A Rapid Design Method of Multi-FPGA ASIC Prototyping Platform

Steven Guo, VeriSilicon Microelectronics (Shanghai) Co., Ltd.

Abstract：

With the growing complexity and scale of FPGA-based SoC Verification System Board design, it is a great challenge to shorten cycle time and ensure first time right for the design. The traditional manual design methodology is almost impossible to meet the aggressive Time-to-Market requirement. A rapid design method of Multi-FPGA ASIC Prototyping Platform is presented to addresses such needs by Cadence Orcad FPGA System Planner (FSP).

Interface components library creation、Protocol definition between FPGAs and interface components and Physical optimum Pin assignment, are discussed in this paper with one Multi-FPGA ASIC Prototyping platform based on 4 Virtex-5 XC5LX330-FF1760 as an example. The new methodology has been proven to be very effective and feasible to accelerate design schedule, eliminate unnecessary physical design iterations and enable ASIC to enter the market first.

Key words ： Multi-FPGA 、FSP、Protocol、ASIC、Prototyping、SoC

Contents

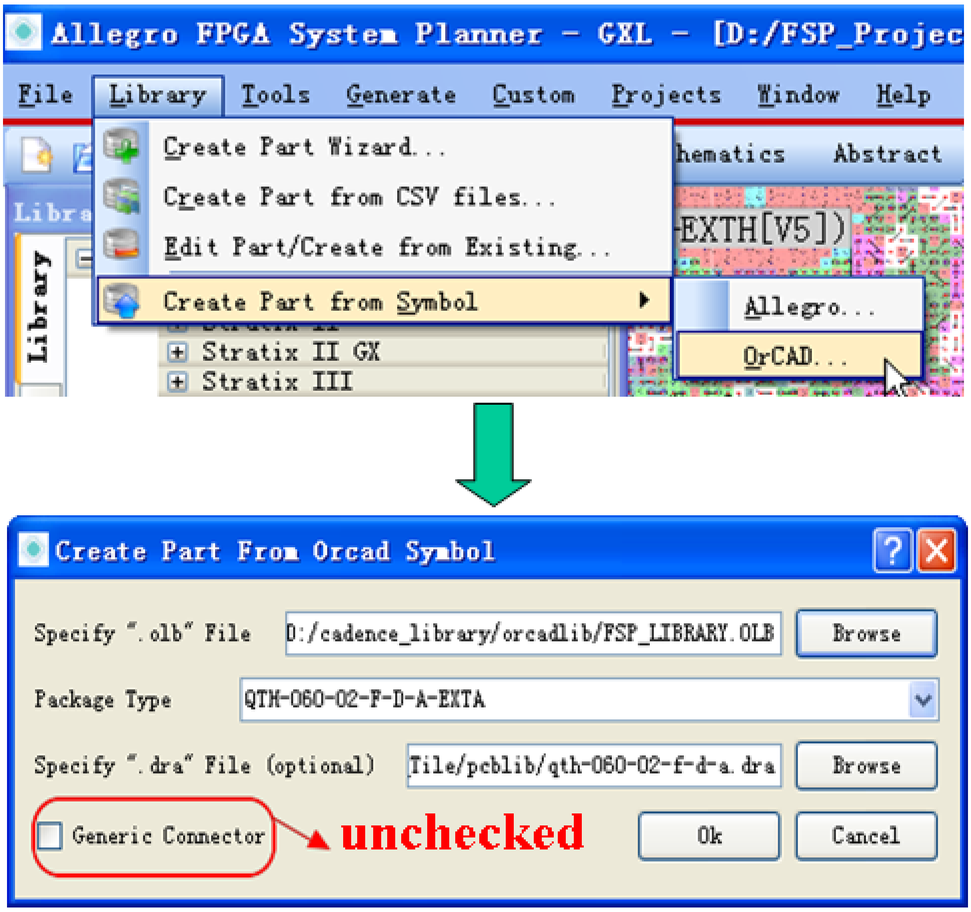
1. Introduction

In order to prototype an ASIC or SoC design using FPGAs within limited cycle time, it is very crucial to customize one scalable and rugged verification platform radiply which enables FPGA & System Engineers to perform Verilog code porting、firmware & embedded operation system development and System validation.

