A Novel High Speed Asynchronous Scalable Variable-Length Self-Routing Packet Switch

Ting-Yuan Wang, Kun-Tso Chen, Hsien-Po Shiang, Min-Lian Yang, Hen-Wai Tsao, and Jingshown Wu
Department of Electrical Engineering, Graduate Institute of Electronics Engineering, and Graduate Institute of Communication Engineering

National Taiwan University,

Taipei, Taiwan, R.O.C.

Abstract—**In this paper, we propose a novel high-speed asynchronous scalable variable-length self-routing packet switch, called NTU-Ⅱ switch. The architecture is based on multi-plane crossbar structure with self-routing switch element. Contrary to conventional crossbar switches, the proposed switch does not need any central controller, complex scheduling schemes, speedup mechanism, or arbiter. The routing mechanism works in each self-routing switch element independently and uses polling handshaking protocol to transfer packets between internal circuits. Therefore, the modular design, highly scalability, and asynchronous transfer become possible and easier. The throughput of proposed switch is analyzed and simulated. Owing to the sub-switch plane mechanism, the throughput of a four plane switches without input port expansion can easily reach 100% simulated in both uniform distribution input traffic and hot spot input traffic model. The architecture has been verified in 100Mbps Ethernet system with FPGA, and completely simulated in 1Gb/s (IEEE 802.3z) Ethernet system with FPGA (VirtexII vc2v3000fg676-4, Xilinx Corp.). Because of its simplicity and scalable design, the proposed switch also may be implemented by VLSI technology to meet the 10Gb/s (IEEE802.3ae) Ethernet switch requirements.**

Keywords- Self-routing switching element; multi-plane crossbar switch; channel grouping; Sub-switc

I. INTRODUCTION

Recently, the Internet has been growing at very rapid pace. In general, a network consists of transmission section and routing/switching section. In the transmission section, the optical transmission technology advances very rapidly in backbone networks such as OC-48 (2.488Gb/s) and OC-192 (9.952Gb/s). Besides, the number of wavelength in a dense wavelength multiplexing (DWDM) system can be several hundreds and exceed one thousand in the future.

In access network, the 1Gb/s (IEEE 802.3z) and 10Gb/s (IEEE 802.3ae)[1] Ethernet network are essentially necessary with the demand of IP telephony, video on demand (VoD), interactive gaming or two-way videoconferencing.

During last two decades, Banyan, knockout, and speedup crossbar architectures have been proposed to implement high capacity switches[3-6]. The combined input output queue (CIOQ) crossbar switch with speedup became very popular because of its low cost and non-blocking properties[7-9]. Most of them are very complex and need speed up to operate in ultra high bit rate or need a powerful central control unit. Besides, they can only handle fixed length packets in synchronous mode. However the length of an Internet protocol (IP) packet may vary from 40 bytes to 65535 bytes. To fragmentize the IP packets into pieces will increase the overhead and the random port access time will reduce the throughput of the switch with synchronous mode. To design a large port count, high speed, high capacity, and variable length packet switch with conventional switch architecture is too complex.

In this paper, we propose a novel high-speed asynchronous scalable variable-length self-routing packet switch with distributed queue, which is fundamentally different from most of the switches reported in the literature. We employ a 2-bit address correlator and cross/bar path splitter in each switching element (SE), which realizes self-routing function and transmits packets in switching fabrics for high-speed pipelining operation. We let the packets enter the switching fabrics sequentially and employ the global busy indicator in each sequencer and the request signal in every row of switching elements to solve the output port contention problem.

For a single plane, input queue crossbar switch without speedup, the maximum throughput under uniform fixed length packet traffic is limited to about 58%[10], and under uniform variable packet length traffic is limited to about 50%[5]. The proposed switch employs multiple crossbar planes to improve throughput and divides the switch fabric into pieces, called sub-switch-plane, with channel grouping mechanism[2] to reduce the equivalent input traffic load and increase the transmission capacity easily. Although the multiple switch planes and channel grouping mechanism will increase the complexity in circuit design, yet it will significantly improve the throughput, especially for 10Gb/s Ethernet. Simulation results show that a reasonable size NTU-Ⅱ switch with four planes and four groups will have 100% maximum throughput with 80% maximum equivalent input traffic load, which matches OQ switches. Besides, the self-routing SE and distribution queue structure not only eliminate the complex central control unit or speedup mechanism but also increase the successful connection probability with the self-routing mechanism and non-blocking property. Moreover, modular and scalable design becomes possible due to the sub-switch plane, self-routing SE, and channel grouping mechanism. So we can scale the size of an NTU-Ⅱ switch from 16 × 16 to 128 × 128 or more. A 8 × 8 100Mb/s Ethernet NTU-Ⅱ switch has fully verified by FPGA (APEX 20KE EP20K 400 EBC 652-1X, Altera Corp.), and a 16 × 16 1Gb/s Ethernet NTU-Ⅱ switch has been completely simulated with four FPGAs (VirtexII xc2v3000fg676-4, Xilinx Corp.).

The remainder of this paper is organized as follows: Section Ⅲ describes the architecture and operation flow of the NTU-Ⅱ switch. Section Ⅳ describes the simple analysis and numerical results and section Ⅴ describes the implementation results, FPGA simulation and experimental results included.

