

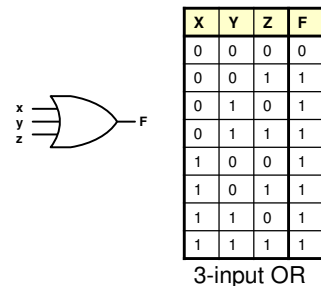
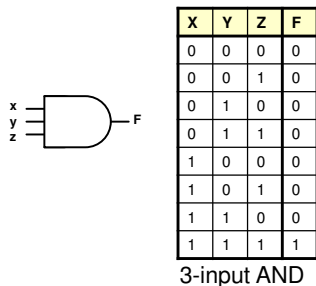
Unit 7

Fundamental Digital Building Blocks: Decoders & Multiplexers

CHECKERS / DECODERS

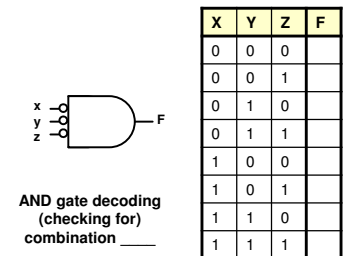
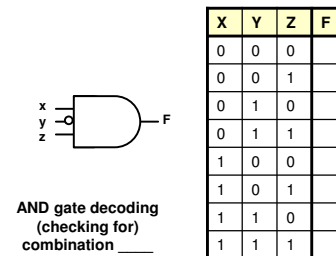
Gates

- Gates can have more than 2 inputs but the functions stay the same
 - AND = output = 1 if ALL inputs are 1
 - Outputs 1 for only 1 input combination
 - OR = output = 1 if ANY input is 1
 - Outputs 0 for only 1 input combination



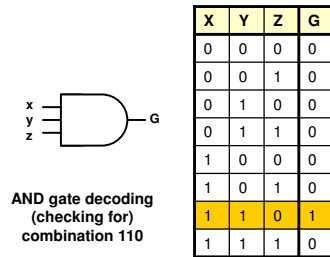
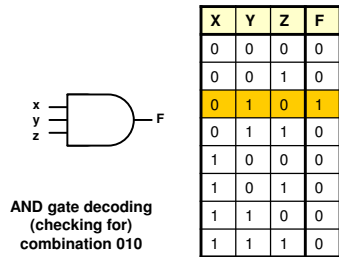
Checkers / Decoders

- An AND gate only outputs '1' for 1 combination
 - That combination can be changed by adding inverters to the inputs
 - We can think of the AND gate as "checking" or "decoding" a specific combination and outputting a '1' when it matches.



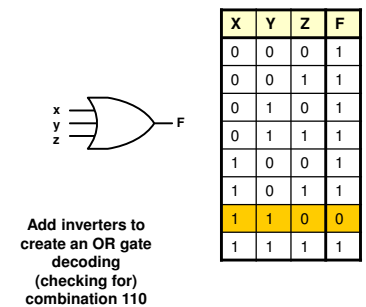
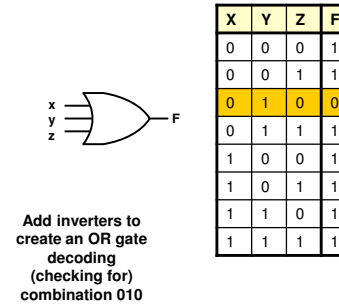
Checkers / Decoders

- Place inverters at the input of the AND gates such that
 - F produces '1' only for input combination $\{x,y,z\} = \{010\}$
 - G produces '1' only for input combination $\{x,y,z\} = \{110\}$



Checkers / Decoders

- An OR gate only outputs '0' for 1 combination
 - That combination can be changed by adding inverters to the inputs



Decoder Exercise

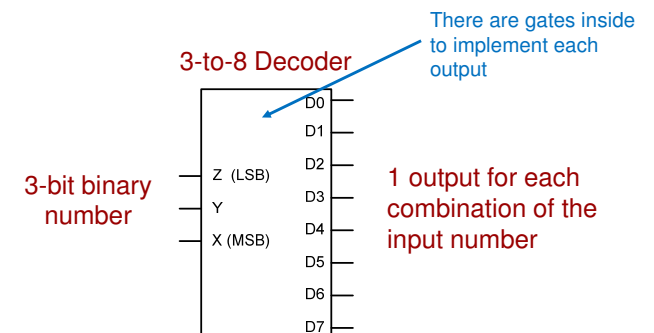
- Compilers translate software to instructions that tell the processor to ADD, LOAD from Memory, Store to Memory, etc.
- These instructions are binary codes
- The processor must decode the instruction
- Create an AND gate decoder for each instruction type in the table that will produce '1' when that instruction is about to be executed

