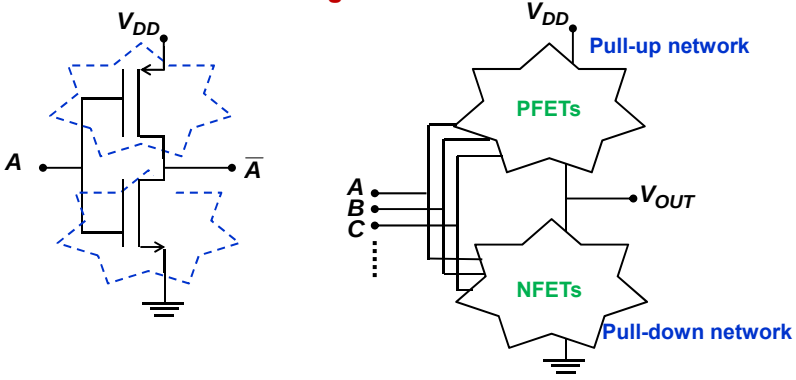
Lecture 24

CMOS Logic Gates and Digital VLSI – II

In this lecture you will learn:

- Static CMOS Logic Gates
- FET Scaling
- CMOS Memory, SRAM and DRAM
- CMOS Latches, and Registers (Flip-Flops)
- Clocked CMOS

CMOS Logic: General Architecture



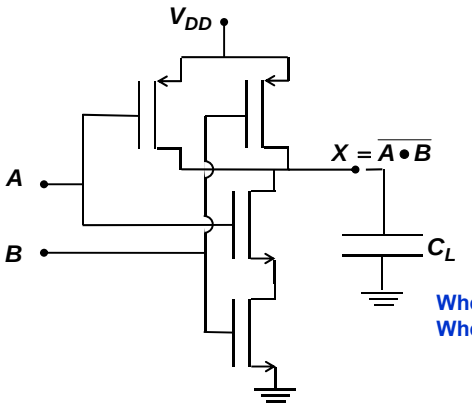
Pull-up network:

Consists of only PFETs
Connects the output node to V_{DD}
Charges the output to HIGH

Pull-down network:

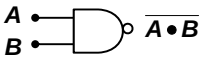
Consists of only NFETs
Connects the output node to ground
Discharges the output to LOW

CMOS NAND Gate

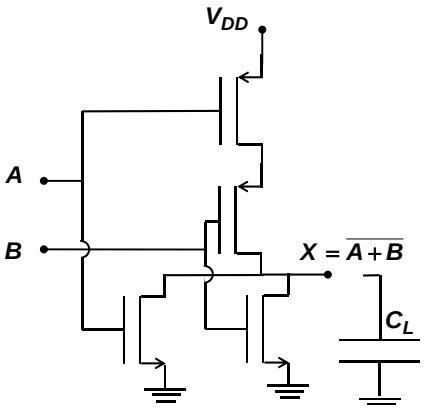


A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

When both A and B are HIGH, output is LOW
When either A or B is LOW, output is HIGH

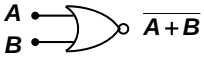


CMOS NOR Gate

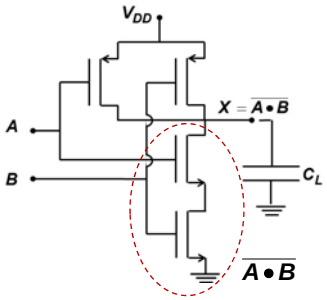


A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

When both A and B are LOW, output is HIGH
When either A or B is HIGH, output is LOW

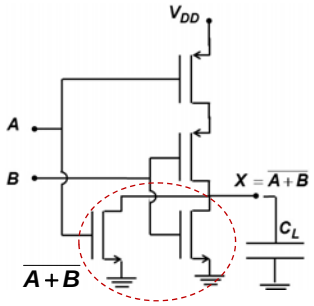


CMOS Gates: Pull Down Network Design



If A and B are both HIGH, output will be LOW

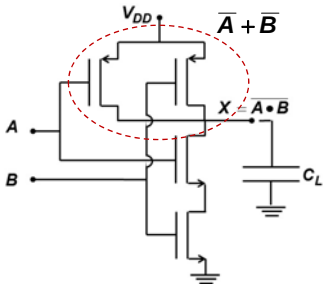
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



