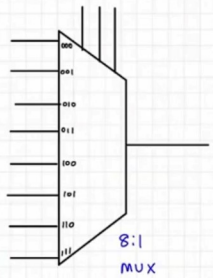
Lecture 7.2

Combinational Logic Design II

Synthesis of any logic function using Muxes

1. $F = \sum_{(A,B,C)}(0, 1, 3, 7)$ using 8:1 MUX

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



We have a Canonical SoP form we want to Synthesize using a MUX

$F = \sum_{(A,B,C)}(0, 1, 3, 7)$

minterms

MS input

LS input

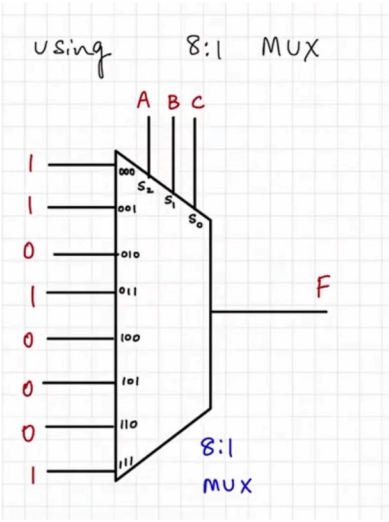
If ABC were switched around, we would have to adjust minterm order

1.) Fill out the Truth Table

So with the Truth Table and Canonical SoP, what are the Selects for the MUX and what are the inputs to the MUX?

2

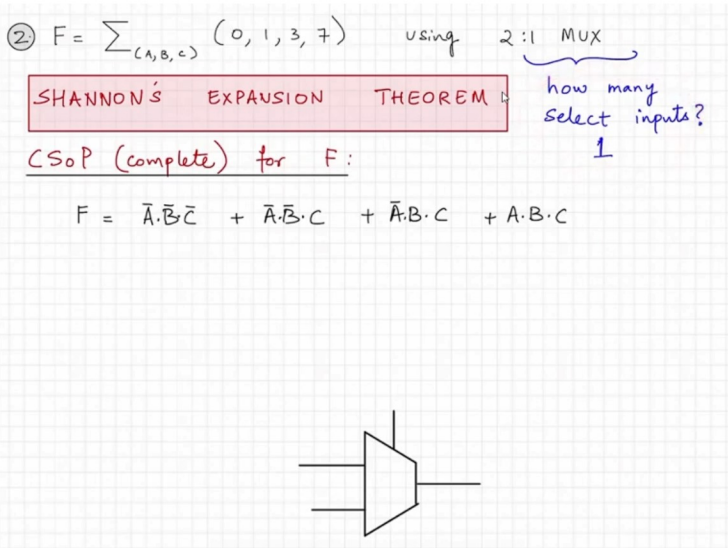
So we fill out the MUX as such



Synthesizing a 3 variable function with a 8:1 MUX is pretty straight-Forward.

To Synthesize using 2:1 MUX requires a different approach. USE Shannon's Expansion Theorem

1st) write out the Function in its Complete Canonical Sum of Products form.



After writing F out, how do we pick the select on our 2:1 MUX

Pick one (A, B, C) as select variable

Pick A

So now we want to rewrite the function while expanding it ~~to~~ with respect to select variable A.

That will Look Like

$$F = \bar{A}[\ ?\ ] + A[\ ?\ ]$$

So what goes in here to that we have something equivalent to $F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + ABC$

SHANNON'S EXPANSION THEOREM

how many select inputs? 1

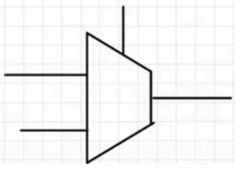
CSoP (complete) for F:

$F = \bar{A}\cdot\bar{B}\bar{C} + \bar{A}\cdot\bar{B}\cdot C + \bar{A}\cdot B\cdot C + A\cdot B\cdot C$

Pick one (A, B, C) as select variable

Andrew picked A

$F = \bar{A}\left[\bar{B}\cdot\bar{C} + \bar{B}\cdot C + B\cdot C\right] + A\left[\ B\cdot C\ \right]$



Now, we have This

can we simplify?