Instruction Type	6-bit OPCODE OP[5:0]
ADD	001000
LOAD	100011
STORE	101011
BRANCH	000100

_____ADD_____ _____LOAD_____ _____STORE_____

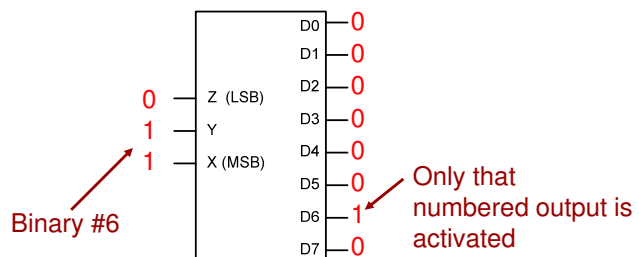
Full Decoders

- A full decoder is a building block that:
 - Takes in an n-bit binary number as input
 - Decodes that binary number and activates the corresponding output
 - Individual outputs for _____ input combinations



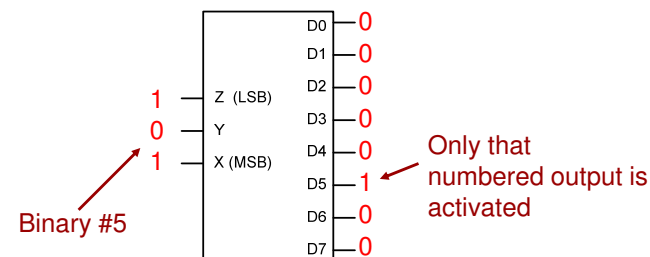
Decoders

- A decoder is a building block that:
 - Takes a binary number as input
 - Decodes that binary number and activates the corresponding output
 - Put in 6=110, Output 6 activates ('1')
 - Put in 5=101, Output 5 activates ('1')



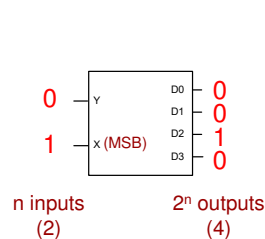
Decoders

- A decoder is a building block that:
 - Takes a binary number as input
 - Decodes that binary number and activates the corresponding output
 - Put in 6=110, Output 6 activates ('1')
 - Put in 5=101, Output 5 activates ('1')

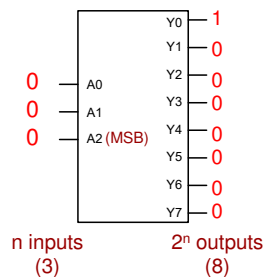


Decoder Sizes

- A decoder w/ an **n-bit input** has **2ⁿ outputs**
 - 1 output for every combination of the n-bit input



