Chapter 30

Oscillators

Objectives

- After completing this chapter, you will be able to:
  - Describe an oscillator and its purpose
  - Identify the main requirements of an oscillator
  - Explain how a tank circuit operates and describe its relationship to an oscillator
  - Draw a block diagram of an oscillator

Objectives (cont'd.)

- Identify LC, crystal, and RC sinusoidal oscillator circuits
- Identify nonsinusoidal relaxation oscillator circuits
- Draw examples of sinusoidal and nonsinusoidal oscillators

Fundamentals of Oscillators

- Oscillator
  - A circuit that generates a repetitive AC signal
  - Output may be sinusoidal, rectangular, or sawtooth waveforms
- Main requirement
  - Output must not vary in frequency or amplitude

Fundamentals of Oscillators (cont'd.)

- Tank circuit
  - Formed when an inductor and capacitor are connected in parallel
  - Oscillates when excited by external DC source
  - Oscillation dampened by resistance of circuit
  - Oscillation maintained by positive feedback

Fundamentals of Oscillators (cont'd.)

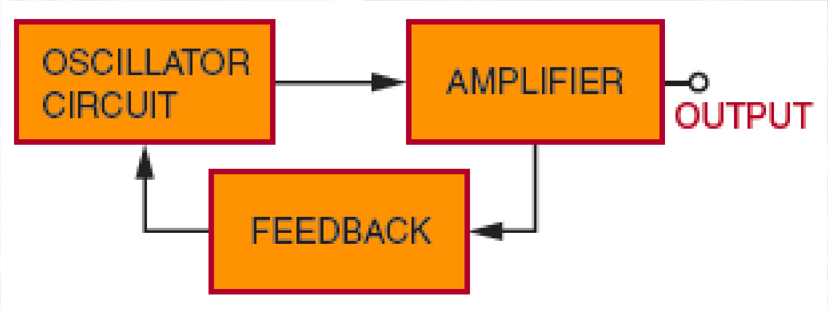


Figure 30-1. Block diagram of an oscillator.

Sinusoidal Oscillators

- Sinusoidal oscillators
  - Produce a sine-wave output
- Three basic types
  - LC oscillators
  - Crystal oscillators
  - RC oscillators

Sinusoidal Oscillators (cont'd.)

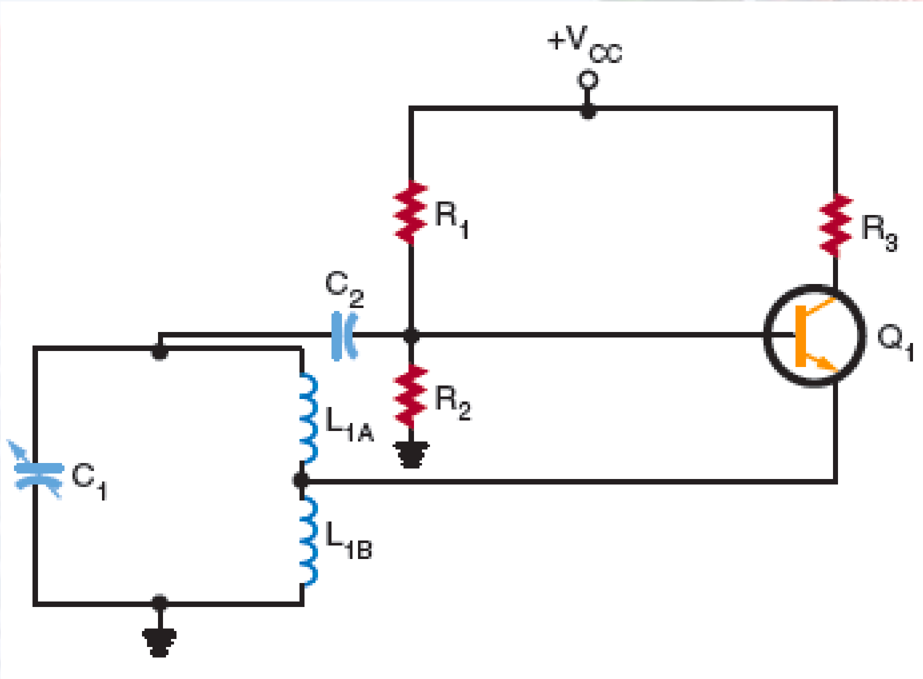


Figure 30-2. Series-fed Hartley oscillator.

Sinusoidal Oscillators (cont'd.)

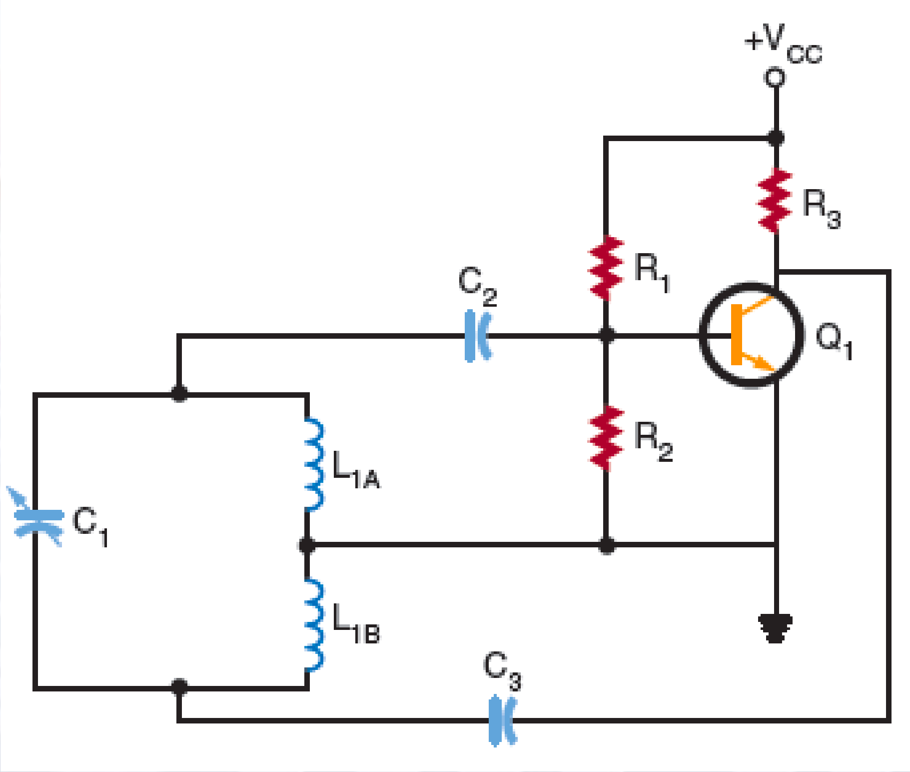


Figure 30-3. Shunt-fed Hartley oscillator.

Sinusoidal Oscillators (cont'd.)

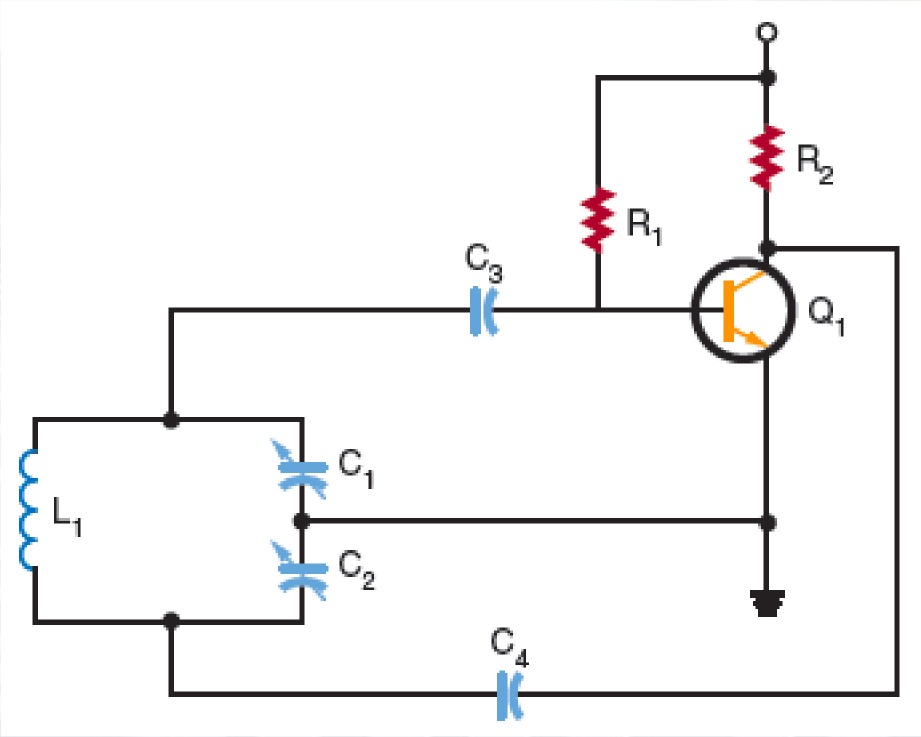


Figure 30-4. Colpitts oscillator.

Sinusoidal Oscillators (cont'd.)

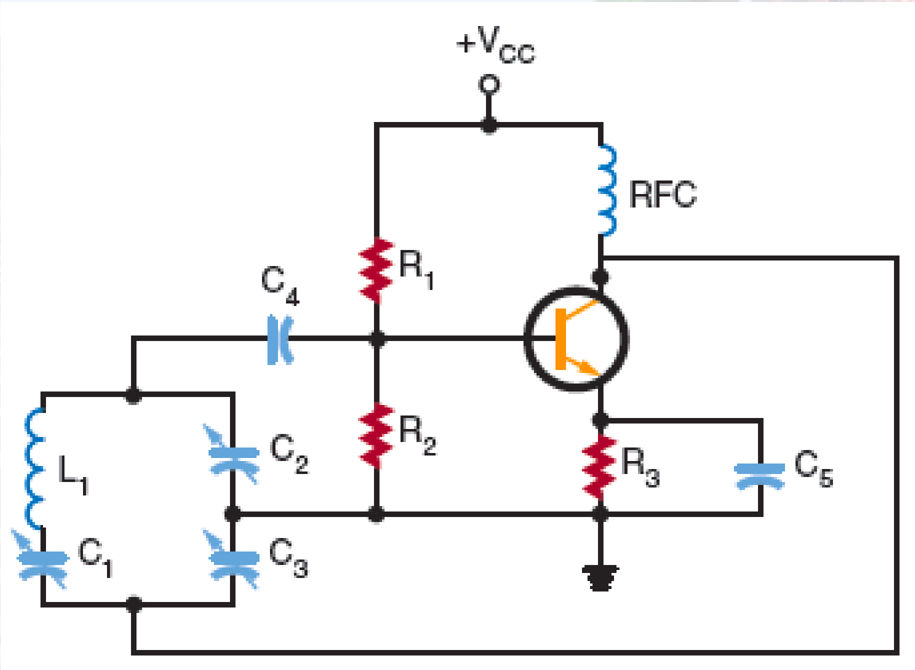


Figure 30-5. Clapp oscillator.

Sinusoidal Oscillators (cont'd.)

