Optical Design of a Fault Tolerant Self-Routing Switch for Massively Parallel Processing Networks

M. Guizani
mohsen@ccse.kfupm.edu.sa
Computer Engg. Dept.
KFUPM Box No. 1969,
Dhahran 31261, KSA

M. A. Memon
atif@ccse.kfupm.edu.sa
Computer Science Dept.
KFUPM Box No. 1009,
Dhahran 31261, KSA

S. Ghanta
ghanta@ccse.kfupm.edu.sa
Computer Science Dept.
KFUPM Box No. 1754,
Dhahran 31261, KSA

Abstract

A fault-tolerant high-performance optical 2 × 2 switch is proposed for massively parallel processing network applications. The switch is designed using all optical components. This allows the exploitation of spatial parallelism. The proposed switch can be used with any multistage interconnection network such as omega, Banyan, shuffle, Benes. For the purpose of this study, the baseline network was used as the underlying network. The reasons for choosing the baseline are that it uses 2 × 2 switches, is not fault-tolerant and is self-routing. Performance/reliability analysis is done and compared with two major fault-tolerant networks. The results show that without redundant switches, better network survivability is achieved. Since the network is assembled using only the 2 × 2 switches, it can be mass produced.

1 Introduction

Due to the increased demand for communication services of all kinds (voice, data, and video), broadband integrated services digital network (B-ISDN) has received increased attention in the past few years. The ability of B-ISDN system to support a wide variety of traffic and diverse services is the key for its success. It is required to support traffic with:

- bandwidth ranging from few Kbps (terminals) to several Mbps (moving image data);
- highly bursty traffic (interactive data and video) and continuous traffic (transfer of large files);
- multimedia traffic, such as real time voice (where small errors are tolerable), but real-time video communications require error-free transmissions as well as rapid transfer.
- Any future services such as high definition TV (HDTV), broadband videotex, and video/document retrieval services [2, 3, 4].

The heterogeneity in the requirements of these different services requires a high speed and robust switching technology. The asynchronous transfer mode (ATM) is the transport technique for B-ISDN recommended by CCITT [5]. Many switches have been proposed to accommodate the ATM, which requires fast packet (cell) switching[6, 7, 8].

Massively parallel processing applications executing on multiprocessor systems require a robust interconnection network [1, 9]. This is because of their need to exchange large volumes of data. These could either be in the form of frequent short bursts or large continuous streams.

Several implementations of fault tolerant interconnection networks can be found in the literature [10, 12, 13]. Almost all of these introduce redundancy in the network in terms of adding extra links and switches. These solutions are expensive since in most cases they increase the number of links [12].

Optical interconnections offer a high communication bandwidth. Using conventional fault tolerant schemes, extra links and switches must be introduced in the network. Under normal conditions these are hardly ever used. However, once there is a fault in the network (e.g. a faulty switch), these links/switches completely take over the operations. Once a cell/switch is detected as faulty, its incoming and outgoing links become useless. This results in wastage of bandwidth. In massively parallel processing applications, this is highly undesirable.

There is a need for switches that are capable of routing unprocessed data to the next stage even in case of failure. The next stage regular/normal switch(es) should be able to process this data. This eliminates the need for extra switches. Moreover, no extra links are used to route the unprocessed data to the next stage.

In the next section, the design and operation of the proposed cell is presented. Section 3 describes the bypass switch and its operation. In section 4, detailed design and control algorithm for error detection and correction circuit are given. The main module, the routing switch, is described in section 5. Reliability analysis with discussion of the results is given in section 6.

0-8186-7101-7/95 $4.00

2 Design and Operation of the Cell

The top level breakdown of the 2×2 switch is shown in Figure 1.

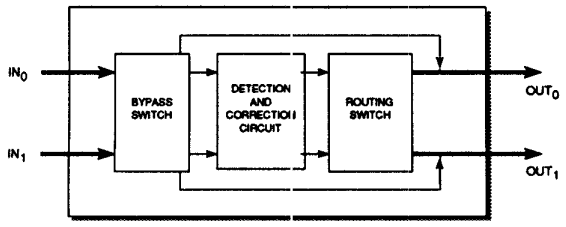

Figure 1: Block diagram of the 2 × 2 cell.