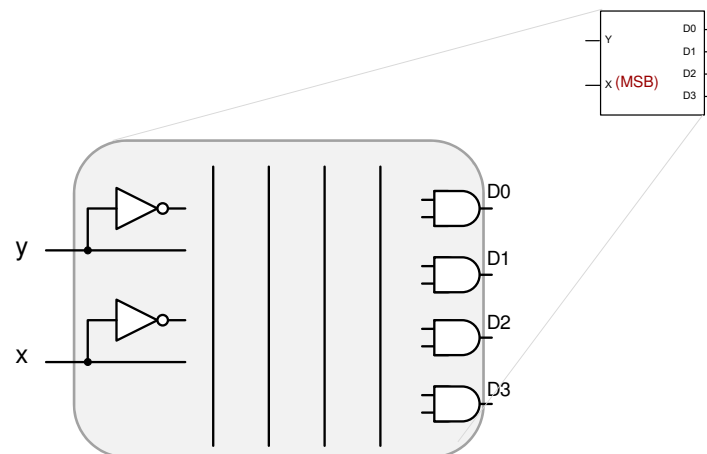
2-to-4 Decoder



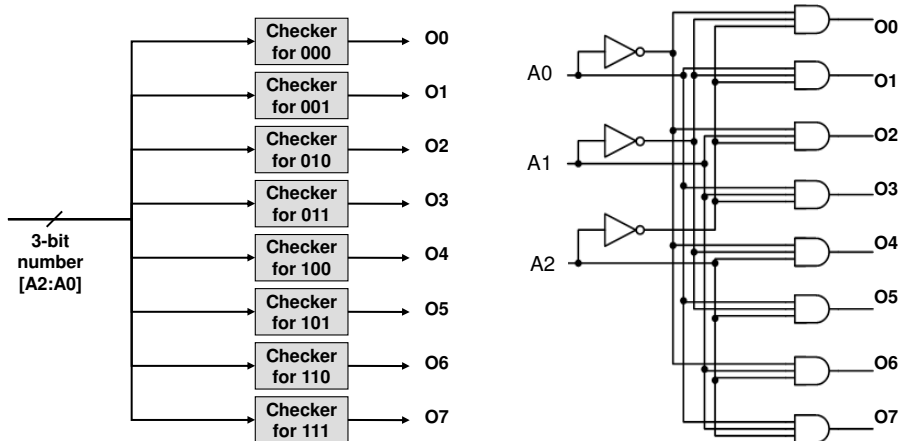
3-to-8 Decoder

Exercise

- Complete the design of a 2-to-4 decoder

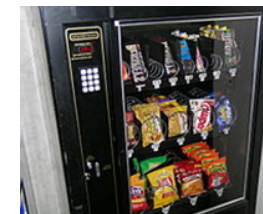
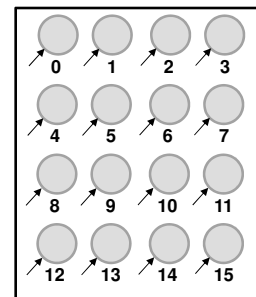


Building Decoders



Vending Machine Example

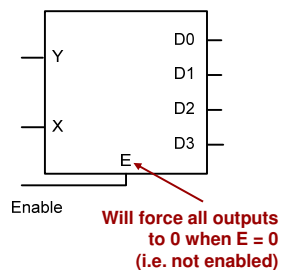
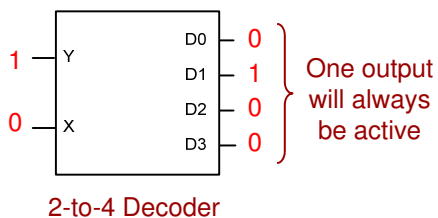
Assuming the keypad produces a 4-bit numeric output, add logic to produce the release signals for each of the 16 vending items.



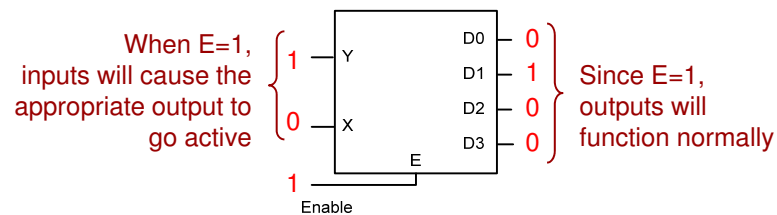
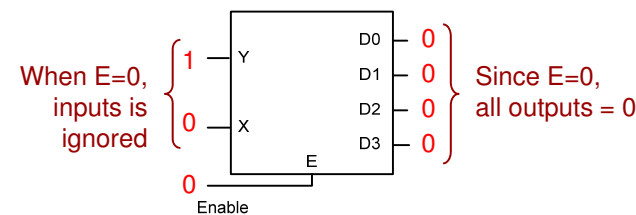
Consider any problems with this design.

