

SPECIFICATION OF COMBINATIONAL SYSTEMS - SUMMARY¹

- HIGH-LEVEL AND BINARY-LEVEL SPECIFICATIONS
- REPRESENTATION OF DATA ELEMENTS BY BINARY VARIABLES; STANDARD CODES FOR POSITIVE INTEGERS AND CHARACTERS 
- REPRESENTATION BY SWITCHING FUNCTIONS AND SWITCHING EXPRESSIONS
- BASIC SWITCHING FUNCTIONS: NOT, AND, OR, XOR, XNOR

- BOOLEAN ALGEBRA (Appendix A) 
- SWITCHING ALGEBRA
- TRANSFORMATION OF SWITCHING EXPRESSIONS USING SWITCHING ALGEBRA
- USE OF VARIOUS SPECIFICATION METHODS:
 - TABLES
 - COMPOSITION OF FUNCTIONS
 - ARITHMETIC, CONDITIONAL AND SWITCHING EXPRESSIONS
- USE OF THE μ VHDL DESCRIPTION LANGUAGE 

COMBINATIONAL SYSTEM

$$z(t) = F(x(t)) \quad \text{or} \quad z = F(x)$$

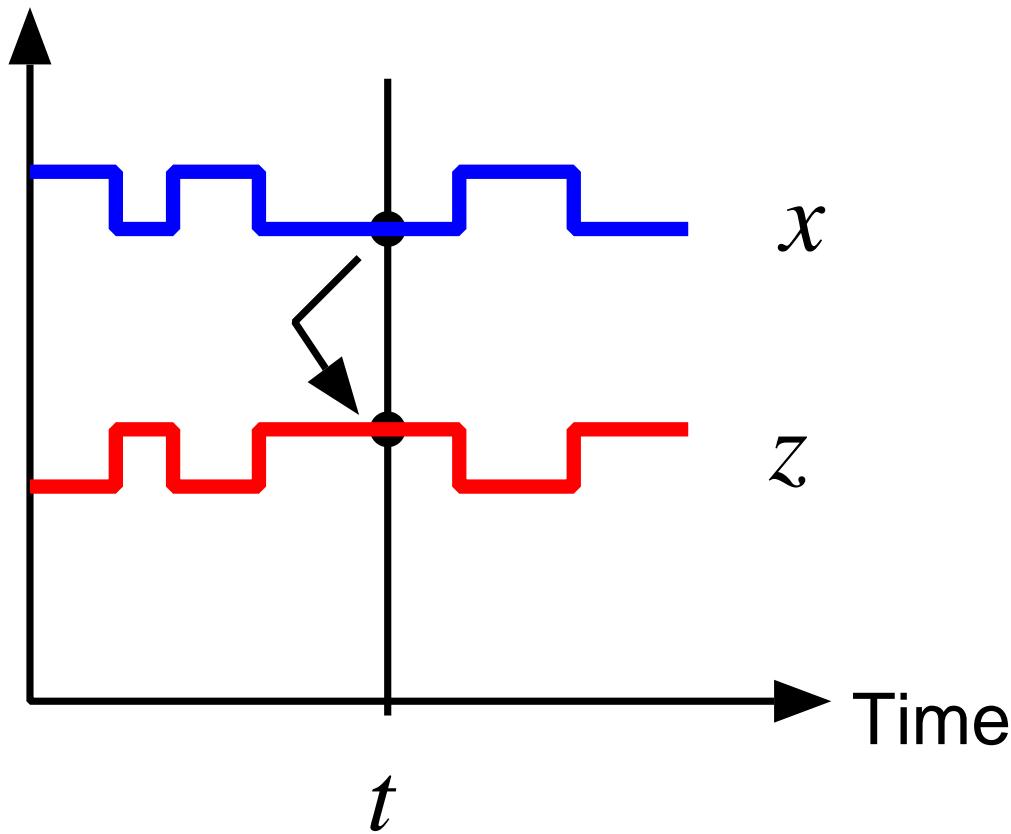


Figure 2.1: Combinational system.

BINARY LEVEL

$$\underline{z}_b = F_b(\underline{x}_b)$$

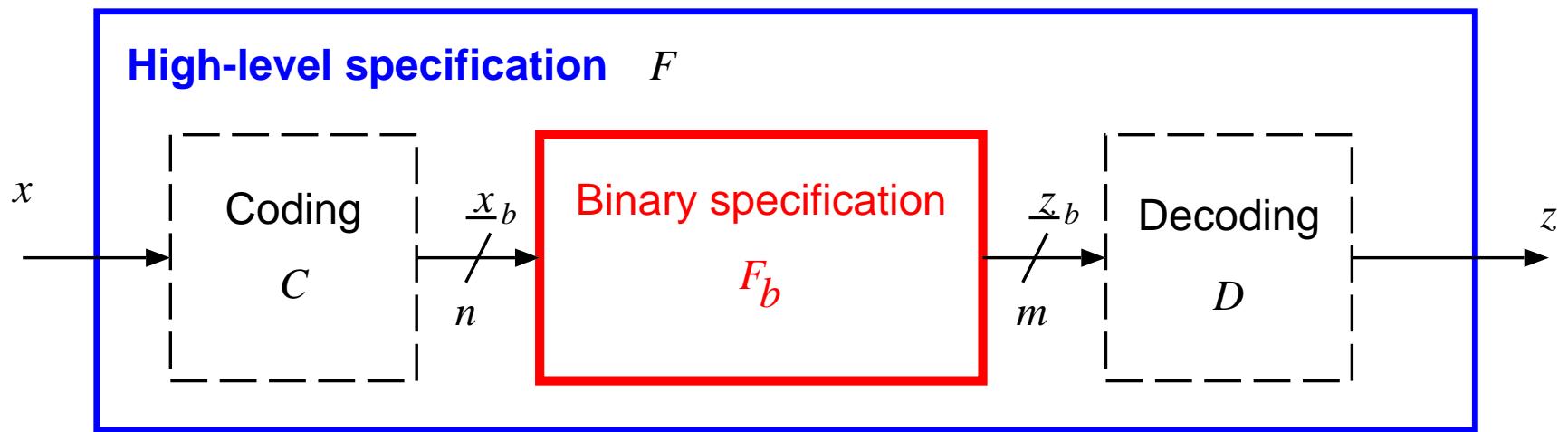


Figure 2.2: High-level and binary-level specification.

Example 2.1:

Input: $x \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Output: $z \in \{0, 1, 2\}$

Function: F is described by the following table

x	0	1	2	3	4	5	6	7	8	9
$z = F(x)$	0	1	2	0	1	2	0	1	2	0

or by the arithmetic expression

$$z = x \bmod 3,$$

Example 2.1 (cont.): BINARY ENCODING OF x AND z

x	0	1	2	3	4	5	6	7	8	9
$\underline{x}_b = C(x)$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001



\underline{z}_b	00	01	10
$z = D(\underline{z}_b)$	0	1	2



Input:

$\underline{x}_b = (x_3, x_2, x_1, x_0), x_i \in \{0, 1\}$

Output:

$\underline{z}_b = (z_1, z_0), z_i \in \{0, 1\}$

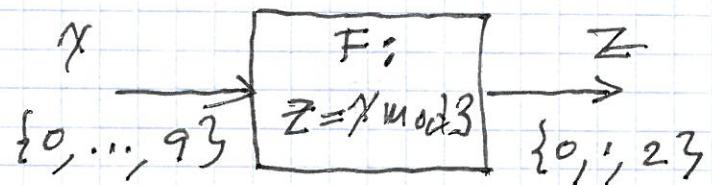
Function: F_b is described by the following table

\underline{x}_b	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
$\underline{z}_b = F_b(\underline{x}_b)$	00	01	10	00	01	10	00	01	10	00

CS MSIA

EXAMPLE 2.1 BLOCK DIAGRAMS

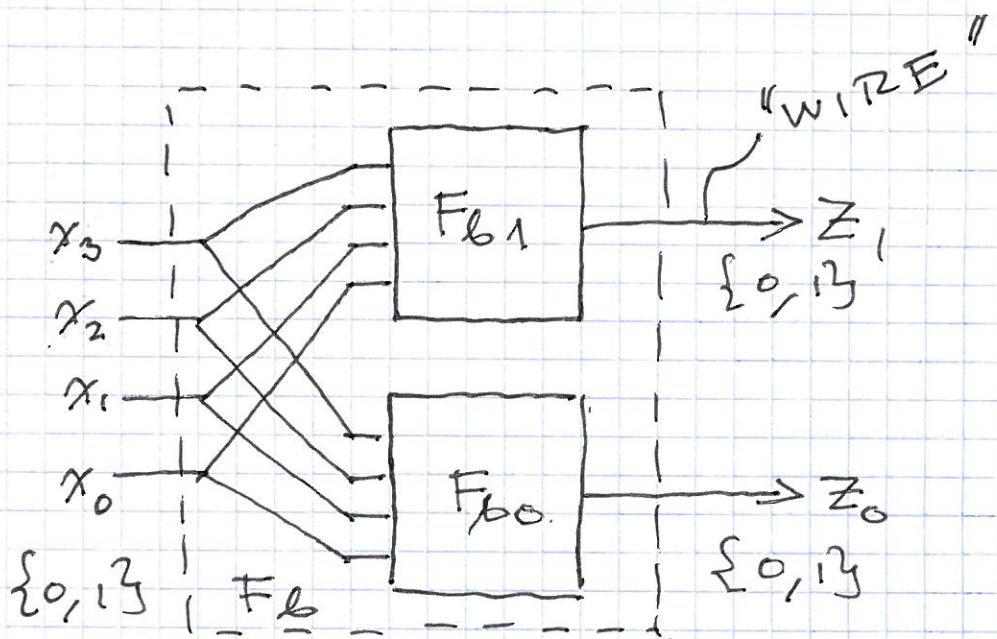
a) HIGH-LEVEL



VARIABLES FROM
INPUT & OUTPUT
SETS

- TYPICALLY
MULTIVALUED

b) BINARY-LEVEL



ALL VARIABLES $\in \{0, 1\}$
(BINARY)



HIGH-LEVEL SPECIFICATION

- SET OF VALUES FOR THE INPUT, *input set*;
- SET OF VALUES FOR THE OUTPUT, *output set*; and
- SPECIFICATION OF THE *input-output function*.

