Chapter 34

Simplifying Logic Circuits

Final Exam Schedule

- Final exam will be taken via **Face-to-face mode**

Date and Time: December 16, 2022, 9:30 a.m.
Building: New Silk Road, **Room No.:** 201

Instructions:
(1) Everyone should wear the mask and arrive before 10 minutes.
(2) Everyone should follow the government guidelines of COVID19.
(3) Student must write the name and student ID on top of the answer sheet.

Syllabus: Chapters- 19, 20, 21, 22, 23, 26, 28, 32, 33, & 34

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.)

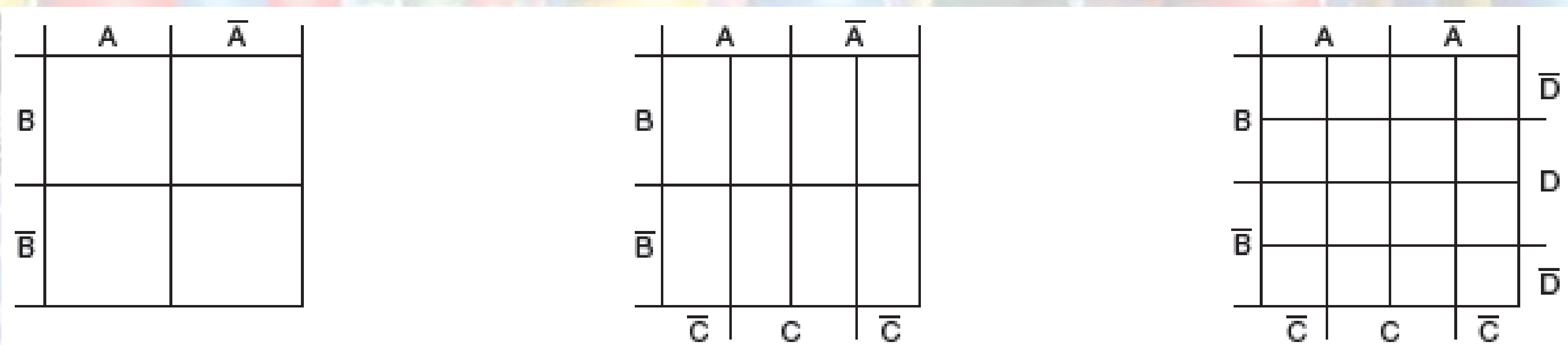
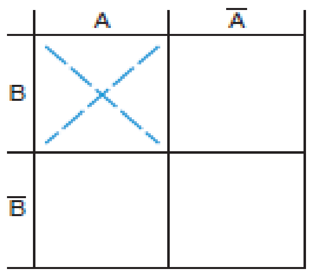


Figure 34-1. Two-, three-, and four-variable Veitch diagrams.

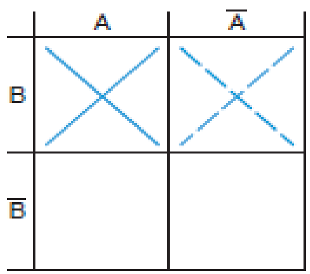
Veitch Diagrams (cont'd.)

- 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

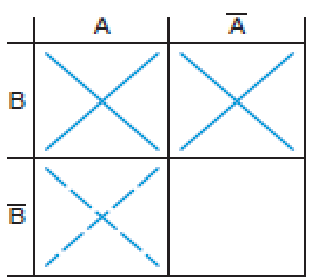
Plot 1st term AB



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

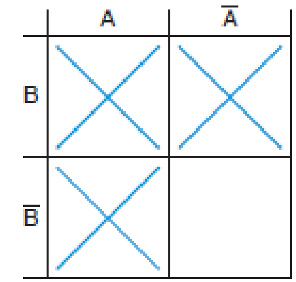


Plot 3rd term $A\bar{B}$

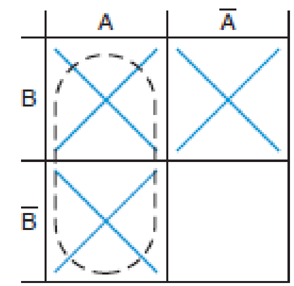


Step 3. Loop adjacent groups of X's in the largest group possible.

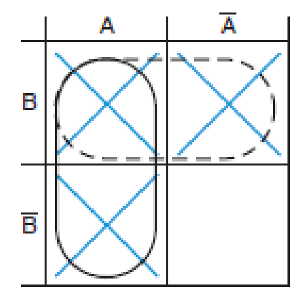
Start by analyzing chart for largest groups possible. The largest group possible here is two.



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 $AB + \bar{A}B + A\bar{B} = Y$ is $A + B = Y$ obtained from the Veitch diagram.

Karnaugh Maps

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

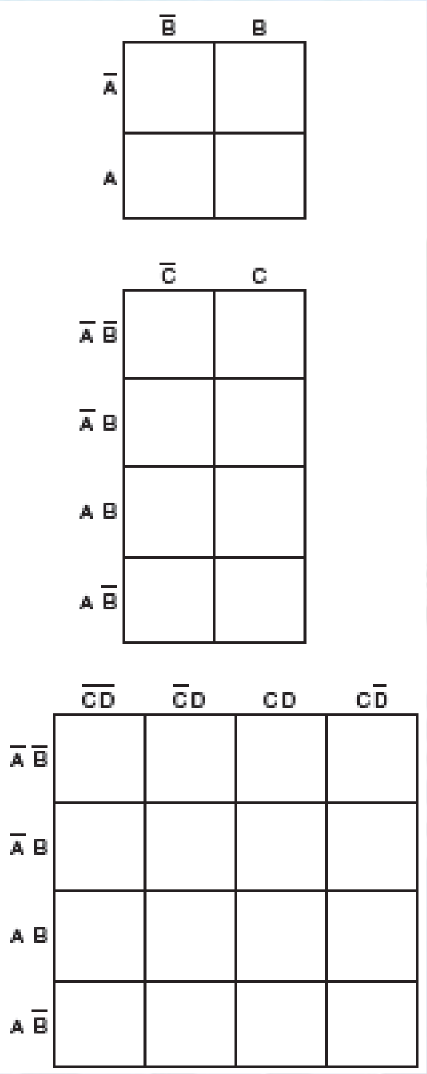


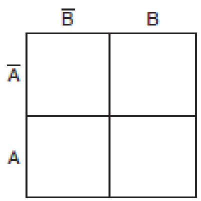
Figure 34-2. Two-, three-, and four-variable Karnaugh maps.

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.



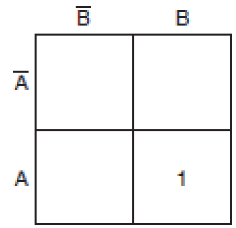
Step 2. Plot the logic function by placing a "1" in each square representing a term

AB – first term

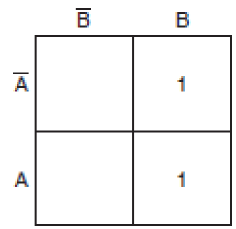
$\overline{A}B$ – second term

$A\overline{B}$ – third term

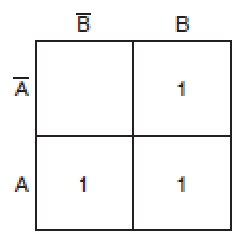
Plot 1st term AB



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

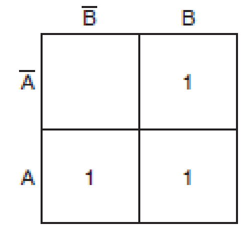


Plot 3rd term $A\overline{B}$

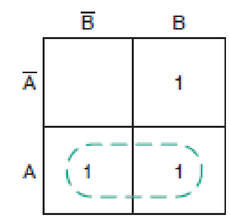


Step 3. Loop adjacent groups of 1s in the largest group possible.

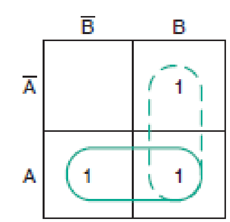
Start by analyzing the map for the largest groups possible. The largest group here is two.



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 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

Chapter 35

Sequential Logic Circuits

Objectives

- After completing this chapter, you will be able to:
  - Describe the function of a flip-flop
  - Identify the basic types of flip-flops
  - Draw the symbols used to represent flip-flops
  - Describe how flip-flops are used in digital circuits

Objectives (cont'd.)

- Describe how a counter and shift register operate
- Identify the different types of counters and shift registers
- Draw the symbols used to represent counters and shift registers
- Identify applications of counters and shift registers

Flip-Flops

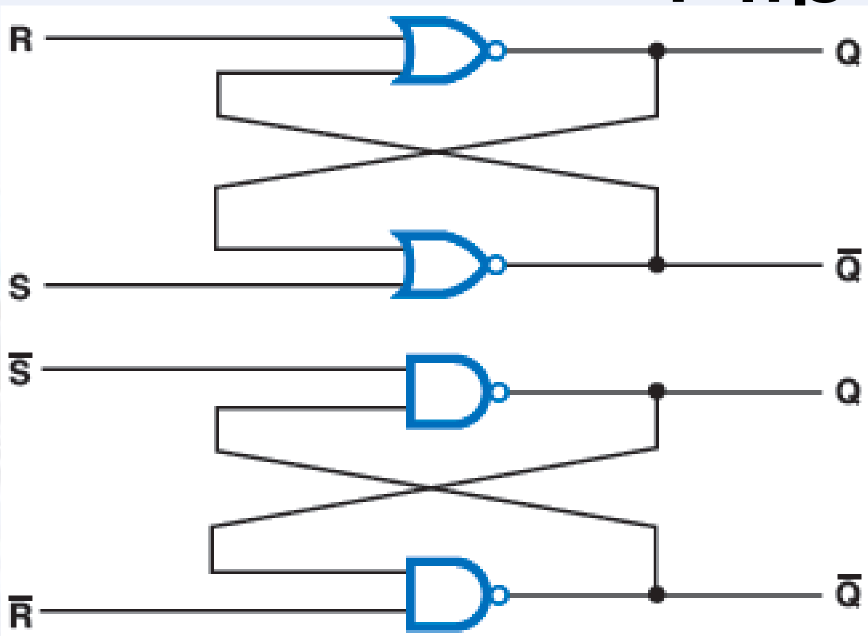


Figure 35-1. Basic flip-flop circuit.

Truth table for an RS flip-flop.

INPUTS		OUTPUTS	
S	R	Q	$\overline{Q}$
0	0	NC	NC
0	1	0	1
1	0	1	0
1	1	0	0

NC = No Change

- **Flip-flop** is a bistable multivibrator whose output is either a high or low voltage, a 1 or a 0.
- Basic flip-flop is the *RS flip-flop. And* is formed by two cross-coupled NOR or NAND gates
- RS flip-flop has two outputs, Q and , and two controlling inputs, R (Reset) and S (Set).
- The outputs are always opposite or complementary

Flip-Flops (cont'd.)

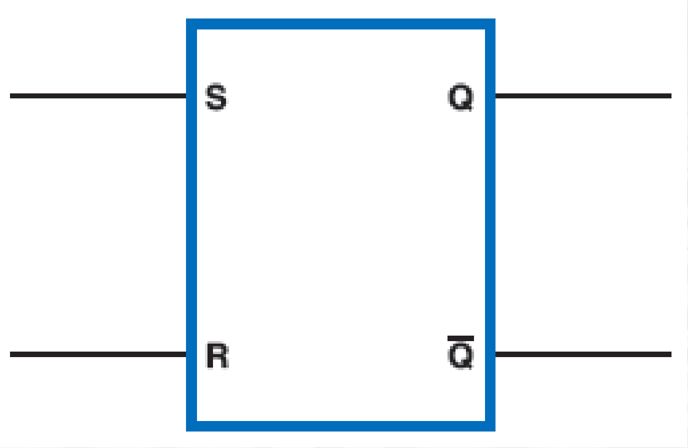


Figure 35-3. Logic symbol for an RS flip-flop.

Flip-Flops (cont'd.)

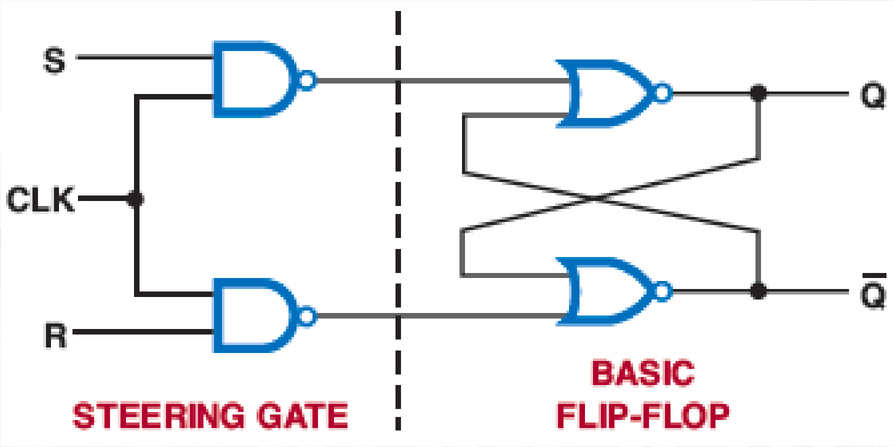


Figure 35-4. Logic circuit for a clocked RS flip-flop.

- Clocked flip-flop is controlled by the logic state of the S and R inputs when a clock pulse is present.

Flip-Flops (cont'd.)

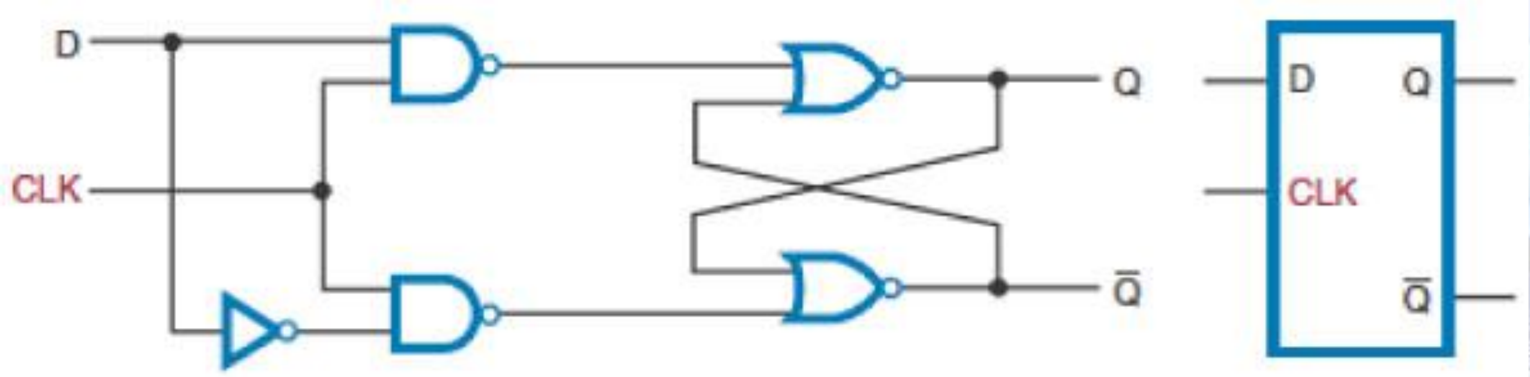


Figure 35-6. Logic circuit and symbol for the D flip-flop.

- The D input is delayed one clock pulse from getting to the output (Q) and D flip-flop has a PS (preset) input and CLR (clear) input.

Flip-Flops (cont'd.)

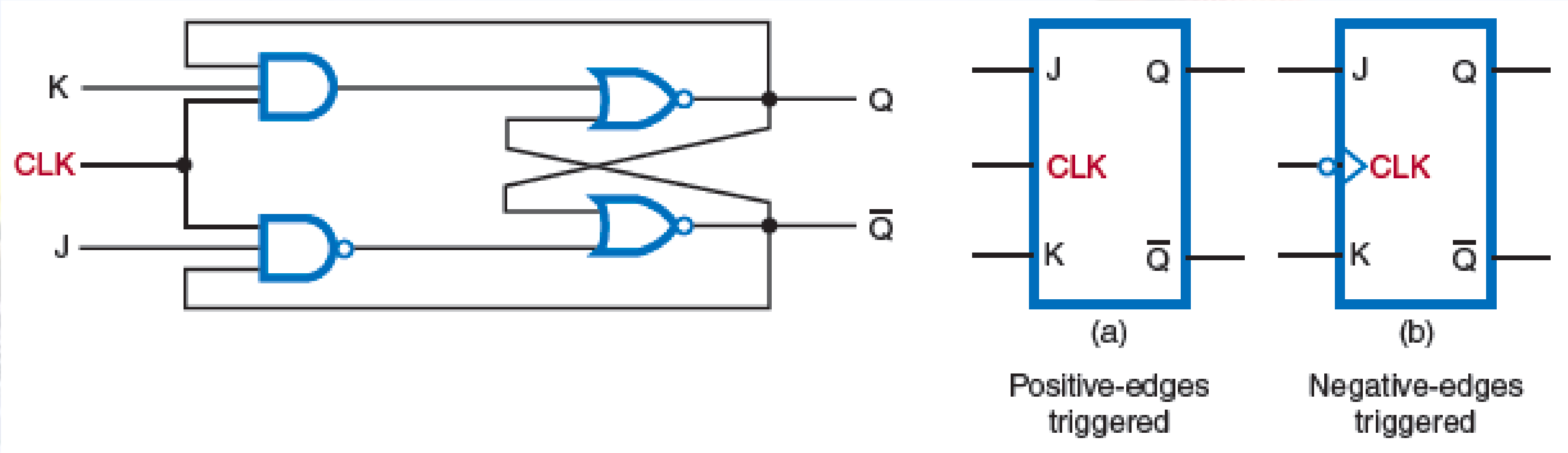
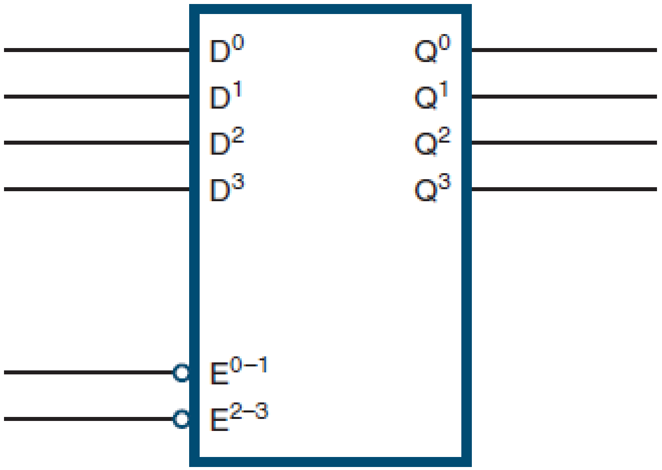


Figure 35-7. Logic circuit and symbol for the JK flip-flop.

- **JK flip-flop** is the most widely used flip-flop
- J and K inputs that are present at the active clock edge (high to low or low to high) that allows the accepting of input data on J and K at a precise instant.

FIGURE 35–8
Four-bit latch.



- **Latch** is a device that serves as a temporary buffer memory and used to hold data after the input signal is removed.
- Without a latch, the information being displayed is removed when the input signal is removed. With the latch, the information is displayed until it is updated.

Counters

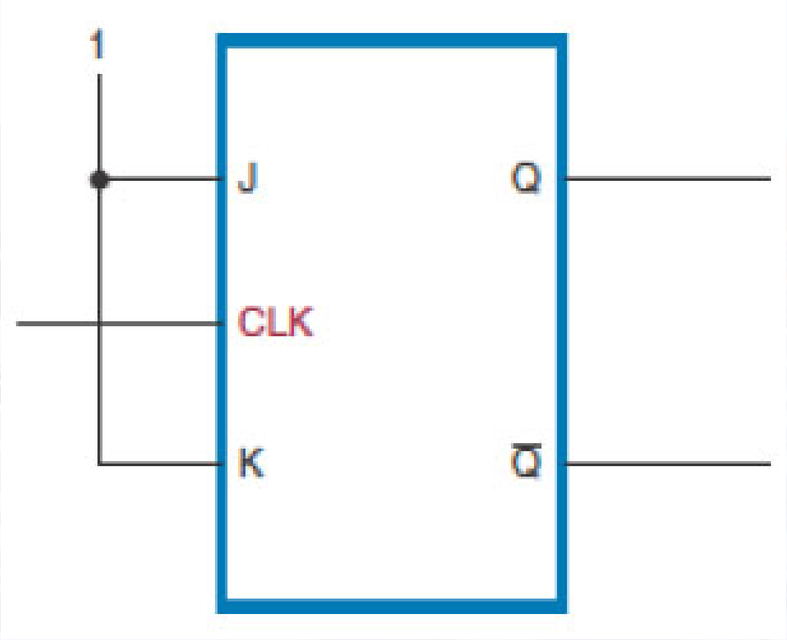


Figure 35-9. JK flip-flop set up for counting.

