Digital Design LAB

Islamic University – Gaza
Engineering Faculty
Department of Computer Engineering
Fall 2012
ECOM 2112: Digital Design LAB
Eng: Ahmed M. Ayash

Experiment # 1

Basic Logical Functions and Gates

September 16, 2012
1. Objectives:
   • To study AND, OR, INV, NAND, NOR and X-OR gates.
   • To study the representation of these functions by truth tables, logic diagrams and Boolean algebra.

2. Theory:

   **Truth Table:** Representation of the output logic levels of a logic circuit for every possible combination of levels of the inputs. This is best done by means of a systematic tabulation.
   Number of possibility is $2^{(\text{NO. of input})}$

   **AND:** A multi-input circuit in which the output is 1 only if all inputs are 1.

<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

   In Boolean expression, $F = AB$

   **OR:** A multi-input circuit in which the output is 1 when any input is 1.

<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

   In Boolean expression, $F = A+B$
**INV:** The output is 0 when the input is 1, and the output is 1 when the input is 0.

```
<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

In Boolean expression, F = \( \overline{A} \)

**NAND:** AND followed by INVERT.

```
<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

In Boolean expression, F = \( \overline{A \cdot B} \)

**NOR:** OR followed by INVERT.

```
<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

In Boolean expression, F = \( \overline{A + B} \)
X-OR: The output of the Exclusive –OR gate, is 0 when it’s two inputs are the same and its output is 1 when its two inputs are different.

![XOR gate diagram]

<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

When A=B, the output F=0.
When A≠B, the output F=1.
In Boolean expression, F= A ⊕ B
The output F of an XOR is equal to A ⊕ B = A'B + AB'. XOR gates can be constructed using NOT, AND, and OR gates.

NOTE: these truth tables are based on "positive" logic where positive voltage represents "1" and negative voltage represents "0". In case negative logic is used the output will be reversed.

XNOR: XNOR equal to XOR followed by NOT; (A XNOR B) = not (A XOR B).

![XNOR gate diagram]

<table>
<thead>
<tr>
<th>STATE</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3. Lab Work:

- **Requirements:**
  1. ICs 7400(NAND), 7402(NOR), 7404(NOT), 7408(AND), 7432(OR), 7486(XOR), LED.
  2. TTL switches for inputs and LED for the output.
- **Prelab:**
  1. Derive the truth table and draw a schematic diagram for each experimental part.
  2. Write out a logical expression for the output for each circuit connection in your experiment.

**Part 1: NOT Gate**
- Connect the following circuits and do the Prelab requirements.

![Diagram of NOT Gate](image)

**Part 2: AND Gate**
- Connect the following circuits and do the Prelab requirements.

![Diagram of AND Gate](image)

**Part 3: OR Gate**
- Connect the following circuits and do the Prelab requirements.

![Diagram of OR Gate](image)
Part 4: NAND Gate
- Connect the following circuits and do the Prelab requirements.

![NAND Gate Circuit](image)

Part 5: NOR Gate
- Connect the following circuits and do the Prelab requirements.

![NOR Gate Circuit](image)

Part 6: X-OR Gate
- Connect the following circuits and do the Prelab requirements.

![X-OR Gate Circuit](image)

Part 7: XNOR Gate
- Connect x-nor gate using 7404(not) and 7404 (x-or) gate.

![XNOR Gate Circuit](image)
Part 8: DeMorgan's Theorem

A NAND gate is equivalent to an inversion followed by an OR

A NOR gate is equivalent to an inversion followed by an AND

a) Write down the expressions of Demorgan's low.

b) Use the truth table of NAND to verify that it is equivalent to (Negative-OR) and the truth table of the NOR gates to verify that it is equivalent to (Negative-AND).

4. Exercises

a) Design a 3-input NAND gate using 7400Ics (2-input NAND gate) only.

b) Build INV gate using a single 2-input NOR gate.

c) Build INV gate using a single 2-input NAND gate.