PACKET SWITCHING PHOTONIC NETWORK SWITCH DESIGN AND ROUTING ALGORITHMS

Hyuk-Jun Lee, Martin M. Morf, and Michael J. Flynn

Technical Report No. CSL-TR-97-734

September 1997

This work is supported by BMDO under contract DAAH04-95-1-0123

Packet Switching Photonic Network Switch Design and Routing Algorithms

by

Hyuk-Jun Lee, Martin M. Morf and Michael J. Flynn

Technical Report No. CSL-TR-97-734

September 1997

Computer Systems Laboratory
Department of Electrical Engineering and Computer Science
Stanford University
Gates Building 2A, Room 230
Stanford, California 94305-9040
pubs@shasta.stanford.edu

Abstract

Maturity of photonic technology makes it possible to construct all optical network switch to avoid optical-to-electrical signal conversion for routing. To realize all optical packet switching, our current network topology and routing algorithms have to be reexamined and modified to satisfy the necessities of all optical network switching such as a fast routing decision, consideration of hardware implementation, buffering etc.

In this paper, first, we will review various switching architectures including crossbar, Benes and Batcher/Banyan. Secondly, optical implementation of a multiple output port network switch will be presented. In many levels of networking from multiprocessor interconnection to wide area networking, multiple latencies resulting from this scheme could improve the overall performance when combined with smart routing schemes. Finally, we present a interpretation of multistage network using a symmetric group. A Cayley graph for a symmetric group and its coset graphs suggest an interesting alternative way to construct a new multistage interconnection network.

<u>Key Words and Phrases:</u> all optical network, packet switching, Benes, Batcher/Banyan, Multiple output port network, Coset graphs

Contents

1. Introduction

In the past decade, many researchers proposed various new network topologies and routing algorithms to meet ever-increasing bandwidth requirement in various levels of networking and make use of state-of-the-art technology to achieve better performance. For instance, in multiprocessor interconnection, efforts to improve overall performance by means of exploiting parallelism using multiprocessors require much larger communication bandwidth while interconnection is required to maintain low latency, good scalability and connectivity. In WAN and LAN, cheap costs of propagation medium such as optical fibers and advent of commercial products for optical switching such as Lithium Niobate($LiNbO_3$) Optic devices[9] and Self-Electrooptic-Effect Devices(SEED)[8] made it possible to build simple optical interconnection. Recent achievements such as Multiple Quantum Well(MQW) modulators[6] and Sagnac exchange/bypass gates[7] which we will discuss in section 2 provide complete 2x2 network switching capabilities that is essential to build a multistage interconnection network. After a brief introduction of these photonic devices in section 2, section 3 will go over a review on various switching architectures. Then, section 4 will discuss the optical datapath design of a multiple output port switch. We will show a multiple output port Batcher/Banyan network as a baseline model. Finally, in section 5, we will introduce the relation between symmetric group and multistage interconnection network(MIN) and explain how its Cayley graphs and coset graphs can be used to design a MIN.

Figure 1&2 show the comparison of two major network topologies in terms of latency and hardware cost assuming the use of 2x2 switching elements. An optimal network line shows that a theoretical bound for the required number of stages to realize a full permutation. From this figure, we can recognize two things. First, Benes network is quite close to an optimal network. But, relatively large control latency, $O(\log^2(n))$ or $O(\log(n))$ depending on the assumption of pipelining, makes Benes network unattractive for packet switching. Parallel routing algorithms allowing dynamic control latency would be a good solution to this problem from that aspect. Secondly, Batcher/Banyan network provides simple self-routing capabilities but suffers from large latency, $O(\log^2(n))$, and hardware costs, $O(n\log^2(n))$. However, this large latency problem can be relieved by allowing multiple latency in routing and letting early routed permutation leave a network as early as they can through multiple ports. These two ideas are the major motivation that we reexamine these networks as a candidate for an all optical network switch in section 3 and 4.

