AE 242
Aerospace Measurements Laboratory

Digital Electronics

Number system

$$(N)_b = d_{n-1}d_{n-2}...d_i....d_1d_0 \cdot d_{-1}d_{-2}...d_{-f}...d_{-m}$$

N - a number

b - radix or base of the number system

n - number of digits in integer portion

m - number of digits in fractional portion

d_{n-1} - most significant digit (msd)

d_{-m} - least significant digit (lsd)

Decimal number can be obtained by multiplying d_i and d_{-f} by their weights and adding the terms. Weights are equal to b^i for the terms on the left of radix point and b^{-f} for the terms on the right of radix point.

Binary number system

Number with a radix two or base two

only two numbers (0 and 1) are used to represent numbers

Binary to decimal conversion

$(11111)_2 = 1x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = (31)_{10}$

$(1100.1011)_2 = 1x2^3 + 1x2^2 + 0x2^1 + 0x2^0 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 1x2^{-4} = (12.6875)_{10}$

Signed binary numbers

In decimal system + is used to represent positive number and - is used to represent negative number

In binary system left most bit, most significant bit (MSB) is used to represent the sign. 1 means negative and 0 means positive

Unsigned binary form $(1000100)_2 = (68)_{10}$

Signed binary form $(11000100)_2 = (-68)_{10}$ (eight bit number one bit reserved for sign)

Signed binary form $(01000100)_2 = (68)_{10}$ (eight bit number one bit reserved for sign)

Maximum representation in unsigned 8 bit binary number is 255

Maximum representation in signed 8 bit binary number is 127

One's complement representation

In a binary number, if 1 is replaced by 0 and 0 by 1, the resulting number is one's complement of the original number.

If one of these number is positive, then the other number will be negative with the same magnitude. Widely used for representing the signed numbers. MSB 0 means positive number, 1 means negative number.

$(+15)_{10} = (01111)_2$ and $(-15)_{10} = (10000)_2$

For n bit number, maximum positive number in one's complement number is $(2^{n-1} - 1)$ and maximum negative number is $-(2^{n-1} - 1)$

All the bits are complemented including sign bit in case of one's complement representation of a negative number

Signed magnitude representation of $(-15)_{10} = (11111)_2$

Only the sign bit is complemented in case of signed magnitude representation of a negative number

Two's complement representation

If 1 is added to one's complement of a binary number, the resulting number is a two's complement. The maximum positive number is +7 and maximum negative number is -8. 2's complement of 2's complement of a number is a number itself.

Decimal number	*Binary number*		
	Sign-magnitude	*One's complement*	*Two's complement*
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111
– 8	—	—	1000
– 7	1111	1000	1001
– 6	1110	1001	1010
– 5	1101	1010	1011
– 4	1100	1011	1100
– 3	1011	1100	1101
– 2	1010	1101	1110
– 1	1001	1110	1111
– 0	1000	1111	—

Two's complement representation

Number	0 1 0 0 1 1 1 0
One's complement	1 0 1 1 0 0 0 1
Add 1	1
Two's complement	1 0 1 1 0 0 1 0

If the LSB of the number is 0, its 2's complement is obtained by scanning the number from LSB to MSB bit by bit and retaining the bits as they are up to and including the occurrence of the first 1 and complement all other bits

Number	0 0 1 1 0 1 0 1
One's complement	1 1 0 0 1 0 1 0
Add 1	1
Two's complement	1 1 0 0 1 0 1 1

If the LSB of the number is 1. It's 2's complement is obtained by changing each 0 to 1 and 1 to 0 except the least significant bit

Binary arithmetic

Binary addition

Augend	Addend	Sum	Carry	Result
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	0	1	10

Binary number addition

```
   0  1  0  1
+  1  1  1  1
-------------
1  0  1  0  0
```

Binary arithmetic

Binary subtraction

Minuend	Subtrahend	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Binary number subtraction

	1	0	1	1	Minuend
-	0	1	1	0	Subtrahend
	0	1	0	1	Difference

Binary arithmetic

Binary multiplication : Similar to decimal multiplication

```
      1  0  0  1      Multiplicand
    x 1  1  0  1      Multiplier
    ------------
      1  0  0  1
   0  0  0  0
1  0  0  1
1  0  0  1          .
-------------------
1  1  1  0  1  0  1   Final Product
```

In digital circuit, the multiplication operation is performed by repeated additions of all partial products to obtain full product.

Binary arithmetic

Binary division : Similar to decimal division

```
                 1101     Quotient
Divisor  1001)1110101  Dividend
              1001
                1011
                1001
                   1001
                   1001
                      0000
```

Answer : 1101

2’s Complement arithmetic

Problem of subtraction can be converted in to a addition problem. This eliminates additional circuit for subtraction.