II. THE NTU-II SWITCH ARCHITECTURE AND OPERATION

Fig. 1(a) shows the architecture of an NTU-IIswitch with broadcasting channel-grouping mechanism. Through this paper, we assume that the switch size is N × N where N is the

Error! Not a valid link. Fig. 1(a) the architecture of NTU-II switch

Error! Not a valid link. Fig. 1(b) the architecture of the sub-switch

Error! Not a valid link. Fig. 1(c) The channel grouping mechanism

number of input and output ports and the group number is G where N/G output ports denote a group. The packets are switched in sub-switches in Fig. 1(a). Fig. 1(b) shows the architecture of the sub-switches in the NTU-II switch, which consists of filtering preprocessor, sequencer, pipeline switch fabric and output-queue-management-unit (OQMU). Channel grouping mechanism is shown in Fig. 1(c) where packets are broadcasted by the linecard and filtered by the filtering preprocessor.

Request-polling and queue-receive-and-transmission mechanism is adopted as handshaking protocol in the internal circuit block. Therefore, each block stores the received packet in the local queue and transmits from there to its destination. When the local distributed queue is free, it will act as a receiver and send a request to the other local queue, which has stored packets waiting to be transmitted. The local queue management unit will transmit the packets after examining the port address, Fig. 1(a) the architecture of NTU-II switch group index and request indicator. Then the local receiver stores the packet in the local distributed queue and the local queue management unit will do the same thing as the first one to transmit packets. Therefore, the packets will be transmitted and received in the local circuit concurrently, or it will be stored in the queue waiting for the inferior request signal for transmission. This mechanism not only guarantees the correctness of packets transmission, but also increases the transmission capacity and reduces the transmission latency in the switch. Fig. 2 shows the operating flow of the handshaking protocol used in the NTU-II switch.

Once the IP packets enter into one of the input port, the linecard will examine the packets and attach a header and a

Error! Not a valid link.
Fig. 2 the operating flow of the hand-shaking protocol

Error! Not a valid link.

Fig. 3 The definition of the attached header and tailer

tailer to each IP packet as denoted in Fig. 3. The destination port address and group index, which are assigned according to the IP address and the group number of the switch respectively, are placed in the header. Since the switch handles variable length packets asynchronously and the transmission data bus is 16-bit, an even/odd (E/O) indicator bit denoting whether the packets using 8-bit data or 16-bit data in the last 16-bit word at the end of the packet transmission is necessary. With the same reason, the packet length information and header/tailer (H/T) indicator bit acts as beginning and ending delimiter for internal operation use in the switch.

After header and tailer attached, the linecard examines the request signals coming from the specific sub-switch group, which is defined by the group index. Then the packet is broadcasted to all the sub-switches. The filtering preprocessor in each sub-switch will check the group index and decides whether the packet can enter the switch fabric. If the group index is not the same as the group to which the preprocessor belongs, the preprocessor will drop the packet. Otherwise, the packet is transmitted and received by preprocessor successfully.

The sequencer receives the packets, which come from the relevant preprocessor and waits for the proper sequential time slot to check the request signal coming form the specific switch element and decides whether the stored packet needs to be broadcasted or not. After the sequencer delivered a packet, all switch elements lying at the same row in the sub-switch plane can fetch the packet from the global data bus and decides to establish the routing path or ignore the packet. Since the request signal is active, the request-polling mechanism guarantees that the routing path is free to wait for packets to establish the transmission sessions. Then the packets are transmitted into the OQMU and passed to the linecard at the output port.

To illustrate the architecture more clearly, we consider an NTU-II switch with size N × N, all output ports divided into G groups, and R input ports share a preprocessor. Each output port of the preprocessor is connected to a single specific sequencer followed by corresponding horizontal input port of a crossbar sub-switch plane. Totally, there is G sub-switches with size of N × N/G, N/R preprocessors, N sequencers, and P sub-switch-plane in each sub-switch. Each crossbar sub-switch-plane consists of N2/PG switching elements, which form an N/P × N/G matrix. The total number of switching elements in an N × N NTU-II switch is N2, which is large when the port number increases. However, because of the fact that structures of all switching elements are the same beside the different output port value which is stored in the correlator. Furthermore, the advantages of no central control unit, the scalability and modular design make the switch to a large size, which is easily partitioned into sub-circuits into a smaller size. So it can be easily realized by today's VLSI technology. At the output stage, there is a unique output queue connected to each output port of the sub-switch plane. Owing to structure with P multiple sub-switch planes, there are P output-queues belonging to different sub-switch plane connected to the same output port of the whole switch. Then the OQMU will read out the packets from the P output queues by multiplexer with FIFO scheduling.

For our convenience to illustrate the operation of the switch, we assume that N is 16, P is 4, G is 4, and R is 2. Therefore, output port 0 to 3 is treated as group 0 and the sub-switch which process the packet to output ports, which belongs to group 0, is taken as sub-switch 0. The linecard, preprocessor, sequencer, and crossbar plane are the key blocks of an NTU-II switch. We will describe them in detail as follows although the linecard is not in the scope of the proposed switch.

A. Linecard

Once the packet comes from MAC, the linecard will attach the header and tailer to the packets as shown in Fig. 3. Then the linecard will examine the request coming from the specific sub-switch, which defined by the group index in header. If they are matched, the packets are broadcasted without considering the status of other request signals from other sub-switch. Otherwise, the packets will be stored in the linecard and wait for the request-enabling signal.

After the first packet has been transmitted, the next packet with different destination address can be transmitted to other sub-switch group right away no matter whether the previous packet is blocked in the sub-switch because the two packets are propagated into two different sub-switches.

