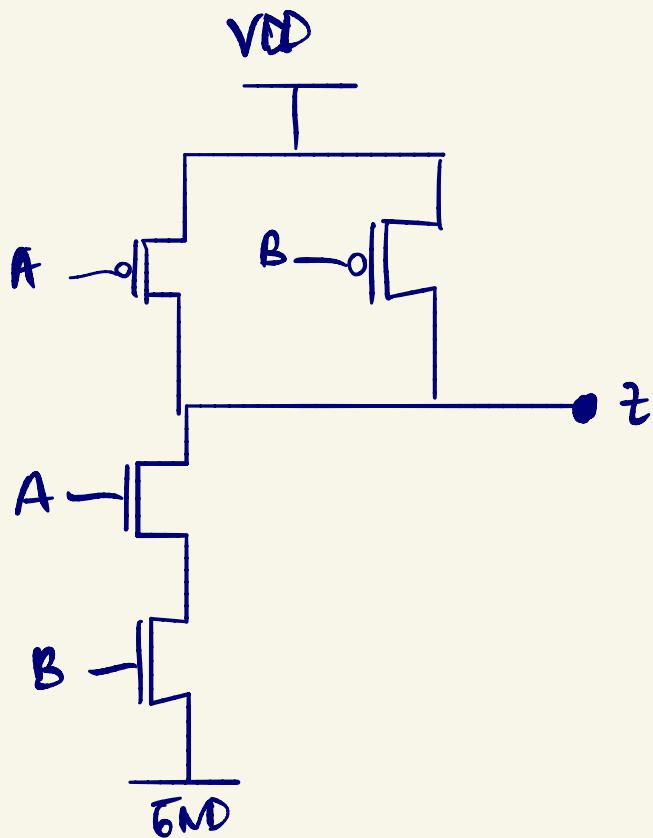


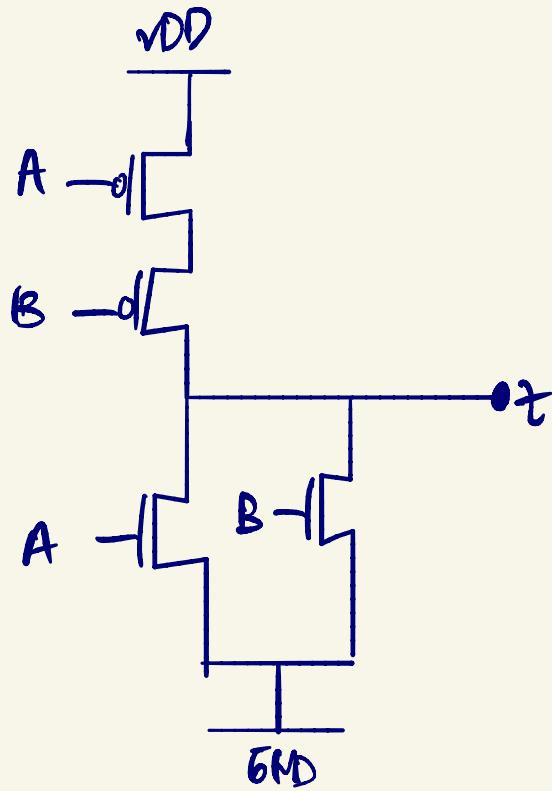
A	\bar{z}
0	1
1	0

Name: INVERTER



A	B	z
0	0	1
0	1	1
1	0	1
1	1	0

Name: **NAND**



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

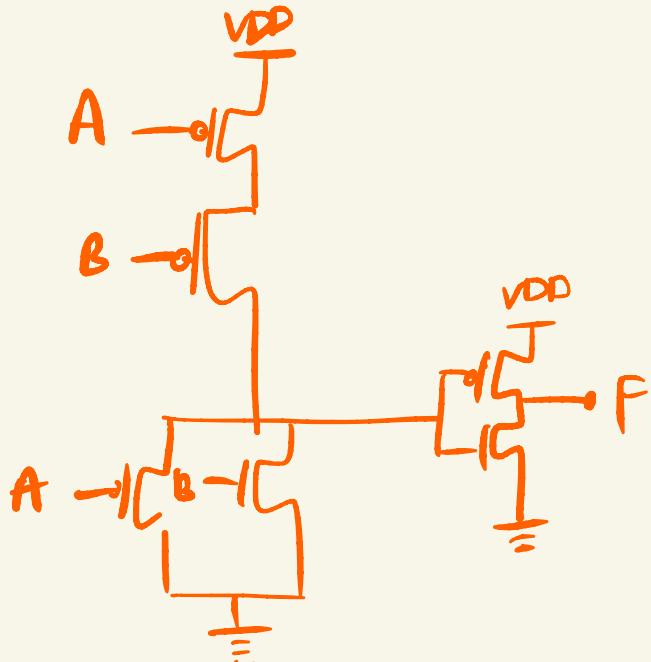
Name: NOR

$$F = A + B$$

① $\bar{F} = \overline{A+B}$

- ② Build pd
OR=parallel
AND=series

- ③ Build PU
④ add inverter

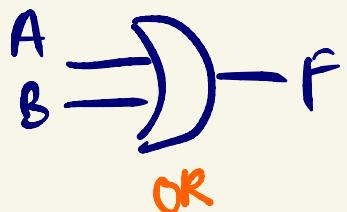


because we were
building pd for $A+B$,
now we have to invert the output

Sum of products

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

$$F = \bar{A}\bar{B} + A\bar{B} + AB$$



$$F = AB + A\bar{B} + \bar{A}B$$

