Digital circuit Experiment manual
Contents

Essential tools and Experimental part list ................................................................. 7
Chapter 1 : Introduction to Digital Electronics.......................................................... 9
Chapter 2 : Number system...................................................................................... 13
Chapter 3 : Introduce the Digital Circuit Experiment board................................. 23
Chapter 4 : How to use the breadboard.................................................................... 27
Experiment 1 NOT gate ......................................................................................... 31
Experiment 2 OR Gate and AND gate................................................................. 35
Experiment 3 NOR gate - The Universal gate...................................................... 39
Experiment 4 NAND gate - The Universal gate.................................................... 43
Experiment 5 Exclusive-OR gate ............................................................................ 47
Experiment 6 Exclusive-NOR gate .......................................................................... 49
Experiment 7 Boolean algebra .............................................................................. 53
Experiment 8 Karnaugh map .................................................................................. 59
Experiment 9 Binary Adder and Subtractor circuit............................................. 67
Experiment 10 Open-collector output ................................................................ 75
Experiment 11 3-States Buffer circuit................................................................. 79
Experiment 12 Latch circuit .................................................................................. 83
Experiment 13 Simple signal generator in the digital circuit............................ 87
Experiment 14 Flip-Flop......................................................................................... 99
Experiment 15 Counter......................................................................................... 105
Experiment 16 Debounce switch - circuit and interface.................................... 117
Essential tools and Experimental part list

Essential tools

1. Digital circuit experiment board (NX-4i is recommended) 1 set
2. Digital or Analog Multimeter included Resistance measurement scale and test lead 1 set
3. 20MHz 2-ch. Oscilloscope with test lead 1 set
4. Solid wire jumper #22AWG

Integrated circuit (IC) for experimental part list

CMOS ICs (CD/MC14xxx/HEF - support)

- 4029/MC14029/CD4029 x 1
- 4069/MC14069/CD4069 x 1
- 4516/CD4516 x 1
- 4543/MC14543/CD4543 x 1

TTL ICs (74LS/HC/HCT/ACT/ALS/F - support)

- 7400 x 2
- 7402 x 1
- 7404 x 1
- 7407/74LS07 x 1
- 7408 x 1
- 7411 x 1
- 7420 x 1
- 7430 x 1
- 7432 x 1
- 7473 x 1
7474 x 1
7476 x 2
7485 x 1
7486 x 1
74HC123 x 1
74HC/HCT125 x 1
74138 x 1
74139 x 1
74147 x 1
74153 x 1
74194 x 1
74HC/HCT541 x 1
6264 Static RAM x 1
AT27C256R pre-programmed ROM x 1
Chapter 1

Introduction to Digital Electronics

In digital electronics there are only two voltage states present at any point within a circuit. These voltage states are either high or low. The meaning of a voltage being high or low at a particular location within a circuit can signify a number of things.

The high and low states can be represented as true and false statements, which are used in Boolean logic. In most cases, high = true and low = false. However, this does not have to be the case—You could make high = false and low = true. The decision to use one convention over the other is a matter left ultimately to the designer. In digital lingo, to avoid people getting confused over which convention is in use, the term positive true logic is used when high = true, while the term negative true logic is used when high = false.

In Boolean logic, the symbols 1 and 0 are used to represent true and false, respectively. Now, unfortunately, 1 and 0 are also used in electronics to represent high and low voltage states, where high = 1 and low = 0.

1.1 Logic level

These are the voltages that represent the binary levels ‘0’ and ‘1’. Using conventional logic (positive logic) binary ‘0’ is represented by a low voltage and binary ‘1’ by a higher (more positive) voltage.

The actual voltage levels are related to that voltage used to supply power to the circuit. In TTL circuits the power supply (+Vcc) is +5 V ±5%, while CMOS systems will work on supply voltages between 3.0 V and 18 V. The Figure 1-1 shows the logic level of digital devices.

(A) TTL logic level

+5V

+2V

0.8V

↓ "0"

↑ "1"

(B) CMOS level logic

+Vcc

2/3Vcc

1/3Vcc

↓ "0"

↑ "1"

Figure 1-1 How to define the logic level of digital devices.
In TTL circuits the logic level '0' could be considered as 0 V while logic level ‘1’ would be defined as +5 V. However there are practical factors which must be considered. To expect voltage levels to be exact is unrealistic, zero volts may actually be ‘almost zero’, i.e. 0.2 V, while +5 V may in reality be 4.6 V.

In CMOS circuits will operate on a supply voltage (VDD) between 3 V and 18 V, consequently the logic thresholds are determined by the actual supply voltage that is used. ‘0’ is represented by voltages between 0 V and 1/3 of VDD, and ‘1’ by voltages between 2/3 of VDD and VDD as shown in Figure 1-1 (B). By allocating voltage thresholds in this fashion very definite regions of the pulse relate to the two logic levels.

1.2 Digital Integrated Circuit voltage parameter

In the Digital ICs datasheet, the voltage parameter which are interest as follows:

1. Vil(min) : The minimum input voltage of logic "LOW". Normally is 0V.
2. Vil(max) : The maximum input voltage of logic "LOW".
3. Vih(min) : The minimum input voltage of logic "HIGH".
4. Vih(max) : The maximum input voltage of logic "HIGH".
5. Vol(min) : The minimum output voltage of logic "LOW".
6. Vol(max) : The maximum output voltage of logic "LOW".
7. Voh(min) : The minimum output voltage of logic "HIGH".
8. Voh(max) : The maximum output voltage of logic "HIGH".

1.3 Logic Gates

A logic gate is a device which can have more than one binary input but a single binary output. The state of the output is determined by the input conditions.

