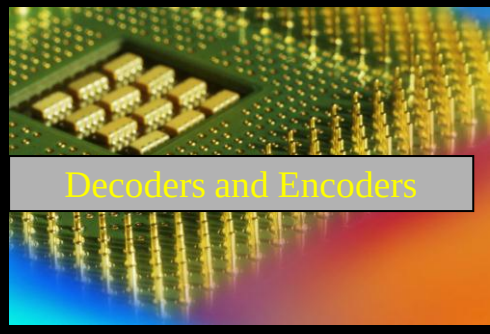
EEE251

Logic Design and Circuits

Assist.Prof. Yavuz BAYAM



Decoders and Encoders

December 9 , 2013

DECODER

•A decoder is a logic circuit that accepts a set of inputs that represents a binary number and activates only the output that corresponds to the input number.

•In other words, a decoder circuit looks at its inputs, determines which binary number is present there, and activates the one output that corresponds to that number ; all other outputs remain inactive

DECODER

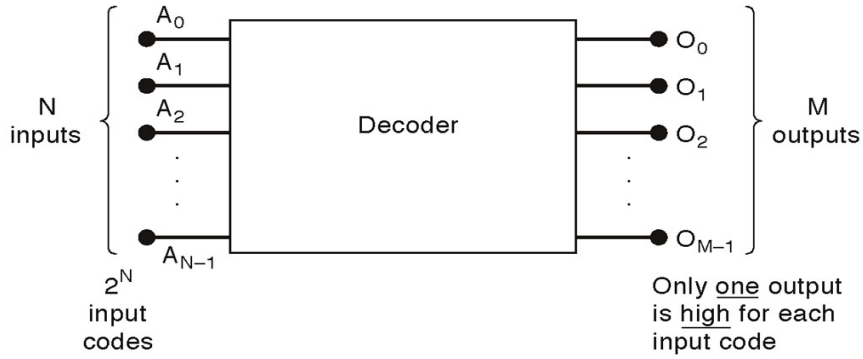
In its general form, a decoder has N input lines to handle N bits and form one to 2^N output lines to indicate the presence of one or more N-bit combinations.

The basic binary function

- An AND gate can be used as the basic decoding element because it produces a HIGH output only when all inputs are HIGH

Refer next slide for example

General decoder diagram

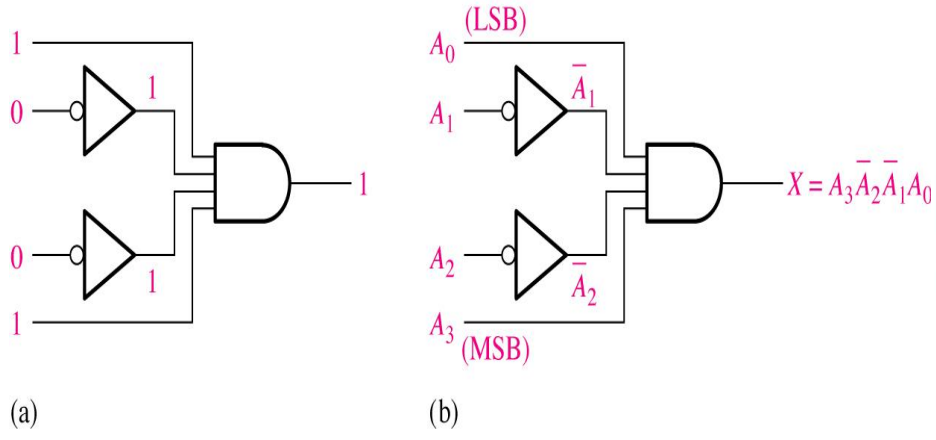


There are 2^N possible input combinations, from A_0 to A_{N-1}.

For each of these input combinations only one of the *M* outputs will be active *HIGH* (1), all the other outputs are *LOW* (0).

DECODER

Decoding logic for the binary code 1001 with an active-HIGH output.

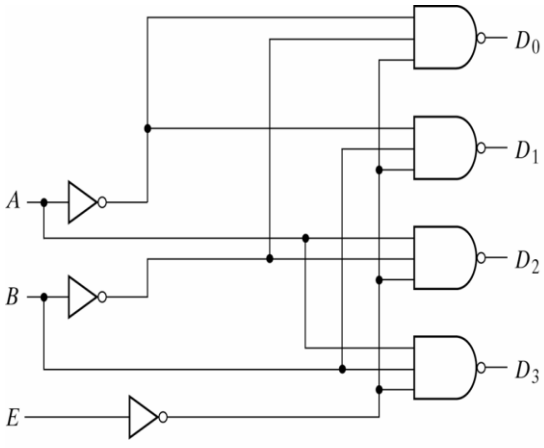


DECODER

- If an active-LOW output (74138, one of the output will low and the rest will be high) is required for each decoded number, the entire decoder can be implemented with
  1. NAND gates
  2. Inverters

- If an active-HIGH output (74139, one of the output will high and the rest will be low) is required for each decoded number, the entire decoder can be implemented with
  - AND gates
  - Inverters

2-to-4-Line Decoder (with Enable input)-Active LOW output (1)...