It consists of three main circuits.

1. Bypassswitch
2. Error detection and correction circuit
3. Routing switch

The first two are useful for handling faults in the network. If the routing switch of a stage i fails to respond correctly, its bypass switch routes the data to the next stage $i + 1$. At this stage, routing decisions of the data are taken.

3 The Bypass Switch

This switch takes over the operations whenever the routing switch (within the same cell) fails. Due to its simple design, it does not perform complex operations on the incoming data. The main purpose of this switch is to channel incoming data to the next stage of the network. Incoming packets are channeled to all possible outputs (in this case two) of the cell. This duplication temporarily creates extra traffic in the network. It is up to the cell in the next stage to determine whether the incoming data packet is indeed destined to it.

To enable successive stages to correctly detect and remove data duplication in the network, an extra field of *error bits* is introduced by the bypass switch. To minimize complexity in the bypass switch hardware, 0 (1) is appended to the incoming packet depending on whether it is routed to the upper (lower) link.

An example for comparing the output of a faulty cell to a properly working cell is shown in Figure 2. In part (a) of this figure, the data is routed to the lower output link based on the routing bits (destination address). Note that there are no bits in the error field and the most significant bit (1) is dropped off from the routing bits. Part (b) of Figure 2 illustrates the operation of a faulty cell. Note that the input is replicated to both high and low outputs of the cell with the error field containing 0 and 1, respectively. The routing bits remain unchanged.

The component level schematic of the bypass switch is shown in Figure 3. Under normal operation, the

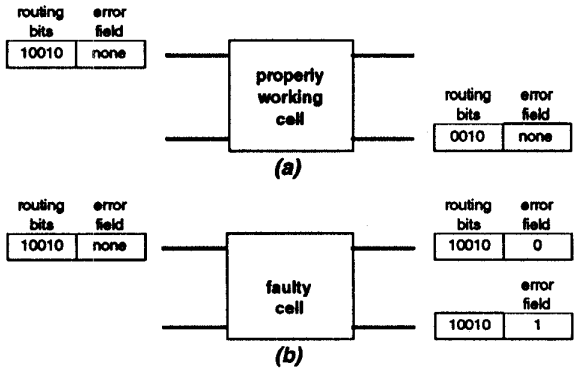

Figure 2: Comparison between traffic generated by a faulty and a properly working cell.

E flag (error indication) is reset allowing both inputs (IN_0 and IN_1) to appear directly at the outputs (OUT_0 and OUT_1). If the routing switch (in the same cell) detects an error in its operations (details in Section 5), it sets the E flag which allows bypassing the data to the next stage through OUT_{E0} and OUT_{E1} appending 0 and 1 to the error field, respectively. C is a flag that is used to broadcast the upper (lower) input to OUT_{E0} and OUT_{E1} when $C = 0$ ($C = 1$).

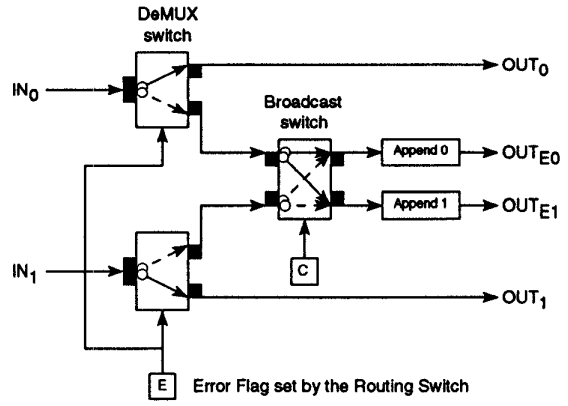

Figure 3: Schematic of the bypass switch

The control algorithm of the bypass switch is as follows:

```
Component BypassSwitch(IN0,  IN1,
                       OUT0, OUT1,
                       OUTE0, OUTE1)
```

```
If (E = 0)
   t1 = IN0
   t2 = IN1
   With (C = 0)
         OUTE0 <- Append[t1, 0]
         OUTE1 <- Append[t1, 1]
   With (C = 1)
         OUTE0 <- Append[t2, 0]
         OUTE1 <- Append[t2, 1]
else (* E = 1 *)
   OUT0 <- IN0
   OUT1 <- IN1
END;
```

The bypass switch is constructed from two components. The first is the broadcast switch which allows data to be directed from any one of the inputs to both the outputs depending on the control signal. Figure 4 shows its block diagram and control algorithm. The second component is the demultiplexing switch which directs data from an input to either of the two outputs. Figure 5 shows its block diagram and control algorithm.

