International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 31015-31022

http://www.ripublication.com

Programmable Hardware Scheduler for Reconfigurable MPSoCs

[1]P.M.Lalley and [2]Dr.T.Latha

[1]Narayanaguru College of Engineering, Manjalumoodu, Tamilnadu, India.
[2]St.Xavier's Catholic College of Engineering, Chunkankadai, Tamilnadu, India.

Abstract

Multiprocessor System on Chip (MPSoC) platform plays a vital role inparallel processor architecture design. However the growth of number of processing element in on chip, task decomposition and schedulingbecome major bottlenecks of MPSoCarchitecture. In this paper, we propose a programmable hardware scheduler which accomplish of both task decomposition andscheduling activities at run-time using NIOS II soft-core embedded processor. The NIOS II soft-core embedded processor was implemented in Altera cyclone III FPGA. The efficiency of programmablescheduler was evaluated using standard benchmark and its results are presented

Keywords :MPSoC, FPGA,NIOS II

1. Introduction

Due to the advantages of high performance and power efficiency, Multi-Processor Systems-on-Chip (MPSoCs) are nowadays widely used lot of applications such as networking, multimedia, and mobile devices. Moreover MPSoC offer better performance with lower energy consumption for complex and real time systems compared to uniprocessor embedded systems[1][2][3][4][5]. Recently, Intel andTilera proposed homogeneous MPSoCs with 80 and 100 PEs respectively, connected by a NoC. IBM, Sony and Toshiba proposed a heterogeneous MPSoC composed of one manager processor and 8 floating-point units. Future MPSoC architectures are anticipated to contain thousands of PEs in a single die.Typically MPSoC system composed of processing element, shared memory, scheduling unit and on chip interconnect[6]. The programmable processors and dedicated hardware IP cores are used as processing elements. Due to rapid increase of number processing element in on chip, task decomposition and scheduling become difficult challenges which

remarkably affect the efficiency of such systems[7][8][9].Typically MPSoC scheduling mechanisms has classified into static and dynamic scheduling. Static scheduling systems, control and synchronization information are embedded in the software[10][11][12][13]. In dynamic scheduling systems, a dedicated scheduler unit (which can be implemented in hardware, middleware or operating system) usually performs these activities[10][14], [15]. In contrast to dynamic scheduling approach; static scheduling approach has potential advantages to find more optimal solutions in comparison with dynamic scheduling. Because the programmer or compiler can see the entire application and can also compare different solutions but a dynamic scheduler must make a decision according to the available resources and pending tasks, instantaneously. On the other hand, the number of computational resources must be constant and predetermined in static scheduling. In order to overcome the above issue we propose runtime programmable hardware scheduler which use the structural hardware and software interfaces to perform all necessary activities for parallel execution of a program[16]. An online scheduling algorithm is exploited to perform both task decomposition and scheduling. The rest of the paper is organized as follows. First, we outline and summarize the related work in Sect. 2. Then Sect. 3 discusses about the proposed MPSoC architecture. Experiment setup and data analysis is presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2. **Related work**

In literature there are remarkable advantages of adaptive and dynamic MPSoCsystems have motivated many researches to work on novelarchitectures and techniques for the development of suchsystems, in recent years [17–14].Several operating systems have been developed for Multiprocessor reconfigurable computing systems those survey results are discussed here Ullmann et al. [17] presents a runtime system for an embedded processor with reconfigurable accelerators in a homogeneous shape and size, which was optimized for automotive applications.Broderson et al. [18] is based on a standard Linux kernel which provides software and hardware processes with the same UNIX interface and with access to the same OS services. BORPH was evaluated on the BEE2 platform.Patel et al. [19] present a framework for executing parallel applications on FPGA-based multiprocessors that employ multiple chips. They uses a distributed-memory model, where each computing engine is assigned its own local memory.Karanam et al. [20] target the Smith-Watterman algorithmusing 14 Microblazes for data processing and one Microblaze for the overall system control.Similarly, Mplemenosand Papaefstathiou [21] target the BLAST algorithm usingan MPSoC consisting of 80 Microblaze cores. In [22], which proposes an NIOS II-based multiprocessorsystem. The architecture is totally built for the occasion usinga pipeline approach, consisting of a chain of 15 processorsconnected point to point. They use a distributed-memoryarchitecture where each processor executes independenttasks, instead of parallelizing a unique task between thedifferent processors.