- A counter is a logic circuit that can count a sequence of numbers or states when activated by a clock input.
- Initially, the flip-flop is reset, the first clock pulse causes it to set (Q = 1). The second clock pulse causes it to reset (Q = 0)

Counters (cont'd.)

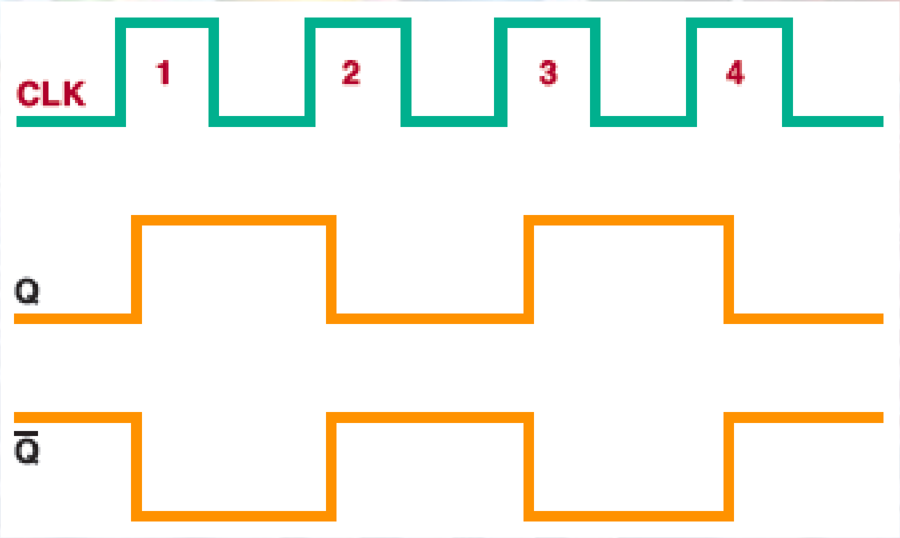


Figure 35-10. Input and output waveforms of JK flip-flop set up as a counter.

Counters (cont'd.)

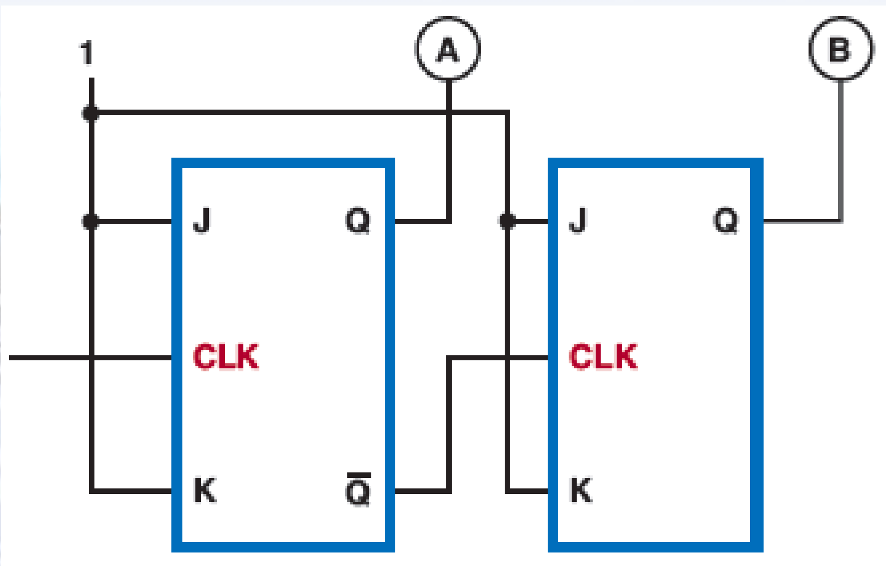
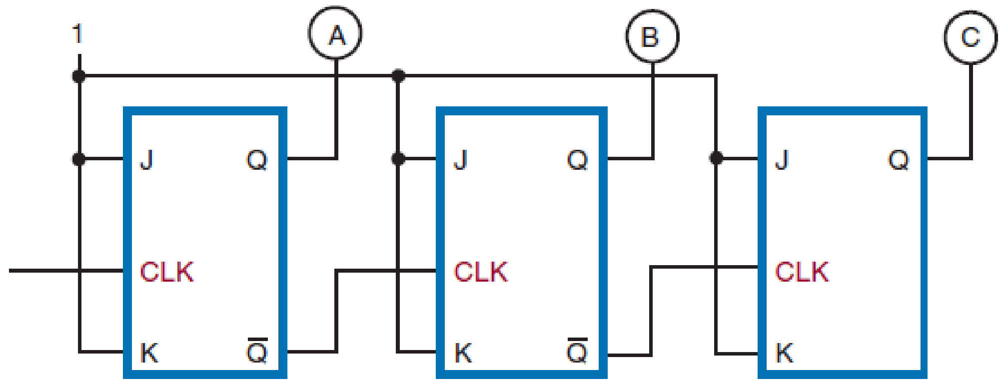


Figure 35-11. Two-stage counter.

Two-stage counter connected for asynchronous operation. Each flipflop in a counter is referred to as a *stage*

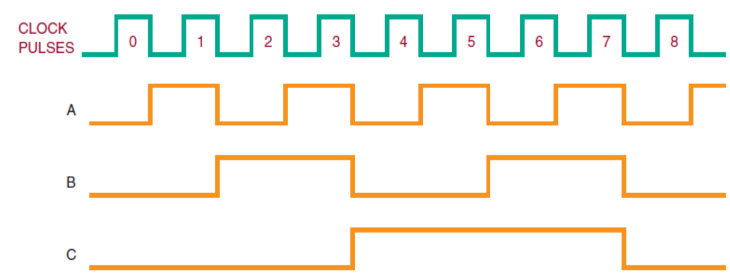
FIGURE 35–12

Three-stage binary counter.



NUMBER OF CLOCK PULSES	BINARY COUNT SEQUENCE			DECIMAL COUNT
	C	B	A	
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

COUNT SEQUENCE



Counters (cont'd.)

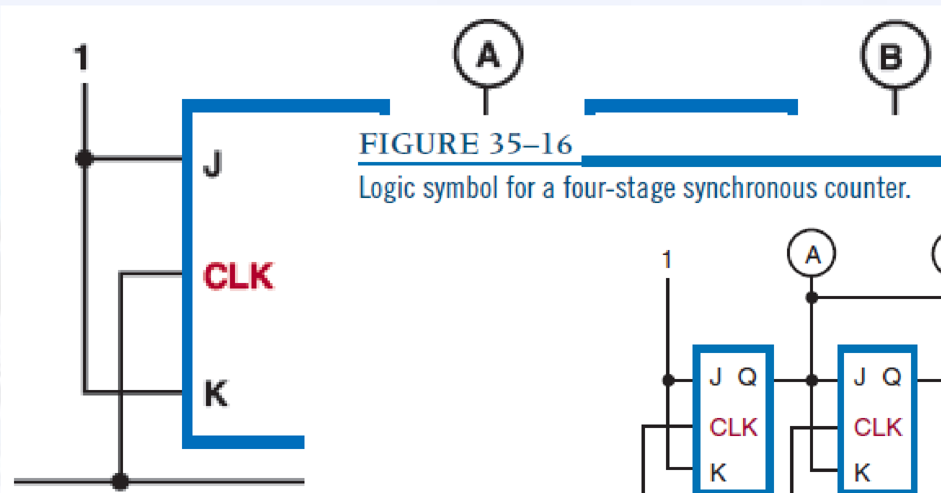
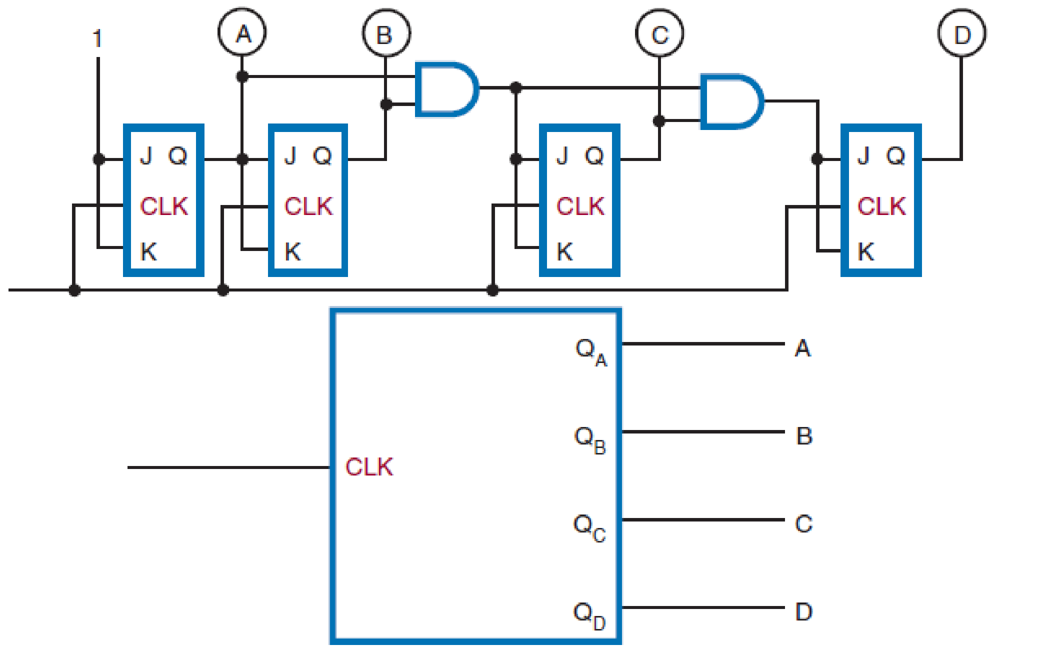


Figure 35-13. T

FIGURE 35–16
Logic symbol for a four-stage synchronous counter.



- A *synchronous counter* is a counter in which each … at the … ned by … k input … of the … nter is … *parallel* … e clock input is connected in parallel to each flip-flop.

Counters (cont'd.)

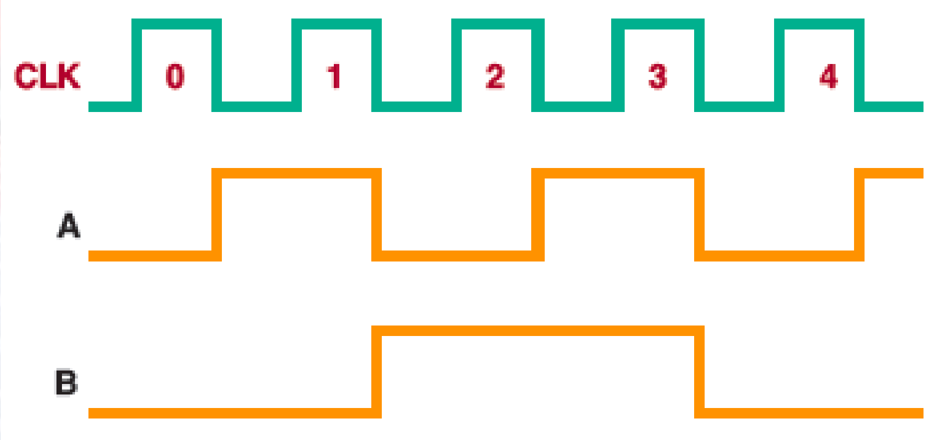


Figure 35-14. Input and output waveforms for the two-stage synchronous counter.

A four-stage binary counter is a divide-by-sixteen device with the output equal to one-sixteenth the input clock frequency

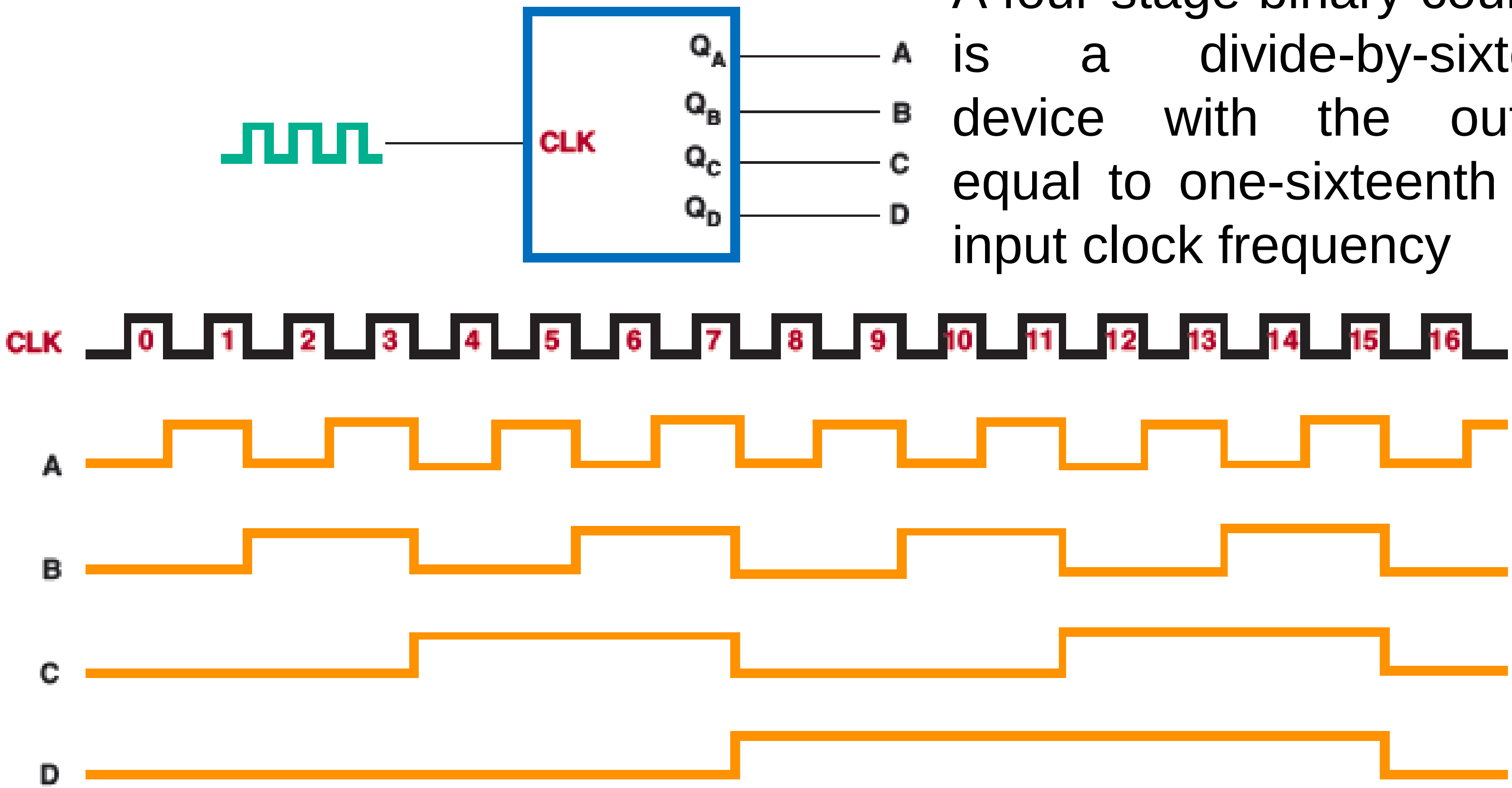


Figure 35-17. A counter as a frequency divider.

Counters (cont'd.)

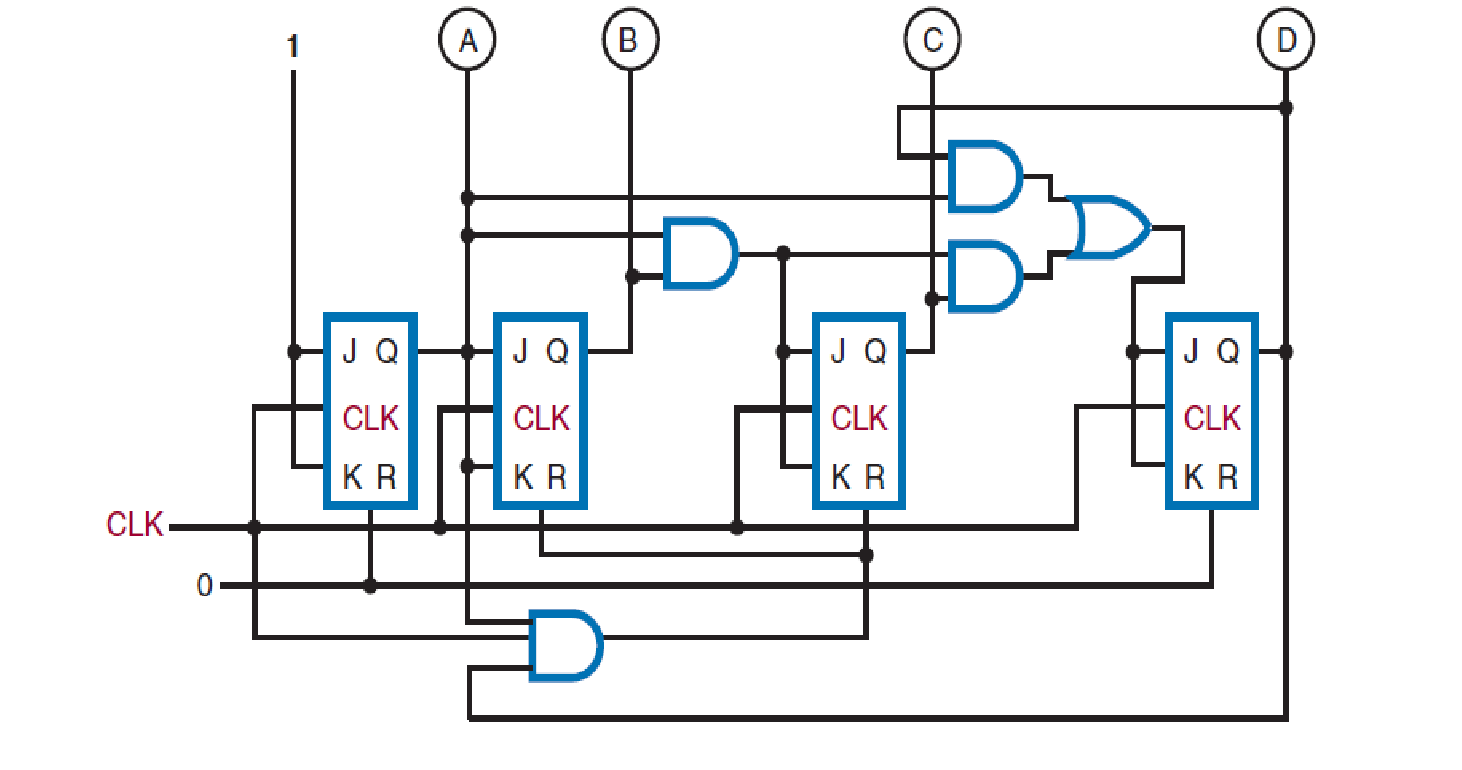


Figure 35-18. Synchronous BCD decade counter.

- **Decade counters** have a modulus of ten, or ten states in their counting sequence. A common decade counter is the BCD (8421) counter, which produces a binary-coded-decimal sequence

Counters (cont'd.)

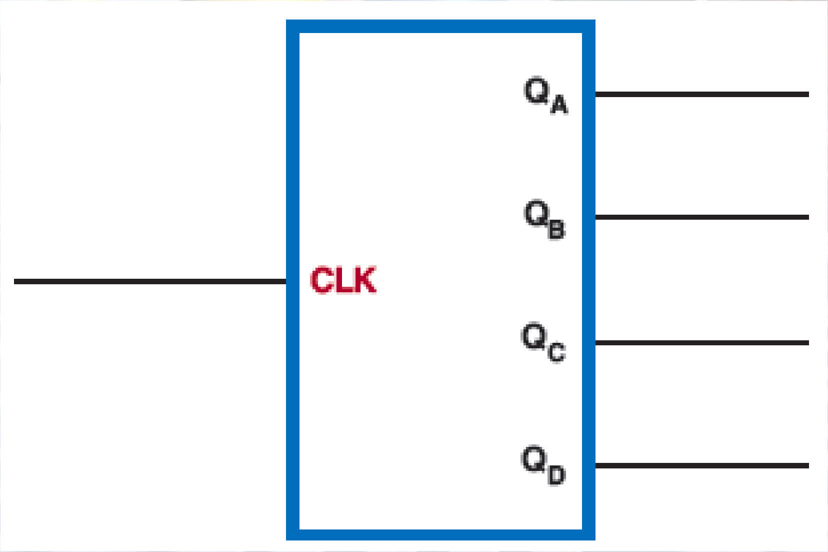


Figure 35-19. Logic symbol for a decade counter.

