Synthesis of VHDL Code for FPGA Design Flow Using Xilinx PlanAhead Tool

Md. Abdul Latif Sarker, Moon Ho Lee
Division of Electronics & Information Engineering Chonbuk National University
664-14 1GA Dekjin-Dong
Jeonju City, Jeonbuk
South Korea 561-756
{latifsarker, moonho}@jbnu.ac.kr

***ABSTRACT*:** *This paper addresses a synthesis process of VHDL code for FPGA design flow using Xilinx PlanAhead tool. This tool provide a low power profile, more hard IP functionality, create a global timing constraint, lower node capacitance & architectural innovations, cost of development, very high DSP performance hardware solutions and easily can be evolutionary algorithms, reconfigured to the development of whole compiler, simulation and synthesis frameworks. It is handle dense logic and memory elements offering very high logic capacity. The logic blocks are replicated in FPGA with interconnects and input-output blocks. This approach attached a new created VHDL code and generate of register-transfer level (RTL) hardware description language (HDL). In this paper, we have presented the FPGA approach of interconnection and its flexibility on example through synthesis process, simulations and implemented results are detailed.*

Keywords: FPGAs Architecture and Design Techniques, Xilinx PlanAhead Tool Based FPGA Design Flow and Steps, Synthesis Techniques and Simulations, Implementation

Received: 18 July 2012, Revised 11 September 2012, Accepted 19 September 2012

1. Introduction

Field Programmable Gate Arrays (FPGA's) are 2D array of flip-flops and logic blocks with an electrically programmable interconnection between logic block's. The logic blocks are implemented using multiple low fan-in gates and it's user a way to configure the interconnection between the logic block's and function of each logic blocks. The special processing or intellectual property (IP) units can be implemented at a given time and dynamically replaced in the name of reconfiguration [2] and [3]. In Xilinx logic block lookup table is used to implement any number of different functionality. It is as hardware platform in reconfigurable applications which gives an attractive hardware solution. FPGA vendors have multiple device product families which has in available resources, package and speed grade. For example: Xilinx has two FPGA more recent families Spartan-6 and vertex-6 [1] which targets, low power, highest performance, reduce global timing, and gate density. In Xilinx PlanAhed tool, the RTL can be synthesized into device specific logic resources in HDL for FPGA's design [4] and [5]. The registers or reallocation of unit delays throughout a circuit in order to reduce the number of combinational logic levels. There is an internal correlation between the largest number of logic levels and the frequencies.

The well-known mean of exchange between users and developers (or designers) used to collect requirements, is scenario. There are This paper organized as below: Section II gives an overview of FPGAs architecture and design techniques. Section III describes our Xilinx PlanAhead tool using FPGA design flow and steps. The synthesis and simulations are given in section IV. The

implementation results on Xilinx vertex-6 FPGA are presented in section V. Conclusions are given in section VI.

2. FPGA Architecture and Design Techniques

FPGA consists of I/O resources, logic resources or a group of Configurable Logic Block's (CLB's), clock resources, memory resources etc. Resource blocks are detailed in below:

2.1 I/O Resources

The I/O resources mean the interconnection of FPGA**.** It is provide extra automation software solutions for I/O design process and easier to I/O design are more flexible. All I/O's are on the edge of the chip and it is grouped into some banks. Each bank has 30-40 I/O blocks and eight clock pins per edge. The differential receiver and on-chip termination is available in all banks. Each I/O pair has two IO logic blocks as master and slave as shown in Figure 1. It can be operate independently. Every I/O logic block contains I/O serial to parallel or parallel to serial converter and DE-serializer (IOSERDES), IO delay and memory resources.

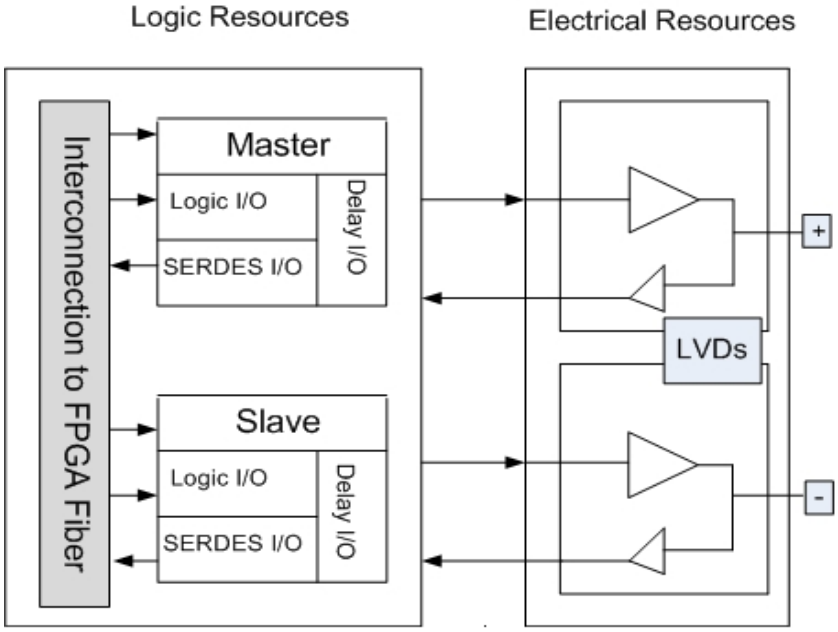

Figure 1. Block Diagram of I/O Resources

2.2 Logic Resources

The input lines go into the input and enable of lookup table. The output of the lookup table gives the result of the logic function that it implements in Figure 2. Lookup table is implemented using SRAM.

