Synthesis and Implementation of Active Mode Power Gating Circuits

Jun Seomun
Dept. of Electrical Engineering, KAIST
Daejeon 305-701, Korea

Insup Shin
Dept. of Electrical Engineering, KAIST
Daejeon 305-701, Korea

Youngsoo Shin
Dept. of Electrical Engineering, KAIST
Daejeon 305- 701, Korea

ABSTRACT

Active leakage current is much larger ($\sim 10\times$) than standby leakage current, and takes a large proportion (30% to 40%) of active power consumption. Active mode power gating (AMPG) has been proposed to extend the application of basic power gating to reducing active leakage; it relies on clock-gating signals to cut the power off a part of combinational gates. The problem to select those gates while integrity of circuit behavior remains intact has not been solved yet. We identify three constraints to solve this problem, namely functional, timing, and current constraints. The problem of synthesizing AMPG circuits is then laid out, and synthesis algorithm is proposed; a group of gates that can be power-gated by each clock-gating signal and the size of footer that is attached to the group constitute a synthesis output. The layout methodology for standard cell designs is proposed to assess AMPG circuits in area and wirelength. Experiments in 1.1 V, 45-nm technology demonstrate that active leakage is reduced by 16% on average compared to clock-gated circuits.

Categories and Subject Descriptors: B.6.3 [**Logic Design**]: Design Aids—*Optimization*; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*VLSI*

General Terms: Algorithms, Design

Keywords: Low power, active-mode power gating, active leakage

1. INTRODUCTION

The static component of CMOS power consumption is a result of device leakage current arising from various physical phenomena [1]. As opposed to dynamic component, which is present only when there is a useful switching computation, leakage current always flows in a circuit: when a circuit is switching or in a very short idle between consecutive switching (called active leakage), and when a circuit is in long idle (called standby leakage) without involving any computation. Active leakage is much larger than standby leakage (e.g. about $10\times$ for 100 MHz frequency in room temperature [2]). This is because device stacking, which significantly reduces leakage when more than one MOS devices that are in series are turned off, can take effect after long time, typically much larger than clock period. The difference between active and standby leakage becomes even bigger as clock frequency increases [2]. The importance of standby leakage is emphasized due to large proportion of standby period in overall operation time, especially in mobile devices. The active leakage, on the other hand, is important due to its net proportion in total active power consumption. Various results of quantitative analysis have been reported to signify the large proportion of active leakage in total active power consumption, e.g. 30% [3, 4] in 65-nm technology.

DAC'10, June 13-18, 2010, Anaheim, California, USA

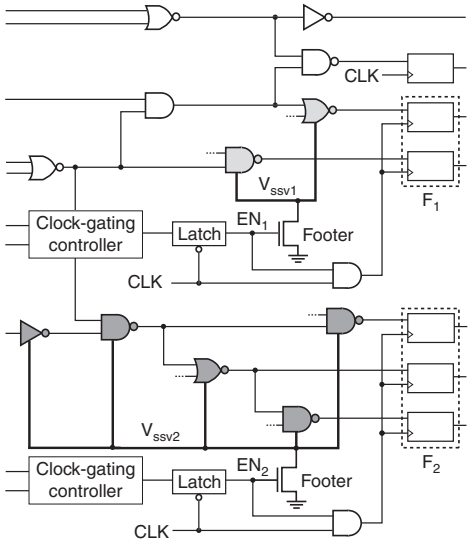

Figure 1: Active mode power gating circuit.

Many circuit techniques have been proposed to reduce leakage current. They can be classified according to the time when they are applied: static techniques are applied in design time while dynamic techniques need explicit control during run time. Dual-V_{dd}, dual-V_t, and dual-gate-length belong to static techniques; they are used to reduce active as well as standby leakage, but the amount of leakage that can be saved is relatively small (e.g. 42% for dual-V_{dd} [5] and 30% for dual-gate-length [6]). Power gating and body biasing are dynamic techniques; their leakage saving is substantial (e.g. $40\times$ using power gating [6]), but they are used to suppress only standby leakage.

