

# OS-aware Performance and Power Analysis Methodology

Hyunjae Woo, Hojin Jo, Woojoo Kim, Youngsik Kim, Seonil Brian Choi

Samsung Electronics Co., LTD. (hyunjae.woo@samsung.com)

**Abstract:** The performance and power are the most important key words in current SOC IPs and it highly depends on each other, thus the performance estimation should be considered within power budget and peak power from SOC and PMIC spec. Since the functionality of IPs are getting complex, it is hard to generate real represent test vector without real driver on OS. In this paper, we propose OS-aware performance and power analysis methodology. The main purpose of this methodology is to estimate accurate performance and power, considering real Benchmarks or day of use scenario on OS at the pre-silicon stage.

## 1. Introduction

The accurate performance analysis is one of the most important factors for IPs and SOC quality. If there are errors on performance estimation, IPs and SOC might be optimized toward wrong direction and this may cause both front-end and back-end design issues, resulting in project fail. For accurate performance analysis, following two items should be secured as follows:

The first item is realistic test vectors, which are significantly necessary but require huge efforts to create. The current IPs' design size and functional complexity are continuously increasing, so bare-metal test (firmware test) is not sufficient to verify all functionality for realistic test cases that mimic real use scenario of device set. That is why the execution test of real application SW based on real OS driver is highly recommended for measuring performance.

The second item is power consumption. The performance analysis without considering power consumption is meaningless, since power consumption may causes various issues such as performance limit due to power budget, PMIC burning, or thermal tripping. So the performance and power should be analyzed on a same environment with same vectors.

Even though existing power analysis methodology based on emulator can accelerate 3000 times faster than simulation [1], however, it is not enough to handle all real applications. Hence, we propose a new methodology, which can overcome the challenges described as above. The rest of this paper is organized as follows: the previous works and challenges on both performance and power measurements are discussed in section 2. Section 3 describes our proposed methodology and shows how to overcome challenges. The experiment results are described in section 4. Section 5 presents the conclusion.

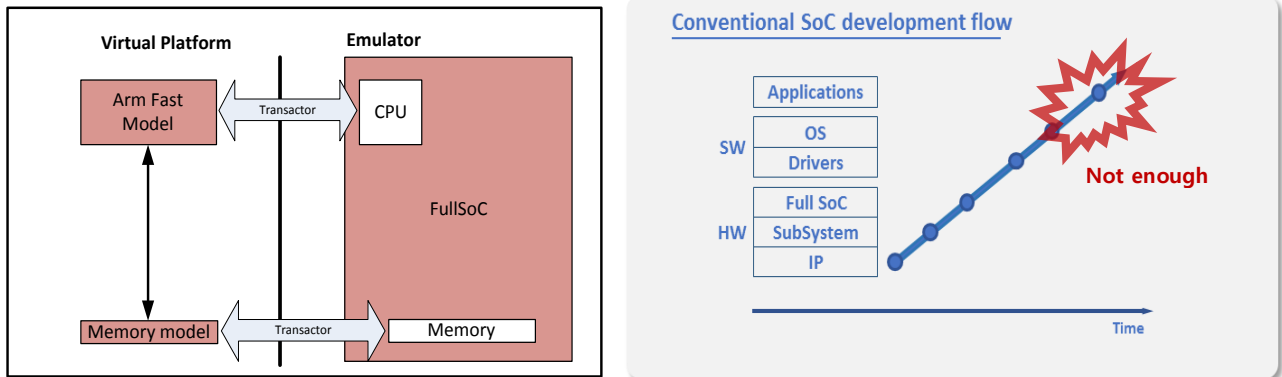
## 2. Related Research (Previous work)

This section introduces conventional (existing) performance and power analysis flow that utilizes emulation for acceleration. At the early stage of a SOC design project, simulation is mostly preferred to verify IPs and components because it is easy to setup an environment and is fast to compile the RTL DUT. When the project moves on and RTL becomes matured, then the emulation starts to be used to perform both performance and power analysis.