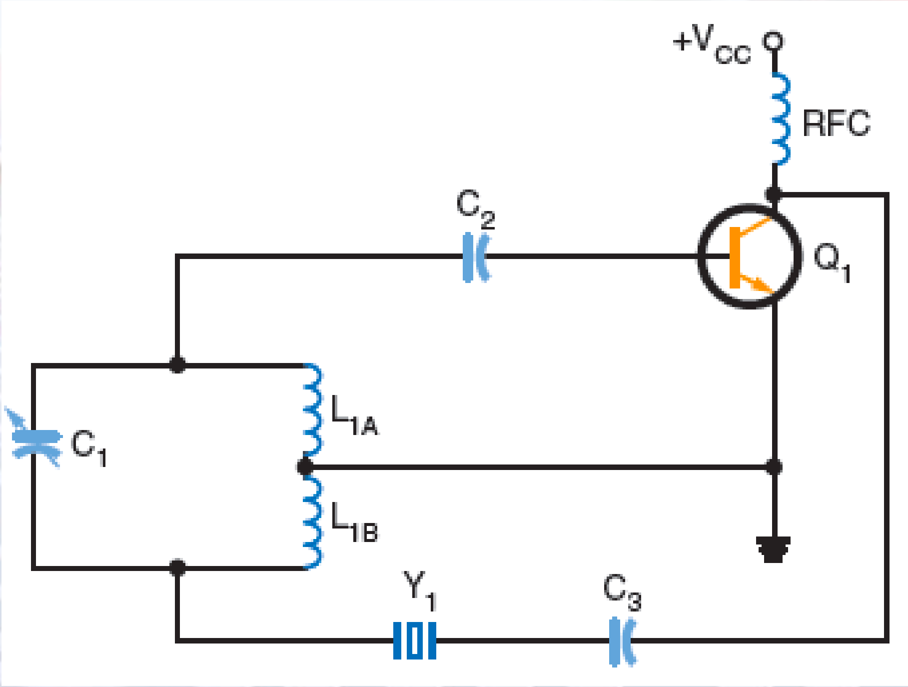


Figure 30-7. Crystal shunt-fed Hartley oscillator.

Sinusoidal Oscillators (cont'd.)

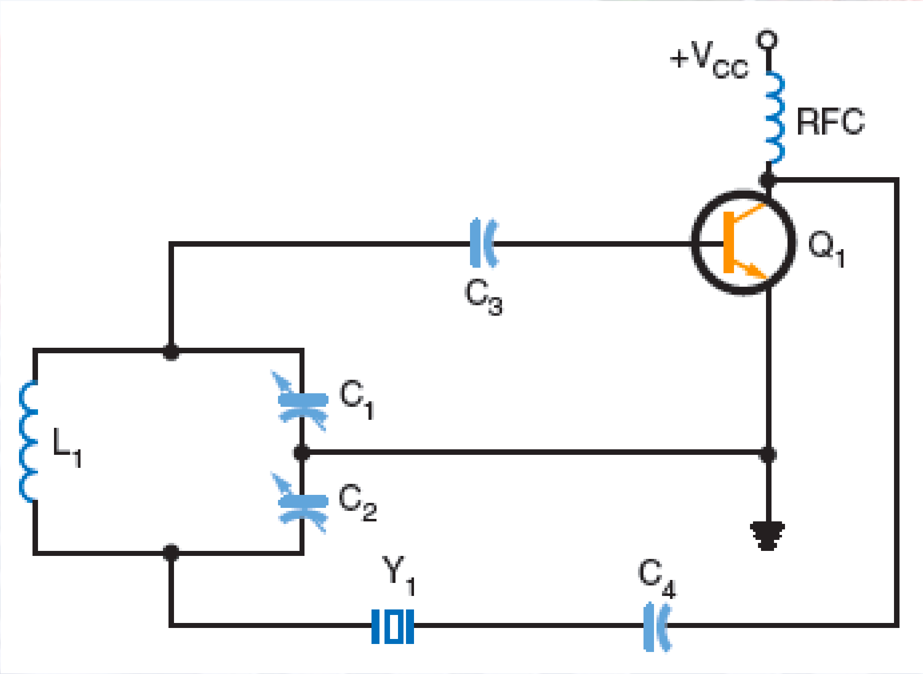


Figure 30-8. Colpitts crystal oscillator.

Sinusoidal Oscillators (cont'd.)

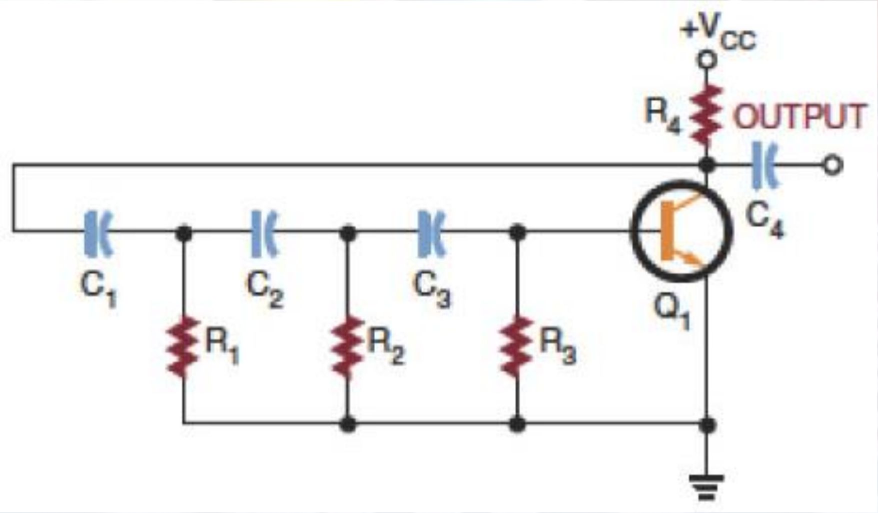


Figure 30-11. Phase-shift oscillator.

Sinusoidal Oscillators (cont'd.)

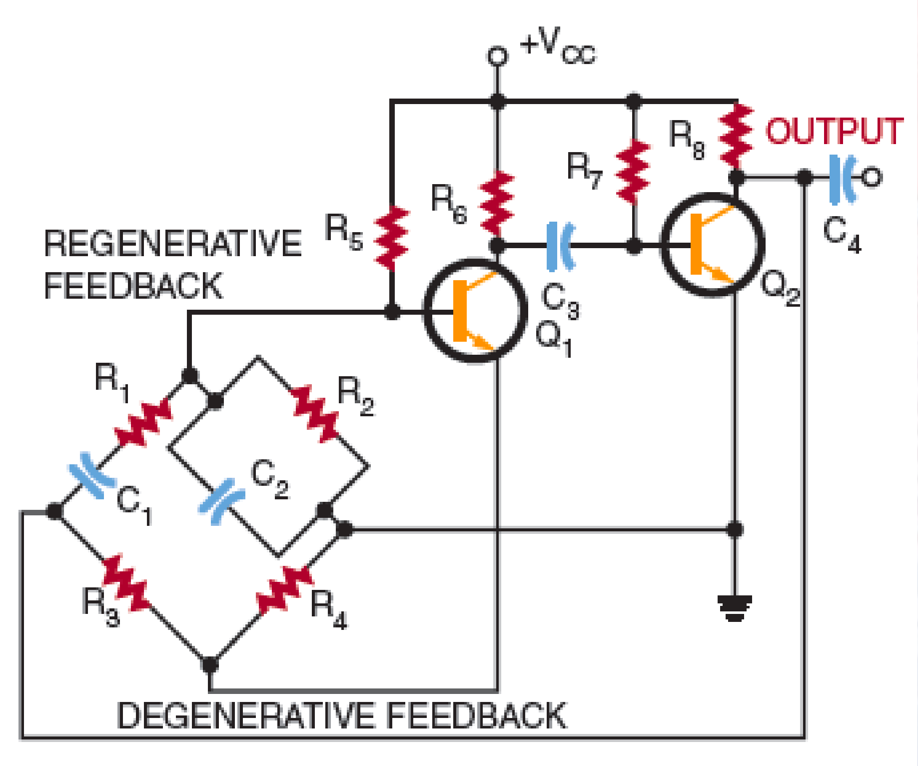


Figure 30-12. Wien-bridge oscillator.

Nonsinusoidal Oscillators

- Nonsinusoidal oscillators
  - Do not produce a sine-wave output
  - Outputs include square, sawtooth, rectangular, or triangular waveforms, or a combination of two waveforms
  - Form of relaxation oscillator

Nonsinusoidal Oscillators (cont'd.)

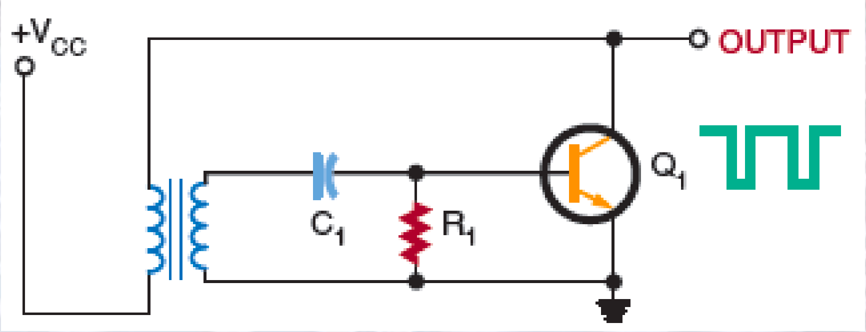


Figure 30-14. Blocking oscillator.

Nonsinusoidal Oscillators (cont'd.)

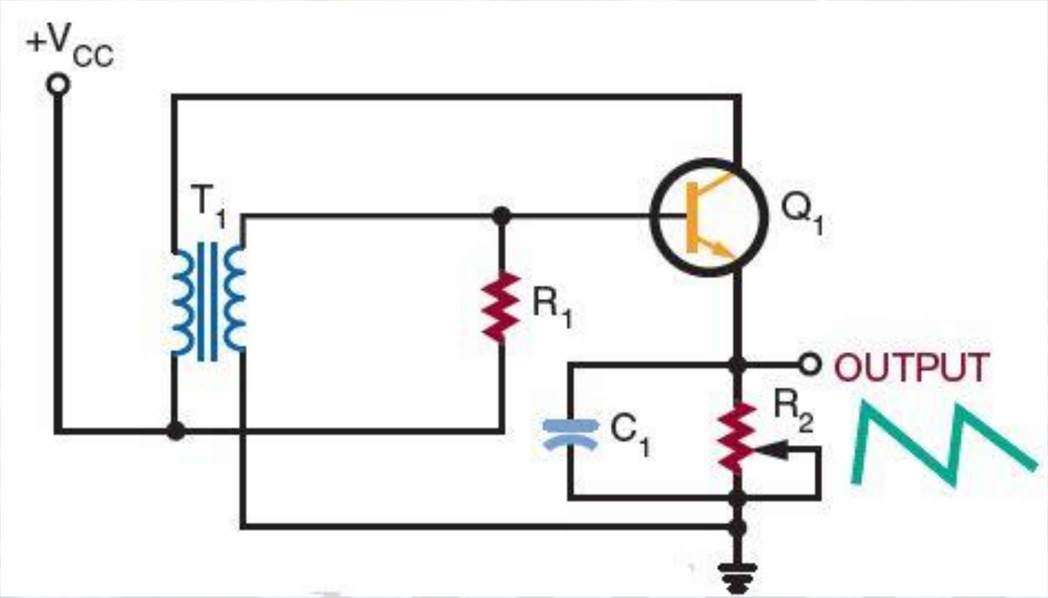


Figure 30-15. Sawtooth waveform generated by a blocking oscillator.

Nonsinusoidal Oscillators (cont'd.)

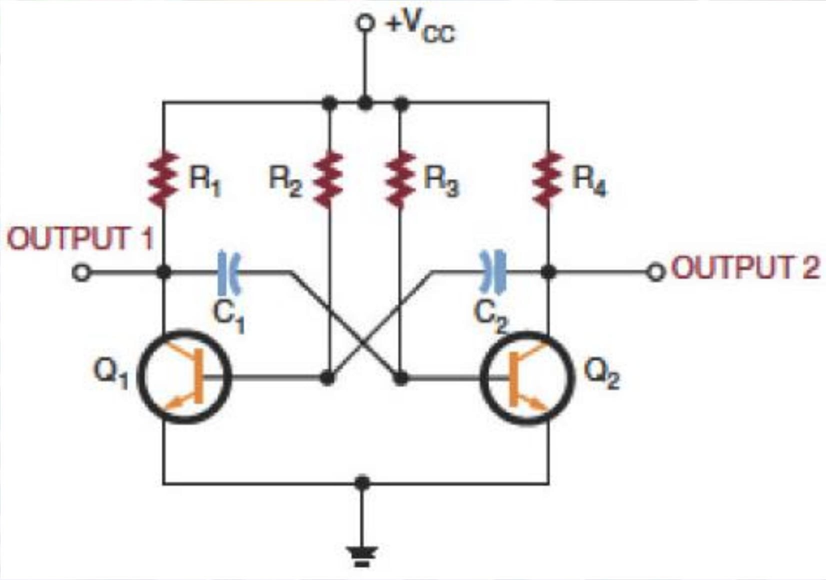


Figure 30-16. Free-running multivibrator.

