IMPLEMENTATION OF VARIOUS LOW POWER TECHNIQUES IN A CMOS SRAM CELL

Dr. R. Prakash Rao
Associate Professor, Electronics and Communication Engineering,
Matrusri Engineering College, #16-1-486, Saidabad, Hyderabad-500059, India.

Abstract:

This work explores the new low power approaches for VLSI logic. While designing a VLSI system power dissipation is one of the major concerns. Up to a certain time dynamic power was the single largest concern; however as the technology feature size shrinks static power has become an important issue as dynamic power. A well-known previous technique called the sleep transistor technique cuts off V_{dd} and/or Gnd connections of transistors to save leakage power consumption. However, when transistors are allowed to float, a system may have to wait a long time to reliably restore lost state and thus may experience seriously degraded performance. Therefore, retaining state is crucial for a system that requires fast response even while in an inactive state. **The two common approaches are sleepy stack and sleepy keeper. Both methods are excellent in this regard. The static and dynamic power of sleepy stack is considerably low. But it has a delay penalty and its area requirement is maximum compared with other processes. Again the sleepy keeper process possesses excellent speed criteria but it requires more static and dynamic power than sleepy stack.** Our goal is to trade off between these limitations and thus propose new methods which reduce both leakage and dynamic power with minimum possible area and delay trade off.

Key words: VLSI system, power dissipation, static power, dynamic power, sleep transistor technique, sleepy keeper process.

1. INTRODUCTION :

For the most recent CMOS feature sizes (e.g., 90nm and 65nm), leakage power dissipation has become an overriding concern for VLSI circuit designers. International technology roadmap for semiconductors (ITRS) [1] reports that leakage power dissipation may come to dominate total power consumption. Power consumption of CMOS consists of dynamic and static components. Dynamic power is consumed when transistors are switching, and static power is consumed regardless of transistor switching. Dynamic power consumption was previously (at 0.18µ technology and above) the single largest concern for low-power chip designers since dynamic power accounted for 90% or more of the total chip power. Therefore, many previously proposed techniques, such as voltage and frequency scaling, focused on dynamic power reduction. However, as the feature size shrinks, e.g., to 0.09µ and 0.065µ, static power has become a great challenge for current and future technologies.

There are many reasons for which power losses occur in CMOS circuit. Figure 1 shows different types of leakage components. They are:

1. Sub-threshold leakage (weak inversion current)
2. Gate oxide leakage (Tunneling current)
3. Channel punch through
4. Drain induced barrier lowering