Although a linecard may transfer packets continuously with 100% traffic load, there are four sub-switches to receive the packet coming from the same linecard. The input traffic load in each input port of the sub-switch is reduced to 25% under uniform output port distribution model and 80% under hot spot distribution model with channel grouping mechanism. This shows that we can significantly improve the throughput of switches with high speed and large port count operation, because the actual traffic load in each sub-switch plane is less than 80%.

B. *Filtering Preprocessor*

The filtering preprocessor consists of two First-In-First-Out (FIFO) buffers, two group filters, a queue management unit (QMU), two independent queues which serve the corresponding input packets coming from the specific linecards, and an output distributor. The packet is sent by the linecard to the FIFO using 18-bit data bus. The 18th bit is the clock signal of the linecard, which is used as the write reference clock of the FIFO to fetch the packets data. The 17th bit is the mask signal transmitted with the packet data that the FIFO used this signal as a write enable signal and the other 16-bit data bus is used to transmit packet data. This is what we called source synchronous mechanism that can solve the problems of transmission skew. The filtering preprocessor accepts the packets from the corresponding input ports. At first, the group filter will check the group index to determine if the output address of the packet belongs to the group. If the output address belongs to the group, the packet is written into the specific queue, otherwise the packet will be dropped. The specific queue is constructed by a 16 × 784(1568/2) 2-port embedded block SRAM. Therefore the packet in the queue can be written and read to transmit by output distributor concurrently to minimize transmission latency. The output distributor acts as a packet transceiver in the preprocessor, which has two output ports connected to the corresponding sequencer at the input of different switch planes. Once the packet has been stored in alternative specific queue, the output distributor checks the request signal coming from the relevant sequencer and transmits the packet to one of the sequencers if any one of them is idle and waiting for packets.

A multiplane switch structure may have several paths destined for one output port concurrently. In conventional multiplane switches, a time stamp is attached to the header and the OQMU uses this time stamp to recover the packet order. But this method complicated the switch design and made it impossible for modular design of OQMU. Our proposed switch can solve this problem by adding specific queue and destination output recorder in the QMU. The destination output recorder records the output address of the last delivered packet. Once the next packet comes from the same input port, the destination recorder will check if its output address is the same. If the successive packets coming from the same input port have the same destination output port, the latter packets must be transmitted to the same sequencer as the previous one, no matter the other connected sequencer is available for transmitting or not. The packet sequence order for the same output port can be easily maintained.

Although adding specific queue and destination output recorder can solve the packet sequence problems, there are still two other performance problems that may occur. For example, there are four consecutive incoming packets with destination ports {3, 3, 1, and 2} sequentially, and outputs of the preprocessor are all empty. The first packet is delivered to sequencer successfully but is waiting for being transmitted into the sub-switch plane. The next packet is stored in the queue and waits for transmission to the same sequencer because the previous packet is blocked at the sequencer. Under this circumstance, the following packet can't be transmitted even if it is to be transmitted to a different destination port because the previous packets occupy the specific queue and sequencer. Fortunately, there is also a destination output recorder in the linecard so that if the destination output of the coming packet in linecard is different from the output address of the last sent packet, the linecard also delivers the packets to other available sequencers. The preprocessor will pass through the following packets to other available sequencers. This mechanism can prevent the packets from being blocked by the consecutive packets with the same destination port.

Error! Not a valid link.

Fig. 4 the schematic of the filtering preprocessor

The other problem arrives if the four consecutive incoming packets have the destination ports like {3, 1, 3, 2}, the 3rd packet may not follow the transmission like the 1st packet because the destination output recorder can only record the last packet. However, since we can achieve 100% throughput with output queue mechanism, this also implies that only 2-packet buffer space of the input queue is enough according to the simulation results [9]. So when the 3rd packet arrives, the 1st one has already been switched to the output port. Therefore, The packet sequence problem caused by the output port sequence like {3, 1, 3, 2} will never happen in NTU-II switch. The architecture and waveform of the filtering preprocessor is shown as Fig. 4.

C. *Sequencer*

There are total N sequencers, which control the sequential time points to deliver packets into the switch fabric. The sequential time points are generated by the shift register as shown in Fig. 5(a). Each of the sequencers associated with a given N/P × N/G sub-switch plane may deliver packets into the switch at the specific time point sequentially. Fig. 5(b) shows a sequencer consists of a local queue, QMU, a shift-time register, and a state controller. The local queue and QMU acts as local receiver to receive packets from the preprocessor and the local queue plays the same role as the buffer in the preprocessor. The shift-time registers in all the N/P sequencers associated with the same switch fabric form a ring-shift-register. It generates trigger signal to enable the specific sequencer to check whether the packet can be delivered into the switch fabric sequentially and cyclically. Therefore, there are P ring shift-registers corresponding to the P switch fabrics and each of them operates independently.

Error! Not a valid link.

Fig. 5(a) the round robin fashion of the sequentially entrance trigger

Error! Not a valid link.

Fig. 5(b) the schematic of the sequencers

Since the ring-shift-register is driven by the system clock without speedup, the sequencers send the packets in a round robin fashion with a time interval τ, where τ is the period of the system clock. So each sequencer can be triggered to check and send packet with a time interval Nτ/P. The performance will deteriorate with increased N since the packets have to wait a longer time to be delivered. Fortunately, considering the constraint of the IC package limitation in physical design, we must keep the size of sub-switch plan as 4 × 4 for modular design. Therefore the sequencer will be enabled with time interval 4τ. Practically the additional waiting time 4τ is negligible because the packet transmission time is much longer than 4τ. So the switch operation is considered as asynchronous.