In the past years, System design Engineers have encountered the tedious 、time-consuming and error-prone manual IO Pin assignments between FPGAs and Peripheral devices or Connectors. Fortunately, FPGA System Planner will be able to release engineers from the very laborious process today. And detailed FSP flow & skills will be discussed by one successfully proven platform with 4 FPGAs.

2. Interface components Library Creation

Firstly of all, it is required for us to create interface components library, the typical flow is shown in Figure 2.1.



Specify ".olb" File D:/cadence_library/orcadlib/FSP_LIBRARY.OLB

Package Type QTH-060-02-F-D-A-EXTA

Specify ".dra" File (optional) File/pcblib/qth-060-02-f-d-a.dra

☐ Generic Connector

Figure 2.1 Windows for building interface components symbols

To achieve one validated symbol for interface components, need to follow below 4 steps in current window " Create Part from Orcad Editor".

a) Logical > Edit Model and Add Group

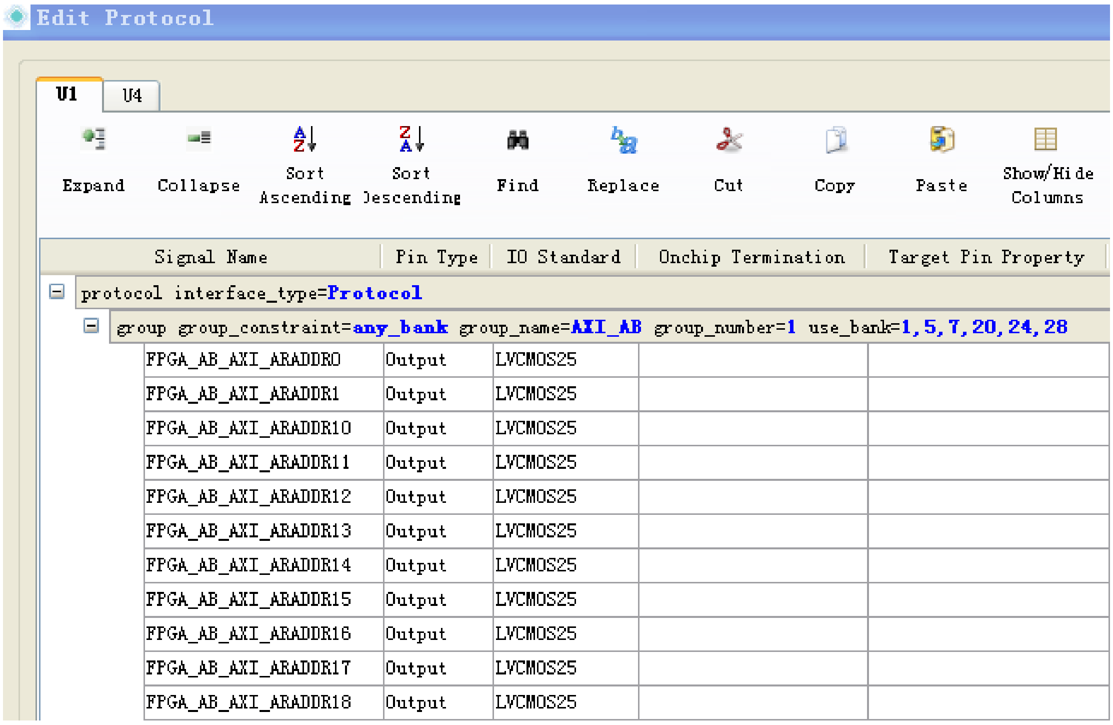
b) Logical > Define Voltage level and IO Standard

c) Layout > Edit pin Properties

d) Check Model and OK

3. Protocol Definition

After the complete placement of FPGA and interface components, we will define protocol between FPGAs or FPGA & interface components. Detailed info are shown as Figure 3.1. Setup includes Signal Name, Pin Type, IO Standard, and specified IO Banks to certain signal group.