Summary

- An oscillator is a nonrotating device for producing alternating current
- Main requirement of an oscillator
  - That output be uniform and not vary in frequency or amplitude
- A tank circuit oscillates when an external voltage source is applied

Summary (cont'd.)

- The three basic types of sinusoidal oscillators
  - LC oscillators, crystal oscillators, and RC oscillators
- Nonsinusoidal oscillators do not produce a sine-wave output
- A relaxation oscillator is the basis of all nonsinusoidal oscillators

Chapter 31

Waveshaping Circuits

Objectives

- After completing this chapter, you will be able to:
  - Identify ways in which waveform shapes can be changed
  - Explain the frequency-domain concept in waveform construction
  - Define *pulse width*, *duty cycle*, *rise* and *fall time*, *undershoot*, *overshoot*, and *ringing* as they relate to waveforms

Objectives (cont'd.)

– Explain how differentiators and integrators work

– Describe clipper and clamper circuits

– Describe the differences between monostable and bistable multivibrators

– Draw schematic diagrams of waveshaping circuits

Nonsinusoidal Waveforms

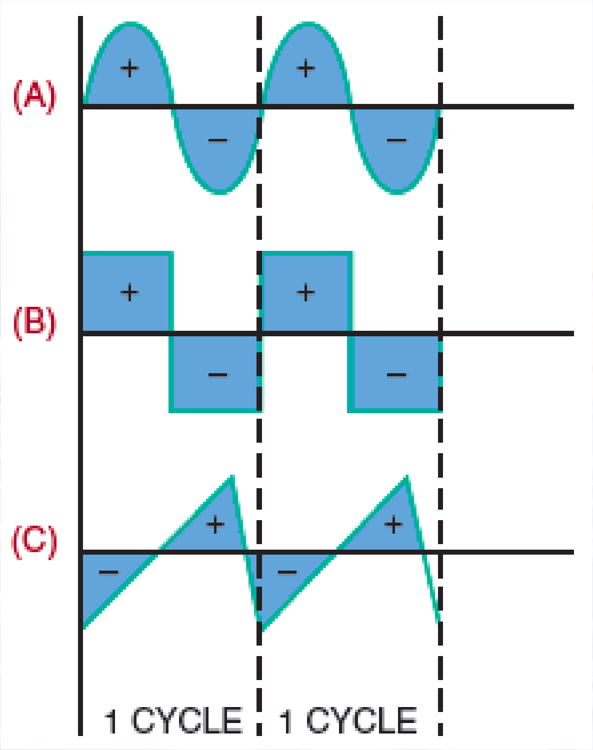


Figure 31-1. Three basic waveforms: (A) sine wave, (B) square wave, (C) sawtooth wave.

FIGURE 31–2

Chart of fundamental frequency of 1000 hertz and some of its harmonics.

(Fundamental)	1st Harmonic	1000 Hz
	2nd Harmonic	2000 Hz
	3rd Harmonic	3000 Hz
	4th Harmonic	4000 Hz
	5th Harmonic	5000 Hz

- Sine waves are important because they are the only waveform that cannot be distorted by RC, RL, or LC circuits

Nonsinusoidal Waveforms (cont'd.)

Type and number of harmonics included in the periodic waveform depend on the shape of the waveform.

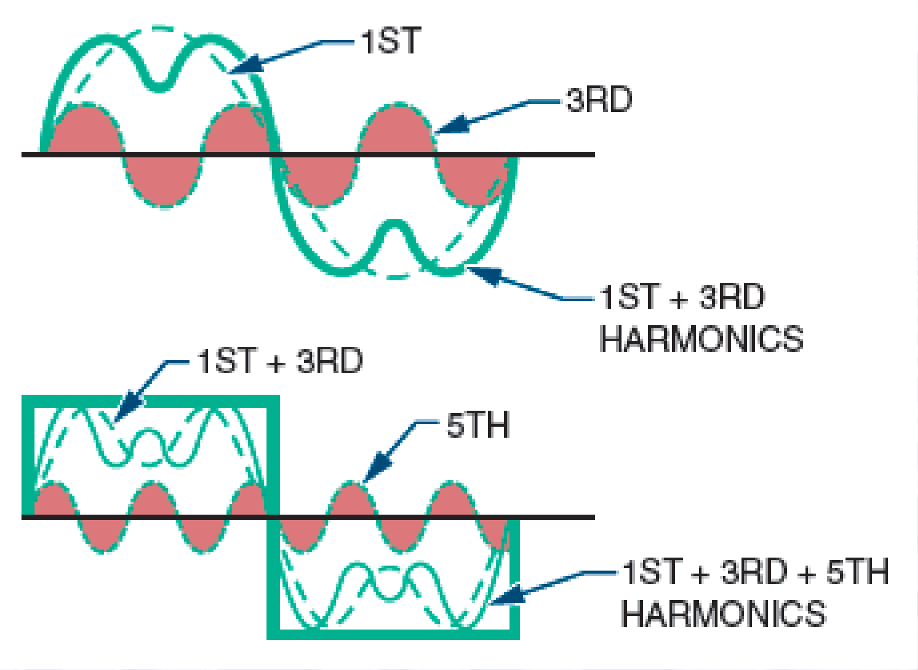
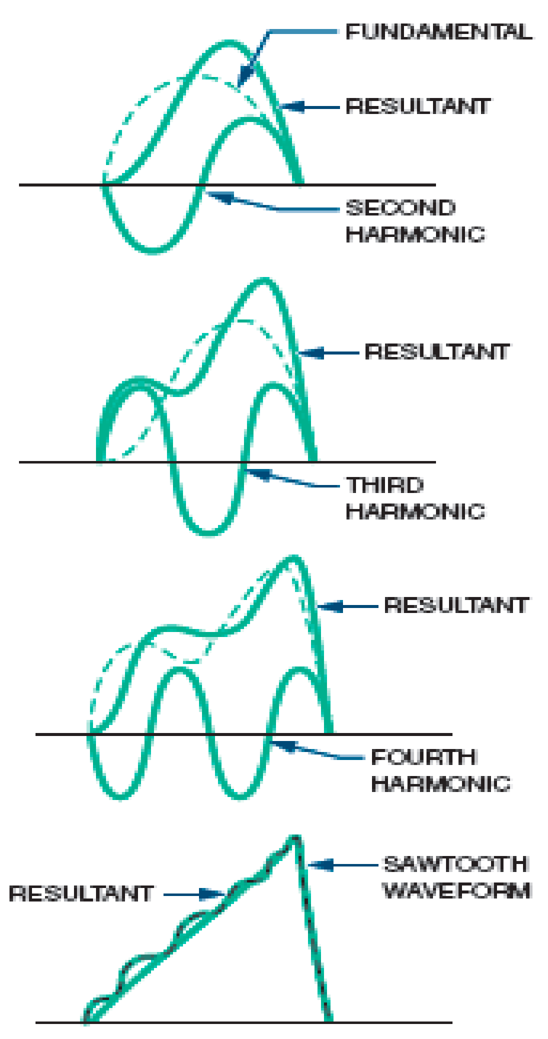


Figure 31-4. Formation of a square wave by the frequency domain method.



- It consists of the fundamental frequency plus odd harmonics in-phase and even harmonics crossing the zero reference line 180 degrees out of phase with the fundamental

Figure 31-5. Formation of a sawtooth wave by the frequency domain method.

Nonsinusoidal Waveforms (cont'd.)

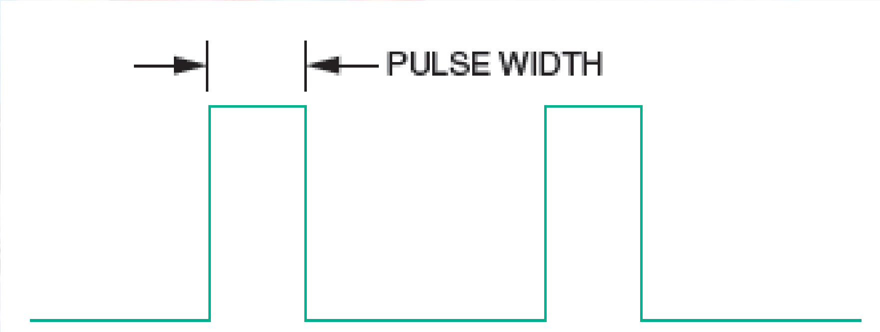
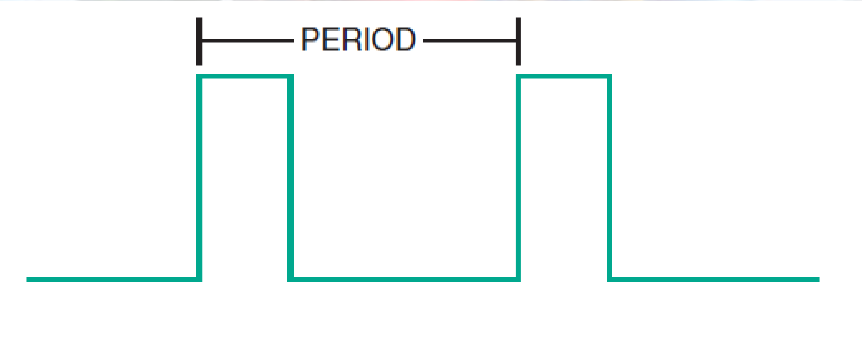


Figure 31-8. Pulse width of a waveform.



Period of a waveform

Nonsinusoidal Waveforms (cont'd.)

- Duty cycle
  - Ratio of the pulse width to the period
  - Can be represented as a percentage:

Duty cycle = pulse width/period

Nonsinusoidal Waveforms (cont'd.)

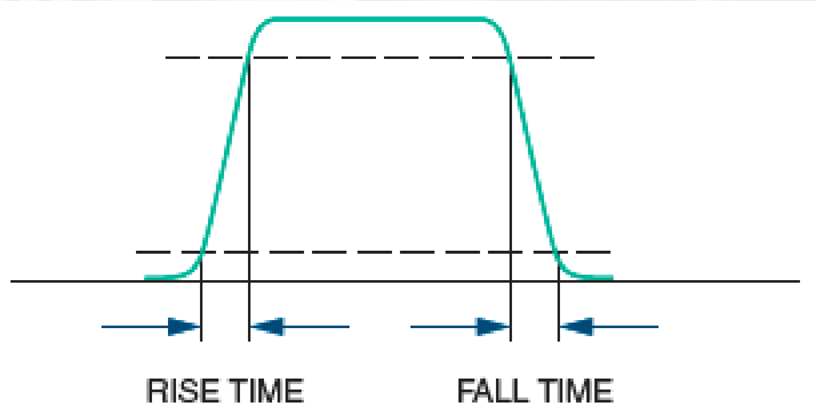


Figure 31-9. The rise and fall times of a waveform are measured at 10% and 90% of the waveform's maximum amplitude.

Nonsinusoidal Waveforms (cont'd.)

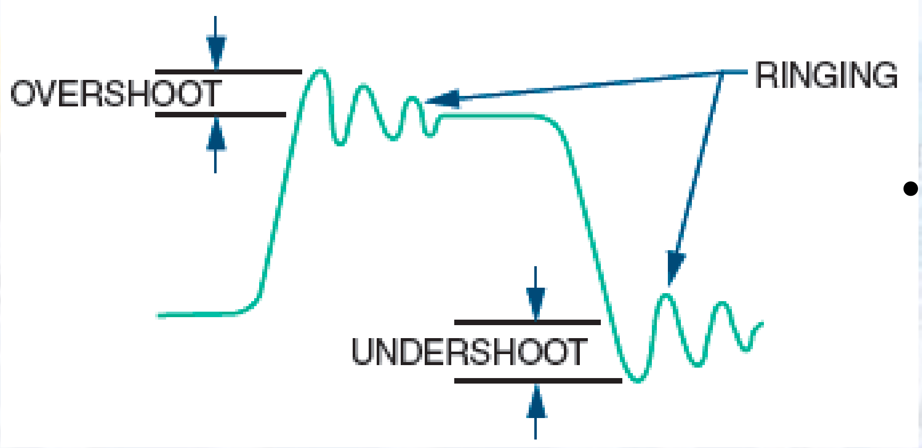


Figure 31-10. Overshoot, undershoot, and ringing.

