Design Of Fast And Scalable Non-Blocking Scheduler For High Capacity Packet Router

Babu. M[#1], Krishnamoorthi. S[#2], Ravindran. G[#3], Rajkumar. S[#4]
Asst.Professor Department Of Ece[#1] Mca[#2,#4] Mba[#3]
Psrr College Of Engineering[#1], Psr Engineering College[#2,#3,#4], Sivakasi

ABSTRACT-As the need for non-blocking internet routers are becoming a need for high speed networks, I have designed an architecture based on high speed networks with the use of a new design of sequential greedy scheduling algorithm (SGS), is a maximal matching algorithm that provides non-blocking in an internet packet routers, when the traffic is policed. In this architecture, packets are split into cells of fixed length and stored at the input ports. The overall function of the system is implemented using Modelsim and output characteristics of the design are done to finalize the proposed design. This project increases the speed of packet transmission.

Key Words- High capacity, non-blocking, packet switching.

1. INTRODUCTION

The fast growth of bandwidth demand on the Internet has led to a need for high capacity packet routers, with large number of ports and high port speeds. Routers with input buffers are the most scalable single-stage routers. SGS algorithm speeds up the transmission of packets without any delay.

2. DESIGN OF THE SCHEDULER

The internal architecture of the input port is given in Figure1. The input port consists of several components: network processor, data memory, linked list memory, queue manager, and output selector. When a packet arrives to the router, its destination IP address is read by network processor (NP) and the router output to which the packet should be sent is determined. The NP also divides packets into smaller fixed length cells and stores them in the appropriate virtual output queue (VOQ).

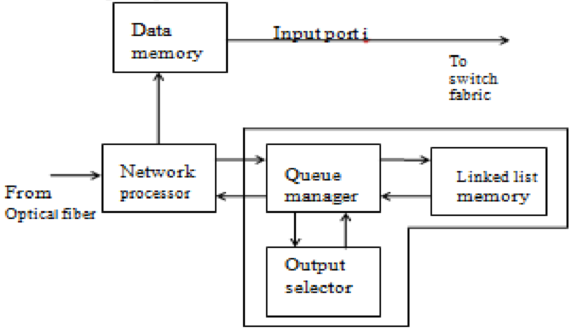


Fig.1 Internal architecture of an input port i

In each input buffer, there are N VOQs that comprise cells bound for particular outputs. Data memory stores the incoming packets/cells until they are scheduled and sent through the switching fabric. Linked list memory stores the data memory addresses of cells in VOQs. Queue manager performs operations on virtual queues. Output selector calculates the schedule for the outstanding cells and stores it in output memory until the cells are read.

2.1. Linked List Memory

Cells in data memory that are bound for the same destination form a VOQ. There are N VOQs, corresponding to N outputs. Linked list memory stores address of cells in different VOQs and addresses of empty locations.

Each location in one virtual queue linked list (VQL) contains the address of the next location in that VQL, or zero (NULL) if no more locations belong to the VQL. Similarly, each location in the empty queue linked list (EQL) contains the address of the next location in EQL. The size of the linked list memory is defined by the number of cells that ought to be stored in the data memory of the router. The data memory should be able to store the number of cells equal to the frame length F. Thus, the linked list memory has F locations.

2.2. Queue Manager

Queue manager performs operations when a cell arrives to the queue manager, when it is scheduled by the output selector, or when it departs the router. The queue manager stores pointers to VQLs. There are three pointers to each VQL: to the beginning of the VQL, the first unscheduled packet in the VQL, and the end of the VQL. Similarly, the EQL is managed with two pointers: to the beginning and the end of the EQL. The pointers are updated each time one of the operations is performed. Multiple operations are carried out in the same time slot, and in general on different VQLs. At the beginning, all memory locations belong to EQL, and each location contains the address of the next location in the memory (except the last that points to NULL). The memory location is removed from the beginning of the EQL to the end of the VQL of some output, when a cell bound for that output arrives to the router. The cell is stored to the address from this memory location.

When input sends a cell to some output, the cell address in the data memory is read from the heading location of the corresponding VQL, and this memory location is added to the end of EQL. When a cell is scheduled, the corresponding VQL is updated. If the pointer to the first unscheduled cell from the same VQL points to the last cell in that VQL, the pointer is set to NULL. Else, the pointer to the first unscheduled cell is set to point to the next element in that VQL. The queue manager was designed in VHDL, and it can be scaled easily for different router sizes.

