EE 330
Lecture 44

Digital Circuits

- Dynamic Logic Circuits
- Higher-level digital blocks

Exam Schedule

Final **Wed May 11** **7:30 a.m.**

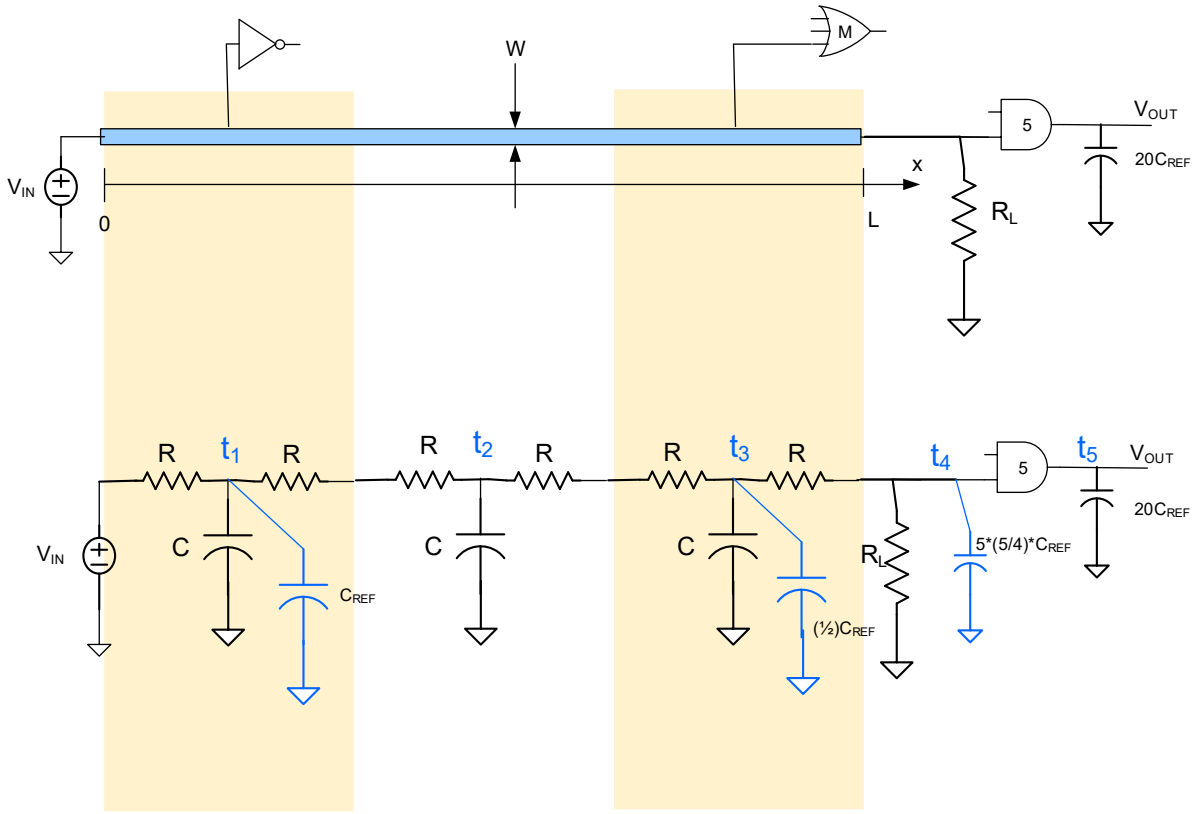
Review: Friday May 6 4:00 p.m. Rm 1012 Coover

Thank You !

to those of you who have worn masks throughout the semester

As a courtesy to fellow classmates, TAs, and the instructor

Elmore Delay Calculations



$$R = \frac{1}{6} R_{\square} \frac{L}{W}$$

$$C = \frac{1}{3} C_D$$

$$t_1 = R\left(C + C_{REF}\right) \qquad t_2 = 3RC \qquad t_3 = 5R\left(C + \frac{1}{2}C_{REF}\right) \qquad t_4 = \left[6R \,/\!/\!/\, R_L\right]\frac{25}{4}C_{REF} \qquad t_5 = t_{REF}\frac{20}{5}$$

$$t_{PROP} = \sum_{i=1}^{5} t_i$$

Power Dissipation in Logic Circuits

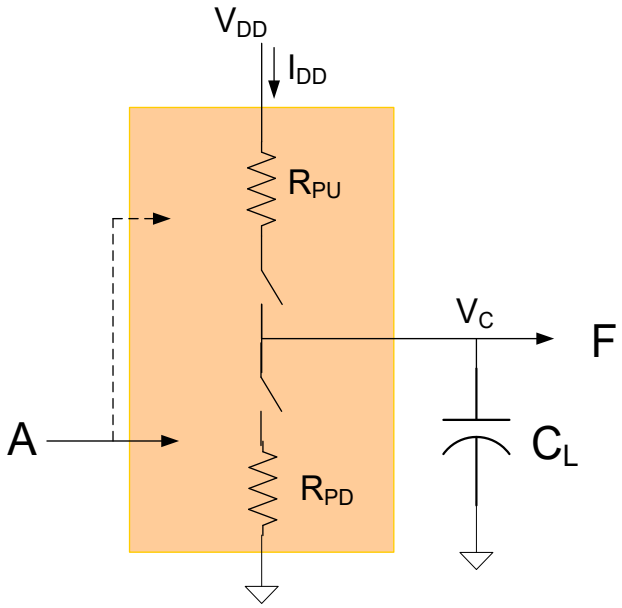
Types of Power Dissipation

- **Static**
- **Pipe**
- **Dynamic**
- **Leakage**
  - **Gate**
  - **Diffusion**
  - **Drain**

Dynamic Power Dissipation

Energy dissipated with clock signal itself

$$P_{DYN} = f_{CL} C_L V_{DD}^2$$



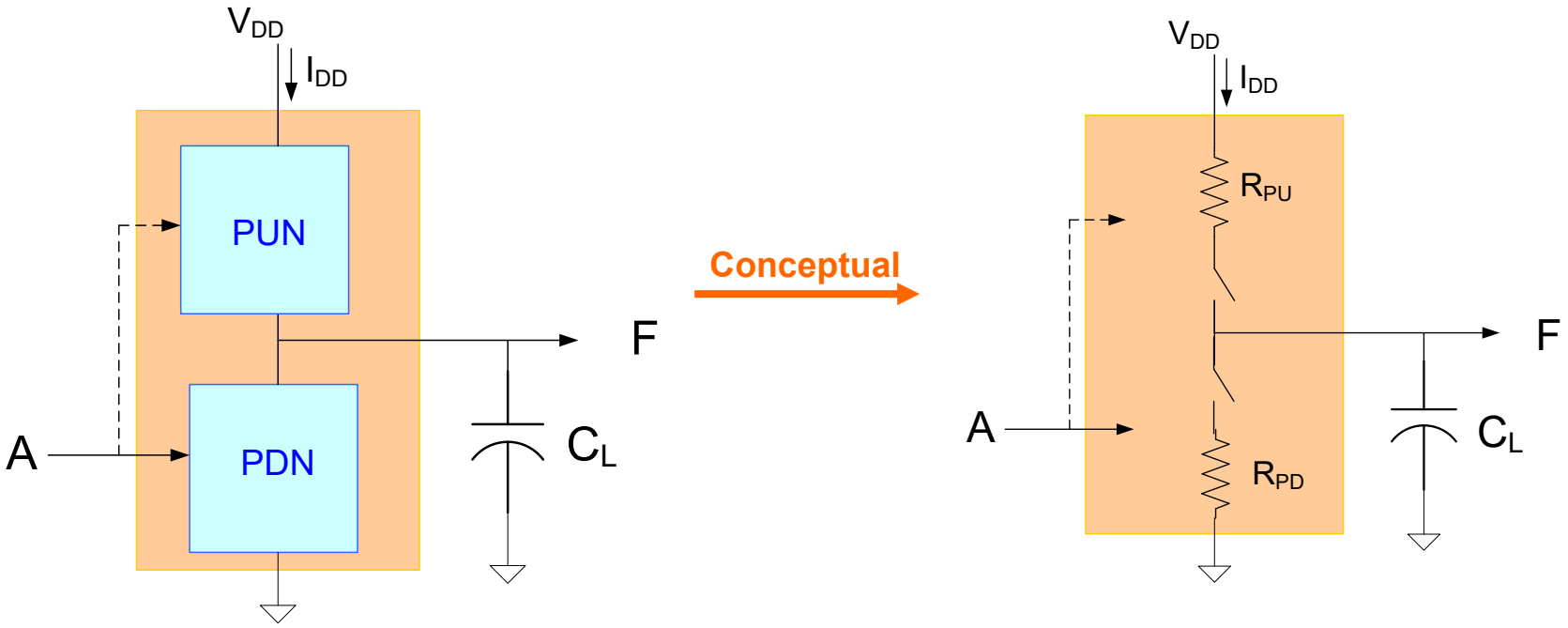
The clock transitions on every clock cycle (i.e. it has a transition duty cycle of 100%)

Clock distribution can cause significant power dissipation

But if a gate has a transition duty cycle of 50% with a clock frequency of f_{CL}

$$P_{DYN} = \frac{f_{CL}}{2} C_L V_{DD}^2$$

Power Dissipation

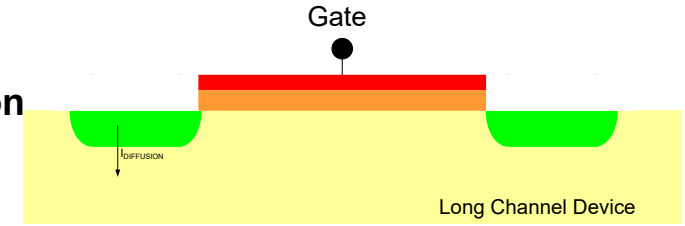


- **All power is dissipated in pull-up and pull-down devices**
- **C_L dissipates no power but PUN and PDN dissipate power when charging and discharging C_L**
- **Dynamic power dissipation reduced by more (often much more) than a factor of 2 if minimum sizing strategy is used**

Leakage Power Dissipation

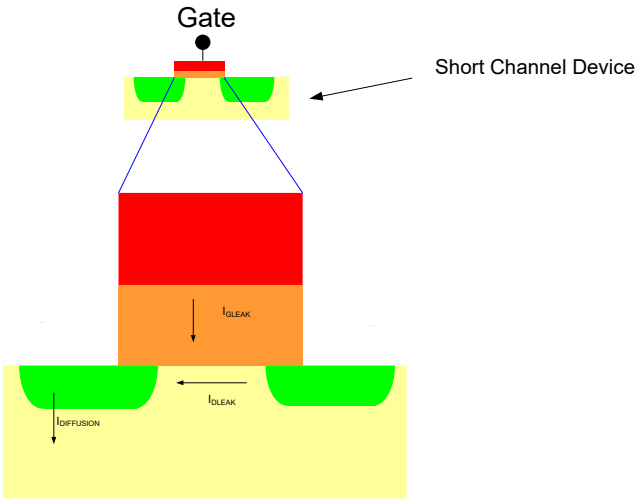
- Gate

- with very thin gate oxides, some gate leakage current flows
- major concern in 60nm and smaller processes
- actually a type of static power dissipation



-Diffusion

- Leakage across a reverse-biased pn junction
- Dependent upon total diffusion area
- May actually be dominant power loss on longer-channel devices
- Actually a type of static power dissipation

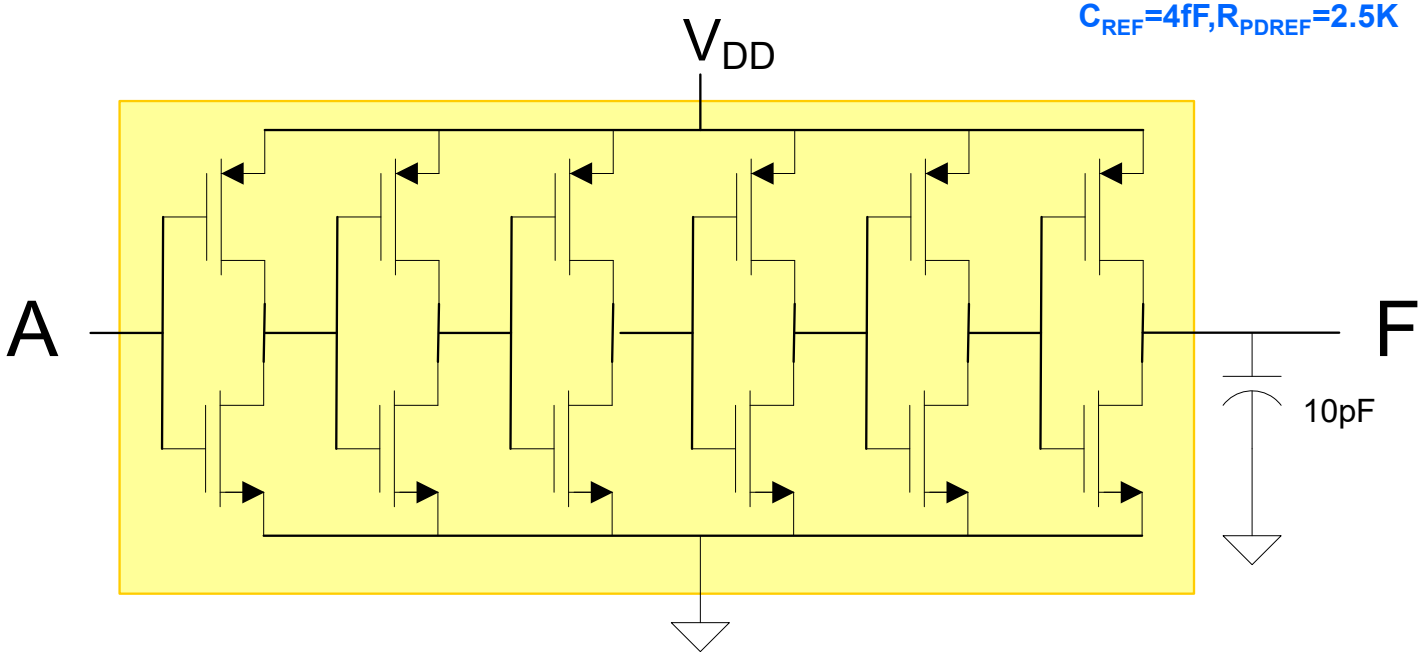


-Drain

- channel current due to small V_{GS}-V_T
- of significant concern only with low V_{DD} processes
- actually a type of static power dissipation

Example: Determine the dynamic power dissipation in the last stage of a 6-stage CMOS pad driver if used to drive a 10pF capacitive load if the system clock is 500MHz and the output changes with 50% of the clock transitions. Assume pad driver with OD of θ=2.5 and V_{DD}=3.5V

In 0.5u proc t_{REF}=20ps, C_{REF}=4fF,R_{PDREF}=2.5K



Solution: (assume output changes with 50% of clock transitions)

$$P_{DYN} = \frac{f_{CL}}{2} C_L V_{DD}^2 = \frac{5E8}{2} \bullet 10pF \bullet 3.5^2 = 30.5\ mW$$

Note this solution is independent of the OD and the process

Digital Circuit Design

- Hierarchical Design
- Basic Logic Gates
- Properties of Logic Families
- Characterization of CMOS Inverter
- Static CMOS Logic Gates
  - Ratio Logic
- Propagation Delay
  - Simple analytical models
    - FI/OD
    - Logical Effort
  - Elmore Delay
- Sizing of Gates
  - The Reference Inverter
- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
- Other Logic Styles
  - Array Logic
  - Ring Oscillators

---

done

partial

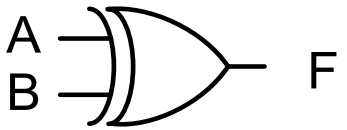
Logic Styles

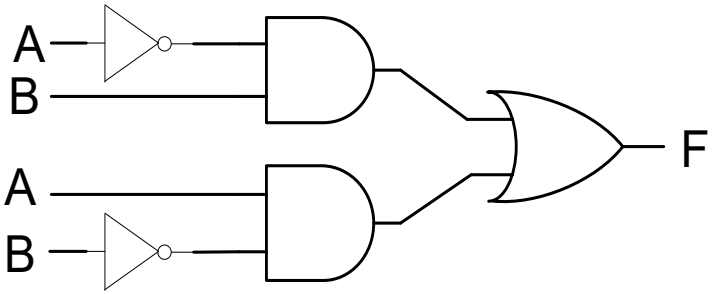
- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

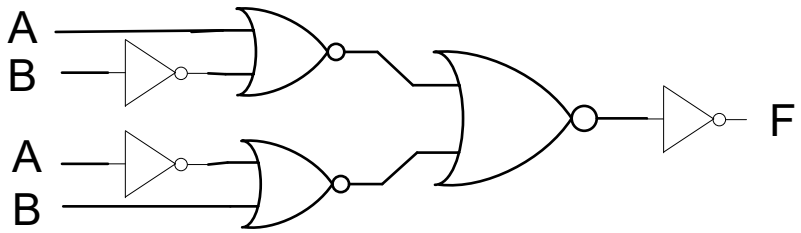
Static CMOS

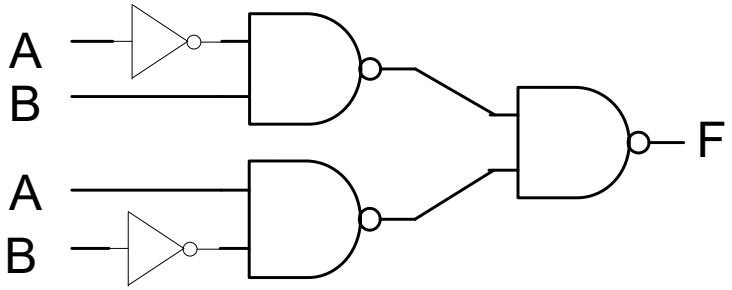
Example: $F=A \oplus B$

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









18 transistors, 4 levels of logic

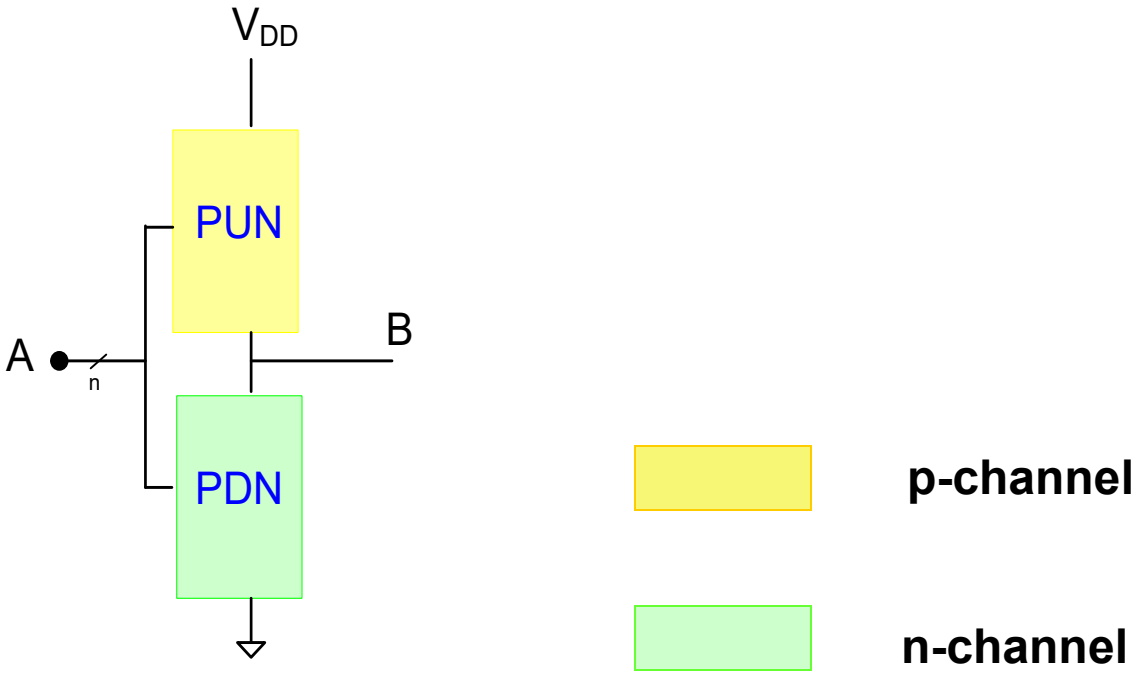
16 transistors, 3 levels of logic

Number of devices is unacceptably large in some applications

Dynamic Power Dissipation can be large, in particular for multiple-input NOR gates because of their large Fan In

