Design and Application of Multimodal Power Gating Structures

Ehsan Pakbaznia and Massoud Pedram
University of Southern California
E-mail: {pakbazni,pedram}@usc.edu

Abstract - Designing a power-gating structure with high performance in the active mode and low leakage and short wakeup time during standby mode is an important and challenging task. This paper presents a tri-modal switch cell that enables implementation of multimodal power gating, including active, data-retentive drowsy, and deep sleep modes. A circuit realization and design methodology are presented that allow one to take advantage of the ultra low leakage deep sleep mode, low leakage, but very fast wakeup, drowsy mode, and an additional low leakage data-retentive mode. Experimental results demonstrate the benefits of this new switch and corresponding power gating technique.

Keywords

Low power, leakage, power gating, MTCMOS, tri-modal switch, multimodal power gating.

1. Introduction

Multi-threshold CMOS (MTCMOS) technology provides a simple and effective power gating structure by utilizing high speed, low Vt (LVT) transistors for logic cells and low leakage, high Vt (HVT) devices as sleep transistors. Sleep transistors disconnect logic cells from the supply and/or ground to reduce the leakage in the standby mode. More precisely, MTCMOS uses low-leakage NMOS (PMOS) transistors as footer (header) switches to disconnect ground (power supply) from parts of a design in the circuit standby mode. There is a large amount of rush-thru current from the power supply to ground when an MTCMOS circuit switches from the sleep to active mode. Due to inductance of the off-chip bonding wires and parasitic inductance of the power rails, rush-thru currents can cause rather large voltage bounces in the on-chip power distribution network due to the *Ldi/dt* effect [1]. On the other hand, when an MTCMOS circuit switches from the sleep to active mode, it takes some time (wakeup latency) for the circuit to be functional and start working at its full performance level. Finally, without some kind of always-on latches, the internal state of the MTCMOS circuit is lost when it is put into the sleep mode.

Because of the large amount of rush-thru current and large wakeup latency for MTCMOS circuits, for short standby periods it is better to put the circuit into an intermediate power-saving mode (called the *drowsy mode*). The reason is that the transition latency from the drowsy to active mode (which we shall call the *ready latency*) is much less than the wakeup time of the circuit when coming out of the sleep mode. Furthermore, if designed appropriately, drowsy circuits can retain pre-standby internal state of the circuit. The downside of putting a circuit into drowsy mode is the higher amount of the leakage current compared to the case when the circuit is put into the sleep mode.

In [2], the authors propose a power gating structure to support an intermediate (drowsy) power-saving mode and the traditional sleep mode. The idea is to add a clamping PMOS transistor in parallel with each NMOS sleep transistor. By applying zero voltage to the gate of the clamping PMOS and NMOS sleep transistors, the circuit can be put in the intermediate power saving mode whereby leakage reduction and data retention are both realized. In the deep sleep mode with no data retention, the gate of the PMOS transistor is connected to V_{DD} while the NMOS sleep transistor is turned off. In this approach, similar to other MTCMOS techniques, the sleep signal is generated by an always-on buffer. To have shorter wakeup times, the sleep buffer uses LVT devices. Therefore, this approach suffers from the high drowsy leakage current due to using always-on buffers. In Section 2 we will see that sleep buffer can also be power-gated during the drowsy mode, and thus, its leakage may be reduced.

The work in [3] describes multiple power modes for the circuit, but it needs multiple supply voltages (stable reference voltages to drive the gate terminal of the sleep transistor which operates in different points of the subthreshold conduction region during the sleep mode), which is costly. In [4], the authors propose a drowsy circuit scheme that automatically controls the degree of the drowsiness of the circuit by using a negative feedback implemented with a sleep inverter. This configuration thereby clamps the voltage level of the virtual ground node using the negative feedback loop. The problem with using this technique is that the circuit will either work in the active or drowsy mode, and the sleep mode is lost. This technique works fine for small standby periods when the circuit switches back and forth between standby and active periods frequently. However, for medium to long standby periods, the technique in [4] fails to be effective due to the large amount of leakage consumption during the long standby period.

In this paper, we present a new tri-modal switch cell that enables three different circuit modes: (i) active, (ii) sleep, and (iii) drowsy. The proposed tri-modal switch benefits from the low-leakage sleep mode and fast and low-cost mode-transition drowsy mode which is achieved by a negative feedback. The remainder of this paper is organized as follows. In Section 2 we introduce the tri-modal switch cell, its leakage equations, and its capability to retain the data in the drowsy mode. Section 3 describes tri-modal switch transistor sizing and related tradeoffs. The proposed architecture for multimode data-retentive power gating using the tri-modal switch is introduced in Section 4. Section 5 represents the results while Section 6 concludes the paper.

2. Tri-modal Switch

2.1 Switch Functionality

Figure 1 shows the proposed tri-modal switch configuration. Both HVT and LVT transistors are used in this design. We use thick lines to draw the gate plate of HVT transistors.

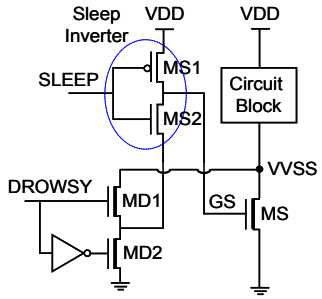

Figure 1. Implementation of the tri-mode footer cell.