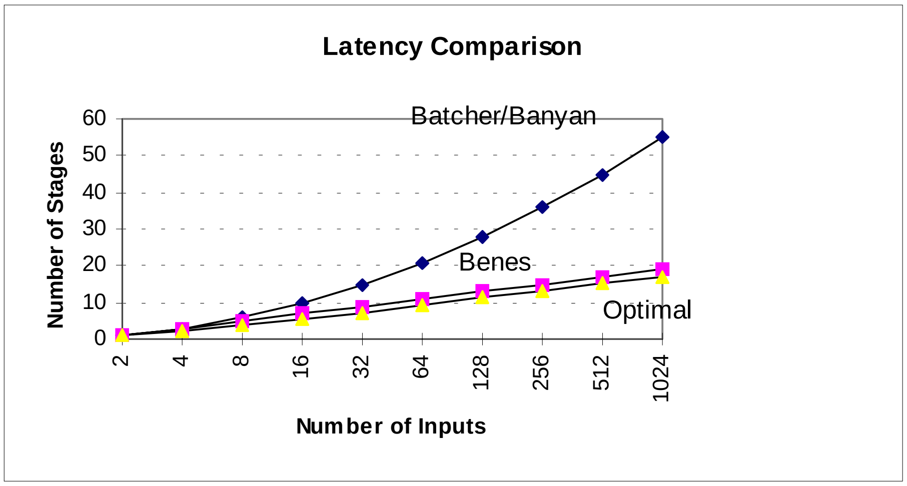


Figure 1

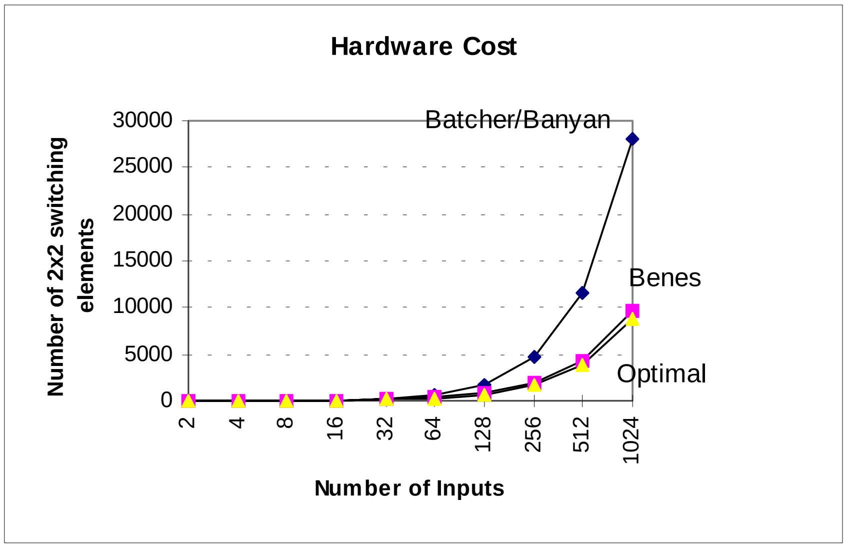


Figure 2

2.A brief introduction to photonic devices

Realizing switching in optical domain has been researched in many years[6]-[9]. Four major methods of realizing switching are amplitude modulation, phase modulation, polarization change, and directional change. Combination of these methods also provides switching. In this paper, we consider two photonic devices, multiple quantum well(MQW) modulator[6] and Sagnac gate[7]. The first device is in the category of the combination of the first and forth methods and the second device is using the first and second methods. From a functional point of view, both devices are realizing a Fredkin logic, figure 3, which basically connect two inputs to two outputs either in interchanged or non-interchanged way depending on the third control input.

The structure of a MQW modulator consists of a slightly asymmetric Fabry-Perot cavity containing top and bottom mirrors composed of 10 and 12.5 period GaAs-AlAs quarter wave stacks surrounding an undoped cavity of quantum wells. The picture is shown in figure 4. From top to bottom, it is doped as p-i-n. Under no bias, light incident on the top mirror is propagated into two paths: reflection and transmission. The transmitted wave is again reflected and transmitted on the bottom mirror. The reflected wave on the bottom is propagated backward and finally combined with the initially reflected wave on the top. The cavity is designed such that these two waves are 180° out of phase with same amplitude and cancel each other. As a result, we don’t see any reflected light and a cavity is transparent, which is a cross mode in a Fredkin logic gate. Under bias, activated quantum wells in the intrinsic region absorb all the transmitted light from the top mirror so that we don’t see any transmitted light coming from the bottom mirror and only observe the reflected portion of incident light on the top mirror due to no cancellation. This is a set-through mode in a Fredkin logic gate.

While a MQW modulator is an electrically controlled optical datapath logic, a Sagnac gate, figure 5, is an optically controlled optical datapath logic. Injected optical signal at input A goes though 3dB coupler and enters a polarization maintaining 3dB coupler in the second stage. Inside this device, optical signal is divided into two and delivered to the upper and lower polarization beam split couplers. Under no control signal at input C, these two optical signal travel the same distance in clockwise and counterclockwise direction though optical fiber and interfere each other constructively at the upper input of the second stage. As a result, optical signal injected at input A is emerged at output A after some delay. Under existence of control signal at input C, optical signal propagating in the clockwise direction co-travels with a control input and experiences Kerr effect. That is, increase in intensity causes light to travel slowly in the non-linear medium, which in turn causes phase difference. The length of fiber is such designed that phase difference between two opposite traveling waves is π. This difference makes original optical signal to interfere itself destructively at the upper input of the second stage and constructively at the lower input. Consequently, optical signal injected at input A is emerged at output B. Considering this device is symmetric, the whole logic performs the Fredkin logic with another input at input B. Table 1 shows the comparison of two devices.

C	A	B	C’	A’	B’
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

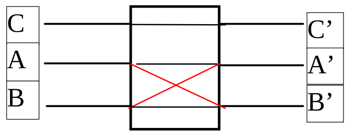


Figure 3. Fredkin logic gate: Red = Cross mode, Black = Set through mode

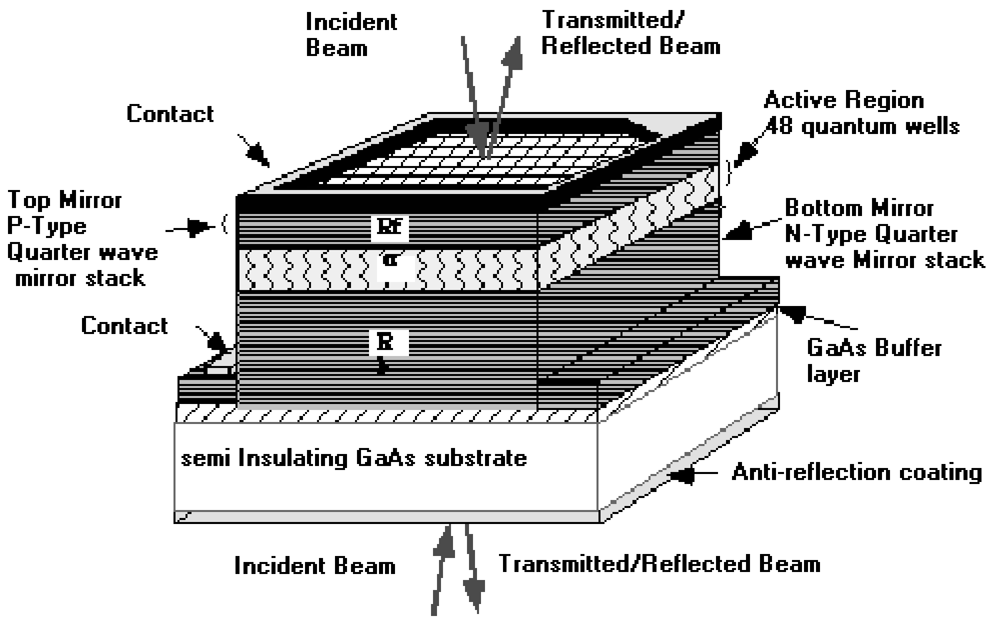

Incident Beam
Transmitted/ Reflected Beam
Contact
Active Region 48 quantum wells
Top Mirror P-Type Quarter wave mirror stack
Rf
α
Bottom Mirror N-Type Quarter wave Mirror stack
Contact
R
GaAs Buffer layer
semi Insulating GaAs substrate
Anti-reflection coating
Incident Beam
Transmitted/Reflected Beam

