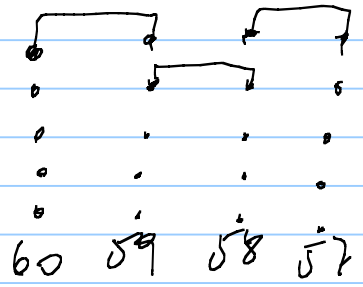
Note Title 2/19/2009

- Circuit 1 - continue
- Design of circuits from incompletely specified functions ("don't-cares") [4.5 text]
- A simple binary adder [4.6]
- Full adders and their use [4.7, first part]
- Karnaugh maps [5] : introduction

Daisy chain of 60 - 57:



Incompletely specified functions (Section 4.5)

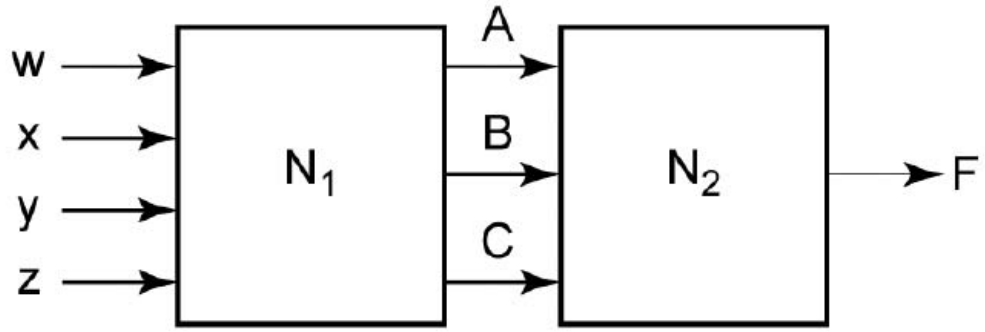


Table 4-5. Truth Table with Don't Cares

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

N_1 does not generate these configurations

(1)(2)(3)

0 1 1

0 0 1

Section 4.5, p. 93

(1) all X = 0: $F = A'B'C' + A'BC + ABC =$

$= A'B'C' + BC$

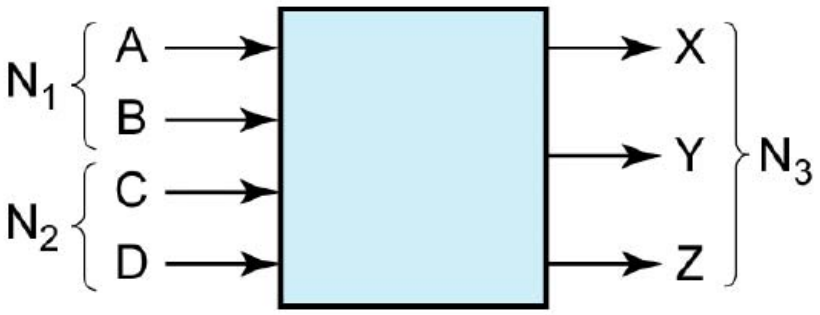
(2) $F = A'B'C' + A'B'C + A'BC + ABC = A'B' + BC$

(3) $F = A'B'C' + A'B'C + A'BC + ABC' + ABC =$

$= A'B' + A'BC + AB = A'[B' + BC] + AB =$

$= A'[B' + C] + AB = A'B' + A'C + AB$

Design of an adder (Example 2 section 4.6)



Section 4.6, p. 95

TRUTH TABLE:

N_1 (A B)	N_2 (C D)	N_3 (X Y Z)
0 0	0 0	0 0 0
0 0	0 1	0 0 1
0 0	1 0	0 1 0
0 0	1 1	0 1 1
0 1	0 0	0 0 1
0 1	0 1	0 1 0
0 1	1 0	0 1 1
0 1	1 1	1 0 0

TRUTH TABLE:

N_1 (A B)	N_2 (C D)	N_3 (X Y Z)
1 0	0 0	0 1 0
[redacted]	[redacted]	[redacted]
1 0	1 0	1 0 0
1 0	1 1	1 0 1
1 1	0 0	0 1 1
1 1	0 1	1 0 0
1 1	1 0	1 0 1
1 1	1 1	1 1 0

Section 4.6, p. 95

E.g., $N_1 =$ [redacted] (A B)

$+N_2 =$ [redacted] (C D)

$N_3 =$ [redacted] (X Y Z)

$X(A,B,C,D) = \sum m(7,10,11,13,14,15)$

$Y(A,B,C,D) = \sum m(2,3,5,6,8,9,12,15)$

$Z(A,B,C,D) = \sum m(1,3,4,6,9,11,12,14)$

Design of Binary Adders (4.7)