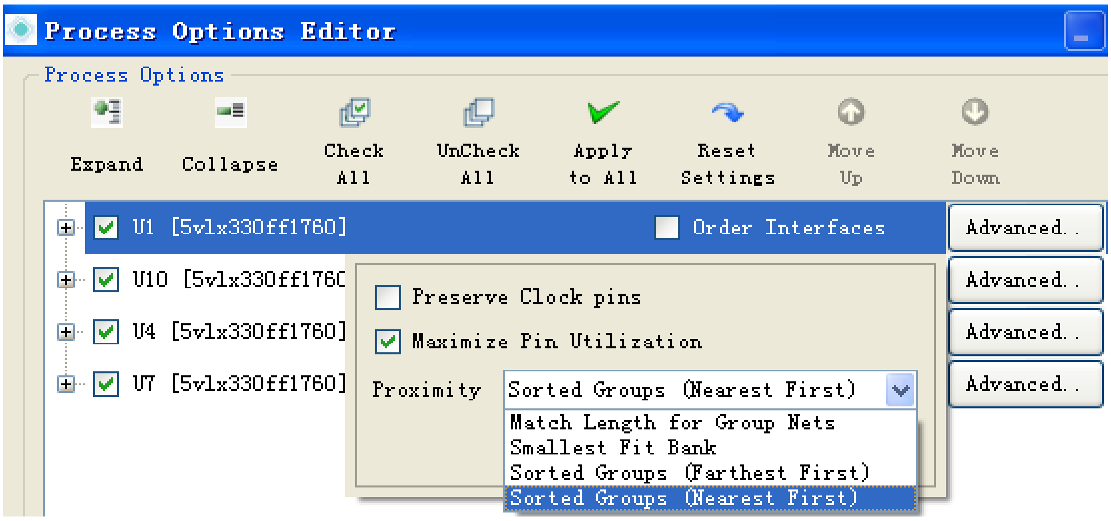
⊟ protocol interface_type=**Protocol**

⊟ group group_constraint=**any_bank** group_name=**AXI_AB** group_number=**1** use_bank=**1,5,7,20,24,28**

Signal Name	Pin Type	IO Standard	Onchip Termination	Target Pin Property
FPGA_AB_AXI_ARADDR0	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR1	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR10	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR11	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR12	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR13	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR14	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR15	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR16	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR17	Output	LVCMOS25		
FPGA_AB_AXI_ARADDR18	Output	LVCMOS25		

Figure 3.1 Protocol Edit window.

4. Setup Proximity rule and Run Design



⊞ ☑ U1 [5vlx330ff1760] □ Order Interfaces

⊞ ☑ U10 [5vlx330ff1760

⊞ ☑ U4 [5vlx330ff1760]

⊞ ☑ U7 [5vlx330ff1760]

□ Preserve Clock pins

☑ Maximize Pin Utilization

Proximity: Sorted Groups (Nearest First)

Figure 4.1 Setup Proximity and Run Design

When finished all protocols definition, then we may setup proximity rule and Run design to perform connections of FPGAs and interface components as Figure 4.2.

