*International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)*
*Volume-3 Number-3 Issue-11 September-2013*

# Review on High Performance Energy Efficient Multicore Embedded Computing

**Archana S. Shinde[1], Swapnali B. Bhosale[2], Ashok R. Suryawanshi[3]**

## Abstract

***With Moore's law supplying billions of transistors on chip, embedded systems are undergoing a transition from single core to multicore to exploit this high transistor density for high performance. The main objective of developing a High Performance Energy Efficient Multicore Embedded Computing is to satisfy both Performance and the Power which is the first order constraint for Embedded Systems. To achieve this objective, this paper outlines typical requirements of embedded applications and provides state of the art hardware/software high performance energy efficient embedded computing (HPEEC) techniques that help meeting these requirements. Finally, this paper presents design challenges and future research directions for HPEEC system development.This paper is a literature review of a High Performance Energy Efficient Multicore Embedded Computing. The techniques reported by the researchers till date is encouraging and motivate further research in this domain.***

## Keywords

***High performance computing (HPC), multicore, energy efficient computing, low power embedded systems.***

## 1. Introduction

Embedded system design is traditionally power centric but there has been a recent shift toward high performance embedded computing (HPEC) due to the proliferation of computeintensive embedded applications.The design challenges are competing because high performance typically requires maximum processor speeds with enormous energy consumption, whereas low power typically requires nominal or low processor speeds that offer modest performance.HPEEC requires thorough consideration of the thermal design power (TDP) and processor frequency relationship while selecting an appropriate processor for an embedded application.

**A. S. Shinde,** PCCOE, Pune University.
**S. B. Bhosale,** PCCOE, Pune University.
**A. R. Suryawanshi,** PCCOE, Pune University.

The distinction between HPC for supercomputers and HPEEC is important because performance is the most significant metric for supercomputers with less emphasis given to energy efficiency whereas energy efficiency is a primary concern for HPEEC.
In this paper, it focuses on high performance and energy efficient techniques that are applicable to embedded systemsto meet particular application requirements. Many of the HPEEC techniques at different levels are complementary in nature and work in conjunction with one another to better meet application requirements.

**Background and relevance**
The trend of increasing a processors speed to get a boost in performance is a way of the past. Multicore processors are the new direction manufacturers are focusing on. Using multiple cores on a single chip is advantageous in raw processing power, but nothing comes for free.

With additional cores, power consumption and heat dissipation become a concern and must be simulated before layout to determine the best floorplan which distributes heat across the chip, while being careful not to form any hot spots.

**Organization of the paper work**
Chapter 2 covers the multicore basics. Chapter 3 covers the proposed techniques under different approaches for achieving High Performance Energy Efficient Multicore Embedded Computing. The proposed techniques are divided under 3 approaches. Under each approach, different techniques are suggested for different embedded application to achieve High Performance with energy efficiency. High Performance Energy Efficient Multicore Processors and discussions are discussed in Chapter 4. Chapter 5 includes embedded applications. Chapter 6 includes conclusion and Chapter 7 consists of future scope of this paper work.

## 2. Multicore Basics

**Brief History of Microprocessors**
Intel manufactured the first microprocessor, the 4bit 4004, in the early 1970s which was basically just a numbercrunching machine. Fig. 1 shows the world's

first single chip processor. Shortly afterwards they developed the 8008 and 8080, both 8 bit. The companies then fabricated 16 bit microprocessors and Intel the 8086 and 8088. The former would be the basis for Intel's 80386 32 bit and later their popular Pentium line up which were in the first consumer based PCs. Each generation of processors grew smaller, faster, dissipated more heat and consumed more power.

**Figure 1: The world's first single chip processor**

**Moore's Law**

One of the guiding principles of computer architecture is known as Moore's Law. In 1965 Gordon Moore stated that the number of transistors on a chip will roughly double each year (he later redefined this, in 1975, to every two years). What is often quoted as Moore's Law is Dave House's revision that computer performances will double every 18 months.

