Active-Mode Leakage Reduction with Data-Retained Power Gating

Andrew B. Kahng[†+], Seokhyeong Kang[†] and Bongil Park[‡]
[†]ECE and [+]CSE Departments, University of California at San Diego, [‡]Samsung Electronics Co., Ltd.
abk@cs.ucsd.edu, shkang@vlsicad.ucsd.edu, bongil.park@samsung.com

***Abstract*—Power gating is one of the most effective solutions available to reduce leakage power. However, power gating is not practically usable in an active mode due to the overheads of inrush current and data retention. In this work, we propose a *data-retained power gating (DRPG)* technique which enables power gating of flip-flops during active mode. More precisely, we combine clock gating and power gating techniques, with the flip-flops being power-gated during clock masked periods. We introduce a retention switch which retains data during the power gating. With the retention switch, correct logic states and functionalities are guaranteed without additional control circuitry. The proposed technique can achieve significant active-mode leakage reduction over conventional designs with small area and performance overheads. In studies with a 65nm foundry library and open-source benchmarks, DRPG achieves up to 25.7% active-mode leakage savings (11.8% savings on average) over conventional designs.**

I. Introduction

The use of clock gating and power gating to reduce dynamic power and static leakage, respectively, is well-understood by both researchers and IC designers [1]. *Clock gating* is considered to be one of the most effective techniques to reduce dynamic power, and its automatic application is supported by EDA tools [2]. Clock gating masks the clock signal when the corresponding circuits are not performing useful computations. *Power gating* [3] drastically reduces leakage power by introducing a switch between the voltage supply (and/or ground) and a given block of functional circuitry; the block's leakage is stopped when the switch cuts off the current path from supply to ground.

To reduce active-mode leakage power, several approaches have been reported which combine clock gating and power gating [4], [5], [6], [7], [8]; we review these in Section II below. However, these previous approaches have associated design complexity and overhead issues which limit their practical implementation.

In this paper, we propose a new circuit-level technique which enables power gating of flip-flops during active mode. We combine both clock gating and power gating, such that flip-flops are power-gated during clock masked periods. The key contributions of our work are the following.

- The proposed technique enables concurrent clock and power gating, and thus achieves significant leakage power reduction during active mode.
- We introduce a *data retention switch* which sustains the voltage level of virtual ground to retain data in flip-flops.
- We provide empirical confirmation of the leakage power reduction achieved by the proposed technique over conventional power gating approaches.

The rest of this paper is organized as follows. Section II reviews previous works and their limitations. Section III presents the proposed data-retained power gating technique. Section IV provides experimental results and analysis. Section V summarizes and concludes the paper.

978-3-9815370-0-0/DATE13/©2013 EDAA

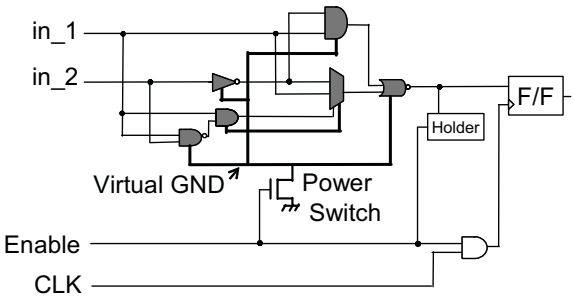

Fig. 1. Basic structure for Run-Time Power Gating [4].

II. Related Work

Power gating is the most effective available technique to reduce standby leakage, with benefits that are magnified by the increasing fraction of overall IC lifetime that modules spend in standby mode. With technology scaling, *active-mode leakage* becomes an increasingly significant portion of total dynamic power. Usami et al. [4] propose Run-Time Power Gating (RTPG) to extend the application of power gating to active-mode leakage reduction. Figure 1 shows the basic structure of RTPG. The enable signals of a gated clock design are exploited to control power switches for combinational logic gates. When the clock enable signal is 0, the power switch is turned off and active-mode leakage is cut off. The holders keep the input voltage of non-power-gated circuits.

Several design (synthesis and layout) flows have been proposed for RTPG implementation. Bolzani et al. [5] present a synthesis flow to combine power gating and clock gating. They partition the circuit into a number of clusters that are clock-gated by the same registers. Li et al. [7] propose an activity-driven optimization for RTPG which integrates clock gating and power gating based on input data. Seomun et al. [8] provide a synthesis and physical design (placement) flow for RTPG circuits.

While RTPG can effectively reduce active-mode leakage power of combinational logic, the approach has several inherent limitations that hamper practical implementation. First, the RTPG approach significantly increases design complexity. Each cluster of gates requires its own control signal to control power gating transistors. Other overheads include special buffer trees using real power network, high-fanout synthesis, power routing for the buffers, and so on. Further, the large number of virtual ground rails must be mutually isolated as well. Second, RTPG implementation incurs significant area overheads from its design complexity and additional circuits, e.g., bus holder circuits. Third, inrush current from power gating can diminish the amount of leakage reduction. If the clock-masked period is short or if flip-flop data is frequently changed, then RTPG will not be applicable due to the inrush current overhead.