After a packet enters the sequencer, the sequencer will check the global busy signal, the request indicator from switch elements, and the address in the packet header. There are N/G request indicators coming from the N/G independent switch elements in the same row, which is connected to the same sequencer of the same sub-switch plane. Each switch element lies on the specific column belonging to the same destination output. The sequencer will check the address index and sense the busy indicator signal with the same destination output address. If the specific global busy line is low and the busy indicator signal is high, the switching path is available. Thus the sequencer will broadcast the packets with the same handshaking protocol as used in the preprocessor. Owing to the pipeline design of sub-switch plane and the high-speed operation, it is impossible to inform the next sequencer that other packets occupies the destination output by using busy indicator feedback signal, which comes from the switch element at the next row. Since the handshaking protocol guarantees once the packets are broadcasted, they must be successfully transmitted and received at the destination output. For this reason, while the packets are delivered, the sequencer will pull the global busy signal high, to inform other sequencers that the specific vertical pipeline bus is used. All other sequencer will receive the global busy indicator signal with vertical pipeline propagating in the same manner as the packet propagating in the sub-switch plane. So the next driven sequencer will not deliver packets with the same output address to the sub-switch plane.

D. *Switch fabric*

The switching fabric is composed of P crossbar sub-switch planes. Each plane consists of N2/PG switching elements arranged in an N/P × N/G matrix form. There are N/G switching elements lying in a row corresponding to the same sequencer and N/P switching elements in a column corresponding to one destination output. The switch element in the i-th row and the j-th column is capable of routing the packet from the i-th sequencer to the j-th output port. Fig. 6 shows the schematic of the sub-switch plane and the block diagram of the switch element.

The switching element consists of a header extractor, an address comparator, a multiplexer, and an output latch. The header extractor will listen to the signal all the time. The switching elements are connected to the horizontal global data bus and vertical pipeline bus. At the beginning, the switching element is idle with the bar state. When the packet is broadcasted in the bus, there is also a packet indicator window signal delivered by the sequencer to inform the header extractor to extract the header and tailer of the packets. If the output port is available and the extracted address from the packet header matches the pre-stored output port address, the address

Error! Not a valid link.

Fig. 6 The schematic of switch elements

comparator will control the cross/bar multiplexer to relay the data from the horizontal bus to vertical pipeline bus. Then the multiplexer output is connected to the output latch with vertical pipeline bus to transmit data to next switching element. In this situation, the switching element is at the cross state and the busy indicator is active. So the following multiplexers in the same column will remain in bar state and pass data to the output port. Because the busy indicator is active, it prevents other switching elements to deliver data to the same output port. Therefore packet collision is impossible.

E. OQMU

The output queue management unit (OQMU) consists of a two-port memory, a memory management unit, and multiplexers as shown in Fig.7. The two-port memory is adopted as the output queue to increase the throughput performance since packets can be written by the sub-switch plane and read out to the linecard concurrently. Each column output of the sub-switch plane is connected to its independent queue to store the packets. The packets with the same destination output coming from four different sub-switch planes are stored into its own memory and read out to the linecard by the 4-to-1 multiplexer with FIFO scheduling. The memory depth is 1536 × 16 bit, which is twice the length of the longest Ethernet packet. The simulation shows that it is enough to implement output queue with 100% throughput.

III. Simple Analysis and Numerical Results

In the following analysis, we assume that there is no distributed queue, and all the input queue are saturated so that packets are always waiting in linecards. Without distributed queue, we guarantee correct handshaking transmission and to reduce internal latency. Each packet has equal probability of 1/N destined to a given output port. The transmission time of the header and the tailer is ignored, because the packet length is long enough. The packet arrival and length are Poisson and geometric distributions. The addresses of consecutive packets are uncorrelated. Each sequencer can store only one packet and send a packet at an allowable time point.

The maximum throughput, ρ max, is defined as the utilization factor per output port. There are totally P crossbar sub-switch planes, N virtual queues corresponding to each

Error! Not a valid link.

Fig. 7 The schematic and operation time chart of output queue management unit

output port and N/P packets including both of transmitting and waiting packets in the virtual queues. The maximum throughput of this system is equivalent to the throughput of a closed queuing network with N/P customers and N queues[5]. Assume that after Sequencer i has transmitted a packet to Output j1, the probability of Sequencer i to deliver the next packet to Output j2 is 1/N and the packet is waiting for serving in the virtual queue of Output j2. This corresponds to the event in the closed queuing network that Input i departs from Queue j1, and is routed to Queue j2 with probability 1/N. Although the service discipline is not FIFO, it will not affect the throughput with the assumption of geometric service time distribution and memoryless property. The busy probability of any particular queue in a closed queuing network is shown as follows[5]. Let λ_i be the total arrival rate at Queue i in a closed queuing network, i =1...N, and μ denote the common service rate of queue i, i=1...N. In the steady state, because of flow conservation we have

$$\lambda_i = \sum_{j=1}^{N} \lambda_j P_{ji}, i = 1 \ldots N.$$

(1)

Under the uniform traffic assumption, the routing probability Pji equals 1/N. Hence, (1) has a particular solution

$$\overline{\lambda_i} = 1, i = 1 \ldots N.$$

(2)

