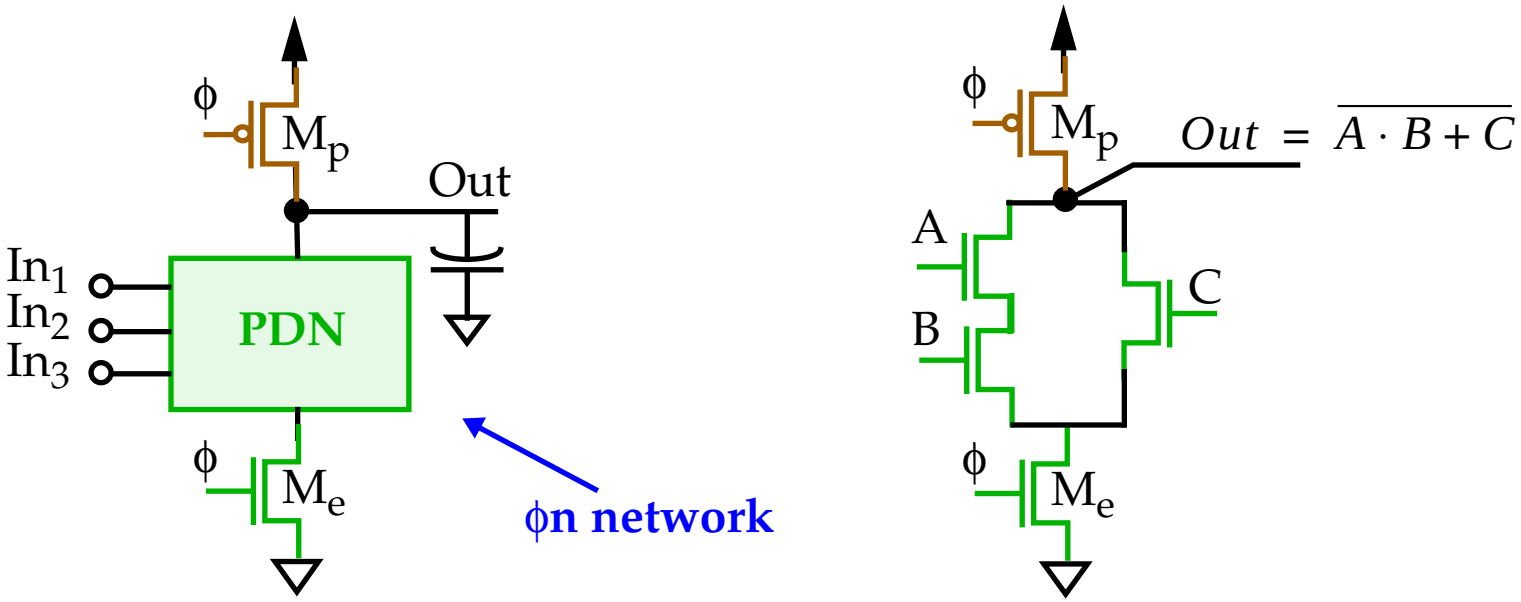
Dynamic Logic

Dynamic logic reduces the fan-in, similar to pseudo-NMOS, **without** the static power consumption.



Precharge:

When ϕ=0, the output node *Out* is precharged to V_{DD} by M_p.

Evaluation:

When ϕ=1, M_e is on and node *Out* discharges conditionally, depending on the value of the input signals.

Dynamic Logic

If no path exists during evaluate, then *Out* remains high via C_L (diffusion, wiring and gate capacitance).

Note that once *Out* is discharged, it cannot be recharged.

Therefore, the inputs can make *at most* **one transition** during evaluation.

Properties:

- The logic function is implemented in the NMOS pull-down network.
- The # of transistors is **N+2** instead of 2N
- It is non-ratioed (noise margin does not depend on transistor ratios).
- It only consumes dynamic power.
- Faster switching due to reduced internal and downsteam capacitance.

Steady-state behavior:

V_{OL} and V_{OH} are GND and V_{DD}.

Our standard definitions of noise margins and switching thresholds *do not include time*, which is required in this case.

Dynamic Logic

Steady-state behavior (cont):

For example, noise margins depend on the length of the evaluate.

If *clk* is too long, leakage affects the high output level significantly.