Active mode power gating (AMPG) [7] has been proposed to extend the application of power gating to reducing active leakage. Its concept is illustrated in Figure 1. The EN_1 and EN_2 are clock-gating signals; when the value of each signal is 0, it disables CLK to corresponding flip-flops whenever there is no need of state change in those flip-flops. The decision is made by a clock-gating controller; the (negative level-sensitive) latch filters out any glitches the controller may generate. The logic gates that are shaded in

Figure 1 are connected to current switches, called footers, which are controlled by clock-gating signals. Thus, when EN_1 (or EN_2) disables CLK, it turns off the footer thereby suppressing leakage current through some combinational gates. Note that the shaded gates are responsible for only the inputs of the clock-gated flip-flops [7], i.e. they can be reached from the clock-gated flip-flops but not from any other flip-flops or circuit outputs, which we call *functional constraint*.

Besides functional constraint, there are two more constraints that have to be respected for correct operation, which we investigate in this paper. The first is *timing constraint*. Once footer is turned off due to EN_1=0 (or EN_2=0), the virtual ground V_{ssv} becomes floating which makes its potential increase toward V_{dd} (but very slowly); this in turn collapses all the logic values of the shaded gates. When footer is turned on again, the charges on V_{ssv} rails have to be drained while shaded gates evaluate their original logic values; these have to be performed by the rising edge of CLK that follows EN_1=1. The second is *current constraint*. Two factors have to be taken into account: rush current during wakeup and active-mode circuit delay. Sizing footer has to be done such that footer can drain maximum rush current that is allowed, and voltage drop across footer when circuit is actively switching is less than specified value. The AMPG synthesis problem, which we address, is to derive a set of gates that can be power-gated by each EN_i signal and to determine footer size such that all three constraints can be honored.

Our main contributions can be summarized as follows:

1. Quantitative analysis of active and standby leakage in 45-nm technology (Section 2).
2. Identifying three constraints of AMPG circuits and formulation of synthesis problem (Section 3.1), together with synthesis algorithm (Section 3.3).
3. Estimation of maximum and average discharge current to respect current constraint (Section 3.2).
4. Extensive experiments to assess active leakage; layout methodology for standard cell AMPG circuits and its evaluation in terms of area and wirelength (Section 4).

2. ACTIVE LEAKAGE

To understand the nature of active and standby leakage, we performed an experiment, which is similar to [2], using 2-input NAND gate in 1.1 V, 45-nm industrial technology. The result is shown in Figure 2(a). When inputs are 01, the internal node capacitance c_m is fully discharged. Once they become 00, M_1 starts to charge c_m using its leakage current, but this leakage is very small because, due to non-zero potential of c_m, M_1 is strongly cut off ($V_{gs} < 0$) and its effective threshold voltage becomes larger ($V_{bs} < 0$), which is called stacking effect. The corresponding transition of leakage takes a long time, long enough to exceed the clock period of typical designs; therefore, if NAND gate receives another value of inputs in the next clock period (say 10 ns), it never reaches the state of lowest leakage, which, on the other hand, can be reached in standby mode. The opposite transition (from 00 to 01) is fast, because c_m is discharged by turned-on M_2; the amount of active and standby leakage corresponding to this transition will be similar. The transition from 10 to 00 is faster than that from 01 to 00, because c_m is discharged by M_2, whose leakage current is larger than that of M_1.

In Figure 2(b), we report active and standby leakage of several ISCAS benchmark circuits. Each leakage was obtained by applying 100 random vectors and taking average via fast SPICE simulation [8]. The clock period was assumed at 10 ns and temperature

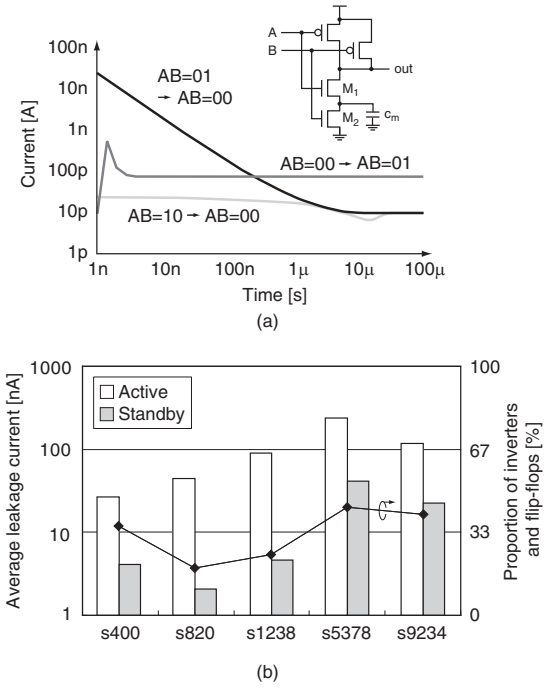

Figure 2: (a) Transient behavior of leakage in 2-input NAND gate, and (b) comparison of active and standby leakage in ISCAS benchmark circuits.

