Digital Logic Design
ENEE 244-010x

Lecture 14

Announcements

- Homework 6 due today

Agenda

- Last time:
  - Binary Adders and Subtracters (5.1, 5.1.1)
  - Carry Lookahead Adders (5.1.2, 5.1.3)

- This time:
  - Decimal Adders (5.2)
  - Comparators (5.3)
  - Decoders (5.4)
  - Encoders (5.5)
  - Multiplexers (5.6)
- After introducing all the MSI components, we will go back and talk about how to use Decoders and Multiplexers for Logic Design.

Decimal Adders

8421 weighted coding scheme or BCD Code

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

Forbidden codes: 1010, 1011, 1100, 1101, 1110, 1111

Decimal Adder

- Inputs: $A_3A_2A_1A_0$, $B_3B_2B_1B_0$, C_{in} from previous decade.
- Output: C_{out} (carry to next decade), $Z_3Z_2Z_1Z_0$.
- Idea: Perform regular binary addition and then apply a corrective procedure.

Comparing Binary and BCD Sums

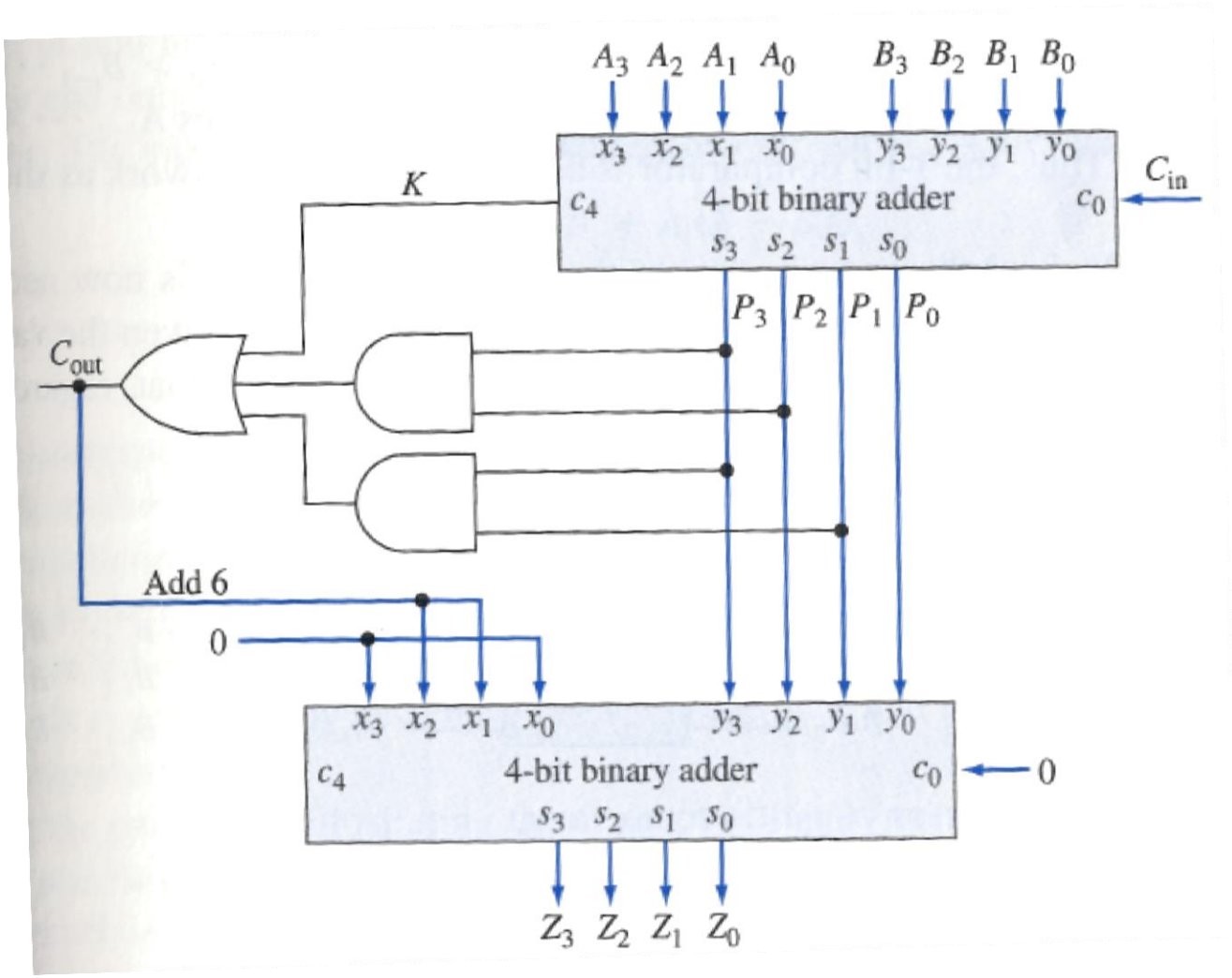
Decimal Sum	K	P_3	P_2	P_1	P_0	C_{out}	Z_3	Z_2	Z_1	Z_0
0-9					--------Same----------					
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

C_{out} is set to 0

Decimal Adder

- No correction needed when the decimal sum is between 0-9.
- Must apply a correction when the sum is between 10-19.
- Case 1:
  - 16-19: K is set to 1. Add binary quantity 0110 to $P_3P_2P_1P_0$.
  - 10-15: $KP_3P_2P_1P_0$ are set to 01010, 01011, . . , 01111. Need to add 6. Use a K-map to obtain a Boolean expression to detect these six binary combinations.

