Lecture – 5

Chapter 2: Sections 4-7

Outline

- Boolean Functions
- What are Canonical Forms?
- Minterms and Maxterms
- Index Representation of Minterms and Maxterms
- Sum-of-Minterm (SOM) Representations
- Product-of-Maxterm (POM) Representations
- Conversions between Representations

Boolean Algebra

Boolean algebra allows us to apply provable mathematical principles to help us design logical circuits

An algebraic structure defined on a set of at least two elements B={0, 1}, together with three binary operators (denoted +, · and ') that satisfies the following basic identities:

Properties of Boolean Algebra

Closure: The structure is closed with respect to the operator +, ·, '

Postulate: 1. closure: the structure is closed with respect to the operator +, ·, '

2.	(a) $X + 0 = X$	(b) $X \cdot 1 = X$	identity
3.	(a) $X + Y = Y + X$	(b) $XY = YX$	Commutative
4.	(a) $X(Y + Z) = XY + XZ$	(b) $X + YZ = (X + Y)(X + Z)$	Distributive
5.	(a) $X + X' = 1$	(b) $X \cdot X' = 0$	Complement

Theorem:

1.	(a) $X + X = X$	(b) $X \cdot X = X$	
2.	(a) $X + 1 = 1$	(b) $X \cdot 0 = 0$	
3.	(a) $(X')' = X$		
4.	(a) $(X + Y) + Z = X + (Y + Z)$	(b) $(XY)Z = X(YZ)$	Associative
5.	(a) $(X + Y)' = X' \cdot Y'$	(b) $(X \cdot Y)' = X' + Y'$	DeMorgan
6.	(a) $X + XY = X$	(b) $X(X+Y) = X$	absorption

Boolean Functions

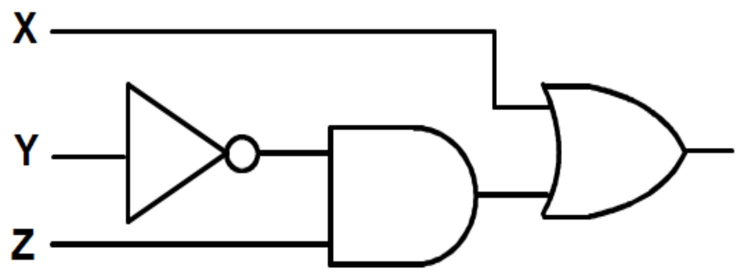
Truth Table

X Y Z	$F = X + Y' \cdot Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equation

$$F = X + Y'Z$$

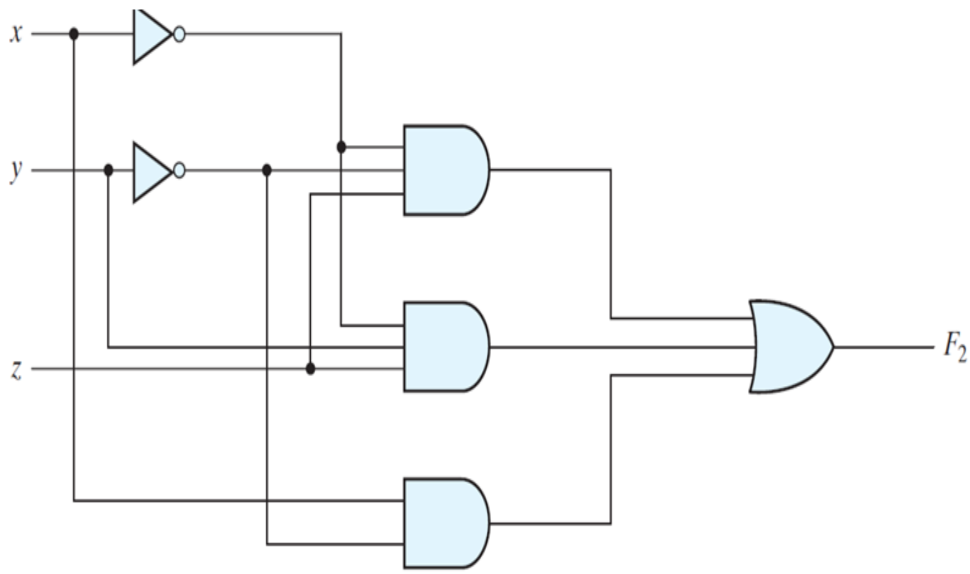
Logic Diagram



- Boolean function described by an algebraic expression consists of binary variables, 0, 1 and logic operation symbols.
- Boolean equations, truth tables and logic diagrams describe the same function!
- Truth tables are unique; expressions and logic diagrams are not. This gives flexibility in implementing functions.

