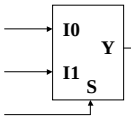
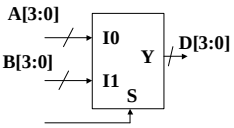
Combinational Building Blocks

2/1 Multiplexor (MUX)



if S = 0, then Y = I0
if S = 1, then Y = I1

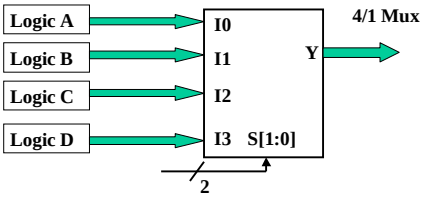
Y = I0 S' + I1 S



Muxes often use to select groups of bits arranged in busses.

What Good are Muxes ??

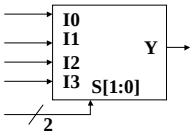
Sometimes want to have a bus be driven from multiple blocks, where only one block is driving the bus at a time.



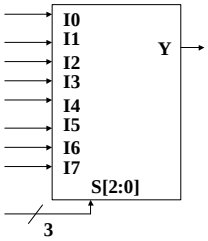
N to 1 mux will select 1 source; Select bus needs to be log2(N). Note that only ONE input can be selected at a time!

Higher Order Muxes

4/1 Mux



8/1 Mux

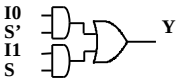


if S = "00", then Y = I0
if S = "01", then Y = I1
if S = "10", then Y = I2
if S = "11", then Y = I3

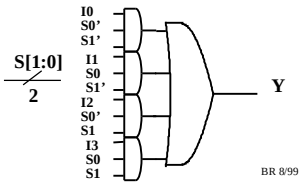
Y = I0 S1' S0' + I1 S1' S0 + I2 S1 S0' + I3 S1 S0

Logic for 2/1, 4/1 Muxes

Y = I0 S' + I1 S



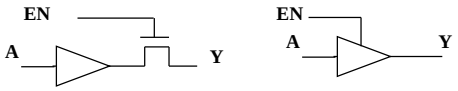
Y = I0 S1' S0' + I1 S1' S0 + I2 S1 S0' + I3 S1 S0



These are called COMBINATIONAL muxes.

Tri State Buffer

There is another way to drive a line or bus from multiple sources. Use a TRISTATE buffer.



When EN = 1, then Y = A.

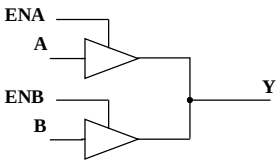
When EN = 0, then Y = ??????
Y is *undriven*, this is called the high impedance state.

Designate high impedance by a 'Z'.

When EN = 0, then Y = 'Z' (high impedance)

Using TriState Buffers

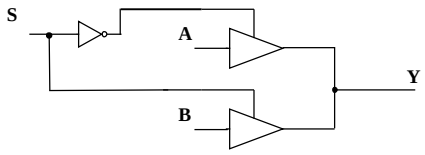
Can use tristate buffers instead of a combinational 2/1 mux



Must make sure that ENA, ENB are not both '1' at same time, or Y will be driven from multiple sources

Using TriState Buffers (cont)

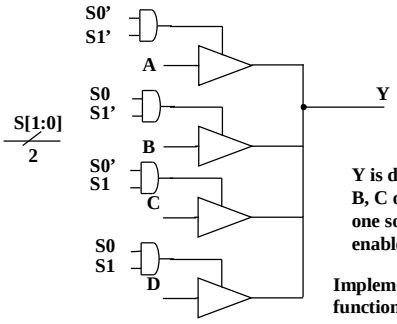
Only A or B is enabled at a time.



Implements 2/1 Mux function

If S=0 then Y = A
If S=1 then Y = B

Using TriState Buffers (cont)



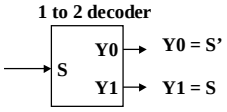
Y is driven by A, B, C or D; only one source is enabled at a time.

Implements 4/1 MUX function.

Decoders

Logic common to both of the previous mux implementations was the *decoder* function.

