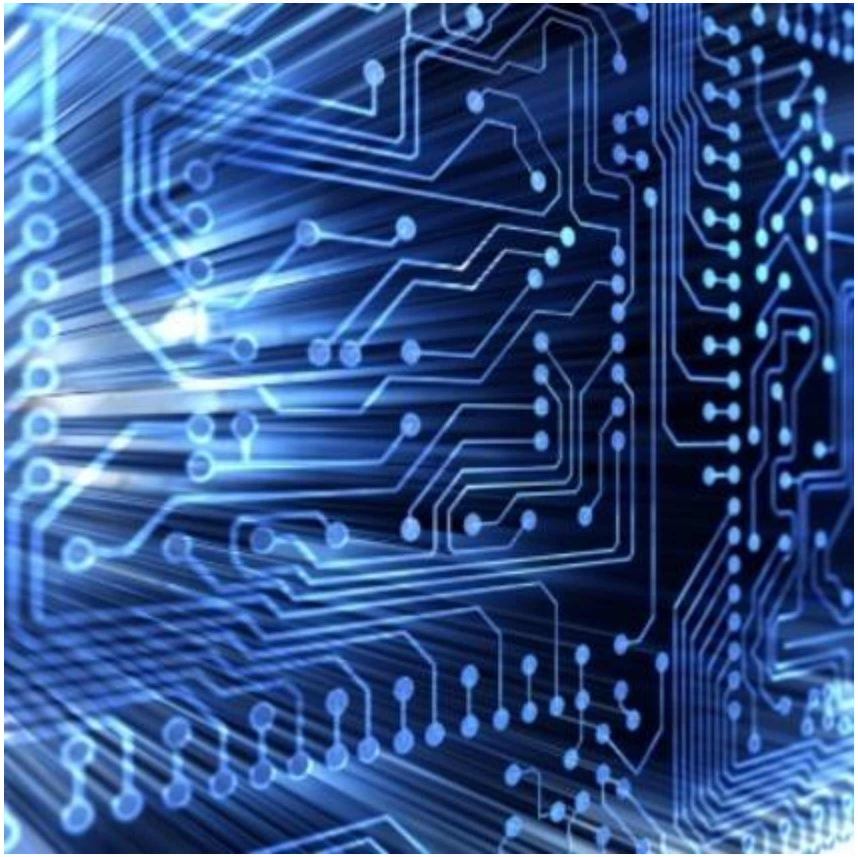
Digital Microelectronic Circuits
(361-1-3021)

Presented by: Adam Teman



Lecture 9:

Pass Transistor Logic

Motivation

- In the previous lectures, we learned about ***Standard CMOS Digital Logic*** design.
- ***CMOS*** is unquestionably the leading design family in use today, do to its many advantages and relative simplicity. However, it has a number of drawbacks that have led to the development of alternative solutions.
- The main drawback of ***Standard CMOS*** is its relatively large area (***2N transistors*** to implement an ***N-input gate***).
- In this lecture, we will learn about an **alternative logic family** that tries to *reduce the number of transistors* needed to implement a logic function, and achieve faster switching times.

What will we learn today?

- 9.1 Pass Transistor Logic
- 9.2 Extending the PTL Concept
- 9.3 Transmission Gates
- 9.4 PTL Logical Effort

9.1

What happens if we look at a MOSFET from the diffusions, instead of through the gate?

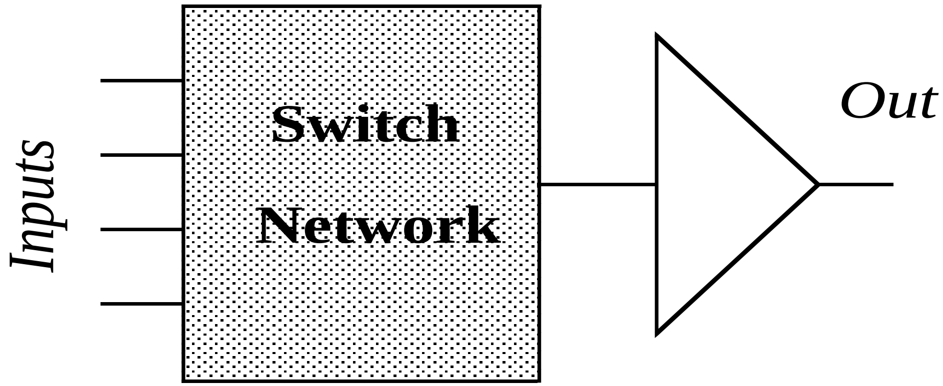
PASS TRANSISTOR LOGIC (PTL)

PTL Concept

- A popular and widely-used alternative to ***Standard CMOS*** is ***Pass Transistor Logic*** (***PTL***).
- ***PTL*** attempts to reduce the number of transistors required to implement logic by allowing the ***primary inputs*** to drive ***source*** and ***drain*** terminals in addition to the ***gate*** terminals.
- Using ***PTL***, we can reduce the number of transistors to implement a ***2-input AND gate*** to ***4*** (instead of ***6*** for ***Standard CMOS***).
- Broadening the ***PTL Concept***, we can make some more interesting gates.

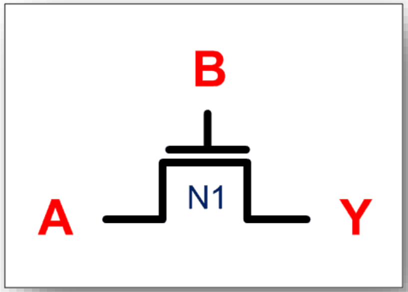
Relay Multiplexers

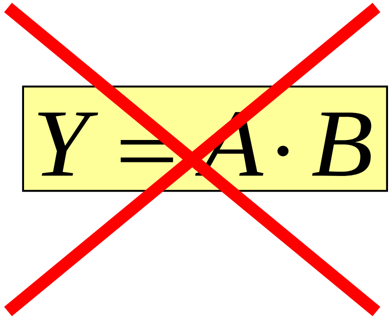
- The ***Pass Transistor*** concept is based on the use of ***relay switches***.
- A number of inputs are connected to ***switches*** and only ***one of the switches is chosen*** to be transferred to the output.
- In essence, we have created a ***Multiplexer***:



PTL Concept

- A simplification of the relay multiplexer would be to connect two inputs to a single nmos transistor – one to the ***gate*** and the other to ***one of the diffusions*** (***source/drain***):

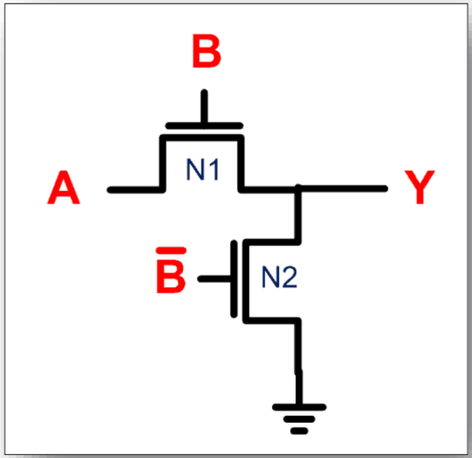




- It looks like we got an ***AND gate*** with a ***single transistor***:
  - When ***B='1'***, it *passes* ***A*** to the output.
  - When ***B='0'*** it *blocks* the output.
- But this is incorrect, as when the ***nMOS*** is ***switched off*** and the ***output node*** stays ***floating***, its value depends on its ***previous state***.

PTL AND Gate

- In fact, this type of a ***switch*** is often used in digital and analog circuits, but it is ***not*** an ***AND gate***.
- We'll take this basic operation and produce an ***AND gate*** by adding a ***path to GND*** when ***B='0'***.
- We can get this by adding an ***nMOS*** with its ***gate*** connected to ***B_*** and its ***source*** connected to ***GND***.
- This is a basic ***PTL AND Gate***!

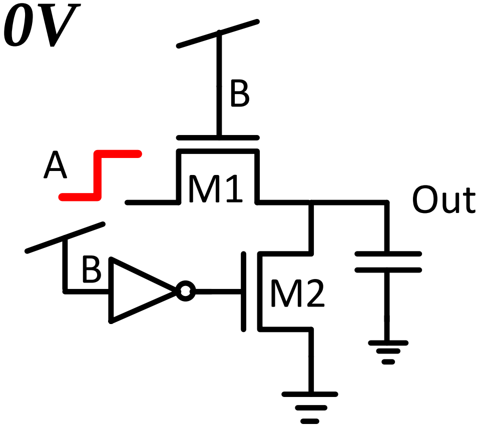


$$Y = A \cdot B$$

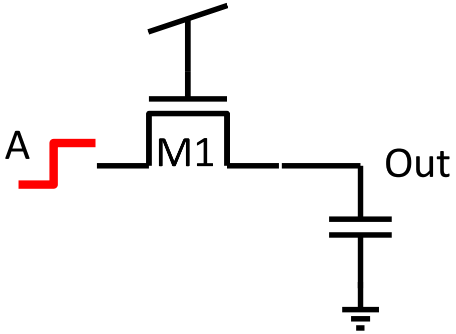
- It's comprised of a total of ***4 transistors*** because we need an ***inverter*** to get ***B_***.

Example – t_{pd} of PTL AND Gate

- Let's find the delay of a *'0'→'1'* transition from the diffusion input.
- Assume that at *t<0*, ***B='1', A*** rises and $V_{out}=0V$

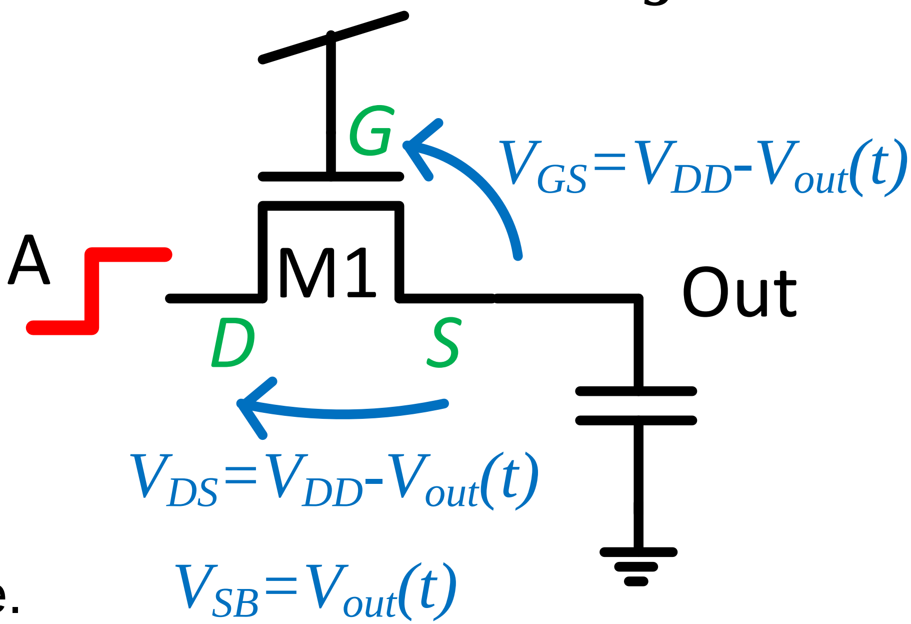


- Since ***M2*** is cut-off, we can just remove it from our equivalent model:



Example – t_{pd} of PTL AND Gate

- Now let's mark the ***source*** and ***drain*** and the ***bias voltages***:



- We see that:
  - The gate's overdrive ($V_{GS}-V_T$) is a function of the output voltage.
  - V_{DS} is a function of the output voltage.
  - V_{SB} is non-zero, so we have to regard the ***body effect***.

