CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Book Chapter

Manycore Platforms

Andrea Marongiu
Vincent Nélis*
Patrick Meumeu Yomsi*

*CISTER Research Centre

CISTER-TR-180706

2018/07/01

Manycore Platforms

Andrea Marongiu, Vincent Nélis*, Patrick Meumeu Yomsi*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: nelis@isep.ipp.pt, pamyo@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

This chapter surveys state-of-the-art manycore platforms. It discusses the historical evolution of computing platforms over the past decades and the technical hurdles that led to the manycore revolution, then presents in details several manycore platforms, outlining (i) the key architectural traits that enable scalability to several tens or hundreds of processing cores and (ii) the shared resources that are responsible for unpredictable timing.

2

Manycore Platforms

Andrea Marongiu[1], Vincent Nélis[2] and Patrick Meumeu Yomsi[2]

[1]Swiss Federal Institute of Technology in Zürich (ETHZ), Switzerland; and University of Bologna, Italy
[2]CISTER Research Centre, Polytechnic Institute of Porto, Portugal

This chapter surveys state-of-the-art manycore platforms. It discusses the historical evolution of computing platforms over the past decades and the technical hurdles that led to the manycore revolution, then presents in details several manycore platforms, outlining (i) the key architectural traits that enable scalability to several tens or hundreds of processing cores and (ii) the shared resources that are responsible for unpredictable timing.

2.1 Introduction

Starting from the early 2000s, general-purpose processor manufacturers adopted the chip multiprocessor (CMP) design paradigm [1] to overcome technological "walls."

Single-core processor designs hit the **power wall** around 2004, when the consolidated strategy of scaling down the gate size of integrated circuits – reducing the supply voltage and increasing the clock frequency – became unfeasible because of excessive power consumption and expensive packaging and cooling solutions [2]. The CMP phisolophy replaces a single, very fast core with multiple cores that cooperate to achieve equivalent performance, but each operating at a lower clock frequency and thus consuming less power.

Over the past 20 years, processor performance has increased at a faster rate than the memory performance [3], which created a gap that is commonly referred to as the **memory wall**. Historically, sophisticated multi-level cache hierarchies have been built to implement main memory access latency hiding techniques. As CMPs use lower clock frequencies, the processor–memory

gap grows at a slower rate, compared to traditional single-core systems. Globally, the traditional latency hiding problem is turned into an increased bandwidth demand, which is easier to address, as the DRAM bandwidth scales much better than its access latency [4].

Single-core designs have traditionally been concerned with the development of techniques to efficiently extract instruction-level parallelism (ILP). However, increasing ILP performance beyond what is achieved today with state-of-the-art techniques has become very difficult [5], which is referred to as the **ILP wall**. CMPs solve the problem by shifting the focus to thread-level parallelism (TLP), which is exposed at the parallel programming model level, rather than designing sophisticated hardware to transparently extract ILP from instruction streams.

Finally, the **complexity** wall refers to the difficulties encountered by single-core chip manufacturers in designing and verifying increasingly sophisticated out-of-order processors. In the CMP design paradigm, a much simpler processor core is designed once and replicated to scale to the multicore system core count. Design reuse and simplified core complexity obviously significantly reduce the system design and verification.

The trend towards integrating an increasing number of cores in a single chip has continued all over the past decade, which has progressively paved the way for the introduction of manycore systems, i.e., CMPs containing a high number of cores (tens to hundreds). Interestingly, the same type of "revolution" has taken place virtually in every domain, from the high-performance computing (HPC) to the embedded systems (ES). Driven by converging needs for high performance requirements, energy efficiency, and flexibility, the most representative commercial platforms from both domains nowadays feature very similar architectural traits. In particular, core *clusterization* is the key design paradigm adopted in all these products. A hierarchical processor organization is always employed, where simple processing units are grouped into small-medium sized subsystems (the *clusters*) and share high-performance local interconnection and memory. Scaling to larger system sizes is enabled by replicating clusters and interconnecting them with a scalable medium like a network-on-chip (NoC).

In the following, we briefly present several manycore platforms, both from the HPC and the ES domains. We discuss the Kalray MPPA-256 at last, and in greater detail, as this is the platform for which the development of the software techniques and the experimental evaluation presented throughout the rest of the book have been conducted.

2.2 Manycore Architectures

2.2.1 Xeon Phi

Xeon Phi are a series of x86 manycore processors by Intel and meant to accelerate the highly parallel workloads of the HPC world. As such, they are employed in supercomputers, servers, and high-end workstations. The Xeon Phi family of products has its roots in the *Larrabee* microarchitecture project – an attempt to create a manycore accelerator meant as a GPU as well as for general-purpose computing – and has recently seen the launch of the Knights Landing (KNL) chip on the marketplace.

Figure 2.1a shows the high-level block diagram of the KNL CPU. It comprises 38 physical *tiles*, of which at most 36 are active (the remaining two tiles are for yield recovery). The structure of a *tile* is shown in Figure 2.1b. Each *tile* comprises two cores, two vector processing units (VPUs) per core, and a 1-Mbyte level-2 (L2) cache that is shared between the two cores.

The core is derived from the Intel Atom (based on the Silvermont microarchitecture [6]), but leverages a new two-wide, out-of-order core which includes heavy modifications to incorporate features necessary for HPC workloads [e.g., four threads per core, deeper out-of-order buffers, higher cache bandwidth, new instructions, better reliability, larger translation look-aside buffers (TLBs), and larger caches]. In addition, the new Advanced

