



Power.org™ White Paper

Power Architecture® Wireless Access Market Differentiators

Version 1.0 – 06 October 2009

Jim Holt, Ph.D.
Core Architecture and Modeling Manager
Freescale Semiconductor

1. Problems/Challenges/Requirements

The wireless access market segment is dominated today by the realities of long-term evolution (LTE). Many companies are strongly committed to LTE and believe that it offers simplified technology, better scalability, and ultimately a better user experience for next-generation mobile broadband.

From its conception, LTE was intended to support downlink bandwidths of up to 300Mbps. Today some claim it may reach as much as 1Gbps. This level of performance is essential to support current and future mobile applications such as Internet browsing, interactive TV, mobile video blogging, etc. Mobile broadband subscriptions are projected to be 2.7 billion by 2014. Considering that in 2009 packet data consumed nearly 20 times the bandwidth of voice data, 2.7 billion users implies a required global wireless network capacity measured in trillions of packets per second. New classes of multimedia applications are emerging that will doubtless demand even greater bandwidth. This explosive growth rate is fueling an intense effort to rapidly provide more network throughput, while controlling the costs of infrastructure (e.g., space, cooling, and power.) This intense effort strongly affects both LTE infrastructure providers and silicon providers.

For LTE infrastructure providers, the realities of the market segment mean that they must aggressively define system architectures that will simplify and reduce the cost of network roll-out and management. Therefore, LTE providers require that the costs of both silicon (in terms of throughput-per-unit cost) and software development must decrease dramatically. Furthermore, power constraints are a primary consideration imposed by densely packed infrastructure components. These cost, programmability, and power requirements directly affect silicon providers.

For silicon providers, these realities mean enabling more processing in parallel using multicore systems. In addition to the bandwidth and power requirements, the complexity of per-packet processing required to support advanced network and application features continues to increase, demanding increasingly powerful processor cores to sustain system throughput goals. For these reasons, the LTE market segment views both system throughput and single-thread processor performance as key metrics. As a result, the silicon roadmap to support LTE must satisfy extremely rigorous demands for high levels of on-chip and chip-to-chip integration within tight power budgets. To meet these demands, there is a need for a balanced SoC Architectural approach consisting of cores, cache memories, I/O, and accelerators to achieve a specific system throughput within a tight budget. The current generation of chips integrate as many as eight high-performance, highly efficient processor cores on a device along with various networking-specific hardware accelerators. The level of integration is expected to continue to increase throughout the next decade.

In addition to requirements for high throughput, high single-thread performance, and managed power consumption, there is also a need for high-quality tools, systems software, and networking stacks. Without these, LTE providers cannot respond quickly to the growth in numbers of advanced wireless devices in use and the accompanying rapid emergence of new mobile applications. Only a potent combination of silicon and software will enable LTE infrastructure providers to compete and win in their market segment.

This whitepaper explores how Power Architecture supports the needs of the LTE market segment through (1) efficient and reliable support for heterogeneous systems integration, (2) scalability for

system-level throughput as well as single-thread performance, (3) power savings and power management features, and (4) ease of programmability and a rich software ecosystem.

2. Approach/Solution

The requirements of LTE are best satisfied using a systems-oriented solution in the form of a heterogeneous system-on-chip (SoC), including a scalable set of multicore processors and applications-specific accelerators such as pattern matching engines, security engines, packet processing engines and other features to improve overall system performance, programmability, and power efficiency. This type of solution requires that the processor cores be designed from the start to integrate well, to scale to meet performance requirements in multicore configurations, and to be power efficient. Each of these areas is explored in detail.

Efficient and Reliable Support for Heterogeneous Integration

Power Architecture supports heterogeneous system-on-chip (SoC) implementations in two important ways: (1) the hypervisor and (2) specific instruction set support for efficient interaction with hardware accelerators.

The most important advantage of embedded virtualization is to provide I/O virtualization to enable sharing and management of hardware accelerators. This is an example of the sophistication and flexibility of sharing that's enabled by the very complete hypervisor and virtualization architecture in Power ISA 2.06. Furthermore, because the hypervisor is a true hardware-supported operating mode that ensures protection of the virtual kernel from guest operating systems. Thus, the hypervisor allows different software systems to run on different cores at the same time with high integrity. This approach allows each software system and its associated private hardware resources to be protected from interference from the others. While different systems are insulated from direct interactions, software systems can establish communication mechanisms with other software systems in a controlled and reliable manner.

