

# ON SEGMENTED CHANNEL ROUTABILITY

William N. N. Hung<sup>1</sup>, Xiaoyu Song<sup>2</sup>, Alan Coppola<sup>3</sup> and Andrew Kennings<sup>4</sup>

<sup>1</sup>Intel Architecture Group, Intel Corporation, Hillsboro, Oregon 97124, USA

<sup>2</sup>Department of ECE, Portland State University, Portland, Oregon 97201, USA

<sup>3</sup>Cypress Semiconductor, 9125 S.W. Gemini Drive, Beaverton, OR 97008, USA

<sup>4</sup>Department of ECE, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

## ABSTRACT

We address the problem of checking the routability of segmented channels using satisfiability. The segmented channel routing problem arises in the context of row-based field programmable gate array (FPGAs). Our approach transforms the routing task into a single large Boolean equation such that any assignment of input variables that satisfies the equation specifies a valid routing. It considers all nets simultaneously and the absence of a satisfying assignment implies that the channel is unroutable. Empirical results show that the method is time-efficient and applicable to large problem instances.

## 1. INTRODUCTION

A wide variety of tactics have been tried to solve the FPGA routing problem. Wong *et al* [4] introduced a global router that uses an off-line, integer linear programming technique to enumerate all possible uses of a switch-block architecture. A global router uses this information as an exact congestion estimator for each small and atomic block of the routing fabric. However, the global router must still handle the complexity of managing the competition for routes between the atomic blocks in the fabric. Wood *et al* [5] considered a routability estimator of a global router in terms of satisfiability. Greene [3] described search-based strategies for segmented channel routing and proved that it is NP-complete. Yang *et al* [6] proposed the algorithms based on graph clique for some restricted segmented channel routability check.

In this work, we give a novel formulation for both routing and routability that provides exact routability guarantees. In [1], Devadas showed a simple but elegant formulation of conventional two-layer channel routing as Boolean satisfiability. We extend his method to the segmented channel routing. Although some work for FPGA routing via satisfiability has been done [2,5,9], but they are designed for the detailed routing, none of them consider the routability inside a segmented channel [3]. We formulate the case analysis for segmented channel routing. This formulation encodes the information present in a channel vertical constraint graph, horizontal constraint graph, and anticipated channel width into Boolean constraints. The resulting routability constraint is satisfiable if the conjunction of all the horizontal and vertical constraints are satisfiable. The resulting Boolean equation fully specifies the set of feasible net-to-track mappings via its satisfying variable assignments.

The idea is to specify the routing problem as a set of interacting assignments of nets to available resources expressed exactly as a large set of Boolean equations. Any assignment of values to input variables that satisfies this Boolean routability equation gives a complete routing. Although satisfiability is itself an NP-hard problem, there has been a promising body of practical solvers available on large satisfiability problems [8]. Our Boolean formulation can be input to any satisfiability solver. It is an incremental method which allows not only for rapid construction of necessary routability constraints, but also it can incorporate incrementally updates for new additional performance driven constraints. Unlike other detailed routers which construct the routing net-by-net, our approach finds a routing for all net simultaneously.

## 2. PRELIMINARIES AND DEFINITIONS

The input to a segmented channel routing [3] problem is a segmented channel consisting of a set  $\Gamma$  of  $T$  tracks and a set  $C$  of  $M$  nets. Each track  $t$  ( $0 \leq t \leq T-1$ ) extends from column 1 to column  $N$ , where  $N \geq M$ , and is divided into a set  $G_t$  of  $g_t$  adjacent segments separated by horizontal switches, i.e.,  $|G_t| = g_t$ . The model of a segmented channel routing is depicted in Figure 1, where a symbol cycle represents a horizontal switch and  $\oplus$  represents a cross switch. For convenience sake, we label the tracks 1 to  $T$ .