A single-decade BCD Adder



Comparators

- Compare the magnitude of two binary numbers for the purpose of establishing whether one is greater than, equal to, or less than the other.
- A comparator makes use of a cascade connection of identical subnetworks similar to the case of the parallel adder.

Comparators

- Consider two n-bit binary numbers:

$$A = A_{n-1} \cdots A_i A_{i-1} \cdots A_1 A_0$$
$$B = B_{n-1} \cdots B_i B_{i-1} \cdots B_1 B_0$$

- Assume A_i, B_i are entering the subnetwork and that the binary numbers are analyzed from right to left.
- Subnetwork is called a 1-bit comparator.

Comparators

- 3 conditions describing the relative magnitudes of $A_{i-1} \cdots A_1 A_0$, $B_{i-1} \cdots B_1 B_0$
- $G_i = 1$ denotes $A_{i-1} \cdots A_1 A_0 > B_{i-1} \cdots B_1 B_0$
- $E_i = 1$ denotes $A_{i-1} \cdots A_1 A_0 = B_{i-1} \cdots B_1 B_0$
- $L_i = 1$ denotes $A_{i-1} \cdots A_1 A_0 < B_{i-1} \cdots B_1 B_0$
- 1-bit comparator is a 5-input 3-output network

Comparators

- Rules:
  - If $A_i = 0, B_i = 1$ then $L_i = 1$
  - If $A_i = 1, B_i = 0$ then $G_i = 1$
  - If $A_i = B_i$ and $L_{i-1} = 1$ then $L_i = 1$
  - If $A_i = B_i$ and $G_{i-1} = 1$ then $G_i = 1$
  - If $A_i = B_i$ and $E_{i-1} = 1$ then $E_i = 1$
- Can use this to construct a truth table.

Comparators

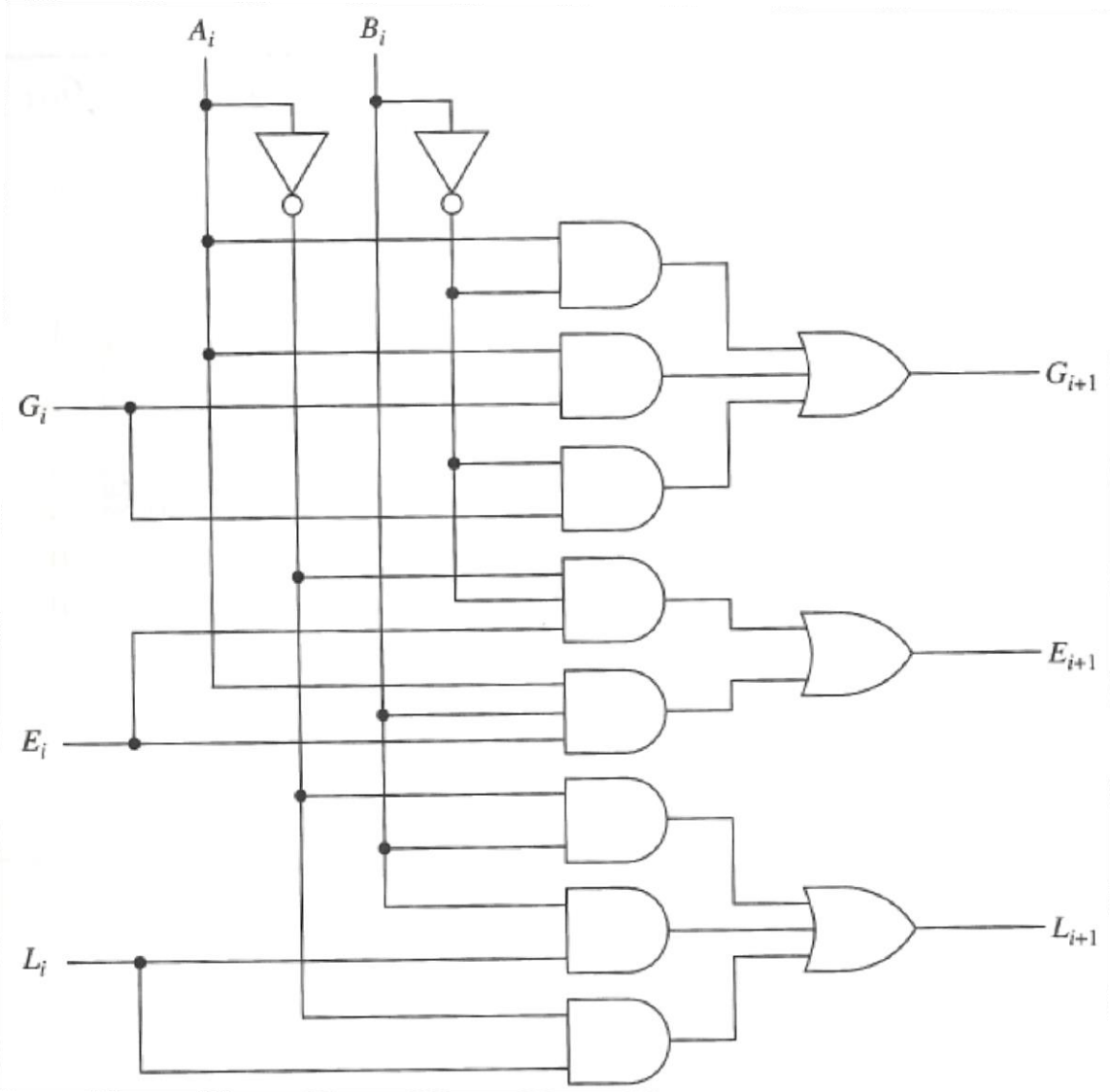
- Minimal Sum Boolean Expressions:

$$G_{i+1} = A_i\overline{B}_i + A_iG_i + \overline{B}_iG_i$$

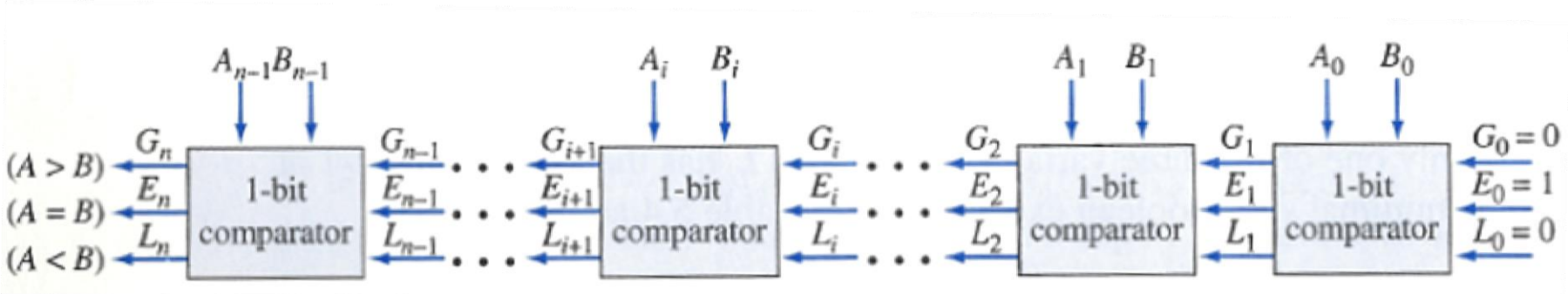
$$E_{i+1} = \overline{A}_i\overline{B}_iE_i + A_iB_iE_i$$

