Fine-Grained FSMD Power Gating Considering Power Overhead

Chi-Hoon Shin, Myeong-Hoon Oh, Jae-Woo Sim, Jae-Chan Jeong, and Seong Woon Kim

As a fine-grained power gating method for achieving greater power savings, our approach takes advantage of the finite state machine with a datapath (FSMD) characteristic which shows sequential idleness among subcircuits. In an FSMD-based power gating, while only an active subcircuit is expected to be turned on, more subcircuits should be activated due to the power overhead. To reduce the number of missed opportunities for power savings, we deactivated some of the turned-on subcircuits by slowing the FSMD down and predicting its behavior. Our microprocessor experiments showed that the power savings are close to the upper bound.

Keywords: Fine-grained power gating, FSMD.

I. Introduction

As CMOS technology has been gradually scaled down, power leakage has taken a significant portion of the power consumption. Among the many leakage control techniques, runtime leakage controls (RTLCs), such as body biasing and power gating, have been extensively researched. RTLC techniques can be classified into two classes according to the size of the power domains: coarse-grained and fine-grained approaches. Power domains are exclusive areas in a circuit, and each is controlled by a single cutoff leakage control signal. The size of a domain is called granularity. Among these techniques, many fine-grained approaches for achieving better utilization of circuit idleness have recently been researched [1]-[4].

Bhunia [1] used a common variable in two cofactors as a power gating signal, but the chance to apply RTLC is limited in a practical circuit. Usami [2] routed enable signals of a clock gating network to power the gating cells; however, if the cutoff signal changes too quickly to meet its break-even time (BET) [3], which is the minimum time at which the power savings equal the energy overhead, it can fail to save power. Yu [4] tuned a global clock into a turnoff signal, assuming that every gate has an idling period within a clock cycle; yet, it can bring about unnecessary on/off switches. Xu [3] tried to find the optimal granularity using input prediction; however, this approach can cause misprediction for irregular patterns.

Thus far, all of the fine-grained RTLC approaches have tried to guess the sequential idleness inside the circuits. Without knowing the circuit designer's actual intention, however, any solution is likely to end up with a sequential idleness that is either too limited, too excessive, or too specific.

We focus on the potential of a finite state machine (FSM) as the brain of a circuit that contains the information of sequential idleness between datapath components. One difficult problem for this approach lay in how to partition a datapath according to the FSM operation. For simplicity, in this letter, we assumed a target circuit has an FSM with datapath (FSMD) structure, as in [5], [6]. An FSMD differs from a traditional FSM in that a substate machine is inherently partitioned with data storage and functional circuits. For example, an original FSMD, like the example shown in Fig. 1(a), can be split into two sub-FSMDs that have their own control and datapath as in the example in Fig. 1(b). Each sub-FSMD has its own FSM and datapath, and they are alternately turned off. There is communication overhead needed for dealing with data transitions between the sub-FSMDs.

Manuscript received Aug. 31, 2010; revised Oct. 28, 2010; accepted Nov. 12, 2010.

Chi-Hoon Shin (phone: +82 42 860 1670, email: cshin@etri.re.kr) and Jae-Woo Sim (email: plipper@etri.re.kr) are with the Software Research Laboratory, ETRI, Daejeon, Rep. of Korea, and also with the Department of Computer Software and Engineering, University of Science and Technology, Daejeon, Rep. of Korea.

Myeong-Hoon Oh (email: mhoonoh@etri.re.kr) and Seong Woon Kim (email: ksw@etri.re.kr) are with the Software Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Jae-Chan Jeong (email: channij@etri.re.kr) is with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea, and also with the Computer Software and Engineering, University of Science and Technology, Daejeon, Rep. of Korea.

