Geometric design rules
Postlab lab 2
Prelab lab 3

Week 4

- Monday lab2
  - Carry gate schematic
- Tuesday
  - Lecture Layout of CMOS gates
- Thursday
  - Postlab review lab 2
  - Geometrical design rules incl. prelab 3
  - Tutorial POTW Layout (Victor)
- Friday Deadline prelab 3
  - Layout of carry ckt of full adder

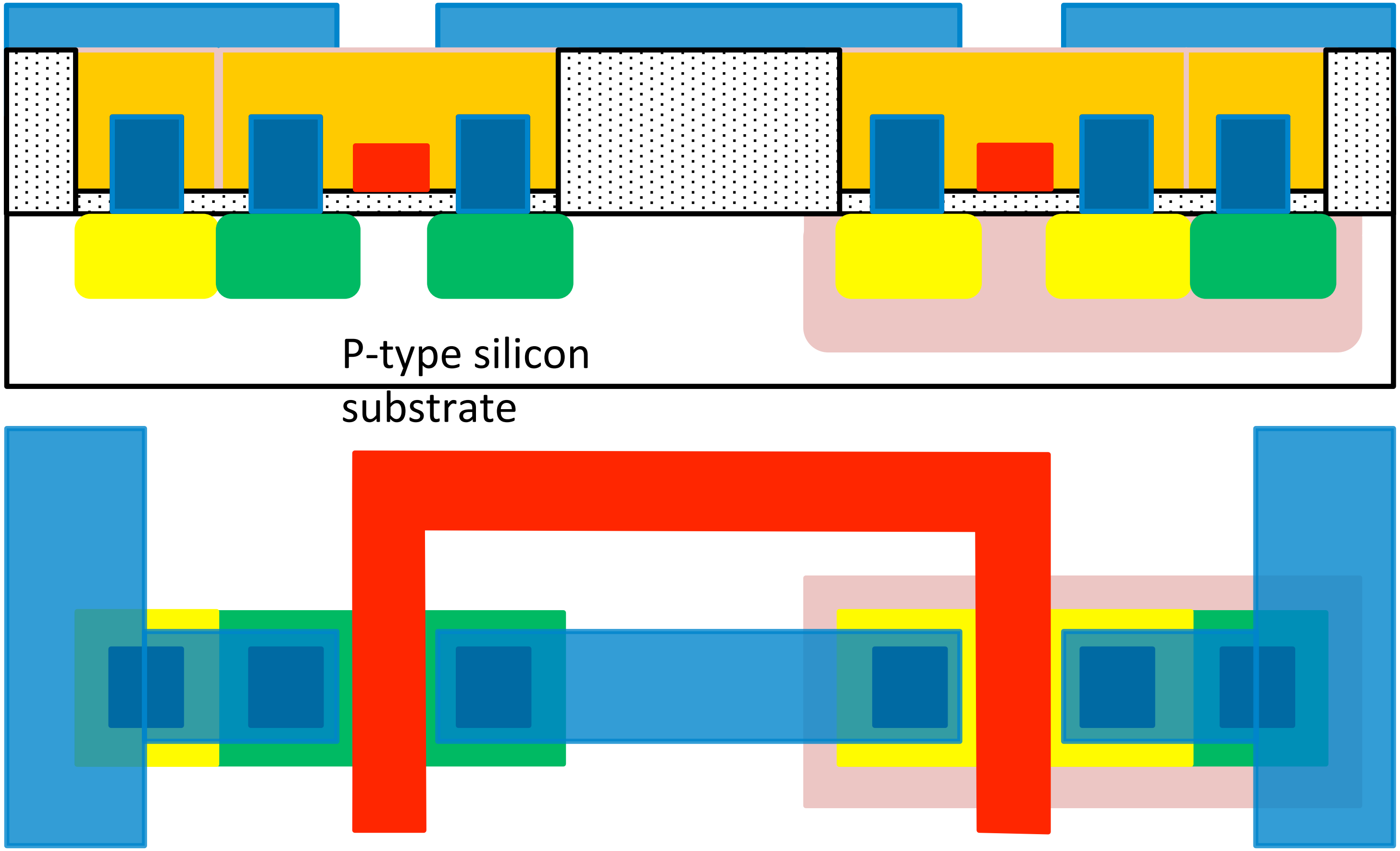
Practical things

- Thursday morning consultation time moved to 9.15-10 starting next Thursday
- Lars Svensson will help out with lab 3
- Mid-course meeting with student representatives held today.
  - Cadence remotely with VPN?
  - Chapter 2 problems?

From MUD cards

- How should metal line overlap contacts?
- Why do we connect input to V_{DD} on the second connection point on P-substrate?
- Euler path
  - How to draw it through odd and even nodes.
  - How to convert path to layout.

Inverter mask set and fabrication

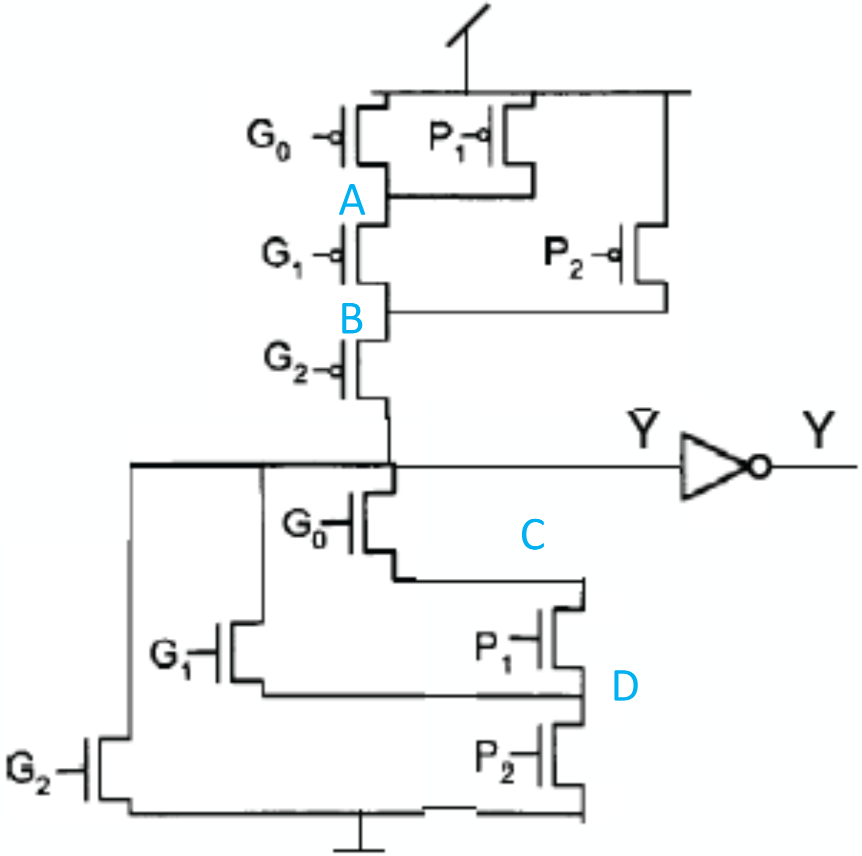


- 🟨 P+ select
- ⬛ Contact cuts
- 🟦 Metal wires
- 🟥 Poly gate
- 🟩 Active areas
- ▨ N-well

Example Euler paths

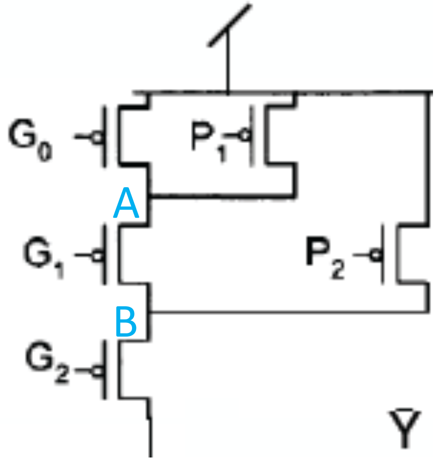
It is a good idea to label the circuit nodes that are neither the output, nor VDD or VSS.

I have done so here.
There are four such nodes:
A, B, C, D.



Example Euler paths

How many transistors are connected to each of the circuit nodes in the **p-net**?
VDD: 3
A: 3
B: 3
Y': 1



For an Euler path to exist there can only be 0 or 2 vertices (that is circuit nodes) with odd number of edges (that is transistors).

So with this schematic for the p-net there is no Euler path.

Example Euler paths

How many transistors are connected to each of the circuit nodes in the **n-net**?

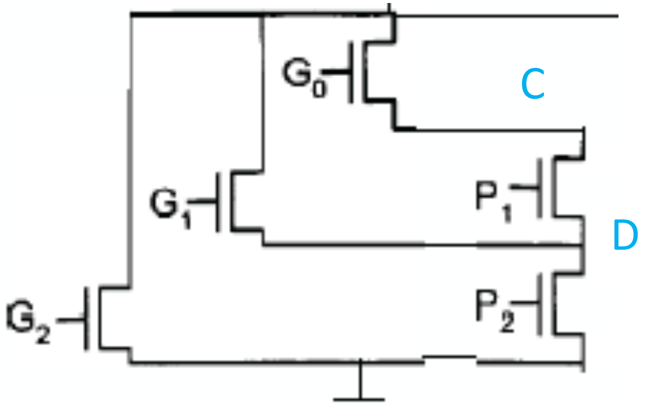
VSS/gnd: 2

D: 3

C: 2

Y’: 3

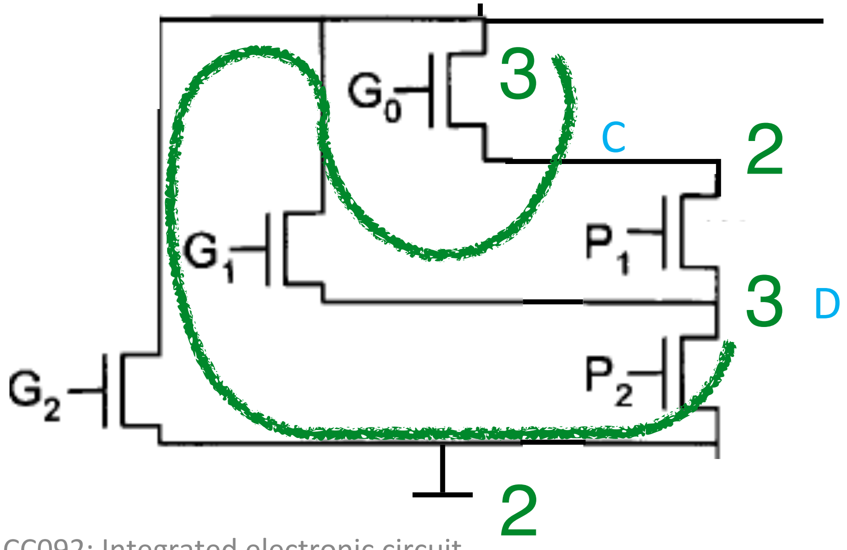
There are two vertices (circuit nodes) with an odd number of edges (transistors). Thus, an Euler path can be formed. It has to start and end in the circuit nodes with odd number of transistors: The output, Y, and internal node D.



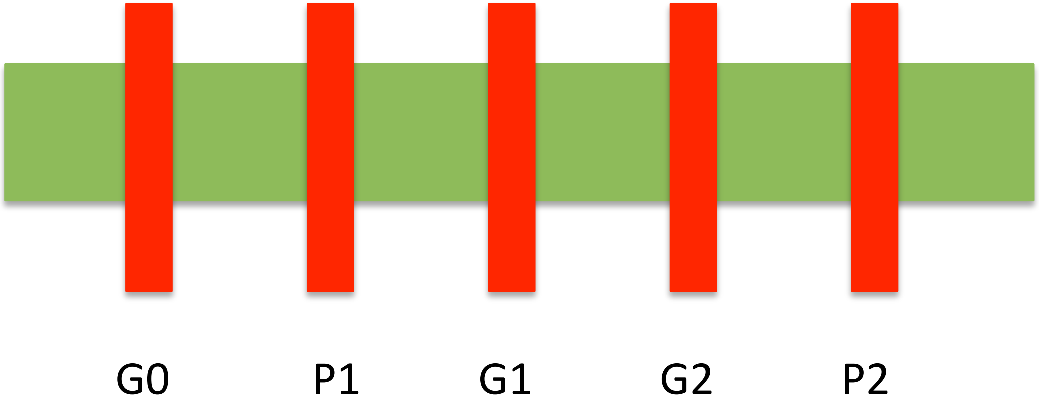
Example Euler paths

We have to start in a odd node and end in the other odd node.

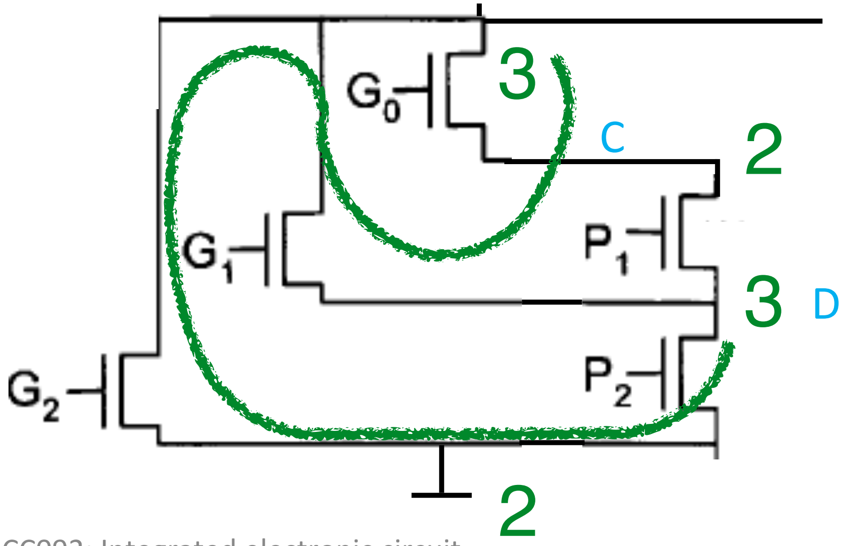
Here is one solution. There are several other possibilities.



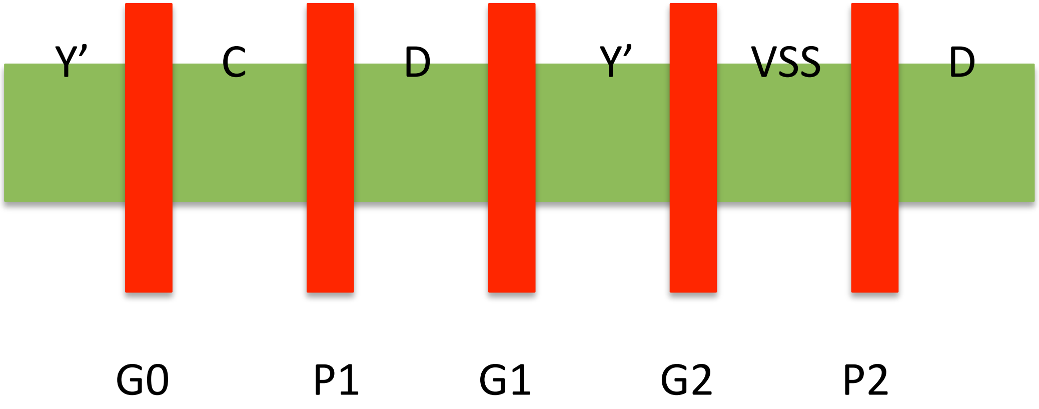
Example Euler paths



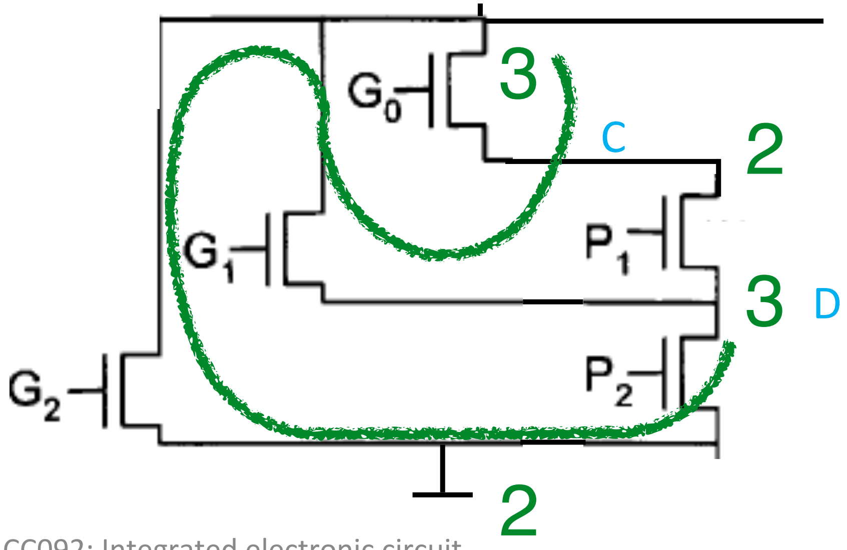
To map Euler path to a continuous line of diffusion start by labeling the transistors with their signal name starting at one end of Euler path.
(If some signals appear more than once give unique names)



