Journal of Parallel and Distributed Computing **61**, 1033–1051 (2001)
doi:10.1006/jpdc.2001.1726, available online at http://www.idealibrary.com on IDEAL®

Generalized Coincident Pulse Technique and New Addressing Schemes for Time-Division Multiplexing Optical Buses

S. Q. Zheng

Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083-0688

Keqin Li

Department of Mathematics and Computer Science, State University of New York, New Paltz, New York 12561-2499

Yi Pan

Department of Computer Science, Georgia State University, Atlanta, Georgia 30303

and

M. Cristina Pinotti

Dipartimento di Matematica, Università degli Studi di Trento, Povo (Trento), Italy

Received December 1, 1999; accepted October 6, 2000

Time-division multiplexing (TDM) optical buses have applications in computer networks, high-performance routers and switches, and interprocessor communication subsystems of parallel computers. Three waveguides are used to implement the previously proposed TDM optical bus. One waveguide is used for data transmission, and the other two waveguides are used for processor addressing. The coincident pulse technique is used to implement a unary addressing scheme. This architecture has two drawbacks: the bandwidth of address waveguides is wasted and the scalability of the bus is limited due to the unary addressing. In this paper, the coincident pulse technique is generalized in order to derive compact addressing schemes for TDM optical buses. Based on this new technique, optimal addressing schemes are presented. Several general methods for designing new addressing schemes are proposed to achieve various trade-offs. Compared with the previously proposed addressing scheme, the new schemes significantly improve the utilization of the bandwidth of optical waveguides and the scalability of a TDM optical bus. © 2001 Academic Press

Key Words: coincident pulse technique; fiber optics; optical bus; network; parallel processing; processor addressing; router; switch; time-division multiplexing.

0743-7315/01 $35.00
Copyright © 2001 by Academic Press
All rights of reproduction in any form reserved.

1. INTRODUCTION

Optical fibers have been successfully used in telecommunication networks to replace electrical wires. Due to their inherent high bandwidth, optical fibers also provide opportunities for their use in the construction of optical buses for implementing computer networks, high-performance routers and switches, and interprocessor communication subsystems of parallel computers. Processors connected by a bus can be considered fully connected. Electronic bus architectures are not scalable because electrical connections require a switching energy proportional to the wire length and an RC delay that grows quadratically with the wire length. Using guided lightwave technology to implement buses has the following advantages: increased communication speed, increased connecting distance, reduced power consumption, increased interconnect density, reduced crosstalk, increased reliability, and reduced cost.

The available fiber optic technology makes it possible to fit a very large number of simultaneous connections within the large bandwidth of a single fiber. One can implement a time-division multiplexed (TDM) bus in which dynamically available connections between processors use different packet slots. A packet slot consists of a series of pulse slots. These pulse slots are divided into two parts: an address frame, which is the "header" of the packet, for addressing the destination processor, and a data frame, which carries the data transmitted by the packet. Based on predictable light propagation delay per unit length, the ability of controlling optical delays precisely, and time-division multiplexing, it is possible to have synchronized concurrent accesses of an optical bus in a pipelined fashion. A pipelined TDM optical bus architecture has been proposed [6, 12, 18]. Physically, the implementation of such an optical bus is considered as a linear processor array. However, due to its TDM transmission mode [2, 8], processors in such an architecture can be considered fully connected. The pipelined TDM optical bus is a prowerful interconnection structure for parallel processing. Several variations of this optical bus have been proposed and algorithms on linear processor arrays with optical buses have been developed [5, 9–11, 14, 15, 20]. An array with reconfigurable optical buses (AROB) [16, 19], an implementation of the well-studied reconfigurable mesh architecture [13], is an array of processors connected by pipelined TDM optical buses.

TDM optical buses also have applications in computer networks. A TDM optical bus can be used to implement a switch/router in.a node of a conventional network (such as the Internet and ATM network) by a set of simple processing elements as follows. The processing elements attached to the bus correspond to the input ports and output ports of the node. Since the processing elements are fully connected, packets from any input port can be routed to any output port efficiently. A simple ATM switch design based on an optical bus is presented in [20]. A TDM optical bus can be used as a local-area network or a cluster (subnetwork) of a local-area network. Such a bus can also be generalized as a TDM link in a multihop TDM optical network [4]. A link is used to connect a set of nodes (processors), and a node can be connected to more than one link. Such a network can be characterized by a hypergraph [1] whose hyperedges correspond to links. Each link provides

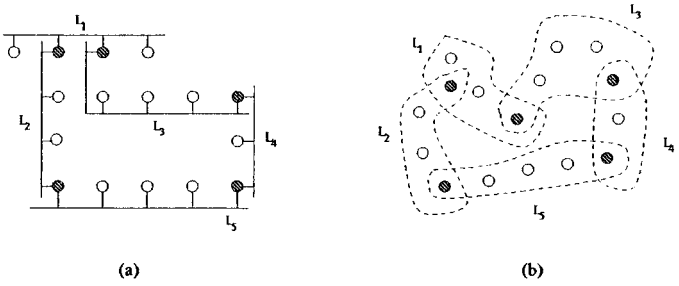

FIG. 1. Multihop optical TDM network, where L_i's are links (hyperedges), circles are nodes, and shaded circles are routing nodes. (a) Nodes are connected by optical TDM links. (b) The hypergraph corresponding to the network of (a).

all-optical communications among its connected nodes. A node that is connected by more than one link is considered as a routing node. Communications between two nodes that are not connected by a link are conducted along a path connecting the two nodes in the hypergraph corresponding to the network, and optical-to-electrical and electrical-to-optical conversions are performed at the routing nodes on the path. Routing in such an optical network consists of two components: self-routing in the optical domain for sending a packet from one node to another on the same link, and forwarding at a routing node by using the conventional routing method (such as table lookup). Figure 1 illustrates the structure of such an optical network. In parallel computing, pipelined TDM optical buses can be used to construct processor arrays. Many efficient parallel algorithms have been developed for such processor arrays (refer to [5, 9–11, 14, 15] for examples).

An addressing scheme based on the coincident pulse technique is used in the previously proposed TDM buses. Addressing in the optical domain increases the signal transmission rate of an optical bus. However, the available bandwidth of waveguides in previously proposed bus architectures is not fully utilized due to the following facts: the unary coding in the existing addressing scheme uses n pulse slots (n is the number of processors connected by a bus), which may take a large portion of a packet, and the address waveguides are only used for addressing purposes. In this paper, we first simplify the previous pipelined TDM optical bus by reducing the number of address waveguides and proposing to also use address waveguides for transmitting data. As a consequence, there is no need to distinguish waveguides for addressing and data transmission. Then we introduce a generalization of the known coincident pulse technique used in processor addressing. As a result of the combination of these two techniques, the utilization of the available bandwidth of all waveguides can be significantly increased. We demonstrate this by introducing several optimal addressing schemes. We also propose several general addressing scheme design methods that can be used to design a wide variety of addressing schemes achieving trade-offs among a few parameters. It is important to note that the applicability of the addressing schemes proposed in this paper is broad. They can be used in optical buses adopting either fixed-assignment TDM (synchronous TDM) multiaccess protocol or demand-assignment TDM (asynchronous TDM) multiaccess protocol, with or without pipelined transmission, even though our discussions focus on the pipelined optical bus using fixed-assignment TDM multiaccess protocol.

