Grenze International Journal of Engineering and Technology, June Issue

Reconfigurable Computing: Inception, Enhancement and Impenitent

Vikram Singh Soni[1], Neha Tripathi[2] and Vikas Rathi[3]
[1-3]Graphic Era (Deemed to be) University, Dehradun, India
Email: vikramsinghsoni@geu.ac.in, nehagarg.it@geu.ac.in, vikas.rth@gmail.com

***Abstract*— Reconfigurable refers to the ability to rearrange and adapt to new circumstances, problems, or modifications. A design, configuration, or model that satisfies this kind of need is called reconfigurable computing. A standard called reconfigurable computing allows hardware to be dynamically adjusted to meet the needs of embedded devices in the future. It makes use of a variety of hardware, including microcontrollers, general purpose processors (GPPs), application-specific integrated circuits (ASICs), processors for Reduced Instruction Set Computers (RISCs), and cores for digital signal processing (DSP). The most adaptable hardware design for reconfigurable computing (RC) is the Field Programmable Gate Array (FPGA). This can be quickly changed to effectively do particular jobs. It draws attention to the benefits of reconfigurable hardware, which include its adaptability, speed, and energy economy. Reconfigurable computing provides a special way to maximize system performance and resource usage by reprogramming the hardware to match the demands of a particular job. real-world uses for reconfigurable computing, such as scientific simulations, image and video processing, signal processing, cryptography, and machine learning.**

I. INTRODUCTION

A computing paradigm known as "reconfigurable computing [1]" makes use of hardware architectures that may be dynamically altered to accommodate various computational demands. Reconfigurable computing systems are different from standard fixed-function hardware like CPUs and GPUs in that they may be reprogrammed or changed to meet the unique needs of a particular computation. This includes exploring new interconnect structures, configurable logic block designs, memory architectures, and exploring trade-offs between flexibility, performance, power consumption, and area utilization. A programmable interconnect and a matrix of customizable logic blocks (CLBs) make up the field-programmable gate array (FPGA)[2], a semiconductor device that is essential to reconfigurable computing. It is possible to program FPGAs to carry out parallel computations and create unique digital logic circuits. The adaptability of FPGAs to various algorithms and processing tasks is made possible by their reconfigurability during runtime. Reconfigurable computing bridges the gap between ASICs and microprocessors by combining the benefits of microprocessor software programmability and ASIC execution speed. Microprocessors, improving software performance without sacrificing the programmability of the hardware. Field Programmable Gate arrays and other reconfigurable devices are made of programmable logic blocks that may be loaded with configuration information. With the help of these logic blocks, the proper routing is completed to create the entire circuit and ultimately carry out the calculations. Although it presents certain difficulties, reconfigurable computing enables the introduction of new flexible computing systems. The first is the

Grenze ID: 01.GIJET.10.2.400

productivity gap: creating hardware tailored to a particular issue is not the same as producing software that is intended for a standard computer architecture. The disparities result from the underlying processes and the amount of time needed to complete a design iteration phase; hardware synthesis takes a long time, whereas software is built fairly quickly. The speedier adoption of reconfigurable computing is impeded by this productivity gap. Thankfully, a number of strategies are being developed to close the gap: An approach called High-Level Synthesis aims to increase the abstraction level at which hardware is built. It's akin to the paradigm change in software design that occurred with the introduction of the first high-level languages [3], which made it unnecessary to develop software exclusively in assembly language.

II. Reconfigurable Computing

Within a host CPU, reconfigurable functional units can be achieved using reconfigurable hardware. This makes it possible to integrate personalized instructions that can alter over time to a typical programming environment. Under the direction of a primary microprocessor, the reconfigurable units operate as functional units. The operands for input and output are stored in registers. Consequently, a hardware component that can be reconfigured and a software-programmable core processor make up the fundamental design of. The application's sequential operations are carried out by the core processor, which also manages data transfers between programmable hardware and data memory.
The reconfigurable hardware's sole purpose is to take use of the applications algorithm's parallelism. Usually, this hardware is made up of several programmable, interconnected components. A unique setup program known as the context determines the elements' interconnectedness and operation.

