Indian Journal of Science and Technology, Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/86696, December 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Operating System Design Challenges for a Reconfigurable Computing Environment

B. Abirami[1]*, K. Sridhar[2] and V. Vaidhyanathan[3]

[1]Department of CSE, M.A.M. School of Engineering, Trichy - 621105, Tamil Nadu, India; nbs.abirami@gmail.com
[2]Department of ECE, M.A.M.C.E.T, Trichy - 621105, Tamil Nadu, India; mkm.sridhar@gmail.com
[3]School of Computing, SASTRA, Thanjavur - 613401, Tamil Nadu, India; vvn@it.sastra.edu

Abstract

Objective: The primary objective of this proposal is to explore the areas of applications for a Reconfigurable computing system and further to analyze the challenges in designing an Operating System suitable for such a computing environment. **Method:** In this paper we have discussed at length the issues in designing an operating system which will practically be suitable for a Reconfigurable computing environment. The different challenges faced by an Operatin System in handling the applications executed by a conventional CPU are analysed. The advantages of using the Field Programmable Gate Array (FPGA) in Reconfigurable Computing System (RCS) to handle complex applications and to increase the efficiency and throughput of the system are also discussed. **Findings:** A primary comparison was made with all existing strategies used in Reconfigurable Computing Systems. The inferences obtained from this analysis gave a clear picture that the objective can be achieved by designing a new OS or making some modifications in the kernel of the OS. By achieving this task, multitasking between the applications or processes can be done frequently yielding a far better performance and throughput. The challenges faced by an OS in dealing with the applications executed by the RCS can be eliminated to a greater extent. **Improvements:** By adopting these kind of designing strategies for the OS in a system with the programmable gate array, there is a considerable improvement and performance of the device also the life span of the device is increased which is considered as most important criteria for any users in computing environment. The new architecture and performance analysis is a research prone area in many of the leading Universities.

Keywords: Design Challenges, FPGA, Operating System, RCS

1. Introduction

In conventional computing, an algorithm can be executed in any one of the two approaches. In one approach, an ASIC can be developed to execute the algorithm with high speed and low power. In this method, the development cost is very high and it is very difficult if not impossible to modify the circuit if the algorithm is modified. This approach is highly suitable for applications or products whose volume of production is very high. Examples are TV and mobile phones.

Another approach is to go for general purpose computing where the algorithm is converted into suitable software and executed. This approach is highly flexible because of the nature of the software and it is preferred method when the volume of production is less or the algorithm is modified frequently. The software approach also has the advantage of reduced time to market.

Reconfigurable computing is a new approach which combines the advantages of both approaches. It has a programmable hardware known as FPGA, in which the hardware equivalent of the algorithm can be mapped and it can be modified like software. Hence the FPGAs execute the algorithms as fast as a conventional hardware and provide the flexibility to modify the circuit as easily as software. A survey of the reconfigurable computing and the systems and software related to their development is given in[1]. The FPGA, which acts as a coprocessor can be interfaced with the main processor in loosely coupled or tightly coupled configuration. New

*Author for correspondence

FPGA architectures such as the time multiplexed FPGA[2], partially reconfigurable FPGA[3] and FPGA co processing boards, such as SPACE 2[4] are all capable of accepting multiple dependent or independent circuits[5].

This paper first introduces a Reconfigurable computing system, its advantages, and the challenges in the implementation and then discusses the modifications to be made in the conventional operating system to make it suitable for Reconfigurable computing system

2. Reconfigurable Computing

Reconfigurable computing is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware[1]. The FPGAs have been shown to accelerate a variety of applications. For example, in Block cipher algorithms such as AES, DES etc., both the data parallelism and fine grained data manipulation is well implemented in an FPGA. Other applications include data compression, signal processing and image processing.

A reconfigurable computing system generally consists of a general purpose processor, its peripherals and a programmable logic resource commonly known as Field Programmable Gate Array (FPGA). FPGAs consist of configurable logic cells arranged in two dimensional rows and columns and are interconnected by a programmable route matrix.

Compilers accept the circuit descriptions in a Hardware descriptive language and convert them into configurations for reconfigurable systems. The FPGAs are classified into single context and multi context FPGAs based on how the configuration information is stored and retrieved. When some applications need only a portion of the FPGA or when only a part of the configuration needs to be modified we go for partial reconfiguration technique. The process of reloading the FPGA with a new configuration during run time is known as Run Time Reconfiguration (RTR). This extends the usefulness of the reconfigurable logic. Since the FPGA resources are shared by a number of tasks or applications simultaneously and different portions of the FPGA have to be reconfigured during run time, the run-time management of resources should be efficient and fair. Therefore the operating system support is essential to design and implement an efficient Reconfigurable computing system with run rime reconfigurable ability.

