Comparative study of ThinLink-based routing optimization

AMANDA DARELL & TONGYING SHI
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Comparative study of ThinLink-based routing optimization

Amanda Darell

Tongying Shi

Department of Electrical and Information Technology Lund University

Supervisor at Ericsson: Moazzam Fareed Niazi, Faruk Sande

Supervisor at LTH: Per Andersson

2024-05-13

Abstract

In recent decades, advancements in silicon technology have led to a substantial increase in chip complexity, primarily driven by the integration of more transistors within a given unit of chip area. Consequently, routing connections between these transistors has emerged as a significant challenge for the physical design of the chips. AMBA's AXI-4 based connectivity is a commonly employed scheme to connect different segments of a chip. However, the number of wires/connections that travel between any two chip segments is quite significant and elevate routing congestion. This research dives into the ThinLink-based AXI Interface optimization, specifically harnessing the capabilities of Arm's TLX-400 Network Interconnect. The core of this study will involve implementing and testing these optimizations on Ericsson's existing radio SoC connectivity, which serves as the design under test (DUT) for managing routing congestion. By examining the effects of this optimization on performance and layout efficiency, the research aims to offer a tangible and actionable solution to counteract routing congestion challenges in AXI connectivity.

The primary goal of this thesis is to present a solution for optimizing routing congestion within a System-on-Chip (SoC) to enhance area efficiency. The area factor is pivotal, if the available routing space prove insufficient, adjust the routing channel size becomes necessary, leading to a corresponding changes in die size and utilization. By addressing congestion, we ensure optimal use of the provided die and channel dimensions, which is critical to achieving the desired area efficiency. In this study, various channel widths have been evaluated and the resulting performance metrics across different configurations have been compared. This comparison will assist developers in selecting the most appropriate channel width that aligns with specific design requirements.

Popular Science Summary

In the complex world of semiconductor design, engineers continually face the challenge of routing congestion—a pivotal issue that affects the performance, power efficiency, and overall effectiveness of chips. This phenomenon is akin to traffic congestion in a bustling city, where too many routes on a limited roadway lead to slowdowns and disruptions. As chips grow denser with transistors and incorporate more functionalities, efficiently managing these electronic pathways becomes critical.

A collection of recent research and discussions from various sources highlights innovative approaches to tackling this enduring challenge. For instance, a study from NIPS introduces joint learning methods, suggesting that a combination of techniques could enhance the optimization processes of chip placement and routing, ensuring smoother and more efficient chip design processes. Another article from ScienceDirect explores a congestion-aware router designed specifically for Network-on-Chip (NoC) architectures, which prioritizes low-latency and high-throughput communication to streamline data flow across the chip.

The integration of artificial intelligence in chip design represents another step forward. Research from Google demonstrates how deep reinforcement learning can be leveraged to optimize chip placement automatically. This AI-driven approach not only speeds up the design process but also learns from previous designs to continually enhance its placement strategies, potentially outperforming human-expert-designed chips.

Meanwhile, practical algorithms are also making significant strides. An article from Semiconductor Packaging News discusses an innovative algorithm that boosts cell utilization rates while mitigating routing congestion. This method ensures that every square millimeter of the chip is used effectively, reducing wasted space and enhancing overall chip functionality.

Other studies delve into specific architectures and design methodologies that address routing congestion directly. For example, research from NTU provides

insights into multilevel routing strategies for the X-architecture, focusing on reducing wire lengths and the number of vias, which can simplify the routing process and aid in faster timing closure.

The previous summary involves a mix of early-stage predictive and planning tools (like AI and machine learning) and mid-to-late-stage optimization techniques (like congestion-aware routers and specific routing strategies). In this thesis, we focus on a specialized approach using the ThinLink-based AXI Interface optimization for managing routing congestion in System-on-Chip architectures. This study is deeply rooted at the RTL design stage of the chip design. It details methodologies applied during the early phase.

Contents

Abbreviations

AMBA	Advanced microcontroller bus architecture
ASIC	Application-specific integrated circuit
AXI	Advanced eXtensible interface
CDC	Clock Domain Crossing
CPU	Central processing unit
DLL	Data link layer
DUT	Design under test
GUI	Graphical user interface
HCG	Hierarchical clock gating
HDL	Hardware description language
IC	Integrated circuit
IL	Interface layer
IP	Intellectual property
IPA	Interconnect Workbench Performance Analyzer
IPC	Interconnect performance characterization
IWB	Interconnect Workbench
NoC	Network-on-Chip
PL	Physical layer
PPA	Power, performance, area
RTL	Register transfer level
SoC	System-on-Chip
STA	Static Timing Analysis
TB	Test bench
UVM	Universal Verification Methodology

1 Introduction

System-on-Chip (SoC) architectures have become the cornerstone of modern electronic systems, consolidating multiple components into a single integrated circuit. As these systems have evolved to meet the growing demand for enhanced functionalities and higher performance, the complexity within the SoC has proportionally increased. This surge in complexity has brought to the forefront a significant challenge: routing congestion.

Routing, the process of creating pathways for signals to traverse between different components of the chip, is pivotal for the efficient functioning of SoCs. Routing is a critical aspect of chip design as it determines how data, control signals and other important information move within the chip to enable proper functionality. However, as SoCs integrate more components and functionalities, the demand for these routing pathways escalates, often leading to a scenario where multiple signals compete for limited routing resources within specific regions of the SoC. This phenomenon, similar to traffic jam in urban settings, is termed as routing congestion.

The implications of such congestions are multifaceted. Not only can they degrade the overall performance of the SoC due to signal delays and longer signal paths, but they can also increase power consumption and necessitate multiple design iterations, leading to higher design costs and extended time-to-market. In extreme cases, routing congestion could result in the failure of certain parts of the chip to function properly, which in turn can result in non-functional chips. Addressing these congestions, therefore, is not just beneficial but imperative for the efficient and cost-effective design of SoC architectures.

Set against this backdrop, this research aims to explore the potential of the ThinLink-based AXI Interface, particularly Arm's TLX-400 Network Interconnect, as a promising solution to mitigate the challenges posed by routing congestions. Using Ericsson's existing radio SoC connectivity as the benchmark or DUT, this study seeks to delve deep into the intricacies of routing conges-

tions, such as their impact, and the innovative solutions that can be employed to address them.

Minimizing routing congestion in SoC design is crucial for ensuring efficient and reliable data transfer in integrated circuits. Routing congestion can occur when there is an excessive demand for routing resources, such as wires or channels, in the chip. This can result in delays, increased power consumption and potential design failures. Therefore, it is of major importance to minimize congestion as this directly relates to the functionality of the SoC as well as the competitiveness in the market.

In this thesis we have used tools such as Arm Socrates to assign values to parameters, Cadence Interconnect Workbench (IWB) for verification of the code, such as performance analysis including latency and bandwidth. Another tool that has been used is the Cadence SimVision which has been used to verify the correctness of the code by showing waveforms for all the signals that have been assigned to values.

1.1 Goals for the master's thesis

In-depth exploration of TLX-400 Network Interconnect

Undertake a comprehensive examination of the TLX-400 architecture, delving into its structural and functional intricacies.

Diagnostic Assessment of Ericsson's radio SoC

Conduct a meticulous study of Ericsson's existing radio SoC connectivity, specifically focusing on its role as the DUT in addressing routing congestion challenges.

Logical Design Partitioning

Formulate strategies to derive logical design partitions, ensuring efficient distribution across multiple physical segments of the SoC.

Partitioning is the process of dividing the chip into smaller blocks during the design phase. The reason for dividing the chip is to simplify routing and verification of the chip (Brooks 2019).

Register slices in integrated TLX design

The usage of register slices in TLX design is to compensate for the routing delay that can occur between two subchip components, ensure optimal synchronization and satisfy timing budget.

Performance Analysis proficiency

Familiarize with the nuances of performance analyzers, aiming to gauge the performance implications of various design architectures effectively.

Synthesis proficiency

Aim to gain expertise in the synthesis flow to comprehensively evaluate the area impact across different design architectures. This ensures optimal space utilization, efficiency, and accurate validation of inserted timing slices relative to the given floorplan and routing length.

1.2 Approach

- Based on ThinLink-based AXI interface optimization using Arm's TLX-400 Network Interconnect
- TLX IP partition as well as integration in connect subchips
- Performance analysis in terms of latency and bandwidth
- RTL design, SystemVerilog code
- Synthesis flow tooling for design space exploration and comparison with existing implementation, static timing analysis

1.3 Detailed work plan

The thesis was partitioned into five milestones, where the duration for each milestone was planned to be around one month.

