Circuitos Aritméticos: Full Adder

Yuri Kaszubowski Lopes

UDESC

Anotações

Circuitos Combinacionais

- Circuitos no geral possuem várias entradas e várias saídas
- Circuito Combinacional
  - A **saída depende exclusivamente da entrada**
  - Exemplo de circuito que não é combinacional:
    - Circuito que possui uma memória interna, e a saída depende da **entrada** e do **estado atual da memória**
    - Chamado de circuito sequencial

Anotações

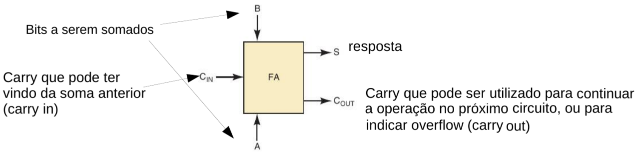
Circuitos Aritméticos

- Circuitos combinacionais comumente encontrados dentro da ALU
  - **ALU**: **A**rithmetic **L**ogic **U**nit (Unidade Lógica e Aritmética)
- Compreende circuitos somadores, deslocadores de bits, operadores lógicos, ...
- Como exemplo, veremos a implementação de um circuito somador simples
  - Assumiremos que os valores somados são sempre positivos
  - No entanto, o somador ainda é válido para representações em complemento de dois, com alguns poucos ajustes

Anotações

Full Adder

- Full Adder (Somador Completo)
- Circuito somador para 1 bit
- Considera o carry ("vai um")
- A soma de dois bits pode gerar um carry, e.g., $1_2 + 1_1$
- Esse carry deve ser considerado no próximo adder, caso ele exista



Bloco representando um Somador. Fonte: Tocci et al.

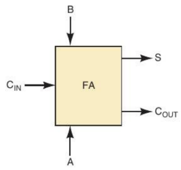
Anotações

Full Adder

Tabela Verdade

- Faça a tabela verdade para o Full Adder

Entrada			Saída	
A	B	C_{in}	S	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



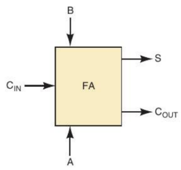
Anotações

Full Adder

Tabela Verdade

- Faça a tabela verdade para o Full Adder

Entrada			Saída	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Anotações

Full Adder

Montando a expressão Booleana

- Como temos duas saídas, vamos definir cada uma separadamente
- Começando por S, temos quatro entradas que geram 1
- Qual a expressão inicial pela Pela soma dos produtos?
  - $S = \overline{A}.\overline{B}.C_{in} + \overline{A}.B.\overline{C_{in}} + A.\overline{B}.\overline{C_{in}} + A.B.C_{in}$

Entrada			Saída	
A	**B**	**C_{in}**	**S**	**C_{out}**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Anotações

Full Adder

Montando a expressão Booleana

- Simplificando
- $S = \overline{A}.\overline{B}.C_{in} + \overline{A}.B.\overline{C_{in}} + A.\overline{B}.\overline{C_{in}} + A.B.C_{in}$
- Fatorando $\overline{A}$ nos dois primeiros termos e A nos dois últimos (distributiva^{-1})
  - $S = \overline{A}.(\overline{B}.C_{in} + B.\overline{C_{in}}) + A.(\overline{B}.\overline{C_{in}} + B.C_{in})$
- Utilizando uma tabela de equivalências lógicas: XOR e XNOR:
  - $S = \overline{A}.(\underbrace{\overline{B}.C_{in} + B.\overline{C_{in}}}_{B \oplus C_{in}}) + A.(\underbrace{\overline{B}.\overline{C_{in}} + B.C_{in}}_{\overline{B \oplus C_{in}}})$
  - $S = \overline{A}.(B \oplus C_{in}) + A.(\overline{B \oplus C_{in}})$
- Considere que $X = B \oplus C_{in}$, então:
- $S = \overline{A}.X + A.\overline{X} = A \oplus X$ (propriedade do XOR)
- Substituindo X pela expressão original:
  - $S = A \oplus (B \oplus C_{in})$

Anotações

Full Adder

Montando a expressão Booleana

- Calculando C_{out}, temos quatro entradas que geram 1
- Qual a expressão inicial pela Pela soma dos produtos?
  - $C_{out} = \overline{A}.B.C_{in} + A.\overline{B}.C_{in} + A.B.\overline{C_{in}} + A.B.C_{in}$

Entrada			Saída	
A	**B**	**C_{in}**	**S**	**C_{out}**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Anotações

