PageRank

Members: Ashik Poojari(ap4613) and Nimisha Limaye(nsl278)

***Abstract*—The paper describes a PageRank algorithm to sort out top 10 websites out of 64 websites. The main module consists of 4 PageRank modules which each stores 16 websites, 4 responder and 4 requestor modules for each PageRank module, one NOC requestor module to request score from websites outside of each PageRank scope, one NOC responder module for sending the correct score to module which requested. Essentially modules requestor and responder are routers which route the data packet from one PageRank module to another. The sorter module outputs top 10 websites' scores along with its ID.**

I. Section I: Architecture

PageRank algorithm consists of 4 PageRank modules holding 16 websites each, 4 responder and 4 requestor modules for each PageRank module, one NOC requestor module, one NOC responder module and one sorter module. Requestor and responder modules are essentially routers performing routing task between the PageRank modules. The architecture of the algorithm is as shown below.

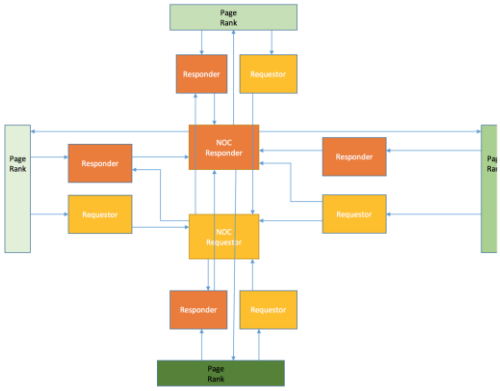


Fig. Architecture of PageRank Algorithm

II. Section – 2: Components of Architecture

A. *PageRank*

There are four PageRank modules each storing scores of 16 websites. The code takes in two local parameters viz. start and finish, to define the starting website and last website available in the individual PageRank module. The module outputs a requestor signal and inputs a responder signal for inter module transfer of website scores. Each PageRank module gets its unique ID during instantiation in high level module. Each PageRank module stores adjacencies and weights of all the 64 websites but has scores of only 16 local websites. The module sends the scores of all its websites as they are updated to the local responder modules, which then forward these updated scores to the modules which request them. Each PageRank module has a local responder and local requestor to store the upcoming requests and updated scores respectively. The NOC requestor and NOC responder ensures transfer between PageRank modules. The architecture of PageRank module is as shown below.

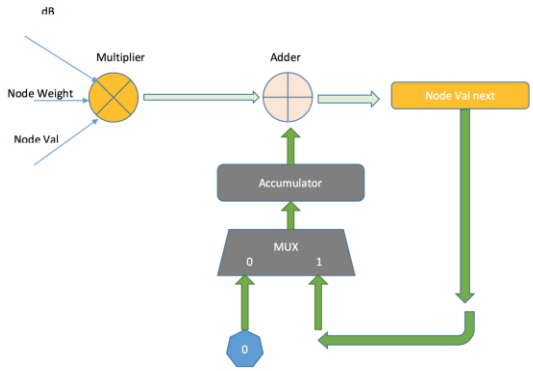


Fig. Architecture of PageRank module

B. *Requestor*

The requestor gets its inputs from PageRank modules in the format as shown below.

Request packet

0	1:2	3:4	5:8
1'b1	Destination port	Source port	Reg Id

Bit 0 is for valid bit. It is made high when module is requesting score of website from another module. Bits 1 and 2 gives address of destination module which has the score of website requested for. Bits 3 and 4 stores the address of source PageRank module who is requesting for score. Bits 5 to 8 stores the local number of website in the destination module, as bits 5 to 8 account for 4 bits which can store values from 0 to 15.

C. *Responder*

The responder gives its output to PageRank modules in the format as shown below.

