Introduction to Digital Systems
Part II (4 lectures)
2024/2025

Combinational Logic Blocks

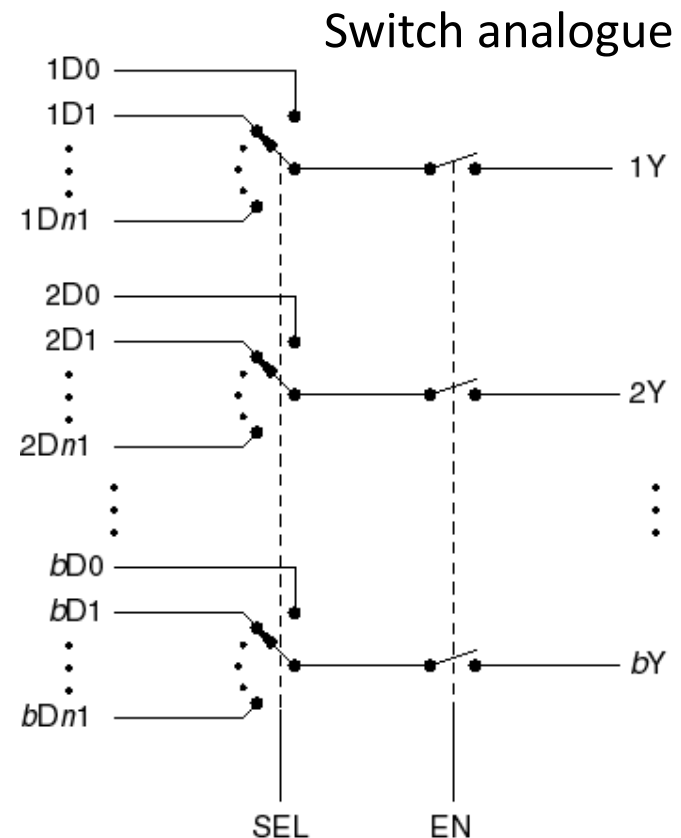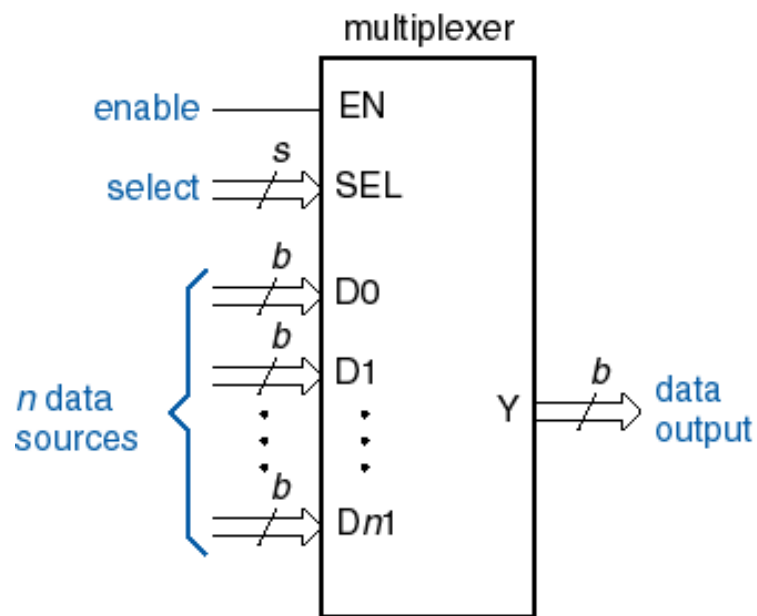Arnaldo Oliveira, Augusto Silva, Iouliia Skliarova
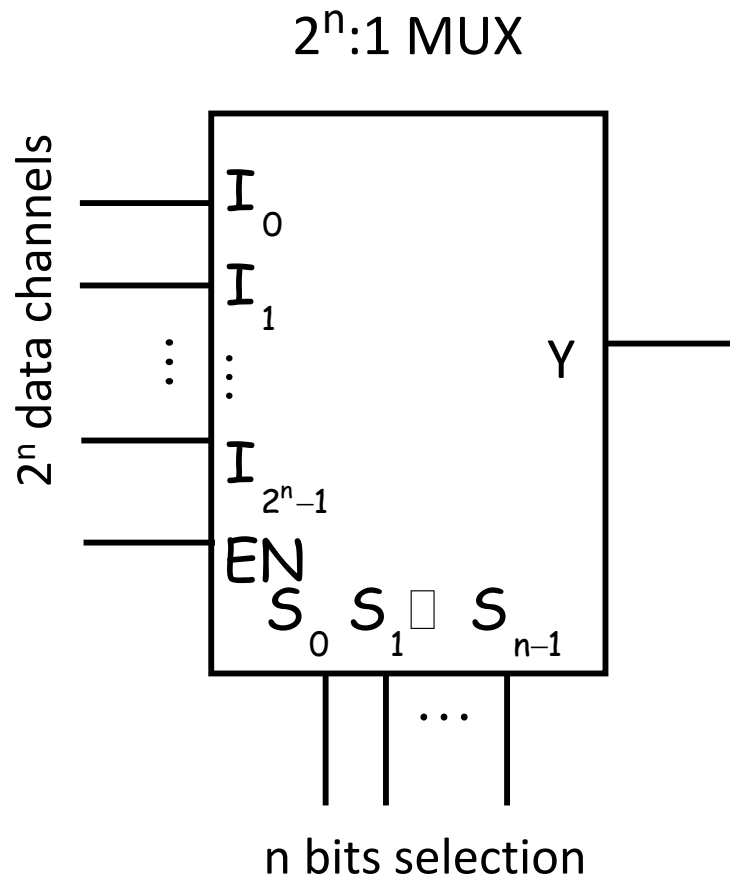
Universidade de Aveiro

Lecture 6 contents

- Block oriented combinational logic design
- Multiplexers
- Demultiplexers

Multiplexers

- A **multiplexer** is a **digital switch:** one out of *n* data sources is passed to a **single output**
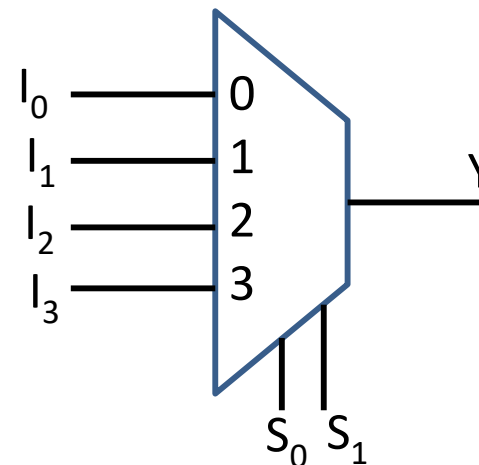 - Information selector

