Chapter 5 Objectives

- *Selected areas covered in this chapter*:
  - Half adder/ Full adder
  - Analyzing/using decoders & encoders in circuits.
  - Advantages and disadvantages of LEDs.
  - Operation of multiplexers and demultiplexers in circuit applications.
  - Comparing two binary numbers by using the magnitude comparator circuit.
  - Function and operation of code converters.
  - digital circuits using the data bus concept.

- Digital systems obtain data and information continuously operated on in some manner:
  - *Decoding/encoding.*
  - *Multiplexing/demultiplexing,.*
  - *Comparison; Code conversion; Data busing.*
- These and other operations have been facilitated by the availability of numerous ICs in the MSI (medium-scale-integration) category.

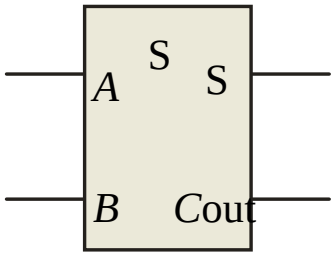
Half-Adder

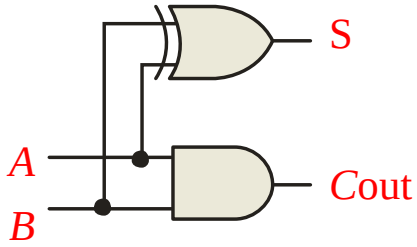
Basic rules of binary addition are performed by a **half adder**, which has two binary inputs (*A* and *B*) and two binary outputs (Carry out and Sum).

The inputs and outputs can be summarized on a truth table.

Inputs		Outputs	
A	*B*	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The logic symbol and equivalent circuit are:



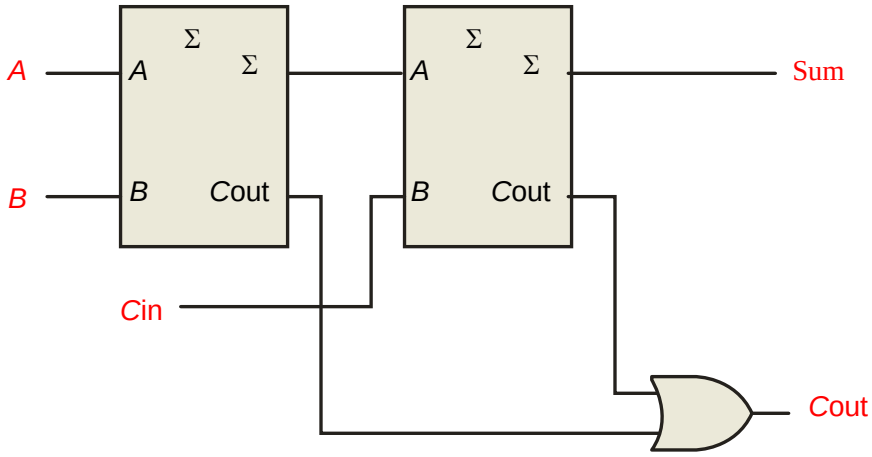


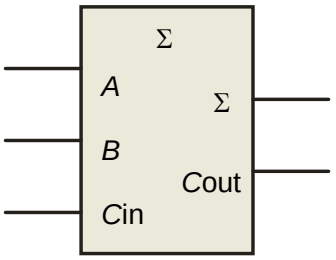
Full-Adder

By contrast, a **full adder** has three binary inputs (*A*, *B*, and Carry in) and two binary outputs (Carry out and Sum). The truth table summarizes the operation.

Inputs			Outputs	
A	*B*	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

A full-adder can be constructed from two half adders as shown:

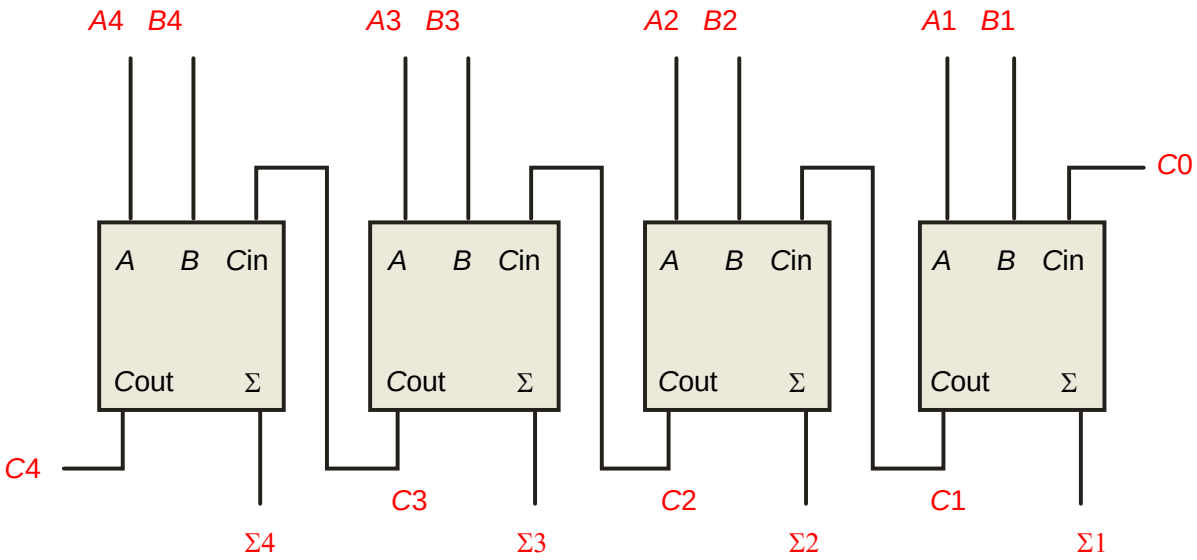




Symbol

Parallel Adders

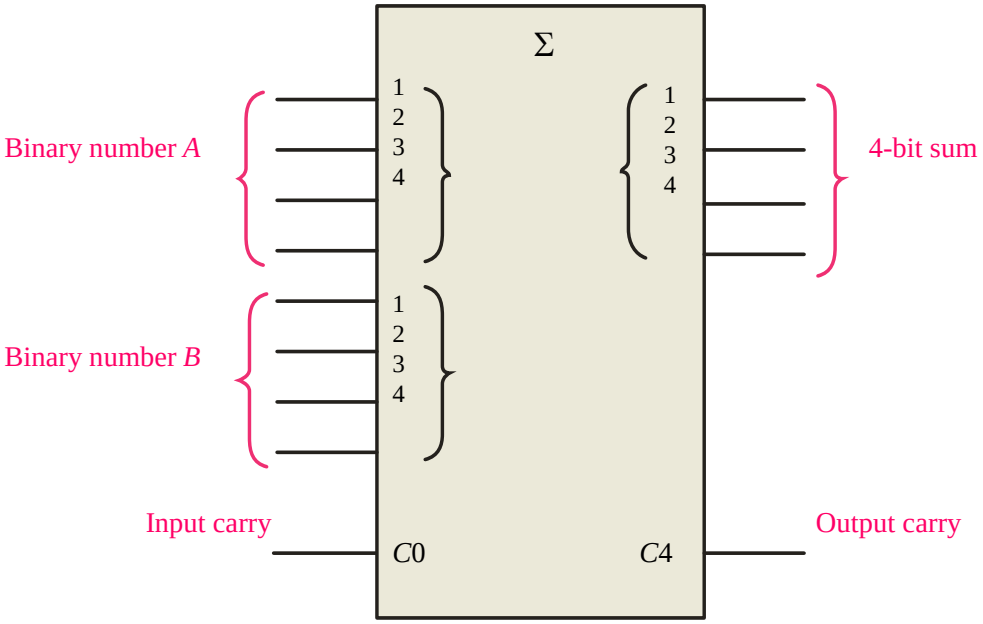
Full adders are combined into parallel adders that can add binary numbers with multiple bits. A 4-bit adder is shown.



The output carry (*C*4) is not ready until it propagates through all of the full adders. This is called *ripple carry*, delaying the addition process.

Parallel Adders

The logic symbol for a 4-bit parallel adder is shown. This 4-bit adder includes a carry in (labeled (*C*0) and a Carry out (labeled *C*4).



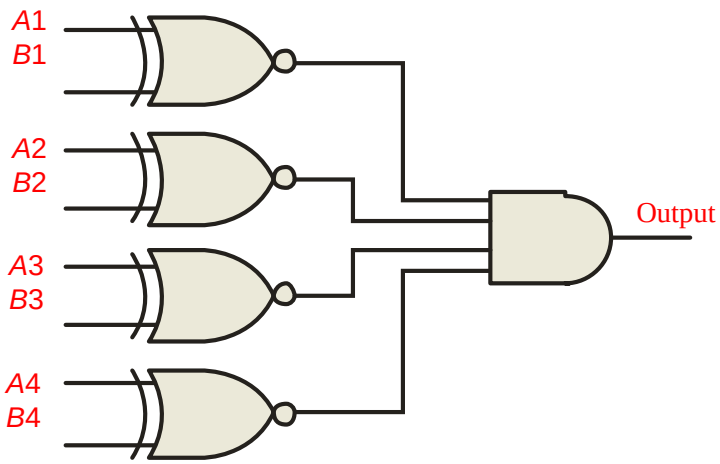
The 74LS283 is an example. It features *look-ahead carry*, which adds logic to minimize the output carry delay. For the 74LS283, the maximum delay to the output carry is 17 ns.

Comparators

The function of a comparator is to compare the magnitudes of two binary numbers to determine the relationship between them. In the simplest form, a comparator can test for equality using XNOR gates.

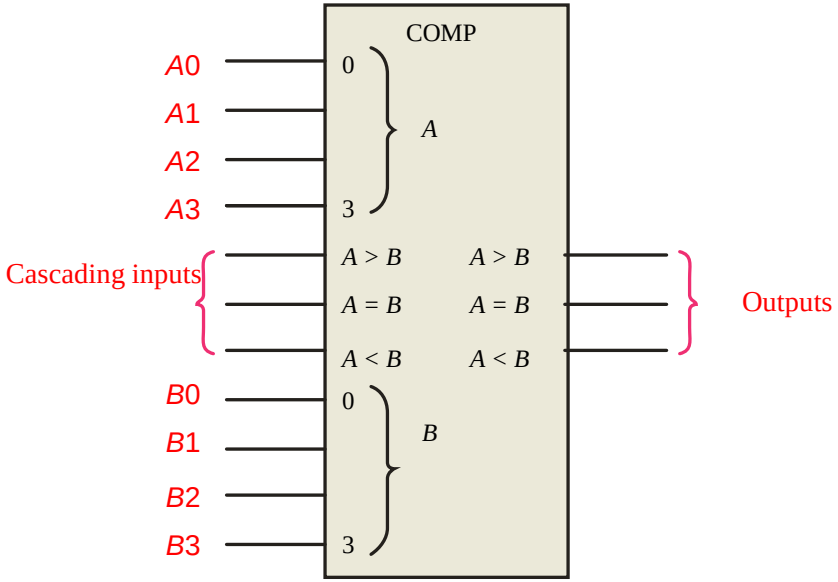
How could you test two 4-bit numbers for equality?

AND the outputs of four XNOR gates.



Comparators

IC comparators provide outputs to indicate which of the numbers is larger or if they are equal. The bits are numbered starting at 0, rather than 1 as in the case of adders. Cascading inputs are provided to expand the comparator to larger numbers.



The IC shown is the 4-bit 74LS85.

