizada com o clock. É o nível padrão

A-010: Descrição RTL (prático)

> código nos arquivos de suporte enviados no formulário



REPOSITÓRIO DO GITHUB

github.com/ci-digital-inatel/SD112-INTRO-VERILOG



A-010: Descrição RTL (continuação)

para o design moderno e utilizado na síntese pelas ferramentas de EDA.

ESTRUTURAL (structural/gate-level): Descreve como o circuito é construído, ou especificar as conexões de forma explícita de componentes primitivos (portas AND, OR, NAND...) e/ou outros módulos, análoga a um esquemático.

3 - O papel da declaração ASSIGN (atribuição contínua) na descrição RTL é modelar lógica puramente combinatorial de forma concurrente. Define uma conexão direta e permanente onde a saída (tipo net, como wire) é atualizada instantaneamente sempre que qualquer um dos sinais na expressão de entrada se alterar.

Diferente do always, a declaração assign é usada fora das blocos procedurais para desenvolver hardware que não depende de um estado anterior ou de uma borda de clock. É a forma mais direta de expressar funções lógicas, multiplexadores, somadores e outras estruturas combinacionais que são reativas.

4 - No MUX 2x1, o seletor (select) funciona como uma chave digital que escolhe qual das duas entradas (A ou B) será passada para a saída (OUT). Um exemplo de implementações: $out = (A \& \neg select) | (B \& select);$

Em Verilog, podemos descrever a relação lógica com um operador ternário em uma atribuição contínua, que mapeia diretamente para a estrutura do multiplexador:

```
assign out = Select ? b : a;
```