FPGA Implementation of Reconfigurable Switch Architecture for Next Generation Communication Networks

M. Hema Lata Rao and Rajeev Tripathi

***Abstract*—FPGA are a special form of Programmable logic devices (PLDs) with higher densities as compared to custom ICs and capable of implementing functionality in a short period of time using computer aided design (CAD) software. Reconfigurability refers to systems incorporating some form of hardware programmability, that customizes how the hardware is used using a number of physical control points. These control points can be changed periodically in order to execute different applications using the same hardware. This paper presents the FPGA implementation of reconfigurable switch architecture for next generation communication networks (NGN's) where the configurations are changed by changing the control signal at the input. The reconfigurable architecture is implemented in HDL(Verilog) and the code is burned in Xilinx Spartan3-XC3S400 series using JTAG mode.**

***Index Terms*—FPGA, MINs, network architectures, reconfigurability.**

I. Introduction

The rapid evolution in the field of telecommunications has led to the emergence of new switching technologies to support a variety of communication services. These communication services provide wide range of transmission rates in a common, unified integrated services network. At the same time, the progress in the field of VLSI technology has brought up new design principles of high-performance, high-capacity switching fabrics to be used in the integrated networks of the future. The Next Generation Network (NGN) [1], [2], [3] is an important future network service used to describe the network that will replace the current PSTN network around the world today used to carry voice, fax, modem signals, etc. The NGN is essentially a managed IP-based (i.e., packet-switched) network that enables a wide variety of services with global mobility. Among those services are VoIP, videoconferencing,

Instant Messaging, e-mail, and similar other kinds of packet switched communication.

Field Programmable means that the FPGA's function is defined by a user's program rather than by the manufacturer of the device. A typical integrated circuit performs a particular function defined at the time of manufacture. In contrast, the FPGA's function is defined by a program written by someone other than the device manufacturer. Depending on the particular device, the program is either 'burned' in permanently or semi-permanently as part of a board assembly process, or is loaded from an external memory each time the device is powered up. This user programmability gives the user access to complex integrated designs without the high engineering costs associated with application specific integrated circuits.

The software translates a user's schematic diagrams or textual hardware description language code then places and routes the translated design. Most of the software packages have hooks to allow the user to influence implementation, placement and routing to obtain better performance and utilization of the device. Libraries of more complex function macros (eg. adders) further simplify the design process by providing common circuits that are already optimized for speed or area.

II. Preliminaries

FPGA stands for Field Programmable Gate Array. There are many forms of devices which are field programmable. These are PAL, PLD, CPLD, and FPGA. These devices differ on their granularity, how the programming is accomplished etc. PAL, PLA and CPLD devices are usually smaller in capacity but more predictable in timing and they can be implemented with Sum-of-Products, Product-of-Sums or both. FPGA devices can be based on Flash, SRAM, EEPROM or Anti-Fuse connectivity. The most successful FPGA devices are based on SRAM. This is because all other memory types are much less dense in terms of area than SRAM. Also some types of connectivity are One-Time Programmable (i.e. Anti-Fuse) so they are not very flexible. FPGA's were introduced as an alternative to custom ICs for implementing entire system on one chip and to provide flexibility of reprogram ability to the user. Introduction of FPGAs resulted in improvement of density relative to discrete SSI/MSI components (within around 10 x of custom ICs). Another advantage of FPGAs over CustomICs is that with the help of computer aided design (CAD) tools circuits could be implemented in a short amount of time (no physical layout process, no mask making, no IC manufacturing). SRAM based FPGA's have no maximum erase cycle limitations either. In an FPGA logic blocks are implemented using multiple level low fan-in gates, which gives it a more compact design compared to an implementation with two-level AND-OR logic. FPGA provides its user a way to configure: (i)The intersection between the logic blocks and

Manuscript received June 22, 2012; revised July 20, 2012.

M. Hema Lata Rao was with the Jagannath Institute for Technology and Management, Parlakhemundi, India. She is now research scholar at Indian Institute of Technology Bhubaneswar (e-mail: hemalata.rao@cutm.ac.in).

Rajeev Tripathi is with the National Institute of Technology, Allahabad, India (e-mail: rt@mnnit.ac.in).

