CHAPTER 4

Applications of Boolean Algebra/
Minterm and Maxterm Expansions

Contents

Objective

- Conversion of English Sentences to Boolean Equations
- Combinational Logic Design Using a Truth Table
- Minterm and Maxterm Expansions
- General Minterm and Maxterm Expansions
- Incompletely Specified Functions (Don’t care term)
- Examples of Truth Table Construction
- Design of Binary Adders(Full adder) and Subtracters

4.1 Conversion of English Sentences to Boolean Equations

- Steps in designing a single-output combinational switching circuit

1. Find switching function which specifies the desired behavior of the circuit
2. Find a simplified algebraic expression for the function
3. Realize the simplified function using available logic elements

1. F is 'true' if A and B are both 'true' → F=AB

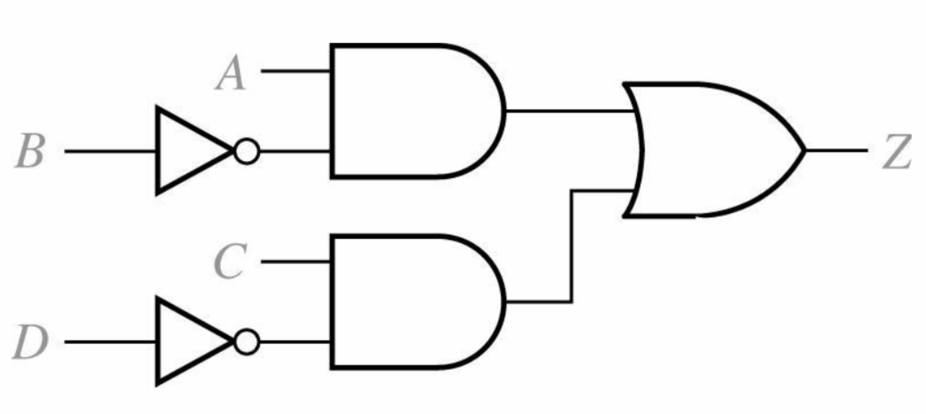
4.1 Conversion of English Sentences to Boolean Equations

1. The alarm will ring(Z) iff the alarm switch is turned on(A) *and* the door is not closed(B'), *or* it is after 6PM(C) and window is not closed(D')

2. Boolean Equation

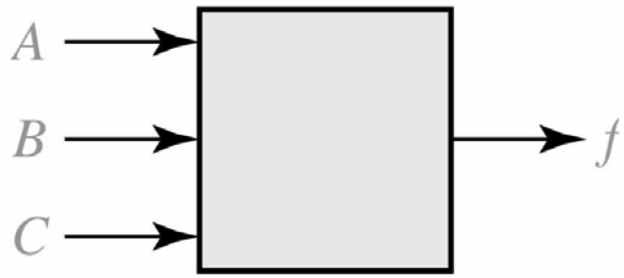
$$Z = AB' + CD'$$

3. Circuit realization



4.2 Combinational Logic Design Using a Truth Table

- Combinational Circuit with Truth Table



(a)

A	B	C	f	f'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

(b)

When expression for $f=1$ →

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

4.2 Combinational Logic Design Using a Truth Table

Original equation →

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

Simplified equation →

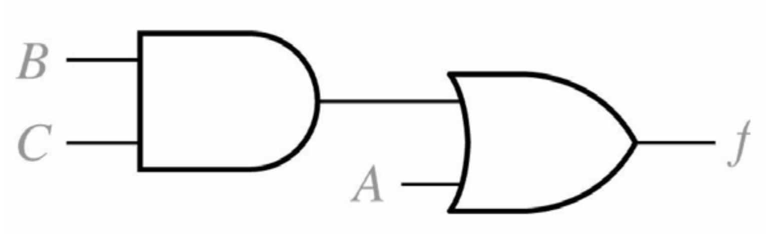
$$f = A'BC + AB' + AB = A'BC + A = A + BC$$

Proof: F = A’BC + AB’(C’ + C) + AB(C’ + C) (eq. 2-8, p.38)

