- A function such as $Q = A \oplus B \oplus C \oplus D$ will only take logic 1 when an odd number of A, B, C and D have logic 1. The function therefore acts as a parity generator.

- An n input Sum-of-Products function requires 2^n sets of test inputs.
  An n input Reed-Muller circuit can be tested using $4+n+2n_e$ sets of test inputs

- Reed-Muller expressions are minimized by selecting either the uncomplemented or complemented form of each of the inputs so as to obtain a minimized expression.

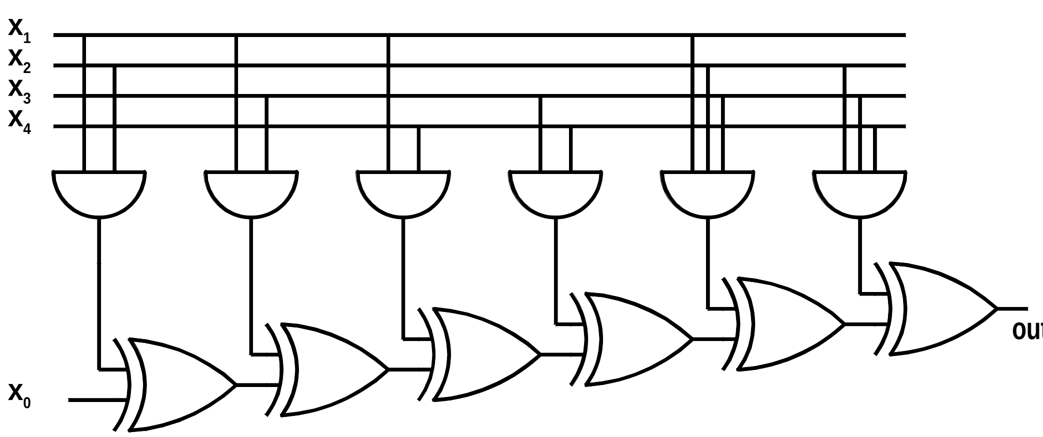
- The Polarity Vector is used to define which form of the input is to be used in the Reed-Muller expression.

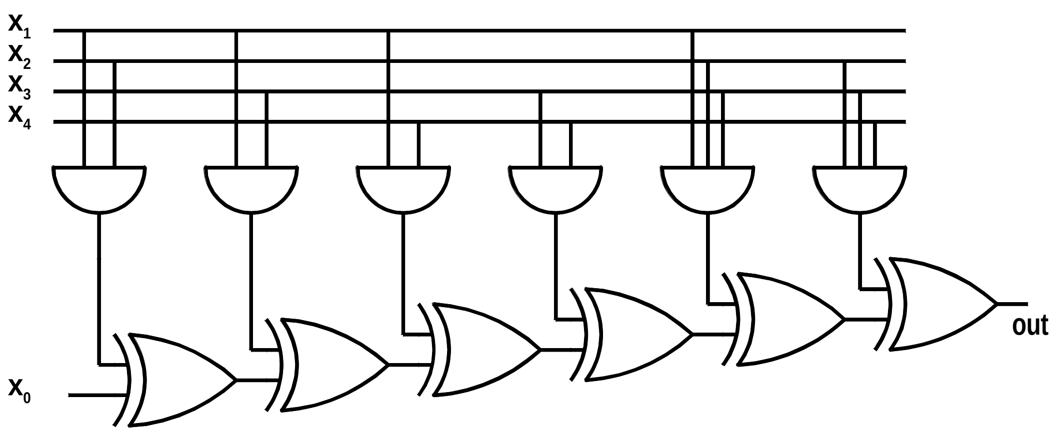
- The techniques for minimizing Reed-Muller expressions are the subject of ongoing research.

- A Sum of Products expression of n variables has 2^n rows in truth table.
- Therefore 2^n tests must be carried out to verify that system is operating correctly.
- Possible for prototype but not for production.
- Use batch statistics for quality control.
- Or change to Reed-Muller Logic.

$$f(x_4, x_3, x_2, x_1, x_0) = x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_3x_4 \oplus x_1x_2x_3 \oplus x_2x_3x_4 \oplus x_0$$