Figure 4. MQW modulator

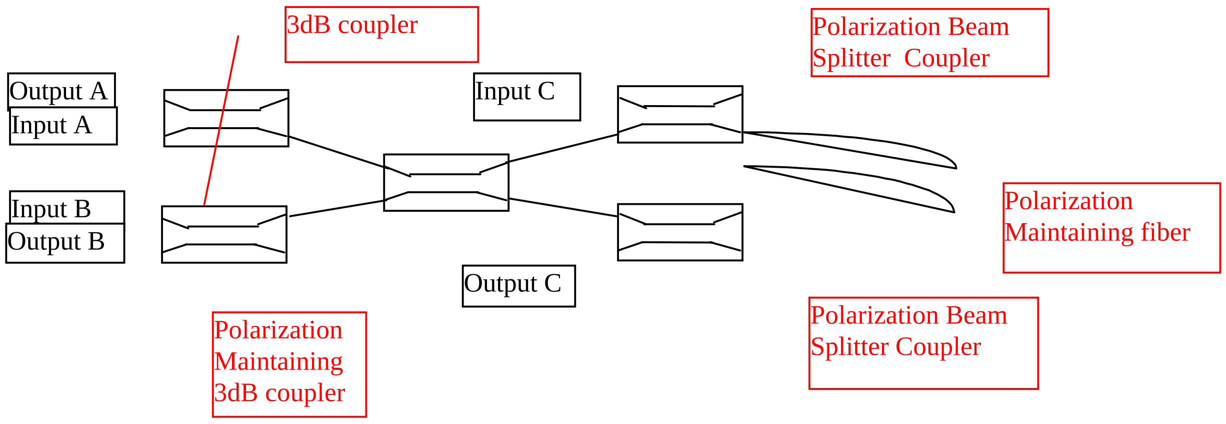

3dB coupler
Polarization Beam Splitter Coupler
Output A
Input A
Input C
Input B
Output B
Polarization Maintaining fiber
Output C
Polarization Beam Splitter Coupler
Polarization Maintaining 3dB coupler

Figure 5. Sagnac Exchange/Bypass Gate

	Switching Speed	Area	Switching Power	Input Wavelength	Latency
MQW modulator	>50 GHz (limited by capacitance)[6]	200 μm x 200 μm	N/A	965-975 nm	N/A
Sagnac Gate	N/A	Discrete components	~10 mW for control	1300-1600 nm	1.25 nsec @ 0.37 m loop

Table 1.

3. Switch Architecture Review

The ATM(asynchronous transfer mode) and SONET(Synchronous Optical Network) standard introduced the concept of a cell switching. Unlike a traditional circuit switched based network, a cell switching network can be viewed as a packet switching network where data and control information are chopped into fixed size packets. Many researchers have proposed various switching architectures for these new standards and architectures are generally categorized by two metrics: buffering techniques and topologies. Depending on where buffering is done, a switch is called an input, output, or input-output buffered switch. In general, an output buffered switch outperforms an input buffered switch[17]. However, output buffering requires larger internal bandwidth than input bandwidth to deliver multiple packets to the same destination, which leads to substantial increase in hardware complexity. Topology of a switch is another way to ramify many proposed switching architectures. In general, topology divides them into crossbar based switches, disjoint-path based switches, and Banyan based switches[17]. Disjoint-path based switches are built using large multiplexers, demultiplexers, and buffers. Obviously, those architectures are not a good candidate for a photonic datapath switch based on 2x2 switching element. Banyan based switches are further divided into a Benes network switch(a special type of Clos network switch) and a Batcher/Banyan network switch(sorting network). We will examine crossbar based switches, and two Banyan based switches in detail as a candidate for a optical datapath network switch.

a. Crossbar

A crossbar based switch has been a popular design because it is nonblocking and has a simple control. However, it has two main drawbacks. First of all, crosspoints complexity grows as $O(n^2)$ when n is the number of inputs. Considering each crosspoint is implemented with a 2x2 switching element, it is too expensive for a large n. Another drawback is that packets with different destinations will show different delay patterns through the switch. As n gets larger, the packet misalignment of optical signal though different free space distance can create another problem to deal with.

b. Benes

The next possible candidate is the Benes network switch. The Benes network has received much attention in interconnection network literature because of its O(n log n) hardware cost and O(log n) depth[1]-[4]. Most known sequential route assignment algorithms, such as the looping algorithm for Benes (1962) networks, are designed forcircuit switching systems where the switching configuration can be rearranged at a relatively low speed, O(n log n).

To realize optical packet switching on Benes network, it is natural to resort to a parallel algorithm that speeds up the routing decision by parallel computing on SIMD architecture. The best parallel algorithm[3][4] reported has a time complexity of $O(\log^2 n)$ where n is the number of inputs to a network and we assume SIMD controllers are fully connected. With an additional assumption of fully pipelined stages of network, which requires the controller cost of O(n log n)[4] rather than O(n), control timing cost decreases to O(log n). While this algorithm is relatively fast over any existing algorithms for a packet switching Benes network, total latency of the switching network is still $O(\log^2 n)$.

The possible improvement comes from recognizing the fact that the parallel algorithm by Oruc[4] allows a control complexity of O(log m) where m is the number of active input packets. This implies that we can take advantage of dynamic control latency when m << n (there are many empty packets to the input of the network switch). In fact, we can even achieve better control latency under full permutation inputs(k=m) i.e., no empty packets at the inputs by slightly changing the termination condition for the iteration. This idea will be discussed further by introducing Oruc's algorithm.

The parallel algorithm by Oruc[4] is based on the fact that configuration of switches for each stage can be done independently by only looking at inputs. That is, to satisfy the requirements for routing, the first log(n) stages only need to be set up in such a way that two inputs whose destinations are same in the first k bits are routed to upper and lower network in the next stage separately, where k is log(n) - stage number. The time complexity of $O(\log^2(n))$ of their algorithms comes from setting up the first log(n) stages which will be explained in detail later in this section. The last log(n)-1 stages are configured so that each switching element performs sorting of two inputs. Since sorting of two numbers only takes constant time, configuration of the last log(n)-1 stages takes only O(1). From this analysis, it is clear that improving algorithm for routing the first log(n) stages is the key to performance gain. We will explain how to achieve improvement in detail using examples.

Figure 6 shows a 16 input binary Benes network. Let us assume we have full permutation inputs such as example 1 and 2 in Table 2. Table 3 and 5 rearrange the entries of table 2 according to three rules. First, inputs with even destinations are put into the second row and odd ones into the third row. Secondly, two inputs connected to a same switching element are either located in two consecutive columns or in the same column and they are connected by either red circles(same rows) or lines(different rows). Finally, two entries of a same column in the second and third row differ by one bit in LSB position. This rearrangement directly shows how we can construct a chain when we suppose there is an imaginary connection between two entries in the same column. For instance, it can be easily seen that rearrangement of example 1 forms a chain, 11-10-2-3-8-9-13-12-7-6-1-0-4-5-14-14 while example 2 has two chains:4-5-3-2-13-12-11-10 and 1-0-9-8-14-15-7-6.

