

Chapter 2

Design Transformation from a Single-Core to a Multi-Core Architecture Targeting Massively Parallel Signal Processing Algorithms

Waqar Hussain, Henry Hoffmann, Tapani Ahonen, and Jari Nurmi

2.1 Introduction

Parallel processing of digital signals demands an ever increasing throughput and is expected to increase in future. Considering the upcoming 5G standards, the chip designs accelerating such computationally intensive tasks are subject to many challenges. This includes but not limited to achieve specified execution-time constraints, resource/area consumption on the chip and to deal with power/energy walls. The Dennardian scaling of transistor feature size was promising to increase the number of transistors on the chip while keeping the power dissipation constant [8]. This scaling was supporting Moore's law, i.e., the number of transistors on a silicon chip will double after every 2 years. However, the down-scaling of the transistor size and increasing the operating frequency increased the power density of the chip to unprecedented thermal levels and further packing of more transistors while keeping the power dissipation constant became infeasible. A solution was found in multi-core scaling which lasted not for long, where multiple cores are combined together on a chip and the workload is distributed over all the cores. Studies by the research group of Michael B. Taylor at the University of California, San Diego, have shown with an experimental evidence that the power-wall is obstructing again the existence of Moore's law. The continuous scaling in the number of cores is not consistent for keeping the power dissipation constant. The study shows that only a 7.0 % area of a 300 mm² homogeneous multi-core chip can

W. Hussain • T. Ahonen • J. Nurmi

Department of Electronics and Communications Engineering, Tampere University of Technology, Tampere, Finland

e-mail: waqar.hussain@tut.fi; ukaata@gmail.com; jari.nurmi@tut.fi

H. Hoffmann (✉)

Department of Computer Science, The University of Chicago, Chicago, IL, USA

e-mail: hankhoffmann@cs.uchicago.edu

be operated at its full frequency within a power budget of 80 W [21]. Hence most of the part of the chip stays under-utilized, i.e., switched-off (Dark) or being operated at a very low frequency (Dim). The overall terminology used for dark or dim part of chip is known as Dark Silicon. Different research groups across the world are addressing this problem in different domains. For example, Hank Hoffmann at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA, presented a Self-Aware Computing model to dynamically control the voltage and operating frequency for power mitigation of a video decoder using an adaptive feedback control system. Efforts have been placed to design 3D integrated circuits for energy efficiency as well as the traditional Dynamic Voltage and Frequency Scaling (DVFS) approach is also actively pursued. The author in [19] has suggested the use of Coarse-Grained Reconfigurable Arrays (CGRAs) as one of the possible solutions to deal with this issue. Since CGRAs provide a large number of Processing Elements (PEs) to work in parallel, they can process computationally intensive signal processing tasks at a high throughput while being operated at a low frequency. The parts of the silicon which are dark can be replaced by custom-computing CGRAs, and since they are reconfigurable, they can adapt to the needs to multiple applications.

In this chapter, we introduce a scalable model of a template-based CGRA and a brief description of the crafting mechanism to generate application-specific CGRA-based accelerators. Then using the CGRAs for integration over a Network-on-Chip (NoC) to establish a Heterogeneous Accelerator-Rich Platform (HARP). HARP is also a template-based design and its instances can contain a variable number of Reduced Instruction-Set Computing (RISC) cores as well as user-specified number of CGRAs. The overall design demonstrates the advantages of loose-coupling where all the cores over the NoC can access each other's communication and computation resources. The design of HARP is an effort to maximize the number of reconfigurable processing resources on a single chip. HARP has been evaluated for a proof-of-concept test as well as for a relatively realistic design problem, i.e., satisfying the execution-time constraints for Fast Fourier Transform (FFT) processing required in the baseband processing of IEEE-802.11n demodulators. In addition to this, HARP has been subjected to constraint-aware workload balancing in conjunction with dynamic frequency scaling. This work showed interesting results and the importance of adopting cost effective self-aware computing models in heterogeneous multi-core architectures. The HARP design has also been evaluated for suitability to 3D Integrated Circuit (IC) design and as a competitive contender in the dark silicon era.

This chapter is organized as follows. The next section describes the existing multi-core platforms that have a similar architecture as of HARP. Section 2.3 explains the design and usage of the CGRAs to instantiate application-specific accelerators to be integrated to HARP nodes. The next following section describes the NoC which is used as a backbone for communication and control between all the nodes. The overall HARP architecture is described in Sect. 2.6 followed by hardware and software integration details presented in Sect. 2.5. The measurement and estimation of different basic and advanced level performance metrics are

described in Sect. 2.8. Section 2.9 establishes comparisons between HARP and the state-of-the-art platforms described in Sect. 2.2. In the last section, conclusions are presented.

2.2 Existing State of the Art

The multi-core platforms exist both in homogeneous and heterogeneous forms. A homogeneous platform is generally composed of similar cores loosely connected with each other while a heterogeneous platform is a loose or tight integration of different cores over a communication infrastructure. Both forms of the multi-core platforms are programmable in C but for heterogeneous platforms, additional custom tools may be required than a simple C compiler. In a heterogeneous platform, cores are generally application-specific targeting only a set of applications. For example, reconfigurable Application-Specific Instruction-set Processors (rASIPs), Fine-Grained, Mid-Grained, and Coarse-Grained Reconfigurable Array devices, Very Long Instruction Word (VLIW) processors, and nevertheless to mention the Digital Signal Processors (DSPs).

2.2.1 MORPHEUS

Morpheus is an integrated platform made of a Fine-, Mid-, and a Coarse-Grained Reconfigurable Array (CGRA) [23]. The platform supports an operating system and uses a custom-designed tool chain for reconfigurable accelerators. The Fine-, Mid-, and Coarse-Grained devices are called as FlexEOS [22], DREAM [6], and XPP-III [7] and these devices communicate over an NoC. FlexEOS is a high-density multi-logic FPGA fabric made of SRAM cells. DREAM is a reconfigurable DSP core which can perform general-purpose processing using a 32-bit RISC processor and the reconfiguration feature is supported by PiCoGA [16]. XPP-III is a CGRA integrated into the datapath of a VLIW processor.

2.2.2 P2012

P2012 is a homogeneous multi-core platform composed of 16 processors equally divided into four clusters [18]. All the processors in a cluster are locally synchronous and globally asynchronous with processors in other clusters. The P2012 platform is tested for image processing algorithms and shows a performance of 0.04 Giga Operations Per Second per Milli Watt (GOPS/mW) on a 90 nm Complimentary Metal-Oxide Semiconductor (CMOS) technology.

