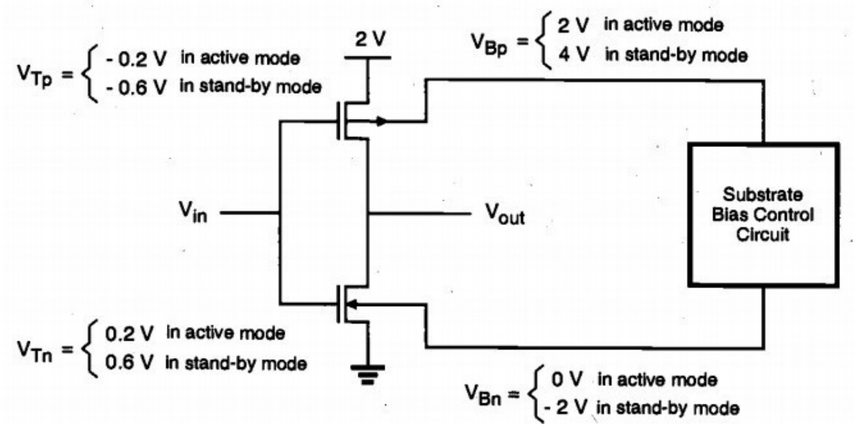
iv.) Variable-Threshold CMOS (VTCMOS) Circuits

We have seen that using a low supply voltage (V_{DD}) and a low threshold voltage (V_T) in CMOS logic circuits is an efficient method for reducing the overall power dissipation, while maintaining high speed performance. Yet designing a CMOS logic gate entirely with low-V_T transistors will inevitably lead to increased subthreshold leakage, and consequently, to higher stand-by power dissipation when the output is not switching. One possible way to overcome this problem is to adjust the threshold voltages of the transistors in order to avoid leakage in the stand-by mode, by changing the substrate bias.

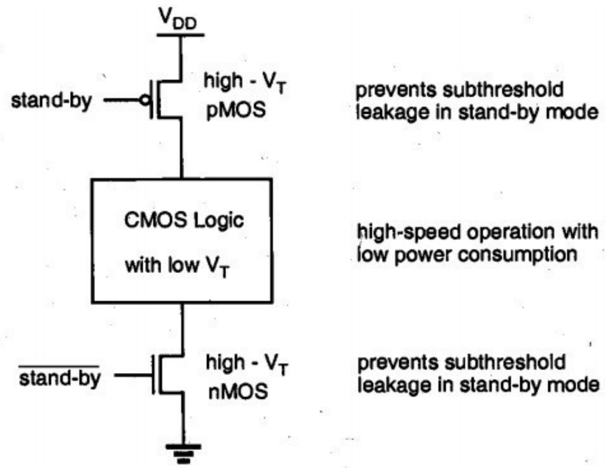


$V_{Tp} = \begin{cases} -0.2\text{ V} & \text{in active mode} \\ -0.6\text{ V} & \text{in stand-by mode} \end{cases}$
2 V
$V_{Bp} = \begin{cases} 2\text{ V} & \text{in active mode} \\ 4\text{ V} & \text{in stand-by mode} \end{cases}$
V_{in}
V_{out}
Substrate Bias Control Circuit
$V_{Tn} = \begin{cases} 0.2\text{ V} & \text{in active mode} \\ 0.6\text{ V} & \text{in stand-by mode} \end{cases}$
$V_{Bn} = \begin{cases} 0\text{ V} & \text{in active mode} \\ -2\text{ V} & \text{in stand-by mode} \end{cases}$

Fig. A variable-threshold CMOS (VTCMOS) inverter circuit. The threshold voltages of NMOS and PMOS transistors are increased by adjusting the substrate bias voltage, in order to reduce subthreshold leakage currents in the stand-by mode.

v.) Multiple-Threshold CMOS (MTCMOS) Circuits

Another technique which can be applied for reducing leakage currents in low voltage circuits in the stand-by mode is based on using two different types of transistors (both NMOS and PMOS) with two different threshold voltages in the circuit. Here, low-V_T transistors are typically used to design the logic gates where switching speed is essential, whereas high-V_T transistors are used to effectively isolate the logic gates in stand-by and to prevent leakage dissipation. The generic circuit structure of the MTCMOS logic gate is shown in Fig.



V_{DD}
stand-by
high - V_T pMOS
prevents subthreshold leakage in stand-by mode
CMOS Logic with low V_T
high-speed operation with low power consumption
$\overline{\text{stand-by}}$
high - V_T nMOS
prevents subthreshold leakage in stand-by mode

Fig. Generic structure of a multiple-threshold CMOS (MTCMOS) logic gate.