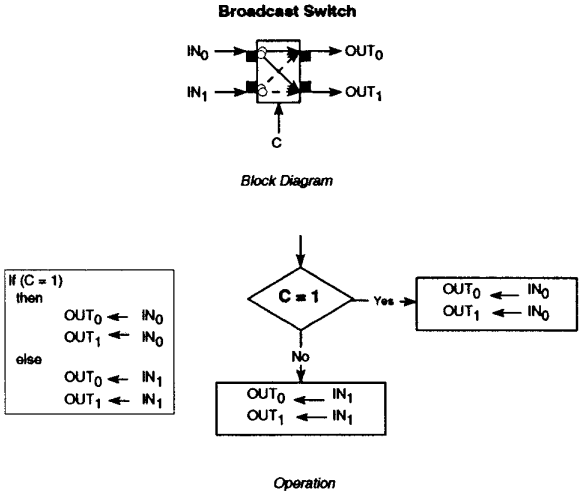

Figure 4: Working of the broadcast switch

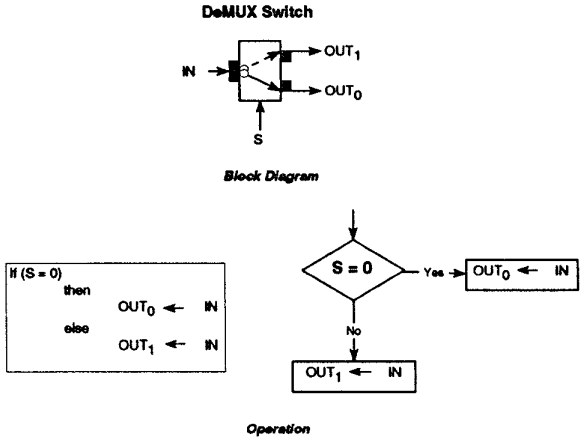

Figure 5: Working of the demultiplexer

4 The Error Detection and Correction Circuit

In Figure 6, F_1 represents a faulty cell. Therefore, its bypass switch sends the data packet to both output links connected to W_1 and W_2 that represent working cells in the next stage. Note that the error field of the packet contains only one bit since it encountered only one faulty cell. At W_1 and W_2 stage, the extra traffic created due to F_1 needs to be detected and removed.

On receiving both data packets, W_1 and W_2 strip off the error field bits and an equal number of routing bits and compare them. Equivalence of these bits validates presense of the data packet at that cell, otherwise it is rejected. Consequently, in Figure 6, W_1 rejects the first packet and accepts the second while W_2 accepts the first packet and rejects the second. Since both of these cells are working, the data is routed to the correct destination despite the presence of a faulty cell in the network. This can be achieved even in the presence of $n-1$ faulty cells, where n is the network size (number of stages).

Figure 7 shows the schematic diagram of the error detection and correction circuit. This circuit is capable of detecting and correcting any number of errors in the network due to the presence of faulty cells. This is achieved by examining the error field in the recieved data packet. The circuit is composed of three components. The first is an optical routing device which splits the input beam into three outputs. The second is the equivalence gate (see Figure 8) which checks whether the two received inputs are the same. If they are, it sets a signal to load a latch. If they are not, it does not load the latch and therefore it discards the packet. The third is the latch (see Figure 9) which is loaded to send the data out to the queue of the routing switch.

