Optical Pipelined Multi-bus Interconnection Network Intrinsic Topologies

Brian Joseph d'Auriol

Digital all-optical parallel computing is an important research direction and spans conventional devices and convergent nano-optics deployments. Optical bus-based interconnects provide interesting aspects such as relative information communication speed-up or slow-down between optical signals. This aspect is harnessed in the newly proposed All-Optical Linear Array with a Reconfigurable Pipelined Bus System (OLARPBS) model. However, the physical realization of such communication interconnects needs to be considered. This paper considers spatial layouts of processing elements along with the optical bus light paths that are necessary to realize the corresponding interconnection requirements. A metric in terms of the degree of required physical constraint is developed to characterize the variety of possible solutions. Simple algorithms that determine spatial layouts are given. It is shown that certain communication interconnection structures have associated intrinsic topologies.

Keywords: Free-space optics, Interconnect topology, OLARPBS, Optical bus, Parallel computing model, Unconventional computing.

I. Introduction

Nowadays, there is renewed interest and a quickening pace of research and development of digital optics in computing. One area of interest in this field is the incorporation of optics in parallel computing models and systems. Such models can be broadly classified into two types: optoelectronic, where electronic processors are interconnected by an optical interconnection network, and all-optical, where optical processing elements replace electronic processors. In the former, the optoelectronic interface imposes the conventional separation of communications and computations, whereas in the latter, this separation is eliminated, resulting in a bits-in-flight operative mode. The reader is referred to [1], [2] for additional comments about optics in computing.

Optoelectronic models and systems are common in the literature. Advantages include an already mature installation base in supercomputers [3], [4], and large-scale data centers [4], [5], as well as in networking and telecommunications. Moreover, there is a mature foundation of parallel models and systems. Furthermore, there are additional advantages arising from the special properties of photonic-based communication technologies [6]. Examples include bus-based optical connections [4], [7], an early version of the ternary optical computer architecture [8], the C4 satellite cluster computation and operation system [9], and the suite of optical bus parallel computing models, of which 16 have been proposed (see [10]–[12] for more details) with the most recent work, the Parameterized Linear Array with a Reconfigurable Pipelined Bus System (LARPBS(p) [13], [14]. Other similar work includes the Multidrop Bus [15]. Optical Transpose Interconnection System (OTIS) (see [16] for recent work) uses both electronic interconnections for

Manuscript received Dec. 22, 2016; revised July 30, 2017; accepted Aug. 7, 2017.

Brian Joseph d'Auriol (dauriol@acm.org) is the head of Visual Enabled Computing Research Group, Anyang, Rep. of Korea.

short distances and optical interconnections for long distances.

There has also been recent work in digital all-optical computing models and systems. These include a recent version of the ternary optical computer architecture [17], the non-Turing all-optical computer in [18], the light-weight filter-based in-transit algorithm approach in [19], and the All-Optical Linear Array with a Reconfigurable Pipelined Bus System (OLARPBS) optical conduit parallel computing model [10], [11].

Optical interconnections may be realized by various technologies including fiber, conventional free-space, silicon photonics [4], polymers [4], [20], glasses, crystals, and metamaterials [20]. These technologies support either waveguided (that is, light is guided through the medium, such as fiber) or free-space (that is, light propagates in line-of-sight mode). Conventional free space makes use of bulk devices for large-scale deployments such as light sources, lenses, and mirrors. Recently, nano-optic devices that enable both guided and free-space modes are under development [20]. The signal-traverse path in free-space mode can be controlled using diffraction, reflection, and refraction methods [20].

Optical interconnects are used for various purposes: commonly to overcome the inherent disadvantages of electrical interconnects in current and future systems, that is, increased speed and bandwidth, higher link density, reduced power consumption, better heat-management requirements, no electromagnetic interference, low propagation loss, broadband impedance matching, high carrier frequency, and long-distance communications [4], [7], [15], [20]. In addition to these direct benefits, optical interconnects provide additional use capabilities. Delays along with optical splitters that enable controlled fan-outs are used in the non-Turing all-optical computer [18]. In many of the optical conduit and bus parallel computing models, including the OLARPBS model, processor addressing is accomplished using the coincident pulse-addressing technique, which essentially encodes the addresses in terms of a relative time dilation between two optical signals. The OLARPBS model also incorporates the unique requirement of adjusting the relative spatial locations of messages along different optical paths by, in particular, shortening or lengthening one optical path relative to another, subject to propagation time constraints [11]. However, beyond proposing delay buffers to support path lengthening [11], a general treatment of the physical realization of this model's requirement has not been considered. The problem is summarized as follows: how to physically realize multiple optical paths, each with time constraints, which interconnect a set of processing elements in a free-space optics environment.

There are several approaches to physically realizing such relative path shortening aimed at free-space implementations. One possible approach relies on adjusting the optical propagation speed in the light paths, for example, using the concept of microscopic optical buffers [21], the applications of slow light [22], [23], or by light storage [24]. However, it is uncertain whether such technology is feasible and capable of supporting the requirements in the near future, and moreover, they may require additional resources for its implementation with associated increased cost and complexity.

This paper focuses on the solution that involves physical spatial layout requirements of the free-space optical paths and the optical processing elements arising from time constraints due to the shortening and/or lengthening of optical paths. Essentially, the idea is to first determine the spatial locations of processing elements that are connected to the shorter path, and subsequently, to determine the spatial locations of those processing elements that are connected to the longer path. This idea is further applied to multiple pairs of optical paths. The resulting topologies may be further modified using macroscopic delay loops (this term is defined in [21]) to adjust the locations within bounded regions, thereby defining intrinsic topologies of the relative optical .path lengths.

Of the previously mentioned models, only the OLARPBS model incorporates this aspect; therefore, this model is selected as the focus model herein and an overview is presented in Section II. That being said, any application with similar relative optical path shortening and/or lengthening would have the same issues as what was identified in this work; thus, the techniques developed and studied in Section III are generally applicable in such cases. Visualizations of several OLARPBS bundle configurations are discussed in Section IV. Finally, conclusions are given in Section V.