We can use Jackson's Theorem for Closed Networks[11] to obtain the steady state probability distribution, P(n), as

$$P(n) = K^{-1} \prod_{i=1}^{N} \hat{P}_i(n_i)$$

(3)

where K is a normalization constant, state n=(n_1,...,n_N) denotes that there are n_i customers in Queue i, with n_1 +...+ n_N = N/P, $\hat{P}_i(n_i)$ and K are given by

$$\hat{P}_i(n_i) = \begin{cases} 1, & \text{if } n_i = 0 \\ \mu^{-n_i}, & \text{if } n_i > 0 \end{cases}$$

(4)

and

$$K = \sum_{\{n|n_1+\ldots+n_N=NR/P\}} \prod_{i=1}^{N} \hat{P}_i(n_i) = \mu^{-NR/P} \binom{N+N/P-1}{N/P}. \qquad (5)$$

Hence the probability of one particular Queue, says Queue 1, being busy is

$$\Pr[busy] = 1 - \sum_{\{n|n_2+\ldots+n_N=NR/P\}} P(n) = 1 - \frac{\binom{N+N/P-2}{N/P}}{\binom{N+N/P-1}{N/P}} = \frac{N}{N+NP-P}. \qquad (6)$$

Therefore, as the packet length is long enough, the maximum throughput of our switch is given by

$$\rho_{\max} = \frac{NP}{N+NP-P}$$

(7)

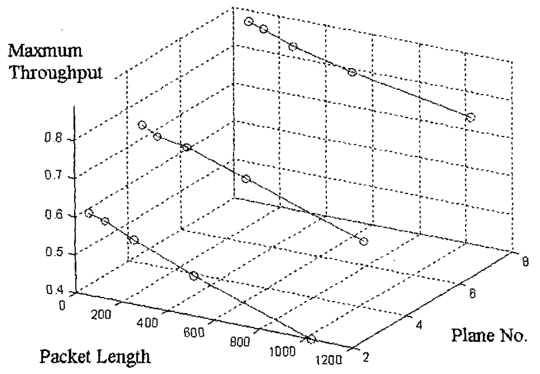


Fig. 8 the analysis result of the maximum throughput vs. variable packet length and multiple sub-switch planes

For uniform traffic distribution, if the input traffic load is 100%, the input traffic load at each sub-switch is 1/G owing to the channel grouping mechanism. With group index chosen to be four, the maximum input traffic load is 25% in each sub-switch and the simulation result shows that 100% throughput can be easily achieved.

We have performed a discrete time event-driven simulation. The propagation time τ is set to one switch clock time. Fig. 8 shows the maximum throughput with various packet lengths and plane numbers. Fig. 9(a) and Fig. 9(b) show the simulated maximum throughput for various switch sizes with one plane and four planes, respectively. In a single plane 16 × 4 switch, the maximum throughput must less than 25%. Obviously that the worst case, which means all 16 input traffic goes into the four output port belongs to the same group, of the maximum throughput increase from 54% to 86% owing to the multiplane structure in each 16 × 4 sub-switch. Fig. 10 shows the maximum throughput with different traffic loads and switch planes. The maximum throughput can reach 100% with 80% input traffic load at most with four switch-planes. And a 16 × 16 switch is constructed by four 16 × 4 multiplane sub-switch.

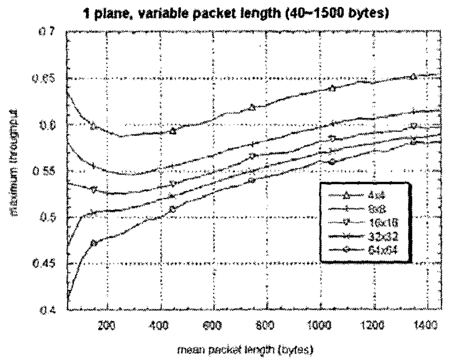


Fig. 9(a) the maximum throughput analysis results vs. difference packet length and difference switch size with only one switch plane

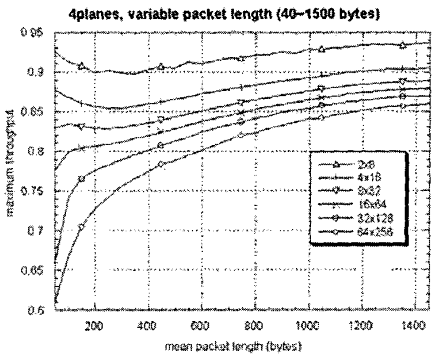


Fig. 9(b) the maximum throughput analysis results vs. difference packet length and difference switch size with four switch planes

In uniform case, each sub-switch has only 25% input load and has the capability of 80% output throughput. The multi-plane crossbar switch can improve at least 30% performance contrary to the traditional architecture. Therefore, four switch-planes with channel grouping mechanism can achieve 100% maximum throughput with 100% traffic load.