Since the pull down network starts to conduct when the input signal exceeds V_{Tn}, it is reasonable to set V_M, V_{IH}, $V_{IL} = V_{Tn}$.

Therefore, NM_L is very low.

Note that this is a conservative estimate since *subthreshold leakage* occurs for inputs below V_{Tn}.

Also note that the high output level is sensitive to noise and coupling disturbances because of its **high** output impedance.

The high value of NM_H compensates for this increased sensitivity.

Dynamic Logic

Dynamic behavior:

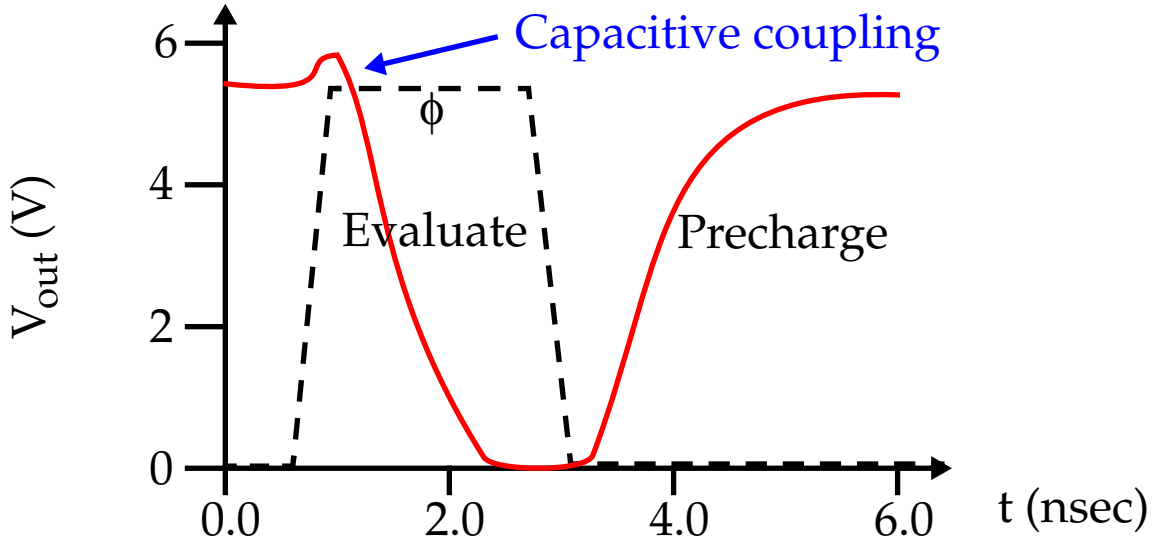
Also, after precharge, the output is high. Therefore, $t_{pLH} = 0$!

This is somewhat unfair since it ignores the precharge time.

The designer is free to choose the size of the PMOS device, smaller is faster but increases load and t_{pHL}.

The t_{pHL} is proportional to C_L and current-sinking capabilities of PDN.

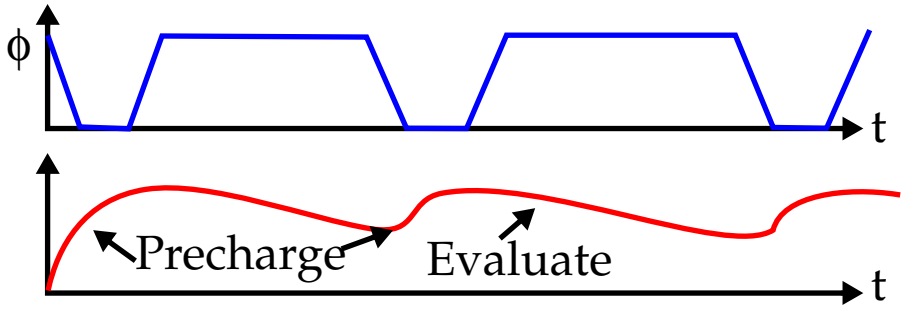
M_e slows down the gate a little.