The bypass switch of each faulty cell appends one bit to the error field. Hence the size of the error field is exactly equal to the number of faulty cells encountered during the routing. The number of bits to be compared can vary from one cell to the other. Therefore, there is a need for a hardware that can adapt to these variations. This can be done electronically by using a serial circuit in which bits are recirculated and compared. This is an expensive operation as it takes time proportional to the number of bits in the error field. However, use of optical devices in designing such circuits reduces both the time and design complexity. Optics employs spatial parallelism which allows comparision of multiple bits in constant time. Also,

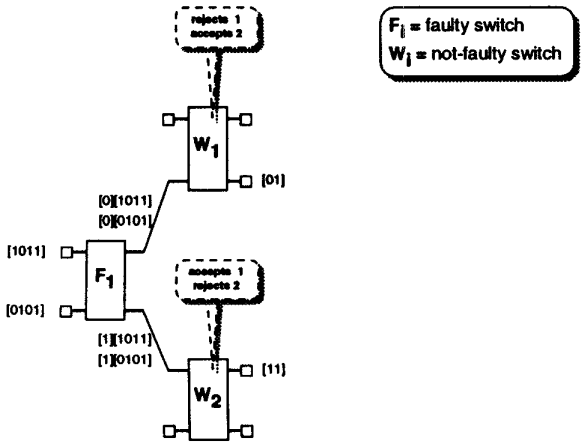

Figure 6: Example of one faulty cell, redundant data and error recovery.

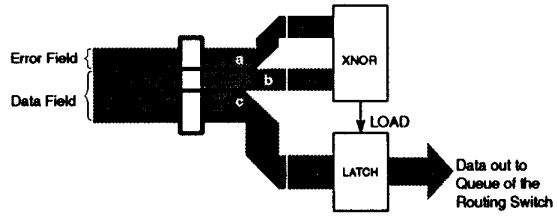

Figure 7: The network error detection and correction circuit.

designing an n bit comparator has the same design complexity as designing a one bit comparator [11].

5 The Routing Switch

Figure 10 shows the routing switch which represents the main module of the cell. It is responsible for routing the input data to the next stage of the network. In the presence of a contention, one of the input packets is chosen randomly, if no priority is assigned, to propagate to the next stage. The other input packet is recirculated to the contention controller (CC). The CC attaches a priority to the recirculating packet and submits it again as an input. The more a packet is recirculated, the higher priority it holds so that it has greater chance of being transmitted to the next stage. The routing algorithm is described next.

```
Component RoutingSwitch(IN0,  IN1,
                        OUT0, OUT1,
                        EF)
      On (IN0)
          Insert(Q0, {IN0, 1})
```

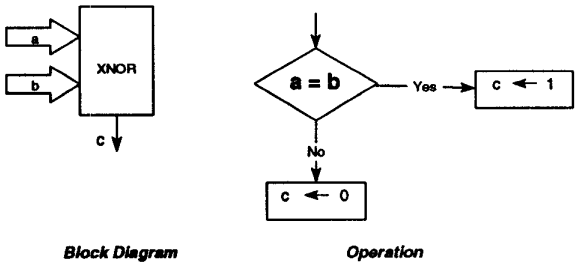

Figure 8: Working of the equivalence gate

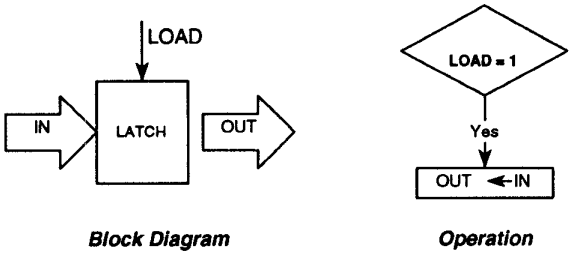

Figure 9: Working of the latch

```
      On (IN1)
          Insert(Q1, {IN1, 1})

      {t0, p0} = RemoveMaxPriority(Q0)
      {t1, p1} = RemoveMaxPriority(Q1)

      {t00, t01} = IC0({t0, p0}}
      {t10, t11} = IC1({t00, t01})

      {OUT0, OUTCC0} = DC0(t00, t10)
      {OUT1, OUTCC1} = DC1(t01, t11)

      {INC0, INC1} = CC(OUTCC0, OUTCC1)

      EF = CCK(F0, F1)

END;
```

The routing switch is composed of five different parts. The following is a brief description of each part with the control algorithm.

