Development of Low Power Many-Core SoC for Multimedia Applications

Takashi Miyamori, Hui Xu, Takeshi Kodaka, Hiroyuki Usui, Toru Sano, Jun Tanabe
Center for Semiconductor Research & Development
Semiconductor & Storage Products Company, Toshiba Corporation
Kawasaki, Japan

Abstract— **New media processing applications such as image recognition and AR (Augment Reality) have become into practical on embedded systems for automotive, digital-consumer and mobile products. Many-core processors have been proposed to realize much higher performance than multi-core processors. We have developed a low-power many-core SoC for multimedia applications in 40nm CMOS technology. Within a 210mm2 die, two 32-core clusters are integrated with dynamically reconfigurable processors, hardware accelerators, 2-channel DDR3 I/Fs, and other peripherals. Processor cores in the cluster share a 2MB L2 cache connected through a tree-based Network-on-Chip (NoC). Its total peak performance exceeds 1.5TOPS (Tera Operations Per Second). The high scalability and low power consumption are accomplished by parallelized firmware for multimedia applications. It operates the 1080p 30fps H.264 decoding about 400mW and the 4K2K 15fps super resolution under 800mW.**

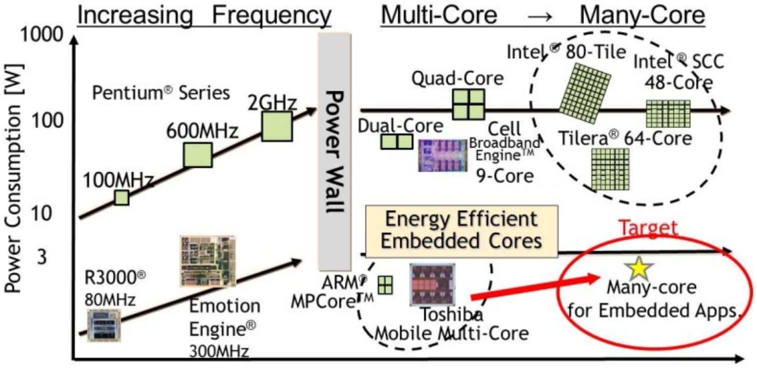
Keywords—Many-core; Network-on-Chip; VLIW; Low power ; Power gating; H.264; Super resolution

I. INTRODUCTION

New media processing applications such as image recognition and AR (Augment Reality) have become into practical on embedded systems for automotive, digital-consumer and mobile products. To achieve high performance with low power consumption, multi-core architecture is now widely used. Furthermore, as shown in Fig.1, many-core processors have been proposed for high performance computing (HPC) domain by using high-performance cores [1-2]. However, multimedia embedded applications still confront the challenge of achieving high computing power, more than hundreds of GOPS within a few watts.

We have developed a low power and highly integrated many-core SoC with two 32-core clusters. To meet requirements of embedded applications, the many-core cluster utilized low-power embedded processor cores. The power consumption of a many-core cluster is less than 1W, while the other published many-cores consumer several tens to more than one hundred watts [1]. Each cluster consists of 32 cores and an L2 cache, and they are connected by a tree-based Network-on-Chip (NoC). This many-core architecture based on a shared-memory model keeps software compatibility with our previous multi-core architecture [3].

In the next section, the hardware architecture of the many-core SoC and its chip implementation are described. Then, its parallel processing scheme on software is explained. Finally, we show the evaluation results of the many-core SoC and conclude this paper.



Intel and Pentium are registered trademarks of Intel Corporation in the U.S. and/or other countries.
R3000 is registered trademarks of. MIPS Technologies, Inc.
Emotion Engine is a registered trademark of Sony Computer Entertainment, Inc.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc.
ARM is a registered trademark of Tilera Corporation.
MPCore is a trademark of ARM Limited.
Tilera is a registered trademark of Tilera Corporation.

Figure 1.Trends of many-core processors for embedded applications.