DOI: 10.7763/IJET.2012.V4.481

(ii)The function of each logic block. Logic block of an FPGA can be configured in such a way that it can provide functionality as simple as that of transistor or as complex as that of a microprocessor. It can used to implement different combinations of combinational and sequential logic functions. Logic blocks of an FPGA can be implemented by any of the following:(i)Transistor pairs (ii)combinational gates like basic NAND gates or XOR gates (iii)n-input Lookup tables(iv) Multiplexers (v)Wide fan-in And-OR structure.

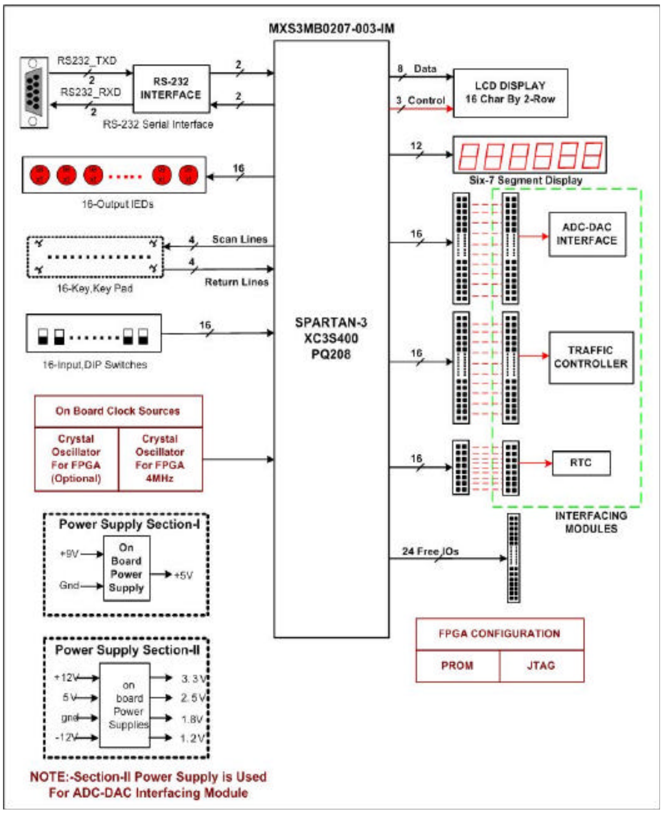

Fig. 1. Block diagram of Spartan 3-XC3S400

III. FPGA Implementation of Reconfigurable Switch Architecture

The goal is to have a reconfigurable [4], [5] switch fabric where the configuration can be changed according to the need by changing the links in between the stages and to download the program on FPGA [6]. The four switching fabrics used in reconfigurable switch architecture are generalized- cube network [7], omega network, banyan network [8], [9] and indirect binary n-cube network [10], [11] for *N*=8 .The four configurations are merged to form a single reconfigurable architecture as shown in Fig. 2. The architecture is programmed in Verilog, [12] synthesized in design compiler (synopsis) [13] and is burned in Xilinx Spartan3-XC3S400 series using JTAG cable. The architecture shown in Fig. 2 has the routing control within the switching element itself. The destination address and the content can be provided at the input of the configuration. The control signal selects the configuration according to the inputs at the four multiplexers. The output shown in the simulation Fig. 4 is the content reaching to specified destination address.

The merging of four different configurations in single architecture provides minimized area and power with a marginal increase in delay. The main advantage of this reconfigurable switch architecture is that the set of admissible permutation for each configuration is different and hence the permutation set blocked in one configuration can be allowed by other configuration. The flexibility achieved in terms of number of configurations is high leading to the reduced area and power with a very marginal increase in delay. Since all four architectures share similarities in their structures, the potential is given for efficiently combining them in a single reconfigurable VLSI circuit, leading to competitive designs in terms of area, performance, and power consumption.

At first individual architectures are dumped on the chip with 8 inputs (each input of 3 bit) and 8 outputs (each output of 3 bit) using verilog code. The input pins considered for Spartan3-XC3S400 series are 16 dip switches (p34, p35, p36, p37, p39, p40, p42, p43, p44, p45, p46, p48, p50, p51, p52, p57) and 8 pins are selected from I/O connector (J7).Similarly for output, 16 onboard LED's and 8 pins from I/O connector (J7) are used. The HDL (Verilog) code for reconfigurable switch architecture is dumped using the same input and output pins mentioned above with two additional pins from I/O connector (J7) for control lines. The design flow as shown in Fig. 3 indicates design entry, simulation, synthesis, implementation and device programming.