Computationally intensive applications which are meant to be executed in FPGA are mapped as digital circuits in logic cells. The execution of applications in FPGA eliminates the fetch and decodes operations during run time. The concurrent execution of a number of applications increases the performance of the system. Even though FPGAs are not as fast as ASICs, the reprogrammable feature of FPGA extends the usefulness of the system to several applications.

Several reconfigurable computing models have been published in the literature. In the earliest model, the reconfigurable logic is implemented as a coprocessor board connected to the system bus of a sequential host processo[6]. This approach uses a simple interface but suffers from the low bandwidth between the host and the processor. In this approach the hardware and software co design is to be considered in advance. Another approach is to integrate the reconfigurable logic that is the configurable logic arrays onto the conventional process dies[7]. The development of systems composed entirely of reconfigurable logic and memory has been reported in[8,9]

3. Operating System Support

In any computing system the operating system plays a major role in controlling the operations. Hence in a Reconfigurable computing system also the operations of FPGAs have to be controlled by the operating system. A number of modifications have to be made in the conventional operating system to make it suitable for Reconfigurable computing system. The operating system support for FPGAs provides the following benefits;

- Greater convenience through abstraction, virtualization and generalization (whether the resource is embedded, attached, or integrated with microprocessor.
- Support for multi tasking.
- Improved system performance and fault tolerance.
- Integration of reconfigurable devices into global computational pool.

Earlier work on operating systems for reconfigurable computing is available in literature. In Xputer[10] an operating system which extends the host operating system to support multiple users has been presented. This OS achieves multi-user by computing circuits sequentially rather than concurrently. Brebner described the issues involved in managing virtual hardware resource[11]. He

proposed decomposing reconfigurable computing applications into Swappable Logic Units [SLUs], which describe circuits of fixed area and input/output (I/O) interfaces, so that multiple independent tasks might share a single FPGA[11]. The fixed size of the circuits limits the designer's flexibility to design circuits and optimize the space available. Diesel[12] described a web based multi-user operating system that shared the SPACE. 2 architecture among eight simultaneous users. In this approach each user is assigned one out of the eight FPGAs and does not allow more than one circuit per FPGA. This does not support concurrent multi-user operations.

4. Services of an Operating System

An operating system handles the tasks such as loader, scheduler, virtual memory system, cache management, protection, inter process communication, and I/O. In this section they are briefly explored.

4.1 Loader

This is a fundamental service provided by a conventional OS for initiating the execution of the program. Loading a reconfigurable computing "program" means placing the circuit and embedded RAM on the FPGA and then routing its external I/O interface to either neighboring circuits or to a local or global communications bus

The main difference between a software and a reconfigurable hardware is that loading an application onto an FPGA implies immediate execution but loading a program on RAM is not so.

4.2 Scheduler

Yet another important issue on conventional OS is Scheduling. Typically an application being executed by the Reconfigurable Hardware does not go through the fetch, decode and execute cycles. Hence, scheduling of these applications is different from the conventional mechanisms. As there is no predefined point of completion time for applications, most of the FPGAs do not delete a running application preemptively. So, while developing an OS, the designer of the application can provide a well defined signal for completion of applications or for segmented applications, the algorithms doing the segmentation can be set to insert the signals at the cut point.

4.3 Virtual Memory

The concept of virtual memory for a reconfigurable computing system is quite similar to that of a conventional computing system. The Operating system is able to deal with applications exceeding its capacity and it decides the select applications to be executed by FPGA. The proximity of solving the problem which arises when this allocation of applications to FPGA takes place dynamically during runtime is very less. Because, the area on the FPGA and the area which is preoccupied by other circuits could not be decided in advance for a dynamic partitioning strategy. Wider investigations have been made on the issues of special and temporal localities in this context[5].

4.4 Circuit Protection

Circuit protection is common in all operating systems. But for a system with FPGA, it means that the circuits cannot be interconnected beyond the applications. Studies reveal that the conventional methods are sufficient for this issue.

4.5 Cache Management

Managing of cache memory is yet another issue for the OS. The FPGA circuits can be organized to be in a hierarchy[13] and then another hierarchy of RAM can augment for storage of temporary configurations. Similar architectures of FPGA working at different speed grades can be arranged like this and the most critical applications can be allotted to faster speed FPGAs and less time critical applications can be assigned to lesser speed FPGAs.