doi:10.4218/etrij.11.0210.0328

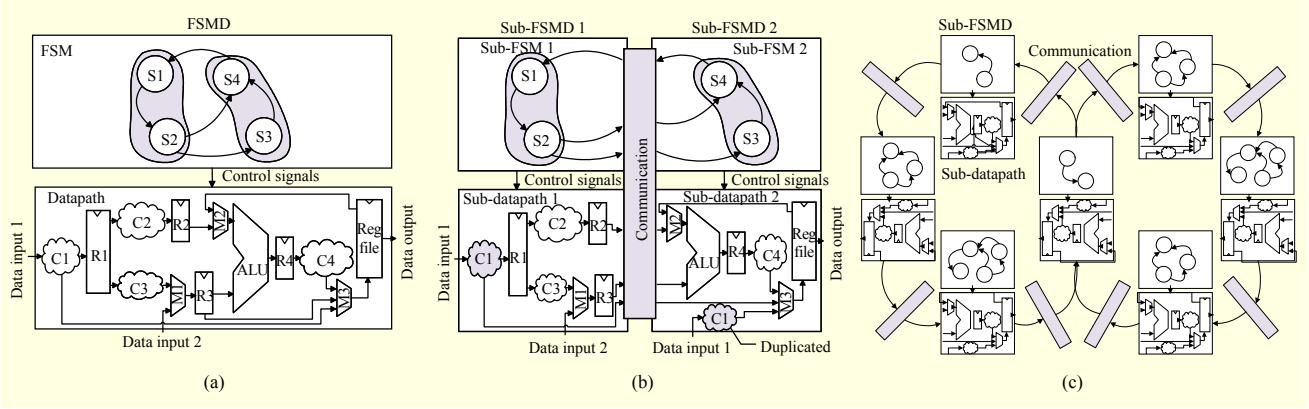

Fig. 1. (a) Unoptimized FSMD consisting of control FSM and datapath, (b) FSMD of (a) split into two sub-FSMDs, and (c) extended sub-FSMD topology.

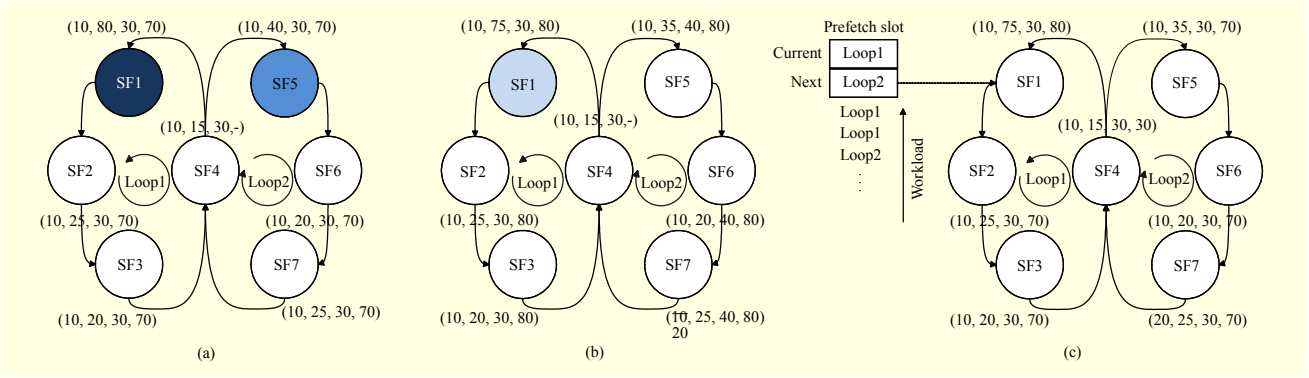

Fig. 2. Sub-FSMD in black represents a nondeterministic case, while the sub-FSMD in gray is half-deterministic, where (a) each sub-FSMD except SF1 and SF5 can be turned off after its execution, (b) return time is slowed down for meeting the BET, and (c) workload is pre-fetched for covering the half-deterministic case.