- Work plan 1: literature review, research and familiarization with the TLX-400 architecture. Analysis of Ericsson's existing radio SoC connectivity, such as AMBA buses, SoC connect structure and performance analysis.
- Work plan 2: dive deep into the AXI interface to understand its operation and specifications. Investigate how this protocol works with the TLX-400 architecture.
- Work plan 3: development of the optimized architecture using ThinLink, accomplished by RTL design, as this whole project is within ASIC top level integration.

- Work plan 4: comparative study and evaluation using performance analysis and synthesis flow.
- Work plan 5: synthesis of findings and proposal of the optimized solution. Report and presentation.

1.4 Structure of the report

Chapter 1: Introduction

Introduces the topic.

Chapter 2: Background

All terms such as network interconnect, AMBA bus, register slice and routing will be described.

Chapter 3: Tools

This chapter describes the tools used in this thesis. We have used Git, Socrates, Cadence Interconnect Workbench, Cadence Interconnect Performance Analyzer, and Cadence SimVision.

Chapter 4: AXI protocols

Description of the protocols AXI4 and AXI-Stream developed by Arm, and the design requirement is also covered.

Chapter 5: Working progress

Description of the RTL code, synthesis and learning objective.

Chapter 6: Results

Results of our work, discussion and further improvements.

1.5 Related work

Reducing routing congestion on a chip can be accomplished in several ways. This report clarifies that the effective method to mitigate congestion is through the reduction of the number of physical wires between two IPs. Additionally, the report explores the implementation of a new algorithm where routes are selected to efficiently pass packets through congested areas to reach their final destination.

2 Background

The following section will describe the topics of ASIC design, interconnect, AMBA bus, routing and register slices.

2.1 ASIC design

ASIC design is a methodology that aims to reduce the cost and area of an electronic circuit, product or system into one single circuit, the ASIC. Electronic products usually consist of many integrated circuits (ICs) which have been connected to each other by using an interconnect with a purpose of fulfilling a specific function (Brooks 2019). ASIC design has three main optimization goals - performance, power and area (PPA). Since these parameters will be affected by each other, it is important that there is a balance between them. The main focus in this thesis will be within the performance and area aspect, but power will also be considered.

2.2 AXI Network Interconnect

An interconnect facilitates the exchange and processing of data. The purpose of the AXI Interconnect is to connect one or more AXI memory-mapped manager devices to one or more memory-mapped subordinate devices, as can be seen in figure 1 below. The structure of an interconnect can be seen in the picture below (Arm 2022).

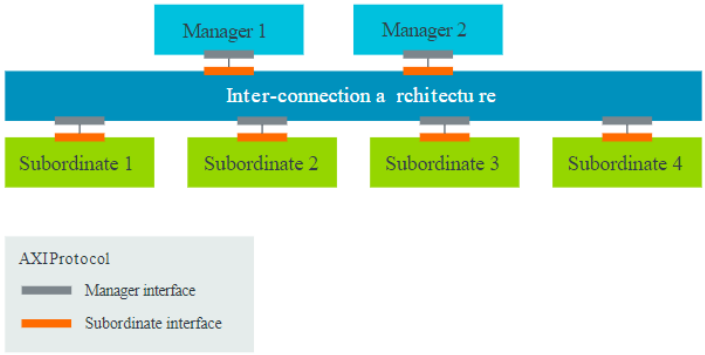

Figure 1: the structure of an interconnect

2.2.1 NIC-400 Network Interconnect

The NIC-400 Network Interconnect is highly configurable and enables the user to create a high performance, optimized and AMBA-compliant network infrastructure. There are multiple possible configurations for this interconnect, ranging from a single bridge component such as a protocol conversion bridge between AXI protocols, to an interconnect consisting of up to 128 managers and 64 subordinates of AMBA protocols (Arm 2016*a*).

2.2.2 TLX-400 Network Interconnect

The Corelink TLX-400 Network Interconnect ThinLink is an extension of the NIC-400 Network Interconnect. TLX-400 provides a mechanism to reduce the number of signals in an AXI point-to-point connection (forward and reverse links) and enables it to be routed over a longer distance. Each one of these forward and reverse links can be independently configured to reduce the number of wires that the connection requires. By reducing the number of signals on a SoC, the probability that routing congestion occurs decreases significantly and this is also one of the key features of TLX-400, that it reduces routing congestion. Since the thesis topic is ThinLink based routing optimization, it is suitable to use TLX-400 Network Interconnect. TLX-400 is implemented as forward and

reverse links where each link is partitioned into three layers; the physical layer (PL), the data link layer (DLL) and the interface layer (IL) (Arm 2016*b*).

2.2.3 Routing

Routing is the process of establishing physical connections based on logical connectivity. In other terms, routing means connecting components such as cells, macros and power with metal traces. Routing interconnects networks and the Internet. There are several advantages of using routing, such as minimizing the total wire length, minimizing the path delay and meeting the timing constraints (Shirshendu 2020).

2.2.4 Routing Congestion

The main goal of the thesis is to reduce routing congestion on a chip, so a brief description of the term will be given here. Congestion on a chip refers to the issue when the number of routing tracks is less than the number of needed routing tracks (Shukla 2022). The reason why this issue continues to increase is that more IP blocks and cores are added on chips, which results in an increase of the number of routes between cores. Routing congestion is considered when handling the constraints of timing, power and area. The main focus in the thesis is to reduce the physical area by reducing the number of wires between the subchips. There are several reasons why it is of major importance to handle routing congestion, as high congestion results in an increased size of the chip or more metal layers (*Routing Congestion: The Growing Cost of Wires in Systems-on-Chip* n.d.). This aspect means that routing congestion is costly. Keeping the cost of production of chips as low as possible is important, which means that this issue must be addressed and handled.

2.3 TLX-400 ThinLink

A subchip is a functional block that encapsulates specific circuitry designed to perform certain tasks or functions within an integrated circuit. Each subchip is

dedicated to different functions such as processing or memory storage.

The TLX-400 ThinLink that has been implemented in this master thesis project, is a link with fewer physical wires between subchips. Using fewer wires on this link can reduce the area and thereby receive a more efficient solution. There are several subchips, but we have primarily focused on a link between two subchips, since this implementation then can be used for other links as well.

The TLX-400 Network Interconnect mentioned previously is then placed on these subchips, and not on the physical link between them.

2.3.1 Forward and reverse link

TLX-400 is implemented as a link that operates in both forward and reverse direction, as can be seen in figure 2 below. Each link is also partitioned into three functional layers - interface layer (IL), data link layer (DLL) and physical layer (PL).

The interface layer displays the data to the users as an interface.

The data link layer is responsible for buffering of transfer packets at the end of each link as well as ensuring error-free transmission of information.

The physical layer is the layer on the chip that is closest to the physical connection between devices.

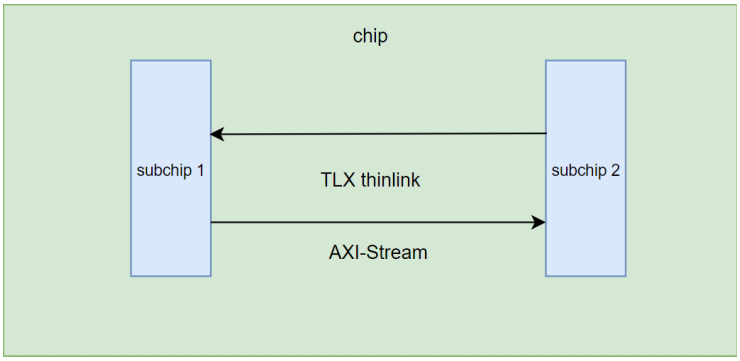

Figure 2: the structure of the subchips

2.4 AXI Register Slice

A register slice in digital design is a segment of a register, i.e., a memory used to store data. The AXI Register Slice specifically, can be used between AXI endpoints such as the end of a manager and subordinate respectively or between switches such as interconnects. This means that the register slice provides the infrastructure to insert a pipeline stage between an AXI4-Stream manager and subordinate (Instruments 2023).The benificial part is to break critical timing paths and achieve higher clock frequencies. For each register slice instance, it is possible to enable pipelining on any of the five AXI channels, write address, write data, write response, read address and read data.