Magnitude Comparator

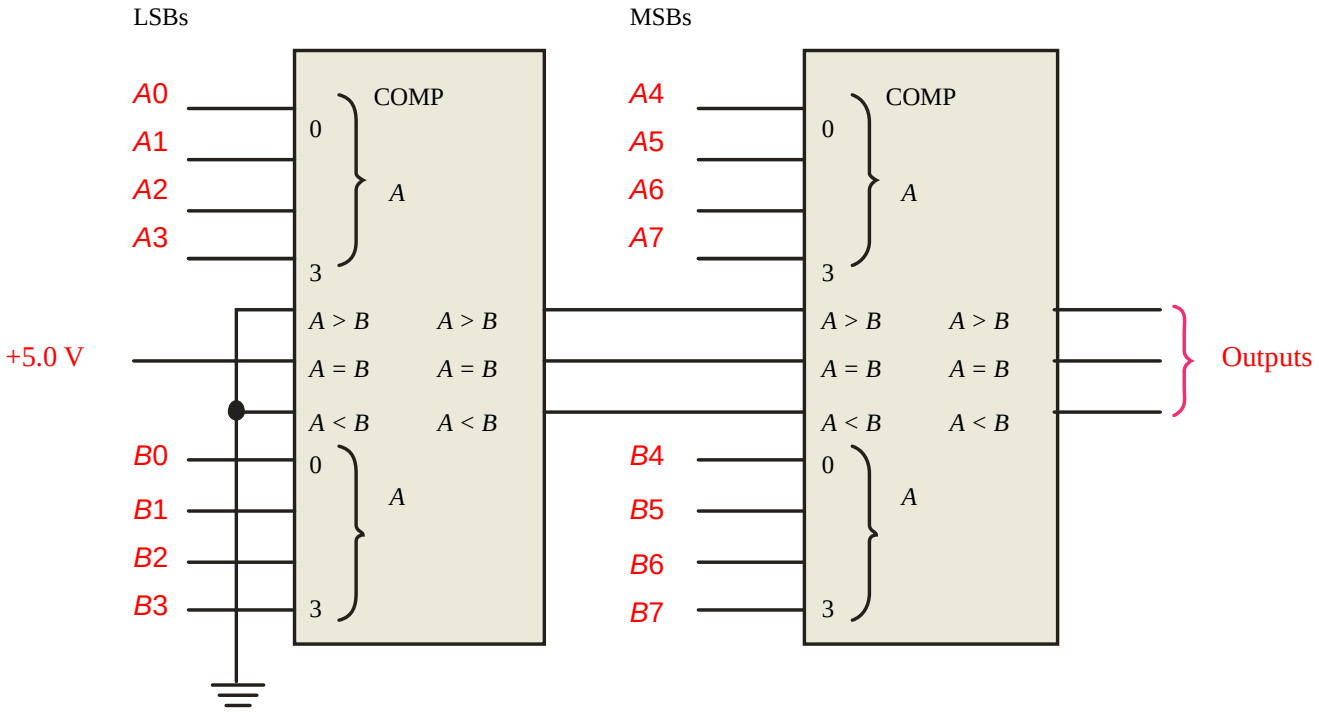
TRUTH TABLE

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

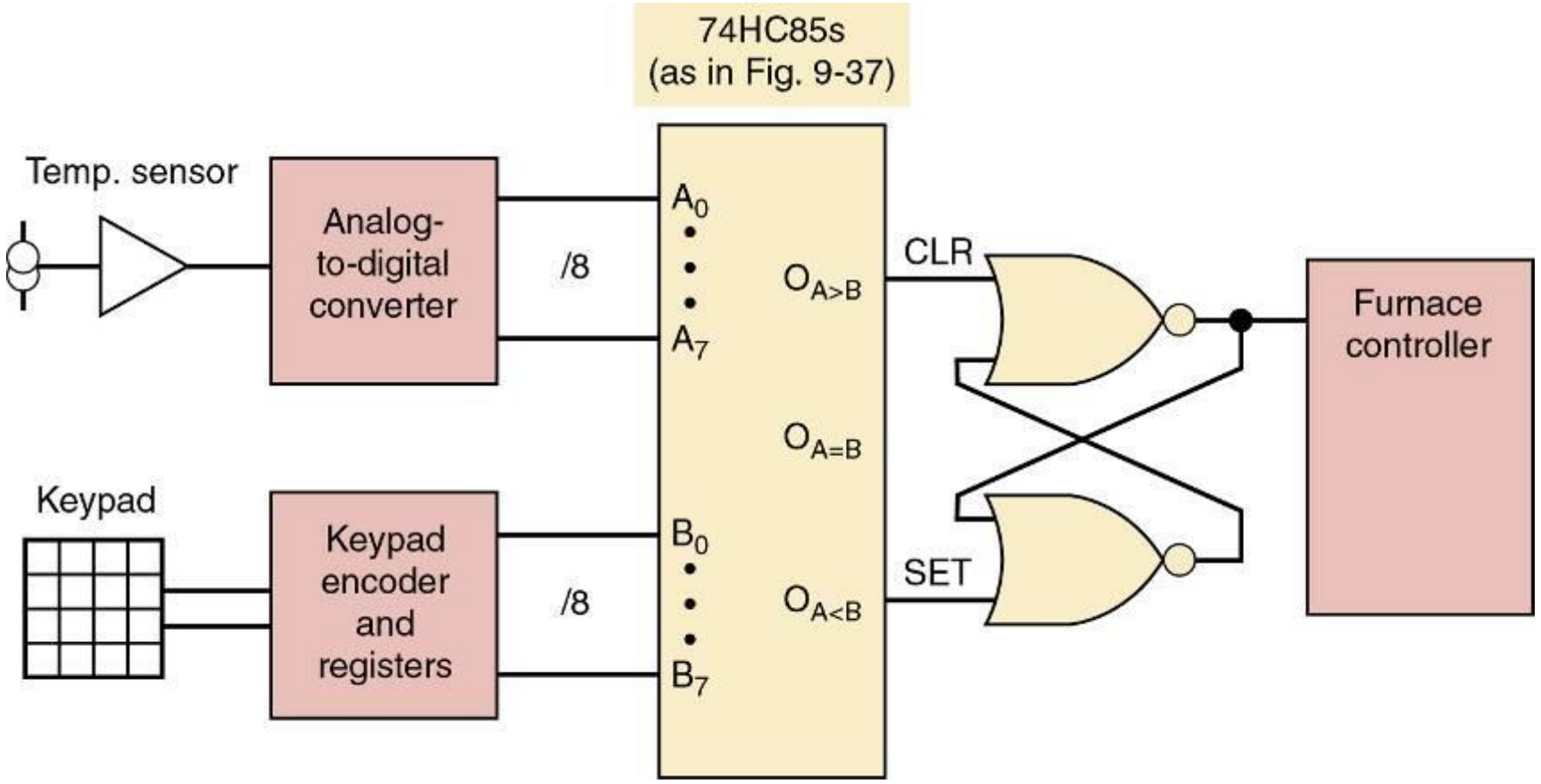
COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3>B_3$	X	X	X	X	X	X	H	L	L
$A_3<B_3$	X	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2>B_2$	X	X	X	X	X	H	L	L
$A_3=B_3$	$A_2<B_2$	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	L	L	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	H	L	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	X	X	H	L	L	H
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	L	H	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	H	L	L	L	L

Comparators

IC comparators can be expanded using the cascading inputs as shown. The lowest order comparator has a HIGH on the $A = B$ input.



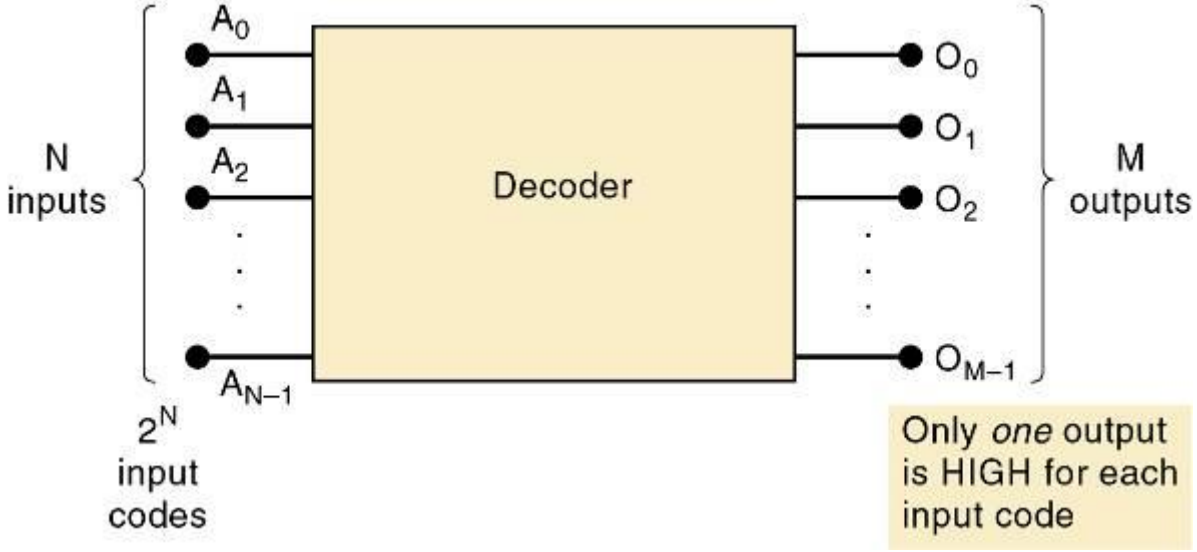
Magnitude comparator used in a digital thermostat.



5-1 Decoders

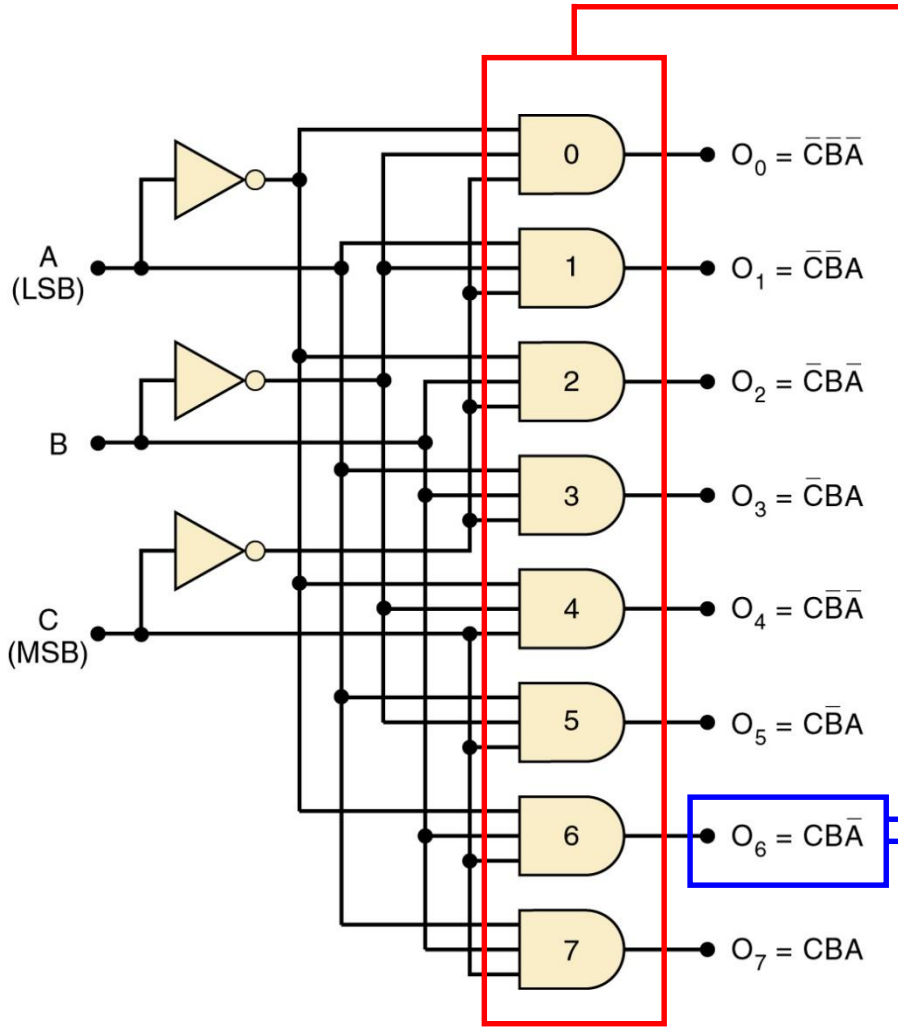
- Decoders are used when an output or a group of outputs is to be activated only on the occurrence of a specific combination of input levels.
- A decoder accepts a set of inputs that represents a binary number - activating only the output that corresponds to the input number.

For each of these input combinations, only one of the *M* outputs will be active (HIGH); all the other outputs are LOW.



Many decoders are designed to produce active-LOW outputs, where only the selected output is LOW while all others are HIGH.

Circuitry for a decoder with three inputs and 8 outputs.

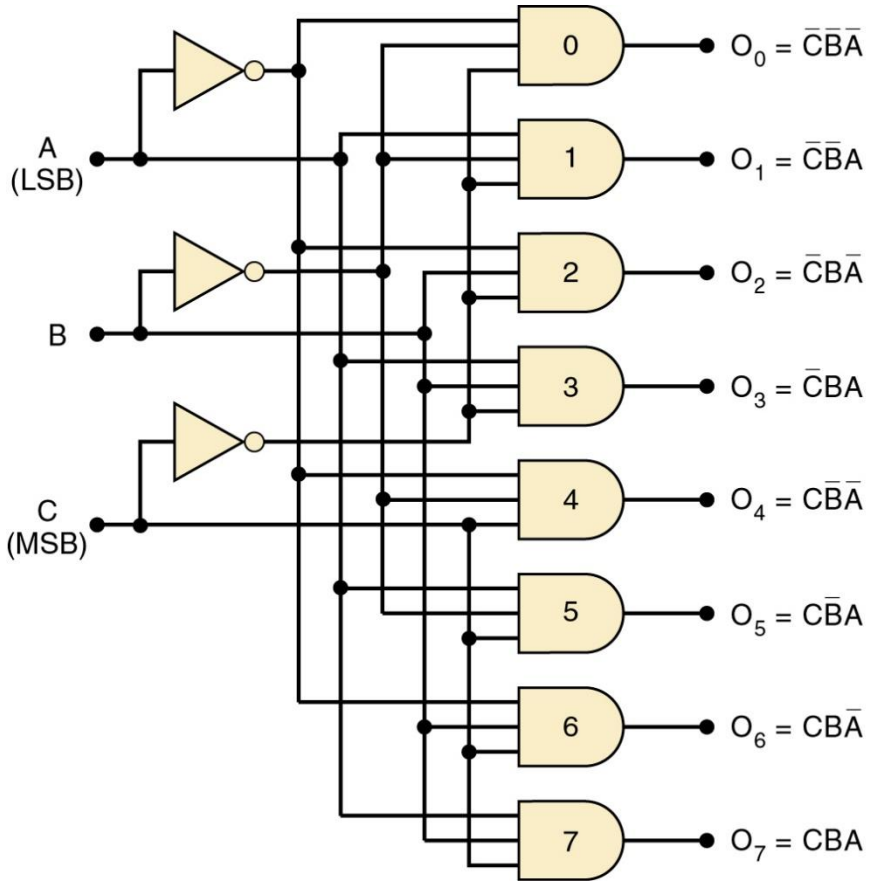


It uses all **AND** gates, so outputs are active-HIGH

C	B	A	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Output O_6 goes HIGH only when *CBA* $110_2 = 6_{10}$.