Switch analogue

2ⁿ:1 Mux models

2^n:1 MUX

2^n data channels

I_0
I_1
...
I_{2^n-1}
EN
S_0 S_1 ☐ S_{n-1}

Y

n bits selection

4:1 Mux Truth Table

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

Alternate Symbol

I_0 — 0
I_1 — 1
I_2 — 2
I_3 — 3

Y

S_0 S_1

Functional description

4:1 Mux Truth Table

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

Alternate Symbol

$$Y = EN.\left[\sum_{k=0}^{2^n-1} m(S)_k I_k\right]$$

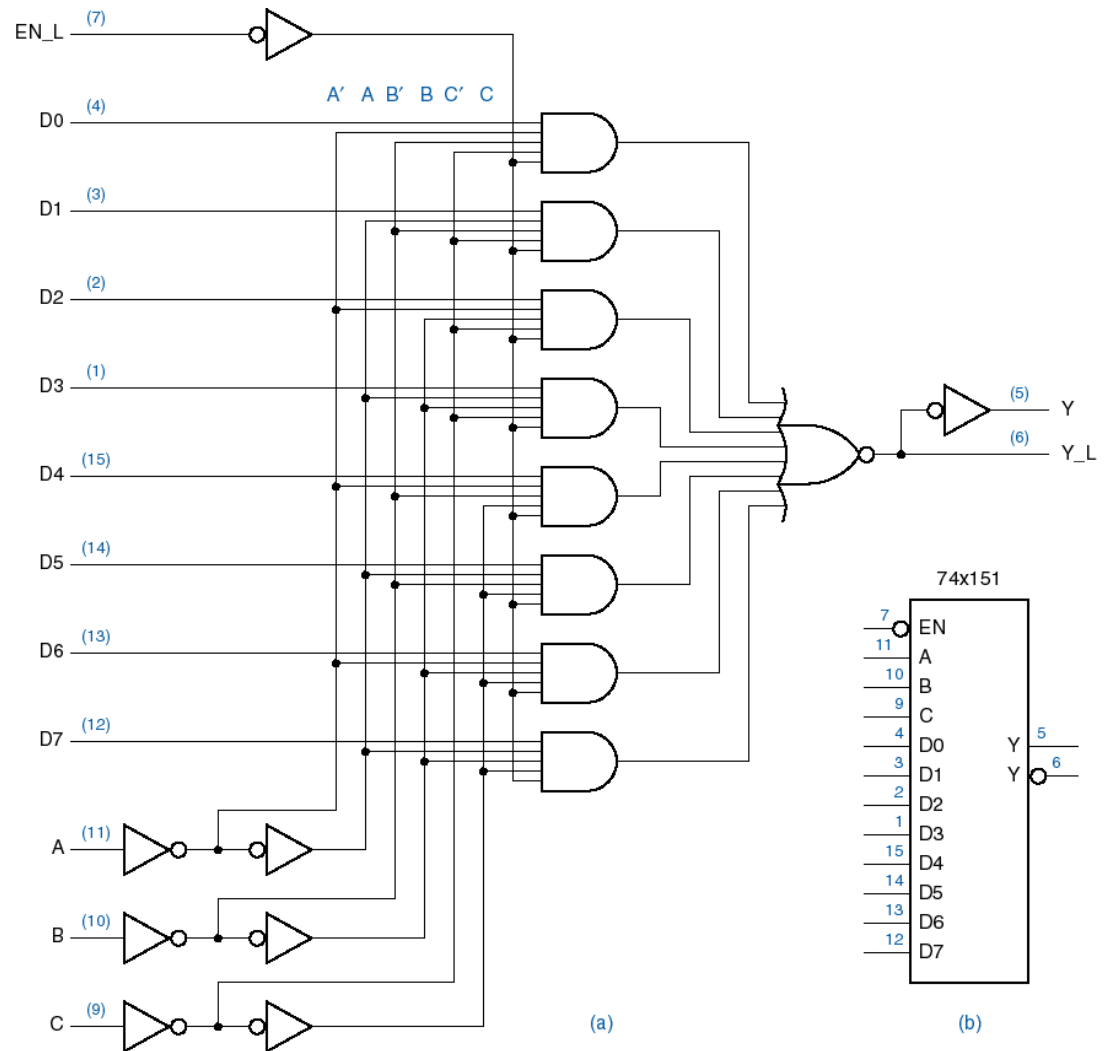$m(S)_k$ is the kth minterm on the selection variables $S_0...S_{n-1}$
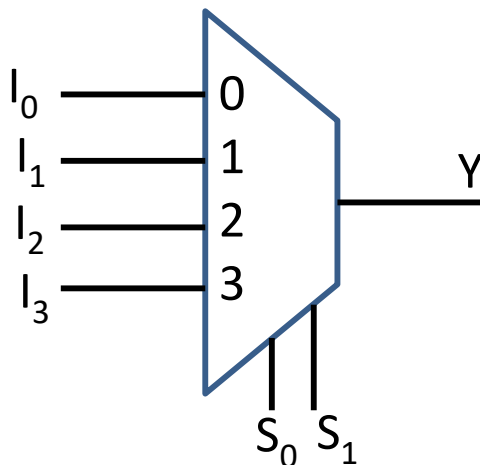
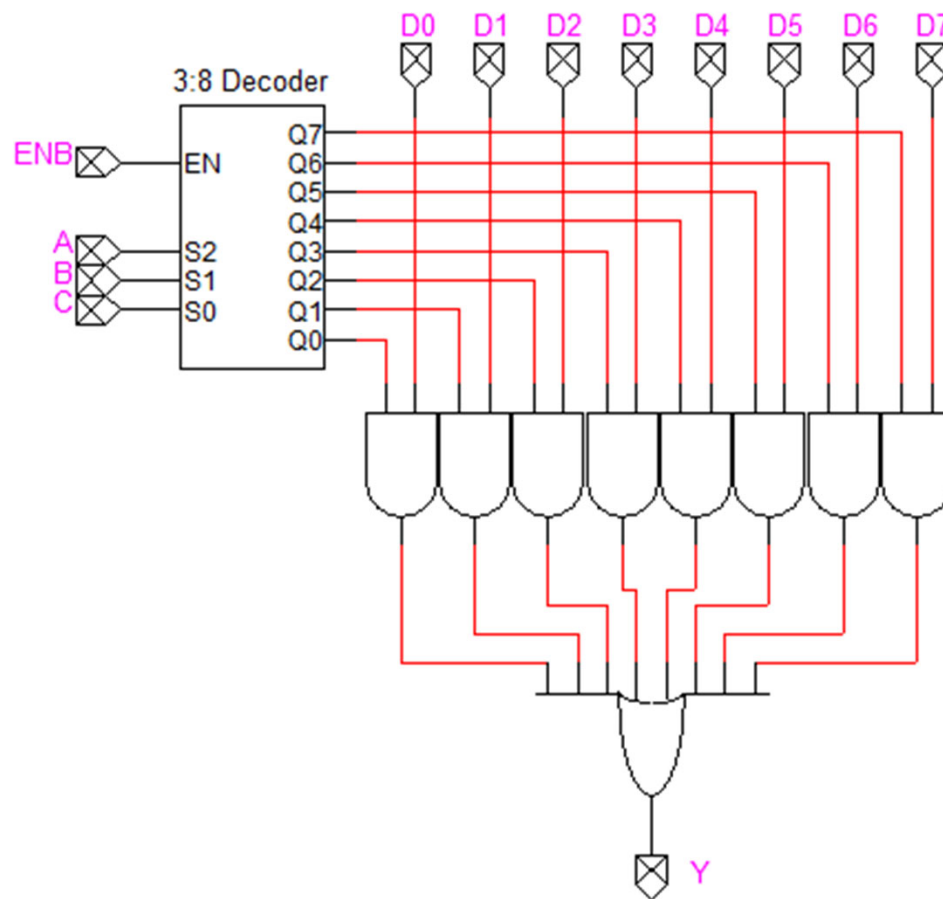Exercise: Draw the 4:1 Mux internal logic circuit

The 74151 model

- 8:1 mux
- Obtain the truth table
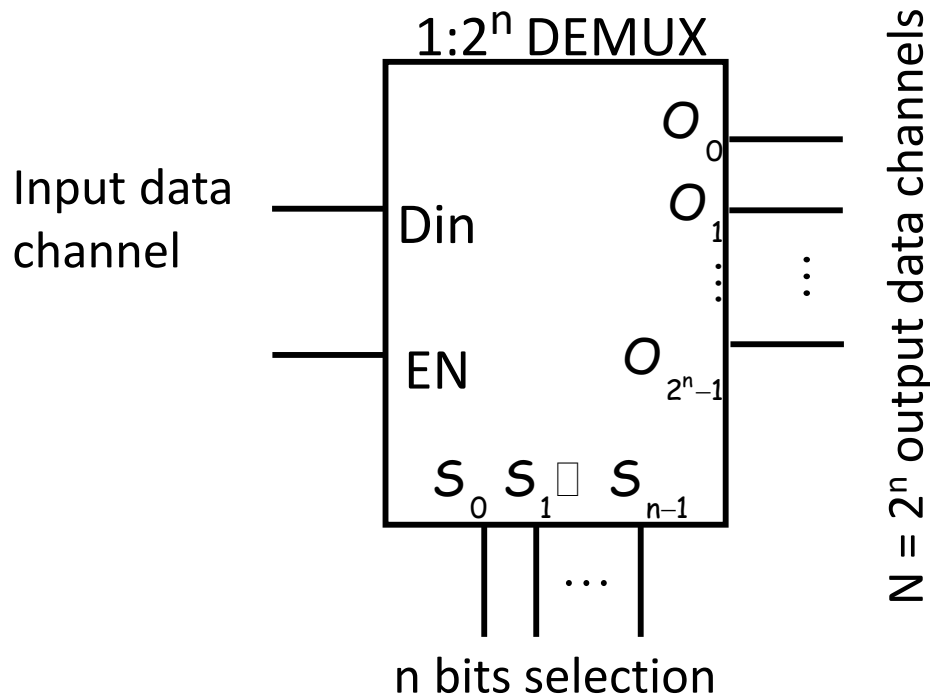- Write the output equations

Mux and decoders

- Verify that the logic circuit is a 8:1 Mux

Demultiplexers

- Functional inverse of a multiplexer

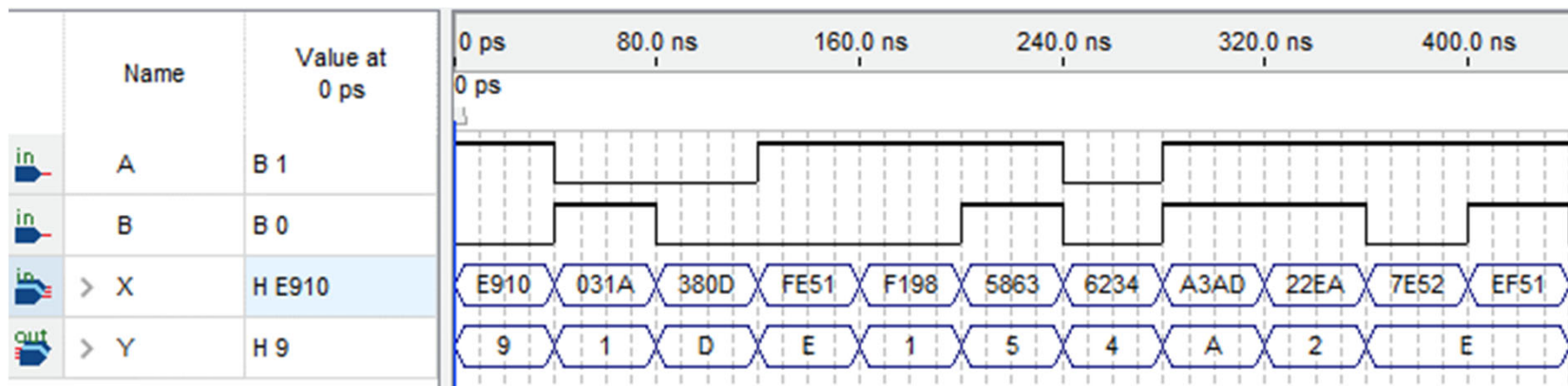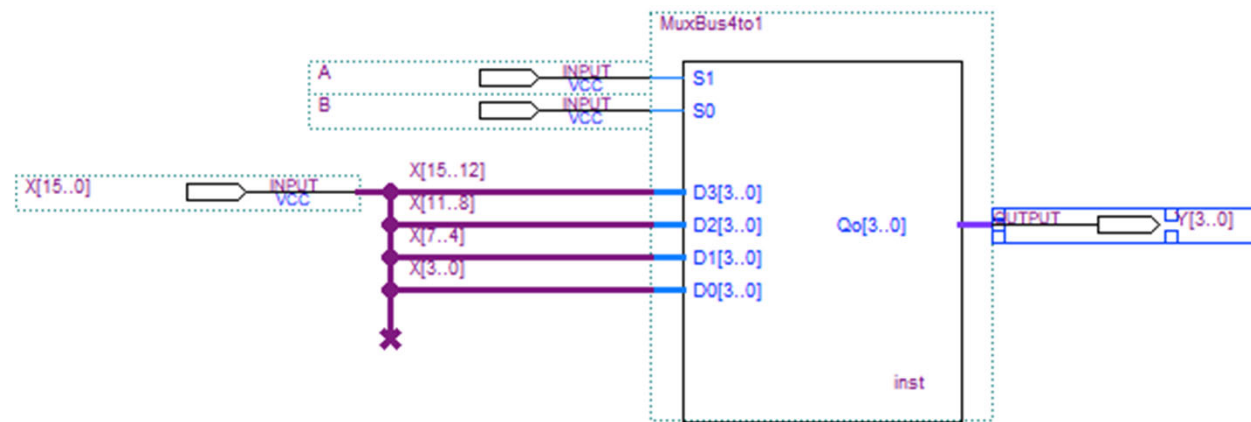 - An inverse digital switch: a single input is "routed" to one out of *N outputs*