The CLB's contained slices, logic cells, lookup table, flip-flops connected through intra-interconnect. The CLB's access the interconnect fabric through Connection Block's (CB's) and CLB's wire are interconnected through switch matrix blocks [6].

There are two types of slices in vertex-6 FPGAs like as sliceM and sliceL. SliceM is the full slice (25%) whose lookup table can be used for logic and memory. The sliceL is a logic and arithmetic slice (75%) only. Its lookup table can be used for logic not memory.

2.3 Clock Resources

The clock resource has high fan-out clock distribution buffer, regional clock routing ability, clock regions, global clocking and independently programmable clock outputs. In vertex-6 FPGAs, the Clock Management Tile (CMT) has two PLL (Phase Locked Loop) and up to nine CMTs per device. This block performs frequency synthesis, clock de-skew and jitter filtering. The special clock network dedicated to IOLOGIC resources and multiple sources for clocking IOLOGIC like as BUFIO2 and BUFPLL as shown in Figure 3.

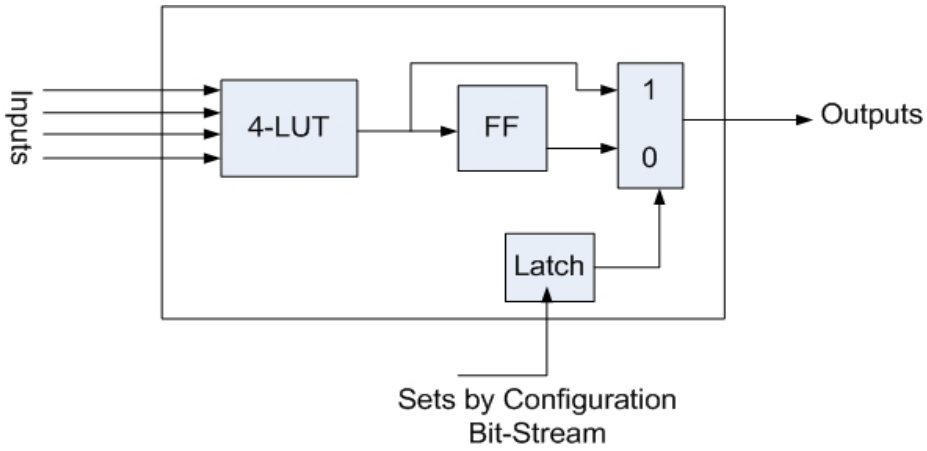

Figure 2. (a) 4-input Lookup Table and Sets by Configuration Bit-Stream

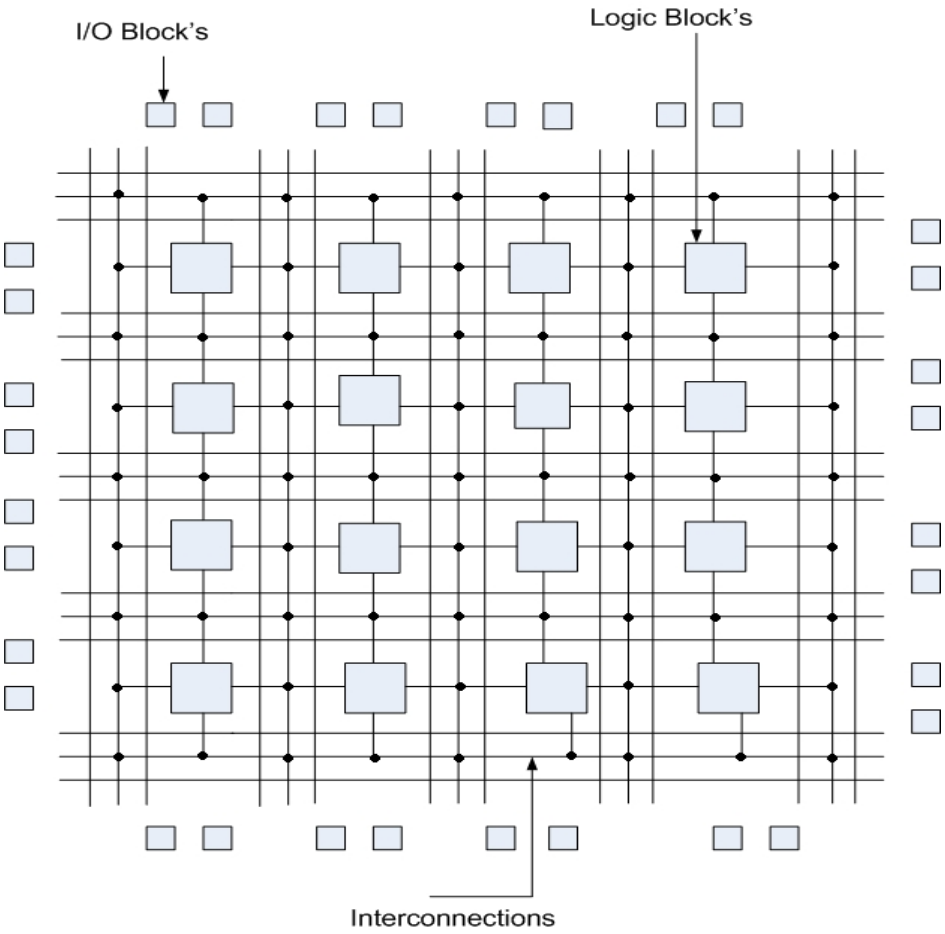

Figure 2b. Input/output Block's, Configurable Logic Block's and Interconnection of FPGA's