Fukuoka et al. [11] present a clock gating scheme for partially-depleted SOI, which controls V_{th} of each transistor by body biasing

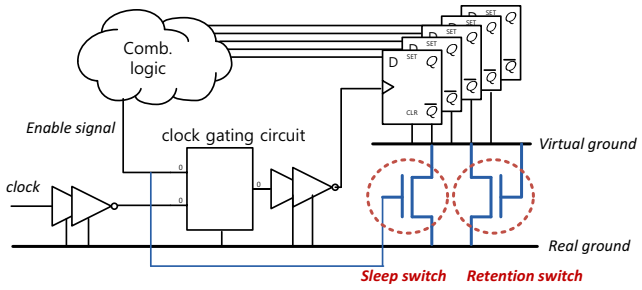

Fig. 2. Proposed circuits to combine clock gating and power gating.

associated with the clock gating signal. Their approach reduces active-mode leakage of flip-flops with the dynamic body biasing. However, the body biasing technique requires significant design and area overheads from the biasing circuits and voltage regulators.

Kim et al. [9] have proposed a tri-mode power gating approach that provides a choice between a large leakage reduction without data retention (IDLE) and an intermediate level of leakage reduction with data retention (PARK). The authors of [9] add a single PMOS switch to an NMOS footer switch in parallel to provide the intermediate power-saving mode. However, their intermediate mode is applied to an entire submodule, and cannot be used for a fine-grained RTPG approach. We exploit the idea of the intermediate power gating, and apply a similar approach into our RTPG.

III. Data-Retained Power Gating

A. Integrated Clock and Power Gating

Most commercial synthesizers [15], [17] support automatic insertion of clock gating logic without any modification of RTL codes. The inserted clock gating logic has clock gating control and enable signals. Clock signals are transparent during enable periods, and masked during disable periods. During a given masked period, the state of clock-gated flip-flops stays unchanged. As a consequence of recent product architectures as well as commercial synthesis tools' capabilities, flip-flops are masked for most of the IC's running time [10]. This offers an immediate motivation: If we could apply a power gating scheme to flip-flops during this masked period, then we could reduce active-mode leakage power. However, active-mode power gating requires that internal data state be retained, and according to existing practice, this requires huge overheads on both operation (e.g., data control to save and restore) and circuit design (e.g., retention flip-flops).

In this work, we introduce a new switch circuit to combine clock gating and power gating as shown in Figure 2. In the figure, the switch consists of two transistors; one is a normal sleep switch and the other is a retention switch. When the clock gating is disabled, the sleep switch is off. However, the retention switch induces a threshold voltage drop between virtual ground and real ground. This voltage drop reduces the operating voltage of flip-flops and leakage current. However, the flip-flops can retain the previous state with the reduced voltage.

The idea of a retention switch has been previously proposed by Kim et al. [9], as mentioned above in Section II. However, their technique requires additional layout area to implement N-well for the PMOS transistor. The PMOS can be replaced with an NMOS transistor by connecting the source and gate terminals to virtual ground. With such an approach, although the virtual ground may rise up to $V_{n,th}$ (NMOS threshold voltage), the flip-flops can retain state with reduced leakage.

Figure 3 shows HSPICE simulation results for data-retained power gating of a DFQ flip-flop in TSMC 65GP technology. Figure 3(a)

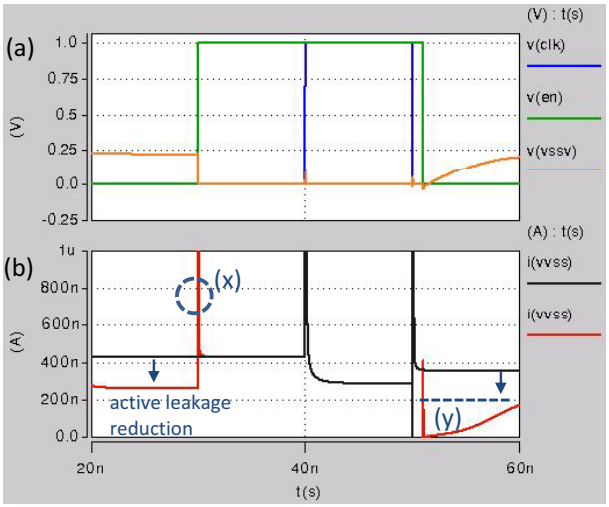

Fig. 3. HSPICE results for DFQ (TSMC 65GP) cell. (a) Gated-clock (clk), clock enable (en) signals and virtual ground voltage (vssv). (b) Current plot on VSS (black: w/o power gating, red: DRPG).

shows the voltage of virtual ground according to the clock enable signal (*en*). Figure 3(b) shows the current (on VSS) of the flip-flop for the DRPG and conventional (no power gating) cases. During the clock- and power-enabled period ($en = 1$), both cases show the same leakage power consumption. During power-gated (clock-disabled) periods ($en = 0$), the proposed retention switch sustains the voltage of virtual ground (0.25V) and enables retention of the internal status of logic. The supply voltage (0.75V) during the power-gated periods is sufficient to retain the flip-flop data [12]. With the increased virtual ground voltage, our power gating approach achieves significant leakage savings (35%).

