FUNDAMENTALS OF VLSI DESIGN

(19EC6DCFOV)

Module-2

Circuit Design Processes: MOS layers, Stick diagrams, Design rules and layout – lambda-based design and other rules. Examples. (Text book-1)

CMOS Sub System Design: Introduction, Addition/Subtraction, Single bit addition, Full adder design, Comparators, LFSR,XOR/XNOR circuits (Text book-3)

Text books:

1. Douglas A. Pucknell, Kamran E., "Basic VLSI Design", 3rd Edition, *PHI Publication*, India.
3. Neil H.E. Weste, Harris, Banerjee, "CMOS VLSI design", *Pearson*, Third Edition, 2007.

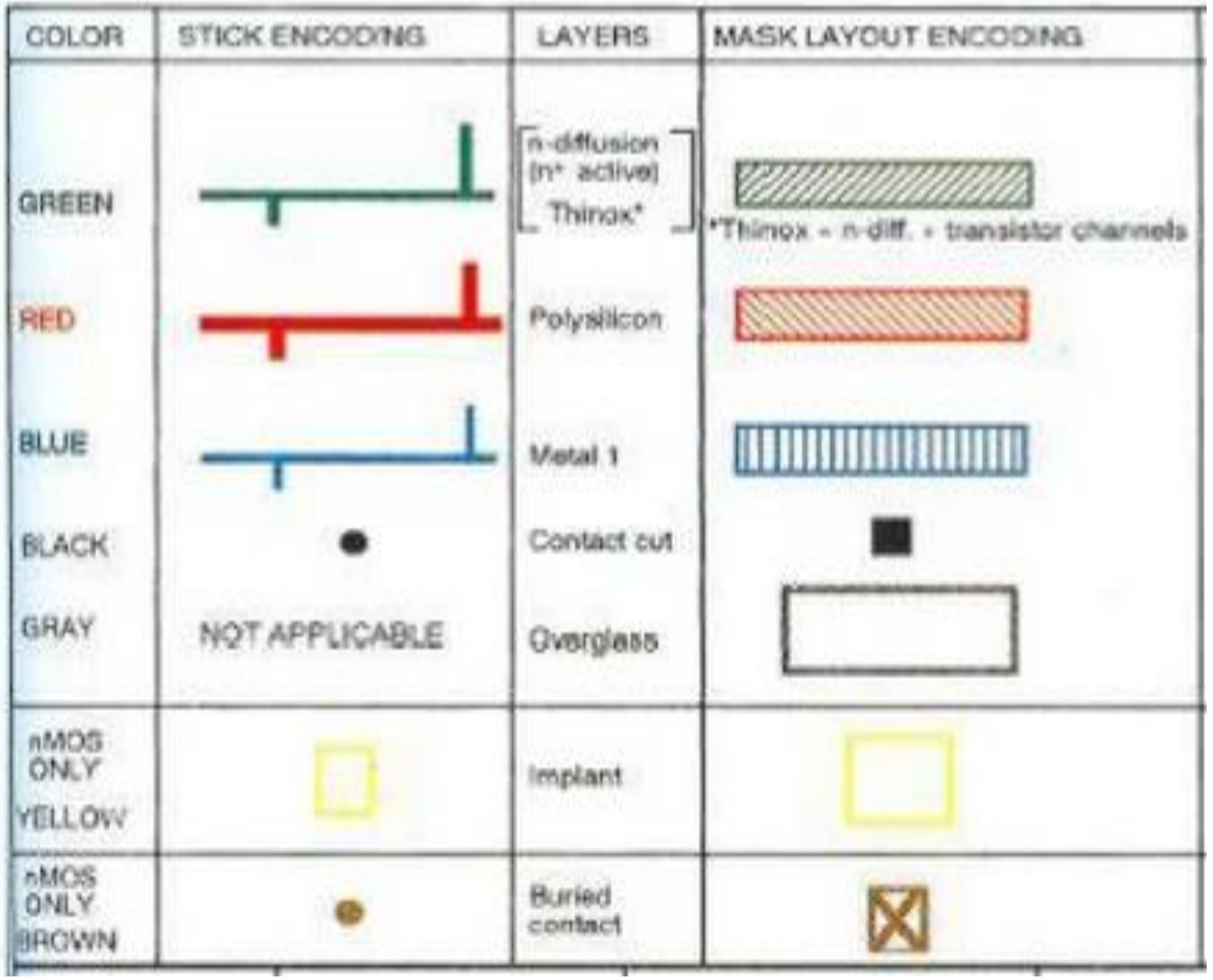
MOS LAYERS

- MOS design aim – **turning a specification into masks** for processing silicon to meet the specification.
- Four basic layers
    - n- diffusion
    - p- diffusion
    - Polysilicon
    - Metal
- **Isolated from one another** by thick or thin (thinox) silicon dioxide insulating layers.
- **Polysilicon and thinox regions interact** so that a **transistor is formed** where they cross one another.
- The **thin oxide (thinox) mask region** includes **n-diffusion, p-diffusion, and transistor channels.**
- **Contacts** are **used to join the layers**

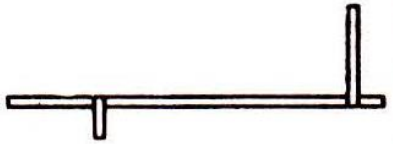
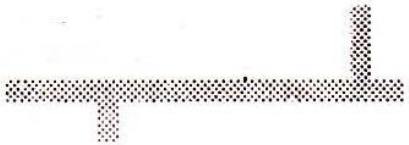
STICK DIAGRAMS

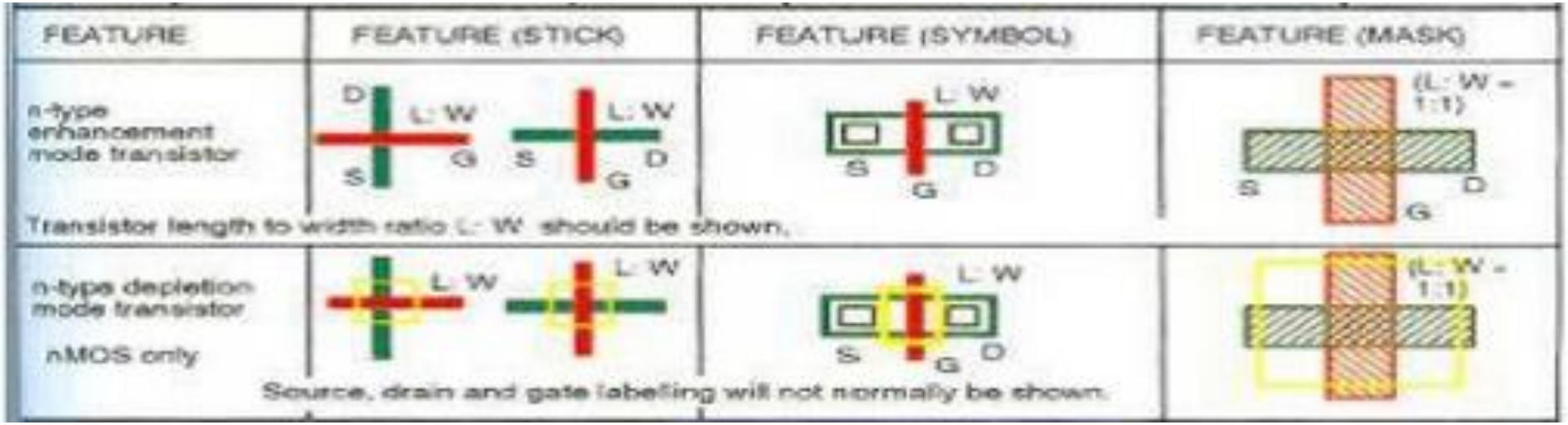
- Stick diagrams are used **to convey layer information through the use of a colour code.**
- Example, for CMOS design,
    - green for n-diffusion
    - Yellow/Brown for p-diffusion
    - red for polysilicon
    - blue for metal
    - black for contact areas
- **Color coding** has been **complemented by monochrome encoding** of the lines.

NMOS technology



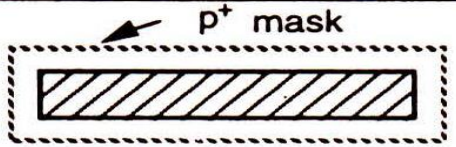
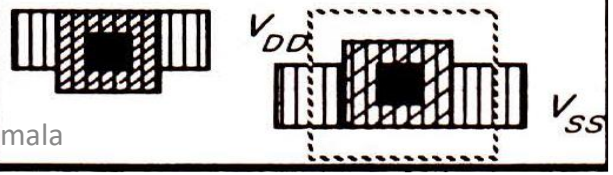
Encodings for a simple metal nMOS process

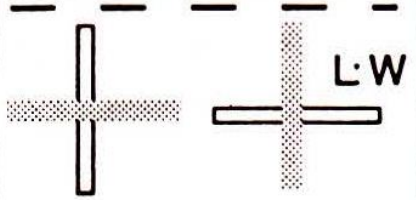
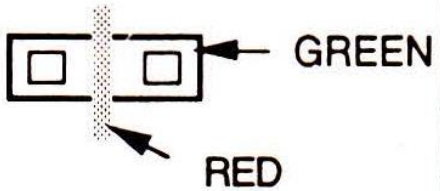
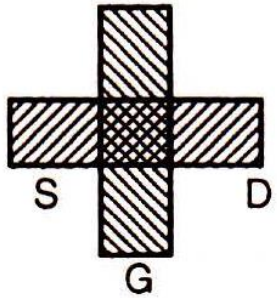
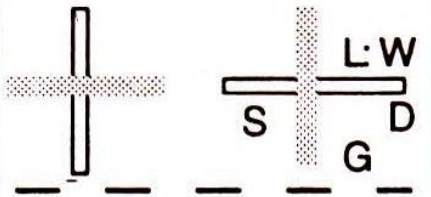
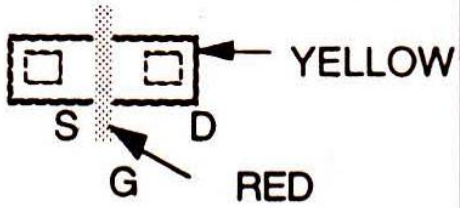
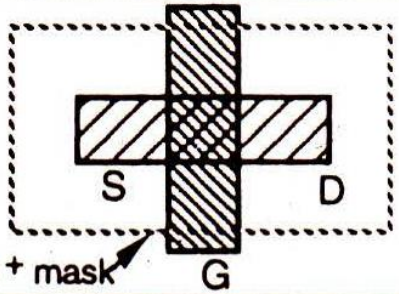
COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING
	MONOCHROME		MONOCHROME
GREEN		n-diffusion (n^+ active) Thinox *	 * Thinox = n-diff. + transistor channels
RED		Polysilicon	
BLUE		Metal 1	
BLACK		Contact cut	
GRAY	NOT APPLICABLE	Overglass	
nMOS ONLY *YELLOW*		Implant	
nMOS ONLY *BROWN*		Buried contact	



FEATURE	FEATURE (STICK) (MONOCHROME)	FEATURE (SYMBOL) (MONOCHROME)	FEATURE (MASK) (MONOCHROME)
n-type enhancement mode transistor	D L:W S G; L:W S D G	GREEN outline (COLOR); S D G; RED line (COLOR)	(L:W = 1:1) S D G
Transistor length to width ratio L:W should be shown but source, drain and gate labeling will not normally be shown.			
n-type depletion mode transistor **nMOS ONLY**	L:W; L:W	GREEN outline (COLOR); S D G; YELLOW (COLOR); RED line (COLOR)	(L:W = 1:1)

Encodings for CMOS process

COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING
GREEN RED BLUE BLACK GRAY	MONOCHROME ENCODING AS IN FIGURE 3–1(a)	n-diffusion (n+ active) Thinox * Polysilicon Metal 1 Contact cut Overglass	MONOCHROME * Thinox = n-diff. + p-diff.+ transistor channels ENCODING AS IN FIGURE 3–1(a)
GREEN IN P+ (MASK) YELLOW (STICK)		p-diffusion (p+ active)	 p+ mask
YELLOW	NOT SHOWN IN STICK DIAGRAM	p+ mask	
DARK BLUE OR PURPLE		Metal 2	
BLACK		VIA	
BROWN	DEMARCATION LINE p-well edge is shown as a demarcation line in stick diagrams	p-well	
BLACK		V_{DD} or V_{SS} CONTACT	 V_{DD} V_{SS}

FEATURE	FEATURE (STICK) (MONOCHROME)	FEATURE (SYMBOL) (MONOCHROME)	FEATURE (MASK) (MONOCHROME)
n-type enhancement mode transistor (as in Figure 3–1(a))	DEMARCATION LINE 		
Transistor length to width ratio L:W may be shown.			
p-type enhancement mode transistor	 DEMARCATION LINE		
Note: p-type transistors are placed above and n-type transistors below the demarcation line			

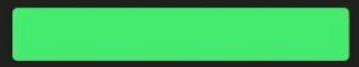
Steps for CMOS Process

- Draw the metal (Blue) Vdd and Vss rails in parallel and create a demarcation line between them. Allowing enough space between them for other circuit elements.
- N-Devices are placed below the demarcation line and P-device are placed above the demarcation line.
- N & P devices are then connected using metal, Polysilicon and contacts.
- Finally the remaining interconnections are made & control signals and data inputs are added.

