Power Gating Design for Standard-Cell-Like Structured ASICs

Sin-Yu Chen, Rung-Bin Lin, Hui-Hsiang Tung, and Kuen-Wey Lin
Computer Science and Engineering line
Yuan Ze University
Chung-Li, Taiwan

***Abstract*—Structured ASIC has been introduced to bridge the power, performance, area and design cost gaps between ASIC and FPGA. As technology scales, leakage power consumption becomes a serious problem. Among the leakage power reduction techniques, power gating is commonly used to disconnect idle logic blocks from power network to curtail sub-threshold leakage. In this paper, we apply power gating to structured ASICs for leakage power reduction. We present a power-gated via-configurable logic block (PGVCLB) and a power gated design flow mostly using existing standard cell design tools. We can configure PGVCLBs in a design to implement fine-grained power gating, coarse-grained/cluster-based power gating or even *distributed sleep transistor network* (*DSTN*). With fine-grained power gating, we can achieve 52% leakage reduction on average with only 8% area and 17% delay overheads when compared to the data obtained using a non-power-gated library.**

Keywords: power-gating, low power, via-configurable, structured ASIC.

I. INTRODUCTION

Structured ASIC is a design style that can bridge the performance, power, area, and design cost gaps between ASIC and FPGA [1-7]. It contains arrays of via-configurable logic blocks (VCLB) with prefabricated transistors and possibly predefined yet via-configurable metal wires [8-17]. Its engineering change order (ECO) cost is considerably smaller than that for ASIC. Its ECO cost mainly involves customizing a few relatively inexpensive via masks, each of which requires far fewer shots than a metal-wire mask does. Its regular layout structure also enables a higher and more predictable manufacturing yield.

Many researches have been done to make structured ASIC more viable. These researches mainly focus on exploring VCLB architectures. Some of these researches also perform routing architecture exploration using their own structured ASIC routers [9,16,17]. However, to the best of our knowledge, few researches have been done for low-power structured ASICs although many power reduction techniques for standard cell designs are around. These techniques include multi-supply voltages, gate sizing, clocking gating, etc. for taming dynamic power and multi-threshold voltages, power gating, body-bias, gate length engineering, etc. for controlling leakage power. Taylor and Schmit [18] propose a VCLB architecture that employs dual supply voltages and gate sizing to reduce structured ASIC dynamic power. The VCLB is comprised of three stages. The first stage has three input buffers. Each buffer can be configured by vias into a level convertor. The second stage is a 3-input lookup table that can be configured to realize any three-variable logic function. The third stage is comprised of an output driver which can be configured into a driver of different drive classes. Although, this VCLB by itself can curtail dynamic power, it has not been used to create a cell library that can be employed by a logic synthesis tool to synthesize a circuit. Nevertheless, the above work has pioneered low-power structured ASIC. As transistor feature size continues to shrink, controlling leakage power is as important as curtailing dynamic power for many low-power applications. Hence, it is necessary to develop methodologies for controlling structured ASIC leakage power. Among leakage power reduction techniques [19-28], power gating [21-28] is one of the most effective methods. The main idea behind power-gating is to serially connect a high-Vth sleep transistor to a logic network and turn off the sleep transistor to reduce sub-threshold leakage current while the logic network is not active. Although this operation idea is simple, achieving optimal power-gating is very challenging [23]. The main problem is that it is hard to accurately assess the delay implication due to voltage drop across a sleep transistor. Hence, sleep transistors should be used under discretion. The voltage drop across a sleep transistor is a function of sleep transistor size and the current flowing through it. Given an amount of delay budget, sleep transistors are sized based on the current flow through it. Normally, we apply power gating only to the logic gates on non-critical paths. The objective is to maximize leakage reduction while minimizing sleep transistor size under delay budget. Sizing sleep transistors in a standard cell design is a difficult task [22]. It is even more challenging for structured ASICs due to transistors being prefabricated before logic and interconnect customization. In this paper, we apply power gating to structured ASICs for leakage power reduction. We present a power-gated via-configurable logic block (PGVCLB) and a power gated design flow mostly using existing standard cell design tools. The main features of PGVCLB and our contributions are as follows.

- PGVCLB has a pair of equal-sized high-Vth pMOS devices which can be configured by vias into sleep transistors of different drive classes.

