

Consensus Theorem (Examples)

Which one is not a consensus of the expressions on the right.

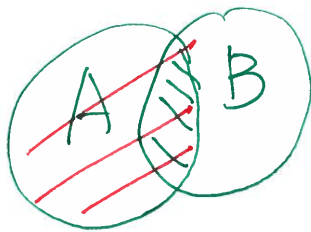
A. B ——— 1. $A + A'B = A + B$

B. BC ——— 2. $A + A'BC = A + BC$

C. $(B+C)D$ ——— 3. $A(B+C) + A'D$

D. BCD ——— 4. $ABC + A'D + BCDE = ABC + A'D$

E. BCDE ——— 4. $ABC + A'D + BCDE = ABC + A'D$



$$A + \underline{AB} = A(1 + B) = A \cdot 1 = A$$

$$A(A+B) =$$

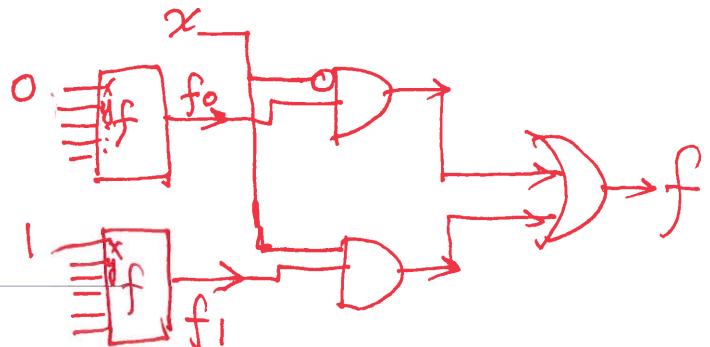
$$A \cdot A + AB$$

$$= A + AB$$

$$= A$$

Shannon's Expansion

- Shannon's expansion assumes a switching algebra system
- Divide a switching function into smaller functions
- Pick a variable x , partition the switching function into two cases: $x=1$ and $x=0$
 - $f(x,y,z,\dots) = xf(x=1,y,z,\dots) + x'f(x=0,y,z,\dots)$
- For example
 - $f(x) = xf(1) + x'f(0)$
 - $f(x,y) = xf(1,y) + x'f(0,y)$



Shannon's Expansion (Example)

$$f(a, b, c) = \underset{1}{a'b'} + \underset{2}{bc} + \underset{3}{ab'c}$$

$$\begin{aligned} 1+2 &\rightarrow a'c \\ 1+3 &\rightarrow b'c \\ 2+3 &\rightarrow ac \end{aligned}$$

Shannon's expansion:

$$f(a, b, c) = af(1, b, c) + a'f(0, b, c)$$

$$f(1, b, c) = \cancel{1'b'} + bc + 1b'c = bc + b'c + c = c$$

$$f(0, b, c) = 0'b' + bc + \cancel{0b'c} = b' + bc + c = b' + c$$

$$f(a, b, c) = af(1, b, c) + a'f(0, b, c)$$

$$= a \cdot c + a'(b' + c)$$

$$= \underline{ac} + a'b' + \underline{a'c} + c$$

$$= a'b' + c$$

1-<31>

Shannon's Expansion: Example

$$f(x, y, z) = xf(x = 1, y, z) + x'f(x = 0, y, z)$$

$$f(x, y) = (x + f(x = \textcircled{?}, y))(x' + f(x = \textcircled{?}', y))$$

• A. ?=0 $f(a, b, c) = (a' + b')(b + c)(a + b' + c)$

• B. ?=1. $f(a, b, c) = (a + f(a = 0, b, c))(a' + f(a = 1, b, c))$

$$f(a = 0, b, c) = (\cancel{0'b'}) + (b + c)(0 + b' + c) = (b + c)(b' + c) = \textcircled{C}$$

$$f(a = 1, b, c) = (1' + b')(b + c)(1 + \cancel{b' + c}) = b'(b + c) = b'b + b'c = \textcircled{b'c}$$

$$f(a, b, c) = (\underline{a + c})(\underline{a' + b'c}) = (\underline{a + c})(a' + b')(a' + c)c = \underline{a' + b' + c}$$

