Low Power Synthesis of Dynamic Logic Circuits Using Fine-Grained Clock Gating

Nilanjan Banerjee and Kaushik Roy
Purdue University
{nbanerje, kaushik}@purdue.edu

Hamid Mahmoodi
San Francisco State University
mahmoodi@sfsu.edu

Swarup Bhunia
Case Western Reserve University
skb21@cwru.edu

***Abstract* — Clock power consumes a significant fraction of total power dissipation in high speed precharge/evaluate logic styles. In this paper, we present a novel low-cost design methodology for reducing clock power in the active mode for dynamic circuits with fine-grained clock gating. The proposed technique also improves switching power by preventing redundant computations. A logic synthesis approach for domino/skewed logic styles based on Shannon expansion is proposed, that dynamically identifies idle parts of logic and applies clock gating to them to reduce power in the active mode of operation. Results on a set of MCNC benchmark circuits in predictive 70nm process exhibit improvements of 15% to 64% in total power with minimal overhead in terms of delay and area compared to conventionally synthesized domino/skewed logic.**

1. INTRODUCTION

High performance designs often exploit dynamic logic styles such as domino for higher speed of operation and lower area compared to their static CMOS counterparts [1]. The clock signal is essential for dynamic logic circuits since they operate in precharge and evaluation phases. Experiments on logic blocks designed with domino gates show that around 40% of the power consumption comes from clock power. Hence, a low power design methodology for domino circuits should reduce the clock power in addition to switching and leakage power.

It is difficult to use domino circuits in scaled technologies due to the dependence of their noise margin on threshold voltage variation. Skewed CMOS [2] is a specific dynamic logic style that significantly improves the noise tolerance over domino circuits. Similar to domino logic, clock power is a significant component of total power in skewed circuits. Therefore, a low-power synthesis approach for skewed logic should try to minimize the clock power dissipation as well.

Clock gating is a popular technique to reduce clock power. AND-ing the clock with a gate-control signal disables the clock input of a circuit whenever the circuit is not performing any useful computation [4]. It avoids power dissipation due to unnecessary charging and discharging of the unused circuits. This technique has been used at an architectural level to gate clock inputs of complete blocks for microprocessor power reduction [4]. However, block-level clock gating fails to exploit the fact that circuits within the block itself might be idle for long periods of time. Automatic clock-gating insertion at RTL-level to eliminate redundant computations performed by temporally unobservable blocks by exploiting *observability don't care* (ODC) conditions has also been proposed [5]. However, ODC-based clock gating involves gating of control signals for the sequential boundaries only and does not involve gating within the combinational block.

The above methods do not take into account the possibility of reducing clock power in combinational logic implemented with dynamic logic. Since considerable portions of the circuits within each block may remain idle even when the circuit is performing useful computation, there exist opportunities for power savings. In this paper, we present a low-overhead synthesis technique for dynamic logic using fine-grained clock gating. The main contributions of this paper are as follows:

- Novel design techniques for application of fine-grained clock gating in dynamic logic circuits at circuit level granularity. This technique provides a threefold advantage when applied to dynamic circuits: a) it reduces power in the clock line; b) it prevents redundant switching in the idle logic gates; c) it improves noise immunity by reducing power supply noise, a critical issue in domino circuits.
- Combining clock gating and Shannon decomposition to develop a low power synthesis methodology for dynamic logic circuits with minimal overhead on performance and die-area.

The paper focuses on two specific styles of dynamic logic, namely: domino and skewed CMOS. However, the proposed clock gating technique is generally applicable to all styles of dynamic circuit using clock control.

2. DOMINO AND SKEWED LOGIC

2.1. Domino Logic

Fig. 1 shows a typical domino logic circuit [1]. It consists of an n-type domino logic block followed by a static inverter. The circuit operates in two phases: i) Precharge, and ii) Evaluation. During precharge phase (CLK = '0'), the output of the pull-down network (PDN) is charged to V_{dd}, and output of the inverter is set to '0'. During evaluation (CLK ='1'), the outputs of n-logic blocks conditionally discharge (if there is conducting path to GND) and the outputs of inverters undergo a conditional transition of $0 \rightarrow 1$. In absence of a conducting path, output of the PDN-logic stays charged at high.

