

Combinational Logic Design I

Last class, we finished talking about decoders and started talking about encoders

Decoders

- only one output is active at any given time

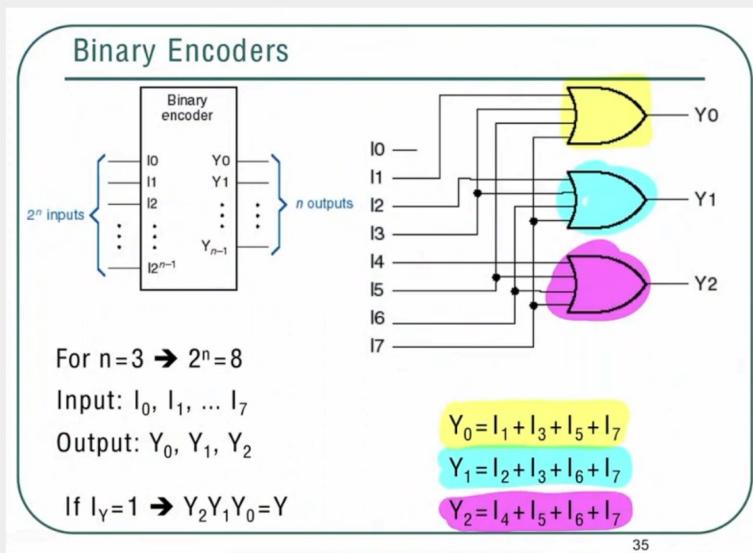
Encoders

- Only one input can be active at any given time
- the user has to take the responsibility to make sure only one input is active
- so establishes priority

We looked at 4.2 Basic Encoder

- We can look at what we did.
- With the row expressions H_0, H_1, H_2, H_3 we can make a logic diagram

2



We have 8 inputs corresponding to 3 outputs

Note, we can also chain encoders to make larger encoders

(we would need to add more combinational logic to our circuit)

And last time we also saw this

4:2 Priority Encoder				Enabled	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

Priority is given to most significant active input.