3. Xilinx PlanAhead tool Using FPGA design flow and steps

Xilinx PlanAhead version 13.1 contains RTL design, synthesize, netlist design, implemented design, chipScope analyzer, iMPACT tools and bit stream generator. It is also included external IP catalog and iSIM tools. The VHDL code may contains either Xilinx IP cores to the FPGAs. A group of Xilinx libraries that indicate how each of the higher level blocks should have must to the

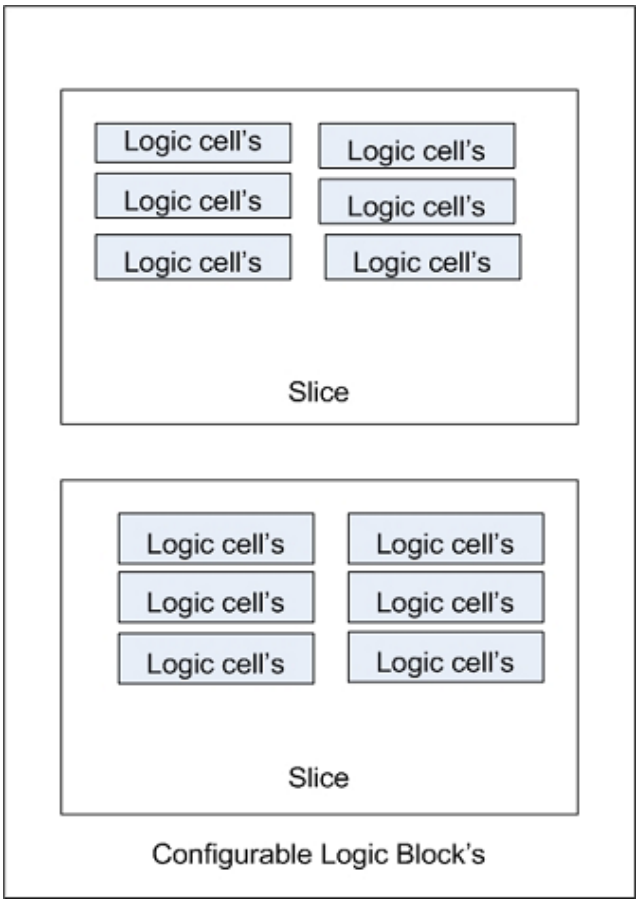

Figure 2c. Internal structure's of CLB's for vertex-6

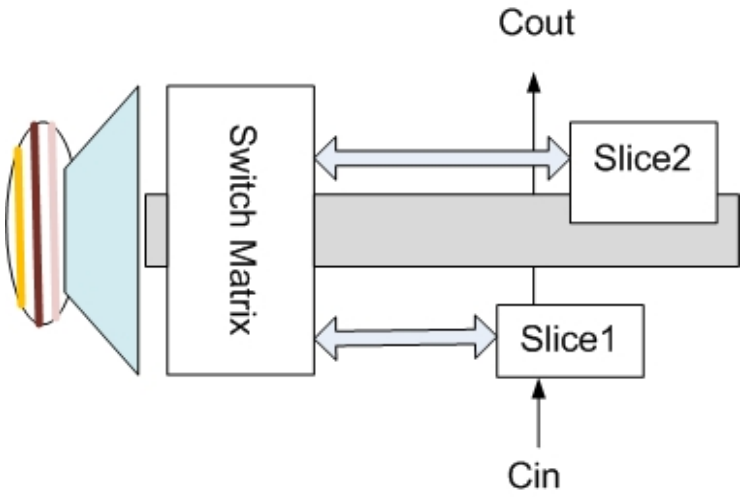

Figure 2d. CLB's contains two slices