2.2.3 *NineSilica*

NineSilica is a homogeneous Multi-Processor System-on-Chip (MPSoC) platform, designed and developed at Tampere University of Technology, Finland [2]. The platform contains nine RISC cores integrated together over an NoC in a topology of three rows and three columns. Each RISC core has an instruction and a data memory. The synchronization between cores is established based on using a shared-memory architecture, where the data is exchanged using flags. NineSilica has a track record of processing many computationally intensive signal processing related algorithms, e.g., Fast Fourier Transform (FFT).

2.2.4 *RAW*

Reconfigurable Architecture Workstation (RAW) is composed of sixteen 32-bit MIPS2000 processors placed in an order of four rows and four columns [20] and the communication between the processors is facilitated by an NoC. RAW supports static and dynamic scheduling hence being similar to a reconfigurable fabric as well as a multi-core platform.

2.2.5 *CRISP*

CRISP was a European Union funded project that stands for Cutting edge Reconfigurable ICs for Stream Processing [1]. It presents a general stream processor which is a highly scalable platform with core-level coarse-grained granularity. The platform targets streaming DSP applications and is designed on the principle of locality-of-reference which indicates different levels of storage. An instance of this platform was generated which was composed of a general-purpose processor and a reconfigurable fabric. The system also contains an ARM core and 45 Xentium DSP cores, where all the cores are integrated with each other using an NoC. The Xentium processor can operate at 200.0 MHz using a 90 nm CMOS technology. The overall general-purpose processor device is manufactured using UMC 130 nm CMOS technology, yielding an operating frequency of 200.0 MHz.

2.2.6 *Intel's Single-Chip Cloud Computer*

A 48 core x86-based Single-Chip Cloud Computer (SCC) was designed by Intel to provide an opportunity for design-space exploration with custom-application implementation and to explore future design challenges and opportunities [17].

The communication between cores is provided using a mesh network and memory controllers. The SCC is implemented using a 45 nm CMOS technology and all the processors can be tuned to different operating frequencies between 100.0 and 800.0 MHz. The communication network can be operated from 0.8 to 1.6 GHz and the platform has been benchmarked by different research groups across the globe.

2.2.7 *TILE64™*

TILE64™ is a homogeneous multi-processor composed of 64 cores arranged in an order of eight rows and eight columns and covers a wide range of embedded applications [4]. Each processor core is a 64-bit VLIW machine but the integer datapath is 32-bit wide to support subword computations. All the cores support SMP Linux kernel and run at 750.0 MHz while delivering a performance of 384 GOPS. All the cores communicate with each using an NoC which provides a bandwidth of 120.0 GB/s. The processor system has three layers of caches and each core has a autonomous DMA engine.

2.3 Scalable CGRAs

In this section, special emphasis is given to briefly describe the family of template-based CGRAs used in the integrated design of HARP. In particular, the CGRAs used are CREMA [9], AVATAR [11], and SCREMA [12]. Figure 2.1 shows combined representation of all the three CGRAs. The unit of processing in all the CGRAs is a Processing Element (PE) which is a 32-bit Arithmetic and Logic Unit (ALU) capable to perform arithmetic and logic operations in integer and IEEE-754 format. Each PE has two inputs (a, b) and two outputs (A, B). The inputs are supplied with operands which reside in the corresponding registers inside each PE. An arithmetic or logic operation can be performed on the operands received by the registers and the output A provides the results. The output B provides the registered value of input b, so to be multiplexed to the inputs of other PEs for the maximum availability of data. The template feature of the CGRAs allows all the processing elements to instantiate only those computational resources that are required by the set of target applications. This is specified by the user manually during the system design time using a custom-made Graphical User Interface (GUI) tool. It optimizes resource utilization on FPGA or area on an ASIC, increases operating frequency, and reduces power and energy consumption. The PEs can exchange data with each other using point-to-point local and global interconnections. Local interconnections are with the neighboring PEs while the global ones are with the PEs that are located relatively farther than the neighboring ones. CREMA is an $R \times 8$, AVATAR is an $R \times 16$, and SCREMA is an $R \times C$ PE CGRA, where R can be any positive integer and C can only have values, i.e., 4, 8, 16, and 32. The CGRAs are equipped with two local

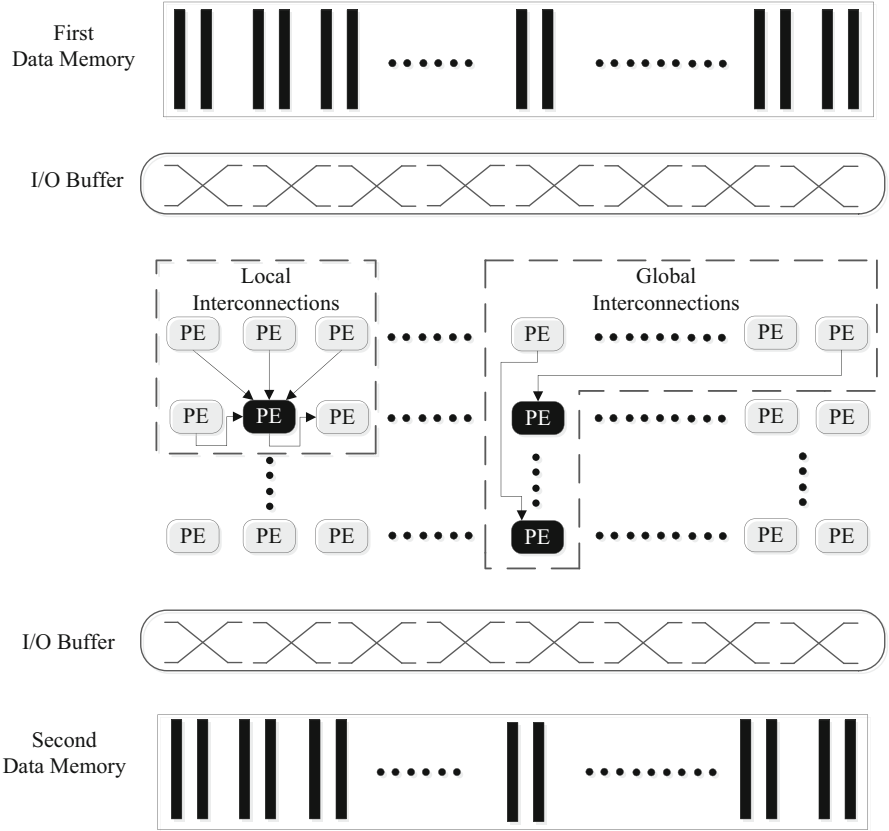


Fig. 2.1 A combined representation of CREMA, AVATAR, and SCREMA. The figure shows local and global interconnections towards destined PEs shown in *black*, I/O buffers, and the data memories [13]

memories to contain the data to be processed as shown in Fig. 2.1. The figure also shows two I/O buffers but they are only instantiated in the designs of CREMA and AVATAR. The I/O buffers make each memory bank in the local memory accessible to almost each and every input of the PE array for read and write operations. I/O buffers are composed of multiplexers and registers. The size of the multiplexer will always be $2C \times 1$ and there will always be $2C$ number of multiplexers in each I/O buffer. SCREMA does not carry I/O buffers as in case of $C = 32$, the number and size of each multiplexer will be very large thus consuming a large portion of the overall design space. These CGRAs were initially designed to work as accelerators with an RISC core. A Direct Memory Access (DMA) device was tightly integrated with each CGRA to transfer configuration and data stream from the main memory of the system to the local data memories of the CGRA during the system startup time. A new configuration can also be loaded as required during the run-time making the

