VLSI DESIGN

Mr.V.R Seshagiri Rao, Associate Professor
Dr. Manisha G, Assistant Professor
Dr V.Vijay, Assistant Professor
K.S.Indrani, Assistant Professor
Department of Electronics and Communication Engineering

UNIT-I
MOSFETs

Mr.V.R Seshagiri Rao, Associate Professor
Dr. Manisha G, Assistant Professor
Dr V.Vijay, Assistant Professor
K.S.Indrani, Assistant Professor
Department of Electronics and Communication Engineering

Unit-I

- Fundamentals of MOSFETs
- Weak & strong inversion conditions
- Threshold voltage concept in MOSFETs
- Current - voltage characteristics of a MOSFET
- MOSFET parasitics
- Trends & projections in VLSI design & technology
- Scaling in MOS devices
- Effects in scaling of MOS devices
- BiCMOS technologies
- CMOS nanotechnology

Fundamentals of MOSFETs

Classes of Field Effect Transistors

- ✓Metal-Oxide-Semiconductor Field Effect Transistor
- ✓To be studied in this course
- ✓Metal-Semiconductor Field Effect Transistor
- ✓Junction Field Effect Transistor
- ✓High Electron Mobility Transistor or Modulation Doped FET
- ✓Fast Reverse / Fast Recovery Epitaxial Diode
- ✓DNA Field Effect Transistor
- ✓The conduction path is through a strand of DNA

Field Effect Transistors

The conductivity (or resistivity) of the path between two contacts, the source and the drain, is altered by the voltage applied to the gate. Device is also known as a voltage controlled resistor.

Types of MOSFETS

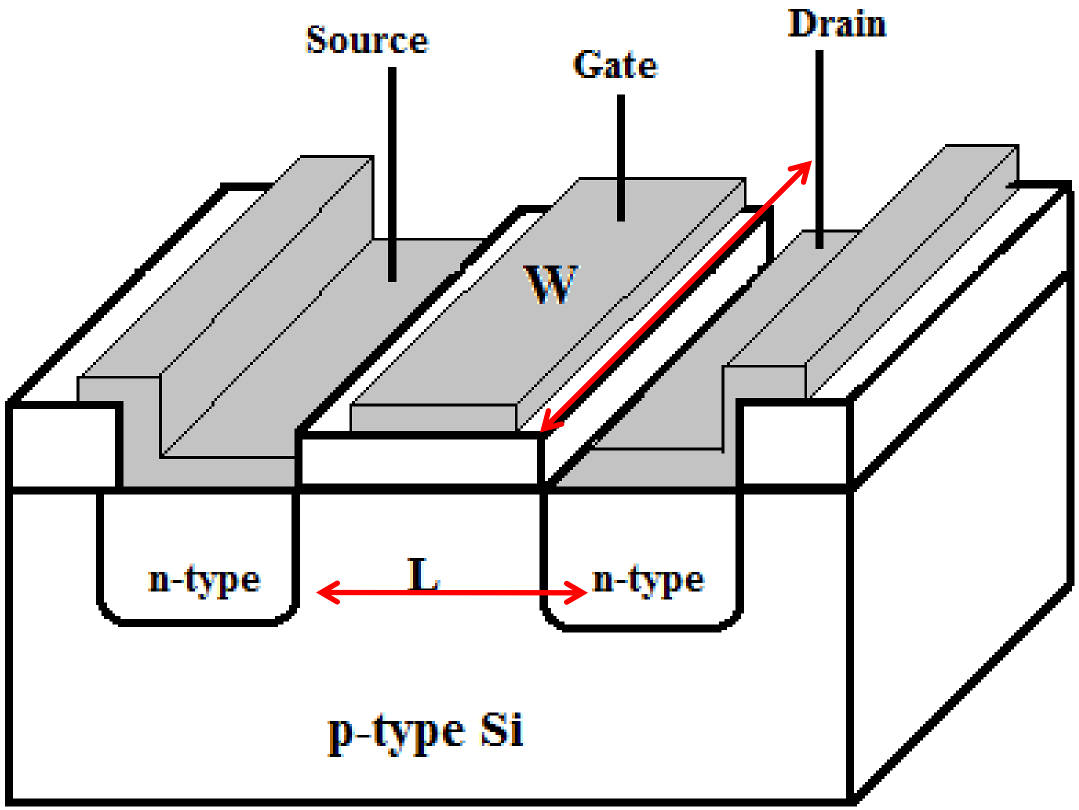
n-channel
Enhancement Mode
(nMOSFET)

p-channel
Enhancement Mode
(pMOSFET)

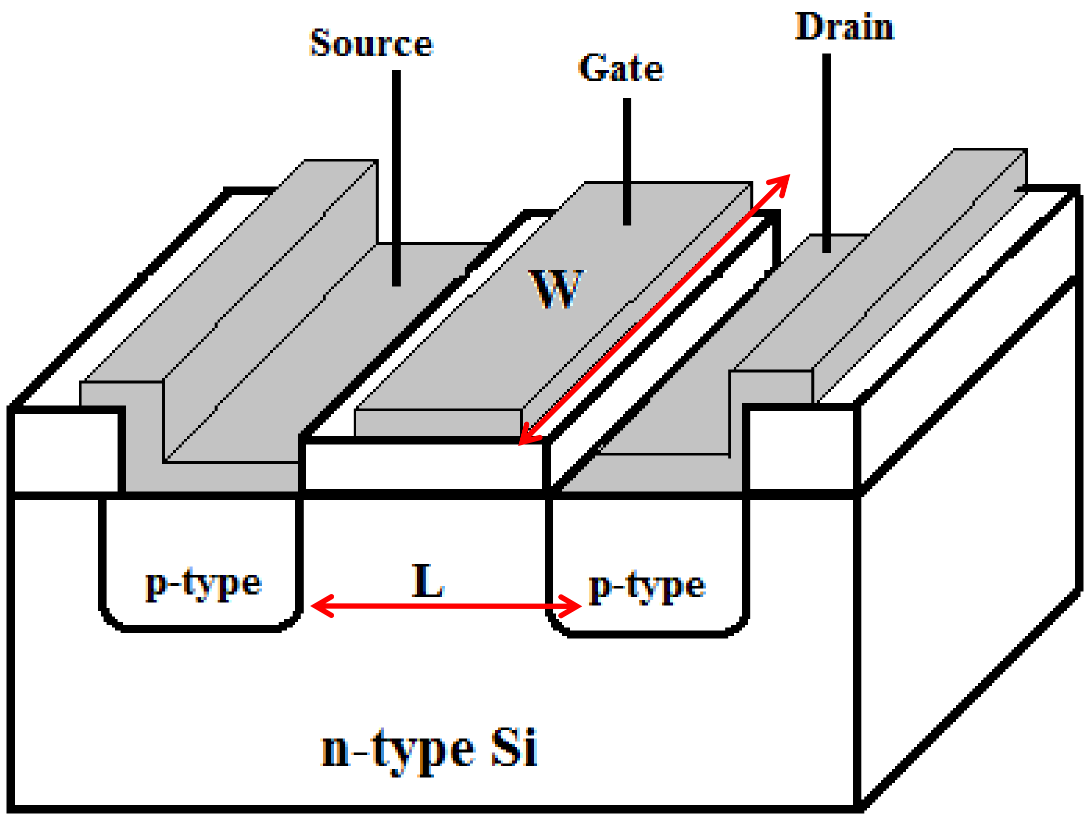
n-channel
Depletion Mode
(nMOSFET)

p-channel
Depletion Mode
(pMOSFET)

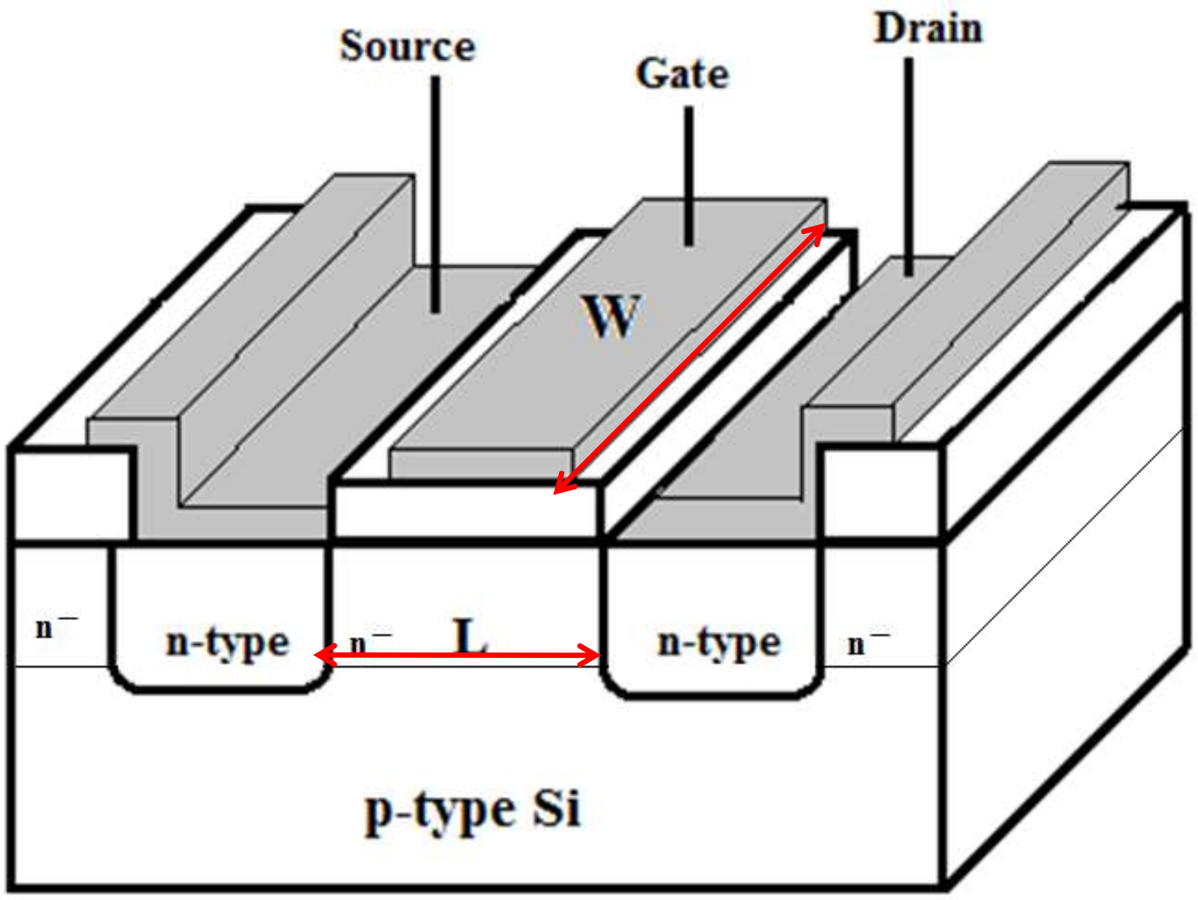
Cross-Sectional View of n channel planar Enhancement Mode Transistor



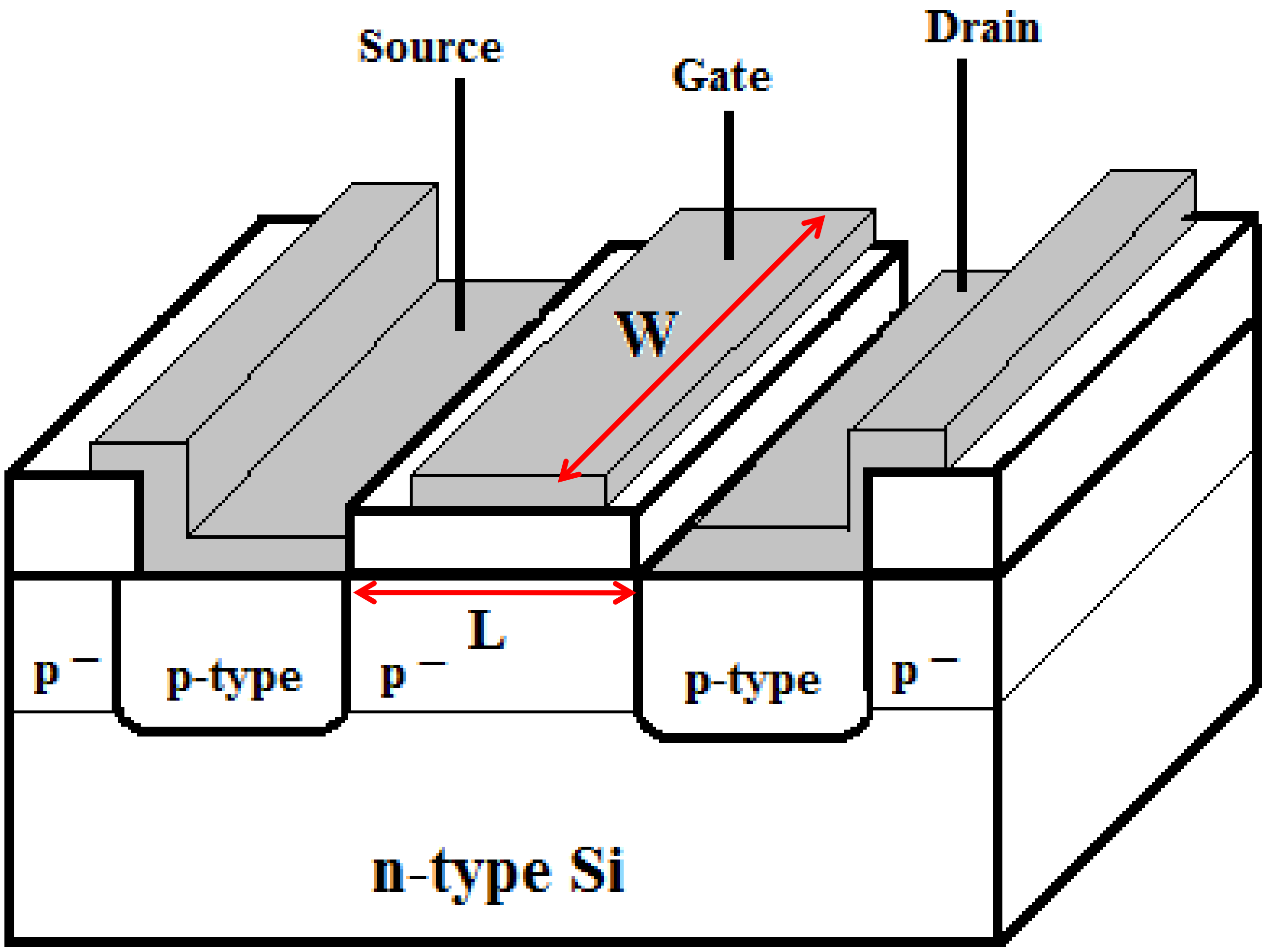
p channel
Enhancement Mode Transistor



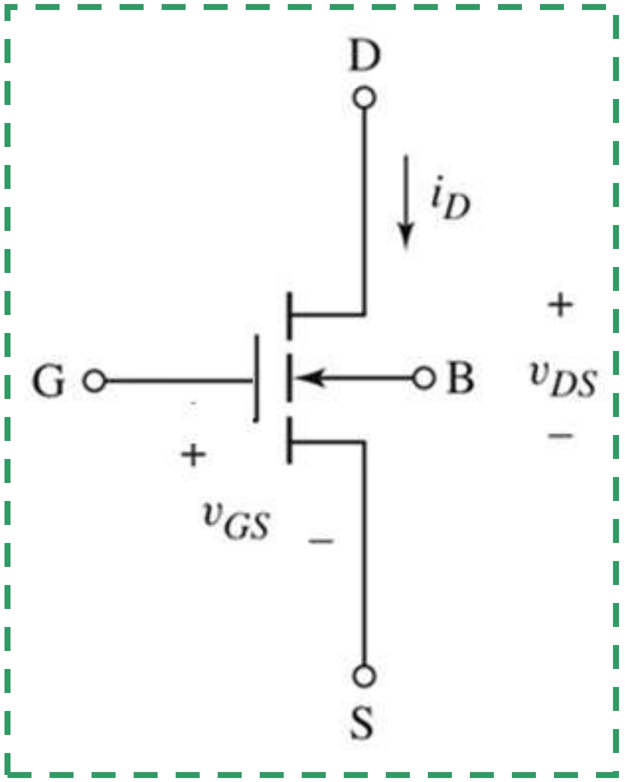
n channel
Depletion Mode Transistor

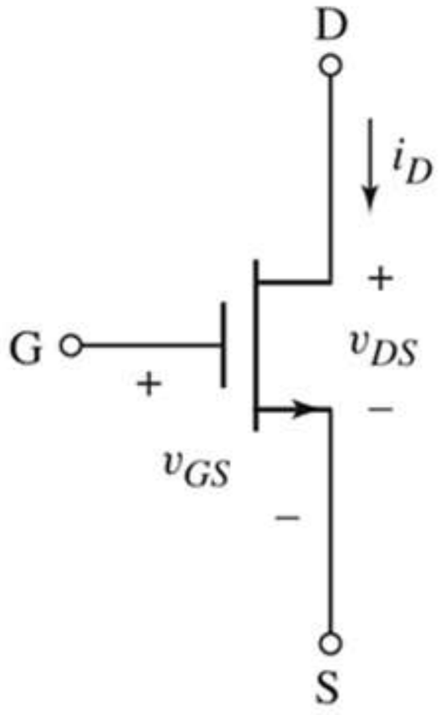


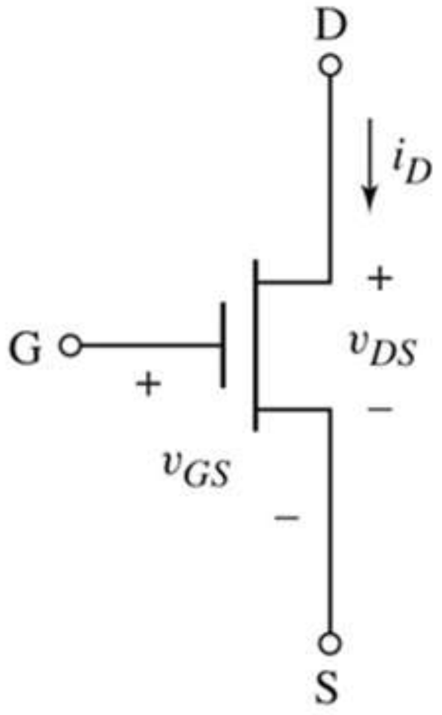
p channel
Depletion Mode Transistor



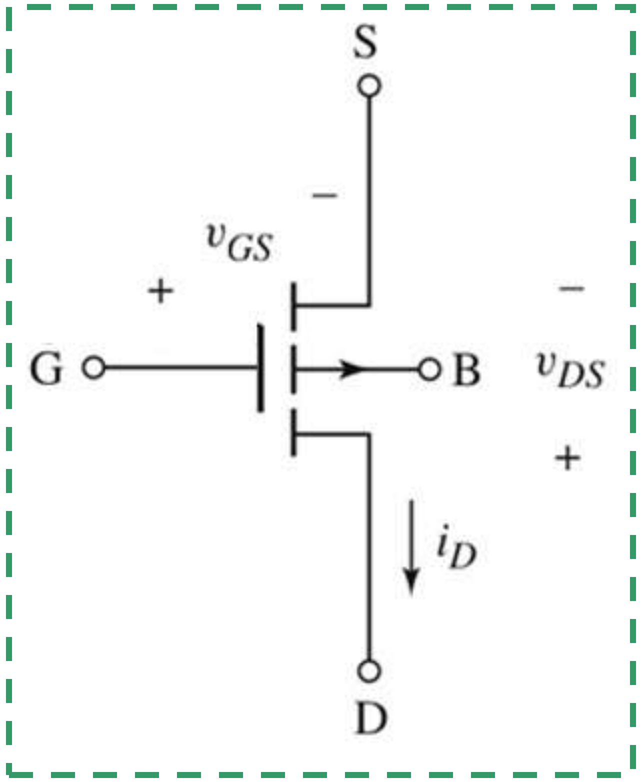
Symbols for n channel Enhancement Mode MOSFET

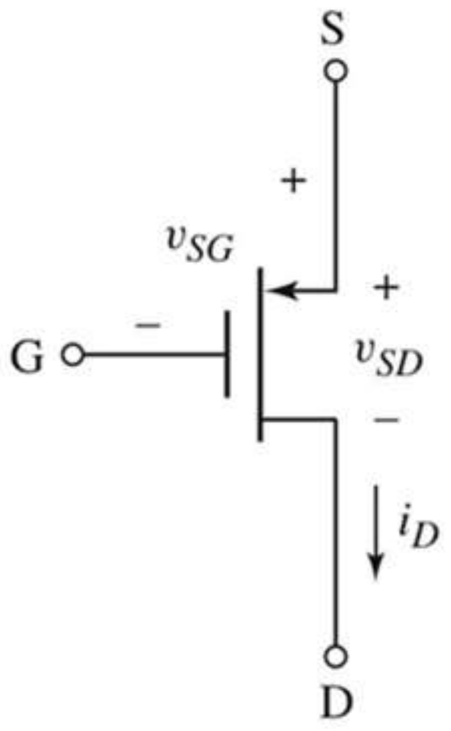


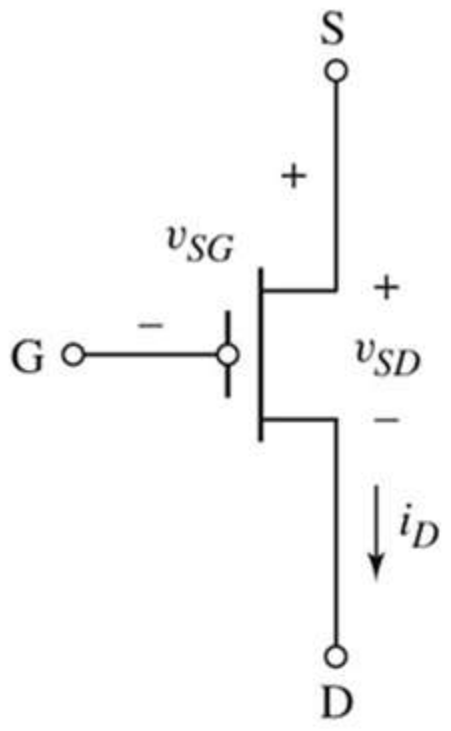




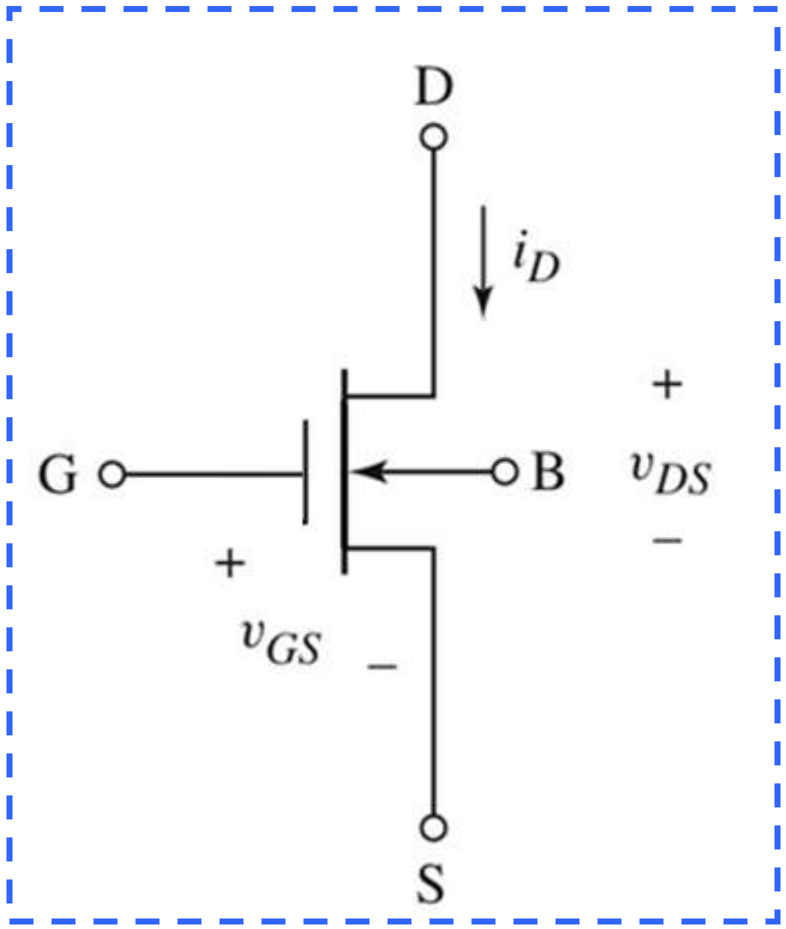
Symbols for p channel Enhancement Mode MOSFET

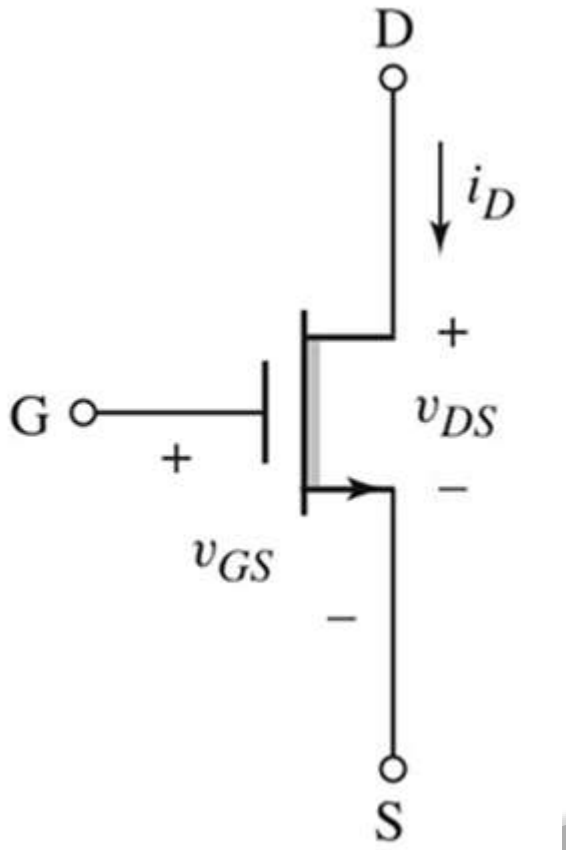




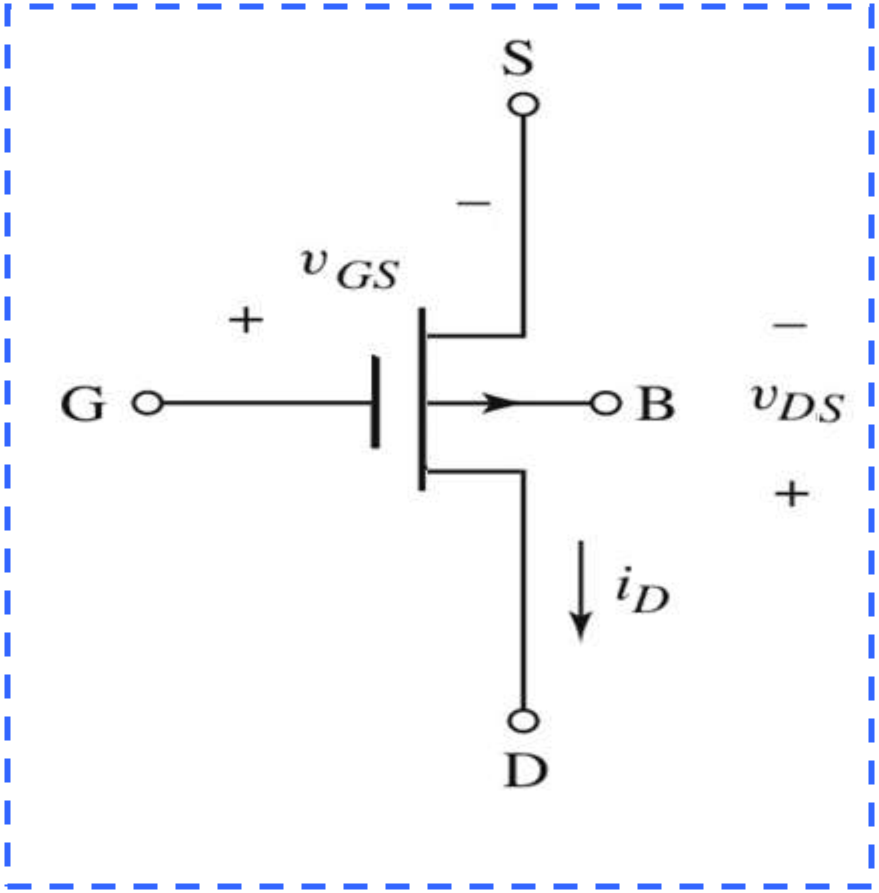


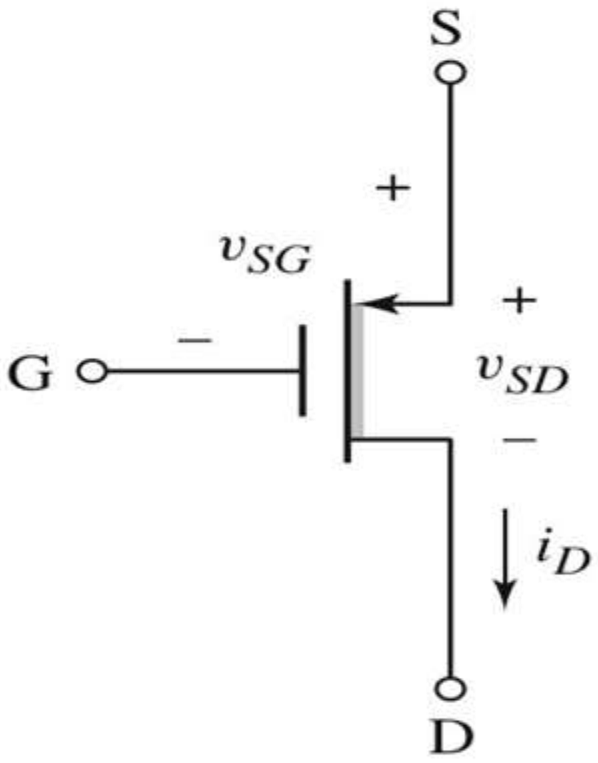
Symbols for n channel Depletion Mode MOSFET



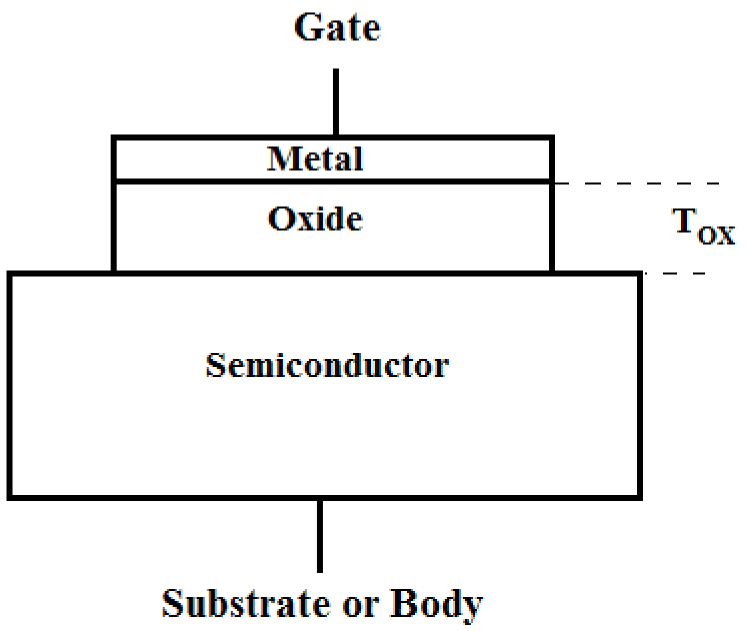


Symbols for p channel Depletion Mode MOSFET



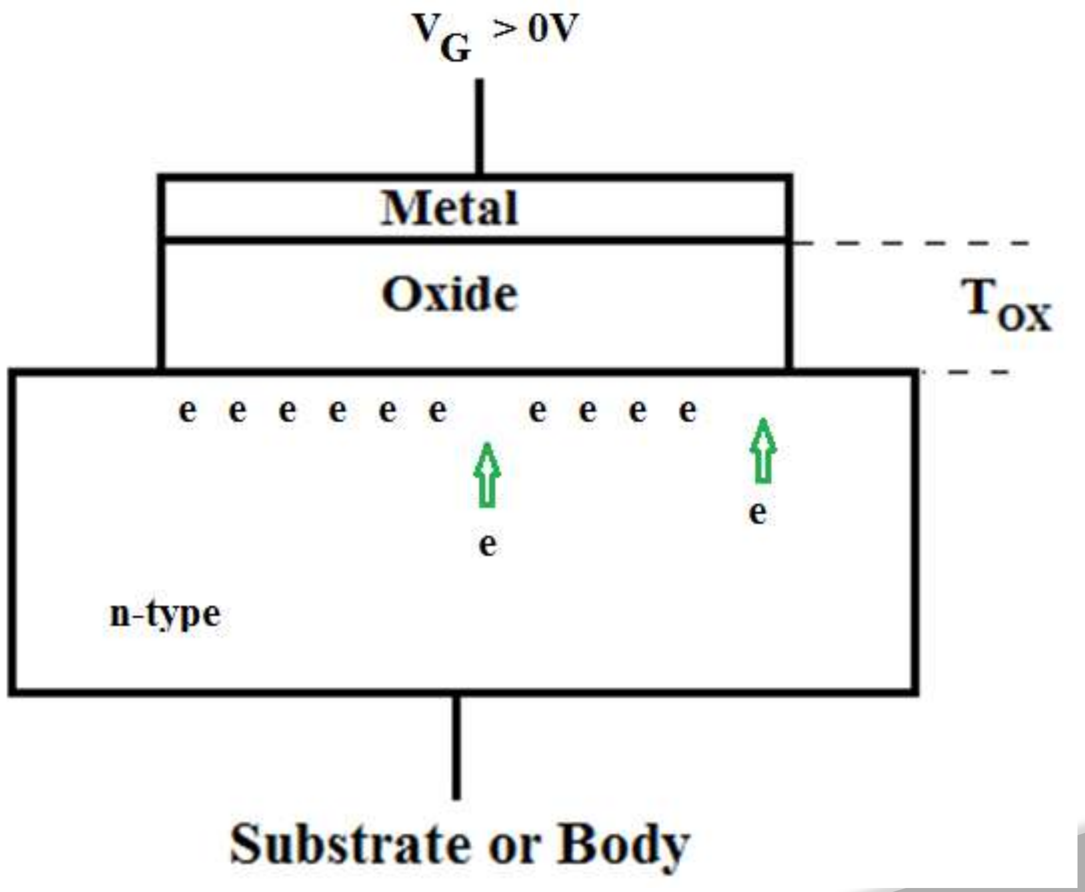


MOS Capacitor



MOS Capacitor

Parallel plate capacitor

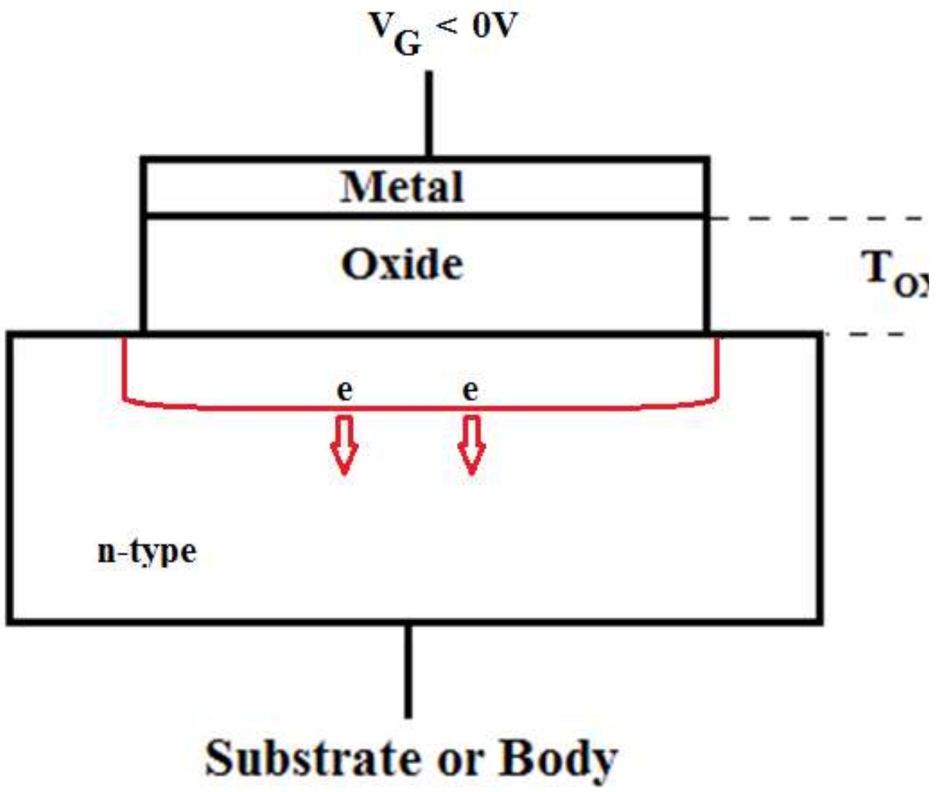


Accumulation

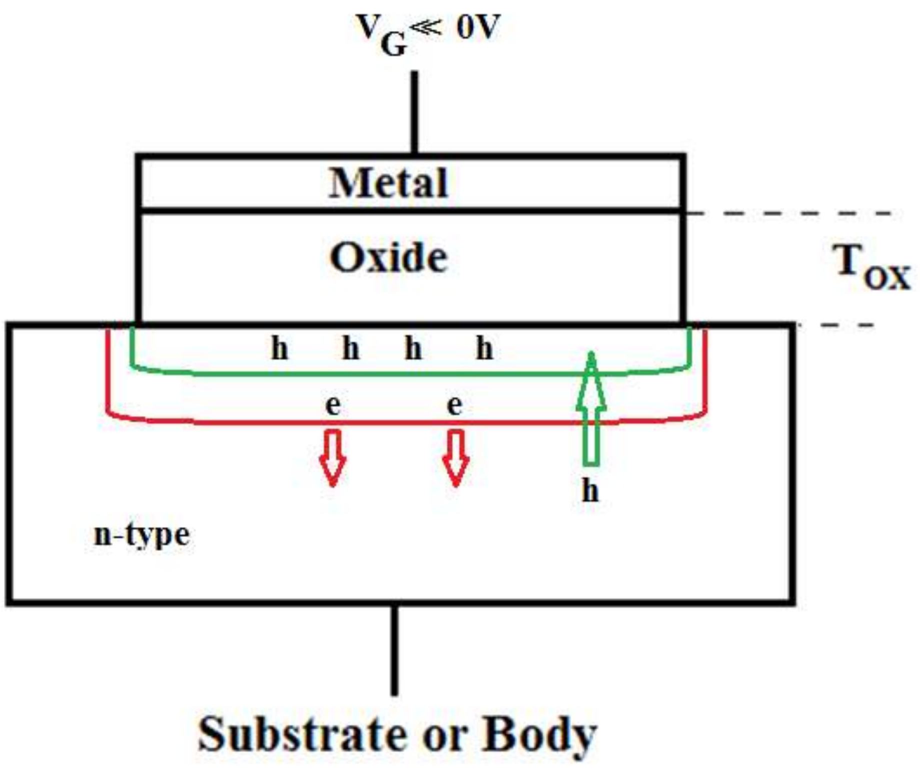
Positive gate bias

Electrons attracted to gate

Negative gate bias: Holes attracted to gate

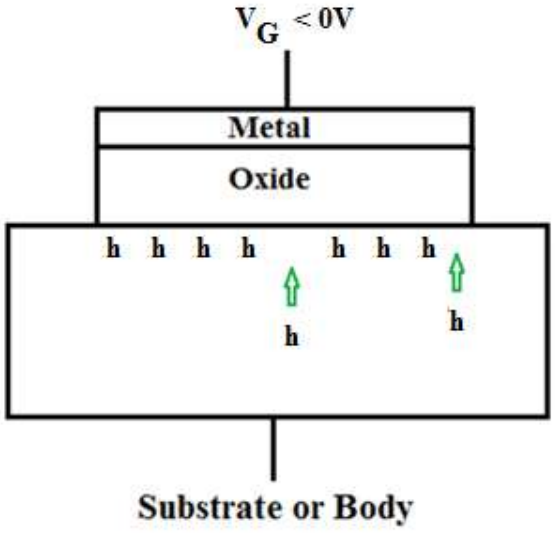


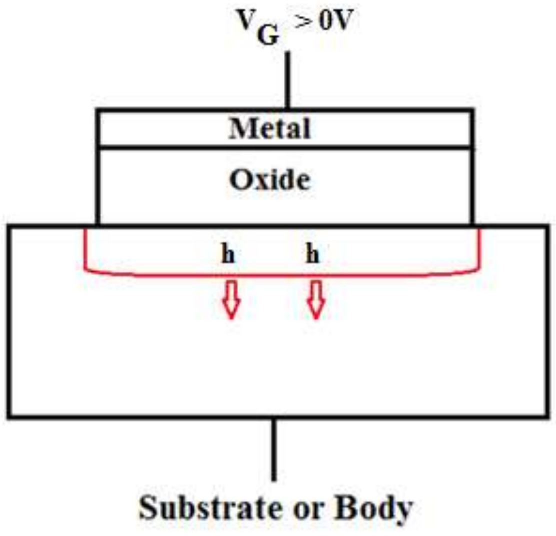
Depletion

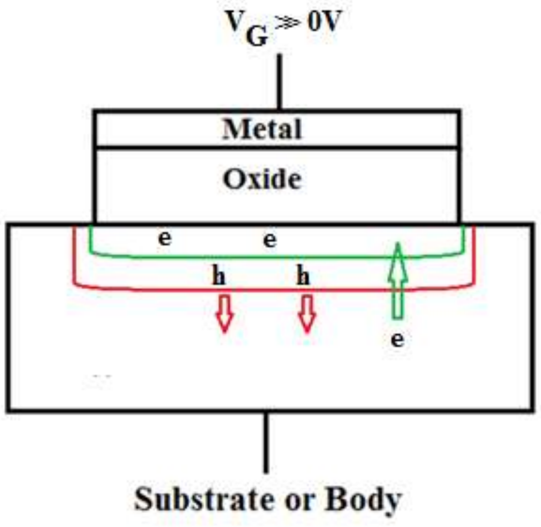


Inversion

MOS Capacitor:p-type semiconductor







Accumulation Depletion Inversion

Threshold Voltage

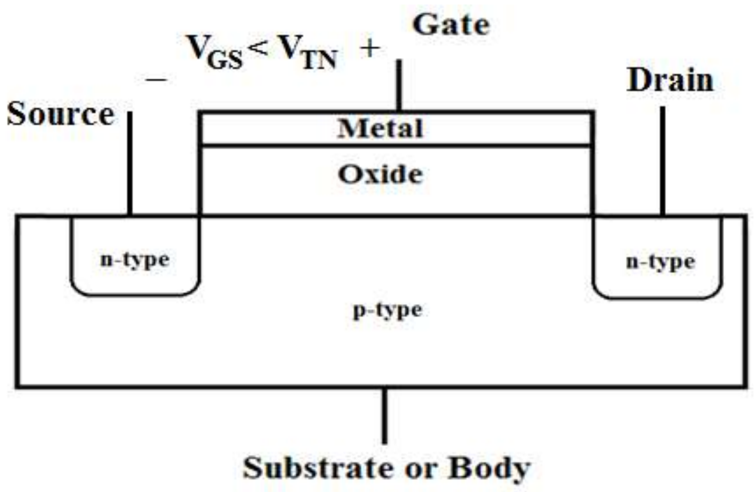
The gate voltage that causes the concentration of electrons immediately under the gate oxide is equal to the concentration of holes is called the threshold voltage.

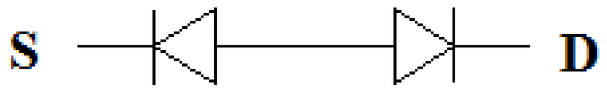
- Enhancement mode FETs
  - NMOS $V_G = V_{TN}$
    - When enough electrons have been attracted to the oxide-semiconductor interface to create a path for current to flow between the source and drain.
  - PMOS $V_G = V_{TP}$
    - When holes have been attracted to the oxide-semiconductor interface to create a path for current to flow between the source and drain.

Threshold Voltage

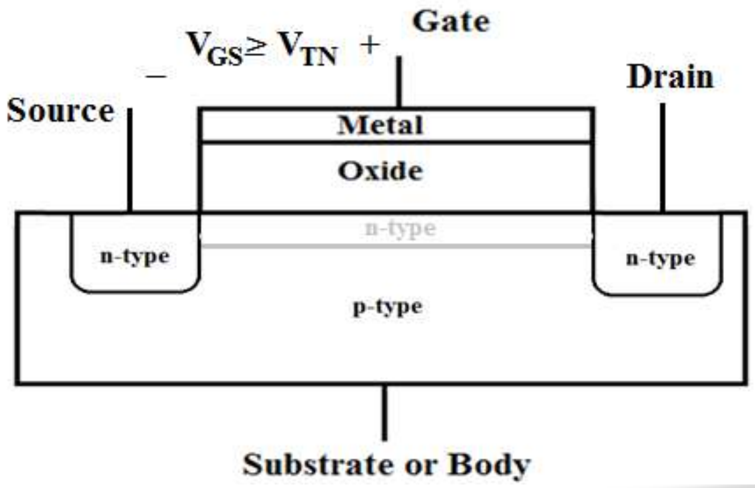
- Depletion mode FETs
  - NMOS $V_G = V_{TN}$
    - When holes have been attracted to the oxide-semiconductor interface to stop current from flowing between the source and drain.
  - PMOS $V_G = V_{TP}$
    - When electrons have been attracted to the oxide-semiconductor interface to stop current from flowing between the source and drain.

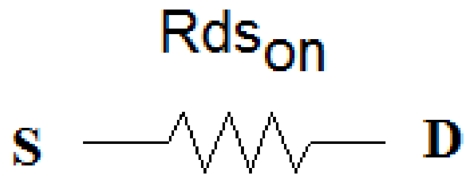
MOSFETS





Before electron inversion layer is formed

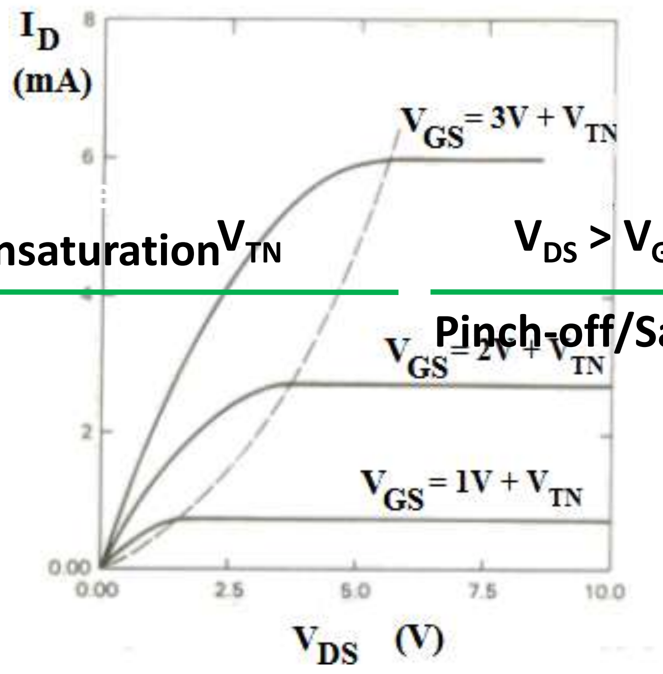




After electron inversion layer is formed

Family of I_D Versus V_{DS} Curves

Enhancement-Mode nMOSFET



Nonsaturation V_{TN}

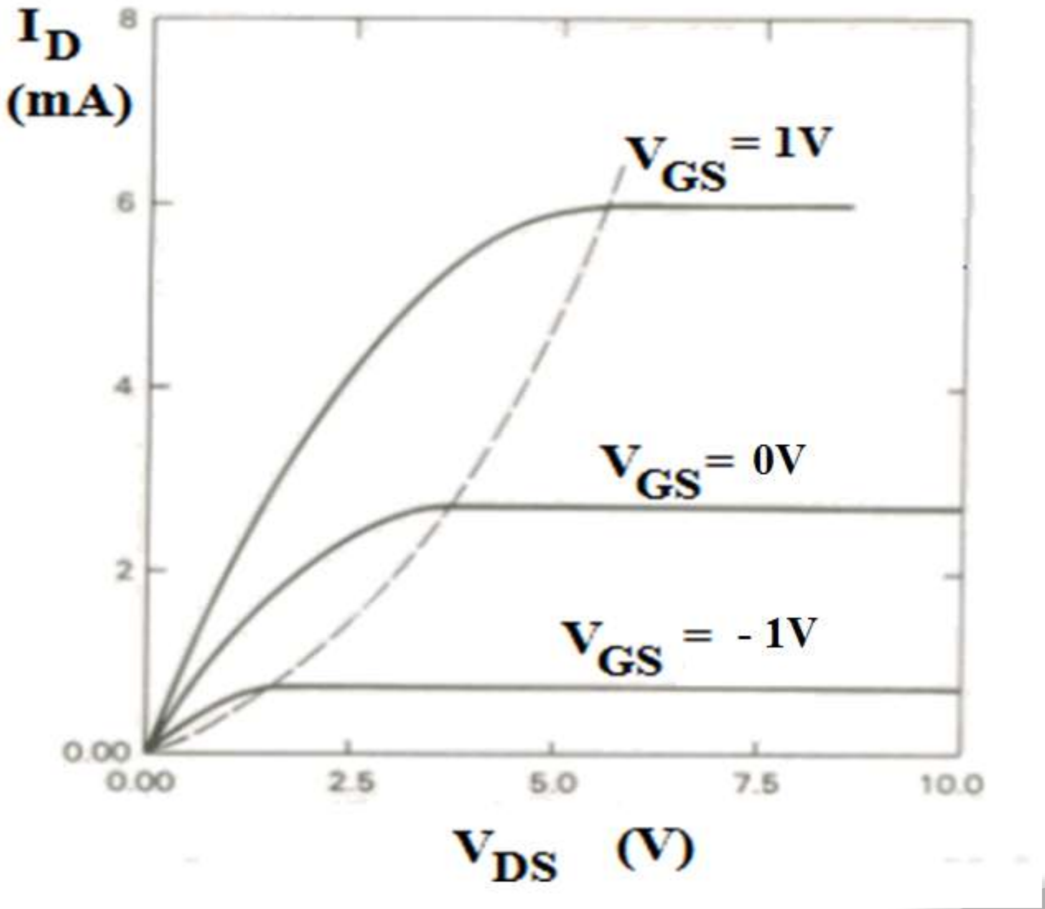
$V_{DS} > V_{GS} – V_{TN}$

Pinch-off/Saturatio

Cut-off $V_{GS} < V_{TN}$

Family of I_D Versus V_{DS} Curves:

Depletion-Mode nMOS



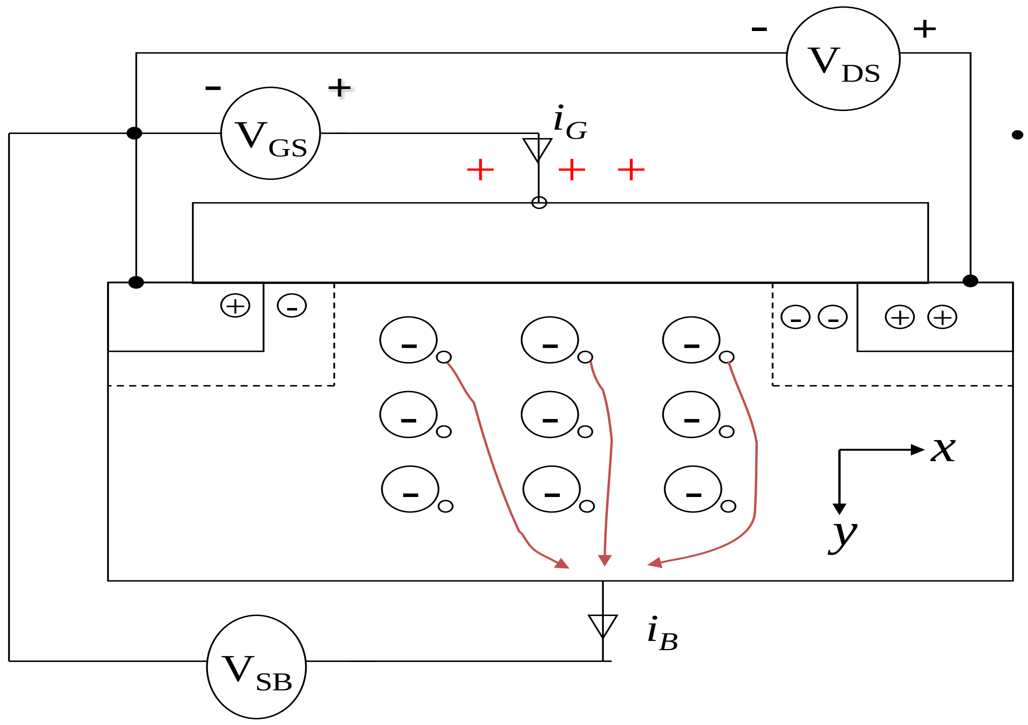
Assuming that $V_{TN} < -1V$

Summary of I-V Relationships

Region	NMOS
Nonsaturation/ Triode	$V_{DS} < V_{DS}(sat)$ $I_D = k_n' \frac{W}{L}\left[(V_{GS} - V_{TN})V_{DS} - \frac{1}{2}V_{DS}^2\right]$ $R_{DSon} = \frac{V_{DS}}{I_D}$
Saturation/ Pinch-off	$V_{DS} > V_{DS}(sat)$ $I_D = \frac{k_n'}{2}\left(\frac{W}{L}\right)[V_{GS} - V_{TN}]^2$
Transition between triode and pinch-off	$V_{DS}(sat) = V_{GS} - V_{TN}$
Enhancement Mode	$V_{TN} > 0$ V, $I_D \geq 0$ mA, $I_D = I_S$, $I_G = 0$ mA

SUB-THRESHOLD CONDITION (DEPLETION)

SUB-THRESHOLD CONDITION (DEPLETION)



- Depletion layer forms

$$i_G = \frac{\partial Q'_G}{\partial t}$$

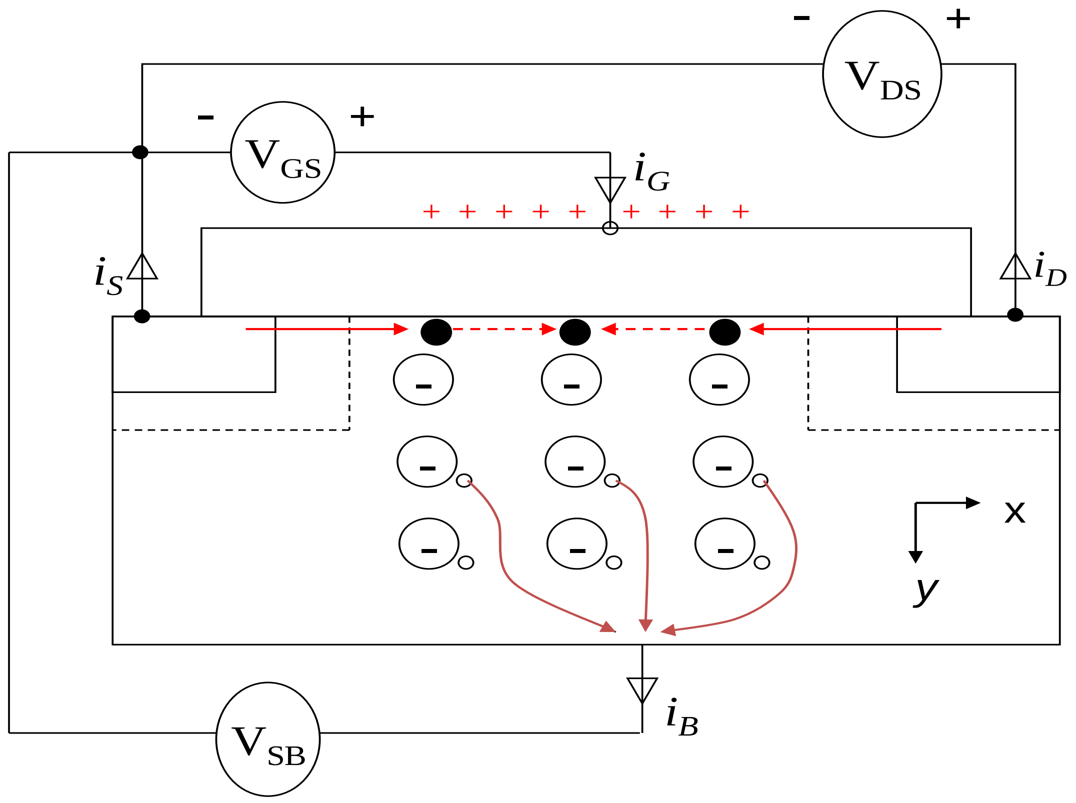
$$i_B = \frac{\partial Q'_B}{\partial t}$$

$$C'_{GG} = \frac{\partial Q'_G}{\partial V_G}$$

$$C'_{BG} = \frac{\partial Q'_B}{\partial V_G}$$

ON CONDITION (Strong Inversion)

ON CONDITION (Strong Inversion)



- Inversion layer forms

$$i_G = \frac{\partial Q'_G}{\partial t} \qquad i_B = \frac{\partial Q'_B}{\partial t}$$

$$i_S = \frac{\partial Q'_{nS}}{\partial t} \qquad i_D = \frac{\partial Q'_{nD}}{\partial t}$$

$$C'_{SG} = \frac{\partial Q'_{nS}}{\partial V_G}$$

CMOS: the Industrial drive

Nodes relate to the DRAM half pitch, i.e., the width, and space in between, metal lines connecting DRAM bit cells

CMOS Device Scaling Demonstration

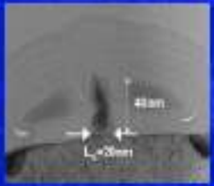
90nm Node
2003

50nm Length
(IEDM2002)

65nm Node
2005

30nm
Prototype
(IEDM2000)

45nm Node
2007



20nm Prototype
(VLSI2001)

32nm Node
2009

15nm Prototype
(IEDM2001)

22nm Node
2011

10nm Prototype
(DRC 2003)

