SCE 0117 - Introdução à Lógica Digital

Representação Numérica e Circuitos Aritméticos

Prof. Dr. Vanderlei Bonato

Inteiros sem sinal

- $D = d_{n-1}, d_{n-2}, d_1 d_0$
- Decimal

  $V(D) = d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \ldots + d_0 \times 10^0$

  $(8547)_{10} = 8 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$

- Binário

  $V(B) = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \ldots + b_0 \times 2^0$

  $(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Definições

- Bit
- Nibble
- Byte
- LSB (Least-Significant Bit)
- MSB (Most-Significant Bit)

Conversão de decimal para binário

Convert $(857)_{10}$

					Remainder	
857	:	2	=	428	1	LSB
428	:	2	=	214	0	
214	:	2	=	107	0	
107	:	2	=	53	1	
53	:	2	=	26	1	
26	:	2	=	13	0	
13	:	2	=	6	1	
6	:	2	=	3	0	
3	:	2	=	1	1	
1	:	2	=	0	1	MSB

Result is $(1101011001)_2$

Figure 5.1. Conversion from decimal to binary.

Representação de números em qualquer base

- $K = k_{n-1}\ k_{n-2}\ \ldots\ k_1\ k_0$
- $V(k) = \sum_{i=0}^{n-1} k_i \times r^i$

Decimal	Binary	Octal	Hexadecimal
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10
17	10001	21	11
18	10010	22	12

Table 5.1. Numbers in different systems.

Sistemas de numeração

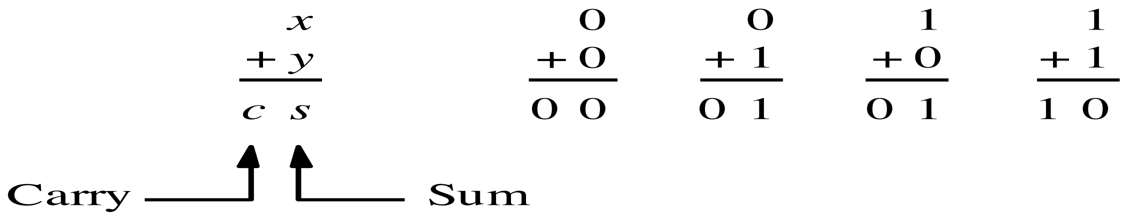
- Em computadores o sistema de números dominante é o binário
- A razão para o uso do sistema octal e hexadecimal (hex) é que eles servem como uma notação simplificada para números binários
  - Em computadores é comum o uso de 32 ou 64 bits

Conversões (continuação)

- Binário para Octal
  - Formar grupo de 3 bits
- Binário para Hexadecimal
  - Formar grupo de 4 bits

- Octal para Binário
  - Cada dígito corresponde a 3 bits
- Hexadecimal para Binário
  - Cada dígito corresponde a 4 bits

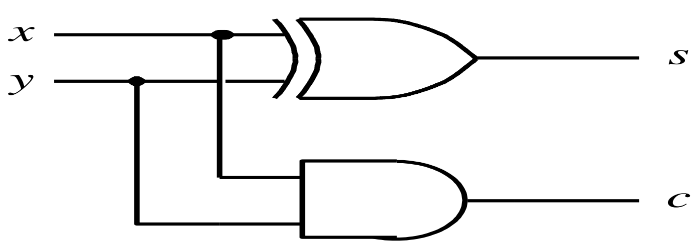
Adição de números binários



(a) The four possible cases

x	y	Carry c	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(b) Truth table



(c) Circuit

(d) Graphical symbol

Figure 5.2. Half-adder.