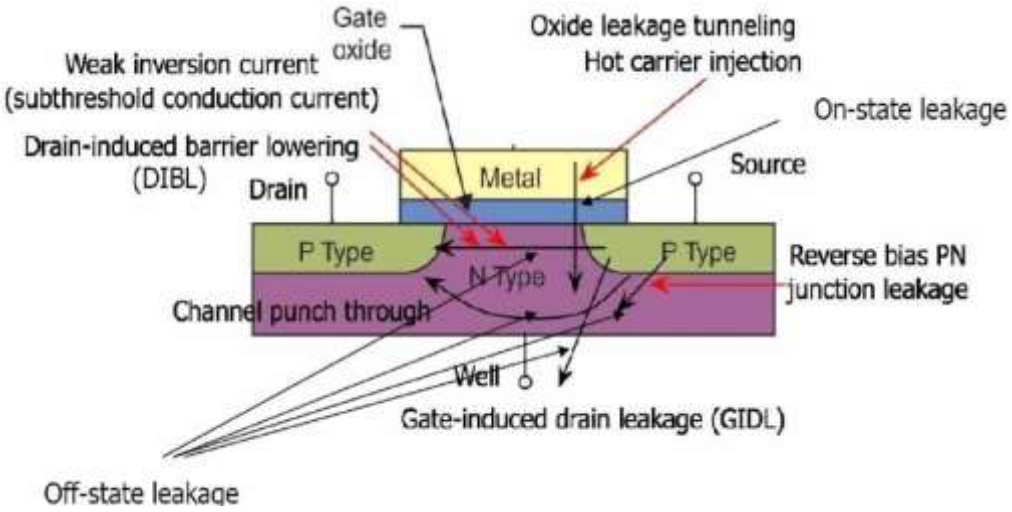

Figure 1: Leakage power components in CMOS

2. PREVIOUS APPROACHES

In this section we analyze the previous approaches which are closely related to our research. Here we discuss previous low power technique that primarily target for reducing leakage. This technique for leakage reduction can be grouped into two categories: (i) State saving (ii) state destructive [1]. State save can have an advantage over the state destructive.

The approaches that are adopted in VLSI design are -

2.1 Base Case

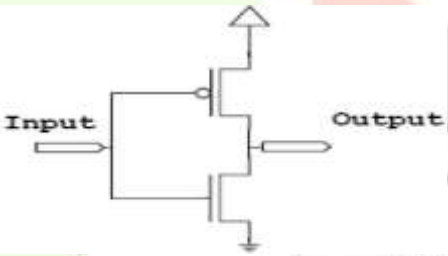

Figure 2.1: Base Case

The base case circuit contains only the PMOS network and the NMOS network and there exists no method to reduce leakage. A base case inverter is shown Figure 2.1. It is a state- saving technique and minimum area requirement.

2.2 Sleep Transistor Technique

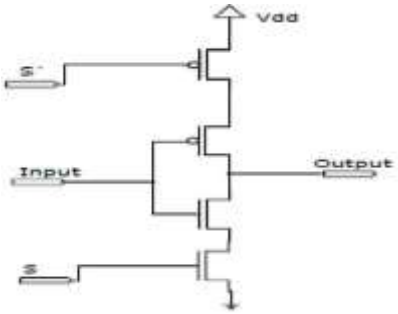

Figure 2.2: Sleep Transistor Technique

State -destructive techniques cut off transistor (pull-up or pull-down or both) networks from supply voltage or ground using sleep transistors. These types of techniques are also called gated V_{dd} and gated-GND (note that a gated clock is generally used for dynamic power reduction). Mutoh et al. propose a technique they call Multi-Threshold- Voltage CMOS (MTCMOS) [2], which adds high-V_{th} sleep transistors between pull-up networks and V_{dd} and between pull-down networks and ground as shown in Figure 3.2 while logic circuits use low- V_{th} transistors in order to maintain fast logic switching speeds. The sleep transistors are turned off when the logic circuits are not in use. By isolating the logic networks using sleep transistors, the sleep transistor technique dramatically reduces leakage power during sleep mode. However, the additional sleep transistors increase area and delay. Furthermore, the pull-up and pull-down networks will have floating values and thus will lose state during sleep mode. These floating values significantly impact the wakeup time and energy of the sleep technique due to the requirement to recharge transistors which lost state during sleep.

2.3 Forced stack

Another technique to reduce leakage power is to stack the transistors. Figure 2.3 shows a forced stack inverter. The effect of stacking the transistor results in the reduction of sub- threshold leakage current when two or more transistors are turned off together.

Figure 2.3: Forced stack inverter

The stacking effect [3] can be understood from the forced stack inverter shown in figure 2.3. In the generic inverter there are only two transistors. But here in case of forced stack inverter two pull up transistors and two pull down transistors are used. All inputs share the same input in the forced stack circuit. If the input is '0', then both transistor M1 and M2 are turned off. Here V_x is the intermediate node voltage. Transistor M2 has its internal resistance. Due to this resistance V_x is greater than the ground potential. This positive V_x results in a negative gate-source (V_{gs}) for the M1 transistor and the negative source-base voltage (V_{sb}) for M1. Here M1 also has a reduced drain-source voltage (V_{ds}), which lower the drain induced barrier lowering (DIBL) effect. All transistors are getting the same input. So this forced stack technique is a state saving technique. That means when the circuit is in OFF mode it saves the current state. The main drawback of this forced stack inverter is that it can not use the high V_{th} transistor. Because if it use the high V_{th} transistor than there is a dramatic increase of delay. This delay increase is 5X larger than the conventional CMOS.

