COMMUNICATION IN THE LARPBS (OPTICAL BUS) MODEL: A CASE STUDY

BRIAN J. D'AURIOL
Department of Computer Science, The University of Texas at El Paso, El Paso, Texas, 79968-0518, USA
email: bdauriol@cs.utep.edu

A case-based analysis is presented of the Binary Prefix Sums computation primitive in the LARPBS optical bus model. A model of meta-data communication is developed for purposes of this analysis. The analysis reveals that the communication in the LARPBS model may be describable by such a meta-data model.

1 Introduction

Optical interconnection networks have received growing attention in the literature, in part, due to the widespread view in the research community that such optical-based networks can support teraflop or beyond high performance computing. Several different kinds of optical interconnection networks have been proposed. In particular, the linear optical bus interconnection network models provide for functional abstractions of communication and computation primitives (e.g. broadcasts and binary prefix sums). Further, it has been claimed [1] that such networks can be practically implemented.

Several linear optical bus-based models have appeared in the literature: *Array with Reconfigurable Optical Buses* (AROB) [2], *Linear Array with a Reconfigurable Pipelined Bus System* (LARPBS) [3], *Linear Pipelined Bus* (LPB) [4], and *Pipelined Optical Bus* (POB) [5]. A brief comparison of the four can be found in [1]. The focus in this paper is on the LARPBS model.

The LARPBS model is a parallel computational model that makes use of properties of light, unidirectional light pulse propagation and predictable propagation delay per unit length, to enable synchronized and concurrent access of the optical bus in a pipelined fashion [1]. Synchronized access is provided by the *coincident pulse technique* [6,3] where pulses along three distinct waveguides must be detected at particular synchronized points. In addition, the optical bus can be reconfigured by segmenting into multiple independent segments. The LARPBS model provides a set of communication (e.g. broadcast) and computation (e.g. binary prefix sums) primitives. These algorithms can be used as 'building blocks' to construct larger applications (see [7]).

One unique and perhaps controversial claim made of the LARPBS model is that many of the primitive operations have $O(1)$ *bus-cycle* complexity.

Hence, it is indicated that certain computations have a significant and moreover *fundamental* speedup over other comparable parallel implementations. However, little justification for this astonishing result has yet been published. In [7], a discussion is given concerning the validity of claiming a constant bus cycle in relation to fewer than approximately 1000 processor bus-based systems. Such justification is based on the relative optical communication speeds vs. processor cycle time and the consistent treatment in literature regarding how communication has been modeled. However, speedup in a parallel application depends on many factors, including how the application's communication pattern is harnessed. This paper focuses on the aspect of the application's communication requirements. Specifically, the Binary Prefix Sums primitive is analyzed as a case study. The analysis reveals that the communication in the LARPBS model may be describable by a meta-data model.

This paper is organized as follows. The LARPBS model is reviewed in Section 2. In Section 3, a meta-data model is developed as a tool for the case-based analysis. In Section 4 the meta-data model is applied to the Binary Prefix Sums computation on the LARPBS model. In Section 5, an algorithm is proposed to broadcast the conditional delay switch settings. Conclusions are presented in Section 6. The LARPBS model and the Binary Prefix Sums algorithm is presented in [1]. However, some of the notation is augmented and some minor notational errors have been corrected in this presentation.

2 LARPBS

A review of the LARPBS model as presented in [1] is conducted in this section.

The LARPBS model is illustrated in Figure 1. There are N processors $\mathcal{P} = (P_0, P_1, \ldots, P_{N-1})$ arranged in a linear topology that is connected by three waveguides, one for the data transmission of a message, one for the transmission of a *reference* pulse, and one for the transmission of the *select* pulse. The pulses are generated by appropriate optical devices shown in the figure as pulse injectors. The pulses can be detected by the pulse detectors. Due to the linear propagation of the pulse, pulses propagate from low numbered processors to higher number processors along the upper part of the waveguide (i.e., left to right in the figure). Hence, these pulses propagate along the bottom part of the waveguide from high numbered processors to lower number processors and are eventually detected by the pulse detectors. The waveguides have a defined length such that the first one half of the length (called the *transmission segment*) is reserved for the pulse injectors while the second one half of the length (called the *receiving segment*) is reserved for the pulse detectors.