Circuitry for a decoder with three inputs and 8 outputs.



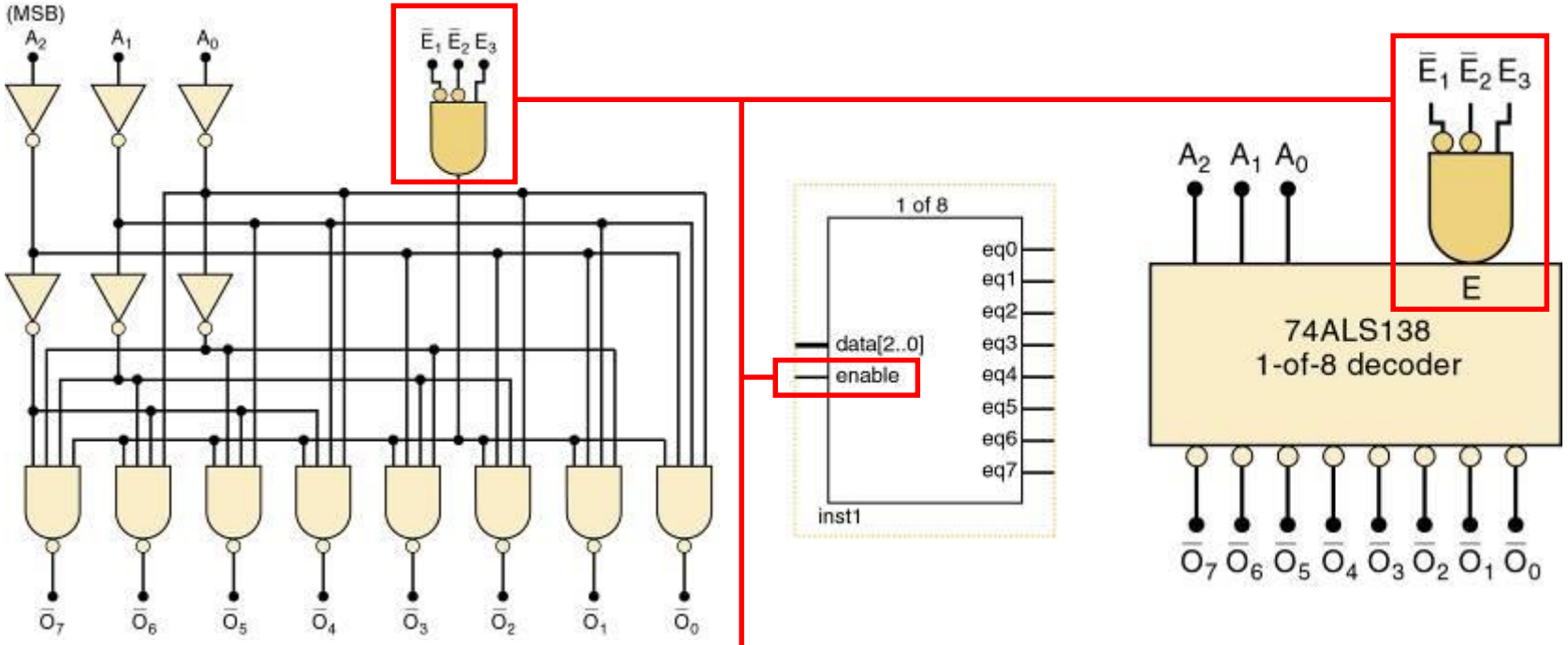
This can be called a *3-line-to-8-line decoder*—it has three input lines and eight output lines.

Also called a *binary-to-octal decoder* or *converter*—taking three-bit binary input code and activating one of eight (octal) outputs.

Also referred to as a *1-of-8 decoder*—only 1 of the 8 outputs is activated at one time.

- Some decoders have one or more enable inputs used to control the operation of the decoder.
  - The decoder is enabled only if *ENABLE* is HIGH.
- With common *ENABLE* line connected to a fourth input of each gate:
  - If *ENABLE* is HIGH, the decoder functions normally.
    - *A, B, C* input will determine which output is HIGH.
  - If *ENABLE* is LOW, *all* outputs will be forced LOW.
    - *Regardless* of the levels at the *A, B, C* inputs.

The 74ALS138 decoder.



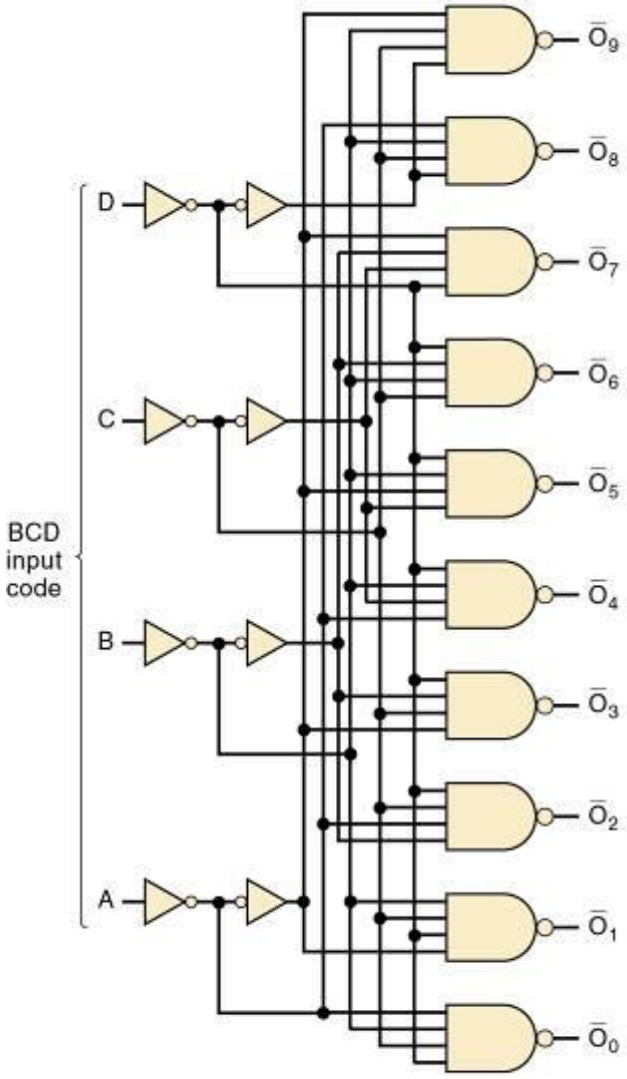
ENABLE inputs

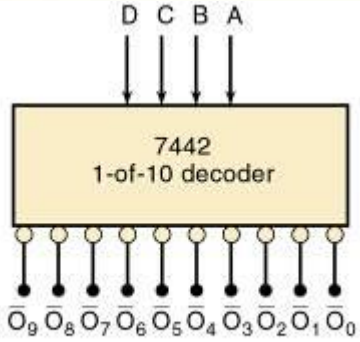
$\bar{E}_1$	$\bar{E}_2$	E_3	Outputs
0	0	1	Respond to input code $A_2A_1A_0$
1	X	X	Disabled – all HIGH
X	1	X	Disabled – all HIGH
X	X	0	Disabled – all HIGH

5-1 Decoders

7442
BCD-to-decimal decoder.

This decoder does *not* have an enable input.

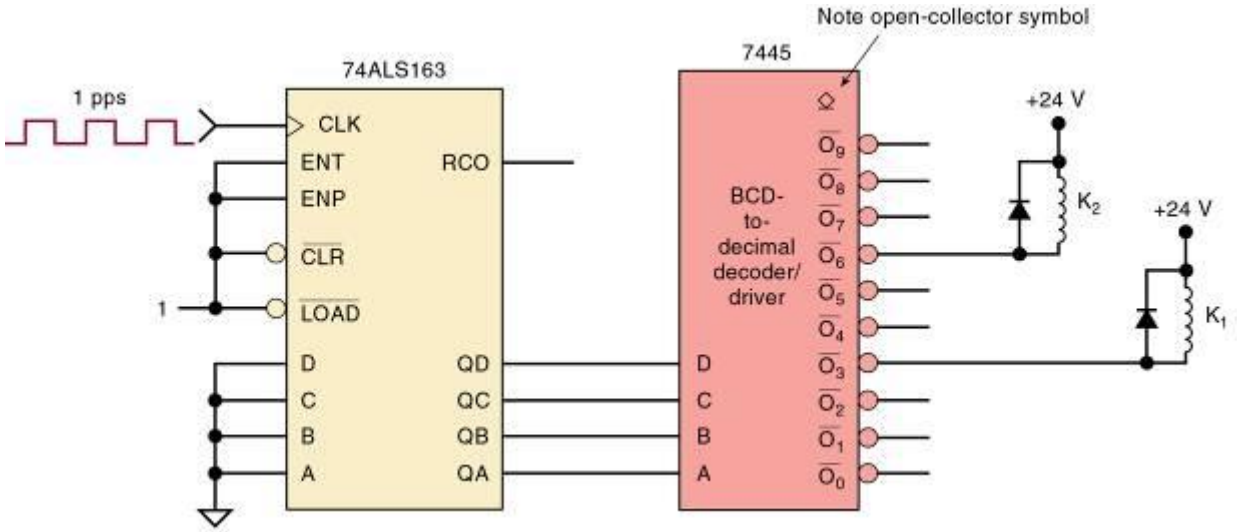


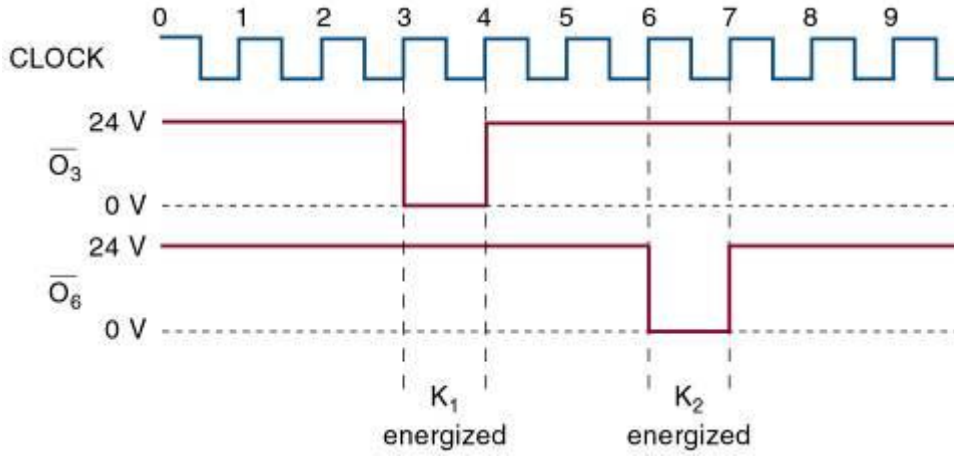


Inputs				
D	C	B	A	Active Output
L	L	L	L	$\overline{O}_0$
L	L	L	H	$\overline{O}_1$
L	L	H	L	$\overline{O}_2$
L	L	H	H	$\overline{O}_3$
L	H	L	L	$\overline{O}_4$
L	H	L	H	$\overline{O}_5$
L	H	H	L	$\overline{O}_6$
L	H	H	H	$\overline{O}_7$
H	L	L	L	$\overline{O}_8$
H	L	L	H	$\overline{O}_9$
H	L	H	L	None
H	L	H	H	None
H	H	L	L	None
H	H	L	H	None
H	H	H	L	None
H	H	H	H	None

H = HIGH Voltage Level
L = LOW Voltage Level

7445 BCD-to-decimal decoder/driver.



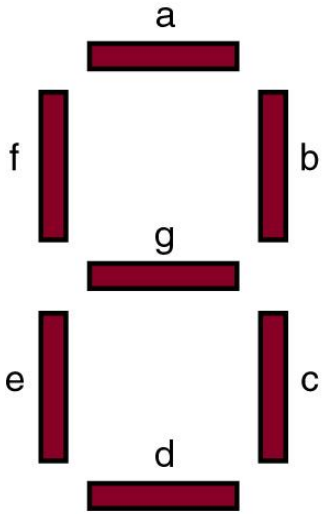


Suitable for directly driving loads such as indicator LEDs or lamps, relays, or dc motors.

Termed a *driver* because this IC has open-collector outputs that operate at higher current/voltage limits than a normal TTL output.

5-2 BCD-to-7 Segment Decoder/Drivers

- The 7-segment display is a common way to display decimal or hexadecimal characters.
  - One common arrangement uses light emitting diodes (LEDs) for each segment.

