Engr354: Digital Logic Circuits

Chapter 4: Logic Optimization

Curtis Nelson

Logic Optimization

In chapter 4 you will learn about

- Synthesis of logic functions;
- Analysis of logic circuits;
- Techniques for deriving minimum-cost implementations;
- Graphical representation of logic functions (Karnaugh maps);
- Entered-variable mapping;
- Use of CAD tools to implement logic functions.

Motivation

- It is difficult to minimize a function using algebraic manipulation alone because it is not systematic;
- Graphical techniques allow for a more systematic (and visual) approach to minimization;
- Although software tools are used for logic optimization, designers like us must understand the process;
- This chapter presents methods for logic minimization that can be automated using CAD tools.

Example Function

- Minimizing with algebra can be difficult:

Row Number	a b c	f
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	1
6	1 1 0	1
7	1 1 1	0

The function $f = \Sigma\ m(0, 2, 4, 5, 6)$

2-Variable Karnaugh Map

- Logical adjacencies

a	b	
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

(a) Truth table

b \ a	0	1
0	m_0	m_2
1	m_1	m_3

(b) Karnaugh map

Example 2-Variable Function

a b	f(a,b)
0 0	1
0 1	1
1 0	0
1 1	1

The function $f = \Sigma\, m(0, 1, 3)$

3-Variable Karnaugh Map

a	b	c	
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

(a) Truth table

c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

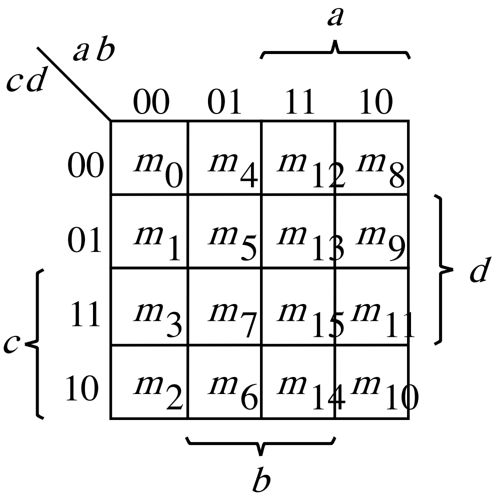
(b) Karnaugh map

Example 3-Variable Function

Minterm	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	1
6	1 1 0	1
7	1 1 1	0

The function $f = \Sigma\, m(0, 2, 4, 5, 6)$

4-Variable Karnaugh Map



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

4-Variable Example

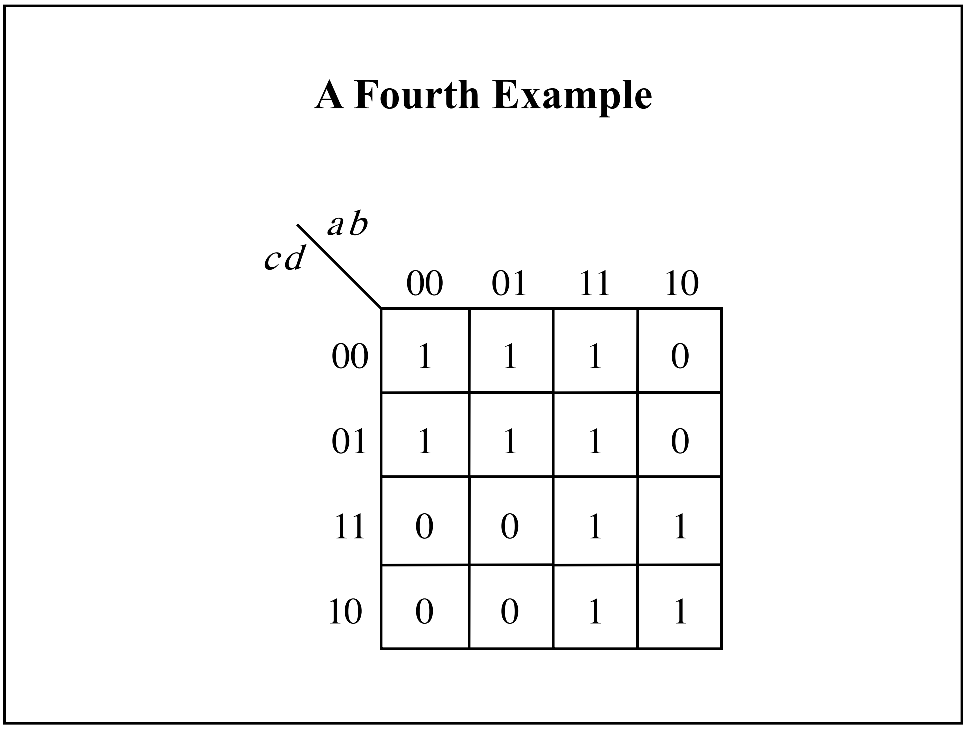
$cd \backslash ab$	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	0	1
10	1	0	0	1