As shown in Figure 2, the number of transistors has roughly doubled every 2 years. Multicore processors are often run at slower frequencies, but have much better performance than a single core processor because "Two heads are better than one." [5].

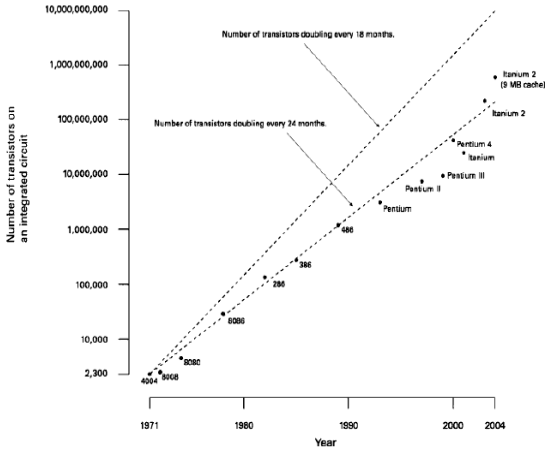


**Figure 2: Depiction of Moore's Law**

**Multicore**

A multicore architecture is an architecture where on a single computer chip multiple processors are integrated. A multicore processor has two or more independent cores.Each of the processor is referred to as a core. The core is part of the processor that is responsible for correctly reading and executing the instructions. Fig.3 shows (a) Shared Memory Model, (b) Distributed Memory Model. If we set two cores side by side, one can see that a method of communication between the cores and to main memory is necessary. This is usually accomplished either using a single communication bus or an interconnection network. The bus approach is used with a shared memory model whereas the inter connection network approach is used with a distributed memory model. After approximately 32 cores the bus is overloaded with the amount of processing, communication, and competition, which leads to diminished performance therefore a communication bus, has a limited scalability.

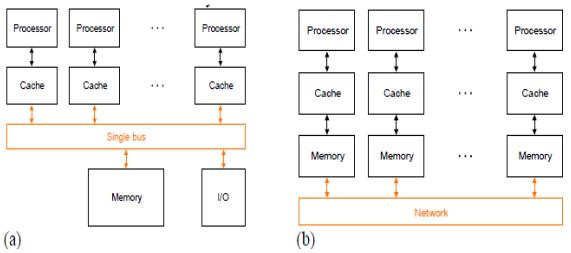


**Figure 3: (a) Shared Memory Model, (b) Distributed Memory Model**

Table 1 below shows a comparison of a single and multicore (8 cores in this case) processor used by the Packaging Research Center at Georgia Tech [5]. With the same source voltage and multiple cores run at a lower frequency we see an almost tenfold increase in bandwidth while the total power consumption is reduced by a factor of four.

**Table 1:Single Core vs. Multicore [5]**

| Parameters | Single Core Processor (45nm) | Multicore processor (45nm) |
|---|---|---|
| Vdd | 1.0V | 1.0V |
| I/O pins total | 1280 (ITRS) | 3000 (ITRS) |
| Operating Frequency | 7.8 GHz | 4 GHz |
| Chip Package Data Rate | 7.8 Gb/s | 4 Gb/s |
| Bandwidth | 125 Gbyte/s | 1 TeraByte/s |
| Power | 429.78W | 107.39W |
| Total no of pins on chip | 3840 | 9000 (Estimated) |
| Total no of pins on the package | 2480 | 4500 (Estimated) |

**The Need for Multicore**

The single core processors are capable of executing only one instruction at a time.As we move from one generation to the other i.e. from 8086,80186,80286,80386 towards Pentium IV we see that the processor clock frequency increases.The faster clock speeds typically require additional transistors and higher input voltages, resulting in greater power consumption. The increasing clock speed is creating a power dissipation problem for semiconductor manufacturers.