Conventional footer sleep transistors use a single control input called SLEEP. As seen in Figure 1, the proposed tri-modal switch has an additional input called DROWSY. We show how this switch enables three different circuit operation modes: sleep, drowsy, or active, depending on the value of the two control signals (see Table

1 for the functionality of the tri-modal switch in terms of its input signals). When SLEEP = '0', MS1 is ON and the voltage level at GS (gate of MS) is VDD. Thus, independent of the value of the DROWSY input, the MS transistor is ON, virtual ground (VVSS) is connected to actual ground (VSS), and the circuit is in the active mode. When SLEEP = '1', the tri-modal switch operates in the sleep or drowsy mode depending on the value of the DROWSY signal. In particular, if DROWSY = '0', MS2 and MD2 will both be ON, and the output of the sleep inverter GS will be '0' which turns the sleep transistor MS OFF. In this case, the tri-modal switch cell will put the circuit in the sleep mode. If DROWSY = '1', MS2 and MD1 will be ON, creating a negative feedback between VVSS and GS nodes which puts the circuit block into the drowsy mode.

Table 1. Tri-mode switch functionality.

SLEEP/DROWSY	Tri-mode Switch Function
0X	Active
10	Sleep
11	Drowsy

Unlike the conventional power-gating techniques, the sleep inverter in the tri-modal switch cell is power gated through the MS transistor during the sleep mode, thus it has low leakage. In addition the drowsy signal changes only when we make a transition from the sleep to drowsy or vice versa which means that the drowsy signal need not be fast. Therefore, the always-on drowsy inverter shown in Figure 1 can be implemented using HVT devices to lower the leakage. The transistor-count overhead of the proposed tri-modal switch is only four: MD1, MD2, and the two transistors inside the drowsy inverter. The two transistors inside the sleep inverter, MS1 and MS2, are already used by all other power gating structures. In Section 3 we shall see that all these additional transistors are all minimum sized independent of the circuit block or the sleep transistor size, therefore, the actual area overhead of these additional transistors is quite small.

2.2 Leakage Equations

There is a sneak leakage path from the VVSS to VSS nodes of the tri-modal switch of Figure 1 in both sleep and drowsy modes due to presence of MD1 or MD2. In the sleep mode, MD2 is ON and the sneak path goes through MD1 which operates in the sub-threshold region whereas in the drowsy mode, MD1 is ON and the sneak path passes through MD2 which operates in the sub-threshold region. To minimize leakage of these sneak paths, MD1 and MD2 must be HVT transistors. To calculate the final voltage level of the VVSS node in the sleep and drowsy modes, ignoring the gate leakage, we write a KCL equation for leakage components at the VVSS node.

We use the transistor sub-threshold leakage equation [5]:

$$I_{sub} = Ae^{\frac{q}{nkT}(V_{GS}-V_{TH}+\eta V_{DS})}\left(1-e^{-\frac{qV_{DS}}{kT}}\right) \text{ with } A=\mu_0 C_{ox}\frac{W}{L_{eff}}\left(\frac{kT}{q}\right)^2 e^{1.8} \quad (1)$$

In this equation V_{GS}, V_{DS}, and V_{TH} denote the gate-source, drain-source, and the (body-affected) threshold voltages of the transistor, respectively; η is the DIBL (Drain Induced Barrier Lowering) coefficient representing the effect of V_{DS} on the threshold voltage; C_{ox} is the gate oxide capacitance per unit area; μ_0 is the zero-bias carrier mobility; and n denotes the sub-threshold swing coefficient of the transistor.

During the sleep mode, SLEEP = '1', DROWSY = '0', MS2 and MD2 are ON, and MD1 and MS are in the sub-threshold region. In this case, if we assume the voltage level of the VVSS node is V_X, the KCL equation at VVSS yields:

$$I_{leak,CB}(V_X) = I_{sub,MS}(V_X) + I_{sub,MD1}(V_X) \quad (2)$$

where $I_{sub,MS}$ and $I_{sub,MD1}$ are the sub-threshold leakage currents of MS and MD1, respectively, and $I_{leak,CB}$ denotes the leakage current of the circuit block (CB). Substituting the sub-threshold leakage current from (1) into (2), we obtain:

$$I_{leak-CB}(V_X) = A_{MS}e^{\frac{q}{nkT}(-V_{TN}+\eta V_X)}\left(1-e^{-\frac{qV_X}{kT}}\right) + A_{MD1}e^{\frac{q}{nkT}(-V_{TN}+\eta V_X)}\left(1-e^{-\frac{qV_X}{kT}}\right) \quad (3)$$