Static CMOS Logic Gates

Any multiple-input NAND or NOR gate can be represented as:



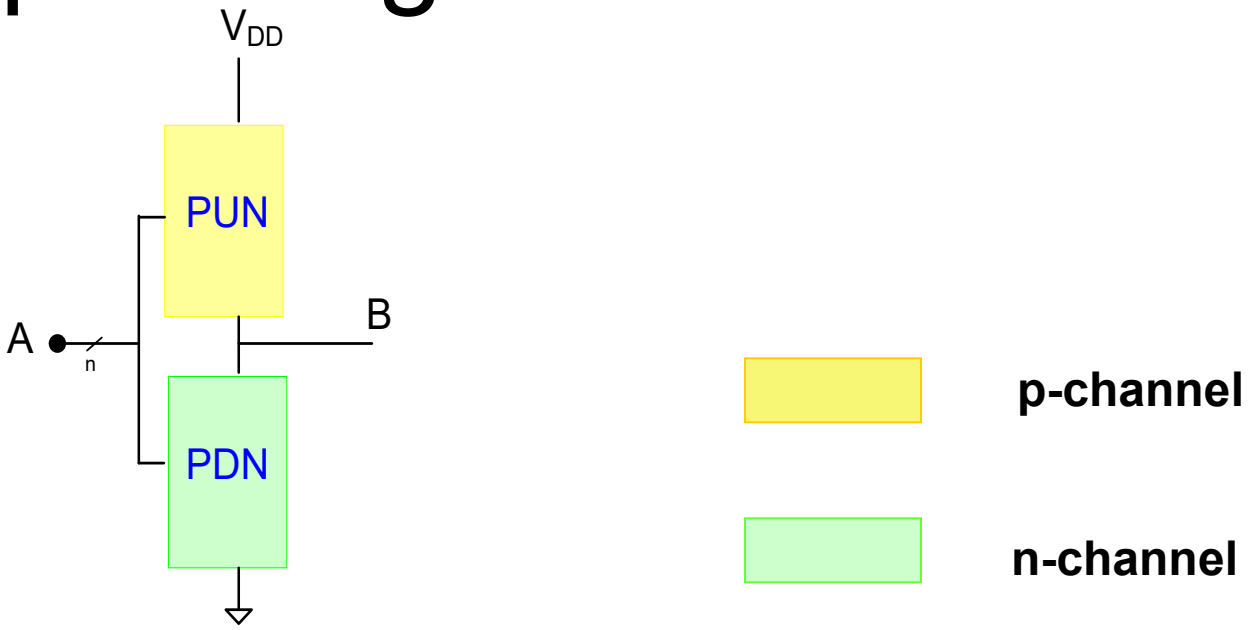
- Implement $\overline{B}$ in PDN
- Implement B in PUN with complimented input variables
- Zero static power dissipation
- $V_H=V_{DD}$, $V_L=0V$ (or V_{SS})
- Complimented input variables often required

Have implemented the logical function twice (once in PU, again in PD) and this is a major contributor to increased area and dynamic power dissipation

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Complex Logic Gates



- Implement B in PDN
- Implement B in PUN with complimented input variables

- Zero static power dissipation
- $V_H=V_{DD}$, $V_L=0V$ (or V_{SS})
- Complimented input variables often required

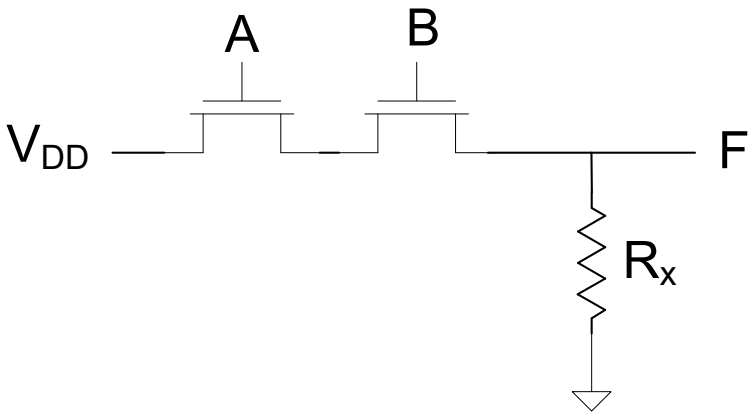
Can reduce the number of levels of logic and the total device count for some functions

Have implemented the logical function twice (once in PU, again in PD) and this is a major contributor to increased area and dynamic power dissipation

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

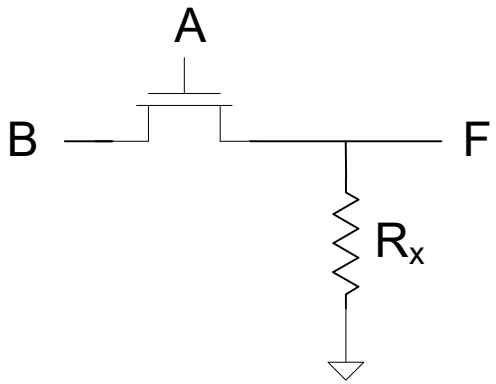
Pass Transistor Logic



F=AB

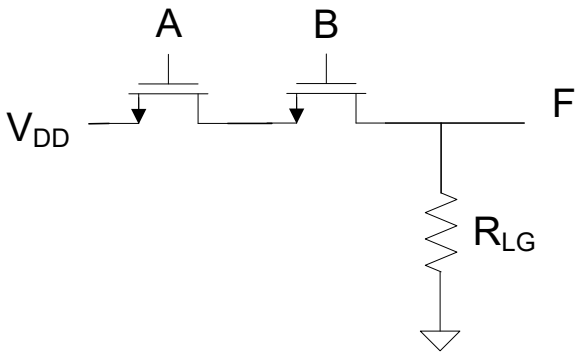
AND Gate

Requires only 3 components



Even simpler AND gate, requires only 2 components

Pass Transistor Logic



F=A•B

Observations about PTL

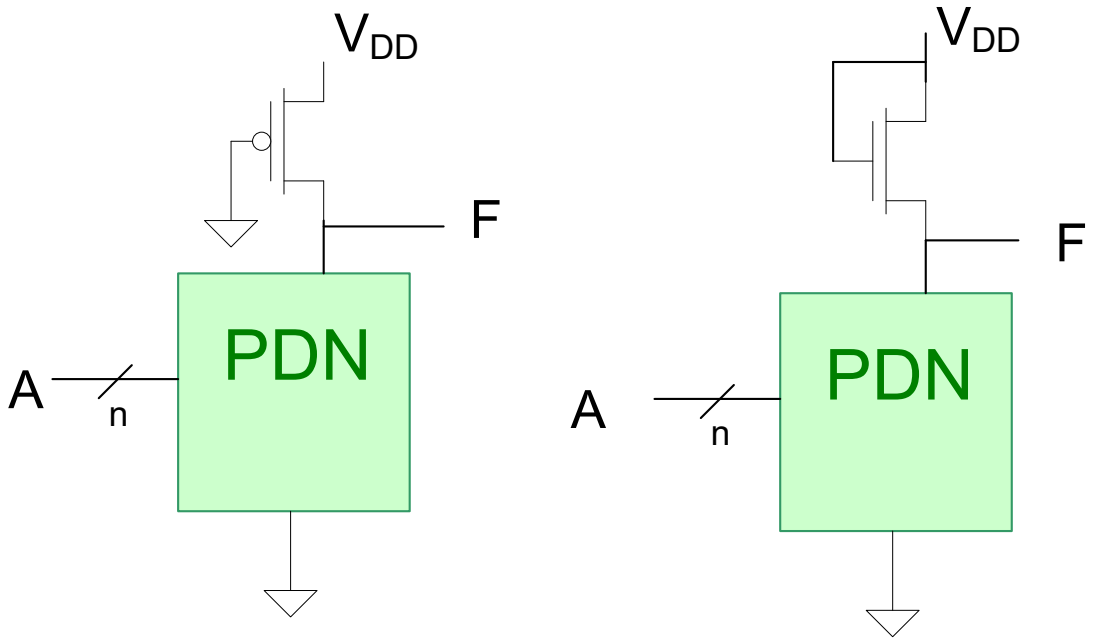
- **Low device count implementation of non inverting function (can be dramatic)**
- **Logic Swing not rail to rail**
- **Static power dissipation not 0 when F high**
- **R_{LG} may be unacceptably large**
- **Slow t_{LH}**
- **Signal degradation can occur when multiple levels of logic are used**
- **Widely used in some applications**
- **Implements basic logic function only once!**
- **Fan In can be very small – so low dynamic power dissipation !**

Is there a way to take advantage of the dynamic power dissipation advantages of a small fan-in without the dramatic energy penalty of a large static power dissipation?

Logic Styles

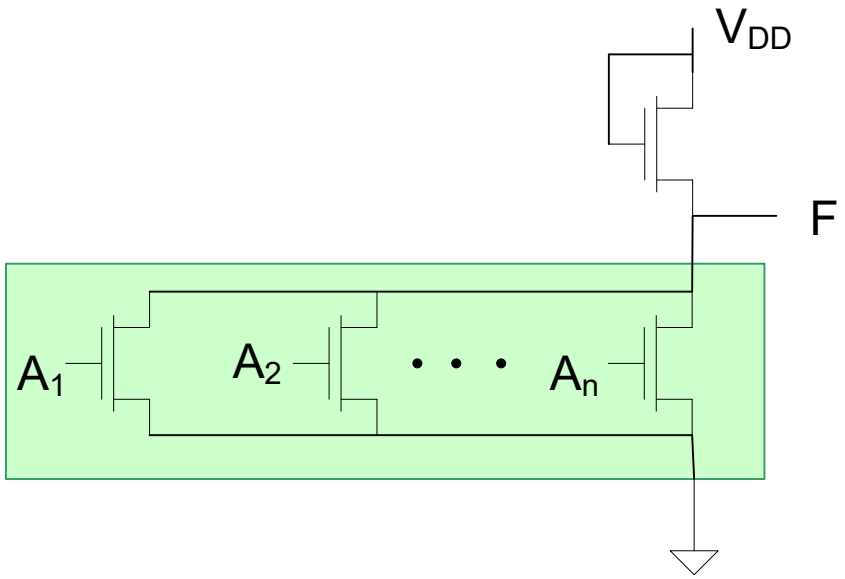
- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Pseudo NMOS Logic



- **May be viewed as a special case of PTL**
- **Ratioed Logic**
- **Static power dissipation not 0 (in PD state)**
- **Often used for really large number of inputs – e.g. NOR**
- **Only one additional transistor for each additional Boolean input**
- **Would be particularly useful for identifying one (or more) of many events that occur very infrequently**

Pseudo NMOS Logic



n could be several hundred or even several thousand

Static power dissipation independent of the number of inputs

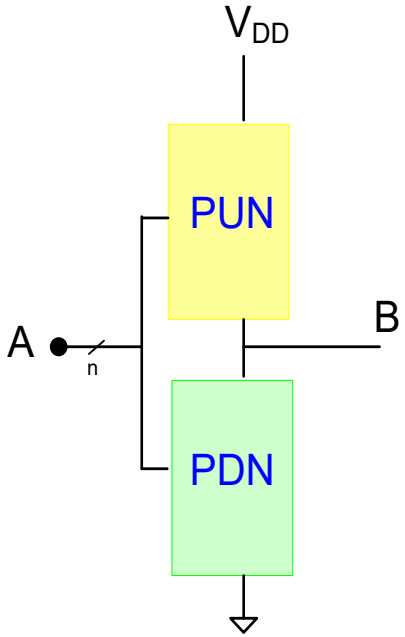
May justify paying the static power dissipation penalty if a large number of inputs are needed, particularly if the conditions to trigger the HL transition occur very rarely

Logic Styles

- Static CMOS
- Complex Logic Gates
- Pass Transistor Logic (PTL)
- Pseudo NMOS
- Dynamic Logic
  - Domino
  - Zipper

Dynamic Logic

- **PTL reduced complexity of either PUN or PDN to single "resistor"**
- **PTL relaxed requirement of all n-channel or all p-channel devices in PUN/PDN**



What is the biggest contributor to area?

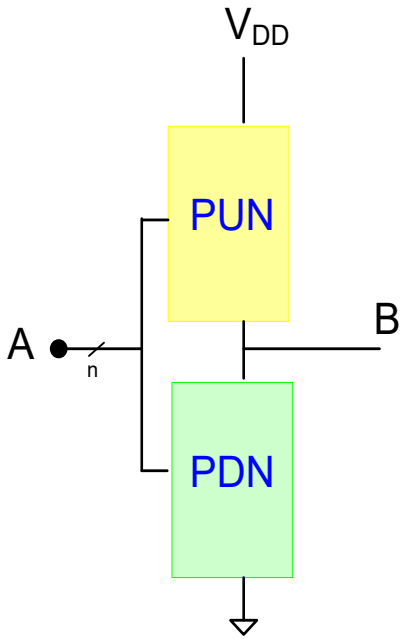
PUN (3X active area for inverter, more for NOR gates, and Well)

What is biggest contributor to dynamic power dissipation ?

PUN and is responsible for approximately 75% of the dynamic power dissipation in equal rise/fall inverter, and much more in NOR gates !

Can the PUN be eliminated W/O compromising signal levels and power dissipation?

Dynamic Logic

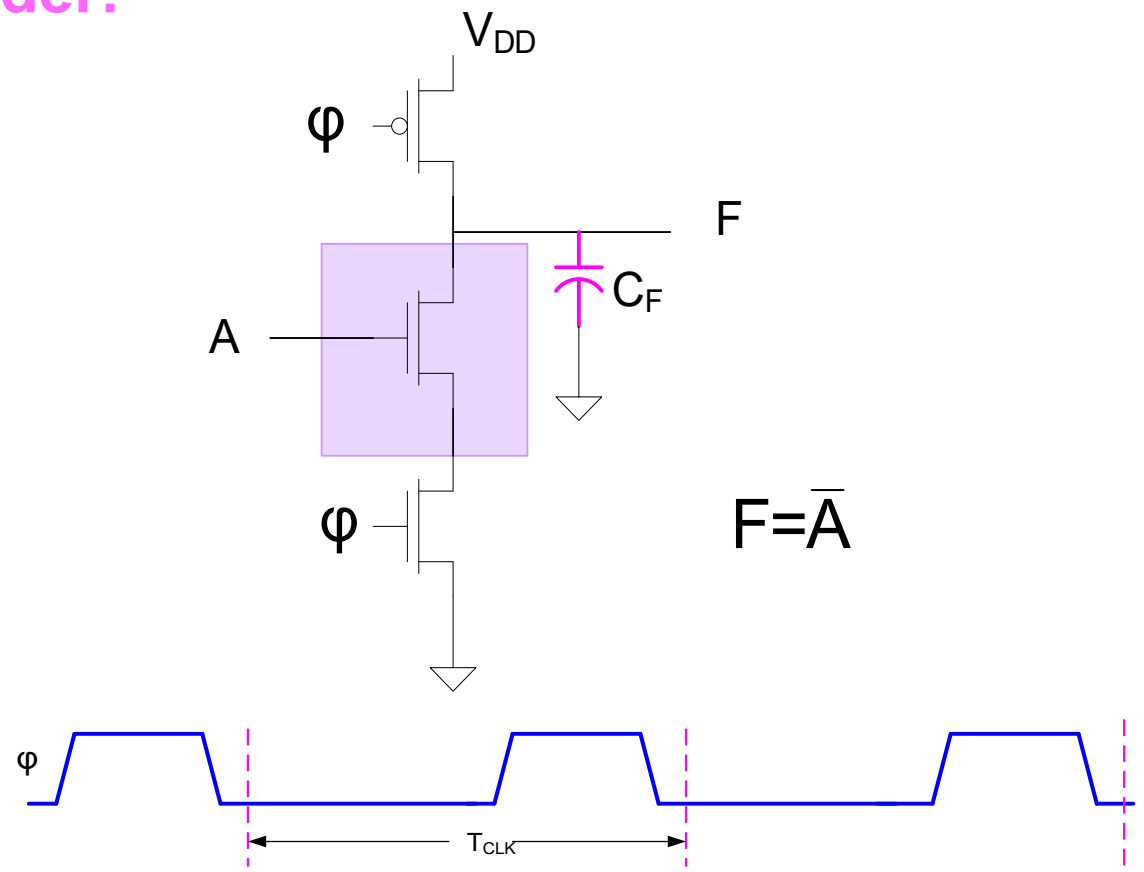


Can the PUN be eliminated W/O compromising signal levels and power dissipation?

Benefits could be most significant !