<32>

Shannon's Expansion

- Decompose the switching function into minterms

$$f(x, y) = xf(1, y) + x'f(0, y)$$

$$= x(yf(1,1) + y'f(1,0)) + x'(yf(0,1) + y'f(0,0))$$

$$= xyf(1,1) + xy'f(1,0) + x'yf(0,1) + x'y'f(0,0).$$

- Decompose the switching function into maxterms

$$f(x, y) = (x' + f(1, y)) \cdot (x + f(0, y))$$

$$= (x' + (y' + f(1,1)) \cdot (y + f(1,0))) \cdot (x + (y' + f(0,1)) \cdot (y + f(0,0)))$$

$$= (x' + y' + f(1,1))(x' + y + f(1,0))(x + y' + f(0,1))(x + y + f(0,0))$$

<35>

Shannon's Expansion: Example

Which variable in $ab' + ac + bc$ can be used for expansion?

A. a

B. b

C. c

D. None of the above

<36>

Shannon's Expansion: Example

$$F(A,B,C)=AB'+AC+BC$$

<37>

Remark: The choice of the variable for expansion is a nontrivial question.

<38>

Shannon's Expansion

Boole-Shannon

- Decompose the switching function into minterms

$$\begin{aligned}
 f(x, y) &= x f(1, y) + x' f(0, y) \\
 &= x (y f(1, 1) + y' f(1, 0)) + x' (y f(0, 1) + y' f(0, 0)) \\
 &= x y f(1, 1) + x y' f(1, 0) + x' y f(0, 1) + x' y' f(0, 0).
 \end{aligned}$$

id	x	y	f(x,y)	
0	0	0	f(0,0)	$x'y'$
1	0	1	f(0,1)	$x'y$
2	1	0	f(1,0)	xy'
3	1	1	f(1,1)	xy

Shannon's expansion can decompose a switching function into a truth table.

<33>

Shannon's Expansion vs Truth Table

Example: $f(x,y) = x + x'y$

$$\begin{aligned}
 f(x, y) &= x f(1, y) + x' f(0, y) \\
 &= x (y f(1, 1) + y' f(1, 0)) + x' (y f(0, 1) + y' f(0, 0)) \\
 &= x y f(1, 1) + x y' f(1, 0) + x' y f(0, 1) + x' y' f(0, 0).
 \end{aligned}$$

id	x	y	f(x,y)
0	0	0	f(0,0) = 0
1	0	1	f(0,1) = 1
2	1	0	f(1,0) = 1
3	1	1	f(1,1) = 1

Shannon's expansion can decompose a switching function into a truth table.

<34>

Review Summary: Switching Algebra and Karnaugh Map

Shannon's expansion and consensus theorem are used for logic optimization

- Shannon's expansion **divides** the problem into smaller functions
- Consensus theorem finds common terms when we **merge** small functions
- Karnaugh map mimics the above two operations in two dimensional space as a visual aid.