Due to reduced number of transistors per gate and a single transistor load per fan-in, the load capacitance for domino gates is substantially lower than standard CMOS, resulting in faster switching speeds. Domino circuits can be made

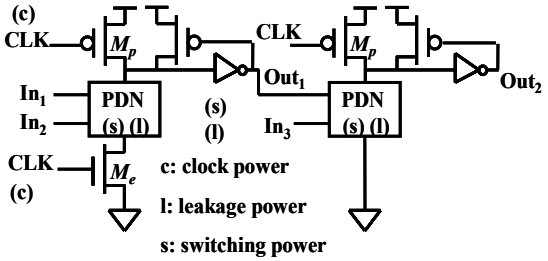

Fig. 1: Domino logic with various sources of power dissipation- c: Clock, s: Switching, and l: Leakage Power

more robust by adding a level restoring (keeper) transistor to reduce the parasitic effects of charge sharing and charge loss. To achieve higher speeds of operation in domino circuits, it is customary to have a clocked input footer transistor only for the first level gates [1]. Fig. 1 also shows the main sources of power dissipation for a circuit implemented in domino logic.

2.2 Skewed CMOS

However, two inherent drawbacks of domino logic limit its usefulness for scaled technologies. First, the noise margin of domino logic circuits is relatively small compared to static CMOS since it depends on the threshold voltages of transistors. This makes domino logic circuits extremely susceptible to failures due to threshold voltage variation, noise injection, and high sub-threshold leakage. Second, domino logic dissipates much more power than static circuits due to higher activity; therefore, it is not suitable for low power operation.

To overcome drawbacks of domino logic, an alternative noise-immune high performance logic style, called skewed logic [2] has been proposed. Skewed logic circuits are CMOS circuits, with the size of pull-down network (PDN) decreased and that of pull-up network (PUN) increased, or vice versa, for fast low-to-high or high-to-low transitions, respectively. Sizing the PDN and PUN to favor one transition direction is referred to as *skewing* [2]. Similar to domino logic, skewed logic is operated in precharge-evaluation fashion for high performance with fast transition for evaluation, and slow transition for precharge. Precharging can be accomplished either by clocked skewed logic gates, which precharge just like domino gates, or by the propagation of precharged logic values through the logic chain originating from a clocked gate [2]. For fast evaluation, skewed-down gates are followed by skewed-up gates, and vice versa. Skewed logic is comparable to domino logic in terms of speed. At the same time, skewed logic has better noise immunity than domino logic due to its complementary nature. The sources of power consumption for skewed circuits are similar to that of domino circuits.

3. SYNTHESIS OF CLOCK-GATED DOMINO LOGIC

Section 2 emphasizes that clock is critical for both logic styles (domino/skewed) and that clock power is a significant fraction of the total power dissipation. Therefore, synthesis strategies targeting clock power reduction is extremely useful for such designs. In this section, we develop a synthesis methodology for fine-grained clock gating of domino circuits in the active mode by Shannon based Boolean partitioning of a logic function and apply it to a benchmark to evaluate the power savings.

A. Shannon Expansion

Shannon expansion partitions any Boolean expression into disjoint sub-expressions as shown below:

$$f(x_1,...,x_i,...,x_n) = x_i \bullet f(x_1,...,x_i=1,...,x_n) + x_i' \bullet f(x_1,...,x_i=0,...,x_n)$$
$$= x_i \bullet CF_1 + x_i' \bullet CF_2 \quad (1)$$
$$CF_1 = f(x_1,...,x_i=1,...,x_n); \quad CF_2 = f(x_1,...,x_i=0,...,x_n)$$

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. The output of CF_1 and CF_2 are combined using a multiplexer (MUX), which is controlled by x_i. If the boolean expression f contains subexpressions independent of control variable x_i, then a *Shared Cofactor* (*sCF*) might be present. Shared cofactor performs useful computation irrespective of the state of the control variable. To further reduce the area, the common sub-expressions among CF_1, CF_2, and *sCF* should be identified and shared. The shared sub-expressions common to CF_1/CF_2, CF_1/sCF and CF_2/sCF are moved to the Pre-MUX shared logic (*Pre-Mux sCF* shown in Fig. 3(a)). The output of the MUX (which directs the output of the active cofactor) must be OR-ed (for a sum-of-products representation) with the output of the *sCF* to obtain the final output. The overall circuit after Shannon expansion is shown in Fig. 3(a).