The main reason for identifying chains is that after forming a chain we can walk througha chain or chains in parallel and set up switching elements. Table 4 and 6 show the result of switch configuration by this method. Identifying and forming chains can be done in parallel at a cost of log(n) by pointer jumping method[5]. However, walking through a chain and setting up switches after forming a chain is highly sequential. For this reason, Oruc[4] forms two sub-chains for each chain: one chain connecting only inputs in the odd positions of a original chain and the other connecting inputs in the even positions while still keeping the complexity to O(log(n)). Since two chains are formed, when the formation is done each switch can automatically know how it should configure itself.

From this algorithm, we can realize that forming chains in example 2 will be faster than that in example 1. This is because forming two chains of length 8 in parallel is faster than one chain of length 16. In general, there can be many sub-chains formed depending on input patterns and the time taken for switch configuration is the time to form the longest sub-chain. For this reason, dynamic speedup can be obtained by recognizing that all the sub-chains are formed and stopping the iteration.

This new idea implicitly assumes that Benes network is able to deal with multiple latency control. That is, switches in each stage can be set up with different latencies according to their input patterns. This requires additional hardware so that necessary stages agree upon whether they can pass their packets to the next stage. The required rule is simple. Each stage can pass its packets to the next stage if all the later stages are ready to pass their packets to the next stages. Simple AND gates would be sufficient. The agreement is fed back to each stage involved so that each stage lets its packets go. Red boxes in Figure 6 indicate the places where "ready" signals in each stage are extracted and buffers for packets are located. Two boxes between stage 2 and 3 or between 3 and 4 can be omitted since Benes network for 4 and 2 inputs are same as Batcher/Banyan and a simple sorting algorithm can replace the parallel algorithm. With Batcher/Banyan algorithm, those two stages can be routed in minimum latency, which leads to no buffers.

c. Batcher/Banyan

The final candidate is Batcher/Banyan sorting network. This switching architecture has been quite popular due to its self-routing capability. Self-routing means that it takes only constant time, O(1), to set up a 2x2 switching element. Hardware cost, O(nlog^2(n)), and total latency, O(log^2(n)), are the major drawbacks. However, latency can be improved using multiple output port scheme, which will be discussed in the next section.

Port Num	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Example 1	10	2	3	8	9	13	12	7	6	1	0	4	5	15	14	11
Example 2	15	7	12	11	3	5	9	7	2	13	14	8	6	1	10	15

Table 2

Port Num	0	1	3	6	8	10	11	14

even destination	10	2	8	12	6	0	4	14
odd destination	11	3	9	13	7	1	5	15
Port Num	15	2	4	5	7	9	12	13

Table 3. Example 1

Port Num	0	1	3	6	8	10	11	14
even destination	0	1	1	0	0	0	1	0
odd destination	1	0	0	1	1	1	0	1
Port Num	15	2	4	5	7	9	12	13

Table 4. Switching element setup for Example 1: 0=upper network, 1=lower network

Port Num	15	8	2	14	7	11	10	12
even destination	4	2	12	10	0	8	14	6
odd destination	5	3	13	11	1	9	15	7
Port Num	5	4	9	3	13	6	0	1

Table 5. Example 2

Port Num	15	8	2	14	7	11	10	12
even destination	1	0	0	0	0	0	1	0
odd destination	0	1	1	1	1	1	0	1
Port Num	5	4	9	3	13	6	0	1

Table 6. Switching element setup for Example 2: 0=upper network, 1=lower network

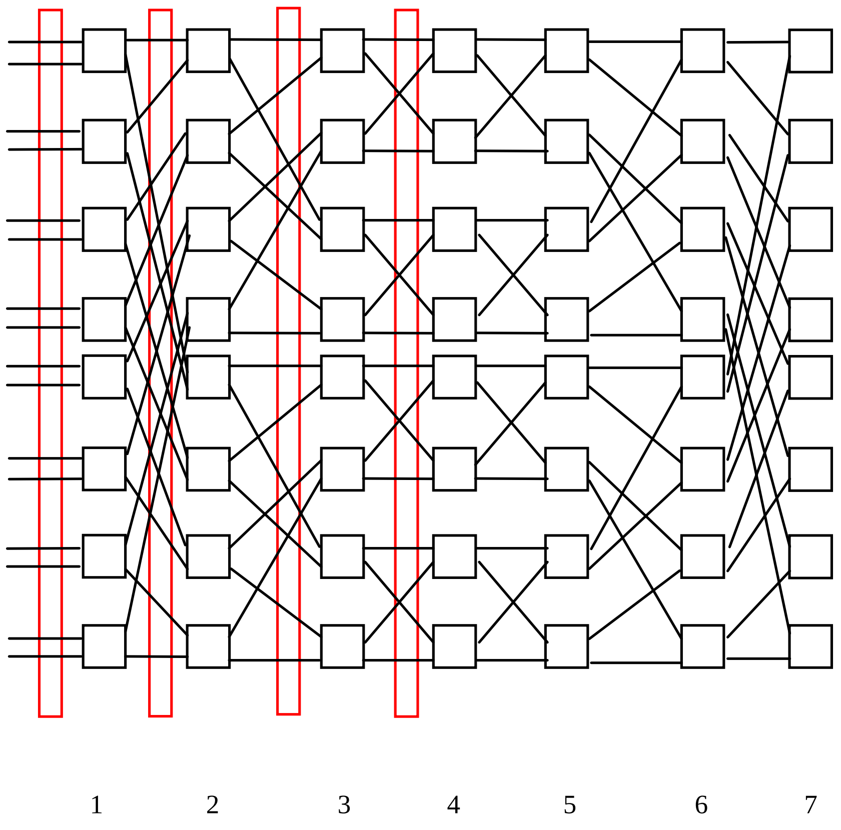


Figure 6. Benes Network(input=16)

4. Multiple output port optical network switch

Figure 7 shows a proposed multiple output port photonic network switch using MQW modulators (a) and Sagnac gates (b).