Example – t_{pd} of PTL AND Gate

- We'll check two points for delay, ***t=0*** and ***t=t_pd***:
  - At ***t=0***:

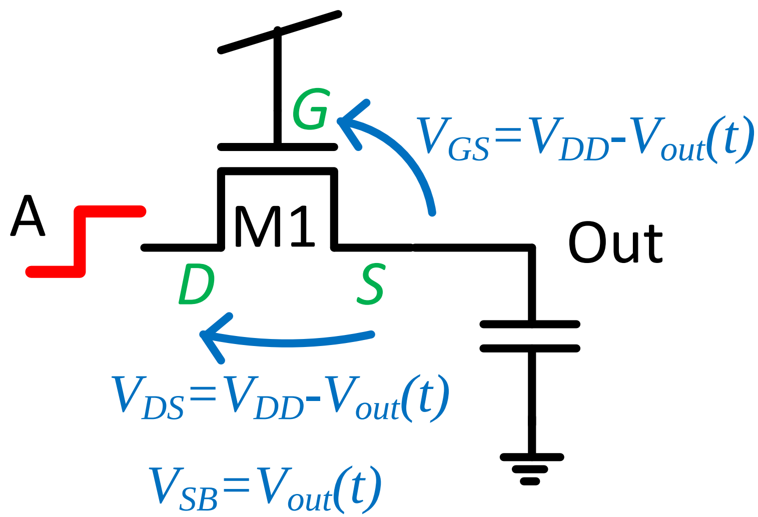
$$\left.\begin{aligned} V_{GS} &= V_{DD} - 0 = V_{DD} \\ V_{DS} &= V_{DD} - 0 = V_{DD} \\ V_{DSeff} &= \min\left(V_{DS}, V_{DSAT}, V_{GT}\right) = V_{DSAT} \end{aligned}\right\} Vel.Sat$$

$$V_{SB} = 0 \Rightarrow V_T = V_{T0}$$

  - At ***t=t_pd***:

$$\left.\begin{aligned} V_{GS} &= V_{DD} - {V_{DD}}/{2} = {V_{DD}}/{2} \\ V_{DS} &= V_{DD} - {V_{DD}}/{2} = V_{DD} - {V_{DD}}/{2} \\ V_{DSeff} &= \min\left(V_{DS}, V_{DSAT}, V_{GT}\right) = V_{GT} \end{aligned}\right\} Sat^*$$

$$V_{SB} = {V_{DD}}/{2} \Rightarrow V_T > V_{T0}$$



*Depending on given values…

Example – t_{pd} of PTL AND Gate

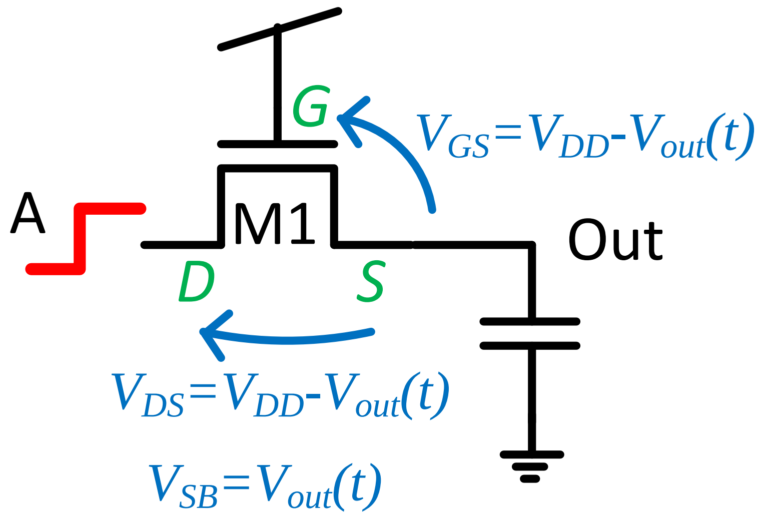
- To find t_{pd}, we need to solve an ***integral on the current***:

$$i_c = c\frac{dv_c}{dt} \Rightarrow dt = c\frac{dv_c}{i_c}$$

$$\int_0^{t_{pd}} dt = \int_0^{V_{DD}/2} \frac{c}{i_c} dv_c = c\left(\int_0^{V_{GT}=V_{DSAT}} \frac{dv_c}{i_c} + \int_{V_{GT}=V_{DSAT}}^{V_{DD}/2} \frac{dv_c}{i_c} \right)$$

$$i_c = k_n\left(V_{GT}V_{DSeff} - 0.5V_{DSeff}^2\right)\left(1+\lambda V_{DS}\right)$$

$$V_T = V_{T0} + \gamma\left(\sqrt{\left|-2\Phi_F + V_{SB}\right|} - \sqrt{\left|-2\Phi_F\right|}\right)$$

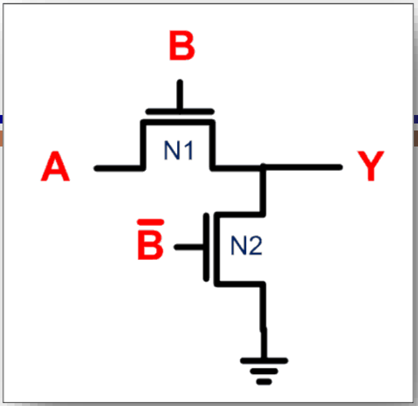


- But since this is "long and ugly", we can probably just take ***average currents***.

$$t_{pd} \approx c\frac{\Delta v_c}{i_{avg}} \approx c\frac{V_{DD}/2}{0.5\left(i_{t=0} + i_{t=t_{pd}}\right)}$$

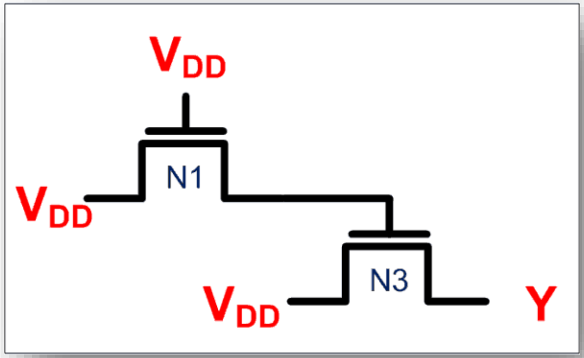
Cascading PTL AND Gates

- This ***AND gate*** has a big drawback...
- Remember that ***nMOS*** transistors pass a ***Weak '1'***?



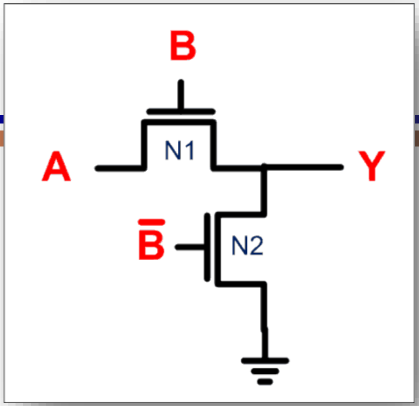
$$Y = A \cdot B$$

- Well, we can see that V_{OHmax} of this gate is only $V_{DD}-V_{Tn}$, at which point the switch will ***turn off***.
- This means that we cannot drive another ***PTL gate input*** with this ***output***.



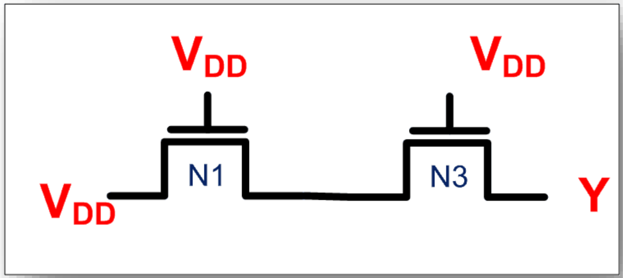
$$Y = V_{DD} - 2 \cdot V_{Tn}$$

Cascading PTL AND Gates



$$Y = A \cdot B$$

- However, we can connect the ***output*** to the next gate's ***diffusion input***:



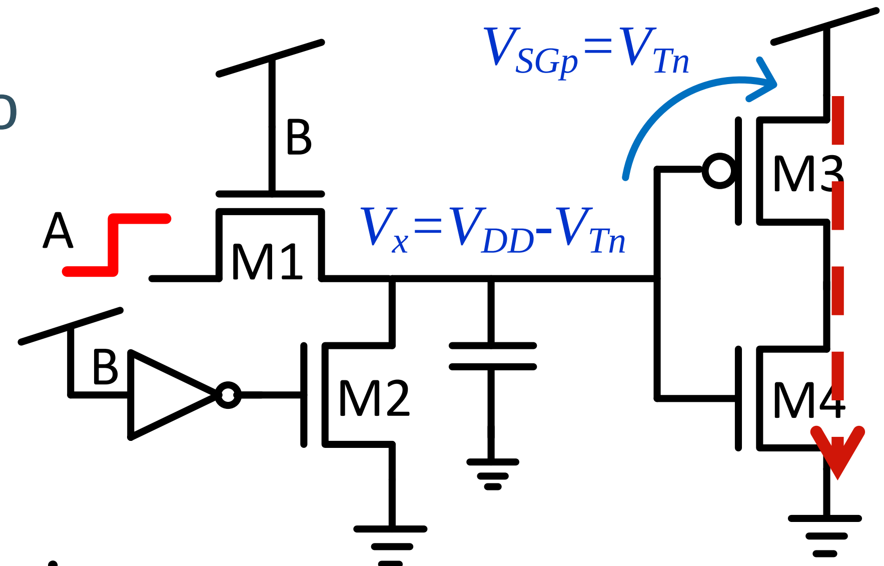
$$Y = V_{DD} - V_{Tn}$$

- There is some ***signal degradation***, so we need to add a ***CMOS Inverter*** every few gates to ***replenish*** the level.

- While this gate requires ***less power*** than a ***CMOS AND*** (***lower capacitance***, ***reduced swing***), it may cause ***static power*** on the ***partially on inverters*** it drives.

Static Power Problem

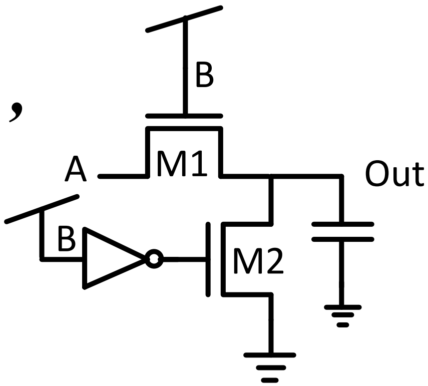
- For example, let us cascade an inverter after a ***PTL AND*** gate and drive the input ***high***.
- The output will be pulled up to $V_{DD}-V_{Tn}$, but due to the ***body effect***, $V_{Tn}>V_{Tn0}$.
- The input to the next stage provides $V_{SGp}=V_{DD}-(V_{DD}-V_{Tn})$. If this is larger than V_{Tp}, then the ***pmos*** is conducting and ***static current*** will flow freely.

