Transistor Logic Families

Digital Logic Gates

- We began the course by introducing digital logic gates.
- We then saw how these basic building blocks are used to construct various digital modules and how these modules can be combined to create digital systems using schematics and hardware description languages.
- In the next three lectures we will examine digital logic gates again, but from an **electronic physical circuit** perspective.

Logic Families

- The first logic families consisted of vacuum tubes and mechanical relays.
- We will examine the transistor logic families
  - RTL – Resistor Transistor Logic
  - DTL – Diode Transistor Logic
  - TTL – Transistor Transistor Logic
  - ECL – Emitter Coupled Logic
  - MOS Logic – Metal Oxide Semiconductor Logic
  - CMOS Logic – Complementary MOS Logic

Switching Circuits

- Transistors are used to implement logic gates because of their usefulness as switching circuits.
- We will examine how the transistor logic families are used to implement the switching circuits, in particular corresponding to the NAND and NOR logic gates.
- These basic circuits are the primary building blocks from which all other more complex digital components are obtained.

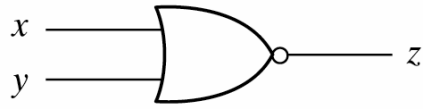
Positive/Negative Logic

- The ideal switching circuits are approximately realised using transistor switches.
- The binary symbols 1 and 0 are represented by voltages.
- In **positive logic** a high voltage represents a 1 and a low voltage represents a 0.
- In **negative logic** a high voltage represents a 0 and a low voltage represents a 1.

Inputs		Output
x	y	z
L	L	H
L	H	H
H	L	H
H	H	L

Positive Logic NAND Gate

Inputs		Output
x	y	z
L	L	H
L	H	L
H	L	L
H	H	L



Positive Logic NOR Gate

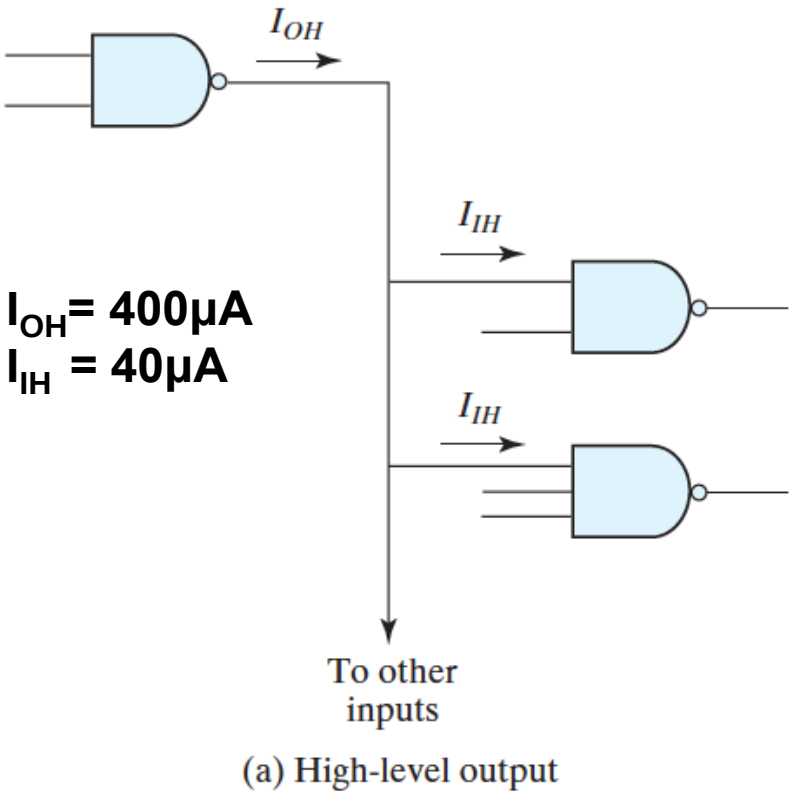
Special Characteristics

- The characteristics of digital logic families are usually compared by analyzing the circuit of the basic gate in each family.
- The most important parameters that are evaluated and compared are:
  - Fan-Out
  - Power Dissipations
  - Propagation Delay
  - Noise Margin

Fan-Out

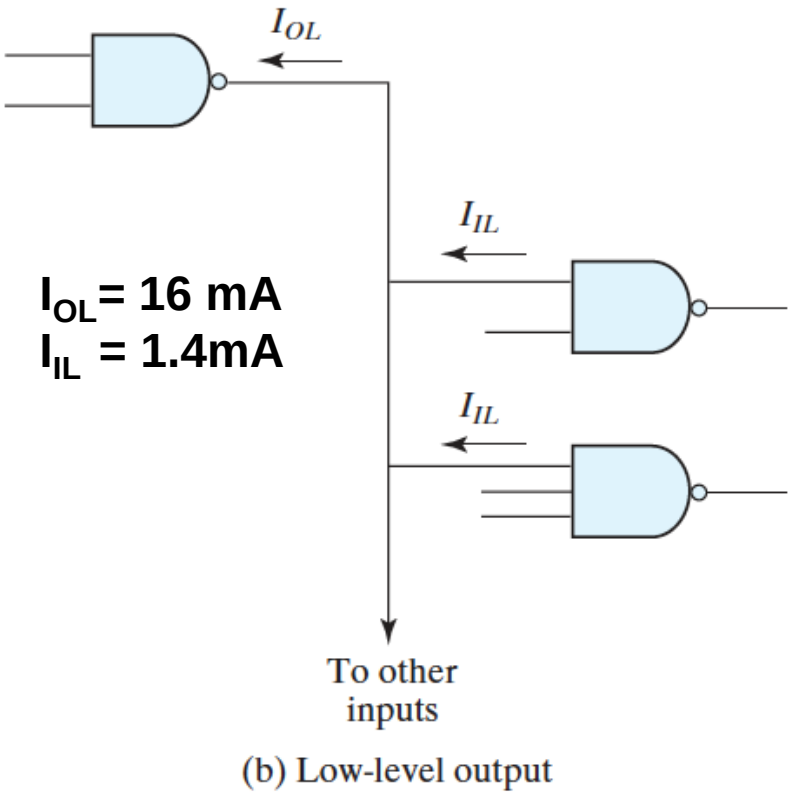
- The fan-out of a gate specifies **the number of standard loads** that can be **connected to the output of the gate** without degrading its normal operation.
- It is calculated from the amount of current available in the output of a gate and the amount of current needed in each input of a gate, namely, the ratio I_{OH}/I_{IH} (high-level operation) or I_{OL}/I_{IL},(low-level operation), whichever is smaller.

Fan-Out



(a) High-level output

I_{OH}= 400μA
I_{IH} = 40μA



(b) Low-level output

I_{OL}= 16 mA
I_{IL} = 1.4mA

Fan-out = 10

Power Dissipation

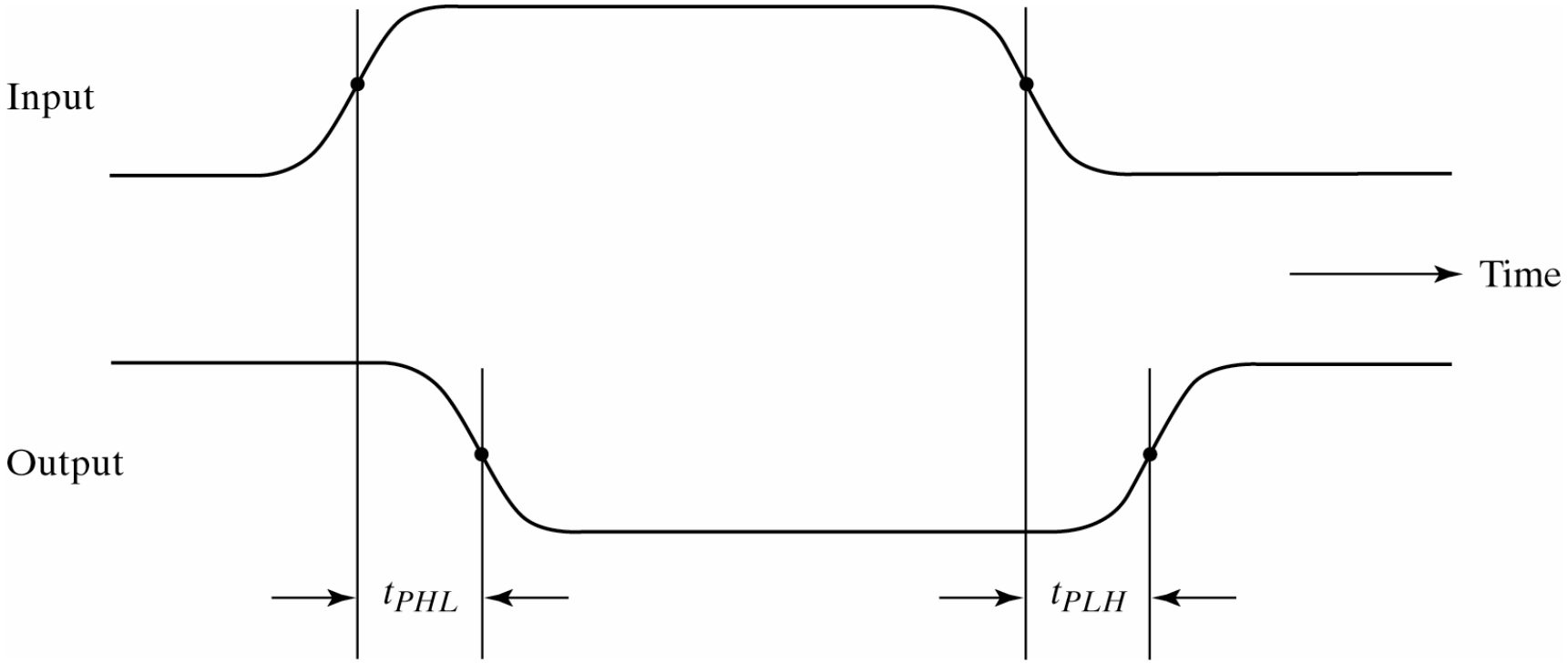
- The power dissipation is a parameter expressed normally in milliwatts (mW) and represents the amount of power needed by the gate.
- The amount of power that is dissipated in a gate is calculated from the supply voltage Vcc and the average current of the outputs in the high-voltage level I_{CCH} and in the low-voltage level I_{CCL}.

$$I_{CC}(avg) = \frac{I_{CCH} + I_{CCL}}{2}$$

$$P_D(avg) = I_{CC}(avg)' V_{CC}$$

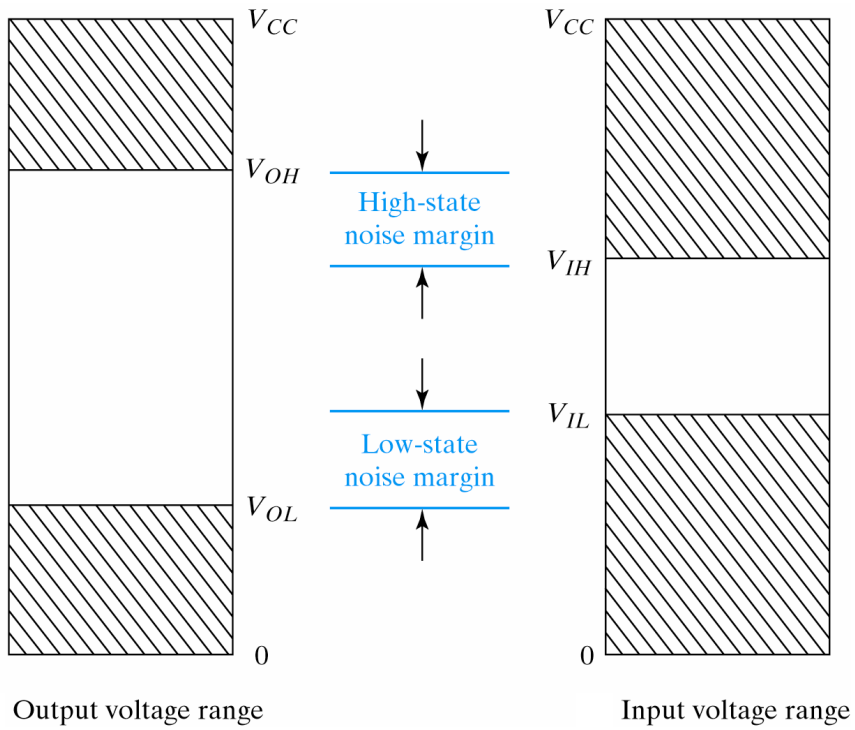
Propagation Delay

- The propagation delay is the average time it takes for an input signal to propagate form the input to the output.
- This delay limits the maximum clocking speed a circuit can achieve.