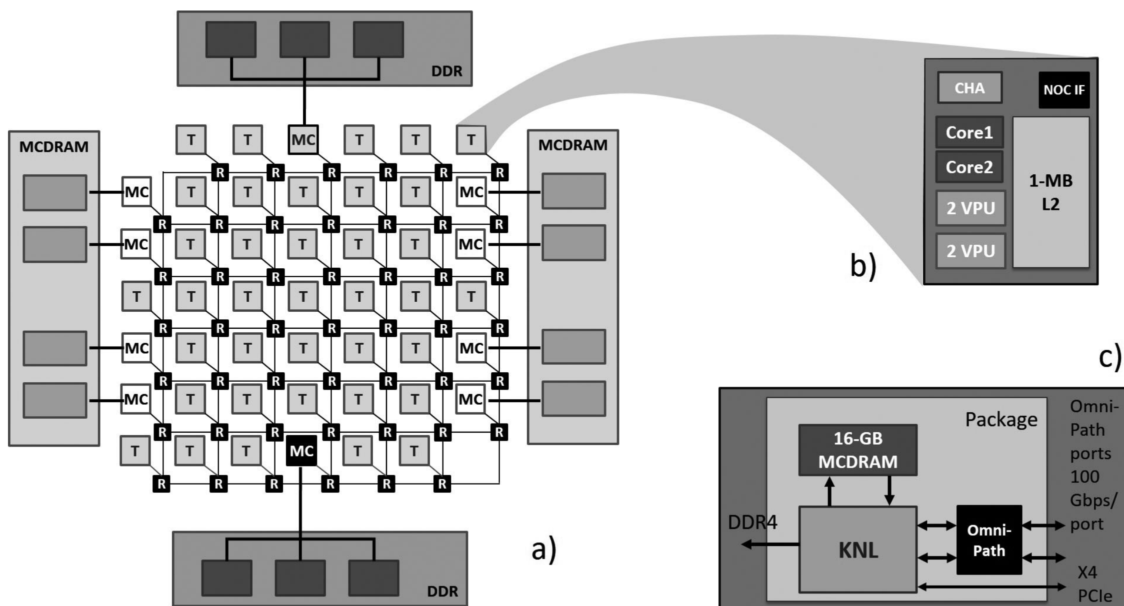

Figure 2.1 Knights Landing (KNL) block diagram: (a) the CPU, (b) an example tile, and (c) KNL with Omni-Path Fabric integrated on the CPU package.

Vector Extensions instruction set, AVX-512, provides 512-bit-wide vector instructions and more vector registers.

At the top level, a 2D, cache-coherent mesh NoC connects the tiles, memory controllers, I/O controllers, and other agents on the chip. The mesh supports the MESIF (modified, exclusive, shared, invalid, forward) protocol, which employs a distributed tag directory to keep the L2 caches in all tiles coherent with each other. Each tile contains a caching/home agent that holds a portion of the distributed tag directory and also serves as a connection point between the tile and the mesh.

Knights Landing features two types of memory: (i) multichannel DRAM (MCDRAM) and (ii) double data rate (DDR) memory. MCDRAM is organized as eight devices – each featuring 2-Gbyte high-bandwidth banks – integrated on-package and connected to the KNL die via a proprietary on-package I/O. The DDR4 is organized as six channels running at up to 2,400 MHz, with three channels on each of two memory controllers.

The two types of memory are presented to users in three memory modes: cache mode, in which MCDRAM is a cache for DDR; flat mode, in which MCDRAM is treated like standard memory in the same address space as DDR; and hybrid mode, in which a portion of MCDRAM is cache and the remainder is flat. KNL supports a total of 36 lanes of PCI express (PCIe) Gen3 for I/O, split into two x16 lanes and one x4 lane. Moreover, it integrates the Intel Omni-Path Fabric on-package (see Figure 2.1c), which provides two 100-Gbits-per-second ports out of the package.

The typical power (thermal design power) for KNL (including MCDRAM memory) when running a computationally intensive workload is 215 W without the fabric and 230 W with the fabric.

2.2.2 Pezy SC

PEZY-SC (PEZY Super Computer) [7] is the second generation manycore microprocessor developed by PEZY in 2014, and is widely used as an accelerator for HPC workloads. Compared to the original PEZY-1, the chip contains exactly twice as many cores and incorporates a large amount of cache including 8 MB of L3$. Operating at 733 MHz, the processor is said to have peak performance of 3.0 TFLOPS (single-precision) and 1.5 TFLOPS (double-precision). PEZY-SC was designed using 580 million gates and manufactured on TSMC's 28HPC+ (28 nm process).

In June 2015, PEZY-SC-based supercomputers took all top three spots on the Green500 listing as the three most efficient supercomputers:

1. **Shoubu**: 1,181,952 cores, 50.3 kW, 605.624 TFlop/s Linpack Rmax;
2. **Suiren Blue**: 262,656 cores, 40.86 kW, 247.752 TFlop/s Linpack Rmax;
3. **Suiren**: 328,480 cores, 48.90 kW, 271.782 TFlop/s Linpack Rmax.

PEZY-SC contains two ARM926 cores (ARMv5TEJ) along with 1024 simpler RISC cores supporting 8-way SMT for a total of 8,192 threads, as shown in Figure 2.2. The organization of the accelerator cores in PEZY-SC heavily uses clusterization and hierarchy. At the top level, the microprocessor is made of four blocks called "*prefectures*." Within a *prefecture*, 16 smaller blocks called "*cities*" share 2 MB of L3$. Each *city* is composed of 64 KB of shared L2$, a number of special function units and four smaller blocks called "*villages*." Inside a *village* there are four execution units and every two such execution units share 2 KB of L1D$.

The chip has a peak power dissipation of 100 W with a typical power consumption of 70 W which consists of 10 W leakage + 60 W dynamic.

2.2.3 NVIDIA Tegra X1

The NVIDIA Tegra X1 [8] is a hybrid System on Module (SoM) featured in the NVIDIA Jetson Development boards. As a mobile processor, the Tegra X1 is meant for the high-end ES markets, and is the first system to feature a

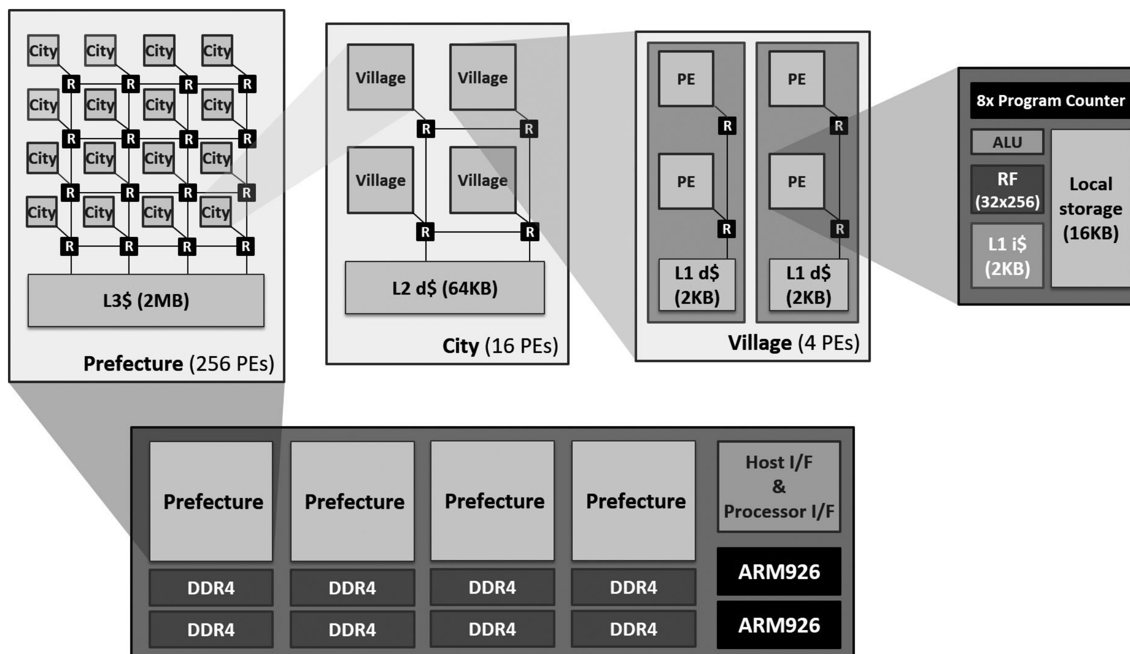

Figure 2.2 PEZY-SC architecture block diagram.