(a) Logic diagram

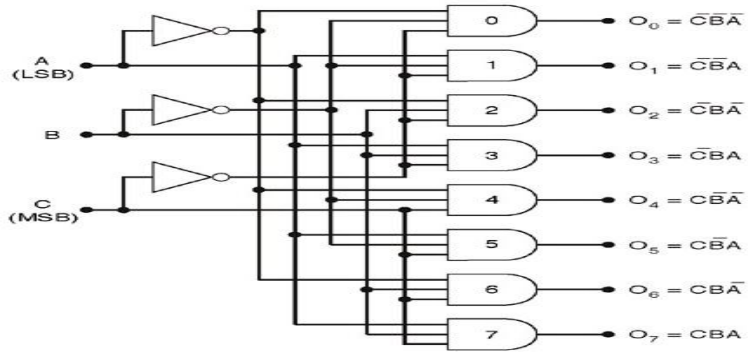
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

2-to-4-Line Decoder (with Enable input)-Active LOW output (2)

- The circuit operates with complemented outputs and a complement enable input. The decoder is enabled when E is equal to 0.
- Only one output can be equal to 0 at any given time, all other outputs are equal to 1.
- The output whose value is equal to 0 represents the minterm selected by inputs A and B
- The circuit is disabled when E is equal to 1.

3-8 line decoder (active-HIGH)

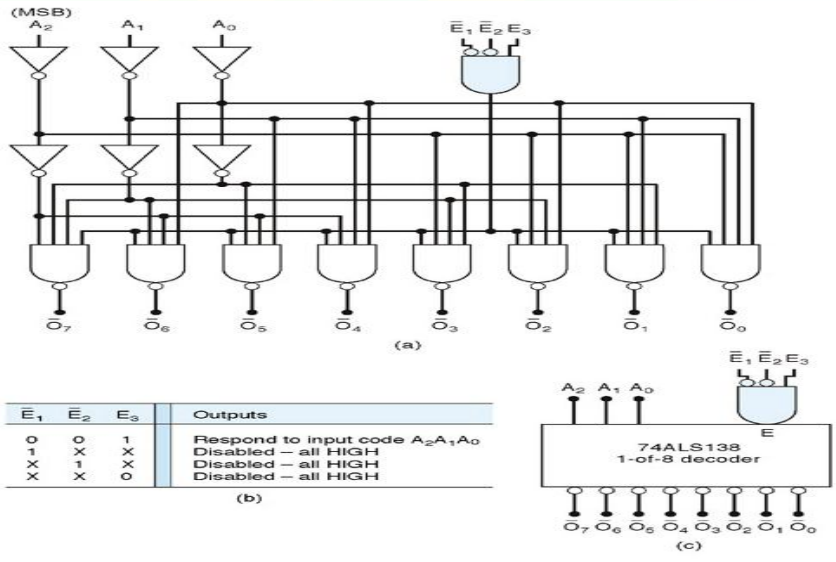


C	B	A	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

3-8 line decoder (active-HIGH)

- This decoder can be referred to in several ways. It can be called a 3-line-to- 8-line decoder, because it has three input lines and eight output lines.
- It could also be called a binary-octal decoder or converters because it takes a three bit binary input code and activates the one of the eight outputs corresponding to that code.
- It is also referred to as a 1-of-8 decoder, because only 1 of the 8 outputs is activated at one time.

Logic diagram of 74138 (Example of a 3–Bit Decoder)



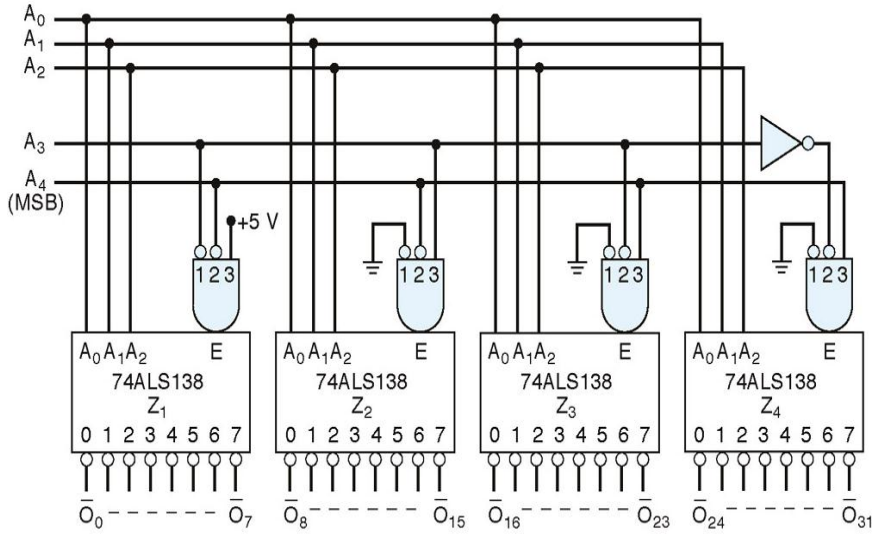
Truth table of 74138 (Example of a 3– 8 Bit Decoder) active-LOW