was set to 25°C. Active leakage is, on average, 12× of standby leakage, but there are some variations, e.g. 21× in s820 but 5× in s9234. Since the difference of active and standby leakage is caused by stacking effect, which happens when there are MOS devices in series such as in NAND or NOR, we count the number of inverters and flip-flops (flip-flop mainly consists of inverters, which are free from stacking effect) of each circuit. Their proportion in total gate count is reported in the right y-axis of Figure 2(b). Less stacking effect can be expected in s820, which explains its large difference of active and standby leakage; s9234, on the other hand, has higher proportion of gates that are free from stacking effect.

3. SYNTHESIS OF AMPG CIRCUITS

3.1 Problem Formulation

We are given a sequential circuit, which has clock-gating signals $EN_1, EN_2, \ldots, EN_n$ (see Figure 1). Signal EN_i enables or disables a clock to a set of flip-flops F_i, where F_is are disjoint. The synthesis problem of AMPG circuits is to derive a set of gates G_i that are power-gated by EN_i, where G_is are disjoint. The problem is subject to three constraints we have briefly mentioned in Section 1.

3.1.1 Functional Constraint

The gates that are responsible for only the inputs of flip-flops in F_i can be power-gated by EN_i, i.e. a gate in G_i has to be reachable only from flip-flops in F_i. We thus propagate a tag i from a flip-flop if it belongs to F_i toward primary inputs; we also propagate 0 from all circuit outputs and flip-flops that are not clock-gated. A gate that has only i, which is non-zero, as a tag becomes a member of G_i. We finally remove, from G_is, all the gates that are part of clock-gating controllers, because they have to be alive all the time.

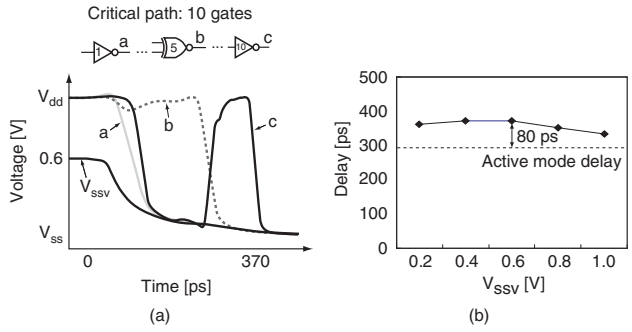

Figure 3: (a) Transient waveforms of V_{ssv} and internal nodes during wakeup and (b) wakeup delay for various V_{ssv}.

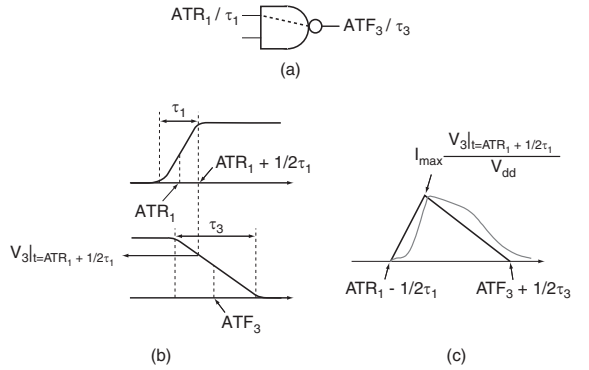

Figure 4: Modeling of discharge current: (a) 2-input NAND gate, (b) input and output waveforms, and (c) model of output discharge current overlapped with SPICE simulation.

