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

V (high voltage)

In 1

In 2

Out
Logic Gates

- **AND Gate**
  - Input: 0, 1
  - Output: AND

- **OR Gate**
  - Input: 0, 1
  - Output: OR

- **XOR Gate**
  - Input: 0, 1
  - Output: XOR

- **NOT Gate**
  - Input: 1
  - Output: NOT
Logic Circuit -- 4 input Multiplexor

Control 0 1 0

In 0 1 2 3

Out

0 1
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>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</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>1</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>

\[(A \text{ and } B \text{ and } C) \text{ or } (A \text{ and } B \text{ and } \neg C) \text{ or } (A \text{ and } \neg B \text{ and } C) \text{ or } (\neg A \text{ and } B \text{ and } C)\]
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>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- \((A \land B \land C)\) or \((A \land \neg B \land \neg C)\) or \((\neg A \land B \land \neg C)\) or \((\neg A \land \neg B \land C)\)

**Odd Parity**
PLA, Alternate Representation

AND Block uses DeMorgan Equivalence

\[ A \text{ and } B = \neg (\neg A \text{ or } \neg B) \]

OR Block uses direct or
PLA, Alternate Representation

A
B
C

Outputs
0
1
2
3

Unused

Unused
PLA, Alternate Representation

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

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

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

Reduce number of PLA lines used for expression.
The truth about transistors use power whenever connected to ground

Not Gate

V (high voltage)

In

Out
NOR gate

V (high voltage)

In 1

In 2

Out
CMOS

Complementary Metal Oxide Semiconductor

Replaces single transistors with a pair, one p-type (like we used), one n-type (current passes when control is low)

Set up so that in steady state, no current flows to ground

Only uses power when switching
CMOS circuits use far less power – the current main constraint on the development of faster cpus.

Since 1976 the better power saving technology

Most processors today use this technology