2. PIPELINED TDM OPTICAL BUS ARCHITECTURE

A pipelined optical bus consists of three folded waveguides, the *message waveguide*, the *reference waveguide*, and the *select waveguide*, connecting linearly ordered n processors, $P_0, P_1, ..., P_{n-1}$ (refer to Fig. 2). The message waveguide is used for carrying data. The reference waveguide and the select waveguide, which are also called *address waveguides* collectively, are used together for carrying address information encoded by using the coincident pulse technique [2, 8]. Each waveguide of the bus is divided into two segments, the *transmitting segment*, which is the upper half of the waveguide with taps originating from processors, and the *receiving segment*, which is the lower half of the waveguide with taps leading to processors. Let d be the pulse duration in time and v_l be the velocity of light in these waveguides. Define a *unit delay* to be the spatial length $d \cdot v_l$. The *generic pipelined TDM optical bus configuration* is given in Fig. 3, where only one waveguide is shown. In this configuration, the spatial separation of any two adjacent taps on a waveguide has D unit delays. The bus architecture of Fig. 2 is obtained from the generic bus configuration by adding a loop between every two adjacent taps on the receiving segments of the reference waveguide and the message waveguide. Each loop is an extra fiber segment of a unit delay. These added delays are used for optical self-routing purposes.

Referring to Figs. 2 and 4a, we explain the *coincident pulse technique*. When a source processor sends a packet (which will be defied shortly), it sends a reference pulse and a select pulse. The select pulse is transmitted later than the reference pulse with an appropriate time delay so that the two pulses arrive at the destination processor at the same time. That is, the coincidence of the two pulses occurs at the desired destination. Whenever a processor detects a coincidence of a reference pulse and a select pulse, it starts to read from the message waveguide. More specifically, suppose that a processor is going to send a packet to processor j. We use t_{ref} and t_{sel} to denote the time when the processor transmits its reference pulse and its select pulse, respectively. The two light pulses will coincide at processor j if and only if $t_{sel} = t_{ref} + j$.

We call the duration of each light pulse a *pulse slot*. Define a *packet* as a collection of information including an *address frame* and a *data frame* (see Fig. 4c). An address frame consists of a sequence of n pulse slots, as shown in Fig. 4a. The pulse slots in the address frame are named $a_0, a_1, ..., a_{n-1}$. A reference pulse is sent in a_0 of the reference waveguide. The existence of a select pulse at pulse slot a_j means that

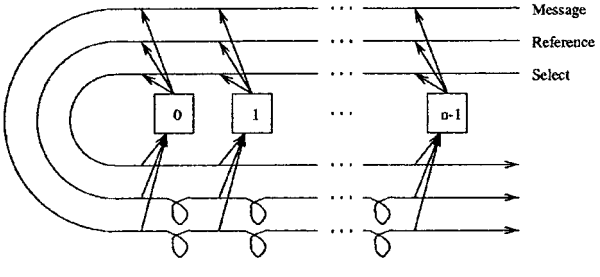

FIG. 2. A pipelined optical bus system.

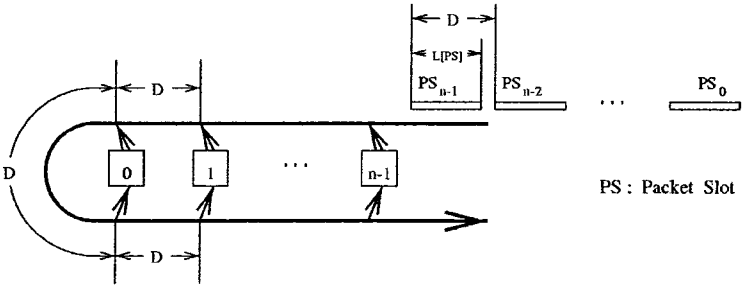

FIG. 3. A train of packet slots on a generic pipelined TDM optical bus.

a packet is to be sent to the processor j. The address frame for broadcasting can be easily implemented by setting a select pulse at each pulse slot of an address frame (see Fig. 4b). For easy reference, we call this addressing scheme the *unary addressing scheme*.

The simplest multiaccess protocol for this bus is the classical fixed-assignment TDM (FA-TDM). Figure 3 illustrates this bus access method on a generic pipelined TDM optical bus configuration. Using this method, the data transmission is partitioned into a sequence of phases, each consisting of a sequence of n *packet slots*, from $n-1$ down to 0 (see Fig. 3). Packet slot (PS for short) PS_i is fixed for processor i. Imagine a train of n packet slots is originated from the entry of a bus. If processor i has a packet to send, it loads its packet to PS_i. Let $L[PS]$ be the packet length in terms of unit delays. In order to provide correct unary addressing and prevent packet overlaps, the condition $D \geqslant L[PS] > n$ must hold. Several variations of this bus can be found in [11, 14, 20]. The coincident pulse addressing method also works for processor addressing in a TDM optical bus without using pipelined transmission.

In order for the pipelined optical TDM bus to operate properly, precise timing is required for the synchronization of processor operations. One can use one or more separate waveguides to connect all processors to a single global clock to ensure that all processors virtually share an identical global time. Different clock distribution implementation methods based on predictable optical signal delays are discussed in [17]. Suppose that the global (system) clock provides starting pulses of packet slots. These global clock pulses serve as packet envelopes. Then each processor can generate finer clock signals in the way shown in Fig. 5, and these finer clock signals are used to synchronize the transmitters and receivers of the processor. The small circles in Fig. 5 represent the extra segments that are required to delay the starting pulse of each packet. For easy reference, we call the time interval

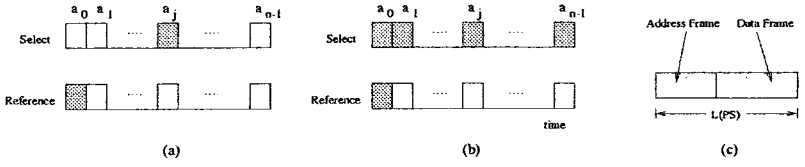

FIG. 4. Coincident pulse addressing: (a) an address frame for sending a packet to processor P_j; (b) an addressing frame for broadcasting; (c) a packet.

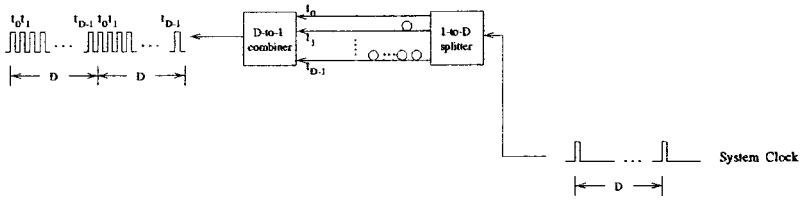

FIG. 5. Generating clock signals for each processor.

between the rising edges of two adjacent packet starting pulses a packet period, and we call t_i in such an interval the ith pulse of the packet period. Clearly, repeated subsequences of the finer clock signals can be extracted by optical hardware similar to the one shown in this figure. All discussions in the rest of this paper are based on precise timing and synchronization.

Note that this clock distribution method can also be applied to the optical links in a multihop TDM optical network. In such a network, the spatial separations of adjacent taps may not be the same. One can use one waveguide to distribute the clock signals. The rightmost processor of Fig. 2 (processor $n-1$) is called the head processor of the bus. If the packet envelop pulses are sent by the head processor, these pulses can be used to synchronize the transmitters and receivers of all processors. As long as all the spatial separations of adjacent taps are greater than the packet length $L[PS]$, pipelined packet transmission can be ensured.