$$O_k = EN.D_{in}m_k(S), \qquad k = 0, \dots 2^n - 1$$

1:2^n DEMUX

Input data channel

Din

O_0

O_1

$\cdots$

EN

O_{2^n-1}

$S_0\ S_1\ \square\ S_{n-1}$

$\cdots$

n bits selection

N = 2^n output data channels

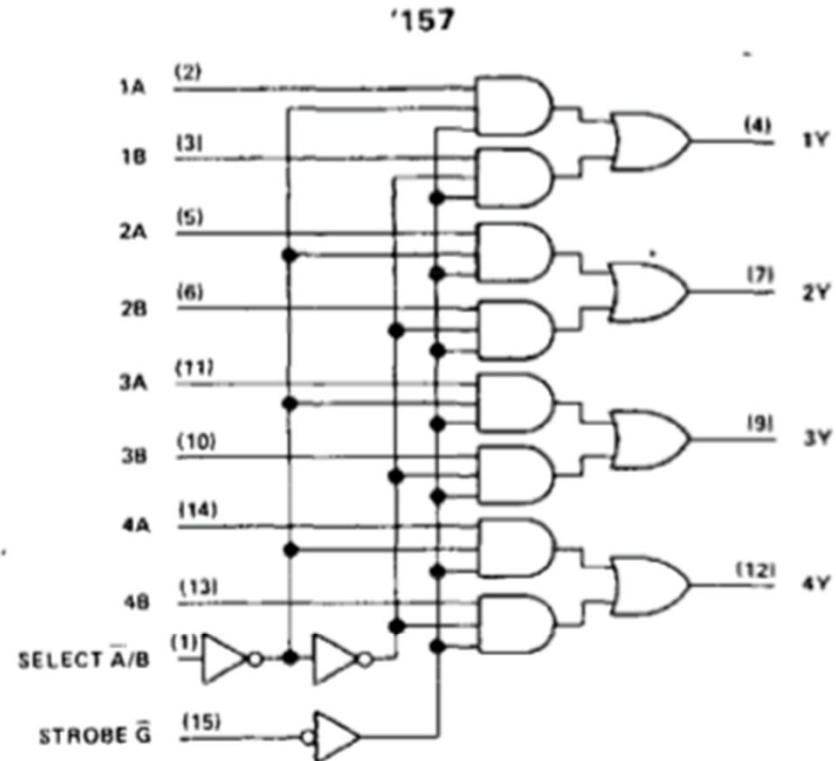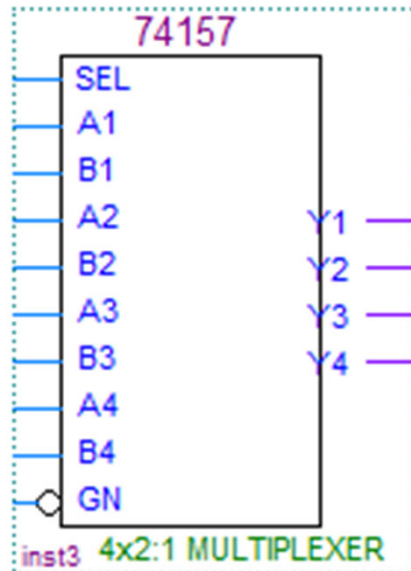Question: How can we use a demux as a decoder?

Multiplexing multibit data channels

- Mux 4:1, 4 bit input data channels
- Explain the timing diagram
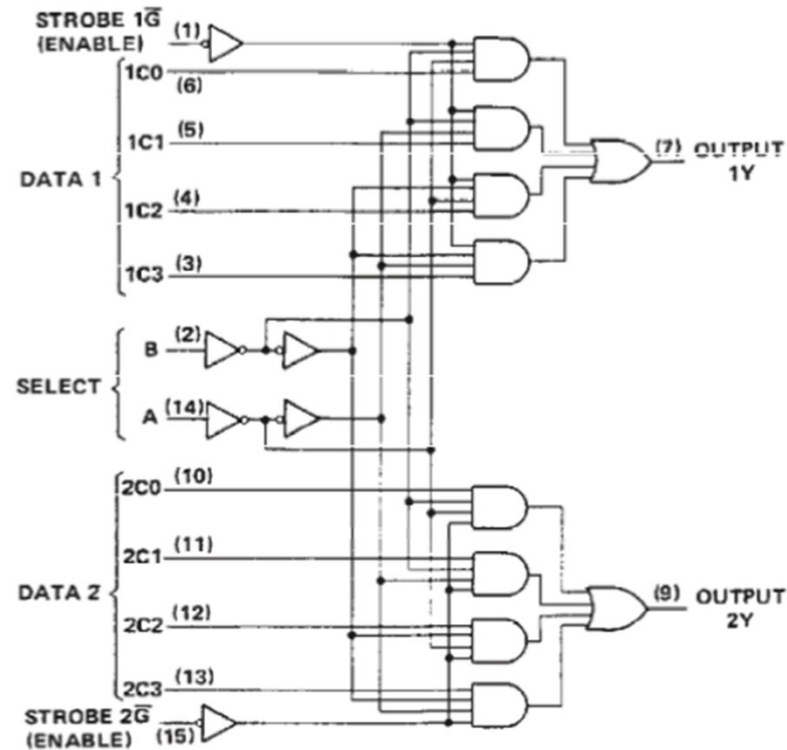
The 74157 model

- 74157: 4x2:1

74157
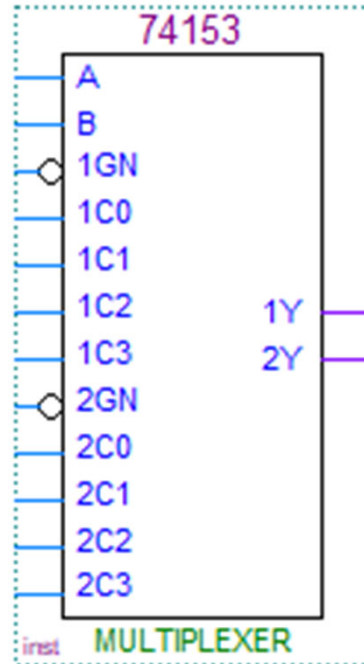
```
SEL
A1
B1
A2          Y1
B2          Y2
A3          Y3
B3          Y4
A4
B4
GN
```
inst3 4x2:1 MULTIPLEXER

'157

1A	(2)
1B	(3)					(4) 1Y
2A	(5)
2B	(6)					(7) 2Y
3A	(11)
3B	(10)					(9) 3Y
4A	(14)
4B	(13)					(12) 4Y
SELECT A̅/B	(1)
STROBE G̅	(15)

FUNCTION TABLE

INPUTS				OUTPUT Y	
STROBE G̅̅	SELECT A̅/B	A	B	'157, 'LS157, 'S157	'LS158 'S158
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L

