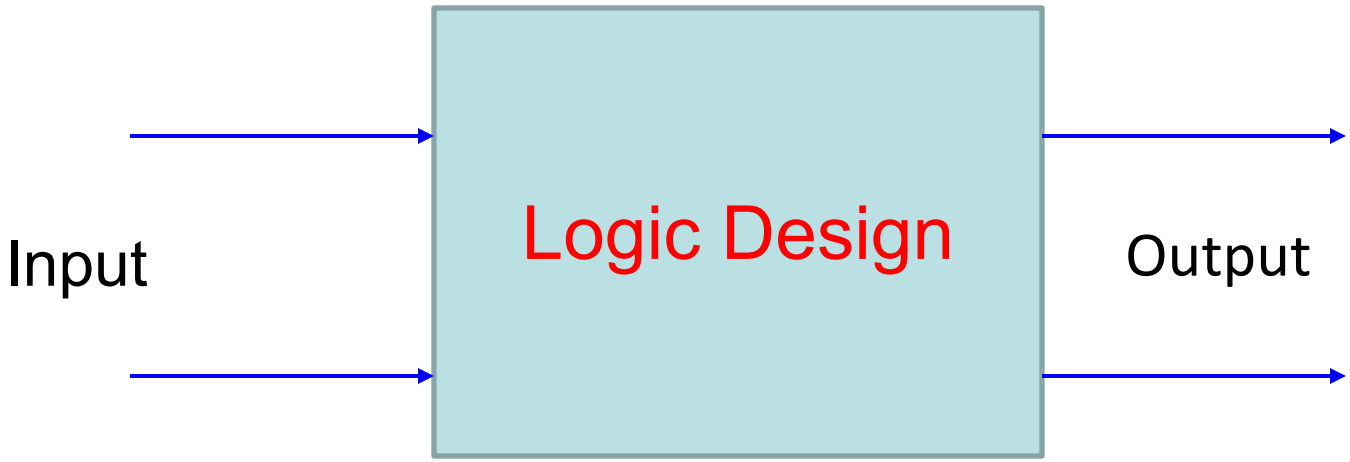
EE-224: Digital Design

Common Functions & Implementation

Virendra Singh
Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

Digital System



Digital Logic Design

- Express input output relationship using Truth table
- Generate the logical expression by disjunction (OR) of terms (conjunction of variables – AND) where system evaluates to true
- Replace all operators by the logic gates
- Replace logic gates by equivalent switching network (e.g., transistor level circuit)

Specification: Logic Expression

Truth Table

X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Logic Expression

$$F = \overline{X}.\overline{Y}.Z + X.\overline{Y}.\overline{Z} + X.\overline{Y}.Z$$

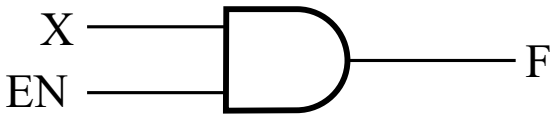
$$+ X.\overline{Y}.Z + X.Y.Z$$

Common Functions

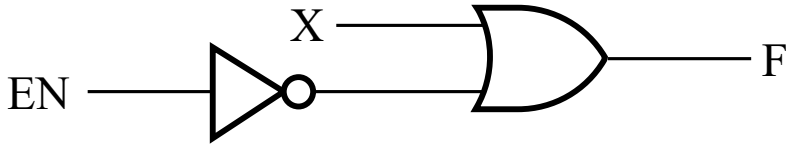
Enabling Function

- *Enabling* permits an input signal to pass through to an output
- *Disabling* blocks an input signal from passing through to an output, replacing it with a fixed value
- When disabled, 0 output
- When disabled, 1 output

EN	X	Y
0	0	0
0	1	0
1	0	0
1	1	1



(a)



(b)

