IIT Armour College of Engineering
ILLINOIS INSTITUTE OF TECHNOLOGY

ECE 587

HARDWARE/SOFTWARE CO-DESIGN

PROFESSOR: JIA WANG

DUE DATE: 03/11/2016

PROJECT REPORT - 1

NAME: KRISHNA PRAMOD KANAKAPURA UMESH

CWID: A20337195

Network-on-Chip Prototyping

OBJECTIVE:

The main objective of this project is to setup the simulation environment for SystemC and to understand the behaviors of NoC routers and processing elements (PEs). Network-on-Chip (NoC) is a promising communication mechanism for future Multiprocessor System-on-Chip (MPSoC) systems. To understand basic NoC behaviors, we will build a NoC prototype in the course projects using the SystemC modeling language. In this project we also determine the utilization rates ad queue sizes of the routers and processing elements.

INTRODUCTION:

Network-on-Chip (NoC) is a communication subsystem on an integrated circuit (commonly called a "chip"), typically between intellectual property (IP) cores in a system on a chip (SoC). NoCs can span synchronous and asynchronous clock domains or use unclocked asynchronous logic. NoC technology applies networking theory and methods to on-chip communication and brings notable improvements over conventional bus and crossbar interconnections. NoC improves the scalability of SoCs, and the power efficiency of complex SoCs compared to other designs. A Network-on-Chip system consists of routers and processing elements (PEs) interconnected by communication links. The packets are generated and consumed by the PEs while the routers are responsible to route the packets from source PEs to destination PEs. For this project we assume that a single PE is attached to every router so that addresses can be to the PEs based on the locations of their routers within the network. Usually, the network is organized into a 2-D mesh and we assume so for our projects. Therefore, both the location of a router and the address of the PE attached to it are represented by a pair of integers (x,y) denoting the two coordinates within the network. Moreover, the 2-D mesh structure requires each router to to have four pairs of input/output ports that connect to neighboring routers on the four sides. Adding the pair of input/output ports for the router to communicate with the attached PE, we illustrate the router with these five pairs of ports in Fig. 1, and name them as NORTH, SOUTH, EAST, WEST, PE. Note that every PE should also have a pair of input/output ports such that it can communicate with the router it attaches to.

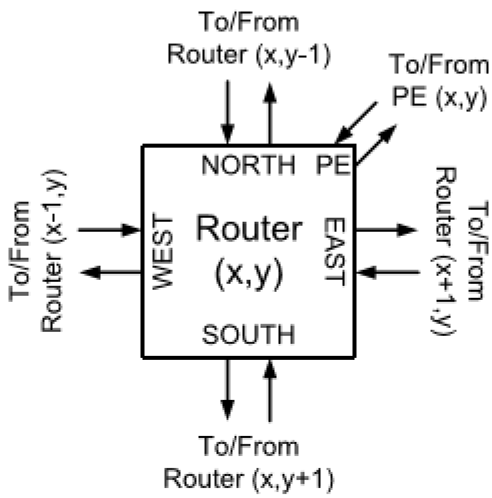


Figure 1: Router

For first part of the project, we consider a NoC architecture consisting of 2 routers, 2 PEs, and 6 communication links as shown in Fig. 2. We map the SDF model as shown in Fig. 3 to this architecture such that both processes PI and PO are mapped to PE(0,0) while the process P1 is mapped to PE(1,0).

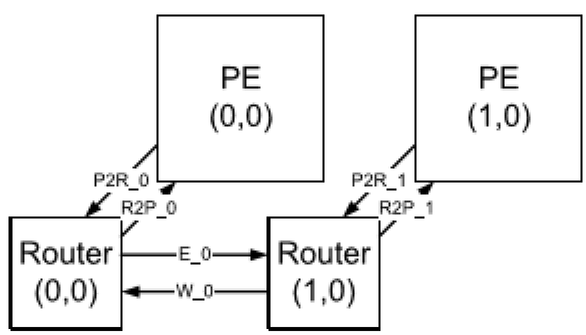


Figure 2: NoC with 2 routers and 2 PEs

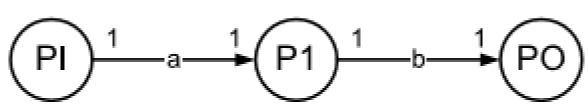


Figure 3: SDF model with 3 processes

IMPLEMENTATION:

Router:

The router is implemented in the files router.h and router.cpp. The packet is also defined in these two files. The router module contains various members to describe the structure of the router and its behavior. The basic functionality of the router is to read the packets from the input queue of the particular port, then to route the packet to the desired destination and lastly to the write the packet to the output queue.