$$\bar{B}\bar{C} + \bar{B}C + BC = \bar{B}\cdot(\bar{C}+C) + B\cdot C$$

$$= \bar{B} + B\cdot C$$ — if $B=1$, depends on C; if $B=0$, expression is True

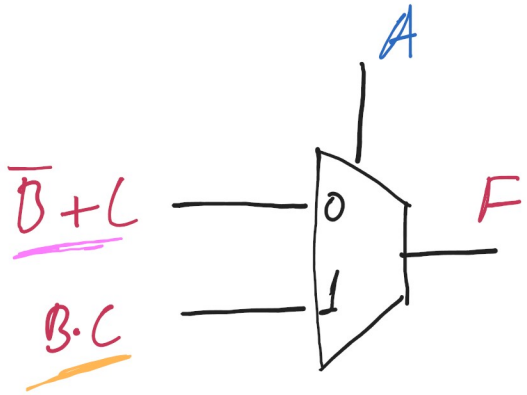
$$= \bar{B} + C$$

(use 11D from boolean Algebra properties)

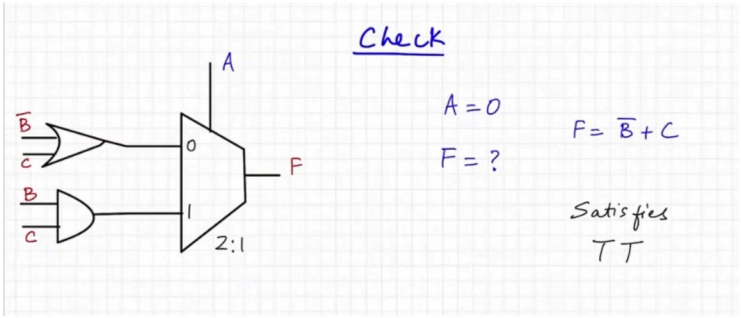
So now

$$F = \bar{A}(\bar{B}+C) + A(B \cdot C)$$

To Synthesize this as a MUX



We still need Logic gates To Synthesize this



← This satisfies The Truth Table

NOTE within a shannon's expansion Theorem, we can do another shannon's expansion Theorem.

If we did that to F, we would need another 2:1 MUX and thus another select

Another Example of MUX Synthesis

(3) $F = \sum_{(A,B,C)}(0,1,3,7)$ using a 4:1 MUX

So pick 2 select lines Pick B, C

So write $F = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + \bar{A}\cdot B\cdot C + A\cdot B\cdot C$

Do B first

Pick one (A,B,C) as Most significant select variable.

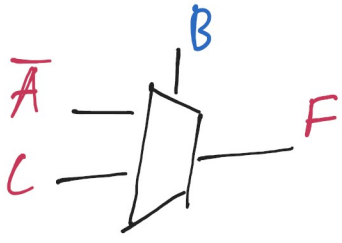
→ B

$$F = \bar{B}\left[\underbrace{\bar{A}\cdot\bar{C} + \bar{A}\cdot C}_{=\bar{A}}\right] + B\left[\underbrace{\bar{A}\cdot C + A\cdot C}_{=C}\right]$$

↖ This would be the Top variable in the MUX because of B

If we Look at the previous example, we can compare.

If we choose B, we don't need Logic gates.



B is MS input

6

Depending on which input you pick as the select variable, it will change the cost of your implementation

We can't stop here because this implementation requires a 2:1 MUX, we have a 4:1 MUX.

We need to do a shannon's expansion within these expressions

$$F = \bar{B}\left[\bar{A}\cdot\bar{C} + \bar{A}\cdot C\right] + B\left[\bar{A}\cdot C + A\cdot C\right]$$

$\bar{A}\cdot\bar{C} + \bar{A}\cdot C = \bar{A}$

$\bar{A}\cdot C + A\cdot C = C$

Pick Another one, Another select Variable (A, C)

Pick C

$$F = \bar{B}\left[\bar{C}(\quad) + C(\quad)\right] + B\left[\bar{C}(\quad) + C(\quad)\right]$$

These must match

These must match

The issue is that you can't break down $\bar{A}$ any further but we can still synthesize this. So where does $\bar{A}$ go?