Counters (cont'd.)

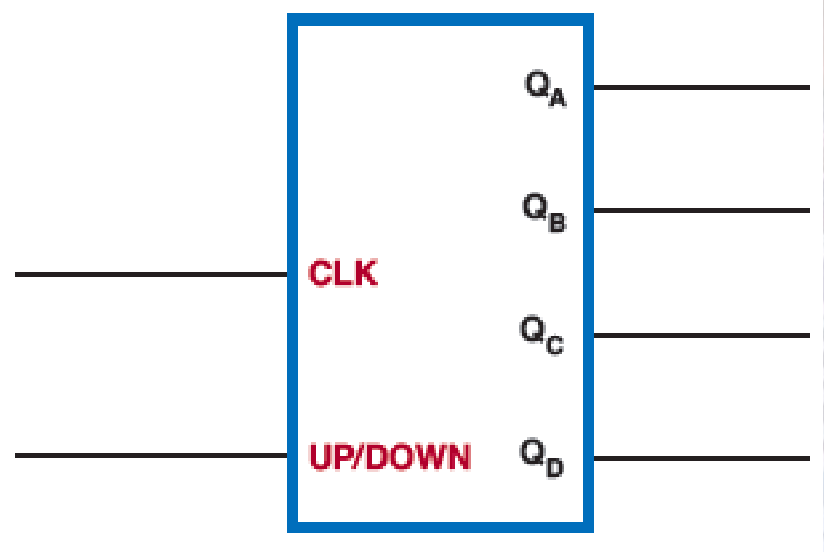


Figure 35-20. Logic symbol for an up-down counter.

Shift Registers

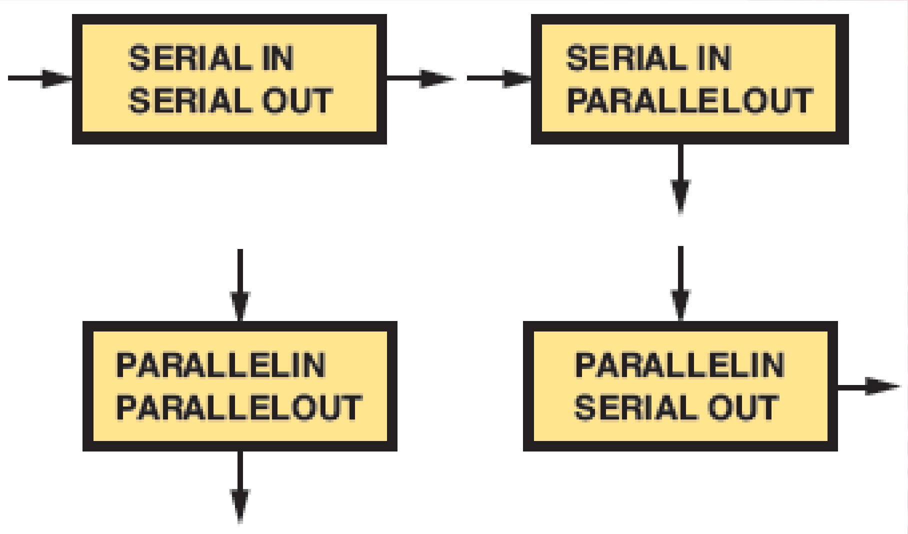


Figure 35-23. Methods of loading and reading data in a shift register.

Shift Registers (cont'd.)

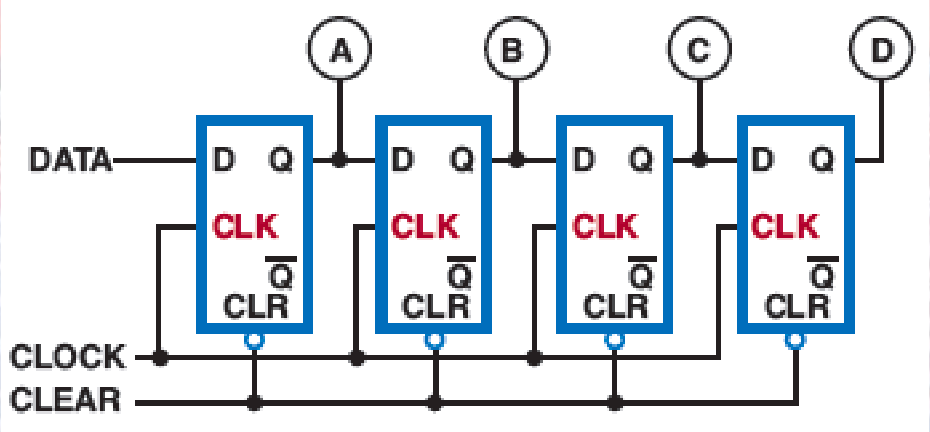


Figure 35-24. Shift register constructed from four flip-flops.

Shift Registers (cont'd.)

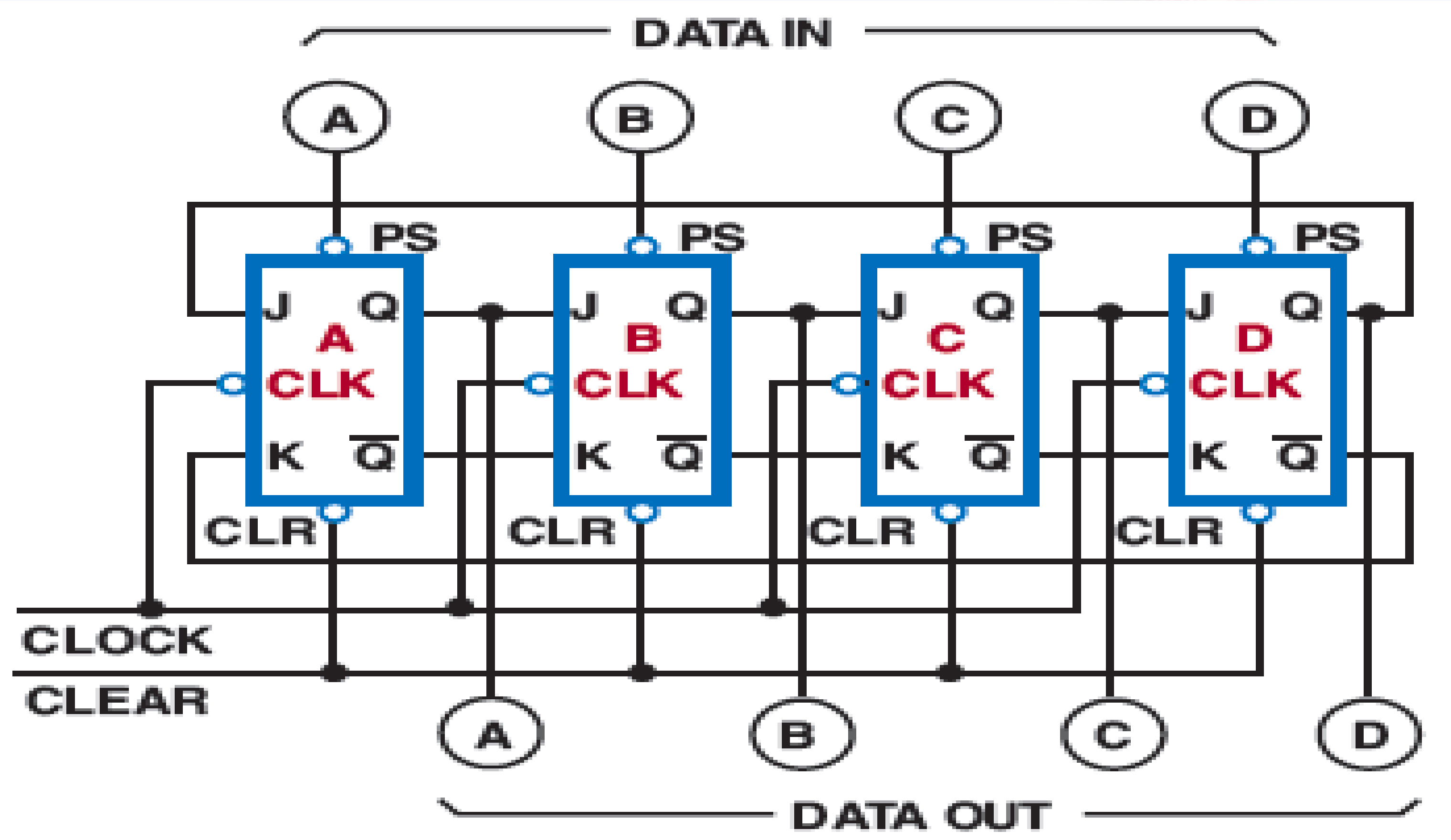


Figure 35-27. Loading a shift register using parallel input.

Shift Registers (cont'd.)

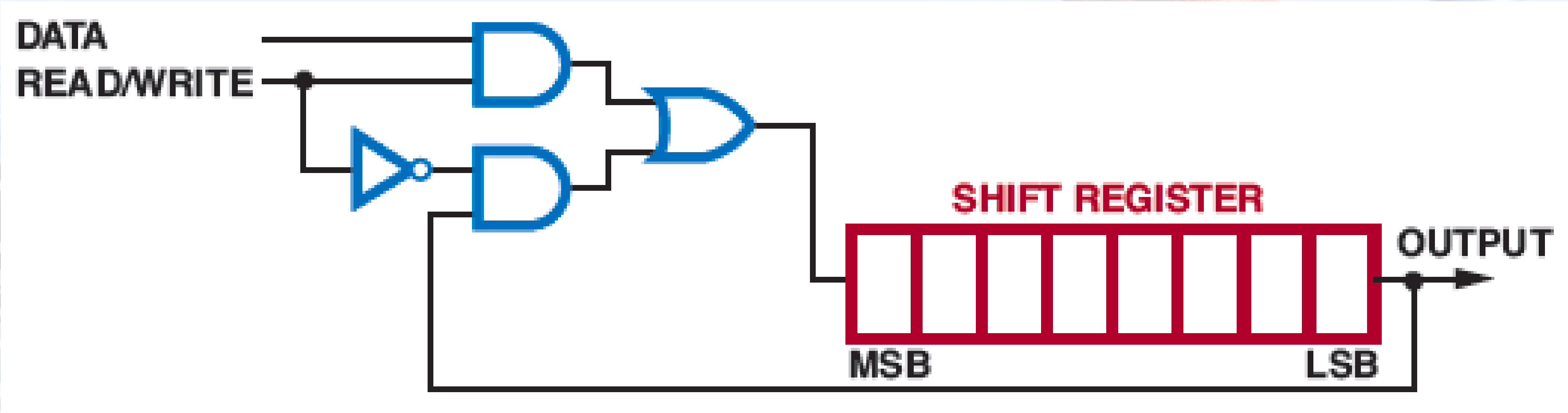


Figure 35-28. Shift register circuitry for maintaining and reading data.

Memory

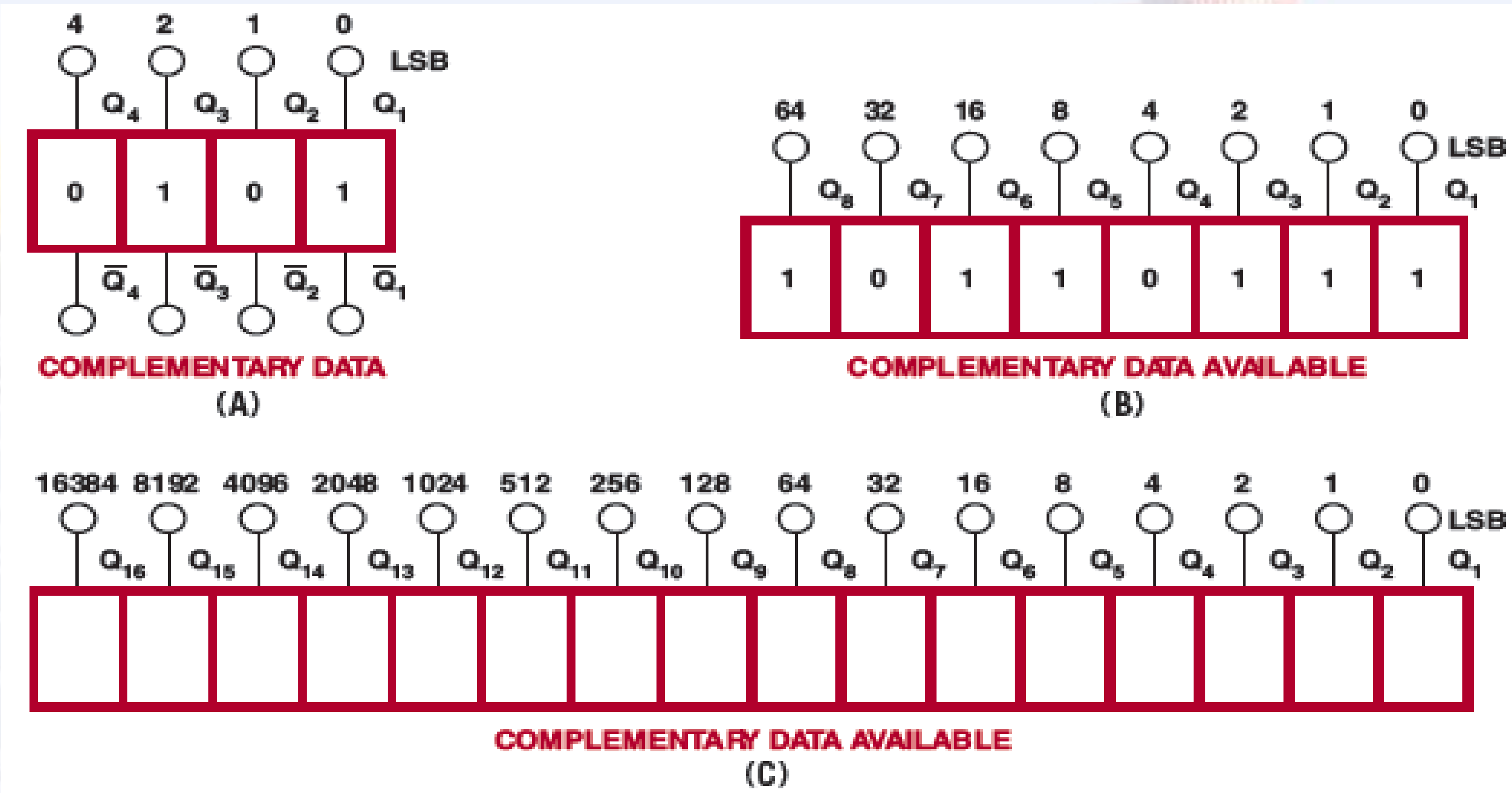
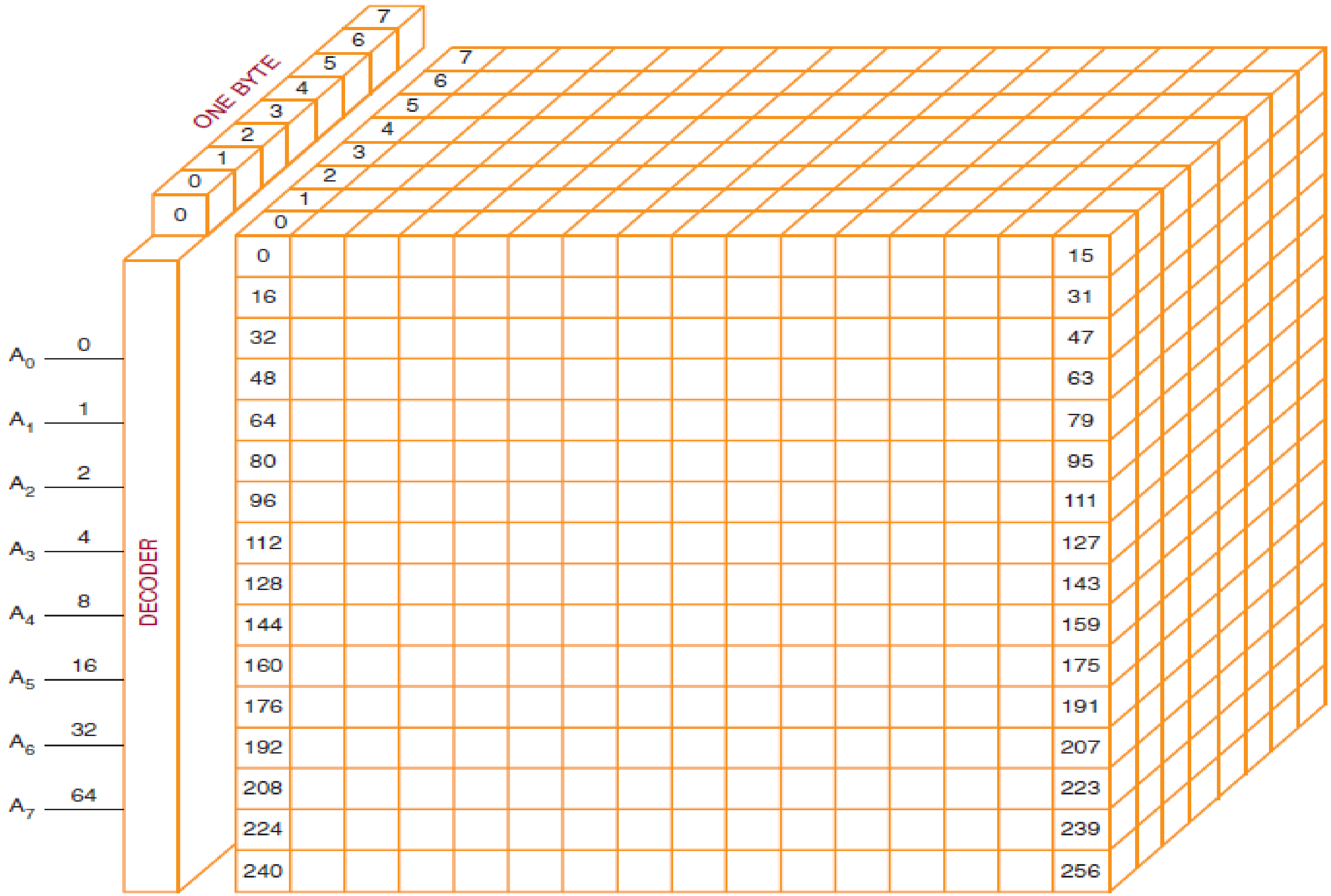


Figure 35-29. Four-, eight-, and sixteen-bit registers.

FIGURE 35–30
Large memory array (eight bit).



Summary

- Flip-flop
  - A bistable multivibrator whose output is either a high or a low
  - Types include RS, Clocked RS, D, JK
  - Used in digital circuits such as counters
- Counter
  - A logic circuit that can count a sequence of numbers or states
  - Asynchronous or synchronous

Summary (cont'd.)

- Shift register
  - Used to store data temporarily
  - Can move data to the left or right
- Memory
  - Stores data temporarily or permanently

Chapter 36

Combinational Logic Circuits

Objectives

- After completing this chapter, you will be able to:
  – Describe the functions of encoders, decoders, multiplexers, adders, subtractors, and comparators
  – Identify the schematic symbols for encoders, decoders, multiplexers, adders, subtractors, and comparators

Objectives (cont'd.)

- Identify applications for combinational logic circuits
- Develop truth tables for the different combinational logic circuits

Combinational logic circuits are circuits that combine the basic components of AND gates, OR gates, and inverters to produce more sophisticated circuits.

