International Journal of Advancements in Electronics and Electrical Engineering– IJAEEE
Volume 4 : Issue 2 [ISSN : 2319-7498]
Publication Date : 30 October, 2015

Implementation and Execution of Math Partial Reconfiguration Region and LED Dealing with Xilinx PlanAhead

Ipseeta Nanda S M Ali

Abstract— **Dynamic Partial Reconfiguration (DPR) of Field Programmable Gate Array (FPGA) was introduced to overcome the need for more resources on the FPGA.PDR could change the functionality and efficiency of the system in order to accommodate more hardware module, save power and fabric area.PDR involves the design of modules that are independent of each other so that they can be loaded in the same fabric area one after other. In this paper the authors reconfigure some specific region of the FPGA with a new functionality at run time while the remaining areas remain static during this time. The complexities during the runtime can be simplified by a tool called PLANAHEAD which was introduced by XILINX that is able to implement during runtime reconfigurable systems for all VIRTEX field programmable gate array. PLANAHEAD is the first graphical environment for partial reconfiguration which gives the flexibility for reducing the board space, change a design in the field and also reduces the power consumption.**

***Keywords*— Dynamic Partial Reconfiguration (DPR), Field Programmable Gate Array (FPGA), PLANAHEAD**

I. Introduction

FPGA Reconfiguration technology has begun to address some of many complexities associated with the development of hardware and configurable hardware whose functionality can be changed during operation. The potential of self-adaptable hardware to enable revolutionary new applications has led to a large and growing body of work in configurable computation. An ability to reconfigure the logic of such devices after deployment enables revision of design logic, as necessary, to correct design deficiencies or to alter or enhance functionality. The logic of an FPGA device is specified /programmed in the manner of up loading one or more files to the device that specifies the configuration of the device that specifies the configuration of the device logic. Thus files are commonly referred to as bitstream files.DPR can be used to correct, enhance, or replace firmware functional blocks in the FPGA reconfigurable logic while that device is operating in the system without disruption of other ongoing functions. Programming these devices with partial bitstreams can now be accomplished over high speed multi gigabit per second serializer/de serializer (SerDes)interfaces and devices which will be configured and programmed specifically to support ongoing device level needs.

Ipseeta Nanda
Silicon Institute of Technology, Sambalpur
India

S M Ali
Institution of Engineers, Kolkotta
India

II. Dynamic Partial Reconfiguration

An FPGA has the flexibility of being configured and reconfigured after manufacturing, which allows the user to change the functions of the device without fabricating it. The DPR feature allows us to modify a subset of resources on an operating FPGA by loading the reconfigurable bit file. After downloading the bit file to configure the FPGA, the design of an operating FPGA can be dynamically modified by loading the partial bitstream in the reconfigurable logic blocks, while the application running the remaining logic is not interrupted. The concept of Partial Reconfiguration (PR) action is depicted in the fig.1

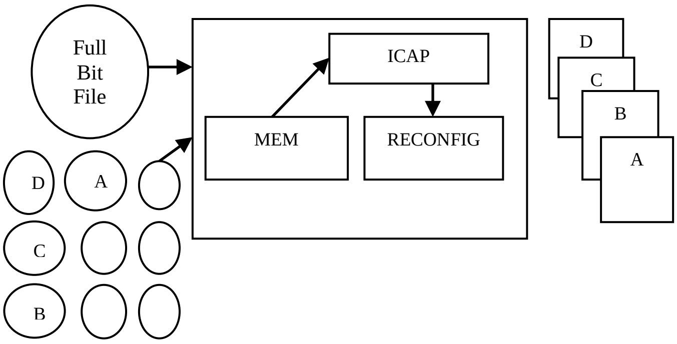

Fig 1Partial Reconfiguration

DPR is also known as active partial reconfiguration, permits to change a part of the device while the rest of FPGA is still running i.e. while the partial data is sent into the FPGA, the rest of the device is still running and the new data is being configured into FPGA.