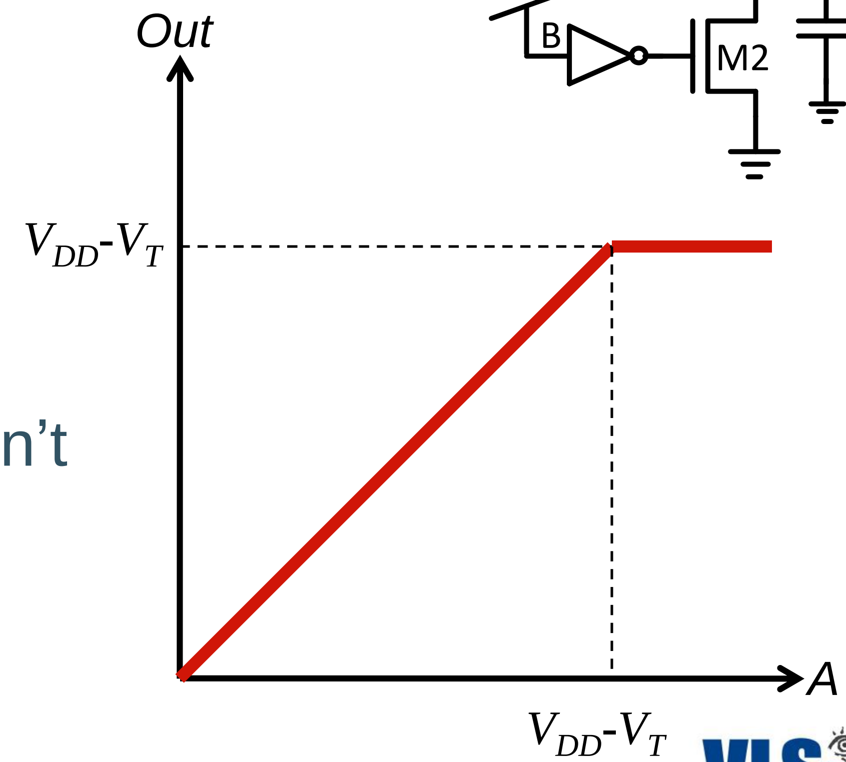


- Even if it $V_{SGp}<V_{Tp}$, this transistor is in ***weak inversion*** and dissipates substantial ***static power***.

PTL AND VTC

- To analyze the static properties of the ***PTL AND*** gate, we will draw its ***VTC***.
- We'll start with the ***VTC*** from ***A*** to ***Out*** with ***B='1'***
- In this case, the output simply follows the input until the pass transistor closes at $V_{DD}-V_T$.

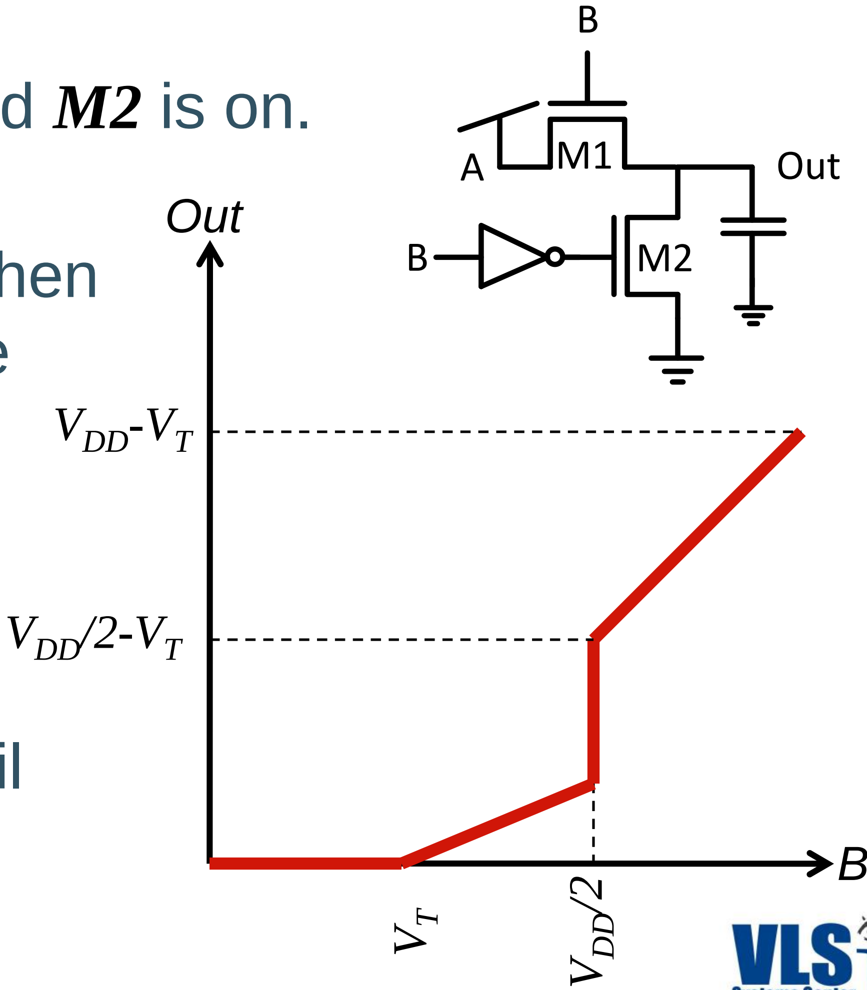




- In other words, this input doesn't have the required ***regenerative property*** for a digital gate!

PTL AND VTC

- What about the ***VTC*** from ***B*** to ***Out*** with ***A='1'***? This case is more complex.
- Starting at $B<V_T$, ***M1*** is off and ***M2*** is on. We get $V_{out}=0$.
- ***M2*** is on until $B=V_{DD}/2$, but when $B=V_T$, ***M1*** turns on. Therefore V_{out} will slowly rise with ***B***.
- At $B=V_{DD}/2$, ***M2*** turns off and ***M1*** has no contention.
- Therefore, V_{out} will "jump" to $V_{DD}/2-V_T$ and rise linearly until $V_{OHmax}=V_{DD}-V_T$

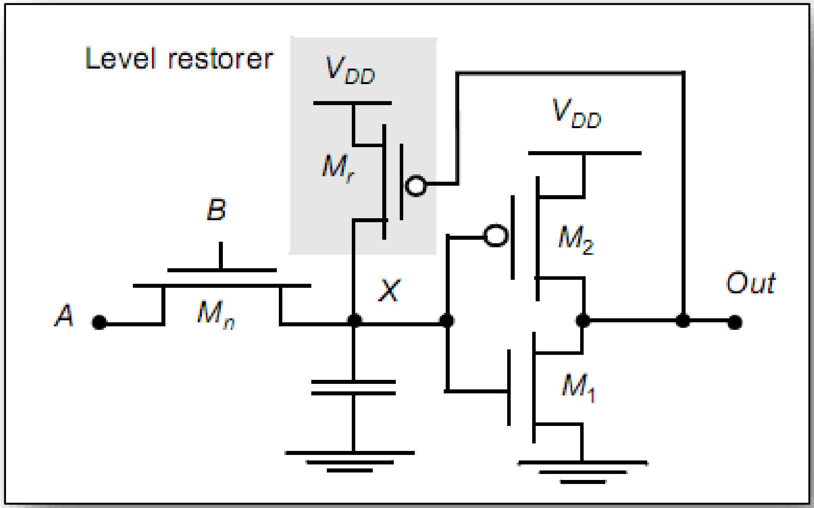


PTL AND Gate Summary

- ***PTL*** gates are ***non-regenerative*** and therefore ***not digital***.
  - To use them as digital gates they must be followed by a CMOS buffer!
- ***PTL*** gates do not present a ***rail-to-rail swing***
  - Therefore cascaded stages may dissipate static power.
  - Cascading PTL gates through gate inputs causes loss of signal and is therefore not allowed.
- However, certain functions can be implemented with ***fewer transistors*** than ***CMOS***
  - And in certain cases, specific transitions may be faster.

Level Restoration

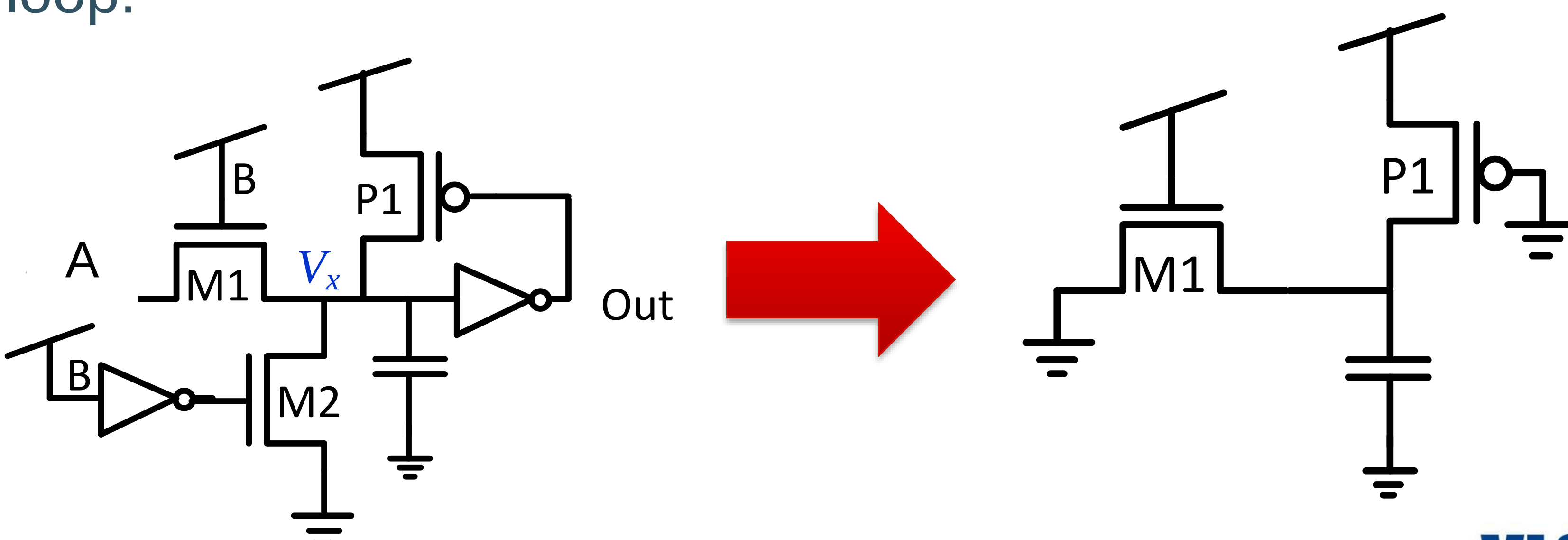
- One of the options to solve the problem of the ***Weak '1'*** is ***Level Restoration***.
- This can be achieved by using a ***PTL AND gate***, followed by an ***inverter*** with a ***feedback loop*** to a ***pMOS*** transistor.
  - When node ***X*** is high ($V_{DD}-V_{Tn}$), the ***Inverter*** outputs a ***'0'***, opening the ***pMOS "bleed" transistor***.
  - This restores the level at ***X*** to V_{DD}.
  - When node ***X*** makes a ***'1'*** to ***'0'*** transition there is a "***fight***" between the ***bleed transistor*** and the ***low input***.



- This means we need careful ***Ratioed Sizing*** to make the circuit work properly.