= A’BC + AB’ + AB = A’BC + A(B’ + B) (eq. 2-8, p.38)

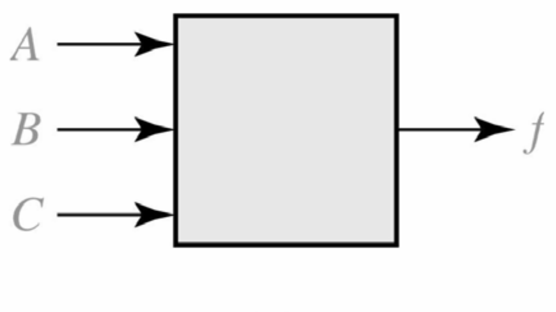
= A’BC + A = A + BC (eq. 2-14D, p.41)

Circuit realization →



4.2 Combinational Logic Design Using a Truth Table

- Combinational Circuit with Truth Table



(a)

A	B	C	f	f'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

(b)

When expression for *f=0* →

$$f = (A+B+C)(A+B+C')(A+B'+C)$$

$$f = (A+B)(A+B'+C) = A + B(B'+C) = A + BC$$

When expression for *f '=1* →

and take the complement of *f '*

$$f' = A'B'C' + A'B'C + A'BC'$$

$$\longrightarrow \quad f = (A+B+C)(A+B+C')(A+B'+C)$$

4.3 Minterm and Maxterm Expansions

- *literal* is a variable or its complement (e.g. *A, A'*)

- **Minterm, Maxterm** for three variables

Row No.	A B C	Minterms	Maxterms
0	0 0 0	$A'B'C' = m_0$	$A+B+C = M_0$
1	0 0 1	$A'B'C = m_1$	$A+B+C' = M_1$
2	0 1 0	$A'BC' = m_2$	$A+B'+C = M_2$
3	0 1 1	$A'BC = m_3$	$A+B'+C' = M_3$
4	1 0 0	$AB'C' = m_4$	$A'+B+C = M_4$
5	1 0 1	$AB'C = m_5$	$A'+B+C' = M_5$
6	1 1 0	$ABC' = m_6$	$A'+B'+C = M_6$
7	1 1 1	$ABC = m_7$	$A'+B'+C' = M_7$

4.3 Minterm and Maxterm Expansions

- *Minterm* of *n* variables is a product of *n* literals in which each variable appears exactly once in either true (*A*) or complemented form(*A'*), but not both. (→ m_0)

-Minterm expansion,

-Standard Sum of Product →

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$

$$f(A,B,C) = m_3 + m_4 + m_5 + m_6 + m_7$$

$$f(A,B,C) = \sum m(3,4,5,6,7)$$

4.3 Minterm and Maxterm Expansions

- *Maxterm* of *n* variables is a sum of *n* literals in which each variable appears exactly once in either true (*A*) or complemented form(*A'*) , but not both.(→ M_0)

- Maxterm expansion,

- Standard Product of Sum →

$$f = (A+B+C)(A+B+C')(A+B'+C)$$

$$f(A,B,C) = M_0 M_1 M_2$$

$$f(A,B,C) = \prod M(0,1,2)$$

4.3 Minterm and Maxterm Expansions

$$f(A,B,C) = m_3 + m_4 + m_5 + m_6 + m_7$$

$$f' = m_0 + m_1 + m_2 = \sum m(0,1,2)$$

$$f(A,B,C) = M_0 M_1 M_2 \longrightarrow f' = \prod M(3,4,5,6,7) = M_3 M_4 M_5 M_6 M_7$$

- *Minterm* and *Maxterm* expansions are complement each other

$$f' = (m_3 + m_4 + m_5 + m_6 + m_7)' = m'_3\, m'_4\, m'_5\, m'_6\, m'_7 = M_3 M_4 M_5 M_6 M_7$$

$$f' = (M_0 M_1 M_2)' = M'_0 + M'_1 + M'_2 = m_0 + m_1 + m_2$$

By Using DeMorgan's Law (eq. 2-21, 2-22, p.45)

