

**EEL7020 – Sistemas Digitais**  
**Aula 5: Circuitos combinacionais -**  
**Circuitos Aritméticos**

Prof. Djones Vinicius Lettnin  
lettnin@eel.ufsc.br  
<http://lettnin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the authors E. Batista, E. Bezerra e J. Güntzel.

**Revisão**

**- Multiplexadores (ou seletores)**

- Multiplexador 4:1 (ou seletor 4:1) Implementação com portas lógicas básicas

$$Y = \text{sel1} \cdot \text{sel0} \cdot A + \text{sel1} \cdot \text{sel0} \cdot B + \text{sel1} \cdot \text{sel0} \cdot C + \text{sel1} \cdot \text{sel0} \cdot D$$

© J. Güntzel – Adapted by D. Lettnin

2

**Revisão**

**- Demultiplexadores**

- Exemplo: Demux de 8 saídas

SELECT code	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>	O <sub>5</sub>	O <sub>6</sub>	O <sub>7</sub>
0 0 0	0	0	0	0	0	0	0	0
0 0 1	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	1	0	0	0
0 1 1	0	0	0	0	0	1	0	0
1 0 0	0	0	0	1	0	0	0	0
1 0 1	0	0	0	1	0	0	0	0
1 1 0	0	0	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0

© E. Batista – Adapted by D. Lettnin

**Revisão**

**- Decodificadores**

Decodificador 2:4 Tabela-verdade e símbolo

entradas	saidas
A1 A0	S0 S1 S2 S3
0 0	1 0 0 0
0 1	0 1 0 0
1 0	0 0 1 0
1 1	0 0 0 1

© J. Güntzel – Adapted by D. Lettnin

4

**Revisão**

**- Decodificadores**

Decodificador 2:4

- Cada uma das 4 saídas corresponde a um mintermo diferente.

entradas	saidas
A1 A0	S0 S1 S2 S3
0 0	1 0 0 0
0 1	0 1 0 0
1 0	0 0 1 0
1 1	0 0 0 1

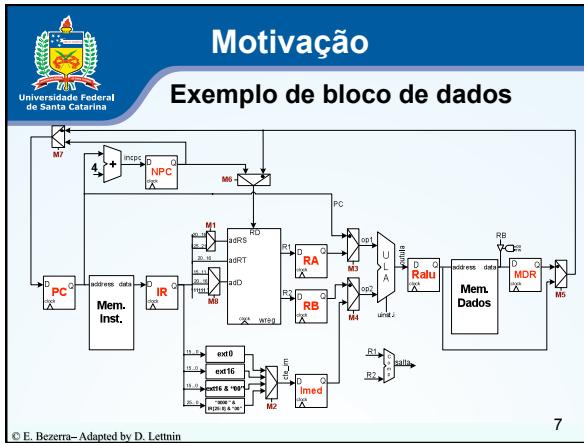
Implementação independente de tecnologia

© J. Güntzel – Adapted by D. Lettnin

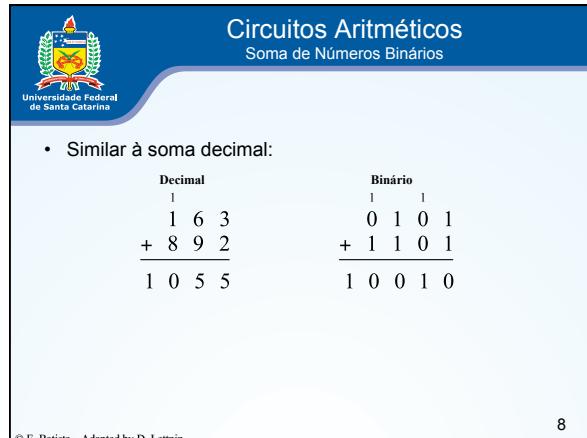
**Plano de Aula**

- Circuitos Aritméticos
  - Somadores
  - Subtratores
- Somadores paralelos
  - Carry ripple, Carry look-ahead, Carry select