Dynamic Logic

There are three sources of noise:

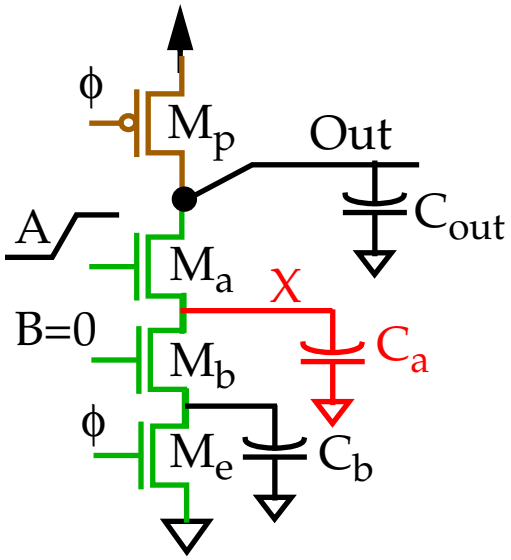
- *Charge Leakage*



Via reversed-biased diffusion diodes and subthreshold leakage

Sets the minimum clock to 250Hz to 1kHz (testing difficulties)

- *Charge Sharing*



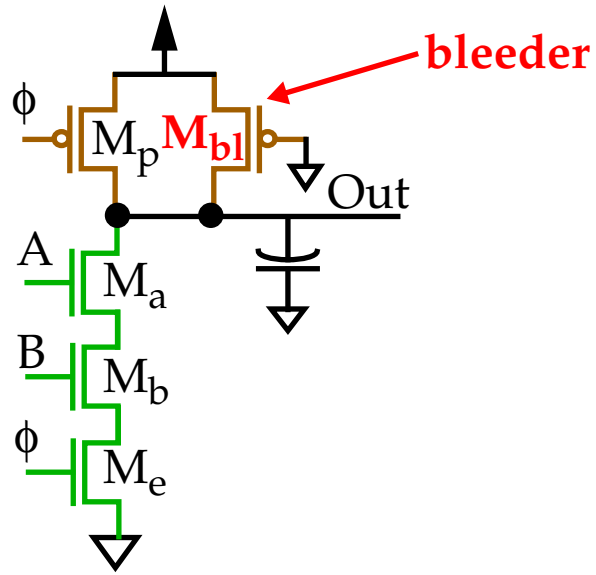
If $\Delta V_{out} > V_{Tn}$ then V_{out} and V_x reach the same value.

$$\Delta V_{out} = -V_{DD}\left(\frac{C_a}{C_a + C_L}\right)$$

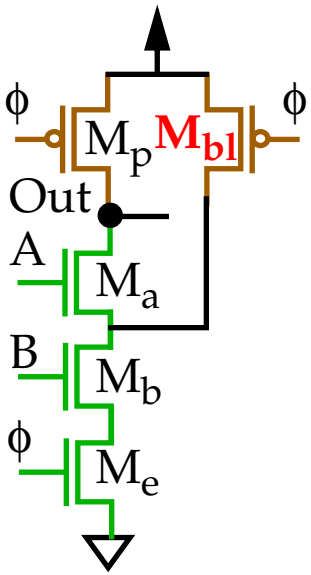
Target is to keep $\Delta V_{out} < |V_{Tp}|$ since output may drive a static gate. $C_a/C_L < 0.2$.

Dynamic Logic

One way to combat both of these:



Static bleeder



Precharge Internal nodes

Pseudo-static: M_{bl} is a highly resistive (long and narrow) PMOS transistor.

Alternatively, *precharge* internal nodes using a clock driven PMOS.