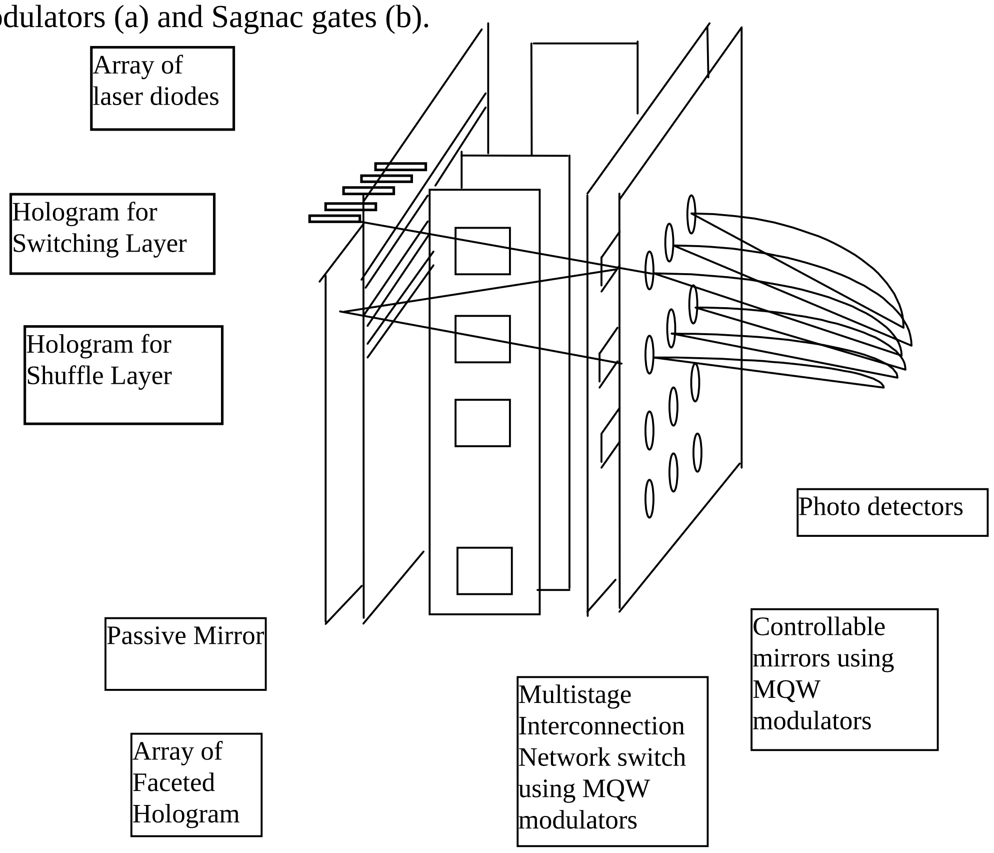

Array of laser diodes
Hologram for Switching Layer
Hologram for Shuffle Layer
Photo detectors
Passive Mirror
Controllable mirrors using MQW modulators
Multistage Interconnection Network switch using MQW modulators
Array of Faceted Hologram

Figure 7 (a)

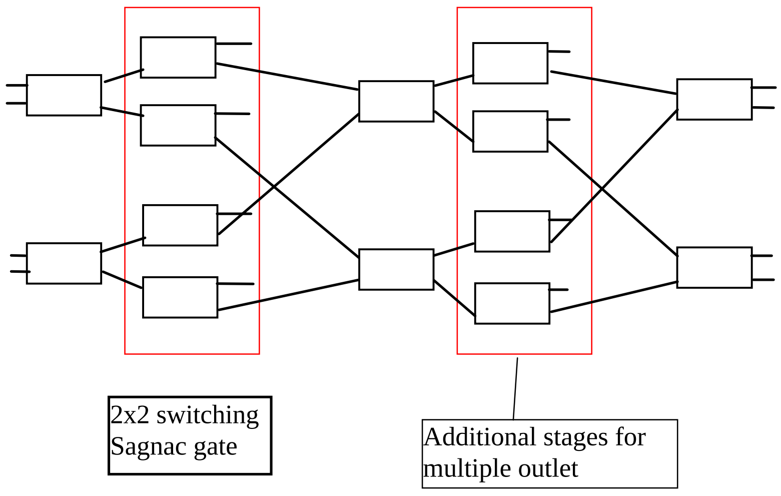
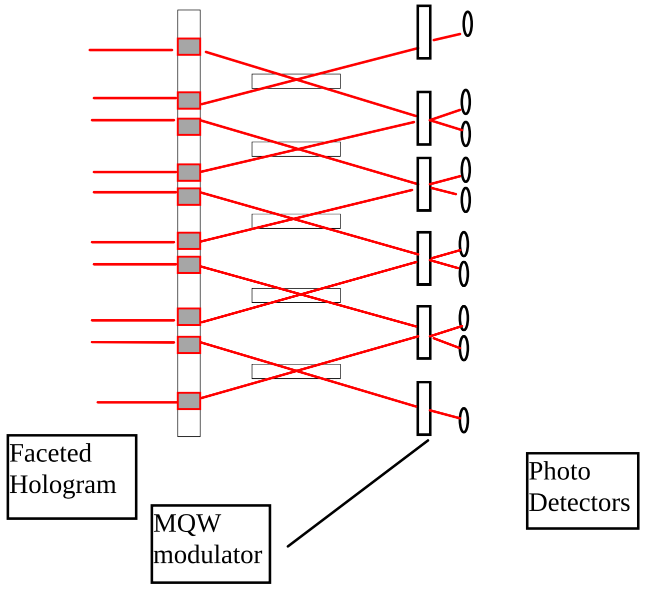
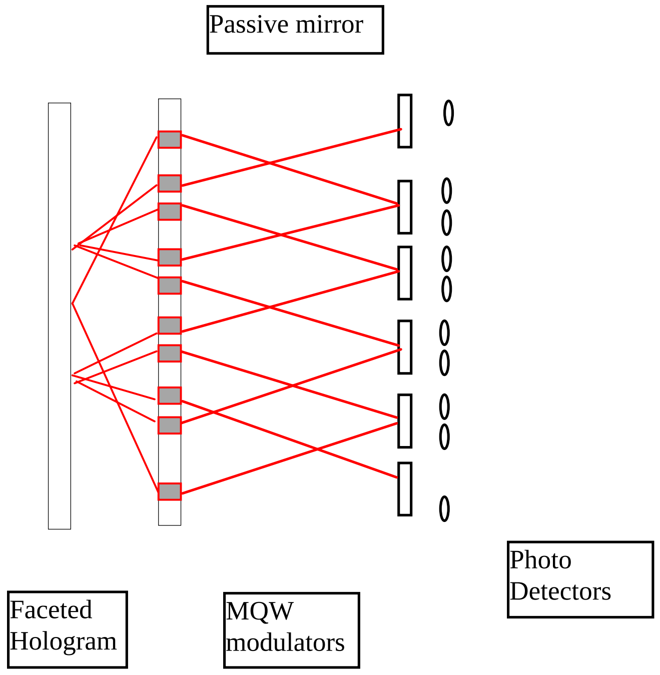