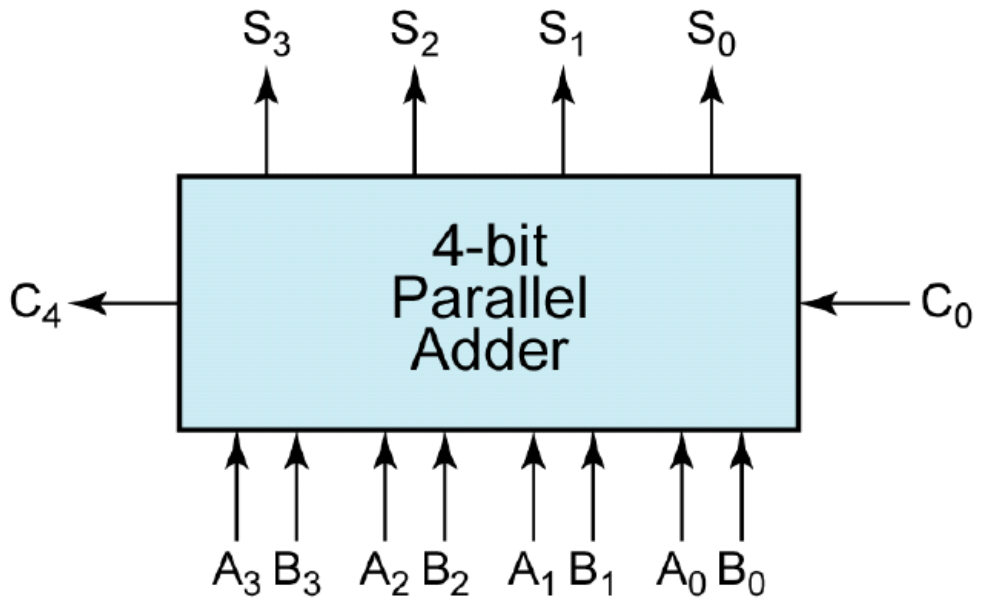


Figure 4-2: **Parallel Adder for 4-Bit Binary Numbers**

This circuit has nine inputs — too big!

Add two 4-bit unsigned numbers, giving a 4-bit number as result. Also, allow a 1-bit carry as input and a one-bit carry as output

$$\begin{array}{rcccccc} & & & & & & C_0 \\ + & & A_3 & A_2 & A_1 & A_0 \\ + & & B_3 & B_2 & B_1 & B_0 \\ \hline & C_4 & S_3 & S_2 & S_1 & S_0 \end{array}$$

Instead, connect four full adders.

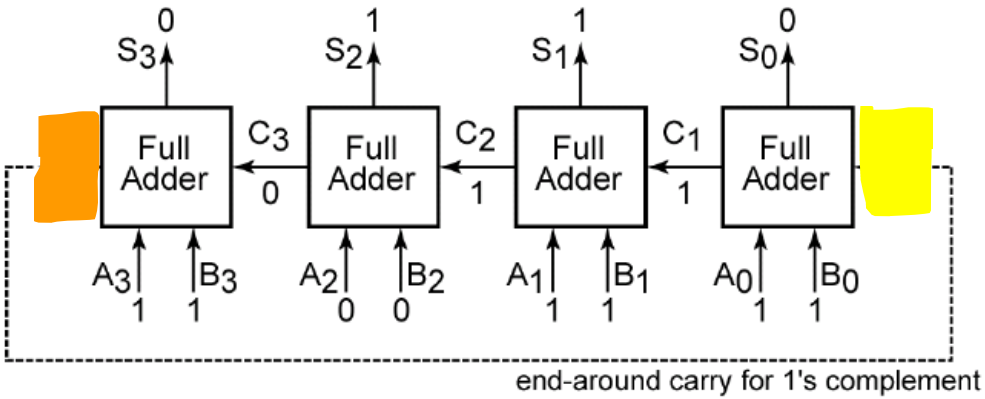
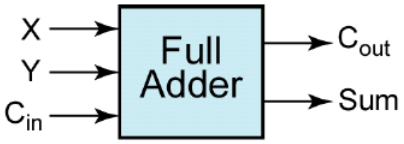


Figure 4-3: Parallel Adder Composed of Four Full Adders

Each full adder adds two one-bit numbers and a carry, and produces one one-bit number and a carry.

