CAO: Lecture 5
Combinational Logic Blocks: Examples

Topics Covered

- Examples of Combinational circuits
- Decoder
- 2:4 Decoder
- 3:8 Decoder
- Combinational Circuit Design with Decoders
- Multiplexers

Examples of Combinational Circuits

- a) Decoders
- b) Encoders
- c) Multiplexers
- d) Demultiplexers

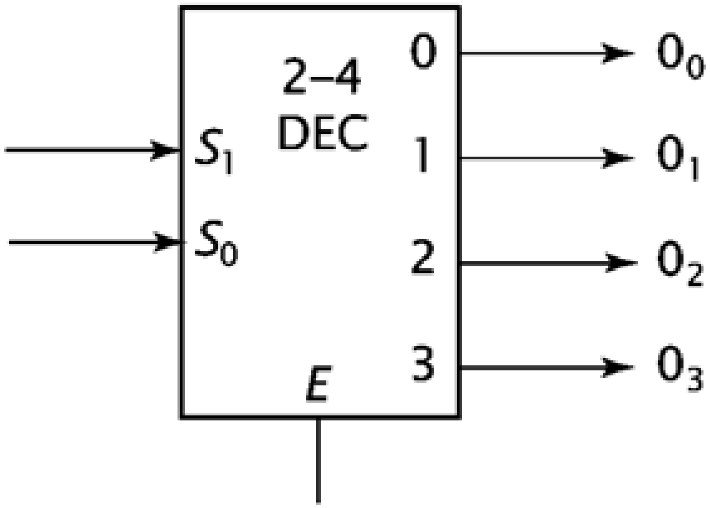
Decoder

- **Accepts a value and decodes it**
  - **Output corresponds to value of *n* inputs**
- **Consists of:**
  - Inputs (n)
  - Outputs (2^n , numbered from 0 $\rightarrow$ 2^n - 1)
  - Selectors / Enable (active high or active low)

The truth table of 2-to-4 Decoder

S_1	S_0	E	0_0	0_1	0_2	0_3
X	X	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

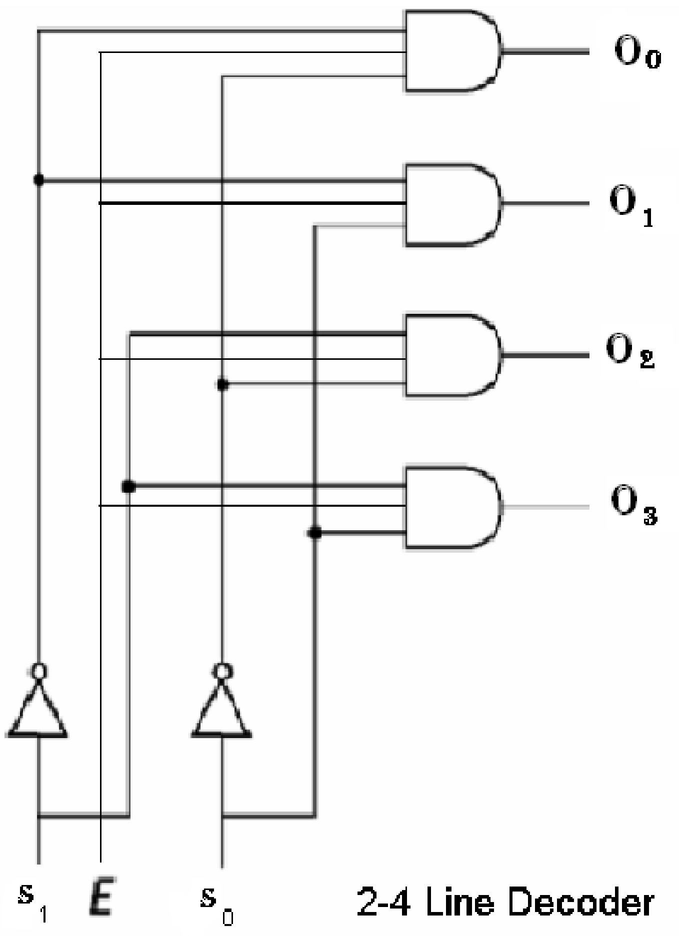
2-to-4 Decoder



S_1	S_0	E	0_0	0_1	0_2	0_3
X	X	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

(b)