<39>

Part I. Combinational Logic

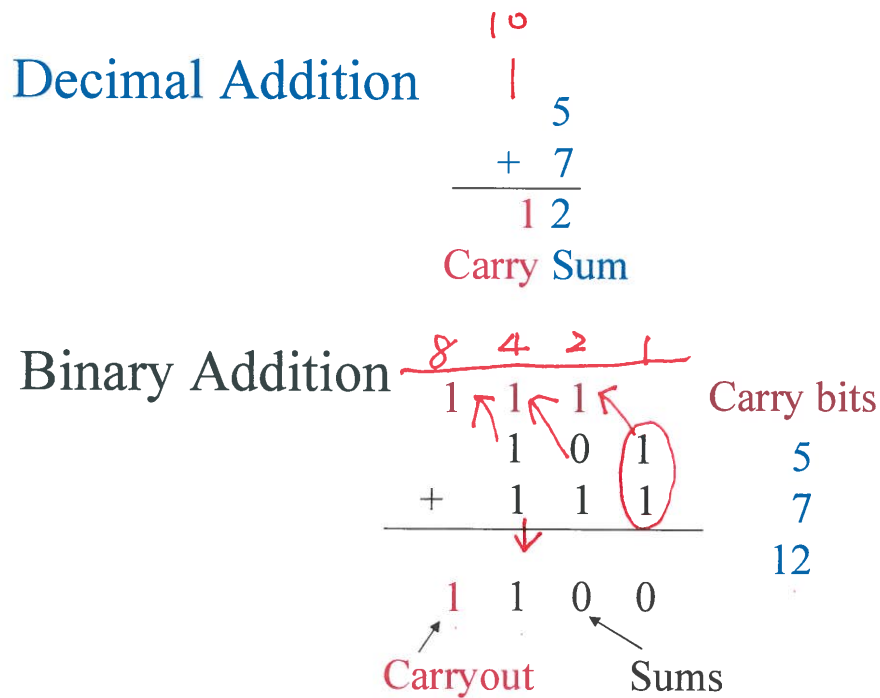
II) Specification

1. Language
2. Boolean Algebra

Canonical Expression: Sum of minterms and
Product of maxterms

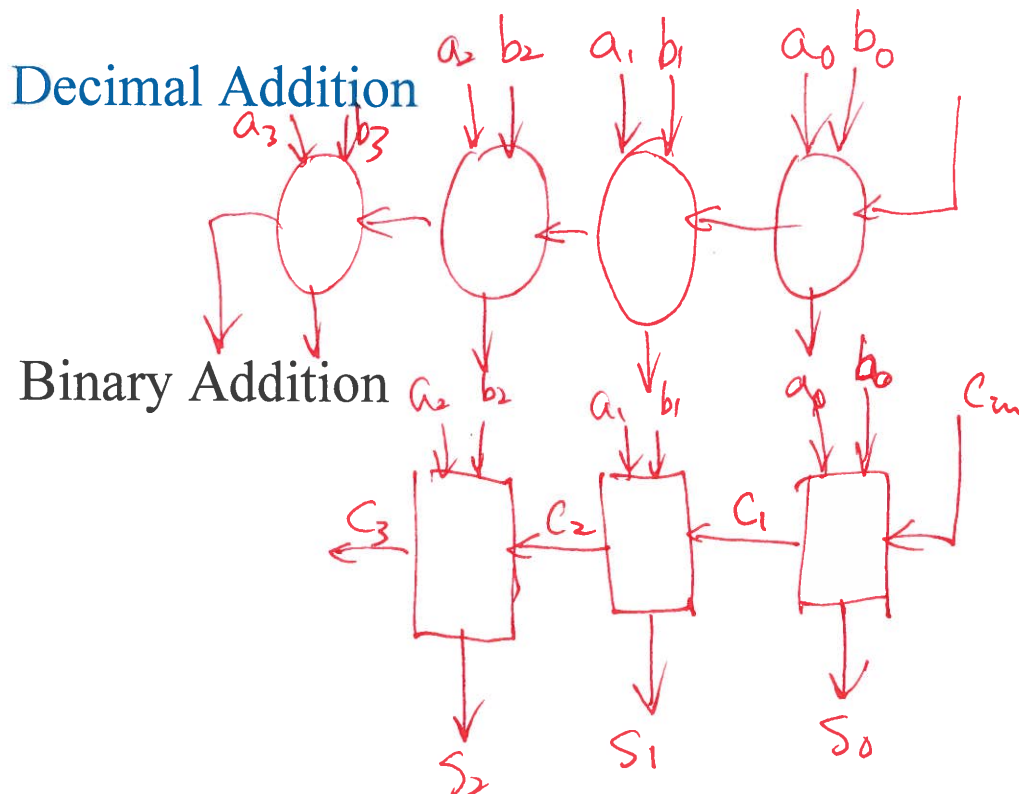
3. Truth Table: **minterms and maxterms**
4. Incompletely Specified Function