Ex.:

$$\begin{array}{rl} & 0 \\ 1011 & = A_3 A_2 A_1 A_0 \\ +\ 1011 & = B_3 B_2 B_1 B_0 \\ \hline 10110 & = S_3 S_2 S_1 S_0 \end{array}$$



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

Figure 4-4: **Truth Table for a Full Adder**

$$Sum = \sum m(1,2,4,8) =$$

$$= X'Y'C_{in} + X'YC_{in} + XY'C_{in}' + XYC_{in} =$$

$$= X'(Y'C_{in} + YC_{in}') + X(Y'C_{in}' + YC_{in}) =$$

$$= X'(Y \oplus C_{in}) + X(Y \equiv C_{in}) =$$

$$= X'(Y \oplus C_{in}) + X(Y \oplus C_{in})' =$$

$$= X \oplus (Y \oplus C_{in}) = ([3\text{-}13]: \text{associativity of exclusive-or}) = X \oplus Y \oplus C_{in}$$

$$C_{out} = \sum m(3,5,6,7) = X'YC_{in} + XY'C_{in} + XYC_{in}' + XYC_{in} =$$

$$= (X'YC_{in} + XYC_{in}) + (XY'C_{in} + XYC_{in}) + (XYC_{in}' + XYC_{in}) =$$

$$= YC_{in} + XC_{in} + XY$$

Here are the resulting circuits:

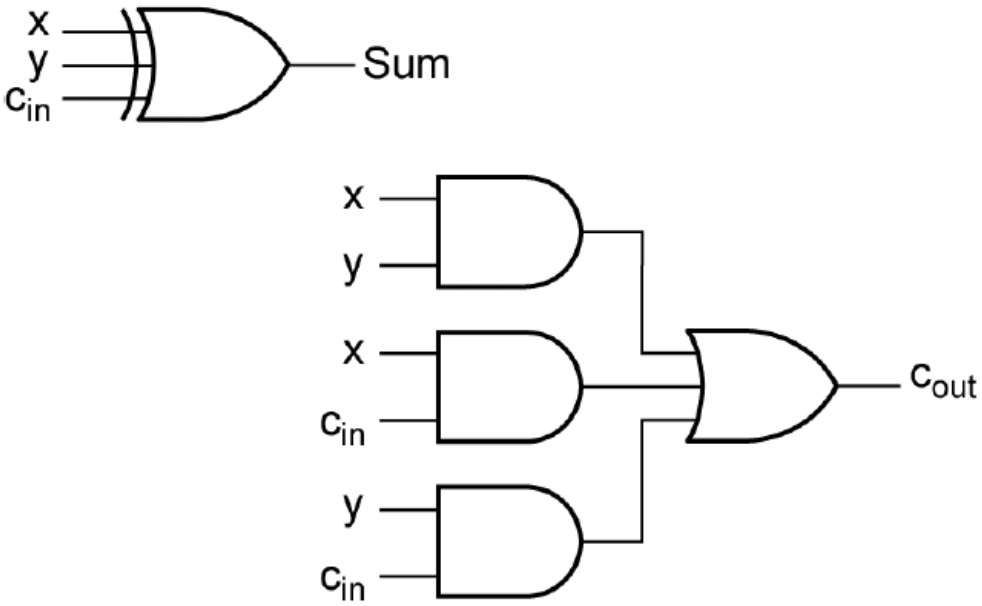
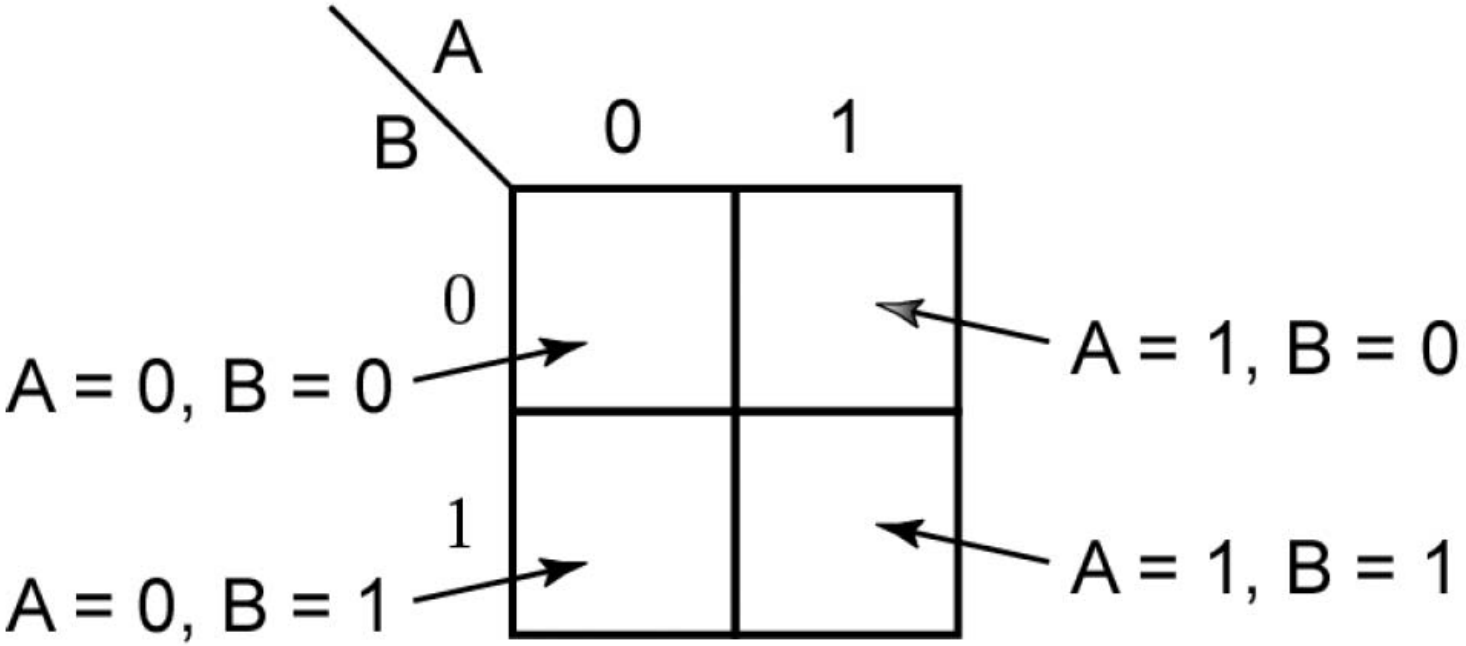