Every logic gate can be depicted as a symbol, Boolean expression and truth table. While the truth tables and Boolean expressions are universally accepted there are considerable differences in the gate symbols. In the United Kingdom logic symbols are devised by the British Standards Institute (BSI). Unfortunately however, the American standard symbols (MIL/ANSI) for logic gates have a greater following. This has now reached such a level that the BSI symbols for logic gates are being almost universally superseded by the MIL/ANSI symbols. The reason for this is probably that the American versions are clearer and more easily understood. It is quite possible that in the not to distant future BSI symbols will be changed so that they conform to those of the MIL/
ANSI. For reasons of clarity MIL symbols for logic circuits will be used throughout this book.

1.4 Fan-out

A logic gate is an electronic circuit and as such can only supply a certain amount of output current before its output voltage will be affected. A gate has only one output and this may be connected to the input of a number of other gates (see Fig. 2.8). These gates will each draw current, so there will eventually come a point when the gate to which they are connected will become overloaded and its output may change to an indeterminate level.

The ‘fan-out’ of a logic gate refers to the maximum number of other similar gate inputs that can be connected to a single gate output without changing its specified logic output level. **Typical fan-outs are: TTL = 10 CMOS = 50.**

1.5 Fan-in

One input to a gate is considered to be a standard load and represents a fan-in of 1. A complete logic circuit or network may have only a single input but this will draw more current than a single gate, so it is expressed in terms of the number of standard loads it represents.

From the designers point of view it is the ‘fan-out’ capability of a gate that is of major importance.

1.6 Truth table

A truth table is a mathematical table used in logic. It is specifically in connection with boolean algebra and boolean functions. In particular, truth tables can be used to tell whether a propositional expression is true for all legitimate input values, that is, logically valid.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 1-2 Show the example of simple truth table. First 2 column are input values and last column in the right is result or output value
Truth tables for classical logic are limited to Boolean logical systems in which only two logical values are possible, \textit{false} and \textit{true}, usually written \textit{F} and \textit{T}, or sometimes \textit{0} or \textit{1}, respectively.

**1.7 Boolean Algebra**

One of the most significant mathematical tools available to electronics designers was actually invented for quite a different purpose. This tool is to use mathematical techniques to represent and rigorously test logical and philosophical arguments.

In digital circuit design, designers must set the purpose or result to a Truth table. After that make the logic equation from the Truth table. Boolean algebra will use to simplify the original equation to simplest logic equation. It means the smallest digital circuit and using minimum logic gates.

**1.8 Digital circuit experiment tooling**

For the digital circuit experimental in this manual require a Digital Circuit Experiment Board which have many tools as follows:

1. Logic monitor
2. Logic switch
3. Debounce switch
4. +5V DC Power supply
5. Function generator or Pulse generator
6. LED 7-segments decoder
7. Logic probe
8. High-current load driver circuit
9. Digital Volt and Ammeter
10. Jumper wires (#22AWG solid wire)

Addition, the optional tool are Analog or Digital Storage Oscilloscope (20MHz and 1 or 2-ch. minimum requirement) and Cutter plier for cutting and peel the insulator of the jumper wires.
Chapter 2
Number system

The basic number system in digital electronics is Binary. The binary number can extend to octal number, decimal number and hexadecimal number for more complex calculations. Understanding about the number system is very important. It includes the number value, calculation, conversion, number code and parity checking.

In this chapter will introduce about detail of each number system, bit variable and number system conversion.

2.1 Binary number system

This is the basic code consisting of a numbering system to the base 2. Each binary digit or bit represents a power of 2 as follows:

**Total value =** $2^\text{digit}$

For 2 digits, total value is $2^2 = 4$

For 3 digits, total value is $2^3 = 8$

For 4 digits, total value is $2^4 = 16$

2.1.1 Counting value in Binary number

There are 2 values in the Binary number system. They are 0 and 1. Increasing the value is add a digit. For the changing clearly, we will use the decimal number for comparison as follows:

<table>
<thead>
<tr>
<th>Binary</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
</tr>
<tr>
<td>1111</td>
<td>15</td>
</tr>
</tbody>
</table>
2.1.2 Bit variables

In operation of binary number from 1 digit to 2, 4 and 8 digits cause the new variables as follows:

(1) **bit**: It is binary number one digit or call binary digit. There are 2 numbers as 0 and 1.

(2) **byte**: It is 8 binary digits or bits.

(3) **LSB**: Least Significant Bit: It is last right digit of binary number. The digit weight is lowest as $2^0$. If this bit as “1”, the value is equal $1 \times 2^0 = 1 \times 1 = 1$. If as “0”, the value is equal $0 \times 2^0 = 0 \times 1 = 0$.

(4) **Digit or Bit assignment**: The last right digit is assigned to Bit 0 (b0) or LSB. Next bit is assigned to Bit 1 (b1) and count to left order until complete. The 8-bit assignment is b7 b6 b5 b4 b3 b2 b1 b0.

(5) **MSB**: Most Significant Bit: It is last left digit of binary number. In 8-bit number, the MSB is bit 7. The digit weight is $2^7$ or 128. The digit weight would be change following the number of digit.

2.1.3 Digit weight

The digit weight of binary can calculate from power of 2. At the last right bit as bit0 or LSB, the digit weight is $2^0$ or equal 1. Next bit is bit1. The digit weight is $2^1$ or 2. Next bit2, the digit weight is $2^2$ or 4. The information below is summary of 8-bit digit weight.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Digit weight</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$2^0$</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$2^1$</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>$2^2$</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>$2^3$</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>$2^4$</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>$2^5$</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>$2^6$</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>$2^7$</td>
<td>128</td>
</tr>
</tbody>
</table>
2.2 Binary number conversion

2.2.1 Binary to Decimal conversion

Refer the digit weight of binary number, you can convert number system from Binary to Decimal following some examples as follows.

**Example 2-1** Convert the binary number 1011 to decimal

1. Define the digit weight

<table>
<thead>
<tr>
<th>Digit</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digit weight</td>
<td>$2^3$</td>
<td>$2^2$</td>
<td>$2^1$</td>
<td>$2^0$</td>
</tr>
<tr>
<td>Binary</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2. Multiply the digit weight by binary number at each bit. After that, add all multiplication together.

   \[
   \text{Decimal} = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\
   = (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1) \\
   = 8 + 0 + 2 + 1 \\
   = 11
   \]

2.2.2 Decimal to Binary conversion

The suggestion method is Repeated Division by 2. Any decimal number divided by 2 will leave a remainder of 0 or 1. Repeated division by 2 will leave a string of 0s and 1s that become the binary equivalent of the decimal number.

