DIGITAL DESIGN AND COMPUTER ORGANIZATION

Combinational logic

Department of Computer Science and Engineering

DIGITAL DESIGN AND COMPUTER ORGANIZATION

canonical and Standard forms

Department of Computer Science and Engineering

canonical and Standard forms

From Truth Table to Boolean Formula and its Minimization

- Given a combinational logic circuit or Boolean formula, we have learnt to construct its truth table
- **But, given a truth table, how to construct a Boolean formula (or combinational logic circuit) for it?**
- **Also, as there are multiple Boolean formulas / logic circuits for each truth table, how to pick the minimal one?**
- Above problem is called **logic minimization**

????

canonical and Standard forms

From Truth Table to Boolean Formula and its Minimization

- Given a combinational logic circuit or Boolean formula, we have learnt to construct its truth table
- **But, given a truth table, how to construct a Boolean formula (or combinational logic circuit) for it?**
- **Also, as there are multiple Boolean formulas / logic circuits for each truth table, how to pick the minimal one?**
- Above problem is called **logic minimization**
  - many metrics: smallest, fastest, least power cosumption
  - our metric: smallest two level Sum of Products formula
  - may be more than one solution

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

- A binary variable may appear either in its normal form (x) or in its complement form (x\`). Now consider two binary variables x and y combined with an AND operation. Since each variable may appear in either form, there are four possible combinations: xy , x\`y\`, x\`y , and xy\`. Each of these four AND terms is called a minterm, or a standard product.

- In a similar manner, n variables can be combined to form 2^n minterms. The binary numbers from 0 to 2^n-1 are listed under the n variables.

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms	
x	**y**	**z**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'yz'$	m_2
0	1	1	$x'yz$	m_3
1	0	0	$xy'z'$	m_4
1	0	1	$xy'z$	m_5
1	1	0	xyz'	m_6
1	1	1	xyz	m_7

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

In a similar fashion, n variables forming an OR term, with each variable being primed or unprimed, provide 2^n possible combinations, called **maxterms, or standard sums.**

Table 2.3
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

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

It is important to note that

(1) each maxterm is obtained from an OR term of the n variables, with each variable being unprimed if the corresponding bit is a 0 and primed if a 1, and

(2) each maxterm is the complement of its corresponding minterm and vice versa. A Boolean function can be expressed algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

Table 2.4
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

f1 in Table 2.4 is determined by expressing the combinations 001, 100, and 111 Since each one of these minterms results in f1=1, we have

f1=x\`y\`z+xy\`z\` +xyz
=m1+m4+m7

f2 = x\`yz + xy\`z + xyz \`+ xyz
= m3 + m5 + m6 + m7

Sum of Min Terms

Now consider the complement of a Boolean function. It may be read from the truth table by forming a minterm for each combination that produces a 0 in the function and then ORing those terms. The complement of f1 is read as

$F`_1 = x`y`z` + x`yz` + x`yz + xy`z + xyz`$

Which is

$f1 = (x + y + z)(x + y` + z)(x` + y + z`)(x` + y` + z)$

$= M0 . M2 . M3 . M5 . M6$

Similarly

$f2 = (x + y + z)(x + y + z`)(x + y` + z)(x` + y + z)$

$= M0M1M2M4$

Product of max terms

canonical and Standard forms

Canonical and Standard form (T1: section 2.6)

- The procedure for obtaining the product of maxterms directly from the truth table is as follows: Form a maxterm for each combination of the variables that produces a 0 in the function, and then form the AND of all those maxterms.

- **Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form**

canonical and Standard forms

Canonical and Standard form

- **Application of SOP and POS form:**

SOP Form:

Fire Alarm System

Fire alarm activates if:
Smoke is detected (S)
Temperature is high (T)
CO gas is detected (C)

Fire Alarm = S + T + C
This is a **classic SOP** — the output is true if **any one danger is present**.