B. Dynamic Clock Gating (DCG) scheme for domino circuits using Shannon-based partitioning

Equation 1 implies that at any given time instant only one cofactor performs useful computation while the other cofactors perform redundant computations. The proposed DCG scheme for domino logic circuits using Shannon's expansion is illustrated in Fig. 3(b) for one level of expansion. The AND-gates used for clock gating of CF_1 and CF_2 are controlled by x_i and x_i', respectively, where x_i is the control variable. Therefore, when x_i is active and the clock signal is high the clock signal input of CF_1 is '1', whereas the clock input of other cofactor is gated to '0'. Gating the clocks of the cofactors in this fashion eliminates redundant computation in the idle cofactor as well as saves its clock power. It should be noted that all these operations are performed in the active mode of circuit operation. The procedure can be performed hierarchically for multiple levels of expansion (CF_1 can be further expanded to CF_{11} and CF_{12} *and so on*) for additional power savings while satisfying the area and delay constraints. The shared logic is always turned on and is therefore not gated (Fig. 3(b)).

C. Selection of control variable for circuit partitioning

The choice of the control variable is guided by the objective of minimizing total power in active mode. Therefore, a control variable is selected to maximize the logic in gated cofactors. This minimizes the shared logic which performs active computation all the time and which cannot be clock-gated. The control variable selection 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*

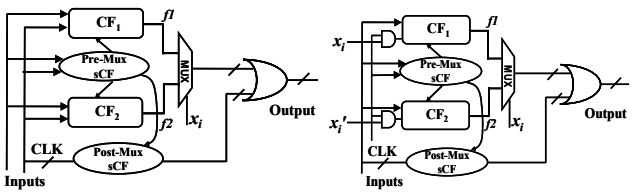

Fig. 3: (a) Circuit after application Shannon expansion (one level), (b) Dynamic Clock gating using Shannon expansion

from each output expression are initially combined together to determine the optimal control variable. One efficient approach for control variable selection for multi-output circuits is presented in [9].

Fig. 4 shows the optimal synthesis flow for one level of dynamic clock gating (DCG) using Shannon expansion. The Boolean expression of the logic circuit is taken as input in sum-of-products (SOP) format. In step 1, a conventional logic optimization (common sub-expression elimination, etc.) is performed on the input Boolean expression. We use a simple synthesis technique and technology map the resulting logic to a gate library consisting of AND gates, OR gates and static inverters. These static inverters are utilized to generate the inverted version of those inputs which are present in a SOP representation. Hence, the resulting SOP expression becomes a unate function with both the original and the inverted inputs present as primary inputs. The product terms are mapped to two input domino-AND gates, while the sums are computed with wide fan-in domino-OR gates (8-input, 16-input etc.), whichever is applicable.

Let us illustrate the above mapping with the following Boolean function: $f = x_1x_2x_3'x_4 + x_5x_6 + x_3x_7x_8x_9$. A static inverter is used to generate x_3'. Then $f_1=x_1x_2$, $f_2=x_3'x_4$, $f_3=x_5x_6$, $f_4=x_3x_7$ and $f_5=x_8x_9$ are mapped to domino-AND gates. The outputs f_1, f_2 and f_5, f_6 are again mapped to AND gates to generate $f_6=x_1x_2x_3'x_4$ and $f_7 = x_3x_7x_8x_9$. Finally the outputs f_3, f_6 and f_7 are OR-ed using a domino OR gate. This synthesis technique ensures that we do not have inverting logic inside the optimized Boolean representation and thus no re-convergence problem (and therefore no logic duplication) would happen. Once mapped, the resulting power and delay (P_{orig} and D_{orig}) are estimated in step 2. The power for the original circuit is compared with that obtained from DCG to determine power saving. The estimated delay after application of DCG is used to verify whether it satisfies the specified delay constraint.

Steps 3 to 8 of the flow illustrate the synthesis steps for DCG. The optimized logic function obtained in SOP format from step 1 is utilized to identify the optimal control variable in Step 3 and generate the corresponding cofactors (CF_1 and CF_2) and the shared logic (SL). Each of the cofactors (CFs) and shared logic (SL) are individually optimized also. Then, the expressions of Pre-Mux shared logic (logic common to the optimized cofactors and shared logic), Post-Mux shared logic (SOP terms not containing the control variable, shown in Fig. 3(a)), CF_1, and CF_2 are generated in Step 4. Considering the same function f, the control variable used for supply and clock gating is x_3, $CF_1 = x_7x_8x_9$, $CF_2 = x_1x_2x_4$ and Post-Mux shared logic= x_5x_6. These logic functions (CF_1, CF_2, SL) are separately synthesized and mapped to the technology library (AND, OR, inverter) in the same manner as the original circuit. The individually synthesized functions are merged together with MUX-OR logic as shown in Fig. 3(b). The corresponding delay (D_{level1} = *func*(critical path delay of one cofactor and MUX-OR logic)) and power (P_{level1} = Σ*func*(P_{CF1}, P_{CF2}, P_{SL}, P_{MUXOR})) are estimated from a graph representation of the combined logic.