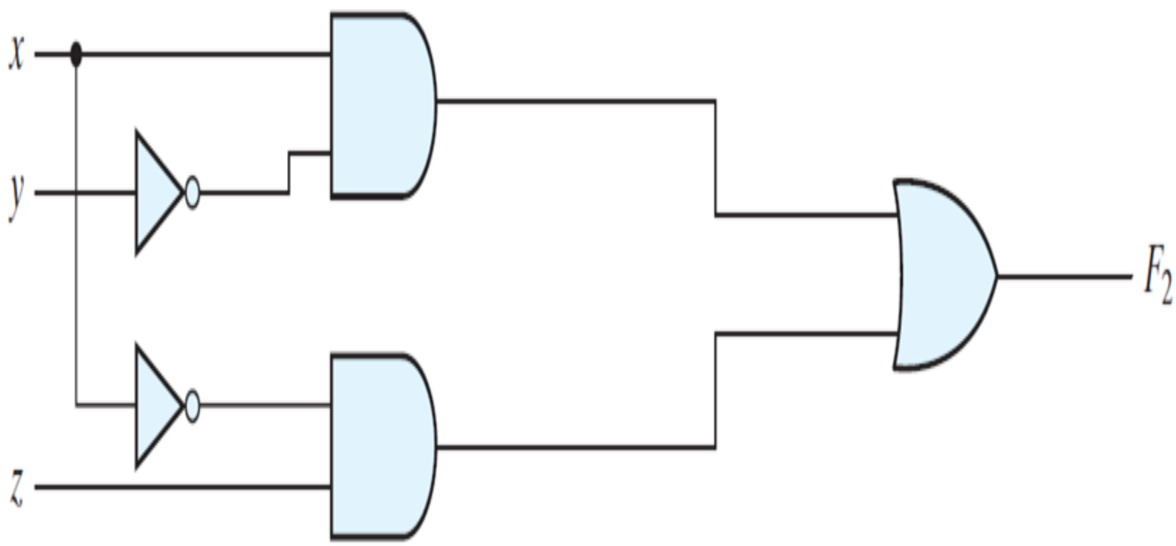
Boolean Functions

$F_2 = x'y'z + x'yz + xy'$



Boolean Functions

$$F2=x'y'z+x'yz+xy'=x'z(y'+y)+xy'=x'z+xy'$$



By manipulating a boolean expression, obtain a simpler expression and thus reduce the number of gates and the number of inputs (reduce the cost of a circuit)

Expression Simplification

- **Each term requires a gate.**
- **Literal: a single variable within a term, in complemented or uncomplemented form.**
  $F_2=x'y'z+x'yz+xy'$ **(3 terms, 8 literal)**
  $F_2=x'y'z+x'yz+xy'=x'z+xy'$ **(2 terms, 4 literal)**

- **Simplify to contain the smallest number of literals (complemented and uncomplemented variables):**
  $(x+y)(x+y')$

FUNCTION EXAMPLE

Functions of Three Variables

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- **f_1 = x'y'z+xy'z'+xyz**

  = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)

- **f_2 = x'yz+xy'z+xyz'+xyz**

  = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)

Minterm Function Example

Example: f1 = x'y'z + xy'z' + xyz

x y z	index	x'y'z+		xy'z'	+	xyz	= f1
0 0 0	0	0	+	0	+	0	= 0
0 0 1	1	1	+	0	+	0	= 1
0 1 0	2	0	+	0	+	0	= 0
0 1 1	3	0	+	0	+	0	= 0
1 0 0	4	0	+	1	+	0	= 1
1 0 1	5	0	+	0	+	0	= 0
1 1 0	6	0	+	0	+	0	= 0
1 1 1	7	0	+	0	+	1	= 1

Maxterm Function Example

- **Example: Implement f1 in maxterms:**

$f_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$

x y z	i	x+y+z ·	x+y'+z ·	x+y'+z' ·	x'+y+z' ·	x'+y'+z =	f1
0 0 0	0	0 ·	1 ·	1 ·	1 ·	1 =	0
0 0 1	1	1 ·	1 ·	1 ·	1 ·	1 =	1
0 1 0	2	1 ·	0 ·	1 ·	1 ·	1 =	0
0 1 1	3	1 ·	1 ·	0 ·	1 ·	1 =	0
1 0 0	4	1 ·	1 ·	1 ·	1 ·	1 =	0
1 0 1	5	1 ·	1 ·	1 ·	0 ·	1 =	0
1 1 0	6	1 ·	1 ·	1 ·	1 ·	0 =	0
1 1 1	7	1 ·	1 ·	1 ·	1 ·	1 =	1