With conventional power gating, the voltage of virtual ground goes to supply voltage, which causes a large inrush current upon wake-up. It is because of this inrush current that conventional power gating is not suitable for use during active mode. However, in our approach, inrush (discharge) current is small ((x) in Figure 3) due to the suspended virtual ground voltage, and the inrush current overhead is compensated during the idle state ((y) in Figure 3).

B. Flip-Flop Implementation

The suspended virtual ground affects the output value of the flip-flop during power gating. Non-zero output value causes significant leakage overhead on the flip-flop's fanout cells. To solve this problem, we add a level-shifter circuit into the flip-flop. Figure 4 shows a schematic of the proposed flip-flop circuit. We add $P0$, $N0$ and $N1$ switches into the conventional flip-flop circuit to adjust the voltage level of output port (Q).

A conventional level shifter has significant delay and area overhead. For example, HSPICE-measured delay overhead can be over 400ps in a 65nm LP process at worst corner; such a delay impact cannot be ignored. We observe that a conventional level shifter changes operating voltage level between two different operating voltages, while the proposed circuit changes ground level from $V_{n,th}$ to $0V$. Considering this requirement, we can move the level shifter circuit location from the output to the input of the final buffer. This change reduces the delay overhead, and also allows use of minimum transistor size in the implementation.

When the gate voltage of the *Pinv* transistor is $V_{n,th}$, *Pinv* turns ON and Q will be VDD. Hence, the $P0$ and $N0$ transistors turn OFF and $N1$ is completely ON. Finally, the gate voltage level of *Ninv* transistor goes to $0V$. On the other hand, when the gate voltage of

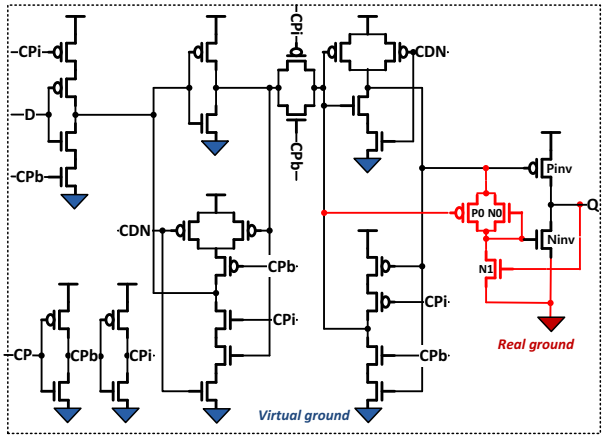

Fig. 4. Flip-flop implementation with a level shifter. We add $P0$, $N0$ and $N1$ switches to adjust the voltage level of output port (Q).

the *Pinv* transistor is VDD, then the gate voltage of $P0$ is low and $P0$ completely turns ON. Hence, the gate voltage of $N0$ will go high and the *Ninv* transistor turns ON. Finally, the output voltage of the final inverter is $0V$ and $N1$ transistors will turn OFF. The transistor ratio of $P0||N0$ and $N1$ should have a large value to minimize delay overhead. We have empirically determined transistor sizes based on HSPICE simulation results. Since $N1$ is only used to achieve $0V$ for the gate voltage of *Ninv*, its transistor width is minimum ($120nm$). Widths of $P0$ and $N0$ are $400nm$ and $200nm$, respectively, in the TSMC 65GP process.

With the additional devices, the flip-flop has a delay overhead, which we examine in detail in Section IV-B below.

C. Physical Implementation

During standard-cell placement, flip-flops driven by the same clock gating logic are placed within a bounded region. In other words, since they are tightly coupled to each other and have the same clock behavior, commercial P&R tools place them closely together. In addition, the clock gating logic is placed near its related flip-flop cluster – e.g., in the center of the cluster. Thus, a sleep control signal (enable signal of clock gating logic) requires just one or two buffers to control the sleep switch transistors, and can immediately turn on the sleep switches. To guarantee the correct operation of DRPG, flip-flops should be woken up before the arrival time of the clock signal that comes from clock gating logic. The feasibility of DRPG is validated in Section IV-B.

Our data-retained power gating can be implemented with global power gating (data is not retained) as shown in Figures 5(a) and (b) for the header switch and footer switch cases. For the footer switch case, additional *AND* gates are required. *PGEN* is a global power gating enable signal and *CKEN* is a clock enable signal. When DRPG is combined with global power gating, flip-flops will have three modes – (1) active mode ($PGEN = 1$ & $CKEN = 1$), (2) retention mode ($PGEN = 1$ & $CKEN = 0$) and (3) standby mode ($PGEN = 0$).[1]

Some modern design methodologies use multi-bit flip-flop cells, which can reduce physical design overhead since each can be treated as a single standard cell. This is also amenable to data-retained power gating by including sleep and retention switch inside as shown in Figure 5(c). Global power gating switch is not included in the standard cell implementation, and can be connected as shown in Figure 5(a). Figure 6 shows a physical layout of four-bit DRPG flip-flop. In this layout, four DRPG flip-flops share a single sleep switch, which is controlled by a clock enable signal.

[1]In standby mode, current paths from supply to ground are cut off with conventional power gating. In this paper, we do not address advantages and overheads of conventional power gating techniques, since they have been extensively studied in previous works (e.g., [3]).