II. OLARPBS Overview

The OLARPBS optical conduit parallel computing model [10], [11] consists of an organized conduit-bundle hierarchy of bus-based optical paths that connect groups of optical processing elements. An artist's concept drawing and architecture overview are shown in Fig. 1. The OLARPBS model is based on the earlier LARPBS (p) [13], [14], which in turn was based on the original LARPBS model [25]. A basic architecture consists of densely packaged VCSEL high data rate (currently 50 to 70 Gbps in the lab, expected 100 Gbps in the future)

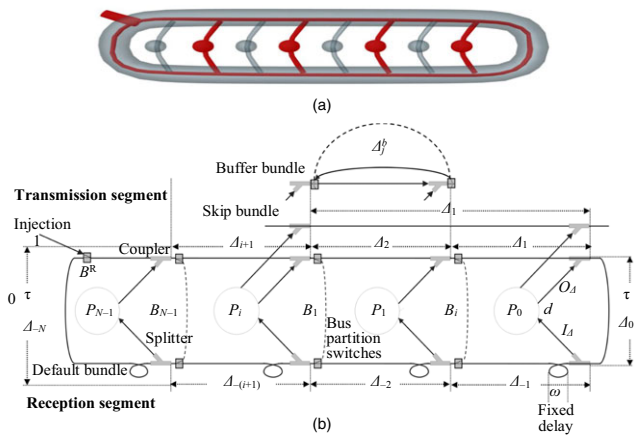

Fig. 1. OLARPBS architecture overview: (a) artist's concept of an eight-processing element architecture (gray) and a skip bundle (red), and (b) architecture schematic where three bundles are shown: the default bundle, a skip bundle, and a buffer bundle, along with processing elements, Delta subsegments, switches, delay loops, and architecture parameters are labeled.

light sources, guided or free-space optical paths, couplers, splitters, optical switches, delay paths (such as loops of fiber), as illustrated in the figure. Feasibility studies involving previously developed optoelectronic predecessor models provide evidence for its practicality [15], [26]–[32]; in particular, as the OLARPBS is designed based on some of the optical budget power recommendations stated in [32], it is estimated that the OLARPBS supports at least 15 processing elements. The unique and advantageous characteristics of the OLARPBS compared with predecessor models are very high bandwidth of the order of exa bits per second and beyond with modest high storage capability of the order of tera bits, which entails integrated bits-in-flight communications with serial and parallel computation. Further, it includes other advantages from predecessor models and all-optical technologies, including high speed and low latency. The OLARPBS model and its operations are abstracted by a number of parameters, thereby enabling the focus to be on system and algorithm development, as shown in the figure.

Processing elements are identified in one of two ways: by a unique pid number as well as by its location within conduits (the latter may be useful when processing elements connect with multiple conduits). There are four parameters that abstract the processing elements: operation description (such as addition) along with the three parameters IDelta, operation duration, and ODelta, which describe the duration of the input, processing element, and output data paths, respectively.

The communications interconnect is organized into a hierarchy with conduits consisting of bundles that comprise message optical paths along with two special optical paths called the reference and select paths. The latter two implement the coincident pulse-addressing technique, wherein the time dilation between optical signals on these two paths determine the destination processing element: a zero time dilation addresses processing element zero, and a time dilation of k integral units of Omega addresses the kth processing element, where Omega is the propagation time of a single bit, and is determined from the light source's data rate. Architecturally, this is achieved via delays of Omega time duration that are placed on the reference optical path, as shown in Fig. 1.

A bundle is described in terms of a transmission segment, which accepts the outputs from the processing elements, and a reception segment, which provides inputs to the processing elements. Importantly, a bundle is subdivided into subsegments of length Delta, where Delta is usually set (as is the case in this paper) as the constant physical distance between the couplers on the transmission segment and the splitters on the reception segment. The OLARPBS model identifies these Delta subsegments by subscripted Deltas based on the processing element identification, where positive subscripting identifies transmission subsegments, and negative subscripting identifies reception subsegments [10]. Messages are contained within Delta subsegments. At each successive time step, messages move from one Delta subsegment to the next according to a bundle update function, which defines the piece-wise Delta subsegment to subsegment connections for each bundle type. The propagation time, Tau, corresponds with Delta; and the message size is a function of Tau divided by Omega.

A taxonomy of bundle types appears in Fig. 2 and is briefly described below. There are three main bundle types: Delta order, skip, and buffer bundles.

Delta order bundles are bundles that connect each processing element in a specified Delta subsegment to Delta subsegment ordering, thereby establishing a linear array ordering of the processing elements. The default bundle [10], [11] is a Delta order bundle that is ordered in sorted decreasing order of processing element identifications.

Skip bundles [11] connect subsets of processing elements in the same order when compared with a reference bundle (usually the default bundle), where the parameter $d \geq 1$ describes the number of Delta subsegments in the reference bundle that connect the same two processing elements as in the skip bundle. The example conceptually illustrated in red in Fig. 1(a) illustrates a particular skip bundle of $d = 2$, that is, the

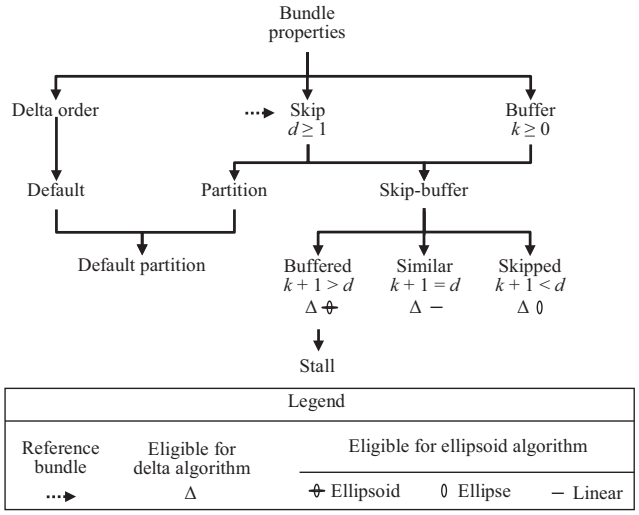

Fig. 2. Bundle taxonomy.

minimum configuration of skipping a single processing element. The degenerate case of one Delta-ordered bundle considered as a skip bundle with reference to the exact same Delta ordered bundle has $d = 1$. A more general skip bundle is illustrated in Fig. 1(b).

Partition bundles are a type of skip bundle where the bus partition switches are used to directly connect the transmission segment to the reception segment, thereby skipping all the lower numbered processing elements (see Fig. 1(b)). There is a single Delta subsegment for each of the partitions, and these are identified in the OLARPBS model with the superscript of B. Default-partition bundles are default bundles that include all of the partition Delta subsegments. Historically, most predecessor optical bus parallel computing models incorporated a single optical message path of this type.