In the drowsy mode, SLEEP = '1', DRWOSY = '1', MS2 and MD1 are ON, and MD2 and MS are in the sub-threshold region. In this case, if we assume the voltage level of the VVSS node is V_X, the KCL equation at VVSS yields:

$$I_{leak,CB}(V_X) = I_{sub,MS}(V_X) + I_{sub,MD2}(V_X) + I_{sub,MS1}(V_X) \quad (4)$$

where $I_{sub,MS}$, $I_{sub,MD2}$ and $I_{sub,MS1}$ are the sub-threshold leakage currents of MS and MD2 and MS1, respectively. Substituting the sub-threshold leakage current from (1) into (4), we obtain:

$$\begin{aligned} I_{leak\text{-}CB}(V_X) = {} & A_{MS}e^{\frac{q}{nkT}(-V_{TN}+(1+\eta)V_X)}\left(1-e^{-\frac{qV_X}{kT}}\right) \\ & + A_{MD2}e^{\frac{q}{nkT}(-V_{TN}+\eta V_X)}\left(1-e^{-\frac{qV_X}{kT}}\right) \\ & + A_{MS1}e^{\frac{q}{nkT}(-|V_{TP}|+\eta(V_{DD}-V_X))}\left(1-e^{-\frac{q(V_{DD}-V_X)}{kT}}\right) \end{aligned} \quad (5)$$

Now we show that the V_X value obtained for the drowsy mode is strictly smaller than that obtained for the sleep mode.

Theorem 1 Assume $W_{MD1}=W_{MD2}$. Let V_{X1} and V_{X2} denote the solutions of equations (3) and (5), respectively. Then, $V_{X1} > V_{X2}$.

Proof by contradiction: Suppose $V_{X2} \geq V_{X1}$. Since $W_{MD1}=W_{MD2}$, we have $A_{MD1}=A_{MD2}$. We can easily show that:

$$A_{MS}e^{\frac{q}{nkT}(-V_{TN}+(1+\eta)V_{X2})}\left(1-e^{-\frac{qV_{X2}}{kT}}\right) \geq A_{MS}e^{\frac{q}{nkT}(-V_{TN}+\eta V_{X1})}\left(1-e^{-\frac{qV_{X1}}{kT}}\right) \quad (6)$$

The assumption of $V_{X2} \geq V_{X1}$ will result in the following:

$$A_{MD2}e^{\frac{q}{nkT}(-V_{TN}+\eta V_{X2})}\left(1-e^{-\frac{qV_{X2}}{kT}}\right) \geq A_{MD1}e^{\frac{q}{nkT}(-V_{DD}-|V_{TP}|+(1+\eta)V_{X1})}\left(1-e^{-\frac{qV_{X1}}{kT}}\right) \quad (7)$$

We also have:

$$A_{MS1}e^{\frac{q}{nkT}(-|V_{TP}|+\eta(V_{DD}-V_{X2}))}\left(1-e^{-\frac{q(V_{DD}-V_{X2})}{kT}}\right) > 0 \quad (8)$$

Adding both sides of inequalities in (6)-(8), and comparing both sides of the result with (3) and (5) results in $I_{leak\text{-}CB}(V_{X2}) > I_{leak\text{-}CB}(V_{X1})$, but this is contradiction, because if $V_{X2} \geq V_{X1}$, we must have: $I_{leak\text{-}CB}(V_{X2}) \leq I_{leak\text{-}CB}(V_{X1})$ i.e., we have $V_{X2} < V_{X1}$. ■

Based on Theorem 1, we can argue that in the proposed tri-modal switch, the voltage level of VVSS in the drowsy mode is strictly less than that in the sleep mode.

2.3 Data Retention and Noise Stability

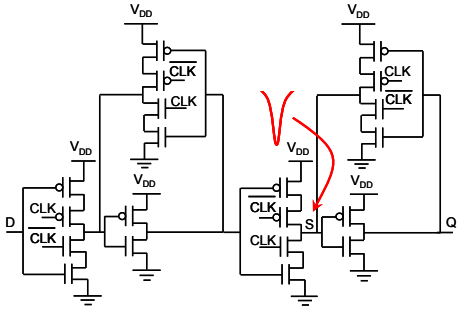

Figure 2. A DFF and negative noise applied on its internal node. Ground connection goes thru a tri-modal footer cell.

Figure 2 shows a master-slave D flip-flop (DFF). Initially the DFF is holding a logic "0' value at the Q output. In the drowsy

mode, however, this value rises to some value around 250mV with a VDD of 1.8V. Our simulations show that the VVSS voltage level in the drowsy mode is a weak function of the circuit block and is always around 250mV for this technology, which is TSMC0.18um. To assess the data stability of the DFF, a negative voltage perturbation is applied to the internal node, S, of the DFF when it is holding a logic "1' value (Q=1). Simulations also show that data in the DFF is retained for perturbations smaller than ΔV=609mV (cf. Table 2). The maximum tolerable perturbation (noise margin) for the same flip-flop when no power gating is employed is 825mV. VVSS voltage and maximum tolerable perturbation values vary under different circuit parameter variation which results in different noise stability characteristics. As Table 2 reports, the drowsy DFF shows good noise stability characteristics even under these variations.

Table 2. Stability and data retention of DFF in drowsy mode.