\downarrow

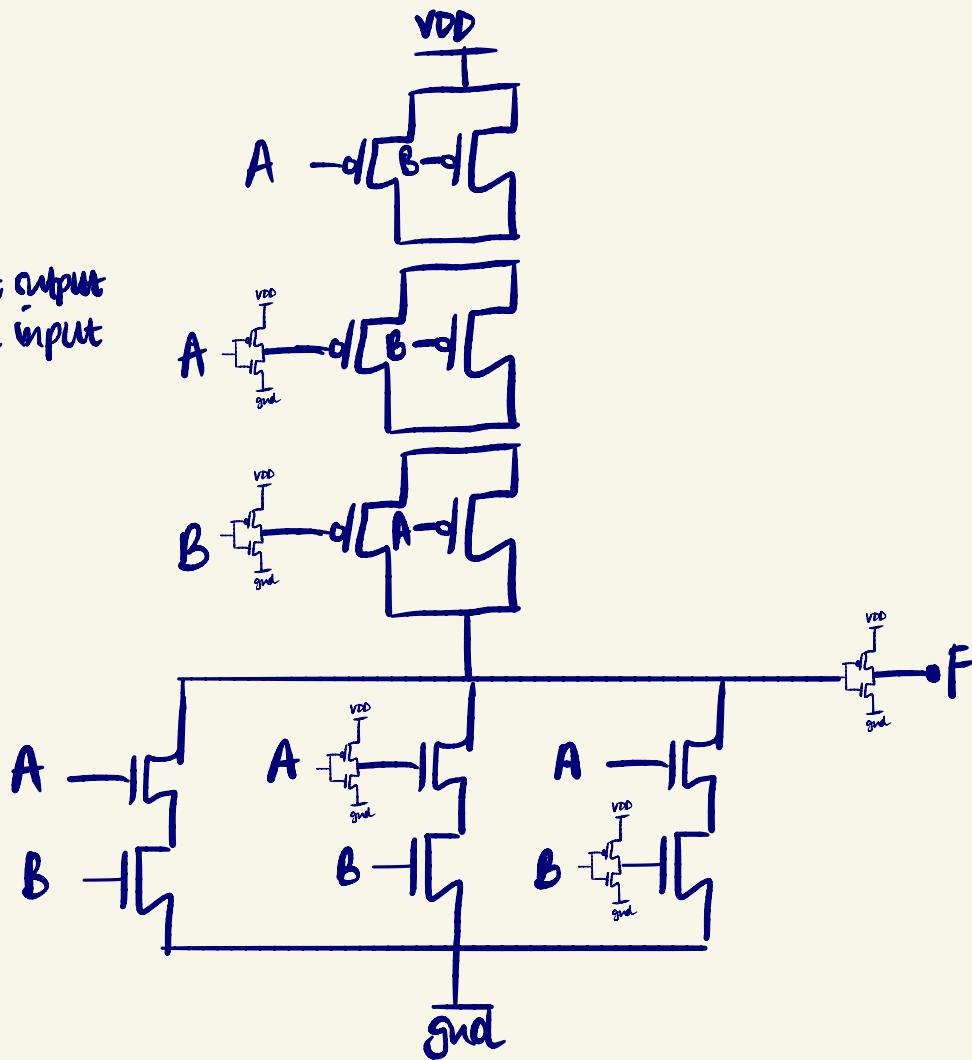
parallel NFET

$$\overline{F} = \overbrace{AB}^{\text{series NFET}} + \overbrace{A\bar{B}}^{\text{inv at output}} + \overbrace{\bar{A}B}^{\text{inv at input}}$$

series NFET

how many mosfets?

18



Understanding:

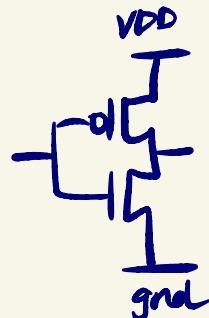
- ↳ figure out pull down circuit
- ↳ what combo results in 0 output?

↓

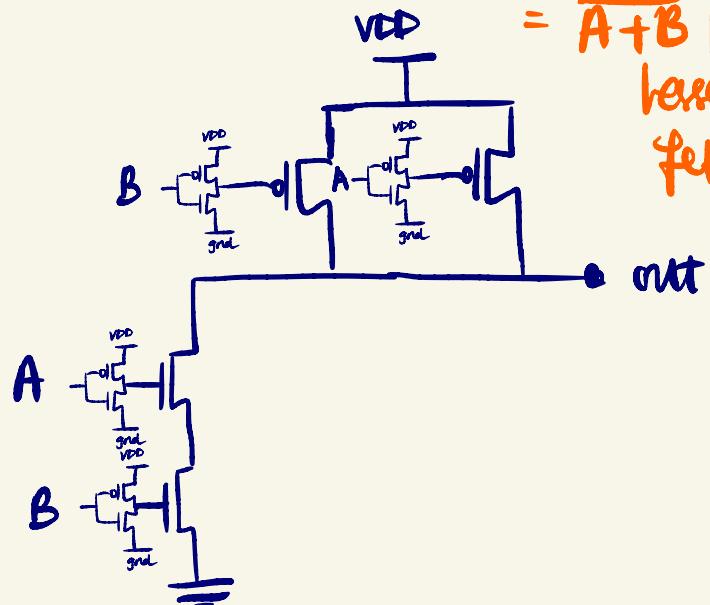
$\bar{A}\bar{B}$ gives 0

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

$\bar{A}\bar{B}$
series Nfet



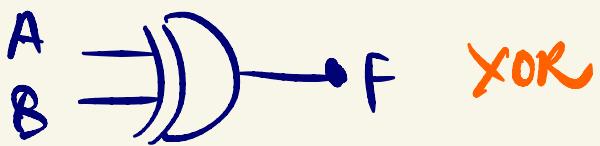
= $\bar{A} + \bar{B}$ has
less
fets · (6)



