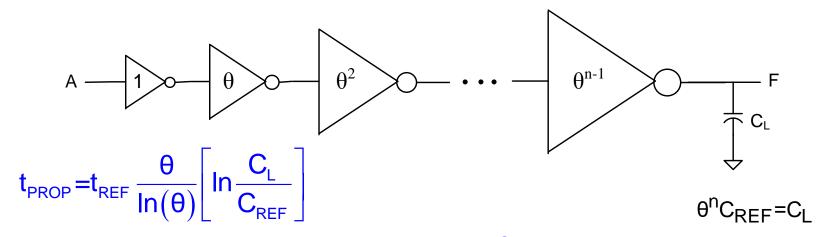
# EE 330 Lecture 44

### **Digital Circuits**

- Power Dissipation
- Dynamic Logic Circuits
- Higher-level digital blocks

### Optimal Driving of Capacitive Loads



Is suffices to minimize the function

ffices to minimize the function 
$$\frac{df}{d\theta} = \frac{\ln(\theta) - \theta \cdot \left(\frac{1}{\theta}\right)}{\left(\ln(\theta)\right)^2} = 0$$

$$ln(\theta) - 1 = 0 \rightarrow \theta = e$$

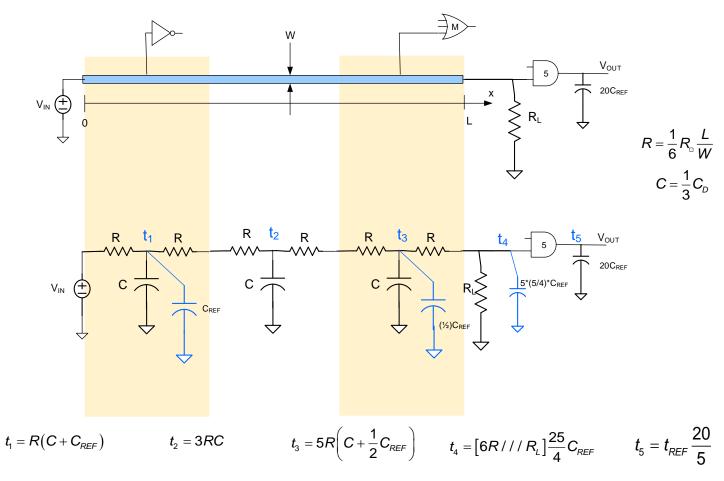
$$n = \frac{1}{\ln(\theta)} \ln \left( \frac{C_L}{C_{REF}} \right) \rightarrow n = \ln \left( \frac{C_L}{C_{REF}} \right)$$

### Propagation Delay in "Logic Effort" approach

$$t_{PROP} = t_{REF} \sum_{k=1}^{n} f_k = t_{REF} \sum_{k=1}^{n} g_k h_k = t_{REF} \sum_{k=1}^{n} \frac{F_{I(k+1)}}{OD_k}$$

- Note this expression is identical to what we have derived previously (t<sub>REF</sub> scaling factor not included in W\_H text)
- Probably more tedious to use the "Logical Effort" approach
- Extensions to asymmetric overdrive factors may not be trivial
- Extensions to include parasitics may be tedious as well
- Logical Effort is widely used throughout the industry

# Elmore Delay Calculations



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

## 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

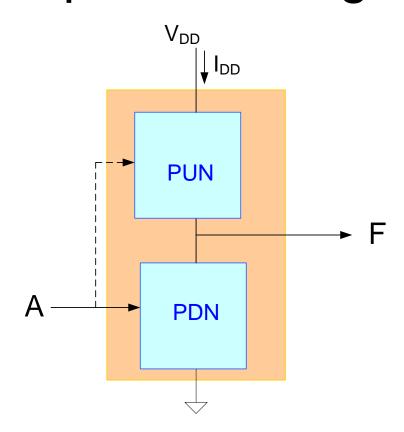
Other Logic Styles

Array Logic

Ring Oscillators

done nartial

### Power Dissipation in Logic Circuits



Assume current periodic with period  $T_{CL}$ 

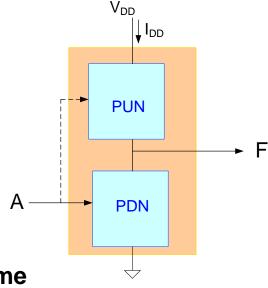
$$P_{AVG,T} = \frac{1}{T_{CL}} \int_{t_1}^{t_1 + T_{CL}} V_{DD} I_{DD}(t) dt$$

# Power Dissipation in Logic Circuits

### **Types of Power Dissipation**

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

### Static Power Dissipation



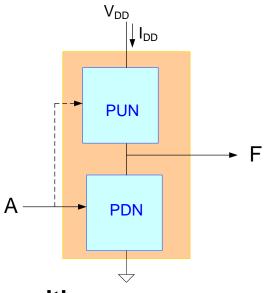
If Boolean output averages H and L 50% of the time

$$P_{STAT,AVG} = \frac{P_{H} + P_{L}}{2}$$

$$P_{STAT,AVG} = \frac{V_{DD}(I_{DDH} + I_{DDL})}{2}$$

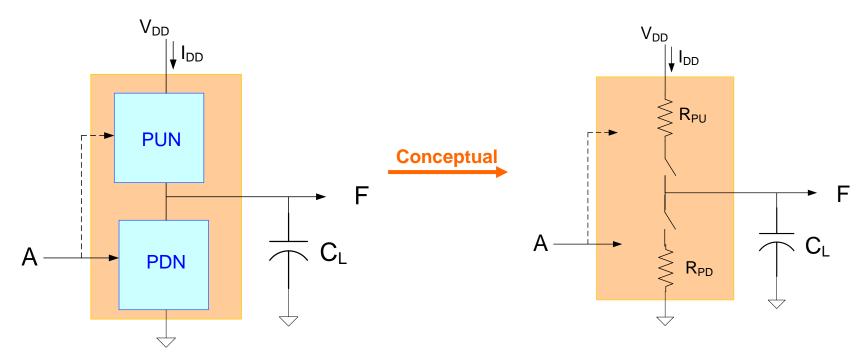
- Generally decreases with V<sub>DD</sub>
- I<sub>DDH</sub>=I<sub>DDL</sub>=0 for static CMOS gates so P<sub>STAT</sub>=0
- A major source of power dissipation in ratio logic circuits and the major reason CMOS is so widely used

### Pipe Power Dissipation



Due to conduction of both PUN and PDN during transitions

- Can be made small if transitions are fast
- Usually negligible in Static CMOS circuits



Due to charging and discharging C<sub>L</sub> on logic transitions

 $C_L$  dissipates no power but PUN and PDN dissipate power during charge and discharge of  $C_L$ 

C<sub>L</sub> includes all gate input capacitances of loads and interconnect capacita

#### Energy supplied by V<sub>DD</sub> when C<sub>L</sub> charges

$$E = \int_{1}^{\infty} V_{DD}I_{DD}(t)dt$$

$$t_{1}$$

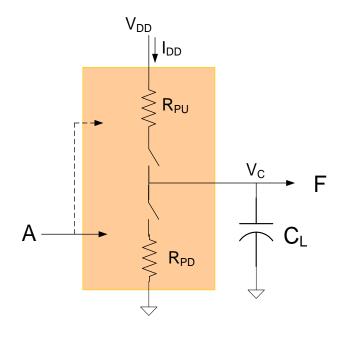
$$I_{DD} = C_{L} \frac{dV_{C}}{dt}$$

$$E = \int_{t_1}^{\infty} V_{DD} C_L \frac{dV_C}{dt} dt$$

$$E = \int_{V_C=0}^{V_{DD}} V_{DD} C_L dV_C = V_{DD} C_L \int_{V_C=0}^{V_{DD}} dV_C = V_{DD} C_L V_C \Big|_{V_C=0}^{V_{DD}} = V_{DD}^2 C_L$$