### 2.1 The performance measurement and analysis

The performance should be measured by realistic test scenario and it is getting harder to generate, so in this section only focusing on running real benchmark on OS. There are several approaches to do it, such as simulation, pure emulation (full chip emulation), full SOC level hybrid emulation and IP level hybrid emulation. As the design size grows, simulation speed reduces exponentially. In case of several hundred million gate scale, simulation speed can be dropped to a few dozen KHz [2], so it is expected to take about 602 days only booting up Android OS[8]. Pure SOC emulation speed can be from several hundred KHz to few MHz [3][4] and it takes 13 hours to boot-up Android

OS so it is used to use only for bare metal tests or Linux based simple SW initialize tests. The Figure2.11 show the full SOC level hybrid emulation and its limitation. Full SOC level Hybrid emulation methodology is the best solution to overcome TAT constraint of pure HW emulation. It uses co-emulation methodology. CPU and memory components are allocated to ESL virtual platform side with TLM LT (Loosely Timed) models, because they issue many transactions during boot up time of Android Platform and it takes only 49 min to boot up Android OS. Nevertheless, there are two main issues, one is that there are not enough time because all IP and IP's SW should be ready. The others is that it is painful to debug full SOC because many IPs and IP's SW can occur the bugs so even narrow down which IP or IP's SW issue the bug, are time consuming.

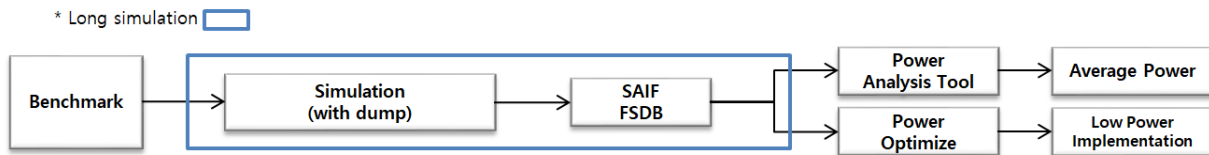


**Figure 2.11 Full SOC level Hybrid Emulation & Limitation**

IP level hybrid emulation methodology is accelerating 5 times and solves above limitations [6]. This methodology is for the SW early development and HW-SW co-validation focusing on IP itself, but it is good to use for performance and power measurement and analysis. Because the proposed methodology operates based on IP level hybrid emulation platform, it is shown more detail later at section 3.

## 2.2 The power measurement and analysis

The simulation-based power flow is shown the Figure2.21, during simulation SAIF or FSDB is dumped to measure average power or optimize implementation. It is still the major methodology in this field, but there are huge limitations because of long TAT. Therefore, only small IPs or short test case can be applicable. There are several technologies & methodologies to overcome long simulation time such as multi-core simulation, fsdb replay based parallel simulation (Siloti) and RTL based power analysis. Even though using those, however, it is just few times faster and it is still not enough to measure power for long test vector and huge IPs such as GPU, ISP. Therefore, it is hard to measure precise silicon power for real silicon usage.



**Figure 2.21 Simulation based Power Analysis Flow**

To overcome these limitations on simulation, emulation based power measurements are getting popular. The Figure2.22 shows conventional emulation based power analysis flow. The intermediate waveform dump of simulation is accelerated by emulator. Once it is dumped, compute servers performs post processing to convert intermediate waveform into SAIF(Switching Activity Interchange Format) file which is used for power estimation.



**Figure 2.22 Emulation based Power Analysis Flow**

Figure 2.23 shows peak power period analysis which extracts SAIF for peak power consumption period, that is also performed during post processing. Suppose the total simulation time is 500us. Then the simulation time is divided into 10 pieces (such that each piece is 50us period), and the average power consumption of each piece is calculated. After that, the piece which has biggest average power consumption is chosen. In our example, T9 piece is selected. The next step is to divide this T9 piece into 10 pieces such that each piece's duration is 5 us period. Then the average power of each piece are calculated, followed by selecting the piece that has biggest average power consumption. (T9-7 in our example) Similar process is performed for one more time, resulting that the period of a piece is 0.5us and the average power consumption piece is chosen as T9-7-4. In our example, we stop this iterative process at this point since the specification of our PMIC(Power Manager IC) defines the minimum period of peak power as 500ns, and we assume that the peak power is same as the average power consumption of T9-7-4 period. However, the actual peak power consumption happens in T3 piece. The reason that we missed this peak point is that our existing analysis flow is based on the assumption that real peak power consumption happens at the piece, of which the average power consumption is bigger than other pieces.