This work is supported by National Science Council, Taiwan, under the grants NSC 96-2221-E-155-064-MY3

978-3-9810801-6-2/DATE10 © 2010 EDAA

- We can configure PGVCLBs in a design to implement fine-grained power gating, coarse-grained/cluster-based power gating or even *distributed sleep transistor network* (*DSTN*).
- We exploit an existing logic synthesis tool such as Synopsys Design Compiler to implement fine-grained power gating simply assigning a virtaul area to each cell in the cell lbrary based on PGVCLB.
- With fine-grained power gating, we can achieve 52% leakage reduction on average with only 8% area and 17% delay overheads when compared to the data obtained using a non-power-gated cell library.

The rest of this article is organized as follows. Section II gives related background about power gating and via-configurable structured ASICs. Section III presents our PGVCLB. Section IV describes our power-gating design methodology. Section V shows some experimental results. Final section draws a conclusion and presents our future work.

II. PRELIMNARY

A. Power Gating

Power-gating technique uses a high-Vth sleep transistor in series with the pull-up and/or the pull-down of a low-Vth logic block to reduce leakage power [19,20]. The sleep transistor can be turned off when the low-Vth logic block is inactive, thus resulting in a significant reduction of sub-threshold leakage current. Figure 1 shows a sleep transistor used for power gating. A sleep transistor can be a high-Vth nMOS or pMOS transistor. A pMOS sleep transistor served as a header switch connects the power network to virtual VDD. An nMOS sleep transistor served as a footer switch connects the ground network to virtual GND. Normally, either a header switch or a footer switch is used to conserve area and reduce timing penalty caused by voltage drop across sleep transistors. In this work, we will consider only header switch. The logic network in Figure 1 can be as simple as a logic gate. If it is a logic gate in a standard cell library, we call it a power-gated cell. If the power-gated cells in a standard cell design are used in an isolated manner, i.e., their virtual VDDs are not connected, we call this power gating approach *fine-grained power gating*. If a sleep transistor is shared by a cluster of power-gated logic cells, possibly located in close proximity, such kind of power gating is called *coarse-grained* or *cluster-based power gating*. If the virtual VDDs of power gated cells (clusters) in a design are connected together, the sleep transistors of these power-gated cells (clusters) are in fact form a transistor network. We call such kind of power gating *distributed sleep transistor network* (*DSTN*). The main idea of DSTN is to put together the logic gates (clusters) whose peak currents do not occur at the same time, or preferably which do not switch simultaneously, so that the sleep transistors can be trimmed for saving area [22]. Doing DSTN optimally in a standard cell design environment is quite difficult. To the best of our understanding, commercial tool like Cadence Encounter Power System (EPS) adopts cluster-based power gating. EPS

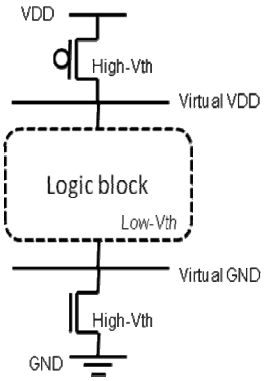

Figure 1. Power gating.

requires creating a power-gate library for power-gate cells. The so-called power-gate cell, different from our power-gated cell which is a logic cell with a sleep transistor, is simply a sleep transistor. A power-gate cell conceptually and physically is an inseparable entity so that making EPS work with DSTN might be quite challenging. We hence will not consider using EPS for our power gating design. However, to test our idea, we will resort to fine-grained power gating. Nevertheless, our PGVCLB will be designed also to facilitate cluster-based power gating and DSTN.

Fine-grained power gating can be done during logic synthesis. A logic synthesis tool should be designed to include more power-gated cells on non-critical paths to trade slack for leakage reduction. It normally needs to cooperate with a timing analyzer to watch timing violations. To avoid developing a logic synthesis tool and timing analyzer, we have proposed a novel way to do fine-grained power gating using an existing logic synthesis tool. This approach will be detailed later.

B. Via-Configurable Logic Blocks

There are various types of VCLBs. First, a VCLB can be an FPGA-like look-up table (LUT). LUT-based VCLBs proposed in [11] can be laid out in a way similar to standard cells. The major problem of LUT-based VCLBs is that each LUT should be accompanied by a flip-flop. If flip-flops are not used as frequently as LUTs, the utilization of transistors in VCLBs will be low. The second type of VCLB is based on PLA structure [12]. PLA enables implementation of two-level logic functions, but it still needs flip-flops for implementing a sequential design. The third type of VCLBs [8,9,15] uses series-parallel layout structures to realize either combinational logic gates or flip-flops. Since our work is closely related to this type of VCLB, especially the VCLB presented in [15], we will give a more detailed look into it, which is designed to support standard cell designs. Standard cell design style enables simple power/ground (P/G) distribution. P/G buses between adjacent cells can be abutted and P/G straps using higher metal layers can be easily deployed to form a P/G network. The main layout features of the VCLB in [15] are as follows.