$X = x_4x_3x_2x_1x_0$	0 1 1 1 1	$(15)_{10}$
$+\,Y = y_4y_3y_2y_1y_0$	0 1 0 1 0	$(10)_{10}$
	1 1 1 0 ← Generated carries	
$S = s_4s_3s_2s_1s_0$	1 1 0 0 1	$(25)_{10}$

Figure 5.3. An example of addition.

c_i	x_i	y_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(a) Truth table

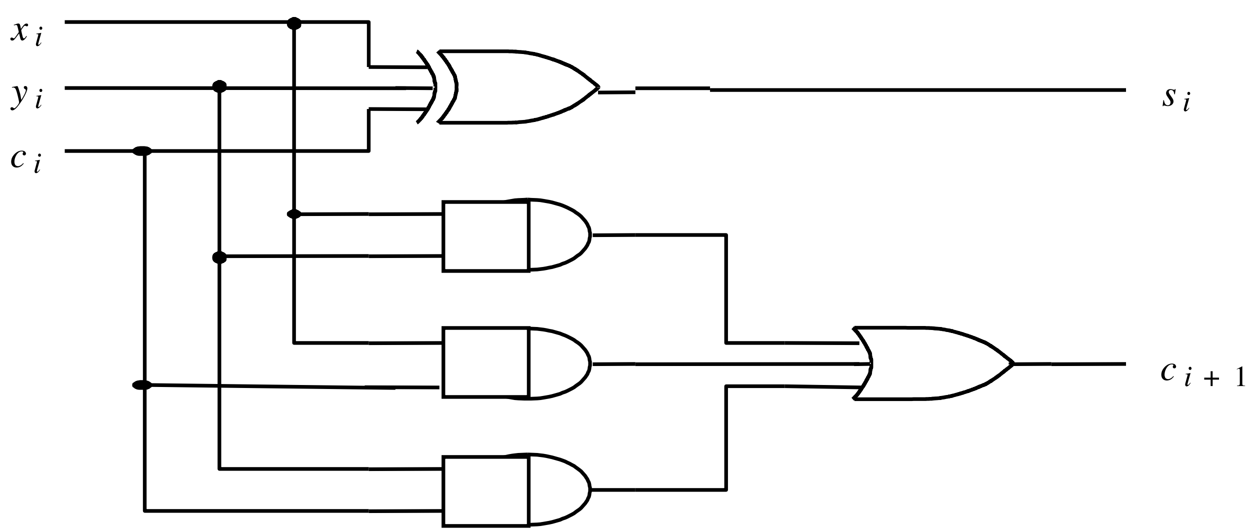
c_i \ $x_i y_i$	00	01	11	10
0		1		1
1	1		1	

