Feb.2012

Benefits of the big.LITTLE Architecture

Hyun-Duk Cho, Ph. D. Principal Engineer (hd68.cho@samsung.com)
Kisuk Chung, Senior Engineer (kiseok.jeong@samsung.com)
Taehoon Kim, Vice President (taehoon1@samsung.com)
System LSI Business
Samsung Electronics

White paper

SAMSUNG ELECTRONICS

Abstract

The demand for performance from portable computing devices, such as tablet computers and smartphones, has been steadily rising. As an immediate consequence of this trend, computing- and display-intensive applications, such as gaming and high-definition video playback are turning out to be enormous drains on the device's power source, that is, the on-board battery. Current generation handheld computers and smartphones need high-performance CPUs, for instance, quad-core CPUs running at or above 1.5 GHz, to leverage all of the device features. However, batteries from the present generation are incapable of supporting very high computing power required in such devices, thus fail to provide considerable durations of usage to users. Moreover, when such processors are combined with high-resolution LCDs in any device, the drain is significantly higher and much quicker.

One of the proposed solutions to increase battery life in such devices is to use CPUs with a different, power-efficient architecture. This paper introduces the "big.LITTLE" architecture launched by ARM®, and its adoption for the Exynos line of processors by Samsung. The paper also discusses several key advantages of this architecture, compares it to other competing architectures, and demonstrates its efficiency and superiority over other architectures, especially in extending battery life in portable, handheld computing and communications devices.

1. Introduction

Rapid advancements in technology have brought about a remarkable increase in the performance of the CPUs used in mobile computing devices, such as smartphones and tablet computers. Operating frequencies or clock rates of such CPUs have increased from a few hundred MHz to several GHz, and the devices have expanded from having a single core to multiple, high performance, separate cores. Due to this rise in available computing power, a mobile computing device today can effortlessly run applications only a traditional computer, such as a desktop or notebook computer, could run in the past.

However, technological developments in traditional computers have also demonstrated equal growth. A significant and noticeable gap remains in the performance between a mobile computing device and a traditional desktop or laptop computer.

Current generation technology allows pushing the performance limits for mobile computing devices up to a certain extent. This can be accomplished by increasing the clock rates of the CPUs used in the devices or by increasing the number of cores within the CPUs, or both. However, the immediate downside to any of these techniques is a significant rise in the power consumed by the device's CPU.

For battery-powered mobile devices, this translates to a higher drain on the device's battery, resulting in a reduced battery life and lower usable duration of the device for the user. It may be noted that in contrast to traditional computing devices, battery capacities for mobile computing devices cannot be increased beyond certain limits, due to the physical and form-factor limitations of the device. As an example, it may not be possible to increase the size of displays on smartphones and tablet computers beyond, say, 5" and 11", respectively. Such restrictions limit the sizes and capacities of the batteries designed to power such devices.

To achieve higher computing performances at the same or lower power consumption levels in mobile computing devices, several new processor architectures have been designed. One of these is the asynchronous clock architecture, which uses the concept of running one core at a lower voltage and clock frequency, and the other at a higher voltage and clock frequency, in a dual-core CPU. Such a CPU schedules computationally intensive tasks for the faster core, while less demanding tasks are diverted to the slower core.

Samsung has adopted a different architecture, known as the big.LITTLE architecture of ARM®, for its Exynos line of high performance CPUs, designed specifically for the mobile computing and communications platform. The big.LITTLE architecture combines multiple cores to create a CPU that can execute both high- and low-intensity tasks in the most efficient manner. The following sections of this paper introduce the big.LITTLE architecture, and demonstrate its superiority over the asynchronous clock architecture in terms of both performance and power consumption.

2. Overview of The big.LITTLE Architecture

The big.LITTLE architecture is designed using the technique of employing separate cores with different computing powers within the same CPU. Figure 1 illustrates the block diagram of a representative CPU designed using the big.LITTLE architecture.