2x2 switching Sagnac gate
Additional stages for multiple outlet

Figure 7 (b)



(a) Switching Layer



(b) Shuffle Layer

Figure 8

MQW modulator implementation of a multiple latency network switch is composed of 6 components. The first is an array of laser diodes performing electrical-to-optical signal conversion and injecting parallel optical signal into a multiple latency network switch. These optical parallel bit streams are refracted by the second, faceted holograms, and modulated by the third component, columns of MQW modulators. The faceted holograms have two alternating rows. One row in the first top view, figure 8(a), corrects the direction of optical signal so that it is incident on the MQW modulators with a correct angle. The other row in the second top view, figure 8(b), realizes shuffle connection required in each stage of the network. MQW modulators perform 2x2 switching. The fourth component is a plane of 2 dimensional arrays of MQW modulators. In reflection mode(set through) of MQW modulators, optical signal is redirected to the next stage of a switching network. In transparent mode(cross), optical signal is collected by the fifth component, arrays of photo detectors. That is, if a requested permutation of packets is realized in several hops, the corresponding row of fourth components are set to a transparent mode and packets are converted to electrical signal by photo detectors. Otherwise, optical signal is reflected by MWQ modulators and processed further in the next stage. The last component is a passive mirror that is a part of shuffle layers. Implementation with Sagnac gates is straightforward. Figure 7(b) is 4x4 Batcher/Banyan or Benes network using Sagnac gates. Both implementation doesn't't require any topological assumptions and can be applied to any topology.

Although Batcher/Banyan is far from optimal in term of hardware cost and maximum latency compared to a theoretical bound, it is still interesting to apply the multiple latency technique to Batcher/Banyan network for several reasons. First, it can be used as a baseline model. Secondly, the real performance of a switch combined with a smart scheduler or a parallel compiler in multiprocessor interconnection and a smart routing scheme in WAN would give a shorter average latency.

The natural points where we can tap the outputs from Batcher/Banyan network are the ends of stage 1, 3, 6, ...logN(logN+1)/2 where N is the number of inputs to a network. It can be easily seen that $(2!)^{N/2}$, $(4!)^{N/4}$, $(8!)^{N/8}$,....,N! input patterns can be routed at tapping points, 1,3,6,... logN(logN+1)/2. For instance, a permutation 43215867 can be fully routed by the end of stage 3 in figure 9. However, the permutation output of this stage would be 12348765 since this permutation is still in the middle of bitonic sorting. But, this problem can be simply corrected without changing controlling algorithms by designing external connections from photo detectors properly. External connections for 8 input Batcher/Banyan network are shown in the bottom part of figure 9. Decisions on whether some permutation is fully routed at a certain tapping point can be made by looking at most significant logN-i bits where $i \in \{ 1,2,3..\log N \}$ stands for the i th tapping point. This additional hardware doesn't't affect the performance since a latency of several AND gates chain is shorter than a latency of a comparator used for a sorting decision. Finally, an output contention problem due to multiple latency routing can be resolved in several ways. The simplest method is giving higher priority to permutation outputs with longer latencies. Permutations with shorter latencies can be further routed to the next tapping point since further routing wouldn't't change the permutation. Another method can be maintaining a set of external buffers, optical or electrical, and making a decision based on the given priorities of packets .

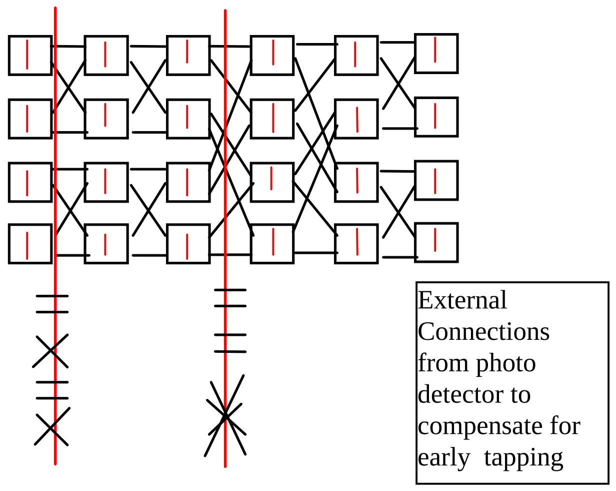


Figure 9. 8x8 Batcher/Banyan network

The major advantage of multiple output ports is that routing algorithms or scheduling algorithms can make use of multiple latency information in various levels of networking. In a multiprocessor interconnection network, a process scheduler and a parallel compiler can use this information in such a way that communication penalty between processorsdoesn't degrade the performance gain from parallel processing by allocating processes to processors physically connected with small latency or selecting the proper number of forked processes. Also, in LAN and WAN routing where usually several alternative paths are kept in a routing table for deflection routing, routing decisions are made such that packets are routed as quickly as possible while avoiding output contention without extra buffers. Figure 10 shows a binary radix sorter network has a depth of $\log(n)(\log^2(n)+5)/6$ which is worse than Batcher/Banyan, $\log(n)(\log(n)+1)/2$, in higher number of n. However, this network can be routed by just looking at 1 bit in each stage while Batcher/Banyan needs a full comparison of headers for two inputs.