$\bar{A}$ goes in both

$$F = \bar{B}\left[\bar{C}(\bar{A}) + C(\bar{A})\right] + B\left[\bar{C}(\quad) + C(\quad)\right]$$

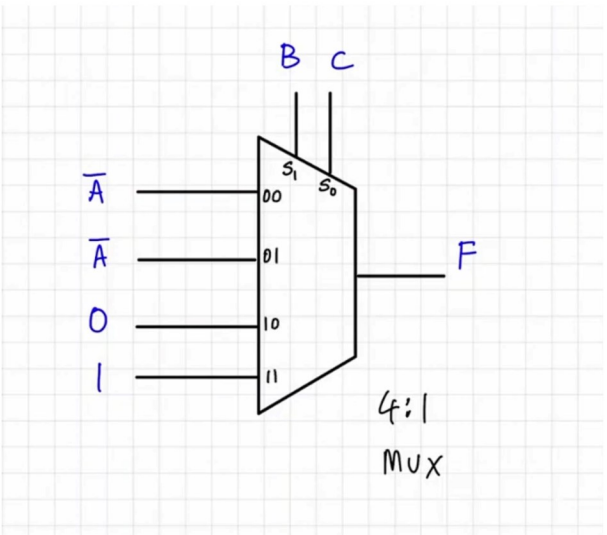
If we simplified this further, we'd get just $\bar{A}$.

(If it's not obvious where to put it, try putting it in both)

To solve this guy, it must match with the previous expression

$$F = \bar{B}\left[\bar{C}(\bar{A}) + C(\bar{A})\right] + B\left[\bar{C}(0) + C(1)\right]$$

In a 4:1 MUX, B is our MS select and C is our LS select.



And now we can check this

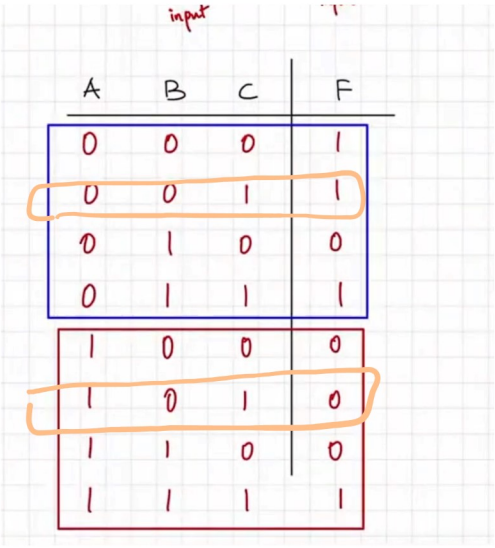
B = 0

C = 1

F = ?

$F = \bar{A}$

8 To check this, refer to the Truth Table



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

$B=0$
$C=1$

just look at A for $B=0, C=1$

if $A=0$, $F=1$ and when

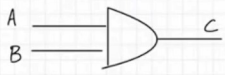
$A=1$, $F=0$

So $F=\bar{A}$

So if you had a Canonical PoS form, To implement that it would be a good idea to convert it to SoP from.

→ USE SoP forms to Synthesize via MUX.

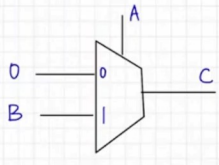
Synthesize a 2-input AND gate using a 2:1 MUX



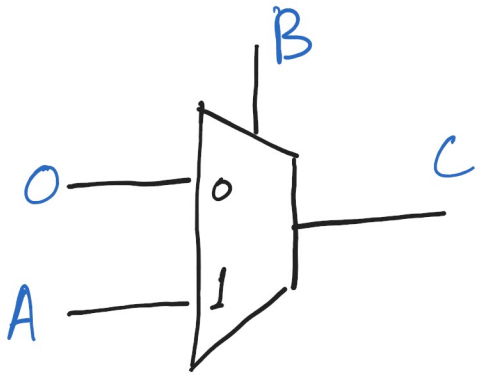
$C = A \cdot B$

Pick expansion variable (A, B). Drew picked A.

$C = \bar{A}[\ 0\ ] + A[\ B\ ]$

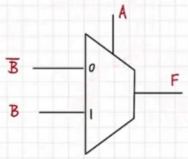


If you choose B as your Select variable



What if we already have a MUX and we want to find the Logic function?

A and B are related to F through which logic gate?