3. Proposed system

Functional block diagram of proposed MPSoC architecture is shown in figure which composed of programmable hardware scheduler and heterogeneous processing elements. Programmable scheduler unit composed of NIOS II embedded soft core processor which plays as kernel unit to perform task partition and scheduling.Scheduler is also charge of providing application running environment and programming interfaces to users. Processing unit composed of both embedded soft-core processor, hardware accelerator andcommunication controller are interconnect through avalonswitch fabric.

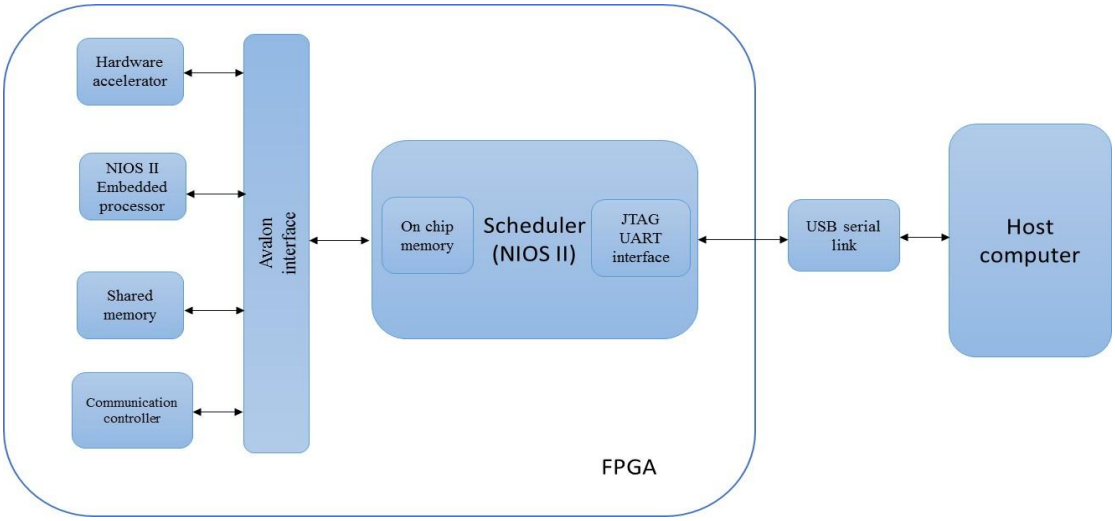


Fig. 1: Block diagram ofMPSoC architecture

Programmable hardware scheduler

The programmable hardware scheduler unit carryout three different tasks such as hardware tasks which can only run in hardware accelerator; software jobs that can only run on embedded processors; and hardware and software tasks those can either run on software or hardware. Although a same hardware and software tasks can either run on embedded processor or hardware accelerator, the partition and mapping operation are performed by scheduler servant automatically. Every function is bound to a type of software functions or hardware modules. Scheduler runs an online scheduling algorithm to detect data dependences automatically. If the tasks have no intertask data dependencies from each other, they will be sent to different processing elements in parallel. However, if different tasks running on different processors require data from each other, these tasks must be issued in sequence, which means scheduler has to wait until previous task is finished.

NIOS II embedded processor

The NIOS II embedded processor has Reduced Instruction Set Computer (RISC) architecture. Its arithmetic and logic operations are performed on operands in the

general purpose registers. It has three operating mode Supervisor mode, User mode and Debug mode. JTAG UART interface is used to control the NIOS II processorwhich provides a Universal Serial Bus (USB) link to the host computertodownloading programs into memory, online task scheduling, starting and stopping execution, setting breakpoints, and collecting real-time execution trace data.