$$L_{i+1} = \overline{A}_iB_i + B_iL_i + \overline{A}_iL_i$$

Comparators



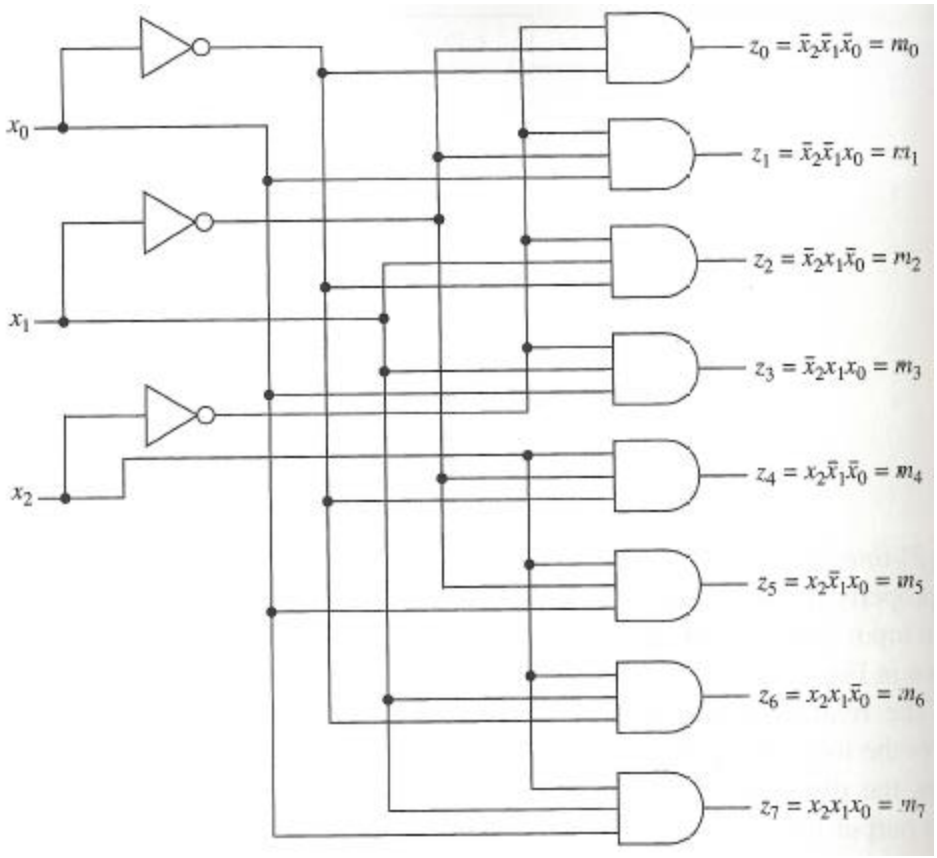
Comparators



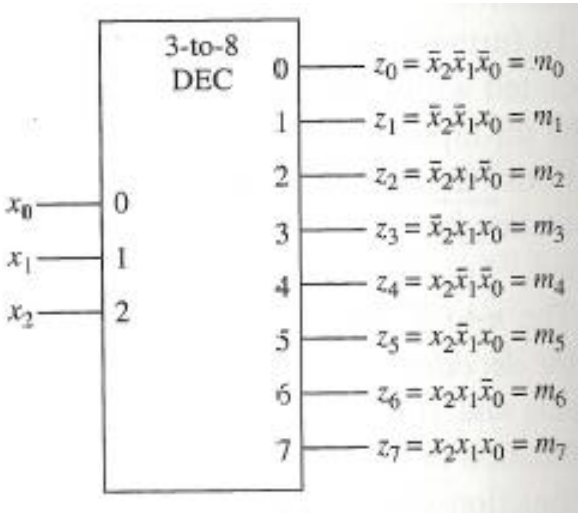
Decoder

- Digital information represented in some binary form must be converted into some alternate binary form.
- n to 2^n-line decoder.
- Only one of the 2^n output lines responds, with a logic-1, to a given input combination of values on its n-input lines.

Realization



Logic Diagram



Symbol

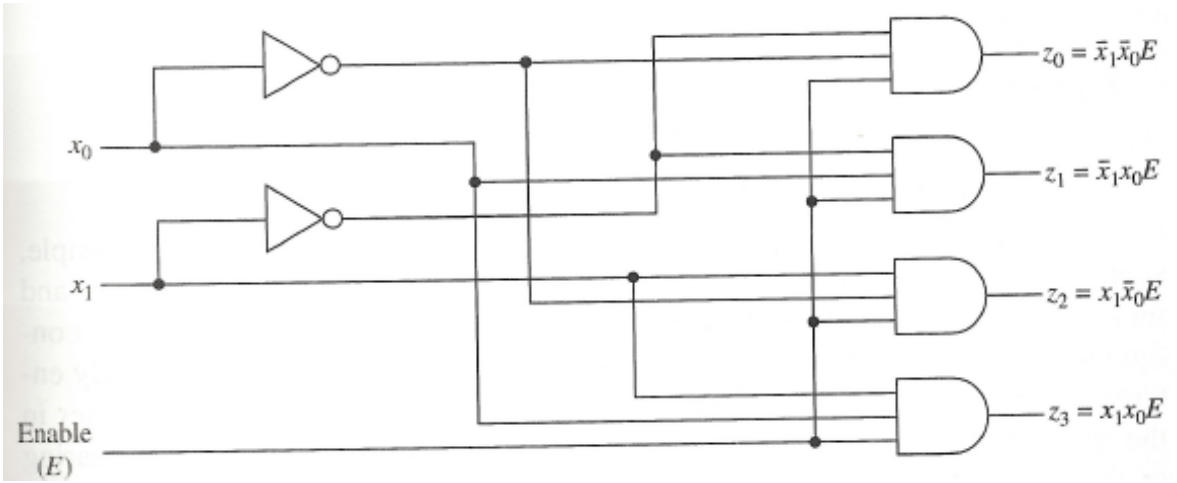
Inputs			Outputs							
x_2	x_1	x_0	z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Truth Table

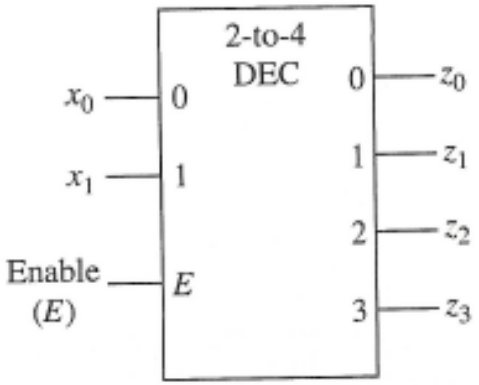
Decoder

- Input combinations can be regarded as binary numbers with the consequences that the j-th output line is at logic-1 for j = 0, 1, . . , 7 only when input combination j is applied.