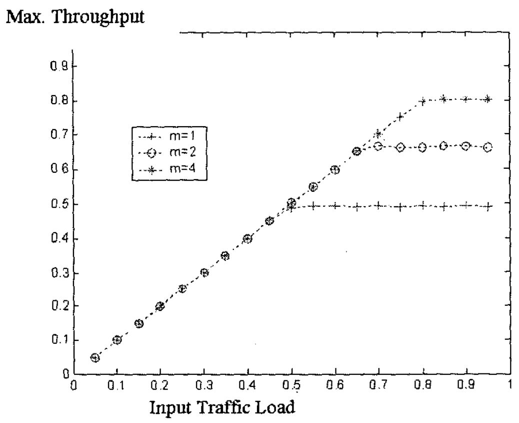


Fig. 10 the maximum throughput analysis results with different input traffic load and different sub-switch plane numbers with buffer size of 1536 × 16 bit

Fig. 11 shows the packets loss when the traffic load increases with different plane numbers. The packet loss can be less then 10^{-3} in four switch-plane systems.

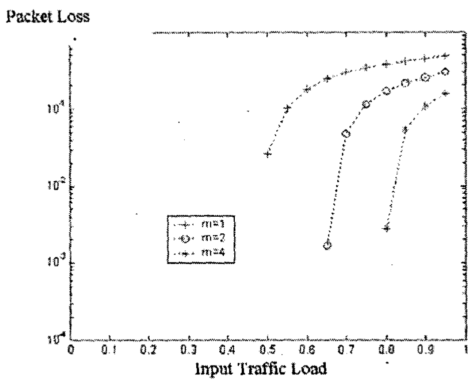


Fig. 11 the packet loss analysis results with different input traffic load and different sub-switch plane number with buffer size of 1536×16 bit

Consider the worst case of non-uniform traffic load condition[12]. Hot spot traffic is a simple kind of non-uniform traffic. Assume that there are two groups of outputs: hot spot group and normal group. For simplicity, we assume only two probabilities for the hot spot traffic, which means that the output ports in the same group share the same probability.

The Group factor, q, is used to define how large the hot spot group can be. If the number of ports in group i is $|G_i|$, and the total number of output ports is N, the group factor, q, can be defined as:

$$q \equiv \frac{|G_1|}{N}$$, where we assume the hot spot is group one. (8).

A packet arriving at any input is assigned any output of Group 1 with equal probability P_{g1} or any output of Group 2 with equal probability P_{g2}. Note that $\sum|G_i|\times p_{gi}=1$. Then the relative intensity factor, p, can be defined as:

$$p \equiv \frac{p_{g1}}{p_{g1}+p_{g2}} \quad (9).$$

Because Group 1 is the hot spot group, $P_{g1} \geqq P_{g2}$, i.e. $0.5 \geqq p \geqq 1$.

From (8) and (9), we can further define the probability of choosing a port from Group 1 or Group 2. Let P_i be the probability of choosing an output from Group i, we have$P_1+P_2=1$. Based on $\sum|G_i|\times p_{gi}=1$ and $|G_1| + |G_2| = N$, we can find that

$$P_1 = |G_1| \times p_{g1} = \frac{pq}{1-p-q+2pq} \quad (10).$$

$$P_2 = |G_2| \times p_{g2} = \frac{1-p-q+pq}{1-p-q+2pq} \quad (11).$$

For the physical design consideration, we use group index as four. Then the group factor is 0.25 as constant. Fig. 12 shows the probability throughput versus p and q. It is obvious that when q is 0.25, the maximum hot spot probability is less than 75% no matter how large p is. It means that the input traffic load of each sub-switch is not more than 75% and the throughput can reach 100% with four switch-planes architecture easily.

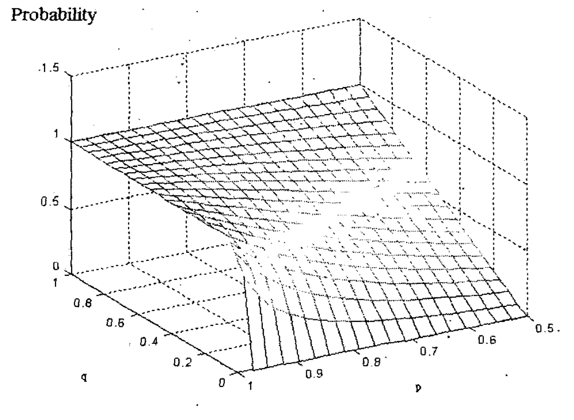


Fig. 12 The hot spot probability analysis results

IV. Implementation Results

A. FPGA Simulation

The proposed switch with size 16 × 16 has been fully post-simulated by FPGA (XC2V3000-4 fg676, XILINX Corp.) Four sub-switches with size 16 × 4 compose the proposed switch and each sub-switch is implemented using a single FPGA chip. The operation frequency of the sub-switch is 62.5MHz, which is enough for 1Gb/s (IEEE 802.3z) use since the data bus width is expanded to 16-bit. The hardware of a prototype 1Gb/s Ethernet NTU-II packet switch system is now under construction.

B. Experimental Results

The proposed switch with size 8 × 8 has been implemented and fully verified by FPGA (APEX 20KE EP20K 400 EBC 652-1X, ALTERA Corp.) for 100Mb/s Ethernet system. The entire testing environment consists of traffic generator (IXIA, supported by CCL of ITRI), PHY(88E1000, MARVELL Corp), MAC(FPGA supported by Computer and Communication Research Lab(CCL) of Industrial Technology Research Institute(ITRI)), and output queue(IDT 70V3589S ,Integrated Device Technology, Inc). The 8 × 8 NTU-II switch is shown in Fig.13.