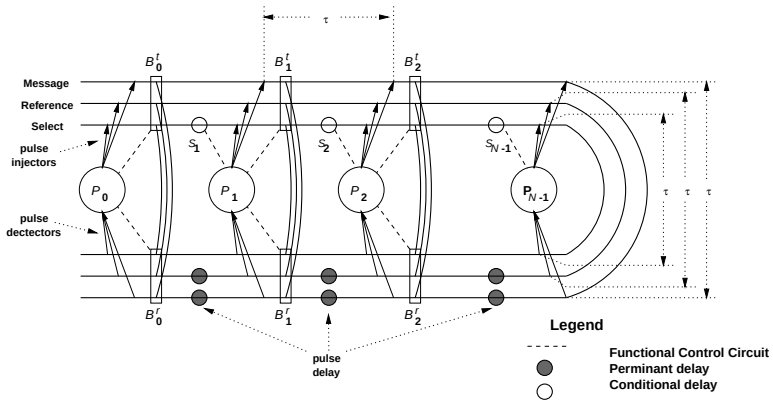

Figure 1. Overview of the LARPBS model

The reconfigurability of the optical bus is provided by the optical bus partition switches, B_i^t and B_i^r for $0 \leq i \leq N-2$ which directly connect the three waveguides from the transmission segment to the corresponding waveguides in the receiving segment, respectively, and at the same time, disconnects these waveguides as appropriate along the transmission and receiving segments. Thus, multiple independent optical buses can be configured. In this paper, $B_i^t = B_i^r = 1$ indicates a bus-split at P_i whereas $B_i^t = B_i^r = 0$ indicates that these switches do not partition the bus. For convenience, B_i will denote the combination of the two bus switches at P_i.

Let the propagation time of a pulse between any two processors be denoted by τ and let ω denote the duration of a pulse. In Figure 1, τ is illustrated as the constant ratio distance by the speed of light. For purposes of this study, the distance between the the pulse injector and the pulse detector for P_{N-1} is constant for all three waveguides and a pulse requires τ propagation time. In [1], for an optical signal of b bits, a collision-free requirement is that $\tau > b\omega$. To maximize the communication bandwidth, $b\omega$ should be as large as possible. A delay on the message and reference waveguides equal to ω is placed between every pair of detectors. In addition, *conditional* delay switches are placed between each pair of injectors on the select waveguide. Let $S_i, 1 \leq i \leq N-1$ denote a conditional switch. If $S_i = 1$ then a delay of ω is introduced and when $S_i = 0$, no delay is introduced. The set of all switches is denoted by $S = (S_1, S_2, \cdots, S_{N-1})$.

The *latency* L is defined as the sum of the propagation time of a pulse

along a waveguide emitted by a source processor P_i and detected by a destination processor P_j. The latencies for the message, reference and select waveguides, denoted by L_m, L_r and L_s respectively are given by

$$L_m = L_r = (2N - 1 - i - j)\tau + (N - j - 1)\omega \tag{1}$$

$$L_s = (2N - 1 - i - j)\tau + \omega \sum_{k=i+1}^{N-1} S_k \tag{2}$$

The *bus cycle* is defined to be the optical signal propagation delay from P_0 to P_0, and in the literature (see for example [1]) has been stated to be $2N\tau$.

LARPBS uses a *coincident pulse* addressing scheme such that when P_j detects a pulse on both the select and reference waveguides, P_j then reads the b bit message from the message waveguide. A coincident pulse occurs in two situations,

- firstly, whenever

$$L_r = L_s, \qquad \text{that is,} \qquad j = N - \sum_{k=i+1}^{N-1} S_k - 1 \tag{3}$$

- secondly, whenever P_i delays the injection of the select pulse by whole units of ω.

Since communication is based on coincident pulses, communication based on these two situations is exploited in the LARPBS model. Clearly from Eq. (3), the switch settings affect addressability of the communication.