Decoders with an Enable Input



Logic Diagram



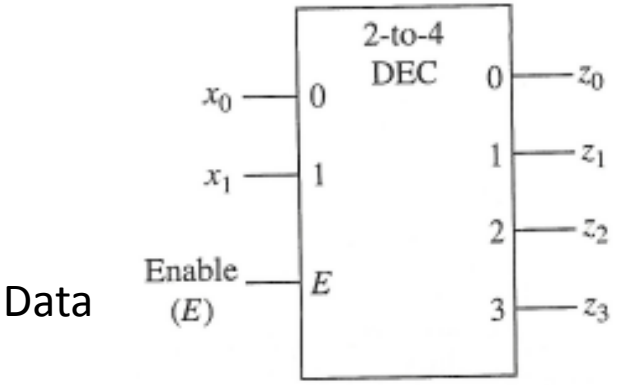
Symbol

Inputs			Outputs			
E	x_1	x_0	z_0	z_1	z_2	z_3
0	×	×	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Truth Table

Decoders with enable inputs

- When disabled, all outputs of the decoder can either be at logic-0 or logic-1.
- Enable input provides the decoder with additional flexibility. Idea: data is applied to the enable input.
- Process is known as demultiplexing.



$\overline{x}_0\overline{x}_1E$

If $x_0 = 0, x_1 = 0$ then data appears on line z_0.

- Enable inputs are useful when constructing larger decoders from smaller decoders.

Constructing Larger Decoders

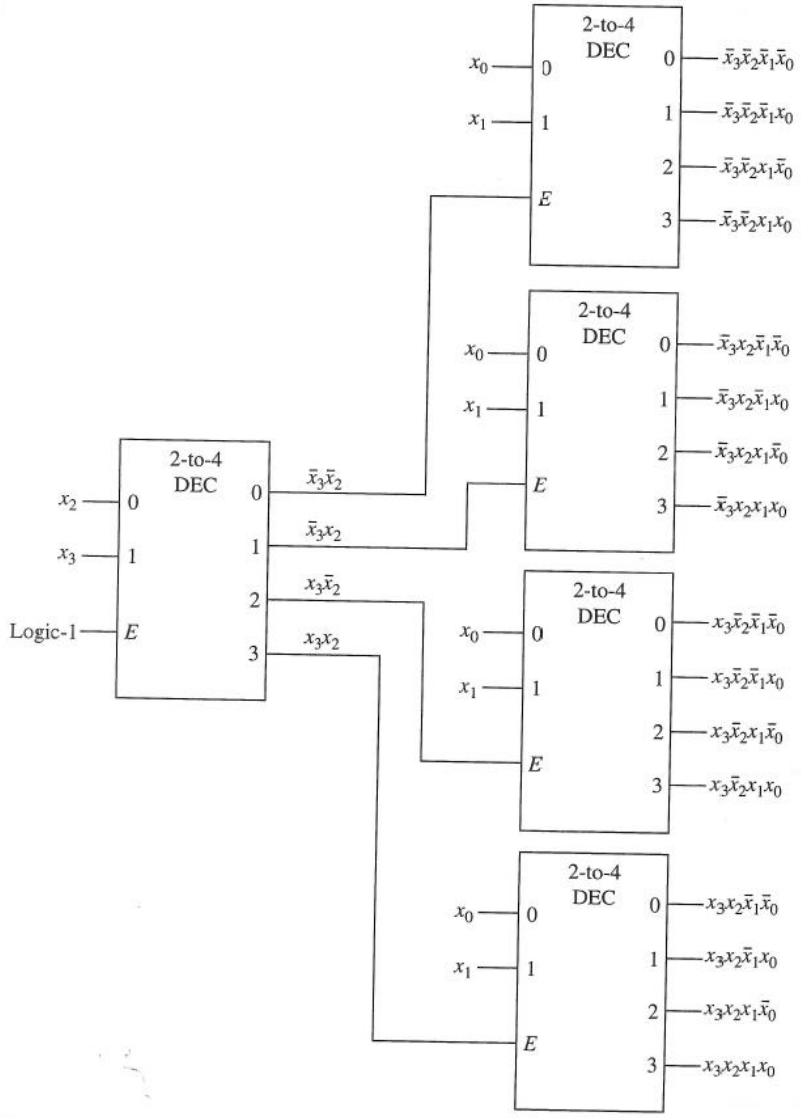


Figure 5.29 A 4-to-16-line decoder constructed from 2-to-4-line decoders.

Encoders

- Encoders provide for the conversion of binary information from one form to another.
- Encoders are essentially the inverse of decoders.
- 2^n-to-n-line encoder in which an assertive logic value on one of its 2^n-input lines causes the corresponding binary code to appear at the output lines.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_0	z_1	z_2
0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0

Encoders

- Equations for 8-to-3-line encoder:

$$z_0 = x_1 + x_3 + x_5 + x_7$$
$$z_1 = x_2 + x_3 + x_6 + x_7$$
$$z_2 = x_4 + x_5 + x_6 + x_7$$

- In general, the Boolean expression for the output z_i is the sum of each input x_j in which the binary representation of j has a 1 in the 2^i-bit position.