If either A or B is HIGH, output will be LOW

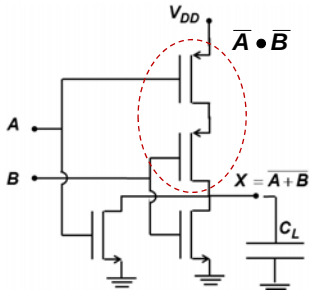
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

CMOS Gates: Pull Up Network Design



If either A or B is LOW, output will be HIGH

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

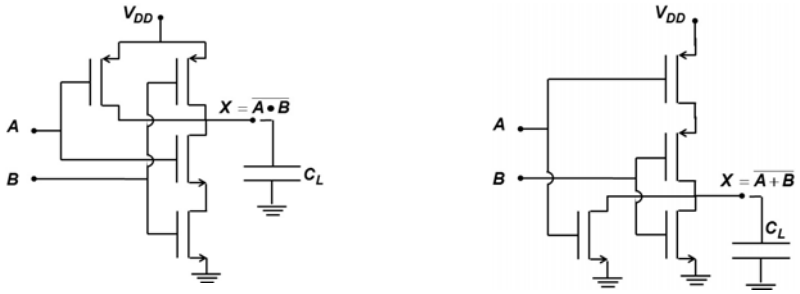


If A and B are both LOW, output will be HIGH

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

$\overline{A \bullet B} = \overline{A} + \overline{B}$ ⟵ De Morgan's Law ⟶ $\overline{A + B} = \overline{A} \bullet \overline{B}$

CMOS Gates: Pull Up and Pull Down Network Design



Pull up and pull down networks are dual of each other!

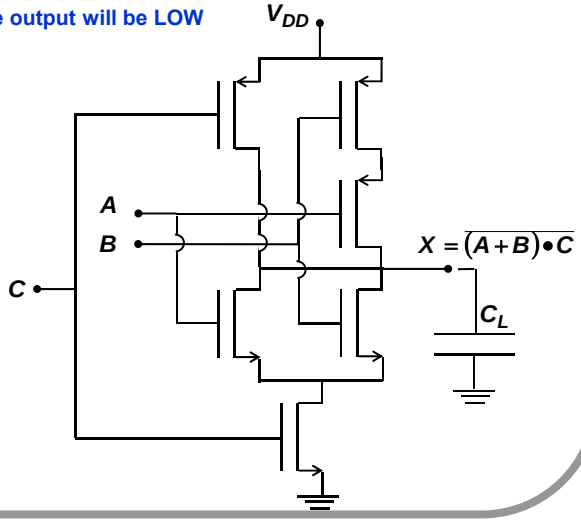
CMOS Gates: More Complex Logic Gates

Suppose we need to design a logic gate for:

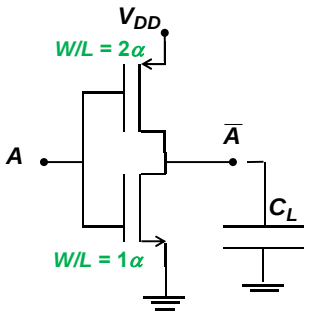
$$X = \overline{(A + B) \bullet C}$$

If (A or B) and C are HIGH, the output will be LOW

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



FET Scaling – Inverter



The mobility of electrons in NFETs is generally almost twice that of the holes in PFETs

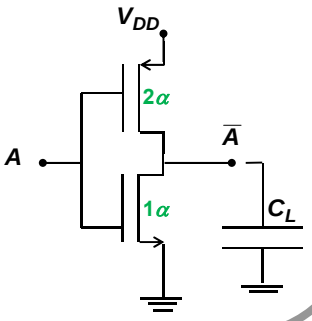
One would want the current drives for charging and discharging the output and, consequently, the rise and fall times for the output to be identical (i.e. one would want the NFETs and the PFETs to have the same current drives)

$$t_r \approx \frac{2C_L}{k_p(V_{DD}+V_{TP})}\left\{\frac{-V_{TP}}{V_{DD}+V_{TP}}-\frac{1}{2}\ln\left[\frac{0.1V_{DD}}{2(V_{DD}+V_{TP})-0.1V_{DD}}\right]\right\}$$