Buffer bundles [11] include $k \geq 0$ additional Delta subsegments identified with the superscript of b which are used to buffer the optical messages, but which do not interconnect with processing elements.

Skip-buffer bundles combine the concept of a skip-bundle with buffer Delta subsegments, and serves as the basis for the spatial layouts considered later. There are three types: skipped $k + 1 < d$, which provides a path shortening compared with the reference bundle, and may contain zero or more buffer Delta subsegments; similar $k + 1 = d$, for which both bundles have paths of the same length; and buffered $k + 1 > d$, which provides a path lengthening compared with the reference bundle (as a technicality, note that the reference bundle is defined with respect to the skip bundle type, but not the buffer type). Note that the case where there are no Delta buffer subsegments $k = 0$ eliminates the buffered skip-buffer by definition, and it ensures that the similar skip-buffer meets the definition of the degenerate case of skip bundles; in all other cases, it reduces to skipped skip-buffer types. Additional examples are illustrated in subsequent sections of this paper.

Stall bundles [11] are a buffered skip-buffer type (that is, they include buffer Delta subsegments that increase the bundle path distance, thereby increasing the communication time) that implements a controlled stall of messages in the bundle by cycling the data in each Delta subsegment that is behind the stall point backwards using the additional buffer subsegments.

III. Bundle Spatial Layouts

In the following, a right-handed coordinate system is used; that is, x horizontal, y vertical, and positive z out of the page and towards the reader. The transmission segment is aligned in the x dimension and placed in the negative z half plane (that is, behind the screen) with the optical signal propagation in the positive direction (that is, left-to-right from lower to higher x values, and which also matches propagation in decreasing processing element identification order). Thus, the reception segment is also aligned parallel and placed in the positive z half plane (that is, in front of the screen) with the propagation flow in the opposite direction (that is, right-to-left). It is assumed that the minimum propagation distance from every processing element's input to output is Delta. It is convenient to align this plane with the x-z plane (that is, $y = 0$), and hence forth is termed the *primary plane*; Hereafter, this layout is referred to as the *primary layout*. This is consistent with the majority of diagrams used in the literature to describe the OLARPBS and predecessor models, which shows, as in Fig. 1(b), a top view of the architecture.

A folded paper metaphor is used as shown in Fig. 3 to discuss the development of the three-dimensional (3D) spatial layout. The surface plane of a paper represents an initial planar layout, where the line segment from A′ to B′

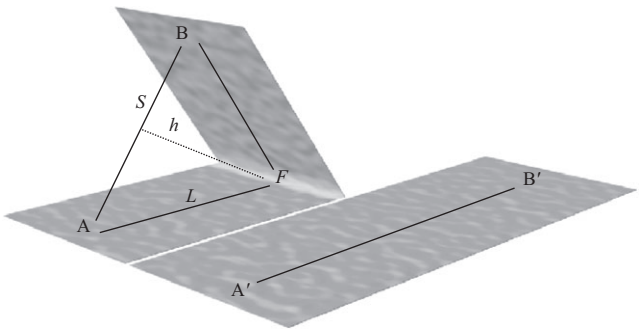

Fig. 3. Folded paper metaphor.

represents the initial optical path (bundle) from processing element A to processing element B. The paper is folded at point F so that the distance S from A to B represents the shorter of the optical bundle paths, and the distance L from A to F to B represents the longer of the bundle paths. Thus, the triangle AFB is formed and oriented such that A and B remain in the primary plane; however, F, which is the only point of inflection, is vertically displaced by h. Two simple algorithms are developed based on this metaphor and are hereafter referred to as Delta and ellipsoid algorithms to determine the spatial layouts of bundle configurations.

When given a starting and an ending Delta subsegment of two optical paths (bundles), the Delta algorithm determines whether the paths have the same length, or if not, it then identifies which has the shorter and which has the longer path length. There are three preconditions for this algorithm: first, two paths exist such that the set of processing elements connected by one is a subset of the set of processing elements connected by the other. Second, both paths are presorted in descending Delta subsegment order. Third, both paths share the highest Delta subsegment. The first precondition ensures that the starting and ending coordinates align when spatially laid out, that is, it identifies A and B in the folded paper metaphor. The second requires a consistent Delta subsegment identification convention between both paths, which is inherent in the definition for skip bundle types, but which may require appropriate buffer Delta subsegment labeling for buffer bundles. The third establishes a convenient starting point; nevertheless, a special case can be devised if this condition is not met. The essence of the algorithm is as follows: start at the highest-numbered Delta subsegments in both paths, iteratively search along both paths (in decreasing Delta subsegment order) to locate the same Delta subsegment identification, terminate the search when found, and record the number of Delta subsegments that were searched, setting S to the length of the shorter path and L to the length of the longer path.

The ellipsoid algorithm is applied after the preceding Delta algorithm. The inputs to this algorithm are S and L, and a starting x coordinate of S in the primary plane. The algorithm proceeds as follows. The first step is to choose the layout for S as it affects the subsequent layout for L; placing S at x in the positive direction results in a minimum of no inflection points (that is, a straight line). However, S may be laid out in connected piecewise linear segments, thereby shortening the effective distance denoted by S_e of S in the primary plane (that is, applying the folded paper metaphor in this case for S), but at the cost of an increased number of inflection points. Once the ending x coordinate of S is determined, the connecting processing elements corresponding to the starting and the ending locations of S are embedded in the primary plane (recall, with direction vector [0, 0, −1]). Next, if S and L are equal (that is, they have a similar skip-bundle), then L is laid out in a manner similar to S above (for example, an obvious layout would be coincident with S, and which is assumed to cause no interference).