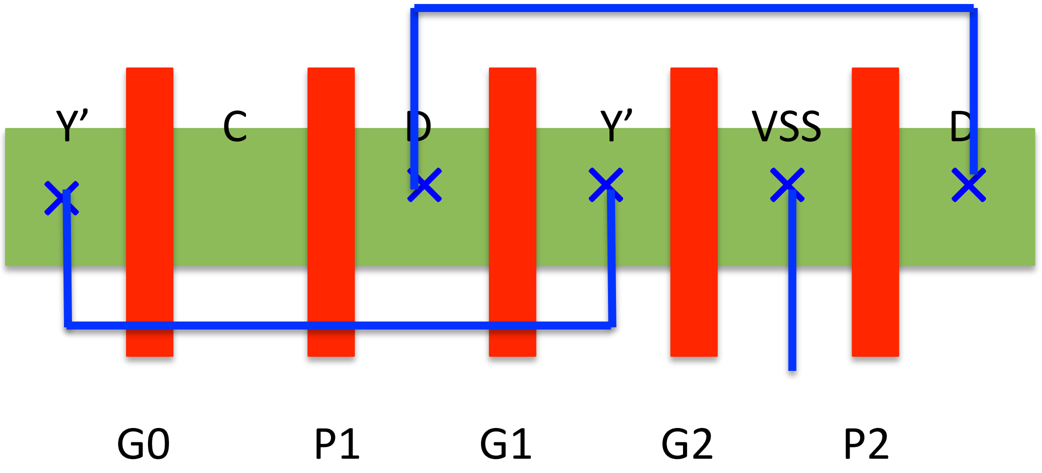
Example Euler paths

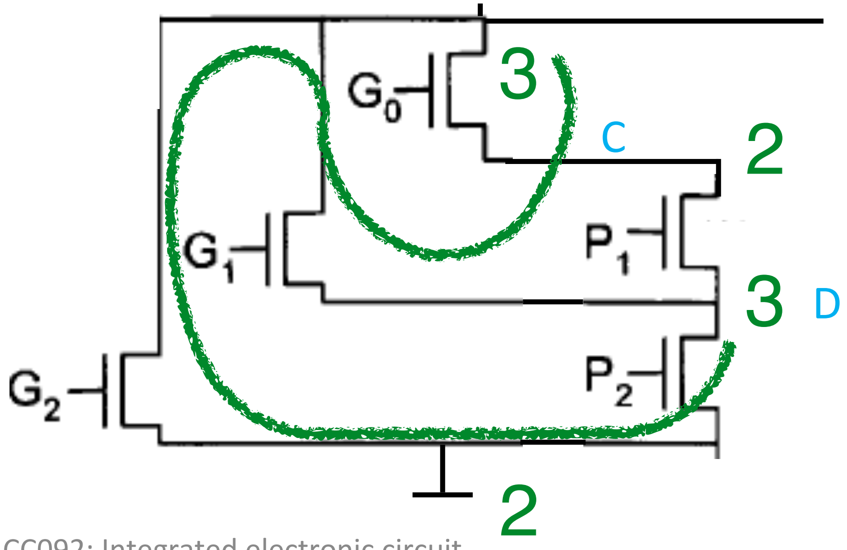


Then label also the circuit nodes along the path.



Example Euler paths





The make the connections for the source-drain areas that have the same circuit-node label.
I just draw it as a stick diagram here.

Example with quiz

Draw the schematic for a static CMOS gate with the logical function:

$f=\overline{(A\cdot B+C\cdot D)\cdot(E+F+G)}$

1 The n-net can be laid out with single line of diffusion. T/F

2 The p-net can be laid out with single line of diffusion. T/F

3 Can you make the order ABCDEFG work for the n-net?

4 Can you make the order ABCDEFG work for the p-net?

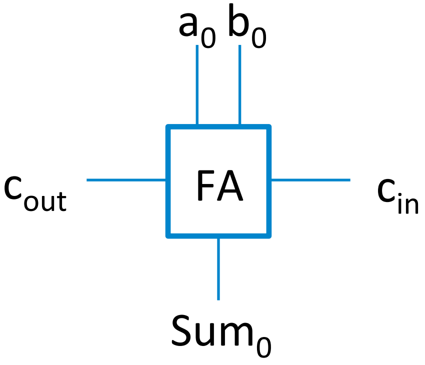
Postlab 2

Prelab 2

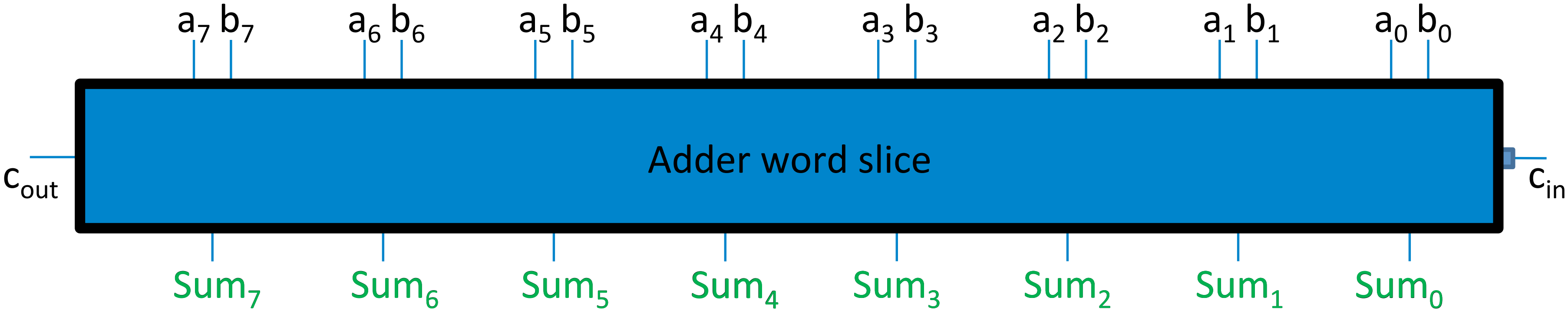
- Design task: design with MOSFETs the carry bit-cell for an 8-bit ripple-carry adder!
- The 8-bit carry logic is to be implemented by an iterative logic array consisting of eight instances of the bit-cell that you have designed.
- The carry bit-cell has three inputs (two bits a, b and a carry-in memory bit), and one output, the carry-out memory bit to the next more significant bit.

Iterative logic arrays: FULL ADDER

Find Boolean equations from truth table!

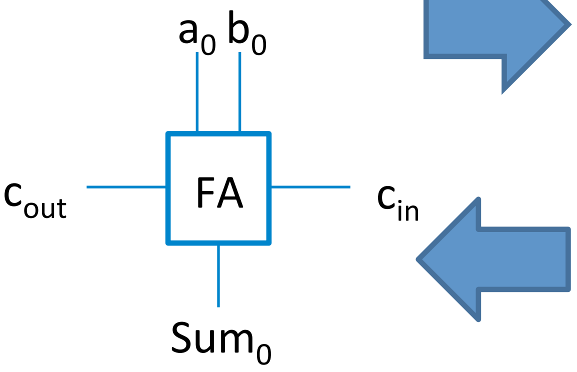


BOOLEAN TRUTHTABLE				
A	B	CIN	COUT	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



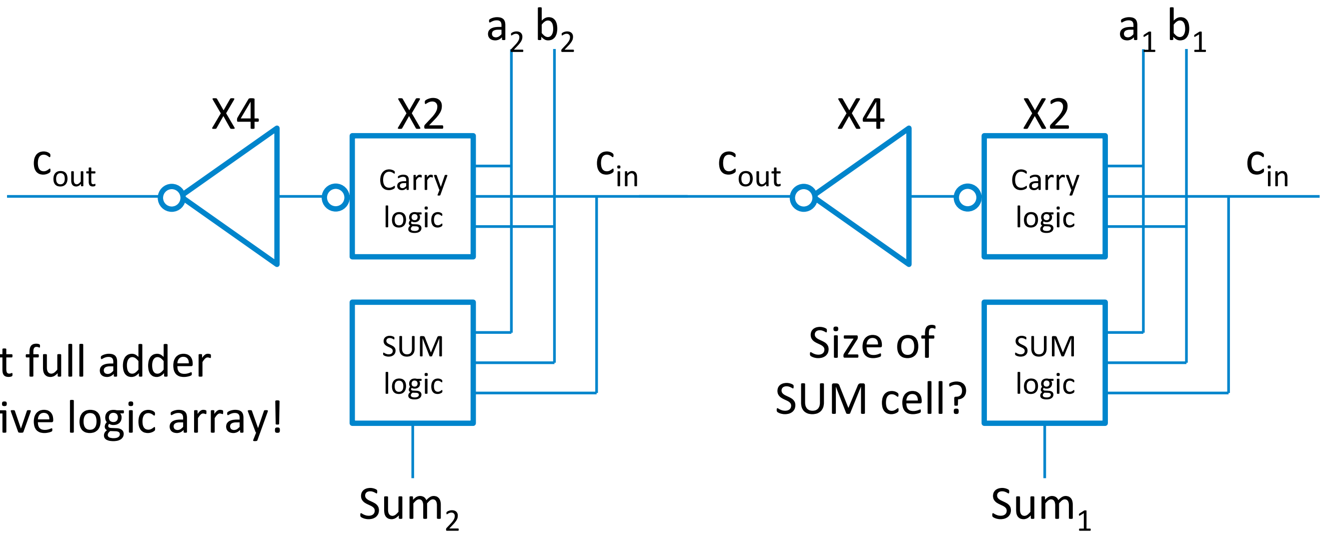
Iterative logic arrays: FULL ADDER

Find Boolean equations from truth table!



BOOLEAN TRUTH TABLE				
A	B	CIN	COUT	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Design full adder by divide and conquer

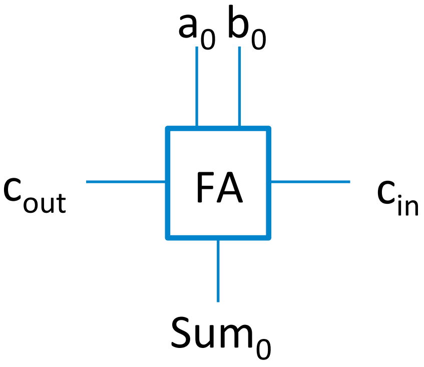


Size of SUM cell?

Design 8-bit full adder as an iterative logic array!

Iterative logic arrays: FULL ADDER

Find Boolean equations from truth table!



Propagation delay estimation:

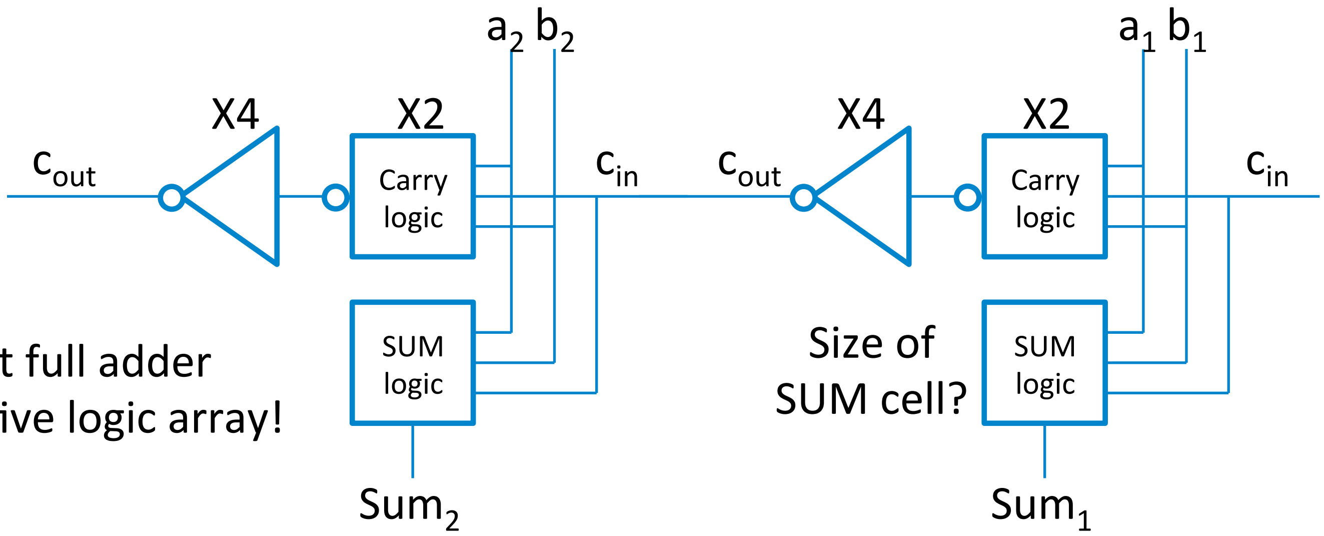
Assume carry cell has logical effort *g* and parasitic delay *p.*

8-bit adder delay: A first crude estimate!

$d=8\times(p+g\times2+p_{inv}+1/2)\approx8\times10=80 \rightarrow t_{pd}= 400$ ps

Corresponding to 1.25 GHz clock frequency

But load from SUM cell was neglected and will slow down the design!

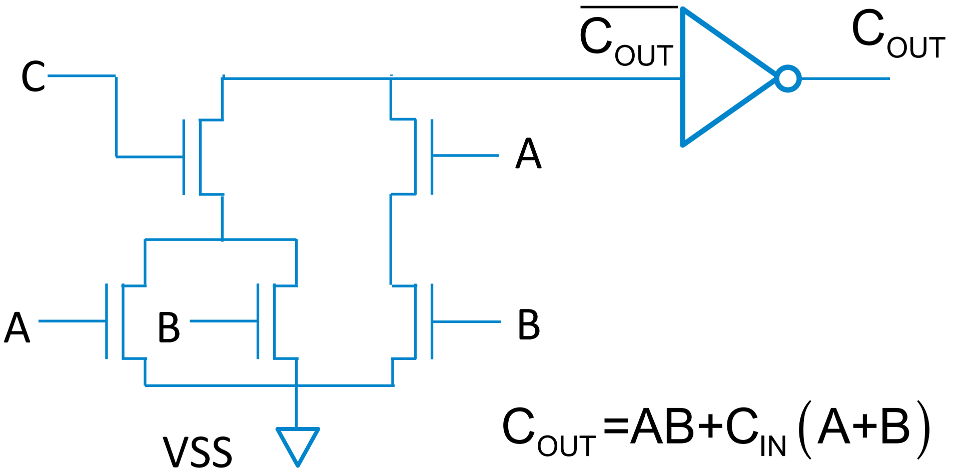


Size of SUM cell?

Design 8-bit full adder as an iterative logic array!