Inputs						Outputs							
Enables			2^2	2^1	2^0	Active–*LOW*							
E_3	$\overline{E}_1$	$\overline{E}_2$	A_2	A_1	A_0	$\overline{O}_7$	$\overline{O}_6$	$\overline{O}_5$	$\overline{O}_4$	$\overline{O}_3$	$\overline{O}_2$	$\overline{O}_1$	$\overline{O}_0$
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H	**L**
H	L	L	L	L	H	H	H	H	H	H	H	**L**	H
H	L	L	L	H	L	H	H	H	H	H	**L**	H	H
H	L	L	L	H	H	H	H	H	H	**L**	H	H	H
H	L	L	H	L	L	H	H	H	**L**	H	H	H	H
H	L	L	H	L	H	H	H	**L**	H	H	H	H	H
H	L	L	H	H	L	H	**L**	H	H	H	H	H	H
H	L	L	H	H	H	**L**	H	H	H	H	H	H	H

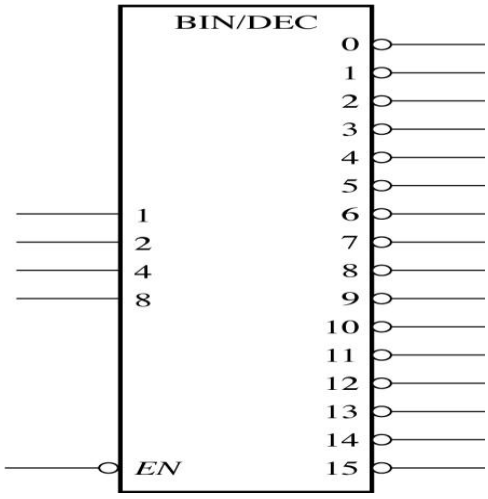
74138 (Example of a 3– 8 Bit Decoder)

- There is an enable function on this device, a *LOW* level on each input E'_1, and E'_2, and a *HIGH* level on input E_3, is required in order to make the enable gate output *HIGH*.
- The enable is connected to an input of each NAND gate in the decoder, so it must be *HIGH* for the NAND gate to be enabled.
- If the enable gate is not activated then all eight decoder outputs will be *HIGH* regardless of the states of the three input variables A_0, A_1, and A_2 .

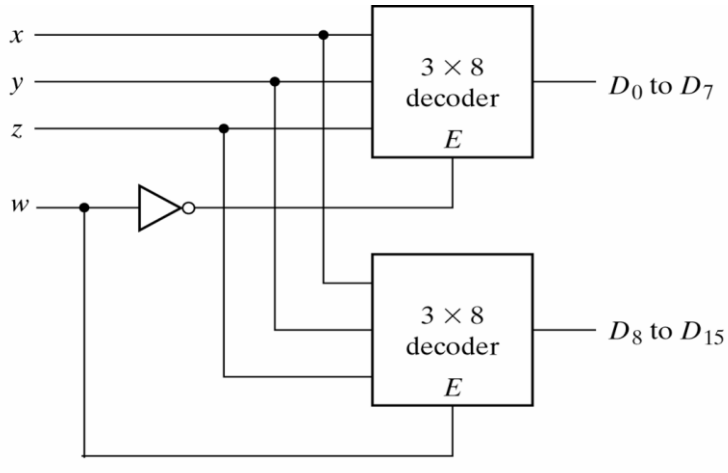
Example of a 5 to 32 Bit Decoder



Logic symbol for a 4-line-to-16-line (1-of-16) decoder . 74HC154



4-line-to-16 line Decoder constructed with two 3-line-to-8 line decoders (1)...

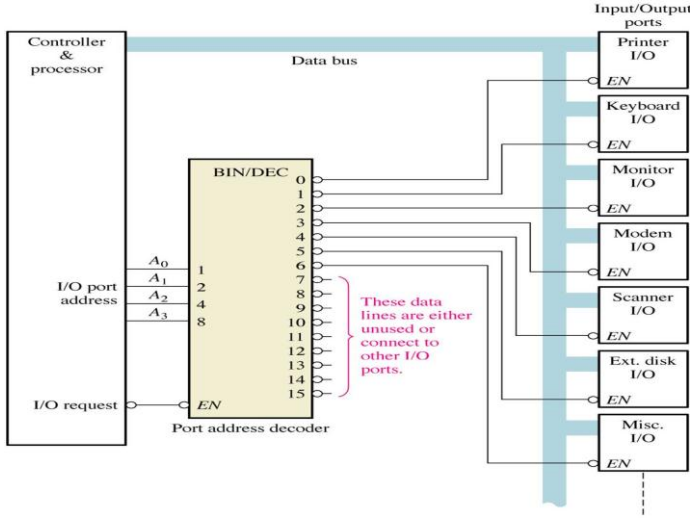


4-line-to-16 line Decoder constructed with two 3-line-to-8 line decoders (2)

- When w=0, the top decoder is enabled and the other is disabled.
- The bottom decoder outputs are all 0's , and the top eight outputs generate min-terms 0000 to 0111.
- When w=1, the enable conditions are reversed.
- The bottom decoder outputs generate min-terms 1000 to 1111, while the outputs of the top decoder are all 0's.