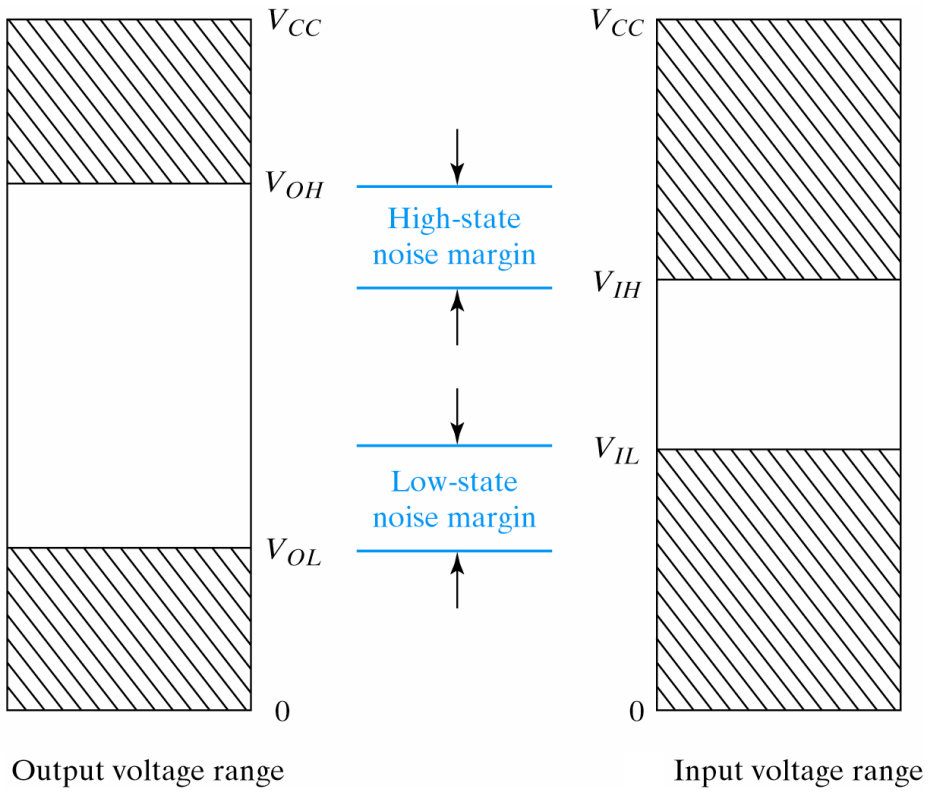
Noise Margin

- Logic circuits must be robust to the electrical noise caused by spurious signals.
- Each gate has a valid operating range for the low and high voltage levels
- e.g., the gate output voltage is low when it lies between 0 and V_{OL} volts. The gate output voltage is high when it lies between V_{CC} and V_{OH} volts.



Noise Margin

- The noise margin is the maximum noise voltage added to an input signal of a digital circuit that does not cause an undersirable change in the circuit's output.
- It is calculated from the difference V_{OH}-V_{IH} or V_{IL}-V_{OL}, whichever is samller.

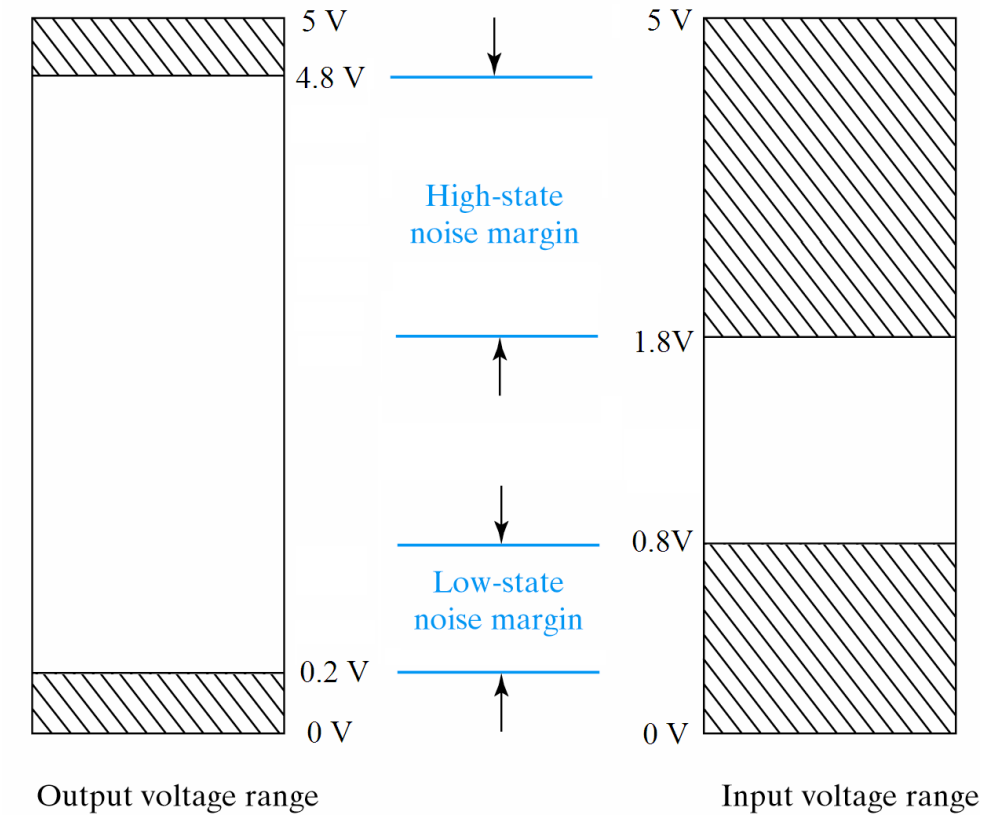


Noise Margin

- V_{OH} = 4.8 V
- V_{OL} = 0.2 V
- V_{IL} = 0.8 V
- V_{IH} = 1.8 V

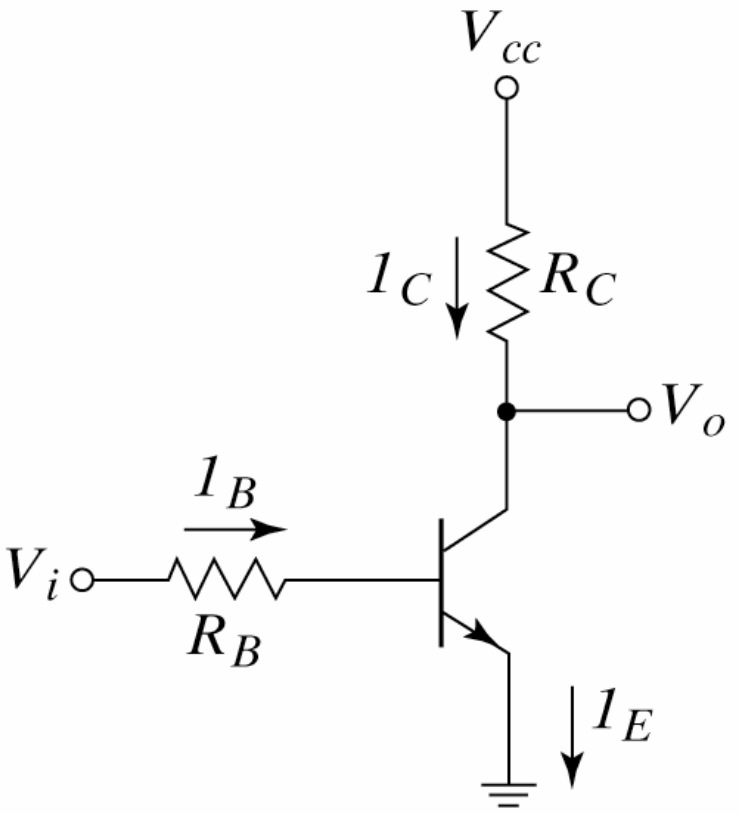
High-state noise margin = 3.0 V

Low-state noise margin = 0.6 V



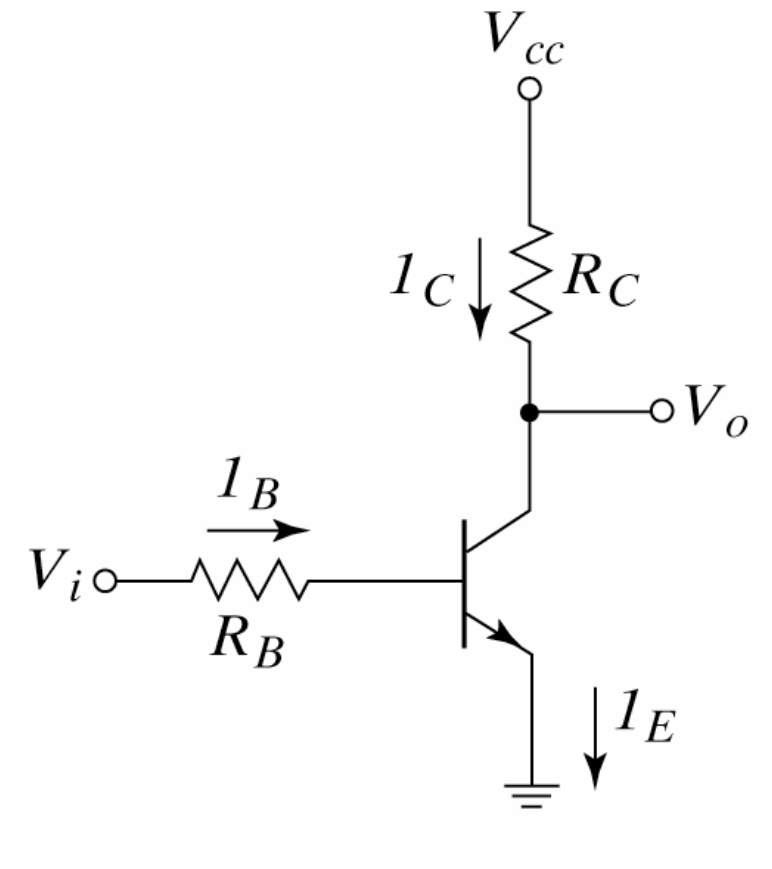
Resistor Transistor Logic

- The bipolar transistor has three modes of operation, cutoff, active and saturation.
- We are interested in the cutoff and saturation modes.
- For a low voltage at V_i the transistor is in cutoff and approximates as an open switch.
- For a high voltage at V_i the transistor is in saturation and approximates a closed switch.



Resistor Transistor Logic

- The **resistor transistor logic (RTL)** family was the first technology to be used within ICs.
- The RTL inverter operates in a similar way to the ideal switching circuit.
- The input and output current must remain within certain limits, such that the transistor does not deviate from the cutoff or saturation modes of operation.

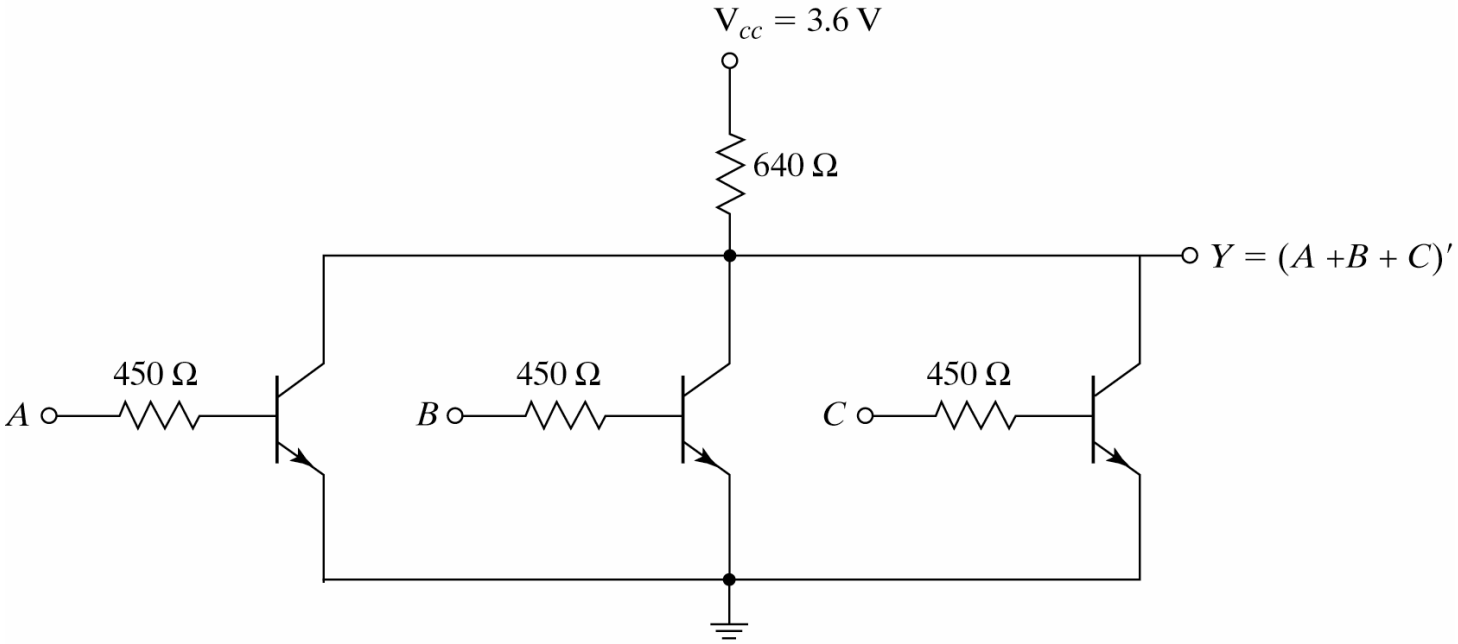


Practical Limitations

- Transistor logic suffers from some practical limitations,
  - such as the number of gates that may be connected to the output of a gate,
  - the time it takes for a logic signal to pass through a gate,
  - the level of noise that the circuit can tolerate
  - and the power requirements for operation.