II. ARCHITECTURE OF THE MANY-CORE SOC

A. *Overview of Chip Architecture*

Figure.2 shows the block diagram of the many-core SoC. Unlike the other previous many-core LSIs [1-2] that consist of only an array of processor cores and minimum I/Fs, our many-core SoC is highly integrated with two 32-core clusters, two ARM® Cortex®-A9 cores, two reconfigurable processors, hardware accelerators and other peripherals. The reconfigurable processors [5] are specially designed for bitstream processing such as CABAC and CAVLD of video codecs. As the hardware accelerators, it includes block matching, histogram, affine transformation, and filter processing accelerators that are frequently used for image recognition algorithm [6]. Its total peak performance exceeds 1.5TOPS (Tera Operations Per Second). The two-channel DDR3 I/Fs can provide peak memory bandwidth of 10.7GB/s. Moreover, miscellaneous I/Fs, such as two PCIe, four video inputs, and one video output are also integrated. The interruption controller and the 512KB SRAM on the cluster bus are utilized to facilitate the communication between two clusters.

ARM and Cortex are registered trademarks of ARM Limited.

978-3-9815370-0-0/DATE13/©2013 EDAA

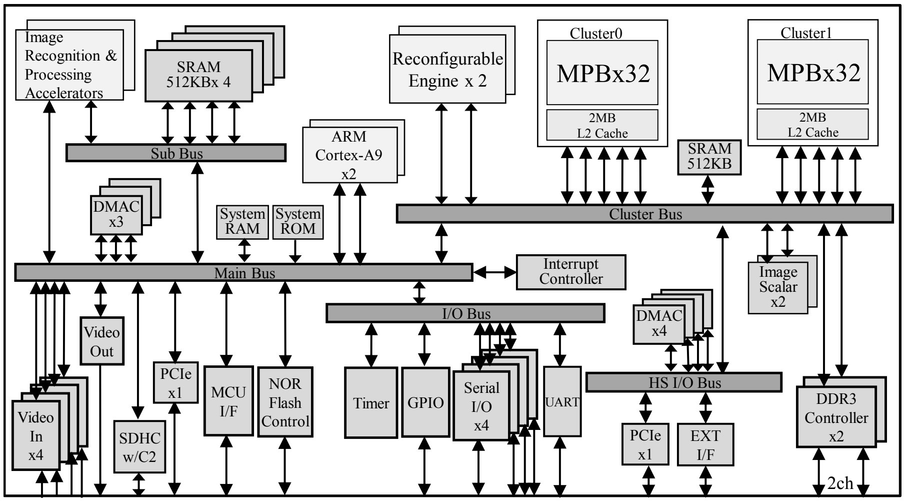


Figure 2. Block diagram of many-core SoC

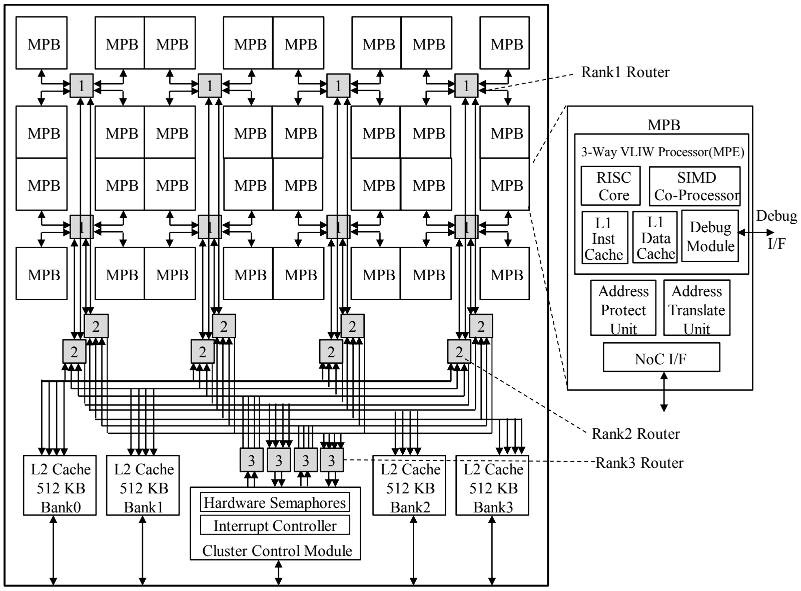


Figure 3. Block diagram of many-core cluster and Media Processing Engine Block (MPB)