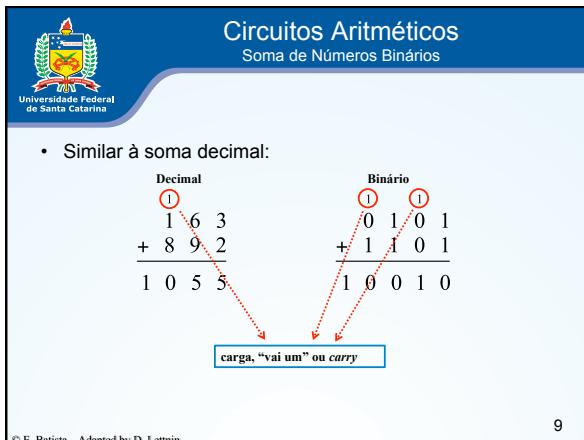
6



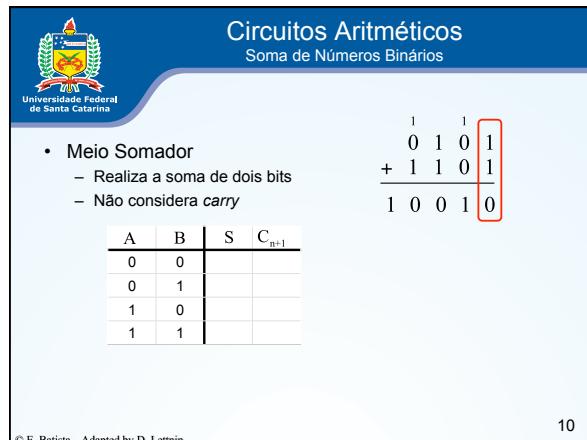
7



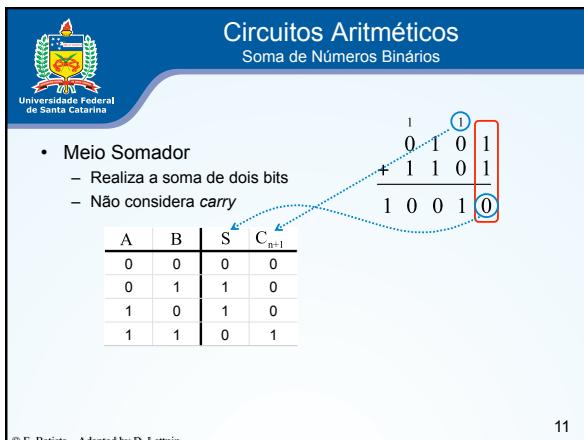
8



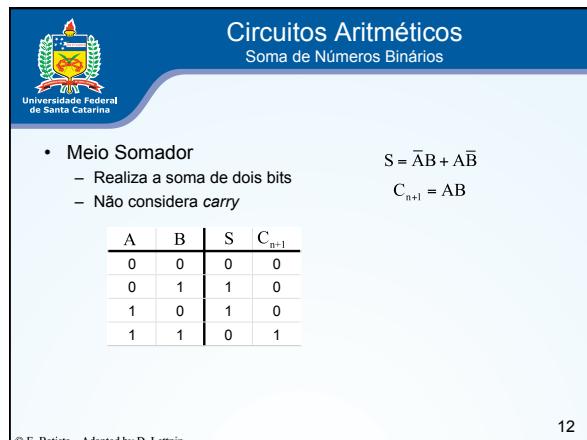
9



10



11



12

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Meio Somador**
  - Realiza a soma de dois bits
  - Não considera carry

A	B	S	C <sub>n+1</sub>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C_{n+1} = AB$$

13

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Completo**
  - Realiza a soma de dois bits,

