Fiber-ribbon pipeline ring network for high-performance distributed computing systems

Magnus Jonsson, Bertil Svensson, Mikael Taveniku and Anders Åhlander

Jonsson M, Svensson B, Taveniku M, Åhlander A. Fiber-ribbon pipeline ring network for high-performance distributed computing systems. In: Proceedings of the Third International Symposium on Parallel Architectures, Algorithms, and Networks, 1997. (I-SPAN '97). IEEE; 1997. p. 138-143.

DOI: http://dx.doi.org/10.1109/ISPAN.1997.645084

M. Jonsson, B. Svensson, M. Taveniku, and A. Åhlander, "Fiber-ribbon pipeline ring network for high-performance distributed computing systems," Proc. International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'97), Taipei, Taiwan, Dec. 18-20, 1997, pp. 138-143.

Fiber-Ribbon Pipeline Ring Network for High-Performance Distributed Computing Systems

Magnus Jonsson[1], Bertil Svensson[1, 2], Mikael Taveniku[2, 3], and Anders Åhlander[1, 3]

1. Centre for Computer Systems Architecture, Halmstad University, Halmstad, Sweden
2. Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden
3. Ericsson Microwave Systems AB, Mölndal, Sweden

Magnus.Jonsson@cca.hh.se, svensson@ce.chalmers.se, micke@ce.chalmers.se, Anders.Ahlander@cca.hh.se, http://www.hh.se/cca

Abstract

In this paper, we propose a high-bandwidth ring network built up with fiber-ribbon point-to-point links. The network has support for both packet switched and circuit switched traffic. Very high throughputs can be achieved in the network due to pipelining, i.e., several packets can be traveling through the network simultaneously but in different segments of the ring. The network can be built today using fiber-optic off-the-shelf components. The increasingly good price/performance ratio for fiber-ribbon links indicates a great success potential for the proposed kind of networks. We also present a massively parallel radar signal processing system with exceptionally high demands on the communication network. An aggregated throughput of tens of Gb/s is needed in this application, and this is achieved with the proposed network.

1 Introduction

In [1], we presented a WDM (Wavelength Division Multiplexing) star network for high-performance distributed real-time systems and analyzed how it performs in a massively parallel radar signal processing system. This system has several processing nodes, each comprising an array of processing elements. Although the WDM star architecture is very attractive and scales well to hundreds of these high-performance processing nodes, systems which require only a few tens of nodes can alternatively be realized by using optical fiber-ribbon links. Fiber-ribbon links offering an aggregated bandwidth of several Gb/s have already reached the market [2]. The price performance ratio is very promising.

In this paper, we present a pipeline ring network based on optical fiber-ribbon point-to-point links. In a pipeline ring network, several packets can be traveling through the network simultaneously, thus achieving an aggregated throughput higher than the capacity of a single link. Motorola OPTOBUS™ bidirectional links with ten fibers per direction are used but the links are arranged in a unidirectional ring architecture (Figure 1) where only M / 2 bidirectional links are needed to close a ring of M nodes. Nine of the fibers are used for time multiplexed circuit switched traffic, eight fibers for data and one fiber for clocking. The tenth fiber is dedicated for packet switched traffic using, for example, a token ring protocol. This fiber also carries control messages to reconfigure the TDMA (Time Division Multiple Access) schedule, (i.e., circuit establishment) for the other nine fibers.

The node synchronization requirement is relaxed compared to a traditional TDMA network, because the

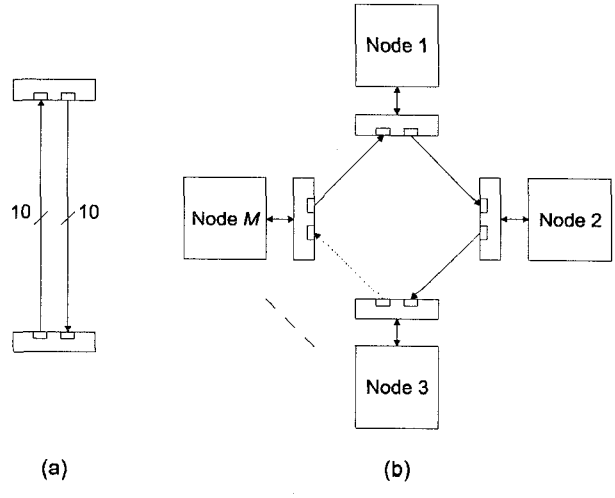

Figure 1: (a) Bi-directional fiber-ribbon link. (b) Unidirectional ring network built up with *M*/2 bi-directional links.