Dynamic Logic

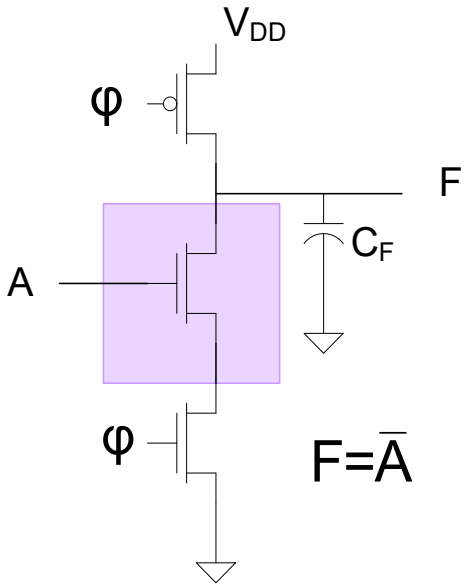
Consider:

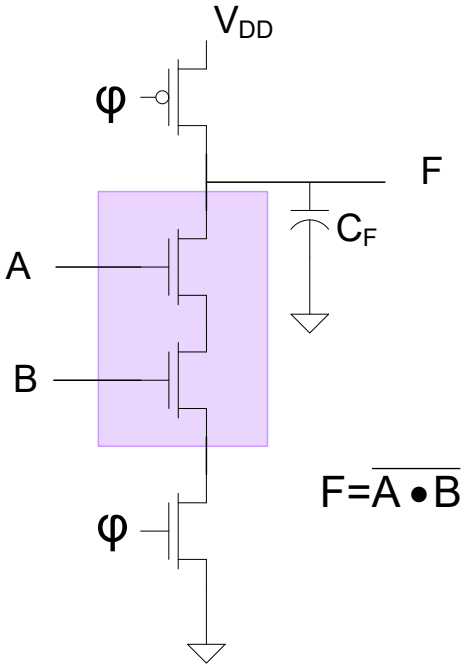


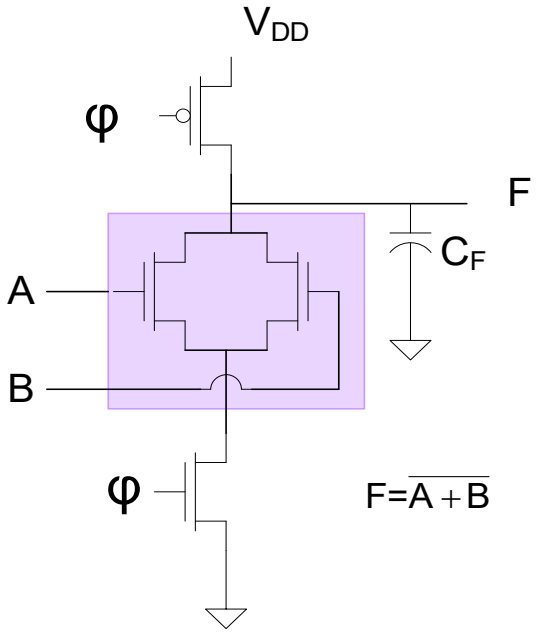
Precharges C_F to "1" when φ is low
F either stays high if output is to be high or changes to low on evaluation
C_F is usually the parasitic capacitances on the node (drain diffusion and gates)

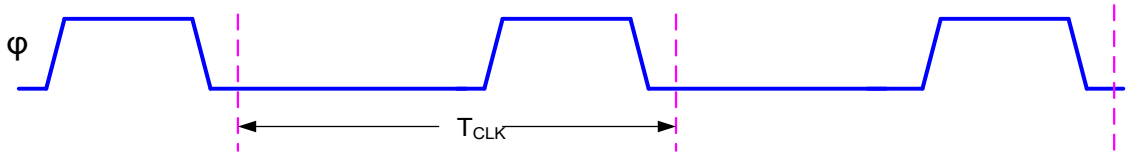
Dynamic Logic

Consider:





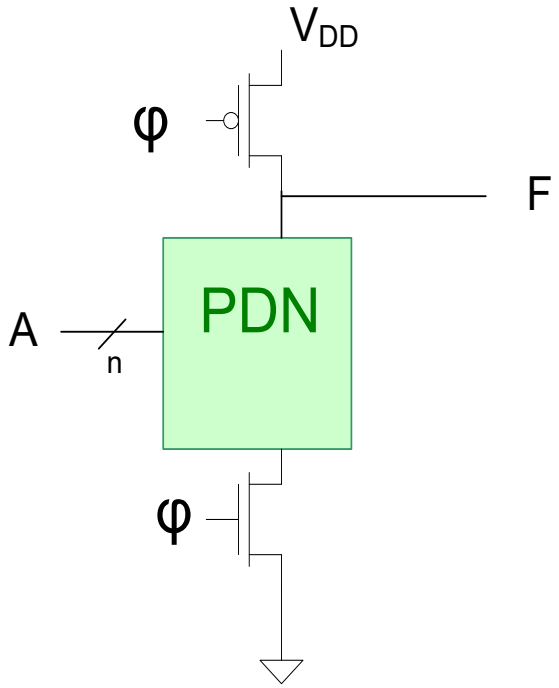




- **Termed Dynamic Logic Gates**
- **Parasitic capacitors actually replace C_F**
- **If Logic Block is n-channel, will have rail to rail swings**
- **Logic Block is simply a PDN that implements $\overline{F}$**

Dynamic Logic

Basic Dynamic Logic Gate

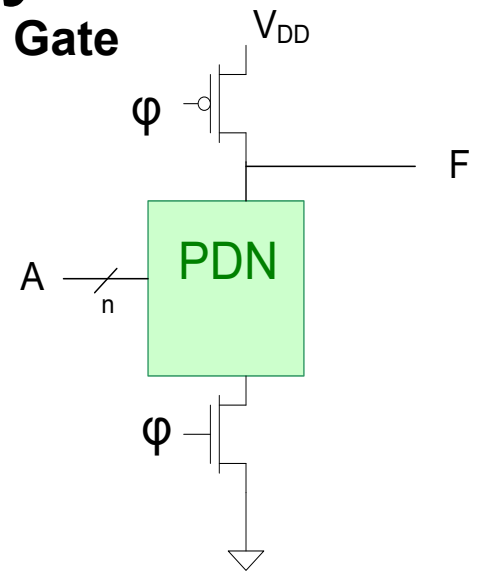


Any of the PDNs used in complex logic gates would work here !

- **Have eliminate the PUN !**
- **Ideally will have a factor of 4 or more reduction in C_{IN}**
- **Ideally will have a factor of 4 or more reduction in dynamic power dissipation relative to that of equal rise/fall !**
- **Ideally will have a factor of 2 reduction in dynamic power dissipation relative to that of minimum size!**

Dynamic Logic

Basic Dynamic Logic Gate



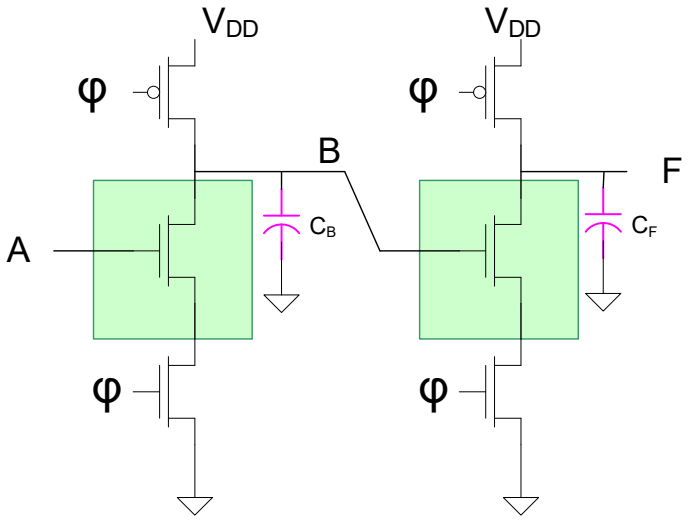
Advantages:

- **Lower dynamic power dissipation (Ideally 4X)**
- **Improved speed (ideally 4X)**

Limitations:

- **Output only valid during evaluate state**
- **Need to route a clock**
  (and this dissipates some power)
- **Premature Discharge !**
- **More complicated**
- **Charge storage on internal nodes of PDN**
- **No Static hold if output H**

Dynamic Logic



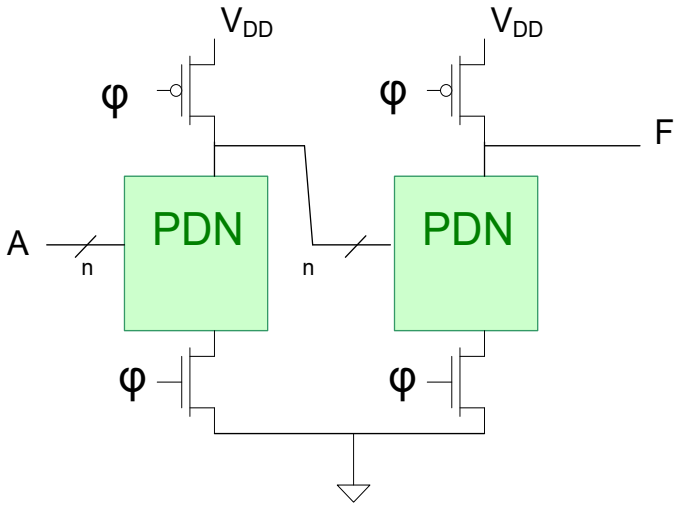
Premature Discharge Problem

B will be pulled high during the pre-charge state and try to discharge C_F thus pulling F low

If input A is high, then if F goes low at the start of the evaluate cycle, there is no way to recover a high output later in the evaluate phase - i.e. there may be a boolean error!.

Can not reliably cascade dynamic logic gates !

Dynamic Logic



Premature Discharge Problem

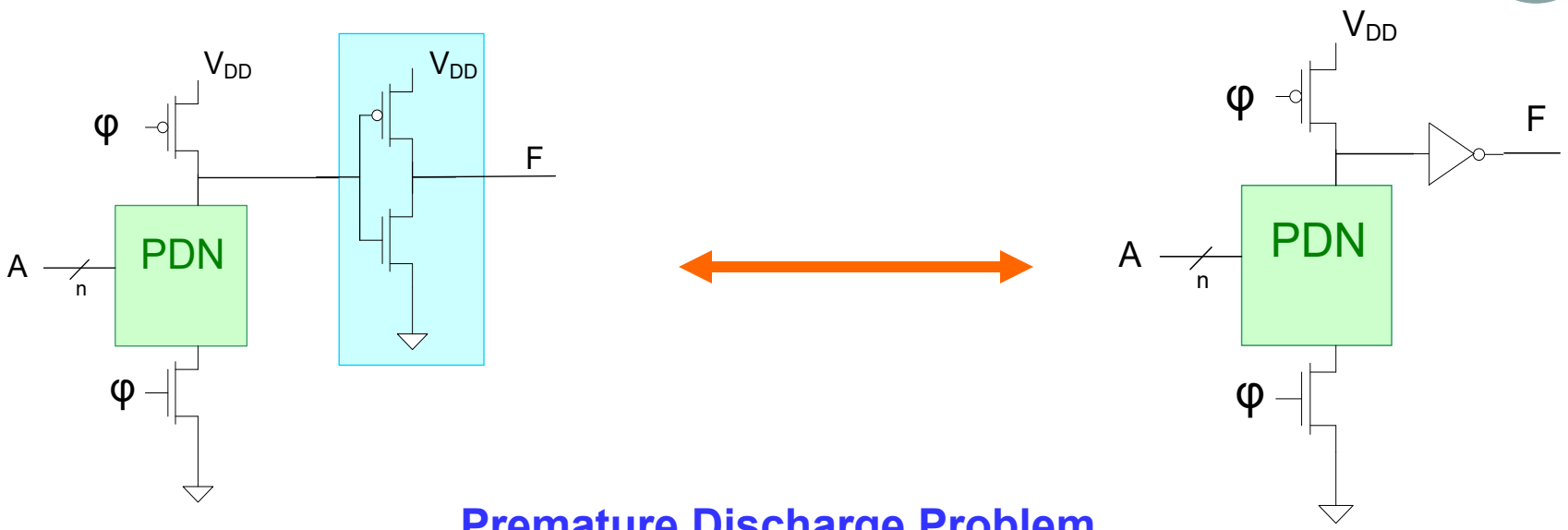
This problem occurs when any inputs to an arbitrary dynamic logic gate create an R_{PD} path in the PDN during at the start of the evaluate phase that is not to pull down later in that evaluate phase

How can this problem be fixed?

Precharging to the low level all inputs to a PDN that may change to the high state later in the evaluate cycle (called domino)

Alternating gates with n-channel and p-channel pull networks (Zipper Logic)

Dynamic Logic



Premature Discharge Problem

Adding an inverter at the output will cause F to precharge low so it can serve as input to subsequent gate w/o causing premature discharge

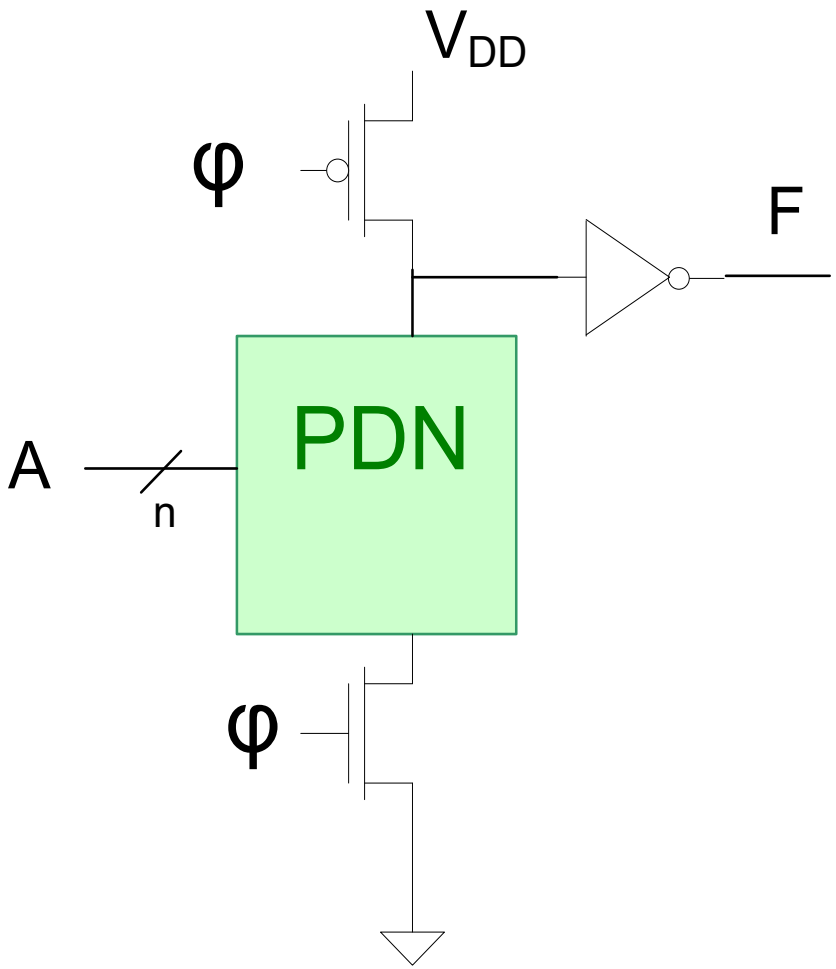
Implement F instead of $\overline{F}$ in the PDN

Termed Domino Logic

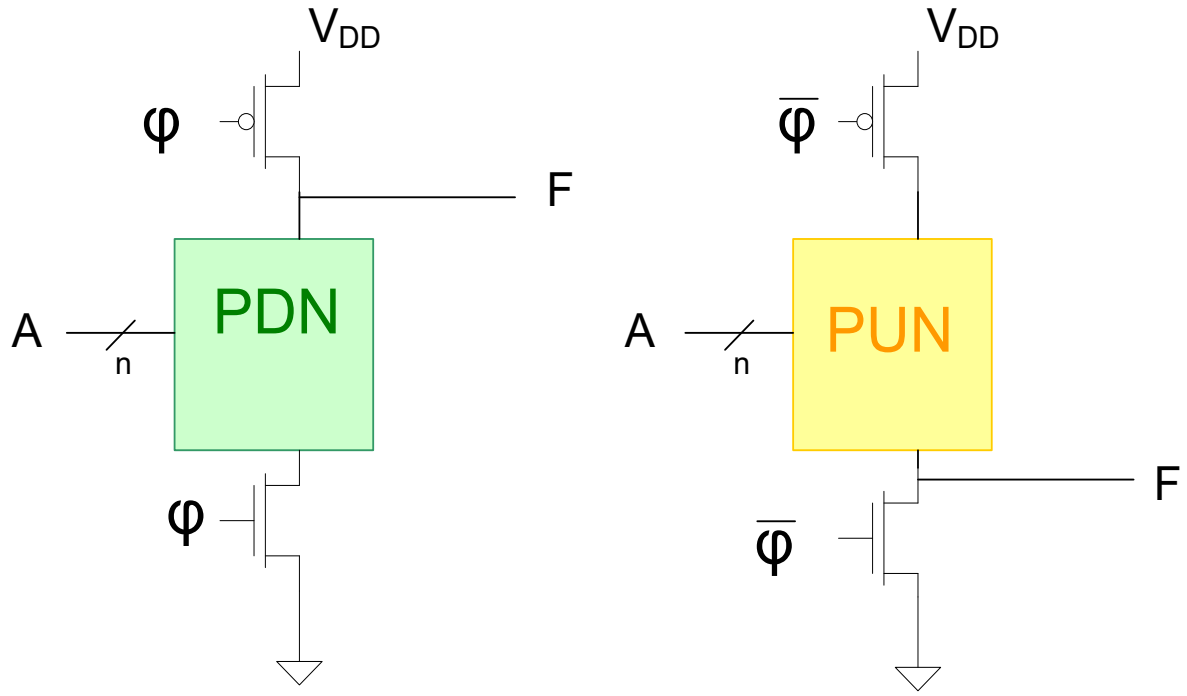
Some additional dynamic power dissipation in the inverter