- *Overshoot* occurs when the leading edge of a waveform exceeds its normal maximum value.
- *Undershoot* occurs when the trailing edge exceeds its normal minimum value.
- Both conditions are followed by damped oscillations known as ringing.

Waveshaping Circuits

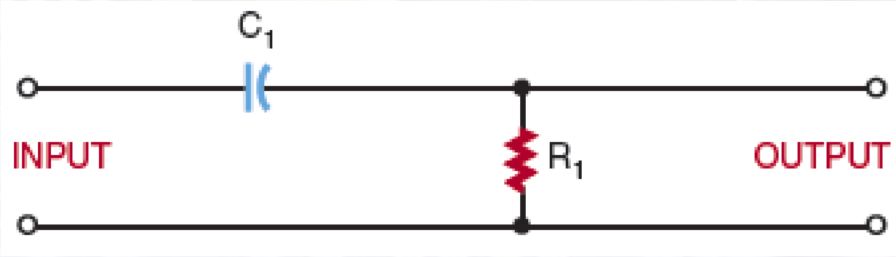


Figure 31-11. Differentiator circuit.

Waveshaping Circuits (cont'd.)

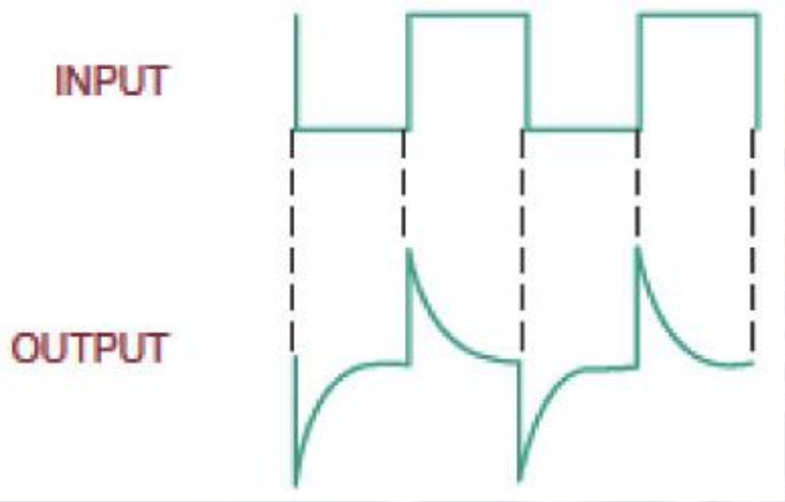
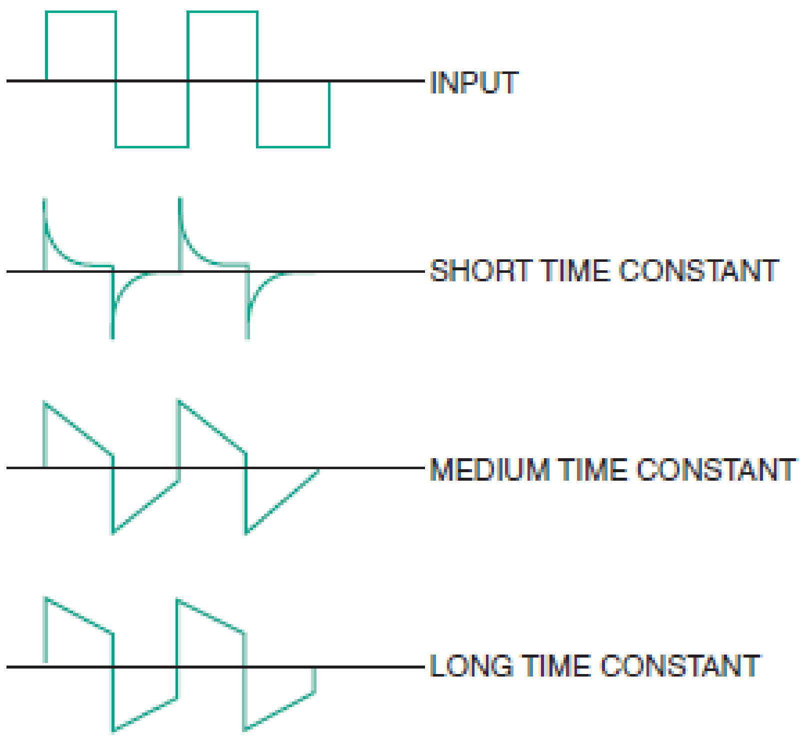


Figure 31-12. Result of applying a square wave to a differentiator circuit.

FIGURE 31–13

Effects of different time constants on a differentiator circuit.



Waveshaping Circuits (cont'd.)

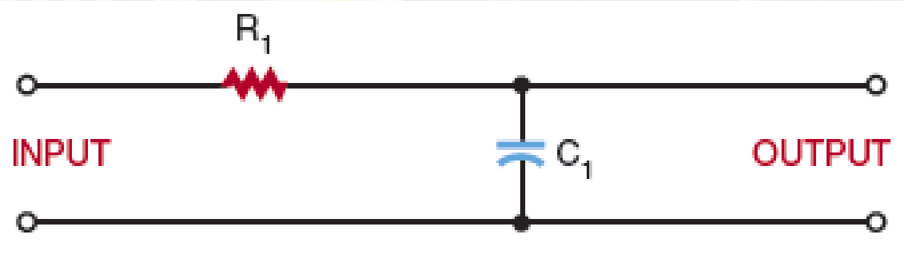


Figure 31-14. Integrator circuit.

Waveshaping Circuits (cont'd.)

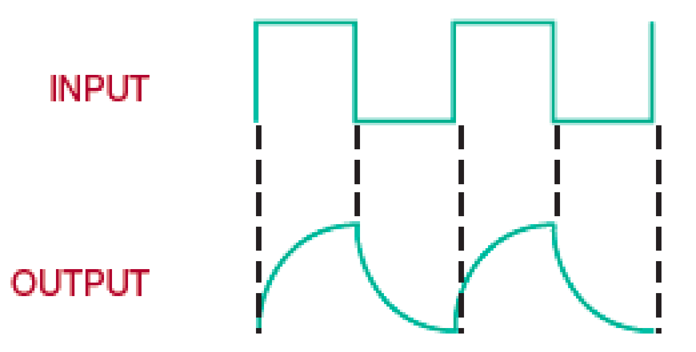
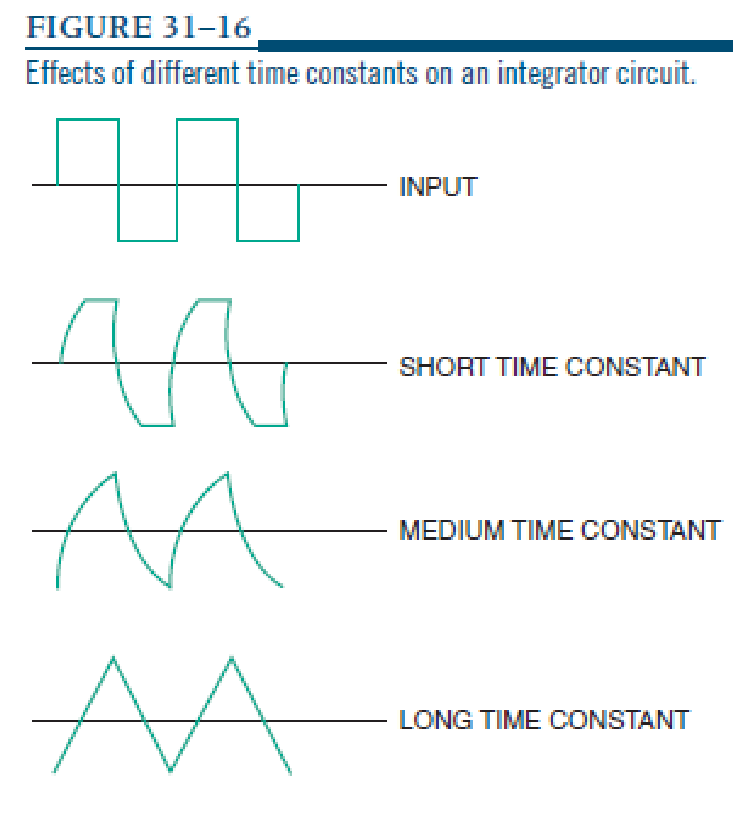


Figure 31-15. Result of applying a square wave to an integrator circuit.



Waveshaping Circuits (cont'd.)

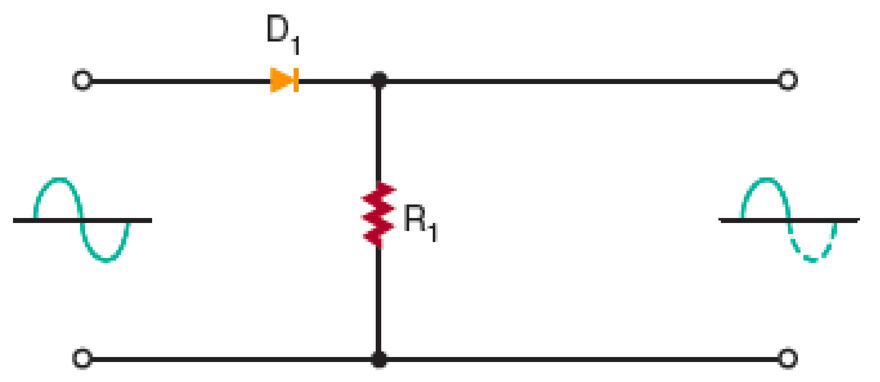


Figure 31-17. Basic series diode clipping circuit.

In a biased series clipping circuit, diode cannot conduct until the input signal exceeds the bias source

FIGURE 31–18
Result of reversing the diode in a clipping circuit.

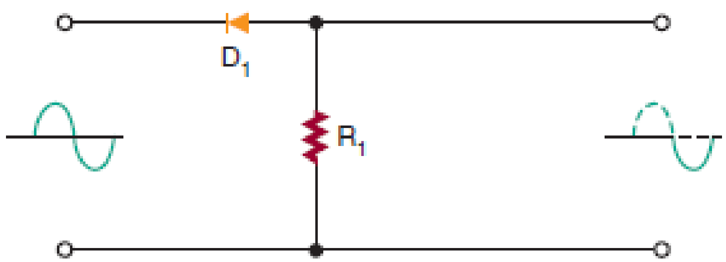


FIGURE 31–19
Biased series diode clipping circuit.

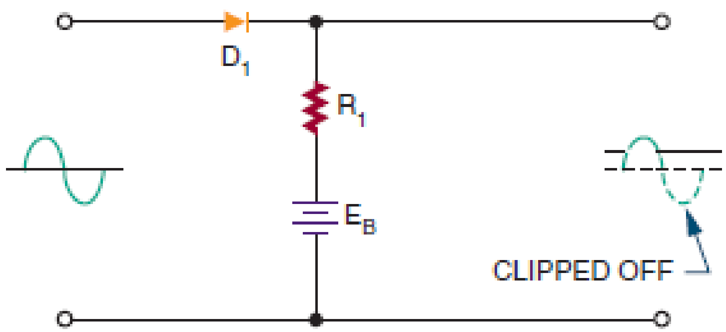


FIGURE 31–23

Biased shunt diode clipping circuit.

