Segmented Channel Routing with Pin Rearrangements via Satisfiability

Fei He*†, William N. N. Hung‡, Xiaoyu Song§, Ming Gu† and Jiaguang Sun†
* Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China
† School of Software, Tsinghua University, Beijing, P.R. China
‡ Synplicity Inc., California, USA
§ ECE Department, Portland State University, Oregon, USA

***Abstract*— We address a new segmented channel routing problem with pin rearrangements in FPGA Technology. In our routing model, the pins in each module have certain degree of freedom to be rearranged. With this flexibility, the wire routability can be improved in segmented channel routing. We present an efficient SAT-based approach to solve the problem. We use one of the best SAT-solvers, zChaff, to perform our experiments. Experimental results show the promising performance of the method.**

I. Introduction

Field Programmable Gate Arrays (FPGA's) combine the flexibility of mask programmable gate arrays with the convenience of field programmability [1]. It is widely used as an important technology in hardware emulators for rapid prototyping of designs.

As discussed in [2], the FPGA architectures can be classified into two classes: island-style architecture and row-based architecture. The row-based architecture (segmented channel) [3] is consisted of rows of logic cells separated by segmented routing channels. Unlike the conventional channel architecture, the segmented routing channel contains predefined wiring segments of various lengths that are interconnected by programmable switches. These switches include vertical switches at the crossing of vertical segment and horizontal segment, and horizontal switches at the joint of two adjacent horizontal wiring segments. By programming these switches, we can build paths between terminals that belong to the same net. Due to manufacturing technologies, the switches have significant resistances and capacitances. So for nets should be routed using as little switches as possible.

Conventional channel routers have two assumptions: (i) the logic blocks that form the border of the channel are fixed; and (ii) the pins on the boundary of every block are fixed. Thus, if a routing instance cannot be routed by the conventional routing model, we have no remedial measure. The logic block locations are determined by the placement algorithm, so they are not easily changeable by the router alone. However, the pin positions within each block are determined by the block's internal logic configuration. Since there are usually many alternative logic configurations, the pin arrangements within each block can be changed. If we give some degree of freedom to rearrange the pins for each block, then some routing cases that are non-routable in conventional routers become routable. As a result, the routability is higher with the new formulation. Even for the cases that are routable in conventional routers, we may also find better routing results with this flexibility in which less switches (segments) are used. As a result, the quality of the routing result is improved.

The idea of channel pin assignment was first proposed by Gopal [4] and have been studied in [5]–[8]. One proposal assumed that pin positions are fully permutational. Some proposed movable pins, where the either side of the channel may have empty (unused) slots such that pins can be moved to these positions. However all these studies are limited to conventional channel architectures. Their main objective was to minimize the conventional channel width (number of tracks) or to minimize the routing cost. In this paper, we consider channel pin assignment in segmented channel routing that arises in FPGA designs. We define this new problem of segmented channel routing with pin rearrangement. The segmented channel routing is different from conventional channel because the channel width (number of tracks) is already fixed by the FPGA architecture. Our objective is to increase routability and to minimize the routing cost (e.g. number of segments used by each net).

Our pin rearrangement is specified by the user as a set of possible permutation patterns (ordering of pins) for each block. The reason for our proposal is that not every permutational pattern may be feasible in hardware. So our router only considers feasible permutational patterns specified by the user. These permutation patterns may include unused pin positions. Hence the problems of permutational pins and movable pins are all subsets of our formulation.

We present a formal description for the problem of segmented channel routing with pins rearrangement. An approach based on satisfiability is proposed to solve this problem. We use a set of variables to designate the specific detailed route for each net and the adopted pins permutation pattern for each block. Based on these variables, both the horizontal constraint and vertical constraints for the problem are translated into Boolean equations. Any assignment of Boolean variables that satisfies the conjunction of all the equations specifies a valid routing.

The rest of this paper is organized as follows. In Section II, we introduce related work about the problem. In Section III, we present a formal problem description. Section IV develops the satisfiability formulations. Section V tests our approach for

0-7803-8834-8/05/$20.00 ©2005 IEEE.

performance by benchmarks. Section VI concludes this paper.

