Tools for Architecture Configuration & Synthesis using FPGA in Embedded System

Sunil Kr. Singh*
Dr. R. K. Singh**
Dr. M. P. S. Bhatia***
Ratnakar Madan#

Abstract

In the present high-performance computing environment, embedded systems face many challenges, due to availability of several classes of modern high performance applications. They are demanding very high performance from system with minimum resource. Reconfigurable computing system is emerging technology as an important for system design for present and future computation environment which fulfill the requirement of application in the area of flexibility and performance. In this paper, we discuss the tools for dynamic runtime architecture configuration and synthesis in Embedded System by using reconfigurable computing devices like FPGA. High performance with limited resources needs application-specific architectures (ASIC), while flexibility requires adaptive capabilities. Reconfigurable computing devices promise to meet both needs i.e. flexibility and performance. Reconfigurable computing using FPGAs (Field Programmable Gate Array) is emerging as an alternative to conventional ASICs (Application Specific Integration circuit) and general purpose processors We identified some major key step in embedded system design using reconfigurable computing as system hardware and software partitioning, architecture synthesis analysis, application synthesis analyses, different reconfigurable method tool, libraries, compilation & scheduling of process/task, use of FPGA and some other programmable devices in designing of reconfigurable computing system, and the state-of-the art of ESs . While these devices & tools are currently available but the issue is, how to use these tools. Finally, in this paper, we try to solve some of above design issue tools and method for dynamic runtime architecture configuration and synthesis using FPGA in Embedded System.

Keywords : Reconfigurable Computing, ASIC, FPGA, HW/SW synthesis, Dynamic Reconfiguration.

1. Introduction

Many emerging application in the fields of multi-media,(mobile) communication, networking, computing, consumer electronics etc. require high performance, flexibility and wide variety of functionalities after the system utilization. That's why we try to present in this paper that system architecture has been a emerging topic for research and development and much has change over last few years of the history of modern high-performance embedded system. Flexible system must function in rapidly changing environment in multiple mode of operation. For such system, efficient hardware architecture must mach with algorithm to maximize performance and minimize resource. So that structurally adaptive reconfigurable architecture can meet both these need, achieving high performance with changing algorithm.

Reconfigurable computing device, like FPGA (Field programmable Gate Array) allow the implementation of architecture that change in response to the changing environment. These reconfigurable hardware components are already being used in combination with traditional processor to deliver novel ways of implementing application. So FPGA technology shows great promise computational system. Thus the target systems are built on a heterogeneous computing

*Uttarnchal Technical University, Uttrakhand, INDIA
**Professor, Uttarnchal Technical University, Uttrakhand, INDIA,
***Professor, Netaji Subhash Institute of Technology (NSIT), New Delhi. INDIA,
#UG Research scholar, Bharati vidyapeeth college of engineering, New Delhi, INDIA

platform: including reconfigurable H/W, ASIC and general purpose processor.

2. Related Research Work

Research related area in the field of Reconfigurable Computing is rapidly advancing for scientific and high performance multimedia applications. While today's Field programmable Gate Array (FPGA) technology shows great promise for implementing reconfigurable computational systems, their capabilities in certain areas (such as floating point arithmetic) cannot equal other technologies. For this reason, efficient system architectures must encompass an heterogeneous mix of the best technologies. The target systems are built on a heterogeneous computing platform: including configurable hardware, ASIC and general purpose processors and FPGA based upon partial reconfigurable SRAM. The primary difficulty in this approach lies in system design. A designer must now maintain a set of different system architectures, which exist at different times in the system's lifetime, and map these architectures onto the same group of resources. The designers must manage the behavior of the system, determining the operational modes of the system, the rules for transitioning between operational modes, and the functional properties within each operational mode. In addition, the system must make efficient use of the resources, enabling the designer to minimize the envelope of hardware required to support the union of all operational modes. Currently system design tools are insufficient to manage this complexity. In this paper, we try to propose some basic tool for architecture configuration and synthesis using FPGA in Embedded System.

