Note Title 2/19/2009

- HW 4 posted – tentative; due date to be determined

- Circuit kit will be handed out on Monday.
  We will start work on Circuit 1.

Ch. 4 (partly already covered)

Conversion of English sentences to Boolean equations (and circuits)

Example

"The alarm will ring if and only if (iff) the alarm switch is turned on and the door is not closed, or it is after 6pm and the door is not closed."

First step: match phrases with (propositional) variable, or literal

$\underbrace{\text{The alarm will ring}}_{Z}$ iff $\underbrace{\text{the alarm switch is on}}_{A}$ and $\underbrace{\text{the door is not closed}}_{B'}$ or $\underbrace{\text{it is after 6pm}}_{C}$ and $\underbrace{\text{the window is not closed}}_{D'}$.

Each phrase is true when the corresponding variable/literal is true; each phrase is false when the corresponding variable literal is false

Note that we represent "the door is closed" with B and therefore "the door is not closed" is B'. Similarly for D.

Second step: write the expression for the variable corresponding to the desired action (here: Z, "the alarm will ring") using the other variables

$$Z = AB' + CD'.$$

Third step: simplify according to [illegible] or [illegible]

Here, the expression $Z = AB' + CD'$ is already in sum-of-product form. It also has the minimum number of products and, among expressions with the minimum number of products, each product has the minimum size.