Subtraction using 2’s complement

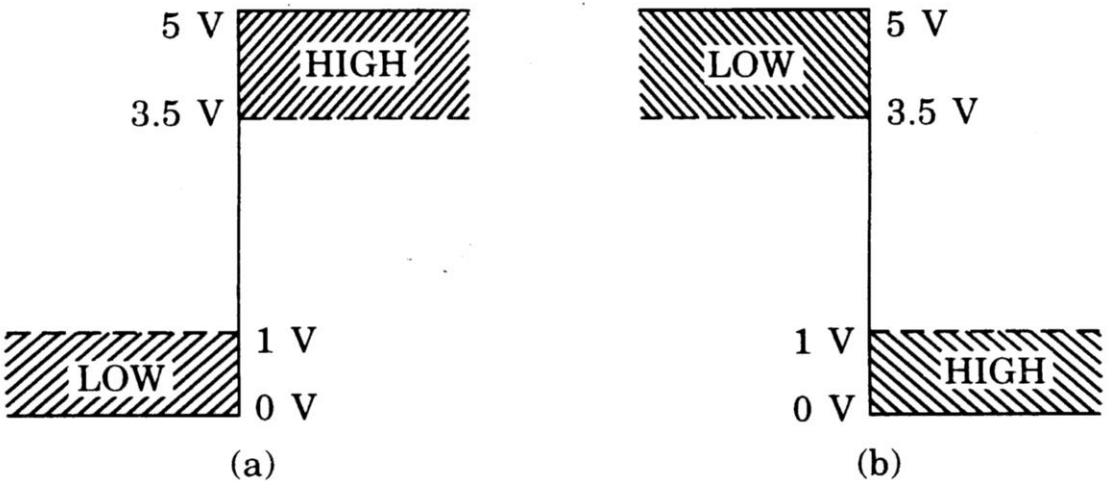
Binary subtraction can be performed by adding the 2’s complement of the subtrahend to the minuend. If the final carry is generated, discard the carry and the answer is remaining bits which is positive (minuend > subtrahend). If the final carry is 0, the answer is negative (minuend < subtrahend) and it is in 2’s complement.

7	0 1 1 1	Minuend
-5	1 0 1 1	2’s Complement
+2	1 0 0 1 0	

5	0 1 0 1	Minuend
-7	1 0 0 1	2’s Complement
-2	1 1 1 0	

Digital signal

Digital system only two discrete levels or values, low (0) or high (1). In case of positive logic, high is from 3.5-5v and low is 0-1v. In case of negative logic high is 0-1v and low is 3.5-5v. As long as voltage remains in these levels the state is considered low or high depending on logic used. Digital system are less susceptible to noise due to the range in logic voltages. High is also called as on and low as off.



For TTL

V_{in} = 0-0.8 (low)

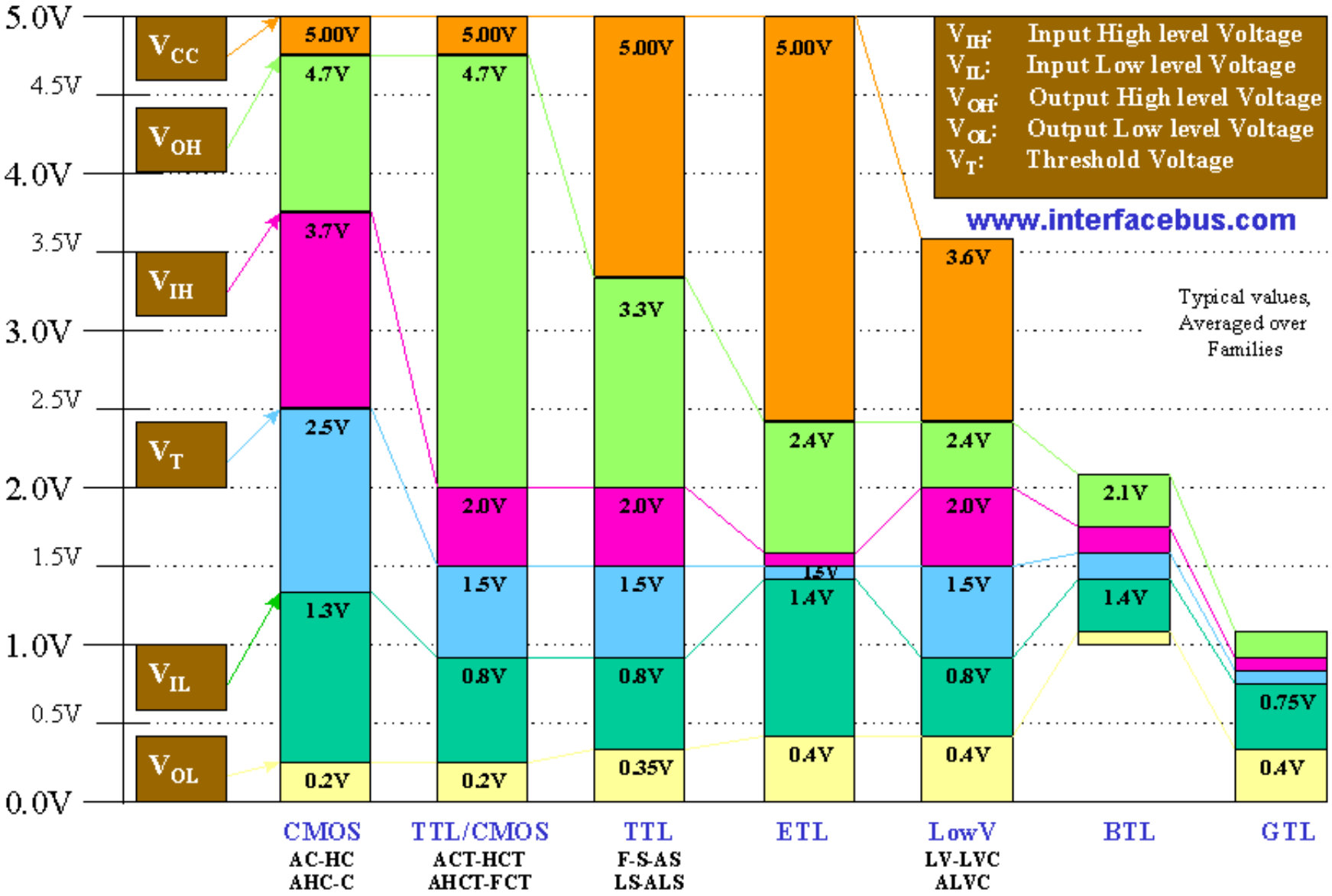
V_{in} = 2-5 (high)