To verify the function, we distribute packet traffic into the four input ports. The packet lengths are of different size (i.e. 64, 256, 512, and 1512 bytes). We record the dynamic statistical received results until each output port has received one-gigabyte data. Fig. 14 illustrates the testing results where the 6th and 8th row indicates the sending and receiving rate, and the 1st row shows the line rate with 100Mbps. The system has been connected to computers with Ethernet network card and tested by the FTP software. All of the testing results shows that the NTU-II switch works correctly in 100Mbps Ethernet

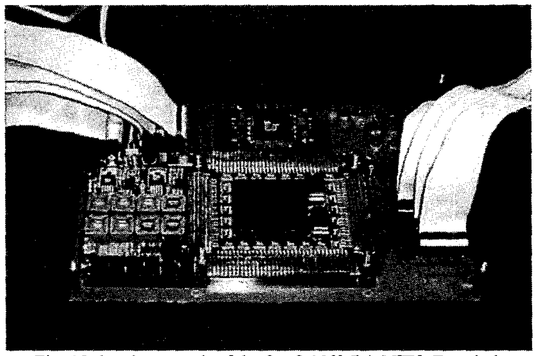
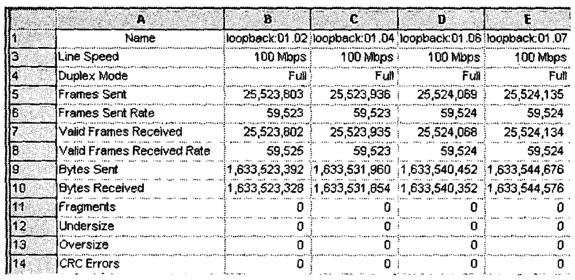
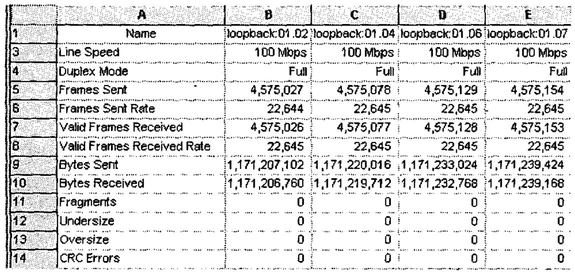


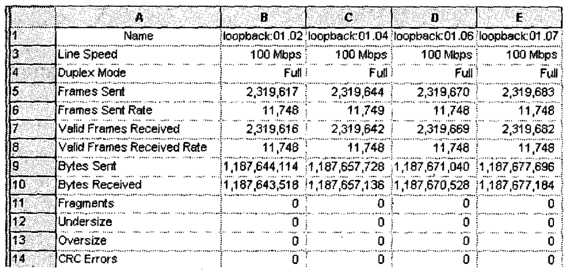
Fig. 13 the photograph of the 8 × 8 100Mb/s NTU-II switch



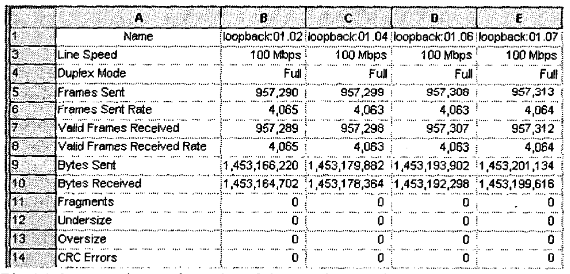
	A	B	C	D	E
1	Name	loopback:01.02	loopback:01.04	loopback:01.06	loopback:01.07
3	Line Speed	100 Mbps	100 Mbps	100 Mbps	100 Mbps
4	Duplex Mode	Full	Full	Full	Full
5	Frames Sent	25,523,803	25,523,936	25,524,069	25,524,135
6	Frames Sent Rate	59,523	59,523	59,524	59,524
7	Valid Frames Received	25,523,802	25,523,935	25,524,068	25,524,134
8	Valid Frames Received Rate	59,525	59,523	59,524	59,524
9	Bytes Sent	1,633,523,392	1,633,531,960	1,633,540,452	1,633,544,676
10	Bytes Received	1,633,523,328	1,633,531,854	1,633,540,352	1,633,544,576
11	Fragments	0	0	0	0
12	Undersize	0	0	0	0
13	Oversize	0	0	0	0
14	CRC Errors	0	0	0	0



	A	B	C	D	E
1	Name	loopback:01.02	loopback:01.04	loopback:01.06	loopback:01.07
3	Line Speed	100 Mbps	100 Mbps	100 Mbps	100 Mbps
4	Duplex Mode	Full	Full	Full	Full
5	Frames Sent	4,575,027	4,575,078	4,575,129	4,575,154
6	Frames Sent Rate	22,644	22,645	22,645	22,645
7	Valid Frames Received	4,575,026	4,575,077	4,575,128	4,575,153
8	Valid Frames Received Rate	22,645	22,645	22,645	22,645
9	Bytes Sent	1,171,207,102	1,171,220,016	1,171,233,024	1,171,239,424
10	Bytes Received	1,171,206,760	1,171,219,712	1,171,232,768	1,171,239,168
11	Fragments	0	0	0	0
12	Undersize	0	0	0	0
13	Oversize	0	0	0	0
14	CRC Errors	0	0	0	0