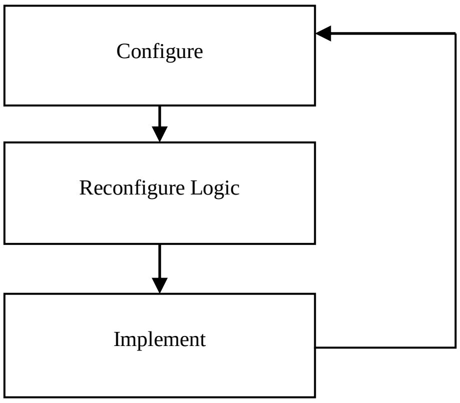

Fig 2 Dynamic Partial Reconfiguration

The DPR is supported by all types of Virtex series device. This improves timing performance and simplifies the process of building a Partial Reconfiguration(PR) design. Static module is the design remains in operation during the PR process. Partial reconfiguration module can be swapped in and out of the device. Multiple Partial reconfiguration module can be defined for a specific region. Partial reconfiguration region is the part of the FPGA that is set aside for Partial reconfiguration modules. Here more than one PRRs can be set aside for reconfiguration and a proxy look up table (LUT) is used which automatically connects the reconfiguration modules to static module which possible in Virtex 4 kit.Busmacros have been used in partial design flows by the vendor Xilinx and the signal to wire binding has been achieved by initiating a macro consisting of a LUT in the reconfigurable region. By placing bus macros at a defined position on the partial module border, signals are bound to the internal macro wires.

A. *PlanAhead*

PR can be implemented through the use of Xilinx PlanAhead software. Modular design requires additional procedures for synthesis and implementation partial reconfiguration generally requires the use of a modular design flow.

PlanAhead’s main purpose is to customize the way circuits designed in ISE are laid out on the FPGA as well as providing timing and placement analysis to improve circuit function. By using this tool uses can group circuits and modules. The benefit of this is that if each module is in its own area, the task of PR becomes much easier as only one area is being programmed and will not affect any of the other sections. This task is known as floor planning. PlanAhead also provides a useful set of design rule checks which can used to improve designs and provide suggestions to the designer.[1]. The partial reconfiguration meets the requirements needed by complex design. Indeed, the dynamic partial reconfiguration makes hardware more and more flexible and allows user to modify internal structure of FPGA on the fly, without having to turn off. The partial reconfiguration allows the configuration of parts of FPGA while other parts are still running. A new FPGA chips with partial reconfiguration capability such as Virtex II, Virtex II Pro, and Virtex-4 families are available. These new devices include two subsets of resources; the system resources and the operational resources. The system resources may be composed by a processor and by other internal peripherals.

The operational resources are zones in the device reserved to the user; it may be used to build reconfigurable modules. During the execution of application, the content of reconfigurable module may be loaded by many functions, which are configured one after one. The most important, for the new FPGA families, is their ability to be self-reconfigured under software control through a reconfiguration hardware interface called Internal Configuration Access Port (ICAP). The system utilizes the On-Chip PowerPC core and FPGA logic to automatically reconfigure bit streams from an external memory, like compact flash. The main benefit to be gained from using self reconfiguration is the time needed to configure the design. The benefit of reducing reconfiguration time is particularly apparent when making small changes to large designs but reduced benefit can still accrue with larger changes on smaller designs.

III. Experimental Modular Design

There are various steps to be performed for creating a PR module:

- Create a Reconfiguration Partition(RP)
- Add a Reconfigurable Module(RM)
- Define P block ranges for the reconfigurable partitions
- Add Bus Macros
- Run DRC check
- Implement Configuration
- Verify the configuration

The PR modules can be implemented using the PlanAhead tool. Partitions are created for each and every module and p blocks will be defined and then bus macros will be placed. The DRC checks will be performed and the configuration will be verified.

IV. Implementation Flow

In this section the implementation flow is discussed for difference based and module based PR, handled internally and externally. Discussion here is around Xilinx FPGAs and tools as we have implemented various applications around it. With the Difference Based flow the designer must manually edit low-level changes, while the Module Based flow allows the designer to split the whole system into modules.