Some additional delay during the evaluate state in inverter

Domino Logic

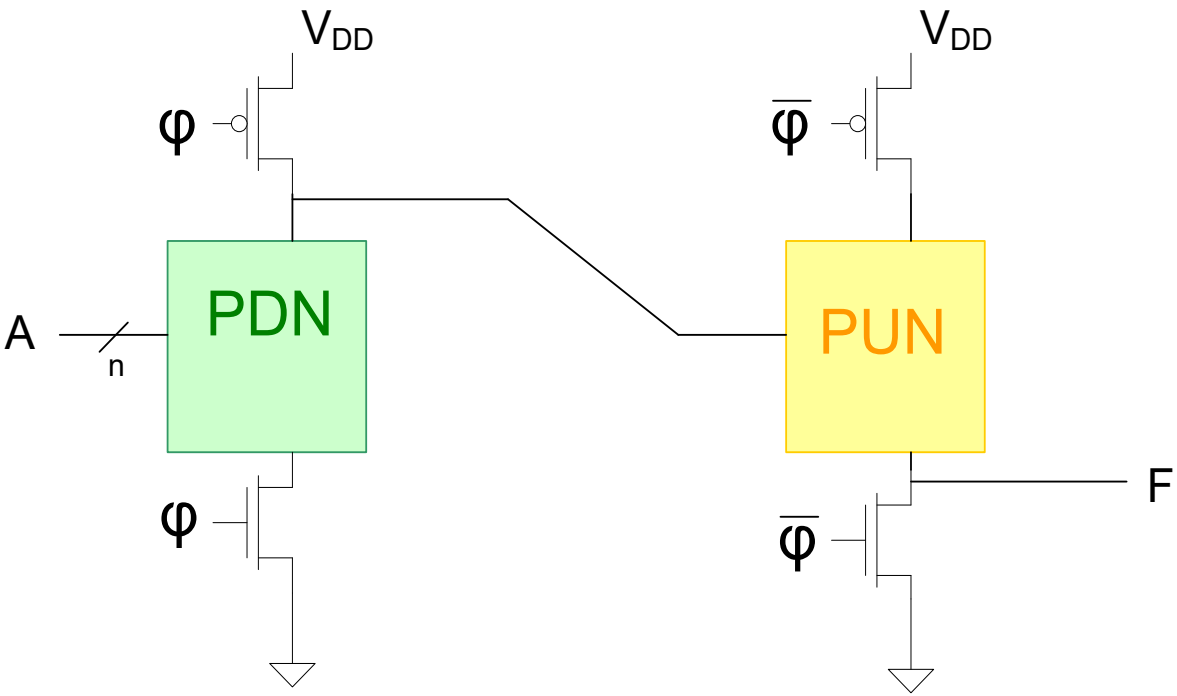


Dynamic Logic



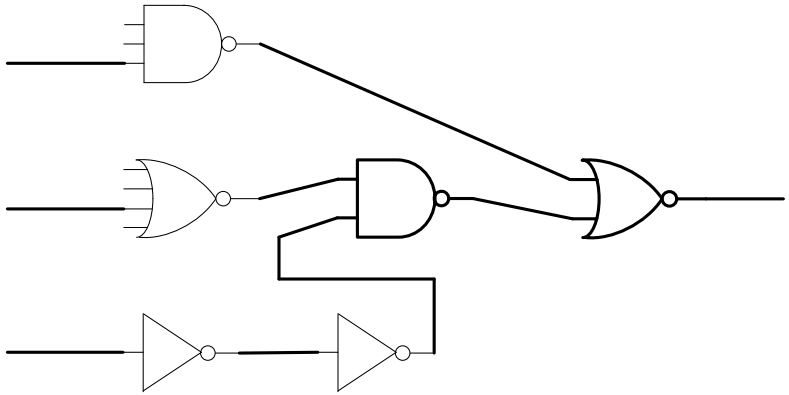
- **p-channel logic gate will pre-charge low**
- **Phasing of PUN and PDN networks is reversed**
- **Some performance loss with p-channel logic devices**
- **Direct coupling between alternate type dynamic gates is possible without causing a premature discharge problem**

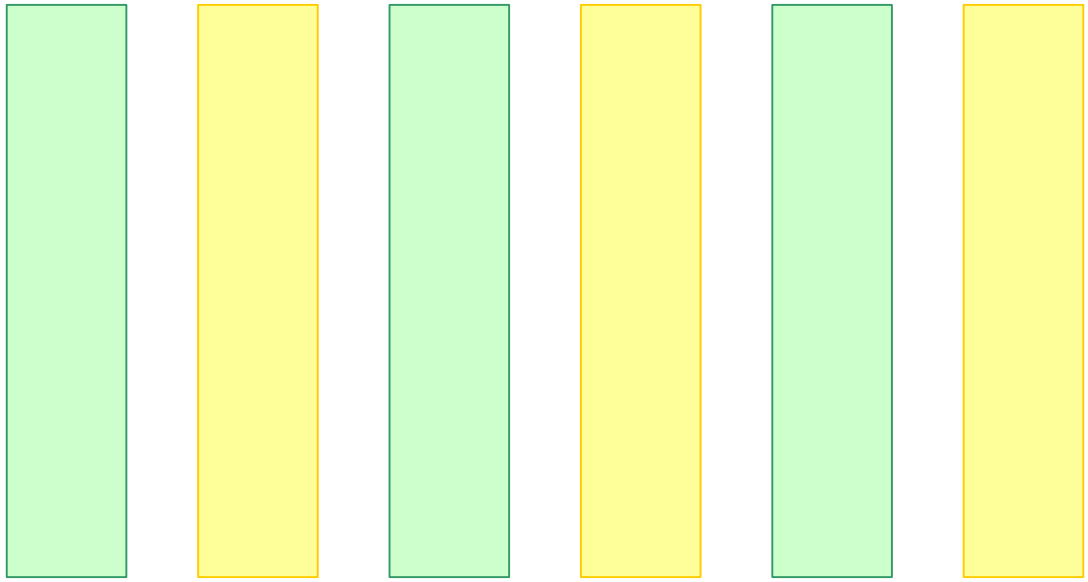
Dynamic Logic



Direct coupling between alternate type dynamic gates

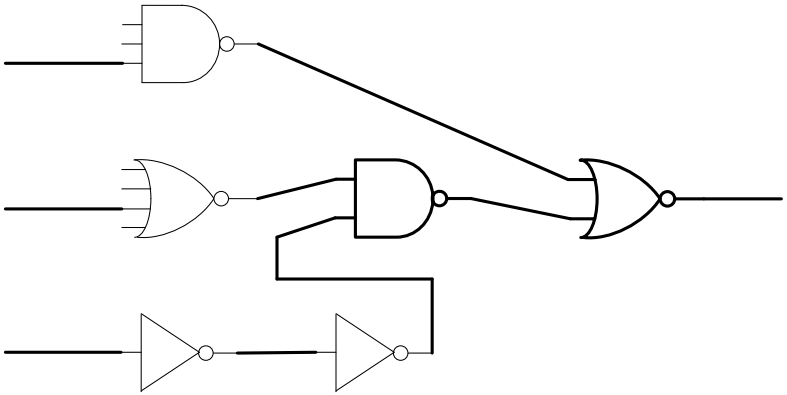
Zipper Logic

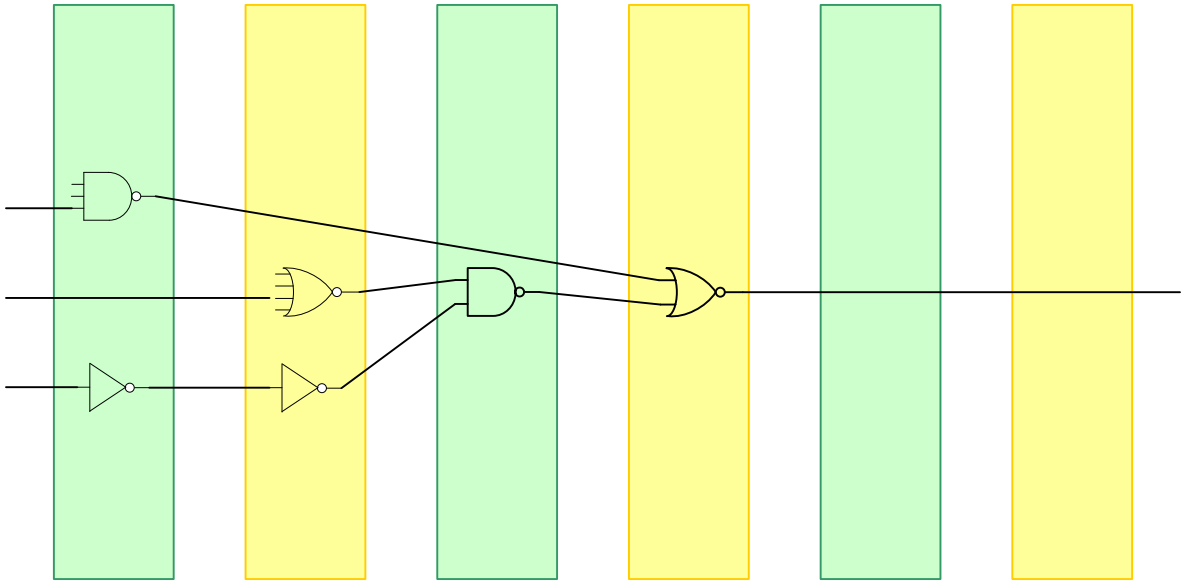




Map gates to appropriate precharge type

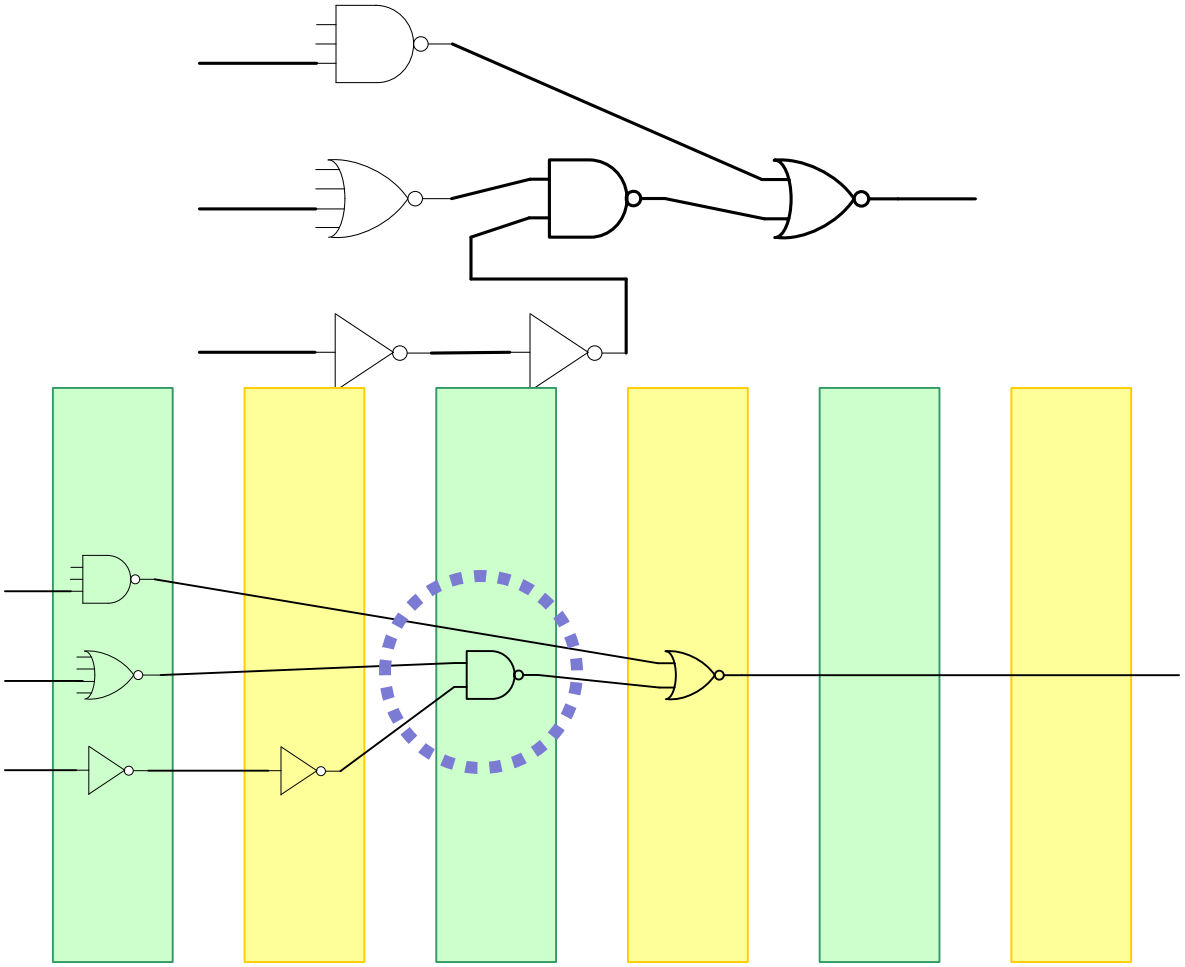
Zipper Logic





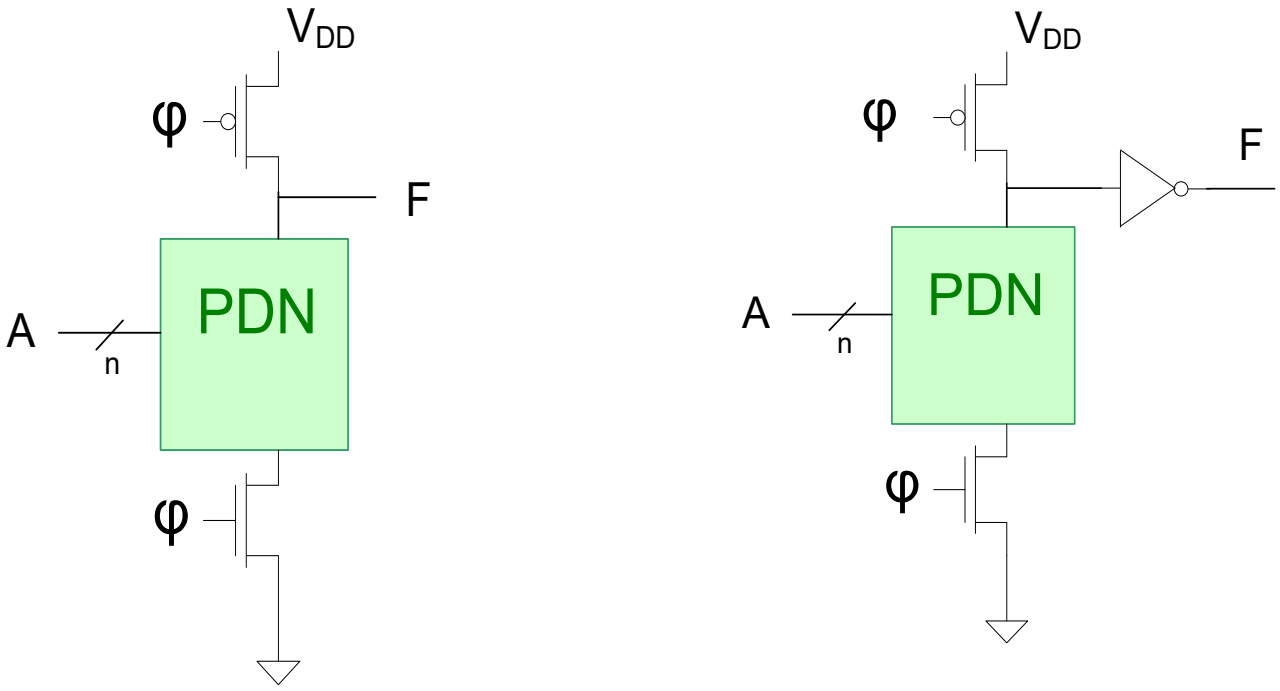
Acceptable Implementation in Zipper

Zipper Logic



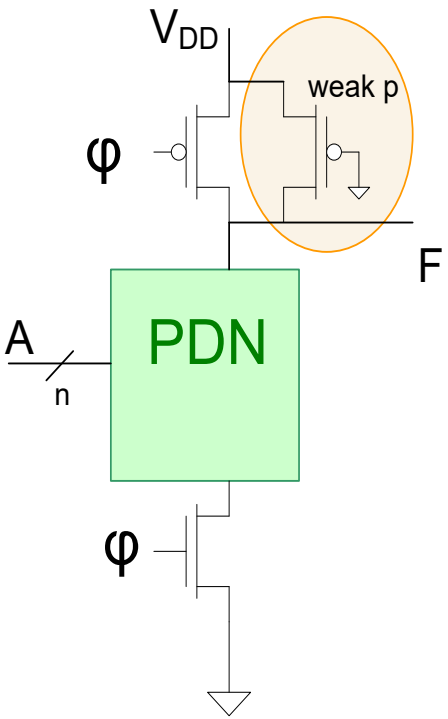
Unacceptable Implementation in Zipper
- Premature discharge at output of 2-input NAND

Static Hold Option

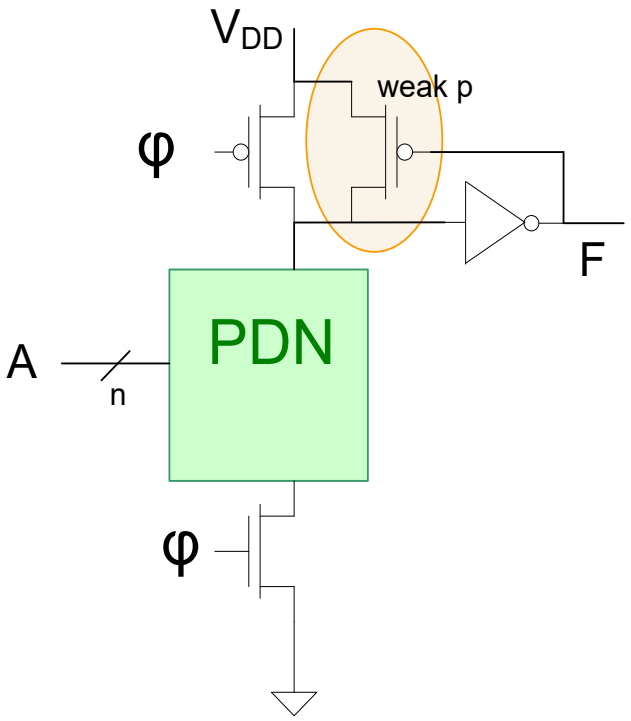


If not clocked, charge on upper node of PDN will drain off causing H output to degrade

Static Hold Option

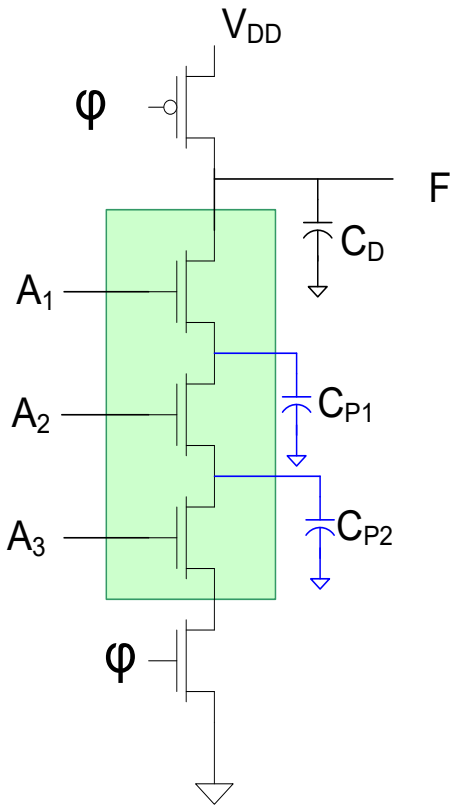


- **weak p will hold charge**
- **size may be big (long L)**
- **some static power dissipation**
- **can use small current source**
- **sometimes termed “keeper”**