- M2 P/G lines are located at the top and bottom boundaries so that VCLBs can be abutted seamlessly to form a cell row.

A cell row can be flipped with respect to X-axis so that row abutment can be done to form wider P/G lines. This layout feature also enables free deployment of M3 (or higher layers) P/G straps above the core area.

- The VCLB has some M1 jumpers at both sides for inter VCLB connections. These jumpers enable the composition of multiple VCLBs to realize a more complex logic function.

A standard cell enabling layout design is essential to the establishment of a structured ASIC design flow mostly using existing standard cell design tools. Since it is not required to develop many new tools and purchase extra tools to establish such a flow, we believe that a standard-cell-like structured ASIC design flow will be instrumental to the adoption of structured ASIC in industry.

III. POWER-GATED VCLB (PGVCLB)

To apply power-gating to structured ASICs, we extend the VCLB proposed in [15] to include some via-configurable sleep transistors (VCST). We call this VCLB power-gated VCLB (PGVCLB). Figure 2 shows a stick diagram of PGVCLB. The lower part (covering the region between GND and virtual VDD (VVDD) of PGVCLB is similar to the VCLB proposed in [15] and the upper part of PGVCLB is a VCST block. The VCST block of PGVCLB has two equal-sized high-Vth pMOS transistors, each serving as a header switch. Just like VCLB, PGVCLB has a standard cell enabling layout design.

The two pMOS transistors in a VCST block can be connected in parallel to form a single sleep transistor of larger driving capability or each of them can be used independently as a sleep transistor. Installing some vias between M1 and M2 wires in the dotted regions of a VCST block, we can configure a PGVCLB into a logic cell without power gating (called PGVCLBS0), with power gating using only one high-Vth transistor (called PGVCLBS1), or with power gating using both two high-Vth transistors (called PGVCLBS2). Clearly, PGVCLBS0 has the largest leakage but smallest delay whereas PGVCLBS1 has the smallest leakage but largest delay. PGVCLBS0, PGVCLBS1, and PGVCLBS2 have the same physical area if they implement the same logic function of a certain driving class. For example, PGVCLBs in Figure 3 are configured to realize a PGVCLBS1, PGVCLBS2, and PGVCLBS0 of NAND2X1, respectively.

As just mentioned, the five dotted regions in VCST are the places where vias can be installed to deploy a sleep transistor in a PGVCLB. Installing vias in dotted regions 1 and 4 but none of vias in regions 2, 3, and 5, the sleep transistor on the left of Figure 2 is deployed (PGVCLBS1). Vias installed in region 1 connect the drain of the associated sleep transistor to virtual VDD whereas vias installed in region 4 connect the sleep control signal (SL) to the associated sleep transistor. Not installing vias in region 2 prevents VDD from directly connecting to virtual VDD. Similarly, installing vias in regions 3 and 5 but none of vias in regions 2, 1, and 4, the sleep transistor on the right is deployed (PGVCLBS1). Installing vias in regions 1, 4, 3, and 5 but none of vias in region 2, both sleep transistors are deployed (PGVCLBS2). On the other hand, installing vias only in region 2 but none of vias in regions 1, 4, 3, and 5, both sleep transistors are not used (PGVCLBS0).

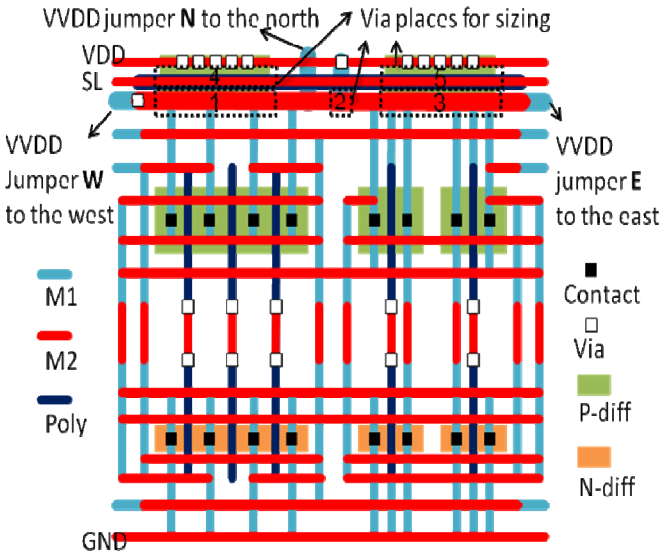

Figure 2. Stick diagram of PGVCLB.

