A Novel Synthesis Approach for Active Leakage Power Reduction Using Dynamic Supply Gating

Swarup Bhunia, Nilanjan Banerjee, Qikai Chen, Hamid Mahmoodi, and Kaushik Roy
School of Electrical and Computer Engineering, Purdue University, West Lafayette.
{bhunias, nbanerje, qikaichen, mahmoodi, kaushik}@purdue.edu

Abstract:
Due to exponential increase in subthreshold leakage with technology scaling and temperature increase, leakage power is becoming a major fraction of total power in the active mode. We present a novel low-cost design methodology with associated synthesis flow for reducing both switching and active leakage power using dynamic supply gating. A logic synthesis approach based on Shannon expansion is proposed that dynamically applies supply gating to idle parts of general logic circuits even when they are performing useful computation. Experimental results on a set of MCNC benchmark circuits in a predictive 70nm process exhibits improvements of 15% to 88% in total active power compared to the results obtained by a conventional optimization flow.

Categories & Subject Descriptors: B.7.1 [**Integrated Circuits**]: Types and Design Styles – *supply gating, logic synthesis*

General Terms: Algorithms, Design, Performance

1. Introduction

As CMOS technology continues to scale down to achieve higher performance and higher level of integration, power dissipation is becoming a serious barrier to scaling. The power dissipation is due to both switching and leakage current and is given by:

$$P = P_{switching} + P_{leakage} = \alpha f \cdot C \cdot V_{dd}^2 + I_{leakage} \cdot V_{dd} \qquad (1)$$

where, V_{dd} is supply voltage, α is switching activity, f is the clock frequency, C is the average switched capacitance of the circuit, and $I_{leakage}$ is the average leakage current. The switching power is due to charging and discharging of circuit capacitances, and therefore, is directly proportional to the switching activity and frequency. Leakage power in bulk scaled technologies is mainly due to subthreshold leakage, gate leakage, and reverse-biased source-substrate and drain-substrate junction tunneling leakage (JT) because of halo implants [1]. Subthreshold leakage increases exponentially as the technology scales because of reduced threshold voltages (V_t) required to maintain transistor 'ON' current at reduced supply voltages. Gate leakage increases exponentially because of reduced oxide thickness required to maintain the gate control over the channel to reduce short channel effects. The reverse biased junction tunneling increases because of increased doping levels used in the halo implants to suppress Drain Induced Barrier Lowering (DIBL) and V_t roll-off [2]. Hence, leakage power is becoming a significant fraction of total power dissipation [3]. Leakage is not only important in the standby mode but also in the active mode of operation. In fact, the leakage in the active mode (active leakage) is significantly larger due to higher die temperature in the active mode and the exponential temperature dependence of subthreshold leakage [3]. Fig. 1 shows the temperature dependence of different leakage components in a predictive 50nm process [1]. Gate leakage is not temperature dependent, whereas, JT leakage is weakly dependent on temperature [2]. Therefore, in the active mode of operation (high temperature), subthreshold leakage is the dominant component of leakage. Experiments on high performance microprocessors show that more than 40% of the total power dissipation is due to leakage (both active and standby leakage) [3]. A low-power design methodology in scaled technologies, therefore, has to target both the switching and leakage components of power in the active mode of operation.

Dual V_t assignment has been used as a static method for reducing the leakage power [6]. However, dual V_t technique does not reduce the leakage on critical paths. Moreover, dual V_t assignment increases the number of critical paths in a design, degrading the design yield under process variations [9]. In dynamic leakage reduction methods, the leakage reduction techniques are applied only in the standby mode. These methods include input vector control, dynamic body biasing, and supply gating [4, 5, 6]. Input vector control uses the state dependence of leakage to apply best input vector to the circuit in the standby mode [6]. However, input vector control can be ineffective because it may not be possible to force all logic gates to their best leakage state by controlling the state of primary inputs. Dynamic body biasing applies forward (or zero) body bias in the active mode to achieve high performance and an optimal reverse body bias in the standby mode to minimize leakage. The technique becomes less effective with technology scaling since the optimal reverse body bias becomes closer to zero body bias as technology scales [7]. Moreover, body bias does not reduce gate leakage. Dual-VDD and dynamic voltage scaling are used for power reduction without impacting system performance [4]. However, dual-VDD requires extra supply voltage and is not applicable in performance critical circuits. Dual-VDD also results in more critical paths in a design, which adversely affects the design yield under parameter variations. Dynamic voltage scaling suffers from large energy and transition delay overhead for changing the supply voltage. Supply gating has been proposed and used as a method to reduce standby leakage current [4, 5]. The idea is to disconnect the global supply voltage of the circuit in the standby mode when the circuit is not performing any useful computation.

The above-mentioned dynamic leakage reduction methods cannot be applied in the active mode since the circuit is required to do computation at a target speed. However, we have observed that considerable portions of circuits are idle for periods of time even in the active mode of operation. Therefore, there exists opportunities for dynamic application of leakage reduction techniques in the active mode as well.