A Second Example

cd \ *ab*	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	1	1	1
10	1	1	1	1

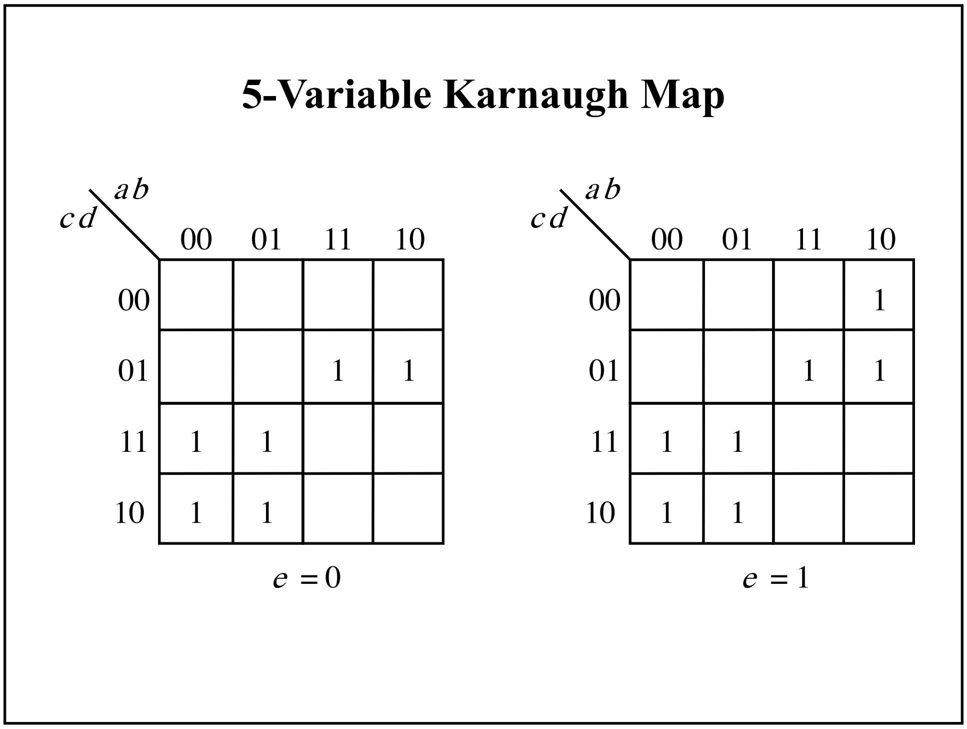
A Third Example

cd \ *ab*	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	1	1	1	0
10	1	1	0	1



A Fourth Example

cd \ ab	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	0	0	1	1
10	0	0	1	1



5-Variable Karnaugh Map

cd \ ab	00	01	11	10
00				
01			1	1
11	1	1		
10	1	1		

$e = 0$

cd \ ab	00	01	11	10
00				1
01			1	1
11	1	1		
10	1	1		

$e = 1$

Terminology

- A variable either uncomplemented or complemented is called a *literal;*
- A product term that indicates when a function is equal to 1 is called an *implicant;*
- An implicant that cannot be combined into another implicant that has fewer literals is called a *prime implicant;*
- A *cover* is a collection of implicants that accounts for all input combinations in which a function evaluates to 1;
- An *essential prime implicant* includes a minterm covered by no other prime;
- *Cost* is the number of gates plus the number of gate inputs, assuming primary inputs are available in both true and complemented form.

Minimization Procedure as per CAD Tools

- Generate all prime implicants;
- Find all essential prime implicants;
- If essential primes do not form a cover, then select a minimal set of non-essential primes.