Otherwise, L is spatially laid out as follows. In general, an ellipsoid with foci at $(x, 0, z)$ and $(x + S_e, 0, z)$ for $z = 0.5$ and $z = -0.5$ with major axis parameter $a = L/2$, minor axes parameters $b = c = \sqrt{a^2 - S_e^2/4}$, and eccentricity $e = \sqrt{1 - b^2/a^2}$ bounds the volume of all possible spatial layouts of L. The locus of points on the ellipsoid are the single points of inflection of all possible triangle paths. The locus of points such that the x coordinate is between the two foci represents a forward-motion optical signal path. However, for the remaining points on the locus, the optical signal must reverse direction with respect to its positive x-propagation direction. Interior points (that is, not on the locus of the ellipsoid) represent other possible points of inflection of the long path with a piece-wise segmented spatial layout (that is, not a single triangle). The use of such interior points compacts the bounding volume at the cost of increased complexity (for example, multiple reflective devices). An extreme case of such compacting uses macroscopic delay loops (for example, incorporating switches). The ellipsoid algorithm is thus parameterized as follows: choice of S layout, choice of triangle layout including forward- or reverse-motion triangle paths, choice of negative or positive y displacement in the case of skip bundles or the 3D path for buffer bundles, and choice of single or multiple inflection points. The ellipsoid is reduced to an ellipse in the $z = \pm 0.5$ planes in the case of skipped skip-bundle types as the long path L has associated connecting processing elements. See Fig. 4 for an illustration of the algorithm, and Fig. 5 for examples of ellipse regions.

The shape that the ellipsoid (for buffered skip-buffer types) or ellipse (for skipped skip-buffer types) takes on is illustrated in Fig. 6, which shows the eccentricity versus the long path length L. In addition, because the ellipsoid parameter a increases proportional with $2/L$, the bounding volume grows quickly. The shape for the maximum linear distance embedded into the primary plane (recall that this occurs when $S_e = S$) is bounded by the solid lines (blue bottom and red top). In the figure, the pair S, L denotes the lengths of the short and long paths, respectively. For example, 1, L indicates that the short path S has a fixed length of one Delta subsegment, while the long path varies in length (blue line). Associated with each point is an

https://doi.org/10.4218/etrij.17.0116.0948

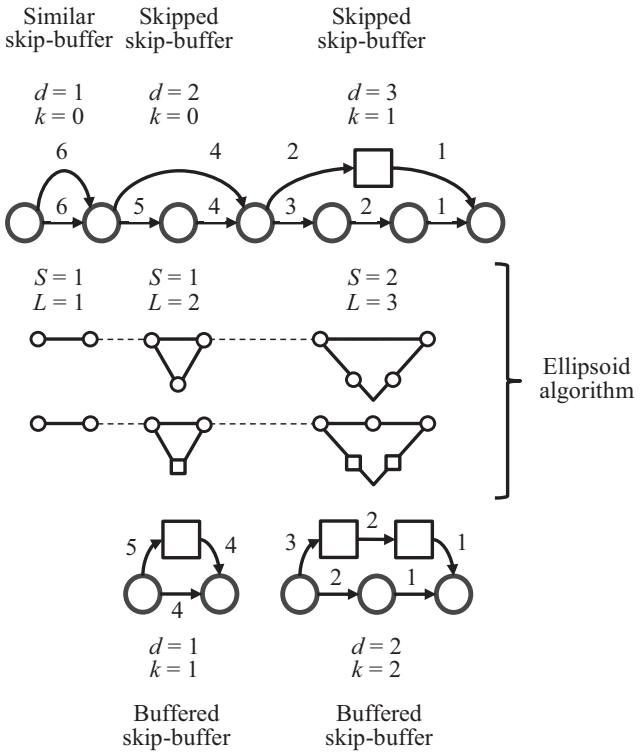

Fig. 4. Ellipsoid algorithm examples. Default (reference) bundle is shown with straight left-to-right arrows. Skip-buffer bundles are shown with curved arrows. Processing elements are shown by circles, and are placed at the end points of S. Further, they are placed along L for skipped skip-buffer types, but are placed along S for buffered skip-buffer types. Buffer subsegments are shown delimited by squares for convenience only. Dashed line indicates the primary plane. Line and triangles layouts are drawn to scale.

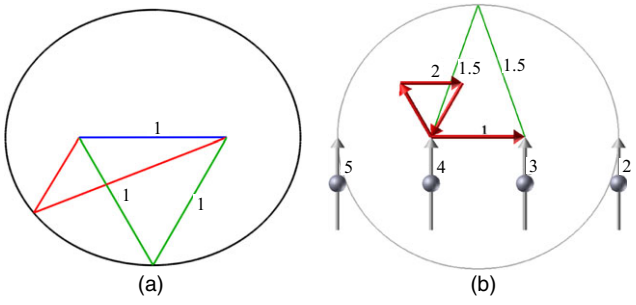

Fig. 5. Ellipse regions of triangle layout: ellipse boundary shown in gray, blue defines S (horizontal line in (a)), green defines L as an isosceles triangle path, red defines L as another possible path: (a) general ellipse layout, and (b) buffer bundle layout along with processing elements (spheres) and processing element data paths (gray).

ellipsoid or ellipse, depending on the skip-buffer type. The minimum skip configuration (recall that for the skip-bundle, $d = 2$) as well as the buffer bundle $k = 1$ are both

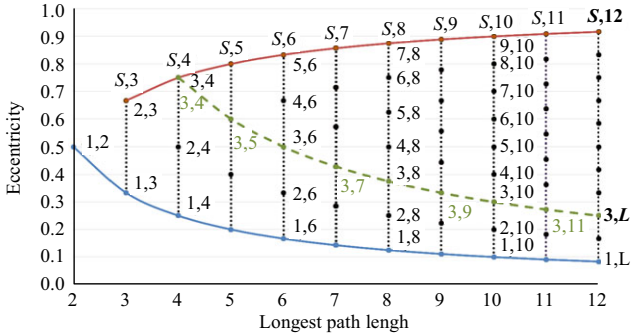

Fig. 6. Ellipsoid eccentricity for buffered and skipped skip-buffer bundle types. The notation S, L represents the lengths of the short and long paths, respectively.

represented by the point 1, 2 (that is, $S = 1$ and $L = 2$), and which has eccentricity of 0.5. The two triangle examples in Fig. 4 correspond with points 1, 2 and 2, 3, respectively. The effect of decreasing the effective distance (that is, $S_e < S$) lowers the eccentricity bounds (as shown by the dashed blue line, bottom, for $S_e = 0.5$) with corresponding decreases in the upper bound depending on the effective distances of layouts of the short path. However, compacting the shape by choosing interior (nonlocus) points more dramatically alters the bounding shape and needs to be computed for each specific case.