In this paper, we present a low-overhead design methodology for efficiently reducing active leakage power using supply gating. Besides, the proposed method reduces switching power by preventing redundant switching in idle parts of a circuit. We also propose a synthesis methodology based on Shannon expansion to provide opportunities for supply gating in the active mode for general combinational circuits. The proposed method results in automatic savings in standby leakage because of stacking [6].
Our contributions in this paper are as follows:

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

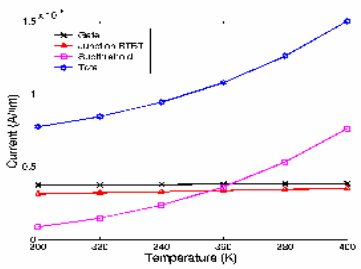
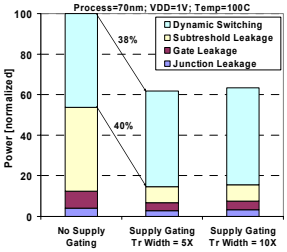
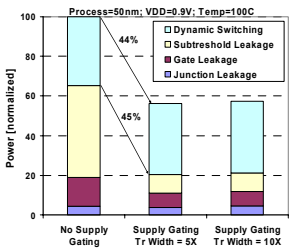


Fig. 1: Different leakage components vs. temperature for 50nm NMOS [1].



(a) 70nm process



(b) 50nm process

Fig. 3: Effectiveness of supply gating for power reduction

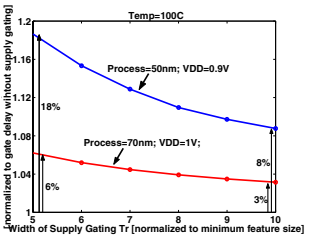


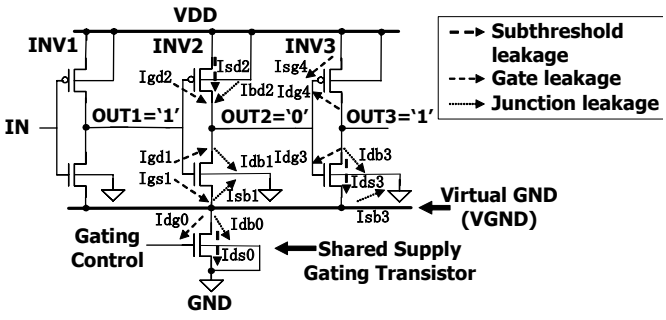
Fig. 4: Effect of supply gating on delay

- Novel circuit techniques to reduce active power (both switching and leakage) using supply gating. The technique has been applied to a decoder circuitry to show large improvements in active power with minimal area and delay overhead.
- Extension of supply gating for power reduction in active mode to general logic circuits using Shannon expansion based synthesis method.
- Sizing of supply gating transistors for minimal impact on performance while maximizing power reduction. A pre-computation based method for hiding the delay of control signal generation for supply gating transistors is proposed.

2. Supply Gating for Reducing Active Power

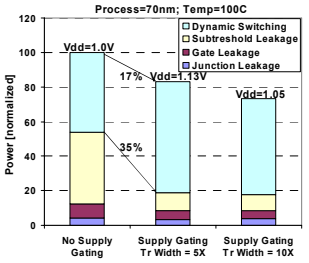
Assuming that part of a circuit is identified to be idle in the active mode, redundant switching in that part of the circuit results in wasted switching power in addition to leakage power. By applying supply gating to that portion of the circuit, both components of the wasted power can be reduced. Supply gating can prevent propagation of signal activities from primary inputs to the intermediate and output nodes of the idle circuit.

Fig. 2 illustrates supply gating applied to an inverter chain. In this circuit, supply gating is implemented using an NMOS transistor that controls the connection of the virtual ground (VGND) node to the real ground (GND). In the supply gated mode, due to circuit leakage, the voltage of the virtual ground node reaches an intermediate voltage level, resulting in stacking effect for leakage reduction [5]. In addition to significant reduction in leakage current, supply gating prevents redundant switching in the idle blocks. To understand the impact of supply gating on overall and individual components of leakage, let us consider two inverters that are in two different states as shown in Fig. 2 (INV2 and INV3) and observe different components of leakage currents in the supply gated mode. The different components of leakage and the direction of current flow in each logic gate depend on the state of the logic. The detailed leakage components are illustrated in Fig. 3 for this state of the circuit in two processes (70nm and 50nm).

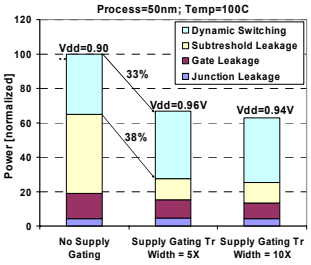


In supply gated mode (Gating Control ='0'): if OUT1='1' → V(OUT1)=V(OUT3)=V_{dd} and V(OUT2)=V(VGND)≈V_{dd}-V_t

Fig. 2: Supply gating for prevention of input switching propagation and leakage reduction