2. Logic and gate level

Logic design was once the primary abstraction level where automatic design synthesis begins. With the advance of logic synthesis tools and structured VLSI design practice today, logic design is seldom performed manually. However, logic analysis is still the basis of VLSI chip design. A strong foundation in logic design is the key to produce high quality chips.
Even with the use of text-based hardware description language design style, there are still many techniques for the VLSI designer to reduce power at the logic level. The most prevalent theme in logic level power optimization techniques is the reduction of switching activities. Switching activities directly contribute to the charging and discharging capacitance, and the short-circuit power. The choice of logic encoding, data representation and Boolean function implementation has significant effects on the power dissipation of a digital circuit.

i.) Signal Gating

Signal gating refers to a class of general techniques to mask unwanted switching activities from propagating forward, causing unnecessary power dissipation. Most signal gating techniques are applied at the logic level because switching activities of the signals can be easily analyzed.
There are many different ways to implement signal gating.
The simplest method is to put an AND/OR gate at the signal path to stop the propagation of the signal when it needs to be masked.

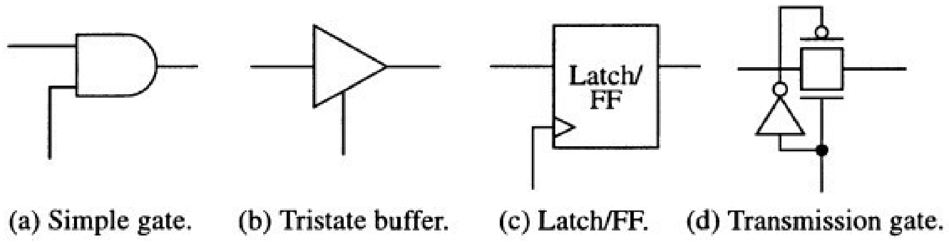


Fig. Various logic implementations of signal gating.

ii.) Logic Encoding

The logic designer of a digital circuit often has the freedom of choosing a different encoding scheme as long as the functional specification of the circuit is met. For example, an 8-bit counter can be implemented using the Binary counting sequence or the Gray code sequence. Different encoding implementations often lead to different power, area and delay trade-off. Usually, encoding techniques require the knowledge of signal statistics in order to make design decisions. This section discusses some techniques for using different logic encoding to achieve low power consumption.

Example: Binary versus Gray Code Counting

Table. Binary and Gary code counting sequences.

Binary code		Gray code	
Sequence	No. toggles	Sequence	No. toggles
000	3	000	1
001	1	001	1
010	2	011	1
011	1	010	1
100	3	110	1
101	1	111	1
110	2	101	1
111	1	100	1

Since power dissipation is related to toggling activities, a Gray counter is generally more power efficient than a Binary counter.

iii.) State Machine Encoding

The state transition graph is a functional description of a machine specifying the inputs and outputs of the machine under a particular state and its transition to the next state. The very first step of a state machine synthesis process is to allocate the state register and assign binary codes to represent the symbolic states. This process is called the encoding of a state machine. The encoding of a state machine is one of the most important factors that determine the quality (area, power, speed, etc.) of the gate-level circuit

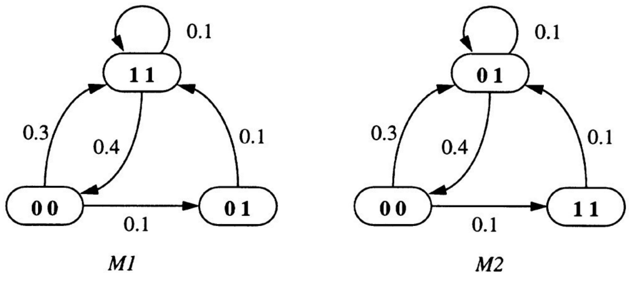


Fig. Functionally identical state machines with different encoding.

Expected state transitions:

$$E[M_1] = 2(0.3+0.4) +1(0.1+0.1) = 1.6$$

$$E[M_2] = 1(0.3+0.4+0.1) +2(0.1) =1.0$$

machines with lower E [M] are more power efficient because:

- fewer transitions of the state register lead to lower power dissipation. And
- fewer transitions are propagated into the combinational logic of the machine.

However, a state encoding with the lowest E [M] may not be the one that results in the lowest overall power dissipation. The reason is that the particular encoding may require more gates in the combinational logic, resulting in more signal transitions and power.