Digital Electronics

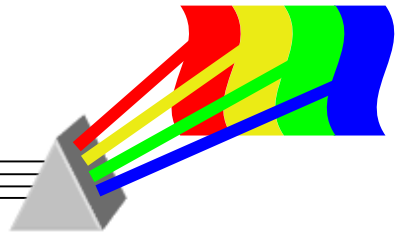
Principles & Applications

Fifth Edition

Roger L. Tokheim

Chapter 10

Arithmetic Circuits

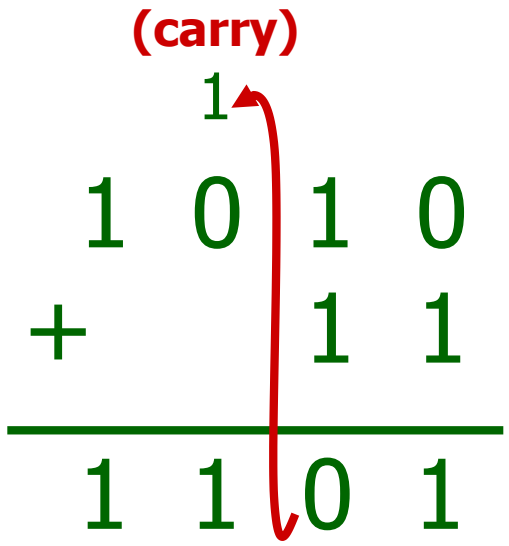


CHAPTER 10 PREVIEW

- Binary Addition
- Half & Full Adders
- Binary Subtraction
- Half & Full Subtractors
- Parallel Adders and Subtractors
- Using Adders for Subtraction
- Binary Multiplication
- Binary Multipliers
- Half & Full Adders
- 2s Complement

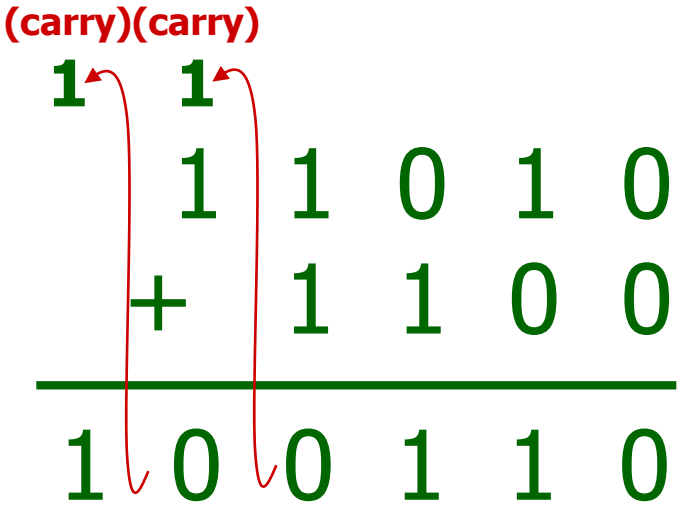
BINARY ADDITION

- **Conceptually similar to decimal addition**
- ***Example:*** **Add the binary numbers 1010 and 11**