(a) 70nm process



(b) 50nm process

Fig. 5: Power reduction by supply gating at iso-delay

The leakage breakdown of the circuit (INV2 and INV3) with and without supply gating is shown in Fig. 3. Dynamic switching power is also added to obtain the total power. Dynamic switching power is measured in the active mode for a frequency of 1GHz and input switching activity of 20%. In the supply gated case, two sizes of the supply gating transistors are considered: 5 times the minimum size (5X) and 10 times the minimum size (10X). In the circuit without supply gating, the subthreshold leakage is the dominant component of leakage (more than 50% and 60% of total in 70nm and 50nm). By supply gating, the subthreshold leakage reduces dramatically due to the stacking effect (negative Vgs and body effect on the OFF NMOS transistors). The overall gate leakage reduces because of smaller voltage drop across gate oxides of transistors due to the raised virtual ground voltage (reduction in the effective voltage drop across the supply lines of the circuit: VDD and VGND). The reverse biased junction tunneling leakage is not affected much by supply gating because voltage drop across some junctions reduce (Ibd2) whereas voltage drop across some other junctions increase (Idb1 and Isb1). Since the overall leakage is dominated by subthreshold (and gate leakage in such a scaled technology), supply gating remains an effective method for total leakage reduction. Another observation from Fig. 3 is that the overall leakage in the supply gated mode is weakly dependent on the size of the supply gating transistor. There is a slight increase in leakage by upsizing the supply gating transistor due to small increase in each component of leakage. The switching power in the active mode is insignificantly affected by the supply gating. However, due to reduction in the leakage, there is an overall power reduction of 38% and 44% in total power in 70nm and 50nm nodes, respectively. The result clearly shows the effectiveness of supply gating in scaled technologies.

From Fig. 4, it is observed that the delay reduces by upsizing the supply gating transistor. In 70nm, supply gating has a delay overhead of 6% to 3% as the size of the supply gating transistor varies from 5X to 10X. The delay overhead can be reduced by increasing the supply voltage. However, high voltage reduces the power savings of the supply gating technique. Fig. 5 provides an iso-delay comparison of power dissipation between the original design (no supply gating) and the design with dynamic supply gating. By increasing the supply voltage of the supply gated circuit, it is possible to avoid the delay penalty. In that case, the power saving

reduces mainly due to increase in the dynamic switching power. However, the overall power still remains less than the original design. Under iso-delay voltage scaling, the supply gated circuit shows power reduction of 17% and 33% in 70nm and 50nm process, respectively. Another interesting observation from Fig. 5 is that by upsizing the supply gating transistor, the required supply voltage for maintaining the delay is reduced and hence, more power reduction is achieved under iso-delay.

Since the delay improvement becomes marginal beyond the size of 10X for the supply gating transistor, we have chosen this size for the supply gating transistor in our designs. In a real circuit, all the logic gates do not switch simultaneously. Therefore, by sharing the supply gating transistor, the sizing of the shared transistor can be reduced. We have used the following rule for sizing the shared supply gating transistor. Assuming half of the logic gates in a circuit switch at a time (statistically speaking), the size (width) of a shared supply gating transistor is given by:

$$W = (10 \times L_{min}) \times (n/2) \qquad (2)$$

where n is the total number of logic gates in the circuit and L_{min} is the minimum feature size in a given process technology. If further delay reduction is required, the size of the supply gating transistor can be increased without much impact on leakage reduction (Fig. 3).

3. A Circuit Example: Active Leakage Reduction in Memory Address Decoder

In this section, we show that supply gating for active leakage reduction can be easily applied to any circuit with a tree structure. A memory address decoder is used as an example to explain the power reduction capability of the supply gating technique in the active mode. In address decoders, the switching activity of logic gates is low, especially for the final buffers, which drive the global word line (WL). Furthermore, to drive the global WL, which has a large capacitance, large buffers are used. In scaled technologies, such large buffers can dissipate significant leakage power.

A row address decoder consists of pre-decoders, final-decoders, and WL drivers [6]. The decoder structure shows that considerable portions of the circuit are inactive during regular operations. By using the output of the pre-decoder, it is possible to turn off (by supply gating) certain parts of the final decoder, thereby, achieving active leakage saving in the logic gates of the idle blocks. Fig. 6 shows an 8-bit row decoder with supply gating based active leakage management. As shown in Fig. 6, the most significant bits of the row address are fed into the NAND gates (G1) within the pre-decoder. The output of G1 is sent to the final decoder together with the outputs of the other pre-decoder gates as in the conventional design. Moreover, the output of G1 turns on or off the supply gating transistors (M0 and M1 in Fig. 6) so as to activate or deactivate certain blocks of the final decoders. In Fig. 6, the WL drivers are selectively gated to GND or VDD. This is due to the fact that a floating WL will reduce the memory cell stability. Hence, the voltage of a WL has to be stable at zero if it is not accessed. Moreover, the supply gating transistors, M0 and M1, can be shared among all the final decoding logic controlled by the same G1 output. This is due to the fact that, even in the active mode, only one path in these blocks is triggered.