A	B	C <sub>n</sub>	S	C <sub>n+1</sub>
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

$$\begin{array}{r} 1 \\ 0 \ 1 \ 0 \ 1 \\ + 1 \ 1 \ 0 \ 1 \\ \hline 1 \ 0 \ 0 \ 1 \end{array}$$

14

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Completo**
  - Realiza a soma de dois bits,

A	B	C <sub>n</sub>	S	C <sub>n+1</sub>
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

$$\begin{array}{r} 1 \\ 0 \ 1 \ 0 \ 1 \\ + 1 \ 1 \ 0 \ 1 \\ \hline 1 \ 0 \ 0 \ 1 \ 0 \end{array}$$

15

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Completo**
  - Realiza a soma de dois bits,

A	B	C <sub>n</sub>	S	C <sub>n+1</sub>
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

$$S = A \oplus (B \oplus C)$$

$$C_{n+1} = A(B \oplus C) + BC$$

16

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo**
  - Soma de números de vários bits

17

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo**
  - Soma de números de vários bits
    - Exemplo: números de 4 bits

$$\begin{array}{r} a_3 \ a_2 \ a_1 \ a_0 \\ + b_3 \ b_2 \ b_1 \ b_0 \\ \hline r_4 \ r_3 \ r_2 \ r_1 \ r_0 \end{array}$$

18

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - Soma de números de vários bits
  - Exemplo: números de 4 bits

$$\begin{array}{r}
 & a_3 & a_2 & a_1 & a_0 \\
 + & b_3 & b_2 & b_1 & b_0 \\
 \hline
 r_4 & r_3 & r_2 & r_1 & r_0
 \end{array}$$

© F. Batista – Adapted by D. Lettini

19

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - Soma de números de vários bits
  - Exemplo: números de 4 bits

$$\begin{array}{r}
 & a_3 & a_2 & a_1 & a_0 \\
 + & b_3 & b_2 & b_1 & b_0 \\
 \hline
 r_4 & r_3 & r_2 & r_1 & r_0
 \end{array}$$

© F. Batista – Adapted by D. Lettini

20

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - De maneira geral:

© F. Batista – Adapted by D. Lettini

21

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - De maneira geral:

– Considerando número fixo de bits para saída igual ao número de bits de entrada, **overflow** é indicado por  $r_{n+1}$

© F. Batista – Adapted by D. Lettini

22

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - Outra forma:

© F. Batista – Adapted by D. Lettini

23

**Circuitos Aritméticos**  
Soma de Números Binários

Universidade Federal de Santa Catarina

- Somador Binário Paralelo
  - Outra forma:

– Desvantagem: custo ligeiramente maior.

© F. Batista – Adapted by D. Lettini

24

**Plano de Aula**

Universidade Federal de Santa Catarina

- Circuitos Aritméticos
  - Somadores
  - Subtradores
- Somadores paralelos
  - Carry ripple, Carry look-ahead, Carry select

25

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Uma primeira idéia seria projetar um subtrator completo
  - Conseqüência: 2 circuitos, um para soma e outro para subtração
- Alternativa mais eficaz: usar repr. em **complemento de 2**
  - Somas de números com sinais diferentes são feitas diretamente

© F. Batista – Adapted by D. Letmin

26

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Representação em complemento de 2
  - Exemplos: +9 e -9 representados com 8 bits

<b>sinal</b> $+9 = 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$ $-9 = 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1$ <b>Magnitude</b>
--

© I. Guentzel – Adapted by D. Letmin

27

**Complemento 2**

Universidade Federal de Santa Catarina

- Método de representação de números negativos com representações complemento 2 com **d bits** (dígitos binários):

Intervalo de valores:  $[-(2^{d-1}); +(2^{d-1}-1)]$

Negativo → Positivo

© J. Fraga – Adapted by D. Letmin

28

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Representação em complemento de 2
  - Exemplos: +9 e -9 representados com 8 bits