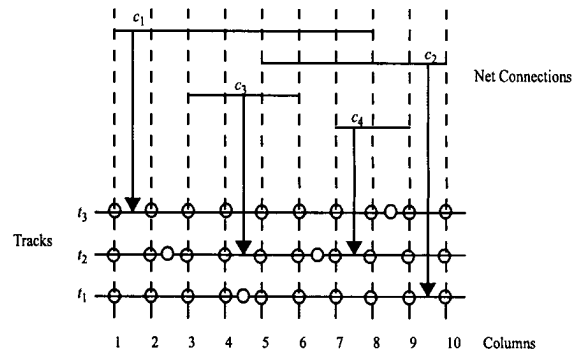


Figure 1. The model of a segmented channel routing.

Each net  $c$  is a connection between two or more columns. The span of the net  $c$  is defined by its leftmost and rightmost columns,  $left(c)$  and  $right(c)$ , respectively. There are two or more terminals for each net  $c$ . Each terminal occupies a column. Denote the set of terminal columns of net  $c$  by  $L(c)$ . The leftmost and rightmost columns are both terminals of net  $c$ :  $left(c), right(c) \in L(c)$ . Each

terminal of  $c$  must be located within the span of net  $c$ , i.e.,  $\text{left}(c) \leq i \leq \text{right}(c)$ , for all  $i \in L(c)$ .

A performance driven track, is a track where, due to critical timing constraints, the wire is shorter than average tracks. We use  $\Psi$  to represent the set of nets on the critical path. We use  $\wp$  to represent the set of performance driven tracks.

Let  $s_{t,i}$  denote segment  $i$  on track  $t$ , where the index  $i$  is numbered from left to right along the track. Let  $\text{left}(s_{t,i})$  and  $\text{right}(s_{t,i})$  be the leftmost and rightmost columns in which this segment is present respectively, we have  $1 \leq \text{left}(s_{t,i}) \leq \text{right}(s_{t,i}) \leq N$ . Since the switches between adjacent segments are placed between consecutive columns, we have  $\text{left}(s_{t,i+1}) = \text{right}(s_{t,i}) + 1$  for all  $t = 1, \dots, T$  and  $i = 1, \dots, g_t - 1$ .

### 2.1. Dogleg-free Segment Routing

In dogleg-free segment routing, a net connection can be assigned to at most one track. Each net  $c_i$ ,  $1 \leq i \leq M$ , is uniquely characterized by the span of its leftmost and rightmost columns,  $\text{left}(c_i)$  and  $\text{right}(c_i)$ . When a net  $c$  is assigned to a track  $t$ , the segments in track  $t$  that are present in the columns spanned by  $c$  are considered *occupied*. More precisely, a segment  $s$  in track  $t$  is occupied by a net  $c$  assigned to that track iff  $Q(s,c)$  is true:

$$Q(s, c) = (\text{right}(s) > \text{left}(c)) \wedge (\text{left}(s) < \text{right}(c))$$

A routing,  $R$ , of a set of nets is an assignment of each net to a track such that no segment is occupied by more than one net. A *K-segment channel routing* is a routing that satisfies the additional requirement that each net occupies at most  $K$  segments on a track.

### 2.2. Doglegged Segment Routing

In a doglegged segment routing, a net can be assigned to more than one track. This is also called *generalized routing or doglegging* [3]. A net  $c$  can be split into  $p$  ( $p \geq 1$ ) parts:  $(\text{left}(c), l_1)$ ,  $(l_1+1, l_2)$ ,  $(l_2+1, l_3), \dots, (l_{p-1}+1, \text{right}(c))$ , such that each part can be assigned to different tracks. A column  $l_i$ , where a net is split, is referred to as a column where net  $c$  changes tracks. Roychowhury proposed three important special cases of the generalized routing problem:

1. Determine a generalized routing that uses at most  $K$  segments.
2. Determine a generalized routing that uses at most  $l$  different tracks for routing any net.
3. Determine a generalized routing where connections can switch tracks only at predetermined columns.

## 3. DOGLEG-FREE ROUTABILITY

Given a segmented channel, we want to assign each net to at most one track in order to connect the top and bottom terminals as specified. We define a variable vector  $\vec{v}(c)$  for each net  $c$  as the binary (encoding) representation of the track index where the net  $c$  is assigned to, i.e.,  $\vec{v}(c) = v_1(c)v_2(c)\dots v_w(c)$  where  $w = \lceil \log_2(T) \rceil$  and  $v_i(c) \in \{0, 1\}$  for  $i = 1, 2, \dots, w$ .

