QM Logic Minimization
Outline

• Study of Components
  – Logic Implementation Using MUX & Decoder
    – 4 Bit Adder, BCD adder
• Quine-McCluskey (QM) Logic Minimization
• Examples
• Writing C/C++ program for QM Method
Decoders

• Internal design
  – AND gate for each output to detect input combination

• Decoder with enable e
  – Outputs all 0 if e=0, Regular behavior if e=1

• n-input decoder: $2^n$ outputs

Covers All Minterms
Boolean Function Implementation using Decoders

• Using a n-to-2n decoder and OR gates any functions of n variables can be implemented.

• Example:

  \[ S(x,y,z)= \Sigma(1,2,4,7) , \quad C(x,y,z)=\Sigma(3,5,6,7) \]

• Functions S and C can be implemented using a 3-to-8 decoder and two 4-input OR gates
Mux

4x1 mux

Covers All Minterms
Implementing logic Function using MUX

\[ F(A, B) = \sum m(0, 2) \]

4x1 mux
Adding Two One-bit Operands

• One-bit Half Adder:
  \[
  \begin{array}{c|c|c}
  A & B & Sum \\ 
  \hline
  0 & 0 & 0 \\
  0 & 1 & 1 \\
  1 & 0 & 1 \\
  1 & 1 & 0 \\
  \end{array}
  \]
  \[\text{Sum} = A \oplus B\]
  \[\text{Cout} = A \cdot B\]

• One-bit Full Adder:
  \[
  \begin{array}{c|c|c|c|c}
  C_{in} & A & B & Sum & Cout \\ 
  \hline
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 1 & 0 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 1 & 1 & 0 & 1 \\
  1 & 0 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 & 1 \\
  1 & 1 & 0 & 0 & 1 \\
  1 & 1 & 1 & 1 & 1 \\
  \end{array}
  \]
  \[\text{Sum} = A \oplus B \oplus C_{in}\]
  \[\text{Cout} = A \cdot B + B \cdot C_{in} + A \cdot C_{in}\]
N-Bit Ripple-Carry Adder: Series of FA Cells

- To add two n-bit numbers

- Adder delay = $T_c \times n$
  - $T_c = (C_{in} \text{ to } C_{out} \text{ delay})$ of a FA
4 bit Binary Adder

A3  B3  A2  B2  A1  B1  A0  B0
  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
 FA  FA  FA  FA  Ci
  ↓  ↓  ↓  ↓  ↓
 S2  S2  S1  S0
  ↓  ↓  ↓  ↓
B3  B2  B1  B0  A3  A2  A1  A0  Co
  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
  ↓  ↓  ↓  ↓
C0  S3  S2  S1  S0
  ↓  ↓  ↓  ↓  ↓
  ↓  ↓
  4 Bit Adder
Binary Adder (Two Level)

- Treat as 9 input & 5 output functions
- Generate Truth Table for each outputs
- Solve each function using KMAP/QM Method
- Only Two Level: No carry Propagation

![Adder Diagram]
Decimal Adder

- Decimal numbers are represented with BCD code.
- When two BCD digits A and B are added
  - if $A+B<10$ result is a valid BCD digit
  - if $A+B>9$ result will not be valid BCD digit. It must be corrected by adding 6 to the result
- If $A+B>9$ add 6 to solve this issue
Decimal Adder

K 4 bit binary adder
Z₈ Z₄ Z₂ Z₁

Output

Output Carry

C = K + Z₈Z₄ + Z₈Z₂
If (C) add 6

4 bit binary adder
BCD Adder (Two Level)

• Treat as 9 input & 5 output functions
• Generate Truth Table for each outputs
• Solve each function using KMAP/QM Method
• Only Two Level: No carry Propagation
**Quine-McCluskey Method for Minimization**

- KMAP methods was practical for at most 6 variable functions
- Larger number of variables: need method that can be applied to computer based minimization
- **Quine-McCluskey** method

- For example:

\[ \sum m(0,1,2,3,5,7,13,15) \]
QM Method

• Phase I: finding Pis
  – Tabular methods: Grouping and combining

• Phase II: Covers minimal PIs
QM Method

• Minterms that differ in one variable’s value can be combined.
• Thus we list our minterms so that they are in groups with each group having the same number of 1s.
• So the first step is ordering the minterms according to their number of 1s (0-cube list)
• only minterms residing in adjacent groups have the chance to be combined.):
QM Method

$$\sum \ m (0, 1, 2, 3, 5, 7, 13, 15)$$

<table>
<thead>
<tr>
<th>0_Cube</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>0010</td>
</tr>
<tr>
<td>2</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td>0101</td>
</tr>
<tr>
<td>3</td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td>1101</td>
</tr>
<tr>
<td>4</td>
<td>1111</td>
</tr>
</tbody>
</table>
QM Method: Combining Adjacent

- Compare minterms of a group with those of an adjacent one to form 1-cube list.
- When doing the combining, we put checkmark alongside the minterms in the 0-cube list that have been combined.
### QM Method

<table>
<thead>
<tr>
<th>0_Cube</th>
<th>1_Cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000X</td>
</tr>
<tr>
<td>0000</td>
<td>000X0</td>
</tr>
<tr>
<td>0010</td>
<td>001X</td>
</tr>
<tr>
<td>0011</td>
<td>010X</td>
</tr>
<tr>
<td>0101</td>
<td>011X</td>
</tr>
<tr>
<td>0111</td>
<td>101X</td>
</tr>
<tr>
<td>1101</td>
<td>111X</td>
</tr>
<tr>
<td>1111</td>
<td>X11X</td>
</tr>
</tbody>
</table>

Clues: 0000 0010 0011 0101 0111 1011 1111
QM Method: Combining Adjacent

• Do same combination of comparing adjacent group minterms
  – To form 2-cubes, 3-cubes and so on.

• Only minterms of adjacent groups have the chance of being combined
  – Which have an X in the same position.
QM Method

1_Cube

<table>
<thead>
<tr>
<th></th>
<th>000X</th>
<th>00X0</th>
<th>00X1</th>
<th>0X01</th>
<th>001X</th>
<th>01X1</th>
<th>X101</th>
<th>X111</th>
<th>11X1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2_Cube

<table>
<thead>
<tr>
<th></th>
<th>00XX</th>
<th>0XX1</th>
<th>X1X1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Indicates a valid output.
Q-M Method: Cover PIs

• PIs: terms left without checkmarks.

• After identifying our PIs, we list them against the minterms needed to be covered

\[ \sum m(0, 1, 2, 3, 5, 7, 13, 15) \]
QM Method : Covers

• To find a minimal cover, we first need to find essential PIs
• To do this we need to find columns that only have one checkmark in them, the according row will thus show the essential PI.
• After identifying essential PIs, that are necessarily part of the cover, we cover any remaining minterms using a minimal set of PIs.

In this example: \( F(a, b, c, d) = \overline{ab} + bd \)
Thanks