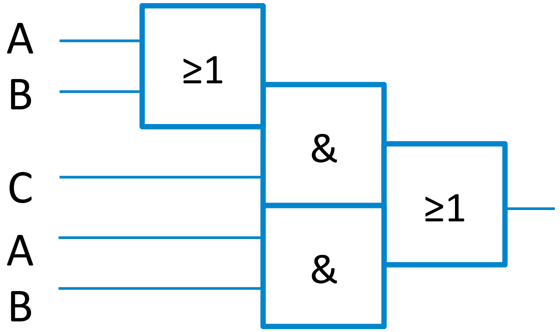
Designing the carry cell

$$\overline{C_{OUT}}=\left(\overline{A}+\overline{B}\right)\left(\overline{C_{IN}}+\overline{A}\,\overline{B}\right)=\overline{A}\,\overline{B}+\overline{C_{IN}}\left(\overline{A}+\overline{B}\right)$$



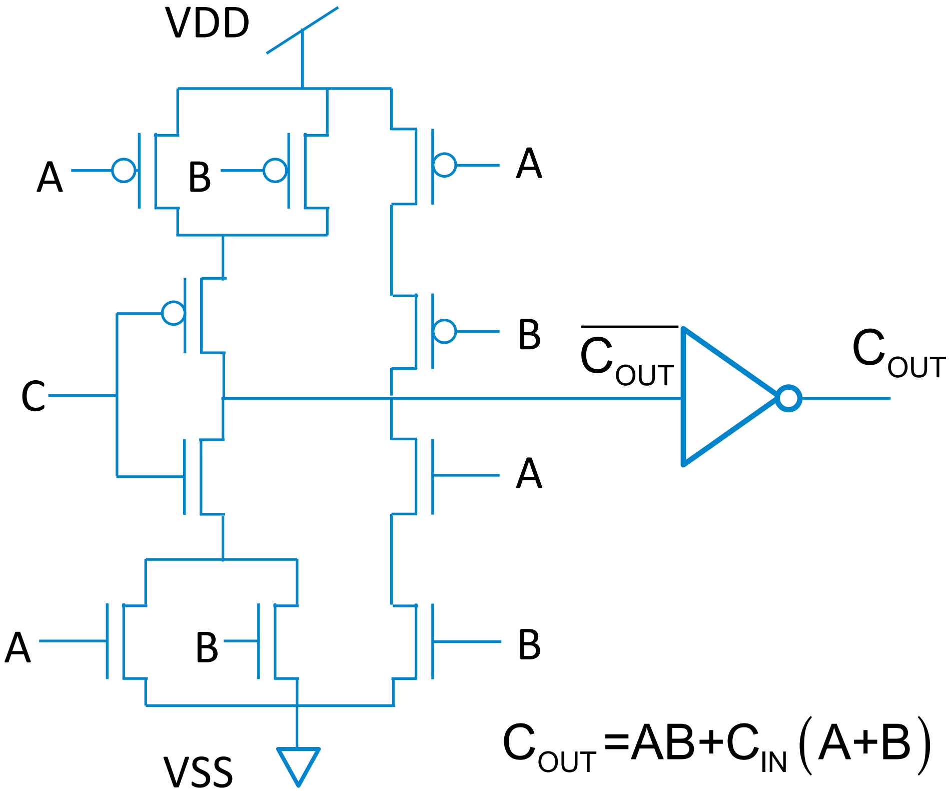
$$C_{OUT}=AB+C_{IN}\left(A+B\right)$$

$$Y=\left(A+B\right)C_{IN}+AB$$

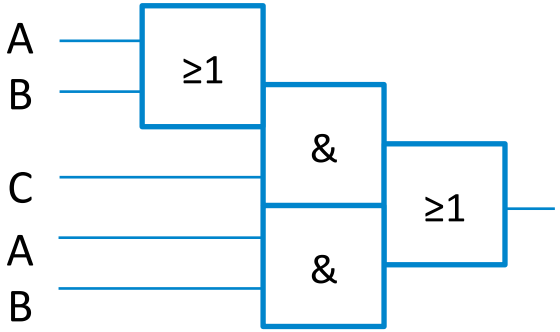


Designing the carry cell

$$\overline{C_{OUT}}=\left(\overline{A}+\overline{B}\right)\left(\overline{C_{IN}}+\overline{A}\,\overline{B}\right)=\overline{A}\,\overline{B}+\overline{C_{IN}}\left(\overline{A}+\overline{B}\right)$$

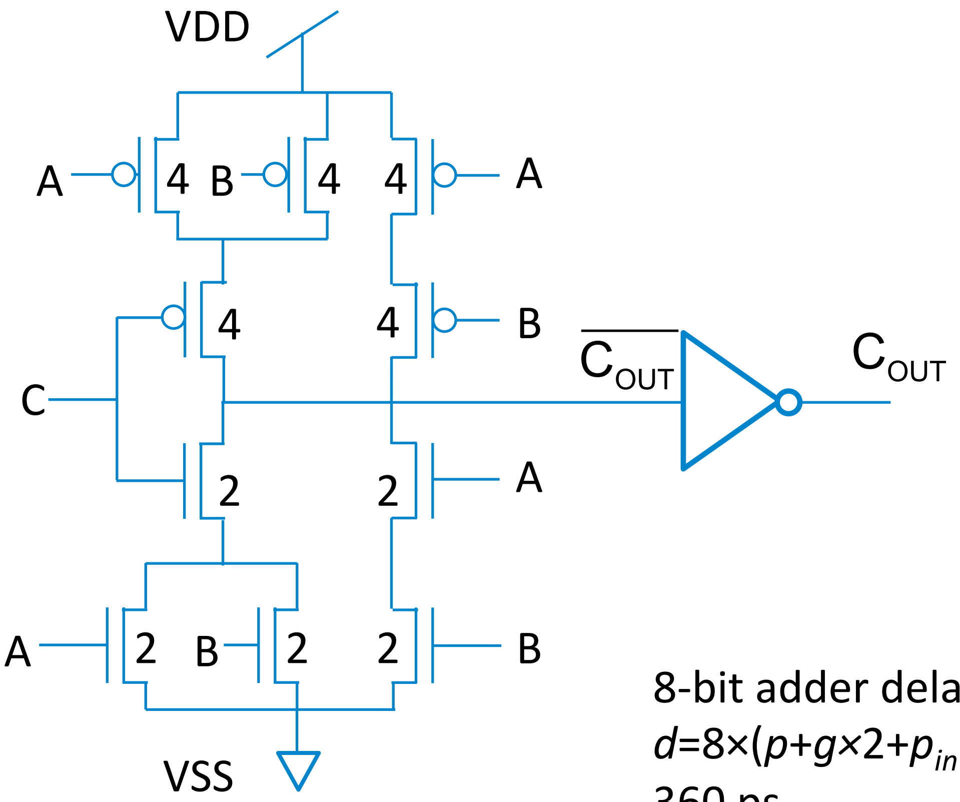


$$C_{OUT}=AB+C_{IN}\left(A+B\right)$$



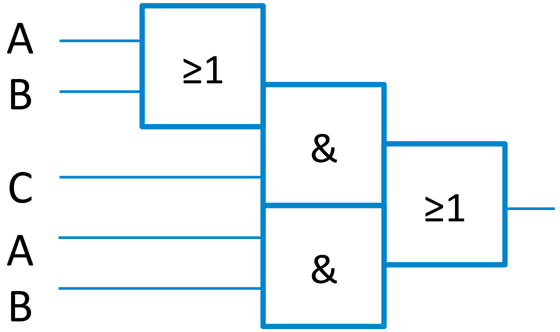
Designing the carry cell

$$\overline{C_{OUT}}=\left(\overline{A}+\overline{B}\right)\left(\overline{C_{IN}}+\overline{A}\overline{B}\right)=\overline{A}\overline{B}+\overline{C_{IN}}\left(\overline{A}+\overline{B}\right)$$



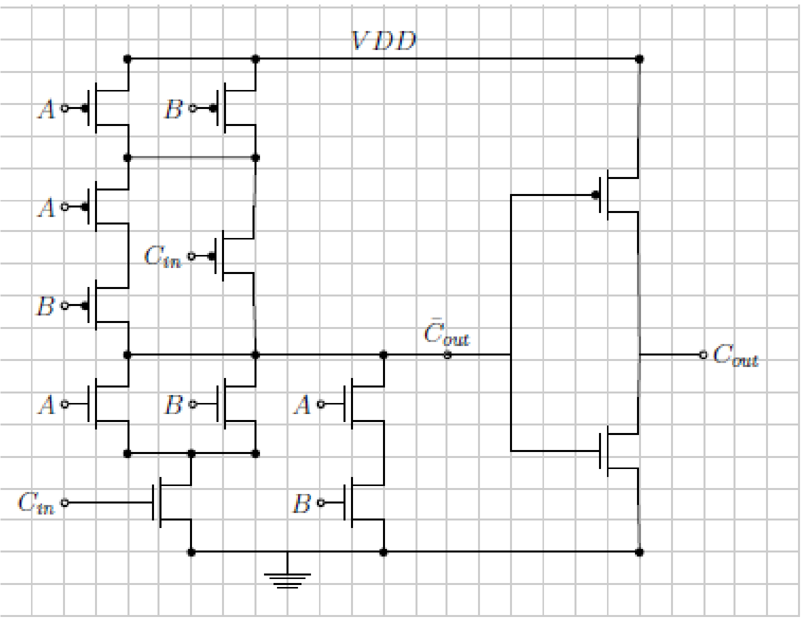
$$C_{OUT}=AB+C_{IN}\left(A+B\right)$$

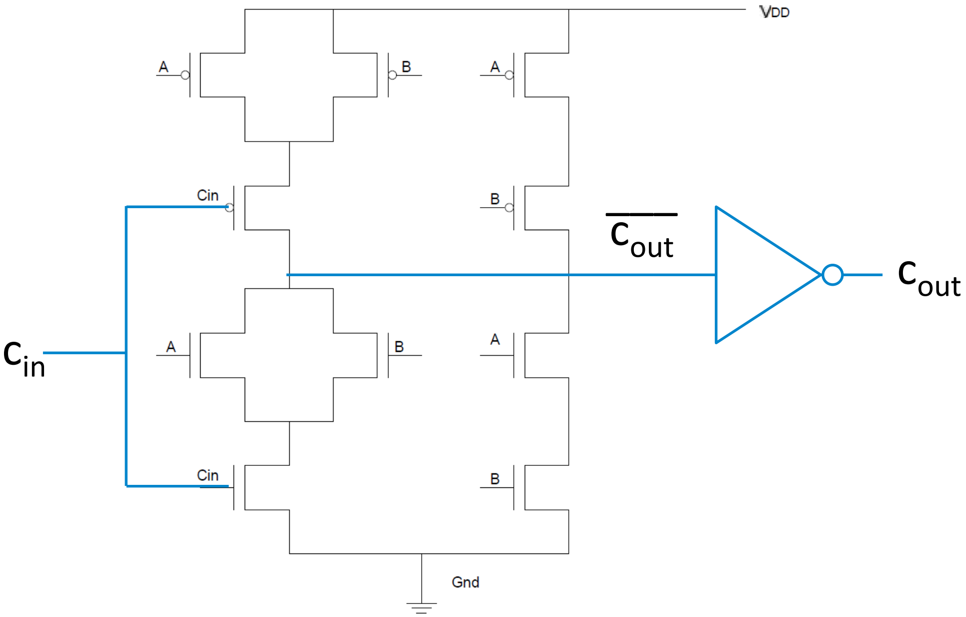
Logical effort g=2,
Parasitic delay $p=4p_{inv}$



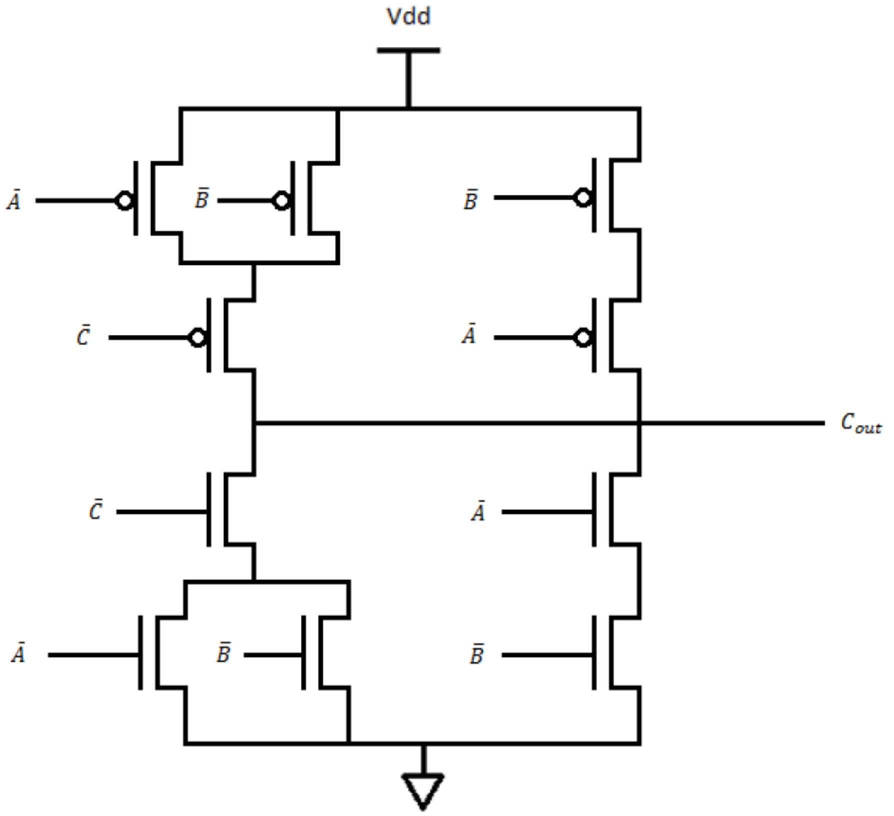
8-bit adder delay: Improved estimate w pinv = 0.9
$d=8\times(p+g\times2+p_{inv}+1/2)\approx8\times(5p_{inv}+2\times2+0.5)=72\rightarrow t_{pd}=$ 360 ps

