Channel Routing for Integrated Optics

Christopher Condrat (chris@g6net.com)
Priyank Kalla (kalla@ece.utah.edu)
Steve Blair (blair@ece.utah.edu)

Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT, USA

***Abstract*—Increasing scope and applications of integrated optics necessitates the development of automated techniques for physical design of optical systems. This paper presents an automated, planar channel routing technique for integrated optical waveguides. Integrated optics is a planar technology and lacks the inherent signal restoration capabilities of static-CMOS. Therefore, signal loss minimization—as a function of waveguide crossings and bends—is the primary objective of this technique. This is in contrast to track and wire-length minimization of traditional VLSI routing. Our optical channel router guarantees minimal waveguide crossings by drawing upon sorting-based techniques for waveguide routing. To further improve our solutions in terms of signal loss, we extend the router to reduce the number of bends produced during routing. Finally, we implement the optical channel routing technique and describe the experimental results, comparing the costs of routing solutions with respect to waveguide crossings, bends, and channel height.**

1. Introduction

Recent breakthroughs in silicon-based integrated optics – *Silicon Photonics* – are establishing the viability of silicon for integrated optics. The use of silicon enables fabs to leverage already existing and mature silicon processes and infrastructure for optical device fabrication as well as integration for electro-optical systems. Investment in Si-photonics integration is significant [1], [2]; also significant are the open foundary initiatives and developmental programs such as the OPSIS framework [3]. These developments are enabling applications far beyond traditional roles of optics in communications – such as *optical routing and photonic networks-on-chips* [4], *signal processing* [5], and also *optical digital logic* [6], [7], *quantum and reversible computation* [8]–[10].

As the availability and applications of integrated optics expand, the need for automated design space exploration, optimization, and physical synthesis of integrated electro-optical systems is also beginning to appear. For this reason, the Electronic Design Automation (EDA) community is investigating how automatic design space exploration techniques can be adapted to the photonics domain [6], [11]–[13]. This paper also takes a step in this direction and presents a methodology and solutions for detailed routing of integrated optical waveguides. In particular, we show that the detailed routing problem manifests itself as a *channel routing* problem, where (Silicon) optical waveguides are fabricated on a planar substrate and are connected to devices at the ends of the channel[1]. Planar routes require waveguides to bend (curve) and cross each other—causing loss of signal power. Channel routing techniques are therefore needed that minimize waveguide crossings and bends. Drawing inspirations from sorting-based routing techniques [14], [15], we present an efficient solution for channel routing of integrated optical waveguides that minimizes signal loss as a function of waveguide crossings and bends within the channel.

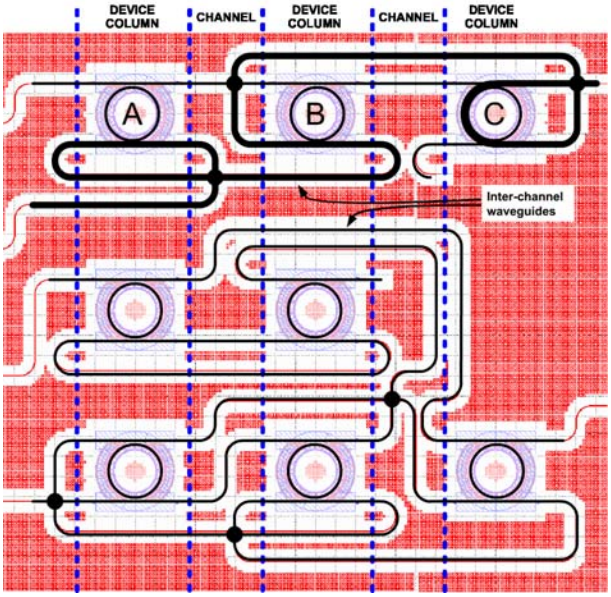


Fig. 1: Routing channels of optical GDS layout

A. Optical Routing Problem Formulation and Objectives

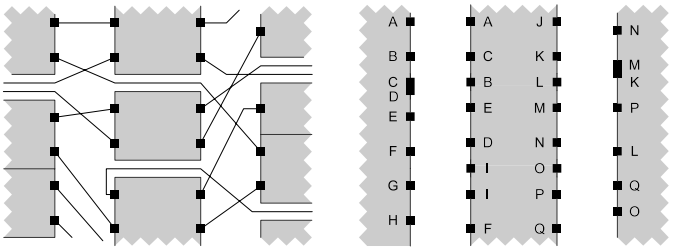
The main motivation for solving this problem stems from physical design of integrated optical logic circuits [6], [7], [10], [16]. Such circuits comprise a set of pre-designed optical devices — such as modulators, switches, splitters and detectors — placed on a planar substrate, and connected together with waveguides. Consider the optical network depicted in Fig.1. Eight (8) ring resonators are arranged into columns by a device placer such as to minimize area as well as routing complexity. The column arrangement induces the presence of vertical routing regions between device columns denoted as **channels**,

This research was funded in part by a sub-contract to AFOSR grant FA9550-09-1-0661.

[1]In the VLSI domain, channel routing is no more a topic of extensive research investigations due to the availability of a large number of metal layers and over-the-cell routing. This paper revisits channel routing specifically for optical technology, which introduces new optimization criteria not addressed by VLSI channel routers.

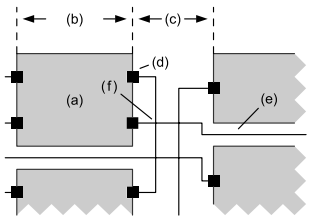
UU IR Author Manuscript

with device connection-points, denoted as **ports**, facing the channels. Inter-channel waveguides are used to allow routes between devices in other channels.