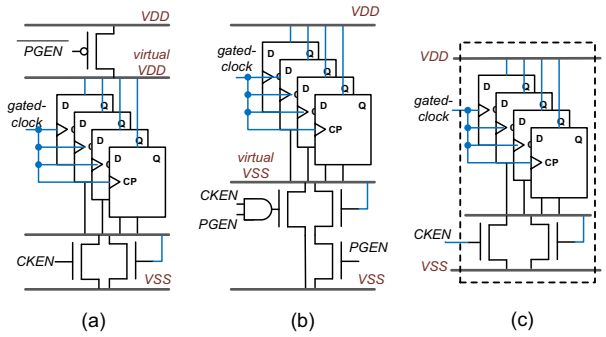

Fig. 5. Implementation example of DRPG – (a) global power gating with header switches, (b) global power gating with footer switches, and (c) standard cell implementation for a multi-bit flip-flop. [PGEN: global power gating enable; CKEN: clock enable signal.]

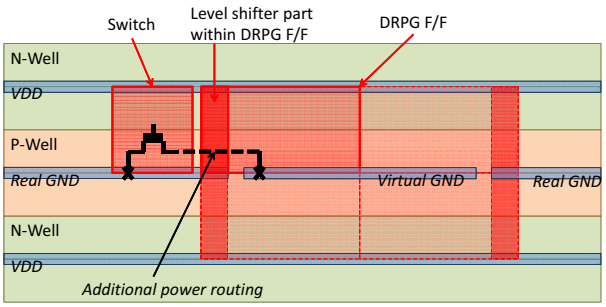

Fig. 6. Physical layout of DRPG flip-flop.

IV. Experimental Setup and Results

To analyze leakage power, cell delay and functionality of the proposed power gating, we perform circuit-level and design-level experiments. We implement our data-retained flip-flop with multi-V_{th} (HVT, NVT and LVT) and gate-length biasing, and evaluate delay and leakage power consumption of the implemented flip-flops (Section IV-B). We compare our data-retained flip-flop and a conventional retention flip-flop when they are used for the DRPG technique (Section IV-C). Finally, with design-level implementations, we provide empirical confirmation for the leakage reduction of DRPG (Section IV-D).

A. Experimental Setup

For the circuit-level experiments, we implement SPICE netlists of the proposed flip-flops (Figure 4) using TSMC 65GP SPICE models. To measure the cell delay and leakage power of implemented circuits, we use *Synopsys HSPICE vE-2010.12* [18]. For the design-level experiments, we use 11 open-source designs from the *OpenCores* site [16]. We use a TSMC 65GP cell library for the design implementation, and timing library models (*Synopsys Liberty*) for our data-retained flip-flops are prepared using *Cadence Library Characterizer v9.1* [13]. We synthesize the designs using *Synopsys DesignCompiler vF-2011.09* [17] and perform place-and-route with *Cadence Encounter Digital Implementation System v9.1* [14]. During synthesis, we use the clock-gating optimization of DesignCompiler, which inserts clock-gating cells automatically. We execute leakage optimization in DesignCompiler to replace clock-gated flip-flops with our data-retained flip-flops. After the placement and routing, we perform a post-layout leakage optimization with *UCSD SensOpt*.

TABLE I
DELAY, LEAKAGE AND AREA RESULTS OF PROPOSED DATA-RETAINED FLIP-FLOPS.

flip-flops		delay (ns)		delay overhead		single flip-flop			multi(8)-bit flip-flop		
V_{th}	cell-type	rising	falling	rising	falling	leakage (uW)	leakage reduction	area overhead	leakage (uW)	leakage reduction	area overhead
HVT	SDFQ	0.172	0.173	9.5%	19.3%	0.134	33.8%	15.0%	0.735	53.5%	8.0%
	SDFCNQ	0.190	0.177	4.3%	18.6%	0.140	36.4%	15.2%	0.750	56.5%	8.1%
	SDFSNQ	0.189	0.178	3.7%	19.5%	0.141	35.9%	17.6%	0.756	56.1%	9.4%
NVT	SDFQ	0.142	0.143	12.5%	20.7%	0.377	32.8%	15.0%	2.587	41.7%	8.0%
	SDFCNQ	0.168	0.146	14.3%	20.1%	0.416	34.5%	15.2%	2.823	43.9%	8.1%
	SDFSNQ	0.148	0.152	11.9%	20.1%	0.422	33.8%	17.6%	2.863	43.3%	9.4%
LVT	SDFQ	0.130	0.124	15.0%	19.1%	0.910	38.8%	15.0%	5.383	45.7%	8.0%
	SDFCNQ	0.155	0.127	16.9%	18.7%	0.982	42.0%	15.2%	5.737	49.6%	8.1%
	SDFSNQ	0.134	0.132	14.3%	18.2%	1.008	40.7%	17.6%	5.947	48.2%	9.4%

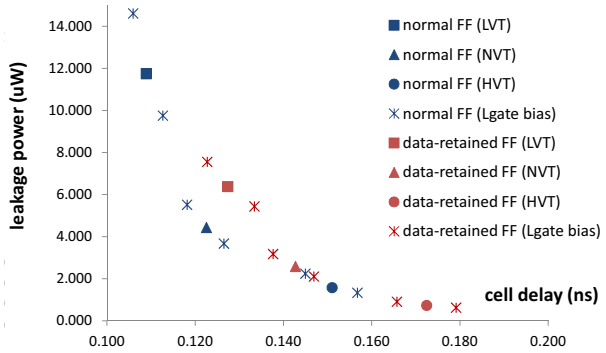

Fig. 7. Delay and leakage power comparison for normal flip-flops and data-retained flip-flops (multi(8)-bit SDFQ flip-flop).