Overview – Canonical Forms

- **What are Canonical Forms?**
- **Minterms and Maxterms**
- **Index Representation of Minterms and Maxterms**
- **Sum-of-Minterm (SOM) Representations**
- **Product-of-Maxterm (POM) Representations**
- **Conversions between Representations**

Canonical Forms

It is useful to specify Boolean functions in a form that:

- **Allows comparison for equality.**
- **Has a correspondence to the truth tables**

Canonical Forms in common usage:

- **Sum of Minterms (SOM)**
- **Product of Maxterms (POM)**

Minterms

- **Minterms** are AND terms with every variable present in either true or complemented form.

- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., x'), there are 2^n minterms for n variables.

- **Example:** Two variables (X and Y)produce

  2 x 2 = 4 combinations:
  XY (both normal)
  XY' (X normal, Y complemented)
  X'Y (X complemented, Y normal)
  X'Y' (both complemented)

- Thus there are four minterms of two variables.

Maxterms

- **Maxterms** are OR terms with every variable in true or complemented form.

- Given that each binary variable may appear normal (e.g., x) or complemented (e.g., x'), there are 2^n maxterms for n variables.

- **Example:** Two variables (X and Y) produce

  2 x 2 = 4 combinations:
  X+Y (both normal)
  X+Y' (x normal, y complemented)
  X'+Y (x complemented, y normal)
  X'+Y' (both complemented)

MAXTERMS AND MINTERMS

Minterms and Maxterms for Three Binary Variables

				Minterms		Maxterms	
Index	x	y	z	Term	Designation	Term	Designation
0	0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
1	0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
2	0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
3	0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
4	1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
5	1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
6	1	1	0	xyz'	m_6	$x' + y' + z$	M_6
7	1	1	1	xyz	m_7	$x' + y' + z'$	M_7

- **Minterms** and **Maxterms** are designated with a subscript
  The subscript is a number, corresponding to a binary pattern
  The bits in the pattern represent the complemented or normal

- **For Minterms:**
  - "0" means the variable is "Complemented".
  - "1" means the variable is "Not Complemented" and

- **For Maxterms:**
  - "0" means the variable is "Not Complemented" and
  - "1" means the variable is "Complemented".

Index Examples – Four Variables

Index i	Binary Pattern	Minterm m_i	Maxterm M_i
0	0000	a'b'c'd'	a+b+c+d
1	0001	a'b'c'd	?
3	0011	?	a+b+c'+d'
5	0101	a'bc'd	a+b'+c+d'
7	0111	?	a+b'+c'+d'
10	1010	ab'cd'	a'+b+c'+d
13	1101	abc'd	?
15	1111	abcd	a'+b'+c'+d'

Minterm and Maxterm Relationship

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

- **Review: DeMorgan's Theorem (x+y+z)'=x'y'z' and (xyz)'=x'+y'+z'**

  Thus M_i is the complement of m_i and vice-versa.

- **Since DeMorgan's Theorem holds for *n* variables, the above holds for terms of *n* variables**

- **giving:** $M_i = m_i'$ and $m_i = M_i'$

  Thus M_i is

Observations

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

- In the function tables:
  - Each **min**term has one and only one 1 present in the 2^n terms (a minimum of 1s). All other entries are 0.
  - Each **max**term has one and only one 0 present in the 2^n terms All other entries are 1 (a maximum of 1s).

MINTERM FUNCTION EXAMPLE

Functions of Three Variables

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- **We can implement any function by "ORing" the minterms corresponding to "1" entries in the function table. These are called the minterms of the function.**

- **f1 = x'y'z+xy'z'+xyz= m1 + m4 + m7**

- **f2 = x'yz+xy'z+xyz'+xyz = m3+ m5+ m6+ m7**

Minterm Function Example

- **Example: Find f1 = m1 + m4 + m7**
- **f1 = x'y'z + xy'z' + xyz**

x y z	index	m1	+	m4	+	m7	= F1
0 0 0	0	0	+	0	+	0	= 0
0 0 1	1	1	+	0	+	0	= 1
0 1 0	2	0	+	0	+	0	= 0
0 1 1	3	0	+	0	+	0	= 0
1 0 0	4	0	+	1	+	0	= 1
1 0 1	5	0	+	0	+	0	= 0
1 1 0	6	0	+	0	+	0	= 0
1 1 1	7	0	+	0	+	1	= 1

