Halmstad University Post-Print

Control-channel based fiber-ribbon pipeline ring network

Magnus Jonsson

N.B.: When citing this work, cite the original article.

©1998 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Jonsson M. Control-channel based fiber-ribbon pipeline ring network. In: Fifth International Conference on Massively Parallel Processing : proceedings : June 15-18, 1998, Las Vegas, Nevada. IEEE; 1998. p. 158-165.

DOI: <http://dx.doi.org/10.1109/MPPOI.1998.682139>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2734>

Control-Channel Based Fiber-Ribbon Pipeline Ring Network

Magnus Jonsson

Centre for Computer Systems Architecture, Halmstad University, Halmstad, Sweden

Magnus.Jonsson@cca.hh.se, <http://www.hh.se/dep/cca>

Abstract

In this paper, we propose a control-channel based ring network built up of fiber-ribbon point-to-point links. One of the fibers in each link forms part of the control-channel ring, over which medium access control information is sent immediately before data transmissions. This increases performance of the network. 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 meet high performance demands in, e.g., massively parallel signal processing systems, which is shown by example in the paper. Also, real-time demands can be met using slot reserving. The network, called CC-FPR (Control-Channel based Fiber-ribbon Pipeline Ring), can be built today using fiber-optic off-the-shelf components, and a prototype is currently under development. The increasingly good price/performance ratio for fiber-ribbon links indicates a great success potential for the proposed kind of 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, and less expensively, 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. The network is

called CC-FPR (Control-Channel based Fiber-ribbon Pipeline Ring). 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 uni-directional ring architecture (Figure 1) where only $M/2$ bidirectional links are needed to close a ring of M nodes (assuming that M is an even number).

As shown in Figure 2, the physical ring network is divided into two rings or channels, one data ring and one control ring. In each fiber-ribbon link, eight fibers carry data and one fiber is used to clock the data, byte for byte. Together, these fibers form a data channel that carries data-packets. The access is divided into slots like in an ordinary TDMA (Time Division Multiple Access) network. The tenth fiber is dedicated for bit-serial transmission of control-packets which are used for the arbitration of data transmission in each slot. The clock

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

Figure 2: Each link in the the physical ring network is divided into a data channel and a control channel, forming two sub-networks.

signal on the dedicated clock fiber, which is used to clock data, also clocks each bit in the control-packets.

The node synchronization requirement is relaxed compared to a traditional TDMA network and the network is somewhat related to a slotted ring network (but without the need for a central controller). This is because the 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 (for each slot) be partitioned into segments to obtain a pipeline ring network where several transmissions can be performed simultaneously (see the example in Figure 3). Even simultaneous multicast transmissions are possible when the multicast segments do not overlap. In order to provide guaranteed bandwidth in real-time computer systems, slot reserving is also supported.

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], [7] and [8] and more references are found in [6]. A fiber-ribbon ring network supporting pipelined circuit-switched traffic on nine fibers and packet-switched traffic on one fiber is described in [9]. Advantages of the CC-FPR network over these other networks include the use of high-bandwidth fiber-ribbon links (except for the network described in [9] which also uses fiber-ribbon links) and the tight relation between a dedicated control channel and the data channel without disturbing the flow of data-packets. In other words, control and data are overlapped in time. With less header overhead in the data-packets the slot-length can be shortened, to reduce latency, without offering too much in bandwidth utilization. Also, separating clock- and control-fibers simplifies the

Figure 3: In each slot, the ring can be segmented to allow multiple transmissions simultaneously.

transceiver hardware implementation, which is verified by the current prototype development.

The network described in [7] also relies on a separate control channel but needs a central control node that brings additional cost in hardware as well as additional latency when waiting for response from the central control node. The CC-FPR network is insensitive to propagation delay in the sense that no feedback is needed, from other nodes or from a central controller, between control-packet and data-packet transmissions.