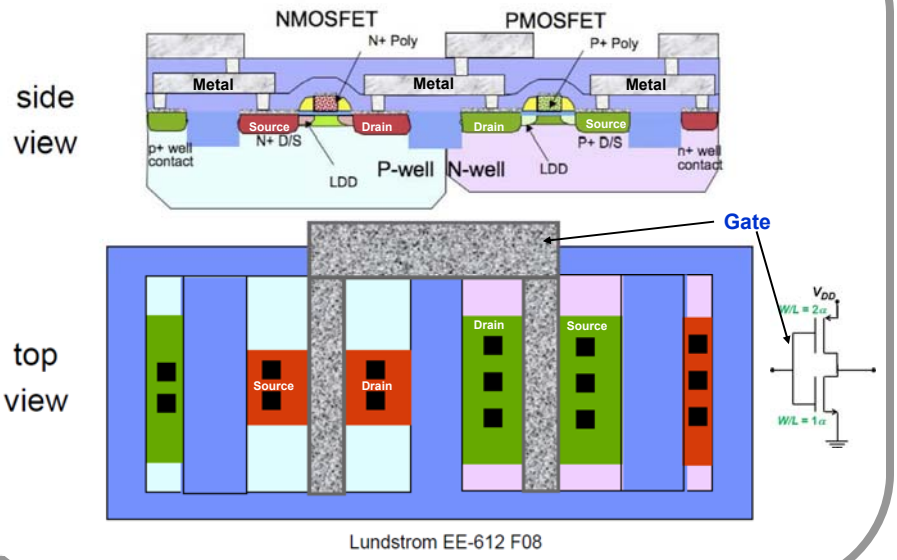
$$t_f \approx \frac{2C_L}{k_n(V_{DD}-V_{TN})}\left\{\frac{V_{TN}}{V_{DD}-V_{TN}}-\frac{1}{2}\ln\left[\frac{0.1V_{DD}}{2(V_{DD}-V_{TN})-0.1V_{DD}}\right]\right\}$$

Need $k_n = k_p$

Therefore the *W/L* ratio of the PFET is chosen to be twice that of the NFET in an inverter

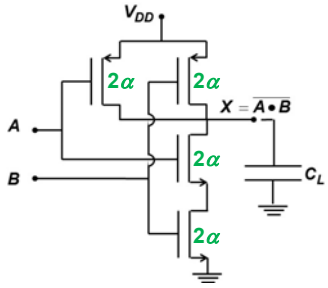


Fabricated FET Inverter: Dual Well CMOS Technology



Lundstrom EE-612 F08

FET Scaling – NAND Gate



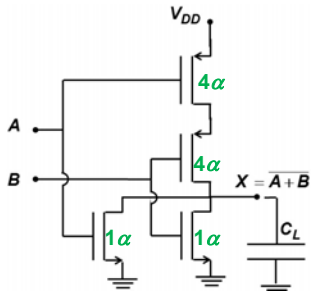
As in the inverter case, one scales the PFETs by 2α

A HIGH output has to be discharged through the two NFETs in series

Two FETs in series, with the same gate voltage, are like one FET that is twice as long

Therefore, in order to keep the same current drive in discharging a HIGH output in the NAND gate as in the simple inverter, one needs to scale the NFETs by 2α each

FET Scaling – NOR Gate



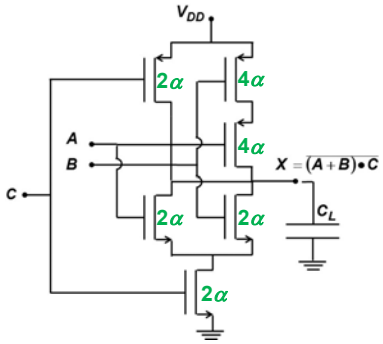
As in the inverter case, one scales the NFETs by 1α

A LOW output has to be charged through the two PFETs in series

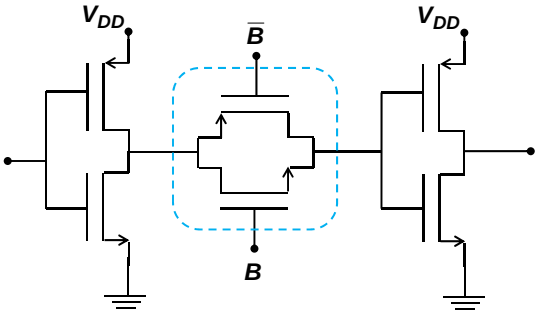
Two FETs in series, with the same gate voltage, are like one FET that is twice as long

