

Shannon's Expansion vs Truth Table

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

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

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

Consensus Theorem

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

WZB
<33>

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

<35>

Shannon's Expansion

- Decompose the switching function into minterms

$$\begin{aligned} f(x,y) &= xf(1,y) + x'f(0,y) \\ &= x(yf(1,1) + y'f(1,0)) + x'(yf(0,1) + y'f(0,0)) \\ &= xyf(1,1) + xy'f(1,0) + x'yf(0,1) + x'y'f(0,0). \end{aligned}$$
- Decompose the switching function into maxterms

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

x	y	
0	0	$M_0 \quad x+y$
0	1	$M_1 \quad x+y'$
1	0	$M_2 \quad x'+y$
1	1	$M_3 \quad x'+y'$

<34>

Shannon's Expansion: Example

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

B $F(A, \text{true}, C) \rightarrow A + C$

B' $F(A, \text{false}, C) \rightarrow A + AC$

<36>

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

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

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

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

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

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

$$\begin{array}{c} \textcircled{abc} + \textcircled{a'cd} + \textcircled{bcd} \\ a \quad \textcircled{bc} \quad \cap \quad \textcircled{bcd} \\ a' \quad \textcircled{cd} \end{array}$$

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

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.

II. Specification (use addition as an example)

Decimal Addition

$$\begin{array}{r} 1009 \\ + 5 \\ + 7 \\ \hline 112 \end{array}$$

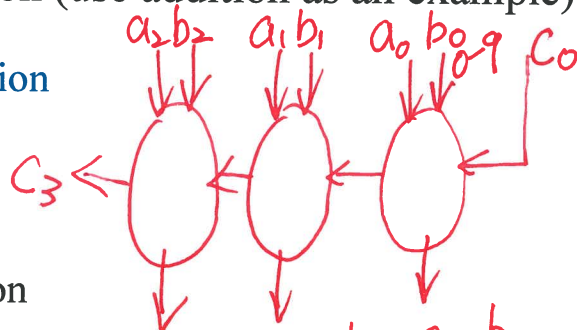
Binary Addition

	1	1	1	1	
	1	1	0	1	5
+	1	1	1	1	7
	1	1	0	0	12
Carryout	1	1	0	0	

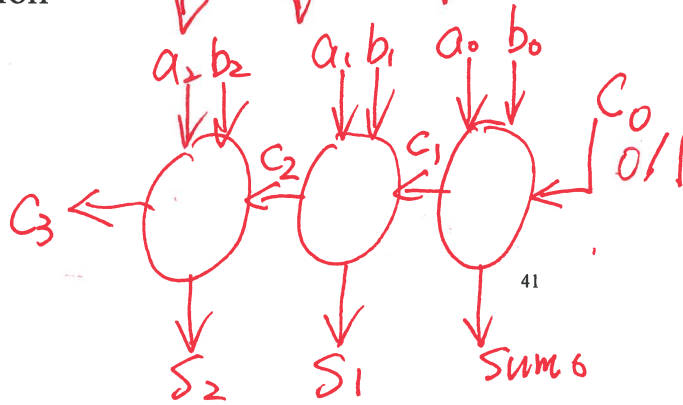
Carry bits

II. Specification (use addition as an example)

Decimal Addition



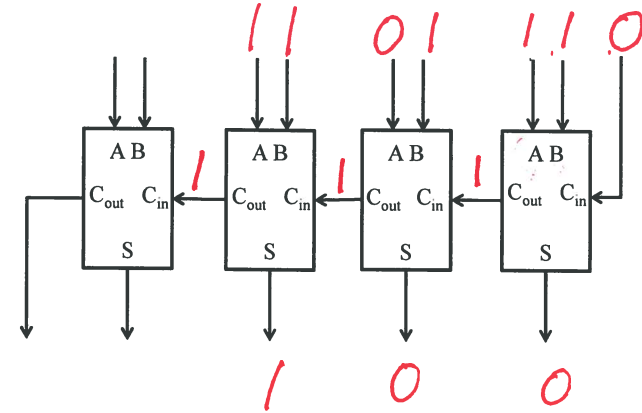
Binary Addition



Binary Addition: Hardware

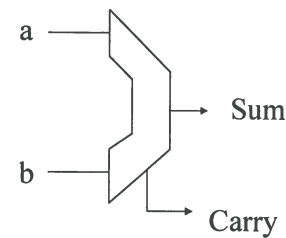
- Half Adder: Two inputs (a,b) and two outputs (carry, sum).
- Full Adder: Three inputs (a,b, c_{in}) and two outputs (carry, sum).