Energy stored in  $C_L$  after  $C_L$  is charged to  $V_{DD}$ :

$$E = \frac{1}{2}C_L V_{DD}^2$$



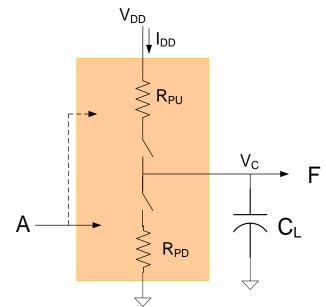


Energy supplied by  $V_{DD}$  and dissipated in  $R_{PU}$  when  $C_L$  charges

$$E_{DIS} = \frac{1}{2}C_L V_{DD}^2$$

Energy stored on C<sub>L</sub> after L-H transition

$$E_{STORE} = \frac{1}{2}C_L V_{DD}^2$$



Thus, energy from V<sub>DD</sub> for one L-H: H-L output transition sequence is

$$E = E_{DIS} + E_{STORE} = C_L V_{DD}^2$$

When the output transitions from H to L, energy stored on C<sub>L</sub> is dissipated in PDN

If f is the average transition rate of the output, determine P<sub>AVG</sub>



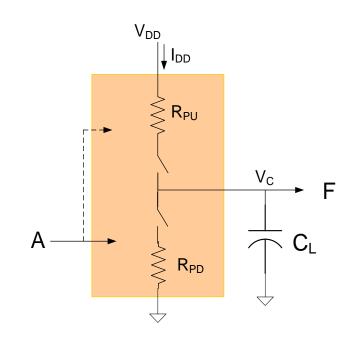
Energy from V<sub>DD</sub> for one L-H: H-L output transition sequence is

$$E=C_LV_{DD}^2$$

If f is the average transition rate of the output, determine  $P_{AVG}$ 

$$P_{AVG} = \frac{E}{T} = Ef$$

$$P_{DYN} = fC_L V_{DD}^2$$



If a gate has a transition duty cycle of 50% with a clock frequency of f<sub>CL</sub>

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

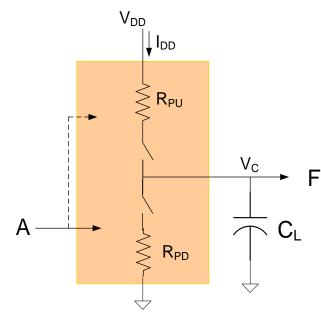
Note dependent on the square of  $V_{DD}$ ! .... Want to make VDD small !!!

Major source of power dissipation in many static CMOS circuits for L<sub>min</sub>>0.1u



**Energy dissipated with clock signal itself** 

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



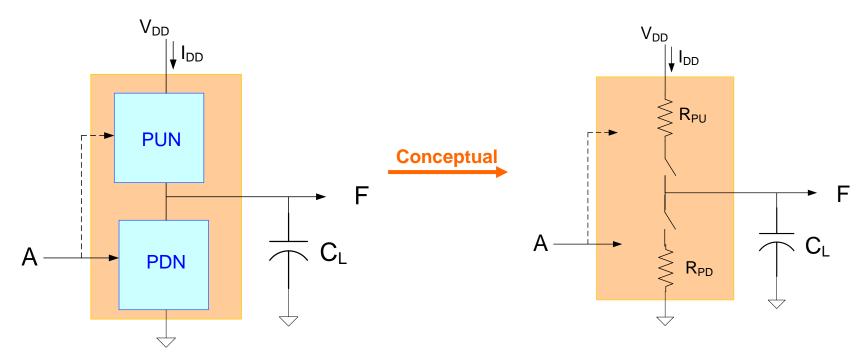
The clock transitions on every clock cycle (i.e. it has a transition duty cycle of 100%)

Clock distribution can cause significant power dissipation

But if a gate has a transition duty cycle of 50% with a clock frequency of fcl

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

### **Power Dissipation**



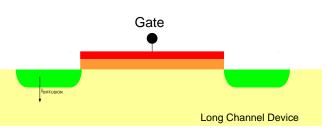
- All power is dissipated in pull-up and pull-down devices
- C<sub>L</sub> dissipates no power but PUN and PDN dissipate power when charging and discharging C<sub>L</sub>
- Dynamic power dissipation reduced by more (often much more) than a factor of 2 if minimum sizing strategy is used

# 'n

### 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

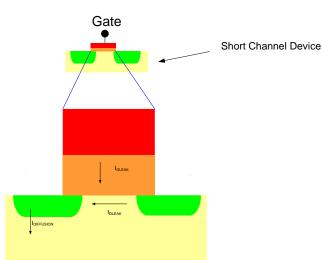


#### -Diffusion

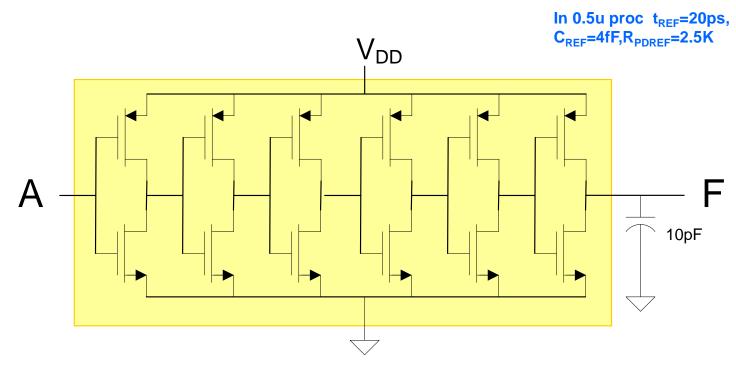
- Leakage across a reverse-biased pn junction
- Dependent upon total diffusion area
- May actually be dominant power loss on longerchannel devices
- Actually a type of static power dissipation

#### -Drain

- channel current due to small V<sub>GS</sub>-V<sub>T</sub>
- of significant concern only with low V<sub>DD</sub> 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 clocked at 500MHz. Assume pad driver with OD of  $\theta$ =2.5 and  $V_{DD}$ =3.5V

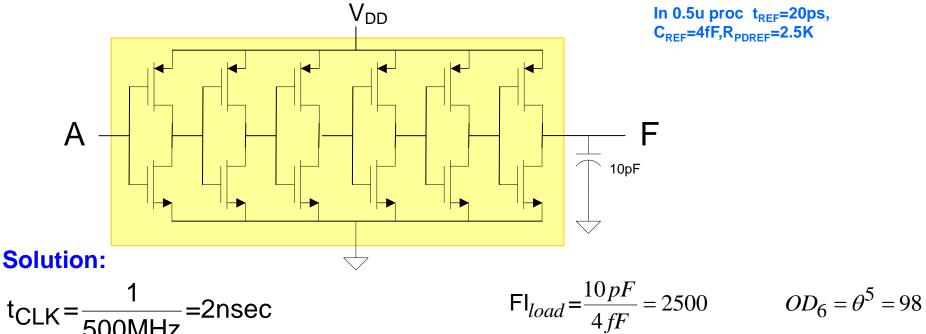


Solution: (assume output changes with 50% of clock transitions)

$$P_{DYN} = \frac{f_{CL}}{2} C_L V_{DD}^2 = 5E8 \cdot 10pF \cdot 3.5^2 = 61mW$$

Note this solution is independent of the OD and the process

#### Example: Will the CMOS pad driver actually be able to drive the 10pF load at 500MHz in the previous example in the 0.5u process?