is implemented by

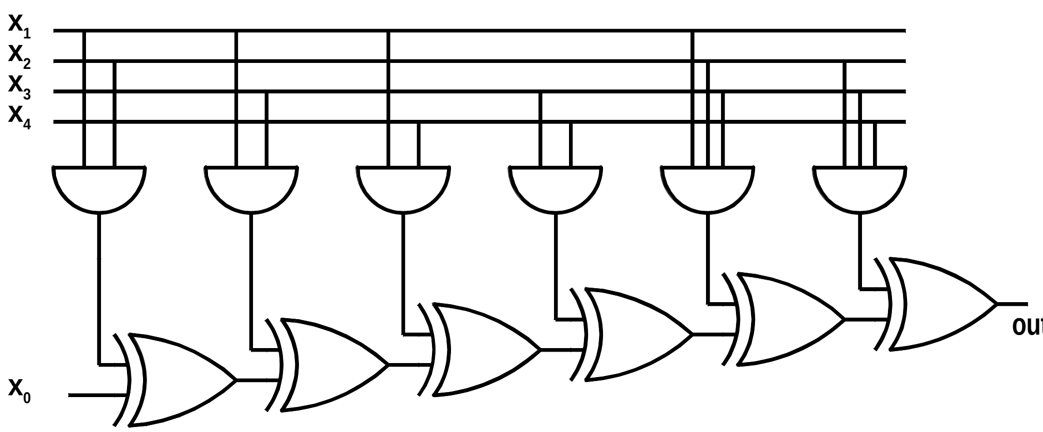




Test vector

$$\begin{matrix} & x_0 & x_1 & x_2 & x_3 & x_4 & \text{out} & \\ (& 0 & 1 & 1 & 1 & 1 & 0 &) \end{matrix}$$

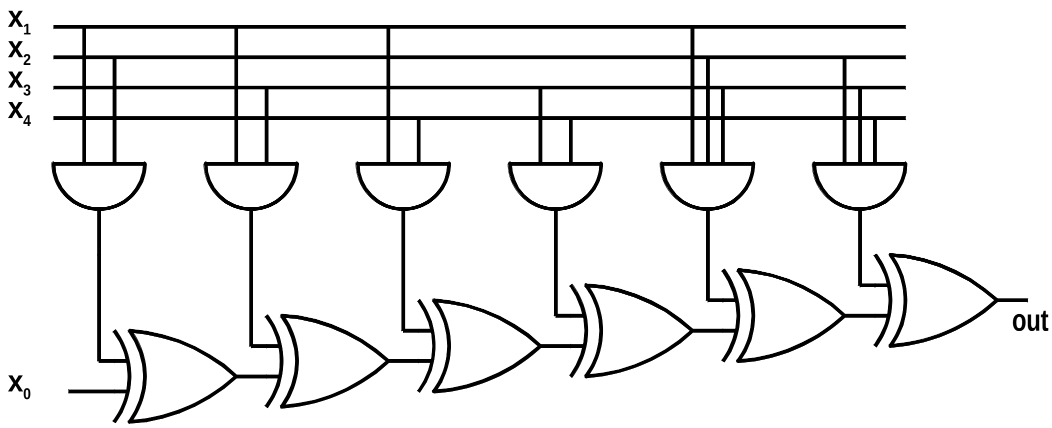
Verifies that all AND gates can give 1 out and
that XOR cascade operates



Test vector

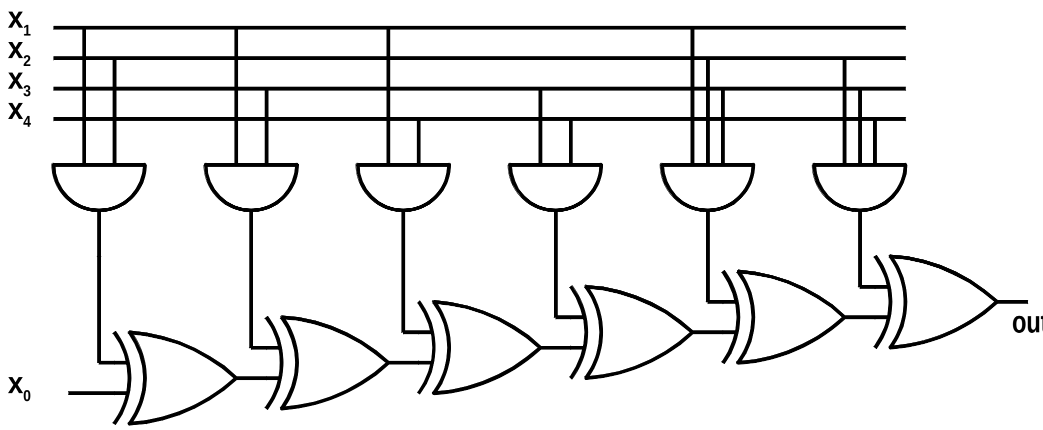
	x_0	x_1	x_2	x_3	x_4	out	
(	0	0	0	0	0	0	)