IV. OLARPBS Bundle Configurations

The application of the Delta and ellipsoid algorithms from the previous section will construct spatial layouts, one-at-a-time, in the order of the highest to lowest numbered Delta subsegments. Because either the default or default-partition bundles are commonly used as the reference bundle, the spatial placements of the processing elements are first determined in the transmission segment, and subsequently, in the reception segment. For buffered skip-buffer bundles, this poses no issues because only the bundle path, but not the processing elements, is spatially determined, as illustrated by the stall bundle shown in Fig. 7, where the buffer spatial layouts follow the pattern shown in Fig. 5(b).

For skipped skip-buffer bundles, the processing elements' spatial layouts form spatial constraints upon the overall bundle spatial layout. That is, spatial constraints are introduced for each pair-wise Delta subsegment to Delta subsegment connection that involves a processing element connection (that is, ellipsoid/ellipse placement). Beyond the orthogonal processing element path structure that was previously defined as part of the primary layout, there are three spatially connectable regions: those that occur only within the transmission segment (referred to as the transmission spatial region), those that occur only

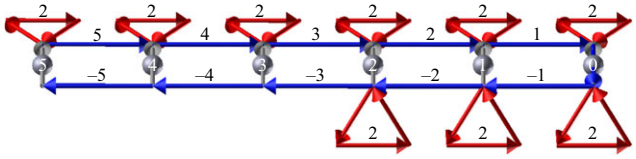

Fig. 7. Example of a $d = 1$, $k = 2$, $S = 1$, $L = 3$ stall bundle configured to stall at processing element 2. This example illustrates the spatial freedom of the ellipsoid algorithm by the $y = 0$ plane layout in the transmission segment, while the $z = 0.5$ plane layout in the reception segment. Buffer Delta segments are shown in red (which form triangles), and are based on the layout variations shown in Fig. 5(b). For Delta subsegments, the stall bundle coincides with the default bundle shown in blue up to processing element 2.

within the reception segment (reception spatial region), and those that occur between the transmission and reception segments (inter-spatial region).

Assuming consistent Delta and ellipsoid algorithm parameters, bundles that have the same processing element connections in the transmission and reception segments will result in spatial layouts that have the characteristic of graph reflection over the z axis, that is, the spatial layouts for the reception segment (in the positive z plane) is the reflection of that for the transmission segment (in the negative z plane). This is a special case of transmission and reception spatial regions, where the constraints introduced during the transmission segment processing are satisfied while processing the reception segment. Importantly, the default and default-partition bundles have this characteristic. Other instances of skipped skip-bundles may also have this characteristic, as does the eight processing element OLARPBS skip bundle of Fig. 1(a), a visualization of which appears in Fig. 8.

However, bundles that have different processing element connections in the transmission segment as compared with the reception segment become constrained by the processing element placements made in the transmission segment. There are two resulting cases: either the spatial layout in the reception segment follows the constrained path, that is, the constraints are satisfied, as exemplified in Fig. 9 and Fig 10. Alternatively, the constraints are not satisfied, and thus, adjustments to the processing elements' layouts need to be made, as exemplified in Fig. 11.

Bundles that have few constraints, as in the previous examples, may be conveniently adjustable (for example, in the latter, by simply reversing the Delta subsegment processing order in the Delta algorithm); however, more general solutions are needed for larger constraint systems. In the following, several more complex bundle structure

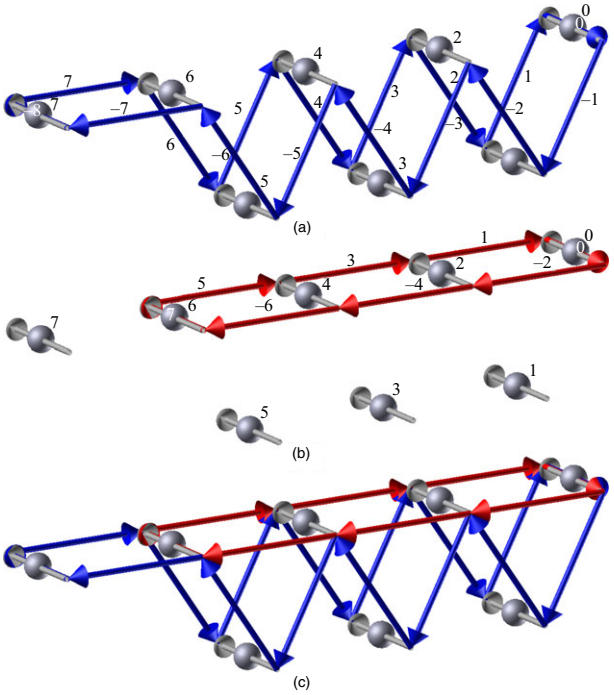

Fig. 8. Skip bundle layout of an $N = 8$ OLARPBS corresponding with Fig. 1(a) ($d = 2$, $k = 0$, $S = 1$, $L = 2$) using isosceles triangle layouts, default bundle shown in blue, skip bundle shown in red. (a) Default bundle (shown in gray in Fig. 1(a)), (b) skip bundle (shown in red and which is planar in Fig. 1(a)), and (c) combined default and skip bundles. Delta subsegment subscripts are labeled. Processing elements and their data paths are labeled and shown in gray. Arrows denote the optical message propagation direction.

patterns are considered with corresponding spatial layout solutions.

The first bundle pattern involves all Delta i (for some range of i) connected with a Delta j for fixed j less than i

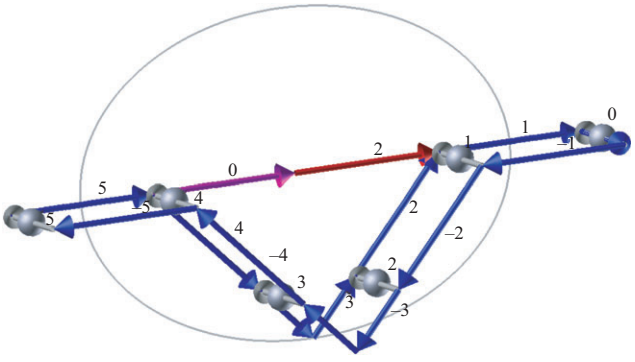

Fig. 9. Skipped skip-buffer bundle for $N = 6$. Blue path indicates default bundle; red path (labeled 2 horizontal), skip subsegment; and purple path (labeled 0 horizontal), skipped buffer subsegment. Ellipse (light-gray) marks the boundary area for the default bundle's long path spatial placement. Skipped skip-bundle will follow the default bundle path in the reception segment.

