Digital Logic and Circuit
Paper Code: CS-102

Outline

Combinational circuit

- **Code conversion**
- **Encoders & decoders**

Code Converter

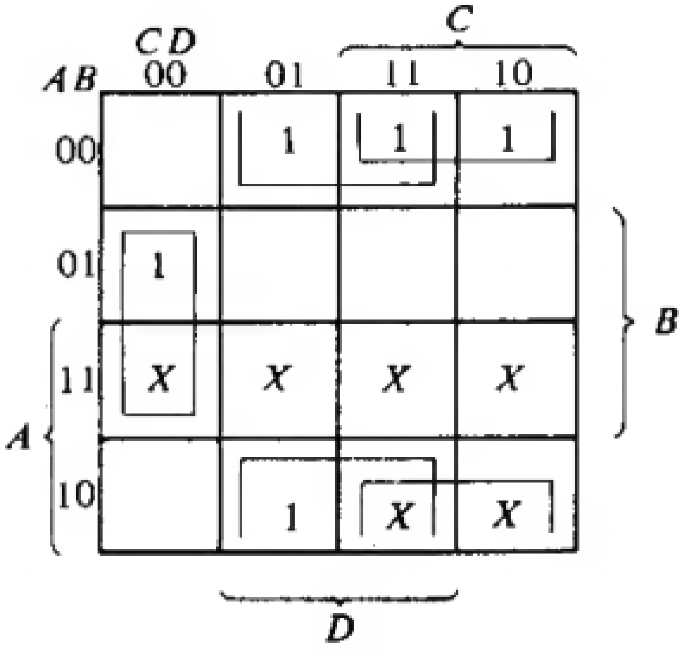
- The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital systems.
- It is sometimes necessary to use the output of one system as the input to another.
- A conversion circuit must be inserted between the two systems if each uses different codes for the same information.
- Thus, a code converter is a circuit that makes the two systems compatible even though each uses a different binary code.
- To convert from binary code A to binary code B, the input lines must supply the bit combination of elements as specified by code A and the output lines must generate the corresponding bit combination of code B.
- A combinational circuit performs this transformation by means of logic gates.

Conversion from the BCD to the excess-3 code.

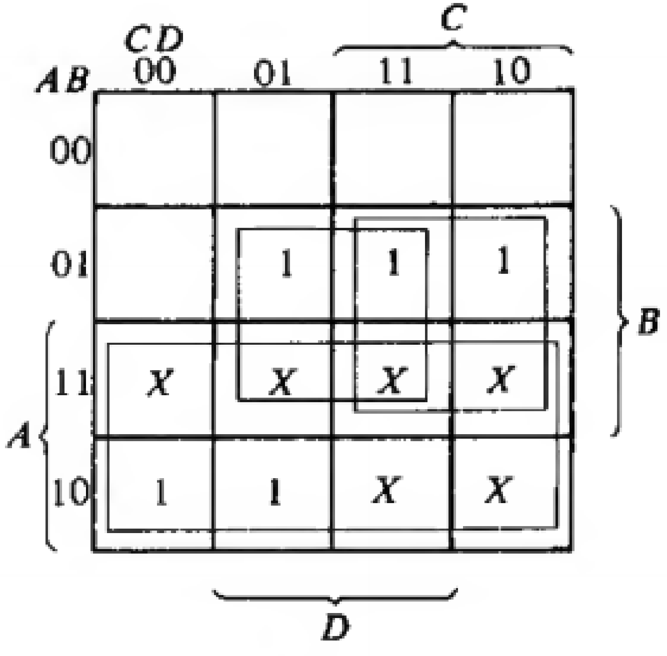
- The bit combinations for the BCD and excess-3 codes are of 4 bit. Since each code uses four bits to represent a decimal digit, there must be four input variables and four output variables.
- Let us designate the four input binary variables by the symbols A, B , C, and D , and the four output variables by w, x, y, and z.
- The truth table relating the input and output variables is shown in Table.
- Note that four binary variables may have 16 bit combinations, only 10 of which are listed in the truth table.
- The six bit combinations not listed for the input variables are don’t-care combinations.
- Since they will never occur, we are at liberty to assign to the output variables either a 1 or a 0, whichever gives a simpler circuit.