B. *32-Core Cluster Architecture*

The architecture of the 32-core cluster is presented in Fig.3. The processor core in the cluster is based on a 3-way VLIW processor called Media Processing Engine Block (MPB). It is organized by a 5-stage pipelined 32b RISC core with a 64b SIMD coprocessor and executes maximum 40 8b operations simultaneously. Each MPB has a 32KB instruction and a 16KB L1 data caches. From our previous multi-core processor [3-4], the processor core has been extended with an address translate unit, an address protect unit and a Network-on-Chip (NoC) interface to adapt the many-core system.

Furthermore, the many-core processor maintains software compatibility with the previous multi-core processor [3-4]. The software model is based on a shared memory model. The L2 cache is shared among all MPBs in the cluster keeping its cache coherency by software. To increase the bandwidth, the 2MB L2 cache are interleaved and divided into four 512KB banks. The interleave granularity can be changed to 1, 4, 8, 16 lines of the L2 cache depending on memory accessing patterns of applications. The associativity and line size of the L2 cache are 32-way and 256B respectively. Atomic accesses such as Test & Set access are implemented and it can ensure access ordering of the L2 cache and the global memory.

During the architecture design phase, we investigated cross-bar interconnection and Network-on-Chip implementations with mesh, torus, and tree topologies to find an appropriate interconnection. The simulation results of multimedia applications showed that the tree topology, by connecting MPBs on leaf sides and the L2 cache banks on root

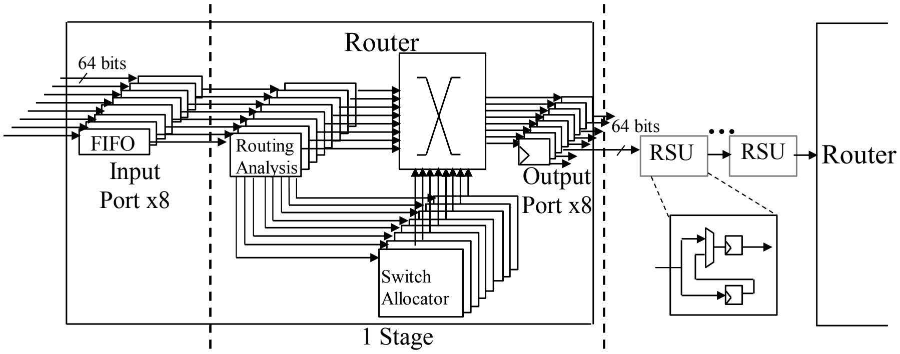


Figure 4. Architecture of single stage router.

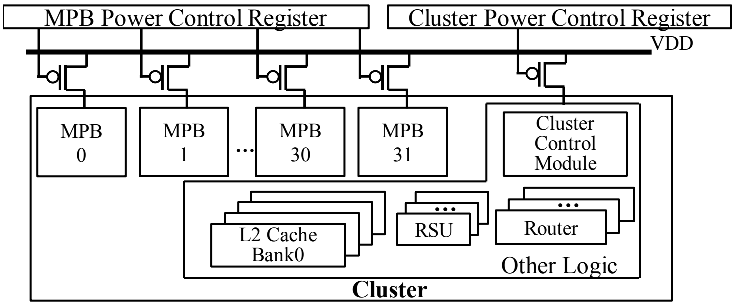


Figure 5. Structure of power gating in many-core cluster.