V_{out} = 0-0.4 (low)

V_{out} = 2.4-5 (high)

(a) Positive logic and (b) negative logic

Digital signal

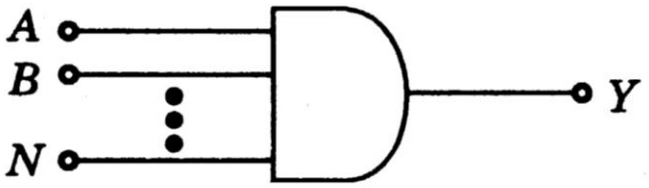


Basic digital circuits

AND operation

$$Y = A \cdot B \cdot C \cdot ... N$$

$$Y = ABC...N$$



Standard symbol for AND gate

A,B,C .. N are input variables (possible values only 0 &1) and Y is output. Y will be high only when all the inputs are high (positive logic)

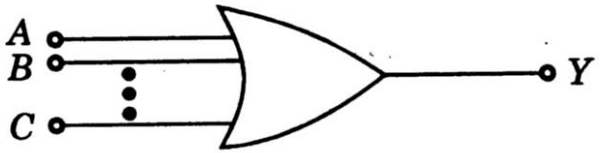
Inputs		*Output*
A	*B*	*Y*
0	0	0
0	1	0
1	0	0
1	1	1

Truth table for a 2-input AND gate

Basic digital circuits

OR operation

$$Y = A + B + C + \ldots N$$



Standard symbol for OR gate

Output of OR gate is 1, if and only if one or more inputs are 1

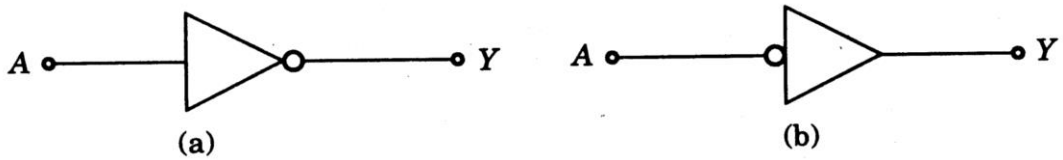
Inputs		*Output*
A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	1

Truth table for a 2-input OR gate

Basic digital circuits

NOT operation

$$Y = \overline{A}$$



Standard symbol for NOT gate

NOT gate also known as inverter. It is one input (A) and one output (Y) device. Output is complement of input. Bubble in the circuit always denotes inversion in digital circuits.

Input *A*	*Output* *Y*
0	1
1	0

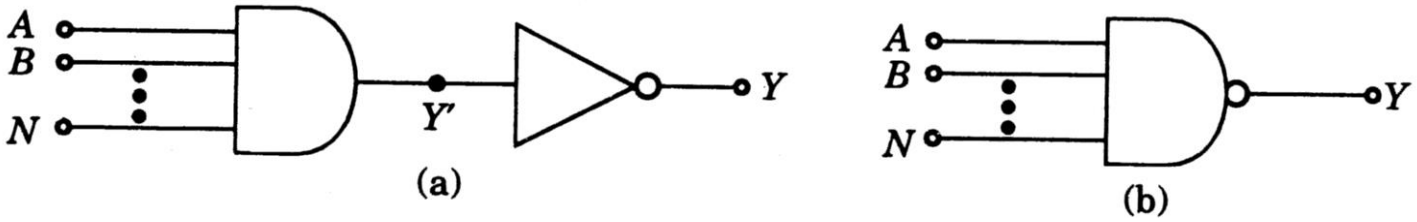
Truth table for a NOT gate

Derived gates

Basic gates in digital circuitry are AND, OR and NOT. Using these three basic gates any Boolean (logic expression) can be realized. Other type of gates obtained using basic gates are NAND, NOR, XOR, XNOR

NAND operations

NOT-AND operation is known as NAND, it is a AND gate followed by a NOT gate. Complemented output of AND. Standard symbol is AND with bubble.



