Theory Of Computer Arithmetic

Disclaimer

This follows the lessons from Mr. Oudjida given in course, major additions, changes or rearrangements can be applied on the course for the sake of clarity, better explainations or just subjective opinions, written by HADIOUCHE Azouaou.

USE THIS AT YOUR OWN RISK.

Chapter 1

Classical logic

The most basic part of executing a computation on a machine is to describe the most basic information, which is true/false, and then compose them into a statement or a proposition.

informally, given a sentence, it is said to be a *statement* if
- its declaritive, either affirmative or negative.
- its possible truth values are true or false.
- its verifiable in reality.

on those statements, we have some rules to give them truth values
- law of identity: $A = A$ is a true statement.
- law of non-contradiction: $\neg(A \wedge \neg A)$ is false statement.
- law of excluded middle: either $\neg A$ or A is true statement.

1.1. Transistors

One of the biggest advancements in our modern world is the creation of a transistor, in principle the idea is simple, a transistor is simply an electrically affected button. It is represented as follows in electrical circuits

multiple basic operations can be made using these transistors, from a baasic memory to some logic gates and then bigger sequential circuits.

- AND OR NOT
- NOR XOR
- Data Latch

Note: *Some examples will be added soon.*

1.2. Boolean Algebra

Let $\mathbb{B} := \{0, 1\}$ denote the set of boolean values, which can be represented too with true/false. Any variable ta

Definition 1.2.2 (Boolean Variable): *Let x be a variable, x is said to be a* boolean variable *if it can assume values in $\mathbb{B}$. Let $f : \mathbb{B}^n \to \mathbb{B}$ a map, f is called a* boolean function.

Definition 1.2.3 (Boolean Operations): *Let x, y be two boolean variables, we define the operations $+, \cdot, \overline{}$ to be the logical or, and, not respectively, which have the following truth tables.*

y	x	$\overline{x}$	$x + y$	$x \cdot y$
0	0	1	0	0
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

there are some other operations that are as follows
- $x \mid y = \overline{x \cdot y}$.
- $x \otimes y = x \cdot \overline{y} + \overline{x} \cdot y$.
- $x \Rightarrow y = \overline{x} + y$.

Proposition 1.2.4 (Boolean Identites):
- $\overline{\overline{x}} = x$.
- $x + x = x, x \cdot x = x$.
- $x + 0 = x, x \cdot 1 = x$.
- $x + 1 = 1, x \cdot 0 = 0$.
- $x + y = y + x, x \cdot y = y \cdot x$.
- $x + (y + z) = (x + y) + z, x \cdot (y \cdot z) = (x \cdot y) \cdot z$.
- $\overline{x + y} = \overline{x} \cdot \overline{y}, \overline{x \cdot y} = \overline{x} + \overline{y}$.

Definition 1.2.5 (Duality): *Let* $f : \mathbb{B}^n \to \mathbb{B}$ *be a boolean function, we define the dual of* f *as the map* $(x_1, ..., x_n) \mapsto \overline{f(\overline{x}_1, ..., \overline{x}_n)}$. *We can obtain the dual of a function* f *by swapping* $+$ *with* $\cdot$, 0 *with* 1 *and keep the variables unchanged.*

Definition 1.2.6 (Conjunctive/Disjunctive Normal Form): *Let* $f : \mathbb{B}^n \to \mathbb{B}$ *be a boolean function,* $\{J_i\}_{i=1}^{n}$ *a set of subsets of* $[\![1, n]\!]$

- *CNF:* $f(x_1, ..., x_n) = \prod_{i=1}^{k} \sum_{j \in J_i} y_j$.
- *DNF:* $f(x_1, ..., x_n) = \sum_{i=1}^{k} \prod_{j \in J_i} y_j$.

with $k \in \mathbb{N}$, $y_i = x_i$ *or* $y_i = \overline{x_i}$.

Proposition 1.2.7:

- $\{+, \overline{}\}$, $\{\cdot, \overline{}\}$, $\{|\}$ *are all a complete set of connectives, that is, any boolean function can be written using only one of those sets.*
- *Any boolean function can be written in the CNF or DNF.*