Digital Electronics

Class 10

Lab 18

UNITED TRUST

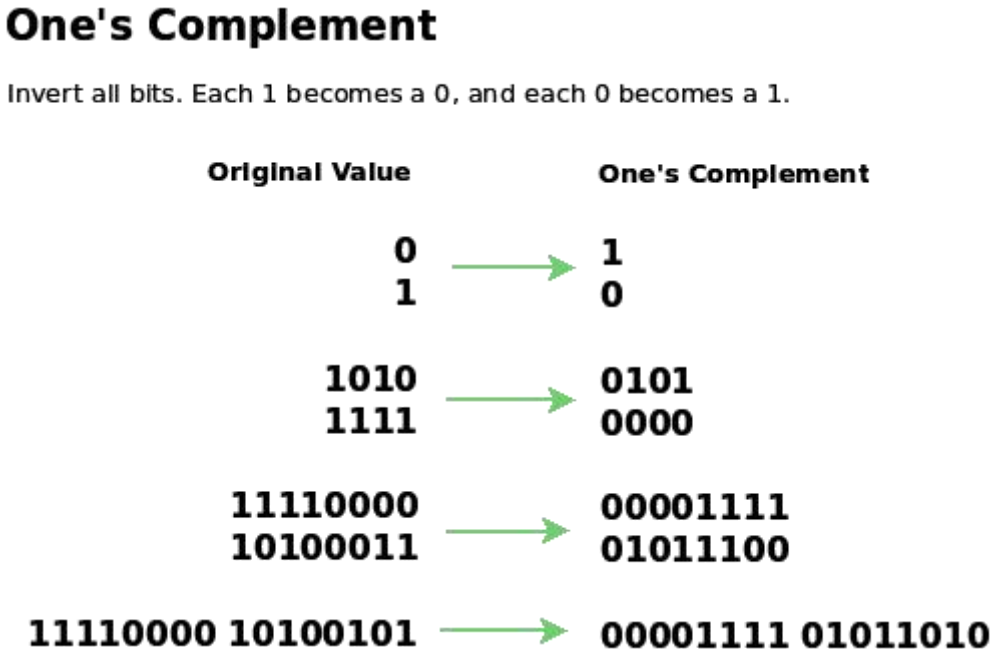
bdeducation
Bangladesh Digital Education Research Limited
সময়ের পাঠশালা

Lab Objectives:

- Number Complement
- 1's complement
- 2's complement

1's complement

1's complement of a binary number is another binary number obtained by toggling all bits in it, i.e., transforming the 0 bit to 1 and the 1 bit to 0.



Examples:

Let numbers be stored using 4 bits

1's complement of 7 (0111) is 8 (1000)

1's complement of 12 (1100) is 3 (0011)

2's complement

2's complement of a binary number is 1 added to the 1's complement of the binary number.

Example #1

```
 5 = 00000101
     ↓↓↓↓↓↓↓↓   } Complement Digits
     11111010
           +1   } Add 1
     --------
-5 = 11111011
```

Example #2

```
-13 = 11110011
      ↓↓↓↓↓↓↓↓   } Complement Digits
      00001100
            +1   } Add 1
      --------
 13 = 00001101
```

Let numbers be stored using 4 bits

2's complement of 7 (0111) is 9 (1001)

2's complement of 12 (1100) is 4 (0100)

Difference

These representations are used for signed numbers.

The main difference between 1′ s complement and 2′ s complement is that 1′ s complement has two representations of 0 (zero) – 00000000, which is positive zero (+0) and 11111111, which is negative zero (-0);

whereas in 2′ s complement, there is only one representation for zero – 00000000 (+0) because if we add 1 to 11111111 (-1), we get 00000000 (+0) which is the same as positive zero. This is the reason why 2′ s complement is generally used.

Another difference is that while adding numbers using 1′ s complement, we first do binary addition, then add in an end-around carry value. But, 2′ s complement has only one value for zero, and doesn’t require carry values.

Boolean Postulates

Assume **A**, **B**, and **C** are logical states that can have the values **0** (false) and **1** (true).
"+" means **OR**, **"·"** means **AND**,
and **NOT[A]** means **NOT A**.

Consider the binary numbers 0 and 1, Boolean variable (x) and its complement (x'). Either the Boolean variable or complement of it is known as literal. The four possible logical OR operations among these literals and binary numbers are shown below.

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + x' = 1$$

Similarly, the four possible logical AND operations among those literals and binary numbers are shown below.

$$x.1 = x$$

$$x.0 = 0$$

$$x.x = x$$

$$x.x' = 0$$

These are the simple Boolean postulates. We can verify these postulates easily, by substituting the Boolean variable with '0' or '1'.

Postulates and Theorems of Boolean Algebra

Postulates

(1)	A + 0 = A	A · 1 = A	identity
(2)	A + ^{NOT}[A] = 1	A · ^{NOT}[A] = 0	complement
(3)	A + B = B + A	A · B = B · A	commutative law
(4)	A + (B + C) = (A + B) + C	A · (B · C) = (A · B) · C	associative law
(5)	A + (B · C) = (A + B) · (A + C)	A · (B + C) = (A · B) + (A · C)	distributive law

Theorems

(6)	A + A = A	A · A = A	
(7)	A + 1 = 1	A · 0 = 0	
(8)	A + (A · B) = A	A · (A + B) = A	
(9)	A + (^{NOT}[A] · B) = A + B	A · (^{NOT}[A] + B) = A · B	
(10)	(A · B) + (^{NOT}[A] · C) + (B · C) = (A · B) + (^{NOT}[A] · C)	A · (B + C) = (A · B) + (A · C)	
(11)	^{NOT}[A + B] = ^{NOT}[A] · ^{NOT}[B]	^{NOT}[A · B] = ^{NOT}[A] + ^{NOT}[B]	de Morgan's theorem