A hypervisor alone does not guarantee multicore efficiencies. Heterogeneous multicore processing efficiency requires additional hardware support from the processor core. For example, networking domain applications like LTE require numerous system-wide statistics to be maintained across many threads of computation. If such counters are implemented as shared memory locations, then access to them must be controlled to prevent race conditions. In software, this synchronization can lead to serious performance degradation because of the time required to obtain a semaphore before accessing and updating the statistics counter. Furthermore, many interactions with hardware accelerators are performance using accesses to memory mapped registers, which can be time consuming if the interactions must be done through a read-modify-write transaction to these registers. Power Architecture supports efficient interaction with shared statistics counters and with hardware accelerators using decorated load and store instructions. When used to implement statistics counters or in conjunction with SoC accelerators, these instructions allow efficient three-operand (address, data, command) atomic fire-and-forget transactions. In effect, these instructions replace a series of transactions between cores and memory or between cores and accelerators with highly optimized transactions.

Achieving System Throughput and Single-Thread Performance

Power Architecture has been proven to scale from embedded systems up to supercomputers in terms of power, area, and frequency. The smallest Power Architecture cores consume less than 1 watt of power while achieving highly competitive DMIPS/MHz performance. The most powerful cores can combine to produce systems with multiple petaflops of computing power. To add to this proven scalability, Power Architecture supports several special features which can combine to increase system throughput. These include special instructions and specialized feature categories designed to increase system performance.

A critical component of both system throughput and single-thread performance is memory bandwidth. Power Architecture cores employ an advanced memory bandwidth enhancement architecture that exploits weakly ordered memory semantics. Weakly ordered memory semantics allow the processor and the system to reorder memory accesses, resulting in higher performance. The vast majority of memory accesses do not have to occur in order as long they appear to be performed in order on the processor that initiates them. Therefore, the processor can issue loads earlier, reducing latency on cache misses, and perform stores at times that interfere less with loads that are required in the current instruction stream. In coherent multiprocessor operation, particularly in lock acquisition and release, ordering of certain operations is required. Power Architecture defines various memory barriers to efficiently enforce ordering. For example, the lightweight sync (*lwsync*) instruction provides an efficient ordering barrier for multiprocessor locking that can be issued between the final store of a shared data structure and the store to release the data structure's lock. The barrier does not have to be broadcast to other processors, but merely orders the accesses to the cache. (In this case, the stores are performed in order because of the barrier.) The rest of the memory accesses of the machine on each side of the barrier are free to perform their operations in the most efficient order. Many performance benefits can be derived from this approach. For example, it can be used to implement faster critical sections in operating systems kernels for symmetric multiprocessing (SMP) operating systems, thereby improving application throughput by decreasing the overhead of operating systems.

Other multicore efficiencies are provided by Power Architecture to enhance system throughput by eliminating latencies associated with interprocessor communications. Two examples are the Embedded.External Process Identifier (PID) load/store category and the Embedded.Processor Control category. The Embedded.External PID load/store category provides efficient communication between different processes and guest operating systems and a hypervisor kernel. This setup supports the need to compose multiple software systems on a single chip with maximum efficiency, providing a better ratio of communication to computation overhead between systems. The Embedded.Processor Control category (using the *msgsnd* instruction) provides lightweight context to context (thread to thread for multithreading) doorbells. These doorbells are virtualized so that software need not know physical topology and can avoid the overhead of needing to discover that topology. These are specialized system interrupts that do not require the intervention of an interrupt controller. This approach provides significant benefits to multicore systems which improves system throughput by providing a high performance out-of-band interprocessor communications mechanism.

Managing Power Constraints for the Wireless Access Market

Power Architecture cores provide important capabilities for dynamic power management. Some of these are enabled internally in the core. For example, it is common for execution units in the processor pipeline to be power-gated when idle. Furthermore, Power Architecture cores offer software-selectable power-saving modes. These power-saving modes reduce function in other areas, with some modes limiting cache and bus-snooping operations, and some modes turning off all functional units except for interrupts. These techniques are an effective way to reduce power, because they reduce switching on the chip and give operating systems a means to exercise dynamic power management.

