

Embedded High Performance Computing

ABDUL RASHEED A.

Lecturer in Electronics Engineering,
Govt. Polytechnic College, Chelakkara, Kerala.

Abstract:

We evaluate the use of High-Performance Computing in the Design Space Exploration of a complex highly parameterized Very Long Instruction Word based System-on-a-Chip platform in this paper. Experiments show that the conventional belief of a linear decrease in exploration time as the number of available processors increases is debunked beginning with a relatively low number of processors, owing primarily to communication overhead and I/O bottleneck. Because well-known HPC systems use personal computers or servers, implementing HPC as a curriculum in universities may consume a lot of resources. Using PCs as practical modules necessitates a large amount of resources and space.

Keywords:

High Performance Computing, Embedded System, HPC, Embedded Computing, Embedded HPC, HPC Accelerators,

I. Introduction:

High-performance embedded computing provides high computing levels for mission-critical applications that must operate in harsh and small environments. These rugged embedded systems are essential in defence and aerospace applications that require the highest level of computing resource reliability.

HPEC brings supercomputing performance that was previously only available in data centres to dense, ruggedized systems that can be deployed in today's most rigorous defence applications. The sheer volume of data produced by Electro-Optical and Infrared (EO/IR), Electronic Warfare (EW), radar, and Signals Intelligence (SIGINT) sensors necessitates multiple processing elements and I/O interconnected via high-throughput, low-latency switched fabrics for analysis. Even small unmanned vehicles use sensors that generate large amounts of data, necessitating high performance embedded computing capability at the network's edge. This necessitates difficult size, weight, and power requirements to meet the environmental demands of a rugged vehicle. [1]

Embedded HPC

In recent years, there has been a surge in the production of embedded electronics, many of which contain extremely powerful microprocessor designs that outperform the capabilities of desktop computers available ten years ago. Smartphones are a common example, but there are hundreds more. Computer vision, automotive systems, UAV (drones), security cameras, and network security are just a few of the applications for embedded electronics.

Embedded HPC is a response to the growing demand for computational power in embedded applications. These applications necessitate the presence of a CPU and an FPGA on the same board, if not in the same chip. As a result, Aldec's TySOM boards are built around the Xilinx Zynq family of MPSoCs and chipsets that integrate an ARM Programmable System and an FPGA. [2]

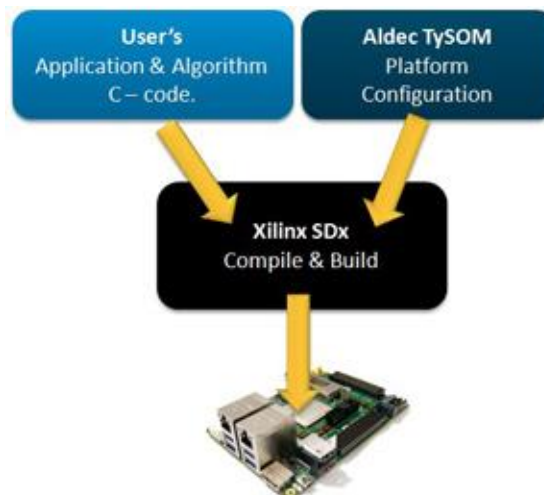


Table 1: Embedded HPC Accelerators

	TySOM-3A (ZU19EG)	TySOM-3(ZU7)	TySOM-2(7Z100)
Logic Cells	1143000	504,000	444,000
DSP Blocks	1,968	1,728	2,020
On-chip RAM	70.6 Mb	38 Mb	25.5 Mb
PS ARM Subsystem Spec.	CPU: Quad-core ARM Cortex-A53 Dual-core ARM Cortex-R5 ARM Mali™-400 MP2	CPU: Cortex-A53 Quad core RT CPU: Cortex-R5 Dual Core GPU: Mali-400 H.264/H.265 Integrated Codec	CPU: Cotext-A9 Dual-Core

Design Automation

Aldec offers TySOM System Platforms that can be loaded into the Xilinx SDx tool to help with the design of embedded HPC applications. The SDx C/C++ algorithm files are then compiled and converted to RTL code suitable for FPGA. Finally, the RTL code is automatically connected to the system platform of choice. [3] When all of the compilation and implementation steps are completed, running an embedded application is as simple as uploading files generated by SDx to an SD card and starting execution on the TySOM board.



