Design Preservation Methodology based on FPGA

Bin Bao [1, a], Xuenan Cui [2,b] and Ning Lei [3,c]

[1] Dept 7. Beijing Institute of Space Mechanics and Electricity, Beijing, China

[2] Dept 7. Beijing Institute of Space Mechanics and Electricity, Beijing, China

[3] Dept 7. Beijing Institute of Space Mechanics and Electricity, Beijing, China

[a]baobin4012@163.com, [b]379316898@qq.com, [c]nwpu_ln@sina.com

Keywords: Back-end Tool; Directed Routing Constraints; Place&Route; Design Preservation

Abstract. More and more (Field Programmable Gate Arrays)FPGAs were used in space missions, due to the characteristics of aerospace equipment ,such as satellite. The correctness of FPGA design was not only determined by the code, but also need FPGA design suite to complete the process which generates the circuit, only the final circuit would ensure the efficiency of the design. In this paper, two design methodology based on FPGA were summarized, they were back-end tool Directed Routing Constraints and Place&Route Tool Design Preservation. The information of place and route about the key signal or the unchanged design part would be preservatized through these design methodology. Thus it would greatly reducing the impact of the upgrade to the fully tested design project due to partial modules. By using these methodologies, iterative development process of FPGA project would be improved effectively, and the testing process of FPGA project would be optimized.

Introduction

Field Programmable GateArrays (FPGA) was a programmable signal processor. Users could change the configuration which defined the real function of FPGA, to meet the design requirements. Compared with the traditional digital circuit system, FPGA had advantages of programmable, high integration, high speed, and high reliability, etc. The original PCB level design could implement on the chip by configuring the internal logic and the input/output port of the fpga, to improve the circuit performance, reduces the workload and difficulty of PCB design, effectively increase flexibility of the design. More and more FGPAs were used in space missions due to the characteristics of the equipment such as satellite, which are small batch, high degree of customization, etc. As the development of semiconductor technology more and more functional modules were integrated in to FPGA, such as embedded processor, lots of RAM, Advanced DSP slices, High-Speed I/O module, etc. FPGA was very good to meet the needs of space missions because of abundant resources and reconfiguration

With the application of large-scale FPGA in engineering projects, in the FPGA development process, designers often needed to modify or update several modules of the whole project. In engineering projects, FPGA usually had some external devices which are very sensitive to the signal from FPGA. At this time, if the entire project was recompiled, replaced and rerouted, it was very easy to have an impact on the well-done high precision signals. Secondly, rerun the whole project would bring a lot of time wasting, low efficiency in FPGA project update. This paper introduces two design methodologies in the follow-up, to reduce the impact on the fully test design for updating of several modules

Two design methodologies would be introduced in the 2、3 sections. The influence on the design of two methodologies was analyzed in detail. At the end, two methodologies were summarized, advantages and disadvantages of the two methodologies were compared. A reference about which methodology was suitable for the different types of engineering needs was provided.

Back-end tool directed routing

Back-end Tool was a tool which is integrated in EDA of fpga. As a graphical application interface, Back-end Tool was used to display the results of the underlying Place&Route and configure the FPGA. Critical components could be manually placed and routed before running the automatic place and route function. The place and route result could also be manually adjusted to meet the design needs after the automatic place and route.

Back-end Tool had a function of generating routing constraints directly. Back-end Tool could read the net-list file of FPGA after place and route. So, routing information of any line in the net-list file could be extracted by Back-end Tool. The extracted information was stored in the constraint file .

```
NET "gen_calib[0].u_phy_calib/cal1_idel_dec_cnt_share0000<5>"
ROUTE="{3;1;5vlx110tff1136;380b5424!-1;-86824;207296;S!0;-1136;929!1;"
"1317;-2021!2;3192;796!3;683;-88;L!}";
INST "gen_calib[0].u_phy_calib/Maddsub_cal1_idel_dec_cnt_share0000_xor<6>" LOC=SLICE_X19Y143;
INST "gen_calib[0].u_phy_calib/cal1_idel_dec_cnt_mux0001<5>1" LOC=SLICE_X20Y143;
INST "gen_calib[0].u_phy_calib/cal1_idel_dec_cnt_mux0001<5>1" BEL="D6LUT";
```

Fig. 1 Constraint of FPGA

Back-end routing information was preserved in internal format of FPGA company. While the components associated with this line were fixed in the corresponding position in terms of constraints. This would guide the place and route tools of EDA to process this line in this way.