**Example 2-2** Convert the decimal number 13 to binary

1. Divide the decimal number by 2 and note a remainder

   \[13 \div 2 = 6 + \text{remainder } 1 \text{ (LSB)}\]

   The remainder is the least significant bit of the binary equivalent of 13.

2. Divide the quotient from the previous division and note the remainder. The remainder is the second LSB.

   \[6 \div 2 = 3 + \text{remainder } 0\]

3. Continue this process until the quotient is 0. The last remainder is the most significant bit of the binary number.

   \[3 \div 2 = 1 + \text{remainder } 1\]
   \[1 \div 2 = 0 + \text{remainder } 1 \text{ (MSB)}\]

4. To write the binary equivalent of the decimal number, read the remainders from the bottom up.

   \[13_{10} = 1101_2\]
2.3 Signed of Binary number

Binary number can assign to Positive and Negative value by assignment at MSB bit. If MSB is "0", that binary number is positive. In the other hand, if MSB is "1" that binary number is negative.

These informations below present the unsigned and signed binary number compare with decimal number to make the understanding clearly.

<table>
<thead>
<tr>
<th>Binary</th>
<th>Signed number</th>
<th>Decimal</th>
<th>Unsigned number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>-8</td>
<td>8</td>
<td>-8</td>
</tr>
<tr>
<td>1001</td>
<td>-7</td>
<td>9</td>
<td>-7</td>
</tr>
<tr>
<td>1010</td>
<td>-6</td>
<td>10</td>
<td>-6</td>
</tr>
<tr>
<td>1011</td>
<td>-5</td>
<td>11</td>
<td>-5</td>
</tr>
<tr>
<td>1100</td>
<td>-4</td>
<td>12</td>
<td>-4</td>
</tr>
<tr>
<td>1101</td>
<td>-3</td>
<td>13</td>
<td>-3</td>
</tr>
<tr>
<td>1110</td>
<td>-2</td>
<td>14</td>
<td>-2</td>
</tr>
<tr>
<td>1111</td>
<td>-1</td>
<td>15</td>
<td>-1</td>
</tr>
</tbody>
</table>

In signed binary number to decimal conversion can use similar technique.

For example, 1000 binary number the MSB is "1". The digit weight is $2^3$ or 8. This signed binary number is equal to -8. In conversion the remain 3 bit is positive value. Thus, the conversion is $-8 + 0 = -8$.

Next example, convert 1101 signed number to decimal. At MSB is -8 and 3 last right bit is positive as $+5$ [$101 = (2^2 x 0) + (2^1 x 0) + (2^0 x 1)]$. The conversion is $-8 + 5 = -3$. 
2.4 Binary number calculation

2.4.1 Addition

The simplest arithmetic operation in binary is addition. Adding two single-digit binary numbers is relatively simple:

(i) \(0 + 0 = 0\)
(ii) \(0 + 1 = 1\)
(iii) \(1 + 0 = 1\)
(iv) \(1 + 1 = 0\); Carry = 1

This is 2-digit addition example.

The format of Binary adder with carry can show below.

\[
\begin{array}{c}
\text{(i)} & 1 & 0 \\
+ & 0 & 1 \\
\hline
1 & 1 \\
\end{array} 
\]

\[
\begin{array}{c}
\text{(ii)} & 1 & 1 \\
+ & 0 & 1 \\
\hline
1 & 0 & 0 \\
\end{array} 
\]

\[
\begin{array}{c}
\text{(iii)} & 1 & 1 \\
+ & 1 & 1 \\
\hline
1 & 1 & 0 \\
\end{array} 
\]

<table>
<thead>
<tr>
<th>Carry in</th>
<th>Denominator</th>
<th>Adder</th>
<th>Result</th>
<th>Carry out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+ 0</td>
<td>+ 0</td>
<td>= 0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>+ 0</td>
<td>+ 0</td>
<td>= 1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>+ 1</td>
<td>+ 0</td>
<td>= 1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>+ 1</td>
<td>+ 0</td>
<td>= 0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>+ 1</td>
<td>+ 1</td>
<td>= 0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>+ 1</td>
<td>+ 1</td>
<td>= 1</td>
<td>1</td>
</tr>
</tbody>
</table>

2.4.2 Subtraction

Subtracting two single-digit binary numbers is relatively simple:

(i) \(0 - 0 = 0\)
(ii) \(1 - 0 = 0\)
(iii) \(1 - 1 = 0\)
(iv) \(0 - 1 = 1\); Borrow = 1 or \(10 - 1 = 1\)

Subtracting a positive number is equivalent to adding a negative number of equal absolute value; computers typically use two’s complement notation to represent negative values. This notation eliminates the need for a separate “subtract” operation. The subtraction can be summarized with this formula:

\[A - B = A + \text{not } B + 1\]
2.4.3 two’s complement

The two’s complement of a binary number is the value obtained by subtracting the number from a large power of two (specifically, from \(2^N\) for an N-bit two’s complement).

A two’s-complement system or two’s-complement arithmetic is a system in which negative numbers are represented by the two’s complement of the absolute value.

First step - invert all bits of subtractor "0" to "1" and "1" to "0".

Second step - add "1" to the result from first step.

Final step - add the result from second step with the denominator.

