Outline

- What is a Multilevel Gate Network?
  - Example
- New gates: Universal gates
  - NAND
  - NOR
- Functionally complete set of gates
- Multilevel NAND circuits
- Multilevel NOR circuits
What is a Multilevel Gate Network?

- **# of levels of gates:** maximum # of gates cascaded in series between an input and output (inverters excluded).
- **# levels of gates is proportional to total propagation delay** through logic.
- Decreasing the # of levels can decrease / increase the # of gates and inputs

Ex: \( O = (ab + cd)(ef + gh) \)

Number of levels of gates: 3

Ex: \( O = (ab + cd)(ef + gh) = abef + abgh + cdef + cdgh \)

Number of levels of gates: 2
Example – AND-OR Network

- \( f(a,b,c,d) = \sum(1,5,6,10,13,14) \)

- \( f = a'c'd + bcd' + b'c'd + acd' \)

Network

<table>
<thead>
<tr>
<th>Network</th>
<th>#level</th>
<th>#gates</th>
<th>#inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND-OR</td>
<td>2</td>
<td>5</td>
<td>16</td>
</tr>
</tbody>
</table>
OR-AND-OR Network

- \[ f = a'c'd + bcd' + b'c'd + acd' \]
  \[ = (a' + b')c'd + (a + b)cd' \]

<table>
<thead>
<tr>
<th>Network</th>
<th>#level</th>
<th>#gates</th>
<th>#inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND-OR</td>
<td>2</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>OR-AND-OR</td>
<td>3</td>
<td>5</td>
<td>12</td>
</tr>
</tbody>
</table>
OR-AND Network

- The product of sums expression of \( f \) is obtained using the 0’s in the karnaugh map:

- \( f' = c'd' + cd + ab'd + a'b' d' \)

\[ f = (c'd' + cd + ab'd + a'b' d')' \]

DeMorgan’s

\[ f = (c+d)(c'+d')(a'+b+d')(a+b+d) \]

<table>
<thead>
<tr>
<th>Network</th>
<th>#level</th>
<th>#gates</th>
<th>#inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND-OR</td>
<td>2</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>OR-AND-OR</td>
<td>3</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>OR-AND</td>
<td>2</td>
<td>5</td>
<td>14</td>
</tr>
</tbody>
</table>
\[ f = (c+d)(c'+d')(a+b+d)(a'+b+d') \]
\[ = (cd'+c'd)(ab+ad'+a'b+b+bd'+a'd+bd) \]
\[ = (cd'+c'd)(b+ad'+a'd) \]

<table>
<thead>
<tr>
<th>Network</th>
<th>#level</th>
<th>#gates</th>
<th>#inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND-OR</td>
<td>2</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>OR-AND-OR</td>
<td>3</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>OR-AND</td>
<td>2</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>AND-OR-AND</td>
<td>3</td>
<td>7</td>
<td>15</td>
</tr>
</tbody>
</table>
Outline

- What is a Multilevel Gate Network?
  - Example
- New gates: Universal gates
  - NAND
  - NOR
- Functionally complete set of gates
- Multilevel NAND circuits
- Multilevel NOR circuits
Universal gate NAND

- Truth table \((O = (ab)')\)

\[
\begin{array}{ccc}
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>AND</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
\end{array}
\]

- Gate Symbol:
Universal gate NOR

- Truth table: \( O = (a+b)' \)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>OR</th>
<th>NOR (O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Gate Symbol:

```
\begin{circuitikz}
\draw (0,0) to [short] (1,0) to [nor, thick] (0,1) to [short] (0,0.5) node [above] {O};
\end{circuitikz}
```
A set of logic operations is said **functionally complete** if any Boolean function can be expressed in terms of this set of operations.

### Example of Functionally Complete Set

<table>
<thead>
<tr>
<th>AND, OR, NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND, NOT</td>
</tr>
<tr>
<td>NAND</td>
</tr>
</tbody>
</table>

Show why AND and NOT gates can realize an OR gate:

$$X + Y = (X + Y)' = (X'.Y')'$$
Multilevel NAND circuits

- How to implement NOT and AND, OR operations with NAND gates:
  - **NOT:** \( a' = (a')' \)
  - **AND:** \( a \cdot b = (a \cdot b)' \)
  - **OR:** \( a + b = (a + b)' = (a' \cdot b')' \)
  - **Equivalent Symbol:** \( (ab)' = (a' + b') \)
From AND-OR to NAND-NAND

1\textsuperscript{st} Method: DeMorgan’s and Boolean expression

- \( f = a + b c' + b' c d \)

- \( f = (a + b c' + b' c d)'' = (a'.(b c')'.(b' c d))'' \)

\[ f = a + b c' + b' c d \]
\[ f = (a + b c' + b' c d)'' = (a'.(b c')'.(b' c d))'' \]
From AND-OR to NAND network

2nd Method: Graphically

- \( f = a + bc' + b'cd \)

\( f = (a'(bc')'(b'cd'))' \)

- Graphical representation of the logic circuit.
From AND-OR to NAND-NAND

Example 2 (1/2): 1st Method

- $f = a'c'd + bcd' + b'c'd + acd'$

- $f = (a'c'd + bcd' + b'c'd + acd')''$
  $= ((a'c'd)'(bcd')'(b'c'd')(acd'))'$
From AND-OR to NAND-NAND
Example 2: 2\textsuperscript{nd} Method
Multilevel NOR circuits

How to implement NOT and AND, OR operations with NOR gates:

- **NOT:**
  \[ a \rightarrow a' \]

- **OR:**
  \[ a + b = (a + b)' \]

- **AND:**
  \[ ab = (ab)' = (a' + b')' \]

- Equivalent Symbol: \[ (a + b)' = a'b' \]
From OR-AND to NOR-NOR

Example

\[ f = (c+d)(c'+d')(a+b+d)(a'+b+d') \]

\[ f = ((c+d)(c'+d')(a+b+d)(a'+b+d'))'' = ((c+d)' + (c'+d')' + (a+b+d)' + (a'+b+d')')' \]
From OR-AND to NOR-NOR

Example