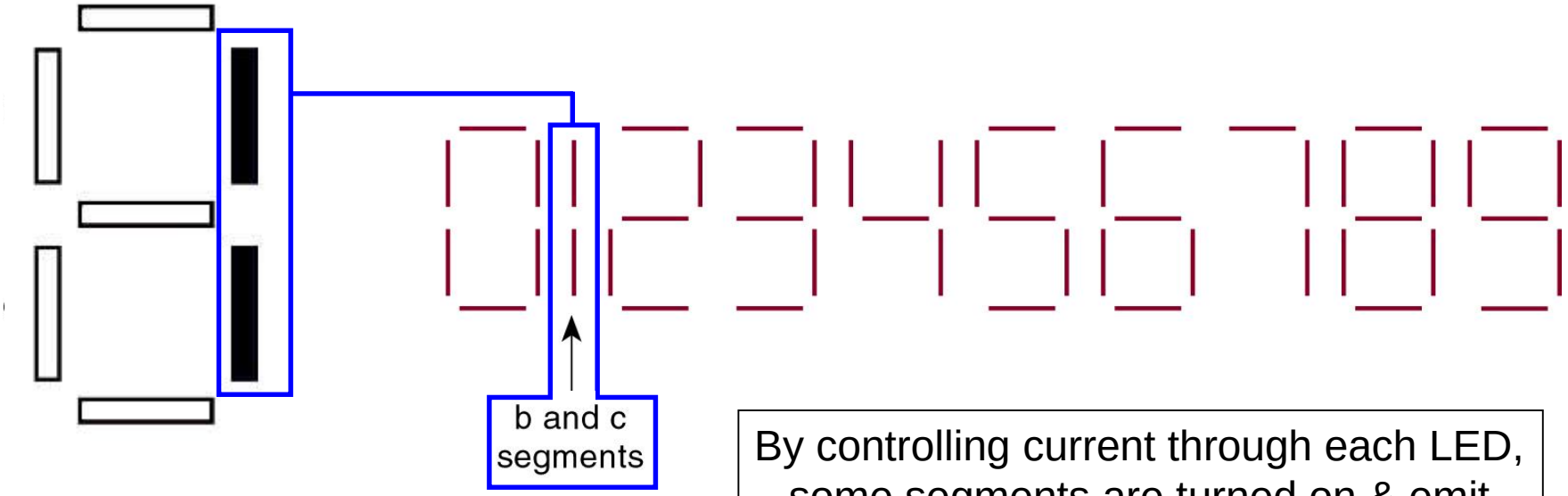


Diodes allow current to flow in one direction, but block flow in the other direction.

When the LED anode is more positive than the cathode by approximately 2 V, the LED will light up.

5-2 BCD-to-7 Segment Decoder/Drivers

- The 7-segment display is a common way to display decimal or hexadecimal characters.
  - One common arrangement uses light emitting diodes (LEDs) for each segment.

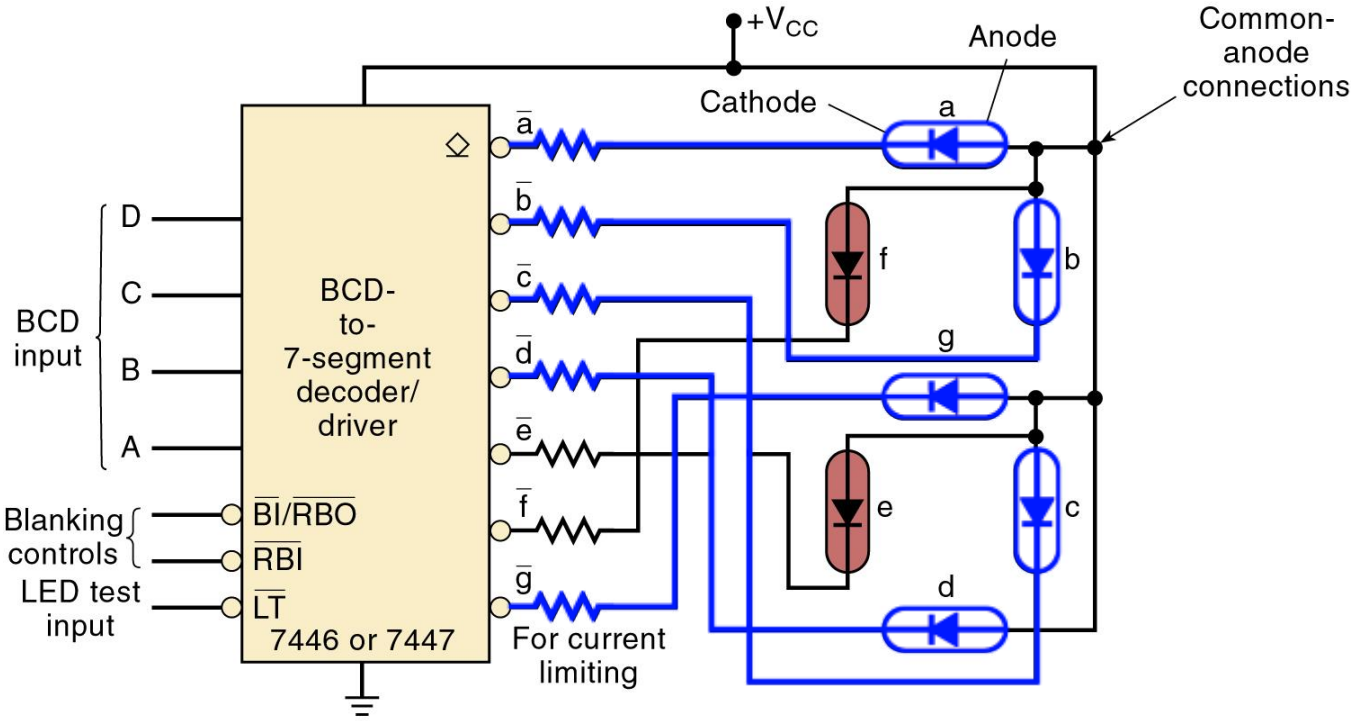


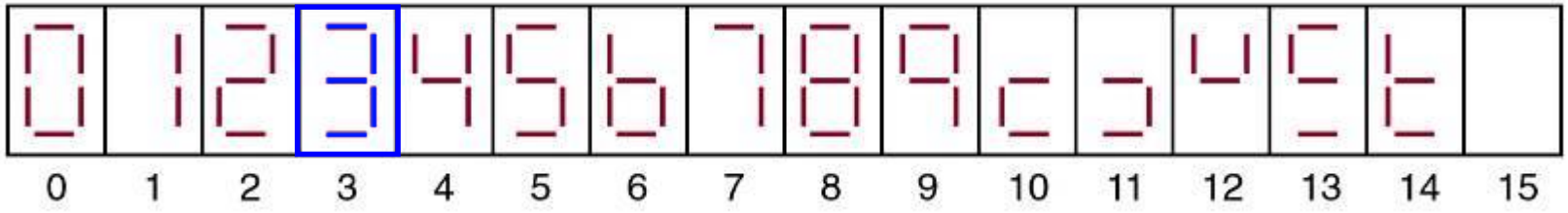
By controlling current through each LED, some segments are turned on & emit light, while others are turned off, which generates the desired character pattern.

5-2 BCD-to-7 Segment Decoder/Drivers

BCD-to-7-segment decoder/driver.

The 7446/47 activates specific segment patterns in response to input codes

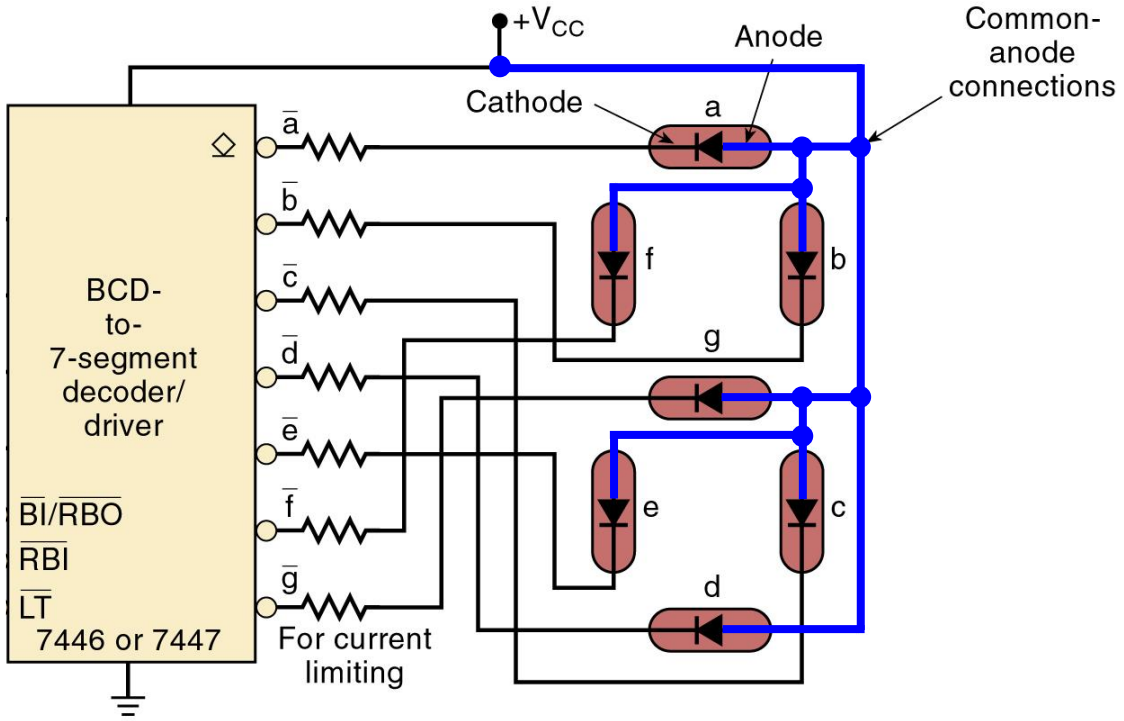




BCD-to-7-segment decoder/driver.

This is a **common-anode** LED display.

The anodes all of segments are tied together to V_{CC} .



Another type uses a **common-cathode** method, with each segment requiring 10 to 20 mA of current. TTL/CMOS devices are normally not used to drive a common-cathode display directly—a transistor interface circuit is often used

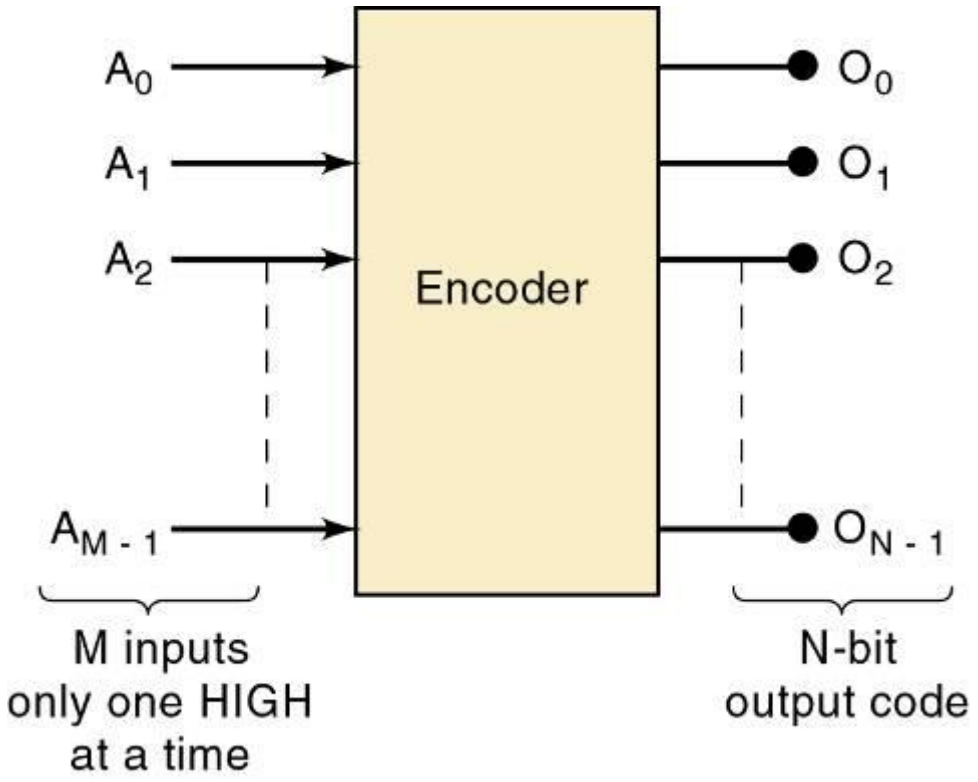
5-3 Encoders

- Most decoders accept an input code & produce a HIGH (or LOW) at *one* and *only one* output line.
- The opposite of decoding process is encoding. Performed by a logic circuit called an encoder.

An encoder has a number of input lines, only **one** of which is activated at a given time.