2.4 Sleepy stack approach

In the sleepy stack structure the forced stack and the sleep transistor techniques are combined together [4]. Hence the names sleepy stack. Figure 2.4 shows a sleepy stack inverter.

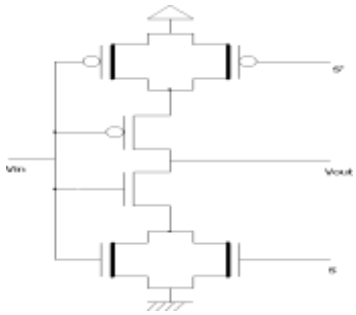

Figure 2.4: Sleepy stack inverter

The sleepy stack inverter in figure 2.4 uses the aspect ratio W/L = 3 for the pull up transistors and W/L =1.5 for the pull down transistors. At the same time the conventional inverter with the same input capacitance uses the aspect ratio W/l = 6 for the pull up transistors and W/L = 3 for the pull down transistors. Here $\mu_n = 2\mu_p$ is assumed. The sleep transistor and the stacked transistor in each network are made parallel. Here the width of the sleep transistors is reduced. Changing the width of the sleep transistors may provide additional tradeoffs between delay, power and area. The activity of the sleep transistors in sleepy stack is same as the activity of the sleep transistors in the sleep transistor technique. The sleep transistors are turned on during active mode and turned off during sleep mode. The sleepy stack structure can reduce the circuit delay in two ways. First, since the sleep transistors are always on during active mode so there is always a current flow through the circuit. That's why it gives a faster switching time than the forced stack structure. The high V_{th} transistors are used for the sleep transistor and the transistors parallel to the sleep transistor without incurring large delay increase. The delay time is increasing here but it gives low leakage. During sleep mode both the sleep transistors are turned off. But the sleepy stack structure maintains exact logic state. As high V_{th} transistor is used here so the leakage power is suppressed. The stacked transistors also suppressed the leakage power consumption. So sleepy stack structure achieves ultra low leakage power consumption during sleep mode while retaining the exact logic state. But the main drawback of this sleepy stack technique, however, is increasing area a lot.

2.5 Sleepy keeper approach

In the traditional CMOS design the NMOS are placed always at the pull down network because it is well known that NMOS transistors are not efficient at passing V_{dd} shown in Figure 2.5. On the other hand PMOS transistors are placed at the pull up network because PMOS transistors are not efficient at passing GND.

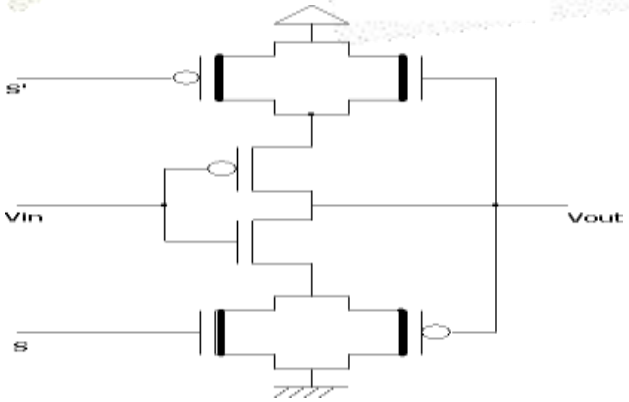

Figure 2.5: Sleepy keeper approach

Let us maintain a value of '1' in sleep mode and assume that the value has already been calculated. The sleepy keeper [5] circuit in figure 2.5 uses this output value of '1' and an NMOS transistor maintains this value during sleep mode. An additional NMOS transistor is added in parallel to the pull up sleep

transistor connected to V_{dd}. At sleep mode this NMOS transistor is the only source of V_{dd} to the pull-up network since the sleep transistor is off. Similarly, to maintain a '0' value, assume that the value is already calculated. The sleepy keeper approach uses this output value of '0' and a PMOS transistor maintains the value during sleep mode.