Some of the key features for the AXI Register Slice are the following - it is fully configurable to match the AXI port widths, it is ideal for breaking up a critical timing path and to set each of the five channels independently in order to achieve optimal latency and performance characteristics (Xilinx 2022).

The reason for using register slices in this thesis, is that the clock speed can be increased at the cost of area by an appropriately placed register slice between the manager and the subordinate in the TLX-400 architecture.

2.5 Advanced microcontroller bus architecture

AMBA is an interconnect specification developed by Arm, used for the connection of functional blocks in SoC designs. The purpose of using AMBA is to simplify development of multi-processor designs with a large number of controllers and components in a bus architecture (Arm 1999). It is commonly used in ASIC development.

2.6 Hierarchical clock gating

Hierarchical clock gating enables a system to transition to another power state, which can be a low-power state where the power consumed by the clock tree can be saved.

2.7 Design requirements

- Incorporate both AXI4 and AXI-Stream protocols in the proposed solution
- The solution will be based on Arm's TLX-400 network Interconnect, which is an extension of NIC-400 Network Interconnect
- Minimize routing congestion on SoC by reducing the number of physical wires and thereby reduce the physical area

3 Tools

In order to compile and verify the RTL code written in SystemVerilog, the following tools have been used.

3.1 Git

Git is a version control system that tracks updates in the code, particularly suitable to use when collaborating with other developers. Throughout the thesis, we have used Git on a daily basis to push and pull our code updates. By using Git the right way, we have avoided most merge conflicts that occurred initially in the project when we were not familiar with the commands and it also enabled us to keep track of our progress and each other's work.

3.2 Cadence Interconnect Workbench (IWB)

The Interconnect Workbench can process design meta data from a structured CSV (comma-separated value) file. It is a tool developed by Cadence, with a purpose of simplifying the verification process by generating a test bench. IWB has three functions, the first one is to generate a UVM environment, which create a universal way of verifying designs. This environment is meant for building functional testbenches for the design. The second one is for functional coverage and the third one is performance analysis such as latency and bandwidth measurements. The main advantage of IWB is that it is more time efficient since the test bench is already generated, which results in a quicker verification process (Cadence 2016*b*). In the CSV file, it is possible to assign values to parameters for both the manager and subordinate interfaces such as address bit widths and clock and reset domains. It is also possible to assign the RTL code to the DUT.

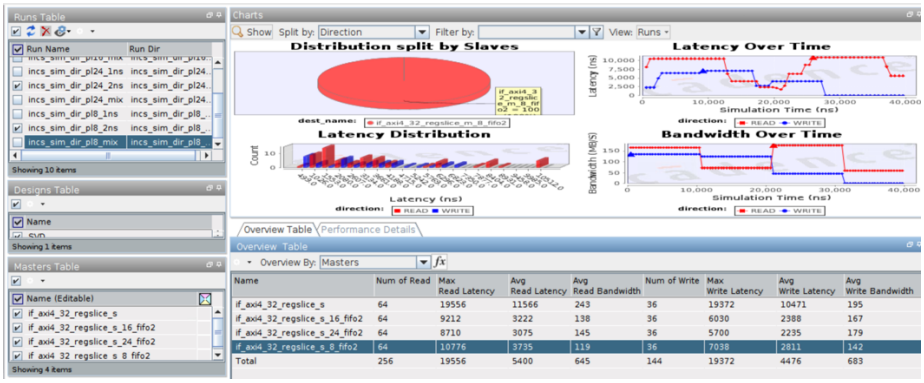

Figure 3: Traffic and performance chart with period 2 ns

3.2.1 Interconnect Workbench Performance Analyzer (IPA)

A tool inside the IWB, is the performance analyzer which is the GUI for the test bench. In the IPA, metrics and graphics for latency and bandwidth for both the manager and subordinate under different circumstances are shown. These metrics and graphics seek to facilitate the understanding of how the tests perform against a plan. We have used IPA to get an overview of how the latency and bandwidth differ in the TLX-400 Network Interconnect based on a different number of register slices on both the manager and subordinate domains on the link.

Figure 3 offers a visualization of the performance obtained from simulation tests under a unified 2 ns clock period. This figure serves as an snapshot into the functionality of a performance analyzer GUI used in optimization studies.

Upper left corner lists various run names, capturing all test cases collected within specific folders. It provides a quick reference to the scope and range of simulation runs being analyzed. In the bottom left corner, the master filter tab represents all the transaction sent from interconnect to system slaves. The upper right quadrant of the GUI shows the *Latency Distribution* and *Bandwidth Over Time* graphs, assessing the temporal performance of the system, offering insights into latency trends and bandwidth fluctuations throughout the simulation period.

The transaction details per master port are summarized in the bottom right corner, providing a granular view of the data interactions.

3.3 Cadence SimVision

SimVision is another tool developed by Cadence, used in a graphical debugging environment. SimVision can be used to debug both analog and digital design written in several hardware description languages (HDL), such as Verilog and SystemVerilog (Cadence 2016*a*), which are the two HDL used in this thesis. During the initial phase of the thesis, SimVision was used to ensure that compilation was working correctly.

3.4 Arm Socrates

Socrates is a tool developed by Arm, and it is used to generate a design specification. Socrates has the capability to generate RTL for any level of hierarchical design such as IP or a subsystem as well as package IP from RTL (Wallace 2017). In this tool, the user can configure parameters such as interface address width for both the manager and subordinate, data width for both the manager and subordinate as well as enabling clocks and clock gating. We used Socrates to set parameters such as the number of register slices and parameter width initially for the transfers on the ThinLink, before assigning values to these parameters in RTL code instead.

4 AXI protocols

Two AXI protocols are used in this thesis, AXI-Stream and AXI4. They both apply the handshake process, an important concept for AXI protocols which will be described further down in the report.

4.1 AXI-Stream

The AXI-Stream protocol is a point-to-point protocol, connecting a single transmitter and a single receiver. This means that this protocol only has one channel, going in two directions. It consists of one control signal and one flow signal (Arm 2021).

4.2 AXI4

The AXI4 protocol on the other hand, has five channels. Three of these are write channels and the other two are read channels. The five channels can be seen in figure 4 below and they are called write address (AW), write data (W), write response (B), read address (AR) and read data (R). AW, W and AR are channels going from manager to subordinate while B and R are channels going the opposite direction. The AXI4 protocol was developed for high bandwidth and low latency applications. The AXI4 protocol is designed to allow communication between manager and subordinate devices.

Each of these five channels consists of the VALID/READY handshake signals as well as a third signal, called the payload signal which is the one containing data. Payload should remain stable during an ongoing transaction. The direction of channels matches the direction of the VALID signal, which has the opposite direction of the READY signal.

The channel transaction begins when the VALID signal is being asserted, that is, the signal is 1. The receiver in the system can then assert the READY signal to inform the channel manager that the channel transaction was accepted. Once this transaction was accepted, the next transaction can begin. A complete

Figure 4: write and read channels

channel transaction is accomplished when both these signals are high. VALID cannot be deasserted until READY is asserted, which means that the source cannot send two valid signals in a row, the first one must be accepted by the receiver first (Arm 2023).