Level Restorer Sizing

- The level restorer "fights" the pass transistor when pulling down through the diffusion input.
- Therefore the pass transistor must be strong enough to flip the cascaded inverter.
- We will solve this problem by disconnecting the feedback loop:

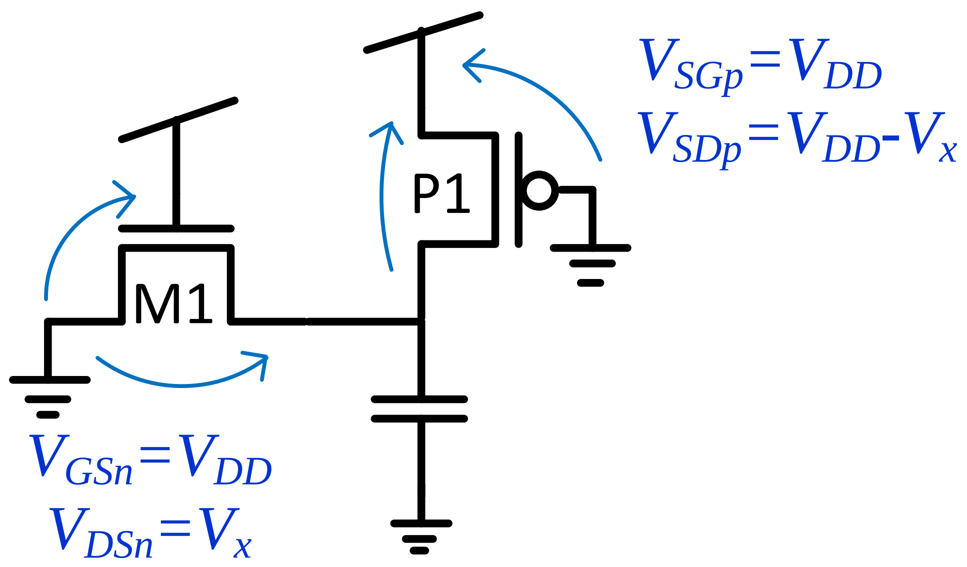


Level Restorer Sizing

- Now we just have to make sure that the stable state of V_X is lower than the inverter's V_M.

$$I_{DSn}(sat) = I_{SDp}(vel.sat)$$

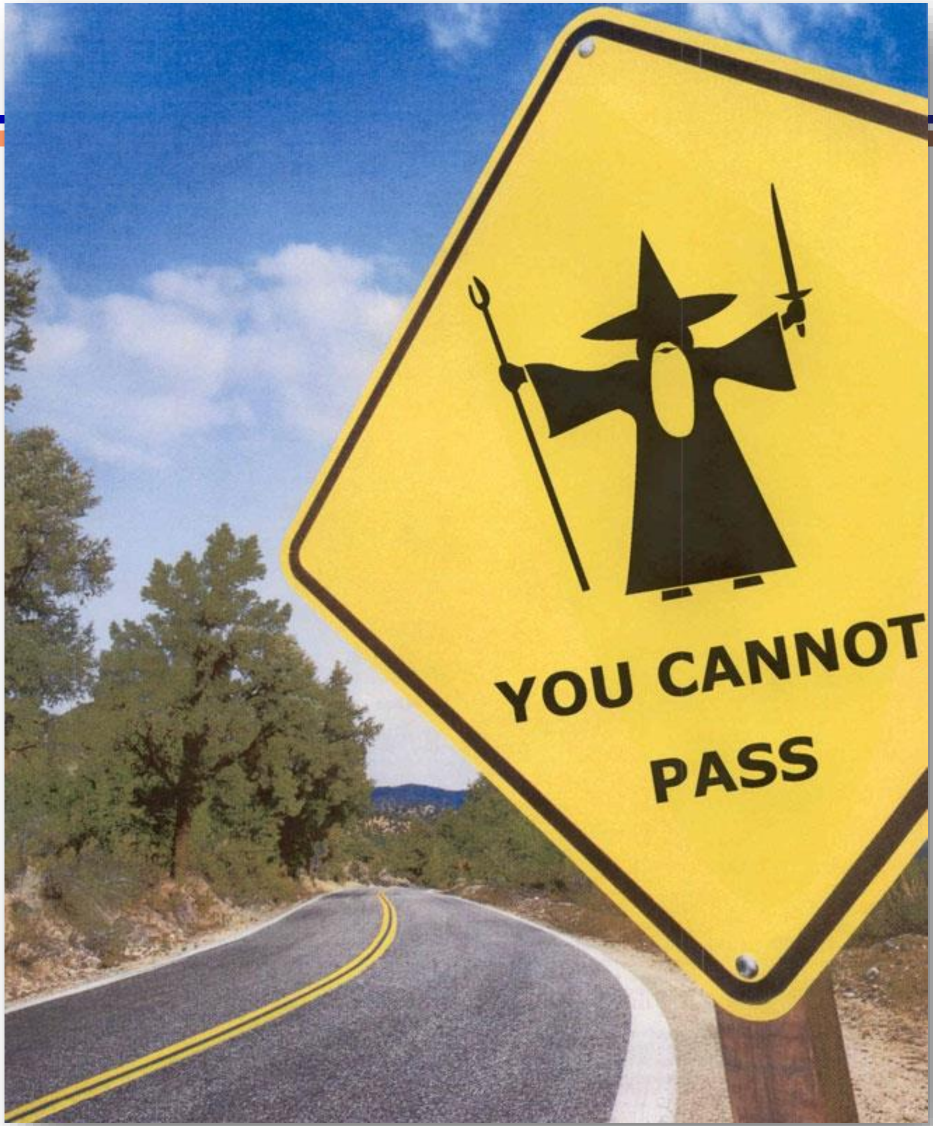
$$find\ {}^{k_n}\!/\!_{k_p} \Rightarrow V_x < {}^{V_{DD}}\!/\!_{2}$$



- *Advice from the guys who write the test…*

Solve this problem at home!

9.2

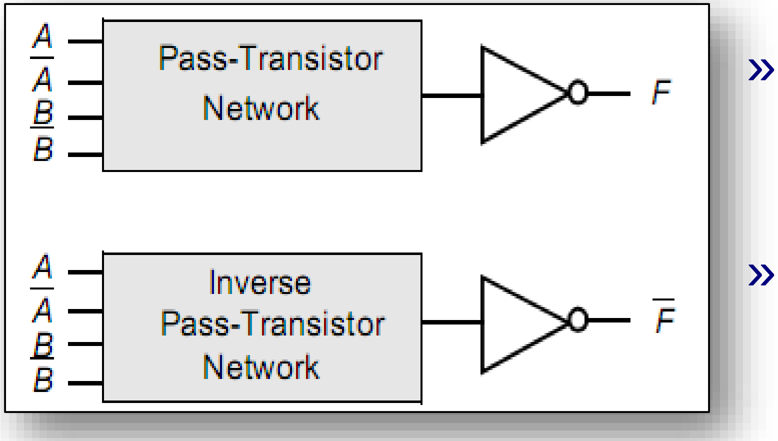


So based on the pass transistor concept, let's try to compose some useful circuits

EXTENSION OF THE PTL CONCEPT

CPL

- Using the ***PTL concept***, we can assemble an interesting ***highly modular*** gate family called ***Differential*** or ***Complementary Transmission Logic*** (***DPL*** or ***CPL***).

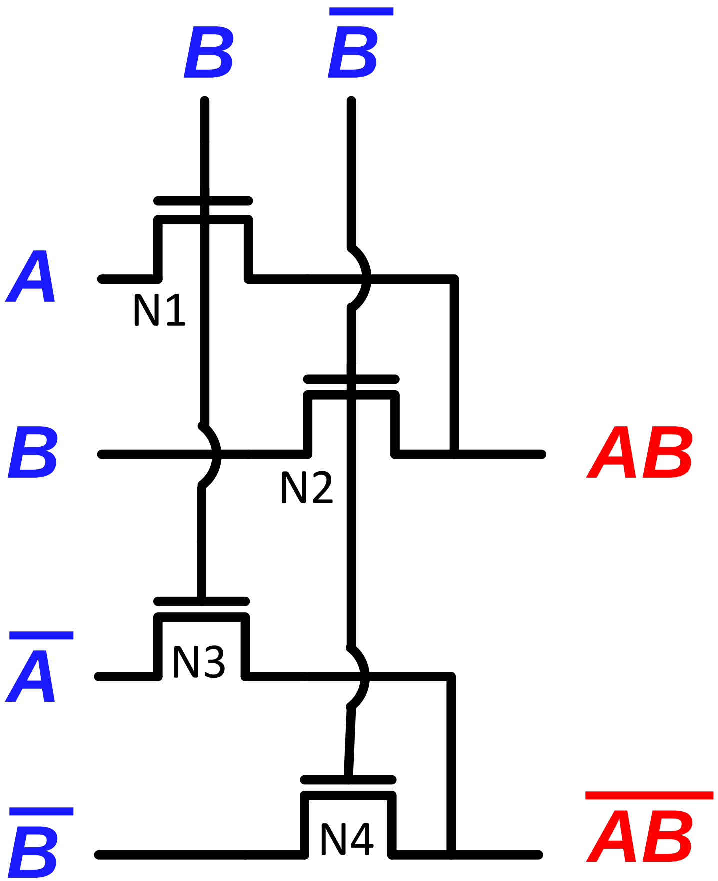


  - These gates inherently create ***differential outputs***, in other words, both ***a logic function*** and ***its complement***.
  - These can reduce the overall transistor count, as the extra ***inverters*** aren't needed.

- ***CPL gates*** enable us to efficiently realize some complex gates, such as ***XORs*** and ***Adders*** with a relatively small number of transistors.
- All ***CPL*** gates have the same ***topology***, using ***4 pass transistors*** and ***complementary inputs***.

CPL

- If we take the basic topology and connect different inputs, we can make many different functions:

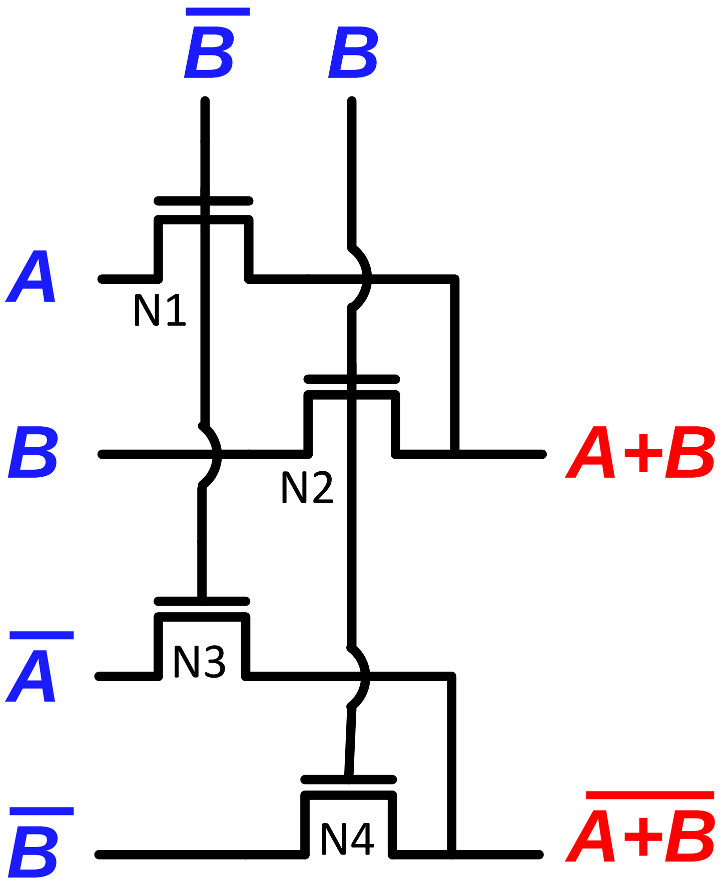


A	B	f
0	0	
0	1	
1	0	
1	1	

A	B	f
0	0	
0	1	
1	0	
1	1	

CPL

- If we take the basic topology and connect different inputs, we can make many different functions:

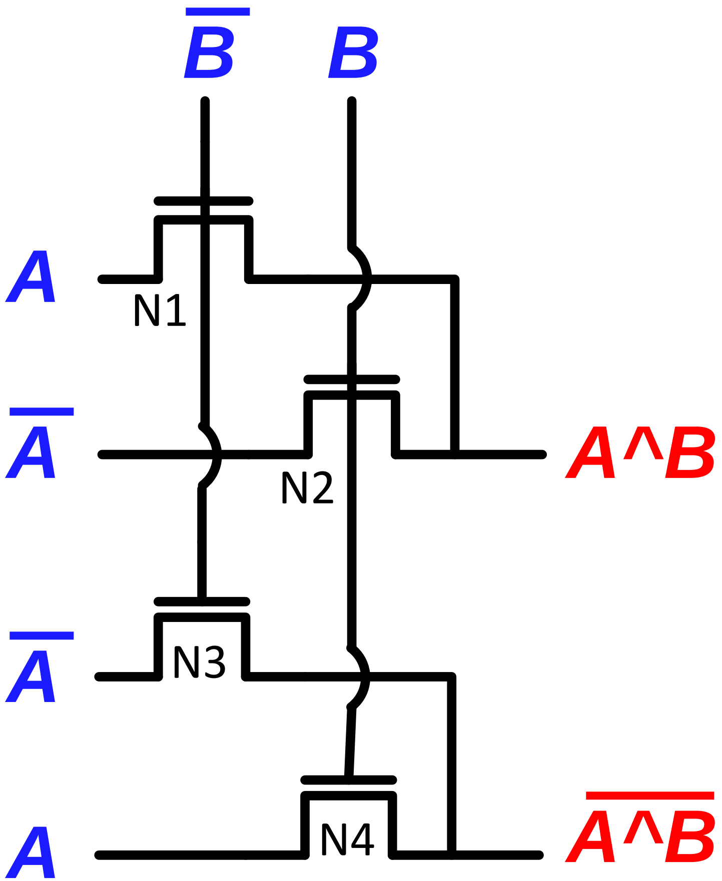


A	B	f
0	0	
0	1	
1	0	
1	1	

A	B	f
0	0	
0	1	
1	0	
1	1	

CPL

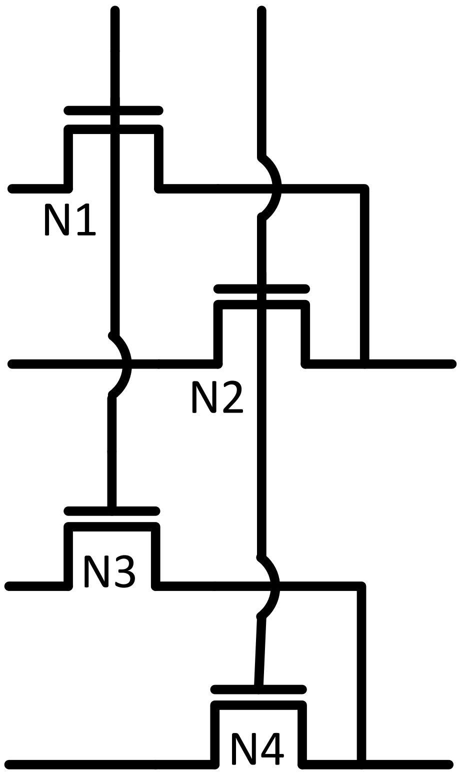
- If we take the basic topology and connect different inputs, we can make many different functions:



A	B	f
0	0	
0	1	
1	0	
1	1	

A	B	f
0	0	
0	1	
1	0	
1	1	

Solving the Weak '1' Problem in CPL



- Doesn't load the output.
- Less of a ratio problem
(the restorer is turned off by the opposite circuit).

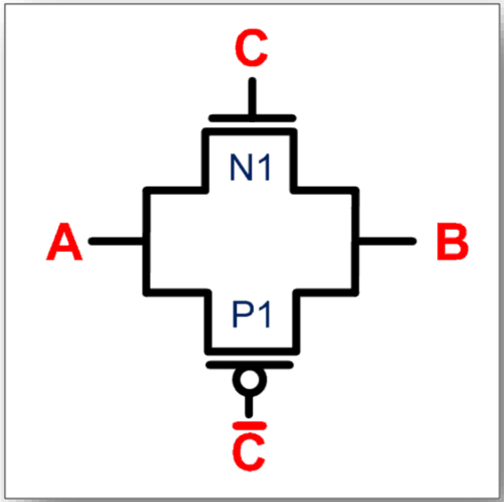
9.3

So PTL has its drawbacks, but we will often find the concept used as part of the

TRANSMISSION GATE

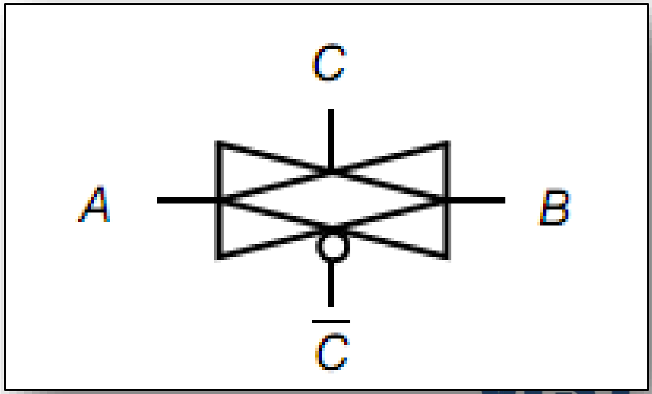
Transmission Gates

- The most commonly used implementation of ***PTL architecture*** is in ***Transmission Gates***.
- These gates use an ***nMOS*** and a ***pMOS*** connected in ***parallel***, utilizing the advantages of each.
- In this way, we can get both a ***Strong '1'*** and a ***Strong '0'***, thus achieving a ***full swing***.



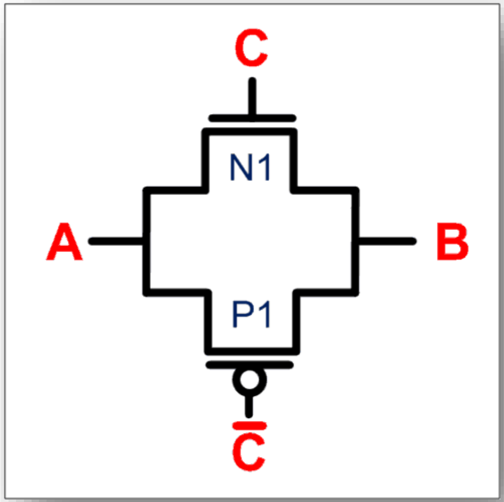
$$B = A,\ if\ C = '1'$$

- The basic ***Transmission Gate*** is a ***bidirectional switch***, passing a signal through when the ***control signal*** is ***on***.
- The symbolic representation of a ***Transmission Gate*** is shown here:



Transmission Gates

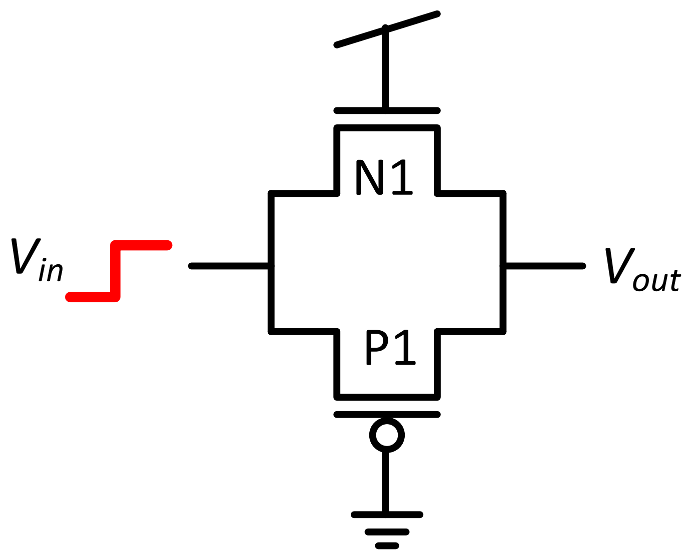
- The ***Transmission Gate*** uses ***4 transistors*** (the ***inverted*** control signal is needed to control the ***pMOS***).
- This means that it doesn't necessarily ***reduce the area*** to implement logic functions, but in certain cases, ***very efficient functions*** can be easily realized.



$$B = A, \ if \ C = '1'$$

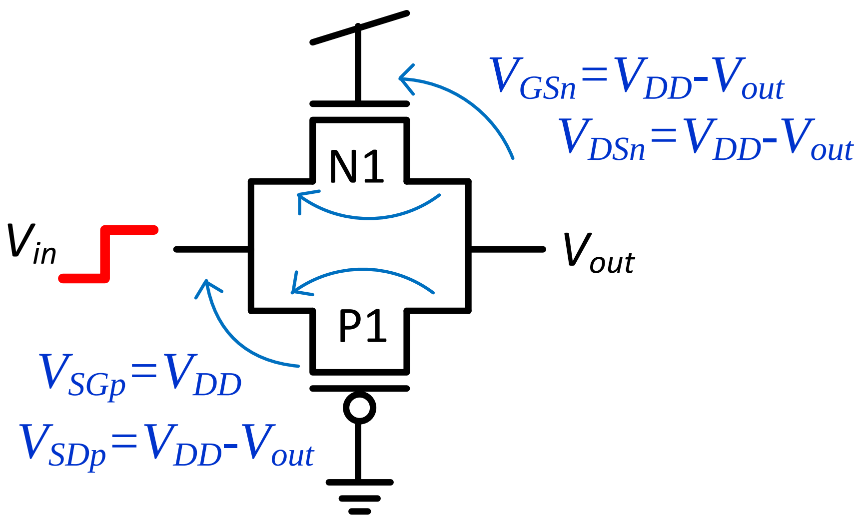
Transmission Gate Example

- During a transmission gate transition, both transistors are on during the operation.
- One transistor passes a “strong signal” with maximum overdrive, while the other passes a much weaker signal.
- Let’s take a ‘0’ to ‘1’ transition as an example:



Transmission Gate Example

- As usual, we will mark the sources and drains.



- At the beginning of the transition, $V_{out}=0$, so both transistors are strongly velocity saturated.
- But as the output is charged, the resistance of the nMOS rises, while the resistance of the pMOS stays relatively constant.