$$t_{CLK} = \frac{1}{500MHz} = 2nsec$$

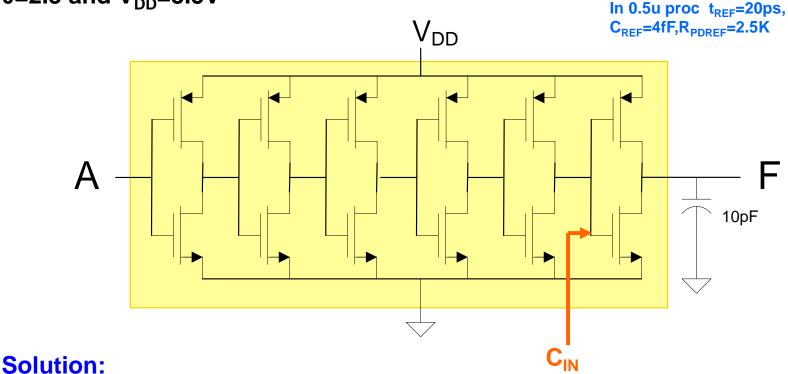
$$t_{PROP} = 50 \bullet t_{REF} + \frac{Fl_{load}}{OD_6} t_{REF} = \frac{Fl_{load}}{OD_6} = \frac{2500}{98} \approx 25$$

$$\frac{\mathsf{FI}_{load}}{OD_6} = \frac{2500}{98} \cong 25$$

$$t_{prop} = 5 \cdot 2.5 \cdot 20psec + 25 \cdot 20psec = (12.5 + 25)20psec = 0.75nsec$$

since t<sub>CLK</sub>>t<sub>PROP</sub>, this pad driver can drive the 10pF load at 500MHz

Example: Determine the dynamic power dissipation in the <u>next to the last stage</u> of a 6-stage CMOS pad driver if used to drive a 10pF capacitive load if clocked at 500MHz. Assume pad driver with OD of  $\theta$ =2.5 and  $V_{DD}$ =3.5V

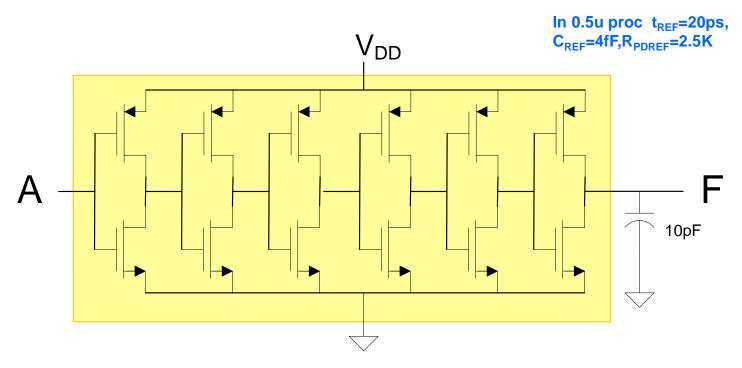


F

$$C_{IN} = \theta^5 C_{REF} = 2.5^5 \cdot 4fF = 390fF$$

$$P_{DYN} = f_{CL}C_LV_{DD}^2 = 5E8 \cdot 390 fF \cdot 3.5^2 = 2.4 mW$$

Example: Is the 6-stage CMOS pad driver adequate to drive the 10pF capacitive load as fast as possible? Assume pad driver with OD of  $\theta$ =2.5 and  $V_{DD}$ =3.5V



#### **Solution:**

$$n_{OPT} = ln \left( \frac{C_L}{C_{RFF}} \right) = ln \left( \frac{10pF}{4fF} \right) = 7.8$$

No – an 8-stage pad driver would drive the load much faster (but is not needed If clocked at only 500MHz)

Example: Determine the power that would be required in the last stage of a CMOS pad driver to drive a 32-bit data bus off-chip if the capacitive load on each line is 2pF. Assume the clock speed is 500MHz and that each bit has an average 50% toggle rate. Assume  $V_{DD}$ =3.5V

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

#### **Solution:**

$$P_{DYN} = 32 \bullet \frac{f_{CL}}{2} C_L V_{DD}^2 = 32 \bullet \frac{5E8}{2} \bullet 2pF \bullet 3.5^2 = 196mW$$

Note: A very large amount of power is required to take a large bus off-chip if bus has a high rate of activity.

## 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

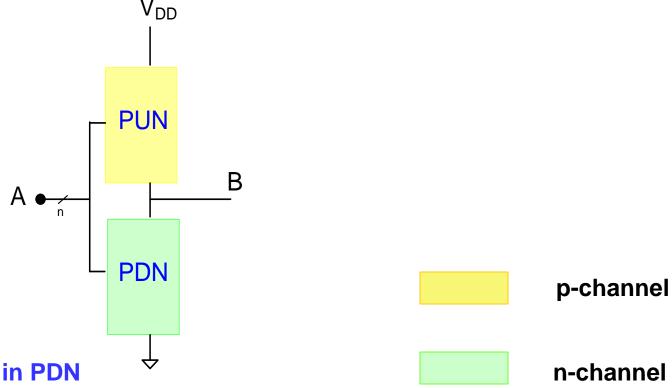
- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
- Other Logic Styles
  - Array Logic
  - Ring Oscillators

done partia

# Logic Styles

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

## Complex Logic Gates

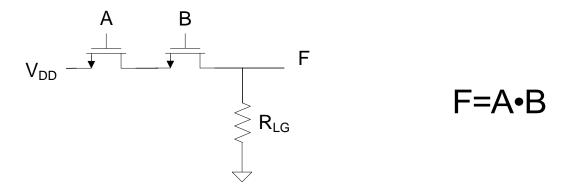


- Implement B in PDN
- Implement B in PUN with complimented input variables
- Zero static power dissipation
- $V_H = V_{DD}$ ,  $V_L = 0V$  (or  $V_{SS}$ )
- Complimented input variables often required

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



## **Pass Transistor Logic**



#### **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<sub>LG</sub> may be unacceptably large
- Slow t<sub>LH</sub>
- Signal degradation <u>can</u> occur when multiple levels of logic are used

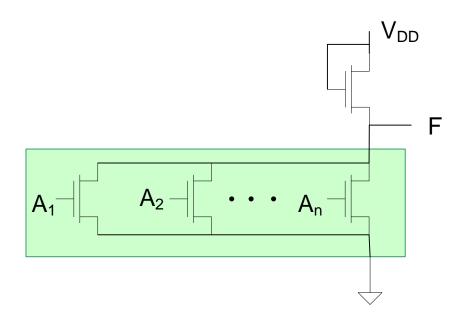


- Widely used in some applications
- Implements basic logic function only once!
- Fan In can be very small so low dynamic power dissipation!



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

# Pseudo NMOS Logic



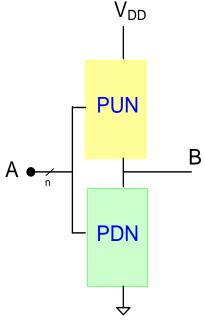
n could be several hundred or even several thousand



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** 



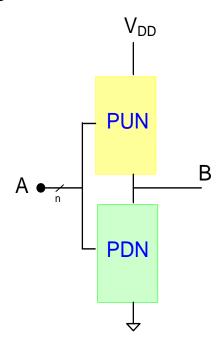
What is the biggest contributor to area? PUN (3X active area for inverter, more for NOR gates, and Well)

What is biggest contributor to dynamic power dissipation?

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

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



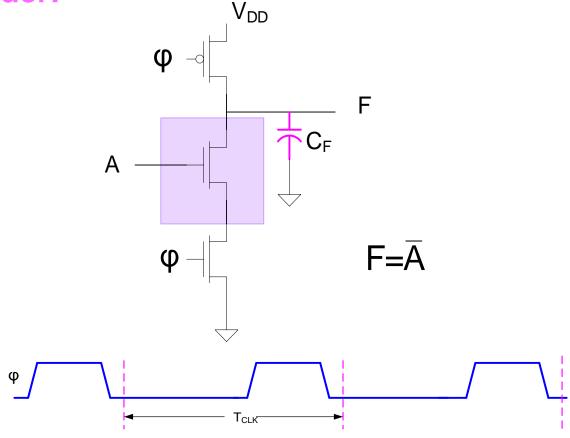


Can the PUN be eliminated W/O compromising signal levels and power dissipation?

Benefits could be most significant!