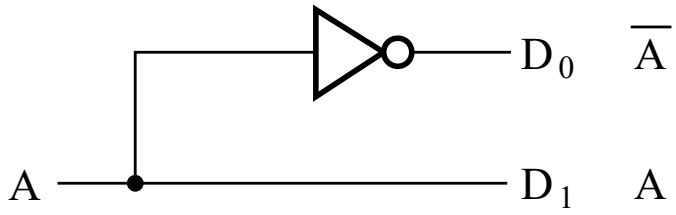
Decoding Function

- Decoding - the
  - Conversion of *n*-bit input to *m*-bit output
  - Given $n \leq m \leq 2^n$
- Circuits that perform decoding are called *decoders*
  - Called *n*-to-*m* line decoders, where $m \leq 2^n$, and
  - Generate 2^n (or fewer) 1's in output for the *n* input variables

Decoder

- 1-to-2-Line Decoder

A	D_0	D_1
0	1	0
1	0	1

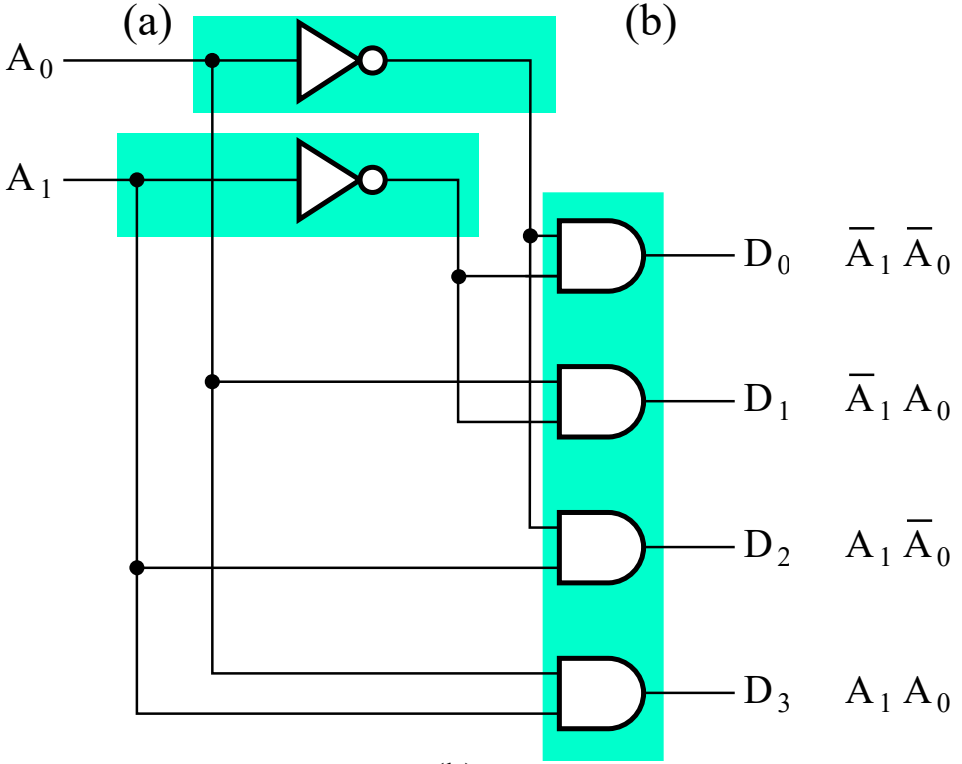


(a) (b)

- 2-to-4-Line Decoder

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(a)



(b)

Encoding Function

- Encoding - the opposite of decoding
  - Conversion of *m*-bit input to *n*-bit output
- Circuits that perform encoding are called *encoders*
  - An encoder has 2^n (or fewer) input lines and *n* output lines which generate the binary code corresponding to the input values
  - Typically, an encoder converts a code containing exactly one bit that is 1 to a binary code corres-ponding to the position in which the 1 appears.

Encoder

- A decimal-to-BCD encoder
  - Inputs: 10 bits corresponding to decimal digits 0 through 9, (D_0, …, D_9)
  - Outputs: 4 bits with BCD codes
  - Function: If input bit D_i is a 1, then the output (A_3, A_2, A_1, A_0) is the BCD code for i,
- The truth table could be formed, but alternatively, the equations for each of the four outputs can be obtained directly.