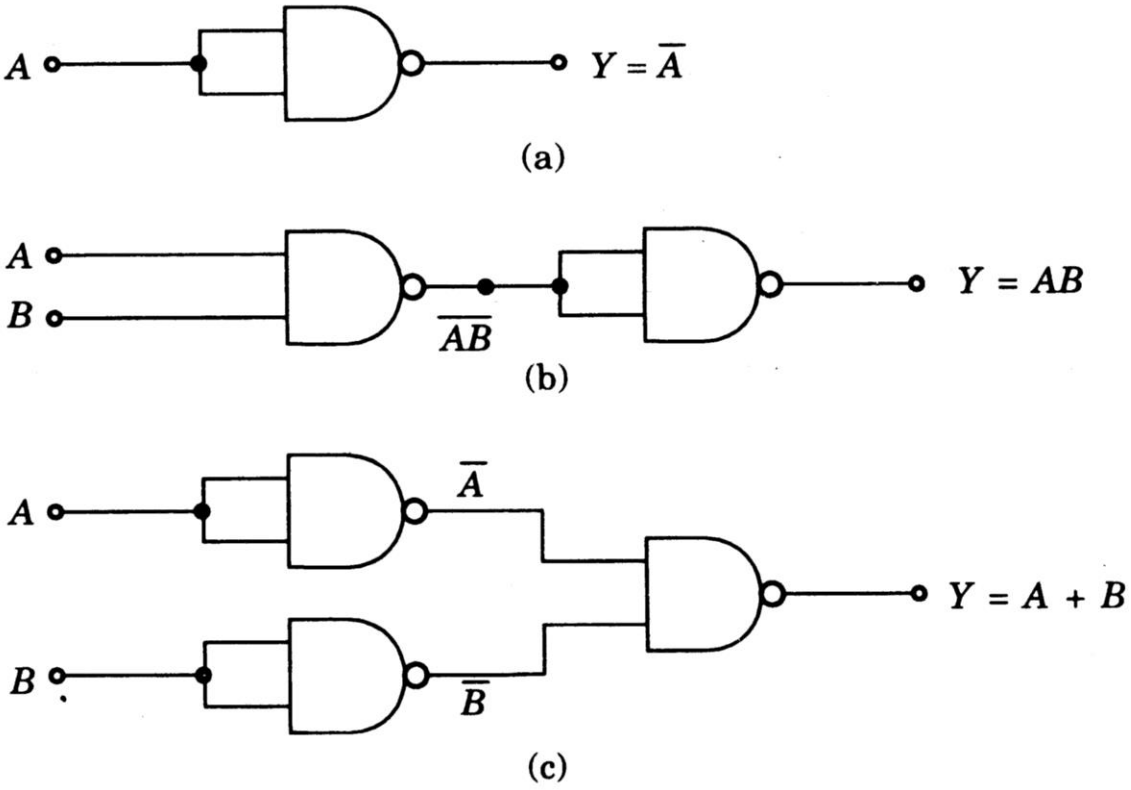
$$Y = \overline{A \cdot B \cdot C \cdot \ldots N}; Y = \overline{ABC \ldots N}$$

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Truth table for a NAND gate

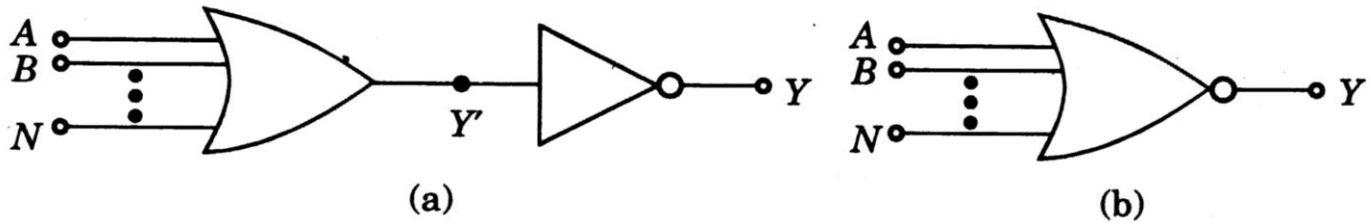
NAND as universal gate

Using NAND gate, basic digital circuits i.e. AND, OR, NOT can be obtained and this property makes it a universal gate.



NOR operations

NOT-OR operation is known as NOR. It is OR gate followed by a NOT gate. Complemented output of OR gate. Standard symbol is OR gate with a bubble.