II. Specification (use addition as an example)



41

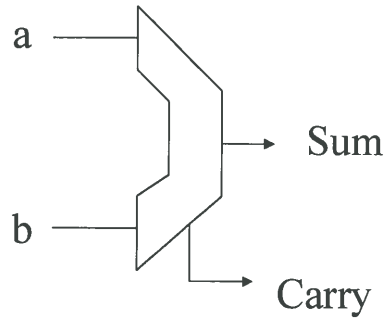
II. Specification (use addition as an example)



42

Half Adder

Truth Table



id	a	b	carry	sum
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

45

Switching Function

Switching Expressions:

$$\text{Sum (a,b)} = a' \cdot b + a \cdot b'$$

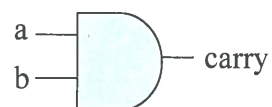
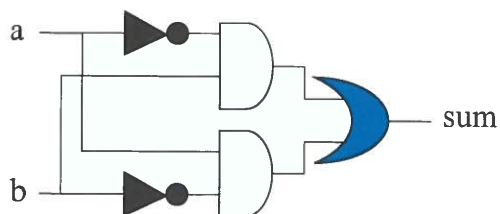
$$\text{Carry (a, b)} = a \cdot b$$

Ex:

$$\text{Sum (0,0)} = 0' \cdot 0 + 0 \cdot 0' = 0 + 0 = 0$$

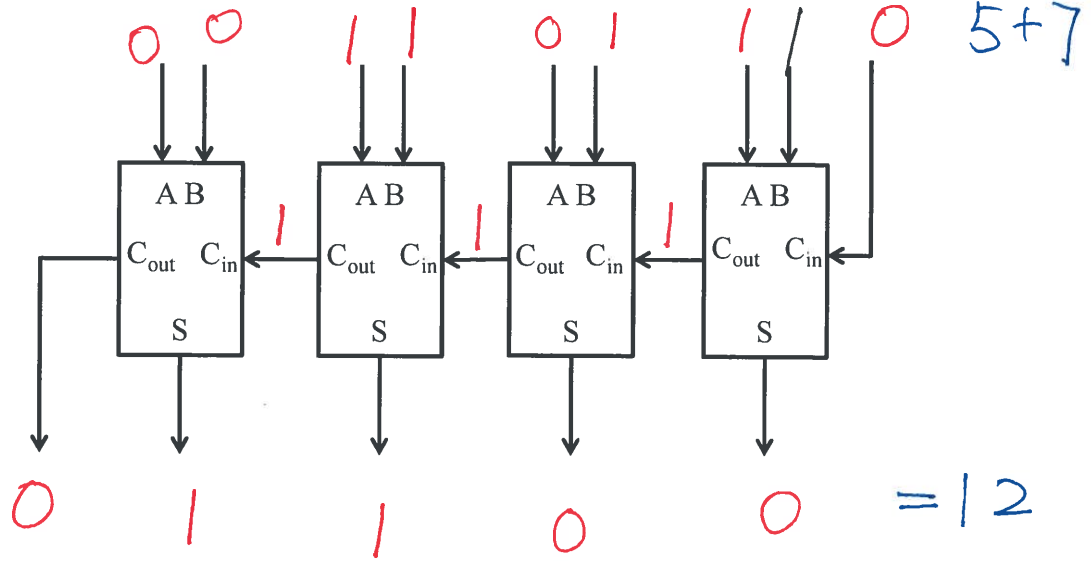
$$\text{Sum (0,1)} = 0' \cdot 1 + 0 \cdot 1' = 1 + 0 = 1$$

$$\text{Sum (1,1)} = 1' \cdot 1 + 1 \cdot 1' = 0 + 0 = 0$$



46

II. Specification (use addition as an example)



43

Binary Addition: Hardware

- Half Adder: Two inputs (a,b) and two outputs (carry, sum).
- Full Adder: Three inputs (a,b,c_{in}) and two outputs (carry, sum).