INPUT AND OUTPUT SETS

$$\{UP, DOWN, LEFT, RIGHT, FIRE\}$$

$$\{x \mid (5 \leq x \leq 10^4) \text{ and } (x \bmod 3 = 0)\}$$

Examples of vectors 

Vector type		Example
Digit	$\underline{x} = (x_{n-1}, x_{n-2}, \dots, x_0)$ $x_i \in \{0, 1, 2, \dots, 9\}$	$\underline{x} = (7, 0, 6, 3)$
Character	$\underline{c} = (c_{n-1}, c_{n-2}, \dots, c_0)$ $c_i \in \{ , A, B, \dots, Z\}$	$\underline{c} = (B, O, O, K)$
Set	$\underline{s} = (s_{n-1}, s_{n-2}, \dots, s_0)$ $s_i \in \{\text{red, blue, white}\}$	$\underline{s} = (\text{red, blue, blue})$
Bit	$\underline{y} = (y_{n-1}, y_{n-2}, \dots, y_0)$ $y_i \in \{0, 1\}$	$\underline{y} = (1, 1, 0, 1, 0, 0)$ $\underline{y} = 110100$

INPUT-OUTPUT FUNCTION

1. TABLE

x	z
A	65
B	66
C	67
D	68
E	69

2. ARITHMETIC EXPRESSION

$$z = 3x + 2y - 2$$

3. CONDITIONAL EXPRESSION

$$z = \begin{cases} a + b & \text{if } c > d \\ a - b & \text{if } c = d \\ 0 & \text{if } c < d \end{cases}$$

INPUT-OUTPUT FUNCTION (cont.)

4. LOGICAL EXPRESSION

$z = (\text{SWITCH1} = \text{CLOSED}) \text{ and } (\text{SWITCH2} = \text{OPEN})$
or $(\text{SWITCH3} = \text{CLOSED})$

5. COMPOSITION OF SIMPLER FUNCTIONS

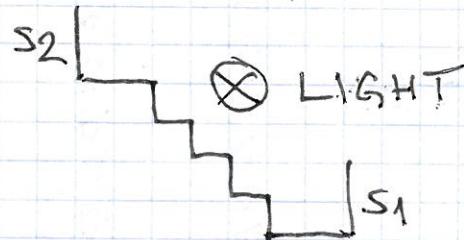
$$\max(v, w, x, y) = \text{GREATER}(v, \text{GREATER}(w, \text{GREATER}(x, y)))$$

in which

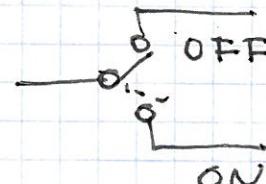
$$\text{GREATER}(a, b) = \begin{cases} a & \text{if } a > b \\ b & \text{otherwise} \end{cases}$$

#4 LOGICAL EXPRESSION

PROBLEM: SPECIFY A SYSTEM TO CONTROL

A STAIRWELL LIGHT FROM TWO FLOORS
USING SWITCHES S2 AND S1

S



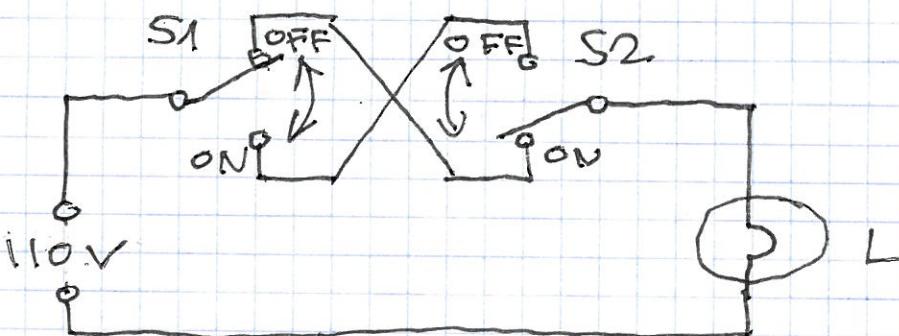
L IS ON IF ((S1 IS OFF) AND (S2 IS ON))

OR ((S1 IS ON) AND (S2 IS OFF))

$$\Rightarrow L = f(S_1, S_2)$$

S1	S2	L
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	OFF

CONTROL SYSTEM:

DARK \rightarrow LIGHTLIGHT \rightarrow DARK

FLIP A SWITCH

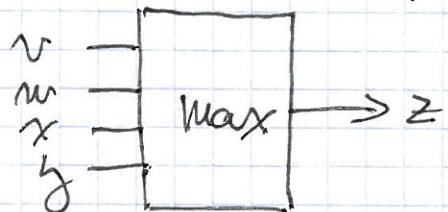
CHALLENGE: EXTEND TO 3 FLOORS!

AT HIGH LEVEL

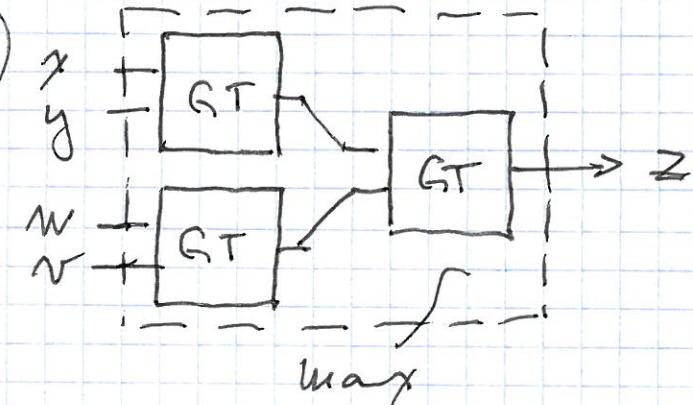
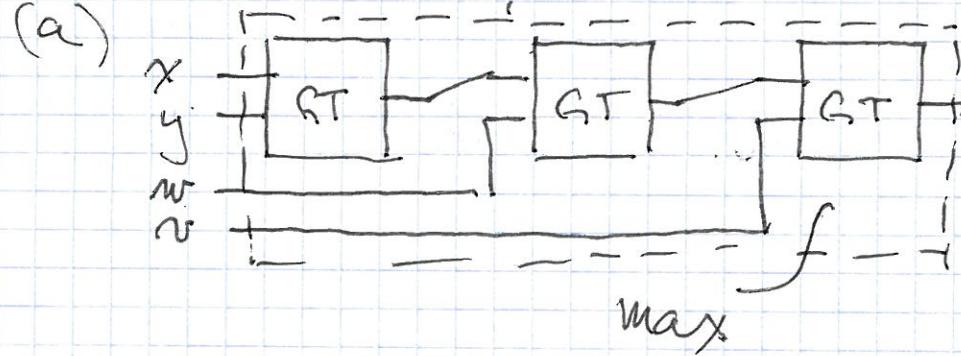
#5 COMPOSITION OF SIMPLER FUNCTIONS

- ONE OF THE MOST IMPORTANT IDEAS IN THE DESIGN OF DIGITAL SYSTEMS
- FOLLOWS A HIERARCHICAL APPROACH
(DIVIDE & CONQUER)

PROBLEM: $Z = \text{Max}(n, m, x, y)$ (+ INTEGERS)



SOLUTIONS:



SCHEME: COST DELAY

a	3	3
---	---	---

b	3	2
---	---	---



Example 2.2

Inputs: $\underline{x} = (x_3, x_2, x_1, x_0)$,
 $x_i \in \{A, B, \dots, Z, a, b, \dots, z\}$
 $y \in \{A, B, \dots, Z, a, b, \dots, z\}$
 $k \in \{0, 1, 2, 3\}$

Outputs: $\underline{z} = (z_3, z_2, z_1, z_0)$,
 $z_i \in \{A, B, \dots, Z, a, b, \dots, z\}$

Function: $z_j = \begin{cases} x_j & \text{if } j \neq k \\ y & \text{if } j = k \end{cases}$

Input: $\underline{x} = (C, A, S, E)$, $y = R$, $k = 1$

Output: $\underline{z} = (C, A, R, E)$

DATA REPRESENTATION AND CODING

$\{Al, Bert, Dave, Jerry, Len\}$



	Fixed-length		Variable-length
	Code 1	Code 2	Code 3
Al	000	0110	01
Bert	010	0101	001
Dave	100	0011	0001
Jerry	110	1001	00001
Len	111	1111	000001



ALPHANUMERIC CODES

Character	Codes	
	ASCII	EBCDIC
A	100 0001	1100 0001
B	100 0010	1100 0010
C	100 0011	1100 0011
:	:	:
Y	101 1001	1110 1000
Z	101 1010	1110 1001
0	011 0000	1111 0000
1	011 0001	1111 0001
2	011 0010	1111 0010
:	:	:
8	011 1000	1111 1000
9	011 1001	1111 1001
blank	010 0000	0100 0000
.	010 1110	0100 1011
(010 1000	0100 1101
+	010 1011	0100 1110
:	:	:

A blank C A B .
 01000001 00100000 01000011 01000001 01000010 00101110

REPRESENTATION OF POSITIVE INTEGERS

Level 1: INTEGER \iff *DIGIT-VECTOR*

Level 2: DIGIT \iff *BIT-VECTOR*

Level 1: Integer (Digit-vector)	5	6	3	0								
Level 2: Bit-vector	1	0	1	1	1	0	0	1	1	0	0	0

1. NUMBER REPRESENTATION SYSTEM (NRS)

SET OF VALUES (FINITE)

$$\{x\} \xrightarrow{\text{NRS}} \{\underline{x}\}$$
SET OF DIGIT-VECTORS

DIGIT-VECTOR

$$\underline{x} = (x_5, x_4, \dots, x_0)_2$$
2 - RADIX

DIGIT SET

$$x_i \in \{0, 1, \dots, 2-1\}$$
(BASE)

$$2 = 4$$

$$\underline{x} = (3, 0, 1, 2)_4 = 3012_4$$

VALUE OF \underline{x}

$$x = \sum_{i=0}^{n-1} x_i \cdot 2^i$$
n - # OF DIGITS

$$\underline{x} = 3012_4 \rightarrow x = 3 \cdot 4^3 + 0 \cdot 4^2 + 1 \cdot 4^1 + 2 \cdot 4^0 = 198_{10}$$