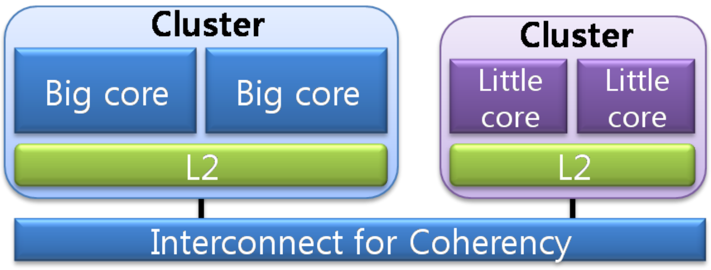


Figure 1: The big.LITTLE architecture

The more powerful (or "big") core is responsible for handling computing intensive tasks, such as high-definition video playback, whereas the less powerful (or "LITTLE") core handles lesser demanding tasks, such as text editing. Both the cores implement exactly the same processor architecture (ARMv7), and are capable of executing the same instructions. The only difference lies in the way the cores handle the execution. While the "big" core is designed with performance as its primary goal, the "LITTLE" core is designed with efficiency as its principal target. Thus, an application or program designed to run on one core would also run on the other, but with different performance levels.

The big.LITTLE architecture from ARM uses a Cortex™ A15 processor as its "big" CPU, along with a Cortex™ A7 processor as the "LITTLE" CPU to form the combination of high- and low-powered cores within a single unit. The high performance Cortex™ A15 is an out-of-order, triple issue processor with a pipeline length of between 15 and 24 stages. The low power Cortex™ A7 is an in-order, non-symmetric dual-issue processor with a pipeline length of between 8 and 10 stages. Because of the different lengths of the pipeline in Cortex™ A15 and Cortex™ A7, the power consumption levels of both the processors are significantly different. The Cortex™ A15 is capable of delivering higher performance at the cost of lower power efficiency, whereas the Cortex™ A7 design enables it to run in the most power efficient mode.

The big.LITTLE architecture benefits from the fact that the two separate cores are identical, capable of handling the same instructions and the same higher-level software applications. This proves to be extremely beneficial in power-conscious applications, such as smartphones. The big.LITTLE architecture can be deployed in smartphones, with the "LITTLE" core handling normal telephony-related functions of the device, and the "big" core taking over control from the "LITTLE" core when higher levels of performance, such as multimedia playback, are needed.

To efficiently realize a system based on the big.LITTLE architecture, the time needed to migrate a task between the cores must be taken into account. A finite amount of time would be needed to save the state (which includes the content of all the registers and caches) of both the cores, and this time should not be too long. If the context switching latency is beyond a certain threshold, the system will suffer from noticeable performance degradation. To complete the switching in the minimum possible time, a specially designed interconnection bus, known as the CCI (Cache Coherent Interconnect) is used to transfer data among the cores, as shown in Figure 1. For an application running on a device at 1 GHz, the big.LITTLE architecture ensures that this switching is completed in less than 20,000 clock cycles (or 20 µs), a duration so small that it does not affect or interfere with the end user's application running on the device [1].

The big.LITTLE architecture also provides the flexibility of configuring the number of "big" and "LITTLE" cores within the CPU. The present version of the architecture allows up to four cores of both types to be configured, letting the designer have extensive control over the performance of the CPU. Furthermore, due to differences in computing powers, both the "big" and "LITTLE" cores occupy different areas on the die, with the "LITTLE" core taking up just 13% of the area of the "big" core. It is recommended that the number of "big" and "LITTLE" cores in a particular CPU be kept equal, to simplify the context switching and task migration software.