(a) Placement of devices with global routes

(b) Resulting channel for detailed routing



(c) Detailed routing formulation

Fig. 2: Channel routing for an optical device network

In general, for a layout such as Fig.2(c), pre-designed optical devices are represented as rectangular blocks (a) that are arranged (placed) in fixed-width columns (b). Such a placement gives rise to vertical routing channels (c), which are routing regions that separate the placed devices. Waveguides are routed between devices at ports (d) that face the channels. For routing between ports in different columns, connections are made to inter-channel waveguide, as depicted in (e). In addition, due to the planar nature of waveguide, waveguides may cross each other, but only perpendicular to each other (f) — at the cost of a small amount of signal loss.

The waveguide connecting two ports is denoted as a **net** and comprises a single route with no signal sharing (fanout). Signal sharing is explicitly provided by pre-designed *waveguide splitter devices*; these devices are treated as placed, 3-port, pre-designed optical devices, with ports for routing. Therefore, our methodology renders every net a **two-terminal net** within the channel.

This paper is concerned with channel routing and not with device placement. It is assumed that a (column-based) placement of optical devices is already given, along with the general routing path/topology of optical signals. This, subsequently, gives rise to a channel routing problem — such as the one depicted in Fig.2 — which we solve while minimizing signal loss.

B. Signal Loss

We identify signal loss as the primary guiding metric in integrated optics routing. All devices, including bulk waveguides, have insertion losses measured in decibels (dB). These losses are pre-characterized through device analysis (e.g. FDTD modeling and simulation) and technology parameters.

In terms of planar routing, we identify the following loss mechanisms:

- **Waveguide crossings [0.1–0.2 dB / crossing]** Per-crossing losses are on the order of 0.1–0.2dB per crossing [17], [18], affecting both crossing waveguides.
- **Waveguide bends [0.001–0.3 dB / bend]** Losses dependent on inherent waveguide properties (materials, geometry, etc.), radius of curvature of the bend, and surface roughness due to fabrication [19]–[21].
- **Bulk waveguides [0.01–2 dB / cm]** As these losses are extremely low (dB per *centimeter*, e.g. 0.03dB/cm [22]), we consider bulk waveguides essentially *lossless*.

Optimization Objective: The primary optimization objective in our routing formulation is *signal loss* minimization. Within the channel, this is achieved by 1) minimization of the total number of waveguide crossings; 2) minimization of the number of waveguide bends. Minimization of the number of tracks (channel height) is the subsequent secondary objective.

We optimize for the *total signal loss within the channel* due to optical feedback within the system. For example, consider the 1-bit full-adder circuit implemented using ring-resonator-based switches in silicon photonics in Fig.1. We designed this circuit and fabricated it through OpSIS [3]; the design is currently under testing and characterization. A signal may be routed such that it enters a given channel multiple times, as depicted in the highlighted signal path. Therefore, instead of minimizing losses on a per-net basis, we minimize for *total* losses within a channel.

C. Contributions of this work

This paper presents methods for channel routing of integrated optical waveguides fabricated on a planar substrate. Our channel routing technique is based on non-Manhattan routing grids and positional net sorting. We draw inspirations from a sorting-based channel router [14], [15] that has the useful property of being *minimal in terms of crossings*. However, the original sort-router formulation suffers from a number of unaddressed limitations and detrimental side-effects that make it impractical for optical routing. We show that there are fundamental flaws in the way the swap/sort-routing channel problems are encoded, requiring excessive area and introducing more waveguide bends.

We overcome these problems with our own sorting-based router that: 1) operates on both sides of the channel; 2) better analyzes the position of other nets and routing area; and 3) constrains the swap/sort operation to avoid unnecessary route detours. As a result, our router not only retains minimal crossings, but further minimizes the number of waveguide bends. Track utilization is also improved over the original technique.

Our channel routing techniques are then applied to a number of large channel routing problems. We evaluate the techniques with respect to each other in terms of crossings and channel height.

2. Previous Work

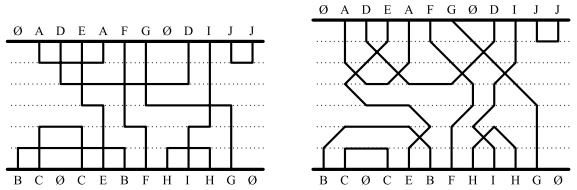
State-of-the-Art in Photonics Design Automation: One of the main focus of current investigations is toward *archi-*

tectural explorations for photonic interconnection networks in multi-core processor systems [4], [12], [23]–[25]. At the functional/logic-design level, there have been investigations into use of optical components as building-blocks, connected by waveguides, to design optical computing systems [6], [8], [10], [16], [26], [27]. High-level synthesis, using technology-mapping with a library of optical device building-blocks, has also been presented [11]. The focus of these works is on architectural and functional analysis and optimization; physical design and fabrication details are beyond the scope of such works.

At the much lower (physical) level, [13] demonstrates a *full-custom layout* of photonic structures using a commercial CMOS-based layout editor (Cadence Design Systems Virtuoso). Waveguide curves are discretized at a fine level into rectangular geometry, enabling waveguides to be represented in a format that traditional foundries accept. This methodology is significant in that it provides a building-block pathway for producing foundry-ready layouts and masks for non-Manhattan device geometries (rings, arcs, waveguide curvature). However, for such methodologies, design automation is essentially absent, and must be optimized manually. Similarly, the commercially available RSoft [28] Photonics CAD suite provides a framework for physical device design, analysis and (FDTD) simulation engines for performance analysis of optical design components. However, automated techniques for design space exploration during physical synthesis – *automated floorplanning, placement, waveguide routing while optimizing for physical parameters such as insertion-loss, bend-loss, phase coherence issues, etc.* – are not available.