Cmos recipe

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

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



$$\overline{F} = \overline{\overline{A}\overline{B}} + \overline{A}\overline{\overline{B}}$$

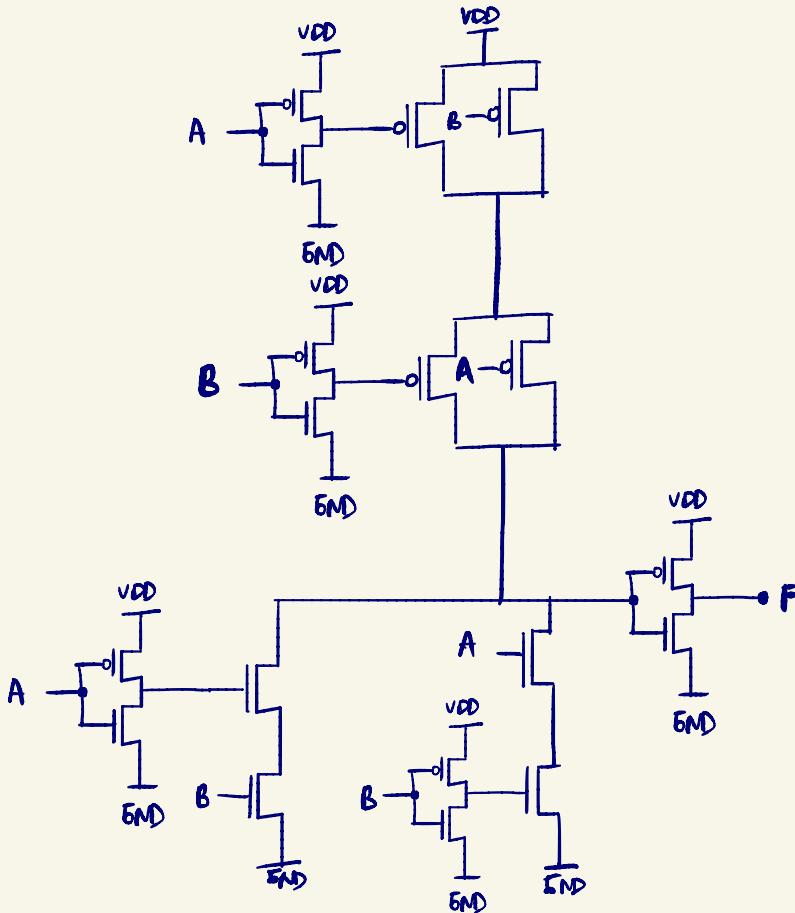
OR = parallel nFET

INV at output

AND = series Nfet

How many mosfets in total?

18



Understanding:

↳ figure out pull down circuit

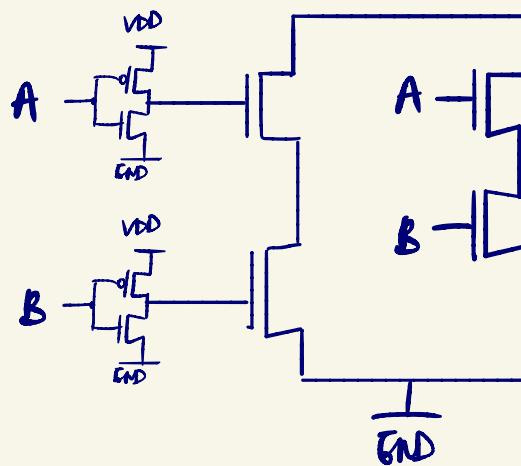
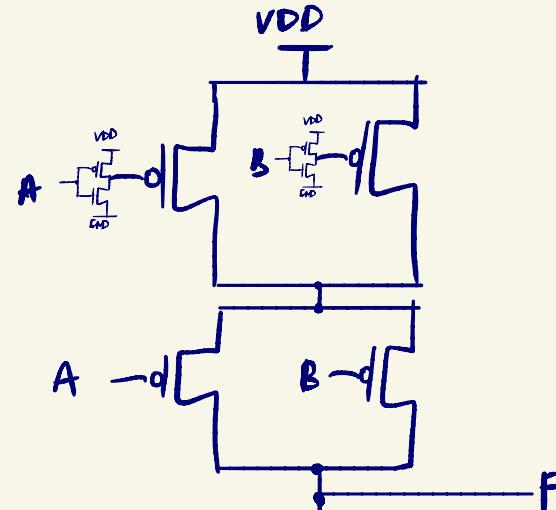
↳ what combo results in 0 output?