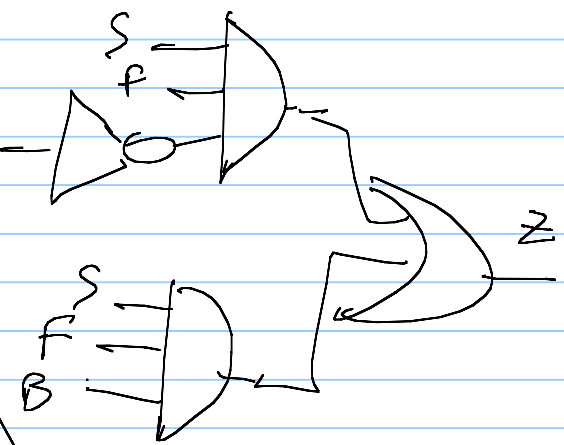
Q6 Z = Elevator door opens

S = Elevator is stopped

F = Elevator is level with Floor

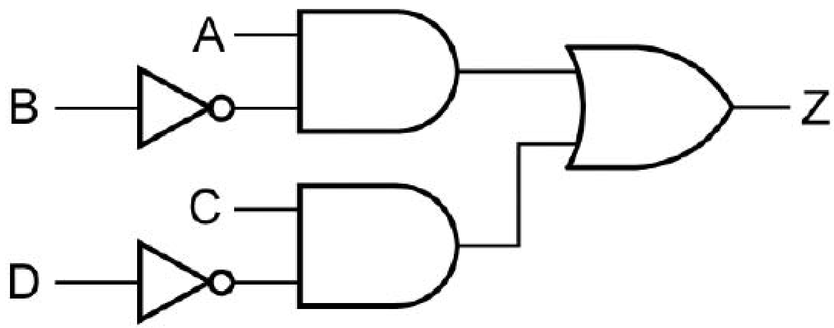
T = Timer has expired

B = Button is pressed

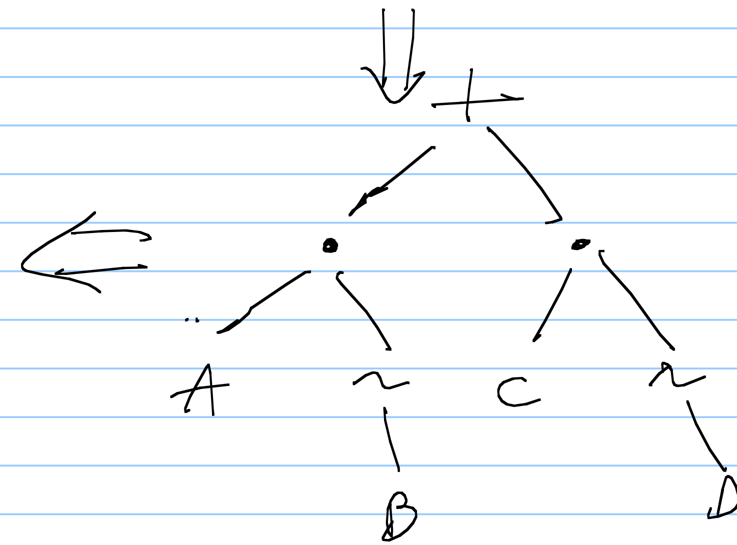


$$Z = SFT' + SFB \quad (= SF(T' + B))$$

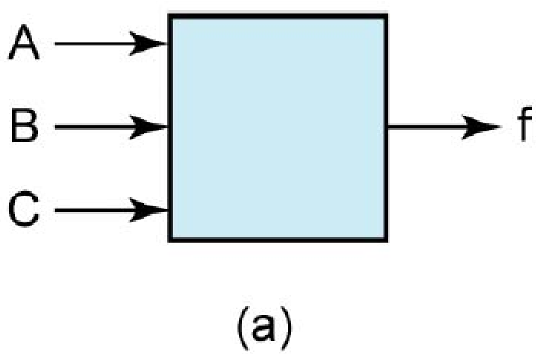
Circuit equivalent to the expression $Z = AB' + CD'$.



Section 4.1, p. 85



Combinatorial design using a truth table (section 4.2)



(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)

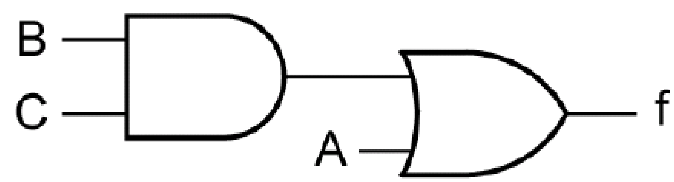
Figure 4-1: **Combinational Circuit with Truth Table**

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

$= A'BC + AB' + AB = \underbrace{A'}_{y'}\underbrace{BC}_{x} + \underbrace{A}_{y} =$

$= [11\,D] = A + BC$

$\Downarrow$



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

$= [9D] = (A+B)(A+B'+C) = ([8D]$, 2nd distributive law$) = A + B(B'+C) = A + BC$ $\Rightarrow$

Section 4.2, p. 86

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

$f = (A'B'C' + A'B'C + A'BC')' = \overbrace{(A+B+C)(A+B+C')(A+B'+C)}^{\Uparrow}$

$= A'B' + A'BC' = A'(B' + BC') = [11D] = A'(B'+C')$

$f = (A'(B'+C'))' = [\text{de Morgan}] = A + (B'+C')' = [\text{de Morgan}] = A + BC$

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

Minterm/maxterm notation with

- Σm (for minterms, sum of products)
- ΠM =(for maxterms, product of sums.

The minterm expansion (sum of products) and maxterm expansion (product of sums) may be found by writing a truth table or by algebraic expansion, as in the following example.

$= a'(b'+d) + acd'$

$$f = a'b' + a'd + acd'$$
$$= a'b'(c+c')(d+d') + a'd(b+b')(c+c') + acd'(b+b')$$
$$= a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + \cancel{a'b'c'd} + \cancel{a'b'cd}$$
$$+ a'bc'd + a'bcd + abcd' + ab'cd' \qquad (4\text{-}9)$$

$$f = \underset{0000}{a'b'c'd'} + \underset{0001}{a'b'c'd} + \underset{0010}{a'b'cd'} + \underset{0011}{a'b'cd} + \underset{0101}{a'bc'd} + \underset{0111}{a'bcd} + \underset{1110}{abcd'} + \underset{1010}{ab'cd'}$$
$$f = \sum m(0, 1, 2, 3, 5, 7, 10, 14) \qquad (4\text{-}10)$$

$=$

Minterm Expansion (p. 89)

Expansion by introduction of missing variables, using $X + X' = 1 \Rightarrow Y(X+X') = Y$

$$f = a'(b'+d) + acd'$$
$$= (a'+cd')(a+b'+d) = (a'+c)(a'+d')(a+b'+d)$$
$$= (a'+bb'+c+dd')(a'+bb'+cc'+d')(a+b'+cc'+d)$$
$$= (a'+bb'+c+d)(a'+bb'+c+d')\cancel{(a'+bb'+c+d')}(a'+bb'+c'+d')$$
$$(a+b'+cc'+d)$$
$$= \underset{1000}{(a'+b+c+d)}\underset{1100}{(a'+b'+c+d)}\underset{1001}{(a'+b+c+d')}\underset{1101}{(a'+b'+c+d')}$$
$$\underset{1011}{(a'+b+c'+d')}\underset{1111}{(a'+b'+c'+d')}\underset{0100}{(a+b'+c+d)}\underset{0110}{(a+b'+c'+d)}$$
$$= \prod M(4, 6, 8, 9, 11, 12, 13, 15) \qquad (4\text{-}11)$$

