Introduction to Programmable Logic

Any combinational logic function can be expressed as:

1) Boolean Equations
   \[ X = A \oplus B \]
   \[ Y = A \cdot (B \cdot C) \]

2) Truth Tables/Kmaps
   
<table>
<thead>
<tr>
<th>ABC</th>
<th>XY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0</td>
</tr>
</tbody>
</table>

3) Sum-of-Products/Product-of-Sums
   \[ X = AB' + A'B \]
   \[ Y = AB' + AC' \]

4) Other representations as well

The first three representations above are important for implementing programmable logic which use:

1) ROMs – Read Only Memories
   We can program a ROM (8 words by 2 bit/word), then use the address lines as the 3 input signals (A,B,C) and the ROM outputs as the output signals (X,Y)

2) RAMs – Random Access Memories
   We can write the truth table into a RAM (8 words by 2 bit/word), then disable the WE and use the address lines as the 3 input signals (A,B,C) and the RAM outputs as the output signals (X,Y) – this is the now the same thing as the ROM except we can re-program the logic function by re-writing the RAM

3) PLAs – Programmable Logic Arrays
   In the minimized truth table, there are only 3 words producing logic 1 for the output signals and the PLA allows implementing only those 3 words (and not the other 5) \(\Rightarrow\) smaller than ROM
Any sum-of-products can be implemented as a 2-level AND-OR or NAND-NAND logic function (assume 2-rail inputs: bit & bit-bar).

Any sum-of-products can also be implemented as a 2-level NOR-NOR logic function (assume 2-rail inputs) if we invert the inputs and the output.

\[ X = AB' + A'B \]
\[ Y = AB' + AC' \]

Note: \( AB' \) is a shared product term in all three implementations.
Programmable Logic Arrays (PLAs) take advantage of the NOR-NOR implementation of logic functions and the large fan-in limit of NMOS NOR gates.
Introduction to Programmable Logic

Programming technologies for Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs):

1. Fuse/Anti-fuse
2. Floating Gate Technology
3. RAM

Consider a PLA cross-point

Floating gate technology includes:

1. UVEPROM (UV Erasable Programmable ROM)
2. EEPROM (Electrically Erasable Programmable ROM)
3. Flash Memory (like in memory keys)

Important Programming Technology Terms

OTP (one-time programmable) – characteristic of fuse/anti-fuse
ISP (in-system programmable) – some floating gate technologies are ISP
ISR (in-system re-programmable) – characteristic of RAM

OTP and floating gate technologies are non-volatile and logic will retain its configuration when powered down.
RAM is volatile and must be configured when power is turned on.