An additional PMOS transistor is added in parallel to the pull down sleep transistor connected to GND. At sleep mode this PMOS transistor is only source of GND the pull down network since the sleep transistor is off. In 0.07 um technology the sleepy keeper approach (with dual V_{th}) achieves 2290X leakage reduction over the base case and 175X leakage reduction over the stack approach. The result is similar to the best previous leakage reduction technique with state saving, sleepy stack, but in sleepy keeper the delay is less than the sleepy stack. The drawbacks of sleepy keeper is that it consumes 31% more dynamic power than the sleepy stack with single V_{th} and 41% more dynamic power with dual V_{th}. The dynamic power increases around 15% over the base case and 10% over the sleep transistor approach. The area usage by sleepy keeper increase 93% over the base case but 49% smaller area usage than sleepy stack.

2.6 Dual Sleep approach

Sleep transistors are crucial part in any low leakage power design. In dual sleep method is shown in figure 2.6, two sleep transistors in each NMOS or PMOS block are used. One sleep transistor is used to turn on in ON state and the other one is used to turn on in OFF state. Again in OFF state a block containing both PMOS and NMOS transistors are used in order to reduce the leakage power.

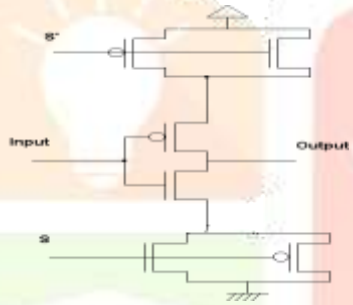

Figure 2.6: Dual sleep method

Like the sleep, sleepy stack and sleepy keeper approaches, dual V_{th} technology can be applied in dual sleep approach to obtain greater leakage power reduction . Since high- V_{th} results in less leakage but lowers performance, high-V_{th} is applied only to leakage reduction transistors, which are sleep transistors, and any transistors in parallel to the sleep transistors.

Dual sleep approach uses the advantage of using the two extra pull-up and two extra pull-down transistors in sleep mode either in OFF state or in ON state. It uses two pull-up sleep transistors and two pull-down sleep transistors. When S=1 the pull down NMOS transistor is ON and the pull-up PMOS transistor is ON since S'=0. So the arrangement works as a normal device in ON state. During OFF state S is forced to 0 and hence the pull- down NMOS transistor is OFF and PMOS transistor is ON and the pull-up PMOS transistor is OFF while NMOS transistor is ON. So in OFF state a PMOS is in series with an NMOS both in pull-up and pull-down circuits which is liable to reduce power.

3. PROPOSED METHODS

Forced sleep approach, Stack sleep approach and Variable body biasing techniques are successfully implemented in SRAM cell.