Example 2-3 Calculate the result of 0010 - 0100

(1) Invert bit of subtractor 0100 >> 1011
(2) Add 1 >> 1011 + 1 = 1100
(3) Add with the denominator >> 0010 + 1100 = 1110

Signed number of 1110 = -2 in decimal

2.4.4 Multiplication

Multiplying two single-digit binary numbers is relatively simple:

(i) \(0 \times 0 = 0\)
(ii) \(0 \times 1 = 0\)
(iii) \(1 \times 0 = 0\)
(iv) \(1 \times 1 = 1\)

Example 2-4 Calculate the result of 1001 x 1011

\[
\begin{array}{cccc}
  & 1 & 0 & 0 & 1 \\
\times & 1 & 0 & 1 & 1 \\
\hline
  & 1 & 0 & 0 & 1 \\
  & 1 & 0 & 0 & 1 \\
+ & 1 & 0 & 0 & 1 \\
\hline
  & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{array}
\]
2.5 Hexadecimal number

This is a number system to the base 16 and like octal code gives a shorthand version of large binary numbers. It is particularly useful in microprocessor based systems where 8 and 16 bit binary words are used. The word hexadecimal means that six alphabetical characters are used along with those representing decimal values 0 to 9. In this way letters A to F are used to represent numbers 10 to 15 with the code shown below.

<table>
<thead>
<tr>
<th>Binary</th>
<th>Decimal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
<td>B</td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
<td>C</td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
<td>D</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td>E</td>
</tr>
<tr>
<td>1111</td>
<td>15</td>
<td>F</td>
</tr>
<tr>
<td>10000</td>
<td>16</td>
<td>10</td>
</tr>
</tbody>
</table>
2.6 Hexadecimal number conversion

2.6.1 Hexadecimal with Binary conversion

A digit of hexadecimal number presents with 4-bit of binary number. In binary to hexadecimal conversion must group the binary number to 4 bits and replace with hexadecimal number.

**Example 2-5** Convert binary number 101011100011 to hexadecimal

Binary number 101011100011
4-bit grouping 1010 1110 0011
Convert to Hexadecimal A E 3

The result is AE3H

In hexadecimal to binary conversion is similar. Separate each digit of hexadecimal number and convert to 4 bits of binary in each digit.

**Example 2-6** Convert hexadecimal number BC75 to binary

Hexadecimal number BC75H
Separate digit B C 7 5
Convert to binary number 1011 1100 0111 0101

The result is 1011110001110101

2.6.2 Decimal to Hexadecimal conversion

Easiest method is convert decimal to binary. After that convert the binary number to hexadecimal number.

**Example 2-7** Convert decimal number 302 to hexadecimal

Decimal number 302
Convert to binary number 100101110
4-bit grouping from last right digit 1 0010 1110
Convert to hexadecimal number 1 2 E

The result is 12EH (H means Hexadecimal)
2.6.3 Hexadecimal to Decimal conversion

Hexadecimal numbers can be converted to hexadecimal by the sum-of-weighted-hex-digits method. The conversion step are:

(i) Separate hexadecimal digit
(ii) Conver to decimal.
(iii) Multiply the decimal with the digit weight of hexadecimal.
(iv) Add all multiplication from step (iii).

**Example 2-8** Convert hexadecimal number AE3 to decimal.

<table>
<thead>
<tr>
<th>Hexadecimal number</th>
<th>A</th>
<th>E</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert to decimal in each digit</td>
<td>10</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>Digit weight</td>
<td>256 ((16^2))</td>
<td>16 ((16^1))</td>
<td>1 ((16^0))</td>
</tr>
<tr>
<td>Multiplication</td>
<td>2560</td>
<td>224</td>
<td>3</td>
</tr>
<tr>
<td>Total addition</td>
<td>2560+224+3 = 2787</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The result of conversion is 2787*
Chapter 3
Introduce the Digital Circuit Experiment board

The Digital circuit experiment board is introduced in this manual as NX-4i Versatile Digital Circuit Experiment Board. This board contains many necessary tools for supporting all basic digital circuit experiment and more complex.

3.1 Features of the NX-4i experiment board

The main features of NX-4i board is:

- Dual DC power supply
  ±12V 1A and ±5V 1A with Short circuit protection and indicators
- 8-ch. Logic switch with indicators
- 16-ch. Logic monitor
- 2-Debounce switch
- 3-State TTL Logic probe
  Indicate Hi-Lo Logic and Pulse
- Function generator
  1Hz to 100kHz frequency output.
  Button Selection with Frequency range indicators.
  3 Waveform selection:
    Sine, Square and Triangle
    Complete with a TTL output
- Binary to Hexadecimal decoder to four 7-segment LEDs
- 8-Bit Analog to Digital Converter
  Maximum input voltage +5V
- Adjustable 0-5V Voltage reference source
  for Analog to Digital Conversion experiment
- 8-Bit Digital to Analog Converter
  Output of 0 to +5V R-2R ladder type
Breadboard
- Measuring 5" x 7"
- Point contact 1,600 points

7-segment LED in Common Anode and Cathode
- 2 digits with limit-current resistor each

250mW Audio Amplifier with 250mW 8Ω Speaker

8-ch. High-Current Driver
- +12V 500mA Load driver
- Supplied with Screw terminal block

Figure 3-1 shows the layout of NX-4i experiment board.

3.2 How to use the tools on NX-4i Experiment Board

3.2.1 How to use the power supply

On the NX-4i board has 4 DC supply: +12V, -12V, +5V and -5V regulated. The connecting of DC supply can do directly and DC voltage has the common ground. All output has short circuit protection that cut-off the terminal suddenly if short output is occurred. The warning sound will be lound and indicator turned-on to report experimenters.

**POWER SUPPLY**

±12V 1A and ±5V 1A with short circuit protection

**SHORT CIRCUIT INDICATOR**

The indicator of each output will on if that output is short-circuit or overload occured.
3.2.2. How to use the Function Generator

3.2.2.1 Waveform Selection

Press WAVEFORM button and see the Waveform indicator. Press until indicator target on. For example, if would like to select SINE, press WAVEFORM button until SINE’s indicator on. In Squarewave and Triangle waveform selection are same method.

3.2.2.2 Frequency Selection

(1) Target frequency is 500Hz. Press RANGE button until $x_{100}$ Range’s indicator on.

