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

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

Definitions

- Literals x_i or x_i'
- Product Term $x_2x_1'x_0$
- 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: Two minterms are adjacent if they differ by only one variable.

Definitions

- Minterm of n variables: A product of n literals in which every variable appears exactly once.
  - E.g. For funciton 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: Two minterms are adjacent if they differ by only one variable.
  - E.g. abcde and a’bcde are adjacent, while a’b’cde and abc’d’e’ are not

iClicker

For two terms 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.

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.

How to completely specify the truth table in canonical form?

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

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

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

Don't care $\mathcal{D}$: 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,b)=(1,0)

$\mathcal{R}$ covers input patters (a,b)=(0,0) and (1,1)

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

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

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

Reducing Incompletely Specified Functions

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

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

A. F(1,1)=1
B. F(1,1)=0
C. Neither A or B
D. Both A and B

Reducing Incompletely Specified Functions

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

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

$$F(a,b)=a$$

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 => Logic Diagram

Karnaugh Map: A 2-dimensional truth table

Flow 1:
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:
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

Truth Table vs. Karnaugh Map

2-variable function, f(A,B)

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	f(0,0)	f(0,1)
A=1	f(1,0)	f(1,1)

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

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.

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

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

Hence, we have

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

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

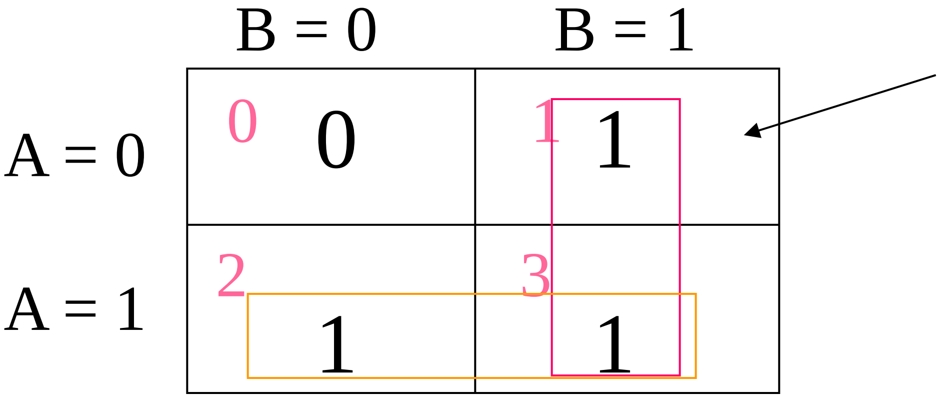
K-Map: Truth Table in 2 Dimensions

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

	B = 0	B = 1
A = 0	0 0	1 1
A = 1	2 1	3 1

K-Map: Truth Table in 2 Dimensions

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



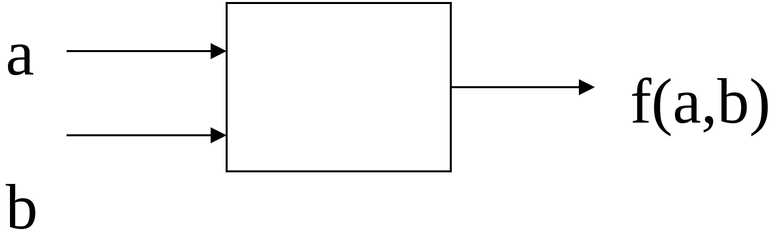
$$f(A,B) = A + B$$

Two Variable K-maps

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)