Sinal	Regra de formação	Exemplo
positivos	= sinal-magnitude	$+9 = 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$
negativos	1. Toma-se a representação em sinal-magnitude 2. Inverte-se o número, bit a bit 3. Soma-se 1	$+9 = 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$ $1\ 1\ 1\ 1\ 0\ 1\ 1\ 0$ $1\ 1\ 1\ 1\ 0\ 1\ 1\ 1 = -9$

© I. Guentzel – Adapted by D. Letmin

29

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Mas, por que em **complemento de 2** podemos somar números positivos e negativos diretamente?
  - Exemplificando com 5 bits:

$\begin{array}{r} 3 \\ + 9 \\ \hline 12 \end{array}$	$\begin{array}{r} 0\ 0\ 0\ 1\ 1 \\ + 0\ 1\ 0\ 0\ 1 \\ \hline 0\ 1\ 1\ 0\ 0 \end{array}$
--	---

© F. Batista – Adapted by D. Letmin

30

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Mas, por que em **complemento de 2** podemos somar números positivos e negativos diretamente?
- Exemplificando com 5 bits:

$$\begin{array}{r} 3 \\ + 9 \\ \hline 12 \end{array} \quad \begin{array}{r} 0\ 0\ 0\ 1\ 1 \\ + 0\ 1\ 0\ 0\ 1 \\ \hline 0\ 1\ 1\ 0\ 0 \end{array}$$

$$\begin{array}{r} 9 \\ - 3 \\ \hline 6 \end{array} \quad \begin{array}{r} 9 \\ + (-3) \\ \hline 6 \end{array} \quad \begin{array}{r} 0\ 1\ 0\ 0\ 1 \\ + 1\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 0 \end{array}$$

31 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Mas, por que em **complemento de 2** podemos somar números positivos e negativos diretamente?
- Exemplificando com 5 bits:

$$\begin{array}{r} 4 \\ - 7 \\ \hline - 3 \end{array} \quad \begin{array}{r} 4 \\ + (-7) \\ \hline - 3 \end{array} \quad \begin{array}{r} 0\ 0\ 1\ 0\ 0 \\ + 1\ 1\ 0\ 0\ 1 \\ \hline 1\ 1\ 1\ 0\ 1 \end{array}$$

$$\begin{array}{r} 2 \\ - 8 \\ \hline - 10 \end{array} \quad \begin{array}{r} (-2) \\ + (-8) \\ \hline - 10 \end{array} \quad \begin{array}{r} 1\ 1\ 1\ 1\ 0 \\ + 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 0\ 1\ 1\ 0 \end{array}$$

32 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Princípio da subtração de números binários:
  - Usar representação em complemento de 2 e fazer:

$$X - Y = X + (-Y)$$

adição      número negativo

33 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

$X - Y = X + (-Y)$

↓

$X - Y = X + (\text{complemento de 2 de } Y)$

34 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Uma forma de determinar o **complemento de 2**:
  - Obter o complemento de 1 do número positivo (inversão de todos os bits)
  - Somar 1

35 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Uma forma de determinar o **complemento de 2**:
  - Obter o complemento de 1 do número positivo (inversão de todos os bits)
  - Somar 1

36 © F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Funciona também para transformar números negativos em positivos

37

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Mas, como fazer estrutura abaixo?

$X - Y = X + (-Y)$  a partir da estrutura abaixo:

38

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Mas, como fazer estrutura abaixo?

$X - Y = X + (-Y)$  a partir da estrutura abaixo:

39

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Subtração de Números Binários

Universidade Federal de Santa Catarina

- Circuito para subtração:

40

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

- Somador:

- Subtrator:

41

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

- Somador:

- Subtrator:

42

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

- Para obter um somador/subtrator, precisamos de um inverter que possa ser ativado e desativado

43

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

- Para obter um somador/subtrator, precisamos de um inverter que possa ser ativado e desativado

C control	E inativa	S saída
0	0	0
0	1	1
1	0	1
1	1	0

44

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

- Para obter um somador/subtrator, precisamos de um inverter que possa ser ativado e desativado

C control	E inativa	S saída
0	0	0
0	1	1
1	0	1
1	1	0

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

XOR funciona como um inverter controlado.