Fig. 7 shows the percentage of improvement in total power dissipation achieved by dynamic supply gating in decoders designed in 70nm and 50nm nodes. With the increase in the size of the decoder (number of bits), the power savings increase considerably. This is due to the fact that the number of final decoders increases exponentially with the increase in the number of bits. For such a situation, the total power of row address decoders is dominated by the leakage of the logic gates in the final decoders. Fig. 7 also shows that there is more power reduction in 50nm than 70nm. Hence, the effectiveness of the dynamic supply gating for active leakage power reduction improves

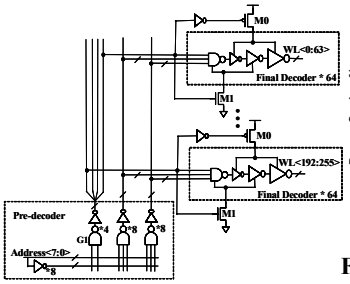


Fig. 6: 8-bit row decoder

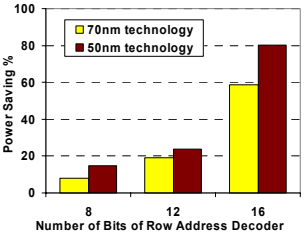


Fig. 7: Power saving by dynamic supply gating in decoders

with technology scaling.

The overhead of supply gating in row address decoders is minimal. Since the output of the pre-decoder is used to control the gating transistors, the gating transistors are turned on by the time the inputs propagate to the final decoder. Therefore, the delay of turning on the supply gating transistors is hidden by the pre-decoder delay. We observe that the delay overhead is about 9% of the total decoder delay for both 70nm and 50nm technologies. Since the gating transistors are shared, the area overhead is very low (only 1.3% of the decoder area).

4. Active Leakage Reduction in General Logic Circuits: A Synthesis Technique Based on Shannon Expansion

We extend the principle of supply gating described in Section 3 to develop a synthesis flow for application of dynamic supply gating to general combinational circuits. The synthesis technique should distinguish between the active logic gates and the idle ones during the active mode of operation and dynamically apply supply gating to the idle gates without causing any final output node to get a floated state. In this section, we develop such a synthesis approach using Shannon expansion [8].

4.1. Dynamic Supply Gating (DSG) Scheme using Shannon Expansion

Shannon expansion has been used in logic synthesis for logic simplification and optimization [8]. It partitions any Boolean expression into disjoint sub-expressions as shown below:

$$\begin{aligned} f(x_1,...,x_i,..., x_n) &= x_i \cdot f(x_1,...,x_i=1,..., x_n) + x_i' \cdot f(x_1,...,x_i=0,..., x_n) \\ &= x_i \cdot CF_1 + x_i' \cdot CF_2 \\ CF_1 = f(x_1,...,x_i=1,..., x_n); &\quad CF_2 = f(x_1,...,x_i=0,..., x_n) \end{aligned} \qquad (3)$$

where, x_i is called the control variable, and CF_1 and CF_2 are called cofactors. From the above expression, it is clear that depending on the state of the control variable (x_i), the computed output of only one of the cofactors (CF_1 or CF_2) is required at any given instant. This implies that the other cofactor does redundant computation and leaks at any time instant. Hence, this provides an opportunity for gating the supply of the idle cofactor circuit to eliminate its redundant computation and leakage energy. We utilize Shannon theorem to identify the active/idle parts of a circuit for dynamic supply gating (DSG). The proposed DSG scheme using Shannon expansion is illustrated in Fig. 8(a). The supply gating transistors of CF_1 and CF_2 are controlled by x_i and x_i', respectively. The output of CF_1 and CF_2 are merged using a multiplexer (MUX) controlled by x_i. The MUX directs the output of the active cofactor to the final output.

4.2. Areas of Optimization

There are areas of optimization to further reduce power dissipation in the proposed DSG scheme. The Boolean function itself has to be initially optimized to minimize the number of literals before applying the Shannon expansion. This optimization ensures that the derived cofactors from the Shannon expansion are also optimized for minimal area and therefore power. Let us consider the following Boolean function f:

$$f = x_1'x_2 + x_1x_2' + x_1x_4x_5x_6 + x_1'x_3x_5x_6 + x_1x_7x_8 + x_7x_8x_9x_{10}x_{11} + x_1'x_{10}x_{11} + x_1x_5x_6 + x_4x_7x_8$$

After initial optimization, the following optimized function is obtained (f_{opt}):

$$f_{opt} = x_1'x_2 + x_1x_2' + x_1x_5x_6 + x_1'x_3x_5x_6 + x_1x_7x_8 + x_7x_8x_9x_{10}x_{11} + x_1'x_{10}x_{11} + x_4x_7x_8$$

An optimized Boolean function may contain minterms that do not include the control variable. These minterms will be included in each of the cofactors determined by the Shannon expansion. This would involve duplication of the same logic realization of these minterms, which is not desirable in terms of area and leakage. Therefore, to minimize area overhead, it is better to include them as a separate shared logic (SL) circuit common to both the cofactors.

However, the shared logic cannot be supply gated because its computation is required irrespective of the state of the control variable. Therefore, the optimal strategy is to choose a control variable that would minimize the shared logic. In the above example, the optimal control variable is x_1, as it appears in the largest number of minterms (minimizes the shared logic). The cofactors determined by the Shannon expansion are as follows:

$$\text{Control Variable} = x_1 \Rightarrow$$

$$CF1 = x_2' + x_5x_6 + x_7x_8 + x_7x_8x_9x_{10}x_{11} + x_4x_7x_8$$

$$CF2 = x_2 + x_3x_5x_6 + x_{10}x_{11} + x_7x_8x_9x_{10}x_{11} + x_4x_7x_8$$

The last two minterms of CF1 and CF2 are common because they are the minterms of f_{opt} that do not contain x_1. Therefore, those two minterms are implemented as a shared logic (SL) as follows:

$$f_{opt} = x_1 \bullet CF1opt + x_1' \bullet CF2opt + SL$$

$$CF1opt = x_2' + x_5x_6 + x_7x_8$$

$$CF2opt = x_2 + x_3x_5x_6 + x_{10}x_{11}$$

$$SL = x_4x_7x_8 + x_7x_8x_9x_{10}x_{11}$$

The circuit realization of the above expression with DSG is shown in Fig. 8(b). The final output is derived by OR-ing the MUX output and the output of the shared logic.