The rest of the paper is organized as follows. The protocol is presented in Section 2. In Section 3, slot reserving is described, and in Section 4 a case study is presented to show the efficiency of the network. The paper is then concluded in Section 5.

2 Protocol description

In the first subsection we will describe the CC-FPR protocol. Then, in Subsection 2.2, we will discuss performance aspects related to protocol implementation. Throughout Section 2, slot reserving, as described in the next section, is assumed not being used.

2.1 The CC-FPR protocol

Before explaining the arbitration mechanism we will describe how data-packets travel on the ring. The access to the network is cyclic and each cycle consists of M time-slots, where M is the number of nodes. Each node is denoted as m_i , $1 \leq i \leq M$. In each slot there is always one node responsible for initiating the traffic around the ring. This node is called the slot-initiator. Each node is slot-

Figure 4: The role of being slot-initiator is cyclically repeated. Each of the M nodes is slot-initiator in one slot per cycle.

Figure 5: The node succeeding the slot-initiator initiates the control-packet transmission.

initiator in one slot per cycle as shown in Figure 4. At the end of the slot, the role of being slot-initiator is asynchronously handed over to the next node downstream. This can be done implicitly by just sensing the end of the slot.

The CC-FPR medium access protocol is based on the use of a control-packet that, for each slot, travels almost one round (over $M - 1$ links) on the control-channel ring as shown in Figure 5. The node that will be slot-initiator in the next slot initiates the transmission of the control-packet as shown in the figure. In the time domain the control-packet always travels around the ring in the time-slot preceding the time-slot for which it controls the arbitration (see Figure 6). The control-packet will hence always pass each node one time-slot before the data-packet it is related to passes.

The contents of the control-packet is shown in Figure 7. The control-packet consist of a start-bit followed by a M bit long link-reservation field and a M bit long destination field, where M is the number of nodes. Each bit in the

Figure 6: In each slot, a node passes/transmits one control-packet and one data-packet, where the control-packet is used for the arbitration for the next slot.

Link-reservation field					Destination field				
Bit 0	Bit 1	Bit 2	...	Bit M	Bit $M+1$	Bit $M+2$	...	Bit $2M$	
Start bit	Link 1	Link 2	...	Link M	Node 1	Node 2	...	Node M	

Figure 7: A control-packet contains a start bit, a link-reservation field, and a destination field.

link-reservation field tells if the corresponding link is reserved for transmission in the next slot. In the same way each bit in the destination field tells if the corresponding node will have a data-packet destined to it in the next slot. Additional information for, e.g., node insertion might also be included in the control-packet but is for clarity not shown in the figure.

Each node succeeding the slot-initiator checks the control-packet when it passes the node to see: (i) if it will receive a data-packet in the next slot, which is indicated by the node's bit in the destination field, and (ii) if a data-packet will pass the node in the next slot, which is indicated by the bit in the link-reservation field corresponding to the outgoing link of the node. If no data-packet will pass the node, i.e., the rest of the ring back to the slot-initiator is free, then the node will have the possibility to transmit a data-packet in the next slot in this part of the ring.

If a node has a packet possible for transmission it prepares, in advance, new link-reservation and destination fields to reserve needed links and notify destination node(s). In this way the node can immediately change the control-packet when it passes if the bit in the link-reservation field, corresponding to the outgoing link of the node, was set to zero. Since there is no data-packet that will pass the node, succeeding nodes have no use of the overwritten information in the control-packet.