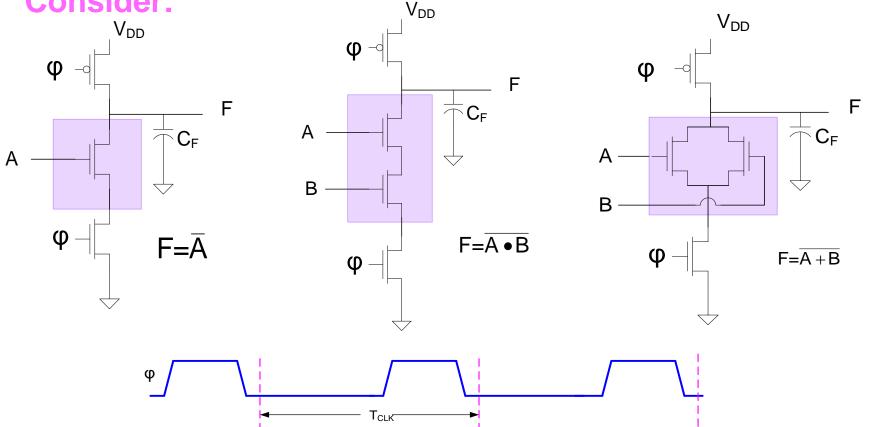
#### **Consider:**



Precharges  $C_F$  to "1" when  $\phi$  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)



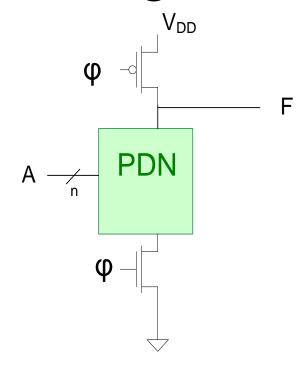
### **Consider:**



- Termed Dynamic Logic Gates
- Parasitic capacitors actually replace C<sub>F</sub>
- If Logic Block is n-channel, will have rail to rail swings
- Logic Block is simply a PDN that implements F



**Basic Dynamic Logic Gate** 



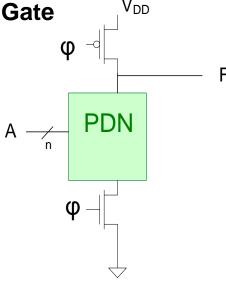
#### Any of the PDNs used in complex logic gates would work here!

- Have eliminate the PUN!
- Ideally will have a factor of 4 or more reduction in C<sub>IN</sub>
- 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!

### From Wikipedia: Dec 7 2016

In <u>integrated circuit</u> design, **dynamic logic** (or sometimes **clocked logic**) is a design methodology in combinatory logic circuits, particularly those implemented in MOS technology. It is distinguished from the so-called static logic by exploiting temporary storage of information in stray and gate capacitances.[1] It was popular in the 1970s and has seen a recent resurgence in the design of high speed digital <u>electronics</u>, particularly <u>computerCPUs</u>. Dynamic logic circuits are usually faster than static counterparts, and require less surface area, but are more difficult to design. Dynamic logic has a higher toggle rate than static logic but the <u>capacitative loads</u> being toggled are smaller so the overall power consumption of dynamic logic may be higher or lower depending on various tradeoffs. When referring to a particular logic family, the dynamic adjective usually suffices to distinguish the design methodology, e.g. dynamic <u>CMOS<sup>[4]</sup></u> or dynamic <u>SOI</u> design.<sup>[2]</sup> Dynamic logic is distinguished from so-called static logic in that dynamic logic uses a <u>clock signal</u> in its implementation of <u>combinational logic</u> circuits. The usual use of a clock signal is to synchronize transitions in sequential logic circuits. For most implementations of combinational logic, a clock signal is not even needed.

**Basic Dynamic Logic Gate** 



#### Advantages:

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

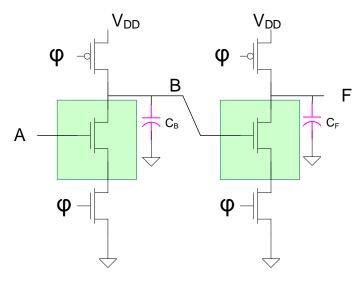
#### **Limitations:**

- 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



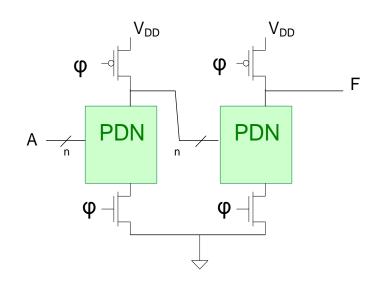


**Premature Discharge Problem** 

B will be pulled high during the pre-charge state and try to discharge C<sub>F</sub> 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!



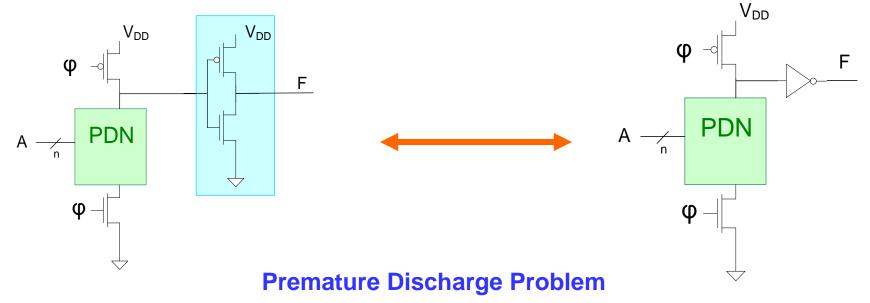
**Premature Discharge Problem** 

This problem occurs when any inputs to an arbitrary dynamic logic gate create an R<sub>PD</sub> path in the PDN during at the start of the evaluate phase that is not to pull do

How can this problem be fixed? wn later in that evaluate phase 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)





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

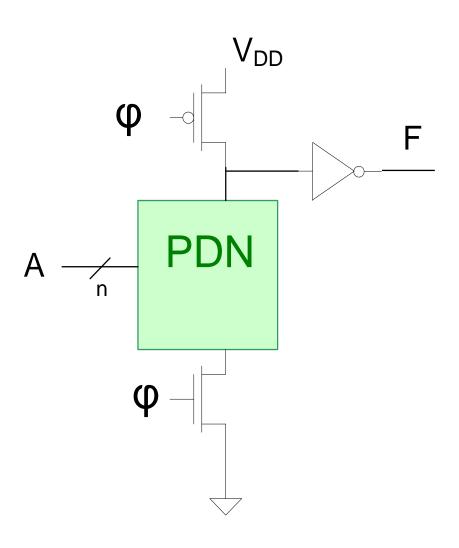
Implement F instead of  $\overline{F}$  in the PDN

**Termed Domino Logic** 

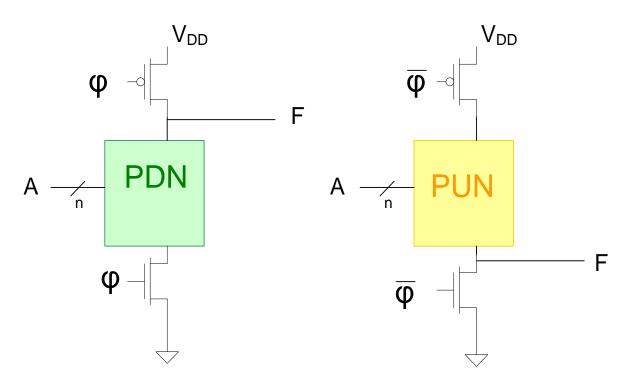
Some additional dynamic power dissipation in the inverter