Enables

- In a normal decoder exactly one output is active at all times
- It may be undesirable to always have an active output
- We can add an extra input (called an enable) that can independently force all the outputs to their inactive values

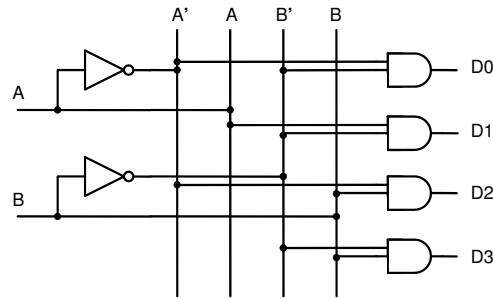


Enables



Implementing Enables

- Original 2-to-4 decoder



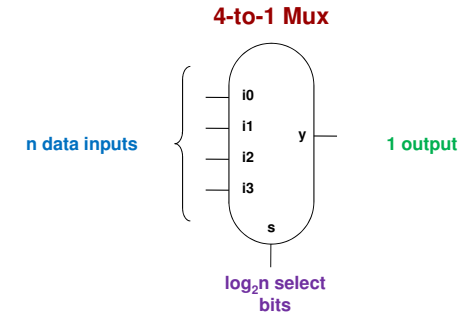
E —

When E=0, force all outputs = 0

When E=1, outputs operate as they did originally

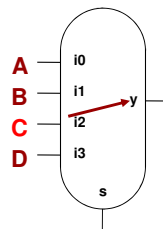
Multiplexers

- Multiplexers are one of the most common digital circuits
- Anatomy: n data inputs, $\log_2 n$ select bits, 1 output
- A multiplexer ("mux" for short) selects one data input and passes it to the output



Multiplexers

4-to-1 Mux

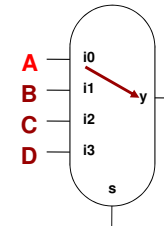


② Thus, input 2 = C is selected and passed to the output

① Select bits = $10_2 = 2_{10}$.

Multiplexers

4-to-1 Mux,
32-bit wide
mux

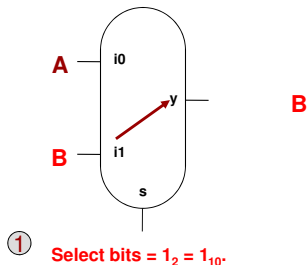


② Thus, input 0 = A is selected and passed to the output

① Select bits = $00_2 = 0_{10}$.

Multiplexers

2-to-1 Mux,
32-bit wide mux

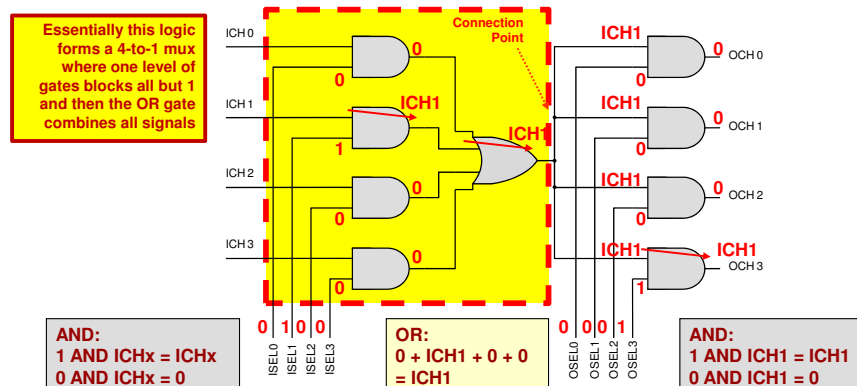


② Thus, input 1 = B is selected and passed to the output

① Select bits = $1_2 = 1_{10}$.

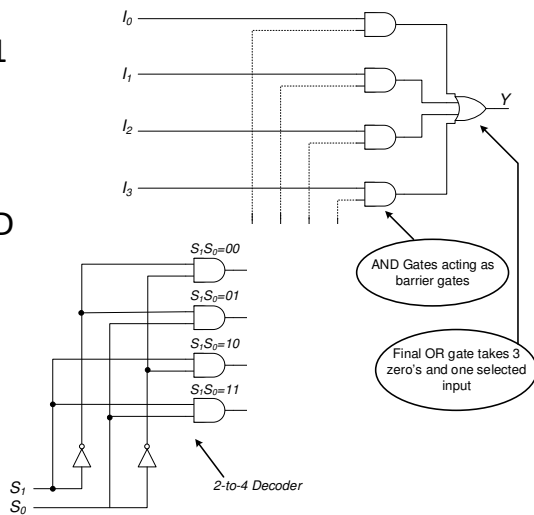
Recall Using T1/T2

- 1st Level of AND gates act as barriers only passing 1 channel
- OR gates combines 3 streams of 0's with the 1 channel that got passed (i.e. ICH1)
- 2nd Level of AND gates passes the channel to only the selected output