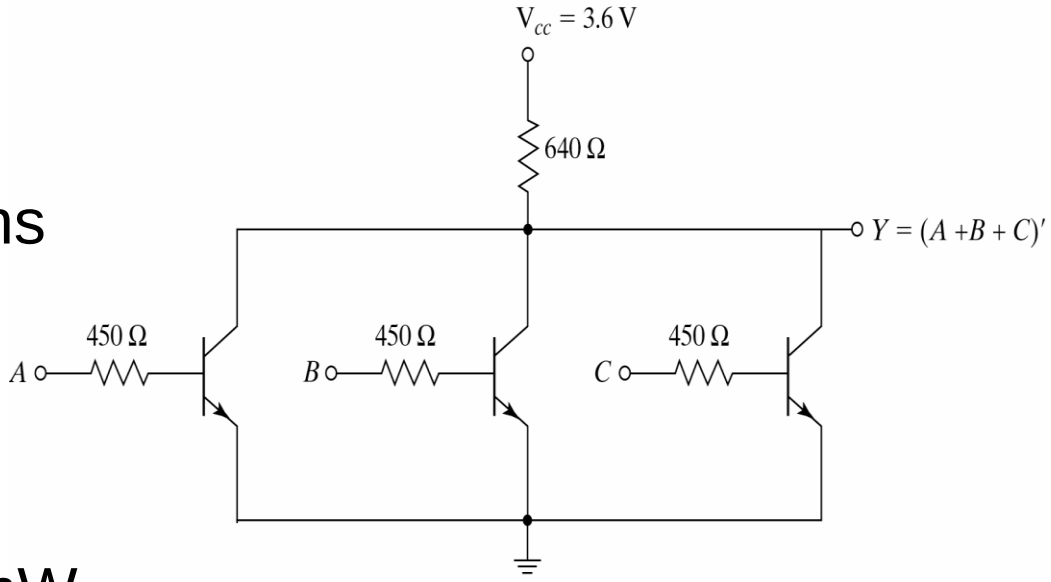
RTL NOR Gate

- If any input is high the corresponding BJT goes into saturation and so output goes low.
- When all inputs are low the BJTs are in cutoff and so output goes high.



RTL NOR Gate

- Fan-Out: 5
- Propagation Delay: 25ns
- Noise Margin: 0.4V
- Power Dissipation: 12mW

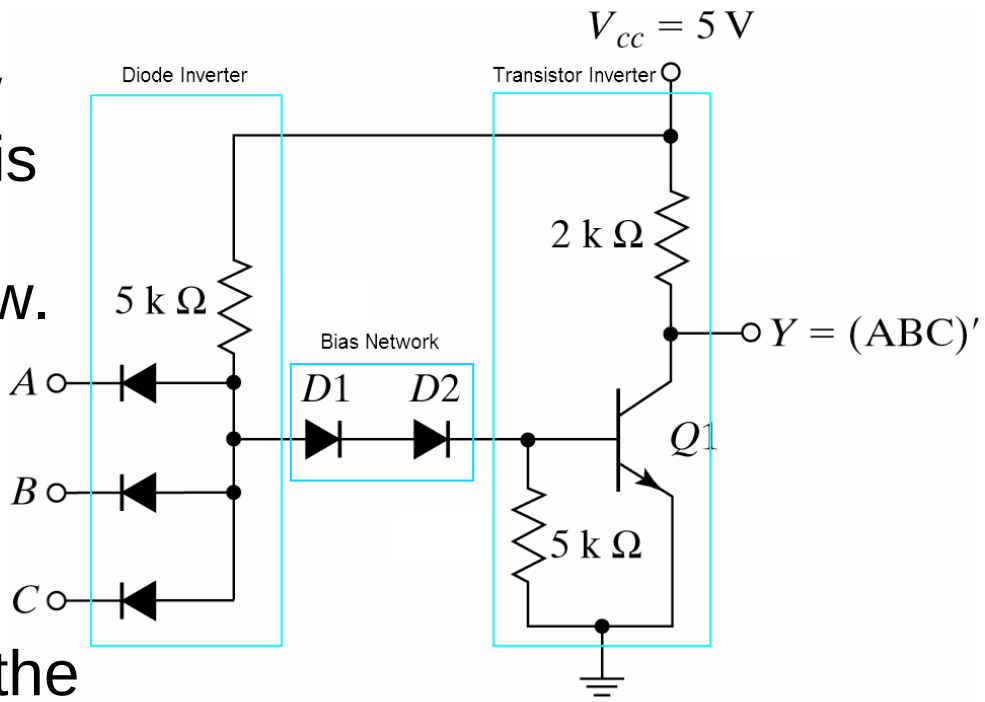


Diode Transistor Logic

- Resistors are usually avoided.
  - Take up too much silicon area
  - A pure resistive element is difficult to realise in silicon. A silicon resistor will contain unwanted inductive and capacitive components.
- The first alternative to the RTL family were circuits using **diode transistor logic (DTL)**.

DTL NAND

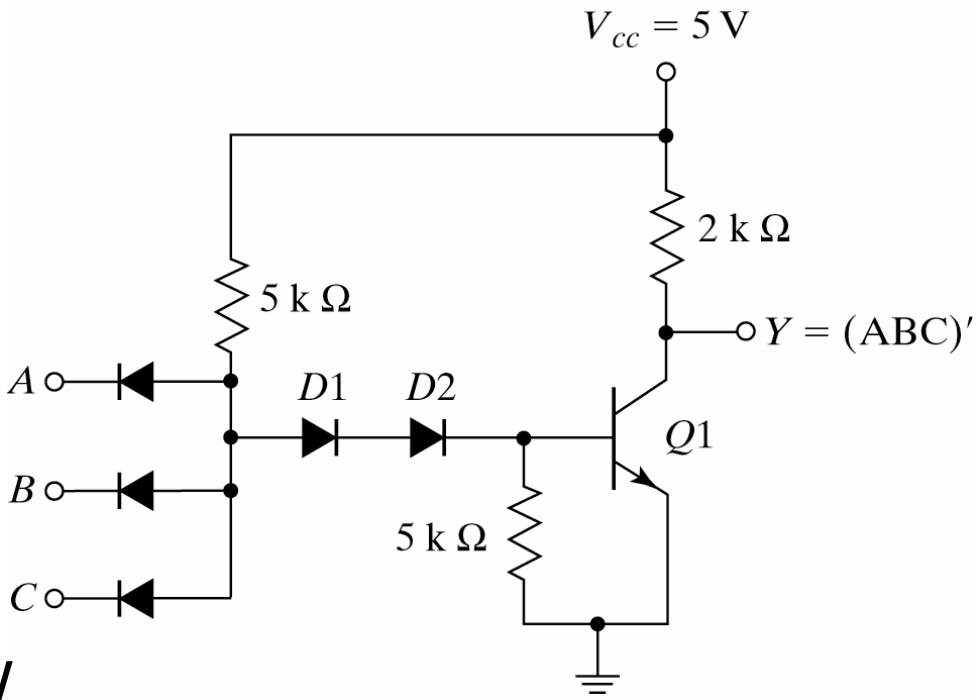
- If any of the inputs to the diode inverter are set low the corresponding diode is activated and the input to the bias network is set low.
- The transistor inverter converts the low voltage input into a high voltage output.
- When all inputs are high the input to the bias network is set high and the output of the transistor inverter is low.



DTL Basic NAND Gate

DTL NAND

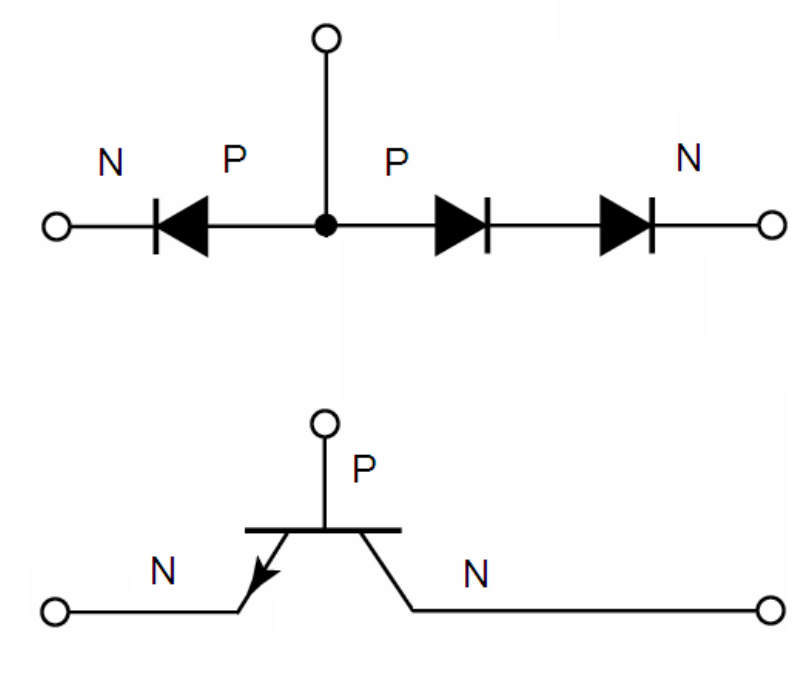
- Fan-Out: 8
- Propagation Delay: 30ns
- Noise Margin: 1V
- Power Dissipation: 12mW



DTL Basic NAND Gate

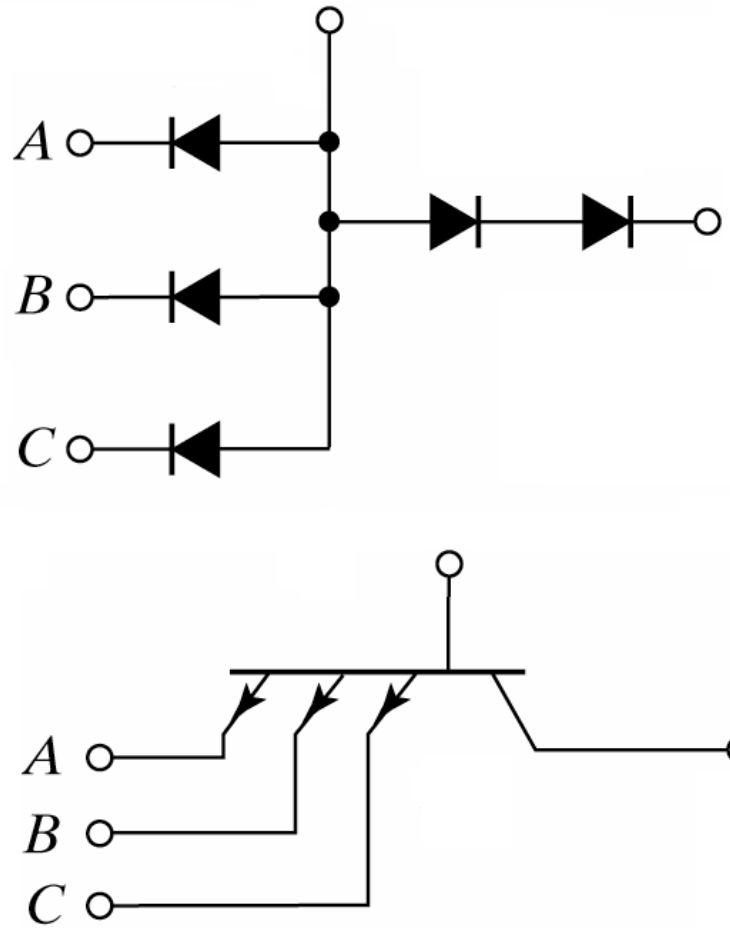
Transistor Transistor Logic

- Since it takes as much silicon area to produce a diode as a transistor, DTL was replaced by the **transistor transistor logic (TTL)**.
- The N-P and P-N junction configuration of the DTL gate is replaced by an NPN transistor.



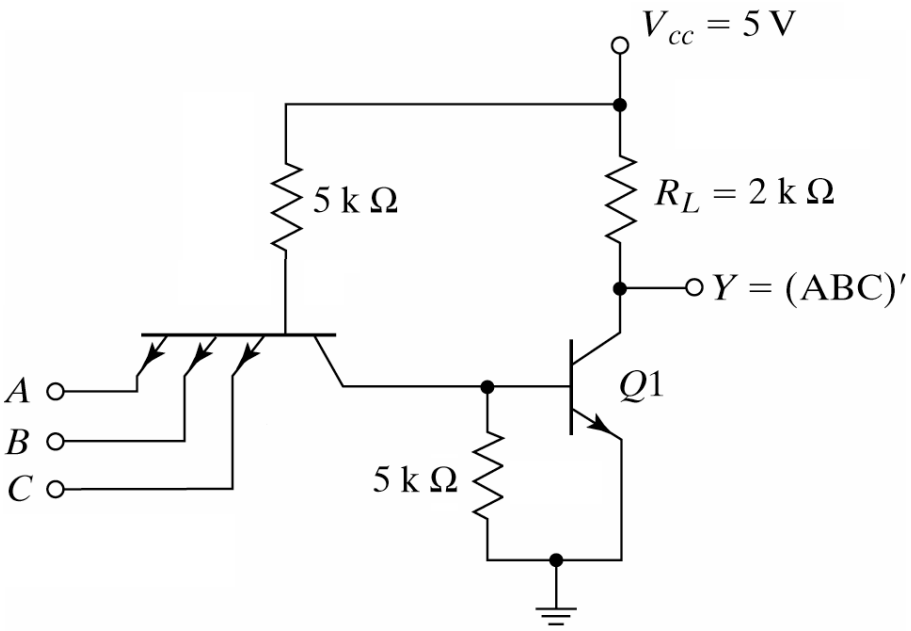
Transistor Transistor Logic

- Parallel diode inputs are realised using a transistor with multiple emitters.