The cofactors CF1opt, CF2opt and the shared logic SL may have common sub-expressions in their minterms. These common sub-expressions represent the same logic gates with same inputs, which are duplicated in separate blocks after the logic is mapped to a library. To further reduce the area, the common sub-expressions among CF1opt, CF2opt, and SL should be identified and shared. The shared sub-expressions common to CF1opt/CF2opt, CF1opt/SL and CF2opt/SL are moved to the Pre-MUX shared logic as shown in Fig. 9. A new variable (y_i) is assigned to any shared sub-expression. In the above example, the common sub-expressions are as follows:

$$y_1 = x_5x_6; \quad y_2 = x_7x_8; \quad y_3 = x_{10}x_{11}$$

The remaining logic in SL after the sub-expression sharing is represented as Post-MUX shared logic as shown in Fig. 9. The expressions CF1opt, CF2opt and Post-MUX are modified in terms of the new variables (y_i's) as shown below for the above example:

$$CF1opt = x_2' + y_1 + y_2; \quad CF2opt = x_2 + x_3y_1 + y_3; \quad SL = x_4y_2 + x_9y_2y_3$$

The logic of the shared minterms (y_i's) is implemented in Pre-MUX shared logic and provides outputs to CF1opt, CF2opt and Post-MUX blocks as shown in Fig. 9. These blocks will be individually synthesized using the above expressions (y_i's are treated as primary inputs).

The above-mentioned design methodology targets overall power reduction. It can be recursively applied for factoring of CF1opt, CF2opt and SL to further reduce power. However, there is some delay/area and switching energy overhead associated with added supply gating transistors and the multiplexer at each level of recursion. Beyond certain number of recursion levels the added overhead may offset the savings obtained by the above design methodology. Therefore, there is an optimal number of levels (hierarchy) for recursive application of our design methodology to minimize power dissipation, while satisfying a given delay constraint.

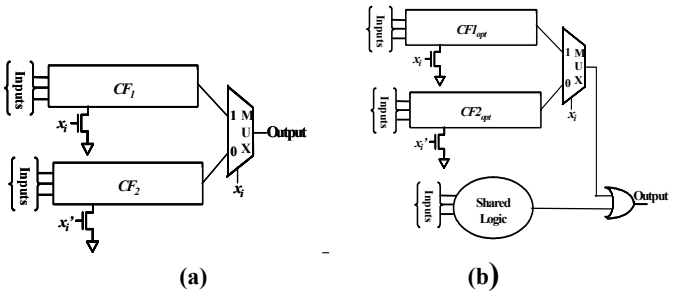


Fig. 8: Proposed dynamic supply gating based on Shannon expansion: (a) basic idea, (b) with sharing among minterms

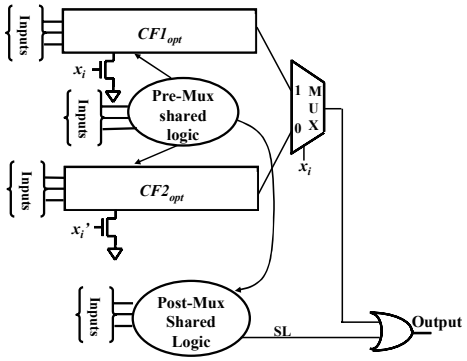


Fig. 9: Common sub-expression as shared logic (without supply gating)

4.3. Automated Synthesis Flow for Dynamic Supply Gating

In this section, we propose an automated synthesis flow for dynamic supply gating (DSG) using Shannon expansion. The automated synthesis flow considers all the optimization steps described in the previous section. The complete synthesis flow is shown in the Fig. 10. Part (a) of Fig. 10 represents the optimal synthesis flow for one level of DSG using Shannon expansion. Part (b) of Fig. 10 highlights the algorithm for recursive application of the method described in part (a) for multi-level expansion.

In part (a) of the flow, conventional logic optimization and synthesis (step 1) is performed on the input Boolean expression and the resulting logic is technology-mapped to a gate library. Then, the resulting power and delay (P_{orig} and D_{orig}) are estimated using a graph representation of the optimized logic. The power estimated in this part of the flow will be used to compare the power resulting from DSG synthesis flow to determine if any power saving is obtained by dynamic supply gating. The estimated delay is used to verify whether it satisfies the specified delay constraint.

Part (a) of the flow illustrates the steps of synthesis for DSG. The optimized logic function obtained from step 1 is converted to a two-level format (sum-of-products) in step 2. In step 3, the optimal control variable is identified and the corresponding cofactors (CF1 and CF2) and the shared logic (SL) are generated. The heuristic proposed to select the optimal control variable is discussed in detail in Section 4.4.

