LECTURE NOTE

on

DIGITAL DESIGN
With an Introduction to the Verilog HDL

Department of Electronics & Communications Engineering

Institute of Technical Education & Research

S"O"A Deemed to be University, Bhubaneswar

Chapter 1

Lecture Note-1

Introduction

Text Book: Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog, 6th Edition M. Morris R. Mano Mark Distribution
This subject comes under grading pattern 1 and follows Relative Grading System.
Mark Distribution (Total Mark = 100)
- Attendance : 5 Marks
- Major Lab / Session Assignments / Quizzes : 10 Marks
- Minor Assignments : 10 Marks
- Mid-term Examination : 15 Marks
- **Total Internal : 40 Marks**
- In-Lab Examination : 15 Marks
- Theory Examination : 45 Marks
- **Total External : 60 Marks**

1.1 Digital Systems

Digital systems have such a prominent role in everyday life that we refer to the present technological period as the digital age. Digital systems are used in communication, business transactions, traffic control, spacecraft guidance, medical treatment, weather monitoring, the Internet, and many other commercial, industrial, and scientific enterprises. We have digital telephones, digital televisions, digital versatile discs, digital cameras, handheld devices, and, of course, digital computers.

These devices have graphical user interfaces (GUIs), which enable them to execute commands (the GUI is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation). Most, if not all, of these devices have a special-purpose digital computer embedded within them.

One characteristic of digital systems is their ability to represent and manipulate discrete elements of information. Early digital computers were used for numeric computations. In this case, the discrete elements were the digits. From this application, the term digital computer

emerged. Discrete elements of information are represented in a digital system by physical quantities called signals. Electrical signals such as voltages and currents are the most common. Electronic devices called transistors predominate in the circuitry that implements these signals. The signals in most present-day electronic digital systems use just two discrete values and are therefore said to be binary. A binary digit, called a bit, has two values: 0 and 1. Discrete elements of information are represented with groups of bits called binary codes. For example, the decimal digits 0 through 9 are represented in a digital system with a code of four bits (e.g., the number 7 is represented by 0111). Through various techniques, groups of bits can be made to represent discrete symbols, not necessarily numbers, which are then used to develop the system in a digital format. Thus, a digital system is a system that manipulates discrete elements of information represented internally in binary form. In today's technology, binary systems are most practical because, as we will see, they can be implemented with electronic components.

Discrete quantities of information either emerge from the nature of the data being processed or may be quantized from a continuous process. In many cases, the quantization of a process can be performed automatically by an analog-to-digital converter, a device that forms a digital (discrete) representation of an analog (continuous) quantity.

The general-purpose digital computer is the best-known example of a digital system. There are fundamental reasons that commercial products are made with digital circuits. Like a digital computer, most digital devices are programmable. By changing the program in a programmable device, the same underlying hardware can be used for many different applications, thereby allowing its cost of development to be spread across a wider customer base. Dramatic cost reductions in digital devices have come about because of advances in digital integrated circuit technology. As the number of transistors that can be put on a piece of silicon increases to produce complex functions, the cost per unit decreases and digital devices can be bought at an increasingly reduced price. Equipment built with digital integrated circuits can perform at a speed of hundreds of millions of operations per second. Digital systems can be made to operate with extreme reliability by using error-correcting codes.

A digital system is an interconnection of digital modules. To understand the operation of each digital module, it is necessary to have a basic knowledge of digital circuits and their logical function.

A major trend in digital design methodology is the use of a HDL to describe and simulate the functionality of a digital circuit. An HDL resembles a programming language and is suitable for describing digital circuits in textual form. It is used to simulate a digital system to verify its operation before hardware is built. It is also used in conjunction with logic synthesis tools to automate the design process.

1.2 Why should we learn HDL?

We note that industry has largely abandoned schematic-based design entry, a style which emerged in the 1980s, during the nascent development of CAD tools for integrated circuit (IC) design. Schematic entry creates a representation of functionality that is implicit in the layout of the schematic.

Unfortunately, it is difficult for anyone in a reasonable amount of time to determine the functionality represented by the schematic of a logic circuit without having been instrumental in its construction, or without having additional documentation expressing the design intent.

Consequently, industry has migrated to HDLs (e.g., Verilog) to describe the functionality of a design and to serve as the basis for documenting, simulating, testing, and synthesizing the hardware implementation of the design in a standard cell-based ASIC or an FPGA.

The utility of a schematic depends on the careful, detailed documentation of a carefully constructed hierarchy of design modules.