Application example

A simplified computer I/O port system with a port address decoder with only four address lines shown.



Application example

•Decoders are used in many types of applications. One example is in computers for I/O selection as in previous slide

•Computer must communicate with a variety of external devices called peripherals by sending and/or receiving data through what is known as input/output (I/O) ports

•Each I/O port has a number, called an address, which uniquely identifies it.

•When the computer wants to communicate with a particular device, it issues the appropriate address code for the I/O port to which that particular device is connected .

•The binary port address is decoded and appropriate decoder output is activated to enable the I/O port

•Binary data are transferred within the computer on a data bus, which is a set of parallel lines

BCD -to- Decimal decoders

•The BCD- to-decimal decoder converts each BCD code into one of Ten Positionable decimal digit indications.

•It is frequently referred as a 4-line -to- 10 line decoder

•The method of implementation is that only ten decoding gates are required because the BCD code represents only the ten decimal digits 0 through 9.

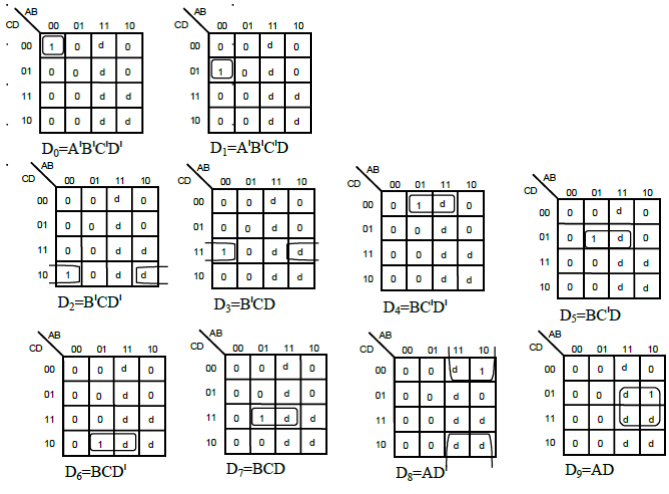
•Each of these decoding functions is implemented with NAND gates to provide active -LOW outputs.

•If an active HIGH output is required, AND gates are used for decoding

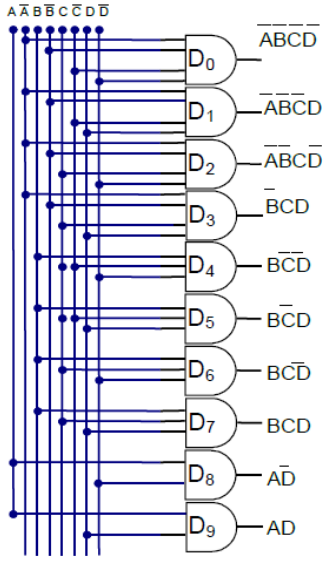
BCD -to- Decimal decoders

A	B	C	D	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

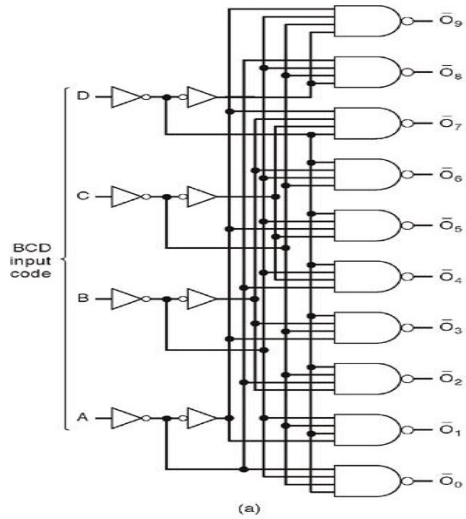
BCD -to- Decimal decoders

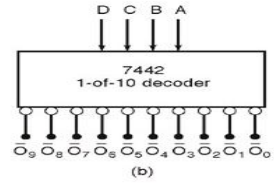


BCD -to- Decimal decoders



Logic diagram of BCD - decimal decoder (Active LOW output)





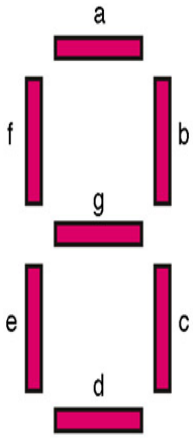
Inputs

D	C	B	A	Active Output
L	L	L	L	$\bar{O}_0$
L	L	L	H	$\bar{O}_1$
L	L	H	L	$\bar{O}_2$
L	L	H	H	$\bar{O}_3$
L	H	L	L	$\bar{O}_4$
L	H	L	H	$\bar{O}_5$
L	H	H	L	$\bar{O}_6$
L	H	H	H	$\bar{O}_7$
H	L	L	L	$\bar{O}_8$
H	L	L	H	$\bar{O}_9$
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

(c)

