Chapter 2

Related Work on MTCMOS and Our Motivation

In this Chapter, we will review the previous work on sizing the sleep transistor in MTCMOS circuits. Our motivation for this work will also be presented.

During the active mode, the sleep transistor can be approximated very closely by a linear resistor R as shown in Figure 2.1[8]. When gates are discharged, this structure generates a finite voltage drop equal to $(I \times R)$ across the virtual ground where I is the current flowing through the sleep transistor.

Voltage drop on the virtual ground reduces the driving capability of gates and slows down the logics. Therefore, to design the sleep transistor size so as to retain required performance can be formulated as to control the amount of current " I " flowing through the sleep transistor. Hence, the worst case

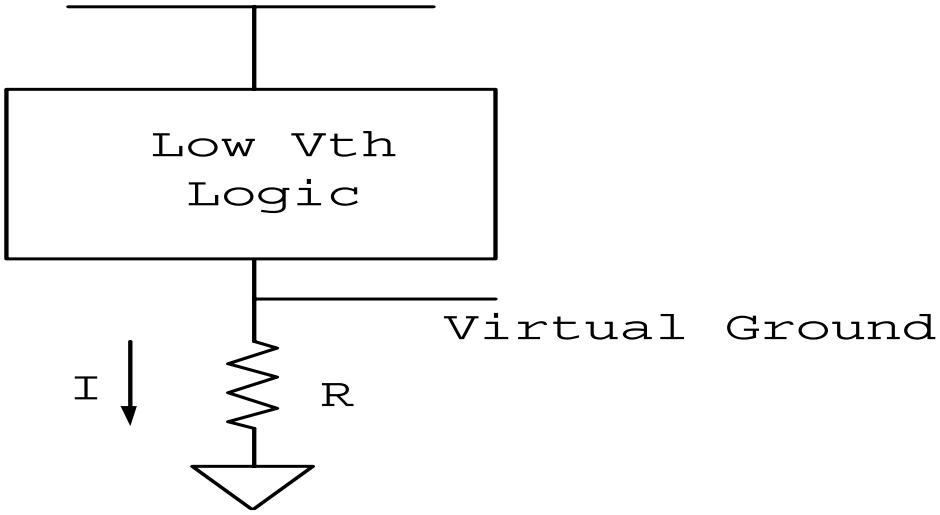

Figure 2.1: Sleep transistor modeled as resistor

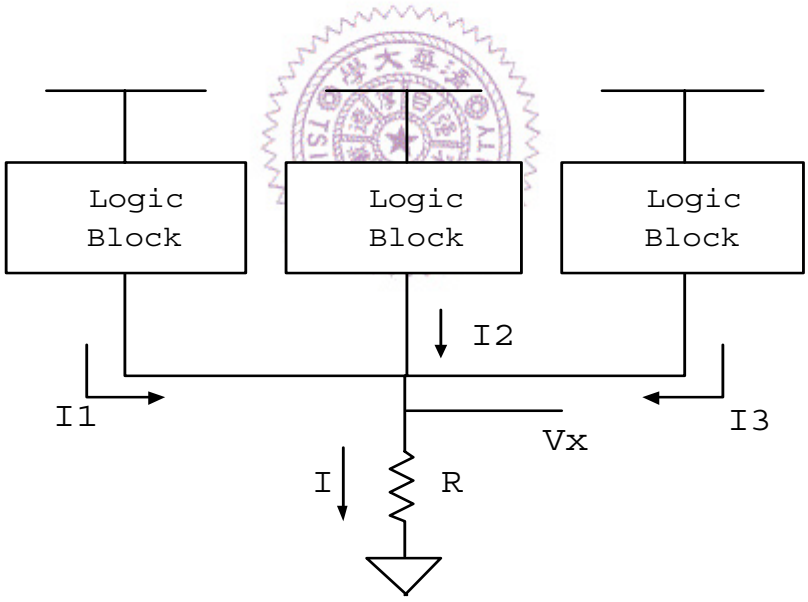

Figure 2.2: Discharge scenario

delay in an MTCMOS circuit is strongly dependent on the discharge patterns of internal gates. The most direct but difficult way to correctly size the sleep transistor of an MTCMOS circuit is to exhaustively simulate for the worst case input vector and to ensure that the worst case delay meets a fixed performance constraint. In [8], a switch level simulator has been proposed to provide fast MTCMOS simulations to reduce the search space.

On the other hand, designing the sleep transistor size based on mutual exclusive discharge patterns rather than worst case input pattern is another approach. Consider Figure 2.2. The worst case scenario takes place if all the gates supported by one sleep transistor are switching at the same time. The sleep transistor must be sized up enough to sustain the sum of the discharge current of gates ($I = I1+I2+I3$). However, if the gates are discharged mutually exclusive, the sleep transistor is only required to be sized to satisfy the maximum current of these gates ($I = \max\{I1, I2, I3\}$). Therefore, based the above example, how to correctly choose gates connected to sleep transistors is the key step of this approach.