$$F = 0 \text{ when } \bar{A}\bar{B} + AB$$

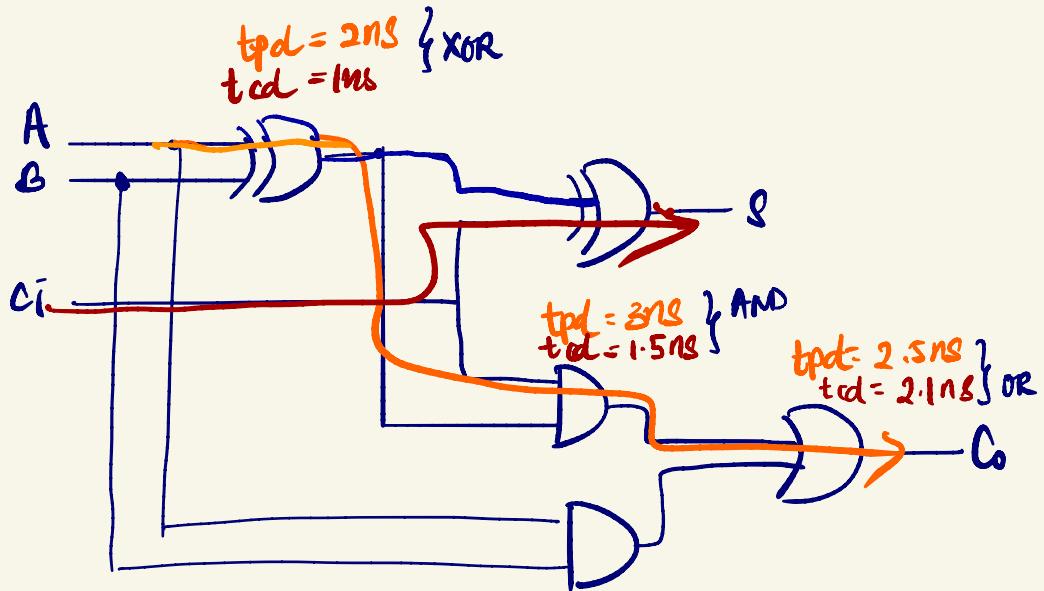
How many MOSFETs?

16



overall tpd?