TTL NAND Gate

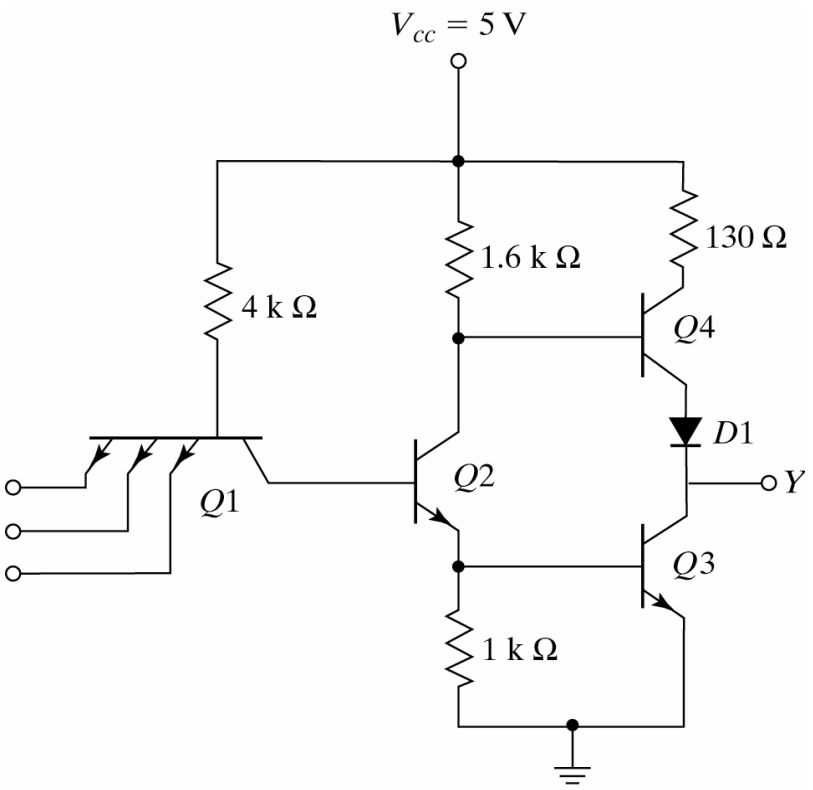
- The TTL NAND gate follows from the DTL NAND gate.
- The transistor acts to actively pull the output to zero when the cutoff mode is entered.
- The load resistor R_L acts as a **passive pull-up** circuit. Pulling the load up to the high voltage V_{cc}. The speed at which this occurs is governed by the speed the collector of the BJT can be charged.



TTL Basic NAND Gate

TTL Totem Pole

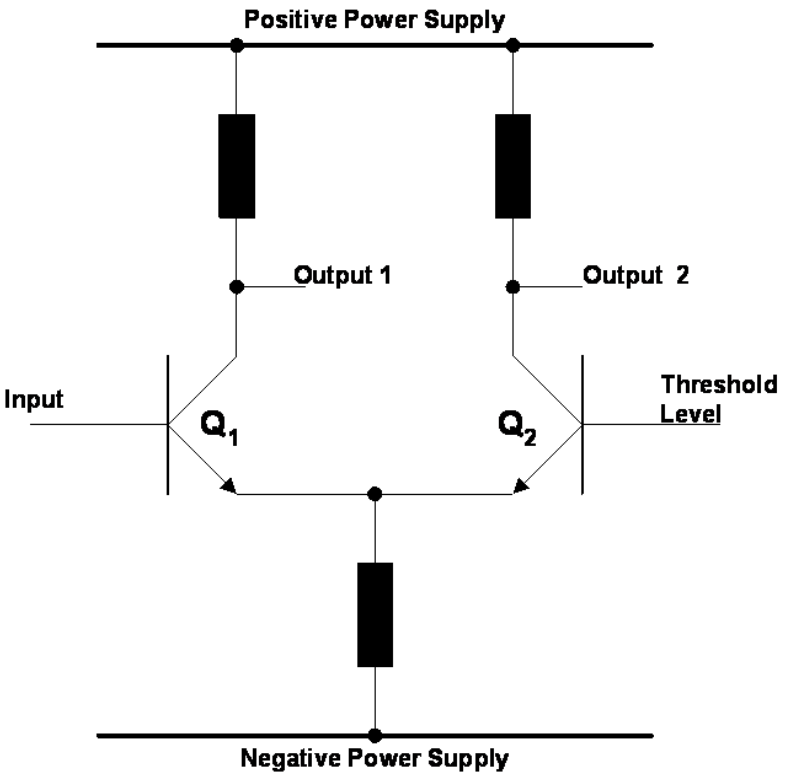
- To improve the speed of the circuit the passive load is replaced by an active load.
- The active totem pole configuration allows faster operation. Now the collector of Q_3 is actively charged by Q_4 when saturation is entered.
- Propagation Delay: 10 ns



TTL Gate with Totem-Pole Output

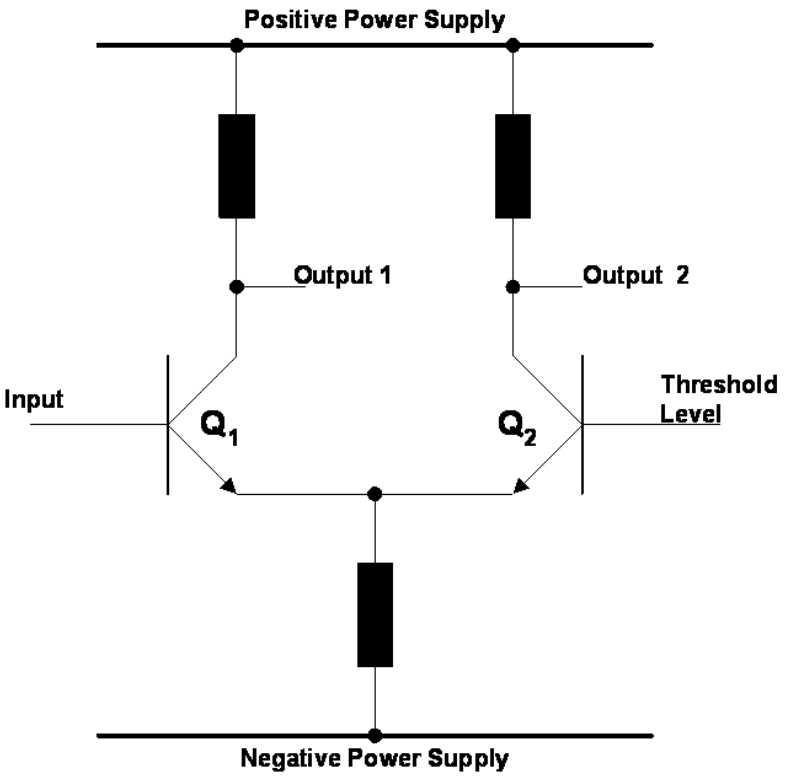
Emitter Coupled Logic

- **Emitter Coupled Logic (ECL)** is based upon this symmetric transistor circuit.
- When the input voltage is higher than the threshold level
  - Q_1 is active and Q_2 is in cutoff
  - Output 1 is low and output 2 is high
- When the input voltage is lower than the threshold level
  - Q_1 is cutoff and Q_2 is in active
  - Output 1 is high and output 2 is low



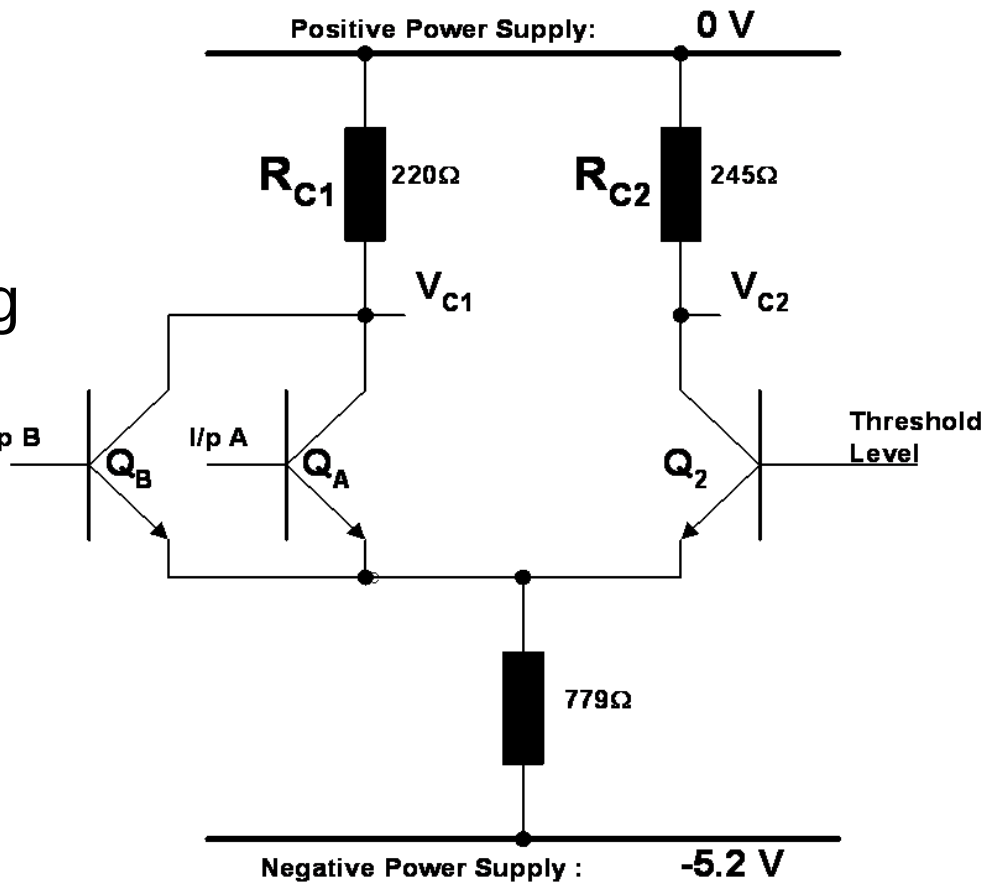
Emitter Coupled Logic

- The transistors in the ECL circuit both hover either side of the boundary between the active and cutoff regions.
- The switching between these two modes occurs rapidly.
  - Propagation Delay: 1-2 ns
- Because current is always flowing the power usage is high
  - Power Dissipation: 25 mW



ECL NOR-OR Gate

- Multiple inputs can be introduced.
- Now when any of the inputs are high (above threshold level) Q_2 is turned off putting V_{c2} high.
- When all inputs are low Q_2 is turned on putting V_{c2} low.
- The V_{c2} output corresponds to an OR gate and the V_{c1} output corresponds to a NOR gate.



ECL Design

- ECL is a difficult logic family for designing a digital system.
  - Due to speed with which ECL operates
  - Narrower Noise margins
  - Expensive
  - Harder to cool
  - Difficult to interconnect

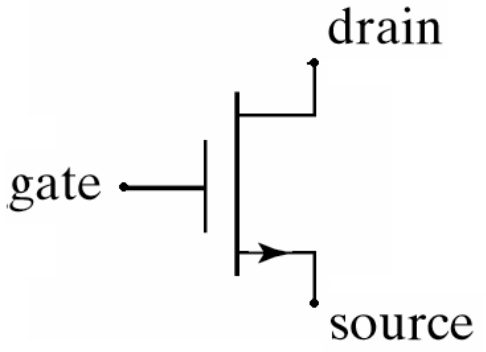
- But can be useful when fast implementation is required for an application.

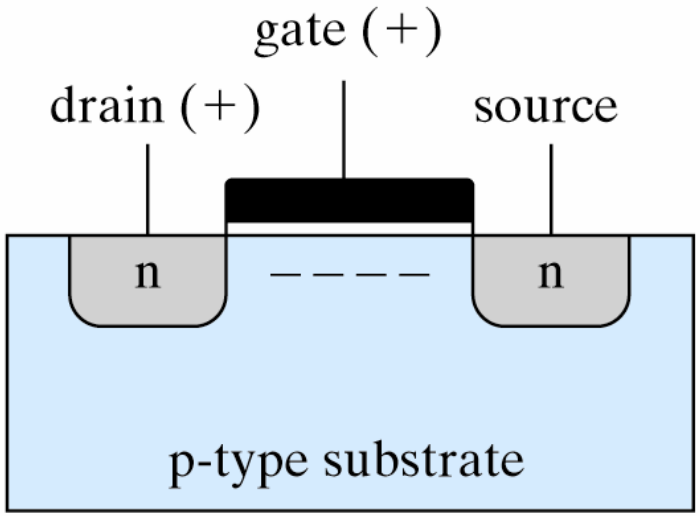
MOS Transistors

- MOS transistors have a simple structure.
- A n-type (p-type) silicon substrate has two p-type (n-type) wells separated by an n-type (p-type) channel.
- Above the channel there is a gate electrode. The gate electrode is separated form the channel by the insulating material Silicon Oxide.