3. IMPROVED UNARY ADDRESSING

Observe that it is a big waste if the select and reference waveguides only serve the purpose of addressing. In fact, these two waveguides can also be used for data transmission.

Consider the bus architecture of Fig. 2, and suppose that processor P_i wants to send data to processor P_j. Processor P_i uses the address setting of Fig. 4a, and loads the data frame of a packet slot on all three waveguides. After this is done, we have a situation shown in Fig. 6a. The snapshot at time that processor P_j detects the coincidence of select and reference pulses is shown in Fig. 6b. At this time, processor P_j expects to read a data frame from each reference and message waveguide after exactly $n-1$ additional pulse slots, and it expects to read a data frame from select waveguides after exactly $n-j-1$ additional pulse slots. Actually, the unit delay loops on the receiving segment of the message waveguide (see Fig. 2) are not necessary. Assume, that these loops are not present. Then, at the time processor P_j detects the coincidence of select and reference pulses, it expects to read a data frame from each of select and message waveguides after exactly $n-j-1$ additional pulse slots, and read a data frame from reference waveguide after exactly $n-1$ additional pulse slots, as shown in Fig. 6c. The subsequences of clock signals for sampling the data frame of each waveguide can be provided by optical hardware similar to that shown in Fig. 5. Consequently, the message waveguide becomes redundant and can be eliminated. Compared with the previously proposed method, this modified method not only reduces the number of required waveguides by one, but also doubles the data transmission rate.

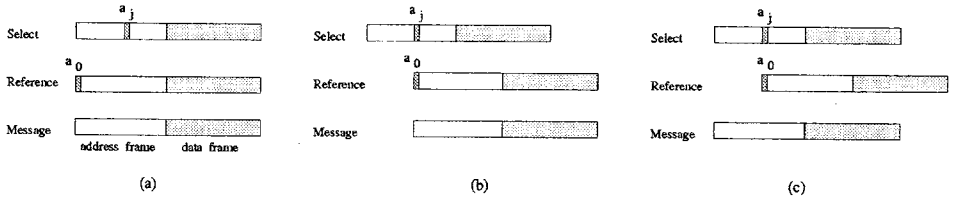

FIG. 6. Data transmission using address waveguides.

4. GENERALIZED COINCIDENT PULSE TECHNIQUE FOR REDUCING ADDRESS FRAME LENGTH

The bandwidth utilization of the waveguides in a TDM optical bus is restricted by the unary addressing scheme because of the requirement that $L[PS] > n$. The length of unary addresses increases linearly as a function of the number of processors, and this makes packet size $L[PS]$ large for a bus system with many processors. Large packet size results in inefficiency. Furthermore, the requirement $D \geqslant L[PS]$ for a pipelined TDM bus makes the bus less scalable. Thus, reducing the address frame size is of importance in improving the system efficiency and scalability.

In the rest of this paper, we introduce addressing schemes with reduced address frame lengths and discuss the hardware required for supporting these schemes. All these new addressing schemes are based on the generic TDM optical bus architecture depicted in Fig. 3 and the coincident pulse technique. One or more waveguides are used for both addressing and carrying data. Unlike the bus architecture described in Section 2, there are no delay loops on the receiving segments of the waveguides of new bus architectures. We generalize the coincident pulse technique by using unit delay loops (if required) on the taps leading to a processor and optical logical gates, such as optical AND, XOR (exclusive OR), and NOR gates [7], to form an address detecting circuitry for the processor. This circuitry evaluates in the optical domain a Boolean expression associated with the processor. The values of the variables of this expression are transmitted sequentially on one or more waveguides.

Suppose that w waveguides $A_0, A_1, ..., A_{w-1}$ are used to implement a TDM optical bus. We will not separate message waveguides from address waveguides. All waveguides are used for carrying address information as well as data. To facilitate our discussion, we generalize the notion of a packet. A two-dimensional packet slot PS of length $L[PS]$ is a $w \times L[PS]$ array F of pulse slots, which is divided into two parts, a $w \times p$ subarray AF and a $w \times (L[PS] - p)$ subarray DF called (two-dimensional) address frame and (two-dimensional) data frame, respectively. As before, address frame and data frame are used for addressing and carrying data, respectively. The kth column of F corresponds to timing signal t_k of a packet slot. An optical pulse in pulse slot $F_{r,s}$, $0 \leqslant r \leqslant w-1$ and $0 \leqslant s \leqslant L[PS]-1$ can only be injected by the sending processor at its time t_s of the packet slot dedicated to the processor. The pulses in the same column of F are injected into the waveguides at the same time, and the pulses in row i of F are transmitted over waveguide A_i. The state of $F_{r,s}$ can be detected by a receiving processor. If a processor wants to send

a packet to processor P_j, it sends address information that encodes j in the address frame AF of the packet. Processor P_j determines whether or not a packet is destined for it by checking the optical binary information carried in the address frame AF of its enclosing packet using its optical address detecting circuitry, and the coincidence of a set of specified pulses indicates that the packet is addressed to P_j. Our coincident pulse technique generalizes the known technique by allowing one to detect the presence and absence of pulses transmitted over one or more waveguides.

Remarks. The delay loops are introduced on the waveguides of the bus structure of Section 2, and if we exchange the connecting positions of two processors on such a bus, then their addresses must be exchanged. Since the unit delay loops used in our generalized coincident pulse technique are not introduced on the waveguides but on the (input) taps leading to receiving processors, we can consider input taps of a processor as part of the address-detecting circuitry of the processor. Then, if we exchange the connecting locations of two processors (along with their address detecting circuits), their addresses remain unchanged. Thus, using generalized coincident pulse technique, processor addresses are not dependent on the relative physical positions of processors. Therefore, the addressing schemes based on our generalized coincident pulse technique provide flexibility for the bus construction compared with the previous known coincident pulse technique.

5. BASIC ADDRESSING SCHEME DESIGNS

We show how to use the generalized coincident pulse technique discussed in the previous section to design basic compact addressing schemes. In Section 5.1, we consider two extreme cases—addressing schemes with one pulse slot and addressing schemes with one waveguide, and in Section 5.2, we consider addressing schemes that use multiple pulse slots and multiple waveguides. The generalization of these schemes will be discussed in Section 6.

5.1. Vertical and Horizontal Addressing Schemes

We call a $w \times 1$ addressing scheme a *vertical addressing scheme* and a $1 \times p$ addressing scheme a *horizontal addressing scheme*. A horizontal addressing scheme uses a minimum number of waveguides, and the length of a vertical addressing scheme is the shortest possible.

First, we introduce a simple vertical address scheme. Assume $n = 2^w$. Let $j_{w-1} j_{w-2} \cdots j_1 j_0$ denote the w-bit binary representation of integer j, and label each processor P_j, $0 \leqslant j \leqslant 2^w - 1$, by $j_{w-1} j_{w-2} \cdots j_1 j_0$. There are $2w$ waveguides, $A_0, A_1, ..., A_{2w-1}$. Relabel these waveguides by B_i, $\overline{B_i}$, for $0 \leqslant i \leqslant w-1$. Such a TDM optical bus connecting 8 processors (where $w = 3$) is shown in Fig. 7.