BCD-7segment decoders/drivers



- Most digital equipment has some means for displaying information in a form that can be understood by the user.
- This information is often numerical data but also be alphanumeric.
- One of the simplest and most popular methods for displaying numerical digits uses a 7-segment configuration to form digital characters 0 to 9 and some times the hex characters A to F

BCD-7segment decoders/drivers

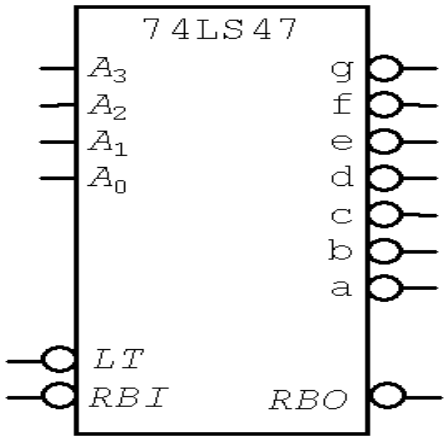
- One common arrangements uses light-emitting diodes (LED's) for each segment.
- By controlling the current thru each LED, some segments will be light and others will be dark so that desired character pattern will be generated

Figure shows the segment pattern that are used to display the various digits. For example, to display a "6" the segments a,c,d,e,f and g are made bright while segment b is dark

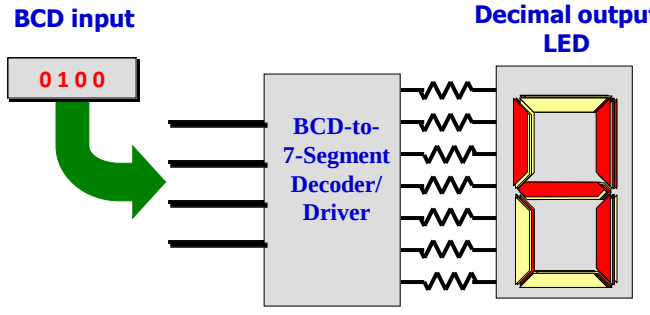
7-segment decoder

- A BCD-7 segment decoder/driver is used to take four-bit BCD input and provide the outputs that will pass current through the appropriate segments to display the decimal digit.
- The logic for this decoder is more complicated than the logic of decoders of earlier case, because each output is activated for more than one combination of inputs.

74LS47 (BCD–to–Seven–Segment Decoder)



BCD-7segment decoders/drivers



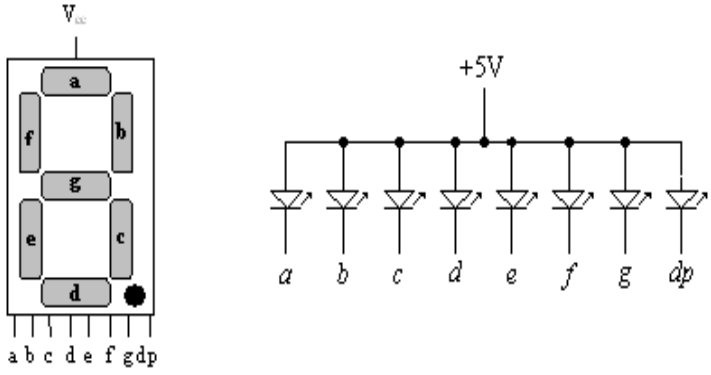
- **Electronic decoders are available in IC form.**
- **This decoder translates from BCD to decimal.**
- **Decimals are shown on an 7-segment LED display.**
- **This IC also drives the 7-segment LED display.**

7-segment display

- There are two types of 7–segment LED displays;
- common - anode
- common – cathode

Common Anode

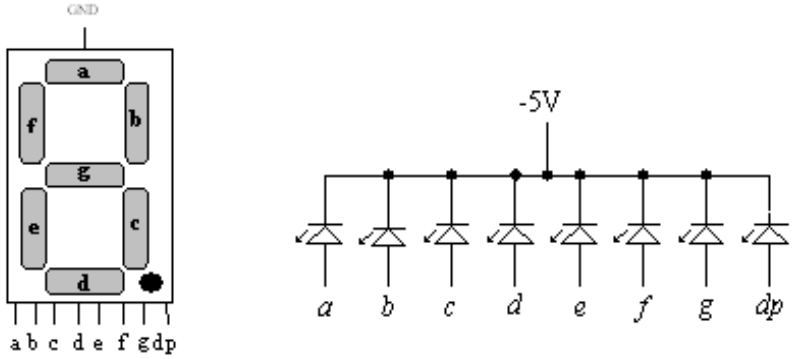
In common–anode, the anode of all of the LEDs are tied together to positive of the power supply (V_{cc}) as shown



Common–anode 7–segment LED display.

Common Cathode

- In common–cathode, the cathode of all of the LEDs are tied together to ground as shown.



Common–cathode 7–segment LED display.

Combinational Logic Circuit Implementation using a Decoder

- Any combinational logic circuit with n inputs and m outputs can be implemented with an n-to-2n-line decoder and m OR gates.
- Procedure:
  - Express the given Boolean function in sum of min-terms
  - Choose a decoder to generate all the min-terms of the input variables.
  - Select the inputs to each OR gate from the decoder outputs according to the list of min-term for each function.