Maxterm Expansion (p. 90)

Expansion by introduction of missing variables, using $XX' = 0 \Rightarrow Y + XX' = Y$

Table 4-2. General Truth Table for Three Variables

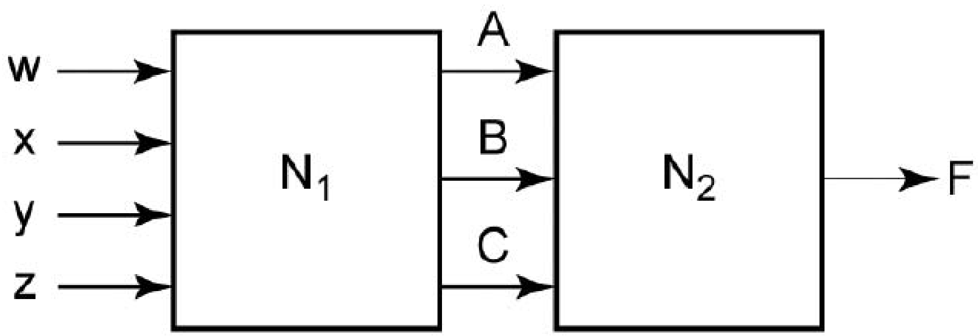
A	B	C	F
0	0	0	a_0
0	0	1	a_1
0	1	0	a_2
0	1	1	a_3
1	0	0	a_4
1	0	1	a_5
1	1	0	a_6
1	1	1	a_7

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

Table 4-3. Conversion of Forms

GIVEN FORM \ DESIRED FORM	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 for F	————	minterm nos. are the same as maxterm nos. of F	list maxterms not present in F

Incompletely specified functions (Section 4.5)



Section 4.5, p. 93

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

A	B	C	F	(1)	(2)	(3)
0	0	0	1			
0	0	1	X	0	1	1
0	1	0	0			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	X	0	0	1
1	1	1	1			

N_1 does not generate these configurations

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

$= A'B'C' + BC$

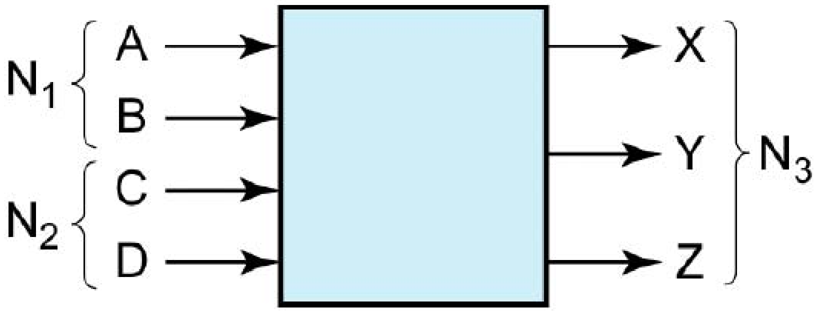
(2) $F = A'B'C' + A'B'C + A'BC + ABC = A'B' + BC$

(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
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., $N_1 =$ A B
$+N_2 =$ C D
$N_3 =$ X Y Z

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

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

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

Design of Binary Adders (4.7)

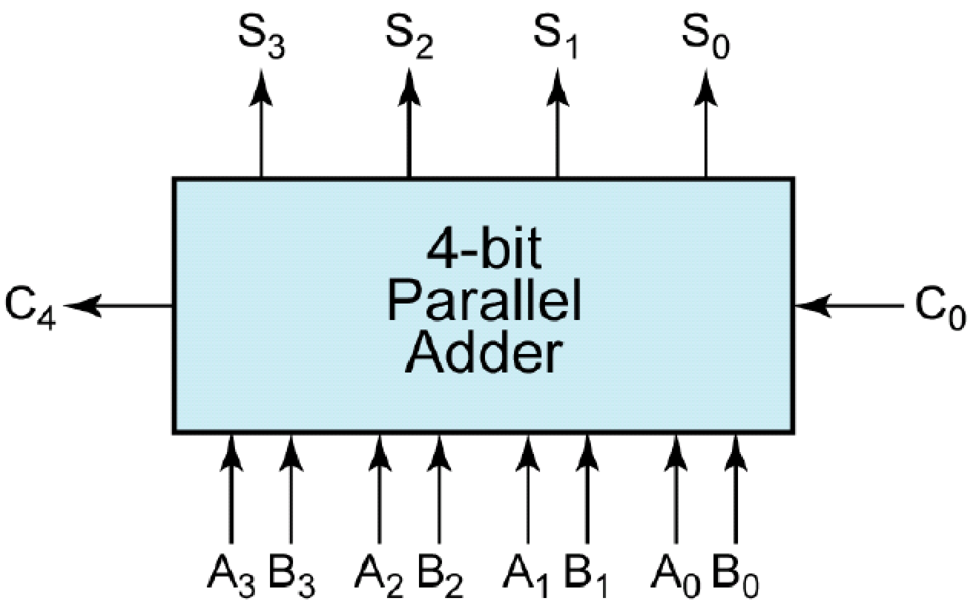


Figure 4-2: **Parallel Adder for 4-Bit Binary Numbers**

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

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

This circuit has nine inputs — too big!