	A	B	C	D	E
1	Name	loopback:01.02	loopback:01.04	loopback:01.06	loopback:01.07
3	Line Speed	100 Mbps	100 Mbps	100 Mbps	100 Mbps
4	Duplex Mode	Full	Full	Full	Full
5	Frames Sent	2,319,617	2,319,644	2,319,670	2,319,683
6	Frames Sent Rate	11,748	11,749	11,748	11,748
7	Valid Frames Received	2,319,616	2,319,642	2,319,669	2,319,682
8	Valid Frames Received Rate	11,748	11,748	11,748	11,748
9	Bytes Sent	1,187,644,114	1,187,657,728	1,187,671,040	1,187,677,696
10	Bytes Received	1,187,643,518	1,187,657,136	1,187,670,528	1,187,677,184
11	Fragments	0	0	0	0
12	Undersize	0	0	0	0
13	Oversize	0	0	0	0
14	CRC Errors	0	0	0	0



	A	B	C	D	E
1	Name	loopback:01.02	loopback:01.04	loopback:01.06	loopback:01.07
3	Line Speed	100 Mbps	100 Mbps	100 Mbps	100 Mbps
4	Duplex Mode	Full	Full	Full	Full
5	Frames Sent	957,290	957,299	957,308	957,313
6	Frames Sent Rate	4,065	4,063	4,063	4,064
7	Valid Frames Received	957,289	957,298	957,307	957,312
8	Valid Frames Received Rate	4,065	4,063	4,063	4,064
9	Bytes Sent	1,453,166,220	1,453,179,882	1,453,193,902	1,453,201,134
10	Bytes Received	1,453,164,702	1,453,178,364	1,453,192,298	1,453,199,616
11	Fragments	0	0	0	0
12	Undersize	0	0	0	0
13	Oversize	0	0	0	0
14	CRC Errors	0	0	0	0

Fig.14 the experimental results of the 8 × 8 100Mb/s NTU-Ⅱ switch with 64, 256, 512 and 1512 packet length

operation.

V. CONCLUSIONS

This paper presents a novel high-speed asynchronous scalable variable-length self-routing packet switch, NTU-Ⅱ switch, which is based on multiplane switching fabric with self-routing capability and channel grouping mechanism. The architecture of NTU-Ⅱ switch is considered from hardware design and implemental ability point of views. The usage of channel grouping mechanism reduces the equivalent input traffic load. The hand-shaking protocol and busy indicator signal make the switching of variable length packet operation without fragmentation. The sequential packet entering time point control enables the switch to deliver asynchronous packets easily. Owing to the self-routing switch element designed using pipeline architecture, the central arbiter and speedup mechanism are not required. Furthermore, the modular and scalable design for the switch fabric should become possible and easier for high speed such as 10Gb/s Ethernet and OC-192.

A 8 × 8 100Mb/s NTU-Ⅱ switch has been fully implemented and another 16 × 16 1Gb/s switch is under construction with post-simulation completed. The 16 × 16 10Gb/s NTU-Ⅱ switch is under design with the same architecture and a 128 × 128 10Gb/s NTU-Ⅱ switch is also can be designed easily by multiple stage switch design with the same architecture in the near future.

REFERENCES

[1]. L. E. Frenzel, "*Physical-Layer Interfaces Reach Beyond 10 Gbit/s*", Electronic Design, vol. 49, No. 11, pp. 58-76, MAY, 2001

[2]. Glen Kramer and Gerry Pesavento, "*Ethernet Passive Optical Network(EPON):Building a Next-Generation Optical Access Network*", IEEE Communications Magazine, pp. 66-73, Feb., 2002

[3]. Y. S Yeh, M. G. Hluchyj and A. S. Acampora, "*The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching*", IEEE J. Selected Areas Commun, Vol. SAC-5, No. 8, pp. 1274-1283, Oct, 1987

[4]. A. K. Gupta and N. D. Georganas, "*Analysis of a Packet Switch with Input and Output Buffers and Speed Constraints*", Proc. IEEE INFOCOM'91, Miami, FL, pp. 694-700, Apr., 1991

[5]. Steve W. Fuhrmann, "*Performance of a Packet Switch with Crossbar Architecture*", IEEE Trans. Commun, Vol. 41, No. 3, pp. 486-491, March, 1993

[6]. L. R. Goke and G. J. Lipovksi, "*Banyan Networks for Partitioning Processor Systems*", Proc. 1st Annual Symp. Computer Architecture, pp. 21-28, Dec., 1973

[7]. M.G. Hluchyj and M. J. Karol, "*Queueing in High-Performance Packet Switching*", IEEE J. Selected Areas Commun., Vol. 6, No. 9, pp. 1587-1597, Dec., 1988

[8]. T. E. Anderson, S. S. Owicki, J. B. Saxe and C. P. Thacker, "*High-Speed Switch Scheduling for Local-Area Networks*", ACM Trans. Computer Systems, Vol. 11, No. 4, pp. 319-352, Nov., 1993

[9]. N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "*Achieving 100% Throughput in an Input-Queued Switch*", IEEE Trans. Commun., Vol. 47, No. 8, pp. 1260-1267, Aug., 1999

[10]. M. J. Karol, M. G. Hluchyj and A. P. Morgan, "*Input Versus Output Queueing on a Space-Division Packet Switch*", IEEE Trans. Commun., Vol. Com-35, No. 12, , pp. 1347-1356, Dec., 1987

[11]. D. Bertsekas and R. Gallager, Data Networks, 2rd ed. New Jersey: Prentice-Hall, pp. 233-235, 1992

[12]. Hsien-Po Shiang, "*Performance evaluation of the NTU switch: a high-speed variable length self-routing packet switch*", M.S. Thesis, Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, pp., Aug., 2002