The cofactors and the shared logic (CF1, CF2 and SL) are area optimized by utilizing the Common Sub-expression Elimination (CSE) described in Section 4.2. Then, the expressions of Pre-Mux shared logic, Post-Mux shared logic, CF1opt, and CF2opt are generated. After this optimization step, each of the logic functions (eg. CF1, CF2, SL) are separately synthesized and mapped to technology library. The individually synthesized functions are

connected together with MUX and OR (Fig. 8). The corresponding delay (D_{level1}) and power (P_{leve11}) are estimated from a graph representation of the combined logic.

The estimated power (P_{level1}) is compared to that of the original design (P_{orig}) to evaluate the power saving. If no power saving is achieved by DSG, supply gating is not used for the current level of expansion. If there is power reduction, the delay (D_{level1}) is compared with the given delay constraint to check if the DSG synthesized circuit meets the delay requirement. If the delay constraint is not met, delay reduction methods such as upsizing supply gating transistors and reducing logic sharing are applied and the power/delay conditions are rechecked. If both the power and delay conditions are satisfied, the circuit of current level of DSG is selected as the optimized output.

The recursive application of the DSG synthesis at multiple hierarchies is highlighted in part (b) of the flow (Fig. 10). The decision to partition the j^{th} cofactor at the hierarchy level 'i-1' (denoted by $CF_{i-1,j}$) is based on: 1) comparison of the total power of its cofactors/shared logic circuits ($CF1_{i,k}$, $CF2_{i,k}$ and $SL_{i,k}$) with its original power consumption, 2) comparison of circuit delay with the delay constraint (D_{spec}) after expansion of $CF_{i-1,j}$ and application of supply gating to each of its cofactors. If the power of the circuit consisting of the cofactors and the shared logic ($CF1_{i,k}$, $CF2_{i,k}$ and the $SL_{i,k}$) is less than $CF_{i-1,j}$ and the delay constraint is satisfied ($D(CF1_{i,k}$, $CF2_{i,k}$ and $SL_{i,k}) < D_{spec}$), DSG expansion is performed at that hierarchy level. Otherwise, the recursion stops at the level 'i-1' for that cofactor circuit ($CF_{i-1,j}$).

4.4. Optimal Selection of Control Variable

In a circuit, the total power consists of both switching and leakage power. To estimate the total circuit power by its Boolean expression, the following assumptions are made:

- All logic gates have the same average switching power denoted by P_{sw} and the same average leakage power denoted by P_{leak}.
- The number of logic gates after synthesis is proportional to the number of literals in the Boolean expression.
- In a 2-level Boolean logic function, a particular input variable x_i is associated with 'a' number of literals (whenever x_i appears in one minterm, the other literals in the same minterm are counted) and its complement, x_i', is associated with 'b' number of literals. The total number of literals is 'n'.
- The signal probability of x_i=1 is P_{xi}. The switching probability of x_i is S_{xi}.
- The switching power of the gated transistor is $P_{GatingTr}$.

With the above assumptions, the power consumption of the circuit after applying Shannon expansion is estimated as follows:

$$P_{total} \approx \underbrace{[n-(a+b)](P_{sw}+P_{leak})}_{\text{Shared Logic Power}} + \underbrace{P_{xi}[a(P_{sw}+P_{leak})]}_{\text{CF1 Power (co-factor of } x_i)} + \underbrace{(1-P_{xi})[b(P_{sw}+P_{leak})]}_{\text{CF2 Power (co-factor of } xi')} + \underbrace{S_{xi}\cdot P_{Gating\_Tr}}_{\text{Gating Tr. Power}}$$

$$\approx [n-(a\cdot(1-P_{xi})+b\cdot P_{xi})](P_{sw}+P_{leak})+S_{xi}\cdot P_{Gating\_Tr}$$

As shown by the above formulation, with the knowledge of P_{xi}, S_{xi} (from input signal statistics), a, b (from the Boolean function) and P_{sw}, P_{leak}, $P_{GatingTr}$ (from the library), a greedy algorithm can be implemented to search for the optimal input variable, which leads to minimum overall power after factorization and application of supply gating at a particular level. This variable is selected as the control variable to apply Shannon expansion to the Boolean equation.

4.5. Synthesis for Multiple Output Circuits

The DSG synthesis method can be easily extended to multi-output circuits by choosing a common control variable for all outputs at each level of expansion. For a multiple output circuit, all the minterms from every output expression are initially combined together to determine the control variable. There might be identical minterms in the combined function (from the different output expressions) during the selection of the control variable. These identical minterms are counted only once since in the circuit representation, the circuit for this minterm is shared among all the outputs. After selection of the control variable, DSG synthesis is applied to determine the cofactors (CF1s and CF2s) and shared logic (SL) for all the output functions.

The multi-output circuit synthesis is illustrated with an example. Consider a 3-output circuit described by the function:

$$Out_1 = x_1x_2x_3 + x_1'x_6 + x_2x_4$$

$$Out_2 = x_1x_2x_3 + x_1'x_4x_5 + x_5x_6 + x_3x_4$$

$$Out_3 = x_1x_2 + x_1'x_4x_3 + x_5x_6$$

In the combined minterm representation, $x_1x_2x_3$ is present in expressions for both Out_1 and Out_2.