While the previous FSMD-based works [5], [6] have dealt with coarse-grained relations between few subcircuits, a modern digital circuit is spacious and complicated enough to be depicted as the example shown in Fig. 1(c), where each vertex denotes a sub-FSMD and an edge represents the communication for data transitions between sub-FSMDs. On the complicated topology, a fine-grained view of an FSMD circuit can show more promise for potential higher power savings, and the BET can significantly impact the power savings. In this letter, we investigate the impact of the BET on an FSM-based fine-grained RTLC and propose optimization methods for finding even more potential power savings.

II. Methods

In this section, we describe the methods for fulfilling the power saving potential through an analysis of an FSMD.

Our methods exploit sequential idleness in an FSM. Let us assume that the sub-FSMDs in an FSMD are executed in sequence for simplicity, such as in the example state transition graph shown in Fig. 2(a). Only one sub-FSMD at a time is expected to be activated in runtime; in turn, the other FSMDs can be turned off. Thus, ideally, only a control path and a datapath included in a sub-FSMD consume power, while the components in the other sub-FSMDs do not.

Considering the BET, however, some sub-FSMD modules might have to remain turned on because the BETs of those modules can be longer than the return time, that is, the time until a sub-FSMD is activated again after its previous execution. In this case, the modules cannot be turned off as described in the SF1 and SF5 cases in Fig. 2(a). We use the following denotations: α is the latency of the sub-FSMD, β is the BET of the module, γ is the minimum return time during which only the sub-FSMD within the inner loop is activated, and δ is the minimum return time when the turn in the FSM escapes to an outer loop. SF1's BET β is larger than both minimum return times γ and δ. Thus, SF1 has the least possibility of being turned off. In contrast, SF5's β is between γ and δ. After proceeding from SF5 to SF4, if SF1 in Loop1 will take a turn following SF4, SF5 can be turned off because the return time

(70) will be longer than the BET (40). In fact, the reason for this is that a precise expectation of the return time is not possible because an erratic workload affects the selection decision between the inner loop (the shortest loop for a sub-FSMD to be activated, from the perspective of SF1 in Fig. 2: the inner loop is Loop1) and an outer loop (any loop escaping from the inner loop: for SF1, Loop2 is an outer loop) at the junction node, as in the SF4 in Fig 2. To simplify this problem, we deal with three typical cases that are deterministic or can be made deterministic which cover most of the savings potential.

Let the workload order be $\{i, i+1, i+2, i+3, \ldots, i+k\}$, p: the sub-FSMD can be turned off.

Deterministic case.

$$(\gamma > \beta) \wedge (\delta > \beta) \Rightarrow \mathrm{p}$$

Half-deterministic case.

$$((\gamma \le \beta) \wedge (\delta > \beta)) \Rightarrow \begin{cases} \text{if } i+1 \in \text{inner loop, then, } \neg\mathrm{p} \\ \text{if } i+1 \in \text{outer loop, then, p} \end{cases}$$

or

$$((\delta \le \beta) \wedge (\gamma > \beta)) \Rightarrow \begin{cases} \text{if } i+1 \in \text{inner loop, then, p} \\ \text{if } i+1 \in \text{outer loop, then, } \neg\mathrm{p} \end{cases}$$

Nondeterministic case.

$$(\gamma \le \beta) \wedge (\delta \le \beta) \Rightarrow \begin{cases} \text{if } k = 1, \text{ then, } \neg\mathrm{p} \\ \text{if } \exists x(\Sigma_{j=2}^{x} j^{\text{th}}\mathrm{T}_{\text{loop}} > \beta) \text{ then, p.} \end{cases}$$

In a nondeterministic case, to turn a module off, we need to know k, but unless the entire workload is pre-evaluated in advance, k is only available at the moment the k-th loop starts. Until then, the module should remain turned on. In the half-deterministic and nondeterministic cases, a simple solution to achieve more potential is to slow the return time for meeting the BET, as shown in Fig. 2(b). For example, by slowing SF7 down from 10 to 20 in its latency (α), SF5 can always be turned off after its execution (changed from half-deterministic to deterministic), and that of SF1 can be selectively gated in the same manner as SF5 shown in Fig. 2(a) (changed from nondeterministic to half-deterministic). As an example, among the many ways for implementing the method, it can be realized by inserting a redundant state in the FSM of the sub-FSMD. This method is available only when slower performance is allowed, however. As presented in Fig. 2(c), when β is far larger than γ, a degradation in performance may be unacceptable. This problem can be solved using a light predictor that pre-fetches the next workload as depicted in Fig. 2(c). SF1 can be turned off just after its execution, predicting that the machine will jump to an outer loop (Loop2) by referring to the pre-fetched workload. This predictor is simpler and more reliable than the statistics-based predictor in [4].