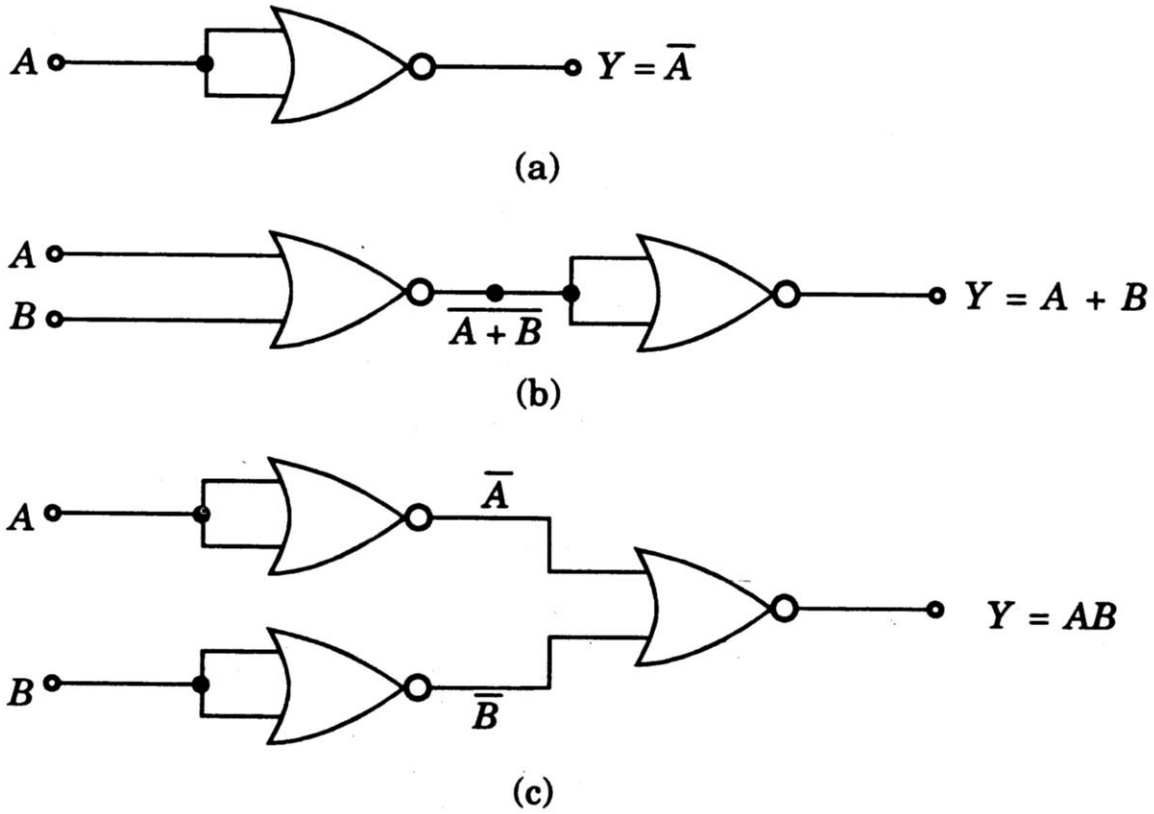
$$Y = \overline{A + B + C + \ldots N}$$

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Truth table for a two input NOR gate

Using NOR as universal gate

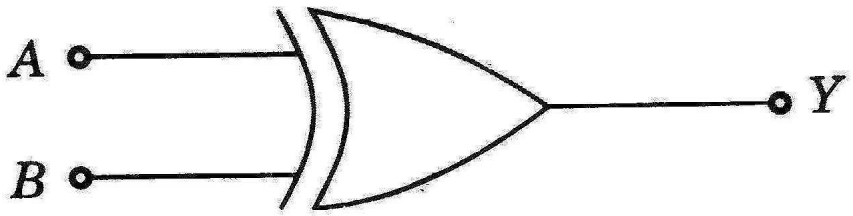
Using NOR gate, basic digital circuits i.e. AND, OR, NOT can be obtained and this property makes it a universal gate.



Exclusive-OR (EX-OR) operations

It is not a basic gate, and the operation can be performed using the basic gates - AND, OR and NOT. Output is logic one when odd number of inputs are one.

Standard notation $Y = A \oplus B$ $Y = A\overline{B} + \overline{A}B$

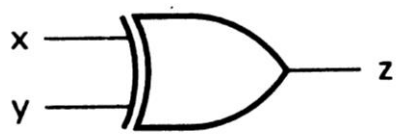


Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Truth table for a EX-OR gate

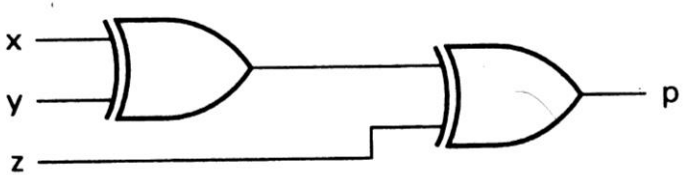
Exclusive-OR (EX-OR) operations

Truth table for three inputs. Output is logic one when odd number of inputs are one.



$z \equiv x \oplus y$

(a) 2–Input EX-OR Gate



$p = x \oplus y \oplus z$

(b) 3–Input EX-OR function derived from Two 2-input-EX-OR gates

Input			Output
x	y	z	p
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth table for a EX-OR gate

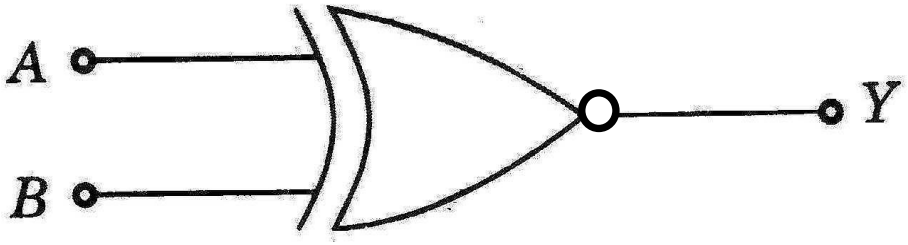
Exclusive-NOR (EX-NOR) operations

It is EX-OR gate followed by a NOT

Standard notation $Y = A \odot B$

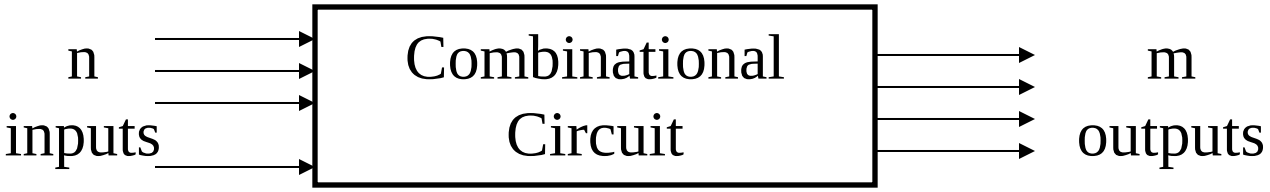
$$Y = \overline{A\overline{B} + \overline{A}B}$$

$$Y = \overline{A}\,\overline{B} + AB$$



Digital circuit classification

Mainly two type of classification in digital circuits. *Combinational logic* circuits and *Sequential logic* circuits. Combinational logic circuits only depends on the combination of the gates. Output at any time are determined from the present combination of inputs. Where as sequential logic circuit depends on the combination of gates and sequence of input, it is history dependent. Sequential circuit employs storage elements in addition to logic gates.