CGRA dynamically reconfigurable. Similarly, for continuous processing, new data can be loaded whenever required. The user can write the custom program flow which is essentially composed of three major portions. (1) Loading the configuration and data to be processed; (2) Enabling different context for cycle-accurate processing; and (3) Transferring the computed results from the CGRA to the main memory of the system. A context means a pattern of interconnections among the PEs and the operation to be performed by each PE at a particular cycle. These CGRAs have a track record of processing a number of computationally intensive signal processing related algorithms, for example, different length and types of FFTs, complex and real Matrix-Vector Multiplication (MVM), and Correlations. The implementation details can be found in [10, 12] and [11], respectively.

2.4 The Network-on-Chip

As described earlier, the communication over HARP is supported by an NoC, a network of nine nodes placed in a geometry of three rows and three columns. The NoC nodes in HARP can act as supervisors or workers based on the user-specified custom design. The overall architecture of HARP is template based, so the user can select particular nodes to be instantiated at the system design time. All the nodes can be integrated to either an RISC core or to the CGRAs mentioned in Sect. 2.3. A combined view of the NoC node integrated with RISC or the DMA is shown in Fig. 2.2. The NoC nodes function as routers for a master and two slave devices. If a node is connected to an RISC core, the master interface of the node is integrated with an RISC core and rest of the two slaves are interfaced with the instruction and data memory. The RISC can access these memories locally within the node and can also write to the network to send data to other devices connected to other nodes. A node can also be interfaced with the CGRA which is tightly integrated with a DMA device. The DMA device has master and slave ports which are interfaced with the master and slaves ports of the NoC node. The other slave interface of the node is integrated with the data memory. The DMA slave can be written from the NoC to invoke the DMA master, so to access the data memory, conduct CGRA related operations, and also to write to the NoC.

2.5 Hardware/Software Integration

The HardWare/SoftWare (HW/SW) integration is implemented by addressing special-purpose registers and dedicated shared-memory locations in the overall architecture. The addressing and data transfers are carried out by implementing special Application Programming Interfaces (APIs) which can be repetitively used

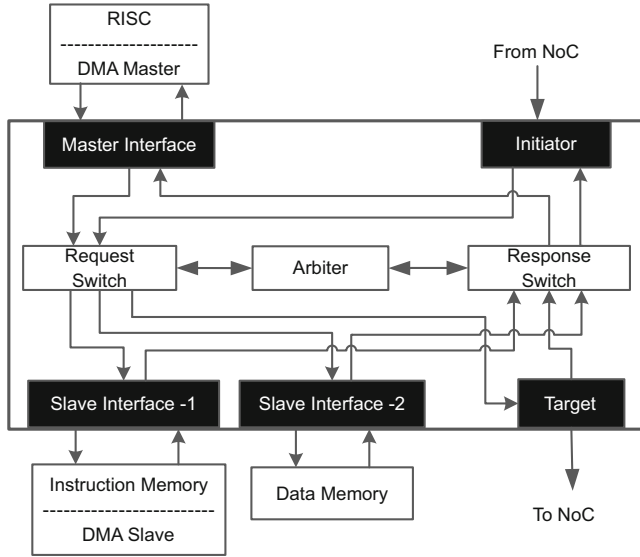


Fig. 2.2 An NoC node interfaced with RISC core or the DMA device [13]

by the programmer for algorithm mapping. The HW/SW integration is divided into three layers fundamentally required for processing, i.e., Loading the Configuration and Data, Execution, and Synchronization. The general description of APIs related to these procedures is presented in the following subsection.

2.5.1 Loading Configuration and Data

The major part of the configuration stream required for processing is generally loaded at system startup time. A configuration word is composed of address and data fields. The configuration can be injected straight into a CGRA and based on the address fields, it reaches the destination PEs after passing through a pipelined distribution infrastructure.

1. **Configuration Transfer:** The API developed for this task requires the base addresses of the configuration stream in the main memory and the configuration memories of the CGRA. Since the data and configuration memories operate at the same frequency across the platform, there is one word per clock cycle transfer between the nodes, therefore no particular synchronization protocol is required.
2. **Data Transfer:** The data transfer between two nodes is either between the data memories of the respective nodes or if the DMA of an associated CGRA transfers the data from the local memories of the CGRA to the data memory of another node. During the system startup time, the data from the supervisor node's

(RISC processor) data memory is fetched and sent to the data memory of a CGRA node. As explained in the previous point, this step does not need any synchronization, but the data from the node's memory has to be placed in a time-multiplexed manner in the local memories of the CGRA for custom processing. This task may require several clock cycles. To conduct this task, the supervising RISC processor sends a control word to the DMA of the associated CGRA node which starts reading the data from the memory and sends the data to the local memories of the CGRA. The control word contains the source and destination addresses, the number of read and write cycles, and information related to time-division multiplexing for the placement of data to be processed. When an internal data transfer is in progress within a node, i.e., between the node's data memory and the local data memories of the CGRA, the supervising RISC node should wait for the current data transfer to complete before invoking another internal data transfer through the DMA.

2.5.2 Context Enabling and Execution

As soon as the configuration and data transfer complete, the supervising RISC node enables CGRA execution by sending a control word for cycle-accurate processing. The control word contains information for the CGRA, i.e.,

1. Direction of data flow for processing, that is from the first local data memory of the CGRA to the other or vice-versa;
2. The context to be enabled;
3. Read and write latency, cycles, and offsets;
4. Total number of operation cycles.

During this task, different user-designed contexts are enabled in a sequence for a specified number of clock cycles for the custom computing of the target application.

2.5.3 Synchronization

There are two types of data transfers: (1) between data memories of NoC nodes and (2) between the node's data memory and the local memory of the CGRA and vice-versa. The data memories of the NoC nodes operate on the same clock frequency, so there is no specific synchronization required. The transfer rate is about one word per clock cycle after a latency equal to the number of nodes encountered in the transfer path. The transfer between the node's data memory and the data memory of the CGRA is established by the RISC core which sends a control word to the DMA device tightly integrated with the CGRA. The control word is packed with cycle-accurate information that enables data transfer in a specific time-multiplexed pattern. This internal transfer may require several clock cycles and the

RISC processor must not invoke another DMA transfer for this node, unless the current one is active. During this time, the RISC processor can send commands to other nodes and can be busy with subsequent tasks. The RISC processor before invoking a DMA transfer within a node sets a reserved location in its data memory. As soon as the DMA transfer completes, the DMA writes to the NoC to reset the same reserved location in the data memory of the supervising RISC processor. Meanwhile, the RISC processor continuously monitors the reserved location and as soon as it finds it being reset, it assigns another DMA transfer for the same node.