In the old paradigm, designers relied upon their years of experience to create a schematic of a circuit to implement functionality.

In today's design flow, designers using HDLs can express functionality directly and explicitly, without years of accumulated experience, and use synthesis tools to generate the schematic as a byproduct, automatically.

Industry practices arrived here because schematic entry dooms us to inefficiency, if not failure, in understanding and designing large, complex ICs.

In this course we will try to give emphasis on how hardware is designed, to better prepare a student for a career in today's industry, where HDL-based design practices are dominant.

Lecture Note-2

Number Systems

1.3 Definition

A number system is defined as a system of writing for expressing numbers. It is the mathematical notation for representing numbers of a given set by using digits or other symbols in a consistent manner. Different types of number systems are:

1.3.1 Decimal Number System

It uses 10 digits from 0 to 9. For example - 35, 64, 135, 2345, etc.

1.3.2 Binary Number System

It uses two digits, 0 and 1. For example - 100, 011, 101, 1100, 10, etc. It is also called "base 2" number system.

1.3.3 Octal Number System

It uses eight digits, 0, 1, 2, 3, 4, 5, 6, and 7. It is also called "base 8" number system. For example - 35, 64, 71, 135, etc.

1.3.4 Hexadecimal Number System

It uses 10 digits and 6 letters, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The letters represent numbers starting from 10. Like A = 10, B = 11, C = 12, D = 13, E = 14 and F = 15. It is also called "base 16" number system. For example - 135, 782, 18D, 6FC etc. A decimal number such as 7,392 represents a quantity equal to 7 thousands, plus 3 hundreds, plus 9 tens, plus 2 units. The thousands, hundreds, etc., are powers of 10 implied by the position of the coefficients (symbols) in the number. To be more exact, 7,392 is a shorthand notation for what should be written as $7 \times 10^3 + 3 \times 10^2 + 9 \times 10^{1+2\times 10^0}$.

In general, $(a_n a_{n-1} \cdots a_0)_{10} = (a_n \times 10^n) + (a_{n-1} \times 10^{n-1}) + \cdots + (a_0 \times 10^0)$

The coefficients a_j (where j = 0 to n) are any of the 10 digits $(0, 1, 2, \cdots, 9)$, and the subscript value "j" gives the place value and, hence, the power of 10 by which the coefficient must be multiplied.

The decimal number system is said to be of **base**, or **radix**, 10 because it uses 10 digits and the coefficients are multiplied by powers of 10.
The binary system on the other hand is a different number system. The coefficients of the

binary number system have only two possible values: 0 and 1.
There are many different number systems. In general, a number expressed in a base-r system has coefficients multiplied by powers of r.

$$a_n.r^n + a_{(n-1)}.r^{(n-1)} + \cdots + a_2.r^2 + a_1.r + a_0 + a_{(-1)}.r^{(-1)} + a_{(-2)}.r^{(-2)} + \cdots + a_{(-m)}.r^{(-m)}$$

The coefficients a_j range in value from 0 to r-1. For example the coefficient values for base 5 can be only 0, 1, 2, 3, and 4. To distinguish between numbers of different bases, we enclose the coefficients in parentheses and write a subscript equal to the base used. An example of base 5 number is $(4021)_5$. Similarly, an example of base 8 or octal number is $(2746.51)_8$. An example of binary number would be $(11010.01)_2$.

It is customary to borrow the needed r digits for the coefficients from the decimal system when the base of the number is less than 10. The letters of the alphabet are used to supplement the 10 decimal digits when the base of the number is greater than 10 like in the case of hexadecimal numbers.

1.4 Number Base conversions

Representations of a number in a different radix are said to be equivalent if they have the same decimal representation. For example, (0011)8 and (1001)2 are equivalent as both have decimal value 9. The conversion of a number in base r to decimal is done by expanding the number in a power series and adding all the terms.

For example, the decimal equivalent of the binary number 11010.11 is 26.75, as shown from the multiplication of the coefficients by powers of 2

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{(-1)} + 1 \times 2^{(-2)} = 26.75$$

Another example of a base 5 number

$$(4021.2)_5 = 4 \times 5^3 + 0 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 + 2 \times 5^{(-1)} = (511.4)_{10}$$

Practice: Convert $(127.4)_8$ to decimal

1.4.1 Decimal to base r conversion.

Now let us look at a general procedure for the reverse operation of converting a decimal number to a number in base r. If the number includes a radix point, it is necessary to separate the number into an integer part and a fraction part, since each part must be converted differently. The conversion of a decimal integer to a number in base r is done by dividing the number and all successive quotients by r and accumulating the remainders.
For example convert $(41)_{10}$ to binary.