16 nm node
2013

11nm node
2015

TBD

8 nm node
2017

TBD

Intel research devices scale to 10nm (16nm node)
Channel engineering solutions (Nanowires/Nanotubes) are being investigated to extend device scaling through end of next decade

intel

*Source: Intel; Morales and Lieber Science, **279**, 208, 1998*

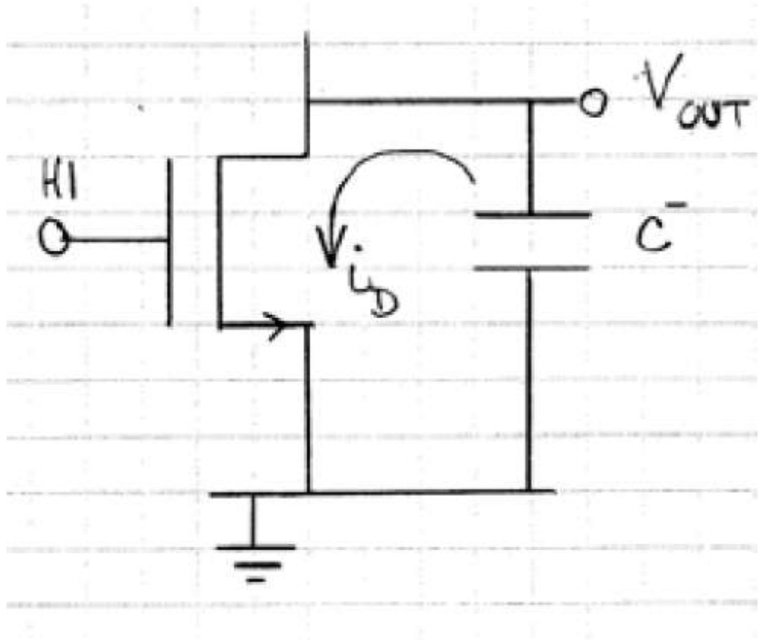
Logic speed is about Q and I

$$\tau = \frac{C \Delta V}{I}$$

$$I_D = Z C_{ox} \left[V_{GS} - V_{T0} - m \frac{V_{DS}}{2} \right] \cdot \mu_{\text{eff}} \frac{V_{DS}}{L}$$

Need:

- high μ - certainly
- Low L - but it adversely affects V_T
- High C_{ox} - but low $C_{ox}ZL$
- Low V_{DD} - but it adversely affects I_{ON}
- Low V_T - but it adversely affects I_{SUBT}

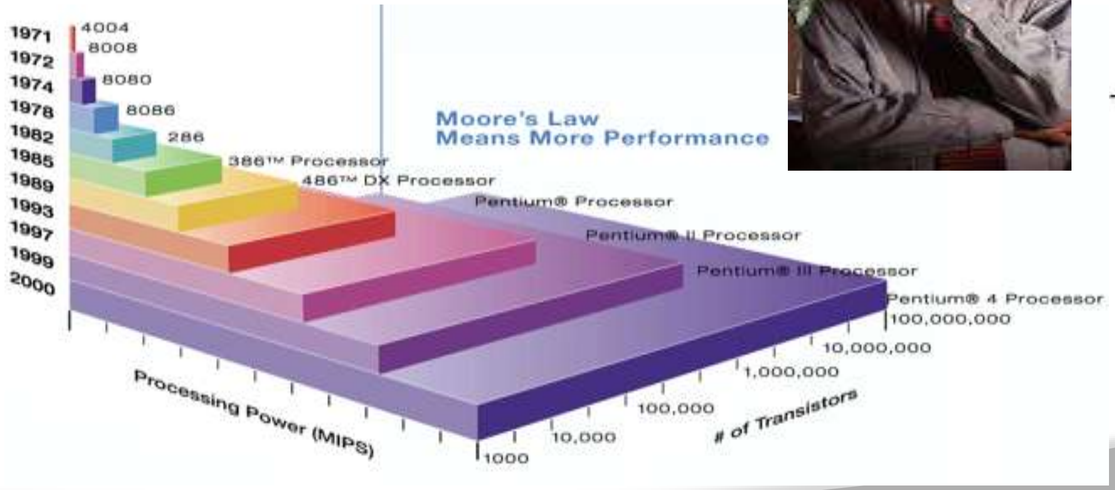


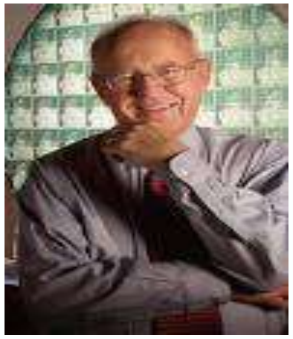
3 major concerns for digital CMOS

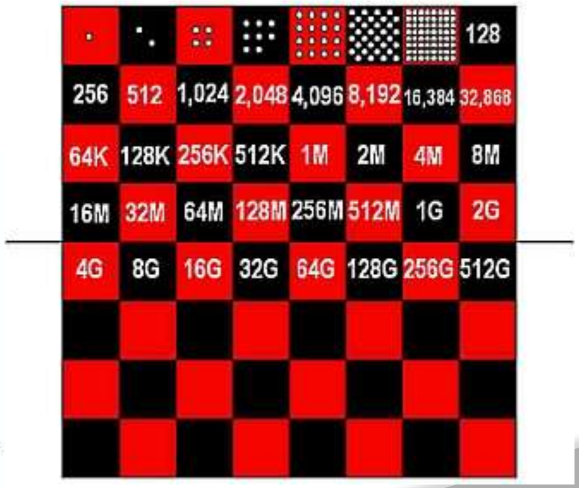
1. Increasing I_{ON} via mobility improvement
2. Reducing gate leakage via thicker, high-*k* dielectrics
3. Controlling V_T and I_{subt} via suppression of the short-channel effect

Moore’s Law economics…

- Moore’s Law - #DRAM Bits per chip doubles every 18 months
- ~25% bigger chips/wafers
- ~25% design improvements
- ~50 % Lithography – ability to print smaller features

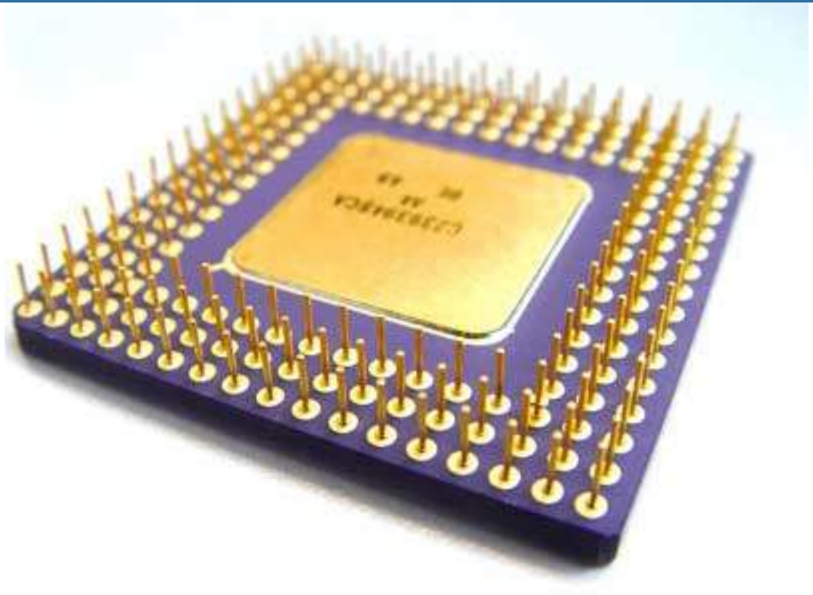





128
256 512 1,024 2,048 4,096 8,192 16,384 32,868
64K 128K 256K 512K 1M 2M 4M 8M
16M 32M 64M 128M 256M 512M 1G 2G
4G 8G 16G 32G 64G 128G 256G 512G

ponentials !!
(461 B tonnes)

When the chip's down…



- With feature size shrink of √2 (typical generation)
  - 2x #transistors/unit area
  - 2X Higher speed (f_{max})
  - Fixed cost per wafer
  - Smaller (2x), Faster (2x), cheaper – strong economic driving force
  - 30% improvement in cost per function per year

CMOS Device Scaling

Parameters	Variables	Scaling Factor
Dimensions	W, L, x_{ox}, x_j	$1/\lambda$
Potentials	V_{ds}, V_{gs}	$1/k$
Doping Concentration	N	λ^2/k
Electric Field	E	λ/k
Current	I_{ds}	λ/k^2
Gate Delay	T	k/λ^2

λ=dimensional scaling factor

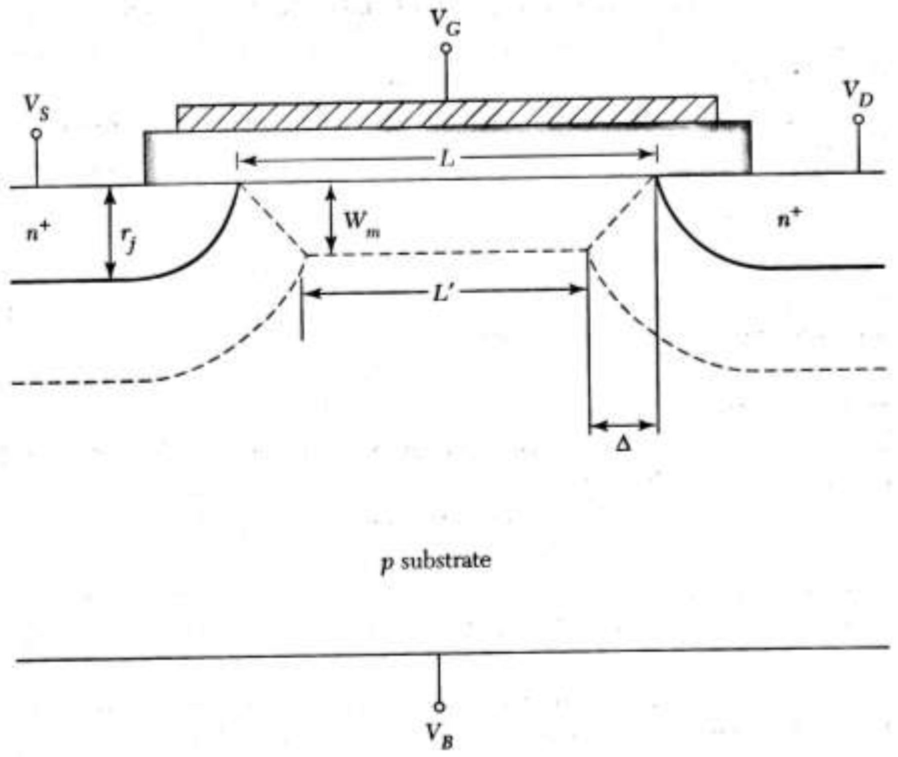
k=supply voltage scaling factor

Constant Field Scaling: keep E constant in channel k= λ

Constant Voltage Scaling: keep supply voltage constant **k=1**(used for submicron scaling)

Parameters	**Const Field**	**Const Volt**
Dimensions	$1/\lambda$	$1/\lambda$
Potentials	$1/\lambda$	1
Doping Concentration	λ	λ^2
Electric Field	1	λ
Current	$1/\lambda$	λ
Gate Delay	$1/\lambda$	**$1/\lambda^2$**

Short Channel MOSFET Geometry



How much channel charge does the gate control?

Threshold Voltage Shift

- **Threshold Voltage**

$$V_T = \psi_s + V_i = 2\psi_B - \frac{Q_B}{C_i}$$

$$\Delta V_T = V_T(short) - V_T(long) = \frac{(Q_{BS} - Q_{BL})}{C_i}$$

- **Long Channel transistor:**

$$Q_{BL} = -qN_A \frac{ZLW_{\max}}{ZL} - qN_A W_{\max}$$

$$Q_{BL} = -qN_A \frac{ZLW_{\max}}{ZL} - qN_A W_{\max}$$

Charge control region...

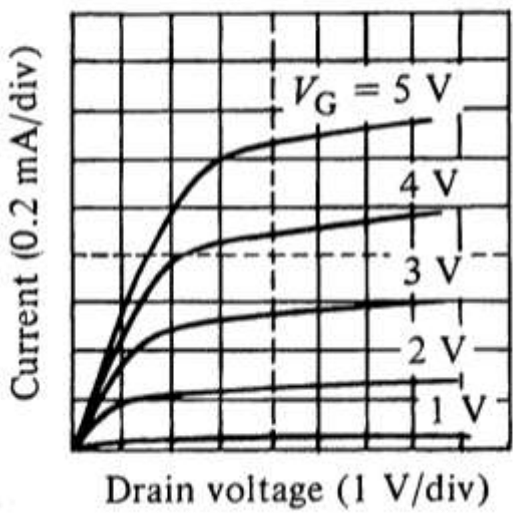
- **Short Channel transistor: charge in trapezoidal region**

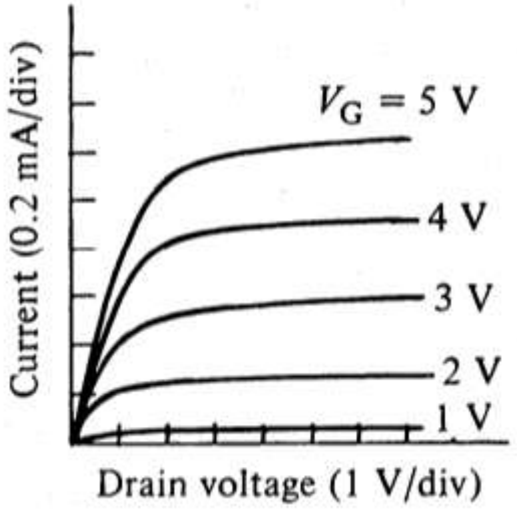
$$Q_{BS} = -qN_A \frac{\left[\frac{L+L'}{2}\right] ZW_{\max}}{ZL} = -qN_A W_{\max} \frac{L+L'}{2L}$$

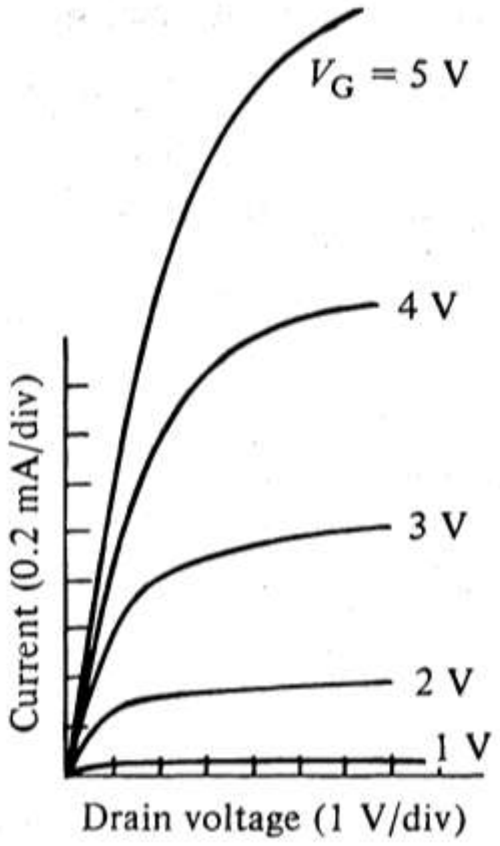
$$\Delta V_T = \frac{(Q_{BS} - Q_{BL})}{C_i} = -\frac{qN_A W_{\max}}{C_i}\left(1 - \frac{L+L'}{2L}\right)$$

Velocity Saturation Effect – suppressed drain current

$$g_{m-sat} \equiv \left.\frac{\partial I_D}{\partial V_G}\right|_{sat} = ZC_i \upsilon_{sat} = const.$$





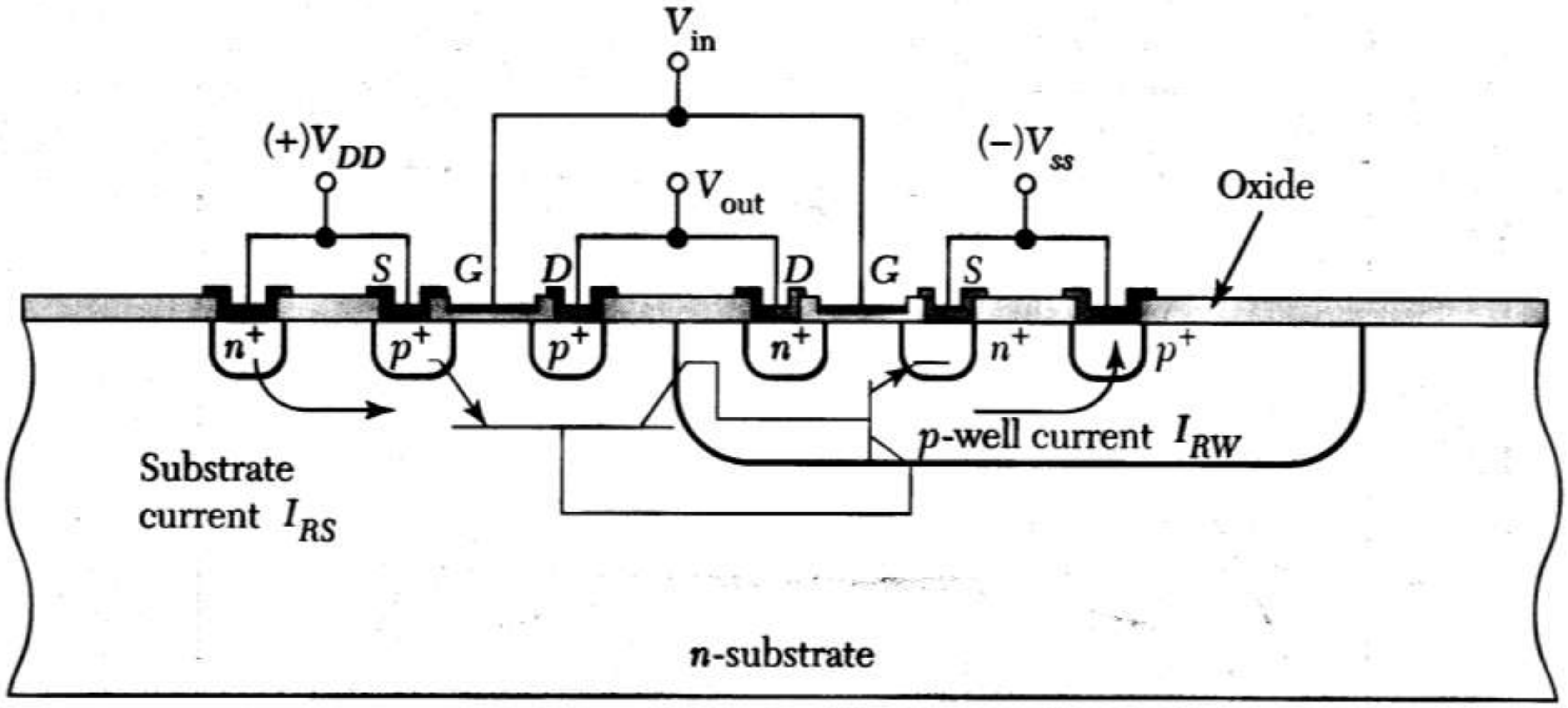


w/o

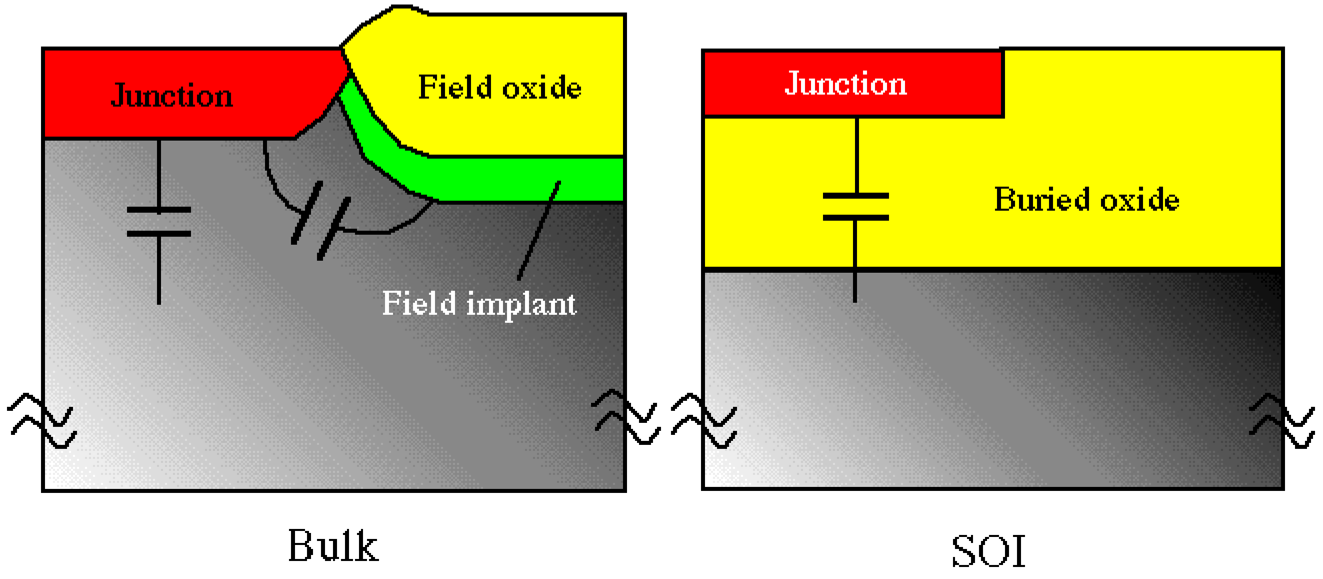
Velocity Saturation

Velocity Saturation

CMOS Inverter Latch-up – p-well technology

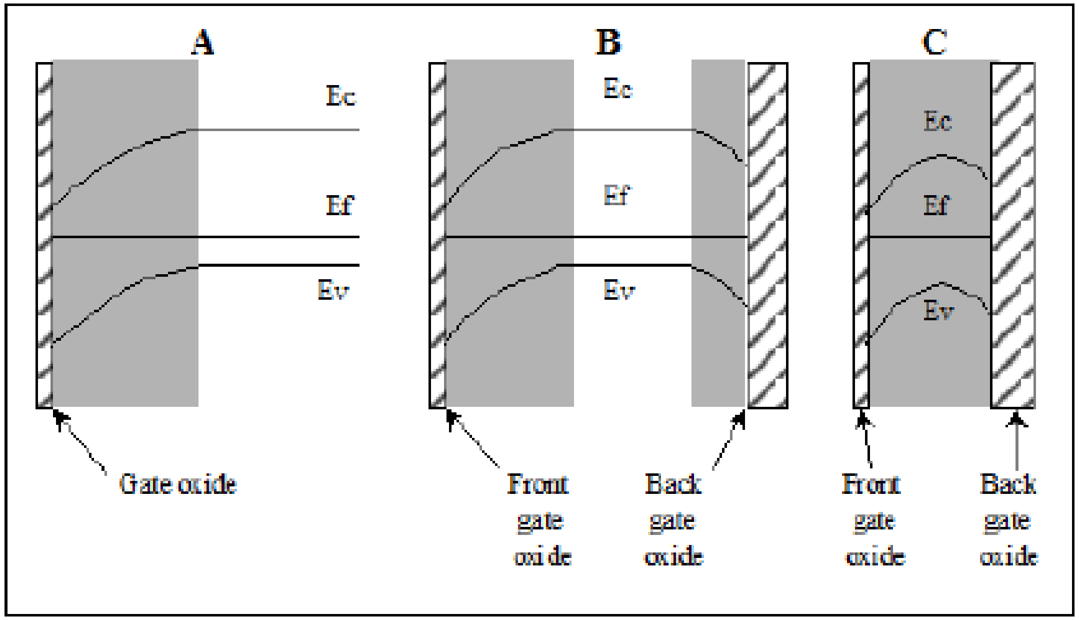


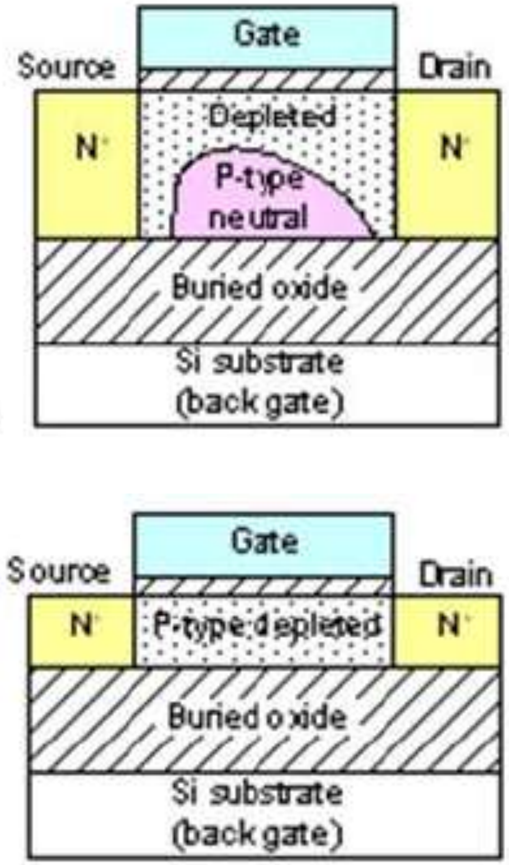
SOI



- No body effect parameter (depletion width fixed)
- Junction to substance parasitic capacitance small (faster switch!)
- No latch-up between NMOS and PMOS (no substrate)

Band diagram in SOIs





Fully depleted SOI MOSFET

Bi-CMOS Technology

BiCMOS technology combines Bipolar and CMOS transistors onto a single integrated circuit where the advantages of both canbe utilized.

Advantages of BiCMOS Technology

- Improved speed over CMOS
- Lower power dissipation than Bipolar
- Flexible input/outputs
- High performance analog
- Latch up immunity

Evolution of BiCMOS from CMOS

- BiCMOS technology has evolved from CMOS processes in order to obtain the highest CMOS performance possible.
- The bipolar processing steps have been added to the core CMOS flow to realize the desired device characteristics such as adding masks for bipolar transistor (BJT) fabrication.

BiCMOS Process Flow

We start up with a lightly-doped P-type wafer and form the buried N+ layer by ion implantation of antimony. The pattern is etched in a thick oxide covering the substrate. The structure before the antimony implantation

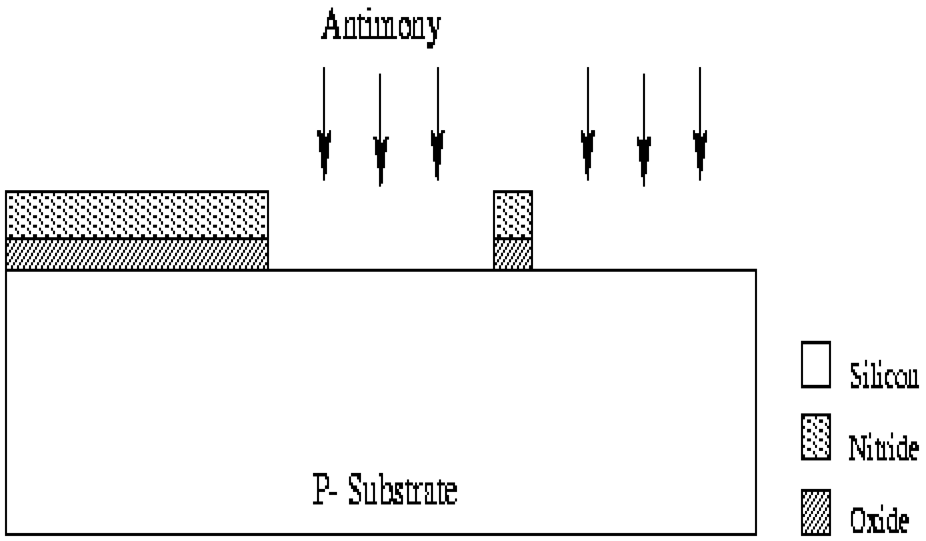


Figure 1: Device cross-section of BiCMOS process showing *N+* buried layer implant.

BiCMOS Process Flow

A high temperature anneal is performed to remove damage defects and to diffuse the antimony into the substrate. During this anneal an oxide is grown in the buried N+ windows. To achieve breakdown between the buried N+ regions a self-aligned punch through implant is performed. Therefore, the nitride mask is selectively removed and the remaining oxide serves as blocking mask for the buried P-layer implant (see Fig. 2).

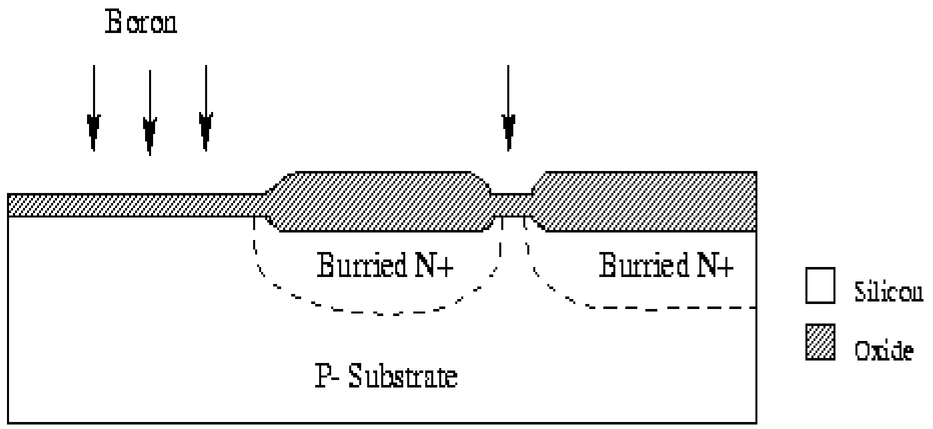


Figure 2: Device cross-section of BiCMOS process showing *P* buried layer self aligned implant

BiCMOS Process Flow

After removing all oxide a thick epitaxial layer with intrinsic doping is grown on top (see Fig. 3).

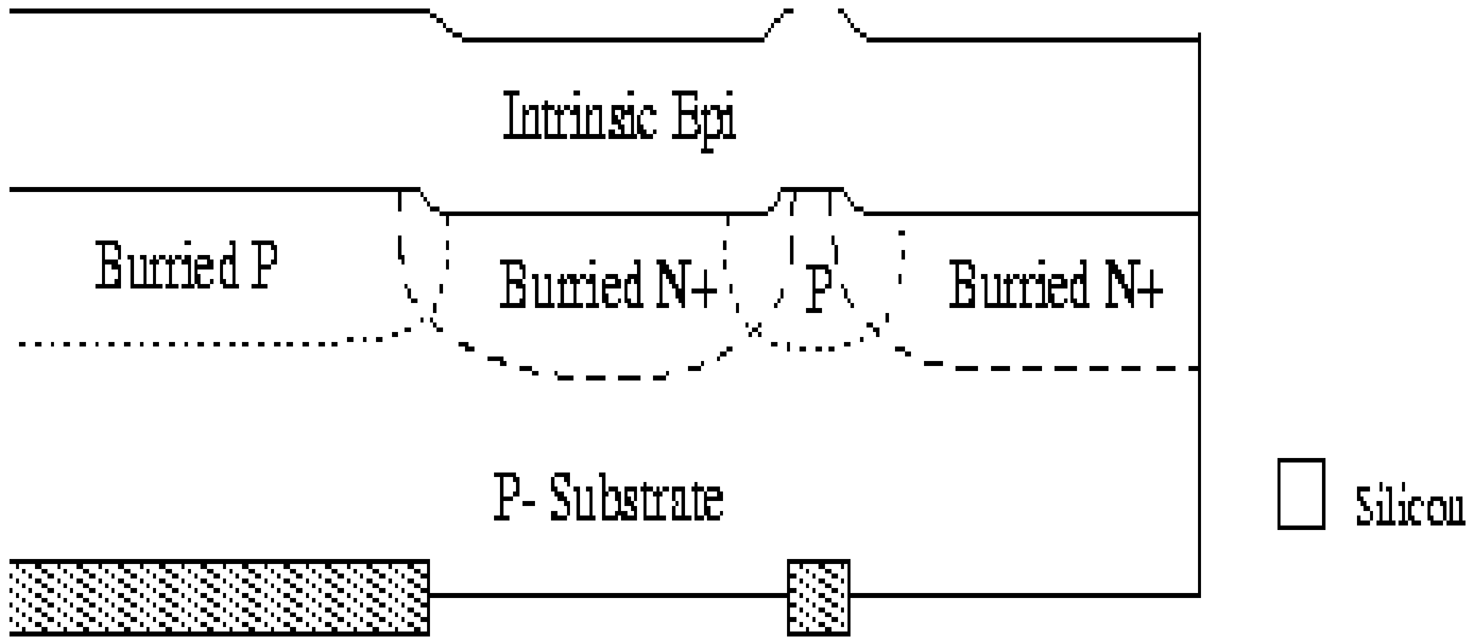


Figure 3: Device cross-section of BiCMOS process after growth of the EPI-layer.

BiCMOS Process Flow

After that a twin well process is used to fabricate the N-well of the PMOS and the collector of the NPN device. Again, the wafer is capped with a nitride layer which is opened at the N+ regions.

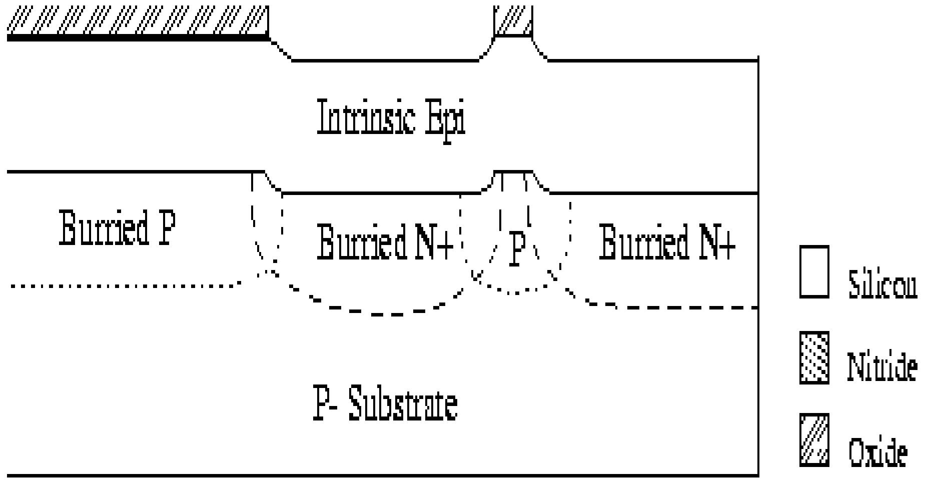


Figure 5.2-4: Device cross-section of BiCMOS process showing EPI-layer and masking for N-well implant.

BiCMOS Process Flow

After implanting the N-type dopant a thick oxide is grown and the nitride is stripped from the P+ regions. The subsequent P-well implant is self-aligned to the well edge (see Fig. 5).

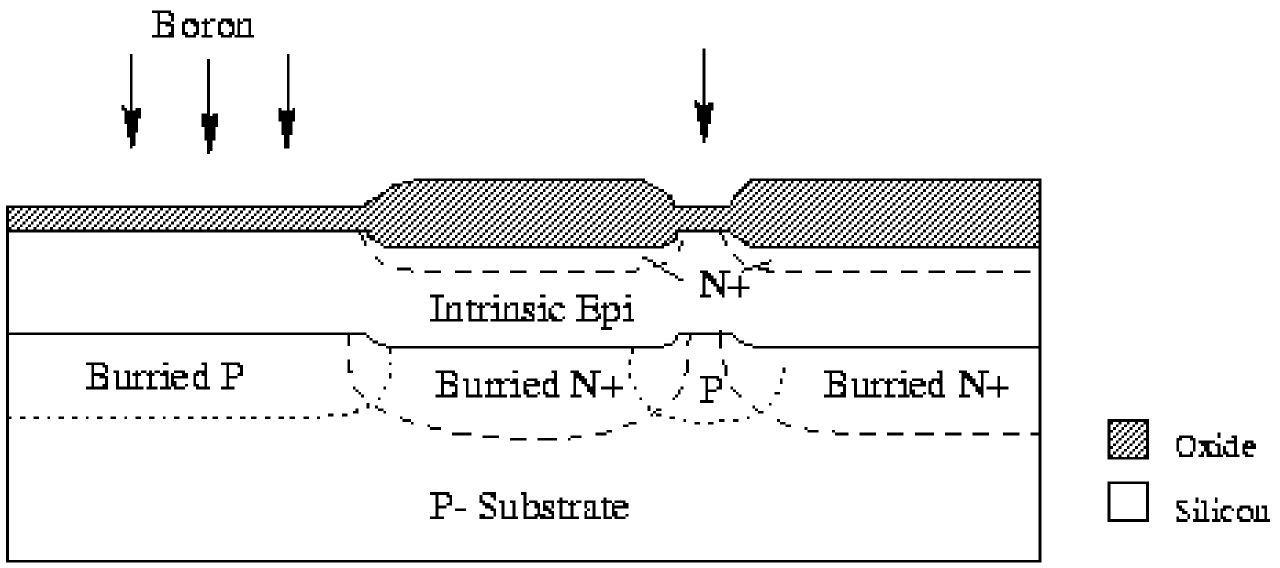


Figure 5: Device cross-section of BiCMOS process showing self-aligned *P*-well implant.

BiCMOS Process Flow

After the wells are fabricated the whole wafer is planarized and a pad oxide is grown. The oxide is capped with a thick nitride. After patterning the active regions of the device, an etch step is used to open up the field isolation regions. Prior to field oxidation, a blanket channel stop is implanted (see Fig. 6).

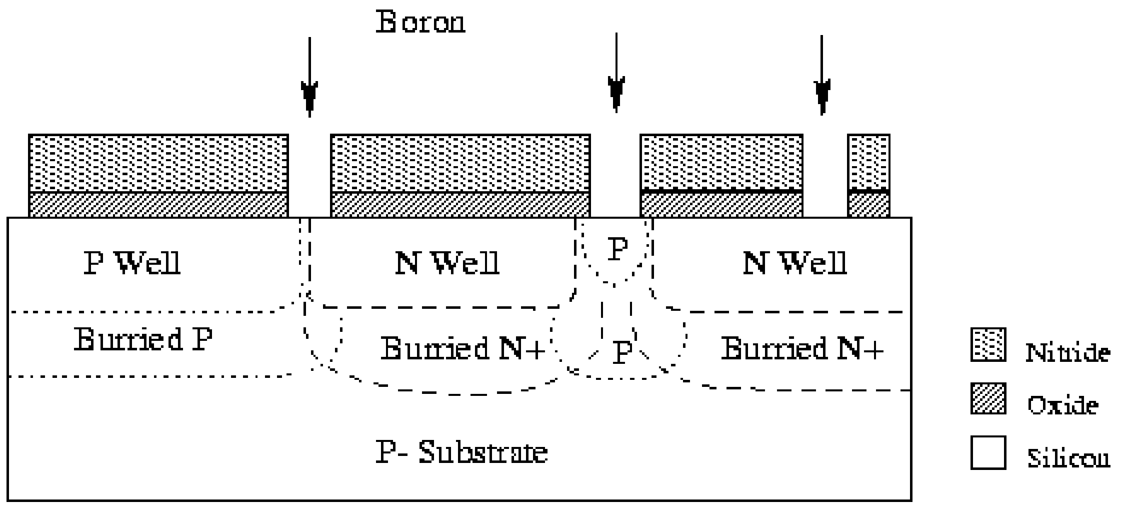


Figure 6: Device cross-section of BiCMOS process showing channel stop

BiCMOS Process Flow

Oxidation is used to fabricate a thick field oxide. After removal of the nitride masks from the active regions, phosphorus is implanted into the *N*-well of the collector region to implant the deep *N*+ subcollector (see Fig. 7). The PMOS and NMOS devices are protected by the photoresist.

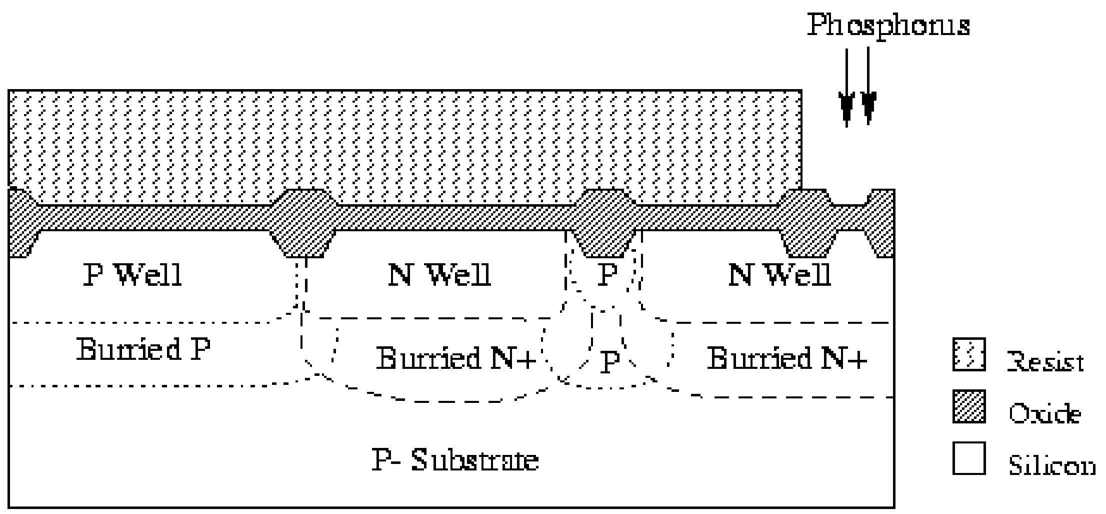


Figure 7: Device cross-section of BiCMOS process showing deep *N*+ subcollector implant.

BiCMOS Process Flow

For the fabrication of the intrinsic base for the bipolar device, the base region is opened and the base implant is performed. To ensure low base-emitter capacitance a thicker gate oxide is deposited after the base implant. The deposited oxide has to be removed from the non base regions by an etch step. The structure after the intrinsic base implant and prior to the base oxide deposition is shown in Figure 8

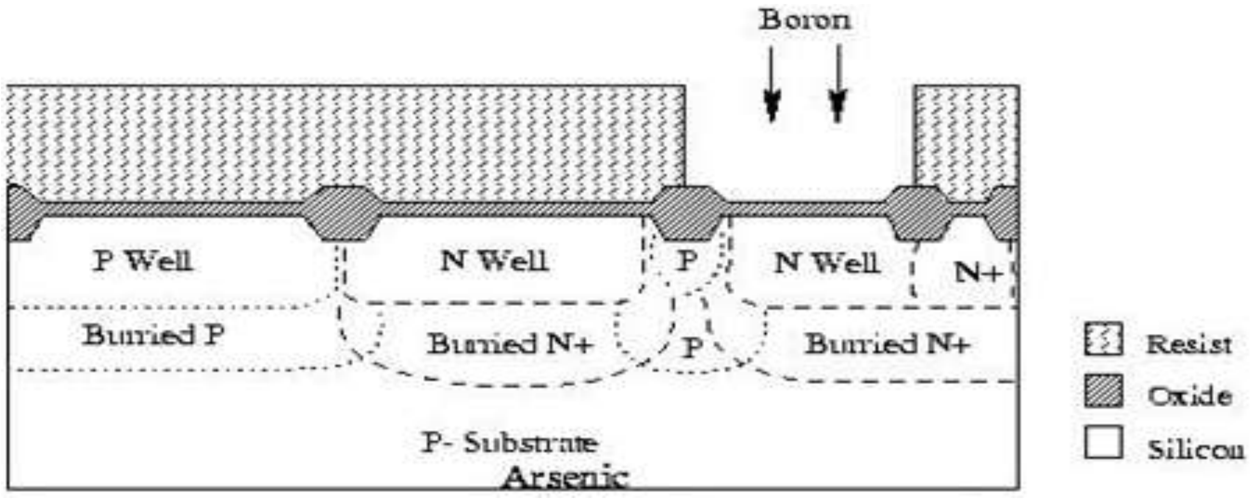


Figure 8: Device cross-section of BiCMOS process showing the intrinsic base implant.

BiCMOS Process Flow

We proceed with the resist strip and perform a pre-gate oxide etch to clean the oxide surface. The active emitter window is patterned and opened up with an etching process until the whole gate oxide is removed in the emitter region. Then a polysilicon layer is deposited, which forms the emitter contact as well as the gate polysilicon layers. This polysilicon layer is implanted with arsenic which will diffuse out from the polysilicon layer at the final source-drain anneal to form the emitter junction (see Fig. 9).

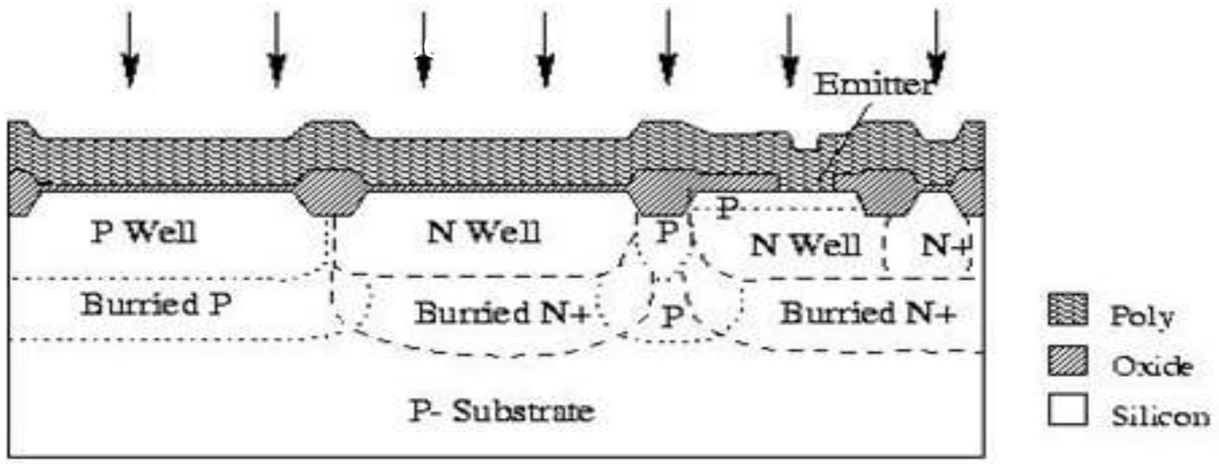


Figure 9: Device cross-section of BiCMOS process showing the fabrication of the polysilicon emitter.

BiCMOS Process Flow

The polysilicon layer is patterned to define the CMOS gates and the bipolar emitter. After emitter formation, all subsequent process steps are well known from CMOS technology. Phosphorus is implanted to form a shallow lightly doped drain(LDD) region for the NMOS device (see Fig 10). The subcollector is opened to collect additional N-type doping.

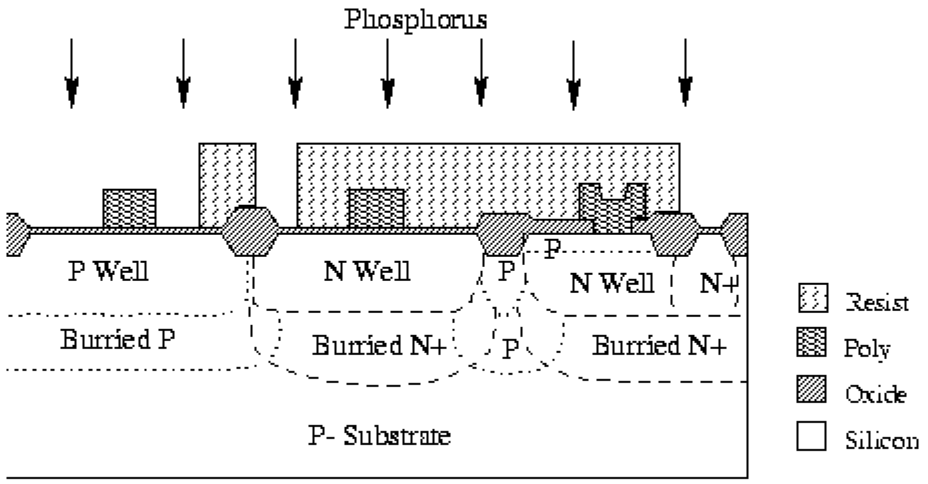


Figure 10: Device cross-section of BiCMOS process before the NMOS LDD doping is implanted.

Contd.

Finally, the fabrication of the active regions is finished by the source-drain anneal, which is optimized for outdiffusion conditions of the bipolar device. The final device structure including the active area doping is shown in Figure 13. Afterwards the structure is scheduled for a double-level interconnect process.

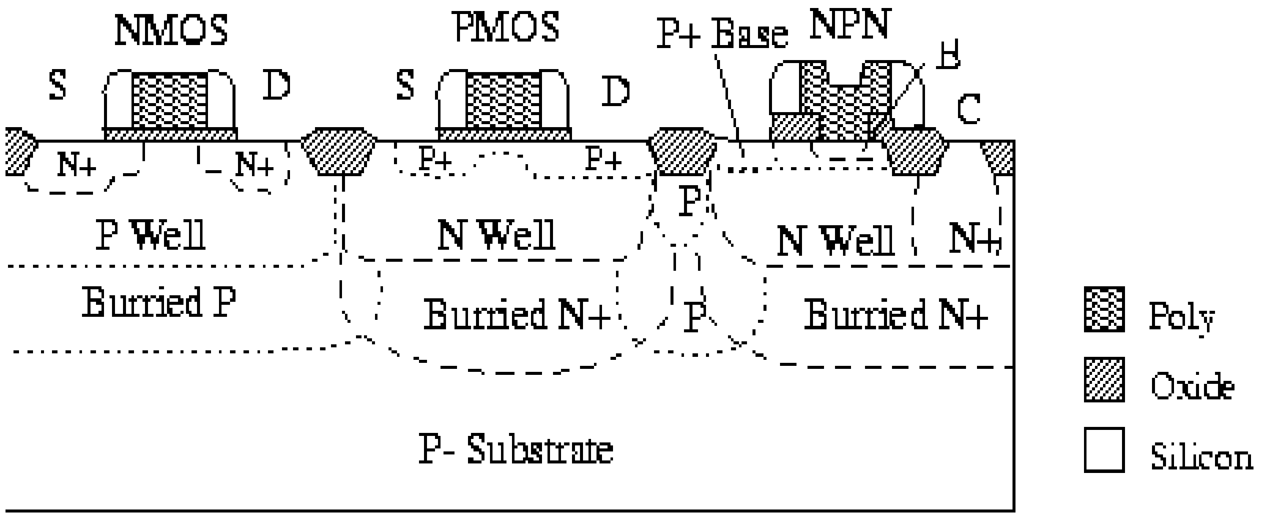


Figure 13: Device cross-section of BiCMOS process after fabrication of the active

Applications of BiCMOS Technology

BiCMOS has been widely used in many applications like

- Static Random Access Memory(SRAM) circuits
- Wireless Communication equipments like Transceivers, Amplifiers, Oscillators etc
- System-on-Chip Technology
- Personal Internet Access Devices
- Set-top boxes
- And many mixed signal applications

UNIT-II

VLSI DESIGN STYLES

Outline

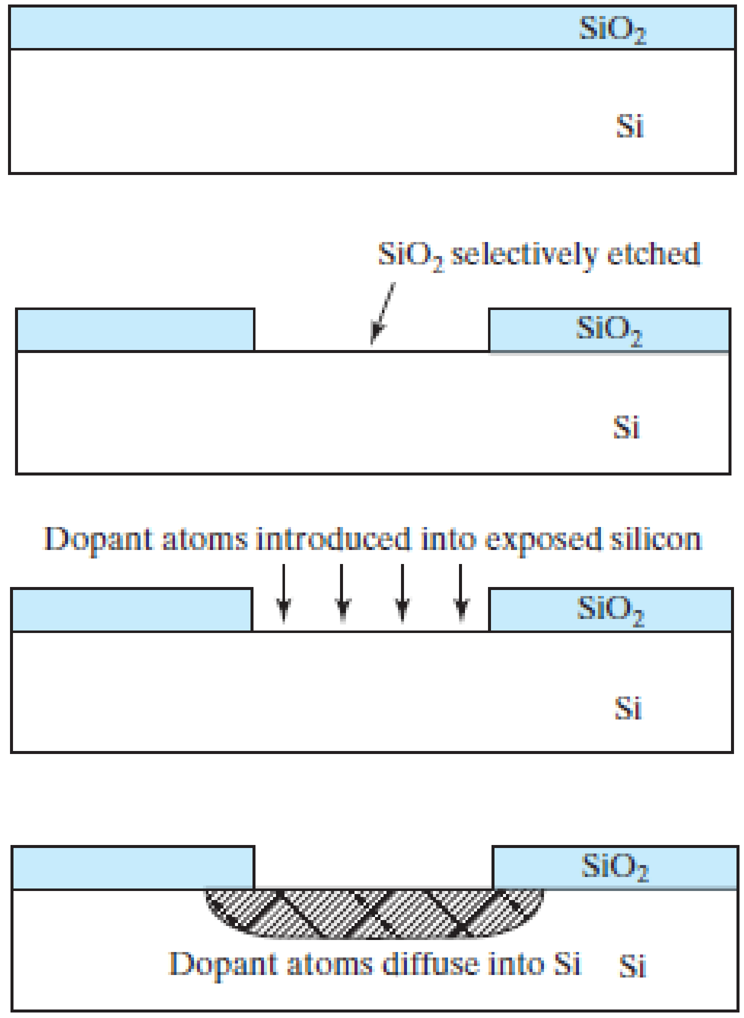
Introduction to Device Fabrication

Lithography & Etching

Ion Implantation

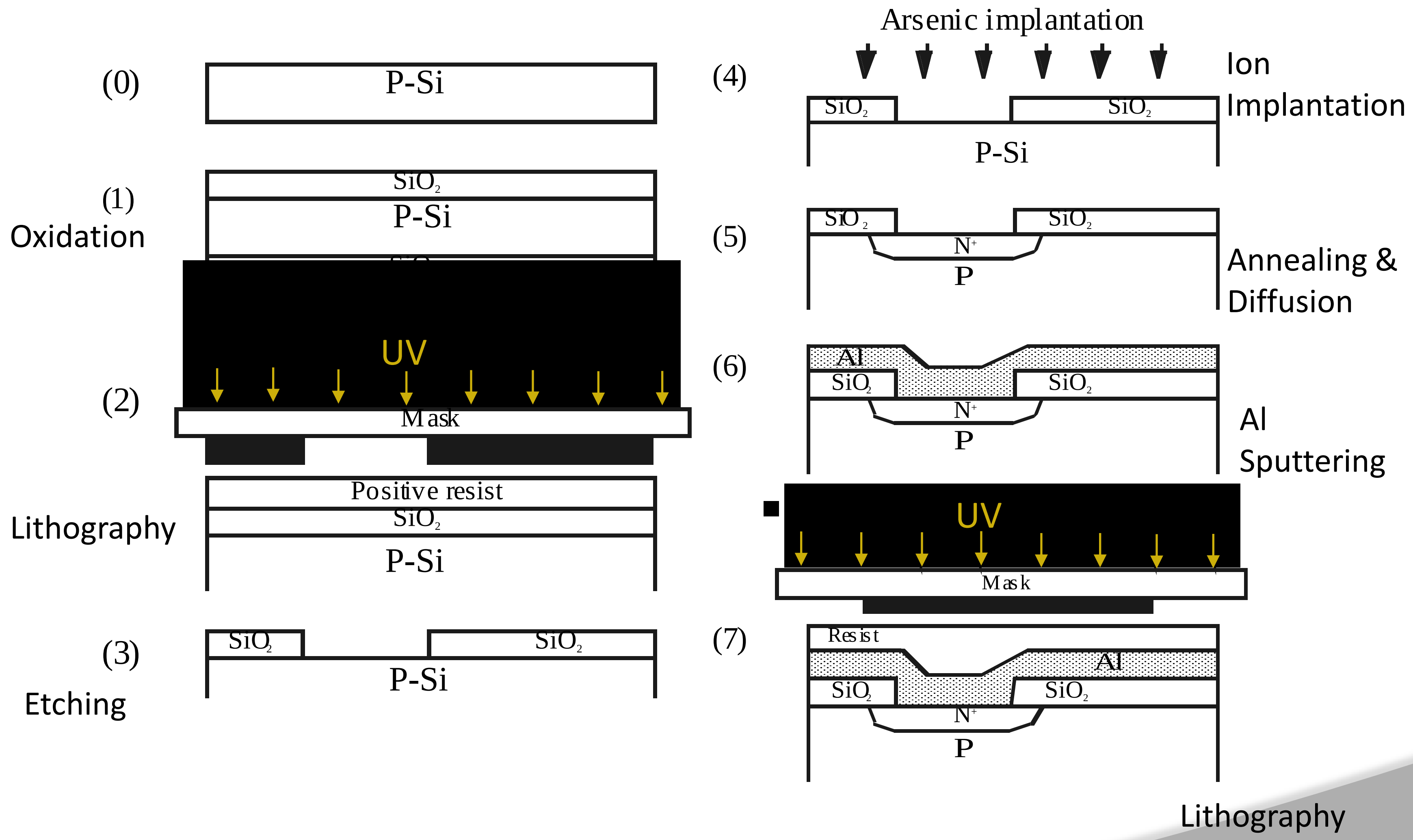
Oxidation

Annealing &Diffusion


SiO_2
Si
SiO_2 selectively etched
SiO_2
Si
Dopant atoms introduced into exposed silicon
SiO_2
Si
SiO_2
Dopant atoms diffuse into Si
Si

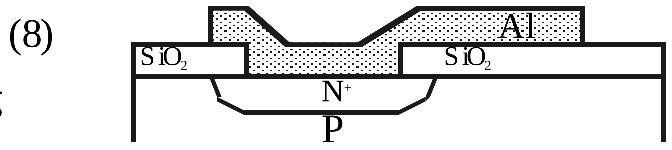
Modern Semiconductor Devices for Integrated Circuits (C. Hu)