II. Specification (use addition as an example)



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

Switching Function

Switching Expressions:

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

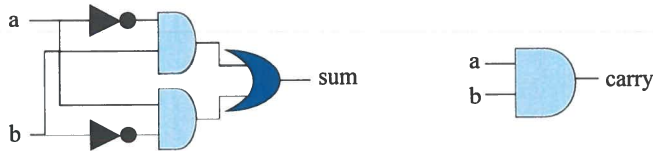
$$\text{Carry}(a,b) = a \cdot b$$

Ex:

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

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

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



45

BSV notes

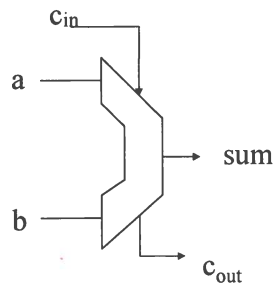
```
function Bit#(2) ha(Bit#(1) a, Bit#(1) b);
  Bit#(1) s = (~a & b) | (a & ~b);
  Bit#(1) c = a & b;
  return {c,s};
endfunction
```

- {c, s} represents bit concatenation

CSE 140L W2017

L01-46

Full Adder



Arithmetic:

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

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

47

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

48

$$\text{Carry} = a'bc + ab'c + abc' + abc$$

$\begin{matrix} 1'01 & 101 & 101' & 101 \end{matrix}$

Maxterms

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

49

$$\text{Carry}(a,b,c) = (a+b+c)(a+b+c')(a+b'+c)(a'+b+c)$$

$$\text{Minterms } 010 : \underline{010} \quad \underline{010'} \quad \underline{01'0} \quad \underline{0'10}$$

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

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

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

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

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

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

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

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

51

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).

50

Maxterms

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

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

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

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

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

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

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

$$f_2(0,1,0) = (0+1+0)(0+1+0')(0+1'+0)(0'+1+0) = 0$$

52

$$f_1(a,b,c)' = (a+b'+c')(a'+b+c')(a'+b'+c)(a'+b'+d)$$

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

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

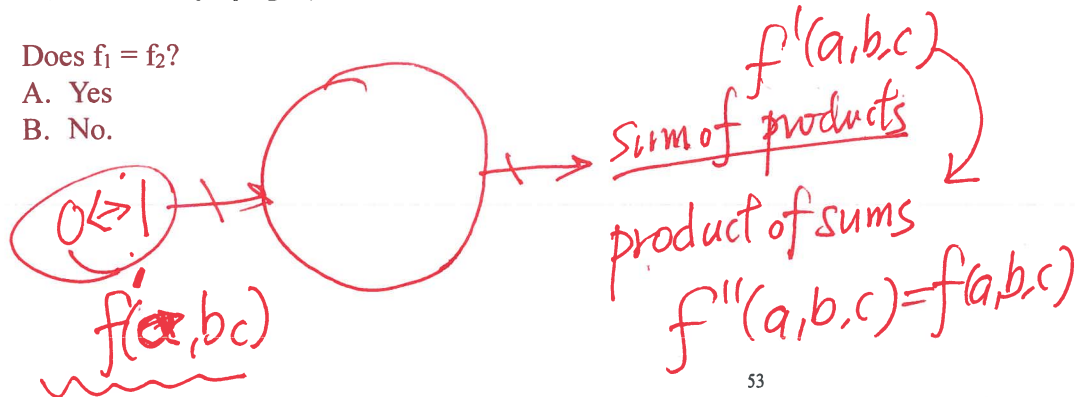
$$f_1(a,b,c) = m_3 + m_5 + m_6 + m_7 = \Sigma m(3,5,6,7)$$

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

Does $f_1 = f_2$?

A. Yes

B. No.



53

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.

54

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.

55

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

56

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
1	0	1	0
2	1	0	1
3	1	1	-

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

Examples:

- Decimal number 0... 9 uses 4 bits. (1,1,1,1) does not happen.
- Final carry out bit (output is ignored).

Designer has the option to decide 0 or 1 to achieve the goal

57

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

$$g_1(a,b,c) = a'b'c + a'bc + ab'c' + abc$$

$$= m_1 + m_3 + m_4 + m_7$$

$$= \sum m(1,3,4,7)$$

$$g_2(a,b,c) = (a+b+c)(a'+b'+c)$$

$$= M_0 M_6$$

$$= \prod M(0,6)$$

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

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

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

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

A: Yes

B: No

the rest rows false
0, 2, 5, 6

rows 1, 3, 4, 7
True

60

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 $x_2 x'_1 x_0$
- 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.

prime & unprime
 $ab, a'b, a'b', abc, abc'$
 1 2 3 4 5
 $\frac{abc}{a'bc}$
 term | neighbor | 2
 terms 1 & 3 $b \cdot b = 0$

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

ab b'cde are neighbors | *abc abd are not "*

Examples

For two minterms $ab\bar{c}d'$ and $ab'cd'$, are they adjacent?

- A. Yes
- B. No

ab abc are not neighbors
ab' a'bc are not "

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

neighbor: two product terms differ by one variable x in the sense that x & x' appear in these two terms

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

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

Example: The truth table on right has

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

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

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

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

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

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

8