Encoders

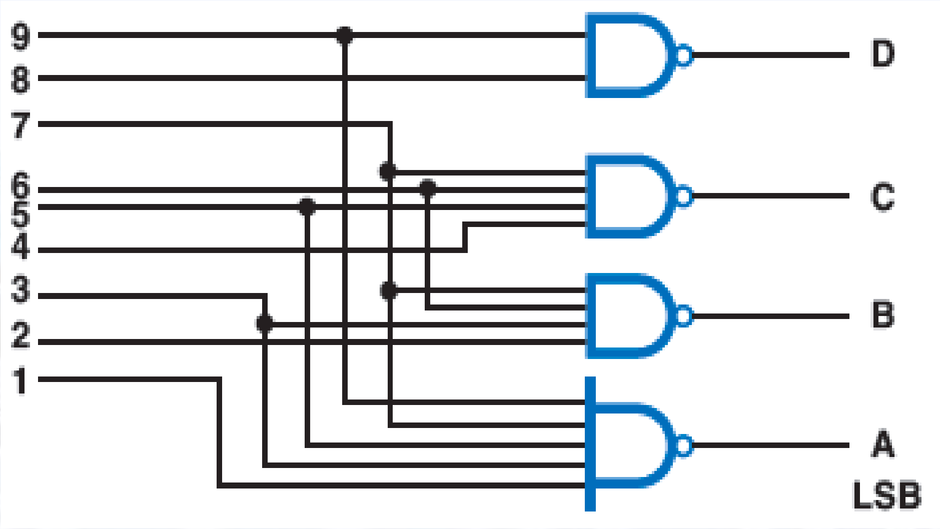
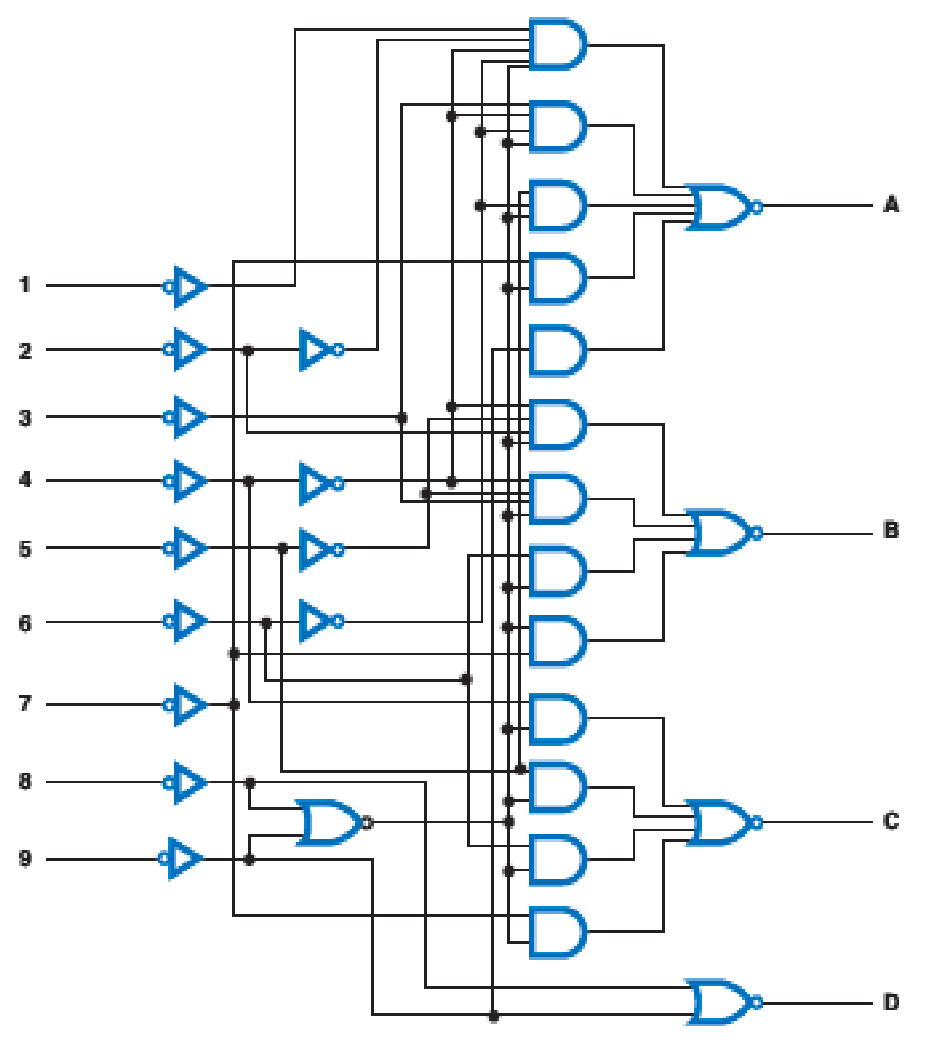


Figure 36-1. Decimal-to-binary encoder.

- A combinational logic circuit that accepts one or more inputs and generates a multibit binary output.
- *Encoding* is the process of converting any keyboard character or number as input to a coded output such as a binary or BCD form

- a 10-line-to-4-line encoder
- Digit 4 on the keyboard that produces low or a 0 on line 4, and 4-bit code 0100 as an output.



- Priority function means that if two keys are pressed simultaneously the encoder produces a BCD output corresponding to the highest order decimal digit appearing on the input

Figure 36-2. Decimal-to-binary priority encoder.

Encoders (cont'd.)

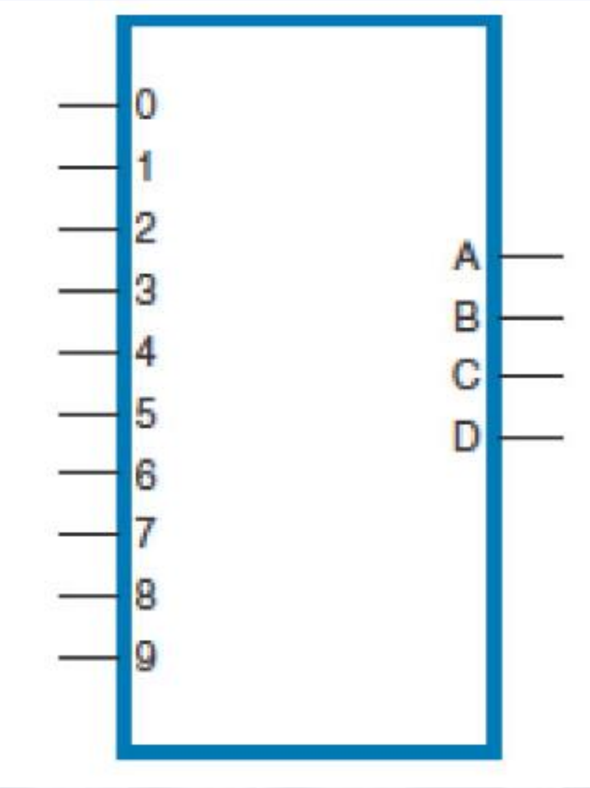


Figure 36-3. Logic symbol for a decimal-to-binary priority encoder.

- This type of encoder is used to translate the decimal input from a keyboard to an 8421 BCD code.
- Includes calculators, computer keyboard inputs, electronic typewriters, and teletypewriters (TTY)

Decoders

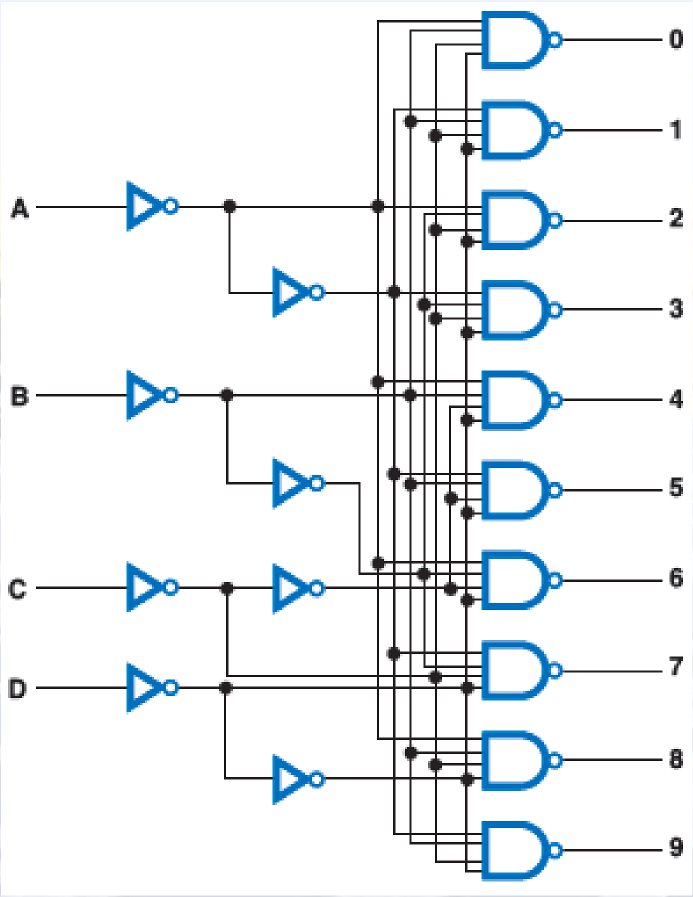


Figure 36-4. Binary-to-decimal decoder.

- Ten NAND gates required for decoding a 4-bit BCD number to its approximate output (one decimal digit).
- When all the inputs to a NAND gate are 1, its output is 0

Decoders (cont'd.)

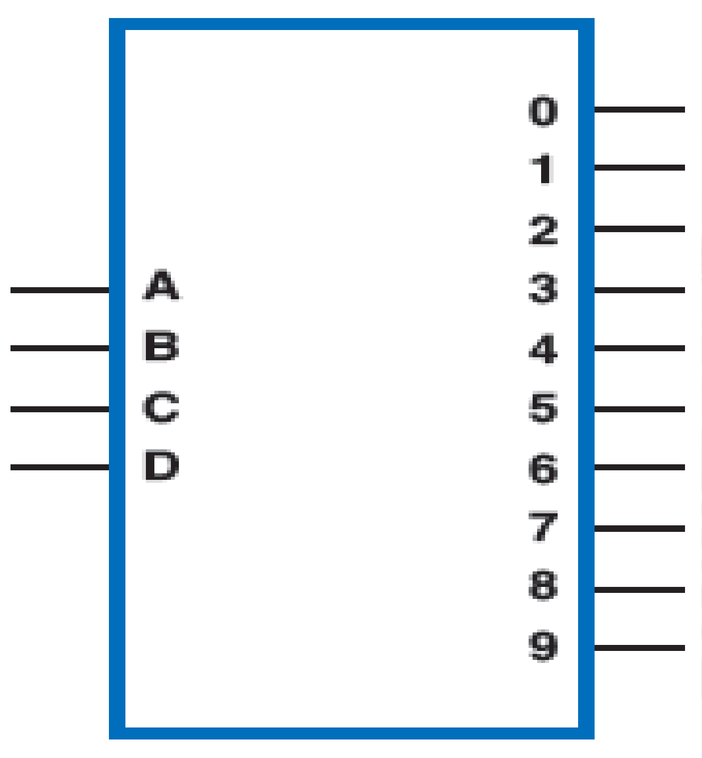
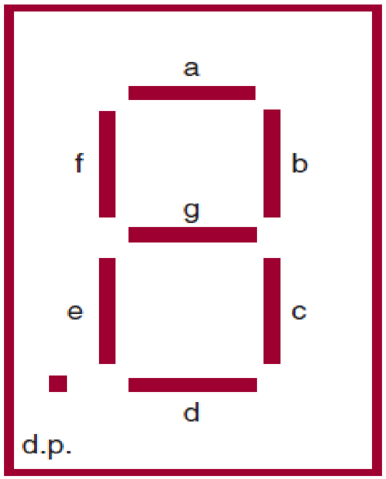


Figure 36-5. Logic symbol for a binary-to-decimal decoder.

FIGURE 36–7
Seven-segment display configuration.



A special type of decoder is the standard 8421 BCD-to-seven-segment decoder. It accepts a BCD input code and generates a special 7-bit output code to energize a seven-segment decimal readout display

FIGURE 36–8
Using the seven-segment display to form the ten decimal digits.

- Consists of seven LED segments that are lit in different combinations to produce each of the ten decimal digits, 0 through 9

Multiplexers

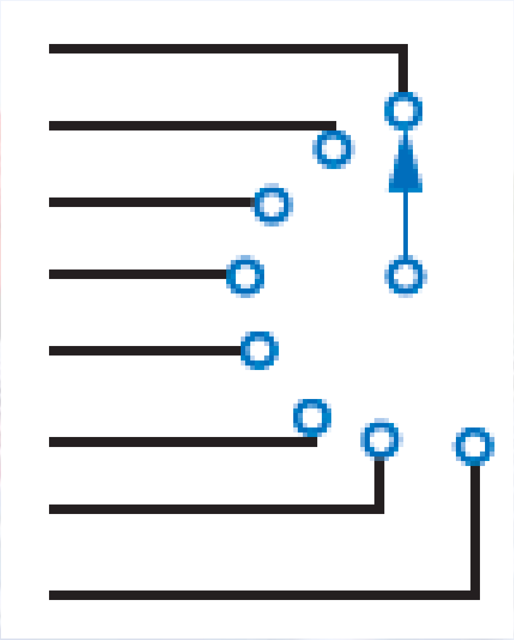


Figure 36-12. A single-pole, multiposition switch can be used as a multiplexer.

- A **multiplexer** is a circuit used to select and route any one of several input signals to a single output.
- Multiplexers handle two basic types of data: analog and digital.
- For analog applications, multiplexers are built of relays and transistor switches.
- For digital applications, multiplexers are built from standard logic gates

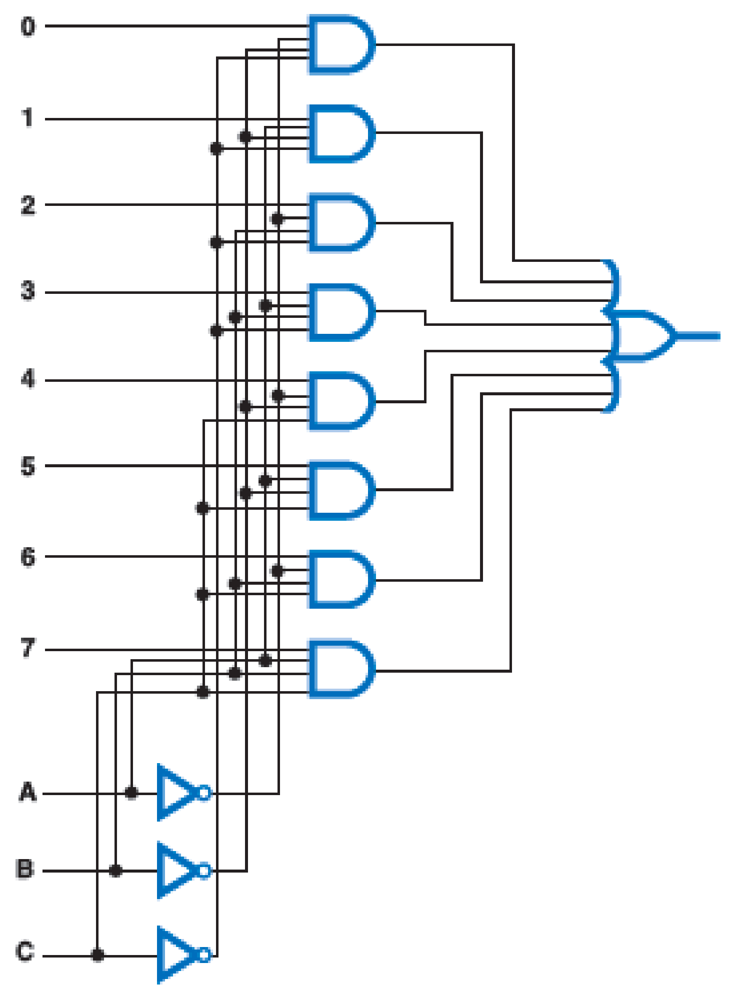


Figure 36-13. Logic circuit for an eight-input multiplexer.

Multiplexers (cont'd.)

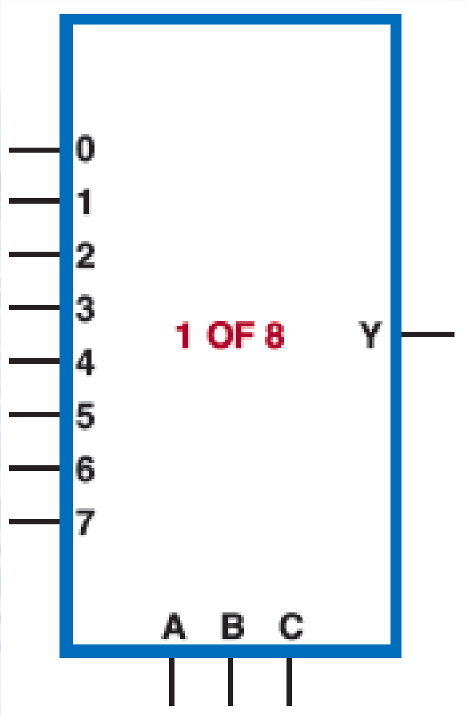


Figure 36-14. Logic symbol for an eight-input multiplexer.

Multiplexers (cont'd.)

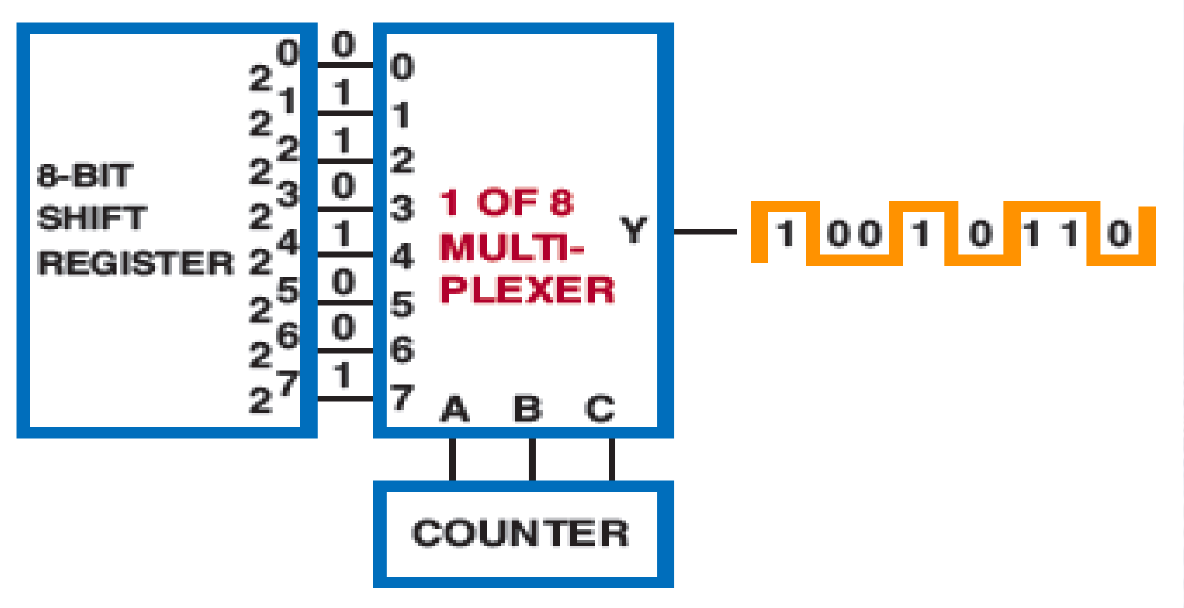


Figure 36-16. Using a multiplexer for parallel-to-serial conversion.

- A 3-bit binary input word from a counter is used to select the desired input.
- The parallel input word is connected to each of the input lines of the multiplexer.

Arithmetic Circuits

INPUTS		OUTPUTS	
A	B	Σ	C_0
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

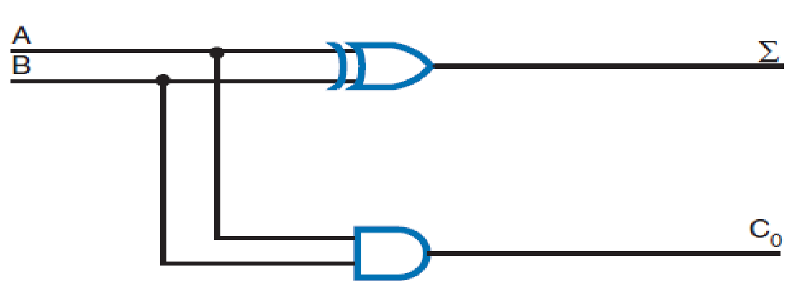
Figure 36-17. Truth table constructed using addition rules.

Adders are designed to work in either serial or parallel circuits.

To understand how an adder works it is necessary to review the rules for adding:

0	0	1		1
+ 0	+ 1	+ 0		+ 1
0	1	1	Carry 1	0

FIGURE 36–20
Half-adder circuit.



Arithmetic Circuits (cont'd.)

INPUTS			OUTPUTS	
A	B	C_{1N}	Σ	C_0
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

C_1 input represents the carry input. The C_0 output represents the carry output.

Figure 36-21. Truth table for a full adder.

Arithmetic Circuits (cont'd.)