Type of Variation	VVSS Voltage (mV)	Peak of the Max. Tolerable Noise at Node S (mV)
No Variation	252	609
$\|V_{th}\|$+15%	237	662
$\|V_{th}\|$−15%	289	547
VDD+10%	256	629
VDD−10%	249	596

3. Transistor Sizing

Correct sizing of different transistors in the tri-modal switch is an important task since it has direct effect on various characteristics of the circuit, including logic gate switching speeds in the active mode, leakage currents in sleep and drowsy modes, wakeup latency, and area overhead. There are a number of design tradeoffs that impinge on transistor sizing for the tri-modal switch. For example, in the active mode when MS is ON, the delay of the circuit block in Figure 1 depends on the size of MS. Larger MS sizes result in higher active mode switching speeds but also increased sleep and drowsy leakage currents and lower VVSS voltage during the drowsy mode, which in turn leads to lower ground bounce and faster wakeup delays.

3.1 Active Mode Performance: Sizing MS

Power-gated circuits suffer from active-mode performance degradation due to the lower effective VDD which is due to the IR-drop on the sleep transistor in the active mode. The sleep transistor in active mode operates in its linear region, thus it can be modeled as a linear resistance. Consider using an NMOS sleep transistor (gated-ground). Each time there is high to low switching at any node in the circuit block, current flows from the node capacitance to the ground through the sleep transistor (MS in Figure 1). This discharging current causes a voltage drop between drain and source of the sleep transistor, resulting in switching speed degradation for the considered transition.

The amount of speed degradation depends on the size of the sleep transistor. The larger the sleep transistor is, the lower the switching speed degradation will be. Typically the maximum tolerable performance degradation in a power-gated design is set to 5-10% of the corresponding non-power-gated circuit. We set the maximum performance degradation to 5%. With this constraint, we size the sleep transistor MS in the tri-modal switch. The sizing technique, which is straight-forward and follows standard sleep transistor sizing techniques, is omitted. Interested readers may refer to [6][7] for sleep transistor sizing.

3.2 Wakeup Latency and Leakage: Sizing MS1

Consider a gated-ground circuit block. During the sleep period, when the sleep transistor is OFF, if the circuit block is large enough, then the VVSS node and all internal nodes in the circuit will charge to a high voltage level [8]. This is due to the higher leakage of the circuit block compared to that of the OFF sleep transistor, which eventually charges up all the internal nodes in the circuit block including the VVSS node. At the edge of the sleep to active mode transition, the sleep transistor is turned on, but the circuit block will not start working at its full speed until all extra charges are removed from internal nodes (including VVSS) through the sleep transistor. There is a wakeup latency associated with this discharging process. The wakeup latency, t_w, is defined as the delay between the time when the SLEEP signal crosses the 50% VDD level as it makes a transition to low state and the time when the VVSS node reaches 5% of the VDD level as it is discharged toward VSS.

Similarly, when the circuit in Figure 1 is put in the drowsy mode, the VVSS node is charged to a non-zero voltage level. Even though the circuit block is still functional, it will not be working at full speed. Therefore, there is a ready latency associated with a drowsy circuit that is brought into active mode. In this paper, the ready latency, t_r, is defined as the delay between the time when the SLEEP signal crosses the 50% VDD level as it falls and the time when VVSS node reaches 5% of the VDD level as it is discharged toward VSS. The wakeup and ready latencies of the circuit configuration in Figure 1 depend on sizes of MS and MS1 and voltage level of the VVSS node in the sleep/drowsy mode. The voltage level of the VVSS node in the sleep/drowsy mode is mainly determined by the size and threshold voltage value of MS. Since MS is sized when considering the active mode performance criterion (c.f. Section 3.1), the wakeup and ready latencies are determined by sizing MS1.

Suppose that we use our tri-modal switch for power-gating of a DFF. Furthermore assume that the MS transistor is already sized for 5% active performance degradation. Figure 3 shows the wakeup and ready latencies as well as the normalized leakage values in sleep and drowsy modes for different values of W_{MS1} for this positive-edge triggered DFF in TSMC0.18um. The leakage data is normalized to the active leakage of the FF when no tri-modal switch is used. As seen in the figure, the ready time is always less than the wakeup time for a fixed size of MS1. In contrast the drowsy mode leakage is always higher than the sleep mode leakage. When we increase the size of MS1 above some threshold, the wakeup and ready latencies reach some saturating values. For this example, the saturation occurs at W_{MS1}=3μm.

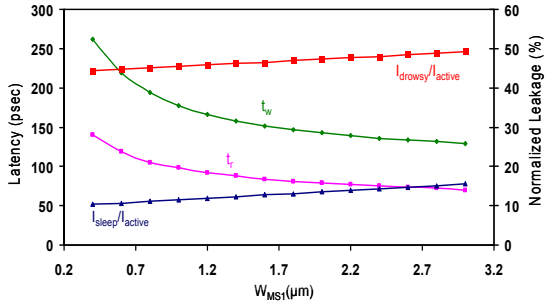

Figure 3. Leakage and wakeup/ready latencies for DFF.

Sleep and drowsy leakage currents increase linearly with W_{MS1}. To optimally size MS1, we must consider wakeup/ready latencies as well as the amount of the leakage current in the sleep/drowsy modes. We define four cost figures. They all are in the form of power-delay products (PDP): a) $PDP_{sleep\text{-}sleep}=I_{sleep}\times VDD\times t_w$, b) $PDP_{sleep\text{-}drowsy}=I_{sleep}\times VDD\times t_r$, c) $PDP_{drowsy\text{-}drowsy}=I_{drowsy}\times VDD\times t_r$,d) $PDP_{drowsy\text{-}sleep}=I_{drowsy}\times VDD\times t_w$ where I_{sleep} and I_{drowsy} denote leakage currents in the sleep and drowsy modes, respectively.