45

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Somador/Subtrator Paralelo

Universidade Federal de Santa Catarina

Se  $C = 0 \rightarrow R = X + Y$   
Se  $C = 1 \rightarrow R = X - Y$

46

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- Overflow ocorre quando o resultado de uma operação é maior (ou menor) do que o valor máximo (ou mínimo) que pode ser representado com um determinado número de bits

47

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- Overflow ocorre quando o resultado de uma operação é maior do que o valor máximo (ou menor do que o valor mínimo) que pode ser representado com um determinado número de bits
  - Exemplo: 5 bits  $\rightarrow 2^5$  valores diferentes

Em complemento de 2: de -16 até 15  
No entanto,

$10 + 7 = 17$   
 $-11 - 9 = -20$

48

© F. Batista – Adapted by D. Lettin

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- **Overflow** ocorre quando o resultado de uma operação é maior do que o valor máximo (ou menor do que o valor mínimo) que pode ser representado com um determinado número de bits
  - Exemplo: 5 bits  $\rightarrow 2^5$  valores diferentes

Em complemento de 2: de **-16 até 15**

No entanto,

$$\begin{array}{r} 10 \\ + 8 \\ \hline 18 \end{array}$$
  

$$\begin{array}{r} 17 \\ - 9 \\ \hline -20 \end{array}$$

– Em geral, o **overflow** precisa ser detectado e/ou tratado

49

© F. Batista – Adapted by D. Lettim

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- **Overflow** só ocorre se os dois números de uma soma são ambos positivos ou ambos negativos
  - Exemplos (5 bits):

$\begin{array}{r} 10 \\ + 8 \\ \hline 18 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ + 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \hline 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{array}$ <span style="color: red;">Errado!</span>
$\begin{array}{r} -9 \\ + -11 \\ \hline -20 \end{array}$	$\begin{array}{r} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ + 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ \hline 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{array}$ <span style="color: red;">Errado!</span>

50

© F. Batista – Adapted by D. Lettim

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- **Overflow** só ocorre se os dois números de uma soma são ambos positivos ou ambos negativos
  - Exemplos (5 bits):

$\begin{array}{r} 10 \\ + 8 \\ \hline 18 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ + 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{array}$ <span style="color: red;">Errado!</span>
$\begin{array}{r} -9 \\ + -11 \\ \hline -20 \end{array}$	$\begin{array}{r} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ + 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ \hline 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{array}$ <span style="color: red;">Errado!</span>

51

© F. Batista – Adapted by D. Lettim

**Circuitos Aritméticos**  
Overflow

Universidade Federal de Santa Catarina

- Somador/Subtrator Paralelo com Detecção de Overflow:

$c_{n+1}$        $x_n$        $y_n$   
 $c$        $x_3$        $y_3$   
 $x_2$        $y_2$   
 $x_1$        $y_1$   
 $x_0$        $y_0$

Se  $C = 0 \rightarrow R = X + Y$   
 Se  $C = 1 \rightarrow R = X - Y$   
 Se  $O = 1 \rightarrow$  ocorreu overflow

52

© F. Batista – Adapted by D. Lettim

**Plano de Aula**

Universidade Federal de Santa Catarina

- Circuitos Aritméticos
  - Somadores
  - Subtratores
- **Somadores paralelos**
  - *Carry ripple, Carry look-ahead, Carry select*

53

© F. Batista – Adapted by D. Lettim

**Revisão: somadores**

Universidade Federal de Santa Catarina

- O somador é um circuito fundamental em sistemas digitais
- Problema central: propagação do *carry*
- Multiplicadores também são limitados pela propagação do *carry*
- Os conceitos utilizados nos projetos de somadores e multiplicadores são utilizados também na solução de outros problemas em sistemas digitais