Figure 2. Logic Resources and interconnection of FPGA's

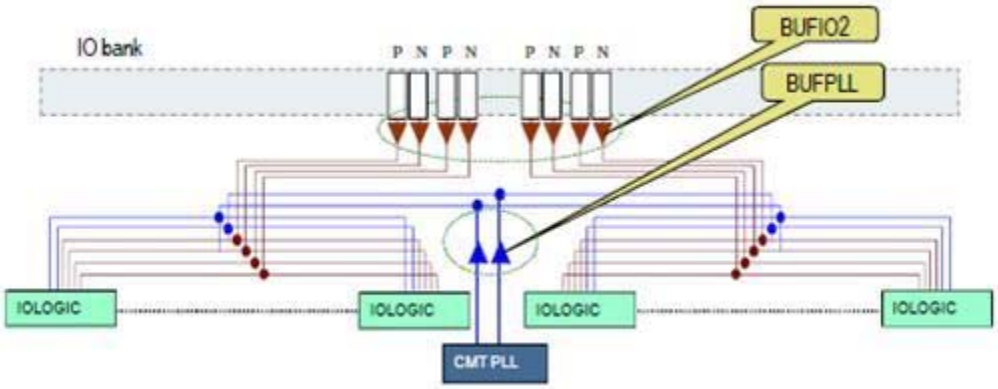

Figure 3. Clock Resources

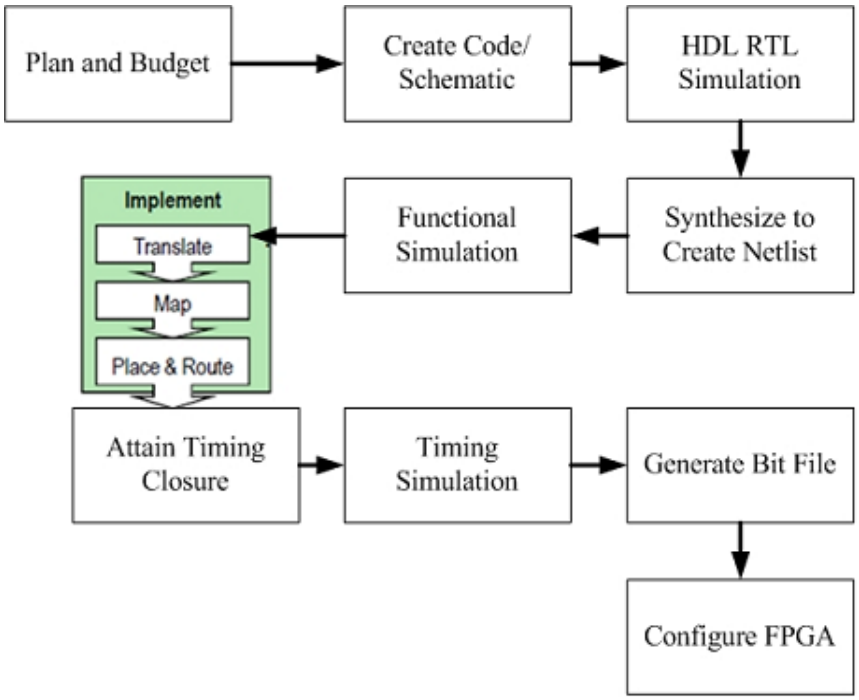

Figure 4. PlanAhead Tool Based FPGA design Flow

In Figure 4, the implementation step process a netlist files which consists of translate, map and place & route phases. The translate step merge multiple design file into a single netlist. The map step group logical symbols from the netlist into physical components (Slice and IOBs) specific to target device. The place & route step place components onto the chip, connect through