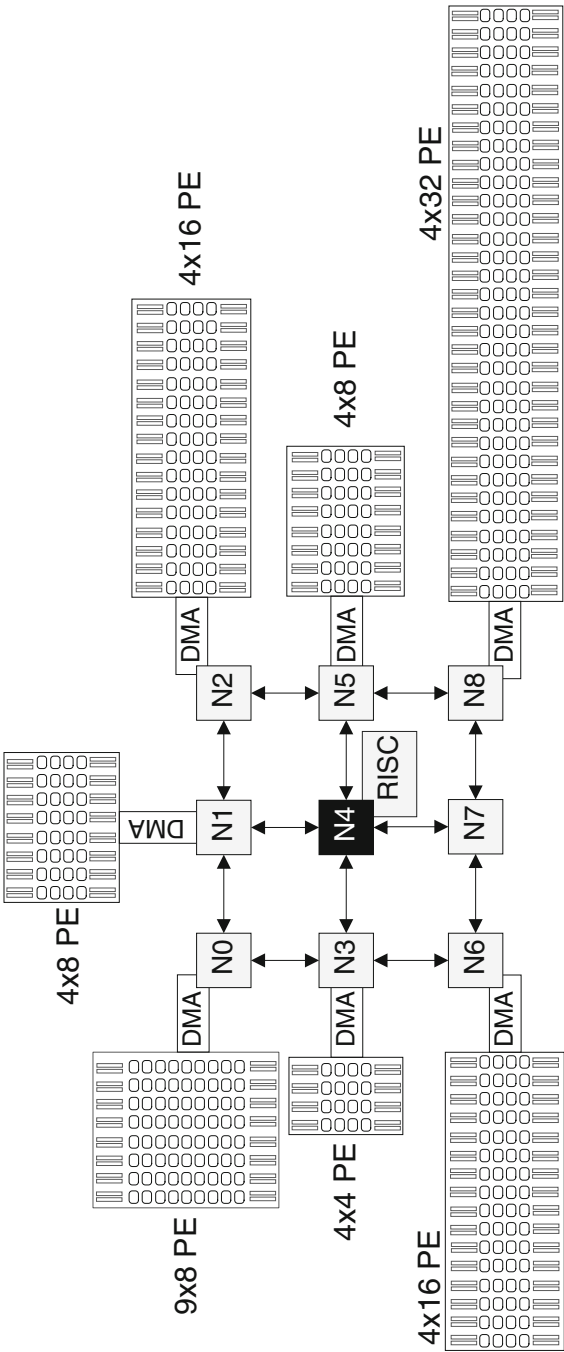
2.6 Heterogeneous Accelerator-Rich Reconfigurable Platform

Currently, HARP template has been used to instantiate four different instances. The first instance is for the proof-of-concept where each application-sized CGRA integrated with the node performs a computationally intensive signal processing task, i.e., 64- and 128-point FFT, 64th-order complex Matrix-Vector Multiplication (MVM), and 32nd-order real MVM. This instance is composed of six CGRA nodes and one central master node integrated with the RISC processor. Node-7 is not instantiated in this design to prove the template feature of the platform.

The second instance is relatively a realistic design problem, that is processing of four different streams of 64- and 128-point of FFT. The third instance is a hybrid architecture where the middle horizontal layer of the NoC is integrated with RISC cores and rest of the nodes are interfaced with CGRAs of application-specific sizes. This hybrid architecture was motivated to organize a distributed control among the RISC processors so that the program workload distribution for the CGRAs can be divided over the three processors. In this way, the tasks related to the distribution of configuration and data streams can be made faster than the instance presented in Fig. 2.3. The hybrid instance of HARP is shown in Fig. 2.4. The layered architecture of this instance is favorable for 3D IC integration. Simulation results have shown that the RISC processors are active most of the time, therefore contributing a major part to the average dynamic power dissipation. In this context, the layer of the RISC cores can be spatially closer towards the heat-sink and the airflow than the layers of the 3D IC containing the CGRAs.

Another instance of HARP was subjected to Dynamic Frequency Scaling (DFS) while considering the computational workload to be equally distributed over all the CGRAs. In this context, the central supervising node integrated to the RISC processor continuously monitors the execution time of each CGRA and then issues instructions to fine tune the operating frequency of the CGRAs until they approach the same execution-time constrained as a user-stated goal. This task is achieved by implementing a feedback control system in both HW and SW. The HW part constitutes a dedicated frequency control register and software part includes the API addressing the register as soon as the frequency of a particular CGRA needs

Fig. 2.3 A proof-of-concept HARP instance [13]



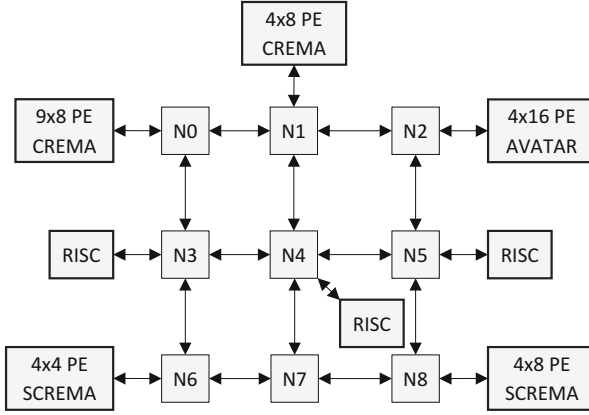


Fig. 2.4 A hybrid instance of HARP [14]