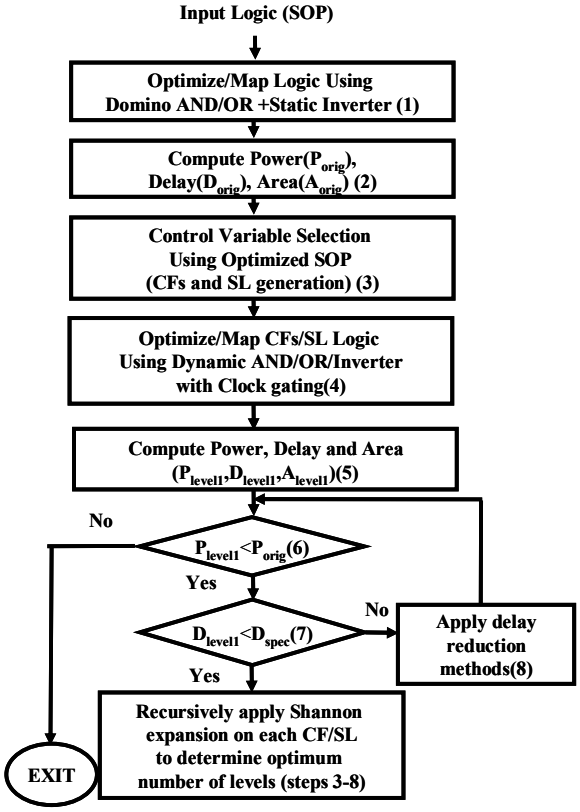

Fig.4: Optimal Low Power Synthesis technique for domino logic to achieve fine-grained clock gating

The estimated power of the first level expansion (P_{level1}) is compared the original design (P_{orig}) in step 6 to evaluate the power saving. If no power saving is achieved by DCG, clock gating is not used for current level of expansion. If there is power reduction, the delay (D_{level1}) is compared in step 7 with the given delay constraint (D_{spec}) to check if the DCG synthesized circuit meets the delay requirement. If the delay constraint is not met, methods such as reduction of shared logic can be applied and the delay/power conditions are rechecked. In case the power and delay conditions are satisfied, the circuit obtained by DCG at the current level is selected as the optimized output.

The recursive application of Shannon's theorem for multiple levels of expansion is similar to single level expansion. For subsequent levels, DCG is performed on the current level cofactors and shared logic. The individual cofactors and shared logic are taken as the input logic (SOP format) in each of the cases. Steps 3 to 8 of the synthesis flow are performed on each of them to determine whether it is effective to perform the DCG for the individual cofactors or the shared logic. Since the sizes of the cofactors and the shared logic progressively reduce with each expansion level, the overhead associated with the switching of the extra logic (multiplexers etc.) offset the power gains obtained with DCG after some point. Our synthesis technique determines the optimal level of expansion for each circuit.

It should be noted that other advanced synthesis techniques [10] for domino logic which enable more efficient mapping can be easily integrated to our synthesis flow. Since the same

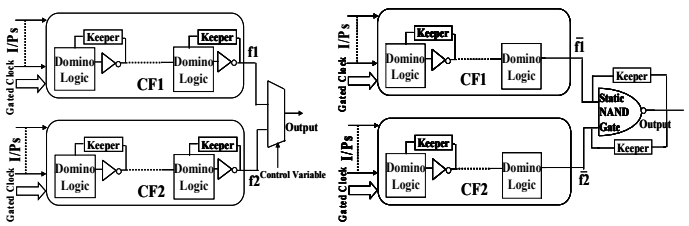

Fig. 5: (a) Original Shannon extended circuit (non-inverting domino-logic and mux) (b) Optimized circuit (inverter at final domino stage + mux replaced by static NAND gate)

technique would be applied to the original circuit and also the cofactors and shared logic generated by DCG, we expect to have similar gains in terms of power.