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



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

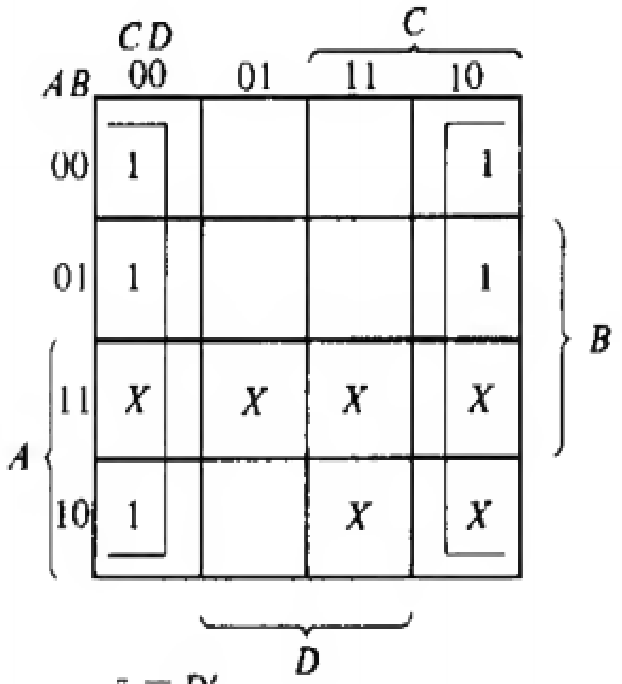


$w = A + BC + BD$

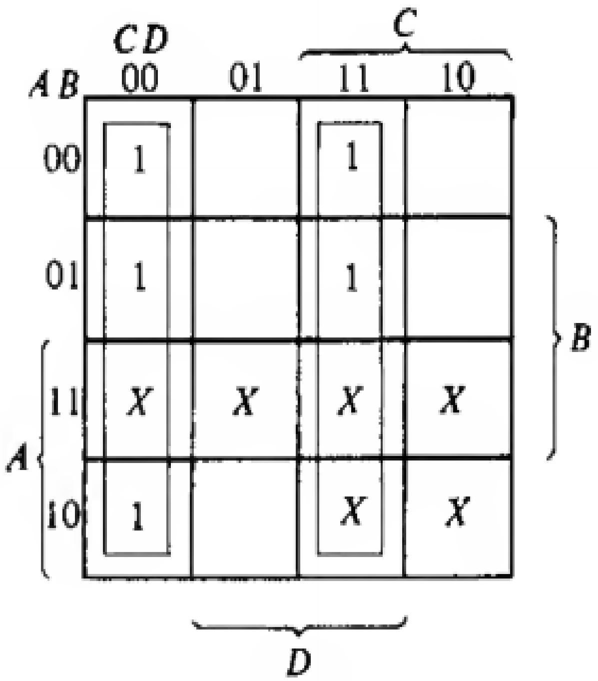
Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Maps for a BCD-to-excess-3-code converter