H = high level, L = low level, X = irrelevant

The 74153 model

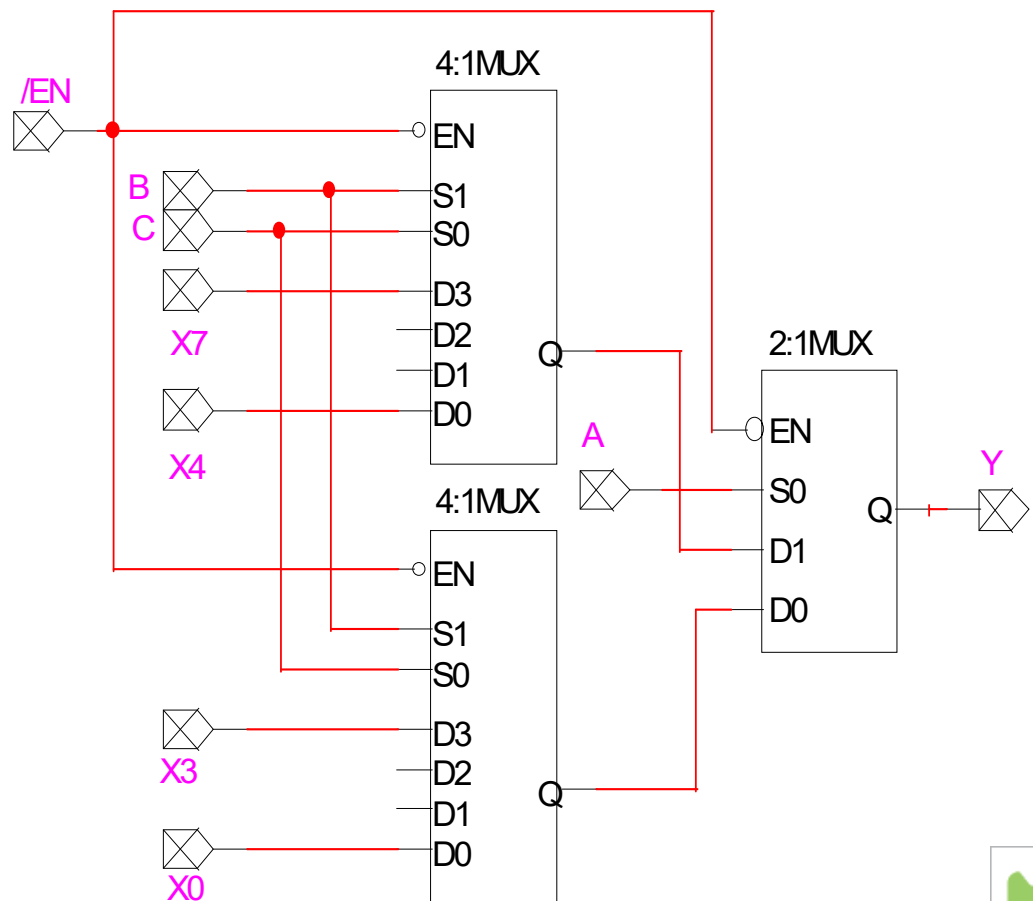- 74153: 2x4:1

Mux hierarchies

- 8:1 with 7x(2:1)

  - Always check the design by obtaining the truth table.

  - Note that IN THIS case A is the most significant selection variable
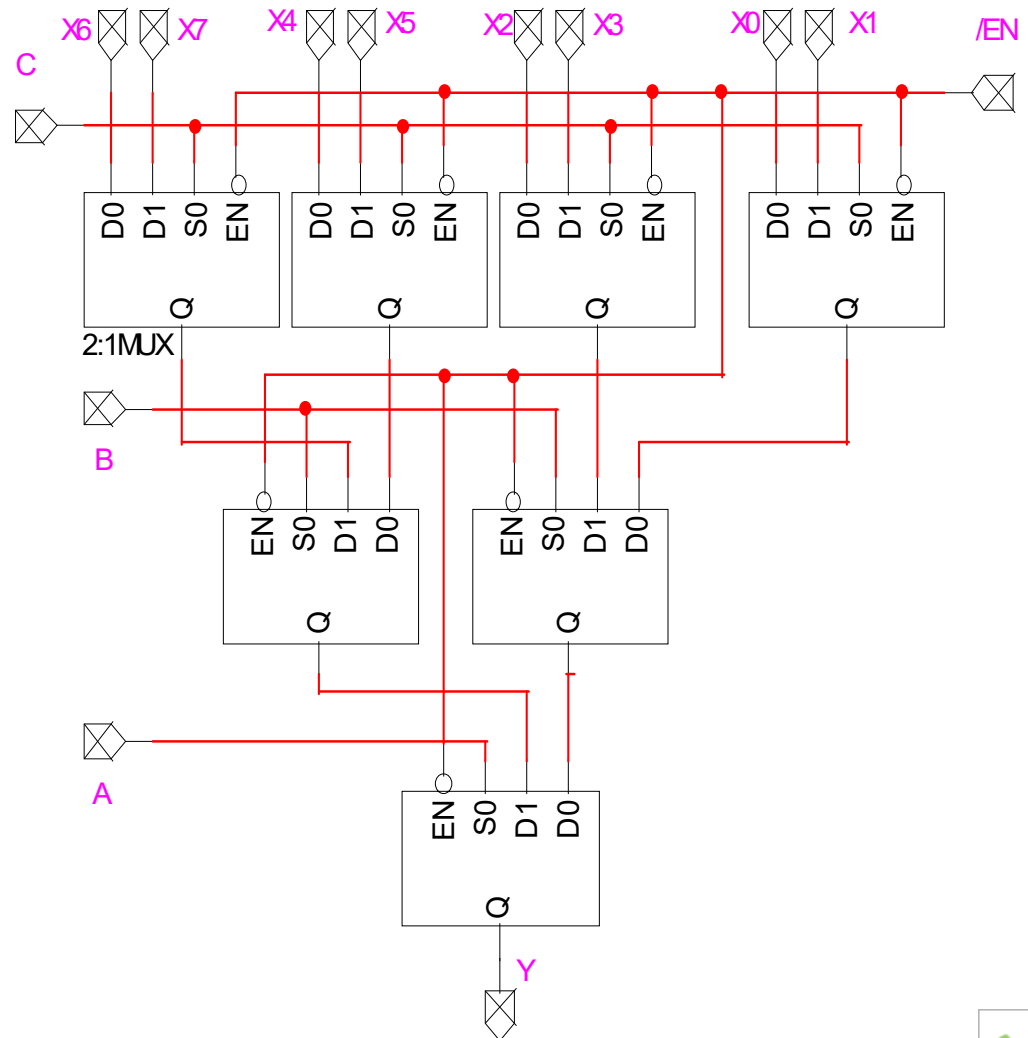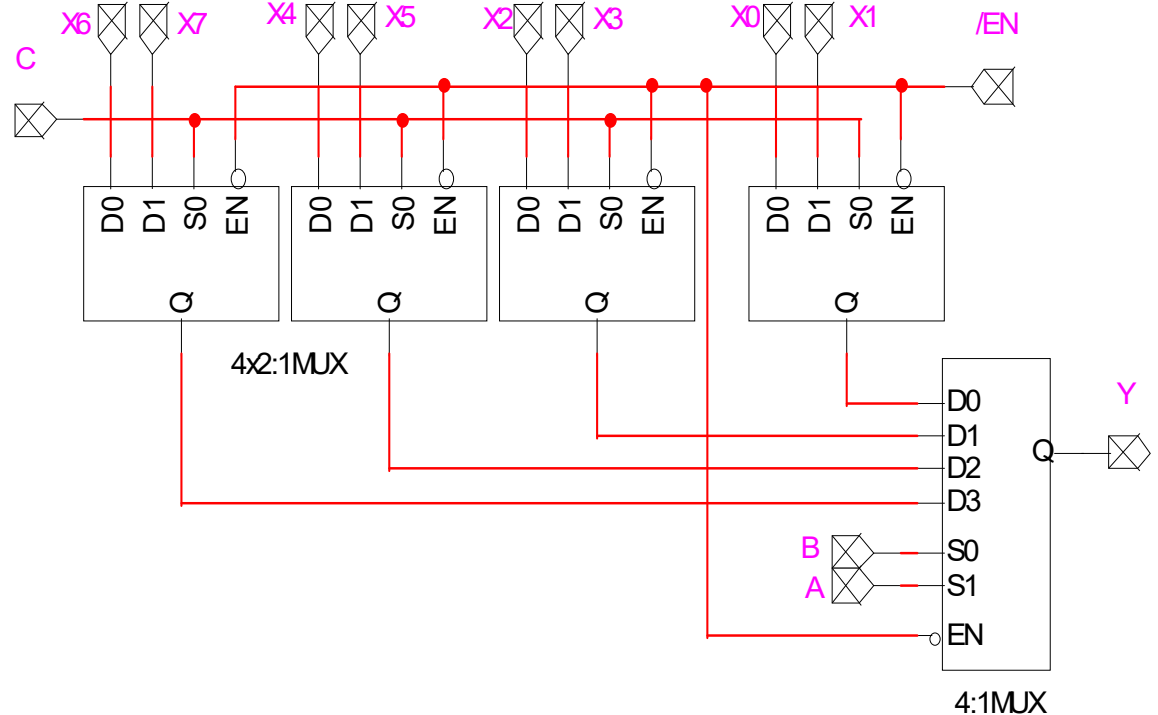
Mux hierarchies

- ## 8:1 with 2x(4:1 MUX) + 1x(2:1 MUX)

  - Always check the design by obtaining the truth table.

  - Note that IN THIS case A is the most significant selection variable

Mux hierarchies

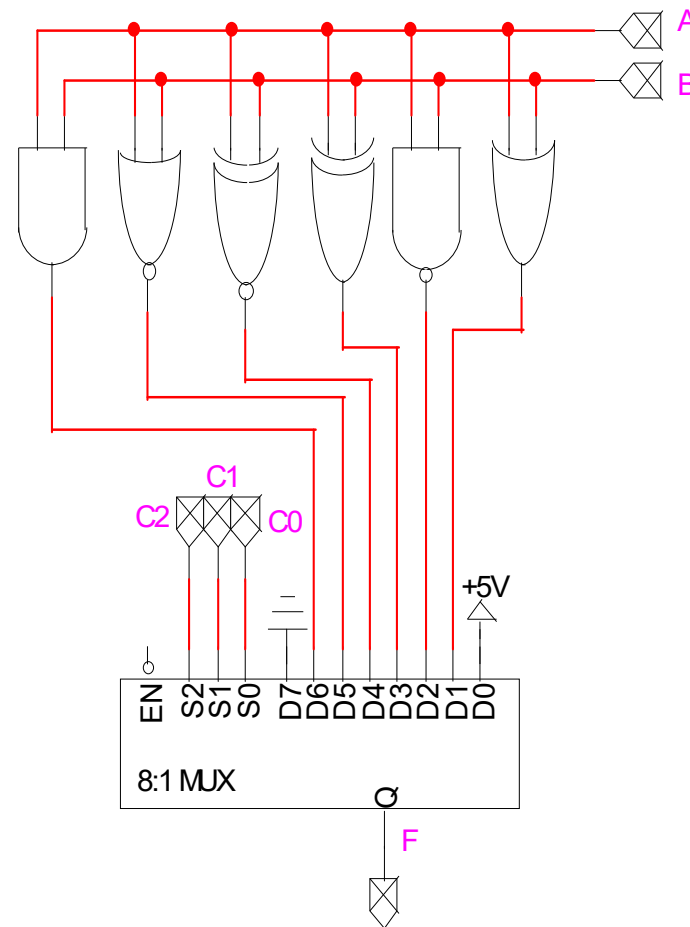- 8:1 with 4x(2:1 MUX) + 1x(4:1 MUX)

- Always check the design by obtaining the truth table.