II. Related Work

In [4], the terminals are assumed to be movable and the channel routing is modeled as the process of finding the optimal wiring solution with some objective functions minimized. Different objective functions are considered in [4]. For some of which, they presented polynomial time algorithms, and for others, they proved they are NP-completeness. In [5], the authors consider the channel pin assignment problem subject to some constraints, such as the terminal position constraints, terminal order constraints and so on. In [6], a specified problem, named as performance-driven channel pin assignment problem that takes the performance-driven net into account is proposed. They prove this problem is NP-complete, and present a polynomial time algorithm for the sub problem which assume each module has at most 2 pin assignment scheme.

Many successful applications showed that the satisfiability-based approach is more efficient than other decision methods ([9], [10]) for many NP-complete problems occurred in layout. Davadas [11] developed a simple formulation of classical two-layer channel routing as Boolean satisfiability. For island-style FPGAs, Wood [12] and Nam [13] presented a satisfiablity-based approach respectively. For the row-based FPGAs, Hung [14] developed a set of Boolean equations whose conjunction determines the routability of this channel. In this paper, we extend the method shown in [11], [14] to our problem.

III. Preliminaries and Definitions

According to the grid-based model, a channel is divided into grids by some horizontal and vertical cut lines. Pins in top and bottom boundaries can only be connected along these cut lines. The horizontal cut line is known as track, and the vertical cut line is known as column.

As defined in [3], consider a channel with height $P+1$ and length $L+1$, there are P tracks and L columns, where the tracks are numbered from bottom to up, columns are numbered from left to right. Let Γ be the set of P tracks and Ξ be the set of L columns. Each track t is separated into a set G_t of g_t segments by $g_t - 1$ switches. We use $s_{t,i}$ to denote the segment i on track t, where the indexes of segments are numbered from left to right. Let $left(s)$, $right(s)$ be the leftmost and rightmost column in which the segment s is present respectively. Since each switch need occupy a column, then $left(s_{t,i+1}) = right(s_{t,i}) + 1$ for all $t = 0, \ldots, P-1$ and $i = 0, \ldots, g_t - 1$.

The top and bottom boundaries of the channel are composed of the boundaries of a set Ω of M blocks, as shown in Fig. 1. Let $left(b)$, $right(b)$ be the leftmost and rightmost columns in which the block b is present respectively. Blocks are indexed from left to right according to their leftmost column. The position for each block is fixed, and the terminals assigned to each block are known. For each block b to which p pins are assigned, we can arrange these pins by k_b different orders. Consider each arrangement for these pins as a permutation pattern, and then there are k_b permutation patterns for block b. If the pins in a block are rearranged by some order, we say this block is implemented by the corresponding permutation pattern.

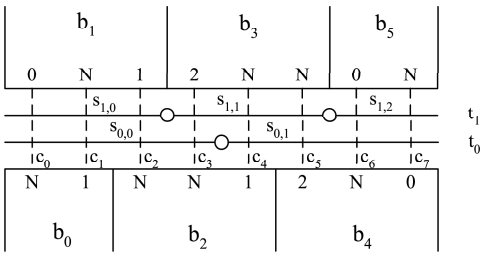

Fig. 1. A segmented channel example with P=2, L=8.

If the pins can be rearranged with complete freedom, the number of permutation patterns for each block would be $N!$, where N is number of pins. But in real world, not every permutation is allowed. The pins rearrangement must be performed under the precondition that the structural intent of the functional block remains unchanged. In practical routing, the precise degree of the freedom for pins rearrangement relies on the designer, with the technology-dependent design rules perhaps been take into account. Different designers may have different constraints for the pins arrangement. They should specify a set of allowed permutation patterns for each block before the satisfiability check.

Let Φ be the set of n nets. A net is a connection of terminals to be connected. The span of a net n is defined by its leftmost terminal ($left(n)$) and rightmost terminal ($right(n)$). However, before every block is implemented, we cannot determinate the exact position of the leftmost and rightmost terminals of a net. Fortunately, from the problem specification we can predicate a set of blocks, in one of which the leftmost terminal (or rightmost terminal) is present in. The set for leftmost terminal of net n is known as *left critical modules* for net n, denoted as $LC(n)$; and the set for rightmost terminal of net n is known as *right critical modules* for net n, denoted as $RC(n)$. Consider the example shown in Fig. 1, the leftmost terminal of net n_0 can be present in only b_1, and the rightmost terminal of net n_0 may be present in either b_4 or b_5, so $LC(n_0) = \{b_1\}$, and $RC(n_0) = \{b_4, b_5\}$. Obviously, the sets $LC(n)$ and $RC(n)$ have at most two elements respectively.