(2) Connect output of Function Generator to Oscilloscope. Adjust AMPLITUDE knob to center. See the output waveform at Scope’s display.

(3) Adjust FREQUENCY knob until the waveform’s frequency is the target value.

3.2.2.3 Amplitude Adjustment

Turn AMPLITUDE knob until the waveform’s amplitude level is target value.
3.2.3 How to use the Amplifier

3.2.3.1 Connect the Output of Amplifier and Ground to SPEAKER input terminal near the Amplifier.
3.2.3.2 Apply the input signal to Amplifier input.
3.2.3.3 Turn VOLUME knob to the target level.

**Warning**
3.2.3.4 Must connect Amplifier’s output to Speaker input before apply the input signal.
3.2.3.5 When finished, must disconnect or remove all circuit wires from The Amplifier and Speaker input.

3.2.4 How to use the logic switch

**To select logic “1”** -press logic switch until LED-red show status logic “1”. It supply +5V at output point, that means logic “1”

**To select logic “0”** -press logic switch until LED-green show the logic status “0”. Output voltage is about 0.8V, it equal logic “0”
3.2.5 How to use the logic monitor

This tool is shown the logic status. On this board has 8 channels which enough for all digital circuit experiments.

At default, the all input are pulled down to ground. The LED output off that mean logic “0” occur.

To test the logic level in the circuit, connect the input point to a test point on the circuit. Observe the result at the LED monitor.

- **Light ON** mean the point under test has logic status “1”
- **Light OFF** mean the point under test has logic status “0”

**Attention:** The logic monitor can use with every digital circuit, which use supply voltage +5V or TTL level.

3.2.6 How to use the TTL logic probe

3.2.6.1 Connect input “IN” to the examined point

3.2.6.2 If the green light at “LO” turn-on, the examined point will be logic level “0”

3.2.6.3 If the red light at “HI” turn-on, the logic level equal “1”

3.2.6.4 If the yellow light at “PULSE” blink, in the same time light at LO and HI blink together. It shows the changing of unstable logic level or it occurs pulse at this point.
3.2.7 How to use the debounce switch

It’s used for supplying the single perfect square-wave pulse for the digital circuit. There is 2 outputs; following,

1. **Rising edge pulse output** - if the switch is not pressed, the logic level will be “0”. If press the switch, it will be “1” up to press it.

2. **Falling edge pulse output** - if the switch is not pressed, the logic level will be “1”. If press the switch, it will be “0” up to press it.

It is noticed that, logic switch, logic monitor and debounce switch can work suddenly, do not use/connect the ground line.

![Debounce switch diagram]

3.2.8 How to use the LED 7 segment decoder (Binary to hexadecimal decoder and display)

This tool’s function is decoding binary data to hexadecimal data and drive to LED 7 segment for displaying. **On this board has 4 units.**

At default, the display show “0” because all inputs is pulled-down to ground. It causes the all inputs receive “0” data.

To use, fed data signal into DCBA inputs and “dp” input, in case wants to use the dot point of LED 7 segment. For driving the dot-point in LED, connect the positive voltage (+5V max.) at “dp” point.

**Thus, please notice like logic switch and logic monitor. It can connect suddenly, don’t connect the ground line.**

![LED 7 segment decoder diagram]
3.2.9 How to use the variable voltage reference 0-5V

On the NX-4i board has the variable voltage reference 0-5V for the analog to digital converter experiment. The voltage can connect directly and use the common ground.

The adjusting is very easy. Experimenter can turn the knob of variable resister to change the voltage value. The experimenter can check the voltage level by connected the multi-meter which selected in DC voltage range at point + V, it will show the changing of the voltage 0-5 V followed by adjusting.

3.2.10 How to use the Analog to Digital converter circuit (ADC0804)

This tool is specially designed for ADC experiment convenience.

3.2.10.1 Connect D0-07 digital data output to the logic monitor.

3.2.10.2 Fed DC voltage that wanted into input “Ain” and can fed voltage not over than +5V and the voltage does not DC supply voltage. For recommend using from variable voltage reference 0-5V on this experiment board. If use voltage input more than +5V, please connect pass attuator circuit on this experiment board for protection ADC circuit

3.2.10.3 Connect WR input to ground or fed logic “0” into this input

3.2.10.4 Adjusted voltage input at Ain. Observe the changing at the logic monitor.

3.2.10.5 If want to keep this value, change the logic at WR to logic “1”

3.2.10.6 If want to convert the new value, connect WR to ground.

3.2.10.7 In case, the automatic conversion, connect the WR point to INTR point. When the analog input changes, the digital data output will change together.
3.2.11 How to use digital to analog converter circuit

Same the ADC circuit, this tool is designed for DAC experiment convenience.

3.2.11.1 Connect D0-07 digital data input to the logic switch.

3.2.11.2 Connect the “Aout” output into voltmeter or oscilloscope or the other analog instruments as you want.

3.2.11.3 Fed data or digital signal as you want.

3.2.11.4 Observe the changing result from analog signal instrument. If connect voltmeter, it show voltage value since 0-5V (approx.) from digital data 00000000-11111111.

3.2.11.5 Can be converted new value all time.

3.2.12 How to use the Free LED 7 segments

On the NX-4i board prepares 2 pairs of LED 7 segments; separated by common anode type and common cathode type (1 pairs per type). Convent for test circuit involved figure output.

3.2.12.1 Connect the a-g and pt input to digital signal directly. Because LED 7 segment has connected the limit currency resister already.

3.2.12.2 In case, to use LED 7 segment common anode type have to connect the COM input of LED 7 segment to +5V volatge or logic signal”1”

3.2.12.3 In case, to use LED 7 segment common cathode type have to connect the COM input into ground or logic signal”0”
3.2.13 How to use 8-channels high current driver

On NX-4i board provides the special tool for experiment about high current driver such as stepper motor driver. It has 8-channels of high current open collector driver. The heart of this circuit is ULN2803 driver-IC.