Reference Designs and Services

For a quick start, Aldec provides several reference designs demonstrating the use of various peripherals and programmable logic (FPGA) to accelerate application algorithms and working embedded Linux implementations. If your organisation recognises the benefits of FPGA-powered embedded HPC but lacks hardware design "know-how," Aldec's custom engineering services can help. Our extensive experience in FPGA hardware design and verification can be used to quickly build a complete system or integrate your algorithms with ready-to-use reference designs. [4]

Main Features

- A portfolio of TySOM boards to meet a variety of requirements
- Compatibility with complementary FMC Daughter-cards
- A complete design and verification environment
- Close integration with Xilinx SDx and Vivado software
- High end and solid hardware based on Aldec's 30+ years of experience

Solution Contents

- Aldec TySOM board
- TySOM reference designs bundle
- Riviera-PRO HDL simulator (option)
- Technical documentation, tutorials and white papers
- SDx hardware platform package (Board Support Package - BSP)
- Custom engineering for RTL Porting Services

II. Review of Literature:

HPC infrastructure and simulation environment To evaluate and compare the performance indices of different architectures for a specific application, the architecture must be simulated while the application's code is running. The Epic-Explorer simulation environment is used in this work. Both the compiler and the simulator must be retargetable in order for architectural exploration to be possible. Trimaran [5] provides these tools and thus serves as the foundation for Epic-Explorer. Epic- Explorer is an environment that not only allows any instance of a platform to be evaluated in terms of area, performance, and power, but also implements various design space exploration techniques. The parameterized system architecture used in this work is based on HPL-PD, a parametric processor meta-architecture designed for research in EPIC/VLIW instruction-level parallelism.

Table 1: On an Opteron 2.6 GHz Linux Workstation, the simulation time (compilation + execution) for several multimedia benchmarks was measured.

Benchmark	Description	Input size (KB)	Evaluation time (sec)
wave	Audio Wavefront computation	625	7.7
g721-enc	CCITT G.711, G.721 and G.723 voice compressions	8	25.9
jpeg-codec	jpeg compression and decompression	128	33.2
mpeg2-dec	MPEG-2 video decoding	400	143.7
adpcm-enc	Adaptive Differential Pulse Code Modulation speech encoding	295	22.6
adpcm-dec	Adaptive Differential Pulse Code Modulation speech decoding	16	20.2
fir	FIR filter	64	9.1

The computation effort required for a single evaluation (i.e. simulation) of a single system configuration for several media and digital signal processing application benchmarks is reported in Table 1.

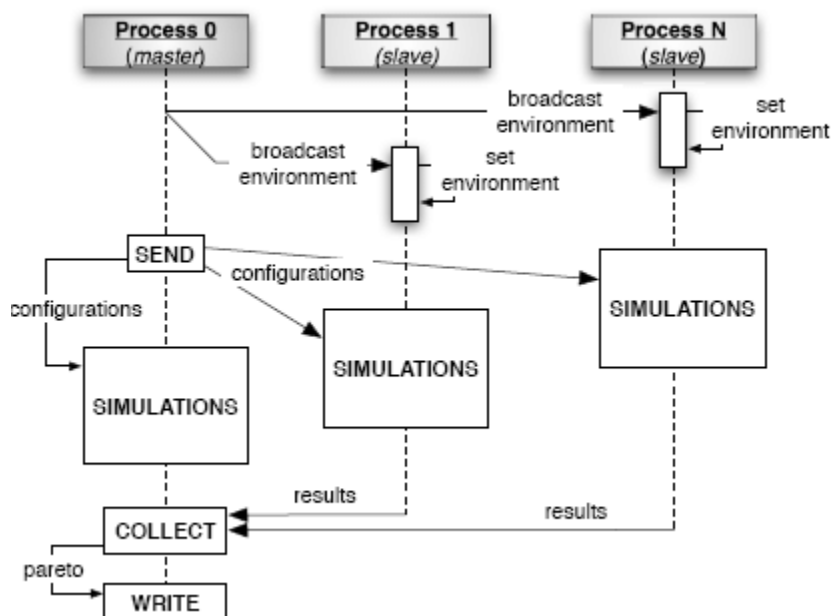


Figure 1: Exploration flow on HPC environment.

The programme ran correctly on 1, 2, and 4 processors, but when we needed 8 processors, it crashed, reporting communication problems. Analysing the issue, we discovered that it was caused by an inconsistency in the environment the job discovered on different hosts. In fact, in our setup, up to four processes can run on a single host, and only five or more require the use of a second host. [6]

Objectives:

1. Analysis of Embedded High Performance Computing
2. Define Embedded HPC
3. Describe Embedded HPC Accelerators
4. Simulation environment and the HPC infrastructure to evaluate and compare
5. Exploration flow on HPC environment.

