International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Hardware Software Partitioning For A Digital System & Its Validation Using FPGA

Gitanjali R. Solanke
MITAOE Alandi

Dipti Y. Sakhare
MITAOE Alandi

Manish M. Patil
MITAOE Alandi

Abstract

Hardware/Software Co-design is a cooperative design of hardware and software components. One of the goals of Co-design is to shorten the time-to-market while reducing the design effort and costs of the designed products. The flexibility of software allows late design changes and simplified debugging opportunities. Furthermore, the possibility of reusing software by porting it to other processors reduces the time-to-market and the design effort. However, the designer always uses hardware when processors are not able to meet the required performance. This trade-off between hardware and software illustrates the optimization aspect of the Co-design problem. This paper presents framework for Research work to analyse effectiveness of System Verilog for system level modelling and Co-design. It further discusses how system Verilog can be used for modelling software components as well as hardware components of the system. One method is also proposed to carry out the Co-simulation and Codesign. Paper also discussed the expected results from the proposed work.

Introduction

All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts.

Co-design can be defined as the cooperative design of hardware and software. Co-design research deals with the problem of designing heterogeneous systems[1]. Co-design is an interdisciplinary activity, bringing concepts and ideas from different disciplines together. Application specific hardware is usually much faster than software but it is significantly expensive. However software is cheaper to create and maintain but it is slow. Hence performance critical components of the system should be realized in hardware and non-critical components in software. Hardware/software partitioning here, is concerned with deciding which functions should be implemented in hardware and which ones in software. It aims at finding an optimal trade-off between conflicting requirements and performance.

Design flow for co-design approach is shown in figure 1. The Co-design process starts with specifying the system behaviour at system level. The system specification is divided into a set of smaller pieces so-called granules these granules may be for processing input signal, displaying the data etc those granules will be characterised separately for hardware module and for software module. Characterisation for those granules can be done and value of cost matrices can be noted.

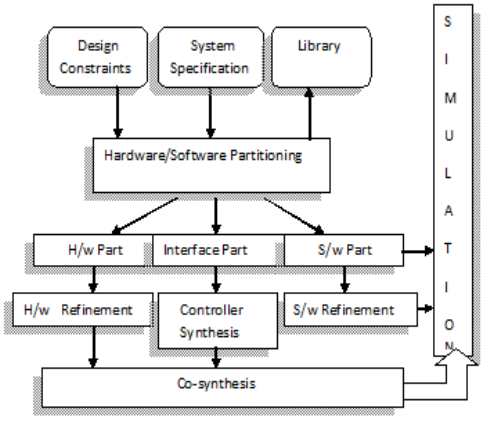

Fig 1: The Design Flow of Co-design Approach

Cost metrics value will help designer in designing whole digital system with minimum conflict between requirement and performance. Hardware cost metrics value are for example execution time, chip area, power consumption or testability and Software cost metrics value may include execution time and the amount of

required program data memory etc. These cost metrics include estimations for hardware or software implementations. All these values of cost matrices are stored in the Library or in data base so that it will be available for further manipulations. In addition to specification of the system, the design constraints should also specify so it will be easy for designer to define global timing and resource constraints related to the overall design costs. In hardware software partitioning step, good mapping of these granules into hardware or software is done The partitioning approach based on genetic algorithms. Output of the step is set of granules which are implemented in hardware or in software. Co-synthesis and simulation of these separated granules can be done Finally, using FPGA characterisation of those granules can be done.by making different combination of hardware and software optimal solution can be obtained If the resulting system matches performance constraints and cost of design program is acceptable then co-design process stops.

Problem Related to Traditional Co-design Approach

The separate development of hardware and software restricts the ability to study hardware software tradeoffs. A Hardware First approach is often pursued with the following characteristics:

Hardware is specified without understanding the computational requirements of the software.

Software development does not influence hardware development and does not follow changes made to hardware during its design process. With this type of process problems encounters as a result late integration , costly modifications and schedule slippage will occurs which increases the time to market factor.

In hardware software codesign, designers consider trade-offs in the way hardware and software components of a system work together to exhibit a specified behavior. Because unified language is not available for specifying whole system the traditional codesign problem takes on many forms. In our project we are trying to suggest a method which makes the use of System Verilog as a unified language to model both hardware and software together on same platform. This paper work will help in solving problem of how system Verilog can be used for modelling software components as well as hardware components of the system. this will help designer not only in implementing functions properly but also meets performance, cost, and reliability goals. In short we are going to provide a platform through which designer can model hardware part and on the same platform software can be modeled which will overcome the problems coming during traditional codesign approach. This common platform is provided by system Verilog.