Introduction

- Stick diagrams convey layer information in a chip.
- Interface between the circuit and the layout.
- Size of transistors, width of layers, wire length etc... are not mentioned in a stick diagram.
- For a chip designer, all CMOS designs consist of the following layers:
  - Substrate or wells of p-type for nMOS transistors and n-type for pMOS transistors
  - Diffusion layers, generally called as Active areas
  - Polysilicon layers, forms the Gate terminals
  - Metal and interconnect layers
  - Contact and via layers

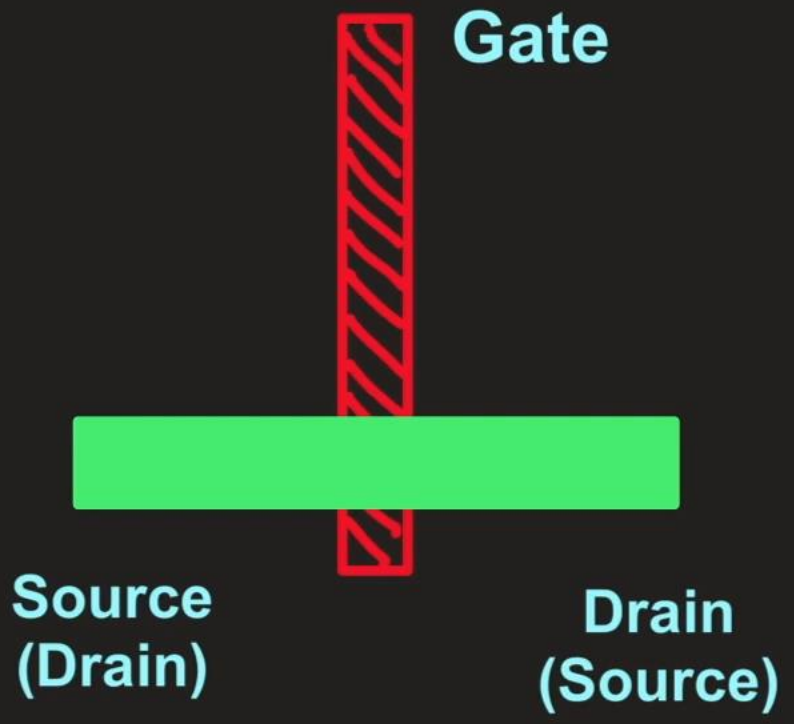
Colour Codes & Patterns

Layers	Colour	Patterns
N diffusion	Green	
P diffusion	Yellow/Brown	
Polysilicon	Red	
Metal 1	Blue	
Metal 2	Magenta	
Contact & Taps	Black	

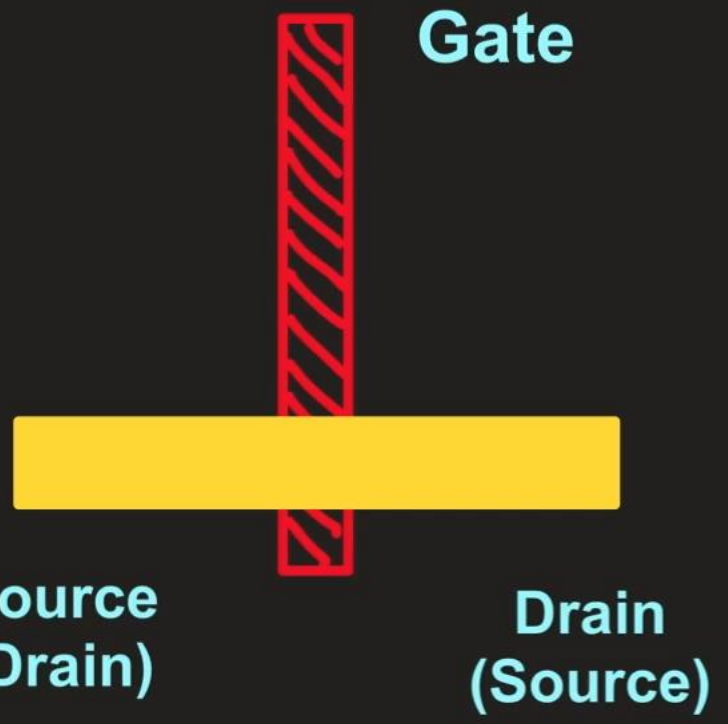
V_{DD} and V_{SS} - metal layers - Blue

Transistors using Colour Code

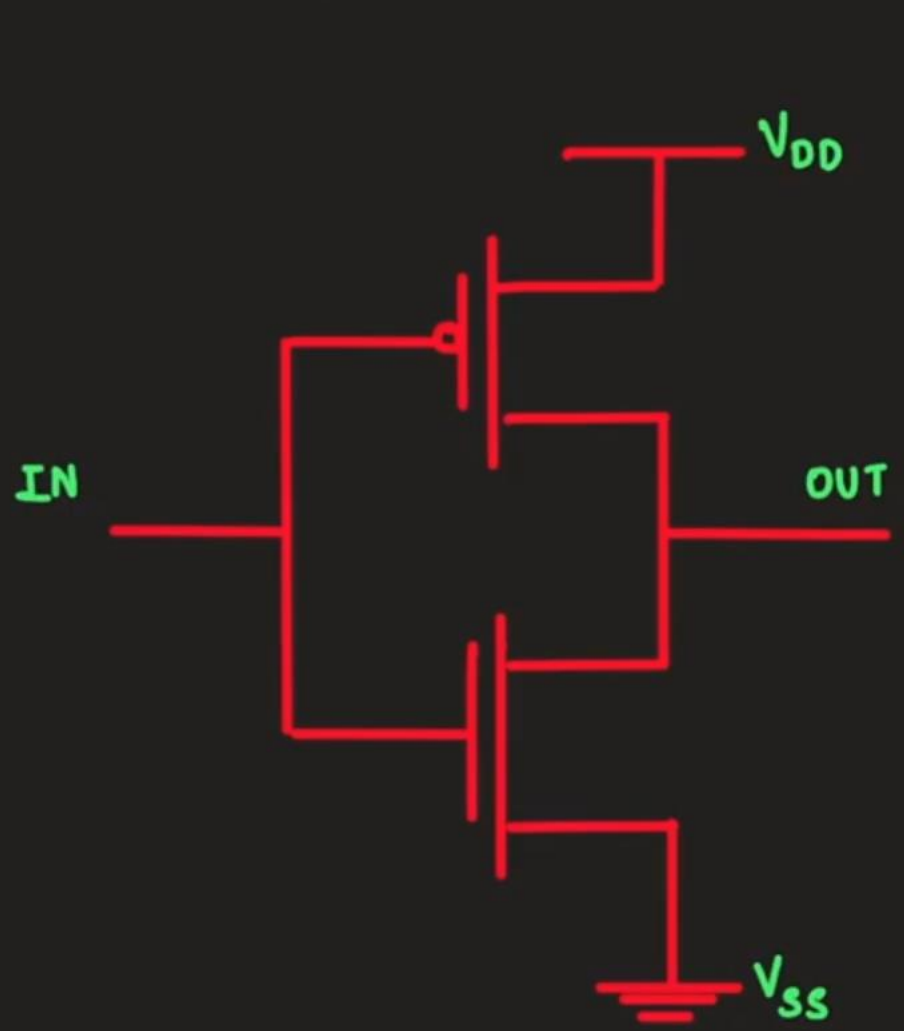
NMOS Transistor

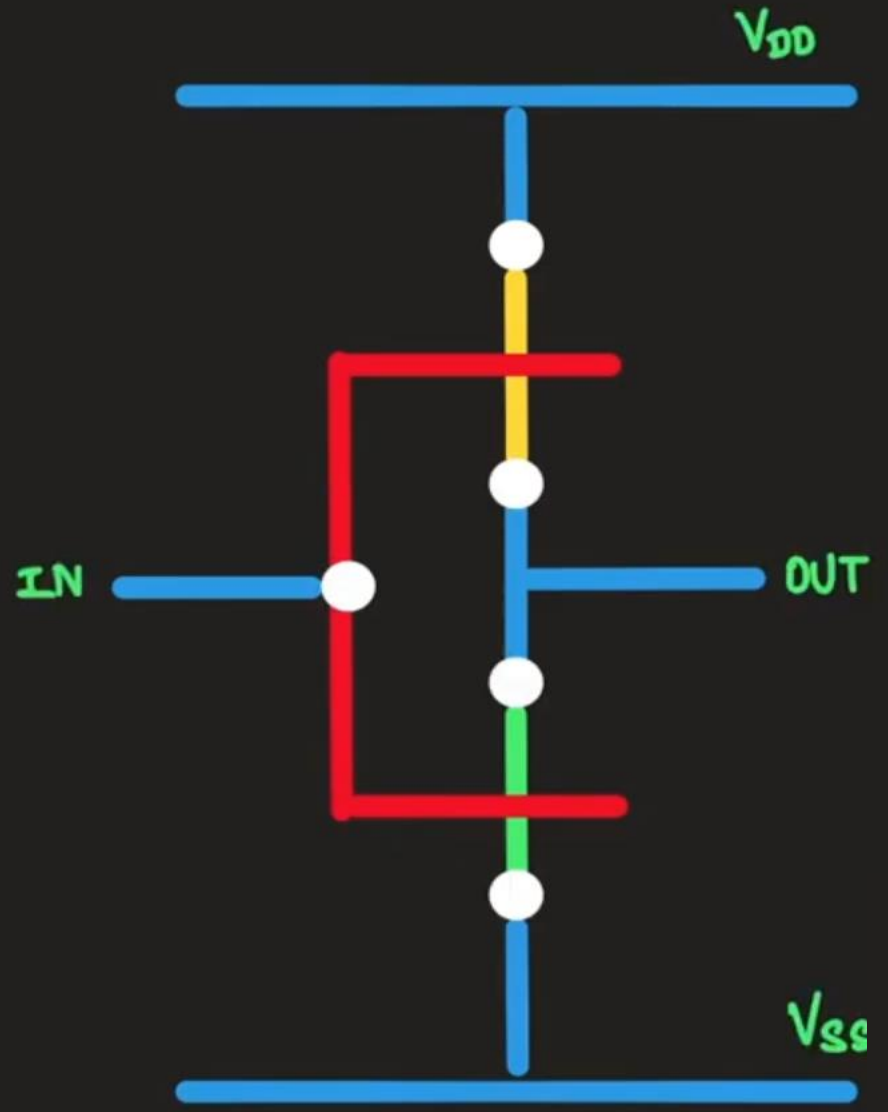


PMOS Transistor



Example - CMOS Inverter





Important Note

- Diffusion paths must not cross the demarcation line.
- N & P diffusion wires must not join.
- The metal should be used to connect n & p features.
- We must place crosses (X) on Vdd and Vss rails to represent substrate and P-well connections respectively.
- Only metal & polysilicon can cross the demarcation line.
- Represent the Vss and Vdd contact croses
    - One on Vdd line for every 4 P-transistors.
    - One on Vss line for every 4 N-transistors.
- Metal lines on different layers (M1 and M2) can cross one another. Contacting 2 metal lines requires a Via.

Steps for NMOS Process

- Draw the metal Vdd and Vss rails in parallel and create a demarcation line between them. Allowing enough space between them for other circuit elements.
- Draw the diffusion paths between the rails.
- Pull up device (depletion type) is to be connected from the output point (Source) to Vdd (Drain)
- Pull down device (enchancement type) is to be connected from the output point (Drain) to Ground (Source)

DESIGN RULES AND LAYOUT

- The **object** of a set of design rules is **to allow** a ready **translation of circuit design concepts**, usually **in stick diagram** or symbolic form, **into actual geometry in silicon**.
- The design rules are the **effective interface between the circuit/system designer and the fabrication engineer**.
- 2-ways
  - μ metal based design rules (orbit)
  - Lambda-based Design Rules
    - **straightforward**
    - **relatively simple to apply**

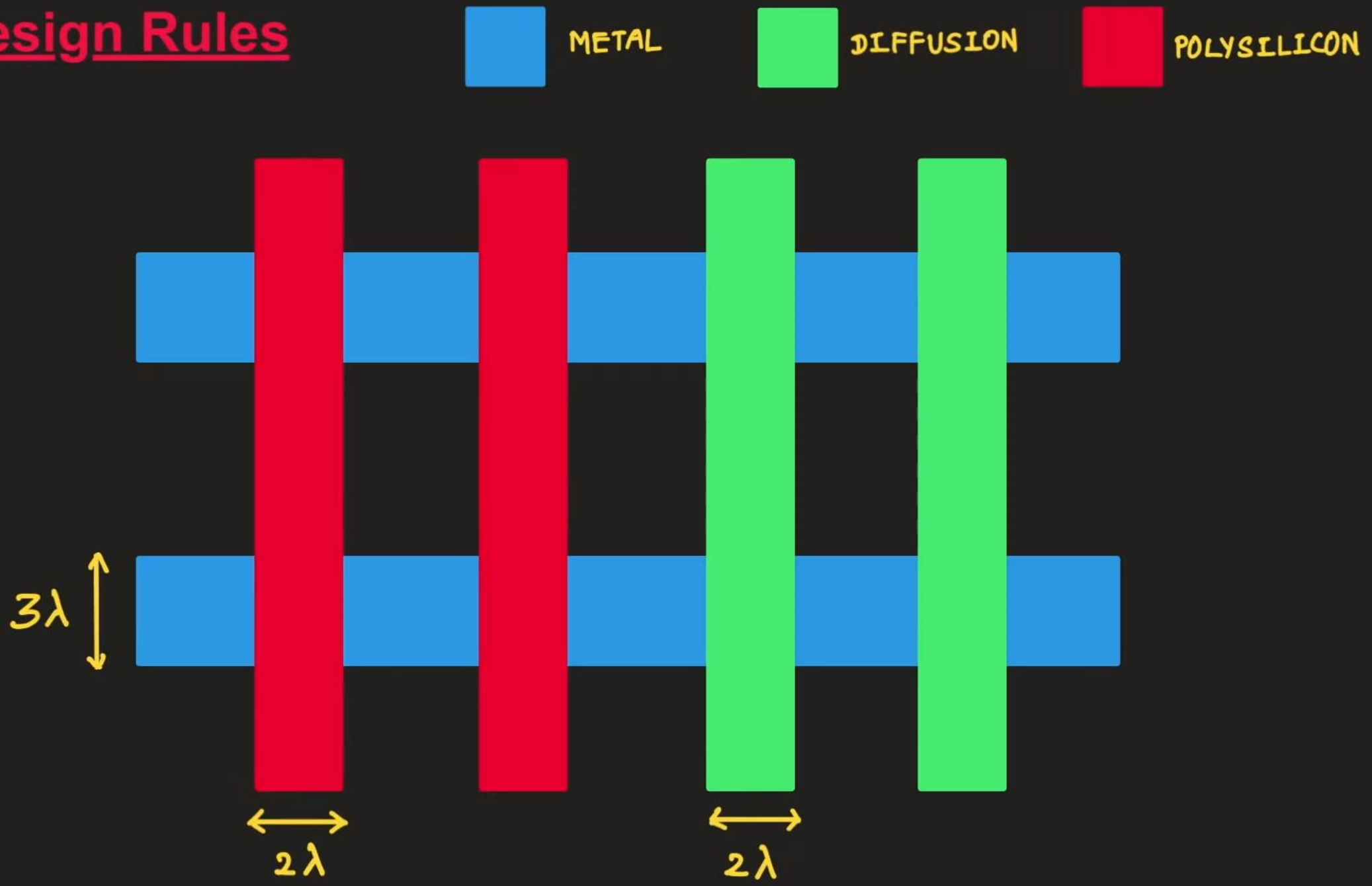
- Design rules are the communication link between the designer specific requirements and the fabricator (metalizing the design)
- Design rules are workable mask layers from which the various layer in silicon.