4.3 Handshake process

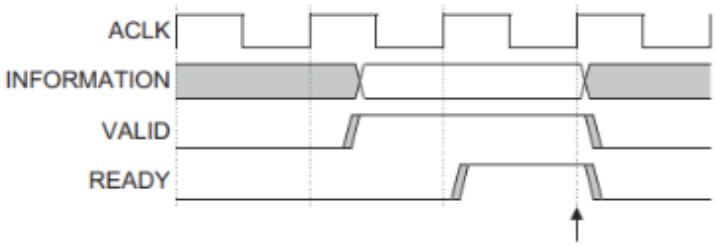

Figure 5: valid before ready handshake

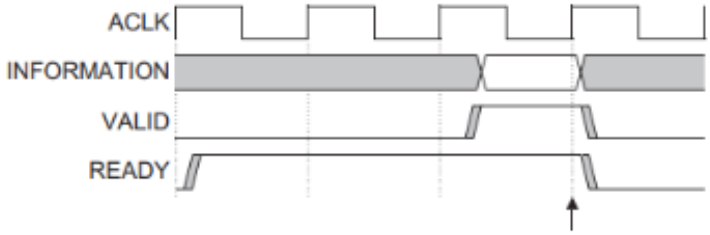

Figure 6: ready before valid handshake

The handshake process describes the relationship between the VALID and READY

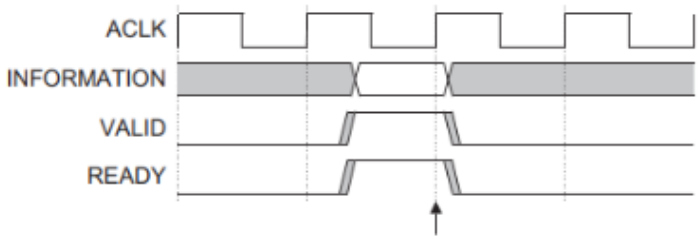

Figure 7: valid with ready handshake

signals. In figure 5, 6 and 7, the three handshake alternatives are shown. The first alternative is when the VALID signal turns high first, the second alternative is when the READY signal turns high first and the third alternative is when both these signals turn high at the same time. The purpose of the handshake process is to provide flow control with only these two control signals. The handshake determines when information is passed across the interface, where the manager sends the VALID signal and the subordinate replies with the READY signal. It is a two-way flow control mechanism which enables both the transmitter and the receiver to control the data rate in the interface. A data transfer occurs when both these are asserted during the same clock cycle, i.e., when both the VALID and READY signals are high. This state is called a complete handshake. They can either be asserted in the same clock cycle or in different ones. VALID and PAYLOAD always operate in the same direction while READY operates in the opposite direction of these two (Arm 2023).

5 Methodology

5.1 Partitioning of the subsystem

The TLX-400 Network Interconnect was initially divided into two different subsystems. The system was divided into one subsystem for the subordinate domain containing the forward physical layer and the reverse data link layer and another subsystem for the manager domain with the forward data link layer and the reverse physical layer. Manager and subordinate domains can be physically distributed across the chip, which means that design partitioning can help the process of transferring data between the different layers. The partitioning of the TLX-400 Network Interconnect could be accomplished in different ways, but we decided to make the partition here for simplicity.

One reason for dividing the manager and subordinate domains, is that partitioning has advantages such as having easier verification and design partitioning, which will result in a more efficient and reliable system design. Another purpose of partitioning the subordinate and manager was to have an AXI stream in the middle of the design that can be set as a parameter of our expectation. For example, the payload width can be set to be a user defined width of 24 or 32 bits, which in the NIC-400 could be over 100 bits.

Afterwards, we placed these two parts separately in two subsystems, reducing the routing congestion between them. Once the compilation was working successfully for both the subsystems, they were merged back together to ensure that these subsystems can operate together since the subsystem of the TLX-400 must include both these domains to work completely. Under the structure of the TLX-400 Network Interconnect, the subordinate is on the left side and the manager on the right side. This means that from a digital design perspective, the manager side of the central processing unit (CPU) will be connected to the subordinate side of the TLX-400 and the subordinate side of the memory will be connected to the manager side of the TLX-400.

5.2 Socrates configurations

The configuration settings were established using the Socrates tool. In the specific System-on-Chip (SoC) design provided by Ericsson, this study focuses on replacing the original NIC-400 with the TLX-400 CoreLink Interconnect. It is crucial to maintain the interface bit width of the manager and subordinate interfaces consistent across the various interconnects. The primary variable altered during the experiment was the internal transfer physical wire count, specifically the AXI stream channel width in the TLX-400 interconnect.

In the process of connecting different IPs via interconnect, the interface bit width for both the manager and the subordinate is fixed at 32-bits, as specified by the provided SoC design. This parameter cannot be arbitrarily modified.

For other project-specific replacements, the interface bit width may be adjusted accordingly. The methodologies and steps applied in this research are adaptable and can be utilized in new projects as well.

Socrates generated RTL files for the logic in the forward and reverse channels such as FIFO blocks for the write and read pointers, FIFO storage and physical layer logic. After configuration, we implemented a top-level RTL file which drives the compilation. Initially, when we divided the TLX-400 Network Interconnect into two domains, we had two top-level files but then we merged them back together to compile the system as a whole, we wrote a new top-level file containing the global clocks and resets as well as an instantiation of the manager and subordinate AXI interfaces. The goal is that the user must not be using Socrates, but rather just changing parameters by changing values in the RTL code in the top-level file to make this application more user-friendly.

5.3 RTL files

As noted earlier, a number of RTL files were produced by Socrates. In addition to these files, we created a top-level file to instantiate the ports of the manager and subordinate interfaces, along with an RTL file for the instantiation of the

files that were generated.

A test bench folder was also established, containing the test bench code, a wrapper file, a CSV file from the IWB, and compilation files for latency and bandwidth measurements. The wrapper file, a high-level module in digital design, encapsulates lower-level hardware components, serving as the top-level design entity and facilitating connections among lower-level modules, IP blocks, or components by specifying the data and control signal interactions.

5.3.1 Difference between configurations

The initial setup commenced with an interface data bit width of 32 bits, as previously outlined. For the AXI stream channel, various data strategies are available within the TLX-400 configurations in Socrates.

The default configuration strategy employs the sum of address width and data width for the forward direction, and the combination of read data width and response for the reverse direction. Although this strategy achieved 100% bandwidth utilization, it failed to offer optimization regarding routing congestion, as it does not reduce the wire count compared to the NIC-400 interconnect.

To decrease the number of wires transmitting data through the AXI stream channel in both the reverse and forward directions, the study opted for a user-defined width strategy. This approach allows for a reduced number of physical link widths—8, 16, or 24 for the data payload.

More specifically, regardless of the direction—forward or reverse—there are two interfaces for each. One is the AXI stream data interface, and the other is the flow control interface. The data interface comprises three signals: valid, ready, and data payload. Similarly, the flow control interface includes three signals: valid, ready, and control payload. The user-defined configuration setting allows adjustment of the data payload width, whereas the widths of other signals remain fixed.

To ascertain the total count of physical wires transmitting data through the

AXI stream channels, it is necessary to aggregate the bit widths of all signals. Table 1 exemplifies this with a user-defined width of 8, resulting in a total of 29 physical wires. Correspondingly, user-defined widths of 16 and 24 result in total physical wire counts of 45 and 61, respectively.

Channel	Payload	Valid	Ready	Sum
Forward data	8	1	1	10
Forward control	1	1	2	4
Reverse data	8	1	1	10
Reverse control	1	1	3	5

Table 1: User Defined Width of 8

The varying thickness of the wires could influence the area, the count of flip-flops, and the bandwidth. Thus, observing the contrast between these scenarios provided us with a comprehensive insight into the influence of the physical wires' thickness on the overall SoC performance.

5.3.2 Different Clock Strategies

As shown in figure 8 below, which is TLX-400 hierarchical structure, it highlights multiple clocks such as pl_fwd_clk, dl_fwd_clk and c0clk for forward direction, demonstrating the system's ability to apply different frequencies to specific layers. This flexibility allows for tailored clock speed to optimize performance and power consumption based on the operational requirements of each layer.