3.1.2 Timing Constraint

In Figure 1, latches are opaque when CLK=1 so that any glitches, say static 1-hazard, from clock-gating controllers do not affect CLK being at 1. Therefore, EN_i is asserted only after falling edge of CLK, which implies that there is half a clock period* for V_{ssvi} to return to its nominal value (which we define as $\pm 5\%$ of V_{dd}) and for all the gates in G_i to return to their original logic values (the logic values before footer is turned off), which we define as wakeup delay.

It should be noted that the two components of delay, the delay for V_{ssvi} and the delay for G_i, do not add up to make the wakeup delay. This is illustrated in Figure 3(a) for an example c5315, one of ISCAS benchmark circuits. The potential of V_{ssv} is assumed at 0.6 V before footer is turned on at time 0; it returns to 5% of V_{dd} at 340 ps. We also plot the waveforms of three different nodes on a critical path. Their potentials during standby mode are all logic high; circuit input is assumed such that all these nodes make a falling transition. Note that they make transitions (node c make multiple transitions due to glitch) as they do in active mode when V_{ssv} is steadily close to V_{ss}, except that the delay of transition will be slightly larger due to transient V_{ssv}. The time when c completes its transition (thus the delay for G_i to return to the state right before standby mode) is 370 ps, while active mode delay is 290 ps. We performed the same experiment while we vary the potential of V_{ssv} from 0.2 V to 1.0 V as shown in Figure 3(b).

From the experiment of Figure 3 and comprehensive experiments with other circuits, we can conclude that wakeup delay is consistently close to active mode delay (with difference being less than 100 ps in 45-nm technology we tried), provided that V_{ssv} discharges fast enough as shown in Figure 3(a). Since we use a footer that is large enough thus can sink large amount of discharge current, so that the voltage drop across it is kept very small as presented in Section 3.1.3, this condition is always satisfied.

Once G_i is obtained following the functional constraint, we check its maximum delay using a static timing analysis (STA). If the delay plus guardband (100 ps in our experiment) exceeds half a clock period, we remove, from G_i, the gate that leads a critical path (gate 1 in Figure 3(a)); we repeat the process until timing constraint is satisfied.

3.1.3 Current Constraint

Once we determine G_i that satisfies functional and timing constraints, we have to determine the size of footer that will be attached to it. This is done by taking two factors into account: rush current during wakeup and active-mode circuit delay. Note that these are conflicting requirements: smaller size is preferred for less rush current, but larger size is required to keep the voltage drop across footer small so that footer does not affect circuit delay too much.

The rush current is typically constrained by the maximum discharge current (MDC) of G_i during active mode, since power rails are designed based on it. Therefore, once MDC is estimated, which we address in Section 3.2, we can accordingly choose the minimum size of footer that can accommodate MDC during wakeup.

To take care of active-mode circuit delay, we rely on average current method (ACM), which has been widely used [9–11]. In practical design, we often require the voltage drop across footer, ΔV, to be very small (say 1% of V_{dd}) so that employing power gating does not introduce any practical increase of delay. In this situation, it is empirically observed that ΔV is not strongly dependent on input patterns [9], i.e. average current can be used for sizing. Thus, we first derive average discharge current (ADC) of G_i, which we also address in Section 3.2. We then derive ΔV across footer, whose size has been determined from rush current constraint. If $\Delta V > \varepsilon V_{dd}$ for some ε (say 1%), we have to remove some gates from G_i until new ΔV (due to smaller ADC) becomes not greater than εV_{dd}. The details of this process will be explained in Section 3.3.

3.2 Estimation of Maximum and Average Discharge Current

3.2.1 Discharge Current Model

Consider 2-input NAND gate shown in Figure 4(a). Arrival time of rising input signal (ATR_1) and its transition time (τ_1) are given after performing STA; the other input is assumed at logic high. We want to estimate the output discharge current in this configuration.

Arrival time of falling output signal (ATF_3) along the timing arc shown in the figure, and its transition time (τ_3) are also obtained by STA. The waveforms of input and output signals are illustrated in Figure 4(b). The output discharge current is approximated as a triangular shape shown in Figure 4(c). As soon as input potential exceeds threshold voltage of nMOS devices, discharge current starts to flow (initially together with short-circuit current); this is approximated as the time when input transition starts, $ATR_1 - 1/2\tau_1$. The discharge completes at the end of output transition, $ATF_3 + 1/2\tau_3$. We make an approximation that discharge current is at its peak

*We assume 0.5 for duty ratio; we also assume that clock-gating controller completes logic evaluation before falling edge of clock. Note that these assumptions are only for convenience of presentation; extension for general case can be readily made.