7.5ns

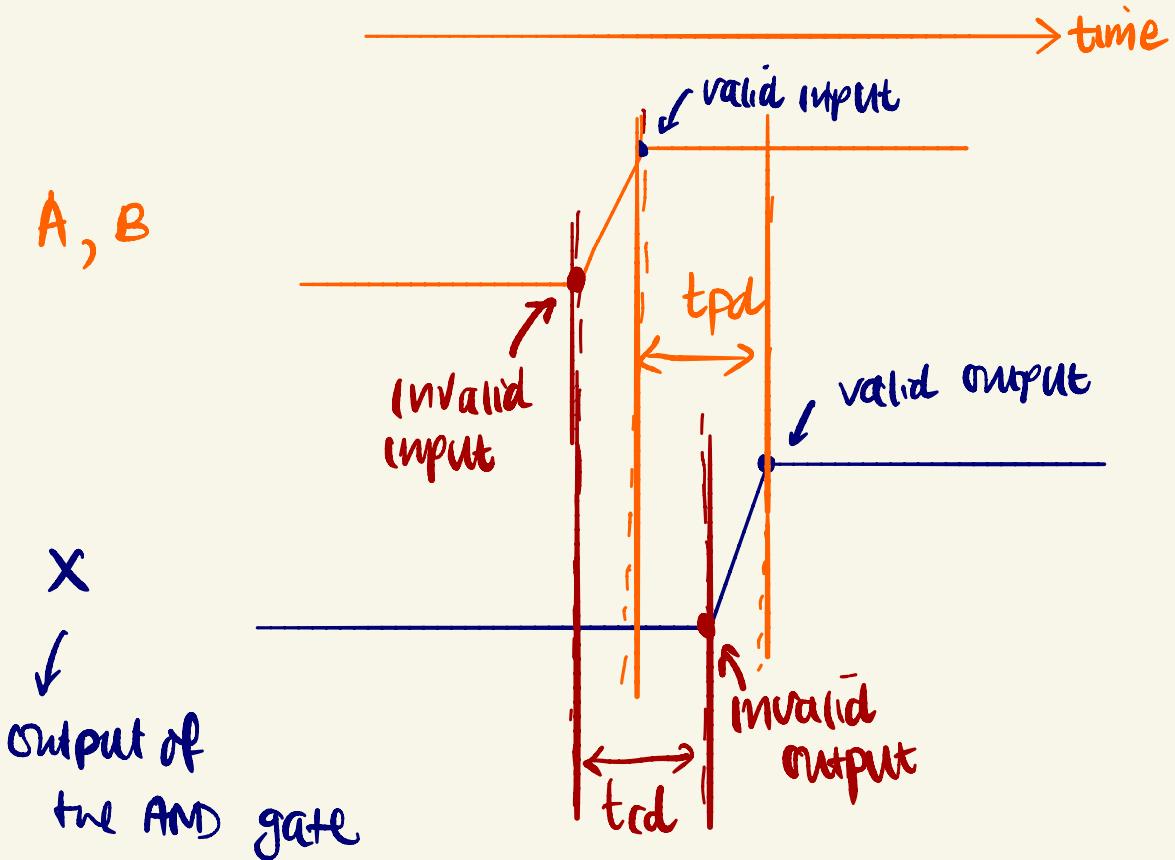


overall tcd? 1ns

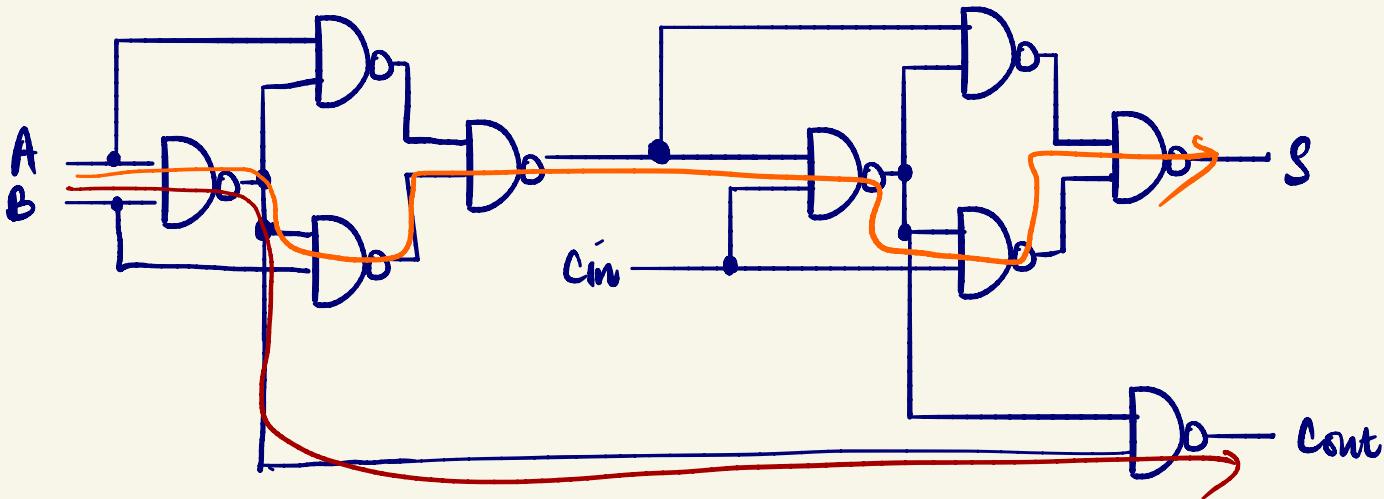
How many mosfets?

$$16 + 16 + 6 + 6 + 6 = 50$$

tpd & tcd
plots



$$tpd = 0.1\text{ns}$$
$$td = 0.2\text{ns}$$



Full adder
with NAND
universal

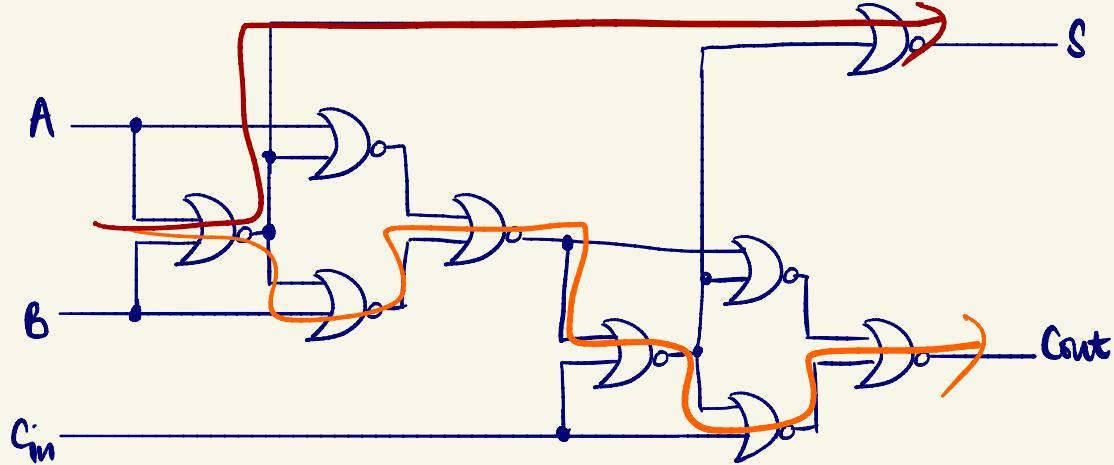
how many mosfets? 36

overall tpd? tdd?

$$0.6$$

$$0.4$$

Full adder
with
NOR
universal



How many mosfets? 36

overall tpd? tcd?

0.6

0.4

Sum of products (gates)

$$F = \overline{AB} + \overline{A}\overline{B}$$

AND

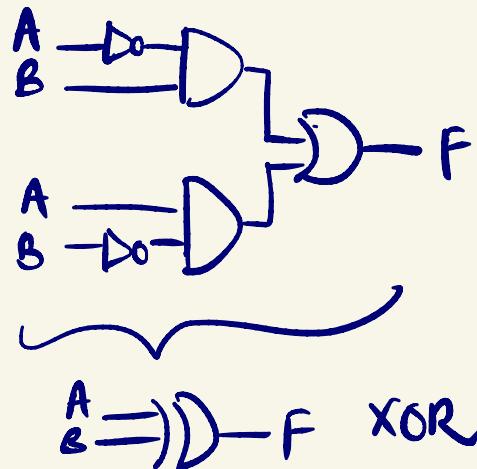
⊕ How many mosfets?