GIVEN β AND n , THE SET OF REPRESENTABLE
VALUES IS

$$0 \leq x \leq \beta^n - 1$$

$$\beta = 2 \quad n = 10 \quad 0 \leq x \leq 1023$$

GIVEN β AND x_{\max}

$$n = \lceil \log_2 x_{\max} \rceil$$

2. REPRESENTATION OF DIGITS

WEIGHTED CODE: DIGIT VALUE of

$$d = \sum_{i=0}^{k-1} d_i w_i$$

FOR $n = 2^k$
(BINARY COMPATIBLE)
→ SEE TABLE 2.2

$$d = \sum_{i=0}^{k-1} d_i 2^i$$

IN GENERAL:

$$w_0 = 2^0 = 1$$

$$w_{i+1} = n w_i / 12$$

$$\text{OR } w_i = 12^i$$

Ex. $n=4$ $k=2$

0L	d_1 , d_0
0	0 0
1	0 1
2	1 0
3	1 1

etc. for
OTHER k 's

FOR $n=10$ SEVERAL POPULAR CODES

→ SEE TABLE 2.3

EXAMPLE OF BINARY CODES FOR DIGITS ($r = 2^k$)

Digit Value (Symbol)	Binary	Quaternary	Octal	Hexadecimal
	$k = 1$	$k = 2$	$k = 3$	$k = 4$
0	d_0	d_1d_0	$d_2d_1d_0$	$d_3d_2d_1d_0$
1	0	00	000	0000
2	1	01	001	0001
3		10	010	0010
4		11	011	0011
5			100	0100
6			101	0101
7			110	0110
8			111	0111
9				1000
10 (A)				1001
11 (B)				1010
12 (C)				1011
13 (D)				1100
14 (E)				1101
15 (F)				1110
				1111

FOR DECIMAL DIGITS:

BCD CODE (BINARY CODED DECIMAL)

WEIGHTS $8, 4, 2, 1$ ($n=4$)

$$5_{10} \leftrightarrow 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \rightarrow 0101_2$$

$$9_{10} \leftrightarrow 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \rightarrow 1001_2$$

2421 CODE

WEIGHTS: $2, 4, 2, 1$

$$5_{10} \leftrightarrow 1011 \text{ OR } 0101$$

Q COMPLEMENT WRT 9, $9-d$, OBTAINED BY
 /,
 COMPLEMENTING EACH BIT WRT 1, $1-d_i$
 → SELF-COMPLEMENTING CODE

USEFUL IN PERFORMING SUBTRACTION

EXCESS-3 CODE: $BCD + 3$ | NOT WEIGHTED;
 $5_{10} = 0101_{BCD} = 1000_{EX-3}$ | SELF-COMPLEMENTING

2-OUT-OF-5 CODE; TWO 1's IN EACH BIT-VECTOR

CODES FOR DECIMAL DIGITS

Digit	BCD			
Value	8421	2421	Excess-3	2-Out-of-5
0	0000	0000	0011	00011
1	0001	0001	0100	11000
2	0010	0010	0101	10100
3	0011	0011	0110	01100
4	0100	0100	0111	10010
5	0101	1011	1000	01010
6	0110	1100	1001	00110
7	0111	1101	1010	10001
8	1000	1110	1011	01001
9	1001	1111	1100	00101

COMPLEMENT $C = 9 - d$

USEFUL IN SUBTRACTION
— HOW TO OBTAIN C?

BCD 9 1 0 0 1
d = 5 0 1 0 1 SUBTRACT
C = 4 0 1 0 0

2421 9 1 1 1 1
d = 5 0 1 0 1 BIT-COMPLEMENT
C = 4 1 0 1 0

EX-3 9 1 1 0 0
d = 5 1 0 0 0 BIT-COMPLEMENT
C = 4 0 1 1 1

2421 & EX-3 ARE SELF-COMPLEMENTING
CODES

ON 2-4-2-1 CODE (TABLE 2-3) [EXERCISE
2.9]

a) IS IT UNIQUE?

b) IF NOT UNIQUE, HOW MANY DIFFERENT
2-4-2-1 CODES ARE THERE?

	2	4	2	1	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	}
	1	0	0	0	}
3	0	0	1	1	}
	1	0	0	1	}
4	0	1	0	0	}
	1	0	1	0	}
5	0	1	0	1	}
	1	0	1	1	}
6	0	1	1	0	}
	1	1	0	0	}
7	0	1	1	1	}
	1	1	0	1	}
8	1	1	1	0	
9	1	1	1	1	

$$2^6 = 64 \text{ CODES}$$

c) ARE ALL OF THESE CODES
SELF-COMPLEMENTING?

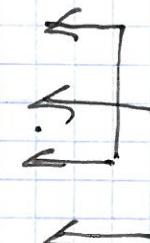
IF NO, HOW MANY SELF-COMPLEMENTING
CODES ARE THERE?

2421 CODE: 64 DIFFERENT CODES; 3 ARE SELF COMPLEMENTARY

CONSIDER

2

0	0	1	0
1	0	0	0
7	0	1	1
1	1	0	1



$\rightarrow \frac{1}{2}$

YES

NO

2-OUT-OF-5 CODE: SINGLE ERROR CAN BE DETECTED

CHECKER C: INPUT $\underline{x} = (x_4, x_3, x_2, x_1, x_0)_2$

$$\Sigma = (s_e, s_i, s_o)$$

$$C = \Sigma'_e \Sigma'_i \Sigma'_o$$

OUTPUT C = $\begin{cases} 1 & \text{IF } \Sigma = \sum_{i=0}^4 x_i = 2 \\ 0 & \text{OTHERWISE} \end{cases}$

($C = 1 \Rightarrow$ NO SINGLE ERROR)

A DESIGN:

$$x_2 + x_1 + x_0 = 2c_a + s_a$$

← FULL ADDER

$$x_4 + x_3 = 2c_b + s_b$$

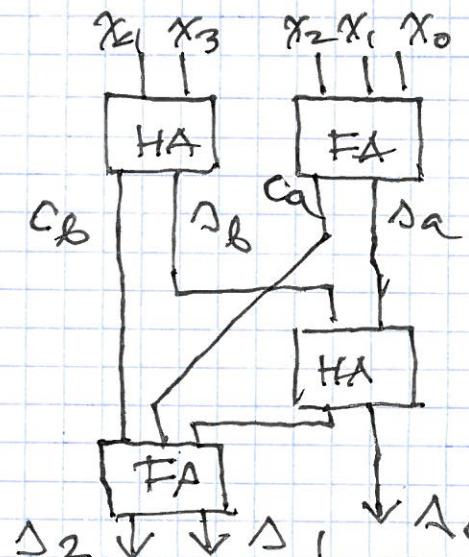
← HALF ADDER

$$\begin{array}{r} c_a \quad s_a \\ \hline \end{array}$$

$$\begin{array}{r} c_b \quad s_b \\ \hline \end{array}$$

$$\begin{array}{r} c_2 \quad s_2 \\ \hline \end{array}$$

$$\begin{array}{r} s_1 \quad s_0 \\ \hline \end{array}$$



1. a) DECIMAL \rightarrow BCD (Ex-3, etc)

G . I O 3
 ↓ ↓
 0110 0001 0000 0011 BCD
 1001 0100 0011 0110 EX-3

b) BCD \rightarrow DECIMAL

SIMILAR FOR OTHER CODES

Binary subtraction diagram:

$$\begin{array}{r} 1000001 \\ - 00101 \\ \hline 1110000 \end{array}$$

The result 1110000 is labeled as BCD.

c) RADIX 2 \leftrightarrow RADIX 2^k
 — MAKE k-BIT GROUPS

$$\underline{X} = \begin{array}{c|cc|c} & 1 & 0 & 0 \\ \hline & 0 & 0 & 0 \end{array} \begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \\ 2 & 1 \end{array} \uparrow \quad 1 \cdot 2^7 + 1 \cdot 2^2 + 1 \cdot 2 + 1 \cdot 1 = 135_{10}$$

EXTEND ————— 

CONVERSIONS

(5)

2. a) DECIMAL TO BINARY (INTEGERS)

- KEEP DIVIDING BY 2 UNTIL QUOTIENT = 1
- TAKE REMAINDERS AS BITS
- LAST QUOTIENT (1) IS THE MOST SIGNIFICANT (MS) BIT

$$\begin{array}{r}
 1 \ 3 \ 5 \longrightarrow 1 \text{ LS} \\
 6 \ 7 \longrightarrow 1 \\
 3 \ 3 \longrightarrow 1 \\
 1 \ 6 \longrightarrow 0 \\
 8 \longrightarrow 0 \\
 4 \longrightarrow 0 \\
 2 \longrightarrow 0 \\
 1 \longrightarrow 1 \text{ MS}
 \end{array}$$

$$\begin{array}{r}
 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \\
 \hline
 128 \\
 \hline
 135
 \end{array}$$

b) DECIMAL TO BINARY (FRACTIONS)

- KEEP MULTIPLYING BY 2 AS NEEDED OR DESIRED
- TAKE INTEGER PART AS BIT AND REMOVE

$$\begin{array}{r}
 0.8 \\
 1.6 \\
 1.2 \\
 0.4 \\
 0.8 \\
 1.6 \\
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 0. \\
 1 \\
 0 \\
 0 \\
 1 \\
 \hline
 \end{array}
 \quad
 \begin{array}{l}
 \left. \begin{array}{r}
 25 \\
 32
 \end{array} \right\} \approx 0.78 \\
 \text{ERROR!} \\
 \text{REPEATS!}
 \end{array}$$



