**On-Chip Interconnection Network**

**CSEE 4340:**

**Computer Hardware Design**

**Fall 2012**

**SECTION 1:**

**Project Title: On-Chip Interconnection Network**

**Team Name:** Flit Busters

**Project Members:** Adil Sadik (ams2378)

Ashwin Ramachandran (ar2648)

Dechhin Lama (ddl2126)

Ayushi Rajeev (ar3110)

**Project Member Duties:**

**Design Team:**

* Adil Sadik (Master)
* Dechhin Lama

**Verification Team:**

* Ashwin Ramachandran (Master)
* Ayushi Rajeev

**Section 2:**

**Design Overview:**

In this project, we will be designing and testing a 4x4 on-chip interconnection network using SystemVerilog HDL.

As the number of IP-modules on System-On-Chip’s increases, bus based interconnections can no longer meet the necessary bandwidth, latency and power consumption requirements. Network on Chips provide a more efficient and scalable solution to this problem. NoC’s consist of routers that are connected to each IP-module. Routers are then connected in various topologies such that they can communicate to neighboring routers (nodes in a network).

\*Adil\*

In this project, we will design a 4x4 Network on Chip of mesh topology. A NoC consists of multiple communicating cores, intermediate router nodes and point-to-point links. Intermediate router nodes are responsible for routing data to its appropriate destination, and thus establish a notion of point-to-point communication between a sender and receiver. Data is represented by *messages*.A *message* can consist of multiple packets, where each packet is composed of multiple *flits*. The very first flit of each packet (*header flit*) contains the information required to locate the destination node. All of the following flits (*body flits*) contain actual messages related to the data.

The placement of routers and cores is realized with a two-dimensional grid, where each node is identified with a unique (X, Y) co-ordinate. The routing algorithm used for this project is *deterministic*. Data will always be routed in a predefined order – first along the Y-axis and then along the X-axis. This is referred to as *dimension ordered routing*. After receiving the first flit of each packet, a router node will locate the information (i.e. co-ordinate) of the destination node and route the flit through the *Y-X dimension-ordered* routing scheme mentioned above. The NoC handles the messages in a *flit-by-flit* basis. This means that the router doesn’t wait until it receives all the flits associate with a packet; instead it transmits each flit as soon as it knows that the next router has available space in its input buffer. All the body flits associated with a header flit follow the same routing path established for the header flit. This is referred to as *wormhole routing*.

To prevent resource starvation and congestion inside the network, the NoC will implement a *distributed* *flow-control* mechanism. Specifically, we use *credit-based flow-control,* which ensures that a router will only transmit data when there are available slots in the next router to accommodate the received flit. Hence, each router keeps tracks of the available free slots in the input buffers of its adjacent routers through the notion of *credits*. The number of credits indicates the number of free slots in the input data buffer of a router. By implementing *credit communication* among adjacent routers, we can ensure that the routers are keeping track of the available resources required to make routing decisions for each flit.

Each router node consists of 5 input and 5 output ports - one pair of input-output ports for data in the north, south, east, and west directions, with one extra pair for the local data. Each input port consists of a queue, which can store up to 5 incoming flits per direction. There is a crossbar switch, which routes one direction’s input to a different direction’s output, and 5 outputs buses, which are capable of sending one flit per cycle. In addition to this routing hardware, there exist credit counters that keep track of how many free slots exist in the neighboring router nodes’ input buffers.

For our project, we will be designing a 4x4 mesh network where packets (a unit of message comprising of one or more flits) are routed to various nodes of the network. Routers will examine the header flit and route the following flits to the desired destination. Since we are designing a 4x4 mesh network, each node will be assigned an X-Y co-ordinate in a two-dimensional grid. Messages addressed to a particular co-ordinate are then passed along the nodes of the network, first in the Y-axis followed by the X-axis. Each router (node) will have four input-output pairs connecting to neighboring routers and one input-output pair to the local module.

In order to prevent cases where a transmitting router sends data when the receiving end is not yet ready to accept, we will be using a Credit-based flow control. Here, each router will have a buffer queue for each input where packets are stored temporarily. The transmitting router will maintain a count on the number of free slots available on the receiving end and transmit data only when there are “credits” available.

We will begin our design such that only single header flits are routed. Once this is successfully implemented and tested, we will then extend our design such that the header flit is followed by 4 body flits that follow through with the header flit to the designated destination (i.e. Wormhole routing). Furthermore, we will optimize our design by modifying the routers on the edge of the network such that unnecessary ports are omitted.