3 Meta-Data in the LARPBS Model

Two modes of communication are possible in the LARPBS model. In the first, a message pulse available on the message waveguide can be utilized to communicate specific information; this is referred to as *informational communication*. The second mode of communication is referred to as *meta-data communication* and is detailed subsequently.

In the case where the message content of a series of messages reflects a pattern of distribution of the input data, the message pulse may be ignored and instead, infer the message contents at the destination processor. For this to be possible, some state information regarding either or both of the LARPBS communication parameters must be globally known. For the moment, the delay switch configurations on the select waveguide are assumed

globally known. Later on, the case that such information can be obtained in $O(1)$ time is presented.

Initially, let X denote the vector of input data: $X = (x_0, x_1, \ldots, x_{n-1})$ such that each $x_i \in X$ resides locally in processor P_i.

Let C denote a *meta-data configuration function* that provides for a partitioning of the switches according to the decision statement:

$$S_i = C(x_i) = \begin{cases} 1 & \text{if } C(x_i) \text{ is true,} \\ 0 & \text{if } C(x_i) \text{ is false.} \end{cases} \tag{4}$$

Thus, C will generate a particular distribution of switch settings.

C partitions $\boldsymbol{\mathcal{P}}$ into two subsets when $P_i, 0 \leq i \leq N-1$ initiate simultaneous pulses (at the beginning of a bus cycle) on the select and reference waveguides. Hence, $t_{\text{ref}_i} = t_{\text{sel}_i}, 0 \leq i \leq N-1$. The two subsets of $\boldsymbol{\mathcal{P}}$ are those which will detect a coincident pulse, denoted by $\boldsymbol{\mathcal{P}}^p$, and those which do not, denoted by $\boldsymbol{\mathcal{P}}^{np}$. This is evident from Eq. (3). Moreover, $(P_0, P_1, \ldots, P_j, P_{j+1}, \ldots P_{N-1})$ will be partitioned such that $\boldsymbol{\mathcal{P}}^{np} = (P_k | 0 \leq k \leq j)$ and $\boldsymbol{\mathcal{P}}^p = (P_k | j+1 \leq k \leq N-1)$. A grouping of the N coincident pulses therefore occurs at each $P_j \in \boldsymbol{\mathcal{P}}^p$. For example, if all $S_i = 1, 1 \leq i \leq N-1$, then $\boldsymbol{\mathcal{P}}^p = \boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{P}}^{np} = \{\}$ with each $P_j \in \boldsymbol{\mathcal{P}}^p$ detecting exactly one coincident pulse, whereas if all $S_i = 0, 1 \leq i \leq N-1$, then $\boldsymbol{\mathcal{P}}^p = \{P_{N-1}\}$ and $\boldsymbol{\mathcal{P}}^{np} = \boldsymbol{\mathcal{P}} - \{P_{N-1}\}$ with P_{N-1} detecting N coincident pulses. The cardinality of $\boldsymbol{\mathcal{P}}^p$ can be found by: $||\boldsymbol{\mathcal{P}}^p|| = 1 + \sum_{i=1}^{N-1} S_i$.

For each $P_j \in \boldsymbol{\mathcal{P}}^p$, a series of relations of the form $P_i \rightarrow P_j$ for some communication source $P_i \in \boldsymbol{\mathcal{P}}$ describes the grouping of coincident pulses detected by P_j. This series of relations can be extended to all $P_j \in \boldsymbol{\mathcal{P}}^p$. Consider a segment of switches $(S_s, S_{s+1}, \ldots, S_t, S_{t+1}), t \geq s$ such that $S_s = 1, S_{t+1} = 1$ and all $S_k = 0, s+1 \leq k \leq t$. Such a segment of switches is induced by C (ignoring the degenerate cases where $S_1 = 0$ or $S_{N-1} = 0$). All sources $P_s, P_{s+1}, \ldots, P_t$ will address the same destination processor over a corresponding unit of time. Hence, $P_s \rightarrow P_j, P_{s+1} \rightarrow P_j, \ldots, P_t \rightarrow P_j$.