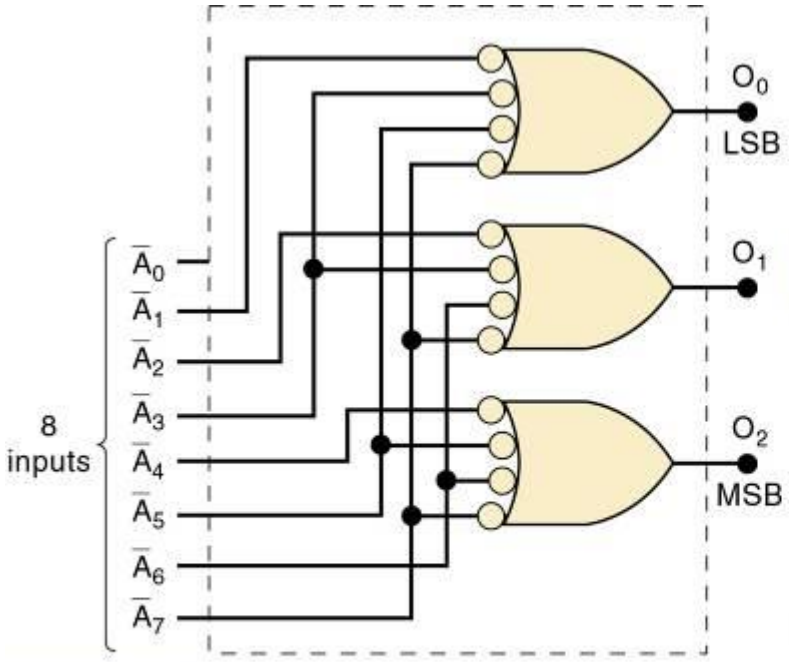
Shown is an encoder with *M* inputs and *N* outputs. Inputs are active-HIGH, which means that they are normally LOW.

It produces an *N*-bit output code, depending on which input is activated.



5-3 Encoders

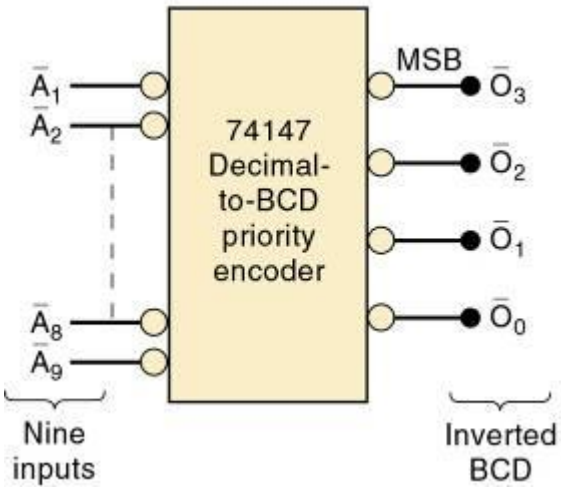
- An *octal-to-binary encoder (8-line-to-3-line encoder)* accepts eight input lines, producing a three-bit output code corresponding to the input.



Inputs								Outputs		
$\overline{A}_0$	$\overline{A}_1$	$\overline{A}_2$	$\overline{A}_3$	$\overline{A}_4$	$\overline{A}_5$	$\overline{A}_6$	$\overline{A}_7$	O_2	O_1	O_0
X	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1

Logic circuit for an octal-to-binary (8-line-to-3-line) encoder. Only one input should be active at one time.

- A **priority encoder** ensures that when two or more inputs are activated, the output code will correspond to the highest-numbered input.



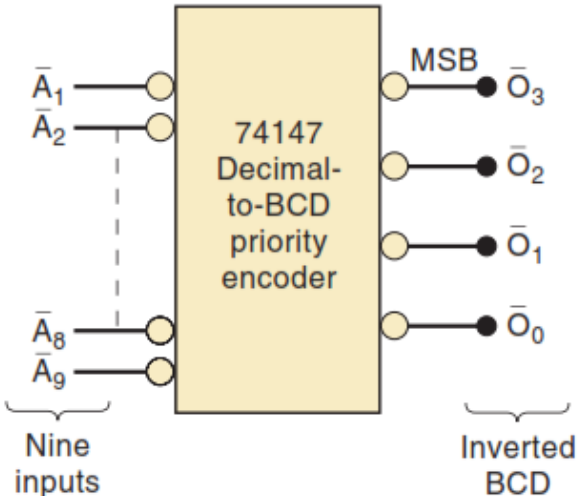
$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$\bar{A}_8$	$\bar{A}_9$	$\bar{O}_3$	$\bar{O}_2$	$\bar{O}_1$	$\bar{O}_0$
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

X = either 0 or 1

It has nine active-LOW inputs represent decimal digits 1 through 9, producing *inverted* BCD code corresponding to the highest-numbered activated input.

5-3 Encoders

74147 Decimal-to-BCD Priority encoder



$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$\bar{A}_8$	$\bar{A}_9$	$\bar{O}_3$	$\bar{O}_2$	$\bar{O}_1$	$\bar{O}_0$
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

X = either 0 or 1

5-4 Switch Encoders

- A **switch encoder** can be used when BCD data must be entered manually into a digital system.
  - The 10 switches might be the keyboard switches on a calculator—representing digits 0 through 9.

5-4 Encoders

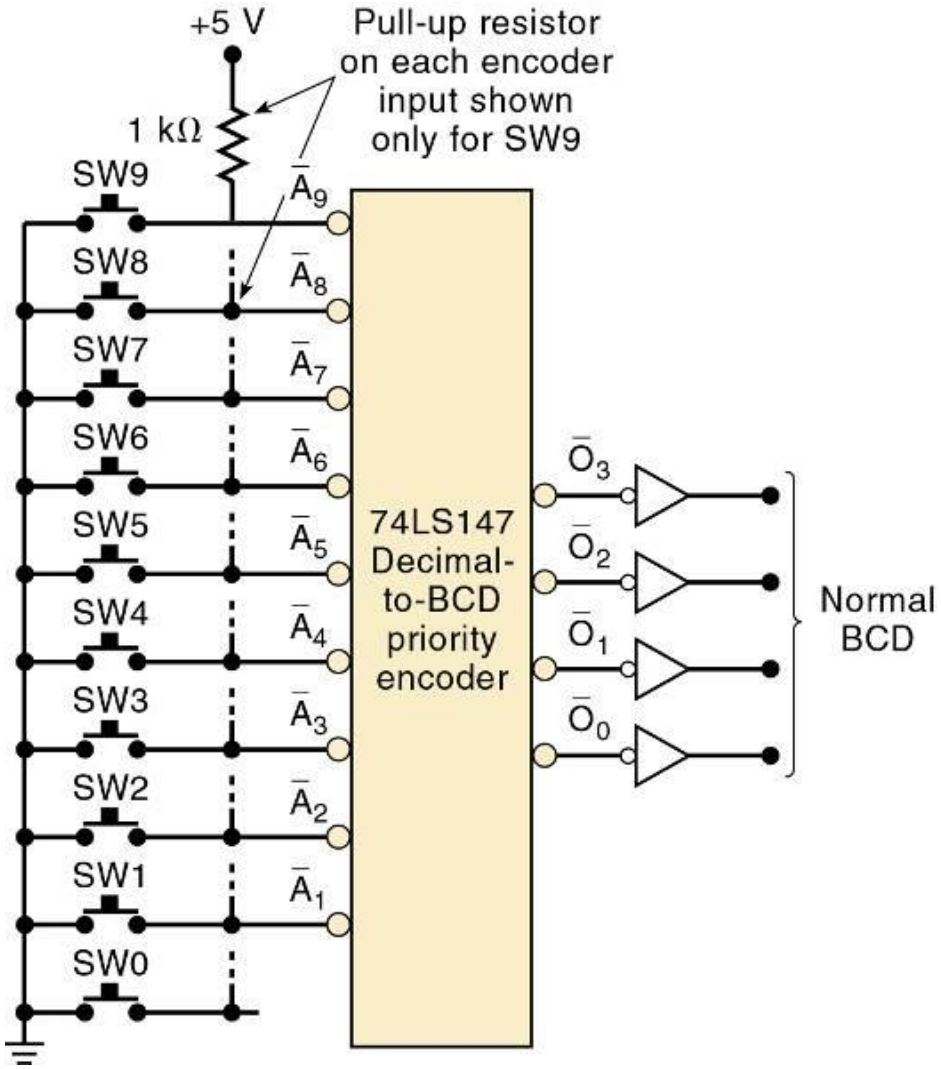
The switches are of the normally open type,

So the encoder inputs are all normally HIGH.

BCD output is 0000.

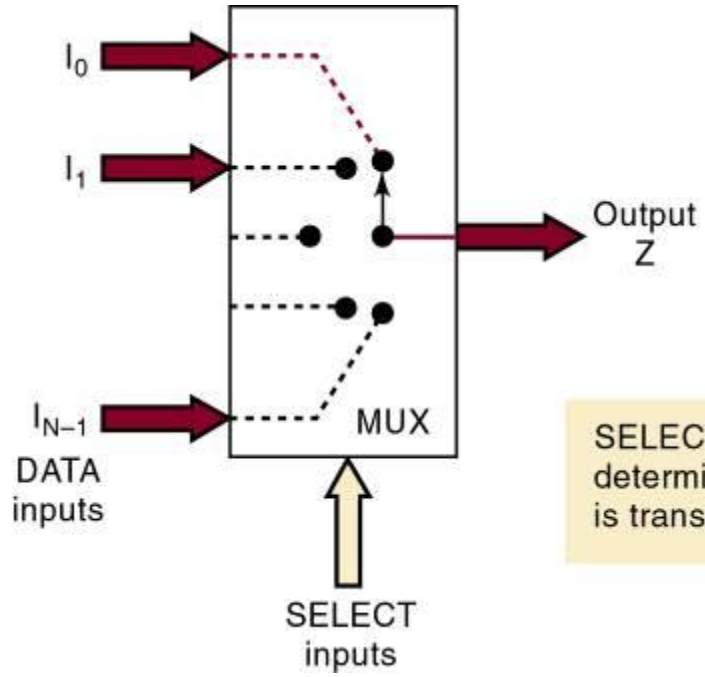
When a key is depressed, the circuit will produce the BCD code for that digit.

The 74LS147 is a *priority* encoder, so simultaneous key depressions produce the BCD code for the *higher-numbered* key.



5-5 Multiplexers (Data Selectors)

- A **multiplexer (MUX)** selects 1 of *N* input data sources and transmits the selected data to a single output—called **multiplexing**.
  - A *digital multiplexer* or *data selector* is a logic circuit that performs the same task.

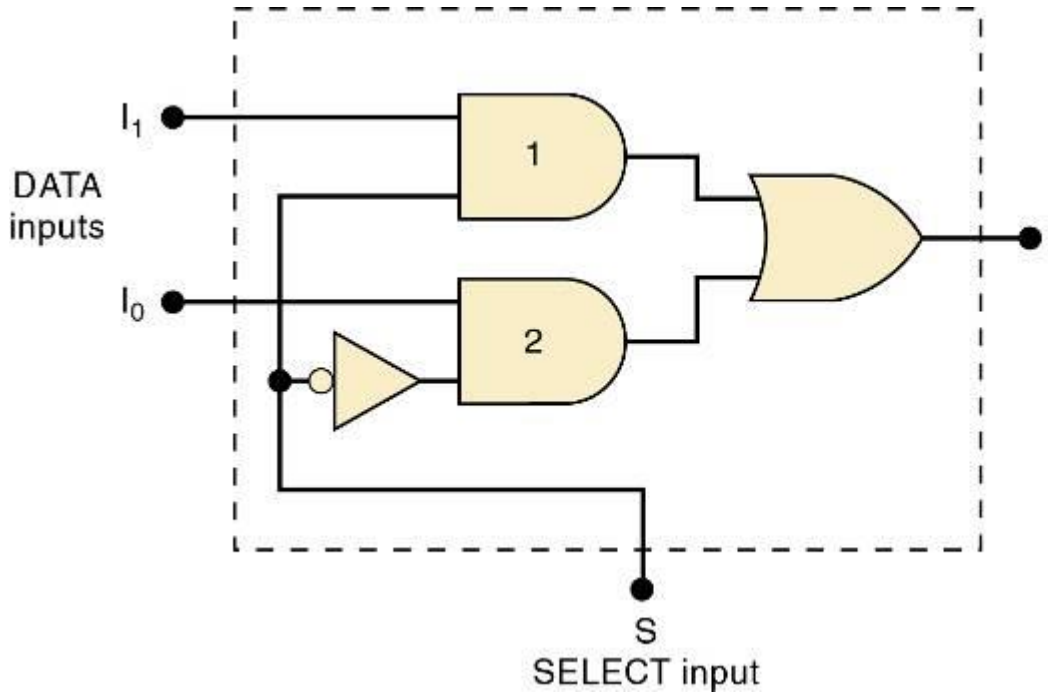


Routing control of desired data input to output by SELECT inputs—referred to as ADDRESS inputs.

SELECT input code determines which input is transmitted to output Z.

5-5 Multiplexers (Data Selectors)

- A two-input MUX could be used in a digital system that uses two different MASTER CLOCK signals.
  - A high-speed clock in one mode and a slow-speed clock for the other.