III. Experiment

In this section, we estimate the power gating potential for a microprocessor model [7] using several application benchmarks. We applied two concepts, the slowing states and pre-fetching instruction proposed in section II, to the model. In particular, through an analysis of FSM, the slowing method was emulated on a test bench level using Verilog. The pre-fetching method was realized by modifying the model's fetch module and re-synthesizing the modified RTL. Since most previous works [1]-[3] verified their concepts only on control-path-dominant benchmarks such as MCNC and ISCAS'85, and there is, to our knowledge, no benchmark suite where we can evaluate the impact of our methods on the datapath of a large-scale system, we used an example circuit from [7] that has a complicated datapath structure.

Here, the following equation was used to estimate the fraction of cycles during which the sub-FSMD can be turned off when the actual return time (T_{Return}) of a sub-FSMD is greater than its BET, β:

$$\text{Idle sub-FSMDs (\%)} = \frac{\Sigma_{i=1}^{\text{all cycles}}(\Sigma_{j=1}^{\text{all sub-FSMD}} Z_{ij})}{\text{total sub-FSMD} \times \text{total execution cycles}} \times 100,$$

$$\begin{cases} Z = 1, & \beta < T_{\text{Return}}, \\ Z = 0, & \beta \ge T_{\text{Return}}, \end{cases}$$

For example, the circuit runs for three cycles, and five, six, and eight sub-FSMDs among the 10 in total are idle at each operation cycle. Nineteen sub-FSMD cycles (the product of the number of sub-FSMDs and cycles, calculated as 5+6+8) were idle out of a total of 30 sub-FSMD cycles, thereby achieving a 63.33% savings potential.

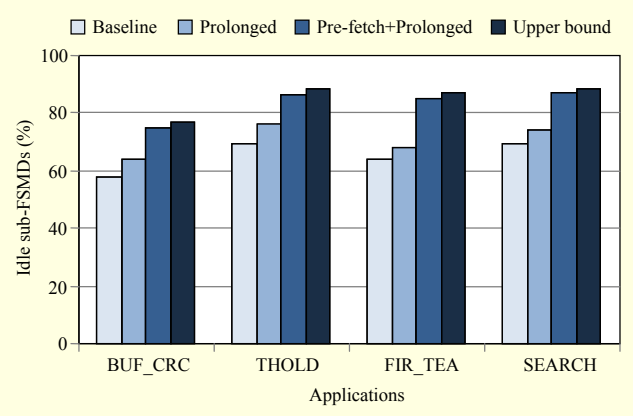

Fig. 3. Numbers of idle sub-FSMDs under four conditions measured using each test application in Sensebench.

Figure 3 shows the power savings, which is indicated with the number of idle sub-FSMDs for both methods, in addition to the baseline method and theoretical upper bound method. In this figure, a lower percentage means lower power saving potential. Under baseline, only the sub-FSMD that meets the condition "$\beta < \gamma$ and $\beta < \delta$" can be turned off; under prolonged, in addition to the idle sub-FSMDs under baseline, the sub-FSMDs that satisfy the condition "$\beta < \gamma'$ and $\beta < \delta$" (γ' and δ' denote slowed return times of γ and δ, respectively) can be power-gated; under pre-fetch, on the top of prolonged, more sub-FSMDs can be turned off by the pre-fetching; under upper bound, all sub-FSMDs except for one active one can be turned off. As shown in Fig. 3, power savings can be significantly improved when both methods are applied. For example, in the case of FIR_TEA, without any optimization (baseline), only 64.27% of all sub-FSMDs (10.28 out of 17 sub-FSMDs) can be turned off, while the maximal power saving potential (upper bound) is 86.78%. Applying the slowing method or the pre-fetch method with slowing can achieve better power saving than baseline, 68.3%, or 85.17% of all sub-FSMDs, respectively.