The exact position for any pin on a module cannot be determined until the module is implemented. Assume the net n has a pin on the boundary of the module b, also assume the module b is implemented by the pin assignment k, then the index of column on which the pin of net n is assigned can be determined by the problem specification, we refer to this as $POS(b, k, n)$.

In this paper, we restrict our attention to the special case that a net is assigned to at most one track. We refer to this case as *dogleg-free segment channel routing*. It is reasonable for us to ignore the doglegged channel routing as the "doglegs" tend

to add greatly to parasitic capacitances and can considerably degrade the circuit performance.

IV. Routing via Boolean Satisfiability

We define a *permutation variable* $\vec{x}(b)$ for each block b as the binary representation of the selected permutation pattern, i.e. $\vec{x}(b) = x_1(b)x_2(b)\cdots x_\alpha(b)$, where $x_i(b) \in \{0,1\}$ for $1 \le i \le \alpha$, and $\alpha = \lceil \log_2(k_b) \rceil$. As in [14], we also define a *track variable* $\vec{y}(n)$ for each net n as the binary representation of the track index where net n is assigned to, i.e. $\vec{y}(n) = y_1(n)y_2(n)\cdots y_\beta(n)$, where $y_j(n) \in \{0,1\}$ for $1 \le j \le \beta$, and $\beta = \lceil \log_2(P) \rceil$. Let $X = (\vec{x}(b_1), \vec{x}(b_2), \ldots, \vec{x}(b_M))$, and $Y = (\vec{y}(n_1), \vec{y}(n_2), \ldots, \vec{y}(n_n))$.

Since each net must be assigned to one track,

$$\forall n \in \Phi, 0 \le \vec{y}(n) \le P - 1 \tag{1}$$

Since each module must be implemented by a permutation pattern,

$$\forall b \in \Omega, 0 \le \vec{x}(b) \le k_b - 1 \tag{2}$$

A. Unlimited Segment Routing

For unlimited segment routing, there is no limit for the number of segments occupied by each net. We define a occupy function for net n to segment s on track t:

$$Occupy(n,s,t) = (right(s) \ge left(n)) \wedge (left(s) \le right(n)) \wedge (\vec{y}(n) = t)$$

where $left(n)$ and $right(n)$ denotes the leftmost and rightmost column of net n respectively,

$$left(n) = \min\{POS(b, \vec{x}(b), n) | b \in LC(n)\} \tag{3}$$

$$right(n) = \max\{POS(b, \vec{x}(b), n) | b \in RC(n)\} \tag{4}$$

By combining above three formulas, we have

$$\begin{aligned} Occupy(n,s,t) = {} & \exists b_l \in LC(n), \exists b_r \in RC(n), \\ & (right(s) \ge POS(b_l, \vec{x}(b_l), n)) \\ & \wedge (left(s) \le POS(b_r, \vec{x}(b_r), n)) \\ & \wedge (\vec{y}(n) = t) \end{aligned}$$

Since each segment on each track cannot be occupied by more than one net,

$$\begin{aligned} & \forall t \in \Gamma, \forall s \in G_t, \forall n_1, n_2 \in \Phi, n_1 \ne n_2 \rightarrow \\ & \qquad \neg(Occupy(n_1,s,t) \wedge (Occupy(n_2,s,t)) \end{aligned} \tag{5}$$

The overall routability check for dogleg-free unlimited segment routing with pins rearrangement is the conjunction of (1), (2), and (5).

B. K-Segment Routing

In K-segment routing, a net can occupy at most K segments on a track [3]. Thus there is:

$$\forall n \in \Phi, \forall t \in \Gamma, (\vec{y}(n) = t) \rightarrow \sum_{s \in G_t} Occupy(n,s,t) \le K \tag{6}$$

The overall routability check for dogleg-free K-Segment routing with pins rearrangement is the conjunction of (1), (2), (5) and (6).