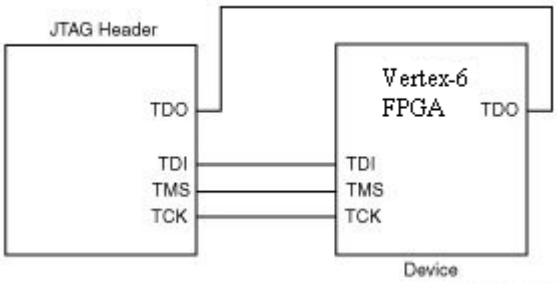

Figure 5. Download Final Bit Files by JTAG Interfaces

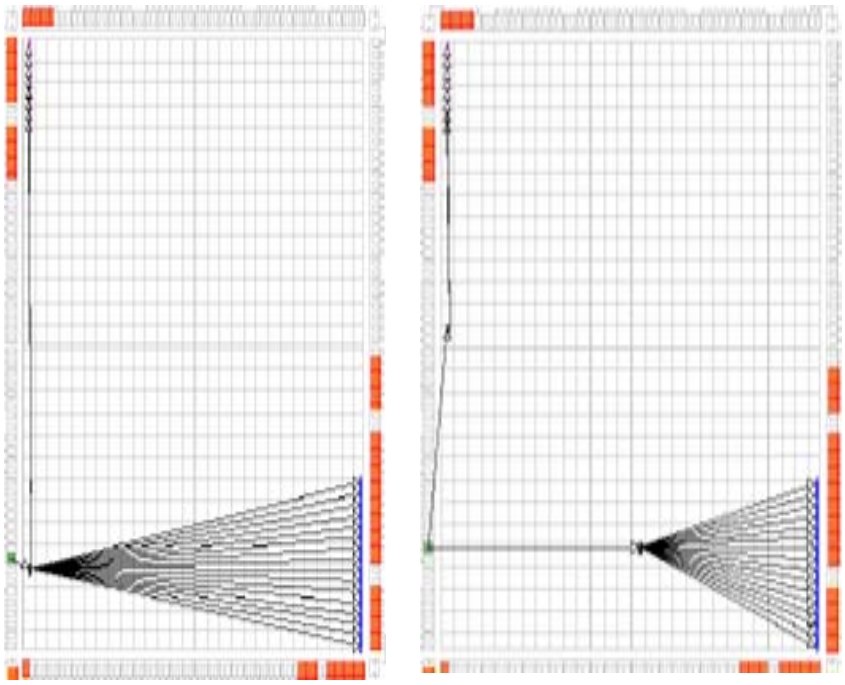

Figure 6. FPGA design Technique using Duplicating Flip-flops process

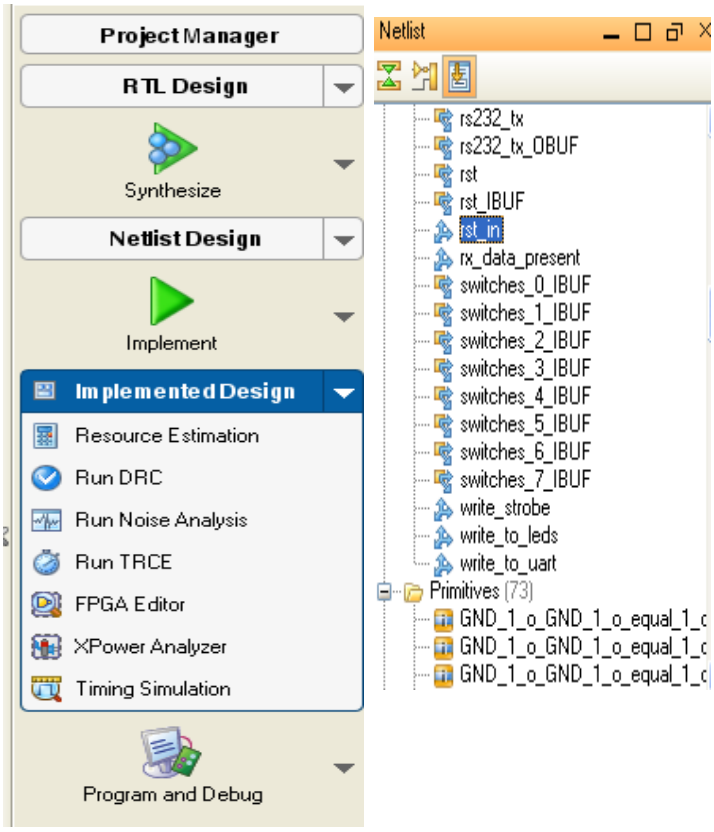

Figure 7. Project manage and synthesis environment using Xilinx PlanAhead

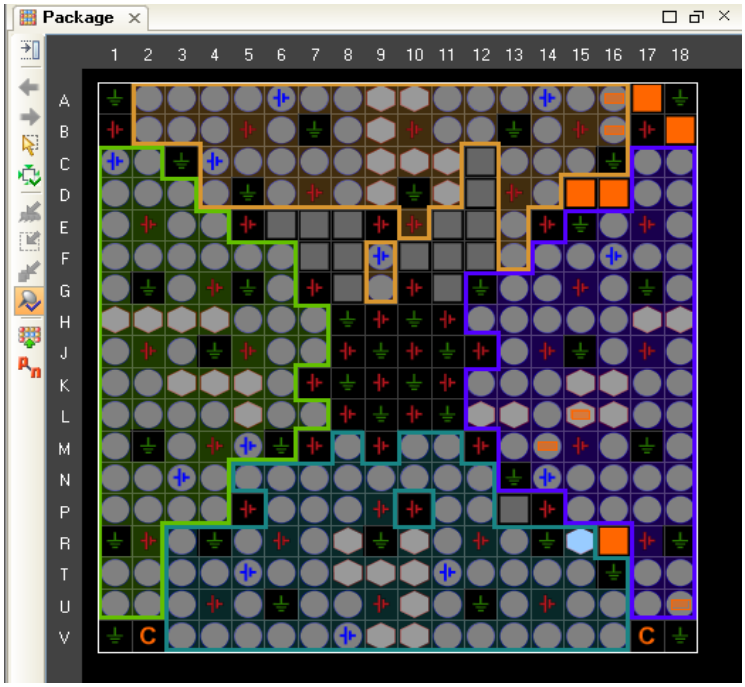

Figure 8. Device package with blocks

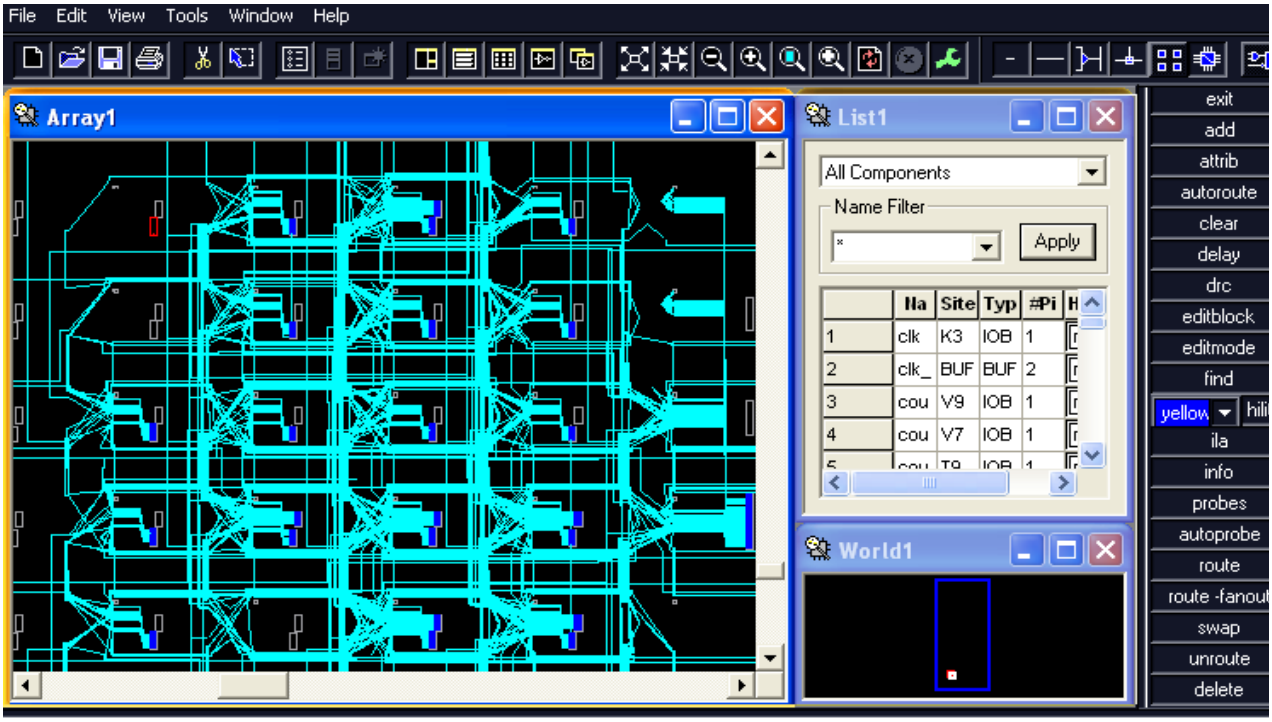

Figure 9. Edited gate arrays by FPGA editor

the switch matrix as shown in Figure 2(d), dedicated routing lines and extract timing data into reports. These extracting reports contain design information like as timing analyzing, floorplanning, FPGA editing, X-powering, slacks histogram conditions etc. After completing this implementation and generate a bit (.bit extension file) file need to testing & verification through HDL simulator. When the design had been totally verified with exact timing, the final bit file will be generated. This bit file can be downloaded directly into the FPGA or through JTAG interface [7] and as shown in Figure 5.