$$s_i = x_i \oplus y_i \oplus c_i$$

c_i \ $x_i y_i$	00	01	11	10
0			1	
1		1	1	1

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

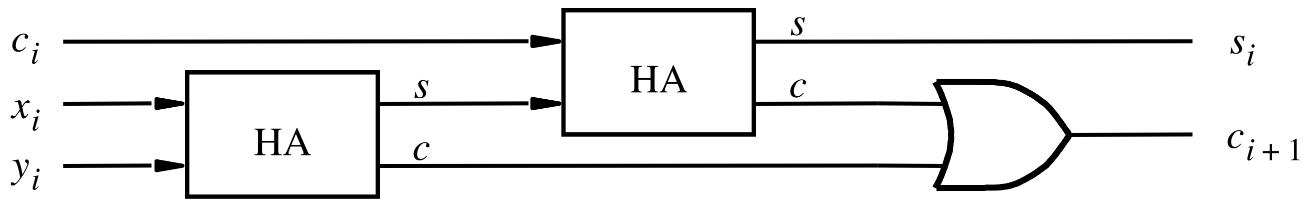
(b) Karnaugh maps



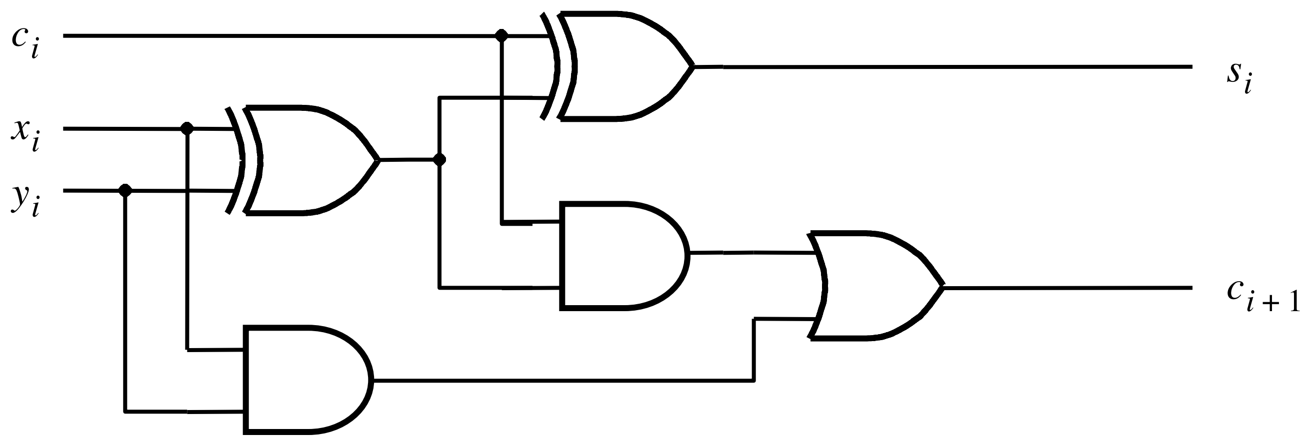
(c) Circuit

Figure 5.4. Full-adder.

Verifique se o comportamento está correto



(a) Block diagram



(b) Detailed diagram

Figure 5.5. A decomposed implementation of the full-adder circuit.

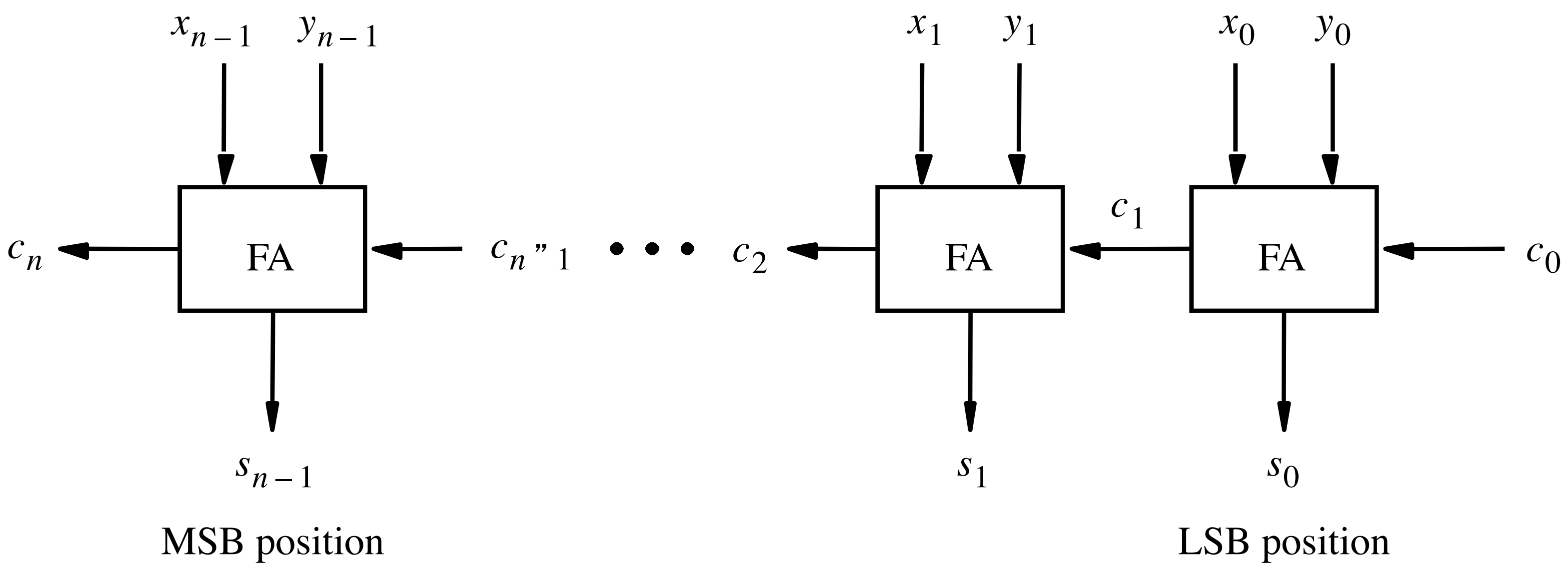


Figure 5.6. An *n*-bit ripple-carry adder.