Advantages of System Verilog and Modelsim

Modelsim provides High performance HDL simulation solution for FPGA & ASIC design teams and it also provides the best mixed language environment and performance in the industry. It has Sign-off support for popular ASIC libraries. ModelSim products are 100% standards based means less risk reuse is enabled and productivity is enhanced. Because of these features.Modelsim is going to help us in every stage of project. ModelSim combines high performance and high capacity with the code coverage and debugging capabilities required to simulate larger blocks and systems and attain ASIC gate-level sign-off. Comprehensive support of system Verilog, VHDL and System C provide a solid foundation for single and multi-language design verification environments.

The best part of the Modelsim is it supports System Verilog language. SystemVerilog has new design modelling capabilities it extends IEEE 1364 Verilog standard. It supports abstraction of c language data types.RTL coding is done more accurately using system Verilog. Along with new modelling capabilities new verification capabilities also introduced. It supports race-free test benches and Object-oriented test programs. SystemVerilog is the next generation of the Verilog standard giving much higher level of modelling abstraction advanced capabilities for design verification

There are other SystemVerilog capabilities we are using, such as assertions, constrained random stimuli, and testbench automation capabilities. If you can use one language to do modeling, then a standard language is much better than using a proprietary one to do the stimulation, functional coverage, and assertions.

The System on programmable chip (FPGA) design including processors, buses, memory and hardware accelerators provide an opportunity for system designers to develop high performance optimal systems on programmable chip. We present here the concepts for a system on programmable chip.

Methodology

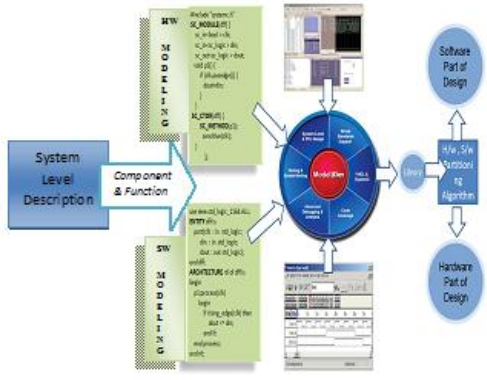

Fig 2: Methodology

The project work starts with designing of small granules for Digital System. Granules are ADC,7 segment, Keyboard etc. using System Verilog .These granules can be modelled in hardware using Verilog so that dedicated hardware can be generated. By using FPGA board and Logic analyzer characterisation is done. Now the same granule can be modelled in software and then characterised. All these characteristics of granules will be stored in library. This enables the designer to compare different design alternatives to find appropriate solutions for different objective functions e.g. high-performance, low-cost or low-power designs. All these things will be easy when we are doing modelling on single platform and this single platform is provided by system Verilog Here in our project Co-design approach is implemented using System Verilog. To get detail idea of project case study is given which will help in understanding concept of project.

Case study

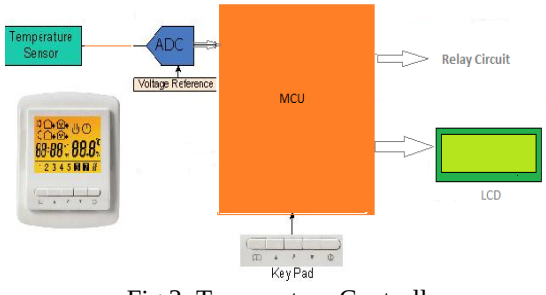

Fig 3: Temperature Controller

To get more breaf idea about project consider an example of temperature controller shown in fig 3. The whole system of temperature controller is divided into small granuels. Those granuels are temperature sensor, amplifier, ADC/DAC microcontroller, LCD and Key pad.Once the granuels has decided next step is to model those granuels. Using System Verilog these granuels can be modelled. As the system verilog provides a common platform for modelling software as well as hardware no need to go for traditional way of co-designing approch. Once the granuels has been designed in system verilog modelsim will help in simulating the results. Granuels will be modelled seperatelly for both hardware and software. A soft processor core on FPGA will help during modelling of granuel in software.Whereas a dedicated hardware on FPGA will help me in modelling of same granuel in hardware.when this process is going on we are trying to analysis how much burden software is getting in doing the activity or how much time is required for software to display result on LCD and how much time is required for dedicated hardware to do same thing.such value of cost matrics of granules can be recorded and stored in library ie in our research work , by giving a platform of system verilog we Are trying to solve the problem occuring in traditional codesign approch.Thus our reserch work is going to help design engineer in designing process.