TEST

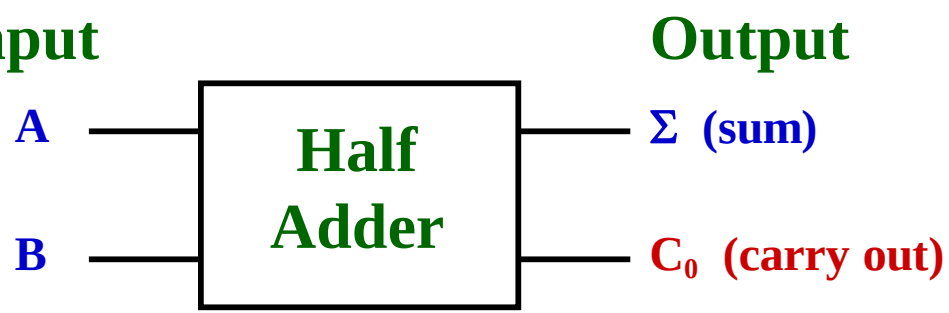
Add the Binary numbers 11010 and 1100



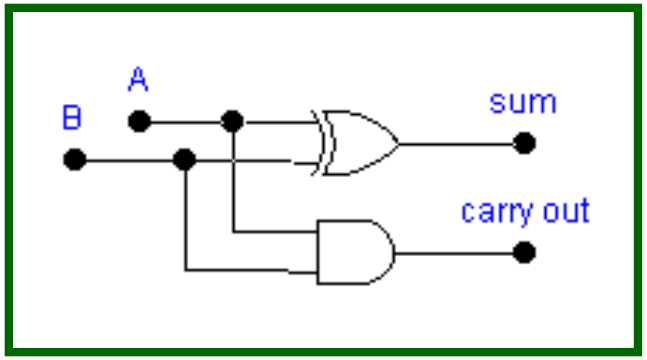
HALF ADDER

- **Logic device that adds two binary numbers**
- **Only adds Least Significant Digit (LSD) column (1s column) in binary addition**

Logic Symbol:



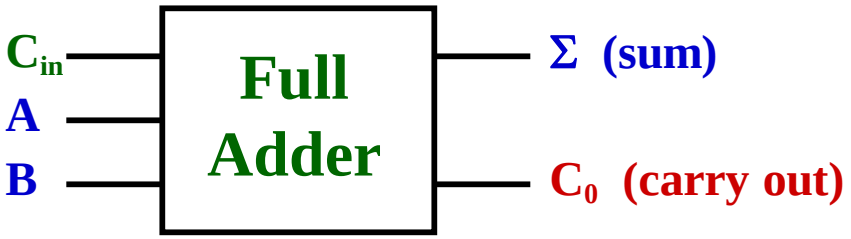
Logic Diagram:



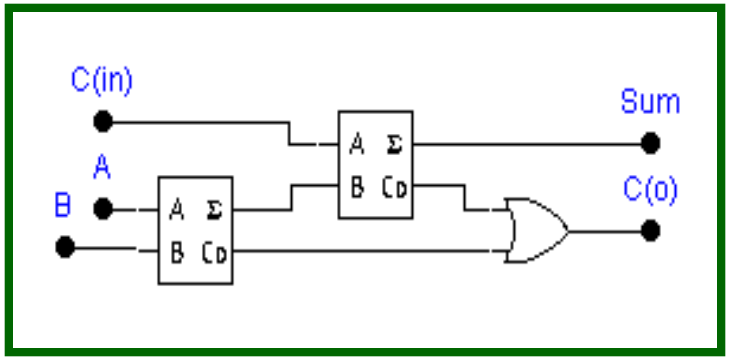
FULL ADDER

Used for adding binary place values other than the 1s place

Logic Symbol:



Logic Diagram:

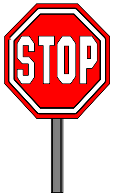


BINARY SUBTRACTION

Example: **Subtract binary number 101 from 1011**

(borrow)

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



TEST

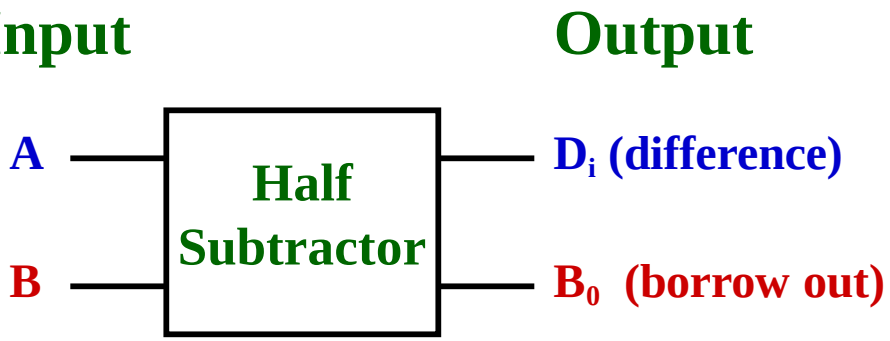
Subtract binary number 11 from 1010

```
      01
  0   10 10 1
  1   0  1  0
-        1  1
-------------
  0   1  1  1
```

