Implicants - any valid 1-sum group

Prime Implicant - max-size implicant

Essential Prime Implicant - P.I. that is essential for a min solution

Minimum Solution:
- always contains all EPI's
- may contain some P.I.'s (not EPI)
- never contains any plain implicants (not P.I. or EPI)

To find min soln:
1) Include all EPIs
2) If necessary, add P.I.s until all minterms are covered.
   Cover X's only if helpful.

Implicants: all groups

PI's: \( AB', AC'D, BC'D, B'D' \)

EPI's: \( AB', BC'D, B'D' \)

\[ m_9 \rightarrow AB' \]
\[ m_7 \rightarrow BCD \]
\[ m_5 \rightarrow B'D' \]
To find EPIs:
1) Choose a minterm (1 on K-map) and look at all P.I.s that cover it.
2) If only 1 P.I., → that one is essential.
   If > 1, P.I., → learn nothing.
3) Repeat for all minterms.

Adders (Unit 4)

Half Adder
\[ \begin{array}{c|c|c}
a & b & s \\
\hline
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array} \]

Full Adder
\[ \begin{array}{c|c|c|c|c}
\text{a} & \text{b} & \text{c} & \text{car} & \text{sum} \\
\hline
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 \\
\end{array} \]

\[ \text{sum} = a \oplus b \oplus c \]
\[ \text{car} = ab + bc + ac \]
Ripple-Carry Adder
- simplest
- slowest

MUX

<table>
<thead>
<tr>
<th>c</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
4:1 multiplexer

2^N inputs, N select bits

Quine-McCluskey (Unit 6)

- Works with many inputs
- Easy to program

1) Find all P.E.
2) Find all E, P.E.
3) Build minterm solution

\[ E = \Sigma m(1, 2, 5, 6, 7) \]

\[ = A'B'C + A'BC' + AB'C + ABC' + ABC \]
List groups by # of 1s in minterm

Compare minterm pairs in adjacent groups

Copy term pairs that differ in one variable to next column and check "\( \checkmark \)" matched terms

Repeat until no more matches
Multi-level Circuits

Level of a circuit = max. # of gates in series between an input and output

SOP: 2-level: AND - OR
POS: 2-level: OR - AND

Ex. \( Z = (ABC + EF + G) \cdot H + ABCD \)

Why do levels matter?
- # of gates
- Fanin ( # of inputs to a gate)
- Fanout ( # of outputs of a gate)
- Delay

NANDs and NORs

Min terms and max terms \( \rightarrow \) AND, OR, NOT are "functionally complete"

AND?: Can't make an inverter \( \rightarrow \) Not functionally complete
\[ A \cdot A \]

OR?: Can't make an inverter \( \rightarrow \) " "
\[ A + A \]
INV?

Don't make AND, OR → Not finite, complete

NAND?

NAND

NOT

\[ \overline{a \cdot \overline{a}} = a \]

AND

OR

NAND is finite, complete!

NOR