4.4 General Minterm and Maxterm Expansions

A B C	F
0 0 0	a_0
0 0 1	a_1
0 1 0	a_2
0 1 1	a_3
1 0 0	a_4
1 0 1	a_5
1 1 0	a_6
1 1 1	a_7

-General truth table

for 3 variables

- *ai* is either '0' or '1'

- Minterm expansion for general function

$$F = a_0 m_0 + a_1 m_1 + a_2 m_2 + \ldots + a_7 m_7 = \sum_{i=0}^{7} a_i m_i$$

a_i=1, minterm m_i is present

a_i=0, minterm m_i is not present

- Maxterm expansion for general function

$$F = (a_0 + M_0)(a_1 + M_1)(a_2 + M_2)\ldots(a_7 + M_7) = \prod_{i=0}^{7}(a_i + M_i)$$

a_i=1, $a_i + M_i$=1 , Maxterm M_i is not present

a_i=0, Maxterm is present

4.4 General Minterm and Maxterm Expansions

$$F' = [\prod_{i=0}^{7} (a_i + M_i)]' = \sum_{i=0}^{7} a'_i M'_i = \sum_{i=0}^{7} a'_i m_i$$

→ All minterm which are not present in *F* are present in *F* ‘

$$F' = [\sum_{i=0}^{7} a_i m_i]' = \prod_{i=0}^{7} (a'_i + m'_i) = \prod_{i=0}^{7} (a'_i + M_i)$$

→ All maxterm which are not present in *F* are present in *F* ‘

$$F = \sum_{i=0}^{2^n - 1} a_i m_i = \prod_{i=0}^{2^n - 1} (a_i + M_i)$$

$$F' = \sum_{i=0}^{2^n - 1} a'_i m_i = \prod_{i=0}^{2^n - 1} (a'_i + M_i)$$

4.4 General Minterm and Maxterm Expansions

If i and j are different, $m_i m_j = 0$

$$f_1 = \sum_{i=0}^{2^n-1} a_i m_i \qquad f_2 = \sum_{j=0}^{2^n-1} b_j m_j$$

$$f_1 f_2 = \left(\sum_{i=0}^{2^n-1} a_i m_i\right)\left(\sum_{j=0}^{2^n-1} b_j m_j\right) = \sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1} a_i b_j m_i m_j = \sum_{i=0}^{2^n-1} a_i b_i m_i$$

Example

$$f_1 = \sum m(0,2,3,5,9,11) \quad \text{and} \quad f_2 = \sum m(0,3,9,11,13,14)$$

$$f_1 f_2 = \sum m(0,3,9,11)$$

Conversion between minterm and maxterm expansions of *F* and *F'*

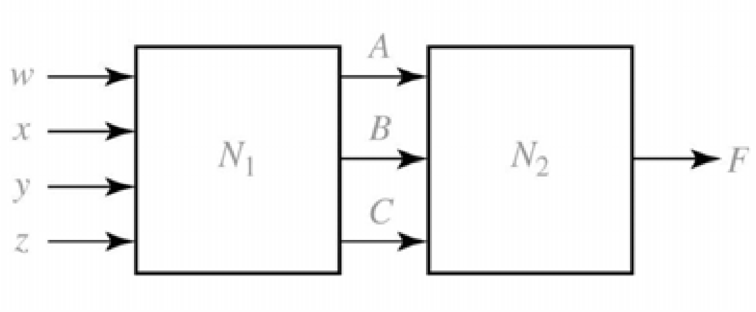
	Minterm Expansion of F	Maxterm Expansion of F	Minterm Expansion of F'	Maxterm Expansion of F'
Minterm Expansion of F		Maxterm nos. are those nos. not on the minterm list for F	List minterms Not present In F	Maxterm nos. Are the same As minterm nos. of F
Maxterm Expansion of F	Minterm nos. Are those nos. Not on the maxterm list forF		Minterm nos. Are the same as maxterm nos. of F	List maxterms not present inF