MAXTERM FUNCTION EXAMPLE

Functions of Three Variables

x	*y*	*z*	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- **We can implement any function by "ANDing" the max-terms corresponding to "0" entries in the function table. These are called the maxterms of the function.**

- **f_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)**

  = M0M2M3M5M6

- **f2 = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)**

  = M0M1M2M4

Maxterm Function Example

- **Example: Implement f1 in maxterms:**
- **$f_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$**

  $= M_0M_2M_3M_5M_6$

x y z	i	M_0 · M_2 · M_3 · M_5 · M_6	= F1
0 0 0	0	0 · 1 · 1 · 1 · 1	= 0
0 0 1	1	1 · 1 · 1 · 1 · 1	= 1
0 1 0	2	1 · 0 · 1 · 1 · 1	= 0
0 1 1	3	1 · 1 · 0 · 1 · 1	= 0
1 0 0	4	1 · 1 · 1 · 1 · 1	= 1
1 0 1	5	1 · 1 · 1 · 0 · 1	= 0
1 1 0	6	1 · 1 · 1 · 1 · 0	= 0
1 1 1	7	1 · 1 · 1 · 1 · 1	= 1

Conversion Between Forms

To convert between sum-of-minterms and product-of-maxterms form (or vice-versa) we follow these steps:

- Change from products to sums, or vice versa.
- Swap terms in the list with terms not in the list.

- Example:Given F as before: $F(x, y, z) = \Sigma_m(1,3,5,7)$

- Give the other form of the orig

$$F(x, y, z) = \Pi_M(0,2,4,6)$$

Canonical Sum of Minterms

- **Any Boolean function can be expressed as a Sum of Minterms (SOM).**
  - **For the function table, the minterms used are the terms corresponding to the 1's**
  - **For expressions, expand all terms first to explicitly list all minterms. Do this by "ANDing" any term missing a variable v with a term (v+v')**
- **Example:**

  Implement f=x+x'y' as a sum of minterms.
  - **First expand terms: f=x(y+y')+x'y'**
  - **Then distribute terms: f=xy+xy'+ x'y'**
  - **Express as sum of minterms: $f = m_3 + m_2 + m_0$**

Another SOM Example

- **Example:** $F = A + B'C$
- **There are three variables, A, B, and C which we take to be the standard order.**
- **Expanding the terms with missing variables:**
- **Collect terms (removing all but one of duplicate terms):**
- **Express as SOM:**

Shorthand SOM Form

- **From the previous example, we started with:**

  F = A + B'C

- **We ended up with:**

  $$F = m_1+m_4+m_5+m_6+m_7$$

- **This can be denoted in the formal shorthand:**

$$F(A,B,C) = \Sigma_m(1,4,5,6,7)$$

- **Note that we explicitly show the standard variables in order and drop the "m" designators.**

Canonical Product of Maxterms

- **Any Boolean Function can be expressed as a Product of Maxterms (POM).**
    - **For the function table, the maxterms used are the terms corresponding to the 0's.**
    - **For an expression, expand all terms first to explicitly list all maxterms. Do this by first applying the second distributive law , "ORing" terms missing variable v with a term equal to $v \cdot v'$ and then applying the distributive law again.**
- **Example: Convert to product of maxterms:**

$$f(x,y,z)=x+x'y'$$

Apply the distributive law:

$$x + x'y' = (x + x')(x + y') = x + y'$$

Add missing variable z:

$$x+y'+z \cdot z'=(x+y'+z)(x+y'+z')$$

Express as POM: $f = M_2 \cdot M_3$

Another POM Example

- **Convert to Product of Maxterms:**

$$f(x,y,z) = xy + x'z$$

Standard Forms

- **Standard Sum-of-Products (SOP) form: equations are written as an OR of AND terms**

- **Standard Product-of-Sums (POS) form: equations are written as an AND of OR terms**