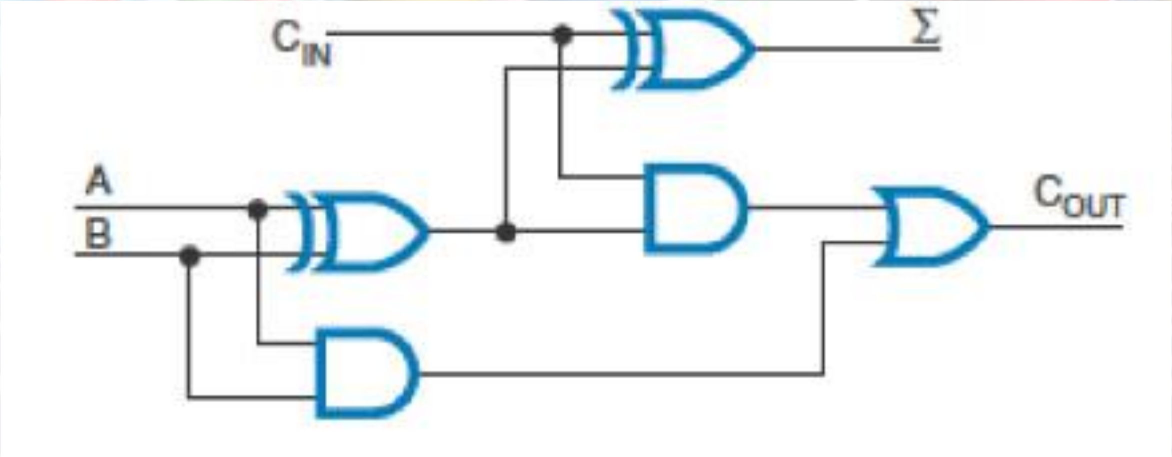


Figure 36-22. Logic circuit for a full adder using two half adders.

Arithmetic Circuits (cont'd.)

FIGURE 36–24
Four-bit parallel adder.

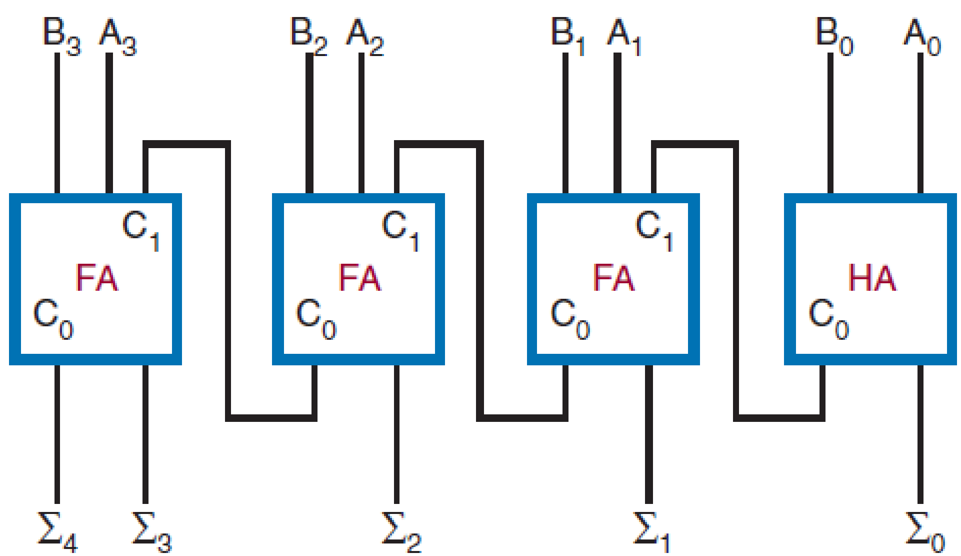


Figure 3 … (B).

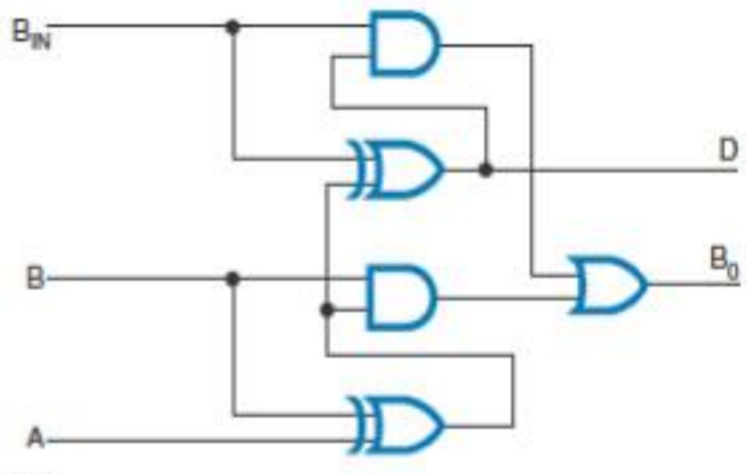
Arithmetic Circuits (cont'd.)

INPUTS		OUTPUTS	
A	B	D	B_0
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

Figure 36-25. Truth table constructed using subtraction rules.

- The letter *D* represents the difference column. The borrow column is represented by B_0.

Arithmetic Circuits (cont’d.)



INPUT			OUTPUT	
A	B	B_{IN}	D	B_0
0	0	0	0	0
1	0	0	1	0
0	1	0	1	1
1	1	0	0	0
0	0	1	1	1
1	0	1	0	0
0	1	1	0	1
1	1	1	1	1

(B)

Figure 36-27. Logic circuit (A) and truth table (B) for a full subtractor.

Arithmetic Circuits (cont'd.)

FIGURE 36–29

Four-bit subtractor.

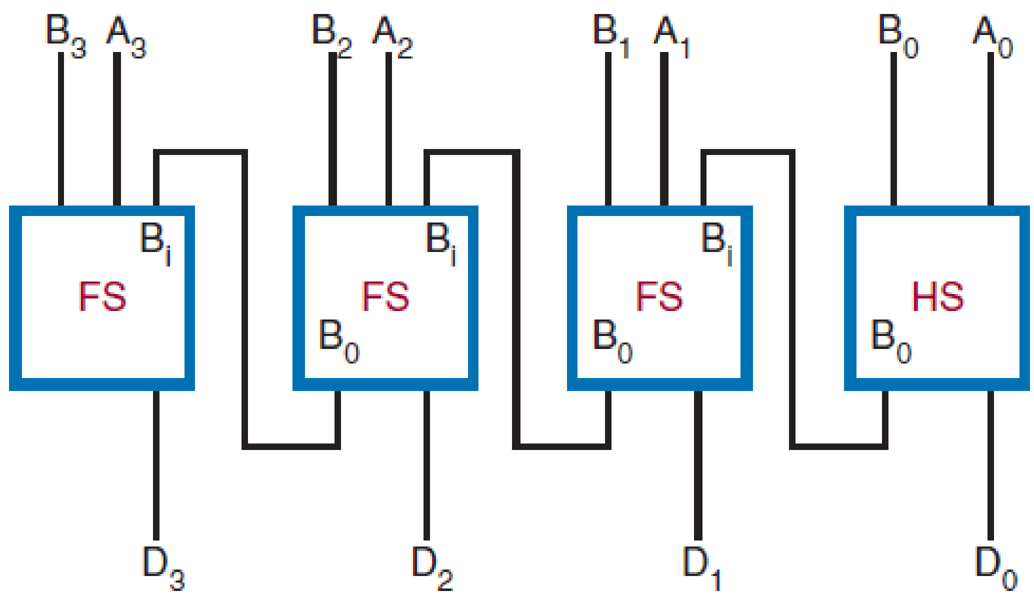


Figure 36-28. Logic symbols for half subtractor (A) and full subtractor (B).

Arithmetic Circuits (cont'd.)

INPUT		OUTPUT
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

Figure 36-30. Truth table for a comparator.

A **comparator** is used to compare the magnitudes of two binary numbers

Arithmetic Circuits (cont'd.)

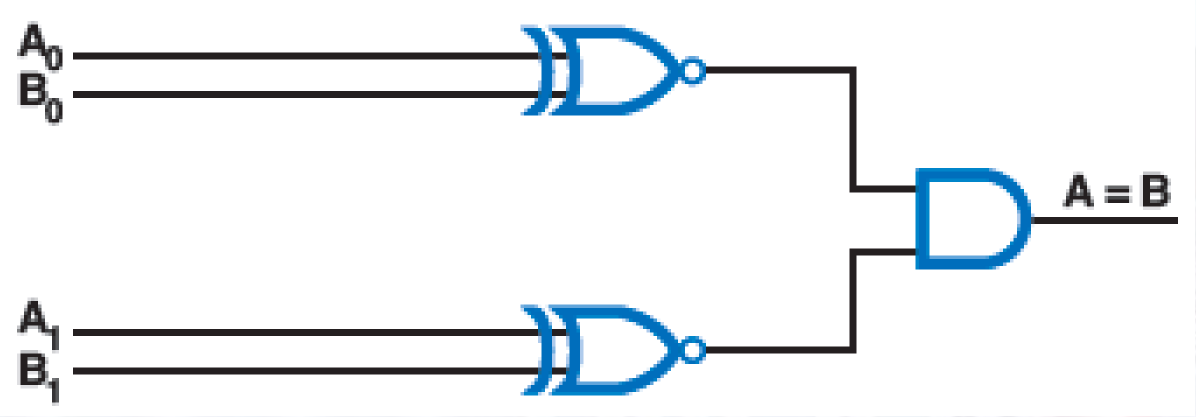
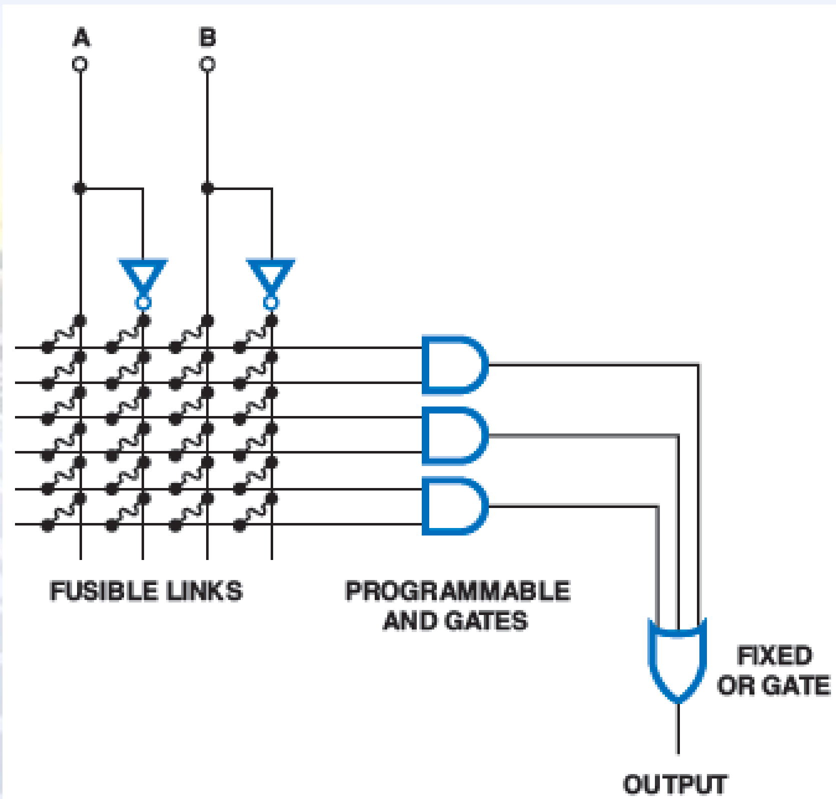


Figure 36-31. Comparing two 2-bit numbers.

Programmable Logic Devices



using Veitch diagrams or Karnaugh maps. For example:

$$\bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \bar{A}B\bar{C}D = Y$$

reduced to its simplest form became:

$$\bar{A}D + A\bar{B}\bar{C} = Y$$

Figure 36-34. Programmable array logic (PAL) architecture.

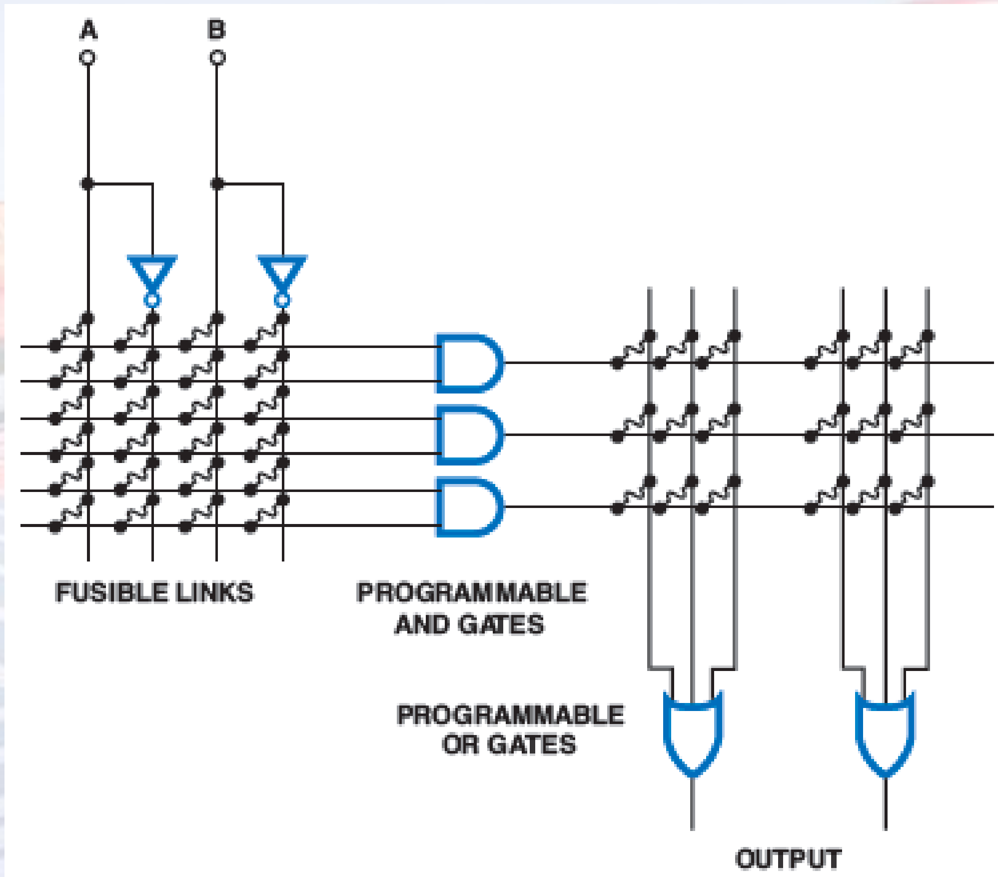


Figure 36-35. Programmable logic array (PLA) architecture.

Summary

- An encoder accepts one or more inputs and generates a multibit binary output
- A decoder processes a complex binary code into a digit or character that is easy to recognize
- A multiplexer allows digital data from several sources to be routed through a common line for transmission to a common destination

Summary (cont'd.)

- The truth table for the adding rules of binary numbers is equivalent to the truth table for an AND gate and an XOR gate
- A comparator is used to compare the magnitudes of two binary numbers
- Programmable logic devices (PLDs) are used to implement complex logic functions

Chapter 37

Microcomputer Basics

Objectives

- After completing this chapter, you will be able to:
  - Identify the basic blocks of a digital computer
  - Explain the function of each block of a digital computer
  - Describe what a program is and its relationship to both digital computers and microprocessors

Objectives (cont'd.)

- Identify the basic registers in a microprocessor
- Explain how a microprocessor operates
- Identify the instruction groups associated with microprocessors
- Identify the purpose of microcontrollers
- Describe the function of microcontrollers in everyday life

Computer Basics

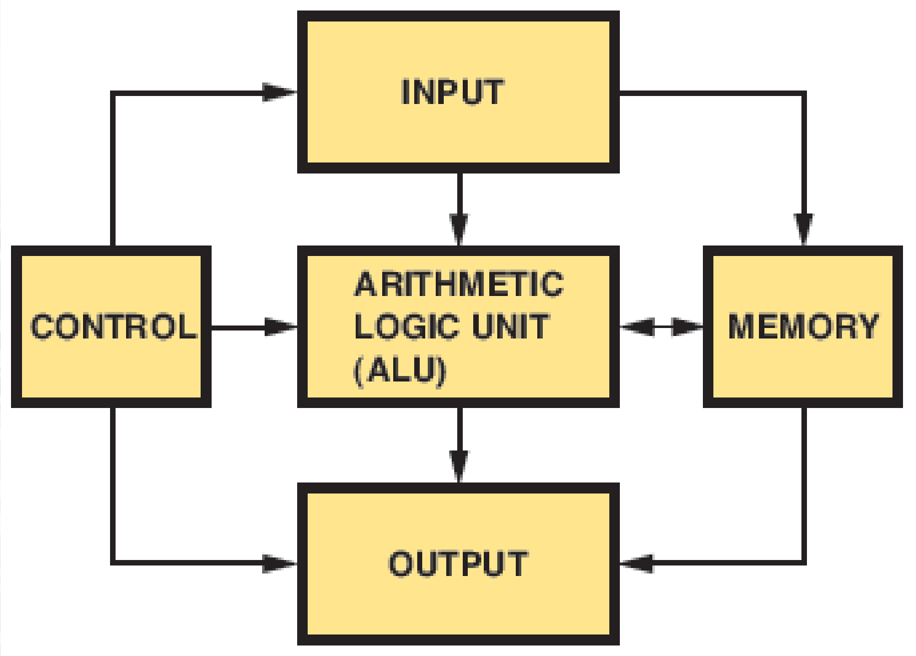


Figure 37-1. Basic blocks of a digital computer.

- Control unit and the arithmetic logic unit are closely related and difficult to separate, which may be collectively called the central processing unit (CPU) or microprocessing unit (MPU).

Computer Basics (cont'd.)

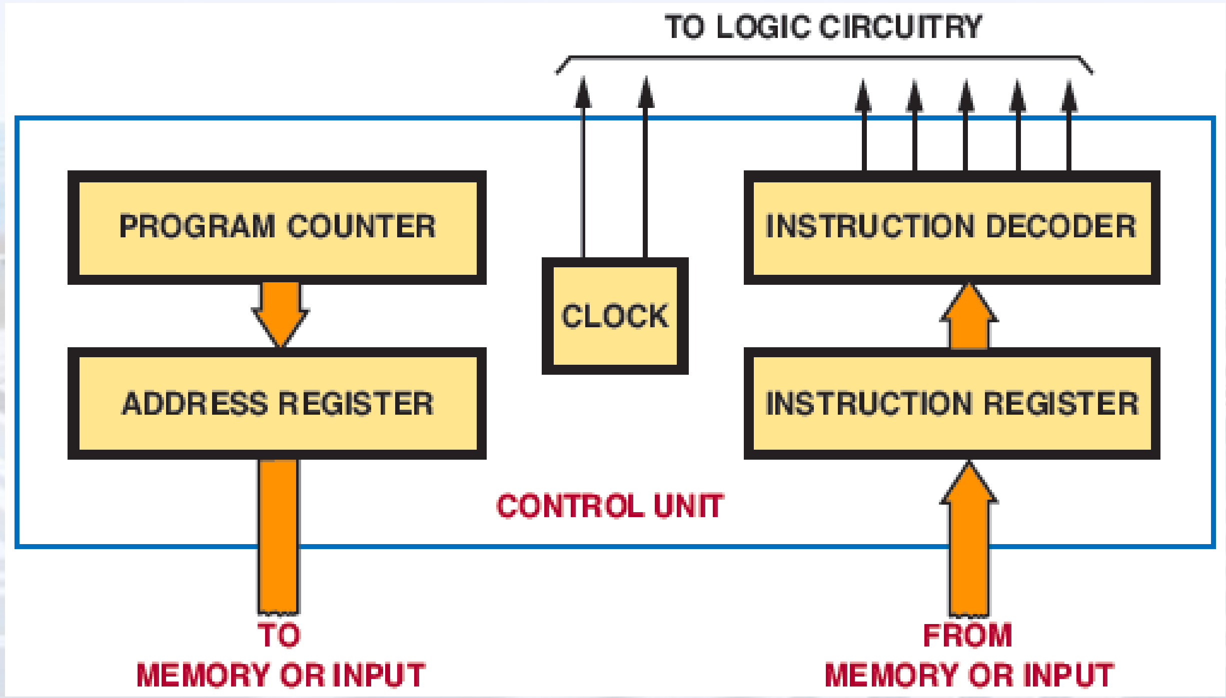
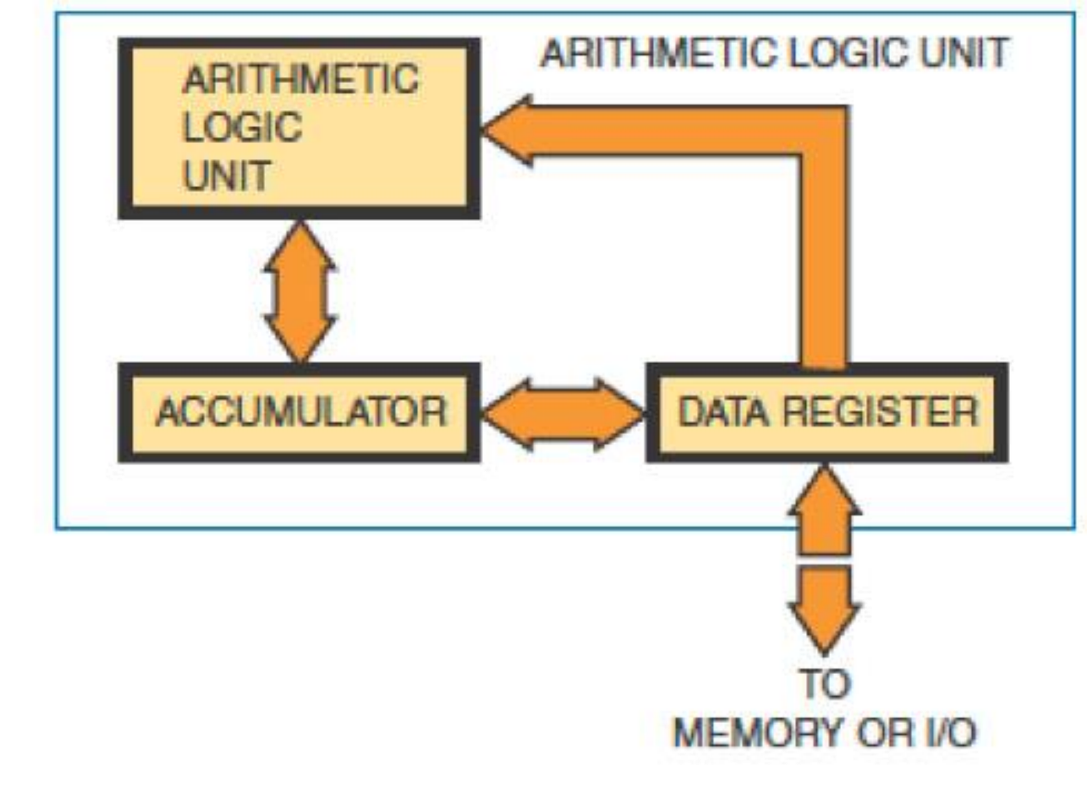


Figure 37-2. Control unit of a computer.