But sometimes only the application software running on the processor has the knowledge required to decide when power can be managed without affecting performance. Recognizing this fact, Power Architecture provides application software with a means for power-optimized solutions through the *wait* instruction (Power ISA 2.06). This instruction allows software to initiate power savings when it is known that there is no work to do until the next interrupt. With this instruction, power savings can now be achieved through user-mode code. This feature is well-matched to the requirements of the LTE market segment, which requires that total SoC power be managed effectively. The combination of CPU power-savings modes, the *wait* instruction, and the ability to wake on an interrupt has been demonstrated to achieve deep sleep power savings with wake up on external event—with no packet loss.

Supporting High Quality Software

A large portion of the time required to develop products using embedded multicore processors is spent on testing and optimizing application software. Sophisticated heterogeneous SoCs require powerful debug and runtime support as well. To meet that need, Power Architecture multicore SoCs support standardized debug, performance monitors, load spreading, device virtualization, and virtualization with real applications. They have an outstanding ecosystem.

But providing tools and runtime capabilities is only part of the equation. Time-to-market is greatly affected by software development effort as well. Power Architecture's community is addressing this impact by providing vertical solutions for market segments such as the LTE segment. For example, Freescale's VortiQa solutions speed development by providing fully integrated, architecturally compatible application software that eliminates or greatly reduces the time needed for these tasks. VortiQa software is optimized to take full advantage of Freescale's heterogeneous SoC architectures, including pattern matching engines, security accelerators, data path accelerants and other features, thereby boosting performance in embedded systems. It eliminates the time-consuming task of creating application-optimal mapping of functions and core assignments for asymmetric multiprocessing (AMP), symmetric multiprocessing (SMP) and hybrid AMP+SMP configurations.

In addition to vertical solutions that enhance time-to-market, the Power Architecture hypervisor further provides a solid foundation for system software in heterogeneous SoC environments,

because it allows an existing embedded operating system to run in guest state or bare-metal state with at most minor changes. Customers can therefore quickly bring up their operating system and associated runtime libraries in a configuration that is optimized for their application, a key requirement for the LTE market segment.

3. Key Competitive Differentiation

Efficient and Reliable Supporting High Levels of Heterogeneous Integration

Power Architecture cores are the only cores that meet the requirements for complex heterogeneous SoC solutions as well as providing full hardware support for virtualization (e.g., a separate privilege level enforced by hardware, ISA extensions, and MMU support). All other cores must provide hypervisor functions through software only. Software-only approaches suffer from performance degradation. Studies have shown that virtualized process context switch time (e.g., context switching without native hardware support) can incur an average of 6.5 times more overhead. Even if the extra overhead of software-only virtualization is acceptable, software-only virtualization can be prone to undetected security loopholes and resource-sharing conflicts which ultimately lead to system deadlocks or crashes. With hardware-supported virtualization, the hardware can guarantee that a crashed virtual partition will not affect other partitions in the virtualized system.

The Power Architecture–decorated load and store instructions provide another significant advantage for networking applications that require both numerous statistics counters as well as interactions with hardware accelerators. No other processor cores capable of meeting the power, performance, and integration demands of LTE provide such a sophisticated solution to minimizing these time-consuming transactions to speed up application performance. Furthermore, reducing the number of transactions between cores, memory, and hardware accelerators translates directly into power efficiency, which is a large benefit for LTE infrastructure providers.

Achieving System Throughput and Single-Thread Performance

Competitors in the LTE market employ cores that do not approach the proven breadth of scale that Power Architecture provides for both single-thread performance and total system throughput. For example, although some competition offers core frequencies up to 1.6 GHz, Power Architecture cores have been shown to scale well beyond this limit. Furthermore, SoCs based on Power Architecture cores have demonstrated eight core configurations that consume less than 35 watts of power while achieving world-class throughput for networking applications. At the same time, Power Architecture cores have established many records for performance in the High Performance Computing (HPC) community. This unmatched breadth of scalability allows appropriate tradeoffs to be made between the computing power required to support the ever increasing complexity of processing a single packet with the need for optimal system level throughput and power efficiency. For simpler protocols systems that comprise Power Architecture, cores can be configured using a higher number of simpler cores for enhanced system throughput.