Table 4-3

Example

	Minterm Expansion of f	Maxterm Expansion of f	Minterm Expansion of f'	Maxterm Expansion of f'
f= $\sum m(3,4,5,6,7)$		$\prod M(0,1,2)$	$\sum m(0,1,2)$	$\prod M(3,4,5,6,7)$
f= $\prod M(0,1,2)$	$\sum m(3,4,5,6,7)$		$\sum m(0,1,2)$	$\prod M(3,4,5,6,7)$

4.5 Incompletely Specified Functions



If N_1 output does not generate all possible combination of A,B,C, the output of N2(F) has 'don't care' values.

Truth Table with Don't Cares

A B C	F
0 0 0	1
0 0 1	X
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	X
1 1 1	1

4.5 Incompletely Specified Functions

Finding
Function:

Case 1: assign ‘0’ on X’s

$$F = A'B'C' + A'BC + ABC = A'B'C' + BC$$

Case 2: assign ‘1’ to the first X and ‘0’ to the second ‘X’

$$F = A'B'C' + A'B'C + A'BC + ABC = A'B' + BC$$

Case 3: assign ‘1’ on X’s

$$F = A'B'C' + A'B'C + A'BC + ABC' + ABC = A'B' + BC + AB$$

→ The case 2 leads to the simplest function

4.5 Incompletely Specified Functions

- Minterm expansion for incompletely specified function

$$F = \sum m(0,3,7) + \sum d(1,6)$$

Don't Cares

- Maxterm expansion for incompletely specified function

$$F = \prod M(2,4,5) \prod D(1,6)$$

4.6 Examples of Truth Table Construction

Example 1 : Binary Adder

a	b	Sum	
0	0	0 0	0+0=0
0	1	0 1	0+1=1
1	0	0 1	1+0=1
1	1	1 0	1+1=2

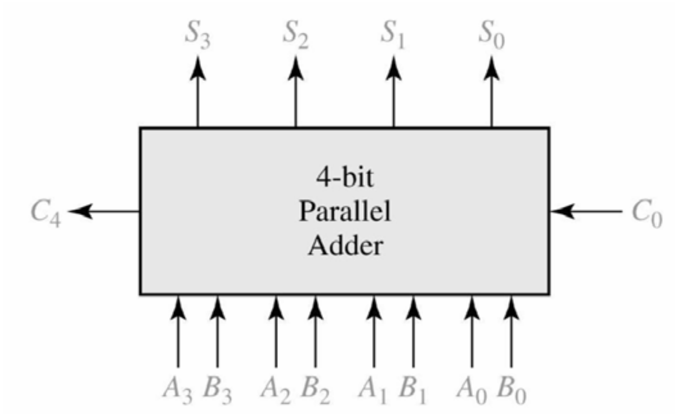
→

A	B	X	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$X = AB, Y = A'B + AB' = A \oplus B$$

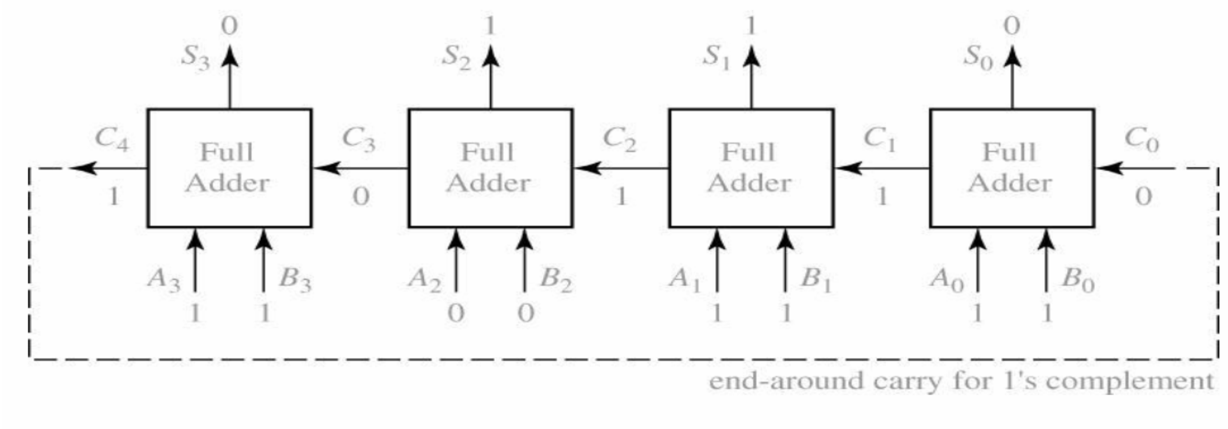
4.7 Design of Binary Adders and Subtracters