Figure 4 illustrates all four PDP's defined above for the DFF circuit. One can confirm from the figure that for all these cases, increasing W_{MS1} results in decreasing PDP until some point when PDP curve saturates at a minimum value. One may size MS1 based on any one of the PDP profiles in Figure 4; however, we use $PDP_{drowsy\text{-}sleep}$ profile to perform sizing. The reason is that the sleep

mode leakage and the drowsy mode ready latency are already small, we thus perform sizing of MS1 based on the drowsy mode leakage and the sleep mode wakeup latency. We size MS1 such that the $PDP_{drowsy\text{-}sleep}$ corresponding to this size is no more than 10% higher than the minimum (saturated) $PDP_{drowsy\text{-}sleep}$ value. In our example, this results in W_{MS1}=1.6μm.

All other transistors in the tri-modal switch cell including MS2, MD1, MD2 and transistors inside the DROWSY inverter are minimum-sized transistors. The reason is that none of these transistors has influence on the wakeup latency. Area, sleep vs. drowsy leakage currents, and energy dissipation for a mode transition are decreased by choosing minimum transistor sizes.

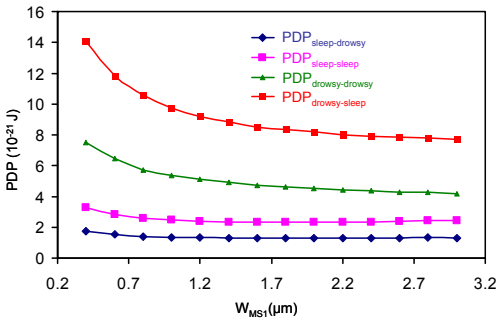

Figure 4. Different power-delay product metrics.

4. Data-Retentive Power Gating

In this section we use the tri-modal switch to realize data-retentive multimodal power gating solutions. By controlling the SLEEP and DROWSY signals for different tri-modal switches in the circuit, we can selectively put various circuit elements in different modes. Let's consider a general multi-stage pipeline circuit. We perform power gating for this structure by using the proposed tri-modal switches, where we have two different types of tri-modal switches: ones disconnecting VVSS net of the flip-flops in pipeline registers from the ground rail and those disconnecting VVSS net of the combinational logic cells in the design from VSS. This implies having two different VVSS nets: one for the flip-flops and another for the other logic cells.

4.1 Proposed Architecture

Consider a K-stage pipeline structure with K−1 pipeline registers. Suppose the design is to be implemented in a standard cell layout style. Cells fit in one of two groups: (i) sequential logic cells (FF's) belonging to pipeline registers, and (ii) combinational logic cells belonging to the pipeline logic blocks. If the pre-standby stored data in the pipeline registers is to be retained when going to sleep, the pipeline registers must be put into the data-retentive drowsy mode while the rest of the cells in the circuit are put in the sleep mode to reduce standby leakage consumption. To realize this architecture, placement of the cells in the design has to be in such a way that the VVSS rail used for pipeline FF's is separated from the VVSS rail used for combinational logic cells in the circuit. This is possible by disconnecting the VVSS rail every time a FF is placed next to a logic cell, which can cause significant breaks and reconnections in the VVSS rail. If FF's are grouped together and placed contiguously in each standard cell row, then there will be only one discontinuity in the VVSS rail of that row. However, this type of placement constraint will adversely impact the quality of the placement solution and likely increase the total wire length of the placed design.

To solve the aforesaid problem, we take the original placement of the design and modify it by moving the cells such that in each row, there are at most a few contiguous sections of FF's and a few contiguous sections of logic cells. Figure 5 shows a legal and an illegal placement. Note that in the case when we have a legal placement with a number of sections in the same row, e.g. Figure 5.(b), the virtual ground rail has to be disconnected at the point where two adjacent sections meet. Next we describe a heuristic approach to minimize the interconnection length cost associated with removing placement conflicts.

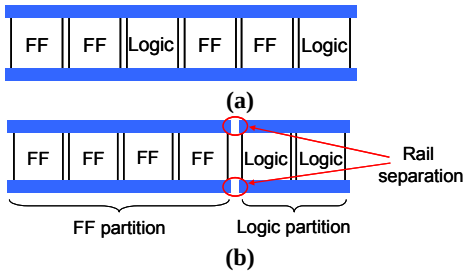

Figure 5. Examples of (a) illegal and (b) legal placements.