3.2. Area Optimization in Domino Logic

The non-inverting nature of domino logic allows us to replace the final stage inverters and the multiplexer by a single NAND gate as shown in Fig. 5(b). The operation of the two circuits is similar and can be explained as follows:

During the precharge phase, the outputs of the two cofactors, *f1'* and *f2'* are both precharged to a value of '1' (*f1* = *f2* = '0'). Therefore, irrespective of the value of the control variable the output of the multiplexer in Fig. 5(a) is '0'. The output of the static NAND gate in this case is also '0' since both *f1'=1* and *f2'=1*. In the evaluation phase, the final output is determined based on the conditional discharge of one of the cofactors. There can be two possibilities:

a) None of the cofactors evaluate to a '0' value. The output of the multiplexer (Fig. 5(a), *f1=0* and *f2=0*) remains unchanged at '0' and so does the output of the static NAND gate (Fig. 5(b), *f1'=1* and *f2'=1*),

b) One cofactor evaluates to a '0' (since at any instant only one cofactor is active). The output values of both the multiplexer-based and the static NAND implementations are identical in this case too. For instance, if the control variable is '1', CF_1 is activated and evaluates to a '0' value (*f1'=0*$\rightarrow$ *f1=1*). The output of the multiplexer in this case is '1' since the control variable chooses the output of the first cofactor. For the static NAND implementation, the output is also '1' since *f1'=0*. Since both the cofactors can never be simultaneously active, there is no possibility of both the cofactors evaluating to a '0' value.

This scheme provides area savings for single output circuits since less transistors are required (four for static NAND gate instead of eleven for inverters/ multiplexer combined). The multiplexer also has a high switching activity depending on the activity of the control variable and the cofactor outputs. This technique can, therefore, also reduce energy consumption since less number of transistors switch at any particular time instant (NAND gate compared to multiplexer). However, minimal area penalty and significant power improvement can be obtained for the following two cases:

- For multiple output circuits [9], each of the output multiplexers and last stage inverters can be replaced by NAND gates, reducing area and switching overhead.
- For circuits where we recursively apply Shannon's expansion to obtain multiple cofactors for enhanced power savings [9], the outputs of each pair of cofactors end in a multiplexer. We can replace the last stage inverters of and multiplexers with a static NAND gate.

We have incorporated this design optimization strategy in the automated synthesis flow for domino logic.

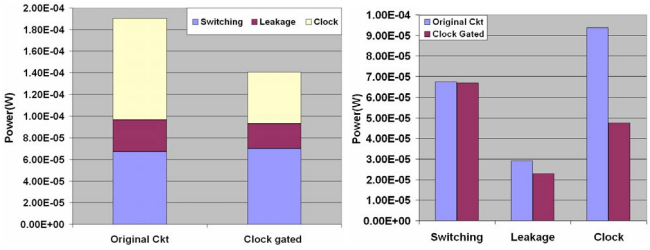

Fig. 6: (a) Total power dissipation in original circuit (OC) and the Clock Gated (CG) configurations (b) Switching, Leakage and Clock power for OC and CG configurations

3.3. Clock Gating in Domino Logic: A Case Study

In the following paragraphs, we analyze circuit level application of clock gating to domino circuits for power reduction, and evaluate the associated impact on delay and area using a standard MCNC benchmark circuit *cm150a*. It should be noted that the idle cofactors are always left in the precharge mode in our clock-gating strategy (clock is gated to '0'). Gating the clock to '0' prevents switching on the internal nodes despite switching at the primary inputs. We implemented the *cm150a* circuit using domino logic in BPTM 70nm technology and simulated using Hspice. The activity of all primary inputs has been kept at 50%.

The total and individual components of power consumption is shown in Fig. 6(a) and Fig. 6(b) respectively. The reduction of overall power in the CG mode can be attributed mainly to the reduction in clock power. The switching power for the CG mode is marginally less than OC for this benchmark. To analyze the effect of the Shannon expansion on switching power, we have to consider two competing issues. First, the average load capacitance at internal nodes presented by each cofactor is less than the original circuit. Also, redundant switching in the idle cofactor is eliminated. Therefore, switching power is expected to reduce for the cofactored circuits. On the other hand, for the CG configuration, there is extra switching associated with the gates AND-ing the clock and also switching overhead associated with some logic duplication due to circuit partitioning. This explains the observed nature of switching power results (Fig. 6(b)). However, this trend varies across benchmarks. For large benchmarks, where clock gating transistors can be shared across many logic gates, switching power associated with gating transistors is reduced.