Summary–A Device Fabrication Example

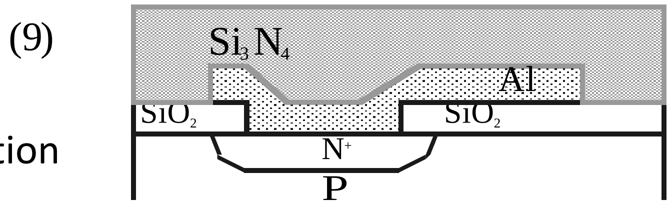


Summary–A Device Fabrication Example

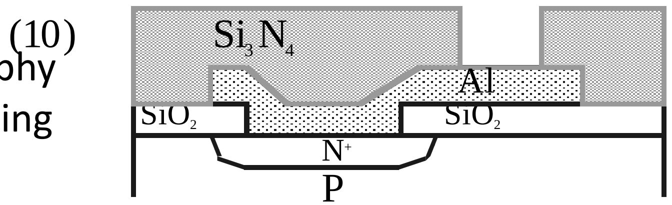
Metal etching



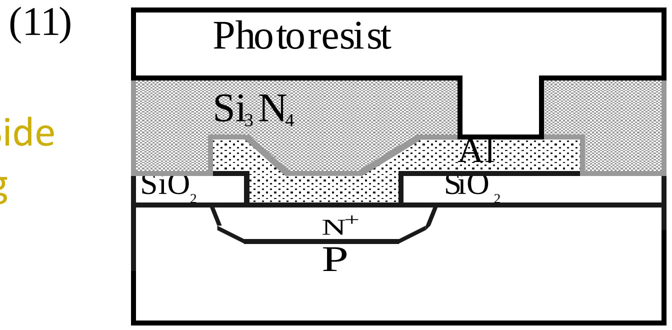
CVD nitride deposition

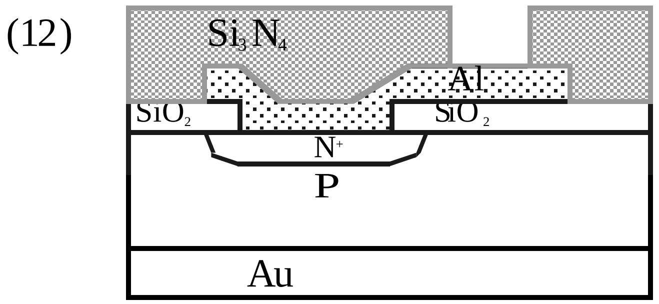


Lithography and etching

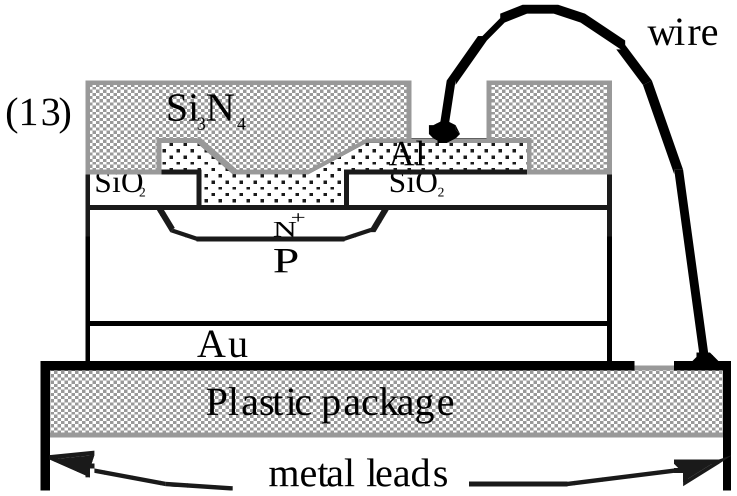


Back Side milling



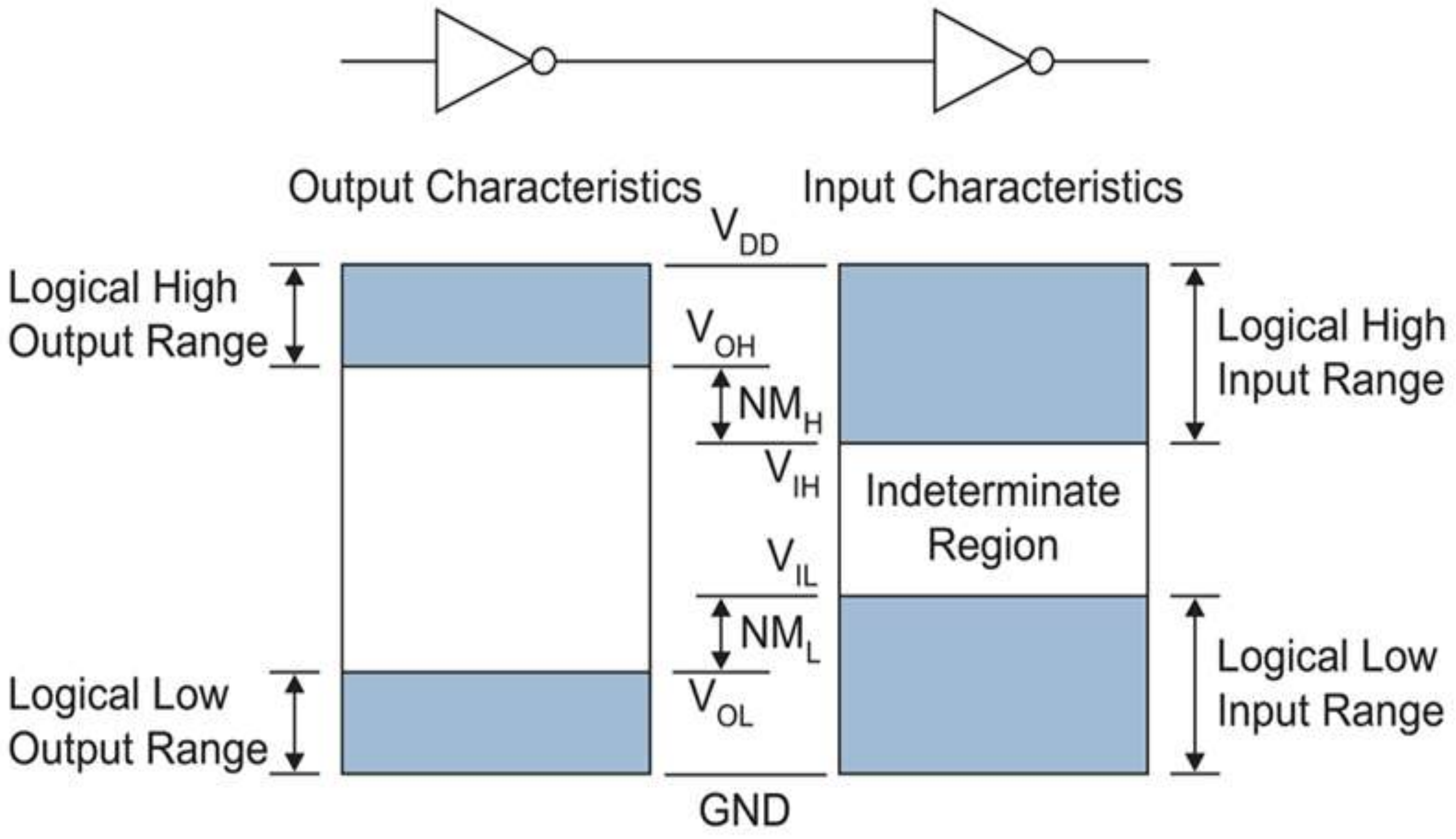


Back side metallization



Dicing, wire bonding, and packaging

Noise Margin



CMOS inverter noise margins

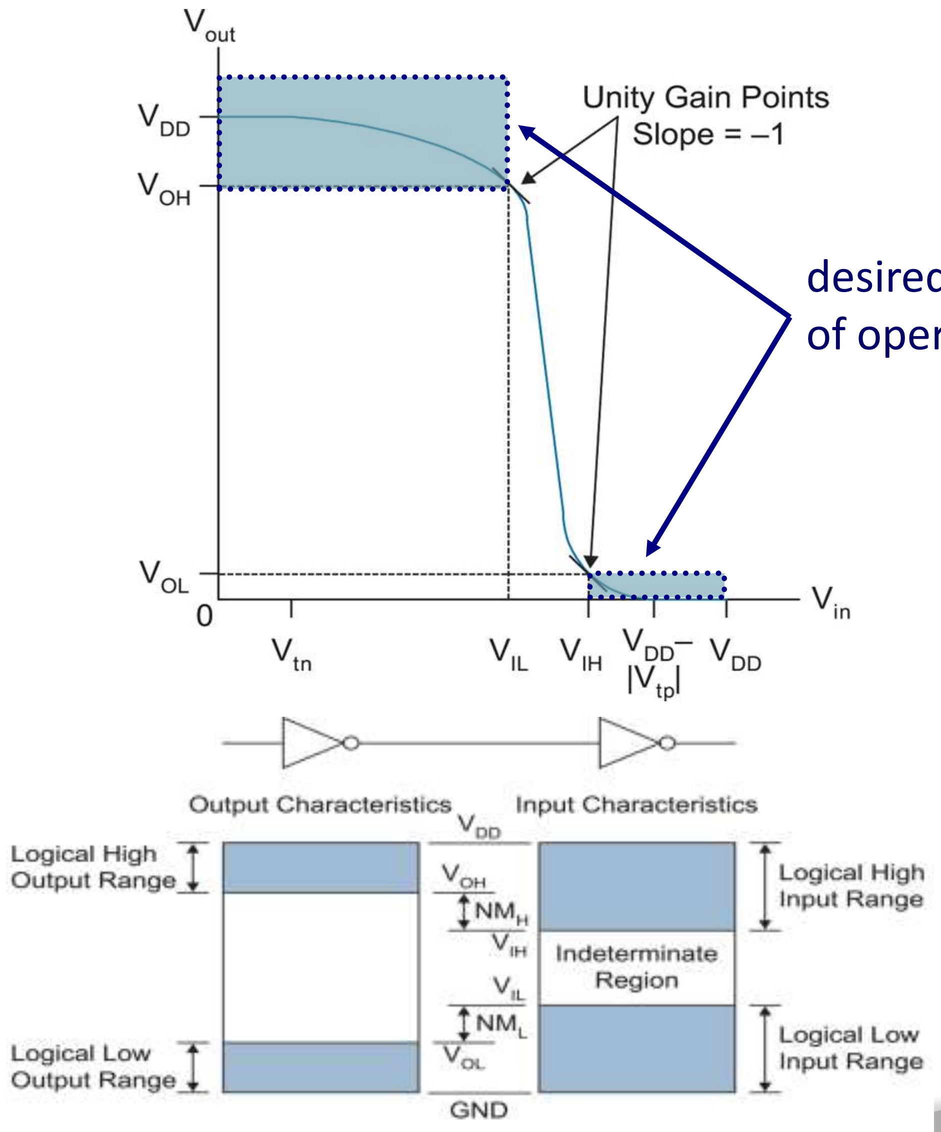
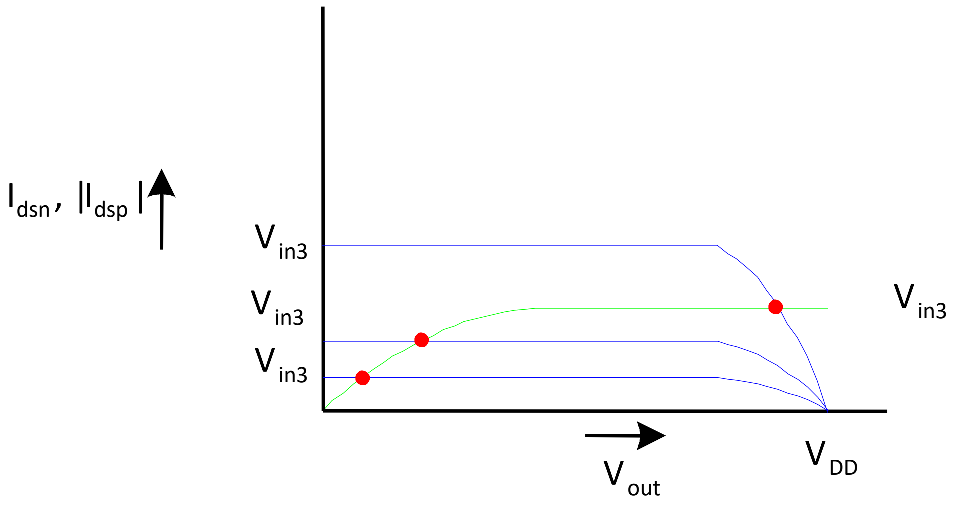
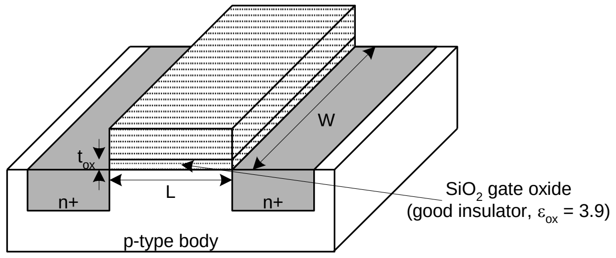


FIG 2.27 Noise margin definitions

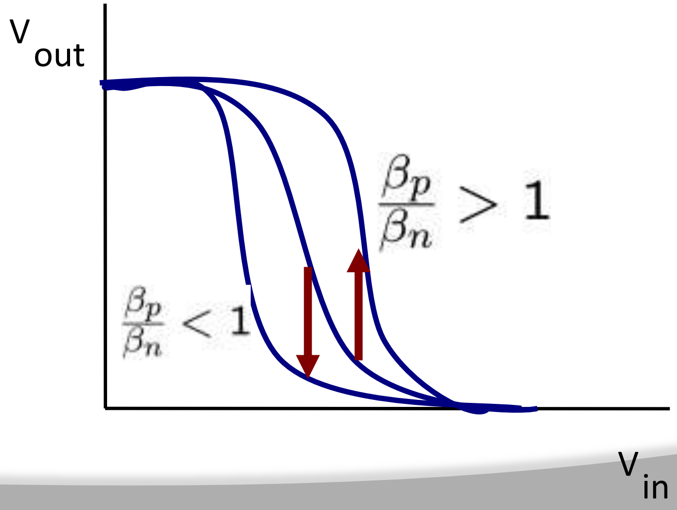
What is the impact of altering the PMOS width in comparison to the NMOS width on the DC char?





$$I_{DS} = \mu_p \frac{\varepsilon_{ox} W}{t_{ox} L} (V_{gs} - V_t)^2$$

If we increase (decrease) the width of PMOS compared to NMOS
$\rightarrow$ for the same input voltage, a higher (lower) output voltage is obtained



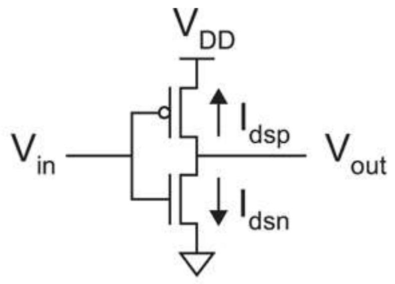


FIG 2.23 A CMOS inverter

Impact of *skewing* transistor sizes on inverter noise margins

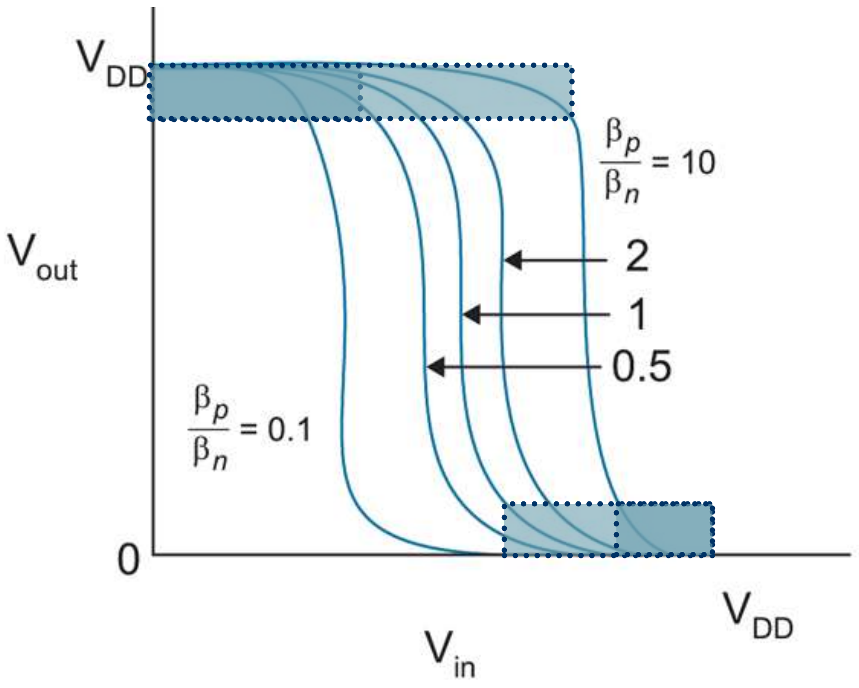
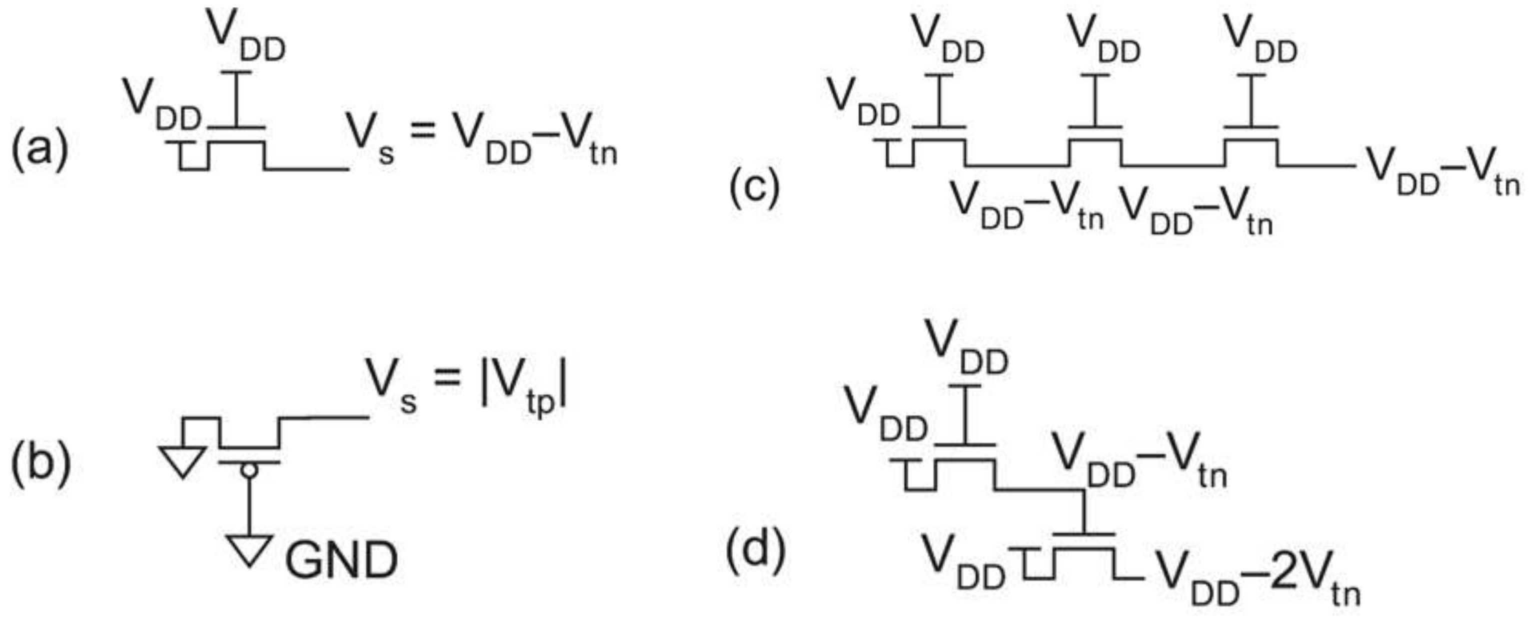


FIG 2.26 Transfer characteristics of skewed inverters

- Increasing (decreasing) PMOS width to NMOS width increases (decreases) the low noise margin and decreases (increases) the high noise margin

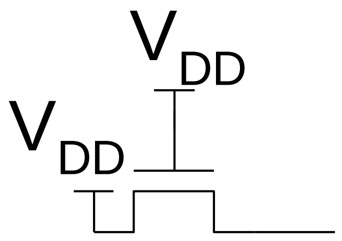
Pass transistor DC characteristics



- As the source can rise to within a threshold voltage of the gate, the output of several transistors in series is no more degraded than that of a single transistor

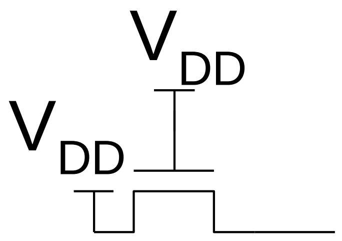
Pass Transistors

- We have assumed source is grounded
- What if source > 0?
  - e.g. pass transistor passing V_{DD}

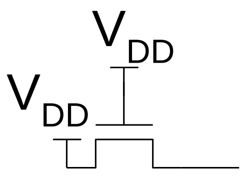


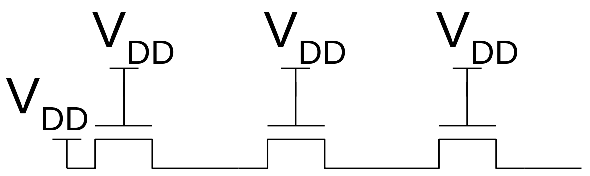
Pass Transistors

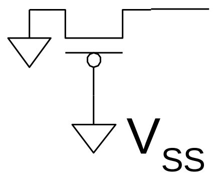
- We have assumed source is grounded
- What if source > 0?
  - e.g. pass transistor passing V_{DD}
- $V_g = V_{DD}$
  - If $V_s > V_{DD}-V_t$, $V_{gs} < V_t$
  - Hence transistor would turn itself off
- nMOS pass transistors pull no higher than $V_{DD}-V_{tn}$
  - Called a degraded "1"
  - Approach degraded value slowly (low I_{ds})
- pMOS pass transistors pull no lower than V_{tp}

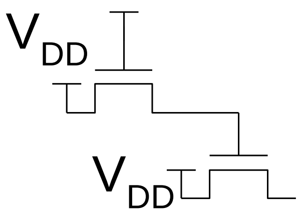


Pass Transistor Ckts

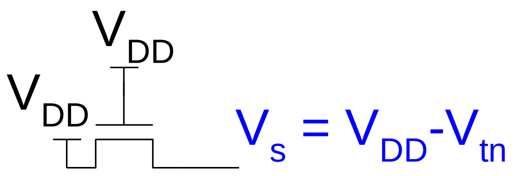


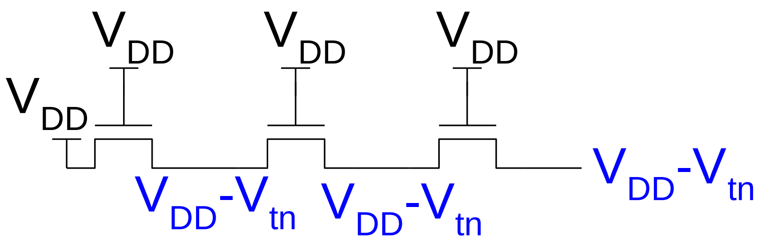


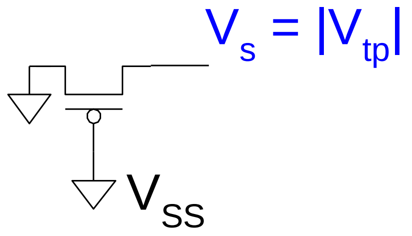


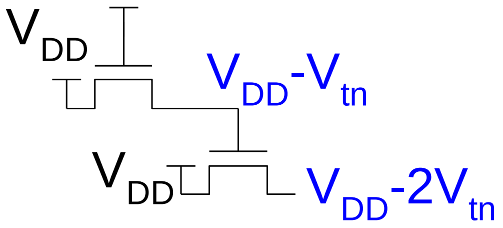


Pass Transistor Ckts







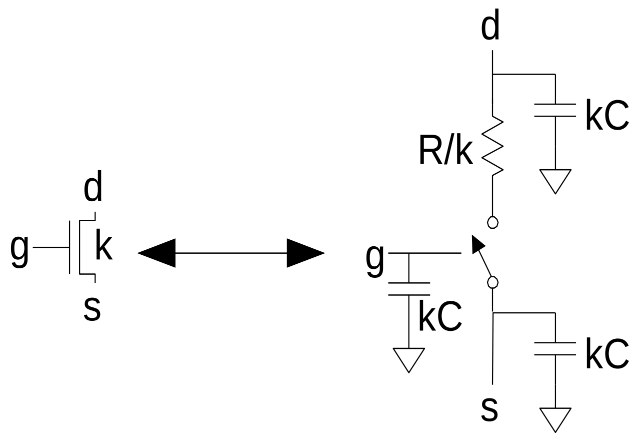


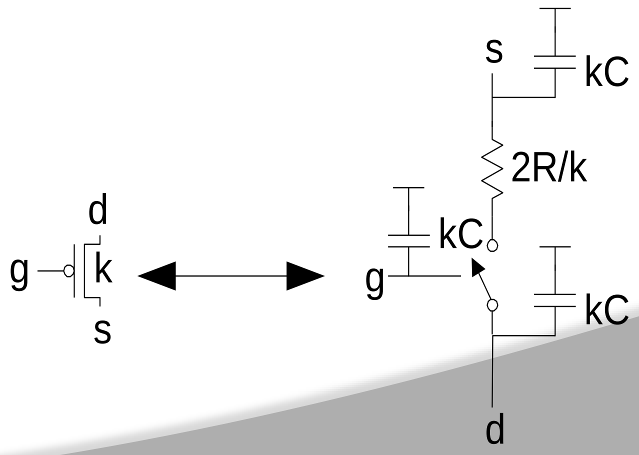
Effective Resistance

- Shockley models have limited value
  - Not accurate enough for modern transistors
  - Too complicated for much hand analysis
- Simplification: treat transistor as resistor
  - Replace $I_{ds}(V_{ds}, V_{gs})$ with effective resistance R
    - $I_{ds} = V_{ds}/R$
  - R averaged across switching of digital gate
- Too inaccurate to predict current at any given time
  - But good enough to predict RC delay

RC Delay Model

- Use equivalent circuits for MOS transistors
  - Ideal switch + capacitance and ON resistance
  - Unit nMOS has resistance R, capacitance C
  - Unit pMOS has resistance 2R, capacitance C
- Capacitance proportional to width
- Resistance inversely proportional to width



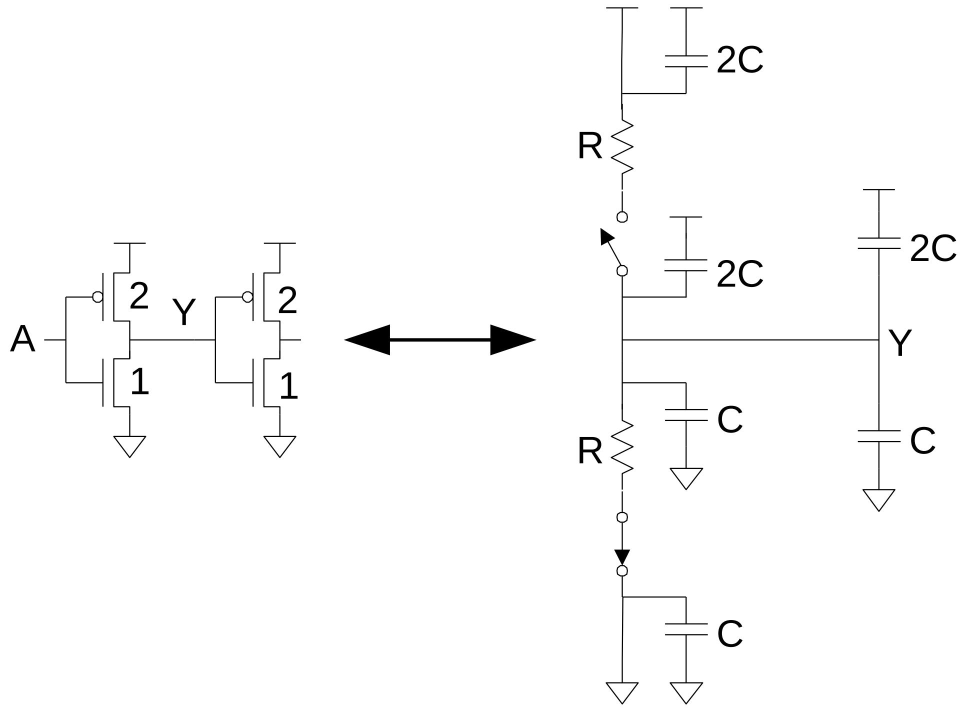


RC Values

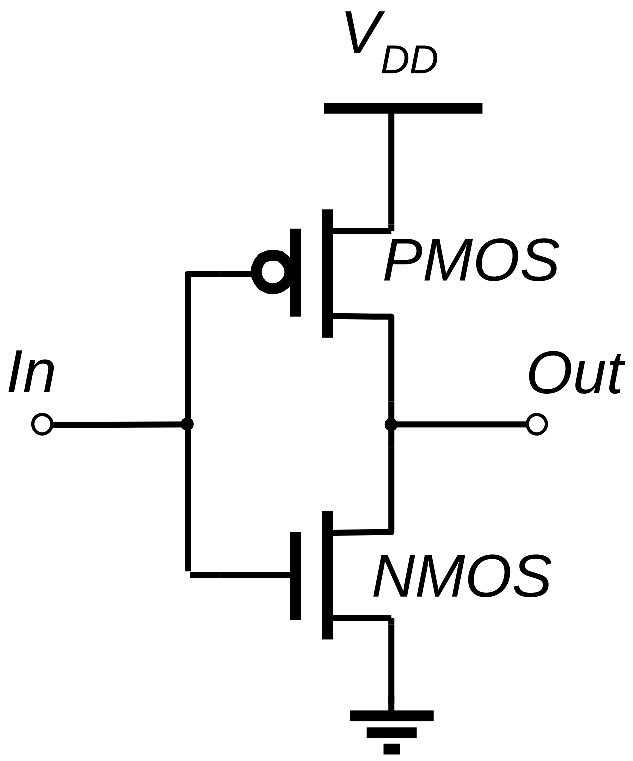
- Capacitance
  - $C = C_g = C_s = C_d$ = 2 fF/mm of gate width
  - Values similar across many processes
- Resistance
  - R $\approx$ 6 KW*mm in 0.6um process
  - Improves with shorter channel lengths
- Unit transistors
  - May refer to minimum contacted device (4/2 l)
  - Or maybe 1 mm wide device
  - Doesn’t matter as long as you are consistent

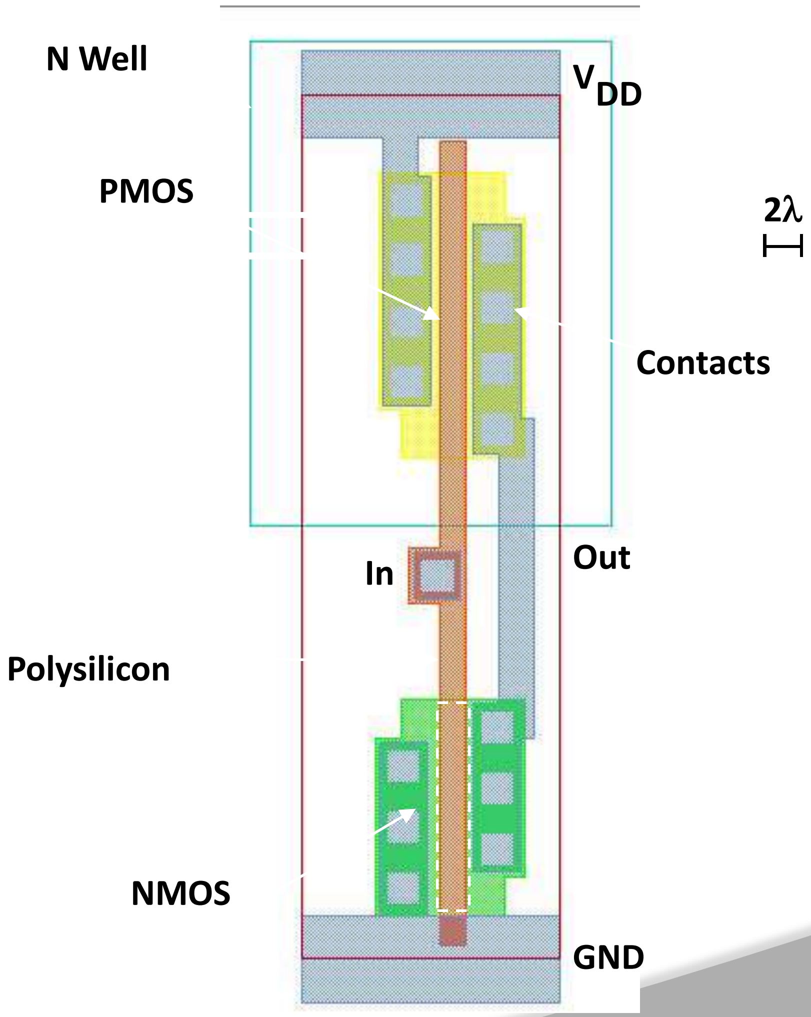
Inverter Delay Estimate

- **Estimate the delay of a fanout-of-1 inverter**

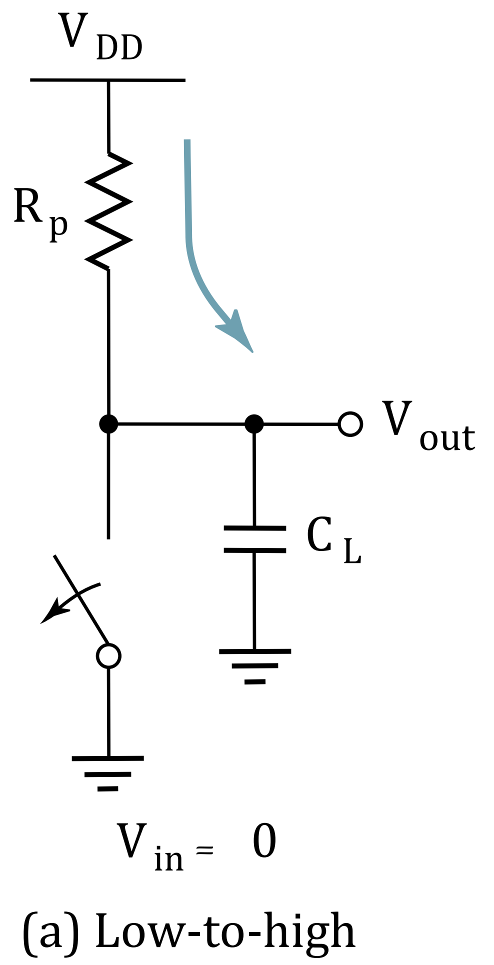


CMOS Inverter

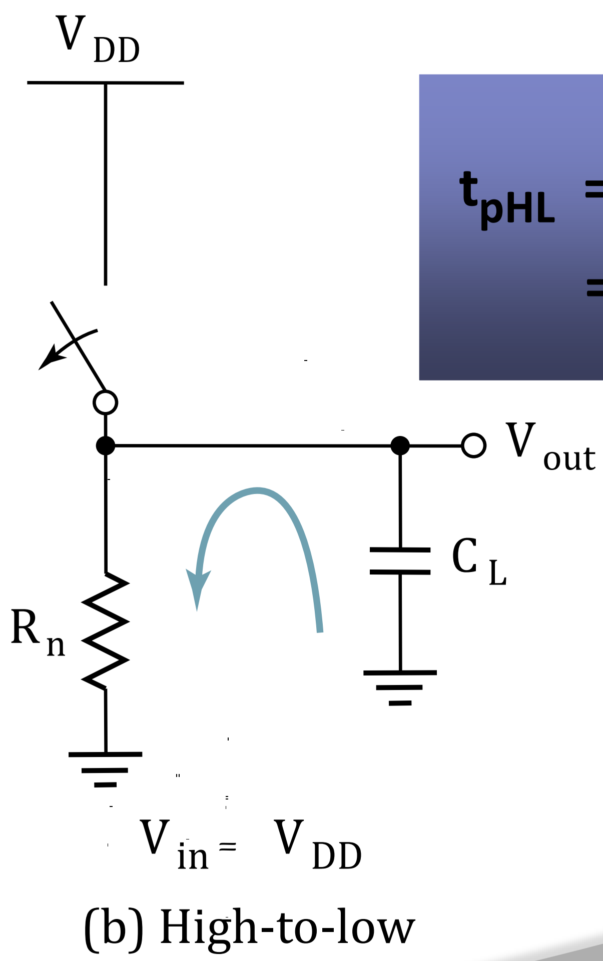




CMOS Inverter: Transient Response



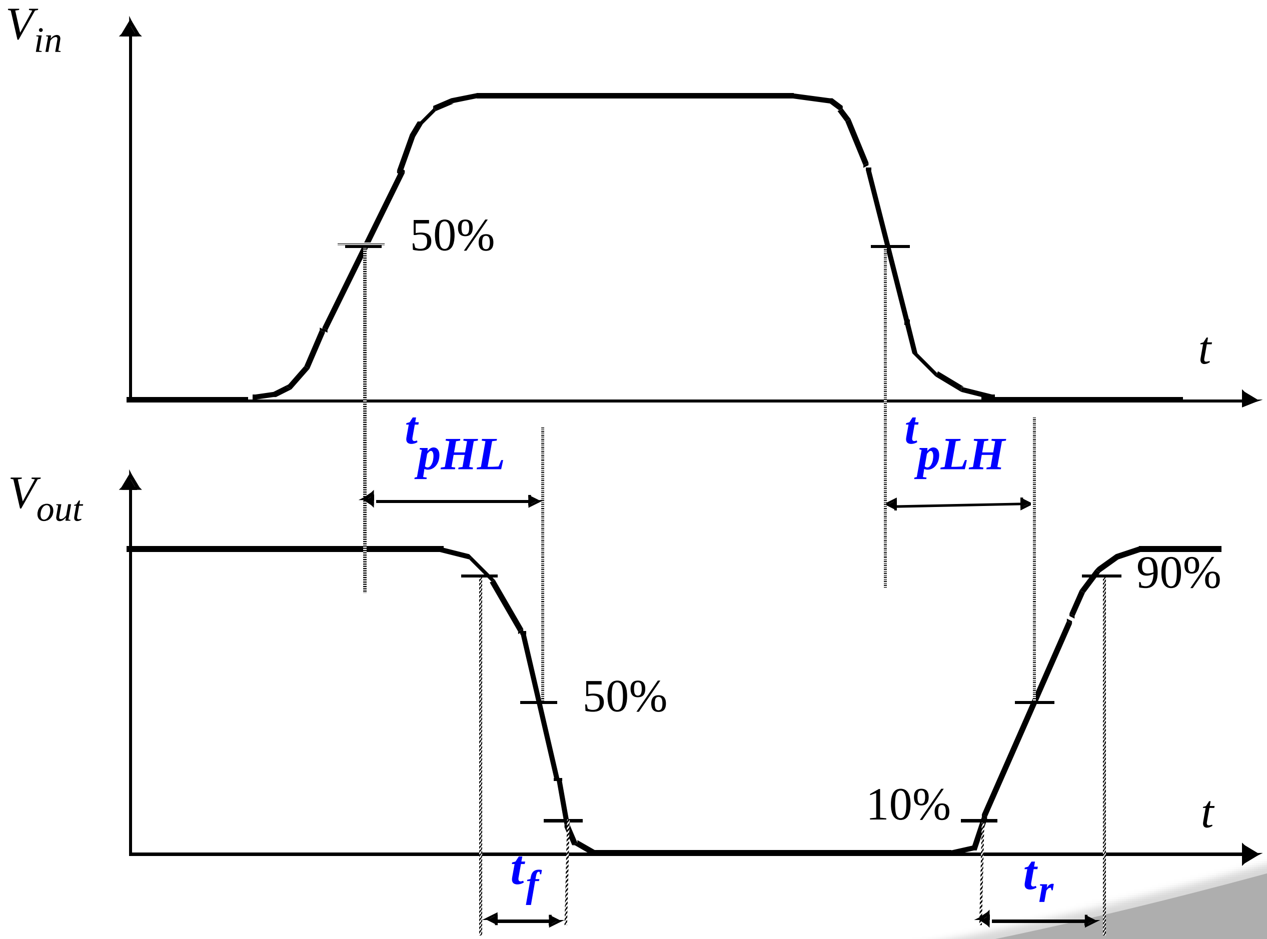
(a) Low-to-high



(b) High-to-low

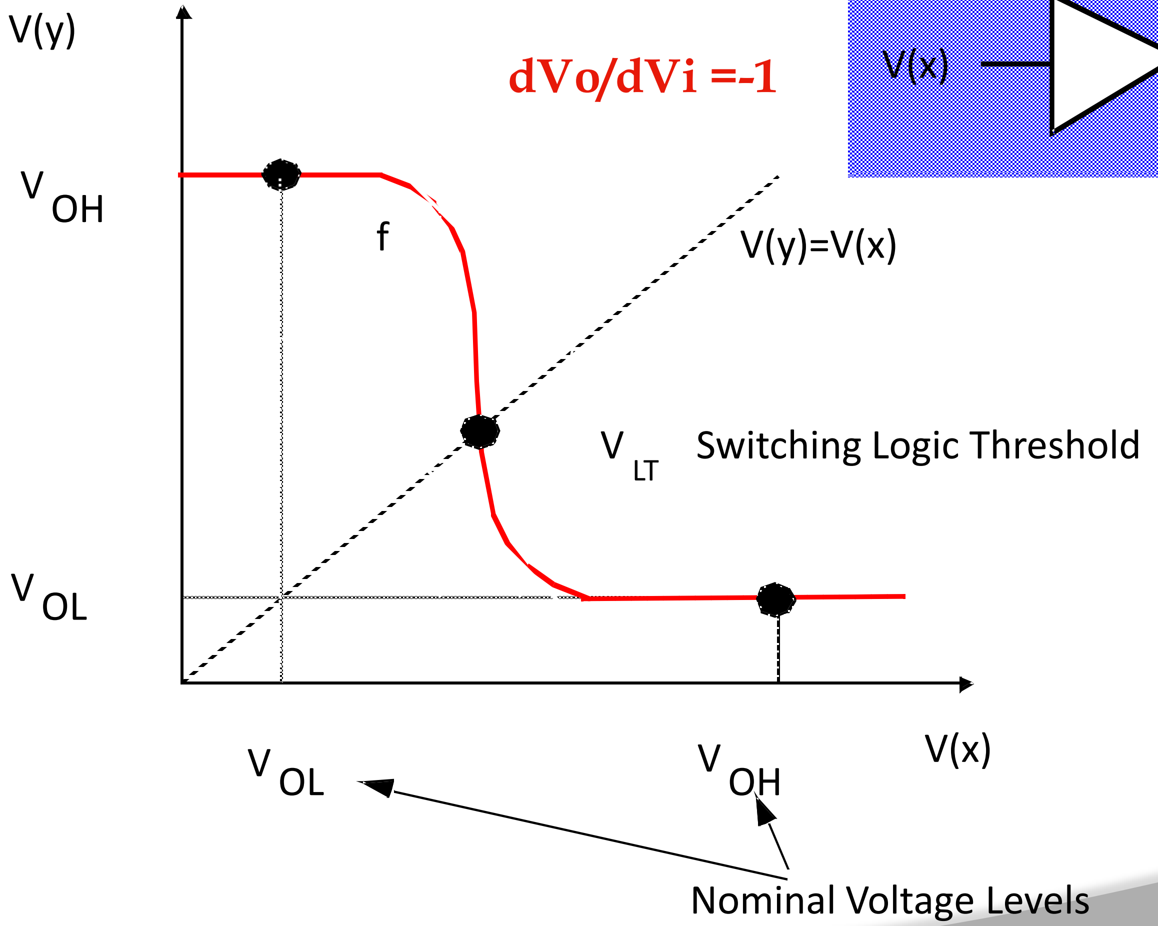
$$t_{pHL} = f(R_{on} . C_L)$$
$$= 0.69\, R_{on} C_L$$

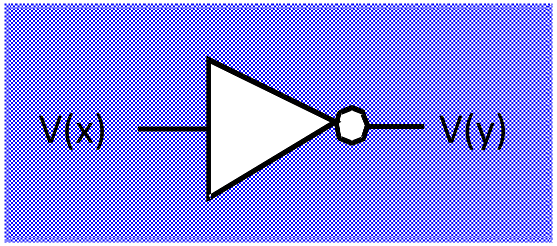
Delay Definitions



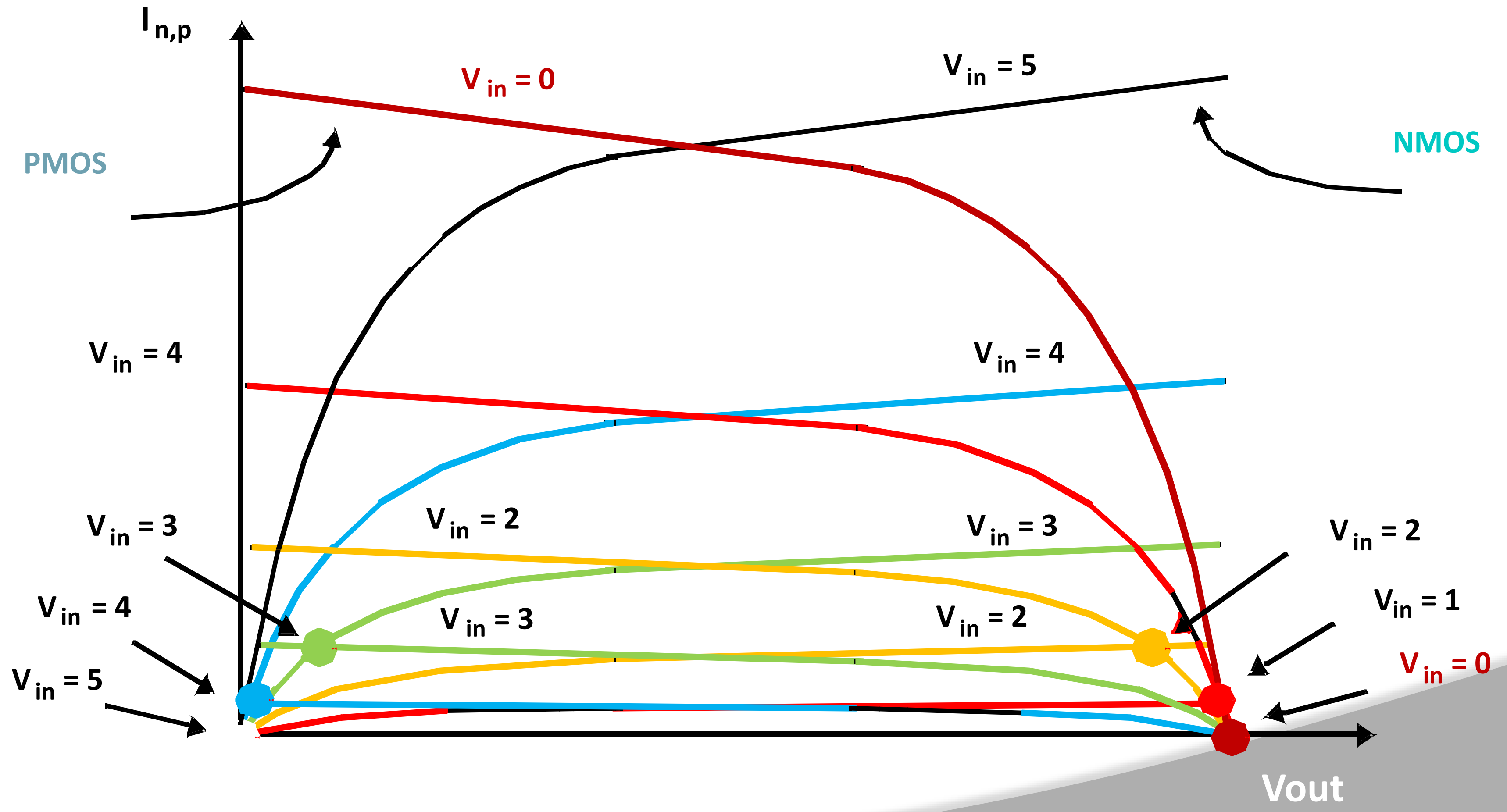
DC Operation:

Voltage Transfer Characteristic

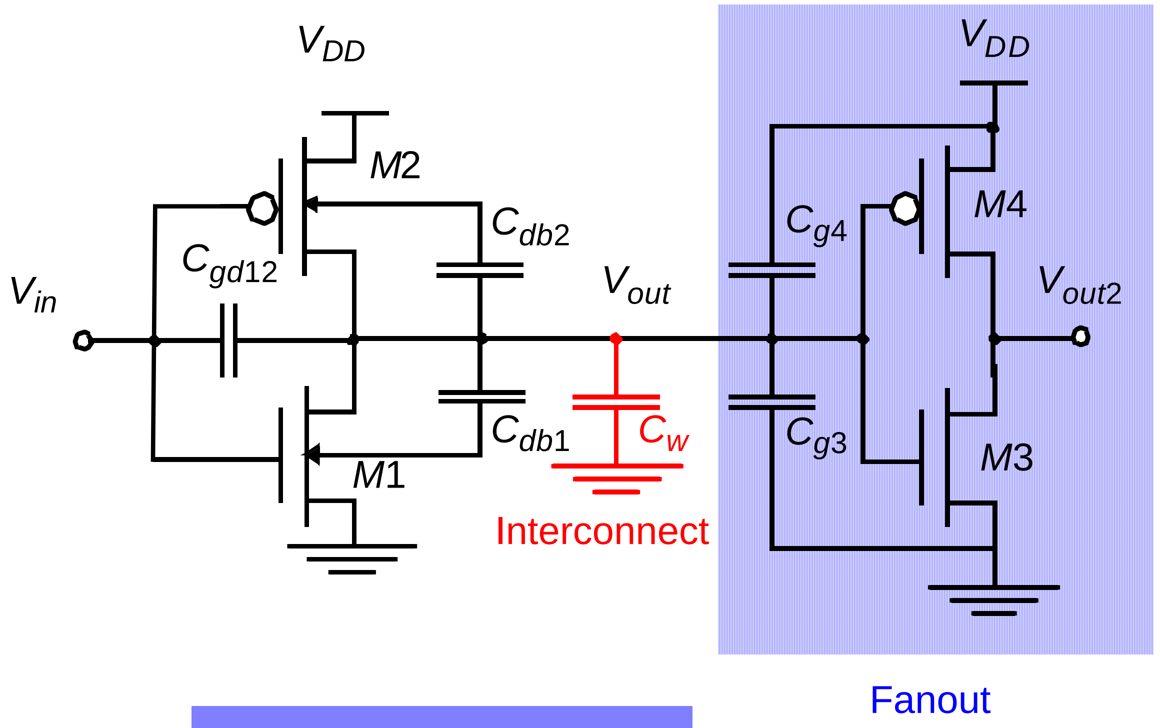




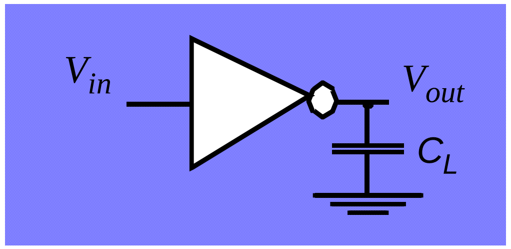
CMOS Inverter Load Characteristics



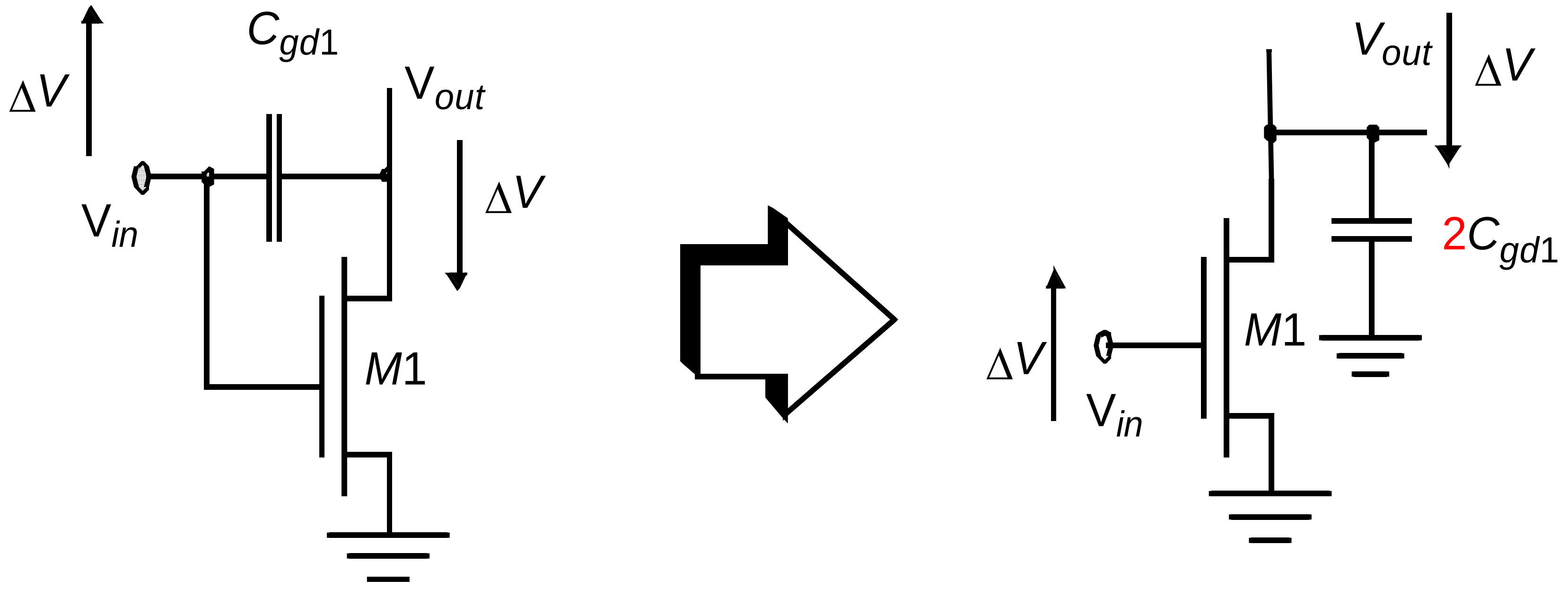
Computing the Capacitances



Simplified Model

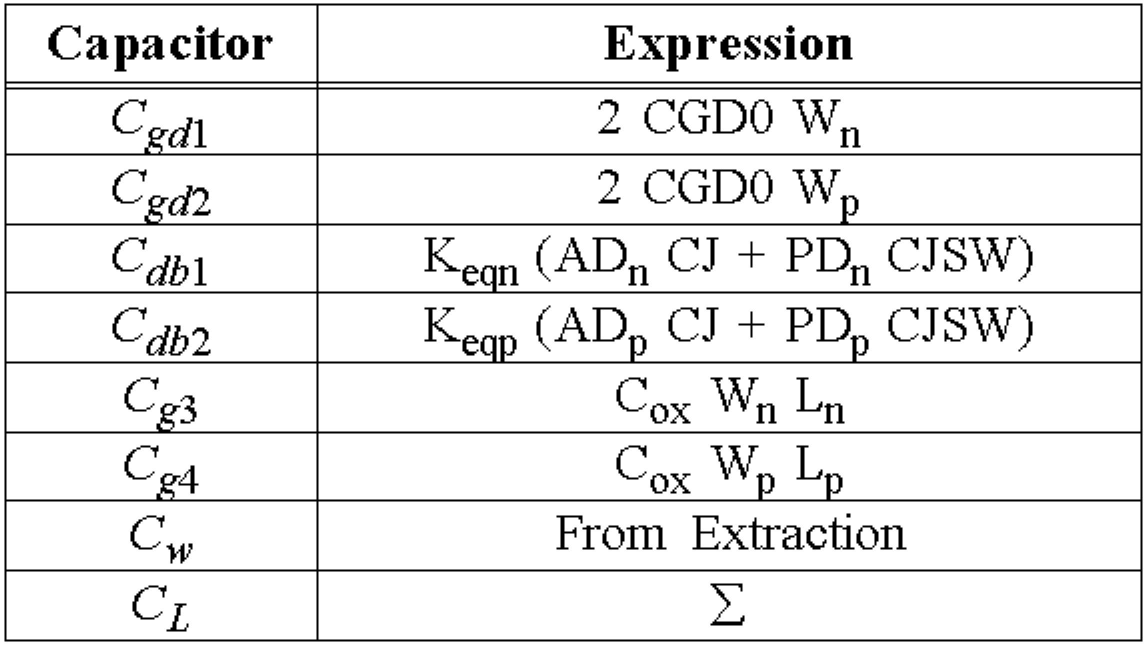


The Miller Effect



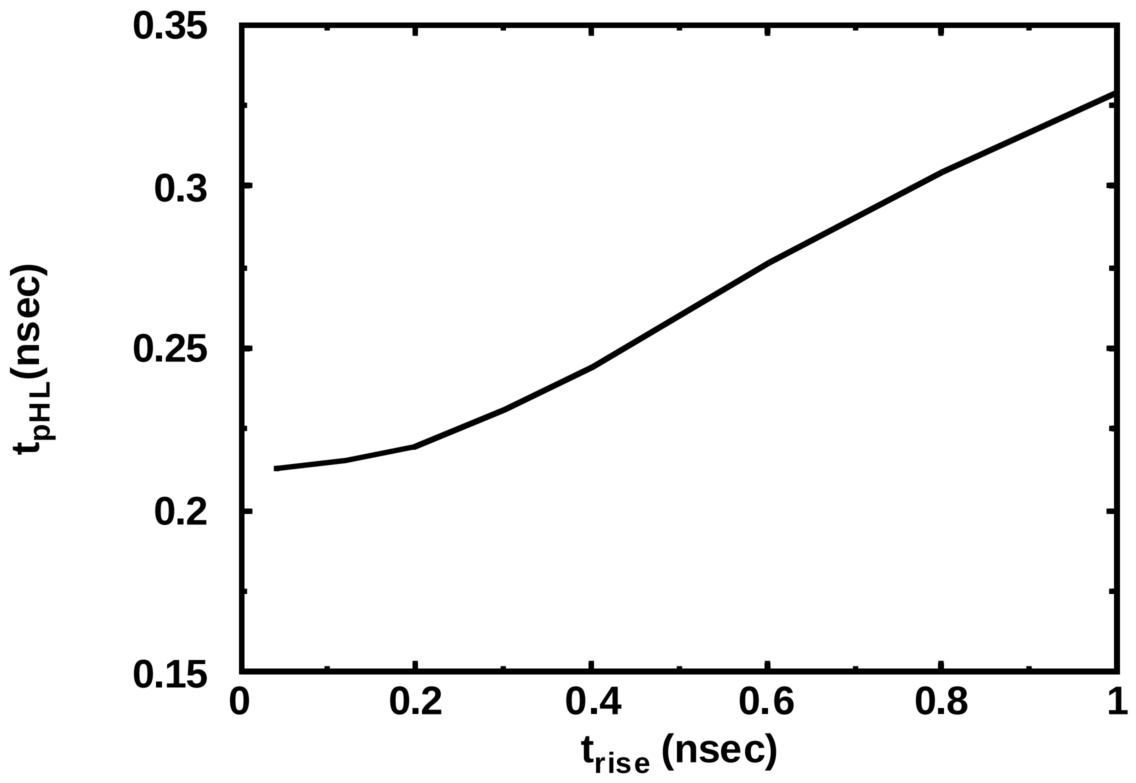
“A capacitor experiencing identical but opposite voltage swings at both its terminals can be replaced by a capacitor to ground, whose value is two times the original value.”