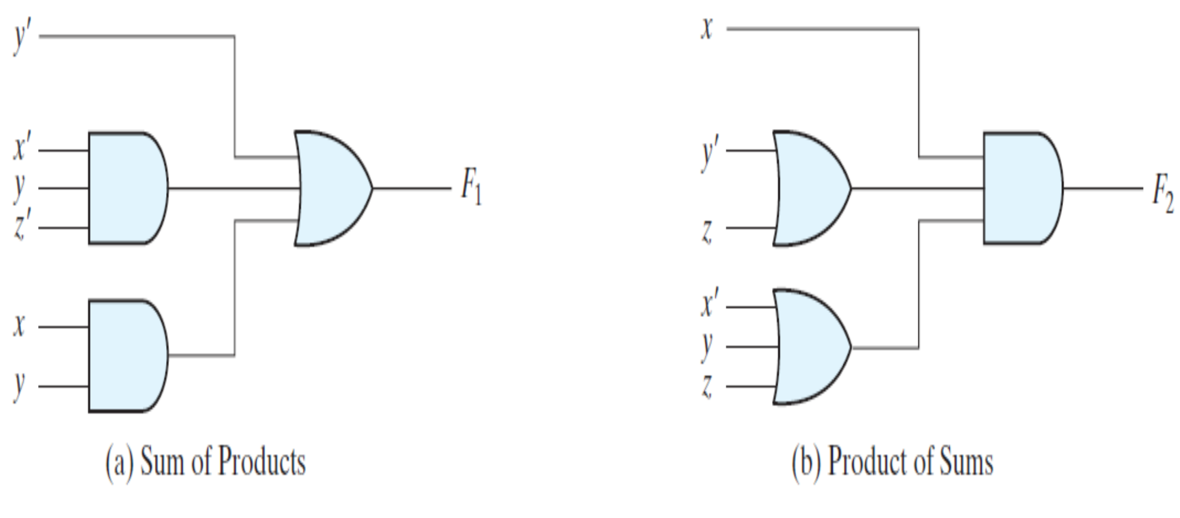
- **Examples:**
  - **SOP:** $ABC + A'B'C + B$
  - **POS:** $(A + B) . (A + B' + C') . C$

- **These "mixed" forms are neither SOP nor POS**
- $(A B + C) (A + C)$
- $A B C' + A C (A + B)$

Standard Sum-of-Products (SOP) and Product-of-Sum (POS)

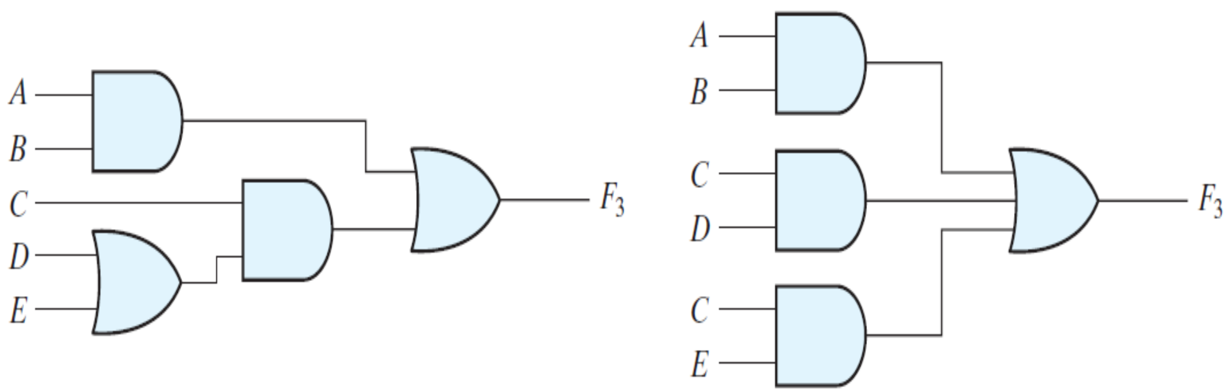
Implementation of SOP (POS) is a two-level network of gates such that:

- **The first level consists of *n*-input AND (OR) gates, and**
- **The second level is a single OR (AND) gate**



AND/OR Two-level Implementation of SOP Expression

- **A Boolean function maybe expressed in a non standard form: F=AB+C(D+E)**
- **It can be changed to standard form.**



(a) $AB + C(D + E)$ (b) $AB + CD + CE$

SOP and POS Observations

The previous examples show that:

- **Canonical Forms (Sum-of-minterms, Product-of-Maxterms), or other standard forms (SOP, POS) differ in complexity**
- **Boolean algebra can be used to manipulate equations into simpler forms.**
- **Simpler equations lead to simpler two-level implementations**

Questions:

- **How can we attain a "simplest" expression?**
- **Is there only one minimum cost circuit?**
- **The next part will deal with these issues.**