NoC-based Systems Design Technique

Under the supervision and guidance of :

Prof. Kanchan Manna

Prepared by :

- **Rishit Patel** 2018A7PS0189G
  (f20180189@goa.bits-pilani.ac.in)

Acknowledgement

I would like to thank Prof. Kanchan Manna for his constant guidance and teaching support throughout the project which made it possible for me to understand the relevant works published on Photonic NoCs and also implement the ILP formulation.

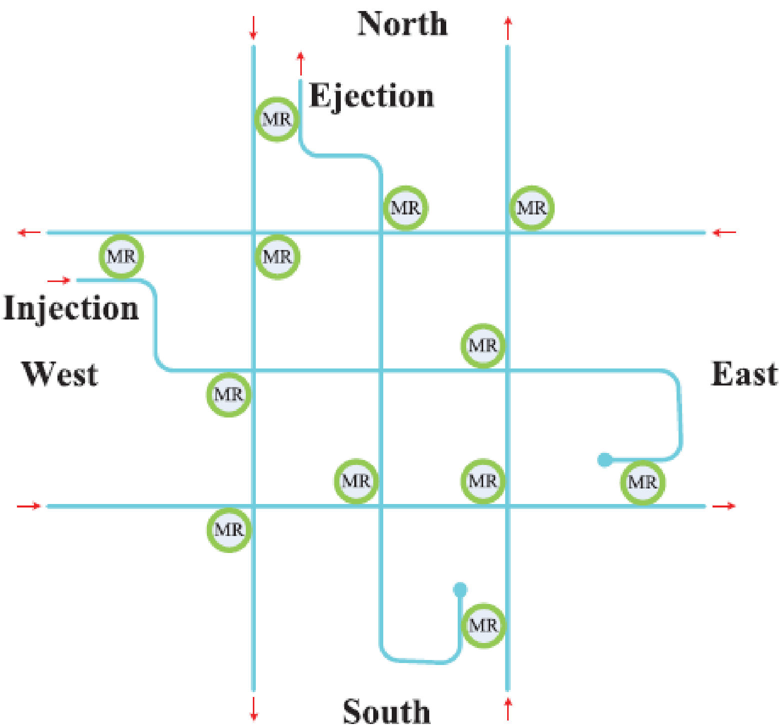
Table of contents

Introduction

Photonic routers serve as one of the key components of PNoCs, and they are considered important when defining performance and cost parameters such as optical loss, power dissipation, and throughput. Based on the two basic switching elements such as waveguides and microring resonators, the Crux optical router was developed.

Crux router uses 12 microresonators to implement the strictly non-blocking 5*5 routing function required by the dimension-order routing algorithm. Crux includes a switching fabric and control unit. The routers have five bidirectional ports such as Injection/Ejection, East, South, West, and North. The Injection/Ejection port is connected to local processor core through an Optical/Electronic(O/E) interface.These five ports are aligned to their intended directions, and the input and output of each port are also properly aligned to ensure that no extra crossing or waveguide bending is required when multiple routers are connected to form Mesh-based Photonic NoCs.

Crux takes the advantages of the parallel switching element to minimize optical loss. Unlike the optimized crossbar in which optical signals travel in one dimension, Crux can passively route them and does not require to power on any microresonator. Only when optical signals use the Injection/Ejection port or make a turn from one dimension to the other, routers need to power on one microresonator. In Crux, the maximum number of waveguide crossings between the ports is five. Moreover, regardless of the network size, the Mesh-based PNoCs based on Crux only need to power on at most three microresonators to inject, turn, and eject an optical signal between any pair of processor cores, Crux is non-blocking.



<u>5x5 Optical Crux Router</u>

Problem Overview

- A Communication Graph(CG) is a graph $G(V, E)$, where each vertex $v_i \in V$ represents a task in the application and each edge $e_{i,j} \in E$ represents a dependency between v_i and v_j.

- A Topology Graph(TG) is a graph $M(T, L)$, where each node $t_i \in T$ denotes the router of a tile in the topology and each edge denotes a physical link $l_{i,j} \in L$ between t_i and t_j .

- **Problem Statement :-** Given a Communication Graph(CG) and a 2-D mesh Topology Graph(TG), find an optimal mapping of each tasks in the CG onto tiles of the TG so that the total number of Microring Resonators(MRs) to be turned on in Optical Crux Routers for the application is minimum, under XY routing mechanism.

- **Solution :-** Formulation of a corresponding ILP problem and using the IBM ILOG CPLEX Optimization Studio Tool to get the exact solutions.