Further investigations are to be carried out in this arena to evolve a suitable architecture satisfying the constraints mentioned above.

4.6 Inter Process Communication

Inter process communication in a Reconfigurable Computing system needs to incorporate suitable interfaces which will agree between different applications. This arrangement is similar to that of the system call interface in the conventional systems. The sequencing of operations can be planned in such a way that the final result can be downloaded from the FPGA by directly accessing the FPGA.

The Operating System which will handle the challenges for a Reconfigurable computing environment can be assumed to have the following structure as given

in Figure 1. It can be clearly inferred that the operating system solves the issues for a conventional computing environment very easily. If the system is slightly modified for a single user FPGA, then the design can be done completely offline because all the circuits and resources are at the disposal of a single user. So, while reaching the execution stage, the operating system would have solved the issues of allotting the tasks to the conventional CPU and FPGA. If the discussion is for a multi user FPGA, then, designing of such a system cannot be completed offline because the demand and usage of circuits and resources will be varying dynamically. Hence, the operating system will be finding it extremely difficult to allocate the applications unless it applies the concept depicted in the figure, that is, partitioning, placement and routing mechanisms. The routing of pre routed applications at the hardware level will further simplify the job of the Operating System.

4.7 Reconfigurable Hardware Operating Systems

Despite the challenges in designing and implementing a Reconfigurable hardware Operating System, it does have few benefits. The productivity and portability are increased as the computationally intensive applications are handled by the FPGA circuits.

The system can be easily partitioned among the different hardware circuits or between the hardware and the software as shown in Figure 2. If an application, if found to consume more of the CPU's time, it can be offloaded to FPGA to improve the system's performance. Without spending much time, the repartitioning can be used for balancing the load. Also, easy interaction between the applications or between the hardware and software makes

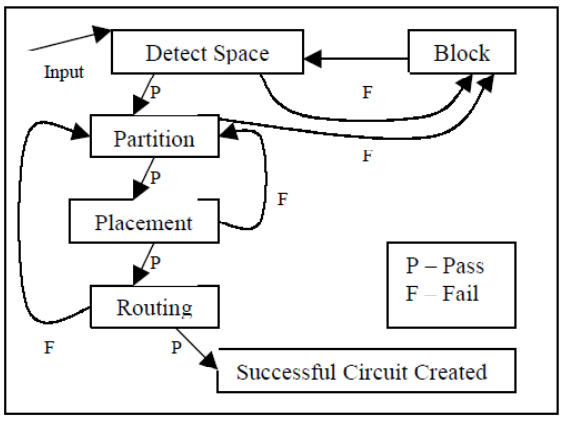

Figure 1. Placement and routing task.

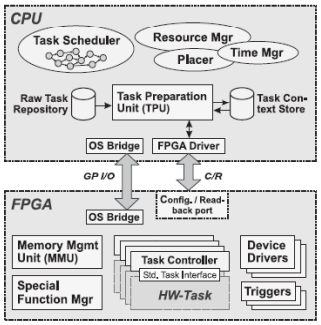

Figure 2. A system with CPU and FPGA.

it easy to locate the bugs and fix them easily. Further, exploring the challenges or issues in an operating system for a reconfigurable computing environment has been undertaken as a research prone area in the leading universities of the world. The latest research works have been concentrating the areas of using an operating system for RCS as multiprocessing or multicore systems[14].

Since the time the RCS was introduced, it has been custom specific for a single application for which the necessary resources can be pre planned at the design time itself. If we go for a system which can run many applications, predicting the required resources during the design stage is a very complex task because the resource requirements will vary dynamically based on the progress of the tasks and there is a possibility that the applications can be given by the users during run time.

According to Wenyin Fu, the operating system can extend its support for catering to the demands of all applications by adopting suitable scheduling algorithms which will fairly allot resources to all applications. Further, he has also suggested that the operating system can be made to access the memory locations of the reconfigurable hardware in order to decide the allocation of resources and computing time for each application[15]

Another interesting work has been accomplished by Haydon Kwok in the area of Reconfigurable Computing Environment. According to his work, a kernel has been designed by name Berkely Operating System for Reprogrammable Hardware (BORPH). In this four FPGAs are connected in a single board where the FPGAs will be communicating with each other via a system bus as shown in Figure 3.

