BME2322 – Logic Design

The Instructors:

Dr. Görkem SERBES (C317)

gserbes@yildiz.edu.tr

https://avesis.yildiz.edu.tr/gserbes/

Lab Assistants:

Nihat AKKAN

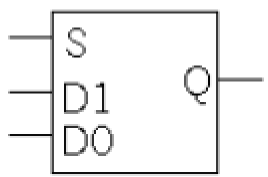
nakkan@yildiz.edu.tr

https://avesis.yildiz.edu.tr/nakkan

LECTURE 8

Multiplexers

- A 2^n-to-1 multiplexer sends one of 2^n input lines to a single output line
  - A multiplexer has two sets of inputs:
    - 2^n data input lines
    - n select lines, to pick one of the 2n data inputs
  - The mux output is a single bit, which is one of the 2n data inputs
- The simplest example is a 2-to-1 mux:

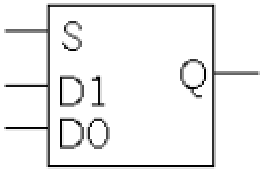


$$Q = S' D0 + S D1$$

- The select bit S controls which of the data bits D0-D1 is chosen:
  - If S=0, then D0 is the output (Q=D0).
  - If S=1, then D1 is the output (Q=D1).

Truth table representation

- Here is a full truth table for this 2-to-1 mux, based on the equation:



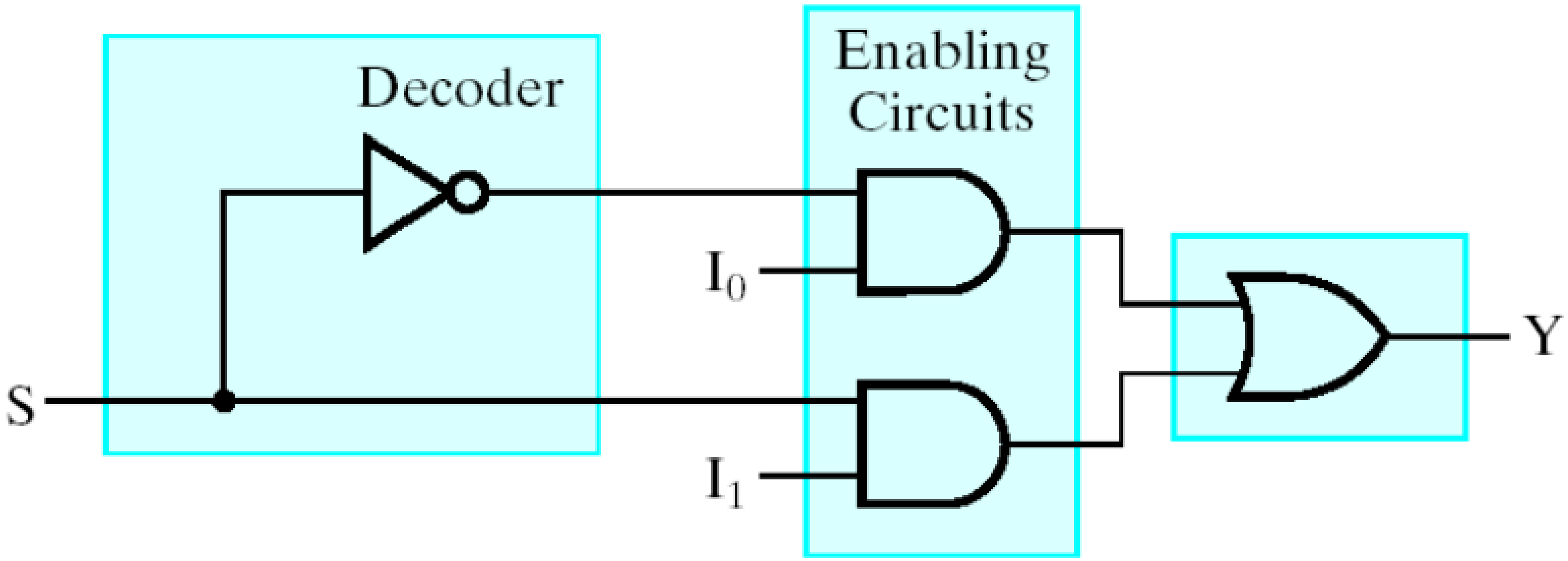
$$Q = S' D0 + S\ D1$$

S	D1	D0	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- Here is another kind of abbreviated truth table
  - Input variables appear in the output column
  - This table implies that when S=0, the output Q=D0, and when S=1 the output Q=D1
  - This is a close match to the equation