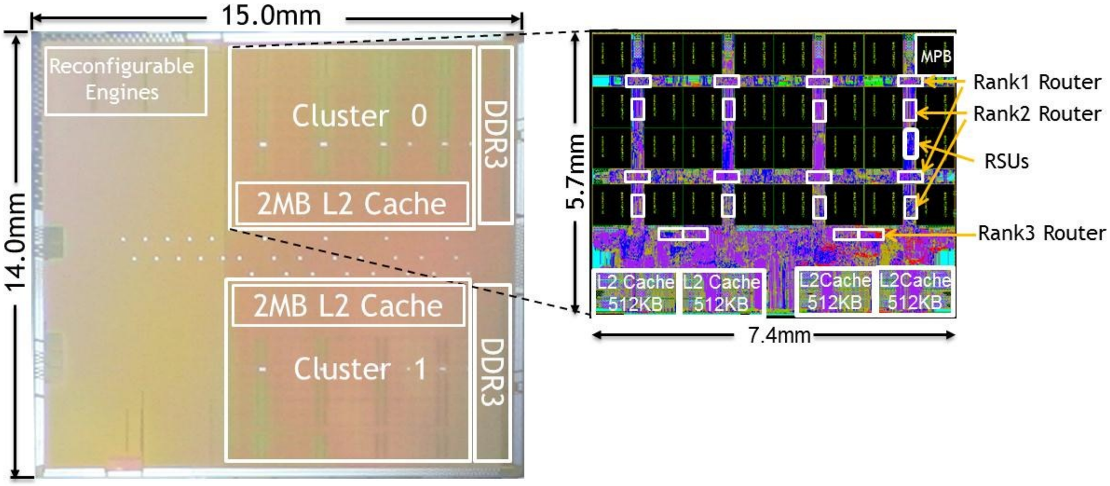


Figure 6. Micrograph of many-core SoC and layout of many-core cluster.

sides, achieved power and area advantages than others. In case of our many-core software model, data in the L2 cache are shared with MPBs. The tree topology is suited for this model, because the accesses from MPBs converge on the banks of the L2 cache. Moreover, the tree topology has the same hop counts and latencies from each MPB to the L2cache, this is appropriate for scalability of the many-core architecture.

We classify the routers into Rank1, Rank2, and Rank3 from leaf to root as shown in Fig.3. A Rank1 router is linked to four MPBs and two Rank2 routers. Each Rank2 router is linked to two Rank1 routers and two L2 cache banks. This tree-based NoC can provide 42.6GB/s bandwidth for the L2 cache. A Rank3 router is linked to the cluster control module which includes the hardware semaphores and the interrupt controller and supports the synchronization of MPBs.

We designed the router with source routing and a single bit on/off flow control to minimize its latency. As shown in Fig.4, the 8-port router operates at one cycle latency. Router Stage Units (RSUs) are inserted between the routers considering the physical length of links.

C. *Low Power Technologies*

We introduced the following three low power technologies for many-core cluster:

- **Clock gating:** Both fine-grained and coarse-grained clock-gating are applied to reduce dynamic power. As coarse-grained clock-gating, each MPB can be clock-gated by software. Moreover each Rank1 router can be automatic clock-gated when the four connected MPBs are clock-gated.
- **Power gating:** To reduce leakage power, the supply power can be gated on a per-core basis. The structure of the power switches are described in Fig.5. Every MPB has its own power switch and the other logic in the cluster shares one power switch. To turn off the whole cluster, all of the power switches of the 32 MPBs and the other logics should be turned off. This separate power switch mechanism contributes to reducing the capacitance of the power line and IR drop.
- **Low Power F/F:** The low power F/Fs (DM-F/Fs [3]) are fully utilized in MPBs by expecting about 30% power reduction than normal F/Fs.

D. *Chip Imlementation*

The micrograph of the many-core SoC is shown in Fig.6. The chip has been fabricated with 40nm CMOS process and integrates 87.5M transistors on 15.0mm x 14.0mm die. The size of the 32-core cluster is 7.4mm x 5.7mm. The maximum operating frequency of the cluster is 333MHz at 1.1V.