NMOS Transistor

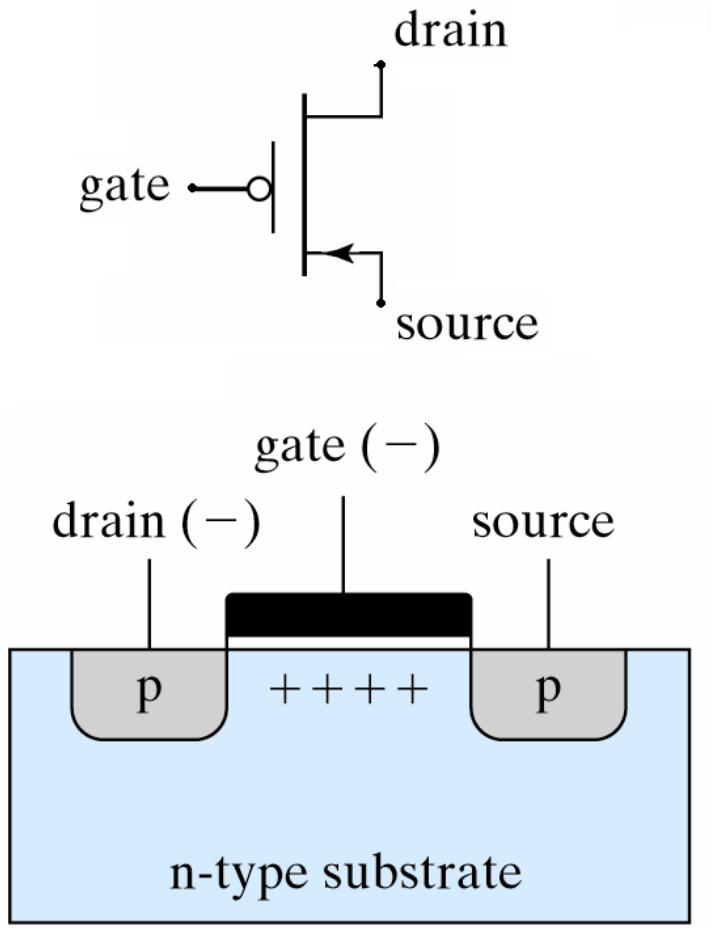
- The n-type MOSFET consists of a p-type substrate containing n-type wells.
- When the gate voltage is below a certain threshold the p-type channel acts as a barrier to conduction between the n-type wells.
- When the gate voltage is above the threshold the negative charge carriers in the channel allow conduction between the n-type wells.





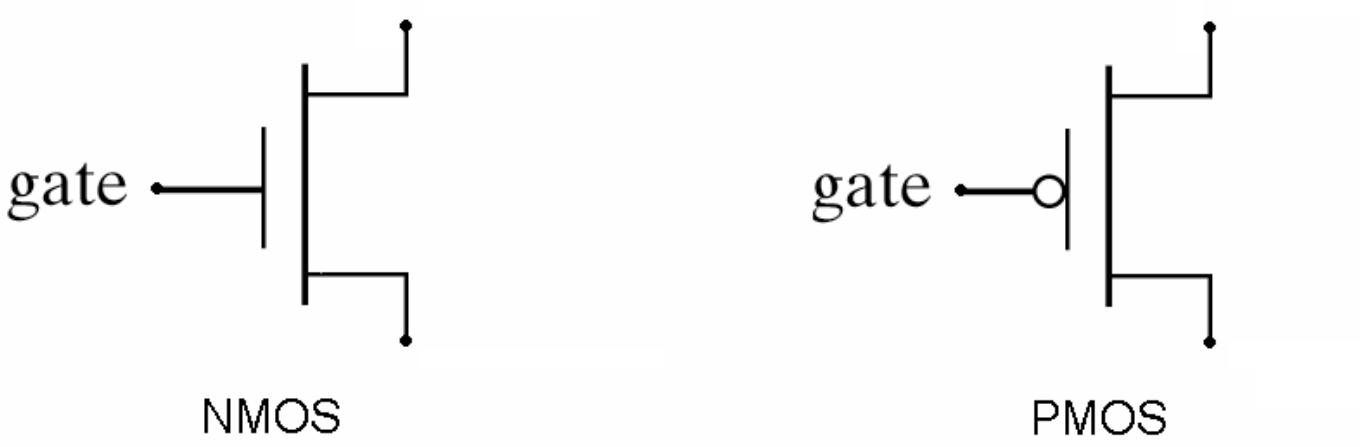
PMOS Transistor

- The p-type MOSFET consists of a n-type substrate containing p-type wells.
- When the gate voltage is above a certain threshold the n-type channel acts as a barrier to conduction between the p-type wells.
- When the gate voltage is below the threshold the positive charge carriers in the channel allow conduction between the p-type wells.



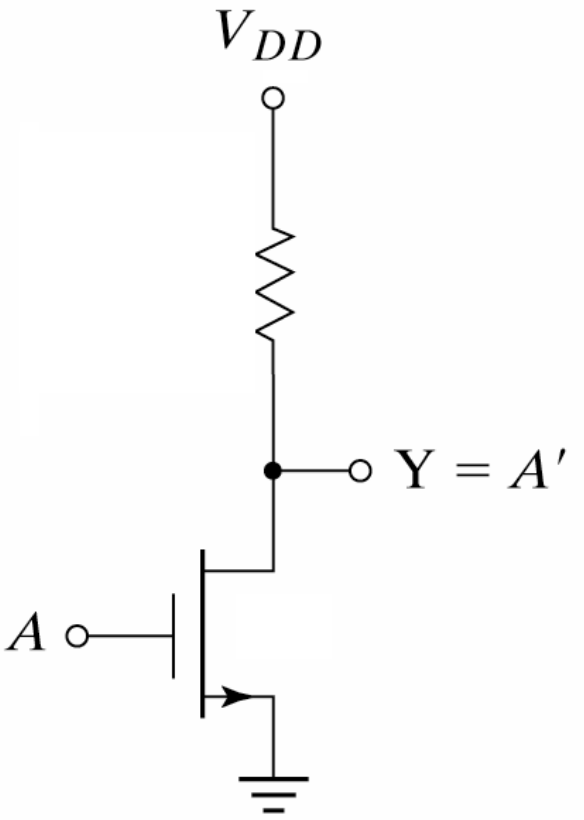
MOS Transistors

- When the gate voltage is high the NMOS transistor acts as a closed switch and when the gate voltage is low the NMOS transistor acts as an open switch.
- When the gate voltage is low the PMOS transistor acts as a closed switch and when the gate voltage is high the PMOS transistor acts as an open switch.



NMOS Inverter

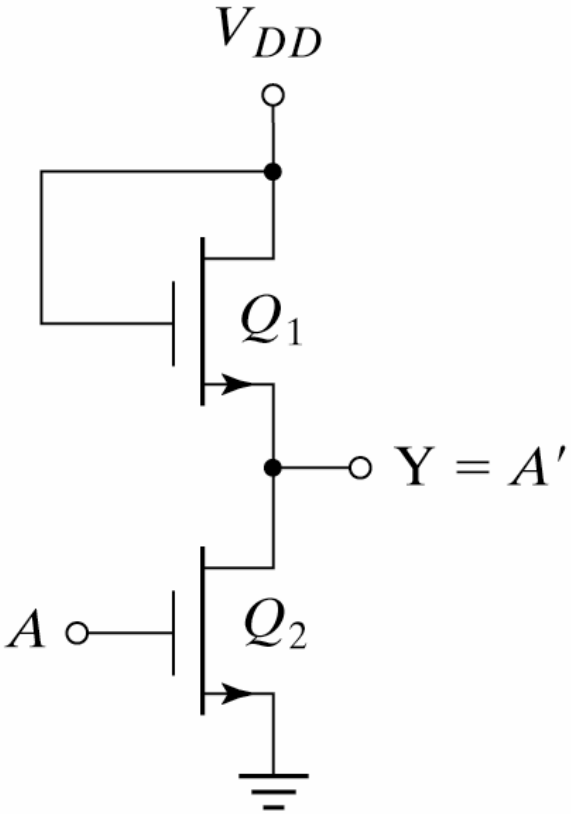
- It follows that the NOT switching circuit can be implemented using an NMOS transistor.
- Case 1: A is low and the NMOS doesn't conduct. No current flows through the resistor and so there is no voltage drop and Y is high.
- Case 2: A is high and the NMOS does conduct. The output is connected to ground through the switch and Y is low.



NMOS Inverter

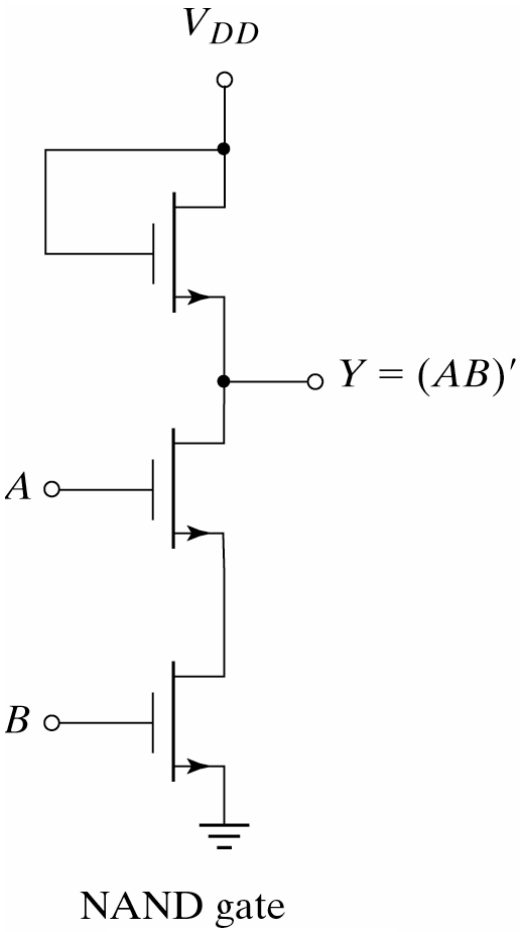
NMOS Inverter

- Another advantage of MOS technology is the ability to create resistive loads using a MOSFET.
- The gate of Q_1 is always set high putting the Q_1 permanently in the saturation region.
- The conducting channel acts as a resistive load. The resistance of this load can specified during manufacture by altering the length and width of the channel.

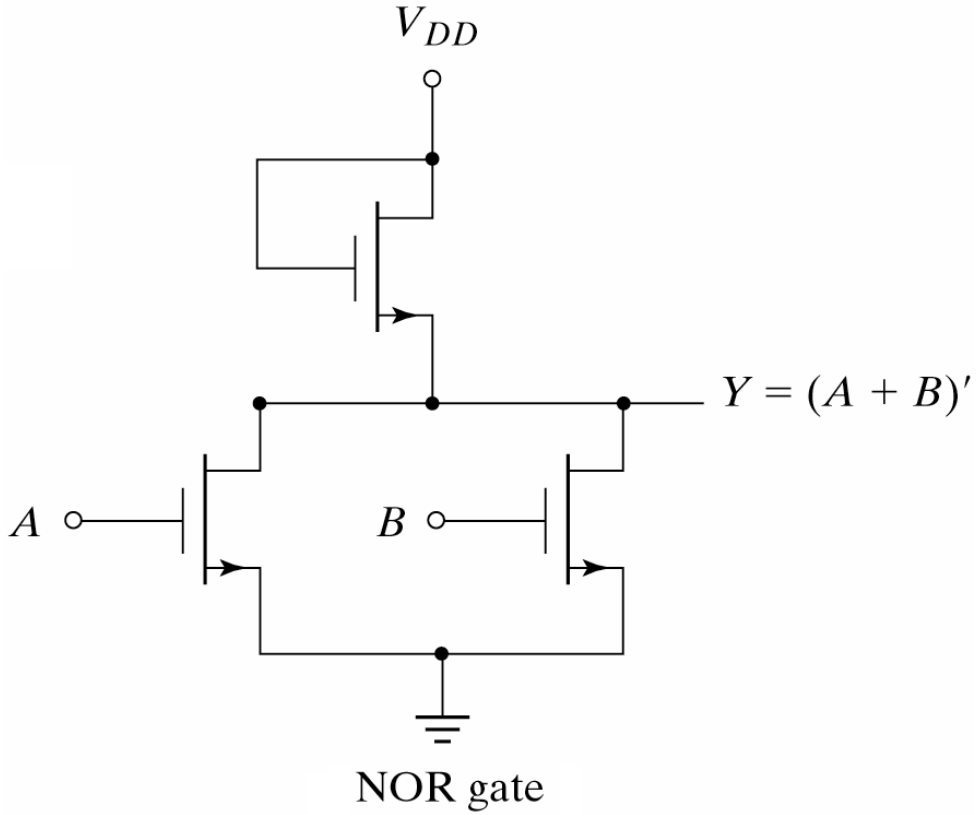


NMOS Inverter

NAND and NOR Gates



NAND gate



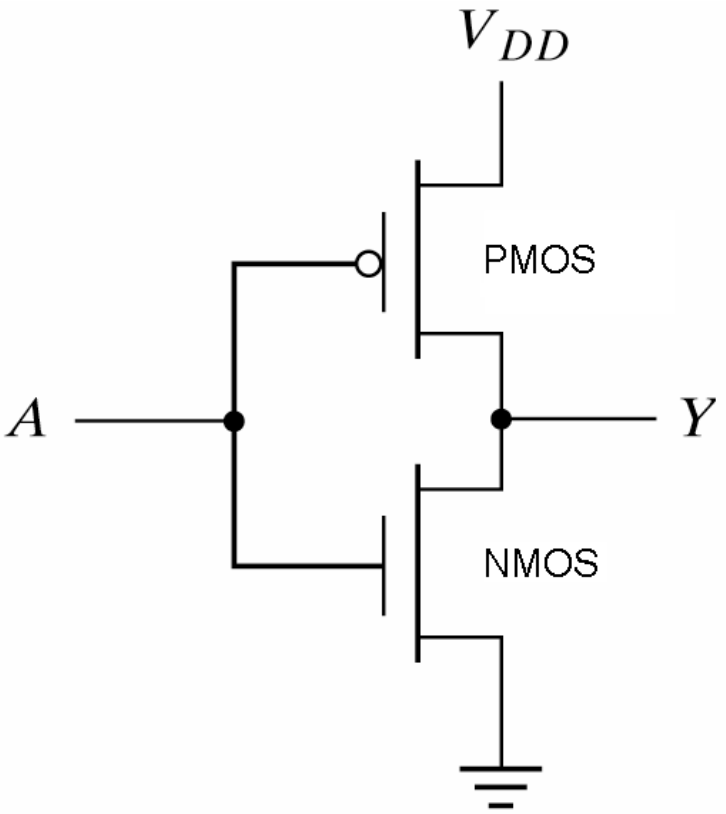
NOR gate

CMOS Logic

- As with the BJT based logic families the NMOS inverter has the disadvantage that current flows when the transistor is in a conducting state.
- This can be overcome by using both NMOS and PMOS technologies.
- The two different gates complement each other, hence the name **CMOS (Complementary MOS)**.