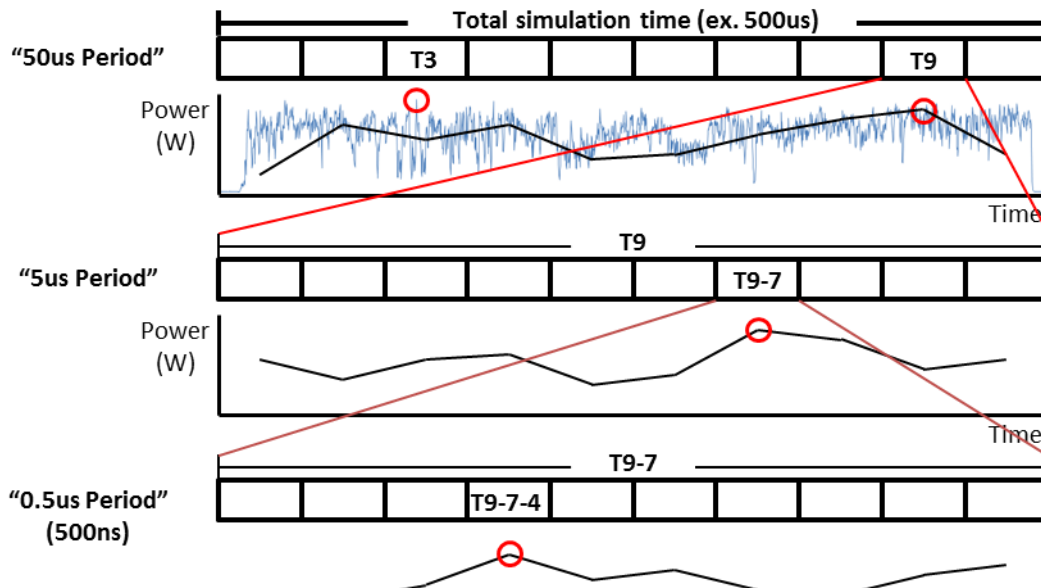
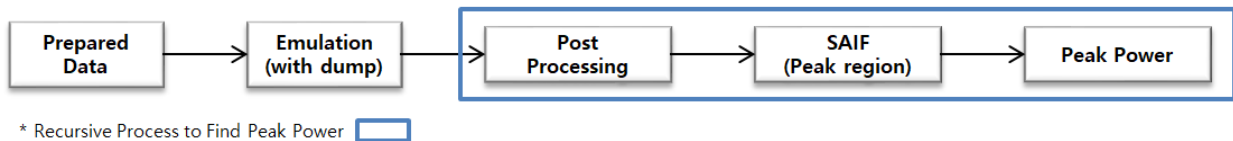


Figure 2.23 Peak Power Region Analysis Flow

The WTC(Weighted Toggle Count) overcome above issues. The equation at Figure 2.24 shows the basic idea about WTC, therefore when the voltage and frequency are constant value, activity factor that is toggle information, and capacitance would represent dynamic power consumption [7]. Each cell's liberty files have the internal power information for both falling and rising toggle. Therefore, if toggles are multiplied by weight from their liberty's power and that weighted toggles are summated every single cycle then the shape of this plot is almost same as that of dynamic power plot. As shown in Figure 2.25, the WTC and SAIF file for the peak power estimation can be extracted from intermediate waveform.

$$\text{Dynamic power} \propto \alpha CV^2f$$

$\alpha$  = activity factor a times, C = capacitance, V = voltage and f = frequency

Figure 2.24 Basic Equation of Dynamic Power



Figure 2.25 Peak Power Region Analysis Flow

Even though this conventional emulation based power analysis methodology is 3000 times faster than simulation and can cover the peak power, there are still two main challenges described as follows. The first challenge is that we cannot run full frame of benchmark because of TAT and repository issues. As the GFX Benchmark for GPU, only 1 frame take 3~4 hour to run with dumping intermediate waveform and the intermediate waveform is around 600GB, that is why we only measure power by selected 4 major frame. GFX benchmark is about 62000ms long, and if run only one second, it will take 400 hours and 60TB roughly estimated. The second challenge is that the emulator is zero-delay based simulation, so there is no glitch during power measurement. In the past, the glitch effect would be ignorable or simply multiply by constant values like 1.1, since deep submicron technology that means the process is getting shrink, net switching power portion is being increased, and even now, net switching power is bigger than cell internal power. Therefore, SDF (Standard Delay Format) aware power measurement is highly necessary.