POS Form:

Traffic Light Control

Used to control light states **only when certain combinations of sensors are false**.
For example, a green light should **not** be given if any pedestrian button is pressed **or** vehicle is waiting.
POS is used to describe “green light condition” as:
Green = (P' + V') → where P = pedestrian, V = vehicle.

canonical and Standard forms

Canonical and Standard form

Sum of Minterms

- The minterms whose sum defines the Boolean function are those which give the 1's of the function in a truth table
- Since the function can be either 1 or 0 for each minterm, and since there are 2^n minterms, one can calculate all the functions that can be formed with n variables to be 2^{2^n}

- Express the Boolean function F = A + B`C as a sum of minterms. The function has three variables: A, B, and C.

canonical and Standard forms

Canonical and Standard form

Sum of Minterms

- Express the Boolean function F = A + B`C as a sum of minterms. The function has three variables: A, B, and C.

- F = A`B`C + AB`C + AB`C + ABC` + ABC = m1 + m4 + m5 + m6 + m7

- When a Boolean function is in its sum-of-minterms form, it is sometimes convenient to express the function in the following brief notation:
- **F(A, B, C) =∑ (1, 4, 5, 6, 7)**
- **The ∑ summation symbol stands for the OR ing of terms;**
- An alternative procedure for deriving the minterms of a Boolean function is to obtain the truth table of the function directly from the algebraic expression and then read the minterms from the truth table

Canonical and Standard form

Sum of Minterms

The truth table shown in Table 2.5 can be derived directly from the algebraic expression by listing the eight binary combinations under variables A, B, and C and inserting 1's under F for those combinations for which A=1 and B`C=01. From the truth table, we can then read the five minterms of the function to be 1, 4, 5, 6, and 7.

Table 2.5
Truth Table for $F = A + B'C$

A	B	C	F
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

canonical and Standard forms

Canonical and Standard form

Product of Maxterms

Each of the 2^{2^n} functions of n binary variables can be also expressed as a product of maxterms.

To express a Boolean function as a product of maxterms, it must first be brought into a form of OR terms. This may be done by using the distributive law, $x+yz=(x+y)(x+z)$.

Express the Boolean function $F=xy+x`z$ as a product of maxterms.

canonical and Standard forms

Canonical and Standard form

Product of Maxterms

Express the Boolean function F=xy+x`z as a product of maxterms.

First, convert the function into OR terms by using the distributive law:

F=xy+x`z

=(xy+x`)(xy+z)

=(x+x`)(y+x`)(x+z)(y+z)

=(x` +y)(x+z)(y+z)

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

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

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

Each OR term is missing one variable;

=M0M2M4M5

F(x, y, z) =π (0, 2, 4, 5)

π denotes the ANDing of maxterms; the numbers are the indices of the maxterms of the function.

canonical and Standard forms

Conversion between Canonical Forms

The complement of a function expressed as the sum of minterms equals the sum of min terms missing from the original function. This is because the original function is expressed by those minterms which make the function equal to 1, whereas its complement is a 1 for those minterms for which the function is a 0. As an example, consider the function $F(A, B, C) = \sum(1, 4, 5, 6, 7)$

This function has a complement that can be expressed as

$F`(A, B, C) = \pi(0, 2, 3) = m0 + m2 + m3$

Now, if we take the complement of F` by DeMorgan's theorem, we obtain F in a different form:

$F = (m0 + m2 + m3)` = m`_0 . m`_2 . m`_3 = M_0M_2M_3 = \pi\ (0, 2, 3)$

$m`j = Mj$

canonical and Standard forms

Conversion between Canonical Forms

Consider, for example, the Boolean expression $F = xy + x`z$

$F(x, y, z) = \sum(1, 3, 6, 7)$

$F(x, y, z) = \pi(0,2,4,5)$

Table 2.6
Truth Table for $F = xy + x'z$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Minterms

Maxterms

canonical and Standard forms

Standard Forms-quick pointers

There are two types of standard forms: the sum of products and products of sums. The sum of products is a Boolean expression containing AND terms, called product terms, with one or more literals each. The sum denotes the ORing of these terms.