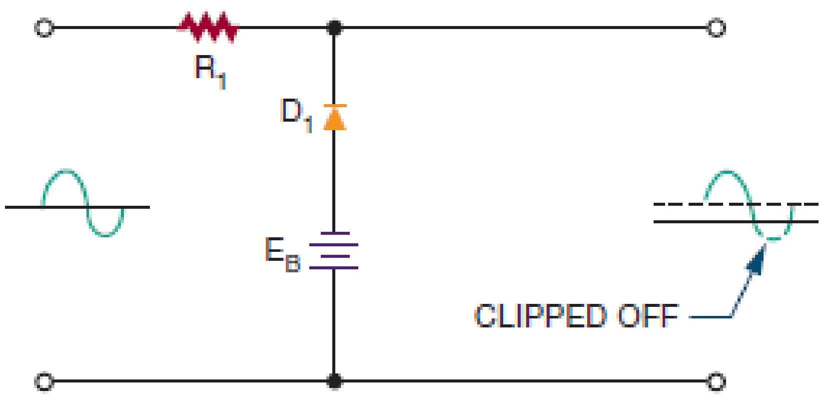
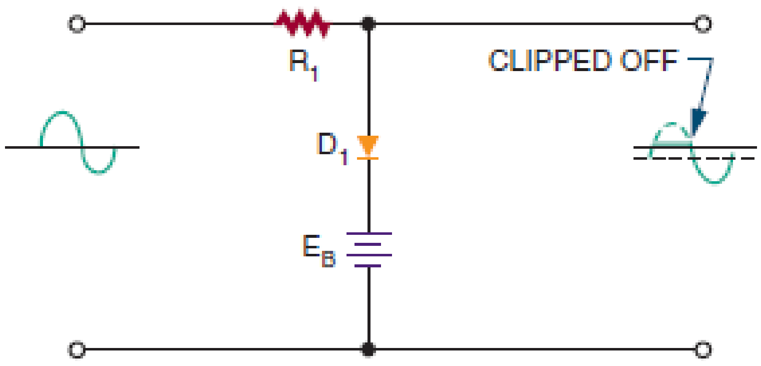


FIGURE 31–24

Effect of reversing the diode and bias source in a shunt diode clipping circuit.



Waveshaping Circuits (cont'd.)

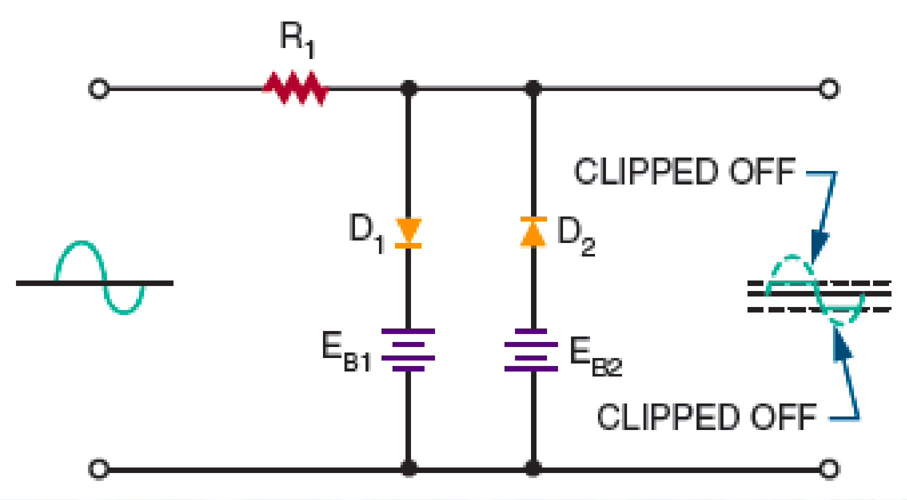
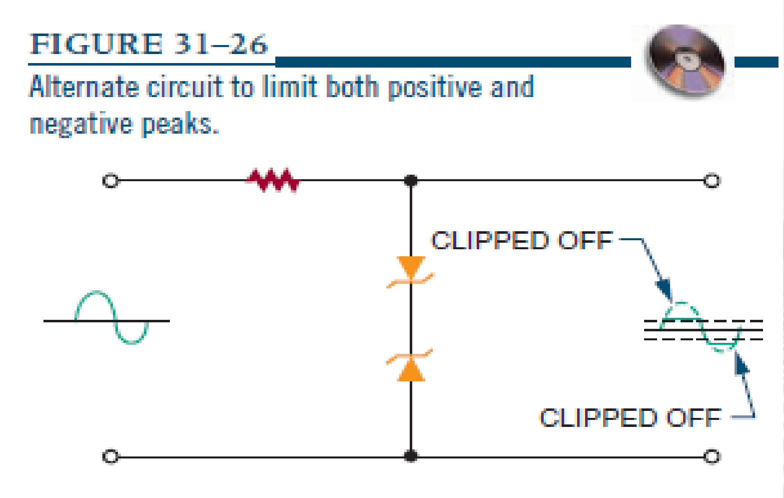


Figure 31-25. Clipping circuit used to limit both the positive and negative peaks.



This circuit prevents the output signal from exceeding predetermined values for both peaks

Waveshaping Circuits (cont'd.)

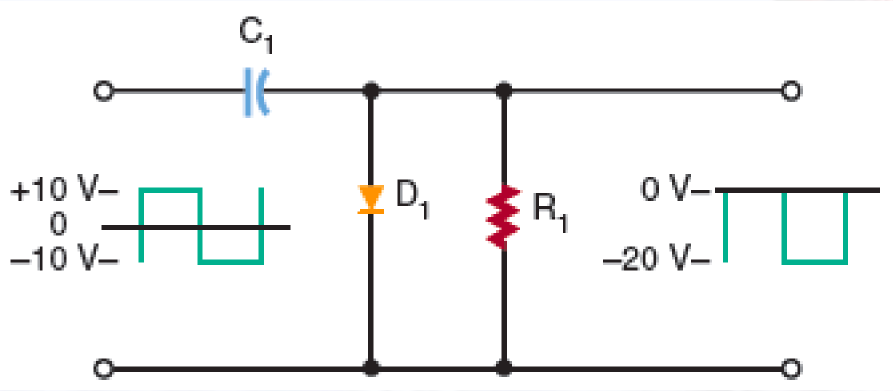


Figure 31-27. Diode clamper circuit.

- A diode clamper is called a DC restorer.
- This circuit is commonly used in radar, television, telecommunications, and computers.
- In the circuit, a square wave is applied to an input signal.
- The purpose of the circuit is to clamp the top of the square wave to 0 volts, without changing the shape of the waveform.

Special-Purpose Circuits

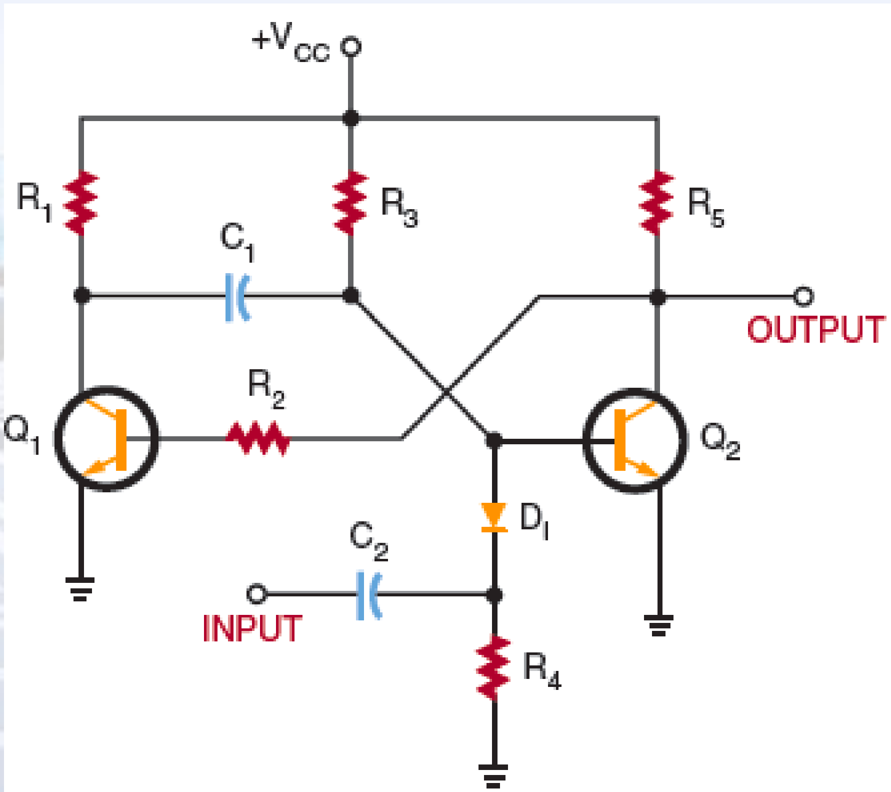


Figure 31-28. Monostable multivibrator.

- It produces one output pulse for each input pulse.
- Output pulse is generally longer than the input pulse.
- This circuit is also called pulse stretcher.
- Circuit is used as a gate in computers, electronic control circuits, and communication equipment.

Special-Purpose Circuits (cont'd.)

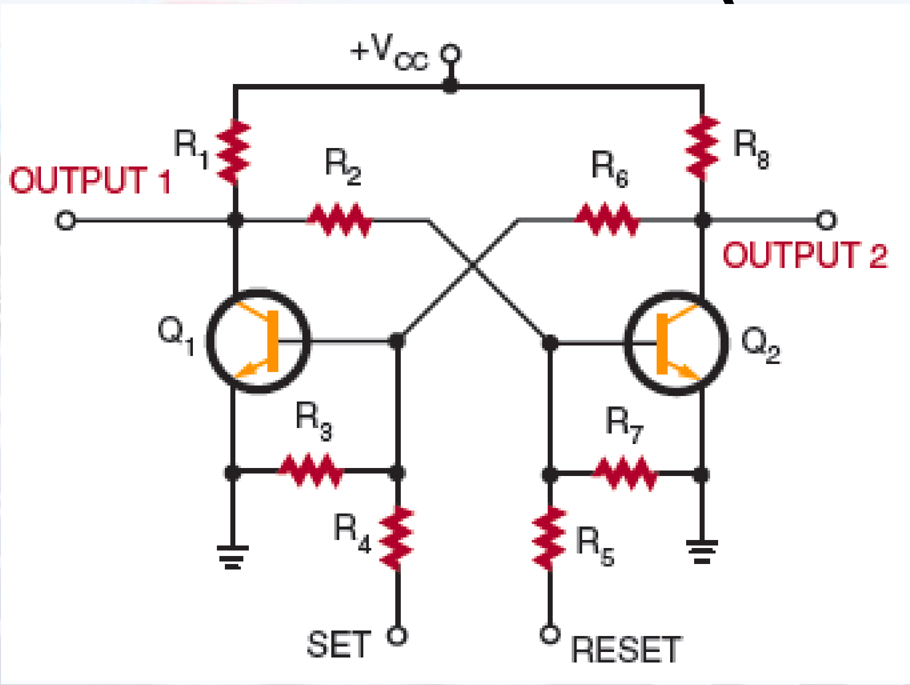
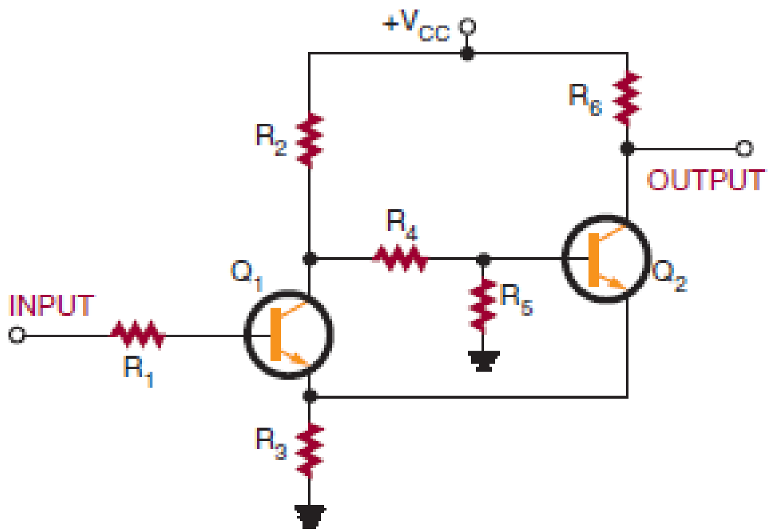


Figure 31-29. Basic flip-flop circuit.

