Investigation of serial data communication in Multi FPGA network of chips

M. Tech. Stage - I Project Presentation

Saurabh Agrawal

123076007

Guided by: Prof. Sachin Patkar

Presentation Outline

- Objective to be achieved
- Components of the project
  - CONNECT NoC
  - Aurora 8B10B
- Previous work
- Modifications and proposed design
- Result of simulation
- Automatic network partitioning script
- Future work

Thesis Objective

Implementation and performance evaluation of high speed serial data communication links in Multi FPGA network of chips (NoC)

- Xilinx high speed serial IP core, Aurora 8B10B
- Performance evaluation of High Speed serial communication in Multi FPGA network of chips compared to parallel and simple UART based serial communication
- Exploration algorithms and automated partitioning methods of NoC

Components

- Aurora 8B10B v5.3 Xilinx IP Core
  - GTP
  - Clock generation and Clock recovery Module
  - Core include frame generator and frame check
- CONNECT NoC Verilog design
  - Router
  - Routing table

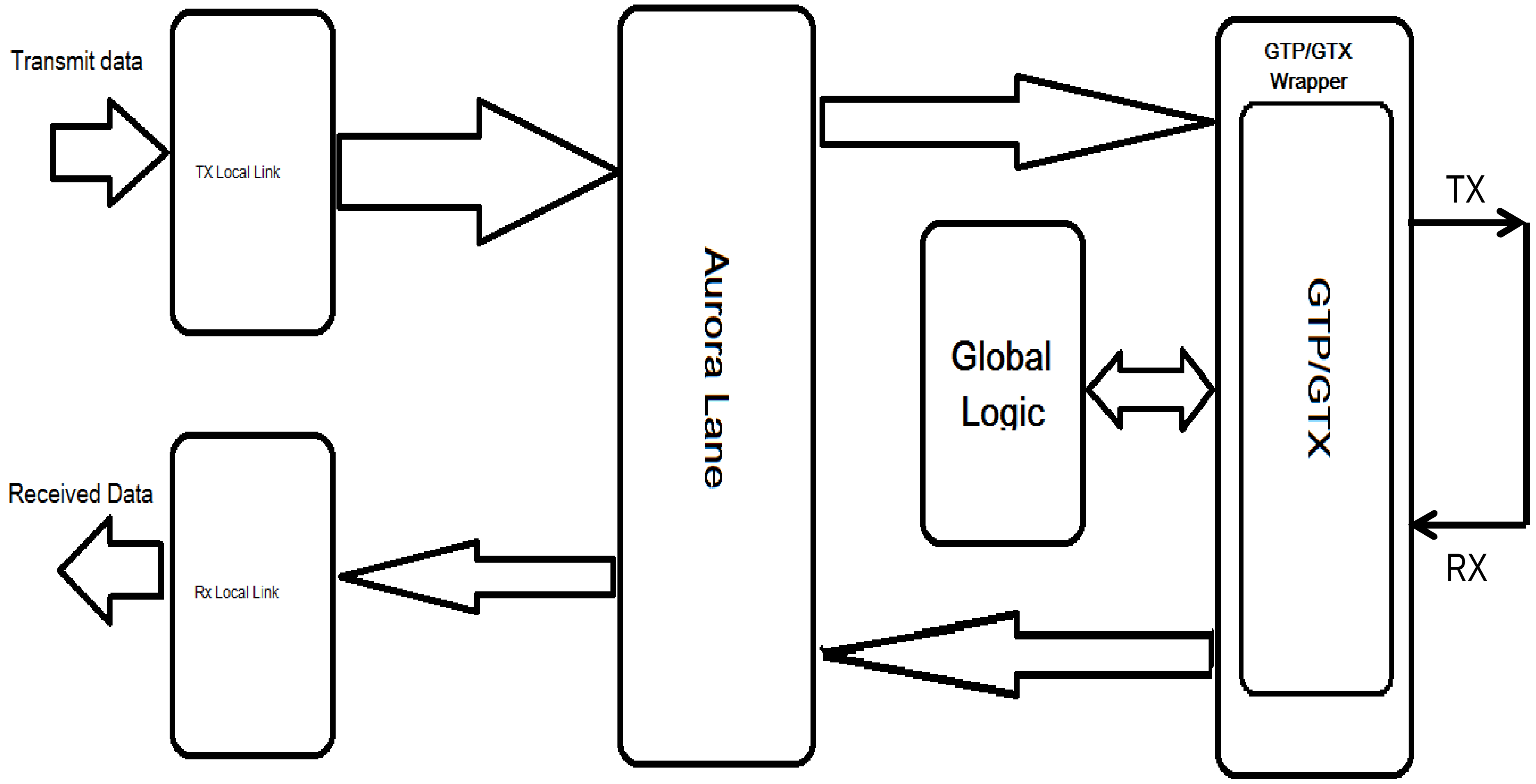
Aurora 8B10B IP Core Advantages

- Aurora Simplex for efficient Multi-gigabit throughput
- Extremely efficient Logic Implementation
- Data transfer upon Initialization
  - Channel initialization – 600-800 clock cycles
- Error checking through 8b/10b coding
- Reset and Re-initialize on catastrophic errors

Aurora Applications

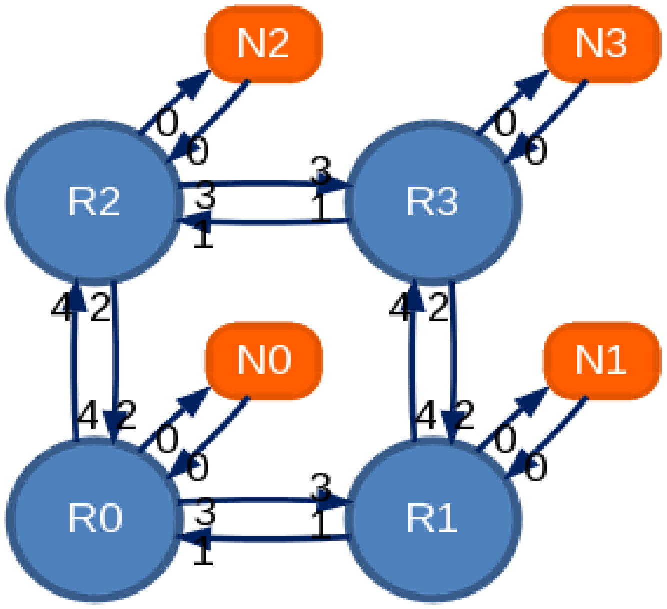
- Chip to Chip
- Board to Board
- ASIC / SoC
- Simplex / duplex Connection