Transmission Gate Example

- At $t=0$:

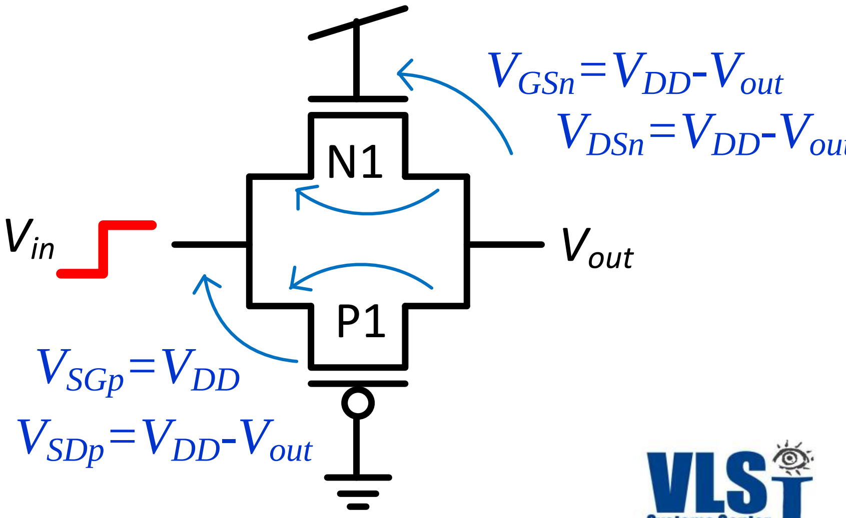
$$I_{out} = I_n + I_p = k_n\left(\left(V_{DD} - V_{Tn}\right)V_{DSat,n} - 0.5V_{DSat,n}^2\right) + k_p\left(\left(V_{DD} - V_{Tp}\right)V_{DSat,p} - 0.5V_{DSat,p}^2\right)$$

- At $t=t_{pd}$:

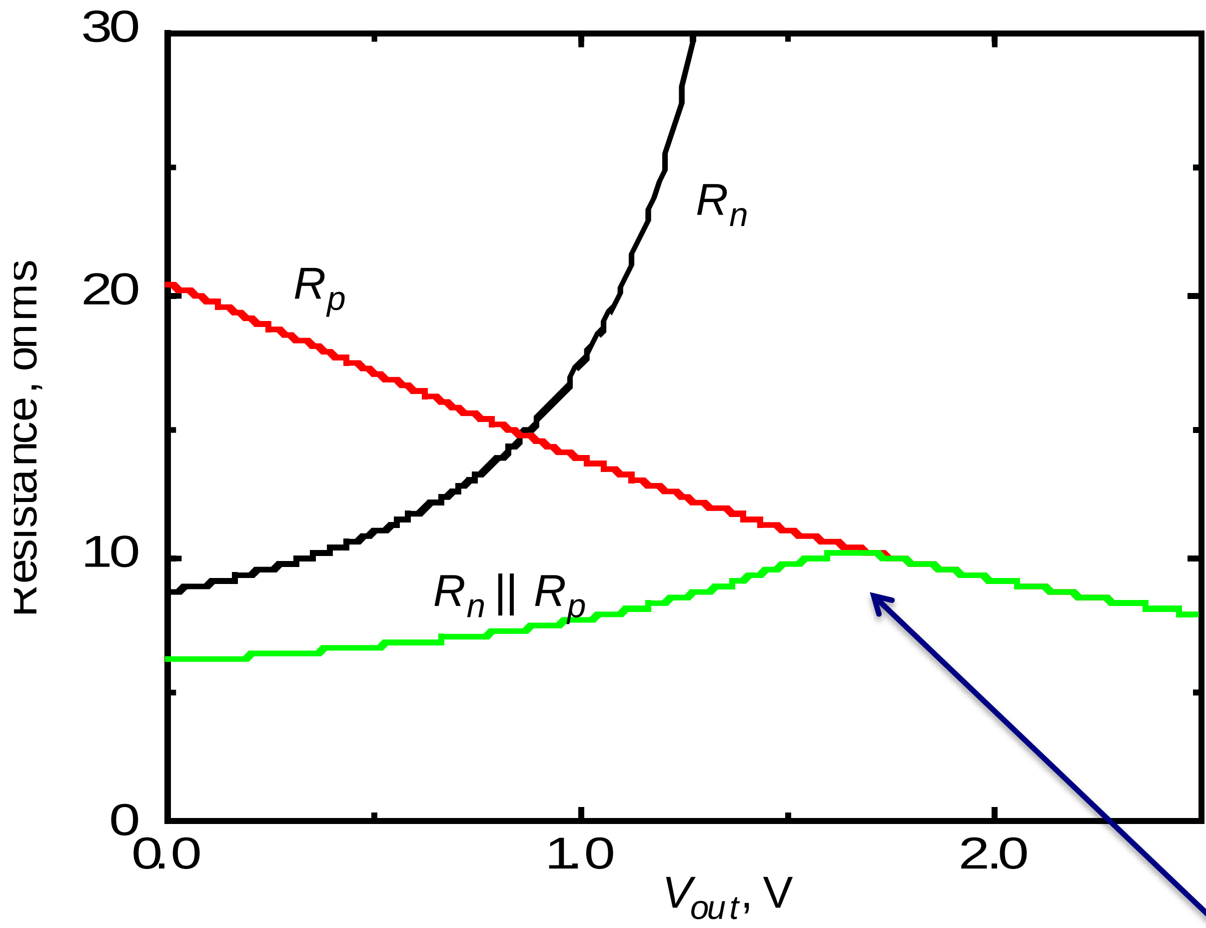
$$I_{out} = k_n\left(\frac{V_{DD}}{2} - V_{Tn}\right)^2 + k_p\left(\left(V_{DD} - V_{Tp}\right)V_{DSat,p} - 0.5V_{DSat,p}^2\right)$$

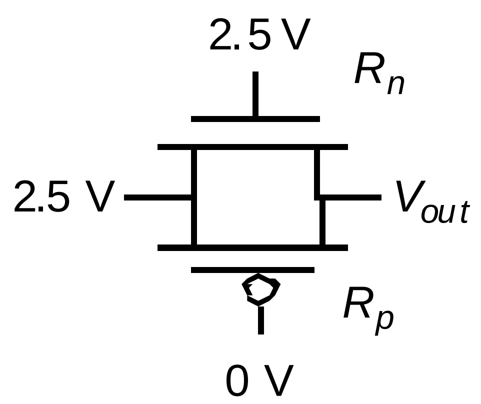
- At $V_{out}=V_{DD}-V_{Tn}$:

$$I_{out} = k_p\left(\left(V_{DD} - V_{Tp}\right)V_{DS} - 0.5V_{DS}^2\right)$$



Resistance of Transmission Gate





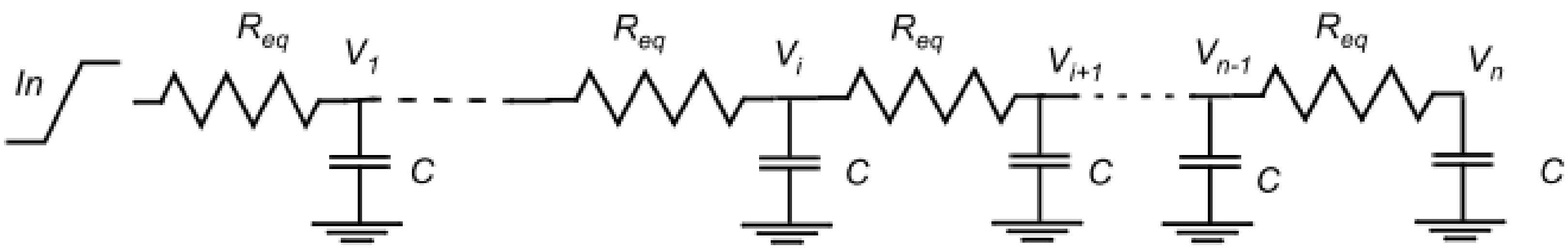
Almost constant resistance!

Delay of TG Chain

- An interesting question is what happens if we cascade several Transmission Gates in series.
- So assuming one gate gives $t_{pd}=0.69R_{eq}C_{dTG}$, we can draw the chain of gates as an RC chain.
- Given N gates and using the Elmore Delay, we get:

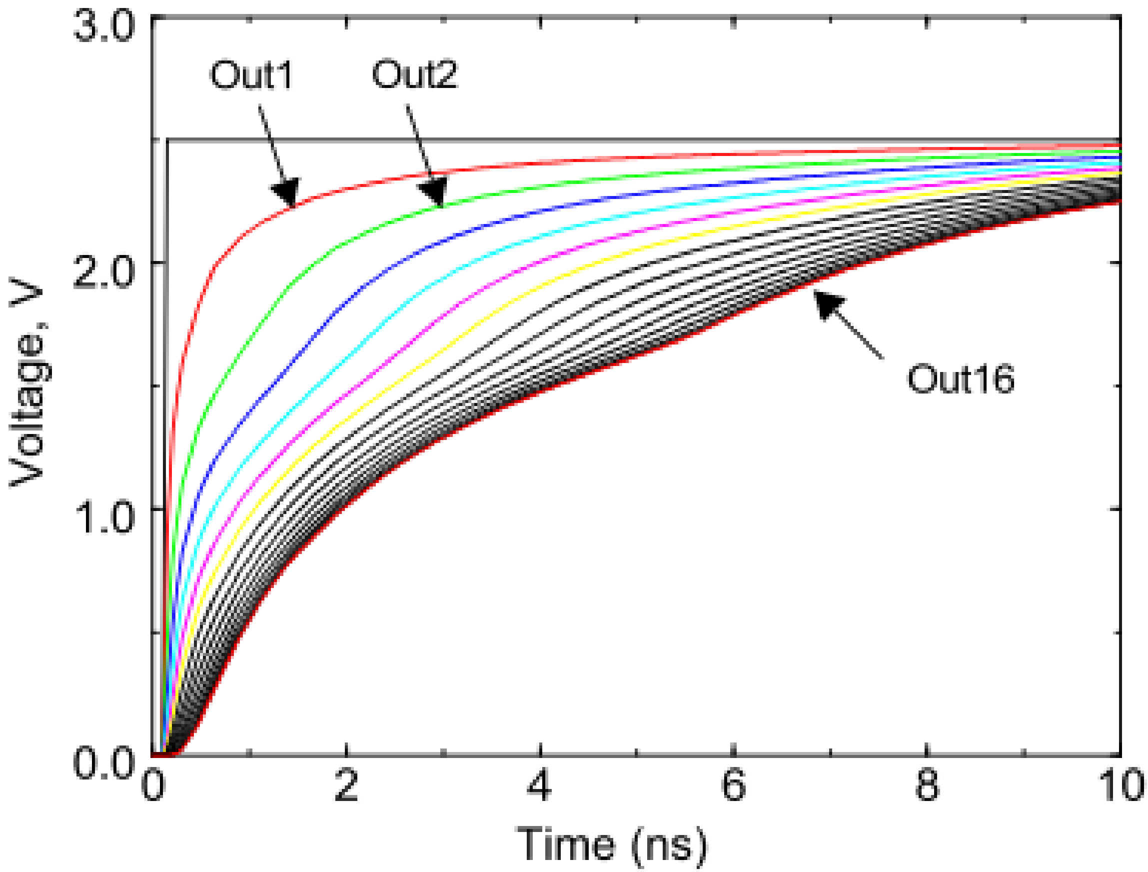
$$\tau_N = C_1 R_{eq} + C_2 \cdot 2R_{eq} + C_3 \cdot 3R_{eq} + \ldots + C_N \cdot NR_{eq}$$

$$= R_{eq} C_{d,TG}\left(1+2+..+N\right) = R_{eq} C_{d,TG}\left(\frac{N\left(N+1\right)}{2}\right)$$



Delay of TG Chain

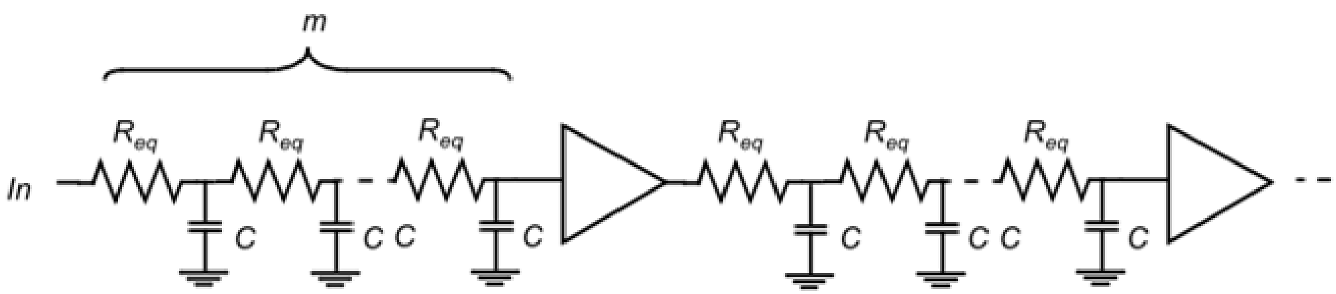
- Delay of 16 TGs comes out 2.7 ns (for 0.25um technology)
- The transition (rise time) is slow.



$$t_{pd} \propto N^2$$

Optimizing a TG Chain

- To optimize this problem, we will insert a buffer every *m* TGs.



