

Aulas 0 e 1

Introdução a Circuitos Combinacionais

Autores

Gabriel A. F. Souza, Gustavo D. Colletta, Leonardo B. Zoccal, Odilon O. Dutra

Unifei

Histórico de Revisões

13 de fevereiro de 2025	1.0	Primeira versão do documento.
-------------------------	-----	-------------------------------

Tópicos

- Introdução
- Definição e Características
- Especificação Funcional
- Temporização e Glitches
- Exemplos de Aplicação
- Descrição de Circuitos Combinacionais
- Atividade Hands-on

Introdução

Estrutura de um Circuito

- Um circuito lógico é composto por:
 - Terminais de Entrada
 - Terminais de Saída
 - Especificação funcional
 - Especificação de temporização

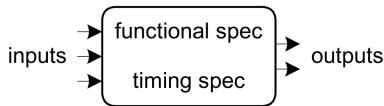


Figura 1: Diagrama de um circuito digital como uma “caixa-preta”

Hierarquia

- Internamente os circuitos são compostos por nós e elementos
- Cada elemento é também um circuito com entradas, saídas e especificações
- Um nó é um fio cuja tensão representa o valor discreto de uma variável.

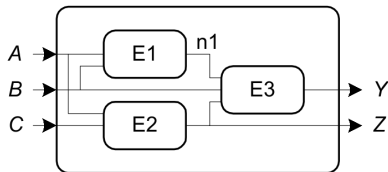


Figura 2: Elementos e nós de um circuito digital

Estrutura Interna

- Os nós são classificados como “entradas”, “saídas” ou “internos”
 - Entradas recebem valores do mundo exterior
 - Saídas entregam valores para o mundo exterior
 - Fios que não são nem entrada nem saídas são chamados de internos

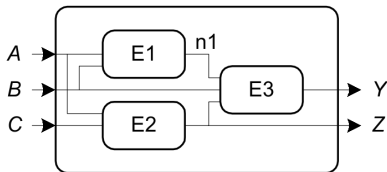


Figura 3: Elementos e nós de um circuito digital

Definição e Características

O que é um circuito combinacional?

- Circuitos Digitais são classificados em duas categorias:
 - Combinacional
 - Sequencial
- Em um circuito combinacional a saída depende somente dos valores atuais das entradas
 - Uma porta lógica é um exemplo de circuito combinacional

Combinacional vs. Sequencial

- Em um circuito sequencial a saída depende dos valores atuais e também dos valores passados das entradas.
- Circuitos combinacionais não possuem memória, circuitos sequenciais possuem memória.

Especificação Funcional

Expressões Booleanas

- Especificação Funcional das saídas em função das entradas (Equações Booleanas)
- Exemplo:
 - $S = F(A, B, C_{in})$
 - $C_{out} = F(A, B, C_{in})$

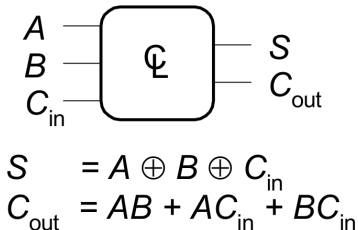


Figura 4: Especificação Funcional

Temporização e Glitches

Temporização

- Tempo de atraso entre alteração na entrada e alteração na saída.

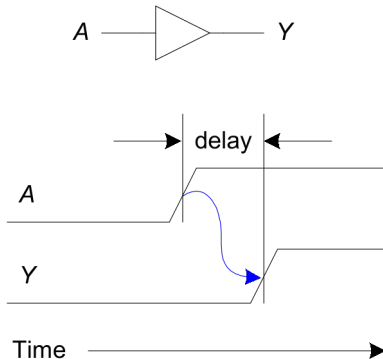


Figura 5: Tempo de Atraso

Propagação e Contaminação

- Tempo de Propagação: atraso máximo (tpd)
- Tempo de Contaminação: atraso mínimo (tcd)

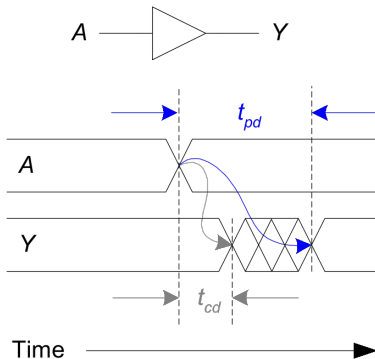


Figura 6: Atrasos de Propagação e Contaminação

Caminho Crítico

- Caminho com o maior tempo de atraso.

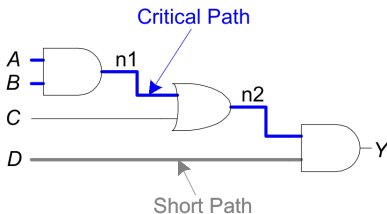
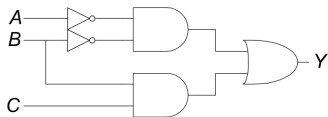


Figura 7: Caminho Crítico

Glitches

- Exemplo: O que acontece quando $A = 0$, $C = 1$ e B muda de 1 para 0?