Because all the nodes succeeding the slot-initiator repeat the procedure of checking the control-packet for the possibility to send, multiple transmissions in different segments of the ring might be possible in the same slot. An

Node	Outgoing control-packet		
	Link	Dest.	Transmission allocated
1	1 1 0 0 0	0 0 1 0 0	To Node 3
2	1 1 0 0 0	0 0 1 0 0	Could not allocate
3	1 1 0 0 0	0 0 1 0 0	Could allocate transmission to Node 4, 5, and 1 but had nothing to send
4	0 0 0 1 1	1 0 0 0 1	Multicast to Node 5 and 1
5	0 0 0 1 1	1 0 0 0 1	Could not allocate

Figure 8: A control-packet travels around a network with five nodes. Node 1 is slot-initiator.

example of how the control-packets travels around a five node network is shown in Figure 8. The arbitration will result in two concurrent data-packet transmissions in the next slot, one single-destination and one multicast packet as shown in Figure 9. Node m_1 is the slot-initiator in the example and therefore initiates the control-packet transmission described in Figure 8. It reserves Link 1 and Link 2 for transmission to node m_3 and informs node m_3 that it will have a data-packet destined to it in the next slot, by setting the corresponding bits in the link-reservation field and the destination field, respectively. Node m_2 and node m_3 do not change the control-packet but checks it to see if there will be any data-packets destined to them in the next slot. Node m_4 reserves Link 4 and Link 5 for a multicast transmission to node m_5 and node m_1 . Node m_5 then receives the control-packet and removes it from the ring.

The reason why the control-packet only travel over the first $M - 1$ links after the slot-initiator is that the clock signal is interrupted before the last link (see Figure 5). The node that initiated the transmission of the control-packet will not return the packet. The consequence of this is that it will not be informed whether there will be a data-packet destined to it or not in the next slot. However, the node just have to wait and see if there will come a data-packet with data or not. There will either come a packet destined to the node or an empty packet.

Each transmitter has $M - 1$ queues, one for each possible destination (the node itself excluded). When a multicast packet arrives for queuing it is put in the queue corresponding to the destination of the multicast destinations furthest away from the source node downstream. In this way multicast packets are treated in the same way as single-destination packets and multiple multicast packets can be traveling in the network at the same time when possible.

Figure 9: Example where Node 1 sends a single-destination packet to Node 3, and Node 4 sends a multicast packet to Node 5 and Node 1.

2.2 Performance aspects

Essential for the performance is that the delay that the control-packet experience in each node it bypasses is minimal, especially in large networks. One method is to organize the bits in the link-reservation field in the control-packet, for each slot, so that they appear in the same order as the control-packet travels. In other words, the first bit corresponds to the outgoing link from the slot-initiator. In this way a node that wants to change the contents of the control-packet does not have to store the whole packet before checking and possibly overwriting it. Instead it can retransmit the packet bit by bit and exchange the remaining part of the slot, if transmission was possible, after reading the bit in the link-reservation field corresponding to its outgoing link. The node's bit in the destination field in the incoming control-packet must, however, be checked before it is thrown away. Using this method, the delay in each node will only be one or a few bits.

As indicated in Figure 10, the bandwidth utilization depends on the ratio between the total propagation delay around the ring and the cycle length. This is an effect related 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 maximum aggregated throughput of the network that is possible due to the asynchronous slot-synchronization method, S_{max} , is:

Figure 10: The bandwidth utilization depends on the ratio between the total propagation delay around the ring and the cycle length. The boxes with bolded text notify which link each slot first propagates through.

$$S_{\max} = \frac{MPT_{\text{slot}}}{MT_{\text{slot}} + T_{\text{prop}}} \quad (1)$$

where M is the number of nodes, P is the average number of packet transmissions in each slot, T_{slot} is the duration of one slot, and T_{prop} is the total propagation delay around the ring. As an example we get a throughput of $S_{\max} = 1.9$ when $M = 16$, $P = 2$, $T_{\text{slot}} = 1 \mu\text{s}$, and $T_{\text{prop}} = 1 \mu\text{s}$ (200 meter fiber).

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, queuing delay, and 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.

Due to the use of a separate control-channel the data-packet header can be very short. Therefore, reception of data-packets is simplified and large software overheads are eliminated. Another positive consequence, in combination with the asynchronous slot-synchronization, is that the slot length can be rather short without significant reduction in bandwidth utilization. In turn, short slot-lengths decrease the latency and give a finer resolution to split messages into packets which, in fact, can increase the bandwidth utilization. A performance analysis of the network will be published elsewhere.