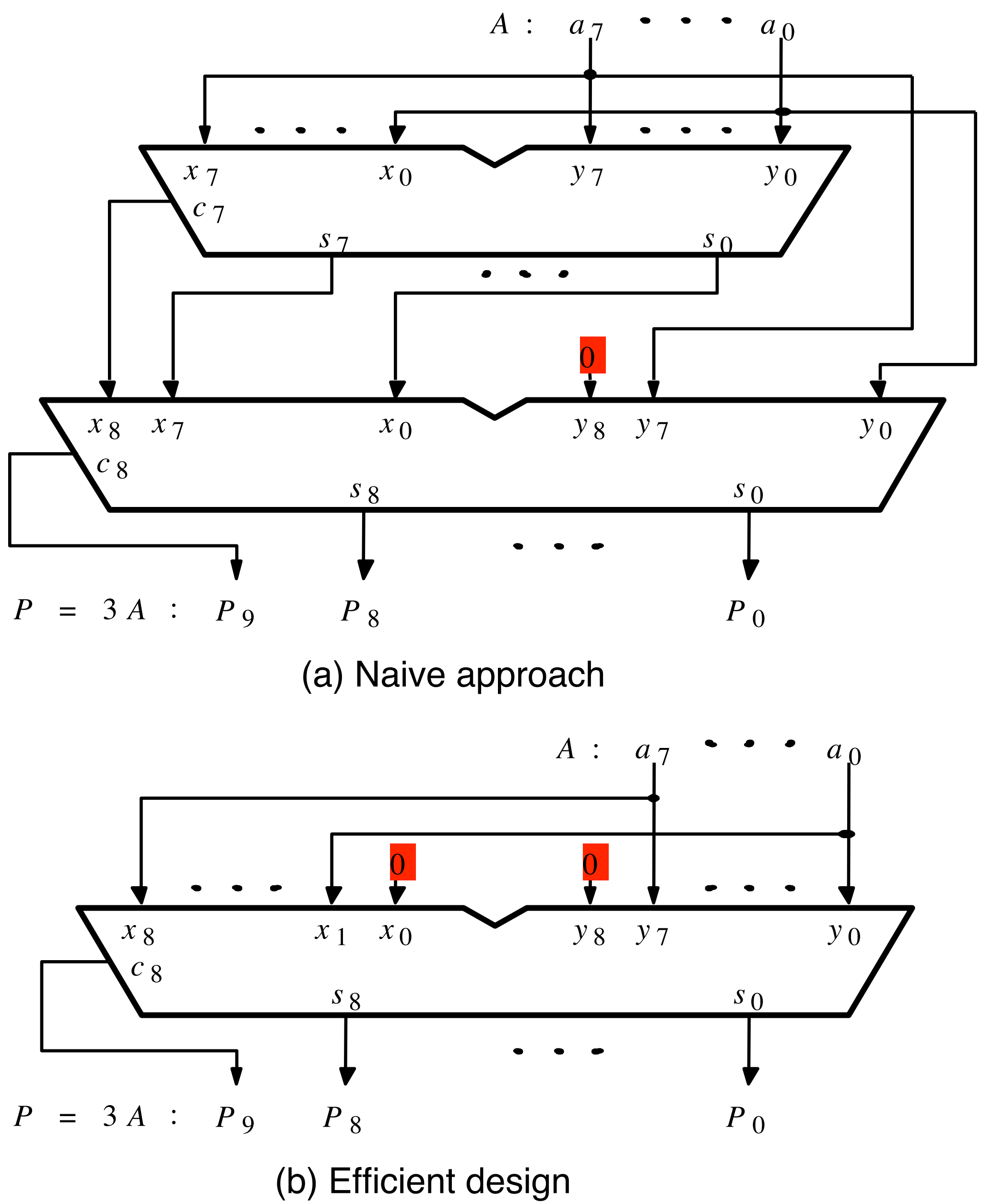


Figure 5.7. Circuit that multiplies an eight-bit unsigned number by 3.

