Transistors and Logic Circuits
Transistor

Control high allows current to flow -- switch is closed (on)

Control low stops current flow switch is open (off)
NOT Gate One transistor

V (high voltage)

In = high, switch is closed so current flows to ground Out is low.

In = low, switch is open so current flows to Out Out is high.
NOR Gate  Two transistors

V (high voltage)

In 1

In 2

Out

In 1 = 1, Out = 0
In 2 = 1, Out = 0
In 1 = 0
In 2 = 0, Out = 1
NAND Gate   Two transistors

V (high voltage)

In 1 = 1,
In 2 = 1,  Out = 0

In 1 = 1
In 2 = 0, Out = 1
AND Gate   Three transistors

V (high voltage)
Logic Gates

AND Gate

OR Gate

XOR Gate

NOT Gate
Logic Circuit -- 4 input Multiplexor

0
1
2
3
In

Control 1 0

Out
Logic Circuit Puzzle 1

Input
Binary Numbers A, B

8 bit Comparator
Output 1 if A = B
Otherwise 0
Logic Circuit Puzzle 2

3 bit Decoder
Select Output Line

In 2
In 1
In 0

D0
D1
D2
D3
D4
D5
D6
D7
Programmable Logic Array

- Any Logic Truth Table can be implemented
- Uses block of AND gates followed by block of OR gates
- Programmable
  - once
  - many times
- Used for implementing different circuits
## Truth Table to Normal Form

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>A and B and ~C</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>A and ~B and C</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>~A and B and C</td>
</tr>
<tr>
<td>0</td>
<td>1</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>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[(A \text{ and } B \text{ and } C) \text{ or } (A \text{ and } B \text{ and } \sim C) \text{ or } (A \text{ and } \sim B \text{ and } C) \text{ or } (\sim A \text{ and } B \text{ and } C)\]
PLA

Input

AND Gates

OR Gates

Output
What is Out 1?
## Normal Form to Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
PLA, Alternate Representation

AND Block uses DeMorgan Equivalence

A and B = not (not A or not B)

OR Block uses direct or
PLA, Alternate Representation

Unused

Unused

Outputs

0
1
2
3
PLA, Alternate Representation

Burned out to disconnect
**PLA  "Don't Cares"**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

\(X = \text{Don't Care}\)
PLA "Don't Cares"

Reduce number of PLA lines used for expression