Result

For Hardware module

1) Area Specifications
i) Total memory usage is 243 Mbytes for ADC module.
ii) Total memory usage is 247 Mbytes for keypad module.
iii)Totalmemoryusageis252Mbytes for Controller module.

2) Total Power Consumption for routing the given module on FPGA is 0.81 W Power Consumption for ADC, Keypad & Controller module is found to be 80.98 m W.

3) For the Temperature Controller System Delay Calculated using Logic Analyzer is 274 µs

For Software module

1) Details of Area required are
i) Total memory usage is 2.34 kilobytes for ADC module.
ii)Total memory usage is 2 kilobytes for keypad module.
iii) Total memory usage is 1.62 kilobytes for Controller module.
2) For the Temperature Controller System Delay

Calculated are 147 ms.

3) Details of Power consumed

Power consumed for Keypad module is 2.5 mW.
Power consumed for ADC module is 15 mW.
Power consumed for Controller module is 50 mW.

Conclusion

Hardware/Software partitioning for Digital System using System Verilog is totally a new concept which we are trying to put forward through our project. Using System Verilog,we will definitely overcome the difficulties involve in traditional hardware/software codesign method. Our project work will help Design Engineers to develop the systems more accurately with minimum efforts and time. Hardware-software co-design using System VERILOG is a methodology for solving design problems in processor based embedded systems also. This hardware-software co-design methodology allows the concurrent design of both hardware and software there by reducing the design cost and meets the performance goals.

Reference

[1]Alexandre Amory, Fernando Moraes, Leandro Oliveira, Ney Calazans, Fabiano essel "A Heterogeneous and Distributed Co-Simulation Environment" Proceedings of the 15th symposium on Integrated circuits and systems design conference , 2002 IEEE, pp 115-119.

[2]Wu Jigang, Qiqiang Sun Thambipillai Srikanthan "Multiple-choice hardware/Software Partitioning: Computing Model and Algorithms" proceeding of 2nd International Conference on Computing & Processing (Hardware/Software)2010 IEEE, pp V2-61-VZ-65.

[3]Fan Zhou, Jun Yang, Longxing Shi, Yu Zhang " Hardware-Software partition of Fixed-point hardware Accelerator from Statistical Perspective " Proceeding on 6th International Conference On ASIC , 2005 IEEE, pp 78 – 81.

[4]Francisco Fons, Mariano Fons, Enrique Cantó "Hardware-Software Co-design of Dynamically Reconfigurable FPGA-based Fuzzy Logic Controller" Conference on Fuzzy Systems , 2006 IEEE, pp 1228-1231.

[5]Shijue Zheng,Yan Zhang, Tingting He " The Application of Genetic Algorithm in Embedded System Hardware-software Partitioning " proceeding of International Conference on Computing& Processing (Hardware/Software) 2009 IEEE,pp 219-222.

[6]K.S. Tham And D.L. Maskell " Software-Oriented Approach To Hardware-Software Co-Simulation For FPGA-Base RISC Extensible Processor" proceeding of International Conference on Components, Circuits, Devices & Systems 2006 IEEE, pp 1-6.

[7]Mariano Lopez,Enrique Canto,Mariano Fons " Hardware-Software Codesign of a Fingerprint Image Enhancement Algorithm" proceeding of 32nd conference on Component Circuit,Devices & Systems 2006, IEEE, pp3496-3501.

[8]Wei Wang, Qiang Wu, Wei Xie "Hardware-Software Co-design for Dynamic Reconfigurable Computing with Collaborative Supports of Architecture and Operating System " proceedings of 2007 11th International conference on Computer supported cooperative work in design 2007 IEEE, pp 275- -279.

[9]Peter Arato, Sandor Juhasz, Adam Mann, Andras Orban, David Peter Papp "Hardware-software partitioning in Embedded System Design"proceeding of International Conference on Communication, Networking & Broadcasting 2003IEEE, pp 197-202.

[10]Félix Morenoa, Ignacio Lópezc and Ricardo Sanzb "A design process for hardware/software system co-design and its application to designing a reconfigurable FPGA" proc. Of 2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, 2010 IEEE DOI, pp 556-562.