Experimenters can connect digital signal to input of ULN2803 directly. At output, it has terminal block for connecting high current load such as +12V uni-polar stepper motor. At output terminal block has 8 of ULN2803’s output and +V. Sample stepper motor connection circuit for experiment see below.
Chapter 4
How to use the breadboard

A breadboard is a reusable solderless device used to build a (generally temporary) prototype of an electronic circuit and for experimenting with circuit designs. A typical breadboard will have strips of interconnected electrical terminals, known as bus strips, down one or both sides - either as part of the main unit or as separate blocks clipped on-to carry the power rails.

4.1 Introduction to the breadboard

A modern solderless breadboard consists of a perforated block of plastic with numerous tin plated phosphor bronze spring clips under the perforations. Integrated circuits (ICs) in dual inline packages (DIPs) can be inserted to straddle the centerline of the block. Interconnecting wires and the leads of discrete components (such as capacitors, resistors, inductors, etc.) can be inserted into the remaining free holes to

Figure 4-1 The breadboard outer and insider structure
complete the circuit topology. In this manner, a variety of electronic systems may be prototyped, from small circuits to complete central processing units (CPUs). However, due to large stray capacitance (from 2-25pF per contact point), solderless breadboards are limited to operating at relatively low frequencies, usually less than 10 MHz, depending on the nature of the circuit. The Figure 4-1 shows the outer and insider structure of breadboard.

It's important to understand how this breadboard works. With a little bit of preparation, it will be even easier to use with the experiments that follow.

The innermost portion of the breadboard is where we will connect our components. This section of the breadboard consists of several columns of sockets (there are numbers printed along the top for reference). For each column there are two sets of rows. The rows are labeled A through E and F through J, respectively. For any column, sockets A through E are electrically connected. The same holds true for rows F through J.

Above and below the main section of breadboard are two horizontal rows of sockets, each divided in the middle. These horizontal rows (often called “rails” or “buses”) will be used to carry +5 volts (Vdd) and Ground (Vss). Our preparation of the breadboard involves connecting the rails so that they run from end-to-end, connecting the top and bottom rails together and, finally, connecting the rails to Vdd and Vss. If we X-Rayed the breadboard, we would see the internal connections following the Figure 4-2.
4.2 The wire jumper for breadboard

The solid wire number 22AWG is suitable wire jumper for plugging into the breadboard. The wire must be coated with Nigle or Tin. It easy to bend and cut.

To start by setting your wire stripper for 22 (0.34 mm$^2$) gauge. Take the spool of black wire and strip a ¼-inch (6 mm) length of insulation from both ends of the wire. With your needle-nose pliers, carefully bend the bare wire 90 degrees. Bend the second bare end 90 degrees so that the wire forms a squared “U” shape with the insulation in the middle. The Figure 4-3 shows example of preparing wire jumper and components for plugging into the breadboard for construction the experimental circuit.

4.3 Circuit construction on the breadboard

In the Figure 4-4 shows the connection of wire jumper after plug into the breadboard. Figure 4-5 shows the example of circuit construction on the breadboard. The good circuit construction includes regulations, easy to checking and use the wire jumper in the quantity that is appropriate. Suggest to construct the circuit from left side to right side and up to down. Assign the input at left side or below of breadboard and output at right side or top side. It causes to easy checking and change connection comfortable.
Experiment 1
NOT gate

Part list

- 7404 x 1
- 4069 or MC14069 x 1

Theory

This produces an output which is the inverse or opposite of the input signal. Therefore if the input to an inverter is Logic 1 the output will be Logic 0. Figure L1-1 shows the NOT gate symbols together with its truth table. The NOT gate Boolean expressions is

\[ X = \overline{A} \]

Procedure

1.1 Construct the circuit following the Figure L1-2.

1.2 Apply the supply voltage +5V to IC. Apply +5V to input of NOT gate and measure the output voltage with the multimeter in DC volt range.

\[ V_o = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots V \]

1.3 Connect the NOT gate to ground and measure the output voltage again.

\[ V_o = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots V \]

Figure L1-1 shows the NOT gate symbols and truth table.

Figure L1-2 The NOT gate experiment circuit for step 1.1 to 1.3
1.4 Construct the experiment circuit in Figure L1-3 and do it following step 1.2 and 1.3. Measure the output voltage: \( V_o \)

\[
\begin{align*}
V_o &= \text{........................................ V (Input = 5V)} \\
V_o &= \text{........................................ V (input = 0V)}
\end{align*}
\]

1.5 Construct the experiment circuit in Figure L1-4

1.6 Adjust VR1 to apply the Vin below. Use the multimeter measure the Vin everytime to change the value. Measure the output voltage.

<table>
<thead>
<tr>
<th>Vin</th>
<th>Output voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1V</td>
<td>..................</td>
</tr>
<tr>
<td>+3V</td>
<td>..................</td>
</tr>
<tr>
<td>+5V</td>
<td>..................</td>
</tr>
<tr>
<td>+7V</td>
<td>..................</td>
</tr>
<tr>
<td>+9V</td>
<td>..................</td>
</tr>
</tbody>
</table>
Experiment 2
OR Gate with AND gate

Part list
1. 7408 x 1
2. 7432 x 1

Theory

**OR gate** is a Logic 1 output is produced if any input is at Logic 1. For a gate with two inputs A and B the Boolean expression would be:

\[ X = A + B \]

\[ X = A \lor B \]

with the plus sign (+) denoting the OR function, showing that the output Q = ‘1’ if any of the inputs A, B are at ‘1’. Figure L2-1 shows the OR gate symbols together with its truth table.

**AND gate** gives a Logic 1 output when all the inputs are at Logic 1. The Boolean expression for a two input gate would be:

\[ X = A \cdot B \]

\[ X = A \land B \]

Figure L2-2 shows the OR gate symbols together with its truth table.