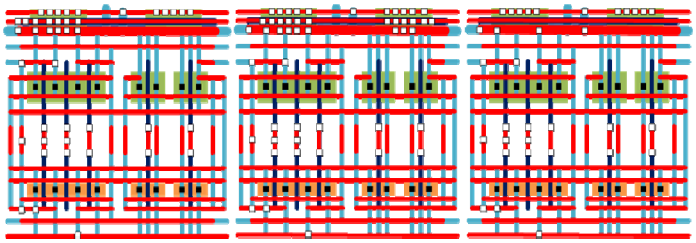

Figure 3. PGVCLBS1 (left) , PGVCLBS2, PGVCLBS0 (right).

The VCST block of a PGVCLB has two via-configurable jumpers E and W for connecting its virtual VDD to neighboring PGVCLBs. It also has a jumper N for connecting its virtual VDD to the north. Such an arrangement serves two purposes. First, these jumpers can be employed to implement a cluster-based power gating and/or a DSTN by joining the VCST blocks in different PGVCLBs, each of which implements a logic function. Second, they can be used to connect the virtual VDD rails of multiple PGVCLBs for realizing a complicate logic function. Figure 4 shows such an example where AOI211X4 gate employs two PGVCLBs. Virtual VDD rails of the two PGVCLBs are connected together through the VVDD jumpers in order to connect their sleep transistors together. Note that to avoid the explosive combinations of sleep transistors, the sleep transistors in each PGVCLB is used in concert with each other. That is to say, for PGVCLBS0, none of sleep transistors in each PGVCLB is used; for PGVCLBS1 only one sleep transistor in each PGVCLB is used; for PGVCLBS2 both sleep transistors in each PGVCLB are used.

A flip-flop which employs three PGVCLBs is also realized using multiple PGVCLBs, but none of sleep transistors are deployed, i.e., not power-gated. The reason is that we do not

want to use a retention circuit, incurring an area overhead, to keep the state of a flip-flop.

In principle, we could have made the sleep transistors in a VCST block more granular, i.e., more but smaller sleep transistors in a VCST block. Doing-so could result in explosive combinations of sleep transistors, which increase the burden of creating a cell library considerably. For example, just taking what we have now in a VCST block, we have three choices of using sleep transistors. We also have four choices of connecting virtual VDD rail to the neighboring PGVCLBs, i.e., to the north, to the east, to the east and north, or not at all. We hence end up with 12 layouts per logic cell. This is the reason why we did not make the sleep transistors more granular. Note that by the same reason there are four layouts per sequential cell.

In addition to VCLB design, via-configurable routing fabric is another important issue for structured ASICs. In this work we will not employ any predefined routing fabrics. However, our PGVCLB design and our power gating design method presented in Section IV can also be applied to the case with a predefined routing fabric.

IV. Power-Gating Design Flow

Figure 5 shows a power-gating design flow mostly using existing tools except logic packing and alignment. Logic packing puts together two simple logic functions, each taking a PGVCLB, into a single PGVCLB for saving more area. Because of limitation on the use of sleep transistors, we can pack two PGVCLBS0's, one PGVCLBS0 with one PGVCLBS1, or two PGVCLBS1's. Alignment arranges the placement output into an array of PGVCLBs.