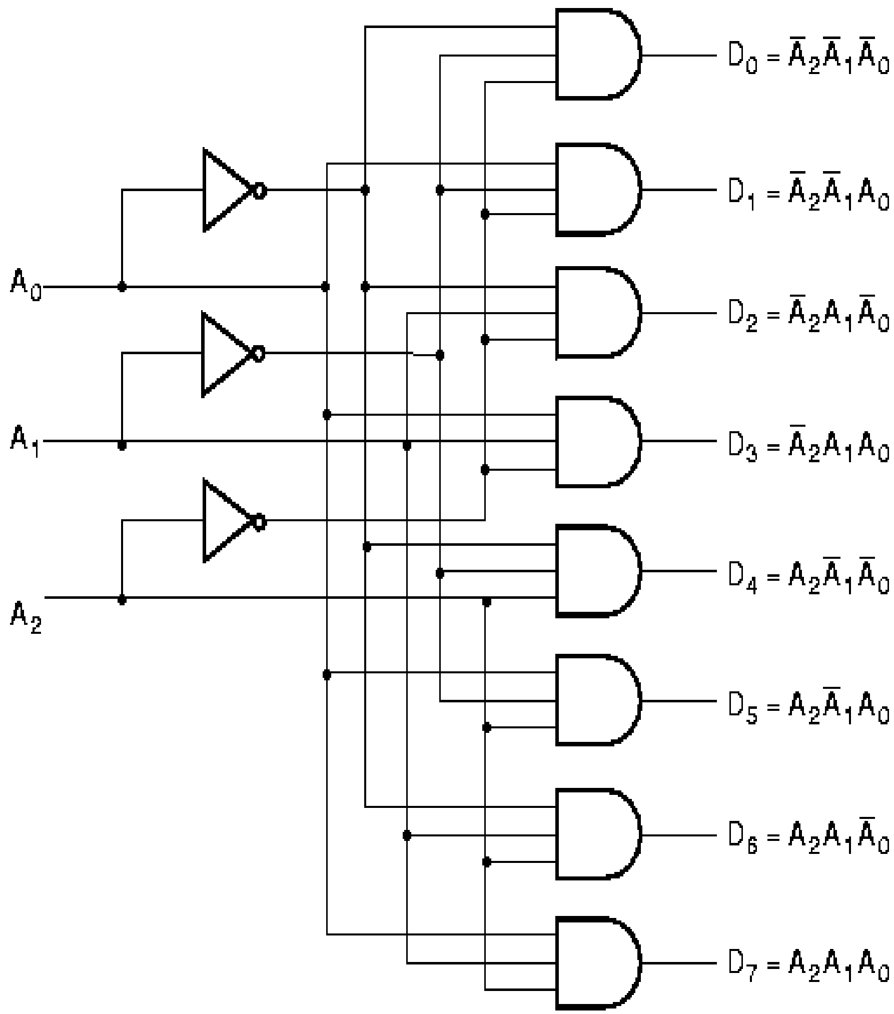
2-to-4 Decoder



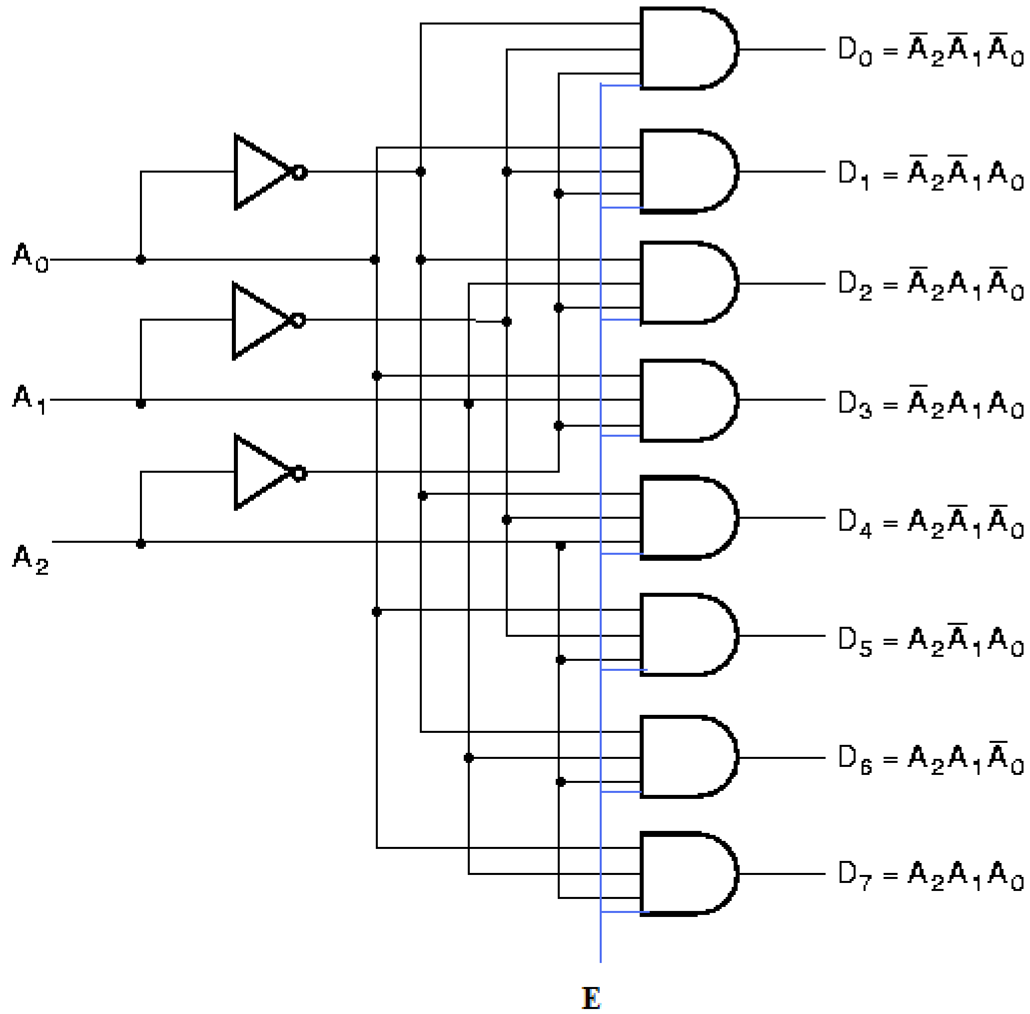
The truth table of 3-to-8 Decoder

A2	A1	A0	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

3-to-8 Decoder

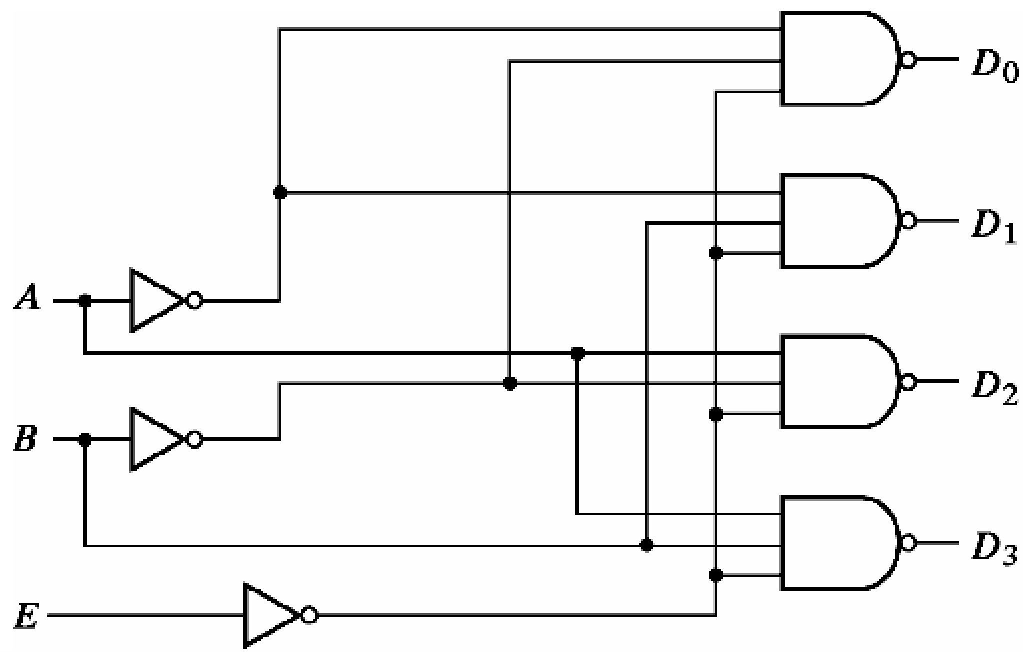


3-to-8 Decoder with Enable