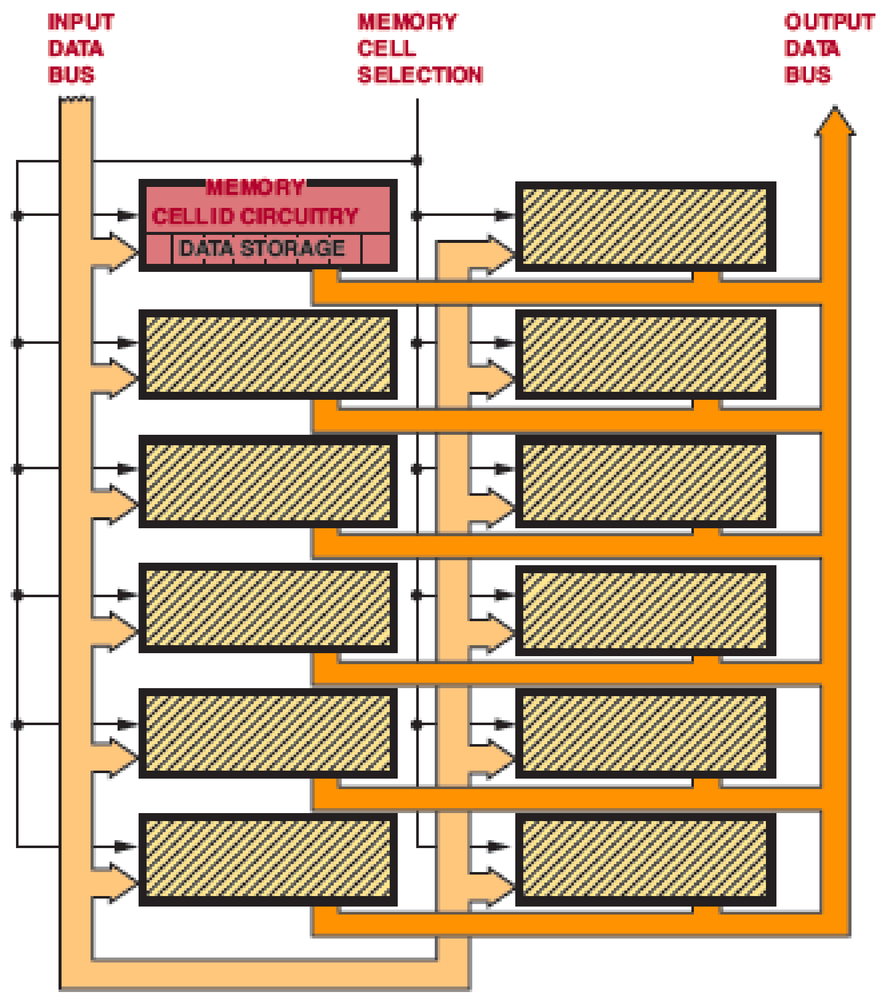
Control unit consists of an address register, an instruction register, an instruction decoder, a program counter, a clock, and circuitry for generating the control pulses

Computer Basics (cont'd.)



- Consists of arithmetic logic circuitry and an accumulator register.
- All data to the accumulator and the ALU are sent via the data register.

Figure 37-3. Arithmetic logic unit (ALU).



- Memory is based on its ability to store (write) or retrieve (read) data, is usually referred to as random access read or write memory (RAM).
- Based on the ability of being able to read only data or instruction from the memory, it is referred to as read-only memory (ROM

Figure 37-4. Memory layout for a computer.

Microprocessor Architecture

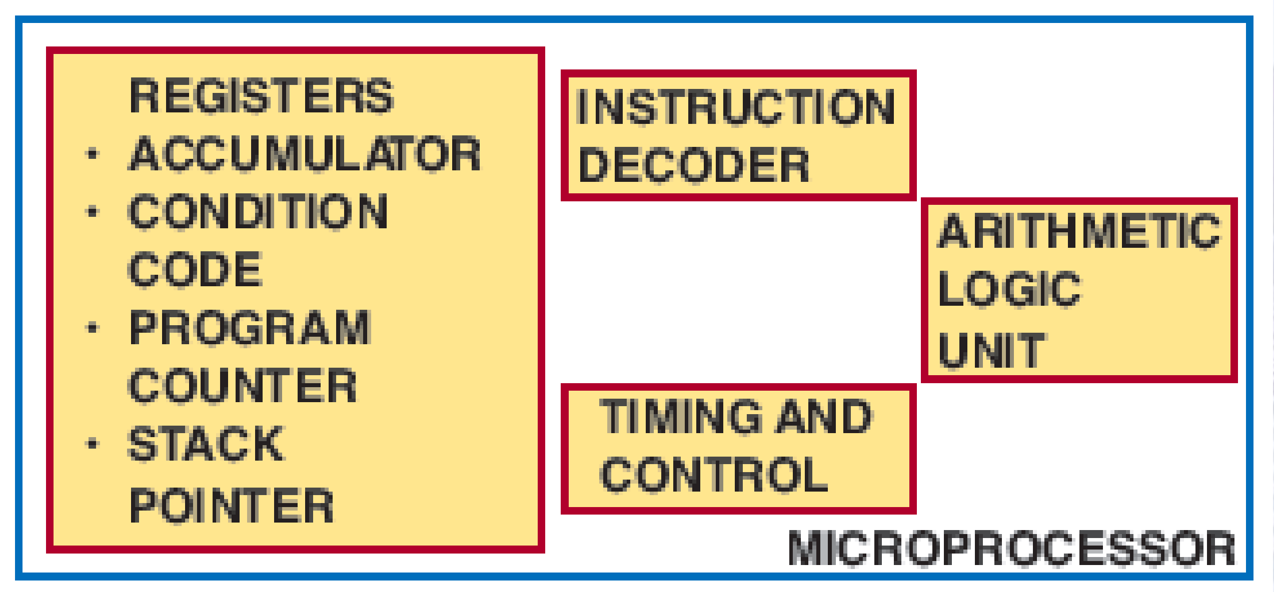


Figure 37-5. Parts of an 8-bit microprocessor.

- Most microprocessors have the same basic set of instructions with different machine codes and a few unique instructions. The basic instructions fall into nine categories:

1. Data movement
2. Arithmetic
3. Logic
4. Compare and test
5. Rotate and shift
6. Program control
7. Stack
8. Input/output
9. Miscellaneous

Microprocessor Architecture (cont'd.)

DESCRIPTION	MNEMONIC	NOTATION	SOURCE	DESTINATION
Load Accumulator	LDA	$M \rightarrow A$	Memory	Accumulator
Load X-Register	LDX	$M \rightarrow X$	Memory	X-Register
Store Accumulator	STA	$A \rightarrow M$	Accumulator	Memory
Store X-Register	STX	$X \rightarrow M$	X-Register	Memory
Transfer Accumulator to X-Register	TAX	$A \rightarrow X$	Accumulator	X-Register
Transfer X-Register to Accumulator	TXA	$X \rightarrow A$	X-Register	Accumulator

Figure 37-6. Data movement instructions.

Microprocessor Architecture (cont'd.)

- Arithmetic instructions
  - Add, subtract, increment, and decrement
  - Allow microprocessor to compute and manipulate data

Microprocessor Architecture (cont'd.)

- Logic instructions
  - Contain at least one Boolean operator
    - AND, OR, and exclusive OR
  - Complement instruction

Microprocessor Architecture (cont'd.)

- Compare instructions
  – Compare data in accumulator with data from a memory location or another register
  – Comparison performed by:
    - Masking
    - Bit-testing

Microprocessor Architecture (cont'd.)

- Rotate and shift instructions
  - Change data in a register or memory
  - Move data to the right or left one bit
- Program-control instructions
  - Change content of the program counter
  - Unconditional or conditional

Microprocessor Architecture (cont'd.)

- Stack instructions
  - Allow storage and retrieval of different microprocessor registers in the stack
- Input/output instructions
  - Deal with controlling I/O devices
- Miscellaneous instructions

Microcontrollers

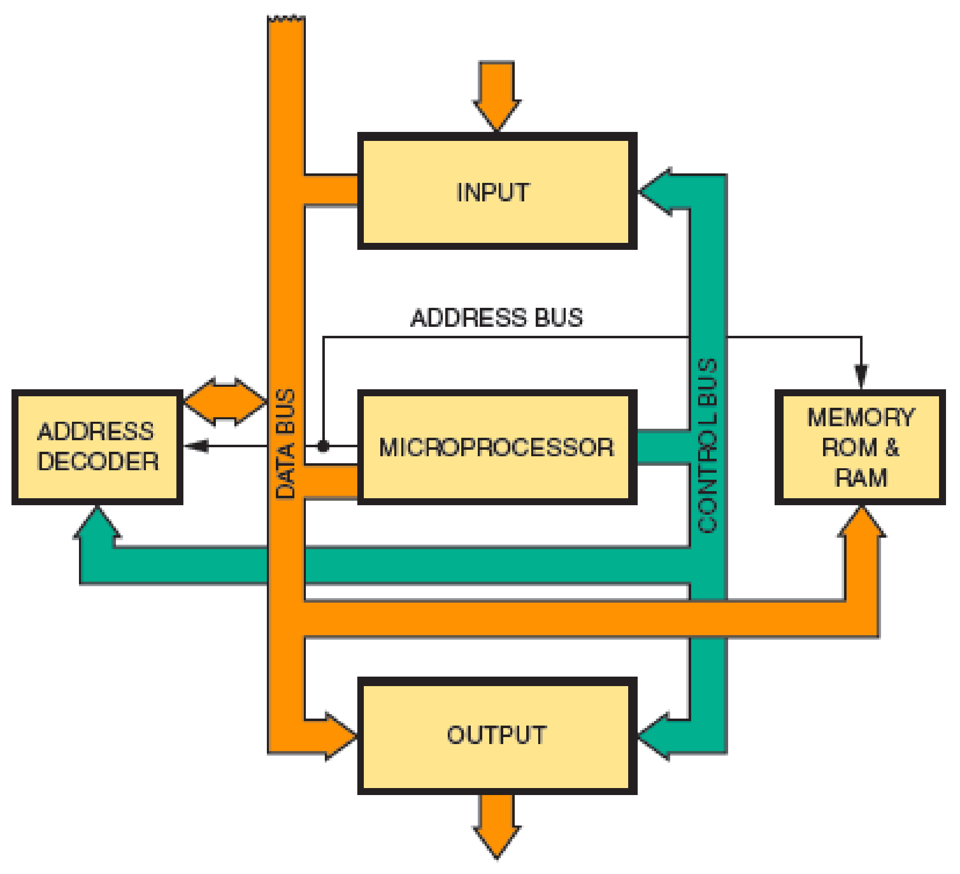
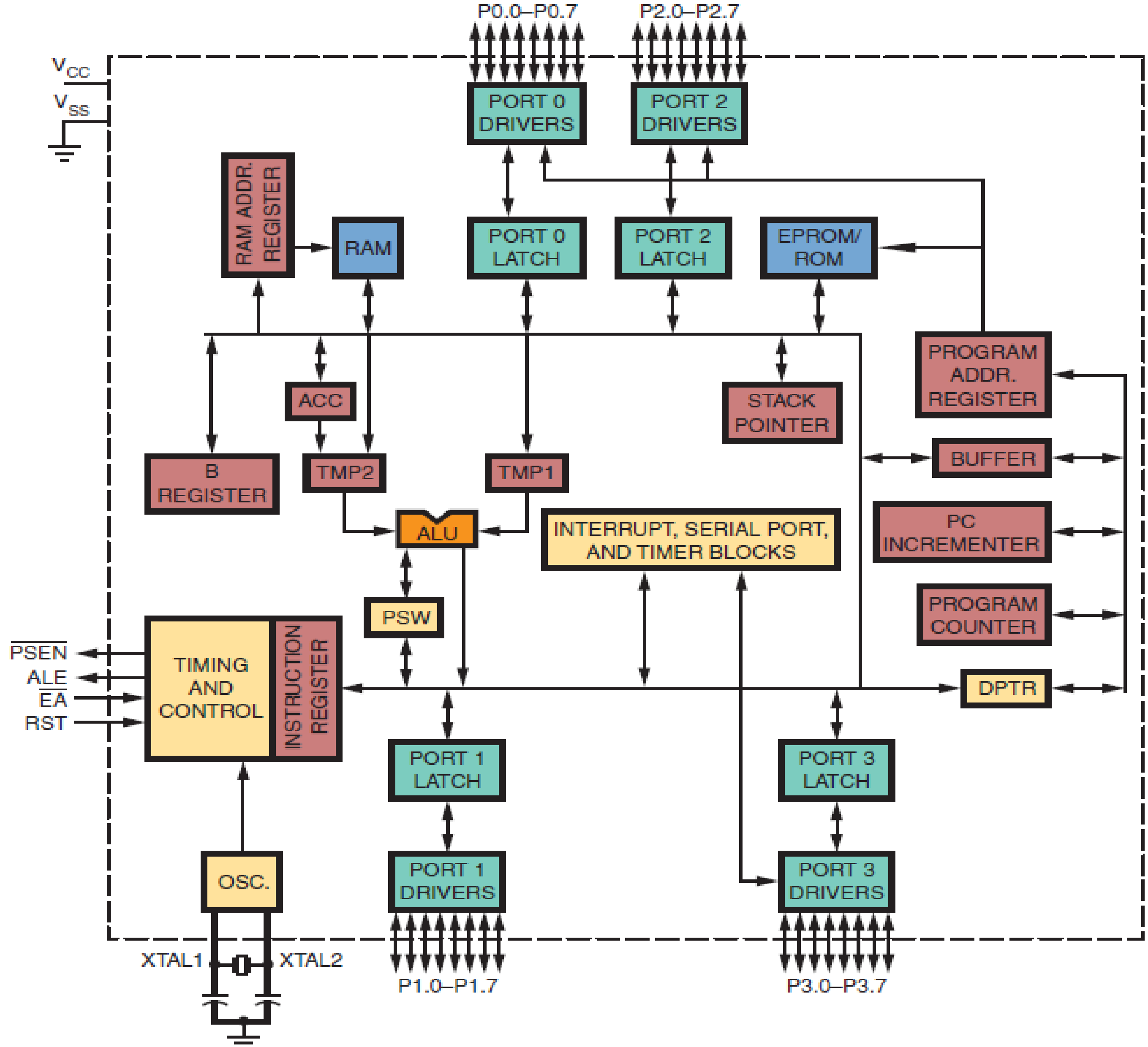


Figure 37-7. Block diagram of a microcontroller.

- Microcontrollers control many appliances (e.g., microwaves, toasters, and stoves), operate high-tech toys and play music in greeting cards.
- Microcontroller is a single-chip computer

FIGURE 37–8

Microcontroller block diagram.



Microcontrollers (cont'd.)

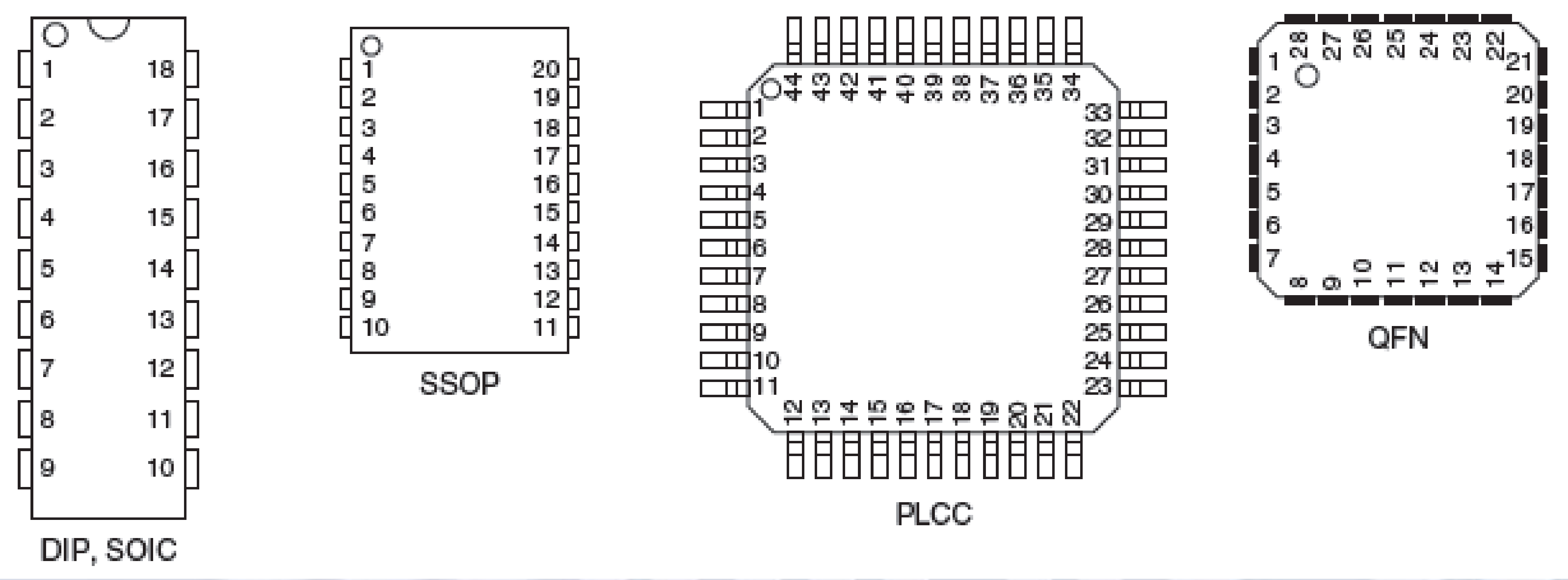


Figure 37-9. Various PIC package outlines.

Summary

- Digital computers consist of a control section, an arithmetic logic unit, memory, and an input/output section
- A program is a set of instructions arranged in a sequential pattern to solve a particular problem
- A microprocessor contains registers, an arithmetic logic unit, timing and control circuitry, and decoding circuitry

Summary (cont'd.)