$Y_0 = H_1 + H_3$

$Y_1 = H_2 + H_3$

Row Expressions

$H_0 = \overline{I}_3 \cdot \overline{I}_2 \cdot \overline{I}_1 \cdot I_0$

$H_1 = \overline{I}_3 \cdot \overline{I}_2 \cdot I_1$

$H_2 = \overline{I}_3 \cdot I_2$

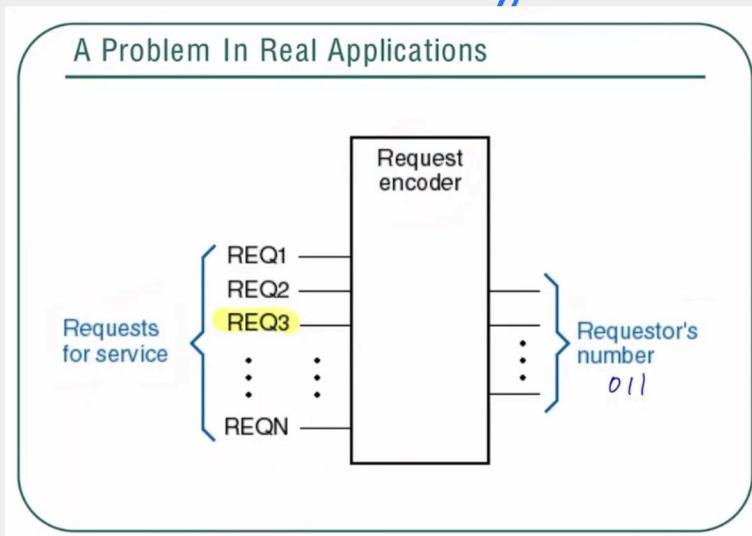
$H_3 = I_3$

ALL are Active high

To Build a Logic diagram, we first need to find the row expressions

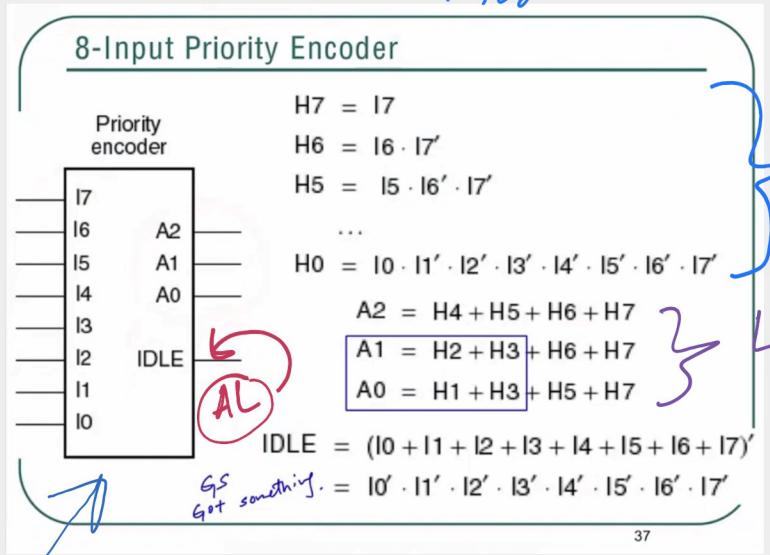
$$H_0, H_1, H_2, H_3$$

We also had an application like a request encoder



The requesters queue up for a number line

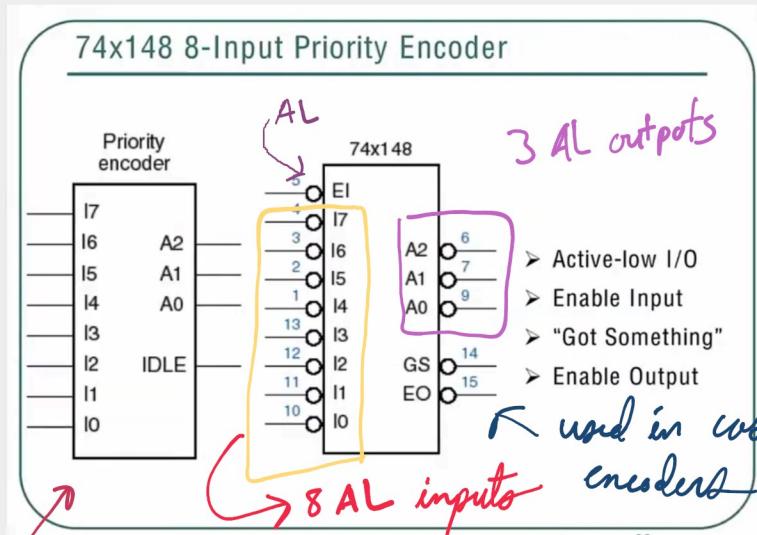
We also saw this



Here are our now expressions

Logic expressions for A_2, A_1, A_0

This is not our standard priority encoder
 The standard encoder is the 74x148 chip



This is from the previous slide

Notes

- we have 1 AL enable ($-EI$)
- we have 8 Active Low Inputs (I7 is Most significant)
- 3 AL outputs
- GS is our IDLE and is AL

- EO is Enable output (Active Low)
- GS and EO are basically used for Cascading

4 74x148 Truth Table

74x148 Truth Table													
Inputs								Outputs					
E _L	I ₀ _L	I ₁ _L	I ₂ _L	I ₃ _L	I ₄ _L	I ₅ _L	I ₆ _L	I ₇ _L	A ₂ _L	A ₁ _L	A ₀ _L	G _S _L	E _O _L
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

We have active Low inputs
 - we can clearly see that
 I₇_L is our most significant input (MS)

We can see that for E_I_L = 1, our enable is not active and so we get X's (don't cares). It doesn't matter what the input is, the mode is disabled.