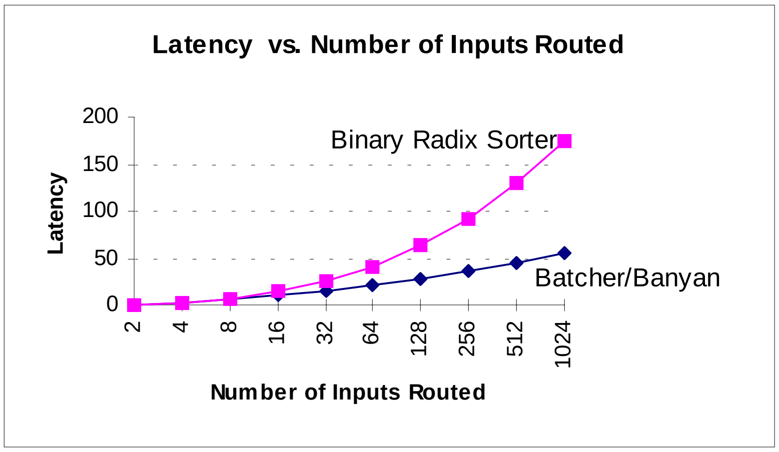


Figure 10

5. Interpretation of Permutation using Symmetric Group, Cayley graph, and Cayley coset graph

Given a set of generators for a finite group G, we can draw a graph, called Cayley graph, in which vertices represent the elements of the group G and the edges represent the action of the generators. That is, there is an edge from an element a to an element b iff there is a generator g such that ag=b in the group G.
A symmetric group, usually denoted as S_n, is composed of n! elements that correspond to all the permutations of n symbols. For instance, S_3 is composed of 123, 132, 213,231,312, and 321. It can be easily seen that 132 and 213 can also represent generators and they are two generators enough to generate the whole group. In fact, there are infinitely many sets of generators that can generate the same group. Their difference is interpreted as different numbers of edges connecting each elements in a corresponding Cayley graph and their connectivity.

Figure 11(a) shows a Cayley graph of S_4 generated with a set of generator a, b, and c. a is 2134, b is 1324, and c is 1243. Routing 4 packets destined to 4 different destinations in network is equivalent to finding a path from any vertex in a Cayley graph for S_4 to a vertex representing an identity element, 1234, in a symmetric group.

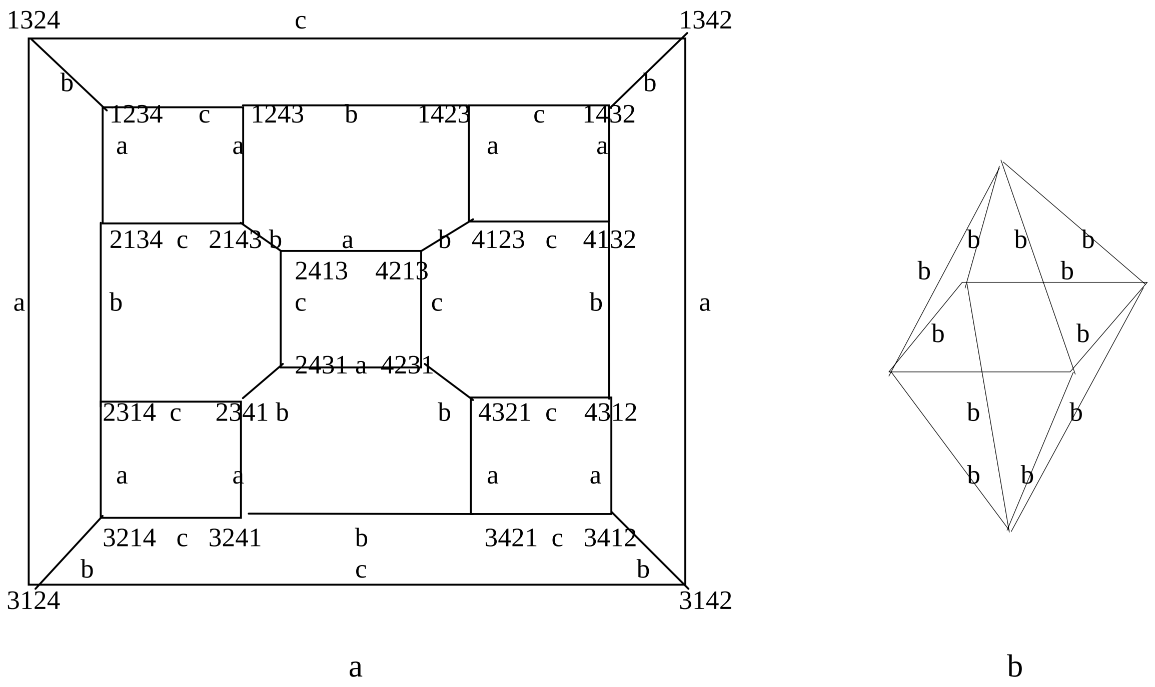


Figure 11

In general, Cayley graphs of a symmetric group with n symbols have n! vertices. The number of edges coming out of vertices is determined by a set of generators that generate the group. There are a few interesting Cayley graphs for a symmetric group reported in papers[14]-[16]. The one we are interested in is a Cayley graph called bubblesort graph. A generator set for this graph is defined as $\{(I,I+1) \mid 0 \leq I \leq n-1\}$ where n is the number of symbols in a symmetric group and (a,b) stands for exchanging two symbols located in ath and bth place. Figure 11(a) is a bubblesort graph with n=4. There are several reasons why we focus on these bubblesort graphs in designing a multistage interconnection network. First, a bubblesort graph with larger number of symbols can be constructed from one with smaller number of symbols in a systematic way. Figure 12 shows how to construct these graphs from smaller ones.