III. Reconfigurable Assemblies

The widespread commercial availability of Field-Programmable Gate Arrays (FPGAs) in the late 1980s gave rise to reconfigurable computing, which is the use of programmable logic to accelerate processing. Conventionally, there have been two methods for implementing various set of rules in a secure manner: using microprocessors or Application Specific Integrated Circuits (ASICs). microprocessor executes the computation by executing a set of instructions and offers programmability of software instructions. Microprocessors have a drawback in that they must constantly retrieve instructions, interpret them, and then carry them out. Consequently, it results in execution overhead for every action. Since ASICs are made for specific or particular functions, they execute computations efficiently. The disadvantage of ASICs is that once they are manufactured, they cannot be changed.
Field Programmable Gate arrays(FPGA) and other reconfigurable devices are made of programmable logic blocks[4] that may be loaded with configuration information. With the help of these logic blocks, the proper routing is completed to create the entire circuit and ultimately carry out the calculations.

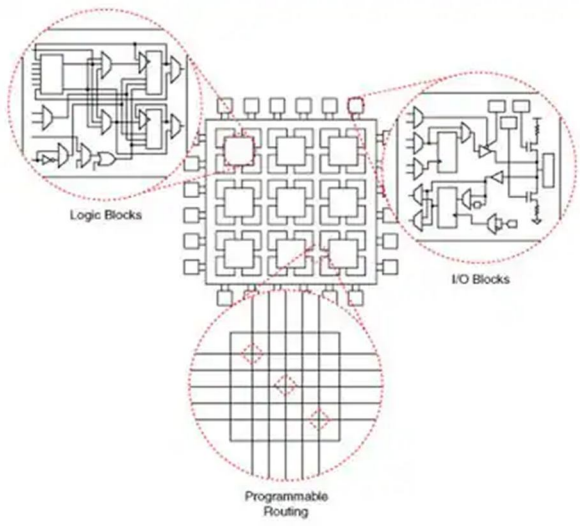

Figure 1*:* The fundamental FPGA architecture (Image Source: National Instruments)

IV. Usage Of Reconfigurable Systems

Reconfigurable computers have been used[5] in a wide range of fields, including bioinformatic, networking, digital signal processing, digital image processing, information coding, space and solar applications, and computers and communications security.

A. Coding of facts and figures

To increase decoder[6] performance, run-time dynamic reconfiguration is applied in response to shifting channel noise conditions. Simulation has been used to determine the decoder's implementation parameters.

B. Digital Signal Processing

A great deal of research has been done in the field of digital signal processing (DSP)[7], including RC hardware implementations of adaptive digital filters, discrete Fourier transform (DFT), FIR filters, and elliptic 2R filters.

C. Bioinformatics

Some work was done on a computer architecture for high-speed voxel data processing. A high-performance computer architecture designed for bioinformatics[8] volume data processing applications requiring a lot of computing power.

D. Networking

In order to assess this system's efficiency in comparison to the current IP routing implementation—which mostly consists of multiple-homed PCs and physical routers—IP routing algorithms are mapped onto it, and its performance is assessed.

E. Defence Safeguard

FPGAs and other reconfigurable devices are very appealing choices for hardware security, particularly when it comes to cryptographic methods. They offer the ease of implementation of a broad variety of algorithms along with the flexibility of a dynamic system.

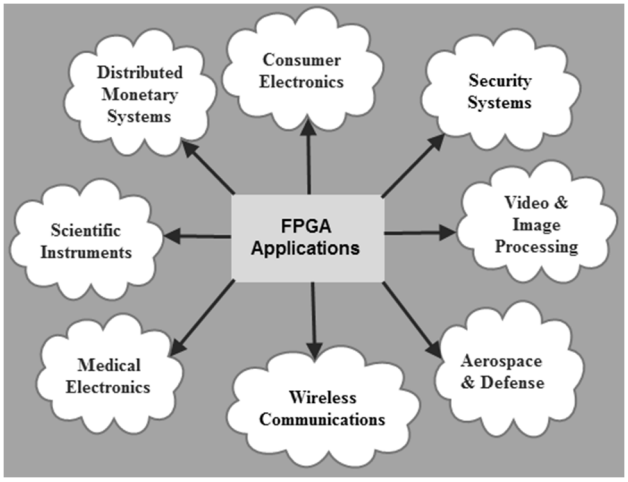

Figure 2: Application of FPGA

V. Implications And Advantages

Large data processes work better with FPGAs. However, since they necessitate sequential access, computations involving huge hash tables exclude the use of FPGA parallelism[8].

The programming of FPGAs is typically done with HDL (Hardware Description Language), which necessitates an extensive background and understanding of circuit design for the developer. Unlike traditional ones, the sequence of events that take place during compilation is distinct.

It is difficult to migrate application software designed for sequential machines to FPGAs, which results in higher performance speed increases than with prior versions. Restructuring the algorithm or application code is typically necessary.