Since each net has to be assigned to one track,  $\vec{v}(c)$  is equal to a track number between 0 and  $(T-1)$ . Thus, the constraint for the domain of  $\vec{v}(c)$  is:

$$0 \leq \vec{v}(c) \leq T-1 \text{ for all } c \in C. \quad (1)$$

Since all the definitions of dogleg-free segment routing involve only the leftmost and rightmost columns of each net, the problem formulation applies to two-terminal as well as multi-terminal routing. In this section, our solution works for multi-terminal routing as well as two-terminal routing.

### 3.1. Unlimited Segment Routing

Each segment can be occupied by at most one net. We define an occupancy function  $h$  for net  $c$  and segment  $s$  on track  $t$ :

$$h(c, s, t) = Q(s, c) \wedge (\vec{v}(c) = t-1)$$

The function  $h$  is true if and only if segment  $s$  on track  $t$  is occupied by net  $c$ . The segment occupancy constraint for each net can be defined in terms of the occupancy function  $h$ :

$$\bigcap_{t=1}^T \bigcap_{s \in G_t} \left( \bigcap_{\substack{m \neq n \\ m, n \in C}} \neg(h(m, s, t) \wedge h(n, s, t)) \right) \quad (2)$$

where  $\cap$  denotes the boolean conjunctive operator.

The necessary and sufficient condition for dogleg-free unlimited segment routing is the conjunction of formulae (1) and (2).

### 3.2. K-segment Routing

We define  $m\text{left}(c, t)$  and  $m\text{right}(c, t)$  to be the *identification numbers* (IDs) of the left-most segment and the right-most segment occupied by net  $c$ , respectively, if that net is assigned to track  $t$ .

$$m\text{left}(c, t) = \min\{j | Q(s_{t,j}, c) \wedge (s_{t,j} \in G_t)\}$$

$$m\text{right}(c, t) = \max\{j | Q(s_{t,j}, c) \wedge (s_{t,j} \in G_t)\}$$

Given  $\text{left}(s)$  and  $\text{right}(s)$ ,  $\text{left}(c)$  and  $\text{right}(c)$  are deducible from the problem specification, the values of  $Q(s,c)$  are deducible as well. Hence the values of  $m\text{left}(c,t)$  and  $m\text{right}(c,t)$  can be computed before the satisfiability check.

To limit the maximum number of segments that the nets of a segmented channel can occupy, we need to ensure that the difference between indices for the leftmost segment and the rightmost segment on the track to which this net is assigned is less than  $K$ . Otherwise, the net cannot be assigned to this track. The  $K$ -segment constraint is:

$$\forall t \in \Gamma \cdot \forall c \in C \cdot (\vec{v}(c) = t) \Rightarrow (K > m\text{right}(c, t) - m\text{left}(c, t)) \quad (3)$$

The overall routability check for dogleg-free  $K$ -segment routing is the conjunction of formulae (1), (2), and (3).

### 3.3. Performance Driven Tracks

When connecting nets on the critical path, we may want to force these nets to be assigned to certain *performance driven tracks*. This requirement introduces an additional constraint:

$$\forall c \in \Psi \cdot \exists t \in \wp \cdot \vec{v}(c) = t \quad (4)$$

The overall routability check for performance driven track routing is the conjunction of formulae (1), (2), (3) and (4).

## 4. DOGLEGGED ROUTABILITY

Given a segmented channel, we aim at finding a generalized routing where doglegs are allowed [3]. Doglegs can help to reduce the wiring area to reach a more compact layout. The problem is more complicated than dogleg-free routing, since we cannot use one vector  $\overrightarrow{v(c)}$  to encode (denote) all the tracks to which the net  $c$  is assigned.