3 Slot reserving

Many computer systems have real-time demands where the network must offer guaranteed bandwidth for certain communication patterns. This can be done in the network by using slot reserving. Either the whole ring is reserved

Figure 11: The bandwidth utilization is improved when using slot reserving. In this example the cycle is prolonged by a fourth slot.

for a specific node in a slot, or several segments of the ring is dedicated to some specific nodes.

When slot reservation is allowed the cycle is prolonged to contain $Q = M + R$ slots, where R is the number of slots used for reservation. The value of R is chosen at system design and remains unchanged during operation of the network if the system function does not change radically. Such a change might be a mode change in a radar system, for example, switching from the task of scanning the whole working range to the task of tracking a certain object. The M ordinary slots cannot be reserved. In Figure 11, it is shown how the cycle from Figure 10 is prolonged by a fourth slot where node m_3 is the slot-initiator. However, all nodes in the network can try to reserve a segment of the ring in the slot.

When a node wants to reserve a slot, 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 reservation could be allocated, the search is continued in other slots. In that case, the node broadcasts a data-packet containing a request to all other nodes to allocate the desired segment in their slots. The packet 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. All nodes send a packet back to the requesting node to notify which slots, if any, that have been allocated. When the requesting node has received the answers, 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 reserved slots immediately. However, it should still send a release packet if more slots than needed was allocated.

All slots, in a cycle, where a node is the slot-initiator must be in sequence to not disturb the efficiency of the asynchronous slot synchronization method. The bandwidth utilization is, at the expense of higher latencies, improved when using slot reserving, as indicated in Figure 11. The maximum aggregated throughput of the network that is

Figure 12: Data flow between the modules in the radar signal processing chain.

possible due to the asynchronous slot-synchronization method, S_{max} , now becomes:

$$S_{max} = \frac{(M+R)PT_{slot}}{(M+R)T_{slot} + T_{prop}} \quad (2)$$

where M is the number of nodes, R is the number of slots for reservation, P is the average number of packet transmissions in each slot, T_{slot} is the duration of one slot, and T_{prop} is the total propagation delay around the ring.

The advantage of this slot reservation method over circuit-switching is that when a node does not need its reserved slot, the slot can be reused by other nodes in the segment. For example consider that node m_1 has a segment reserved containing the four links between itself and node m_5 . If node m_1 does not need the slot in a cycle the other nodes in the segment will be informed of that when the control-packet passes in the slot before. Node m_2 will first have the chance to take over, followed by the node m_3 and m_4 . Even reuse by multiple nodes in the same slot is possible if the communication demands of the other nodes in the segment allow for that.

4 Case study

A typical application with high throughput requirements and a pipelined dataflow between the computational modules is future radar signal processing systems [10] [1]. In Figure 12, a signal processing chain similar to those described in [10] and [1] is shown together with its bandwidth demands. Each computational module in the figure contains multiple processors itself. The chain is a good example containing both multicast, one-to-many, and many-to-one communication patterns. The aggregated throughput demand is 30 Gb/s. We leave out all details of the chain, referring to the two other papers, and do only treat the throughput demands here. The dataflow must not be disturbed by, for example, status

information that the network also has to transport. Slot reserving is therefore a good choice for the dataflow of the signal processing chain.

Today, OPTOBUS links with 800 Mb/s per fiber are available [11]. In the proposed network, this translates to a bandwidth of 6.4 Gb/s for data traffic on eight of the fibers. For simplicity we assume an efficient bandwidth of 6.0 Gb/s after, for example, check-sums have been excluded and Equation 2 has been taken into account. 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 [12], and array modules supporting 12×2.4 Gb/s for, e.g., fiber-ribbon links were reported in [13]. Larger networks can be built using clustering techniques and electronic gateway nodes.