Therefore, it is counted only once in determining the control variable. Since the variable x_1/x_1' is present in the largest number of minterms among all variables in the multi-output logic, x_1 is selected as the control variable. Applying DSG based synthesis to all the three logic expressions in terms of x_1:

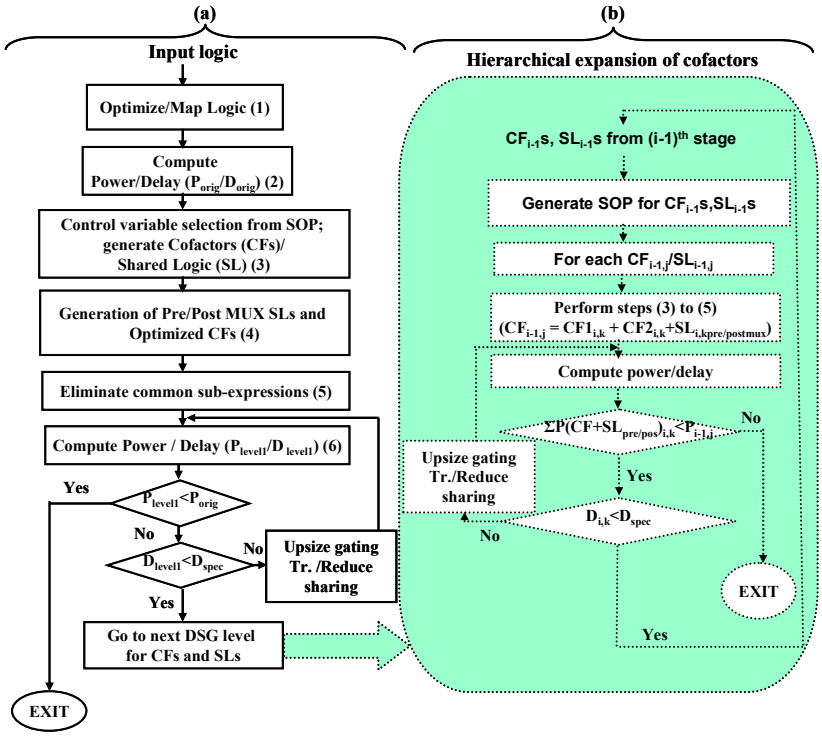


Fig. 10: Optimal synthesis flow for dynamic supply gating

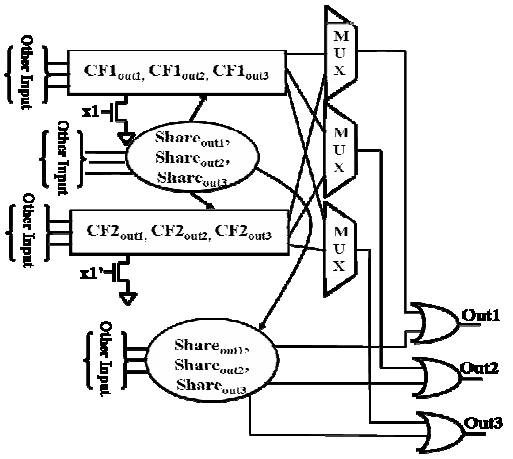


Fig. 11: Synthesis for multi-output circuit

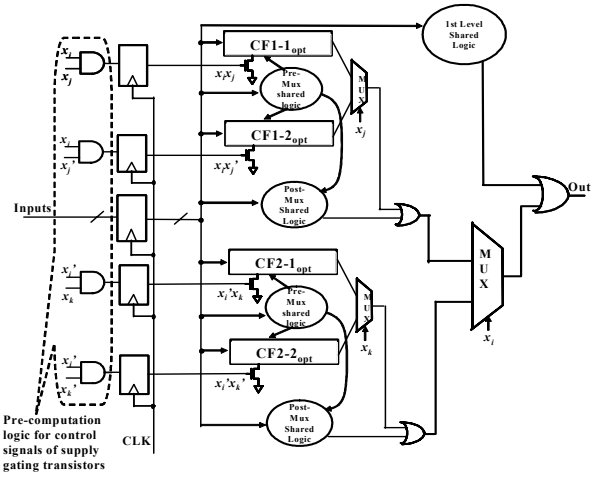


Fig. 12: Pre-computation of supply gating control signals

$$CF1_{out1} = x_2x_3; \quad CF2_{out1} = x_6; \quad SL_{out1} = x_2x_4$$

$$CF1_{out2} = x_2x_3; \quad CF2_{out2} = x_4x_5; \quad SL_{out2} = x_5x_6 + x_3x_4$$

$$CF1_{out3} = x_2; \quad CF2_{out3} = x_3x_4; \quad SL_{out3} = x_5x_6$$

$CF1_{out1}$, $CF1_{out2}$, $CF1_{out3}$, and $CF2_{out1}$, $CF2_{out2}$, $CF2_{out3}$ are synthesized conventionally as three output circuits, respectively, as shown in Fig. 11. The individual blocks undergo a similar synthesis flow for next level of expansion as that of the single output case (refer Fig. 10).