Standard Forms-quick pointers

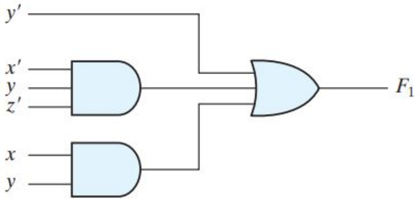
The product is an AND operation. The use of the words product and sum stems from the similarity of the AND operation to the arithmetic product (multiplication) and the similarity of the OR operation to the arithmetic sum (addition). The gate structure of the product-of-sums expression consists of a group of OR gates for the sum terms (except for a single literal), followed by an AND gate, as shown in Fig. 2.3 (b). This standard type of expression results in a two-level structure of gates.

canonical and Standard forms

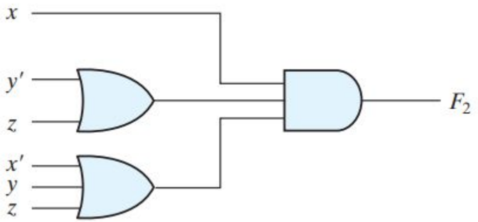
Canonical forms- two level implementation

$$F_1 = y' + xy + x'yz'$$

$$F_2 = x(y' + z)(x' + y + z')$$



(a) Sum of Products



(b) Product of Sums

FIGURE 2.3
Two-level implementation

canonical and Standard forms

Canonical forms- two level implementation

A Boolean function may be expressed in a nonstandard form. For example, the function

$F_3 = AB + C(D + E)$

is neither in sum-of-products nor in product-of-sums form. The implementation of this expression is shown in Fig. 2.4 (a) and requires two AND gates and two OR gates. There are three levels of gating in this circuit.

It can be changed to a standard form by using the distributive law to remove the parentheses:

$F3 = AB + C(D + E) = AB + CD + CE$

Canonical and Standard forms

Canonical forms- three level implementation

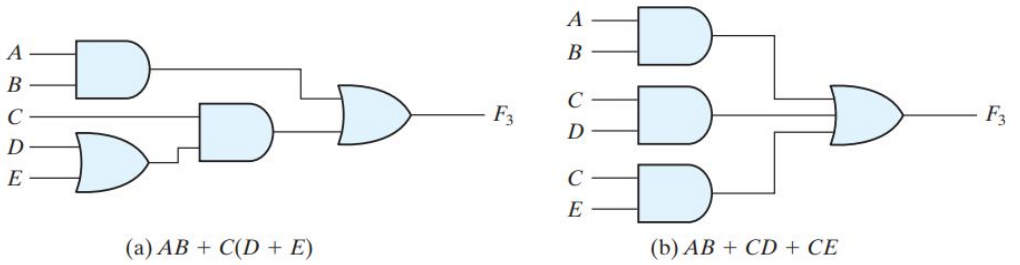


FIGURE 2.4
Three- and two-level implementation

canonical and Standard forms

Canonical forms- three level implementation

In general, a two-level implementation is preferred because it produces the least amount of delay through the gates when the signal propagates from the inputs to the output.

canonical and Standard forms

Questions

Given the Boolean function $f(x, y, z) = \sum m(3, 5, 6, 7)$ (i.e., in SOP form), its equivalent POS (product-of-sum) form is:

A) $\Pi M(3, 5, 6, 7)$
B) $\Pi M(0, 1, 2, 4)$
C) $\Pi M(0, 2, 4)$
D) $\Pi M(1, 3, 5)$

canonical and Standard forms

Questions

Given the Boolean function $f(x, y, z) = \sum m(3, 5, 6, 7)$ (i.e., in SOP form), its equivalent POS (product-of-sum) form is:

A) $\Pi M(3, 5, 6, 7)$

B) $\Pi M(0, 1, 2, 4)$

C) $\Pi M(0, 2, 4)$

D) $\Pi M(1, 3, 5)$

Answer: B

Because the POS corresponds to the maxterms *not included* in the minterm list: $\{0, 1, 2, 4\}$

canonical and Standard forms

The maxterm corresponding to the binary combination X = 1, Y = 0, Z = 1 is:

A) x + y + z

B) x' + y' + z'

C) x' + y + z'

D) x + y' + z

canonical and Standard forms

The maxterm corresponding to the binary combination X = 1, Y = 0, Z = 1 is:

A) x + y + z

B) x' + y' + z'

C) x' + y + z'

D) x + y' + z

Answer: C

THANK YOU

Team DDCO

Department of Computer Science