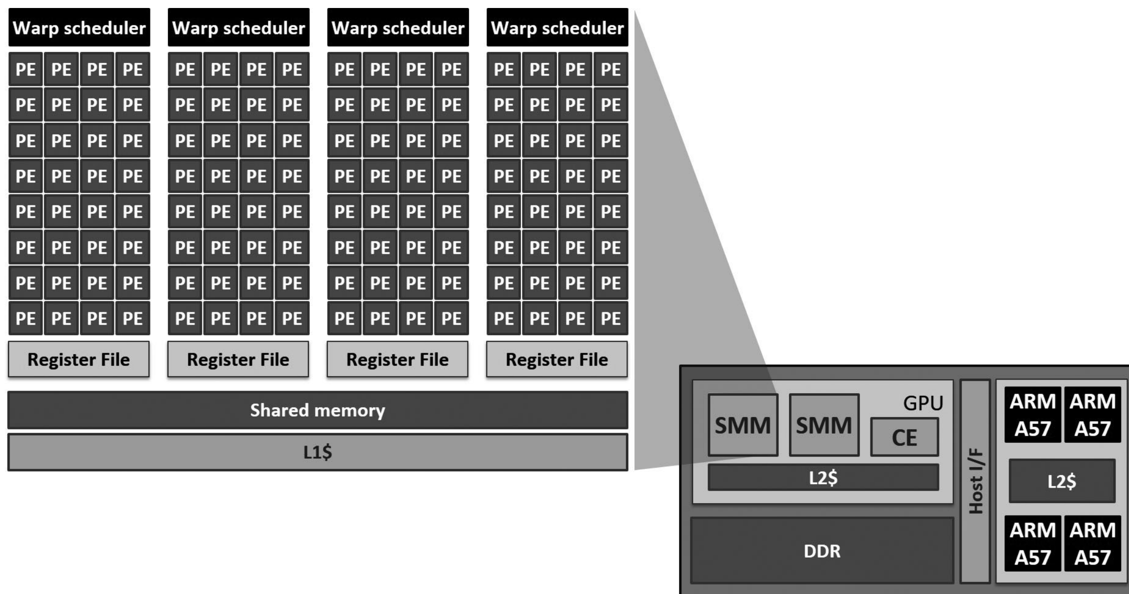

Figure 2.3 NVIDIA Tegra X1 block diagram.

chip powerful enough to sustain the visual computing load for autonomous and assisted driving applications.

As shown in Figure 2.3, the X1 CPU complex consists of a big LITTLE architecture, featuring quad-core 1.9 GHz ARM Cortex-A57 processor (48 KB I-cache + 32 kB D-cache L1 per core, 2 MB L2 cache common to all cores), plus quad-core ARM Cortex A53 processor. A single CPU core can utilize the maximum bandwidth available for the whole CPU complex, which amounts to almost 4.5 GB/s for sequential read operations.

The iGPU is a second-generation Maxwell "GM20b" architecture, with 256 CUDA cores grouped in two Streaming Multi-processors (SMs) (the "*clusters*") sharing a 256 KB L2 (last-level) cache. The compute pipeline of an NVIDIA GPU includes engines responsible for computations (Execution Engine, EE) and engines responsible for high bandwidth memory transfers (Copy Engine, CE). The EE and CE can access central memory with a maximum bandwidth close to 20 GB/s, which can saturate the whole DRAM bandwidth. Indeed, the system DRAM consists of 4 GB of LPDDR4 64 bit SDRAM working at (maximum) 1.6 GHz, reaching a peak ideal bandwidth of 25.6 GB/s.

Despite the high performance capabilities of the SoC (peak performance 1 TFlops single precision), the Tegra X1 features a very contained power envelope, drawing 6–15 W.

2.2.4 Tilera Tile

The *Tile* architecture has its roots in the RAW research processor developed at MIT [9] and later commercialized by Tilera, a start-up founded by the original research group. Chips from the second generation are expected to scale up to 100 cores based on the MIPS ISA and running at 1.5 GHz.

The *Tile* architecture is among the first examples of a cluster-based many-core, featuring *ad-hoc* on-chip interconnect and cache architecture. The architectural template is shown in Figure 2.4. The chip is architected as a 2D array of *tiles* (the *clusters*), interconnected via a mesh-based NoC. Each tile contains a single processor core, with local L1 (64 KB) and a portion (256 KB) of the distributed L2 cache. Overall, the L2 cache segments behave as a non-uniformly addressed cache (NUCA), using a directory-based coherence mechanism and the concept of *home tile* (the tile that holds the master copy) for cached data. The NUCA design makes cache access latency variable according to the distance between tiles, but enables an efficient (space- and power-wise) logical view to the programmer: a large on-chip cache to which all cores are connected. Each tile also features an interconnect switch that connects it to the neighboring tiles, which allows for a simplified interconnect design (essentially, a switched network with very short wires connecting neighboring tiles linked through the tile-local switch).

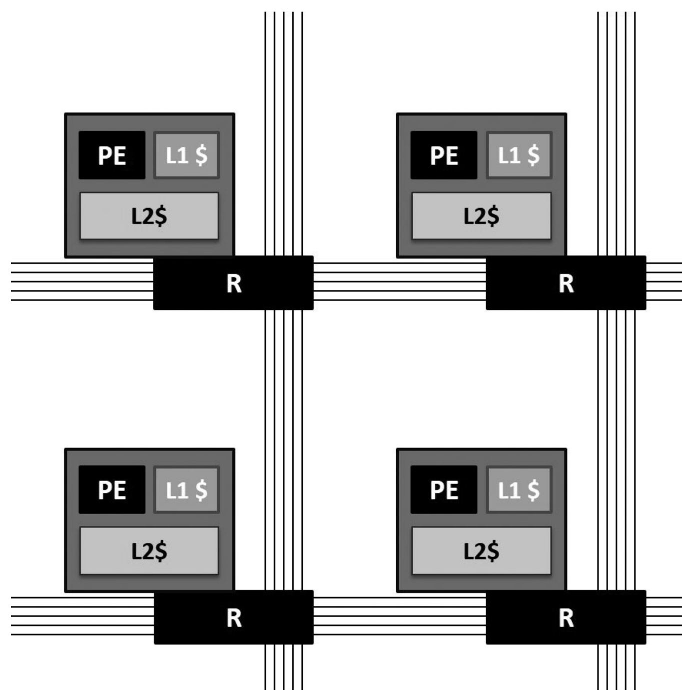

Figure 2.4 Tilera *Tile* architectural template.