Encoder

- Input D_i is a term in equation A_j if bit A_j is 1 in the binary value for i.
- Equations:

$A_3 = D_8 + D_9$

$A_2 = D_4 + D_5 + D_6 + D_7$

$A_1 = D_2 + D_3 + D_6 + D_7$

$A_0 = D_1 + D_3 + D_5 + D_7 + D_9$

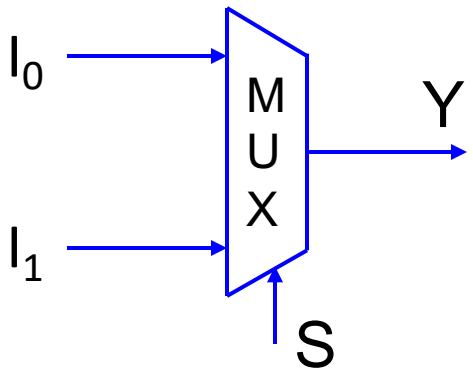
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001

Selection Function

- Selecting of data or information is a critical function in digital systems and computers
- Circuits that perform selecting have:
  - A set of information inputs from which the selection is made
  - A single output
  - A set of control lines for making the selection
- Logic circuits that perform selecting are called multiplexers

Multiplexers

- A multiplexer selects one input line and transfers it to output
  - *n* control inputs (S_{n-1}, … S_0) called *selection inputs*
  - m ≤ 2^n information inputs (I_{2^n-1}, … I_0)
  - output Y



2-to-1-Line Multiplexer

- Since $2 = 2^1$, n = 1
- The single selection variable S has two values:
  - S = 0 selects input I_0
  - S = 1 selects input I_1
- Truth Table
- Symbolic equation:

$$Y = I_0 . \overline{S} + S . I_1$$

- Logic expression

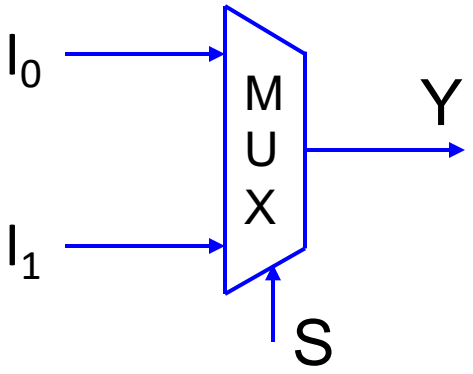
$$Y = \overline{S}.I_0.\overline{I_1} + \overline{S}.I_0.I_1 + S.\overline{I_0}.I_1 + S.I_0.I_1$$

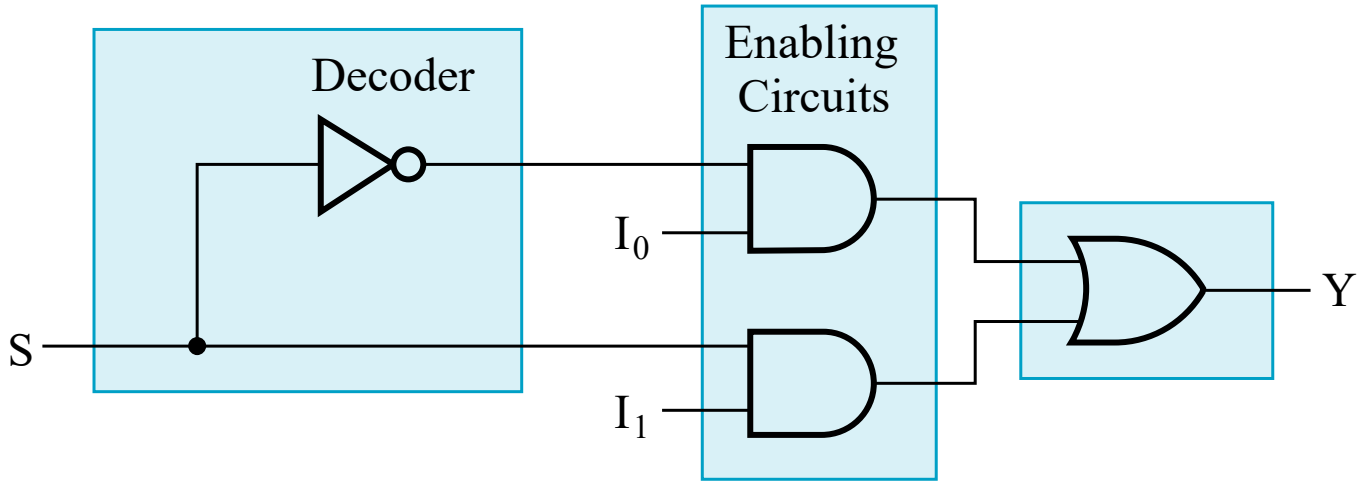
S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