Computing the Capacitances



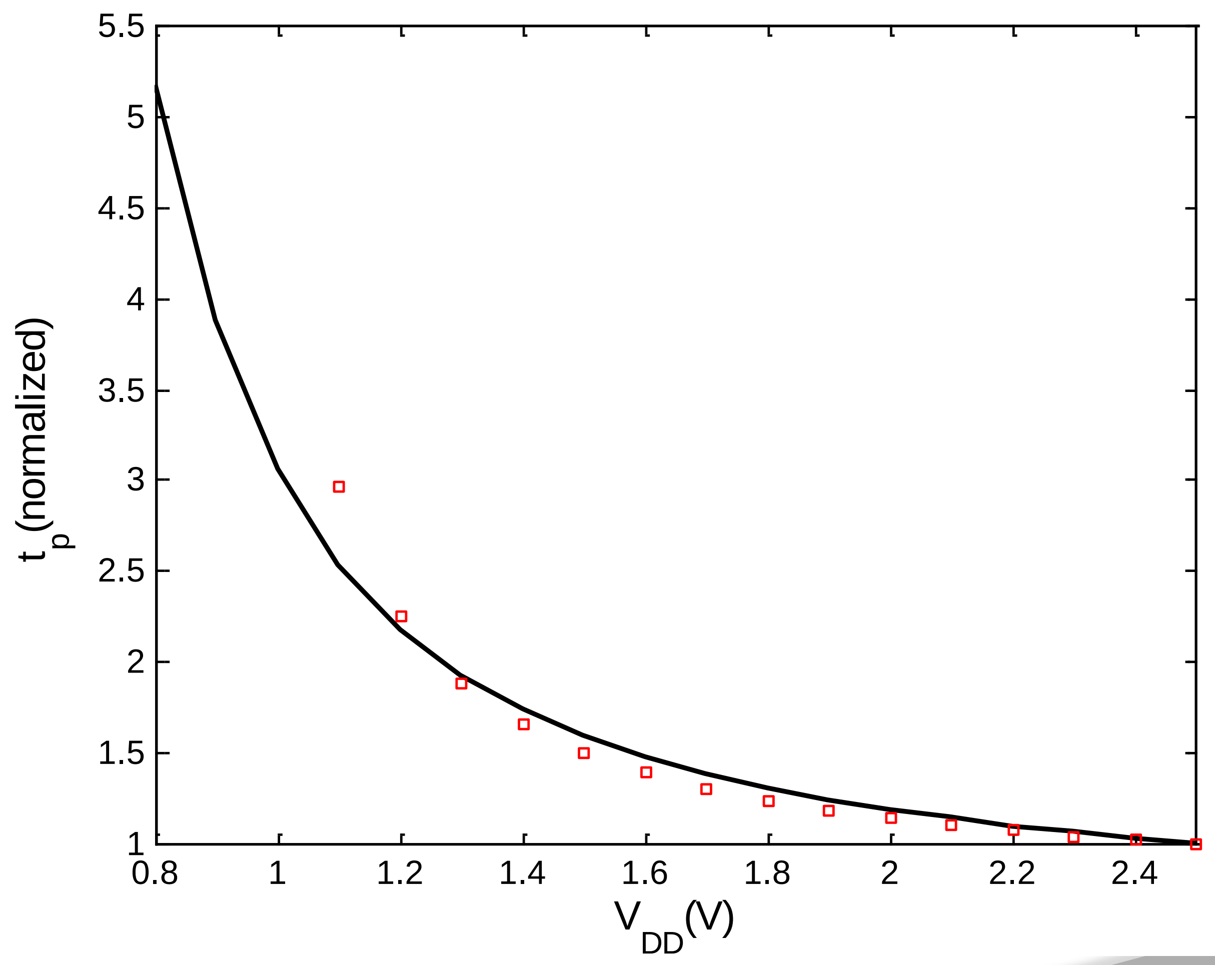
Capacitor	Expression
C_{gd1}	$2\ \mathrm{CGD0}\ \mathrm{W_n}$
C_{gd2}	$2\ \mathrm{CGD0}\ \mathrm{W_p}$
C_{db1}	$\mathrm{K_{eqn}}\ (\mathrm{AD_n}\ \mathrm{CJ} + \mathrm{PD_n}\ \mathrm{CJSW})$
C_{db2}	$\mathrm{K_{eqp}}\ (\mathrm{AD_p}\ \mathrm{CJ} + \mathrm{PD_p}\ \mathrm{CJSW})$
C_{g3}	$\mathrm{C_{ox}}\ \mathrm{W_n}\ \mathrm{L_n}$
C_{g4}	$\mathrm{C_{ox}}\ \mathrm{W_p}\ \mathrm{L_p}$
C_w	From Extraction
C_L	Σ

Impact of Rise Time on Delay

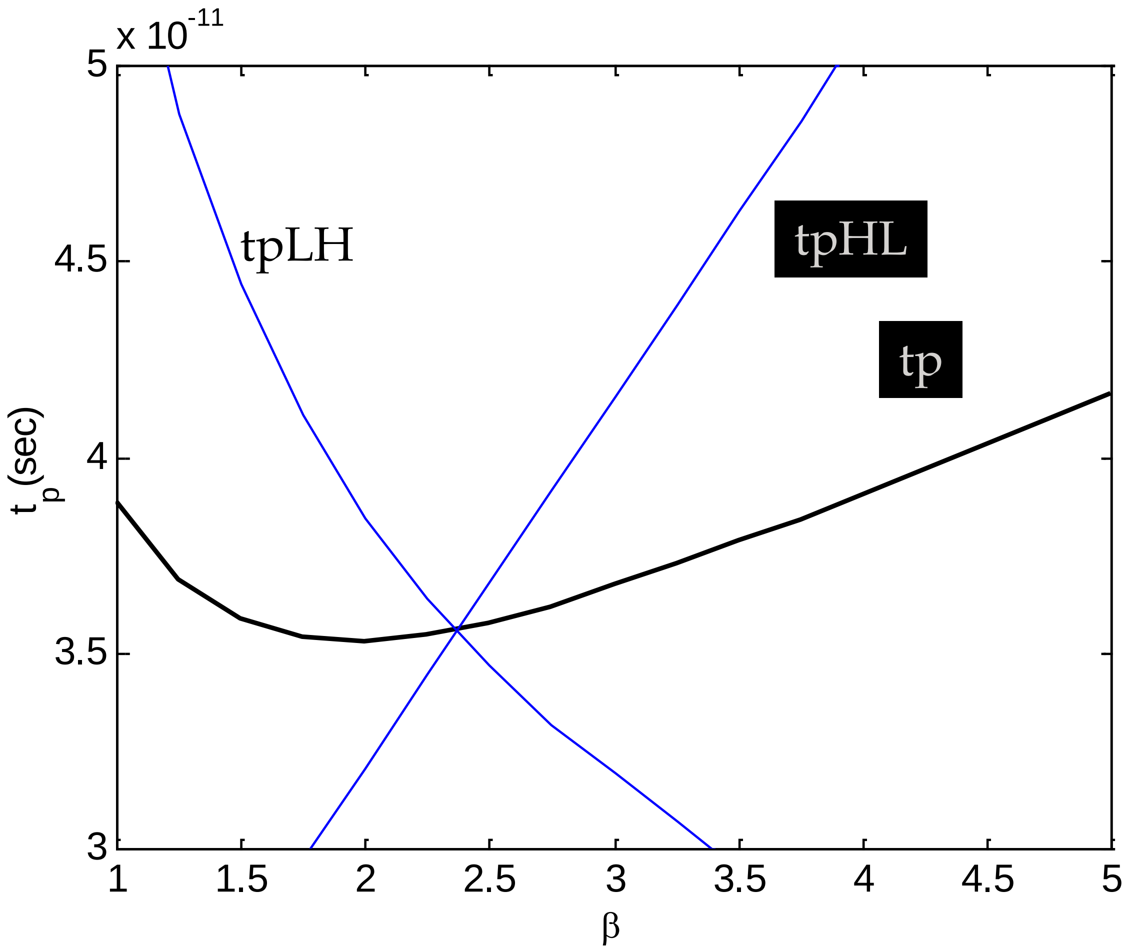


$$t_{pHL} = \sqrt{t_{pHL(step)}^2 + (t_r/2)^2}$$

Delay as a function of V_{DD}



NMOS/PMOS ratio

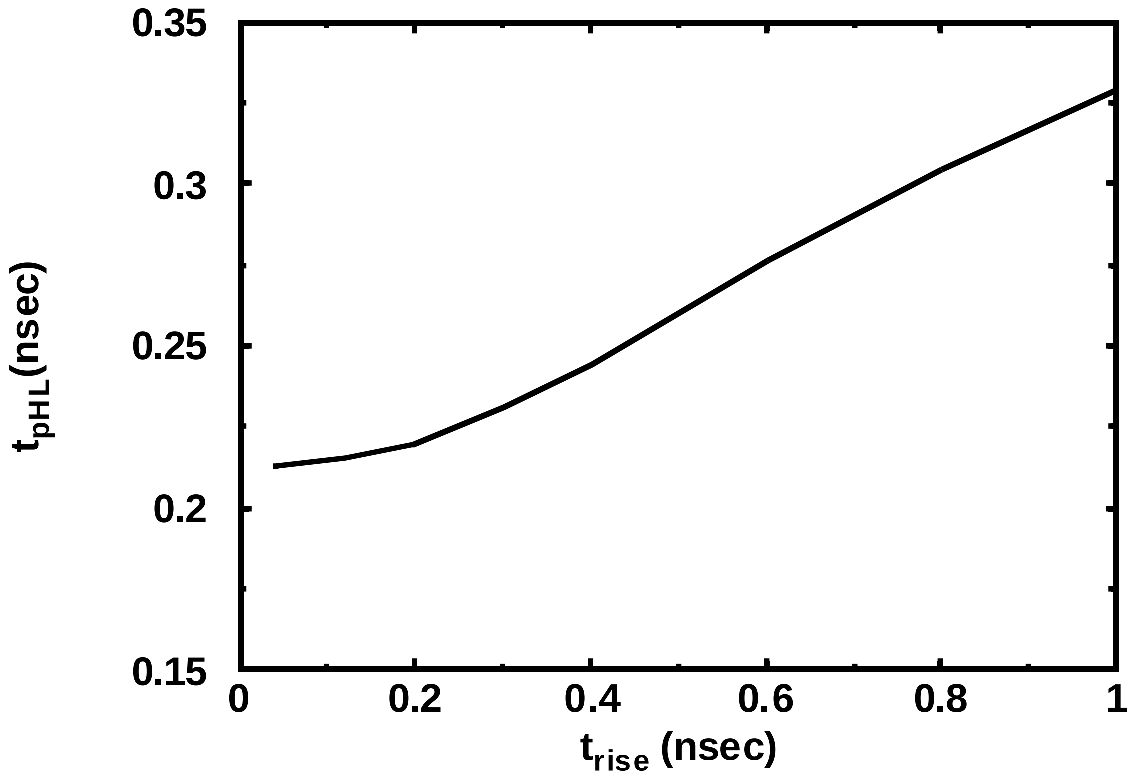


$\beta = W_p/W_n$

Design for Performance

- Keep capacitances small
- Increase transistor sizes
  - watch out for self-loading!
- Increase V_{DD} (????)

Impact of Rise Time on Delay



$$t_{pHL} = \sqrt{t_{pHL(step)}^2 + (t_r/2)^2}$$

UNIT-III
VLSI Physical Design

Basics of Design rules

- Design processes are always associated with certain concepts like stick diagrams and symbolic diagrams .
- But the key element is a set of design rules which forms the communication link between the designer (specifying requirements) and the fabricator (who materializes them).
- Design rules are used to produce workable mask layouts from which the various layers in silicon will be formed or patterned.
- Among the design rules Lambda –based rules are important. They are straightforward and relatively simple to apply.

Basics of Design rules

- However, they are 'real' and chips can be fabricated from mask layouts using the lambda-based rule set.
- Correct and faster designs will be realized if a fabricator's line is used to its full advantage
- such rule sets are needed not only to the fabricator but also to a specific technology.

- MOS LAYERS : MOS design is aimed at turning a specification into masks for processing silicon to meet the specification.

- We have seen that MOS circuits are formed on four basic layers-n-diffusion, p-diffusion, polysilicon, and metal, which are isolated from one another by thick or thin(thinox) silicon dioxide insulating layers.

- The thin oxide (thinox) mask region includes n-diffusion, pdiffusion, and transistor channels. Polysilicon and thinox regions interact so that a transistor is formed where they cross one another

- In some processes, there may be a second metal layer and also, in some processes, a second polysilicon layer.
- Layers may deliberately joined together where contacts are formed.

Stick Diagrams

- VLSI design aims to translate circuit concepts onto silicon.
- A stick diagram is a diagrammatic representation of a chip layout stick diagrams
- Stick diagrams convey layer information through colour codes (or monochrome encoding).
- Acts as an interface between symbolic circuit and the actual layout.
- Used by CAD packages, including Microwind
- For example, in the case of nMOS design, green color is used for n-diffusion, red for polysilicon, blue for metal, yellow for implant, and black

Stick Diagrams

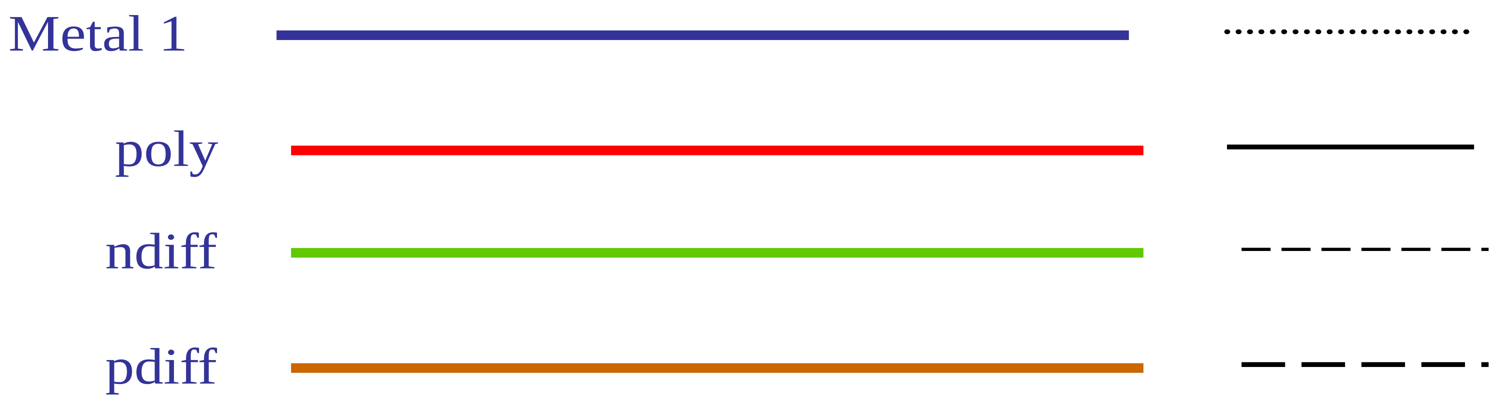
- Does show all components/vias.
- It shows relative placement of components.
- Goes one step closer to the layout
- Helps plan the layout and routing

- Does ***not*** show
- Exact placement of components
- Transistor sizes
- Wire lengths, wire widths, tub boundaries.
- Any other low level details such as parasitics..

Stick Diagrams – Notations

Stick Diagrams

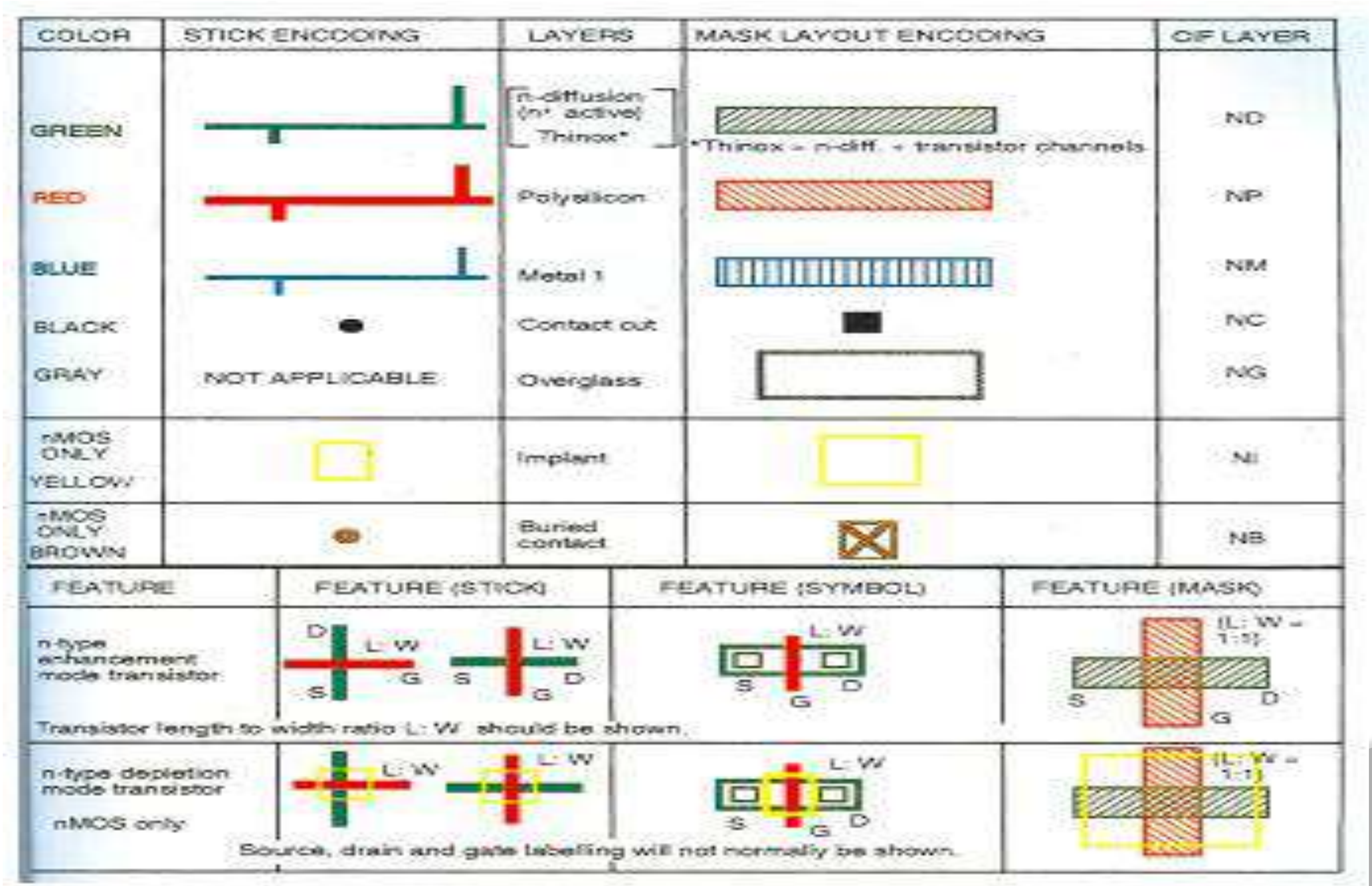
Stick Diagrams – Notations



Can also draw in shades of gray/line style.

Similarly for contacts, via, tub etc..

Stick Diagrams – Notations

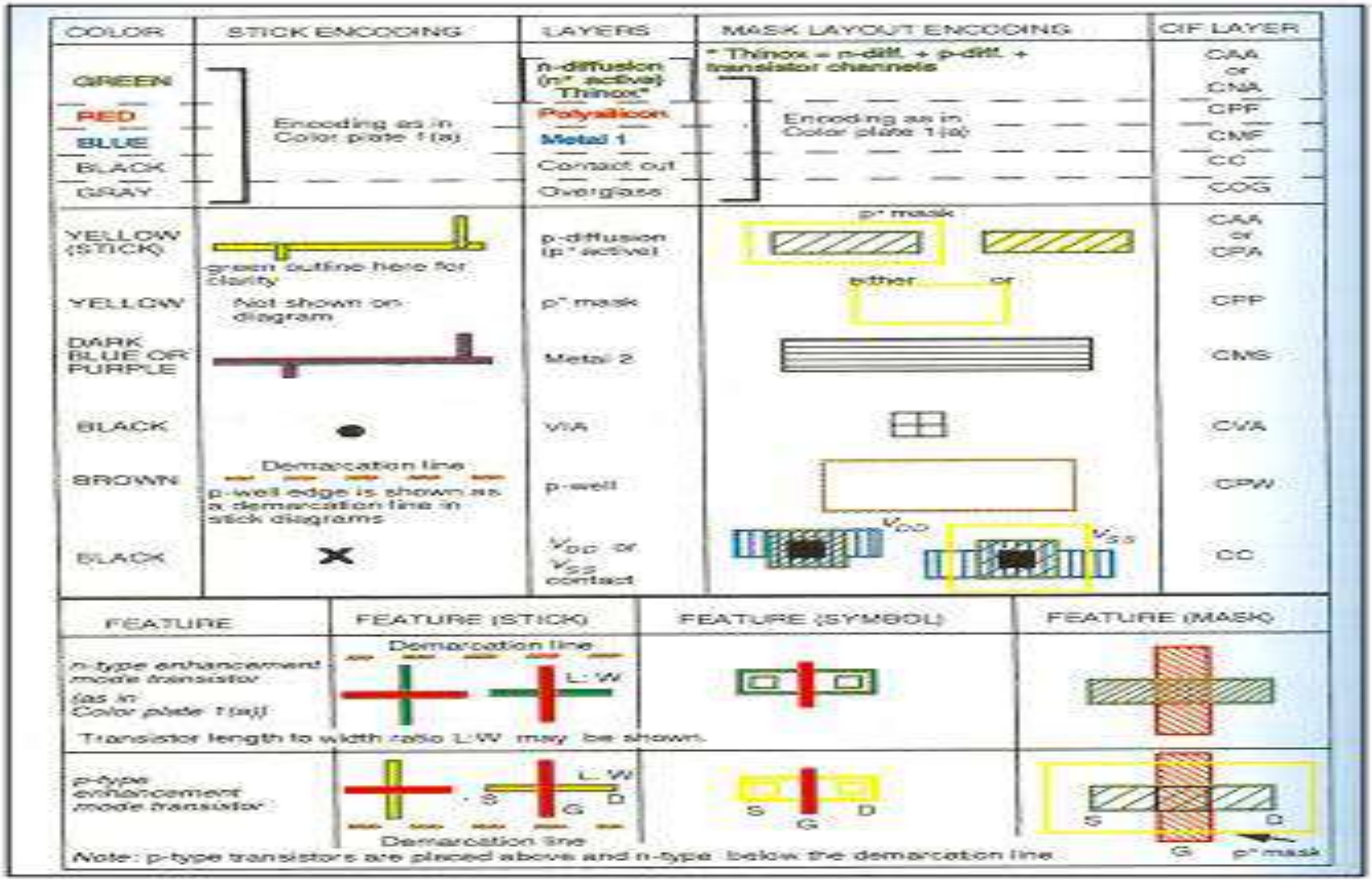


COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN		n-diffusion (n+ active) Thinox*	▨ *Thinox = n-diff. + transistor channels	ND
RED		Polysilicon	▨	NP
BLUE		Metal 1	▥	NM
BLACK	●	Contact cut	■	NC
GRAY	NOT APPLICABLE	Overglass	□	NG
nMOS ONLY YELLOW	□	Implant	□	NI
nMOS ONLY BROWN	●	Buried contact	⊠	NB

FEATURE	FEATURE (STICK)	FEATURE (SYMBOL)	FEATURE (MASK)
n-type enhancement mode transistor	D L:W S G; L:W S D G	L:W S G D	(L:W = 1:1) S D G
Transistor length to width ratio L:W should be shown.			
n-type depletion mode transistor nMOS only	L:W; L:W	L:W S G D	(L:W = 1:1)
Source, drain and gate labelling will not normally be shown.			

Figure 1: NMOS encodings

Stick Diagrams – Notations



COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN	Encoding as in Color plate 1(a)	n-diffusion (n^+ active) Thinox*	* Thinox = n-diff. + p-diff. + transistor channels Encoding as in Color plate 1(a)	CAA or CNA
RED		Polysilicon		CPF
BLUE		Metal 1		CMF
BLACK		Contact cut		CC
GRAY		Overglass		COG
YELLOW (STICK)	green outline here for clarity	p-diffusion (p^+ active)	p^+ mask	CAA or CPA
YELLOW	Not shown on diagram	p^+ mask	either or	CPP
DARK BLUE OR PURPLE		Metal 2		CMS
BLACK	●	VIA		CVA
BROWN	Demarcation line p-well edge is shown as a demarcation line in stick diagrams	p-well		CPW
BLACK	✕	V_{DD} or V_{SS} contact	V_{DD} V_{SS}	CC

FEATURE	FEATURE (STICK)	FEATURE (SYMBOL)	FEATURE (MASK)
n-type enhancement mode transistor (as in Color plate 1(a))	Demarcation line L:W		
p-type enhancement mode transistor	L:W S G D Demarcation line	S G D	S D G p^+ mask

Transistor length to width ratio L:W may be shown.

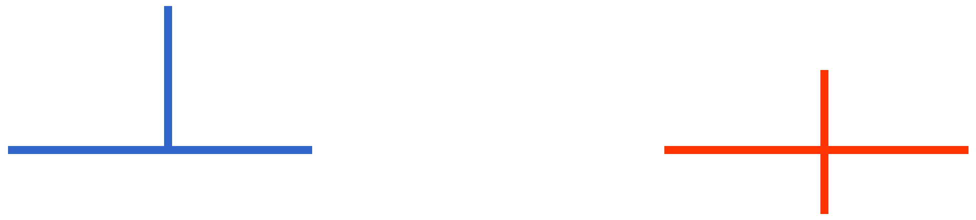
Note: p-type transistors are placed above and n-type below the demarcation line

Figure 2: CMOS encodings

Stick Diagrams – Some rules

Rule 1.

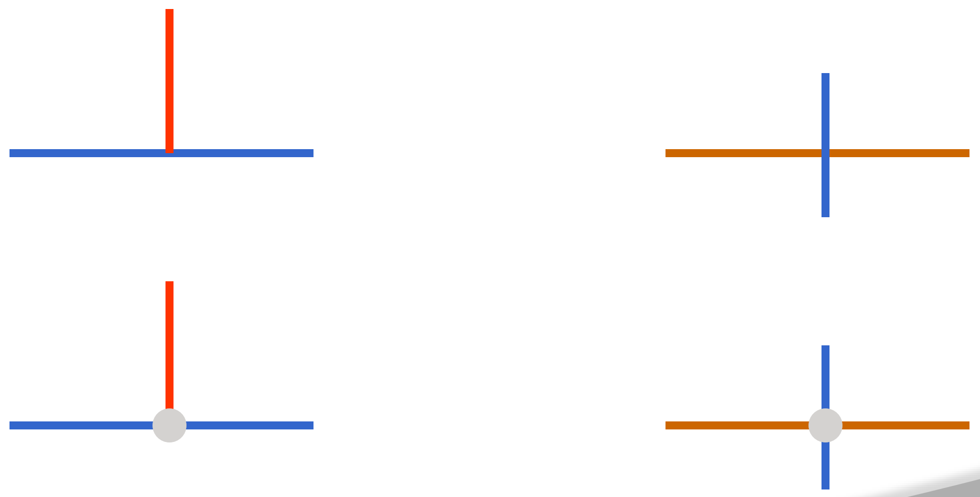
When two or more ‘sticks’ of the same type cross or touch each other that represents electrical contact.



Stick Diagrams – Some rules

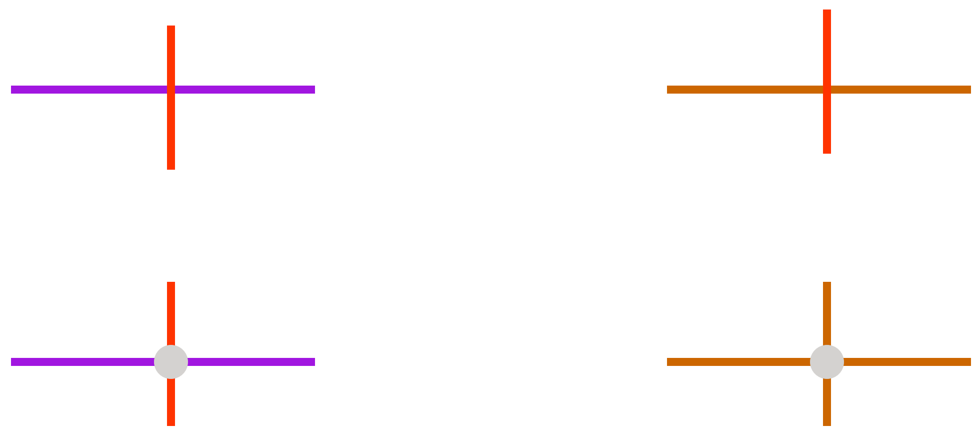
Rule 2.

When two or more ‘sticks’ of different type cross or touch each other there is no electrical contact.
(If electrical contact is needed we have to show the connection explicitly).



Stick Diagrams – Some rules

Rule 3. When a poly crosses diffusion it represents a transistor.

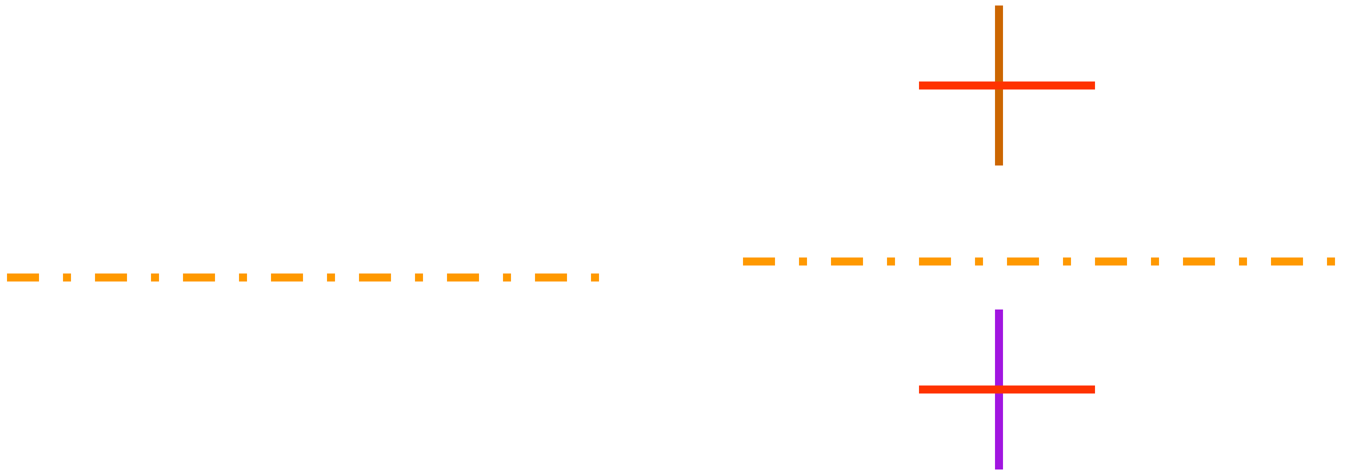


Note: If a contact is shown then it is ***not*** a transistor.

Stick Diagrams – Some rules

Rule 4.

In CMOS a demarcation line is drawn to avoid touching of p-diff with n-diff. All pMOS must lie on one side of the line and all nMOS will have to be on the other side.



nMOS Design Style

nMOS Design Style :

- To understand the design rules for nMOS design style , let us consider a single metal, single polysilicon nMOS technology. The layout of nMOS is based on the following important features.
- • n-diffusion [n-diff.] and other thin oxide regions [thinox] (green)
- • polysilicon 1 [poly.]-since there is only one polysilicon layer here (red);
- • metal 1 [metal]-since we use only one metal layer here (blue);
- • implant (yellow);
- • contacts (black or brown [buried]). A transistor is formed wherever poly. crosses n-diff. (red over green) and all diffusion wires (interconnections) are n-type (green).

Design Style

- When starting a layout, the first step normally taken is to draw the metal (blue) VDD and GND rails in parallel allowing enough space between them for the other circuit elements which will be required.
- Next, thinox (green) paths may be drawn between the rails for inverters and inverter based logic as shown in Fig. below.
- Inverters and inverter-based logic comprise a pull-up structure, usually a depletion mode transistor, connected from the output point to VDD and a pull-down structure of enhancement mode transistors suitably interconnected between the output point and GND.
- This is illustrated in the Fig.(b). remembering that poly. (red) crosses thinox (green) wherever transistors are required.
- One should consider the implants (yellow) for depletion mode transistors and also consider the length to width (L : W) ratio for each transistor. These ratios are important   particularly in nMOS and nMOS- like circuits.

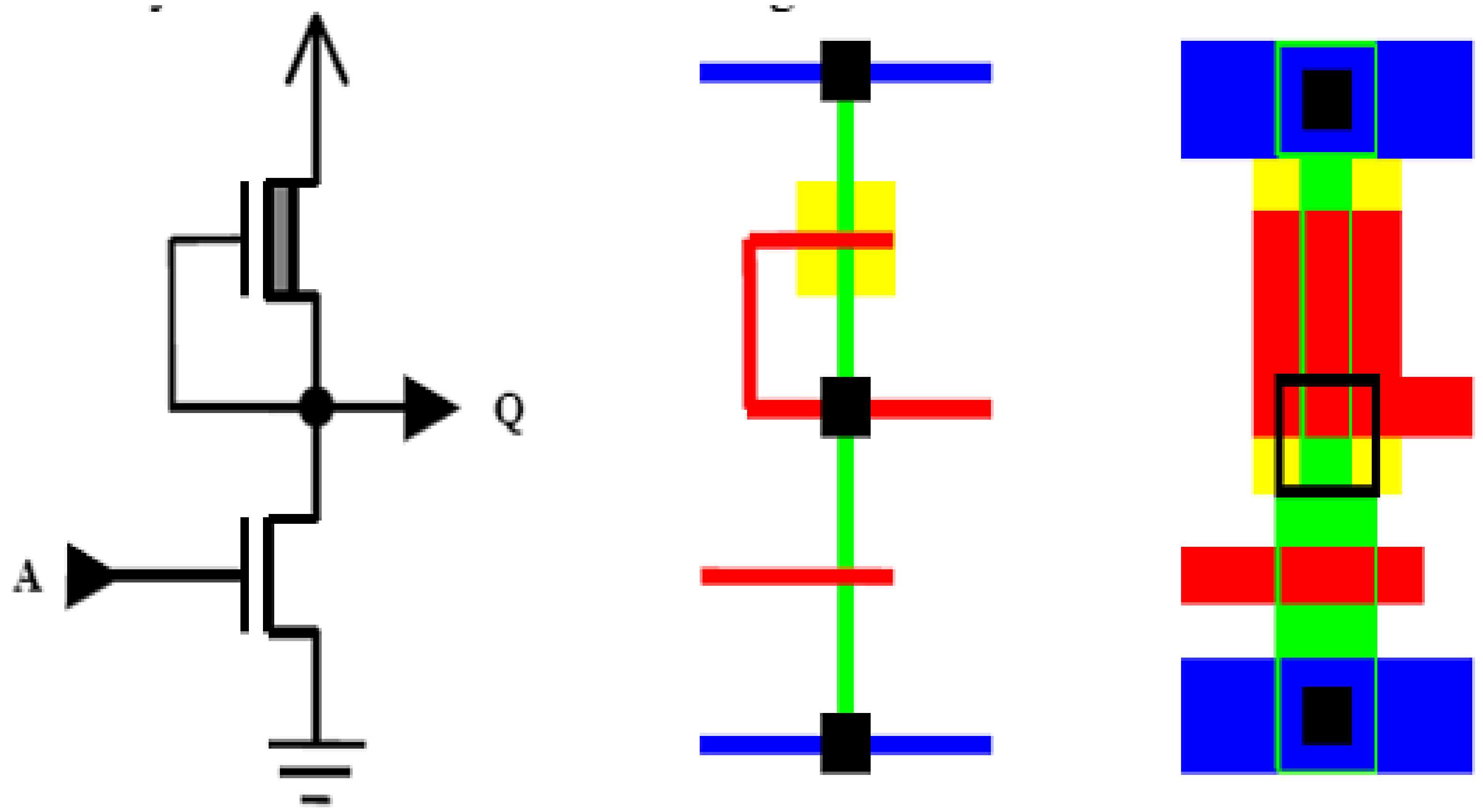
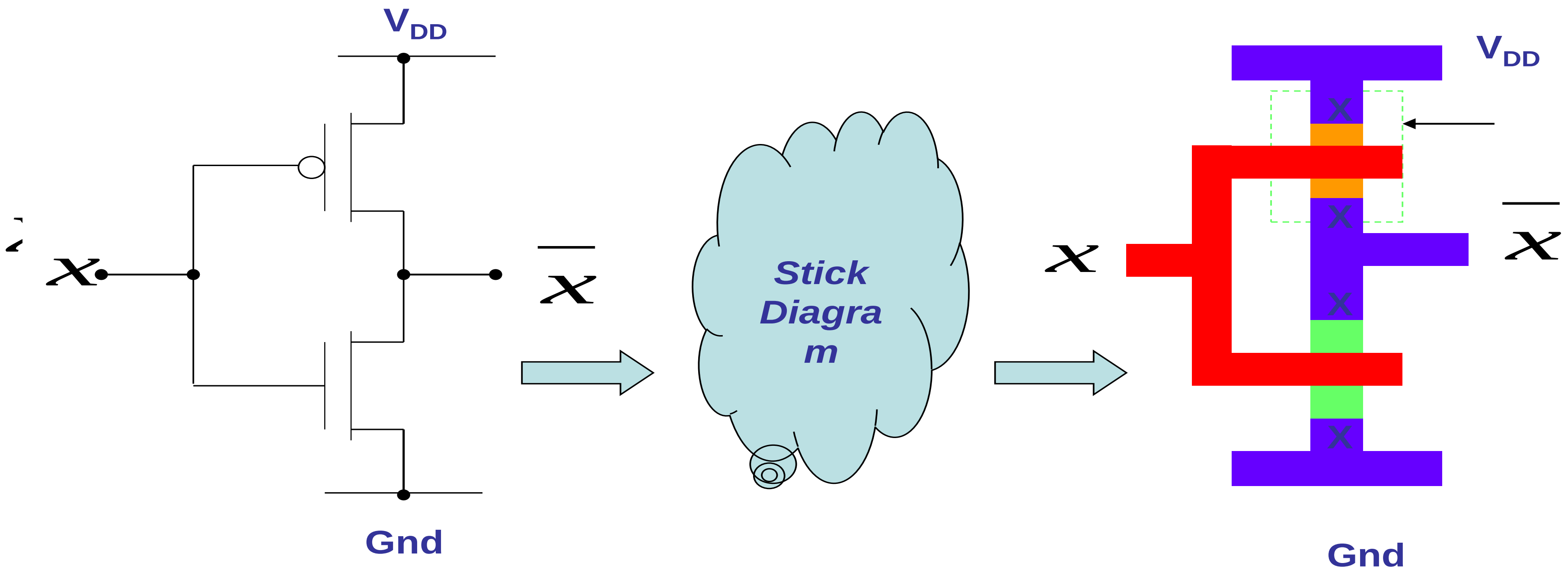


Figure 4: nMOS depletion load inverter

Stick Diagrams

Stick Diagrams

Stick Diagrams



18

CMOS Design Style

- The CMOS design rules are almost similar and extensions of n-MOS design rules except the implant (yellow) and the buried contact (brown).

. In CMOS design Yellow is used to identify ptransistors and wires, as depletion mode devices are not utilized.

.The two types of transistors 'n' and 'p', are separated by the demarcation line (representing the p-well boundary) above which all p-type devices are placed (transistors and wires (yellow). The n-devices (green) are consequently placed below the demarcation line

CMOS Design Style

- Diffusion paths must not cross the demarcation line and n-diffusion and p-diffusion wires must not join.

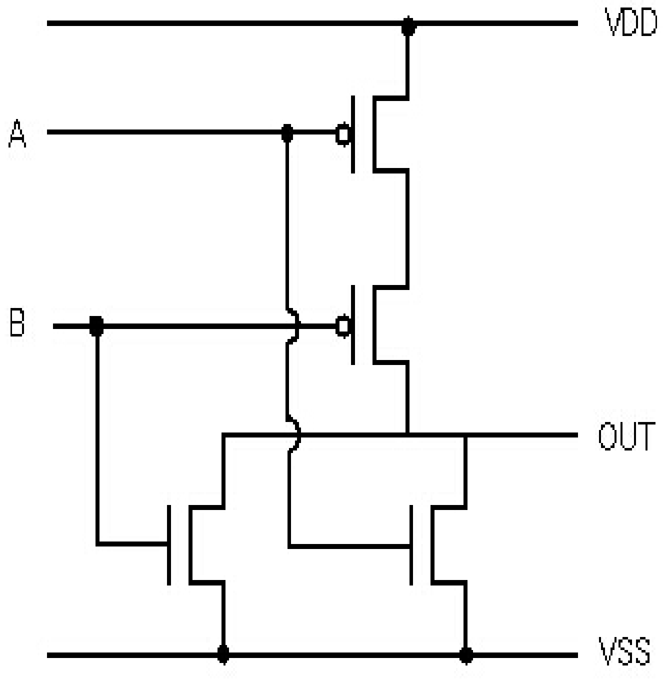
- The 'n' and 'p' features are normally joined by metal where a connection is needed. Their geometry will appear when the stick diagram is translated to a mask layout.

- However, one must not forget to place crosses on VDD and Vss rails to represent the substrate and p-well connection respectively.

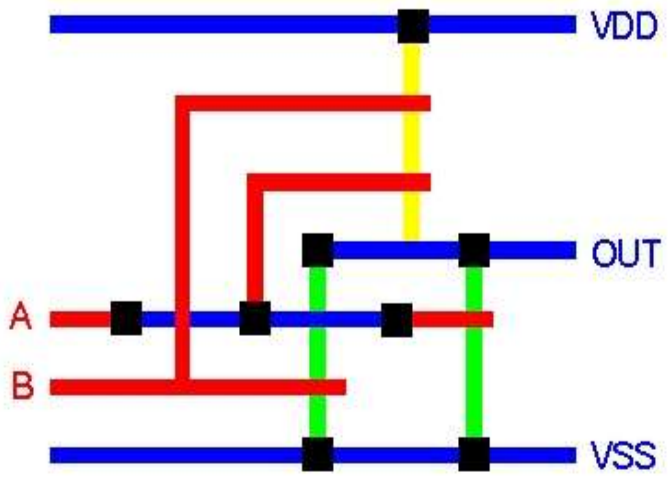
- The design begins with the drawing of the VDD and Vss rails in parallel and in metal and the creation of an (imaginary) demarcation line in-between, as shown in Fig.below.

- The n-transistors are then placed below this line and thus close to Vss, while p-transistors are placed above the line and below VDD

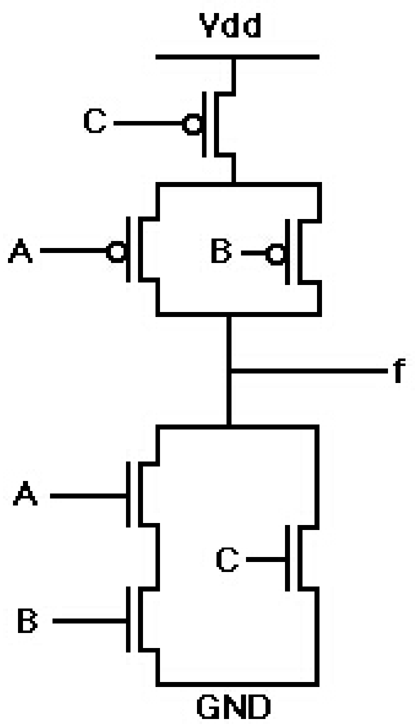
How to draw Stick Diagrams

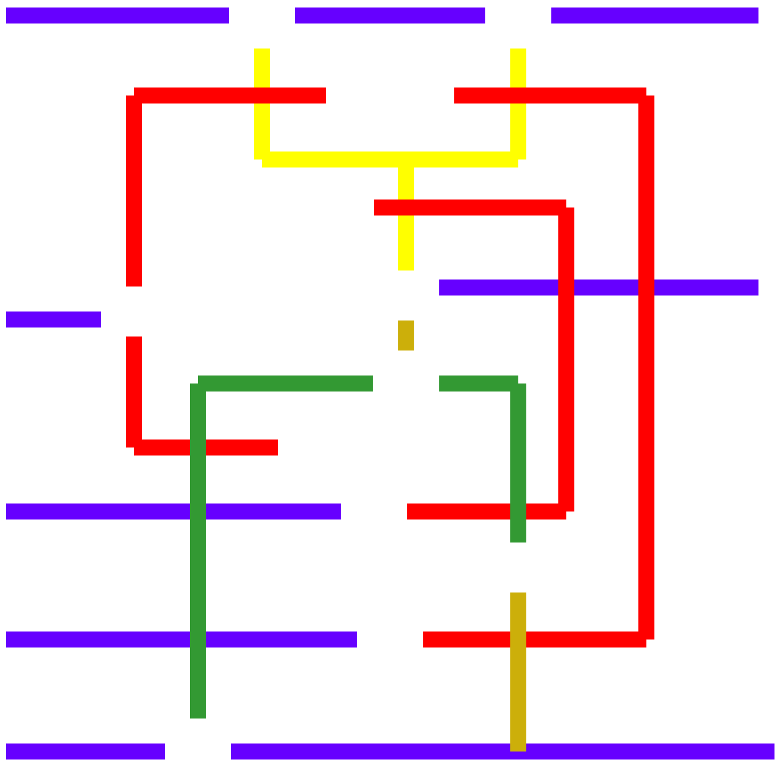


NOR gate in CMOS



Example: $f = \overline{(A \cdot B) + C}$





Lambda Based Design Rules

- Design rules based on single parameter, λ
- Simple for the designer
- Wide acceptance
- Provide feature size independent way of setting out mask
- If design rules are obeyed, masks will produce working circuits
- Minimum feature size is defined as 2 λ
- Used to preserve topological features on a chip
- Prevents shorting, opens, contacts from slipping out of area to be contacted

Design Rules and Layout

- The design rules are formed to translate the 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.
- The design rules also help to provide a reliable compromise between the circuit/system designer and the fabrication engineer.
- In general the circuit designers expect smaller layouts for improved performance and decreased silicon area.
- On the other hand, the process engineer like those design rules that result in a controllable and reproducible process.
- In fact there is a need of compromise for a competitive circuit to be produced at a reasonable cost.

Design Rules - The Reality

- Manufacturing processes have inherent limitations in accuracy and repeatability
- Design rules specify geometry of masks that provide reasonable yield
- Design rules are determined by experience

Problems - Manufacturing

- Photoresist shrinking / tearing
- Variations in material deposition
- Variations in temperature
- Variations in oxide thickness
- Impurities
- Variations between lots
- Variations across the wafer

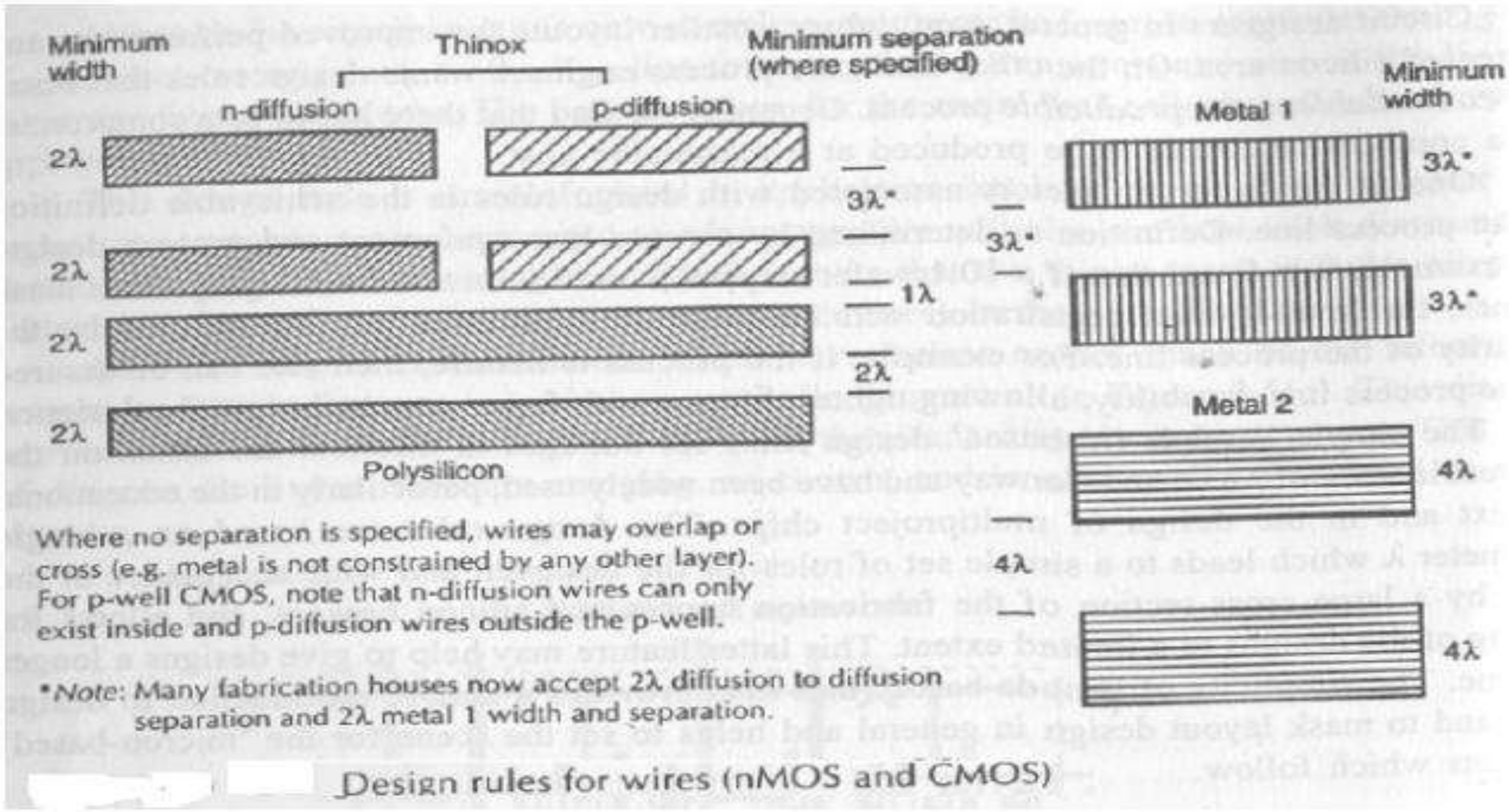
Problems - Manufacturing

- Variations in threshold voltage
  - oxide thickness
  - ion implantation
  - poly variations
- Diffusion - changes in doping (variation in R, C)
- Poly, metal variations in height and width -> variation in R, C
- Shorts and opens
- Via may not be cut all the way through
- Undersize via has too much resistance
- Oversize via may short

Lambda-based Design Rules :

- Design rules include width rules and spacing rules.
- Mead and Conway developed a set of simplified scalable λ -based design rules, which are valid for a range of fabrication technologies.
- In these rules, the minimum feature size of a technology is characterized as 2 λ .
- All width and spacing rules are specified in terms of the parameter λ .
- Suppose we have design rules that call for a minimum width of 2 λ , and a minimum spacing of 3 λ .
- If we select a 2 um technology (i.e., λ = 1 um), the above rules are translated to a minimum width of 2 um and a minimum spacing of 3 um.
- On the other hand, if a 1 um technology (i.e., λ = 0.5 um) is selected, then the same width and spacing rules are now specified as 1 um and 1.5 um, respectively.

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.

Design rules for wires (nMOS and CMOS)

Figure 10: Design rules for the diffusion layers and metal lalyers

Lambda-based Design Rules

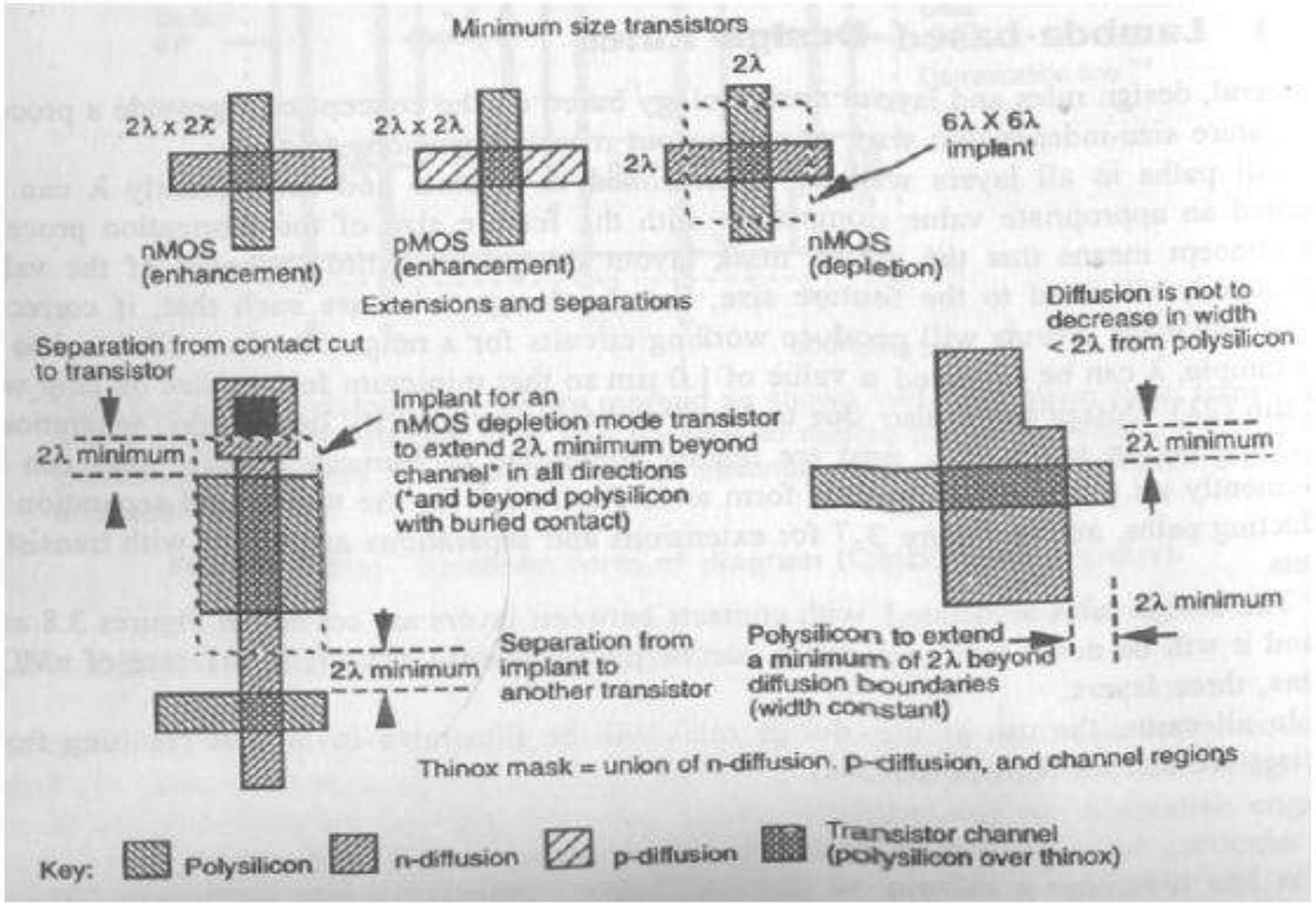
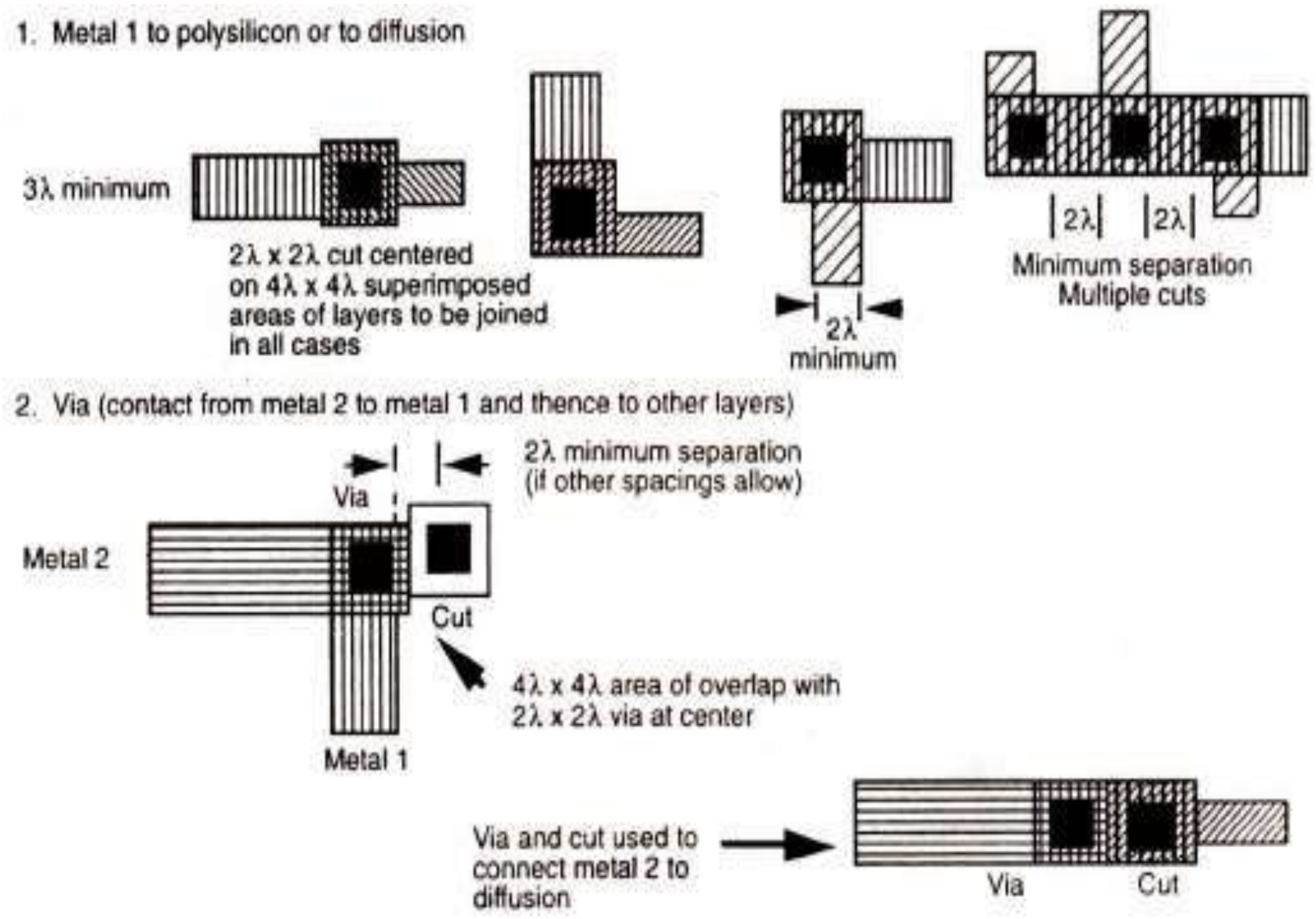

Minimum size transistors
2λ x 2λ
nMOS (enhancement)
2λ x 2λ
pMOS (enhancement)
2λ
2λ
6λ X 6λ implant
nMOS (depletion)
Extensions and separations
Separation from contact cut to transistor
2λ minimum
Implant for an nMOS depletion mode transistor to extend 2λ minimum beyond channel* in all directions (*and 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
Key: Polysilicon
n-diffusion
p-diffusion
Transistor channel (polysilicon over thinox)

Figure 11: Design rules for transistors and gate over hang distance

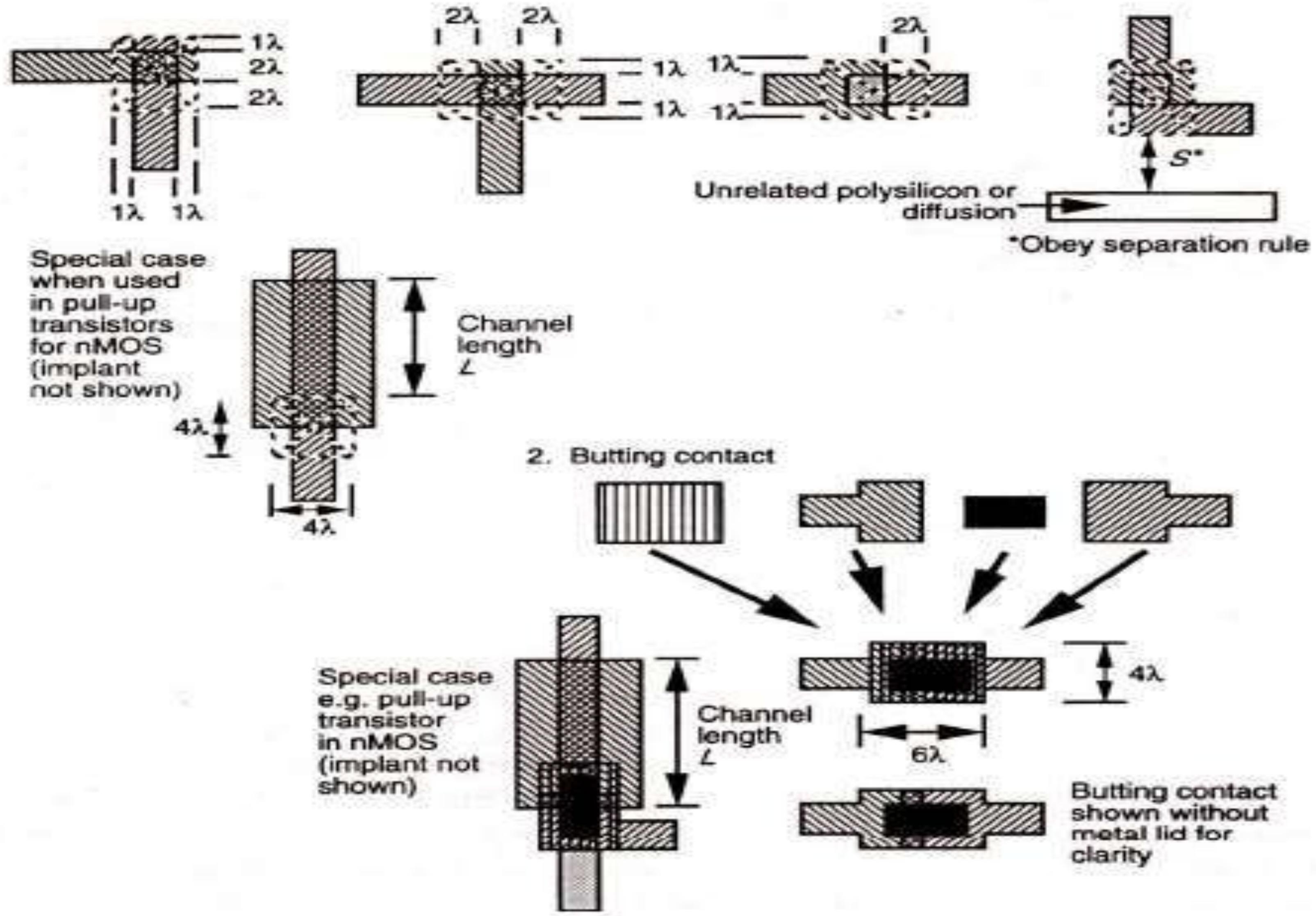
Contact Cuts

- While making contacts between poly-silicon and diffusion in Nmos circuits it should be remembered that there are three possible approaches—
- poly. to metal then metal to diff., or a buried contact
- poly. to diff. ,
- a butting contact (poly. to diff. using metal).
- Among the three the latter two, the buried contact is the most widely used, because of advantage in space and a reliable contact.
- At one time butting contacts were widely used , but now a days they are superseded by buried contacts.
- In CMOS designs, poly. to diff. contacts are always made via metal.