### 3. Proposed Methodology

The proposed methodology consists of emulation platform and flow for measuring power and performance.

Again, it is crucial that the performance and power should be measured at the same simulation environment, configurations and constraints by the realistic test scenarios. The following sub-sections explain more detail about how to implement platform and how to measure the power and performance.

#### 3.1 The proposed platform

We focused on IP's performance and power analysis on OS with real device driver, so QEMU[5] based IP level hybrid emulation technic was adopted [3][4]. We have built a platform such that all HW components related to OS booting are located at virtual side, and the target IP and related components are located at real emulation side, as shown in Figure 3.11. Virtual Ethernet is necessary at virtual side, which downloads APK. The performance monitor is bus traffic analyzer to show AXI bus traffic for the average, min and max latency, average and max MO (multiple outstanding), bandwidth. The most important transactor is the coherent I/F which is the synchronous memory between virtual side and emulation side. We tried to develop general transactor as cache based system to support the all kind of emulators but we fail to optimize performance, so we just use commercial transactor from vendors.

All major emulator vendors have their own memory transactor, which are named by Coherent memory, Smart memory and Fast memory, which are fully optimized for their emulator.

Building subsystem that includes DUT as well as bus fabric and monitors is always painful because there are many signal interconnections and it occur human mistake. That is why we made automation flow that builds the subsystem based on IPXACT.

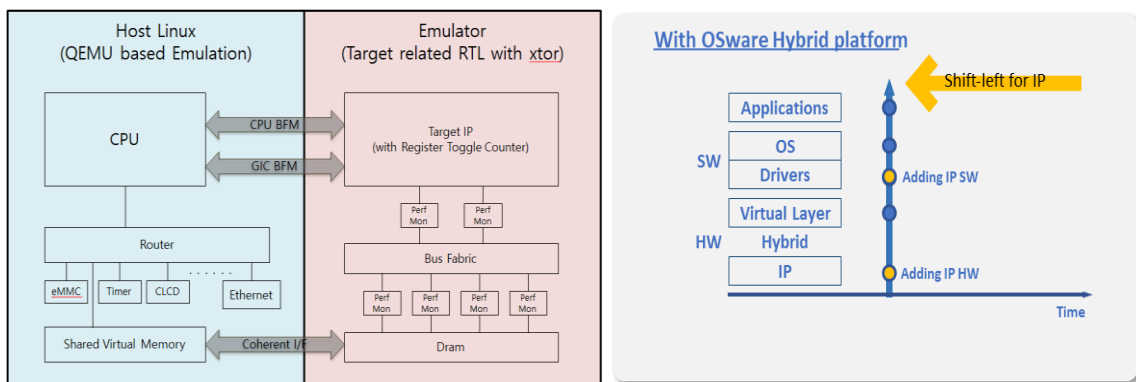


Figure 3.11 Proposed Platform & Effect of Platform

In this platform, it takes 22 minutes to boot up Android Pie, 8 minutes to load GFX Benchmark and approximately 2 minutes to render each frame without power dump.

### 3.2 The proposed power and performance measurements flow.

As we introduce the section 2.2, there are two limitations. The proposed flow includes the solutions to overcome those limitations. One is the fast power profiling and the other is the SDF aware FSDB re-simulation. The fast power profiling is the on the fly based toggle summation by inserting toggle counters in emulator. During emulation, the final toggle summation value store at the internal buffer and only dump to host-machine when internal buffer is full. The key idea is dramatically reduced data between emulator and host-machine to dump so it does not hurt emulation speed and there is no more storage issue.