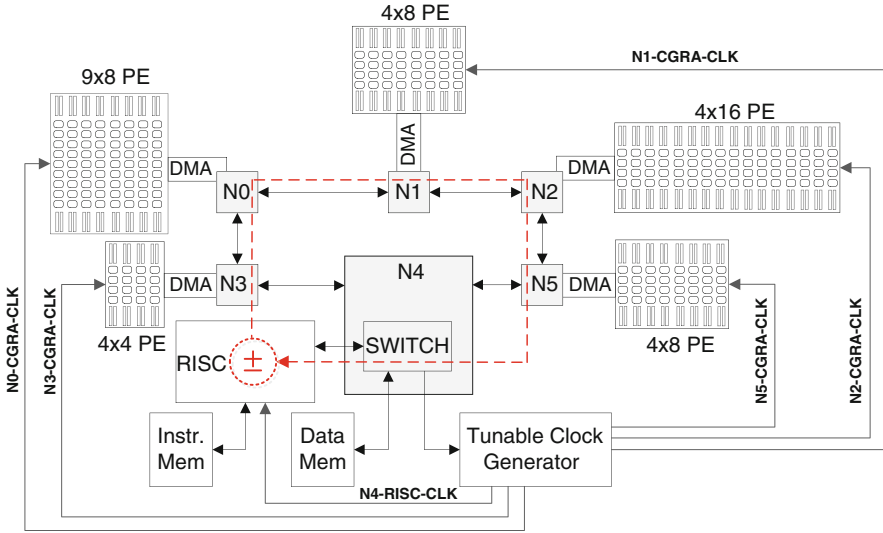


Fig. 2.5 An instance of HARP subjected to DVFS for workload balancing [13]

to be updated. The CGRAs are performing signal processing related tasks similar to the ones mentioned for the first instance of HARP. Figure 2.5 shows the HARP instance being subjected to DFS using a feedback control loop specified in dashed red colored lines. A larger CGRA is designed to process a large amount of data and its overall execution time is relatively lower than the smaller CGRAs. However, if all the CGRAs over the platform are supposed to exchange data with each other, they are required to process their respective tasks in the same time frame which is needed by the most work-loaded CGRA. In this case, the RISC processor while identifying the worst performing CGRA tunes the clock frequency of other CGRAs, so that all

the CGRAs perform closer to the performance of the worst-case. The DFS applied during this process shows that this particular workload balancing scheme mitigates power dissipation. Figure 2.5 also shows a frequency tunable clock generator which is a piece of non-synthesizable VHDL, but when actually implemented on a printed circuit board, or an FPGA, Phase-Locked Loop (PLL) based clock generators need to be adopted.

2.7 Application Mapping

The programming of an instance of HARP can be broadly categorized at two different levels. The first one is at the level of the CGRA, where the application data flow is organized based on the spatial location of the processing elements and establishing the interconnections between them. The second is purely writing the software for the application program flow and control. It includes calling the API functions with specific parameters to address the DMA and the CGRA control registers for cycle-accurate processing. The configuration streams are generated by the GUI tool (mentioned in Sect. 2.3) upon completion of the design of a context which can be loaded at the system start time. In fact, for any instance of HARP, all the configuration streams related to integrated CGRAs are loaded before the overall system starts processing. For any multi-core platform, it is important to demonstrate that the cores can exchange data with each other and also process the data simultaneously and independently of each other. The proof of this concept is implemented so that the CGRA integrated at Node N0 of HARP instance depicted in Fig. 2.3 processes 64-point FFT in radix-4 scheme and then transfers the results to the data memory of node N1. The DMA engine in node N1 fetches this data and sends it to the local data memory of node N1 CGRA which subsequently processes 64th-order complex MVM on the 64-point vector data obtained from FFT processing. The node N1 transfers the results of complex MVM processing back to node N4. The transfer and processing of data between these nodes show that nodes can exchange data with each other for dependent processing. The other nodes N2, N3, N5, N6, and N8 are simultaneously processing the data and independently of each other. The node N2 CGRA processes 128-point FFT in mixed-radix(2, 4) scheme while rest of the CGRA nodes are processing integer MVM of different orders. The second instance of HARP was processing only FFT for IEEE-802.11n demodulator case which requires four different streams of 128-point data to be processed simultaneously. In this case, nodes N1, N2, N5, and N8 are instantiated with mixed-radix(2, 4) FFT CGRAs to perform this task. The details of application mapping on these first two instances of HARP are well described in [13]. Table 2.1 shows the type of the CGRA integrated with a node of HARP and the application mapped on it. The table is a combined representation of the first two instances of HARP, i.e., the homogeneous (hom) and the heterogeneous (het).

The third instance of HARP is a hybrid platform which has a flavor of both homogeneous and heterogeneous cores. In it, the nodes N3, N4, and N5 are

Table 2.1 Types and sizes of the CGRAs integrated with the nodes of HARP and the applications mapped on them [13]

Node	PEs	CGRA type	Kernel
HARP-het-N0	8×9	CREMA	Radix-4, 64-point FFT
HARP-het-N1	4×8	CREMA	Complex MVM, 64th-order
HARP-het-N2	4×16	AVATAR	Radix-(2, 4), 128-point FFT
HARP-het-N3	4×4	SCREMA	Real MVM 32nd-order
HARP-het-N5	4×8		
HARP-het-N6	4×16		
HARP-het-N8	4×32		
HARP-hom-(N0, N2, N6, N8)	4×16	AVATAR	128-point FFT

In the table, hom and het stand for homogeneous and heterogeneous, respectively

integrated with COFFEE RISC cores and rest of the nodes are integrated with CGRAs of application-specific sizes. The three RISC cores communicate with each other to establish distributed supervisory control among all the CGRA cores. The RISC at node N3 supervises the CGRAs at node N0 and N6, the N4 does it for N1 and N7 CGRAs while the N5 performs supervision for N2 and N8 CGRAs. The synchronization and control are established as the RISC cores write and read flags in each others shared-memory locations. Meanwhile, the CGRAs on the completion of a task write acknowledgments in specific memory locations of their supervising RISC cores. The advantage of using the three RISC cores is the simultaneous configuration transfer to all the cores which otherwise in the case of the first instance of HARP is the responsibility of a single RISC core—a situation where the NoC bandwidth was under-utilized. The application mapping in this instance of HARP is similar to the two instances and the CGRA nodes are processing different lengths and types of FFTs, complex and real MVM [13, 14].

The fourth instance of HARP was designed as a six core platform organized in a geometry of two rows and three columns. The nodes N0, N1, N2, N3, and N5 are integrated with CGRAs while the node N4 contains the RISC core acting as a supervisor. The platform is subjected to Dynamic Frequency Scaling (DFS) in an effort to equalize performance on all the CGRAs, thereby mitigating instantaneous dynamic power dissipation. The RISC has a special-purpose counter which counts the cycles elapsed for the execution of an algorithm by the CGRA and stores the measurement in the data memory. This counter is not the part of the main datapath of the processor, so if the RISC processor is busy in general-purpose tasks, its counting remains unobstructed. Once the clock-cycle count is measured for all the CGRA integrated to the HARP platform, then the supervising RISC processor searches for the largest number (worst-case). Based on this worst-case, the RISC sends the data to its frequency control register which updates the operating frequencies of all the CGRAs other than the worst performing. This process is made iterative so in the next iteration, the clock-cycle count is measured for all the CGRAs again and fine tuning is performed in form of a feedback loop. In this way, the CGRAs gradually approach the performance of the worst-case. Once all the CGRAs degrade

their execution time, the instantaneous power consumption is reduced, therefore the overall heat dissipation. This form of DFS is implemented for the proof-of-concept when all the cores on a platform are supposed to exchange data with each other and it is not a requirement if one core needs to operate faster than the other. However, the overall operating frequency control system is software-defined which allows any custom power mitigation technique to be implemented in software.