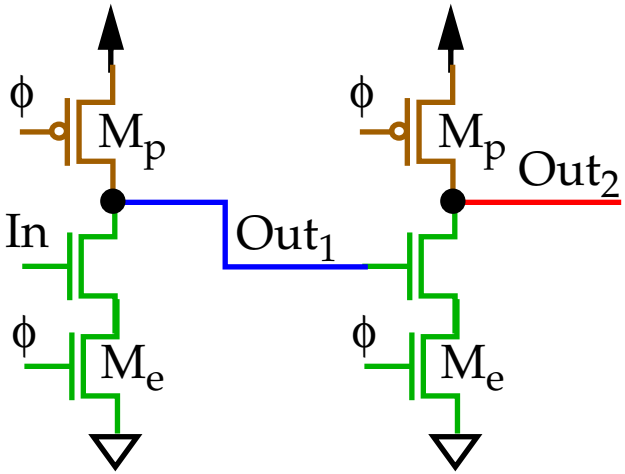
- *Clock Feedthrough*

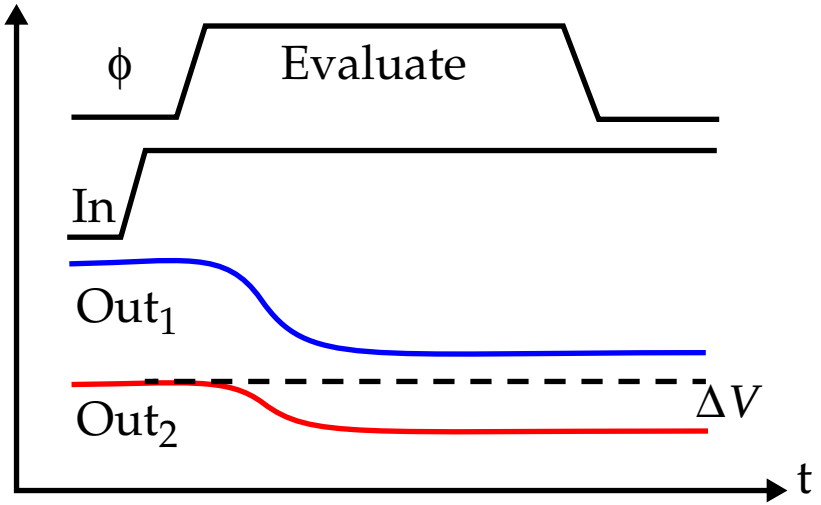
  The clock is coupled to the storage node via C_{gs} and gate-overlap caps.

  May forward bias the junction and inject electrons into substrate.

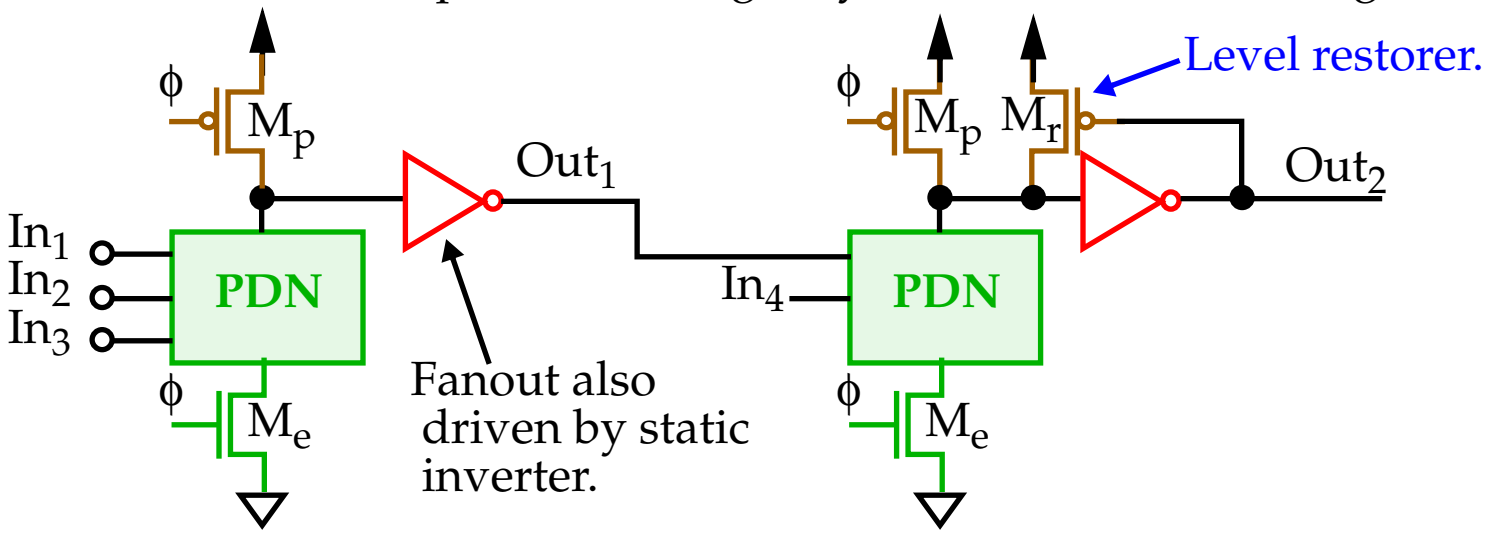
DOMINO Logic

Cascading Dynamic gates





Fix is to restrict the inputs to making only a *0->1* transition during eval.