Layout Design Rules

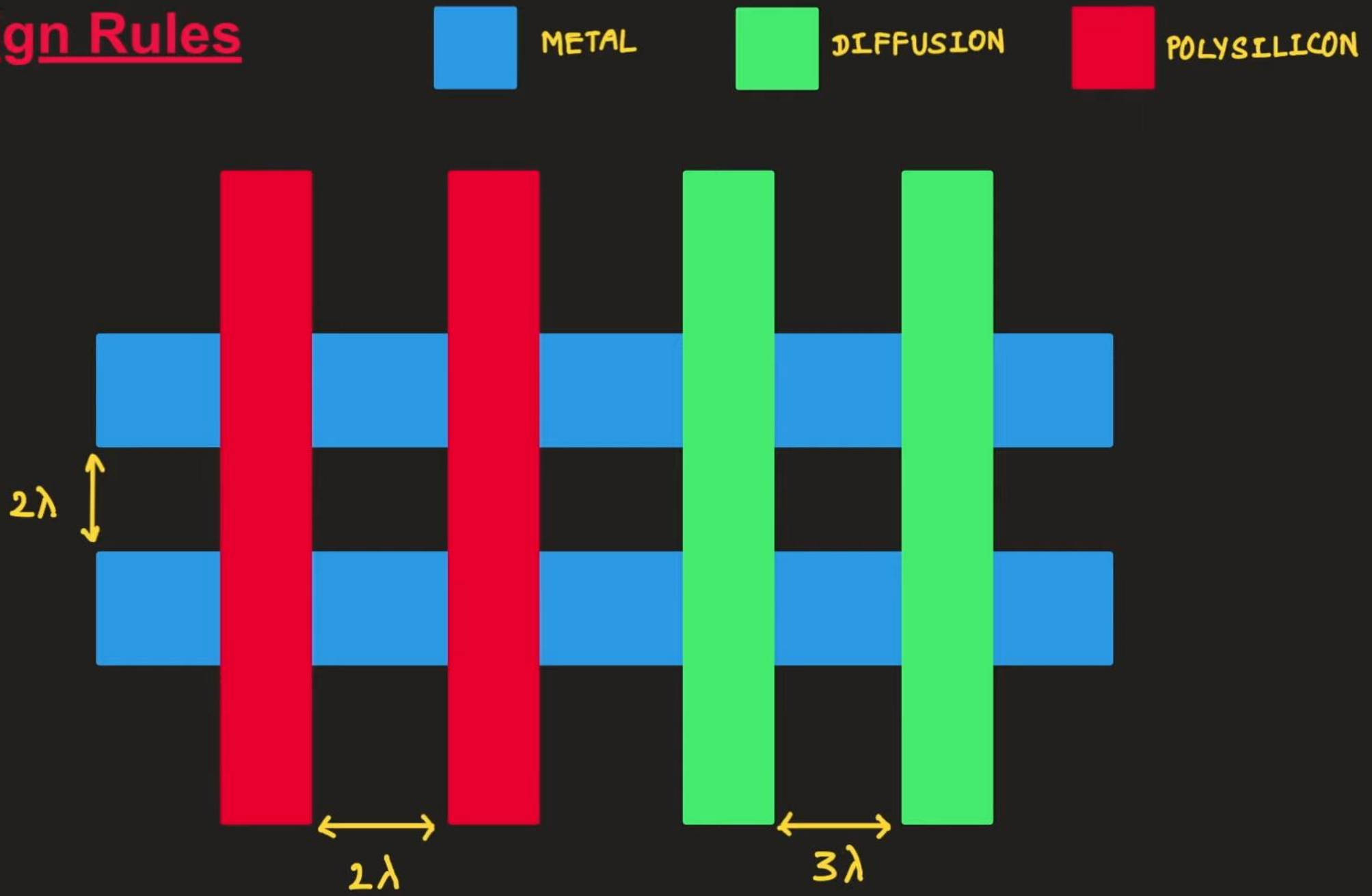
- **Width** - the minimum width of a rectangle
- **Spacing** - the minimum spacing between two rectangles on the same or different layers
- **Overlap** - specifies how much a rectangle must surround another on another layer