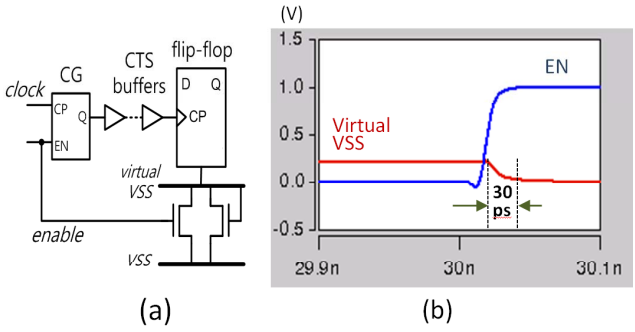

Fig. 8. (a) Clock and enable signal connections for data-retained power gating; (b) waveform of the clock enable signal and virtual ground voltage (SDFQ cell).

B. Circuit-Level Implementations

We implement three types of flip-flops; *SDFQ* (D flip-flop with scan input), *SDFCNQ* (D flip-flop with scan and asynchronous reset signal) and *SDFSNQ* (D flip-flop with scan and asynchronous set signal) with the proposed level-shifter circuit. We also implement HVT (high V_{th}), NVT (normal V_{th}) and LVT (low V_{th}) type versions for each flip-flop. We perform SPICE simulations for the implemented flip-flops with sleep and retention switches as shown in Figure 2. Table I shows *clock*-to-*Q* delay, cell leakage and area information of the implemented flip-flops. Delay overheads and leakage reductions are compared with those of the conventional versions of the flip-flops. The area overheads include the sleep, retention switches and the level-shifter circuit.

We measure the data for both the single flip-flop case and the multi(8)-bit case in which eight flip-flops share a sleep and retention switch together. Commercial synthesizers insert clock gating cells considering the number of driving flip-flops to maximize dynamic power reduction. Typically, the number would be larger than four. We consider the multi(8)-bit case specifically since most data processing modules treat byte-based data. From the results, our proposed flip-flops can reduce active-mode leakage power by 36.5% with 15.9% area overhead on average with the data-retained power gating. When eight flip-flops are implemented in the same cluster (or multi-bit flip-flop is assumed), we can achieve further leakage reduction with smaller area overhead by sharing the sleep and retention switches. The clustered (or multi-bit) flip-flops show 48.7% leakage reduction with 8.5% area overhead, on average. Due to the level-shifter circuit, the proposed flip-flops have an average of 15.4% delay overhead over the corresponding conventional flip-flops.

Figure 7 shows the delay and leakage comparison for normal flip-flops and data-retained flip-flops. From the results, our data-retained flip-flop (HVT type) clearly extends the available tradeoff, and it provides more choices on the cell optimization. We explore gate-length (L_{gate}) biasing cases for each NVT, HVT and LVT cell (+2 and -2*nm*). The results show that data-retained flip-flops offer more leakage-delay choices even when L_{gate} biasing is available as well. (The LVT data-retained flip-flop will never be used since it has no leakage-delay benefit over the NVT type of normal flip-flop.)

For correct operation during clock-enable periods, the wake-up latency when coming out of power gating should be less than the delay of the gated clock signal. In Figure 8(a), the sum of *EN*-to-*Q* delay in the *CG* (clock gating) cell and CTS buffer delays is typically larger than 200*ps*. Figure 8(b) shows the waveform of the clock enable signal and virtual ground voltage from SPICE simulation. From the waveform, the voltage of virtual ground goes to zero within 30*ps*. This means that the wake-up time of DRPG is sufficiently fast for the correct flip-flop operation. On the wake-up, the measured in-rush current is 40.2*uA* (peak), which is 45% of peak current in normal power gating case. Power overhead from the inrush current is compensated as shown in Figure 3.

C. Comparison with Conventional Retention Flip-Flops

Conventional retention flip-flops retain data during power gating, and can also be used for the DRPG. Figure 9 shows a schematic of the live-slave type of retention flip-flop, which provides power into a slave latch during the power gating. If we replace the flip-flop in Figure 2 with the retention flip-flop, we do not need to use the retention switch. We can remove the clock-mask circuit (Figure 9(a)) since the clock is masked from clock-gating circuit. To preserve the proper voltage level at the output port, we should connect real (true) ground to the output inverter (Figure 9(b)).

We implement the live-slave type of retention flip-flop as shown in Figure 9, and used the flip-flop for DRPG. Figure 10 shows (a) virtual ground voltage and (b) current results for live-slave retention flip-flop (blue color) and retention switch (red color). From the results, the conventional retention flip-flop can achieve 25% active-mode leakage reduction without delay overhead, compared with normal flip-flops.