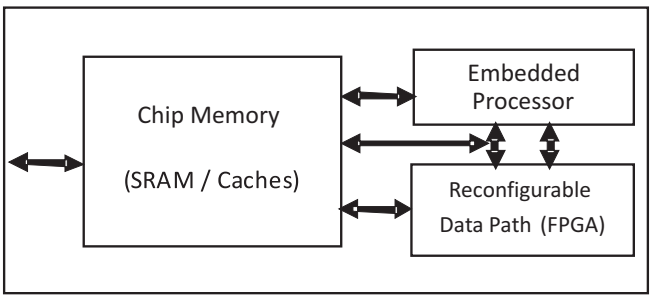

Fig. 1: Architecture of Embedded System using Reconfigurable device FPGA

3. Reconfigurable Computing System

Conventional computing for the execution of algorithm have primary two methods, one is using Von-neumann computing in which programming of microprocessor/ microcontroller using S/W. the second method is making application specific processor or integrated circuits. The first one is more flexible solution with performance degradation. in the later method the system has been designed for particular application , may not be cost effective to modify ao add more feature. So the use of reconfigurable computing system (RCS) which fill the gap between H/W (ASIC/ASIP) & S/W (Microprocessor) approaches. Because conventional computing system have fixed H/W and variable algorithm (S/W) to implement a system but In RCS, it has both H/W as well as algorithm(S/W) are variables have some basic.

With this approach, reconfigurable computing systems have some basic properties and model which are given as:

A) Properties of RCS :

Reconfigurable configurable computing system normally consisting of a matrix of programmable computational units with a programmable interconnection network superimposed on the computational matrix. The main characteristics that have in RCS are

1. Distributed computation,
2. Configurable data path
3. Spatial computation
4. Distributed control and scheduling

B) Designe Models of Reconfigurable Computing

There are some basic model can be used for simulating the architecture of RCS, so that a designer carefully analyze the need of method and component used in design of target system. Some are given as

1. Behavioral modeling
2. Algorithm / Structural Modeling Mesh model
3. Resource Modeling
4. Partially Reconfigurable model
5. Multi-context modeling
6. Constraint modeling
7. Parallel Model

4. Tools & Compilers for Embedded System using FPGA

One important observation of typical embedded system application show that 80 % or more execution time is spent on 20% or less of the code. If all the code is too placed on an FPGA using ES, it is unlikely to generate an efficient application in term of performance & cost.

Developing Embedded System using Reconfigurable devices on the basis of high capacity FPGAs requires many tools, such as FPGA-based boards for rapid prototyping, computer-aided design (CAD) systems, libraries, IP (intellectual property) cores, etc. we try to suggests and describes such tools targeted to the design of reconfigurable embedded systems, whose specifications may change continuously . Some developed tools are given as:

- Software tools for configuring the FPGA, debugging hardware, interactions with the board and providing the required experiments.
- Core Xilinx Spartan 3 FPGA-based prototyping board with an external USB interface.
- Extension boards for connecting typical peripheral equipment and solving application-specific problems, such as implementation of interfaces, communication with standard devices (like memories), etc.
- Hardware/software tools for data compression and decompression that enable the volume of data transmitted from/to FPGA to be reduced size.
- Hardware/software support for reconfiguration of FPGA-based circuits through reloading reconfiguration bits, which have to be preliminary stored in a flash memory available on the board.
- Hardware/software tools for executing operations over binary/ternary vectors and matrices.

- VHDL templates and IP cores for application specific circuits.
- Useful VHDL and Handel-C projects providing essential support for developing reconfigurable embedded systems.
- For dynamic nature of the target applications, use Real-time operating systems, such as VxWorks.
- Use SystemC as simulating language for H/W & S/W simulation and verification in same environment.

5. Configuration & Synthesis using Reconfigurable Devices

A basic model configuration process generates hardware architecture specifications, software modules, process/schedule tables, communications maps, synthesizable hardware specifications, and a run-time Configuration Manger for dynamic adaptation to changing environments. The synthesis process attempts to optimize hardware/software architectures for user- cost such as weight, power, algorithmic accuracy and flexibility.