Link Owners / Links	Data slots 1	2	3	4	5
1 – 2	**1**	5	-	5	5
2 – 3	2	**2**	2	5	5
3 – 4	2	2	**3**	3	5
4 – 5	2	2	3	**4**	-
5 – 1	2	5	3	5	**5**

Figure 2: Example of allocation scheme for the links in a five-node system. The slot-initiators are typed bolded and different segments have different background shading.

access to the network is circulating among the nodes according to the physical order of the nodes in the ring. In addition, the ring can dynamically be partitioned into segments to obtain a pipeline ring network where several transmissions can be performed simultaneously.

Other high-performance ring networks include the WDM passive ring network [3] and the hierarchical WDM ring network [4] which, however, are more related to the WDM star network and star-of-stars network that we proposed in [5] and [1]. Other pipeline ring networks are described in [6] and [7]. However, these networks do not support circuit switching and guaranteed bandwidth for concurrent transmissions. More references to pipeline ring networks are found in [6].

We will describe a radar signal processing system as an example where a high-performance network of this kind is needed. At the same time, the network gives predictable performance which is important in this application. The sample system is a MIMSIMD (Multiple Instruction Streams for Multiple SIMD arrays) computer system for signal processing in future phased-array antenna radar systems. Each computation module, which forms a node in the ring network, contains hundreds of processors.

The rest of the paper is organized as follows. The network is presented in Section 2. In Section 3, a radar signal processing system is presented, and it is shown how the network fits this system. The paper is then concluded in Section 4.

2 Network description

Today, OPTOBUS links with 800 Mb/s per fiber are available [8]. In the proposed network, this translates to a bandwidth of 6.4 Gb/s for circuit switched traffic (on 8 fibers) and a bandwidth of 800 Mb/s for packet switched traffic (on one fiber). Fiber-ribbon links with higher bandwidths have been reported, especially when using each fiber as a separate serial channel (which, however, increases hardware complexity). For example, a 2 Gb/s per channel, 12 channel fiber-ribbon link was reported in [9], and array modules supporting 12×2.4 Gb/s for, e.g., fiber-ribbon links were reported in [10]. Larger networks can be built using clustering techniques and electronic gateway nodes.

Circuit switched and packet switched traffic will be discussed in 2.1 and 2.2 respectively. Then, in 2.3, circuit establishment will be described.

2.1 Circuit switched traffic

The first nine fibers in each link form a high-speed channel. All high-speed channels, together, form a high-speed ring network for circuit switched traffic. The access is divided into slots like in an ordinary TDMA network. However, in each slot, the network can be divided into segments. For each slot there is always one node responsible for initiating the traffic around the ring. This node is called the slot-initiator. At the end of the slot, the role of being slot-initiator is asynchronously handed over to another node, often the next node downstream. This can be done implicitly by just sensing the end of the slot.

The access is cyclic and each cycle consists of K slots. In the typical case, K is a multiple of M, where M is the number of nodes, and each node is slot-initiator in K/M slots. Each node is denoted as m_i, $1 \le i \le M$.

An example of an agreed schedule for a network with $K = M = 5$ slots per cycle is shown in Figure 2. Each column represents one time-slot and contains information on how the ring is segmented in that slot. Each number in a column is the node index of the owner of the corresponding link. The bold-typed numbers indicate the current slot-initiator. In each segment and slot, one, and only one, node can be the owner of the links, and hence has the right to use the segment links for transmission. In the first slot in the example, node m_1 (slot-initiator) owns the link between itself and node m_2. It can hence transmit to node m_2 but not to any other node. In the same slot, node m_2 can transmit to any of nodes m_3, m_4, m_5, or m_1. The choice is made by the process that owns the circuit to which the slot segment is associated with. A multicast to two or more of these nodes is also possible.

In the third slot, the link between node m_1 and node m_2 is free. Although the link is free, node m_1 must not disturb the asynchronous slot synchronization technique. Therefore, it transmits an empty packet to node m_2. In the fifth slot, node m_5 has the capability of transmitting a broadcast packet (a packet to all other nodes in the ring).