- All outputs are disabled

G_S_L = 1 means we didn't get anything

E_O_L = 1 means don't enable the Enable Output

So when I₇_L is active (I₇_L = 0), all the outputs being active

$$A_2_L = 0$$

$$A_1_L = 0$$

$$A_0_L = 0$$

G_S_L = 0 (This means we did get something)

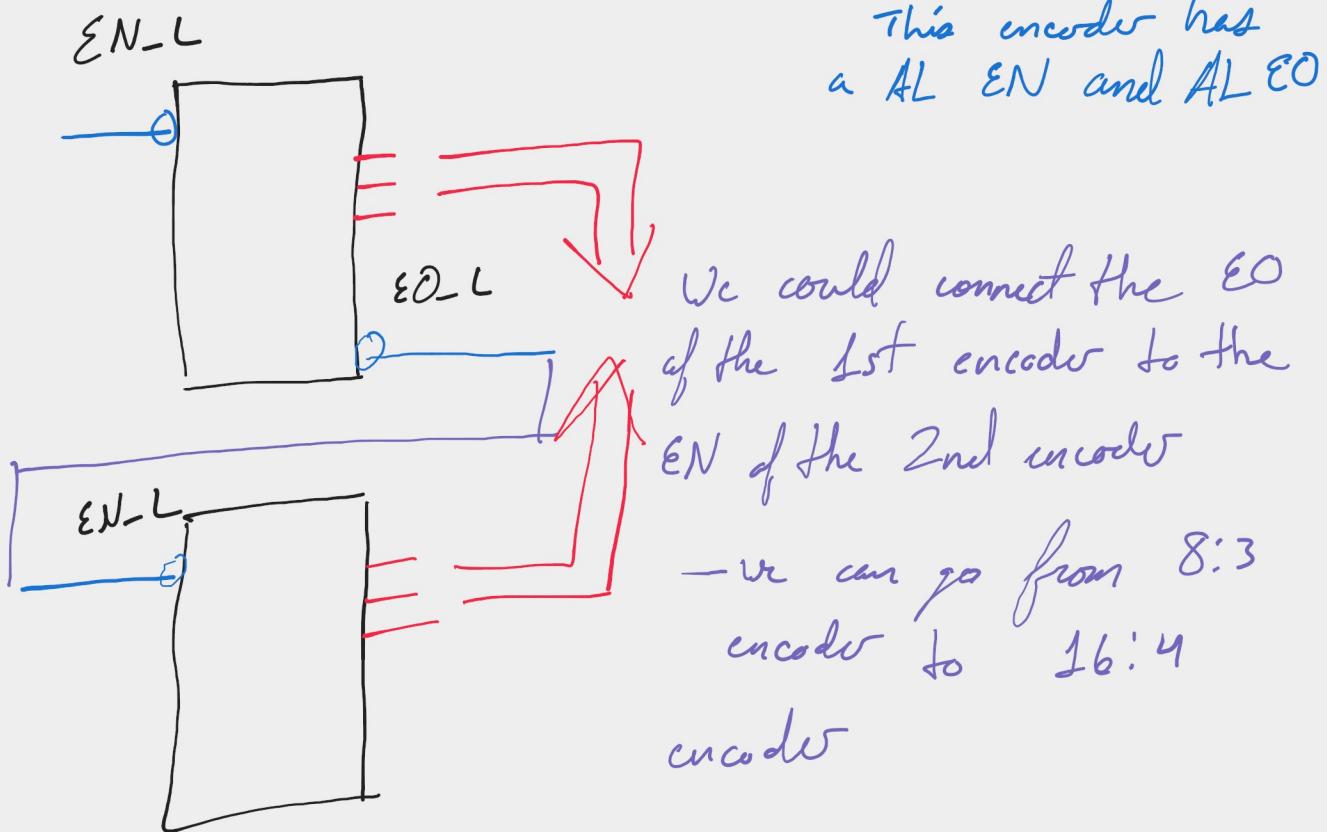
But E_O_L = 1 (we are not enabling the output)

What's EO-L ?