The latest semiconductor technologies support more and more transistors. The downside is that every transistor leaks a small amount of current, the sum of which is problematic.Instead of pushing chips to run faster, CPU designers are adding resources, such as more cores i.e. multicore and more cache to provide comparable or better performance at lower power.Adding a core will double the system performance, and dissipate less heat. Practically the speed of single core processor is faster than every core on the multicore system.

## 3. Proposed Techniques

To meet embedded application requirements different approaches are defined in which different techniques are proposed. In HPEEC different approaches used are Architectural Approaches, Middle Approaches and Software approaches. Fig. 5 gives an overview of the HPEEC domain, which spans architectural approaches to middleware and software approaches.

### A. Architectural Approaches

HPEEC architectural approaches play an important role in meeting varying application requirements. These architectural approaches can be broadly categorized into four categories: core layout, memory design, interconnection networksand reduction techniques. In this section, HPEEC architectural approaches are discussed.

#### 1. Core Layout

There exist various core layout considerations during chip and processor design such as whether to use homogeneous (cores of the same type) or heterogeneous cores (cores of varying types), whether to position the cores in a 2D or 3D layout on the chip, whether to design independent processor cores with switches that can turn on/off processor cores or to have a reconfigurable integrated circuit that can be configured to form processor cores of different granularity.

In this section, a few core layout techniques includes heterogeneous CMP, conjoined core CMP, tiled multicore architectures, 3D multicore architectures, composable multicore architectures, multicomponent architectures and stochastic processors.

#### 2. Memory Design

The cache miss rate, fetch latency, and data transfer bandwidth are some of the main factors impacting the performance and energy consumption of embedded systems [8]. The memory subsystem encompasses the main memory and cache hierarchy and must take into consideration issues such as consistency, sharing, contention, size and power dissipation. In this section, HPEEC memory design techniques which include transactional memory, cache partitioning, cooperative cachingandsmart caching.

#### 3. Interconnection Network

As the number of onchip cores increases, a scalable and highbandwidth interconnection network to connect on chip resources becomes crucial. Interconnection networks can be static or dynamic. Static interconnection networks consist of pointtopoint communication links between computing nodes and are also referred to as direct networks (e.g., bus, ring, hypercube). Dynamic interconnection networks consist of switches and links and are also referred to as indirect networks (e.g., packet switched networks) [1]. This section consists of prominent interconnect topologies (e.g., bus, 2D mesh, hypercube) and interconnect technologies (e.g., packet switched, photonic, wireless).

#### 4. Reduction Techniques

Due to an embedded system's constrained resources, embedded system architectural design must consider power dissipation reduction techniques. Power reduction techniques can be applied at various design levels: the complementary metal oxide semiconductor (CMOS) level targets leakage and short circuit current reduction. The processor level targets instruction/data supply energy reduction as well as power efficient management of other processor components (e.g., execution units, reorder buffers etc.) and the interconnection networklevel targets minimizing interconnection length using an appropriate network layout. In this section, several power reduction techniques include leakage current reduction, short circuit current reduction, peak power reductionand interconnection length reduction.

### B. Hardware Assisted Middleware Approaches

Various HPEEC techniques are implemented as middleware and/or part of an embedded OS to meet application requirements. The HPEEC middleware techniques are implemented in hardware to provide required functionalities.