- But what is the correct value of *m*?
- We already know how to optimize this type of problem…

$$\frac{\partial t_{buffered}}{\partial m} = 0$$

$$t_{buffered} = 0.69\left(\frac{N}{m}\right)R_{eq}C_{d,TG}\left(\frac{m(m+1)}{2}\right)+\left(\frac{N}{m}-1\right)t_{buf}$$

$$= 0.69R_{eq}C_{d,TG}\left(\frac{N(m+1)}{2}\right)+\left(\frac{N}{m}-1\right)t_{buf}$$

Optimizing a TG Chain

$$t_{buffered} = 0.69 R_{eq} C_{d,TG} \left(\frac{N(m+1)}{2} \right) + \left(\frac{N}{m} - 1 \right) t_{buf}$$

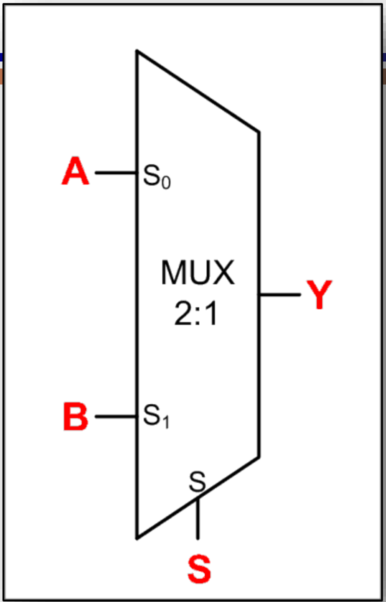
$$\frac{\partial t_{buffered}}{\partial m} = 0$$

$$m_{opt} = 1.7 \sqrt{\frac{t_{buf}}{C_{dTG} R_{eq}}} \approx 3$$

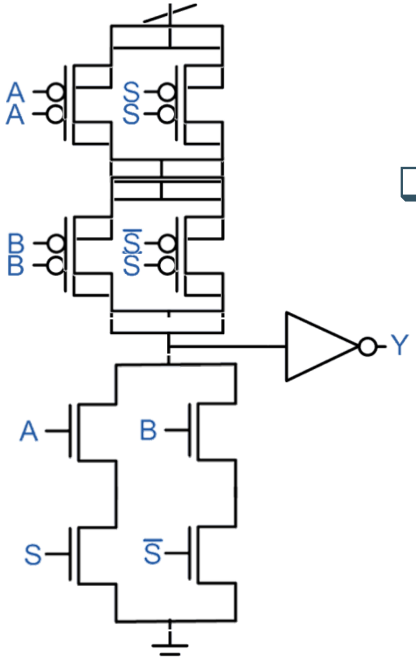
$$t_{pd} \propto N$$

2-Input MUX

- The ***2-input Multiplexer*** is a ***Universal gate*** that is very commonly used in digital circuits, especially for ***signal selection***.



$$F = A \cdot S + B \cdot \overline{S}$$

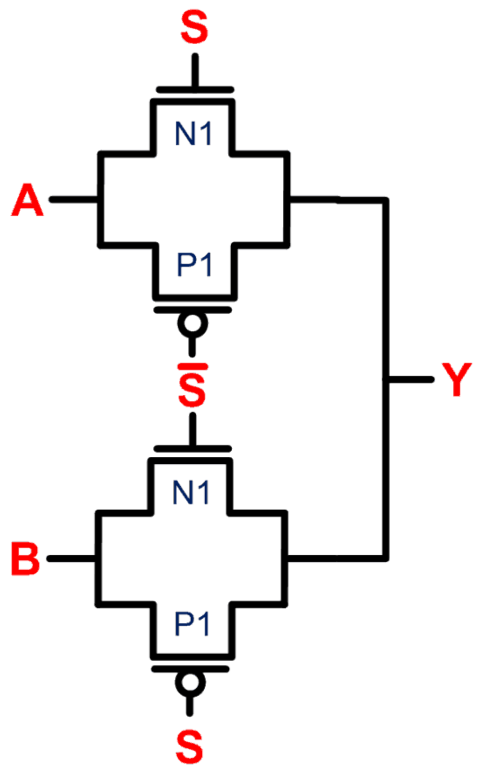


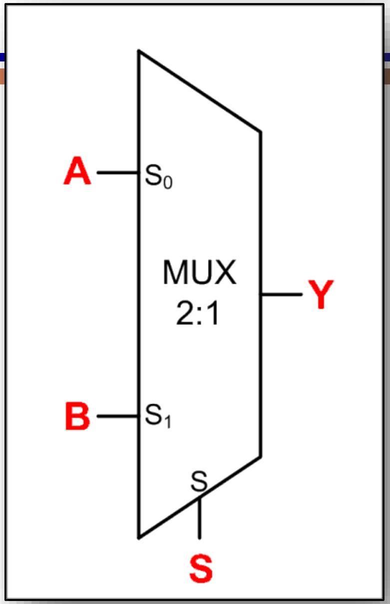
- Let's inspect its implementation in ***Standard CMOS***:
  - » ***PDN***: $\overline{F} = \overline{A \cdot S + B \cdot \overline{S}}$
  - » ***PUN***: $\overline{F} = \overline{A \cdot S + B \cdot \overline{S}} = \overline{A \cdot S} \cdot \overline{B \cdot \overline{S}} = \left(\overline{A} + \overline{S}\right) \cdot \left(\overline{B} + S\right)$

- This implementation requires ***10 or 12 transistors***:

2-Input MUX

- Using ***Transmission Gates***, we can make the same circuit with only ***6 transistors***:

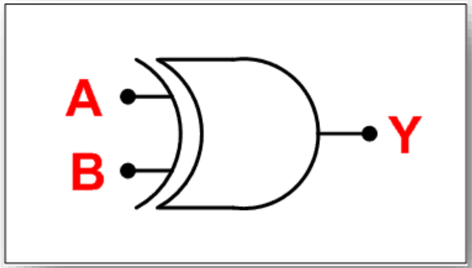




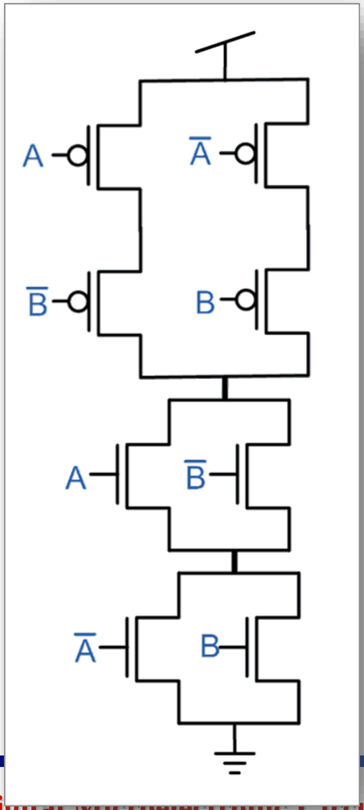
$$F = A \cdot S + B \cdot \overline{S}$$

2-Input XOR

- Another example of an efficient ***Transmission Gate*** is the ***XOR function***.
- This function is very useful, for instance in ***parity*** calculations.



$$F = A\cdot\overline{B} + \overline{A}\cdot B$$

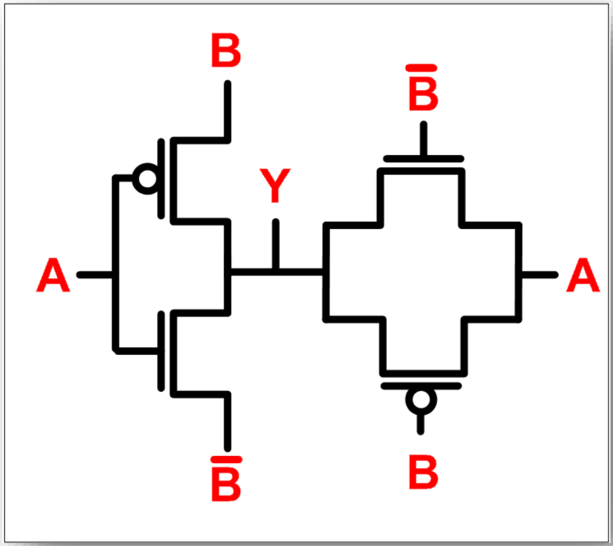


- With ***Standard CMOS***:
  - » ***PUN***: $F = A\cdot\overline{B} + \overline{A}\cdot B$
  - » ***PDN***: $F = \overline{\overline{A\cdot\overline{B} + \overline{A}\cdot B}} = \overline{\overline{A\cdot\overline{B}}\cdot\overline{\overline{A}\cdot B}} = \overline{\left(\overline{A}+B\right)\left(A+\overline{B}\right)}$

- Here we've reached a whopping ***12 transistors***!

2-Input XOR

- With ***Transmission Gates***, we can do it with only ***6***!
  - When ***B='1'***, the ***input stage*** is a ***CMOS inverter*** and the ***Transmission Gate*** is ***closed***. Hence: $Y = \bar{A} \cdot B$
  - When ***B='0'***, the ***input stage closes both transistors***, but the ***Transmission Gate*** is now ***open***, so we get: $Y = A \cdot \bar{B}$



- Together, we get our ***XOR function***:

$$Y = \bar{A} \cdot B + A \cdot \bar{B}$$

Last Lecture

- Pass Transistor Logic

Last Lecture

- Transmission Gates

9.4

9.1 Pass Transistor Logic

9.2 Extending the PTL Concept

9.3 Transmission Gates

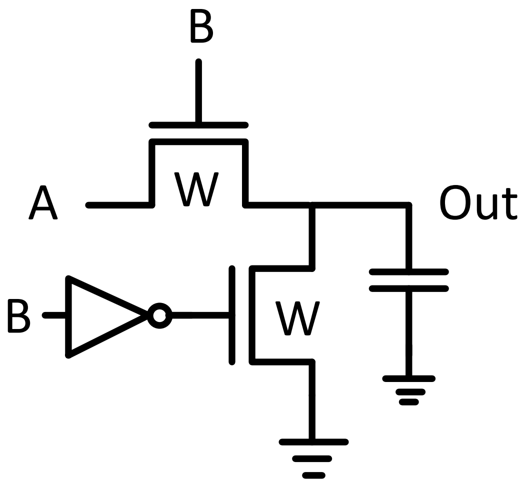
9.4 PTL Logical Effort

Okay, now let's go way beyond and figure out

PTL LOGICAL EFFORT

PTL Logical Effort

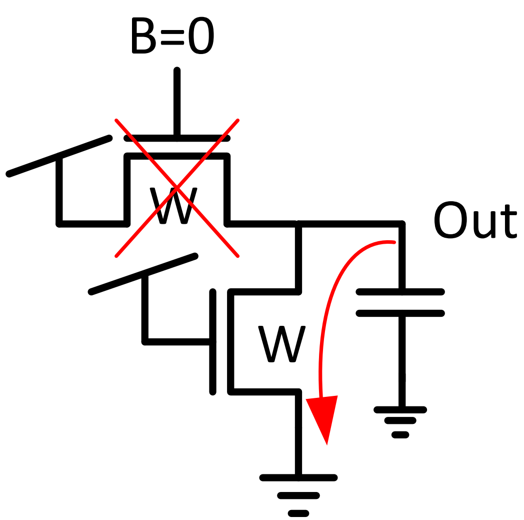
- How do we go about calculating the LE of PTL?
  - Let's take a PTL AND gate.
  - We will arbitrarily size the gate with minimum transistors for calculation.
  - Now we need to differentiate between the various inputs, transitions, and also recognize what makes up the entire circuit.