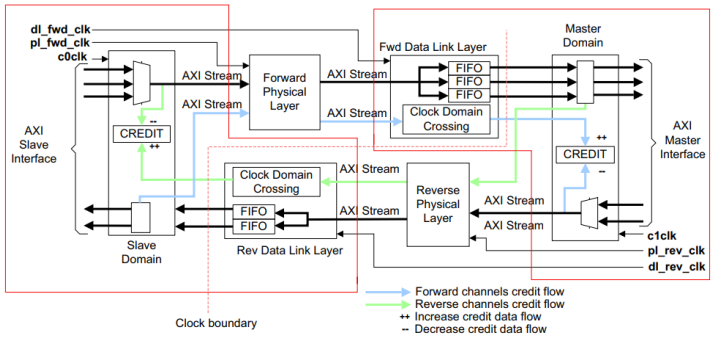

Figure 8: TLX-400 hierarchical structure

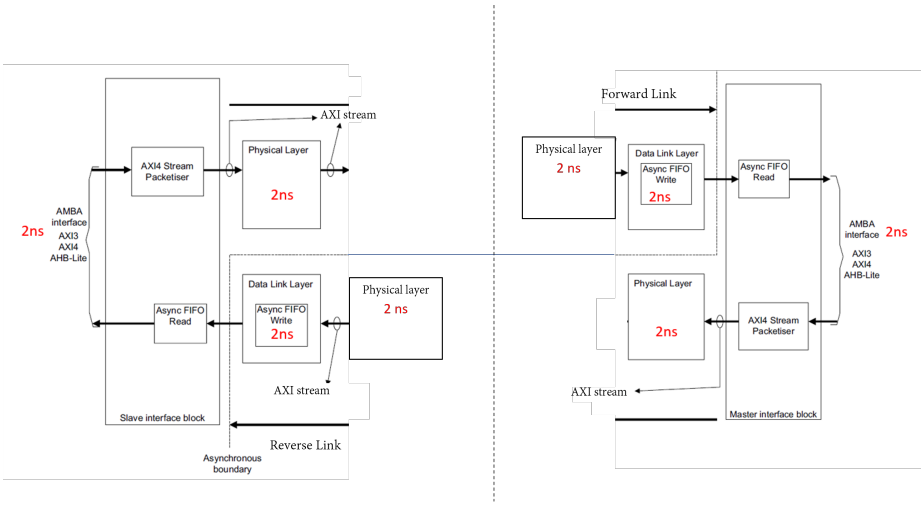

Figure 9: design block with period 2 ns

In the depiction provided in figure 9 above, which is one of the clock strategies in the following study, the red text highlights a uniform 2 ns clock period applied across various layers of the architecture. Specifically, the red text is strategically placed next to each layer (PL, DLL and IL in both forward and reverse links) to clearly indicate that all these components operate under the same 2 ns clock cycle. The other two strategies will be shown in following study, one is union faster frequency of 1 ns, another one is mixed frequency of 1 ns physical layer and 2 ns for the interface and data link layer.

5.3.3 AXI-Stream

Then the actual learning part of the thesis was accomplished when implementing the AXI-Stream link between subchips. We integrated our design into an existing design for the connect subchips. Our design contains parameters for the payload data width as well as a parameter for the number of register slices placed on the ThinLink. The reason for setting these parameters here was to make our design user-friendly, by allowing the user to simply change the number of register slices to see how the area and power are affected as well as the synchronization between these two and the register slices.

6 Results

The results in this thesis project come from the performance analysis and the synthesis, so these terms will be discussed in this section.

6.1 Performance Analysis

The performance analysis was achieved by running the IPA tool. It includes latency and bandwidth measurements for the data transfer between the manager and subordinate domains. Both latency and bandwidth are crucial measurements for optimization of a computing system since they affect the overall performance of the SoC. They contribute to improved system performance, responsiveness and energy efficiency. SoCs typically comprise multiple IP blocks responsible for different functions. Efficient communication between these blocks is crucial for the system's overall performance. This is where latency and bandwidth become important, as low latency and high bandwidth facilitate smooth interaction between the different components in SoCs. Performance analysis was verified only for the interconnect, and not for the interconnect when integrated into the complete subsystem with the connect subchips.

Specifically, in ARM's TLX hierarchy structures as figure 8 above, the interaction between the clock frequency of interfaces and the physical layers is impacting the overall bandwidth capabilities. This section aims to compare the design using the fully AXI4 protocol as communicating interface between sub-chips and the one that utilizes the TLX-400 structure as interconnect in which the AXI stream protocol shows up as the interface between sub-chips, then explore the implications of clock frequency distribution on the system.

6.1.1 Methodology and Verification

To conduct a thorough performance analysis, a controlled methodology is implemented in which all parameters, except the clock frequencies of the interfaces and the physical layers, are kept constant. This isolation enables a direct correlation between clocking strategies and bandwidth performance.

Cadence Interconnect Workbench (IWB) takes in the RTL code, then builds a performance-oriented testbench. When executing on Xcelium simulator, the coverage result and performance metrics will be collected and examined.

There are two types of transaction available to choose from, random transaction length and fixed transaction length. One is verification-oriented and another is performance-oriented. We will cover both in this section.

The benchmark for this study is set by the NIC-400 interconnect with a 2 ns clock period, written as fully AXI4 in the comparison table. Any deviations in bandwidth and latency are measured against this standard.

6.1.2 Bandwidth

Bandwidth is defined as the amount of data that can be sent and received in one second. The higher bandwidth allows for more data to be transferred simultaneously. In SoC, higher bandwidth is essential to handle large amounts of data.

The analysis is split into three clocking options, a uniform clock period of 2 ns, a uniform clock period of 1 ns, and a mixed clock period of 2 ns and 1 ns. Each of them was tailored for a specific requirement or limitation for project variation.

Randomly Generated Transaction Slower Clocking Scenario with 2 ns: as depicted in figure 10, the interfaces and physical layers share identical clock frequencies. Which is saying, we used the same frequency as the NIC-400 structure.

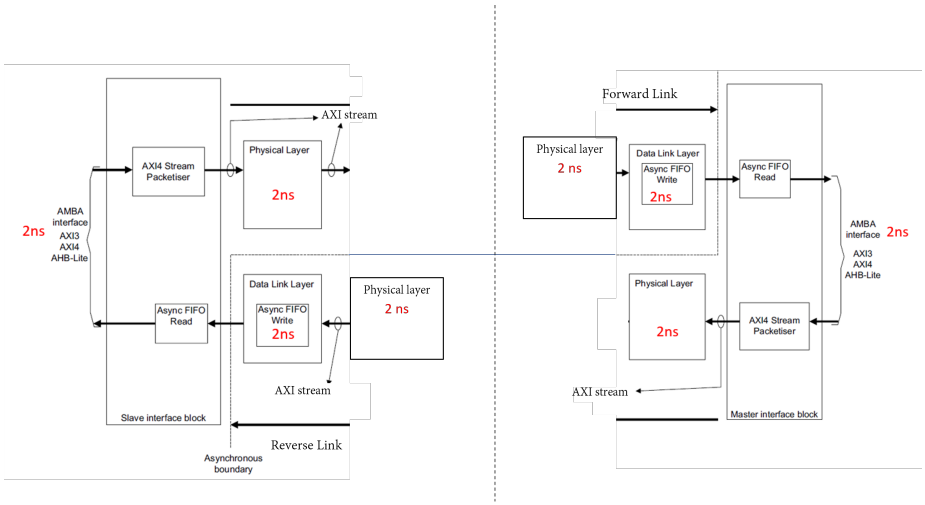

Figure 10: design block with period 2 ns

The purpose here is to observe the throughput and efficient purely caused by the structure difference despite the identical clock frequency. In the TLX-400 design, the bottleneck exists when the payload plus control signal width in total goes down to 29, 45 and 61, which is a huge thickness gap compared with NIC-400's interface signal width 256. If the efficiency behaves as expected, it will drop down below half of the NIC-400 interconnect.

The overview table conclusion from figure 11 suggests a gap in performance between NIC-400 and TLX-400 interconnects. Despite the architecture discrepancies, the operation remain an average read and write bandwidth around 50% percent of the NIC-400 benchmark.

Figure 12 shows a faster clock configuration for the hierarchical TLX-400 interconnect with a clock period of 1 ns, which double that of the established NIC-400 benchmark, synchronizes the interfaces and physical layers.

Theoretically, enhanced clock speed should proportionally augment bandwidth, provided other systemic parameters remain invariant.

While doubling the operational clock frequency could increase the data trans-

Interface	Period(ns)	Ave Read BW (MBPS)		Ave Write BW (MBPS)	
Fully AXI4	2	243		195	
AXI4 Stream Physical link 8	2	119	48%	142	72%
	1	238		284	
	mix	143		170	
AXI4 Stream Physical link 16	2	138	56%	167	85%
	1	276		334	
	mix	169		186	
AXI4 Stream Physical link 24	2	145	59%	179	91%
	1	290		357	
	mix	177		193	

Figure 11: bandwidth comparison with AXI4 register slice

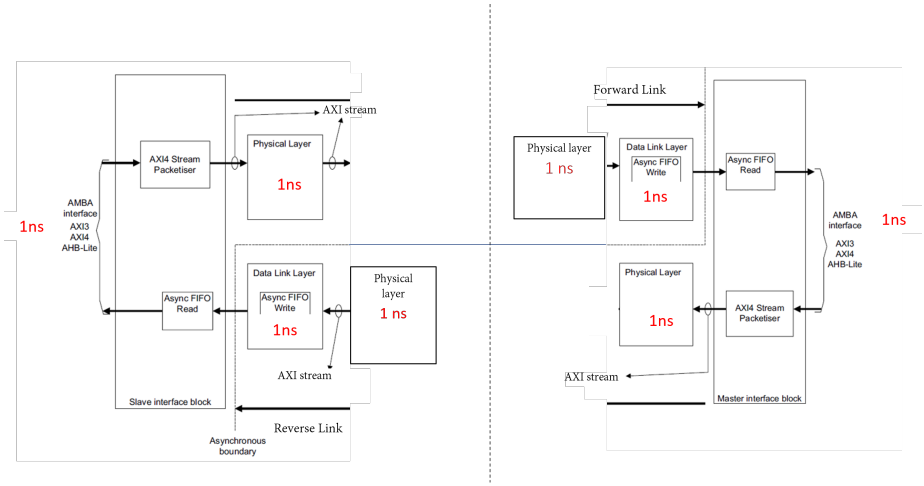

Figure 12: design with period 1 ns

mission capacity. In high-speed digital systems, where the currency is time, a reduction in clock period from 2 ns to 1 ns is significant. The consequent elevation in frequency enables the system to process and transmit twice the amount of data per unit time. However, this acceleration of pace is not without its trade-offs. The primary concern lies in the associated rise in power consumption.