Each input port of the switch requires a limited buffer size. This is shown in Figure 11. Each queue has one external and one internal input. The external input, shown as IN_0, is the output of the previous stage. The internal input, such as IN_{CC}, is coming

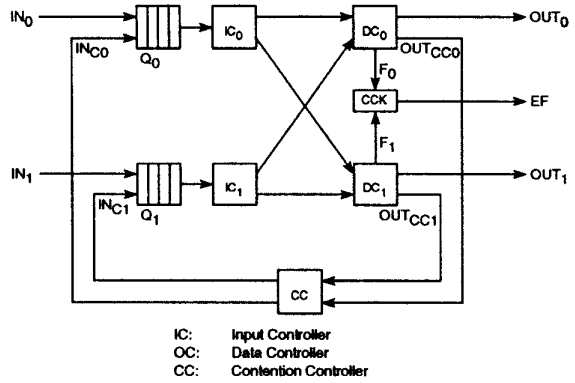

Figure 10: The main routing switch

from the CC. Under normal operation, we assume a FIFO queue. But when an input is coming from the CC, it has been assigned a priority number so that it gives the recirculated data packet the opportunity to be routed in the next available chance. The output of this buffer (Q_i) is the input to the next module denoted by IC_i.

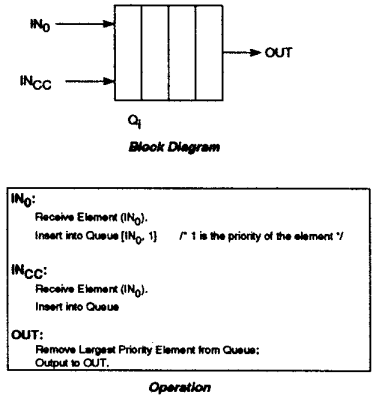

Figure 11: Working of the Queue

The next part is IC_i, which accepts an input from the buffer and provides two outputs. If the MSB is 0, then OUT_{c0} is forwarded as input to the next module DC_0. If the MSB is 1, then OUT_{c1} is forwarded as input to the next module DC_1. The block diagram describing this module and its algorithm are shown in Figure 12.

Figure 13 describes the block diagram and the operation of the third module, the DC_i. This module has two inputs from IC_i (that is IN_{C0} and IN_{C1}), and three outputs OUT_0, OUT_{CC}, and F. If IN_{C0} is the

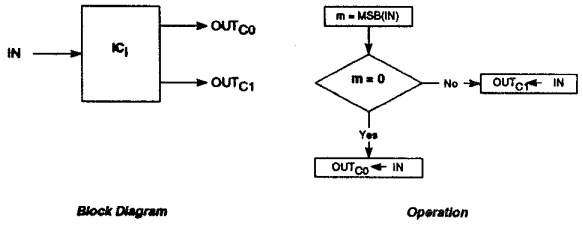

Figure 12: Working of IC_i

only input to this module, then OUT_0 is simply IN_{C0} stripping its MSB. If IN_{C1} is the only input, then it appears at the output OUT_0 with its MSB stripped. If both IN_{C0} and IN_{C1} exist simultaneously, then the input with the higher priority appears at OUT_0 with its MSB stripped, and that with the lower priority appears at OUT_{CC1} with its MSB not stripped. In order to detect an error, and therefore conclude that the switch is not working properly, DC_0 (DC_1) must receive the routing bits with $MSB = 1$ $(MSB = 0)$. This situation sets $F = 1$ which indicates that the main routing switch is in error, and therefore activate the bypass switch.

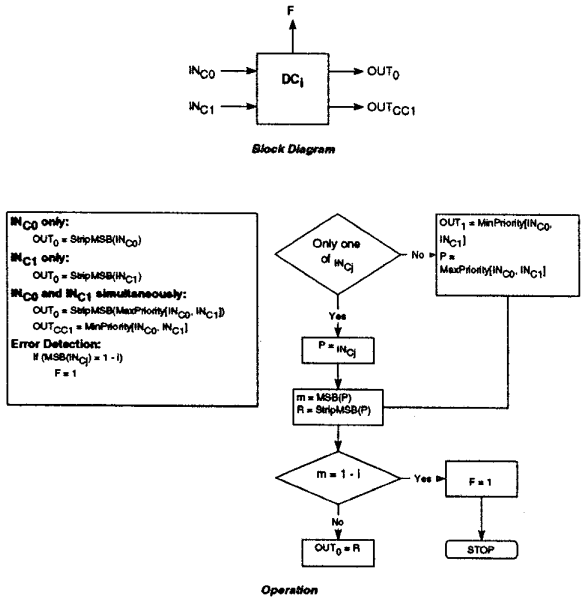

Figure 13: Operation of DC_i

Figure 14 shows the block diagram and the operation of the CC. It has two inputs and two outputs. This module increases the priority of the data packet