### 4.1. Track Variables

In a *generalized routing*, a net  $c$  can be split into  $p$  ( $p \geq 1$ ) parts:  $(\text{left}(c), l_1), (l_1+1, l_2), (l_2+1, l_3), \dots, (l_{p-1}+1, \text{right}(c))$ , such that each part can be assigned to different tracks. These distinct  $p$  parts exist because the net  $c$  can change tracks in columns between  $\text{left}(c)$  and  $\text{right}(c)$ . Let  $p_m(c)$  be the maximum number of parts for net  $c$ , we have:

$$1 \leq p \leq p_m(c) \text{ and } p_m(c) = \text{right}(c) - \text{left}(c)$$

For every net  $c$ , we use a set of variable vectors  $\overrightarrow{u(c, 1)}, \overrightarrow{u(c, 2)}, \dots, \overrightarrow{u(c, p_m(c))}$  to denote the tracks where the net is assigned to:

- $\overrightarrow{u(c, 1)}$  is the track of net  $c$  between  $\text{left}(c)$  and  $\text{left}(c)+1$ ;
- $\overrightarrow{u(c, 2)}$  is the track of net  $c$  between  $\text{left}(c)+1$  and  $\text{left}(c)+2$ ;
- $\overrightarrow{u(c, 3)}$  is the track of net  $c$  between  $\text{left}(c)+2$  and  $\text{left}(c)+3$ ;
- $\dots$
- $\overrightarrow{u(c, p_m(c))}$  is the track of net  $c$  between  $\text{right}(c)-1$  and  $\text{right}(c)$ .

Each variable vector  $\overrightarrow{u(c, i)}$  is a binary (encoding) representation of a track used by net  $c$ :

$$\overrightarrow{u(c, i)} = u_1(c, i), \dots, u_q(c, i) \text{ for } i = 1, 2, \dots, p_m(c) \text{ where}$$

$$q = \lceil \log_2 T \rceil \text{ and } u_j(c, i) \in \{0, 1\} \text{ for } j = 1, 2, \dots, q.$$

The constraint for the domain of  $\overrightarrow{u(c, i)}$  is:

$$0 \leq \overrightarrow{u(c, i)} \leq T - 1 \text{ for all } c \in C \text{ and } i = 1, 2, \dots, p_m(c) \quad (5)$$

### 4.2. Horizontal Wiring Constraints

For every part of net  $c$  on track  $\overrightarrow{u(c, i)}$ , we can deduce the left and right columns of that part of the connection:

$$p_{\text{left}}(c, i) = \text{left}(c) + i - 1 \text{ for } i = 1, 2, \dots, p_m(c)$$

$$p_{\text{right}}(c, i) = \text{left}(c) + i \text{ for } i = 1, 2, \dots, p_m(c)$$

We define an occupancy function  $h_d$  for the  $i^{\text{th}}$  part of net  $c$  and segment  $s$  on track  $t$ :

$$h_d(c, i, s, t) = Q_d(s, c, i) \wedge (\overrightarrow{u(c, i)} = t - 1)$$

$$\text{where } i = 1, 2, \dots, p_m(c) \text{ and}$$

$$Q_d(s, c, i) = (\text{right}(s) \geq p_{\text{left}}(c, i)) \wedge (\text{left}(s) \leq p_{\text{right}}(c, i))$$

The function  $h_d$  is true if and only if segment  $s$  on track  $t$  is occupied by the  $i^{\text{th}}$  part of net  $c$ . The segment occupancy constraint for each net can be defined in terms of the occupancy function  $h_d$ :

$$\bigcap_{t=1}^T \bigcap_{s \in G_t} \left( \bigcap_{m, n \in C} \left( \bigcap_{i=1}^{p_m(m)} \bigcap_{j=1}^{p_m(n)} \neg (h_d(m, i, s, t) \wedge h_d(n, j, s, t)) \right) \right) \quad (6)$$

where  $\cap$  denotes the boolean conjunctive operator. Notice the similarity between formula (6) and formula (2). Their difference is their occupancy functions  $h_d$  and  $h$ .