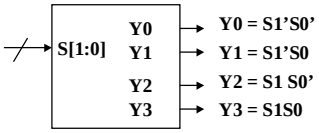
1 to 2 decoder



Y0 = S'

Y1 = S

2 to 4 decoder



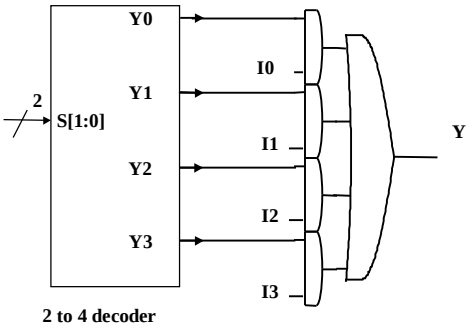
Y0 = S1'S0'

Y1 = S1'S0

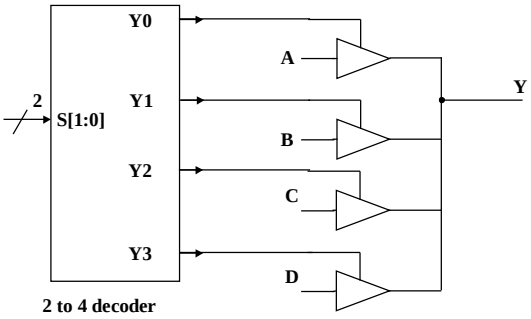
Y2 = S1 S0'

Y3 = S1S0

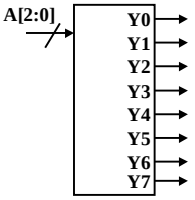
Logic for 4/1 Combinational Mux



4/1 Mux using Decoder + TSBs

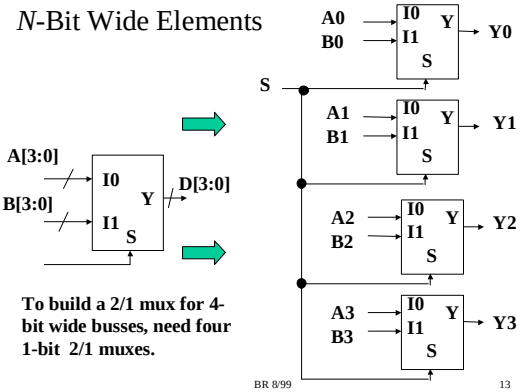


3 to 8 Decoder



if A= 000 then Y0=1 else Y0=0;
if A= 001 then Y1=1 else Y1=0;
if A= 010 then Y2=1 else Y2=0;
if A= 011 then Y3=1 else Y3=0;
if A= 100 then Y4=1 else Y4=0;
if A= 101 then Y5=1 else Y5=0;
if A= 110 then Y6=1 else Y6=0;
if A= 111 then Y7=1 else Y7=0;

N-Bit Wide Elements



To build a 2/1 mux for 4-bit wide busses, need four 1-bit 2/1 muxes.

Review of Memory example

F (A,B,C) = A xor B xor C G = AB + AC + BC

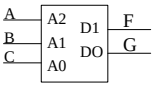
A	B	C	F	G
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Recall that Exclusive OR (xor) is

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Y = A⊕B
= A xor B

8 x 2 Memory



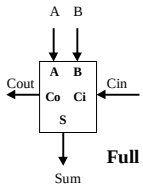
LookUp Table (LUT)

A[2:0] is 3 bit address bus, D[1:0] is 2 bit output bus.
Location 0 has "00",
Location 1 has "10",
Location 2 has "10",
etc….

Binary Adder

F (A,B,C) = A xor B xor C G = AB + AC + BC

These equations look familiar. These define a *Binary Full Adder* :

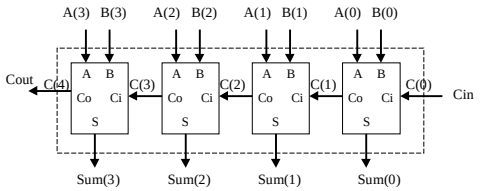


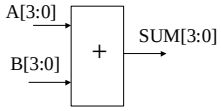
Sum = A xor B xor Cin

Cout = AB + Cin A + Cin B
= AB + Cin (A + B)

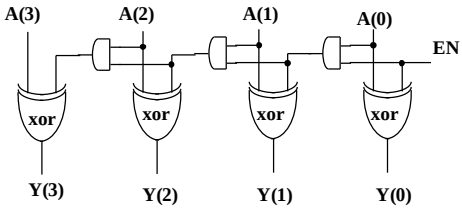
Full Adder (FA)

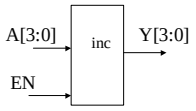
4 Bit Ripple Carry Adder





Incrementer





If EN = 1 then Y = A + 1
If EN = 0 then Y = A

What do You have to know?

- Structures for Muxes, Decoders, Ripple Carry adder, Incrementer
- What a tristate buffer is
- How to build muxes from all combinational logic or from combinational logic + tristate buffers
- Bus naming convention
- How to build N-bit wide elements from 1-bit wide elements