Figure 3.21 shows the SDF aware power flow. The FSDB Re-simulator can replay as FSDB with SDF to generate VCD or FSDB. The results of waveform are used to use for accurate power estimation and physical power verification. In this process, it takes long time to convert intermediate waveform to FSDB and replay with SDF but it is quite enough for peak period.

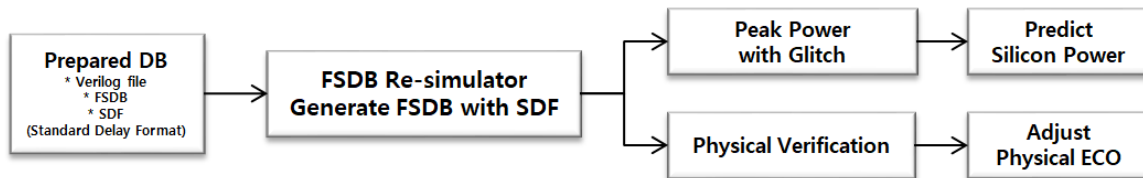


Figure 3.21 Power Analysis Flow with SDF(Standard Delay Format)

The flowchart of proposed performance and power analysis methodology is shown in Figure 3.22

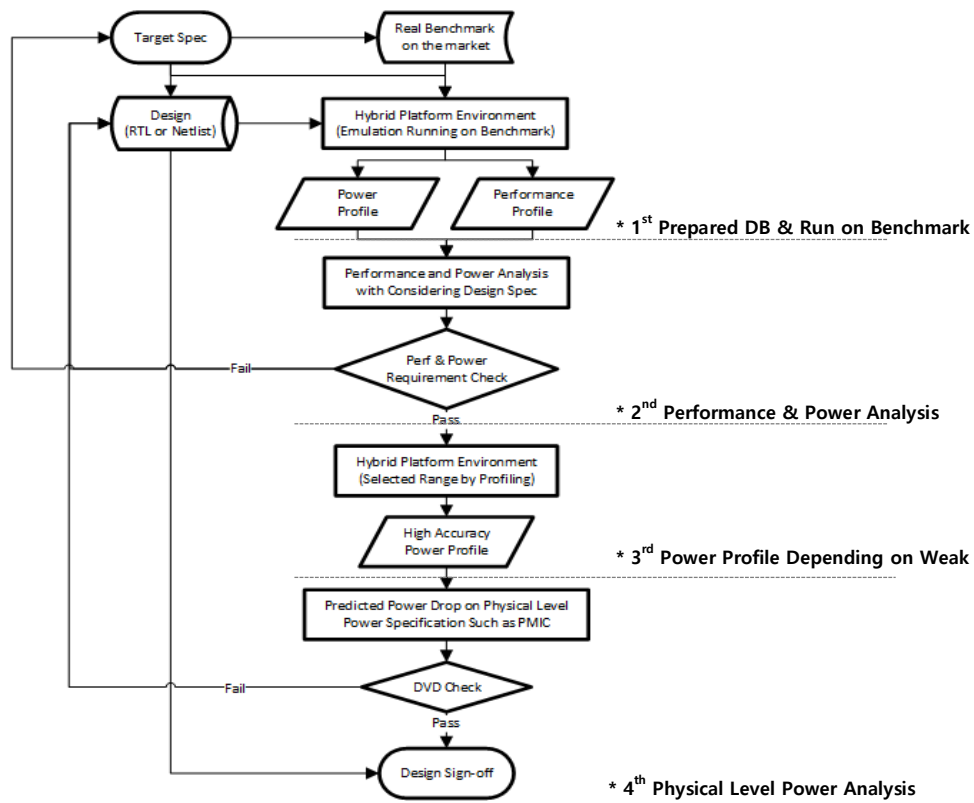


Figure 3.22 Flowchart of Proposed Methodology

Once target spec and design are ready, benchmark to run on proposed platform is selected. The first process in flowchart is running benchmark for the fast power profiling and performance measurement. Because the fast power

profiling is used, there is no emulation speed drop therefore, we can run long benchmark. The results of first process are the toggle plot, performance monitor data and performance score or performance counters from the IP.