The NoC – called *iMesh* by Tilera – actually consists of five different networks, used for various purposes:

- Application process communication (UDN),
- I/O communication (IDN),
- Memory communication (MDN),
- Cache coherency (TDN),
- Static, channelized communication (STN).

The latency of the data transfers on the network is 1–2 cycles/tile, depending on whether there's a direction change or not at the tile. The *TileDirect* technology allows data received over the external interfaces to be placed directly into the tile-local memory, thus bypassing the external DDR memory and reducing memory traffic.

The power budget of the *Tile* processors is under 60 W.

2.2.5 STMicroelectronics STHORM

STHORM is a heterogeneous, manycore-based system from STMicroelectronics [10], with an operating frequency ranging up to 600 MHz.

The STHORM architecture is organized as a fabric of multi-core *clusters*, as shown in Figure 2.5. Each cluster contains 16 STxP70 *Processing*

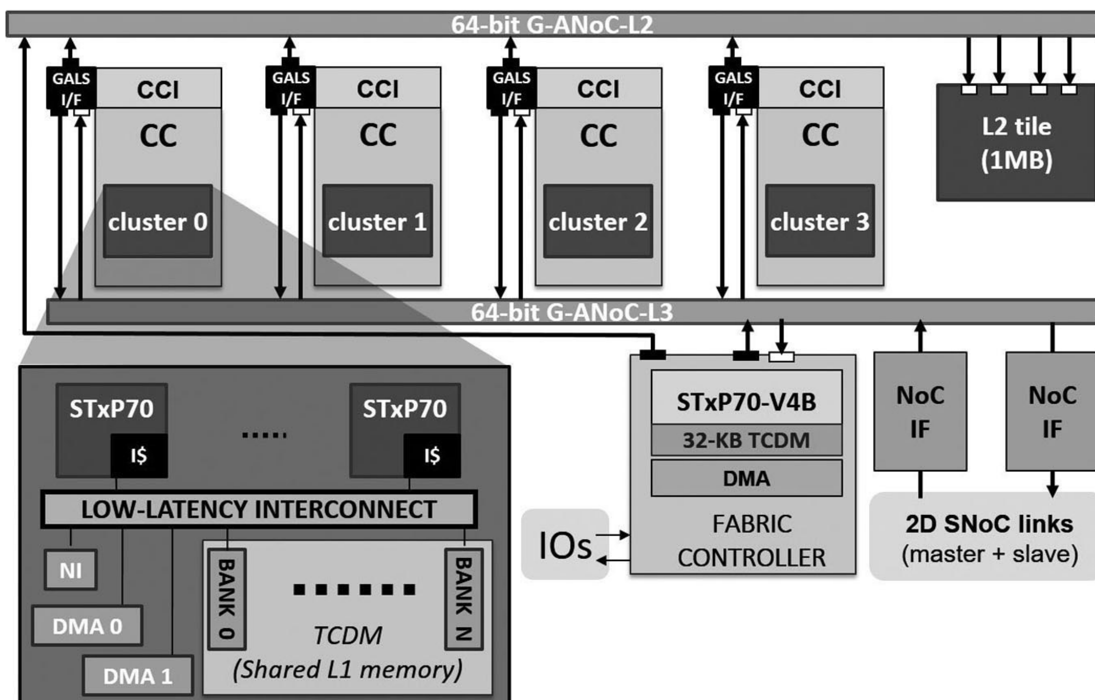

Figure 2.5 STMicroelectronics STHORM heterogeneous system.

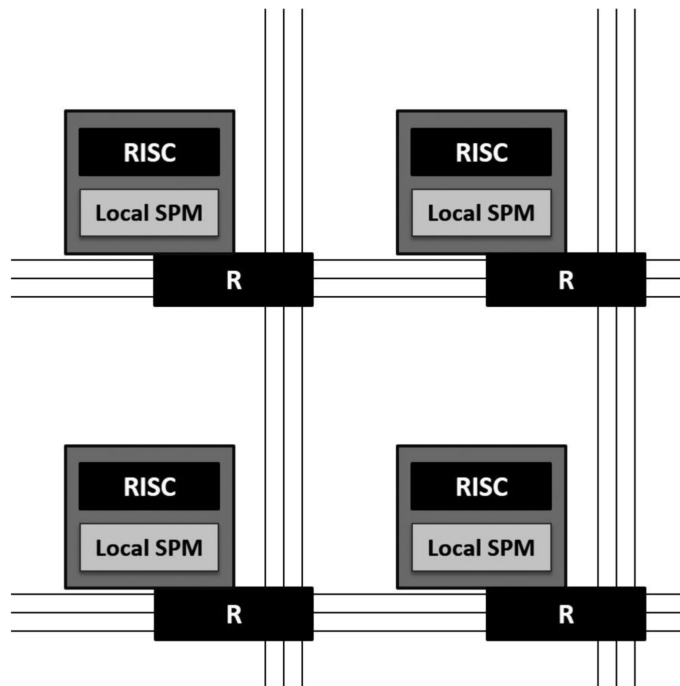

Figure 2.6 Block diagram of the Epiphany-V chip from Adapteva.

Elements (PEs), each of which has a 32-bit dual-issue RISC processor. PEs communicate through a shared multi-ported, multi-bank, tightly-coupled data memory (TCDM, a scratchpad memory). Additionally, STHORM clusters feature an additional core called the *cluster controller* (CC) and meant, as the name suggests, for the execution of control code local to the cluster operation. Globally, four *clusters* plus a *fabric controller* (FC) core – responsible for global coordination of the clusters – are interconnected via two asynchronous networks-on-chip (ANoC). The first ANoC is used for accessing a multi-banked, multiported L2 memory, shared among the four clusters. The second ANoC is used for inter-cluster communication via L1 TCDMs (i.e., remote clusters' TCDMs can be accessed by every core in the system) and to access the offchip main memory (L3 DRAM).

STHORM delivers up to 80 GOps (single-precision floating point) with only 2W power consumption.

2.2.6 Epiphany-V

The Epiphany-V chip from Adapteva [11] is based on a 1024-core processor in 16 nm FinFet technology. The chip contains an array of 1024 64-bit RISC processors, 64 MB of on-chip SRAM, three 136-bit wide mesh Networks-On-Chip, and 1,024 programmable IO pins.

Similar to the Tilera *Tile* architecture, the Epiphany architecture is a distributed shared memory architecture composed of an array of RISC processors communicating via a low-latency, mesh-based NoC, as shown in Figure 2.6. Each cluster (or *node*) in the 2D array features a single, complete RISC processor capable of independently running an operating system [according to the multiple-instruction, multiple-data (MIMD) paradigm]. The distributed shared memory model of the Epiphany-V chip relies on a cacheless design, in which all scratchpad memory blocks are readable and writable by all processors in the system (similar to the STHORM chip).