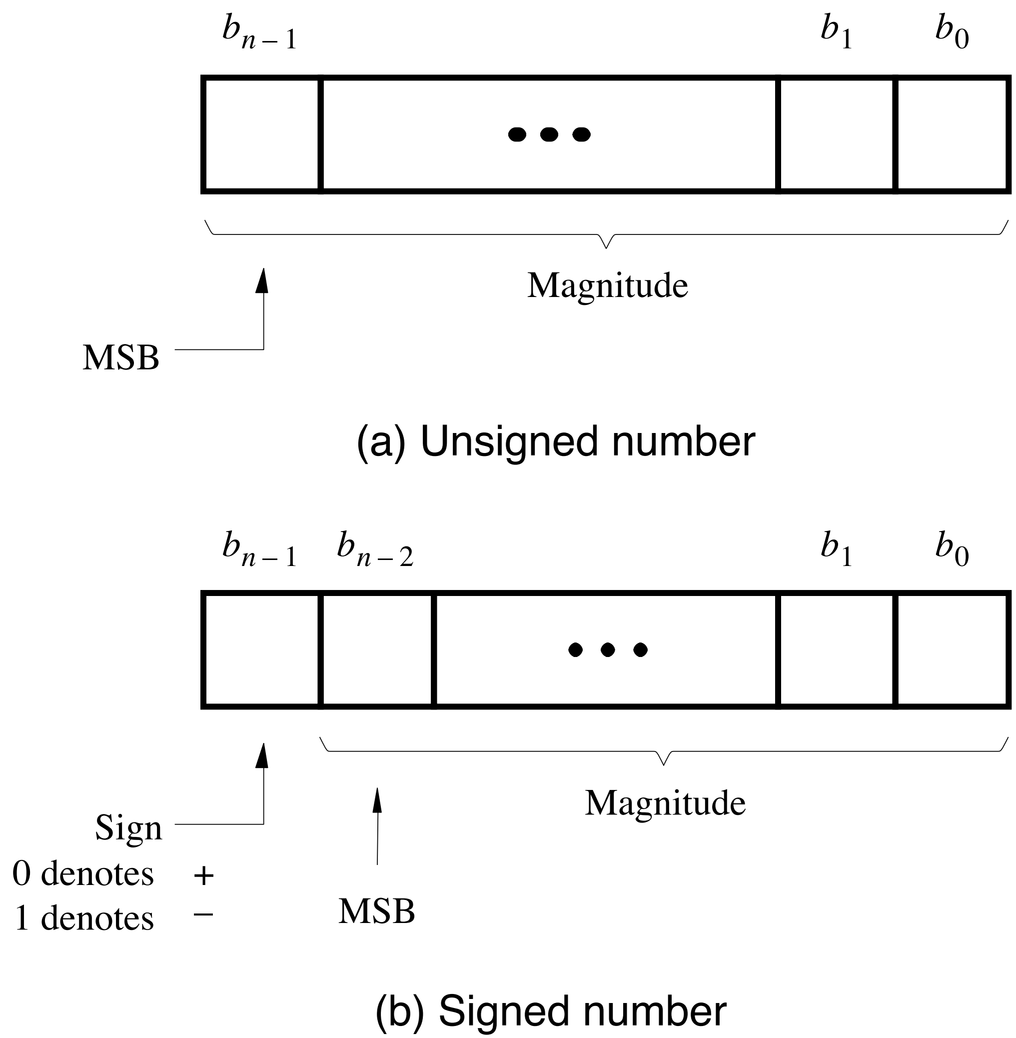


Figure 5.8. Formats for representation of integers.

$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0111	+7	+7	+7
0110	+6	+6	+6
0101	+5	+5	+5
0100	+4	+4	+4
0011	+3	+3	+3
0010	+2	+2	+2
0001	+1	+1	+1
0000	+0	+0	+0
1000	−0	−7	−8
1001	−1	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	−1	−2
1111	−7	−0	−1

Table 5.2. Interpretation of four-bit signed integers.

(+5)	0 1 0 1	(−5)	1 0 1 0
+ (+2)	+ 0 0 1 0	+ (+2)	+ 0 0 1 0
(+7)	0 1 1 1	(-3)	1 1 0 0

(+5)	0 1 0 1	(−5)	1 0 1 0
+ (−2)	+ 1 1 0 1	+ (−2)	+ 1 1 0 1
(+3)	1 0 0 1 0	(−7)	1 0 1 1 1
	→ 1		→ 1
	0 0 1 1		1 0 0 0

Figure 5.9. Examples of 1's complement addition.