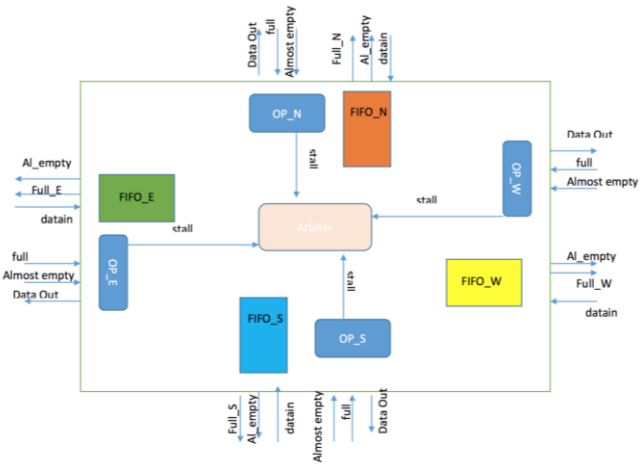
Respond packet

0	1:2	3:4	5:20
1'b1	Destination port	Source port	Score

Bit 0 is for valid bit. It is made high when module is responding the score of website to another module. Bits 1 and 2 gives address of destination module which wants the score of website. Bits 3 and 4 stores the address of source PageRank module who has the score which was requested for. Bits 5 to 20 stores the score of the requested website, as bits 5 to 20 account for 16 bits which is equal to the width of the score of websites.

D. NOC Routers

There are two NOC routers, one for requesting and one for responding. The topology of one NOC router is as shown below.



The NOC router implements Round Robin Arbitration Policy with order as East, West, North and South. East port represents PageRank Module 1 which stores websites 0 to 15, West port represents PageRank Module 2 which stores websites 16 to 31,North port represents PageRank Module 3 which stores websites 31 to 47 and South port represents PageRank Module 4 which stores websites 48 to 63. NOC router consists of FIFO and RAM module to store and route the bits. Each port has one FIFO and one RAM module. We have included a stall signal to report the FIFO to stall when the full signal is high. This way we are not dropping packets during traffic.

E. Sorter

This module is used for outputting top 10 websites with maximum scores along with their IDs. The module takes in scores from all the four PageRank modules and finds out top 10 scores and outputs the score with the IDs. The algorithm used for sorting is Bubble Sort. This algorithm lets us calculate the top 10 scores in just 10 iterations. We are not concerned with ranks of other 54 websites. Hence this algorithm proves efficient speed wise as well as area wise. The technique behind Bubble Sort is as shown below.

Original sequence

No. of iterations	8	7	6	4	3	9	2	5
1	7	6	4	3	8	2	5	9
2	6	4	3	7	2	5	8	9
3	4	3	6	2	5	7	8	9
4	3	4	2	5	6	7	8	9

We can see that to sort top 4 values out of 9 we need only 4 iterations. Therefore to find top 10 values out of 64 we require only 10 iterations. Hence less area and time.

III. DESIGN CHOICES

Total Number of sub-modules	8
Width of Fixed point representation	16bits
Convergence Rule	39 Iterations
NOC Parameters	
-FIFO depth	16bits

IV. EXPERIMENTAL RESULTS

Top 10 Ids	Scores from MATLAB code	Website ID
1	0.024567	11
2	0.023544	35
3	0.019696	56
4	0.019656	1
5	0.019385	61
6	0.019127	47
7	0.019113	17
8	0.019089	4
9	0.019081	15
10	0.019019	49

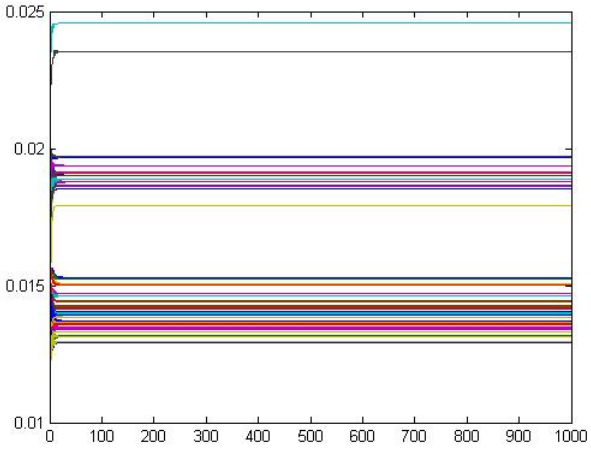


Fig. Graph showing scores of 64 websites against number of iterations.

V. Conclusion

The PageRank algorithm was performed successfully. We were able to display the scores along with IDs of the website without dropping any packet. We were able to display all the individual website scores after 13µs.