Recently, [12], [29] present techniques for global optical interconnect synthesis. Such techniques analyze the routing problem at different levels of abstraction than the techniques presented in this work. Once global routing is performed, local routing is necessary to complete routing. At this routing level, a channel router may be utilized to ensure crossing-minimality, as well heuristically minimal bends.

In VLSI physical design, channel routing algorithms [30]–[32] are textbook knowledge [33], [34]. Crossing minimization in routing has been studied in the context of the crossing distribution problem (CDP) [35], [36]. The CDP is concerned with the distribution of a minimal set of crossings within a routing topology; this is performed through permutations of net orderings. In contrast, while our work also utilizes net orderings to ensure crossing minimality, the final routing is performed with the goal of reducing signal loss with respect to bends — not the distribution of crossings within the routed channel. Channel routing with crossing minimization has also been studied in the context of QCA routing [37]. Track assignments for multi-terminal nets induce varying numbers of crossings; therefore, [37] formulates crossing minimization, heuristically, as a weighted-minimum-feedback-edge-set problem. However, in the context of our problem—utilizing exclusively 2-terminal nets—we exactly minimize waveguide crossings, obviating the need for such an approach.



(a) Track-optimized (b) 2-sided swap (2Swap)

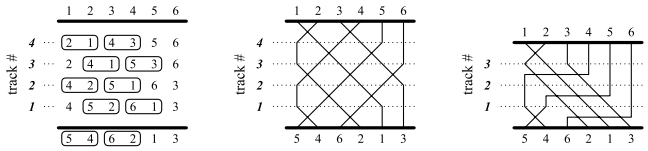
(c) Constrained 2-sided swap (2SwapC)

Fig. 3: Routing solutions for the same channel instance

3. Non-Manhattan Grid, Sorting-based Routing

A channel routing problem is represented by net "pins" fixed to the top and bottom of a channel. The purpose of the channel router is to route all nets in the channel's routing region, while minimizing parameters such as area (channel height, number of tracks), signal delay (net-length), or signal loss. Fig.3(a) depicts a minimum track channel routing obtained by a left-edge router. In traditional VLSI channel routing, area and net-length are primary optimization goals. More recently, however, channel routing also seeks to minimize other objectives, such as vias, and in the case of this paper, signal loss.

Manhattan-based (rectilinear) grids are traditionally employed in VLSI routing, dedicating layers specifically to horizontal or vertical spans for routing flexibility; non-Manhattan-based grids (e.g. octilinear) are rarely utilized, except in local cases. Integrated optics, however, is well suited to non-Manhattan-based routing grids. Such routing grids can more suitably represent waveguide curves, and provides greater routing flexibility in the absence of multiple routing layers.



(a) Swap-sorted net indexes. (b) Swap-sorted solution. (c) Bubble-sorted solution.

Fig. 4: Channel routing performed by sorting indexes. Circled indexes denote a pair that is reordered (sorted).

The work of [14], [15], also described in textbook [34], investigates a non-Manhattan grid channel router based on *sorting*. The nets of a channel are assigned numerical indexes, and routing is performed by sorting the nets in a finite number of permutations. The number of permutations performed represents the number of tracks utilized. Examples of this sorting-based routing are depicted in Fig.4.

Crossing minimality: In addition to utilizing non-Manhattan grids, sort-router's channel solutions are minimal in terms of the number of crossings. Crossing minimality results from the

fact that [14]: 1) crossings only occur if nets are positioned out-of-order during sorting, and 2) once sorted, pairs of nets *never cross each other again during the sorting process.*

The flexibility of a non-Manhattan-based grid and crossing minimality makes sorting-based routing attractive for integrated optics. However, the original sorting-based channel routing solutions presented in [14], [15] have drawbacks that make them impractical. Below, we describe the limitations of the sort routers of [14], [15]; these limitations motivate the design of our own sorting-based channel router, specifically designed for integrated optics.

A. Sorting-based Channel Routing

Two sorting techniques are presented in [14], [15]: swap-sorting and bubble-sorting. The swap-based router swaps positions of pairs of adjacent nets if they are out-of-order. For example, in the bottom track of Fig.4(a), nets 5 and 4 are out-of-order and they swap columns in the transition to track 1; this is reflected in the swap depicted in Fig.4(b). A bubble-sort based technique can also be used, as depicted in Fig.4(c), allowing indexes to sort across multiple column positions. Bubble-sorting, however, causes nets to cross at non-perpendicular angles, and therefore is *unusable* for optical waveguide routing. *Our channel routing technique utilizes swap-based sorting for routing.*

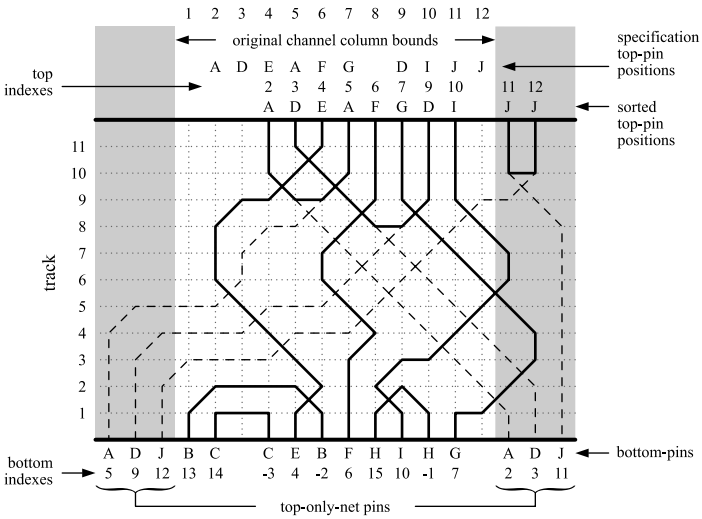


Fig. 5: A swap-router channel solution. Shaded region denotes columns outside the initial channel bounds.