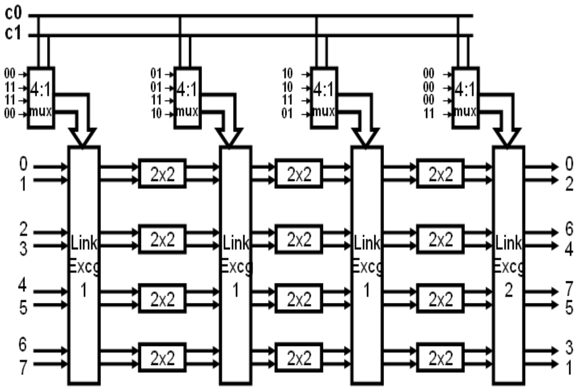

Fig. 2. Reconfigurable switch architecture [internal routing]

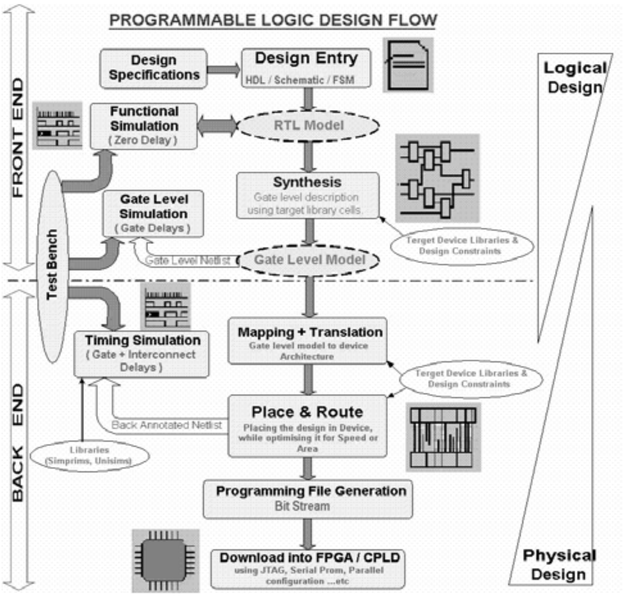
PROGRAMMABLE LOGIC DESIGN FLOW
FRONT END
BACK END
Design Specifications
Design Entry
HDL / Schematic / FSM
Functional Simulation
(Zero Delay)
RTL Model
Synthesis
Gate level description using target library cells.
Gate Level Simulation
(Gate Delays)
Terget Device Libraries & Design Constraints
Gate Level Netlist
Gate Level Model
Test Bench
Timing Simulation
(Gate + Interconnect Delays)
Mapping + Translation
Gate level model to device Architecture
Terget Device Libraries & Design Constraints
Back Annotated Netlist
Place & Route
Placing the design in Device, while optimising it for Speed or Area
Libraries (Simprims, Unisims)
Programming File Generation
Bit Stream
Download into FPGA / CPLD
using JTAG, Serial Prom, Parallel configuration ....etc
Logical Design
Physical Design

Fig. 3. Programmable logic design flow.

IV. Simulations and Synthesis Results

Simulations are run for reconfigurable switch architecture having 8 inputs and 8 outputs nodes using 2×2 switch elements. The Fig. 4 identifies the results derived from simulation. The architecture is synthesized and the area, delay and power is measured using Design Compiler. Table-I gives the performance of reconfigurable switch architecture in terms of area, power and delay and Table-II shows the area, power and delay for individual configurations. The bar graphs shown in Fig. 5, Fig. 6 and Fig. 7 show the area and power comparison between the combined area of four individual architectures and proposed architecture. The delay of proposed architecture is compared with individual delay of each configuration. The graphs show that the architecture four has a minimum area, power and delay compared to other architectures shown.