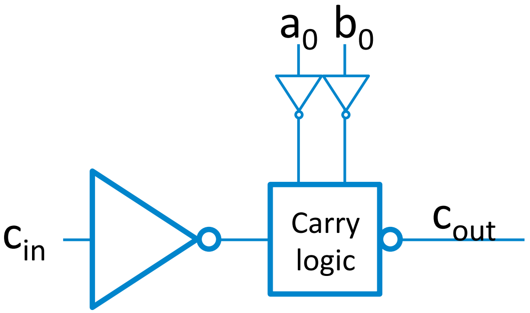
2015 hand-in examples





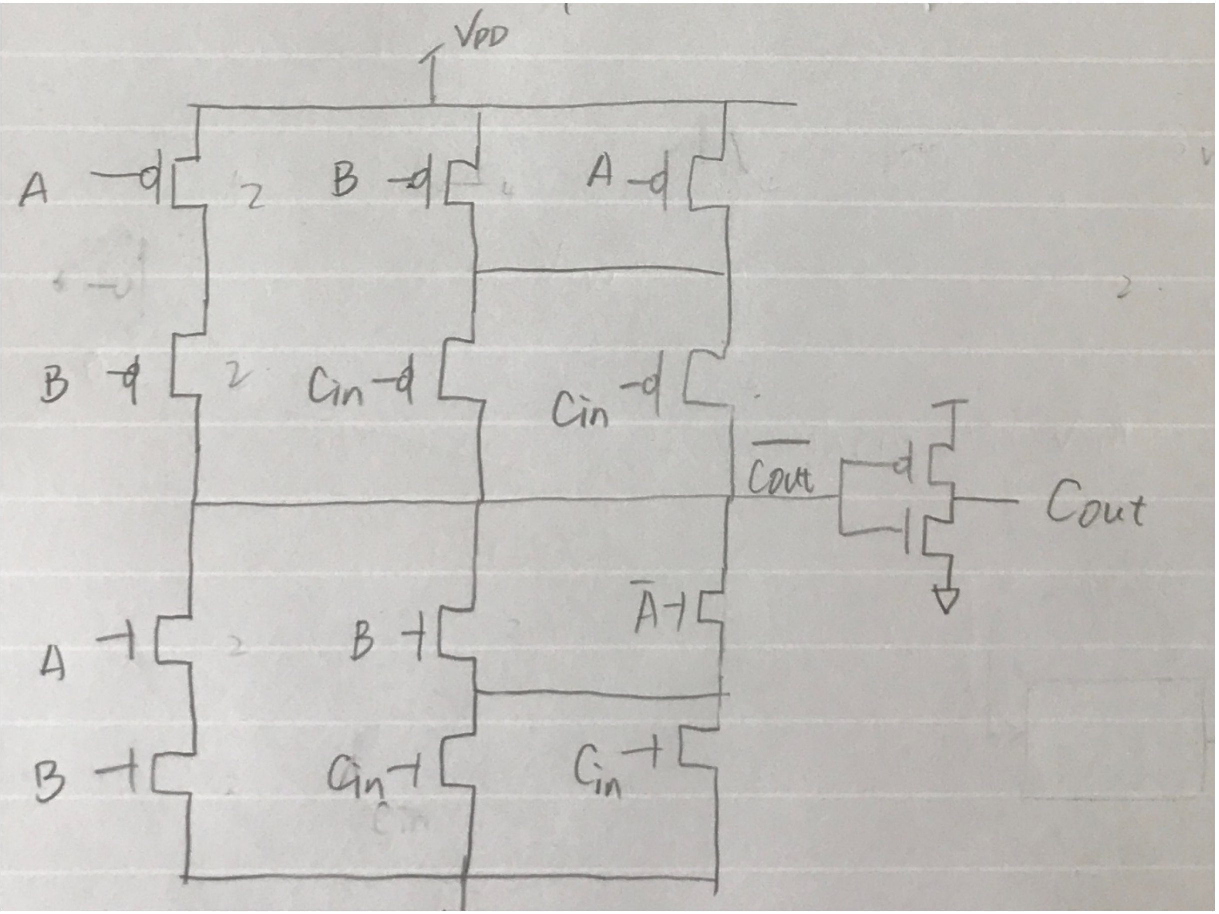
2015 hand-in examples



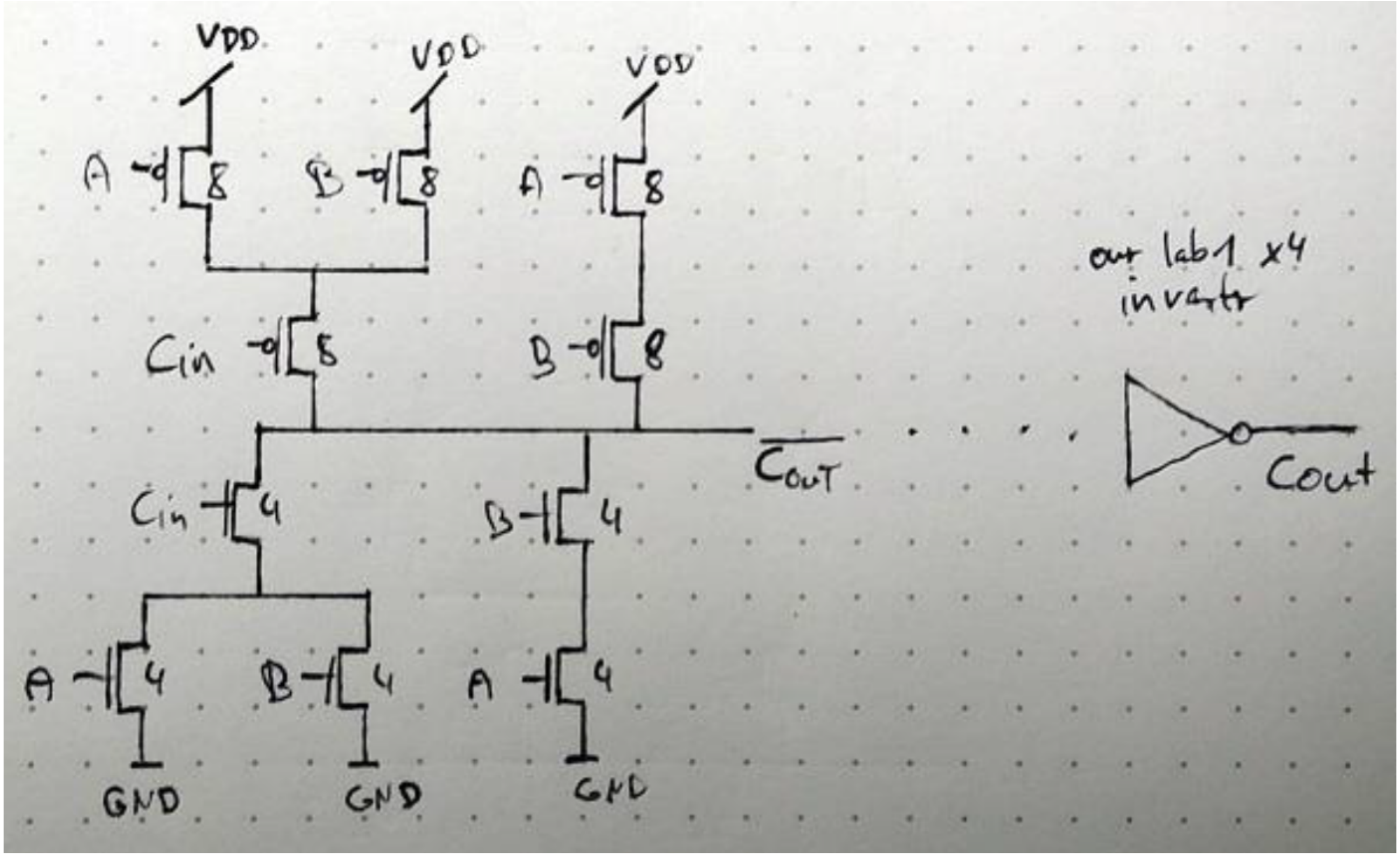


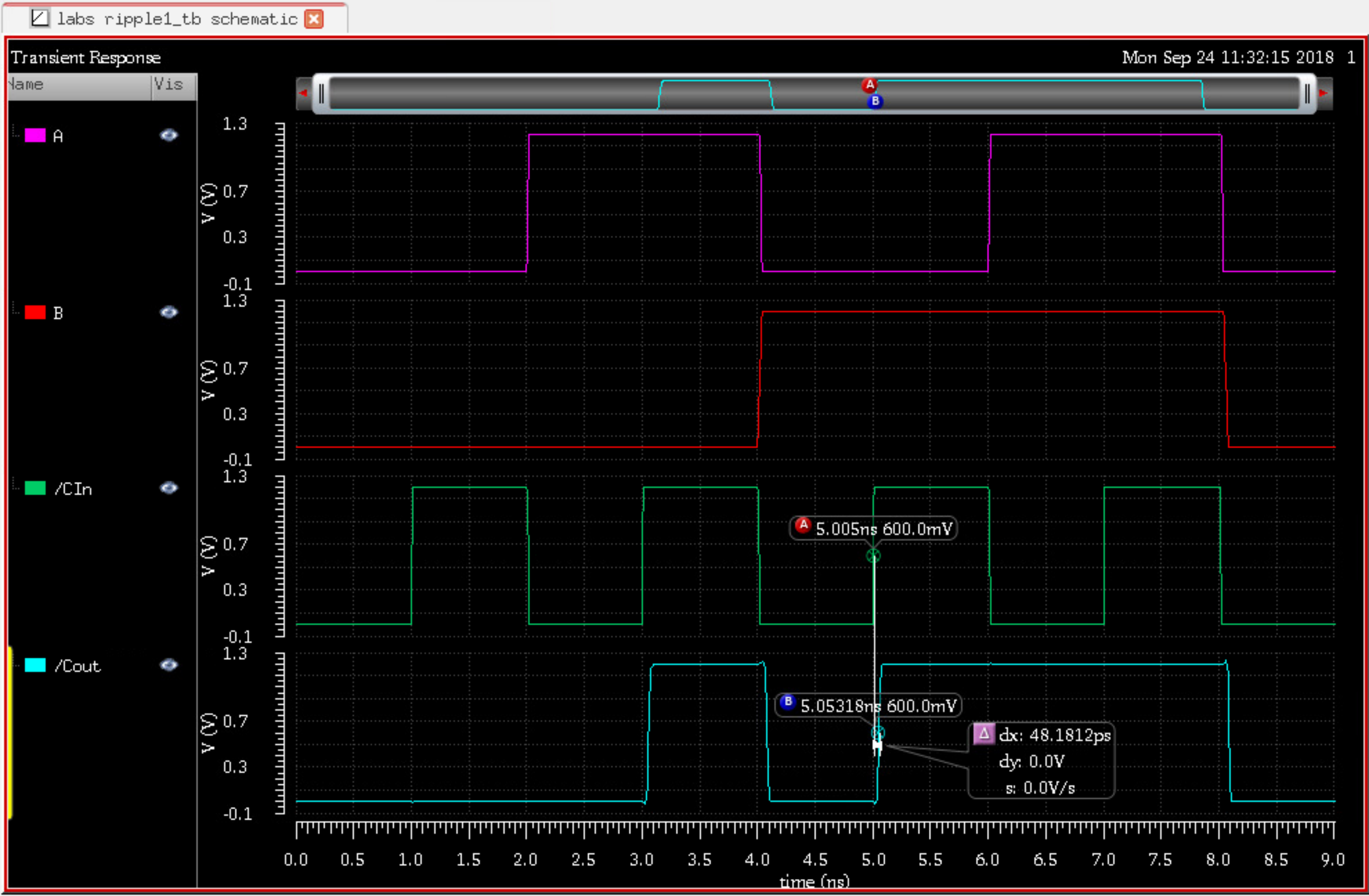
Design structure chosen by a number of 2015 students!

2017 hand-in example



2017 hand-in example





labs ripple1_tb schematic

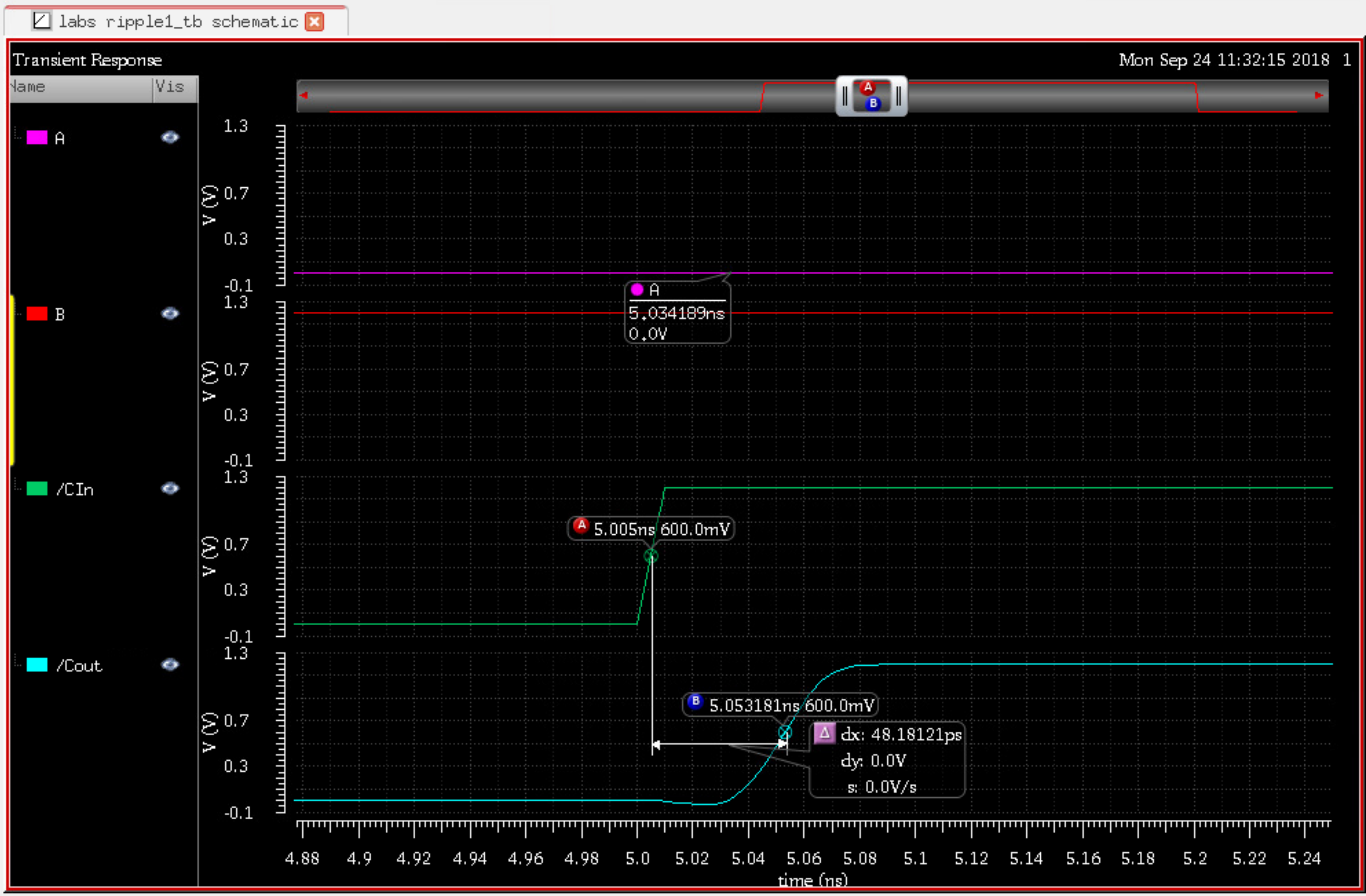
Transient Response

Mon Sep 24 11:32:15 2018 1

A 5.005ns 600.0mV

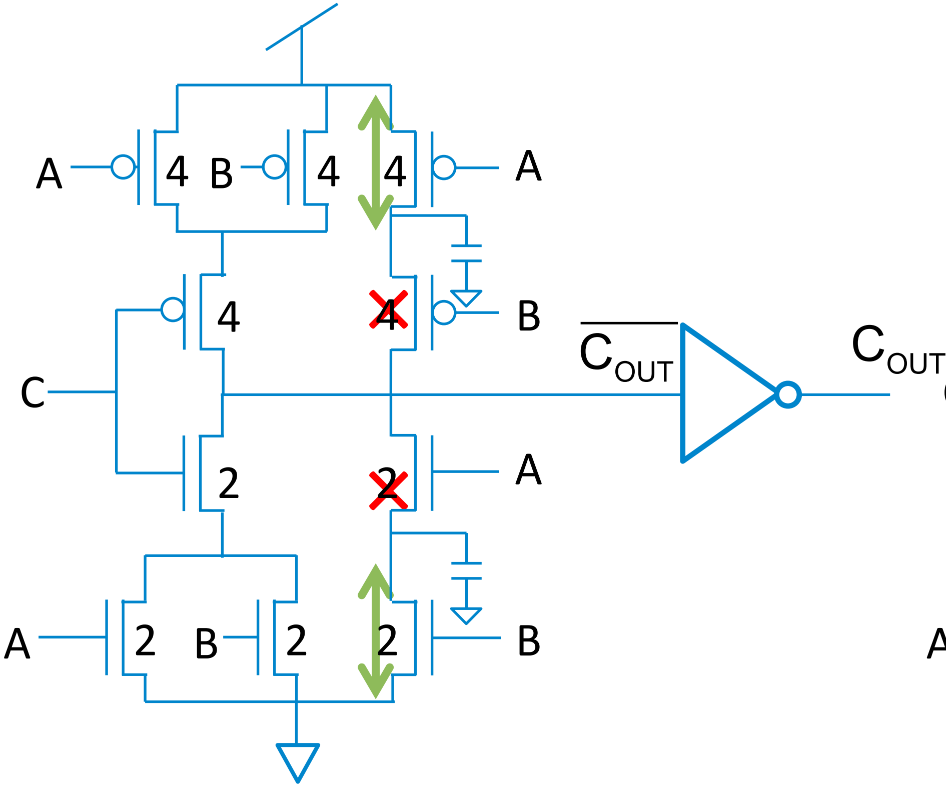
B 5.05318ns 600.0mV

Δ dx: 48.1812ps
dy: 0.0V
s: 0.0V/s

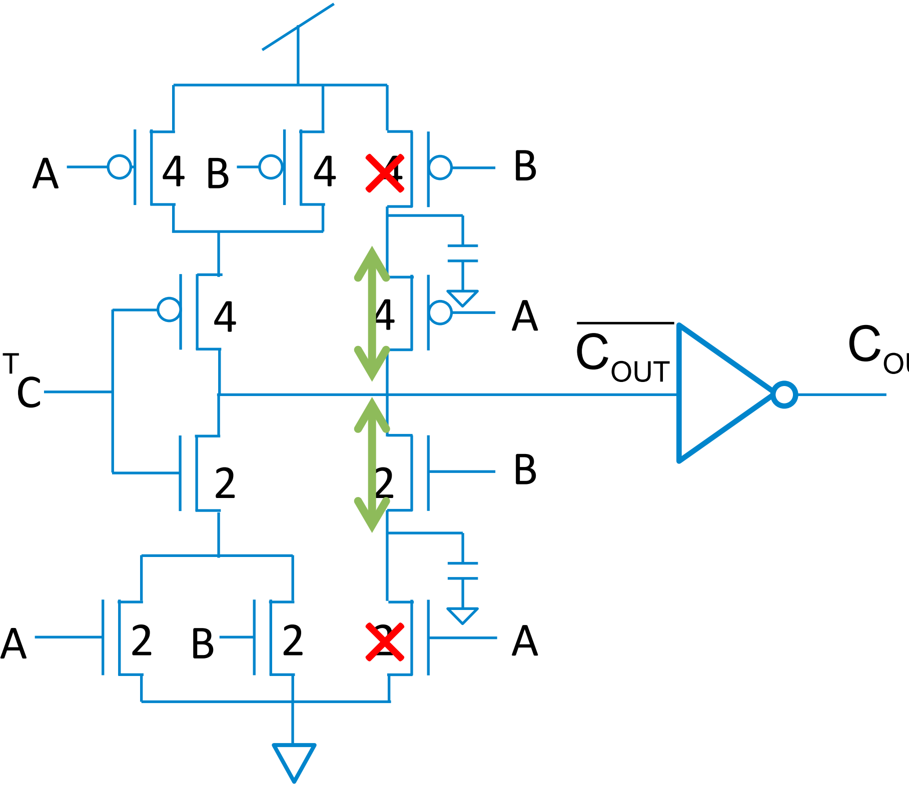

labs ripple1_tb schematic
Transient Response
Mon Sep 24 11:32:15 2018 1
Name
Vis
A
B
/CIn
/Cout
1.3
0.7
0.3
-0.1
V (V)
A
5.034189ns
0.0V
A 5.005ns 600.0mV
B 5.053181ns 600.0mV
Δ dx: 48.18121ps
dy: 0.0V
s: 0.0V/s
4.88
4.9
4.92
4.94
4.96
4.98
5.0
5.02
5.04
5.06
5.08
5.1
5.12
5.14
5.16
5.18
5.2
5.22
5.24
time (ns)