Therefore, in order to keep the same current drive in charging a LOW output in the NOR gate as in the simple inverter, one needs to scale the PFETs by 4α each

FET Scaling – A More Complex Example

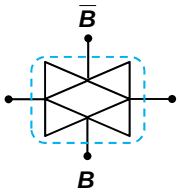


The Transmission Gate



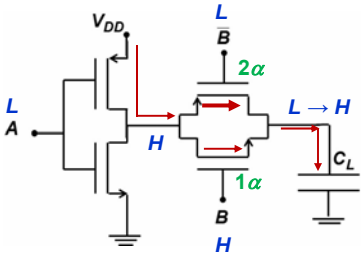
A transmission gate allows the logical value to pass from the input to the output only if the gate is OPEN (meaning the control signal B is HIGH)

If the gate is closed (meaning the control signal B is LOW) the input and the output are disconnected from each other



The Transmission Gate

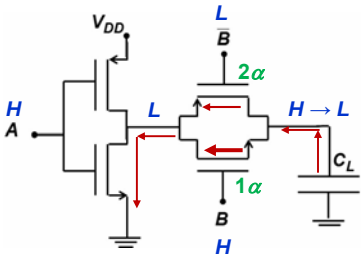
Case I: Input is sitting at HIGH, the output is sitting at LOW, and the gate opens



Although both the NFET and the PFET will pass the current, the NFET will cut-off when the output node is still V_{TN} below logical HIGH value (~V_{DD})

So the PFET is required to charge the output to the HIGH value (~V_{DD})

Case II: Input is sitting at LOW, the output is sitting at HIGH, and the gate opens

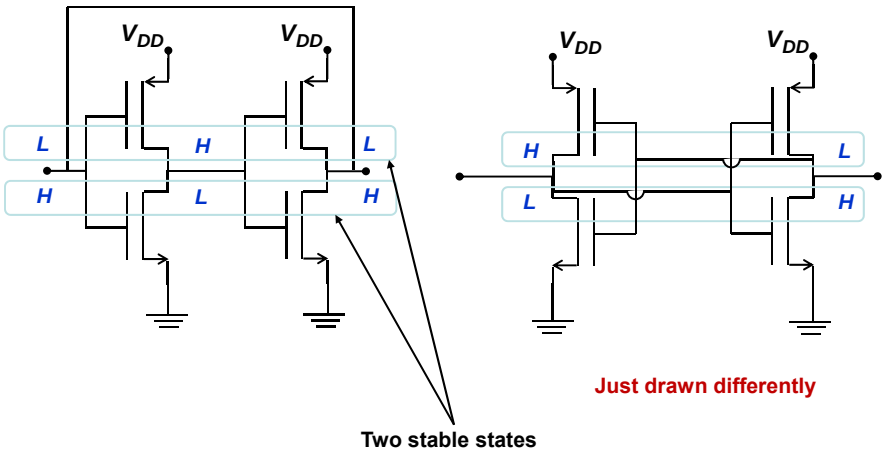


Although both the NFET and the PFET will pass the current, the PFET will cut-off when the output node is still $-V_{TP}$ above the logical LOW value (~0)

So the NFET is required to discharge the output to the LOW value (~0)

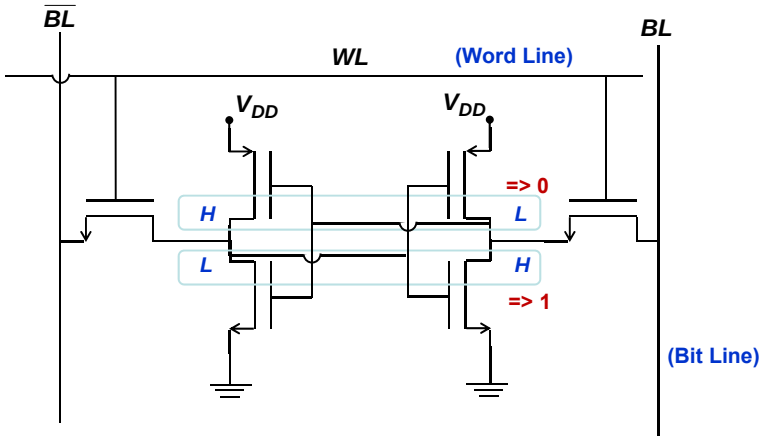
CMOS Memory Element

Two inverters can be used to realize a bistable memory element:



Static Random Access Memory (SRAM)

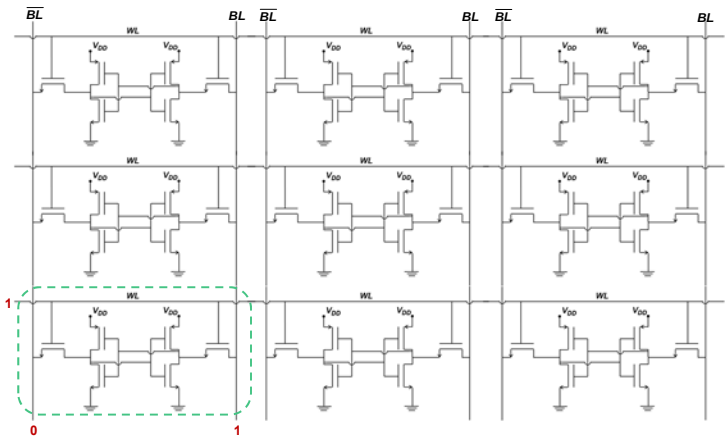
Two inverters can be used together with NFET gates to realize a 6 FET SRAM cell



SRAM is fast

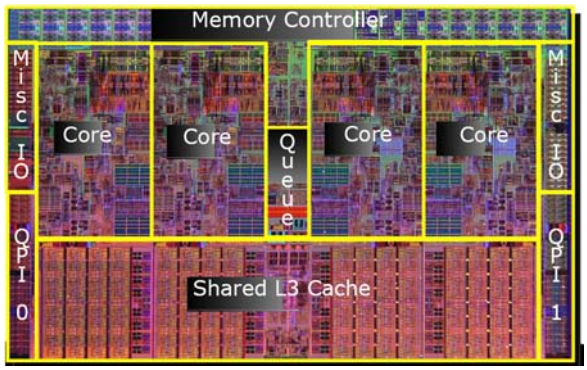
Used for implementing fast caches in microprocessors or fast memories in electronic instruments

Static Random Access Memory (SRAM)

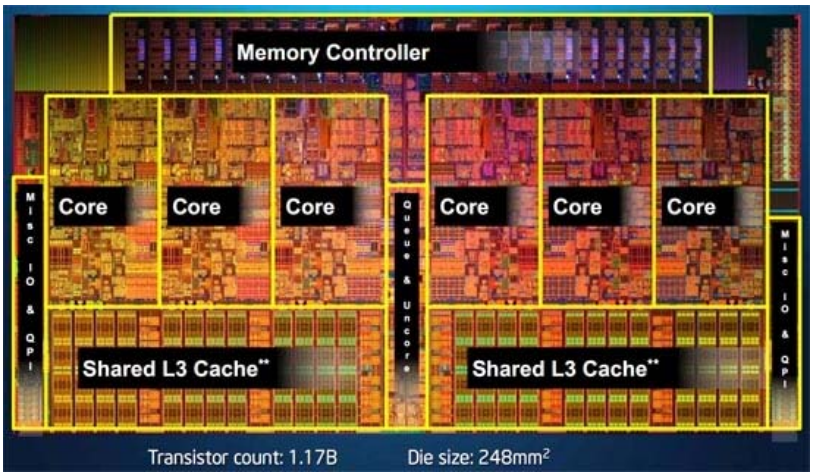


The indicated values show the voltages when a logical 1 is being written in the cell shown

Intel Core-i7 (4 Core) with 8MB L3 Cache (SRAM)