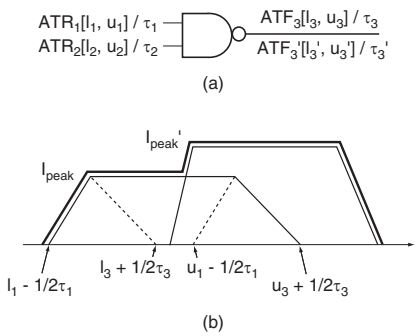

Figure 5: Estimation of MDC: (a) timing parameters of 2-input NAND gate, and (b) envelope of MDC.

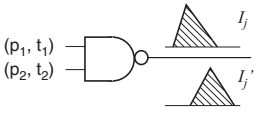

Figure 6: Estimation of ADC.

when input transition completes (i.e. when V_{gs} is at its maximum), $\text{ATR}_1 + 1/2\tau_1$ (see Figure 4(b)). The value of peak current is assumed to be proportional to the output potential, i.e.

$$I_{peak} = I_{max} \frac{V_3|_{t=\text{ATR}_1+1/2\tau_1}}{V_{dd}}, \quad (1)$$

where I_{max} is the maximum discharge current unique to the NAND gate, which is characterized a priori. The discharge current model is compared to SPICE simulation in Figure 4(c), which demonstrates a reasonable accuracy.

3.2.2 Estimation of MDC

Consider 2-input NAND gate again, as shown in Figure 5(a). Arrival time at inputs are given as a bound this time: l_i and u_i correspond to the earliest and latest ATR_i, respectively, returned by STA. The bound of ATF_3 corresponding to ATR_1 (with the second input tied to logic high), $[l_3, u_3]$, is derived. Using $[l_1, u_1]$ and $[l_3, u_3]$ yields two discharge current waveforms following the method in Section 3.2.1: one corresponding to lower bound and the other to upper bound. Two current peaks are then connected as shown in Figure 5(b) implying that we make a conservative assumption that peak discharge current continuously flows between two peaks. The process is repeated for the other input, i.e. the bound of ATF_3 corresponding to ATR_2 (with the first input tied to logic high this time), $[l_3', u_3']$, is derived, followed by computing a current waveform. We finally obtain an envelope of two waveforms as illustrated in Figure 5(b), which corresponds to maximum current of two waveforms.

3.2.3 Estimation of ADC

The process is repeated for all the other gates in G_i; the maximum discharge current of G_i can then be readily obtained. Our method is similar to [12], except that we use an accurate current model described in Section 3.2.1 while [12] uses the same current waveform for all the logic gates to simplify a process. Since we make an assumption that discharging is independent of input patterns, the derived MDC can be loose; extracting the information of mutually exclusive discharging [13] may yield smaller value of MDC.

Algorithm *AMPG_Synthesis*
L1 **for** each EN_i **do**
Functional constraint:
L2 Identify G_i that respects functional constraint
Timing constraint:
L3 **while** (delay of G_i + guardband) $> T_c/2$ **do**
L4 Remove the gate that leads a critical path of G_i
Current constraint:
L5 $I_{mdc} \leftarrow$ MDC of G_i
L6 Size footer such that its current when $\Delta V = V_{dd} < I_{mdc}$
L7 $I_{adc} \leftarrow$ ADC of G_i
L8 **while** $\Delta V|_{I=I_{adc}} > \varepsilon V_{dd}$ **do**
L9 Remove the gate j with max $I_{j,av}$ and leads G_i

Figure 7: Pseudo-code of AMPG synthesis.