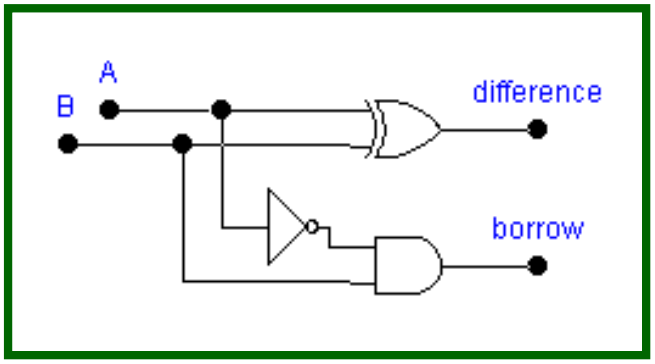
HALF SUBTRACTOR

Subtracts LSD column in binary subtraction

Logic Symbol:



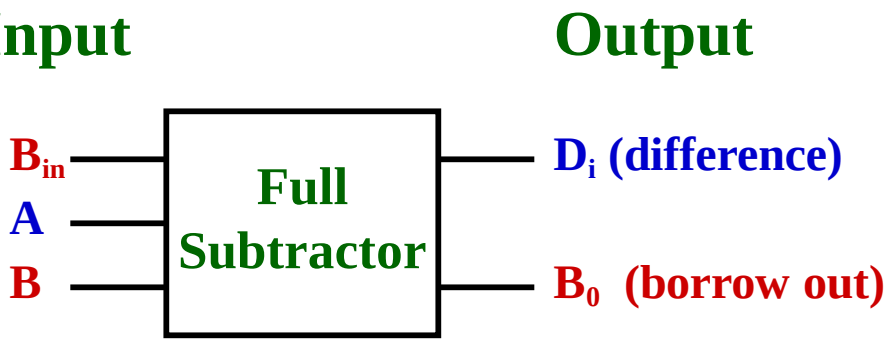
Logic Diagram:



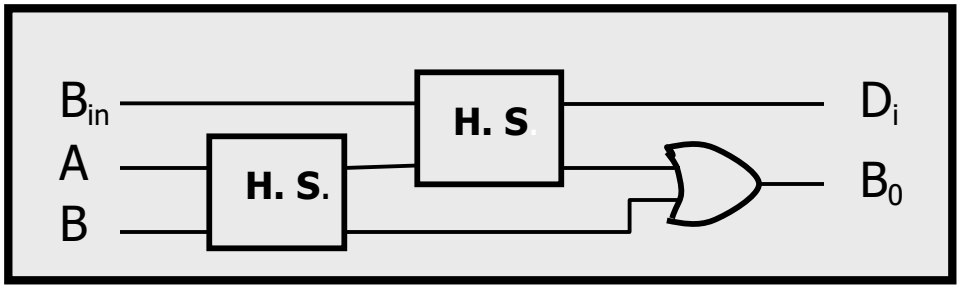
FULL SUBTRACTOR

Used for subtracting binary place values other than the 1s place

Logic Symbol:

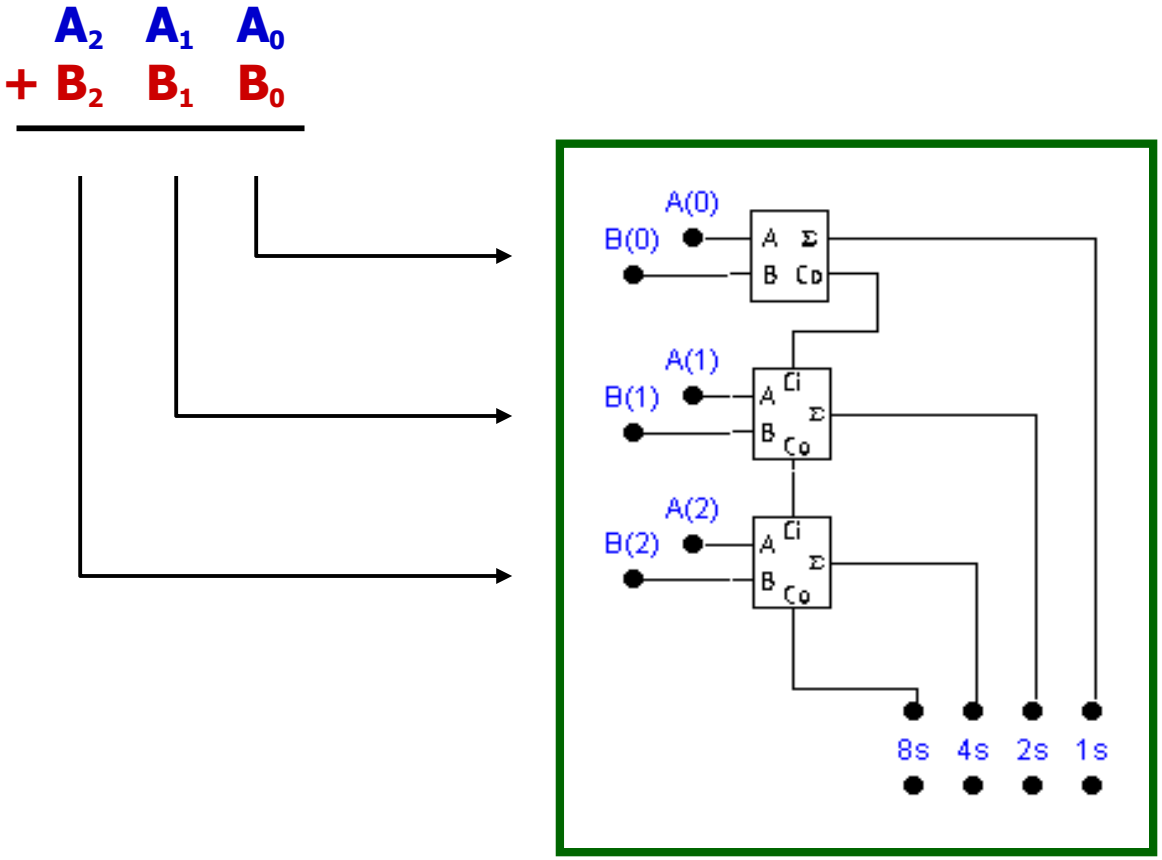


Logic Diagram:



PARALLEL ADDING

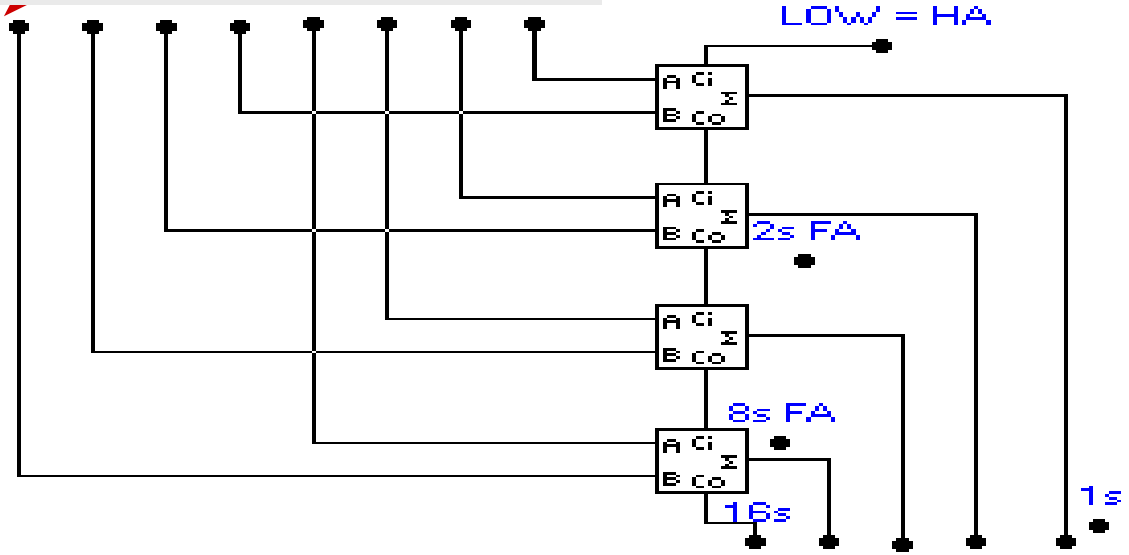
- Use half adder for LSD
- Use full adder for other digits



PARALLEL ADDER

Enter binary numbers to be added

1 1 1 0 + 0 1 1 0



1 0 1 0 0

Parallel adders are available in IC form.