Given the detection of a coincident pulse at P_j and the global state of the switches, P_i can be determined to be in the set of relations $P_i \rightarrow P_j$ for that P_j. Exactly, the meta-data communication from a source processor P_i to a particular destination processor P_j is described by

$$P_i \rightarrow P_{N-1-\Delta_j} \tag{5}$$

where Δ denotes a *time delay function* and is defined by:

$$\Delta_j = \sum_{k>j}^{N-1} S_k. \tag{6}$$

Essentially, Relation 5 is based on the addressing scheme of LARPBS where Eq. (6) determines the address of the recipient processor.

Every $P_j \in \mathcal{P}^p$ calculates Δ_j immediately following a switch reconfiguration (it is noted that Δ_j is computed by P_j). When P_j detects a coincident pulse, a table lookup is performed to determine P_i.

4 Case Study of the Meta-Data Communication in the Binary Prefix Sums Computation

The binary prefix sums calculation can be stated as follows. Given $X = (x_0, x_1, \cdots, x_{n-1})$ for $n \geq 1$ such that each $x_i \in \{0, 1\}$ for $0 \leq i \leq n-1$, compute a $Y = (y_0, y_1, \cdots, y_{n-1})$ such that $y_i = \sum_{j=0}^{j=i}(x_j)$ for $0 \leq i \leq n-1$. This calculation computes the number of '1's in a binary vector that occur previously to a given element in the binary vector, for example, if $X = (0, 1, 0, 1)$ then $Y = (0, 1, 1, 2)$. The sequential running time is $O(N)$. In [8] the parallel running time of prefix sums is given as of $\Theta(\log n)$. The LARPBS based algorithm is given in [1] and is $O(1)$ (bus-cycles).

As per the initial conditions, let X be the input binary vector such that $x_i \in X$ resides locally in processor P_i. The example shown in Table 1 and the associated LARPBS model shown in Figure 2 refers to $X = (1, 0, 1, 0, 1, 0, 1)$ and distributed as shown across the seven, P_0 through P_6, processors.

The decision statement for the first step of this algorithm is: "x_i equals 1.", hence, switch $S_i = 1$ if $x_i = 1$ and $S_i = 0, 1 \leq i \leq N-1$ otherwise. Table 1 illustrates the switch settings for this step in the row labeled Step 1; also, the switches that have had delays introduced are shown filled in Figure 2.

The first communication step of the algorithm has all $P_i, 0 \leq i \leq N-1$ initiate simultaneous pulses (at the beginning of a bus cycle) on the select and reference waveguides. Hence, $t_{\text{ref}_i} = t_{\text{sel}_i}, 0 \leq i \leq N-1$. Each processor, $P_j, N - ||\mathcal{P}^p|| \leq j \leq N-1$, computes the time delay function, Eq. (6), locally. The values computed for the example are shown in Table 1 in the row labeled Step 1. Each destination processor P_j determines the set of each source processors that initiated communication to it by Relation 5. For the example, Table 2 presents these calculations. For destination processor P_6, when it detects a coincident pulse, the set of possible source processors is $\{P_6\}$. However, when P_5 detects a coincident pulse, the set of possible source processors is $\{P_4, P_5\}$. In a similar way, all other processors can determine the source processor set corresponding to the detected coincident pulses. These sets are shown in the table entry labeled 'Time partition groups'. Lastly, the algorithm requires the source processor identification corresponding to the first coincident pulse. This will always be the highest number in the Time

Table 1. Binary prefix sums example ($B_i = 0$ unless otherwise noted).

Step	Description	Computation Results						
	Processors	P_0	P_1	P_2	P_3	P_4	P_5	P_6
	Distribution of X	1	0	1	0	1	0	1
1	Switch configuration	$S_1 = 0$	$S_2 = 1$	$S_3 = 0$	$S_4 = 1$	$S_5 = 0$	$S_6 = 1$	
	Time delay function			2	1	1	0	
	Time partition groups			$0, 1$	$2, 3$	$4, 5$	6	
	Reconstructed message			1	3	5	6	
2	Switch configuration	$S_1 = 0$	$S_2 = 0$	$S_3 = 0$	$S_4 = 0$	$S_5 = 0$	$S_6 = 0$	
	Message Unicasts		3		4		5	6
3	Bus configuration	$B_1 = 1$		$B_3 = 1$		$B_5 = 1$	$B_6 = 1$	
	Switch configuration	$S_1 = 0$	$S_2 = 0$	$S_3 = 0$	$S_4 = 0$	$S_5 = 0$	$S_6 = 0$	
	Message Broadcasts	3	3	4	4	5	5	6
4	Computation Results	4						
5	Switch configuration	$S_1 = 0$	$S_2 = 0$	$S_3 = 0$	$S_4 = 0$	$S_5 = 0$	$S_6 = 0$	
	Message Broadcast		4	4	4	4	4	4
6	Computation Results	1	1	2	2	3	3	4