Aurora Working



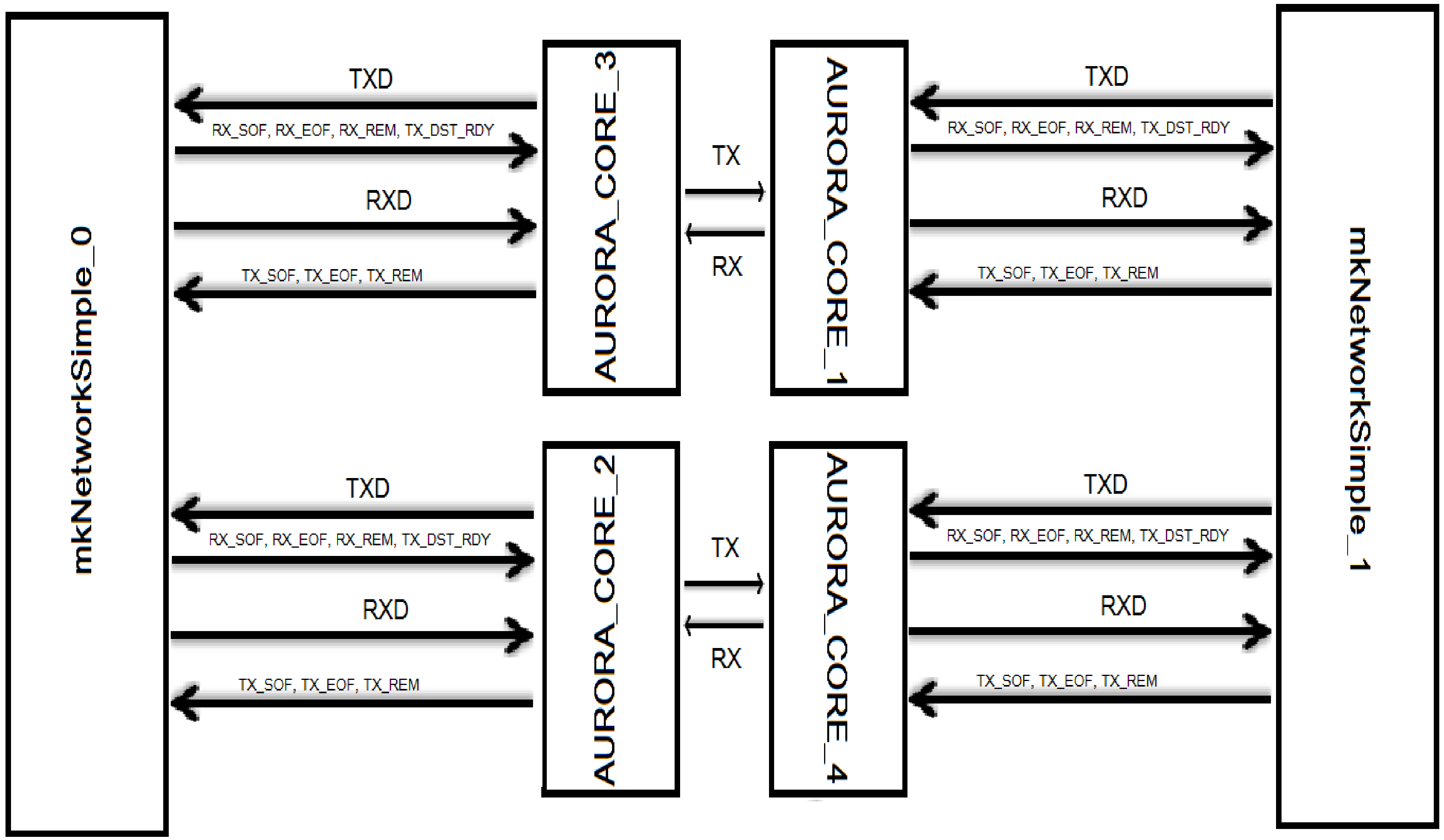
CONNECT 2X2 Mesh Network

Input / Output Flit Format (16 bit)				
1 Bit	1 Bit	3 Bits	1 Bit	10 bits
Valid	Tail	Dest	Virtual channel	Data