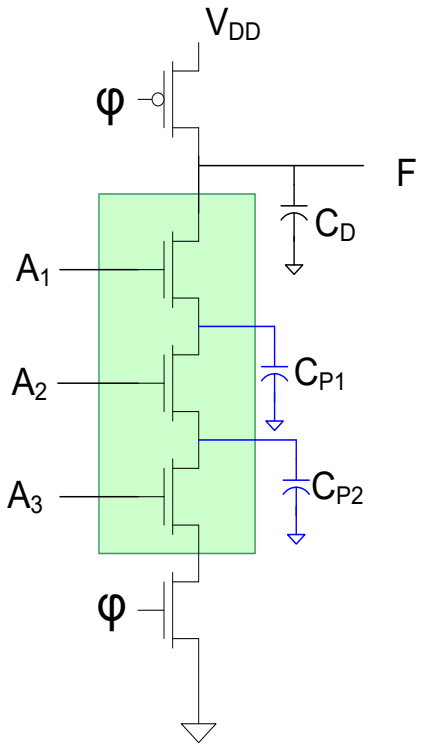
- **weak p will hold charge**
- **size may be big (long L)**
- **can eliminate static power with domino**
- **sometimes termed “keeper”**

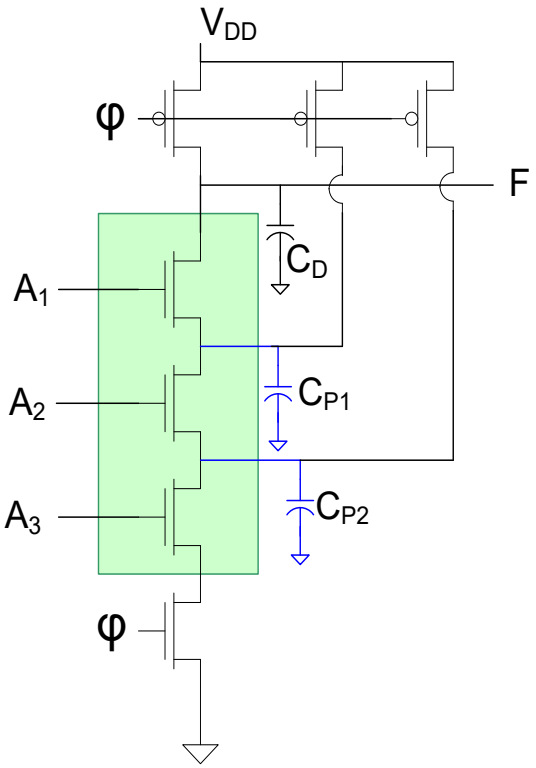
Charge stored on internal nodes of PDN



If voltage on C_{P1} and C_{P2} was 0V on last evaluation, these may drain charge (charge redistribution) on C_P if output is to evaluate high (e.g. On last evaluation $A_1=A_2=A_3=H$, on next evaluation $A_3=L$, $A_1=A_2=H$.)

Charge stored on internal nodes of PDN





Can precahrge internal nodes to eliminate undesired charge redistribution

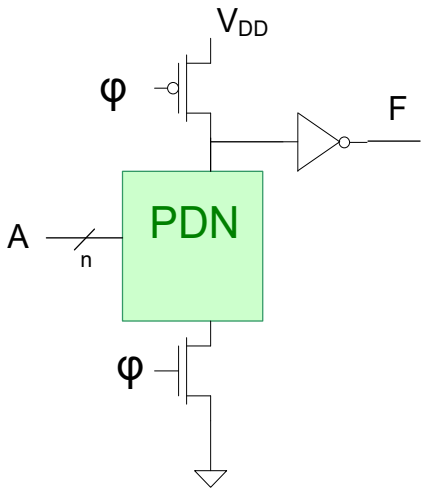
Dynamic Logic

Many variants of dynamic logic are around

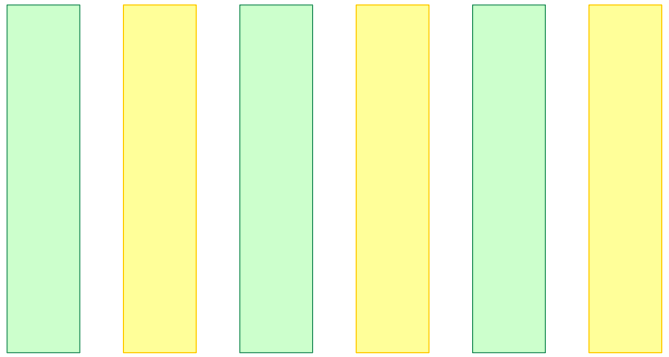
- **Domino**
- **Zipper**
- **Ratio-less 2-phase**
- **Ratio-less 4-phase**
- **Output Prediction Logic**
- **Fully differential**
- **…**

Benefits disappear, however, when interconnect (and diffusion) capacitances dominate gate capacitances

Future of Dynamic Logic



Domino



Zipper

Dynamic logic will likely disappear in deep sub-micron processes because interconnect parasitics will dominate gate parasitics

Other types of Logic (list is not complete and some have many sub-types)

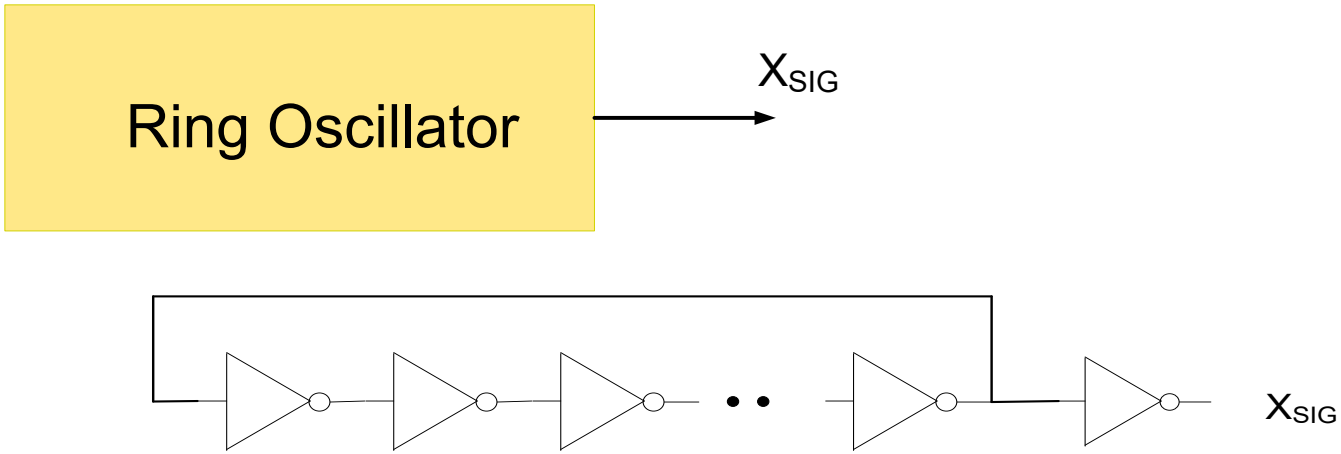
From Wikipedia:

B
BiCMOS
C
CMOS
Cascode Voltage Switch Logic
Clocked logic
Complementary Pass-transistor Logic
Current mode logic
Current steering logic
D
Differential TTL
Diode logic
Diode–transistor logic
Domino logic
Dynamic logic (digital logic)
E
Emitter-coupled logic
F
Four-phase logic
G
Gunning Transceiver Logic
H
HMOS
HVDS
High-voltage differential signaling
I
Integrated injection logic
L
LVDS
Low-voltage differential signaling
Low-voltage positive emitter-coupled logic
M
Multi-threshold CMOS
N
NMOS logic
P
PMOS logic
Philips NORbits
Positive emitter-coupled logic
R
Resistor-transistor logic
S
Static logic (digital logic)
T
Transistor–transistor logic

Digital Building Blocks

- Shift Registers
- Sequential Logic
- Shift Registers (stack)
- Array Logic
- Memory Arrays

Ring Oscillators



- **Odd number of stages will oscillate (even will not oscillate)**
- **Waveform nearly a square wave if n (number of stages) is large**
- **Output will slightly imbalance ring and device sizes can be compensated if desired**
- **Usually use a prime number (e.g. 31)**
- **Number of stages usually less than 50 (follow by dividers)**
- **Frequency highly sensitive to process variations and temperature**

$$f_{OSC} \cong \frac{1}{nt_{PROP}}$$

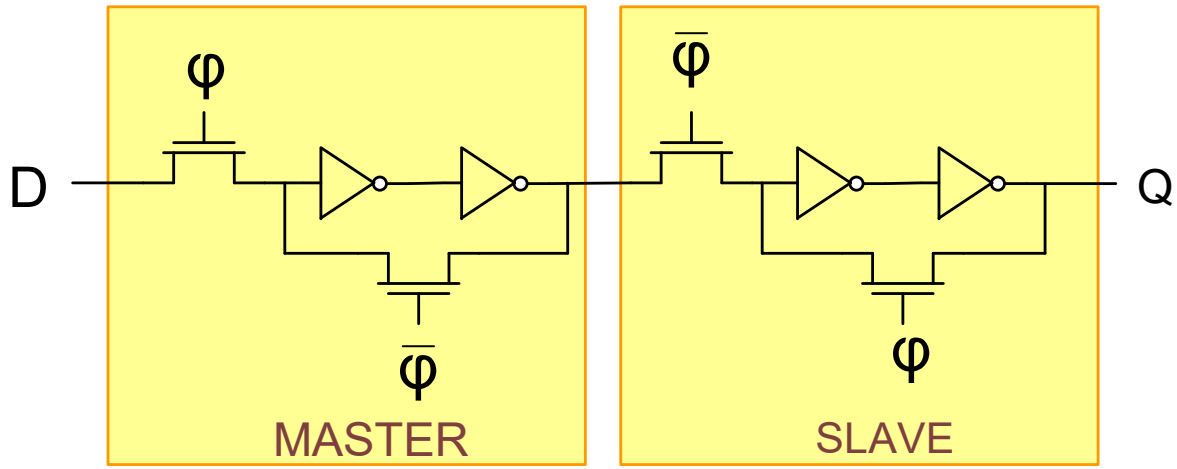
- **n is the number of stages**
- **t_{PROP} is the propagation delay of a single stage (all assumed identical)**

Sequential Logic Circuits

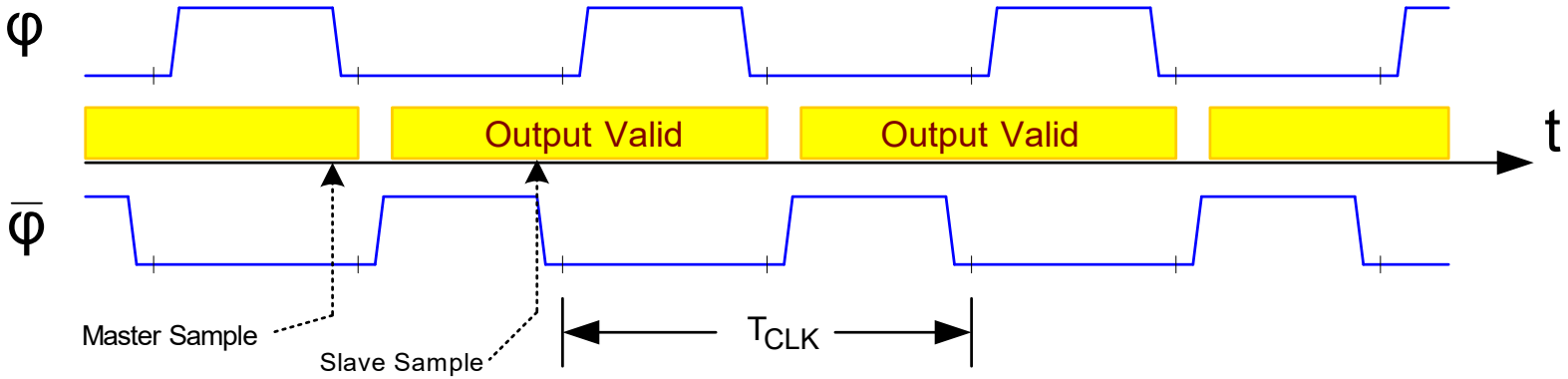
- **Flip Flops needed for sequential logic circuit**
- **Only one type of flip flop is required**
- **Invariably require clocked edge-triggered master-slave flop flops**
- **Flip flop circuits can be very simple**
- **Flip flops are part of Standard Cell Libraries**

Flip Flops

Master-Slave Edge-triggered D Flip Flop



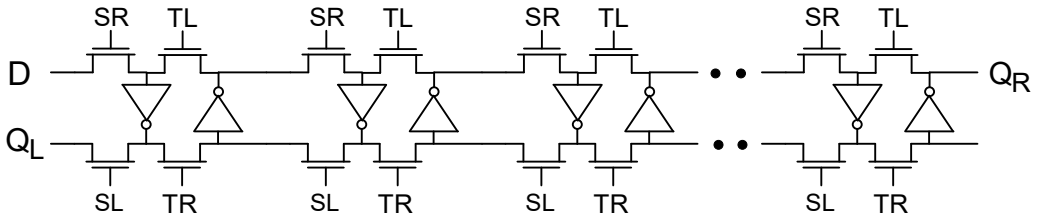
Timing Diagram

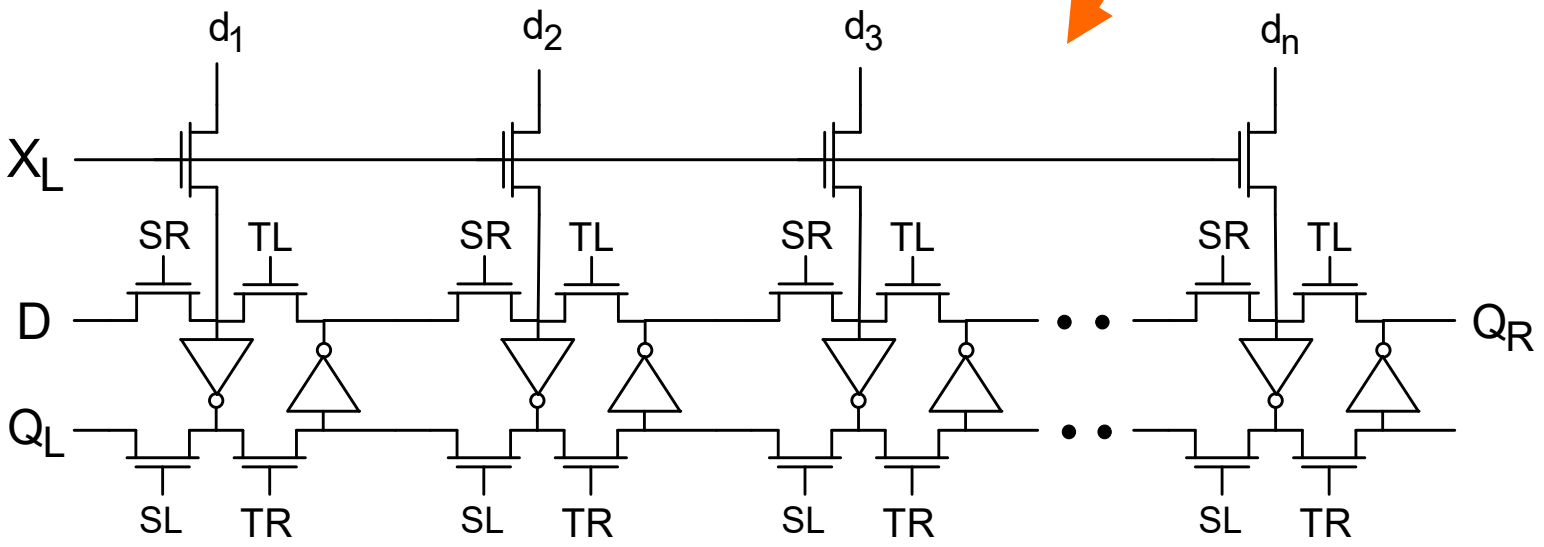


- **12 transistors (but will work with 10)**
- **Many other simple D Flip-flops exist as well**

Shift Registers

Dynamic Shift Register





n-bit Parallel-Load, Parallel-Read Bidirectional Dynamic Shift Register

- **Useful for Parallel to Serial and Serial to Parallel Conversion**
- **Can be put in static hold state if T_L and T_R replaced with HCTL and HCTL**

Array Logic

- Array logic is often used for sections of logic that may change later in the design or that will be changed for different variants of a product
- FPGA are a special case of array logic
- Can personalize array logic with only one layer of metal
  - Very quick turn-around and low incremental costs (as few as one additional mask)

Array Logic

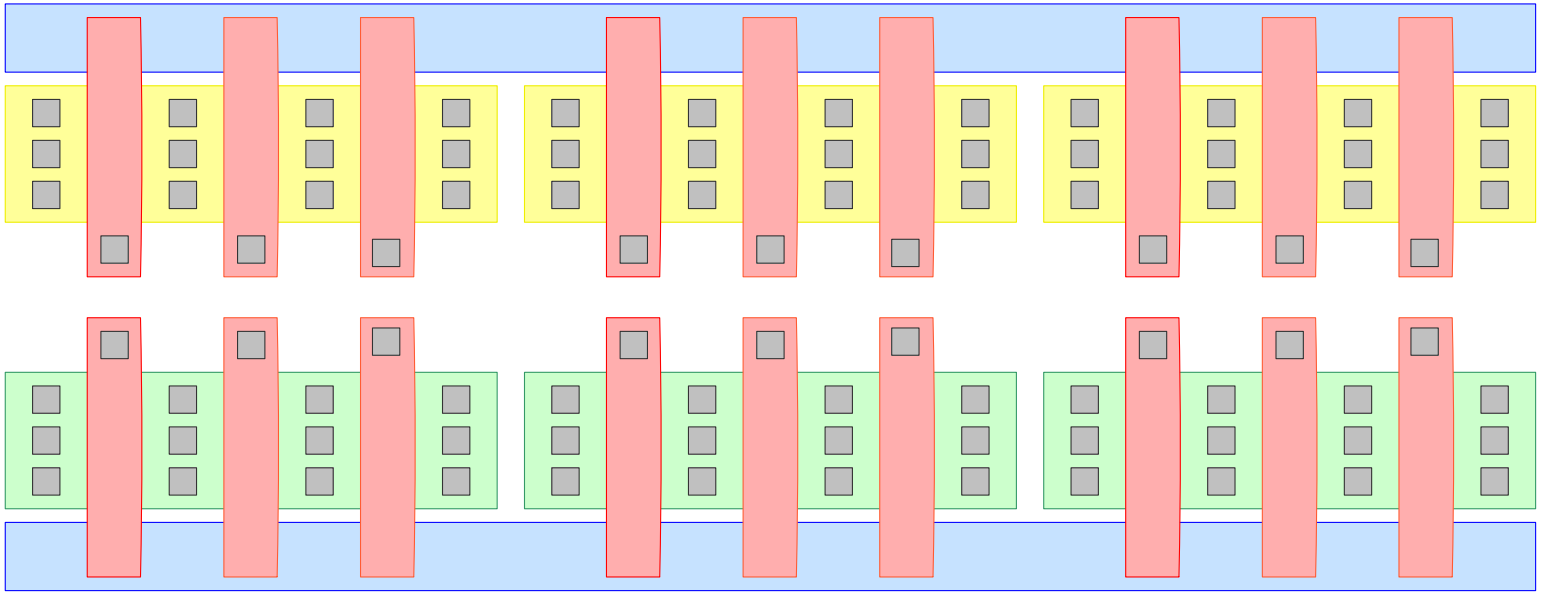
Will consider only two types

- Gate Array
- Sea of Gates

Variants of the following approach are possible depending upon process but this will convey the basic concepts