In Figure 12, 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. We denote the antenna as node m_1 , the modules shown in the figure as node m_i , $2 \leq i \leq 14$, and the master node as m_{15} . The numbers of the modules are indicated in the figure also. The number of ordinary slots is hence 15 but the cycle is prolonged to also contain 30 slots for reservation. In this way there are 45 slots in a cycle, where one slot per cycle corresponds to a bandwidth of 133 Mb/s at a total efficient bandwidth of 6.0 Gb/s.

A feasible allocation scheme of the slots is shown in Figure 13. For clarity, all the reservation slots are placed after the ordinary slots. In a real implementation, however, the reservation slots are spread out so that each node is first slot-initiator in the ordinary slot and then, immediately afterwards, in two reservation slots. Care must, however, be taken when placing out the reservation slots. The reason is that when there are intermediate nodes

Links	Data slots																																				
	1	2		15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33		45													
1 – 2	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1													
2 – 3	2			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		2												
3 – 4				3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		3												
4 – 5				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3		3												
5 – 6				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5		5												
6 – 7				6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6		6												
7 – 8				7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7		7												
8 – 9				8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	-	-	-		-												
9 – 10				-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	9	9		9												
10 – 11				10	10	10	10	-	-	-	-	-	-	-	-	-	-	-	-	-	9	9	9		9												
11 – 12				10	10	10	10	11	11	11	11	-	-	-	-	-	-	-	-	-	9	9	9		9												
12 – 13				10	10	10	10	11	11	11	11	12	12	12	12	-	-	-	-	9	9	9		9													
13 – 14				10	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13	-	-	-		-													
14 – 15				15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-													
15 – 1					-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-													

Figure 13: One cycle of 45 slots where Slot 16 through 45 is reserved. Each number in the table indicates, for Slot 16 through 45, the owner of the corresponding link. Neighboring segments, in each slot, have different background shading. The slot-initiators are indicated in Slot 1 through 15.

between the source and destination nodes, allocation is not possible in those slots where one of the intermediate nodes is the slot-initiator.

The maximum data flow from one module is 4.0 Gb/s which corresponds to having a segment of the ring reserved in all of the 30 slots for reservation. Slots for both of the two 2 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 multicasted to all these nodes. Although this multicast dataflow must remain unchanged until the last CFAR node, it can coexist with the produced dataflow from the CFAR nodes. The reason for this is that the multicast bandwidth is only 2 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.

5 Conclusions

We have presented a ring network based on fiber-ribbon links. Very high throughputs can be achieved in the network, especially in systems where some kind of pipelined dataflow between the nodes exists. The protocol relies on the use of a dedicated control sub-network that is easy to implement using a dedicated fiber on each fiber-ribbon link. Guaranteed bandwidth is supported through

slot reserving, a method that allows for slot reuse when the guaranteed bandwidth is temporarily not needed. In a typical system, slot reserving can be used for time-critical dataflows, guaranteeing that they are not disturbed by, e.g., not so time-critical control information. Also worth mentioning is that the network can be built today using fiber-optic off-the-shelf components and this is an ongoing work.

6 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. OPTOBUS components for the building of a prototype has been sponsored by Motorola.

7 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] H. Jafari, T. G. Lewis, and J. D. Spragins, "Simulation of a class of ring-structured networks," *IEEE Transactions on Computers*, vol. 29, no. 5, pp. 385-392, May 1980.
- [8] M. Xu and J. H. Herzog, "Concurrent token ring protocol," *Proc. INFOCOM'88*, pp. 145-154, 1988.
- [9] 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.
- [10] 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.
- [11] OPTOBUS Home Page, <http://design-net.com/logic/optobus.homepage.html>.
- [12] H. Karstensen, "Parallel optical links - PAROLI, a low cost 12-channel optical interconnection," *Proc. LEOS'95*, vol. 1, pp. 226-227, 1995.
- [13] R. G. Peall, "Development in multi-channel optical interconnects under ESPRIT III SPIBOC," *Proc. LEOS'95*, vol. 1, pp. 222-223, 1995.