We use an existing logic synthesis tool such as Synopsys Design Compiler to implement fine-grained power gating. We exploit a basic rule used by many logic synthesis tools, which is that a logic synthesis tool would normally try to minimize area while satisfying timing constraints of a circuit. This rule stipulates that timing is a constraint whereas area is an objective. Hence, a logic synthesis tool such as Design Compiler would use as many small size logic gates as possible on non-critical paths to minimize circuit area once timing targets are met. Exploiting this basic property, we deliberately give a power-gated logic gate a smaller area so that a logic synthesis tool will automatically assign as many power-gated logic gates (also slower logic gates) as possible to non-critical paths. We call such an area *virtual area*. Virtual area instead of physical area will be used for synthesizing a circuit. We use the following formula to determine the virtual area of a power-gated cell C with PGVCLBSn.

$$A_n(C) = A(C) \times d_0(C) / d_n(C), \qquad (1)$$

where $A(C)$ is the physical area of gate C (remembering that PGVCLBS0, PGVCLBS1, and PGVCLBS2 of the same logic gate have the same physical area). $d_0(C)$ is the delay of C with PGVCLBS0. $d_n(C)$ is the delay of C with PGVCLBSn for $n = 0, 1, 2$. Since delay of a gate depends on input slew, output load, and output switching phase, we set the delay of a gate equal to the average delay of rise and fall delays with a median slew and a median load found in its associated delay tables. The smaller the delay is, the larger the virtual area is. TABLE I shows the virtual areas of NAND2X1. A logic synthesis tool would use as many PGVCLBS1's as possible on non-critical paths to minimize virtual area and thus minimize leakage power. Note that we can also consider using leakage data or combination of delay and leakage data to determine the virtual area of PGVCLBSn. We employ (1) because we concern delay more than leakage.

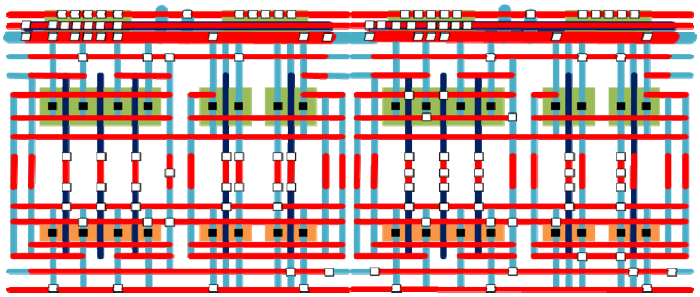

Figure 4. PGVCLBS1 of AOI211X4.

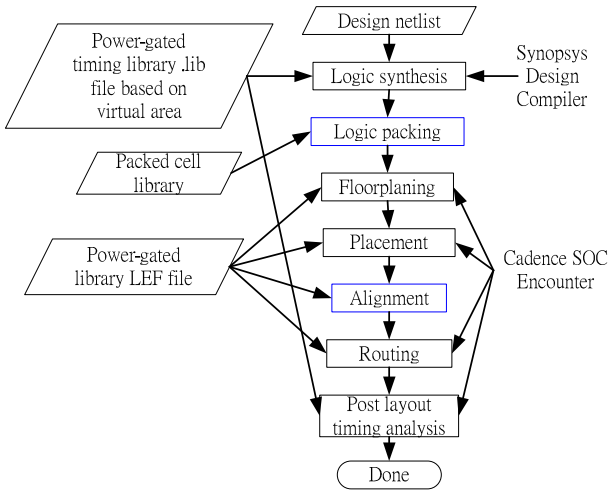

Figure 5. Power gating structured ASIC design flow.

TABLE I. Virtual Areas of NAND2X1

NAND2X1	Area (um^2)	Delay(ns)	Leakage (pW/s)	Virtual area (um^2)
PGVCLBS0	13.317	0.137	4347	13.317
PGVCLBS1	13.317	0.200	1161	9.167
PGVCLBS2	13.317	0.167	1401	10.968

V. Experimental Result

We have implemented a cell library based on our PGVCLB for evaluating our power gating design methodology. When implemented using UMC 90nm process technology, our PGVCLB has an area of 33.2928um^2. The library consists of 1112 unpacked cells and 844 packed cells. Remember that we have 12 layouts for a logic function of a particular drive class. Power and timing data of each cell are characterized using HSPICE. For the purpose of comparison, we synthesize some ISCAS'89 and ITC'99 benchmarks using three cell libraries

based on UMC 90nm process technology. These three libraries are our power-gated library (PGL), our non-power-gated library (NGPL) based on the VCLB proposed in [15], and a commercial yet non-structured standard cell library (STDL), respectively. Synopsys Design Compiler is used for synthesis. Cadence SOC Encounter is used for placement, routing, and post-layout timing and power analysis. Core utilization is 90%. The same parameter settings such as power ring width, core utilization, etc. are used to design chips regardless of which library is used. The designs synthesized using PGL and NPGL are routed with M3 up to M5 layer whereas those using STDL are routed with M1 up to M5 layer.

