A System-level Power-estimation Methodology based on IP-level Modeling, Power-level Adjustment, and Power Accumulation

Masafumi Onouchi[†], Tetsuya Yamada[†], Kimihiro Morikawa[‡], Isamu Mochizuki[‡] and Hidetoshi Sekine[‡]

†Hitachi Ltd., 1-280, Higashi-Koigakubo, Kokubunji-shi, Tokyo, 185-8601, Japan ‡Renesas Technology Corp., 5-20-1, Josuihon-cho, Kodaira-shi, Tokyo, 187-8588, Japan

Abstract— We have developed a specialized rapid power-estimation methodology for multimedia applications. This methodology has adequate accuracy for the first design of a complicated SoC. For a multimedia application, we developed three new methodologies: an IP-level modeling, a power-level adjustment methodology, and a power accumulation methodology. With these methodologies, the system-level power estimation on a SoC executing a practical application becomes so precise and easy that we can revise the SoC design to reduce its power. According to a comparison of the system-level power estimated with these methodologies to board-measured power, the error between the two powers is less than 5.6%.

I. INTRODUCTION

Recently, more and more intellectual properties (IPs) are being integrated on a single chip. These large-scale integration (LSI) chips are known as system-on-a-chips (SoCs). SoCs are mainly applied in digital appliances, such as cellular phones and digital cameras, because of their compactness. Executing applications with the limited power resources of mobile appliances, however, means that SoCs face a severe power restriction. To reduce SoC power dissipation, it is thus important to know which IP dissipates the largest power when executing practical applications. We call power dissipated in practical applications, system-level power. Accordingly, revising this IP's design to reduce system-level power by means of careful analysis of the estimated power, will lead to a power-aware SoC.

Power dissipation can be classified into two categories: dynamic power and leakage power. Leakage power can simply be estimated from the total gate width of the transistors on a SoC, because it is always dissipated through a transistor's gates, sources, and drains. On the other hand, it is difficult to estimate dynamic power because it depends on the switching rates of the transistor's gates and its connected wires.

Present methodologies using abstract models on a SoC

can rapidly estimate the dynamic power dissipation in the early stages of design [1–5]. However, these methodologies do not include the gate-level behaviors of a SoC, therefore the precise estimation of the system-level power is not obtained. Moreover, the practical application, such as a digital broadcast TV service or a TV phone service, is too large to simulate in reasonable time. To improve the accuracy of the estimation, the methodology using gate-level simulation on an IP was developed [6]. In this methodology the gate-level simulation, which supplies more precise estimations, was enabled by splitting both the gate-level description and the benchmark into small units. However, its so expensive to split both of them that it is impossible to simulate the whole application.

In this study, with the above issues in mind, we present rapid and precise system-level-power-estimation methodologies specialized on multimedia applications. Firstly, we estimate an IP's precise dynamic power with a simple benchmark by a gate-level simulation. Secondly, we abstract one-frame processes from the application and split these processes into individual IP processes. This is how we make an IP-level model. What makes this power estimation very simple is that the simple repetition of this model approximately equals the whole application. In calculating system-level power by adjusting the IP's precise dynamic power according to the IP-level model, we developed a power-level adjustment methodology and a power accumulation methodology. These methodologies enable us to design a power-aware SoC in the early stages of design.

II. System-Level Power-Estimation Methodologies

We developed specialized power-estimation methodologies for multimedia applications on a SoC (Fig. 1). Traditional methodologies for the estimation of dynamic power are based on an abstract model of a SoC's architecture. These methodologies, however, do not include the gatelevel behavior of a SoC, therefore it is impossible to precisely estimate the system-level power. In contrast to



Fig. 1. SoCs system-level power-estimation methodologies. The rectangular region is this work.

these methodologies, we firstly calculate the fundamental dynamic power of an IP using a simple benchmark simulation with a gate-level net list (Fig. 1, region A). This simulation is executed with such a simple but typical benchmark for each IP that the estimated power has adequate accuracy. Secondly, we developed an IP-level model abstracted from the practical application processes (Fig. 1, region B). Thirdly, to adjust the IP's fundamental dynamic power according to the IP-level model, we developed both the power-level adjustment and power accumulation methodologies (Fig. 1, region C).

As a target of this estimation, a low-power embeddedapplication SoC for cellular phones, i.e., an SH-Mobile3AS [7], was used (Fig. 2). The SH-Mobile3AS was designed by the Renesas 90-nm low-power process and consists of many IPs, such as a CPU core (SH-X2), a H.264 IP, and a camera IP. The CPU core is designed with a fine-grained clock gating to reduce the power consumption of the flipflops and a clock-tree [8]. The CPU core is a master IP, which can handle all slave IPs (H.264, camera, and so on). Slave IPs perform boot-ups and are stopped by the master IP.