2-to-4 Decoder: NAND implementation

Decoder is enabled when E=0 and an output is active if it is 0



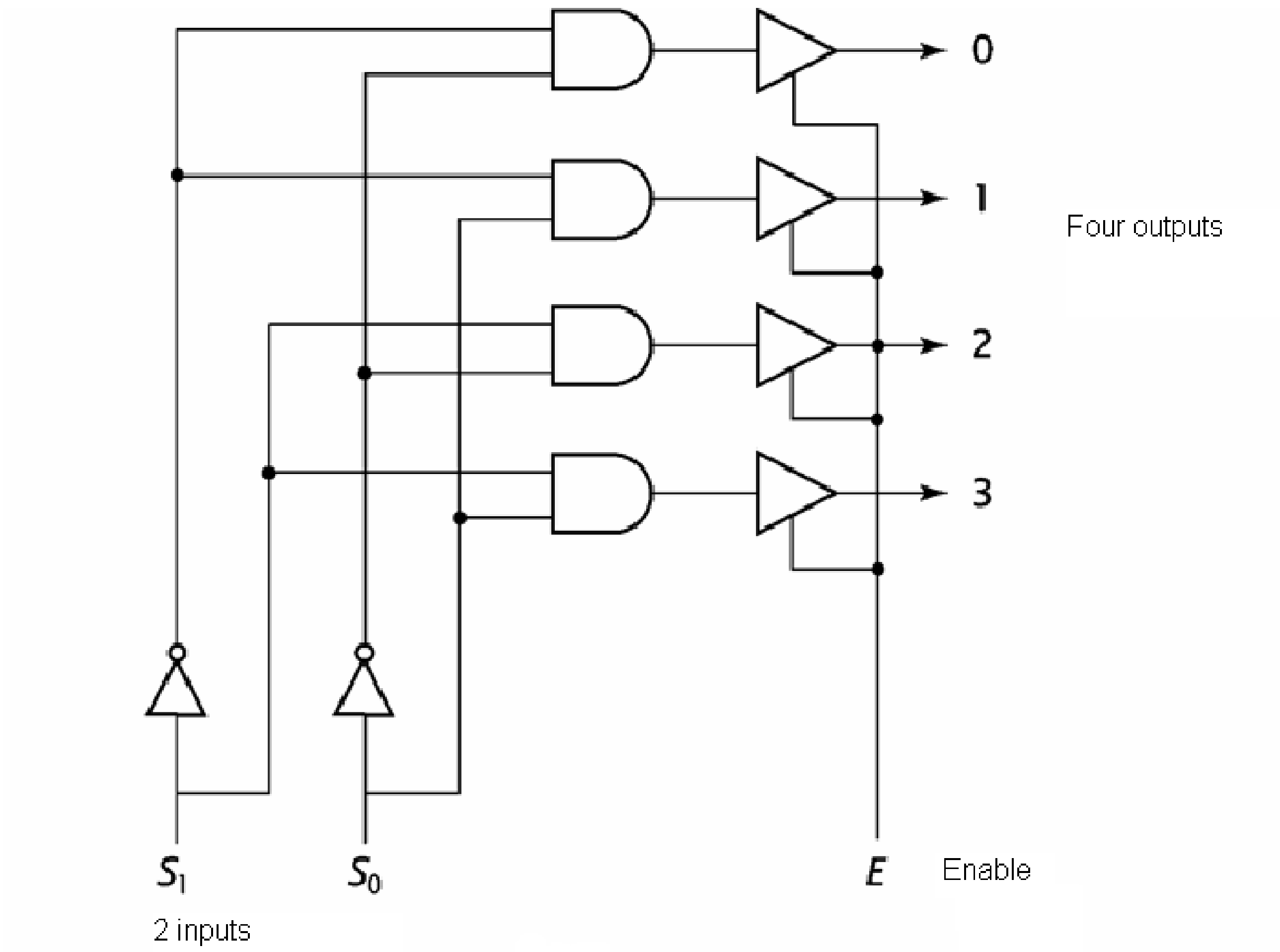
(a) Logic diagram

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

2-4 Decoder with 2-input and Enable

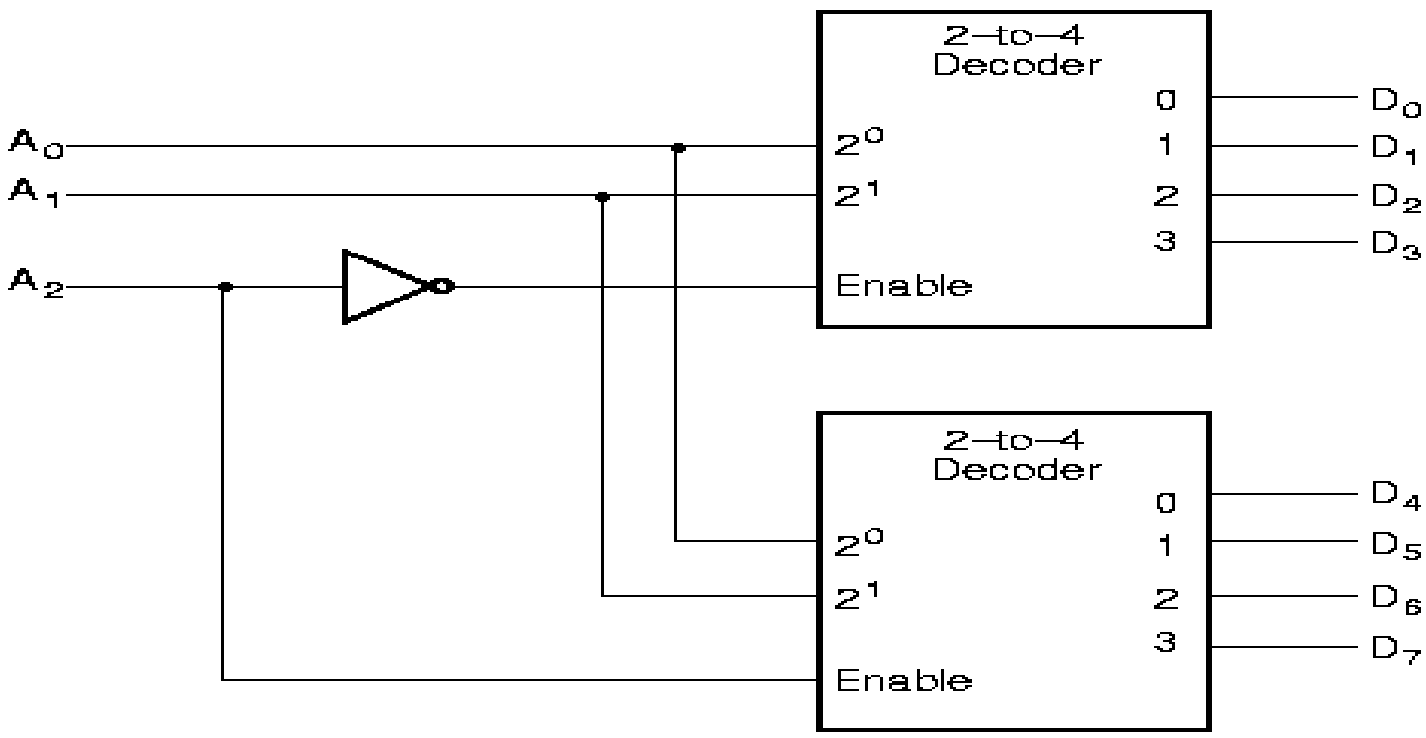