4.2 Placement with Row Sectioning

In a standard cell design, power-gating switch cells can be placed in different ways among the cells in a circuit. Typically, it is desirable to uniformly distribute the switch cells on each standard cell row in order to have a simple power/ground network routing strategy and minimize the worst-case (resistive) parasitic of the virtual net. Figure 6.(a) shows the so-called *column-aligned* sleep transistor placement style. The dashed boxes represent tri-modal switch cells. All other standard cells are assumed to be placed in the blank areas between the switch cells. The True VSS (TVSS) mesh lines are also shown in the figure. They are used to connect to the TVSS pins in various switch cells. With this placement style, there can be only one switch cell under each TVSS line at each row which can be used to power gate a FF section or a combinational logic section as the case may be. We have to decide which TVSS lines are used for FF's and which are used for combinational logic cells. We present a heuristic approach which modifies the original placement (c.f. Figure 6.(a)) and converts it to a legal placement while minimizing the total perturbation to the original placement by moving the FF cells in the design. Note each row is considered separately, and cell interchange between cell rows is not allowed. Also number and placement of the TVSS lines are assumed to be fixed and given.

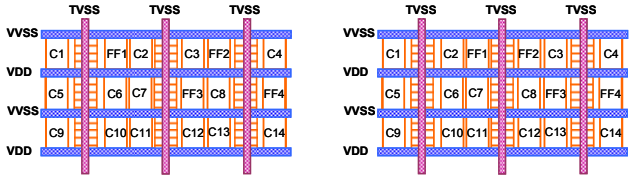

Figure 6. Column-aligned switch placement: (a) before and (b) after removing illegal placements.

Consider an already placed design obtained by any state-of-the-art placement tool. Suppose there are r rows and m TVSS lines in the design. Let's assume that we use at most n_i TVSS lines for the FF's in the i^{th} row ($n_i < m$). For each row, we have to determine: (a) the number of contiguous FF sections, and (b) the TVSS lines around which these FF sections should be placed in order to minimize the total extent of FF displacements compared to the original placement solution.

For the i^{th} row, the heuristic starts by assuming n_i=1 (if no FF lies in the i^{th} row, n_i=0 and we are done). We evaluate each of the m TVSS lines in the i^{th} row by calculating the amount of placement perturbation with respect to that line, i.e., the increase in total perturbation of the circuit when all FF's in the row are moved to new locations on the row so as to make a single contiguous section adjacent to that TVSS line. The FF's are sorted based on their distance from the target TVSS line and moved one after the other in that order. Cell overlaps are removed by pushing overlapping cells

aside to make space for the FF's. After evaluating each of the m TVSS lines, we can determine which one of them minimizes the total placement perturbation. Next we set n_i=2, and evaluate all possible *pairs* of TVSS lines by calculating the placement perturbation with respect to that pair, i.e., when all the FF's in the i^{th} row are moved to make two sections around the pair of TVSS lines. Evaluating each pair of TVSS lines starts by moving the closest FF to any of the TVSS lines in the pair under consideration, then the second closest FF, if exists, to any of the TVSS lines in the pair under consideration, and so forth. The perturbation cost is calculated as in n_i=1 case. After evaluating all possible pairs of TVSS lines, C(m,2), the best pair that results in the minimum placement perturbation is determined. We can keep increasing n_i and do evaluation to m, but the algorithm complexity will become exponential in m. Fortunately, our results show that for a design with a relatively small number of FF's compared to the total logic cell count, which is the typical case, the amount of cost reduction that is achieved by going beyond n_i=2 is negligible (c.f. Section 5).

5. Experimental Results

We designed and implemented a 16×16 pipelined Carry Save Multiplier (CSM). The circuit is divided into two pipeline stages. The 46-bit output of the first stage is latched into the pipeline registers (46 FF's). The first 16 bits out of these 46 bits, which make the least significant bits of the product, are directly passed to the output. The last 30 bits are passed to the second stage to make the most significant bits of the product.

5.1 Design Flow

We implemented the 16×16 pipelined CSM in structural Verilog. After verifying the functionality of the Verilog design, we synthesized the design by using the Synopsys Design Compiler with OSU standard cell library [9] in TSMC0.18um. V_{DD}=1.8V. We performed timing analysis on the synthesized design and achieved the worst-case stage delay of 4.1ns (clock frequency of 244 MHz). After synthesizing the design, the standard delay format (sdf) file was generated, and the design was verified with sdf back-annotation. We then used the Cadence Encounter to complete the placement of the design. We modified the placement using the row sectioning method described in Section 4.2 with n_i=2. The tri-modal switch cells were manually inserted into the design. After the placement was done, the design was routed with Cadence Encounter, timing analysis was performed and an sdf file was generated. The design was then verified again. Finally, we extracted the netlist and performed HSPICE simulations. Figure 7 shows the brief design flow that is used in this paper. Note not all the steps are shown in this figure. Figure 8.(a) depicts the design after the original placement where the FF's, which are scattered in the design, are highlighted with red boxes. Figure 8.(b) and (c) show the same design after the row sectioning technique for FF placement is applied for n_i=1 and n_i=2, respectively. Figure 8.(d) shows the routed design of Figure 8.(c), i.e., with n_i=2.

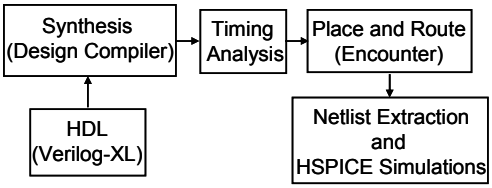

Figure 7. Summarized block diagram of the design flow.