- Instructions for a microprocessor fall into nine categories
  - Data movement, arithmetic, logic, compare and test, rotate and shift, program control, stack, input/output, miscellaneous
- A microcontroller is a single-chip computer

Chapter 38

Printed Circuit Board Fabrication

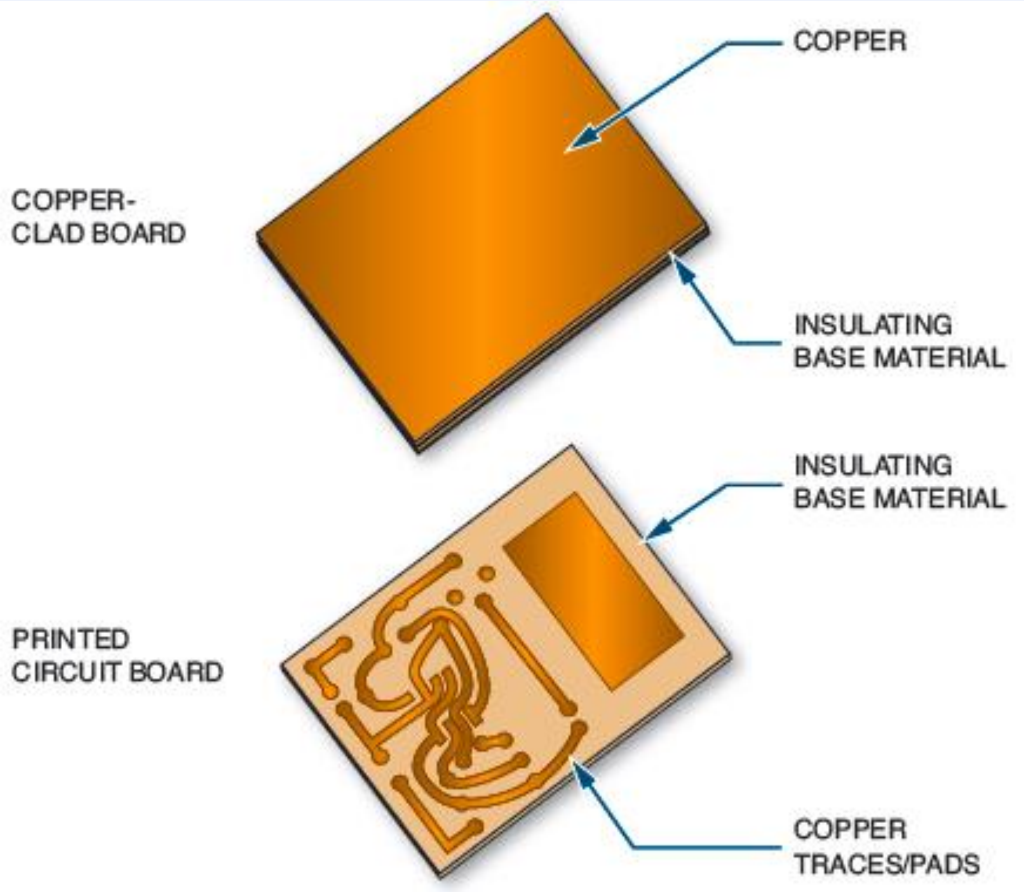
Objectives

- After completing this chapter, you will be able to:
  - Describe the fundamental process for making a printed circuit board
  - Design and lay out a printed circuit board from a schematic diagram
  - Discuss how to transfer a design to a copper-clad board using hand transfer, direct transfer, or screenprinting technique

Objectives (cont'd.)

- Identify techniques to remove the excess copper from a copper-clad board on which a design is formed
- Explain how to drill the appropriate holes in an etched printed circuit board
- Identify the purpose and parts of a material safety data sheet (MSDS)

Fundamentals



A printed circuit board (PCB) consists of an insulating base material that supports the copper traces,

Figure 38-1. Copper-clad board which is made into a printed circuit board.

Fundamentals (cont'd.)

Figure 38-4. Example of an etched printed circuit board.

Fundamentals (cont'd.)

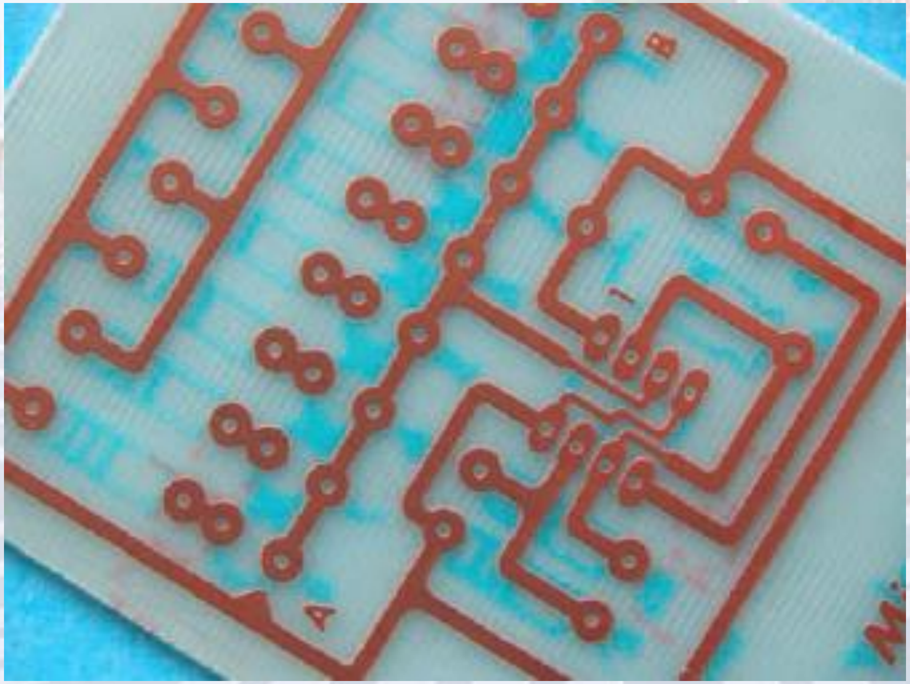


Figure 38-5. Printed circuit board that has been etched and resist removed.

Fundamentals (cont'd.)

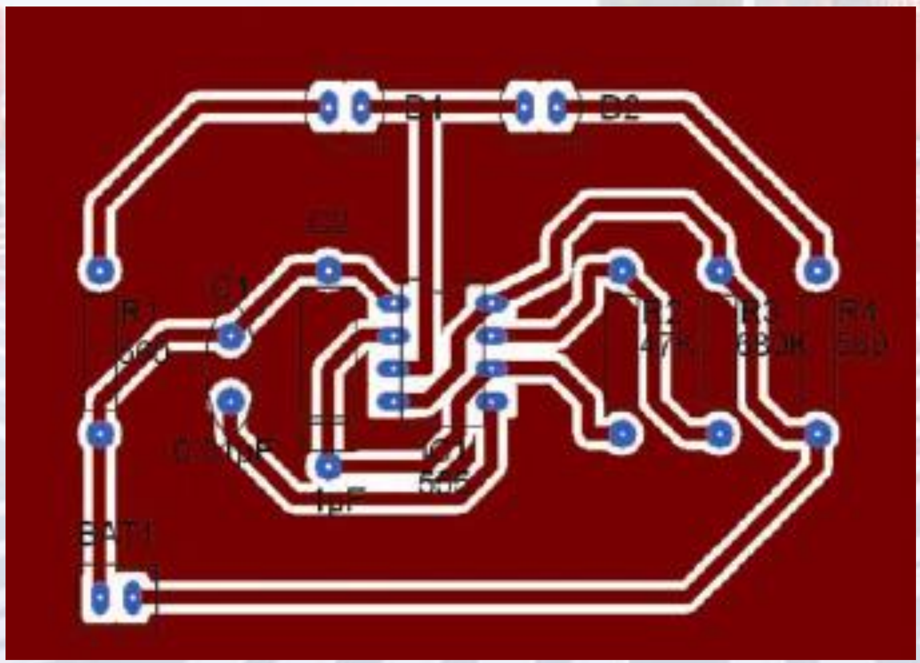


Figure 38-6. X-ray view of a printed circuit board layout from the component side.

Fundamentals

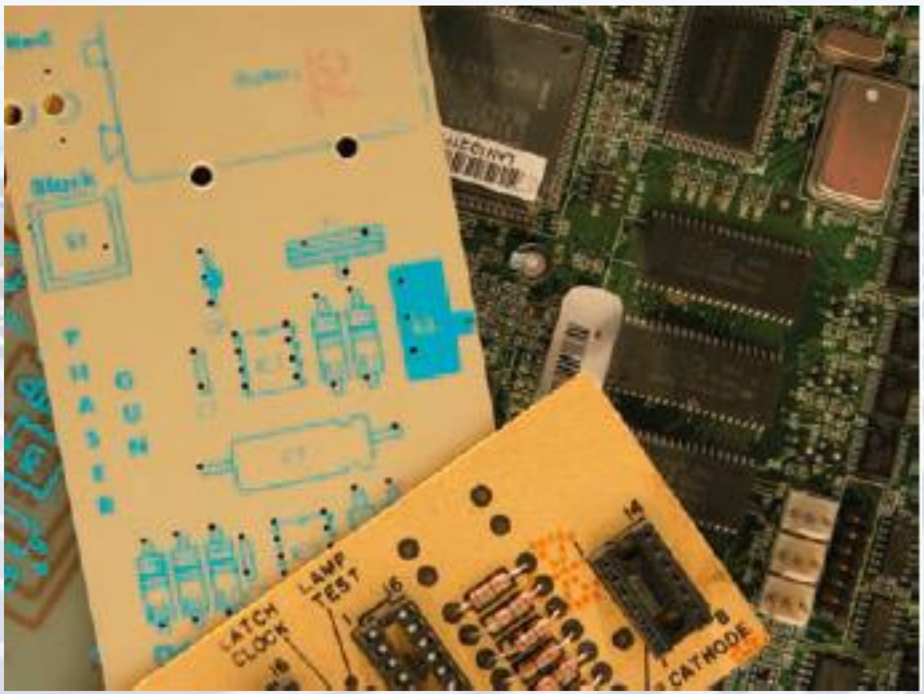


Figure 38-7. Examples of the componen

FIGURE 38–8

Single-sided printed circuit board.

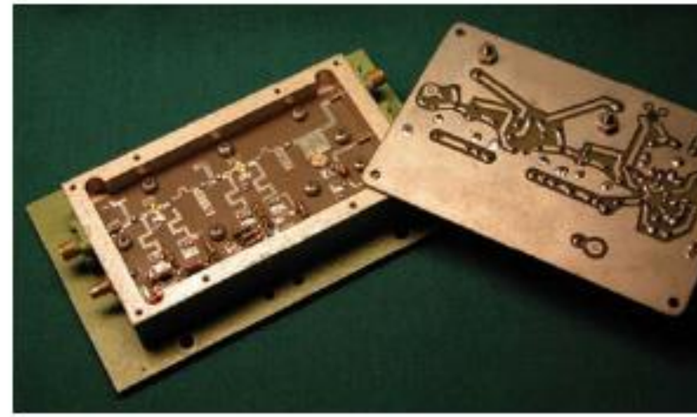
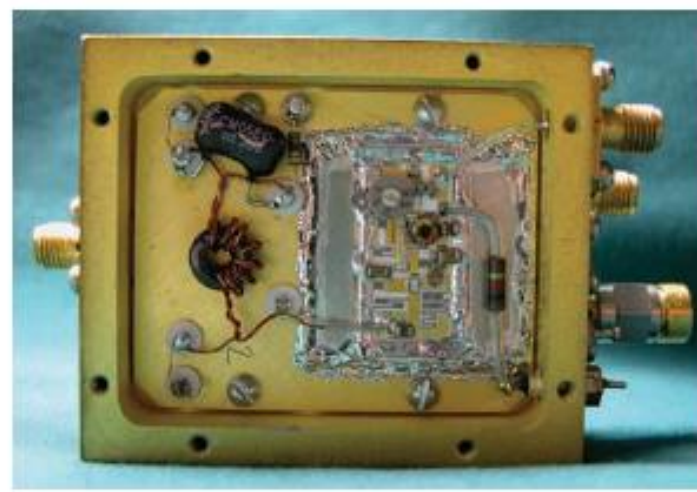


FIGURE 38–9

Ground plane on a printed circuit board.



Fundamentals (cont'd.)

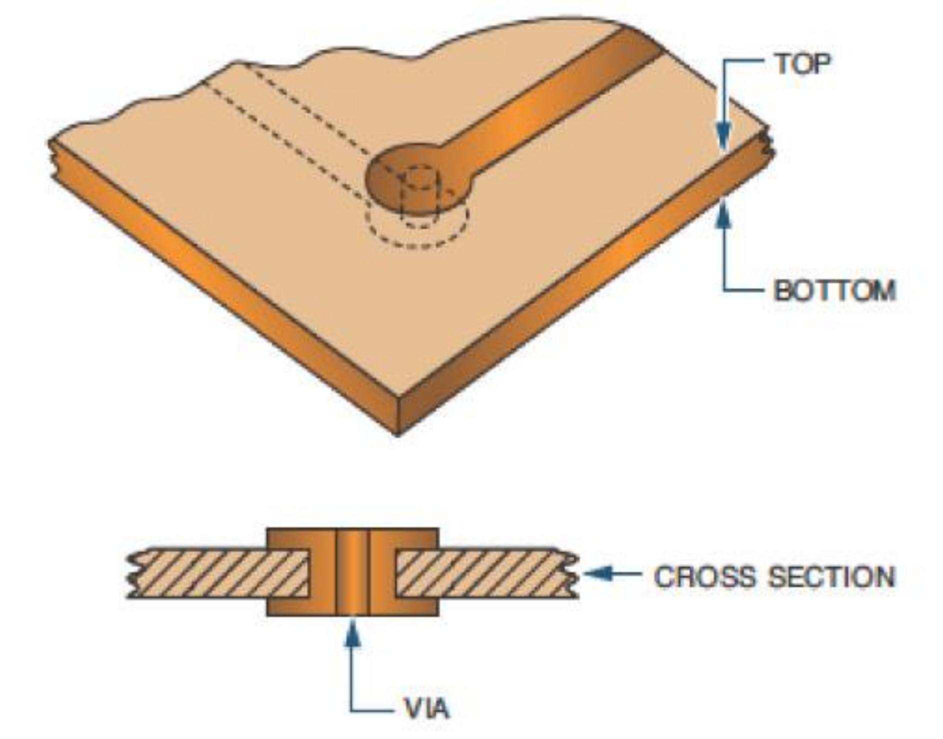


Figure 38-10. Vias connect the top to the bottom trace of a printed circuit board.

Schematic Diagram

Figure 38-11. Block diagram.

Schematic Diagram (cont'd.)

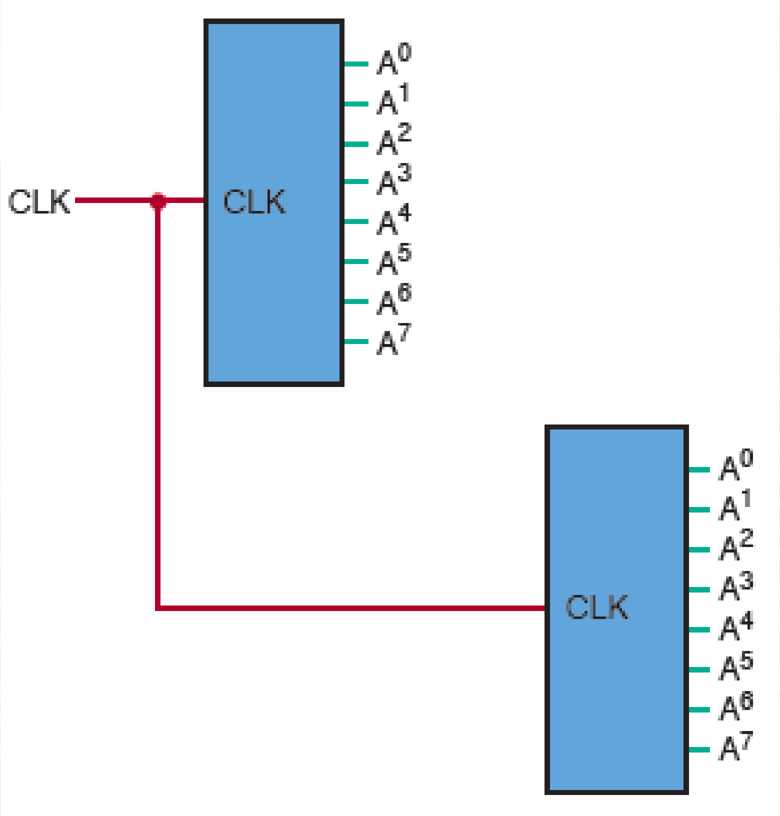


Figure 38-12. Label common signals in digital circuits.

Schematic Diagram (cont'd.)

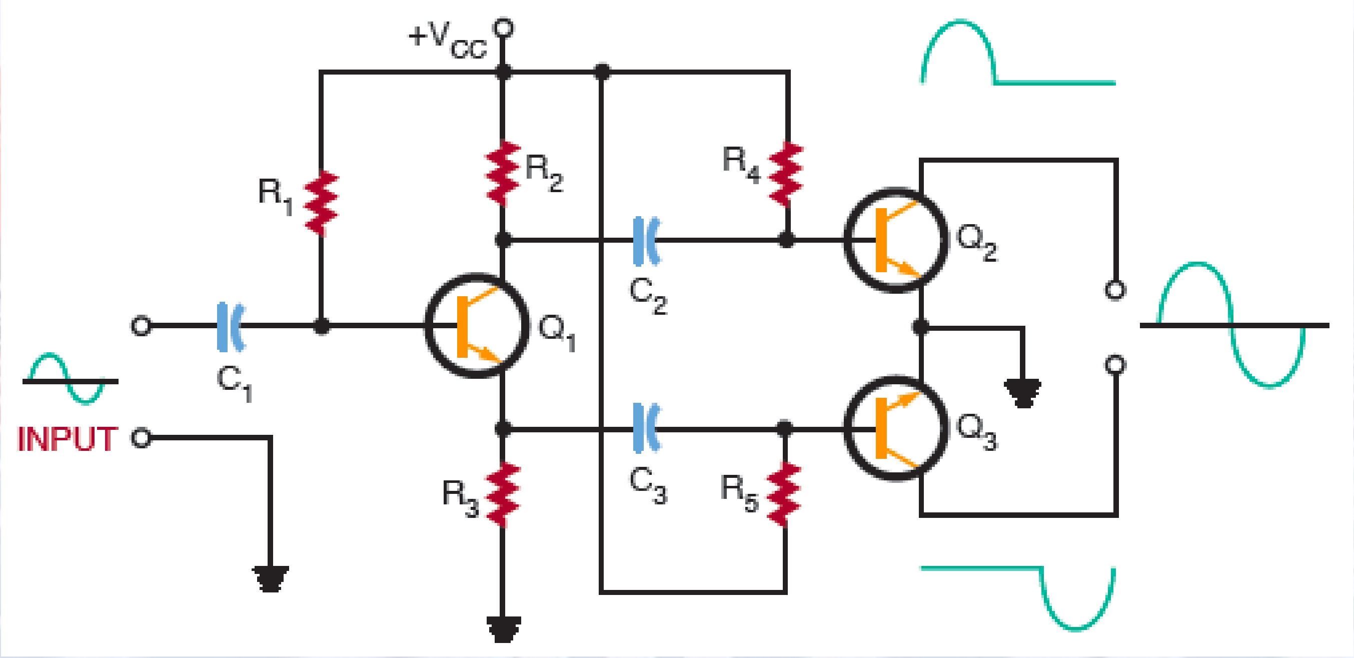


Figure 38-13. Signal flows from left to right, voltage potential has highest potential at top.

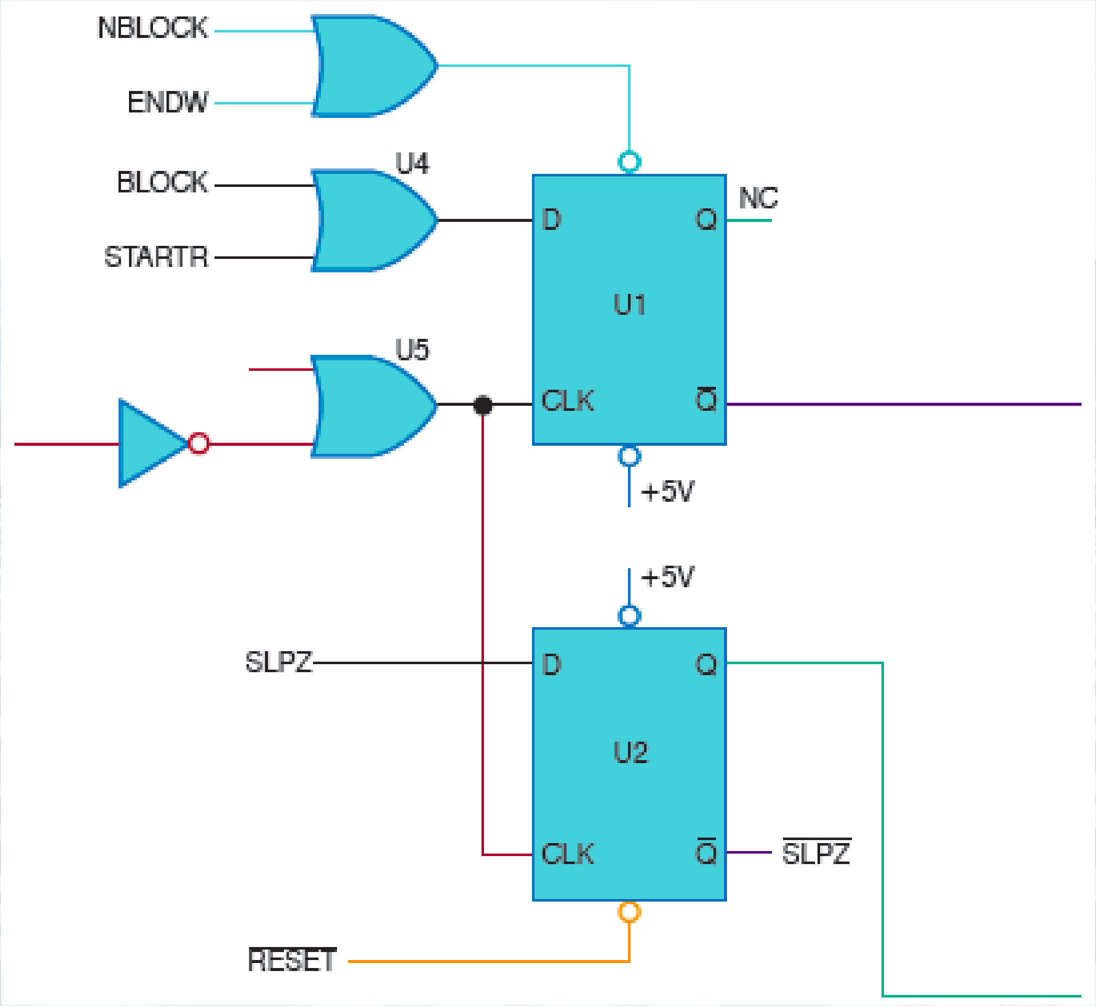


Figure 38-14. Use signal abbreviations rather than draw a maze of lines.