The Epiphany-V chip can deliver two teraflops of performance (single-precision floating point) in a 2W power envelope.

2.2.7 TI Keystone II

The Texas Instrument Keystone II [12], is a heterogeneous SoC featuring a quad-core ARM Cortex-A15 and an accelerator cluster comprising eight C66x VLIW DSPs. The chip is designed for special-purpose industrial tasks, such as networking, automotive, and low-power server applications. The 66AK2H12 SoC, depicted in Figure 2.7, is the top-performance Texas Instrument Keystone II device architecture.

Each DSP in the accelerator cluster is a VLIW core, capable of fetching up to eight instructions per cycle and running at up to 1.2 GHz. Locally,

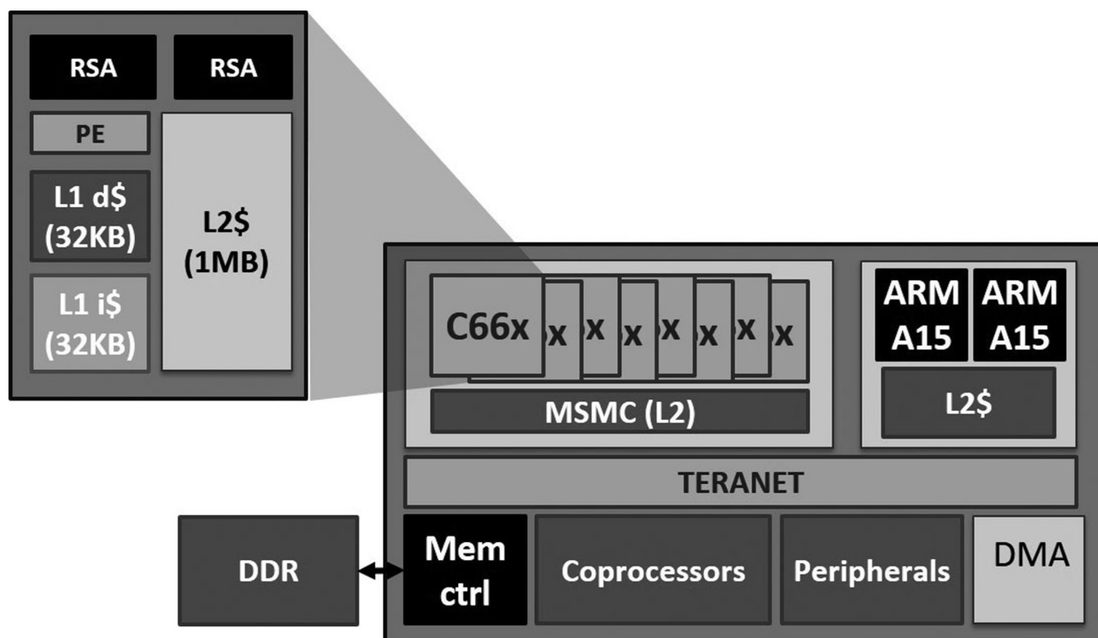

Figure 2.7 Texas Instrument Keystone II heterogeneous system.

a DSP is equipped with 32 KB L1 D-cache and L1 I-cache, plus 1024 KB L2 unified cache. Altogether, the DSPs in the accelerator cluster deliver 160 single-precision GOps.

On the ARM side, there are 32 KB of L1 D-cache and 32 KB of L1 I-cache per core, plus a coherent 4 MB L2 cache.

The computational power of such architecture, at a power budget of up to 14 W, makes it a low-power solution for microserver-class applications. The Keystone II processor has been used in several cloud-computing/microserver settings [13–15].

2.2.8 Kalray MPPA-256

The Kalray MPPA-256 processor of the MPPA (Multi-Purpose Processor Array) MANYCORE family has been developed by the company KALRAY. It is a single-chip programmable manycore processor manufactured in 28 nm CMOS technology that targets low-to-medium volume professional applications, where low energy per operation and time predictability are the primary requirements [16]. It concentrates a great potential and is very promising for high-performance parallel computing. With an operating frequency of 400 MHz and a typical power consumption of 5 W, the processor can perform up to 700 GOPS and 230 GFLOPS. The processor integrates a total of 288 identical Very Long Instruction Word (VLIW) cores including 256 user cores referred to as processing engines (PEs) and dedicated to the execution of the user applications and 32 system cores referred to as Resource Manager (RM) and dedicated to the management of the software and processing resources. The cores are organized in 16 compute clusters and four I/O subsystems to control all the I/O devices. In Figure 2.8, the 16 inner nodes (labeled CC)

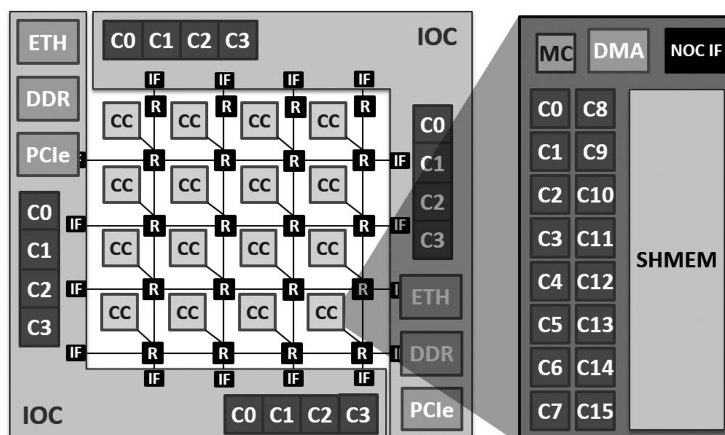

Figure 2.8 High-level view of the Kalray MPPA-256 processor.

correspond to the 16 compute clusters holding 17 cores each: 16 PEs and 1 RM. Then, there are four I/O subsystems located at the periphery of the chip, each holding four RMs. Each compute cluster and I/O subsystem owns a private address space, while communication and synchronization between them is ensured by the data and control NoC depicted in Figure 2.8. The MPPA-256 processor is also fitted with a variety of I/O controllers, in particular DDR, PCI, Ethernet, Interlaken, and GPIO.

2.2.8.1 The I/O subsystem

The four I/O subsystems (also denoted as IOS) are referenced as the North, South, East, and West IOS. They are responsible for all communications with elements outside the MPPA-256 processor, including the host workstation if the MPPA is used as an accelerator.

Each IOS contains four RMs in a symmetric multiprocessing configuration. These four RMs are connected to a shared, 16-bank parallel memory of 512 KB, they have their own private instruction cache of 32 KB (8-way, set-associative) and share a data cache of 128 KB (also 8-way, set-associative), which ensures data coherency between the cores.