The critical path delay results show that CG mode performs better than OC for *cm150a* circuit. However, the delay results may vary across different benchmarks. The delay is determined by three factors:

- average load at each internal node of the original and Shannon-expanded circuit,
- delay incurred in the clock gating transistors and the end multiplexer or the NAND gate (refer Section 3.2),
- wiring delay penalty at each level of expansion.

The CG configuration offers less loading on their internal nodes since it is divided into cofactors. However, there is extra wiring overhead each time the circuit is partitioned by Shannon expansion. The critical path delays for OC and CG configurations of *cm150a* are 210ps and 180ps respectively.

The area penalty in the CG case (because of gating of the clock signal and wiring overhead) for *cm150a* is around 5.4%. However, some benchmarks might have better logic optimization of their cofactors by Shannon expansion and hence total area reduces for these CG circuits [9].

One added advantage from the CG technique is that we reduce one of the domino noise sources –supply noise. This happens due to reduction of the supply current because of less switching action in each cofactor. The noise immunity of the CG circuit is thus improved with respect to the OC.

4. SYNTHESIS OF CLOCK-GATED SKEWED LOGIC

In this section, the clock-gating synthesis method developed for domino circuits has been extended for skewed logic circuits. The key differences between the automated synthesis of domino and skewed logic techniques are also highlighted in this section. The skewed version of benchmark circuit *cm150a* implemented with Shannon-based clock gating has been analyzed for power, area and delay.

4.1. Synthesis of Skewed Logic Circuits

The synthesis flow of skewed CMOS with dynamic clock gating (DCG) is different from that of domino logic as shown in Fig. 7. Initially, the circuit input in SOP format is optimized and mapped to a standard CMOS library. The mapped logic is then optimized using an integer linear programming-based approach to overcome the logic reconvergence problem in skewed logic circuits with minimal logic duplication cost [3]. The gates are then mapped using a skewed CMOS library and a dynamic programming-based heuristic is applied to achieve an optimal selective clocking scheme [3]. The power, delay and area of the circuit are then computed. To apply DCG, the control variable is selected using the optimized SOP from step 1 and the *CFs* and *SL* are generated. These *CFs* and *SL* are mapped to standard CMOS gates. The ILP-based approach to minimize logic duplication is applied to each of the *CFs* and *SL*. Then they are mapped to DCG-based skewed CMOS gates. Of course, optimal number of clocking levels is again determined for each of them. The rest of the synthesis flow is similar to that of domino logic in which power and delay of the original circuit is compared with the Shannon-based expanded circuit recursively to determine the optimum expansion levels as shown in Fig. 7.

4.2. Skewed Logic Implementation

The skewed implementation of *cm150a* consumes less power in CG mode than OC (Fig. 8(a)). The clock power savings obtained is less than its domino logic counterpart (Fig. 8(b)) since the gates in skewed circuit can be selectively clocked. It should also be noted that leakage power for CG configuration is much less than OC unlike the domino logic implementation. This is because the clocked gates at intermediate levels of the circuit in the skewed logic have clocked footer transistors. The footer transistor acts as a supply gating transistor for such gates reducing their leakage