3.1 Forced sleep application in SRAM cell

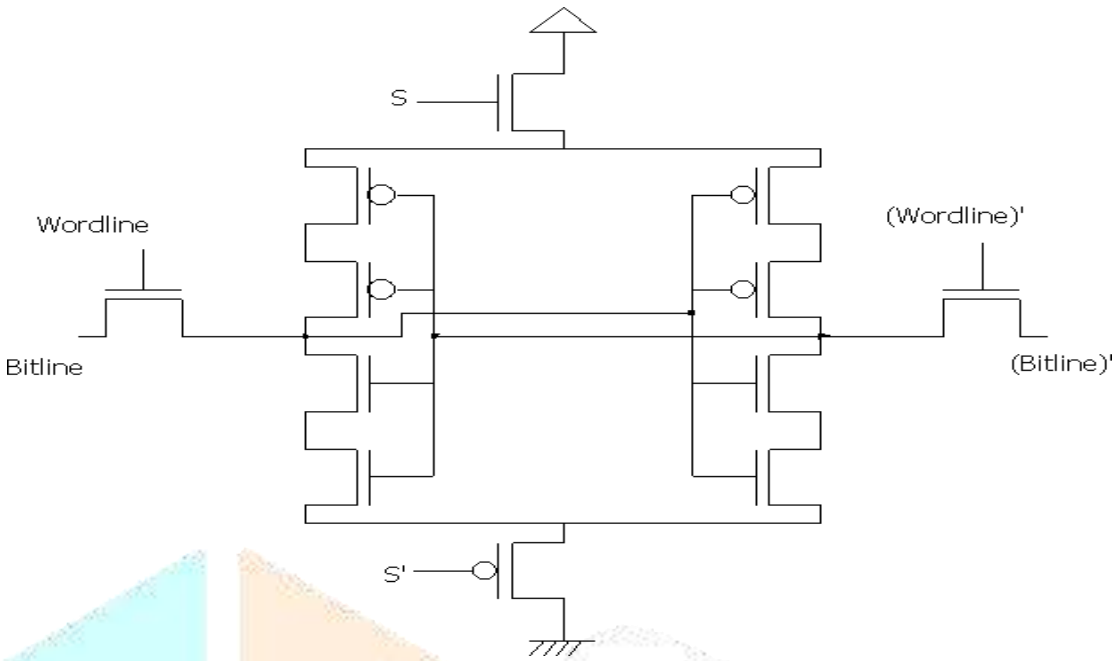

Figure 3.1: SRAM cell using forced sleep method

3.2 Stacked sleep application in SRAM cell

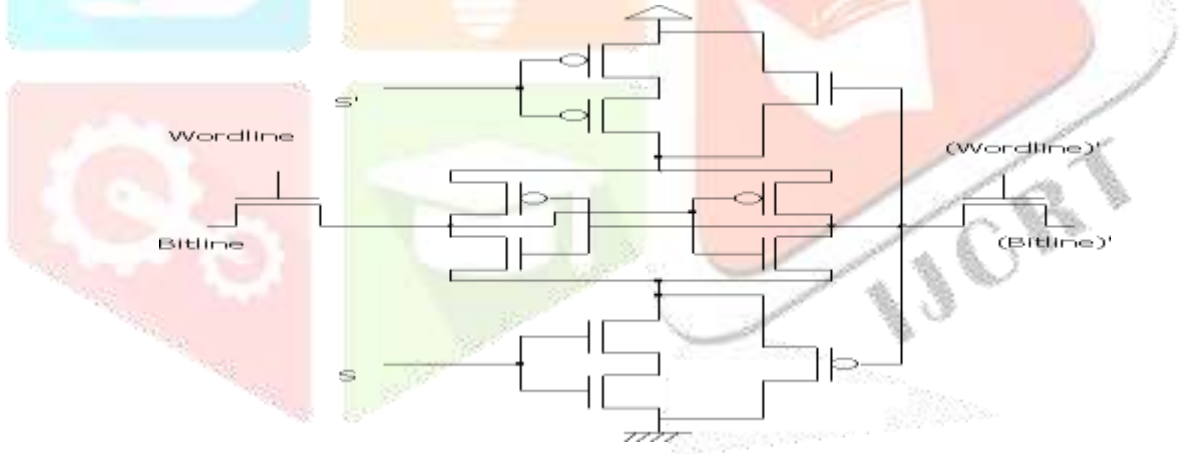

Figure 5.2.2: SRAM cell using stacked sleep approach

3.3 Variable body biasing application in SRAM cell

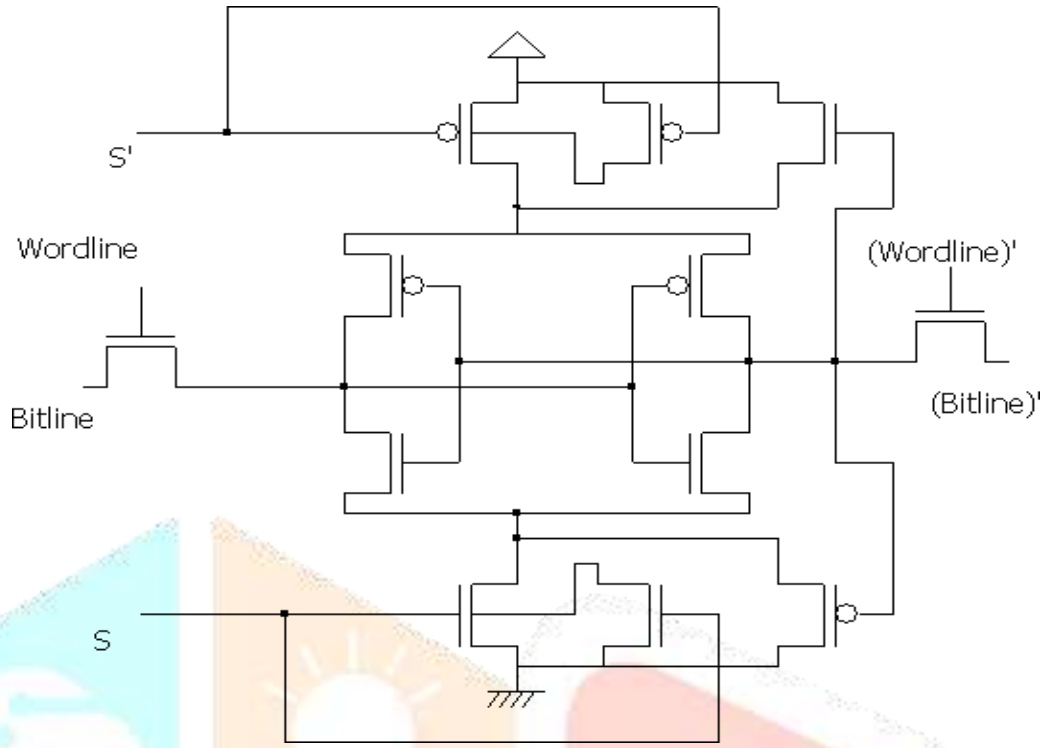

FIGURE 5.2.3: SRAM CELL USING VARIABLE BODY BIASING TECHNIQUE

4 . Simulation results for SRAM cell

The simulation results for SRAM cell with 32 nm technology is given below in table 3.

Table 3: Data for 32 nm technology:

Method	**Propagation delay (s)**	**Static power (w)**	**Dynamic power (w)**	**Area (um^2)**
Stacked sleep	1.3672E-10	3.7760E-06	1.2236E-06	10.0
Variable body biasing	1.2966E-10	2.0018E-06	1.9816E-06	10.0
Forced sleep	1.3284E-10	7.5519E-07	4.5671E-08	9.97

5.Conclusion

The propagation delay is more for the Stacked sleep compared with Variable body biasing and forced sleep but the static power is less to the Variable body biasing compared with the stack sleep and forced sleep. At the same time when compared with the dynamic power , it is less compared with the remaining two. When comparing the area forced sleep has occupied less area.

References:

[1] Linda E. M. Brackenbury, "Design of VLSI Systems- A Practical Introduction",

Department of Computer Science, Manchester University.

[2] KIM, C. and ROY, K., "Dynamic Vt SRAM: a Leakage Tolerant Cache Memory for Low Voltage Microprocessors," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 251–254, August 2002.

[3] KIM, N., AUSTIN, T., BAAUW, D., MUDGE, T., FLAUTNER, K., HU, J., IRWIN, M., KANDEMIR, M., and NARAYANAN, V., "Leakage Current: Moore's Law Meets Static Power," IEEE Computer, vol. 36, pp. 68–75, December 2003.

[4] CHANDRAKASAN, A. P., SHENG, S., and BRODERSEN, R. W., "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, April 1992.

[5] JOHNSON, M., SOMASEKHAR, D., CHIOU, L.-Y., and ROY, K., "Leakage Control with Efficient Use of Transistor Stacks in Single Threshold CMOS," *IEEE Transactions on VLSI Systems*, vol. 10, no. 1, pp. 1–5, February 2002.