The four IOS are dedicated to PCIe, Ethernet, Interlaken, and other I/O devices. Each one runs either a rich OS such as Linux or an RTOS that supports the MPPA I/O device drivers. They integrate controllers for an 8-lane Gen3 PCIe for a total peak throughput of 16 GB/s full duplex, Ethernet links ranging from 10 MB/s to 40 GB/s for a total aggregate throughput of 80 GB/s, the Interlaken link providing a way to extend the NoC across MPPA-256 chips and other I/O devices in various configurations like UARTs, I2C, SPI, pulse width modulator (PWM), or general purpose IOs (GPIOs). More precisely, the East and West IOS are connected to a quad 10 GB/s Ethernet controller, while the North and South IOS are connected to an 8-lane PCIe controller and to a DDR interface for access to up to 64 GB of external DDR3-1600.

2.2.8.2 The Network-on-Chip (NoC)

The NoC holds a key role in the average performance of manycore architectures, especially when different clusters need to exchange messages. In the Kalray MPPA-256 processor, the 16 compute clusters and the four I/O subsystems are connected by two explicitly addressed NoC with bi-directional links providing a full duplex bandwidth up to 3.2 GB/s between two adjacent nodes:

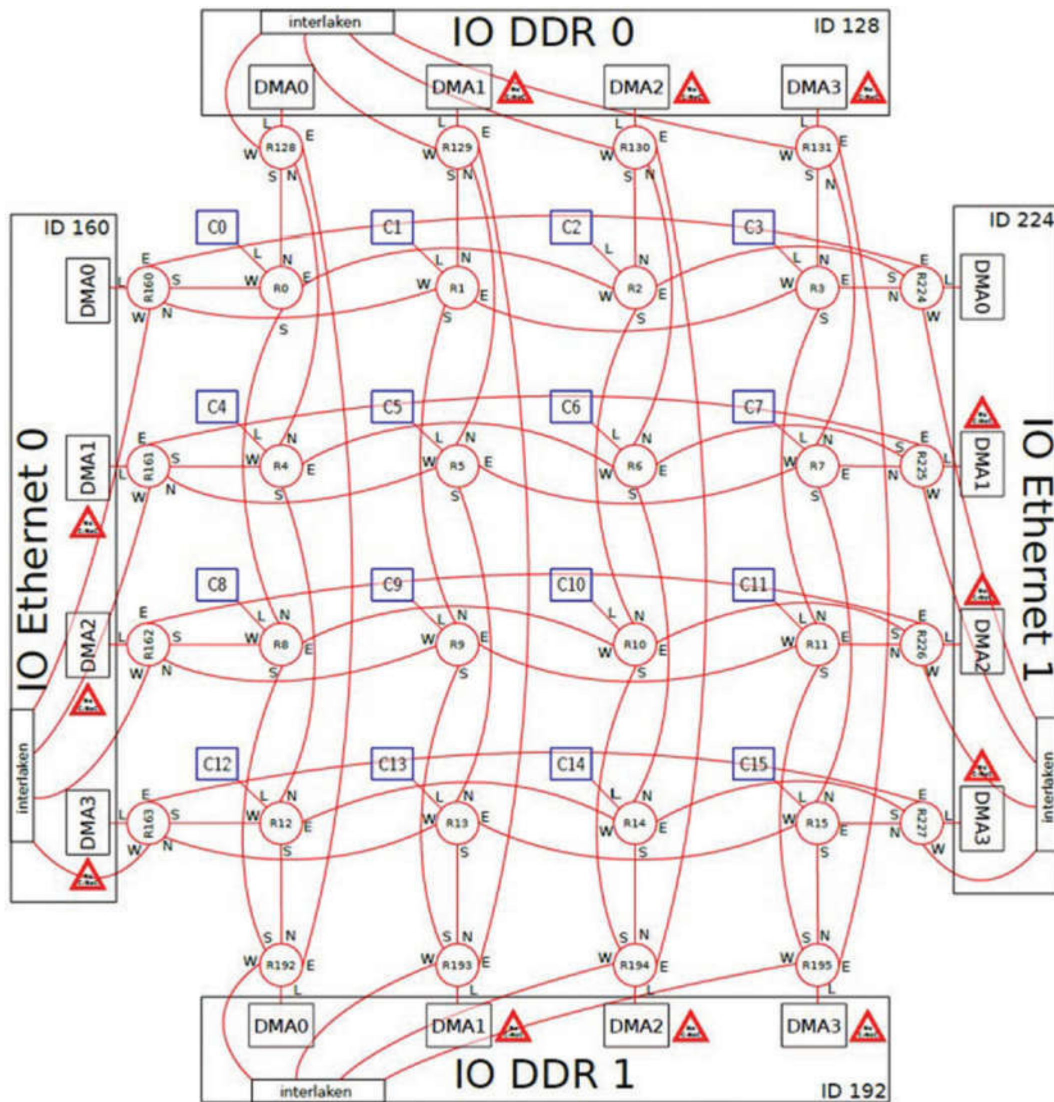

Figure 2.9 MPPA-256 NoC architecture.

- The data NoC (D-NoC). This NoC is optimized for bulk data transfers;
- The control NoC (C-NoC). This NoC is optimized for small messages at low latency.

The two NoCs are identical with respect to the nodes, the 2D-wrapped-around torus topology, shown in Figure 2.9, and the wormhole route encoding. They differ at their device interfaces, by the amount of packet buffering in routers, and by the flow regulation at the source available on the D-NoC. NoC traffic through a router does not interfere with the memory buses of the underlying I/O subsystem or compute cluster, unless that router is the destination node. Besides, the D-NoC implements a quality-of-service (QoS) mechanism, thus guaranteeing predictable latencies for all data transfers.

2.2.8.3 The Host-to-IOS communication protocol

The special hierarchy among the cores in the MPPA-256 processor helps to better divide the workload to be executed on the PEs. When the MPPA-256 is used as an accelerator, tasks are sent to the MPPA-256 processor from a Host workstation. The communication with the MPPA-256 can thus be performed in a couple of steps which can be referred to as Host-to-IOS, IOS-to-Clusters and finally Cluster-to-Cluster communication protocols. The MPPA-256 processor communicates with the Host workstation through I/O subsystems. The chip is connected to the host CPU by a PCle interface and two connectors – **Buffer** and **MQueue** – are available for making this link. The RM core that accommodates the task upon the I/O subsystem is referred to as **Master** (see Figure 2.10). The processor then executes the received task (referred to as **Master task**) as detailed in Section 4.3.1 and at the end of the execution process, it writes the output data in a 4 GB DDR3 RAM memory, which is connected to an I/O subsystem and can be accessed by the host CPU.