The sign + means OR function and • means AND function. In IEC standard would be use \lor for OR function and \land for AND function.
Procedure

**OR gate**

2.1 Construct the circuit in Figure L2-3. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L2-3.

2.2 Construct the circuit in Figure L2-4. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L2-4.

![Figure L2-3](image1)

![Figure L2-4](image2)
AND gate

2.3 Construct the circuit in Figure L2-5. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L2-5.

2.4 Construct the circuit in Figure L2-6. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L2-6.

Figure L2-5 The AND gate experiment circuit for step 2.3

Figure L2-6 The AND gate experiment circuit for step 2.4
Experiment 3
NOR gate - The Universal gate

Part list
7402   x 1

Theory

This is the inverse of the OR gate, i.e. it will give an output that is the opposite of that gate. The Boolean expression for a two input gate will be:

\[ X = \overline{A + B} \]
\[ X = \overline{A \lor B} \]

The bar over the top indicates that the OR function is inverted, i.e. a NOT OR function, showing that if any is at ‘1’ then Q = ’0’. The Figure L3-1 shows the NOR gate symbols together with its truth table.

If connect both inputs of NOR gate together, NOR function will change to NOT function following the Figure L3-2.

NOR to OR gate conversion is connect 2 NOR gates. One acts simple NOR gate, another acts NOT gate following the Figure L3-3. The Boolean expression for this conversion will be:

\[ X = \overline{A + B} = A + B \]
\[ X = \overline{A \lor B} = A \lor B \]

Figure L3-1 shows the NOR gate symbols with truth table.

Figure L3-2 Conversion diagram from NOR to NOT gate.
How to convert NOR to AND gate would be use 3 NOR gates. First 2 gates act NOT gate and the last one acts NOR gate following the Figure L3-4. The Boolean expression for this conversion will be:

\[ X = A + B = A \cdot B \]
\[ X = A \lor B = A \land B \]

Thus, sometime we can call the NOR gate to The Universal Gate.

Procedure

3.1 Construct the circuit in the Figure L3-5. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L3-5.

From the result, the experiment circuit is .................. gate operation.

![Figure L3-5 The NOR gate experiment circuit for step 3.1](image)

![Figure L3-6 The NOR gate experiment circuit for step 3.2](image)
3.2 Construct the circuit in the Figure L3-6. Record the result in the Figure L3-6

*From the result, the experiment circuit is ....................... gate operation.*

3.3 Construct the circuit in the Figure L3-7. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L3-7

*From the result, the experiment circuit is ....................... gate operation.*

![Figure L3-7 The NOR gate experiment circuit for step 3.3](image)

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3.4 Construct the circuit in the Figure L3-8. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L3-8

*From the result, the experiment circuit is ....................... gate operation.*

![Figure L3-8 The NOR gate experiment circuit for step 3.4](image)

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Experiment 4

**NAND gate - The Universal gate**

#### Part list

| 7400 | x 1 |

#### Theory

NAND is the inverse of the AND gate giving an output that is the opposite of the AND function. A two input gate may be expressed as:

\[
X = A \cdot B \\
X = A \land B
\]

The Figure L4-1 shows the NOR gate symbols together with its truth table.

NAND gate can convert to all simple logic gates similar NOR gates as follows:

- **NOT gate conversion** with connect both inputs together followign the Figure L4-2
- **AND gate conversion** connect 2 NAND gates. One acts simple NAND gate, another acts NOT gate following the Figure L4-3. The Boolean expression for this conversion will be:

\[
X = A \cdot B = A \cdot B \\
X = A \land B = A \land B
\]

![Figure L4-1](image1.png)  
![Figure L4-2](image2.png)

Figure L4-1 shows the NAND gate symbols with truth table.  
Figure L4-2 Conversion diagram from NAND to NOT gate.
OR gate conversion would be use 3 NAND gates. First 2 gates act NOT gate and the last one acts NAND gate following the Figure L4-4. The Boolean expression for this conversion will be:

\[
X = \overline{A \cdot \overline{B}} = A + B \\
X = \overline{A} \land \overline{B} = A \lor B
\]

Thus, sometime we can call the NAND gate to The Universal Gate similar NOR gate.

**Procedure**

4.1 Construct the circuit in the Figure L4-5. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L4-5

*From the result, the experiment circuit is ...................... gate operation.*
4.2 Construct the circuit in the Figure L4-6. Record the result in the Figure L4-6

*From the result, the experiment circuit is ...................... gate operation.*

![NAND gate circuit](image1)

**Figure L4-6 The NAND gate experiment circuit for step 4.2**

4.3 Construct the circuit in the Figure L4-7. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L4-7

*From the result, the experiment circuit is ...................... gate operation.*

![NAND gate Truth table](image2)

**Figure L4-7 The NAND gate experiment circuit for step 4.3**

4.4 Construct the circuit in the Figure L4-8. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in Figure L4-8

*From the result, the experiment circuit is ...................... gate operation.*

![NAND gate Truth table](image3)
Figure L4-8 The NAND gate experiment circuit for step 4.4
Chapter 5
Exclusive-OR gate

Part list
1. 7408  x 1
2. 7404  x 1
3. 7432  x 1
4. 7486  x 1

Theory

The Exclusive OR (XOR) gate is like the OR gate except that it produces an output only if one of the two inputs is at Logic "1". If both are at Logic "1" the output will be Logic "0". The Boolean express is:

\[ X = \overline{A}B + AB = A \oplus B \]

The Figure L5-1 shows the Exclusive-OR gate symbols together with its truth table.

Procedure.

5.1 Construct the circuit in the Figure L5-2. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L5-2.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure L5-1 shows the Exclusive-OR gate symbols with truth table.
5.2 Construct the circuit in the Figure L5-3. The main IC is Exclusive-OR gate function. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L5-3.