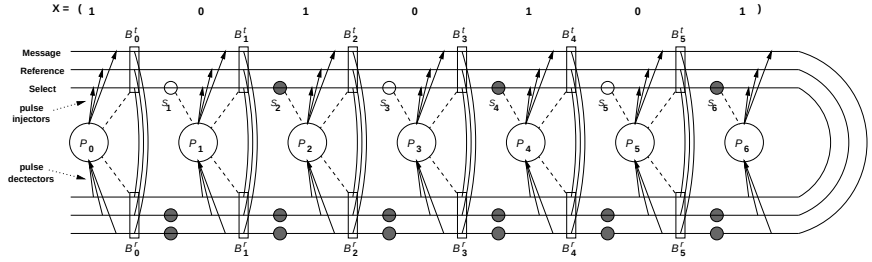

Figure 2. Binary prefix sums example: LARPBS model.

partition group. Thus, the original information can now be reconstructed. In the example, this is shown in the table entry labeled 'Reconstructed message'.

Importantly, the major difference between the algorithm given here during this step and the original given in [1] is that the original requires an information message to be carried via the message waveguide. In the original algorithm, this information message merely carries the processor identification of the source processor. In the method given here, the message waveguide is not utilized, rather, only the detection of the select and reference pulses is

Table 2. Calculation of Relation 5 for Step 1 in the example given in Table 1.

i	Δ_i	Relation 5	i	Δ_i	Relation 5	i	Δ_i	Relation 5	i	Δ_i	Relation 5
0	3	$P_0 \rightarrow P_3$	2	2	$P_2 \rightarrow P_4$	4	1	$P_4 \rightarrow P_5$	6	0	$P_6 \rightarrow P_6$
1	3	$P_1 \rightarrow P_3$	3	2	$P_3 \rightarrow P_4$	5	1	$P_5 \rightarrow P_5$			

required.

In the next step, the processor address for those having received coincident pulses during the first step must be communicated to the originating source processors. In the original algorithm, the message waveguide is utilized to transmit the values obtained during Step 1. Since this reflects informational communication, this step does not exhibit meta-data communication.

The third step requires splitting the optical bus into sub-buses, where, in each sub-bus, a localized broadcast is performed. Since there is informational communication, this step also does not exhibit meta-data communication. Step 3 in Table 1 illustrates this step for the example.

The next step requires a single computation performed by P_0 where $y = x_0 + (N - 1 - j)$ for j the value received by P_0 in Step 3. Step 4 in Table 1 shows the result of this computation for the example.

Step 5 requires a broadcast of the value computed by P_0 to all processors; no meta-data communication is exhibited. The results from the broadcast for the example are shown in Step 5 of Table 1.

The last step of the algorithm requires a local computation for each $P_i, 1 \leq i \leq N - 1$. This computation is $y = -(N - 1 - j)$ for j the value received by P_i from Step 5. The values distributed across the processors are the binary prefix sums for X. Table 1 displays the results for the example in Step 6.

In summary, the binary prefix sums algorithm on LARPBS exhibits both meta-data and informational data communication, where the meta-data communication occurs during the first step of the algorithm. Here, the meta-data configuration function was simply based on the binary value of the data input vector, X. Consequently, a pattern of switch settings was determined, thus impacting upon the latencies of subsequent communications. After the communications have completed, a set of receiving processors has obtained a series of coincident pulses. Assuming each recipient has knowledge of the global state of the switches, each recipient can calculate the originating processor identification for all coincident pulses. Such processor identification is used by the algorithm during subsequent processing. The version of the algorithm given here is also $O(1)$ bus-cycles.