We label the only column in an address frame array by a. Suppose that a processor wants to send a packet to processor P_j. It sends a pulse in pulse slot a of a packet as follows: if $j_k = 0$, it sends a pulse in a of $\overline{B_k}$, and if $j_k = 1$ it sends a pulse in a of B_k. Any processor P_j detects that a packet is addressed to it by sensing coincidence of exactly w pulses from the w address waveguides. The coincidence of

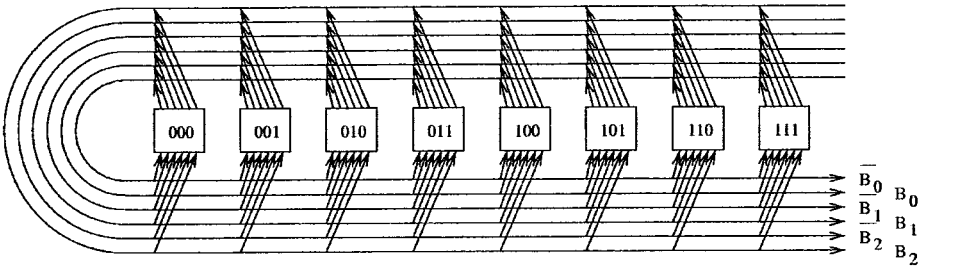

FIG. 7. The optical TDM bus architecture that uses a vertical addressing scheme.

w pulses can be detected by using an optical AND gate in a processor. The inputs of the AND gate are as follows: if $j_k = 0$, then the AND gate has an input from $\overline{B_k}$, and if $j_k = 1$, then the AND gate has an input from B_k. Suppose that processor P_3 wants to send a packet to processor P_5 in the bus shown in Fig. 7. It sets the address pulse in the address frame of six address waveguides as shown in Fig. 8a. All $2w$ waveguides can be used to transmit data.

The receiving circuitry of processor P_5 is shown in Fig. 9. In this figure, a sequence T of timing signals $t_1, t_2, ..., t_{L[PS]-1}$, which are generated by the hardware similar to Fig. 5, are used to sample the pulse slots of a data frame. The timing signal t_0, which corresponds to the pulse slot a, is used to detect the coincidence of three addressing pulses.

Given an addressing scheme AS, let $W[AS]$ and $N[AS]$ denote the number of waveguides implementing AS and the maximum number of addressable processors by AS, respectively. Denote the vertical addressing scheme discussed above by AS_{v_1}. Clearly,

$$N[AS_{v_1}] = 2^{\frac{W[AS_{v_1}]}{2}}.$$

Though the address frame length of this addressing scheme is minimized, the utilization of waveguides in terms of the number of addressable processors is not maximized. We now show how to increase the number of processors connected by an optical bus using vertical addressing schemes.

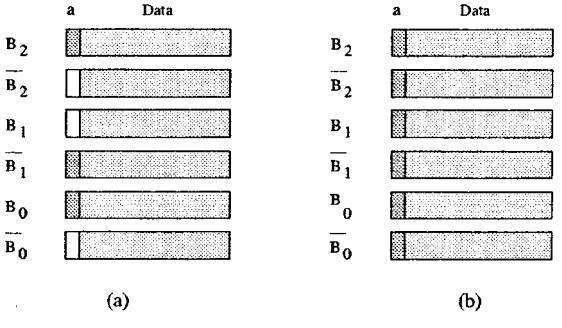

FIG. 8. Address settings for a bus using 1-pulse addressing scheme: (a) sending a packet to processor P_5 and (b) broadcasting to all processors.

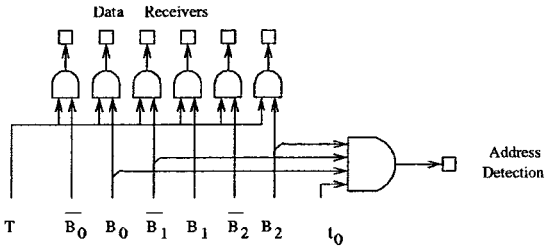

FIG. 9. The receiving circuitry of processor P_5 of the bus shown in Fig. 7.

The basic idea of AS_{v_1} is to associate each processor P_j with a unique subset W_j of address waveguides. A waveguide A_i is used to identify processor P_j if and only if A_i is in W_j. The coincidence of pulses from all waveguides in W_j at time t_0 indicates that a packet is addressed to P_j. However, for two subsets W_j and W_k of a given set of waveguides to address different processors P_j and P_k, they must satisfy $W_j \not\subseteq W_k$ and $W_k \not\subseteq W_j$; otherwise either W_j or W_k addresses both P_j and P_k. If we use each x-subset of a given set of waveguides to address a unique processor, then the total number of distinct x-subsets of a set of w waveguides is $\binom{w}{x}$, which is maximized when $x = \lfloor \frac{w}{2} \rfloor$. Denote this vertical addressing scheme by AS_{v_2}. Then we have

$$N[AS_{v_2}] = \frac{W[AS_{v_2}]!}{\left\lceil \frac{W[AS_{v_2}]}{2} \right\rceil ! \left\lfloor \frac{W[AS_{v_2}]}{2} \right\rfloor !}.$$

For AS_{v_1} and AS_{v_2} using the same number W of waveguides,

$$\frac{N[AS_{v_2}]}{N[AS_{v_1}]} \approx \frac{2^{\frac{W+1}{2}}}{\sqrt{\pi W}}$$

(by Stirling's formula). Vertical addressing scheme AS_{v_2} can connect much more processors than vertical addressing scheme AS_{v_1}. For example, using 6 address waveguides, AS_{v_2} addresses 20 processors, whereas AS_{v_1} addresses 8 processors.

In an $1 \times p$ horizontal addressing scheme, we can also use pulses in a subset of pulse slots in an address frame to address a processor. The selected pulses in different pulse slots of an address frame are delayed using delay loops for appropriate

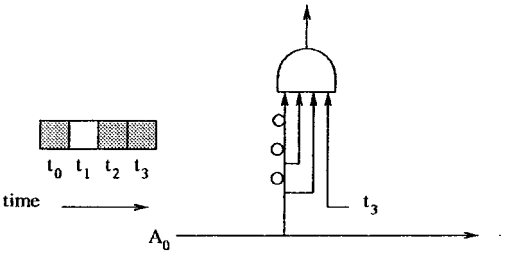

FIG. 10. Detecting the presence of a subset of pulses.

units so that they coincide at time t_{p-1} of the timing sequence. For example, consider a horizontal address frame of length 4. Denote the ith pulse slot of an address frame by AF_i. For detecting whether or not there are optical pulses in pulse slots AF_0, AF_2, and AF_3, the circuitry shown in Fig. 10 can be used. By an argument similar to that for the vertical addressing scheme described above, the maximum number of processors can be addressed by a $1 \times p$ addressing scheme using this technique is

$$\binom{p}{\left\lfloor \frac{p}{2} \right\rfloor} = \frac{p!}{\left\lfloor \frac{p}{2} \right\rfloor! \left\lceil \frac{p}{2} \right\rceil!}.$$

Given an addressing scheme AS, we use $L[AS]$ to denote the length of its address frames. Let $I[AS] = L[AS] \cdot W[AS]$, which is the total number of pulse slots in an address frame AS (over all waveguides). Clearly, $I[AS] \geqslant \lceil \log_2 N[AS] \rceil$. Since $\lceil \log_2 N[AS] \rceil$ is the minimum total number of bits for distinguishing $N[AS]$ objects, we call an addressing scheme AS an *optimal* if $I[AS] = \lceil \log_2 N[AS] \rceil$. An optimal addressing scheme addresses a set of processors using minimum addressing information.