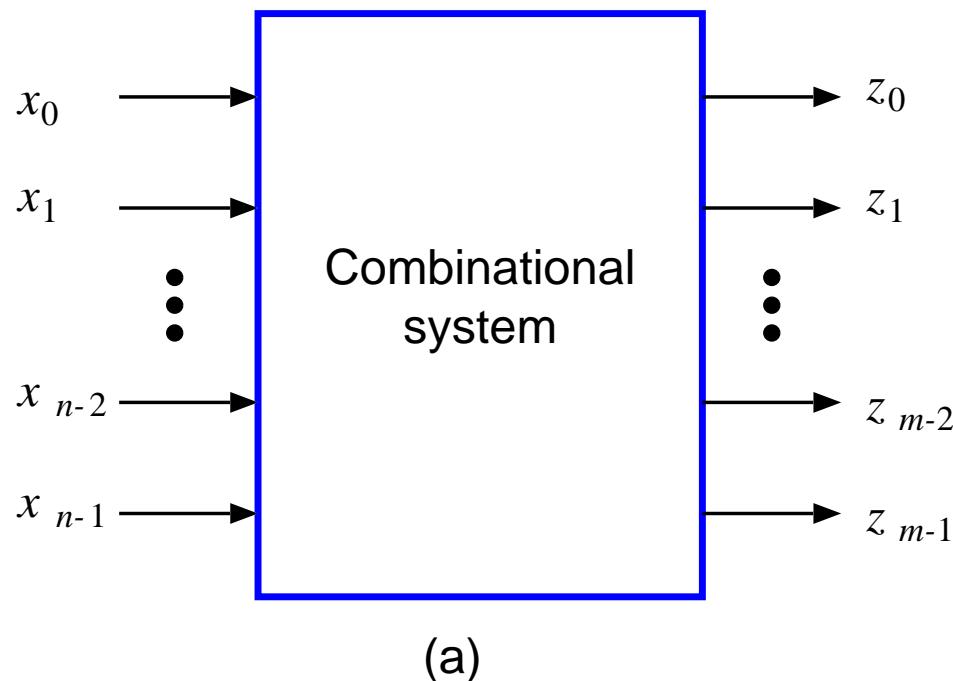
Digit	Gray code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

GRAY CODE BIT-VECTOR REPRESENTATION OF A DIGIT-VECTOR

9	8	7	6												
1	1	0	1	1	1	0	0	0	1	0	0	0	1	0	1

BINARY-LEVEL SPECIFICATION OF C - SYSTEMS

SWITCHING FUNCTIONS



$$\{0,1\}^3 \longrightarrow \{0,1\}$$

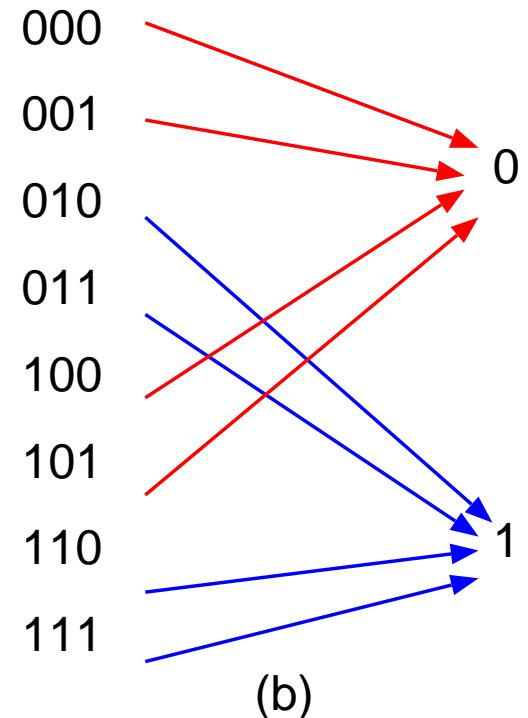


Figure 2.7: a) BINARY COMBINATIONAL SYSTEM; b) A SWITCHING FUNCTION FOR $n = 3$

TABULAR REPRESENTATION OF SWITCHING FUNCTIONS²²

n -tuple notation		Simplified notation	
$x_2x_1x_0$	$f(x_2, x_1, x_0)$	j	$f(j)$
0 0 0	0	0	0
0 0 1	0	1	0
0 1 0	1	2	1
0 1 1	1	3	1
1 0 0	0	4	0
1 0 1	0	5	0
1 1 0	1	6	1
1 1 1	1	7	1

2D TABULAR REPRESENTATION

		$x_2x_1x_0$							
		000	001	010	011	100	101	110	111
x_4x_3	00	0	0	1	1	0	1	1	1
01	01	0	1	1	1	0	1	1	1
10	10	1	1	0	1	1	0	1	1
11	11	0	1	0	1	1	0	1	0

f

More compact



IMPORTANT SWITCHING FUNCTIONS



Table 2.4: SWITCHING FUNCTIONS OF ONE VARIABLE

x	f_0 0-CONSTANT (always 0)	f_1 IDENTITY (equal to x)	f_2 COMPLEMENT (NOT)	f_3 1-CONSTANT (always 1)
0	0	0	1	1
1	0	1	0	1

Table 2.5: SWITCHING FUNCTIONS OF TWO VARIABLES

Function	x_1x_0				
	00	01	10	11	
f_0	0	0	0	0	
f_1	0	0	0	1	AND
f_2	0	0	1	0	
f_3	0	0	1	1	
f_4	0	1	0	0	
f_5	0	1	0	1	
f_6	0	1	1	0	EXCLUSIVE-OR (XOR)
f_7	0	1	1	1	OR
f_8	1	0	0	0	NOR
f_9	1	0	0	1	EQUIVALENCE (EQU)
f_{10}	1	0	1	0	
f_{11}	1	0	1	1	
f_{12}	1	1	0	0	
f_{13}	1	1	0	1	
f_{14}	1	1	1	0	NAND
f_{15}	1	1	1	1	



INCOMPLETE SWITCHING FUNCTIONS

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	—
1	0	0	1
1	0	1	0
1	1	0	—
1	1	1	1



EXAMPLE OF AN INCOMPLETE SF:

$$x \in \{0, \dots, 9\}$$

$$\underline{x} = \{0000, 0001, \dots, 1001\}_{BCD}$$

$$z = f(x) = \begin{cases} 1 & \text{if } x \text{ IS ODD} \\ 0 & \text{OTHERWISE} \end{cases}$$

$\{1010, 1011, \dots, 1111\}$ NEVER OCCUR

j	z
0	0
1	1
2	0
3	1
4	0
5	1
6	0
7	1
8	0
9	1
10	—
11	—
12	—
13	—
14	—
15	—

$$d.c.-SET = \{10, 11, 12, \dots, 15\}$$

$$one-SBT = \{1, 3, 5, 7, 9\}$$

$$zero-SET = \{0, 2, 4, 6, 8\}$$

Can be represented with any of the following

- [*one-set(1,4,7), zero-set(0,2,5)*]
- [*one-set(1,4,7), dc-set(3,6)*]
- [*zero-set(0,2,5), dc-set(3,6)*]

COMPOSITION OF SWITCHING FUNCTIONS



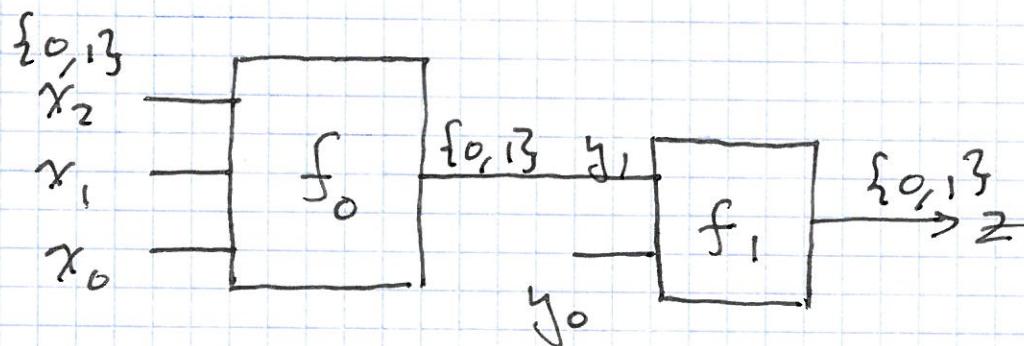
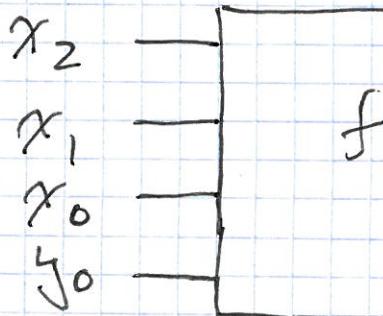
$$\text{AND}(x_3, x_2, x_1, x_0) = \text{AND}(\text{AND}(x_3, x_2), \text{AND}(x_1, x_0))$$

$$\text{XOR}(x_1, x_0) = \text{OR}(\text{AND}(\text{NOT}(x_0), x_1), \text{AND}(x_0, \text{NOT}(x_1)))$$

$$\begin{aligned}\text{MAJ}(x_3, x_2, x_1, x_0) = & \text{OR}(\text{AND}(x_3, x_2, x_1), \text{AND}(x_3, x_2, x_0), \\ & \text{AND}(x_3, x_1, x_0), \text{AND}(x_2, x_1, x_0))\end{aligned}$$



EXAMPLE OF COMPOSITION OF SFs

 $\{0,1\}$  $f_0,13$ 

$$\text{MAJ}(x_3, x_2, x_1, x_0) = \begin{cases} 1 & \text{IF 3 OR 4 VARIABLES = 1} \\ 0 & \text{OTHERWISE} \end{cases}$$

x_3	x_2	x_1	x_0	$\text{MAJ}(x)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	0	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

 $\text{MAJ}(x)$ $=$ $\overline{\text{OR}}($ $\text{AND}(x_2, x_1, x_0),$ $\text{AND}(x_3, x_1, x_0),$ $\text{AND}(x_3, x_2, x_0),$ $\text{AND}(x_3, x_2, x_1))$

-
1. Symbols 0 and 1 are SEs.
 2. A symbol representing a binary variable is a SE.
 3. If A and B are SEs, then
 - $(A)'$ is a SE. This is referred to as “ A complement.” Sometimes we use \overline{A} to denote complementation.
 - $(A) + (B)$ is a SE. This is referred as “ A OR B ”; it is also called “ A plus B ” or “sum” due to the similarity with the corresponding arithmetic symbol.
 - $(A) \cdot (B)$ is a SE. This is referred to as “ A AND B ”; it is also called “ A times B ” or “product” due to the similarity with the corresponding arithmetic symbol.