B. Encoding Side-only Nets

The described channel problem setup assumes that all nets appearing on the bottom of a channel also appear on the top. However, most channel problem instances also incorporate nets with pins exclusively on one side of the channel. Also, empty spaces *between* pins (i.e. "gaps"), or within the routing tracks are also not accounted for by the basic routing algorithm. We first formally define net types with respect to their pin locations, as well as the concept of empty spaces in the routing space:

Definition 3.1. [X-net, T-net, B-net, side-only nets] *A net with pins on both the top and bottom of a channel is an **X-net** (cross-over/shared net). Nets with pins exclusively on one side of a channel are denoted **side-only nets**: a **T-net** (top-only net) has pins on only the top side of the channel; a **B-net** (bottom-only net) is net with pins on only the bottom of the channel.*

Definition 3.2. [Gap] *A **gap** is an empty location in the routing grid, or an empty pin location. Gaps are not assigned sorting indexes, but may be routed over if unoccupied.*

Consider the channel depicted in Fig.5. We must assign net pins initial sorting indexes in order to route the channel nets. Observe nets *E*, *B*, and *D*. Net *E* is an X-net, and therefore the bottom pin of *E* is assigned the index corresponding to its top pin's index-position, in this case 4. Nets *B* and *D* are B-net and T-net types respectively; they have their pins exclusively on one side of the channel—*side-only nets*. The work of [14] provides a means for encoding T-nets and B-nets into the channel sorting problem:

B-net encoding: For a given B-net, the left-side pin is assigned a high-valued index, and the right-side pin is assigned a negative index. For example, in Fig.5 net *B* is a B-net. The left-pin of *B* is assigned index 13—*a value greater than the number of channel columns*—and the right-pin is assigned -2. The result of these index assignments is that the high-valued index causes the route of the left-pin to sort to the right; the negative index causes the route of the right-pin to sort to the left. When the routes meet, the net is considered routed and the indexes are removed from the problem in subsequent tracks.

T-net encoding: T-nets do not exist at all on the bottom track, and therefore are added to the initial bottom sorting track as *additional columns ("virtual columns")* to the left and right of the original channel columns. In Fig.5, these are indicated by the shaded areas. The bottom pins of the T-nets are assigned indexes corresponding to the positions of the top-pins of the net; this causes the respective routes to sort towards these positions during the sorting. The routes of the T-nets must, however, meet at some point within the channel. To facilitate this, T-net pins are assigned to the sides *opposite* of their relative positions in the top track. This causes the routes to cross each other at some point on their way to their final positions.

In Fig.5, net *D*—a T-net—has pins on top of the channel at columns 3 and 9. These column positions are used as the indexes assigned to the pins at the bottom of the channel—index 9 on the left, 3 on the right. As the sorting proceeds, the routes for the T-nets converge towards each other, cross at some point within the channel, and continue to their final sorted position. After the channel routing completes, only the routing above the crossing point of a T-net is retained as the actual T-net route (e.g. the solid-line route of *D* in Fig.5).

C. Limitations of Swap Router

Consider the swap-router solution depicted in Fig.5. Immediately apparent are the following problems:

1) wasted tracks (gaps) with sub-optimal routes;

2) the final (top) positions of routed net terminals are shifted to the right, as compared to the original specification;
3) routes detour away from destination column position, creating more "bends", which can cause optical signal loss;
4) routes exist outside the channel's column bounds.

These problems arise from the following sources:

Gaps in the sorting problem: Empty space (i.e. "gaps") in the channel problem affect the relative positioning of nets in a track, while not providing any sorting information themselves. The end result is that the routing solution only respects the *relative* positioning of nets, not the *absolute* position. This is problematic, as the routed nets of the top track may not reflect the positions of the pins in the original specification; this is demonstrated in the top track of Fig.5. Both B-nets and T-nets introduce gaps into the sorting problem.

T-net encoding: As demonstrated in Fig.5, the encoding for T-nets *negatively impacts virtually all aspects of the channel solution.* T-nets introduce additional channel columns that affect positioning of the net pins, as well as enabling routes to extend outside the channel's original column bounds. The T-net temporary/virtual routes (dashed lines in Fig.5) also cause unnecessary detours for any other route they interact with, increasing track count—a track for each crossing—during the sorting. For example, in Fig.5 the T-net route originating from the left side for net *J* must cross seven (7) other routes, including the right terminal route of *J*, to form the solution route; none of the temporary routing below the cross-over point serves any real purpose in the final channel solution. The routes also cause other routes to move outwards as routes cross, such as the route for *E* detouring left as it crosses the left terminal route for *A*, *D*, and *J*.

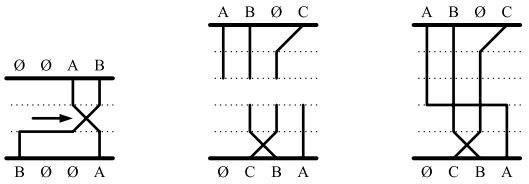
Post-processing: Accepting the solution from the swap router as-is would require an additional post-processing step to: 1) position T-net terminals correctly with respect to the original specification, and 2) post-process routes within the solution to improve track usage and prevent routes from extending outside the bounds of the channel. While some post-processing work is performed in [15] to compact the track space, gap handling remains unaddressed. Larger problems, especially with many T-nets, can produce extremely large solutions, most of which is wasted space. This effectively defeats the purpose of utilizing the swap router in the first place.