As indicated in Figure 3, the bandwidth utilization depends on the ratio between the total propagation delay around the ring and the slot length. This is an effect related

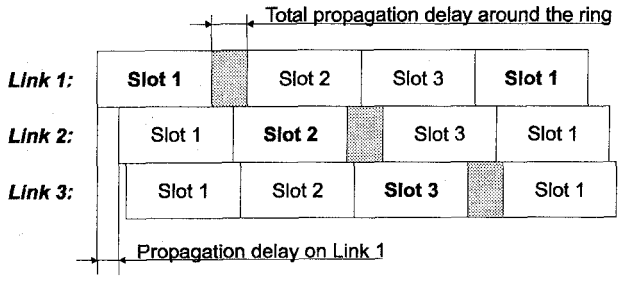

Figure 3: The bandwidth utilization depends on the ratio between the total propagation delay around the ring and the slot length.

to the asynchronous passing mechanism of the slot-initiator assignment. Also, the bandwidth utilization depends on how many segments on the average that can be utilized in each slot.

The latency grows linearly with distance, measured in number of hops (repeating latency in each node). Adding to the latency is also the propagation delay between source and destination node, as well as the delay until the first available slot for transmission. By distributing tasks in such a way as to minimize the number of hops, both latency and remaining bandwidth will be improved.

2.2 Packet switched traffic

The tenth fibers in all links together form a ring network totally dedicated for packet switched traffic. An ordinary ring protocol can be used. However, there are two requirements: (i) the protocol must allow to be halted when special packets for circuit establishment are to be transmitted (see Section 2.3), and (ii) the latency must be upper bounded to assure transmission of the packets for circuit establishment. When using, e.g, a token ring protocol on the packet network, this network will support low latency communication for sporadic packets at moderate traffic rates. At the same time, it is assured that the circuit switched traffic (often real-time traffic) is not disturbed by packet switched traffic.

2.3 Circuit establishment

When a node wants to establish a new circuit, it searches for slots where the required links are free, so allocation of a new segment can be done. First, the node's own slots (i.e., where the node itself is the slot-initiator) are searched. If not enough slots (actually only a segment in each slot) for the circuit could be allocated, the search is continued in other slots. In that case, a special *request packet* is transmitted on the packet network to ask other nodes to allocate the desired segment in their slots. This packet is immediately followed by a *collect packet* to collect information on the success of the slot segment allocations.

The request packet is broadcasted to all other nodes and contains information about the links required and the amount of slots needed. Each node then checks if any of its own slots have the required free links. If so, it prepares to modify the collect packet when it arrives (before forwarding it), to notify the requesting node which slots that have been allocated. However, if any of the previous nodes already have allocated slot segments and modified the collect packet, the number of slots needed has been decreased with the corresponding number of allocated slots. The number of slots still needed is indicated in a dedicated field in the collect packet. In this way, allocation of more slots than needed is avoided. However, several nodes can each allocate some of the slots needed and information about all these allocations is added to the same collect packet.

When the requesting node receives the collect packet after one round, it decides if it is satisfied with the number of allocated slots. If not, it sends a release packet. Otherwise, it can start using the established circuit immediately.

3 A radar signal processing system

The phased array antenna radar systems under consideration have a number of different requirements depending on the application. However, the algorithms are usually well known. They comprise mainly linear operations such as matrix-by-vector multiplications, matrix inversions, FIR-filtering, DFT, etc. In general, the functions will work on relatively short vectors and small matrices, but in a high pace and with large sets of vectors and matrices.

One of the goals of our research is to find a good scalable architecture which can give high enough computing speed for this application without too much loss in generality, i.e., the use of efficient programmable computers is preferred. A solution to this is to use the two-dimensional array SIMD machine as a building block. The two-dimensional array is well known in the literature and a number of machines have been built, e.g., MasPar [11], Connection Machine [12], and DAP [12]. Numerous successful mappings of algorithms on these machines have been done. However, these machines are not efficient when the calculations are too small for the machine size, i.e., a 32 by 32 matrix problem is hard to fit well on a 65k processor machine. So, it can be noted that the mesh is a promising architecture, but that the size of the mesh should be in the order of the size of the data structures in the calculations, i.e., the size of the matrices in the data set. In order to cope with the large number of matrices in the data,

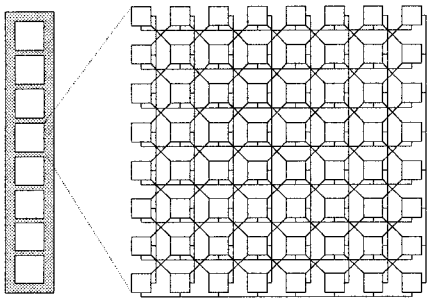

Figure 4: The computation-module architecture.

many computation modules, each with the two-dimensional array topology, can be interconnected to share the load. A very powerful interconnection network is needed for this, because each computation module (hereafter also called node) can produce a sustained data flow of 6 Gb/s. Also, guaranteed bandwidth must be supported to not disturb the dataflow.