IV. Conclusion

We inferred that some sub-FSMDs, which are supposed to be turned off, must remain turned on due to a power overhead, and through optimizing the FSMD, the loss of saving potential can be reduced. Ideally, only one sub-FSMD is supposed to be turned on at any given moment. However, with regard to the power gating overhead, some of the other sub-FSMDs should always be turned on because their BETs might be greater than the minimum return time. The gap between the upper bound and the experimental results are largely attributed to inadequate information on determining an accurate return time. Incorrect information of the return time can lead to an unnecessary switch-on for a subcircuit whose actual return time is longer than its BET. The two methods we proposed effectively narrow the gap, as presented in the previous section, and we found that further prediction will yield only a minor improvement in exploiting the power saving potential. This finding suggests that most cases of needless activation are under the half-deterministic condition (defined in section II), and thus a misguided turn-on can be eased using a one-level instruction pre-fetch.

To extend our idea toward a general FSM for improving the usability, partitioning a datapath into power domains is essential. Although the clustering outcome depends on the design style of the RTL, as long as the RTL is well modularized, the power domain is expected to be recognized through an analysis of the state's output control signals because such signals tend to be exclusively connected to independent circuit modules, such as multiplexers, registers, and functional units.

While the FSM-based approach can help in finding more power saving potential, it creates communication overhead, particularly in terms of power. However, considering the extended scalability the communication provides, we believe the overhead is acceptable.

Also, when multiple sub-FSMDs require the datapath module in a sub-FSMD, they can share the module like the example of C1 in Fig. 1. At the expense of the area redundancy, sharing a module reduces saving potential to some extent because the proportion of the module in total power expenditure increases. As an alternative, a circuit synthesizer can duplicate the shared module, thereby gaining the advantages of less dynamic power consumption and faster operation as a result of eliminating interconnect circuitry.

Furthermore, our FSMD-based power gating and its optimization can be adopted by a high-level synthesizer, enabling a circuit exploiting more saving potential to be automatically synthesized. We will deal with these issues in future work.

Acknowledgement

The authors owe our deepest gratitude to Prof. Yong Jae Suh of University of Science and Technology for his invaluable guidance and keen intellectual judgment on writing up our research.

References

[1] S. Bhunia et al., "A Novel Synthesis Approach for Active Leakage Power Reduction using Dynamic Supply Gating," *Proc. DAC*, 2005, pp. 479-484.

[2] K. Usami and N. Ohkubo, "A Design Approach for Fine-Grained Run-Time Power Gating Using Locally Extracted Sleep Signals," *Proc. ICCD*, 2006, pp. 155-161.

[3] H. Xu, R. Vemuri, and W.B. Jone, "Temporal and Spatial Idleness Exploitation for Optimal-Grained Leakage Control," *Proc. ICCAD*, 2009, pp. 468-473.

[4] B. Yu and M.L. Bushnell, "A Novel Dynamic Power Cutoff Technique (DPCT) for Active Leakage Reduction in Deep Submicron CMOS Circuits," *Proc. ISLPED*, 2006, pp. 214-219.

[5] E. Hwang, F. Vahid, and Y.C. Hsu, "FSMD Functional Partitioning for Low Power," *Proc. DATE*, 1999, pp. 22-28.

[6] N. Agarwal and N. Dimopoulos, "FSMD Partitioning for Low Power Using ILP," *Proc. ISVLSI*, 2008, pp. 63-68.

[7] M.H. Oh, C.H. Shin, and S.W, Kim, "Design of Low-Power Asynchronous MSP430 Processor Core Using AFSM Based Controllers," *Proc. ITC-CSCC*, 2008.