The routing algorithm is implemented in the member function router::route_packet_xy. This algorithm is called XY routing since the packets are routed based on the location of the current router.

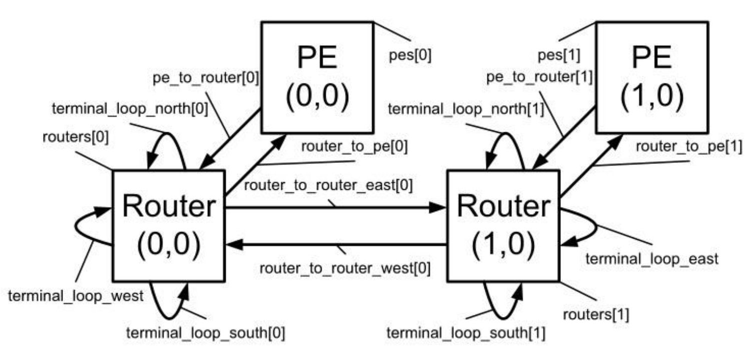
Processing Elements (PEs):

The PEs are implemented in the files pe.h and pe.cpp. We start with the PE_base module that serves as a base class for all PEs. This module defines the communication between the PE and the router and relies on the member function PE_base::execute to perform the actual computations per clock cycle that must be defined in a derived class.

Top Module:

This module basically defines the entire network such as the number of routers, PEs and the communication links between them. It initiates the PE, routers with default values and calls for setting their respective addresses.

NoC Architecture with 2 Routers and 2 PE:



Utilization Rate:

For a communication link, the rate is defined as the ratio of the number of packets that are not empty passing through it to the number of clock cycles.

For a PE, the rate is defined as the ratio of the number of times a process is fired to the number of clock cycles.

The utilization rate for 2 PE with 6 communication links is as shown in the appendix.

Question:

If we are going to simulate for much more clock cycles, how would you expect the rates to change? Why?

Answer:

If we simulate for more number of clock cycles, then the utilization rate increases. This is because as the number of cycles increases, the packets generated at PE(0,0) and routed to PE(1,0). And hence more packets are produced at PE(1,0) and sent back to PE(0,0).

Queue Size:

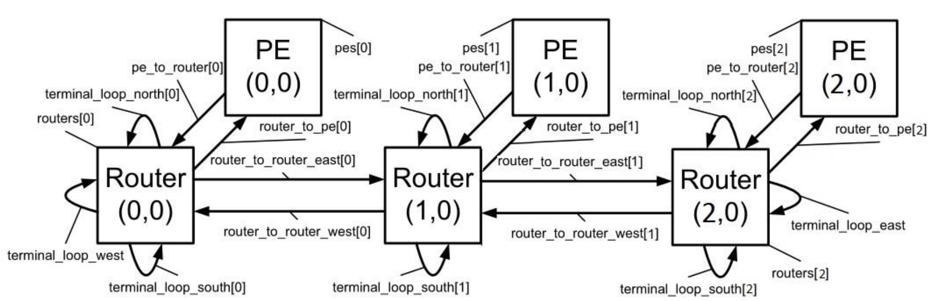
Question:

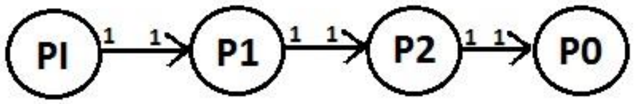
If we are going to simulate for much more clock cycles, how would you expect the sizes to change? Why?

Answer:

The maximum queue size remains the same even if we simulate for more number of clock cycles.

NoC Architecture with 3 PE and 3 Routers:





The Utilization rates and Queue sizes of the PEs and Routers of the NoC architecture with 3 PEs and 3 Routers are as shown in appendix below.

