6.1 Quine-McCluskey Method

There are two main steps

- Eliminate as many literals as possible using $XY+XY'=X$
- Use a prime implicant chart to select a minimum set of prime implicants using Consensus theorem

Determination of Prime Implicant (1st step)

$$XY+XY'=X$$

$$AB'CD'+AB'CD=AB'C$$

1010 + 1011 = 101-

$$X \ Y + X \ Y' = X$$

Determination of Prime Implicant (Example)

Find $f(a,b,c,d)=\Sigma m(0,1,2,5,6,7,8,9,10,14)$

Step One

- Group number of 1
  - (0), (1, 2, 8), (5, 6, 9, 10), (7, 14)
- Apply $XY+XY'=X$
- Go to step one until no more cancellations
- Cancel out duplicate terms
- The result is the terms that are not cancelled out and not combined with others
The Prime Implicant (Example)

\[ f(a, b, c, d) = \sum m(0, 1, 2, 5, 6, 7, 8, 9, 10, 14) \]  \hspace{1cm} (6-2)

is represented by the following list of minterms:

<table>
<thead>
<tr>
<th>group 0</th>
<th>0</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>group 1</td>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>group 2</td>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1001</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1010</td>
</tr>
<tr>
<td>group 3</td>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>1110</td>
</tr>
</tbody>
</table>

The Prime Implicant (Example)

<table>
<thead>
<tr>
<th>TABLE 6-1</th>
<th>Determination of Prime Implicants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column I</td>
<td>Column II</td>
</tr>
<tr>
<td>group 0</td>
<td>0 0000 0 \checkmark 0, 1 0000 \checkmark 0, 1, 8, 9 \checkmark 0000 \checkmark</td>
</tr>
<tr>
<td>group 1</td>
<td>1 0010 0 \checkmark 0, 2 00-0 \checkmark 0, 2, 8, 10 \checkmark 00-0 \checkmark</td>
</tr>
<tr>
<td>group 2</td>
<td>2 0101 0 \checkmark 0, 8 \checkmark 0, 8, 9 \checkmark 08, 9 \checkmark</td>
</tr>
<tr>
<td>group 3</td>
<td>3 0111 0 \checkmark 5, 7 \checkmark 5, 7 \checkmark</td>
</tr>
</tbody>
</table>
6.2 The Prime Implicant Chart

\[ f = a'c'd + a'bd + a'bc + b'c' + b'd' + cd' \]  

(1, 5) (5, 7) (6, 7) (0, 1, 8, 9) (0, 2, 8, 10) (2, 6, 10, 14)

<table>
<thead>
<tr>
<th>Prime Implicant Chart</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 1, 8, 9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 2, 8, 10)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2, 6, 10, 14)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1, 5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5, 7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6, 7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Petrick’s Method (Problem)

\[ F = \sum m(0, 1, 2, 5, 6, 7) \]

<table>
<thead>
<tr>
<th>( P_i )</th>
<th>(0, 1)</th>
<th>(0, 2)</th>
<th>(1, 5)</th>
<th>(2, 6)</th>
<th>(5, 7)</th>
<th>(6, 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 000</td>
<td>( a'b' )</td>
<td>( a'c' )</td>
<td>( b'c' )</td>
<td>( bc' )</td>
<td>( ac )</td>
<td>( ab )</td>
</tr>
<tr>
<td>1 001</td>
<td>( a' )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 010</td>
<td>( 0 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 5 101     | \( 1 \) | \( 1 \) | \( 0 \)
| 6 110     | \( 0 \) | \( 1 \) | \( 1 \)
| 7 111     | \( 0 \) | \( 1 \) | \( 1 \)

\[ P = (P_1 + P_2)(P_1 + P_3)(P_2 + P_4)(P_3 + P_4) \]

Petrick’s Method (Solution)

\[ P = (P_1 + P_2 + P_3 + P_4 + P_5) \]

Next, we use \( X + XY = X \) to eliminate redundant terms from \( P \), which yields

\[ P = P_1P_2P_3 + P_1P_4P_6 + P_2P_4P_5 + P_1P_3P_6 + P_1P_5P_6 + P_2P_3P_6 \]
6.4 Quine-McCluskey with Don’t Care

\[ F(A, B, C, D) = \Sigma m(2, 3, 7, 9, 11, 13) + \Sigma d(1, 10, 15) \]

The don’t care terms are treated like required minterms when finding the prime implicants:

1. 0001 \( \checkmark \)
   (1, 2) \( 0-1 \)
   (1, 3, 9, 11) \( 0-1 \)
2. 0010 \( \checkmark \)
   (1, 9) \( 0-01 \)
   (2, 3, 10, 11) \( 0-1 \)
3. 0011 \( \checkmark \)
   (4, 9) \( 0-001 \)
   (3, 7, 11, 15) \( 0-11 \)
4. 0100 \( \checkmark \)
   (2, 10) \( 0-010 \)
   (9, 11, 13, 15) \( 1-1 \)
5. 1010 \( \checkmark \)
   (3, 7) \( 0-11 \)
6. 0111 \( \checkmark \)
   (3, 11) \( 0-11 \)
7. 1011 \( \checkmark \)
   (9, 11) \( 1-1 \)
8. 1101 \( \checkmark \)
   (9, 13) \( 1-01 \)
9. 1111 \( \checkmark \)
   (10, 11) \( 1-11 \)
10. 1111 \( \checkmark \)
   (7, 18) \( 1-11 \)

The don’t care columns are omitted when forming the prime implicant chart:

\[ F = B'C + CD + AD \]

7.3 Design of 2-level using NAND & NOR

\[ F = A + BC + B'CD = [(A + BC) + B'CD]' \]  
(by 7-11) \( (7-13) \)
\[ = [A' - (BC)' - (B'CD)']' \]  
(by 7-12) \( (7-14) \)
\[ = [A' - (B' + C) - (B + C' + D)']' \]  
(by 7-13) \( (7-15) \)
\[ = A + (B' + C') + (B + C' + D') \]  
(by 7-12) \( (7-16) \)

9.1 Multiplexer

\[ Z = A'I_0 + AI_1 \]

\[ Z = A'B'I_0 + A'B'I_1 + AB'I_2 + ABI_3 \]  
(9-1)
\[ Z = \sum_{i=0}^{2^{n-1}} m_i I_i \]  
(9-2)

9.1 Multiplexer (2)

\[ F(A,B,C) = A'B' + AC \]
\[ F = A'B' + AC \]

MUX Realization of a 4-Variable Function
9.4 Decoders and Encoders

FIGURE 9-13
3-to-8 Line Decoder

a →

b →

c →

3-to-8 line decoder

\[ y_0 = a \bar{b} \bar{c}' \]

\[ y_1 = a \bar{b} c' \]

\[ y_2 = a b' c' \]

\[ y_3 = a b c' \]

\[ y_4 = a b' c \]

\[ y_5 = a b c \]

\[ y_6 = a b' \]

\[ y_7 = a b \]

\[ y_8 = a b' \]

\[ y_9 = a b \]

\[ y_{10} = a b' c \]

\[ y_{11} = a b c \]

\[ y_{12} = a b' \]

\[ y_{13} = a b \]

\[ y_{14} = a b' \]

\[ y_{15} = a b \]

\[ a \quad b \quad c \]

\[ y_0 \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \]

\[ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \]

9.4 Decoders and Encoders

FIGURE 9-14
A 4-to-10 Line Decoder

(a) Logic diagram

(b) Truth Table

9.4 Decoders and Encoders

FIGURE 9-15
Realization of a Multiple-Output Circuit Using a Decoder

a →

b →

c →

d →

4-to-10 Line Decoder

\[ f_1 = (m_1' m_2' m_3')' \]

\[ f_2 = (m_4' m_5' m_6')' \]

11.5 R-S Flip-Flop

FIGURE 11-18
S-R Flip-Flop

Operation summary:
- No state change
- \( S = R = 0 \)
- \( S = 1, R = 0 \) (set \( Q \) to 1 after active \( \text{Clk} \) edge)
- \( S = 0, R = 1 \) (reset \( Q \) to 0 after active \( \text{Clk} \) edge)
- \( S = R = 1 \) (not allowed)

FIGURE 11-19
S-R Flip-Flop Implementation and Timing

(a) Implementation with two latches

(b) Timing analysis
11.6 J-K Flip-Flop

Figure 11-20
J-K Flip-Flop
(Q Changes on the Rising Edge)

(a) J-K flip-flop

(b) Truth table and characteristic equation

Figure 11-22
T Flip-Flop

(a) T Flip-Flop

(b) Timing Diagram for T Flip-Flop (Falling-Edge Trigger)