Decoder Expansion

- Decoder expansion
  - Combine two or more small decoders with enable inputs to form a larger decoder

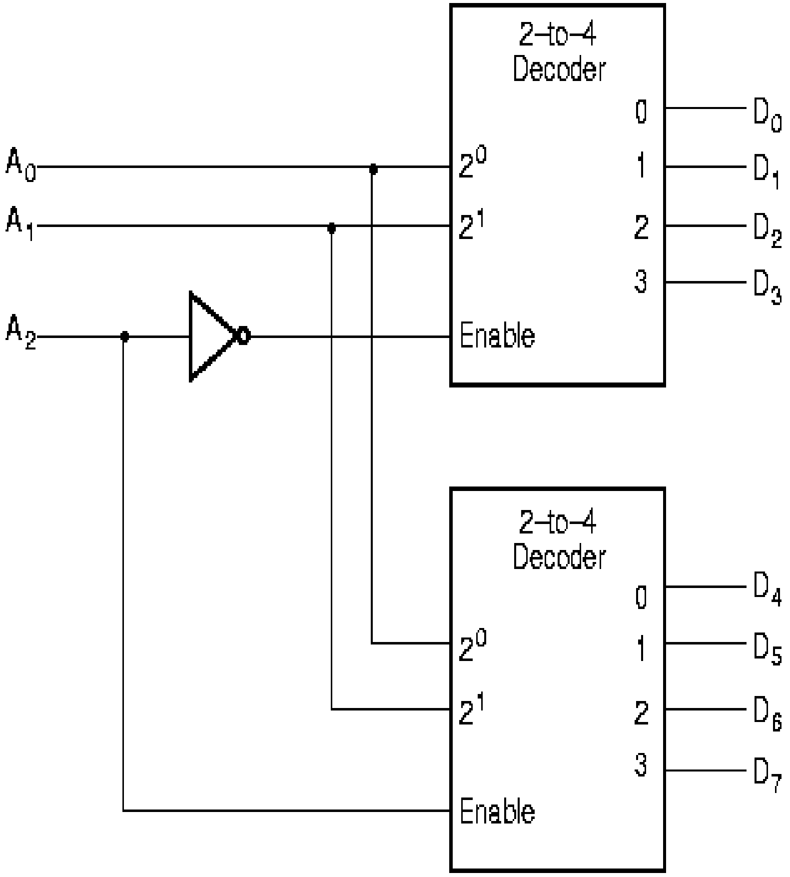
  - 3-to-8-line decoder constructed from two 2-to-4-line decoders
    - **The MSB is connected to the enable inputs**
    - **if A_2=0, upper is enabled; if A_2=1, lower is enabled.**

Decoder Expansion



Combining two 2-4 decoders to form one 3-8 decoder using enable switch

A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_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