In [9], cascaded gates on a path are clustered together to share one sleep transistor. This approach observed that gates on a path must switch at different time and hence these gates are mutually exclusive. Based on unit delay model, this approach is effective for balanced circuits such as tree

structures. However, the approach will over-size sleep transistors for circuits with complicated interconnections and unbalanced structures.

In [10], gates are not clustered with exclusive discharged patterns. Instead, gates are allowed to be clustered with partially overlapping discharge currents. The peak current value of gates and switching time of gates as well as its duration are monitored. This paper presents two techniques for efficient gate clustering in MTCMOS circuits by modeling the problems as Bin-Packing and Set-Partitioning problems. Results showed that this approach effectively decreased the size of sleep transistors in MTCMOS circuits.

One disadvantage of these approaches is that sleep transistors are over-sized because only topologies of circuits are considered. If functionality of circuits is considered, more gates can be found not to make transitions at the same time and hence the size of sleep transistors can be reduced. Take the circuit in Figure 2.3 as an example. Assuming that unit delay is used and one unit-size sleep transistor is used for gates that do not make transitions at the same time. The numbers list above each gate stand for possible switching times of each gate during one clock cycle. It can be seen that gates may make transitions at time slot one, two and three. Among these three time slots, all gates may make transitions at time slot one. So, all gates may make transitions during the same clock cycle. Hence, by using approaches considering

only topology, six sleep transistors are required. However, looking carefully, we can see that some gates can not be discharged simultaneously during the same clock cycle. For example, during the T_{th} clock cycle, to make $g1$ discharge at the first time slot, primary input b must make a $0 \rightarrow 1$ transition from the $(T-1)_{th}$ clock cycle to the T_{th} clock cycle. However, to make $g2$ discharge at the first time slot during the T_{th} clock cycle, primary input b must set to 1 on the $(T-1)_{th}$ clock cycle. Obviously, there is a conflict that $g1$ and $g2$ can not be discharged at the first time slot during clock cycles. Moreover, $g5$ and $g6$ are both the fan-outs of $g4$ and have the same possible transition time slots. During the T_{th} clock cycle, to make $g5$ discharge at the first time slot, the output signal of $g4$ must be 1 on the $(T-1)_{th}$ clock cycle. This causes a conflict with the output signal of $g4$ that must be 0 on the $(T-1)_{th}$ clock cycle when $g6$ is discharged at the first time slot during the T_{th} clock cycle. Similarly, to make both $g5$ and $g6$ discharge at the second and third time slot during the T_{th} clock cycle, there is a conflict between signal transitions of $1 \rightarrow 0$ and $0 \rightarrow 1$ from the $(T-1)_{th}$ to T_{th} clock cycle at the output signal of $g4$. Hence, $g5$ and $g6$ are discharged mutually exclusive. Therefore, taking functionality into consideration, we only need four sleep transistors for gate clusters $\{g1, g2\}$, $\{g3\}$, $\{g4\}$, and $\{g5, g6\}$ rather than six. In this thesis, based on the above observations, we will take advan-

tage of the functionality of circuits to efficiently cluster gates to share sleep transistors.

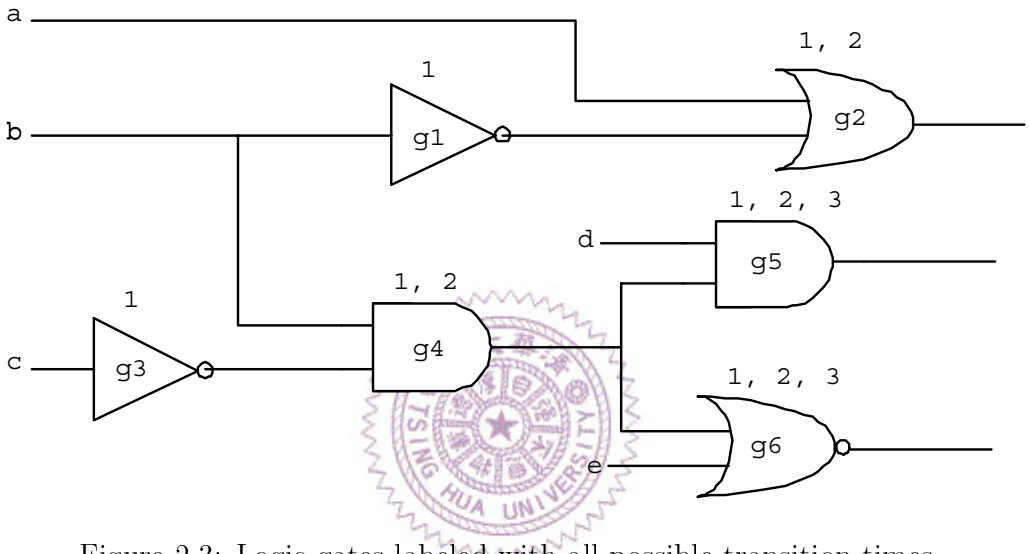

Figure 2.3: Logic gates labeled with all possible transition times