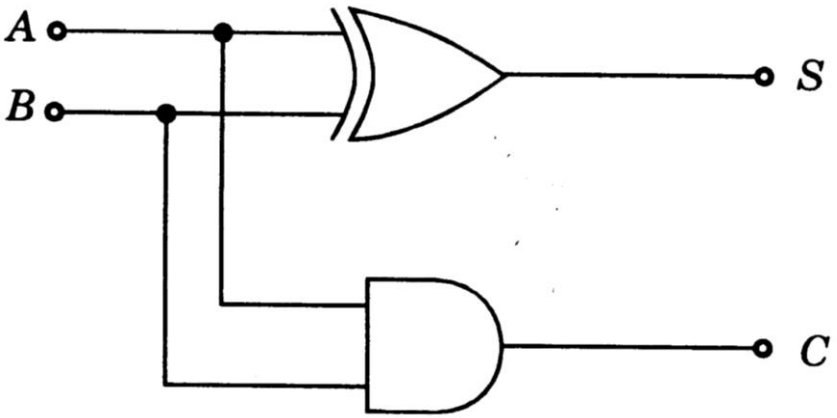


Half adder

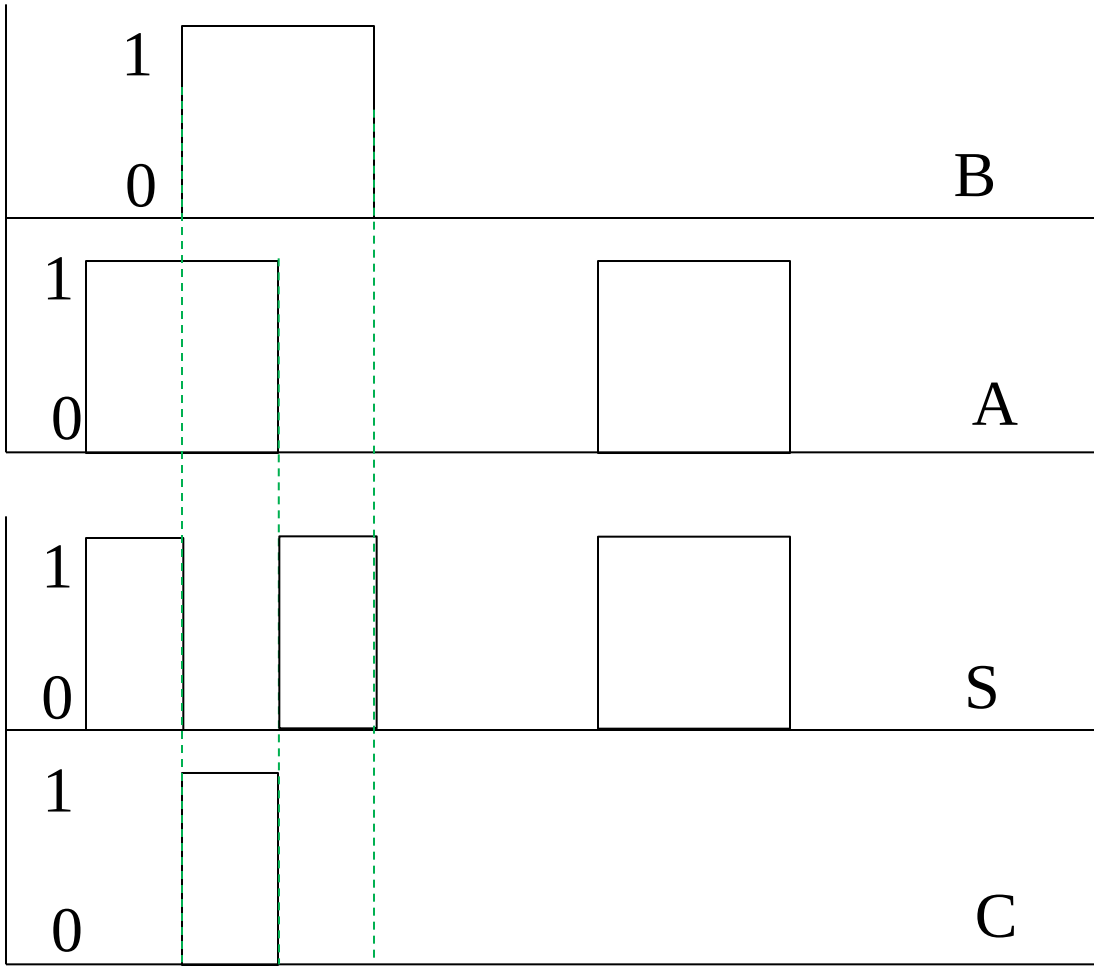
A logic circuit adding two bits is called as half adder. A and B are two inputs. S is sum and C is carry as output.

Sum $S = A \oplus B$

Carry $C = AB$



Inputs		Outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Inputs		Outputs	
A	*B*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

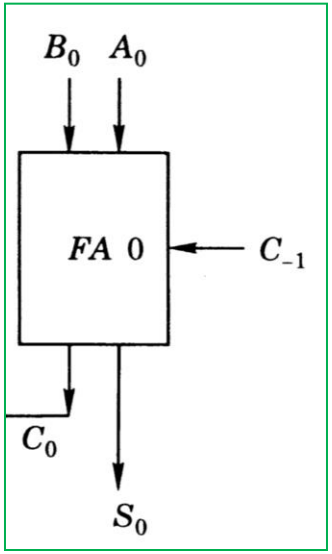
Full adder

In a full adder carry coming from low order bit is also considered. Input is A_n, B_n and C_{n-1} Output is S_n and C_n

$$S_n = \overline{A}_n B_n \overline{C}_{n-1} + \overline{A}_n \overline{B}_n C_{n-1} + A_n \overline{B}_n \overline{C}_{n-1} + A_n B_n C_{n-1}$$