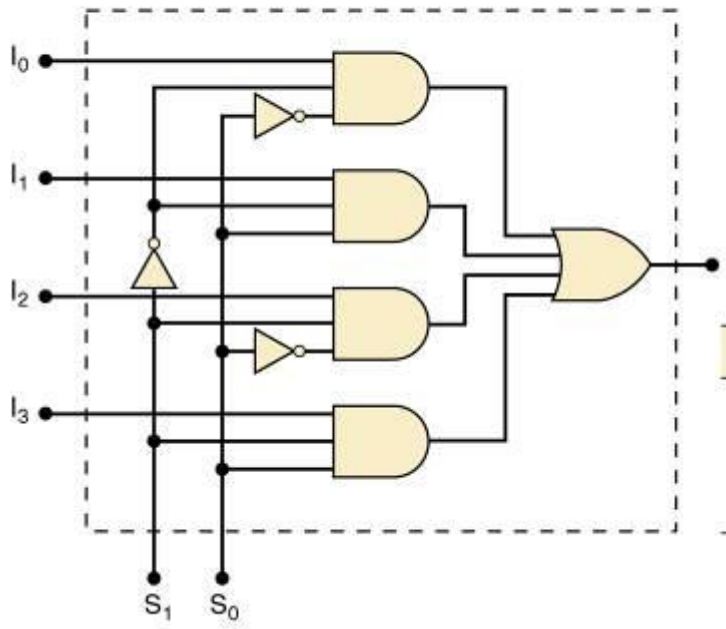


$Z = I_0 \cdot \bar{S} + I_1 \cdot S$

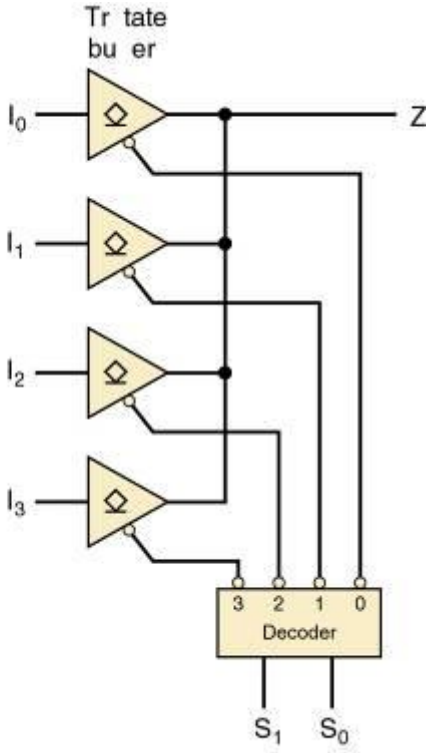
S	Output
0	$Z = I_0$
1	$Z = I_1$

5-5 Multiplexers (Data Selectors)

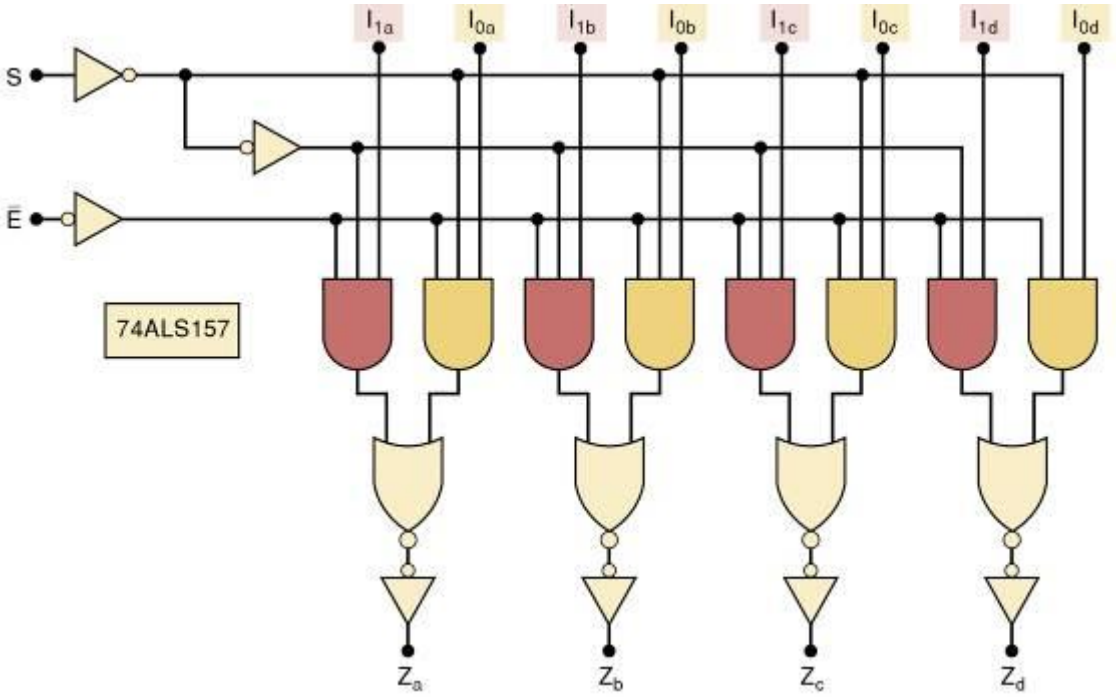
- Two-, four-, eight-, and 16-input multiplexers are available in the TTL and CMOS logic families.
  - These basic ICs can be combined for multiplexing a larger number of inputs.



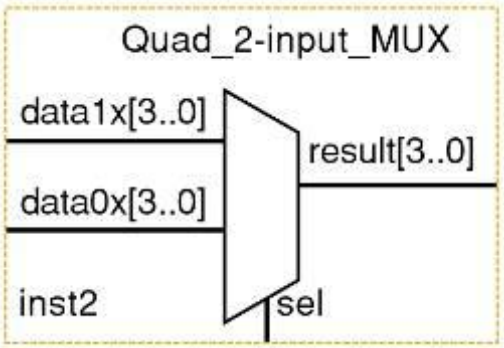
S_1	S_0	Output
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$

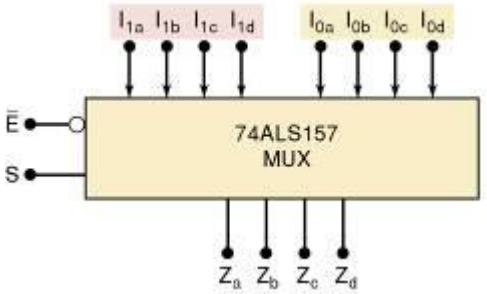


The 74ALS157 contains four two-input multiplexers



Ē	S	Z_a	Z_b	Z_c	Z_d
H	X	L	L	L	L
L	L	I_{0a}	I_{0b}	I_{0c}	I_{0d}
L	H	I_{1a}	I_{1b}	I_{1c}	I_{1d}



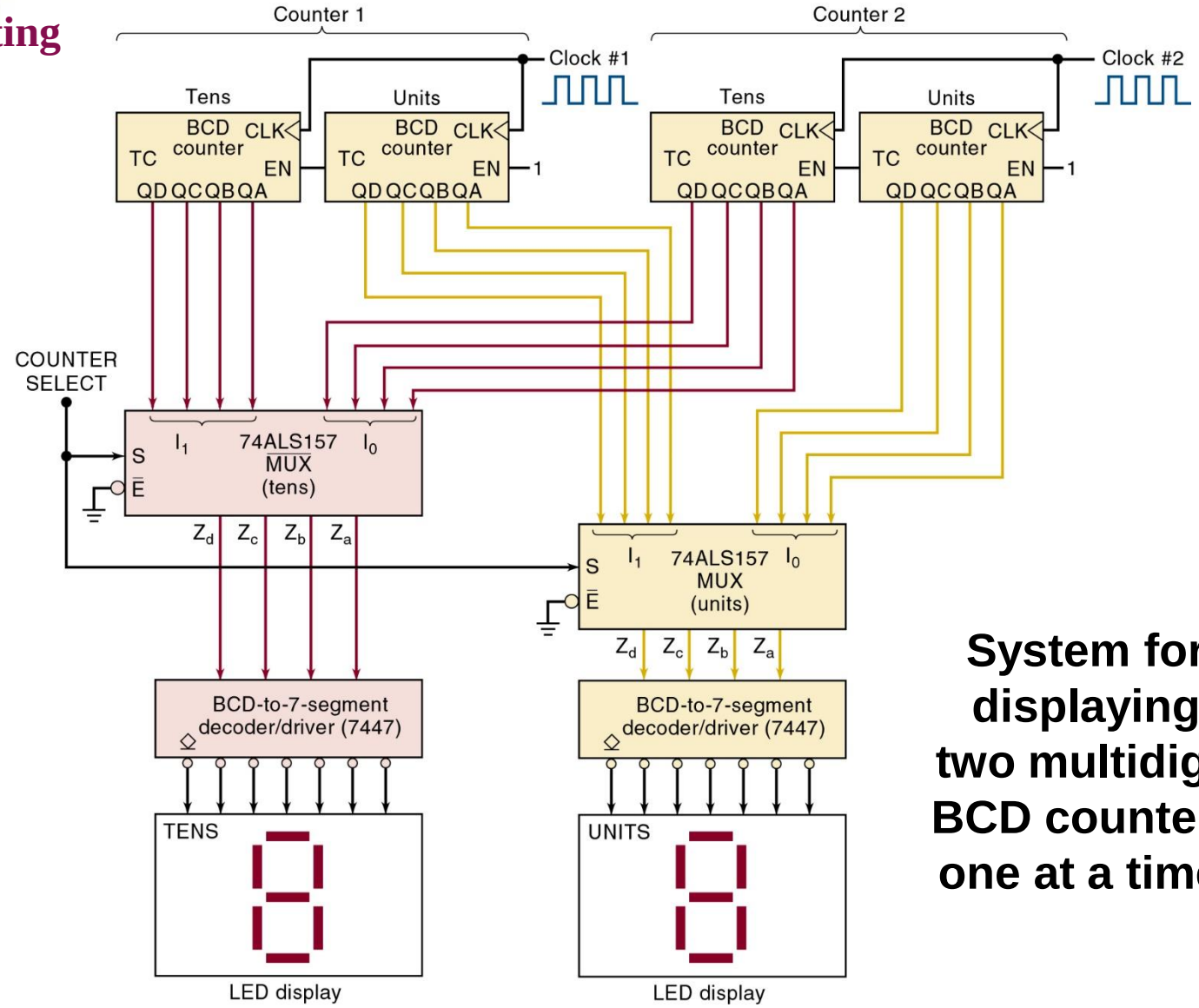


5-6 Multiplexer Applications

- Multiplexer circuits find numerous and varied applications in digital systems of all types.
  - Data selection/routing, parallel-to-serial conversion.
  - Operation sequencing.
  - Waveform/logic-function generation.

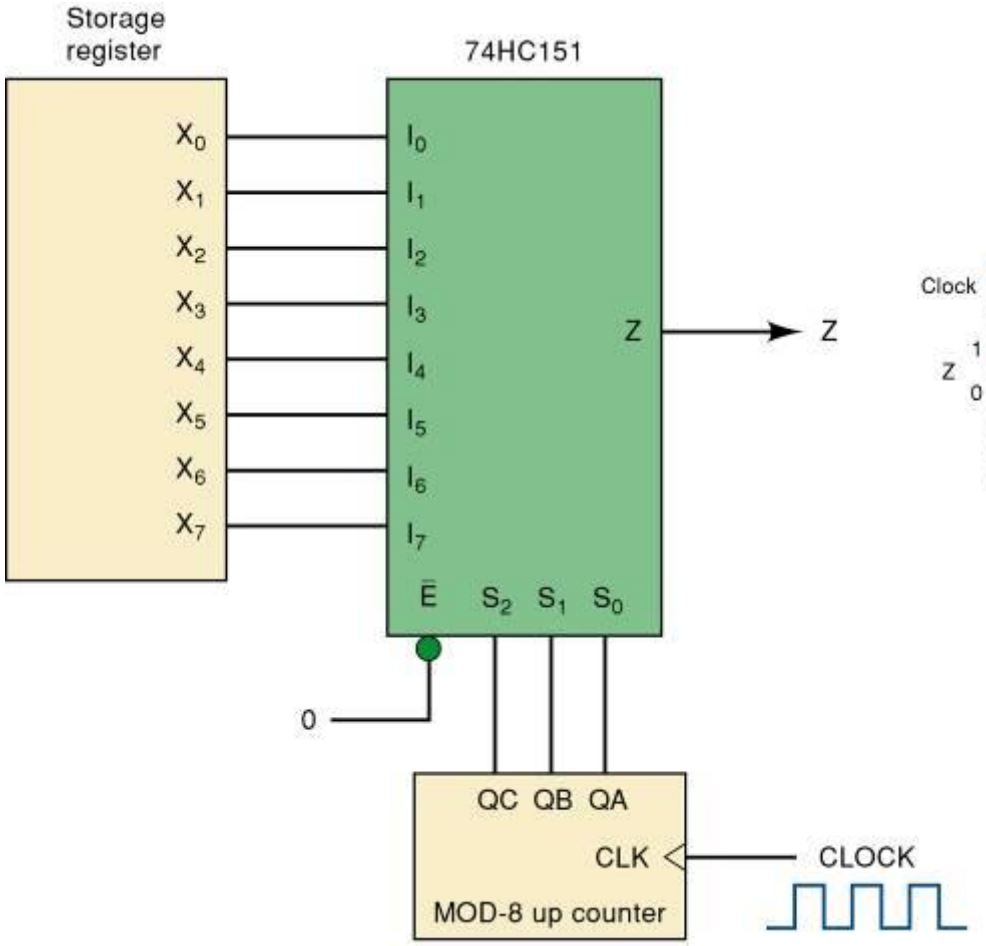
5-6 Multiplexer Applications

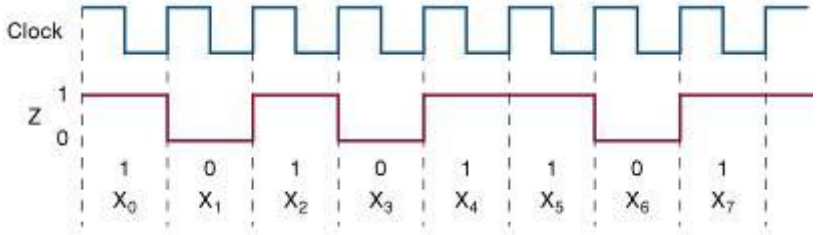
Data routing



System for displaying two multidigit BCD counters one at a time.