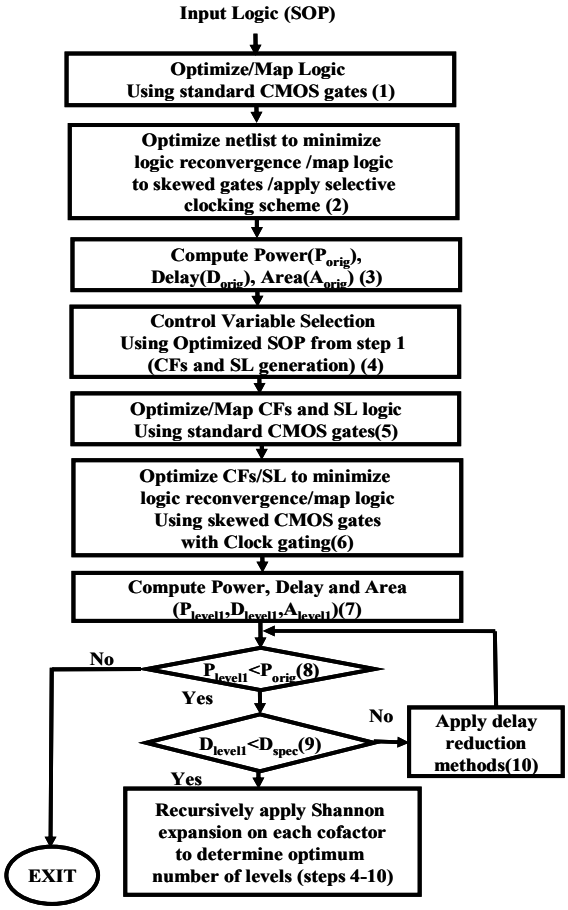

Fig. 7: Optimal Synthesis technique for skewed logic for fine-grained clock gating

power. The delay for the OC and the CG implementations are 215ps and 217ps respectively. Skewed circuits are inherently CMOS circuits with higher noise immunities as compared to domino. The CG configuration does not affect the noise margins of the skewed logic as compared to OC. However, reliability improves due to less supply current and less supply noise. The area increase is 1.55% for CG case.

5. RESULTS

The results of the proposed DCG synthesis approach on a set of MCNC benchmark circuits have been presented in this section. We have integrated our synthesis tool with SIS [11] to perform logic optimization. The benchmarks in SOP format are initially optimized by applying *script.rugged* several times. Similar optimization is performed for each of the cofactors and shared logic blocks to form a fair comparison in terms of area between the original and the DCG circuits.

5.1. Domino Circuits

After initial optimization with SIS, we technology map the SOP logic using a library consisting of inverter, 2-input AND gates and variable fan-in OR-gates (2, 3, 4, 6, 8, 16). We perform accurate power estimation by simulating the resulting netlists with Hspice. The corresponding delay is estimated by activating the critical path during evaluation mode. The area is computed by calculating active area for transistors and adding a suitable wire load factor for routing purposes. Similar optimization and mapping is also

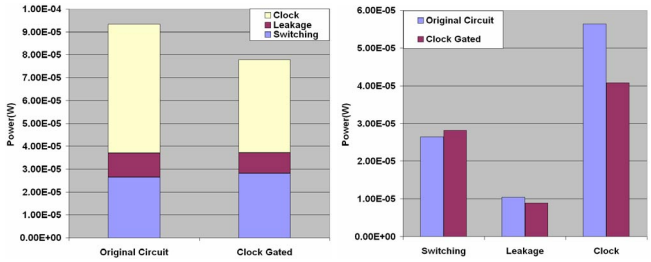

Fig. 8: Skewed CMOS: (a) Total power dissipation in original circuit (OC) and clock-gated (CG) mode (b) Switching, Leakage and Clock power for OC and CG configurations

performed on the DCG based cofactors and shared logic and the power, delay and area values are estimated. The recursive application of DCG in our synthesis tool provides optimal number of expansion stages.

The results of power, delay and area for the OC and CG circuits are compared in Table 1 for one level of expansion. The results show reduction of 20% to 64.8% in total power due to reductions in clock and switching components of power dissipation (Section 3). The delay improves in some cases after clock gating due to less effective loading on internal nodes. The area overhead varies between -8.7% (reduction) to +5.43% (increase). The reduction is attributed to better cofactor optimization, whereas increase is due to logic duplication and area of the clock gating transistors.

5.2. Skewed Circuits

The initial SOP optimization for skewed circuits is performed using *script.rugged* from SIS. The circuit is mapped to a library consisting of these skewed up and down gates: i) clocked and un-clocked inverters, ii) 2-input clocked and un-clocked NAND gates, iii) 2-input clocked and un-clocked NOR gates. The calculation of area and delay is performed similar to the domino synthesis case for

Table 1. Experimental results for domino logic circuit (70nm Process, Vdd=1V, Temp=100°C) (+ = reduction, -= increase)