$$C_n = A_n B_n + B_n C_{n-1} + A_n C_{n-1}$$

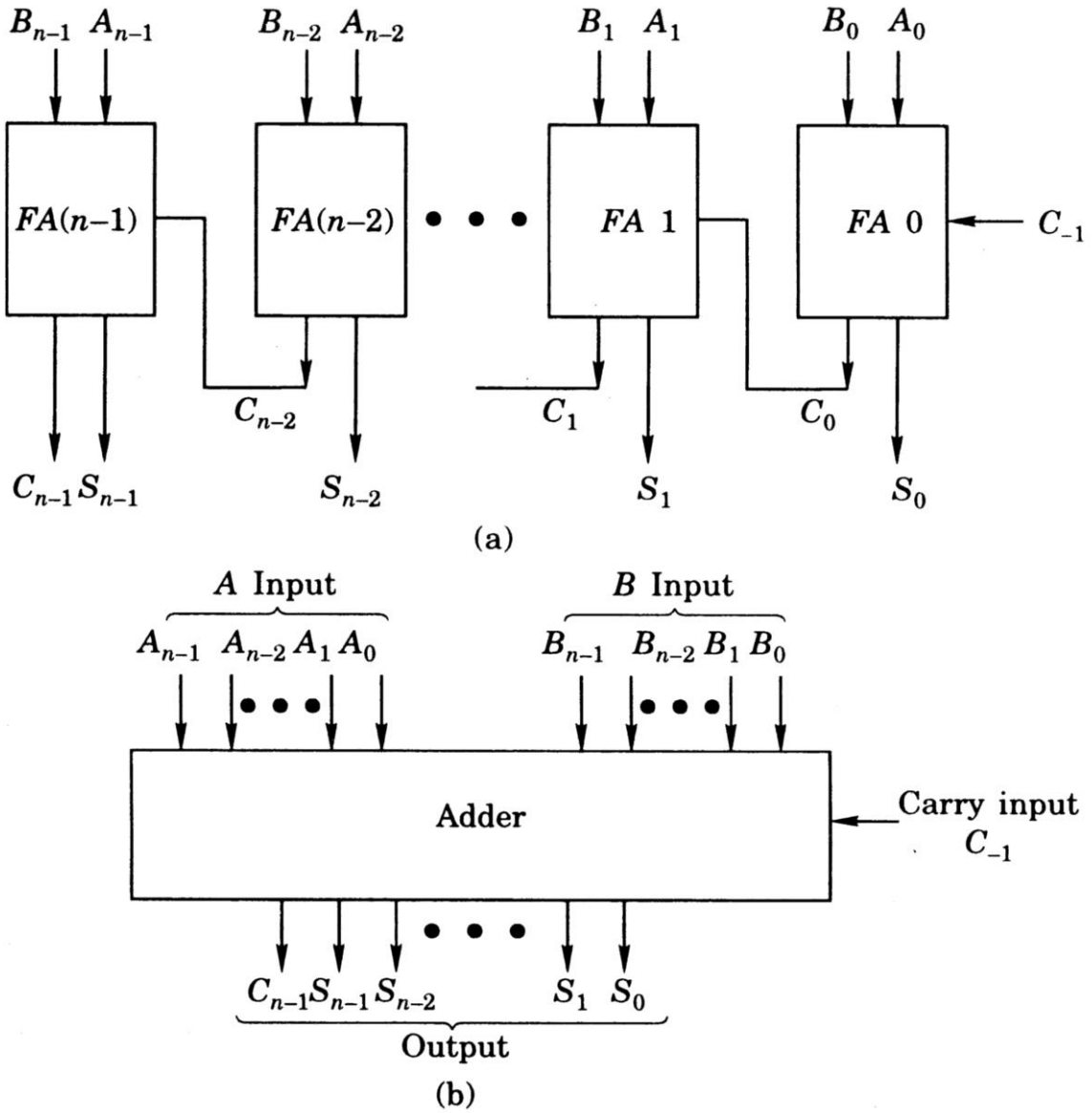
Inputs			Outputs	
A_n	B_n	C_{n-1}	S_n	C_n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