2.3. Output Selector

The output selector of the associated input schedules a cell by choosing the first available output from the set of outputs for which the given input has cells to send.

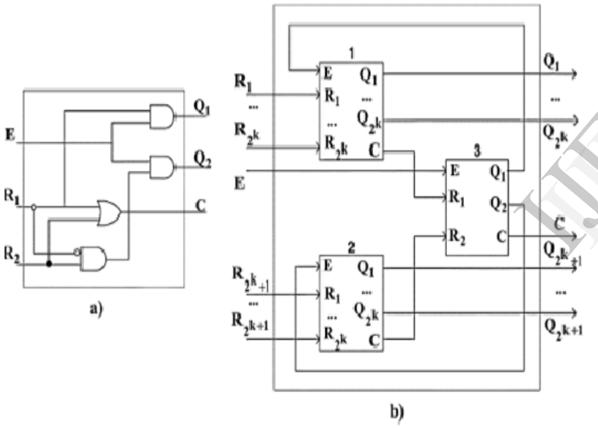


Fig.2 Output selector structure (a) 2 x 2 output selector (b) output selector2 $^{k+1}$ x2^{k+1}

The result of the scheduling process is a vector in which only one bit, which corresponds to the scheduled queue, is set to logical one. The structure of the optimized output selector is shown in Figure2. Bits denoted by R contain information about cells of a particular input, which participate in the contention process. Bit R_j is set to "1" only if there are unscheduled cells for the j th output port, and the j th output port was not selected by previous input ports. As a result of the scheduling process, Q_j is set to "1" when the j th output port is chosen by the given input.

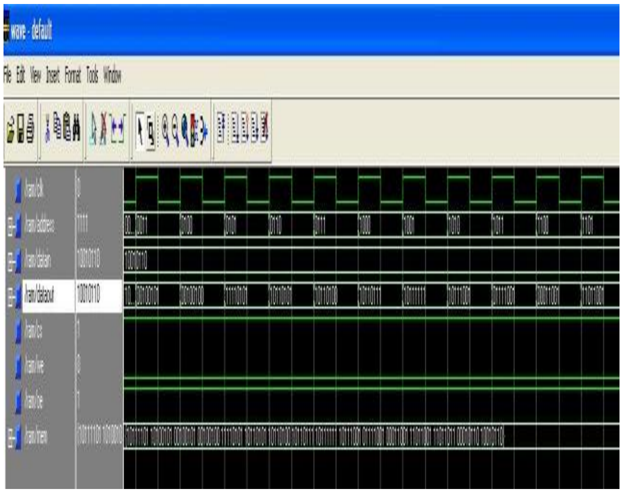
The information about the remaining output ports is forwarded to the next input port in the chain. The output selector is implemented recursively. The output selector for a router with 2^{K+1} output ports can be built recursively by using two output selectors for 2^K ports and one two-port output selector.

3. PACKET SWITCHING

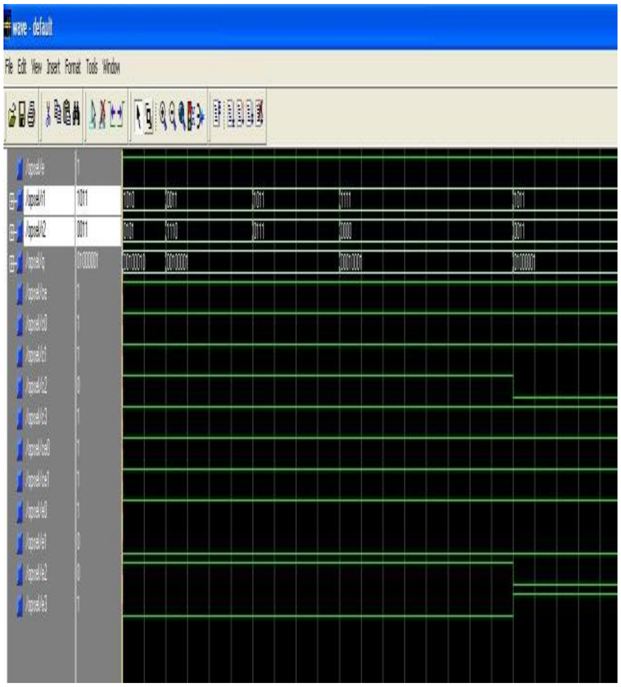
The technique of packet switching was designed especially for the efficient transmission of computer traffic. When using packet switching, all data transmitted by the network user are divided into relatively small fragments known as packets. Each packet is provided with a header containing an address, which is necessary to deliver the packet to the destination. The presence of address in each packet represents one of the most fundamental properties of the packet-switching technique, since each packet may be processed by the switch independently of other packets of the information flow. Besides the header, the packet has another auxiliary field, which is usually located at the end of the packet and therefore is generally known as trailer. The trailer contains the checksum, which allows you to check if the information was corrupted during transmission. Packets are supplied to the network without previously reserving communication links, and at the rate at which source generates them. This rate cannot exceed the bandwidth of the access link.