Lambda-based Design Rules


1. Metal 1 to polysilicon or to diffusion
3λ minimum
2λ x 2λ cut centered on 4λ x 4λ superimposed areas of layers to be joined in all cases
2λ minimum
2λ
2λ
Minimum separation Multiple cuts
2. Via (contact from metal 2 to metal 1 and thence to other layers)
Via
2λ minimum separation (if other spacings allow)
Metal 2
Cut
4λ x 4λ area of overlap with 2λ x 2λ via at center
Metal 1
Via and cut used to connect metal 2 to diffusion
Via
Cut

Lambda-based Design Rules


1λ
2λ
2λ
1λ 1λ
2λ 2λ
1λ
1λ
1λ
1λ
2λ
S*
Unrelated polysilicon or diffusion
*Obey separation rule
Special case when used in pull-up transistors for nMOS (implant not shown)
Channel length L
4λ
4λ
2. Butting contact
4λ
6λ
Special case e.g. pull-up transistor in nMOS (implant not shown)
Channel length L
Butting contact shown without metal lid for clarity

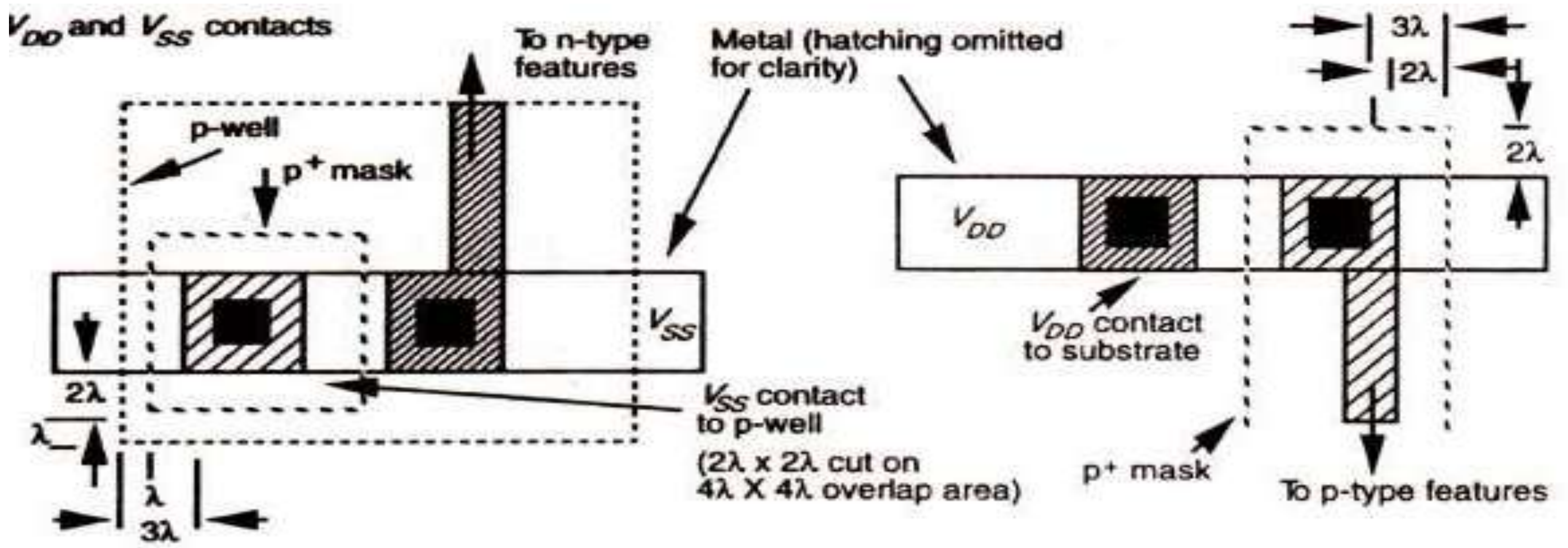
Double metal MOS process rules

- In the MOS design rules a powerful design process is achieved by adding a second metal layer.
- This gives a much greater degree of freedom, in distributing global VDD and Vss(GND) rails in a system.
- From the overall chip inter-connection aspect, the second metal layer in particular is connection. to other layers using metal1 to metal 2 contacts, called vias
- Similarly Poly 2 present along with Poly1
- The important features of double metal process are summarized as follows :
- Use the second level metal for the global distribution of power buses, that is, VDD and GND (Vss), and for clock lines.
- Use the first level metal for local distribution of power and for signal lines.

CMOS Lambda-based Design Rules

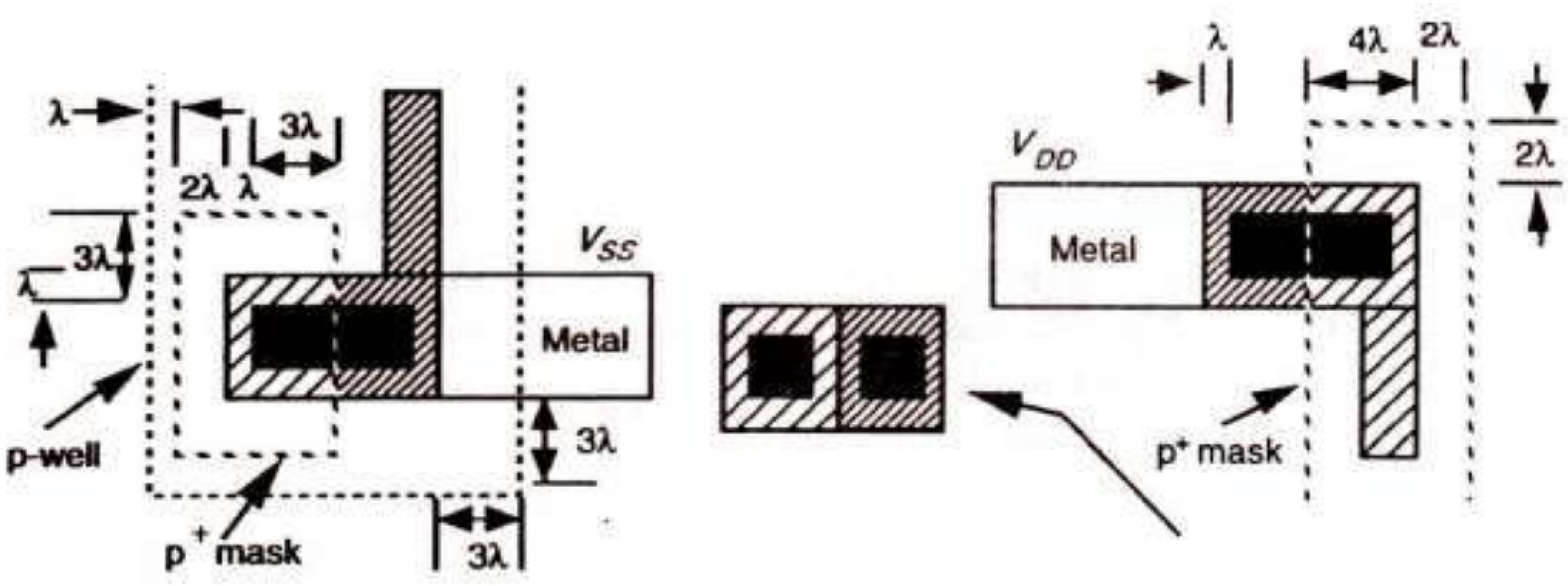
- The CMOS fabrication process is more complex than nMOS fabrication .
- The additional rules are concerned with those features unique to p-well CMOS, such as the p-well and p+ mask and the special 'substrate‘ contacts.

.



V_{DD} and V_{SS} contacts
To n-type features
Metal (hatching omitted for clarity)
p-well
p^+ mask
V_{SS}
2λ
λ
λ
3λ
V_{SS} contact to p-well
(2λ x 2λ cut on 4λ X 4λ overlap area)
3λ
2λ
2λ
V_{DD}
V_{DD} contact to substrate
p^+ mask
To p-type features

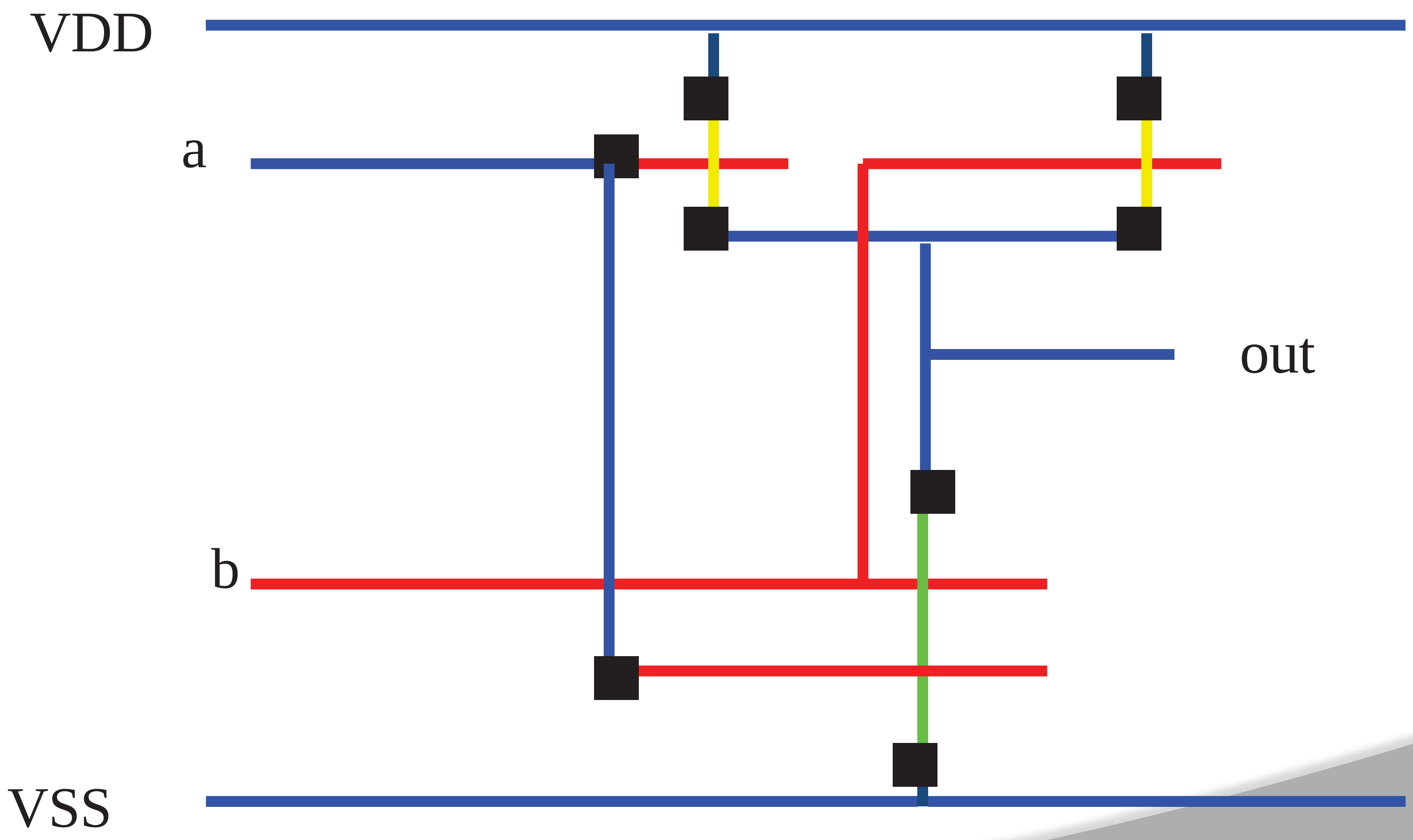
In the diagram above each of the arrangements can be merged into single split contacts.



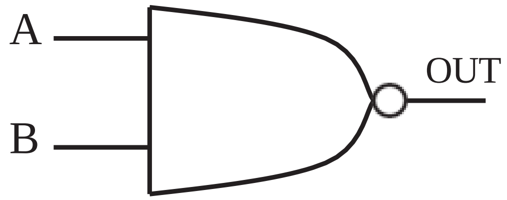
From the above diagram it is also clear that split contacts may also be made with separate cuts

Stick Diagram - Example

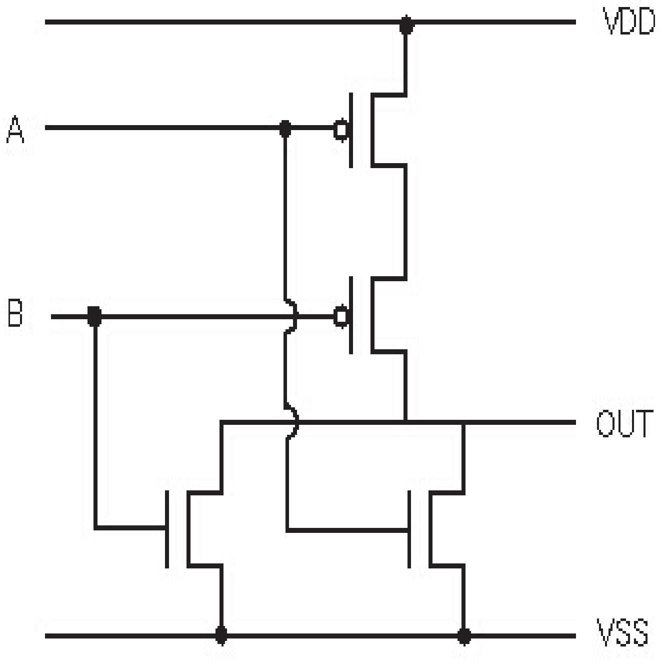
NAND sticks



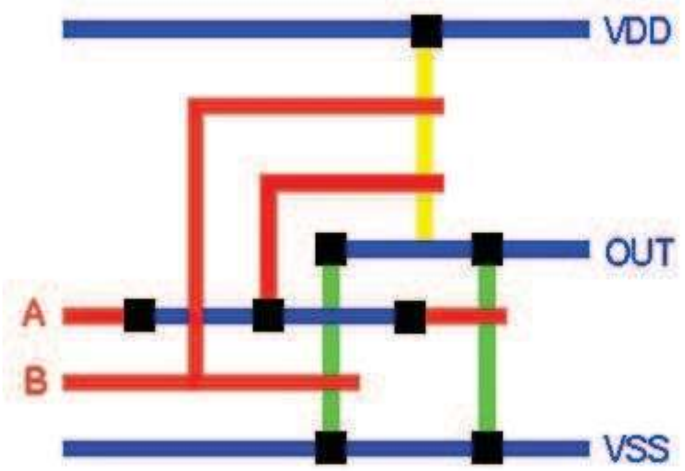
Stick Diagram - Example



NOR Gate

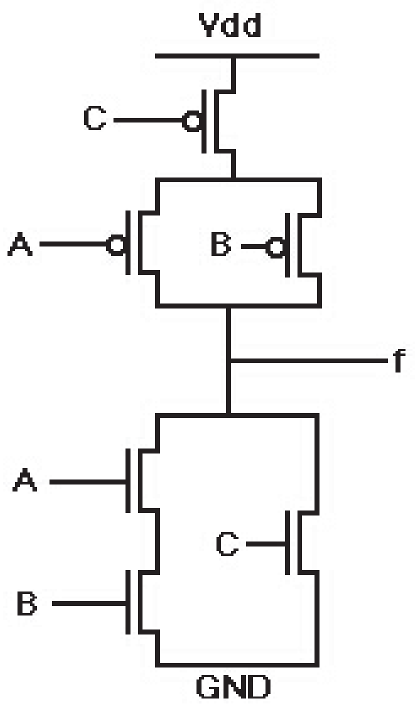


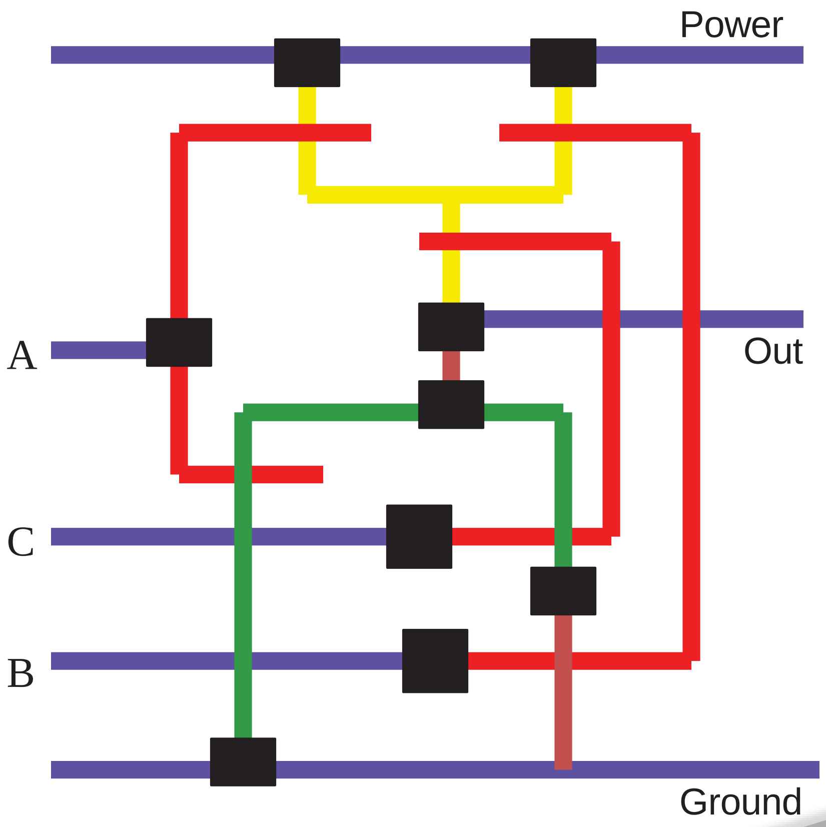
NOR gate in CMOS



Stick Diagram - Example

Example: $f = \overline{(A \bullet B) + C}$



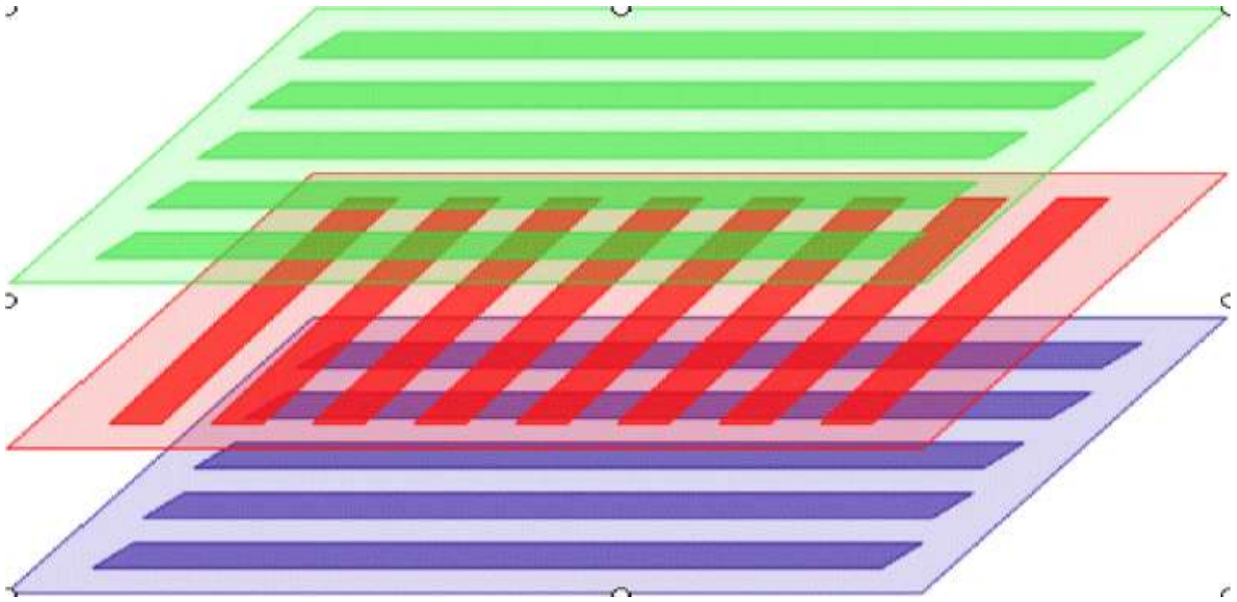


VLSI Interconnects

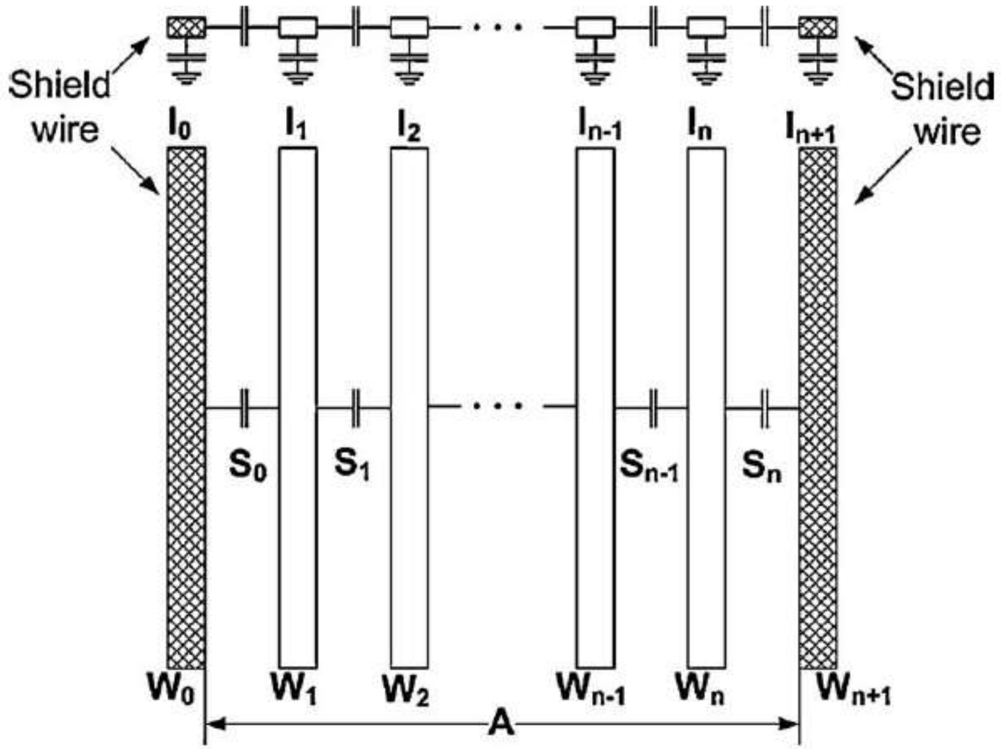
- Introduction
- Wire Resistance
- Wire Capacitance
- Wire RC Delay
- Crosstalk
- Wire Engineering
- Repeaters

Introduction

- Chips are mostly made of wires called *interconnect*
- Alternating layers run orthogonally

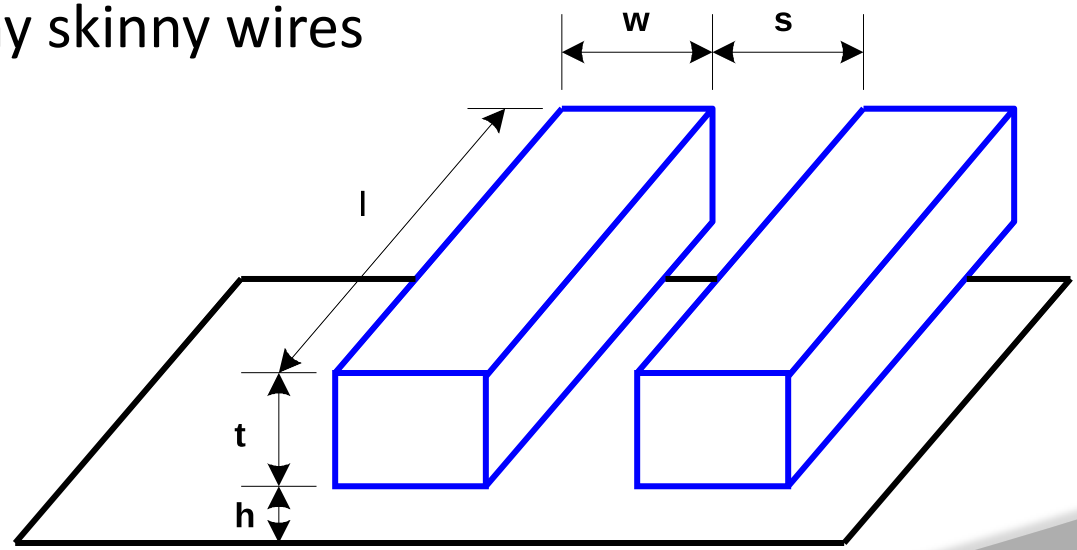


- Wires are as important as transistors
  - Speed
  - Power
  - Noise



Wire Geometry

- Pitch = w + s
- Aspect ratio: AR = t/w
  - Old processes had AR << 1
  - Modern processes have AR ≈ 2
    - Pack in many skinny wires

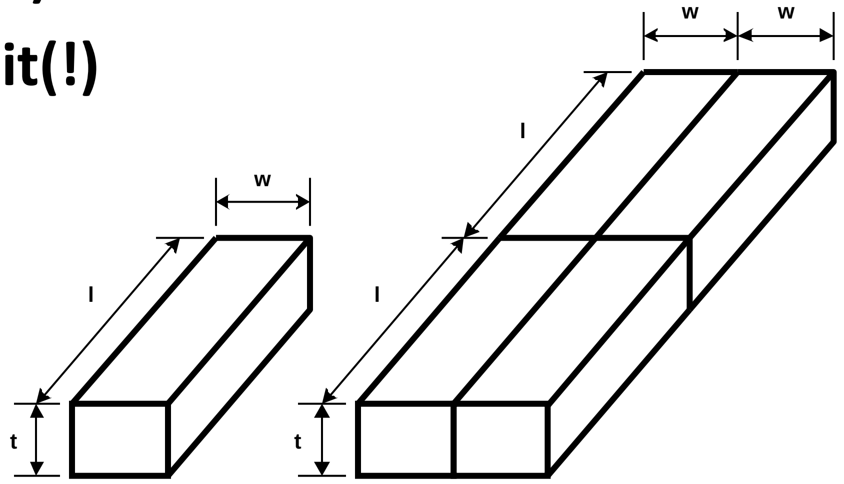


Wire Resistance

- ρ = ***resistivity*** **(Ω*m)**

$$R = \frac{\rho}{t}\frac{l}{w} = R_W\frac{l}{w}$$

- **R□ =** ***sheet resistance*** **(Ω/□)**
  - **□ is a dimensionless unit(!)**
- **Count number of squares**
  - **R = R□ * (# of squares)**



1 Rectangular Block
R = R□ (L/W) Ω

4 Rectangular Blocks
R = R□ (2L/2W)Ω
= R□ (L/W) Ω

Choice of Metals

- Until 180 nm generation, most wires were aluminum
- Modern processes use copper
  - Cu atoms diffuse into silicon and damage FETs
  - Must be surrounded by a diffusion barrier

Metal	**Bulk resistivity (μΩ*cm)**
Silver (Ag)	1.6
Copper (Cu)	1.7
Gold (Au)	2.2
Aluminum (Al)	2.8
Tungsten (W)	5.3
Molybdenum (Mo)	5.3

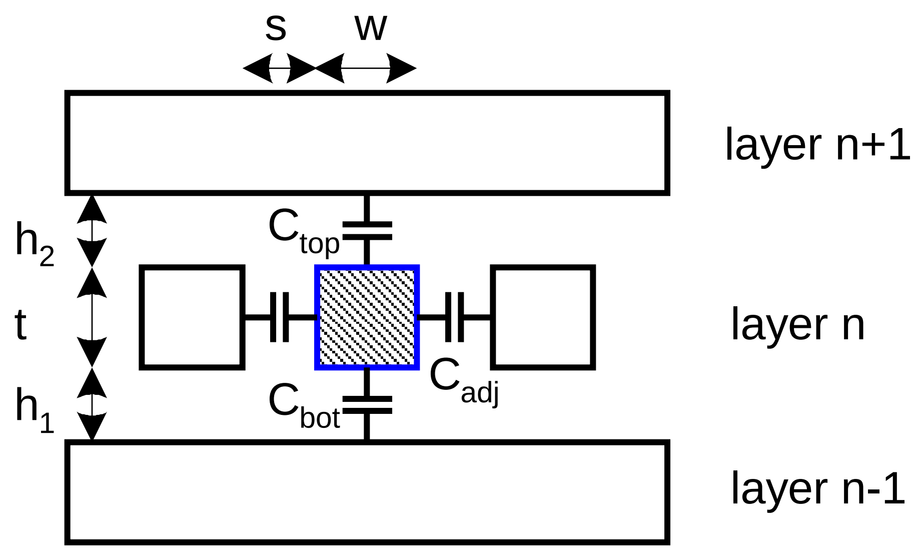
Sheet Resistance

- **Typical sheet resistances in 180 nm process**

Layer	**Sheet Resistance (Ω/□)**
Diffusion (silicided)	3-10
Diffusion (no silicide)	50-200
Polysilicon (silicided)	3-10
Polysilicon (no silicide)	50-400
Metal1	0.08
Metal2	0.05
Metal3	0.05
Metal4	0.03
Metal5	0.02
Metal6	0.02

Wire Capacitance

- Wire has capacitance per unit length
  - To neighbors
  - To layers above and below
- $C_{total} = C_{top} + C_{bot} + 2C_{adj}$

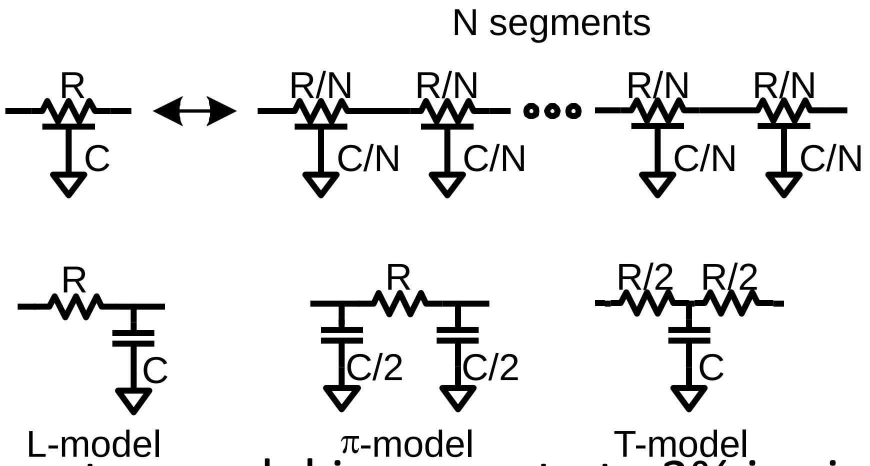


Capacitance Trends

- Parallel plate equation: $C = eA/d$
  - Wires are not parallel plates, but obey trends
  - Increasing area (W, t) increases capacitance
  - Increasing distance (s, h) decreases capacitance
- Dielectric constant
  - $e = ke_0$
- $e_0 = 8.85 \times 10^{-14}$ F/cm
- $k = 3.9$ for SiO_2
- Processes are starting to use low-k dielectrics
  - $k \approx 3$ (or less) as dielectrics use air pockets

Lumped Element Models

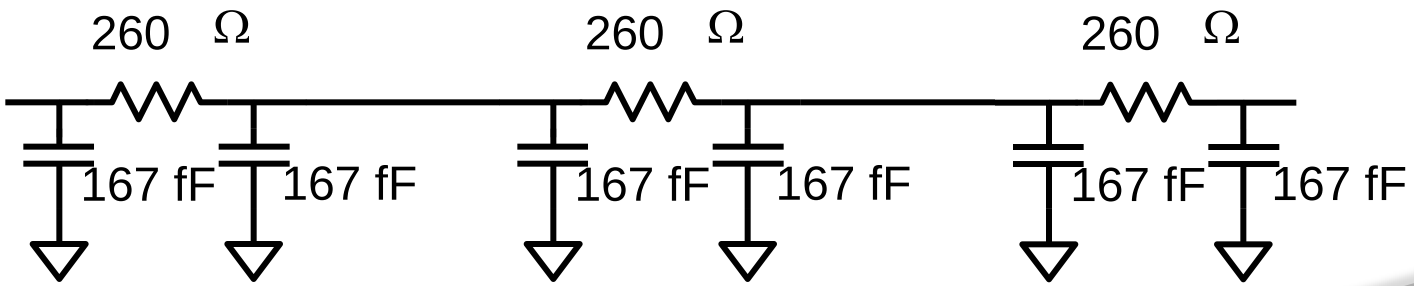
- Wires are a distributed system
  - Approximate with lumped element models



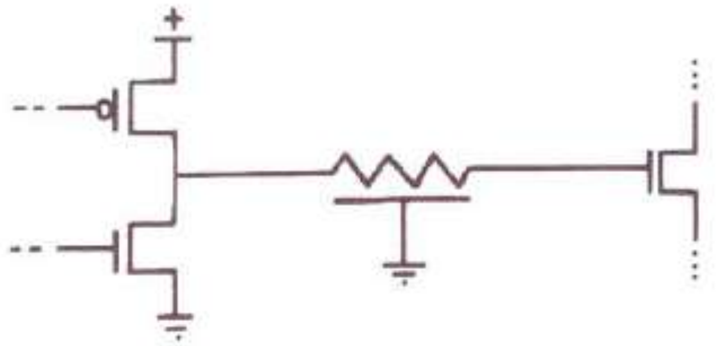
- 3-segment π-model is accurate to 3% in simulation

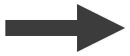
RC Example

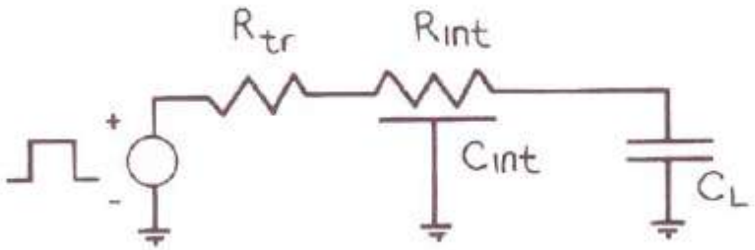
- Metal2 wire in 180 nm process
  - 5 mm long
  - 0.32 mm wide
- Construct a 3-segment p-model
  - $R_{\square}$ = 0.05 W/□ => R = 781 W
  - $C_{permicron}$ = 0.2 fF/mm => C = 1 pF



RC Delay







$$
\begin{aligned}
T_{90\%} &= 1.0R_{int}C_{int} + 2.3(R_{tr}C_{int} + R_{tr}C_L + R_{int}C_L) \\
&\approx (2.3R_{tr} + R_{int})C_{int} \qquad \text{for} \quad C_L \ll C_{int}.
\end{aligned}
$$

$$
\begin{aligned}
T_{50\%} &= 0.4R_{int}C_{int} + 0.7(R_{tr}C_{int} + R_{tr}C_L + R_{int}C_L) \\
&\approx (0.7R_{tr} + 0.4R_{int})C_{int} \qquad \text{for} \quad C_L \ll C_{int}.
\end{aligned}
$$

Crosstalk

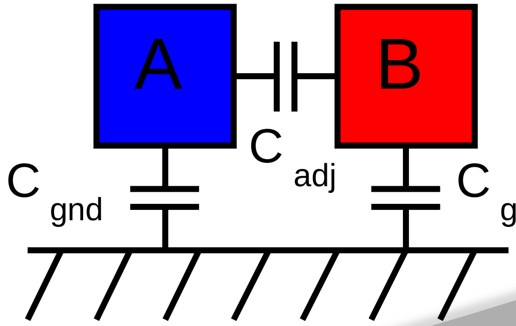
- A capacitor does not like to change its voltage instantaneously
- A wire has high capacitance to its neighbor.
  - When the neighbor switches from 1→ 0 or 0→ 1, the wire tends to switch too.
  - Called capacitive *coupling* or *crosstalk*
- Crosstalk effects
  - Noise on non switching wires
  - Increased delay on switching wires

Miller Effect

- Assume layers above and below on average are quiet
  - Second terminal of capacitor can be ignored
  - Model as $C_{gnd} = C_{top} + C_{bot}$
- Effective C_{adj} depends on behavior of neighbors
  - *Miller effect*

B	**ΔV**	**$C_{eff(A)}$**	**MCF**
Constant	V_{DD}	$C_{gnd} + C_{adj}$	1
Switching with A	0	C_{gnd}	0
Switching opposite A	$2V_{DD}$	$C_{gnd} + 2\ C_{adj}$	2

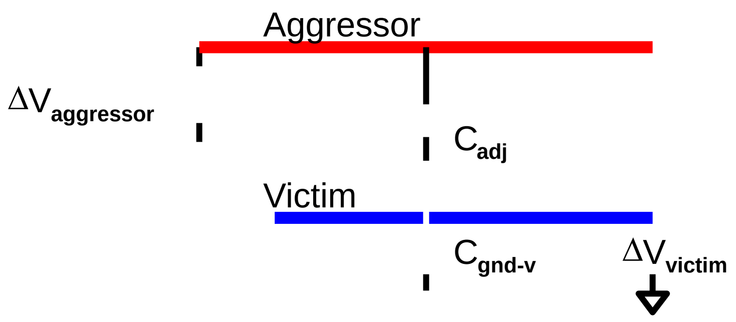
Delay and power increase



Crosstalk Noise

- Crosstalk causes noise on non switching wires
- If victim is floating:
  - model as capacitive voltage divider

$$\Delta V_{\text{victim}} = \frac{C_{\text{adj}}}{C_{\text{gnd-v}} + C_{\text{adj}}} \Delta V_{\text{aggressor}}$$

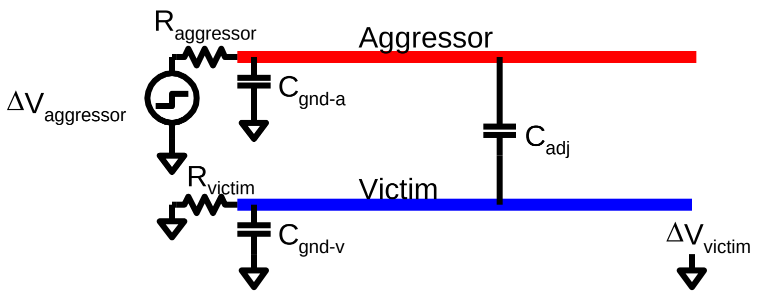


Driven Victims

- **Usually victim is driven by a gate that fights noise**
  - **Noise depends on relative resistances**
  - **Victim driver is in linear region, agg. in saturation**
  - **If sizes are same, $R_{\text{aggressor}}$ = 2-4 x R_{victim}**

$$\Delta V_{\text{victim}} = \frac{C_{\text{adj}}}{C_{\text{gnd-v}} + C_{\text{adj}}} \frac{1}{1+k} \Delta V_{\text{aggressor}}$$

$$k = \frac{\tau_{\text{aggressor}}}{\tau_{\text{victim}}} = \frac{R_{\text{aggressor}}\left(C_{\text{gnd-a}} + C_{\text{adj}}\right)}{R_{\text{victim}}\left(C_{\text{gnd-v}} + C_{\text{adj}}\right)}$$

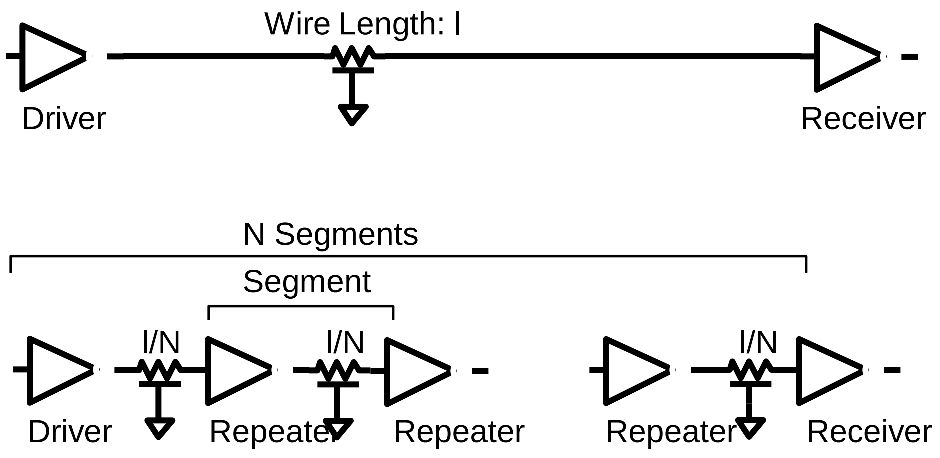


Noise Implications

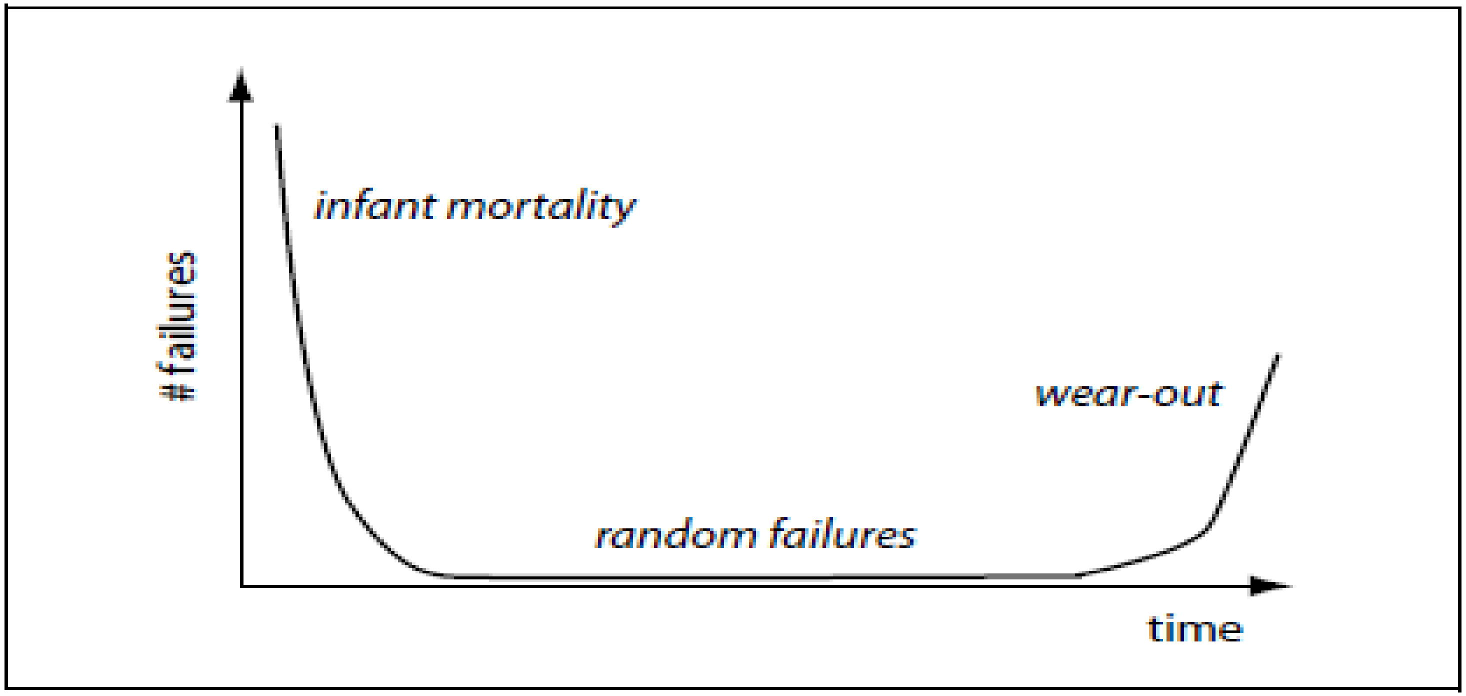
- *So what* if we have noise?
- If the noise is less than the noise margin, nothing happens
- Static CMOS logic will eventually settle to correct output even if disturbed by large noise spikes
  - But glitches cause extra delay
  - Also cause extra power from false transitions
- Dynamic logic never recovers from glitches
- Memories and other sensitive circuits also can produce the wrong answer

Repeaters

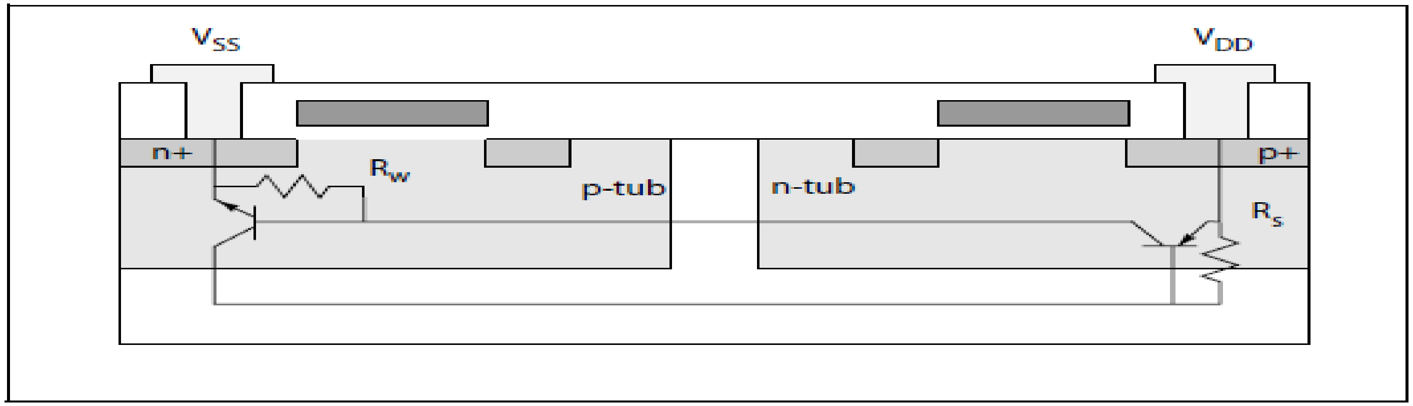
- R and C are proportional to l
- RC delay is proportional to l^2
  - Unacceptably large for long wires
- Break long wires into N shorter segments
  - Drive each one with an inverter or buffer

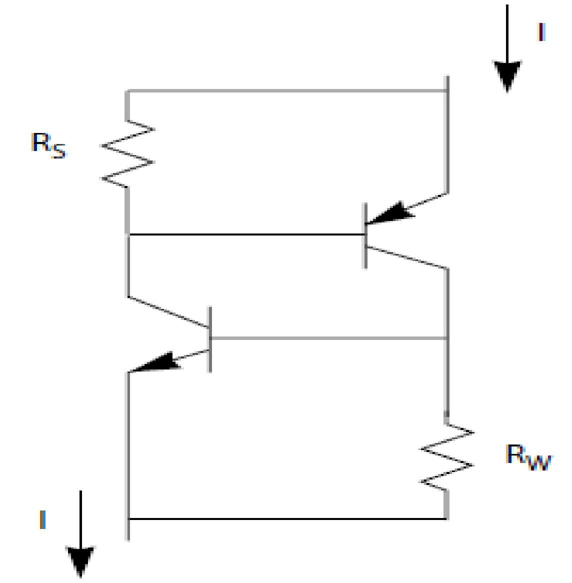


Reliability issues in CMOS VLSI

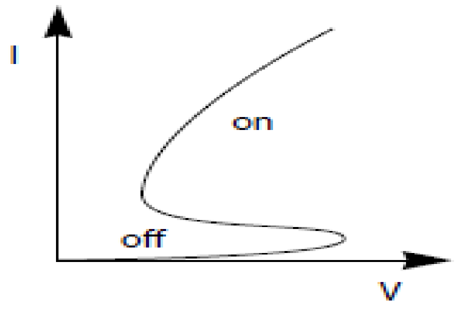


Latching



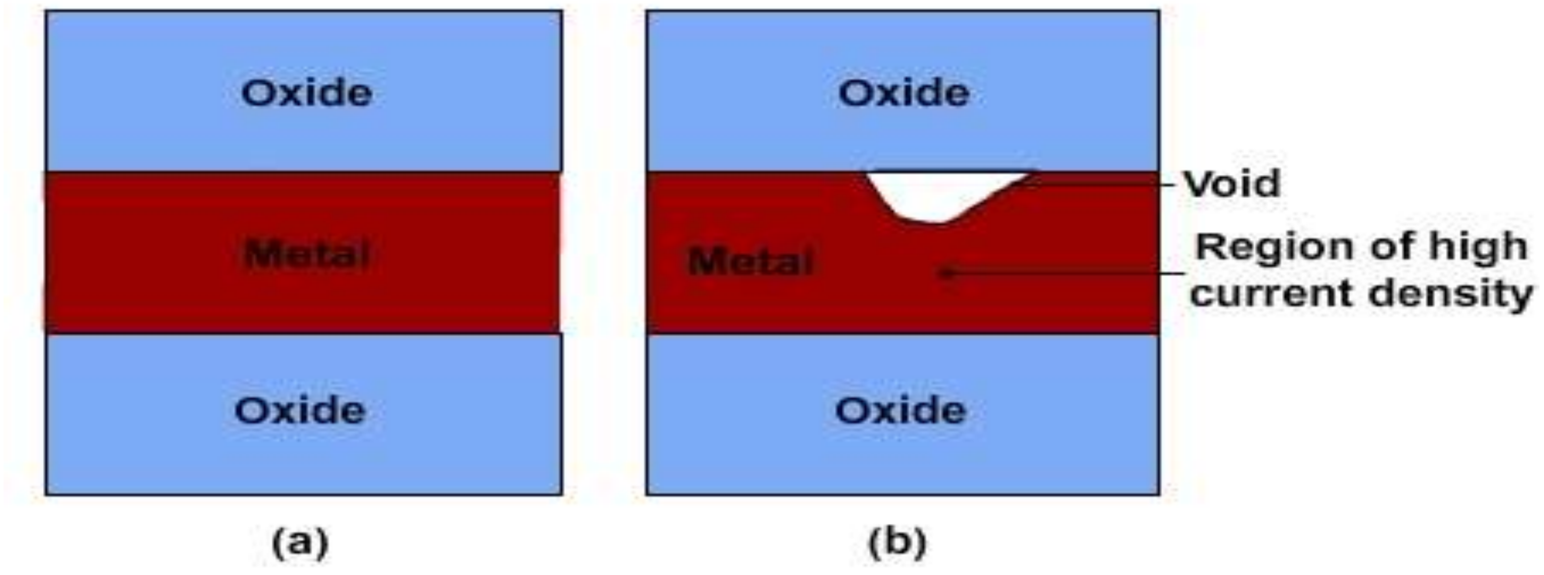


equivalent circuit



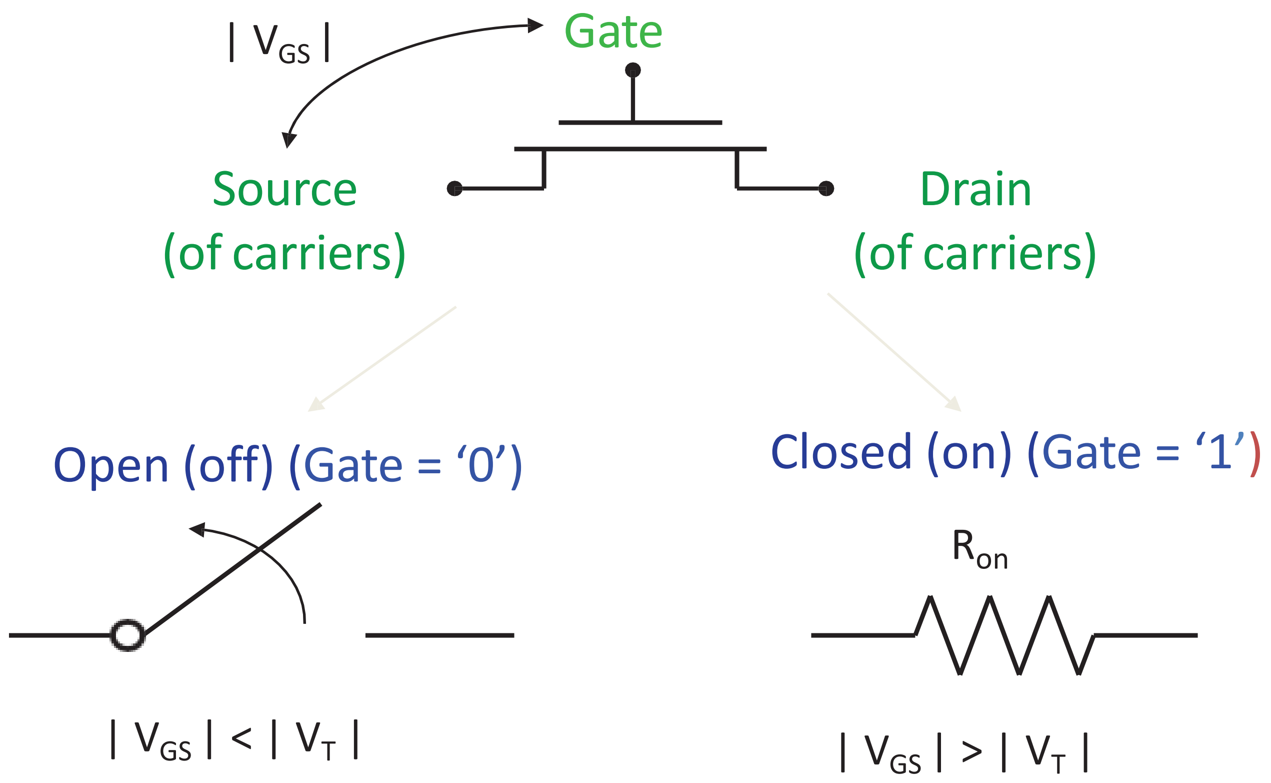
terminal behavior

ELECTRO MIGRATION

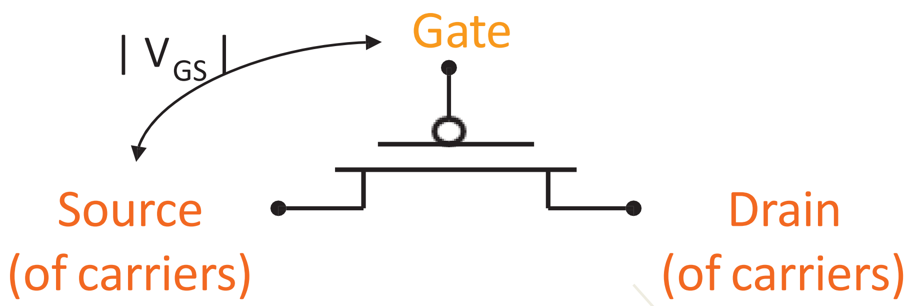


UNIT IV
LOGIC DESIGN AND IMPLEMENTATION STRATEGIES

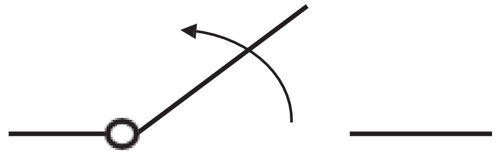
Switch Model of NMOS Transistor



Switch Model of PMOS Transistor

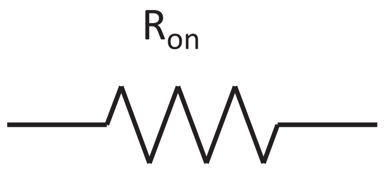


Open (off) (Gate = '1')



$|V_{GS}| > |V_{DD} - |V_T||$

Closed (on) (Gate = '0')

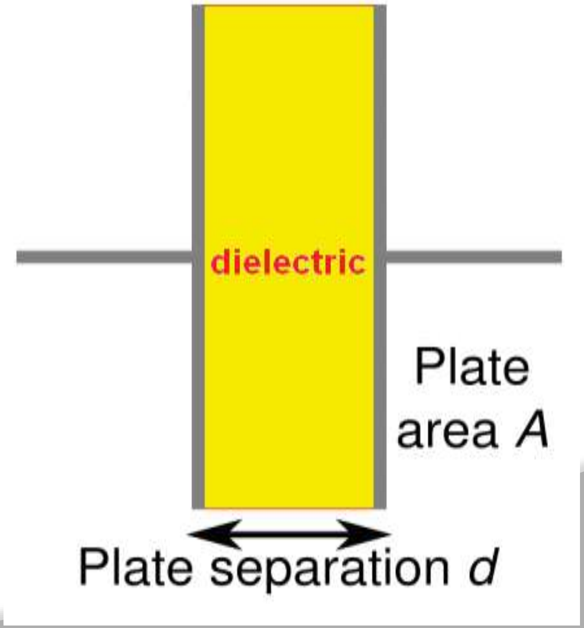


$|V_{GS}| < |V_{DD} - |V_T||$

Capacitors

- Composed of two conductive plates separated by an insulator (or dielectric).
  - Commonly illustrated as two parallel metal plates separated by a distance, d.