Parallel Adder for 4 bit Binary Numbers

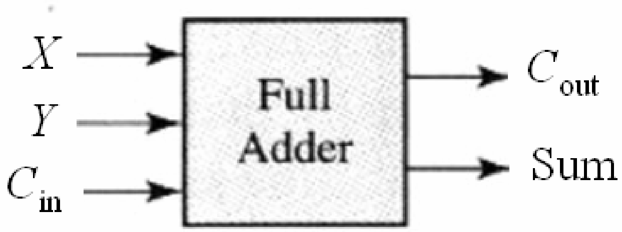


$$
\begin{array}{r}
10110 \text{ (carries)} \\
1011 \\
+1011 \\
\hline
10110
\end{array}
$$

Parallel adder composed of four full adders ← Carry Ripple Adder (slow!)



4.7 Design of Binary Adders and Subtracters



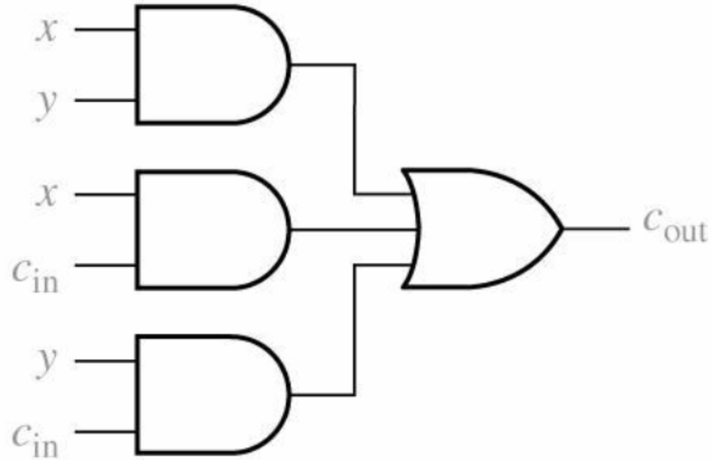
Truth Table for a Full Adder

X	Y	C_{in}	C_{out}	Sum
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

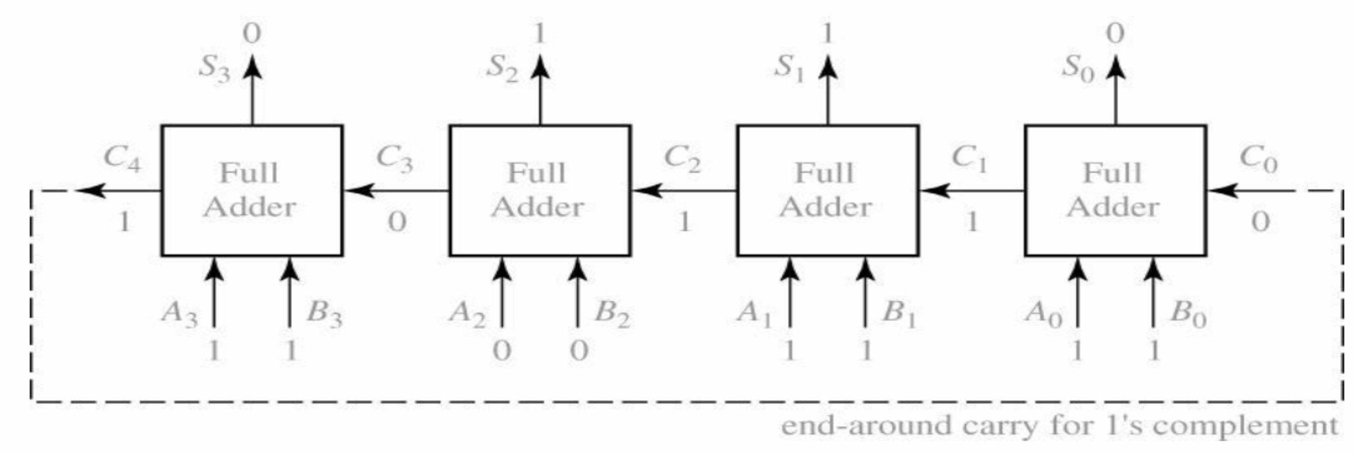
4.7 Design of Binary Adders and Subtracters