PRECEDENCE RULES: ' \cdot ' PRECEDES ' $+$ ' WHICH PRECEDES ' $'$

WELL-FORMED SWITCHING EXPRESSIONS ARE:

$$x_0 \quad x_1 + x_2 x'_3 \quad 1 + 0(x + y)$$

$(x_1 + 'x_2 +)x_3$ AND

“THIS IS A SWITCHING EXPRESSION”

ARE NOT WELL-FORMED SWITCHING EXPRESSIONS.

BOOLEAN ALGEBRAS

- IMPORTANT CLASS OF ALGEBRAS EXTENSIVELY USED FOR MANY PURPOSES
- BASIS OF THE *SWITCHING ALGEBRA* (SA) USED FOR FORMAL TREATMENT OF SWITCHING CIRCUITS:
 - TRANSFORMATION OF SWITCHING EXPRESSIONS
 - IDENTITIES FROM BA ENABLE GRAPHICAL AND TABULAR TECHNIQUES FOR MINIMIZATION OF SWITCHING EXPRESSIONS LEADING TO SIMPLER/FASTER CIRCUITS



A **BOOLEAN ALGEBRA** IS A TUPLE $\{B, +, \cdot\}$:

- B IS A SET OF ELEMENTS;
- $+$ AND \cdot ARE BINARY OPERATIONS APPLIED OVER THE ELEMENTS OF B ,

SATISFYING THE FOLLOWING POSTULATES:

P1: If $a, b \in B$, then

$$\begin{aligned} \text{(i)} \quad & a + b = b + a \\ \text{(ii)} \quad & a \cdot b = b \cdot a \end{aligned}$$

That is, $+$ and \cdot are COMMUTATIVE.

P2: If $a, b, c \in B$, then

$$\begin{aligned} \text{(i)} \quad & a + (b \cdot c) = (a + b) \cdot (a + c) \\ \text{(ii)} \quad & a \cdot (b + c) = (a \cdot b) + (a \cdot c) \end{aligned}$$

P3: THE SET B HAS TWO DISTINCT *IDENTITY ELEMENTS*, DENOTED AS 0 AND 1 , SUCH THAT FOR EVERY ELEMENT IN B

- (i) $0 + a = a + 0 = a$
- (ii) $1 \cdot a = a \cdot 1 = a$

THE ELEMENTS 0 AND 1 ARE CALLED THE *ADDITIVE IDENTITY ELEMENT* AND THE *MULTIPLICATIVE IDENTITY ELEMENT*, respectively.

P4: FOR EVERY ELEMENT $a \in B$ THERE EXISTS AN ELEMENT a' , CALLED THE *COMPLEMENT* OF a , SUCH THAT

- (i) $a + a' = 1$
- (ii) $a \cdot a' = 0$

COMMENTS

- SYMBOLS $+$ AND \cdot ARE NOT THE ARITHMETIC ADDITION AND MULTIPLICATION SYMBOLS

For convenience $+$ and \cdot called “plus” and “times”

Expressions $a + b$ and $a \cdot b$ called “sum” and “product,”
 $+$ and \cdot also called “OR” and “AND”

- THE ELEMENTS OF THE SET B ARE CALLED CONSTANTS:
0 AND 1 ARE CONSTANTS
- SYMBOLS REPRESENTING ARBITRARY ELEMENTS OF B ARE VARIABLES: a, b and c in the postulates are variables.

- *PRECEDENCE ORDERING DEFINED ON THE OPERATORS:* \cdot HAS PRECEDENCE OVER $+$.

$$a + (b \cdot c) \quad \text{can be written as} \quad a + bc$$

- *SYMBOLS a, b, c, \dots IN THEOREMS AND POSTULATES ARE GENERIC VARIABLES:* THEY CAN BE SUBSTITUTED BY
 - COMPLEMENTED VARIABLES
 - EXPRESSIONS (FORMULAS)

IMPORTANT THEOREMS IN BOOLEAN ALGEBRA

Theorem 2 PRINCIPLE OF DUALITY.

EVERY ALGEBRAIC IDENTITY DEDUCIBLE FROM THE POSTULATES OF A BOOLEAN ALGEBRA REMAINS VALID IF

- *the operations + and · are interchanged throughout; and*
- *the identity elements 0 and 1 are also interchanged throughout.*
- This theorem is useful because it reduces the number of different theorems that must be proven: every theorem has its dual.

Theorem 6 IDEMPOTENT LAW.

For every $a \in B$

$$(1) \quad a + a = a$$

$$(2) \quad a \cdot a = a$$

Proof:

$$\begin{aligned}
 (1): \quad a + a &= (a + a) \cdot 1 && \text{P3 (ii)} \\
 &= (a + a) \cdot (a + a') && \text{P4 (i)} \\
 &= (a + (a \cdot a')) && \text{P2 (i)} \\
 &= a + 0 && \text{P4 (ii)} \\
 &= a && \text{P3 (i)}
 \end{aligned}$$

$$(2): \quad \text{duality}$$

■

Theorem 8 ABSORPTION LAW. For every pair of elements $a, b \in B$,

$$(1) \quad a + a \cdot b = a$$

$$(2) \quad a \cdot (a + b) = a$$

Proof:

by

$$\begin{aligned} (1): \quad a + ab &= a \cdot 1 + ab && \text{P3 (ii)} \\ &= a(1 + b) && \text{P2 (ii)} \\ &= a(b + 1) && \text{P1 (i)} \\ &= a \cdot 1 && \text{Theorem 4 (1)} \quad \text{QED} \\ &= a && \text{P3 (ii)} \end{aligned}$$

$$(2) \qquad \qquad \qquad \text{duality}$$

■

OTHER EXAMPLES OF BOOLEAN ALGEBRAS

ALGEBRA OF SETS. The elements of B : set of all subsets of a set S ($P(S)$). The operations: set-union (\cup) and set-intersection (\cap).

$$M = (P(S), \cup, \cap)$$

ALGEBRA OF LOGIC (PROPOSITIONAL CALCULUS).

The elements B : T and F (true and false). The operations: LOGICAL AND and LOGICAL OR.

This algebra is isomorphic with the switching algebra.

- *Switching algebra:*

two elements 0 and 1

operations +, ·, and '

+	0	1	.	0	1	'	
0	0	1	0	0	0	0	1
1	1	1	1	0	1	1	0

$$E(x_2, x_1, x_0) = x_2 + x'_2 x_1 + x_1 x'_0$$



The value of E for assignment $(1, 0, 1)$ is

$$E(1, 0, 1) = 1 + 1' \cdot 0 + 0 \cdot 1' = 1 + 0 + 0 = 1$$

REPRESENTING SFs BY SWITCHING EXPRESSIONS

$E(x_2, x_1, x_0) = x_2 + x'_2x_1 + x_1x'_0$ **represents** f :

$x_2x_1x_0$	f
000	0
001	0
010	1
011	1
100	1
101	1
110	1
111	1



SWITCHING EXPRESSIONS FOR BASIC FUNCTIONS



	2 variables	n variables
AND	x_1x_0	$x_{n-1}x_{n-2}\dots x_0$
OR	$x_1 + x_0$	$x_{n-1} + x_{n-2} + \dots + x_0$
XOR	$x_1x'_0 + x'_1x_0 = x_1 \oplus x_0$	
EQUIV	$x'_1x'_0 + x_1x_0$	
NAND	$(x_1x_0)' = x'_1 + x'_0$	$(x_{n-1}x_{n-2}\dots x_0)' = x'_{n-1} + x'_{n-2} + \dots + x'_0$
NOR	$(x_1 + x_0)' = x'_1x'_0$	$(x_{n-1} + x_{n-2} + \dots + x_0)' = x'_{n-1}x'_{n-2}\dots x'_0$

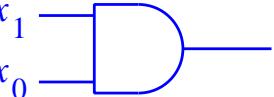
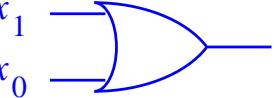
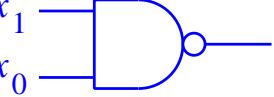
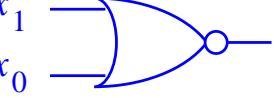
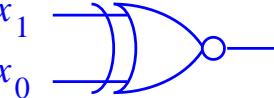
Gate type	Symbol	Switching expression
NOT	 or 	$z = x'$
AND		$z = x_1 x_0$
OR		$z = x_1 + x_0$
NAND		$z = (x_1 x_0)'$
NOR		$z = (x_1 + x_0)'$
XOR		$\begin{aligned} z &= x_1 x_0' + x_1' x_0 \\ &= x_1 \oplus x_0 \end{aligned}$
XNOR		$z = x_1' x_0' + x_1 x_0$

Figure 2.8: Gate symbols

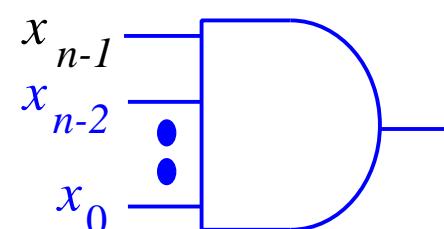
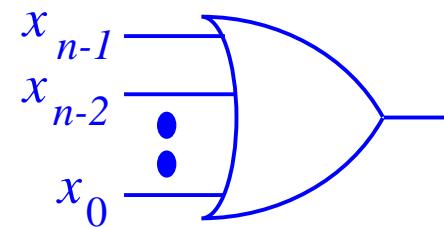
Gate type	Symbol	Switching expression
AND		$z = x_{n-1} x_{n-2} \dots x_0$
OR		$z = x_{n-1} + x_{n-2} \dots + x_0$