III. Research Methodology:

In this work, we show the emerging application timing requirements, and we propose to exploit the probabilistic real-time theory to achieve the required time predictability. After a brief recap of the fundamentals of this methodology, we focus on its applicability to HPC systems to check their ability to satisfy such conditions. In particular, we studied the advantages of having heterogeneous processors in HPC nodes and how resource management affects the applicability of the proposed technique. In this paper we assess the use of High Performance Computing in Design Space Exploration of a complex highly parameterized Very Long Instruction Word based System-on-a-Chip platform.

IV. Result and Discussion:

In this section, we conduct a thorough evaluation of the proposed framework by exploring 1000 configurations of the parameterized VLIW-based system architecture under consideration. The simulated configurations were chosen at random from the design space in Table 2 and executed on the cluster infrastructure described in subsection.

The HPC Infrastructure

The following configuration was used for testing: 16 IBMLS21 Blades with two Opteron 2.6 GHz dual core processors (for a total of four cores per blade), eight gigabytes of DDR2 memory, and a 73GB SATA hard drive linked by Gigabit Ethernet. The cluster used for the tests was configured to allow one process per core (4 processes per host), and one host is reserved to run cluster services and manage jobs, so the maximum number of MPI processes that could be reached was $415 = 60$. [7-8]

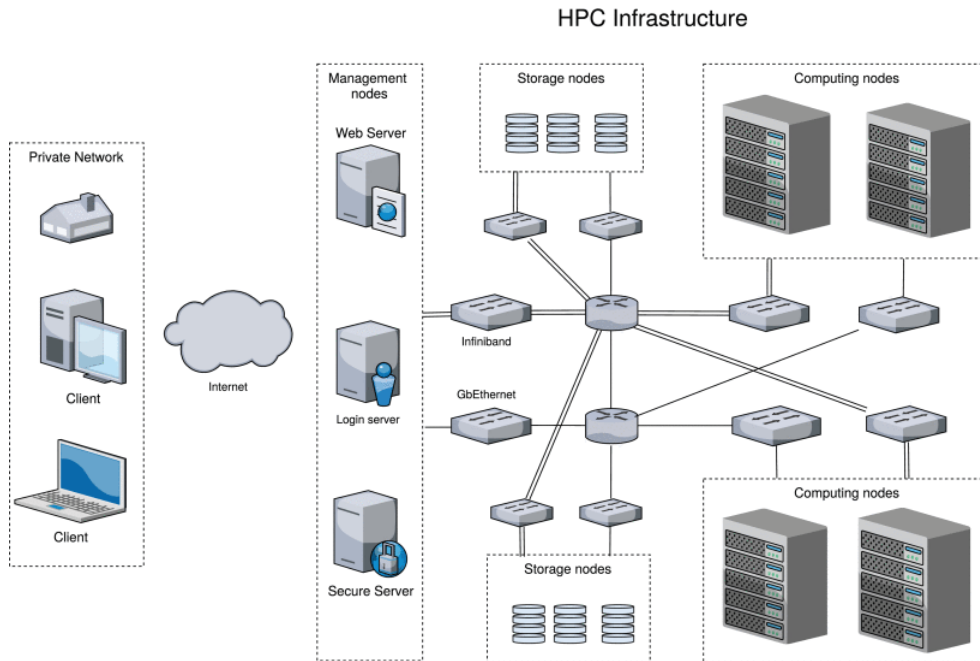


Table 2: Design space of the parameterized VLIW based system architecture.

Parameter	Parameter space
Integer Units	1,2,3,4,5,6
Float Units	1,2,3,4,5,6
Memory Units	1,2,3,4
Branch Units	1,2,3,4
GPR/FPR	16,32,64,128
PR/CR	32,64,128
BTR	8,12,16
L1D/I cache size	1KB,2KB,...,128KB
L1D/I cache block size	32B,64B,128B
L1D/I cache associativity	1,2,4
L2U cache size	32KB,64KB...,512KB
L2U cache block size	64B,128B,256B
L2U cache associativity	2,4,8,16
Space size	7.739×10^{10}

The software package was installed on the cluster coordinator's dedicated host. The batch queue manager mirrored the package per-job via SCP on all hosts involved in the parallel computation.

Because the tests were conducted without interference from other jobs in the cluster, the scheduler attempted to allocate the processes as close together as possible in order to fill one host before moving on to the next. This is advantageous because it reduces the number of software copies required.

Figure 2 depicts the total time required for an exploration requiring the simulation of 1000 configurations on an increasing number of processors. The total amount of time is normalised by the amount of time required on a single processor. As shown in Figure 2, even with an increasing number of processors, the wall clock time was only reduced by an order of magnitude.