3-Variable Example

c \ *a b*	00	01	11	10
0	1	1	0	0
1	1	1	1	0

Three-variable function $f = \Sigma\ m(0, 1, 2, 3, 7)$

First 4-Variable Example

c d \ *a b*	00	01	11	10
00				
01		1	1	
11	1	1		1
10	1	1	1	1

Four-variable function $f = \Sigma\ m(2, 3, 5, 6, 7, 10, 11, 13, 14)$

Second 4-Variable Example

cd \ *ab*	00	01	11	10
00	1	1	1	1
01			1	
11			1	1
10				1

The function $f = \Sigma\, m(0, 4, 8, 10, 11, 12, 13, 15)$

Third 4-Variable Example

cd \ *ab*	00	01	11	10
00	1	1		
01		1	1	
11			1	1
10	1			1

The function $f = \Sigma\, m(0, 2, 4, 5, 10, 11, 13, 15)$

Minimization of POS Forms

- Find a cover of the 0’s and form maxterms

$c \backslash ab$	00	01	11	10
0	1	1	0	0
1	1	1	1	0

Incompletely Specified Functions

- Often, certain input conditions cannot occur;
- These are called *don’t cares;*
- A function with don’t cares is called an *incompletely specified function;*
- Don’t cares can be used to improve the quality of the logic designed;
- This book uses a “*d*” to indicate a don’t care situation whereas standard industry practice typically uses a “ϕ”.

Example 4-Variable Function

$cd \backslash ab$	00	01	11	10
00	0	1	d	0
01	0	1	d	0
11	0	0	d	0
10	1	1	d	1

$f = \Sigma\, m(2, 4, 5, 6, 10) + d(12, 13, 14, 15)$

Entered Variable (EV) Mapping

- Allows many variables to be presented using a reduced size K-map;
- Occurs quite frequently in digital systems, especially state machines;
- Requires K-map compression and expansion;
- References (Entered-variable mapping or map-entered variables)
  - Tinder, Engineering Digital Design, Second Edition (Library)
  - Other digital logic textbooks
  - YouTube
  - Internet

Entered Variable (EV) Mapping Example

a	b	c	*f*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$f = \Sigma\, m(2, 5, 6, 7)$

c \ *ab*	00	01	11	10
0	0	1	1	0
1	0	0	1	1

f = a’bc’ + ab’c + abc’ + abc
= bc’ + ac

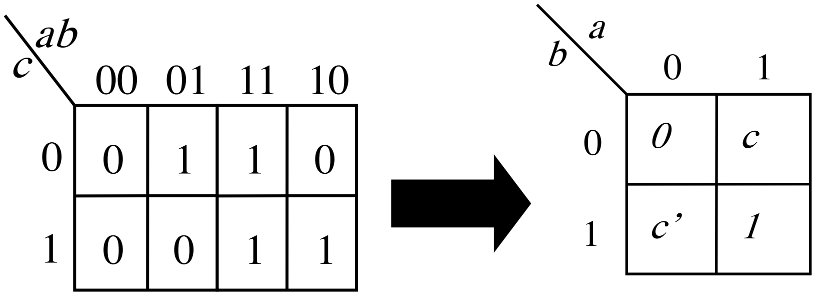
Entered Variable Truth Table Compression

a	b	c	*f*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

➡

a	*b*	*f*
0	0	*0*
0	1	*c’*
1	0	*c*
1	1	*1*

Entered Variable Map Compression

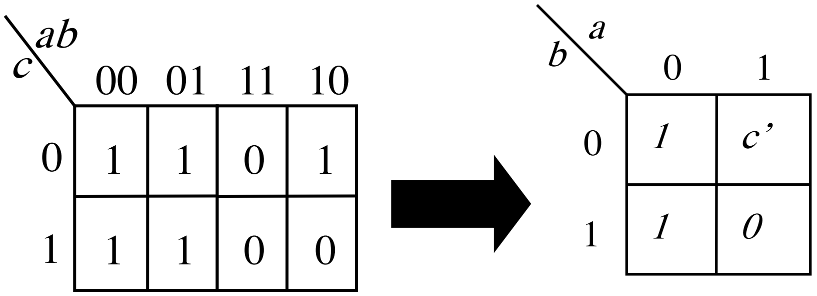


c \ ab	00	01	11	10
0	0	1	1	0
1	0	0	1	1

b \ a	0	1
0	*0*	*c*
1	*c'*	*1*

f = a'bc' + ab'c + abc' + abc
= bc' + ac

Three Variable Compression Example



c \ ab	00	01	11	10
0	1	1	0	1
1	1	1	0	0

b \ a	0	1
0	*1*	*c'*
1	*1*	*0*

1 = c + c'