One way to do this is to write statements

When $A=0$, $F=?$ $F=\overline{B}$

When $A=1$, $F=?$ $F=B$

Or we can do shannon's expansion in Reverse.

$F=\bar{A}[\quad]+A[\quad]$

$F=\bar{A}[\ \overline{B}\ ]+A[\ B\ ]$

Top input

Bottom input

$=\bar{A}\cdot\bar{B}+A\cdot B=\overline{A\oplus B}$

exclusive NOR

XNOR

Mux summary

- Muxes are data switches
- They can switch one or more bits
- They route incoming data to a selected output
- They can realize arbitrary logic functions

- Demuxes distribute data from one source to n outputs
- They too can switch more than one bit

10 Demultiplexers (DEMUX)

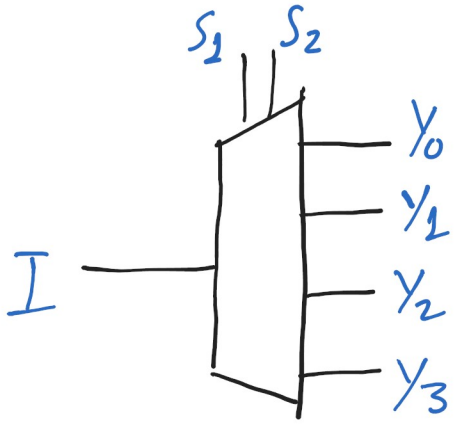
Earlier, we were selecting one from Many, Now we are routing 1 input to one of many outputs. (Depending on what you select)

Demultiplexers (DEMUX)

Opposite of MUX

- Demux: single data input, n control inputs, 2^n outputs
- control inputs (called selects S) represent binary index of output to which the input is connected

8



$S_1S_0 = 00 \Rightarrow Y_0 = I$

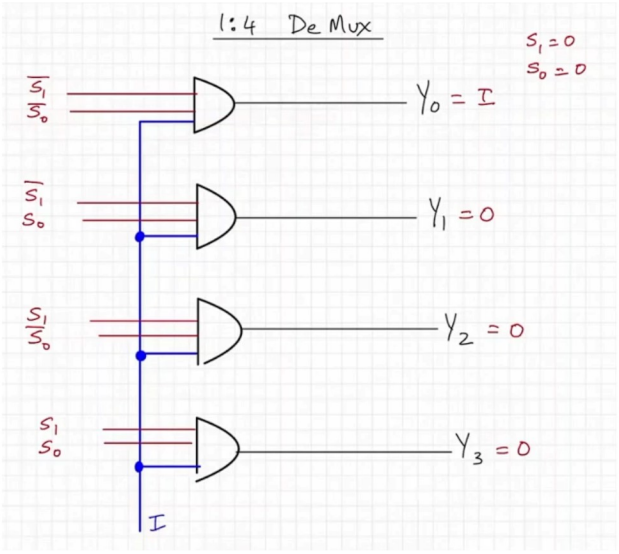
$S_1S_0 = 01 \Rightarrow Y_1 = I$

$S_1S_0 = 10 \Rightarrow Y_2 = I$

$S_1S_0 = 11 \Rightarrow Y_3 = I$

1:4 DeMUX

What's inside of a DeMUX? What's the Logic circuit?



This is the Logic Diagram

with $S_1 = 0$

$S_0 = 0$

DeMUX example

This is a block diagram that shows a MUX being used to select between many sources.

USE a bus to go to a destination.

Once at the destination, we have to choose one from many destination points.