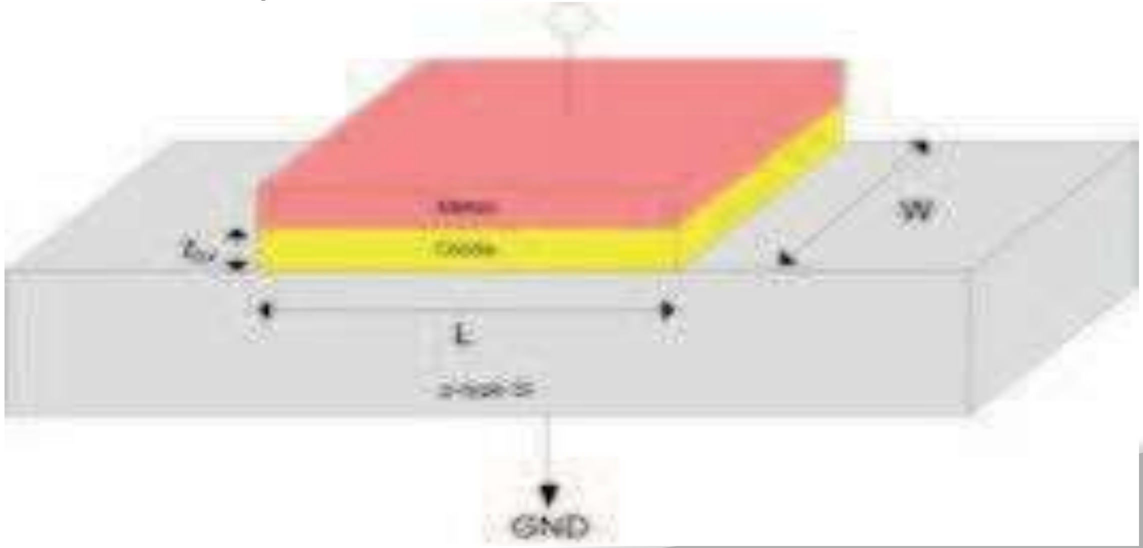
  - C = e A/d
  - where e = er eo
  - er is the relative dielectric constant
  - eo is the vacuum permittivity



CMOS Capacitors

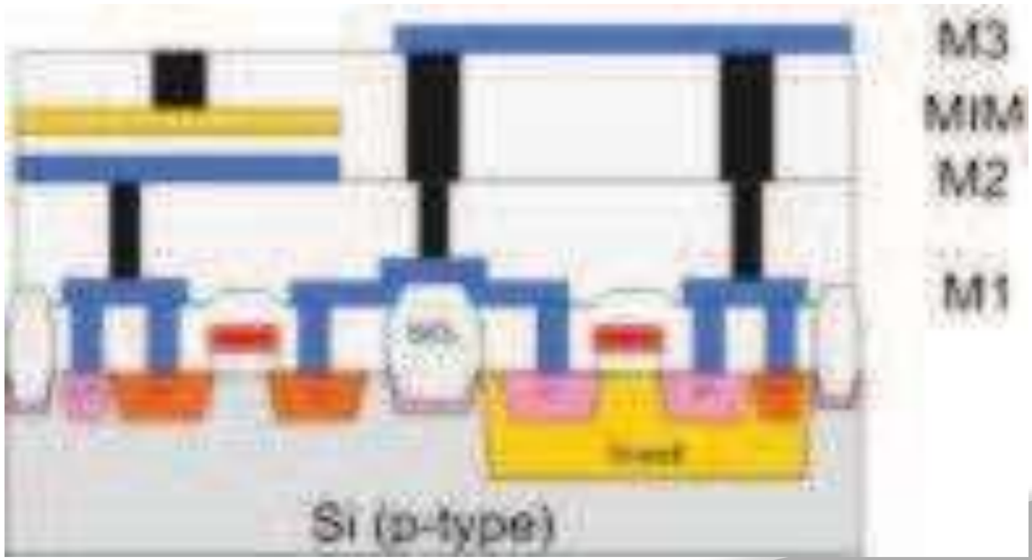
-There are 3 common ways to make a capacitor

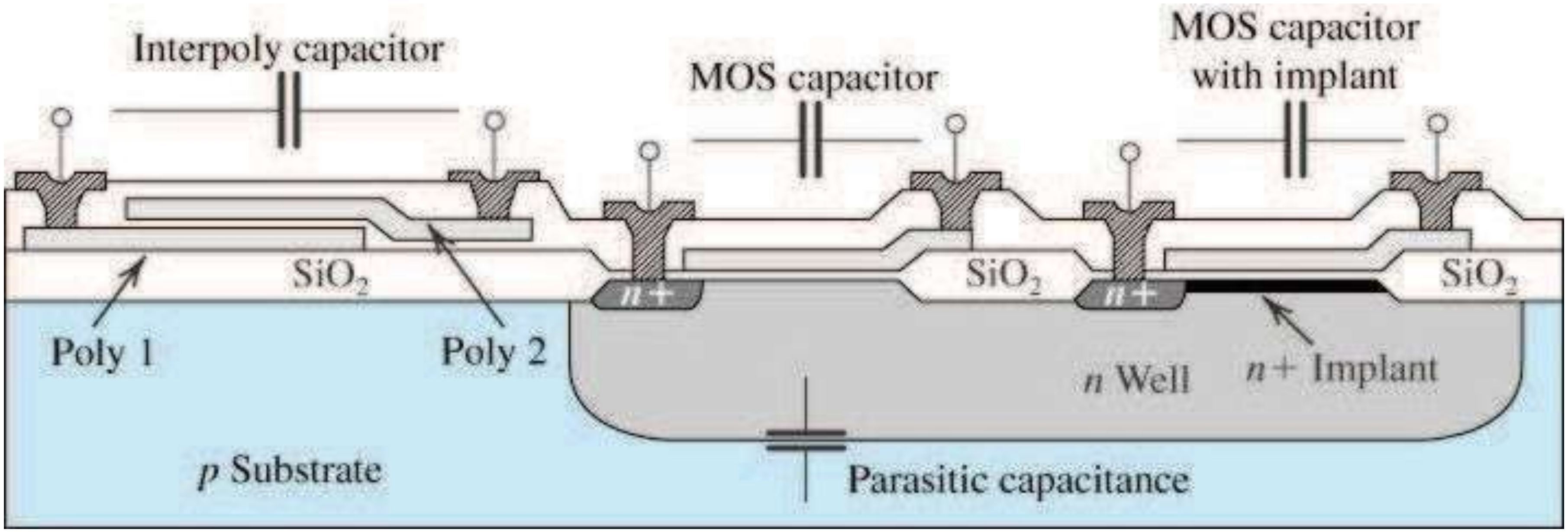
1) MOS Capacitor:-simply create a MOS structure where the Gate (Metal) terminal is one terminal and the Body (Semiconductor) terminal is Ground - while this is easy to implement, the capacitance changes with the bias voltage (i.e., VG) due to the depletion and inversion which occurs



MIM Capacitor

-"Metal Insulator Metal"-this is simply a parallel plate capacitor using two metals and an insulator
-This type of capacitor is created using an extra process step that puts in an additional metal layer that can be very close to one of the other Metal layers to get a smaller plate-to-plate separation
-Since the plates are made of metal, the capacitance doesn't change with bias voltage-these capacitors are not as large as MOS capacitors





Interpoly and MOS capacitors in an *n*-well CMOS process.

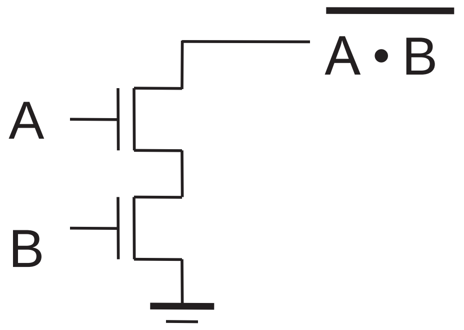
LOGIC DESIGN AND IMPLEMENTATION STRATEGIES

Topics

- Logic gates and other complex gates
- Switch logic
- Alternate gate circuits
- Time delays
- Driving large capacitive loads
- Wiring capacitances
- Fan-in and fan-out, Choice of layers

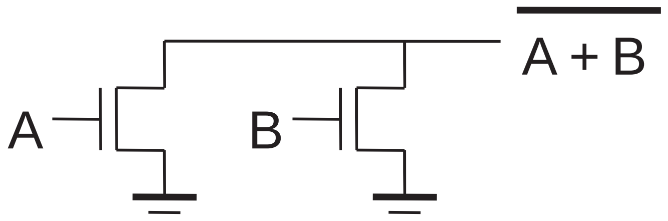
NMOS Gate construction

•NMOS devices in series implement a NAND function



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

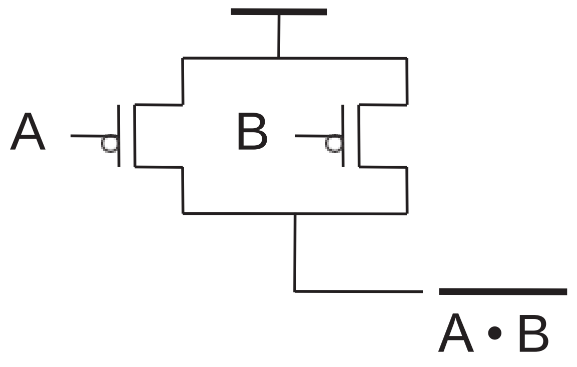
•NMOS devices in parallel implement a NOR function



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

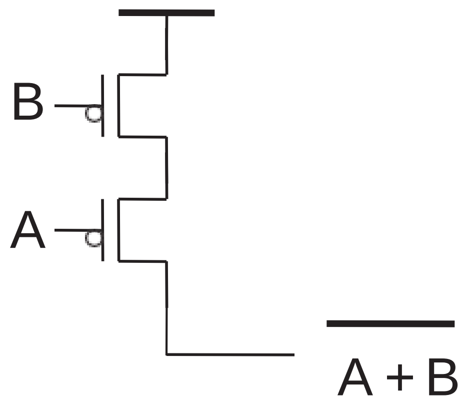
PMOS Gate construction

•PMOS devices in parallel implement a NAND function



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

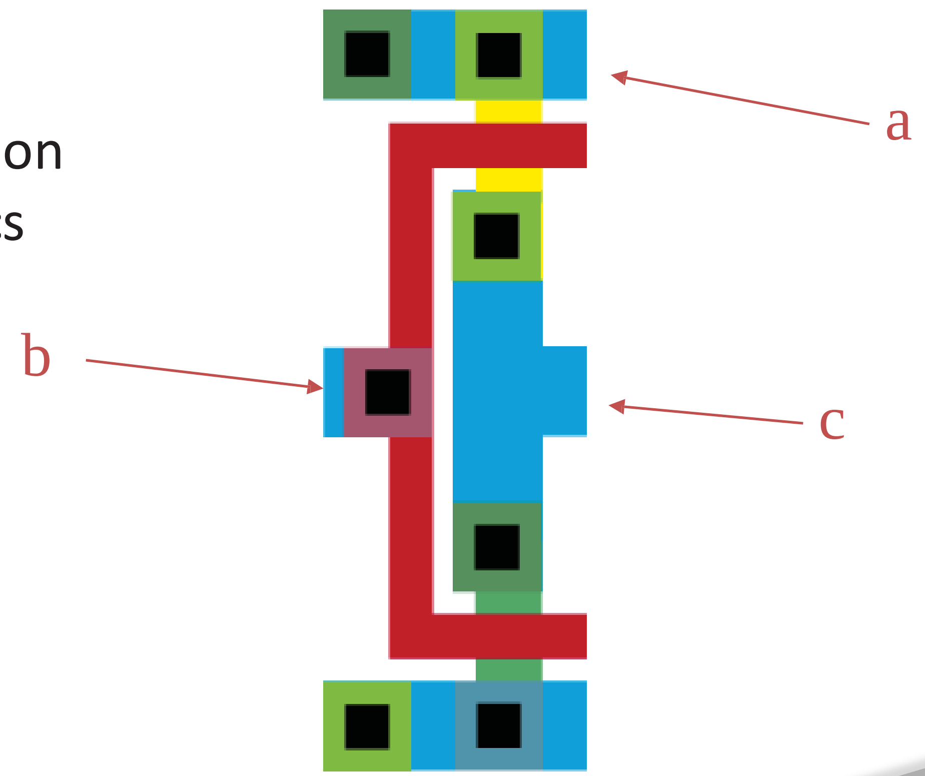
•PMOS devices in series implement a NOR function



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

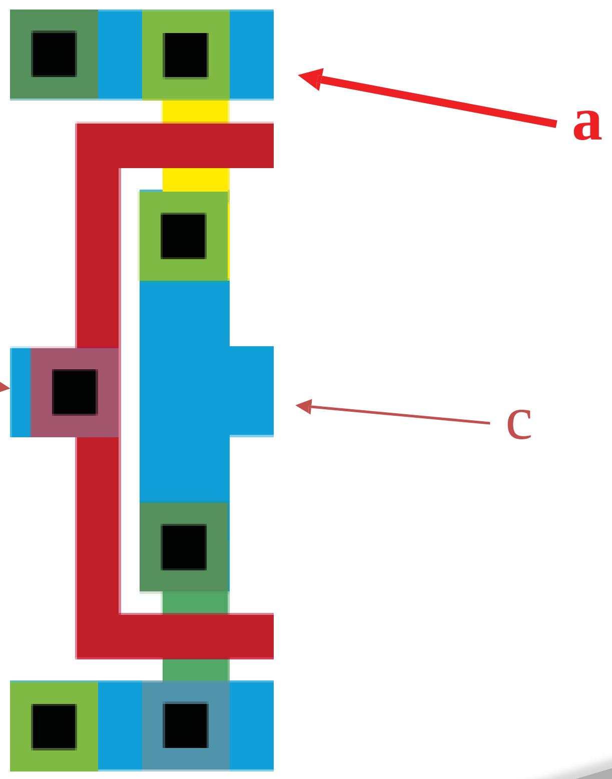
Parasitics and Performance

- Consider the following layout:
- What is the impact on performance of parasitics
  - At point a (VDD rail)?
  - At point b (input)?
  - At Point c (output)?



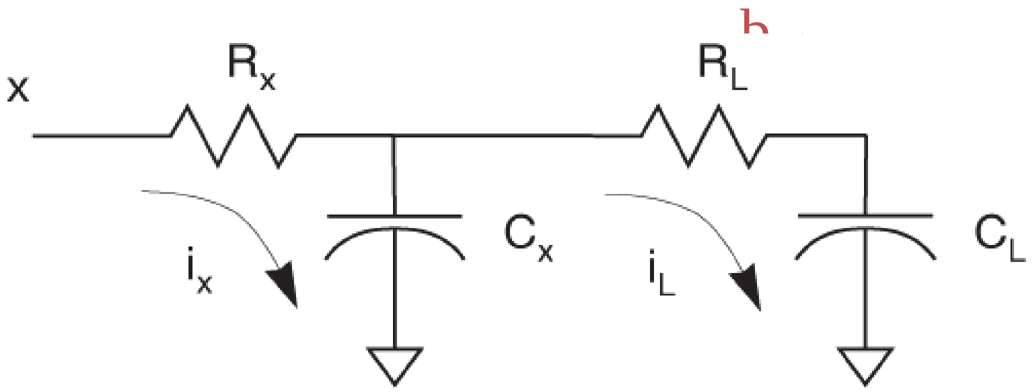
Parasitics and Performance

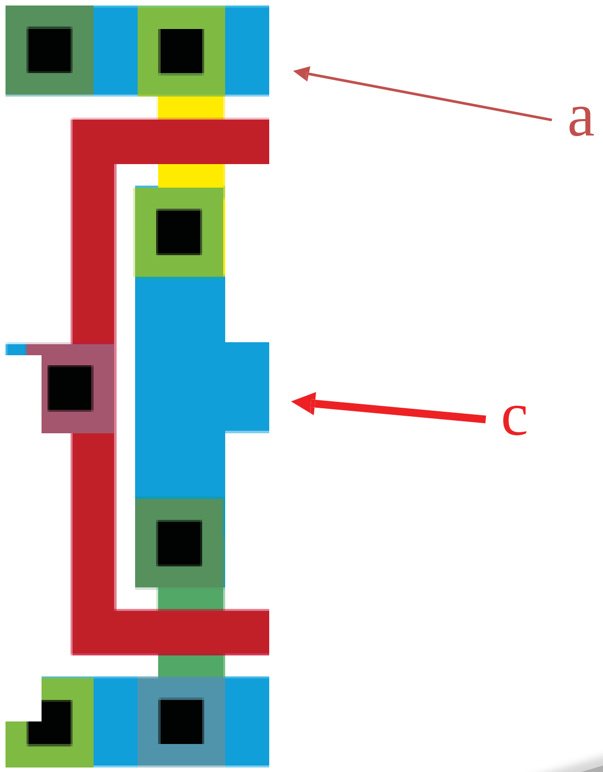
- a - power supply connections
  - capacitance - no effect on delay
  - resistance - increases delay
    - minimize by reducing difffusion length
    - minimize using parallel vias



Parasitics and Performance

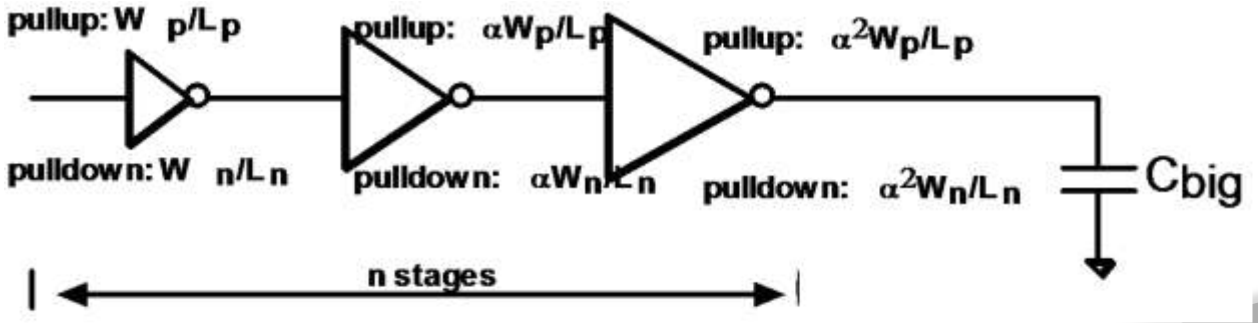
- c - gate output
  - resistance, capacitance increase delay
  - Resistance & capacitance "near" to output causes additional delay





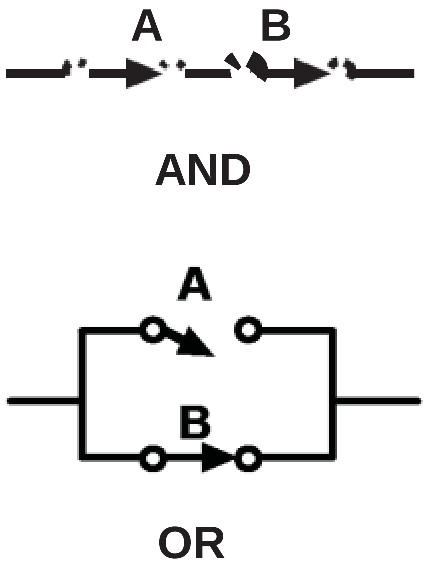
Driving Large Loads

- Off-chip loads, long wires, etc. have high capacitance
- Increasing transistor size increases driving ability (and speed), but in turn increases gate capacitance
- Solution: stages of progressively larger transistors
  - Use nopt = ln(Cbig/Cg).
  - Scale by a factor of a=e



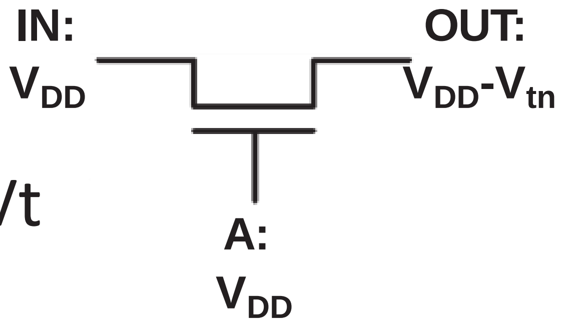
Switch Logic

- Key idea: use transistors as switches
- Concern: switches are bidirectional



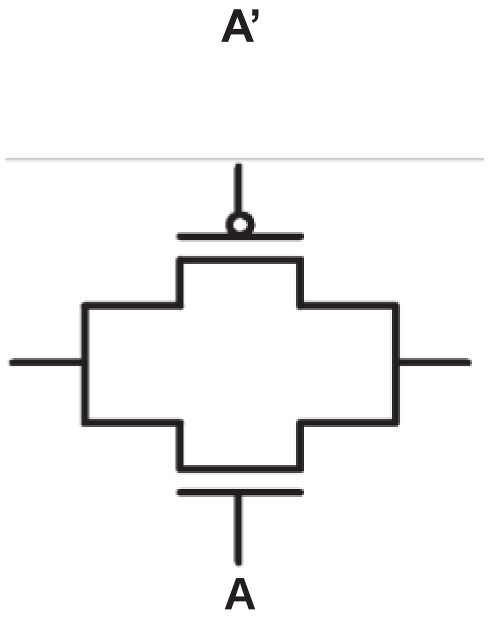
Switch Logic - Pass Transistors

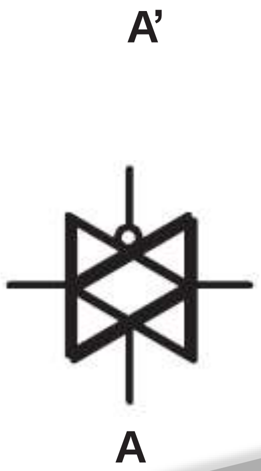
- Use n-transistor as "switches"
- "Threshold problem"
  - Transistor switches off when Vgs < Vt
  - VDD input -> VDD-Vt output
- Special gate needed to "restore" values



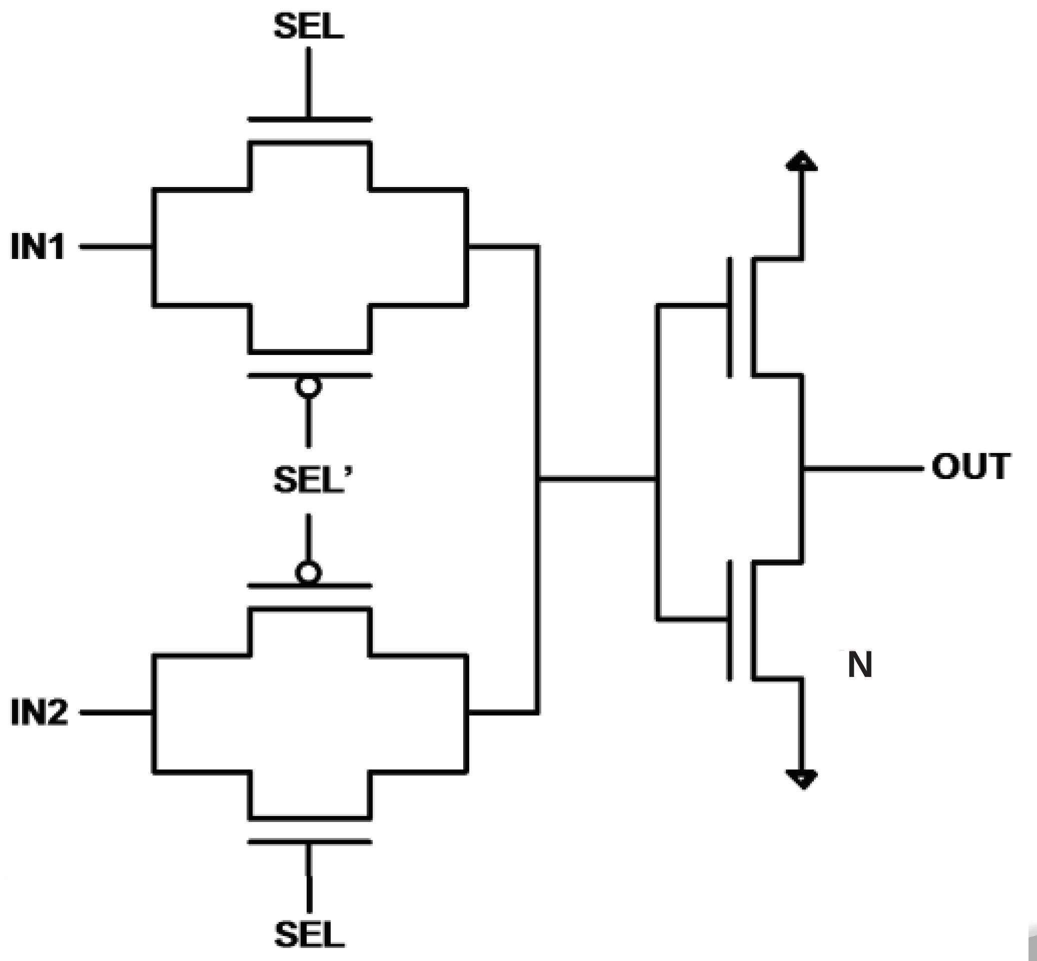
Switch Logic - Transmission Gates

- Complementary transistors - n and p
- No threshold problem
- Cost: extra transistor, extra control input
- Not a perfect conductor!



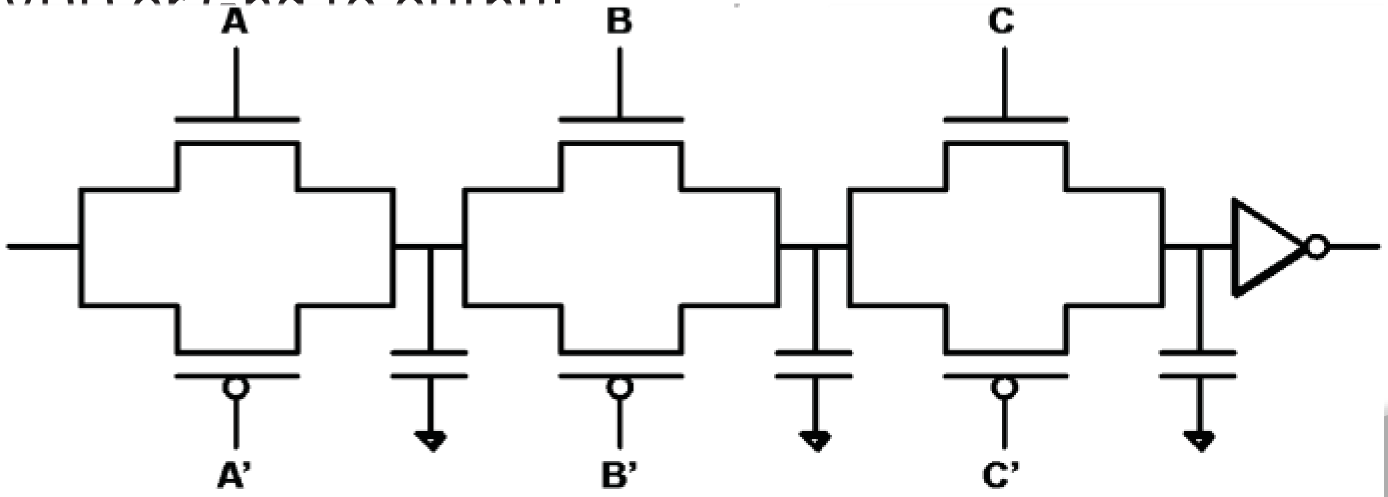


Switch Logic Example - 2-1 MUX



Charge Sharing

- Consider transmission gates in series
  - Each node has parasitic capacitances
  - Problems occur when inputs change to redistribute charge
  - Solution: design network so there is always a path from VDD or Gnd to output

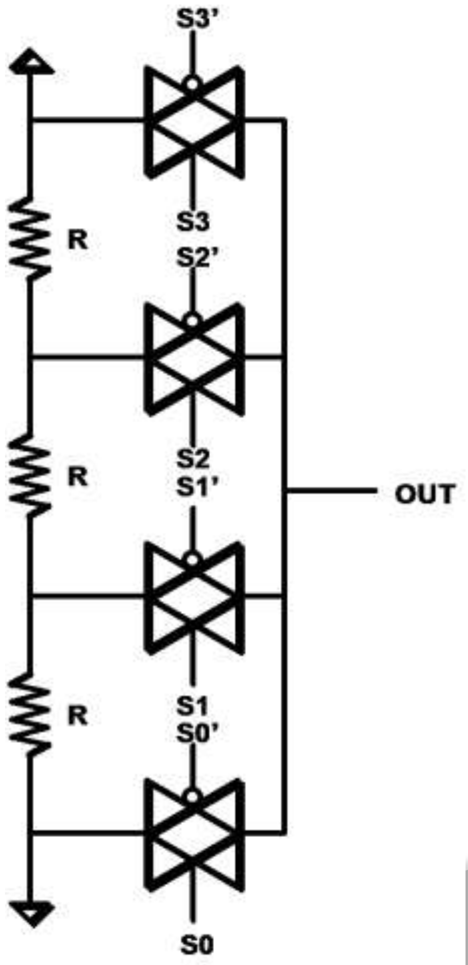


Transmission Gates

- Transmission Gates

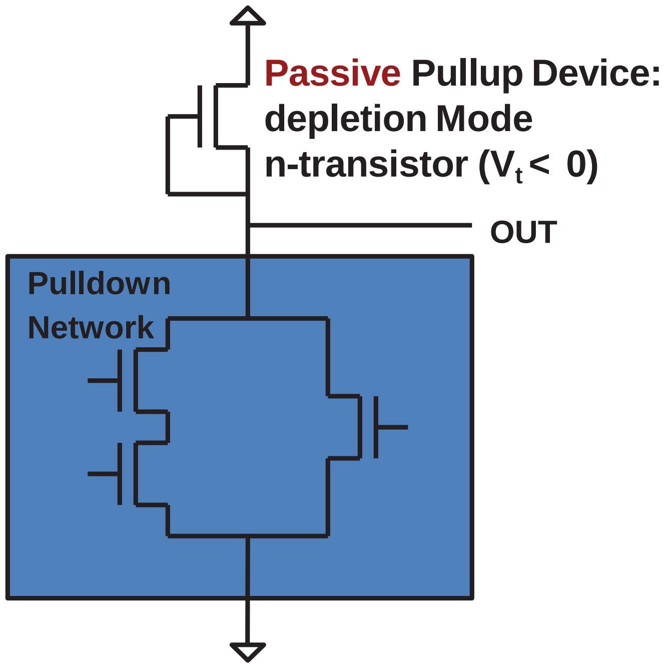
work with analog values, too!

- Example:
  Voltage-Scaling D/A Converter



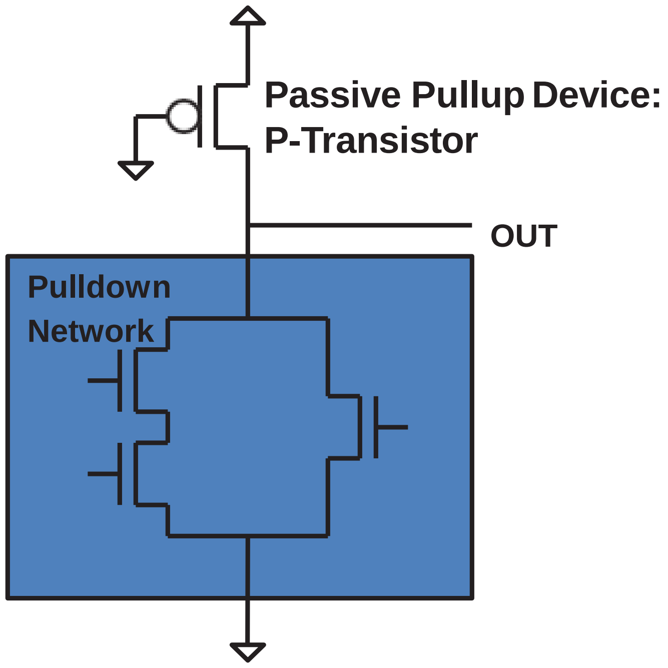
NMOS Logic

- Used before CMOS was widely available
- Uses only n transistors
  - Normal n transistors in pull-down network
  - depletion-mode n transistor (Vt < 0) used for pull-up
  - "ratioed logic" required
- Tradeoffs:
  - Simpler processing
  - Smaller gates
  - higher power!
  - Additional design considerations for ratioed logic


Passive Pullup Device: depletion Mode n-transistor ($V_t <$ 0)
OUT
Pulldown Network

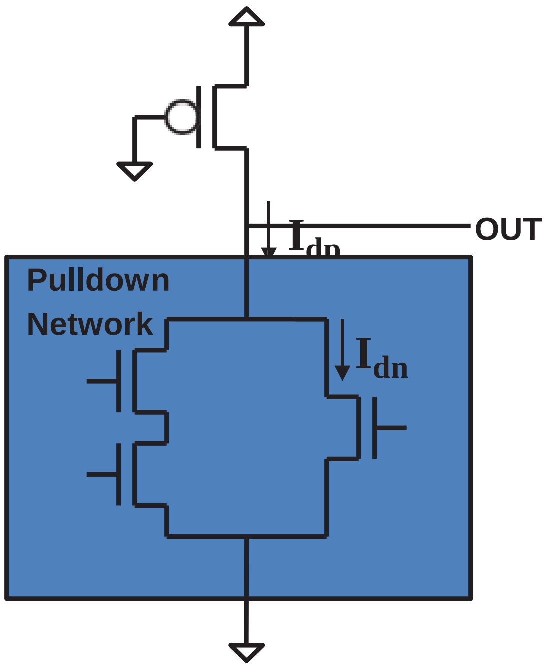
Pseudo-nMOS Logic

- Same idea, as nmos, but use p-transistor for pullup
- "ratioed logic" required for proper design (more about this next)
- Tradeoffs:
  - Fewer transistors -> smaller gates, esp. for large number of inputs
  - less capacitative load on gates that drive inputs
  - larger power consumption
  - less noise margin (VOL > 0)
  - additional design considerations due to ratioed logic



Rationed Logic for Pseudo-nmos

- Approach:
  - Assume VOUT=VOL =0.25*VDD
  - Assume 1 pulldown transistor is on
  - Equate currents in p, n transistors
  - Solve for ratio between sizes of p, n transistors to get these conditions
  - Further calculations necessary for series connections



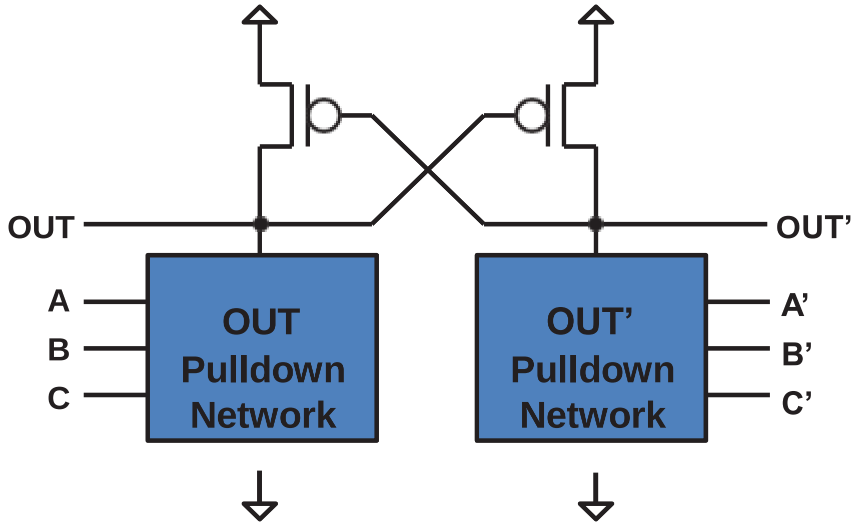
$$I_{dn} = I_{pn}$$

$$\frac{1}{2} k'_n \frac{W_n}{L_n} \left(V_{gs,n} - V_{tn}\right)^2 = \frac{1}{2} k'_p \frac{W_p}{L_p} \left[2\left(V_{gs,p} - V_{tp}\right)V_{ds,p} - V_{ds,p}^2\right] \quad (EQ\ 3-21)$$

$$\frac{W_p / L_p}{W_n / L_n} = 3.9 \qquad (EQ\ 3-22) - \text{Assuming } V_{DD} = 3.3V$$

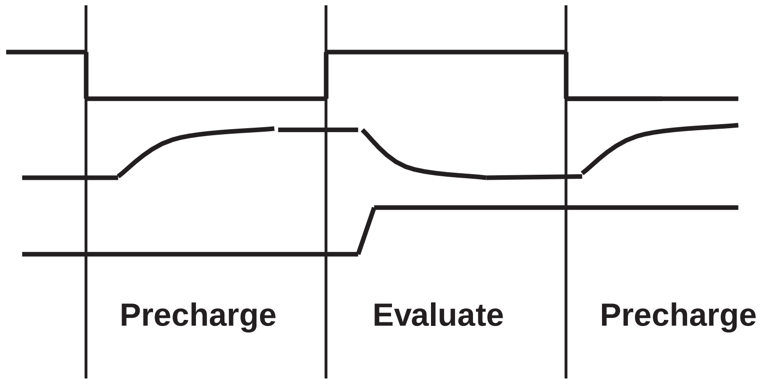
DCVS Logic

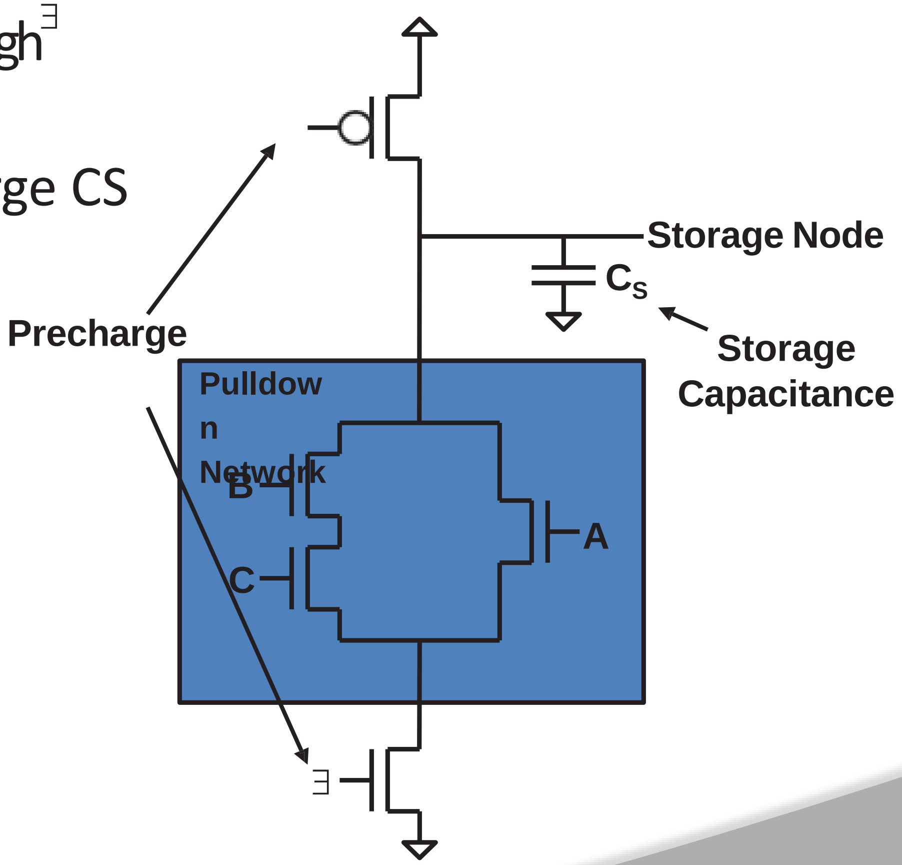
- DCVS - Differential Cascode Voltage Switch
- Differential inputs, outputs
- Two pulldown networks
- Tradeoffs
  - Lower capacitative loading than static CMOS
  - No ratioed logic needed
  - Low static power consumption
  - More transistors
  - More signals to route between gates



Dynamic Logic

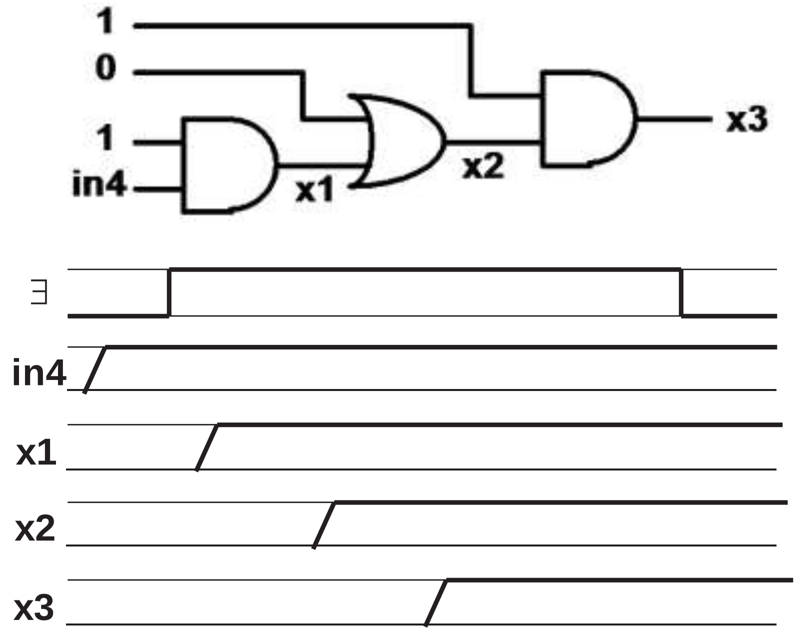
- Key idea: Two-step operation
  - precharge - charge CS to logic high ∃
  - evaluate - conditionally discharge CS
- Control - precharge clock f Signal

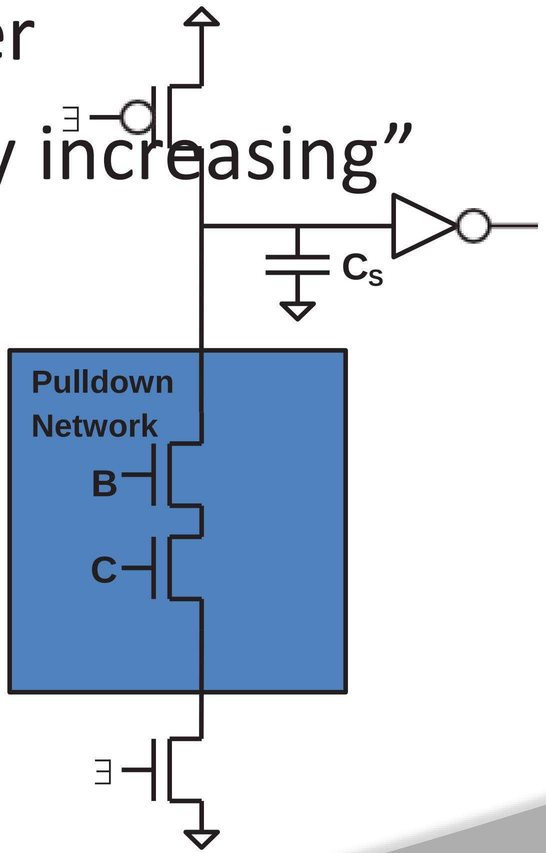




Domino Logic

- Key idea: dynamic gate + inverter
- Cascaded gates - “monotonically increasing”





Domino Logic Tradeoffs

- Fewer transistors -> smaller gates
- Lower power consumption than pseudo-nmos
- Clocking required
- Logic not complete (AND, OR, but no NOT)

Types of ASICs

- A full-custom IC includes some (possibly all) logic cells that are customized and all mask layers that are customized. A microprocessor is an example of a full-custom IC—designers spend many hours squeezing the most out of every last square micron of microprocessor chip space by hand. It allows designers to include analog circuits, optimized memory cells. Full-custom ICs are the most expensive to manufacture and to design. The manufacturing lead time (the time it takes just to make an IC—not including design time) is typically eight weeks for a full-custom IC.
- Semicustom ASICs , in which all of the logic cells are predesigned and some (possibly all) of the mask layers are customized. Using predesigned cells from a cell library makes our lives as designers much, much easier. There are two types of semicustom ASICs:
- standard-cell–based ASICs & gate-array–based ASICs.
- programmable ASICs ,in which all of the logic cells are predesigned and none of the mask layers are customized. There are two types of programmable ASICs: the programmable logic device and, the newest member of the ASIC family, the field-programmable gate array.

Full-Custom ASICs

- In a full-custom ASIC an engineer designs some or all of the logic cells, circuits, or layout specifically for one ASIC. This means the designer abandons the approach of using pretested and precharacterized cells for all or part of that design.
- It makes sense to take this approach only if there are no suitable existing cell libraries available that can be used for the entire design. This might be because existing cell libraries are not fast enough, or the logic cells are not small enough or consume too much power.
- You may need to use full-custom design if the ASIC technology is new or so specialized that there are no existing cell libraries or because the ASIC is so specialized that some circuits must be custom designed.
- Fewer and fewer full-custom ICs are being designed because of the problems with these special parts of the ASIC. There is one growing member of this family, though, the mixed analog/digital ASIC.

Library-Cell Design

•The optimum cell layout for each process generation changes because the design rules for each ASIC vendor’s process are always slightly different—even for the same generation of technology.
•For example, two companies may have very similar 0.35 m m CMOS process technologies, but the third-level metal spacing might be slightly different.

• If a cell library is to be used with both processes, we could construct the library by adopting the most stringent rules from each process.

• A library constructed in this fashion may not be competitive with one that is constructed specifically for each process.

•Even though ASIC vendors prize their design rules as secret, it turns out that they are similar—except for a few details. Unfortunately, it is the details that stop us moving designs from one process to another.
•Unless we are a very large customer it is difficult to have an ASIC vendor change or waive design rules for us. We would like all vendors to agree on a common set of design rules.
• This is, in fact, easier than it sounds. The reason that most vendors have similar rules is because most vendors use the same manufacturing equipment and a similar process.
•It is possible to construct a highest common denominator library that extracts the most from the current manufacturing capability.
• Some library companies and the large Japanese ASIC vendors are adopting this approach.

Library-Cell Design, Cont

•Each standard cell in a library is rectangular with the same height but different widths. The bounding box (BB) of a logic cell is the smallest rectangle that encloses all of the geometry of the cell. The cell BB is normally determined by the well layers.

•Cell connectors or terminals (the logical connectors) must be placed on the cell abutment box (AB). The physical connector (the piece of metal to which we connect wires) must normally overlap the abutment box slightly, usually by at least 1 l , to assure connection without leaving a tiny space between the ends of two wires.

•The standard cells are constructed so they can all be placed next to each other horizontally with the cell ABs touching (we abut two cells).

•When a library developer creates a gate-array, standard-cell, or datapath library, there is a trade-off between using wide, high-drive transistors that result in large cells with high-speed performance and using smaller transistors that result in smaller cells that consume less power.

•A performance-optimized library with large cells might be used for ASICs in a high-performance workstation, for example. An area-optimized library might be used in an ASIC for a battery-powered portable computer.

Gate-Array Design

•Each logic cell or macro in a gate-array library is predesigned using fixed tiles of transistors known as the gate-array base cell (or just base cell). We call the arrangement of base cells across a whole chip in a complete gate array the gate-array base.

• ASIC vendors offer a selection of bases, with a different total numbers of transistors on each base. For example, if our ASIC design uses 48k equivalent gates and the ASIC vendor offers gate arrays bases with 50k-, 75k-, and 100k-gates, we will probably have to use the 75k-gate base it is unlikely 96 percent of the transistors on the 50k-gate base).

•We isolate the transistors on a gate array from one another either with thick field oxide (in the case of oxide-isolated gate arrays) or by using other transistors that are wired permanently off (in gate-isolated gate arrays). Figure 3.14 (a) shows a base cell for a gate-isolated gate array .

•This base cell has two transistors: one p -channel and one n -channel. When these base cells are placed next to each other, the n -diffusion and p -diffusion layers form continuous strips that run across the entire chip broken only at the poly gates that cross at regularly spaced intervals (Figure 3.14b).

• The metal interconnect spacing determines the separation of the transistors. The metal spacing is determined by the design rules for the metal and contacts.

Gate-Array Design

•In Figure 3.14 (c) we have shown all possible locations for a contact in the base cell. There is room for 21 contacts in this cell and thus room for 21 interconnect lines running in a horizontal direction (we use m1 running horizontally). We say that there are 21 horizontal tracks in this cell or that the cell is 21 tracks high.

• In a similar fashion the space that we need for a vertical interconnect (m2) is called a vertical track . The horizontal and vertical track widths are not necessarily equal, because the design rules for m1 and m2 are not always equal.

•We isolate logic cells from each other in gate-isolated gate arrays by connecting transistor gates to the supply bus—hence the name, gate isolation .

•If we connect the gate of an n -channel transistor to V_{SS} , we isolate the regions of n -diffusion on each side of that transistor (we call this an isolator transistor or device, or just isolator). Similarly if we connect the gate of a p -channel transistor to V_{DD} , we isolate adjacent p -diffusion regions.

Gate-Array Design

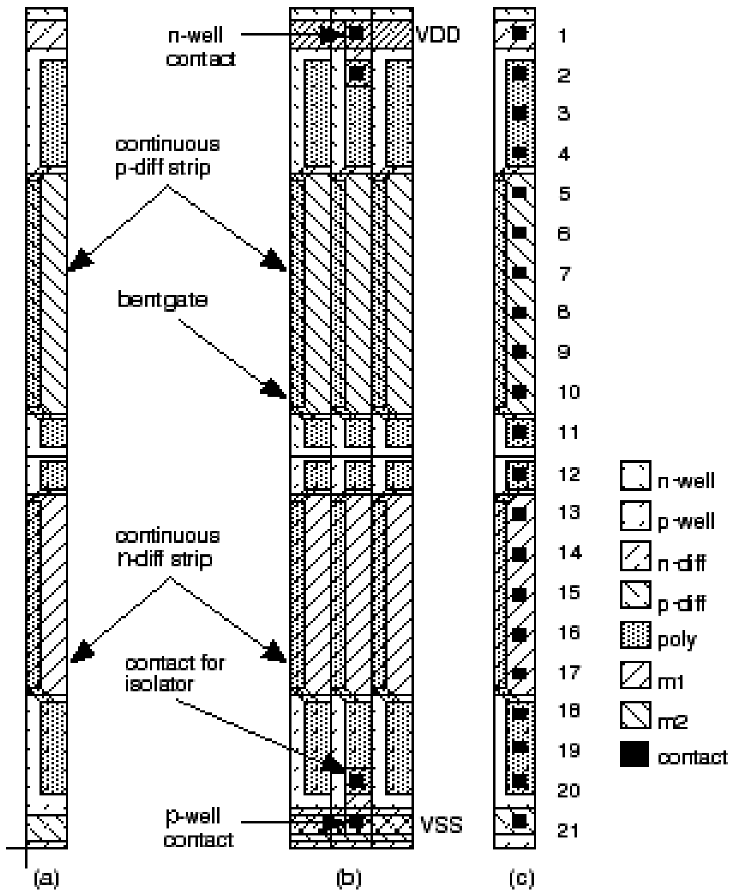


FIGURE 3.14 The construction of a gate-isolated gate array. (a) The one-track-wide base cell containing one p -channel and one n -channel transistor.
(b) Three base cells: the center base cell is being used to isolate the base cells on either side from each other. (
c) A base cell including all possible contact positions (there is room for 21 contacts in the vertical direction, showing the base cell has a height of 21 tracks).

Gate-Array Design

•FIGURE 3.15 An oxide-isolated gate-array base cell. The figure shows two base cells, each containing eight transistors and two well contacts.

•The p -channel Oxide-isolated gate arrays often contain four transistors in the base cell: the two n –channel transistors share an n -diffusion strip and the two p -channel transistors share a p -diffusion strip. This means that the two n -channel transistors in each base cell are electrically connected in series, as are the p -channel transistors.

• The base cells are isolated from each other using oxide isolation . During the fabrication process a layer of the thick field oxide is left in place between each base cell and this separates the p -diffusion and n -diffusion regions of adjacent base cells.
•Figure 3.15 shows an oxide-isolated gate array .

•This cell contains eight transistors (which occupy six vertical tracks) plus one-half of a single track that contains the well contacts and substrate connections that we can consider to be shared by each base cell,and n -channel transistors are each 4 tracks high (corresponding to the width of the transistor).

•The leftmost vertical track of the left base cell includes all 12 possible contact positions (the height of the cell is 12 tracks). As outlined here, the base cell is 7 tracks wide (we could also consider the base cell to be half this width).

•Figure 3.16 shows a base cell in which the gates of the n -channel and p -channel transistors are connected on the polysilicon layer.

•Connecting the gates in poly saves contacts and a metal interconnect in the center of the cell where interconnect is most congested.

• The drawback of the preconnected gates is a loss in flexibility in cell design. Implementing memory and logic based on transmission gates will be less efficient using this type of base cell, for example.

Gate-Array Design

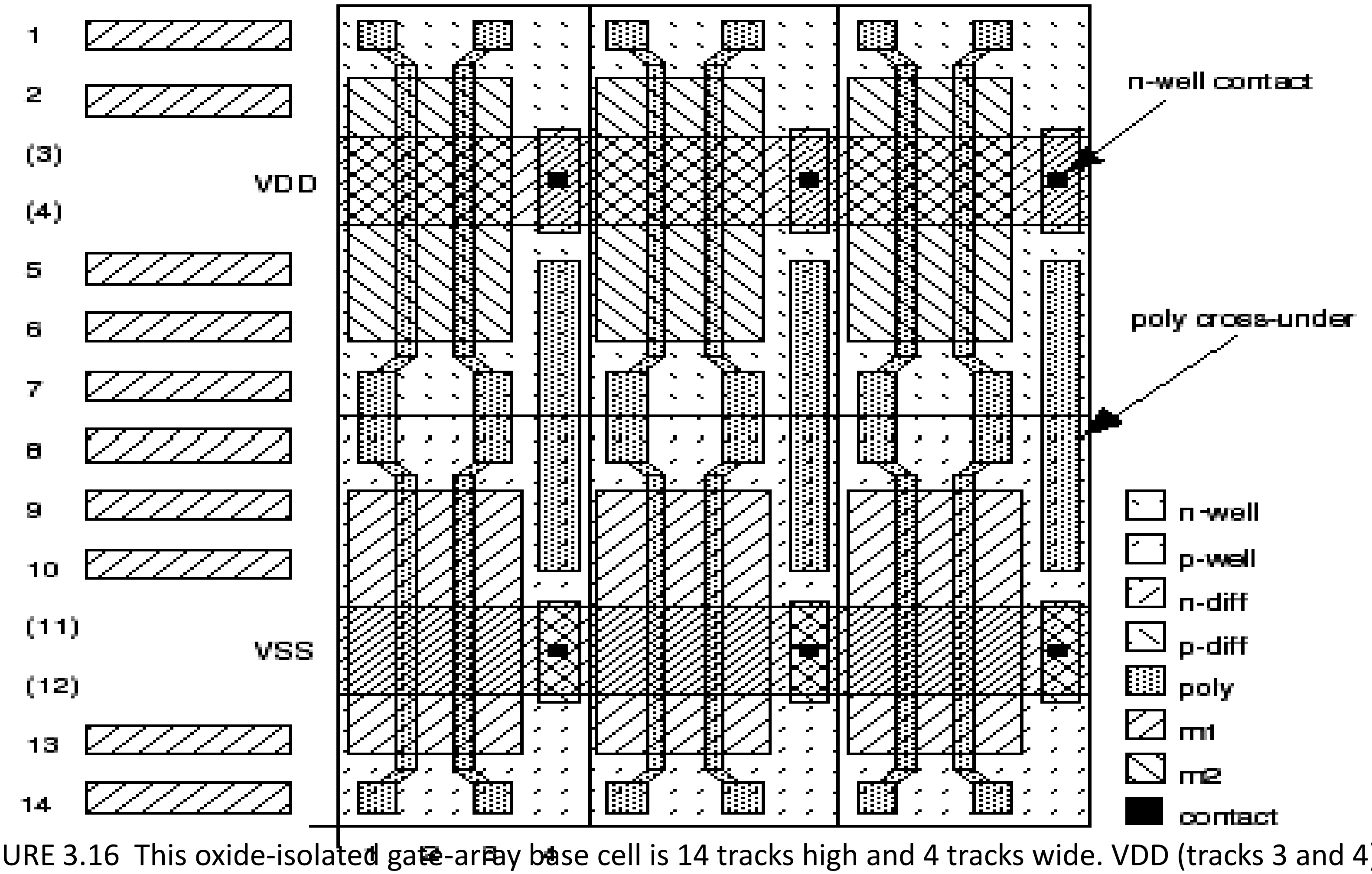


FIGURE 3.16 This oxide-isolated gate-array base cell is 14 tracks high and 4 tracks wide. VDD (tracks 3 and 4) and GND (tracks 11 and 12) are each 2 tracks wide. The metal lines to the left of the cell indicate the 10 horizontal routing tracks (tracks 1, 2, 5–10, 13, 14). Notice that the p -channel and n -channel polysilicon gates are tied together in the center of the cell. The well contacts are short, leaving room for a poly cross-under in each base cell.