Fig. 2. Photograph of sample chip integrating CPU core (SH-X2) and many other IPs.

A. Calculation of fundamental dynamic power of an IP by simple benchmark simulation

We can use two environments for calculating dynamicpower dissipation. One is a Register-Transfer Level (RTL) evaluation for calculating the indirect dynamic power dissipation. In the RTL evaluation, it does not take such a long time because only the toggle information on the flip-flops' wires is extracted by the simulation; thus, it is suitable for short TAT analysis. The toggle information indicates how many times each wire is charged or discharged, which causes power dissipation.

The other environment is a gate-level evaluation for calculating the absolute dynamic power dissipation. In the gate-level evaluation, an IP's estimated power has adequate accuracy because all gates' and wires' toggle information is obtained and dissipated power on each cell and wire is calculated. The gate-level evaluation has two versions: pre-layout and post-layout evaluations. The postlayout version has high accuracy with a synthesized clocktree and a back-annotated net load. The pre-layout version has no clock-tree or net load. The power dissipated on the clock-tree has a large uncertainty, because the clock-tree structure has not yet been synthesized. Therefore, we use pseudo-CTS (clock-tree synthesis) for the prelayout version [8]. In pseudo-CTS, clock-buffer trees are inserted into a clock-tree net, whose fan-out and structure are similar to those of the post-layout.

The gate-level evaluation consists of the two stages shown in Fig. 3. Firstly, a simple benchmark is simulated with a gate-level net list to obtain the toggle information. The simulation generally takes a long time to get the toggle information of all wires. Even the gate-level evaluation can be executed in hours, because a simple benchmark, such as a one-frame H.264, decodes a small picture. Secondly, the dynamic power of IPs is calculated with power libraries and the toggle information obtained in the first stage. We named this calculated power, fundamental dynamic power. This power is classified into circuit components, such as clock-trees, random logic, flip-flops, and SRAM.



Fig. 3. Evaluation flow of fundamental dynamic power of IPs with gate-level net list.

The fundamental dynamic power on the SH-Mobile3AS estimated by the pre-layout gate-level simulation is shown in Fig. 4. This figure shows the IP's name, dynamic power, and the number of operating cycles for several benchmarks. The processes, such as the rotation, format transformation, and display, are less cost effective than H.264 decoding. Several of these processes can be simultaneously executed on the camera IP.



Fig. 4. Fundamental dynamic power and number of operating cycles for simple benchmarks.

B. IP-Level model of a practical application

For abstracting a practical application we developed an IP-level model of it. A digital broadcast TV service, which is a new multimedia application on third-generation cellular phones, was assumed. The digital broadcast TV, which can supply high-definition movies, applies H.264 decoding to animation and AAC decoding for sound.



Fig. 5. IP-level model in digital broadcast TV service.

Figure 5 illustrates the IP-level model of the digital broadcast TV service. In this model, the application is split into each IP process in executing a one-frame picture. For example, the CPU, as a master IP, handles some slave IPs and executes AAC decoding for sound. A slave IP (H.264 IP and Camera IP) executes the "H.264 decoding," "rotation," "picture-format transformation," and "display" in sequence. Its data are transferred by a bus, peripheral IP, and SDRAMs. The picture size is QVGA (320×240 pixels) for all processing, and the picture format is converted from a YUV format to an RGB format in process 3 in Fig. 5.

Power estimation becomes very simple if we focus on the fact that the repetition of this model approximately equals to the whole practical application, namely the averaged power dissipated in one-frame-picture processing approximately equals that of in the whole multimedia applications. Therefore system-level power can easily be estimated by summing up only the IPs power contained in this model.

C. Power-level adjustment methodology and power accumulation methodology with fundamental dynamic power



Fig. 6. Power-level adjustment methodology using each fundamental dynamic power and operating cycles.

To adjust the fundamental dynamic power according to the IP-level model, the power-level adjustment methodology was developed (Fig. 6). In this case, we also used the digital broadcast TV service. In Fig. 6, the vertical length of the rectangle denotes the fundamental dynamic power, and the horizontal length of the rectangle denotes the IP's adjusted operating cycles. In the slave IPs, the operating cycles are mainly determined by the picture size. For example, in the H.264 IP, the number of macro blocks, which are fundamental units in encoding/decoding picture, determines its operating cycles. The number of macro blocks doubles, so twice the number of cycles is required. In this model, we assume that the clock signal for the slave IP is running only when the IP is executing the processes. In the master IP (CPU), we estimate the operating cycles using an instruction-set simulator (ISS). Lastly, in the bus and peripheral IPs, we use the averaged IP power executing the benchmark. Because these IPs are almost always active when the other IPs are executing the processes, and power is mainly dissipated in the clock-tree and flip-flops, that the contents of the transferred data do not affect the power dissipated in the random logic circuits. Therefore, we can obtain each IP's actual operating cycle by adjusting it with this methodology.