2.8 Measurement and Estimation

This section presents the numbers related to different performance metrics for all the four instances of HARP platform, i.e., clock cycles required to perform data transfers and execution, resource utilization on the FPGA device, synthesis frequencies and estimates related to power dissipation and energy consumption. Table 2.2 presents clock cycles required for transfer between the data memories of NoC nodes and from the data memory of an NoC to one of the local data memories of the CGRA. The table also shows the cycles required for execution by the CGRA.

Table 2.3 shows the DSP resource consumption on the FPGA at the node level. The Stratix-V FPGA device has 18-bit DSP elements and to implement one 32-bit DSP element, two 18-bit DSP resources are required. For example, in the first row of the table, 12 32-bit multipliers consume 24 DSP resources. It can also be observed, the DSP resource consumption increases as the complexity of algorithm increases, for example, radix-(2, 4) FFT accelerator requires 56 DSP elements than the 24 required by the radix-4 FFT algorithm.

Table 2.4 shows the resources required by the first two instances of HARP, i.e., hom and het on the FPGA device. As it has been discussed before that the HARP-het and HARP-hom are the instances with 7 and 4 nodes, respectively, but the Adaptive Logic Module (ALM) consumption by homogeneous version is significantly higher

Table 2.2 The number of clock cycles for data transfer and execution at different stages measured with reference to clock frequency of the supervising node [13]

Node-to-node	D. mem to D. mem	D. mem to CGRA	Trans. total	Exe. total
het(N4-N0)	1017	659	1676	420
het(N0-N1)	1036	446*	1482	457
het(N1-N4)	–	448*	–	–
het(N4-N2)	2042	1033	3075	571
het(N4-N3)	75,622	–	75,622	728
het(N4-N5)				609
het(N4-N6)				340
het(N4-N8)				211
hom(N4- (N0, N2, N6, N8))	4× 2042	4× 1033	4× 3075	587

D. Mem, Trans., and Exe. stand for Data Memory, Transfer, and Execution, respectively

* Sign shows data transfer from CGRA to node's data memory

Table 2.3 DSP resource consumption on the FPGA device at the node level for both of the versions of HARP [13]

Node	Accelerator type	Multipliers(32-bit)	DSPs
<i>HARP het</i>			
Node-0	Radix-4 FFT	12	24
Node-1	Complex MVM	12	24
Node-2	Radix-(2, 4) FFT	28	56
Node-3	Real MVM	4	8
Node-4	RISC Core	6	12
Node-5	Real MVM	8	16
Node-6	Real MVM	16	32
Node-7	–	–	–
Node-8	Real MVM	32	64
Total	–	118	236
<i>HARP hom</i>			
Node-0	Radix-(2, 4) FFT	28	56
Node-2	Radix-(2, 4) FFT	28	56
Node-4	RISC Core	6	12
Node-6	Radix-(2, 4) FFT	28	56
Node-8	Radix-(2, 4) FFT	28	56
Total	–	118	236

Table 2.4 Resource utilization by HARP instances on Stratix-V FPGA device (SSGXEA4H1F35C1) [13]

<i>HARP het</i>		
ALMs	78,845/158,500	50 %
Registers	64,436/634,000	10 %
Memory bits	21,000,928/38,912,000	54 %
DSP	236/256	92 %
<i>HARP hom</i>		
ALMs	120,823/158,500	76 %
Registers	67,475/634,000	10.6 %
Memory bits	13,679,616/38,912,000	35 %
DSP	236/256	92 %

than heterogeneous version. The ALM consumption depends on the complexity of the algorithm mapped on the CGRA and in comparison, the complexity of radix-(2, 4) FFT algorithm is the highest and having four of those makes the hom version to have the largest number of ALMs despite the fact that it has less number of CGRAs integrated than the het version.

The first two instances of HARP operate on a single clock source. The HARP-het version synthesizes at 214.64 MHz at 0 °C and 186.22 MHz at 85 °C while the functional voltage is set to 900 mV. These measurements were observed for slow timing model. The fast time model shows 274.65 MHz at 0 °C and 251.0 MHz at 85 °C. The HARP-hom instance synthesizes for slow timing model at 208.42 MHz at 0 °C and 196.89 MHz at 85 °C with the functional voltage equal to 900 mV. The operating frequencies for fast timing model are observed to be 299.31 MHz at 0 °C

and 272.63 MHz at 85 °C. The HARP-hom processes four streams of 128-point FFT within $587 \text{ CC} \times 1/186.22 \text{ MHz} = 3.15 \mu\text{s}$ while considering the operating frequency of 186.22 MHz and the cycles presented in Table 2.2, therefore satisfying FFT execution-time constraints of IEEE-802.11n standard.

Table 2.5 shows the dynamic power and energy consumption by the CGRAs integrated with the nodes of HARP-hom and -het versions. It is evident that a larger CGRA in size has more logic to toggle per unit time and therefore dissipates more power but offers lower execution times as in the case of processing 32nd-order MVM. As energy is the product of power dissipation and execution time, so in this specific case, the dynamic power dissipation does not appear to deviate considerably.

2.9 Evaluation and Comparisons

The HARP instances are synthesized for a 28 nm FPGA device. According to a document produced by Altera Corporation, the average operating frequency of a Stratix FPGA device scales by 20 % as the process size scales [3], which means that a 28 nm FPGA is capable to operate with a 20 % higher operating frequency than a 40 nm FPGA device. On the other hand, Xilinx, Inc claims to provide 50 % higher operating frequency for the synthesized designs with their latest 28 nm FPGA device in comparison to their 40 nm one, but such a performance comes at the cost of increasing the on-chip resources by 2× [24]. The designer can implement 2× parallel resources to achieve 2× speed-up. Since this argument is out of context, therefore Altera’s scaling trend can also be estimated for Xilinx devices as well. A 90 nm ASIC implementation shows 3.4× to 4.6× (average 4.0×) higher operating frequency than an equivalent FPGA implementation as well as consumes 14× lesser dynamic power dissipation [15]. Based on this information, cross-technology comparisons are established in comparison to a HARP instance as shown in Table 2.6. The state-of-the-art platforms selected for comparisons are already described in Sect. 2.2.

The node N0 of HARP-het version computes 64-point FFT in $420 \text{ CC} \times 1/100.0 \text{ MHz} = 4.2 \mu\text{s}$. NineSilica computes the same task in $10.3 \mu\text{s}$ on a 40 nm FPGA using all of its RISC nodes. Since HARP-het is synthesized on a 28 nm FPGA device, scaling on NineSilica’s performance is required to establish comparisons. After scaling, NineSilica is estimated to compute the same task in $8.25 \mu\text{s}$, hence HARP-het shows a 1.96× gain in comparison.