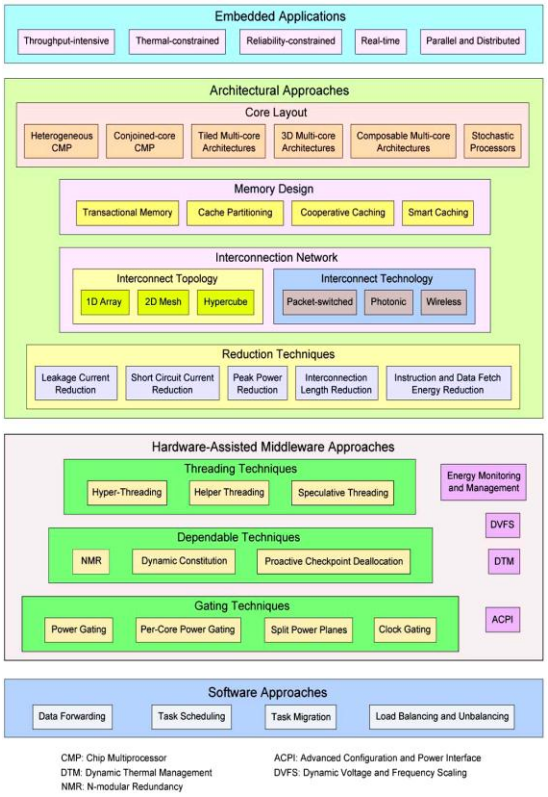


**Figure 4: High Performance Energy Efficient Embedded Computing Domain**

HPEEC hardware assisted middleware techniques includes dynamic voltage and frequency scaling (DVFS), advanced configuration and power interface (ACPI), threading techniques (hyper threading, helper threadingand speculative threading), energy monitoringand management, dynamic thermal management, dependable HPEEC (DHPEEC) techniques (N-modular redundancy (NMR), dynamic constitutionand proactive checkpoint deallocation) and various low power gating techniques (power gating, per core power gating, split power planes, and clock gating) [2].

### C. Software Approaches

The performance and power efficiency of an embedded platform not only depends upon the built in hardware techniques but also depends upon the software's ability to effectively leverage the hardware support. Software based HPEEC techniques assistDPM by signallingthe hardware of the resource requirements of an application phase. Software approaches enable high performance by scheduling and migrating tasks statically or dynamically to meet application requirements. HPEEC softwarebased techniques include data forwarding, task scheduling, task migration and load balancing.The workload aware load unbalancing strategy reduces themean waiting time of aperiodic tasks by 92 percent with similar power efficiency as compared to a workloadunaware load unbalancing strategy [3].

## 4. High Performance Energy Efficient Multicore Processors and Discussions

Silicon and chip vendors have developed various high performance multicore processors that leverage the various HPEEC techniques discussed in this paper. In this, we discuss some prominent multicore processors (summarized in Table 2) and focus on their HPEEC features.

**Table 2: HPEEC Multicore Processors**

| Processor | Cores | Speed | Power | Performance |
|---|---|---|---|---|
| ARM11 MPCore | 1 - 4 | 620 MHz | 600 mW | 2600 DMIPS |
| ARM Cortex A-9 MPCore | 1 - 4 | 800 MHz - 2 GHz | 250 mW per CPU | 4000-10,000 DMIPS |
| MPC8572E PowerQUICC III | 2 | 1.2 GHz - 1.5 GHz | 17.3 W @ 1.5 GHz | 6897 MIPS @ 1.5 GHz |
| Tilera TILEPro64 | 64 tiles | 700 MHz - 866 MHz | 19 - 23 W @ 700MHz | 443 GOPS |
| Tilera TILE-Gx | 16/36/64/100 tiles | 1 GHz - 1.5 GHz | 10-55 W | 750 GOPS |
| Intel Sandy Bridge | 4 | 3.8 GHz | 35-45 W | 121.6 GFLOPS |
| NVIDIA GeForce GTX 460 | 336 CUDA cores | 1.3 GHz | 160 W | 748.8 GFLOPS |
| NVIDIA GeForce 9800 GX2 | 256 CUDA cores | 1.5 GHz | 197 W | 1152 GFLOPS |
| NVIDIA GeForce GTX 295 | 480 CUDA cores | 1.242 GHz | 289 W | 748.8 GFLOPS |

| | | | | |
|---|---|---|---|---|
| NVIDIA Tesla C2050/C2070 | 448 CUDA cores | 1.15 GHz | 238 W | 1.03 TFLOPS |
| AMD FireStream 9270 | 800 stream cores | 750 MHz | 160 W | 1.2 TFLOPS |
| ATI Radeon HD 4870 X2 | 1600 stream cores | 750 MHz | 423 W | 2.4 TFLOPS |