- Note that IN THIS case A is the most significant selection variable
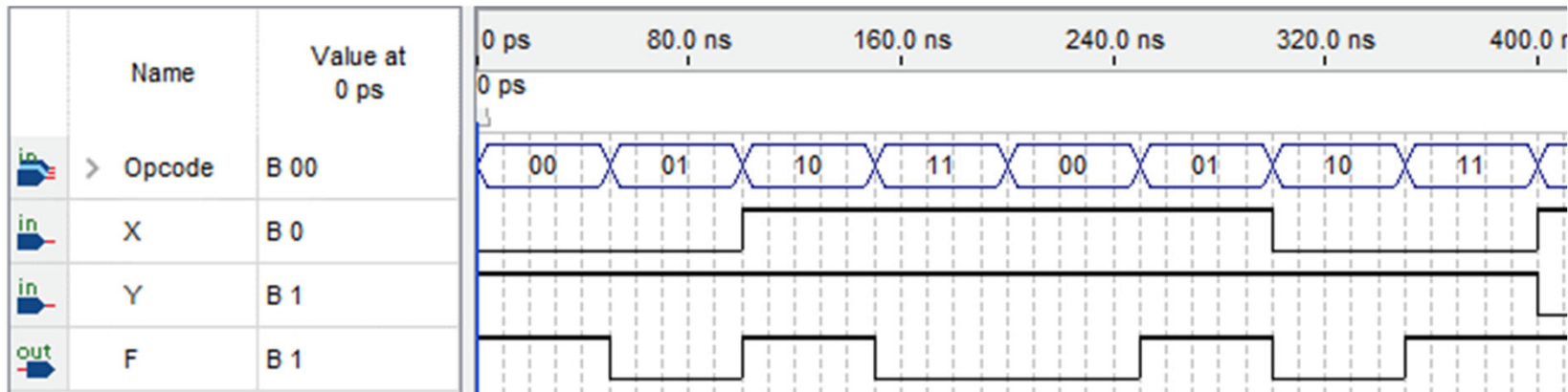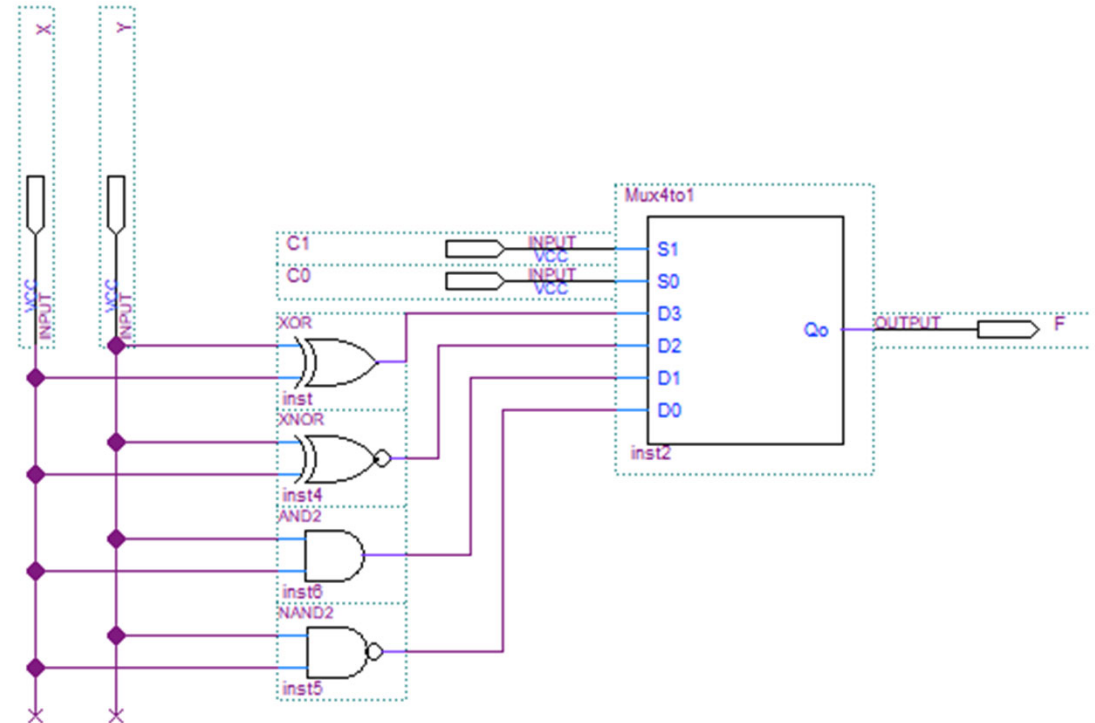
Logic Function Unit (LFU)

- Use $C_2 C_1 C_0$ as function code (Opcode)

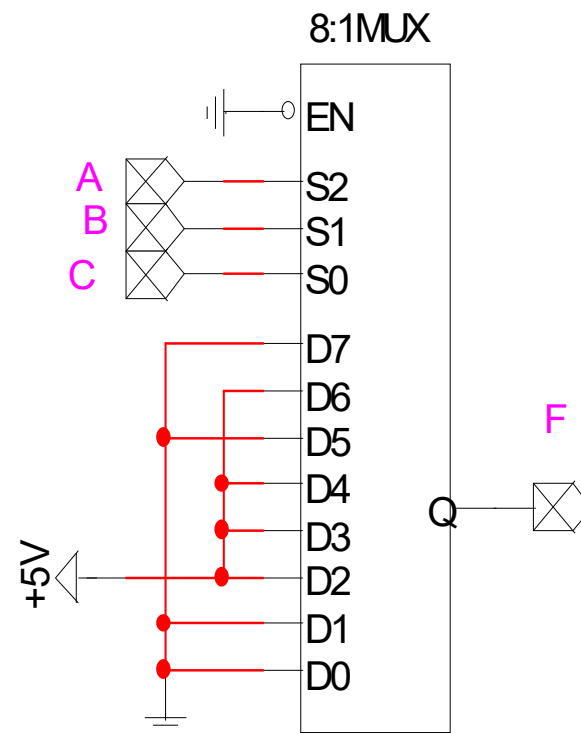| C_2	C_1	C_0	F
0	0	0	1
0	0	1	A+B
0	1	0	(A.B)'
0	1	1	A$\oplus$B
1	0	0	(A$\oplus$B)'
1	0	1	(A+B)'
1	1	0	A.B
1	1	1	0

Exercise

- Explain the LFU timing diagram

Boolean Functions with Multiplexers

- ## Simplest approach:
  - – Direct mapping of the Truth Table
    - Selection = input variables
    - $D_k = F_k$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

8:1MUX

EN
S2
S1
S0
D7
D6
D5
D4
D3
D2
D1
D0
Q
F
A
B
C
+5V

Boolean Functions with Multiplexers

- ## The general case:

  – Selection = a subset of input variables

  – $D_k = g_k$ where each g_k is a simpler Boolean function of the remaining input variables

- ## Example

n-1 input variables used for selection

I1	I2	..	In	F			
…	..	..	0	0	0	1	1
…	…	..	1	0	1	0	1
				0	In	$\overline{In}$	1

Possible output values as a function of I_n

Example

- Implement the Boolean function F(A,B,C,D) using a 8:1 Mux

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

1. Use the Karnaugh map JUST to layout the truth table

2. Choose the subset of inputs to be assigned to the mux selection inputs

   Eg. A,B,C

3. Find the logic values of the mux data inputs as functions of the remaining inputs

   D in this case

Regions of the truth sharing the same value of the selection inputs. (A,B,C) in this case. DO NOT MISINTERPRET as prime implicants
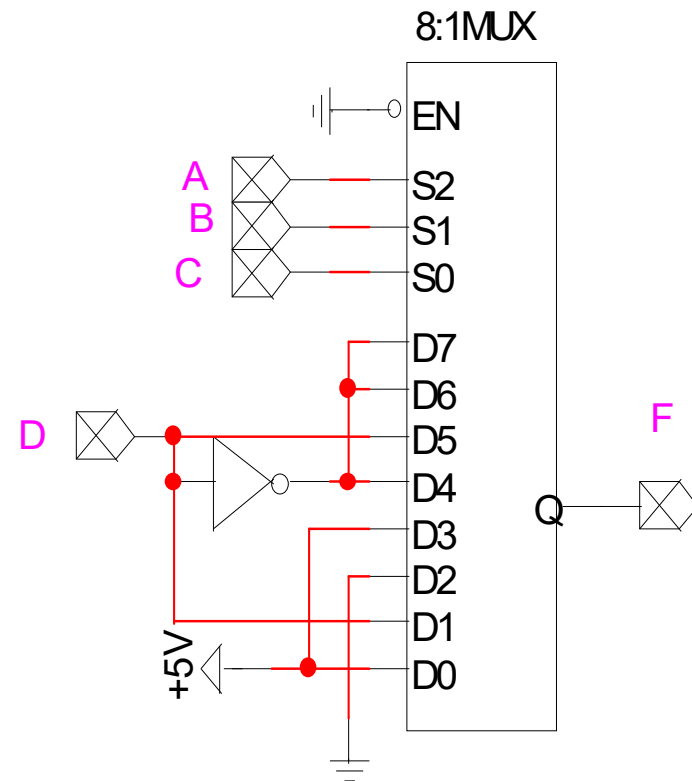
Example