Instead, connect four fulladders.

Ex. :

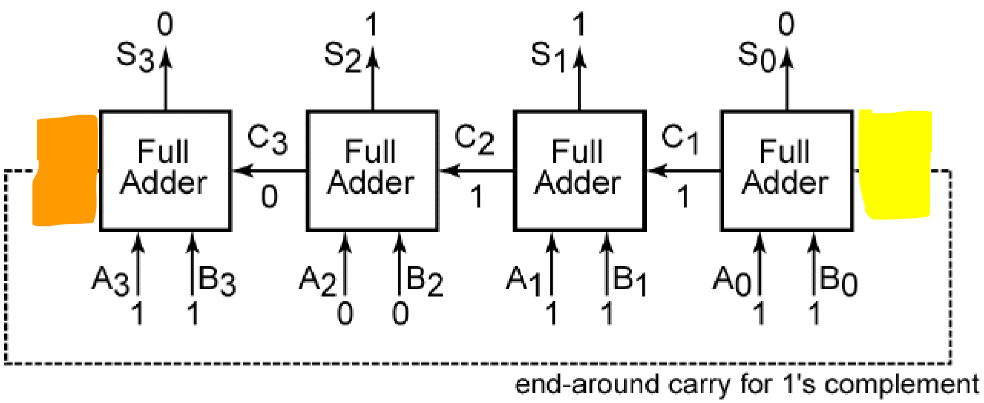
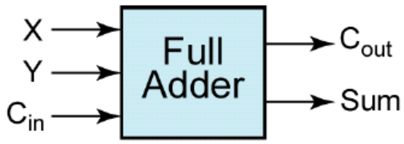


Figure 4-3: **Parallel Adder Composed of Four Full Adders**

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

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



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**

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

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

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

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

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

$= X \oplus (Y \oplus C_{in}) =$ ([3-13]: 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:

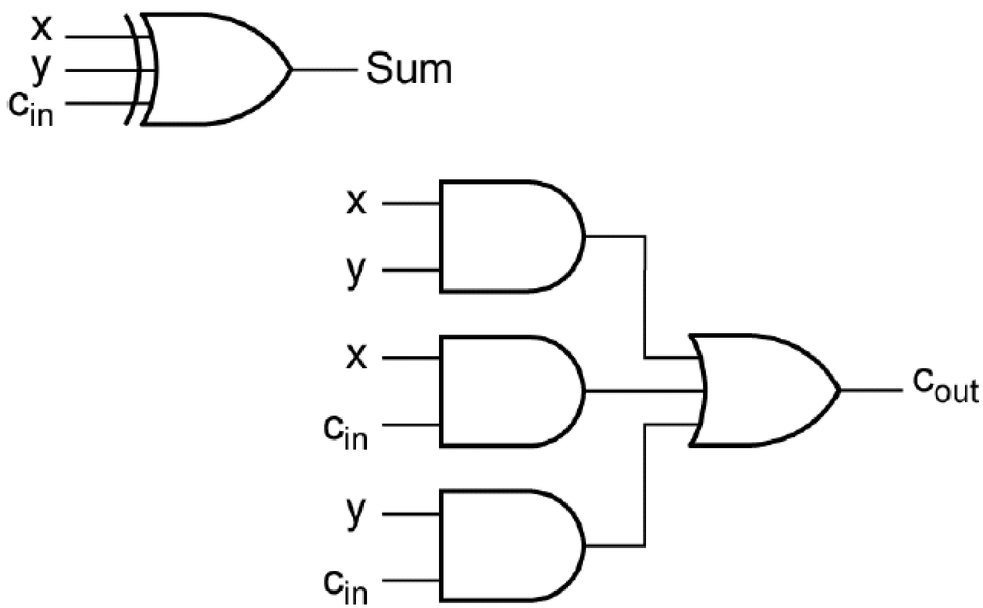


Figure 4-5: **Implementation of Full Adder**