Some additional delay during the evaluate state in inverter

# Domino Logic



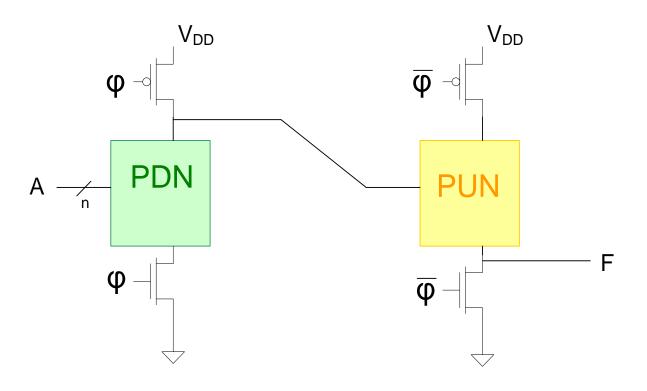
### Dynamic Logic



- 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

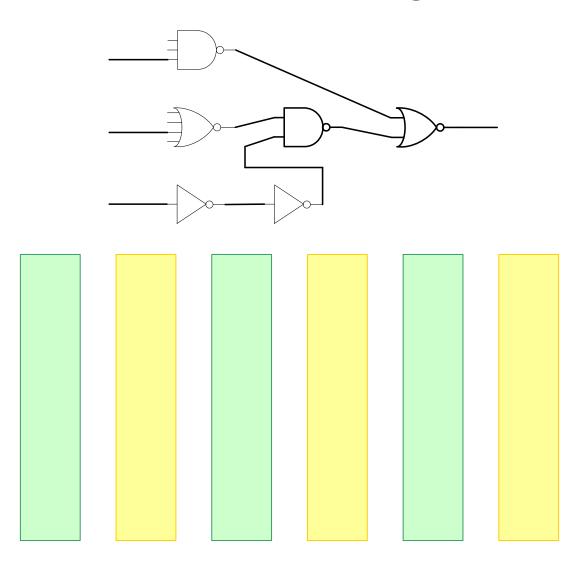
## Dynamic Logic





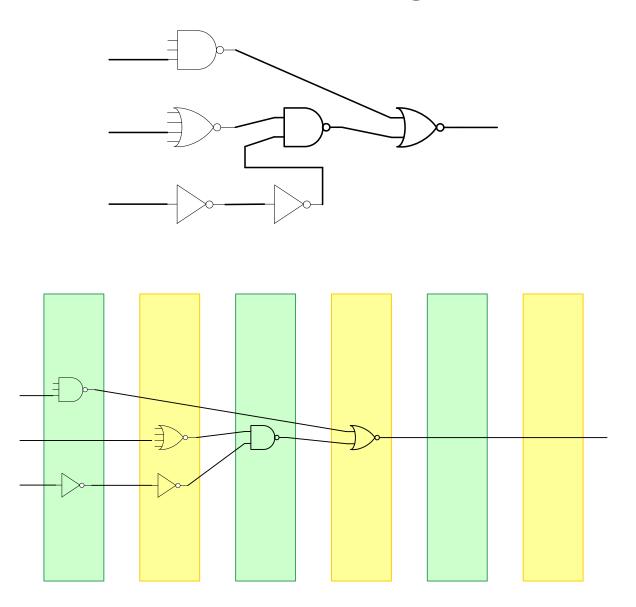
Direct coupling between alternate type dynamic gates

# Zipper Logic



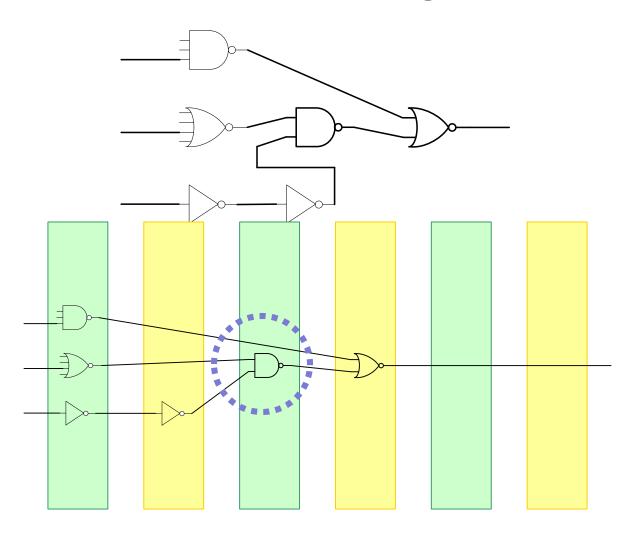
Map gates to appropriate precharge type

# Zipper Logic



**Acceptable Implementation in Zipper** 

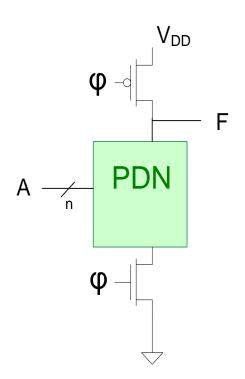
## Zipper Logic

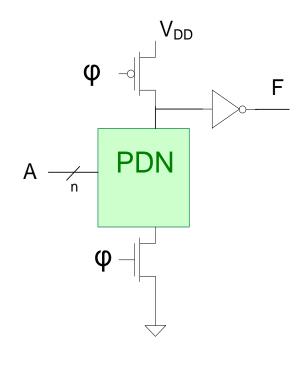


**Unacceptable Implementation in Zipper** 

- Premature discharge at output of 2-input NAND

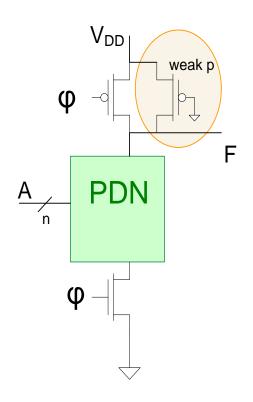
#### Static Hold Option

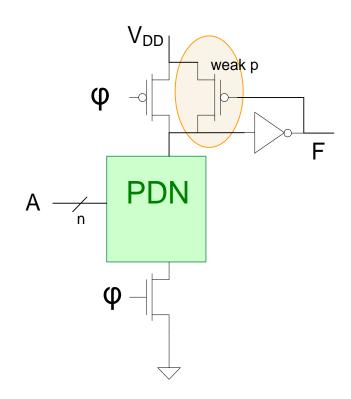




If not clocked, charge on upper node of PDN will drain off causing H output to degrade

#### Static Hold Option



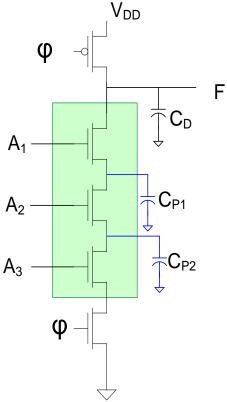


- weak p will hold charge
- size may be big (long L)
- some static power dissipation
- can use small current source

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

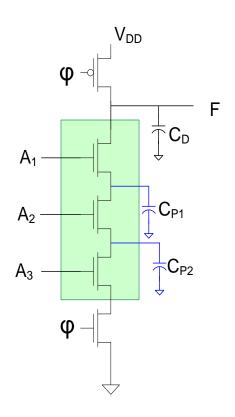
- comptimes towned ((keeps

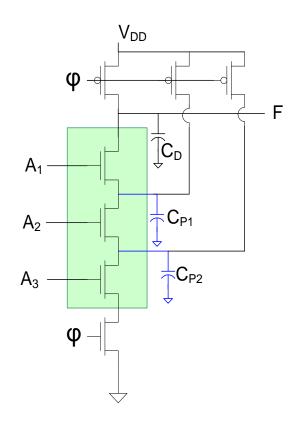
#### Charge stored on internal nodes of PDN



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

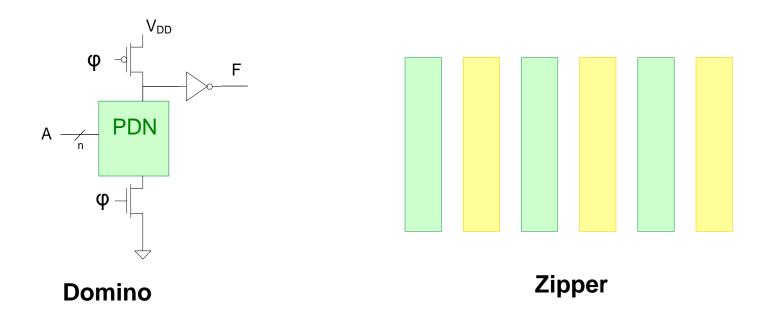
- Domino
- Zipper
- Ratio-less 2-phase
- Ratio-less 4-phase
- Output Prediction