$$\lambda = \frac{\text{channel length (L)}}{2}$$

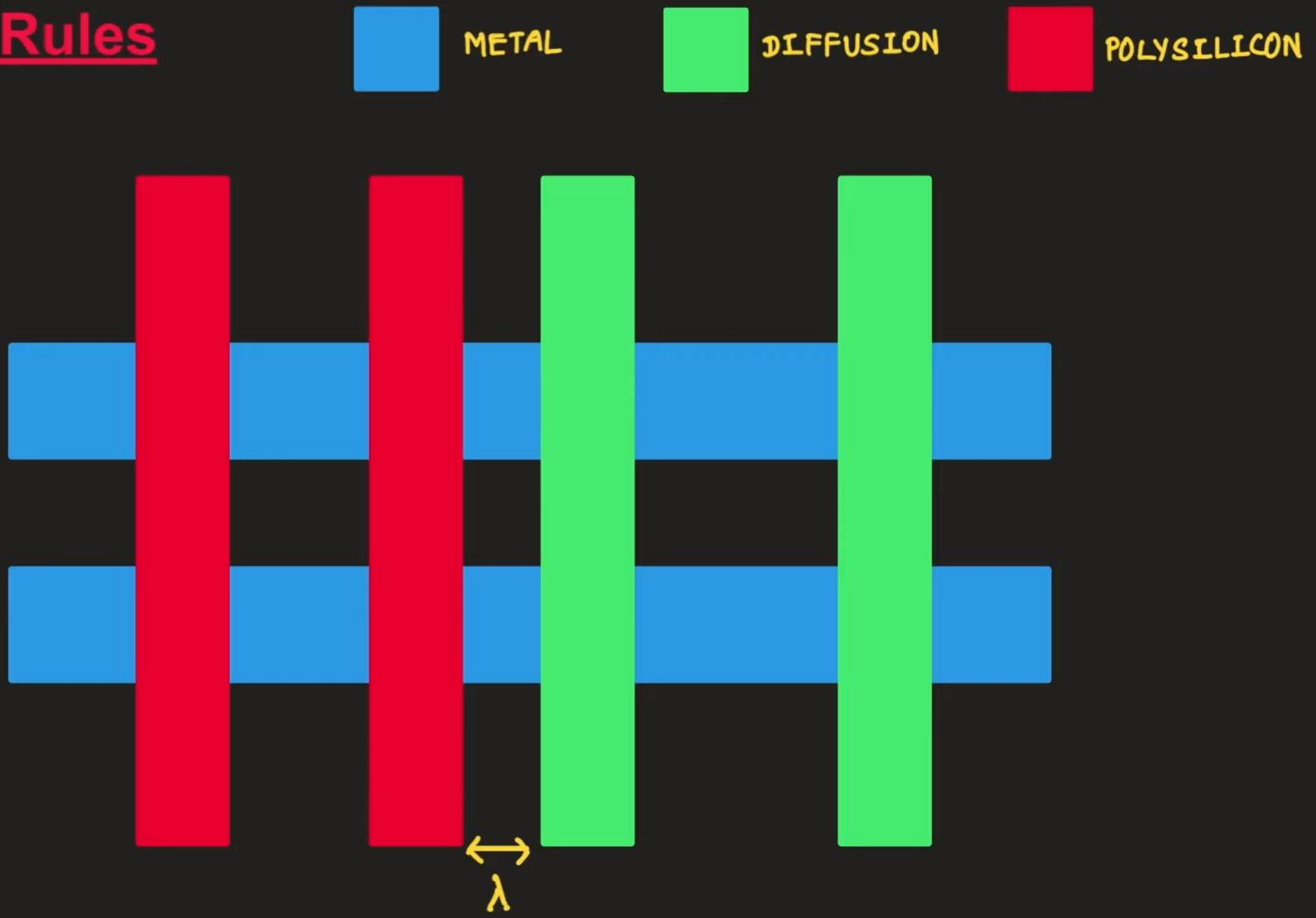
Design Rules



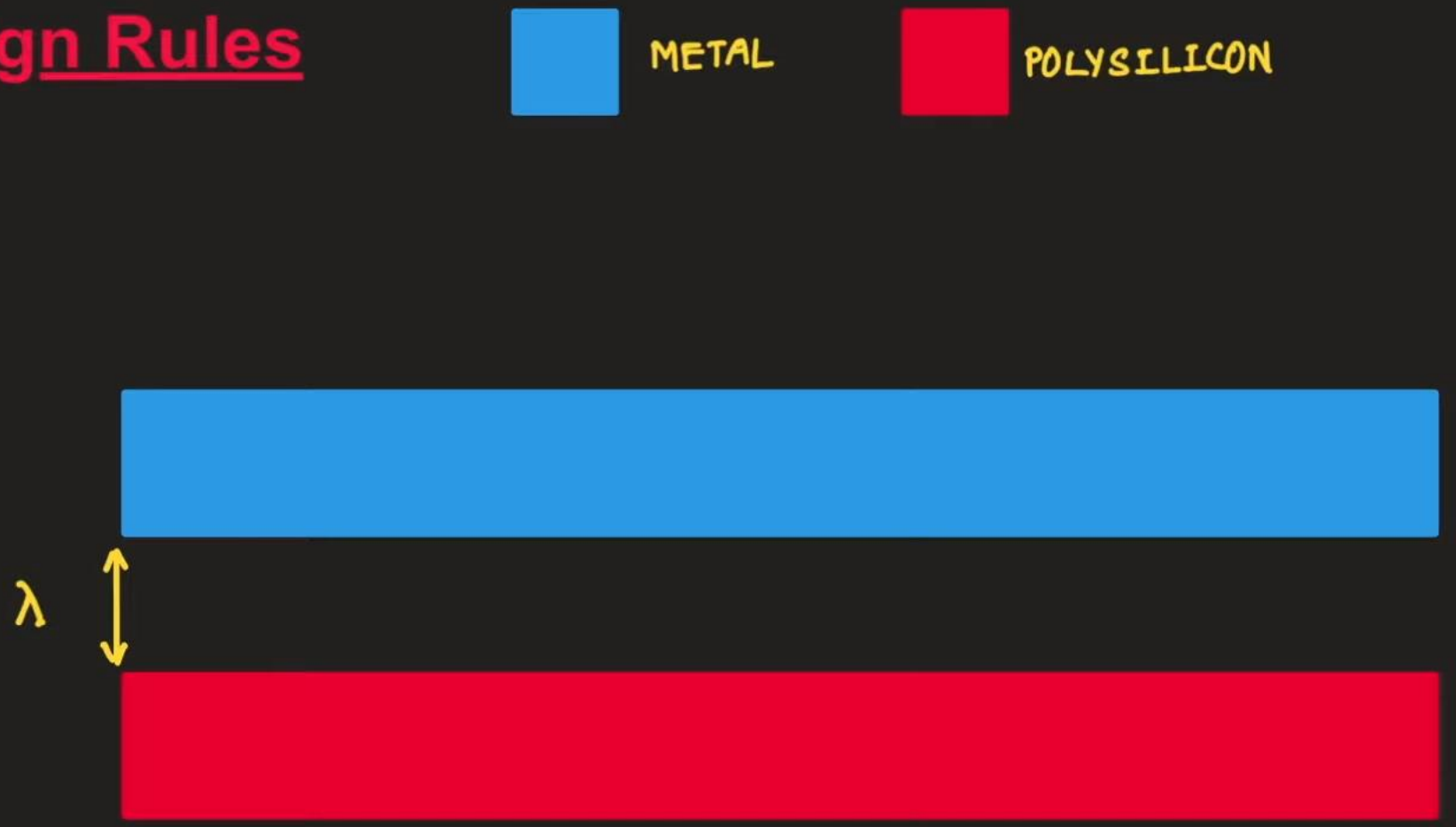
Design Rules



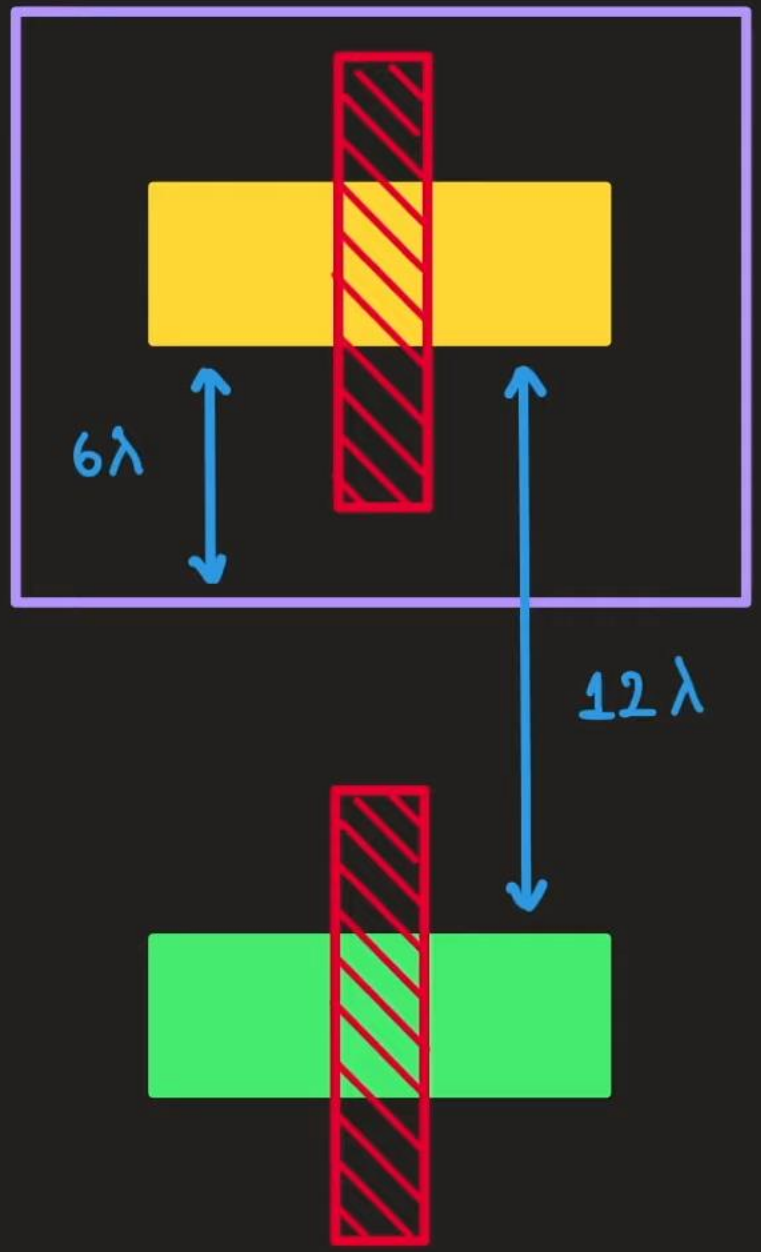
Design Rules

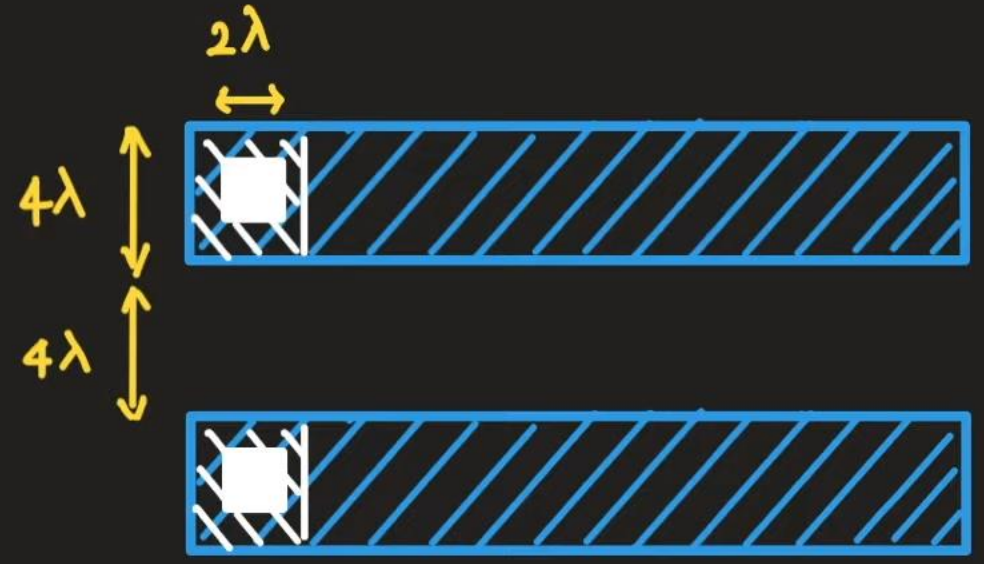


Design Rules

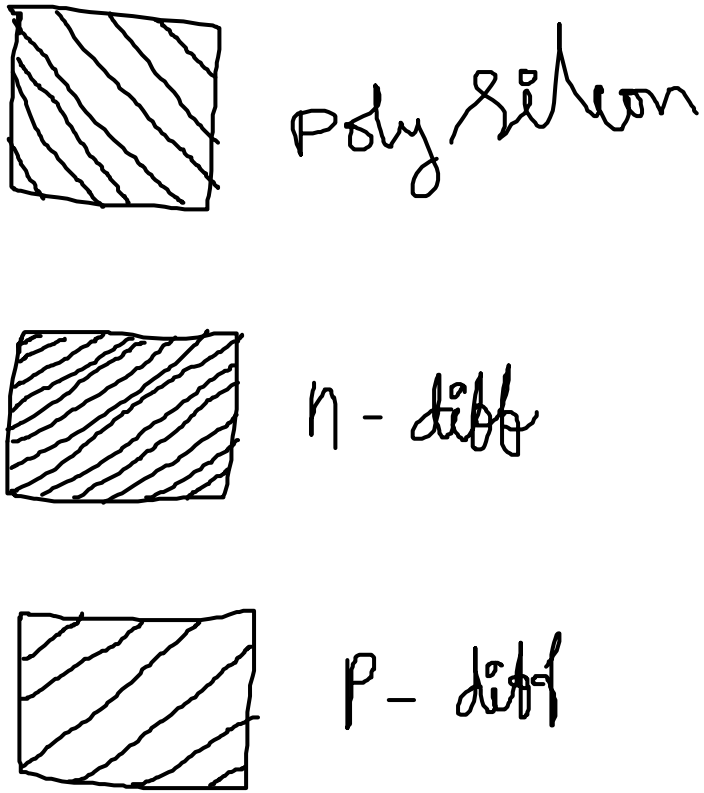


Design Rules

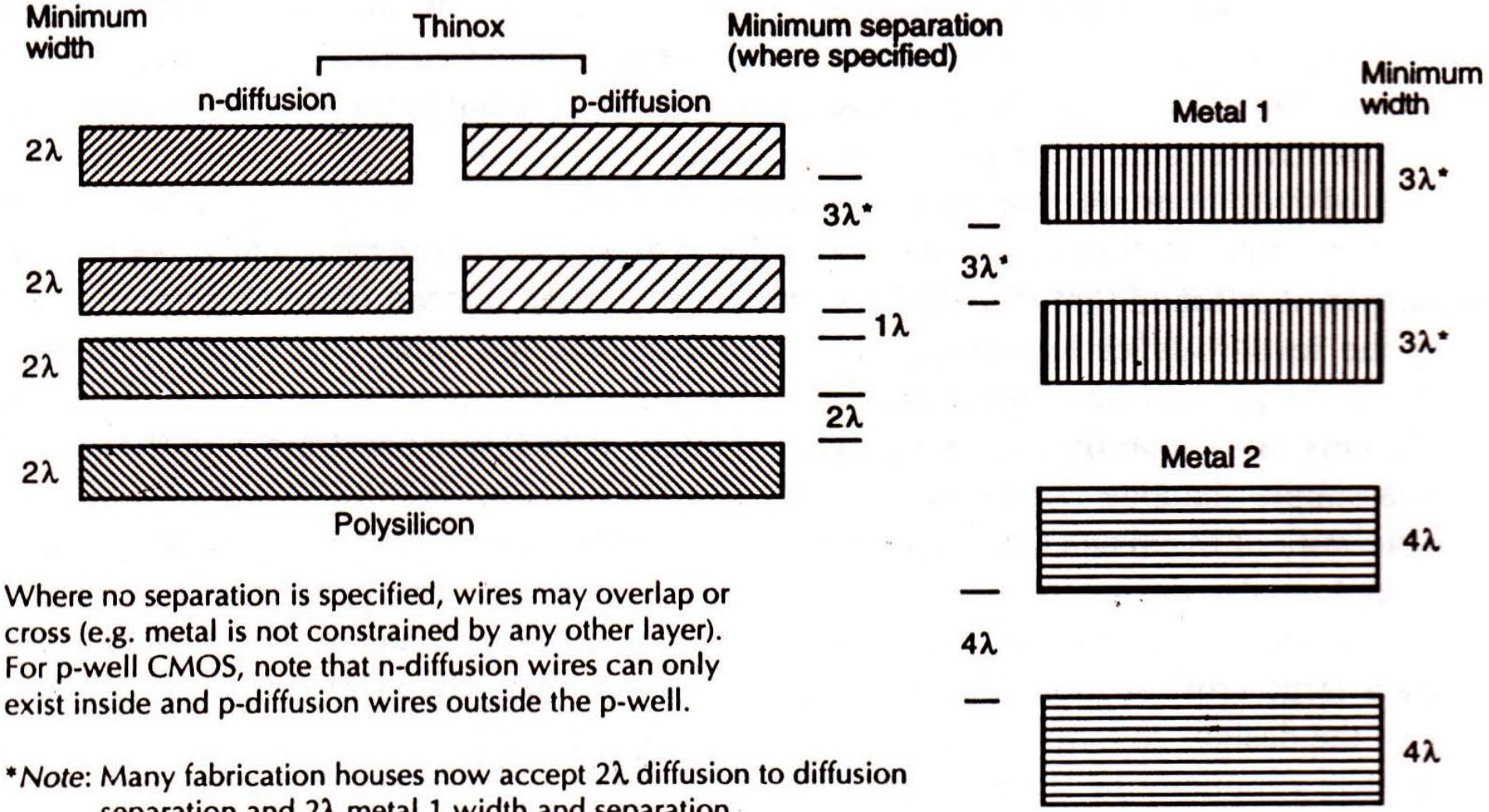




Different layers for Layout



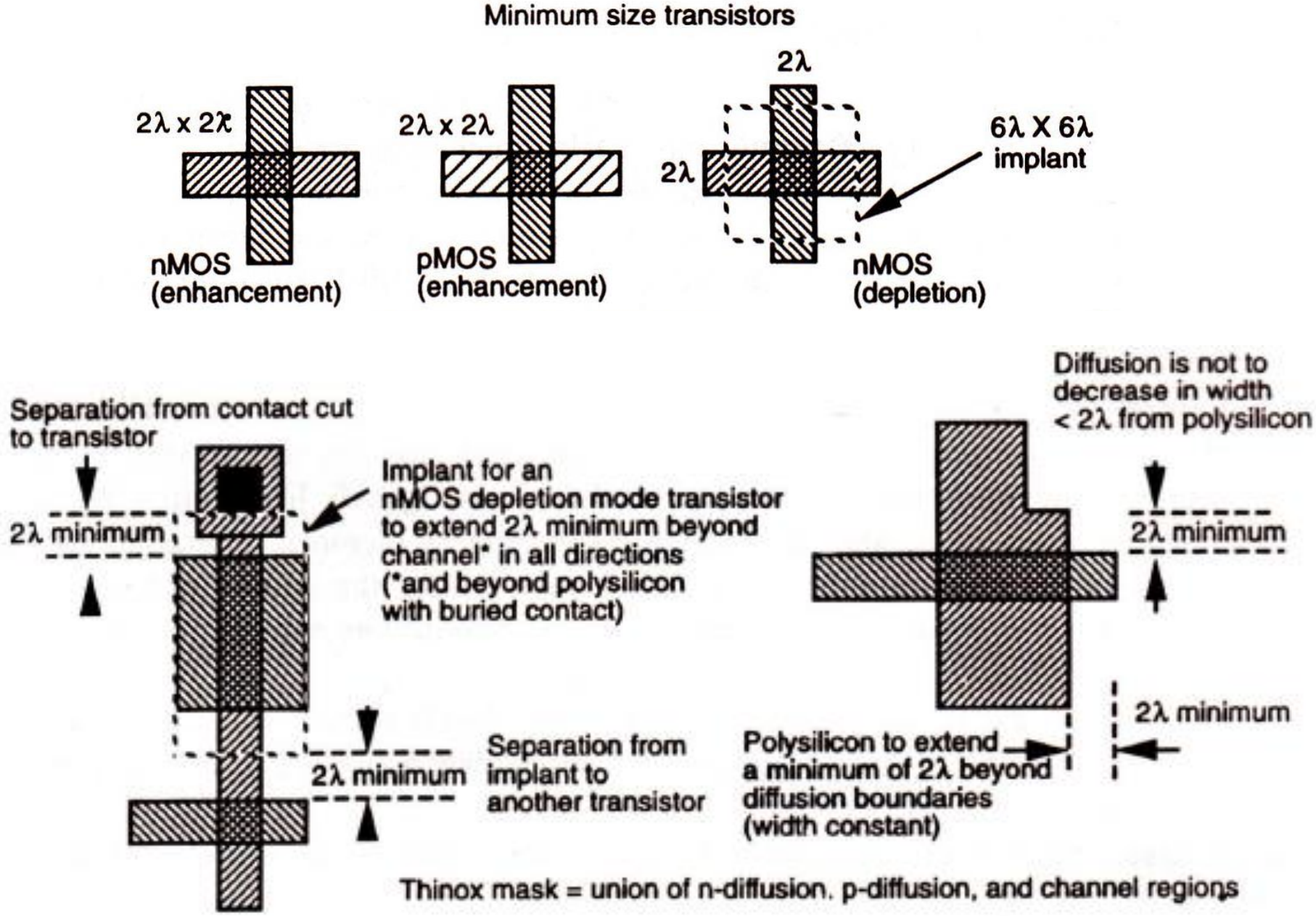
Lambda-based Design Rules



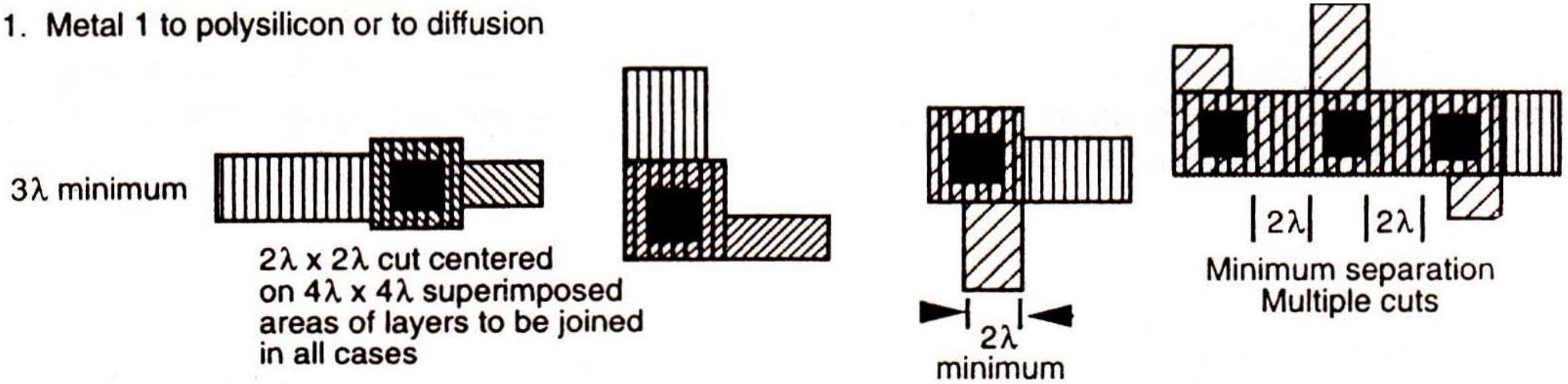
Where no separation is specified, wires may overlap or cross (e.g. metal is not constrained by any other layer). For p-well CMOS, note that n-diffusion wires can only exist inside and p-diffusion wires outside the p-well.

**Note*: Many fabrication houses now accept 2λ diffusion to diffusion separation and 2λ metal 1 width and separation.

Figure 3–6 Design rules for wires (nMOS and CMOS)


Minimum size transistors
2λ x 2λ
nMOS
(enhancement)
2λ x 2λ
pMOS
(enhancement)
2λ
2λ
6λ X 6λ
implant
nMOS
(depletion)
Separation from contact cut
to transistor
2λ minimum
Implant for an
nMOS depletion mode transistor
to extend 2λ minimum beyond
channel* in all directions
(*and beyond polysilicon
with buried contact)
2λ minimum
Separation from
implant to
another transistor
Diffusion is not to
decrease in width
< 2λ from polysilicon
2λ minimum
2λ minimum
Polysilicon to extend
a minimum of 2λ beyond
diffusion boundaries
(width constant)
Thinox mask = union of n-diffusion, p-diffusion, and channel regions

1. Metal 1 to polysilicon or to diffusion



2. Via (contact from metal 2 to metal 1 and thence to other layers)

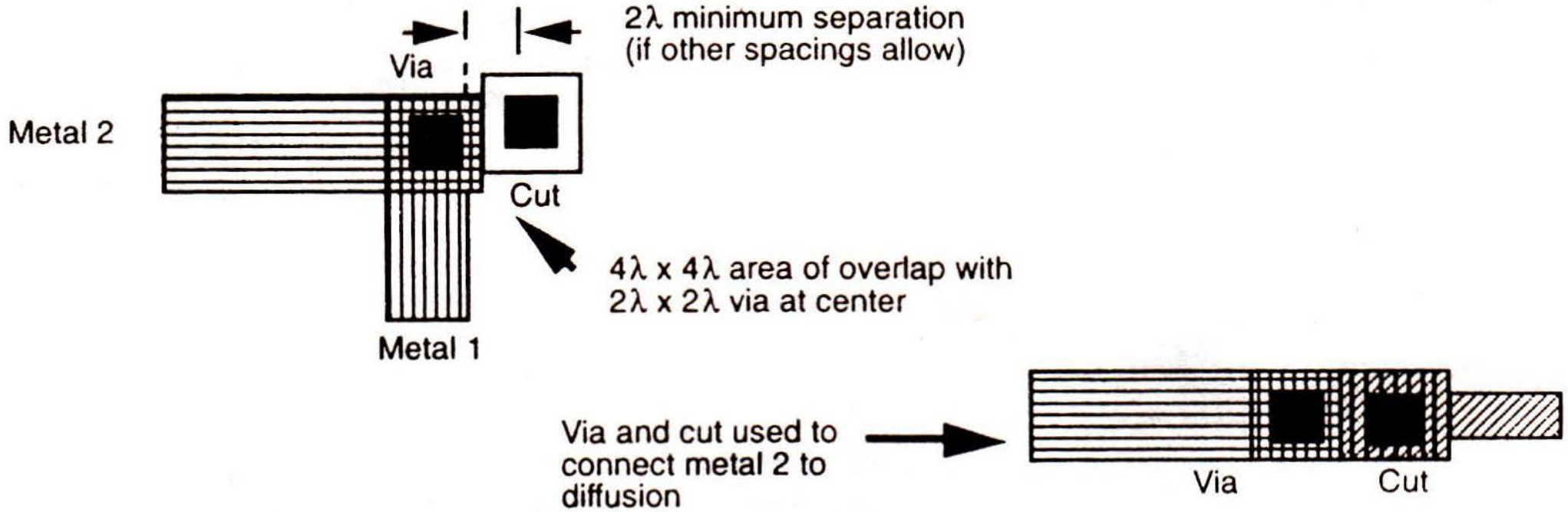


FIGURE 3.8 Contacts (nMOS and CMOS).

CMOS Sub System Design

Introduction

Partitioning the system into subsystems of the types listed below:

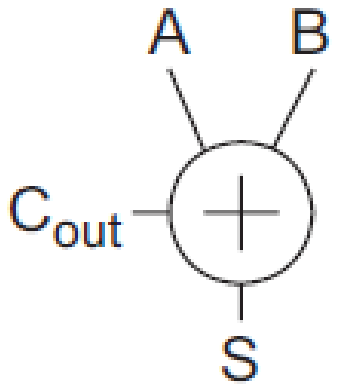
- Datapath operators
- Memory elements
- Control structures
- Special-purpose cells