A homogeneous MIMD MPSoC delivers a performance of 19.2 GOPS while being synthesized on a 90 nm FPGA. After appropriate scaling to establish comparisons, the performance estimate increases to 26.88 GOPS. HARP-het performs at 40.8 GOPS and shows 1.51× of gain in comparison.

A homogeneous MPSoC platform called P2012 is synthesized on a 28 nm CMOS technology. The performance gap ratios discussed above between an ASIC and FPGA implementation are supposed to remain similar even between a 28 nm ASIC

Table 2.5 Estimation of dynamic power/energy for each CGRA integrated in a node for both of HARP instances [13]

Node CGRA size Other HW	Algorithm (Type)	Dynamic power (mW)	Active time (μ s)	Dynamic energy (μ J)
<i>HARP-het</i>				
N0 8 $\times$ 9 PE	Radix-4 FFT (64-point)	129.68	4.20	0.54
N1 4 $\times$ 8 PE	Complex MVM (64-Order)	116.75	4.57	0.53
N2 4 $\times$ 16 PE	Radix-(2, 4) FFT (128-point)	213.12	5.71	1.21
N3 4 $\times$ 4 PE	Real MVM (32nd order)	61.55	7.28	0.44
N4 RISC	General flow control	59.73	–	–
N5 4 $\times$ 8 PE	Real MVM (32nd order)	98.97	6.09	0.60
N6 4 $\times$ 16 PE	Real MVM (32nd order)	175.80	3.40	0.59
N7	–	0.37	–	–
N8 4 $\times$ 32 PE	Real MVM (32nd order)	327.36	2.11	0.69
NoC integration	–	7.12	–	–
Logic	–	302.07	–	–
Total	–	1492.52	–	4.6
<i>HARP-hom</i>				
N0 4 $\times$ 16 PE	Radix-(2, 4) FFT (128-point)	214.68	5.87	7.11
N2 4 $\times$ 16 PE	Radix-(2, 4) FFT (128-point)	214.35		
N4 RISC	General flow control	59.75		
N6 4 $\times$ 16 PE	Radix-(2, 4) FFT (128-point)	212.97		
N8 4 $\times$ 16 PE	Radix-(2, 4) FFT (128-point)	213.89		
NoC integration	–	0.67		
Logic	–	295.64		
Total	–	1211.95		

Table 2.6 Multiple performance metric comparison with HARP Implementation at 100.0 MHz on a 28 nm FPGA [13]

Platform/technology	Performance metric	Platform's value (PV)	Scaled PV	HARP-het	
				Value	Gain
NineSilica [2] /FPGA 40 nm	FFT execution time (μ s)	10.3	Exe. time – Exe. time $\times$ 20% SU_fTfs $= 10.3 - 10.3 \times 0.2 = 8.24$	4.2	1.96 $\times$
[5] /FPGA 90 nm			PV $\times$ 40 % SU_fTfs $= 19.2 \times 1.4 = 26.88$	40.8	1.51 $\times$
P2012/ [18]		19.2	PV $\times$ SD_afTs $\times$ Ps		
CMOS 28 nm	GOPS/mW	0.04	$= 0.04 \times 1/4 \times 1/2 = 0.005$	0.0153	3.0 $\times$
MORPHEUS /[23]			(PV $\times$ SD_afTs $\times$ Ps) $\times$ 60 % SU_fTfs $= (0.02 \times 1/4 \times 1/2) \times 1.6 = 0.004$	0.0153	3.8 $\times$
CMOS 90 nm	GOPS/mW	0.02			

In the table, SU_fTfs, SD_afTs, and Ps stand for Speed-Up for FPGA to FPGA scaling, Speed-Down for ASIC to FPGA scaling, and Power scaling, respectively

and a 28 nm FPGA implementation. The gap shows that performance degrades by a factor of 4 but the dynamic power dissipation increases by a factor of 14 on an FPGA implementation than the ASIC for the same process size. Considering the worst-case scenario, where the static power can be equal to dynamic power dissipation, the factor of 14 can be almost equal to 2. In such a situation, the scaled performance of P2012 is estimated to be 0.005 GOPS/mW as shown in Table 2.6 while HARP shows a performance gain of $3.0\times$.

The 0.02 GOPS/mW performance of MORPHEUS at 90 nm CMOS can be first scaled for a 90 nm FPGA and then be scaled down to a 28 nm FPGA to establish comparison with HARP. The scaling is also shown in the last row of Table 2.6. In comparison, HARP shows a performance of $3.8\times$.

2.10 Conclusions

In this chapter, an extra large-scale template-based homogeneous/heterogeneous accelerator-rich architecture is presented which is capable to process massively parallel computationally intensive signal processing related algorithms. In particular, three instances from the architecture's template were generated in which one is completely heterogeneous, the other one homogeneous, and the third one is hybrid in nature. All of these instances were tested for a proof-of-concept to verify the functionality and architectural features. From the discussion presented in the overall and particularly in the section of evaluation and comparisons, the platform outperforms many state-of-the-art platforms in several performance metrics due to its high density in the number of on-chip computational resources and efficient utilization of the communication channel, i.e., a Network-on-Chip.

References

1. Ahonen, T., ter Braak, T.D., Burgess, S.T., GeiBler, R., Heysters, P.M., Hurskainen, H., Kerkhoff, H.G., Kokkeler, A.B.J., Nurmi, J., Raasakka, J., Rauwerda, G.K., Smit, G.J.M., Sunesen, K., van Zonneveld, H., Vermeulen, B., Zhang, X.: CRISP: cutting edge reconfigurable ICs for stream processing. In: Cardoso, J.M.P., Hübner, M. (eds.) *Reconfigurable Computing*, pp. 211–237. Springer, New York (2011). ISBN: 978-1-4614-0060-8
2. Airolidi, R., Garzia, F., Anjum, O., Nurmi, J.: Homogeneous MPSoC as baseband signal processing engine for OFDM systems. In: *International Symposium on System on Chip (SoC)*, 2010, pp. 26–30, Sept 2010. doi:10.1109/ISSOC.2010.5625562
3. Altera Product Catalog. Release Date: July 2014, Version 15.0, p. 2 (2015). www.altera.com
4. Bell, S., Edwards, B., Amann, J., Conlin, R., Joyce, K., Leung, V., MacKay, J., Reif, M., Liewei, B., Brown, J., Mattina, M., Chyi-Chang, M., Ramey, C., Wentzlaff, D., Anderson, W., Berger, E., Fairbanks, N., Khan, D., Montenegro, F., Stickney, J., Zook, J.: TILE64 - processor: a 64-Core SoC with mesh interconnect. In: *IEEE International Solid-State Circuits Conference (2008)*. ISSCC 2008. Digest of Technical Papers, pp. 88–598, 3–7 Feb 2008