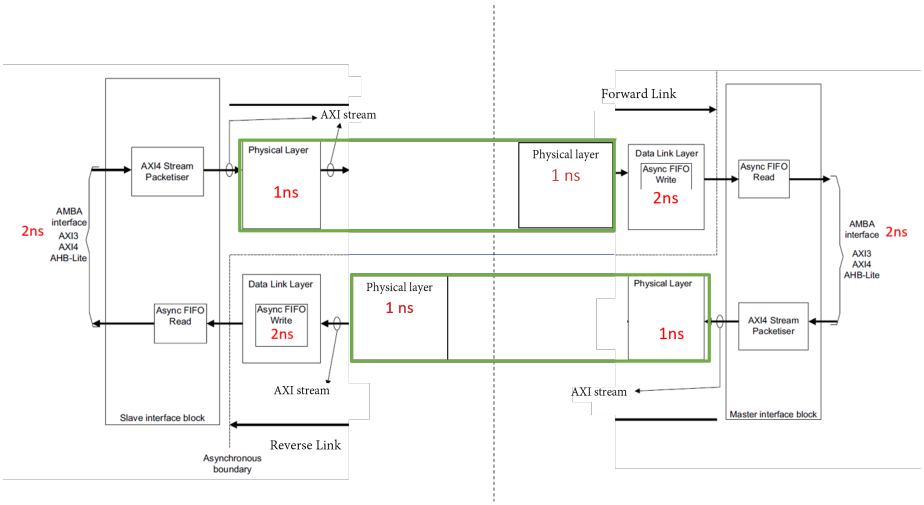

Figure 13: design with mixed period

Evaluation of SoC architectures needs to trade off between operational performance and power efficiency. The mixed frequency plan attempts to address the equilibrium. This proposal goes beyond uniform clocking period schemes, adopting an asynchronous frequency distribution across the interconnect's interfaces and physical layers.

The plan described in figure 14 raises the clock frequency of the physical layers to 1 ns, thus doubling the operational speed relative to the interfaces which remain at a clock period of 2 ns.

This approach is predicated on the assumption that accelerating the physical layer would enable rapid data handling and buffer management, without uniformly scaling power consumption across the interconnect.

The implementation of mixed frequency plan raises concerns regarding asynchronous operation and clock domain crossing complexities. However, it mitigates these challenges by confining the higher frequency, and thus less buffering requirements, to the physical layer where the impact on overall power budget is more controlled. The interfaces, operating at a more conservative frequency, impose a reduced power demand, potentially offsetting the total power budget

Interface	Period(ns)	Ave Read BW (MBPS)		Ave Write BW (MBPS)	
Fully AXI4	2	243		195	
AXI4 Stream Physical link 8	2	119		142	
	1	238		284	
	mix	143	58%	170	87%
AXI4 Stream Physical link 16	2	138		167	
	1	276		334	
	mix	169	69%	186	95%
AXI4 Stream Physical link 24	2	145		179	
	1	290		357	
	mix	177	72%	193	98%

Figure 14: bandwidth comparison with AXI4 register slice

increase.

The bandwidth comparison in figure 14 shows the effectiveness of the mixed frequency. Interfaces with mixed periods' physical layer demonstrate gains in average read and write bandwidths, when compared to a benchmark of NIC-400 operating at a 2 ns period. This proves the capability of the mixed frequency to enhance bandwidth.

The findings from the three clocking strategies—uniform slower, faster, and mixed frequency—revealed two outcomes. First, apply various periods to interface and physical layer impact the bandwidth. Uniform scaling of period is proportional to bandwidth. Increase the physical layer's clock period while remain the speed of interface also shows a better bandwidth performance. Secondly, it shows a consistently high bandwidth retention across all scenarios.

The initial expectation is the observed bandwidth will significantly decrease to the predicted levels—approximately 20% lower than the NIC-400 benchmark, due to the inner wires experienced a 80% reduction.

The implemented strategies, while yielding high bandwidth, did not align with the initial hypothesis, prompting further investigation into the underlying causes of this discrepancy.

There are two assumptions. Firstly, the TLX-400 architecture may possess capabilities of facilitate data transfer rates even under condition of reduced interconnect wires. Secondly, the verification method of utilizing random transaction length didn't explore the boundary of bandwidth.

Therefore, fixed transaction lengths is proposed to isolate the effect of transaction size variability on bandwidth.

Fixed Transaction Interconnect performance characterization test suite generates predefined characterization tests, maximum bandwidth tests and minimum latency tests.

Maximum bandwidth tests provide the option of incrementally increase the length of the burst. The purpose here is to saturate out DUT with traffic. When running the tests with an ever increasing length, it can be seen at what point the DUT start chocking, that is, What is the DUT capability to handle high bandwidth traffic. And after loading the regression data into IPA tool, the bandwidth limits will show up within the diagram.

Minimum latency tests transmit individual read and write burst with different length. The purpose of the tests is to see the minimum latency for propagation of wires and reads of different length through the DUT.

Transaction Type	Transaction Length (Bytes)	Number of transactions	Write Bandwidth (MBPS)	Read Bandwidth (MBPS)	Average Write Latency(ns)	Average Read Latency(ns)
Read Only	1	25		1852		6
Read Only	16	34		1993		36
Read Only	64	51		1999		132
Read Only	128	47		1999		260
Read Only	256	23		1999		516
Write only	1	34	1889		6	
Write only	16	63	1996		36	
Write only	64	44	1999		132	
Write only	128	83	2000		260	
Write only	256	56	2000		516	
Read and Write	1	37	1750	1280		
Read and Write	16	31	1979	1987		
Read and Write	64	24	1995	1995		
Read and Write	128	48	1999	1999		
Read and Write	256	65	2000	2000		

Figure 15: IPC test suite benchmark of NIC-400

To evaluate the performance characteristics of the NIC-400 interconnect, a set of tests was conducted, the results of which are summarized in figure 15.

This benchmark covers an array of transaction types, each with varying lengths, to simulate the operational scenarios the interconnect may face in practical environments. The test suite is segmented into three principal transaction categories: read-only, write-only, and a combination of read-and-write operations. These are further classified based on transaction lengths, ranging from 1 byte to 256 bytes, allowing for an in-depth evaluation of the interconnect's performance across a spectrum of use cases. Bandwidth is assessed in terms of throughput, reported in megabits per second (MBPS), while latency is recorded in nanoseconds (ns), providing insight into the efficiency of the interconnect system.

In summary, The interconnect performance in figure 15, as measured by fixed transaction lengths, provides a clear view of the peak performance characteristics of the NIC-400 benchmark. It serves as a baseline against which the performance of alternative interconnect configurations, such as those employing different clocking strategies or physical layer adjustments, can be measured.

The next three figures depicts TLX-400 with physical link configuration of 8, 16 and 24 under clock period of 2 ns.