Consider the design of an *optimal vertical addressing scheme* that uses w waveguides (A_0, A_1, ..., A_{w-1}) for addressing 2^w processors. Denote the only address pulse slot on address waveguide A_i as $AF_{i,0}$. We use the binary representation $j_{w-1} j_{w-2} \cdots j_0$ to represent the address of processor P_j and associate the presence and absence of a pulse in a pulse slot with binary value 1 and 0, respectively. Treating a pulse slot $AF_{i,0}$ in the address frame as a Boolean variable, then for a processor to send a packet to P_j, it sends a pulse in $AF_{i,0}$ (i.e., setting $AF_{i,0} = 1$), $0 \leqslant i \leqslant w-1$, if and only if $j_i = 1$. Processor P_j detects a packet addressed to it if $(AF_{0,0} = j_0) \wedge (AF_{1,0} = j_1) \wedge \cdots \wedge (AF_{w-1,0} = j_{w-1}) = \bigwedge_{i=0}^{w-1} (AF_{i,0} = j_i) = 1$ at time t_0, where $\wedge$ is the logical AND operator. Processor P_j detects coincidence of $w+1$ pulses by a circuitry constructed using optical AND and NOR gates. For example, for $w = 6$ and $j = 53$ (note: the binary representation of 53 is 110101), the

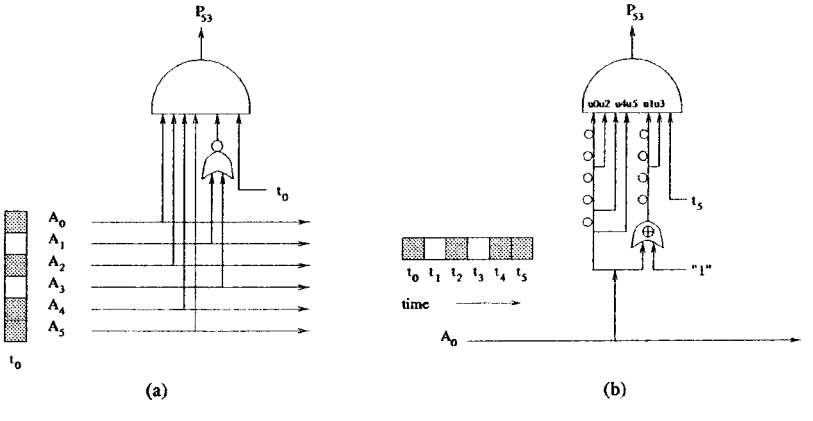

FIG. 11. Address detecting circuitry of processor P_{53}: (a) optimal vertical addressing scheme and (b) optimal horizontal addressing scheme.

coincidence detecting circuitry associated with P_j is shown in Fig. 11a. The NOR gate plays the role of several inverters.

Now we show how to design an *optimal horizontal addressing scheme* that uses one waveguide and address frames of length p to address 2^p processors. Denote the ith pulse slot of an address frame by AF_i. We use the binary representation $j_{p-1}j_{p-2}\cdots j_0$ to represent the address of processor P_j and associate the presence and absence of a pulse in a pulse slot with binary value 1 and 0, respectively. We treat each pulse slot AF_i as a Boolean variable. When a processor sends a packet to P_j, it sends a pulse in AF_i (i.e., setting $AF_i = 1$), $0 \leqslant i \leqslant w-1$, if and only if $j_i = 1$. Processor P_j determines a packet addressed to it if it detects $\bigwedge_{i=0}^{w-1} (AF_i = j_i) = 1$. Processor P_j detects this condition by a circuitry constructed using unit delay loops, and optical AND and XOR gates. More specifically, an XOR gate plays the role of several inverters, and $(p-i-1)$ unit delay loops are needed for enforcing that the delayed signals u_i to coincide with timing signal t_{p-1}. The pulse coincidence detecting circuitry associated with P_{53} is shown in Fig. 11b.

Optimal schemes have several drawbacks, which include excessive splitting loss, additional optical gates and delay loops. Another disadvantage of optimal vertical and horizontal addressing schemes is that they cannot be used to broadcast a message to all processors effectively. Improved broadcasting can be accomplished by either using one more address waveguide or an extra address pulse slot. In the former case, the presence of a specified address pulse on the additional waveguide indicates that the messages on all waveguides are broadcast to all processors. In the latter case, the presence of a pulse in an additional address pulse slot indicates that the messages on all waveguides are broadcast to all processors. In either case, a processor receives the broadcast message(s) after detecting this designated pulse.

5.2. Block Addressing Schemes

In this section, we consider $w \times p$ addressing schemes with $w \geqslant 1$ and $p \geqslant 1$, which are called *block addressing schemes*. Vertical and horizontal addressing schemes are special cases of block schemes with $p = 1$ and $w = 1$, respectively.

Regarding an address frame as a Boolean array, we know that there are exact 2^{wp} distinct $w \times p$ Boolean address patterns, and consequently at most 2^{wp} processors can be addressed by a $w \times p$ block addressing scheme. Thus, a $w \times p$ block addressing scheme AS is an *optimal block addressing scheme* if it can be used to address 2^{wp} processors (note: $I[AS] = L[AS] \cdot W[AS] = w \cdot p$). In the following, we will show how to design an optimal block addressing scheme.

We associate each processor P_k with a unique $w \times p$ Boolean array AP^k (an address pattern) as its address. Any processor who wants to send a packet to P_k sends a pulse in the pulse slot $AF_{i,j}$ of the address frame in the packet if $AP^k_{i,j} = 1$, where $0 \leqslant i \leqslant w-1$ and $0 \leqslant j \leqslant p-1$. By treating $AF_{i,j}$ as a Boolean variable, processor P_k detects a packet addressed to it by detecting condition

$$\bigwedge_{i=0}^{w-1} \bigwedge_{j=0}^{p-1} u_{i,j} = 1$$

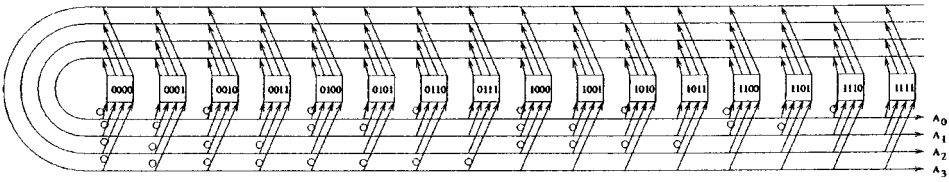

FIG. 12. A 16-processor TDM bus using the base-2 block addressing scheme.

at time t_{p-1} of a timing sequence, where $u_{i,j} = AF_{i,j}$ if $AP^k_{i,j} = 1$ and $u_{i,j} = \overline{AF_{i,j}}$ if $AP^k_{i,j} = 0$. In the construction of the address detection circuitry, one or more optical XOR gates are used as inverters to complement 0 values, unit delay loops are used to align $u_{i,j}$ signals with timing signal t_{p-1}, and one or more optical AND gates are used to detect the coincidence of $wp+1$ signals. Since there are 2^{wp} distinct $w \times p$ Boolean arrays, 2^{wp} processors can be addressed by this method. An optimal block addressing scheme shares the same advantages and disadvantages of the optimal horizontal and vertical addressing schemes.