$z = D'$



$y = CD + C'D'$

Maps for a BCD-to-excess-3-code converter

Truth Table for Code-Conversion Example

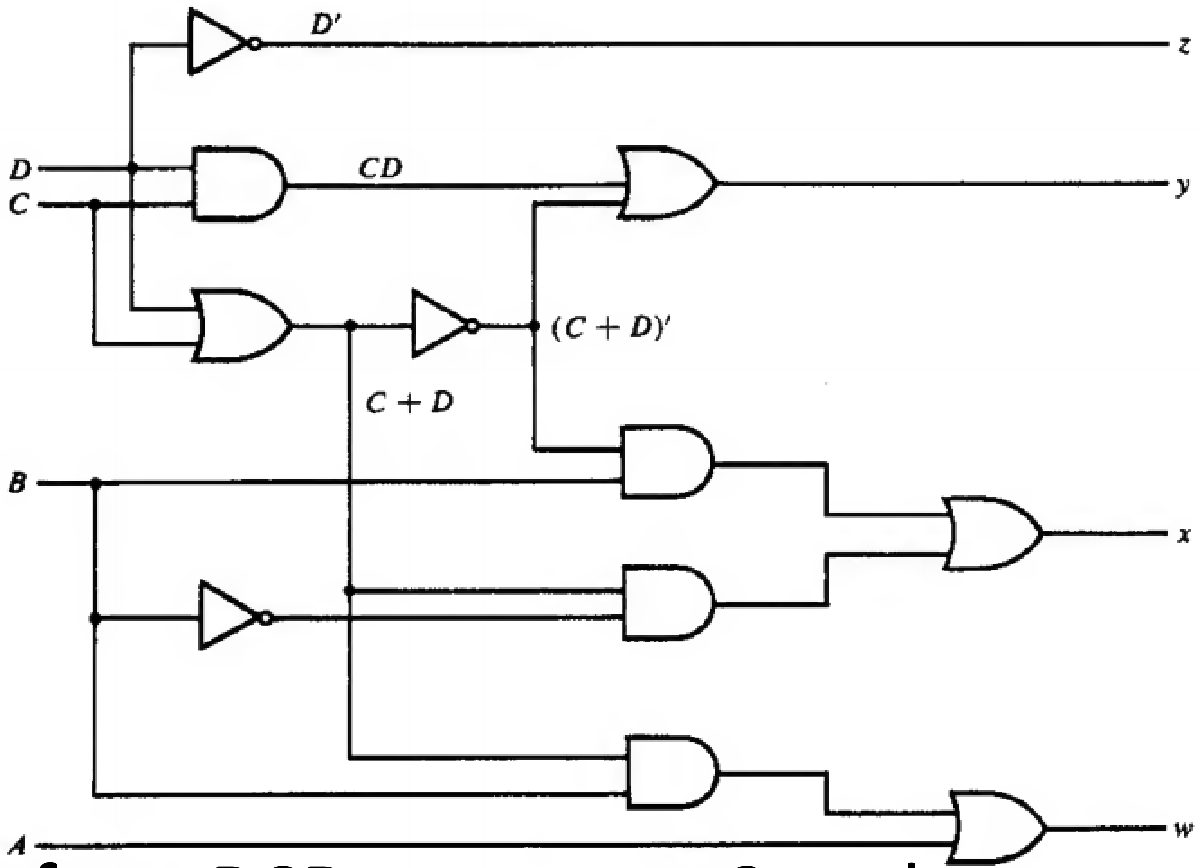
Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

z = D'

y=CD+C'D'= CD + (C+D)'

x = B'C + B'D + BC'D' =B' (C +D) + BC'D' = B' (C + D) + B(C + D) '

w= A +BC +BD = A + B(C +D)