1s place uses half-adder

2s, 4s, 8s places use full adders

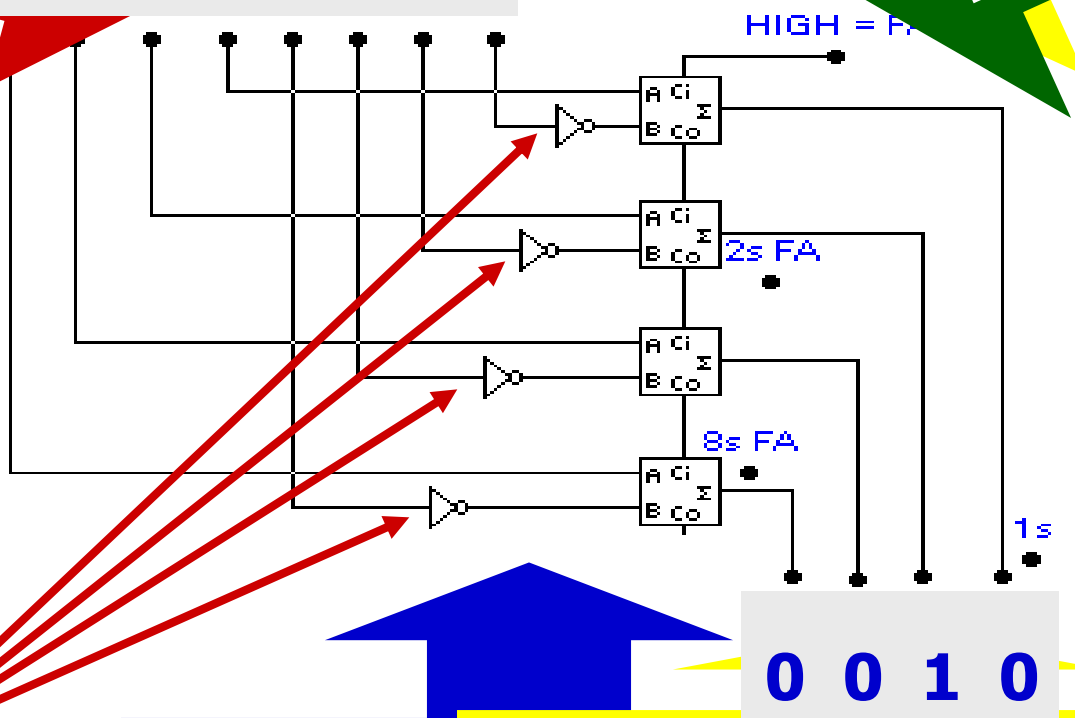
PARALLEL SUBTRACTOR USING FULL ADDERS

Binary numbers to be subtracted are input

1 0 0 1 - 0 1 1 1

HIGH at Carry in input acts like adding +1 to a 1s C number to form the 2s complement.

1sC is formed by four inverters.



Inverters

0 0 1 0

The result (difference) of the subtraction problem will appear here.

Note the us

Also notice the addition of four inverters on the B inputs to the FAs

BINARY MULTIPLICATION

Example: **Multiply the binary numbers 111 and 101.**

1 1 1		**Multiplicand**
x 1 0 1		**Multiplier**
1 1 1		**1st partial product**
0 0 0		**2nd partial product**
1 1 1		**3rd partial product**
1 0 0 0 1 1		**Product**