At this point, the synthesis procedure can generate the actual runtime artifacts. From the behavioral models, a set of tables is produced for the Configuration Manager. The state based behavior is defined in the Behavior Models. These models are transformed into a compact state table. The table contains next state equations for each operational mode. The interfaces to internal and external events are generated to provide the state transition variables to the state machine. These tables and variable interfaces are executed directly by the configuration manager. The synthesis process for hardware, software and both are given as:

A) Hardware Synthesis

For each configurable component (FPGA), a design specification is generated. This design specification includes a hardware design file for each component for each mode. The design for a component*mode is specified in structural VHDL. The VHDL design incorporates computational components from the design library, which can contain user defined VHDL behavioral descriptions and vendor-supplied Intellectual Property (IP) modules. These modules are glued together using components from a standard interface runtime library, which is part of the Runtime Environment described later. These interfaces connect computational components on the same chip with simple FIFO's and asynchronous handshaking interfaces. These interface components manage the physical hardware resources (pins and wires), buffer data, and multiplex multiple logical communications across a single set of wires. When required, data format conversions are also supplied.

B) Software Synthesis

For the general-purpose RISC/DSP components, a set of software specifications is generated. These specifications provide the information needed by the Runtime Environment to enact the desired computational behavior. The Runtime Environment requires several categories of design files:

- Software Load Modules contain executable modules that are downloaded to the processors in the system. The system can generate a common load module that contains the superset of all executable functions (if memory is sufficient) or it will generate a customized module for each of the processors in the system. The customized module is clearly more memory-efficient.
- Real-time schedules contain the list of processes and their priorities. A unique schedule is generated for each processor and for each mode of operation.
- Communication maps describe the information flow between processes. These "streams" can perform communication between two modules on the same processor, or they can transport data across the network, through intermediate processors, and to a remote process anywhere in the system.

C) H/W & S/W Synthesis

Interfaces between software modules and hardware modules/data sources/sinks are automatically inserted during the synthesis process. These interfaces perform the "care and feeding" of hardware interfaces, converting complex communication protocols into simpler hardware compatible protocols. The interfaces also multiplex multiple logical streams over a single physical port and perform data conversion functions.

The result of the synthesis and post processing is a complete executable system, ready for deployment. The deployment is performed in concert with the Runtime Environment.

6. Dynamic Runtime Reconfiguration

The dynamic runtime reconfiguration must support implementation platforms with the following attributes:

- **Heterogeneity :** Optimizing the architecture for performance, size, and power requires that the most appropriate implementation techniques be used. Implementations will require software (implemented on RISC and DSP processors), configurable hardware on FPGAs, and a mix of ASIC components.
- **Low overhead / High Performance :** the runtime environment must minimize overhead, since overhead results in extra hardware requirements.
- **Hard Real-Time:** The target systems have significant real-time constraints.
- **Reconfiguration:** The execution environment must allow hardware and software resources to be reallocated dynamically. During reconfiguration, the application data must remain consistent and real-time constraints must be satisfied.

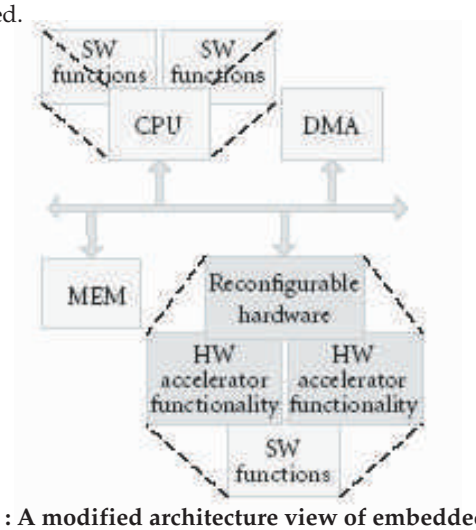

Fig 2 : A modified architecture view of embedded system using reconfigurable hardware

All the above issues must be addressed at multiple levels. At the lowest level, the hardware must be capable of reconfiguration (Figure 2). Software-programmable components, such as DSP's and RISC processors, have excellent inherent hardware support for reconfiguration, since software has the ability to change system function by changing memory contents. Internal CPU hardware structures are designed to restrict dangerous conditions that could damage hardware. FPGA's are an unrestricted collection of gates, switches, and connectors. This process must be provided by a cooperation of the design process and the runtime infrastructure.

