****

This Read Me file is for the Virtual Channel Simulators developed in MPLab, IIT Kharagpur

This covers the simulators – SIM_DY, SIM_DY_ANT and SIM_ACO.

IMPORTANT NOTES:

- SIM_DY_ANT simulator should be used only when results of Antnet algorithm (SIM_ACO simulator) have to be compared with other routing algorithms (Table, XY, OE-R, OE-NOP). It can handle ant packets as well as data packets.
- For all other cases, SIM_DY simulator suffices.
- Antnet algorithm has been implemented in the simulator SIM_ACO.

****

Before running the simulator, the input(s) to the simulator should be prepared.

INPUT FORMAT:

There are three possible inputs to the simulator:

1. Router Connection File.
2. Core Communication File.
3. Command Line Arguments.

Router Connection File:

In this file, the user mentions the connections among the routers i.e. defines topology of the network. Some of the important points regarding this file are:

a. To begin mentioning the connections of a router – host router (say X), "router" keyword should be used i.e *router X*. The routers should be mentioned serially i.e. in ascending numeric order without any discontinuity.

b. If the user wants to use routing techniques other than table-based, he must provide addresses of the routers. The addresses should be prefixed by "*a 0x*". The row and column numbers should be mentioned in hexadecimal format. If router *X* is at location (10, 3) then its address will be *a 0xA3*. **Neither of the row or column number can be greater than 15 (F in hex)**.

c. After this, the connected routers should be mentioned - each prefixed by *"r "* i.e. *r Y*. The connections can be mentioned in any order in OPT format (table-based routing only). But if the user wants to use other routing techniques, he should mention the connections strictly in the following order of the links - EAST, WEST, NORTH and SOUTH.

d. If there are no routers connected to a link, then ‘0’ (zero) should be written without any prefix. This creates a link but there is no router attached to it. Its input is invalid flit.

e. Mentioning a connection implies that the host router’s link input is supplied by the attached router. Note that this establishes unidirectional connection only. Unless the attached router has a connection mentioning this router, the connection is unidirectional.

f. All the cores connected to a router are mentioned one-by-one each prefixed by the keyword “*core* ”. *Core 0 (zero)* means a router has no core connected to it.

g. Each router can have a different buffer size. It is mentioned after the keyword “*buffer*”. But within a router, all virtual channels will have an associated buffer of same size. The maximum buffer size allowed is 8 flits.

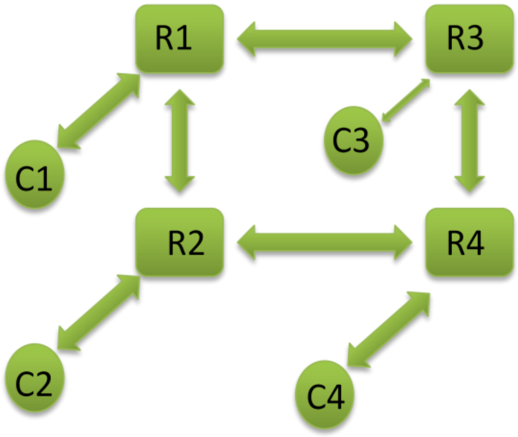
h. The routers and cores can be numbered by the user in any fashion. Only restriction is that the numbering should begin with ‘1’ and should be continuous. Besides, the user can mention addresses too, if required, as mentioned above.

There are two formats for mentioning router connection – Unoptimized format (UNOPT) and Optimized format (OPT).

UNOPT is used when the user wants to use XY, OE or OE-NOP routing schemes in Mesh topology. Every router will have four links and one core connected to it.

OPT can be used for all topologies only when the routing scheme is table-based. In it link numbering is done dynamically.

Let us take an example of 2*2 Mesh Topology:



UNOPT input format for above NoC:

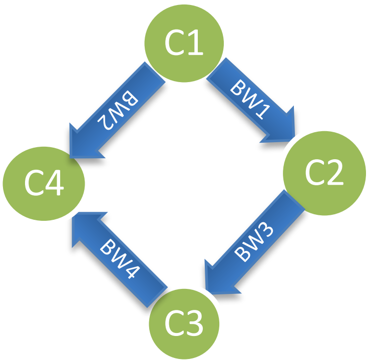
router 1	router 2	router 3	router 4
a 0x00	a 0x10	a 0x01	a 0x11
r 3	r 4	0	0
0	0	r 1	r 2
0	r 1	0	r 3
r 2	0	r 4	0
core 1	core 2	core 3	core 4
buffer 6	buffer 6	buffer 6	buffer 6

`router 1`	// Connections of this router (Host router).
`a 0x00`	// Address of host router (0x<row_no><col_no>). Nos. must be in hex.
`r 3`	// Neighbour (EAST). All these conn. imply input conn.
`0`	// Open link (WEST). Supplied with invalid flit.
`0`	// Open Link (NORTH). Supplied with invalid flit.
`r 2`	// Neighbour (SOUTH).
`core 1`	// Core Connected.
`buffer 6`	// Size of buffer (must be <= 8).