- Flip-flop circuit produces a square or rectangular waveform for use in gating or timing signals or for on-off switching operations in binary counter circuits.
- Discrete versions of the flip-flop find little application today and integrated circuit versions of the flip-flop find wide application.

FIGURE 31–30

Basic Schmitt trigger circuit.



- **Schmitt trigger is** to convert a sine-wave, sawtooth, or other irregularly shaped waveform to a square or rectangular wave.
- The circuit differs from a conventional bistable multivibrator in that one of the coupling networks is replaced by a common-emitter resistor (R_3).

Summary

- Waveforms can be changed from one shape to another using electronic circuits
- The frequency domain concept
  - All periodic waveforms are made of sine waves
- Key concepts introduced in the chapter:
  - Pulse width, duty cycle, rise and fall time, undershoot, overshoot, and ringing

Summary (cont'd.)

- An RC circuit can be used to change the shape of a complex waveform
- A monostable multivibrator (one-shot multivibrator) produces one output pulse for each input pulse
- Bistable multivibrators have two stable states and are called flip-flops

Chapter 32

Binary Number System

Objectives

- After completing this chapter, you will be able to:
  – Describe the binary number system
  – Identify the place value for each bit in a binary number
  – Convert binary numbers to decimal, octal, and hexadecimal numbers

Objectives (cont'd.)

- Convert decimal, octal, and hexadecimal numbers to binary numbers
- Convert decimal numbers to 8421 BCD code
- Convert 8421 BCD code numbers to decimal numbers

- Binary system is a base-two system because it contains two digits, 0 and 1.
- Position of the 0 or 1 in a binary number indicates its value within the number. This is referred to as its place value or weight.

Binary Numbers

- Base-two system
  - Contains two digits, 0 and 1
- Place value

	Place Value					
	32	16	8	4	2	1
Power of 2:	2^5	2^4	2^3	2^2	2^1	2^0

  - Position of the 0 or 1 indicates its value within the number
- Highest number that can be represented
  - $2^n - 1$

Binary and Decimal Conversion

- To convert decimal numbers to binary
  - Divide the decimal number by 2
  - Write down the remainder after each division
  - The remainders, taken in reverse order, form the binary number

Decimal Number	Binary Number 2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0
31	1	1	1	1	1

Figure 32-1. Decimal number and equivalent binary table.

EXAMPLE:

	Place Value					
	32	16	8	4	2	1
Binary number:	1	0	1	1	0	1

$$
\begin{array}{rl}
\text{Value:} \quad 1 \times 32 & = 32 \\
0 \times 16 & = 0 \\
1 \times 8 & = 8 \\
1 \times 4 & = 4 \\
0 \times 2 & = 0 \\
\underline{+\,1 \times 1} & = 1 \\
101101_2 & = 45_{10}
\end{array}
$$

The number 45 is the decimal equivalent of the binary number 101101.

EXAMPLE: Determine the decimal value of the binary number 111011.011.

Binary number	Place value	Value
1	$\times$ 32	= 32
1	$\times$ 16	= 16
1	$\times$ 8	= 8
0	$\times$ 4	= 0
1	$\times$ 2	= 2
1	$\times$ 1	= 1
0	$\times$ 0.5	= 0
1	$\times$ 0.25	= 0.25
+ 1	$\times$ 0.125	= 0.125
111011.011_2		$= 59.375_{10}$

The process can be simplified by writing the numbers in an orderly fashion as shown for converting 25_{10} to a binary number.

EXAMPLE:

2	25		LSB
2	12	1	
2	6	0	
2	3	0	
2	1	1	
	0	1	

Decimal number 25 is equal to binary number 11001.

Fractional numbers are done a little differently: The number is multiplied by 2 and the carry is recorded as the binary fraction.

LSB: least significant bit

EXAMPLE: To convert decimal 0.85 to a binary fraction, progressively multiply by 2.

LSB

$0.85 \times 2 = 1.70 = 0.70$ with a carry of 1

$0.70 \times 2 = 1.40 = 0.40$ with a carry of 1

$0.40 \times 2 = 0.80 = 0.80$ with a carry of 0

$0.80 \times 2 = 1.60 = 0.60$ with a carry of 1

$0.60 \times 2 = 1.20 = 0.20$ with a carry of 1

$0.20 \times 2 = 0.40 = 0.40$ with a carry of 0

Continue to multiply by 2 until the needed accuracy is reached. Decimal 0.85 is equal to 0.110110 in binary form.

Octal Numbers

- Octal numbers
  - Allow reading of large binary numbers
  - Breaks binary number into groups of three
  - Base 8

Decimal Number	Binary Number	Octal Number
0	00 000	0
1	00 001	1
2	00 010	2
3	00 011	3
4	00 100	4
5	00 101	5
6	00 110	6
7	00 111	7
8	01 000	10
9	01 001	11
10	01 010	12
11	01 011	13
12	01 100	14
13	01 101	15
14	01 110	16
15	01 111	17
16	10 000	20
17	10 001	21
18	10 010	22
19	10 011	23
20	10 100	24
21	10 101	25
22	10 110	26
23	10 111	27
24	11 000	30
25	11 001	31
26	11 010	32
27	11 011	33
28	11 100	34
29	11 101	35
30	11 110	36
31	11 111	37

Figure 32-2. Decimal and binary equivalent of octal numbers.

EXAMPLE:

Binary number 100101001110000111010_2

Separate into groups of three:

$100\ 101\ 001\ 110\ 000\ 111\ 010_2$

Convert to octal: $100\ 101\ 001\ 110\ 000\ 111\ 010_2$

$4\quad 5\quad 1\quad 6\quad 0\quad 7\quad 2_8$

Octal equivalent is: 4516072_8

EXAMPLE:

Octal number: 1672054_8

Separate the numbers: $1\quad 6\quad 7\quad 2\quad 0\quad 5\quad 4_8$

Convert to binary: $001\ 110\ 111\ 010\ 000\ 101\ 100_2$

Binary equivalent: 001110111010000101100_2

Lead zeros can be dropped resulting in:

1110111010000101100_2

Octal Numbers (cont'd.)

- To convert binary to an octal number
  - Divide the binary number into groups of three starting from the right
- To convert an octal number to binary
  - Reverse process
  - Convert the octal number to binary groups of three

Octal Numbers (cont'd.)

Powers of 8	Place Value
8^0	1
8^1	8
8^2	64
8^3	512
8^4	4096
8^5	32768
8^6	262144
8^7	2097152

Figure 32-3. Place values of octal numbers.

Hexadecimal Numbers

- Hexadecimal number system
  - Used with microprocessor-based systems
  - Breaks binary number into groups of four
  - Reduces error when entering data
  - Base 16

Hexadecimal Numbers (cont'd.)

- To convert binary to hexadecimal number
  - Divide the binary number into groups of four starting from the right
- To convert hexadecimal number to binary
  - Reverse the process
  - Convert the hexadecimal number to binary groups of four

Hexadecimal Numbers (cont'd.)

Powers of 16	Place Value
16^0	1
16^1	16
16^2	256
16^3	4096
16^4	65536
16^5	1048576
16^6	16777216
16^7	268435456

Figure 32-5. Place values of hexadecimal numbers.

BCD Code

- Binary-coded-decimal (BCD)
  - 8421 code
  - Consists of four binary digits
  - Represents the digits 0 through 9
  - Permits easy conversion between decimal and binary form

Powers of 2:	2^3	2^2	2^1	2^0
Binary weight:	8	4	2	1

Decimal	8421 code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

EXAMPLE: Find the decimal equivalent for each of the following BCD codes: 10010101, 1001000, 1100111, 1001100101001, 1001100001110110.

1001 0101 = 95

0100 1000 = 48

0110 0111 = 67

0001 0011 0010 1001 = 1329

1001 1000 0111 0110 = 9876

EXAMPLE: Convert the following decimal numbers into a BCD code: 5, 13, 124, 576, 8769.

5 = 0101

13 = 0001 0011

124 = 0001 0010 0100

576 = 0101 0111 0110

8769 = 1000 0111 0110 1001

BCD Code (cont'd.)

- To express a decimal number in the 8421 code:
  - Replace each decimal digit by the 4-bit code
- To determine a decimal number from an 8421 code:
  - Break the code into groups of 4 bits
  - Write the decimal digit represented by each 4-bit group

Summary

- The binary number system contains two digits, 0 and 1
  - Place value increases by a power of 2
- To convert from decimal to binary:
  - Divide the decimal number by 2
  - Write down the remainder after each division
  - The remainders, taken in reverse order, form the binary number

Summary (cont'd.)

- Similar steps are used to convert octal and hexadecimal numbers to and from decimal numbers as with the binary number system
- The 8421 code (BCD) is used to represent digits 0 through 9

Chapter 33

Basic Logic Gates

Objectives

- After completing this chapter, you will be able to:
  - Identify and explain the function of the basic logic gates
  - Draw the symbols for the basic logic gates
  - Develop truth tables for the basic logic gates

AND Gate

- Two or more inputs and a single output
- Produces a 1 output only when all inputs are 1s
- Total number of possible combinations:

  $N = 2^n$

  where: n = total number of input variables
- Performs basic operation of multiplication

AND Gate (cont'd.)

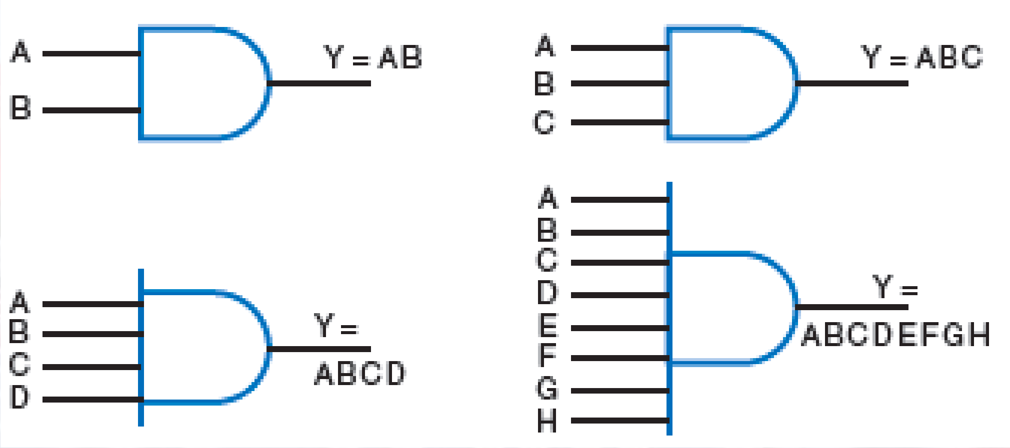


Figure 33-1. Logic symbol for an AND gate.

AND Gate (cont'd.)

INPUTS		OUTPUT
A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

Figure 33-2. Truth table for a two-input AND gate.

OR Gate

- Produces a 1 output if any of its inputs are 1s
- Performs basic operation of addition
- Can have any number of inputs greater than one

OR Gate (cont'd.)

INPUTS		OUTPUT
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

Figure 33-3. Truth table for a two-input OR gate.

OR Gate (cont'd.)

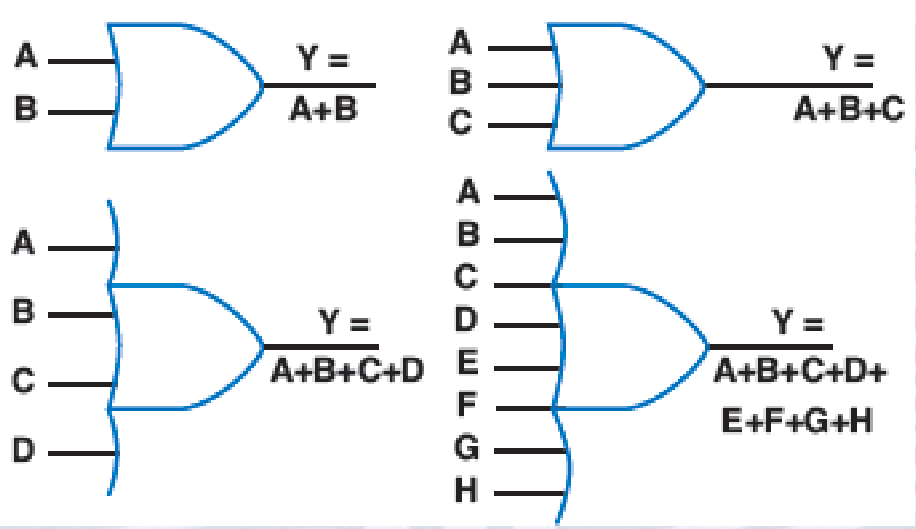


Figure 33-4. Logic symbol for an OR gate.