$$SUM = (A \oplus B) \oplus C_{in}$$

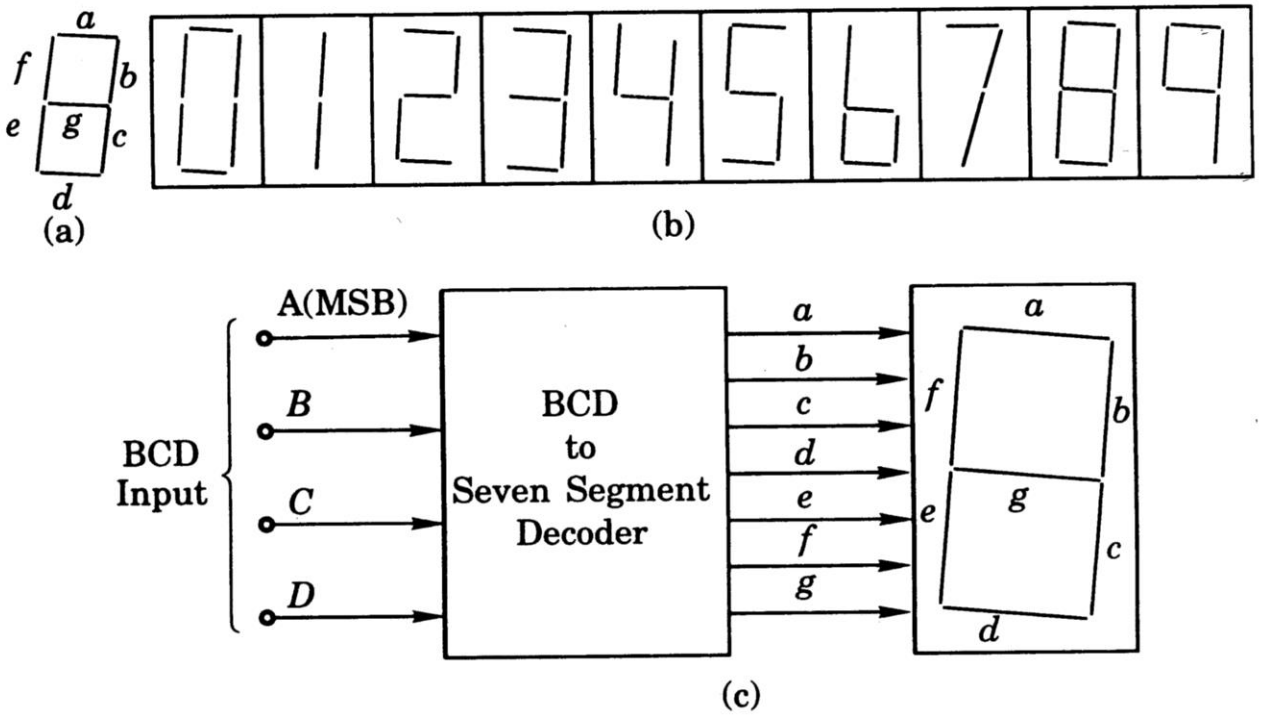
N-bit adder

An adder circuit for two n bit number consist of n full adder circuits. It accepts two n-bit number and produces output of (n+1) bit binary number as sum. Half adder may be used to add least significant bit



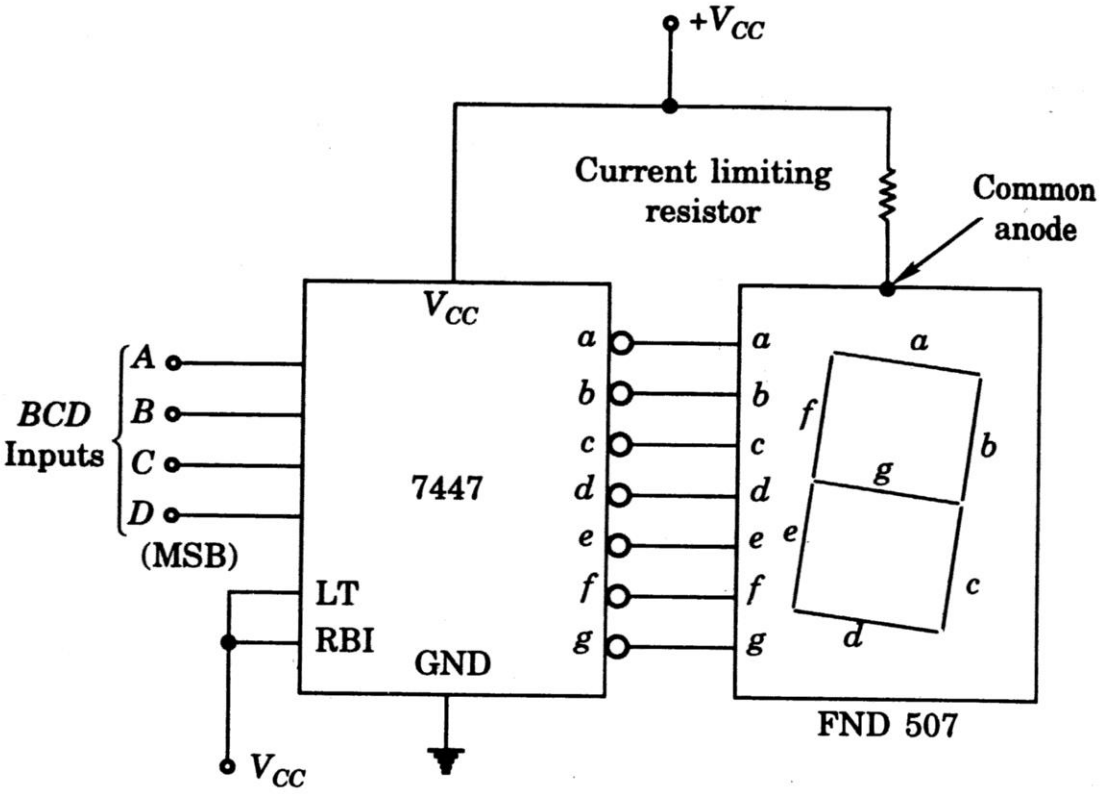
BCD-to-7-segment decoder

Digital display consisting of seven segments is commonly used and can represent all the decimal numerals. This display device requires the logic circuit to convert binary to decimal display in terms of a,b,..g segments.



BCD-to-7-segment decoder

Seven segment display available as common anode (positive) or common cathode (ground). In case of common anode, output of the decoder should be active low i.e. negative logic. Logic is available to make display blank in case of 0, useful in leading zero in an integer, for example 005.



One digit BCD adder

Four bit binary adder can be used to perform additions of BCD. Simple addition will not give correct results. If a carry is obtained in the decimal addition then the output is to be corrected by adding 6 in the second adder.

3+4 = 7	6+4 = 10	6+6 = 12
0100 (4)	0110 (6)	0110 (6)
0011 (3)	0100 (4)	0110 (6)
0111	1010	1100
0000 (0)	0110 (6)	0110 (6)
0111	**1**0000	**1**0010