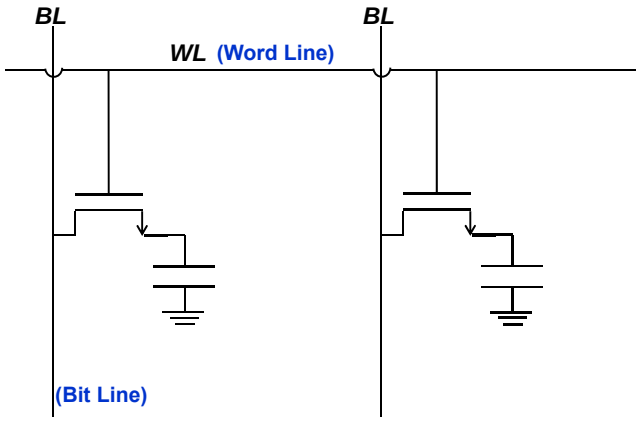
Intel Core-i7 (6 Core) with 12MB L3 Cache (SRAM)



1.17 Billion FETs

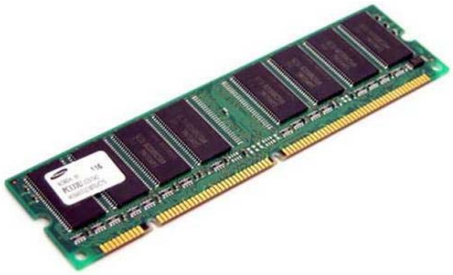
Dynamic Random Access Memory (DRAM)

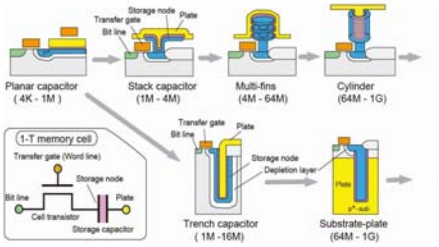
A DRAM cell can be implemented using just one FET and one capacitor



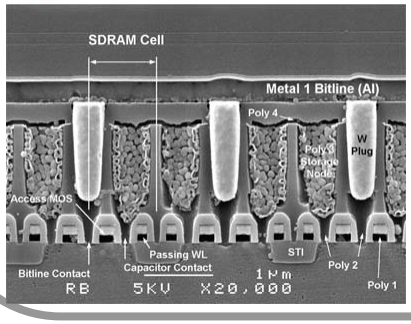
DRAM is much slower than SRAM
Capacitor can discharge via leakage currents (needs periodic refresh)
Reading a stored bit destroys the stored bit – every read must be followed by a write
Used for implementing large memory in computers

Dynamic Random Access Memory (DRAM)





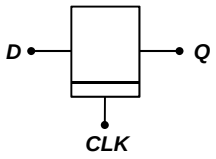
Types of DRAM capacitors

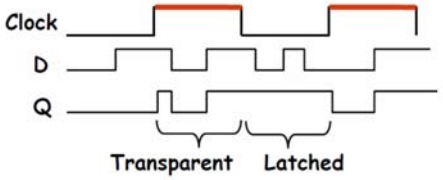


The DRAM cell (0.5 μm^2) uses stacked cylindrical capacitors with hemispherical silicon grains (HSG) in a capacitor (SAMSUNG)

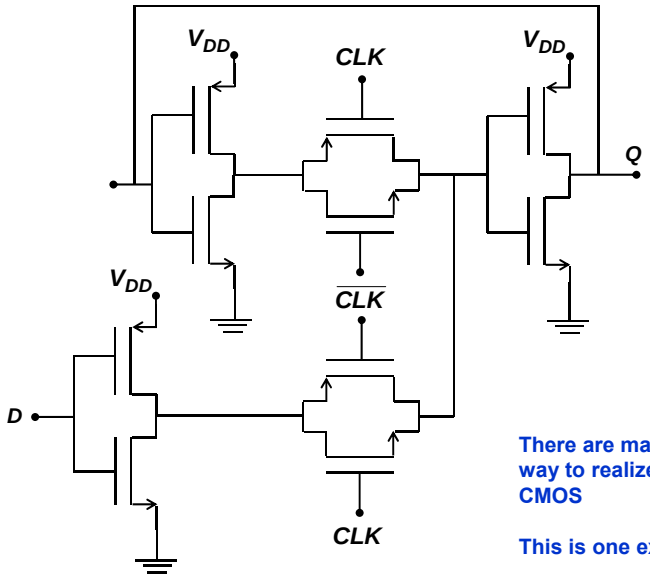
CMOS Latch

- Data passes through from the input (*D*) to the output (*Q*) when the *CLK* is HIGH (i.e. the latch is transparent)
- Data at the output (*Q*) is latched (and held in place) when the *CLK* is LOW