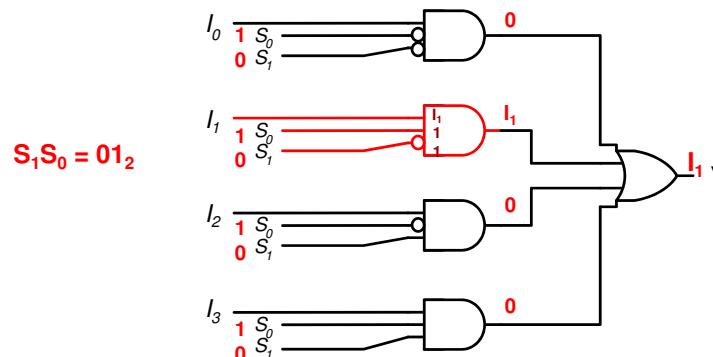
Exercise: Build a 4-to-1 mux

- Complete the 4-to-1 mux to the right by drawing wires between the 2-to-4 decode and the AND gates



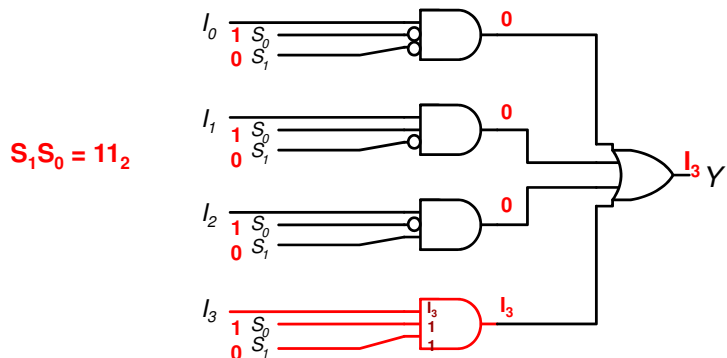
Building a Mux

- To build a mux
 - Decode the select bits and include the corresponding data input.
 - Finally OR all the first level outputs together.



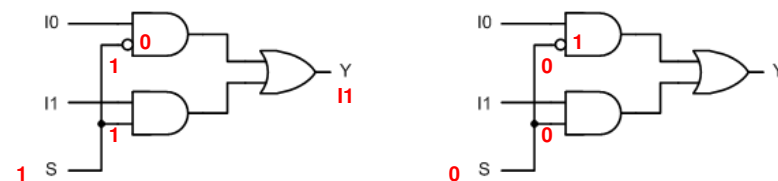
Building a Mux

- To build a mux
 - Decode the select bits and include the corresponding data input.
 - Finally OR all the first level outputs together.



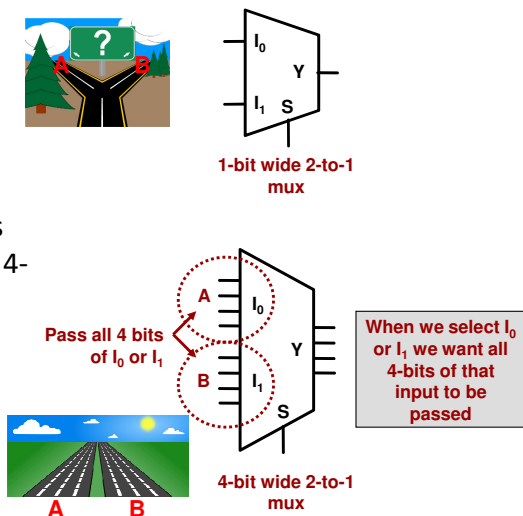
Building a Mux

- To build a mux
 - Decode the select bits and include the corresponding data input.
 - Finally OR all the first level outputs together.