DOMINO Logic

During evaluation, either the output of the first DOMINO stays at 0 (no delay!) or makes a *0->1* transition.

The transition may ripple all the way down the chain.

Properties:

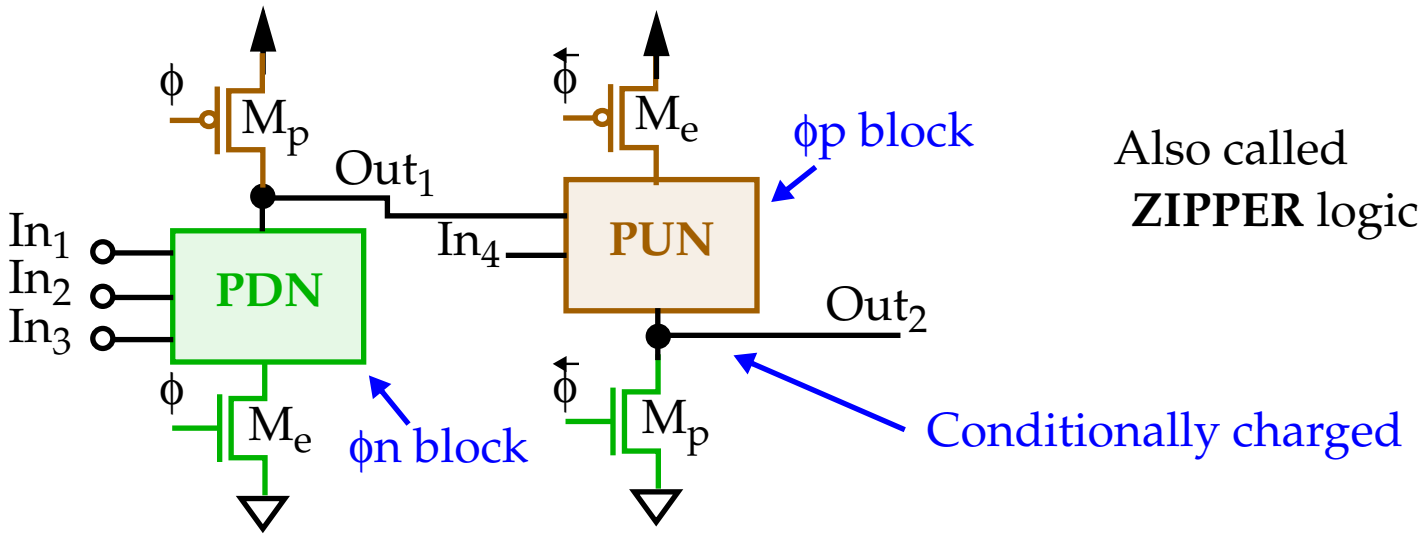
- Only **non-inverting** logic can be implemented.
- Appropriate for complex, large fan-out circuits such as ALUs or control circuits.
- Very high speeds can be achieved, $t_{pHL} = 0$.

In the past, DOMINO was used in the design of a number of high speed ICs. The first 32-bit microprocessor (BellMAC 32) used it.

Recently, pure DOMINO circuits are rare, mainly due to the non-inverting logic property.

np-CMOS Logic

PUN networks replace the static inverters.



Also called **ZIPPER** logic

Note that the ϕp blocks are driven with the *Clk_bar* so that the precharge and evaluate periods coincide.

np-CMOS logic style is **20%** faster than DOMINO, despite the slower PMOS pull-up devices.

The DEC alpha-processor (first at 250MHz) used this logic extensively.

Disadv: $NM_L = V_{Tn}$ and $NM_H = |V_{Tp}|$.

Power Consumption

We've already discussed sources of power consumption in CMOS inverter.

$$P_{dyn} = C_L V_{DD}^2 f_{0->1}$$

We now discuss the effects of *switching activity, glitching* and *direct-path* current.

Note that the factor $f_{0->1}$ complicates the analysis for complex gates.

Factors affecting the **switching activity** include the *statistics of the input signals*, the *circuit style* (dynamic/static), *the function*, and *network topology.*

These are incorporated by:

$$P_{dyn} = C_L V_{DD}^2 P_{0->1} f$$

where f is the average event rate, and $P_{0->1}$ is the **probability** an input transition results in a *0->1* power-consuming event.