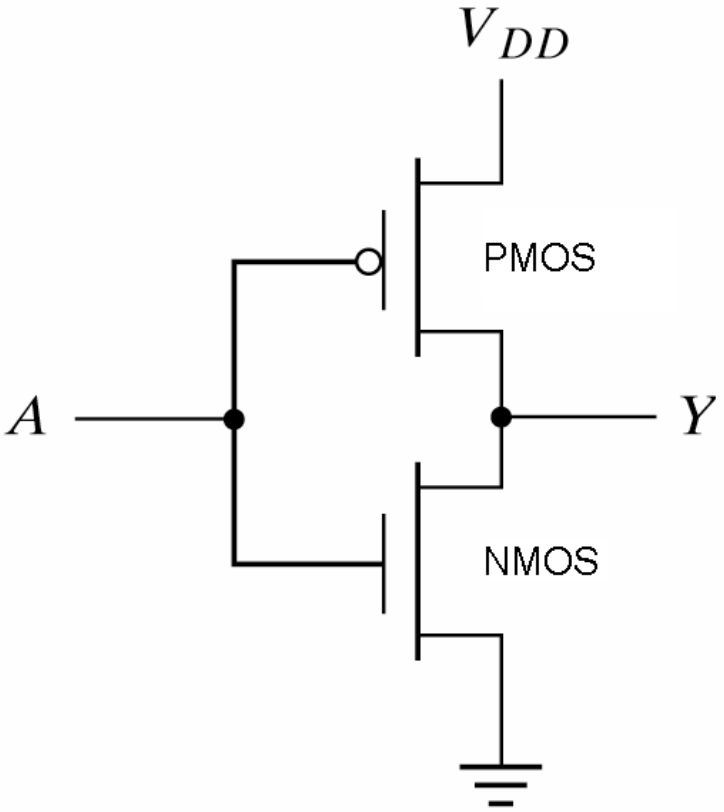
CMOS Inverter

- The CMOS Inverter consists of an NMOS and PMOS pair.
- Case 1:
  - A is set to a high voltage, the NMOS conducts and the PMOS is turned off.
  - The output Y is connected to th ground through the NMOS and so is set low.
  - No current flows because the PMOS is turned off.



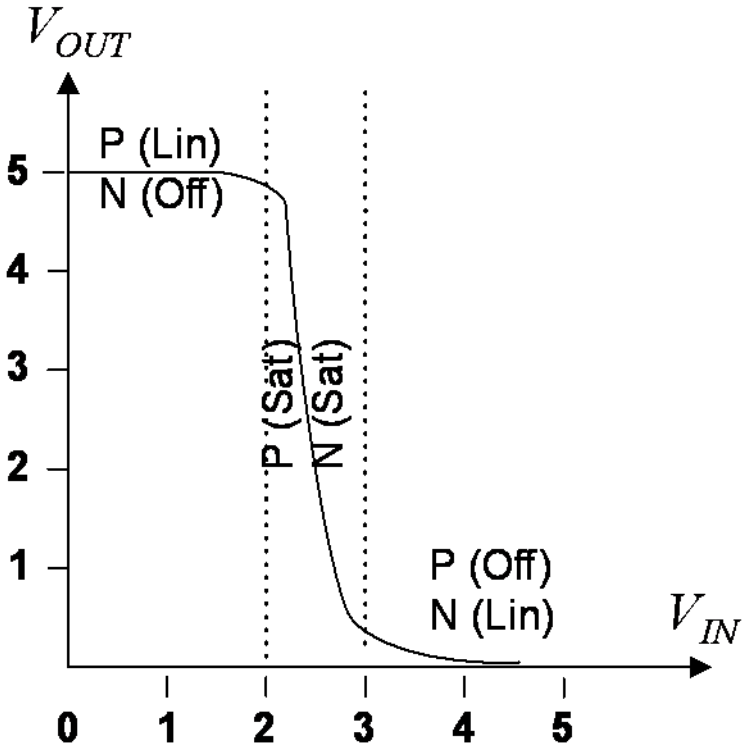
CMOS Inverter

- Case 2:
  - A is set to a low voltage, the NMOS is turned off and the PMOS conducts.
  - The output Y is connected to the line through the PMOS and so is set high.
  - No current flows because the NMOS is turned off.



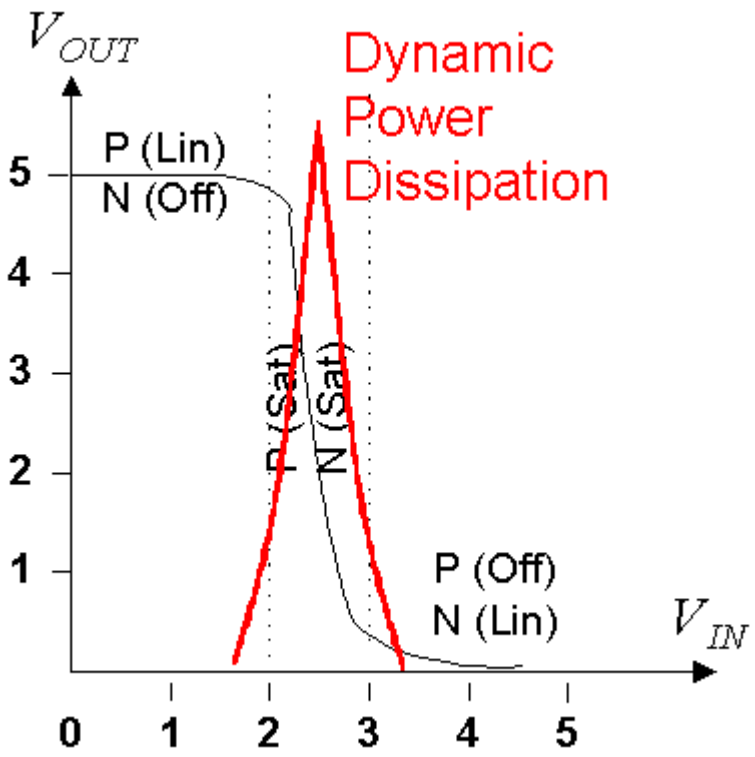
CMOS Power Dissipation

- No current flows in either of the static states!
- So no power is dissipated in either of the static states!!
- The voltage transfer characteristic of a typical CMOS Inverter is almost ideal.
- In the high output state the output voltage reaches 5 V and in the low output state the voltage reaches 0 V.
- The noise margin is 2 V.



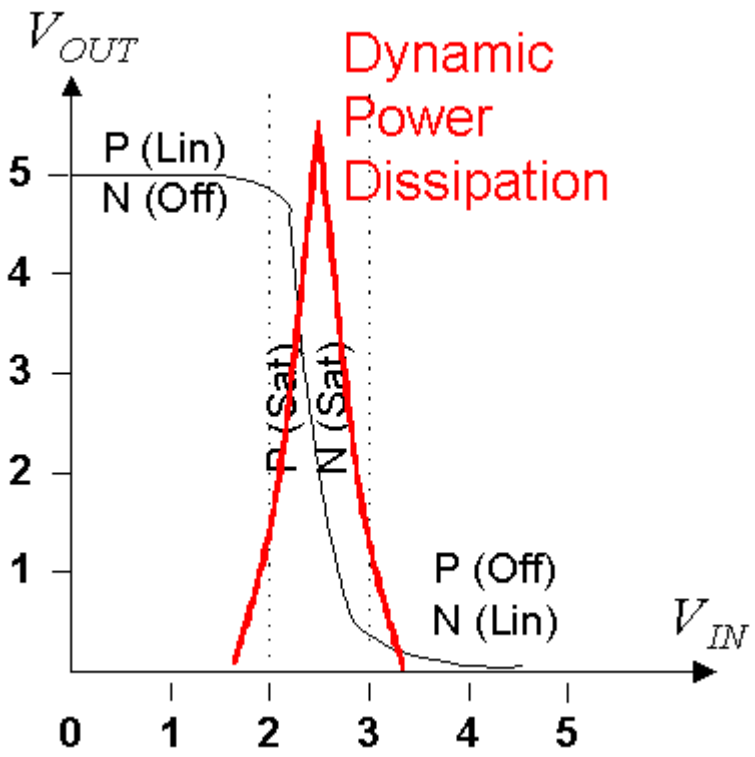
CMOS Power Dissipation

- However during the transition state both the NMOS and PMOS transistors can conduct at the same time.
- As a result CMOS logic does suffer from dynamic power dissipation.
- Dynamic power dissipation increases as the switching speed increases.



CMOS Power Dissipation

- Static power dissipation is of the order of 0.01 mW.
- Dynamic power dissipation is frequency dependent.
  - At 1MHz the dissipation is about 1mW
  - At 10MHz the dissipation is about 5mW

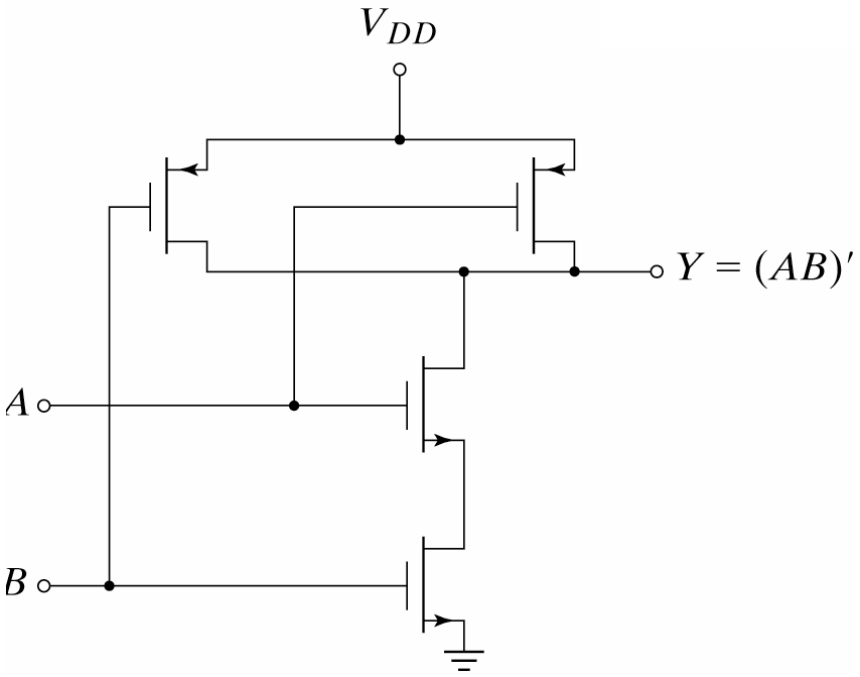


CMOS Fan Out

- Since the inputs to CMOS gates do not require current the fan-out of CMOS gates is much greater than that of TTL gates.
- CMOS fan-out is limited by the capacitive effects introduced by extra gates. The capacitive effects limit the frequency of operation.
- At 30MHz the fan-out is about 30.
- The fan-out decreases as the frequency of operation increases.