The second process is the performance analysis and power profiling, if the performance does not meet the requirements then target spec should be changed such as increasing operation frequency or SOC configuration change and rerun all. If the performance meets the requirements then power profiling is analyzed to find critical range of power. Figure 3.23 shows the input of this process for the power profiling. As our experimental, the fast power profiling can be lost real peak so when we do this, we select 5 critical period candidates to secure the quality.

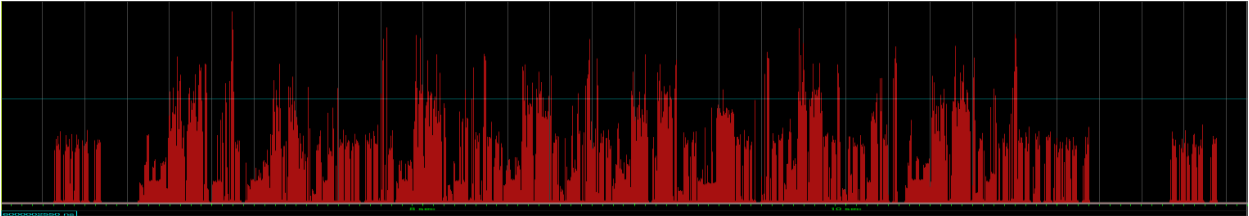


Figure 3.23 Power Profiling Result 1st Prepared DB & Run on Benchmark

The third process is the rerun of the benchmark to dump selected range, which represents the average power and the peak power period for the accurate profiling. The results of third process are multiple intermediate waveforms.

The final process is to analyze accurate power profiling by WTC. Figure 3.24 is the result of WTC, which is used to find peak period. After we select the interesting periods, it is time to convert intermediate waveform to SAIF file or FSDB for the average and peak power measurements. The FSDB re-simulator is used to generate SDF aware VCD that is for checking DVD (Dynamic Voltage Drop) sign-off. If the measured power fails to meet the power spec or DVD sign-off, all process should be rerun after modifying implementation or changing SOC configuration.

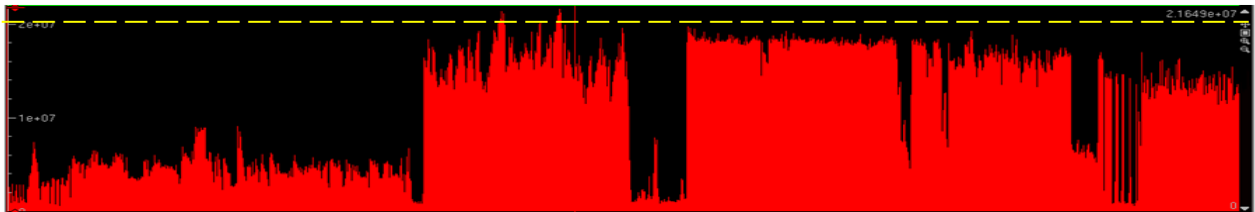


Figure 3.24 Power Profiling Result 3rd Power Profiling Depending on Weak

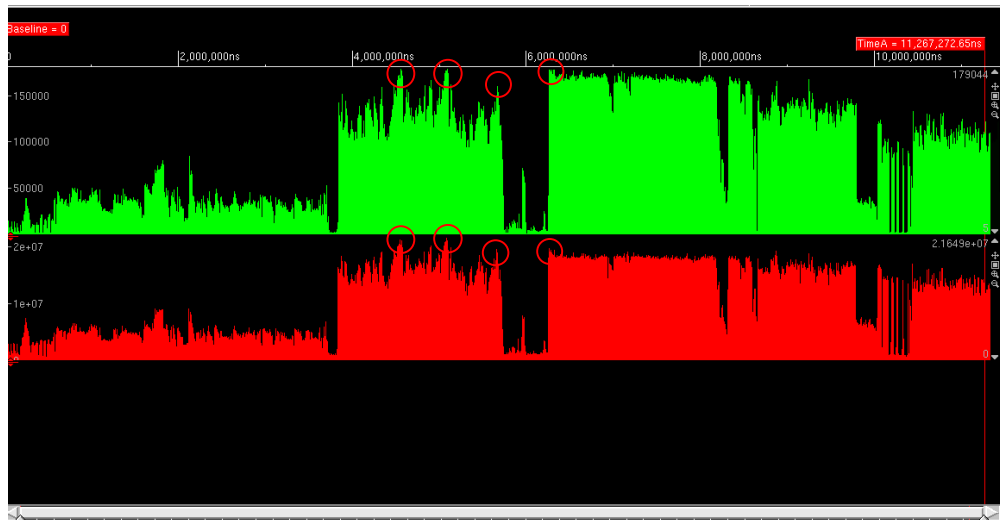
#### 4. Experiment results:

We adopted this proposed methodology on our GPU and NPU design to analyze PPAB (Performance Power Area Bandwidth). Figure 4.1 shows the comparison between existing methodology and proposed methodology for one frame time period of GFX benchmark. The existing methodology takes 7 hours including runtime and post processing time, and 29 Linux host machines are used. However, proposed fast profiling methodology only takes 3 minutes. In addition, proposed methodology does not require huge amount of compute farm.

Power Profiling Method Compare table		
	Exsiting Method	Proposed Method
Runtime	4hours	3minutes
Post Processing	3hours (Mandatory)	None
Resource	Emulator + Comuting Farm	Emulator
Accuracy	99% Represent Profiling	85~90% Represent Profiling
Design Size	Target Design	Target Design + Toggle Counter

Figure 4.1 Comparison Result According to Method

Figure 4.2 shows the accuracy of our proposed methodology, by comparing the power consumption graph that are generated by existing methodology (in red) with the graph that are generated by proposed methodology (in green). The red circles in the figure are candidate points of peak power consumption. Even though the shape of two graph is slight different, the green graph does not lose any peak candidate that are shown in the red graph.



**Figure 4.2 Comparison WTC and fast power profiling**

## 5. Conclusion:

In this paper, OS-aware performance and power analysis methodology is proposed. In order to enhance boot-up speed, virtual domain is adopted and to measure accurate performance and power values, RTL emulation scheme is also adopted. The performance and power consumption can be measured more accurately by considering their impact on each other. Furthermore, the probability of local minima could be reduced by using real benchmark SW on OS. As a result, within a day, OS boot-up and benchmark SW running can be done, furthermore, we can get more accurate performance and power value. Finally, all of these experiments are completed in pre-silicon stage, the proposed methodology can complete optimization of IP/SOC architecture at the RTL design stage.

## References

- [1] Hojin Jo, Jaewon Jeon, Hyeongjin Kim, Hyunjae Woo, Youngsik Kim, Seonil Brian Choi, "Rapid IP and SOC Power Dissipation Analysis By Leveraging Emulation," in Proc. DAC, LV, USA, June 2019
- [2] Y. Nakamura, K. Hosokawa, I. Kuroda, K. Yoshikawa, and T. Yoshimura, "A Fast Hardware/Software Co-Verification Method for System-On-a-Chip by Using a C/C++ Simulator and FPGA Emulator with Shared Register Communication," Proc. DAC, pp. 299-304, San Diego, CA, USA, June 2004
- [3] C.Y. Huang, Y.F. Yin, C.J. Hsu, T. Huang, and T.M. Chang, "SoC HW/SW Verification and Validation," Proc. ASP-DAC, pp. 297-300, Yokohama, Japan, Jan. 2011.
- [4] M. Vavouras, K. Papadimitriou, and I. Papaefstathiou, "High-Speed FPGA-Based Implementations of a Generic Algorithm," Proc. ICSAMOS, pp. 9-16, Samos, Greece, July, 2009.
- [5] QEMU is a generic and open source machine emulator and virtualizer. More detail information is at <https://www.qemu.org/>
- [6] Hyunjae Woo, Woojoo Kim, Youngsik Kim, Seonil Brian Choi, "OS-aware IP Development Methodology," Proc. of DVCon, San Jose, CA, 2019
- [7] Neil H.E. Weste and David Money Harris "Integrated Circuit Design" PEARSON, 2011 Fourth Edition
- [8] W. Kim, H. Park, H. Kim, and S. Choi, "Early Software Development and Verification Methodology using Hybrid Emulation Platform," in Proc. of DVCon, San Jose, CA, 2017.