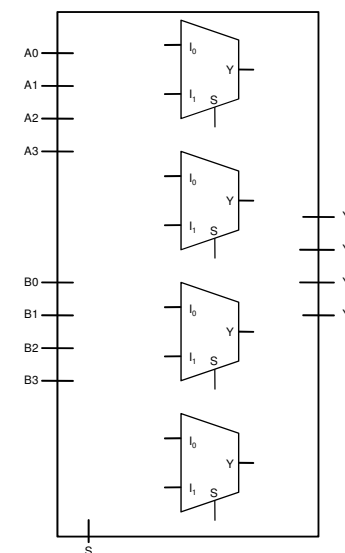
Building Wide Muxes

- So far muxes only have single bit inputs...
 - I_0 is only 1-bit
 - I_1 is only 1-bit
- What if we still want to select between 2 inputs but now each input is a 4-bit number
- Use a 4-bit wide 2-to-1 mux



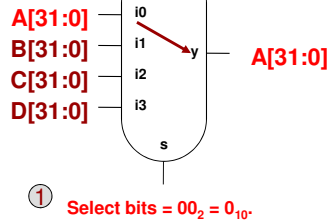
Building Wide Muxes

- Use one mux per _____
 - To build a 4-bit wide 2-to-1 mux, use ___ separate 2-to-1 muxes
- Operation:
 - When $S=0$, all muxes pass their I_0 inputs which means all the A bits get through
 - When $S=1$, all muxes pass their I_1 inputs which means all the B bits get through
- In general, to build an **m-bit wide (i.e. m-lane) n-to-1 mux**, use ___ individual _____ muxes



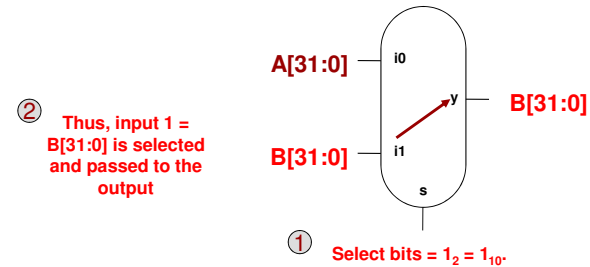
Multiplexers

4-to-1 Mux,
32-bit wide
mux



Multiplexers

2-to-1 Mux,
32-bit wide mux

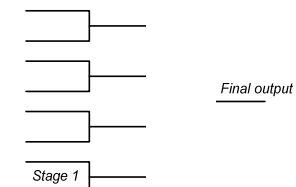


Exercise

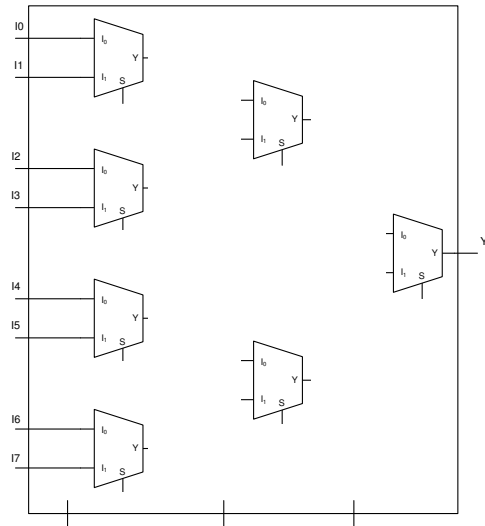
- How many 1-bit wide muxes and of what size would you need to build a **4-to-1, 8-bit** wide mux (i.e. there are 4 numbers: $W[7:0]$, $X[7:0]$, $Y[7:0]$ and $Z[7:0]$ and you must select one)
- How many 1-bit wide muxes and of what size would you need to build a **8-to-1, 2-bit** wide mux?

Building Large Muxes

- Similar to a tournament of sports teams
 - Many teams enter and then are narrowed down to 1 winner
 - In each round winners play _____