To remedy some of the disadvantages of optimal block addressing schemes, a sun-optimal block addressing scheme can be designed by enforcing the address patterns to satisfy certain properties. For example, we may restrict that each row of a $w \times p$ address frame contains exactly one 1, and map each such address frame to a unique processor as follows: a processor's address is represented by a w-digit p-ary number (i.e., a w-digit number of base p) $d_{w-1}d_{w-2}\cdots d_0$, and an address frame AF addresses processor with address $d_{w-1}d_{w-2}\cdots d_0$ if $AF_{j,d_j} = 1$, $0 \leqslant j \leqslant w-1$. Clearly, p^w processors can be addressed by this scheme. The coincidence-detecting circuitry of processor with address $d_{w-1}d_{w-2}\cdots d_0$ is constructed by an optical AND gate with $w+1$ inputs which include a tap from the receiving segment of each waveguides, and a timing signal t_{p-1} as follows: $(p-d_j-1)$ unit delay loops are introduced to the tap leading from the receiving segment of address waveguide A_j to the AND gate. When a processor wants to send a packet to the processor with address $d_{w-1}d_{w-2}\cdots d_0$, it sends pulses in pulses slots AF_{j,d_j}, $0 \leqslant j \leqslant w-1$. The processor with address $d_{w-1}d_{w-2}\cdots d_0$ determines a packet is addressed to it if it detects $\bigwedge_{j=0}^{w-1} AF_{j,d_j} = 1$ at time t_{p-1}. If a processor wants to broadcast a packet to all other processors, it can simply send pulses in all pulse slots in an address frame. It is easy to verify that this addressing scheme works properly. For easy reference, we call this class of addressing schemes *base-p block addressing schemes*. For example, for $w = 4$, the architecture using base-2 block

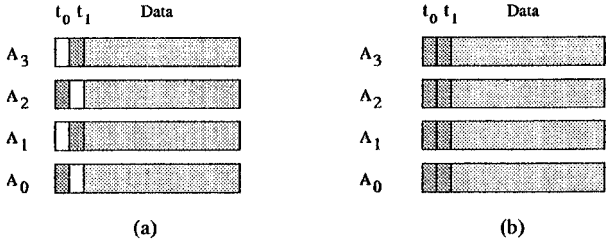

FIG. 13. Address settings for a bus using the base-2 block addressing scheme: (a) sending a packet to processor P_{10} and (b) broadcasting to all processors.

scheme is shown in Fig. 12. The address settings for addressing P_{10} and broadcasting are shown in Figs. 13a and 13b, respectively.

6. GENERALIZATIONS

The addressing schemes presented so far can be used as building blocks for constructing new schemes. In this section, we propose several general methods for designing new addressing schemes from existing ones. These methods can be used to achieve trade-offs among several parameters such as number of waveguides, address frame length, number of processors to be connected, and the complexity of address-detecting circuitry associated with processors. An additional important feature of the addressing schemes derived from these methods is the modular implementation of address-detecting circuitry: the address-detecting circuitry for each processor can be divided into several components (i.e., modules), and the address-detecting circuitry of more than one processor can have components of the same structure.

6.1. Grouped Addressing Schemes

We first introduce a class of block addressing schemes called *grouped addressing schemes*. There are two types of grouped addressing schemes, *vertically grouped addressing schemes* and *horizontally grouped addressing schemes*. A vertically (respectively, horizontally) grouped addressing scheme is constructed as follows. We partition a $w \times p$ address frame vertically (respectively, horizontally) into k subframes AF^m, $0 \leqslant m \leqslant k-1$, such that AF^m is a $w \times p_m$ (respectively, $w_m \times p$) subframe and $\sum_{i=0}^{k-1} p_i = p$ (respectively, $\sum_{i=0}^{k-1} w_i = w$). The processors are also partitioned into k groups PG^m, $0 \leqslant m \leqslant k-1$. The jth processor in PG^m is denoted by P_j^m. Subframe AF^m is used to address all processors in processor group PG^m according to a particular block addressing scheme AS^m, and furthermore, no pulses are present in the address pulse slots not in AF^m. Clearly, the total number of processors can be addressed by such a scheme is $\sum_{i=0}^{k-1} N[AF^i]$.

For example, consider a 16-processor TDM bus using a 3×4 addressing scheme that is vertically partitioned into two 3×2 addressing schemes AF^0 and AF^1. The processors are also partitioned into two groups PG^0 and PG^1, each of them has 8 processors. AF^0 is used to address processors in PG^0 and AF^1 is used to address processors in PG^1, according to base-2 block addressing schemes only using the rightmost 3 bits of the binary processor address. This bus architecture is shown in Fig. 14. The setting of the address frame for addressing processor P_{10} (binary representation of 10 is 1010) is shown in Fig. 16a. Consider a 16-processor TDM bus using a 2×8 addressing scheme that is vertically partitioned into four 2×2 subframes AF^0, AF^1, AF^2, and AF^3. The processors are also partitioned into four groups PG^0, PG^1, PG^2, and PG^3, each of them has 4 processors. Subframe AF^i is used to address processors in PG^i according to base-2 block addressing schemes only using the rightmost 2 bits of the binary processor addresses. This bus architecture is shown in Fig. 15, and the setting of the address frame for addressing processor

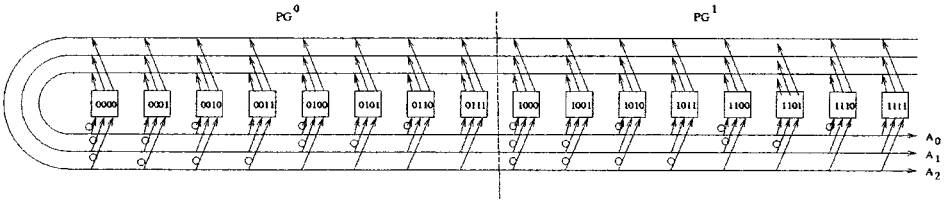

FIG. 14. A 16-processor optical TDM bus architecture using a 3×4 vertically grouped block addressing scheme. There are two processor groups, each of them has 8 processors and is addressed by a 3×2 subscheme.

P_{10} is shown in Fig. 16b. Both grouped addressing schemes can broadcast by sending a pulse in every pulse slot of an address frame.

By comparing Figs. 14 and 15 with Fig. 12, and comparing Fig. 16 with Fig. 13, one can easily see the trade-offs between the number of waveguides and the address frame lengths. The address-detecting logics of processors in the architecture shown in Fig. 15 are simpler than that of the processors in the architecture shown in Fig. 14, and the address-detecting logics of processors in the architecture shown in Fig. 14 are simpler than that of the processors in the architecture shown in Fig. 12. Referring to Fig. 15, one can see that several processors have address detecting circuitry of the same structure.

6.2. *Hierarchical Addressing Schemes*

We further generalize the idea of partitioning processors into groups and addressing a processor using combined information of a processor group that contains the processor and the position of the processor within its group by presenting another class of addressing schemes called *hierarchical addressing schemes.* The idea behind the hierarchical addressing schemes is to partition processors into groups in a recursive fashion.