We also use a discharge current model we developed in Section 3.2.1 to estimate average discharge current (ADC). Consider a 2-input NAND gate shown in Figure 6. We know the signal probabilities (the probability of signal being at logic high), p_1 and p_2, and transition probabilities, t_1 and t_2; these can be obtained by propagating signal probabilities at primary inputs [14], which are given by designers. At the output of NAND gate j, we derive two discharge current: I_j due to input 1 and I_j' due to input 2 (see Figure 4). The current I_j is caused when input 1 makes a rising transition, whose probability is $t_1/2$, and input 2 is at logic high, whose probability is p_2; similar reasoning can be applied to I_j'. The average amount of discharge current at node j is thus given by

$$I_{j,av} = \frac{t_1}{2} p_2 \int I_j dt + \frac{t_2}{2} p_1 \int I_j' dt. \quad (2)$$

The same process is repeated at all nodes of G_i to finally yield its ADC:

$$I_{adc} = \frac{\sum_j I_{j,av}}{T_c}, \quad (3)$$

where T_c is a clock period.

3.3 Algorithm

The overall algorithm to synthesize AMPG circuits is shown in Figure 7. Three constraints are checked one by one to derive a group of gates G_i that can be power-gated by EN_i and the size of footer that will be connected to G_i. Initial group of gates that respect functional constraint is obtained in L2. If the maximum delay of G_i returned by STA plus some guardband to accommodate delay increase due to transient V_{ssvi} is larger than half a clock period (L3), we heuristically remove a gate that leads a critical path (L4); the process repeats (L3) until timing constraint is satisfied. MDC is then obtained (L5), which drives footer sizing such that maximum amount of current footer can drain when it is turned on during wakeup (when $\Delta V = V_{dd}$) does not exceed rush-current constraint, which is equal to MDC (L6). ADC is obtained (L7); if the voltage drop across footer (ΔV) when G_i is actively switching is larger than some threshold (εV_{dd}), we heuristically remove a gate out of the gates that lead G_i, whose average discharge current is maximum (L9); the process then repeats (L8).

3.4 Implementation Aspects

3.4.1 Floating Prevention in Flip-Flop

In Figure 1, once the gates that have direct connection to flip-flops are power-gated, a large amount of short-circuit current can flow in those flip-flops since flip-flop inputs are floating, i.e. the

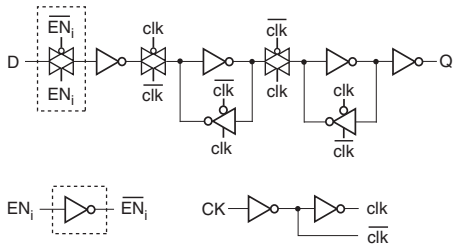

Figure 8: Floating-prevention flip-flop.

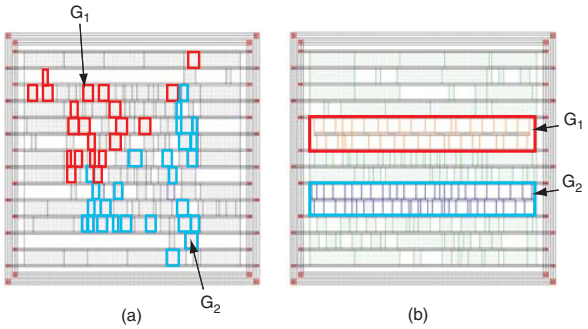

Figure 9: (a) Initial placement of s400 and (b) constrained placement after assigning rows to each G_i.

potential of flip-flop inputs may rise very slowly due to increasing virtual ground potential. This can be alleviated by introducing a transmission gate that isolates a flip-flop once EN is asserted as shown in Figure 8.

We implemented a new flip-flop in 45-nm technology: layout area increased by 18%, setup time increased 17%, and clock-to-Q delay remains the same. A flip-flop of better architecture could be devised to reduce the overhead, which is left for future investigation.

3.4.2 *Placement of Standard Cell AMPG Circuits*

In row-based placement when AMPG circuits are realized using standard cells, each G_i (as well as gates that are not power-gated) has to be placed in its own placement rows, since its virtual ground V_{ssvi} cannot be shared with any other G_is (see Figure 1). To this end, we first determine the number of rows that are required to place G_i based on the area of the cells in G_i. In order to apply double-back layout pattern, which helps reduce layout area, we try to assign even number of rows to each G_i and place them consecutively as shown in Figure 9(b). This may have adverse effect on wiring since the cells of G_i are localized in their placement, but it is not severe because G_i inherently consists of locally connected logic gates.