44

BSV notes

```

function Bit#(2) ha(Bit#(1) a, Bit#(1) b);
  Bit#(1) s = (~a & b) | (a & ~b);
  Bit#(1) c = a & b;
  return {c,s};
endfunction

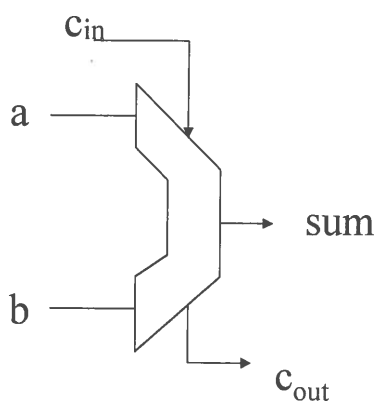
```

- {c, s} represents bit concatenation

CSE 140L W2017

L01-47

Full Adder



id	a	b	c _{in}	c _{out}	sum
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Arithmetic:

$$2c_{out} + sum = a + b + c_{in}$$

R3 $2 \times 1 + 0 = 0 + 1 + 1 = 2$

R7 $2 \times 1 + 1 = 1 + 1 + 1 = 3$

Minterms

A product of all variables in the function.

A minterm is equal to 1 on exactly one row of the truth table.

id	a	b	c	carry	sum
0	0	0	0	0	0
1	0	0	1	0	$a'b'c$
2	0	1	0	0	$a'bc'$
3	0	1	1	$a'bc$	0
4	1	0	0	0	$ab'c'$
5	1	0	1	$ab'c$	0
6	1	1	0	abc'	0
7	1	1	1	abc	abc

49

$$\text{carry} = \underset{m_3}{a'bc} + \underset{m_5}{ab'c} + \underset{m_6}{abc'} + \underset{m_7}{abc} = m_3 + m_5 + m_6 + m_7 = \sum m(3, 5, 6, 7)$$

Maxterms

$$\text{Sum} = \sum m(1, 2, 4, 7)$$

A sum of all variables in the function.

A maxterm is equal to 0 on exactly one row of the truth table.

id	a	b	c	carry	sum
0	0	0	0	$a+b+c$	$a+b+c$
1	0	0	1	$a+b+c'$	1
2	0	1	0	$a+b'+c$	1
3	0	1	1	1	$a+b'+c'$
4	1	0	0	$a'+b+c$	1
5	1	0	1	1	$a'+b+c'$
6	1	1	0	1	$a'+b'+c$
7	1	1	1	1	1

$$\text{Carry} = \Pi M(0, 1, 2, 4)$$

$$\text{Sum} = \Pi M(0, 3, 5, 6)$$

$$a'+b'+c = 0 \text{ iff } (a, b, c) = (1, 1, 0)$$

50

Minterms and Maxterms

id	a	b	c	minterm	maxterm
0	0	0	0	$m_0 = a'b'c'$	$M_0 = a+b+c$
1	0	0	1	$m_1 = a'b'c$	$M_1 = a+b+c'$
2	0	1	0	$m_2 = a'bc'$	$M_2 = a+b'+c$
3	0	1	1	$m_3 = a'bc$	$M_3 = a+b'+c'$
4	1	0	0	$m_4 = ab'c'$	$M_4 = a'+b+c$
5	1	0	1	$m_5 = ab'c$	$M_5 = a'+b+c'$
6	1	1	0	$m_6 = abc'$	$M_6 = a'+b'+c$
7	1	1	1	$m_7 = abc$	$M_7 = a'+b'+c'$

Minterms cover all the outputs which are true (1).

Maxterms cover all the outputs which are false (0).

51

Minterms

$$f_1(a,b,c) = a'bc + ab'c + abc' + abc$$

$$a'bc = 1 \text{ iff } (a,b,c) = (0,1,1)$$

$$ab'c = 1 \text{ iff } (a,b,c) = (1,0,1)$$

$$abc' = 1 \text{ iff } (a,b,c) = (1,1,0)$$

$$abc = 1 \text{ iff } (a,b,c) = (1,1,1)$$

$$f_1(a,b,c) = 1 \text{ iff } (a,b,c) = (0,1,1), (1,0,1), (1,1,0), \text{ or } (1,1,1)$$