### A. Tilera TILEPro64 and TILE-Gx

Tilera revolutionizes high performance multicore embedded computing by leveraging a tiled multicore architecture (e.g., the TILEPro64 and TILE-Gx processor family [9], [10]). The TILEPro64 and TILE-Gx processors offer 5.6 and 32 MB of on chip cache respectively and implement Tilera's dynamic distributed cache (DDC) technology that provides a 2x improvement on average in cache coherence performance over traditional cache technologies using a cache coherence protocol.

### B. Intel Xeon Processor

Intel leverages Hafnium Hi-K andmetal gates in next generation Xeon processors to achievehigher clock speeds and better performance per watt. TheXeon processors also implement hyper threading and widedynamic execution technologies for high performance. Thewider execution pipelines enable each core to simultaneously fetch, dispatch, execute, and retire up to four instructions per cycle [11]. Intel's deep power down technology enables both cores and the L2 cache to be powered down when the processor is idle [12].

### C. Graphics Processing Units

GPUs feature highmemorybandwidth that is typically 10xfaster thancontemporary CPUs. NVIDIA and AMD/ATI are thetwomain GPU vendors. NVIDIA's PowerMizer technology available on all NVIDIA GPUs is a DPM technique that adapts the GPUto suit an application's requirements [13].

## 5. Embedded Applications

Different embedded applications have different characteristics. We discuss below some of these application characteristics in context of their associated embedded domains.

### 1. Throughput Intensive

Throughput intensive embedded applications are applications that require high processing throughput. Networking and multimedia applications, which constitute a large fraction of embedded applications, are typically throughput intensive due to ever increasing quality of service (QoS) demands. An embedded system containing an embedded processor requires a network stack and network protocols to connect with other devices. A telemedicine application requires processing of 5 million blocks per second [6].

### 2. Thermal Constrained

An embedded application is thermalconstrained if an increase in temperature above a threshold could lead to incorrect results or even the embedded system failure. Depending on the target market, embedded applications typically operate above $45^{O}$C (e.g., telecommunication embedded equipment temperature exceeds $55^{O}$C) in contrast to traditional computer systems, which normally operate below $38^{O}$C.

### 3. Reliability Constrained

Embedded systems with high reliability constraints are typically required to operate for many years without errors and/or must recover from errors since many reliabilityconstrained embedded systems are deployed in harsh environments where post deployment removal and maintenance is infeasible. Hence, hardware and software for reliabilityconstrained embedded systems must be developed and tested more carefully than traditional computer systems.

### 4. Real Time

In addition to correct functional operation, realtime embedded applications have additional stringent timing constraints, which impose realtime operational deadlines on the embedded system's response time. Although realtime operation does not strictly imply high performance, realtime embedded systems require high performance only to the point that the deadline is met, at which time high performance is no longer needed. Hence, realtime embedded systems require predictable highperformance. Realtime operating systems (RTOSs) provide guarantees for meeting the stringent deadline requirements for embedded applications [7].

### 5. Parallel and Distributed

Parallel and distributed embedded applications leverage distributed embedded devices to cooperate and aggregate their functionalities or resources. Wireless sensor network (WSN) applications use sensor nodes to gather sensed information (statistics and data) and use distributed fault detection algorithms. Mobile agent (autonomous software agent)based distributed embedded applications allow the process state to be saved and transported to

another new embedded system where the process resumes execution from the suspended point. Many embedded applications exhibit varying degrees (low to high levels) of parallelism, such as instruction level parallelism (ILP) and threadlevel parallelism (TLP). Innovative architectural and software HPEEC techniques are required to exploit an embedded applications available parallelism to achieve high performance with lowpower consumption.