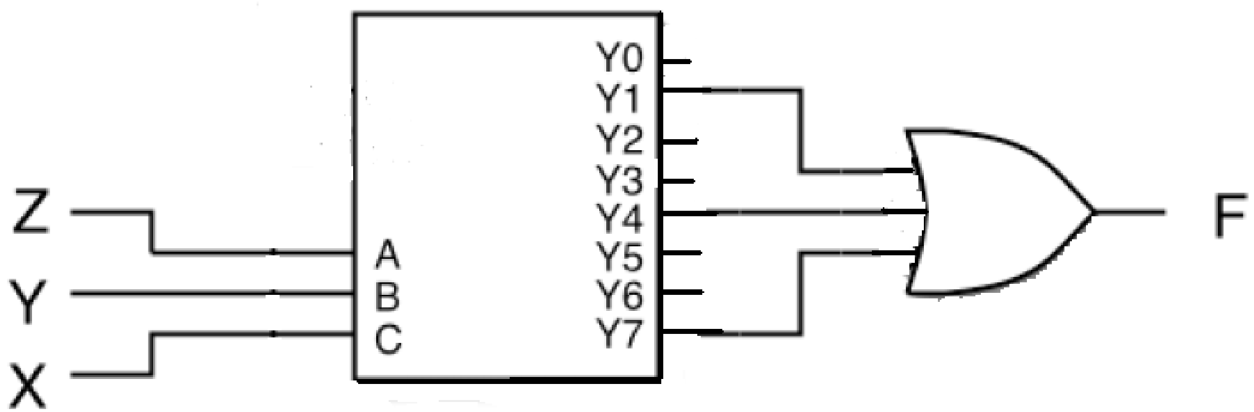
The highest bit is used for the enable

Combinational Circuit Design with Decoders

- Combinational circuit implementation with decoders
  - A decoder provide 2^n minterms of n input variables
  - Since any Boolean function can be expressed as a sum of minterms, one can use a decoder and external OR gates to implement any combinational function.

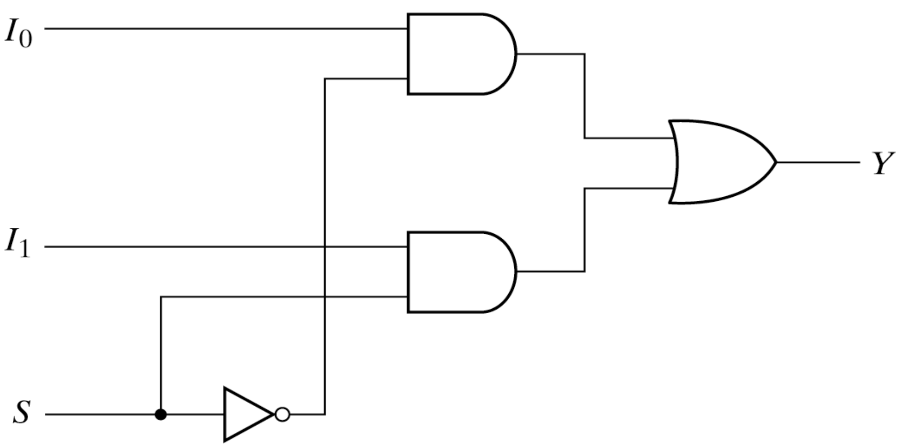
Combinational Circuit Design with Decoders

Example Realize $F(X,Y,Z) = \Sigma(1, 4, 7)$ with a decoder:

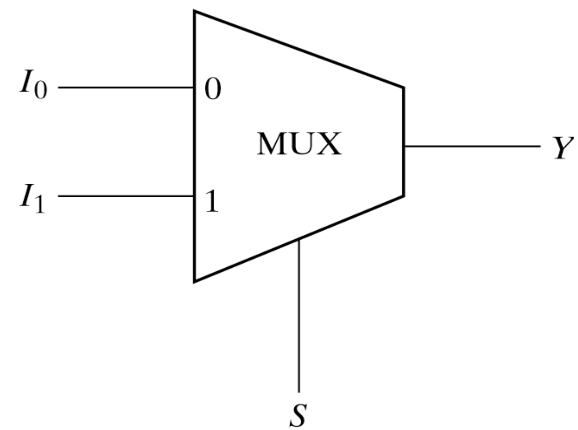


Multiplexers

- Select an input value with one or more select bits
- Use for transmitting data
- Allows for conditional transfer of data
- Sometimes called a mux
- EXAMPLE- 2:1 LINE MUX