5.3 Compare the operation result between circuit in Figure L5-2 and L5-3.

The comparison is  🟥 same  🟢 different
Experiment 6
Exclusive-NOR gate

Part list
1. 7404 x 1
2. 7486 x 1

Theory
The Exclusive NOR (XNOR) gate is the inverse of Exclusive-OR gate. It produces an output at Logic "1" if both input are same logic. The output will be logic "0" if both are different logic. The Boolean express is:

\[ X = \overline{AB} + AB = A \oplus B \]

The Figure L6-1 shows the Exclusive-NOR gate symbols together with its truth table.

Procedure
6.1 Construct the circuit in the Figure L6-2. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L6-2.

Figure L6-1 shows the Exclusive-NOR gate symbols with truth table.

Figure L6-2 The experiment circuit for step 6.1
6.2 Construct the circuit in the Figure L6-3 to extend input of circuit to 3 inputs. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L6-3.

Figure L6-2 The experiment circuit for step 6.2
6.3 Construct the circuit in the Figure L6-4 to extend input of circuit to 4 inputs. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L6-4.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure L6-2 The experiment circuit for step 6.2
6.4 This step is very important. Compare the circuit operation of Exclusive-NOR gate 2, 3 and 4 inputs. These are question:

6.4.1 From the experiment circuit L6-3, it is 3 inputs circuit. The circuit function is 3-input Exclusive-NOR gate TRUE or False?

6.4.2 From the experiment circuit L6-4, it is 4 inputs circuit. The circuit function is 4-input Exclusive-NOR gate TRUE or False?

6.4.3 Can you summary about the number of input of the Exclusive-NOR gate and How?
Experiment 7
Boolean algebra

Part list
1. 7408 x 1
2. 7411 x 1
3. 7432 x 1

Theory

Basic concept of Boolean logic

There are two main methods of reducing Boolean expressions to their simplest forms, one involves the already mentioned Boolean algebra and the other uses Mapping techniques. It is very important to realize that care taken to develop minimization skills at this stage will be amply rewarded later on by the ease with which it will be possible to interpret and design logic circuitry.

Boolean Algebra has 3 basic concepts as:

(1) Complementation inversion
(2) AND multiplication (., or dot)
(3) OR Addition (+ or plus)

The summary of Boolean Logic are

<table>
<thead>
<tr>
<th>Complement</th>
<th>AND</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 = 1</td>
<td>0 • 0 = 0</td>
<td>0 + 0 = 0</td>
</tr>
<tr>
<td>1 = 0</td>
<td>0 • 1 = 0</td>
<td>0 + 1 = 1</td>
</tr>
<tr>
<td>A • A = 0</td>
<td>1 • 0 = 0</td>
<td>1 + 0 = 1</td>
</tr>
<tr>
<td>A + A = 1</td>
<td>1 • 1 = 1</td>
<td>1 + 1 = 1</td>
</tr>
</tbody>
</table>

Remember Logic 0 (‘0’) is not the same as mathematical 0 (ZERO) similarly Logical 1 (‘1’) is not the same as mathematical 1.
Summary of Boolean logic law

1. Commutation
   \[ A + B = B + A \]
   \[ A \cdot B = B \cdot A \]

2. Association
   \[ A + (B + C) = (A + B) + C \]
   \[ A \cdot (B \cdot C) = (A \cdot B) \cdot C \]

3. Distribution
   \[ A \cdot (B + C) = (A \cdot B) + (A \cdot C) \]
   \[ A + (B \cdot C) = (A + B) \cdot (A + C) \]

4. Absorption
   \[ A + (A \cdot B) = A \]
   \[ A \cdot (A + B) = A \]

5. Annulment
   \[ A + 1 = 1 \]
   \[ A \cdot 0 = 0 \]

6. Identity
   \[ A + 0 = A \]
   \[ A \cdot 1 = A \]

7. Tautology
   \[ A \cdot A = A \]
   \[ A + A = A \]

8. Double negation
   \[ = \]
   \[ A = A \]
**De Morgan’s Theorem**

A more comprehensive procedure for DeMorgans is as follows:

1. Change the overall polarity of each of the expression thus $A.B$ becomes $\overline{A \cdot B}$.

2. Then change the polarity of each of the grouped terms to give $\overline{A}$.

3. Finally change the linking logic $\overline{A + B}$. Thus $A.B = \overline{A \cdot B}$.

The Boolean logic expression of De Morgan’s theorem is:

\[
\overline{A \cdot B} = \overline{A} + \overline{B} \\
\overline{A + B} = \overline{A} \cdot \overline{B}
\]

For example: From the Boolean expression therefore

\[X = A + \overline{A \cdot B}\]

If construct the circuit from this expression, must use 3 logic gates following the Figure L7-1. With the Distribution law, the expression will be change to:

\[X = (A + \overline{A}) \cdot (A + B)\]

Next step, use Complement concept $A + \overline{A} = 1$. The result is

\[X = A + B\]

It can reduce the number of logic gate to only an OR gate. The comparison below is show the result of operation between normal and after simplified. The Figure L7-1 shows the logic diagram before and after simplified.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>$\overline{A}$</th>
<th>$\overline{A \cdot B}$</th>
<th>$X = A + \overline{A \cdot B}$</th>
<th>$X = A + B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure L7-1 shows the logic circuit before simplify use 3 logic gates. After simplified, the number of logic gate reduce to only one OR gate.
Procedure

7.1 Construct the circuit in the Figure L7-2. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L7-2.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure L7-2 The experiment circuit for step 7.1
7.2 Construct the circuit in the Figure L7-3. Apply the input with LOGIC SWITCH. The output is connected with LOGIC MONITOR on the experiment board. Record the result in the output table in the Figure L7-3.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure L7-3 The experiment circuit for step 7.2

7.3 From the experiment circuit in Figure L7-2, make the Boolean expression.

$$X = \ldots$$

7.4 Simplify the Boolean expression from step 7.3 with Boolean algebra. Write with step by step below.

7.5 From the experiment circuit in Figure L7-3, make the Boolean expression and compare with the simplified Boolean expression from step 7.4

$$X = \ldots$$

The comparison of Boolean expression is  

- same  
- different