- Find the error in the logic circuit

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

Exercise

- Implement the Boolean function *F* using a a MUX 4:1 and additional elementary logic

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$

- Several choices of input variables are possible to be assigned to the mux selection inputs. Try for example (A,B) and (C,D)
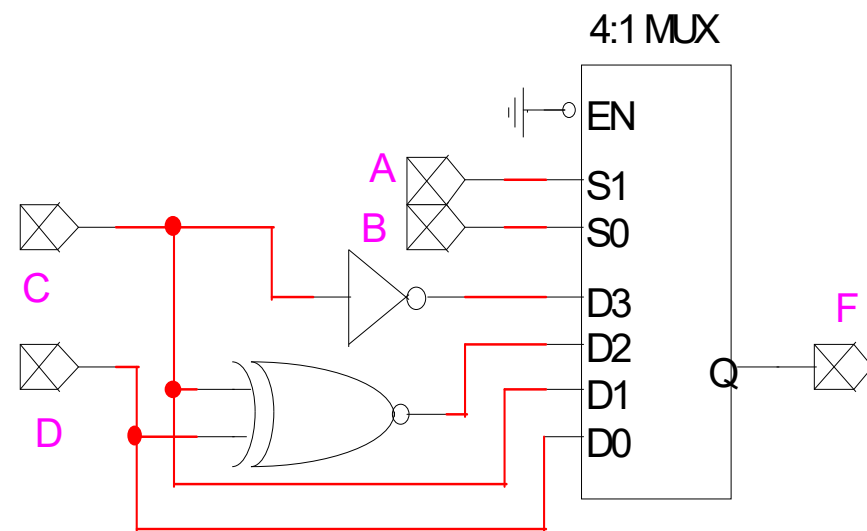
Exercise

- Using A,B for selection

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,13)$$

Exercise

- Using C,D for selection

$$F(A,B,C,D) = \sum m(1,3,6,7,8,11,12,14)$$
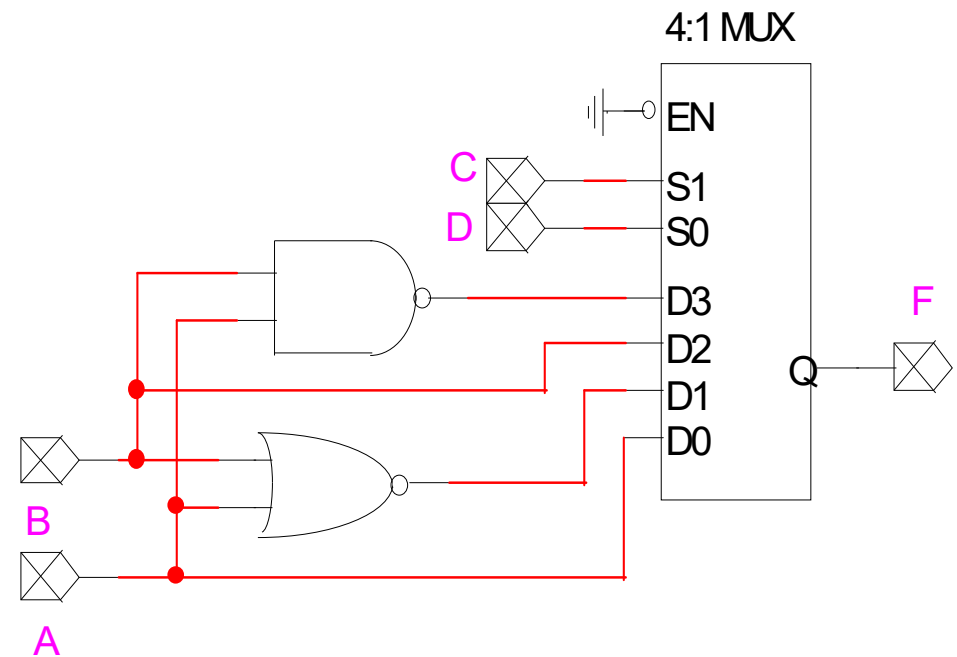
High-Impedance (High – Z)

- A switch model
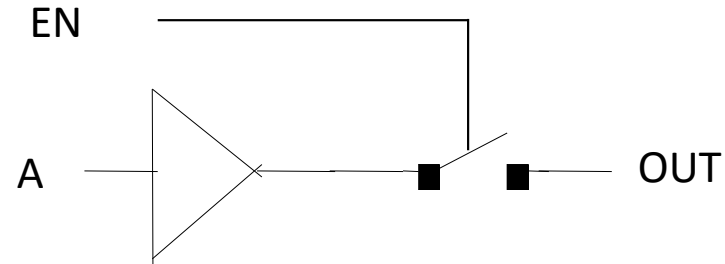
- When the switch is open there is an almost infinite resistance (Impedance) to the signal flow through the "wire" OUT.

- The output signal is left "floating" with neither HIGH or LOW logic levels assigned.

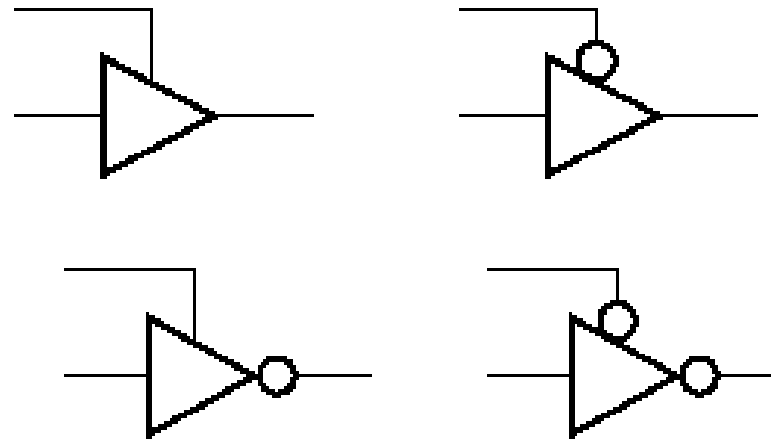- The output is assigned a High-Z state and the device exhibits a 3 State behavior