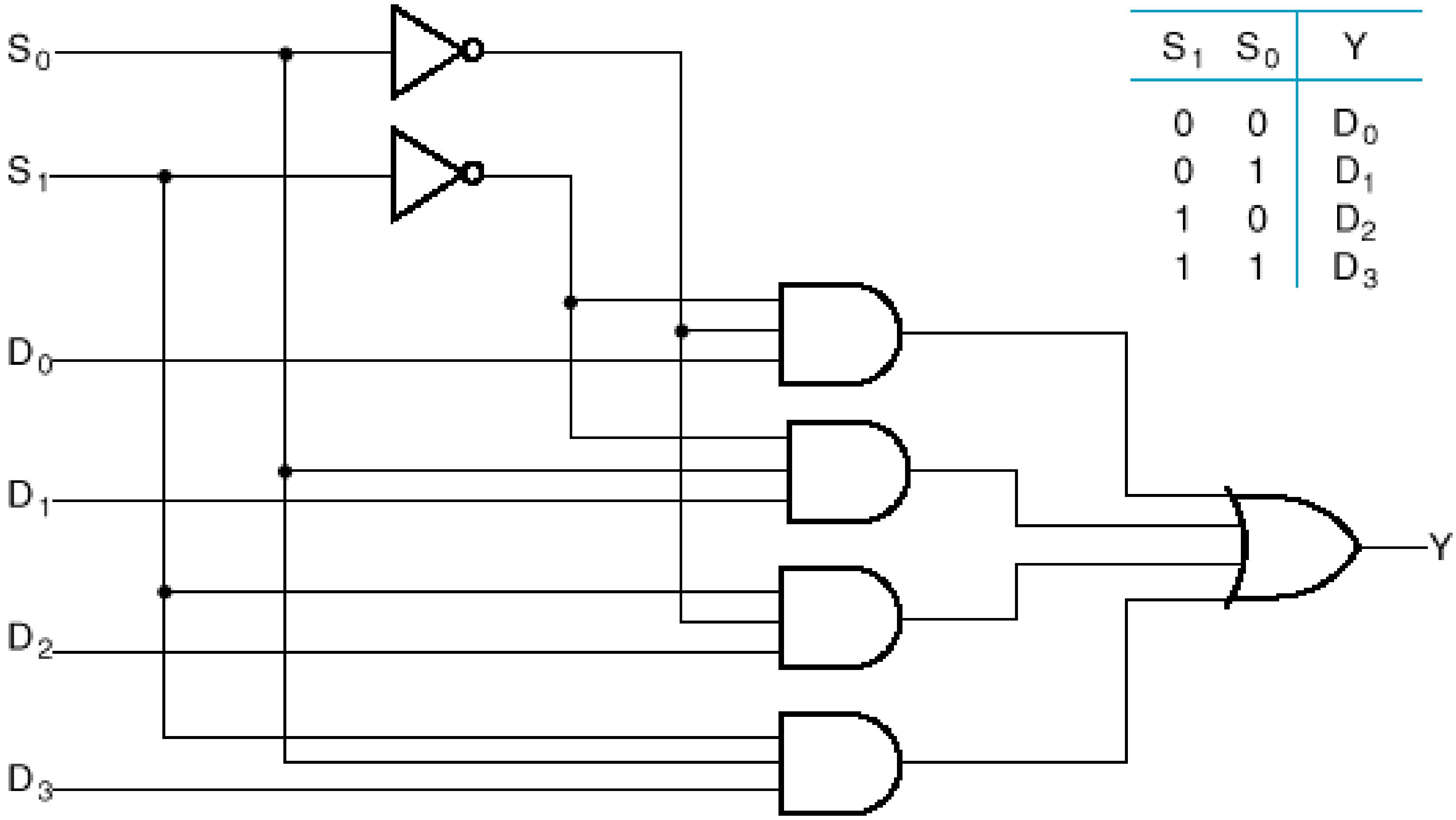


(a) Logic diagram



(b) Block diagram

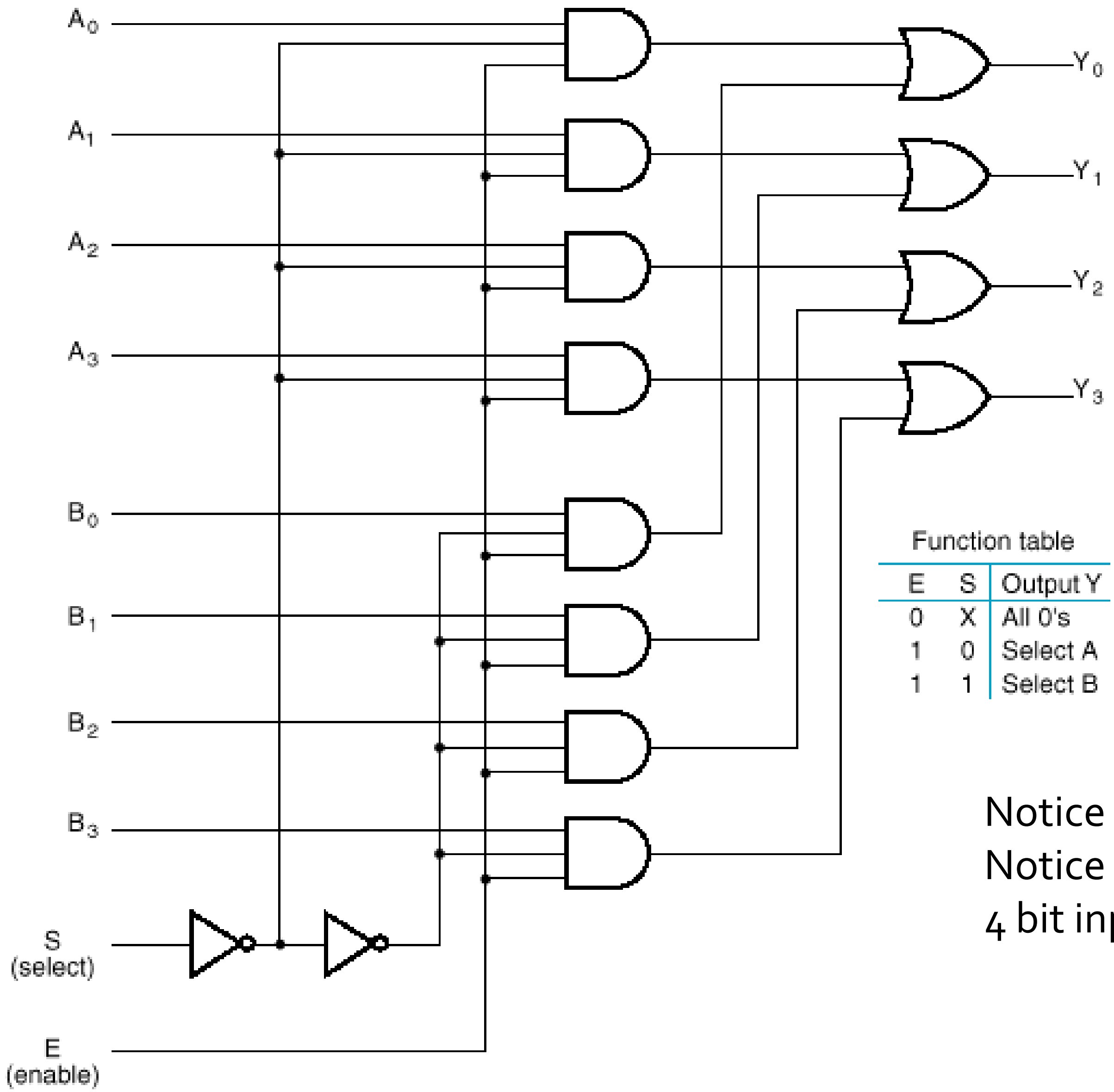
4– to– 1- Line Multiplexer



Function table

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Quadruple 2–to–1-Line Multiplexer



Function table

E	S	Output Y
0	X	All 0's
1	0	Select A
1	1	Select B

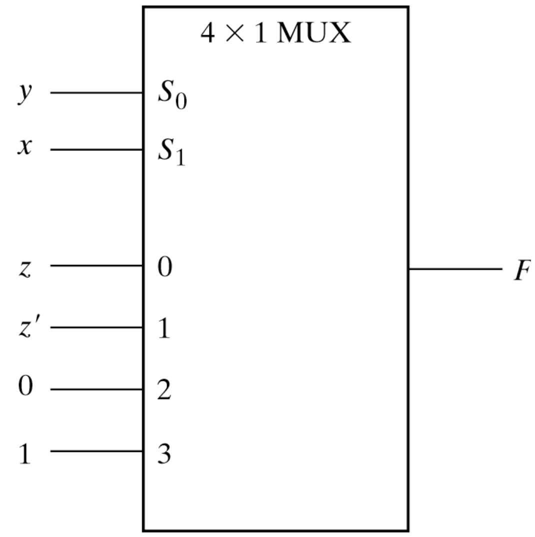
Notice enable bit
Notice select bit
4 bit inputs

Multiplexer as combinational modules

- Connect input variables to select inputs of multiplexer *(n-1 for n variables)*
- Set data inputs to multiplexer equal to values of function for corresponding assignment of select variables
- Using a variable at data inputs reduces size of the multiplexer

x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

(a) Truth table



(b) Multiplexer implementation

Fig. 4-27 Implementing a Boolean Function with a Multiplexer

Implementing a Four- Input Function with a Multiplexer

A	B	C	D	F	
0	0	0	0	0	F = D
0	0	0	1	1	
0	0	1	0	0	F = D
0	0	1	1	1	
0	1	0	0	1	F = $\overline{D}$
0	1	0	1	0	
0	1	1	0	0	F = 0
0	1	1	1	0	
1	0	0	0	0	F = 0
1	0	0	1	0	
1	0	1	0	0	F = D
1	0	1	1	1	
1	1	0	0	1	F = 1
1	1	0	1	1	
1	1	1	0	1	F = 1
1	1	1	1	1