each time it passes through it. This process enables the data packets that were not able to pass to the next stage in the previous trial to be routed.

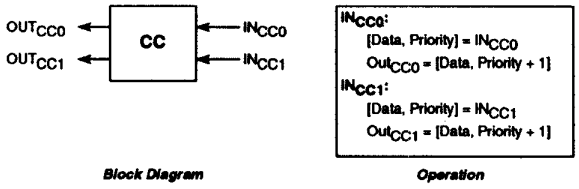

Figure 14: Working of CC

Figure 15 describes the block diagram and the operation of the CCK. It has two inputs, F_0 and F_1, and only one output EF. The output is just the OR operation of both inputs.

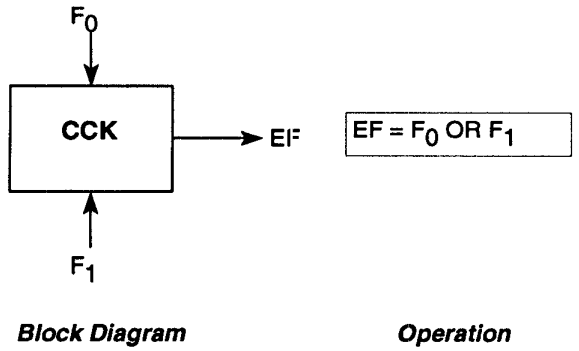

Figure 15: Working of The Circuit Error Reporter

6 Reliability Analyses

The schematic shown in Figure 7 is capable of detecting and correcting multiple successive faults. Figure 16 shows some working and faulty cells of a network. This example shows that data is correctly routed in the presence of three successive faults and absence of redundant switches/links.

The assumptions made in carrying out this analysis are: First, the failure of one switch can in no way affect the reliability of any other switch in the network. Second, the network is said to have failed if at least one connection between input and output ports can not be realized. Most of other reliability analyses given in the literature [12, 14] assume the first and last stages of the network are fully operational under all conditions. This is not a realistic assumption since all the switches in the network are equally likely to fail. Therefore, in this analysis, all switches have equal probability of

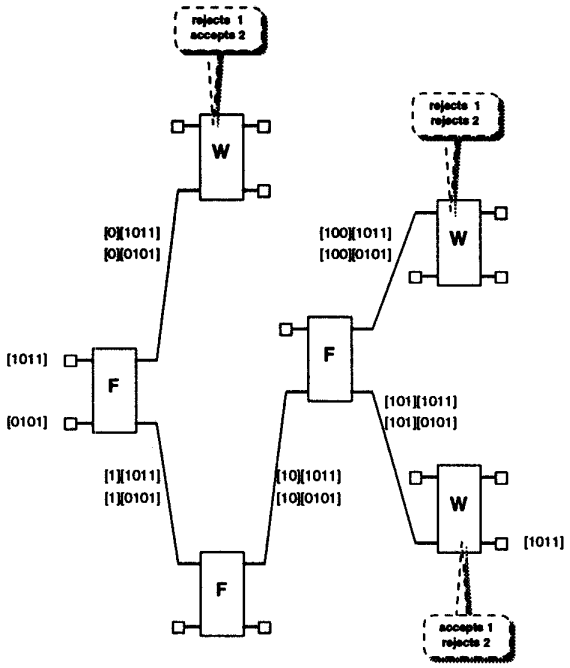

Figure 16: Example of successive faulty cells, redundant data and error recovery.

failure, that is including all switches of the first and last stages.

Figures 17 and 18 show the results of both Itoh's and Benes' networks using the above assumptions. These results show that the survival probability $Q(k)$ is very low (in the orders of ≤ 0.2).