Logic diagram for a BCD-to-excess-3-code converter

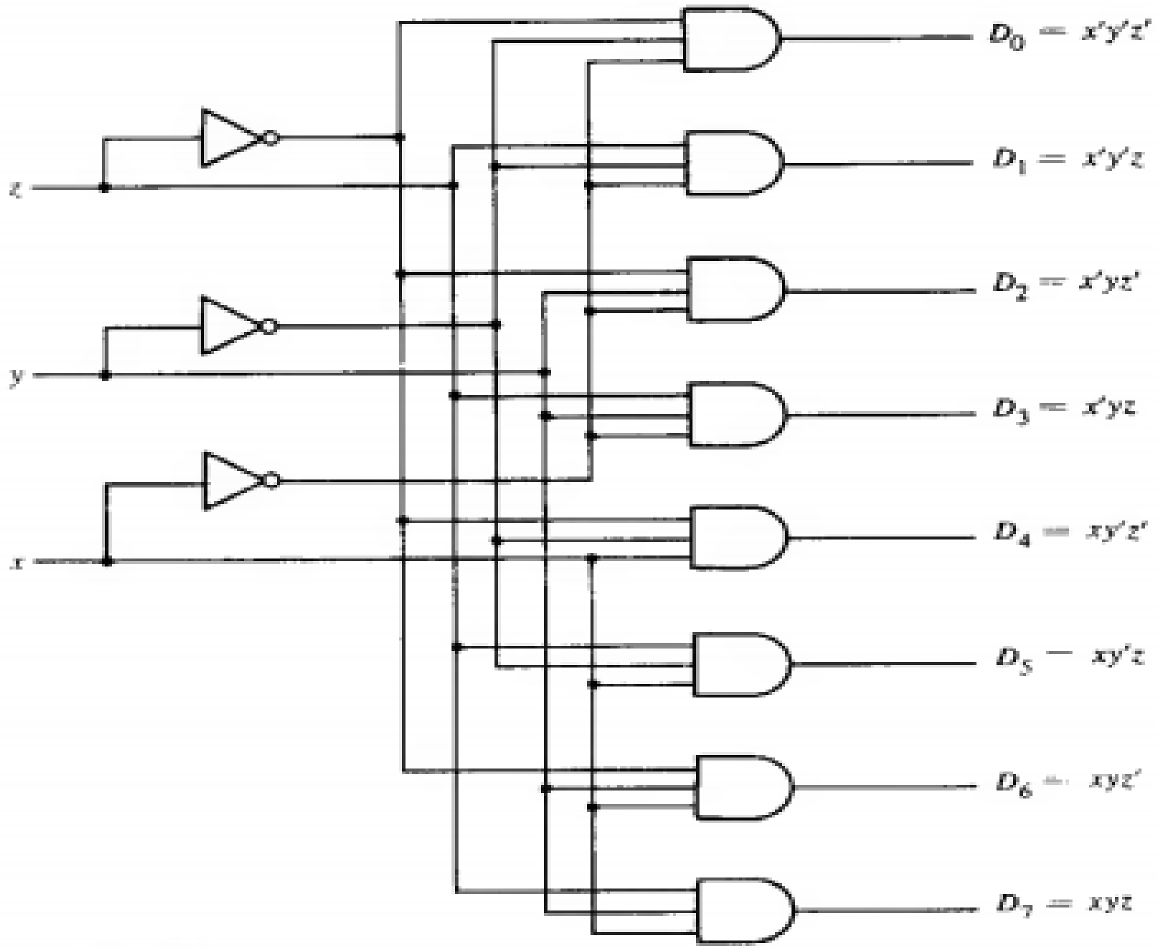
DECODERS

- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.
- If the n-bit decoded information has unused or don't-care combinations, the decoder output will have fewer than 2^n outputs.
- The decoders presented here are called n-to-m-line decoders, where $m \leq 2^n$.
- Their purpose is to generate the 2^n (or fewer) minterms of n input variables.
- The name decoder is also used in conjunction with some code converters such as a BCD-to-seven segment decoder.

- As an example, consider the 3-to-8-line decoder circuit.
- The three inputs are decoded into eight outputs, each output representing one of the minterms of the 3-input variables.
- The three inverters provide the complement of the inputs, and each one of the eight AND gates generates one of the minterms.
- A particular application of this decoder would be a binary-to-octal conversion.
- The input variables may represent a binary number, and the outputs will then represent the eight digits in the octal number system.

Truth Table of a 3-to-8-LIne Decoder

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



z
y
x
D0 = x'y'z'
D1 = x'y'z
D2 = x'yz'
D3 = x'yz
D4 = xy'z'
D5 = xy'z
D6 = xyz'
D7 = xyz

Encoder

- An encoder has 2^n (or fewer) input lines and n output lines.
- The output lines generate the binary code corresponding to the input value.
- An example of an encoder is the octal to-binary encoder whose truth table is given in Table .
- It has eight inputs, one for each of the octal digits, and three outputs that generate the corresponding binary number.
- It is assumed that only one input has a value of 1 at any given time; otherwise the circuit has no meaning.

- The encoder can be implemented with OR gates whose inputs are determined directly from the truth table.
- Output z is equal to 1 when the input octal digit is 1 or 3 or 5 or 7.
- Output y is 1 for octal digits 2, 3, 6, or 7, and
- output *x* is 1 for digits 4, 5, 6, or 7. These conditions can be expressed by the following output Boolean functions:

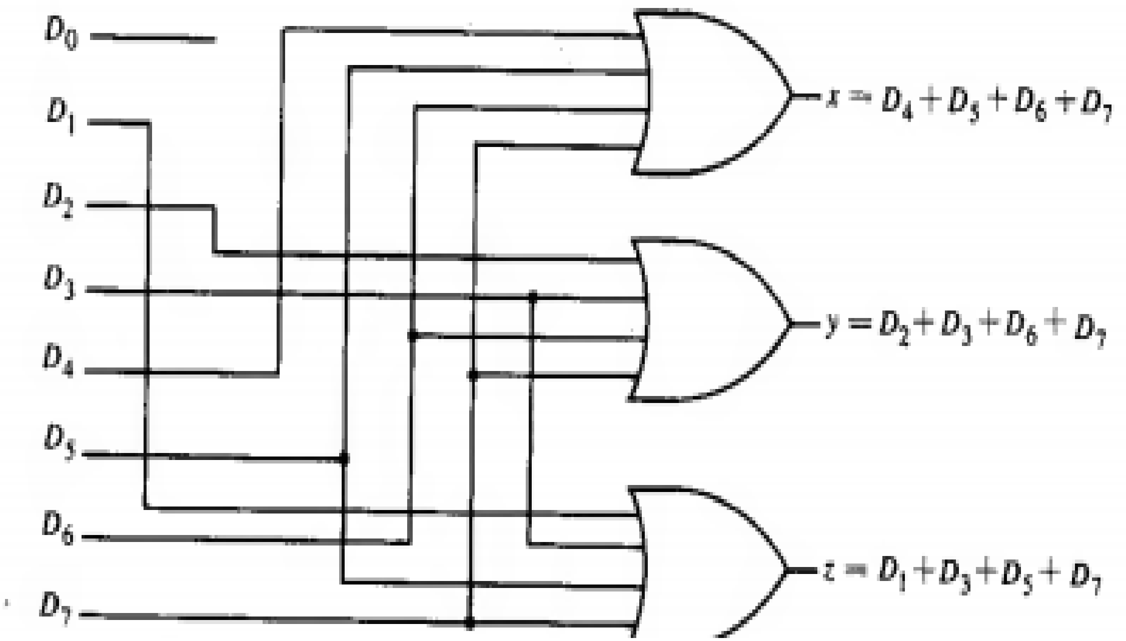
$$z = D_1 + D_3 + D_5 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