5

Imagine if we had to daisy chain (or cascade) our encoders

We have 2 8:3 encoders



Be careful, the outputs from the encoders would need to connect to some combinational circuit to get 4 outputs in a 16:4 daisy chain encoder configuration

So when $I7_L = 0$, $EO_L = 1$ because we don't want to enable the second encoder

6

If $I_6-L=0$, this is our highest priority bit

$$\text{So } A_2-L = 0$$

$$A_1-L = 0$$

$$A_0-L = 1$$

$$\text{and } G_S-L = 0$$

From the truth Table, we see that we get $G_S-L=0$

and $E_0-L=1$ (E_0 is Inactive)

→ for all these cases, we don't want to activate the second encoder

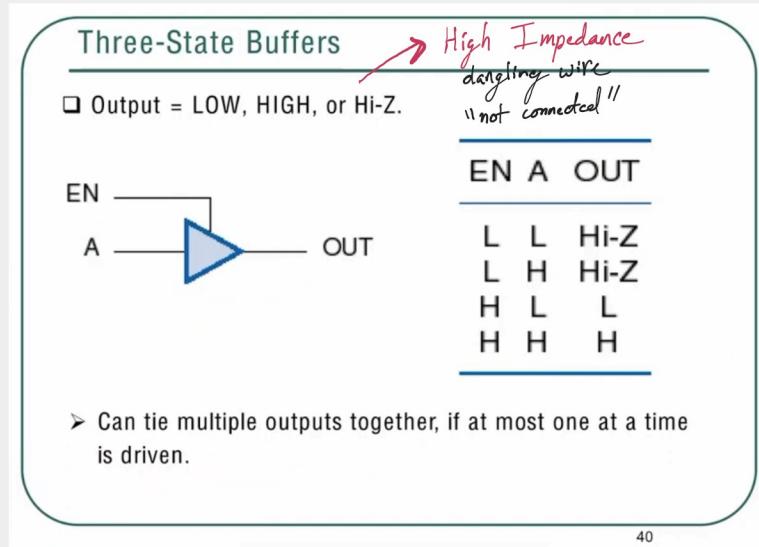
In the Last case, the chip is still enabled, we did not get any active input ($G_S-L=1$, Got Something is Inactive)

→ So we want to activate the next chip

With the truth table we can build Logic Expressions
Logic diagrams, Simplist expression

Tri-State Buffers / Three-state Buffers

7



The point of the Tri-state Buffer is to tie multiple outputs together

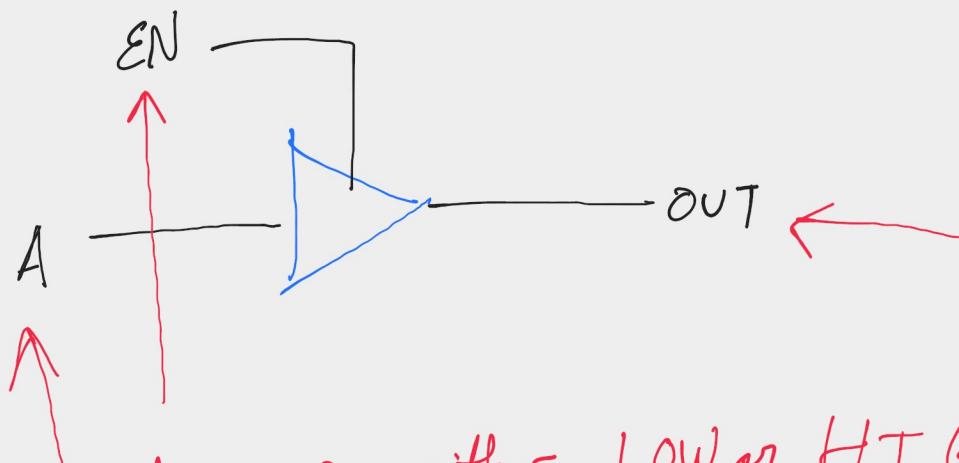
So what does the tri-state Buffer allow us to do?