Verifies that all AND gates can give 0 out and
that XOR cascade operates



$$\mathbf{F_C} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

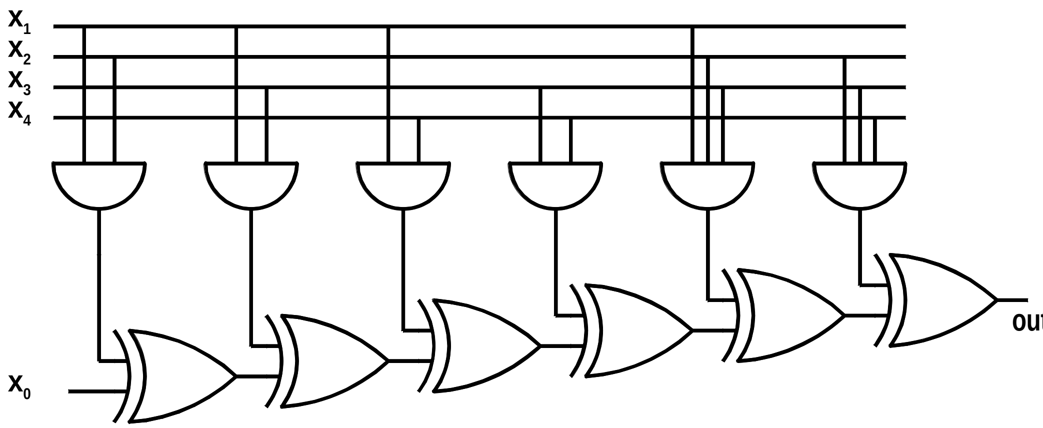
Tests all AND gates and ALL XOR gates.
No Stuck-at-0 or Stuck-at-1 gates.



Are all AND gates connected to inputs?

$$\mathbf{F_A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Faulty input line connected to EVEN number of AND gates not detected in this test.



Test for faults on input lines connected to EVEN number of AND gates with

$$\mathbf{F_{E1}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Similar test matrices are needed for each of the other inputs which also appear in an even number of product terms, so if there are n_e such input terms then there will be $2n_e$

The total number of tests which have to be carried out on a Reed-Muller AND/XOR type of gate array in order to test for correct functioning is therefore $4 + n + 2n_e$ with the one restriction that only single s-a-1 or s-a-0 type faults occur.

We therefore see that the number of tests required for testing AND/XOR gate arrays is a simple linear function of the number of primary inputs whereas the number on tests required to test an AND/OR Boolean type circuit is of the order of 2^n.

- We only used uncomplemented forms x_1, A etc.

- It is possible to use complemented forms $\overline{x_1}$, $\overline{A}$, etc.

- Keep track of which Polarity is being used with Polarity Vector.

- A, B, C, D has Polarity Vctor $\mathbf{k} = (1, 1, 1, 1)$

- $A, \overline{B}, \overline{C}, D$ has Polarity vector $\mathbf{k} = (1, 0, 0, 1)$

To convert expression from form having Polarity vector $\mathbf{k} = (1, 1, 1, 1)$

to form having Polarity vector $\mathbf{k} = (1, 0, 0, 1)$

use the substitutions $x_1 = 1 \oplus \overline{x_1}$
and $x_2 = 1 \oplus \overline{x_2}$
and multiply out

For example a typical conversion to form having polarity vector $\mathbf{k} = (1, 0, 0, 1)$ might be:

$$
\begin{aligned}
& f(x_4, x_3, x_2, x_1) \\
= \; & x_0 \oplus x_1(1 \oplus \overline{x_2}) \oplus x_1(1 \oplus \overline{x_3}) \oplus x_1 x_4 \oplus (1 \oplus \overline{x_3}) x_4 \\
& \oplus x_1(1 \oplus \overline{x_2})(1 \oplus \overline{x_3}) \oplus (1 \oplus \overline{x_2})(1 \oplus \overline{x_3}) x_4 \\
= \; & x_0 \oplus x_1 \oplus x_1 \overline{x_2} \oplus x_1 \oplus x_1 \overline{x_3} \oplus x_1 x_4 \oplus x_4 \oplus \overline{x_3} x_4 \\
& \oplus x_1 \oplus x_1 \overline{x_2} \oplus x_1 \overline{x_3} \oplus x_1 \overline{x_2 x_3} \oplus x_4 \oplus \overline{x_2} x_4 \oplus \overline{x_3} x_4 \oplus \\
= \; & x_0 \oplus (x_1 \oplus x_1 \oplus x_1) \oplus (x_4 \oplus x_4) \oplus (x_1 \overline{x_2} \oplus x_1 \overline{x_2}) \oplus \\
& \oplus x_1 \overline{x_3}) \oplus x_1 x_4 \oplus \overline{x_2} x_4 \oplus (\overline{x_3} x_4 \oplus \overline{x_3} x_4) \oplus x_1 \overline{x_2 x_3} \oplus \\
= \; & x_0 \oplus x_1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus x_1 x_4 \oplus \overline{x_2} x_4 \oplus 0 \oplus x_1 \overline{x_2 x} \\
= \; & x_0 \oplus x_1 \oplus 0 \oplus x_1 x_4 \oplus \overline{x_2} x_4 \oplus x_1 \overline{x_2 x_3} \oplus \overline{x_2 x_3} x_4
\end{aligned}
$$