3 State Buffers
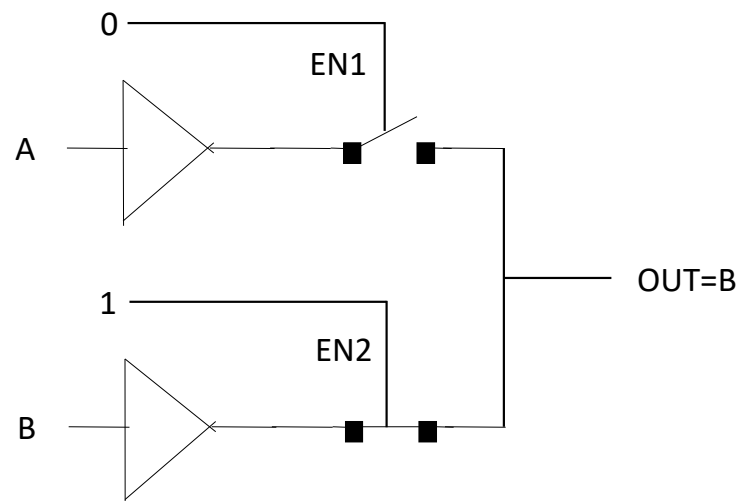
- Possible outputs: HIGH, LOW, High-Z

EN
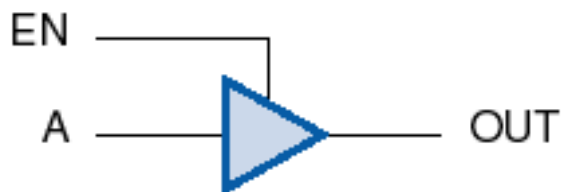
A ————— OUT

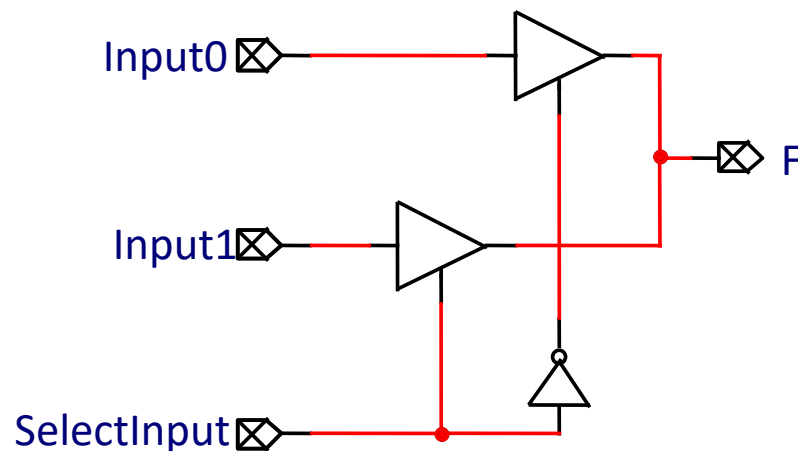| EN	A	OUT
L	L	Hi-Z
L	H	Hi-Z
H	L	L
H	H	H

Alternatives

Wire sharing

- High-Z outputs may be physically connected

- Of course EN1 = EN2 = 1 should never occour.
- Tight control of enabling inputs is required
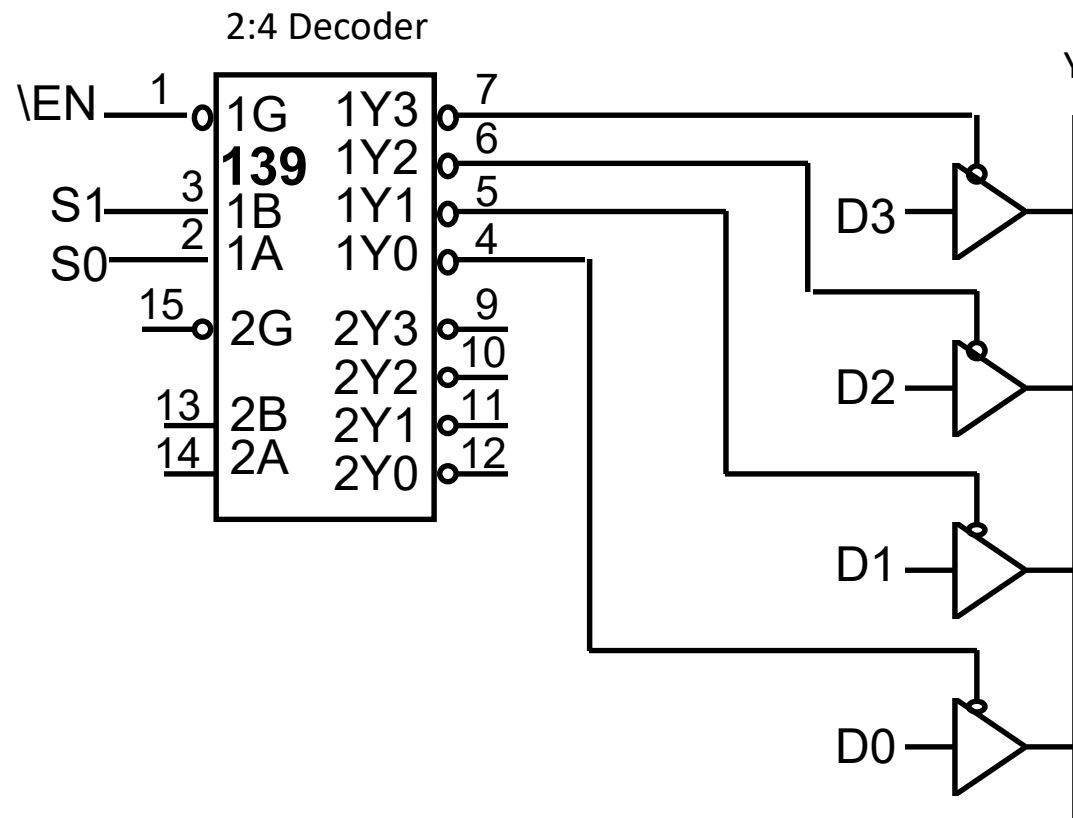
A special kind of Mux

- Efficient multiplexing strategy
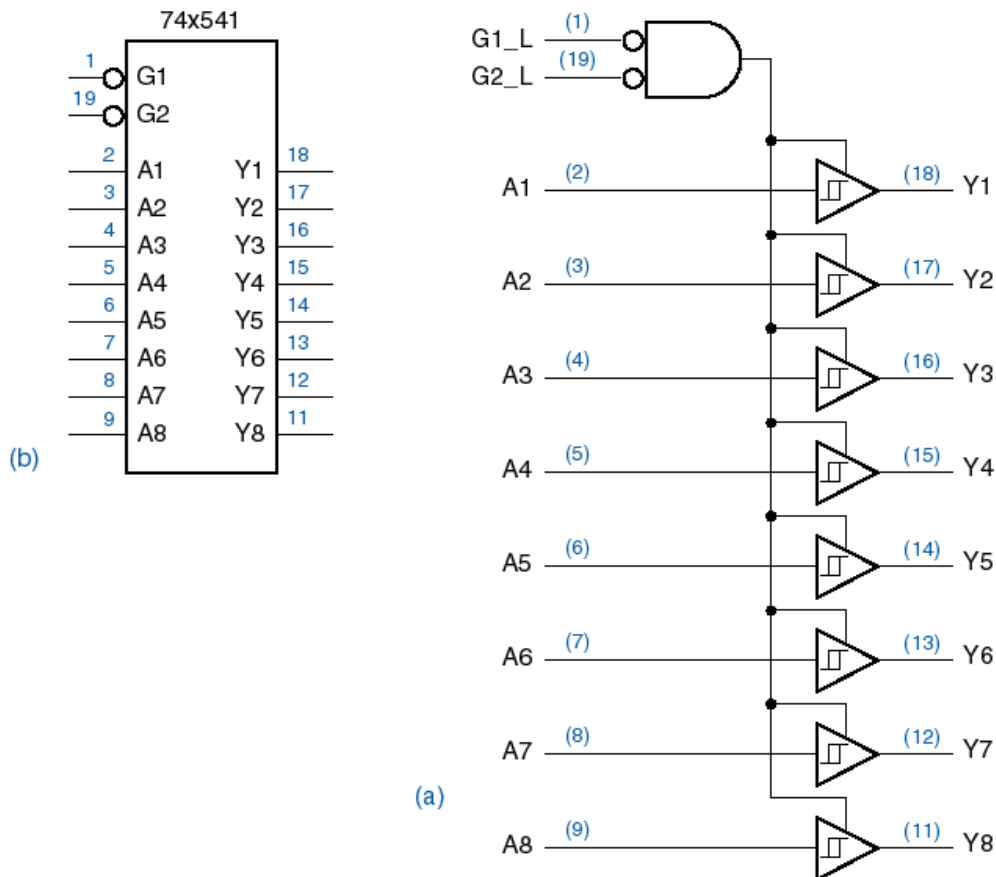- Mux 2:1

- Write the Truth Table

Exercise

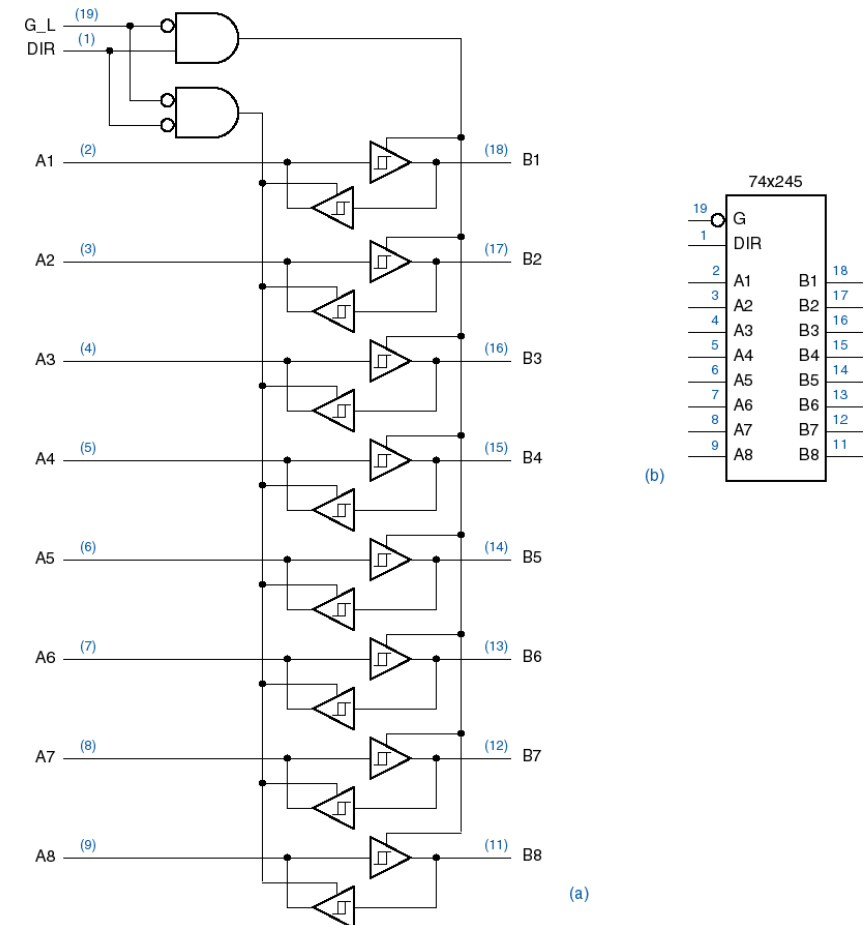- Write the truth table of the circuit and verify that's a 4:1 mux
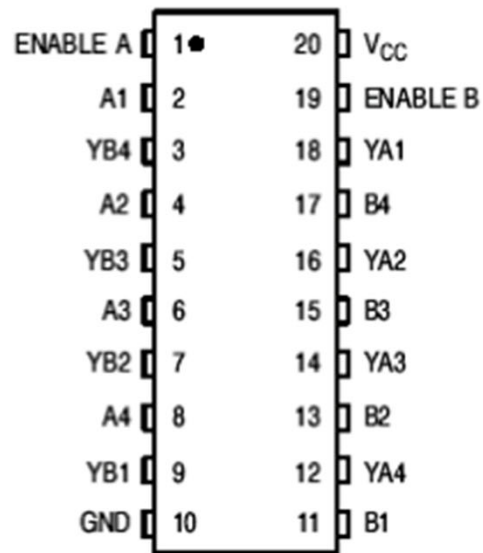
Aggregate 3 State Buffer Models