2-to-1-Line Multiplexer

- The single selection variable S has two values:
  - S = 0 selects input I_0
  - S = 1 selects input I_1
- The logic equation:

$$Y = I_0 \overline{S} + S.I_1$$

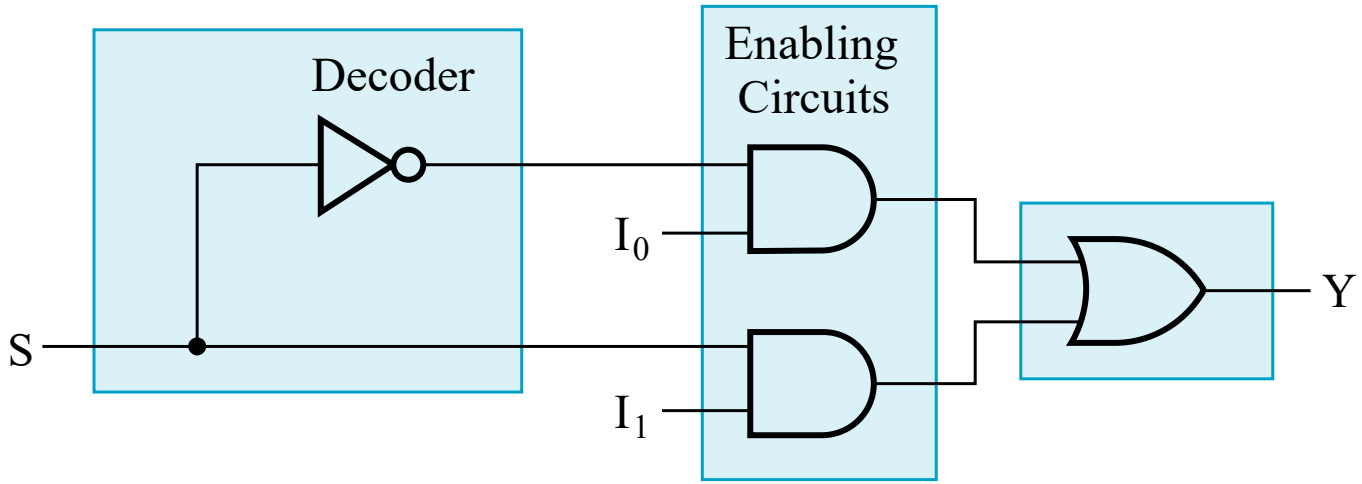




Implementation

Using Logic Gates

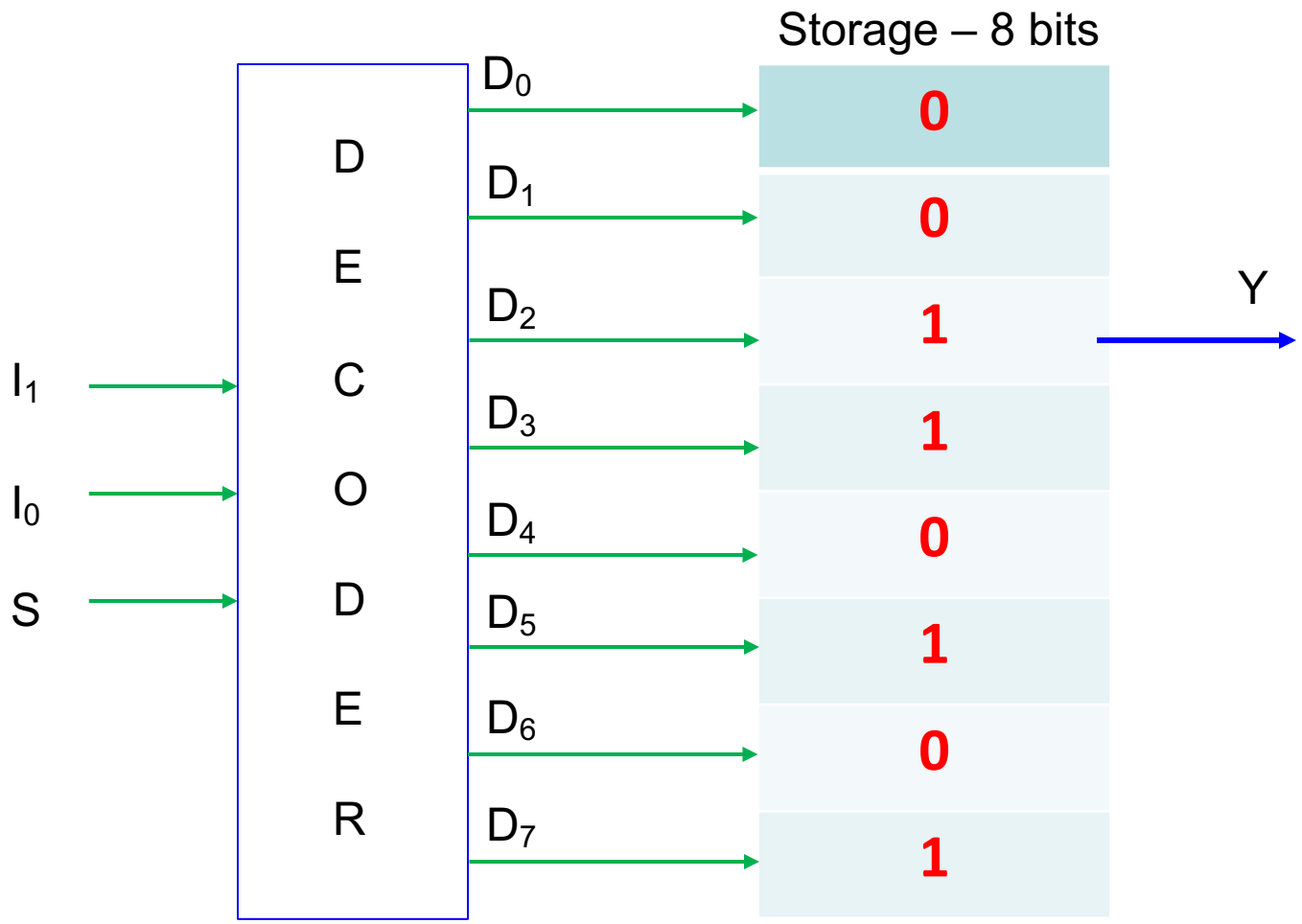
- 2x1 Multiplexer



Truth Table

S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Using Storage Elements



S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Optimization