https://doi.org/10.4218/etrij.17.0116.0948

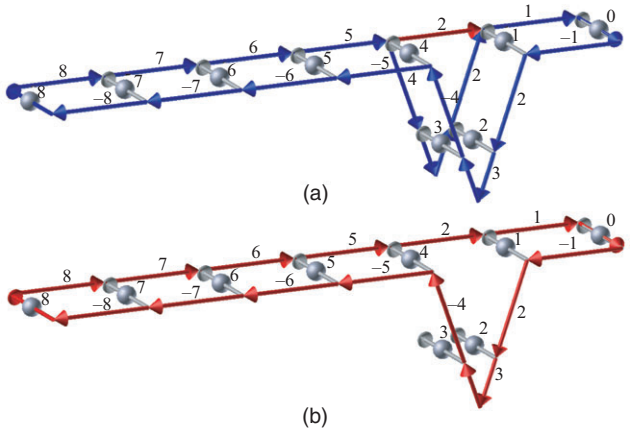

Fig. 10. Skipped skip-buffer bundle for $N = 9$ with a single skip in the transmission segment: (a) default bundle is shown in blue with the skip bundle's Delta two subsegment shown in red, and (b) skip bundle illustrates the constrained path in the reception segment.

and positive j. In other words, the skip occurs in the transmission segment, and hence, in the transmission spatial region. Here, there are multiple constraints such that Delta j is reachable, in single Delta subsegment lengths, from each Delta i. A circular layout oriented in the transmission z-plane (recall $z = -1/2$) is adopted, where the radius is set to one Delta. The Delta i processing element connections are placed at points on the locus of the circle such that the straight-line chords connecting neighboring processing elements (recall that the linear order is established by the default bundle) is strictly less than or equal to one Delta. This means that for a few constraints, processing elements are equidistant along an arc of the circle, but they do not go completely around with a chord distance of one Delta; however, for many constraints, processing elements are equidistant along the entire circle, but with chord distance less than one Delta. Subsequently, all chords must be normalized to a length of one Delta. To do so, the ellipsoid algorithm for which S_e is set to the chord distance is used to determine the specific layout, which may in fact generate out-of-plane paths. The processing element associated with Delta j is placed at the center of the circle. In essence, the Delta algorithm processing occurs as previously described, following the linear ordering of Delta subsegments—only in terms of a circular space instead of a linear one. A variation of this pattern permits graph reflection over z. Figure 12 illustrates this for i in the range of six to two with j equal to one.

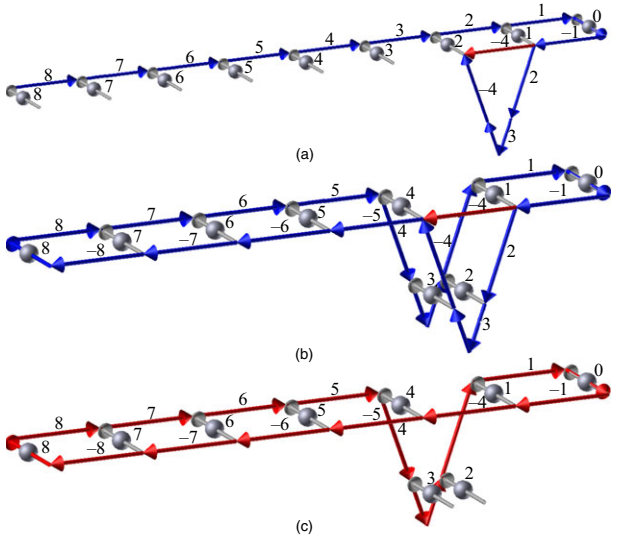

Fig. 11. Layout for a skipped skip-buffer bundle for $N = 9$ with a single skip in the reception segment: (a) Delta and ellipsoid algorithms layout the processing elements in the primary plane. However, there is a problem in the reception segment when the long path is placed in the vertical plane, but with no corresponding processing element placements; as a result, the skip short path connects to the wrong processing element, (b) corrected layout: default bundle is shown in blue with the skip bundle's Delta minus four subsegment shown in red, and (c) corrected layout: the skip bundle illustrating the constrained path in the transmission segment.

The second bundle structure is a generic all Delta to all Delta structure in each of the transmission and reception spatial regions. The approach is similar to that considered above. For the transmission and reception spatial regions, a circular layout oriented in the transmission and reception z-planes is adopted, where the diameter is set to the minimum length of all the shortest paths, that is, minimum over all S_i.

As previously mentioned, processing element connections are also placed at points on the locus of these circles in the same manner. Furthermore, skipped skip-buffer bundles that connect with non-neighboring processing elements will also have chord distances (with the exception of a chord coincident with the diameter) that are less than one Delta. As mentioned previously, all chords are normalized to a length of one Delta by using the ellipsoid algorithm. An advantage of this approach is the generic circular spatial layout that can be used for all bundle structures. However, this may be offset by the disadvantage of possibly over-constrained spatial layouts; for example, an application to the preceding bundle structure would result in the two left-most processing elements being needlessly placed within the circular layout.

The last bundle structure supports pairwise communication packing with the requirements as illustrated in Fig. 13. This bundle structure was theoretically proposed in [11] to support efficient vector parallel reduction operations on

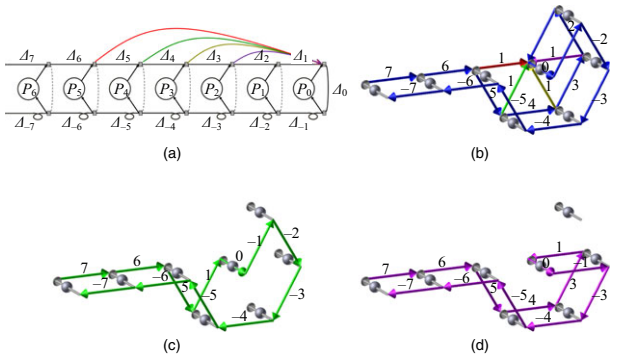

Fig. 12. OLARPBS all to one spatial layout example: (a) OLARPBS bundle schematic (skip colors match those in the visualizations). (b)–(d): spatial visualizations: processing elements shown in gray; default bundle in blue, Delta subsegments identified by text. Spatial layouts using equatorial triangle placements. (b) shows all skip bundles only in the transmission segment, (c) highlights the Delta five to one bundle only in the transmission segment, and (d) highlights the Delta three to Delta one and Delta-1 to Delta-3 (i.e., graph reflection).