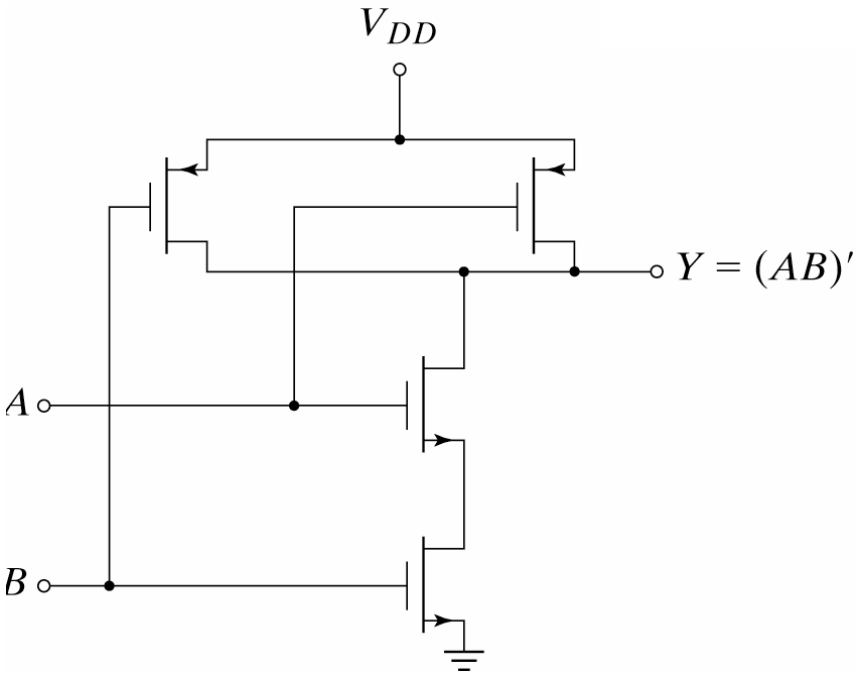
CMOS NAND Gate

- Case 1:
  - A and B are set high, both NMOS transistors are on and both PMOS transistors are off.
  - The output Y is connected to ground through the NMOS transistors and is set low to 0 V.



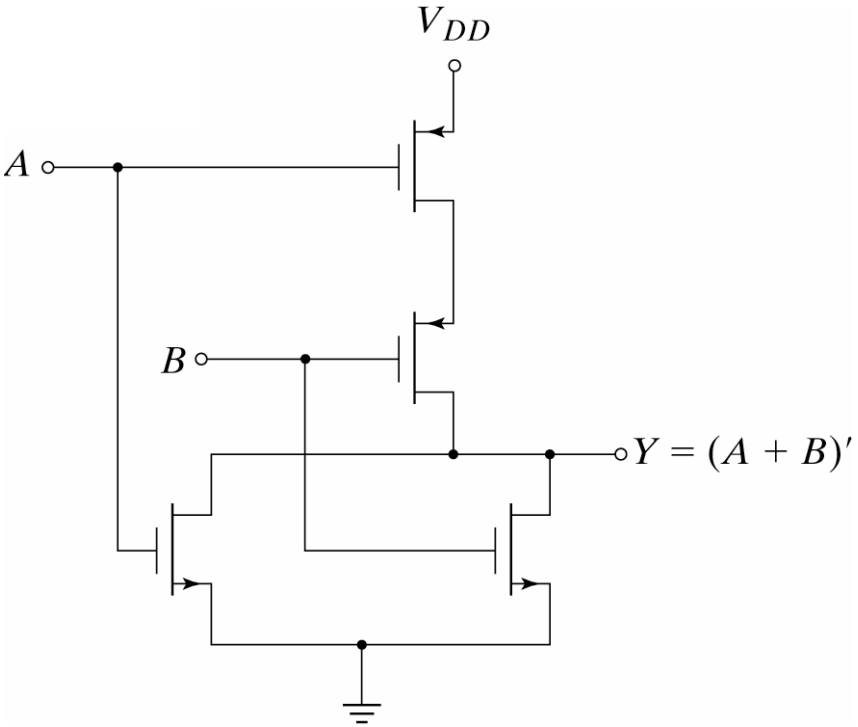
CMOS NAND Gate

- Case 2:
  - A and/or B are set low, one or both of the NMOS transistors are off and one or both of the PMOS transistors are on
  - The output Y is connected to V_{DD} through one or both of the PMOS transistors and is set high to V_{DD}.



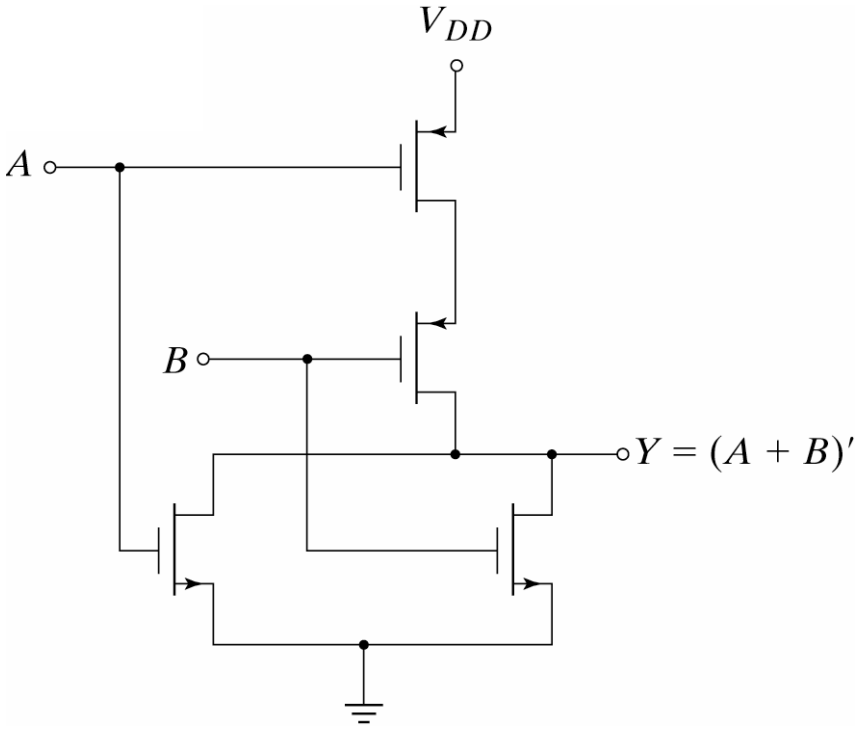
CMOS NOR Gate

- Case 1:
  - A and B are set low, both NMOS transistors are off and both PMOS transistors are on.
  - The output Y is connected to V_{DD} through the PMOS transistors and is set high to V_{DD}.



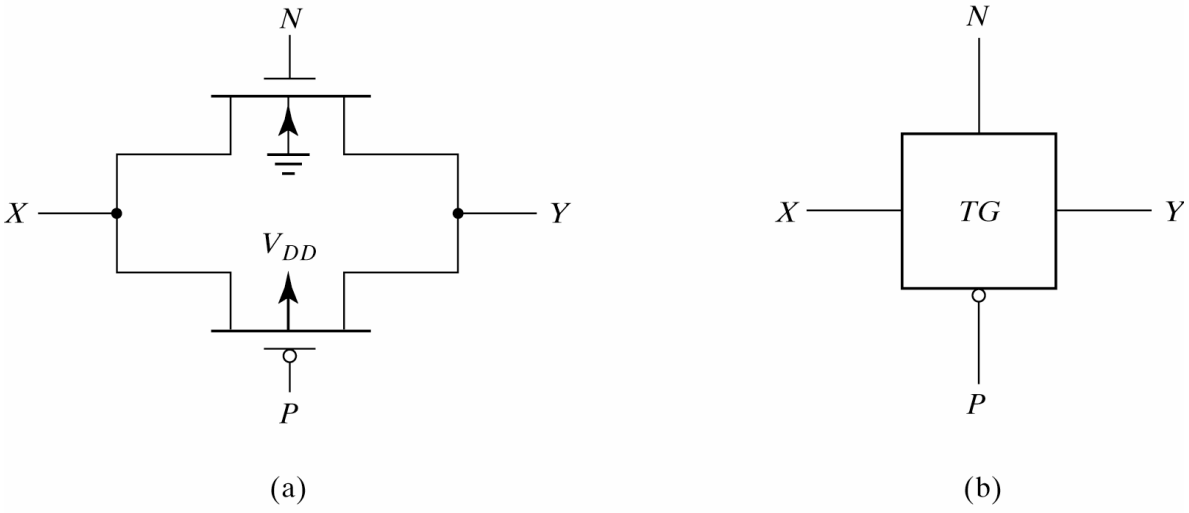
CMOS NOR Gate

- Case 2:
  - A and/or B are set high, one or both of the NMOS transistors are on and one or both of the PMOS transistors are off.
  - The output Y is connected to ground through one or both of the NMOS transistors and is set low to 0 V.



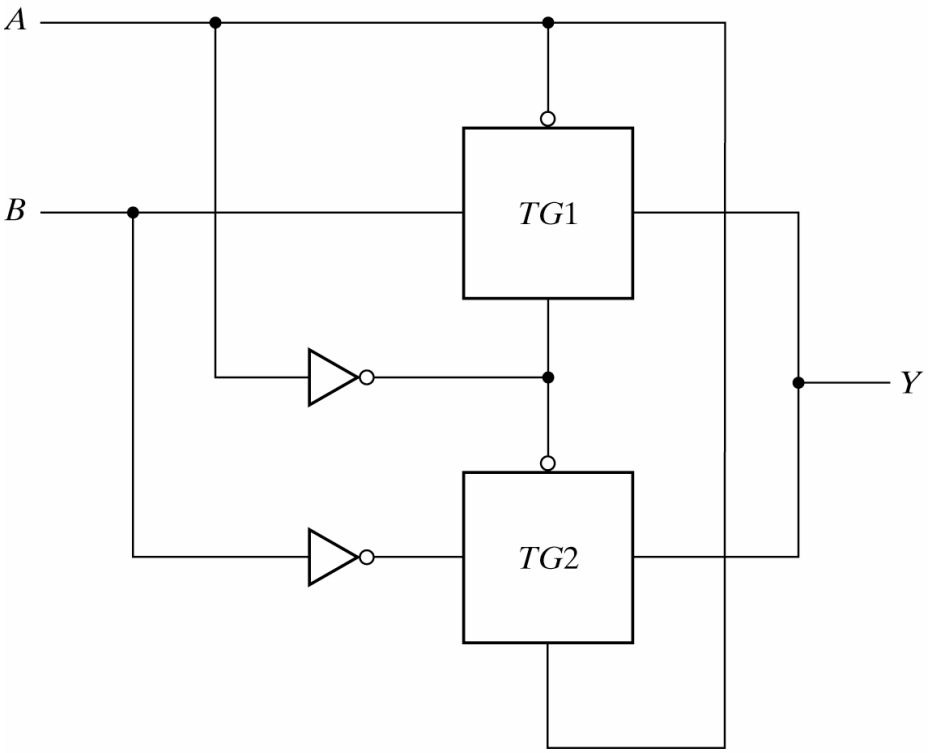
Transmission Gate

- The CMOS **transmission gate** is a useful component for constructing digital circuits in an economical way.
- Case 1: N is high, P is low. Both transistors conduct and there is a closed path between X and Y.
- Case 2: N is low, P is high. Both transistors are off and there is an open circuit between X and Y.



Exclusive OR

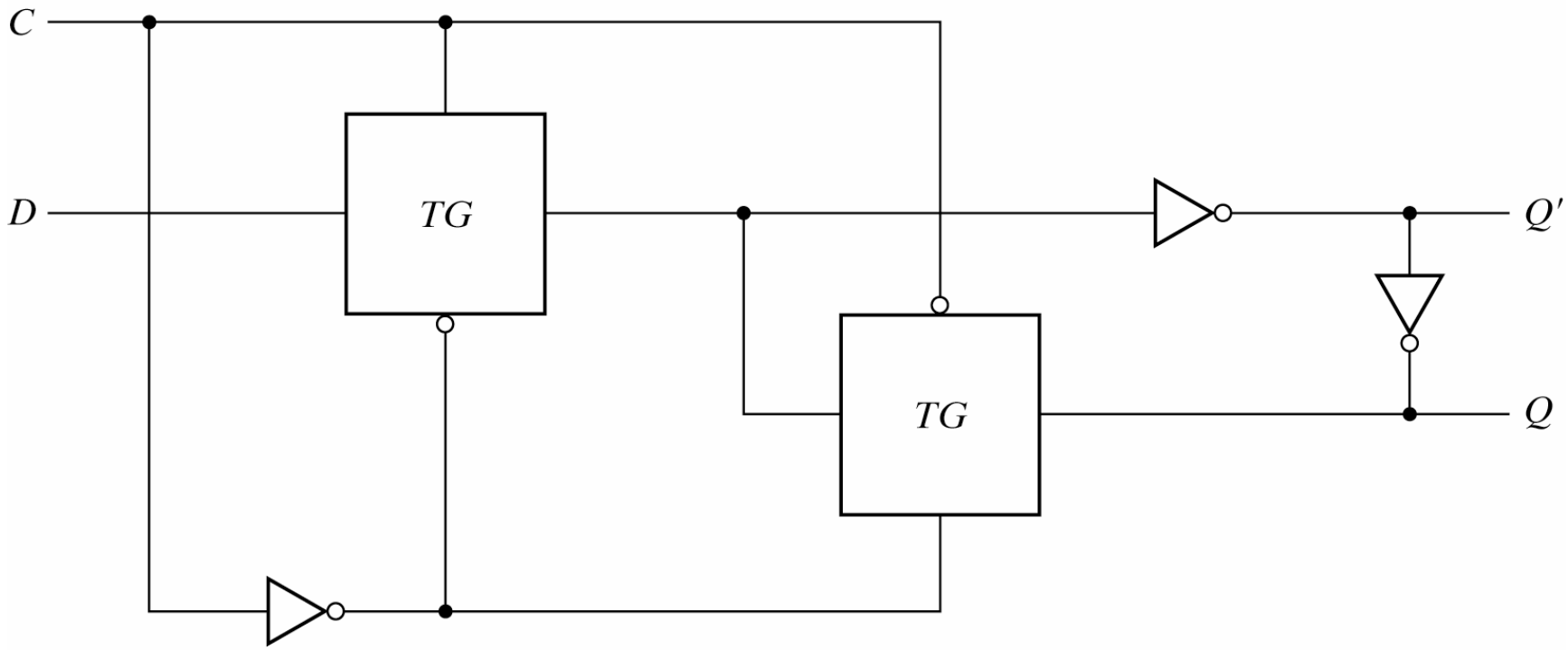
- An exclusive OR gate can be implemented using two transmission gates and two inverters.
- We require 8 transistors.