Programmable ASICs

There are two types of programmable ASICs: programmable logic devices (PLDs) and field-programmable gate arrays (FPGAs).

PLDs started as small devices that could replace a handful of TTL parts, and they have grown to look very much like their younger relations, the FPGAs. designer, can program yourself.

An IC foundry produces FPGAs with some connections missing. You perform design entry and simulation.

Next, special software creates a string of bits describing the extra connections required to make your design—the configuration file . You then connect a computer to the chip and program the chip to make the necessary connections according to the configuration file.

There is no customization of any mask level for an FPGA, allowing the FPGA to be manufactured as a standard part in high volume.

FPGAs are popular with microsystems designers because they fill a gap between TTL and PLD design and modern, complex, and often expensive ASICs. FPGAs are ideal for prototyping systems or for low-volume production.

programming technology

The programming technology may or may not be permanent. You cannot undo the permanent programming in one-time programmable (OTP) FPGAs. Reprogrammable or erasable devices may be reused many times. programming technologies:

The Antifuse

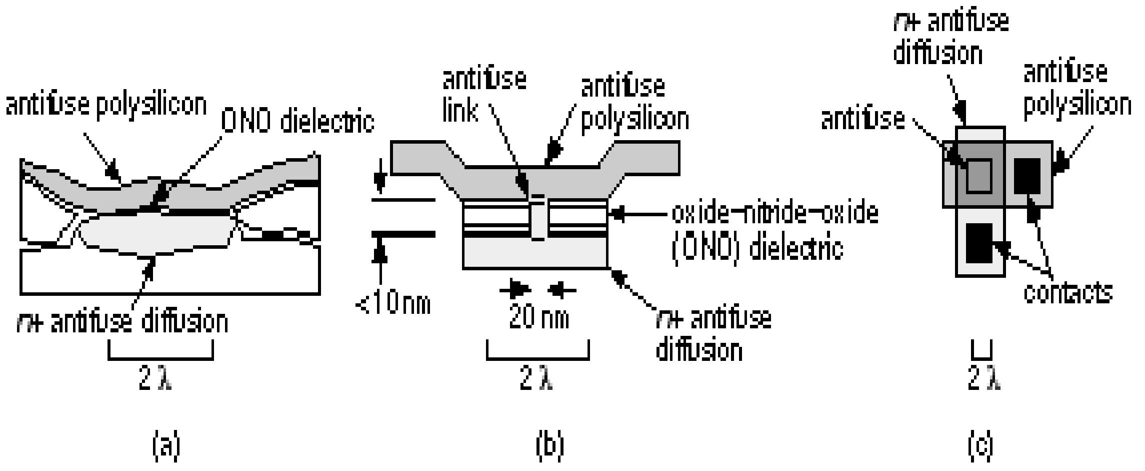
The Static RAM

The EPROM &EEPROM

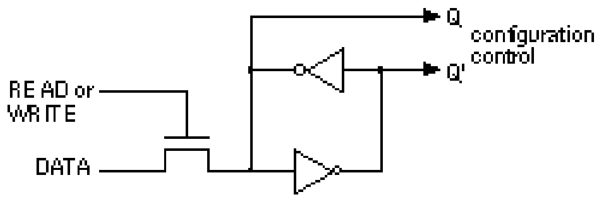
The Xilinx SRAM (static RAM) configuration cell. The outputs of the cross-coupled inverter (configuration control) are connected to the gates of pass transistors or transmission gates. The cell is programmed using the WRITE and DATA lines.

programming technology

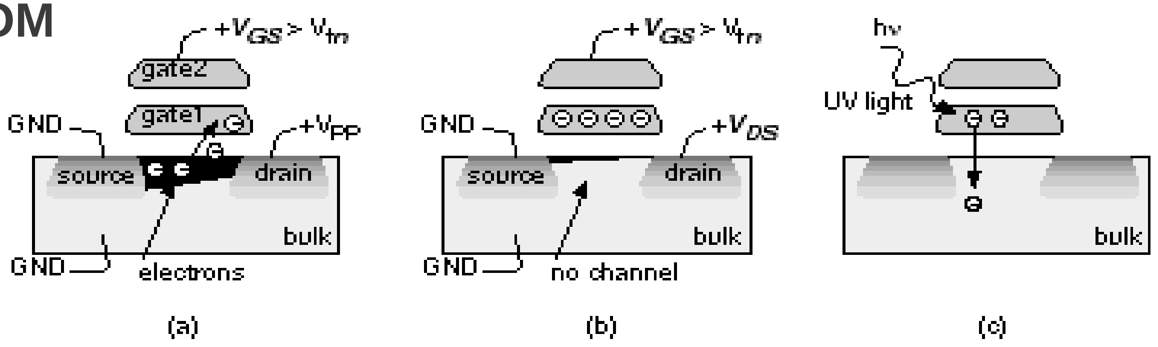
The Antifuse



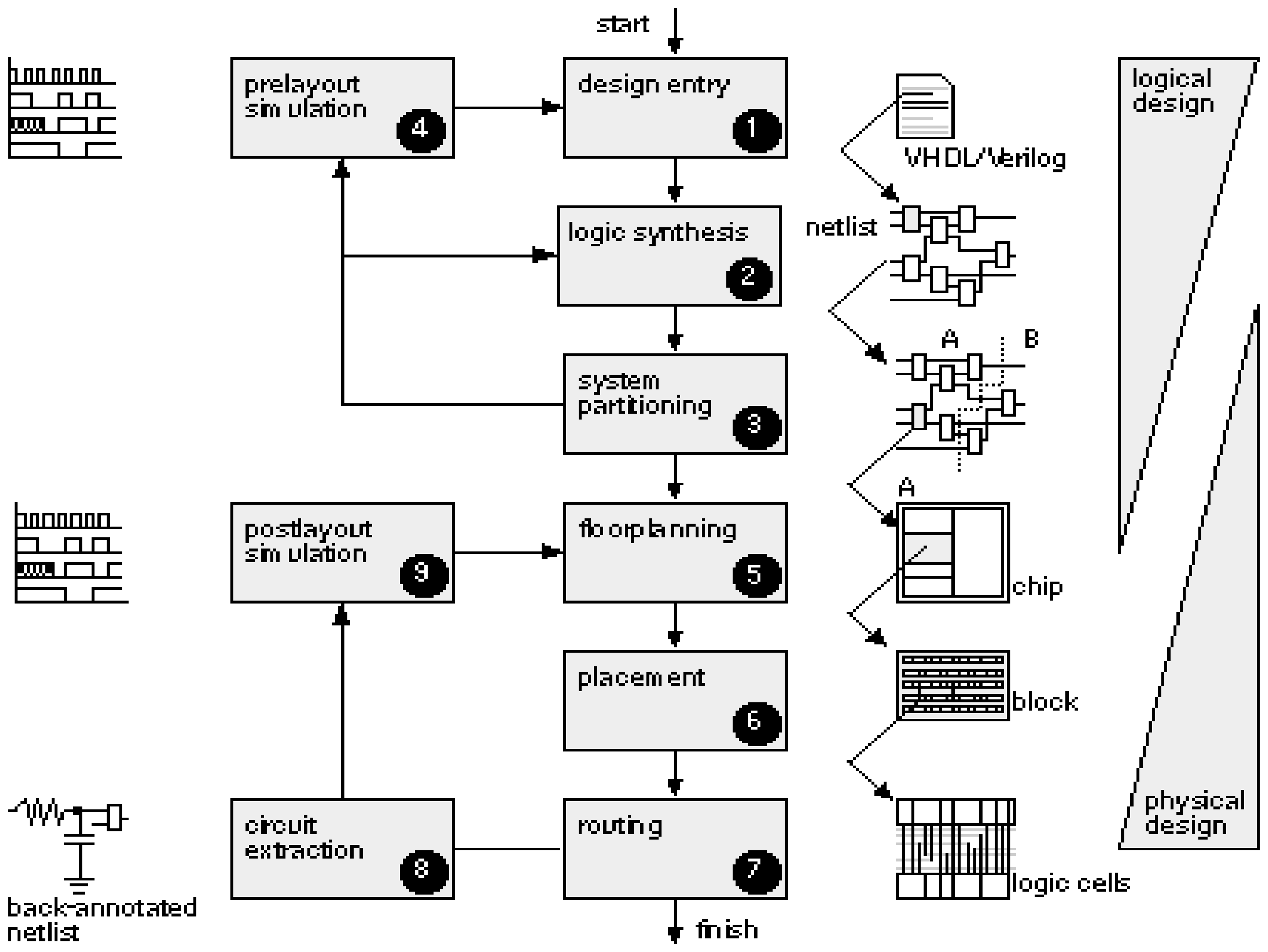
The Static RAM



The EPROM &EEPROM

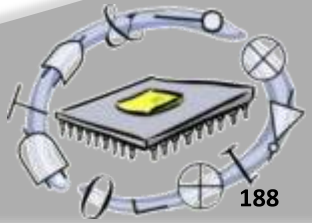


ASIC Design Flow



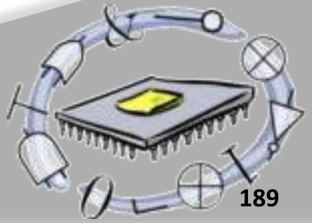
Programmable Logic Device (PLD) Definitions

- Programmable Logic Device (PLD):
  - Also known as “Field Programmable Logic Device (FPLD)”
  - An integrated circuit chip that can be configured by the user to implement different digital hardware.



Purpose of PLDs

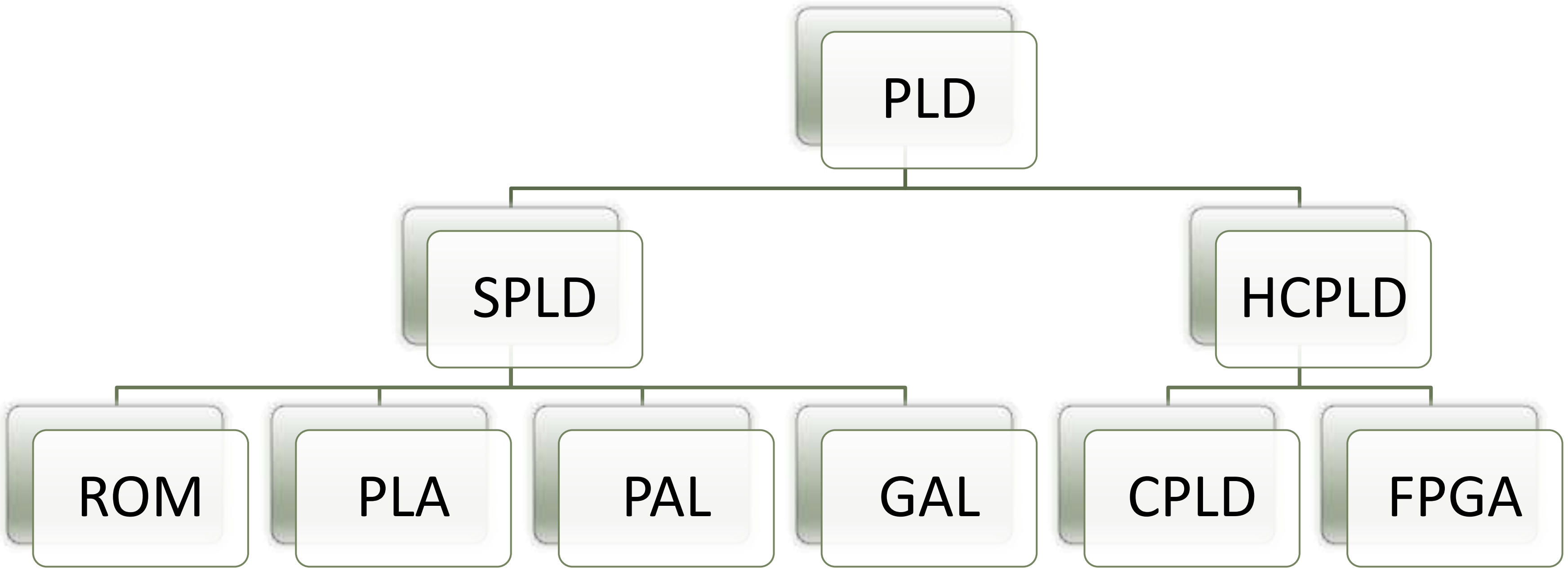
- Permits elaborate digital logic designs to be implemented by the user on a single device.
- Is capable of being erased and reprogrammed with a new design.
- Advantages of PLDs
- Programmability
- Re-programmability
  - PLDs can be reprogrammed without being removed from the circuit board.
- Low cost of design
- Immediate hardware implementation



Types of PLDs

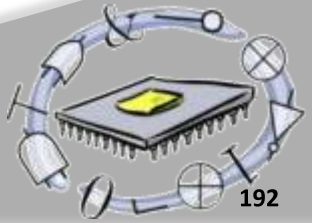
- SPLDs (Simple Programmable Logic Devices)
  - ROM (Read-Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)
  - GAL (Generic Array Logic)
- HCPLD (High Capacity Programmable Logic Device)
  - CPLD (Complex Programmable Logic Device)
  - FPGA (Field-Programmable Gate Array)

Types of PLDs (Cont.)



SPLDs

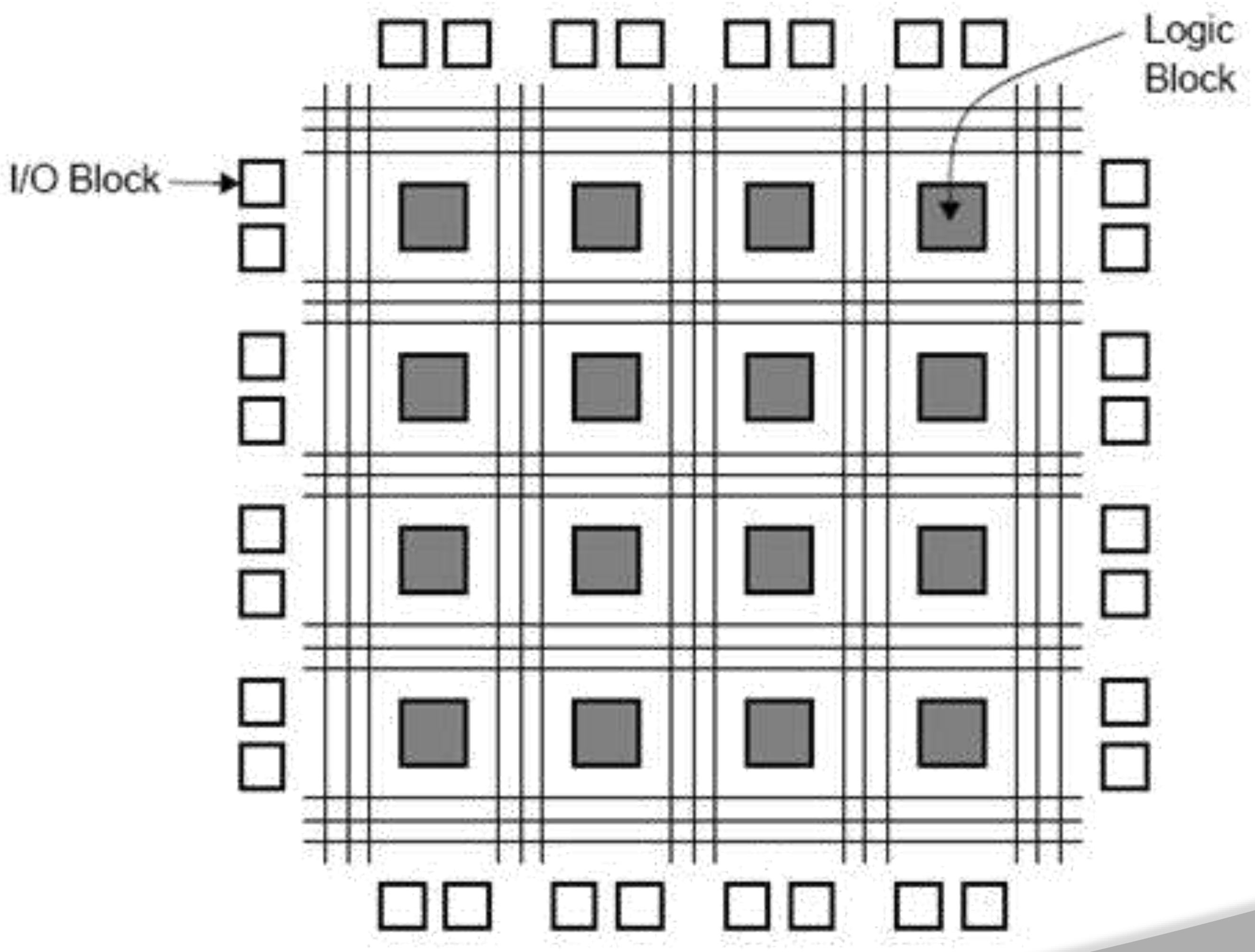
- In ROM, the input connection matrix is hardwired and the user can only modify the output connection matrix.
- In PAL and GAL the output connection matrix is also hardwired and the user can modify the input connection matrix.
- In PLA the user can modify both the input connection matrix and the output connection matrix.



HCPLDs

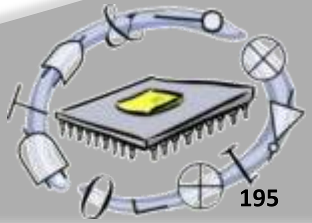
- CPLD (Complex Programmable Logic Device)
  - Lies between PALs and FPGAs in degree of complexity.
  - Inexpensive
- FPGA (Field-Programmable Gate Array)
  - Truly parallel design and operation
  - Fast turnaround design
  - Array of logic cells surrounded by programmable I/O blocks

FPGA Design



PLD Configuration

- Combination of a logic device and memory
- Memory stores the pattern the PLD was programmed with
  - EPROM
    - Non-volatile and reprogrammable
  - EEPROM
    - Non-volatile and reprogrammable
  - Static RAM (SRAM)
    - Volatile memory
  - Flash memory
    - Non-volatile memory
  - Antifuse
    - Non-volatile and no re-programmability



UNIT V
SUB SYSTEM DESIGN

Shift Registers:

- A register consists of a group of flip-flops with a common clock input.
- Registers are commonly used to store and shift binary data.
  - Example: Counters

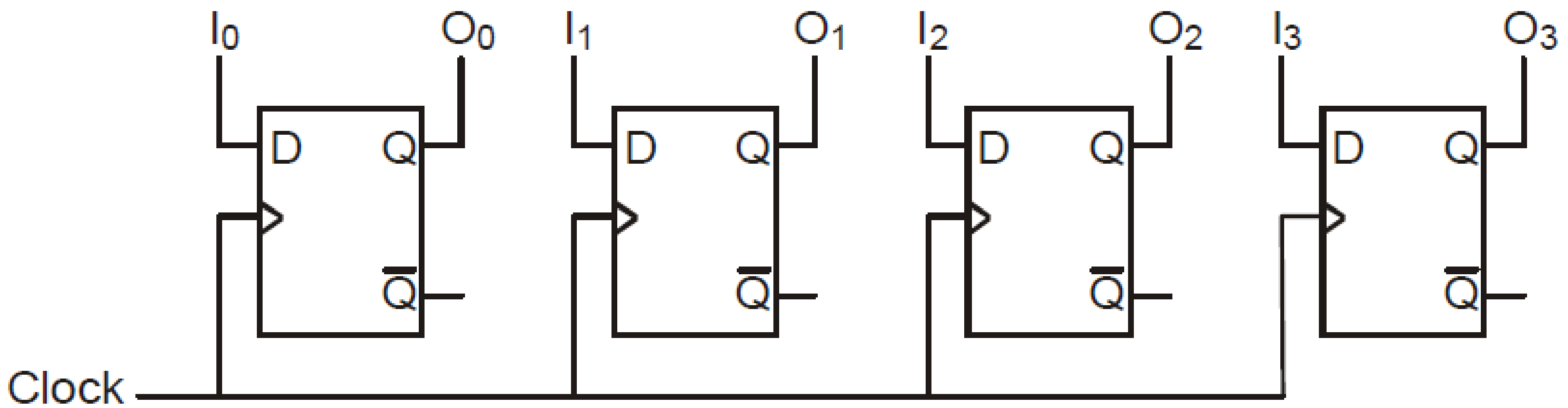
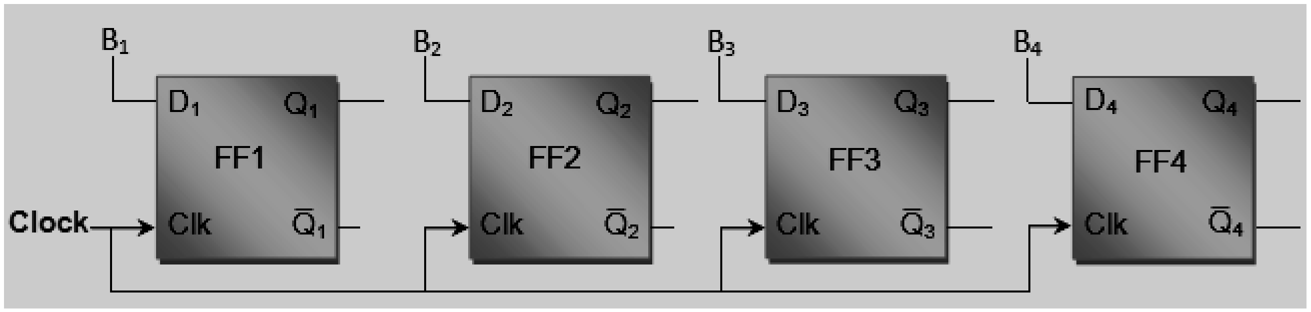


Fig. 4-bit D register

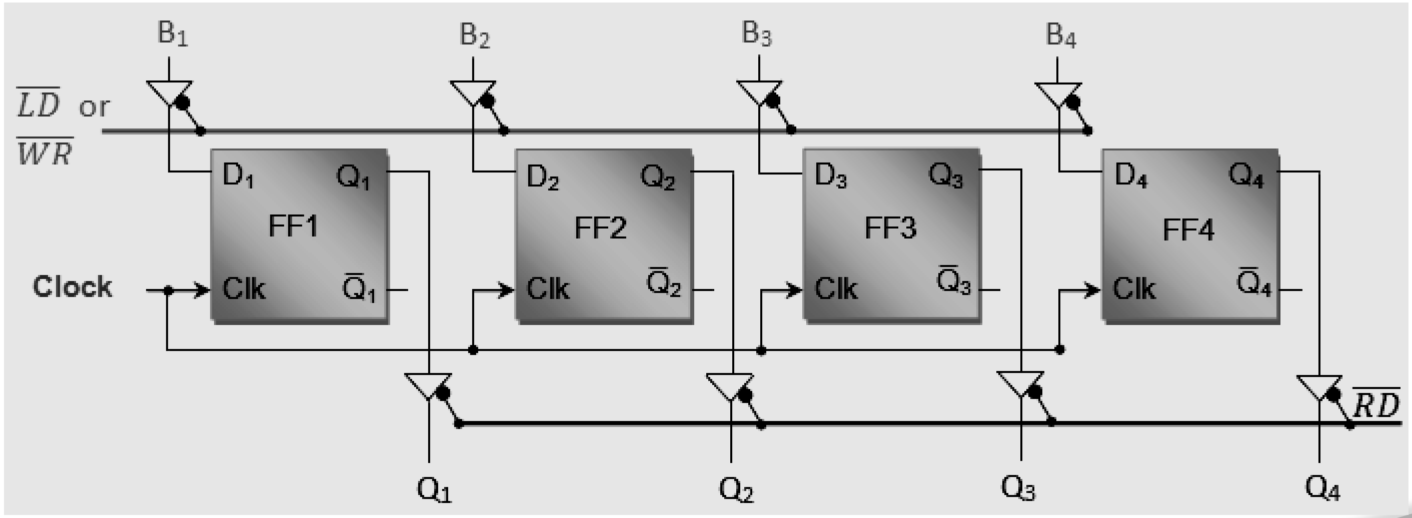
Registers: Types

- Buffer Register.



B_1
B_2
B_3
B_4
D_1 Q_1 FF1 Clk $\bar{Q}_1$
D_2 Q_2 FF2 Clk $\bar{Q}_2$
D_3 Q_3 FF3 Clk $\bar{Q}_3$
D_4 Q_4 FF4 Clk $\bar{Q}_4$
Clock

Fig. 1. 4-bit Buffer register



B_1
B_2
B_3
B_4
$\overline{LD}$ or $\overline{WR}$
D_1 Q_1 FF1 Clk $\bar{Q}_1$
D_2 Q_2 FF2 Clk $\bar{Q}_2$
D_3 Q_3 FF3 Clk $\bar{Q}_3$
D_4 Q_4 FF4 Clk $\bar{Q}_4$
Clock
$\overline{RD}$
Q_1
Q_2
Q_3
Q_4

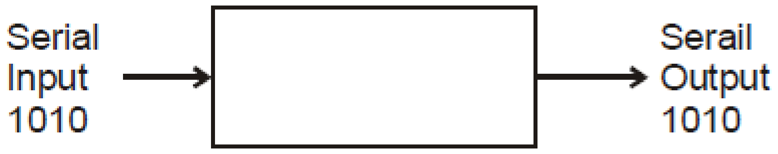
Fig. 2. 4-bit Controlled buffer register

Shift Registers

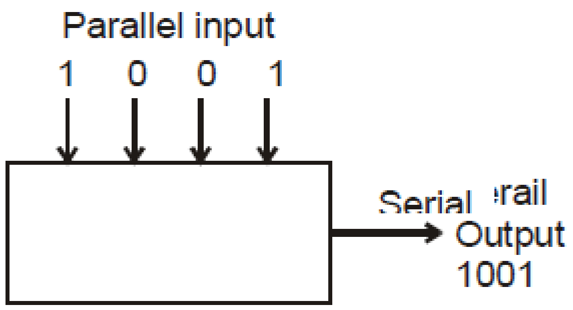
- A shift register is a register in which binary data can be stored, and this data can be shifted to the left or right when a shift signal is applied.

Shift Register: Types

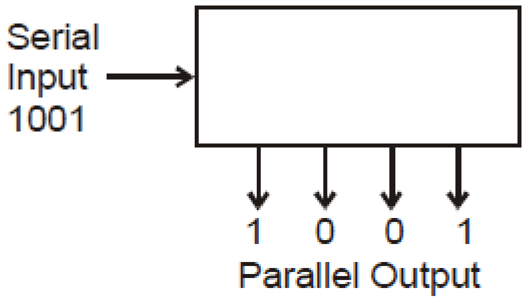
Serial In Serial Out Shift Register



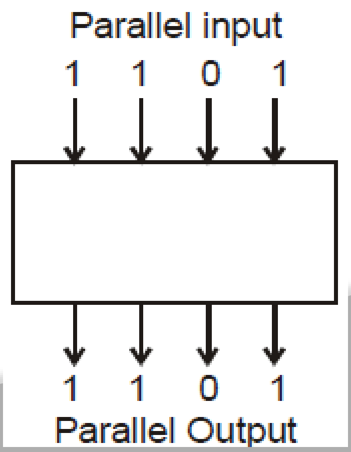
Parallel In Serial Out Shift Register



Serial In Parallel Out Shift Register



Parallel In Parallel Out Shift Register



1. Serial In Serial Out Shift Register

Serial Input 1010
Serail Output 1010

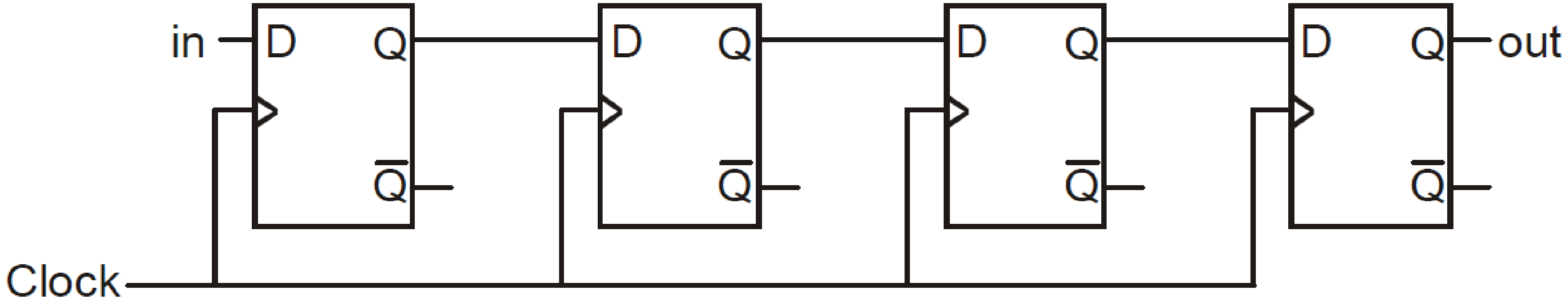


Fig. 4-bit serial-in serial-out shift register

1. Serial In Serial Out Shift Register

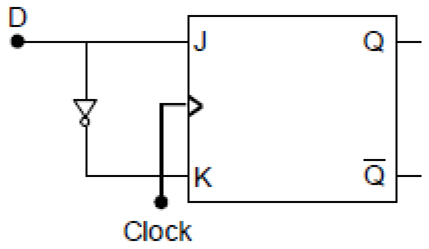


Fig. 1. JK Flip-flop converted into D-Flip-Flop

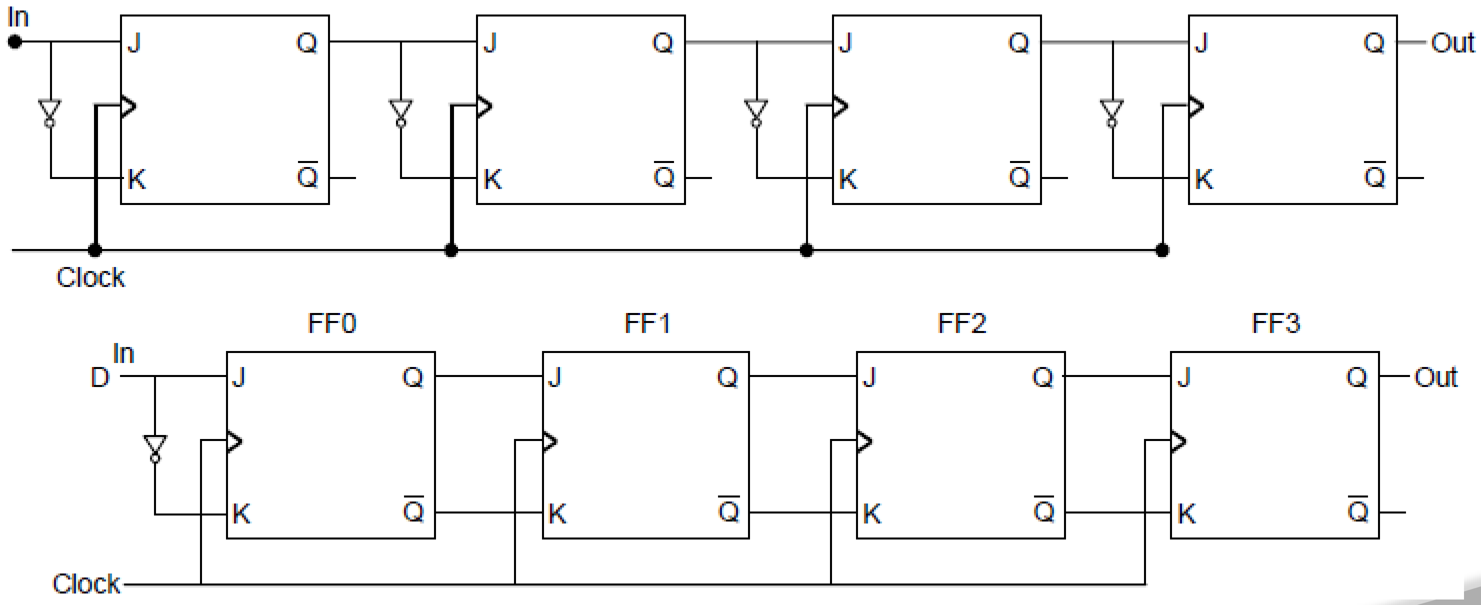
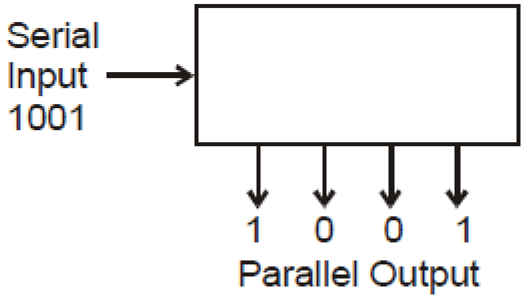


Fig. 2. 4-bit serial in serial out shift register using JK Flip-Flop

2. Serial In Parallel Out Shift Register



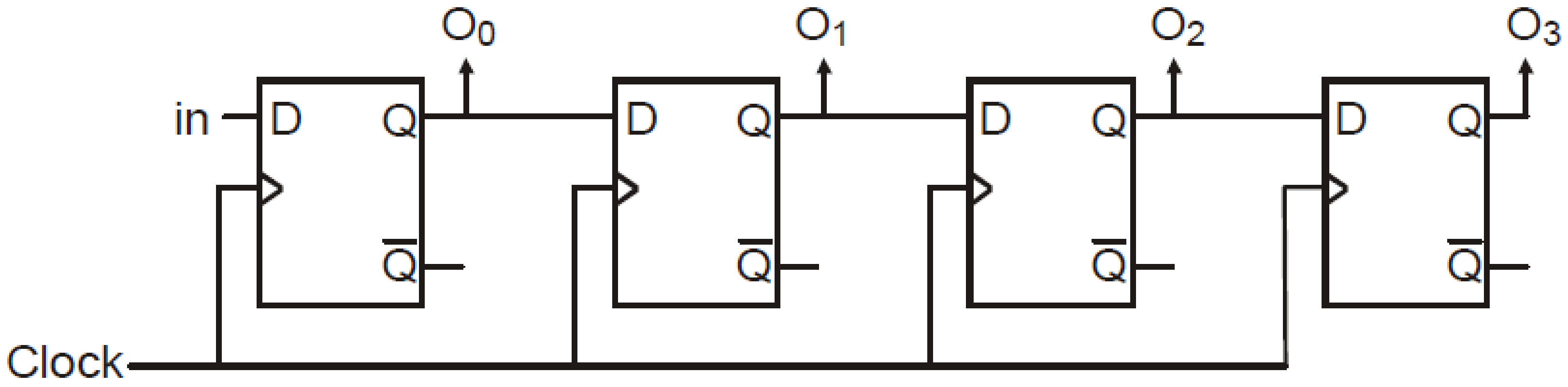
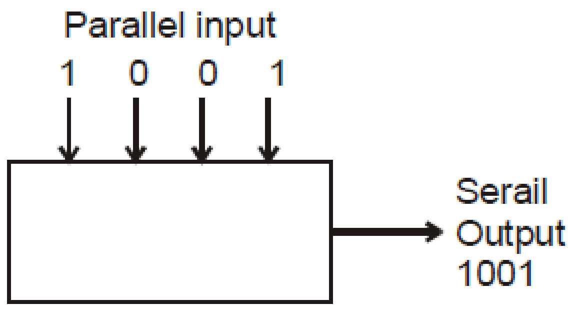


Fig. 4-bit serial-in parallel-out shift register

3. Parallel In Serial Out Shift Register



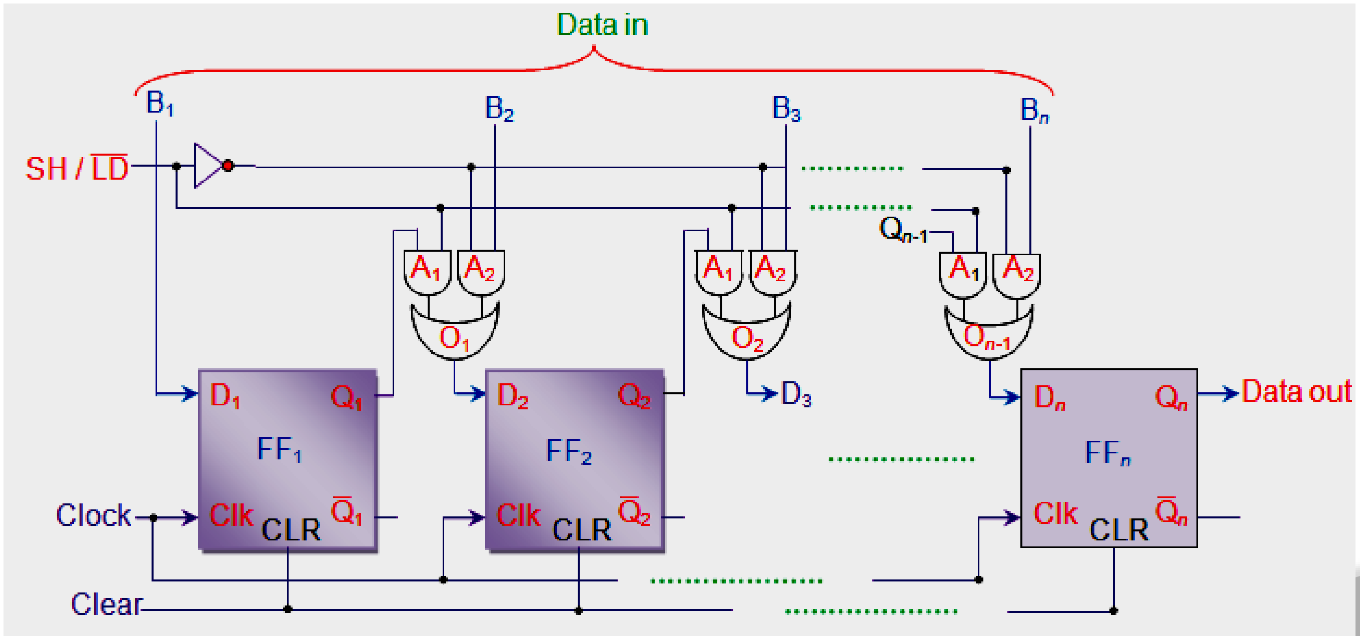
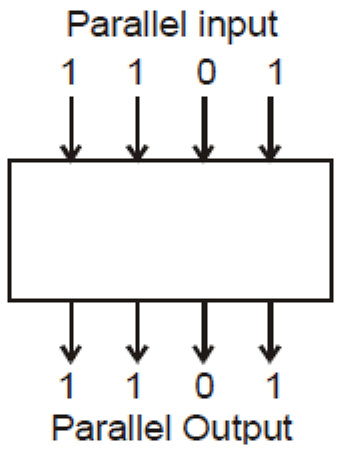


Figure 1 *n*-bit Parallel-In Serial-Out Right-Shift Shift Register

4. Parallel In Parallel Out Shift Register



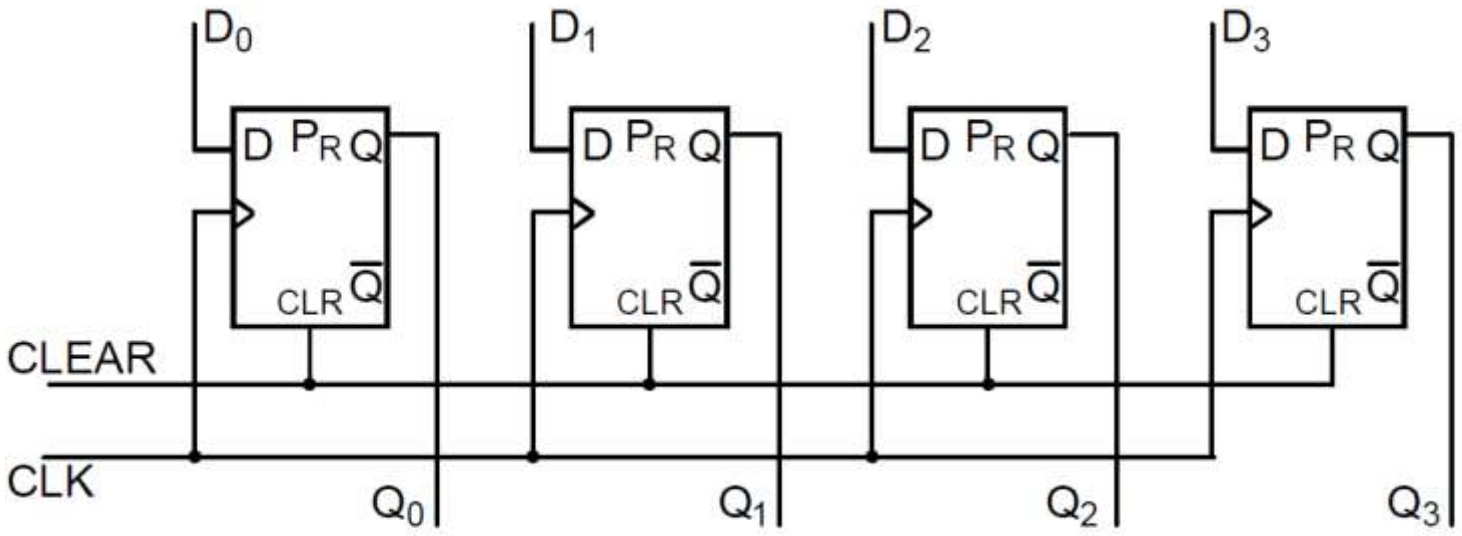
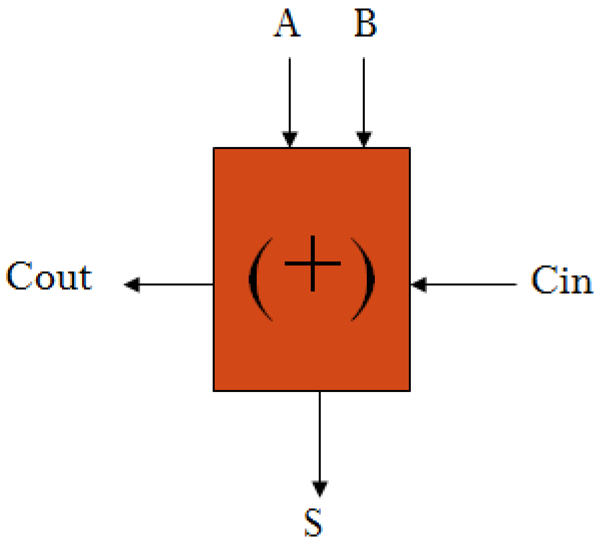


Fig. 4-bit parallel-in parallel-out shift register

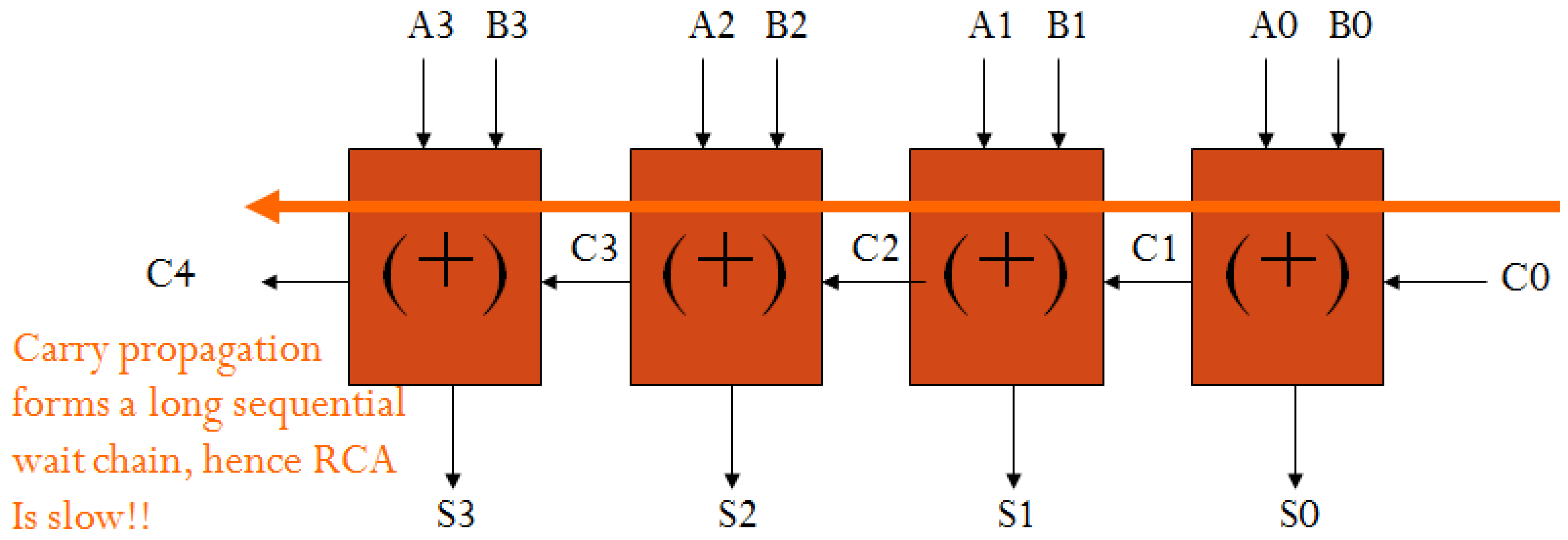
CARRY LOOK AHEAD ADDER

A simple one-bit full adder



- It takes A, B, and Cin as input and generates S and Cout in 2 gate delays (SOP)

4-bit RCA



A3 B3
A2 B2
A1 B1
A0 B0
C4
(+)
C3
(+)
C2
(+)
C1
(+)
C0
Carry propagation forms a long sequential wait chain, hence RCA Is slow!!
S3
S2
S1
S0

•Work from lowest bit to highest bit sequentially.

• With A0, B0, and C0, the lowest bit adder generates S0 and C1 in 2 gate delay.

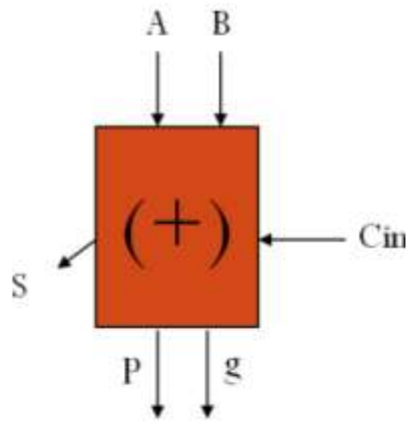
• With A1, B1, and C1 ready, the second bit adder generates S1 and C2 in 2 gate delay.

• Each bit adder has to wait for the lower bit adder to propagate the carry.

Carry Look Ahead Logic

- Now even before the carry in (Cin) is available, based on the inputs (A,B) only, can we say anything about the carry out?
- Under what condition will the bit propagate an outgoing carry (Cout), if there is an incoming carry (Cin)?
- Under what condition will the bit generate an outgoing carry (Cout), regardless of whether there is an incoming carry (Cin)?

1-bit CLA adder



• Instead of Cout, an 1-bit CLA adder block takes A, B inputs and generates p,g

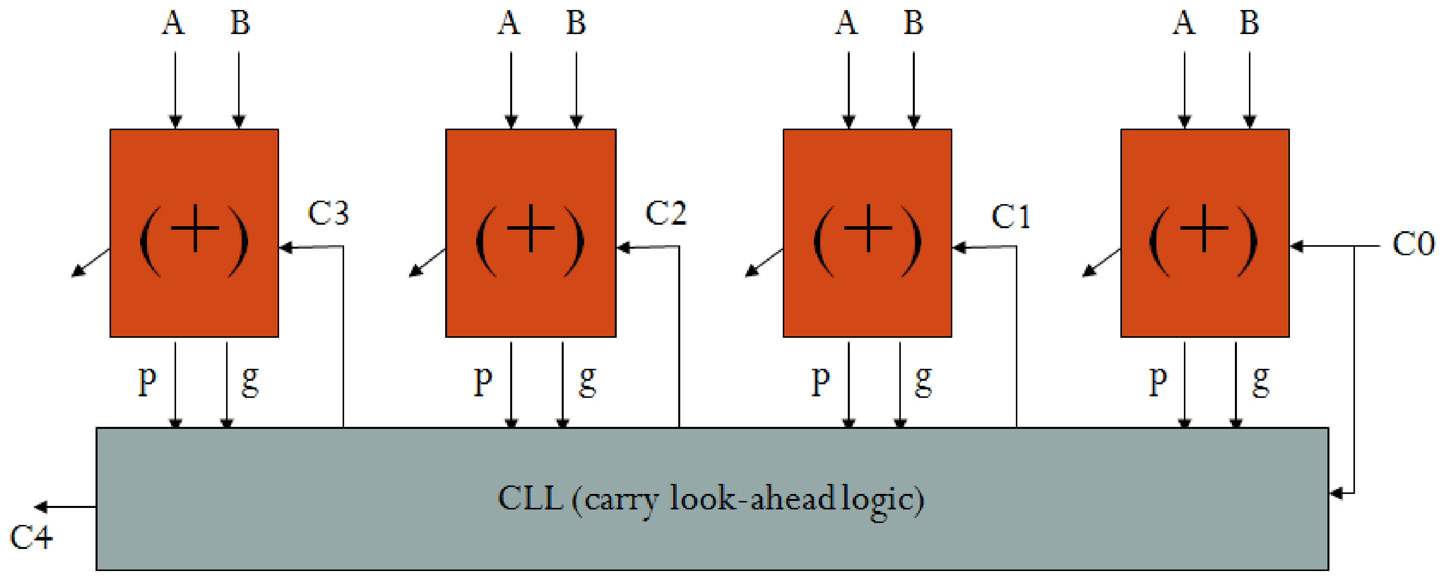
• p=propagator =>I will propagate the Cin to the next bit. p = A+B (If either A or B is 1, Cin=1 causes Cout=1)

• g=generator =>I will generate a Cout independent of what Cin is. g = AB (If both A and B are 1, Cout=1 for sure)

• p,g are generated in 1 gate delay after we have A,B. Note that Cin is not needed to generate p,g.

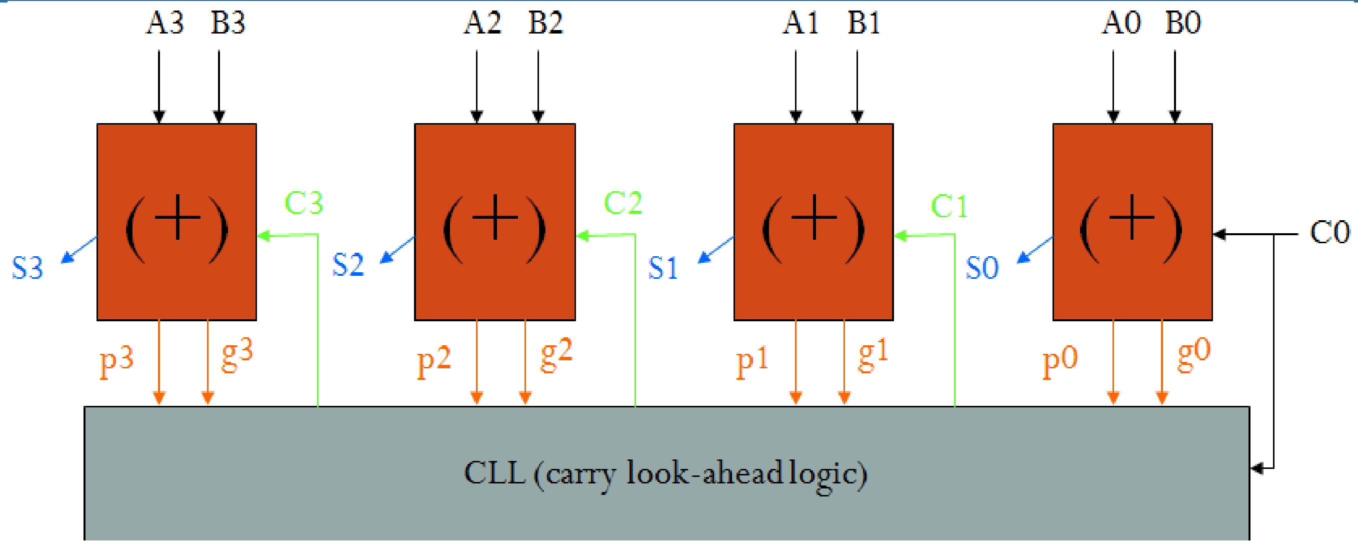
• S is generated in 2 gate delay after we get Cin (SOP).

4-bit CLA



- The CLL takes p,g from all 4 bits and C0 as input to generate all Cs in 2 gate delay.
- C1=g0+p0C0,
- C2=g1+p1g0+p1p0C0,
- C3=g2+p2g1+p2p1g0+p2p1p0c0,
- C4=g3+p3g2+p3p2g1+p3p2p1g0+p3p2p1p0c0 (Note: this C4 is too complicated to generate in 2-level SOP representation)

4-bit CLA



- Given A,B's, all p,g's are generated in 1 gate delay in parallel.
- Given all p,g's, all C's are generated in 2 gate delay in parallel.
- Given all C's, all S's are generated in 2 gate delay in parallel.
- Key virtue of CLA: sequential operation in RCA is broken into parallel operation

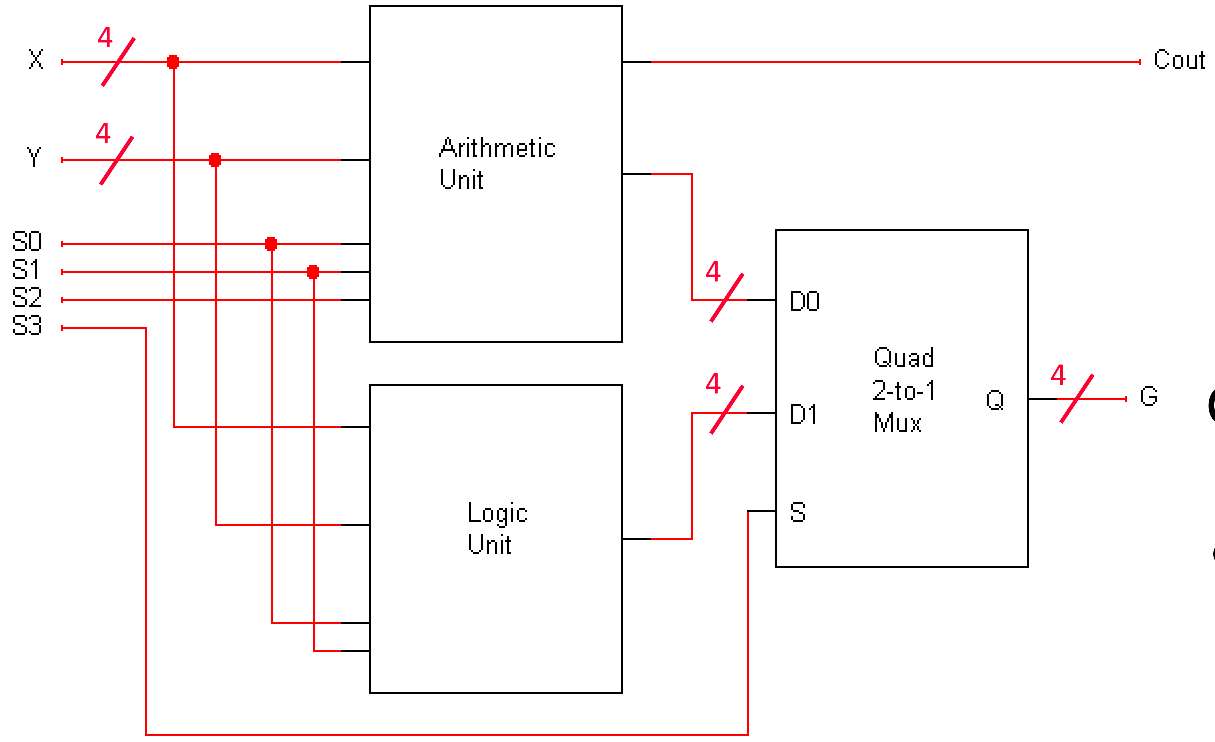
ALU function table

- This table shows a sample function table for an ALU.
- All of the arithmetic operations have S_3=0, and all of the logical operations have S_3=1.
- These are the same functions we saw when we built our arithmetic and logic units a few minutes ago.
- Since our ALU only has 4 logical operations, we don't need S_2. The operation done by the logic unit depends only on S_1 and S_0.

S_3	S_2	S_1	S_0	Operation
0	0	0	0	$G = X$
0	0	0	1	$G = X + 1$
0	0	1	0	$G = X + Y$
0	0	1	1	$G = X + Y + 1$
0	1	0	0	$G = X + Y'$
0	1	0	1	$G = X + Y' + 1$
0	1	1	0	$G = X - 1$
0	1	1	1	$G = X$
1	x	0	0	$G = X$ and Y
1	x	0	1	$G = X$ or Y
1	x	1	0	$G = X \oplus Y$
1	x	1	1	$G = X'$

A complete ALU circuit

The / and 4 on a line indicate that it's actually *four* lines.



C_{out} should be ignored when logic operations are performed (when S3=1).

G is the final ALU output.

- When S3 = 0, the final output comes from the arithmetic unit.
- When S3 = 1, the output comes from the logic unit.

The arithmetic and logic units share the select inputs S1 and S0, but only the arithmetic unit uses S2.