The encoder is implemented with three OR gates



Octal-to-binary encoder

Multiplexer

- A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.

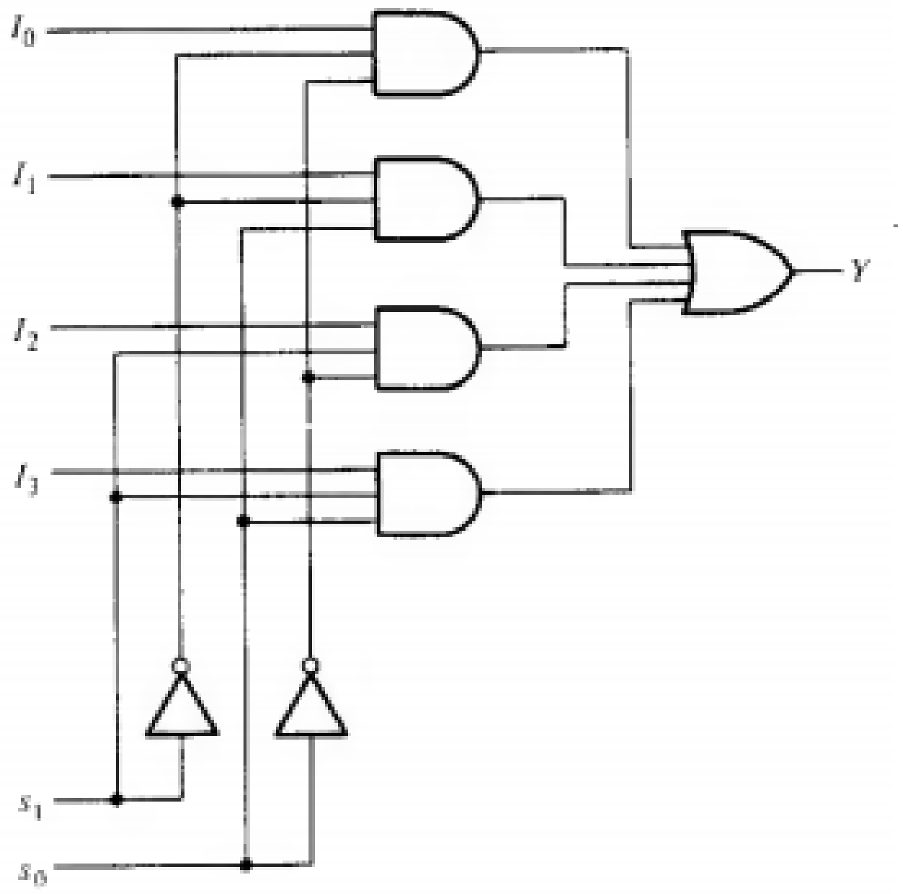
- The selection of a particular input line is controlled by a set of selection lines.

- Normally, there are 2" input lines and n selection lines whose bit combinations determine which input is selected.

- A 4-to- 1 -line multiplexer is shown in the Figure.

- Each of the four input lines, I_0 to I_3, is applied to one input of an AND gate.

- Selection lines S_1 and S_0 are decoded to select a particular AND gate.