54

© E. Bezerra – Adapted by D. Lettim

**Full-Adder**

Universidade Federal de Santa Catarina

- Full-adder:** Adds 3 bits, generates sum and carry
- Design using combinational design process

Step 1: Capture the function

Inputs	Outputs
a b ci 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	co s 0 0 0 1 0 1 1 0 0 1 1 0 1 0 1 1

Step 2: Convert to equations

$$\begin{aligned} co &= a'b'c + ab'c + abc' + abc \\ co &= a'b'c + abc' + abc + abc' + abc \\ co &= (a'+a)bc + (b'+b)ac + (c'+c)ab \\ co &= bc + ac + ab \end{aligned}$$

$$\begin{aligned} s &= a'b'c + a'bc' + ab'c + abc \\ s &= a'(b'c + bc') + a(b'c' + bc) \\ s &= a'(b \oplus c) + a(b \oplus c) \\ s &= a \oplus b \oplus c \end{aligned}$$

Step 3: Create the circuit

5

**Carry-Ripple Adder**

Universidade Federal de Santa Catarina

- Using half-adder and full-adders, we can build adder that adds like we would by hand
- Called a **carry-ripple adder**
  - 4-bit adder shown: Adds two 4-bit numbers, generates 5-bit output
    - 5-bit output can be considered 4-bit "sum" plus 1-bit "carry out"
  - Can easily build any size adder

(a)

(b)

5

**Carry-Ripple Adder**

Universidade Federal de Santa Catarina

- Using full-adder instead of half-adder for first bit, we can include a "carry in" bit in the addition
  - Will be useful later when we connect smaller adders to form bigger adders

(a)

(b)

5

**Carry-Ripple Adder's Behavior**

Universidade Federal de Santa Catarina

Assume all inputs initially 0

0111+0001 (answer should be 01000)

Output after 2 ns (1FA delay)

Wrong answer -- something wrong? No -- just need more time for carry to ripple through the chain of full adders.

5

**Carry-Ripple Adder's Behavior**

Universidade Federal de Santa Catarina

0111+0001 (answer should be 01000)

Outputs after 4ns (2 FA delays)

Outputs after 6ns (3 FA delays)

Output after 8ns (4 FA delays)

Correct answer appears after 4 FA delays

5

**Cascading Adders**

Universidade Federal de Santa Catarina

a7a6a5a4 b7b6b5b4 a3a2a1a0 b3b2b1b0  
4-bit adder ci  
a3a2a1a0 b3b2b1b0  
4-bit adder ci  
a7.. a0 b7.. b0  
8-bit adder ci

(a)

(b)

6

**Faster Adder**

Universidade Federal de Santa Catarina

- Built carry-ripple adder
  - Similar to adding by hand, column by column
  - Con: Slow
    - Output is not correct until the carries have rippled to the left
    - 4-bit carry-ripple adder has  $4 \cdot 2 = 8$  gate delays
  - Pro: Small
    - 4-bit carry-ripple adder has just  $4 \cdot 5 = 20$  gates

© Vahid – Adapted by D. Lettmann

**Faster Adder**

Universidade Federal de Santa Catarina

- Faster adder – Use two-level combinational logic design process
  - Recall that 4-bit two-level adder was big
  - Pro: Fast
    - 2 gate delays
  - Con: Large
    - Truth table would have  $2^{(4+4)} = 256$  rows
    - Plot shows 4-bit adder would use about 500 transistors
- Is there a compromise design?
  - Between 2 and 8 gate delays
  - Between 20 and 500 gates

© Vahid – Adapted by D. Lettmann

**Faster Adder – (Bad) Attempt at “Lookahead”**

Universidade Federal de Santa Catarina

- Idea
  - Modify carry-ripple adder – For a stage's carry-in, don't wait for carry to ripple, but rather *directly compute* from inputs of earlier stages
    - Called "lookahead" because current stage "looks ahead" at previous stages rather than waiting for carry to ripple to current stage