NOT Gate

- Referred to as an inverter
- Converts the input state to an opposite output state
- Only two input combinations possible

NOT Gate (cont'd.)

INPUTS	OUTPUT
A	Y
0	1
1	0

Figure 33-5. Truth table for an inverter.

NOT Gate (cont'd.)

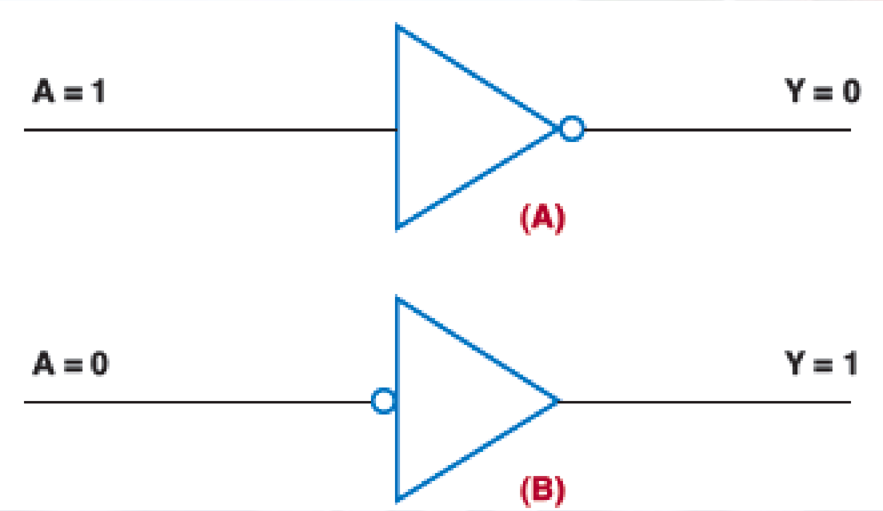


Figure 33-6. Logic symbol for an inverter.

NAND Gate

- Combination of an inverter and an AND gate (NOT-AND)
- Most commonly used logic function
- Produces a 1 output when any of the inputs are 0s
- Available with two, three, four, eight, and thirteen inputs

NAND Gate (cont'd.)

INPUTS		OUTPUT
A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

FIGURE 33–7
Logic symbol for an NAND gate.

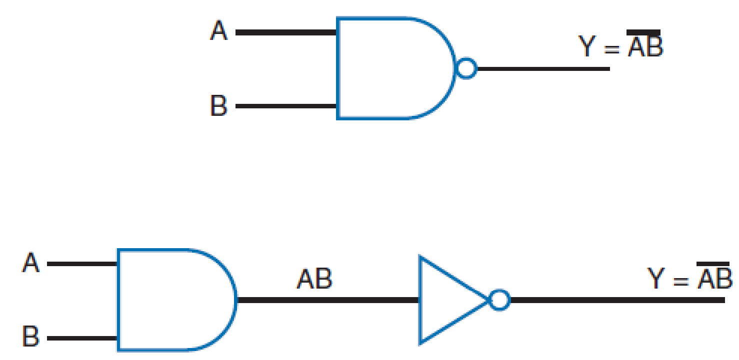


Figure 33-8. Truth table for a two-input NAND gate.

NOR Gate

- Combination of an inverter and an OR gate (NOT-OR)
- Produces a 1 output only when both inputs are 0s
- Available with two, three, four, and eight inputs

NOR Gate (cont'd.)

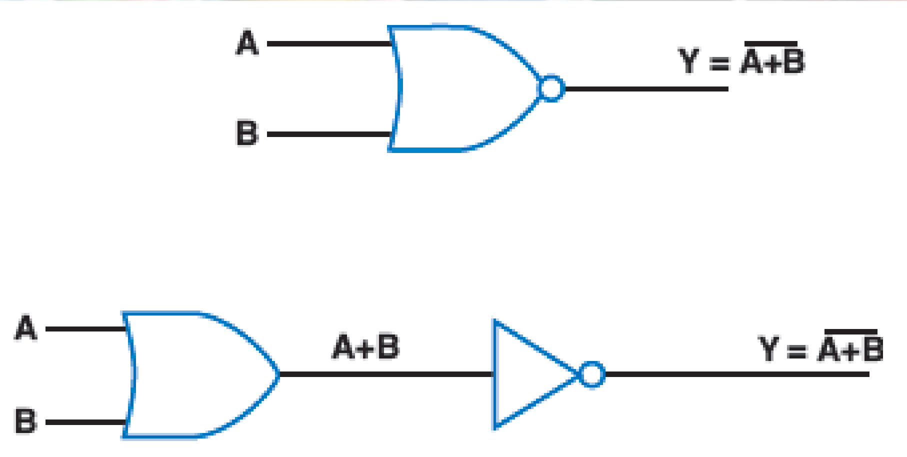


Figure 33-10. Logic symbol for a NOR gate.

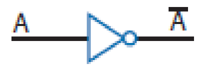
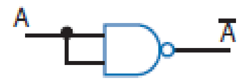
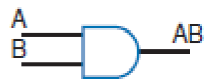
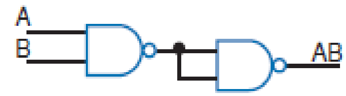
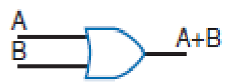
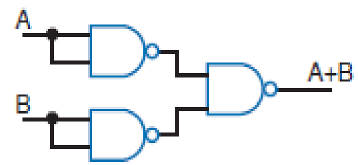
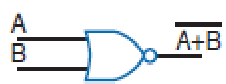
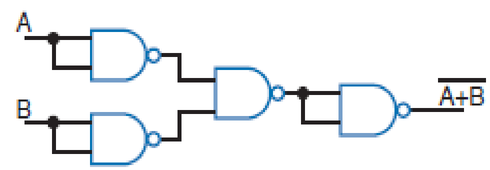
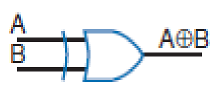
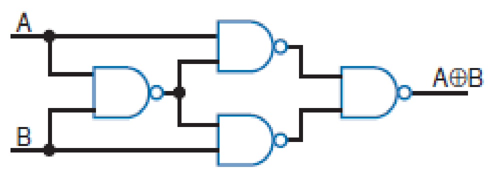
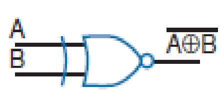
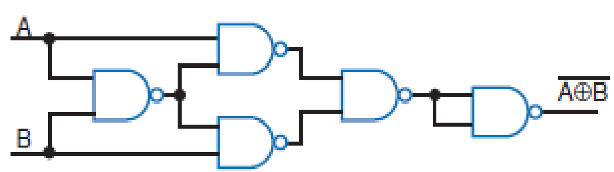
NOR Gate (cont'd.)

INPUTS		OUTPUT
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

Figure 33-11. Truth table for a two-input NOR gate.

FIGURE 33–9

Using the NAND gate to generate other logic functions.

LOGIC	LOGIC SYMBOL	LOGIC FUNCTIONS USING ONLY NAND GATES
INVERTER		
AND		
OR		
NOR		
XOR		
XNOR		

Exclusive OR and NOR Gates

- Exclusive OR gate (XOR)
  - Has only two inputs
  - Produces a 1 output only if both inputs are different
- Exclusive NOR gate (XNOR)
  - Complement of the XOR gate
  - Produces a 1 output only when both inputs are the same

Exclusive OR and NOR Gates (cont'd.)

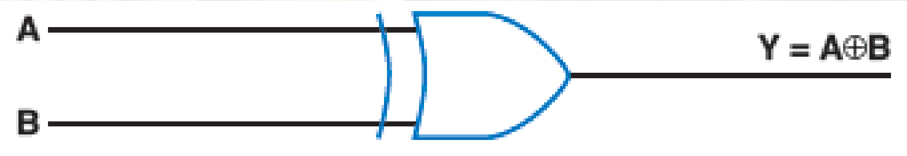


Figure 33-12. Logic symbol for an exclusive OR gate.

Exclusive OR and NOR Gates (cont'd.)

INPUTS		OUTPUT
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

Figure 33-13. Truth table for an exclusive OR gate.

Exclusive OR and NOR Gates (cont'd.)

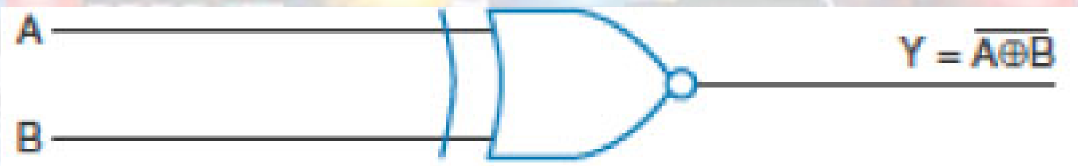


Figure 33-14. Logic symbol for an exclusive NOR gate.

Exclusive OR and NOR Gates (cont'd.)

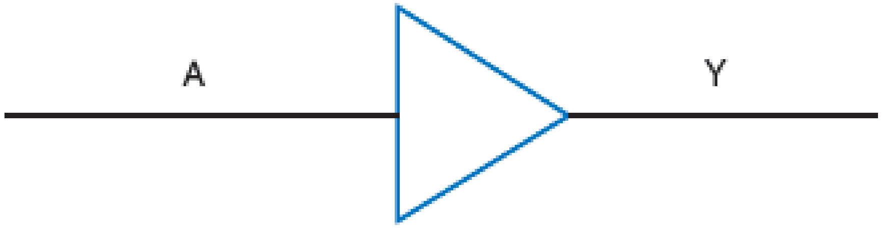
INPUTS		OUTPUT
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

Figure 33-15. Truth table for an exclusive NOR gate.

Buffer

- Isolates conventional gates from other circuitry
- Provides high-driving current for heavy circuit loads
- Provides noninverting input and output

Buffer (cont'd.)

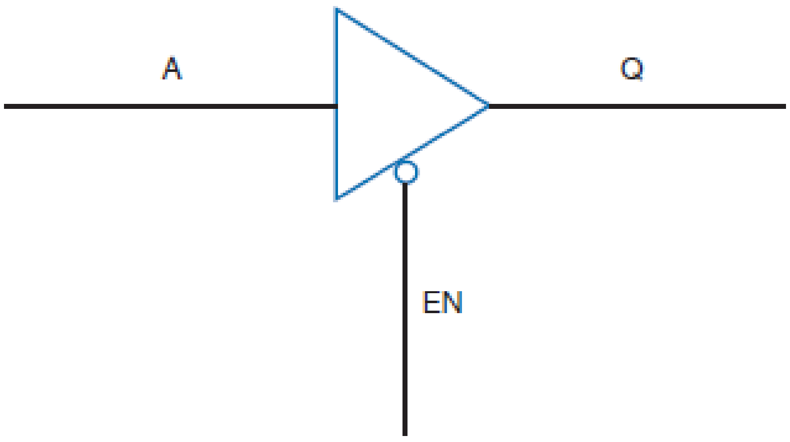


Input	Output
A	Y
0	0
1	1

Figure 33-16. Logic symbol for a buffer.