0 = cc'

Four Variable Compression Example

cd \ ab	00	01	11	10
00	1	1		
01		1	1	
11			1	1
10	1			1

The function $f = \Sigma\ m(0, 2, 4, 5, 10, 11, 13, 15)$

c \ a b	00	01	11	10
0	d’	1	d	0
1	d’	0	d	1

b \ a	0	1
0	c’d’+cd’ =d’	c’0+c1 = c
1	c’1 + c0 = c’	c’d+cd=d

Other Examples

- Expansion;
- Compression and expansion using don’t cares.

Multiple-Output Circuits

- Necessary to implement multiple functions;
- Circuits can be combined to obtain lower cost solution by sharing some gates;
- I only mention this in passing.

Example of Multiple-Output Synthesis

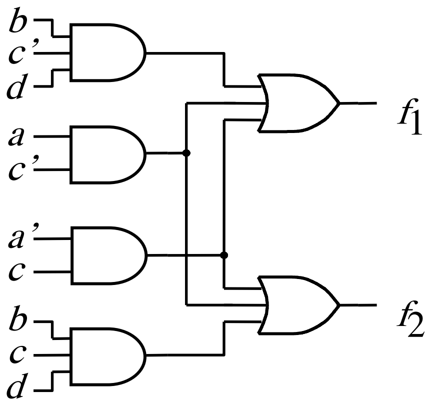
cd \ ab	00	01	11	10
00			1	1
01		1	1	1
11	1	1		
10	1	1		

(a) Function f_1

cd \ ab	00	01	11	10
00			1	1
01			1	1
11	1	1	1	
10	1	1		

(b) Function f_2

Example Multiple-Output Circuit



Combined circuit for f_1 and f_2

Multilevel Synthesis

- SOP or POS circuits have two levels of gates;
- These are only efficient for functions with a few inputs;
- Circuits with many inputs can lead to *fan-in* problems
  - *fan-in* is the number of inputs to a gate;
- Multilevel circuits can also be more area efficient;
- Example: f = ab + ac = a(b+c)

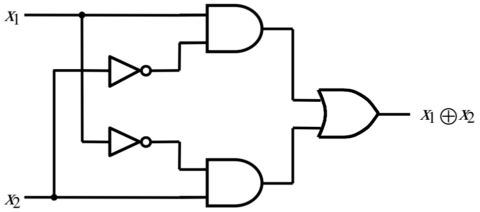
Karnaugh Maps for XOR's and XNOR's

- These functions tend to have non-adjacent, symmetrical patterns in the K-maps.