5.2 Energy and Delay Comparisons

In this section we discuss the results that we achieved by implementing the 16×16 pipelined CSM explained in Section 5.1. Tri-modal switch cells are used to implement all the MTCMOS circuits considered in this section. We compare the leakage current, ground bounce and wakeup/ready latencies for four different cases: a) CMOS, b) MTCMOS: deep-sleep, c) MTCMOS: drowsy, and d) MTCMOS: data-retentive.

No power gating is used for the CMOS circuit and there is no constraint for placement of the FF's. During the active mode, all tri-modal switches are in the active state (SLEEP="0", DROWSY="X") in all versions of MTCMOS circuit. In the standby mode, however, tri-modal switches are put in different states: in deep-sleep MTCMOS, all tri-modal switches are in the sleep mode (SLEEP="1", DROWSY="0"), in drowsy MTCMOS all tri-modal switches are in the drowsy mode (SLLEP="1", DROWSY="1"), while in data-retentive MTCMOS, tri-modal switches used for combinational logic cells are in the sleep mode and tri-modal switches used for FF's are in drowsy mode. We compare different aspects of these four versions of the same 16×16 pipelined CSM.

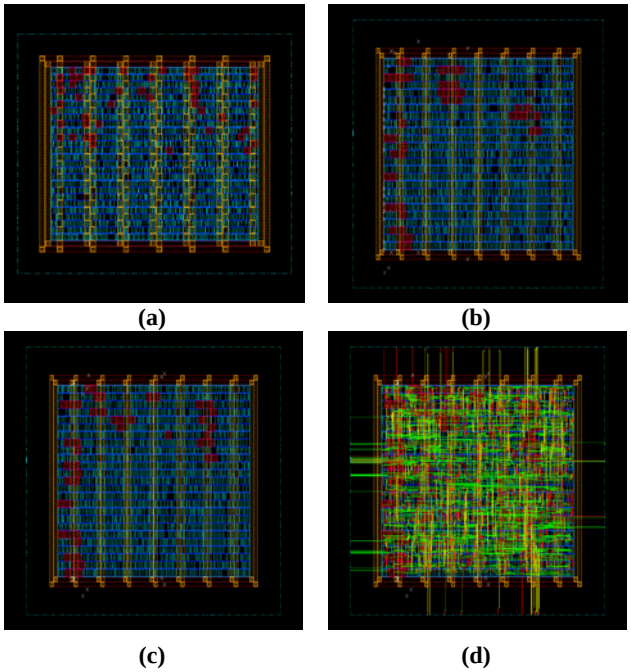

Figure 8. (a) Original placement for 16×16 pipelined CSM, (b) placed design after row sectioning with n_i=1, (c) placed design with n_i=2 (d) routed design with n_i=2.

The second to fourth columns of Table 3, respectively, show the standby leakage current, the peak value of ground bounce (GB), and the wakeup/ready (w/r) latencies for all circuit configurations explained above. It is seen from the table that the deep-sleep MTCMOS circuit has the lowest leakage among all configurations, making it the most appropriate choice for long standby periods. We note that the leakage of the drowsy MTCMOS is only 24% lower than that of the CMOS circuit, i.e., much higher than that of the deep-sleep. The ground bounce for deep-sleep circuit is much higher than that for drowsy circuit.

Table 3. Leakage, GB and w/r latency comparisons.

Circuit Type	Leakage (nA)	Ground-Bounce (mV)	Wakeup/Ready Latency (ns)
CMOS	63	-	-
Deep-Sleep	0.10	473	19.32
Drowsy	48	143	4.83
Data-Retentive	2.85	441	19.32

We assume that the maximum tolerable ground bounce is 150mV. To maintain a ground bounce value less than this threshold, we have resorted to a multi-cycle turn-on strategy similar to the one proposed in [1], where we turn on only some of tri-modal switches at each clock cycle. In particular, 7/45, 9/45, 11/45, and 18/45 portions of the tri-modal switches are turned on during the first,

second, third, and fourth consecutive clock cycles, respectively. Using this turn-on strategy, it takes 4 clock cycles to wake up the deep-sleep circuit while it only takes one clock cycle to wake up the drowsy circuit. Therefore, there is a three clock cycle penalty to wake up from deep sleep mode as compared to waking up from drowsy mode, which is done in one cycle. Now assume this multiplier is used in the execution stage of a five-stage pipelined processor, and has been put into the deep-sleep mode by the power-management unit since it had not been utilized recently. If a new instruction in the IF stage needs to use this multiplier, the processor has to be stalled for three clock cycles for this multiplier to be ready for operation. However, if the multiplier was in the drowsy mode, the processor could perform its regular operation without being stalled. The cycle penalty will increase as the size of the circuit increases.

Despite having a faster wakeup, the drowsy circuit suffers from higher leakage compared to the deep-sleep circuit. Therefore, for longer standby periods when the leakage energy dissipation becomes an issue, we may want to pay the wakeup cycle penalty to achieve low leakage dissipation. In that case, deep-sleep or data-retentive modes are more preferable than the drowsy mode. Therefore, it is important to have a power-gating structure that supports the four power modes discussed above. Figure 9 shows leakage energy versus total (wakeup) latency for the CSM circuit when it is operating for 100,000 clock cycles. We assume that 20% of the time the circuit is operating in the active mode while 80% of the time it is in the standby mode. We compare three different standby policies: (i) CMOS, (ii) MTCMOS: drowsy, and (iii) MTCMOS: deep-sleep. The energy versus total latency curves are shown for different mode transition frequency values, f_{mt}. The mode transition frequency is in the units of per million clock cycles and is defined as the number of the mode transitions that happen in one million cycles. Since we compare leakage energy and the total latency, we do not consider data-retentive circuit in this analysis.