Simulating the carry cell
Impact of test signals

A and B inputs:
B<0:7>= (1,1,1,1,1,1,1,1,)
A<0:7>=(In1,0,0,0,0,0,0,In7)



Expectation: fast



Expectation: slow

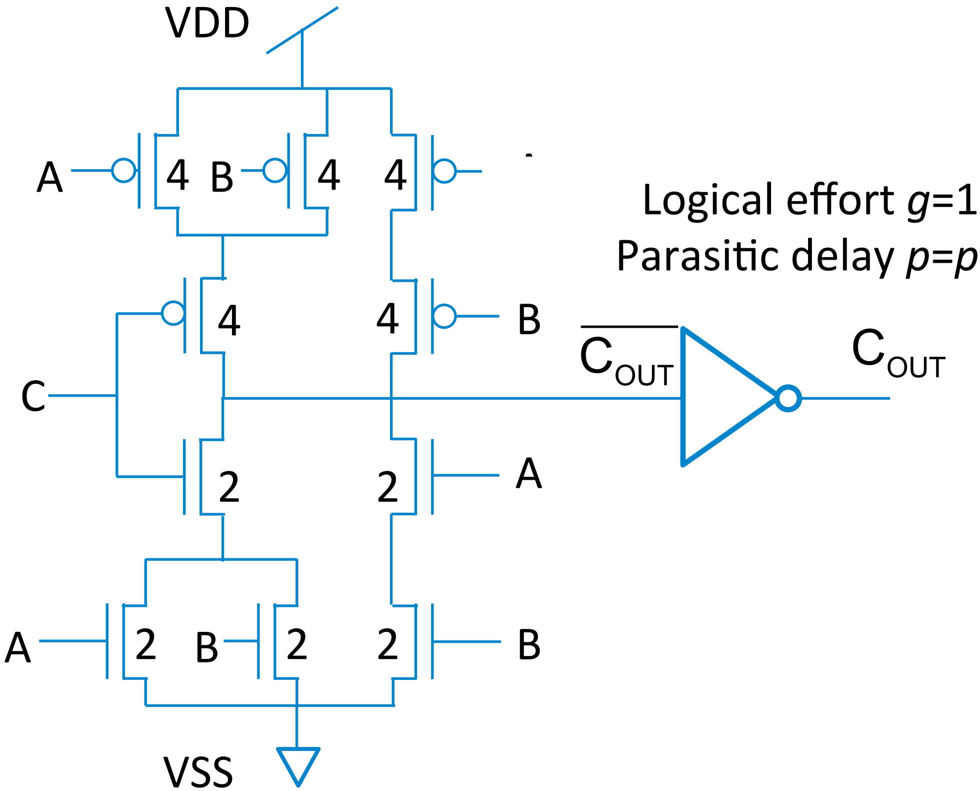
Simulated propagation delays
TT models

Configuration	BAAB	ABBA	BABA	ABAB
Cin-Cout rise [ps]	430	393	453	371
Cin-Cout fall [ps]	446	453	488	413

Conclusion: Symmetrical schematics have less delay variation and the worst-case delay is shorter

Optimizing the size of the inverter

Logical effort *g*=2,
Parasitic delay $p=4p_{inv}$



Logical effort *g*=1,
Parasitic delay $p=p_{inv}$

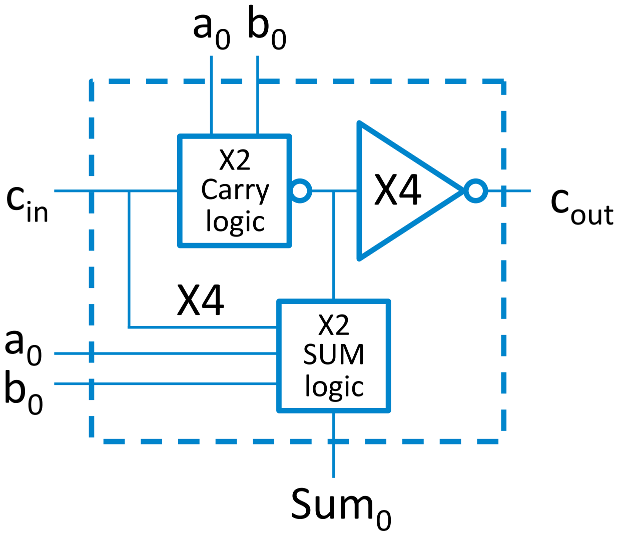
$C_{OUT}=AB+C_{IN}(A+B)$

Optimal path delay:
$G = 2 \times 1 = 2$
$H = C_L/C_{IN} = C_{IN}/C_{IN} = 1$
$F = 2 \times 1 = 2$
$f_{opt} = \sqrt{2}$

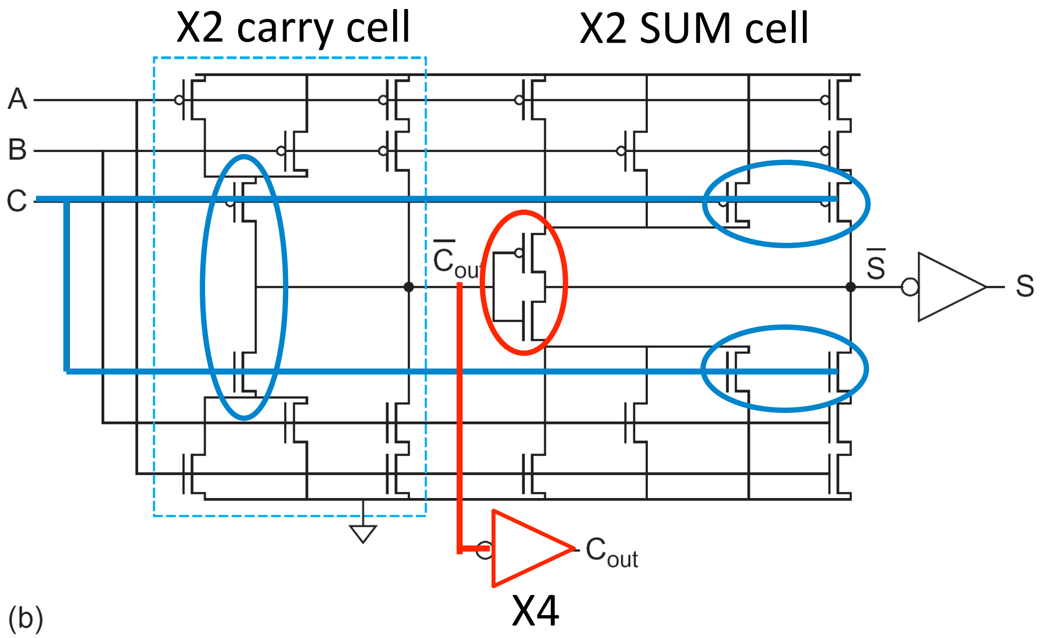
Sizing: Gate √2 larger than inverter

$h_{gate} = 1/\sqrt{2}$
$h_{inv} = \sqrt{2}$

Textbook solution



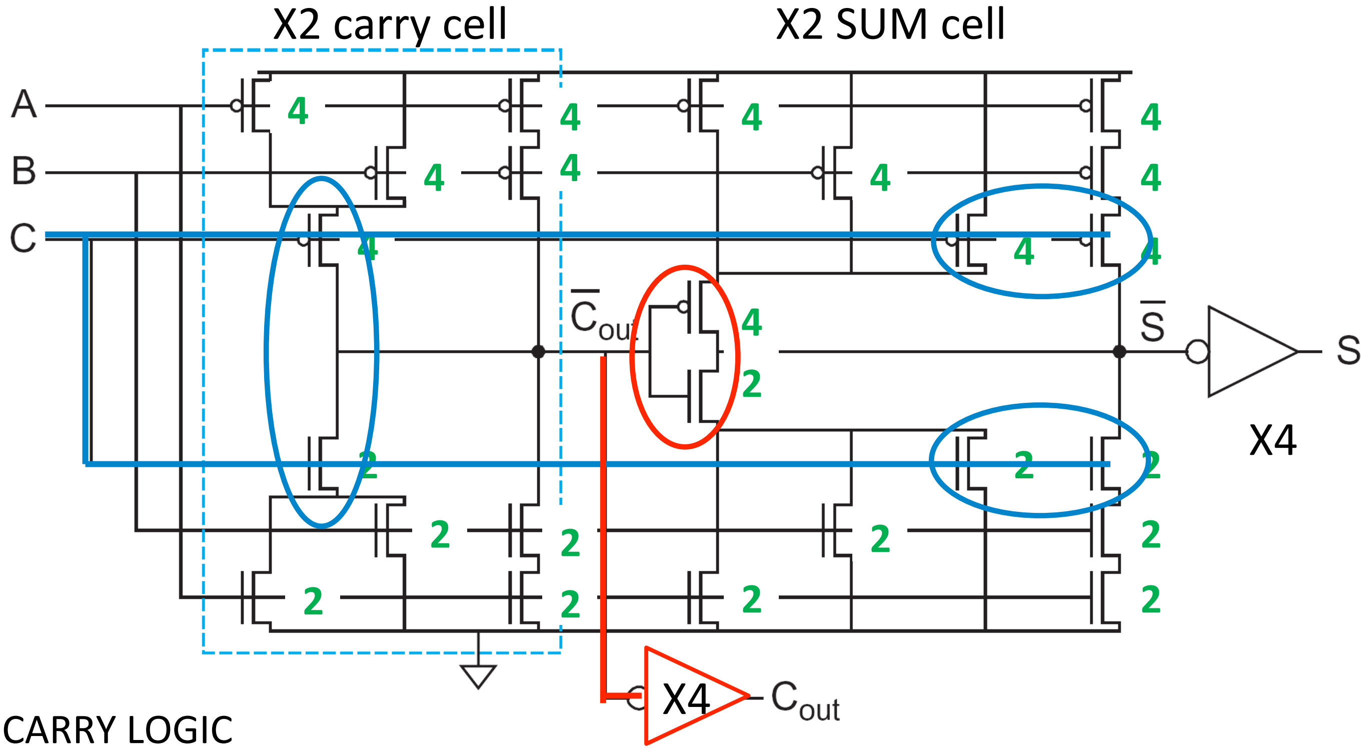
Design structure as proposed in textbook!



Textbook solution

CARRY DELAY (CIN input)

d=4+2x(6+12)/6+1+(6+6+6)/12=4+6+1+1.5=12.5



CARRY LOGIC
(C input)
g=(2+4)/3=2
p=2x(2+4)/3=4

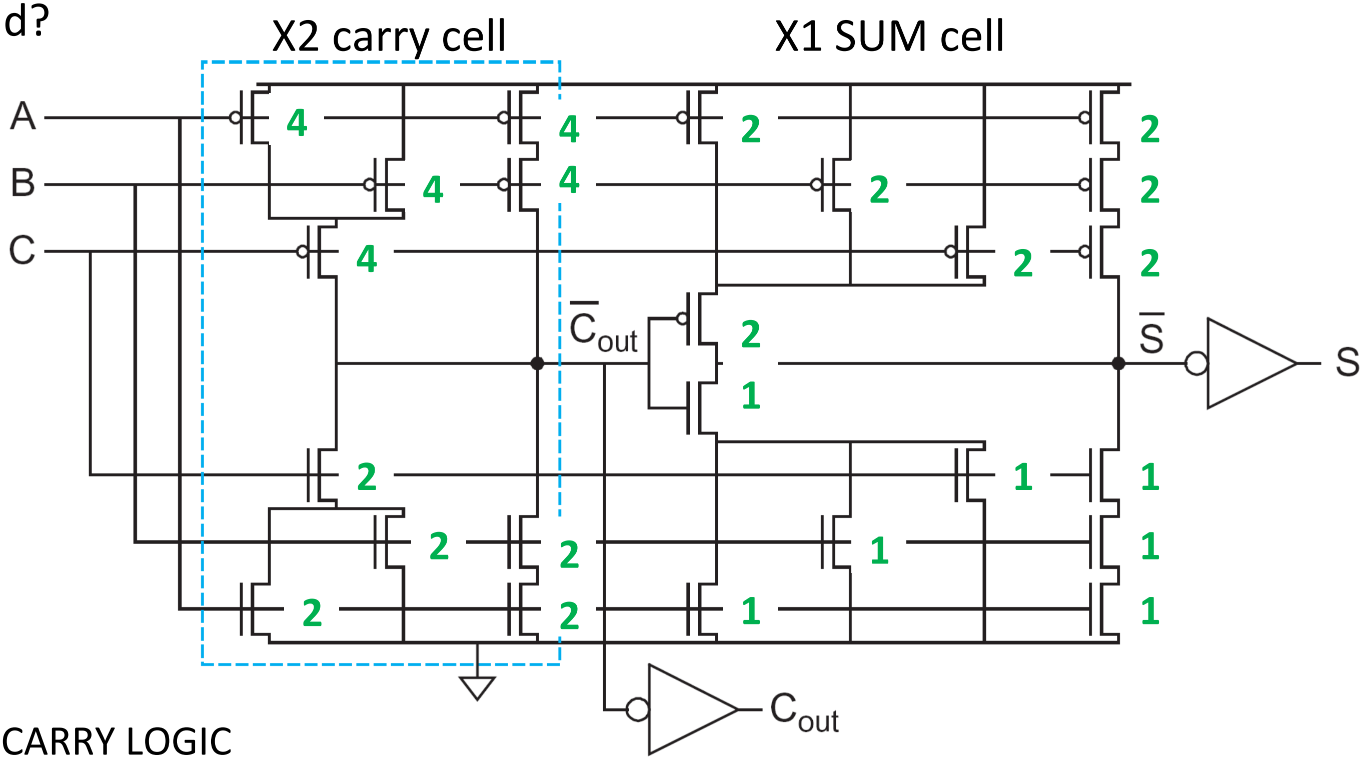
8-bit adder delay: Consider also SUM cell load

d=8×(p+g×(2+4)/2+p_{inv}+3×2/4)≈8×(4+2×3+2.5)=100

→ t_{pd}= 500 ps

Textbook solution

What if we resize MOSFETs in SUM cell for less load?

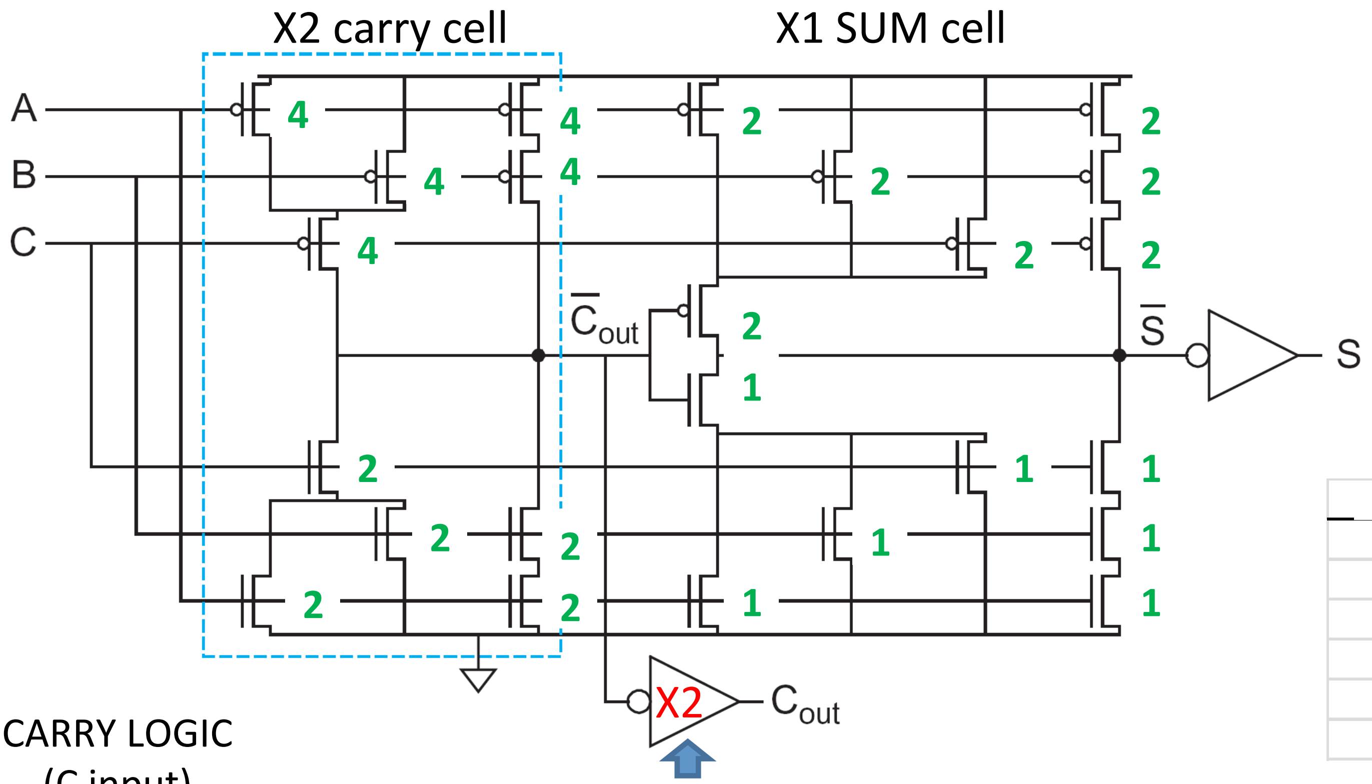


CARRY LOGIC
(C input)
g=(2+4)/3=2
p=2x(2+4)/3=4

8-bit adder delay: Consider also SUM cell load
d=8×(4+2x(1+2)/2+1+(2+1+1))=8×(4+3+1)=8×10=80
→ t_{pd}= 400 ps

Textbook solution

What if we resize
MOSFETs in SUM cell?