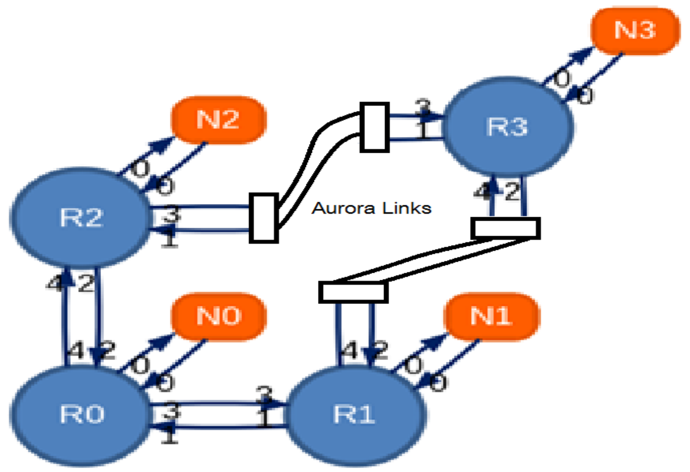


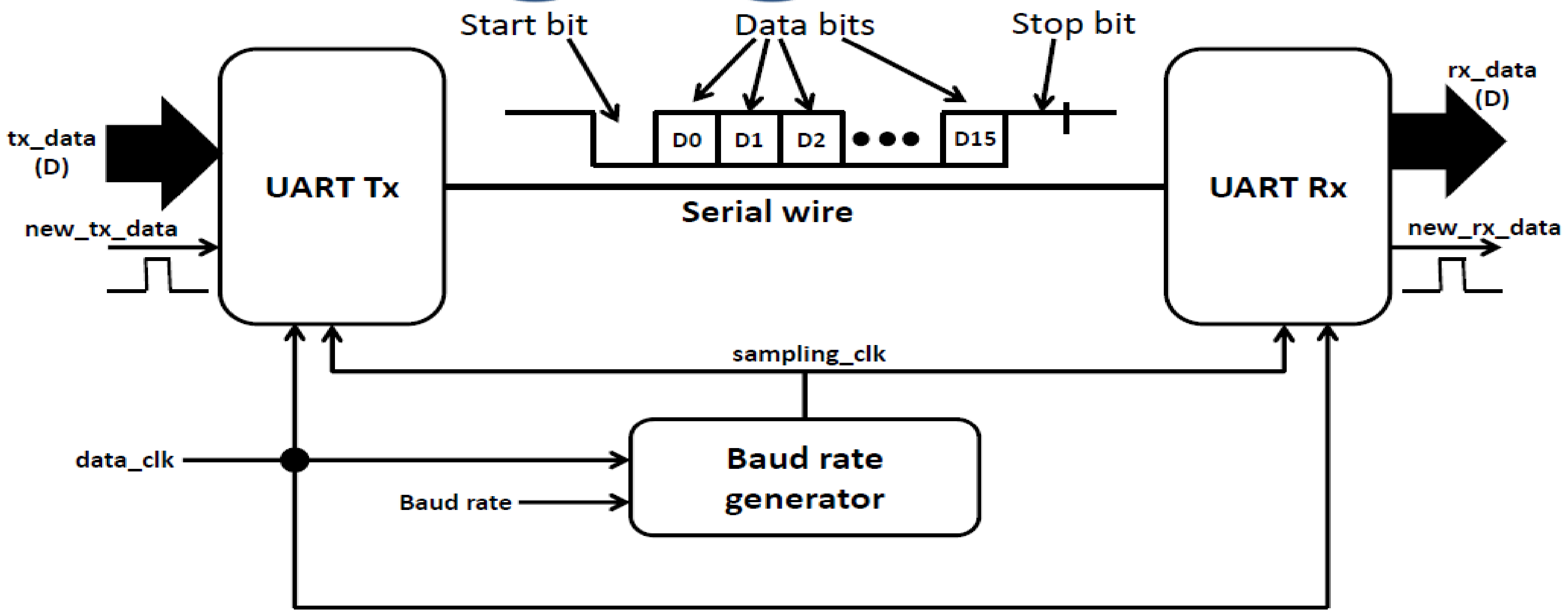
Destination	Router 0	Router 1	Router 2	Router 3
0	Port 0 (000)	Port 1 (001)	Port 2 (010)	Port 2 (010)
1	Port 3 (011)	Port 0 (000)	Port 2 (010)	Port 2 (010)
2	Port 4 (100)	Port 1 (001)	Port 0 (000)	Port 1 (001)
3	Port 4 (100)	Port4 (100)	Port 3 (011)	Port 0 (000)