Specification: Logic Function

Truth Table

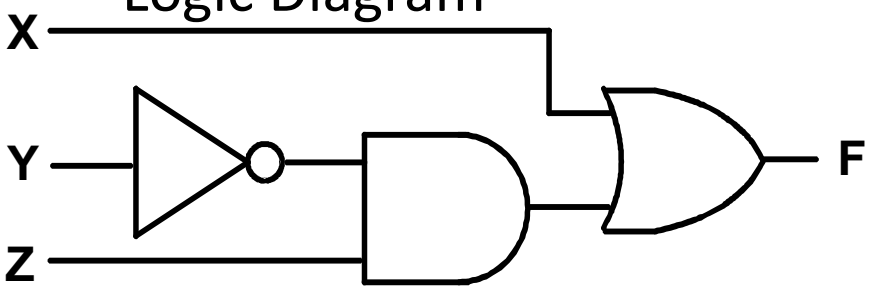
X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Logic Expression

$$F = \overline{X}.\overline{Y}.Z + X.\overline{Y}.\overline{Z} + X.\overline{Y}.Z + X.Y.\overline{Z} + X.Y.Z$$

$$F = X + \overline{Y}.Z$$

Logic Diagram



Optimization Parameters

- Area: # Switches (Gates)
- Performance (Delay): # Switches in series
- Power: # Switches
- Testability: Interconnect network
- Security
- Intelligence

Thank You