Parallel-to-serial converter.

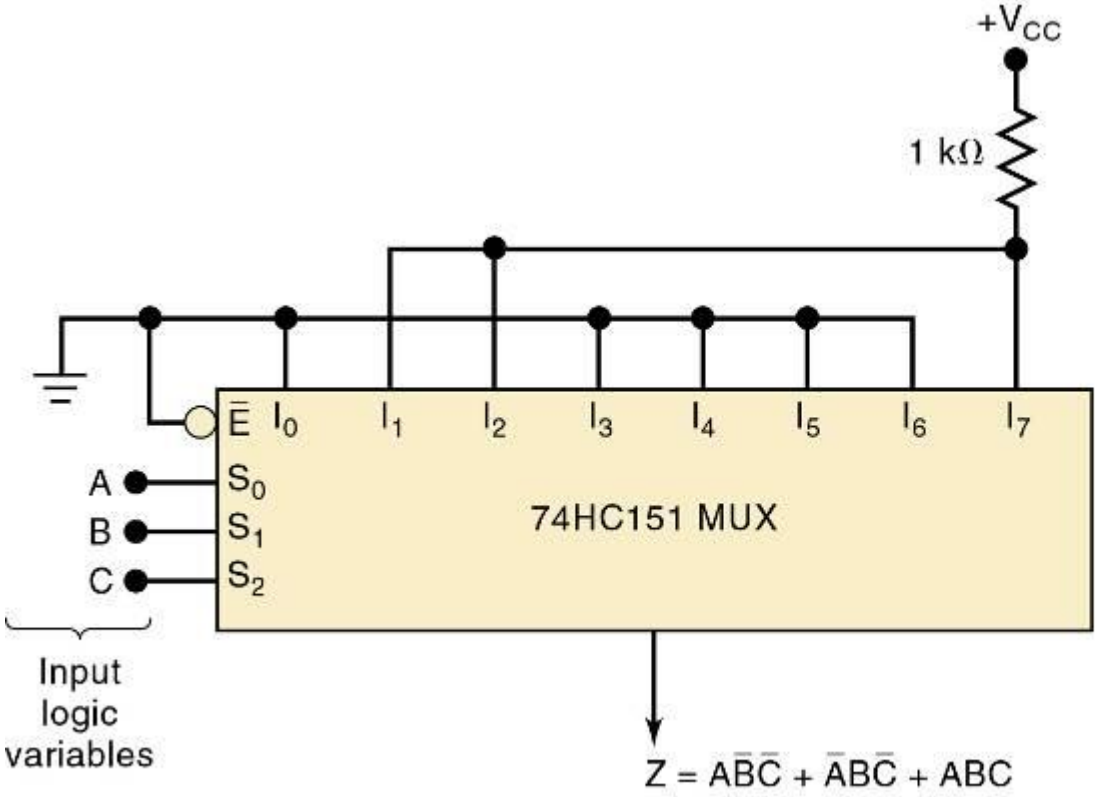




Waveforms for $X_7X_6X_5X_4X_3X_2X_1X_0$

10110101

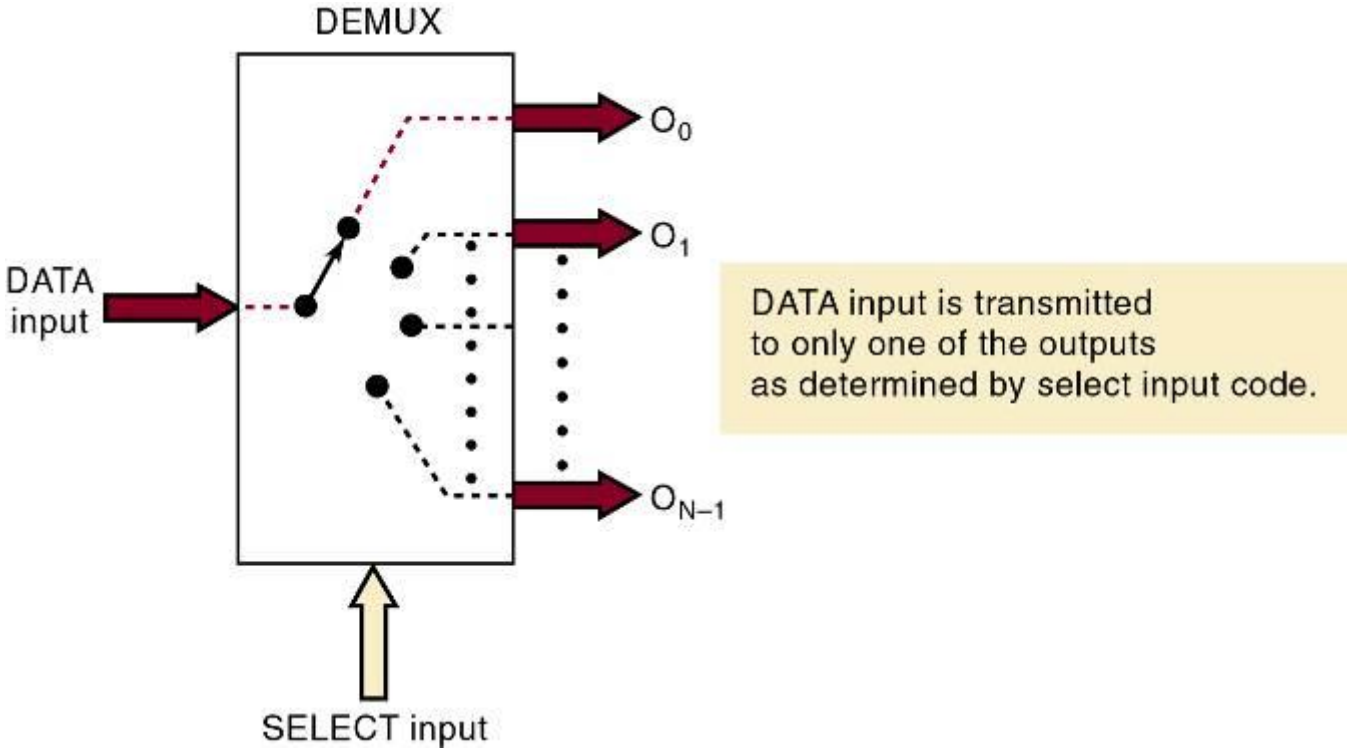
Multiplexer used to implement a logic function described by the truth table.



C	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

5-7 Demultiplexers (Data Distributors)

- A **demultiplexer (DEMUX)** takes a single input and distributes it over several outputs.
  - The select input code determines to which output the DATA input will be transmitted.

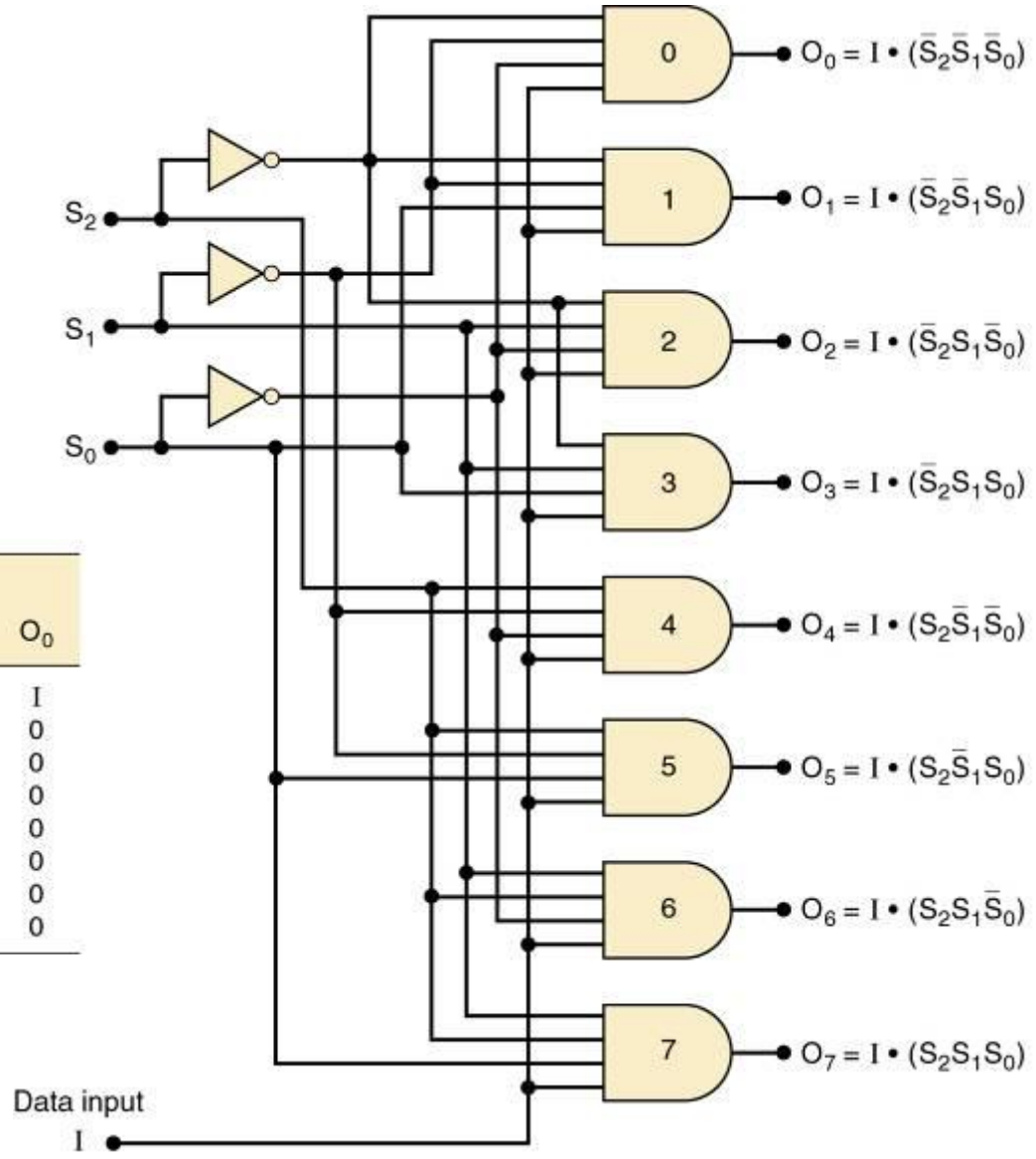


5-7 Demultiplexers (Data Distributors)

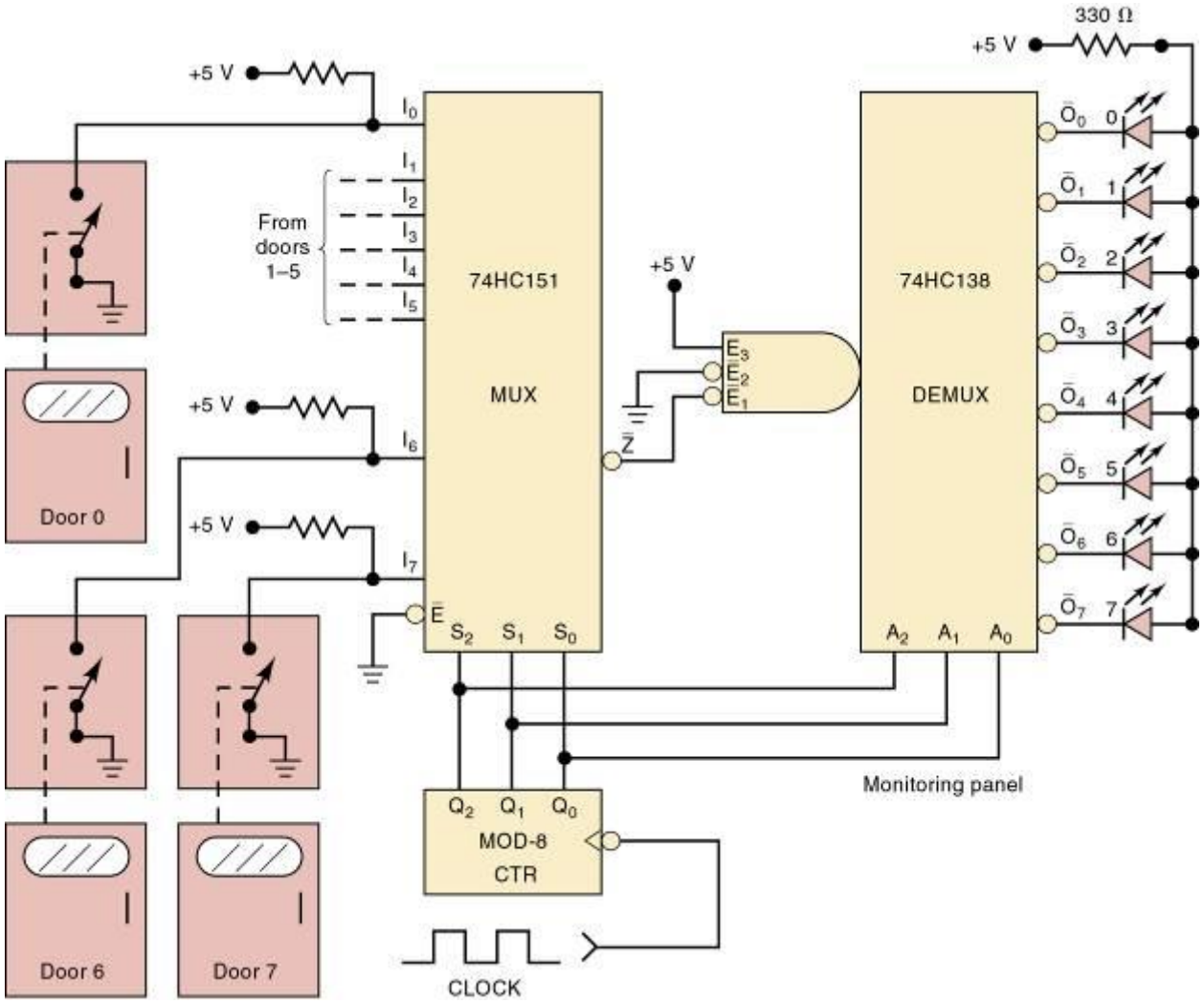
A 1 line to 8 line demultiplexer.