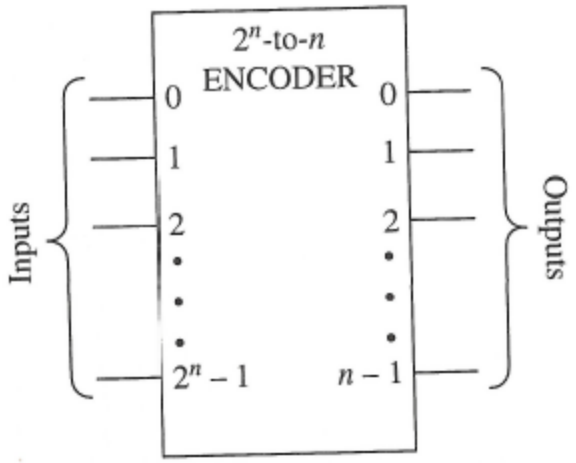


Figure 5.30 A 2^n-to-n-line encoder symbol.

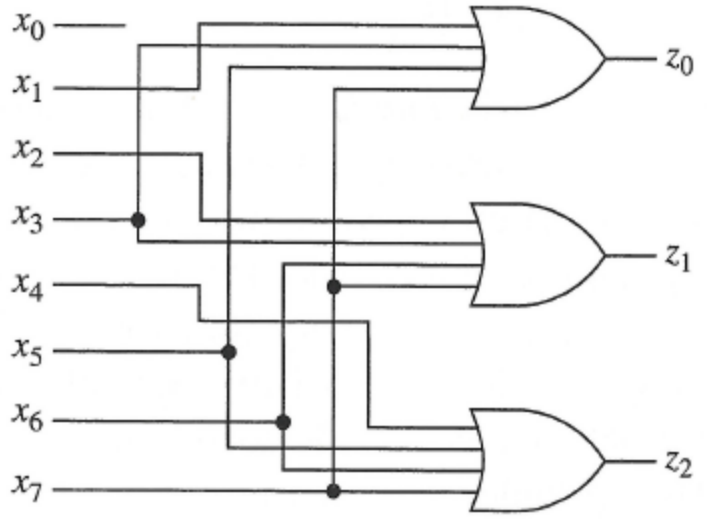


Figure 5.31 An 8-to-3-line encoder.

Priority Encoder

- The assumption that at most a single input to the encoder is asserted at any time is significant in its operation.
  - Example: Both $x_3\ (11)$ and $x_5\ (101)$ are asserted. What is the output?
  - $111\ (x_7)$
- Priority Encoder:
  - A priority scheme is assigned to the input lines so that whenever more than one input line is asserted at any time, the output is determined by the input line having the highest priority.

Priority Encoder

Table 5.5 Compressed truth table for an 8-to-3 line priority encoder

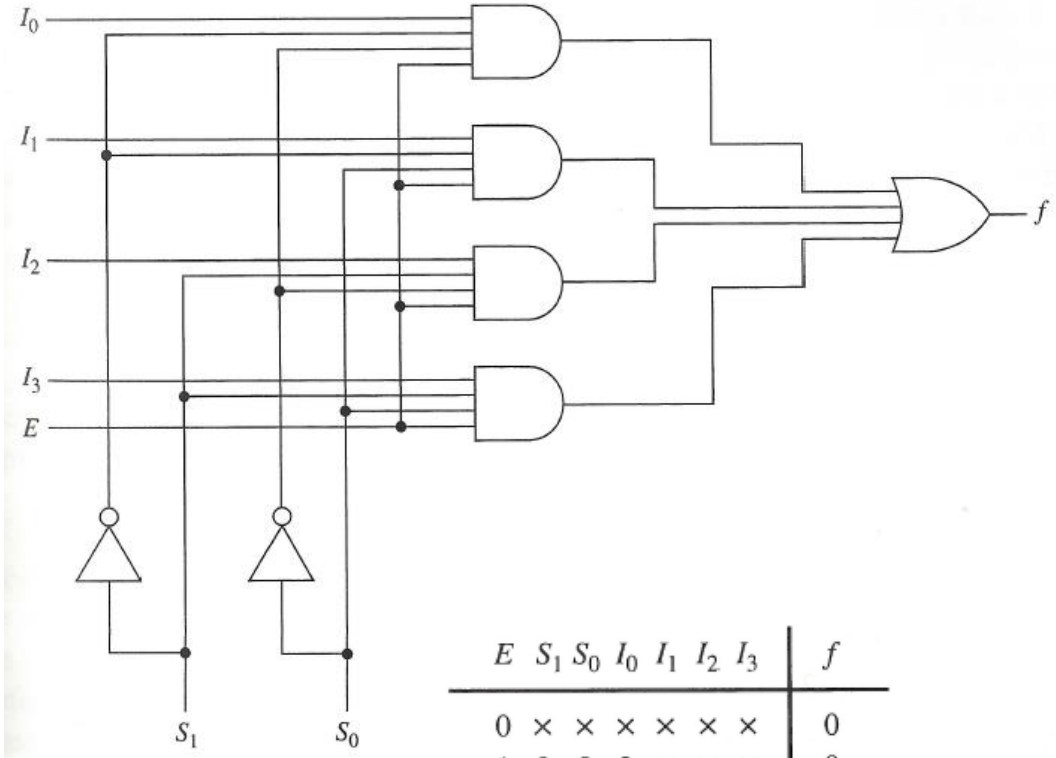
Inputs								Outputs			
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_2	z_1	z_0	Valid
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
×	1	0	0	0	0	0	0	0	0	1	1
×	×	1	0	0	0	0	0	0	1	0	1
×	×	×	1	0	0	0	0	0	1	1	1
×	×	×	×	1	0	0	0	1	0	0	1
×	×	×	×	×	1	0	0	1	0	1	1
×	×	×	×	×	×	1	0	1	1	0	1
×	×	×	×	×	×	×	1	1	1	1	1