The output of any logic function is either going to be LOW or HIGH

We are now going to add one more type of possible output, High Impedance (Hi-Z)

High Impedance: A dangling wire

The Buffer is call "Tri-state" because we have 3 states: LOW, HIGH, Hi-Z



output can be
in the Hi-Z as
well as LOW, HIGH

So if EN is 0 (Inactive)

→ Tri-state Buffer is disabled

→ Hi-Z state

- It's as if the output is disconnected to the input.

- So $A = X$ (A becomes a don't care)

If $EN = 1$ (EN Active)

- OUTPUT is the same as the Input



This is helpful in tying multiple outputs together

9

Different Flavors of Tri-state Buffers



AH enable
Buffer

AL enable
Buffer



AH enable
Inverter



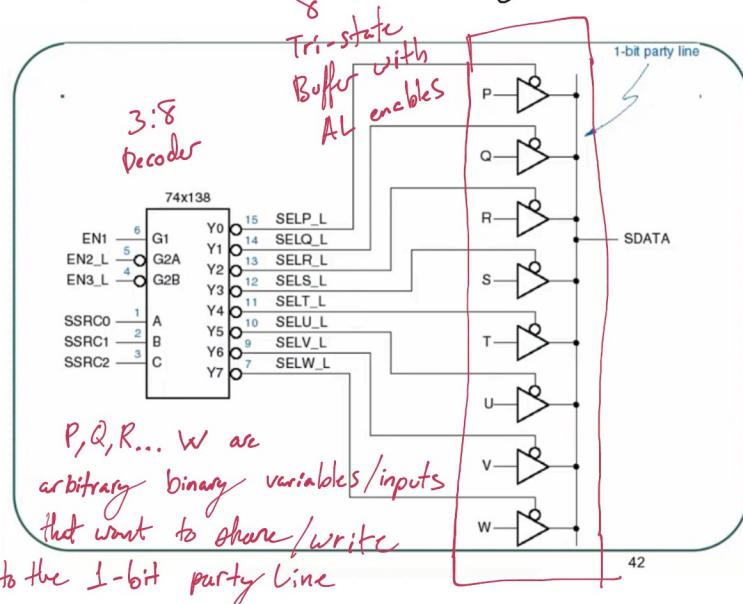
AL enable
Inverter



EN-L	A	Out-L
L	L	H
L	H	L
H	L	Hi-Z
H	H	Hi-Z

10

How we can use Tri state Buffers to Tie multiple outputs together?



So what's going on?

We can see that the Decoder outputs are controlling the Enables of the tri-state buffers

The outputs of the Tri-state Buffers are Tied together on a 1-bit party Line called serial Data (SDATA)

So what could Be the problem here?

When you are Tie-ing (Tying) multiple outputs together, don't you see some potential problems here?

→ If one output is saying 0 and another output is saying 1, That would be a short circuit between the source and the ground

- The chips would be damaged,
- You cannot tie a high to a low directly

To make sure this never happens, we use a decoder.

→ The decoder makes sure that only 1 tri-state buffer is active at any one given time

P, Q, R ... W are arbitrary binary variables / inputs that want to share/write to the 1-bit parity line.

P, Q, R ... W all want to control SDATA
So Let's understand what is happening

We have 3 enables for the 3:8 decoder, 1 is A_H
and 2 are A_L

— Let's assume everything is enabled

We have 3 inputs (A is LS input) (C is MS input)

12

Ex) Let's say we have input

$$SSRC0 = 1$$

$$SSRC1 = 1$$

$$SSRC2 = 0$$

Only One output pin will be active. Which pin?