Combinational Logic Circuit Implementation using a Decoder - An example (1)

Example: From the truth table of the full adder, design the circuit with a decoder

x	y	Z	C	S
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

- the functions can be expressed in sum of min-terms.

  $S(x,y,z) = \Sigma m(1,2,4,7)$

  $C(x,y,z) = \Sigma m(3,5,6,7)$

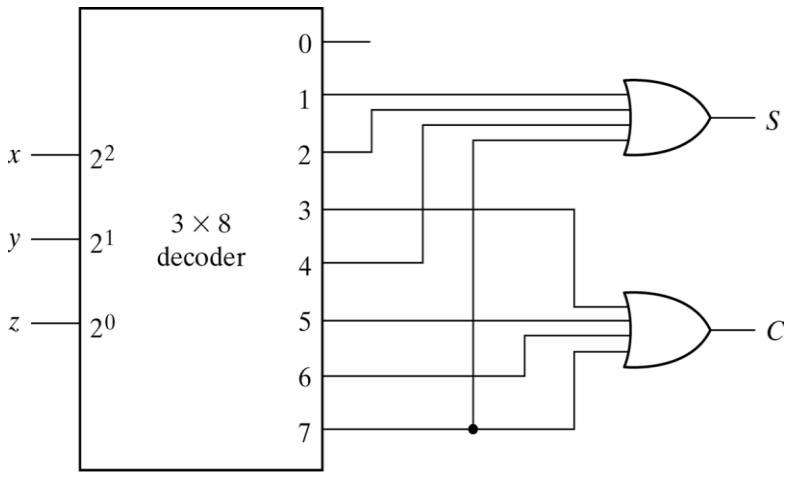
  where Σ indicates sum, m indicates min-term and the number in brackets indicate the decimal equivalent

Combinational Logic Circuit Implementation using a Decoder - An example (2)

Since there are three inputs and a total of eight min-terms, we need a 3-to-8 line decoder.

- The decoder generates the eight min-terms for x,y,z
- The OR gate for output S forms the logical sum of min-terms 1,2,4, and 7.
- The OR gates for output C forms the logical sum of min-terms 3,5,6, and 7

Combinational Logic Circuit Implementation using a Decoder - example (3)



Implementation of a Full Adder with a Decoder

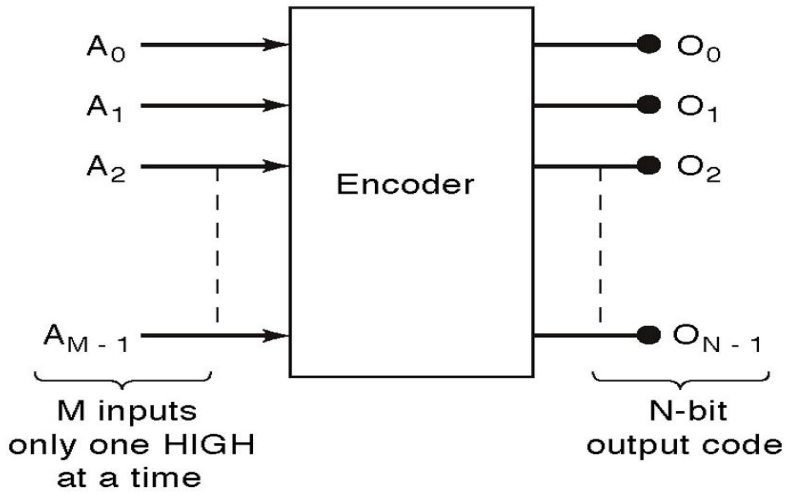
Encoders

- An encoder is a combinational logic circuit that essentially performs a "reverse" of decoder functions.
- An encoder accepts an active level on one of its inputs, representing digit, such as a decimal or octal digits, and converts it to a coded output such as BCD or binary.
- Encoders can also be devised to encode various symbols and alphabetic characters.
- The process of converting from familiar symbols or numbers to a coded format is called ***encoding***.

Encoders

- Most decoders accept an input code and produce a HIGH(or a LOW) at one and only one output line.
- In other words, a decoder identifies, recognizes, or detects a particular code.
- The opposite of this decoding process is called encoding and is performed by a logic circuit called an encoder.
- An encoder has a number of input lines, only one of which input is activated at a given time and produces an N-bit output code,depending on which input is activated.