X	d
1	11
2	10
4	11
5	11,8
8	14,5
10	16,4

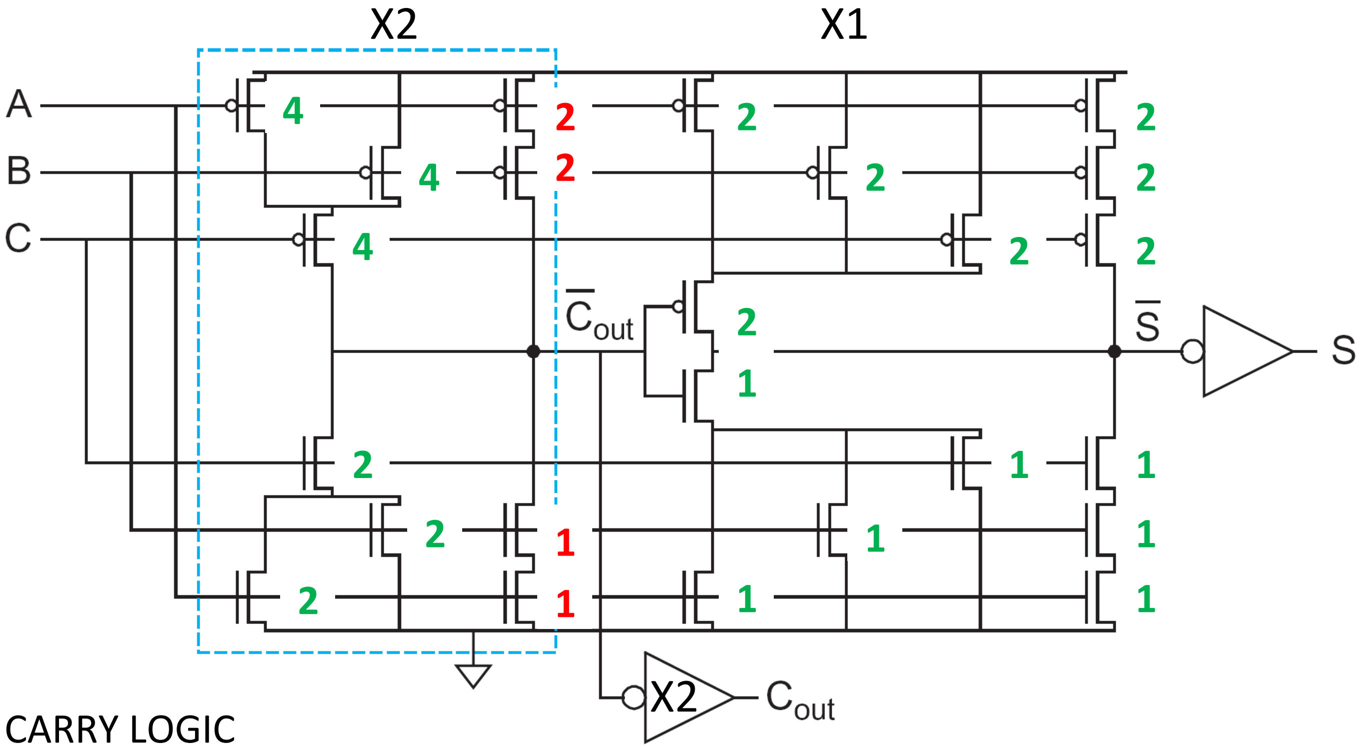
CARRY LOGIC
(C input)
$g=(2+4)/3=2$
$p=2x(2+4)/3=4$

Which size is
optimal? X2? X4?

Textbook solution

CARRY DELAY (CIN input)

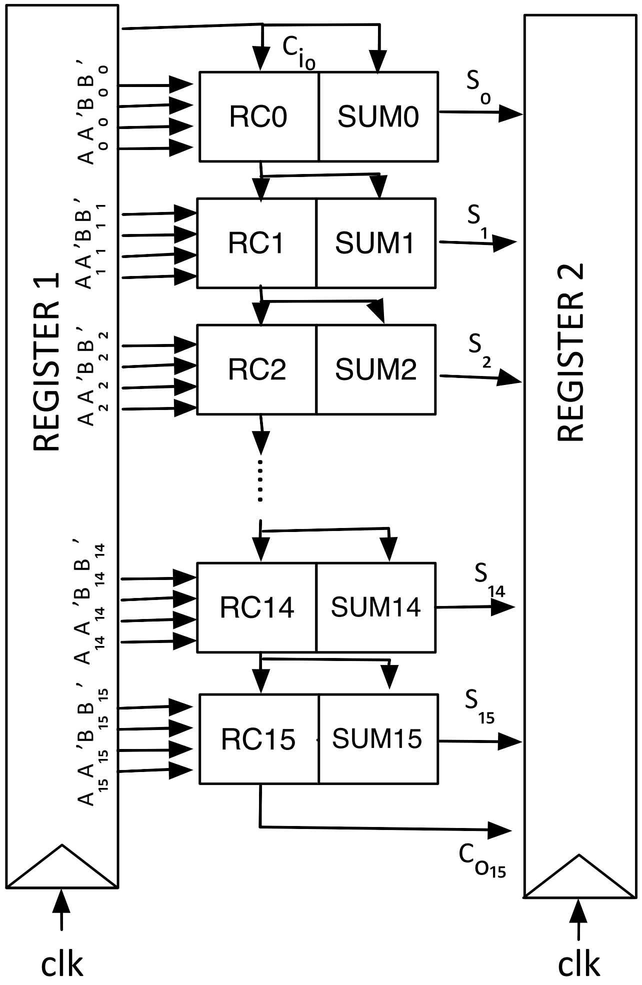
What if we also resize the MOSFETs as shown in red, what is the parasitic delay, *p*, now?



CARRY LOGIC
(C input)
$g=(2+4)/3=2$
$p=(2+4+2+1)/3=3$

CARRY DELAY (C input)
$d=3+2\text{x}(1+2)/2+1+(2+1+1)/2=3+3+1+2=$ 9 units/stage
→ t_{pd}= 360 ps

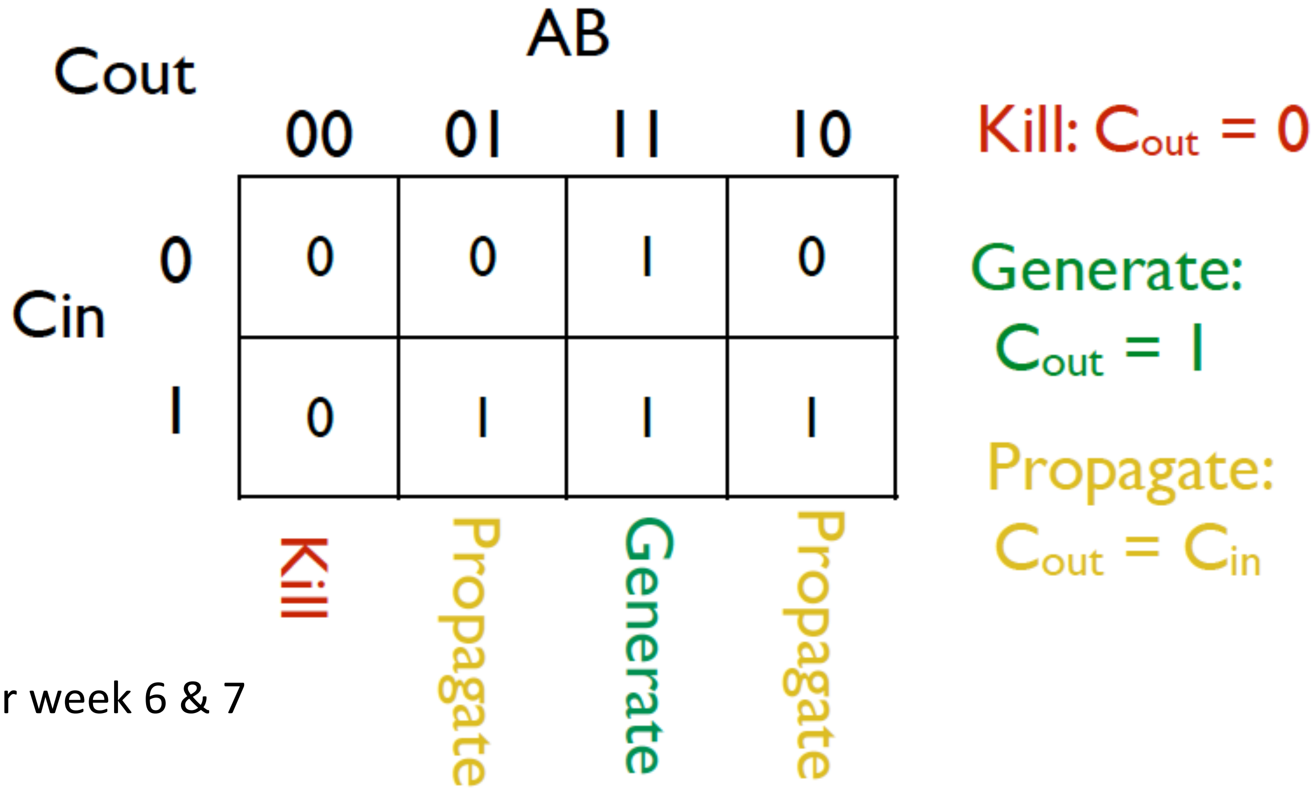
What is the critical path for entire adder?



REGISTER 1
REGISTER 2
A_0 A'_0 B_0 B'_0
A_1 A'_1 B_1 B'_1
A_2 A'_2 B_2 B'_2
A_{14} A'_{14} B_{14} B'_{14}
A_{15} A'_{15} B_{15} B'_{15}
C_{i0}
RC0
SUM0
RC1
SUM1
RC2
SUM2
RC14
SUM14
RC15
SUM15
S_0
S_1
S_2
S_{14}
S_{15}
C_{O15}
clk
clk

What happens at the end of the adder?

Precomputing from A and B simplifies carry (and sum) computing



Cout	Cin \ AB	00	01	11	10
Cin	0	0	0	1	0
	1	0	1	1	1
		Kill	Propagate	Generate	Propagate

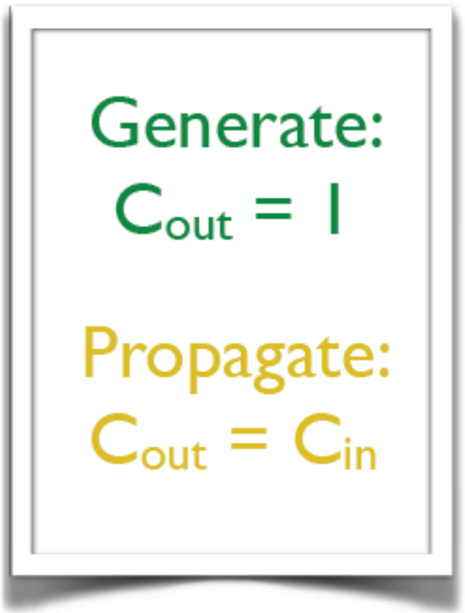
Kill: $C_{out} = 0$

Generate: $C_{out} = 1$

Propagate: $C_{out} = C_{in}$

Preview for week 6 & 7

Precomputing from A and B simplifies carry (and sum) computing

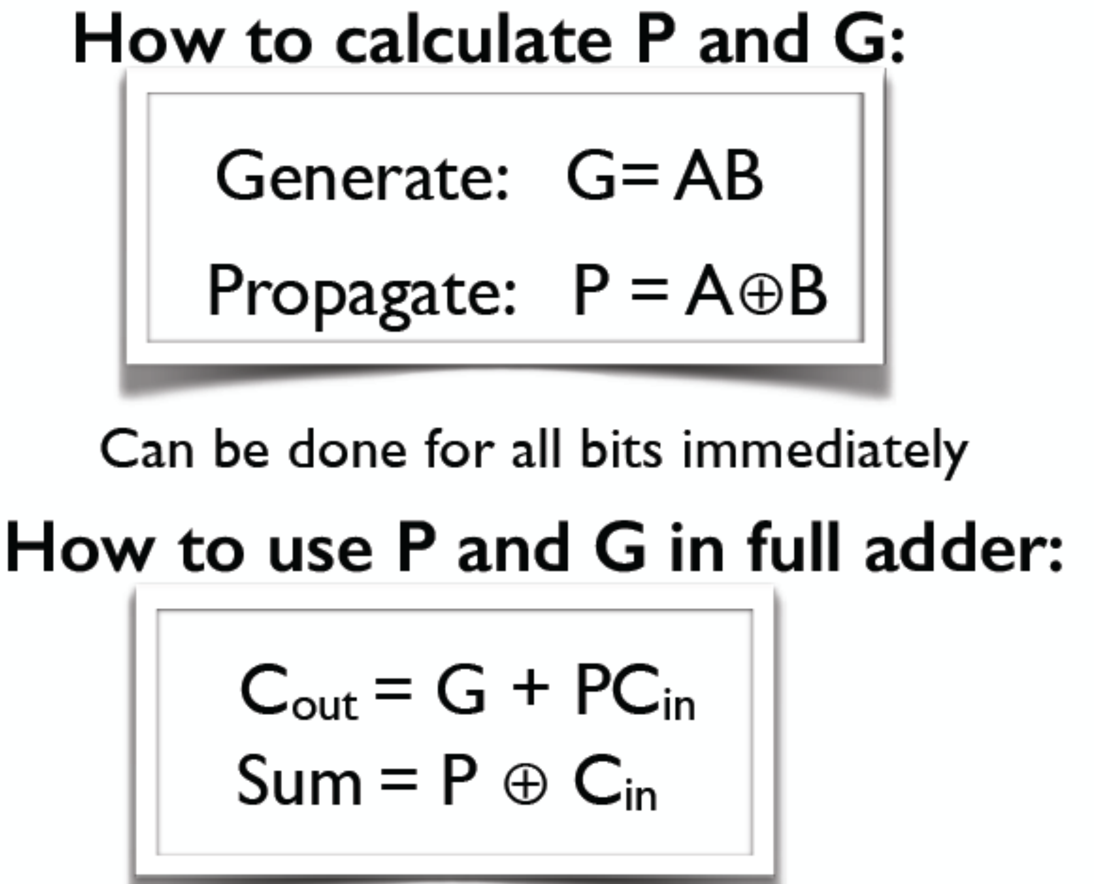


Generate:

$C_{out} = 1$

Propagate:

$C_{out} = C_{in}$



How to calculate P and G:

Generate: $G = AB$

Propagate: $P = A \oplus B$

Can be done for all bits immediately

How to use P and G in full adder:

$C_{out} = G + PC_{in}$

$Sum = P \oplus C_{in}$

Which operation has the longest propagation delay?