Complex Static Gate Power Consumption

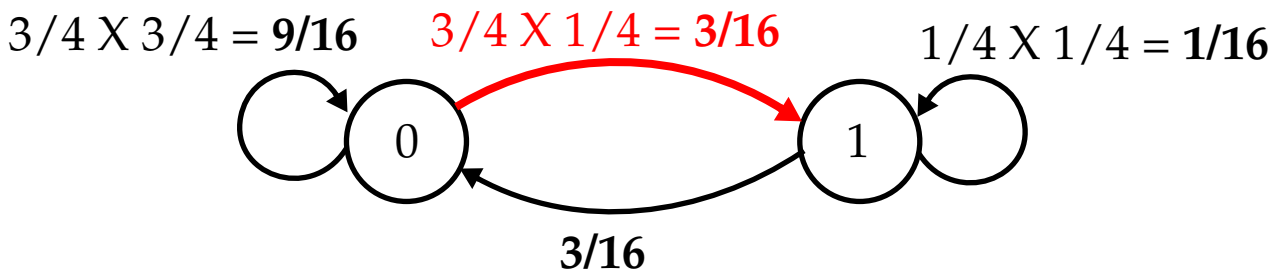
Consider a *2-input* NOR gate, assume the input signals have a uniform distribution of high and low values.

e.g., the 4 input combinations, *AB* = *00*, *01*, *10*, *11*, are equally likely.

Therefore, the probability the output is low or high is **3/4** and **1/4**, respectively.

The probability of an energy consuming transition is the probability that the output is initially low, 3/4, times the probability it will become high, 1/4.

$$P_{0->1} = P_0 P_1 = (1 - P_1) P_1 = \frac{3}{4} \times \frac{1}{4} = \frac{3}{16}$$



Complex Static Gate Power Consumption

Note that the output probabilities are **no longer uniform**.

This suggests that the input signals are not uniform, since gates are typically cascaded.

The probability that the output is 1 (P_1) is a function of the *input distributions*, P_A and P_B (the probabilities the inputs are 1).

$$P_1 = (1 - P_A)(1 - P_B)$$ for the NOR gate.

The transition probability is then:

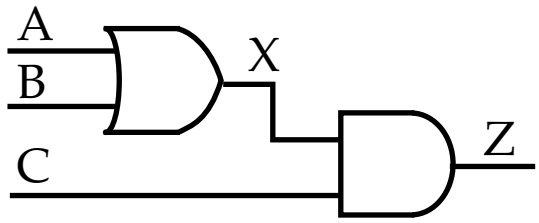
$$P_{0->1} = (1 - P_1)P_1 = [1 - (1 - P_A)(1 - P_B)][(1 - P_A)(1 - P_B)]$$

3-D graph shown in text.

Derive these expressions for AND, OR and XOR.

Complex Static Gate Power Consumption

For example:



No reconvergent fan-out

With no reconvergent fan-out, the probability that *X* undergoes a power consuming transistion is **3/16**.

X = *1*, 3 out of 4 times. Therefore, *X* has an uneven distribution yielding a transition probability on *Z* as:

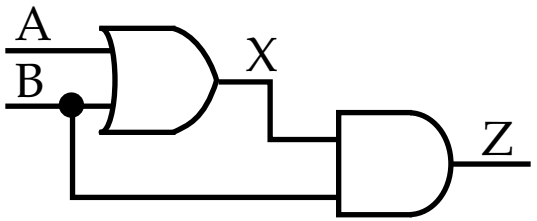
$$Z = (1 - P_X P_C) P_X P_C = \left(1 - \frac{3}{4} \times \frac{1}{2}\right)\left(\frac{3}{4} \times \frac{1}{2}\right) = \frac{15}{16}$$

The orderly calculations from input to output is not possible for

- Circuits with **feedback** (sequential circuits).
- Circuits with **reconvergent fanout**.

Complex Static Gate Power Consumption

In the latter case, the input signals are **not** independent.



Reconvergent fan-out

The procedure above yeilds **15/64** for the transition probability.

However, reduction yields $Z = B$, and the $P_{0->1}$ transition probability on Z is $(1/2 \text{ X } 1/2) = 1/4$.