Figure 2.9: n -input AND and OR GATE SYMBOLS

EQUIVALENT SWITCHING EXPRESSIONS

W and Z SWITCHING EXPRESSIONS ARE EQUIVALENT

$$\begin{aligned} W &= x_1 x_0 + x'_1 \\ Z &= x'_1 + x_0 \end{aligned}$$


CORRESPONDING SWITCHING FUNCTIONS ARE

$x_1 x_0$	W	Z
00	1	1
01	1	1
10	0	0
11	1	1

EQUIVALENCE CLASSES

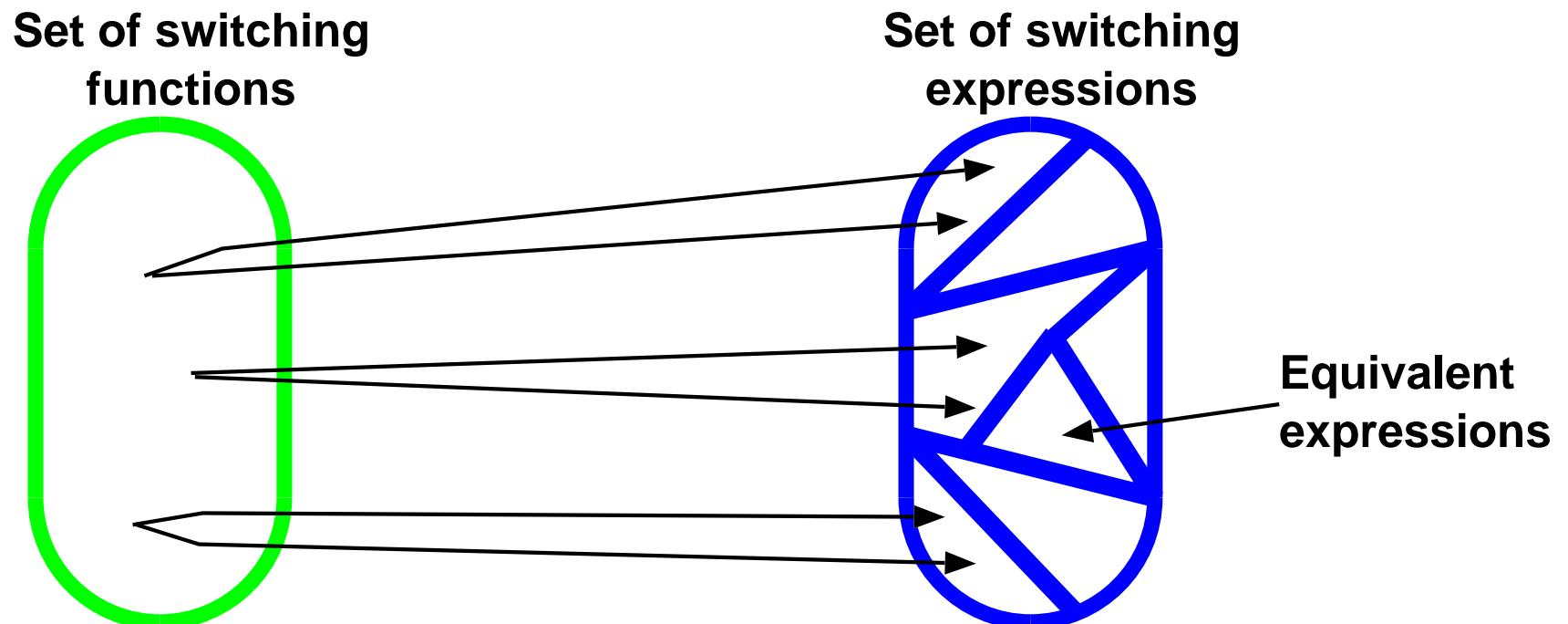


Figure 2.10: CORRESPONDENCE AMONG SFs AND SEs

ALGEBRAIC METHOD OF OBTAINING EQUIVALENT EXPRESSIONS

● MAIN IDENTITIES OF BOOLEAN ALGEBRA

1.	$a + b = b + a$	$ab = ba$	Commutativity
2.	$a + (bc) = (a + b)(a + c)$	$a(b + c) = (ab) + (ac)$	Distributivity
3.	$a + (b + c) = (a + b) + c$ $= a + b + c$	$a(bc) = (ab)c$ $= abc$	Associativity
4.	$a + a = a$	$aa = a$	Idempotency
5.	$a + a' = 1$	$aa' = 0$	Complement
6.	$1 + a = 1$	$0a = 0$	
7.	$0 + a = a$	$1a = a$	Identity
8.	$(a')' = a$		Involution
9.	$a + ab = a$	$a(a + b) = a$	Absorption
10.	$a + a'b = a + b$	$a(a' + b) = ab$	Simplification
11.	$(a + b)' = a'b'$	$(ab)' = a' + b'$	DeMorgan's Law

— PROOFS

10: (i) $a + a'b = a + b$

SIMPLIFICATION

(ii) $a(a' + b) = ab$

SHOW (i): $a + a'b = (a + a')(a + b)$ — BY DISTRIBUTIVITY P2(i)

$$\begin{aligned} &= 1 \cdot (a + b) \\ &= a + b \end{aligned}$$

— BY COMPLEMENT P9(i)

— BY IDENTITY P3(i)

#11 (DEMORGAN'S LAW)

(i) $(a + b)' = a'b'$

(ii) $(ab)' = a'b'$

SHOW (i): FIRST PROVE $(a + b)' = (a'b')'$ (SAME AS (i) !)

BY P4: $(a + b) + a'b' = 1$

$(a + b)(a'b') = 0$

$$\begin{aligned} (a + b) + a'b' &= [(a + b) + a'][[(a + b) + b']] \\ &= 1 \cdot 1 \end{aligned}$$

P2(i)

$= (b+1)(a+1) = 1 \cdot 1 = 1$ ✓

$(a + b)(a'b') = aa'b' + ba'b' = 0 + 0 = 0$ ✗

EXAMPLE

SHOW THAT E_1 AND E_2 ARE EQUIVALENT:

$$\begin{aligned} E_1(x_2, x_1, x_0) &= x_2x_1 + x_2x'_1 + x_2x_0 \\ E_2(x_2, x_1, x_0) &= x_2 \end{aligned}$$

$$\begin{aligned} x_2x_1 + x_2x'_1 + x_2x_0 &= x_2(x_1 + x'_1) + x_2x_0 && \text{using } ab + ac = a(b + c) \\ &= x_2 \cdot 1 + x_2x_0 && \text{using } a + a' = 1 \\ &= x_2(1 + x_0) && \text{using } ab + ac = a(b + c) \\ &= x_2 \cdot 1 && \text{using } 1 + a = 1 \\ &= x_2 && \text{using } a \cdot 1 = a \end{aligned}$$

SUM OF PRODUCTS AND SUM OF MINTERMS

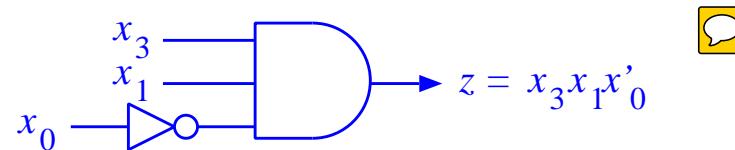
LITERALS

x, y, z', x'

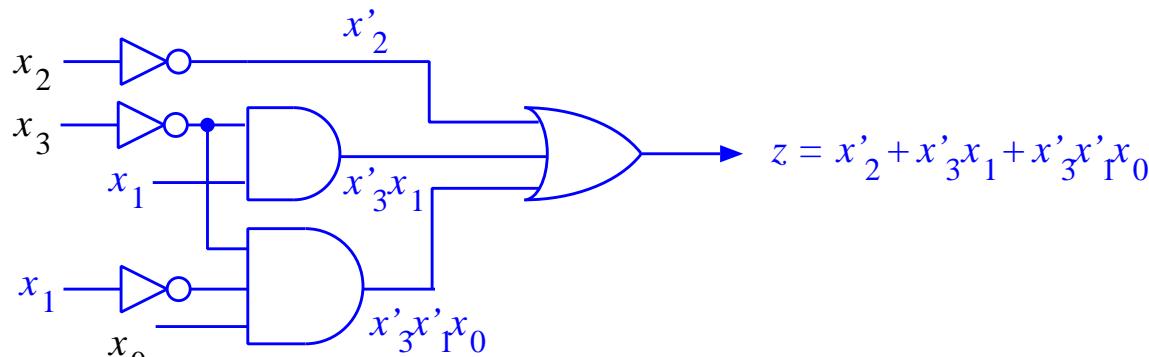
PRODUCT TERMS

$x_0, x_2x_1, x_3x_1x'_0$

SUM OF PRODUCTS (SP) $x'_2 + x_3x'_1 + x'_3x'_1x_0$



(a)



(b)

Figure 2.11: SUM OF PRODUCTS AND AND-OR GATE NETWORK: a) PRODUCT TERM. b) SUM OF PRODUCTS.

MINTERM NOTATION



41

$$x_i \longleftrightarrow 1; \quad x'_i \longleftrightarrow 0$$

MINTERM m_j (product of all n variables), j INTEGER

EXAMPLE: MINTERM $x_3x'_2x'_1x_0$ DENOTED m_9
BECAUSE $1001 = 9$

$$m_j(\underline{a}) = \begin{cases} 1 & \text{if } a = j \\ 0 & \text{otherwise} \end{cases}$$

$$a = \sum_{i=0}^{n-1} a_i 2^i$$

EXAMPLE: $m_{11} = x_3x'_2x_1x_0$
– HAS VALUE 1 ONLY FOR $\underline{a} = (1, 0, 1, 1)$

MINTERM FUNCTIONS

$x_2x_1x_0$	m_0 $x'_2x'_1x'_0$	m_1 $x'_2x'_1x_0$	m_2 $x'_2x_1x'_0$	m_3 $x'_2x_1x_0$	m_4 $x_2x'_1x'_0$	m_5 $x_2x'_1x_0$	m_6 $x_2x_1x'_0$	m_7 $x_2x_1x_0$
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	1	0
111	0	0	0	0	0	0	0	1