Our experiments are carried out in a manner similar to the one presented in [15]. First, we synthesize each circuit without any timing requirements but aim only to minimize the area. We then find the longest path delay for each circuit after the circuit is routed. We call such a delay *base_delay*. We then perform logic synthesis for each circuit with clock period specified as $k \times base\_delay$, $k \in K = \{1, 0.9, 0.8, 0.65, 0.5, 0.35, 0.2, 0.1\}$. Placement, routing, and post-layout timing analysis is performed as usual. Carrying out experiments in this way is to see how far we can push the delay envelope of the designs using each library.

TABLE II summarizes delay, area, and leakage data obtained using PGL and STDL. The data in column "Delay ratio" is calculated as follows.

$$Delay\ ratio = \min_{i \in K} d_i(PGL) / \min_{j \in K} d_j(STDL), \quad (2)$$

where $d_i(PGL)$ is the longest path delay obtained using a clock period of $i \times base\_delay$ to synthesize a design with PGL. $d_j(STDL)$ is defined similarly for STDL. Let *pgl* and *stdl* be the indices so that $d_{pgl}(PGL)$ and $d_{stdl}(STDL)$ are the smallest delays for $d_i(PGL)$ and $d_j(STDL)$, respectively. The data in column "Area ratio" is calculated as follows.

$$Area\ ratio = a_{pgl}(PGL) / a_{stdl}(STDL), \quad (3)$$

where $a_{pgl}(PGL)$ is the area obtained with $k = pgl$. $a_{stdl}(STDL)$ is defined similarly. Leakage ratio is calculated in the same way. If we push the delay envelope, the leakage power of our PGL designs is on average 0.20 times that of the STDL counterparts at the expense of 3.3 times delay and 5.3 times area. TABLE III is similar to Table II, but the data are calculated in terms of the data obtained using NPGL. The leakage power of our PGL designs is on average 0.48 times that of the NPGL counterparts at the expense of 1.17 times delay and 1.08 times area. As one can see, we achieve 52% leakage reduction on average with only 8% area and 17% delay overheads when compared to the data obtained using a non-power-gated library.

TABLE IV is a comparison between NPGL and STDL. The leakage power of designs using NPGL is on average 0.43 times that of the STDL counterparts at the expense of 2.89 times delay and 5.19 times area. The reason why the leakage of the designs with NPGL is smaller than that with STDL is just because the transistors used in the cells of NPGL are smaller. Note that the transistors in the cells of NPGL are of the same size as those of PGL.

TABLE V shows the sleep transistor (power gated cell) usage for each design synthesized with PGL. As expected, when delay is improved, the usage of power-gated cells is reduced on average. However, the high percentage of power-gated cells (slower cells) shows that our power gating method presented in Section IV is effective. TABLE VI shows the non-power-gated cell usage on the most critical path in a design netlist created by Design Compiler. As one can see, as delay target gets tighter, the percentage of non-power-gated cells (faster cells) on the most critical path is increasing. The high usage at the tightest delay constraint shows again that our method presented in Section IV indeed is effective. One may notice that the usage of non-power-gated cells on the most critical path is normally not 100%. The reason is that the fall delay of a power-gated cell is almost the same as that of a non-power-gated cell. Remember that we employ a header switch in a power-gated cell. Hence, if the most critical path delay is dominated by fall transitions, a power-gated cell on it may be replaced by a non-power-gated cell without increasing the most critical path delay.

VI. Conclusions and Future Work

We have designed a power-gated VCLB and hence a power-gated cell library for via-configurable structured ASICs. We also present a fine-grained power-gating design flow mostly using standard cell design tools. The area in a power-gated cell is assigned a value that enables an existing logic synthesis tool to perform fine-grained power gating effectively.

TABLE II. Delay, area and leakage ratio (PGL vs. STDL)

Circuit	Delay ratio	Area ratio	Leakage ratio	Delay * Leakage	Delay * Area
s35932	2.14	6.15	0.31	0.66	13.13
s38417	2.70	5.71	0.32	0.86	15.42
s38584	3.43	6.96	0.32	1.08	23.86
b14	3.09	3.42	0.09	0.26	10.57
b15	2.35	6.20	0.23	0.55	14.59
b17	3.55	2.56	0.18	0.64	9.09
b18	3.99	6.71	0.15	0.61	26.75
b19	3.22	6.46	0.16	0.51	20.84
b20	4.11	5.15	0.16	0.67	21.18
b21	3.78	4.99	0.16	0.59	18.85
b22	3.94	4.04	0.12	0.47	15.94
Average	**3.30**	**5.30**	**0.20**	**0.63**	**17.29**

TABLE III. Delay, Area and Leakage Ratios (PGL vs. NGPL).