MCNC CKT	Power(µW)		Delay (ps)		Area (µm²)	
	Conv.	DCG	Conv.	DCG	Conv.	DCG
count	846.1	397.2 (+53%)	899	839 (+6.6%)	28128	25672 (+8.7%)
cm150a	185.1	146.3 (+20.9%)	210	180 (+14.2%)	5023	5295 (-5.43%)
decod	95.46	56.6 (+41.9%)	133	109 (+18%)	3250	3523 (-8.4%)
alu2	2351	1670 (+ 29 %)	816	643 (+21.2%)	79400	76320 (-3.87%)
mux	313.1	224.4 (+28.4%)	215	223 (-3.7%)	10603	11050 (-4.2%)
cht	820.1	612.3 (+25.3%)	249	261 (-4.8%)	14900	14551 (+2.3%)
pcler8	342.6	244.4 (28.6%)	389	327 (+15.9%)	10900	11473 (-5.25%)
pcle	244.2	161.2 (+34%)	265	282 (-6.5%)	9250	9685 (-4.68%)
sct	240.2	115 (+52.2%)	811	789 (+2.7%)	8290	8495 (-2.47%)
b1	54.3	19.1 (+64.8%)	88	73 (+17%)	622	598 (+3.8%)

Table 2. Experimental results for skewed logic circuits (70nm Process, Vdd=1V, Temp=100°C) (+ = reduction, -= increase)

MCNC CKT	Power(µW)		Delay (ps)		Area (µm²)	
	Conv.	DCG	Conv.	DCG	Conv.	DCG
count	589.2	322.7 (+45.3%)	912	896 (+1.75%)	21096	20024 (+5.08%)
cm150a	93.3	77.8 (+16.2%)	215	217 (-0.1%)	4119	4183 (-1.55%)
decod	82.7	68.2 (+25.1%)	127	114 (+10.2%)	2762	2818 (-2.0%)
alu2	1775	1388 (+22%)	829	678 (+18.2%)	62726	63819 (-1.73%)
mux	149.1	124 (+16.8%)	210	224 (-6.7%)	9463	9361 (+1.07%)
cht	559.3	446.2 (+20.2%)	256	247 (+3.51%)	11771	12638 (+7.36%)
pcler8	245.7	201.1 (+18.1%)	401	377 (+5.98%)	9374	9522 (-1.57%)
pcle	127.1	99.2 (+22%)	276	273 (+1.08%)	7122.5	7651 (-7.42%)
sct	165.2	91.1 (+44.9%)	826	811 (+1.81%)	7461	7490 (-0.3%)
b1	29.3	15.1 (+48.5%)	93	86 (+7.5%)	516	503 (+2.52%)

the DCG based skewed circuits. Table 2 lists the power, delay, and area of the original skewed circuit and the DCG based skewed circuit for one level of expansion. We obtain savings in power (16%- 48%) with maximum delay penalty of 6.7% and maximum area overhead of 7.36%.

6. CONCLUSION

We have developed a fine-grained (at circuit and timing granularity) low overhead clock gating mechanism for dynamic logic styles. The technique results in significant reduction in total circuit power and hence, enhances the usefulness of dynamic circuit in high-speed applications. We also propose a logic synthesis approach based on Shannon expansion for dynamically clock gating the idle parts of dynamic logic circuits during active mode of operation. The methodology proposed in this paper holds good for any style of clock-driven dynamic logic circuit.

REFERENCES:

[1] J. Rabaey et al., Digital Integrated Circuits, 2nd Edition, *Prentice Hall*.
[2] A. Solomatnikov et al., Skewed CMOS: noise-tolerant high-performance low-power static circuit family, *IEEE TVLSI*, Vol. 10, pp. 469-476, 2002.
[3] A. Cao et al., Synthesis of skewed logic circuits, *ACM TODAES*, Vol. 10, pp. 205-228, 2004.
[4] H. Li et al., DCG: Deterministic clock-gating for low-power microprocessor design, *IEEE TVLSI*, Vol. 12, pp. 245-254, 2004.
[5] P. Babighian et al., A scalable algorithm for RTL insertion of gated clocks based on ODCs computation, *IEEE TCAD*, Vol. 24, pp. 29-42, 2005.
[6] R. Krishnarnurthy et al., High-performance and low-power challenges for sub-70 nm microprocessor circuits, *CICC*, pp. 12-15, 2002.
[7] A. Chandrakasan, Design of High-Performance Microprocessor Circuits, *IEEE Press*.
[8] S. Bhunia et al., A Novel Synthesis Approach for Active Leakage Power Reduction Using Dynamic Supply Gating, *DAC*, pp. 479-484, 2005.
[9] L. Lavagno et al., Timed Shannon Circuits: A Power-Efficient Design Style and Synthesis Tool, DAC, pp. 254-260, 1995.
[10] Zhao et al., Technology Mapping Algorithms for domino logic, *ACM TODAES*, Vol. 7, pp. 306-335, 2002.
[11] SIS, University of California at Berkeley.