First, 41 is divided by 2 to give an integer quotient of 20 and a remainder of 1. The coefficient a_0 = 1. Then the quotient is again divided by 2 to give a new quotient 10 and a remainder of 0. The coefficient $a_1 = 0$. The process is continued until the integer quotient

becomes 0 and all the coefficients are collected from the remainders. In this particular example $a_2 = 0, a_3 = 1, a_4 = 0$ and $a_5 = 1$. After that no further division is possible as the quotient is 0. So the answer is thus written $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$.

Another example: Convert decimal 153 to octal. The required base r is 8. First, 153 is divided by 8 to give an integer quotient of 19 and a remainder of 1. Then 19 is divided by 8 to give an integer quotient of 2 and a remainder of 3. Finally, 2 is divided by 8 to give a quotient of 0 and a remainder of 2. So the answer would be $(153)_{10} = (231)_8$.

The conversion of a decimal fraction to binary is accomplished by a method similar to that used for integers. However, multiplication is used instead of division, and integers instead of remainders are accumulated. First, the fraction is multiplied by 2 to give an integer and a new fraction. Then the new fraction is multiplied by 2 to give a new integer and a new fraction. The process is continued until the fraction becomes 0 or until the number of digits has sufficient accuracy. The coefficients of the binary number are obtained from the integers.

Example: Convert $(0.6875)_{10}$ to binary. First 0.6875 is multiplied by 2 to give an integer of 1 and a new fraction 0.3750 (since 0.6875 multiplied by 2 is 1.3750). Here coefficient $a_{-1} = 1$. Continuing in a similar fashion we get$a_{-2} = 0, a_{-3} = 1$ and $a_{-4} = 1$. By this time the fraction is 0. So the answer is $(0.6875)_{10} = (0.a_{-1}.a_{-2}.a_{-3}.a_{-4})_2 = (0.1011)_2$.

Practice: Convert the following decimal numbers to binary.
(i) $(11)_{10}$
(ii) $(4)_{10}$
(iii) $(17)_{10}$
(iv) $(47.2)_{10}$

1.4.2 Octal and Hexadecimal numbers

The conversion from and to binary, octal, and hexadecimal plays an important role in digital computers, because shorter patterns of hex characters are easier to recognize than long patterns of 1's and 0's. Since $2^3 = 8$ and $2^4 = 16$, each octal digit corresponds to three binary digits and each hexadecimal digit corresponds to four binary digits.

The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right. The corresponding octal digit is then assigned to each group. The following example illustrates the procedure:

$(10\ 110\ 001\ 101\ 011.111\ 100\ 000\ 110)_2 = (26153.7406)_8$
2 6 1 5 3 7 4 0 6

Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of four digits:

$(10\ 1110\ 0110\ 1011.1111\ 0010)_2 = (2C6B.F2)_{16}$
2 C 6 B F 2

Conversion from octal or hexadecimal to binary is done by reversing the preceding procedure. Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent. The procedure is illustrated in the following examples:

$(673.124)_8 = (110\ 111\ 011.001\ 010\ 100)_2$

$6 \quad 7 \quad 3 \quad 1 \quad 2 \quad 4$

And

$(306.D)_{16} = (0011\ 0000\ 0110.1101)_2$

$3 \quad 0 \quad 6 \quad D$

Practice:

Convert $(0.513)_{10}$ to octal

$0.513 \times 8 = 4.104$
$0.104 \times 8 = 0.832$
$0.832 \times 8 = 6.656$
$0.656 \times 8 = 5.248$
$0.248 \times 8 = 1.984$
$0.984 \times 8 = 7.872$

The answer, to seven significant figures, is obtained from the integer part of the products:

$(0.513)_{10} = (0.406517.\ldots\ldots..)_8$

1.4.3 Problems

1. Convert the following numbers with the indicated bases to decimal:

(a) $(4310)_5$ (b) $(198)_{12}$

(c) $(435)_8$ (d) $(345)_6$

2. Convert the hexadecimal number 64CD to binary, and then convert it from binary to octal.

3. Express the following numbers in decimal:

(a) $(10110.0101)_2$ (b) $(16.5)_{16}$

(c) $(26.24)_8$ (d) $(DADA.B)_{16}$

(e) $(1010.1101)_2$

4. Convert the following binary numbers to hexadecimal and to decimal:

(a) 1.10010 (b) 110.010

5. Convert the decimal number 431 to binary in two ways:

(a) Convert directly to binary;

(b) Convert first to hexadecimal and then from hexadecimal to binary.
Which method is faster?