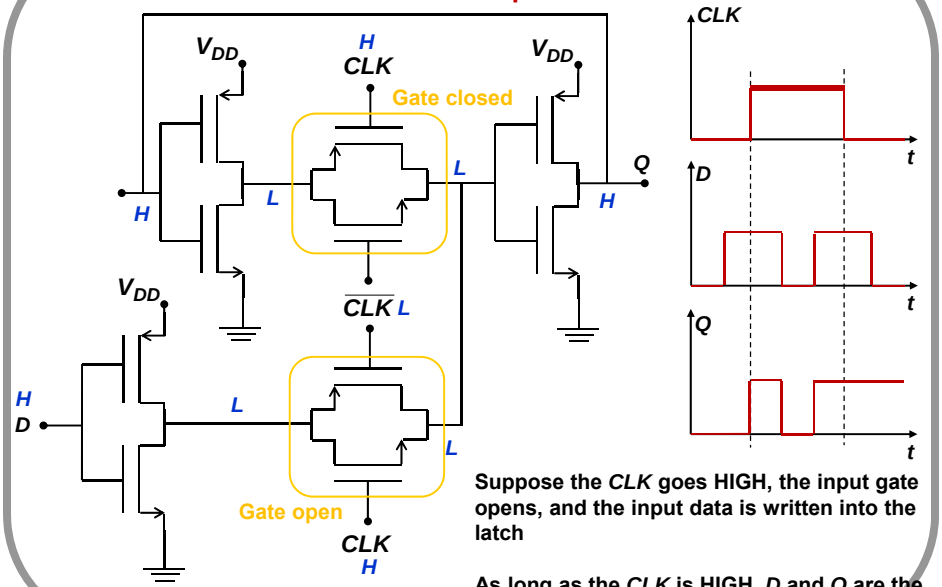
CMOS Latch: One Example



There are many different way to realize a latch in CMOS

This is one example…

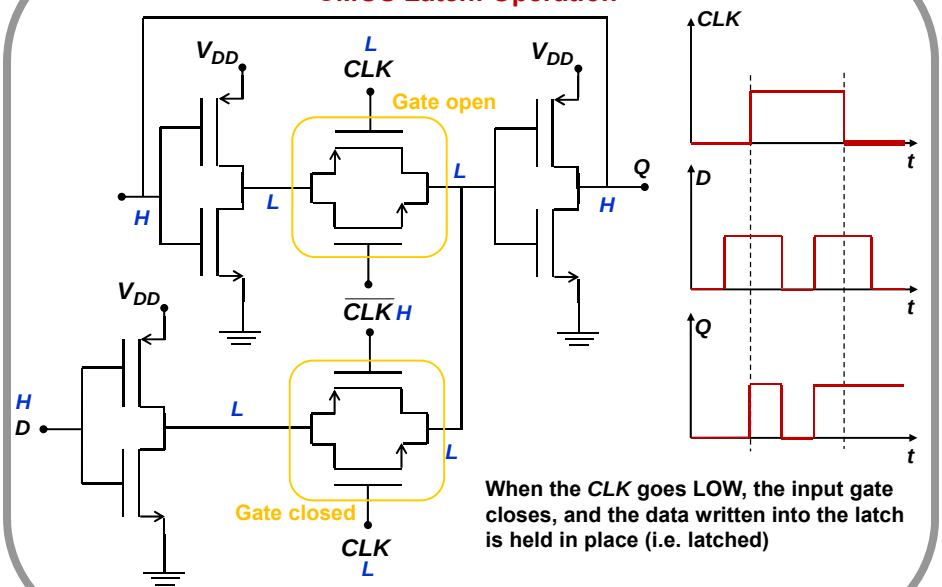
CMOS Latch: Operation



Suppose the *CLK* goes HIGH, the input gate opens, and the input data is written into the latch

As long as the *CLK* is HIGH, *D* and *Q* are the same (i.e. the latch is transparent)