Transaction Type	Transaction Length (Bytes)	Number of transactions	Write Bandwidth (MBPS)	Read Bandwidth (MBPS)	Average Write Latency(ns)	Average Read Latency(ns)
Read Only	1	94		178		66
Read Only	16	61		222		332
Read Only	64	61		222		1196
Read Only	128	20		222		2348
Read Only	256	62		222		4652
Write only	1	26	120		68	
Write only	16	92	254		288	
Write only	64	45	263		1008	
Write only	128	49	265		1968	
Write only	256	81	266		3888	
Read and Write	1	65	72	103		
Read and Write	16	45	234	220		
Read and Write	64	24	257	221		
Read and Write	128	25	263	222		
Read and Write	256	81	265	222		

Figure 16: IPC test suite for TLX-400 with physical link of 8 under 2 ns

Transaction Type	Transaction Length (Bytes)	Number of transactions	Write Bandwidth (MBPS)	Read Bandwidth (MBPS)	Average Write Latency(ns)	Average Read Latency(ns)
Read Only	1	97		245		50
Read Only	16	97		266		268
Read Only	64	46		266		988
Read Only	128	83		267		1948
Read Only	256	59		267		3868
Write only	1	88	218		52	
Write only	16	88	307		240	
Write only	64	20	307		864	
Write only	128	83	308		1696	
Write only	256	82	308		3360	
Read and Write	1	47	136	185		
Read and Write	16	48	297	266		
Read and Write	64	12	305	266		
Read and Write	128	43	307	267		
Read and Write	256	25	307	267		

Figure 17: IPC test suite for TLX-400 with physical link of 16 under 2 ns

Transaction Type	Transaction Length (Bytes)	Number of transactions	Write Bandwidth (MBPS)	Read Bandwidth (MBPS)	Average Write Latency(ns)	Average Read Latency(ns)
Read Only	1	100		280		44
Read Only	16	70		285		246
Read Only	64	70		286		918
Read Only	128	86		286		1814
Read Only	256	20		286		3606
Write only	1	46	274		44	
Write only	16	53	333		216	
Write only	64	59	333		792	
Write only	128	36	333		1560	
Write only	256	80	333		3096	
Read and Write	1	81	179	271		
Read and Write	16	39	317	285		
Read and Write	64	53	330	285		
Read and Write	128	22	332	285		
Read and Write	256	66	332	286		

Figure 18: IPC test suite for TLX-400 with physical link of 24 under 2 ns

Figures 16, 17, and 18 present the outcomes of tests conducted with fixed transaction lengths and different physical link configurations under a 2 ns clock period, offering a detail view of the interconnect's throughput and latency across various transaction scenarios.

The performance ceiling of the TLX-400 was observed to be reached as the transaction length increased, conforming to the theoretical models that suggest a diminishment of bandwidth efficiency with larger data sizes. Besides, the bandwidth remains 10-20% compared to the benchmark performance of the NIC-400 interconnect, aligning with expectations that indicate a direct correlation between reduced internal wires, optimized by using AXI Stream protocol, and reduced bandwidth.

The result underscores the influence of verification methodologies on performance. The high percentage bandwidth retained compare to benchmark of NIC-400 during tests with random transaction lengths can be attributed to the verification method rather than the intrinsic capabilities of the TLX-400 structure itself.

Across all transaction types—whether reading, writing, or a combination of both—the augmentation of the physical link width from 8 to 24 consistently resulted in increased bandwidth. A wider physical link provide a thicker payload width, thus bolstering the throughput of the interconnect.

In summary, the findings affirm that the TLX-400 interconnect, when assessed with the fixed transaction length approach, exhibits a predictable performance profile. And the choice of configurations provide the future projects with different interconnect options.

6.1.3 Latency

Latency is defined as the amount of time that data use to reach its destination and return to the source. Since it is a measure of delays in the system, it is desirable to keep it as low as possible. Low latency in the SoC ensures quicker

response times, contributing to a more responsive system.

As we observed from figures 16, 17, and 18, the narrower widths in the TLX-400 can lead to increased serialization of data, which might contribute to higher latency when compared to the NIC-400's wider interfaces that allow for more parallelism in data transmission. This difference in data bus width directly impacts the amount of data that can be transferred per clock cycle, which in turn affects the overall latency of the interconnect system.

Besides, each additional register slice, while potentially facilitating timing closure, incrementally contributes to the overall latency. Currently, alternative strategies to mitigate this latency have not been thoroughly investigated

6.2 Synthesis

The synthesis flow consists of two parts, elaboration and compilation. Synthesis is the process of converting a high-level hardware description of a digital circuit into a netlist, which is a representation of the circuit including components such as gates and flip-flops. The synthesis tool takes an HDL input such as SystemVerilog and analyzes the code to generate a structural representation of the circuit (Pandey 2023).

6.2.1 Area

When running synthesis, the tool performs optimizations to improve the performance, area and power consumption. What has been especially important to optimize in this thesis, was to reduce the physical area for the wires between the two subchips. This area has been reduced by reducing the number of physical wires in between subchips, which has been accomplished by implementing register slices on the ThinLink.

When we started running synthesis on the RTL code, we did it on the subchips containing only the NIC-400 register slices. We kept this result as our golden reference, the one that we have compared all of our other synthesis results with. In our design, we kept some NIC-400 register slices and then we integrated the TLX-400 register slices that we implemented in the design for the connect subchip. By doing so, we were able to see how the area was affected when adding the register slice. On the link between the two subchips where we placed the TLX register slice, we substituted the AXI-Stream slice with the TLX register slice and then we compared this result with our golden reference. When running synthesis, we changed the number of register slices for a physical link of data width 8, 16, 24 and 32 until the number of flip-flops, i.e., the area, decreased. It was important that we saw a decrease in area, otherwise our design would not be efficient enough to replace the AXI-Stream register slice.

After running synthesis for all the physical link data widths multiple times, we

have come to a conclusion about how many register slices are needed on the two connect subchips respectively. The manager and subordinate do have a different number of register slices, since they have different attributes. These numbers of register slices are defined by a decrease of the total amount of flip-flops, which means that the physical area between the subchips will also decrease. This is the purpose of the master's thesis, so finding the correct number of register slices has been important. It is not useful to place more register slices than needed since it is costly in terms of economy? bandwidth? Therefore, the chosen number of register slices for each physical link data width has been selected according to the lowest possible number when seeing a decrease in the number of flip-flops.

The AXI-Stream register slice had around 400 physical wires, while the TLX-400 register slice had around 200 wires for its default value. Our aim was to decrease the number of physical wires as much as possible in order to reduce the area between the connect subchips. For the physical link data width 8, there were 29 physical wires, for the physical link data width of 16, there were 45 physical wires and then for the physical link data width of 24 there were 61 physical wires. By stating this, it is obvious that a physical link data width of 8 is the most suitable option at first sight, but then area and performance have not yet been considered, which also must be taken into consideration. For one of the subchips, the width of 8 bits required a lower number than the width of 24 bits, which makes the selection easy for the user when not considering performance.

Area result including manager domain in subchip x

Subchip	Register slices	Physical wires	Flip-flops
x	1	200	34474
x	2	200	34991
x	3	200	35508

Table 2: Golden reference for subchip x

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
x	1	29	35080	+606 (up 1.76%)
x	2	29	35142	+151 (up 0.4%)
x	3	29	35204	-304 (down 0.8%)

Table 3: 29 physical wires between the connect subchips

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
x	1	45	35112	+638 (up 1.86%)
x	2	45	35208	+297 (up 0.6%)
x	3	45	36104	-204 (down 0.58%)

Table 4: 45 physical wires between the connect subchips

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
x	1	61	35145	+671 (up 1.9%)
x	2	61	35275	+284 (up 0.8%)
x	3	61	35405	-103 (down 0.3%)

Table 5: 61 physical wires between the connect subchips

According to the three tables above, table 2, 3 and 4, three register slices are needed for the manager domain before the number of flip-flops decreases, i.e., a decrease of area, compared to the golden reference. The last column, difference in flip-flops, is a comparison with the golden reference on top. Area is one of the three main aspects to consider in this thesis, and seeing a decrease is important because it means that we can use fewer physical wires which in turn reduces routing congestion. Therefore, in terms of area, three register slices are needed for the manager to have a smaller area no matter how many physical wires are used between the subchips. As stated above, in this thesis, 29, 45 and 61

physical wires have been used between the connect subchips and they require two or three register slices for the manager domain in between in order to decrease the area. A smaller area also results in using less static power consumption, which is a topic not considered in this thesis, yet worth to mention since using less power can be of importance for developers. Using less power consumption is a topic that can be further analyzed in a future project.