EXAMPLE

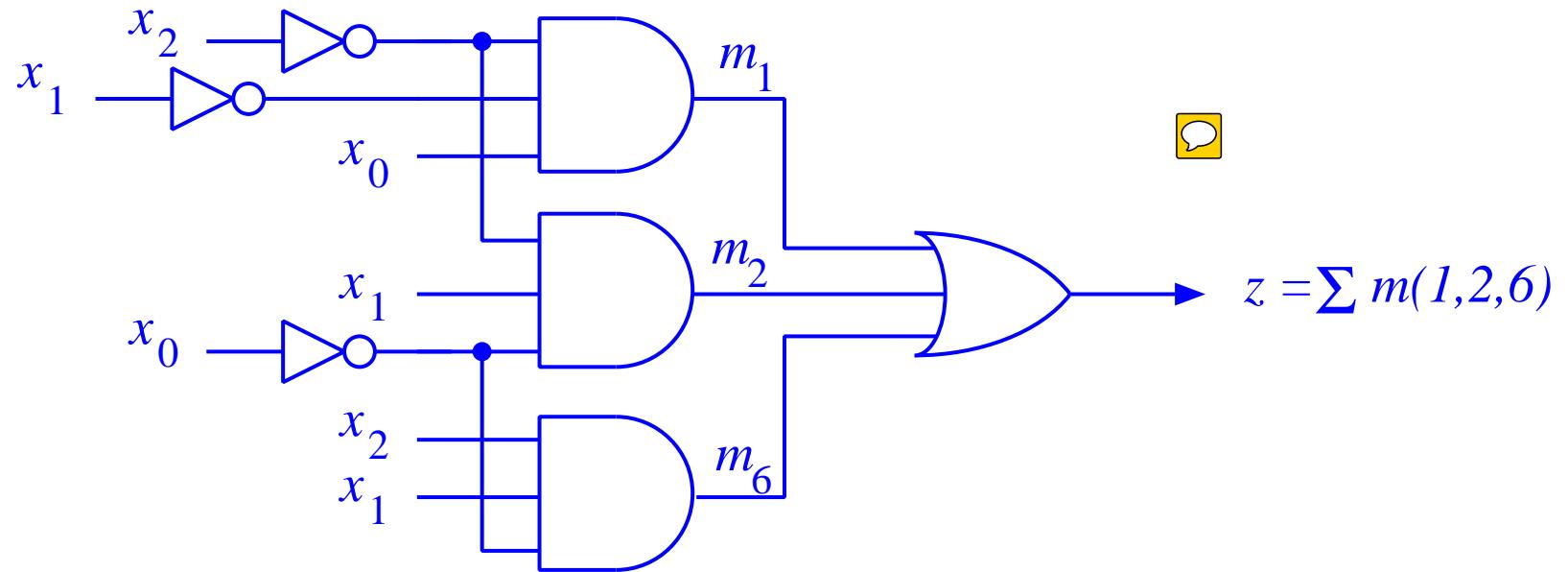


Figure 2.12: GATE NETWORK CORRESPONDING TO $E(x_2, x_1, x_0) = \sum m(1, 2, 6)$.

Example 2.12: TABLE → SUM OF MINTERMS

j	x_2	x_1	x_0	f
0	000			0
1	001			0
2	010			1
3	011			1
4	100			0
5	101			1
6	110			0
7	111			0

$$E = \sum m(2, 3, 5) = x'_2 x_1 x'_0 + x'_2 x_1 x_0 + x_2 x'_1 x_0$$

CONVERSION TO SUM OF MINTERMS

1. CONVERT TO EQUIVALENT SUM OF PRODUCTS

$$\begin{aligned} E(x_2, x_1, x_0) &= (x_2 x_1)' x_0 \\ &= (x_2' + x_1') x_0 \\ &= x_2' x_0 + x_1' x_0 \end{aligned}$$

2. CONVERT PRODUCT TERMS TO MINTERMS

$$\begin{aligned} E(x_2, x_1, x_0) &= x_2' x_0 + x_1' x_0 \\ &= x_2' x_0 (x_1 + x_1') + x_1' x_0 (x_2 + x_2') \\ &= x_2' x_1 x_0 + x_2' x_1' x_0 + x_2 x_1' x_0 + x_2' x_1' x_0 \end{aligned}$$

3. ELIMINATE REPEATED MINTERMS

$$E(x_2, x_1, x_0) = x'_2x'_1x_0 + x'_2x_1x_0 + x_2x'_1x_0$$



Example 2.13: CONVERSION TO SUM OF MINTERMS

$$\begin{aligned} E(x_2, x_1, x_0) &= x_2x'_1 + x_2x'_0 + x_1x'_0 \\ &= x_2x'_1(x_0 + x'_0) + x_2x'_0(x_1 + x'_1) + x_1x'_0(x_2 + x'_2) \\ &= x_2x'_1x_0 + x_2x'_1x'_0 + x_2x'_0x_1 + x_2x'_0x'_1 + x_1x'_0x_2 + x_1x'_0x'_2 \\ &= x'_2x_1x'_0 + x_2x'_1x'_0 + x_2x'_1x_0 + x_2x_1x'_0 \\ &= \Sigma m(2, 4, 5, 6) \end{aligned}$$

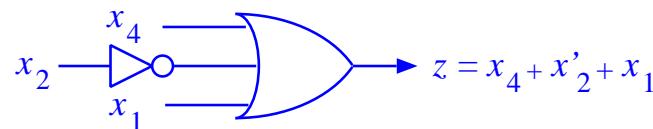
PRODUCT OF SUMS AND PRODUCT OF MAXTERMS

SUM TERMS

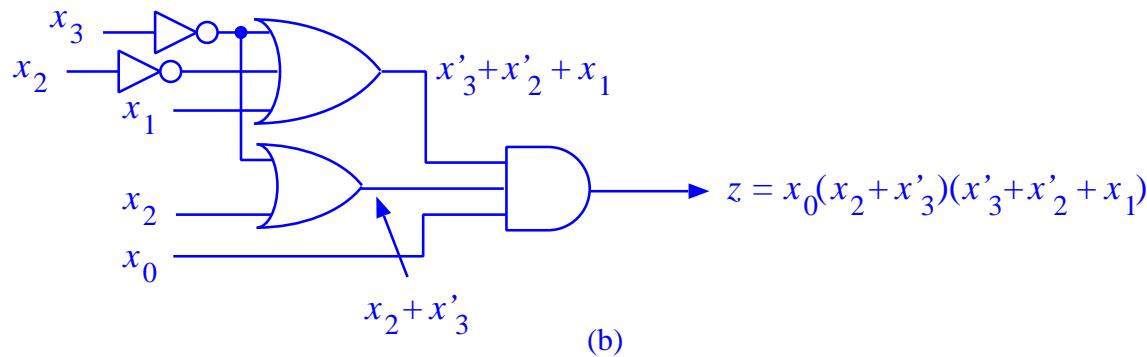
$$x_0, x_2 + x_1, x_3 + x_1 + x'_0$$

PRODUCT OF SUMS

$$(x'_2 + x_3 + x'_1)(x'_3 + x_1)x_0$$



(a)



(b)

Figure 2.13: PRODUCT OF SUMS AND OR-AND GATE NETWORK. a) SUM TERM. b) PRODUCT OF SUMS.

MAXTERM NOTATION

$$x_i \longleftrightarrow 0; \quad x'_i \longleftrightarrow 1$$

MAXTERM M_j , (sum of all n variables), j INTEGER

EXAMPLE: MAXTERM $x_3 + x'_2 + x_1 + x'_0$ DENOTED M_5
BECAUSE $0101 = 5$

$$M_j(\underline{a}) = \begin{cases} 0 & \text{if } a = j \\ 1 & \text{otherwise} \end{cases}$$

EXAMPLE: $M_5 = x_3 + x'_2 + x_1 + x'_0$
– HAS VALUE 0 ONLY FOR ASSIGNMENT 0101

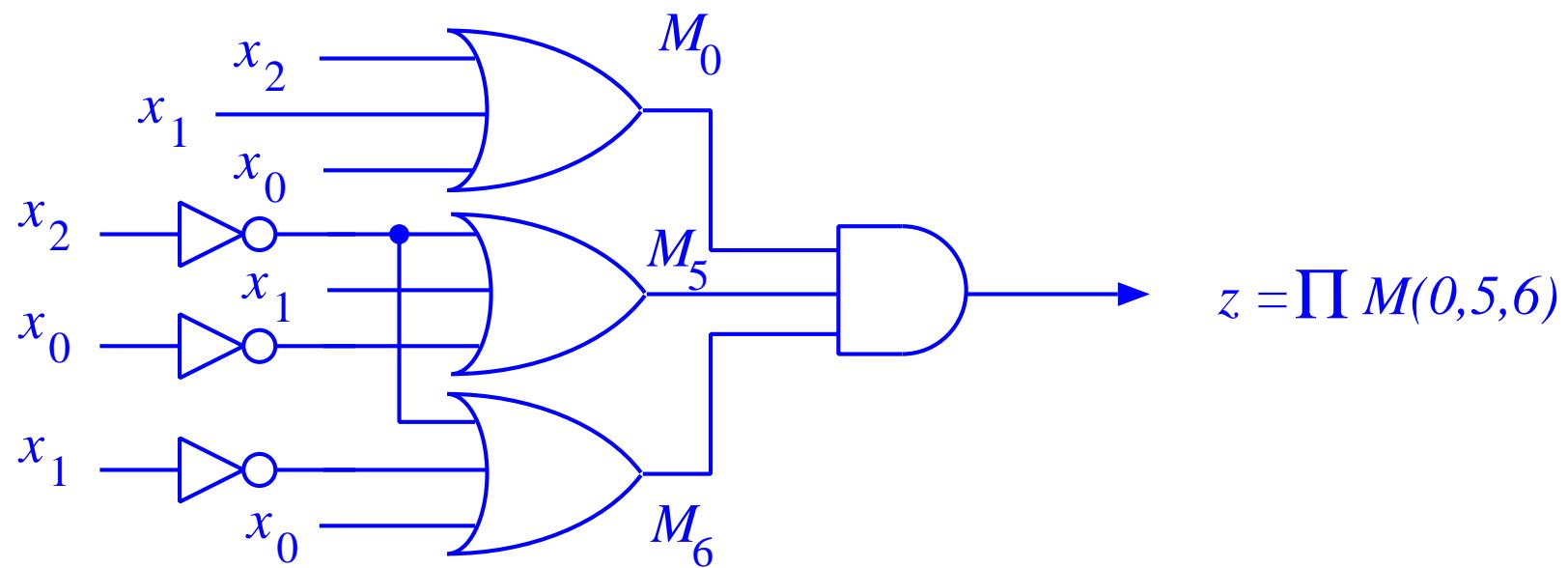


Figure 2.14: GATE NETWORK CORRESPONDING TO $E(x_2, x_1, x_0) = \prod M(0, 5, 6)$.