Avalon Interface

The Avalon bus is an active, on-chip bus architecture consisting of logic and routing resources which provide the link between processor, scheduler and hardware accelerators through memory-mapped register system.The functional model of Avalon interface is shown in fig xx.

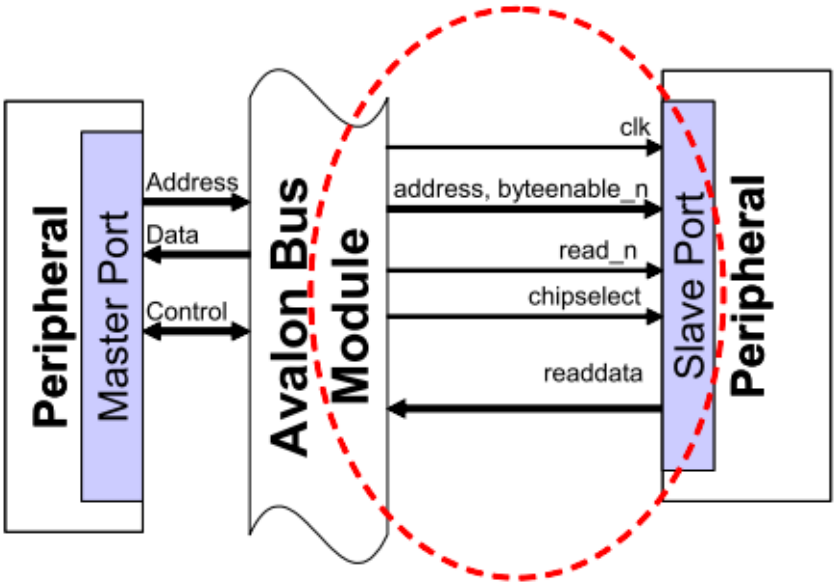


Fig. 2: functional model of Avalon interface

It uses separate address, data and control lines and built-in address decoding logic. Avalon implements simultaneous multi-master bus architecture. The main advantage of this architecture lies in its eliminating the bandwidth-bottleneck as it offers increased bandwidth between peripherals regardless of the bus standard that connects them.The timing diagram of Avalon Interface with processing element is shown figure xx.