4. 2-SIDED SWAP ROUTING (2SWAP)

To overcome the limitations of the original sorting routing, we introduce the 2-sided Swap Router (2Swap): a sorting-based router that performs routing from both sides of the channel simultaneously. Sorting still remains a key component of routing, ensuring crossing-minimality; however, the routing from both sides overcomes two key limitations of the original router:

- The elimination of T-nets from the routing solution. No additional columns are added to the channel problem and no temporary routes are needed.
- Sorting in the presence of gaps is addressed, and pins on each side of the channel are fixed as per the original specification.

Crossing minimality: As in the original swap router, the 2Swap router produces a crossing-minimal solution. This is ensured by updating the position of route pins at each iteration of the routing. A swap that takes place on one side of the channel is reflected when the other side is routed, retaining the sorting-based assurance of crossing-minimality.



(a) Horizontal gap spanning (b) Net spanning: *A* cannot cross *B* or *C* diagonally (c) Horizontal connection across nets

Fig. 6: Horizontal connections to complete routing

Gap crossing: Gaps are an intrinsic part of virtually every channel routing instance. The presence of gaps, however, is not even mentioned by [14], and other examples in literature [15] only depict and describe dense (gapless) routing problems such as Fig.4. The gap-crossing extension described below was actually applied to the channel in Fig.5 to provide a routed solution.

We address the existence of gaps within the sorting problem by allowing routes to span horizontally across gaps to the point of crossing. During sorting, pairs of routes are analyzed across gaps. Should a swap be possible, a horizontal span is created up to the point of crossing, and the swap then occurs. This is depicted in Fig.6(a), where a route for net *B* traverses multiple gaps (∅) to cross over route *A* on the right.

Two-sided swap-routing: The 2Swap router alternates between both sides of the channel for swap-routing, while performing bookkeeping to ensure that crossings are not performed twice and sorting-indexes are updated. We now define two different net-concepts:

Definition 4.1. [Source/destination sides, S-net, D-net] *Given a channel side being routed, routing is performed from the **source** (src) to **destination** (dest) sides of the channel. The src/dest sides of the channel are analogous to the bottom/top sides when routing is performed from the bottom-to-top. Given a channel side being routed, an **S-net** is an **source-side net**—a net with both pins exclusively on the source side of the channel. Likewise, a **D-net** is an **destination-side net**—a net with both pins exclusively on the destination side of the channel.*

The technique works in the following manner:

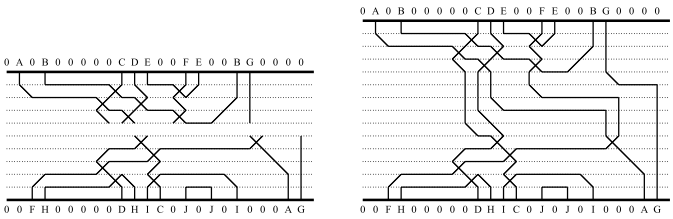
1) For each iteration of the 2Swap router, swap-routing is performed for a single track on both sides of the channel, subject to the following conditions:
   - For each iteration of the sorting, relative ordering of nets is recomputed for swap-routing, utilizing the

current set of sorted tracks. This helps ensure that the routing performed for a given side is aware of net-swaps that took place on the opposite side of the channel.
- For a given side of a channel being routed, only S-nets and X-nets are encoded as indexes and routed. D-nets are treated as gaps on the destination track for purposes of routing. This ensures that D-nets are only routed on their respective side as S-nets, avoiding the problems previously associated with T-nets in the original sorting-based router.
- Any S-nets that have completed routing are removed in subsequent tracks (replaced by gaps).

2) Routing completes when routes on both the top and bottom are completely sorted with respect to each other.
3) A post-sort routing is performed to provide a routable channel.

The 2Swap technique is given in Algo.1.

Post-sort routing: The sorting phase of routing completes when all S-nets are routed, and all X-net-indexes are sorted with respect to each other. The latter condition, however, does not guarantee that the channel is fully routed. Consider the 2Swap routing solution depicted in Fig.7(a). While the X-nets in the middle of the channel are *sorted* with respect to each other, they are not fully *routed* as their column positions are misaligned. In such cases, a post-sort routing must be applied, as demonstrated in Fig.7(b).



(a) Sorted, but not fully-routed solution

(b) After post-sort routing

Fig. 7: Post-sort routing for 2Swap solution.

Algorithm 1 2-sided Swap Routing

```
T_sol := [ ]            (solution tracks for channel)
repeat
    for each side S of the channel do
        T_src := copy of current track of src side
        T_dest := copy of current track of dest side
        Assign all D-net nets in T_dest to ∅
        I_src := net-indexes of T_src with respect to T_dest
        T_sorted := SwapSort (I_src)
        Add T_sorted to T_sol
        Remove completed S-nets from T_sorted
        Set current track for S to T_sorted
    end for
until (all B-nets and T-nets are routed) and (all X-nets are sorted with
respect to each other)
T_sol := PostSortRouting(T_sol);
```

A. Solution Quality

We route the channel depicted in Fig.5 using the new 2Swap router and observe that it produces a far more usable solution (Fig.3(b)). Fewer tracks are utilized (6 vs 11) and no additional columns are added to the routing solution. In addition, the top and bottom pins are the same as the specification. Overall, the solution produced by the 2Swap router is improved with respect to the original swap-router. However, further improvements are still possible, especially in terms of waveguide bends.

1) Excess bends and their cause: The 2Swap route produces a large number of excess bends in its solutions. This is especially apparent in larger channel instances such as Fig.8(a). The excess bends in 2Swap routes are due to the fact that while the position of net *terminals* are fixed to *absolute* positions, the intermediate sorting is still a *relative* sorting-position operation. Pairs of routes for a given net are therefore not actively converging towards each other during the sorting. For example, in Fig.3(b), net E swaps with net B, shifting E to the right—in the opposite direction of its destination pin. As a result, route E must detour across a large expanse of space to connect both sides—despite being only one column away in the original problem. The same problem afflicts net I, which detours left in its swap with H.