$$\begin{aligned}Sum &= X'Y'C_{in} + X'YC'_{in} + XY'C'_{in} + XYC_{in} \\ &= X'(Y'C_{in} + YC'_{in}) + X(Y'C'_{in} + YC_{in}) \\ &= X'(Y \oplus C_{in}) + X(Y \oplus C_{in})' = X \oplus Y \oplus C_{in}\end{aligned}$$

$$\begin{aligned}C_{out} &= X'YC_{in} + XY'C_{in} + XYC'_{in} + XYC_{in} \\ &= (X'YC_{in} + XYC_{in}) + (XY'C_{in} + XYC_{in}) + (XYC'_{in} + XYC_{in}) \\ &= YC_{in} + XC_{in} + XY\end{aligned}$$



When 1’s complement is used, the end-around carry is accomplished by connecting C4 to C0 input.



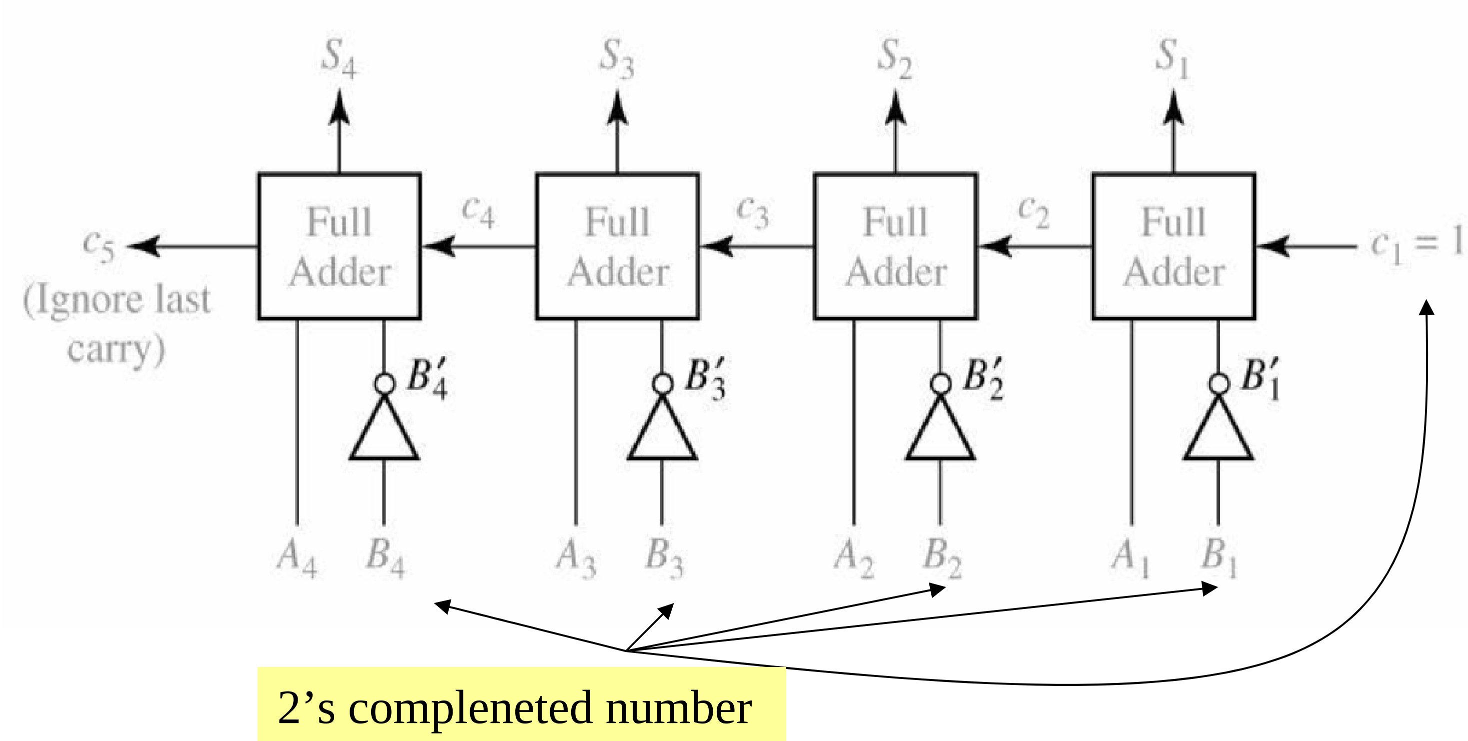
Overflow(V) when adding two signed binary number