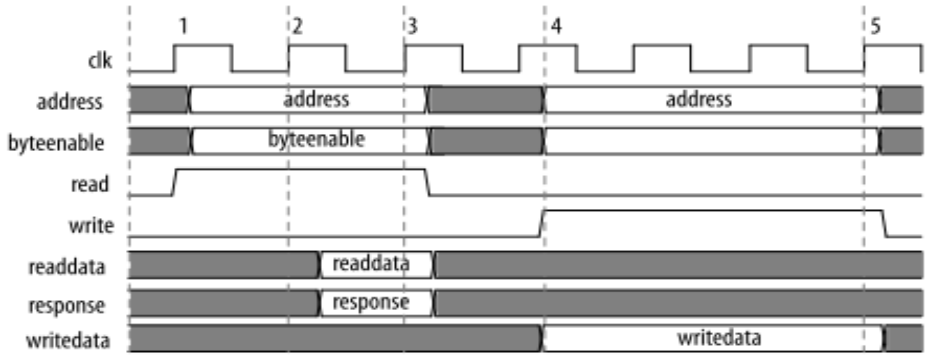


Fig. 3: Timing diagram

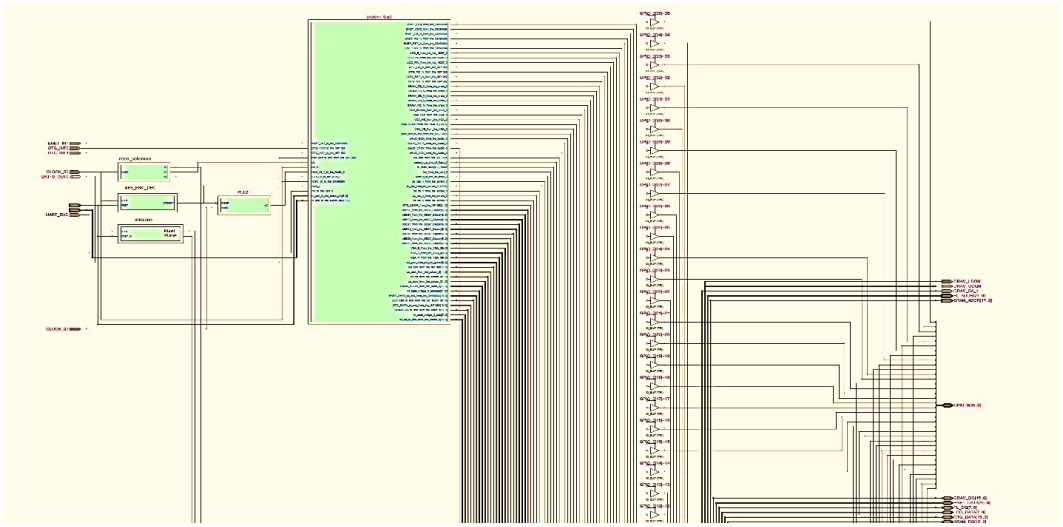
It takes two clock cycle for read and write the data from processing element and one clock cycle for address the device.

