To implement any Boolean expression in circuits, we want:

inputs \{ \ldots \} \rightarrow \text{outputs} \rightarrow 1 \text{ chip}

Want outputs to be any function of any inputs (could be a very large circuit!)

Instead, settle for restricted set of expressions:

- Realize Sum-of-Products expressions for each output
- Programmable AND inputs
- Programmable OR inputs
- Limited \# of inputs
- Limited \# of outputs per chip
- Combine several chips to create multi-level circuits for more complex functions.

Programmable Logic Arrays (PLA)
Ex: \[ X = ABC + DE \]
\[ Y = AB + CD + EF \]
\[ Z = B'c' + ABC + CD \]

How are the connections made?

**Custom chip**: Add or leave out diode, transistor, ... at correct intersections

**One-time Programmable**: Blow "fuse" (create open ckt) or "anti-fuse" (creates short ckt) at manufacturing time or first programming

**Programmable (Temporary or Volatile)**: Loses configuration when power turned off (configuration stored in memory)

**Programmable (Permanent until Erased or Nonvolatile)**: Configuration written electrically, erased electrically or with U.V. light.
Possible programmable connections:

- Diode
- FET
- Programmable

**Alternative Connections**

"Wired AND" 

\[ V_{cc} (5V) \]

- \( A = 1 \rightarrow \text{Diodes off, } Z = 1 \)
- \( B = 1 \)

- \( A = 0, \text{ Diode on, } Z = 0 \)
- \( B = X \)

"Wired OR"

- \( A = 0 \rightarrow \text{Diodes off, } Z = 0 \)
- \( B = 0 \)

- \( A = 1 \rightarrow \text{Diode on, } Z = 1 \)
- \( B = X \) pulls \( Z \) high, \( Z = 1 \)
Programmable Array Logic

Similar concept to PLA but the OR array is fixed:

\[ A \quad B \quad C \quad D \cdots \]

Drawback: Cannot share product terms among outputs as with PLAs.

LAB 4 PAL (Generic Array Logic) 22V10
- 11 inputs
- 10 outputs
- 8-16 ANDs per OR output