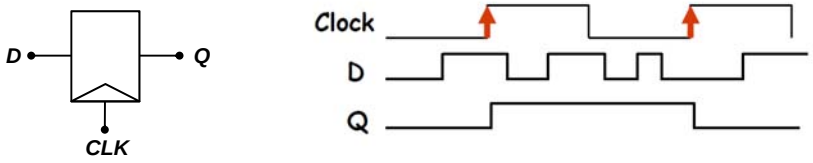
CMOS Latch: Operation



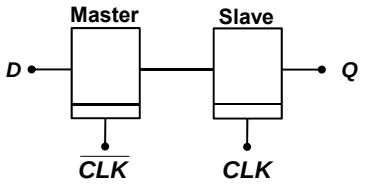
When the *CLK* goes LOW, the input gate closes, and the data written into the latch is held in place (i.e. latched)

CMOS Data Register or a Flip-Flip

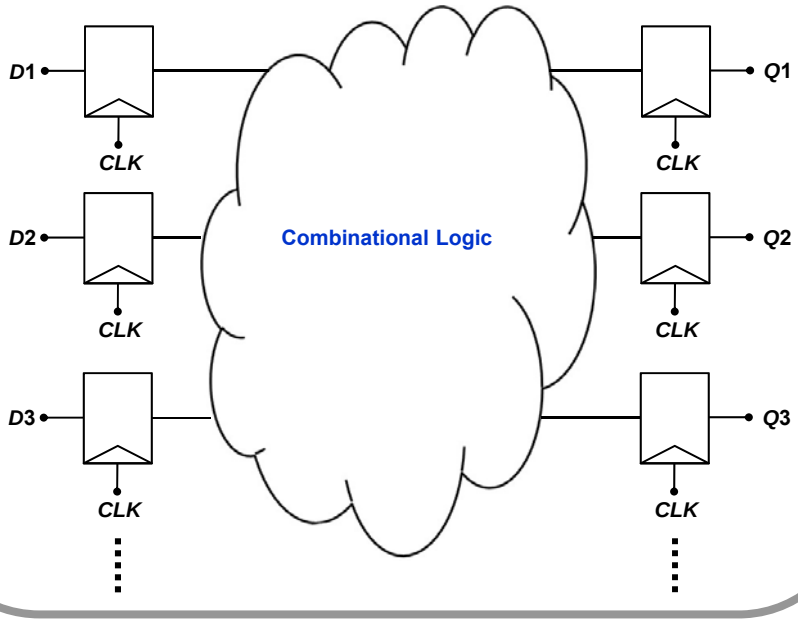
- When the *CLK* goes from LOW to HIGH, input data (*D*) is transferred to the output (*Q*) and held in place



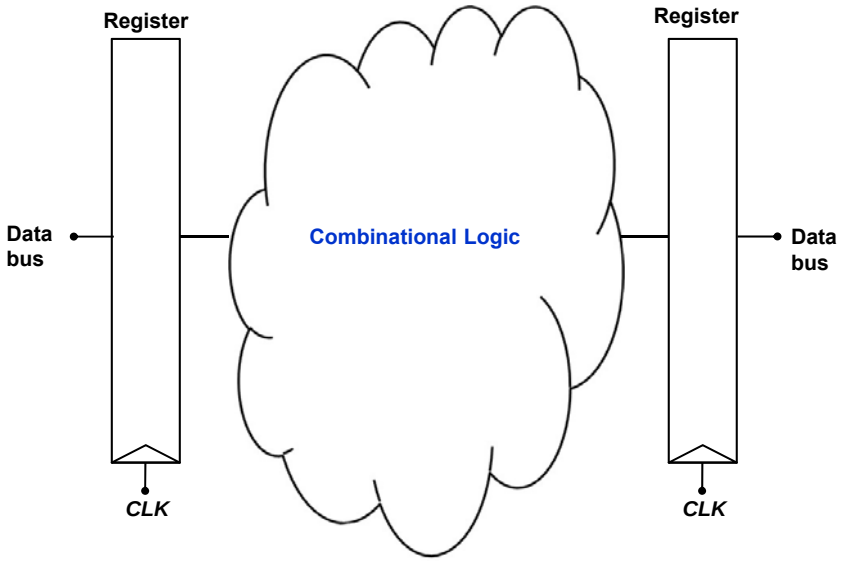
- A data register or a flip-flop can be realized by using two master/slave latches



Clocked CMOS Design



Clocked CMOS Design



Pipelined CMOS Digital Design