Project

Course	💻 Computer Architecture
Date	@November 21, 2023
Type	Assignment
Status	Not Started

Computer Architecture Project

Designing a Cycle Accurate Simulator for Network-on-Chip (NoC) router and mesh with provision to study the impact of Process Variations

This C++ program simulates a network of routers using two routing algorithms i.e. XY and YX. It reads traffic data from a file named traffic.txt and simulates the movement of flits (data packets) from source routers to destination routers through the network. The simulation results are logged in log.txt and a summary report is generated in report.txt.

Prerequisites

- Before running the simulation, make sure you have the following:

  → C++ compiler installed on your system.

How to Use

Compilation:

- Compile the program using a C++ compiler. For example, using g++:

  ```
  → g++ -o Noc Noc.cpp
  ```

Execution:

- Run the compiled program with the routing type as a command-line argument. For example, for XY routing:

```
→ ./Noc XY
// OR
→ ./Noc YX
```

Input:

- The program reads input traffic data from the traffic.txt file. Make sure this file exists and contains valid traffic data. Each line of the file should be in the format: Clock cycle Source_Router Destination_Router Flit_Binary.
- BONUS: The simulator support a modified traffic file in the format: Clock cylce Source_ID Packet (instead of a single flit, the packet has all three types of flits concatenated).

Output:

- The simulation results are logged in the log.txt file, detailing the movement of flits at each cycle.
- The summary report, including delay information, comparison between PVA and PVS mode is stored in the report.txt file.
- Paths taken by each flit are listed in the log.txt file.
- The performance of PVA mode surpasses that of PVS mode, primarily because it requires fewer clock cycles to transmit a packet from the source router to the destination router. In PVS mode, the router delay occasionally exceeds the clock cycle, necessitating an additional cycle for completion.

Files

- **Noc.cpp:** Main C++ source code file containing the simulation logic.
- **traffic.txt:** Input file containing traffic data (format: Cycle Source_Router Destination_Router Flit_Binary).
- **Delays.txt:** Configuration file containing delay values.

- **log.txt:** Output file logging the simulation details.
- **report.txt:** Output file containing the summary report.

Assumptions:

1. The packets from the traffic file will come in increasing order and the first clock cycle will always be 0.
2. If the source and the destination are the same, then there will be no output.
3. The simulation is doing congestion on the basis of first come first serve and a packet ID is being formed according to it.