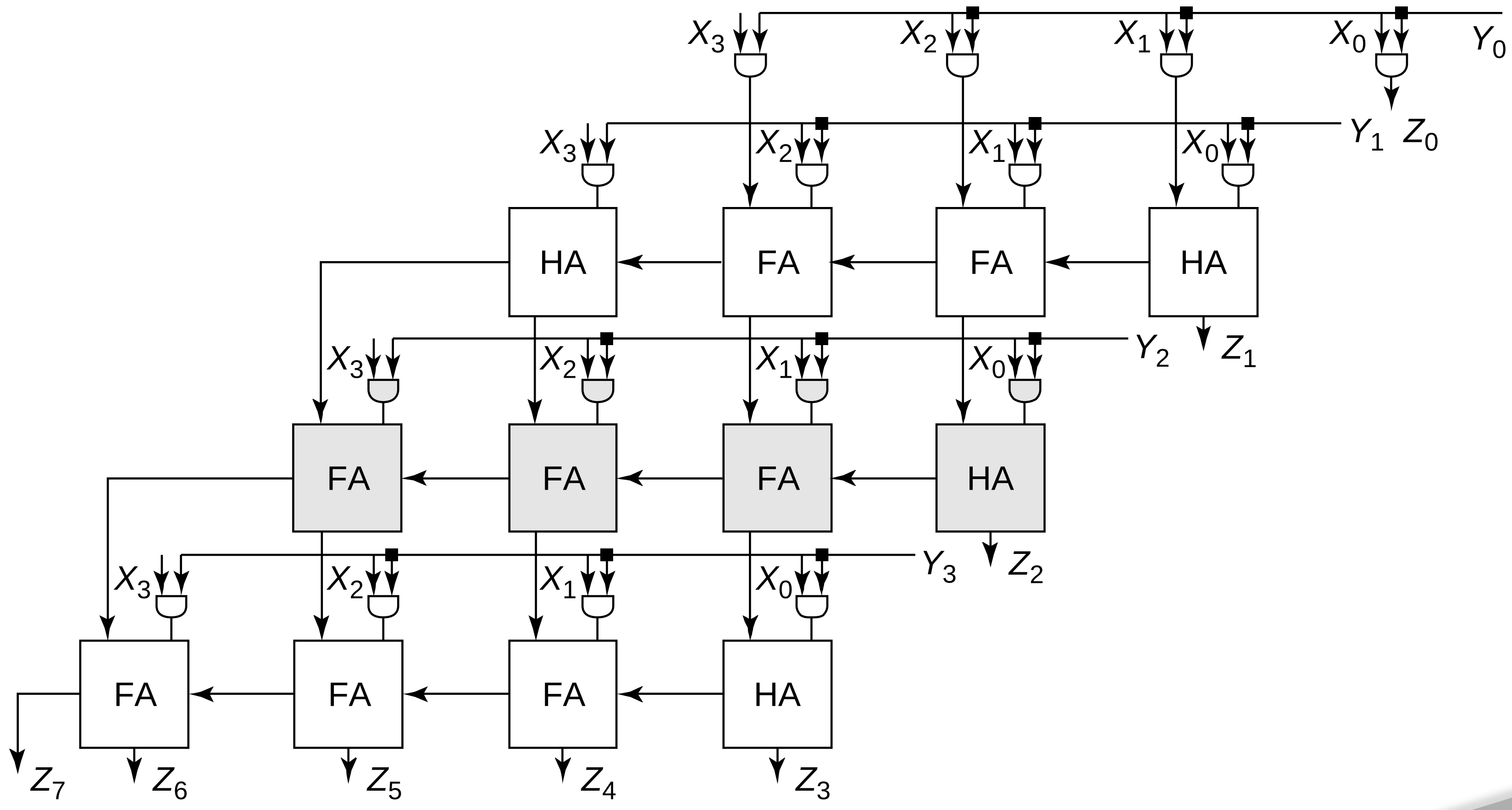
Multiplier Design: Basic Building Blocks

- Datapath
- Execution units
- Adder, multiplier, divider, shifter, etc.
- Register file and pipeline registers
- Multiplexers, decoders
- Control
- Finite state machines (PLA, ROM, random logic)
- Interconnect
- Switches, arbiters, buses
- Memory
- Caches (SRAMs), TLBs, DRAMs, buffers

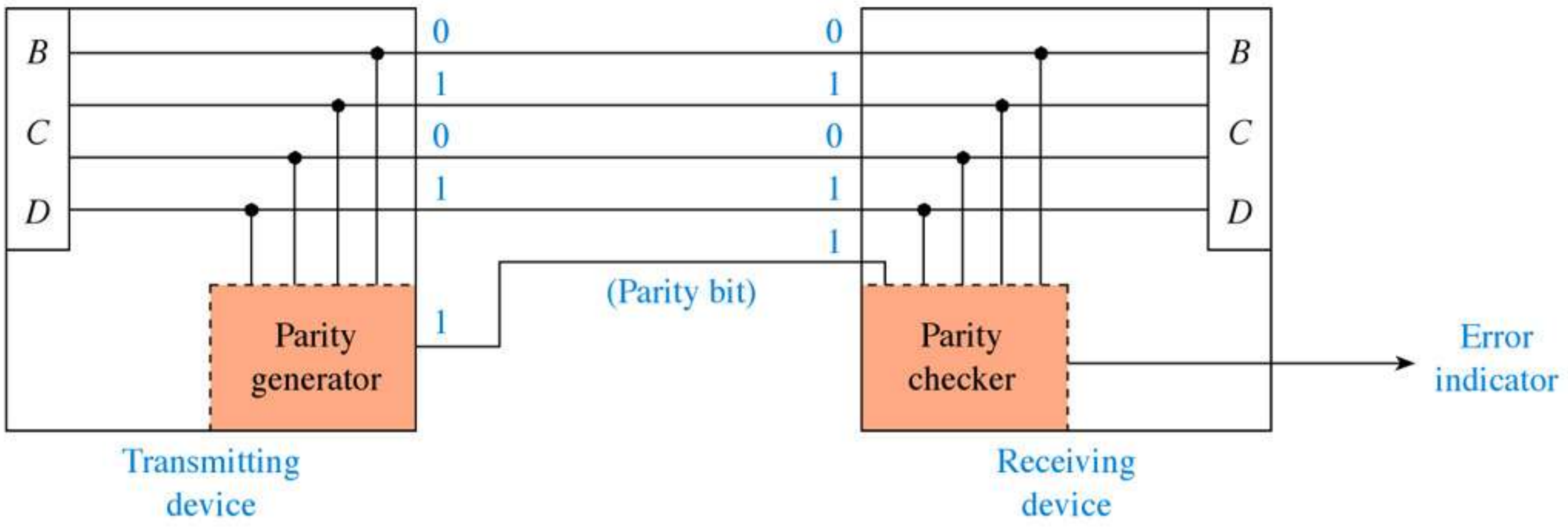
The Binary Multiplication

```
        1 0 1 0 1 0       Multiplicand
x           1 0 1 1       Multiplier
-------------------
        1 0 1 0 1 0   ⎫
      1 0 1 0 1 0     ⎪
    0 0 0 0 0 0       ⎬   Partial products
+ 1 0 1 0 1 0         ⎭
-------------------
  1 1 1 0 0 1 1 1 0       Result
```

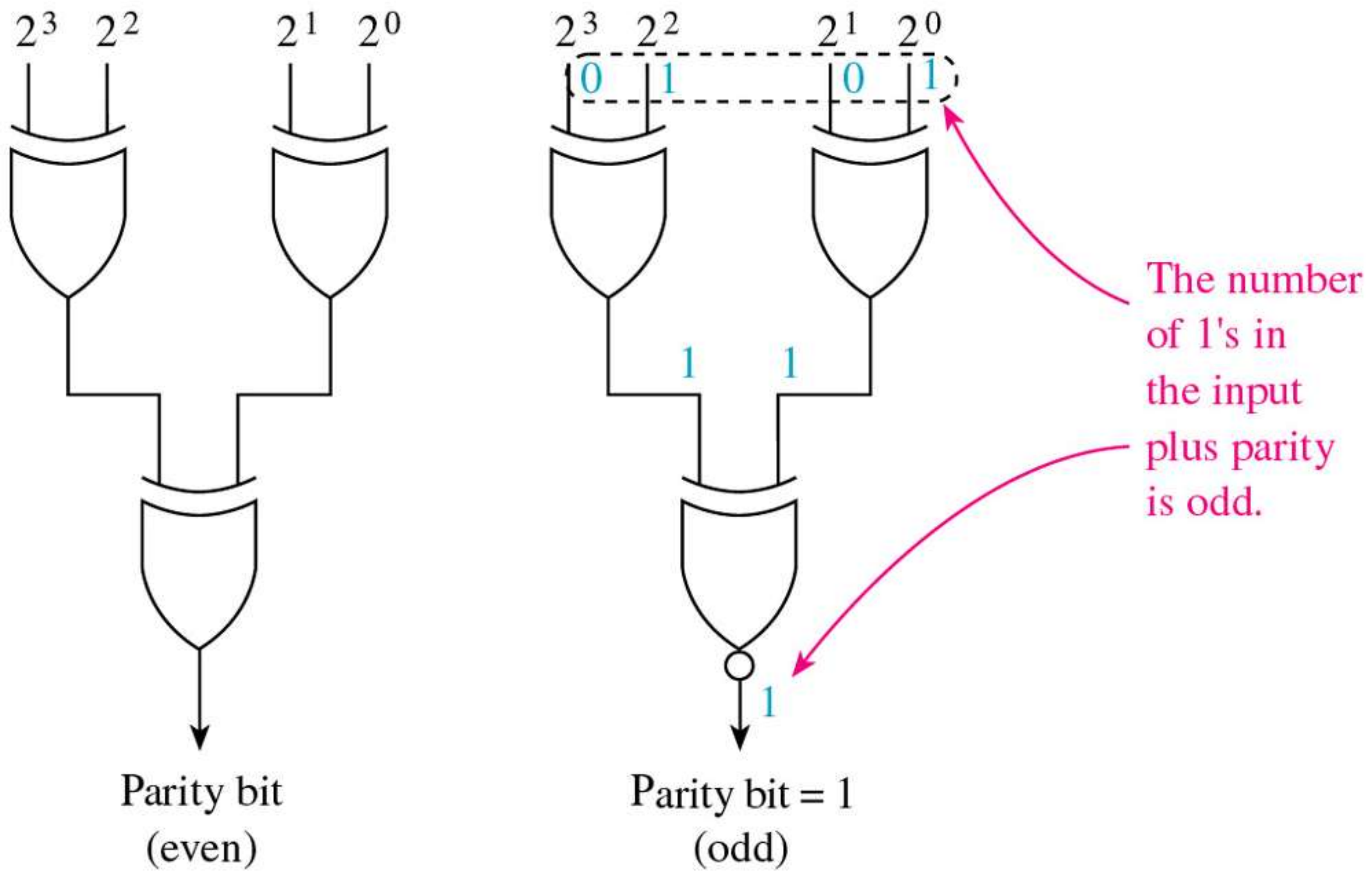
The Array Multiplier:



Parity Generator / Checker



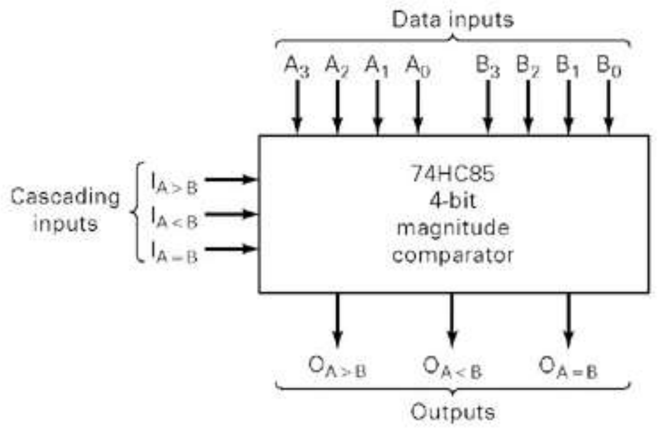
Four-bit even- and odd-parity generators



Comparator

- A comparator is a precision instrument employed to compare the dimension of a given component with a working standard (usually slip gauges).
- It thus does not measure the actual dimension but indicate how much it differs from the basic dimension.

Magnitude Comparator

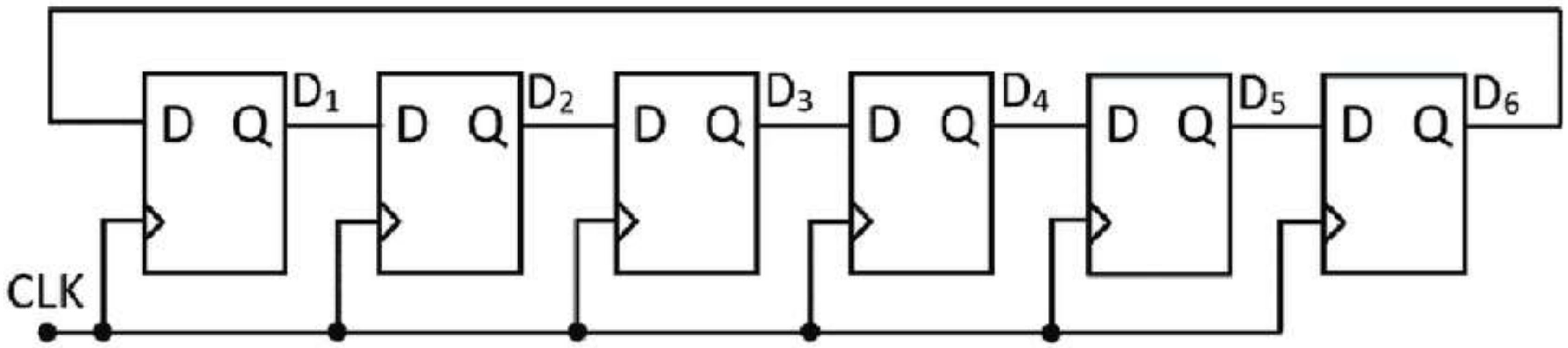

Data inputs
A_3 A_2 A_1 A_0 B_3 B_2 B_1 B_0
Cascading inputs
$I_{A>B}$
$I_{A<B}$
$I_{A=B}$
74HC85 4-bit magnitude comparator
$O_{A>B}$ $O_{A<B}$ $O_{A=B}$
Outputs

TRUTH TABLE

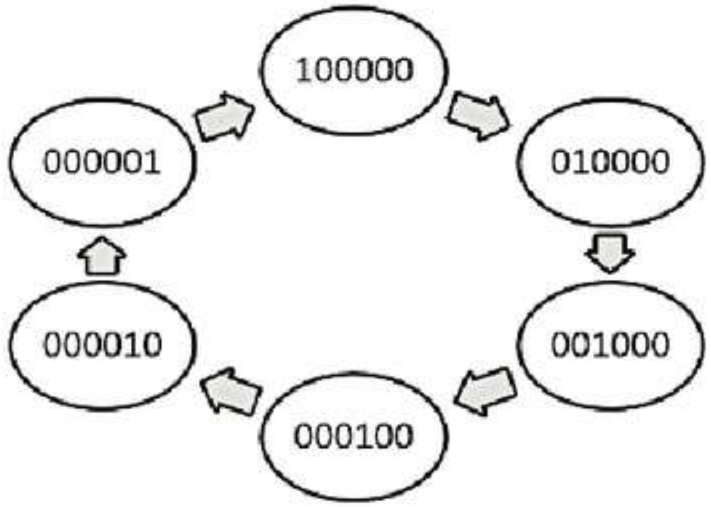
COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3>B_3$	X	X	X	X	X	X	H	L	L
$A_3<B_3$	X	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2>B_2$	X	X	X	X	X	H	L	L
$A_3=B_3$	$A_2<B_2$	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	L	L	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	H	L	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	X	X	H	L	L	H
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	L	H	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	H	L	L	L	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

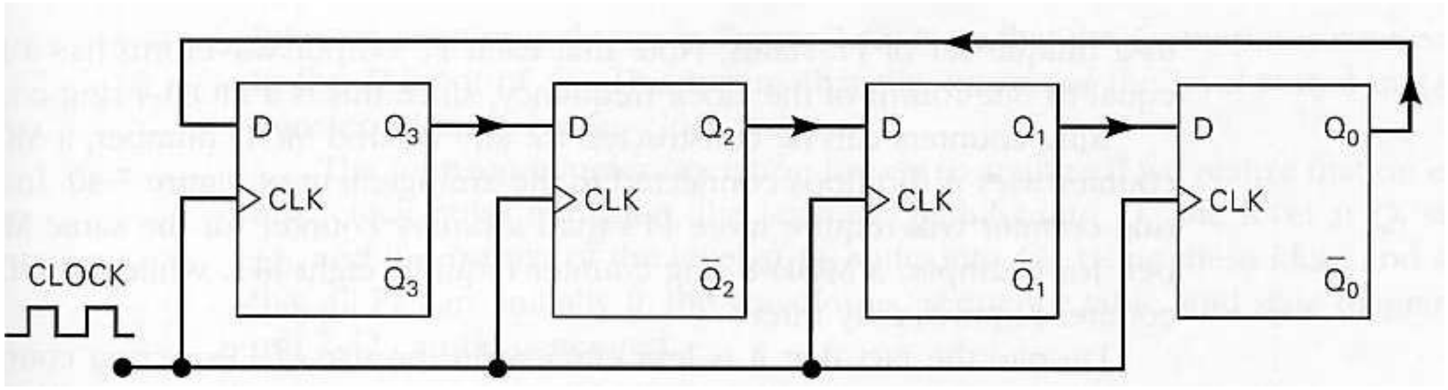
Ring Counter

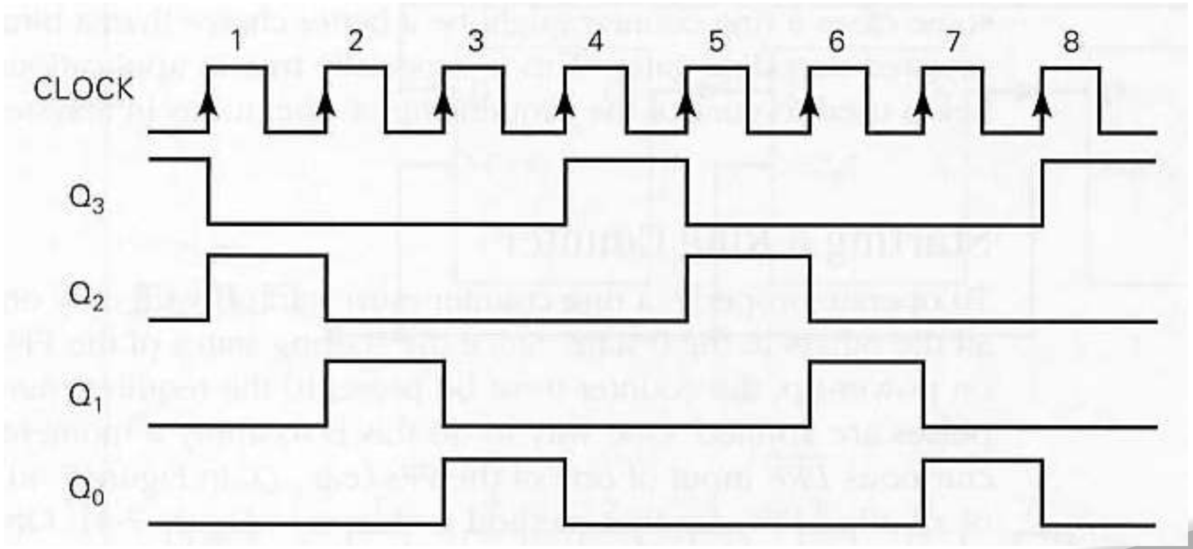


CLK	D_1	D_2	D_3	D_4	D_5	D_6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

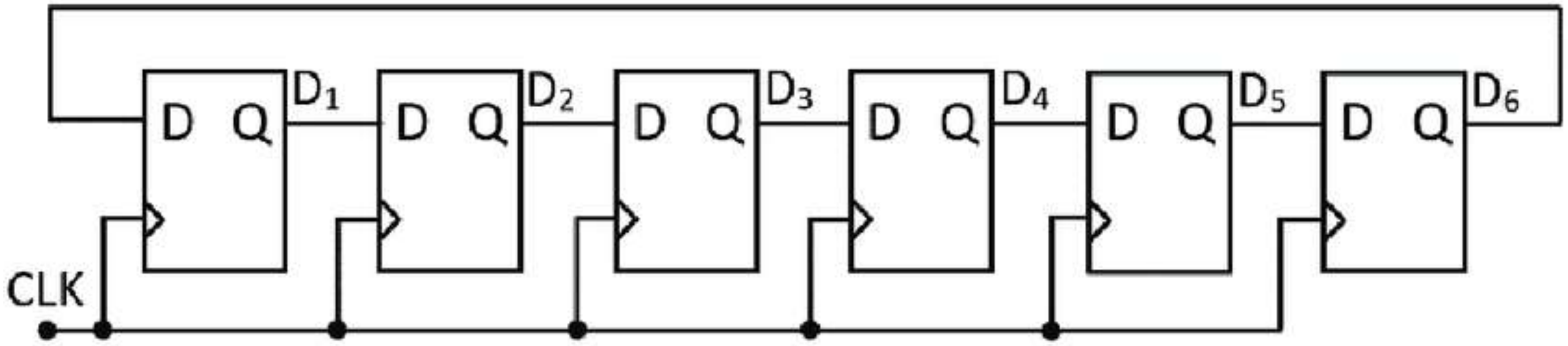


Ring Counter

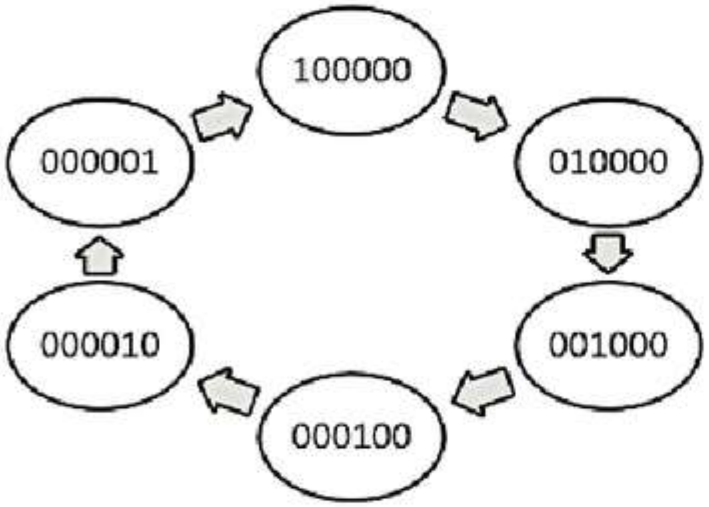




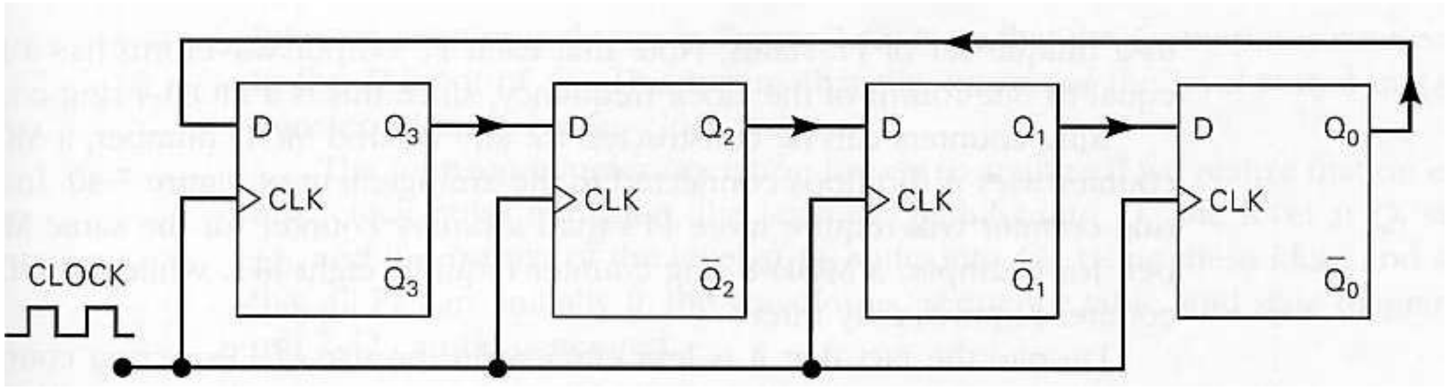
Ring Counter

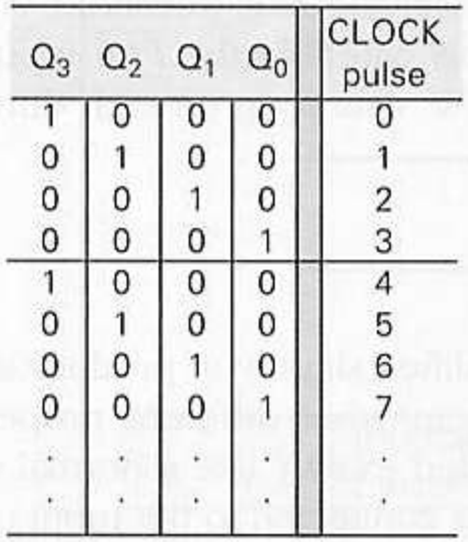


CLK	D_1	D_2	D_3	D_4	D_5	D_6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1



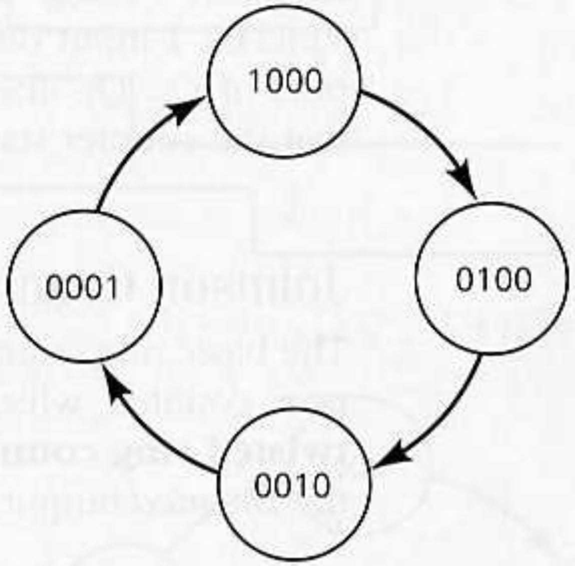
Ring Counter





Q_3	Q_2	Q_1	Q_0	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.
.	.	.	.	.

(c)



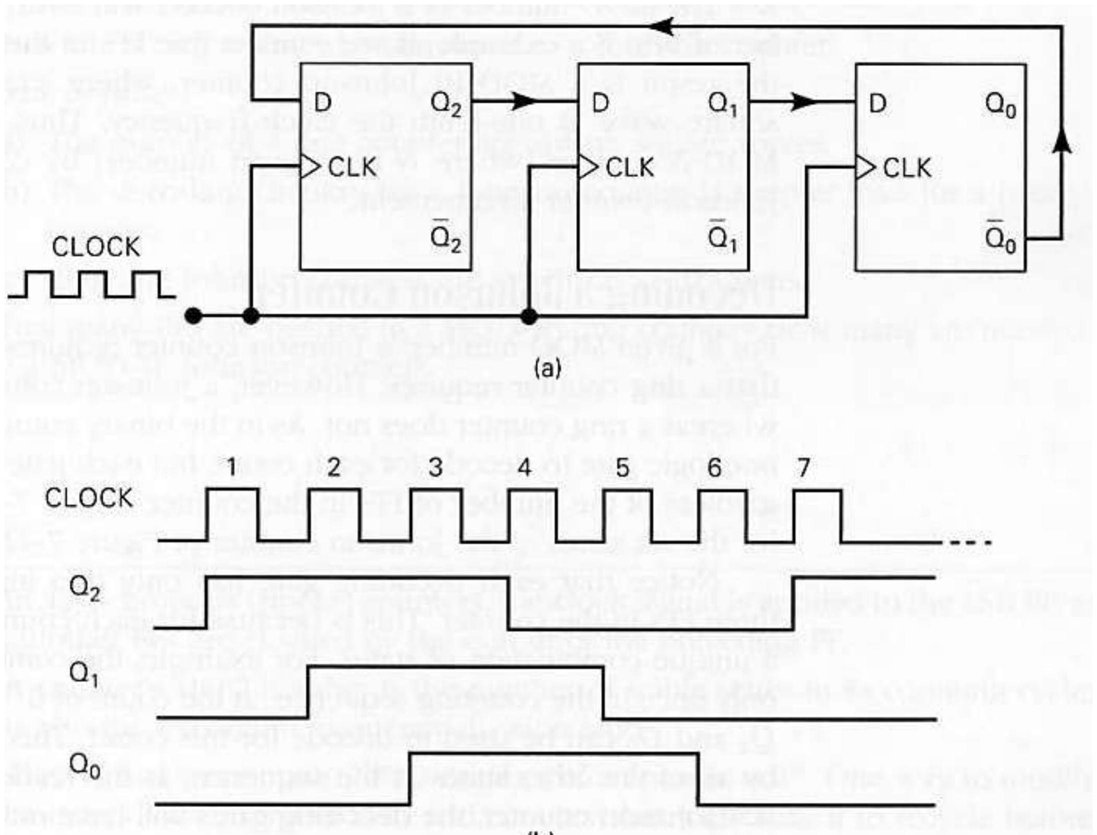
(d)

Johnson Counter

- Also known as the twisted-ring counter.
- Same as the ring counter except that the inverted output of the last FF is connected to the input of the first FF.
- Counting sequence:

  000→100→110→111→011→001→000

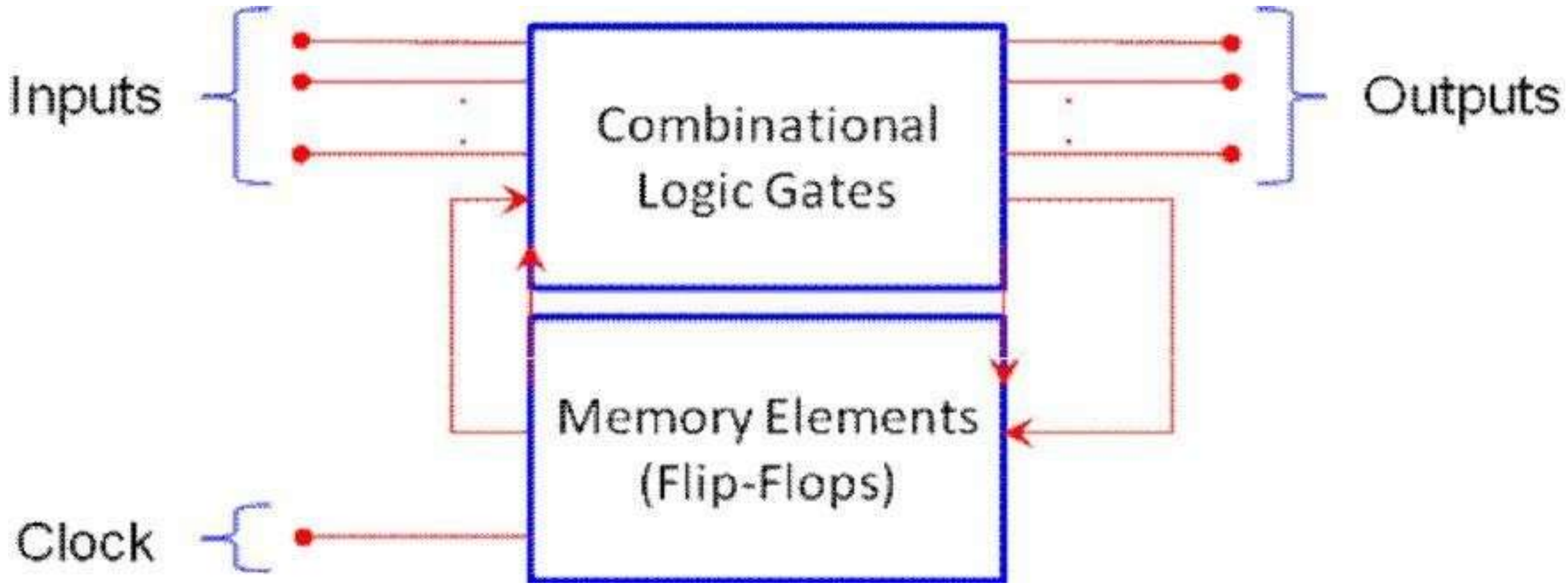
Johnson Counter



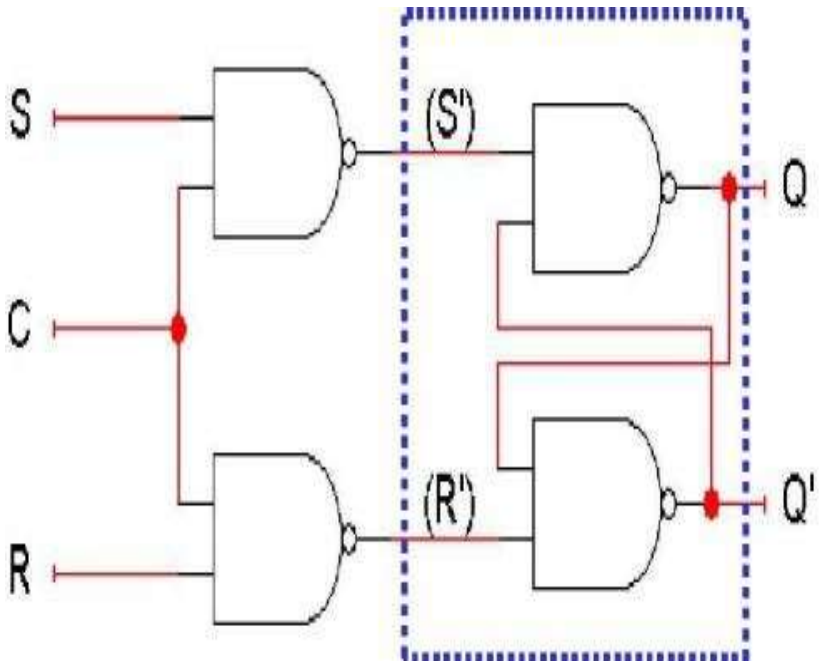
(a)

(b)

Sequential Circuits



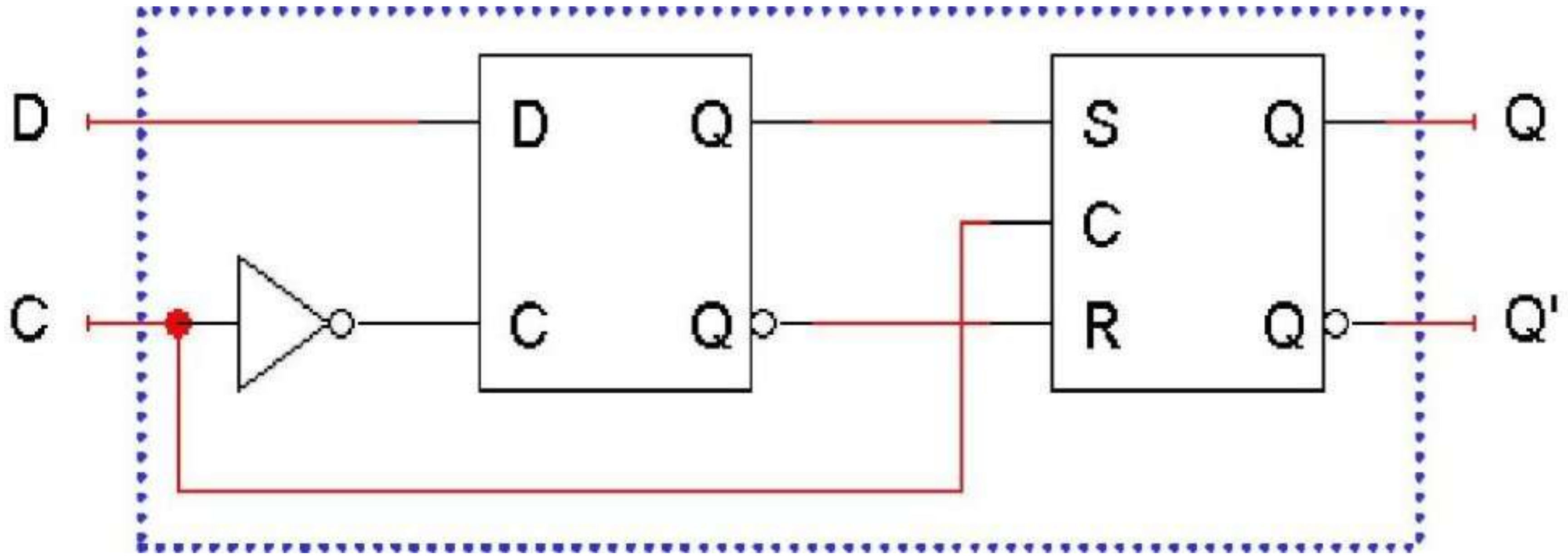
SR Latch



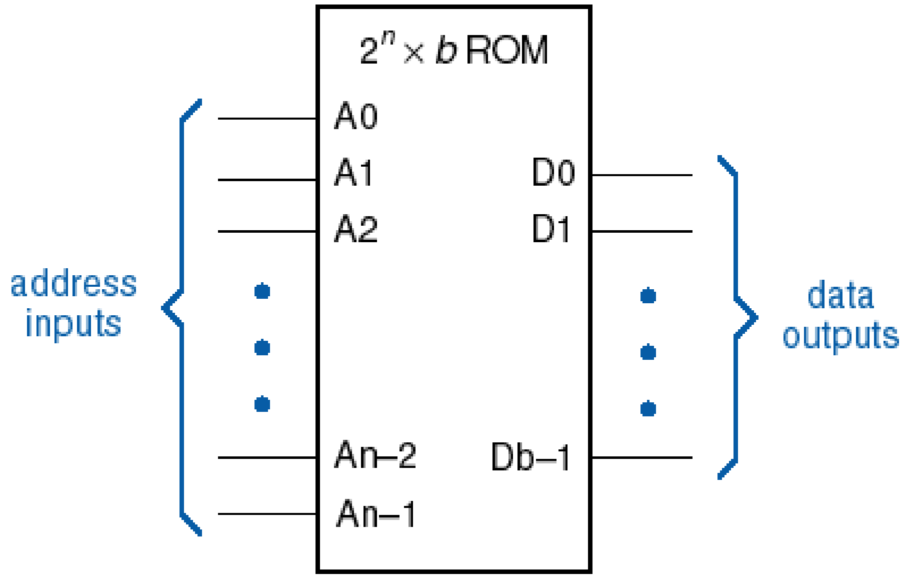
▪

▪

D flip-flop

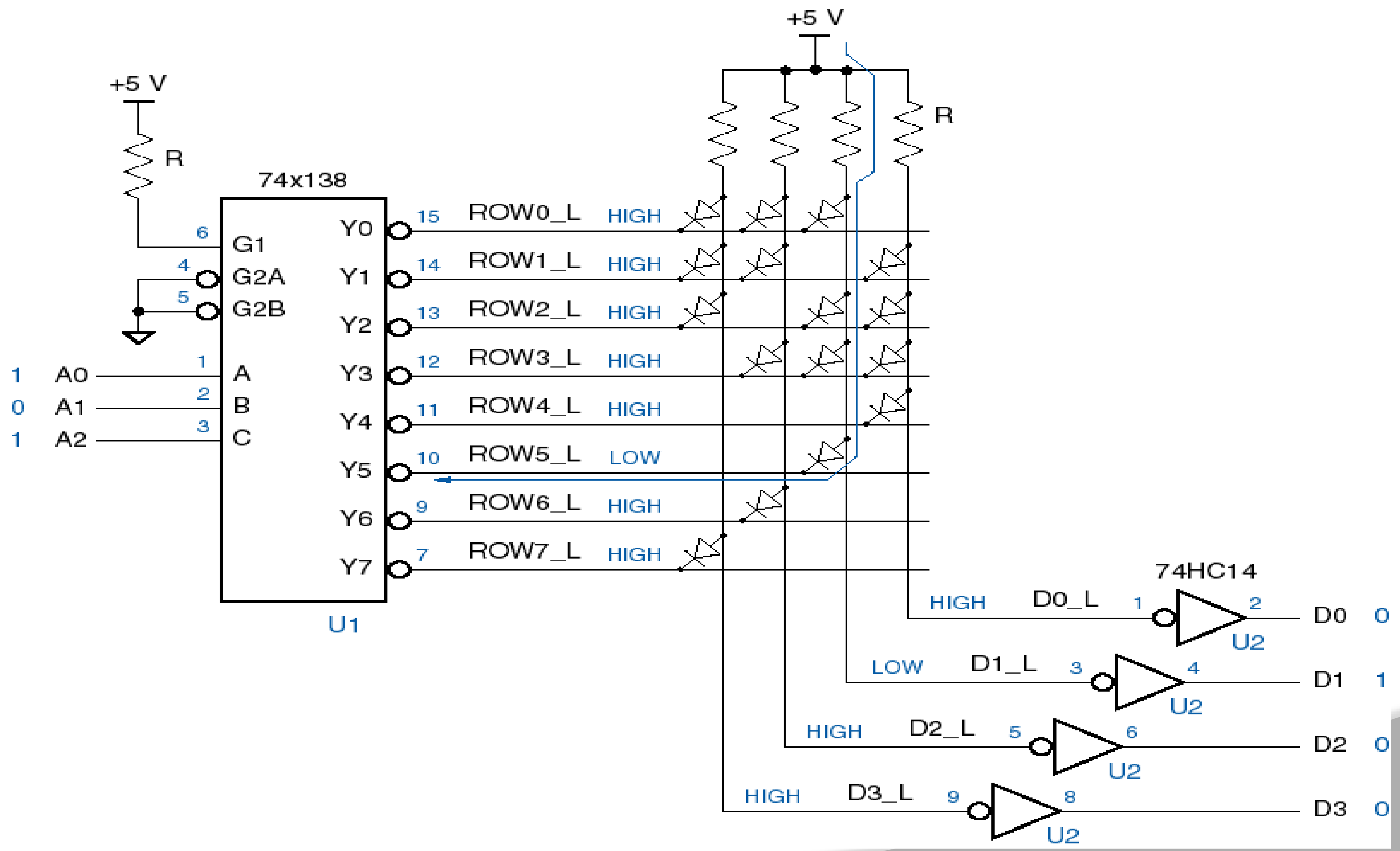


Read-Only Memory (ROM)

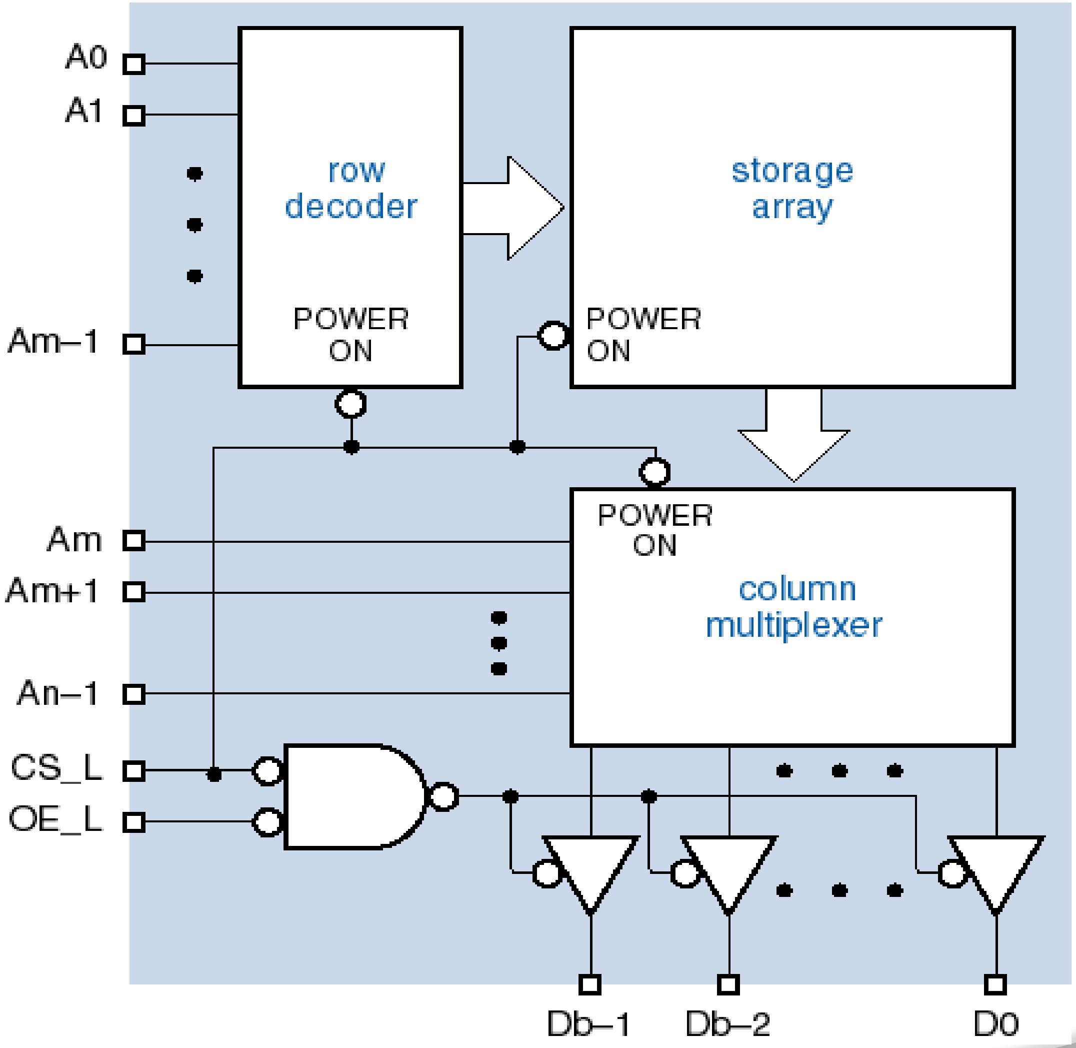


- ➢The inputs are called Address inputs and are traditionally named A0, A1, …An-1
- ➢The outputs are called Data outputs and are typically named D0, D1, …Db-1

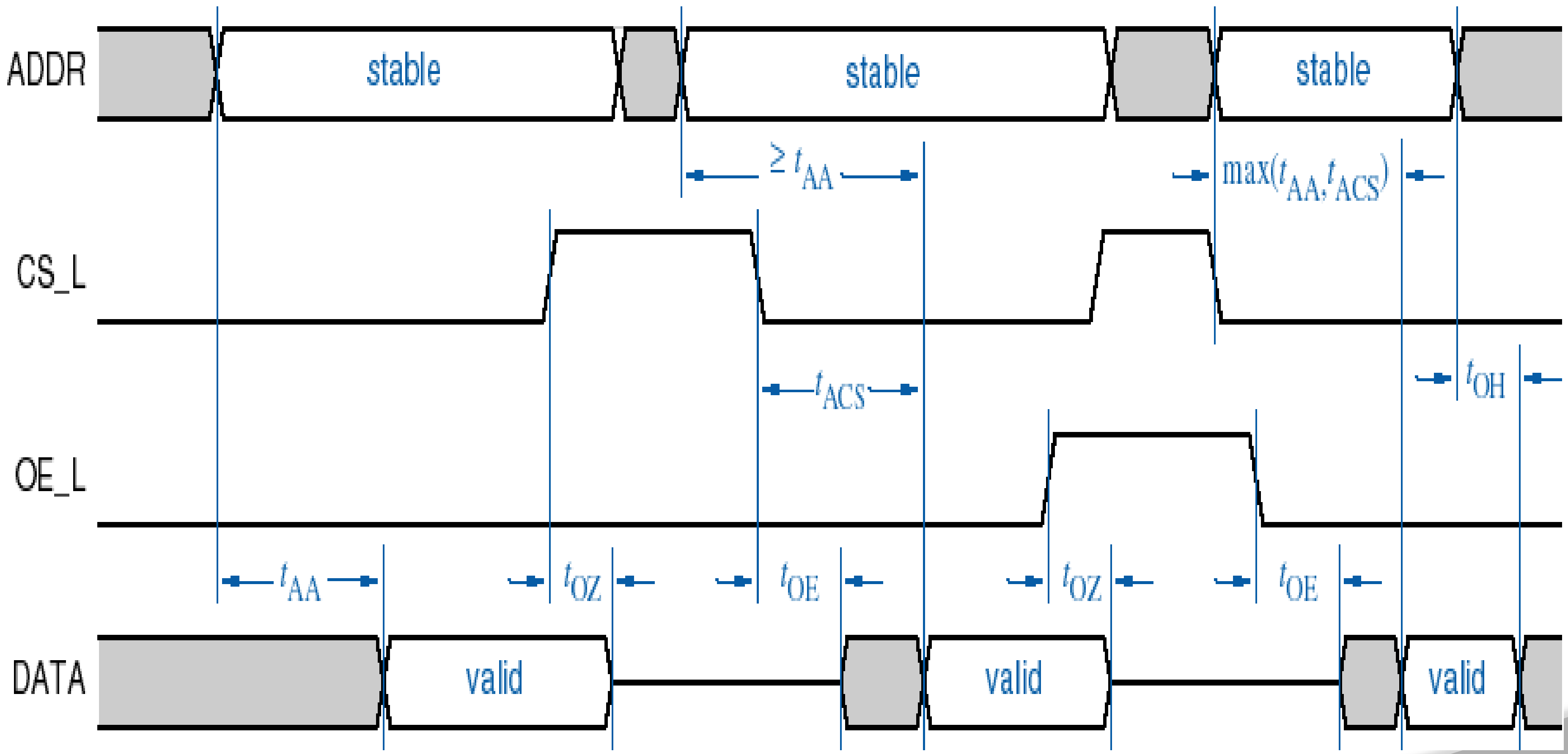
Logic Diagram of simple 8x4 diode ROM



Internal ROM structure, showing use of control inputs



ROM timing

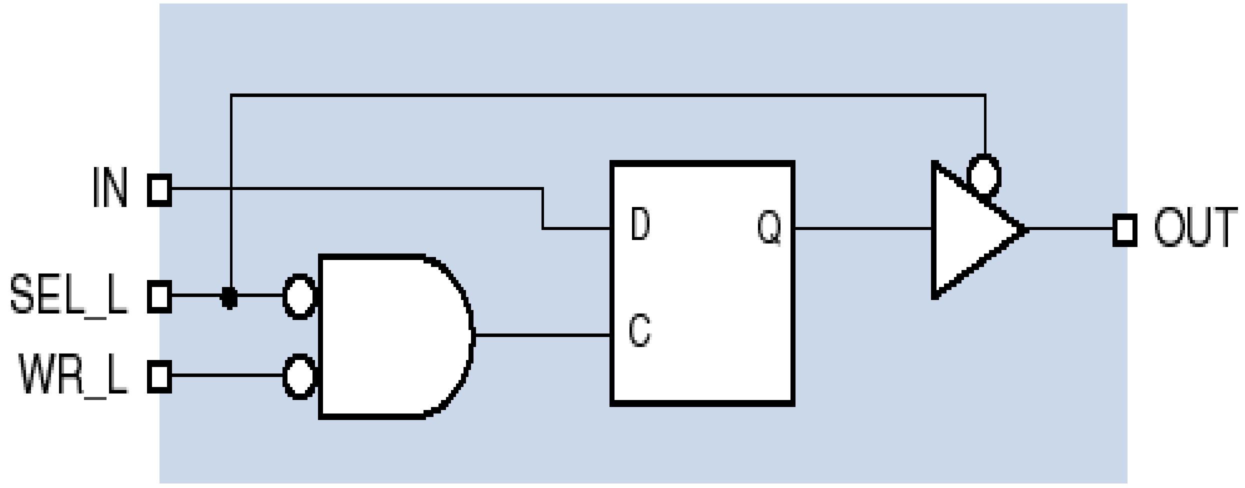


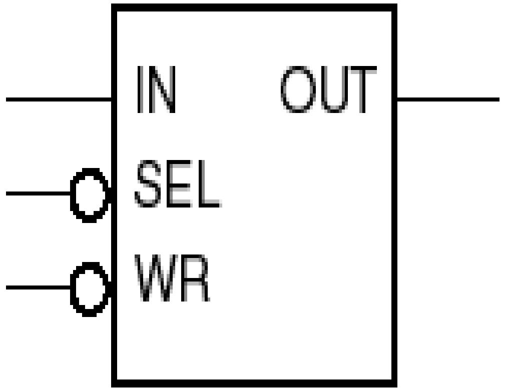
ADDR
stable
stable
stable
$\geq t_{AA}$
$\max(t_{AA}, t_{ACS})$
CS_L
t_{ACS}
t_{OH}
OE_L
t_{AA}
t_{OZ}
t_{OE}
t_{OZ}
t_{OE}
DATA
valid
valid
valid

Static-RAM Internal Structure

➢Each SRAM cell has the same functional behavior in the SRAM circuit.

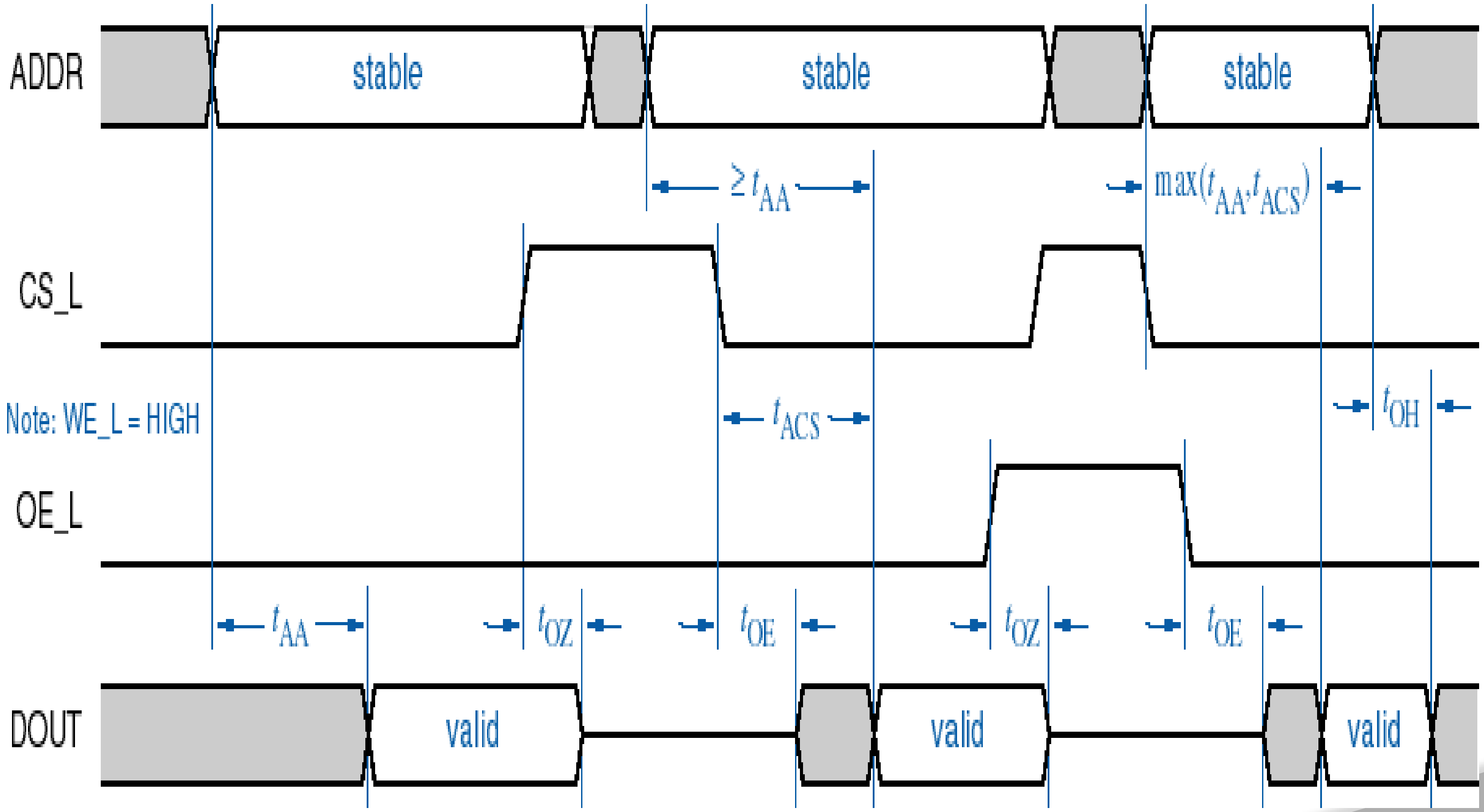
➢The storage device in each cell is a D-latch.



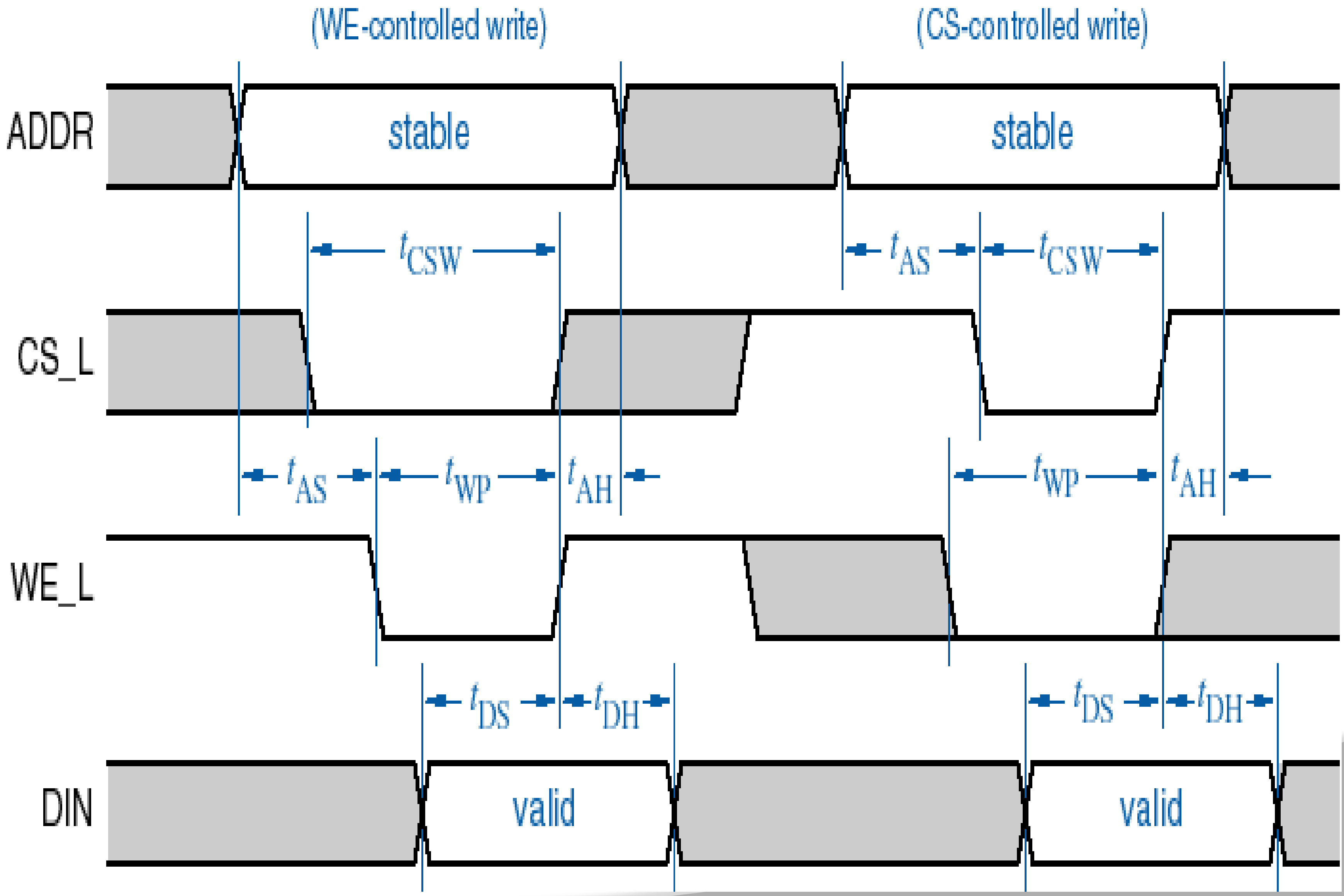


Static RAM cell

SRAM read timing



SRAM write timing



THANK YOU