Circuit	Delay ratio	Area ratio	Leakage ratio	Delay * Leakage	Delay * Area
s35932	1.14	1.07	0.75	0.86	1.22
s38417	1.19	1.08	0.69	0.82	1.28
s38584	1.33	1.24	0.58	0.76	1.64
b14	1.25	1.08	0.36	0.45	1.35
b15	0.80	1.14	0.65	0.52	0.92
b17	1.14	1.17	0.54	0.61	1.34
b18	1.20	1.15	0.31	0.37	1.38
b19	1.12	1.12	0.31	0.35	1.26
b20	1.19	0.89	0.36	0.43	1.06
b21	1.32	0.90	0.35	0.46	1.18
b22	1.20	1.00	0.35	0.43	1.20
Average	**1.17**	**1.08**	**0.48**	**0.55**	**1.26**

Experimental results show that we can achieve 52% leakage reduction on average with only 8% area and 17% delay overheads when compared to the data obtained using a non-power-gated library.

In the future, we will study the problem of post-placement fine-grained power gating for faster timing closure. We will also explore cluster-based/coarse-grained power gating or even DSTN for larger leakage saving.

TABLE IV. DELAY, AREA AND LEAKAGE RATIOS (NGPL VS. STDL).

Circuit	Delay ratio	Area ratio	Leakage ratio	Delay * Leakage	Delay * Area
s35932	1.88	5.74	0.41	0.77	10.77
s38417	2.27	5.28	0.46	1.06	12.02
s38584	2.58	5.62	0.55	1.41	14.51
b14	2.47	3.15	0.24	0.59	7.80
b15	3.64	5.42	0.43	1.57	19.73
b17	3.12	4.85	0.40	1.25	15.11
b18	3.33	5.81	0.50	1.66	19.36
b19	2.87	5.79	0.51	1.48	16.60
b20	3.46	5.79	0.45	1.55	20.04
b21	2.87	5.57	0.44	1.27	15.97
b22	3.27	4.05	0.34	1.10	13.26
Average	2.89	5.19	0.43	1.25	15.02

TABLE V. POWER-GATED CELL USAGE (%).

Circuit	k=1	0.9	0.8	0.65	0.5	0.35	0.2	0.1
s35932	71	71	71	72	71	67	63	48
s38417	72	72	71	68	57	48	46	47
s38584	76	76	76	76	75	74	74	63
b14	95	91	81	82	85	80	71	51
b15	91	91	89	87	81	75	57	54
b17	91	91	90	89	85	78	63	60
b18	93	91	86	81	81	82	80	81
b19	93	92	90	83	82	83	82	82
b20	95	88	78	93	86	75	60	66
b21	95	89	78	93	85	73	59	66
b22	95	89	74	74	87	77	62	64
Average	88	86	80	82	80	74	65	62

TABLE VI. NON-POWER-GATED CELL USAGE ON CRITICAL PATHS (%).

Circuit	k=1	0.9	0.8	0.65	0.5	0.35	0.2	0.1
s35932	50	75	100	43	89	80	83	100
s38417	14	30	41	67	90	91	82	96
s38584	13	13	67	96	84	95	97	100
b14	39	83	82	77	71	70	85	88
b15	10	10	13	10	5	27	76	87
b17	9	6	6	6	6	58	65	87
b18	21	78	94	97	96	92	92	80
b19	2	15	67	97	84	87	79	83
b20	74	75	88	66	76	82	89	89
b21	61	90	83	23	79	80	86	83
b22	35	69	93	91	68	77	83	83
Average	30	49	67	61	68	76	83	89

REFERENCES

[1] B. Zahiri, "Structured ASICs: opportunities and challenges," ICCD, pp. 404-409, 2003.

[2] HardCopy ASIC Design Flow [Online]. Available: http://www.altera.com/products/software/flows/asic/qts-structured_asic.html

[3] eASIC: The configurable logic company [Online]. Available: http://www.easic.com

[4] K. C. Wu and Y. W. Tsai, "Structured ASIC, evolution or revolution?" ISPD, pp.103-106, 2004.