A	B	$TG1$	$TG2$	Y
0	0	close	open	0
0	1	close	open	1
1	0	open	close	1
1	1	open	close	0

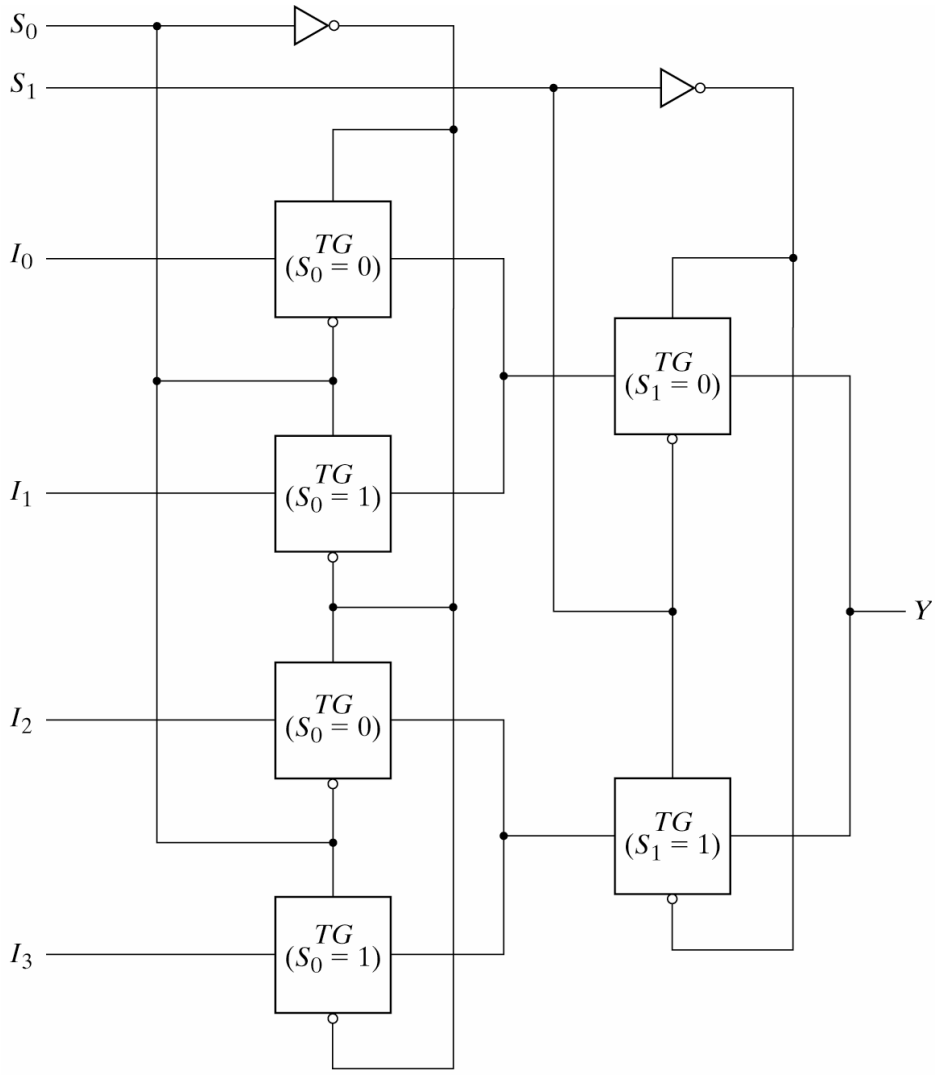
D Latch

- A D Latch can be realised using 10 transistors.



Multiplexer

- Here a 4 to 1 line multiplexer is realised using 14 transistors.

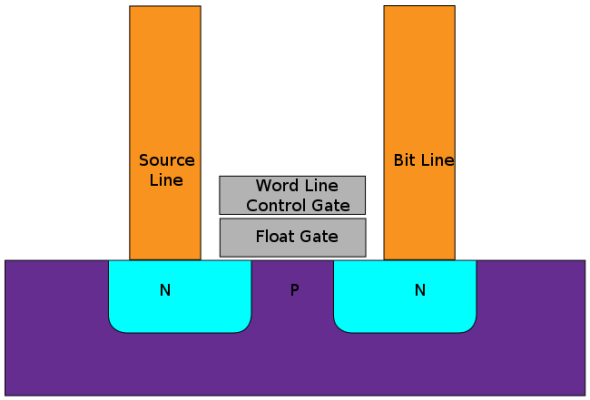


CMOS Logic Family

- Negligible static power dissipation
- Dynamic power dissipation increases and fan-out decreases with frequency.
- Complementary transistor implemented on same substrate take up even less silicon area.
- Transmission gates simplify logic implementations.

Flash Memory

- Flash memory stores information in an array of memory cells. Each memory cell is made from floating-gate transistors.
- These transistors have two gates: on top is the control gate (CG), as in other MOS transistors, but below this there is a floating gate (FG) insulated all around by an oxide layer.
- The FG is interposed between the CG and the channel. Because the FG is electrically isolated by its insulating layer, any electrons placed on it are trapped there and, under normal conditions, will not discharge for many years.

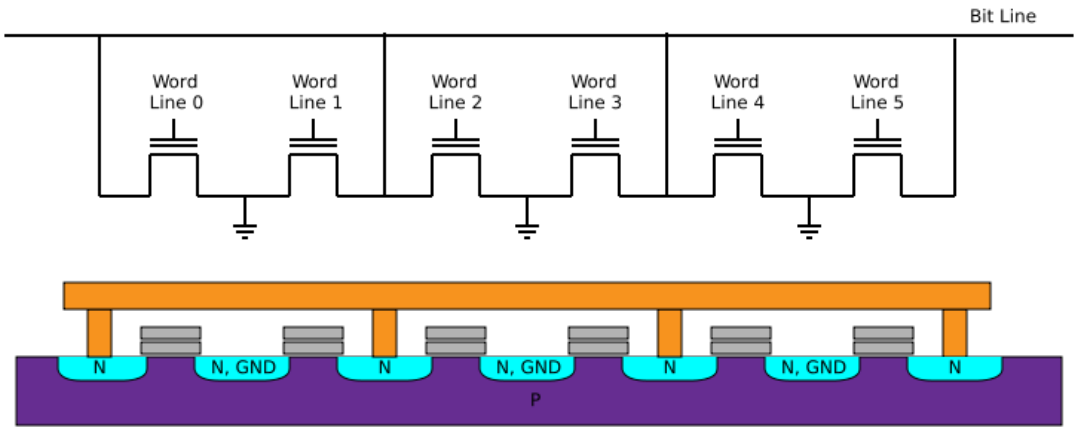


Flash Memory

- When the FG holds a charge, it screens (partially cancels) the electric field from the CG, which modifies the threshold voltage (V_T) of the cell.
- During read-out, a voltage intermediate between the possible threshold voltages is applied to the CG, and the MOSFET channel will become conducting or remain insulating, depending on the V_T of the cell, which is in turn controlled by charge on the FG.
- The current flow through the MOSFET channel is sensed and forms a binary code, reproducing the stored data.

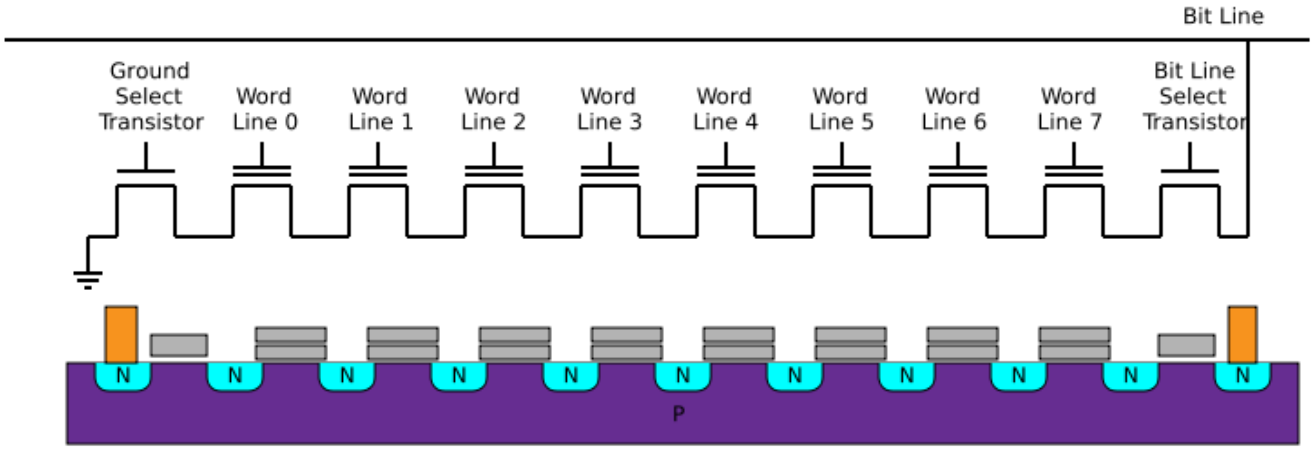
NOR Flash

- In NOR gate flash, each cell has one end connected directly to ground, and the other end connected directly to a bit line.
- This arrangement is called "NOR flash" because it acts like a NOR gate: when one of the word lines is brought high, the corresponding storage transistor acts to pull the output bit line low.



NAND Flash

- In NAND flash, several transistors are connected in series, and only if all word lines are pulled high (above the transistors' V_T) is the bit line pulled low. These groups are then connected via some additional transistors to a NOR-style bit line array.



Error Detection and Correction

- Memory arrays are complex structures containing many interconnection lines and many multi-input gates .
- This complexity level can lead to errors in storing and retrieving information.
- To overcome errors memory units use error detecting and correction codes.
- A common method is to generate so called **parity** bits for each word and store them with the words in memory.
- When the word is read the parity bits can be used to check for errors.

Hamming Code

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...	
Encoded data bits		p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15		
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X			P_1
	p2		X	X			X	X			X	X			X	X			X	X		...	P_2
	p3				X	X	X	X					X	X	X	X					X		P_4
	p4								X	X	X	X	X	X	X	X							P_8
	p5																X	X	X	X	X		P_{16}

General Rule:

- Every power of 2 position used for parity bits
- Every other position used to store data bits
- Parity groups for bit n ($n=2^{i-1}$): skip n-1, check n, skip n, check n, skip n, ...

  → each bit has unique parity bit coverage

Hamming Code

- As an example we consider the 8 bit word 10110100
- We generate 4 parity bits

$$\{P_1, P_2, P_4, P_8\}$$

and place them at the 1st, 2nd, 4th and 8th bit positions in a new 12 bit word.

1	2	3	4	5	6	7	8	9	10	11	12
P_1	P_2	1	P_4	0	1	1	P_8	0	1	0	0

Hamming Code

- The parity bits are generated using an exclusive OR operation on data bits.
  - an even number of ones returns a 0
  - an odd number of ones returns a 1

$$P_1 = \text{XOR of bits}(3, 5, 7, 9, 11) = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P_2 = \text{XOR of bits}(3, 6, 7, 10, 11) = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{XOR of bits}(5, 6, 7, 12) = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$P_8 = \text{XOR of bits}(9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

Hamming Code

- The twelve bit word

1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	0	1	1	1	0	1	0	0

  is stored in memory.
- When the word is read four check bits are generated

$$C_1 = \text{XOR of bits}(1, 3, 5, 7, 9, 11)$$
$$C_2 = \text{XOR of bits}(2, 3, 6, 7, 10, 11)$$
$$C_4 = \text{XOR of bits}(4, 5, 6, 7, 12)$$
$$C_8 = \text{XOR of bits}(8, 9, 10, 11, 12)$$

Hamming Code

- If there are no errors each of the check bits will be zero.
- If say the fifth bit is in error i.e. the word

1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	1	1	1	1	0	1	0	0

  is read.
- The check bits are now

$$C_1 = \text{XOR of bits}(1, 3, 5, 7, 9, 11) = 1$$

$$C_2 = \text{XOR of bits}(2, 3, 6, 7, 10, 11) = 0$$

$$C_4 = \text{XOR of bits}(4, 5, 6, 7, 12) = 1$$

$$C_8 = \text{XOR of bits}(8, 9, 10, 11, 12) = 0$$

Hamming Code

- The non zero check bits indicate an error.
- We need to find out where the error is to correct it. How?
- The check bits give us this information…

$$\{C_8, C_4, C_2, C_1\} = 0101 \Rightarrow 5$$

  i.e. the fifth bit position was read in error.
- Since it is a binary system we can replace the 1 in the fifth position with a 0.
- Only can detect and correct a single bit error, OR detect double bit errors but not correct them.