Digital Logic

COMP370
Introduction to Computer Architecture

Logic Simplification

- It is frequently possible to simplify a logical expression. This makes it easier to understand and requires fewer gates to implement.
- There are several simplification techniques including Boolean algebra and Karnaugh maps.

Boolean Simplification Example

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

F = A'B'C' + A'B'C + A'BC' + ABC'

Applying Boolean Identities

F = A'B'C' + A'B'C + A'BC' + ABC'	Rule
F = A'B'(C' + C) + A'BC' + ABC'	Distribution
F = A'B'(1) + A'BC' + ABC'	Complement
F = A'B' + A'BC' + ABC'	Identity
F = A'B' + BC'(A' + A)	Distribution
F = A'B' + BC'(1)	Complement
F = A'B' + BC'	Identity

Karnaugh maps

- Another way to simplify Boolean expressions is to use Karnaugh maps
- Karnaugh maps use a visual table to help identify parts of a Boolean expression that can be combined.
- It only works well for Boolean expressions that have 3 or 4 input variables.

3 Variable Karnaugh Map

- Create a table of the 8 possible input combinations in a specific *(gray code)* order.

	B'C'	B'C	BC	BC'
A'	0	1	3	2
A	4	5	7	6

The numbers shown in the table are the decimal value of the three input bits when considered as a binary number.

Karnaugh Example

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Copy the output values into the Karnaugh map.

	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	0	0	1

Grouping Ones

- To simplify the equation, you want to group the squares with a "1" into as large a group as you can.
- Imagine that the table wraps around both sides and top and bottom. It is a *Torus*
- You want to create big groups rather than small groups.
- A single square can be in multiple groups if it helps make big groups.

Grouping Ones

- To simplify the equation, you want to group the squares with a "1" into as large a group as you can.

	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	0	0	1

F = A'B' + BC'

Group of 8

- If every square is a "1" then the function is always true.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = 1

Groups of 4

- A group of 4 indicates a single variable.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = A'   F = A

Groups of 4

- A group of 4 indicates a single variable.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = B'   F = B

Groups of 4

- A group of 4 indicates a single variable.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = C'   F = C

Mapping Column to Variable

- Each column represents a pair of variables.

B'   B

	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	0	0	1

C

C'

Groups of 2

- A group of 2 indicates a pair of variables.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = B'C'   F = B'C   F = BC   F = BC'

Groups of 2

- A group of 2 indicates a pair of variables.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = A'B'   F = AB'   F = A'B   F = AB

Groups of 2

- A group of 2 indicates a pair of variables.

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

F = A'C' F = A'C F = AC' F = AC

Groups of One

- If a single square in the Karnaugh map cannot be combined with any other then it is expressed by all three variables.

	B'C'	B'C	BC	BC'
A'	0	1	0	0
A	0	0	0	1

F = A'B'C + ABC'

Karnaugh Example

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Copy the output values into the Karnaugh map.

	B'C'	B'C	BC	BC'
A'	1	1	0	0
A	0	1	0	0

Karnaugh Example

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Copy the output values into the Karnaugh map.

	B'C'	B'C	BC	BC'
A'	1	1	0	0
A	0	1	0	0

F = A'B' + B'C

Simplify this Truth Table

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

	B'C'	B'C	BC	BC'
A'				
A				

Simplified Truth Table

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

	B'C'	B'C	BC	BC'
A'	0	1	1	1
A	0	0	1	1

F = B + A'C

Simplify Another Truth Table

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

	B'C'	B'C	BC	BC'
A'				
A				

Another Simplified Truth Table

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

	B'C'	B'C	BC	BC'
A'	1	1	0	1
A	0	1	0	0

F = A'C' + B'C

Simplify the Boolean Equation

F = A’B’C’ + A’BC + ABC

	B’C’	B’C	BC	BC’
A’				
A				

Simplified Boolean Equation

F = A’B’C’ + A’BC + ABC

	B’C’	B’C	BC	BC’
A’	1	0	1	0
A	0	0	1	0

F = A’B’C’ + BC

Ignored Input Combinations

- Sometimes the logical system will not have certain input combinations.
- It may not matter what the output is for input combinations that will not occur.
- Karnaugh map squares can be marked with a “d” for Don’t care.
- A “don’t care” may be included to make a big group, but do not have to be included.

Truth Table with “Don’t Care”

A	B	C	F
0	0	0	d
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	d
1	1	1	0

Note that one “d” was included to make a big group while the other was ignored.