Proposed Design

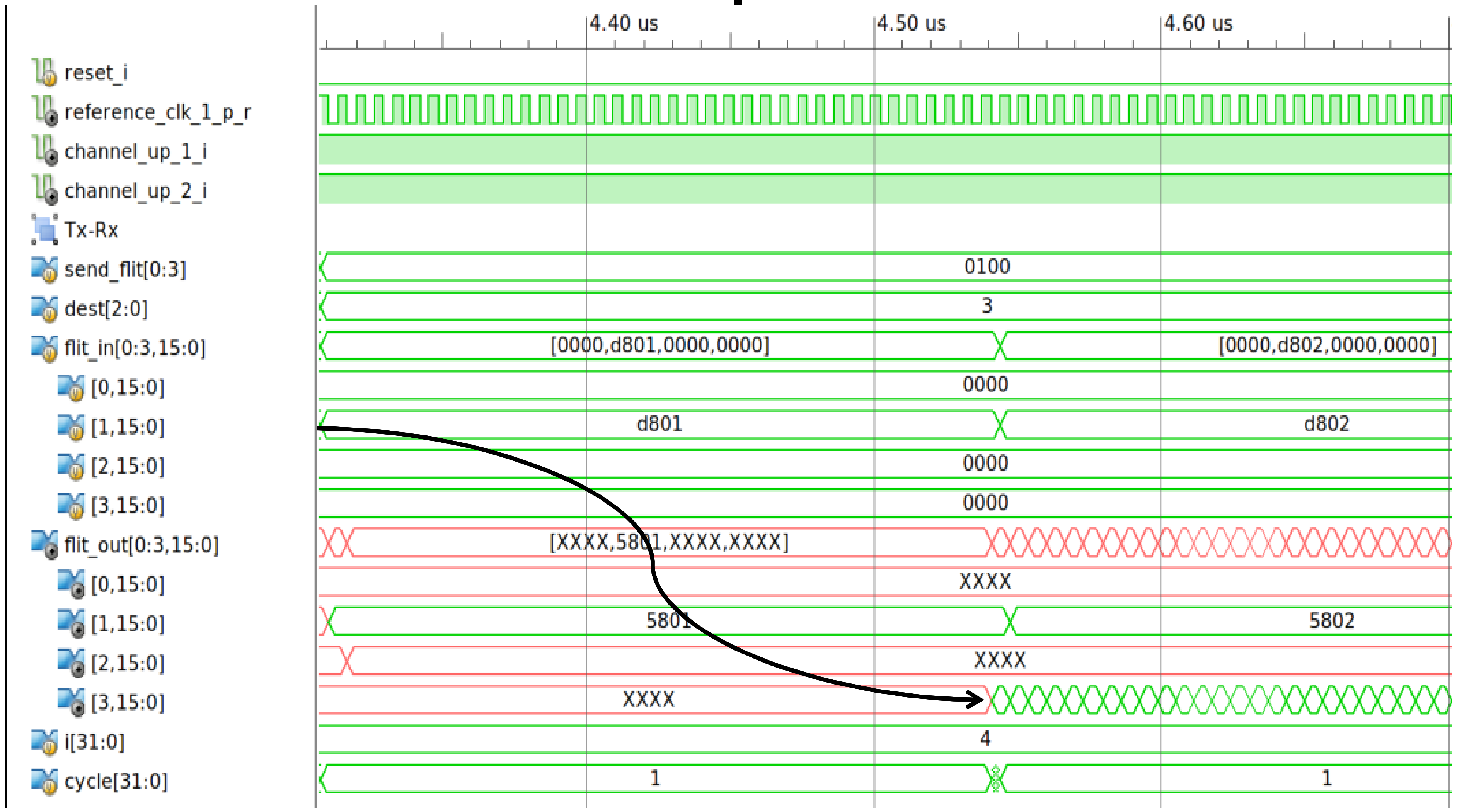


Previous Work





Output


4.40 us
4.50 us
4.60 us
reset_i
reference_clk_1_p_r
channel_up_1_i
channel_up_2_i
Tx-Rx
send_flit[0:3]
dest[2:0]
flit_in[0:3,15:0]
[0,15:0]
[1,15:0]
[2,15:0]
[3,15:0]
flit_out[0:3,15:0]
[0,15:0]
[1,15:0]
[2,15:0]
[3,15:0]
i[31:0]
cycle[31:0]
0100
3
[0000,d801,0000,0000]
[0000,d802,0000,0000]
0000
d801
d802
0000
0000
[XXXX,5801,XXXX,XXXX]
XXXX
5801
5802
XXXX
XXXX
4
1
1

Results

```
@1: Sending flit 001 into send port 0 from Router = 1 to Router = 3
@0: Receiving flit 001 at receive port of router = 3
@37: No. of cycles from source to destination

@1: Sending flit 002 into send port 0 from Router = 1 to Router = 3
@0: Receiving flit 002 at receive port of router = 3
@37: No. of cycles from source to destination

@1: Sending flit 003 into send port 0 from Router = 1 to Router = 3
@0: Receiving flit 003 at receive port of router = 3
@37: No. of cycles from source to destination

@1: Sending flit 004 into send port 0 from Router = 1 to Router = 3
@0: Receiving flit 004 at receive port of router = 3
@37: No. of cycles from source to destination
```

Conclusions

	Un-partitioned NoC	UART Partitioned NoC	Aurora Partitioned NoC
16 Bits	6.4 ns	1.48 msec	23 usec

Resource	Number Used		Approx. Percentage Use	
	UART	AURORA	UART	AURORA
Slice Registers	28,794	29213	9 %	~9%
IOBs	+18	+16	1 %	~1%
Slice LUTs	50,482	50845	33 %	~33 %
Block RAMs	+144	~	34 %	~

Future Work

- Automatic partitioning of any network
- Instantiation of Aurora Links
- Implementation partitioned network for applications such as Boolean Matrix Vector Multiplication (BMVM) on a Virtex V Development platform
- Investigation of algorithm development for application based NoC partitioning

Script Parameters for NoC partitioning

```
     #### Python script Parameters for automatic partitioning ####
import re
routers         = [0, 1, 2, 3] #No. of router in partitioned network
part            = [0, 1, 2]    #No. of router in this partitioned network
part_no         = 0            #Partition part No.
ports_per_router = 3           #No. of ports in each router
flit_data_width = 16           #Data width
vc_bits         = 1
dest_bits       = 2            #Destination width
data_width      = 2
hex_filename    = <Routing_table_<filename.hex>
#This script is supposed to partition the given network and instantiate#
              #Serial Aurora module and interfaces using#
```

Thank You