5. Bonnot, P., Lemonnier, F., Edelin, G., Gaillat, G., Ruch, O., Gauguet, P.: Definition and SIMD implementation of a multi-processing architecture approach on FPGA. In: *Proceedings of Design, Automation and Test in Europe (DATE '08)*, pp. 610–615. ACM, New York, NY (2008)
6. Campi, F., Deledda, A., Pizzotti, M., Ciccarelli, L., Rolandi, P., Mucci, C., Lodi, A., Vitkovski, A., Vanzolini, L.: A dynamically adaptive DSP for heterogeneous reconfigurable platforms. In: *Proceedings of Design Automation and Test in Europe (DATE '07)*, pp. 9–14. EDA Consortium, San Jose, CA (2007)
7. Campi, F., König, R., Dreschmann, M., Neukirchner, M., Picard, D., Jüttner, M., Schuler, E., Deledda, A., Rossi, D., Pasini, A., Hübner, M., Becker, J., Guerrieri, R.: RTL-to-layout implementation of an embedded coarse grained architecture for dynamically reconfigurable computing in systems-on-chip, SOC 2009. In: *International Symposium on System-on-Chip, 2009*, pp. 110–113, 5–7 Oct (2009)
8. Dennard, R., Gaensslen, F.H., Rideout, V.L., Bassous, E., LeBlanc, A.R.: Design of ion-implanted MOSFET's with very small physical dimensions. In: *JSSC*, Oct (1974)
9. Garzia, F., Hussain, W., Nurmi, J.: CREMA, a coarse-grain re-configurable array with mapping adaptiveness. In: *Proceedings of 19th International Conference on Field Programmable Logic and Applications (FPL 2009)*. Prague, IEEE, New York (2009)
10. Garzia, F., Hussain, W., Airolidi, R., Nurmi, J.: A reconfigurable SoC tailored to software defined radio applications. In: *Proceedings of 27th Norchip Conference*, Trondheim, NO (2009)
11. Hussain, W., Garzia, F., Ahonen, T., Nurmi, J.: Designing fast Fourier transform accelerators for orthogonal frequency-division multiplexing systems. *J. Signal Process. Syst. Signal Image Video Technol.* **69**, 161–171 (2012). Springer, Berlin
12. Hussain, W., Ahonen, T., Nurmi, J.: Effects of scaling a coarse-grain reconfigurable array on power and energy consumption. In: *2012 International Symposium on System on Chip (SoC)*, pp. 1–5. Tampere (2012). doi:[10.1109/ISSoC.2012.6376372](https://doi.org/10.1109/ISSoC.2012.6376372)
13. Hussain, W., Airolidi, R., Hoffmann, H., Ahonen, T., Nurmi, J.: HARP2: an X-scale reconfigurable accelerator-rich platform for massively-parallel signal processing algorithms. *J. Signal Process. Syst.* **85**, 341–353 (2015). Springer, New York
14. Hussain, W., Hoffmann, H., Ahonen, T., Nurmi, J.: Design of a hybrid multicore platform for high performance reconfigurable computing. In: *Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC)*, pp. 1–8. Oslo (2015). doi:[10.1109/NORCHIP.2015.7364376](https://doi.org/10.1109/NORCHIP.2015.7364376)
15. Ian, K., Rose, J.: Measuring the gap between FPGAs and ASICs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **26**(2), 203–215 (2007). doi:[10.1109/TCAD.2006.884574](https://doi.org/10.1109/TCAD.2006.884574)
16. Lodi, A., Mucci, C., Bocchi, M., Cappelli, A., De Dominicis, M., Ciccarelli, L.: A Multi-context pipelined array for embedded systems. In: *International Conference on Field Programmable Logic and Applications*, Aug 2006. FPL'06, pp. 1–8 (2006)
17. Mattson, T.G., Riepen, M., Lehnig, T., Brett, P., Haas, W., Kennedy, P., Howard, J., Vangal, S., Borkar, N., Ruhl, G., Dighe, S.: The 48-core SCC processor: the programmer's view. In: *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC '10)*. IEEE Computer Society, Washington, DC (2010)
18. Melpignano, D., Benini, L., Flamand, E., Jego, B., Lepley, T., Haugou, G., Clermidy, F., Dutoit, D.: Platform 2012, a many-core computing accelerator for embedded SoCs: performance evaluation of visual analytics applications. In: *Proceedings of 49th Annual Design Automation Conference (DAC '12)*, pp. 1137–1142. ACM, New York, NY (2012)
19. Taylor, M.B.: Is dark silicon useful: harnessing the four horsemen of the coming dark silicon apocalypse. In: *Proceedings of the 49th Annual Design Automation Conference (DAC-2012)*, pp. 1131–1136. ACM, New York (2012)

20. Taylor, M.B., Kim, J., Miller, J., Wentzlaff, D., Ghodrat, F., Greenwald, B., Hoffman, H., Johnson, P., Lee, J.-W., Lee, W., Ma, A., Saraf, A., Seneski, M., Shnidman, N., Strumpen, V., Frank, M., Amarasinghe, S., Agarwal, A.: The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *Micro IEEE* **22**(2), 25–35 (2002)
21. Venkatesh, G., Sampson, J., Goulding, N., Gracia, S., Bryksin, V., Martinez, J.L., Swanson, S., Taylor, M.B.: Conservation cores: reducing the energy of mature computations. In: *ASPLOS'10*, pp. 205–218 (2010)
22. Voros, N.S., Rosti, A., Hübner, M.: Flexeos embedded FPGA solution. In: *Dynamic System Reconfiguration in Heterogeneous Platforms. Lecture Notes in Electrical Engineering*, vol. 40, pp. 39–47. Springer Netherlands, Berlin (2009). ISBN 978-90-481-2426-8
23. Voros, N.S., Hübner, M., Becker, J., Kühnle, M., Thomaitiv, F., Grasset, A., Brelet, P., Bonnot, P., Campi, F., E. Schüler, H. Sahlbach, S. Whitty, R. Ernst, E. Billich, C. Tischendorf, U. Heinkel, F. Ieromnimon, D. Kritharidis, A. Schneider, J. Knaeblein, W. Putzke-Röming. MOR-PHEUS: a heterogeneous dynamically reconfigurable platform for designing highly complex embedded systems. *ACM Trans. Embed. Comput. Syst.* **12**(3), 33 pp. (2013).Article 70
24. Wu, X., Gopalan, P.: Xilinx next generation 28 nm FPGA technology overview. White Paper: 28nm Technology, July 23 Version 1.1.1, p. 5 (2013). www.xilinx.com

Computing Platforms for Software-Defined Radio

Hussain, W.; Nurmi, J.; Isoaho, J.; Garzia, F. (Eds.)

2017, XII, 240 p. 19 illus., 9 illus. in color., Hardcover

ISBN: 978-3-319-49678-8