#### Logic

Fully differential

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

## Future of Dynamic Logic



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

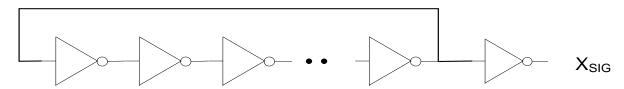
**Transistor-transistor logic** 

### 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

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

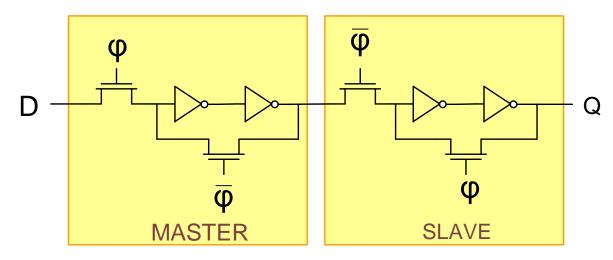
- n is the number of stages
- t<sub>PROP</sub> is the propagation delay of a single stage (all assumed identical)

### Sequential Logic Circuits

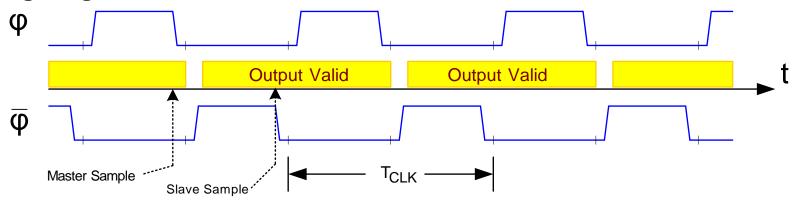
- Flip Flops needed for sequential logic circuit
- Only one type of flip flop is required
- Invariably require clocked edge-triggered master-slave flop flops
- Flip flop circuits can be very simple
- Flip flops are part of Standard Cell Libraries

### Flip Flops

#### Master-Slave Edge-triggered D Flip Flop



#### **Timing Diagram**

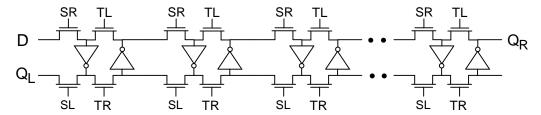


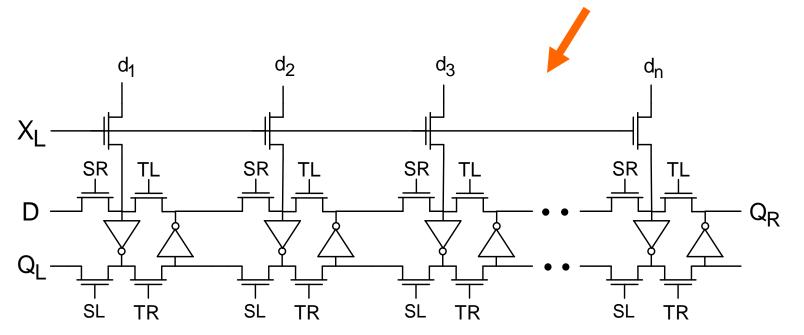
- 12 transistors (but will work with 10)
- Many other simple D Flip-flops exist as well

### 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 H & TL and H & TL

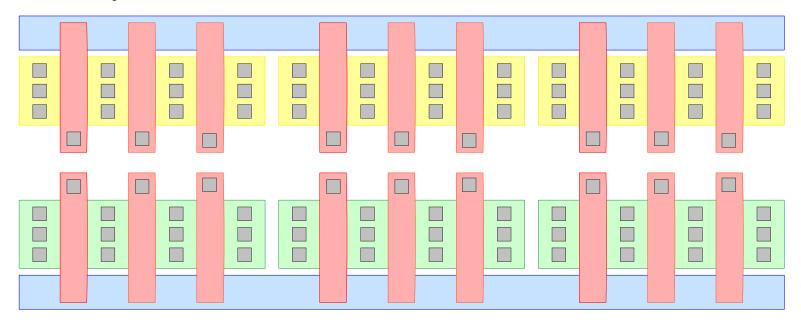
- Array logic is often used for sections of logic that may change later in the design or that will be changed for different variants of a product
- FPGA are a special case of array logic
- Can personalize array logic with only one layer of metal
  - Very quick turn-around and low incremental costs (as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

#### Will consider only two types

- Gate Array
- Sea of Gates

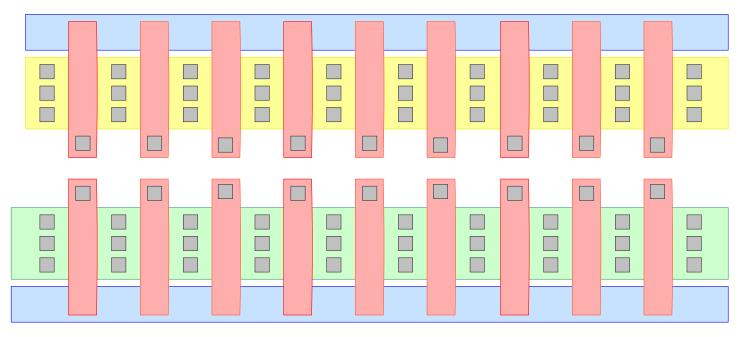
Variants of the following approach are possible depending upon process but this will convey the basic concepts

#### **Gate Array**



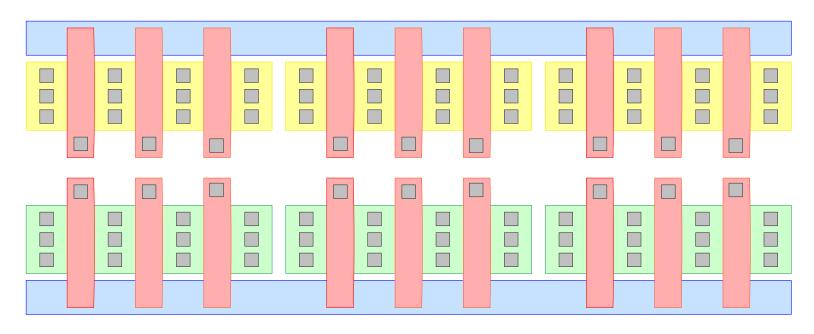
- Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2)
   (3 simple masks)
- Upper and lower metal shown actually lie above poly and are automatically present
- Assume upper M1 is V<sub>DD</sub> and lower M1 is V<sub>SS</sub>
- Array can be very large
- Routing channels between segments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

#### **Sea of Gates**

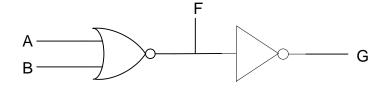


- Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2)
   (3 simple masks)
- Upper and lower metal shown actually lie above poly and are automatically present
- Assume upper M1 is V<sub>DD</sub> and lower M1 is V<sub>SS</sub>
- Array can be very large
- Routing channels between segments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

#### **Gate Array**

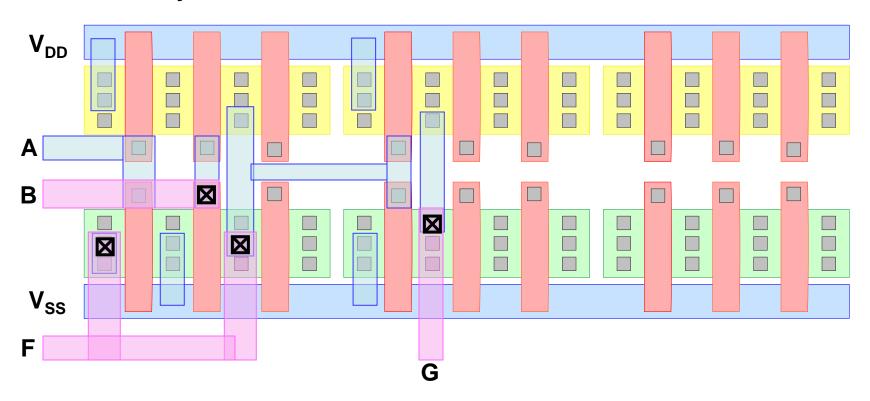


#### **Example:**

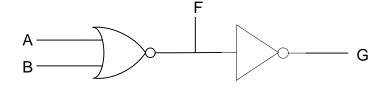


- ☑ Via (M1 to M2)
- Contact (M1 to diff,Poly)