First, all processors are partitioned into k_1 groups $PG^0, PG^1, ..., PG^{k_1-1}$. Then, each group PG^i is partitioned into k_2 subgroups $PG^{i,0}, PG^{i,1}, ..., PG^{i,k_2-1}$. Such a process continues for l levels until a group $PG^{i_1, i_2, ..., i_l}$ contains only one processor, and this processor can be uniquely denoted by $P_{i_1, i_2, ..., i_l}$. Let all processors constitute a level-0 group, and define group $PG^{i_1, i_2, ..., i_k}$ as a level-k processor group. Let C_{i_j} be a Boolean expression indicating that a processor is in the i_jth subgroup of a level-$(j-1)$ group if and only if $C_{i_j} = 1$. Then, the condition for a processor to be $P_{i_1, i_2, ..., i_l}$ is $C_{i_1, i_2, ..., i_l} = \bigwedge_{j=1}^{l} C_{i_j} = 1$. Thus, for a processor $P_{i_1, i_2, ..., i_l}$ to detect that

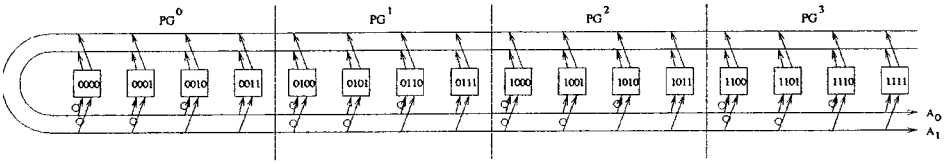

FIG. 15. A 16-processor optical TDM bus architecture using a 3×8 vertically grouped block addressing scheme. There are four processor groups, each of them has 4 processors and is addressed by a 2×2 subscheme.

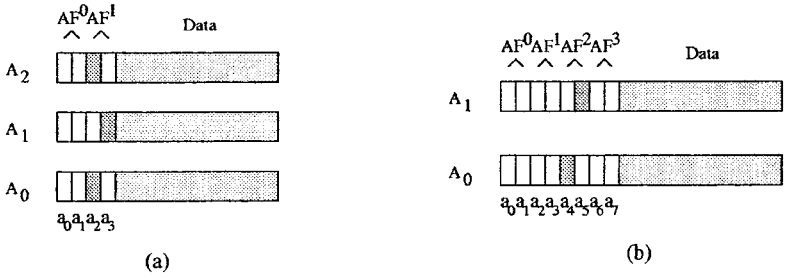

FIG. 16. Addressing processor P_{10} (a) using a 2-group vertically grouped addressing scheme and (b) using a 4-group vertically grouped addressing scheme.

a packet is uniquely destined for it, we must enforce that a unique address frame state is used in the packet so that the address-detecting circuitry associated with $P_{i_1, i_2, \ldots, i_l}$ detects that $C_{i_1, i_2, \ldots, i_l}$ is true.

We can implement this idea in two ways, which leads to two different types of hierarchical addressing schemes, *vertically oriented hierarchical addressing schemes* and *horizontally oriented hierarchical addressing schemes*. An l-level vertically (respectively, horizontally) oriented hierarchical addressing scheme is constructed by partitioning a $w \times p$ address frame AF vertically (respectively, horizontally) into l subframes AF^k, $1 \leqslant k \leqslant l$, such that AF^k contains p_k columns (respectively, w_k rows) of AF and $\sum_{k=1}^{l} p_k = p$ (respectively, $\sum_{k=1}^{l} w_k = w$). All processors (note: they form the level-0 processor group) are partitioned into k_1 level-1 groups PG^0, PG^1, ..., $PG^{k_1 - 1}$. An address scheme AS^1 defined on AF^1 is used for addressing these k_1 processor groups in such a way that all processors in a group PG^{i_1} are addressed if condition C_{i_1} is true. Each processor group PG^{i_1} is partitioned into k_2 level-2 groups $PG^{i_1, 0}$, $PG^{i_1, 1}$, ..., $PG^{i_1, k_2 - 1}$. An addressing scheme AS^2 defined on AF^2 is used for addressing all k_2 level-1 subgroups of any level-1 group PG^{i_1} in such a way that all processors in a level-2 subgroup PG^{i_1, i_2} of any level-1 group PG^{i_1} are addressed if condition C_{i_1} is true and C_{i_2} is true. In general, each level-j processor subgroup $PG^{i_1, i_2, \ldots, i_j}$ is partitioned into k_{j+1} level-$(j+1)$ subgroups $PG^{i_1, i_2, \ldots, i_j, 0}$, $PG^{i_1, i_2, \ldots, i_j, 1}$, ..., $PG^{i_1, i_2, \ldots, i_j, k_{j+1} - 1}$. An addressing scheme AS^{j+1} defined on AF^{j+1} is used for addressing k_{j+1} level-$(j+1)$ subgroups of any level-j subgroup $PG^{i_1, i_2, \ldots, i_j}$ in such a way that all processors in a level-$(j+1)$ subgroup $PG^{i_1, i_2, \ldots, i_j, i_{j+1}}$ of any level-j subgroup $PG^{i_1, i_2, \ldots, i_j}$ are addressed if condition $C_{i_{j+1}}$ is true. A level-l subgroup $PG^{i_1, i_2, \ldots, i_l}$ contains one processor, and this processor is $P_{i_1, i_2, \ldots, i_l}$. The condition for this processor to be addressed is $C_{i_1, i_2, \ldots, i_l} = \bigwedge_{k=1}^{l} C_{i_k} = 1$. Condition C_{i_k} can be defined by coincidence of address pulses and selected timing signals according to addressing scheme AS^k. Clearly, the total number of addressable processors using such an addressing scheme is $\prod_{i=1}^{l} N[AS^i]$.

The reusability of address-detecting logic components is an inherent property of hierarchical addressing schemes. The circuitry for detecting condition C_{i_k} using our generalized coincident pulse technique is a component of the address-detecting circuitry of all processors in the same level-k processor group. Processors in the i_{k+1}th group of different level-k processor groups use the same circuitry for $C_{i_{k+1}}$ to detect their memberships in level-$(k+1)$ processor groups.

It is easy to see that the optimal block addressing scheme presented in Section 6 can be implemented as a horizontally oriented hierarchical scheme. In the optimal block-addressing scheme, a processor P_k detects a packet addressed to it by detecting condition $\bigwedge_{i=0}^{w-1} \bigwedge_{j=0}^{p-1} u_{i,j} = 1$, where $u_{i,j} = AF_{i,j}$ if $AP_{i,j}^k = 1$, and $u_{i,j} = \overline{AF_{i,j}}$ otherwise. We can partition AF into subframes AF^i, $0 \leqslant i \leqslant w-1$, each being a row of AF. Subframe AF^0 is used to address 2^p level-1 processor groups by checking $\bigwedge_{j=0}^{p-1} u_{0,j}$. Subframe AF^i is used to address 2^p level-$(i+1)$ processor groups of a level-i group defined by the state of subarray consisting AF^0 through AF^{i-1}, and each such level-$(i+1)$ group corresponds to a unique condition $\bigwedge_{j=0}^{p-1} u_{i,j}$. Similarly, the optimal block-addressing scheme can also be implemented as a vertically oriented hierarchical scheme. Readers can easily see that the base-p block-addressing scheme can be implemented as a horizontally oriented hierarchical scheme.

We can combine vertically and horizontally oriented addressing schemes to obtain addressing schemes by allowing partitioning an address frame horizontally and vertically in a recursive fashion. Corresponding to such a recursive partition of an addressing scheme is the recursive partition and addressing of processor groups.

6.3. *Hybrid Addressing Schemes*

Both grouped addressing schemes and hierarchical addressing schemes use the idea of partitioning an address frame into subframes and partitioning processors into groups. The difference between these two classes of addressing schemes lies in the ways the processor groups are addressed by subframes. In a grouped scheme, the separation of address subframes directly corresponds to the separation of processor groups: different subframes are used to address disjoint processor groups. We call this feature the *direct separation property*. In a hierarchical scheme, a processor group is addressed by a common state of a subset of address subframes: a smaller processor group G'' is contained in a larger processor group G', the subframes used to address G' form a subset of the subframes addressing G'', and the state of the subframes for addressing G' is also used for addressing G'' (with additional address information). We call this feature the *inductive separation property*. By combining the direct and inductive properties of grouped and hierarchical schemes, we obtain a new class of addressing schemes, called the *hybrid addressing schemes*. Basically, a hybrid addressing scheme is obtained by partitioning an address frame recursively in alternating directions and correspondingly partitioning processors into processor groups using direct and inductive separation properties.