To determine how we interleave rows corresponding to different G_is, we perform an initial placement assuming that cells can be placed anywhere in the placement region, i.e. we run conventional placement. An example is shown in Figure 9(a). We then identify the row that is most populated by the cells belonging to G_i, and heuristically assign that row exclusively to G_i; we continue until the number of rows that we assign to G_i matches the number of rows that are required for G_i. Once rows are assigned to all G_is, we run constrained placement to derive a final layout (see Figure 9(b)), which is then submitted to routing.

Table 1: Benchmark circuits

Name	#Gates	#FFs	#POs	#ENs	$\cup_i FIC(F_i)$	$\Sigma\|G_i\|$
s400	126	21	6	2	99	51
s1238	427	18	16	2	56	36
s1423	573	74	5	2	397	352
s5378	854	176	49	2	323	193
s9234	784	145	39	3	433	335
s15850	2416	513	150	3	1321	1259
s35932	5590	1728	288	5	4610	3993
s38584	7118	1275	304	5	4394	4214

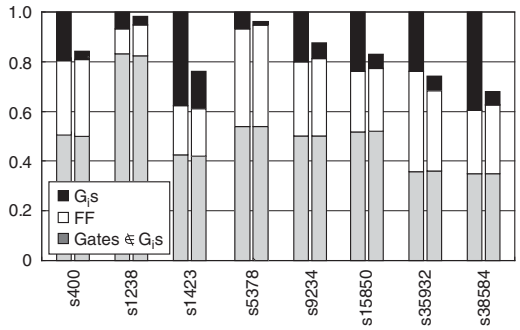

Figure 10: Comparison of (normalized) active leakage between clock-gated (left-hand bars) and AMPG circuits (right-hand bars).

4. EXPERIMENTS

We carried out experiments on a set of sequential circuits taken from the ISCAS benchmark. The second and third columns of Table 1 report the number of combinational gates and the number of flip-flops of each circuit, which was synthesized [15] with commercial 1.1 V, 45-nm bulk CMOS technology.

The fifth column reports the number of clock-gating signals, which were automatically synthesized [15] followed by manually merging some of them when too many signals are generated. Column 6 indicates the total number of gates that are in the fan-in cone of clock-gated flip-flops (F_i), which is equal to the total number of gates that respect functional constraint; the difference between columns 2 and 6 is caused by the gates that belong to clock-gating controller and the gates that are in the fan-in cone of primary outputs or flip-flops that are not clock-gated. The last column shows the total number of gates that were determined to be power-gated after running our synthesis algorithm shown in Figure 7, which was implemented in OpenAccess [16]. The difference between columns 6 and 7 is due to timing and current constraints.

4.1 Active Leakage

In Figure 10, we report active leakage of AMPG circuits (right-hand bars), which is normalized to that of original clock-gated circuits (left-hand bars). In each bar, three sources of leakage are identified: the gates that can be power-gated (G_is), flip-flops, and the remaining combinational gates, which are never power-gated. Total active leakage is reduced by 16% on average. Note that only the gates that belong to G_is can benefit the saving of active leakage; the proportion of their leakage in clock-gated circuits is 23% on average. The small proportion of G_is in total leakage as well as in gate count can be understood from relatively large number of pri-

Table 2: Comparison of area, total wirelength, and average congestion of clock-gated (CG) and AMPG circuits

Name	Area (μm^2)			Wirelength (μm)			Average congestion (%)			% of
	CG	AMPG	Inc. (%)	CG	AMPG	Inc. (%)	CG	AMPG	Diff.	V_{ssv}-rows
s400	186	195	5	896	1192	33	7	10	3	31
s1238	390	398	2	3545	4121	16	14	17	3	25
s1423	788	853	8	4781	5495	15	12	13	1	44
s5378	1439	1522	6	10135	13582	34	14	18	4	17
s9234	1258	1328	6	8982	11285	26	14	17	3	18
s15850	4656	5118	10	36688	48786	33	19	28	9	26
s35932	12524	13519	8	89138	139514	57	16	28	12	28
s38584	11646	12458	7	132938	173134	30	27	32	5	28
Average			6			30			5	27

mary outputs reported in the fourth column of Table 1. All the gates that belong to the fan-in cone of primary outputs are excluded from G_is, since they cannot be power-gated. Therefore, the circuits that have small number of primary outputs and large number of clock-gated flip-flops can enjoy more saving in active leakage, which is the case in s1423 (24% of saving). Active leakage of G_is alone is reduced by 71% on average.