111 x 101 can also be calculated: 111 + 111 + 111 + 111 + 111

TEST

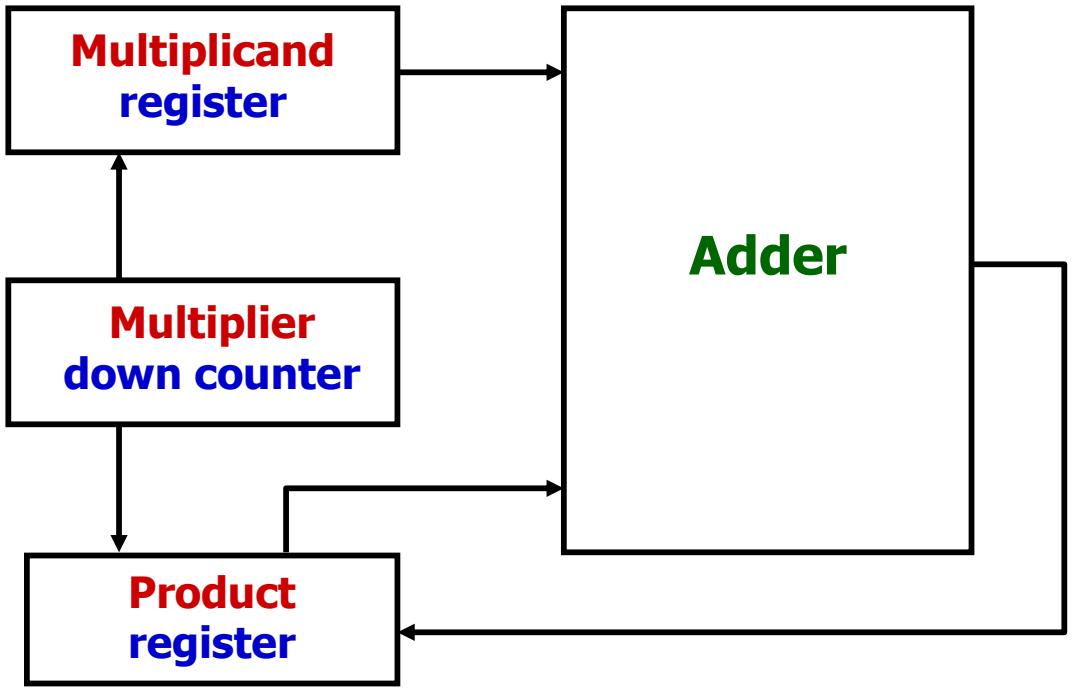
Multiply the binary numbers 101 and 100.

```
    1 0 1
  x 1 0 0
  -------
    0 0 0
  0 0 0
1 0 1
-------
1 0 1 0 0
```

BINARY MULTIPLIERS

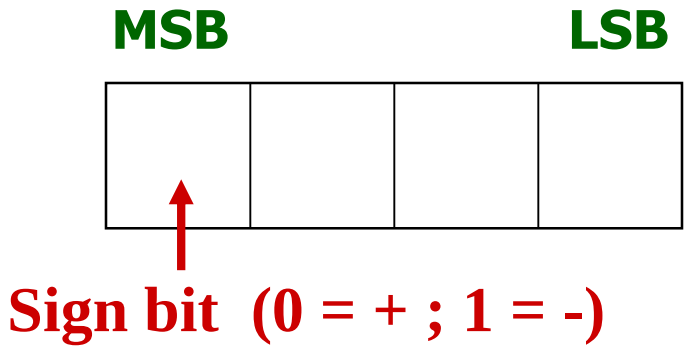
Binary multiplier circuits - utilize repeated addition.

Block Diagram:



2s COMPLEMENT NOTATION

- **2s complement representation - widely used in microprocessors.**
- **Represents *sign* and *magnitude***



Decimal:	+7	+4	+1	0	-1	-4	-7
2s Complement:	0111	0100	0001	0000	1111	1100	1001

2s COMPLEMENT - CONVERSIONS

- **Converting positive numbers to 2s complement:**
  - **Same as converting to binary**
- **Converting negative numbers to 2s complement:**

Decimal to 2s Complement

- 4 (decimal)

↓ **Convert decimal to binary**

0 1 0 0

↓ **1s complement**

1 0 1 1

↓ **Add 1**

- 4 = 1 1 0 0 (2s Complement)

2s Complement to Binary

1 1 0 0 (2s C)

↓ **1s complement**

0 0 1 1

↓ **Add 1**

0 1 0 0 (Binary)

ADDING/SUBTRACTING IN 2s COMPLEMENT

2s complement notation makes it possible to add and subtract signed numbers

(Decimal)	2s Complement
(- 1)	1 1 1 1
+ (- 2)	+ 1 1 1 0
(- 3)	1 1 1 0 1

Discard

2s complement

(+1)	0 0 0 1
+ (- 3)	+ 1 1 0 1
(- 2)	1 1 1 0

2s complement

TEST

Add the following 2s complement numbers:

(+5)	0 1 0 1
+ (- 4)	+ 1 1 0 0
(+1)	1 0 0 0 1

Discard

PRACTICAL SUGGESTION FOR BINARY MATH

- **Use a scientific calculator.**
- **Most scientific calculators have DEC, BIN, OCT, and HEX modes and can either convert between codes or perform arithmetic in different number systems.**
- **Most scientific calculators also have other functions that are valuable in digital electronics such as AND, OR, NOT, XOR, and XNOR logic functions.**