[5] A. Levinthal and R. Herveille, "FlexASIC structured array: a solution to the DSM challenge," DesignCon 2005.

[6] T. Okamoto, T. Kimoto, N. Maeda, "Design methodology and tools for NEC electronics' structured ASIC ISSP," ISPD, 2004, pp.90-96.

[7] White paper, "RapidChip technology: fast custom silicon through platform-based design," LSI Logic, 2004.

[8] Y. Ran and M. Marek-Sadowska, "Designing via-configurable logic blocks for regular fabric", IEEE Trans. on VLSI Systems, Vol. 14, No. 1, Jan. 2006.

[9] Y. Ran and M. Marek-Sadowska, "Via-configurable routing architectures and fast design mappability estimation for regular fabrics," IEEE Trans. on VLSI Systems , pp. 998-1009, Vol. 14, Sept. 2006.

[10] L. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan,V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, K.Y. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," DAC, pp. 782-78, 2004.

[11] C. Patel, A. Cozzie, H. Schmit, and L. Pileggi, "An architectural exploration of via patterned gate arrays," ISPD, pp. 184-189, 2003.

[12] F. Mo and R. Brayton ,"PLA-based regular structures and their synthesis," IEEE Trans. on CAD, Vol. 22, no. 6, pp. 723-729, June 2003.

[13] A. Nakamura, M. Kawarasaki, K. Ishibashi, M. Yoshikawa, T. Fujino, "Regular fabric of via programmable logic device using exclusive-or array (VPEX) for EB direct writing," IEIEC Trans. Electron., Vol. E91-C, No. 4, pp. 509-516, April 2008.

[14] S. Gopalani, R. Garg, S. P. Khatri, M. Cheng, "A lithography-friendly structured ASIC design approach," GLSVLSI, 2008, pp. 315-320.

[15] M. C. Li, H. H. Tung, C. C. Lai, and R. B. Lin, " Standard cell like via-configurable logic block for structured ASICs, " ISVLSI, pp. 381–386, 2008.

[16] N. V. Shenoy, J. Kawa, and R. Camposano, "Design automation for mask programmable fabrics," DAC, 2004, pp. 192-197.

[17] V. Kheterpal, A. J. Strojwas, and L. Pileggi, "Routing architecture exploration for regular fabrics," DAC, pp. 204 – 207, 2004.

[18] R. R. Taylor and H. Schmit, "Creating a power-aware strcutured ASIC," ISLPED, pp. 74-77, 2004.

[19] S. Mutoh, T. Douseki, and Y. Matsuya, "1-V power supply high-speed digital circuit technology with multi-threshold voltage CMOS," *IEEE JSSC*, pp. 847-854, 1995.

[20] M. Anis, S. Areibi, and M. Elmasry, "Design and optimization of multi-threshold CMOS (MTCMOS) circuits," *TCAD*, pp. 1324-1342, 2003.

[21] A. Sathanur, L. Benini, A. Macii., E. Macii, and M. Poncino, "Multiple power-gating domain (Multi-VGND) architecture for improved leakage power reduction," in ISLPED, pp. 11–13., 2008.

[22] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," *DAC*, pp. 495-500, 1998.

[23] K. Shi and D. Howard, "Sleep transistor design and implementation – simple concepts yet challenges to be optimum," in *VLSI DAT*, pp. 1-4, 2006

[24] V. Khandelwal and A. Srivastava, "Leakage control through fine-grained placement and sizing of sleep transistors," *ICCAD*, pp. 533 – 536, 2004.

[25] Y. Wang, Y. Liu, R. Luo, H. Yang, and H. Wang, "Two-phase fine-grain sleep transistor insertion technique in leakage critical circuits," *ISLPED*, pp. 238 – 243, 2006.

[26] C. Long and L. He, "Distributed sleep transistor network for power reduction," *IEEE TVLSI*, pp. 937 – 946, 2004.

[27] D. S. Chiou, S. H. Chen, S.C. Chang, and C. Yeh, "Timing driven power gating," *DAC*, pp. 121 – 124, 2006.

[28] L. Guo, Y. Cai, Q. Zhou, L. Kang, and X. Hong, "A novel performance driven power gating based on distributed sleep ransistor network," *GLSVLSI*, pp. 255-260, 2008.