4.2 Area and Wirelength

In Table 2, we compare clock-gated circuits and AMPG circuits in terms of area and wirelength, as well as wiring congestion. The layout of AMPG circuits was obtained following the procedure explained in Section 3.4.2. The area is the sum of the areas of all the cells in the design; utilization of placement area was set to 70% during automatic placement. Metal layers up to M6 were allowed for routing.

The area increases by 6% on average, which is a result of increased area of flip-flops (to prevent floating) and footer cells. The wirelength, on the other hand, increases significantly, 30% on average; this is mainly due to restricted placement and heuristic way to interleave rows corresponding to different G_is, which we presented in Section 3.4.2. The increase of average wiring congestion, however, is not too bad. The proportion of rows that are used by G_is is reported in the last column.

5. CONCLUSION

We have presented a method to synthesize AMPG circuits. The key components in synthesis are the three constraints (functional, timing, and current), which, when respected, make it possible to apply power gating during active mode while correct functioning of a circuit is guaranteed. The amount of saving in active leakage, 16% on average, is promising considering that power gating is applied to a small proportion of total combinational gates.

The impact on physical design, especially on wirelength, however calls attention. A new placement algorithm specific to AMPG circuits or taking physical design into account during AMPG synthesis may alleviate this impact, which is left for future investigation. Another direction of future research is to consider clock gating and AMPG synthesis as a whole, i.e. if we extract clock-gating signals such that AMPG can be better applied to a circuit, we could expect more saving on active leakage. The trade-off between switching power saved by clock gating and active leakage saved by AMPG should be taken account.

References

[1] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[2] Y. Ye, S. Borkar, and V. De, "A new technique for standby leakage reduction in high-performance circuits," in *Proc. Symp. on VLSI Circuits*, June 1998, pp. 40–41.

[3] H. Mair et al., "A 65-nm mobile multimedia applications processor with an adaptive power management scheme to compensate for variations," in *Proc. Symp. on VLSI Circuits*, June 2007, pp. 224–225.

[4] S. Rusu et al., "A 65-nm dual-core multithreaded Xeon processor with 16-MB L3 cache," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 17–25, Jan. 2007.

[5] Y. Shimazaki, R. Zlatanovici, and B. Nikolic, "A shared-well dual-supply-voltage 64-bit ALU," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 494–500, Mar. 2004.

[6] P. Royannez et al., "90nm low leakage SoC design techniques for wireless applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2005, pp. 138–139.

[7] K. Usami and H. Yoshioka, "A scheme to reduce active leakage power by detecting state transitions," in *Proc. Int. Midwest Symp. on Circuits and Systems*, July 2004, pp. 493–496.

[8] Synopsys, "NanoSim User Guide," Dec. 2007.

[9] S. Mutoh, S. Shigematsu, Y. Gotoh, and S. Konaka, "Design method of MTCMOS power switch for low-voltage high-speed LSIs," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 1999, pp. 113–116.

[10] H.-S. Won et al., "An MTCMOS design methodology and its application to mobile computing," in *Proc. Int. Symp. on Low Power Electronics and Design*, Aug. 2003, pp. 110–115.

[11] C. Hwang, P. Rong, and M. Pedram, "Sleep transistor distribution in row-based MTCMOS designs," in *Proc. Great Lakes Symp. on VLSI*, Mar. 2007, pp. 235–240.

[12] H. Kriplani, F. Najm, and I. Hajj, "Maximum current estimation in CMOS circuits," in *Proc. Design Automation Conf.*, June 1992, pp. 2–7.

[13] C. Hsieh, J. Lin, and S. Chang, "Vectorless estimation of maximum instantaneous current for sequential circuits," *IEEE Trans. on Computer-Aided Design*, vol. 25, no. 11, pp. 2341–2352, Nov. 2006.

[14] S. Ercolani et al., "Estimate of signal probability in combinational logic networks," in *Proc. European Test Conf.*, Apr. 1989, pp. 132–138.

[15] Synopsys, "Design Compiler User Guide," Mar. 2007.

[16] OpenAccess, ," 2009, http://www.si2.org/.