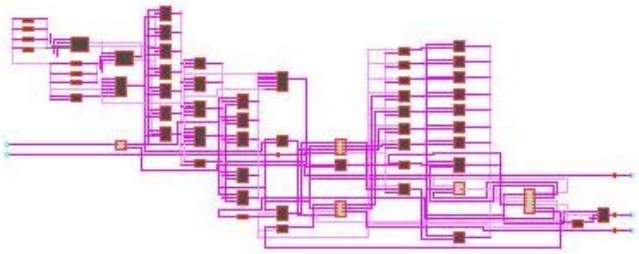

Figure 10. Edited arrays in FPGA

Clock Regions

	Id	Name	Row	Column	I/O Banks
	1	X0Y0	0	0	2, 3
	2	X0Y1	1	0	
	3	X0Y2	2	0	3
	4	X0Y3	3	0	3
	5	X0Y4	4	0	3
	6	X0Y5	5	0	3
	7	X0Y6	6	0	
	8	X0Y7	7	0	0, 3
	9	X1Y0	0	1	1, 2
	10	X1Y1	1	1	1
	11	X1Y2	2	1	1
	12	X1Y3	3	1	1
	13	X1Y4	4	1	1
	14	X1Y5	5	1	1
	15	X1Y6	6	1	
	16	X1Y7	7	1	0, 1

Table 1. Clock Regions

User Input Data	Confidence	Details	Action
Design implementation state	High	Design is completely routed	
Clock nodes activity	High	User specified more than 95% of clocks	
I/O nodes activity	High	User specified more than 95% of inputs	
Internal nodes activity	Medium	User specified less than 25% of internal nodes	Provide missing internal node
Device models	High	Device models are Production	
Overall confidence level	Medium		