Preview for week 6& 7

Conclusion

- The test vectors matter.
  - What is the worst case?
- Optimal sizing of inverter helps
  - Removing inverter entirely would be even better
- Global optimization of critical path
  - Size down circuits not in critical path to limit their capacitive load
- Precomputing “intermediate signals” from A and B also helps

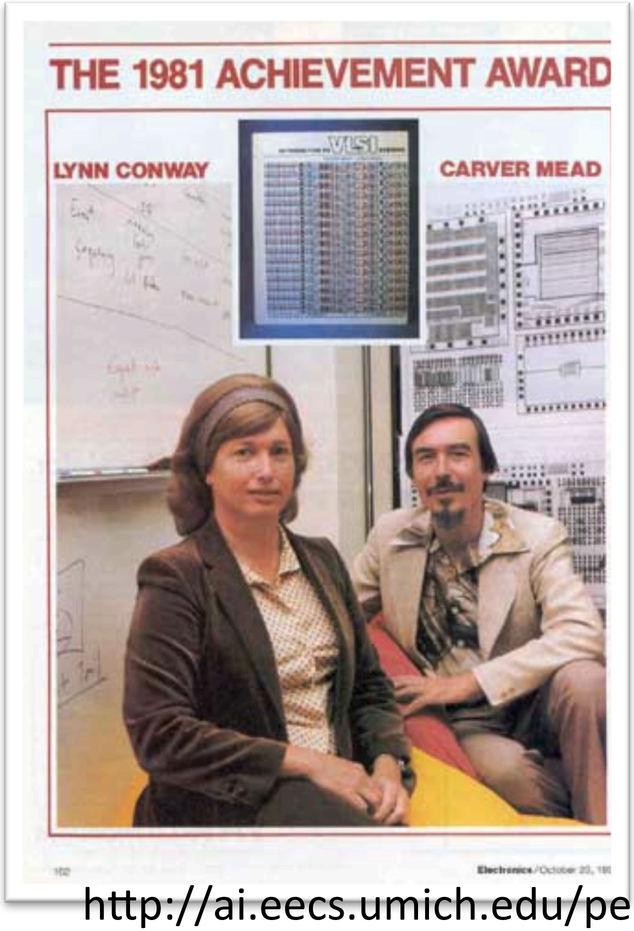
Aim of the lecture

- To give some basic understanding of intralayer and interlayer geometric design rules
  - Intralayer rules concerns minimum dimensions and spacing between objects in **the same** layer
  - Interlayer rules concerns minimum spacing and overlap rules between objects in two **different** layers:
    - MOSFET rules between poly and active
    - Contact and via rules between contacts or vias to their bottom and top contacting layers
- Discuss the prelab assignment for the layout lab session

Why rules?

- To minimize the risk of faulty circuits after fabrication.
  - Malfunction due to shorts
  - Improper forming of components or contacts
  - Electrical charging during fabrication.
- More complex nowadays due to:
  - Smaller feature sizes
  - More complex fabrication processes
    - For example polishing for each metal layer

Who invented the simple (lambda) CMOS rules?



THE 1981 ACHIEVEMENT AWARD

LYNN CONWAY

CARVER MEAD

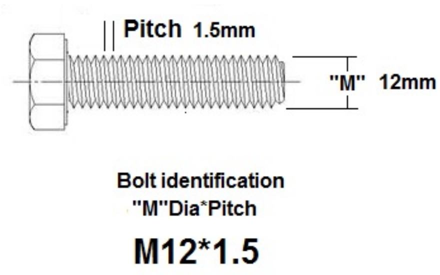
102

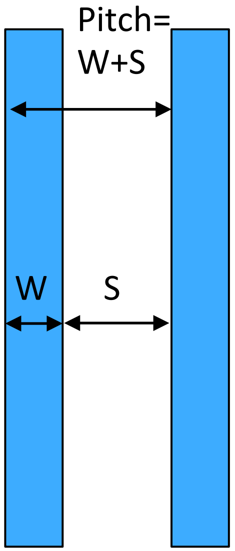
Electronics/October 20, 198

Carver Mead
Lynn Conway

http://ai.eecs.umich.edu/people/conway/Memoirs/VLSI/SSCM/VLSI_Reminiscences.pdf

Geometric design rules

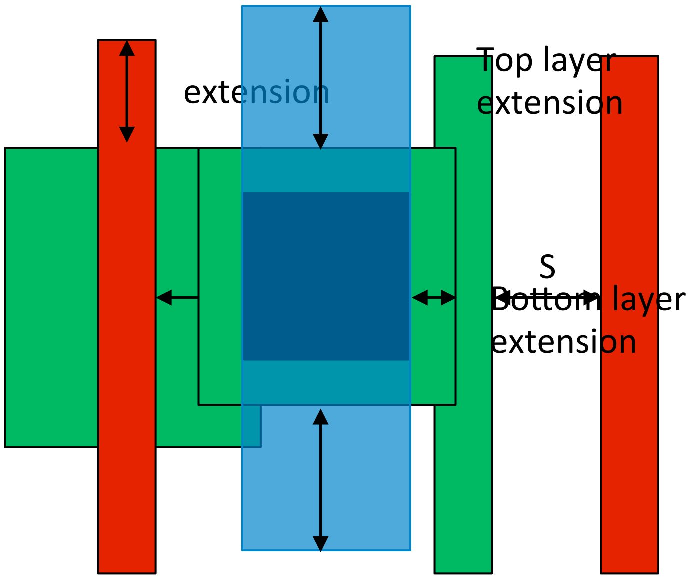




pitch = width + separation

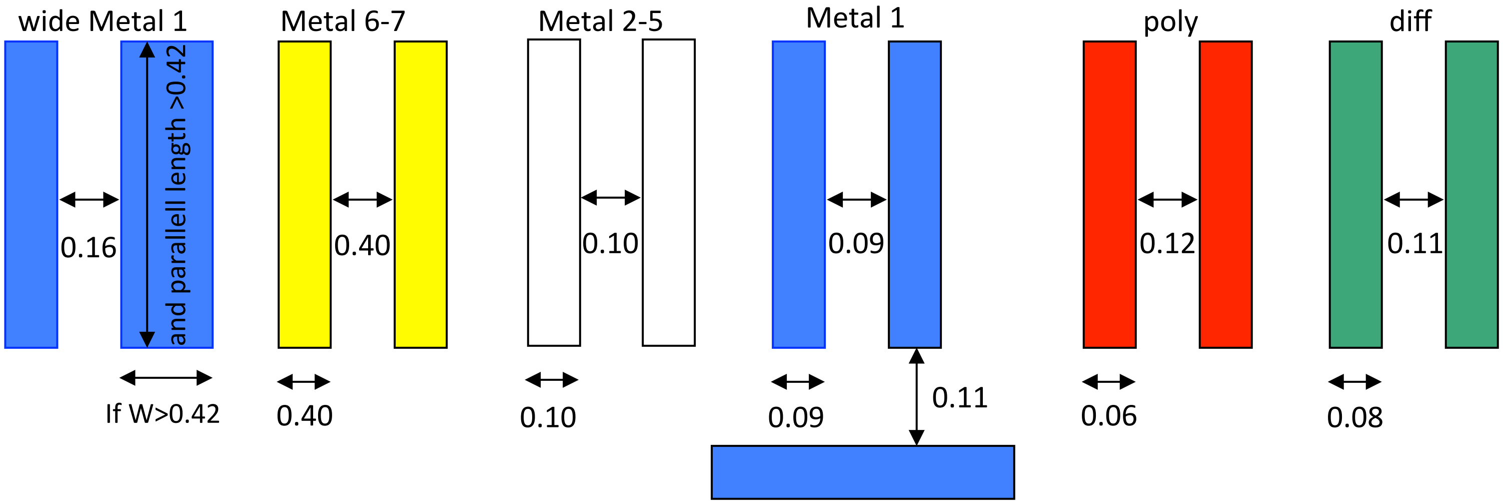
Interlayer rules

Active to poly
Contacts and vias



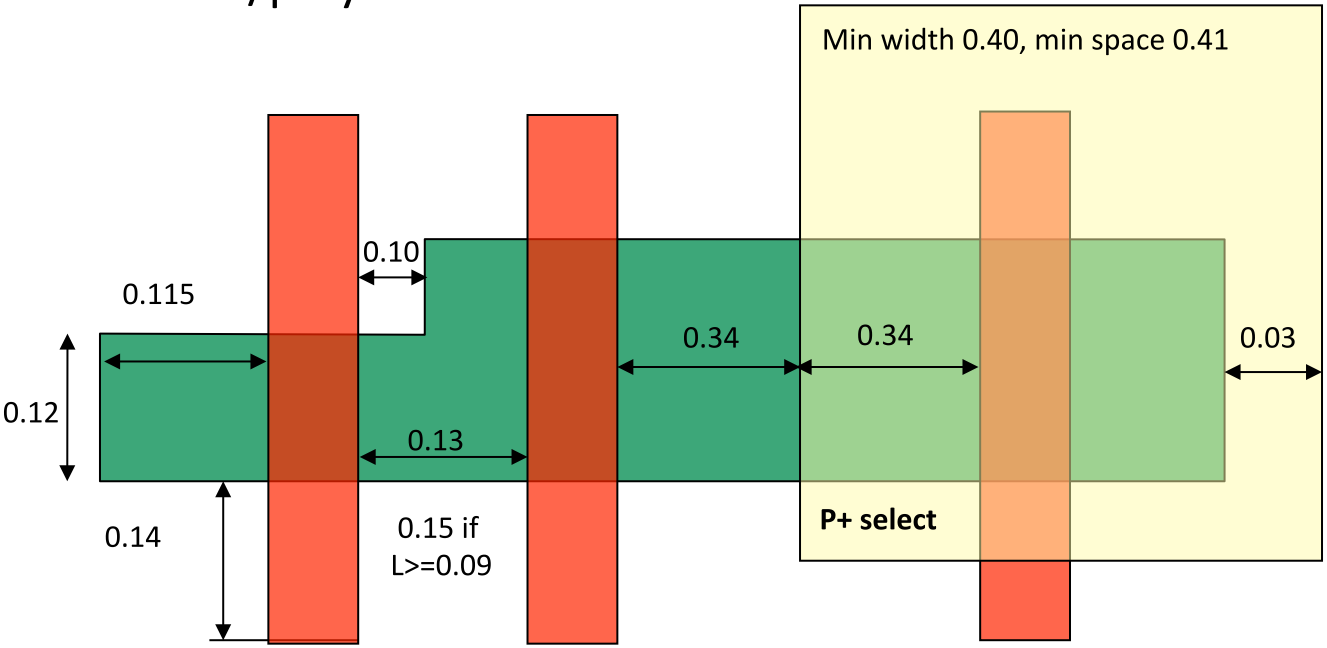
Geometric design rules

Intralayer rules



Geometric design rules

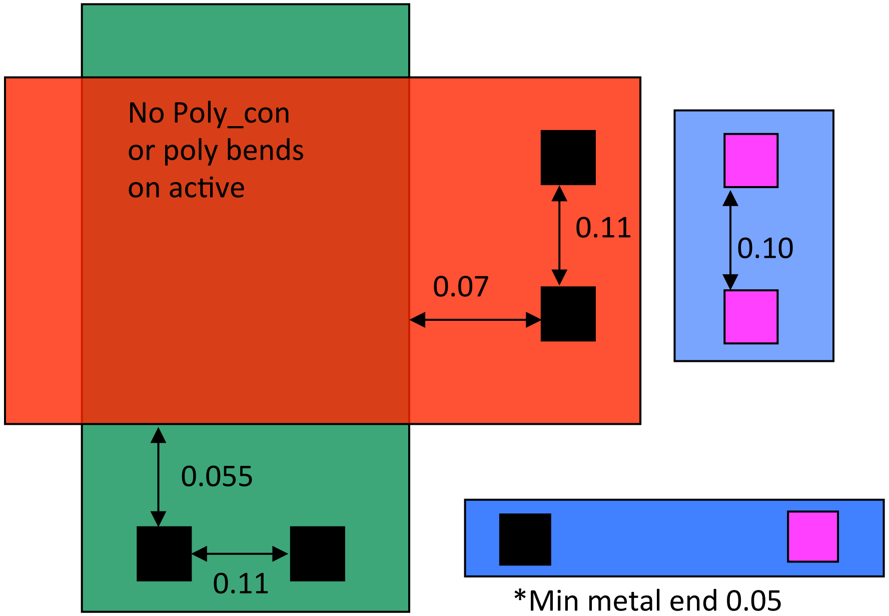
Details of active/poly rules



Geometric design rules

Contact and via* rules

*A via is a contact between two metal layers.

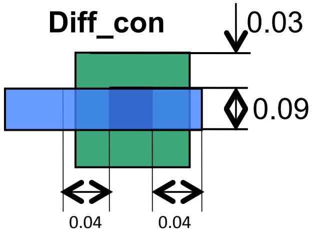


*Min metal end 0.05

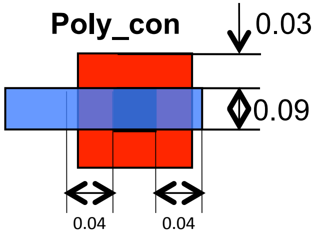
Geometric design rules

Contact and via rules

Diff_con



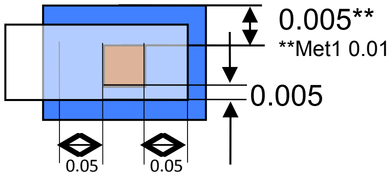
Poly_con



Also allowed to enclose contact with metal1 0.025 on all four sides.

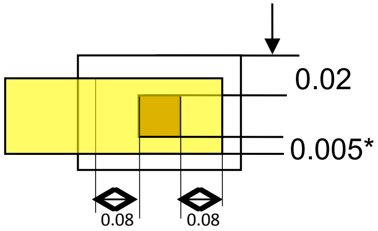
For minimum width metal lines at least 0.04 enclosure is required on two opposite sides (bottom measures).

Via 1-4: 0.10x0.10

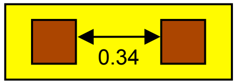


At least two 0.05 metal ends on metal pad

Via 5-6: 0.36x0.36

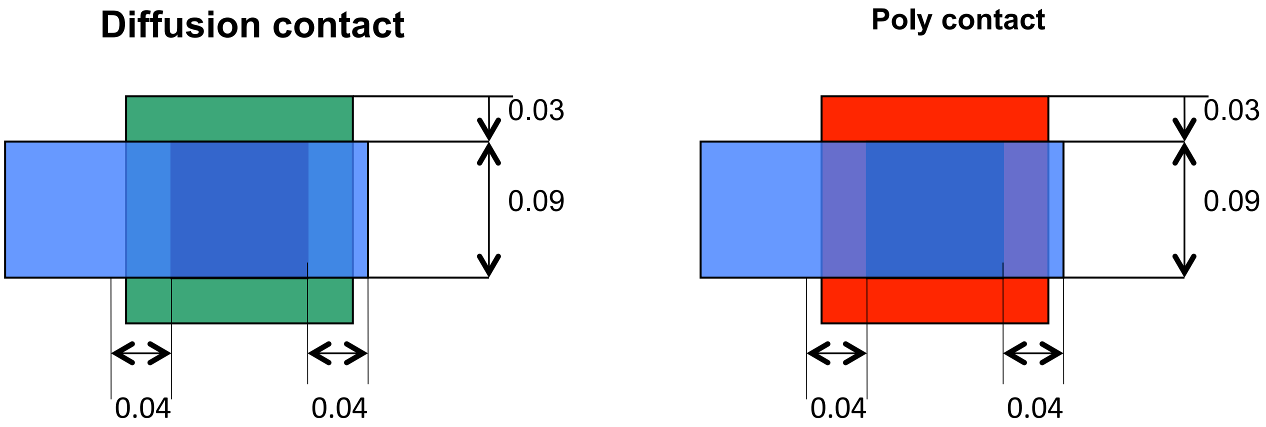


At least two 0.08 metal ends on metal pad



(min 0.54 in array)

Detailed contacts



The contacts drawn here are the ones we recommend.
For minimum width metal lines (width = 0.09) at least 0.04 enclosure is required on two opposite sides, while on remaining two sides enclosure is 0.0.

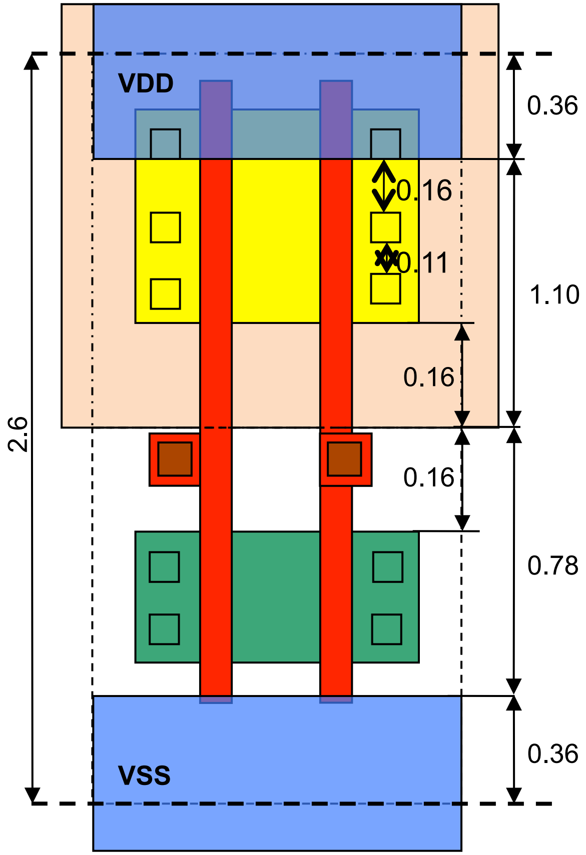
Options:
It is also allowed to have metal1 enclose contact 0.025 in all four directions.