Select Code			Outputs							
S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

Note: I is the data input



Security monitoring system using the 74ALS138.



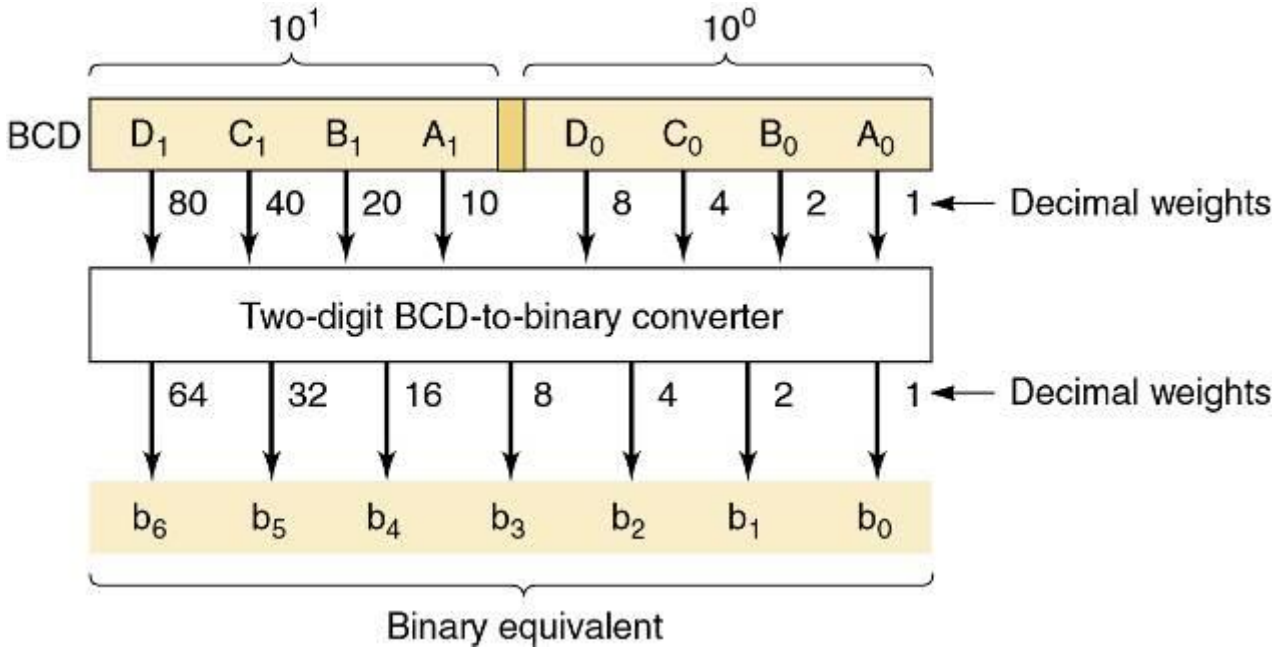
The system shown can handle eight doors, but can be expanded to any number.

The door switches are data inputs to the MUX.

They produce a HIGH when a door is open and a LOW when it is closed.

5-8 Code Converters

- A code converter is a logic circuit that changes data presented in one type of binary code to another type of binary code.



Basic idea of a two-digit BCD-to-binary converter.

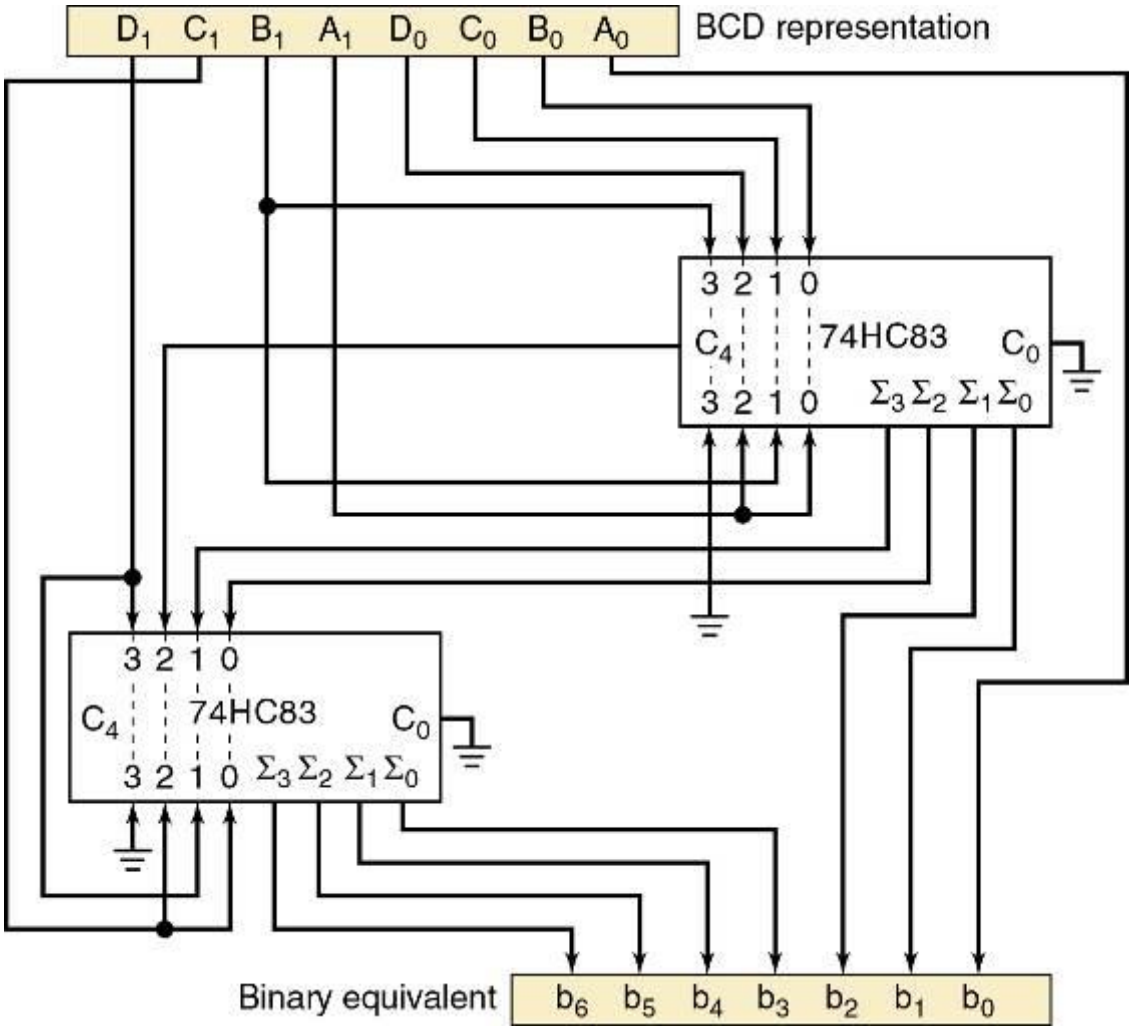
5-8 Code Converters

- The bits in a BCD representation have decimal weights that are 8, 4, 2, 1 within each code group.
  - That differ by a factor of 10 from one code group (decimal digit) to the next.

The decimal weight of each bit in the BCD representation can be converted to its binary equivalent.

BCD Bit	Decimal Weight	Binary Equivalent						
		b_6	b_5	b_4	b_3	b_2	b_1	b_0
A_0	1	0	0	0	0	0	0	1
B_0	2	0	0	0	0	0	1	0
C_0	4	0	0	0	0	1	0	0
D_0	8	0	0	0	1	0	0	0
A_1	10	0	0	0	1	0	1	0
B_1	20	0	0	1	0	1	0	0
C_1	40	0	1	0	1	0	0	0
D_1	80	1	0	1	0	0	0	0

5-8 Code Converters

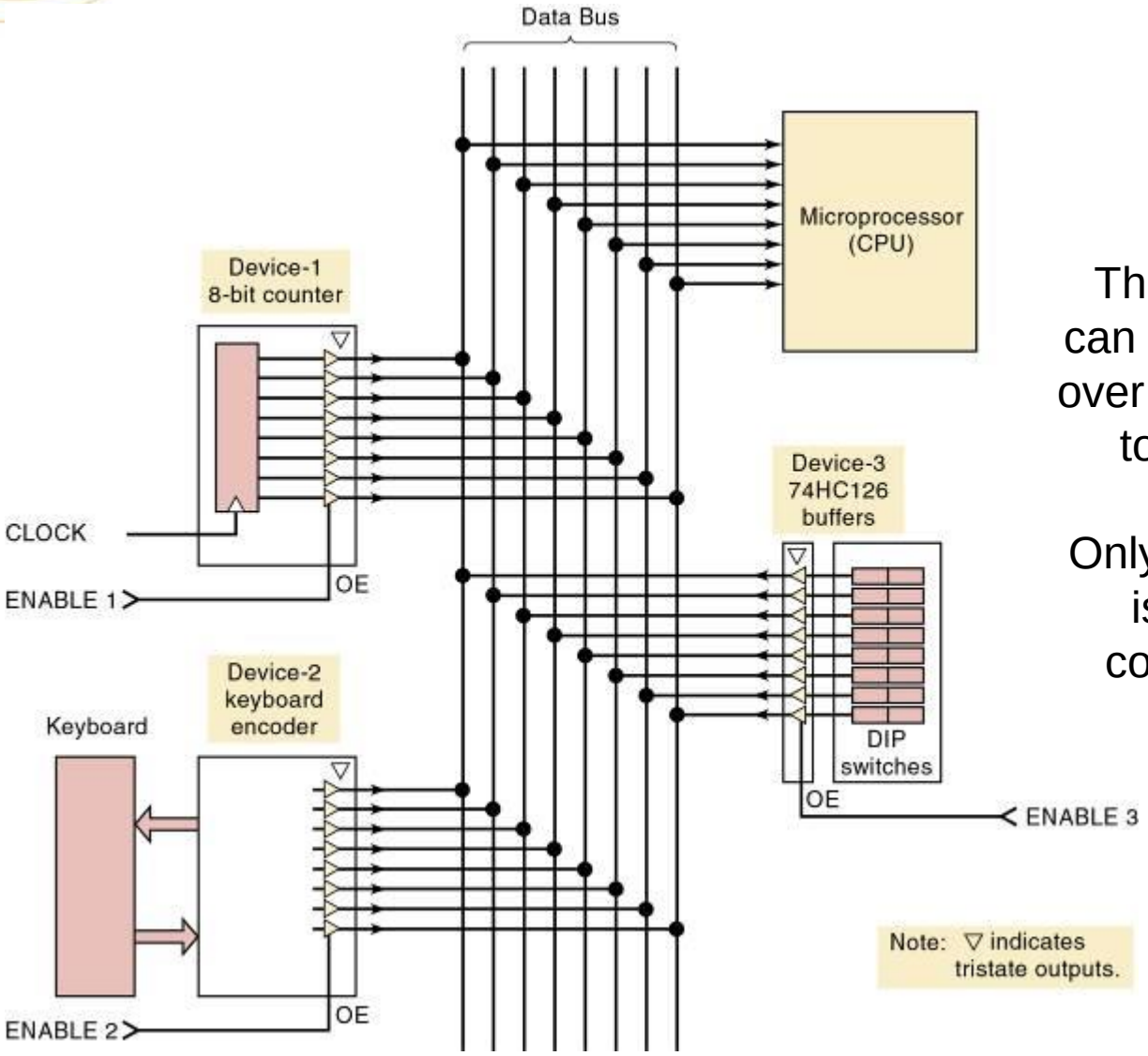


One way to implement the logic circuit that performs conversion process is to use binary adder circuits.

5-9 Data Busing

- In computers, transfer of data takes place over a common set of connecting lines called a **data bus**.
  - Devices tied to the data bus will often have tri-state outputs, or be tied to the data bus by tristate buffers.
- Devices commonly connected to a data bus:
  - Microprocessors; Semiconductor memory chips.
  - Digital-to-analog and analog-to-digital converters.

5-9 Data Busing



Three different devices can transmit eight-bit data over an eight-line data bus to a microprocessor.

Only one device at a time is enabled—so bus contention is avoided.