|  |
| --- |
| FPGA Cookbook for NetSpeed Gemini  Version: GEMINI-15.10  October 5, 2015 |

FPGA Cookbook for NetSpeed Gemini

About This Document

This document is a cookbook for using NetSpeed Gemini on an FPGA.

Audience

This document is intended for users of NocStudio:

* NoC Designers
* SoC Designers
* FPGA Designers

Prerequisite

Before proceeding, you should generally understand:

* Basics of FPGA Technology

Related Documents

The following documents can be used as a reference to this document.

* NetSpeed NocStudio Gemini User Manual
* NetSpeed Gemini Physical Design Guidelines
* NetSpeed Gemini IP Integration Spec

Customer Support

For technical support about this product, please contact [support@netspeedsystems.com](mailto:support@netspeedsystems.com)

For general information about NetSpeed products refer to: [www.netspeedsystems.com](http://www.netspeedsystems.com)

**Contents**

[About This Document 2](#_Toc427399492)

[Audience 2](#_Toc427399493)

[Prerequisite 2](#_Toc427399494)

[Related Documents 2](#_Toc427399495)

[Customer Support 2](#_Toc427399496)

[1 Introduction 4](#_Toc427399497)

[2 Techniques To Implement Design in FPGA 5](#_Toc427399498)

[2.1.1 Partitioning 5](#_Toc427399499)

[2.1.2 Data-width (area) and frequency trade-off 5](#_Toc427399500)

[2.1.3 Module boundaries 5](#_Toc427399501)

[2.1.4 Internal bus structure 6](#_Toc427399502)

[2.1.5 Clocks 6](#_Toc427399503)

[2.1.6 Resets 6](#_Toc427399504)

[2.1.7 Big multiplexer structures 6](#_Toc427399505)

[2.1.8 High-level Floorplan 6](#_Toc427399506)

# Introduction

NetSpeed NoC IP products support implementation of application-tailored NoCs for FPGAs. This document outlines the methodology and techniques of FPGA based system designs and how NoCs can be implemented to meet the requirements.

Some benefits of FPGA based system design are ability to reconfigure, ease of debug by internal logic access, product upgrade in-field, no DFT challenges/risk/NRE costs associated with of ASIC development. NetSpeed NoC design meets the following key objectives of FPGA based system design:

* Higher performance to meet demanding system performance goals
* Higher and efficient area utilization to use smallest possible FPGA
* Short development and turn-around cycle with consistent results
* Ease of RTL code migration from one FPGA vendor/family to another

# Techniques To Implement Design in FPGA

FPGA based system designs have various interdependent requirements such as maximizing device utilization, optimizing IO placement, minimizing routing congestion and reaching timing goals. In the following subsections, methodology and techniques to be adopted in various phases of the development cycle are explained.

During the architecture phase FPGA device vendor and family must be chosen. Furthermore certain design aspects should be considered such as data-width and frequency tradeoff, flexible communication protocols, resource sharing, splitting system into multiple sub-systems based on concurrent functional requirements. While finalizing the architecture, both FPGA area and speed requirements need to be looked into, as they are tightly coupled.

Following are some important techniques for the above stated aspects:

### Partitioning

Architectural partitioning shall create manageable blocks which favor incremental refinements and easier timing closure. NetSpeed NoC by construction meets this criterion. NoC is partitioned into NoC elements for easy timing closure.

### Data-width (area) and frequency trade-off

Same throughput can be achieved by having xbits width running at y MHz or 2x-bits width running at y/2 MHz. The 2x-bit width might look better from frequency requirement, but based on other aspects it may not be. For instance, if this data is stored in on-chip memory, the number of memories required becomes double with 2x-bit width. As such number of memories required may not be a big concern, but definitely fixed locations of onchip memories, huge data multiplexer area overhead and routing congestion needs to be considered. While defining the NoC this consideration shall be kept in mind.

### Module boundaries

All modules interfaces shall be on register boundary. While defining the NoC the following prop\_defaults shall be used to meet this criterion.

* prop\_default credit\_output\_register yes
* prop\_default ifce\_credit\_output\_register yes
* prop\_default ifce\_output\_register yes
* prop\_default local\_credit\_output\_register yes
* prop\_default local\_output\_register yes
* prop\_default output\_register yes

### Internal bus structure

A well defined internal point-to-point bus structure is preferred than routing all signals back and forth. NetSpeed NoC by construction meets this criterion. All the NoC elements have a well defined point-to-point bus structure to interface to each other.

### Clocks

Clock multiplexing and gating shall be avoided and if required shall be done based on device capabilities. While defining the NoC appropriate switch shall be used to meet this criterion.

### Resets

Number of resets in the system shall be optimized based on dedicated reset routing resources available. NetSpeed NoC by construction meets this criterion.

### Big multiplexer structures

It is not preferred to build very big multiplexer structures (say 256:1) especially for timing critical paths. Instead smaller multiplexers can be built, which are more controllable. While defining the data widths of the NoC, this criterion should be considered. Furthermore the channel widths should be kept uniform to avoid up-/down-sizing between channels.

### High-level Floorplan

NocStudio allows users to specify high level floorplan information and generate floorplan aware NoC designs. A high-level floorplan including IO planning shall be worked-out based on the gate count and other macro estimates. Also spare area shall be planned for future/field upgrades. At this stage it is not necessary to fix the IO locations but it is necessary to fix the IO banks in FPGA. Having done the high level floorplan; the budgeted area shall be known to module level designers. Also interface module floorplan locations shall be known to the FPGA implementer, which will enable them to further floorplan allocated area if necessary. Some of the high level floorplanning considerations are:

* Controlling congestion along with proximity
* Draw the data flow diagram of the design with the memories that are used to terminate the data paths and do module level area allocation
* Interdependent modules should be closer
* Module level area allocated shall be close to Macros which it is interfacing to
* Free area (rows and columns) between module area allocations, which will aid in inter module routing in full chip
* Clock resources and routing limitations if any

2670 Seely Ave

Building 11

San Jose, CA 95134

(408) 914-6962

<http://www.netspeedsystems.com>