As a signal loss mechanism, bends must also be accounted for, and we address these with a *constrained* 2-sided Swap Router.

5. Constrained 2-Sided Swap Routing

The 2SwapC router introduces two key extensions over 2Swap:

- Routing is subject to a *convergence constraint*: pairs of net-routes are constrained such that they only converge towards each other.
- Routes may cross using horizontal/vertical crossings, in addition to diagonal crossings.

Convergence constraint: Given a pair of routes for the same net, a new constraint is placed on the basic sorting technique: *the routes cannot be sorted (routed) away from each other.* We denote this as the **convergence constraint.**

The convergence constraint *overrides* the swap-sorting of the basic routing technique; this has effect that it can prevent routing from completing. For example, consider the channel problem instance in Fig.6(b-c). In the first iteration, routes move in the direction of their paired-routes, e.g. B and C swap on the bottom, and the top-route of C moves left. In the second iteration, Fig.6(b), the bottom routes of A and C cannot swap; this would violate the convergence constraint by forcing C to the right. The bottom route for A can only move vertically. At this point routing stops, leaving the solution in an incomplete state. We therefore introduce our second extension: horizontal spanning across vertical routes.

Horizontal spans, and crossings: The bottom route of A is *blocked* by C, due to the convergence constraint. Such blocked routes must be allowed to cross each other without swapping. This is achieved by allowing routes to cross using horizontal/vertical crossings (HV-crossings) under certain conditions.

The formation of vertical spans across tracks is the key factor in enabling horizontal spans to cross over, as depicted in Fig.6(c). Horizontal spans are allowed to cross only those

vertical spans that traverse two or more tracks. This ensures that bends will not occur at the junction of an HV-crossing. Furthermore, to reduce the number of bends, routes may span horizontally only if they can make a direct connection to the column of their paired routes. The effect of this is that any given net will only make one horizontal span within the channel. This can be observed in the channel solution depicted in Fig.8(b).

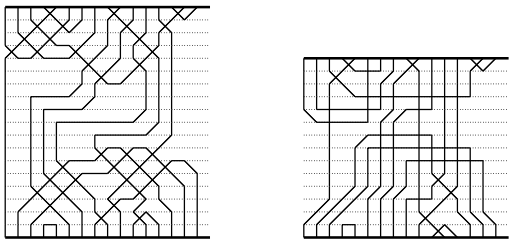
Comparison with 2Swap: While the number of crossings is the same, channel solutions produced by 2SwapC generally have fewer bends than 2Swap. This is evident in Fig.8 where the 2Swap router produces many routes that "zig-zag" throughout the routing region, whereas 2SwapC utilizes straight routes that converge towards each other.

6. Experimental Results

TABLE I 2Swap = 2-sided swap; 2SwapC = Constrained 2-sided swap router; N = # nets; B/T = # bottom/top-only nets; W = channel width; X = Crossings; Loss = bend-loss, Tr = tracks

						2Swap		2SwapC	
Design	**N**	**W**	**B**	**T**	**X**	**Loss**	**Tr**	**Loss**	**Tr**
rnd.16.0	16	16	5	5	35	20.9	17	13.2	13
rnd.16.1	16	24	5	5	32	22.7	17	13.0	12
rnd.16.2	16	32	5	5	24	25.3	18	12.9	14
rnd.16.3	16	40	5	5	21	24.3	11	11.9	11
rnd.16.4	16	48	5	5	33	41.1	17	12.9	24
rnd.16.5	16	56	5	5	32	42.2	11	16.0	21
rnd.16.6	16	64	5	5	20	31.5	16	11.5	26
rnd.16.7	16	72	5	5	33	44.5	20	14.0	22
rnd.32.0	32	32	10	10	140	76.4	37	37.8	32
rnd.32.1	32	48	10	10	182	139.9	33	39.5	50
rnd.32.2	32	64	10	10	182	143.5	34	36.8	43
rnd.32.3	32	80	10	10	134	116.1	24	30.4	51
rnd.32.4	32	96	10	10	135	127.9	28	38.2	80
rnd.32.5	32	112	10	10	114	127.2	25	32.4	42
rnd.32.6	32	128	10	10	152	149.5	32	49.0	111
rnd.32.7	32	144	10	10	145	140.9	30	35.9	83
rnd.48.0	48	48	16	16	281	104.3	40	49.1	48
rnd.48.1	48	72	16	16	322	177.6	42	53.8	67
rnd.48.2	48	96	16	16	307	226.3	41	56.9	73
rnd.48.3	48	120	16	16	262	217.9	36	56.7	79
rnd.48.4	48	144	16	16	285	256.6	41	60.0	77
rnd.48.5	48	168	16	16	339	317.6	47	69.4	125
rnd.48.6	48	192	16	16	347	318.3	47	67.0	101
rnd.48.7	48	216	16	16	352	330.9	47	82.7	219
rnd.64.0	64	64	21	21	525	190.7	41	80.2	63
rnd.64.1	64	96	21	21	591	335.1	53	91.1	81
rnd.64.2	64	128	21	21	578	420.3	51	94.8	77
rnd.64.3	64	160	21	21	573	444.7	54	121.4	136
rnd.64.4	64	192	21	21	505	464.9	54	90.5	115
rnd.64.5	64	224	21	21	645	582.8	57	105.6	142
rnd.64.6	64	256	21	21	552	477.2	54	80.3	127
rnd.64.7	64	288	21	21	651	587.6	62	99.3	156
opt.0	17	22	3	8	37	26.9	16	12.9	10
opt.1	15	18	3	3	47	26.5	21	13.6	16
opt.2	19	34	2	3	69	4049	28	22.5	38
opt.3	31	36	10	15	88	61.1	21	27.8	31
opt.4	34	40	6	8	227	117.8	39	43.9	49
opt.5	31	49	6	7	152	92.0	24	33.2	38
opt.6	35	84	3	4	235	158.1	47	55.9	61

We compare the 2Swap and 2SwapC routers in terms of bend-loss and track utilization. The original sort-based router is not compared, as it produces routing solutions that violate the original channel specifications. Problem instances vary in terms of nets to be routed, the width of the channel, and the number of side-only nets. The number of crossings is also shown, and is the same for both routers.



