ELEC 1202 Digital Systems and Microprocessors

Combinatorial logic design

Jing Huey Khor
Combinational logic

Boolean Algebra is used to developed digital electronics.

Boolean algebra started in mid-19th century. Charles Dodgson (Lewis Carroll) and George Boole where the major contributors to the early inception of Boolean algebra (algebra of logic).

The engineering application of Boolean algebra came much later (1940), when it was implemented using relays (switching devices) by Claude Shannon.

The switching device have progressively improved moving to valve circuits and eventually transistors (small switches).

When transistors start being integrated in a single chip initially as Small Scale Integration (SSI) then as the technology matured to Medium Scale Integration (MSI) to Large Scale Integration (LSI), Very Large Scale Integration (VLSI) and eventually Ultra Large Scale Integration (ULSI) the usefulness of Boolean algebra was fulfilled.
Combinatorial logic

a. Logical proposition – is a variable or a fact that can be either FALSE or TRUE. (it cannot be anything else!!!)

Example: My car will not start.

b. Logical statement – is a combination of logical propositions.

Example: I cannot use my car if the tyres are flat or I have no petrol in the tank.

Each of the proposition can have two values: FALSE or TRUE. The validity of the statement (S) depends of the validity (value) of propositions (P1, P2).

Therefore there will be four possible combination, hence the statement can be either FLASE or TRUE depending on the four possible combinations.
Combinatorial logic

Example: I cannot use my car if the tyres are flat or I have no petrol in the tank.

Proposition 1 $P_1$ if the tyres are flat.  
Proposition 2 $P_2$ I have no petrol in the tank.

The validity of the statement depends on the validity of the proposition.

2 Propositions each with 2 values = 4 combinations

1. Both propositions are FALSE
2. FALSE/TRUE
3. TRUE/FALSE
4. Both propositions are TRUE

There are no other possible combinations as long as the proposition can have either of two values FALSE or TRUE.
Combinatorial logic

Example: I cannot use my car if the tyres are flat or I have no petrol in the tank.

Truth table – summarises the validity statement

<table>
<thead>
<tr>
<th>Proposition 1 ( P_1 )</th>
<th>Proposition 2 ( P_2 )</th>
<th>Statement ( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE (F)</td>
<td>FALSE (F)</td>
<td>FALSE (F)</td>
</tr>
<tr>
<td>FLASE (F)</td>
<td>TRUE (T)</td>
<td>TRUE (T)</td>
</tr>
<tr>
<td>TRUE (T)</td>
<td>FALSE (F)</td>
<td>TRUE (T)</td>
</tr>
<tr>
<td>TRUE (T)</td>
<td>TRUE (T)</td>
<td>TRUE (T)</td>
</tr>
</tbody>
</table>

- \( S \) can drive my car.
- I have no petrol.
- I have flat tyres.
- I have flat tyres and no petrol.
Combinatorial logic – logical functions

Statements are Logical Functions. There are three main (most important) Logical Functions

<table>
<thead>
<tr>
<th>logical OR</th>
<th>logical AND</th>
<th>logical NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \lor B )</td>
<td>( A \land B )</td>
<td>( \neg A )</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Example: I cannot use my car if the tyres are flat or I have no petrol in the tank.

Example: I will go home if I have finished my work and my car is working.