Array Logic

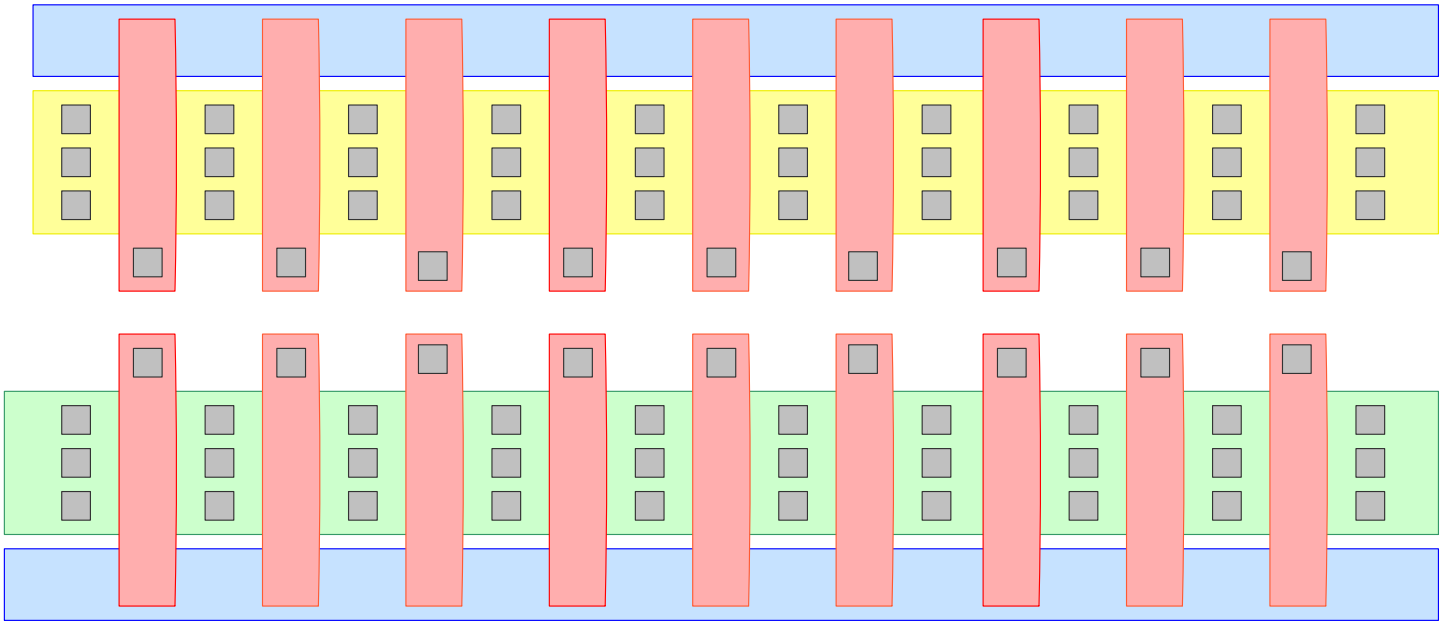
Gate Array



- **Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2) (3 simple masks)**
- **Upper and lower metal shown actually lie above poly and are automatically present**
- **Assume upper M1 is V_{DD} and lower M1 is V_{SS}**
- **Array can be very large**
- **Routing channels between segments of array**

Array Logic

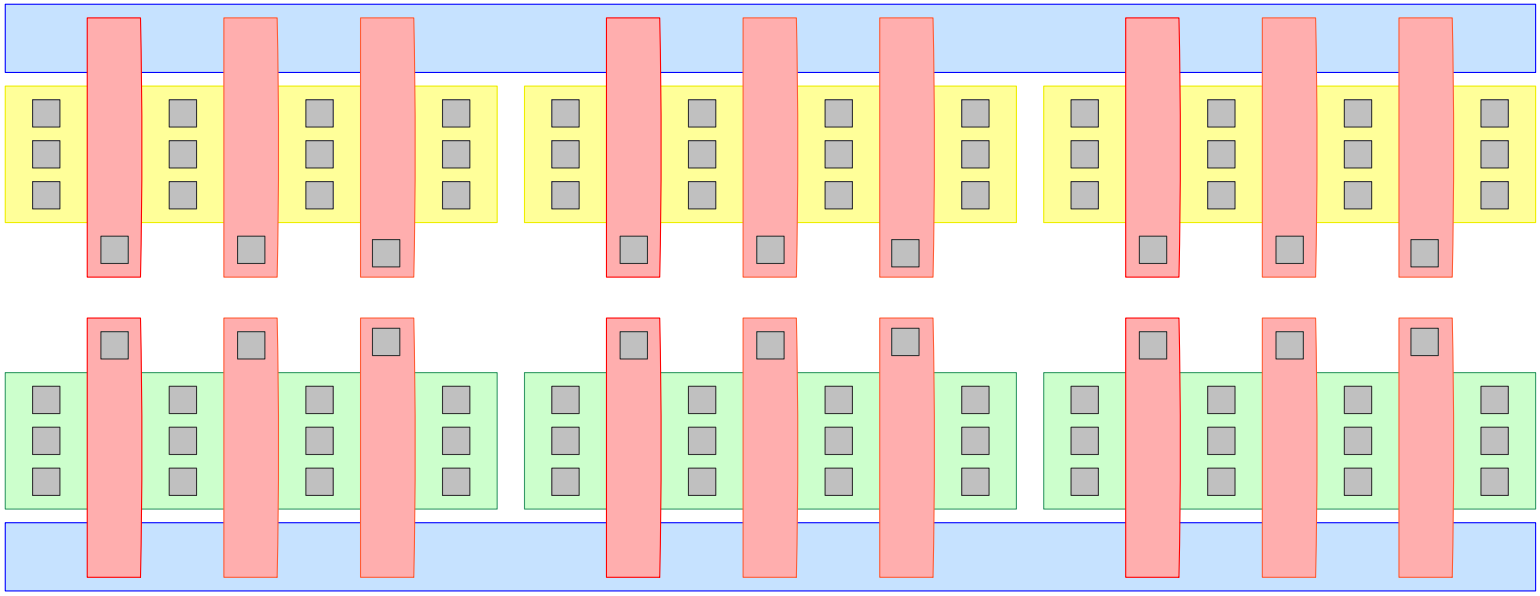
Sea of Gates



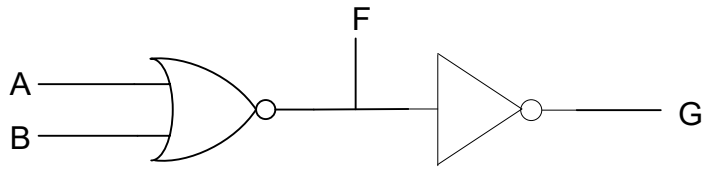
- **Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2) (3 simple masks)**
- **Upper and lower metal shown actually lie above poly and are automatically present**
- **Assume upper M1 is V_{DD} and lower M1 is V_{SS}**
- **Array can be very large**
- **Routing channels between segments of array**

Array Logic

Gate Array



Example:

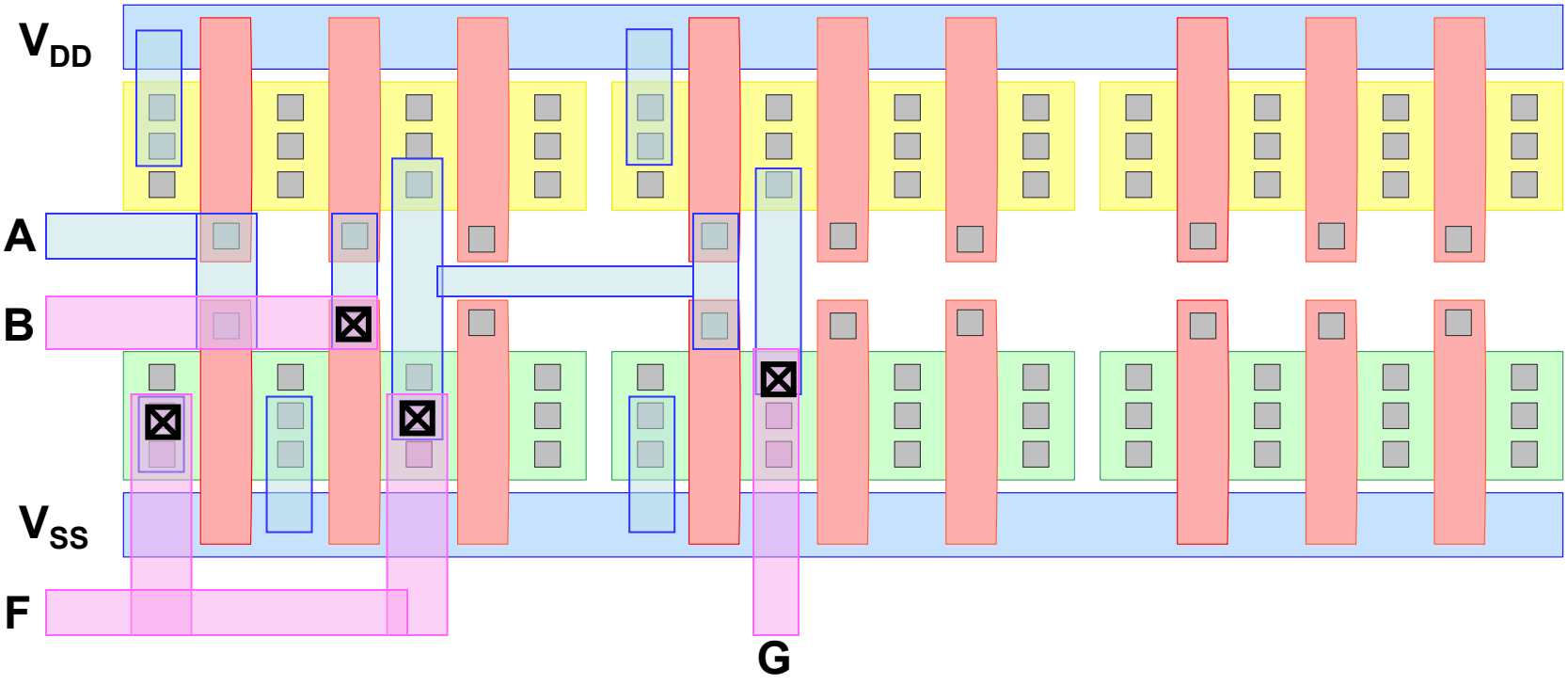


Via (M1 to M2)

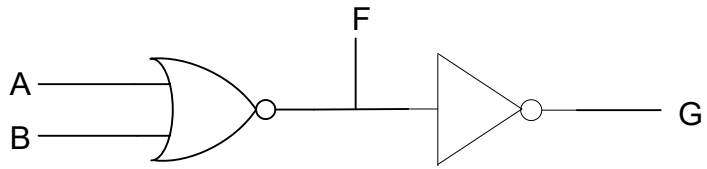
Contact (M1 to diff,Poly)

Array Logic

Gate Array



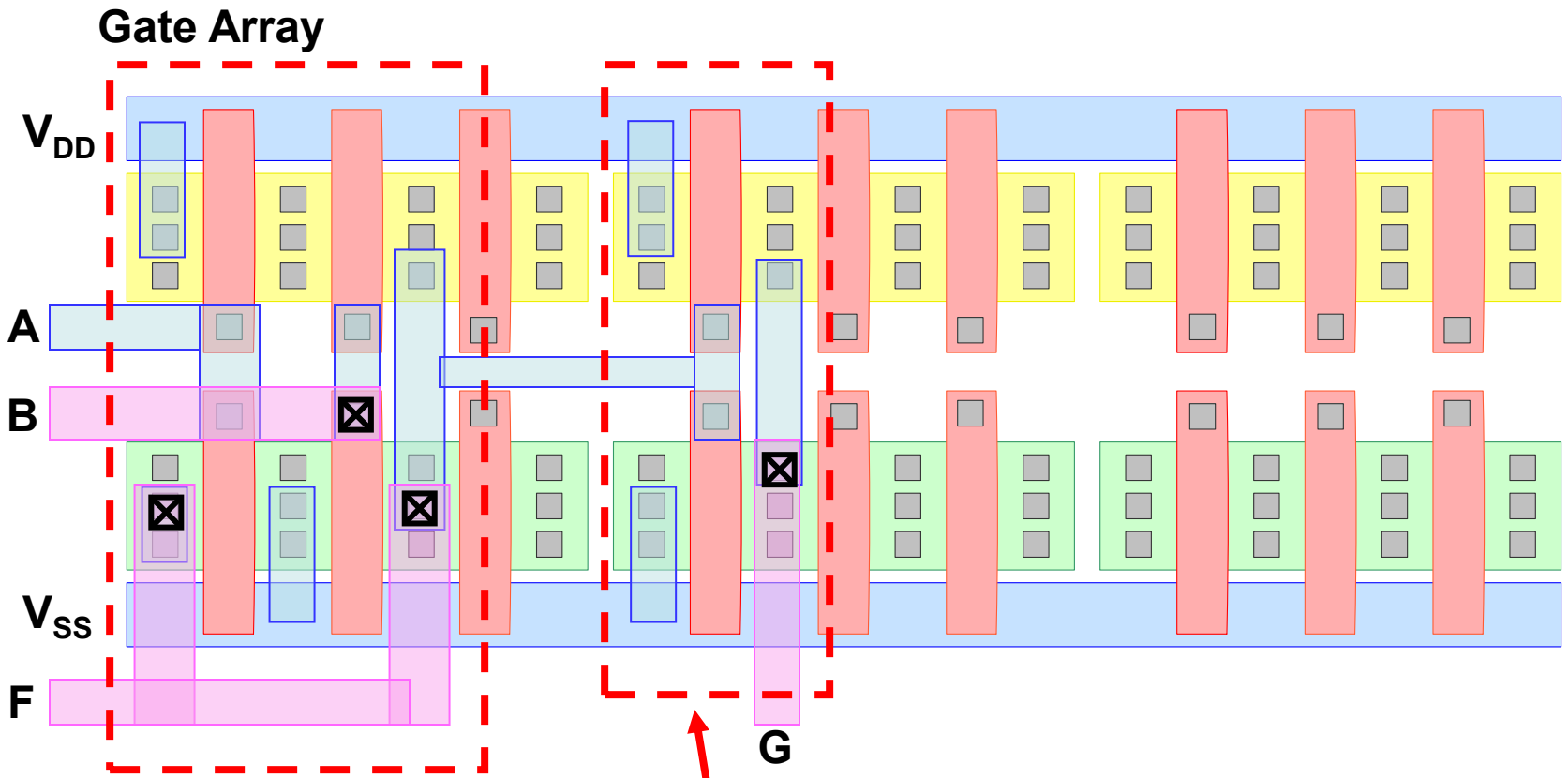
Example:



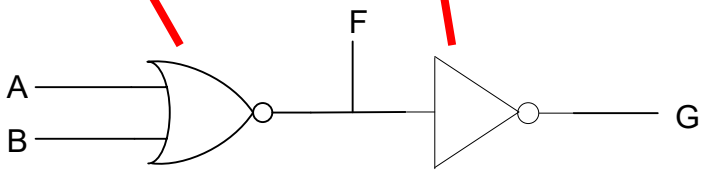
Via (M1 to M2)

Contact (M1 to diff,Poly)

Array Logic



Example:

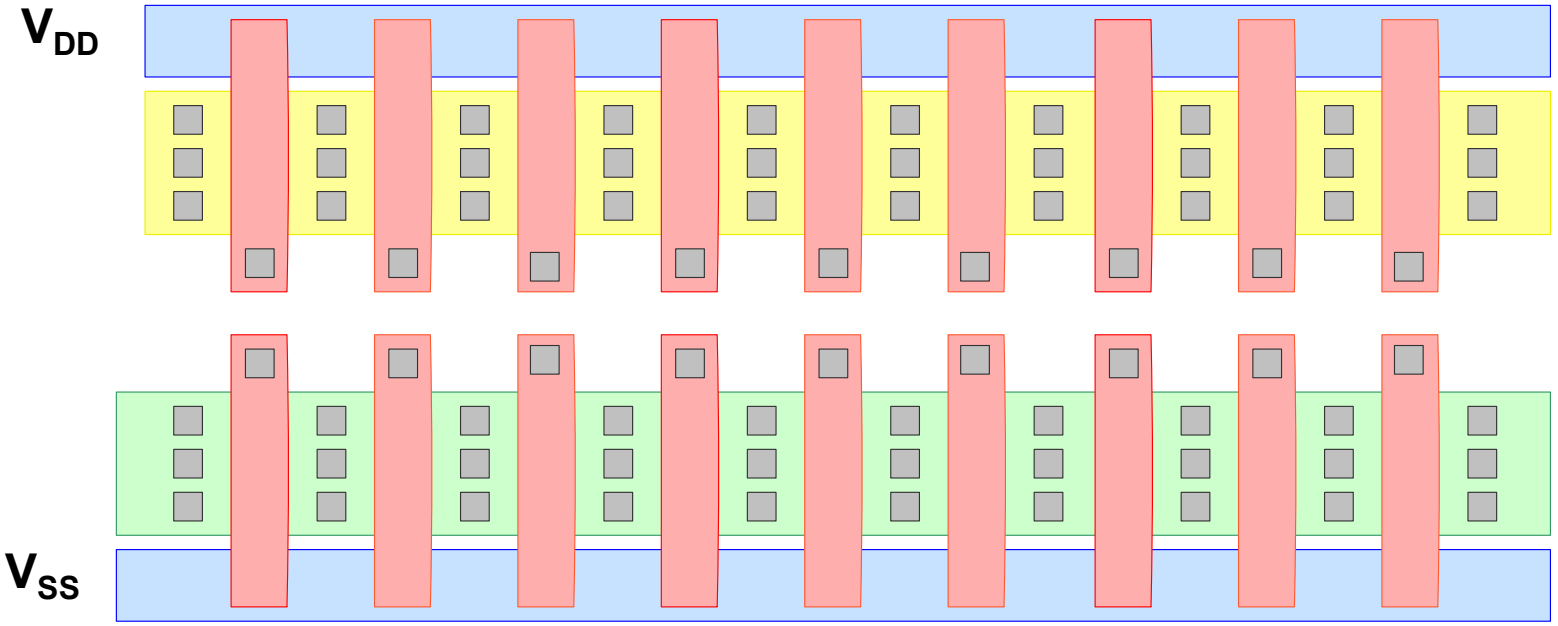


Via (M1 to M2)

