University of Southern California

EE-577B spring 2016 – Project - Phase 3

10th April 2016

Andrew Adam

Harsh Bhardwaj

# Team Details

*Team Name -* ***JHH***

## Team Members

1. Andrew Adam [aadam@usc.edu](mailto:aadam@usc.edu)
2. Harsh Bhardwaj [hbhardwa@usc.edu](mailto:hbhardwa@usc.edu)

## Distribution of work

* Design of NIC – Andrew
* CPU changes for NIC interfacing – Harsh
* NIC test-bench – Andrew, Harsh
* CMP test-bench – Andrew, Harsh
* Verification of result – Andrew, Harsh
* Synthesis – Andrew

## NIC Test-bench

For the standalone NIC, we designed 9 tests, listed below:

#### Test 1 – Simple processor-to-router communication

In this test, we verified the processor store to NIC, and NIC-to-router send-handshake signals.

#### Test 2 – Processor-to-router communication with busy router

This test-case helped us verify the flow control of an outgoing packet when the router is busy and doesn’t accept packets until a few cycles later.

#### Test 3 – Processor-to-router communication with busy router and processor sending multiple packets

This test-case allowed us to verify that the output channel buffer doesn’t get overwritten until the first packet has been successfully sent out to the router.

#### Test 4 – Simple router-to-processor communication with incoming packet having an odd polarity

In this test-case, we verified if the NIC received the packet on the correct polarity, and if the input channel status and the input channel buffer work as expected. This also tested the processor special-load instruction.

#### Test 5 – Simple router-to-processor communication with incoming packet having an even polarity

This test-case was similar to the previous one except that the packet arriving from the router at the NIC had an even polarity.

#### Test 6 – Router-to-processor communication with busy processor

In this test-case, a packet was sent to the NIC from the router side but the processor was busy. This checked the ability of the input channel buffer to hold the packet till the processor was ready to receive the packet.

#### Test 7 – Router-to-processor communication with busy processor and router trying to send multiple packets

This test-case allowed us to verify the ability of the input channel buffer to not get over-written by new packets until the first packet has been received by the processor

#### Test 8 – Simultaneous router-to-processor and processor-to-router communication

This test-case checked if the NIC was able to carry out simultaneous two-way communication of packets.

#### Test 9 – Simultaneous two-way communication with busy router and busy processor

This test-case was similar to the previous, except that both the router and processor were busy and wouldn’t accept packets for few cycles.

## CMP Test-bench

For the CMP, the only test that was executed was the suggested gather scenario – where each router sends packet to all the other routers, and then proceeds to receive packets sent to itself.

This test allowed us to verify the basic functionality of the CMP – processors communicating with each other through a NOC router based on a packet-switching scheme.