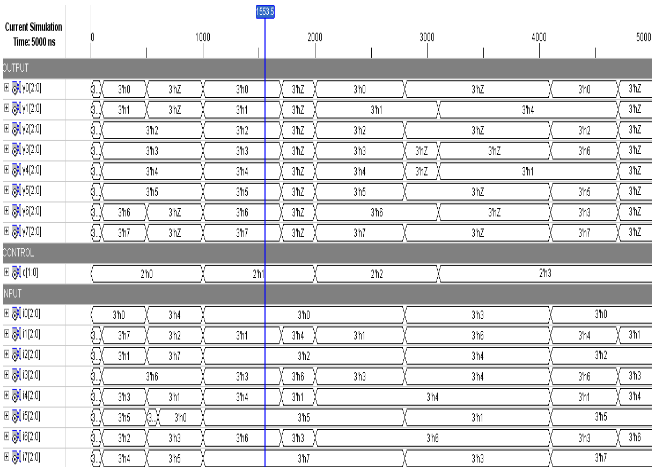

Fig. 4. Simulation results for reconfigurable switch architecture [internal routing].

TABLE I: Comparison of Reconfigurable Architectures [Internal Control]

	RECONF. ARCH1		RECONF. ARCH2		RECONF. ARCH3	
	compile	Compile ultra	compile	Compile ultra	compile	compile ultra
AREA [µm2]	1250	896	1250	873	1232	896
DELAY [ns]	33.47	61.36	33.47	46.9	33.47	61.36
POWER [µw]	929.80	752	873.38	683	925	752

TABLE II: Comparison of Individual Architectures [Internal Control]

	BANYAN		**CUBE**		**OMEGA**		**INDIRECT CUBE**	
	c	cu	c	cu	c	cu	c	cu
AREA [µm2]	780	588	780	588	780	588	780	588
DELAY [ns]	21.4	39.4	21.4	39.4	21.4	39.4	21.4	39.4
POWER [µw]	312.9	257.	312.9	257	312.9	257	312.9	257

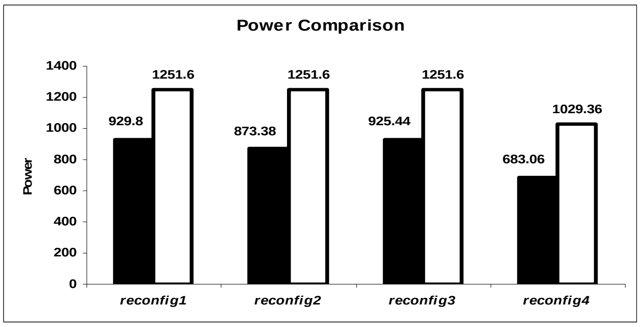

Fig. 5. [internal control]power comparison [µw]

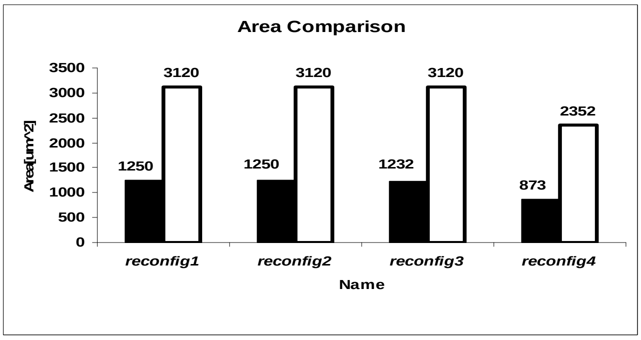

Fig. 6. [internal control] area comparison [µm^2]

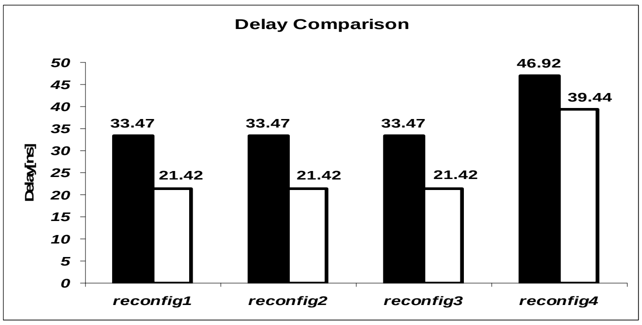

Fig. 7. [internal control] delay comparison [ns]

V. Conclusion and Future Work

An efficient architecture and VLSI implementation of the reconfigurable switch for next generation communication network is achieved and is successfully burned semi-permanently according to the board assembly. Four cases have been implemented and VLSI performance factors are measured. The performance and bar graphs indicate the comparison for area, delay and power for four different reconfigurable architectures. Since all four merged architectures share similarities in their structures, the potential is given for efficiently combining them in a single reconfigurable VLSI circuit, leading to competitive designs in terms of area, performance, and power consumption.