VI. Present-Day And Forthcoming Evolution

Reconfigurable computing systems imprint the circuit onto the hardware and alter it according to the application, combining the advantages of both ASICs and microprocessors. The reconfigurable hardware can be simply mapped to the computer-intensive elements of the program to speed them up. They can be employed with different variation designs ranging from reconfigurable hardware with the host microprocessor to an independent processing unit. Operating systems may employ reconfigure computing in the future to offer capabilities like load balancing, self-adaptive core monitoring, and patching without requiring a system reboot.

VII. Conclusion

Numerous configurable systems that are currently in use have given us various insights into how to address various issues in the design of a configurable system. To identify the best solutions, we have attempted to categorize the issues and conduct a case study visit. Some of the conclusions drawn from these approaches are set out.

Reconfigurable speed: Since FPGA-based technology is used in the majority of the options provided, their reconfiguration speed is constrained. To minimize the possibility of a reconfiguration bottleneck, some of them, however, allow the reconfiguration of one FPGA while the others continue to operate. The addition of partially reconfigurable FPGAs to traditional FPGAs can further enhance FPGA-based systems.

Dimensions, velocity, and compactness: The most common programmable circuit technology, FPGA, currently limits the speed, size, and compactness of these circuits; trade-offs between speed and capacity must be negotiated. On the other hand, a sizable proportion of systems just use FPGA for prototyping, therefore a lot of semicustom built solutions should be available soon.

Interface and memory assemblies: Systolic array-based algorithms are among the most used algorithms on configurable systems. Greater complexity in operations results in longer execution times, which reduces the requirement for memory access. However, substantial parallelism exploration can be accomplished. In these situations, a single port memory access is insufficient due to the growing need for memory. The most popular solution to this issue is to achieve a multiplied memory access by dividing up the memory among the processing components.

Tools for developing applications: The automated development of applications and the investigation of parallelism are two important factors that have contributed to the industry's and designers' growing interest in configurable systems. None of the systems have demonstrated these factors at a level that is adequate as of yet. Some systems have attempted to speed up executions by removing loops, frequently used conditions, and arithmetic expressions from the code and mapping them onto the customizable hardware, even though the majority of systems employ the hardware design paradigm to program the adjustable sections.

References

[1] K. Compton and S. Hauck, “Reconfigurable Computing: a Survey of Systems and Software”. In ACM Comput. Surveys, Volume 34, Issue 2, pp. 171-210, 2002.

[2] Guo, Z., Najjar, W., Vahid, F., Vissers. K, “A Quantitative Analysis of the Speedup Factors of FPGAs over Processors”, In 12th international symposium on Field programmable gate arrays, 2004, pp 162 - 170.

[3] D. Levi and S.A. Guccione, "BoardScope: A Debug Tool for Reconfigurable Systems, in Configurable Computing Technology and its use in High Performance Computing, DSP and Systems Engineering", In Proceedings. SPIE Photonics East, John Schewel, ed., SPIE - The International Society for Optical Engineering, Bellingham WA, November 1998.

[4] Abhay Sharma; Mridul Gupta; Tarun Rawat, “FPGA Implementation of Lattice-Wave Half-Order Digital Integrator using Radix-2r Digit Recoding”, 6th International Conference on Electronics, Communication and Aerospace Technology, IEEE, Pages 300-305, December 2022.

[5] S.A. Guccione and E. Keller, "Gene matching using JBits", In 12th International Field-Programmable Logic and Applications Conference, September 2002.

[6] M. Wirthlin and B. Hutchings. A dynamic instruction set computer. IEEE Workshop on FPGAs for Custom Computing Machines. 1995, 99-107.

[7] Abhay Sharma, Tarun Kumar Rawat, “Efficient Modeling and Simulation of Signal Processing Systems”, Book, “Advanced Numerical Simulation in Science and Engineering”, Pages 32-43, Publisher Amazon Publishing, 2020.

[8] Debasis Mitra, Debanjan Mitra, Mohamed Sabri Bensaad, Somya Sinha, Kumud Pant, Manu Pant, Ankita Priyadarshini, Pallavi Singh, Saliha Dassamiour, Leila Hambaba, Periyasamy Panneerselvam, Pradeep K Das Mohapatra, “Evolution of Bioinformatics and its impact on modern bio-science in the twenty-first century: Special attention to pharmacology, plant science and drug discovery”, Computational Toxicology, Pages 100-248, Elsevier, Publication date 2022/10/13.