"BUS" Driver

Transceiver

Aggregate 3 State Buffer Models
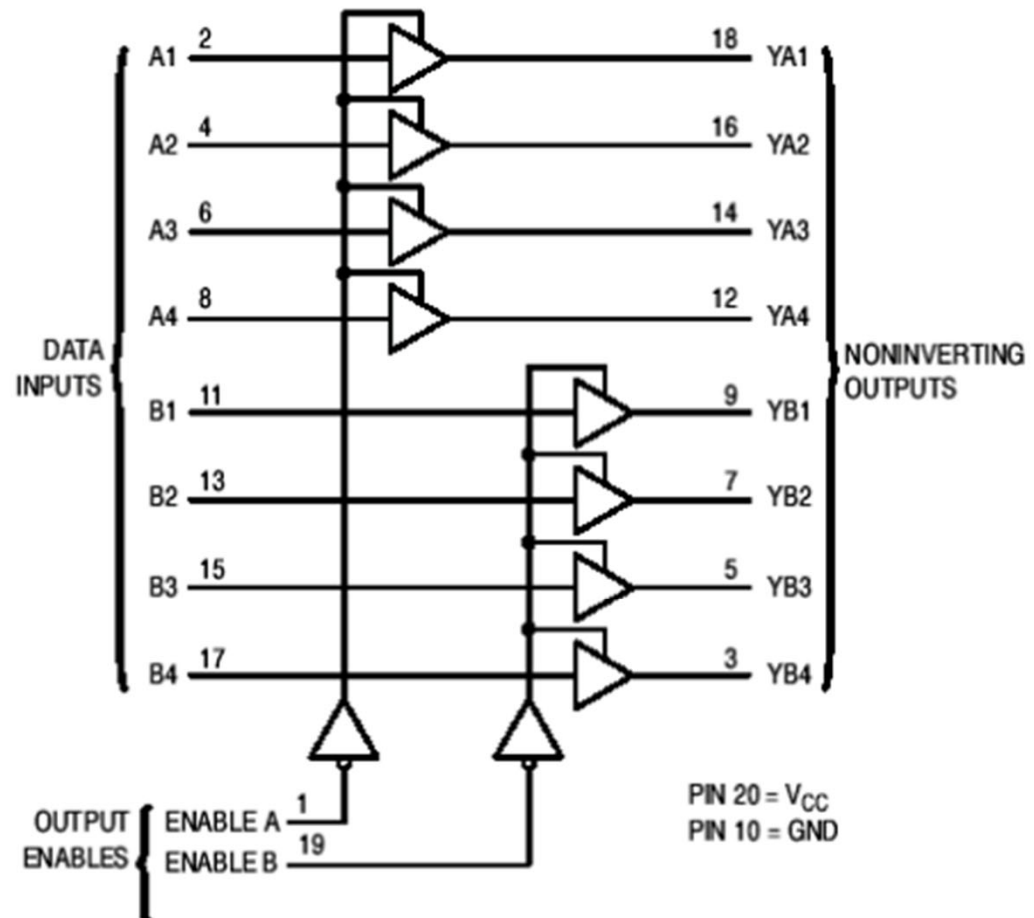

74HC244

PIN ASSIGNMENT

ENABLE A	1	20	V_CC
A1	2	19	ENABLE B
YB4	3	18	YA1
A2	4	17	B4
YB3	5	16	YA2
A3	6	15	B3
YB2	7	14	YA3
A4	8	13	B2
YB1	9	12	YA4
GND	10	11	B1

LOGIC DIAGRAM

FUNCTION TABLE

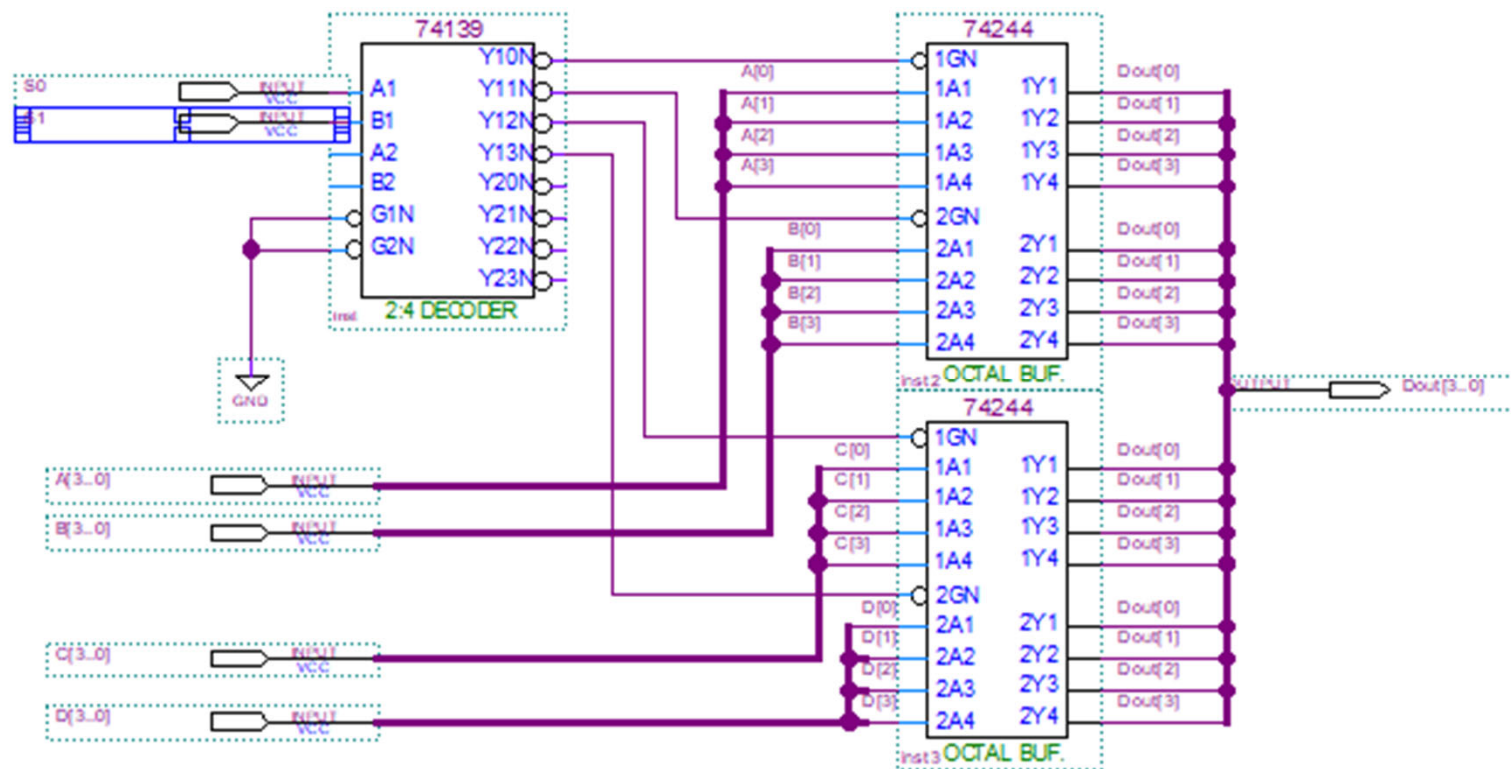| Inputs		Outputs
Enable A, Enable B	A, B	YA, YB
L	L	L
L	H	H
H	X	Z

Z = high impedance
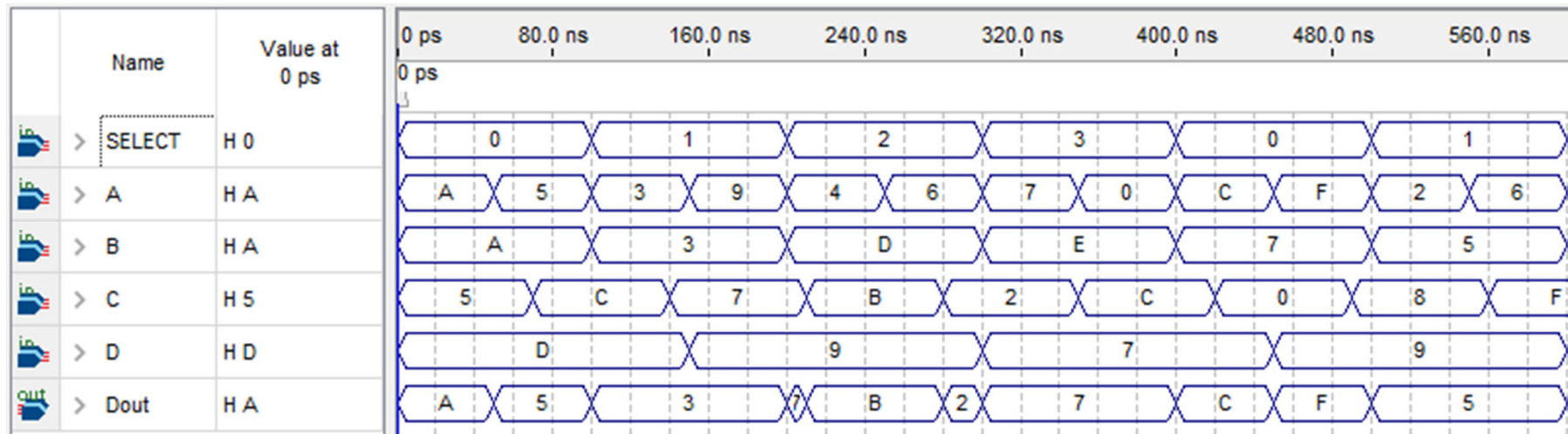
Word Multiplexing

- ## Main idea:

  – Decode the 3 State
    buffers enabling inputs

  – Share the output
    data bus

Exercise

- Explain the timing diagram of the previous circuit

  - SELECT = (S1,S0)

Final Remarks

- Always recall
  - The block symbol
  - The types of inputs and outputs
    - Data
    - Control
  - The truth table
  - The output equations
- Design with encapsulated logic requires mastering all the functional details of each block