The BORPH kernel selects the FPGA for specific applications. As there is no swapping concept used in this system, at the maximum of four processes can be executed concurrently. Further researches can invite more

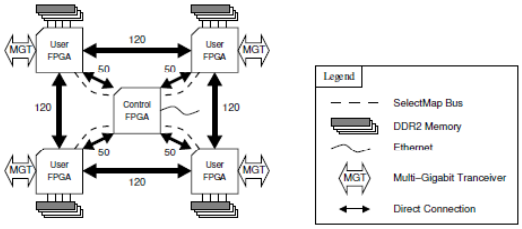

Figure 3. The BEE2 compute model in BORPH.

FPGAs to be placed on the same board to increase the concurrency and performance. Still there is a main bottleneck stated by the author is the communication between the user gateware design and the BORPH kernel[16].

5. Conclusion

From the above details, it has been made clear that a reconfigurable computing environment will outweigh the performance of a conventional computing system. The literature survey has thrown open the avenues for pursuing research in the areas of support to be rendered by the Operating System for implementing a perfect Reconfigurable Computing System. The possibility of developing a Reconfigurable Computing system, efficiently controlled by an Operating System can open new avenues in the embedded system domain.

6. References

1. Compton K, Hauck S. Reconfigurable Computing: A survey of Systems and Software. ACM computing surveys. 2002 Jun; 34(2):171–210.
2. Trimberger S. A time multiplexed FPGA. Proceedings of FPGAs for Custom Computing Machines, FCCM97, IEEE Computer Society. 1997 Apr 16-18; 22–8.
3. Xilinx X. C6200 Field Programmable Gate arrays, Xilinx, Inc., 1997 Apr 24; 1–73.
4. Gunther B. SPACE 2 as a reconfigurable stream processor. Proceedings of PART'97, the 4th Australian Conference on Parallel and Real-time Systems, Singapore. 1997 Sep. p. 286–97.
5. Wigley G, Kearney D. The first real operating system for Reconfigurable computers. 6th Australian Computing Science week (ACSAC'01); Gold Coast, Australia: IEEE Computer Society; 2001 Jan 29-30; 130–37.
6. Vuillemin JE, Bertin P, Roncin D, Shand M, Touati HH, Boucard P. Programmable active memories: Reconfigurable systems come of age. IEEE Transactions on Very Large Scale Integration (VLSI) SYSTEMS. 1996 Mar; 4(1):56–69.
7. Wittig RD, Chow P. One Chip: An FPGA processor with reconfigurable logic. IEEE symposium on FPGAs for custom computing Machines (FCCM'96); Los Alamitos, CA. 1996 Apr 17-19; 126–35.
8. Donlin A. Self modifying circuitry – A platform for tractable virtual circuitry. Proceedings of 8th International Workshop on Field Programmable Logic and Applications, From FPGA to computing Paradigm, FPL'98; 1998. p. 199–208.
9. Hauser JR, Wawrzynek J. Garp: A MIPS processor with a reconfigurable co processor. Proceedings of the 5th Annual IEEE Symposium on FPGAs for Custom Computing Machines (FCCM'97); Los Alamitos, CA. 1997 Apr 16-18; 12–21.
10. Hartenstein RW, Kress R, Nageldinger U. An operating system for custom computing machines based on xputer paradigm. Proceeding of the 7th International Workshop on Field Programmable Logic (FPL'97); 1997 Sep 1-3. p. 304–13.
11. Brebner G. A virtual Hardware Operating System for the Xilinx XC6200. Field Programmable logic: smart applications, New Paradigm and compilers. Proceedings of the 6th International Workshop FPL'96; 1996. p. 327–36.
12. Diessel O, Kearney D, Wigley G. A web based multi-user operating system for reconfigurable computing. Proceedings of the Parallel and Distributed Processing IPPS/SPDP'99; 1999. p. 579–87.
13. Kearney D, Kiefer R. A High Bandwidth I/O Architecture for a Reconfigurable Computer. Proceedings 4th Australasian Computer Architecture Conference; 1999 Jan.
14. Walder H, Platzner M. Reconfigurable Hardware Operating Systems: From Design Concepts to Realizations. Computer Engineering and Networks Lab, Swiss Federal Institute of Technology (ETH); Zurich, Switzerland. 2003.
15. Fu W. Operating System support for Reconfigurable Computing Systems [PhD Thesis]. University of Wisconsin, Madisson. 2010.
16. Kwok H. BORPH: An Operating System for Reconfigurable Computers [PhD Thesis]. University of California, Berkeley. 2007.