S	Q
0	D0
1	D1

2-to-1 Mux



4-to-1 Mux

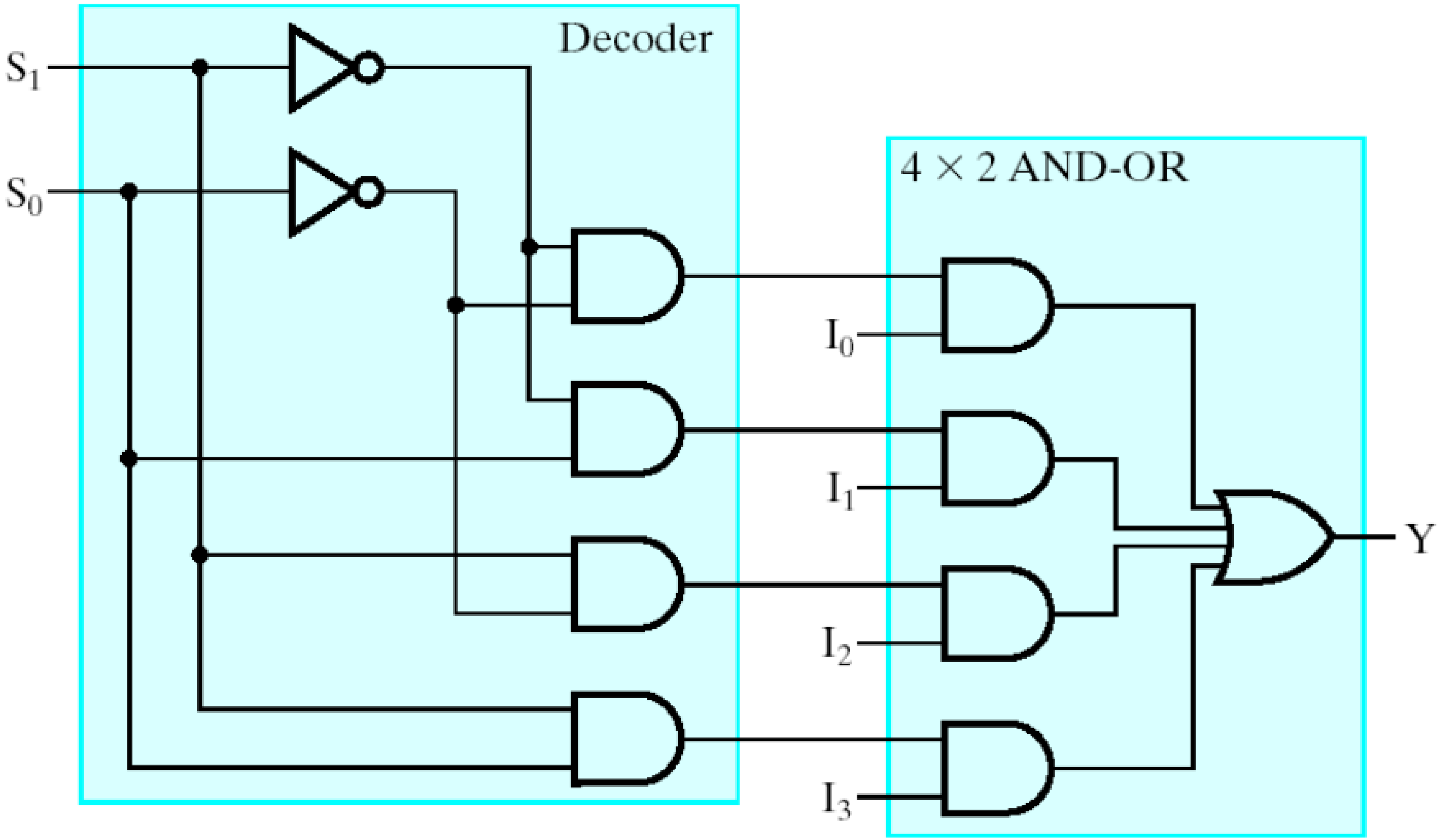
Condensed Truth Table for 4-to-1-Line Multiplexer

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

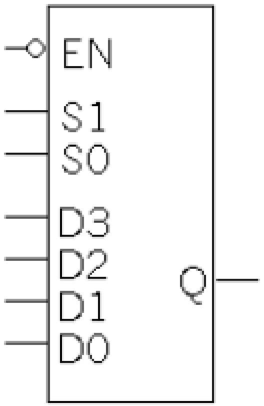
$$Y = (S1'S0')I0 + (S1'S0)I1 + (S1S0')I2 + (S1S0)I3$$

Decoder

4-to-1 Mux



4-to-1 Mux with Enable Input



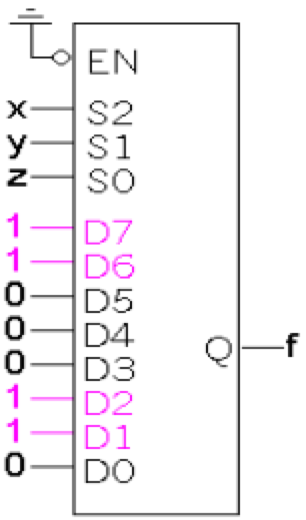
EN	S1	S0	Q
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	x	x	1

$$Q = EN'S1'\ S0'\ D0 + EN'S1'\ S0\ D1 + EN'S1\ S0'\ D2$$
$$+ EN'S1\ S0\ D3 + EN$$

Implementing functions with multiplexers

- Muxes can be used to implement arbitrary functions
- One way to implement a function of n variables is to use an 2^n-to-1 mux:
  - For each minterm mi of the function, connect 1 to mux data input Di. Each data input corresponds to one row of the truth table
  - Connect the function's input variables to the mux select inputs. These are used to indicate a particular input combination
- For example, let's look at $f(x,y,z) = \Sigma m(1,2,6,7)$.