Conditional probabilities take signal inter-dependencies into account.

For example, $Z = 1$ iff B and $X = 1$.

$$P_Z = P(Z=1) = P(B=1, X=1)$$

This expresses the probability that B and X are 1 simultaneously.

If a dependency exists, a *conditional probability* is required for expansion:

$$P_Z = P(X=1|B=1) \cdot P(B=1|X=1) = P(X=1|B=1) \cdot P(B=1)$$

Dynamic Gate Power Consumption

What about **dynamic circuits**?

During precharge, the output node is charged to 1.

Therefore, power is consumed every time the PDN is on (output is 0), independent of the preceding or following values!

Power consumption is determined solely by signal value probabilities, and **not** by transition probabilities.

These is always *larger* than the transition probability, since the latter is the product of two signal probabilities both of which is smaller than 1.

For example, the *0-probability* of a *2-input* NOR is

$$P_0 = (P_A + P_B - P_A P_B)$$

If the inputs are equally probably, there is a 75% chance of a *1->0*.

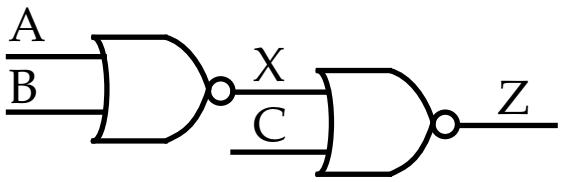
$$P_{NOR} = 0.75 C_L V_{DD}^2 f_{clk}$$

Note C_L is smaller than a static gate but the clock load must be considered.

Glitches in Static CMOS Circuits

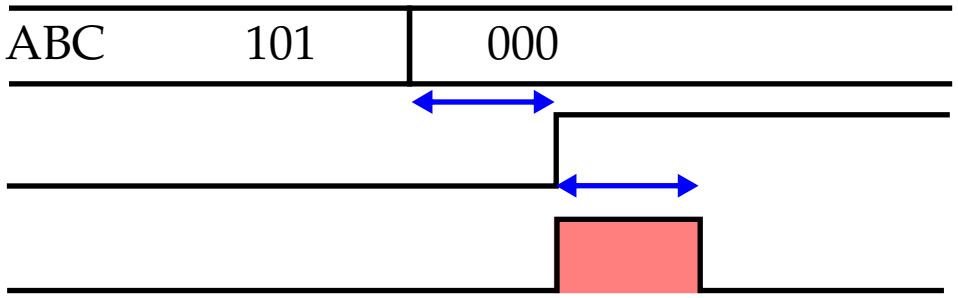
The finite propagation delay through gates in a network can cause spurious transitions called **glitches**, **critical races** or **dynamic hazards**.

These are multiple transitions during a single clock cycle.



Assume a unit delay and all inputs arrive at the same time.

The second NOR evaluates **twice**, the first one with the previous value of X. This consumes unnecessary power.



Unit delay

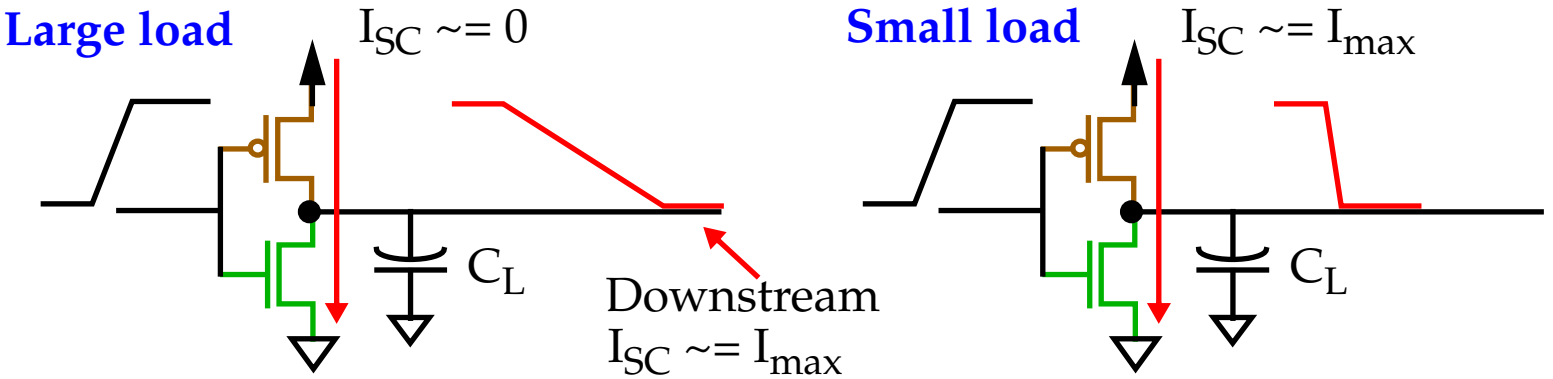
Redesign can eliminate glitches by matching delays along signal paths.