6. Determine the base of the numbers in each case for the following operations to be correct:

(a) $14/2 = 5$ (b) $54/4 = 13$ (c) $24 + 17 = 40$

Lecture Note-3

Binary arithmetic (addition and subtraction) Complements of Numbers (r and r-1 complement)

1.5 Binary Arithmetic

1.5.1 Binary Addition

Basic mathematical operations with binary numbers work similar to the decimal system. However there are a few rules specific to the binary system.

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$ (with a carry of 1)

For example:

1.
$$\begin{array}{r} {\scriptstyle 1} \\ 0\;\;1\;\;0\;\;1 \\ +\;0\;\;1\;\;1\;\;0 \\ \hline 1\;\;0\;\;1\;\;1 \end{array}$$

2.
$$\begin{array}{r} {\scriptstyle 1\;\;1} \\ 1\;\;0\;\;1\;\;1\;\;1 \\ +\;1\;\;0\;\;0\;\;1\;\;0 \\ \hline \underline{1}\;\;0\;\;1\;\;0\;\;0\;\;1 \end{array}$$

Here we have an overflow of 1

1.5.2 Binary Subtraction

Binary subtraction is also similar to that of decimal subtraction with the difference that when 1 is subtracted from 0, it is necessary to borrow 1 from the next higher order bit and that bit is reduced by 1 (or 1 is added to the next bit of subtrahend) and the remainder is 1.

Binary subtraction is much easier than the decimal subtraction when you remember the following rules:

0 - 0 = 0
0 - 1 = 1 (with a borrow of 1)
1 - 0 = 1
1 - 1 = 0

For example:

$$\begin{array}{rcccc} & & 0 & 10 & \\ & 1 & 1 & 0 & 0 \\ (-) & \underline{1} & \underline{0} & \underline{1} & \underline{0} \\ & 0 & 0 & 1 & 0 \end{array}$$

1.6 Complements of numbers

Complements are used in digital computers to simplify the subtraction operation and for logical manipulation. Simplifying operations leads to simpler, less expensive circuits to implement the operations.

There are two types of complements for each base-r system: the radix complement and the diminished radix complement.

The first is referred to as the $r's$ complement and the second as the $(r-1)'s$ complement.

When the value of the base r is substituted in the name, the two types are referred to as the $2's$ complement and $1's$ complement for binary numbers and the $10's$ complement and $9's$ complement for decimal numbers.

1.6.1 Diminished Radix Complement

Given a number N in base r having n digits, the $(r-1)'s$ complement of N, i.e., its diminished radix complement, is defined as $(r^n - 1)$ - N.

For decimal numbers, $r = 10$ and r - 1 = 9, so the $9's$ complement of N is $(10^n - 1) - N$. In this case, 10^n represents a number that consists of a single 1 followed by n $0s$. $10^n - 1$ is a number represented by n $9's$. For example, if $n = 4$, we have $10^4 = 10,000$ and $10^4 - 1$ = 9999. It follows that the $9's$ complement of a decimal number is obtained by subtracting each digit from 9.

For example:

1. Calculation of the $9's$ complement of 546700

$$\begin{array}{r} 9\ 9\ 9\ 9\ 9\ 9 \\ (-)\ 5\ 4\ 6\ 7\ 0\ 0 \\ \hline 4\ 5\ 3\ 2\ 9\ 9 \end{array}$$

2. Calculation of the $9's$ complement of 012398

$$\begin{array}{r} 9\ 9\ 9\ 9\ 9\ 9 \\ (-)\ 0\ 1\ 2\ 3\ 9\ 8 \\ \hline 9\ 8\ 7\ 6\ 0\ 1 \end{array}$$

For binary numbers, $r = 2$ and $r - 1 = 1$, so the $1's$ complement of N is $(2^n - 1)$ - N. Again, 2^n is represented by a binary number that consists of a 1 followed by n $0's$. 2^n – 1 is a binary number represented by n $1's$.

For example, if $n = 4$, we have $2^4 = (10000)_2$ and 2^4 - 1 = $(1111)_2$. Thus, the $1's$ complement of a binary number is obtained by subtracting each digit from 1. However, when subtracting binary digits from 1, we can have either $1 - 0 = 1$ or $1 - 1 = 0$, which causes the bit to change from 0 to 1 or from 1 to 0, respectively. Therefore, the $1's$ complement of a binary number is formed by changing $1's$ to $0's$ and $0's$ to $1's$.