Figure 4-5: **Implementation of Full Adder**

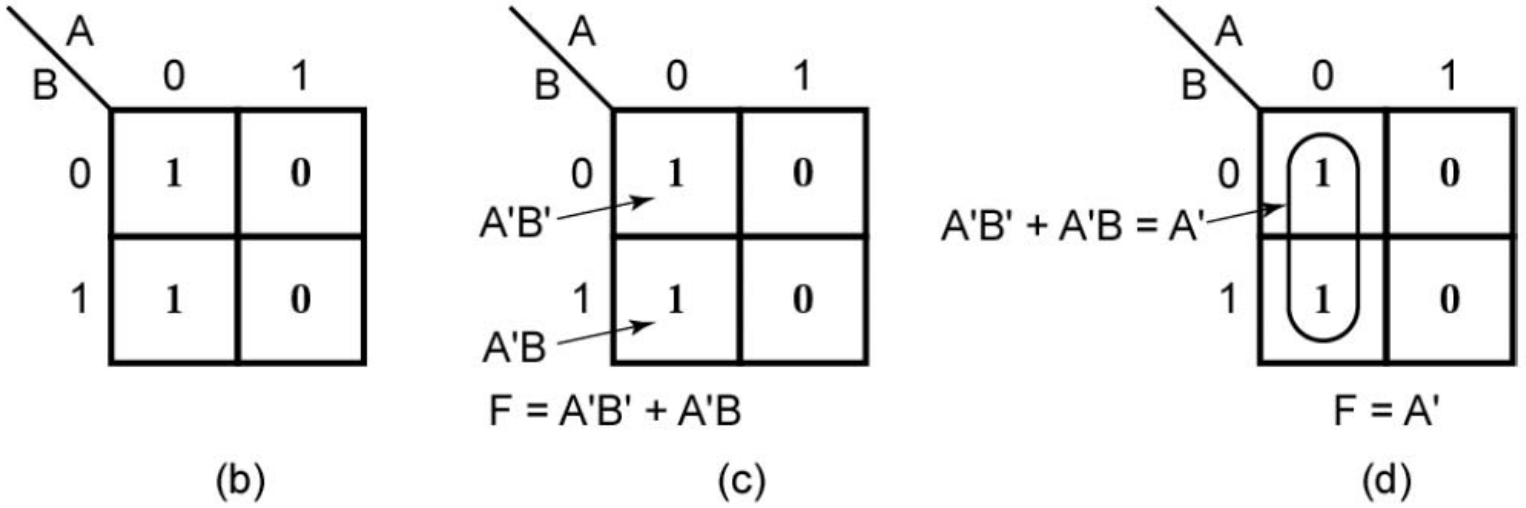
A Karnaugh map for two variables (A and B)



Section 5.2, p. 121

(a)

A	B	F
0	0	1
0	1	1
1	0	0
1	1	0



(b) (c) (d)

Figure 5-1a, b, c, and d