This section briefly describes our proposal for a computer system, which is a multiple SIMD mesh system, intended to meet the imposed requirements in terms of computing power, generality, size, and power consumption. The computation-module architecture is presented in 3.1, while the signal processing chain is briefly described in 3.2. Then, the communication demands are discussed in 3.3, and communication aspects for mode changes in 3.4.

3.1 Computation-module architecture

The computation module used in the system is shown in a simplified form in Figure 4. Each module consists of eight 8-by-8 meshes working in a SIMD fashion. Internally in the meshes, the processors (PEs) are connected with nearest neighbor connections (x-grid), together with row and column broadcast lines. Computations and inter-PE communication can be performed simultaneously. In addition to the PE meshes, the module holds I/O-buffers, memory, and control units.

The module is designed for a processing performance of 12 GOPS (24 MOPS/PE). The I/O bandwidth is 750 MB/s in and out, respectively.

3.2 Signal processing chain

During a *Coherent Processing Interval* (CPI), which in this system is 16 ms, a number of pulses are transmitted and pulse samples (pulse echoes) are collected. A sample corresponds to a certain pulse, receiver channel, and time of collection after a transmitted pulse, i.e., the distance to target. The data collected during a CPI is here referred to

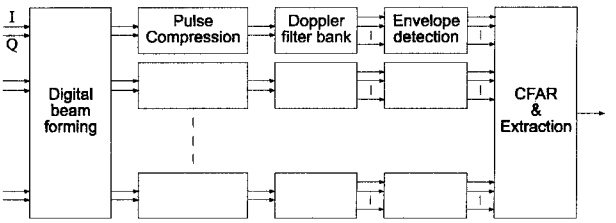

Figure 5: The signal processing chain.

as a *batch*. Each element in the batch is a complex number with 24 bit resolution of I and Q, respectively. The data from the receiver channels is fed into the signal processing chain illustrated in Figure 5. Into each channel, complex numbers are fed at a rate of 2 MHz. Below, the different steps in the signal processing chain are described.

The *digital beamformer* creates lobes from the receiver channel signals by multiplying them with complex weight factors. The weight factors are chosen so that signals in the respective pointing directions of the lobes are added in phase. In this system, a number of equally distributed beams are created and the beamforming is in fact a DFT over the input channels. Two computation-modules are used for this. The 64 point DFT is split up in sixteen 8 point DFTs which are calculated one per processor mesh. The two computation-modules are interconnected in a way which utilizes the symmetry in the 64 point DFT calculation. All the calculations are carried out in a systolic fashion, and thereby new data vectors can be fed in at full rate.

The goal of the *pulse compression* is to collect all received energy from one target into a single range bin, i.e., we get a better resolution in range. The received phase- or frequency-coded signal is correlated with the same code as used in the transmitter. In this system, one FIR-filter with a length of 16 is used for each lobe. Here, two computation-modules, working in a time-interleaved fashion, are used. The rows in the PE-arrays act as individual linear arrays, one per lobe, and the compressed values are systolically calculated.

The *Doppler filter bank* transforms the pulse bins to velocity bins. During a CPI, a number of pulse samples are collected. For each lobe, the pulses corresponding to the same range are processed using an FFT. The number and size of the FFTs are dependent on the working mode of the radar. In order to perform an FFT, a number of 8-point FFTs, one per PE, are calculated. The results from the PEs on the same row are combined via the PE memories, and the complete FFT is formed.

After the pulse compression and Doppler filtering have been performed, the phase information from the sample is no longer needed. The *envelope detection* removes, by an absolute-value calculation, the phase