Example 2.15: TABLE → PRODUCT OF SUMS

j	$x_2x_1x_0$	f
0	000	0
1	001	1
2	010	1
3	011	0
4	100	0
5	101	0
6	110	1
7	111	0

$$\begin{aligned}
 E(x_2, x_1, x_0) &= \pi M(0, 3, 4, 5, 7) \\
 &= (x_2 + x_1 + x_0)(x_2 + x'_1 + x'_0)(x'_2 + x_1 + x_0) \\
 &\quad (x'_2 + x_1 + x'_0)(x'_2 + x'_1 + x'_0)
 \end{aligned}$$

CONVERSION AMONG CANONICAL FORMS

SUM OF MINTERMS \longleftrightarrow *one-set*

PRODUCT OF MAXTERMS \longleftrightarrow *zero-set*

\Rightarrow CONVERSION STRAIGHTFORWARD

$$\sum m(\{j \mid f(j) = 1\}) = \prod M(\{j \mid f(j) = 0\})$$

EXAMPLE:

m-NOTATION:

$$f(x, y, z) = \sum m(0, 4, 7)$$

M-NOTATION:

$$f(x, y, z) = \prod M(1, 2, 3, 5, 6)$$

CSMSIA

OBTAINING STANDARD EXPRESSIONS

(SOP, POS, CSP, CPS.)

(i) GETTING SOP & POS

$$E(a, b, c, d) = [(a+b')c + a'd]' \quad - \text{NOT A STANDARD FORM}$$

TRANSFORM TO SOP USING IDENTITIES

$$\begin{aligned} E(a, b, c, d) &= [(a+b')c]'. [a'd]' \\ &= [(a+b')' + c'] (a+d') \\ &= (a'b + c')(a+d') \end{aligned}$$

POS

$$\sum = ac' + a'b'd' + c'd' \leftarrow \text{SOP}$$

$$E(a, b, c, d) = (a'+c')(b+c')(a+d') \leftarrow \text{POS}$$

- USED DISTRIBUTIVITY $xy + z = (x+z)(y+z)$

(ii) GETTING CANONICAL FORMS

SOP \rightarrow CSP

$$\begin{aligned}
 ac' + a'b'd' + c'd' &= a(b+b')c'(d+d') \\
 &\quad + a'b(c+c')d' \\
 &\quad + (a+a')(b+b')c'd' \\
 &= abc'd + abc' d' + ab'c'd + ab'c'd' \\
 &\quad + a'bcd' + a'b c'd' \\
 &\quad + abc'd' + ab'c'd' + a'b c'd' + a'b'c'd' \\
 &= m_{13} + m_{12} + m_5 + m_8 \\
 &\quad + m_6 + m_4 \\
 &\quad + \cancel{m_{12}} + \cancel{m_8} + \cancel{m_4} + m_0 \\
 &= \sum m(0, 4, 6, 8, 9, 12, 13) \quad \checkmark
 \end{aligned}$$

POS \rightarrow CPS

$$\begin{aligned}
 (a' + c')(b + c')(a + d') &= (a' + b b' + c')(a a' + b + c')(a + b b' + d') \\
 &= (a' + b + c')(a' + b' + c') \\
 &\quad (a + b + c') (a' + b + c') \\
 &\quad (a + b + d') (a + b' + d') \\
 &= (a' + b + c' + d d') (a' + b' + c' + d d') \\
 &\quad (a + b + c' + d d') (a + b + c c' + d') \\
 &\quad (a + b' + a c' + d') \\
 &= (a' + b + c' + d) (a' + b + c' + d') \\
 &\quad (a' + b' + c' + d) (a' + b' + c' + d') \\
 &\quad (a + b + c' + d) (a + b + c' + d') \\
 &\quad (a + b + c + d') (a + b + c' + d') \\
 &\quad (a + b' + c + d') (a + b' + c' + d') \\
 &= M_{10} \cdot M_{11} \cdot M_{14} \cdot M_{15} \cdot M_2 \cdot \cancel{M_3} \cdot M_1 \cdot M_3 \cdot M_5 \cdot M_7 \\
 &= \prod M(1, 2, 3, 5, 7, 10, 11, 14, 15) \quad \checkmark
 \end{aligned}$$

EXAMPLE 2.19: RADIX-4 COMPARATOR

INPUTS: $x, y \in \{0, 1, 2, 3\}$

OUTPUT: $z \in \{\text{G,E,S}\}$

FUNCTION:
$$z = \begin{cases} \text{G} & \text{if } x > y \\ \text{E} & \text{if } x = y \\ \text{S} & \text{if } x < y \end{cases}$$

		y				
		0	1	2	3	
		0	E	S	S	S
		1	G	E	S	S
		2	G	G	E	S
		3	G	G	G	E
						z

Example 2.19 (cont.)

CODING:

$$x = 2x_1 + x_0 \quad \text{and} \quad y = 2y_1 + y_0$$

z	$z_2 z_1 z_0$
G	100
E	010
S	001

BINARY SPECIFICATION:

$$z_2 = \begin{cases} 1 & \text{if } x_1 > y_1 \text{ or } (x_1 = y_1 \text{ and } x_0 > y_0) \\ 0 & \text{otherwise} \end{cases}$$

$$z_1 = \begin{cases} 1 & \text{if } x_1 = y_1 \text{ and } x_0 = y_0 \\ 0 & \text{otherwise} \end{cases}$$

$$z_0 = \begin{cases} 1 & \text{if } x_1 < y_1 \text{ or } (x_1 = y_1 \text{ and } x_0 < y_0) \\ 0 & \text{otherwise} \end{cases}$$

		y_1y_0			
		00	01	10	11
x_1x_0	00	010	001	001	001
	01	100	010	001	001
10	100	100	010	001	
11	100	100	100	010	

$z_2 z_1 z_0$

Example 2.19 (cont)

SWITCHING EXPRESSIONS:

$$z_2(x_1, x_0, y_1, y_0) = \sum m(4, 8, 9, 12, 13, 14)$$

$$z_1(x_1, x_0, y_1, y_0) = \sum m(0, 5, 10, 15)$$

$$z_0(x_1, x_0, y_1, y_0) = \sum m(1, 2, 3, 6, 7, 11)$$

EXAMPLES OF BINARY
SPECIFICATIONS

1. FULL ADDER (FA) (ADDEND, AUGEND)

INPUTS: $x_i, \underline{y_i} \in \{0, 1\}$ $c_i \in \{0, 1\}$ (CARRY-IN)

OUTPUTS: $s_i \in \{0, 1\}$ SUM $c_{i+1} \in \{0, 1\}$ (CARRY-OUT)

FUNCTION:

$$s_i = (x_i + y_i + c_i) \bmod 2 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{SOLUTIONS}$$

$$c_{i+1} = \begin{cases} 1 & \text{IF } x_i + y_i + c_i \geq 2 \\ 0 & \text{OTHERWISE} \end{cases} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{OF}$$

SWITCHING FUNCTIONS

$$s_i = f_0(x_i, y_i, c_i) \quad c_{i+1} = f_1(x_i, y_i, c_i)$$

$$\begin{aligned} x_i + y_i + c_i \\ = 2c_{i+1} + s_i \end{aligned}$$

j	Σ	x_i	y_i	c_i	c_{i+1}	s_i
0	0	0	0	0	0	0
1	1	0	0	1	0	1
2	1	0	1	0	0	1
3	2	0	1	1	1	0
4	1	1	0	0	0	1
5	2	1	0	1	1	0
6	2	1	1	0	1	0
7	3	1	1	1	1	1

$$s_i = \sum m(1, 2, 4, 7)$$

$$c_{i+1} = \sum m(3, 5, 6, 7)$$

2. CHARACTER MATCHING

INPUTS $x, y \in \{A, B, \dots, Z, a, b, \dots, z, 0, 1, \dots, 9, +, /, \dots\}$
 OUTPUT $z \in \{0, 1\}$

FUNCTION $z = \begin{cases} 1 & \text{IF } x = y \\ 0 & \text{IF } x \neq y \end{cases}$

$$x \rightarrow \underline{x} = (x_7, x_6, \dots, x_0)$$

$$\underline{y} = (y_7, y_6, \dots, y_0)$$

16 VARIABLES \rightarrow TABLE NOT PRACTICAL

OBSERVE : $x = y \text{ IFF } x_i = y_i, \forall i$

AN SE FOLLOWS DIRECTLY

$$z = (x_7y_7 + x'_7y'_7)(x_6y_6 + x'_6y'_6) \dots (x_0y_0 + x'_0y'_0)$$



3. 2-BY-2 MULTIPLIER (2-BIT x 2-BIT)

(i) HL SPEC

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	4	6
3	0	3	6	9
x				p

INPUTS $x, y \in \{0, 1, 2, 3\}$ MULTIPLICAND x
 OUTPUT/FUNCTION $P = x \times y \in \{0, 1, 2, 3, 4, 6, 9\}$ MULTIPLIER y
 PRODUCT

(ii) BL SPEC - ASSUME BINARY CODE

$$x = (x_1, x_0) = 2x_1 + x_0 \quad x_i, y_i \in \{0, 1\}$$

$$y = (y_1, y_0) = 2y_1 + y_0$$

$$P = (P_3, P_2, P_1, P_0) = \sum_{i=0}^3 p_i 2^i \quad p_i \in \{0, 1\}$$

	00	01	10	11
00	0000	0000	0000	0000
01	0000	0001	0010	0011
10	0000	0010	0100	0110
11	0000	0011	0110	1001
x, x_0				P_3, P_2, P_1, P_0

	00	01	10	11
00	m_0	m_1	m_2	m_3
01	m_4	m_5	m_6	m_7
10	m_8	m_9	m_{10}	m_{11}
11	m_{12}	m_{13}	m_{14}	m_{15}

$$m_i(x_1, x_0, y_1, y_0)$$

$$P_3 = m_{15}$$

$$P_2 = \sum m(10, 11, 14)$$

$$P_1 = \sum m(6, 7, 9, 11, 13, 14)$$

$$P_0 = \sum m(5, 7, 13, 15) = x_0 y_0$$

DETERMINE THE COST

NOT # AND # OR

2 by 2 multiplier

direct implementation of SOP expressions