$$\text{INV} = 2 \times 2$$

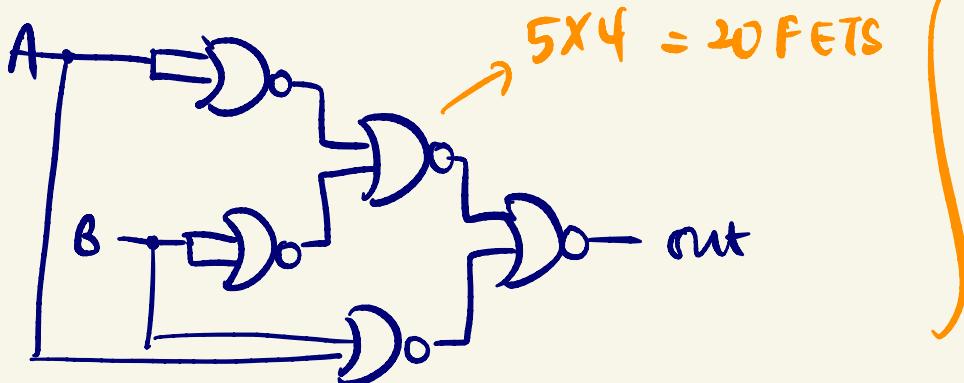
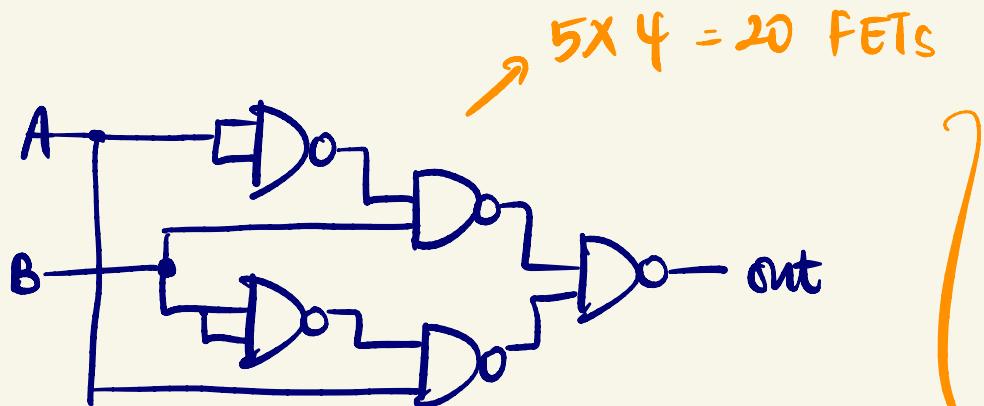
$$\text{AND} = 6 \times 2$$

$$\text{OR} = 6 \times 1$$

$$22$$



Universal gates : NAND & NOR



Both has XOR functionalities .

Boolean Simplification

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

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

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

$$= I$$

$$F = \overline{AB} (\bar{A} + B)(\bar{B} + B)$$

$$= \overline{AB} (\bar{A} + B) I$$

$$= \overline{AB} (\bar{A} + B)$$

$$= (\bar{A} + \bar{B})(\bar{A} + B)$$

$$= \bar{A}(\bar{B} + B)$$

$$= \bar{A}$$

how many gates initially?
how many gates after?

