

Synthesis Configurable Virtual Channel Router

Run Instructions

Team:

Member 1:

Name: Varun Saxena

GT id: 903562211

Member 2:

Name: Venkata Hanuma Sandilya Balemarthy

GT id:903482005

Source code:

1. Github repository: https://github.com/varunsax12/Virtual_Channel_Router
2. Tar Ball: virtual_channel_router_files.tar.gz

Run Environment:

Server: ece-linlabsrv01.ece.gatech.edu

Tool Setup: (specific to server)

Run the following commands:

1. Synopsys VCS
 - a. export VCS_HOME=/tools/software/synopsys/vcs/R-2020.12-SP2/
 - b. export PATH=/tools/software/synopsys/vcs/R-2020.12-SP2:/tools/software/synopsys/vcs/R-2020.12-SP2//bin:\$PATH

Directory Structure:

Top folder: Virtual_Channel_Router

1. ./rtl: Contains all the RTL source code
 - a. ./libs: Contains all generic library modules created to support router
 - b. ./router_modules: Contains all router stages and custom router modules
 - c. router_top.sv: Top router module
2. ./synthesis: Folder for running synthesis
 - a. rtl.tcl: TCL file to run the synthesis
3. /testbench: Contains all the testbenches created to test RTL
 - a. ./libs: Testbenches for a libs modules
 - b. ./router_modules: Testbenches for all router specific modules
 - c. tb_router_top.sv: Testbench for top router module
4. Makefile
5. VR_define.vh: Top level define file for changing the router configurations
6. README.md: Github readme file
7. sample_out.log: Sample output from the testbench run of router top module

Steps to run functional verification:

Running router top testbench:

1. Run the tool setup commands mentioned in the "Tool Setup" section
2. Run the following commands:
 - a. make clean
 - b. make

Expected Output:

1. Command line report generated
2. test.vcd: VCD file for visual debug which can be opened using "gtkwave -f test.vcd"

The report contains the status of each router stage at each time stamp. Different stages displayed (demarcated by ***** identifiers):

1. INPUT SIGNALS: Inputs and outputs to and from the router top connections.
2. VC BUFFER STATUS: Shows current status of each buffer (shows only the top/head of the buffer)
3. BUFFER WRITE: The buffer the flit will be written to for each port
4. VC AVAILABILITY: Shows the available output VCs during the VC allocation stage along with the mask generated.
5. VC ALLOCATION: Shows output VC allocated for each input VC
6. SA ALLOCATION: Shows output port allocated for each input port
7. BUFFER READ: Shows the buffer which will be read
8. SWITCH TRAVERSAL: Shows outputs signals post switch traversal

The data propagation can be tracked starting "Time = 80" when the inputs are applied.

Running topology testbench:

1. Updated the Makefile. In line 24, edit the line to state:
TESTBENCH = testbench/tb_topology.sv
2. Run the following commands:
 - a. make clean
 - b. make

Expected Output:

1. Command line report generated
2. test.vcd: VCD file for visual debug which can be opened using "gtkwave -f test.vcd"

The report contains the flits injected into the network. Flit format (for 5x5 torus configured into the testbench), 5 MSB bits represent the destination. The LSB 11 bits are randomly added to created tracking IDs for tracking the flit across the report. The report shows:

1. Input flits injected into each router
2. Output flits ejected from each router

Steps to run synthesis:

1. Run: cd ./synthesis
2. Run: /tools/software/cadence/genus/latest/bin/genus -legacy_ui
3. Run: source ./rtl.tcl in the genus prompt

The generated logs and rep can be viewed in the same folder.