Y C	AB			
	00	01	11	10
0	1	0	0	0
1	1	1	1	0

$$Y = \overline{A}\overline{B} + BC$$

Figura 8: Exemplo de um Glitch em Circuito Combinacional

Glitches

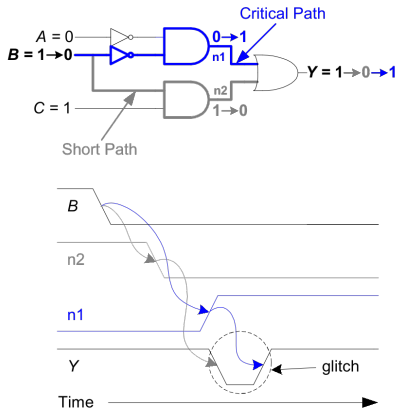


Figura 9: Exemplo de um Glitch em Circuito Combinacional

Possível Solução

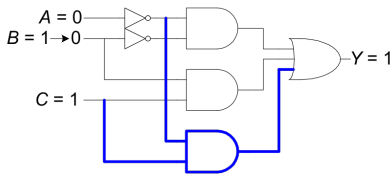
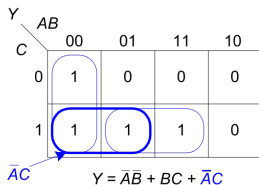


Figura 10: Exemplo de solução de um Glitch

Relevância

- Importante reconhecer glitches
- Nem sempre é possível remover todos os glitches
- Importante levar isso em consideração durante projeto
 - Exemplo: Convenções de circuitos sequenciais síncronos evitam problema com glitches nos blocos combinacionais

Exemplos de Aplicação

Exemplos de Circuitos Combinacionais

- Circuitos Lógicos
 - Expressões Lógicas
- Circuitos de Manuseio de Dados
 - Conversores de Código, Codificadores, Decodificadores
 - Multiplexadores e Demultiplexadores
- Circuitos Aritméticos
 - Somadores, Subtratores
 - Comparadores
 - Unidade Lógica Aritmética (ULA)
 - Multiplicadores

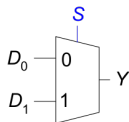
Descrição de Circuitos Combinacionais

Tipos de Descrição

- Circuitos combinacionais podem ser descritos usando diferentes tipos de descrição:
 - Fluxo de Dados
 - Estrutural
 - Comportamental

Fluxo de Dados

- Descrição por meio de “assign”
- Traduzir expressões booleanas
- Exemplo: Mux 2:1



S	D ₁	D ₀	Y	S	Y
0	0	0	0	0	D ₀
0	0	1	1	1	D ₁
0	1	0	0		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	1		
1	1	1	1		

Figura 11: Tabela Verdade de um Mux 2:1

Expressão Lógica

- Simplificação por mapa de Karnaugh
- Expressão Lógica

Y S	$D_0 D_1$			
	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$Y = D_0 \bar{S} + D_1 S$$

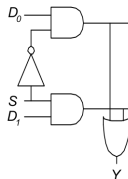


Figura 12: Expressão Lógica e Esquemático de um Mux 2:1

Expressão Lógica

```
1 module mux(a, b, sel, y);  
2   input a, b, sel;  
3   output y;  
4  
5   assign y = (!sel & a) | (sel & b);  
6  
7   endmodule
```

Circuito Inferido

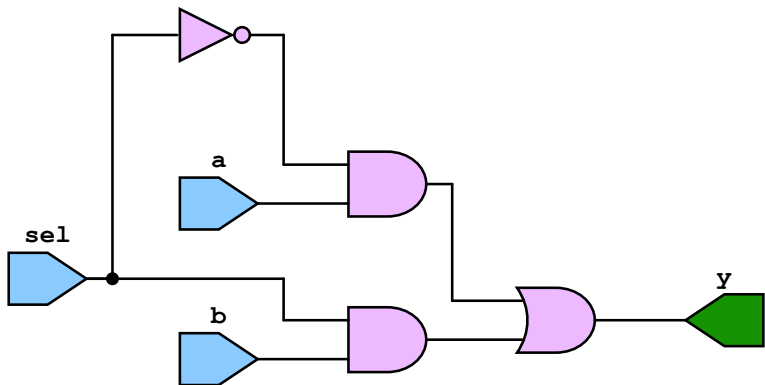


Figura 13: Circuito inferido com descrição utilizando expressão lógica

Atribuição Condicional

```
1 module mux2(a, b, sel, y);  
2   input a, b, sel;  
3   output y;  
4  
5   // assign saida = condicao ? if_true : if_false  
6  
7   assign y = sel ? a : b;  
8  
9   endmodule
```

Circuito Inferido

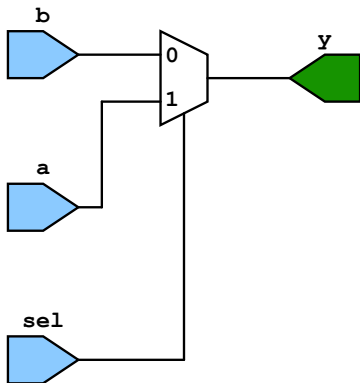


Figura 14: Circuito inferido com descrição utilizando atribuição condicional

Estrutural

- Descreve como vários módulos são interconectados
- Cada módulo contém outros módulos (instâncias)
- Descreve uma hierarquia