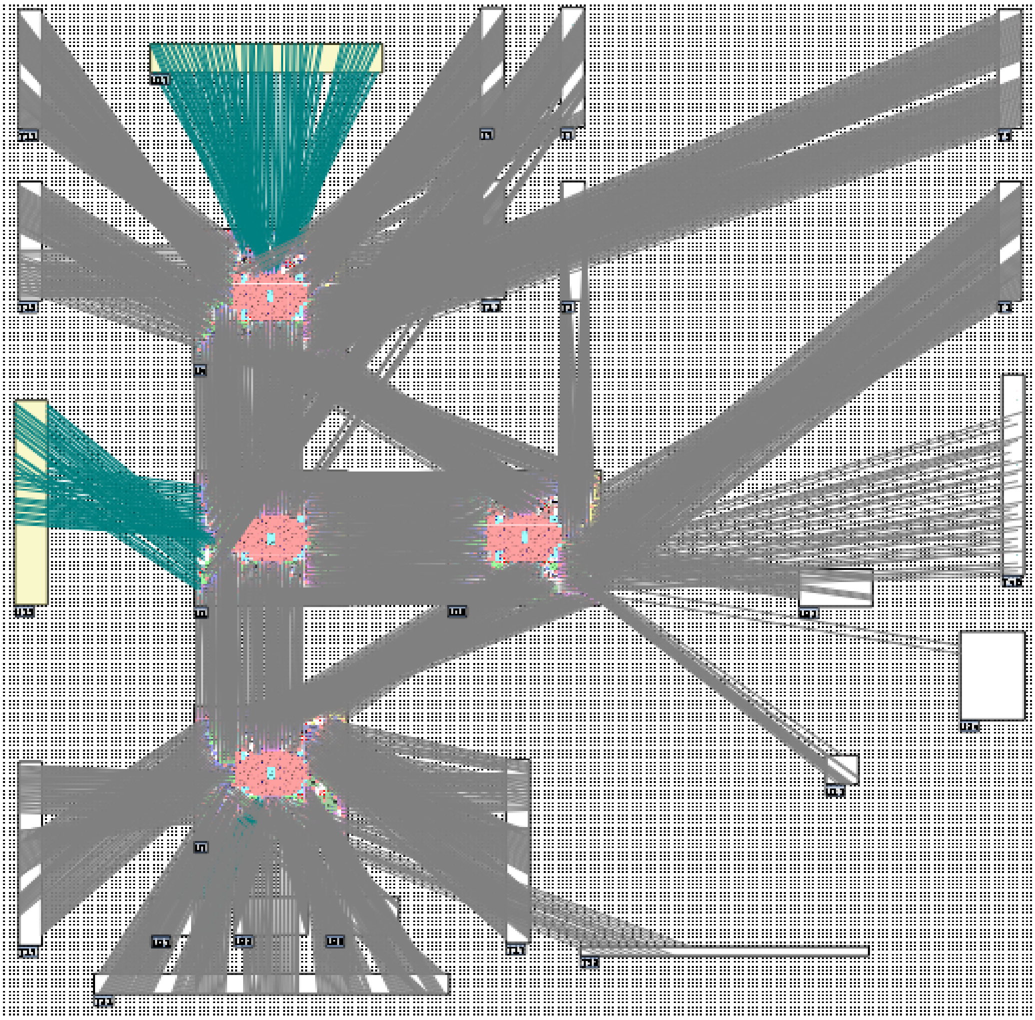
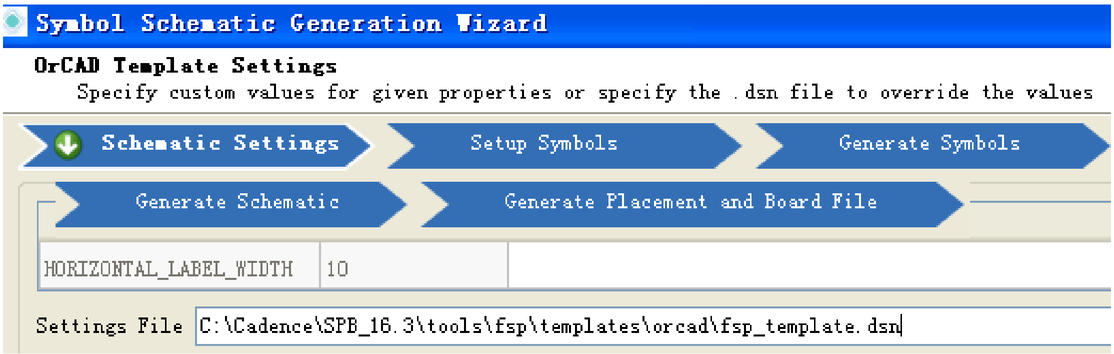


Figure 4.2 Connections of FPGAs and Interface components

5. Generate Symbol and Schematic



Symbol Schematic Generation Wizard

OrCAD Template Settings

Specify custom values for given properties or specify the .dsn file to override the values

HORIZONTAL_LABEL_WIDTH 10

Settings File C:\Cadence\SPB_16.3\tools\fsp\templates\orcad\fsp_template.dsn

Figure 5.1 Symbol Schematic Generation Wizard

And then, we can generate symbol and schematic according to connection results and Symbol schematic Generation Wizard which is showed in Figure 5.1.