Input parameters for ILP

- *tasks* : Number of tasks in the CG
- *edges* : Number of edges in the CG
- *xdim* : X dimension of 2D Mesh
- *ydim* : Y dimension of 2D Mesh
- *edge_tasks* : Edges between tasks in the CG

Decision Variables For ILP

- *mapping[t][i][j]* = 1, if task t is mapped to the tile (i , j)

  0, otherwise

- *x_dist[i][j][a]* = 1, if distance in x dimension between tasks i and j is equal to a ;

  0, otherwise

- *y_dist[i][j][b]* = 1, if distance in y dimension between tasks i and j is equal to b ;

  0, otherwise

- *x[i]* = The X coordinate of tile to which the task a of ith edge(a,b) is mapped

- *y[i]* = The Y coordinate of tile to which the task a of ith edge(a,b) is mapped

- *x1[i]* = The X coordinate of tile to which the task b of ith edge(a,b) is mapped

- *y1[i]* = The Y coordinate of tile to which the task b of ith edge(a,b) is mapped

Constraints For ILP

- For each task t, summation of mapping[t] [i] [j] over all tiles (i,j) should be equal to 1.
  - Ensures that each task is mapped onto one tile
- For each tile (i,j), summation of mapping[t] [i] [j] over all tasks t should be less than or equal to 1.
  - Ensures that each tile has been mapped with at most one task
- For the k^{th} edge (a,b) , mapping[a] [x[k]] [y[k]] = 1 **and** mapping[b] [x1[k]] [y1[k]] = 1
  - Ensures that decision variables - x, y, x1, y1 are consistent with the mapping decision variable

- For the k^{th} edge (a,b) , x_dist[a] [b] [|x[k]-x1[k]|] = 1
    - Ensures that x_dist shows the correct distance along x dimension between all edge tasks
- For the k^{th} edge (a,b), summation of x_dist[a] [b] [d] over all distance values d should be less than or equal to 1.
    - Ensures that x_dist shows only one distance value along x dimension between all edge tasks
- For the k^{th} edge (a,b) , y_dist[a] [b] [|x[k]-x1[k]|] = 1
    - Ensures that y_dist shows the correct distance along y dimension between all edge tasks
- For the k^{th} edge (a,b), summation of y_dist[a] [b] [d] over all distance values d should be less than or equal to 1.
    - Ensures that y_dist shows only one distance value along y dimension between all edge tasks

Objective Function For ILP

- For each task edge $e_{i,j}$:

  Number of MRs to be turned on = **2** , if tasks i and j are mapped along the same dimension

  Number of MRs to be turned on = **3** , otherwise

- The objective function to be minimized is :

  $\sum$ [For k^{th} task edge $e_{i,j}$] :

  [(3 - x_dist[i][j][0] - y_dist[i][j][0]) **{i}**

  +

  (|x[k]-x1[k]|+|y[k]-y1[k]|) / (xdim*ydim)] **{ii}**

- Part **{i}** of the objective function denotes the number of MRs that need to be turned on for each task edge . Its value is either 2 or 3 for each task edge.

- Part **{ii}** of the objective function ensures that once the minimization of MRs is achieved, the summation of Manhattan Distance (minimum number of hops) between two tasks is also minimum. Its value is less than 1 for each task edge.

Effect of Routing Algorithm on the ILP formulation

The objective function of the above ILP minimization is valid for the XY routing algorithm for which there exists only one possible path for each source-destination pair(deterministic in nature). However, for adaptive routing algorithms (path is determined based on the congestion or load in the next router) like West-first, North-Last and Negative-First for which there can exist multiple possible paths between a source-destination pair, the objective function of the above ILP minimization might not remain valid.

Our ILP formulation assumes that if the tasks are mapped along different dimensions then the packets would take exactly one turn to change the dimension and hence the number of MRs to be turned on would increase by one as compared to when the tasks are mapped along the same dimension. However, for adaptive routing algorithms the packets might take more than one turn if the tasks are mapped along different dimensions depending upon the presence of blocked channels and load in adjacent routers. Hence, under adaptive routing mechanisms, the number of MRs to be turned on and the hop count(or Manhattan distance) might actually be greater than that stated in the objective function.

However, the ILP objective function can remain valid for routing algorithms like West-first, North-last and Negative-first if certain modifications were to be made to these mechanisms. For example, following is the pseudocode for West-First algorithm:

Algorithm 1: West-first [32]

```
if at destination then
  arrive
else if destination in same column then
    if destination is North then
        go North
    else
        go South
   end if
else if destination in same row then
  if destination is East then
     go East
  else
     go West
  end if
else if destination is West then
  go West
else if destination is North then
  go North or East
else if destination is south then
  go South or East
end if
```

It can be seen that the algorithm is deterministic if the destination is in the west but adaptive if the destination is in the east. However, we can modify the algorithm such that, if the destination is in north-east or south-east ,then the packet would always travel along the Y-dimension (north or south) first followed by travelling to east at the end. If such modification is applied to West-First algorithm, then the ILP formulation would take into account the correct Manhattan distance and number of MRs to be turned on, since if the tasks are mapped along different dimensions then the packets would take exactly one turn to change the dimension just like the deterministic XY routing mechanism.

****