4. SIMULATION RESULTS

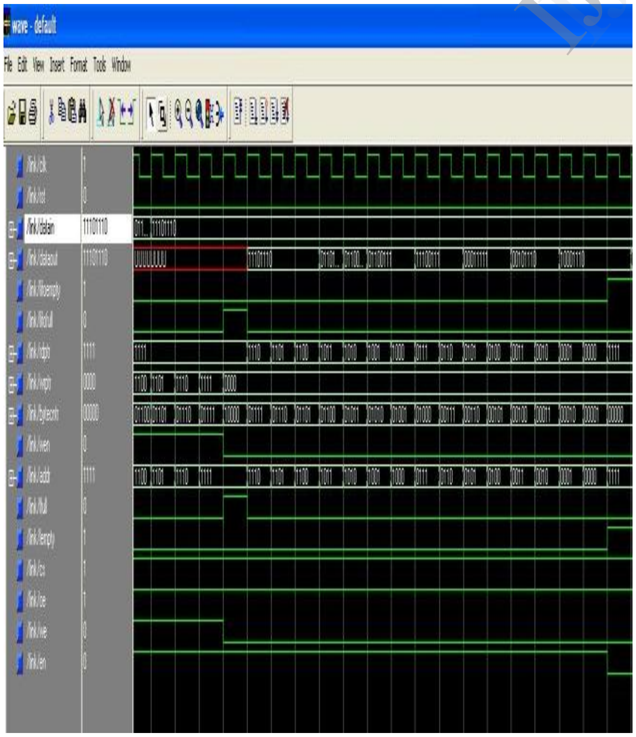
SIMULATION RESULT OF DATA MEMORY



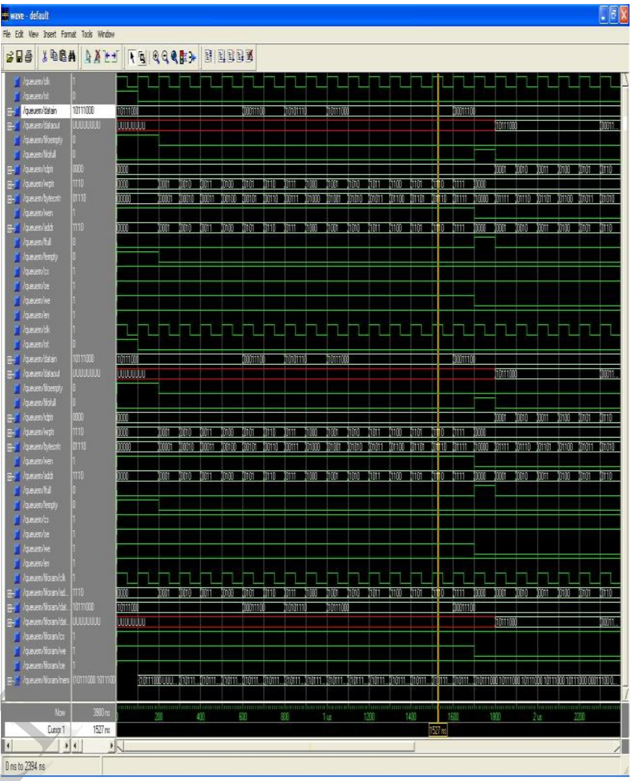
SIMULATION RESULT OF OUTPUT SELECTOR



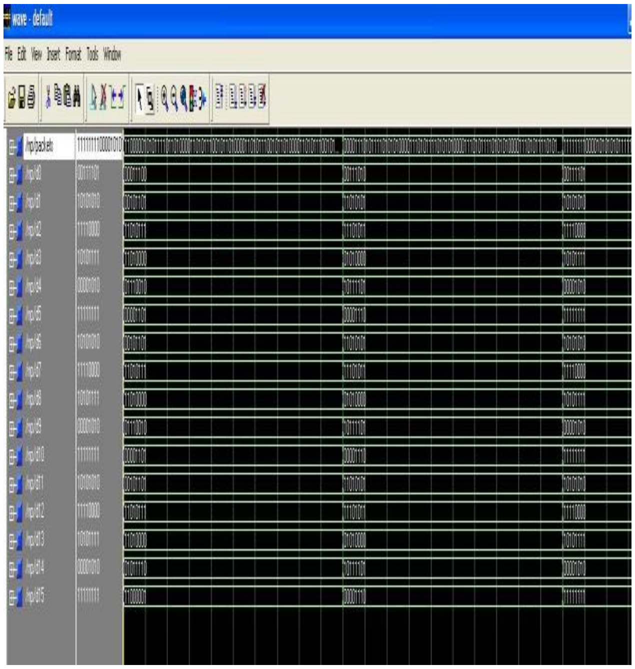
SIMULATION RESULT OF LINKED LIST MEMORY



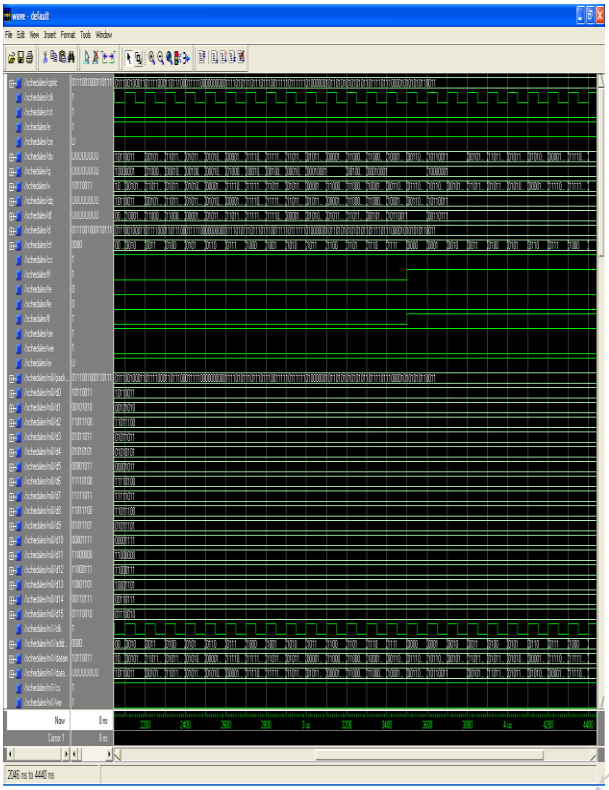
QUEUE MANAGER SIMULATION RESULT



SIMULATION RESULT OF NETWORK PROCESSOR



SIMULATION RESULT OF SCHEDULER



5. CONCLUSION

Proposed design of the scheduler for the non-blocking internet router based on the SGS algorithm optimized the scheduler sub components so as to provide lower packet delays. The proposed scheduler can be used in high capacity packet routers that provide delay guarantees.

REFERENCES

[1]A, Smiljanic, Milos petrovic, Milos blagojevic, “Design of the Switching Controller for the High-capacity Non-blocking Internet Router.”IEEE Trans.vlsi system,vol,17,No,8,August 2009.

[2]A,Smiljanic, “Flexible bandwidth allocation in terabit packet routres”, in proc, IEEE HPSR, Jun 2000,pp,233-241.

[3]H.J Chao, “Saturn: A terabit packet router using dual-round robin” IEEE Commun, mag,, vol,38,no 12,pp,78-84,dec 2000.

[4]N.Mckeown, M, lzzard, a, Mekkittikul, W, Ellersick, and M. Horowitz, “The Tiny tera: A packet router core,” IEEE Micro, vol,17, no,1, pp,jan-feb1997.

[5] A.Smiljanic, “Bandwidth reservations by maximal matching algorithm” IEEE commun.Lett,vol.8,no,,3 pp,177-179,Mar,2004.

[6]M.Petrovic, M,Blagojevic, ”Design, implementation, and testing of the Controller for the terabit packet routers” in proc,IEEE ICCCAS,2006,vol,3,pp1701-1705.