  - I/O
  - Power distribution
  - Clock generation and distribution
  - Analog and RF

Addition/Subtraction

- Basis for many processing operations- counting, multiplication, filtering etc..
- add two binary numbers
- adder architectures serve different speed/power/area requirements

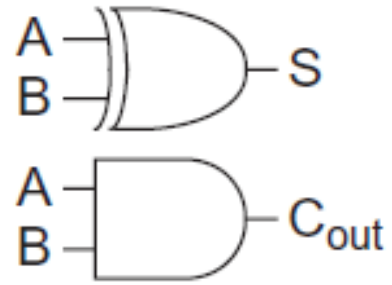
Single bit addition

1) Half adder

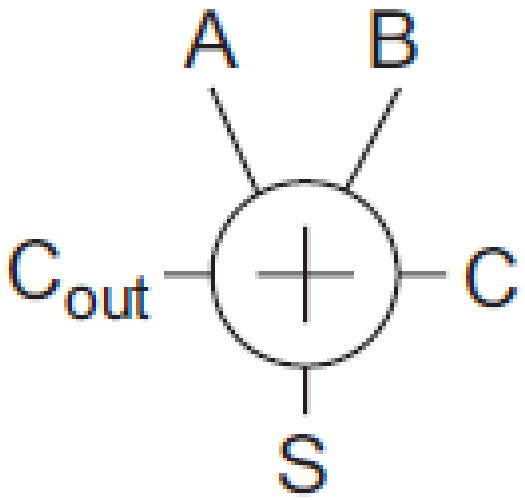


A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$
$$C_{out} = A \cdot B$$



2) Full adder



A	B	C	G	P	K	C_{out}	S
0	0	0	0	0	1	0	0
		1				0	1
0	1	0	0	1	0	0	1
		1				1	0
1	0	0	0	1	0	0	1
		1				1	0
1	1	0	1	0	0	1	0
		1				1	1

Generate (G): The adder generates a carry when Cout is true independent of Cin

$$G = A \cdot B.$$

Propagate (P): The adder propagates a carry; i.e., it produces a carry-out if and only if it receives a carry-in

$$P = A \oplus B.$$

Kill (K): The adder kills a carry when Cout is false independent of Cin

$$K = \bar{A}.\bar{B} = \overline{A + B}$$

A	B	C_i	S	C_o	*Carry status*
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate

Full adder design

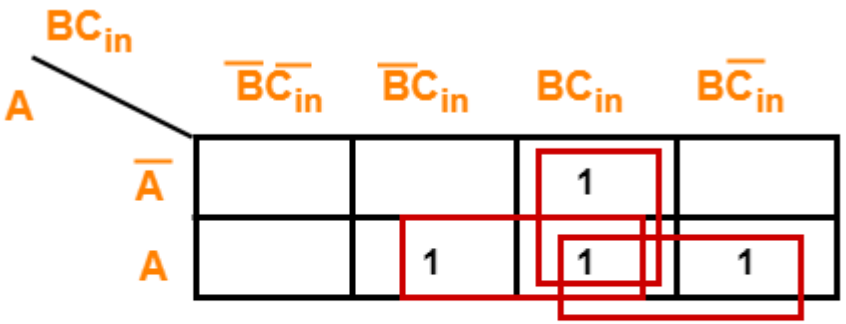
For S:

$$S = \overline{A\overline{B}C_{in} + \overline{A}\overline{B}C_{in} + ABC_{in} + \overline{A}BC_{in}}$$

$$S = A \oplus B \oplus C_{in}$$

$$S = \overline{A\overline{B}C_{in} \cdot \overline{A}\overline{B}C_{in} \cdot ABC_{in} \cdot \overline{A}B\overline{C_{in}}}$$

$$= \overline{(\overline{A}+B+C_{in}) \cdot (A+B+\overline{C_{in}}) \cdot (\overline{A}+\overline{B}+\overline{C_{in}}) \cdot (A+\overline{B}+C_{in})}$$



$$C_{out} = \overline{\overline{AB} + \overline{BC_{in}} + \overline{C_{in}A}}$$

$$C_{out} = \overline{\overline{AB} \cdot \overline{BC_{in}} \cdot \overline{C_{in}A}}$$

$$= \overline{(\bar{A} + \bar{B}) \cdot (\bar{B} + \bar{C_{in}}) \cdot (\bar{C_{in}} + \bar{A})}$$

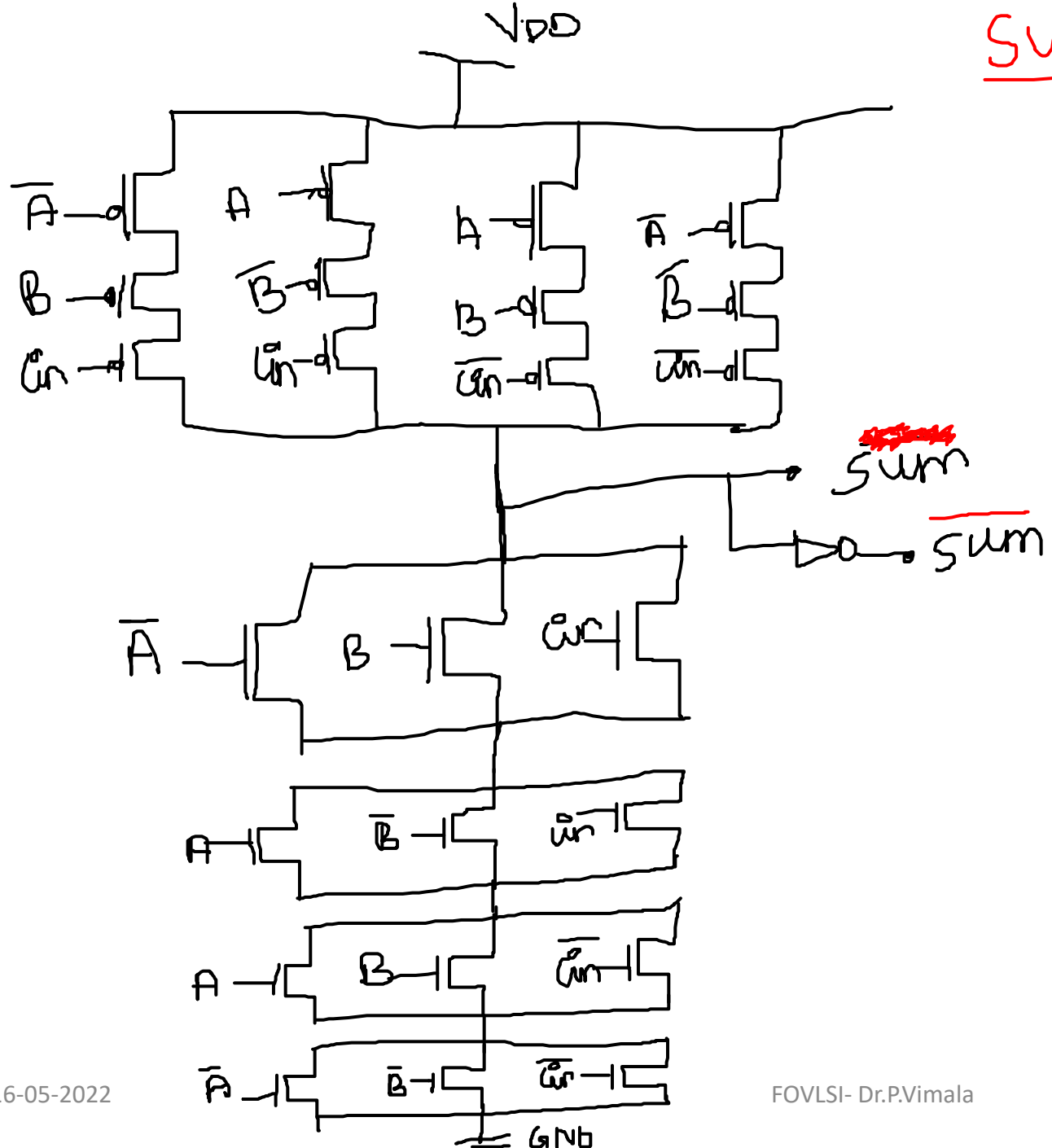
Common procedure used for all type of circuits:

NMOS

OR operation (**+**) **parallel**
AND operation (**.**) **series**

PMOS

OR operation (**+**) **series**
AND operation (**.**) **parallel**

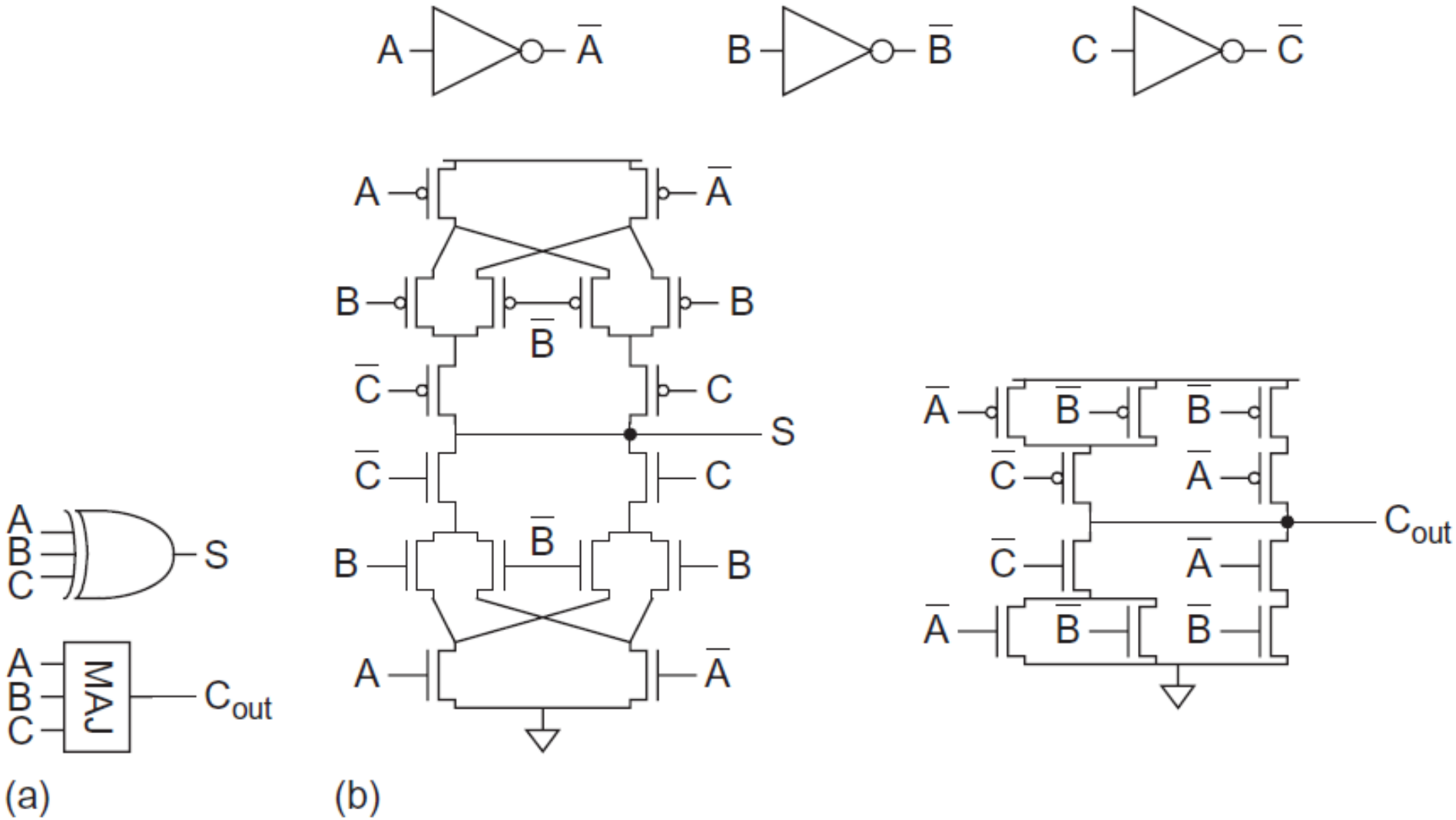


Sum

No of transistors

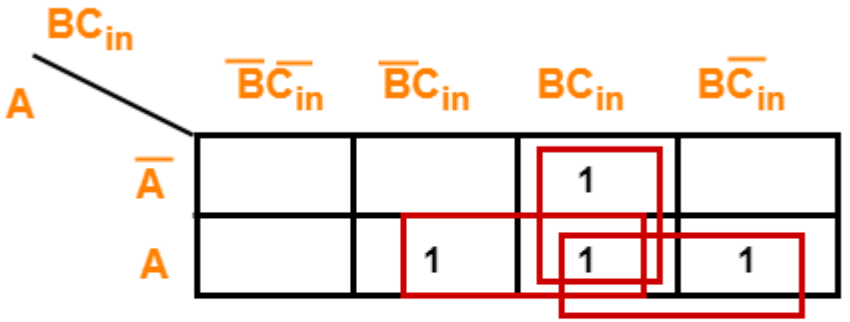
24

26



32 transistors (6 for the inverters, 10 for the majority gate, and 16 for the 3-input XOR).

A	B	C	G	P	K	C_{out}	S
0	0	0	0	0	1	0	0
		1				0	1
0	1	0	0	1	0	0	1
		1				1	0
1	0	0	0	1	0	0	1
		1				1	0
1	1	0	1	0	0	1	0
		1				1	1



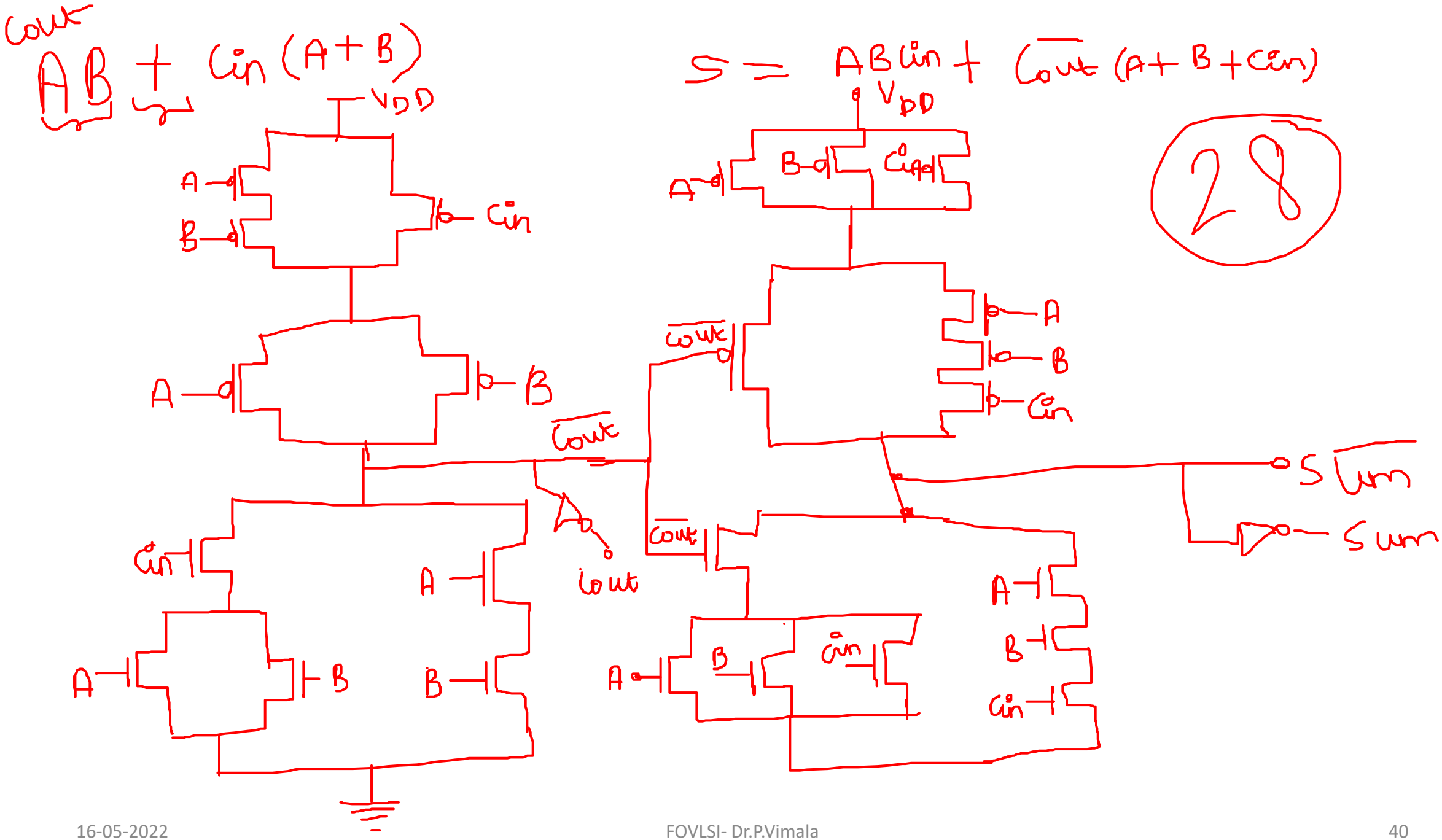
$C_{out} = AB + BC_{in} + C_{in}A$

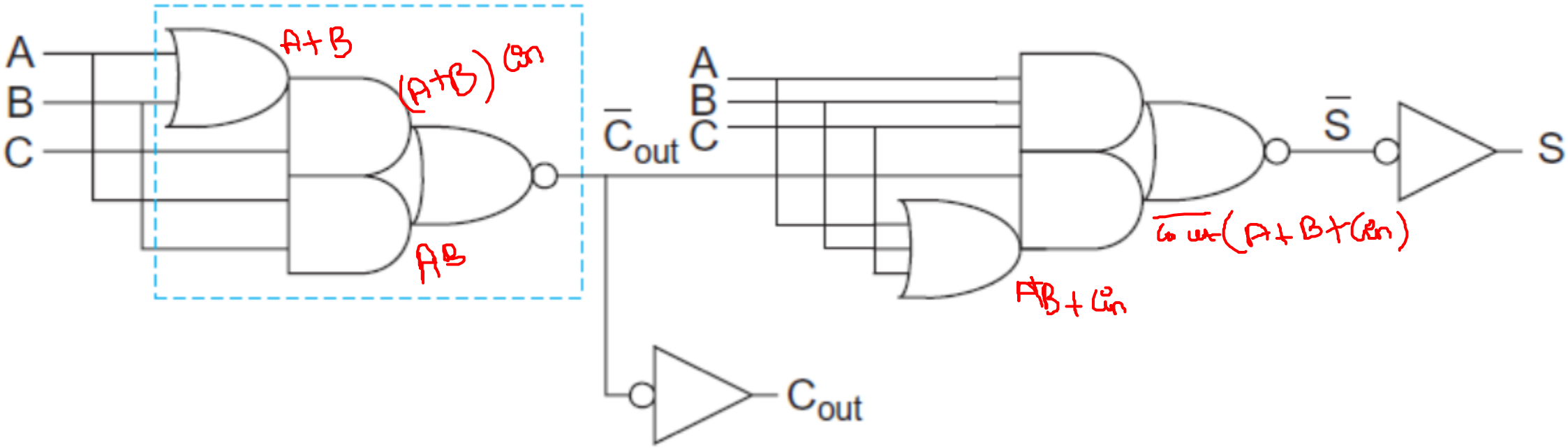
A	B	Cin	$\overline{Cout}$	S
0	0	0	1	0
0	0	1	1	1 ✓
0	1	0	1	1 ✓
0	1	1	0	0
1	0	0	1	1 ✓
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1 ✓