### 4.3. Vertical Wiring Constraints

On the vertical side, each column can be occupied by at most one net. A column is occupied if any net has a terminal on that column or if any net changes tracks on that column. The terminal locations are already given by the problem specification. But the places where each net changes tracks must be resolved so that they do not conflict with the terminals or track changes of other nets.

Each net  $c$  changes tracks on column  $i$  if and only if it spans over that column and its parts on the left side and right side of column  $i$  are assigned to different tracks. We use a boolean value  $x(c, i)$  to denote a change of track for net  $c$  on column  $i$ :

$$x(c, i) = \begin{cases} \text{FALSE} & R(c, i) \\ (\overrightarrow{u(c, i)} \neq \overrightarrow{u(c, i+1)}) & \neg R(c, i) \end{cases}$$

$$\text{where } R(c, i) = (i \leq \text{left}(c)) \vee (i \geq \text{right}(c))$$

If net  $c$  does change tracks on column  $i$ , this column is occupied by the net  $c$ . This column cannot be used by another net (either as a terminal or to change tracks). This requirement is formulated as the following constraint:

$$\bigcap_{i=1}^N \left( \bigcap_{m, n \in C} (x(m, i) \Rightarrow \neg(x(n, i) \vee (i \in L(n)))) \right) \quad (7)$$

The necessary and sufficient condition for doglegged segmented channel routing is the conjunction of formulae (5), (6) and (7).

### 4.4. K-segment Routing

To limit the maximum number of segments used by any net, we need to find out if segment  $s$  on track  $t$  is occupied by net  $c$

$$\text{used}(c, s, t) = \exists i \in \{1, \dots, p_m(c)\} \cdot h_d(c, i, s, t)$$

The total number of segments used by net  $c$  for all segments on all tracks cannot exceed  $K$ .

$$K \geq \sum_{t=1}^T \sum_{s \in G_t} \text{used}(c, s, t), \text{ for all } c \in C \quad (8)$$

Notice that the arithmetic addition operator that is used to implement the summation is applied to boolean formulae here. We compute the addition over boolean symbolic formulae and create a constraint that limits the sum to be less than  $K$ .

The  $K$ -segment generalized routability is the conjunction of formulae (5), (6), (7) and (8).

### 4.5. Track Limitations

To limit the maximum number of distinct tracks used by each net  $c$ , we define a boolean function for the distinct formulation for each part  $i$  of net  $c$ :

$$\text{diff}(c, i) = \forall j \in \{1, \dots, p_m(c)\} \cdot (i \neq j) \Rightarrow (\overrightarrow{u(c, i)} \neq \overrightarrow{u(c, j)})$$

$$\text{where } i = 1, 2, \dots, p_m(c)$$

The constraint on the maximum number of distinct tracks is phrased using the sum of the above boolean bits:

$$l \geq \sum_{i=1}^{p_m(c)} \text{diff}(c, i), \text{ for all } c \in C \quad (9)$$

A generalized routing that uses at most  $l$  different tracks for routing any net is essentially the conjunction of formulae (5), (6), (7) and (9).

#### 4.6. Column Limitations

Given a segmented channel, we want to find a generalized routing where each net  $c$  can only switch tracks at predetermined columns. Let  $\Theta(c)$  be the set of columns where net  $c$  is allowed to switch tracks. The constraint can be formulated as:

$$\bigcap_{c \in C} \left( (p_m(c) > 1) \Rightarrow \bigcap_{i=1}^{p_m(c)-1} \text{nw}(c, i) \right) \text{ where}$$

$$\text{nw}(c, i) = ((\text{left}(c) + i) \notin \Theta(c)) \Rightarrow (\overline{u(c, i)} = \overline{u(c, i+1)}) \quad (10)$$

A generalized routing where connections can switch tracks only at predetermined columns is essentially the conjunction of formulae (5), (6), (7) and (10).