(a) 2Swap routed channel (b) 2SwapC routed channel

Fig. 8: 2Swap vs 2SwapC solutions for a channel problem.

Both routing techniques are implemented as compiled script-code. Problem instances incorporating as many as 512 nets were tested, and routing completed in under 30 seconds. Most routings complete in under a second.

Channel problem instances: Channel problem instances are provided as a mixture of both randomly generated channels, and channels derived from digital optical logic designs from [6]. For the optical designs, devices are placed into rows using [38], forming channels in between. Randomly generated instances were varied in their channel parameters in order to observe trends in the routing solutions.

Bend-loss: In our octilinear grid, bends can be either 90° or 135°, having different loss characteristics. Bend-loss is computed for each using Eqn.1 [39] as a function of radius of curvature:

$$\alpha_{bend}(r) = C_1 \cdot \exp(-C_2 \cdot r) \tag{1}$$

The constants C_1 and C_2 are dependent on the physical parameters of the waveguides. For simplicity and convenience, we use a unit grid for calculating bend-radius, and select $C_1 = C_2 = 1$. This results in bends-loss for $\alpha_{bend}(\{135°, 90°\}) = \{0.135, 0.368\}$.

A. Analysis of results

The two routers are tested on a variety of channel problem instances. The results of the routings are found in Tbl.I, with random results on top, and design-derived results on bottom. Total time for both routers to complete all channel problem instances was less than 10 seconds. This was unsurprising given the low computational complexity of underlying sorting-based technique.

Analysis of the results reveals that, in terms of bend-loss, 2SwapC produces far less bend-loss as compared to 2Swap, often many times less. Both routers see increases in bend-loss as a function of both net count and channel width. Conversely, 2Swap often utilizes fewer tracks than 2SwapC, with the latter increasing as a function of channel width as well. We attribute the 2Swap router's ability to utilize fewer tracks in the presence of wider channels, as compared to 2SwapC, to its frequent use of horizontal spanning. In general, the design-derived channels ("opt"-series) reflect the same trends as the random ("rnd") series of designs.

Overall, there is a trade-off in terms of tracks versus bend-loss between the routers. However, with the primary objective

UU IR Author Manuscript

of this routing being signal loss, the 2SwapC router ultimately produces better results. Even with equal weighting, the amount of bend-loss produced by 2Swap grows at a far faster rate than the number of tracks utilized by 2SwapC, especially when considering densely packed tracks. Bend-loss could possibly be traded for area if needed.

7. CONCLUSION

This paper presents sorting-based channel routers for integrated optics. Our investigations show that automated channel routing techniques can be effectively applied to optical waveguide routing, optimizing for signal-loss as a primary objective. Optical constraints, specifically involving waveguide crossing constraints and bend-loss, are incorporated into our channel routing models. Crossings are minimized through the use of sorting-based routing techniques, and our non-Manhattan-based routing grid provide routing flexibility well-suited for integrated optics.

Our routers address and overcome the shortcomings of previous sort-based routers through the use of two-sided swap routing (2Swap), providing usable routing solutions, while retaining crossing minimality in solutions. We have further extended 2Swap to incorporate positional net information in 2SwapC, as a means of reducing bend-losses. Bend-losses are markedly improved, as demonstrated in tests on many channel routing instances, and 2SwapC demonstrates that it is superior in terms of signal loss. Overall our channel routers provide an effective means for automated optical waveguide routing with signal loss as a primary metric.