Diffusion rectangular: extension has to be 0.03 on two opposite sides and 0.015 on the two remaining sides.

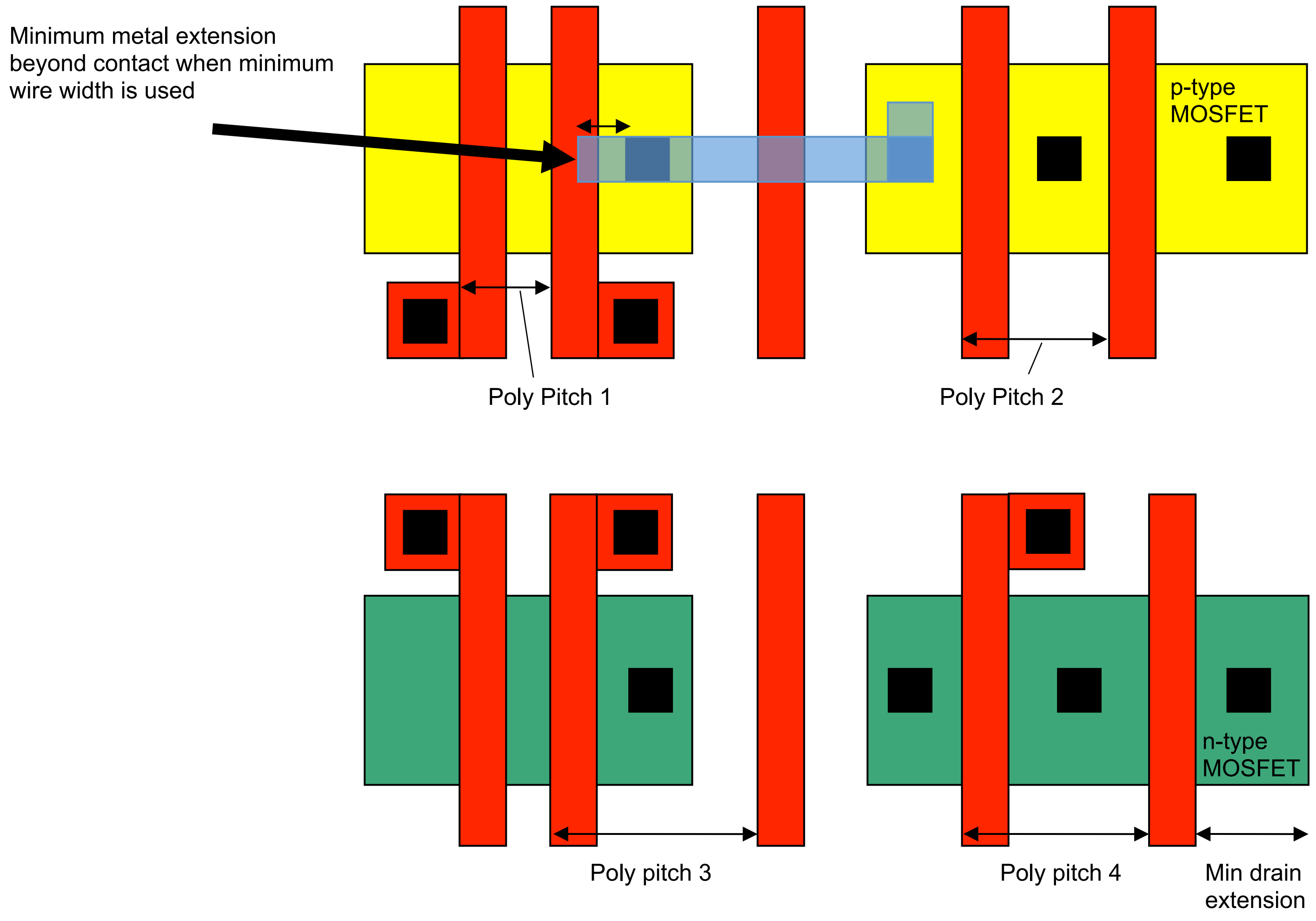
Poly rectangular: extension has to be 0.04 on two opposite sides and 0.01 on the two remaining sides.

Geometric design rules

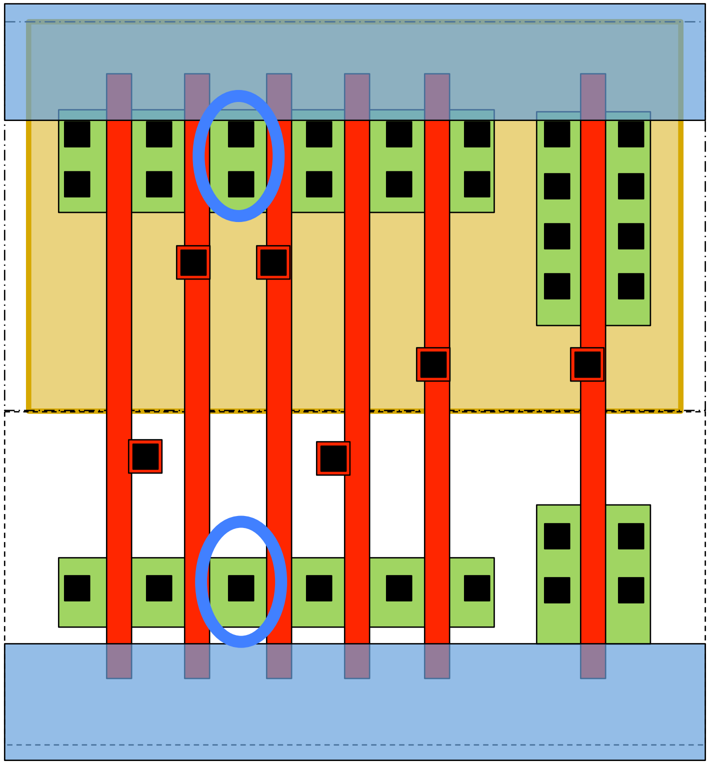
Standard cell template
with rails, n-well, pplus and nplus design rules

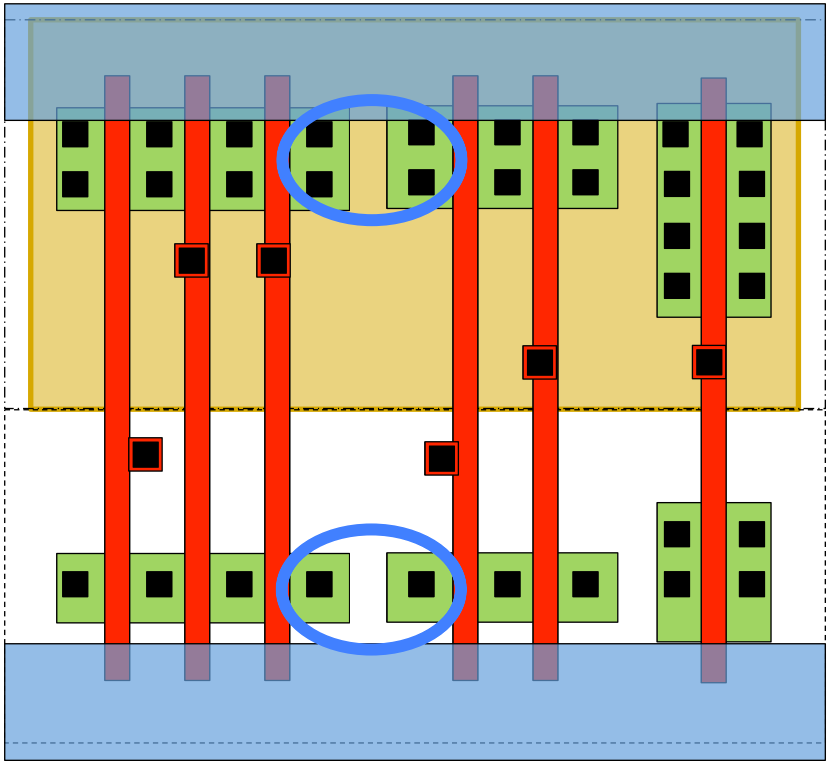


Prelab 3 assignment

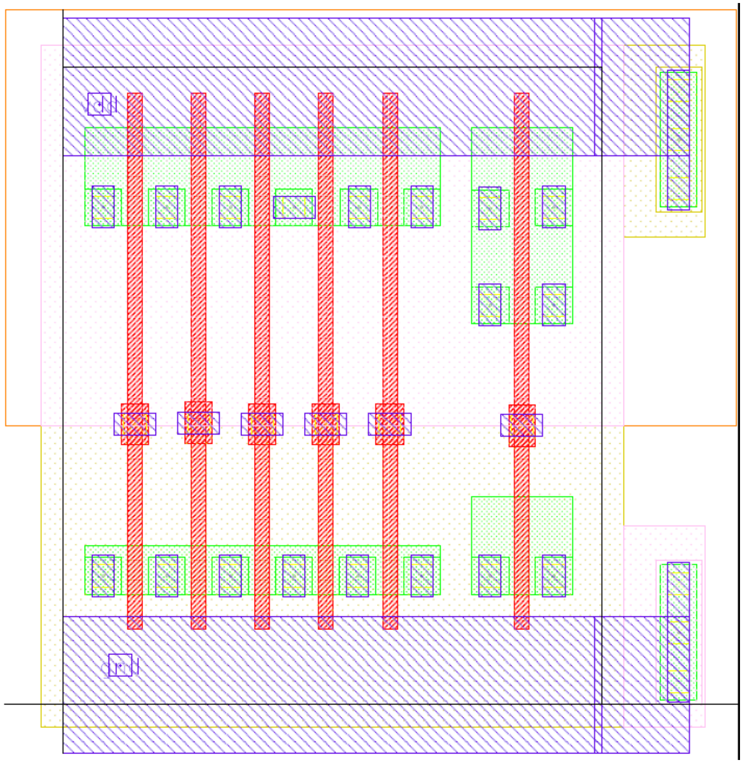


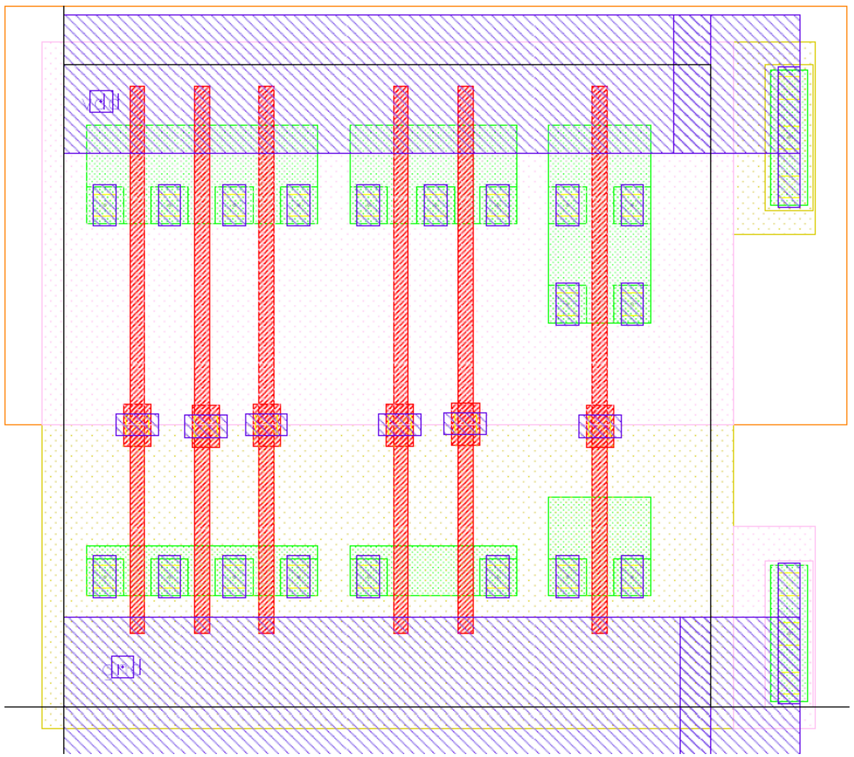
Prelab 3 layout templates





Prelab 3 layout templates



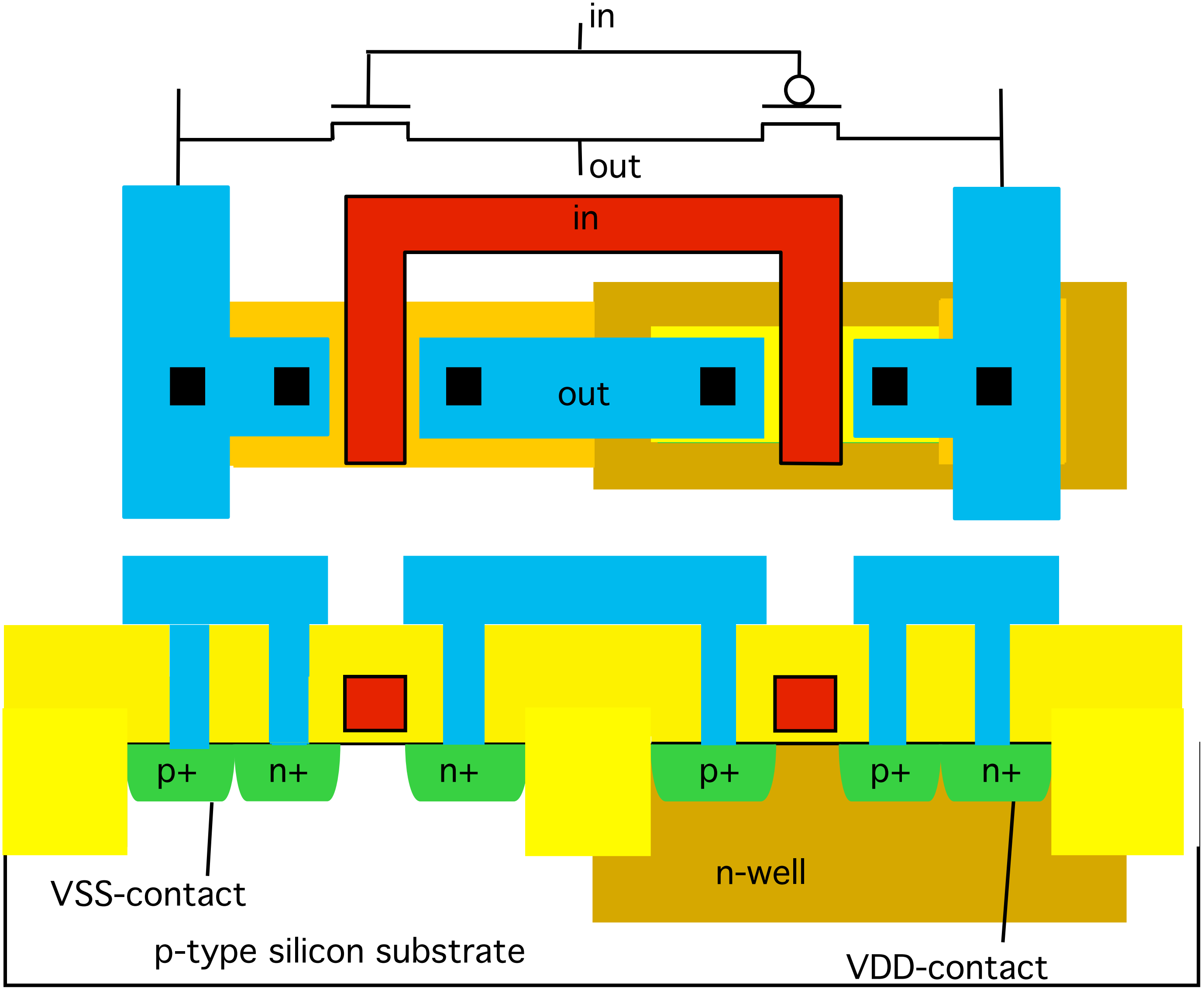


The layers

- **Physical layers** in your layout:
  - N-well (NW)
  - Diffusion/Active (OD)
  - Poly (PO)
  - **Metal-1 (M1)**
  - Contact hole (CO)
- **Non-physical layers** required in your layout:
  - P-select (PPLUS) (pink dots)
  - N-select (NPLUS) (yellow dots)

Example of layers

CMOS Layout



Well-mask

Active-mask
Shallow trench
Isolation (STI)
poly-mask

N⁺-select-mask
N⁺-diffusion

P⁺-select-mask
P⁺-diffusion

Body-ties

Field oxide

Contact-mask

Contact-etch

Metal-mask

Summary

In this lecture, we have

- introduced intralayer and interlayer geometric design rules
- defined the minimum pitch:
  - the sum of the minimum width and the minimum separation between two objects in the **same** layer
- studied interlayer rules between objects in two different layers, like
  - MOSFET rules between poly and active
  - contact and via rules between contacts or vias to their bottom and top contacting layers
- discussed the prelab assignment to the layout lab session