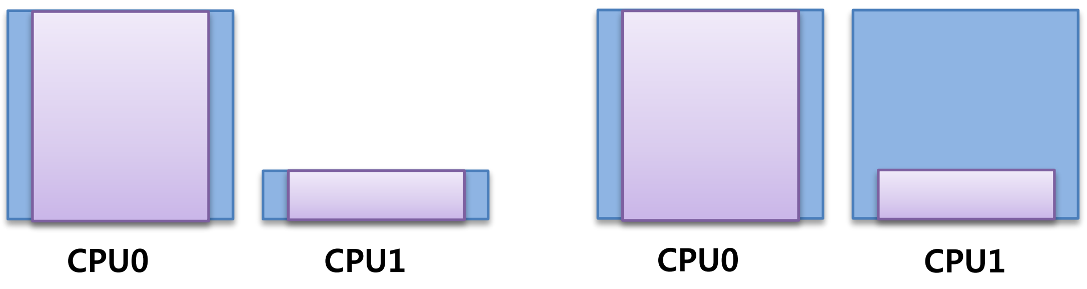
3. Comparison with Asynchronous Clock Architecture

Asynchronous clock architecture is a design for multi-core processors, in which, each core is operated at different voltage levels and clock frequencies. As stated earlier, such a design schedules computationally intensive tasks for the faster core, while less demanding tasks for the slower core. Asynchronous clock architecture can thus be considered as one of the techniques to strike an optimum balance between performance and power consumption in multi-core processor based systems.

In contrast, synchronous clock architecture implements a CPU with separate cores, with each core running at the same, fixed values of voltages and frequencies. Processors based on the big.LITTLE architecture adopt the synchronous clock architecture.

Figure 2 illustrates both asynchronous clock architecture and synchronous clock architecture. The figure depicts the interrelationship between the computing tasks (shown as the inner violet boxes) and operating clock frequencies (shown as the outer blue boxes) for both the cores on an asynchronous and a synchronous CPU in case of a dual-core processor. A CPU core with an asynchronous architecture (a) is capable of adjusting its operating clock frequency in accordance with the required computing power. Thus, for less-intensive tasks, a core (CPU1) may be operated at a lower clock, correspondingly reducing its power envelope. This is also a graphical representation of the reasons for the lower power consumption in any asynchronous architecture.

The adjoining figure (b) illustrates that a synchronous architecture is incapable of providing such computing-power dependent clock frequencies. Consequently, irrespective of the demand for computing power, the core (CPU1) always runs at a fixed, constant clock rate, without any control over the power envelope.



(a) Asynchronous Case (b) Synchronous Case

Figure 2: Power management by DVFS in both Asynchronous and Synchronous Architectures

Nonetheless, asynchronous architecture suffers from performance degradation, as explained by the following paragraphs.

Although CPU1 of the synchronous architecture processor consumes more power than CPU1 of the asynchronous architecture processor, being operated at a higher clock frequency, it completes the execution of its scheduled tasks faster. Consequently, any simple power management scheme can be deployed to turn it off (or put it in a low-power state) once it has completed executing its tasks. Overall, the net 'energy' consumption (which is the summation of 'power' consumed over a particular duration of time) in both the asynchronous architecture and synchronous architectures is the same.

Figure 3 illustrates the comparison of power dissipation and energy consumption, in both the asynchronous and synchronous architecture example discussed above.

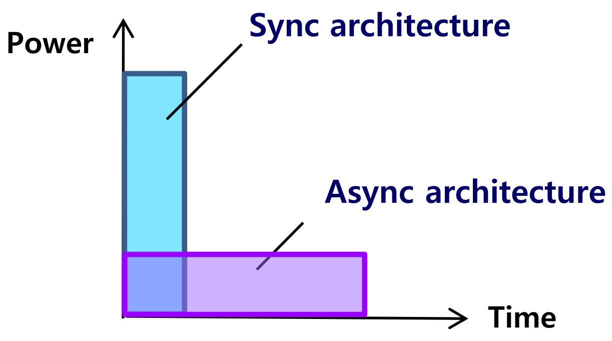


Figure 3: Comparison of power dissipation of CPU1 cores in both the synchronous and asynchronous architectures

The performance degradation in asynchronous architecture occurs during two kinds of data transfers. In an asynchronous architecture CPU, each CPU core, along with its L1 and L2 caches, and SCU (Snoop Control Unit) run from different clock domains. If an L1 cache miss occurs on one core, the requested data are transferred from the L2 cache to the L1 cache via the SCU, as shown by the arrow 'A' in Figure 4. Because of the different clock domains for the CPU, SCU, and L2 cache, additional clock cycles are required to complete this data transfer, resulting in a degradation of the overall performance of the processor.