For example:

1. Calculation of the $1's$ complement of 1011000

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ (-)\ 1\ 0\ 1\ 1\ 0\ 0\ 0 \\ \hline 0\ 1\ 0\ 0\ 1\ 1\ 1 \end{array}$$

2. Calculation of the $1's$ complement of 0101101

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ (-)\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 0\ 0\ 1\ 0 \end{array}$$

1.6.2 Radix Complement

The $r's$ complement of an $n - digit$ number N in base r is defined as $r^n - N$ for $N \neq 0$ and as 0 for $N = 0$. Comparing with the $(r-1)'s$ complement, we note that the $r's$ complement is obtained by adding 1 to the $(r-1)'s$ complement, since $r^n - N = [(r^n - 1) - N] + 1$.

Thus, the $10's$ complement of decimal 2389 is $7610 + 1 = 7611$ and is obtained by adding 1 to the $9's$ complement value.

$$\begin{array}{r} 9\ 9\ 9\ 9 \\ (-)\ 2\ 3\ 8\ 9 \\ \hline 7\ 6\ 1\ 0 \\ +\qquad\quad 1 \\ \hline 7\ 6\ 1\ 1 \end{array} \text{---------- } 10's \text{ complement}$$

The $2's$ complement of binary 101100 is 010011 + 1 = 010100 and is obtained by adding 1 to the $1's$ complement value.

$1's$ complement of $1\ 0\ 1\ 1\ 0\ 0 = 0\ 1\ 0\ 0\ 1\ 1$

$$\begin{array}{r} +\qquad\quad 1 \\ \hline 0\ 1\ 0\ 1\ 0\ 0 \end{array} \text{---------- } 2's \text{ complement}$$

Since 10^n is a number represented by a 1 followed by $n0's$, $10^n - N$, which is the $10's$ complement of N, can be formed also by leaving all least significant $0's$ unchanged, subtracting the first non-zero least significant digit from 10, and subtracting all higher significant digits from 9.

For example:

1. $10's$ complement of $012398 = 987602$

$$\begin{array}{r} 1\ 0\ 0\ 0\ 0\ 0 \\ (-)\ 0\ 1\ 2\ 3\ 9\ 8 \\ \hline 9\ 8\ 7\ 6\ 0\ 2 \end{array}$$

2. $10's$ complement of $246700 = 753300$

$$\begin{array}{r} 1\ 0\ 0\ 0\ 0\ 0 \\ (-)\ 2\ 4\ 6\ 7\ 0\ 0 \\ \hline 7\ 5\ 3\ 3\ 0\ 0 \end{array}$$

The $10's$ complement of the first number is obtained by subtracting 8 from 10 in the least significant position and subtracting all other digits from 9.

The $10's$ complement of the second number is obtained by leaving the two least significant $0's$ unchanged, subtracting 7 from 10, and subtracting the other three digits from 9.

Similarly, the $2's$ complement can be formed by leaving all least significant $0's$ and the first 1 unchanged and replacing $1's$ with $0's$ and $0's$ with $1's$ in all other higher significant digits.

For example:

1. $2's$ complement of the binary number $1101100 = 0010100$

2. $2's$ complement of the binary number $0110111 = 1001001$

The $2's$ complement of the first number is obtained by leaving the two least significant $0's$ and the first 1 unchanged and then replacing $1's$ with $0's$ and $0's$ with $1's$ in the other four most significant digits.

The $2's$ complement of the second number is obtained by leaving the least significant 1 unchanged and complementing all other digits.

It is also worth mentioning that the complement of the complement restores the number to its original value. To see this relationship, note that the $r's$ complement of N is $r^n - N$, so that the complement of the complement is $r^n - (r^n - N) = N$ and is equal to the original number.

1.7 Practice Problems

1. Add and multiply the following numbers without converting them to decimal.

(a) Binary numbers 1011 and 101.

(b) Hexadecimal numbers $2E4$ and 34.

2. Obtain the $1's$ and $2's$ complements of the following binary numbers:

(a) 00010000 (b) 00000000

(c) 11011010 (d) 10101010

(e) 10000101 (f) 11111111

3. Find the $9's$ and the $10's$ complement of the following decimal numbers:

(a) $25,478,03$ (b) $63,325,600$

(c) $25,000,000$ (d) $00,000,000$

4.

(a) Find the $16's$ complement of $C3DF$.

(b) Convert $C3DF$ to binary.

(c) Find the $2's$ complement of the result in (b).

(d) Convert the answer in (c) to hexadecimal and compare with the answer in (a)