Figure 19 shows the survival probability $Q(k)$ of the proposed network. It is clear that it performs better for the same range of faults in addition to having lower slope of the corresponding curves. Note that in the proposed network, no additional switches/links are required. The mathematical formulation of the probability $Q(k)$ is given by

$Q(k)$ = survival probability of the network.

Total number of switches = $n2^{(n-1)}$

Any of these switches can fail.

k failures $(0 \leq k \leq n2^{(n-1)})$

Number of configurations in which k failures can occur = $\binom{n2^{(n-1)}}{k}$

$$Q(k) = \frac{\binom{n2^{(n-1)}}{k}}{\binom{(n-1)2^{(n-1)}}{k}}$$

Figure 20 compares the total number of switches needed to achieve the reliability for all three cases. It is clear that this number is doubled in the case of Itoh's

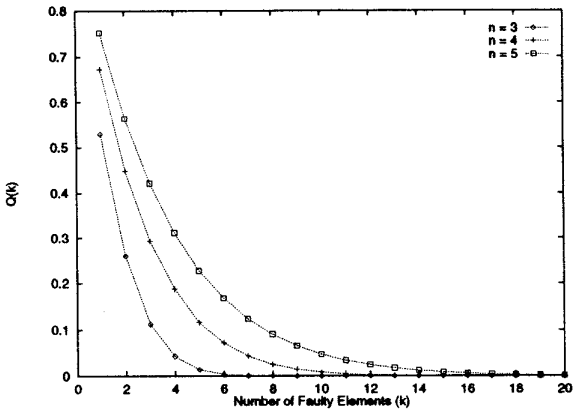

Figure 17: Itoh's Network.

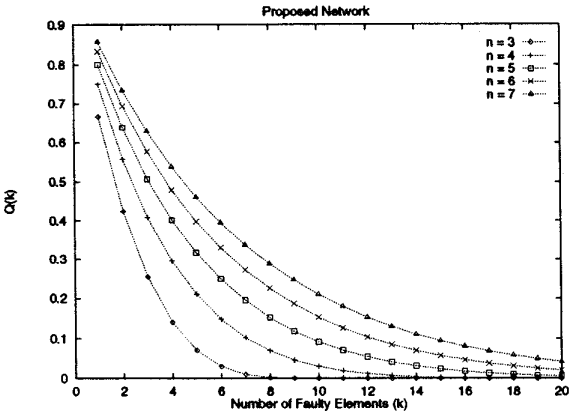

Figure 19: Proposed Network.

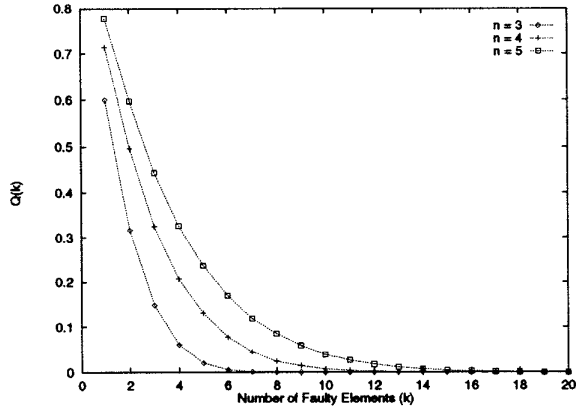

Figure 18: Benes Network.

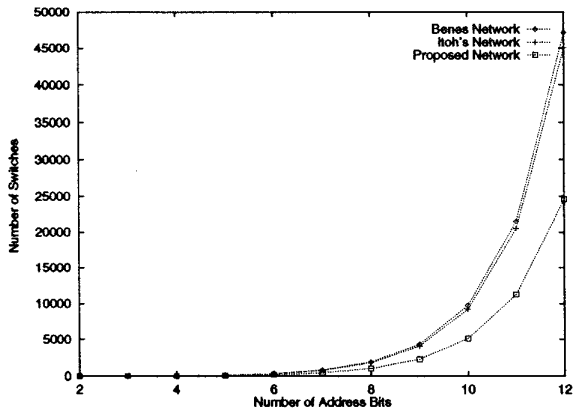

Figure 20: Number of Switches.

and Benes' networks. This reduction of the number of switches in the proposed network is desirable since it reduces the cost and complexity.