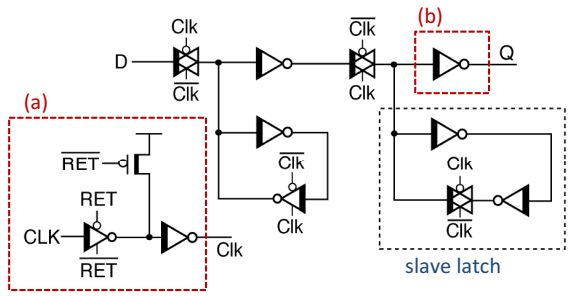

Fig. 9. Live-slave retention flip-flop. To use the flip-flop for DRPG, (a) clock-mask circuit can be removed, and (b) output inverter should be connected to the real ground.

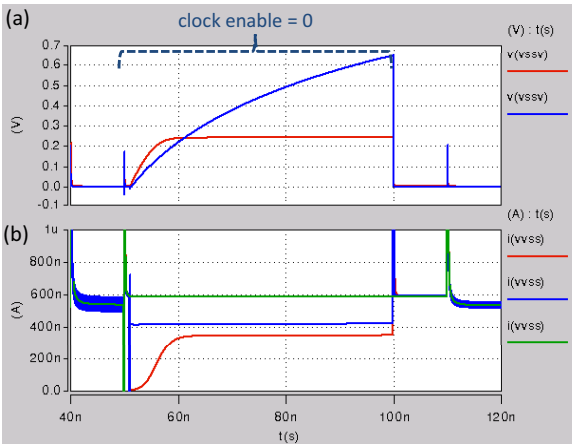

Fig. 10. (a) Virtual ground voltage (vssv) and (b) leakage current for normal flip-flop (green color), live-slave retention flip-flop (blue color) and power gating with retention switch (red color).

However, as discussed above in Section III-A, the voltage of virtual ground goes to near high voltage (VDD) during the power gating, and there is significant inrush current with turning on of the sleep (footer) switch. Because of the inrush current, the conventional retention flip-flop is not suitable for active-mode power gating.

D. Leakage Reduction for Implemented Designs

We implement 11 benchmark designs to assess the active-mode leakage reduction from our power gating approach. We use multi-V_{th} (HVT, NVT and LVT) standard library cells including data-retained flip-flops (N.B.: recall from Section IV-B above that the LVT data-retained flip-flop is never instantiated). Three different timing constraints are used – (a) tight constraint: maximum available frequency, (b) normal constraint: 20% longer clock period than tight constraint, and (c) loose constraint: 50% longer clock period than tight constraint. Figure 11 shows area breakdowns of combinational logic, non-clock-gated flip-flops, and clock-gated flip-flops for the implemented designs with the normal timing constraint. From the results, the portion of clock-gated flip-flops varies according to the designs. Some designs (e.g, AES_CIPHER and WB_CONMAX) do not permit significant clock gating. However, we can see that most of the designs can use clock-gating logic extensively.

We have applied our power gating technique to the implemented designs. Table II shows the implemented results and leakage power reduction over the conventional designs, which do not power-gate during active mode. The amount of leakage reduction depends on (1) the portion of clock-gated flip-flops as shown in Figure 11 and (2) the timing constraints. Designs with the small portion of clock-gated flip-flops (e.g., AES_CIPHER and WB_CONMAX) show small (or no) leakage reduction from our DRPG technique. As shown in Table I,

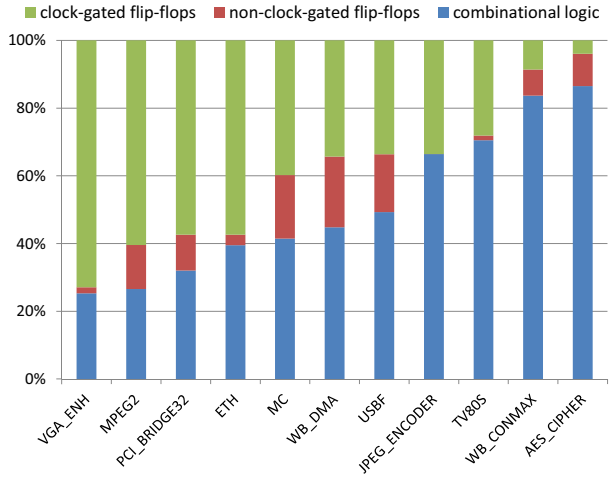

Fig. 11. Breakdown of area for implemented designs (clock-gated flip-flops, non-clock-gated flip-flops and combinational logic).