**Table 1. Dogleg-free Segmented Routability**

N	M	T	clauses	literals	routability	Chaff
30	15	10	4979	10271	Y	0.04
50	15	15	6651	13675	Y	0.01
50	18	15	9564	19576	Y	0.01
50	20	15	11594	23686	Y	0.03
75	25	20	26141	53555	Y	55.04
100	15	10	5100	10544	Y	1.22
100	18	10	7163	14739	N	17171.20
100	30	25	46340	94328	Y	0.41
100	30	30	54092	109892	Y	0.17
100	30	36	66549	136157	Y	0.53
100	40	36	111937	227953	Y	1.42
100	45	36	140375	285339	Y	1.35
100	50	36	172741	350581	Y	3.97

### 5. EXPERIMENTAL RESULTS

We formulated the segmented routability problems with boolean equations in Sections 3 and 4, compiled them through BCSat [7] into DIMACS format, and used Chaff [8] to solve these problem instances. The runtime is in seconds on 850MHz Pentium®III with 1GB RDRAM (only small part of memory has been used).

Table 1 shows the routability checks for conventional segmented channels with no doglegging. The columns N, M, T corresponds to the number of columns, nets, tracks respectively. It takes a lot longer time to verify unroutability (for N=100, M=18, T=10) than routability. This is because it's much easier for the SAT checkers to find one satisfiable instance than searching every possible case for unsatisfiability. The parameters of the largest channel model (N=100, T=36) are close to Actel's family A1280 [6].

Table 2 shows the routability checks for segmented channels with doglegging. Since there are more flexibility in doglegging, the search time is generally longer compared to dogleg-free cases.

**Table 2. Doglegged Segmented Routability**

N	M	T	clauses	literals	routability	Chaff
20	6	5	1447	3167	Y	0.01
20	7	5	2061	4491	Y	0.03
20	8	5	2155	4619	N	0.02
20	9	5	1697	3483	N	0.02
20	9	7	2643	5473	Y	0.01
50	20	15	32396	70882	Y	0.60
75	25	20	126027	283005	Y	18999.8

### 6. CONCLUSION

We have studied a satisfiability-based method for solving segmented channel routability. The approach transformed the routing task into a single, large Boolean equation with the property that any assignment of input variables that satisfies the equation specifies a valid routing. Absence of a satisfying assignment implies that the channel is unroutable. Experimental results demonstrate its time-efficiency and applicability to large problem instances.

### 7. REFERENCES

- [1] S. Devadas, "Optimal layout via Boolean satisfiability," in *Proc. ACM/IEEE ICCAD*, 1989, pp. 294-297.
- [2] G. Nam, F. Aloul, K. Sakallah and R. Rutenbar, "A comparative study of two Boolean formulations of FPGA Detailed routing constraints," in *Proc. ACM ISPD '01*, pp. 222-227, 2001.
- [3] V.P. Roychowdhury, J. Greene, and A. El Gamal, "Segmented channel routing," *IEEE Trans. on CAD*, 12(1), pp. 79-95, 1993.
- [4] S. Thakur, Y. W. Chang, D. F. Wong, and S. Muthukrishnan, "Algorithms for an FPGA switch module problem with application to global routing," *IEEE Trans. on CAD*, vol. 16, 1997.
- [5] R. G. Wood, R. A. Rutenbar, "FPGA routing and routability estimation via Boolean satisfiability," *IEEE Trans. on VLSI Systems*, 6(2), 1998.
- [6] C. Yang, S. Chen, J. Ho, and C. Tsai, "Efficient routability check algorithms for segmented channel routing," *ACM Trans. on Design Automation*, 5(3), July, 2000, pp. 735-747.
- [7] T. Junttila and I. Niemelä, "Towards an Efficient Tableau Method for Boolean Circuit Satisfiability Checking," in J. Lloyd et al., editors, *Computational Logic; First International Conference, Lecture Notes in Artificial Intelligence*, vol. 1861, pp. 553-567, July 2000.
- [8] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, S. Malik, "Chaff: Engineering an Efficient SAT Solver," Design Automation Conference, Las Vegas, June 2001.
- [9] X. Song, W. N. N. Hung, A. Mishchenko, M. Chrzanowska-Jeske, A. Coppola, and A. Kennings, "Board-Level Multiterminal Net Assignment," in *Proc. ACM Great Lakes Symposium on VLSI*, New York City, April 18-19, 2002.