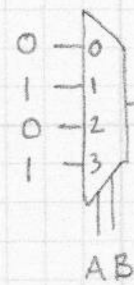
9.2a Using muxes to implement combinational logic

A) Number of mux inputs = $2^{\text{Number of input bits}}$

A	B	Z
0	0	0
0	1	1
1	0	0
1	1	1



B) Number of mux inputs = $\frac{1}{2} \cdot 2^{\text{Num. of input bits}}$

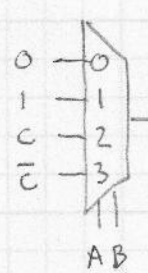
A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

AB = 00 $Z_{00} = 0$

AB = 01 $Z_{01} = 1$

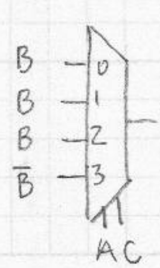
AB = 10 $Z_{10} = C$

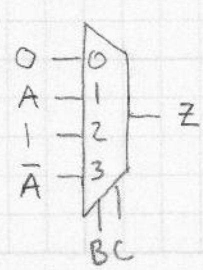
AB = 11 $Z_{11} = \overline{C}$



- Assign any input variables to mux control inputs
- For each mux input, there are only four possibilities:
  1) 0
  2) 1
  3) last input variable
  4) last input variable inverted
- ∴ Any boolean expression can be implemented

Other solutions:

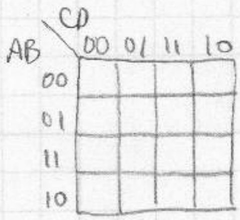




c) Number of mux inputs $= \frac{1}{2^2} \cdot 2^{\text{Number of input bits}}$

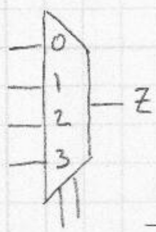
1) Approach #1

Look at K-map instead of truth table:

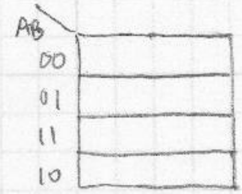


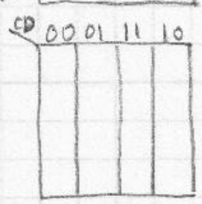
There are 6 ways to choose 2 of the 4 input variables:

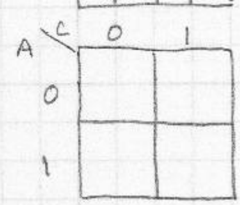
$$\text{"4 choose 2"} = \binom{4}{2} = \frac{4!}{2!(4-2)!} = \frac{24}{2 \cdot 2} = 6$$

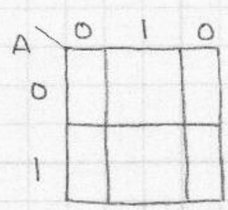


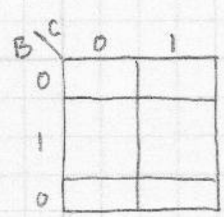
	Each mux input covers a:
AB	row in K-map
CD	column in K-map
AC	box
AD	split boxes
BC	split boxes
BD	split boxes

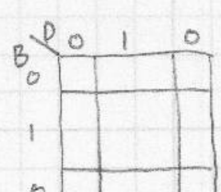












There are 16 possible combinations for each box of 4 outputs. 6 are trivial to implement, the other 10 require a gate.

- Designer judgment is required to find the best solution

2) Approach #2

a) Find minimum SOP or POS solution

b) Use mux to implement a sub-term

Ex: $Z = AC'D + BD$