OPT input format:

router 1	router 2	router 3	router 4
r 2 // link 1	r 4	r 1	r 2
r 3 // link 2	r 1	r 4	r 3
core 1	core 2	core 3	core 4
buffer 6	buffer 6	buffer 6	buffer 6

Core Communication File:



```
4
C1        2        BW1        BW2        C2        C4
C2        1        BW3        C3
C3        1        BW4        C4
C4        0
```

Note: Bandwidths must be in bits per second. It is bidirectional BW.

*** **

COMPILATION:

To compile the simulator, use the following command:

```
g++ -I. -I$SYSTEMC_HOME/include -L. -L$SYSTEMC_HOME/lib-linux
-o sim tableTopMain.cpp -lsystemc –lm
```

where **`$SYSTEMC_HOME`** is the path of SystemC installation directory,

`sim` is the executable name and **`tableTopMain.cpp`** is the main file of the simulator.

****

EXECUTION:

To execute the simulators (SIM_DY and SIM_DY_ANT only), following command should be used:

```
./sim –r <no_of_routers> –l <max_no_of_links> -c <max_no_of_cores> –
sim <simulation_time> -sat <saturation_time> -con <conn_filename> -
folder <folder_name> -routing <option> -tab <routing_table_file> -
traffic <core_graph> -inj <inj_rate> -trace
```

To execute the simulator SIM_ACO, following command should be used:

(*Some options are not available for this simulator e.g. routing algorithm is fixed to be OE-Antnet and right now, it works for Mesh topology only.)

```
./sim –r <no_of_routers> –sim <simulation_time> -sat
<saturation_time> -con <conn_filename> -folder <folder_name> -
traffic <core_graph> -inj <inj_rate>
```

Each argument is preceded by an identifier with name starting with '-'. The arguments have been discussed below one by one:

a. **Number of routers:-** This is limited to a maximum value of 256 because of use of 8-bit addressing. Its default value is 32.
b. **Maximum number of global links:-** Any router can have a maximum of this many number of global links (router-to-router links) or routers attached to it. But some routers (generally, corner and edge routers) may have lesser or no links at all

(isolated router). This minimizes the link power consumption as there are no extra links in the system. Its default value is 4.

c. **Maximum number of cores/local links:-** Any router can have a maximum of this many number of cores connected to it. Some routers may have lesser or even no cores connected to them. Its default value is 1.

d. **Simulation Time:-** This is the time for which simulation is run. Its default value is 20,000 cycles.

e. **Saturation Time:-** This is the time after which network statistics are collected. In this time, the network stabilises. Its default value is 1,000 cycles.

f. **Router Connection File:-** It is a mandatory input. It mentions the connections among the routers. It must be in the same folder as the executable.

g. **Routing Options:-** At present, there are four options (each identified by a number assigned to it) :

    i. **Table-Based Routing (0):-** In it, user has a choice of either giving his own routing table in a text file or let the simulator build it for him (default choice). The format of the routing file is the same as the routing table. It is a matrix of order (No. of routers)*(No. of cores). Every location will contain a link number. If at location (x, y), z is written it implies that for router x to send data to core y, output link z of router x will be used. **The link numbering must begin with '1' for any router.**

    ii. **XY Routing (1):-** The user must supply routers' addresses in (row, col) format in router connection file to use this routing scheme.

    iii. **OE Routing (2):-** This needs routers' addresses too.

    iv. **OE-NoP Routing (3):-** This needs routers' addresses too.

h. **Traffic:-** There are two options for generating traffic – user-generated and simulator-generated. The user himself can generate traffic and put it in the "*input*" folder inside the result folder or the main folder (containing the executable). **The "input" folder must be created by the user in this case either in result folder or main folder**. For the simulator to generate the traffic, the user must supply core communication file in the prescribed format. The file should be in the same folder as the executable. The user need not do anything else. The user should be careful to select only one of the above choices otherwise simulator-generated traffic will overwrite user-traffic.

i. **Injection Rate:-** Giving injection rate implies user wants the simulator to generate synthetic traffic. Its unit should be in Packets/Cycle/IP. If the user gives this value, he should not give core communication file (for AS-traffic, may result in erroneous traffic) or create any folder (unnecessary). There is no concept of locality for the synthetic traffic generated here.

j. **Result Folder:-** In this folder, all results are stored. If it is not given, then main folder (containing the executable) is used. **Folder name must be followed by a slash '/'.**

Except for the router connection file, all other parameters are optional. If the user doesn't provide some parameter(s) then the corresponding default value(s) (as mentioned in the "*configParam.h*" file) is (are) taken.