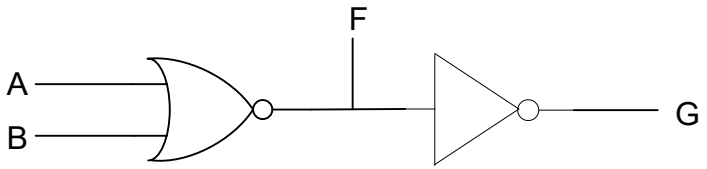
Contact (M1 to diff,Poly)

Array Logic

Sea of Gates



Example:

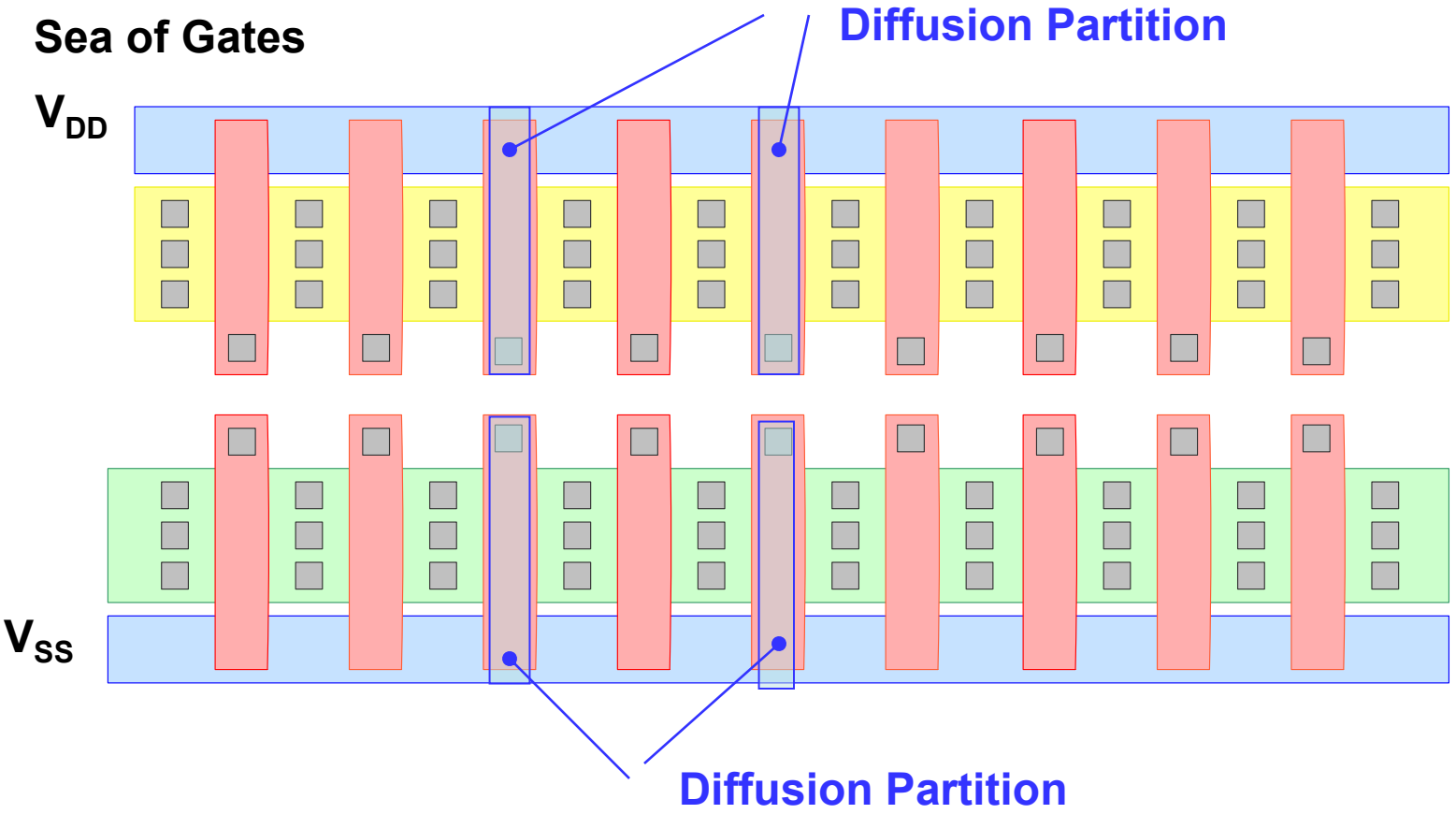


Via (M1 to M2)

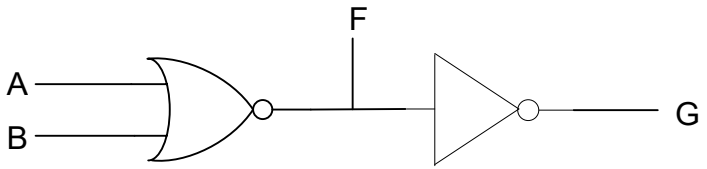
Contact (M1 to diff,Poly)

Array Logic

Sea of Gates



Example:

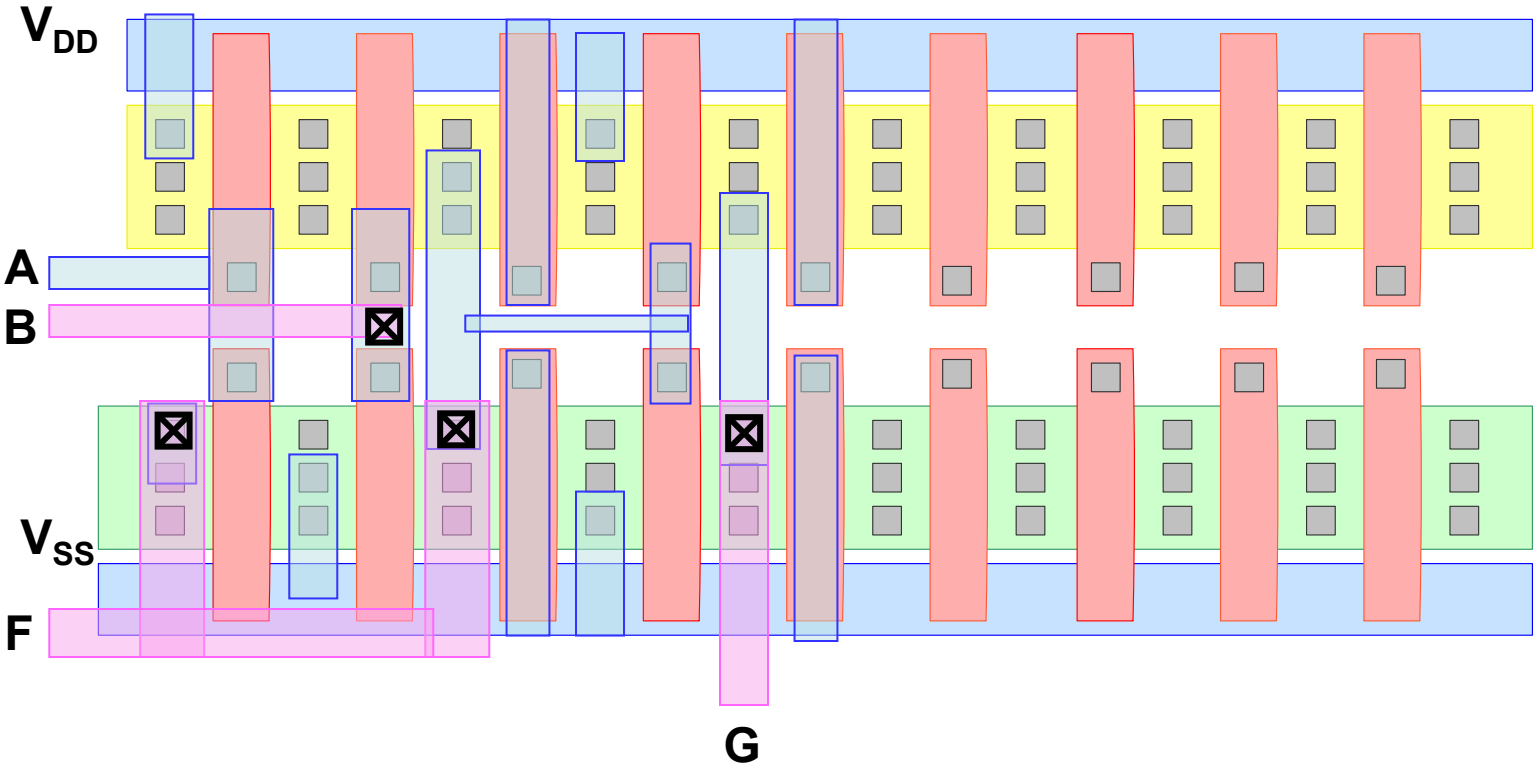


Via (M1 to M2)

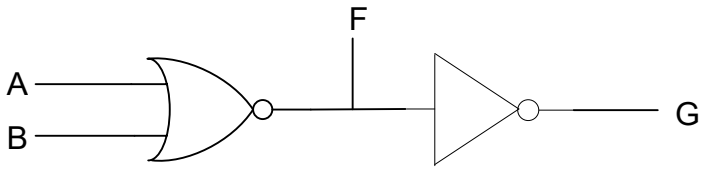
Contact (M1 to diff,Poly)

Array Logic

Sea of Gates



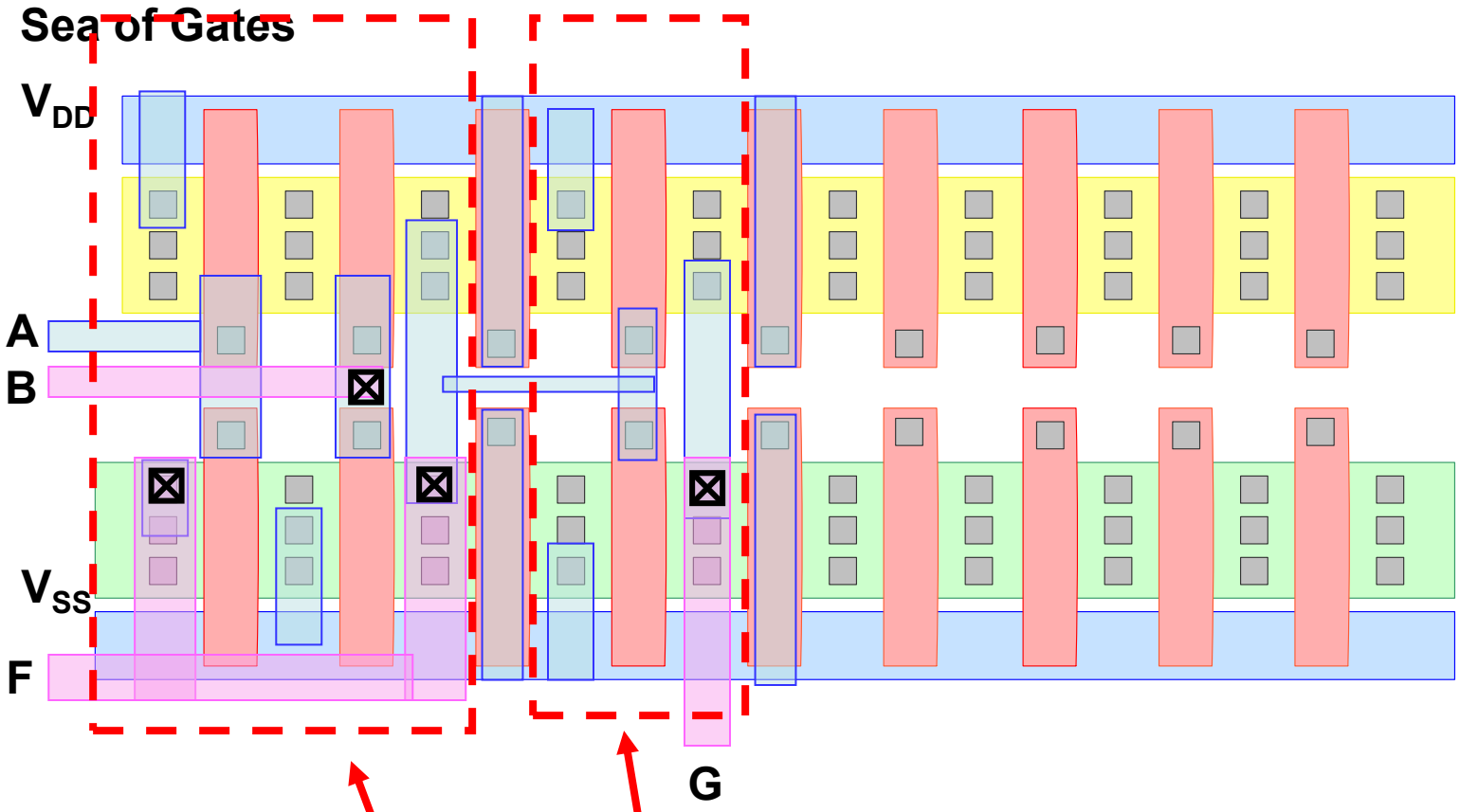
Example:



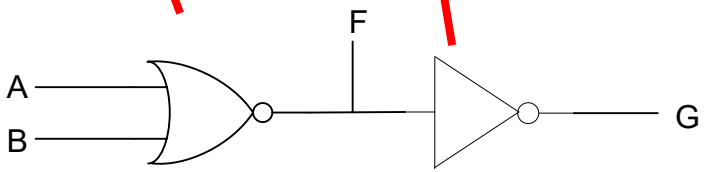
Via (M1 to M2)

Contact (M1 to diff,Poly)

Array Logic



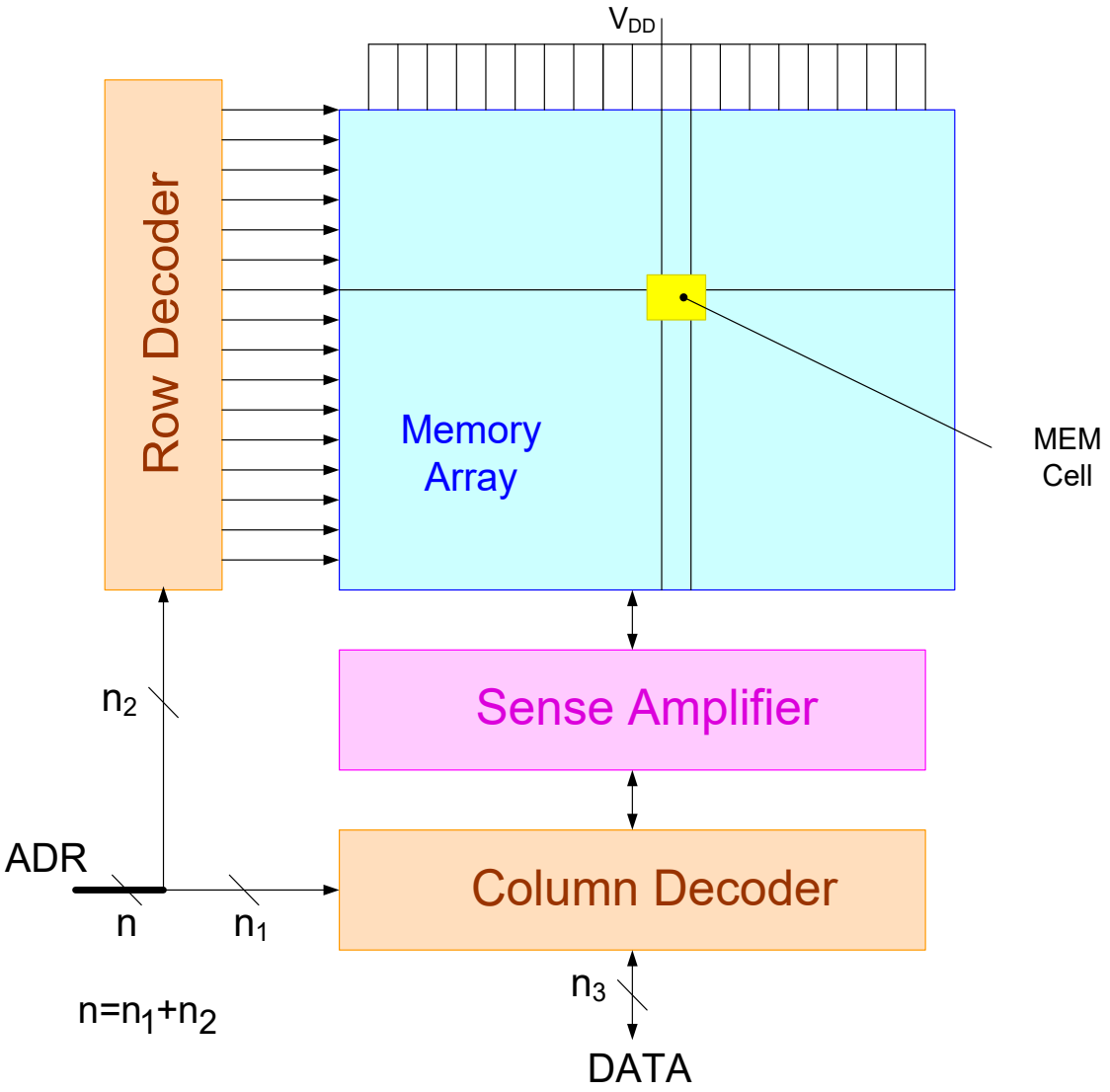
Example:



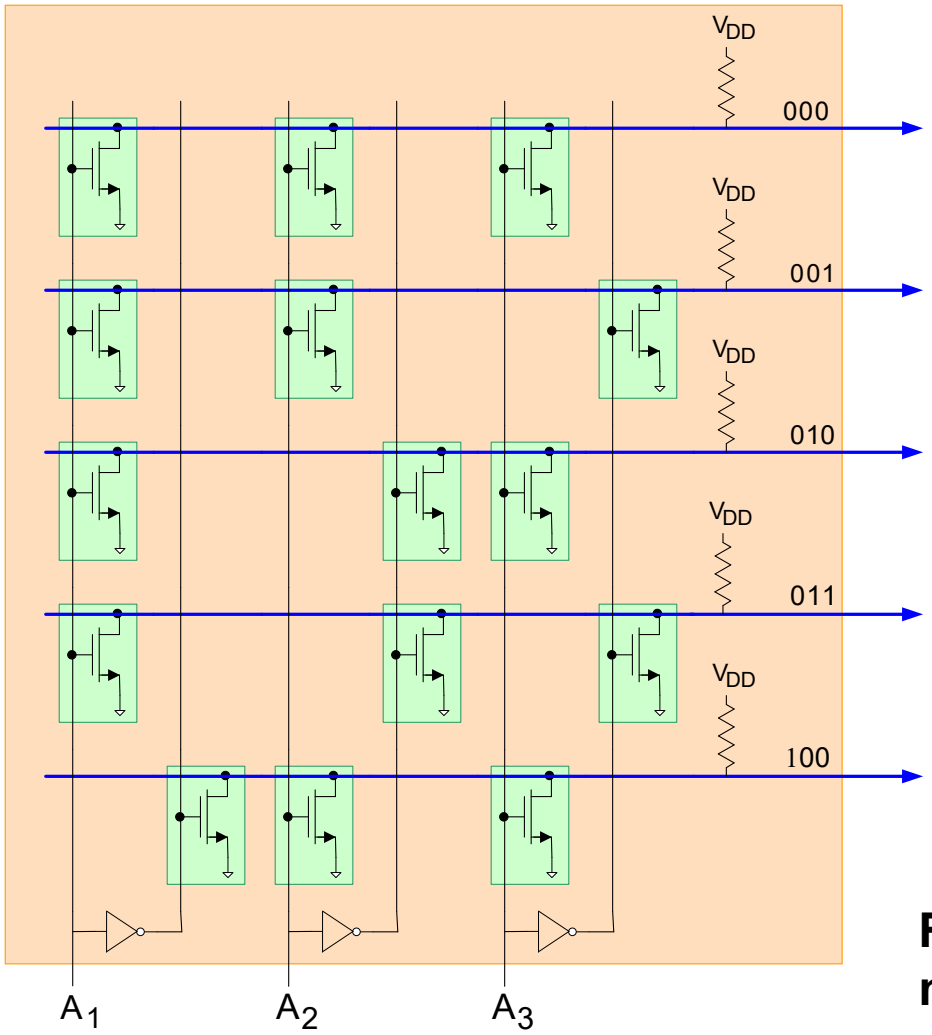
Via (M1 to M2)

Contact (M1 to diff,Poly)

Typical Memory Structure



Row Decoder Architectures



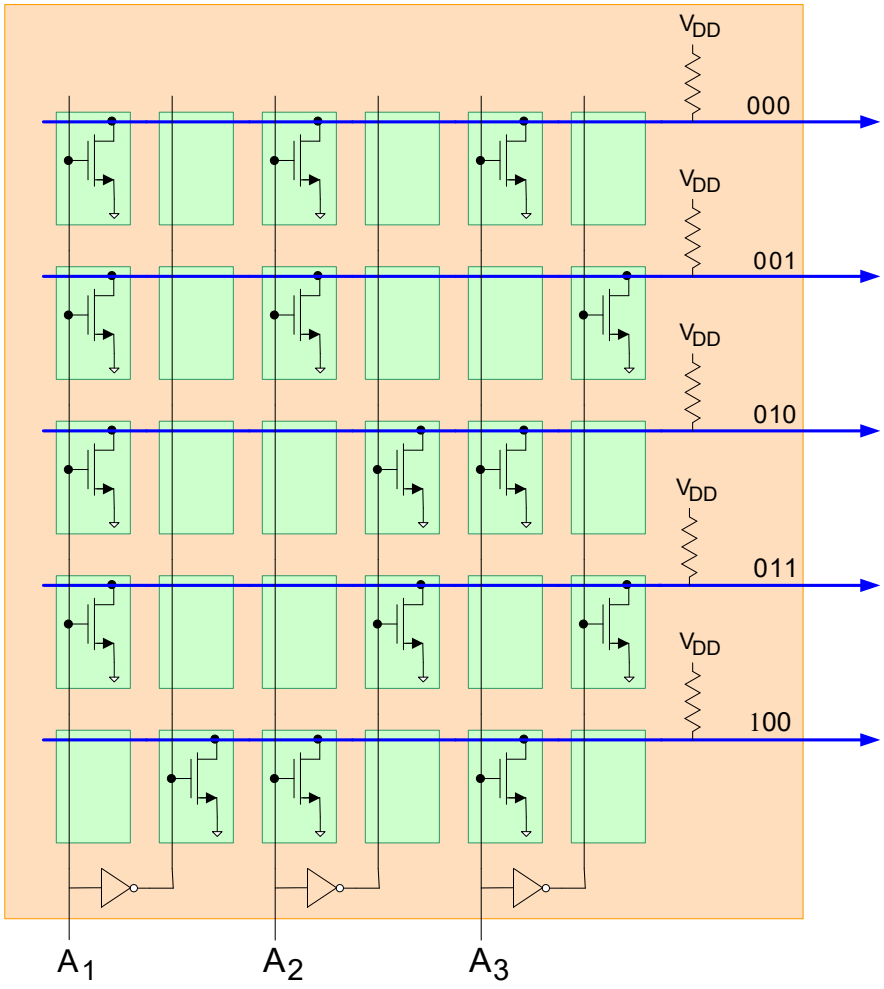
$$R_k = \bar{A}_1 \bullet \bar{A}_2 \bullet \bar{A}_3$$

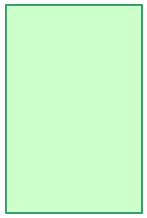
$$R_k = \overline{(A_1 + A_2 + A_3)}$$

Row decoder is Pseudo n-MOS NOR Gate

Typically n/2 inputs where n is th address length

Row Decoder Architectures

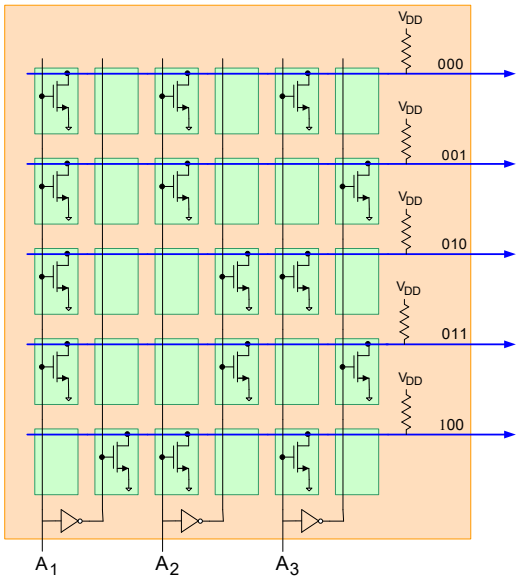


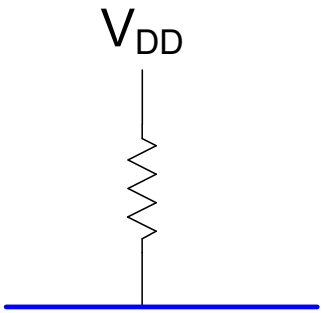


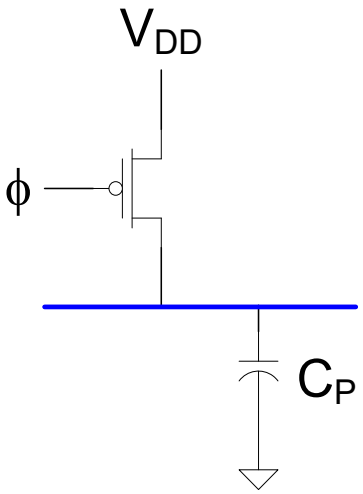
Transistor sites typically reserved in the layout for efficient, compact layo

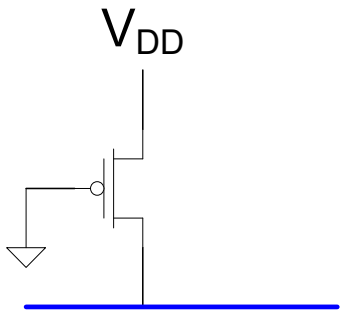
Row Decoder Architectures

Pull-up resistor implemented with either weak p or with dynamic precharge by taking clock ϕ low to precharge to high (thus dynamic NOR gate)

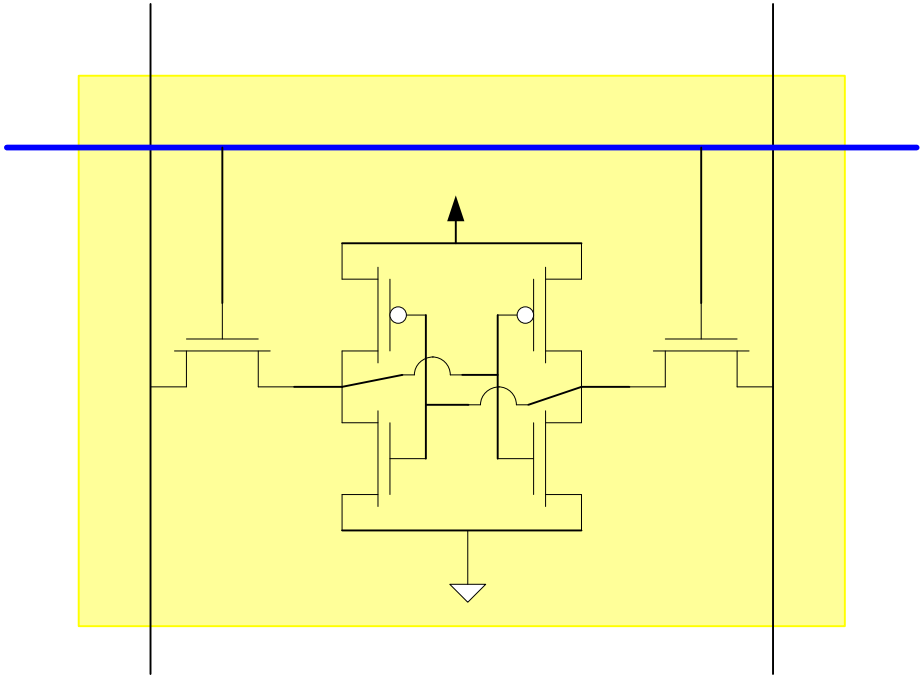








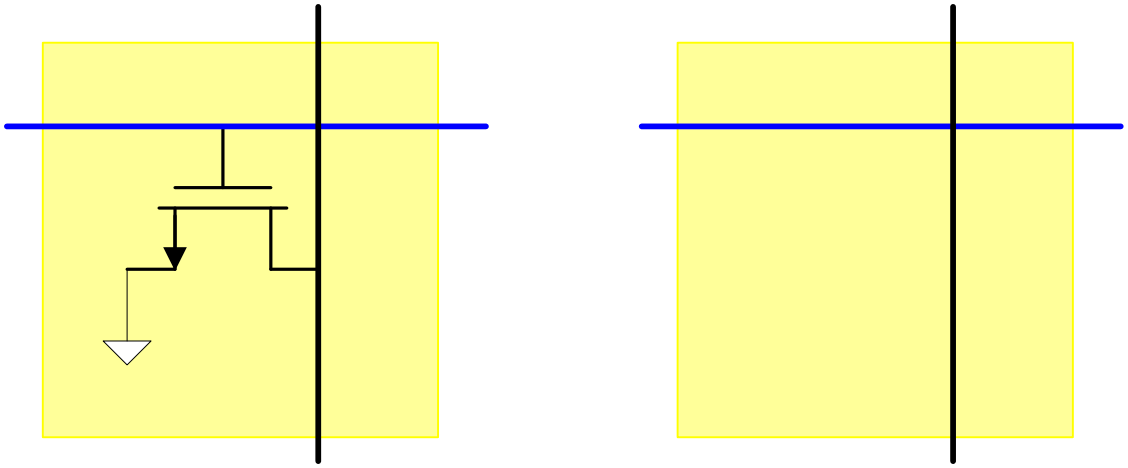
Mem Cells



Static RAM (SRAM)

- **Uses PTL and cross-coupled inverters**
- **Sizing of “switches” must be strong enough to write to cell**
- **No static power dissipation in this PTL implementation**

Mem Cells



Transistor Present

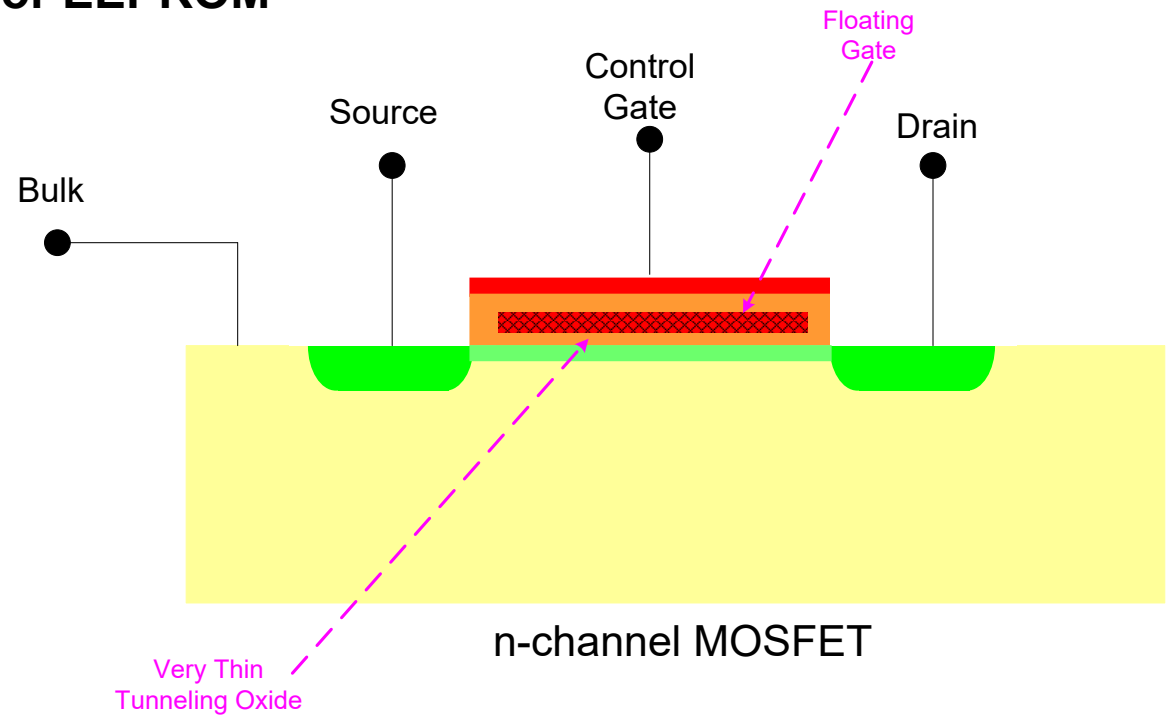
Transistor Absent

Static ROM (Mask programmable ROM)

- **Site reserved for possible transistor**
- **Actually programmed with contact to gate and diffusion**
- **Can personalize with one or two masks**
- **Single transistor per bit**
- **Uses only one column line**

Mem Cells

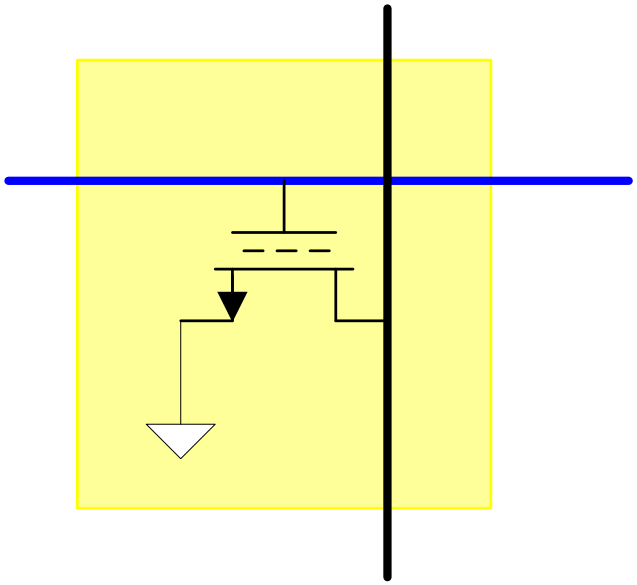
EPROM or EEPROM



Floating Gate Transistor

- **Very thin floating gate**
- **Charge tunnels onto gate during programming to change V_T a lot**
- **Conceptual diagram only**
- **Somewhat specialized processing for reliable floating gate devices**

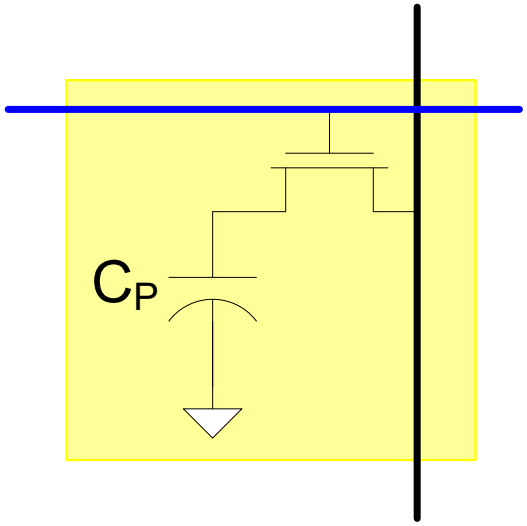
Mem Cells



EPROM or EEPROM

- **Floating Gate Transistor**
- **Programmed by Changing the Threshold Voltage**
- **Nonvolatile Memory**
- **Can be electrically programmed with EEPROM**
- **Limited number of read/write cycles (but enough for most applications)**
- **Uses only one column line**

Mem Cells



DRAM

- **Charge stored in small parasitic capacitor**
- **Very small cells**
- **Volatile and dynamic**
- **Special processes to make C_P large in very small area**
- **C_P is actually a part of the transistor**
- **Somewhat tedious architecture (details not shown) needed to sense very small charge**

Stay Safe and Stay Healthy !

End of Lecture 44