5 Delay Switch Configuration

In the previous sections, it had been assumed that the switch configurations were globally known. This might be the case if, for example, the algorithm can be globally parameterized. The presentation of the binary prefix sums algorithm in Section 4 included qualitative descriptions of such global parameters for steps two through five: all switches set to 'off'. It is also possible that the switch control network be enhanced to provide for obtaining the switch settings by the processors. In general, however, the global state of the switch may not be known.

Consider Eq. (6) which describes the only parameter required to reconstruct the original message during meta-data communication. Eq. (6) essentially tallies the number of switches set to 'on' between P_j and P_{N-1} In fact, this is remarkably similar to the binary prefix sums algorithm presented earlier. Trivially, two further steps to the binary prefix sums algorithm may be added: a Step 7 where P_{N-1} broadcasts its computed value (4 in the example shown in Table 1) and a Step 8 where each processor subtracts the calculated prefix sum from the value broadcast in the previous step. Note that in fact the 'time delay function' (as in Table 1) can be computed from this procedure. Since the communication in Step 7 is $O(1)$, and if the original definition of the algorithm as given in [1] is used, then all necessary pre-information to complete a meta-data communication can be obtained by the processors in $O(1)$ bus-cycles. (optimizations may be considered in the algorithm to broadcast the switch states.)

6 Conclusions

Optical interconnection networks offer three advantages: high speed, high bandwidth and computational influence. The former two are well known advantages and essentially are the primary reasons for the evolutionary progress in this field. The latter advantage means that the nature of the communications has a positive performance effect on the computation beyond merely the high speed and bandwidth interconnect. This paper focused on the LARPBS optical bus model and specifically, on one computation primitive defined in that model. A case study based analysis was conducted of the Binary Prefix Sums computation primitive. The study was based on a notion of meta-data communication. The analysis reveals that the communication in the LARPBS model may be describable by such a meta-data model. On-going work includes the formalization of the meta-data model and its applicability in bus-based models.

Acknowledgements

The assistance of The Dept. of Math. & Comp. Sci. at the Univ. of Akron and the helpful suggestions from Dr. Abdullah Abonamah are acknowledged.

References

1. Yi Pan. Basic data movement operations on the LARPBS model. In K. Li, Y. Pan, and S.Q. Zheng, editors, *Parallel Computing Using Optical Interconnections*, pages 227–247. Kluwer Academic Publishers, 1998.
2. S. Pavel and S.G. Akl. On the power of arrays with optical pipelined buses. In H.R. Arabnia, editor, *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'96), Vol. III*, pages 1443–1453, Sunnyvale, California, USA, August 1996.
3. Y. Pan and K. Li. Linear array with a reconfigurable pipelined bus system — concepts and applications. In H.R. Arabnia, editor, *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'96), Vol. III*, pages 1431–1441, Sunnyvale, California, USA, August 1996.
4. J.L. Trahan, A.G. Bourgeois, Y. Pan, and Ramachandran Vaidyanathan. Optimally scaling permutation routing on reconfigurable linear arrays with optical buses. In *Second Merged Symposium IPPS/SPDP, 13th International Parallel Processing Symposium & 10th Symposium on Parallel and Distributed Processing*, San Juan, Puerto Roco, April 1999.
5. Y. Li, Y. Pan, and S.Q. Zheng. Pipelined time-division multiplexing optical bus with conditional delays. *Optical Engineering*, 36(9):2417–2424, September 1997.
6. S.P. Levitan, D.M. Chiarulli, and R.G. Melhem. Coincident pulse techniques for multiprocessor interconnection structures. *Applied Optics*, 29(4):2024–2033, 1990.
7. K. Li. Constant rime boolean matrix multiplication on a linear array with a reconfigurable pipelined bus system. *Journal of Supercomputing*, 11(4):391–403, 1997.
8. S.G. Akl. *The Design and Analysis of Parallel Algorithms.* Prentice Hall, 1989.