Up to now, we have accomplished one complete & optimum Pin assignment and schematic with routable and physical consideration.

Meanwhile, PCB Engineer will work on PCB layout and System Engineer will continue schematic optimization until frozen design ultimately.
As shown in Figure 5.2, this is the picture of first-time-right Multi-FPGA ASIC Prototyping platform. And we have already fulfilled Verification for two ASIC projects successfully using the platforms.

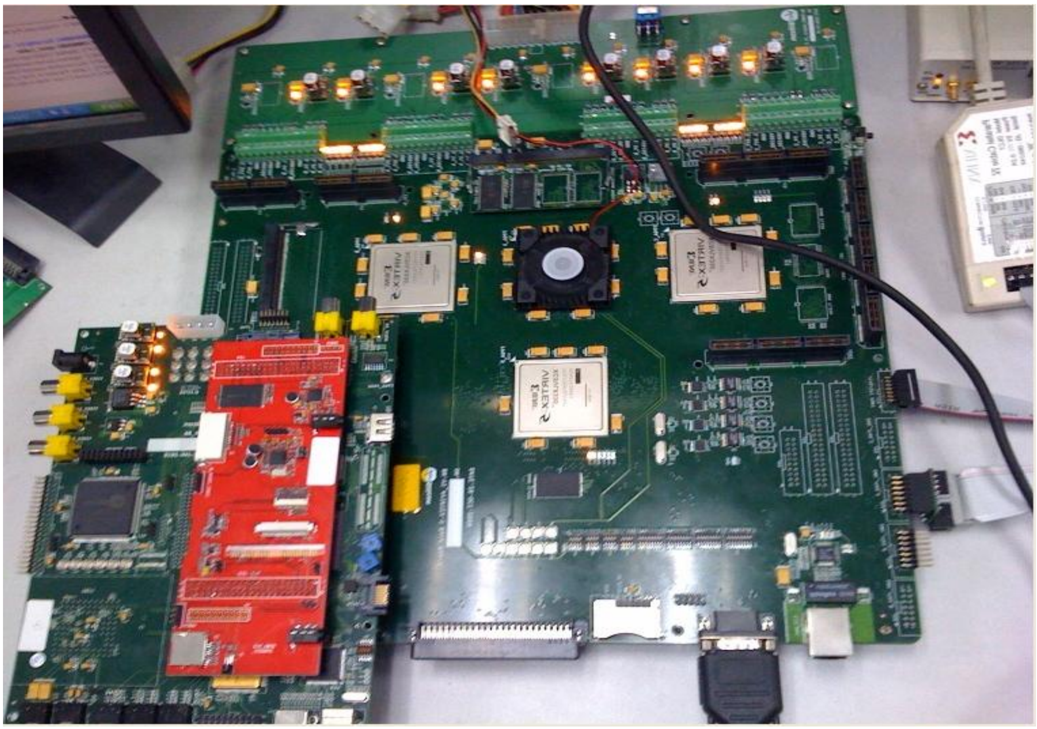


Figure 5.2 First-Time-Right Multi-FPGA ASIC Prototyping platform

Finally, it should be pointed out that only key steps have been mentioned in this paper. In addition to these, other features such as Map power and Pinswap are also interesting and useful to improve & optimize connections and assignment.

6. Conclusion

A flexible, scalable, rugged platform such as the one presented in this paper is as example of a ASIC prototyping platform that can be build for hardware and software co-verification and co-development.
For time-to-market pressures, it is required for semiconductor companies to ensure first-time-right Chip design. This paper indicates that FSP solution enables us to implement quick physical routable pin assignment 、shorten the cycle time of schematic and PCB design of platform and reduce risks to a great extent.

References:

[1]. Co-verification of Hardware and Software for ARM SoC Design - Jason R. Andrews. 2005

[2]. Cadence Orcad FPGA System Planner Datasheet. 2009