the OLARPBS model (for example, $1 + 2 + 3 + 4 = (1 + 2) + (3 + 4)$), as used in a matrix multiplication algorithm; however, the physical realization was not addressed in the earlier work. The previous bundle structure solution is suitably applied here as there is no issue with over-constraints. In addition, it is combined with a single limited inter-spatial region around processing element zero. Thus, the skips that are confined in the transmission segment are laid out as per the circular spatial layout of diameter one (these are shown in dark blue, green, and purple in Fig. 13(a)). Moreover, the skips between the transmission and reception segments, of which there are three, are laid out as follows (these are shown in red and orange in Fig. 13(a)). To accommodate these three requires that the inter-plane distance be adjusted locally, that is, that the distance between the two planes be shortened. The specific distance is determined by normalizing the longest skip path length in the circular spatial layout and adjusting the processing element's path accordingly.

As previously mentioned, the ellipsoid algorithm is applied to all Delta subsegments other than those that are already normalized to determine the actual path.

V. Conclusions

Digital optics combined with parallel computing in an all-optical computing environment is an important research direction for future high-performance computing systems. When combined with related optical, nano, and quantum convergent technologies, it suggests an exciting

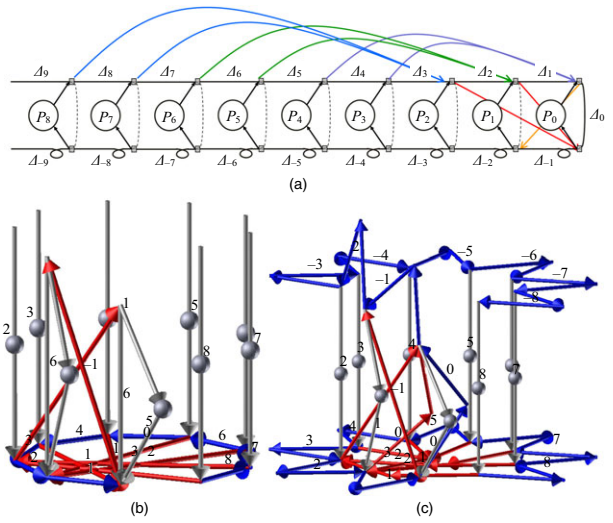

Fig. 13. OLARPBS pairwise communication packing spatial layout: (a) OLARPBS bundle schematic, where each pairwise skip is shown in matching colors, (b) and (c) spatial visualizations: processing elements shown in gray; default bundle in blue, skip paths in red, Delta subsegments identified by text, (b) simplified circle layout illustrating the default path in the transmission segment, the *z*-plane confined chords, and the adjustment of processing elements zero and one, and (c) actual spatial layout using isosceles triangle placements.

future era of high-bandwidth, high-speed, and high-storage capable computing infrastructures. Digital optic communication buses can be harnessed to provide interesting aspects in addition to only communication; aspects such as information encoded by time differentials, or increases or decreases in the relative information communication speed between optical signals. This paper examines the physical spatial layout requirements to support these aspects for free-space based infrastructures that are suitable for conventional bulk or future nano-based free-space systems. Various spatial layout structures that vary in degrees of spatial constraints have been proposed in this paper. The OLARPBS parallel model [10], [11] is designed to take advantage of these aspects. As such, the work reported in this paper addresses the physical spatial realization of OLARPBS communication structures, which was lacking in previous works.

A potential point for future consideration is suggested, in which the development of interconnection structures should be accompanied by its physical layout topologies.

References

[1] Y.A. Zaghloul, A.R.M. Zaghloul, and A. Adibi, "Passive All-Optical Polarization Switch, Binary Logic Gates, and

Digital Processor," *Opt. Exp.*, vol. 19, no. 21, Oct. 2011, pp. 20332–20346.

[2] R. Athale and D. Psaltis, "Optical Computing: Past and Future," *Opt. Photon. News*, vol. 27, no. 6, June 2016, pp. 1–7.

[3] C. Schow, F. Doany, and J. Kash, "Get on the Optical Bus," *IEEE Spectr.*, vol. 47, no. 9, Sept. 2010, pp. 32–56.

[4] N. Bamiedakis et al., "A 40 Gb/s Optical Bus for Optical Backplane Interconnections," *J. Lightw. Technol.*, vol. 32, no. 8, Apr. 2014, pp. 1526–1537.

[5] C. Kachris, K. Kanonakis, and I. Tomkos, "Optical Interconnection Networks in Data Centers: Recent Trends and Future Challenges," *IEEE Commun. Mag.*, vol. 51, no. 9, Sept. 2013, pp. 39–45.

[6] B. Jalali et al., "Silicon Photonics Coprocessors for Energy Efficient Computing," *Conf. Opto-Electron. Appl. Opt. (IEM OPTRONIX)*, Vancouver, Canada, Oct. 16–17, 2015, pp. 1–2.

[7] L. Li, "Design and Evaluation of Optical Bus in High Performance Computer," *Bullet. Adv. Technol. Res.*, vol. 3, no. 3, Mar. 2009, pp. 32–39.

[8] J. Yi, H. Huacan, and L. Yangtian, "Ternary Optical Computer Architecture," *Phys. Scripta*, vol. 2005, no. T118, 2005, p. 98.

[9] B.J. d'Auriol and T. Ghosh, "A Systems Model for Computation, Communication, Command and Control (C4) in a Spacecraft or Satellite Cluster," *Int. Conf. Parallel Distrib. Comput., Applicat. Technol.*, Taipei, Taiwan, Dec. 2006, pp. 285–290.

[10] B.J. d'Auriol, "All-Optical Linear Array with a Reconfigurable Pipelined Bus System (OLARPBS) Optical Bus Parallel Computing model," *J. Supercomput.*, vol. 72, no. 2, Feb. 2016, pp. 753–769.

[11] B.J. d'Auriol, "High Bandwidth Flexible Interconnections in the All-Optical Linear Array with a Reconfigurable Pipelined Bus System (OLARPBS) Optical Conduit Parallel Computing Model," *J. Supercomput.*, vol. 73, no. 2, Feb. 2017, pp. 900–922.