Figure 21 shows that the number of redundant paths in the proposed network is similar to that of the Benes network and much less than Itoh's network.

7 Conclusion

In this paper, we have proposed an all optical fault-tolerant 2×2 switch. The use of optics allows the exploitation of spatial parallelism. Thus, it can substantially increase the reliability of any multistage interconnection network which is a desirable feature for massively parallel processing applications that require robust interconnection networks. The analysis of the proposed switch was done using the baseline network. However, it can be employed as the basic building block of any multistage interconnection network. The results of the performance/reliability analysis show that without using any additional hardware, better network survivability is achieved.

Acknowledgment

The authors would like to thank King Abdulaziz City for Science and Technology (KACST) for the support under project no. KACST AR13-12.

References

[1] M. Guizani, "Picosecond multistage interconnection networks architecture for optical computing," *Appl. Opt.*, Vol. 33, March 1994.

[2] H. Ishikawa et al., "High-speed packet switching systems for multimedia communications," *IEEE J. Select. Areas Commun.*, Vol. SAC-5, Oct. 1987, pp. 1336-1345.

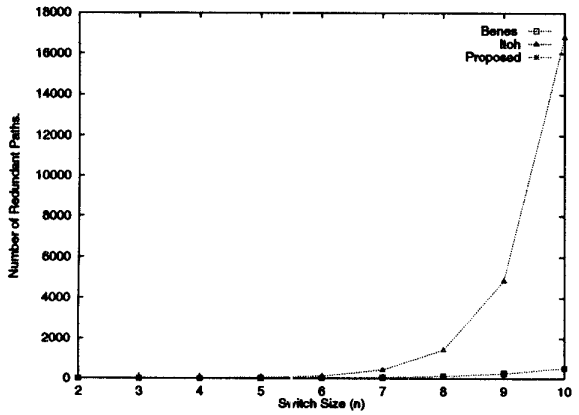

Figure 21: Number of Redundant Paths.

[3] A. Hac and H. B. Mutlu, "Synchronous optical network and broadband ISDN protocols," *Computer*, Vol. 22, No. 11, Nov. 1989, pp. 26-34.

[4] R. Handel, "Evolution of ISDN towards broadband ISDN," *IEEE Network*, Jan. 1989, pp. 7-13.

[5] "Broadband aspects of ISDN," *CCITT Recommendation I.12*, Nov. 1988.

[6] J. S. Turner, "New Directions in Communications," *Proc. IZS'86*, 1986, Paper A3, pp. 1-8.

[7] J. J. Kulzer and W. A. Montogomery, "Statistical switching architecture for future services," *Proc. ISS'84*, 1984, paper 43A.1, pp. 1-6.

[8] H. Imagawa et al., "A new self-routing switch driven with input-output address difference," *Proc. GLOBECOM'88*, Dec. 1988, pp. 1607-1611.

[9] T. Y. Feng, "A survey of interconnection networks," *IEEE Computer Mag.*, Vol. 4, Dec. 1981, pp. 12-27.

[10] H. S. Kim and A. Leon-Garcia, "A self-routing multistage switching network for broadband ISDN," *IEEE J. Select. Areas Commun.*, Vol. 8, No. 3, April 1990, pp. 459-466.

[11] A. Ghafoor, M. Guizani, and S. Sheikh, "An All-Optical Circuit-Switched Multistage Interconnection Networks", *IEEE J. on Selected Areas in Communications*, Vol. 8, No. 7, Oct. 1990, pp. 1595-1607.

[12] A. Itoh, "A fault-tolerant switching network for B-ISDN," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 8, Oct. 1991, pp. 1218-1226.

[13] N. Tzeng et al., "A fault-tolerant scheme for multistage interconnection networks," em Proc. 12th Int. Symp. Comp. Archit., June 1985, pp. 368-375.

[14] U. Garg and Y. Huang, "Decomposing Banyan networks for performance analysis," *IEEE Trans. Comp.*, Vol. C-37, No. 3, March 1988, pp. 371-376.