OR distributive law

$$A(B+C) = AB + AC$$

$$A + (BC) = (A+B)(A+C)$$

AND distributive law

$$F = (A+C)(AD + A\bar{D}) + AC + C$$

$$= (A+C)A + AC + C$$

$$= AA + AC + AC + C$$

$$= A + AC + C$$

$$= A + C$$

how many gates initially?
how many gates after?
↓
4 OR
3 AND
1 OR

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + B\bar{C}\bar{D} + BCD$$

$$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}D + A\bar{B}C\bar{D}$$

		CD	00	01	11	10
		AB	00	01	11	10
AB	CD	00	1	1	1	1
		01	1			1
AB	CD	11	1			1
		10	1	1	1	1

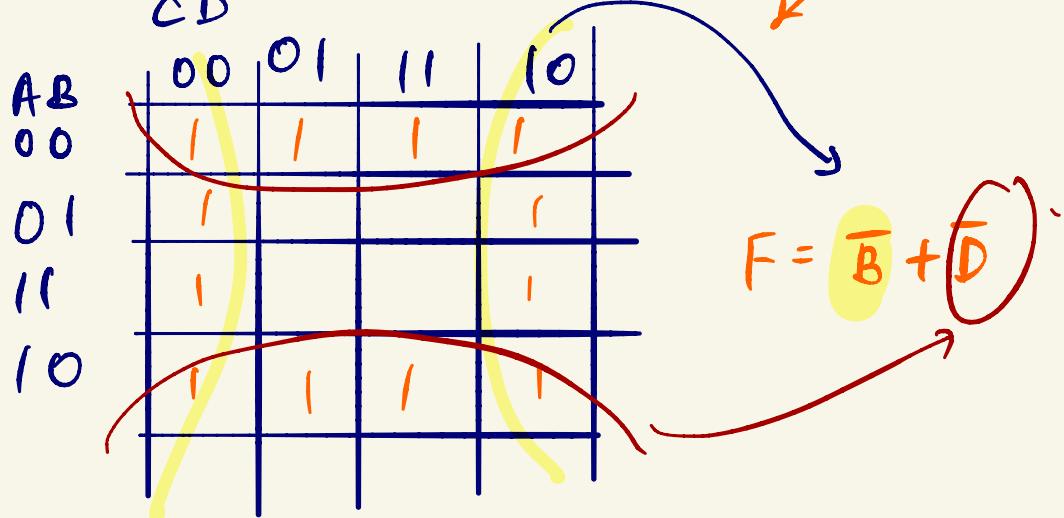
fill the map

- * form as large 2^x boxes
- * can "fold" and combine from the corners

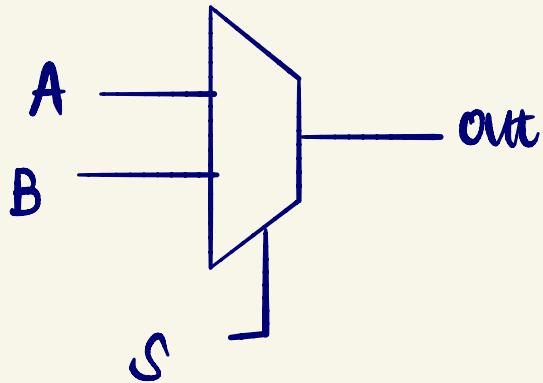
Eg: $\bar{A}\bar{B}\bar{C}\bar{D}$ results in 1 according to the sum of products, hence put 1 in cell AB=00, CD= 00

Karnaugh map

$$F = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + B\bar{C}\bar{D} + BCD \\ + A\bar{B}\bar{C}\bar{D} + A\bar{B}D + A\bar{B}C\bar{D}$$



Multiplexer



$$F = \bar{A}BS + A\bar{B}\bar{S} + \\ AB\bar{S} + ABS$$

A	B	S	out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

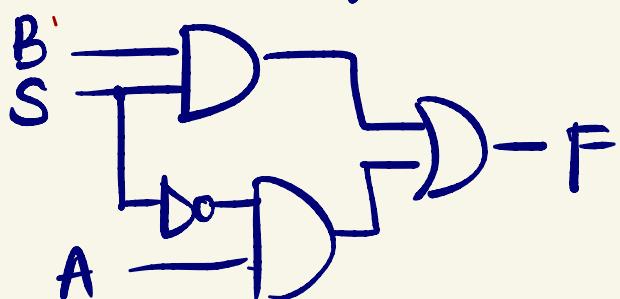
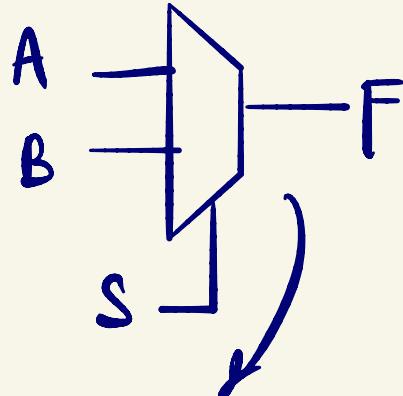
$$F = \bar{A}BS + A\bar{B}\bar{S} + AB\bar{S} + ABS$$

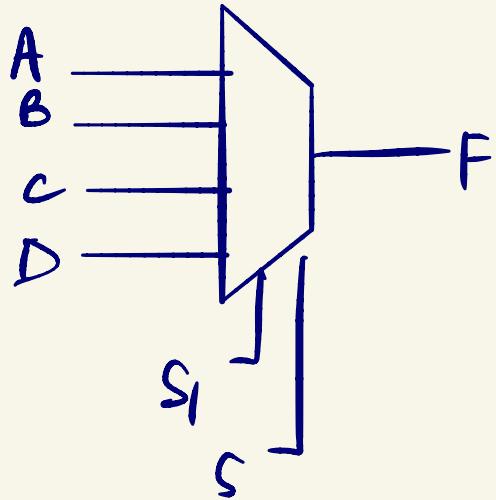
$$= \underbrace{\bar{A}BS + ABS}_{B(\bar{A}S + AS)} + \underbrace{A\bar{B}\bar{S} + AB\bar{S}}_{A(\bar{B}\bar{S} + B\bar{S})}$$

$$B(\bar{A}S + AS)$$

$$= B(S)$$

$$= BS + A\bar{S}$$



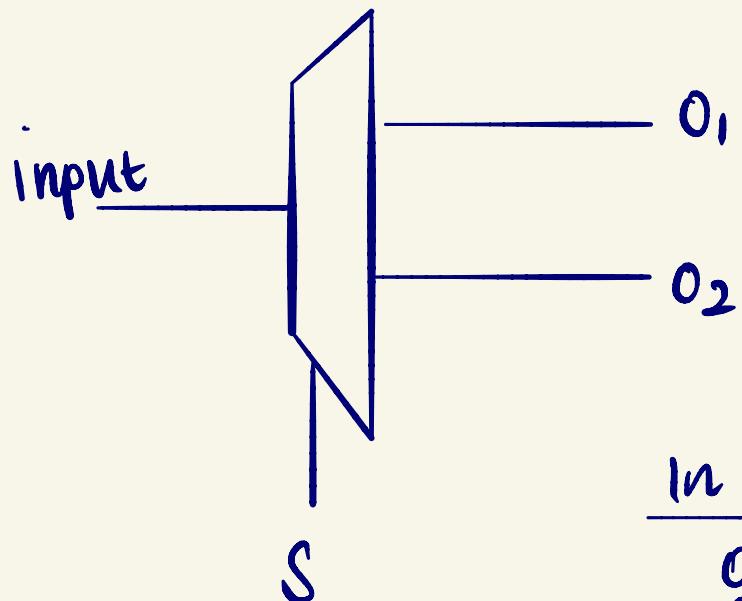


for n input bits,
 $\log_2(n)$ selector signals

S_1, S_2	A	B	C	D	out
0 0					A
0 1					B
1 0					C
1 1					D

An orange diagonal line with the word "any" written along it connects the selector inputs S_1, S_2 to the output "out".