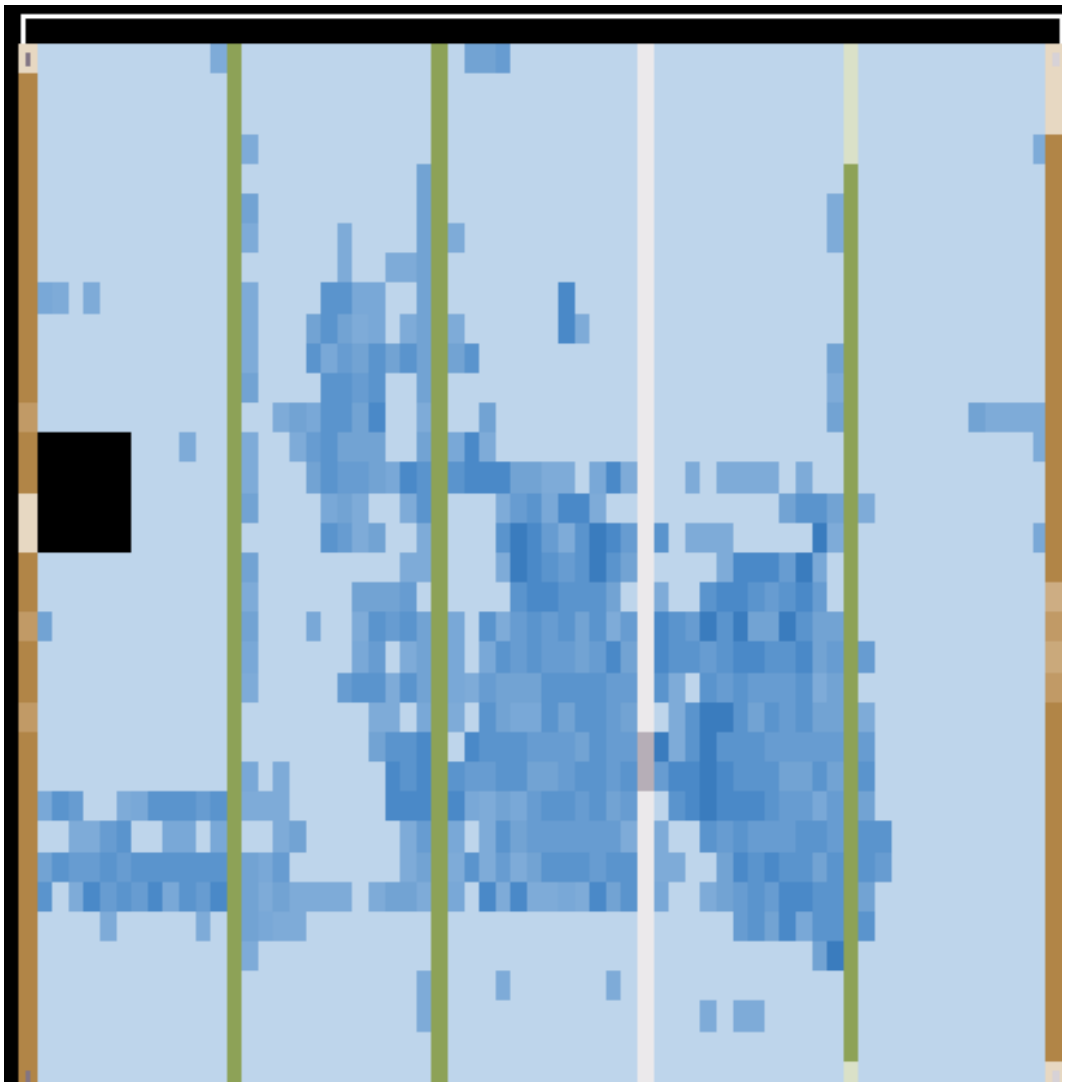
4. Synthesis results

The proposed hardware scheduler was implemented in ALTERA cyclone III FPGA. The eniter core was run at 50 MHz clock frequency. We use NIOS IIstandard version for configure schedulerand fast version for processing unit. Quartus II version 12.1 and SOPC Builder are used to configure the NIOS II processer. Nios II EDSsoftware tool has used to build the application code. For hardware scheduler we choose four standard hardware accelerator such as AES encryption / decryption, reed Solomon error detection and correction unit, integer to floating point convertor core and JPEG 2000 encoder decoder from ALTERA mega wizard plug in manager. All the processing unit and scheduler are connected through avalon switch fabric.The synthesize results of RTL view and chip planner view of proposed system is show in fig xx and The FPGAresource utilizations and dynamic power dissipation of all individual component are listed in TABLE 1.

Table 1 synthesis results

Component	Resource utilization	Power dissipation
NIOS II fast	766	83.11
NIOS II economy	263	28.26
AES encryption / decryption	9765	392.84
reed Solomon error detection and correction	1360	94.13
JPEG 2000	1227	91.53
Canny edge detector	4805	290.22





Conclusion

This paper proposes, designs and evaluates novelprogrammable hardware scheduler for MPSoC. Programmable hardware scheduler exploits a NIOS II embedded processor for run-time task decomposition andscheduling operations.The programmable hardware scheduler can be configured at run time to enable the various algorithms such as time sliced priority scheduling, Earliest Deadline First and Least Slack Time. The entire cores have been implemented on ALTERA cyclone II FPGA and its performance was measured in terms of resource utilization and dynamic power dissipation. The Synthesis resultsshow that programmable scheduler unit consume less logic element and low dynamic power dissipation.

References

[1] M. Technology, W. Wolf, A. A. Jerraya, G. Martin, S. Member, and A. The, "Multiprocessor System-on-Chip," vol. 27, no. 10, pp. 1701–1713, 2008.

[2] B. Tafesse and V. Muthukumar, "Framework for simulation of heterogeneous MpSoC for design space exploration," VLSI Des., vol. 2013, no. 2, pp. 1–16, 2013.

[3] A. Habibi, M. Arjomand, and H. Sarbazi-Azad, "Multicast-Aware Mapping Algorithm for On-chip Networks," Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on. pp. 455–462, 2011.

[4] L. Carro and M. B. Rutzig, "Handbook of Signal Processing Systems," 2010.

[5] D. Theodoropoulos, "Efficient runtime support for embedded MPSoCs," … (SAMOS XIII), 2013 …, pp. 164–171, Jul. 2013.