2.2.8.4 Internal architecture of the compute clusters

The compute cluster (Figure 2.11) is the basic processing unit of the MPPA architecture. Each cluster contains 17 Kalray-1 VLIW cores, including 16 PE cores dedicated to the execution of the user applications and one RM core. Among other responsibilities, the RM is in charge of mapping and scheduling the threads on the PEs and managing the communications between the clusters and between the clusters and the main memory. The 16 PEs and the RM

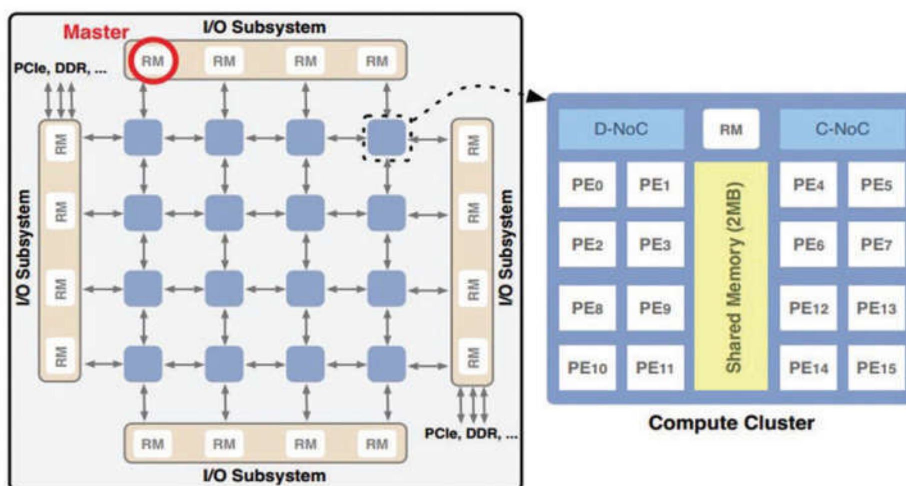

Figure 2.10 A master task runs on an RM of an I/O subsystem.

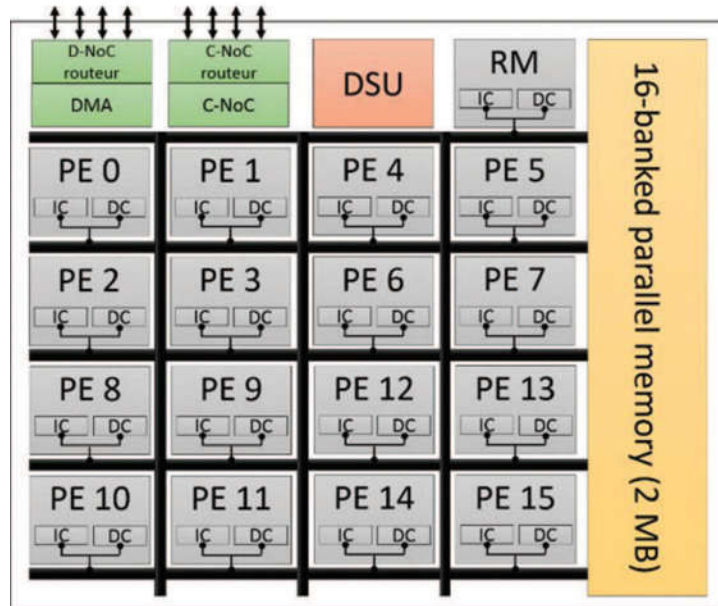

Figure 2.11 Internal architecture of a compute cluster.

are connected to a shared memory of 2 MB. A direct memory access (DMA) engine is responsible for transferring data between the shared memory and the NoC or within the shared memory. The DMA engine supports multi-dimensional data transfers and sustains a total throughput of 3.2 GB/s in full duplex. The Debug and System Unit (DSU) supports the compute cluster debug and diagnostics capabilities. Each DSU is connected to the outside world by a JTAG (IEEE 1149.1) chain. The DSU also contains a system trace IP that is used by lightly instrumented code to push up to 1.6 GB/s of trace data to an external acquisition device. This trace data gives almost non-intrusive insight on the behaviour of the application.

2.2.8.5 The shared memory

The shared memory (SMEM) in each compute cluster (yellow box in Figure 2.11) comprises 16-banked independent memory of 16,384 x 64-bit words = 128 kB per bank, with a total capacity of 16 x 128 kB = 2 MB, with error code correction (ECC) on 64-bit words. This memory space is shared between the 17 VLIW cores in the cluster and delivers an aggregate bandwidth of 38.4 GB/s.

The 16 memory banks are arranged in two sides of eight banks, the left side and the right side. The connections between the memory bus masters are replicated in order to provide independent access to the two sides. There are two ways of mapping a physical address to a specific side and bank.

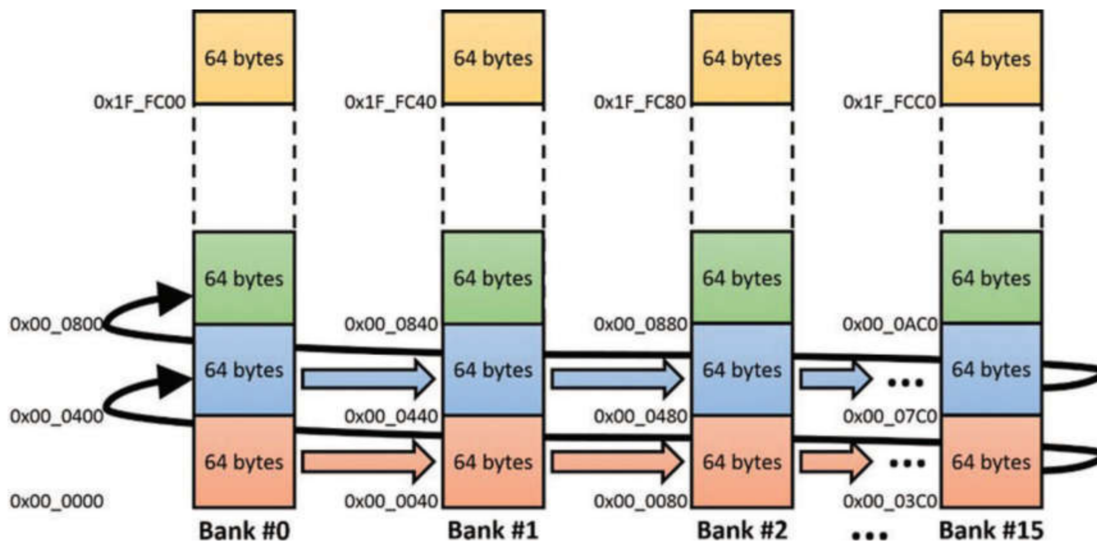

Figure 2.12 Memory accesses distributed across memory banks (interleaved).