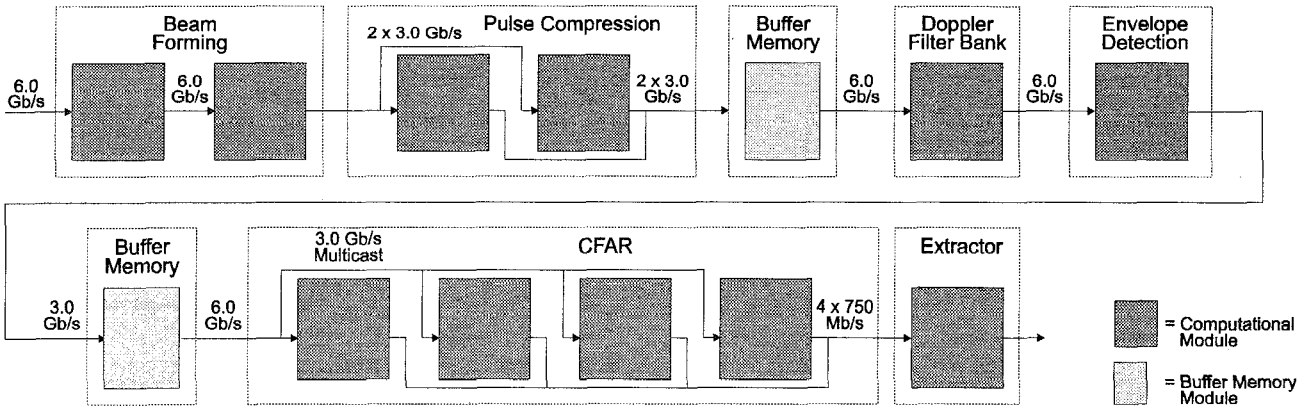

Figure 6: Data flow between the modules in the radar signal processing chain.

information from the samples. One computation-module takes care of this.

The *Constant False Alarm Ratio* (CFAR) processing is intended to reduce the number of false targets in each CPI, without loss in sensitivity. This combined constraint is usually solved by setting a threshold, based on some statistics, that makes the probability of a false alarm constant over time. Thus the name Constant False Alarm Ratio. Here it is assumed that the CFAR calculates the threshold based on the mean value of a neighbourhood to each resolution cell. Four modules are used for this.

The *extractor* calculates the centre of mass of a target detected in several resolution cells at the same time. The extraction could be combined with the CFAR calculations. However, here the extractor is mapped on one separate module.

Further information on the mapping of the computations on the modules is found in [13].

3.3 Communication demands

The signal processing system resulting from the signal processing chain described in Section 3.2, is shown in Figure 6, where also the bandwidth demands on the inter-module communication are given. As seen in the figure, the data flow is pipelined through the chain. In addition to the computational modules, two buffer-memory modules are needed to corner-turn the batch. Although the data flow has a very simple structure there can exist control messages in other directions. Also, the algorithms, and hence the data flow, can be changed when switching to a different working mode. A general network is therefore required. The maximum data flow from one module is 6.0 Gb/s. For simplicity, bandwidth for, e.g., error checking codes, is neglected in the analysis.

In the figure there are 13 nodes. In addition, the antenna is seen as one node (feeds the first node in the chain with data) and there is one master node responsible for supervising the whole system and interacting with the user. For simplicity, we assume all nodes being the slot-initiator in one slot per cycle, except for the master node which has two slots. In this way there are 16 slots per cycle, where one slot per cycle corresponds to a circuit bandwidth of 400 Mb/s. For a 6 Gb/s circuit between two nodes, segments in 15 of the 16 slots are needed. When the two nodes are nearest neighbors, all 16 slots can actually be allocated, as long as the link between the nodes must not be shared with any other nodes. When there are intermediate nodes between the source and destination nodes, allocation is not possible in those slots where one of the intermediate nodes is the slot-initiator.

Slots for both of the two 3 Gb/s dataflows to the pulse compression nodes can be allocated, since one of the two dataflows is tapped before adding the produced dataflow from the same node. The incoming dataflow to the CFAR nodes is broadcasted to all these nodes. Although this broadcast dataflow must remain unchanged until the last node, it can coexist with the produced dataflow from the CFAR nodes. The reason for this is that the broadcast bandwidth is only 3 Gb/s. The rest of the dataflows are pure pipeline flows and map easily on the network as long as the calculations are mapped on the nodes according to the pipeline order.

3.4 Mode changes

A number of different working modes are possible in a radar system. The task of one mode can, for example, be to scan the whole working range, while the task of another mode can be to track a certain object. Normally, the algorithm mapping and communication patterns are different for two different modes. The change of the circuits at mode changes can be performed in two different ways: (i) switching between the various slot-allocation schemes for a known set of modes, schemes that are statically stored in each node, and (ii) dynamically

changing the slot-allocation scheme in each node at a mode change, by establishing new circuits as described in Section 2.3.