Decimal	Binary
(+5)	0101
+ (+2)	+ 0010
(+7)	0111

Decimal	Binary
(–5)	1011
+ (+2)	+ 0010
(–3)	1101

Decimal	Binary
(+5)	0101
+ (–2)	+ 1110
(+3)	1 0011 (leading 1: ignore)

Decimal	Binary
(–5)	1011
+ (–2)	+ 1110
(–7)	1 1001 (leading 1: ignore)

Figure 5.10. Examples of 2's complement addition.

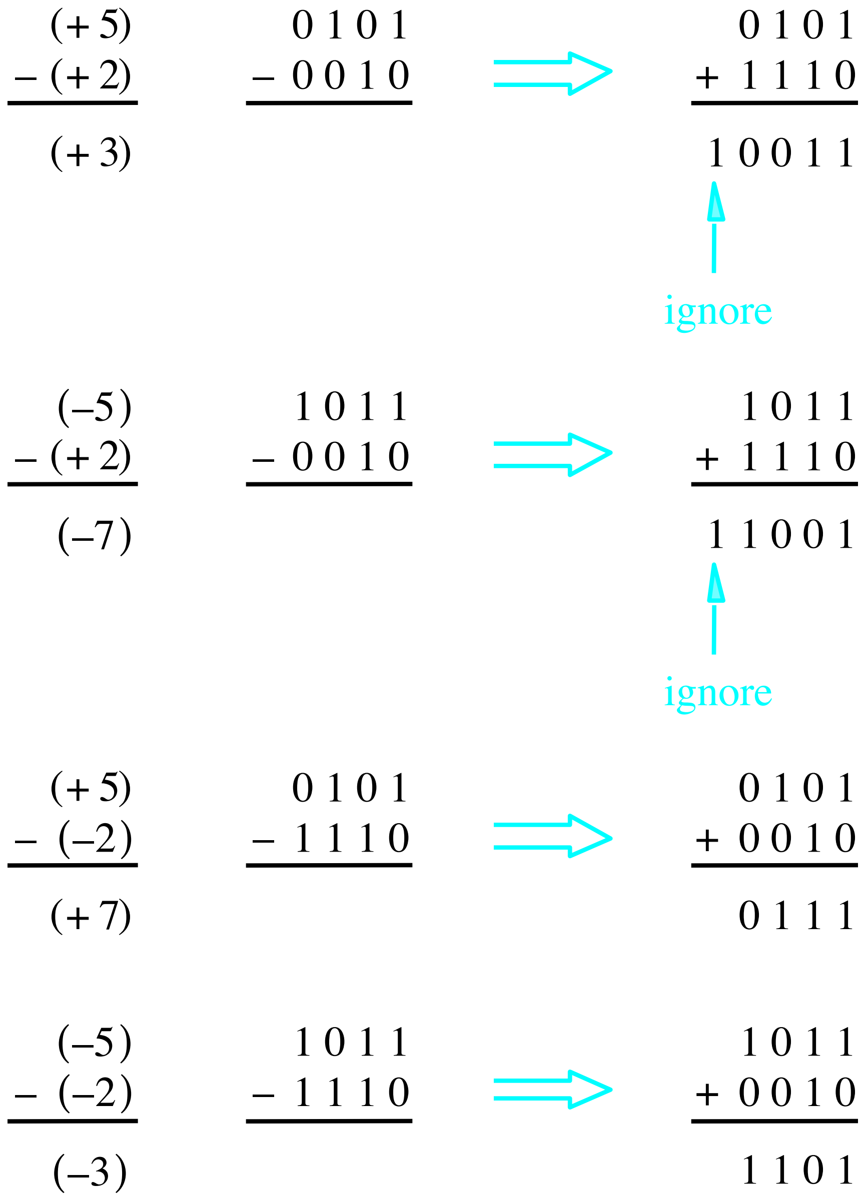


Figure 5.11. Examples of 2's complement subtraction.