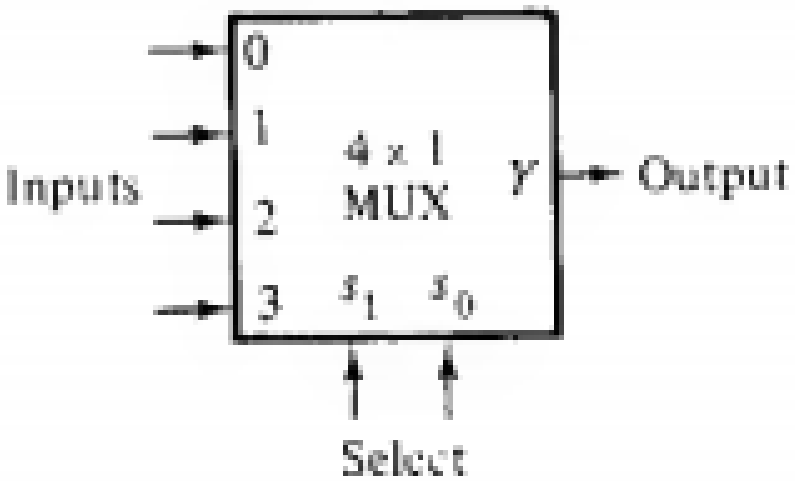
(a) Logic diagram

- The function table lists the input-to-output path for each possible bit combination of the selection lines.

s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

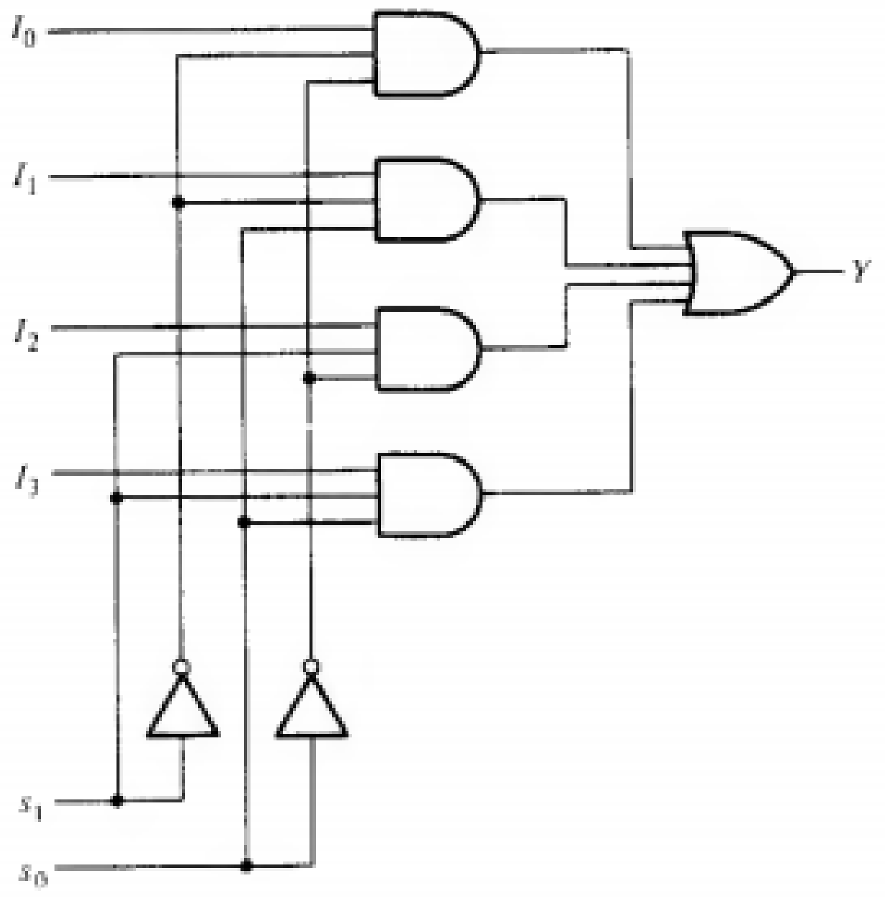
(b) Function table

- When this MSI function is used in the design of a digital system, it is represented in block diagram form, as shown in Figure (c)



(c) Block diagram

- To demonstrate the circuit operation, consider the case when S_1S_0 = 10.
- The AND gate associated with input I_2 has two of its inputs equal to 1 and the third input connected to I_2 .
- The other three AND gates have at least one input equal to 0, which makes their outputs equal to 0.
- The OR gate output is now equal to the value of I_2 , thus providing a path from the selected input to the output.
- A multiplexer is also called a data selector, since it selects one of many inputs and steers the binary information to the output line. A multiplexer is often abbreviated as MUX.



(a) Logic diagram

Suggested Reading

- M. Morris Mano, Digital Logic and Computer Design, PHI.

Thank you