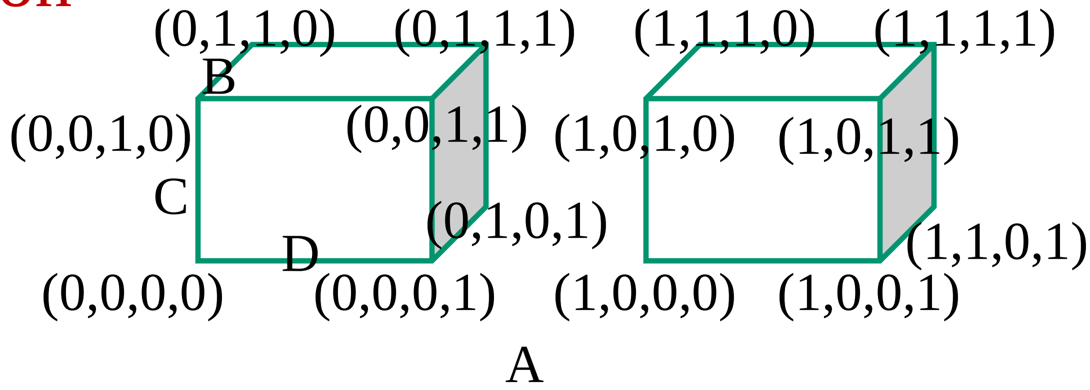
 # possible 2-variable functions:

For 2 variables as inputs, we have $4=2^2$ entries. Each entry can be 0 or 1. Thus we have $16=2^4$ possible functions.



Representation of k-Variable Func.

- Boolean Expression
- Truth Table
- Cube
- K Map
- Binary Decision Diagram



A cube of 4 variables: (A,B,C,D)

Truth table

Id	a	b	c	f (a,b,c)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Corresponding three variable K-map

Gray code (a,b)

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0				
c = 1				

Karnaugh Maps (K-Maps)

- K-maps minimize equations graphically
- Note that the label decides the order in the map

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	0	0	0

Y: C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

K-map

- Circle 1’s in adjacent squares
- Find rectangles which correspond to product terms in Boolean expression

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	0	0	0

$$y(A,B)=A'B'C'+A'B'C=A'B'(C'+C)=A'B'$$

Another 3-Input truth table

Id	a	b	c	f (a,b,c)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	X
7	1	1	1	1

Corresponding K-map

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0 0	2 1	6 X	4 1
c = 1	1 0	3 0	7 1	5 1

Another 3-Input truth table

Id	a	b	c	f (a,b,c)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	X
7	1	1	1	1

Corresponding K-map

b = 1

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 0	2: 1	6: X	4: 1
c = 1	1: 0	3: 0	7: 1	5: 1

a = 1

f(a,b,c) = a + bc’

3 Input Truth table with Don’t cares

Id	a	b	c	f (a,b,c,d)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	X
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Corresponding K-map

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 1	2: X	6: 0	4: 1
c = 1	1: 1	3: 0	7: 0	5: 1

PI Q: In the above case the minimal Boolean expression is obtained when the don’t care term is:

A. Included i.e. Circled in with one or more minterms that evaluate to 1

B. Excluded when grouping the minterms that evaluate to one

3 Input Truth table with Don’t cares

Id	a	b	c	f (a,b,c,d)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	X
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Corresponding K-map

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 1	2: X	6: 0	4: 1
c = 1	1: 1	3: 0	7: 0	5: 1

PI Q: In the above case the minimal Boolean expression is obtained when the don’t care term is:

A. Included i.e. Circled in with one or more minterms that evaluate to 1

B. Excluded when grouping the minterms that evaluate to one

3 Input Truth table with Don't cares

Id	a	b	c	f (a,b,c,d)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	X
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Corresponding K-map

b = 1

	(0,0)	(0,1)	(1,1)	(1,0)
c = 0	0: 1	2: X	6: 0	4: 1
c = 1	1: 1	3: 0	7: 0	5: 1

a = 1

f(a,b,c) = b'

Proof of Consensus Theorem using K Maps

Consensus Theorem: A' B+AC+BC=A' B+AC

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Y: C \ AB	00	01	11	10
0	**0**	**1**	0	0
1	**0**	**1**	**1**	**1**

Y: C \ AB	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$