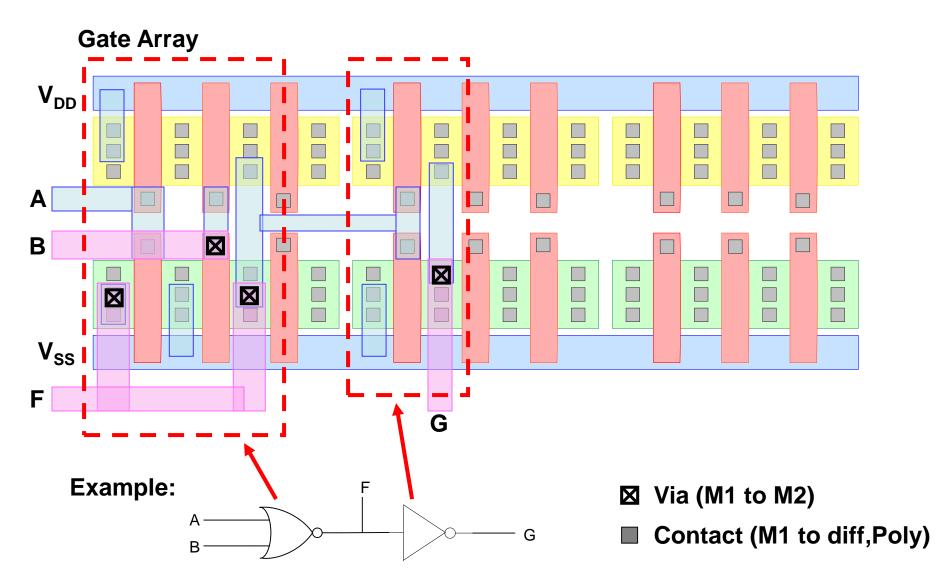
#### **Gate Array**



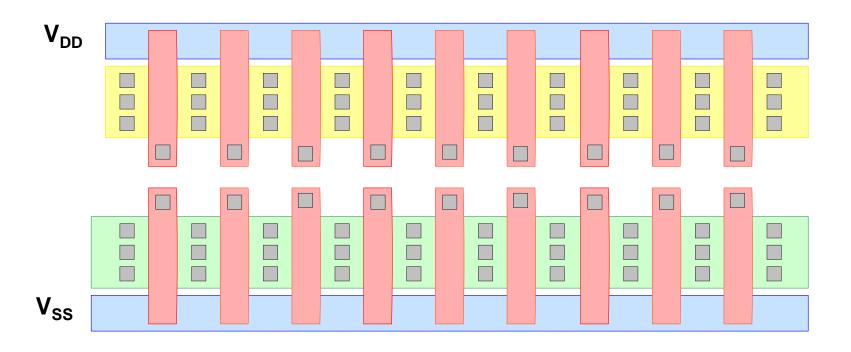
#### **Example:**

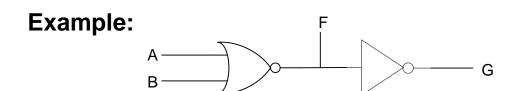


- ☑ Via (M1 to M2)
- Contact (M1 to diff,Poly)

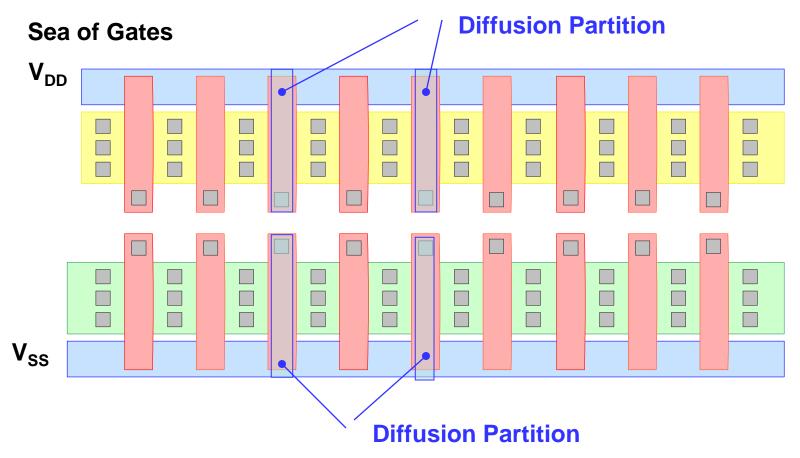


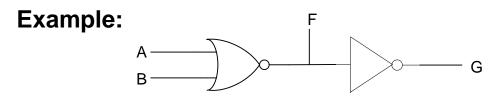
#### **Sea of Gates**





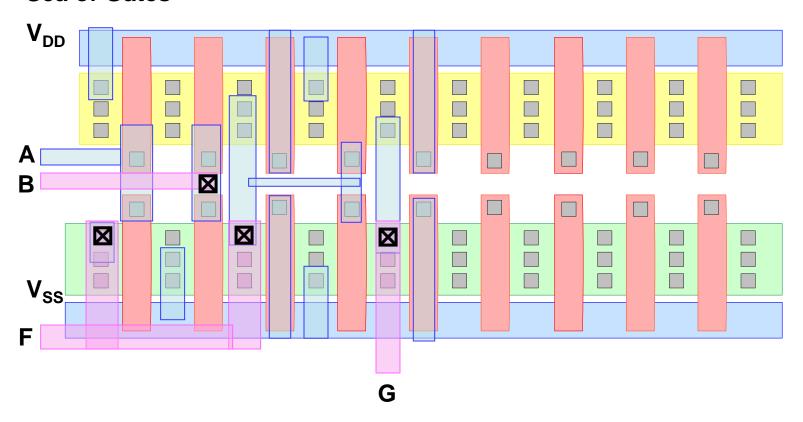
- ☑ Via (M1 to M2)
- Contact (M1 to diff,Poly)



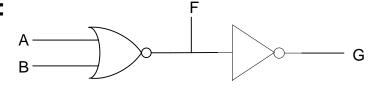


- **☒** Via (M1 to M2)
  - Contact (M1 to diff,Poly)