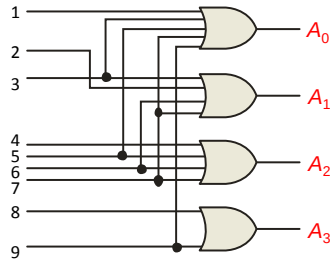
General encoder diagram



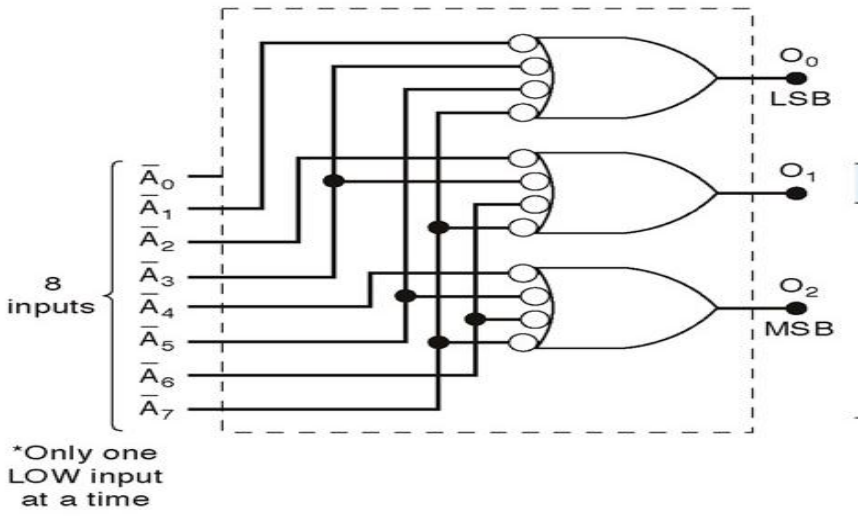
Encoders

An **encoder** accepts an active logic level on one of its inputs and converts it to a coded output, such as BCD or binary.

- The decimal to BCD is an encoder with an input for each of the ten decimal digits and four outputs that represent the BCD code for the active digit.
- The basic logic diagram is shown. There is no zero input because the outputs are all LOW when the input is zero.



Logic circuit for octal-to binary encoder [8-line-3-line]

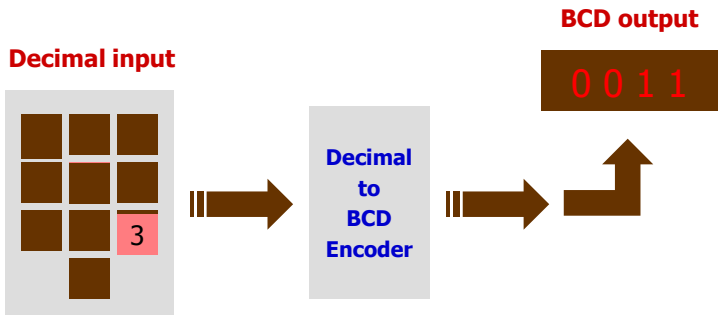


Truth table for octal-to binary encoder [8-line- 3-line]

Inputs								Outputs		
$\bar{A}_0$	$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	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

A low at any single input will produce the output binary code corresponding to that input. For instance , a low at A_3' will produce O_2 =0, O_1=1 and O_0 =1, which is binary code for 3. A_0' is not connected to the logic gates because the encoder outputs always be normally at 0000 when none of the inputs is LOW

Encoders

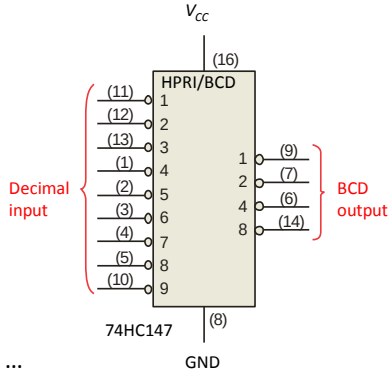


- **Encoders are available in IC form.**
- **This encoder translates from decimal input to BCD output.**

Encoders

The 74HC147 is an example of an IC encoder. It has ten active-LOW inputs and converts the active input to an active-LOW BCD output.

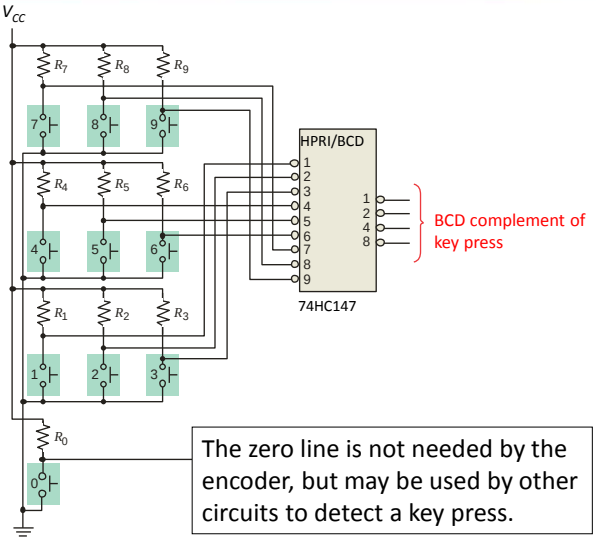
This device offers additional flexibility in that it is a **priority encoder**. This means that if more than one input is active, the one with the highest order decimal digit will be active.



The next slide shows an application …

Encoders

Keyboard encoder

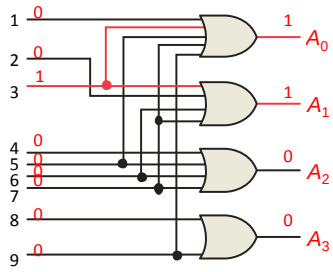


The zero line is not needed by the encoder, but may be used by other circuits to detect a key press.