- Essentially, we have to recognize that:
  - Input *A* is driven through a Buffer.
  - Input B drives a gate.
    B! is a different signal on a different path.

PTL Logical Effort

- So let's start with input *B* (with *A='1'*):
  - When B='1'→'0' we get:
  - The output discharges through the nMOS, so:



$$R_{eq} = R_{\min}$$

$$C_g(B) = C_{g\min}$$

$$C_d = 2C_{d\min}$$

$$p = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{d,gate}}{3C_{d,\min}} = 1 \cdot \frac{2}{3} = \frac{2}{3}$$

$$LE = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{g,gate}}{3C_{g,\min}} = 1 \cdot \frac{1}{3} = \frac{1}{3}$$

  - It looks as if the PTL gate is a great driver!

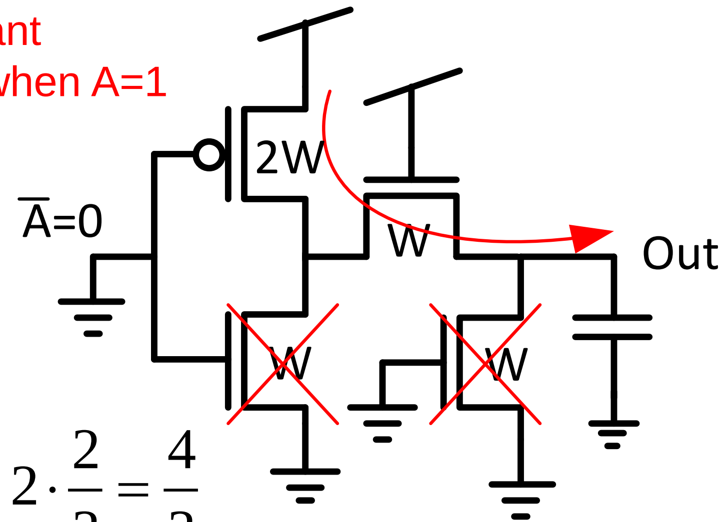
- But that was only one of numerous transitions…

PTL Logical Effort

- Now when B='0'→'1':

  The only relevant transition is when A=1

  - The output charges through the series connection of the buffer's pMOS and the PTL nMOS:



$$R_{eq} = 0.5R_p + R_n = 2R_{\min}$$

$$C_g(B) = C_{g\min}$$

$$C_d = 2C_{d\min}$$

* The buffer's output was already charged.

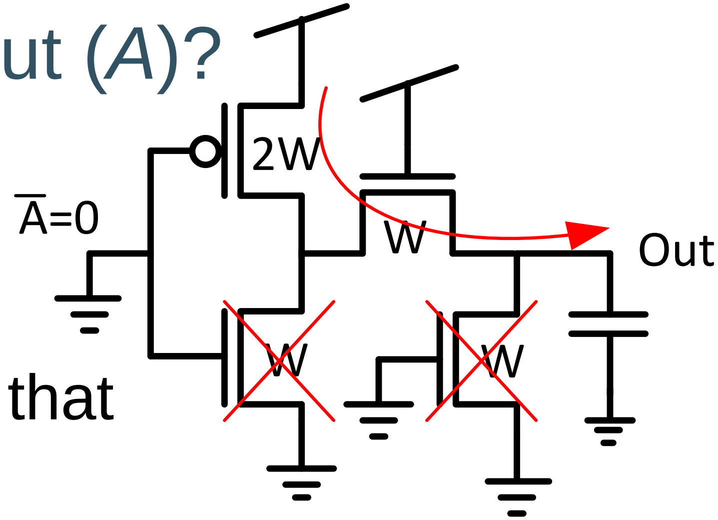
$$p = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{d,gate}}{C_{d,\min}} = 2 \cdot \frac{2}{3} = \frac{4}{3}$$

$$LE = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{g,gate}}{C_{g,\min}} = 2 \cdot \frac{1}{3} = \frac{2}{3}$$

  - So driving a PTL through the gate input (B) is pretty good!

PTL Logical Effort

- But what about the diffusion input (*A*)?
  - » When B='1' and A='0'→'1' we have the same model, but now the input is *A*.
  - » Therefore the gate capacitance is that of an inverter = 3W.



  - » Plus, the buffer's capacitance is initially discharged.

$$R_{eq} = 0.5R_p + R_n = 2R_{\min}$$

$$C_g(A) = C_{g,inv} = 3C_{g\min}$$

$$C_d = C_{d,inv} + C_{out} = (3+2)C_{d\min}$$

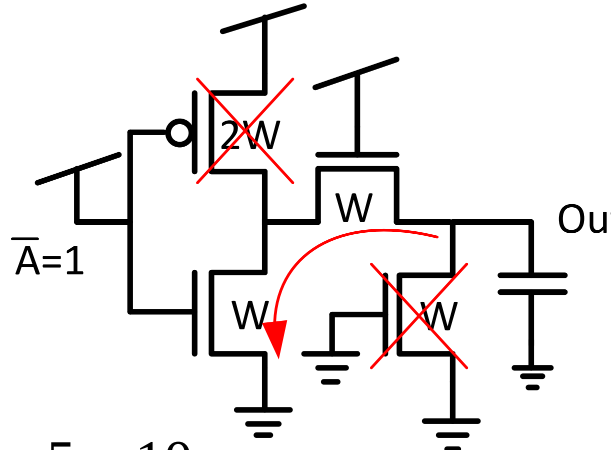
$$p = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{d,gate}}{C_{d,\min}} = 2 \cdot \frac{5}{3} = {}^{10}\!/_{3}$$

$$LE = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{g,gate}}{C_{g,\min}} = 2 \cdot \frac{3}{3} = 2$$

  - » So we get really bad performance.

PTL Logical Effort

- The opposite transition is similar:
  - Now A='1'→'0'.



$$R_{eq} = R_n + R_n = 2R_{\min}$$

$$C_g(A) = C_{g,inv} = 3C_{g\min}$$

$$C_d = C_{d,inv} + C_{out} = (3+2)C_{d\min}$$

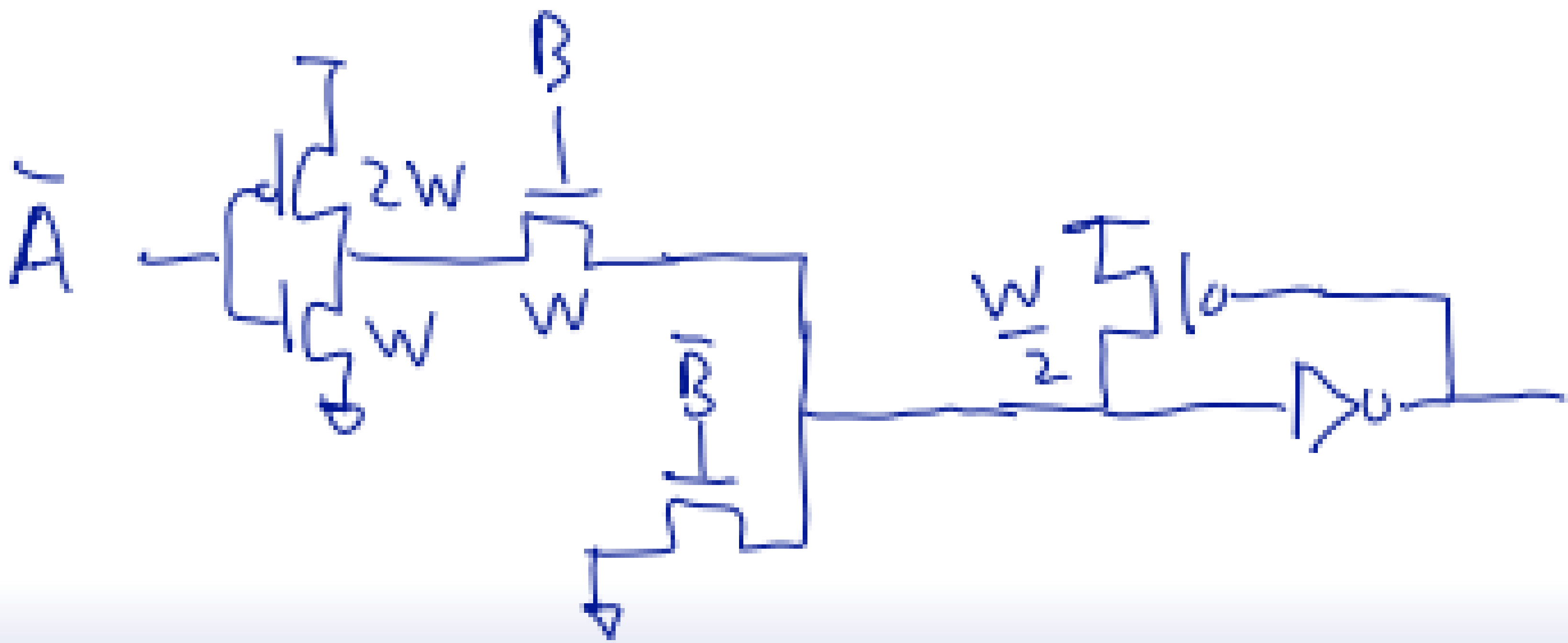
$$p = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{d,gate}}{C_{d,\min}} = 2 \cdot \frac{5}{3} = \frac{10}{3}$$

$$LE = \frac{R_{gate}}{R_{inv}} \cdot \frac{C_{g,gate}}{C_{g,\min}} = 2 \cdot \frac{3}{3} = 2$$

- So, using a PTL gate through the diffusions is really bad.

PTL Logical Effort

With level restore:



Summary

- Pass transistor logic is a low transistor count CMOS alternative, but:
  - » It is non-digital, so every few stages we must insert a CMOS gate.
  - » It suffers from depleted high levels, so we should consider using a level-restorer.
  - » It is very asymmetric, so we should carefully analyze each path before using it.
- However, the concept of a pass transistor can be very useful:
  - » We can build special gates (transmission gate, XOR, MUX).
  - » We can use it as a switch.
  - » We can build interesting logic families (CPL, GDI, etc.)