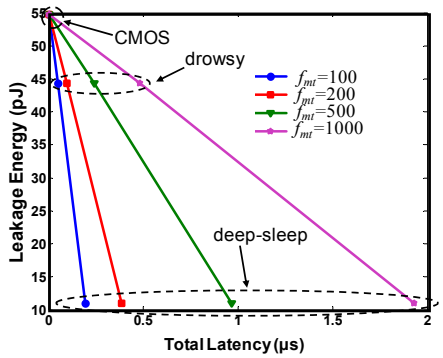

Figure 9. Leakage versus total latency for different mode-transition frequencies in the unit of per million cycles.

Table 4. Delay, area and routing comparisons.

Circuit Type	Stage delay (ns)	Cell area (um^2)	Wire length (um)	Wire length (um)
			n_i=1	n_i=2
CMOS	4.54	54720	54402.6	54402.6
MTCMOS	4.83	55710	59008.4	56077.2
Increase (%)	6.4	1.8	8.5	3.1

Table 4 compares delay, cell area and total wire length for CMOS and MTCMOS circuits. The placement modification discussed in Section 4.2, i.e., row sectioning, increases the for signal routing cost. The total wire length is reported for n_i=1 and n_i=2. It can be seen that we have a 5% reduction in total wire length when we use n_i=2 as compared to n_i=1; however, our experiments show that if we use $n_i > 2$, the total wire length reduction is negligible compared to n_i=2. For example, for the CSM design, the total wire length reduction by going from n_i=2 to n_i=4 is less than 1%. The total MTCMOS cell area increase reported in Table 4 is due to the area occupied by tri-modal switches. As seen in the table, the overall area increase is only 1.8%. Note that the sleep transistors have been sized for maximum 5-7% active delay increase compared to the (non-power-gated) CMOS circuit. We could have achieved lower MTCMOS active delay by upsizing the sleep transistor inside the tri-modal switch.

5.3 Technology Scaling

We have done similar simulations for TSMC90nm technology with VDD=1.2V to show the scalability of the proposed technique. Results are summarized in Table 5. It can also be seen from the table that the leakage current in the drowsy circuit is reduced by 77% as compared to that for the CMOS circuit. This means that leakage saving of the drowsy circuit compared to deep sleep mode becomes relatively better with technology scaling.

Table 5. Leakage comparisons for TSMC90nm.

Circuit Type	Leakage (μA)
CMOS	150
Deep-Sleep	0.6
Drowsy	35
Data-Retentive	2.35

6. Conclusion

In this paper, we presented a novel integrated circuit and architectural-level technique for general pipeline designs that allows us to benefit from very low leakage deep-sleep mode, very fast recovery drowsy mode, and an additional low leakage data-retentive mode. We described a novel tri-modal switch cell that enables us to realize this circuit architecture. We showed that the circuit can be put in a number of power-gating modes which are depending on the duration of the standby period. We also observed that the data-retentive power gating delivers low standby leakage current while storing the internal circuit state.

References

[1] S. Kim, S.V. Kosonocky, Stephen, and D.R. Knebel, "Understanding and minimizing ground bounce during mode transition of power gating structures", *Proc. Int'l Symp. on Low Power Electronics and Design*, pp. 22-25, 2003.

[2] S. Kim, S.V. Kosonocky, D. R. Knebel, and K. Stawiasz, "Experimental measurement of a novel power gating structure with intermediate power saving mode," *Proc. Int'l Symp. on Low Power Electronics and Design*, pp. 20-25, 2004.

[3] K. Agarwal, H. Deogun, D. Sylvester, K. Nowka, "Power Gating with Multiple Sleep Modes," *Proc. Int'l Symposium on Quality Electronic Design*, pp. 633 – 637, 2006.

[4] Tada, H. Notani, and M. Numa, "A novel power gating scheme with charge recycling," *IEICE Electronics Express*, no. 12, pp. 281-286.

[5] Z. Chen, L. Wei, M. Johnson, and K. Roy, "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks," *Proc. of Int. Symposium on Low Power Electronics and Design,* 1998, pp.239-244.

[6] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor Sizing Issues and Tool for Multi Threshold CMOS Technology," *Proc. Design Automation Conference*, pp. 409-414, 1997.

[7] A. Ramalingam, B. Zhang, A. Devgan and D. Pan, "Sleep transistor sizing using timing criticality and temporal currents," *Proc. ASP-DAC,* pp. 1094-1097, 2005.

[8] E. Pakbaznia, F. Fallah and M. Pedram "Charge recycling in power-gated CMOS circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,* Vol. 27, No. 10, pp. 1798-1811, Oct. 2008.

[9] http://vcag.ecen.okstate.edu/projects/scells/