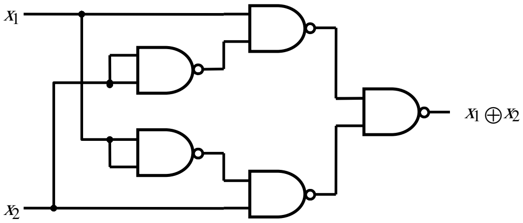
$b \backslash a$	0	1
0		1
1	1	

$$f = ab' + a'b = a \oplus b$$

Implementation of XOR



Sum-of-products Implementation



NAND Gate Implementation

Karnaugh Maps for XOR's and XNOR's

$cd \backslash ab$	00	01	11	10
00	1	1		
01			1	1
11	1	1		
10			1	1

The function $f = \Sigma\, m(0, 3, 4, 7, 9, 10, 13, 14)$

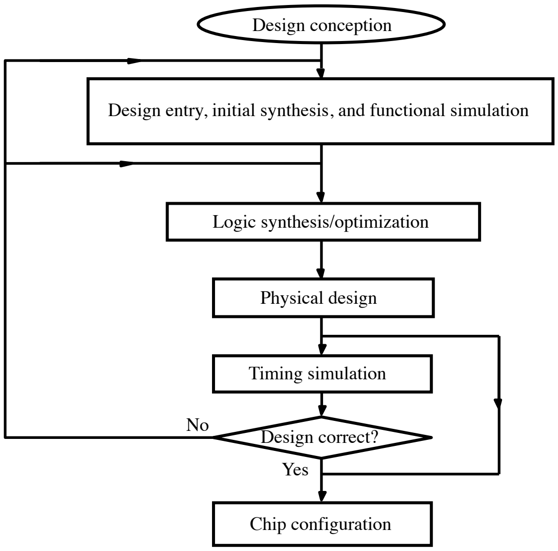
Issues We Will Not Cover

- Multilevel synthesis
  - Factoring;
  - Functional decomposition.
- Analysis of multilevel circuits
- Cubical representation
  - Cubes and Hypercubes.
- Tabular methods for minimization (Quine-McCluskey method)
  - Generation of prime implicants;
  - Determination of minimum cover.
- Cubical techniques for minimization

CAD Tools

- *espresso* – finds exact and heuristic solutions to a 2-level synthesis problem.
- *sis* – performs multilevel logic synthesis.
- Numerous commercial CAD packages are available from Cadence, Mentor, Synopsys, and other Electronic Design Automation (EDA) vendors.

A Complete CAD system

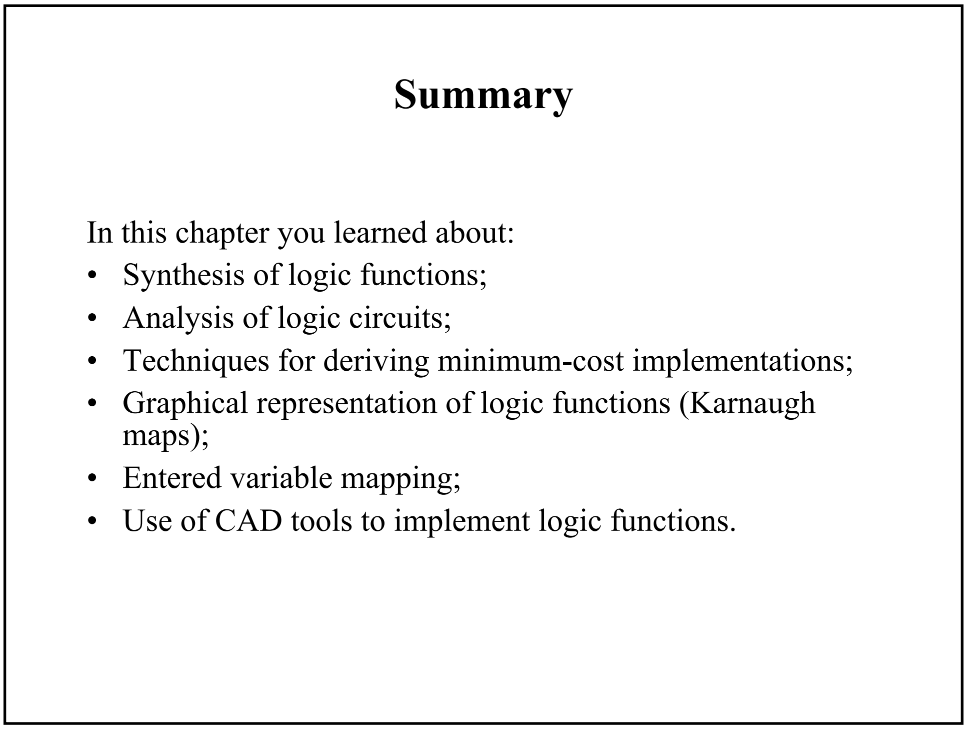


Physical Design

- *Physical design* determines how logic is to be implemented in the target technology
  - *Placement* determines where in the target device a logic function is realized;
  - *Routing* determines how devices are to be interconnected using wires.

Timing Simulation

- Functional simulation does not consider signal propagation delays
  - After physical design, more accurate timing information is available;
  - *Timing simulation* can be used to check if a design meets performance requirements.



Summary

In this chapter you learned about:

- Synthesis of logic functions;
- Analysis of logic circuits;
- Techniques for deriving minimum-cost implementations;
- Graphical representation of logic functions (Karnaugh maps);
- Entered variable mapping;
- Use of CAD tools to implement logic functions.