Latency to memory is a key component of performance. For this reason, Power Architecture exploits weakly ordered memory semantics for system throughput as well as single-thread performance. Other embedded processors do not provide such advanced techniques. For example, ARM cores support only strict memory ordering and furthermore only support coherency for clusters of no more than four cores using a shared L2 cache. Performance of ARM cores suffers significantly when locks must be acquired, especially in SoC configurations with more than four cores. Although the MIPS instruction set does provide a less advanced form of weak memory ordering with an associated sync instruction, the Power Architecture *lwsync* instruction is a more advanced feature with significant performance advantages.

Interprocessor communications is another critical element of performance in multicore systems. The Power Architecture Embedded.External PID load/store category provides efficient communication between different processes as well as between guest operating systems and a hypervisor kernel. ARM and MIPS architectures do not provide the same level of performance acceleration for interprocessor communications. Furthermore, nothing comparable to the Power Architecture Embedded.Processor Control category (which provides accelerated interprocessor communications with the *msgsnd* instruction) is provided by most competitors. And where similar capability is provided, it is not provided with other associated benefits such as weakly ordered memory access semantics. Thus they cannot realize the full benefits provided by Power Architecture.

Managing Power Constraints

The strong alliance of Power.Org partners ensure that Power Architecture cores will continue to be available on leading edge process technologies—a future that competitors will struggle to guarantee. Today, Power Architecture cores are being built in 45-nanometer process technology. This structure allows eight high-performance cores on an SoC with large degrees of integration with hardware accelerators in under 30 watts. Four core versions can be approximately 30% lower in power. In the future, Power Architecture cores will continue to aggressively migrate to new technology nodes.

Power Architecture allows for even better power-optimized solutions in a cost-sensitive market using the *wait* instruction (Power ISA 2.06). This instruction allows software to initiate power savings when there is no work to do until the next interrupt. The combination of this capability with other important power-saving features positions Power Architecture to provide rich power management features.

Providing High Quality Tools, Systems Software, and Networking Stacks

Software developers for Power Architecture cores have long benefited from a vibrant support ecosystem, including high-quality tools, operating systems, and network protocol stacks. Power.org is continuously working with ecosystem partners to ensure that this remains the case for multicore Power Architecture products. For example, SoCs incorporating Power Architecture cores now provide sophisticated mechanisms such as system-level hardware queue managers.

These queue managers interact with the cores and the rest of the SoC to allow for easier multicore programming as well as to provide system-level performance benefits. This mechanism reduces the complexities of programming, debugging, load balancing, and optimizing software. Such an approach bypasses the headaches of creating software for cores that require programming and debugging multiple hardware threads per core, a common technique proposed on competitors' roadmaps. This unique combination of mature and reliable software ecosystem with efficient multicore programmability sets Power Architecture apart from its competitors.

4. Overall Value Proposition and Conclusion

Power Architecture cores are suitable for high levels of heterogeneous integration using multiple cores and various hardware accelerators. The family is composed of a wide range of cores and multicore configurations providing cost-optimized solutions that meet performance and power goals: Power Architecture cores have demonstrated proven implementations from the smallest devices to the supercomputers. Furthermore, Power Architecture incorporates hypervisor for sharing and management of system resources, I/O virtualization, and support for a broad set of system software configurations. System throughput is a key metric, but single-thread performance is also important. Power Architecture has held world records in performance and innovations for 18 years. Power efficiency and reliability have been proven as well: Power Architecture cores were used on all space missions to Mars, executing in extremely harsh environmental conditions with severe power constraints. Furthermore, Power Architecture fulfills the requirements for high-quality tools, systems software, and stacks, providing a unified programming model across a wide range of cores. These advantages result from the fact that Power Architecture is open and allows choice of vendors and suppliers.

References

“LTE—an introduction.”, Ericsson,
http://www.ericsson.com/technology/whitepapers/lte_overview.pdf

”LTE Base Station Spending to Hit \$3.3 Billion in 2011”, C114,
<http://www.cn-c114.net/583/a426332.html>

“LTE – Long Term Evolution”, Alcatel-Lucent,
<http://www1.alcatel-lucent.com/technology/lte/>

Barham, et. Al, “**Xen and the Art of Virtualization**”, *SOSP'03*, October 19, 2003, Bolton Landing, New York, USA.