A. *Difference based PR*

The difference based PR allows designer to make small logic changes using FPGA Editor. After completing the design flow till generation of bitstream (without change), Steps shown in Fig.3 are to be followed to create changes and generating partial bit file. As the outcome of these steps, a bit stream is generated that programs only the difference between the two versions of design as partial bitstream. Switching the configuration of module from one implementation to another is very quick because the bitstream difference can be much smaller than the entire device bitstream. There are various types of small changes that can be made to FPGA like changing I/O standards, Block RAM contents, LUT equation MUXs, Flip-Flop initialization and reset value, pull-ups and pull downs on external pins, Clock manager(DCM) settings and block RAM write mode.A difference-based partial reconfiguration bitstream can be created with the BitGen utility using the switch. This switch produces a bitstream that contains only the differences between the input .ncd file and the original bit file. Depending on the changes, this partial bitstream can be much smaller than the original bitstream. The bit length and reprogramming speed of a particular partial bitstream are directly proportional. These bitstreams can be loaded quickly and easily by the software. Changing any property or value that would impact routing is not recommended due to the risk of internal contention. This design flow is also not recommended for making large changes in the functionality or structure of a design, for example, changing an entire algorithm. When there are sizable changes or the routing has to be modified, the recommended flow is to start from the HDL. The bit files generated by using above flows can be downloaded using JTAG programmer. This approach is known as External configuration as the configuration is controlled by external entity.

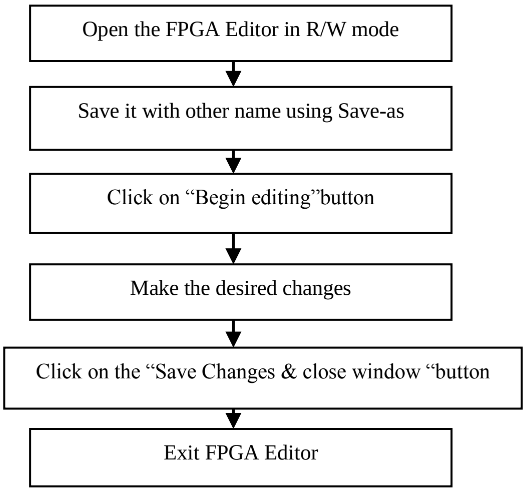
Open the FPGA Editor in R/W mode
Save it with other name using Save-as
Click on "Begin editing"button
Make the desired changes
Click on the "Save Changes & close window "button
Exit FPGA Editor

Fig 3 Steps for Editing Design in FPGA Editor

B. *Module based PR*

Module based Partial reconfiguration involves defining distinct portions of an FPGA design to be reconfigured while the rest of the device remains in active operation. These portions are referred to as reconfigurable modules. When the substantial amount of logic is to be changed or when routing is to be modified this approach is suitable. The methodology is totally different than the difference based PR. First step is to recognize mutually exclusive functionality in the design. Then partition the design into static and dynamic components. Determine PR regions and associated RMs. Next is to decide Bus Macro interfaces. Next step is to create Top level design context with all globals such as IO ports, clock constructs (DCMs, BUFs), black boxes instantiation for PR regions, all Bus Macros and signals etc. HDL file hierarchy is to be created as discussed in [6]. It is recommended to build flat design with all RMs to check the functionality and signal routing. It is not compulsory but highly recommended step. Last step is to build modular PR design. Complete flow is shown in Fig.3. The details regarding properties of RM, placement constraints, use of communication interface can be found in [10].

Early Access PlanAhead is a hierarchical design and analysis tool which supports partial reconfiguration design flow. The tool can be deployed for all steps between synthesis and PAR. It helps to correlate logic and physical design view graphically. The tool allows to place PRRs and Bus Macros of design and to run multiple design rule checks. It outputs constraints file. We can invoke Explore Ahead implementation tool to generated static full or power up bit file, all partial and blank bit files for all PRRs from its GUI details about tool can be found in .The partial bit files generated are smaller than the bit file for complete FPGA. The downloading time of these partial bit files is less and depends upon the size of PRR. The comparative study for sizes of bit files and reconfiguration time of developed application using above methodology in our lab can be found in [10].The bit files generated by using above flows can be downloaded using JTAG programmer. This approach is known as External configuration as the configuration is controlled by an external entity.

V. Control System

The control system design, shown in fig 4(a) and fig 4(b) consists of two PRR (Partial reconfiguration region) each having two RMs. The math PRR consists of two functions: addition and subtraction whereas the LED PRR consists of right and left shifting LEDs.In this paper math PRR interacted by using HyperTerminal whereas interaction with LED PRR has been achieved using push buttons. All the modules have been pre-synthesized and the netlist files were stored in synth directory. The data directory carries the ucf, bus macro, and additional netlist files.