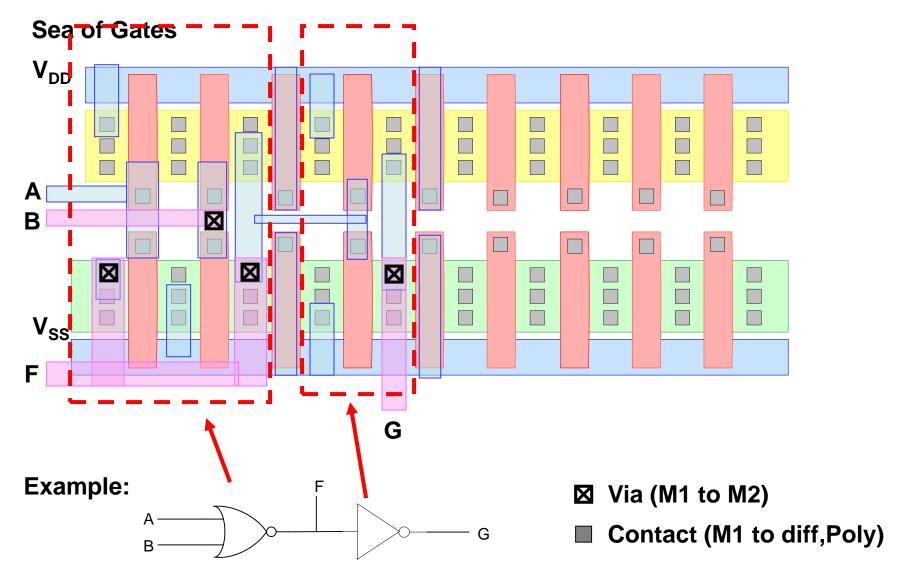
#### **Sea of Gates**



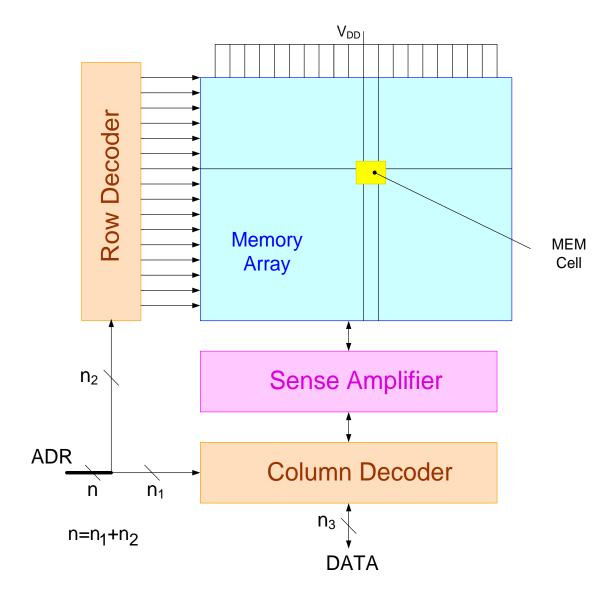
**Example:** 



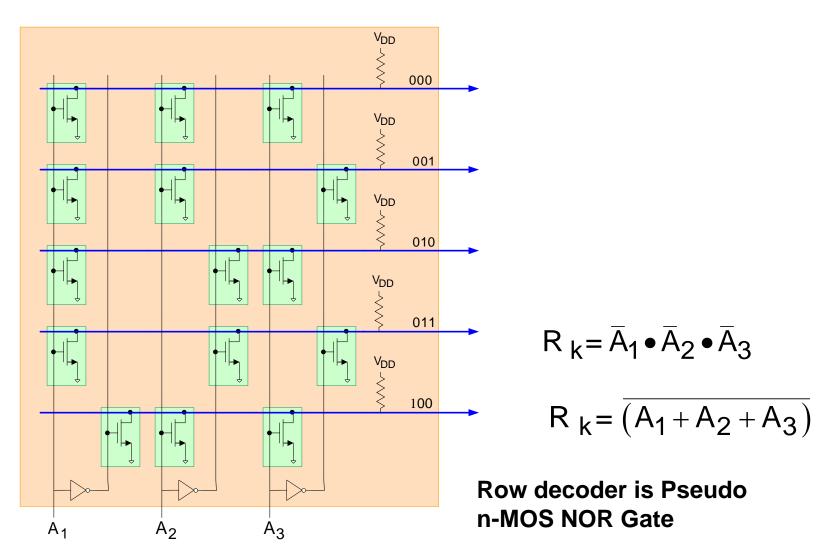
- ☑ Via (M1 to M2)
- Contact (M1 to diff,Poly)



## Typical Memory Structure

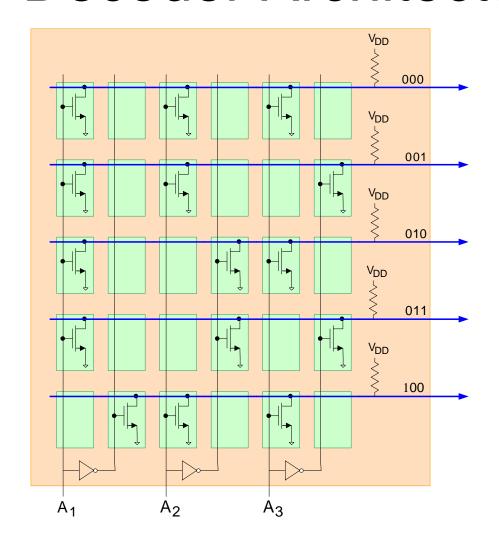


#### Row Decoder Architectures



Typically n/2 inputs where n is the address length

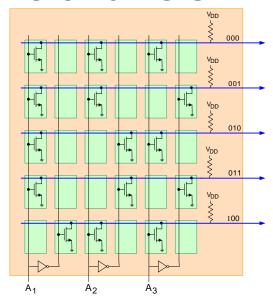
#### Row Decoder Architectures

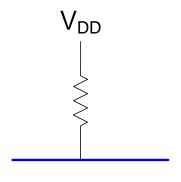


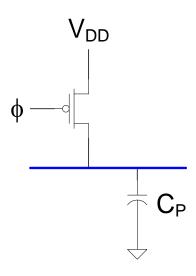
Transistor sites typically reserved in the layout for efficient, compact layout

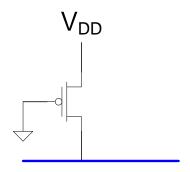
#### Row Decoder Architectures

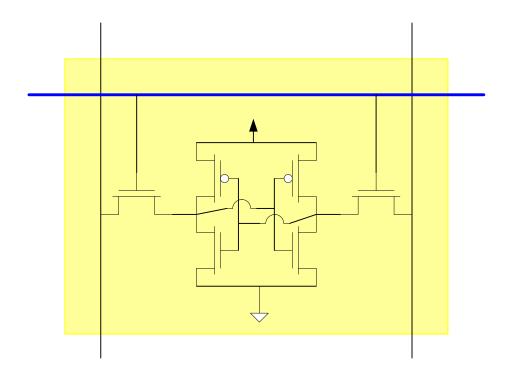
Pull-up resistor implemented with either weak p or with dynamic precharge by taking clock  $\phi$  low to precharge to high (thus dynamic NOR gate)





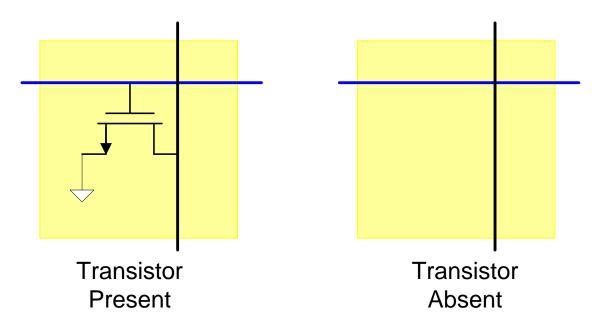






Static RAM (SRAM)

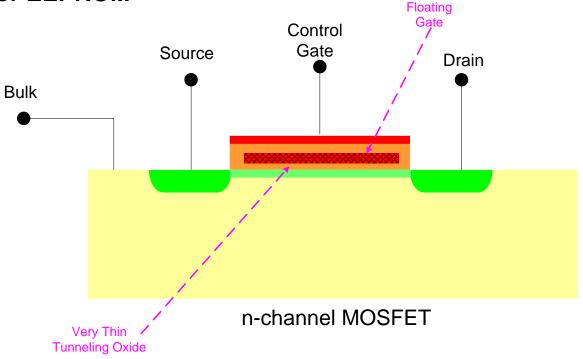
- Uses PTL and cross-coupled inverters
- Sizing of "switches" must be strong enough to write to cell
- No static power dissipation in this PTL implementation



Static ROM (Mask programmable ROM)

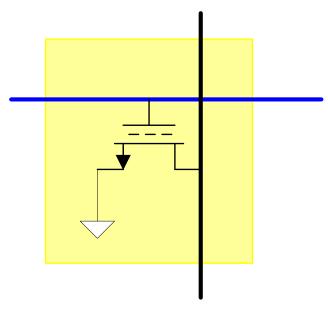
- Site reserved for possible transistor
- Actually programmed with contact to gate and diffusion
- Can personalize with one or two masks
- Single transistor per bit
- Uses only one column line

#### **EPROM or EEPROM**



**Floating Gate Transistor** 

- Very thin floating gate
- Charge tunnels onto gate during programming to change V<sub>T</sub> a lot
- Conceptual diagram only
- Somewhat specialized processing for reliable floating gate devices

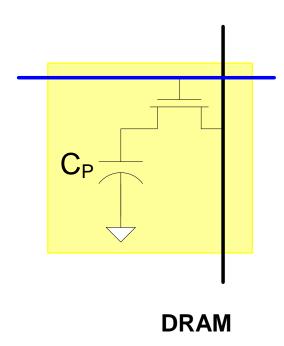


**EPROM or EEPROM** 

- 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



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

### **End of Lecture 44**