FIGURE 33–17

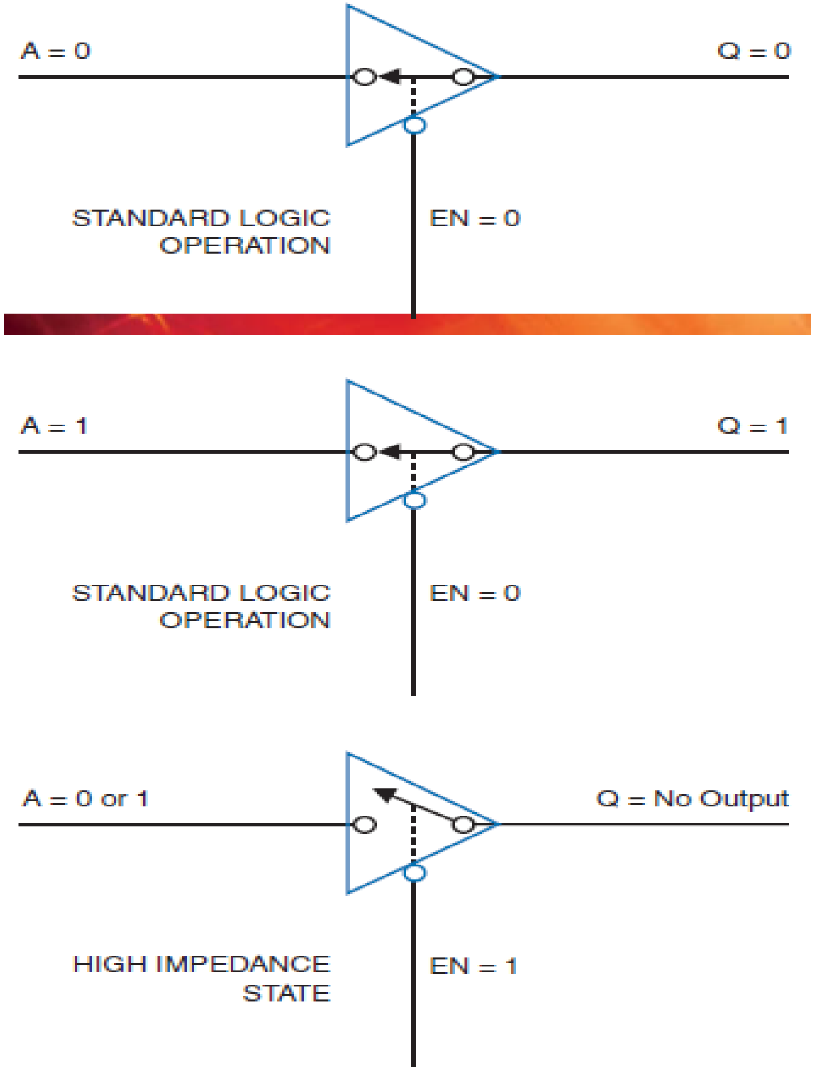
Logic symbol for a 3-state buffer.



Input		Output
A	EN	Q
0 or 1	0	0 or 1
0 or 1	1	No Output

FIGURE 33–18

3-state buffer operation.



Summary

- An AND gate produces a 1 output when all of its inputs are 1s
- An OR gate produces a 1 output if any of its inputs are 1s
- A NOT gate converts the input state to an opposite output state
- A NAND gate produces a 1 output when any of the inputs are 0s

Summary (cont'd.)

- A NOR gate produces a 1 output only when both inputs are 0s
- An exclusive OR (XOR) gate produces a 1 output only if both inputs are different
- An exclusive NOR (XNOR) gate produces a 1 output only when both inputs are the same

Chapter 34

Simplifying Logic Circuits

Objectives

- After completing this chapter, you will be able to:
  - Explain the function of Veitch diagrams
  - Describe how to use a Veitch diagram to simplify Boolean expressions
  - Explain the function of a Karnaugh map
  - Describe how to simplify a Boolean expression using a Karnaugh map

Veitch Diagrams

- Veitch Diagrams
  - Easy method for reducing a complicated expression to its simplest form
  - Can be constructed for two, three, or four variables

Veitch Diagrams (cont'd.)

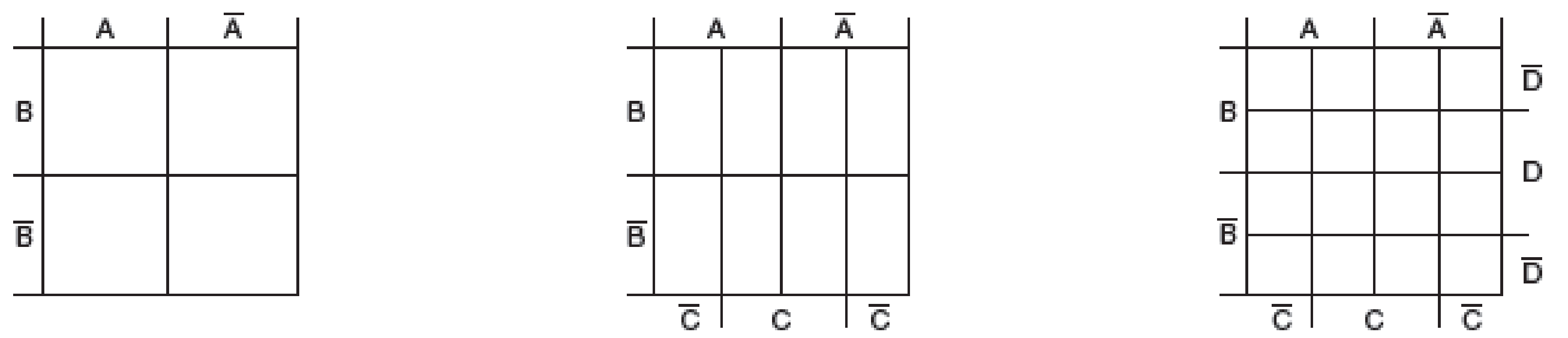
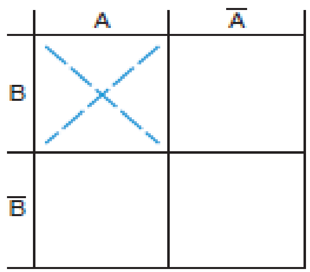


Figure 34-1. Two-, three-, and four-variable Veitch diagrams.

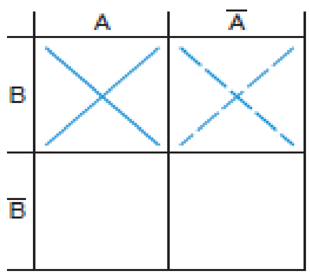
Veitch Diagrams (cont'd.)

- To use a Veitch diagram:
  - Draw the diagram based on the number of variables
  - Plot the logic function by placing an X in each square representing a term
  - Loop the groups
  - "OR" the loops with one term per loop
  - Write the simplified expression

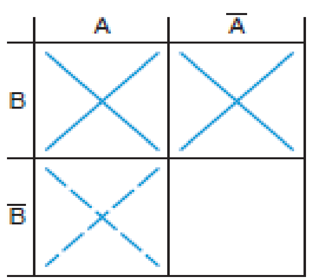
Plot 1st term AB



Plot 2nd term $\overline{A}B$

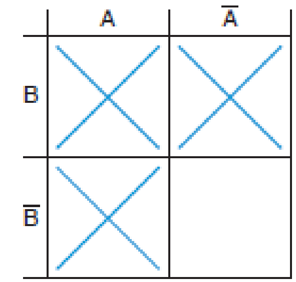


Plot 3rd term $A\overline{B}$

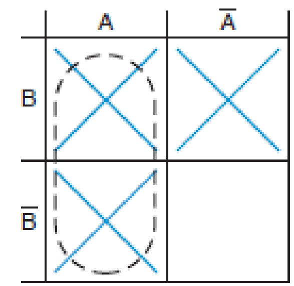


Step 3. Loop adjacent groups of X's in the largest group possible.

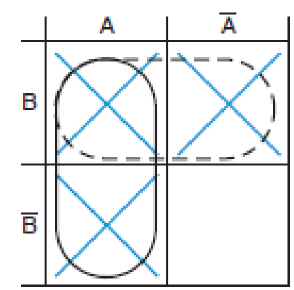
Start by analyzing chart for largest groups possible. The largest group possible here is two.



One possible group is the one indicated by the dotted line.



Another group is the one indicated by this dotted line.



Step 4. "OR" the groups: either A or B = $A + B$

Step 5. The simplified expression for $AB + \overline{A}B + A\overline{B} = Y$ is $A + B = Y$ obtained from the Veitch diagram.

Karnaugh Maps

- Karnaugh maps
  - Similar to Veitch diagrams
  - Technique for reducing complex Boolean expressions

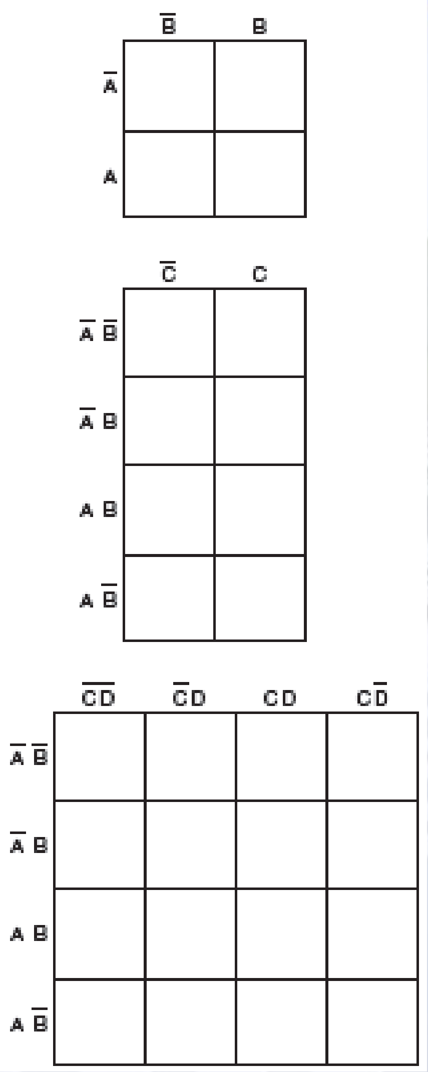


Figure 34-2. Two-, three-, and four-variable Karnaugh maps.

Karnaugh Maps (cont'd.)

- To use a Karnaugh map:
  - Draw the diagram based on the number of variables
  - Plot the logic functions by placing a "1" in each square representing a term
  - Loop adjacent groups of 1s in the largest group possible
  - "OR" the loops with one term per loop
  - Write the simplified expression

EXAMPLE: Reduce $AB + \overline{A}B + A\overline{B} = Y$ to its simplest form.

Step 1. Draw the Karnaugh map. There are two variables, A and B, so use the two-variable map.

	$\overline{B}$	B
$\overline{A}$		
A		

Step 2. Plot the logic function by placing a "1" in each square representing a term

AB – first term

$\overline{A}B$ – second term

$A\overline{B}$ – third term

Plot 1st term AB

	$\overline{B}$	B
$\overline{A}$		
A		1

Plot 2nd term $\overline{A}B$

	$\overline{B}$	B
$\overline{A}$		1
A		1

Plot 3rd term $A\overline{B}$

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

Step 3. Loop adjacent groups of 1s in the largest group possible.

Start by analyzing the map for the largest groups possible. The largest group here is two.

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

One possible group is the one indicated by the dotted line.

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

Another group is the one indicated by this dotted line.

	$\overline{B}$	B
$\overline{A}$		1
A	1	1

Step 4. "OR" the groups: either A or B = A + B.

Step 5. The simplified expression for $AB + \overline{A}B + A\overline{B} = Y$ is $A + B = Y$ obtained from the Karnaugh map.

Summary

- Veitch diagrams
  - Provide a fast and easy way to reduce complicated expressions to their simplest form
  - The simplest logic expression is obtained by looping groups of two, four, or eight X's and "OR"ing the looped terms

Summary (cont'd.)

- Karnaugh maps
  - Provide a fast and easy method to reduce complex Boolean expressions to their simplest form
  - The simplest logic expression by looping groups of two, four, or eight 1s and "OR"ing the looped terms

Homework

1. What is the binary number system ?
2. Convert the following decimal numbers to binary form:
   a) 27
   b) 12
   c) 40