	B’C’	B’C	BC	BC’
A’	d	1	0	0
A	1	1	0	d

F = B’

4 Input Karnaugh maps

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Note the Gray Code order both horizontally and vertically.
The top and bottom are adjacent as well as the left and right.

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

F = C’ F = C

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

F = A’ F = A

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

F = B’ F = B

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

F = D’ F = D

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

4 Input Example

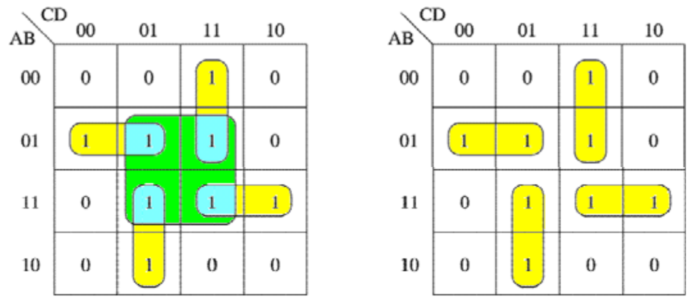
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

F = AB + C’D

	C’D’	C’D	CD	CD’
A’B’	0	1	0	0
A’B	0	1	0	0
AB	1	1	1	1
AB’	0	1	0	0

Unnecessary Groups

- Avoid groups that do not contain any squares that are not already in a group.



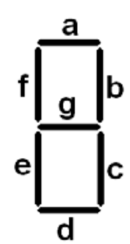
(a) Nonminimal simplification (b) Minimal simplification

Karnaugh Map Process

- Fill the Karnaugh map with 0's, 1's and d's
- Group the 1's into as big as group as possible. Include d's only if necessary.
- Do not include 0's in the groups.
- Write the Boolean expression for each group with the groups ORed together.

7 Segment Display

A	*B*	*C*	*D*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9



Inputs 0xA – 0xF should not occur, so the output is *"don't care"*

Segment "b" of Display

A	*B*	*C*	*D*	*b*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

	C'D'	C'D	CD	CD'
A'B'	1	1	1	1
A'B	1	0	1	0
AB	d	d	d	d
AB'	1	1	d	d

Segment “b” of Display

A	*B*	*C*	*D*	*b*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

	C’D’	C’D	CD	CD’
A’B’	1	1	1	1
A’B	1	0	1	0
AB	d	d	d	d
AB’	1	1	d	d

seg b = B’ + C’D’ + CD

Segment “a” of Display

A	*B*	*C*	*D*	*a*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

	C’D’	C’D	CD	CD’
A’B’	1	0	1	1
A’B	0	1	1	0
AB	d	d	d	d
AB’	1	1	d	d

Segment “a” of Display

A	*B*	*C*	*D*	*a*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1

	C’D’	C’D	CD	CD’
A’B’	1	0	1	1
A’B	0	1	1	0
AB	d	d	d	d
AB’	1	1	d	d

sega=B’D’ + AD + CD + AB’

Simplify Segment “f”

A	*B*	*C*	*D*	*f*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1

	C’D’	C’D	CD	CD’
A’B’				
A’B				
AB				
AB’				

Simple Segment "f"

A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1

	C'D'	C'D	CD	CD'
A'B'	1	0	0	0
A'B	1	1	0	1
AB	d	d	d	d
AB'	1	1	d	d

F = A + C'D' + BC' + BD'

Simplify Segment "d"

A	B	C	D	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0

	C'D'	C'D	CD	CD'
A'B'				
A'B				
AB				
AB'				

Simplified Segment "d"

A	B	C	D	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0

	C'D'	C'D	CD	CD'
A'B'	1	0	1	1
A'B	0	1	0	1
AB	d	d	d	d
AB'	1	0	d	d

F = B'D' + B'C + CD' + BC'D

Karnaugh Maps for Programming

```
if (((S <6)&&(L>10)) ||
   ((S>=6)&&(N==G)&&(L>10)) ||
   ((N==G)&&(L<=10)) ||
   ((S>=6)&&(N==G)) ||
   ((S>=6)&&(N!=G)&&(L>10)) )  {

        print “OK”;
}
```

Fill Table with IF Clauses

S is (S>=6) L is (L>10) N is (N==G)

S' is (S<6) L' is (L<=10) N' is (N!=G)

	S'N'	S' N	S N	S N'
L'	0	1	1	0
L	1	1	1	1

```
if ((N==G) || (L>10))
```