The output is determined by the asserted input having the highest index.
x_i has higher priority than x_j if $i > j$.
"Valid" indicates that at least one input line is asserted.
This distinguishes the situation that no input line is asserted from when the x_0 input line is asserted, since in both cases $z_2 z_1 z_0 = 000$.

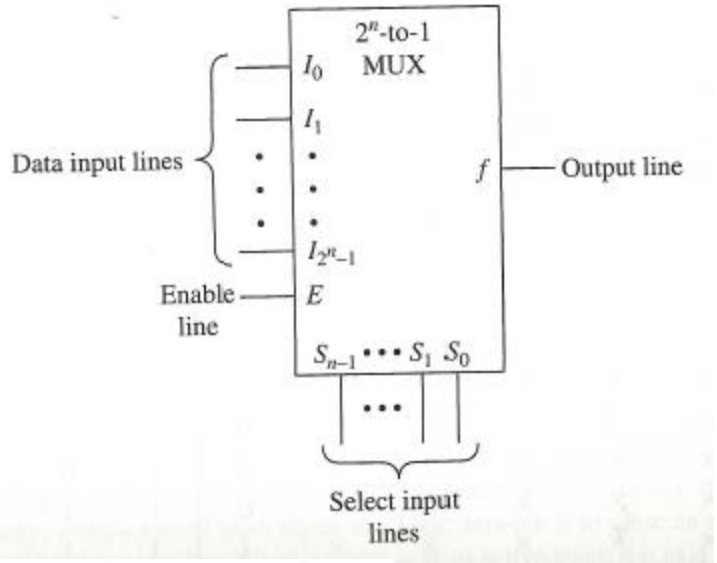
Multiplexer

- Also called data selectors.
- Basic function: select one of its 2^n data input lines and place the corresponding information onto a single output line.
- n input bits needed to specify which input line is to be selected.
  - Place binary code for a desired data input line onto its n select input lines.

Realization of 4-to-1 line multiplexer

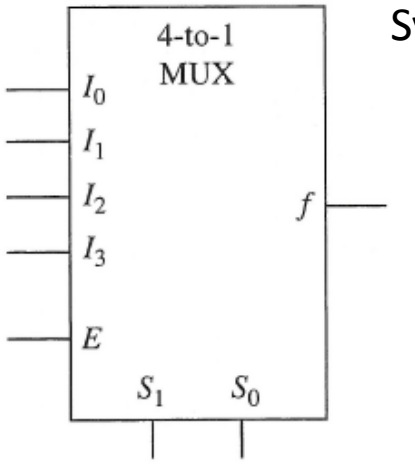


Logic Diagram



E	S_1	S_0	I_0	I_1	I_2	I_3	f
0	×	×	×	×	×	×	0
1	0	0	0	×	×	×	0
1	0	0	1	×	×	×	1
1	0	1	×	0	×	×	0
1	0	1	×	1	×	×	1
1	1	0	×	×	0	×	0
1	1	0	×	×	1	×	1
1	1	1	×	×	×	0	0
1	1	1	×	×	×	1	1

Truth Table



Symbol

Realization of 4-to-1 line multiplexer

- Alternate description:

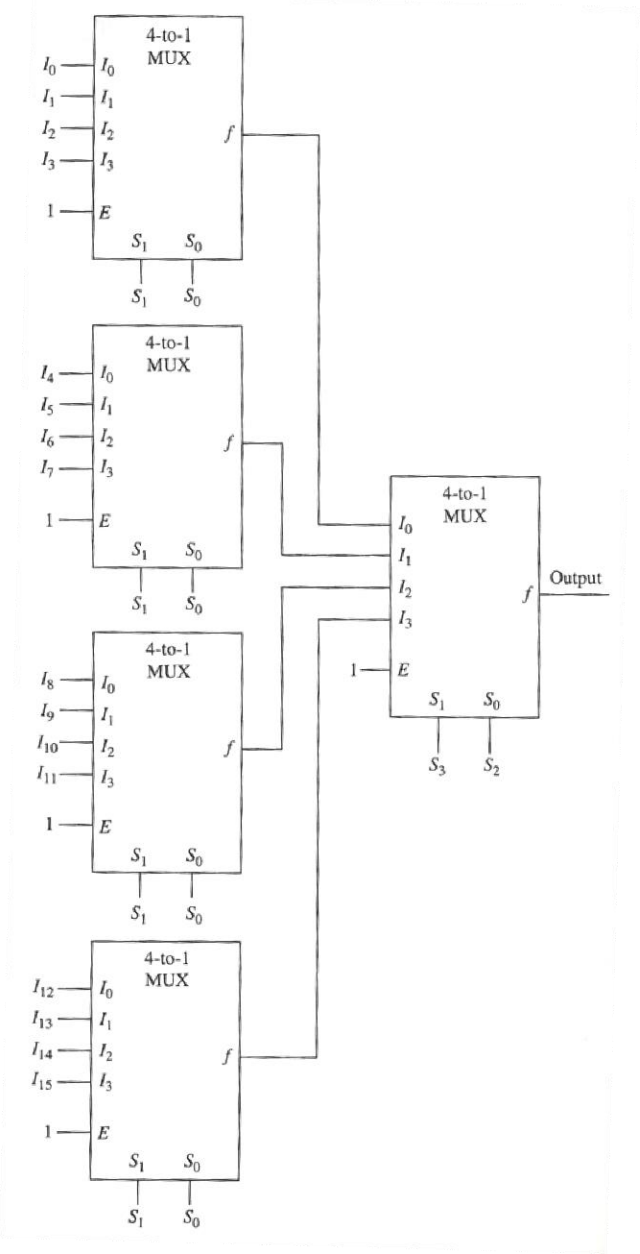
Table 5.6 Function table for a 4-to-1-line multiplexer

E	S_1	S_0	f
0	×	×	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

- Algebraic description of multiplexer:

$$f = \left(I_0\overline{S}_1\overline{S}_0 + I_1\overline{S}_1S_0 + I_2S_1\overline{S}_0 + I_3S_1S_2\right)E$$

Building a Large Multiplexer



Multiplexers

- One of the primary applications of multiplexers is to provide for the transmission of information from several sources over a single path.
- This process is known as multiplexing.
- Demultiplexer = decoder with an enable input.

Multiplexer/Demultiplexer for information transmission

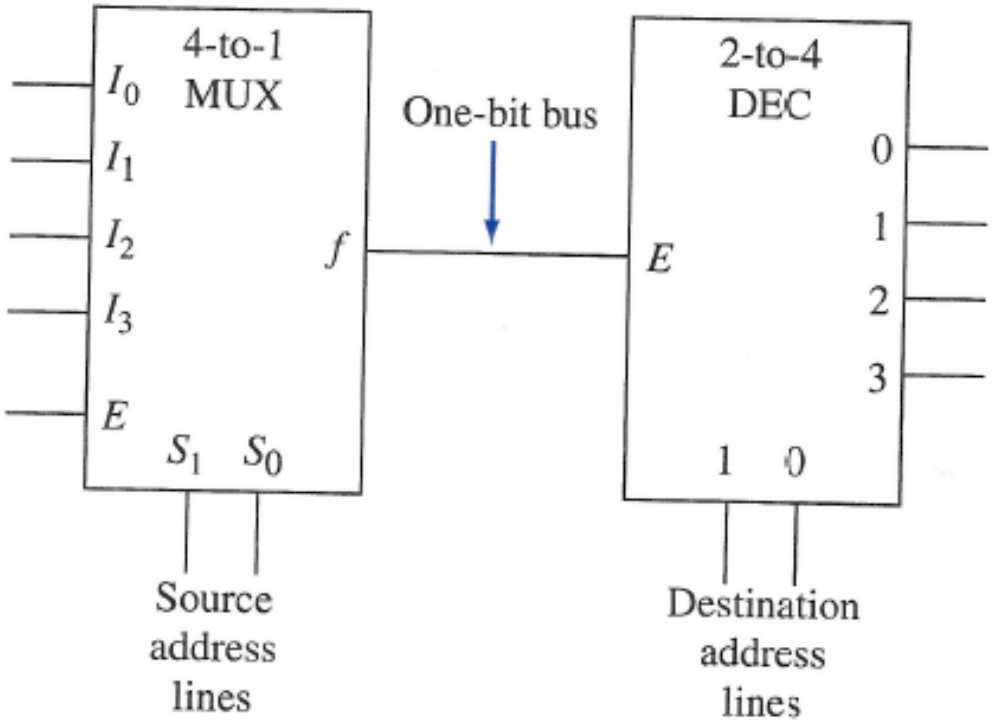


Figure 5.35 A multiplexer/demultiplexer arrangement for information transmission.