→ pin 12 will be active (Y_3)

011 corresponds to decimal 3 (3_{10})

This activates the S connected Tri-state Buffer

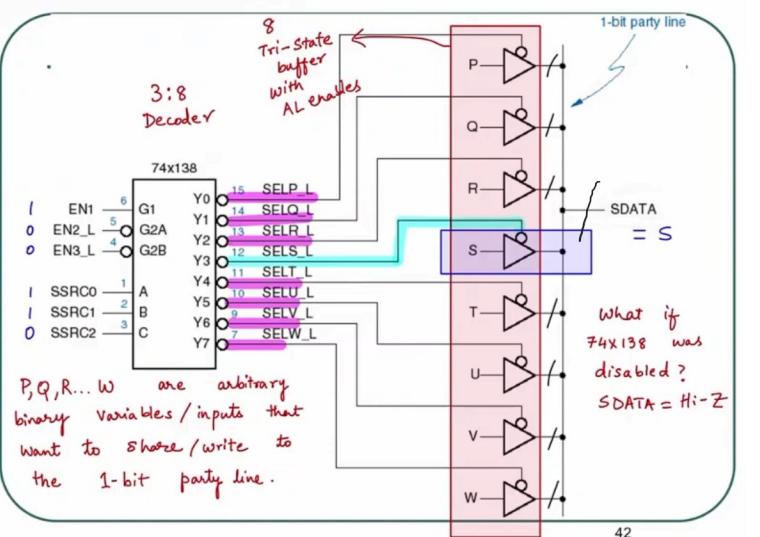
So All the other Tri-state Buffers are in the Hi-Z state

(That effectively makes them disconnected from the SDATA line)

So Now, $SDATA = S$

Ex) What if 74x138 was disabled? What would be SDATA?

→ $SDATA = \text{Hi-Z}$

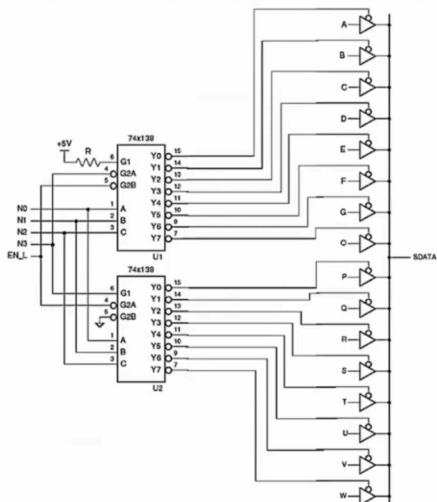


So when would

$SDATA = \text{Hi-Z}$ become useful?

→ It would be useful if we were making a bigger diagram

→ If we have multiple 74×138 chips we can disable multiple groups of chips



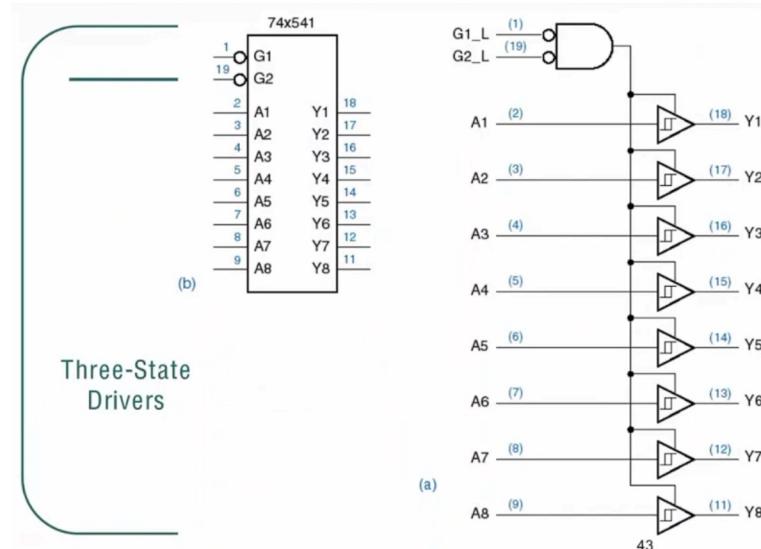
This is an example of multiple decoders and Tri-state Buffers and SDATA