[6] O. Arnold and G. Fettweis, "Power aware heterogeneous MPSoC with dynamic task scheduling and increased data locality for multiple applications.," ICSAMOS, pp. 110–117, Jul. 2010.

[7] T. Dorta, J. Jiménez, J. L. Martín, U. Bidarte, and A. Astarloa, "Reconfigurable Multiprocessor Systems: A Review," Int. J. Reconfigurable Comput., vol. 2010, pp. 1–10, 2010.

[8] S. M. Fakhraie, S. Mohammadi, and S. Vakili, "Evolvable multi-processor: a novel MPSoC architecture with evolvable task decomposition and scheduling," IET Comput. Digit. Tech., vol. 4, no. 2, pp. 143–156, Mar. 2010.

[9] Y. Cho, N.-E. Zergainoh, S. Yoo, A. A. Jerraya, and K. Choi, "Scheduling with accurate communication delay model and scheduler implementation for multiprocessor system-on-chip," Des. Autom. Embed. Syst., vol. 11, no. 2–3, pp. 167–191, Jul. 2007.

[10] R. Buchmann, U. Pierre, L. I. P. Upmc, A. Greiner, U. Pierre, and L. I. P. Upmc, "A Fully Static Scheduling Approach for Fast Cycle Accurate SystemC Simulation of MPSoCs," no. December, 2007.

[11] Y. Zhang, C. J. Xue, C. Yang, and A. Orailoglu, "Migration-aware adaptive MPSoC static schedules with dynamic reconfigurability," J. Parallel Distrib. Comput., vol. 71, no. 10, pp. 1400–1410, Oct. 2011.

[12] Y. Sun, L. Li, and H. Luo, "Design of FPGA-Based Multimedia Node for WSN," Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on. pp. 1–5, 2011.

[13] H. Shen, M. Hamayun, and F. Pétrot, "On Software Simulation for MPSoC. A Modeling Approach for Functional Validation and Performance Estimation," Des. Technol. …, pp. 91–113, 2012.

[14] D. Li and L. Zhang, "An Efficient Dynamic Scheduling for Multiprocessor Architecture," vol. 11, pp. 294–300, 2012.

[15] D. Li, Y. Hou, Z. Huang, and C. Xiao, "A framework of fuzzy dynamic scheduling for MPSoC architecture," 2011 IEEE Int. Conf. Cyber Technol. Autom. Control. Intell. Syst., pp. 78–83, Mar. 2011.

[16] C. Wang, X. Li, J. Zhang, X. Zhou, and A. Wang, "A star network approach in heterogeneous multiprocessors system on chip," J. Supercomput., vol. 62, no. 3, pp. 1404–1424, Aug. 2012.

[17] M. Ullmann, M. Hübner, B. Grimm, J. Becker: "On-Demand FPGA Run-Time System for Dynamical Reconfiguration with Adaptive Priorities"; FPL 2004, pp. 454-463, August 2004.

[18] H. K.-H. So, R. Broderson: "A Unified Hardware/Software Runtime Environment for FPGA-based Reconfigurable Computers using BORPH"; ACM Trans. on Embedded Computing Systems, vol. 7, no. 2, pp. article 14, Feb. 2008.

[19] Arun Patel et al., "A Scalable FPGA-based Multiprocessor," in FCCM, April 2006, pp. 111–120.

[20] Ravi KiranKaranam et al., "A Stream Chip-Multiprocessor for Bioinformatics," in ACM SIGARCH Computer Architecture News, vol. 36, May 2008, pp. 2–9.

[21] George Mplemenos, IoannisPapaefstathiou, "MPLEM: An80-processor FPGA Based Multiprocessor System," inFCCM, April 2008, pp. 273–274.

[22] E. Salminen, A. Kulmala, and T. D. H¨am¨al¨ainen, "HIBI-basedmultiprocessor soc on FPGA," in Proceedings of the IEEEInternational Symposium on Circuits and Systems (ISCAS '05),vol. 4, pp. 3351–3354, May 2005.