This gives evidence that function-specific reconfigurable circuits can achieve considerable improvements in at least one design objective with only a moderate degradation in others. Concentrating on blocking factor of the present reconfigurable switch architecture, dynamic behavior can be explored, leading to future work.

REFERENCES

[1] Sadiku and N. O. Mathew, "Next Generation Network," *IEEE Potentials*, vol. 21, pp. 6-8, April 2002.

[2] M. H. L. Rao and R. Tripathi, "Design of reconfigurable switch architecture for next generation communication networks," in *Proc. of Tencon. Conf. Emerging Technologies for Sustainable Devlopment*, Singapore, 2009, pp. 1-6

[3] V. S. Tripathi, "Performance evaluation of a multistage switching fabric for next generation networks," Ph. D thesis, Dept Electronics and Communication Eng., NIT Allahabad, Allahabad, India, 2007.

[4] Y. Audi, S. Meftali, and M. Abid, "Design and performance evaluation of a reconfigurable delta MIN for MPSOC," *IEEE-ACM-Dec* 2007, 978-1-4244-1847-3/07.

[5] E. L. horta and S. T. Kofuji, "A run time reconfigurable ATM switch," in *Proc. of IPDPS 02.16th International Parallel and Distributed Processing Symposium*, USA, 2002, pp. 41.

[6] All Programmable Technologies from Xilinx Inc. [Online]. Available: http: www.xilinx.com

[7] H. J. Siegel, W. C. Nation, C. P. Kruskal, and L. M. Napolitano, "Using the multistage cube network topology in parallel supercomputers," in *Proc. of IEEE*, vol. 77, no. 12, pp. 1932-1953, Dec. 1979.

[8] J. H. Park, H. Yoon, and H. Lee, "The deflection self routing banyan network: A large –scale ATM switch using the fully adaptive self routing and its performance analysis," in *Proc. of IEEE ACM transactions on Networking*, vol. 7, no. 4, August 1999.

[9] A. Tobagi, "Fast packet switch architectures for broadband integrated services digital networks," in *Proc. of the IEEE*, vol.78, no.1, pp. 133-67, Jan. 1990.

[10] T. Y. Feng, "A survey of interconnection networks," in *Proc. of IEEE*, vol. 14, pp. 12-27, Dec 1981.

[11] H. J. Siegel, *Interconnection Networks for Large Scale Parallel Processing: From Theory and Case Studies*, McGraw-Hill Publishing Company, 1990.

[12] J. Cavanagh, "Digital design and verilog HDL fundamentals," *CRC Press Taylor and FrancisGroup*, 2008.

[13] Installing the Software – Synopsys. [Online]. Available: http: //solvnet.synopsys.com.

M. Hema Lata Rao was born in the Siliguri district of West Bengal, India, in 1981.She has received her B-tech degree in Electronics and Instrumentation Engineering from Biju Patnaik University of Technology, Orissa, India in july 2003.She is presently working towards her PhD degree in the Department of Electrical Engineering, Indian Institute of Technolgy, Bhubaneswar.Her present research interests include wide band gap materials and defects levels in semi-insulating substrates.

Rajeev Tripathi received the B. Tech. degree from Allahabad University , Uttar Pradesh, in 1986; M. Tech from Allahabad University in 1992; and Ph.D. degree from Allahabad University, Allahabad, India in 1998.He worked as a Technical Officer (Tele) in Cabinet Secretariat, Government of India from 1990 to 1991. From 1995 to 1999, he has worked as Senior Lecturer in the Electronics Engineering Department, MNNIT, Allahabad, India. From 1999 to 2002, he has worked as Senior Lecturer in the Electronics Engineering Department, MNNIT, Allahabad, India From 2002 to 2004, he has worked as Lecturer in the Department of Electrical and Computer Engineering, The University of The West Indies, St. Augustine, Trinidad (WI) (On leave from MNNIT, Allahabad, India). Since 2005 he has been working as a Professor in the Department of Electronics and Communication Engineering, Motilal Nehru National Institute of Technology, Allahabad-India Prof. Rajeev Tripathi has published more than 31 research papers in international journals and conference proceedings.

His research interests include Data Communication Networks , High Speed Communication Networks , Mobile Communication and Mobile Ad-hoc Networking.