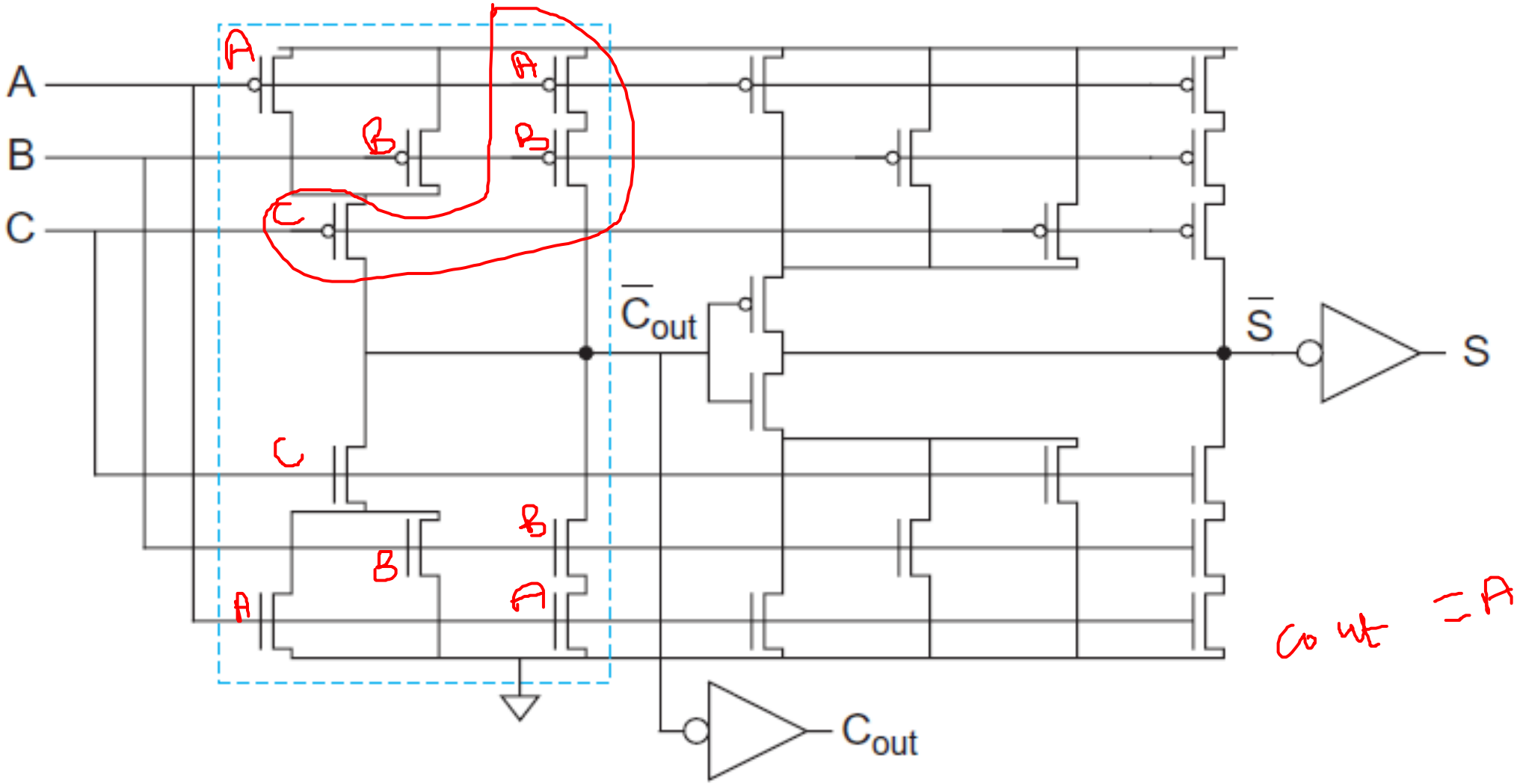
$Cout = AB + Cin(A + B)$ ✓

$S = Cin\overline{Cout} + B\overline{Cout} + A\overline{Cout} + ABCin$

$S = \overline{Cout}[A + B + Cin] + ABCin$


Cout
AB + Cin (A+B)
S = ABCin + Cout (A+B+Cin)
28
VDD
A
B
Cin
Cout
Sum
Cout


A
B
C
A+B
(A+B) Cin
AB
$\overline{C}_{out}$
A
B
C
$\overline{S}$
S
$\overline{C_{out}}$(A+B+Cin)
AB+Cin
C_{out}

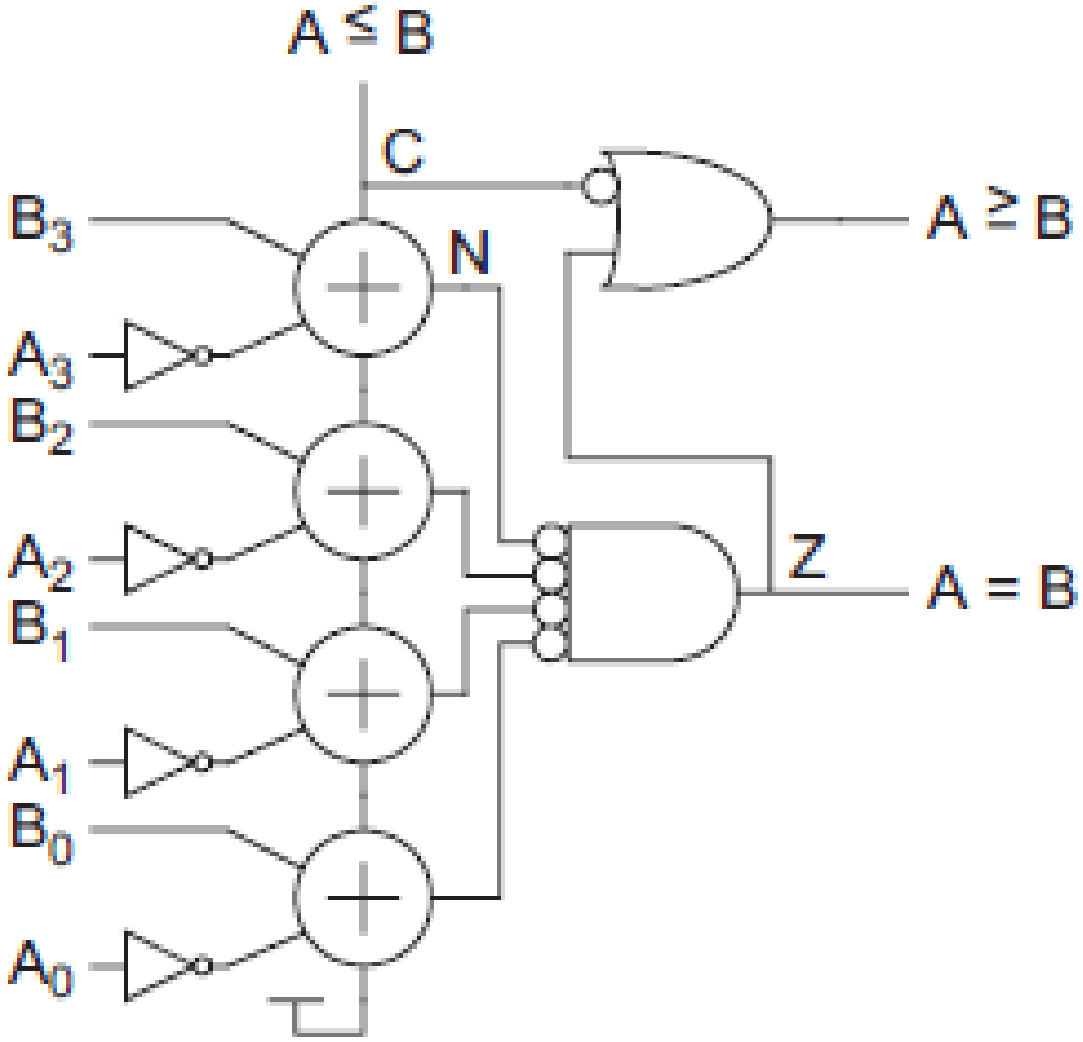


pMOS network is identical to the nMOS network rather than being the conduction complement, so the topology is called a **mirror adder.**

Comparators

1) Magnitude Comparator

- A magnitude comparator determines the larger of two binary numbers.
- To compare two unsigned numbers A and B, **compute B – A = B + A + 1.**
  - **If there is a carry-out, A ≤ B;**
  - **otherwise, A > B.**
  - **A zero detector indicates that the numbers are equal.**



Unsigned magnitude comparator

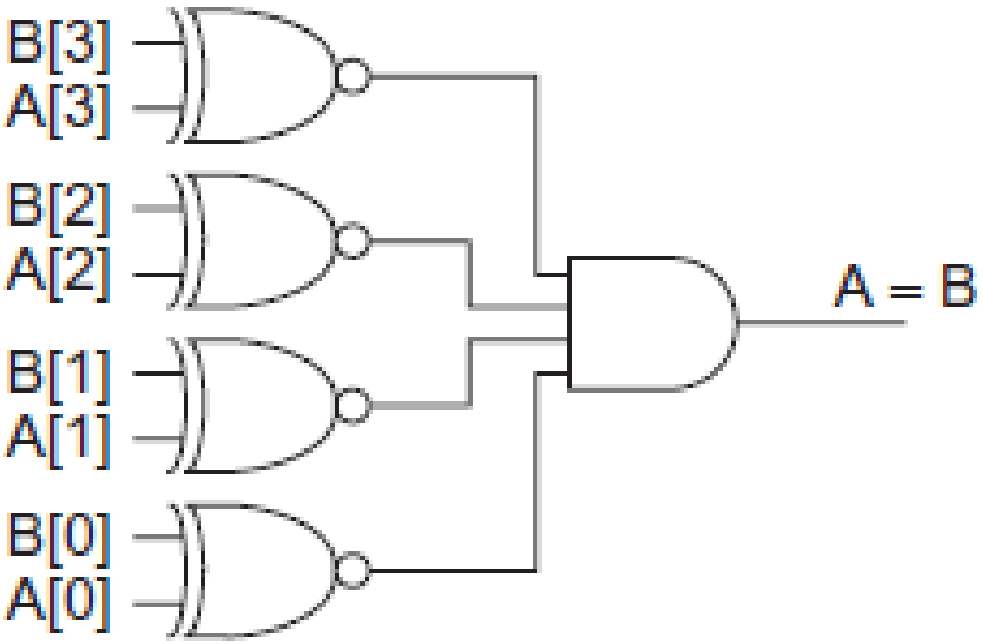
Table 8.4 Magnitude comparison

Relation	Unsigned Comparison	Signed Comparison
$A = B$	Z	Z
$A \neq B$	$\overline{Z}$	$\overline{Z}$
$A < B$	$\overline{C + Z}$	$\overline{(N \oplus V) + Z}$
$A > B$	$\overline{C}$	$(N \oplus V)$
$A \leq B$	C	$(\overline{N \oplus V})$
$A \geq B$	$\overline{C} + Z$	$(N \oplus V) + Z$

- Comparing signed two's complement numbers is complicated - possibility of overflow when subtracting two numbers with different signs.
- Must determine if the result is negative (N, indicated by the most significant bit of the result) and if it overflows the range of possible signed numbers. **overflow signal V.**
- If the inputs had different signs and the output sign is different from the sign of B. Then, **V** is true
- The actual sign of the difference B – A is $S = N \oplus V$ because overflow flips the sign.
- If this corrected sign is negative (S = 1), we know A > B.

2) Equality Comparator

- An equality comparator determines if (A = B).
- This can be done more simply and rapidly with XNOR gates and a ones detector



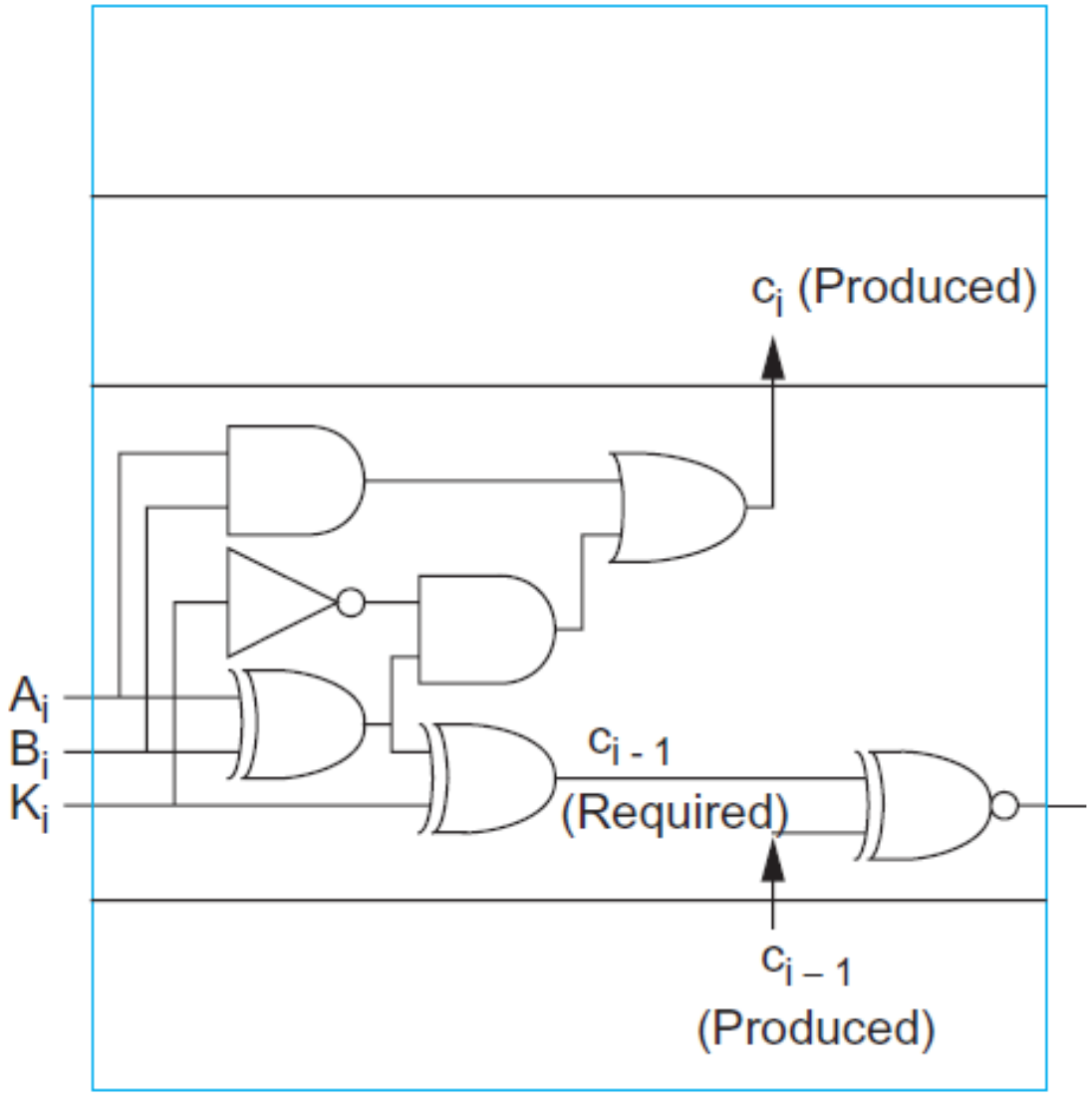
3) K = A + B Comparator

- Used in sum addressed memory
- This comparison can be done faster than computing A + B because **no carry propagation is necessary.**
- The **key** is that if you know A and B, you also know what the carry into each bit must be if K = A + B.
- you only need to check adjacent pairs of bits
  - **verify that the previous bit produces the carry required by the current bit**
  - **use a ones detector to check that the condition is true for all N pairs.**