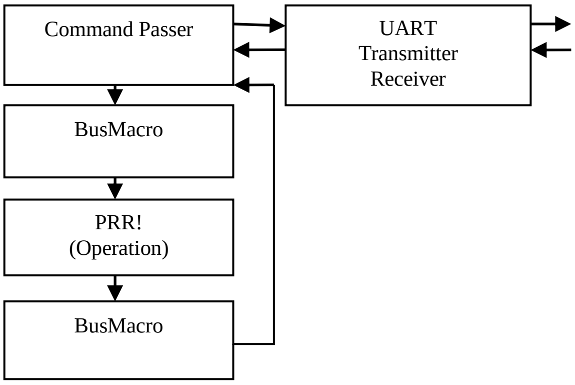

Fig 4(a) Control System design

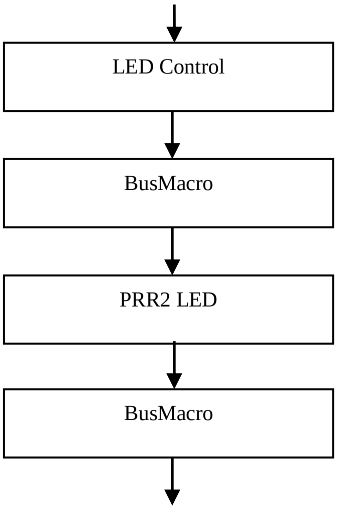

Fig 4(b) Control System design

VI. Experiment & Result

The PR module has been implemented using the PlanAhead tool. Partitions were created for each and every module and the p-block ranges has been defined and then bus macros were placed .In this paper two reconfigurable modules such as Math PRR i.e. adder and subtractor and LED PRR i.e. right shift and left shift being reconfigured. The partitions and the p-block ranges are created for each module using PlanAhead tools.The result can be seen as either a change in direction of LED movements or the application requesting input operand for addition or subtraction depending on the option selected.Running the design rule check (DRC) allows us to make sure that there is no errors up to this moment.it is highly recommended to perform DRC at various steps of the design flow to isolate error early.The DRC checks have been performed and configured were verified as shown in fig (5).

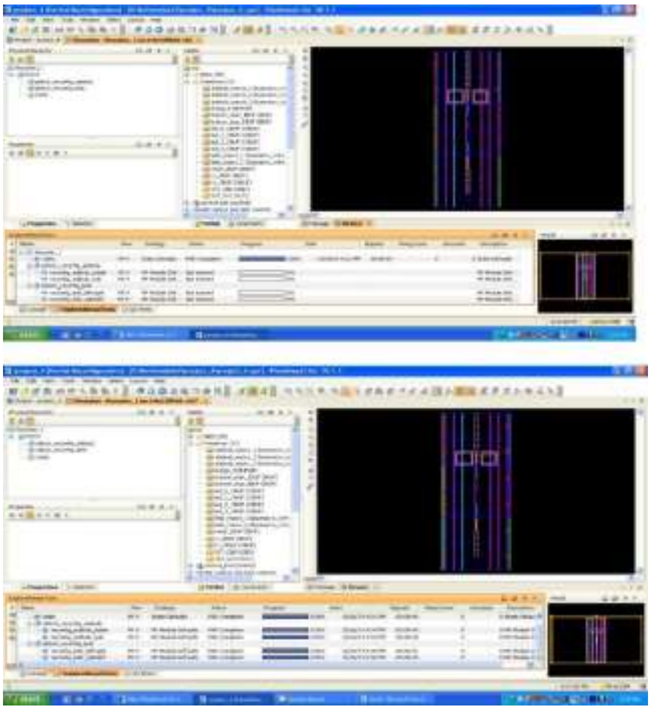

Fig 5 Result

Acknowledgment

Ipseeta Nanda expresses her thanks to Dr. Sheetal Bhandari, PCCOE, Pune for her valuable time & support to learn this technique in the research lab.She would like to thank all faculty member of the institute for their moral support.