REFERENCES

[1] M. Peach, "Silicon photonics forges ahead", *Optics.org*, Feb 2012.
[2] "IBM Research: Silicon Integrated Nanophotonics", www.research.ibm.com/photonics.
[3] "OpSIS: Optoelectronic System Integration in Silicon", http://depts.washington.edu/uwopsis.
[4] J. Chan, G. Hendry, K. Bergman, and L. Carloni, "Physical layer modeling and system-level design of chip-scale photonic interconnection networks", *IEEE Trans. on Computer-Aided Design of Electronic Systems*, 2011, In Press.
[5] C. Madsen and J. Zhao, *Optical Filter Design and Analysis: A Signal Processing Approach*, Wiley, 1999.
[6] C. Condrat, P. Kalla, and S. Blair, "Logic Synthesis for Integrated Optics", *Proc. ACM Great Lakes Symp. on VLSI*, pp. 13 – 18, 2011.
[7] H. J. Caulfield, C. S. Vikram, and A. Zavalin, "Optical logic redux", *Optik*, vol. 117, pp. 199–209, 2006.
[8] A. Politi, J. Matthews, and J. O'Brien, "Shor's Quantum Factoring Algorithm on a Photonic Chip", *Science*, vol. 325, pp. 1221, 2009.
[9] A. J. Poustie and K. J. Blow, "Demonstration of an all-optical Fredkin gate", *Optics Communications*, vol. 174, pp. 317 – 320, 2000.
[10] S. Kotiyal, H. Thapliyal, and N. Ranganathan, "Mach-Zehnder Interferometer based Design of all Optical Reversible Binary Adder", *in Proc. Design Automation & Test in Europe (DATE) Conf.*, 2012.
[11] D. Ding and D. Pan, "Oil: A nano-photonics optical interconnect library for a new photonic network architecture", *in System-Level Interconnect Prediction Workshop (SLIP)*, 2009.
[12] D. Ding, Y. Zhang, H. Huang, R. T. Chen, and D. Z. Pan, "O-Router: an optical routing framework for low power on-chip silicon nano-photonic integration", *in Design Auto. Conf.*, pp. 264–269, 2009.
[13] J. Orcutt and R. Ram, "Photonic Device Layout within the Foundry CMOS Design Environment", *IEEE Photonics Tech. Lett.*, 2010.
[14] K. Chaudhary and P. Robinson, "Channel Routing by Sorting", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 10, pp. 754 –760, jun 1991.
[15] J. Yan, "An improved optimal algorithm for bubble-sorting-based non-Manhattan channel routing", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 18, pp. 163 –171, feb 1999.
[16] H. J. Caulfield et al., "Generalized optical logic elements GOLEs", *Optics Communications*, vol. 271, pp. 365–376, mar 2007.
[17] W. Bogaerts, P. Dumon, D. V. Thourhout, and R. Baets, "Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides", *Opt. Lett.*, vol. 32, pp. 2801–2803, Oct 2007.
[18] F. Xu and A. W. Poon, "Silicon cross-connect filters using microring resonator coupled multimode-interference-based waveguide crossings", *Opt. Express*, vol. 16, pp. 8649–8657, Jun 2008.
[19] J. Cardenas, C. B. Poitras, J. T. Robinson, K. Preston, L. Chen, and M. Lipson, "Low loss etchless silicon photonic waveguides", *Opt. Express*, vol. 17, pp. 4752–4757, Mar 2009.
[20] Y. Vlasov and S. McNab, "Losses in single-mode silicon-on-insulator strip waveguides and bends", *Opt. Express*, vol. 12, pp. 1622–1631, Apr 2004.
[21] Y. Qian, S. Kim, J. Song, G. P. Nordin, and J. Jiang, "Compact and low loss silicon-on-insulator rib waveguide 90° bend", *Opt. Express*, vol. 14, pp. 6020–6028, Jun 2006.
[22] G. Li, J. Yao, H. Thacker, A. Mekis, X. Zheng, I. Shubin, Y. Luo, J. Lee, K. Raj, J. E. Cunningham, and A. V. Krishnamoorthy, "Ultralow-loss, high-density SOI optical waveguide routing for macrochip interconnects", *Opt. Express*, vol. 20, pp. 12035–12039, May 2012.
[23] M. Petracca, B. G. Lee, K. Bergman, and L. Carloni, "Design exploration of optical interconnection networks for chip multiprocessors", *in IEEE Symp. High Perf. Interconnects*.
[24] M. Cianchetti, J. Kerekes, and D. Albonesi, "Phastlane: A rapid transit optical routing network", *in Proceedings of the 36th annual International Symposium on Computer Architecture*, ISCA '09, pp. 441–450, New York, NY, USA, 2009. ACM.
[25] R. Beausoleil et al., "A Nanophotonic Interconnect for High-Performance Many-Core Computation", *Symposium on High-Performance Interconnects*, pp. 182–189, 2008.
[26] J. Hardy and J. Shamir, "Optics Inspired Logic Architecture", *Optics Express*, vol. 15, pp. 150–165, 2007.
[27] C. Condrat, P. Kalla, and S. Blair, "Exploring Design and Synthesis for Optical Digital Logic", *International Workshop on Logic Synthesis*, 2010.
[28] RSoft Inc., *RSoft Photonics CAD Suite*, www.rsoftdesign.com.
[29] D. Ding, B. Yu, and D. Pan, "GLOW: A global router for low-power thermal-reliable interconnect synthesis using photonic wavelength multiplexing", *in Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pp. 621–626, 30 2012-Feb. 2.
[30] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures", *in Proceedings of the 8th Design Automation Workshop*, DAC '71, pp. 155–169, New York, NY, USA, 1971. ACM.
[31] D. Deutsch, "A Dogleg channel router", *in Proceedings of the 13th Design Automation Conference*, DAC '76, pp. 425–433, New York, NY, USA, 1976. ACM.
[32] T. Yoshimura and E.S. Kuh, "Efficient Algorithms for Channel Routing", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 1, pp. 25 – 35, january 1982.
[33] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI Physical Design: from Graph Partitioning to Timing Closure*, Springer , 2010.
[34] S. Sait and H. Youssef, *VLSI Physical Design Automation*, IEEE Press, 1995.
[35] M. Marek-Sadowska and M. Sarrafzadeh, "The crossing distribution problem [IC layout]", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, pp. 423–433, Apr.
[36] S. Xiaoyu and W. Yuke, "On the crossing distribution problem", *ACM Trans. Des. Autom. Electron. Syst.*, vol. 4, pp. 39–51, jan 1999.
[37] S. B. Stephen and L. S. Kyu, "QCA channel routing with wire crossing minimization", *in Proceedings of the 15th ACM Great Lakes symposium on VLSI*, GLSVLSI '05, pp. 217–220, New York, NY, USA, 2005. ACM.
[38] J. Roy, D. Papa, S. Adya, H. Chan, A. Ng, J. Lu, and I. Markov, "Capo: robust and scalable open-source min-cut floorplacer", *in Proceedings of the 2005 international symposium on Physical design*, ISPD '05, pp. 224–226, New York, NY, USA, 2005. ACM.
[39] E. A. J. Marcatili and S. E. Miller, "Improved relationships describing directional control in electromagnetic wave guidance", *Bell Syst. Tech J.*, vol. 48, pp. 2161–2188, 1969.