4.6. Pre-Computation of Supply Gating Control

The control signals of supply gating transistors are generated by decoding the selected control variables by the DSG synthesis flow. This decoding delay can become a critical part of the circuit delay if not properly hidden. That is because, the computation in a cofactor cannot start until the control signal of the supply gating transistor of that cofactor is decoded from the primary inputs and the gating transistor of that cofactor is turned on. Therefore, if the decoding delay is not hidden, it adds a considerable overhead to the circuit delay. In order to hide this decoding delay, a pre-decoding technique is used to compute the decoded control signals ahead of time so that the signals are ready at the same time as the primary inputs of cofactors. A pre-computation scheme for a 2-level DSG circuit is shown in Fig. 12. The supply gating control signals are computed in the previous cycle and applied to the supply gating transistors at the same time as the primary inputs. In addition to existing latches that capture the primary inputs, extra latches are required to sample the pre-computed control signal. However, this does not add any significant hardware overhead since the number of required supply gating control signals is small compared to the number of primary inputs of the circuit.

5. Experimental Results

To verify the effectiveness of the proposed dynamic supply gating synthesis approach, experiments are performed on a set of MCNC benchmark circuits. We have used SIS [11] as a general logic optimizer in our synthesis flow. Inputs are assumed to be random (switching activity and signal probability of 0.5). The benchmarks are synthesized using the DSG synthesis flow (Fig. 10). For a basis of comparison, the benchmarks are also optimized for area using SIS (without supply gating). For accurate power estimation, after technology mapping to a standard cell library, the resulting Spice netlists are simulated using Nanosim. The circuit delay and area are calculated using Synopsys design compiler. The resulting netlists from both approaches are compared in terms of power, delay, and area as shown in Table 1. The results show reduction of 15% to 88% in total power, demonstrating the effectiveness of the DSG synthesis approach for low power design. The reductions in power are attributed to reductions in both switching and leakage components of power dissipation. Despite the insertion of supply gating, the delay improves in most of the cases due to less effective loading on internal nodes as a result of circuit partitioning into cofactors. The area simulation for the 10 benchmarks, listed in Table 1, shows an average area overhead of 20%, which is due to the circuit partitioning.

Table 1. Experimental results (70nm Process [10], Vdd=1V, Temp=100C)
(% numbers are percentages of reduction)

MCNC CKT	Power(μW)		Delay (ns)		Area (μm^2)	
	Conv.	DSG	Conv.	DSG	Conv.	DSG
count	125.3	71.2 **(+43%)**	0.60	0.45 (+25%)	1698	1504 (+11%)
cm150a	26.1	17.1 **(+34%)**	0.35	0.3 (+14%)	241	251 (-4%)
decod	18.1	14.7 **(+19%)**	0.21	0.162 (+22%)	191	331 (-73%)
alu2	204.1	174.3 **(+15%)**	1.14	1.14 (0%)	1526	2873 (-88%)
mux	26.6	7.4 **(+72%)**	0.39	0.27 (+31%)	284	298 (-5%)
Cht	87.9	51.4 **(+41%)**	0.33	0.28 (12%)	645	677 (-5%)
pcler8	49.0	17.2 **(+65%)**	0.42	0.57 (-34%)	645	763 (-18%)
pcle	74.2	8.7 **(+88%)**	0.42	0.51 (-20%)	570	570 (0%)
sct	71.2	16.2 **(+77%)**	0.41	0.40 (+2%)	578	677 (-7%)
x2	14.3	11.3 **(+21%)**	0.39	0.33 (+15%)	284	290 (-2%)

6. Conclusion

We have presented a low-overhead design methodology that targets reducing both active leakage and switching power using dynamic supply gating. A logic synthesis approach based on Shannon expansion is proposed that dynamically applies supply gating to idle parts of general logic circuits during active mode of operation. The proposed technique results in automatic leakage power reduction in the standby mode as well. Experimental results on a set of MCNC benchmarks show promising results in terms of power saving in scaled technologies.

7. References

[1] S. Mukhopadhyay et al., "Modeling and estimation of failure probability due to parameter variations in nano-scale SRAMs for yield enhancement," Symp. on VLSI Circuits, pp. 12-14, 2003.

[2] Y. Taur and T. H. Ning, Fundamentals of Modern VLSI Devices, New York: Cambridge Univ. Press, 1998

[3] G. Sery et al., "Life is CMOS: why chase the life after?" Design Automation Conf., 2002, pp. 78–83.

[4] R. Krishnarnurthy et al., "High-performance and low-power challenges for sub-70 nm microprocessor circuits," CICC, pp. 12-15, May 2002.

[5] J.W. Tschanz et al. "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," IEEE JSSC, vol. 38, pp. 1838–1845, 2003.

[6] A. Chandrakasan, Design of High-Performance Microprocessor Circuits, IEEE Press.

[7] A. Keshavarzi et al., "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs," ISLPED, pp. 207-212, 2001.

[8] L. Lavagno et al., Design Automation Conf. pages 254–260, 1995.

[9] M. Liu et al., "Leakage power reduction by dual-Vth designs under probabilistic analysis of Vth variation," I SLPED, pp. 2-7, Aug. 2004.

[10] Predictive Technology Model,www.device.eecs.berkeley.edu

[11] SIS, University of California at Berkeley.