Demultiplexer



S input	O_1	O_2
0	any input	0
1	any input	0

Truth tables for the Demultiplexer:

Top Truth Table:

In	S	O_1	O_2
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Middle Truth Table:

In	S	O_1	O_2
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

In	S		O ₁
0	0		0
0	1		0
1	0		0

In	S		O ₂
0	0		0
0	1		0
1	0		0

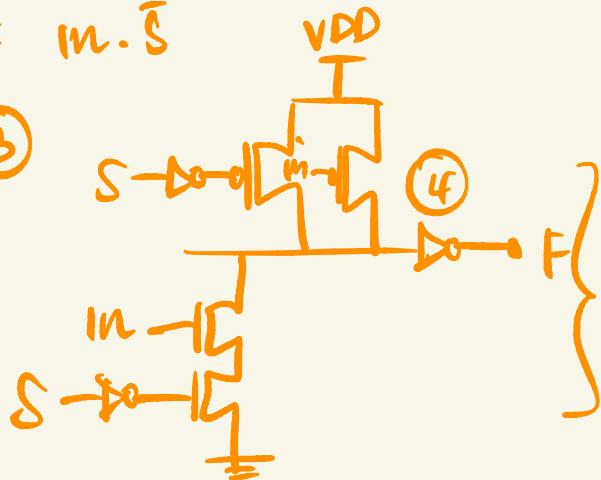
① AND . gate
6 fetes



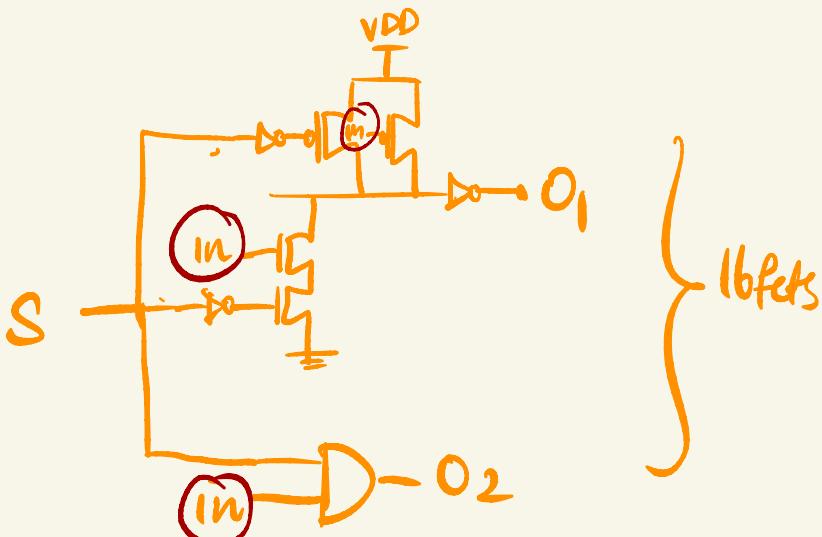
$$F = \overline{In \cdot S}$$

$$\textcircled{1} \bar{F} = \overline{\overline{In} \cdot \overline{S}}$$

② & ③

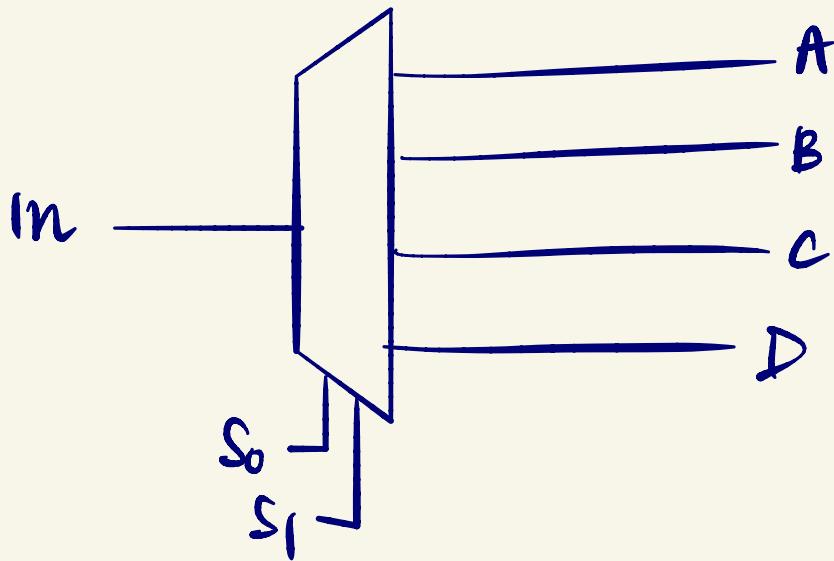


10 fetes



16 fetes

"Demux"



N output signals (bits)



$\log_2(N)$ selectors



You can make
this a ROM!