Schematic Diagram (cont'd.)

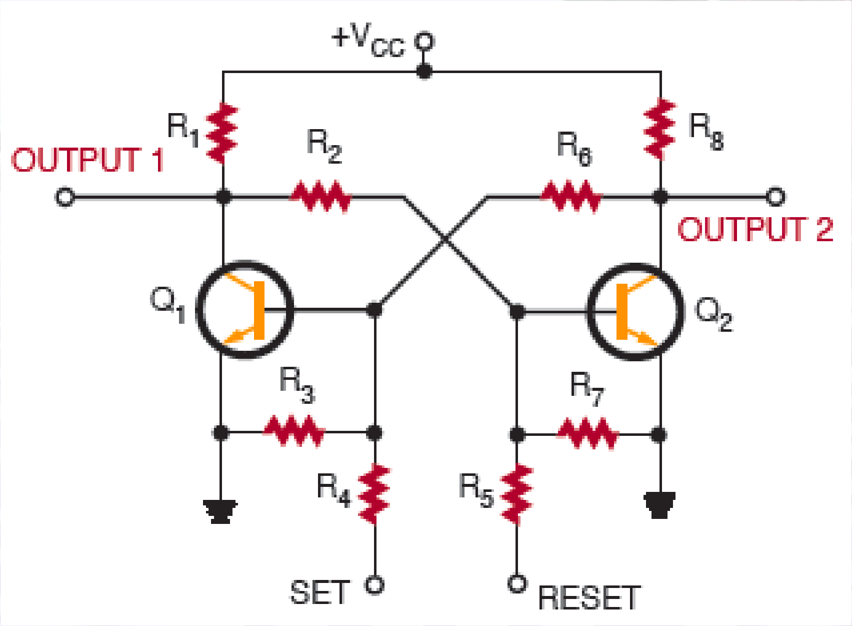


Figure 38-15. Label components starting at the left side and move top to bottom repeating across the schematic.

FIGURE 38–16
Schematic symbols for common electronic components.

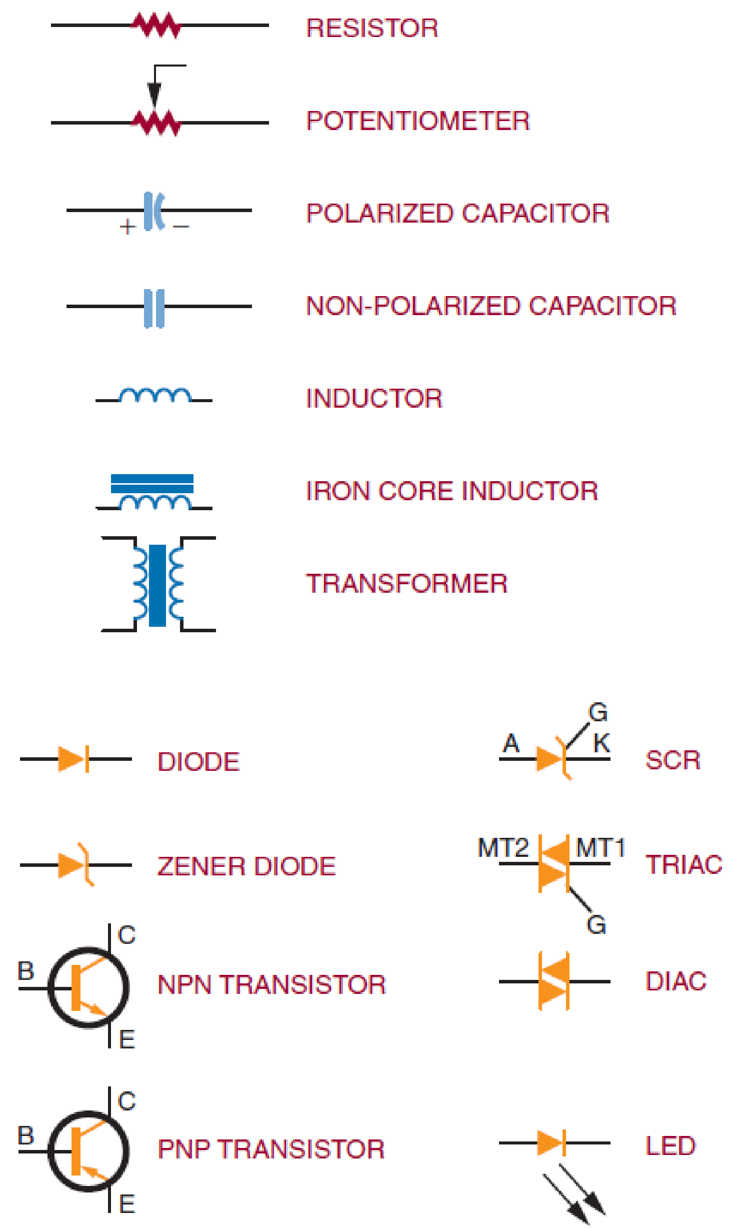
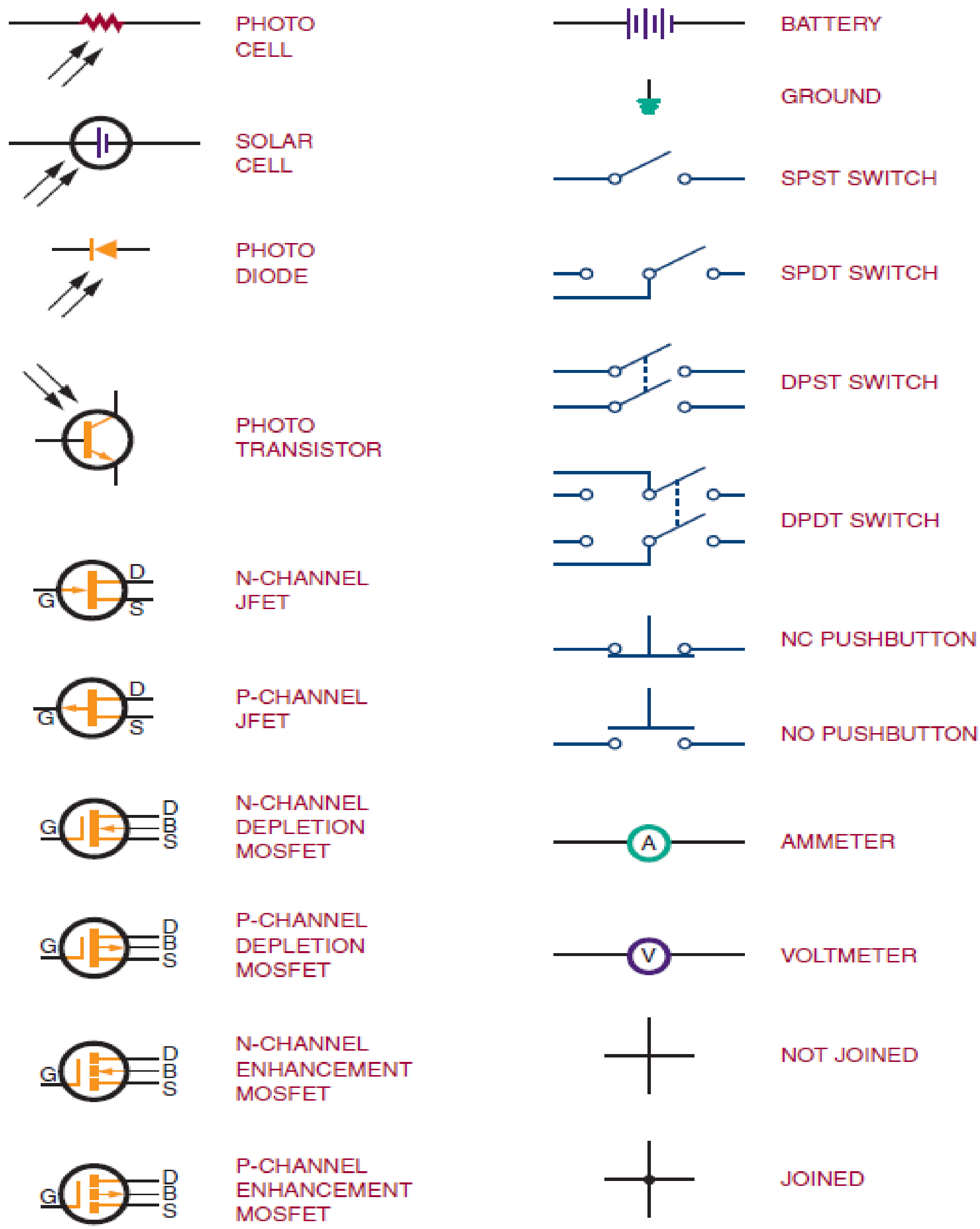


FIGURE 38–16

Schematic symbols for common electronic components. (Continued)



Breadboarding

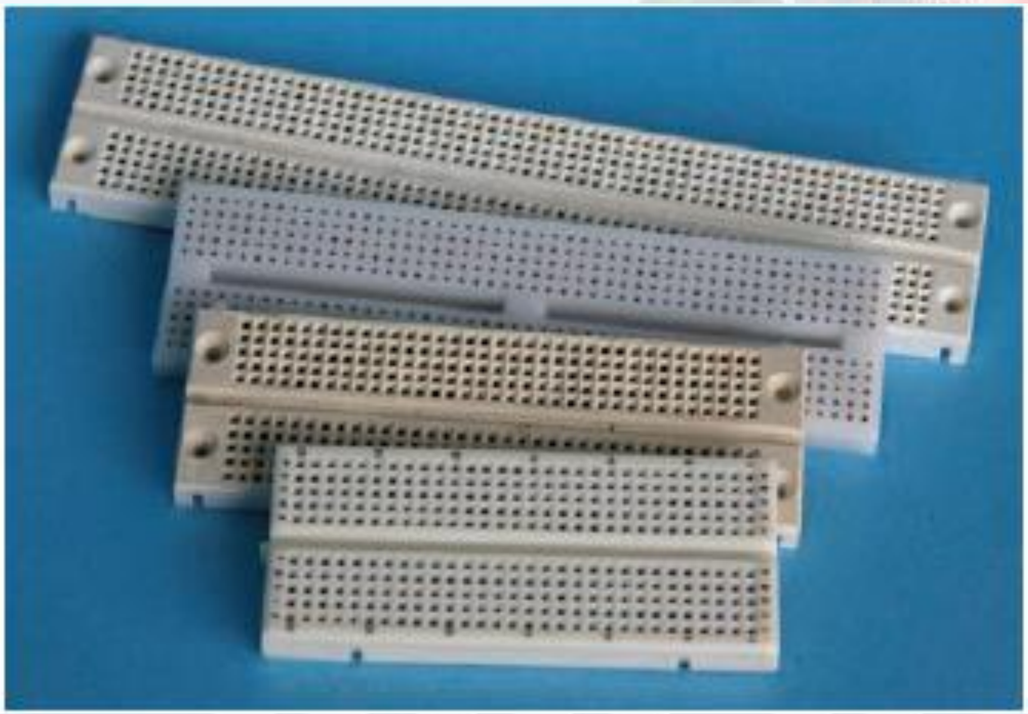


Figure 38-19. Solderless breadboards.

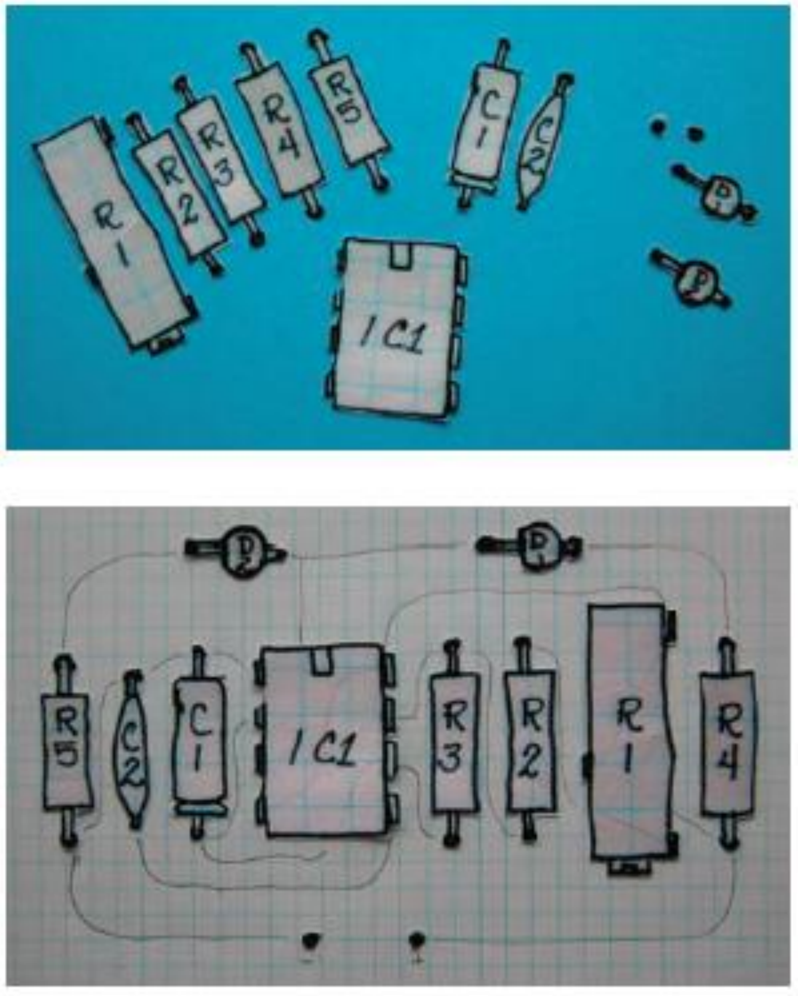


Figure 38-20. Using paper dolls for laying out a printed circuit board design.

Laying Out Printed Circuit Boards (cont'd.)

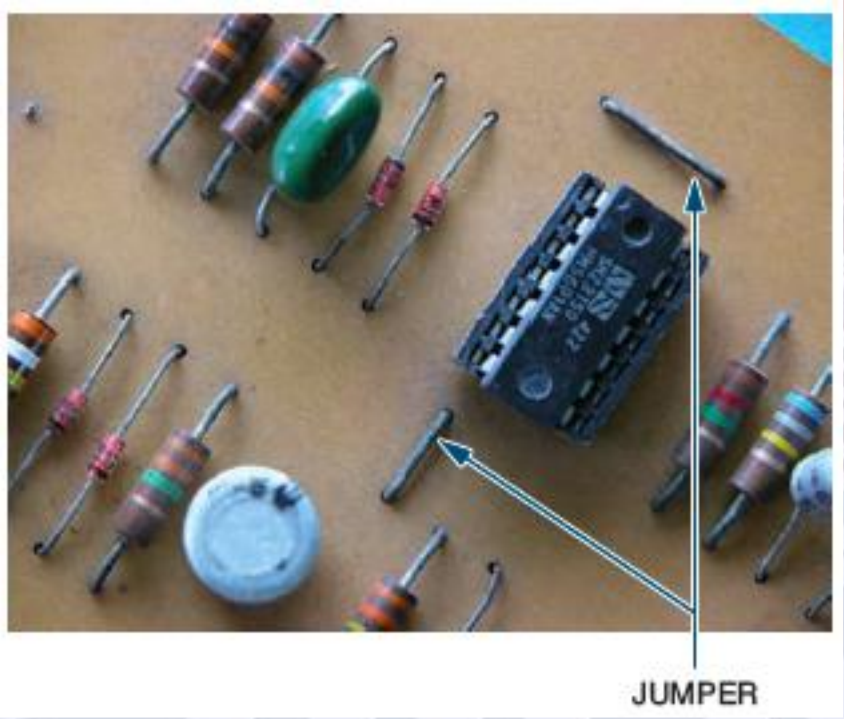


Figure 38-22. Single-sided printed circuit board using jumpers.

Transferring Designs

- Hand-draw the design on the board
  - Resist pen
  - Permanent marker with a very fine tip
- Positive film transparency
- Screenprinting

Etching Printed Circuit Boards

- Techniques for removing excess copper
  - Use a mild acid such as:
    - Ferric chloride
    - Ammonia persulphate
  - Use a CAD type program and a CNC (computer numerical control) machine

Figure 38-28. Examples of commercial spray etchers.

Preparing the Etched Printed Circuit Board

Figure 38-32. High-speed drill for drilling holes in a printed circuit board.

Preparing the Etched Printed Circuit Board (cont'd.)

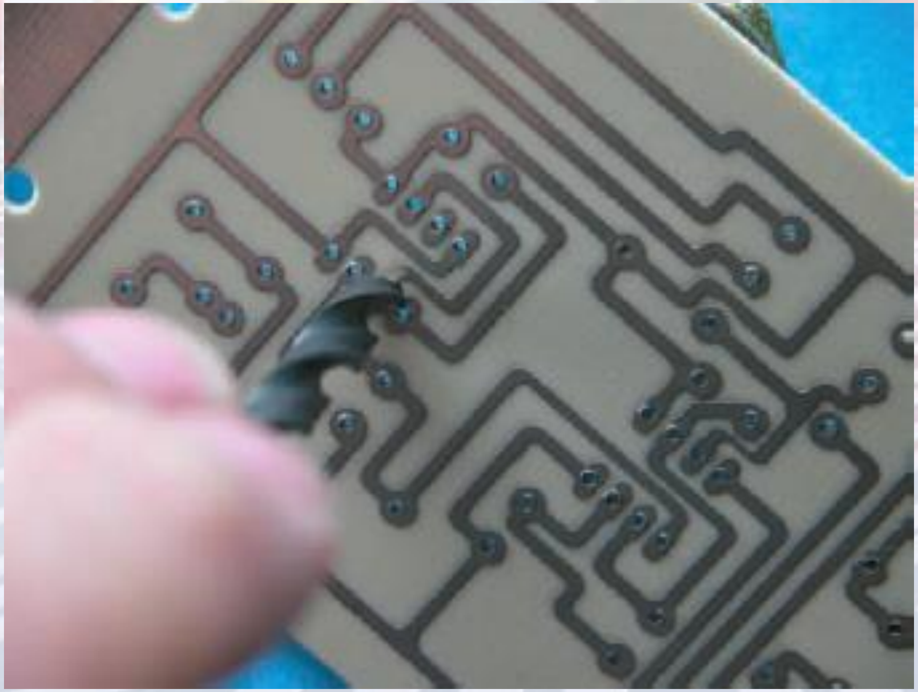


Figure 38-33. Removing burrs from drilled holes.

Material Safety Data Sheet

- MSDSs should be kept on file
  - To help protect from injury and exposure hazards
  - It is required by law

Summary

- A schematic diagram should show the entire circuit in as few drawings as possible
- Transferring the artwork to a copper-clad board
  - Can be done by hand, with transparency film, and through screenprinting

Summary (cont'd.)

- To remove copper from a copper-clad board to form a printed circuit board
  - Use a mild acid or a CAD-type program with a CNC machine
- Use a high-speed drill for drilling all holes in a printed circuit board
- Material Safety Data Sheet (MSDS)
  - Data needed for the safe handling and storage of the hazardous substance

Thank you