© Vahid – Adapted by D. Lettmann

**Faster Adder – (Bad) Attempt at “Lookahead”**

Universidade Federal de Santa Catarina

- Want each stage's **carry-in** bit to be function of **external inputs** only ( $a$ 's,  $b$ 's, or  $c_0$ )
- Recall full-adder equations:
  - $s = a \oplus b$
  - $c = bc + ac + ab$

Stage 0: Carry-in is already an external input:  $c_0$

Stage 1:  $c_1 = c_0 \oplus a_1 b_1$

Stage 2:  $c_2 = c_1 \oplus a_2 b_2 + a_1 c_1$

Stage 3:  $c_3 = c_2 \oplus a_3 b_3 + a_2 c_2 + a_1 b_1$

© Vahid – Adapted by D. Lettmann Continue for c3

**Faster Adder – (Bad) Attempt at “Lookahead”**

Universidade Federal de Santa Catarina

- Carry lookahead logic function of **external inputs**
  - No waiting for ripple
- Problem
  - Equations get too big
  - Not efficient
  - Need a better form of lookahead

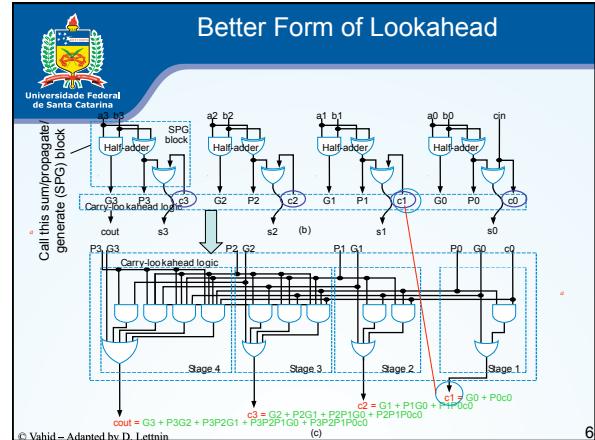
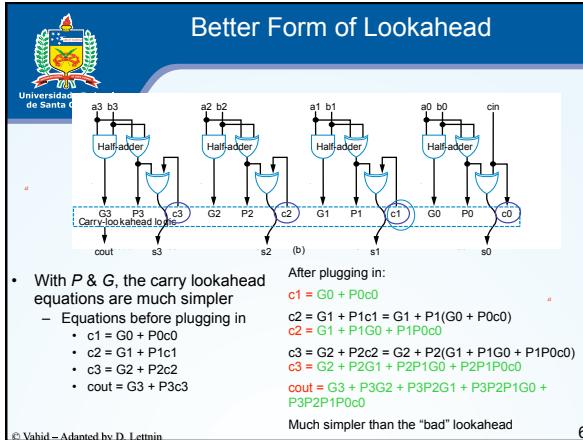
© Vahid – Adapted by D. Lettmann

**Better Form of Lookahead**

Universidade Federal de Santa Catarina

- Have each stage compute two terms
  - Propagate:**  $P = a \oplus b$
  - Generate:**  $G = ab$
- Compute lookahead from **P** and **G** terms, *not from external inputs*
  - Why P & G? Because the logic comes out much simpler
    - Very clever finding; not particularly obvious though
    - Why those names?
  - G: If a and b are 1, carry-out will be 1 – "generate" a carry-out of 1 in this case
  - P: If only one of a or b is 1, then carry-out will equal the carry-in – propagate the carry-in to the carry-out in this case

© Vahid – Adapted by D. Lettmann



**EEL7020 – Sistemas Digitais**  
**Aula 5: Circuitos combinacionais -**  
**Circuitos Aritméticos**

Prof. Djones Vinicius Lettnin  
 lettnin@eel.ufsc.br  
<http://lettnin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin  
 from the original slides made available by the authors  
 E. Batista, E. Bezerra e J. Güntzel.