References:

Ece587-prj01.pdf

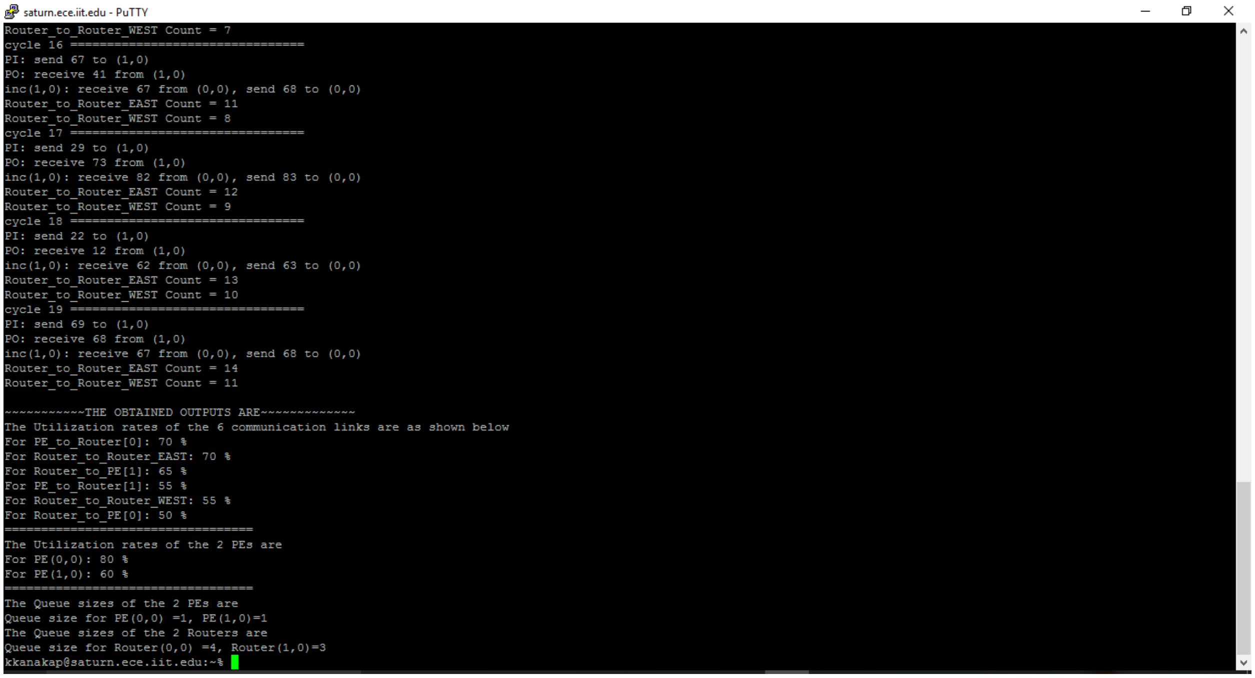
https://en.wikipedia.org/wiki/Network_on_a_chip

APPENDIX:

NoC with 2 PEs:

saturn.ece.iit.edu - PuTTY

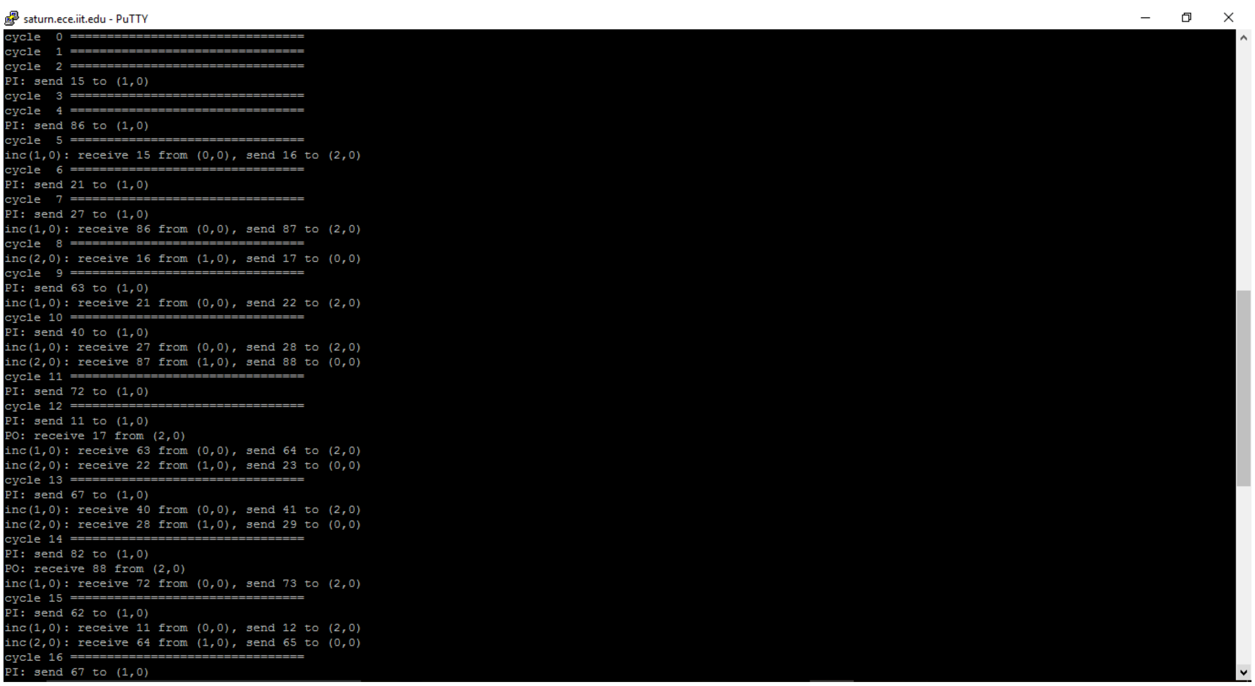
```
             SystemC 2.2.0 --- Aug  9 2013 22:53:09
        Copyright (c) 1996-2006 by all Contributors
                    ALL RIGHTS RESERVED
cycle  0 ================================
cycle  1 ================================
cycle  2 ================================
PI: send 15 to (1,0)
cycle  3 ================================
Router_to_Router_EAST Count = 1
cycle  4 ================================
PI: send 86 to (1,0)
cycle  5 ================================
inc(1,0): receive 15 from (0,0), send 16 to (0,0)
Router_to_Router_EAST Count = 2
cycle  6 ================================
PI: send 21 to (1,0)
Router_to_Router_WEST Count = 1
cycle  7 ================================
PI: send 27 to (1,0)
inc(1,0): receive 86 from (0,0), send 87 to (0,0)
Router_to_Router_EAST Count = 3
cycle  8 ================================
PO: receive 16 from (1,0)
Router_to_Router_EAST Count = 4
Router_to_Router_WEST Count = 2
cycle  9 ================================
PI: send 63 to (1,0)
inc(1,0): receive 21 from (0,0), send 22 to (0,0)
cycle 10 ================================
PI: send 40 to (1,0)
PO: receive 87 from (1,0)
inc(1,0): receive 27 from (0,0), send 28 to (0,0)
Router_to_Router_EAST Count = 5
Router_to_Router_WEST Count = 3
cycle 11 ================================
PI: send 72 to (1,0)
Router_to_Router_EAST Count = 6
Router_to_Router_WEST Count = 4
cycle 12 ================================
PI: send 11 to (1,0)
PO: receive 22 from (1,0)
inc(1,0): receive 63 from (0,0), send 64 to (0,0)
Router_to_Router_EAST Count = 7
cycle 13 ================================
```