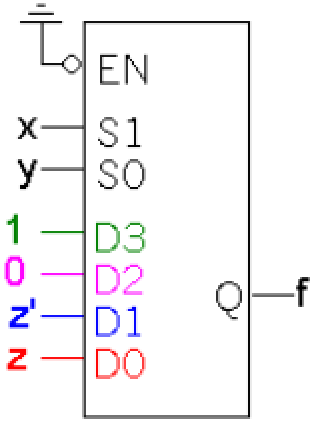
x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



A more efficient way

- We can actually implement f(x,y,z) = Σm(1,2,6,7) with just a 4-to-1 mux, instead of an 8-to-1
- Step 1: Find the truth table for the function, and group the rows into pairs. Within each pair of rows, x and y are the same, so f is a function of z only.
  – When xy=00, f=z
  – When xy=01, f=z'
  – When xy=10, f=0
  – When xy=11, f=1
- Step 2: Connect the first two input variables of the truth table (here, x and y) to the select bits S1 S0 of the 4-to-1 mux.
- Step 3: Connect the equations above for f(z) to the data inputs D0-D3.

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



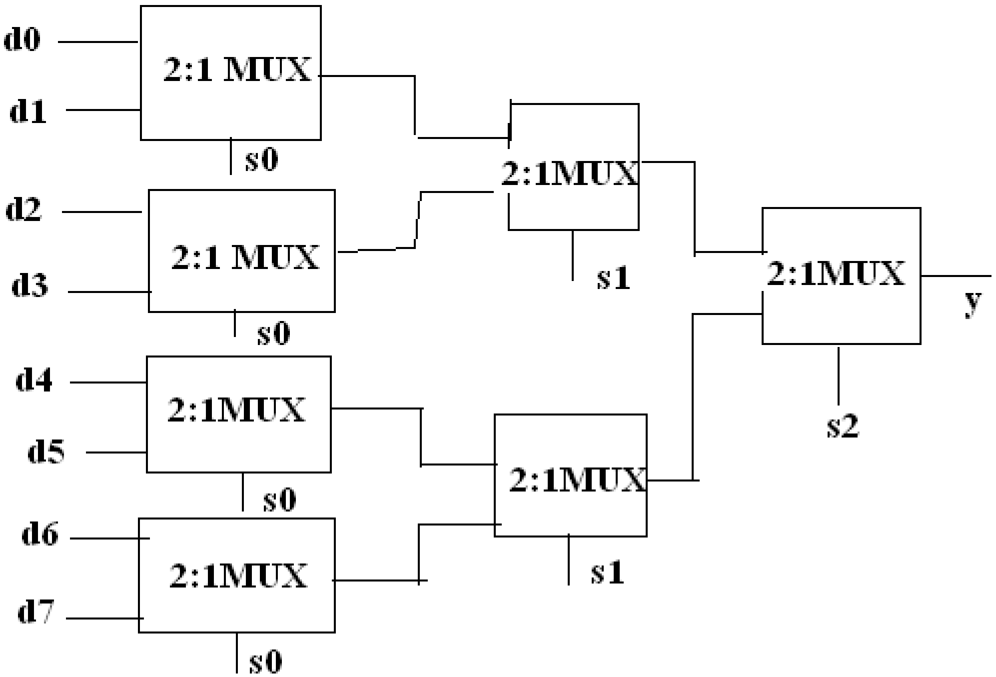
Example 1

- Implement a 4×1 mux by using 2×1 muxes

Example 2

- Implement a 8×1 mux by using 2×1 muxes

Example 2 Cont.



Example 3

- Implement a 8×1 mux by using 4×1 muxes

Example 4

- Implement $F(A,B,C,D) = \sum m(1,4,5,7,9,12,13)$ using a 4×1 mux

Summary of Multiplexers

- A 2n-to-1 multiplexer routes one of 2n input lines to a single output line
- Just like decoders,
  – Muxes are common enough to be supplied as stand-alone devices for use in modular designs.
  – Muxes can implement arbitrary functions
- We saw some variations of the standard multiplexer:
  – Smaller muxes can be combined to produce larger ones
  – We can add active-low or active-high enable inputs
- As always, we use truth tables and Boolean algebra to analyze things