

# Chapter 1: From Zero to One

Note of Digital Design and Computer Architecture

2023 Oct 9 - 2023 Oct 10

Nemo

## Contents

1 The Game Plan .....	2
2 The Art of Managing Complexity .....	2
2.1 Abstraction .....	2
2.2 Discipline .....	2
2.3 The Three-Y's .....	2
3 The Digital Abstraction .....	2
<i>Definition 1: Amount of information</i> .....	3
4 Number Systems .....	3
4.1 Skipping a Few Topics ... ..	3
4.6 Signed Binary Numbers .....	3
4.6.1 Sign/Magnitude Numbers .....	3
4.6.2 Two's Complement Numbers .....	3
5 Logic Gates .....	3
<i>Definition 2: Truth table</i> .....	4
<i>Definition 3: Boolean equation</i> .....	4
5.1 Not Gate .....	4
5.2 Buffer .....	4
5.3 And Gate .....	4
5.4 Or Gate .....	4
5.5 Other Two-Input Gates .....	4
5.6 Multiple-Input Gates .....	4
6 Beneath the Digital Abstraction .....	4

# 1 The Game Plan

The book mainly focus on the design of digital systems. And a great advantage of digital systems is that it only involves 1's and 0's. At first buliding a microprocessor may be too complex to work on, so an important part of the book is managing complexity.

## 2 The Art of Managing Complexity

Morden digital systems commonly consist of millions or billions of transistors, which no human being can understand them simply by using equations or physics. An important approach to understand this is through managing complexity without getting mired in a morass of detail.

### 2.1 Abstraction

The critical technique for managing complexity is *abstraction*: hiding details when they are not important.

For example:

Application Software → Operating Systems → Architecture → Micro-Architecture  
→ Logic → Digital Circuits → Analog Circuits → Devices → Physics

The book covers from Architecture to digital circuits. It is good to know the levels both above and below the current abstract level you are working on.

### 2.2 Dicipline

*Dicipline* is the act of intentionally restricting your design choices so that you can work more productively at a higher level of abstraction.

e.g. :

- **Using interchangeable parts** : In the early stage of firearms, gunmakers make a entire gun once at a time, but then to improve efficiency, they begin to limit the components to standerized set and well-defined tolerance, which indeed worked well and made making a gun faster.

### 2.3 The Three-Y's

In addition to abstraction and dicipline, designers use three “-y's” to manage complexity: hierarchy, modularity, and regularity.

- *Hierarchy* involves dividing a system into modules, then further sub-dividing each of these modules until the pieces are easy to understand.
- *Modularity* states that the modules have well-defined functions and interfaces, so that they connect together easily without unanticipated side effects.
- *Regularity* seeks uniformity among the modules. Common modules are reused many times, reducing the number of distinct modules that must be designed.

## 3 The Digital Abstraction

Digital system use variables with a finit number of distinct values, contridict to analog systems. We maily focus on digital systems that use variables with 2 distinct values: 1 and 0, which are called binary variables.

**Definition 1: Amount of information**

The amount of information  $D$  in a discrete values variable with  $N$  distinct states is measured in units of **bits** as

$$D = \log_2 N \text{ bits}$$

George Boole developed a system of logic operating on binary variables that is known as *Boolean logic*. Each of Boole's variabel can be 'TRUE' or 'FALSE'. In the note I use '1', TRUE, and HIGH synonymously, and '0', FALSE, and LOW interchangeably.

## 4 Number Systems

### 4.1 Skipping a Few Topics ...

### 4.6 Signed Binary Numbers

Several schemes exist to represent signed binary numbers; the two most widely employed are called sign/magnitude and two's complement.

#### 4.6.1 Sign/Magnitude Numebrs

An  $N$ -bit sign/magnitude numebr uses the most significant bit as the sign and the remaining  $N-1$  bits as the magnitude. A sign bit of 0 indicates positive and a sign bit of 1 inditates negative.

e.g.  $-5_{10} = 1101_2$     $5_{10} = 0101_2$

However, here comes two shortcomings:

1. Ordinary binary addition does not work.
2. There are two representations for zero.

#### 4.6.2 Two's Complement Numbers

Two's complement numbers are identical to unsigned binary numbres except that the most significant bit position has a wegiht of  $-2^{N-1}$  instead of  $2^{N-1}$ . So positive numebrs have the most significant bit 0 and the negative numbers have 1 on the most significant bit, so the MSB can be seen as a sign bit just as that in the *Sign/Magnitude* scheme, but do note that the remaining bits are interpreted differently.

The sign of a two's complement number is reversed in a process called taking the two's complement, which consists of inverting all the bits and add 1 to the **LEAST significant bit**. It's useful to present the magnitude of a negative number.

The range of a  $N$ -bit two's complement number spans  $[-2^{N-1}, 2^{N-1} - 1]$ , which is not symmetric. This causes problems, because the smallest number it can represent  $100...00_2$  has no positive counterpart. The number is also called the *weird number*.

When an two's complement number is extended to more bits, its sign bit must be moved to the new MSB, which is called *sign extension*.

## 5 Logic Gates

Now we know that binary variables are used to represent information, but only the information itself is not enough to form a digital system. Here *Logic Gates* are simple circuits that take one or more binary inputs and produce a binary output. Typically, logic gates drawn as symbles take inputs on the left (or top) and produce output on the right (or buttom). Inputs are usually represented by letters at the beginning of the alphabet and use the letter "Y" as the output. A logic gate can be described with a *truth table* or a *Boolean equation*.

**Definition 2: Truth table**

A truth table is a table that lists inputs on the left and the corresponding output on the right.

**Definition 3: Boolean equation**

A mathematical expression using binary expression.

**5.1 Not Gate**

A	Y	$Y = \overline{A}$ The <i>NOT gate</i> , also called an <i>inverter</i> , “inverts” the input.
0	1	
1	0	

**5.2 Buffer**

A	Y	$Y = A$ The buffer simply copies the input to the output, so it work just as a wire in the view of logic. However, form the analog point view, it might have several desiable characteristics such as the ability to quicily send its output to multiple gates.
0	0	
1	1	

**5.3 And Gate**

A	B	Y	$Y = AB$ The AND gate only outputs TRUE when A and B are both TRUE.
0	0	0	
1	0	0	
0	1	0	
1	1	1	

**5.4 Or Gate**

A	B	Y	$Y = A + B$ The AND gate outputs TRUE when either A or B is TRUE. Note that the “+” symbol in the Boole equaiton is pronounced “or”, or the equaiton can also be written like $Y = A \cup B$
0	0	0	
1	0	1	
0	1	1	
1	1	1	

**5.5 Other Two-Input Gates**

XOR			NAND			NOR		
$Y = A \oplus B$			$Y = \overline{AB}$			$Y = \overline{A + B}$		
A	B	Y	A	B	Y	A	B	Y
0	0	0	0	0	1	0	0	1
1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	1	0
1	1	0	1	1	0	1	1	0

**5.6 Multiple-Input Gates**

Simliar and easy to understand, I might just skip it first.

**6 Beneath the Digital Abstraction**

Not essencial, but I’ll come back later, I promise.