Area result including subordinate domain in subchip y

Subchip	Register slices	Physical wires	Flip-flops
y	1	200	21904
y	2	200	22425
y	3	200	22946

Table 6: Golden reference for subchip y

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
y	1	29	22227	+323 (up 1.48%)
y	2	29	22289	-136 (down 0.6%)

Table 7: 29 physical wires between the connect subchips

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
y	1	45	22259	+355 (up 1.6%)
y	2	45	22355	-70 (down 0.3%)

Table 8: 45 physical wires between the connect subchips

Subchip	Register slices	Physical wires	Flip-flops	Difference in flip-flops
y	1	61	22292	+388 (up 1.78%)
y	2	61	22422	-3 (down 0.013%)

Table 9: 61 physical wires between the connect subchips

According to the three tables above, table 6, 7 and 8, two register slices are needed for the subordinate domain before the number of flip-flops decreases, i.e., a decrease of area, compared to the golden reference. The last column, difference in flip-flops, is a comparison with the golden reference on top. As previously mentioned, area is one of the three main aspects to consider in this thesis, and seeing a decrease is important because it means that we can use fewer physical wires which in turn reduces routing congestion. Therefore, in terms of area, two register slices are needed for the subordinate to have a smaller area no matter how many physical wires are used between the subchips. As stated above, in this thesis, 29, 45 and 61 physical wires have been used between the connect subchips and they all require two register slices in between in order to

decrease the area.

As can be seen in the tables above, the difference in flip-flops compared to the golden reference, increases when using more physical wires between the subchips. This is a desired pattern, since the aim is to use as few physical wires as possible. When using the lowest possible number of physical wires, 29, the number of flip-flops decreases by 0.6%, which is the largest decrease when comparing all the three options for the physical wires.

Since the three main aspects in this thesis are routing congestion, area and performance, they all have to be considered when describing a solution. Although it should be mentioned that there is not just one ideal solution, it rather depends on what the user expects. Less wires result in a smaller area which results in a decrease in terms of performance. If a decreased area is of importance, then the user should go for this alternative, and if an increase in performance is desired, then go for this alternative.

As can also be seen in the tables above, the number of flip-flops decreases when the number of physical wires decreases, just as expected. Less wires means less flip-flops and thereby smaller area. For the case of 29 physical wires, there is a difference of 62 flip-flops between each addition of a new register slice, 96 flip-flops for 45 physical wires and 130 flip-flops for 61 physical wires.

This result is valid for both the connect subchips, since we have tried with the same number of physical wires between them both. As a conclusion of this, we can see that less and less flip-flops are added when decreasing the number of physical wires, which is a design we expect since it is desired to use as few wires as possible. This means that in terms of area, the most suitable result to choose is 29 physical wires and three register slices.

Therefore, regarding area, the most suitable option for the user is to choose the first alternative with 29 physical wires for both the connect subchips, as these alternatives only require two and three register slices respectively before seeing a decrease in the number of flip-flops as well as the lowest amount of physical

wires between the subchips.

Conclusion regarding area, performance and routing congestion

As a conclusion to this thesis project, it should be mentioned that there is not one single ideal solution since the thesis contains three main aspects to consider - area, performance and routing congestion. As a result of this, there are several solutions regarding routing optimization. It all depends on which parameters - area, latency or bandwidth are of importance for the developer.

Routing congestion refers to physical wires, which means that using less wires will result in a system less susceptible to noise, interference and crosstalk. Less wires between subchips has resulted in smaller channels to route signals between the hardened IPs. In our synthesis report, we can see that the number of flip-flops, and thereby the area, decreases which means that we have found an optimization regarding routing congestion by minimizing the number of wires. Hence, by reducing the number of wires in between the connect subchips, area decreases and routing congestion is optimized.

The results of the IPC test suite for the TLX-400 interconnect demonstrate that performance metrics such as bandwidth and latency are closely related to the choice of physical link configuration, which is area. Opting for a design that employs fewer wires, and consequently, a more pronounced area reduction. This adaptation result in increased latency and diminished bandwidth. Conversely, a design that incorporates more wires, while achieving less area reduction, tends to enhance bandwidth and reduce latency a bit, highlighting a performance improvement.

Ultimately, the TLX-400 interconnect emphasizes the balancing among design elements. Area reductions must be weighed against potential impacts on latency and bandwidth, ensuring tailored solutions for system requirements. In terms of area, it is desirable to use as few wires as possible to decrease area. In terms of bandwidth, as many physical wires as possible should be used to increase

bandwidth. Therefore, using TLX-400 is mostly suitable when it comes to area reduction. This study underlines the necessity for navigating these trade-offs, ensuring that optimizing one aspect does not compromise the system's overall functionality.

7 Discussion and further improvements

This thesis has really helped us to gain a broader perspective and understanding of how SoC design is implemented. We have learnt all the steps needed to create functions on a chip, for example the fact that digital design starts with implementing the design in RTL code and then continuing with verification by compiling the RTL code using a test bench. Thereafter, we performed the performance analysis and last of all, the synthesis. It has been interesting to go through the complete work flow, as this will be an advantage when we start working after graduation. We both had knowledge about performance analysis and synthesis before starting with the thesis, but barely any hands-on experience from it. Therefore, it has been rewarding to work with the entire flow as we have been able to implement the theoretical knowledge we had prior starting with the thesis into actual work. Synthesis is a very important step in digital ASIC design and we have spent a lot of time running it.

We implemented a TLX-400 register slice as a replacement to the existing NIC-400 register slice in between connect subchips. Our aim was to place this register slice on channels between connect subchips, using different bit widths. Due to lack of time, we were only able to run synthesis and do the performance analysis on the physical link data width of 32 bits, and not for all physical link data widths such as 64 and 256. Therefore, an improvement would be to run a synthesis to get an area result and do performance analysis on all the physical links, compare the result, and then analyze how the placement and the number of register slices differ depending on which link the register slice is placed on.

The depth of the FIFO buffer emerges as another variable. An increased FIFO depth is capable of managing greater volumes of serialized data transfers across a given number of clock cycles, impacting bandwidth. This enhancement is also beneficial for clock domain crossing (CDC) between the producer and consumer, as it can bridge frequency disparities. Within the experimental framework, the

maximal tested frequency differential between and the physical layer and the interface was a factor of two. If the depth is deeper, the differential could be higher.

Another topic would be to analyze the affect of power consumption more, especially static power consumption as this is related to area. While power considerations are integrated into the performance analysis phase, we did not actually measure it. Since this topic is important when developing radio connectivity, it would be crucial to obtain precise quantifications of power usage and see how power can be affected depending on how many register slices are used in between the connect subchips.

The research conducted in this thesis may serve as a reference for Ericsson's forthcoming projects, particularly when evaluating the NIC-400 and TLX-400 interconnects for integration into designs. The thesis delineates three TLX-400 configurations, documenting their respective performance metrics and area. Additionally, a comparative analysis of the two interconnects is presented. The thesis also underscores the flexibility of the TLX-400's design, allowing for the adjustment of register slices to meet specific timing closure requirements.

References

Arm (1999), 'Introduction to the AMBA Buses', *Arm* .

Arm (2016*a*), 'ARM CoreLink NIC-400 Network Interconnect', *Arm* .

Arm (2016*b*), 'ARM® CoreLink™ TLX-400 Network Interconnect Thin Links', *Arm* .

Arm (2021), 'AMBA AXI-Stream', *Arm* .

Arm (2022), 'Learn the architecture - An introduction to AMBA AXI', *Arm* .

Arm (2023), 'AMBA AXI Protocol', *Arm* .

Brooks, E. (2019), 'A Guide To ASIC Design', *System to ASIC* .

Cadence (2016*a*), '', *University of Idaho* .

Cadence (2016*b*), 'Renesas Adopts Cadence Interconnect Workbench to Accelerate Performance Analysis and Verification of On-Chip Interconnect', *PR Newswire* .

Instruments, N. (2023), '', *National Instruments* .

Pandey, P. (2023), 'ASIC Backend Design Flow in VLSI (Part-1) : RTL/Logic Synthesis', *Microship* .

Routing Congestion: The Growing Cost of Wires in Systems-on-Chip (n.d.), *Design and Reuse* .

Shirshendu, R. (2020), 'Placement and Routing for ASIC', *Digital System Design* .

Shukla, I. (2022), 'Enhancing VLSI Design Efficiency: Tackling Congestion and Shorts with Practical Approaches and PnR Tool (ICC2)', *Design Reuse* .

Wallace, J. (2017), 'The future of tooling from IP configuration to SoC verification', *Arm* .

Xilinx (2022), 'AXI Register Slice', *Xilinx* .

Printed by Tryckeriet i E-huset, Lund 2024

LUND
UNIVERSITY

Series of Master´s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2024-975
http://www.eit.lth.se