$$\text{Ex: } f_1(1,0,1) = 1'01 + 10'1 + 101' + 101 = 1$$

$$f_1(1,0,0) = 1'00 + 10'0 + 100' + 100 = 0$$

52

Maxterms

$$f_2(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b+c)$$

$$a + b + c = 0 \text{ iff } (a,b,c) = (0,0,0)$$

$$a + b + c' = 0 \text{ iff } (a,b,c) = (0,0,1)$$

$$a + b' + c = 0 \text{ iff } (a,b,c) = (0,1,0)$$

$$a' + b + c = 0 \text{ iff } (a,b,c) = (1,0,0)$$

$$f_2(a,b,c) = 0 \text{ iff } (a,b,c) = (0,0,0), (0,0,1), (0,1,0), (1,0,0)$$

$$\begin{aligned} \text{Ex: } f_2(1,0,1) &= (1+0+1)(1+0+1')(1+0'+1)(1'+0+1) = 1 \\ f_2(0,1,0) &= (0+1+0)(0+1+0')(0+1'+0)(0'+1+0) = 0 \end{aligned}$$

53

$$f_1(a,b,c) = a'bc + ab'c + abc' + abc$$

$$f_2(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b+c)$$

$$f_1(a, b, c) = m_3 + m_5 + m_6 + m_7 = \Sigma m(3,5,6,7)$$

$$f_2(a, b, c) = M_0M_1M_2M_4 = \Pi M(0, 1, 2, 4)$$

Does $f_1 = f_2$?

A. Yes

B. No.

The coverage of a single minterm. e.g. $m_4 = ab'c'$

Id	a	b	c_{in}	carry	minterm $_4 = ab'c'$
0	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	0	0
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

Only one row has a 1.

55

The coverage of a single maxterm. E.g. $M_4 = a'+b+c$

Id	a	b	c_{in}	carry	maxterm $_4 = a+b+c$
0	0	0	0	0	1
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	1
6	1	1	0	1	1
7	1	1	1	1	1

Only one row has a 0.

56

Minterms and Maxterms: Summary

$$f_1(a,b,c) = a'bc + ab'c + abc' + abc$$

$$f_2(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b+c)$$

Canonical presentation of logic functions

Conversion between truth tables and switching functions

57

Incompletely Specified Function

Don't care set is important because it allows us to minimize the function

Id	a	b	f(a, b)
0	0	0	1 $a'b'$
1	0	1	0
2	1	0	1 ab'
3	1	1	- 0

$$f(a,b) = b' \quad f(a,b) = a + b'$$

1) The input does not happen.

2) The input happens, but the output is ignored.

Spec: maximize don't cares

Design: Utilize don't cares to reduce the logic.

Examples:

-Decimal number 0... 9 uses 4 bits. (1,1,1,1) does not happen.

-Final carry out bit (output is ignored).

58



Incomplete Specification - Inputs

- Seven Segment Display Driver – CD4543B
- D, C, B, A only matter when LD, BI', PH'

CD4543B Types

TRUTH TABLE FOR CD4543B

INPUT CODE							OUTPUT STATE							DISPLAY CHARACTER
LD	BI	Ph*	D	C	B	A	a	b	c	d	e	f	g	
X	1	0	X	X	X	X	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0	1	1	0	0	0	0	1
1	0	0	0	0	1	0	1	1	0	1	1	0	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	0	0	0	1	1	0	0	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0	0	0	0
1	0	0	1	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	0	0	1	1	1	1	1	0	1	1	1
1	0	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	X	X	X	X	**							**
†	†	1	†				Inverse of Output Combinations Above							Display as above

X=Don't care.

†=Above combinations.

*=For liquid-crystal readouts, apply a square wave to Ph.

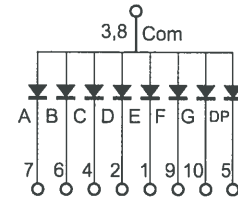
For common cathode LED readouts, select Ph=0.

For common anode LED readouts, select Ph=1.