Option 1 (Interleaving address mapping) – In the address space, bits 6–9 of the byte address select the memory bank, so sequential addresses move from one bank to another every 64 bytes (every 8 x 64-bit words), as depicted in Figure 2.12. This address-mapping scheme is effective at distributing the requests of cores across memory banks, while ensuring that each cache refill request involves only one memory bank and benefits from a burst access mode. Furthermore, this address scheme also allows the "simultaneous" access (respecting the activation time) of those memory banks in which the cache line is stored. As the side selection depends on the sixth bit of the byte address, the bank selection by sequential addresses alternates between the left side and the right side every 64 bytes.

Option 2 (Contiguous address mapping) – It is possible to disable the memory address shuffling, in which case each bank has a sequential address space covering one bank of 128 KB as depicted in Figure 2.13. The high-order bit of the address selects the side (i.e., the right side covers addresses from 0 to 1 MB and the left side covers addresses above 1 MB). When zero interference between cores is needed, cores within a given pair must use a different side.

2.3 Summary

Back in the early days of the new millennium, multicore processors allowed computer designers to overcome several technological *walls* that traditional single-core design methodologies were no longer capable of addressing. This

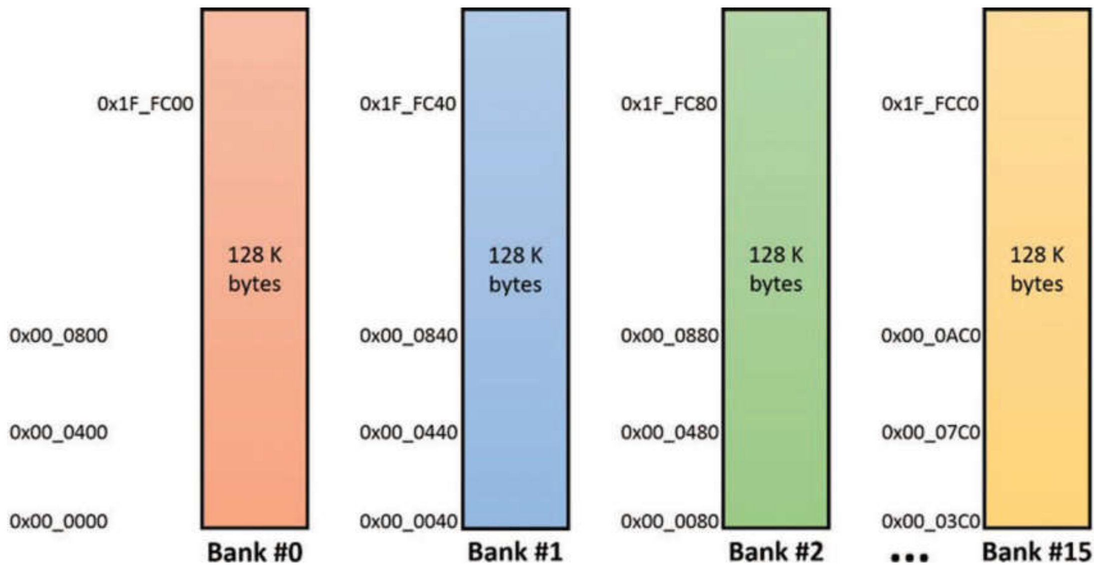

Figure 2.13 Memory accesses targeting a same memory bank (contiguous).

design paradigm is to date the standard, with an ever-increasing number of processing cores integrated on the same chip. While manycore processors enabled over the past 15 years the seamless continuation of compute performance scalability for general-purpose and scientific workloads, real-time systems have not been able to embrace this technology so far, due to the lack of predictability in execution time implied by hardware resource sharing. This chapter has surveyed several state-of-the-art manycore processors, highlighting the architectural features (i) that enable processor integration scalability and (ii) those that are shared among several processor and that are mostly responsible for the unpredictable execution.

References

[1] Olukotun, K., Nayfeh, B. A., Hammond, L., Wilson, K., and Chang, K., The case for a single-chip multiprocessor. *SIGOPS Oper. Syst. Rev.*, 30, 2–11, 1996.

[2] Fuller, S. H., and Millett, L. I., Computing performance: Game over or next level? *Computer*, 44, 31–38, 2011.

[3] Hennessy, J. L., and Patterson, D. A., *Computer Architecture: A Quantitative Approach*. Elsevier, 2011.

[4] Patterson, D. A., Latency lags bandwith. *Commun. ACM*, 47, 71–75, 2004.

[5] Agarwal, V., Hrishikesh, M. S., Keckler, S. W., and Burger, D., "Clock rate versus ipc: the end of the road for conventional microarchitectures." In *Proceedings of 27th International Symposium on Computer Architecture (IEEE Cat. No.RS00201)*, pages 248–259, 2000.
[6] Naffziger, S., and Sohi, G., Hot chips 26. *IEEE Micro*. 35, 4–5, 2015.
[7] Tabuchi, A., Kimura, Y., Torii, S., Matsufuru, H., Ishikawa, T., Boku, T., and Sato, M., *Design and Preliminary Evaluation of Omni OpenACC Compiler for Massive MIMD Processor PEZY-SC*, pp. 293–305, Springer International Publishing, Cham, 2016.
[8] NVIDIA SRL. *Whitepaper: NVIDIA Tegra X1 – NVIDIA's New Mobile Superchip*. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html, accessed November 07, 2011.
[9] Taylor, M. B., Kim, J., Miller, J., Wentzlaff, D., Ghodrat, F., Greenwald, B., et al. The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *IEEE Micro*, 22, 25–35, 2002.
[10] Melpignano, D., Benini, L., Flamand, E., Jego, B., Lepley, T., Haugou, G., Clermidy, F., and Dutoit, D., "Platform 2012, a many-core computing accelerator for embedded socs: Performance evaluation of visual analytics applications." In *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, pp. 1137–1142, New York, NY, USA, 2012.
[11] Olofsson, A., Epiphany-v: A 1024 processor 64-bit RISC system-on-chip. *CoRR*, abs/1610.01832, 2016.
[12] Stotzer, E., Jayaraj, A., Ali, M., Friedmann, A., Mitra, G., Rendell, A. P., and Lintault, I., *OpenMP on the Low-Power TI Keystone II ARM/DSP System-on-Chip*, pp. 114–127, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
[13] Verma, A., and Flanagan, T., A Better Way to Cloud. *Texas Instruments white paper*, 2012.
[14] Hewlett-Packard Development Company L.P. *HP ProLiant m800 Server Cartridge*.
[15] nCore HPC LLC. *Brown Dwarf Y-Class Supercomputer*.
[16] Amdahl, G. M., "Validity of the single processor approach to achieving large scale computing capabilities." In *Proceedings of the April 18–20, 1967, Spring Joint Computer Conference, AFIPS '67* (Spring), pp. 483–485, New York, NY, USA, 1967.