Fig. 7. Accumulated system-level power.

To obtain actual system-level power, we developed the power accumulation methodology. Firstly we estimate each IP's actual dynamic power by calculating the area of the rectangle(Fig. 6). Secondly we accumulate these powers and other components such as the "global clock" and "leakage" power. The "global clock" includes the clock generation and clock distribution from a phase locked loop (PLL) to each IP. The leakage power is mainly from the transistor's gate and subthreshold leakage.

III. EXPERIMENTAL RESULTS

To compare the estimated system-level power and an actual power, we measured the SH-Mobile3AS board under almost the same conditions as the estimation in section II.

In this estimation, because we use the gate-level net list of the pre-layout version, the value estimated in section II needs to be revised for the post-layout version. In the post-layout version, many extra buffers are inserted to fix the hold-time violations, and a number of transistor's gate sizes enlarge to meet the set-up-time constraints [8].



Fig. 8. Comparison between revised system-level power and board measured power for sample chip.

The system-level power and board-measured power are shown in Fig. 8. This figure includes "debug" power, which is dissipated in the executing software for the debugging, and estimated from the board-measured power when the debugging software is executed on the CPU.

According to this comparison, the revised system-level power is 5.6% smaller than the board-measured power. The accuracy of this estimation is adequate enough for the first design of a SoC. There are mainly two reasons for this error. Firstly, because the software used in the board measurement is not fully optimized, the clock signal for the slave IPs is not completely stopped when these IPs do not execute any processes. The second reason is the lack of accuracy in estimating each IP's load. With more mature software and more accurate estimations of IP's loads, the accuracy of the estimation will improve.

IV. CONCLUSION

We developed a IP-level modeling, a power-level adjustment methodology, and a power accumulation methodology for the system-level power estimation on a complicated SoC. Firstly, an IP's fundamental dynamic power is estimated with a gate-level simulation. Secondly, we make an IP-level model, abstracting a practical multimedia application. We then can obtain the system-level power only by calculating the averaged power dissipated in this model, because the repetition of this model approximately equals the whole practical application. To adjust the fundamental dynamic power according to the IP-level model, a power-level adjustment methodology and a power accumulation methodology were developed. Therefore, we can quickly estimate the system-level power without executing a time-consuming simulation. A comparison of the estimated system-level power with these methodologies to a board-measured power shows that the error between the simulations is less than 5.6%.

V. Acknowledgements

The authors would like to thank Kunio Uchiyama, Naohiko Irie, Kenichi Osada, Hiromi Watanabe, Hiroshi Hatae and Hiroshi Ueda for their continuing support.

References

- D. Brooks et al., "Wattch: a framework for architectual-level power analysis and optimizations," *Proc. of the 27th ISCA*, pp. 83-94, 2000.
- [2] A. Dhodapkar et al., "TEMPEST: A thermal enabled multimodel power/performance estimator," Proc. of the Workshop on Power-Aware Computer Systems, pp. 112-125, 2000.
- [3] S. Gunther et al., "Managing the Impact of Increasing Microprocessor Power Consumption," Intel Tech. Journal, Q1, 2001.
- [4] D. Brooks et al., "New methodology for early-stage, microarchitecture-level power-performance analysis of microprocessors," *IBM Journal of Research and Development*, Vol. 47, Num. 5/6, pp. 653-670, 2003.
- [5] K. M. Büyükşahin et al., "Early Power Estimation for VLSI Circuits," *IEEE Trans. on Computer-aided design of integrated circuits and systems*, Vol. 24, pp. 1076-1088, 2005.
- [6] Y. Nakamura et al., "A Fast Chip-Scale Power Estimation Method for Large and Complex LSIs Based on Hierarchical Analysis," ISCAS, pp. 628-631, 2005.
- [7] M. Saen et al., "Elastic Shared Resource Scheduling SOC Interconnect Architecture for Real-time System," *CICC Dig. Tech. Papers*, pp. 787-790, 2005.
- [8] T. Yamada et al., "Low-power design of 90-nm SuperHTM processor core," *ICCD Dig. Tech. Papers*, pp. 258-263, 2005.