Routing information generated in this way was opacity. If a large number of routing in the project were maintained in this way. First, for a design engineer, it would be a very difficult task to maintain the correspondence of routing constraints with the code which are large and unreadable. Second, it was difficult to implement for the back-end Place&Route tool of EDA. Because Place&Route tool would face a very big challenge in the timing closure and area closure when completing the remaining part of the design as a result of these too large immovable designs. Therefore, this methodology should be used with caution.

It was recommended that, before using this method, the drive signals which devices should be particular concerned must be picked out in the early project stage. Then during code design stage, if possible, these drive signals were recommended to output by registers directly and add some additional registers at the final output. An output register sharing by multiple pins should be avoided, because it wasn't conducive to the adjustment of the sharing register. If the combination logic must to output directly, it was recommended to be implemented with the primitives of Xilinx. So it would be convenient for constraint on the signal.

In several actual projects, both registers output and combination logic output were existing. Engineering requirements were met very well by using back-end tool directed routing. In the process of engineering practice such routing constraints were controlled below 40. It didn't cause too much pressure for back-end place&route tools of EDA. All signals of device including constrained signals successfully completed place and route. And it was proved that such methodology was very effective by the actual testing on the hardware.

Design Preservation based on Place&Route Tool

The flow of the design preservation based on Place&Route Tool was more complex than back-end tool directed routing. The original design flow would cause some changes by adopting design preservation based on Place&Route Tool. Designers needed to adapt the changes in the flow, in order to achieve design preservation.

Design preservation would be realized very well by using back-end tool directed routing if only a small amount of signals needed to be kept. When the bottleneck encountered in the project was the internal module, and there were a lot of signals, back-end tool directed routing was not very good to achieve design preservation. Design preservation based on Place&Route Tool could be considered at this time. Adopting the design preservation based Place&Route Tool, it needed to be considered at the beginning of design planning, not after the design had begun to yield in consistent results. Partitions comprised the underlying technology in the design preservation flow, and follow the

logical hierarchy of an HDL design. Therefore the design preservation flow worked best with designs that follow good hierarchical rules. The following optimization limitations were inherent to the insulation created by partitions.

1) No Optimization Across Partition Boundaries adjust the template as follows.

2) If inputs to a partition are tied to a constant value in order to guide optimization, the constant cannot be pushed across the partition boundary. No optimization occurs. This can occur when a constant is used to enable or disable a specific feature in a core or module.

3) There is no optimization of unconnected partition outputs. If the output of a partition does not drive any other element.

4) Logic From One Partition Cannot be Packed With Logic From Another Partition .

Although design preservation based on Place&Route Tool had deficiencies, but it could get better design inheritance, to facilitate the iterative upgrade project.

Change of the flow about design preservation based on Place&Route Tool was mainly reflected in the synthesis of design. Using this methodology, incremental synthesis flow must be adopted. Tool provided by Xilinx was not very good for incremental synthesis flow, could only support the latest family of FPGA devices. So this paper selected Synplify of Synopsys.

"Use Partition Flow"option of Synplify and compile points of modules should be set. A compile point was a module that is treated as a block for incremental mapping. In subsequent synthesis iterations, the software did not resynthesize the compile point unless the original RTL netlist for the compile point changes. If only a part of the design changed it was obvious that it would only affect the change part of the module, the other part of the design would not have a little impact. Firstly, from the synthesis level, it completed the design preservation. Then it could make the netlist of unchanged RTL modules exactly the same as the previous synthesis result. Secondly, in iterative engineering process, especially during late-stage , it could greatly shorten the time of synthesis by using this methodology because the project was comparatively mature and changes was relatively small.

Not only netlist after synthesis needed to be concerned, it should pay more attention to whether the circuit after place and route is retained So after synthesis by Synplify, subsequent place and route preservation should be completed using Place&Route tool.

The synthesized netlist was imported into Place&Route tool. And Pblocks was set for modules which compile point had been set. The purpose of setting Pblocks was to constraint place and route resource in the region, so the changes on these modules would not affect the place and route of other circuits. After setting Pblocks, subsequent place and route could be started by the Place&Route tool. After place and route, Place&Route tool reported implementation status of each module. It could be viewed through the report that whether the place and route results of each module were completely inherited. In the process of development, after place and route, if the result met the requirements after checking all reports. Designers could download the bit file into FPGA for debugging. If it passed the hardware debugging, without design preservation design project was backed up simply. When the module was upgraded slightly, all modules in the project should be resynthesized, replaced and rerouted completely. With design preservation design needed to be promoted by Place&Route tool. When the module was upgraded slightly, only changed module was resynthesized, replaced and rerouted, it didn't affect the other module in any way. So the modules which had been fully tested would be well applied, only changed module needed to be fully tested again.