**=Depends upon the BCD code previously applied when LD=1.



7 Segment Display Pin Out Diagram



www.circuitstoday.com

<http://www.ti.com/lit/ds/symlink/cd4543b.pdf>
<http://www.circuitstoday.com/7-segment-display-to-avr-atmega32-using-cd4511b>

59

Incompletely Specified Function

id	a	b	c	g
0	0	0	0	0
1	0	0	1	1
2	0	1	0	-
3	0	1	1	1
4	1	0	0	1
5	1	0	1	-
6	1	1	0	0
7	1	1	1	1

$$\begin{aligned}
 g_1(a,b,c) &= a'b'c + a'bc + ab'c' + abc \\
 &= m_1 + m_3 + m_4 + m_7 \\
 &= \sum m(1,3,4,7)
 \end{aligned}$$

$$\begin{aligned}
 g_2(a,b,c) &= (a+b+c)(a'+b'+c) \\
 &= M_0 M_6 \\
 &= \prod M(0,6)
 \end{aligned}$$

60

Incompletely Specified Function

id	a	b	c	g
0	0	0	0	0
1	0	0	1	1
2	0	1	0	-
3	0	1	1	1
4	1	0	0	1
5	1	0	1	-
6	1	1	0	0
7	1	1	1	1

$$g_1(a,b,c) = \sum m(1,3,4,7)$$

$$g_2(a,b,c) = \prod M(0,6)$$

Does $g_1(a,b,c) = g_2(a,b,c)$?

A: Yes

B: No

CSE 140: Components and Design Techniques for
Digital Systems

Lecture 3: Incompletely Specified
Functions and K Maps

CK Cheng

Dept. of Computer Science and Engineering
University of California, San Diego

1

Outlines

- Definitions
 - Minterms and Maxterms
 - Incompletely Specified Function
- Implementation: Boolean Algebra vs Map
- Karnaugh Maps: Two Dimensional Truth Table
 - 2-Variable Map
 - 3-Variable Map
 - Up to 6-Variable Map

2

Definitions

- Literals x_i or x'_i
- Product Term $x_2x'_1x_0$
- Sum Term $x_2 + x'_1 + x_0$
- Minterm of n variables: A product of n literals in which every variable appears **exactly** once.
- Maxterm of n variables: A sum of n literals in which every variable appears **exactly** once.
- Adjacency of minterms (maxterms): Two minterms (maxterms) are adjacent if they differ by only one variable.

$a+b+c$ & $a'+b+c$ adjacent
 $a+b+c$ & $a'+b'+c$ not adjacent.

3

Definitions: Examples

- Minterm of n variables: A product of n literals in which every variable appears **exactly** once.
 - E.g. For function $f(a, b, c, d, e)$, $abcde$, $a'b'c'de$ are minterms, while bcd , $a'bcd$ are not.
- Maxterm of n variables: A sum of n literals in which every variable appears **exactly** once.
- Adjacency of minterms: Two minterms are adjacent if they differ by only one variable.
 - E.g. $abcde$ and $a'bcd$ are adjacent, while $a'b'cde$ and $abc'd'e'$ are not

4

Examples

For two minterms $ab'c'd'$ and $ab'cd'$, are they adjacent?

- A. Yes
- B. No

Adjacency allows us to merge the terms to reduce the Boolean expression.

5

Incompletely Specified Function

- Situations where the output of a switching function can be either 0 or 1 for a particular combination of inputs
- This is specified by a don't care in the truth table

Id	a	b	f(a, b)
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	X (don't care)

For example:

- 1) The input pattern does not happen.
- 2) The input pattern happens, but the output is ignored.

Example:

-Decimal number 0... 9 uses 4 bits. (1,1,1,1) does not happen.

6

Incompletely Specified Function

How to completely specify the truth table in canonical form?

We have three types of output which divides the input space into three sets:

On-set $\mathcal{F}$: All the input conditions for which the output is 1

Off-set $\mathcal{R}$: All the input conditions for which the output is 0

Don't care set $\mathcal{D}$: All the input conditions for which the output is a 'don't care'

Example: The truth table on right has

$\mathcal{F}$ covers input pattern $\{(a=1, b=0)\}$

$\mathcal{R}$ covers input pattern $\{(0,0)\}$

$\mathcal{D}$ covers input patterns $\{(0,1), (1,1)\}$

The union of $\mathcal{F}$, $\mathcal{R}$, $\mathcal{D}$ is the whole set of all input patterns.