References

[1] K.Bhuvaneswari, V.Srinivasa Rao, ―*Dynamic Partial Reconfiguration in Low-Cost FPGAs* 2013 International Journal of Scientific & Engineering Research, Volume 4, Page(s):1410-1413

[2] Sheetal U.Bhandari,Shila Subbaraman,Shashank Pujari,Rashmi Mahajan,*Internal dynamic partial reconfiguration for real time signal processing on FPGA"*2010 Indian Journal of Science and Technology. Volume 3 no.4 April 2010,Page(s):365-368

[3] Sheetal U.Bhandari, Shila Subbaraman, Shashank Pujari Digital Signal Modulator on FPGA using On the Fly partial Reconfiguration "2009 International Conference on Advances in Computing, Control and Telecommunication Technologies 2009 IEEE

[4] K. Bondalapati, and V. K. Prasanna,, ―*Reconfigurable computing systems*, Proceedings of the IEEE Volume 90,Issue 7, July 2002, pp. 1201 – 1217

[5] www.xilinx.com

[6] Xilinx, Two Flows for Partial Reconfiguration: Module Based or Difference Based, XAPP290 (v1.2) September 9, 2004

[7] Xilinx, Difference-based partial Reconfiguration, XAPP290 (v2.0) December 3, 2007

[8] Partial Reconfiguration Design with Planahead 9.2 by Brian Jackson Ver1.1, August 2007

[9] Xilinx, Media Presentation ―Using the planAhead 9.2 for Partial reconfiguration flow by Brain Jackson

[10] Salih Bayar, Arda Yurdakul, ― Self Reconfiguration on Spartan III FPGAs with compressed partial Bitstreams via a parallel configuration Access Port (cPCAP) core 2008 Research in Microelectronics and Electronics, 2008. PRIME 2008. Ph.D. Volume, Issue, June 22 2008-April 25 2008 Page(s):137 – 140

[11] Brandon Blodget, Scott McMillan, Patrick Lysaght, ―A lightweight approach for embedded reconfiguration of FPGAs" Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'03) 1530-1591/03

[12] Cindy Koo, ―*Benefits of Partial Reconfiguration*, Xcell journal fourth quarter 2005

[13] Xilinx, Getting started with EDK, manual, January, 2007

[14] Xilinx, XPS HWICAP (v1.00.a), DS586 November 05, 2007

[15] Xilinx, OPB HWICAP, DS 280 (v1.3) March 15, 2004

[16] S. Lopez-Buedo, J. Garrido, E. Boemo,*Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems*, IEEE Transactions on Components and Packaging Technologies, Volume 25, Issue 4, Dec. 2002, pp. 561 – 566.

[17] M. Hübner, J. Becker,Exploiting Dynamic and PartialReconfiguration for FPGAs – Toolflow, Architecture, and System Integration," Proc. 19th SBCCI Symp. on Integrated Circuits and Systems Design, Ouro Preot, Brazil, 2006.

[18] Ipseeta Nanda,C S Panda"A Review on Partial Reconfiguration and its Tool and Application" in the National Seminar organized on the occasion of the 17th Odisha Bigyan Congress under the auspices of Siksha'O'Anusandhan University(SOA),Bhubaneswar and Indian Science Congress Association:Bhubaneswar Chapter in collaboration with the Orissa Environmental Society.6th Dec2014

About Author (s):

IpseetaNanda, presently working as assistant professor at Silicon Institute of Technology, Sambalpur,Odisha,India.She obtained B.Tech (2004,ETC) and M.Tech (2010,ETC specialization in Communication System).She is appointed as a Chairperson of Women in Computing, Computer Society INDIA COUNCIL,IEEE 2014,member of IET,UK and also life member of ISTE. She has been included in four Editorial Board of International Journals (IJECCE, IJEIR, IJAIM, IJISM)

Dr. S M Ali, presently working as a Director (Membership) at The Institution of Engineers (India), Kolkata. He has obtained his M.Tech from Calcutta University, PhD in Electrical Engineering from International University, USA in 2006 and DSc in Engineering from International University, USA in 2008.He had guided 7nos. of PhD students and presented 100 papers in National and International conferences and 95 nos. of papers published in National and International Journals. For his outstanding contribution in the field of Science and Technology Prof. Ali was felicitated more than 25 nos. of National & International Awards. He is life member of CSI, ISTE, IIIE, AIIMA, ISCA, ISTE, SESI, and IET (UK).