saturn.ece.iit.edu - PuTTY

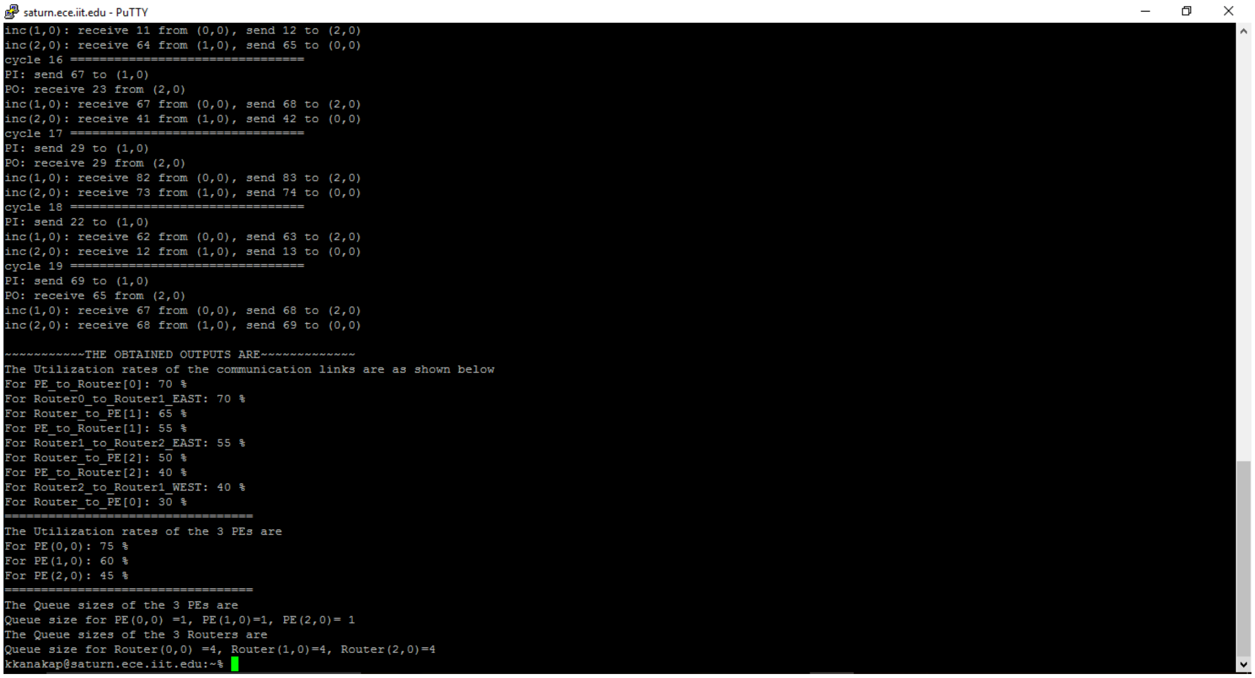
```
Router_to_Router_WEST Count = 7
cycle 16 ================================
PI: send 67 to (1,0)
PO: receive 41 from (1,0)
inc(1,0): receive 67 from (0,0), send 68 to (0,0)
Router_to_Router_EAST Count = 11
Router_to_Router_WEST Count = 8
cycle 17 ================================
PI: send 29 to (1,0)
PO: receive 73 from (1,0)
inc(1,0): receive 82 from (0,0), send 83 to (0,0)
Router_to_Router_EAST Count = 12
Router_to_Router_WEST Count = 9
cycle 18 ================================
PI: send 22 to (1,0)
PO: receive 12 from (1,0)
inc(1,0): receive 62 from (0,0), send 63 to (0,0)
Router_to_Router_EAST Count = 13
Router_to_Router_WEST Count = 10
cycle 19 ================================
PI: send 69 to (1,0)
PO: receive 68 from (1,0)
inc(1,0): receive 67 from (0,0), send 68 to (0,0)
Router_to_Router_EAST Count = 14
Router_to_Router_WEST Count = 11

~~~~~~~~~~~THE OBTAINED OUTPUTS ARE~~~~~~~~~~~~~
The Utilization rates of the 6 communication links are as shown below
For PE_to_Router[0]: 70 %
For Router_to_Router_EAST: 70 %
For Router_to_PE[1]: 65 %
For PE_to_Router[1]: 55 %
For Router_to_Router_WEST: 55 %
For Router_to_PE[0]: 50 %
==================================
The Utilization rates of the 2 PEs are
For PE(0,0): 80 %
For PE(1,0): 60 %
==================================
The Queue sizes of the 2 PEs are
Queue size for PE(0,0) =1, PE(1,0)=1
The Queue sizes of the 2 Routers are
Queue size for Router(0,0) =4, Router(1,0)=3
kkanakap@saturn.ece.iit.edu:~%
```

NoC with 3 PEs and 3 Routers:



saturn.ece.iit.edu - PuTTY