A_i	B_i	K_i	c_{i-1} (required)	c_i (produced)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

- Required carry, $C_{i-1} = A_i \oplus B_i \oplus K_i$
- Produced carry, $C_i = \bar{A}B\bar{K} + A\bar{B}\bar{K} + A \oplus B \quad A B\bar{K} + AB K$

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



$A + B = K$ comparator

- **XNOR gate** is used to make sure that the **required carry matches the produced carry** at each bit position

LFSR

- Linear Feedback Shift Registers
- consists of N registers configured as a shift register.
- input to the shift register comes from the XOR of particular bits of the register
- On reset, the registers must be initialized to a nonzero value (e.g., all 1s)

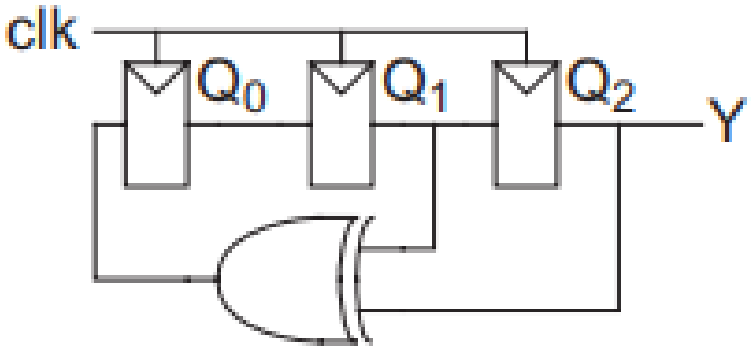


FIGURE 11.54 3-bit LFSR

TABLE 11.7 LFSR sequence

Cycle	Q_0	Q_1	Q_2 / Y
0	1	1	1
1	0	1	1
2	0	0	1
3	1	0	0
4	0	1	0
5	1	0	1
6	1	1	0
7	1	1	1
Repeats forever			

- LFSR is an example of a ***maximal-length shift register*** because its output sequences through all $2^n - 1$ combinations.
- The inputs fed to the XOR are called the **tap sequence** and are often specified with a **characteristic polynomial**.
  - For example, this 3-bit LFSR has the characteristic polynomial $1 + x^2 + x^3$ because the taps come after the second and third registers.
  - characteristic polynomial **defined by XOR positions**

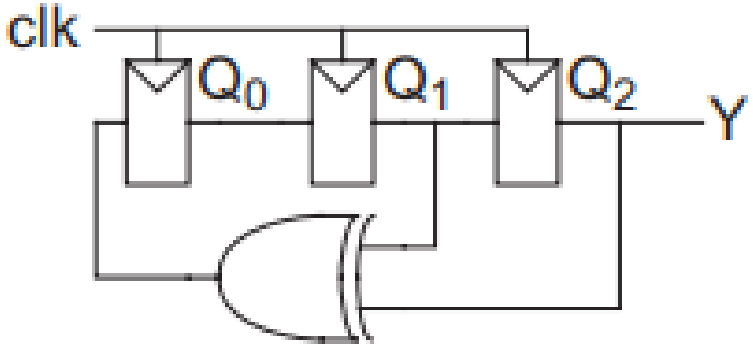
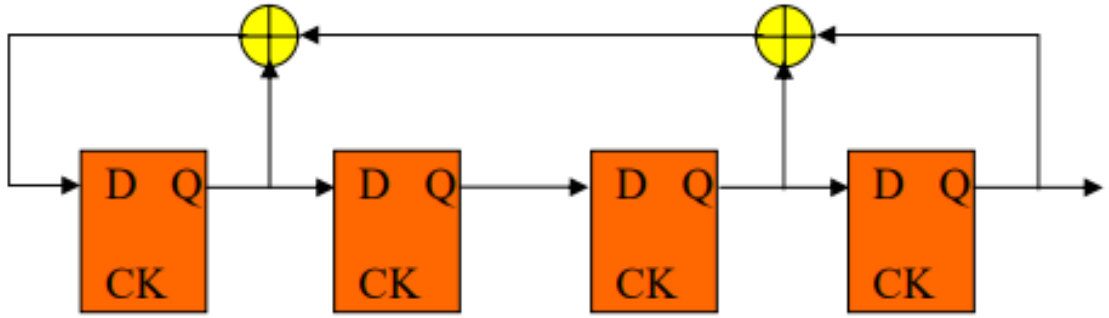
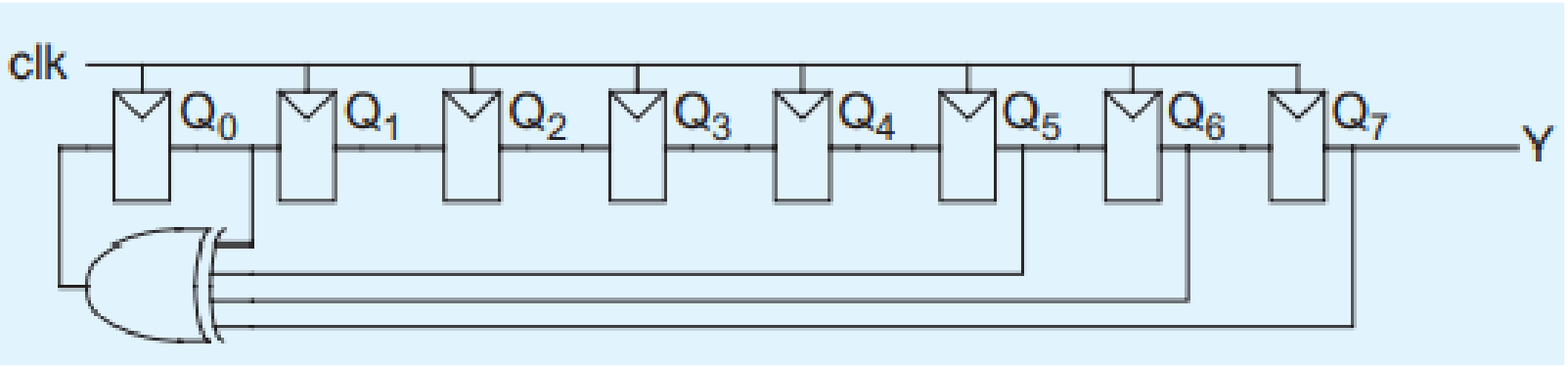


FIGURE 11.54 3-bit LFSR



$P(x) = 1 + x + x^3 + x^4$



$P(x) = 1 + x^1 + x^6 + x^7 + x^8$

- LFSRs are used for **high-speed counters** and **pseudo-random number generators.**
  - pseudo-random sequences are handy for built-in self-test and bit-error-rate testing in communications links
  - many spread spectrum communications systems such as GPS and CDMA

For certain lengths, N, more than two taps

TABLE 11.8 Characteristic polynomials

N	Polynomial
3	$1 + x^2 + x^3$
4	$1 + x^3 + x^4$
5	$1 + x^3 + x^5$
6	$1 + x^5 + x^6$
7	$1 + x^6 + x^7$
8	$1 + x^1 + x^6 + x^7 + x^8$
9	$1 + x^5 + x^9$
15	$1 + x^{14} + x^{15}$
16	$1 + x^4 + x^{13} + x^{15} + x^{16}$
23	$1 + x^{18} + x^{23}$
24	$1 + x^{17} + x^{22} + x^{23} + x^{24}$
31	$1 + x^{28} + x^{31}$
32	$1 + x^{10} + x^{30} + x^{31} + x^{32}$

XOR/XNOR circuit forms

- One of the chronic difficulties in CMOS circuit design is to construct a fast, compact, low-power XOR or XNOR gate

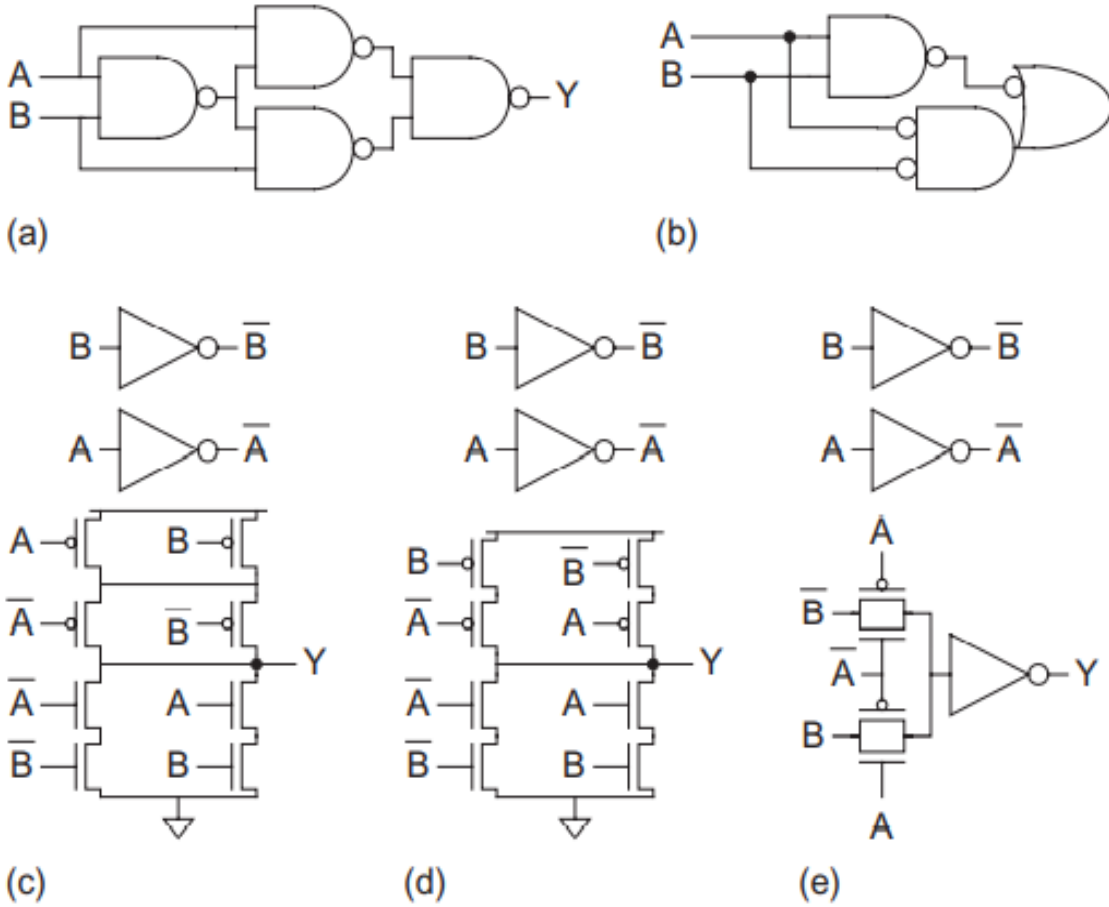


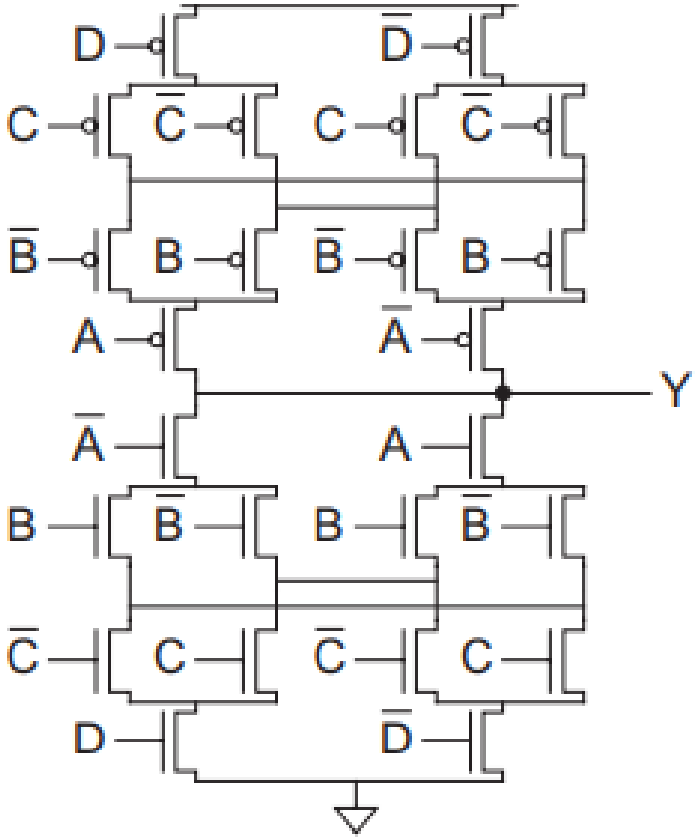
FIGURE 11.59 Static 2-input XOR designs

Figure(f) is the 6-transistor "invertible inverter" design.

- When A is 0, the transmission gate turns on and B is passed to the output.
- When A is 1, the A input powers a pair of transistors that invert B.
- It is compact, but nonrestoring.

- switch-level simulators cannot handle this unconventional design.
- Figure (g) [Wang94] is a compact and fast 4-transistor pass-gate design, but does not swing rail to rail.

XOR gates with 3 or 4 inputs can be **more compact,** although **not necessarily faster** than a cascade of 2-input gates.



4 input CMOS XOR