The students were asked what would SDATA be when given EN-L, N3, N2, N1 and NO

EN_L	N3	N2	N1	NO	SDATA
1	0	0	1	0	Hi-Z
0	0	0	1	0	C
0	0	1	1	1	O
0	1	0	1	0	R
0	1	0	1	0	R
0	0	1	0	0	E
0	0	0	1	0	C
0	1	1	0	0	T

14

Tri-State Drivers



Here are Tri-state Buffers
In a chip

— This is the chip w/
the internal logic diagram

We have AL EN that are connected to tri-state
Buffers thru the AND Gate

schmitt

Tri-state Buffers are following a ~~Schmitt~~-trigger

Process

→ 2 thresholds for rising and falling

Signals

when you get to 70ft
you are up the
hill. If you then
go down to 30ft, you are still up the hill.

.. 70ft

30ft

Schmitt-Trigger are a configuration to counter noise.

Obviously, we have 8 AH inputs, 8 AH outputs 15

so when Active $Y_k = A_k \quad k \in \{1, 2, \dots, 8\}$

when enabled

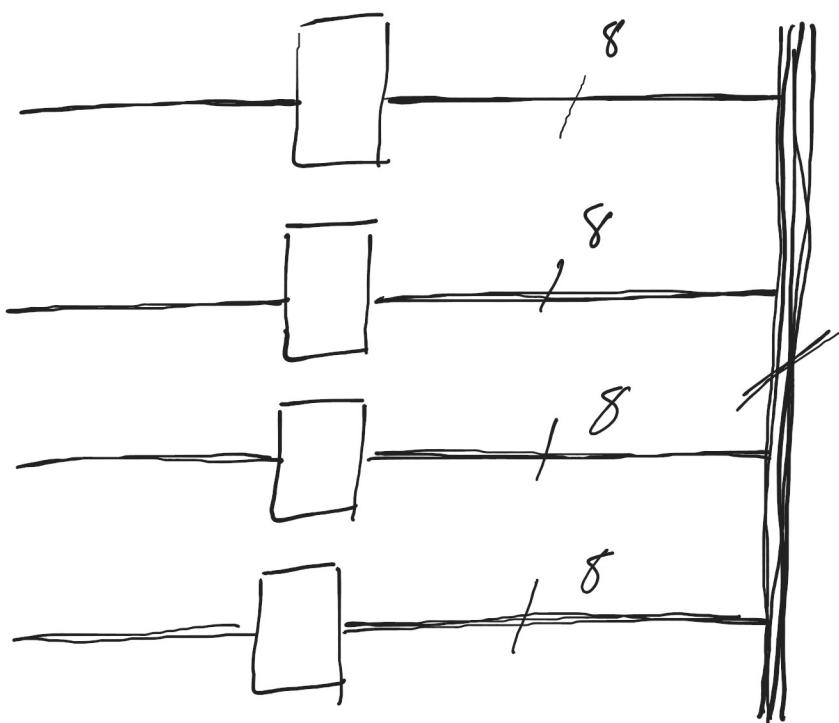
so what happens when disabled?

$Y_k = H_i - Z$

when disabled

A Practice example

74x541 chips



What's going on?