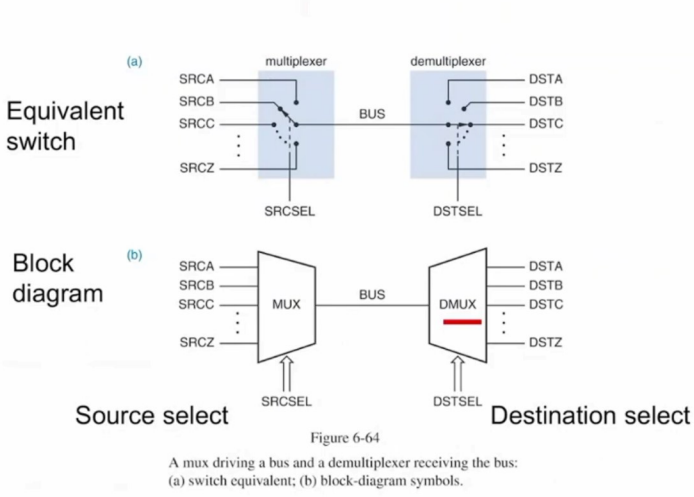
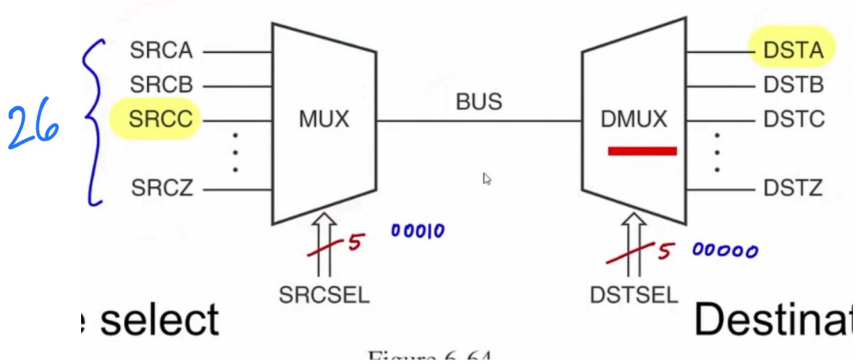


Figure 6-64

A mux driving a bus and a demultiplexer receiving the bus: (a) switch equivalent; (b) block-diagram symbols.

- we would use select lines to DeMUX what ever we got thru the Multiplexer

12



If we have 26 sources, How many selects do we need?

$\rightarrow 2^n \geq 26 \quad n = 5 \quad 2^5 = 32$

5 selects!

If we want to send data from source SRCC to DSTA, what will our selects be for the MUX and the DeMUX?

MUX Sel = 00010

00000 is the 1st source
00001 is the 2nd source

DeMUX Sel = 00000

Using decoders as a DeMUX

Using a 3-to-8 binary decoder as a 1-bit, 8-output demux
(the data is on the GO pin!)

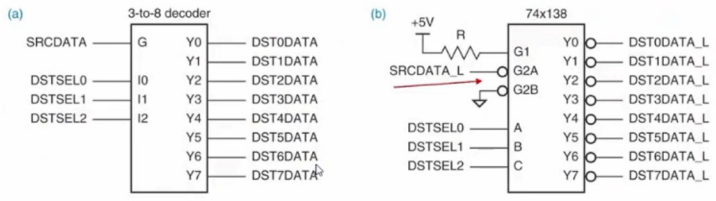


Figure 6-65
Using a 3-to-8 binary decoder as a 1-bit, 8-output demultiplexer: (a) generic decoder; (b) 74×138.

We are using a 74x138 Decoder chip and configuring it as a DeMultiplexer

G1 is Enabled

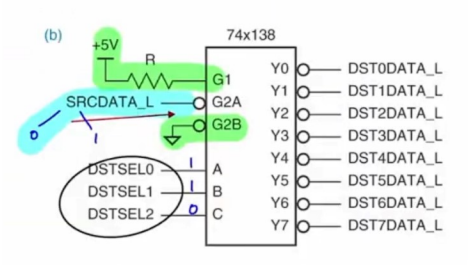
G2B_L is enabled

G2A (G2A_L) is designated as SRCDATA_L which is our Active Low source data,

We could have used G2B as well

We could have used G1 as an Active High source data as well.

Example



is The decoder enabled or disabled?

Enabled

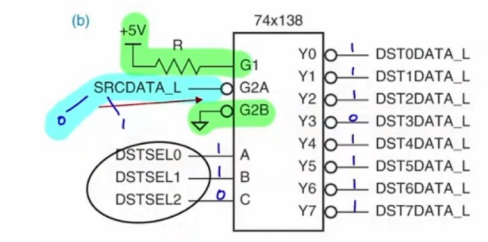
so it will do the job of decoding 1 1 0

what is 110? ABC?

A is LS input, C is MS input

We actually read this as 011 (decimal 3)

So Y3_L (DST3DATA_L) is Active



if SRCDATA_L = 1

All outputs are 1, so we are Good! source is still the same as the destination.