Table 2. Confidence Level

	Type	Slack	...	From	To	Total Delay[1]	Logic Delay	Net Delay	Logic %	Net %
Path 39	SETUP	6.076		my_program/...	in_port_0	11.732	3.475	8.257	29.6	70.4
Path 40	SETUP	6.076		my_program/...	in_port_0	11.732	3.544	8.188	30.2	69.8
Path 38	SETUP	6.068		my_program/...	in_port_0	11.74	3.48	8.26	29.6	70.4
Path 37	SETUP	6.055		my_program/...	in_port_0	11.753	3.544	8.209	30.2	69.8
Path 36	SETUP	6.046		my_program/...	in_port_0	11.755	3.693	8.062	31.4	68.6
Path 35	SETUP	6.017		my_program/...	in_port_0	11.784	3.538	8.246	30	70
Path 34	SETUP	6.008		my_program/...	in_port_0	11.8	3.481	8.319	29.5	70.5
Path 33	SETUP	6.007		my_program/...	in_port_0	11.801	3.481	8.32	29.5	70.5
Path 31	SETUP	5.934		my_program/...	in_port_0	11.867	3.475	8.392	29.3	70.7
Path 32	SETUP	5.939		my_program/...	in_port_0	11.869	3.693	8.176	31.1	68.9
Path 30	SETUP	5.923		my_program/...	in_port_0	11.885	3.63	8.255	30.5	69.5
Path 29	SETUP	5.918		my_program/...	in_port_0	11.89	3.544	8.346	29.8	70.2
Path 28	SETUP	5.879		my_program/...	in_port_0	11.929	3.693	8.236	31	69
Path 27	SETUP	5.864		my_program/...	in_port_0	11.937	3.63	8.307	30.4	69.6
Path 26	SETUP	5.863		my_program/...	in_port_0	11.945	3.481	8.464	29.1	70.9
Path 25	SETUP	5.717		my_program/...	in_port_0	12.091	3.63	8.461	30	70
Path 24	SETUP	5.665		my_program/...	in_port_0	12.143	3.693	8.45	30.4	69.6
Path 23	SETUP	5.648		my_program/...	in_port_0	12.16	3.481	8.679	28.6	71.4
Path 22	SETUP	5.636		my_program/...	in_port_0	12.172	3.538	8.634	29.1	70.9
Path 21	SETUP	5.618		my_program/...	in_port_0	12.19	3.63	8.56	29.8	70.2
Path 20	SETUP	5.617		my_program/...	in_port_0	12.191	3.63	8.561	29.8	70.2
Path 18	SETUP	5.443		my_program/...	in_port_0	12.365	3.475	8.89	28.1	71.9

Table 3. Implemented Design Report From Planahead Tool

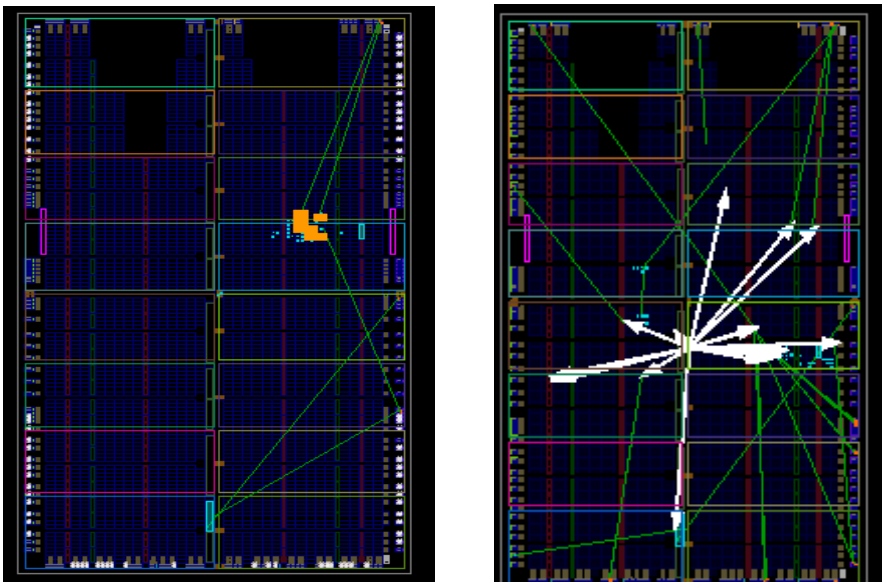

Figure 11. Interconnection and LUT utilization per CLB in FPGAs

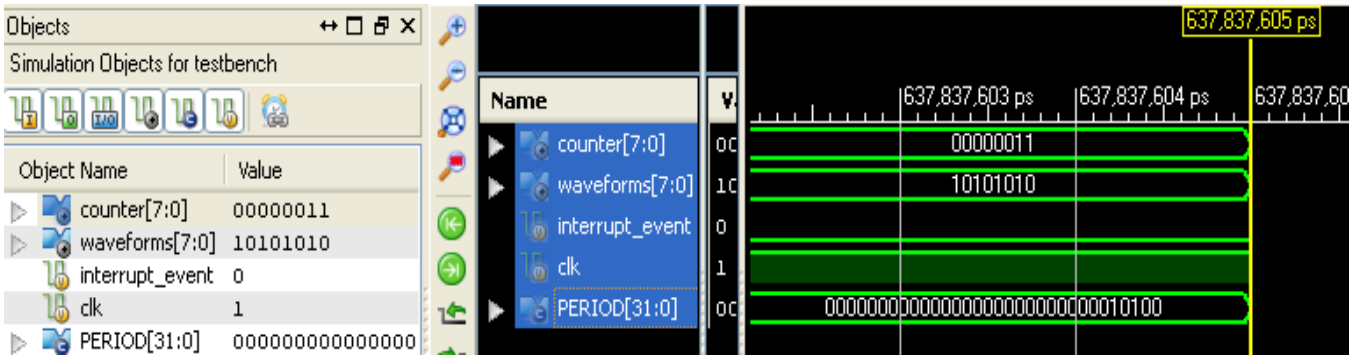

Figure 12. Simulation results of output ASIC data

There is a various type of FPGA design techniques like as duplicating flip-flops, pipelining, I/O flip-flops, circuit synchronizations etc. One of the examples of duplicating flip-flops process in FPGA design technique, if high fan-out nets can be slow and hard to route, the duplicating flip-flops can be fixed both problems and gives the better design trade-off. In Figure 6, the source flip-flops drives two register banks constrained to different region of the chip and the source flip-flops has been duplicated.