```
cycle  0 ================================
cycle  1 ================================
cycle  2 ================================
PI: send 15 to (1,0)
cycle  3 ================================
cycle  4 ================================
PI: send 86 to (1,0)
cycle  5 ================================
inc(1,0): receive 15 from (0,0), send 16 to (2,0)
cycle  6 ================================
PI: send 21 to (1,0)
cycle  7 ================================
PI: send 27 to (1,0)
inc(1,0): receive 86 from (0,0), send 87 to (2,0)
cycle  8 ================================
inc(2,0): receive 16 from (1,0), send 17 to (0,0)
cycle  9 ================================
PI: send 63 to (1,0)
inc(1,0): receive 21 from (0,0), send 22 to (2,0)
cycle 10 ================================
PI: send 40 to (1,0)
inc(1,0): receive 27 from (0,0), send 28 to (2,0)
inc(2,0): receive 87 from (1,0), send 88 to (0,0)
cycle 11 ================================
PI: send 72 to (1,0)
cycle 12 ================================
PI: send 11 to (1,0)
PO: receive 17 from (2,0)
inc(1,0): receive 63 from (0,0), send 64 to (2,0)
inc(2,0): receive 22 from (1,0), send 23 to (0,0)
cycle 13 ================================
PI: send 67 to (1,0)
inc(1,0): receive 40 from (0,0), send 41 to (2,0)
inc(2,0): receive 28 from (1,0), send 29 to (0,0)
cycle 14 ================================
PI: send 82 to (1,0)
PO: receive 88 from (2,0)
inc(1,0): receive 72 from (0,0), send 73 to (2,0)
cycle 15 ================================
PI: send 62 to (1,0)
inc(1,0): receive 11 from (0,0), send 12 to (2,0)
inc(2,0): receive 64 from (1,0), send 65 to (0,0)
cycle 16 ================================
PI: send 67 to (1,0)
```




```
inc(1,0): receive 11 from (0,0), send 12 to (2,0)
inc(2,0): receive 64 from (1,0), send 65 to (0,0)
cycle 16 ================================
PI: send 67 to (1,0)
PO: receive 23 from (2,0)
inc(1,0): receive 67 from (0,0), send 68 to (2,0)
inc(2,0): receive 41 from (1,0), send 42 to (0,0)
cycle 17 ================================
PI: send 29 to (1,0)
PO: receive 29 from (2,0)
inc(1,0): receive 82 from (0,0), send 83 to (2,0)
inc(2,0): receive 73 from (1,0), send 74 to (0,0)
cycle 18 ================================
PI: send 22 to (1,0)
inc(1,0): receive 62 from (0,0), send 63 to (2,0)
inc(2,0): receive 12 from (1,0), send 13 to (0,0)
cycle 19 ================================
PI: send 69 to (1,0)
PO: receive 65 from (2,0)
inc(1,0): receive 67 from (0,0), send 68 to (2,0)
inc(2,0): receive 68 from (1,0), send 69 to (0,0)

~~~~~~~~~~~THE OBTAINED OUTPUTS ARE~~~~~~~~~~~~~
The Utilization rates of the communication links are as shown below
For PE_to_Router[0]: 70 %
For Router0_to_Router1_EAST: 70 %
For Router_to_PE[1]: 65 %
For PE_to_Router[1]: 55 %
For Router1_to_Router2_EAST: 55 %
For Router_to_PE[2]: 50 %
For PE_to_Router[2]: 40 %
For Router2_to_Router1_WEST: 40 %
For Router_to_PE[0]: 30 %
===============================
The Utilization rates of the 3 PEs are
For PE(0,0): 75 %
For PE(1,0): 60 %
For PE(2,0): 45 %
===============================
The Queue sizes of the 3 PEs are
Queue size for PE(0,0) =1, PE(1,0)=1, PE(2,0)= 1
The Queue sizes of the 3 Routers are
Queue size for Router(0,0) =4, Router(1,0)=4, Router(2,0)=4
kkanakap@saturn.ece.iit.edu:~$
```