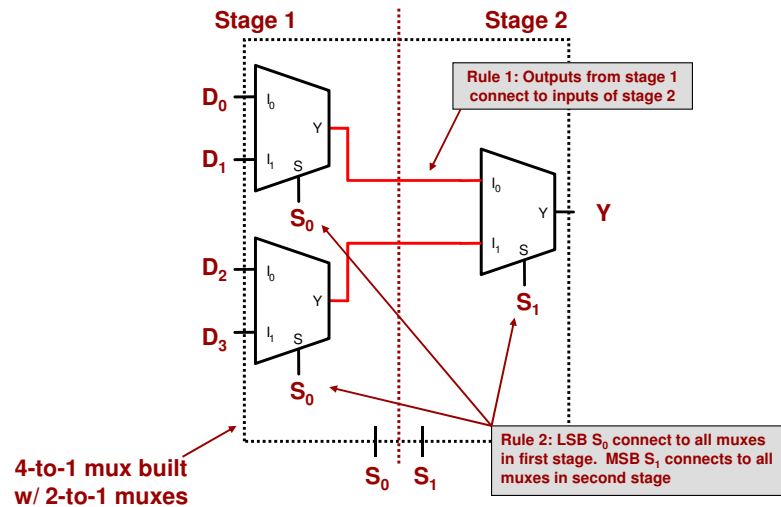
Design an 8-to-1 mux with 2-to-1 Muxes



Cascading Muxes

- Use several small muxes to build large ones
- Rules
 1. Arrange the muxes in stages (based on necessary number of inputs in 1st stage)
 2. Outputs of one stage feed to inputs of the next until only 1 final output
 3. All muxes in a stage connect to the same group of select bits
 - Usually, LSB connects to first stage
 - MSB connect to last stage

Building a 4-to-1 Mux

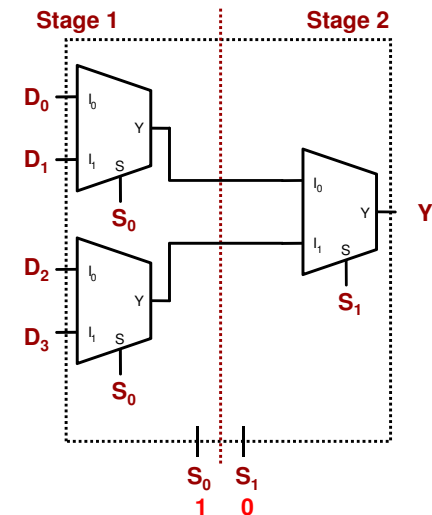


Building a 4-to-1 Mux

S ₁	S ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

Walk through an example:

$$S_1 S_0 = 01$$

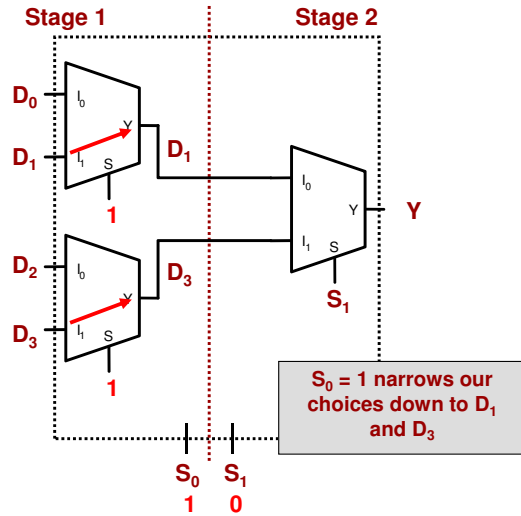


Building a 4-to-1 Mux

S ₁	S ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

Walk through an example:

$S_1S_0 = 01$

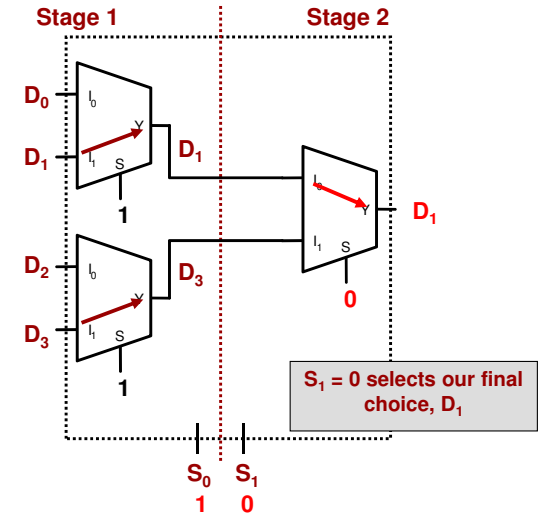


Building a 4-to-1 Mux

S ₁	S ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

Walk through an example:

$S_1S_0 = 01$



Exercise

- Create a 3-to-1 mux using 2-to-1 muxes
 - Inputs: I₀, I₁, I₂ and select bits S₁, S₀
 - Output: Y

S ₁	S ₀	Y
0	0	D ₀
0	1	D ₁
1	0	D ₂

I₀ _
I₁ _
I₂ _

_ Y