Full Adder

Montando a expressão Booleana

- Simplificando
- $C_{out} = \overline{A}.B.C_{in} + A.\overline{B}.C_{in} + A.B.\overline{C_{in}} + A.B.C_{in}$
- Podemos repetir qualquer mintermo sem alterar o resultado
  - Lembre-se: $A + A = A$ (Soma lógica)
  - $C_{out} = \overline{A}.B.C_{in} + A.\overline{B}.C_{in} + A.B.\overline{C_{in}} + A.B.C_{in} + A.B.C_{in} + A.B.C_{in}$
  - No que isso nos ajuda?
    - Agora temos termos em comum em pares de mintermos, o que nos ajuda a fatorar
  - $C_{out} = \overline{A}.B.C_{in} + A.B.C_{in} + A.\overline{B}.C_{in} + A.B.C_{in} + A.B.\overline{C_{in}} + A.B.C_{in}$
- Fatorando:
  - $C_{out} = B.C_{in}.(\overline{A} + A) + A.C_{in}.(\overline{B} + B) + A.B.(\overline{C_{in}} + C_{in})$
- Soma lógica ($X + \overline{X} = 1$):
  - $C_{out} = B.C_{in}.1 + A.C_{in}.1 + A.B.1$
  - $C_{out} = B.C_{in} + A.C_{in} + A.B$ (Multiplicação lógica)

Anotações

Full Adder

Montando a expressão Booleana

- $S = A \oplus (B \oplus C_{in})$
- $C_{out} = B.C_{in} + A.C_{in} + A.B$

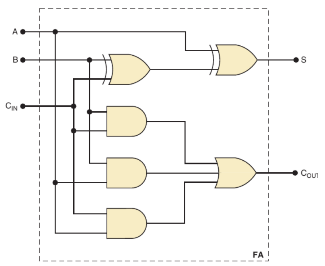
Entrada			Saída	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Anotações

Full Adder

Montando a expressão Booleana

- $S = A \oplus (B \oplus C_{in})$
- $C_{out} = B.C_{in} + A.C_{in} + A.B$
- Montar o circuito



Full Adder. Fonte: Tocci et al.

Anotações

Exercícios

1. Faça a tabela verdade para o circuito do slide anterior e veja que ele realmente possui o comportamento esperado pelo full adder
2. O full adder do slide anterior:
   - Soma 2 bits (A e B) e um carry (C_{in}) e
   - Gera uma saída de 1 bit (S) + carry (C_{out})

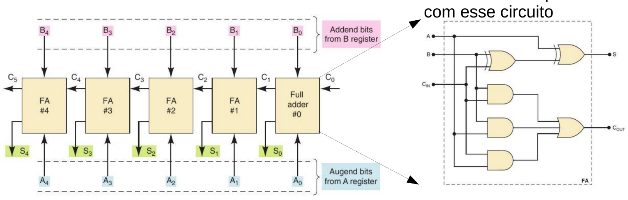
   Então:
   1. Monte um circuito para somar dois bits com outros dois (Exemplo: $11 + 01 \Leftrightarrow AB + CD$)
   2. Mostre o circuito com portas lógicas
   3. Note que o bit menos significativo nunca recebe um carry. Como você vai resolver esse problema com o full adder?

Anotações

Full Adder: Limitação

- Veja a implementação (em blocos) de um somador para 5 bits
  - Onde está a limitação (em tempo) desse circuito?
  - Precisamos esperar os carrys
  - O carry C_1 precisa passar para FA#1, que vai gerar C_2, que será necessário em FA#2, ... até termos o carry final C_5
  - É especialmente custoso, principalmente em somadores grandes (e.g. 64-bits)
    - Esse problema é conhecido como carry propagation (Propagação do carry)

Cada bloco é implementado com esse circuito



Fonte: Tocci et al.

Anotações

Carry Propagation

- Circuitos de alta velocidade devem tratar esse problema
  - Esquemas chamados **carry-lookahead** são adicionados
  - Predizem se determinada operação vai gerar um carry ou não, sempre precisar passar por cada um dos Full Adders
- Requerem uma grande quantidade extra de portas lógicas para serem implementados
- Na disciplina nos contentaremos com o Full Adder sem lookahead
  - Implementações de circuitos de carry-lookahead podem ser encontradas na literatura da disciplina

Anotações

Circuitos integrados

- Existem diversos circuitos integrados que implementam o circuito discutido em aula
  - 7483: Somador completo de 4 bits
  - CD4008: Somador completo de 4 bits com carry look ahead

Anotações

Dica

- Boa parte do conteúdo desta aula foi baseado em Tocci et al (2016)
- Em Bignell e Donovan (2010) o mesmo circuito é criado, mas utilizando uma abordagem diferente
- No livro existe também a implementação do half-adder
- Leia a implementação apresentada no livro caso você tenha tido alguma dificuldade com a abordagem utilizada em sala

Anotações

Exercícios

3. Faça novamente o adder para soma de dois bits com dois bits (AB + CD)
   - Para o bit menos significativo, que não recebe carry, crie um circuito de soma que **não recebe** C_{in} algum para o cálculo
   - O nome desse circuito é **half-adder**
   - Tente definir você mesmo esse circuito
     - Sua implementação também está disponível na literatura

Anotações

Referências

- TOCCI, R.J.; WIDMER,N.S. **Sistemas digitais: princípios e aplicações**. 11a ed, Prentice-Hall, 2011.
- RUGGIERO, M.; LOPES, V. da R. **Cálculo numérico: aspectos teóricos e computacionais**. Makron Books do Brasil, 1996.
- NULL, L.; LOBUR, J. **Princípios Básicos de Arquitetura e Organização de Computadores**. 2014. Bookman, 2009. ISBN 9788577807666.

Anotações

Anotações

Anotações