Encoders

Example Show how the decimal-to-BCD encoder converts the decimal number 3 into a BCD 0011.

Solution The top two OR gates have ones as indicated with the red lines. Thus the output is 0111.



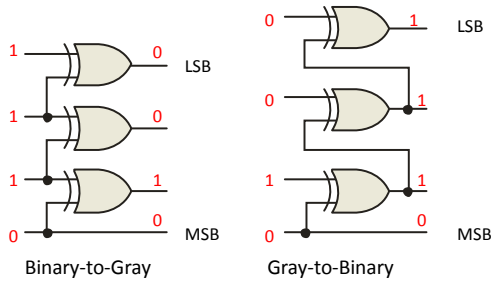
Code Converters

There are various code converters that change one code to another. Two examples are the four bit binary-to-Gray converter and the Gray-to-binary converter.

Example

Show the conversion of binary 0111 to Gray and back.

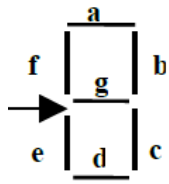
Solution



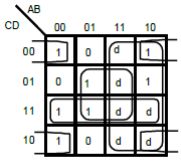
Code Converters

Design a BCD to 7 Segment Display Code Converter

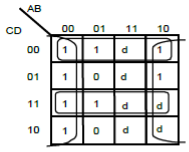
Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1



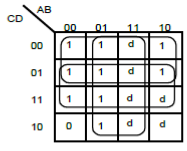
Code Converters



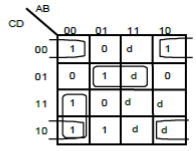
a=A+CD+BD+B'D'



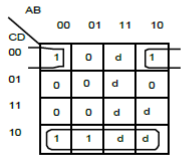
b=B'+CD+C'D'



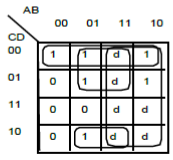
c=A'C'+B'C'+A'D+A'B



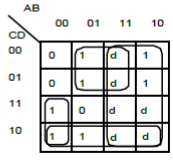
d=C'D'+BC'D+A'B'C+AB'



e=CD'+B'C'D'



f=BCD'+A+BC'+C'D'



g=CD'+BC'+A'B'C+A

Code Converters

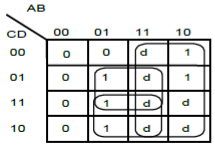
Example: Design a code converter to convert BCD numbers to Excess-three code.

Solution:

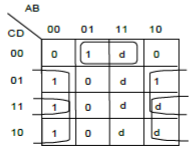
- i- Create the truth table.
- ii- Creat the Karnaugh Map for the truth table.
- iii- Group Karnaugh map and derive the boolean functions
- iv- Draw the circuit for the functions

Code Converters

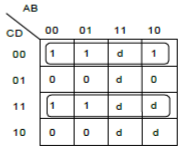
Decimal	A	B	C	D	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0



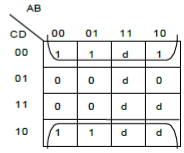
$Q_1=A+BC+BD$



$Q_2=BC'D'+B'D+B'C$

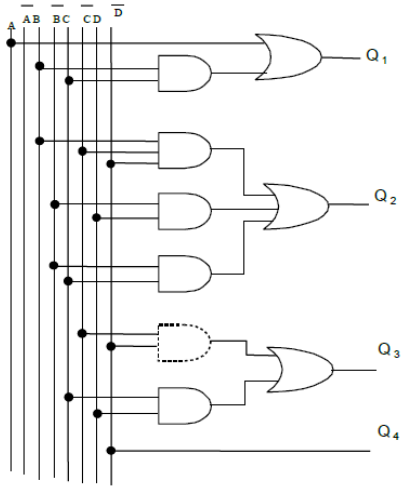


$Q_3=C'D'+CD$

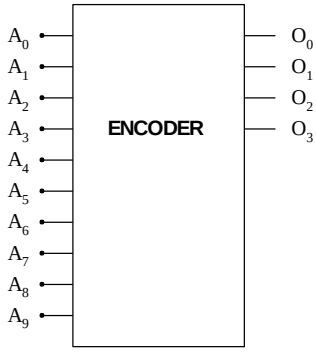


$Q_4=D'$

Code Converters

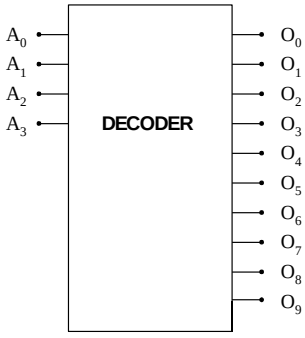


TEST



INPUT	O3	O2	O1	O0
A1=1				
A4=1				
A6=1				
A8=1				

TEST



A3	A2	A1	A0	OUTPUT
0	0	0	0	
0	1	0	1	
0	1	1	1	
1	0	0	1	