Id	a	b	F
0	0	0	0
1	0	1	X
2	1	0	1
3	1	1	X

7

Reducing Incompletely Specified Functions

Which of the following assignment would result in an implementation with the fewest gates?

Id	a	b	F(a,b)
0	0	0	X
1	0	1	0
2	1	0	1
3	1	1	0

A. $F(0,0)=1$

B. $F(0,0)=0$

C. Neither A or B

D. Both A and B

8

Incompletely Specified Function

Example: The truth table on right has
 F covers input pattern $(a,b)=(1,0)$
 $\mathcal{R}$ covers input pattern $(a,b)=(0,0)$
 $\mathcal{D}$ covers input patterns $(a,b)=(0,1) (1,1)$

We assign values to elements in
don't care set for logic designs.

Id	a	b	F
0	0	0	0
1	0	1	X
2	1	0	1
3	1	1	X

Id	a	b	F
0	0	0	0
1	0	1	
2	1	0	1
3	1	1	

9

Implementation

Specification → Schematic Diagram
Net list,
Switching expression
Obj min cost → Search in solution space
(max performance)
Cost: wires, gates → Literals, product terms,
sum terms

For two level logic (sum of products or product of sums),
we want to minimize # of terms, and # of literals

Implementation: Specification $\Rightarrow$ Logic Diagram

Karnaugh Map: A 2-dimensional truth table

Flow 1: Boolean Algebra

1. Specification
2. Truth table
3. Sum of products (SOP) or product of sums (POS) canonical form
4. Reduced expression using Boolean algebra
5. Schematic diagram of two level logic

Flow 2: K Map

1. Specification
2. Truth Table
3. Karnaugh Map (truth table in two dimensional space)
4. Reduce using K'Maps
5. Reduced expression (SOP or POS)
6. Schematic diagram of two level logic

11

Truth Table vs. Karnaugh Map

2-variable function, $f(A,B)$

ID	A	B	$f(A,B)$
0	0	0	$f(0,0)$ $a'b'$
1	0	1	$f(0,1)$ $a'b$
2	1	0	$f(1,0)$ ab'
3	1	1	$f(1,1)$ ab

	B=0	B=1
A=0	$f(0,0)$ $a'b'$	$f(0,1)$ $a'b$
A=1	$f(1,0)$ ab'	$f(1,1)$ ab

12

Truth Table

An example of 2-variable function, $f(A,B)$

ID	A	B	$f(A,B)$	minterm
0	0	0	0	
1	0	1	1	$A'B$
2	1	0	1	AB'
3	1	1	1	AB

13

Function can be represented by sum of minterms:

$$f(A,B) = A'B + AB' + AB$$

This is not optimal however!

We want to minimize the number of literals and terms.

14

To minimize the number of literals and terms.
We factor out common terms –

$$\begin{aligned} &A'B + AB' + AB \\ &= A'B + AB' + \mathbf{AB} + \mathbf{AB} \\ &= (A' + \mathbf{A})B + A(B' + \mathbf{B}) = B + A \end{aligned}$$

Hence, we have

$$f(A,B) = A + B$$

	$b=0$	$b=1$
$a=0$	$a'b'$	$a'b$
$a=1$	ab'	ab

f	a	b
0	0	0
1	0	1
2	1	0
3	1	1

$$\Sigma m(2,3) = a$$

$$\Sigma m(1,3) = b$$

$$f = a + b.$$

15

How can we guarantee the most reduced expression was reached?

- Boolean expressions can be minimized by combining terms
- K-maps minimize equations graphically

ID	A	B	$f(A,B)$
0	0	0	$f(0,0)$
1	0	1	$f(0,1)$
2	1	0	$f(1,0)$
3	1	1	$f(1,1)$

	B=0	B=1
A=0	$A'B'$	$A'B$
A=1	AB'	AB