[12] B.J. d'Auriol and M. Beltran, "A Historical Analysis of Fiber Based Optical Bus Parallel Computing Models,'' *Scalable Comput.: Practice Experience*, vol. 7, no. 1, Mar. 2006, pp. 115–125.

[13] B.J. d'Auriol and R. Molakaseema, "A Parameterized Linear Array with a Reconfigurable Pipelined Bus System: LARPBS(p)," *Comput. J.*, vol. 48, no. 1, Jan. 2005, pp. 115–125.

[14] B.J. d'Auriol, "The Systems Edge of the Parameterized Linear Array with a Reconfigurable Pipelined Bus System (LARPBS(p)) Optical Bus Parallel Computing Model," *J. Supercomput.*, vol. 1, July 29 2008, pp. 183–209.

[15] M.R. Tan et al., "A High-Speed Optical Multidrop Bus for Computer Interconnections," *IEEE Micro*, vol. 29, no. 4, July 2009, pp. 62–73.

[16] B.A. Mahafzah et al., "The OTIS Hyper Hexa-Cell Optoelectronic Architecture," *Comput*., vol. 94, no. 5, 2012, pp. 411–443.

[17] Z. Shen, L. Wu, and J. Yan, "The Reconfigurable Module of Ternary Optical Computer," *Optik-Int. J. Light Electr. Opt.*, vol. 124, no. 13, July 2013, pp. 1415–1419.

[18] K. Wu et al., "Fiber Non-turing All-Optical Computer for Solving Complex Decision Problems," *Conf. Int. Quantum Electron. Conf., Lasers Electro-Opt.*, Munich, Germany, May 12–16, 2013, pp. 1–1.

[19] J. Touch et al., "A Candidate Approach for Optical In-Network Computation," *2016 IEEE Photon. Soc. Summer Topical Meeting Series (SUM)*, Newport Beach, CA, USA, July 11–13, 2016, pp. 8–9.

[20] Y. Fainman et al., "Nanophotonics for Information Systems," in *Information Optics and Photonics, Algorithms, Systems, and Applications*, New York, USA: Springer, 2010, pp. 13–37.

[21] A.V. Dmitriev, N.A. Toropov, and M. Sumetsky, "Miniature Optical Delay Lines and Buffers,'' *Int. Conf. Transparent Optical Netw.*, Trento, Italy, July 10–14, 2016, pp. 1–3.

[22] L. Thévenaz, "Slow and Fast Light in Optical Fibers: Review and Perspectives," *Conf. Lasers Electro-Opt. Conf. Quantum Electron. Laser Sci. Conf.*, Optical Society of America, Washington, DC, USA, June., 2–4, 2009, pp. 1–2.

[23] L. Thévenaz, "Fundamental Aspects of Linear Slow Light Systems," *Int. Workshop Opt. Wave Waveguide Theory Numerical Modelling*, London, UK, Apr. 17–18, 2015, p. 68.

[24] B. Gouraud et al., "Demonstration of a Memory for Tightly Guided Light in an Optical Nanofiber," *Phys. Rev. Lett.*, vol. 114, no. 18, May 8, 2015, pp. 180503:1–180503:5.

[25] Y. Pan and K. Li, "Linear Array with a Reconfigurable Pipelined Bus System — Concepts and Applications," *Int. Conf. Parallel Distrib. Process. Tech. Applicat.*, Sunnyvale, CA, USA, Aug. 1996, pp. 1431–1441.

[26] D.M. Chiarulli, R.G. Melhem, and S.P. Levitan, "Using Coincident Optical Pulses for Parallel Memory Addressing," *IEEE Comput.*, vol. 20, no. 12, Dec. 1987, pp. 48–58.

[27] R.G. Melhem, D. Chiarulli, and S. Levitan, "Space Multiplexing of Waveguides in Optically Interconnected Multiprocessor Systems," *Comput. J.*, vol. 32, no. 4, 1989. pp. 362–369.

[28] S.P. Levitan, D.M. Chiarulli, and R.G. Melhem, "Coincident Pulse Techniques for Multiprocessor Interconnection Structures," *Appl. Opt.*, vol. 29, no. 4, 1990, pp. 2024–2033.

[29] D.M. Chiarulli et al., "An All Optical Addressing Circuit: Experimental Results and Scalability Analysis," *J. Lightw. Technol.*, vol. 9, no. 12, Dec. 1991, pp. 1717–1725.

[30] D. Chiarulli et al., "Optoelectronic Buses for High-Performance Computing," *Proc. IEEE*, vol. 92, no. 11, Nov. 1994, pp. 1701–1709.

[31] S. Zheng et al., "Generalized Coincident Pulse Technique and New Addressing Schemes for Time-Division Multiplexing Optical Buses," *J. Parallel Distrib. Comput.*, vol. 61, no. 8, Aug. 2001, pp. 1033–1051.

[32] B.J. d'Auriol and J.R. Roldán, "An Optical Power Budget Model for the Parameterized Linear Array with a Reconfigurable Pipelined Bus System (LARPBS(p)) Model," *J. Parallel. Distrib. Comput.*, vol. 69, no. 10, Oct. 2009, pp. 815–823.

Brian Joseph d'Auriol received his BSc. (CS) and Ph.D. degrees from the University of New Brunswick (Canada) in 1988 and 1995, respectively. He has held various research and professorial appointments and positions in computer sciences and engineering at: SUNY Korea and Kyung Hee University in the Rep. of Korea; Ohio Supercomputer Center, The University of Texas at El Paso, The University of Akron, and Wright State University in the USA; and the University of Manitoba and the University of New Brunswick in Canada. He organized and chaired the CIC2000-CIC2008 international conferences in the USA and the HPCS'97 international symposium in Canada, and he was involved in varying capacities with many other international professional venues. He has published over 90 peer-reviewed papers in international journals and conference proceedings. His areas of research include insightful serviceable visualizations aimed at understanding and knowledge via learning, insight, impression, and emotion to facilitate creativity and decision-making; exascale and beyond bandwidth optical conduit-based parallel computing; and the design of a personal visualization assistant, which is a big data application. He is a member of the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers IEEE (Computer Society).

https://doi.org/10.4218/etrij.17.0116.0948