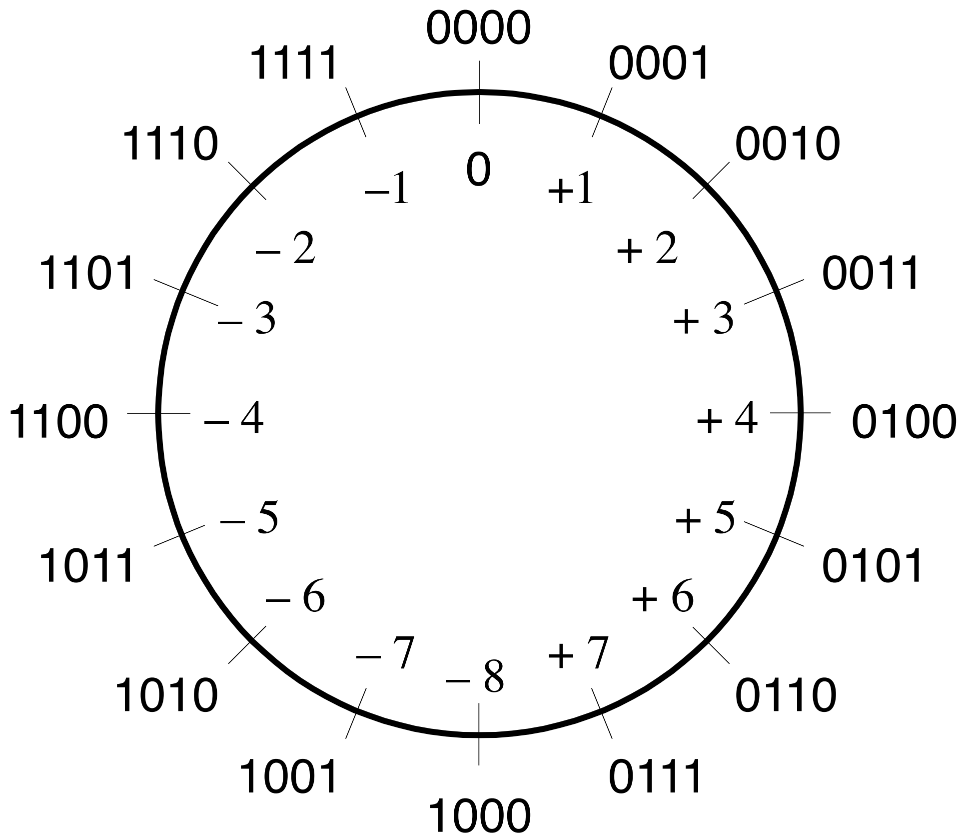


Figure 5.12. Graphical interpretation of four-bit 2's complement numbers.

