- A function is in the Reed-Muller canonical form when it is expressed as an eXclusive-OR of the products of uncomplemented variables.

- XOR has symbol $\oplus$

- $A \oplus B$ is either A OR B but NOT A ABD NOT B

Rules for Boolean algebraic operations using the eXclusive-OR operator, $\oplus$

$$\begin{aligned} A \oplus A &= 0 \\ A \oplus \overline{A} &= 1 \\ 1 \oplus A &= \overline{A} \\ A + B &= A \oplus B \oplus AB = A \oplus \overline{A}B = B \oplus \overline{B}A \\ A(B \oplus C) &= AB \oplus AC \end{aligned}$$

Functions are usually specified in truth table or Sum of Products form

$$\begin{aligned} f(ABCD) &= \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}C\overline{D} + ABCD \\ &= \Sigma m(5,7,10,15) \end{aligned}$$

Only one row of truth table can be valid at one time
Therefore we can replace $+$ by $\oplus$ to get

$$f(ABCD) = \overline{A}B\overline{C}D \oplus \overline{A}BCD \oplus A\overline{B}C\overline{D} \oplus ABCD$$

But terms appear in complemented ($\overline{A}$) as well as uncomplemented form (A).

We simplify using

$$\overline{A} = 1 \oplus A$$

Use

$$\overline{A} = 1 \oplus A$$

to replace the complemented terms by un-complemented terms

$$\begin{aligned} f(ABCD) &= \overline{A}B\overline{C}D \oplus \overline{A}BCD \oplus A\overline{B}C\overline{D} \oplus ABCD \\ &= (1 \oplus A)B(1 \oplus C)D \oplus (1 \oplus A)BCD \\ &\quad \oplus A(1 \oplus B)C(1 \oplus D) \oplus ABCD \\ &= AC \oplus BD \oplus ABC \oplus ABD \oplus ACD \oplus 0 \end{aligned}$$

This is the Reed-Muller form AND/XOR.

Reverse conversion RM to SOP

All variables must appear in the SOP.

$$
\begin{aligned}
f(AB) &= A \oplus B \\
&= A \oplus AB \oplus AB \oplus B \\
&= A(1 \oplus B) \oplus B(1 \oplus A) \\
&= A\overline{B} \oplus B\overline{A} \\
&= A\overline{B} \oplus \overline{A}B \\
&= A\overline{B} + \overline{A}B \\
&= \Sigma m(1,2)
\end{aligned}
$$

Generalize SoP to RM conversion using coefficients

$$
\begin{aligned}
f(A) &= a_0\overline{A} + a_1 A \\
&= a_0\overline{A} \oplus a_1 A \\
&= a_0(1 \oplus A) \oplus a_1 A \\
&= a_0 \oplus a_0 A \oplus a_1 A \\
&= a_0 \oplus (a_0 + a_1) A \\
&= c_0 \oplus c_1 A \\
\text{where } c_0 &= a_0 \\
\text{and } c_1 &= a_0 + a_1
\end{aligned}
$$

or in matrix form $\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} . \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$

For the two variable function:

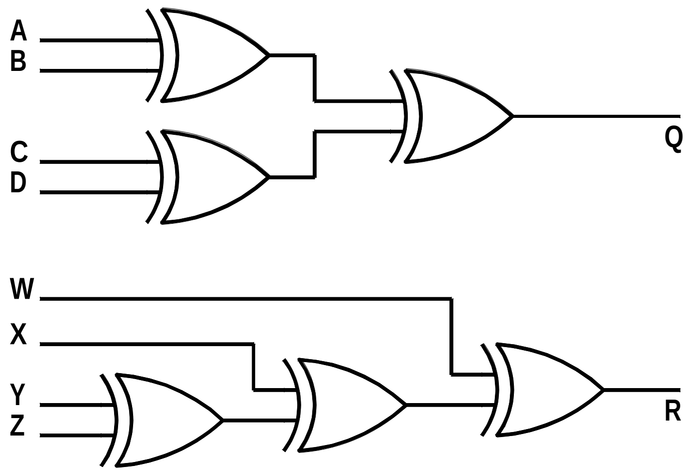
$$
\begin{aligned}
f(AB) &= a_0\overline{AB} + a_1\overline{A}B + a_2A\overline{B} + a_3AB \\
&= c_0 \oplus c_1B \oplus c_2A \oplus c_3AB
\end{aligned}
$$

where
$$
\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} . \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}
$$

or $\mathbf{c} = \mathbf{T}.\mathbf{a}$

In general the $\mathbf{T_n}$ matrix is formed recursively from the $\mathbf{T_{n-1}}$ matrix by:

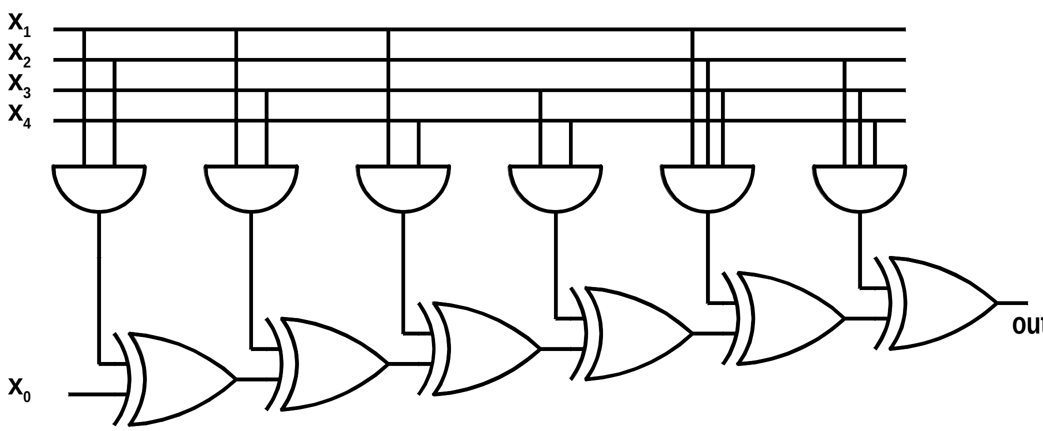
$$\mathbf{T_n} = \begin{pmatrix} \mathbf{T_{n-1}} & \mathbf{0} \\ \mathbf{T_{n-1}} & \mathbf{T_{n-1}} \end{pmatrix} \quad \text{and} \quad \mathbf{T_1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$



Circuit implementation of

$$A \oplus B \oplus C \oplus D$$

Only two inputs to XOR gate.
Binary tree structure and cascade structure.



Circuit which implements

$$f(x_{n-1}...x_0) = f(x_4...x_0)$$
$$= 1 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4$$
$$\oplus x_3x_4 \oplus x_1x_2x_3 \oplus x_2x_3x_4$$

employing a bus structure
$x_0 = 1$ and the other AND gate combinations are picked off the bus
combined into the sequential version of the XOR array.