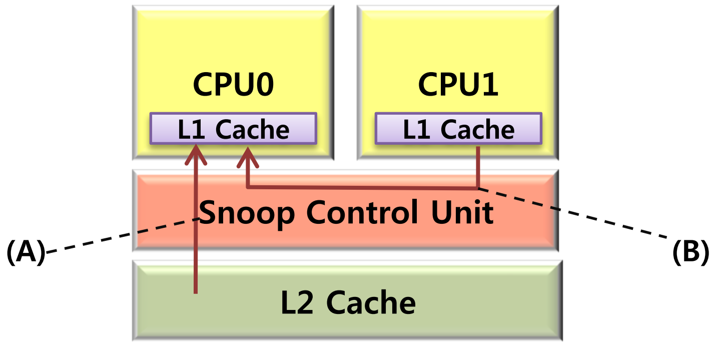


Figure 4: Asynchronous architecture depicting CPU cores, L1 and L2 caches, and SCU.
(A) represents data transfer from L2 to L1, and (B) represents inter-CPU data transfer

The second type of data transfer, which degrades the performance, is low throughputs while transferring data between CPU cores. Arrow 'B' in the above figure illustrates this. When there is an L1 cache miss on any CPU core (say, CPU0), data can be transferred from the L1 cache of the other CPU core. If the source core is operating at significantly lower clock rates than the destination core, this results in severe performance degradation, since the destination core has to wait for additional clock cycles to receive data from the source core, even when the destination core is capable of operating at much higher clock rates (see Figure 5).

In contrast to this, in a synchronous architecture, since each CPU core runs at the same clock frequency, a situation such as this cannot arise (see Figure 6).

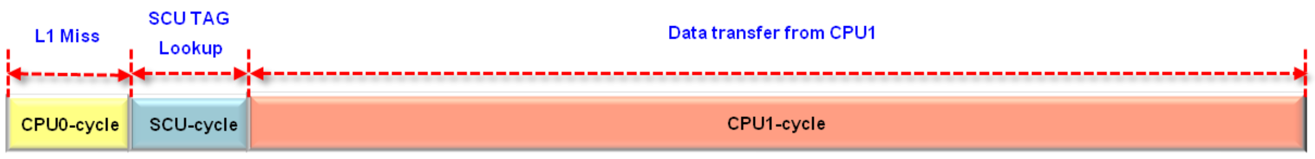


Figure 5: Inter-core data transfer cycle in an asynchronous architecture
for a CPU task load ratio of 5:1

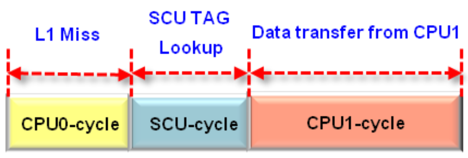


Figure 6: Inter-core data transfer cycle in a synchronous architecture
for a CPU task load ratio of 5:1

The discussions until this point in this paper have proven that the asynchronous architecture may also not prove to be the most efficient, under all operating conditions. The following paragraphs highlight the key benefits of the big.LITTLE architecture, and the advantages it brings for Samsung's Exynos line of processors.

The “mW/DMIPS” metric is commonly used to indicate the power-performance advantage that processors based on the “big.LITTLE” architecture stand to gain. Using the Cortex™ A7 core in this architecture results in a 3.3x reduction in power consumption, compared to the Cortex™ A15 core.