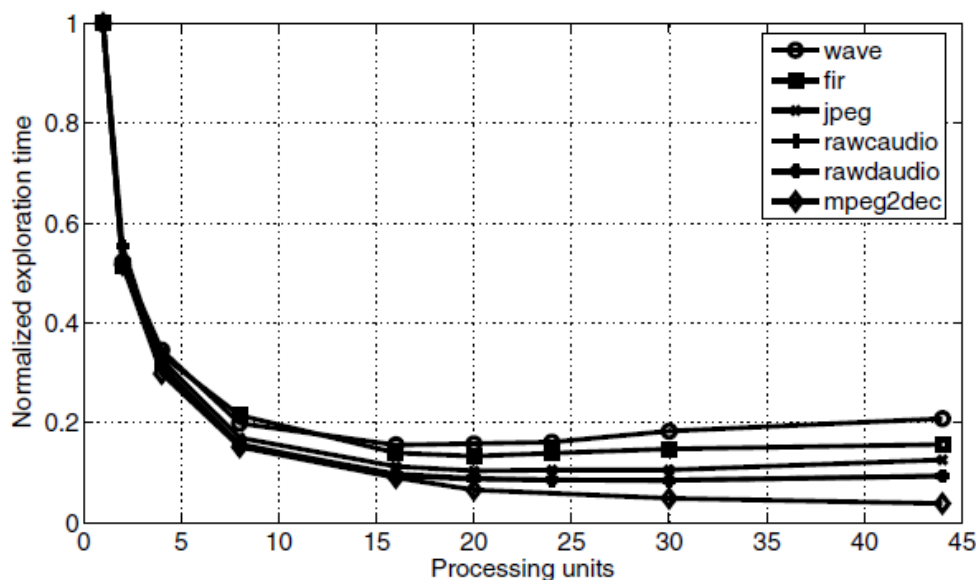


Figure 2: The time required to explore 1000 configurations with an exponentially increasing number of processors.

Increasing the number of processes quickly leads to computation time saturation. We have a very steep performance increase when switching from one to two processes, almost cutting exploration time in half. We still get good benefits from parallelization when using 4 or 8 processes, but we quickly reach saturation. The simpler benchmarks, such as wave, start to perform worse, whereas the more complex ones, such as mpeg2dec, still have some minor advantages by employing as many as 60 processes. [9-10]

V. Conclusion:

We presented a case study of design space exploration of a complex highly parameterized VLIW-based SoC platform in this paper. The platform's 18 free parameters cover a design space of over 109 system configurations. Even with a few seconds of evaluation time for each configuration, exhaustive exploration would take hundreds of years on a single machine. We put High-Performance Computing (HPC) to the test as a viable solution to DSE-related problems.

References:

- [1]. G. Ascia, V. Catania, M. Palesi, and D. Patti. EPIC-Explorer: A parameterized VLIW-based platform framework for design space exploration. In First Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia), pages 65–72, Newport Beach, California, USA, Oct. 3–4 2003.
- [2]. V. Kathail, M. S. Schlansker, and B. R. Rau. HPL-PD architecture specification: Version 1.0. Technical report, Compiler and Architecture Research HP Laboratories Palo Alto HPL-93-80, 2000.
- [3]. K. Keutzer, S. Malik, R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: Orthogonalization of concerns and platform-based design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 19(12):1523–1543, Dec. 2000.
- [4]. P. Nsame and Y. Savaria. A customizable embedded soc platform architecture. In 4th IEEE International Workshop on System-on-Chip for Real-Time Applications, pages 299–304.
- [5]. An infrastructure for research in instruction-level parallelism. <http://www.trimaran.org/>.
- [6]. F. Vahid and T. Givargis. Platform tuning for embedded systems design.
- [7]. *IEEE Computer*, 34(3):112–114, Mar. 2001
- [8]. K. Ikegami, H. Noguchi, C. Kamata, M. Amano, K. Abe, K. Kushida, E. Kitagawa, T. Ochiai, N. Shimomura, S. Itai et al., “Low power and high density stt-mram for embedded cache memory using advanced perpendicular mtj integrations and asymmetric compensation techniques,” in 2014 IEEE International Electron Devices Meeting. IEEE, 2014, pp. 28–1.
- [9]. K. Dichev and A. Lastovetsky, "Optimization of collective communication for heterogeneous HPC platforms", *High-Performance Computing on Complex Environments*, pp. 95-114, May 2014.
- [10]. H. Hussain, S. U. R. Malik, A. Hameed, S. U. Khan, G. Bickler, N. Min-Allah, et al., "A survey on resource allocation in high performance distributed computing systems", *Parallel Comput.*, vol. 39, no. 11, pp. 709-736, 2013.
- [11]. S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges", *J. Grid Comput.*, vol. 14, no. 2, pp. 217-264, Jun. 2016.