In the actual project, some modules, such as various algorithm modules, often needed to be upgraded to improve the performance of the algorithm after debugging, even after system debugging, because of inadequate modeling conditions. Usually the driving circuit of AD device even driving circuit of sensor existed with algorithm modules in the project at the same time, as figure 2.

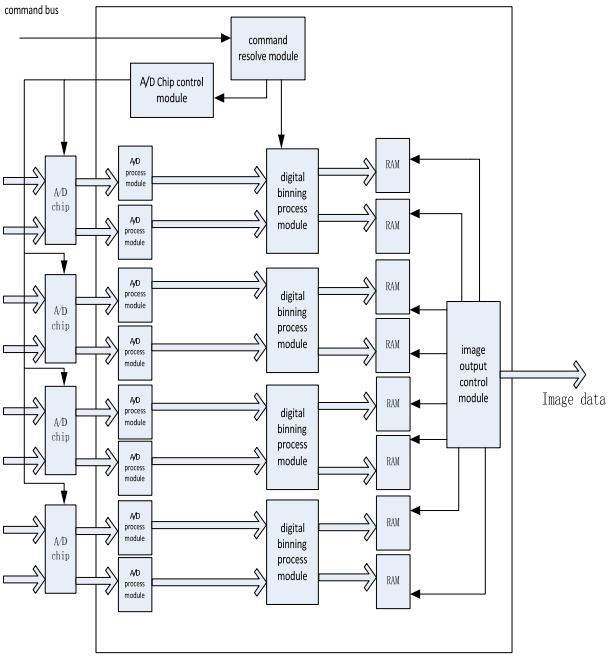

Fig. 2 Diagram of signal-processing system

Those driving circuits might had subtle changes if all modules were resynthesized, replaced and rerouted using the old previous method. Some algorithms were very sensitive to these subtle changes. Better project result would be only achieved by many rounds of iterative by using old method. By using design preservation based on Place&Route tool,only the algorithm modules were resynthesized, replaced and rerouted. The other modules especially driving modules were fully inherit the previous design. The problem about the algorithm module was affected by the subtle change of driving circuit
would be well solved. In the actual project , the methodology was a better solution for upgrade of algorithm module.

Summary

Two methodologies had been analyzed in detail previously. The following compared from the impact on all aspects of engineering practice and distinguished different in a 5-point way.

1) Degree of difficulty: the description about the degree of consistency of most basic design flow, the more consistent, the higher the score.

2) Applicability: applicability about different types of engineering. The higher applicability, the higher the score.

3) Additional charges: in addition to the basic design tool. The less the need for additional tools, the higher the score.

4) Credibility: the credibility of the design is completely maintained with this method. The easier it is verified, the higher the score.

5) Update speed: the speed of the whole project update after several modules are updated, the faster the speed, the higher the score.

Table 1 Back-end tool Directed Routing VS Design Preservation based on Place&Route Tool

	Degree of difficulty	**Applicability**	**Additional charges**	**Credibility**	**Update speed**
Back-end tool Directed Routing	4	1	5	4	2
Design Preservation based on Place&Route Tool	3	4	4	3	4

The two methodologies were compared in a certain degree of quantization in Table.I. Each had advantages and disadvantages in various aspects. Therefore designers need to choose the appropriate methodology to achieve design inheritance according to the needs of different projects to improve the design and debug efficiency.

References

[1] Yang Haigang,Sun Jabin,Wang Wei,“An Overview to FPGA Device Design Technologies,” Journal of Electronics&Information Technology, vol. 32 No3, pp. 714-723, Mar 2010.

[2] Xilinx. Hierarchical Design Methodology Guide. http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/Hierarchical_Design_Methodology_Guide.pdf,2013,4..

[3] Xilinx. Design Preservation Tutorial

http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/PlanAhead_Tutorial_Design_Preservation.pdf,2013,4.

[4] Xilinx. Hierarchical Design Using Synopsys and Xilinx FPGAs http://www.xilinx.com/support/documentation/white_papers/wp386_Hierarchical_Design_Synopsys_Xilinx.pdf,2013,4.

[5] Synopsys.Synopsys FPGA Synthesis User Guide

http://solvnet.synopsys.com/ Synopsys FPGA Synthesis User Guide.pdf,2009,3

[6] Bao Bin, “ A signal-processing system of digital pixel binning based on bi-cubic filtering algorithm” Electro-Optical Remote Sensing, Photonic Technologies, and Applications VII and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing, 2013, vol8897,pp88970G-1-88970G1-7 ,Sep.2013.

[7] Altera Corporation, Stratix II GX device handbook.

http://www.altera.com/literature/hb/stx2gx/stxiigxhandbook.pdf,2005,10