4. Synthesis Techniques and Simulations

Synthesis includes generic optimization, slack optimizations, power optimizations followed by placement and routing. Implementation includes partition, place and route. The output of design implementation phase is bit-stream file. In Figure 7, the project manager manages sources, customize IP and view project details in the project summary. After synthesis, the flow manager includes report timing, slacks histogram in Figure 13, and chipScope pro utilities. The simulated gate arrays as shown in Figure 9 and Figure 10 which can be easily reconfigure, replace and interconnect by included Xilinx PlanAhead FPGA editor. The interconnection and LUT Utilization per CLB in FPGAs and Simulation Results of output ASIC data and VHDL module of activation process as shown in Figure 11 and Figure 12.

5. Implementation

The synthesis output files that are required in PlanAhead tool are in various data format like as EDIF or NGC and UCF (User

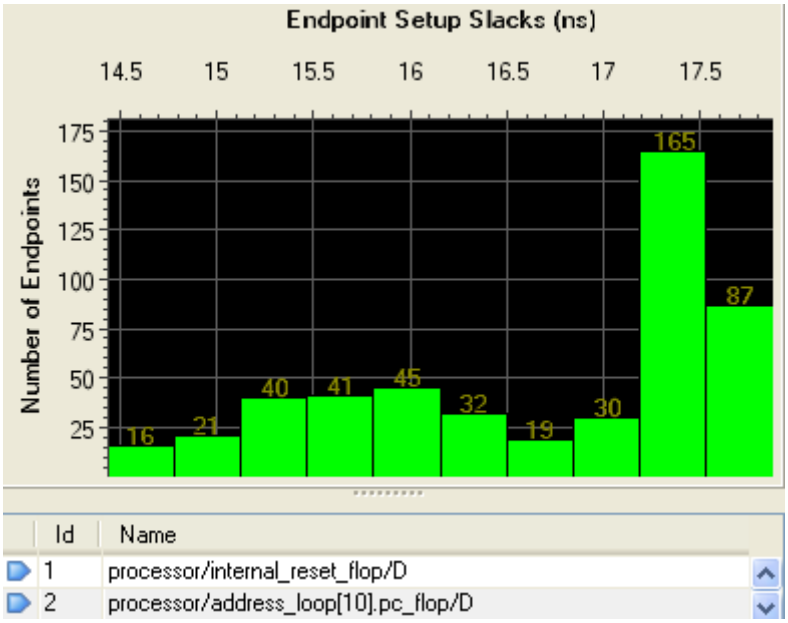

Figure 13. Slacks Histogram Results Report

Constraint File) formats which need to netlist of integrated design, timing constraint, FPGA pin assignment in Figure 8 respectively. After synthesize, the clock resources, confidence level and implemented design report results are given in TABLE 1, 2 and 3. In Figure 11, Interconnection and LUT Utilization per CLB in FPGAs. The output waveforms are tested in ASIC VHDL module of activation process. Bit stream file is fed to a simulator which simulates the design functionality and reports errors in desired behavior of the design. Timing tools are used to determine maximum clock frequency of the design. Now the design is loading onto the target FPGA device and testing is done in real environment.

6. Conclusions

In this paper, we presented a new dynamic interconnection for reconfigurable in FPGAs and the basic concept of interconnection. Xilinx PlanAhead tool can be easily synthesis and presents good implemented results. The advantages of PlanAhead tools are high performance, flexibility, code processing speed and the possibility of dynamic resources placement of module at run-time.

7. Acknowledgment

This work was supported in part by the second stage of Brain Korea 21 Project in 2012 (NRF), World Class University (R32-2010-000-20014-0, NRF, Korea), Fundamental Research (2010-0020942, NRF), and Mid-career Researcher Program (National Leading Research Lab, 2012-002521, NRF and MEST) Republic of Korea.

References

[1] www.xilinx.com.

[2] Gaccione, S.A., Levi, D. (1999). The Advantage of Run-time Reconfiguration, *In*: Jhon Schewel et al editor, Reconfigurable Technology FPGA's for computing and Application, *In*: proc. SPIE 3844, 87-92, Bellingham WA, Sep.

[3] Lysaght, P., Dunlop, J. (1993). Dynamic Reconfiguration of FPGA's in W. Moore, W. Luk [Ed's] more FPGA's proceeding of the international workshop on Field Programmable Logic and Applications, 82-94.

[4] DiMicheli, G. (1994). Synthesis and optimization of Digital Circuits, McGraw-Hill.

[5] Mohd Fadzil, Majid S. Naghmash, Chye, Y. H. (2010). Synthesis of HDL Code for FPGA Design Using System Generator, European Journal of Sciencefic Research, 45 (1) 111-121.

[6] Slavisa Jovanovic, Camel Tanougast, Serge Weber. (2007). A New High- Performance Scalable Dynamic Interconnection for FPGA –based Reconfigurable System, 1-4244-1898-5/08/2008 IEEE. Xilinx ISE 9.2i software manual, Constraint Guide, and Development System Reference Guide, Xilinx Inc.