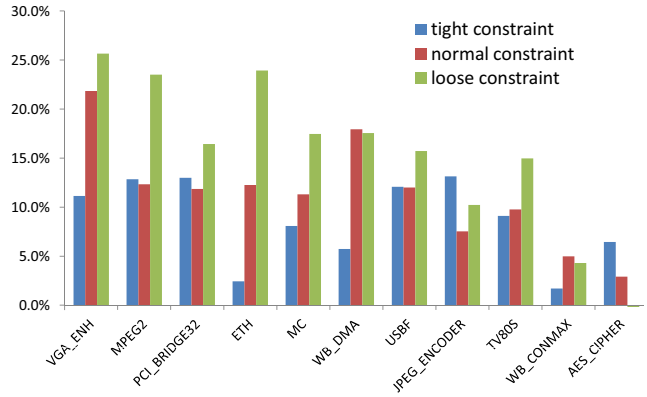

Fig. 12. Leakage reduction for different timing constraints – tight constraint (maximum available frequency), normal constraint (tight constraint + 20% clock period) and loose constraint (tight constraint + 50% clock period).

the proposed data-retained flip-flop has delay overhead. Therefore, we cannot replace normal flip-flops with the data-retained flip-flops if the timing slack is less than the delay overhead; we only exploit available slack, and do not permit performance (timing) degradation, i.e., DRPG is not applied to flip-flops in timing-critical paths. The number of flip-flops in Table II shows that more data-retained flip-flops are used with looser timing constraints. With a tight timing constraint, more flip-flops are in timing-critical paths, and hence fewer flip-flops can be replaced with the data-retained flip-flops. Moreover, with a tight constraint, the leakage contribution of combinational cells increases more than that of flip-flops, since buffer insertion and gate sizing are mainly performed on the combinational cells. As a result, timing constraint effects on achievable leakage reduction vary across testcases, as shown in Figure 12. We have estimated area overheads of the DRPG implementation based on Figure 6. We consider additional areas for DRPG flip-flops and sleep switches in this estimation. As shown in Table II, our DRPG technique shows 3.09% area overhead on average.

From the results, we see that our DRPG technique can reduce leakage power over conventional designs by up to 13.1% (average 8.7%), 21.8% (average 11.3%) and 25.7% (average 15.3%) with tight, normal and loose timing constraints, respectively. The leakage reductions are for digital portions only, and we expect that larger design cases will show similar leakage reductions as in our current experimental results.

TABLE II
LEAKAGE REDUCTION ACHIEVED BY DATA-RETAINED FLIP-FLOPS ON BENCHMARK DESIGNS [CG-FF: CLOCK-GATED FLIP-FLOPS].

design	timing constraint	clock period (*ns*)	# of instances	design area (um^2)	# of CG-FF		leakage power (w)		leakage reduction		area overhead
					normal	DRPG	flip-flops	total	flip-flops	total	
AES_CIPHER	tight	1.15	19,149	51,367	2	154	3.50E-05	4.55E-04	2.9%	6.5%	0.24%
	normal	1.38	14,799	38,254	0	156	3.32E-05	2.94E-04	6.4%	2.9%	0.33%
	loose	1.73	11,911	31,440	0	156	3.10E-05	1.94E-04	5.5%	-1.0%	0.40%
ETH	tight	1.15	45,160	218,335	140	9936	2.97E-04	1.51E-03	44.0%	2.4%	4.45%
	normal	1.38	31,085	168,941	64	10012	2.71E-04	7.05E-04	45.8%	12.3%	5.73%
	loose	1.73	24,894	143,574	21	10055	2.61E-04	4.05E-04	47.5%	23.9%	6.71%
JPEG_ENCODER	tight	1.25	45,020	158,303	133	4192	2.08E-04	1.38E-03	54.9%	13.1%	1.89%
	normal	1.50	36,321	122,791	69	4257	1.63E-04	8.59E-04	38.7%	7.5%	2.46%
	loose	1.88	32,140	110,491	17	4309	1.26E-04	5.81E-04	43.4%	10.2%	2.76%
MC	tight	1.25	4,733	20,293	18	705	4.39E-05	1.10E-04	24.2%	8.1%	2.95%
	normal	1.50	4,027	17,812	7	716	3.74E-05	7.32E-05	23.2%	11.3%	3.39%
	loose	1.88	3,822	16,986	3	720	3.02E-05	5.47E-05	49.4%	17.5%	3.57%
MPEG2	tight	0.95	9,335	52,373	165	2351	1.12E-04	2.43E-04	31.6%	12.9%	3.66%
	normal	1.14	8,303	49,936	90	2425	9.86E-05	1.87E-04	38.0%	12.3%	3.92%
	loose	1.43	7,627	46,343	48	2468	8.12E-05	1.23E-04	43.5%	23.5%	4.27%
PCI_BRIDGE32	tight	0.95	9,556	52,975	292	2480	1.32E-04	2.85E-04	33.0%	13.0%	3.79%
	normal	1.14	8,626	49,443	166	2606	1.17E-04	2.19E-04	30.1%	11.9%	4.22%
	loose	1.43	7,657	45,083	67	2705	9.29E-05	1.45E-04	35.4%	16.4%	4.75%
TV80S	tight	1.30	4,649	14,180	35	307	2.60E-05	1.16E-04	41.6%	9.1%	2.17%
	normal	1.56	4,062	11,527	28	314	2.32E-05	7.27E-05	35.4%	9.8%	2.71%
	loose	1.95	3,393	10,176	15	327	1.55E-05	4.78E-05	52.4%	15.0%	3.13%
USBF	tight	0.95	9,031	35,639	137	1011	8.49E-05	1.89E-04	25.8%	12.1%	2.44%
	normal	1.14	8,181	31,945	70	1078	6.21E-05	1.29E-04	27.3%	12.0%	2.86%
	loose	1.43	7,470	29,295	24	1132	5.06E-05	9.23E-05	29.4%	15.7%	3.23%
VGA_ENH	tight	1.30	52,239	272,409	0	16643	4.14E-04	1.27E-03	48.1%	11.1%	4.82%
	normal	1.56	37,491	214,645	1	16642	4.09E-04	6.96E-04	48.7%	21.8%	6.05%
	loose	1.95	35,557	208,423	4	16639	4.08E-04	5.71E-04	49.1%	25.7%	6.22%
WB_CONMAX	tight	1.20	19,273	56,695	146	238	5.21E-05	2.94E-04	-5.9%	1.7%	0.41%
	normal	1.44	18,478	51,733	83	301	2.80E-05	1.76E-04	16.9%	5.0%	0.53%
	loose	1.80	18,385	49,210	1	383	2.54E-05	1.22E-04	15.7%	4.3%	0.69%
WB_DMA	tight	0.75	2,456	10,099	81	236	3.72E-05	7.04E-05	16.0%	5.7%	2.01%
	normal	0.90	2,161	9,351	50	267	2.39E-05	4.23E-05	32.5%	17.9%	2.39%
	loose	1.13	1,955	8,622	9	308	1.63E-05	2.79E-05	39.5%	17.6%	2.91%