Primitivas

- Exemplo: Mux 2:1 utilizando primitivas

```
1 module mux3(a, b, sel, y);  
2   input a, b, sel;  
3   output y;  
4  
5   wire w1, w2;  
6  
7   or I1(y,w1,w2);  
8   and I2(w1,!sel,a);  
9   and I3(w2, sel,b);  
10  
11 endmodule
```

Circuito Inferido

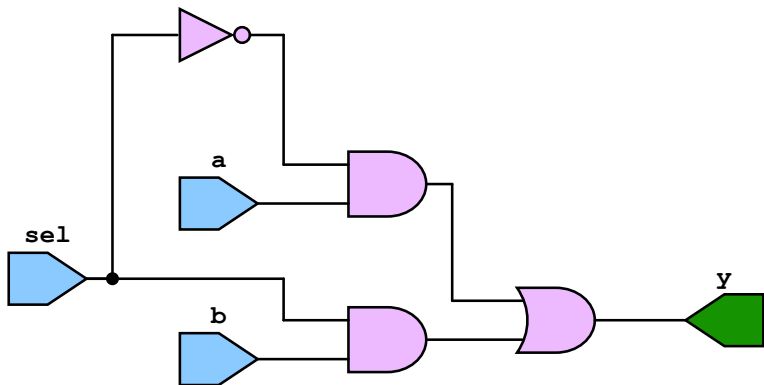


Figura 15: Circuito inferido com descrição utilizando primitivas

Comportamental

- O módulo contém uma descrição funcional do circuito
- Região de código sequencial
 - Permite descrição comportamental (mais abstrata)
 - Região em que os códigos são interpretados sequencialmente:
Bloco always
 - Processo paralelo assim como um “assign”
 - Contém um “programa” executado sempre que uma das entradas muda de valor
 - O “programa” é interpretado em tempo de compilação para gerar um circuito combinacional
 - Descreve a função computada pelo bloco

Bloco always combinacional

- always @ (lista_de_sinais)
 - bloco executa sempre que um dos sinais na lista muda de valor
 - risco de esquecer uma variável
- always @ (*)
 - bloco executa sempre que qualquer uma das entradas muda de valor
 - recomendado para circuitos combinacionais

Exemplo comportamental

- Descrição com bloco always usando if-else

```
1 module mux4(a, b, sel, y);  
2   input a, b, sel;  
3   output reg y;  
4  
5   always@(*) begin  
6       if(sel == 1'b0)  
7           y = a;  
8       else  
9           y = b;  
10  end  
11  
12 endmodule
```

Circuito Inferido

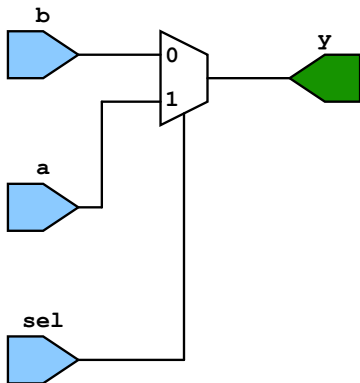


Figura 16: Circuito inferido com descrição utilizando descrição comportamental

Atividade Hands-on

Atividade 1

- Simular cada um dos tipos de descrição utilizados para implementar o mux 2:1
 - Modificar a descrição de forma a utilizar um vetor $D[1:0]$ ao invés das entradas “a” e “b”
- Analisar os resultados: Os circuitos sintetizados em cada caso estão corretos? E os resultados das simulações?

Testbench

- Como testar o funcionamento de um Mux 2:1?
 - Aplicar sinais com períodos diferentes a cada uma das entradas
 - Varrer os valores possíveis do sinal de seleção

Testbench

- Complete o código fornecido para realizar as simulações:

```
1 module mux_tb();
2 reg [1:0] D;
3 reg sel;
4 wire y;
5
6 //inclua aqui a declaracao da instancia do dispositivo sob teste
7 //
8
9 always begin #1 D[0] = !D[0]; end
10 always begin #2 D[1] = !D[1]; end
11 always begin #4 sel = !sel; end
12
13 initial begin
14     sel = 1'b0;
15     D = 2'd0;
16     #8 $stop;
17 end
18
19 endmodule
```

Atividade 2

- Implemente um módulo de um circuito comparador de 2 bits utilizando descrição por fluxo de dados.
- Especificações:
 - Duas entradas de dois bits
 - Uma saída de um bit
 - Saída igual a “1” quando as duas entradas forem exatamente iguais
 - Saída igual a “0” quando as entradas forem diferentes
- Crie um testbench para simular o circuito do comparador

Atividade 3

- Modifique o módulo do circuito comparador de 2 bits da atividade anterior para utilizar uma descrição comportamental.
- Simule utilizando o testbench elaborado anteriormente