TABLE I. CHIP SPECIFICATIONS

Technology	40nm LP Process
Interconnect	8 metal (Cu)
Chip Size	15.0mm x 14.0mm
Transistors	87.5 M Trs.
32-Core Cluster Size	7.4mm x 5.7mm
Cluster Frequency	333MHz, 1.1V

Figure 6 also indicates the floor plan of the many-core cluster. Each MPB occupies less than 0.7mm^2. The floor plan corresponds to the logical NoC topologies as shown in Fig.3. The rank1 routers are placed on the corner of the four MPBs. RSUs are inserted to the long links between the Rank2 routers and the L2 cache banks.

III. SOFTWARE PARALLEL PROCESSING SCHEME

We use the same thread-based parallel execution model with our previous multi-core architecture [3-4] to maintain software compatibility. The execution model has been already designed to achieve the performance scalability and the transparency of the number of processor cores. It performs effectively on the many-core as well.

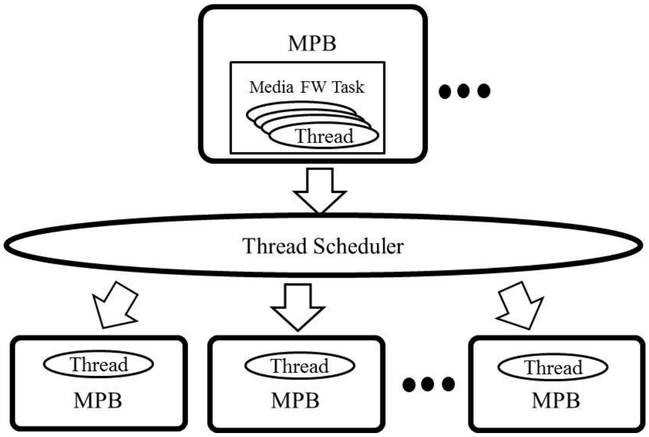


Figure 7. Thread-based parallel execution model.

For the performance scalability, we introduced the thread model that is much lighter than other thread model such as pthread. The thread-based parallel processing is shown in Fig.7. An application program is divided into small threads that are as small as functions in C language. Each thread is assigned to one core by the thread scheduler on the fly. As the number of threads increase, it increases the opportunity to exploit more parallelism and improve the performance. However, it increases overheads of thread scheduling at the same time. To reduce scheduling overhead, the thread model has the following properties:

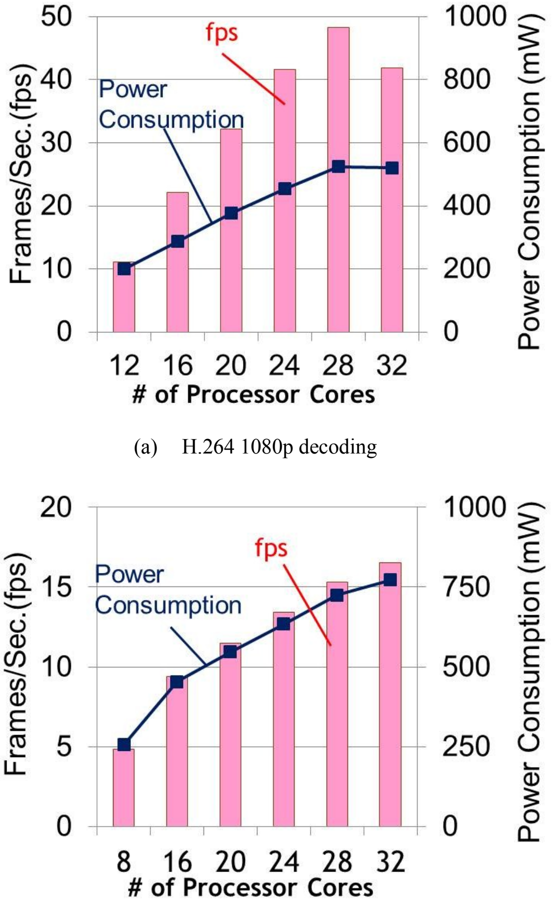
- A thread cannot suspend during its execution
- A thread does not start until all input data are ready

With these properties, there are no overheads due to context switch and synchronization of threads during thread executions.