## 6. Conclusion

Multicore chips are an important new trend in computer architecture. Several new multicore chips in design phases. In this paper, HPEEC techniques are used in multicore processors to achieve its objective. HPEEC is an active and expanding research domain with applications ranging from consumer electronics to supercomputers. The introduction of HPEEC techniques satisfies both high performance and energy efficiency in multicore embedded computing.

**Future Scope**

Although power is a firstorder constraint in HPEEC platforms, several additional challenges facing the HPEEC domain are as below:

1. **Complex Design Space**

Large design space due to various core types and each cores tunable parameters (e.g., instructionwindow size, issue width, fetch gating)

**2. High On Chip Bandwidth**

Increased communication due to increasing number of cores requires high bandwidth on chip interconnects.

**3. Synchronization**

Synchronization primitives (e.g., lock, barriers) result in programs serialization degrading performance.

**4. Shared Memory Bottleneck**

Threads running on different cores make large number of accesses to various shared memory data partitions.

**5. Cache Coherence**

Heterogeneous cores with different cache line sizes require cache coherence protocols redesign and synchronization primitives (e.g., semaphore, locks) increase cache coherence traffic [4].

**6. Cache Thrashing**

Threads working concurrently evict each other's data out of the shared cache to bring their own data.

## References

[1] Shacham, K. Bergman, and L. Carloni, "Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors," IEEE Trans. Computers, vol. 57, no. 9, pp. 1246-1260, Sept. 2008.

[2] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. Cameron, "PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications," IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 5, pp. 658-671, May 2010.

[3] H. Jeon, W. Lee, and S. Chung, "Load Unbalancing Strategy for Multi-Core Embedded Processors," IEEE Trans. Computers, vol. 59, no. 10, pp. 1434-1440, Oct. 2010.

[4] T. Berg, "Maintaining I/O Data Coherence in Embedded Multicore Systems," IEEE MICRO, vol. 29, no. 3, pp. 10-19, May/June 2009.

[5] Bryan Schauer "Multicore Processors – A Necessity", ProQuest Discovery Guides, Sept. 2008.

[6] G. Kornaros, Multi-core Embedded Systems. Taylor and Francis Group, CRC Press, 2010.

[7] Dr. K. V. K. K. Prasad, Embedded Real Time Systems : Concepts, Design & Programming Black Book.

[8] Frank Vahid , Tony Givargis : Embedded System Design.

[9] TILERA, "Manycore without Boundaries: TILEPro64 Processor", http://www.tilera.com/products/processrs/TILEPRO64, June 2011.

[10] TILERA, "Manycore without Boundaries: TILE-Gx ProcessorFamily",http://www.tilera.com/products/processors/TILE-Gx_Family, June 2011.

[11] Intel, "High-Performance Energy-Efficient Processors for Embedded Market Segments", http://www.intel.com/design/embedded/downloads/315336.pdf, June 2011.

[12] Intel, "Intel Core 2 Duo Processor Maximizing Dual-Core Performance Efficiency" ftp://download.intel.com/products/processor/core2duo/mobile_prod_brief.pdf, June 2011.

[13] NVIDIA, "NVIDIA PowerMizer Technology," http://www.nvidia.com/object/feature_powermizer.html, June 2011.

**Archana S. Shinde** received her B.E. degree from CWIT, University of Pune in 2012. She is currently pursuing M.E in VLSI and Embedded Systems from P.C.C.O.E, Pune University.

**Swapnali B. Bhosale** received her B.E. degree from Dr. BAMU, Aurangabad in 2012. She is currently pursuing M.E in VLSI and Embedded Systems from P.C.C.O.E, Pune University.

**Prof. Ashok Suryawanshi** is an assistant professor of electronics and telecommunication engineering at Pimpri Chinchwad College of Engineering, University of Pune, India. He received his B.E. degree from Shivaji University and M.E. degree from M.S. University. His areas of interests include Electronic Devices & Networks, Power Electronics /Industrial Electronics.