In the first case, a mode change request is broadcasted by the subsystem responsible for mode changes (which is the master node). Immediately after the request packet, an acknowledge packet is transmitted. The acknowledge packet is halted in each node until the node is prepared for the requested mode change. In this way the master node knows that all involved nodes are prepared for the mode change when it receives the acknowledge packet after one turn around the ring. Whether or not other packets are allowed during a mode change depends on the tolerable latency of the mode change.

In the second case, the mode change is initiated by the master node in the same manner as a broadcast packet. However, each of the involved nodes is then responsible for requesting its required bandwidth. Each node is also sending its own acknowledge (or negative acknowledge if it failed to establish the required circuits) packet to the master node, indicating that it is prepared for the mode change.

When all the nodes have been prepared for the mode change, the system will change to the new mode in the next batch. The packets coming from the antenna in the new batch will be tagged to indicate the new mode. In that way, the nodes will be triggered to change to the new mode. Nodes that are placed later in the signal processing chain are triggered by the packets generated by succeeding nodes. Even if a node has a totally different job to do (and a different communication pattern) in the new mode, it can be triggered in this way. This is possible as long as two different batches are data independent.

4 Conclusions

We have presented a ring network where very high throughputs can be achieved, especially in systems where some kind of pipelined dataflow between the nodes exists. The network supports packet switched traffic at the same time as guaranteed bandwidth is supported through circuit switching. In a typical system, circuits can be set up for time-critical dataflows, guaranteeing that they are not disturbed by, e.g., control information. These features of the network are very appreciated in the radar signal processing system described in the paper. Also worth mentioning is that the network can be built today using fiber-optic off-the-shelf components.

5 Acknowledgement

This work is part of the REMAP project, financed by NUTEK, the Swedish National Board for Industrial and Technical Development, and the PARAD project, financed by the KK Foundation in cooperation with Ericsson Microwave Systems AB.

6 References

[1] M. Jonsson, A. Åhlander, M. Taveniku, and B. Svensson, "Time-deterministic WDM star network for massively parallel computing in radar systems," *Proc. Massively Parallel Processing using Optical Interconnections, MPPOI'96,* Lahaina, HI, USA, Oct. 27-29, 1996, pp. 85-93.

[2] D. Bursky, "Parallel optical links move data at 3 Gbits/s," *Electronic Design,* vol. 42, no. 24, pp. 79-82, Nov. 21, 1994.

[3] M. I. Irshid and M. Kavehrad, "A fully transparent fiber-optic ring architecture for WDM networks," *Journal of Lightwave Technology,* vol. 10, no. 1, pp. 101-108, Jan. 1992.

[4] A. Louri and R. Gupta, "Hierarchical optical ring interconnection (HORN): scalable interconnection network for multiprocessors and multicomputers," *Applied Optics,* vol. 36, no. 2, pp. 430-442, Jan. 10, 1997.

[5] M. Jonsson, K. Nilsson, and B. Svensson, "A fiber-optic interconnection concept for scaleable massively parallel computing," *Proc. Massively Parallel Processing using Optical Interconnections, MPPOI'95,* San Antonio, TX, USA, Oct. 23-24, 1995, pp. 313-320.

[6] P. C. Wong and T.-S. P. Yum, "Design and analysis of a pipeline ring protocol," *IEEE Transactions on communications,* vol 42, no. 2/3/4, pp. 1153-1161, Feb./Mar./Apr. 1989.

[7] M. Xu and J. H. Herzog, "Concurrent token ring protocol," *Proc. INFOCOM'88,* pp. 145-154, 1988.

[8] OPTOBUS Home Page, http://design-net.com/logic/optobus.homepage.html.

[9] H. Karstensen, "Parallel optical links - PAROLI, a low cost 12-channel optical interconnection," *Proc. LEOS'95,* vol. 1, pp. 226-227, 1995.

[10] R. G. Peall, "Development in multi-channel optical interconnects under ESPRIT III SPIBOC," *Proc. LEOS'95,* vol. 1, pp. 222-223, 1995.

[11] T. Blank, "The MasPar MP-1 architecture," *Proc. IEEE COMPCON Spring '90,* San Francisco, CA, USA, Feb. 26 - Mar. 2, 1990, pp. 20-24.

[12] R. M. Hord, *Parallel Supercomputing in SIMD Architectures.* CRC Press, 1990.

[13] M. Taveniku, A. Åhlander, M. Jonsson, and B. Svensson, "A multiple SIMD mesh architecture for multi-channel radar processing," *Proc. International Conference on Signal Processing Applications & Technology, ICSPAT'96,* Boston, MA, USA, Oct. 7-10, 1996, pp. 1421-1427.