Due to dynamic reconfiguration, the optimization is divided into two stages. The first stage of intra function optimization are the main techniques include specify word length, simple SIMD etc. The intra-function optimization will not affect other parts of the system. In the second stage, after the temporal partitioning, we get the knowledge of different functions that how they are grouped into one configuration and how they are perform inter-function optimization (figure:5). The main technique includes resource, localizing memory access, pipelining and parallelizing among functions. So by using two stage optimization, we get more performance and use small area than a sequentially written C program can provide. As a c based methodology, various design steps are performed on c level, like, profiling, mapping, optimization and restructuring execution flow. Its also reduce the debugging and verification work.

7. Conclusion

The main advantage of Reconfigurable Embedded system is the combined flexibility and performance. However, implementing Embedded system using reconfigurable computing does require extra efforts in the various design stages configuration and synthesis, from the general View architecture of embedded system design to modified view of embedded system using reconfigurable device like FPGA, configurable system level design flow, In this paper, we discuss the different tool required for architecture design, runtime configuration and synthesis of Embedded System by using reconfigurable computing. High performance with limited resources needs application-specific architectures (ASIC), while flexibility requires adaptive capabilities. We also try to present the basic attributes of dynamic runtime reconfiguration and application optimization in embedded system using Reconfigurable device. However, some key step of our design and configuration issue, tools and method for designing of Reconfigurable Embedded System are still done manually, and the demonstrator is under development. We are going to concentrate on different design issue with their respective tools. The main challenge is to develop such system also to develop a compiler which compile the selected and optimize application code for the target embedded system using reconfigurable device like FPGA with the help of Hardware Descriptive Language (HDL). Our goal is to provide the basic tool for different design issue in the view of its architecture configuration and synthesis for reconfigurable embedded system.

8. References

1. *Villasenor, J., Mangione—Smith, W., "Configurable Computing", Scientific American, June, 1997.*
2. *S. Edwards, L. Lavagno, E.A. Lee, A. Sangiovanny- Vincentelli, "Design of Embedded Systems: Formal Models, Validation, and Synthesis", Proc. of the IEEE, vol. 85, no. 3, March, 1997, pp. 366-390.*
3. *www.gaisler.com*
4. *Xilinx , Inc., Virtex- E 1.8V field programmable Gate Array, Feb 2001.*
5. *R. Ernst, J. Henkel, and T. Benner, "Hardware–software cosynthesis for microcontrollers," IEEE design Test Comput., vol. 10, pp. 64–75, Dec.1993.*
6. *K. M. GajjalaPurna and D. Bhatia, "Temporal partitioning and scheduling for reconfigurable computing," Proc. IEEE Symp. FPGAs for Custom Computing Machines, pp. 329–330, 1998.*
7. *K. Bondalapati andV. Prasanna. "Reconfigurable Computing systems in Proc. IEEE, vol.90, no.7, July 2002, pp.1201-1217*
8. *K. Chatta and R.Vemuri, "Hardware–software codesign for dynamically reconfigurable architectures," in Proc. of FPL'99, Glasgow, Scotland, Sept. 1999.*
9. *Juanjo Noguera and Rosa M. Badia, "HW/SW Codesign Techniques for Dynamically Reconfigurable Architectures", IEEE Transactions on VLSI Systems, Vol. 10, NO. 4, august 2002, pp 399-415.*
10. *Xilinx, "Virtex platform datasheet," May 2007, http://www.xilinx.com., Septembe 1999.*
11. *K. Bondalapati and V. Prasanna. "Reconfigurable Computing systems in Proc. IEEE, vol.90, no.7, July 2002, pp.1201-1217*
12. *Juanjo Noguera and Rosa M. Badia, "HW/SW Codesign Techniques for Dynamically Reconfigurable Architectures", IEEE Transactions on VLSI Systems, Vol. 10, NO. 4, August 2002, pp 399-415.*
13. *Andr´e DeHon and John Wawrzynek, "Reconfigurable Computing: What, Why, and Implications for Design Automation", technical report.*
14. *Sunil kr. Singh, M.P.S. Bhatia, Rajni Jindal, "Architectural Modelling for Hardware and Software in Reconfigurable Embedded System" in IJRTE international journal of recent trend in engineering, Vol. 1, No. 1, May 2009.*
15. *Sandeep Neema: "Constraint based System Synthesis", Technical Report, Department of Electrical and Computer Engineering, Vanderbilt University, 1999.*