$$V = A'_3 B'_3 S_3 + A_3 B_3 S'_3$$

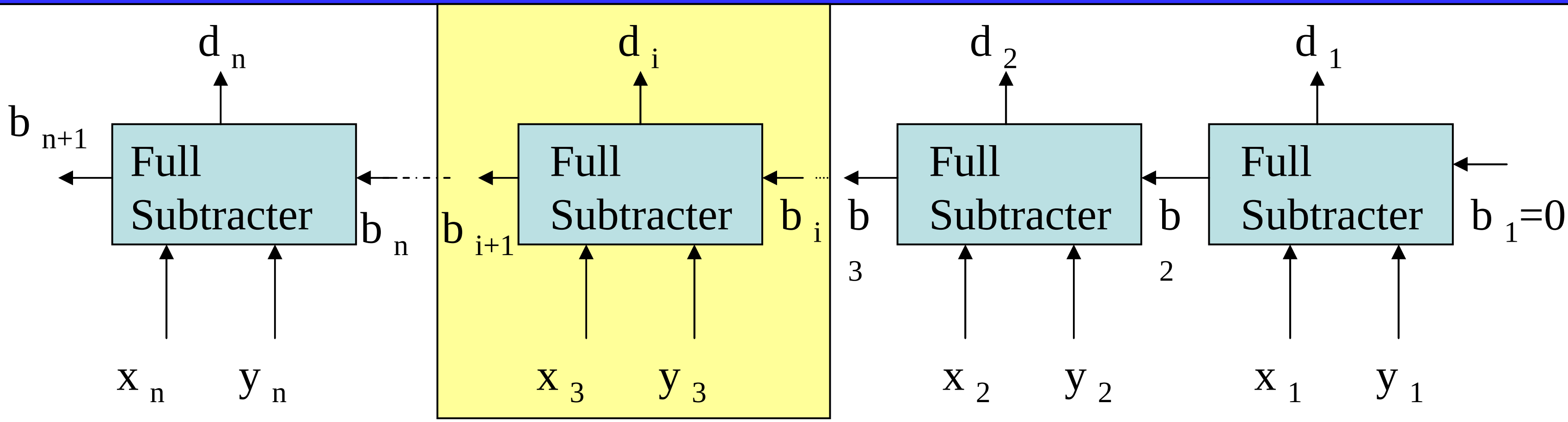
Subtracters

Binary Subtracter using full adder

- Subtraction is done by adding the 2's complemented number to be subtracted



Subtracters- using Full Subtracter



Truth Table for a Full Subtracter

xi	yi	bi	bi+1	di
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1