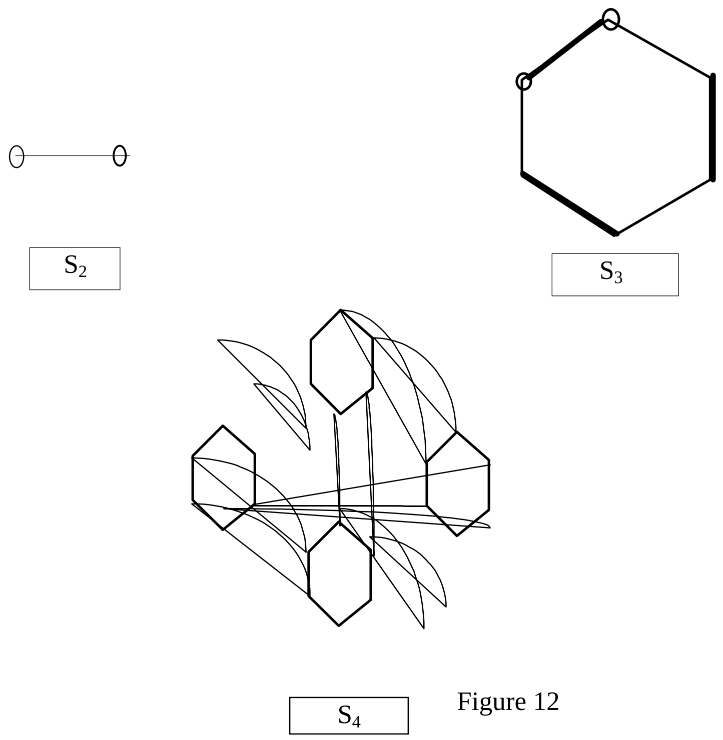


Figure 12

Secondly, complexity of a bubblesort Cayley graph can be significantly reduced by means of coset decomposition of a symmetric group. The reduced Cayley graph generated by coset decomposition is a Cayley graph of the subgroup of a symmetric group and called a Cayley coset graph. An example of Cayley coset graph for S_4 is drawn in figure 11(b). This graph is actually obtained by removing all the edges labeled a and c from figure 11(a). The effect of coset decomposition with respect to a certain generator is removing edges of that generator from the Cayley graph while maintaining connectivity. Then, why this reduced Cayley coset graph is important? Figure 13 shows 4 input Batcher/Banyan or Benes network. It can be easily seen that three generators, (12),(23), and (34) for a 4 symbol bubblesort graph are enough to express the functionality of each switching stage of figure 13. Stage I,III and V can be expressed in terms of (12) and (34). Operations of Stage II and IV are equivalent to (23).

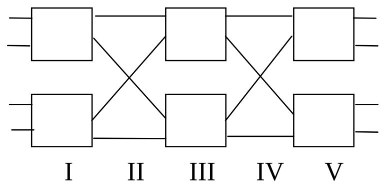


Figure 13

Interpretation of switchable elements in stage I, III, and V is that we can freely move around vertices through edges labeled a and c in the Cayley graph for S_4. For instance, if output permutation of stage II is 2134 in figure 13, new output of III can be 2134, 1234, 1243, and 2143 by setting the switches in stage III properly. This capability of transition to reachable vertices after fixed switching in II and IV enable us to remove the effect of (12) and (34) from the Cayley graph. As a result, we can explain switching operation of 4x4 switch in figure 13 using Cayley coset graph in figure 11(b) which contains only generator (23). Reasoning for requiring two stages in 4x4 switch is that any arbitrary vertex in Figure 11(b) requires two hops to reach a certain vertex.

In general, a coset graph can be used in several ways to build a multistage interconnection network. One is to find the small number of hops in which one can visit from identity to all the elements in the coset graph by choosing a shuffle pattern for each stage. In this process, a coset graph can be even decomposed into another coset graph recursively to find simple control algorithms or to simply the coset graph more. Since the smallest number of hops doesn't give the simplest control algorithm, various configurations should be tested to find good balance between latency and control. Another way to use this coset graph is that find the number of hops as we did in the first method but in an accumulative way. That is, after we construct several stages of the networks and visit some intermediate nodes, we only need to visit those nodes that were not visited yet. This method is actually constructing a multiple input port network similar to one we proposed in section 4. If we find a sequence of generators that have an identical inverse, this method can be used to build a multiple output port network as well. In fact, Benes network is the example.

6. Summary

In this report, we have reviewed several switch architectures. For a small number of inputs to the network switch, the crossbar is a good solution for all optical datapath network due to its simple control and reasonable hardware cost. For a large number of inputs, however, crossbar becomes too expensive and other architectures should be considered. The dynamic control latency Benes network switch and the multiple output port Batcher/Banyan network switch could meet the necessities of all optical datapath network. Finally, connection between group theory and multiple interconnection network(MIN), which was introduced in this report, is expected to become new tools to find a optimal solution for MIN in terms of control and hardware complexity.

7. References

[1] V. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*. New York: Academic, 1965

[2] D. Nassimi and S. Sahni, “A Self-Routing Benes Network and Parallel Permutation Algorithms, IEEE Trans. Comput., pp.332-340, May 1981.

[3] D. Nassimi and S. Sahni, “Parallel algorithms to set up the Benes network”, IEEE Trans. Comput.,pp.148-154, Feb. 1982

[4] C.Y. Lee and A.Y. Oruc, “A fast parallel algorithm for routing unicast assignment in Benes network”, IEEE Trans. Parallel and Distributed Systems, pp.329-334, Mar. 1995

[5] T. Cormen, C. E. Lesiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 1990, pp. 692-701.

[6] J.A. Trezza, M.Morf, and J.S. Harris, Jr., “Creation and Optimization of Vertical-Cavity X-Modulators”, IEEE Journal of Quantum Electronics, pp.53-60, Jan. 1996

[7] A. Huang *et al.*, “Sagnac fiber logic gates and their possible applications: a system perspective”, Applied Optics, pp.6254-6267, Sept. 1994

[8] K. K. Ng, *Complete guide to semiconductor devices*, New York, NY: McGraw-Hill, Inc., 1995

[9] A.M. Yurek *et al.*, “Commercial $LiNbO_3$ Integrated Optic Devices”, Optics & Photonic News, pp.26-30, June 1995.

[10] H.S. Hinton, *An Introduction to Photonic Switching Fabrics*, New York, NY: Plenum Press, 1993.

[11] K.E. Batcher, “Sorting networks and their applications” in 1968 Spring Joint Computer Conference , AFIPS Proc., vol. 32, pp.307-314.

[12] M. J. Narashimha, “The Batcher-Banyan Self-Routing Network: Universality and Simplification”, IEEE Trans. Communications, pp.1175-1178, Oct. 1988.

[13] I. Grossman and W. Magnus, Groups and their graphs, Washington, Washington DC: Mathematical association of America, 1992.

[14] M. Ramas, “Routing Permutation on a Graph”, Networks, pp. 391-398, Vol.23, 1993.

[15] F. Annexstein *et al.*, “Group Action Graphs and Parallel Architectures”, SIAM J. Comput., pp.544-569, 1990

[16] J. Huang, *et al.*, “Analysis of Interconnection Networks based on Simple Cayley Coset Graphs”, Proceedings of 1993 5th IEEE Symposium on Parallel and Distributed Processing, pp.150-157, 1993

[17] R. Y. Awdeh and H.T. Mouftah, “Survey of ATM switch architectures”, Computer Networks and ISDN Systems, pp. 47-93, 1994