V. CONCLUSION

In this work, we propose a new circuit-level technique which enables power gating of clock-gated flip-flops during active mode. We combine clock gating and power gating techniques together, such that the flip-flops are power-gated during clock masked periods. We introduce a retention switch which retains data during the power gating. With the retention switch, correct logic states and functionalities are guaranteed without additional overheads. With small area and performance overheads, our proposed technique can achieve significant dynamic leakage reduction over conventional designs. Using 65*nm* libraries and 11 open-source designs, we demonstrate that the proposed power gating technique can achieve maximum and average leakage savings of 25.7% and 11.8% over conventional designs.

REFERENCES

[1] *International Technology Roadmap for Semiconductors (Design Chapter)*, 2011, http://www.itrs.net .

[2] L. Benini and G. De Micheli, "Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines", *IEEE TCAD* 15(6) (1996), pp. 630–643.

[3] Y. Shin, J. Seomun, K.-M. Choi and T. Sakurai, "Power Gating: Circuits, Design Methodologies, and Best Practice for Standard-Cell VLSI Designs", *ACM TODAES* 15(4) (2010), pp. 1–37.

[4] K. Usami and N. Ohkubo, "Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals", *Proc. ICCD*, 2006, pp. 155–161.

[5] L. Bolzani, A. Calimera, A. Macii, E. Macii and M. Poncino, "Enabling Concurrent Clock and Power Gating in an Industrial Design Flow", *Proc. DATE*, 2009, pp. 334–339.

[6] E. Macii, L. Bolzani, A. Calimera, A. Macii and M. Poncino, "Integrating Clock Gating and Power Gating for Combined Dynamic and Leakage Power Optimization in Digital CMOS Circuits", *Proc. Euromicro DSD*, 2008, pp. 298–303.

[7] L. Li, K. Choi and H. Nan, "Effective Algorithm for Integrating Clock Gating and Power Gating to Reduce Dynamic and Active Leakage Power Simultaneously", *Proc. ISQED*, 2011, pp.74–79.

[8] J. Seomun, I. Shin and Y. Shin, "Synthesis of Active-Mode Power-Gating Circuits", *IEEE TCAD* 31(3) (2012), pp. 391–403.

[9] S. Kim, S. V. Kosonocky, D. R. Knebel, K. Stawiasz and M. C. Papaefthymiou, "A Multi-Mode Power Gating Structure for Low-Voltage Deep-Submicron CMOS ICs", *IEEE TCAS-II* 54(7) (2007), pp. 586–590.

[10] T. Kitahara, F. Minami, T. Ueda, K. Usami, S. Nishio, M. Murakata and T. Mitsuhashi, "A Clock-Gating Method for Low-Power LSI Design", *Proc. ASP-DAC*, 1998, pp. 307–312.

[11] K. Fukuoka, M. Iijima, K. Hamada, M. Numa and A. Tada, "Leakage Power Reduction for Clock Gating Scheme on PD-SOI", *Proc. ISCAS*, 2004, pp. 613–616.

[12] B. H. Calhoun and A. P. Chandrakasan, "Standby Power Reduction Using Dynamic Voltage Scaling and Canary Flip-Flop Structures", *IEEE JSSC* 39(9) (2004), pp. 1504–1511.

[13] *Cadence LC User's Manual*, http://www.cadence.com .

[14] *Cadence Encounter User's Manual*, http://www.cadence.com .

[15] *Calypto PowerPro CG User's Manual*, http://www.calypto.com .

[16] *OpenCores: Open Source IP-Cores*, http://www.opencores.org .

[17] *Synopsys Design Compiler User's Manual*, http://www.synopsys.com .

[18] *Synopsys HSPICE User's Manual*, http://www.synopsys.com .

[19] *UCSD Sensitivity-Based Leakage Optimizer*, http://vlsicad.ucsd.edu/SIZING .