Good morning, everyone.

Today, I am here to present my progress on the project titled "Enhancing On-Chip Network Prediction with Advanced AI Techniques." It is an honor to share this work with you. My supervisor for this project is Doctor Libin Mathew.

Outline of Presentation

My presentation is divided into four main parts:

1. Project Overview
2. State of the Art
3. Progress Review
4. Future Plan

Part 1

Let's start with an overview of the project.

This presentation focuses on the Network-on-Chip (NoC), a critical communication subsystem integrated into chips, predominantly used in system-on-chip designs. The NoC is complex, comprising a network of wires and routers arranged in a grid layout, as depicted in Figure 1. A key component of the NoC is the network interface module, which converts data packets from various IP blocks for efficient transmission across the network. The importance of NoC stems from its scalability, high performance, low latency, power efficiency, and design flexibility. These benefits drive continuous research and development, leading to significant technological advancements.

The core objective of my project is to enhance NoC technology using advanced AI techniques. Specifically, I aim to improve the performance and design flow of NoCs to meet future high-performance computing demands. This involves optimizing predictive models, improving performance metrics, reducing reliance on traditional simulation methods, implementing innovative AI algorithms, and supporting the evolution of mobile computing. These efforts contribute to improving the efficiency and adaptability of NoC systems.

Part 2

Now, let's move to the state of the art in this field.

I have summarized previous works in a table that highlights the references, simulation tools used, and AI technologies implemented. The table also outlines the advantages and limitations of these works. For example, previous works achieved high accuracy, ranging from 88% to 95%, in predicting specific NoC parameters and significantly sped up the prediction process. However, they are often limited to specific NoC configurations and require high computing resources. My project aims to use a lightweight model to achieve higher prediction accuracy and include more evaluation indicators.

For the simulator, most researchers use BookSim2, a robust NoC simulator. It supports various network topologies, routing algorithms, and traffic patterns, allowing for extensive large-scale network simulations. Users can customize parameters such as traffic patterns and buffer sizes and manage the simulator through configuration files and a command-line interface. I have chosen BookSim2 as the simulator for this project.

Part 3

In this section, I will review my progress, explain the work completed, and share the results obtained so far.

Figure 2 shows my initial goals and milestones. I had to make a few changes due to an exam week and some course final project submissions, which pushed the project timeline back a bit. This image represents the true and complete timeline.

Figure 3 outlines the prediction framework of my project, divided into Training and Testing Phases. The Training Phase involves generating data using BookSim2, recording results, data preprocessing, segmenting the dataset, selecting an AI algorithm, and training the model using linear regression. The Testing Phase involves utilizing the trained linear regression model, validating predictions against new simulations, and evaluating the results.

For dataset preparation, I used BookSim2 to generate the required data. Figure 4 shows the configuration details, and Figure 5 illustrates the pseudocode for dataset generation. Figures 6, 7, and 8 present examples of a configuration file, a result file, and part of the dataset abstracted from the result files, respectively.

Data preprocessing is crucial for the training phase. I have summarized this process in a flow chart and described the steps in detail. These steps include loading data, checking data types, applying one-hot encoding to categorical data, converting the data into a Pandas DataFrame, splitting the dataset into training and testing sets, and defining the feature matrix and target vector.

For AI model selection and training, I chose linear regression due to its simplicity and interpretability, making it suitable for predicting continuous values. I used a MultiOutputRegressor with linear regression to handle multiple target variables simultaneously.

Model evaluation metrics include Mean Squared Error (MSE), R-squared (R^2), Variance, and Accuracy. These metrics help assess the quality of my model. The formulas for these metrics are well-known in deep learning evaluation, so I will skip the details here.

For model visualization, I used line plots to compare actual and predicted values for different NoC performance metrics and histograms to analyze error distribution. The main prediction parameters are Packet Latency Average, Hops Average, Network Latency Average, and Total Run Time. The line plots show that the overall difference between actual and predicted values is relatively small, with similar trends. However, the performance for Hops Average is worse, and the overall time using the linear regression method is significantly faster.

The error distribution analysis reveals that for packet delay, network delay, and time, most prediction errors are concentrated around zero, indicating high accuracy. However, for hop count, the errors are more dispersed, indicating some inaccuracy in predictions.

Throughout the project, I encountered several challenges. Learning to configure the NoC network using BookSim2 was difficult due to its steep learning curve. My solution was to study the official documentation and practice simulations. The long simulation run time, taking about four days per round, was inefficient. I optimized parameters and used parallel processing to address this. Data collection and integration posed challenges, which I overcame by using Python scripts for automation and Excel for data formatting. Finally, applying and validating the AI model required thorough testing, benchmarking comparisons, and careful fine-tuning to ensure accuracy and reliability.

In summary, the results show high accuracy levels across most metrics, particularly excelling in network latency predictions. However, there is room for improvement in predicting hops and total run time. I will continue to work on these aspects to achieve better performance.

Part 4

In the future, my main focus will be on writing my thesis. However, I will continue to tweak relevant details, such as trying different training models like CNN or DNN to see if the results improve, and exploring why the prediction of hop averages is not accurate. I expect to have the first draft of my thesis finished by the end of June and will discuss it with my supervisor for further revisions.

End

That's all I have to report today. Thank you for your attention, and I look forward to your comments and discussions!