We have 4 chips

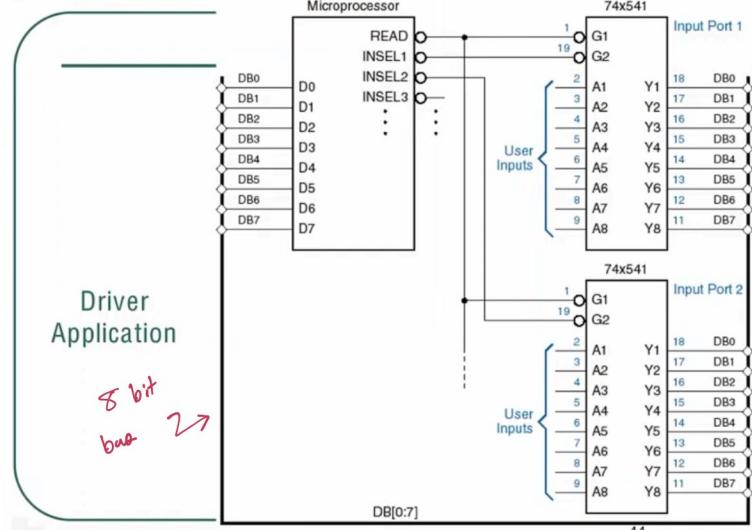
We can tie multiple
8 bit outputs together
on an 8 bit bus

8 bit bus

EN can be used to
Daisy chain

16

How to Apply 74x541



We have a microprocessor
Trying to read from one of the
User-input ports

We are trying to read from input port 1 or input port 2.

Where are we trying to read that data?

- We are trying to read that data into a
microprocessor

So we have 2 sets of user inputs

The question now is that, Can I give out some control signals from the microprocessor to read user input port 1 or user input port 2.

If I want to read, I have to make the read signal active.

If I want to read from input port 1, I have to activate my input-select-1 (INSEL1) from the microprocessor.

Consequently, the other input selects are
Inactive . INSEL2 is Inactive

→ G1 is active from READ
G2 is Inactive from INSEL2

Only the Top 74x541 chip is active and so

The User Inputs $A_k = Y_k$

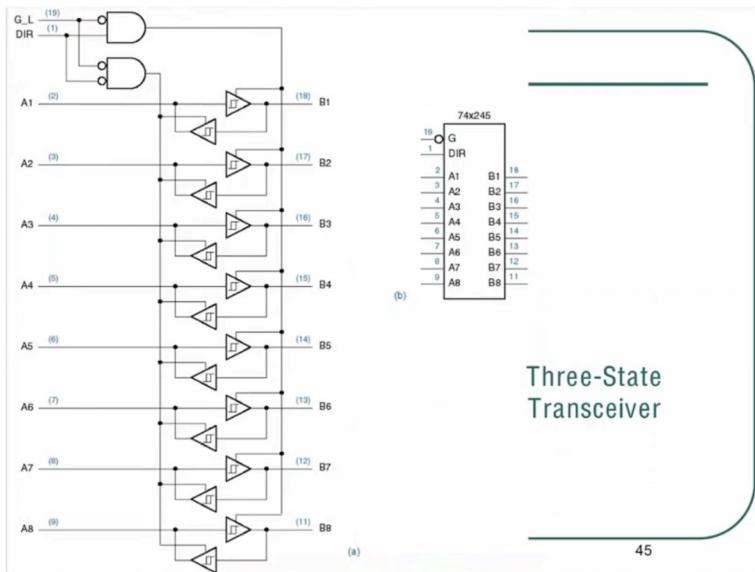
We have an 8 bit Bus connected to The Top chip. This Bus takes the User Inputs and provides them as inputs to the microprocessor.

Therefore Input port L has been read

If you want to activate the Second chip, you must deactivate the first chip. $\text{INSEL1} = \text{Inactive}$
 $\text{INSEL2} = \text{Active}$

Our bus cannot take in multiple Input ports.

18 Tri-State Transmitter - Receiver



- We have 8 inputs and 8 outputs (call A+B)
- We have an AL enable, $G_L = \neg \bar{G}$

DIR tells us which direction things are going.

$A \rightarrow B$ or $B \rightarrow A$

If $DIR = 0$, The Schmitt-trigger tri-state Buffers pointing to the Right are Deactivated

$Dir = 0$
 $B_K \rightarrow A_K$

$Dir = 1$
 $A_K \rightarrow B_K$