For example, consider a hybrid addressing scheme using a $w \times p$ address frame AF. First, AF is partitioned horizontally into three subframes AF^1, AF^2, and AF^3 such that each AF^1 and AF^2 consists of one row of AF. Then, the $(w-2) \times p$ subframe AF^3 is partitioned vertically into p subframes $AF^{3,0} AF^{3,1}$, ..., $AF^{3,p-1}$. Suppose we use AF^1 and AF^2 to implement a horizontally oriented hierarchical subscheme to address 2^{2p} processor groups (this is possible because of the inductive separation property and the optimal horizontal addressing for each AF^1 and AF^2), and we use the vertically grouped addressing scheme AF^3 for addressing processors in each group. If we use a $(w-2) \times 1$ optimal vertical addressing scheme for each

$AF^{3,i}$ (which is the ith column of AF^3), then each $AF^{3,i}$ can be used to address 2^{w-2} processors, and each group addressed by AF^1 and AF^2 has $2^{w-2}p$ processors (due to direct separation property of AF^3). Then, the total number of addressable processors by this hybrid addressing scheme is $2^{2p+w-2}p$.

The class of hybrid addressing schemes includes grouped and hierarchical addressing schemes as subclasses. This approach provides maximum freedom for finding trade-offs among various parameters such as cost, performance, and scalability of a TDM optical bus system.

7. FINAL REMARKS

With the current precision in mechanical layout, subpicosecond time precision is achievable. Techniques employing controlled delay have many applications in data communication using optical waveguides [3]. Pipelined TDM optical buses are examples of such applications. We presented several new addressing schemes for pipelined optical buses using FA-TDM. These schemes improve the utilization of the bandwidth of all the waveguides used, compared with the previously proposed scheme.

The major disadvantage of the fixed-assignment TDM (FA-TDM) is the requirement that each processor has a fixed allocation of packet slot regardless whether or not it has a packet to transmit. A demand-assignment TDM method (DA-TDM) allocates packet slots to processors dynamically according to their demands and the traffic situation. It belongs to the class of asynchronous TDM (ATDM) multiaccess methods. The ATDM multiaccess methods are introduced to reduce the inefficiency associated with fixed slot assignments, with certain degree of penalties for the overheads in polling and/or slot reservations. A pipelined ATDM optical bus with hardware slot reservation was presented in [20]. The new addressing schemes proposed in this paper can be easily incorporated into the pipelined ATDM optical bus of [20].

In parallel computing, the pipelined TDM optical bus with conditional delays proposed in [11] and the reconfigurable pipelined TDM optical bus described in [14] utilize the unary coding to facilitate binary prefix-sums computation. Using unary coding, the operation of prefix-sums over n binary operands, one per processor, can be carried out in one bus cycle, which is the end-to-end propagation time of a light pulse on a waveguide used in the optical buses of [11, 14]. Binary prefix-sums operation has many applications such as sorting, data compaction, data partition, and processor reordering. Many efficient parallel algorithms based on efficient parallel binary prefix-sums have been developed (refer to [5, 9–11, 14, 15] for examples). The new addressing schemes proposed in this paper can be incorporated into the pipelined TDM buses of [11, 14] as follows: use a set of address waveguides for one-to-one and one-to-many communications, and use a pair of additional waveguides for binary prefix-sums computation, which is performed across the boundaries of packet slots. This method can reduce the address frame size and increase the bus scalability while accommodating efficient binary prefix-sums operations.

REFERENCES

1. C. Berge, "Hypergraphs," North-Holland, Amsterdam, 1989.
2. D. Chiarulli, R. Melhem, and S. Levitan, Using coincident optical pulses for parallel memory addressing, *IEEE Comput. Magazine* **20**(12) (1987), 48–58.
3. H. F. Jordan, Exploring precise delay in optical multiprocessor interconnects, *in* "Proceedings of the 2nd International Conference on Massively Parallel Processing Using Optical Interconnects," pp. 357, 1995.
4. P. E. Green, "Fiber Optic Networks," Prentice–Hall, New York, 1993.
5. Z. Guo, Sorting on array processors with pipelined buses, *in* "Proceedings of 1992 International Conference on Parallel Processing," pp. 289–292, 1992.
6. Z. Guo, R. Melhem, R. Hall, D. Chiarulli, and S. Levitan, Array processors with pipelined optical busses, *J. Parallel Distrib. Comput.* **12**(3) (1991), 269–282.
7. J. Jahns and S. H. Lee, "Optical Computing Hardware," Academic Press, San Diego, 1994.
8. S. Levitan, D. Chiarulli, and R. Melhem, Coincident pulse techniques for multiprocessor interconnection structures, *Appl. Opt.* **29**(14) (1990), 2024–2039.
9. K. Li, Y. Pan, and S. Q. Zheng, Fast and efficient parallel matrix multiplication algorithms on a linear array with a reconfigurable pipelined bus system, *IEEE Trans. Parallel Distrib. Systems* **9**(8) (1998), 705–720.
10. K. Li, Y. Pan, and S. Q. Zheng (Eds.), "Parallel Computing Using Optical Interconnections," Kluwer Academic, Dordrecht, 1998.
11. Y. Li, Y. Pan, and S. Q. Zheng, Pipelined TDM optical bus with conditional delays, *Optical Eng.* **36**(9) (1997), 2417–2424.
12. R. Melhem, D. Chiarulli, and S. Levitan, Space multiplexing of waveguides in optically interconnected multiprocessor systems, *Comput. J.* **32** (4) (1989), 362–369.
13. R. Miller, V. K. Prasanna-Kumar, D. Reisis, and Q. F. Stout, Meshes with reconfigurable buses, *in* "Proceedings of MIT Conference on Advanced Research in VLSI," pp. 163–178, 1988.
14. Y. Pan and K. Li, Linear array with a reconfigurable pipelined bus systems—Concepts and applications, *in* "Proceedings of 1996 International Conference on Parallel and Distributed Processing Techniques and Applications," pp. 1431–1442, 1996.
15. Y. Pan, K. Li, and S. Q. Zheng, Fast nearest neighbor algorithms on a linear array with a reconfigurable pipelined bus system, *J. Parallel Algorithms Appl.* **13** (1998), 1–25.
16. S. D. Pavel and S. G. Akl, On the power of arrays with reconfigurable optical buses, *in* "Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications," pp. 1443–1454, 1996.
17. C. Qiao and R. Melhem, Time-division optical communications in multiprocessor arrays, *IEEE Trans. Comput.* **42**(5) (1993), 577–590.
18. C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, Optical multicasting in linear arrays, *Int. J. Optical Comput.* **2**(1) (1991), 31–48.
19. S. Rajasekaran and S. Sahni, Sorting, selection, and routing on the array with reconfigurable optical buses, *IEEE Trans. Parallel Distrib. Systems* **8**(11) (1997), 1123–1132.
20. S. Q. Zheng and Y. Li, Pipelined asynchronous time-division multiplexing optical bus, *Optical Eng.* **36**(12) (1997), 3392–3400.