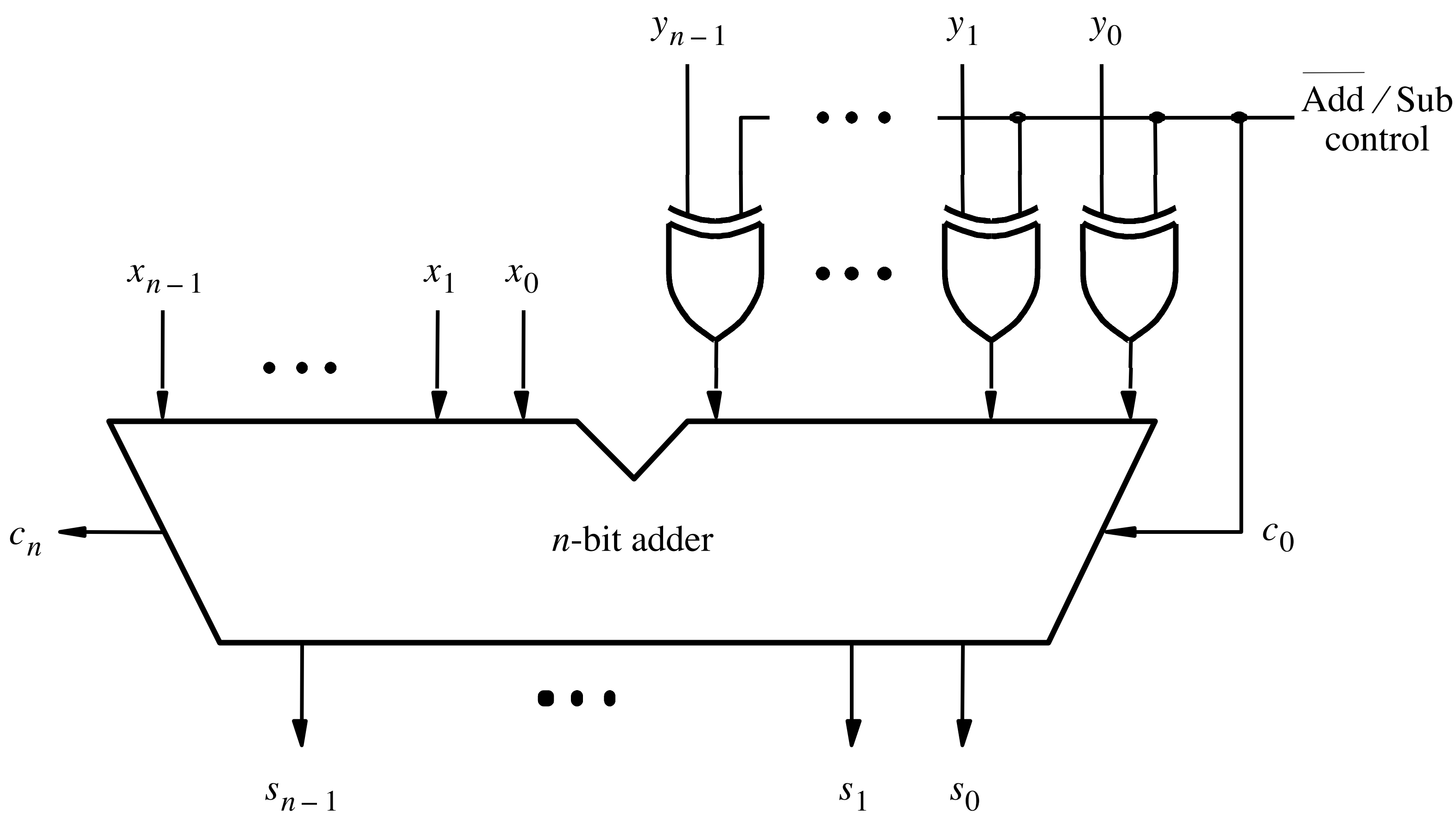


Figure 5.13. Adder/subtractor unit.

$$
\begin{array}{rr}
 & (+7) \\
+ & (+2) \\
\hline
 & (+9)
\end{array}
\qquad
\begin{array}{rr}
 & 0\,1\,1\,1 \\
+ & 0\,0\,1\,0 \\
\hline
 & 1\,0\,0\,1
\end{array}
\qquad
\begin{array}{l}
c_4 = 0 \\
c_3 = 1
\end{array}
$$

$$
\begin{array}{rr}
 & (-7) \\
+ & (+2) \\
\hline
 & (-5)
\end{array}
\qquad
\begin{array}{rr}
 & 1\,0\,0\,1 \\
+ & 0\,0\,1\,0 \\
\hline
 & 1\,0\,1\,1
\end{array}
\qquad
\begin{array}{l}
c_4 = 0 \\
c_3 = 0
\end{array}
$$

$$
\begin{array}{rr}
 & (+7) \\
+ & (-2) \\
\hline
 & (+5)
\end{array}
\qquad
\begin{array}{rr}
 & 0\,1\,1\,1 \\
+ & 1\,1\,1\,0 \\
\hline
 & 1\,0\,1\,0\,1
\end{array}
\qquad
\begin{array}{l}
c_4 = 1 \\
c_3 = 1
\end{array}
$$

$$
\begin{array}{rr}
 & (-7) \\
+ & (-2) \\
\hline
 & (-9)
\end{array}
\qquad
\begin{array}{rr}
 & 1\,0\,0\,1 \\
+ & 1\,1\,1\,0 \\
\hline
 & 1\,0\,1\,1\,1
\end{array}
\qquad
\begin{array}{l}
c_4 = 1 \\
c_3 = 0
\end{array}
$$

Figure 5.14. Examples of determination of overflow.

Exercícios

- Conversões de números
  - (Brown, 2005)
    - Pgs 310-312

FIM