Figure 7 illustrates the comparison of energy efficiencies and power-performance mappings for both the asynchronous and big.LITTLE architectures.

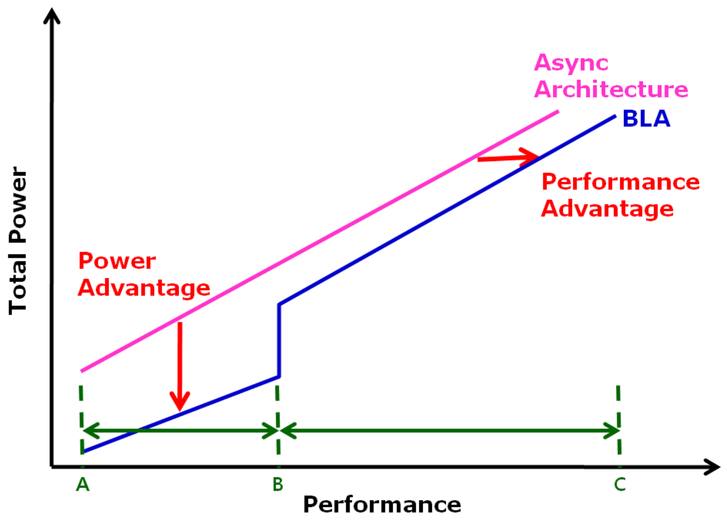


Figure 7: Comparison of energy efficiencies of the asynchronous and big.LITTLE architectures

The graph in the figure highlights two distinct sections (A to B and B to C) for the performance of both the asynchronous and big.LITTLE architectures. For applications such as text editing, which require relatively low CPU performance, big.LITTLE proves to be a superior architecture as compared to the plain asynchronous architecture, since the Cortex ™A7 core in the big.LITTLE architecture handles such applications in the most power-efficient manner. Likewise, when demand for performance increases beyond a certain threshold (such as 3D gaming), the Cortex™ A15 core from the big.LITTLE architecture takes over, and provides a significantly superior performance than that provided by the asynchronous architecture.

By and large, for a majority of commonly used applications on the mobile platform, which include email messaging, web browsing, and multimedia playback, the big.LITTLE architecture proves to be far more superior to the asynchronous architecture.

It is to be noted that frequent applications run on any mobile computing device include those that require lesser computing resources. As stated above, such applications include editing text files, and playing audio files. On the other hand, applications such as 3D gaming and high-definition video playback are seldom used on mobile devices. These statistics are used to deduce an important characteristic for the device: the net ‘energy’ consumption. ‘Energy’ is the summation of ‘power’ consumed over a particular duration of time, and one of the chief design aspects of the big.LITTLE architecture takes into account the fact that frequent uses of low-energy consuming applications prove to be a much smaller drain on the device’s power source (the battery), than occasional uses of high-energy consuming applications. The big.LITTLE architecture thus possesses a significant edge in providing longer battery lives for a majority of mobile computing and communications devices in use.

4. Conclusion

This paper discussed the need to have a different architecture for processors used in the mobile computing platform, and introduced the readers to Samsung's power-efficient Exynos line of processors based on the big.LITTLE architecture from ARM Limited. The paper presented an overview of the big.LITTLE architecture, and its key benefits in providing a superior performance at reduced power consumption. It also compared the big.LITTLE architecture with other competing architectures, such as the asynchronous architecture, and demonstrated the superiority of the former over the latter, with detailed analyses of how and when performance degradation occurs in asynchronous architecture. The paper then compared some of the key power versus performance metrics for the big.LITTLE architecture and a similar competing architecture, and showed why the big.LITTLE architecture is superior when it comes to providing high levels of power efficiency at sustained levels of performance.

References

[1] Peter Greenhalgh, "big.LITTLE Processing with ARM Cortex™-A15 & Cortex-A7 (Improving Energy Efficiency in High-Performance Mobile Platforms)", *White Paper released by ARM*, Sep. 2011.