Short Circuit Currents in Static CMOS Circuits

Crowbar currents occur when both NMOS and PMOS are on simultaneously.

As is true for glitches, these do **not** occur in dynamic circuits.

The power dissipated is a function of the *on-time* of the transistors and their *operation mode*.



For large C_L (left), V_{DS} for the PMOS remains at *0* during entire input transition. I_{SC} is approximately *0* in this case.

For small C_L (right), V_{DS} is V_{DD} and is maximal (saturation).

Neither case is acceptable. Actually, **matching** rise/fall times is optimal.

Low Power Design

Assuming short-circuit current, glitching and leakage can be kept in bounds, the dominant power consumption is **dynamic power**.

$$P_{dyn} = C_L V_{DD}^2 P_{0->1} f$$

Power can be reduced by manipulating V_{DD} and C_{eff}, (C_L * $P_{0->1}$).

Reducing V_{DD} is a big win because of the **quadratic** dependence.

Although **PDP** decreases for lower V_{DD}, delay *increases*, as predicted by.

$$t_p \sim \left(\frac{C_L V_{DD}}{2}\right)\left[\frac{1}{k_n(V_{DD}-V_{Tn})^2} + \frac{1}{k_p(V_{DD}-|V_{Tp}|)^2}\right]$$

Delay increases substantially for V_{DD} close to $2*V_T$.

Therefore, to conserve energy, we should operate at the **slowest** possible speed.

Low Power Design

To maintain throughput (compensate for increased delays), one approach is to *lower* threshold voltages.

As we've seen, lowering threshold voltage, increases *subthreshold leakage*.

This raises the minimum clock frequency (dynamic circuits).

This increases standby currents and reduces noise margins (static circuits).

Note that the concept of zero leakage is preconceived.

For example, the following configurations yield the same performance in a 0.25 um CMOS process.

- V_{DD} = 3V, V_T = 0.7V
- V_{DD} = 0.45V, V_T = 0.1V

However, the power consumption is reduced in the latter by **45X**!

For dynamic circuits, the power savings is only about a factor of 8.

Architectural (**area for power**) compensations are also possible.

Low Power Design

When power supply voltage is lower bound due of external constraints or performance, the only other means is to reduce C_{eff}.

This is achieved by reducing both the *physical cap*. and *switching activity*.

Lowering the physical cap. usually improves performance as well.

For example, a CPL adder reportedly uses 30% less power (at 4V) compared with a static version.

Since most cap. is due to *transistor cap.* (diffusion and gate), this suggests the use of minimum sized devices whenever possible.

Larger is only justified for large fan-outs and wiring capacitances.

Note this contradicts the standard cell philosophy which use larger transistors in order to accommodate a wider ranges of loads.

Reducing switching activity can be accomplished by *point-to-point buses* and *re-ordering inputs* to gates (see text for examples).

Summary

Choosing a logic style depends on Ease of design, Robustness, System clocking requirements, Fan-out, Functionality and Testing.

Static is robust and easy to design (ameanable to design automation).

Complementary complex gates are expensive in area and performance.

Pseudo-NMOS is simple and fast but reduces noise margins and increases power consumption.

Pass-transistor logic is good for certain classes of circuits (MUX/adders).

Dynamic logic gives fast and small circuits but complicates the design process and restricts the minimum clock rate.

For a *4-input* NAND gate:

Style	Ratioed	Static power	# of trans.	Area (um^2)	delay (ns)
Complementary	No	No	8	533	0.61
Pseudo-NMOS	Yes	Yes	5	288	1.49
CPL	No	No	14	800	0.75
Dynamic (np)	No	No	6	212	0.37