IV. EVALUATION RESULTS

We evaluate performance and power consumption of the many-core by running realistic applications. Figure. 8 shows the performance of the many-core cluster when it executes the H.264 1080p decoding and the super-resolution from 1920x1080 to 4K2K. Thanks to the software compatibility, we can use the same source program that was developed for our previous multi-core without modifications. For the many-core, we only need to change thread sizes that should be optimized to utilize 32 cores in the cluster fully. The signal processing parts of the applications are divided into threads by exploiting temporal and spatial parallelism. As described above, a large number of small threads make a heavy load on the thread scheduler and may decrease performance, but they are better in terms of parallel execution of threads to exploit more parallelism. We select the suitable thread sizes, 6 macroblocks for the H.264 decoding and 64 macroblocks for the super resolution. Because the H.264 decoding has a strong spatial dependency, the thread size has to be small to get more a thread-level parallelism.

As the evaluation results on the single cluster, the performance scales up to 28 cores on the H.264 decoding and 32 cores on the super resolution respectively. With the further software optimization the H.264 decoding performance may scale to 32 cores. The H.264 1080p 30fps decoding is achieved by 20 cores with about 400mW at 25°C. The 15fps super



(a) H.264 1080p decoding

(b) Super resolution (1920x1080 to 4K2K)

Figure 8. Performance and power consumption of many-core cluster.

resolution is achieved by 32 cores with less than 800mW at 25°C. This means the 30fps 4K2K super resolution would be realized by using two many-core clusters with less than 2W.

V. Conclusions

We have developed the low power and highly integrated many-core SoC. Within a 210mm2 die, the two 32-core clusters are integrated with the dynamically reconfigurable processors, the hardware accelerators, and other peripherals. Its total peak performance exceeds 1.5TOPS (Tera Operations Per Second). The 32 cores and an L2 cache in the many-core cluster are connected by the tree-based Network-on-Chip (NoC).

Unlike other many-core processor, we used energy efficient embedded processor cores to meet requirements of embedded applications. We verified the performance of the many-core by using the realistic multimedia application programs. It can execute the 1080p 30fps H.264 decoding about 400mW and the 4K2K 15fps super resolution less than 800mW. This result shows the many-core processor with energy efficient cores is feasible for the embedded applications.

Furthermore, the many-core architecture keeps software compatibility with our previous multi-core architecture that is based on the parallel thread execution model. This thread model has been designed very simple and to realize parallel execution of a large number of threads effectively. Therefore, it can extend to many-core processor as well.

References

[1] J. Howard, et al., “A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS,” ISSCC, 2010, pp. 108-109.

[2] S. Vangal, et al., “An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS,” ISSCC, 2007, pp. 98-99.

[3] S. Nomura, et al., “A 9.7mW AAC-Decoding, 620mW H.264 720p 60fps Decoding, 8-Core Media Processor with Embedded Forward-Body-Biasing and Power –Gating Circuit in 65nm CMOS Technology,” ISSCC, 2008, pp. 262-263.

[4] T. Mori, et al., “A Power, Performance Scalable Eight-Cores Media Processor for Mobile Multimedia Applications,” JSSCC, 2009, pp. 2957-2965.

[5] T. Yoshikawa, et al., “FlexGrip™: A small and high-performance programmable hardware for highly sequential application,” Cool Chips XIV, 2011.

[6] Y. Tanabe, et al., “A 464GOPS 620GOPS/W Heterogeneous Multi-Core SoC for Image-Recognition Applications,” ISSCC, 2012.