C. Performance Driven Nets

A performance driven net is a net satisfied following two conditions: 1) its span is shorter than a predetermined bound; 2) it is assigned to a set of predetermined tracks. These requirements sometimes originate due to phyical design issues [15], [16]. Let ϑ be the set of performance driven nets, Σ be the set of performance driven tracks, and l_n be the span upper bound of net n, then the constraint 1) for performance driven nets can be represented as:

$$\forall n \in \vartheta, right(n) - left(n) \le l_n$$

Substitute (3) and (4) into the above formula, we have

$$\begin{aligned} & \forall n \in \vartheta, \forall b_l \in LC(n), \forall b_r \in RC(n), \\ & \qquad (POS(b_r, \vec{x}(b_r), n) - POS(b_l, \vec{x}(b_l), n) \le l_n) \end{aligned} \tag{7}$$

Constraint 2) for performance driven nets can also be represented as:

$$\forall n \in \vartheta, \exists t \in \Sigma, \vec{y}(n) = t \tag{8}$$

The overall routability check for dogleg-free performance driven nets routing with block pins reassigned is the conjunction of (1), (2), (5), (7) and (8).

V. Experimental Results

We created a number of random benchmarks using a generator similar to [14]. All experiments are conducted on Linux with 850MHz Pentium III processor. We used zChaff [17] as our SAT solver. Table I shows the performance of our router on randomly generated benchmarks. For each benchmark, we randomly selected 10% of the tracks to be performance tracks and randomly selected 10% of the nets to be routed on these performance driven tracks. For each problem instance, with applied the router with fixed pins (i.e. the pin locations cannot be permuted), and compared with some movable pin patterns. As shown in Table I, the number of clauses and literals for the CNF increased due to pin rearrangement, which also lead to an increase in CPU seconds. But the problem has changed from unroutable to routable.

For more practical benchmarks, we used the segmentation design proposed by Zhu and Wong [18] and also used by Hung et al [14]. The benchmarks provide a set of channels for segmented channel routing with fixed pins. But not all channels and routing specifications are routable, as reported in [14]. We take the unroutable specifications and re-arrange their pins to make them routable. The results are shown in Table V.

The experimental results demonstrate that our approach is indeed more efficient than the traditional segmented channel router by improving the routability with an acceptable increase of CNF size and CPU time.

VI. Conclusion

In this paper, we consider segmented channel routing with pin rearrangement. In our routing model, the pins assigned to each module can be rearranged according to some user specified patterns. Utilizing this freedom, we improve the

TABLE I

RANDOM BENCHMARKS

Track	Column	Net	Fixed Pins				Movable Pins			
			Clause	Literal	CPU sec.	Routable	Clause	Literal	CPU sec.	Routable
20	60	50	40612	2121	0.01	No	71729	5995	4.55	Yes
30	85	75	131545	4613	0.02	No	204802	12991	1.32	Yes
40	100	70	150611	5374	0.01	No	235780	15358	1.14	Yes
40	150	80	192357	6153	0.02	No	296876	17616	1.60	Yes
40	500	75	170986	5803	0.02	No	265604	16631	1.08	Yes
50	150	75	213434	7094	0.03	No	349437	20762	0.26	Yes
50	200	80	244769	7654	0.03	No	376737	21688	1.02	Yes
50	300	100	354863	9423	0.05	No	543401	27583	5.31	Yes
80	450	150	1187729	21307	0.12	No	1743613	63393	4.33	Yes

TABLE II

ZHU AND WONG'S BENCHMARKS

Benchmark	Fixed Pins				Movable Pins			
	Clause	Literal	CPU sec.	Routable	Clause	Literal	CPU sec.	Routable
D1	56039	3007	0.01	No	119627	10179	0.10	Yes
D2	27980	2001	0.01	No	65919	6110	0.05	Yes
D3	74367	3744	0.02	No	183565	13614	0.65	Yes
D4	99010	4515	0.01	No	231097	15733	2.37	Yes
D5	9591	956	0.01	No	27177	3253	0.02	Yes
D6	24732	1901	0.01	No	65541	6367	0.05	Yes
Geo	58248	3284	0.01	No	165449	12926	0.27	Yes
Norm	10700	1089	0.01	No	39347	4210	0.03	Yes

routability of the segmented channel routing, and present a satisfiability based approach to solve this model. We used zChaff to perform our experiments. The experimental results show the improved routability of our approach to traditional segmented channel router. It should be noted that the efficiency of our satisfiability-based approach would directly benefit from the innovations in SAT solver research. Our approach is also incremental, it can be extended to take more constraints into account without changing other parts.

REFERENCES

[1] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field Programmable Gate Arrays*. Norwell, MA: Kluwer, 1992.

[2] S. Trimberger, "Effects of FPGA architecture on FPGA routing," in *Proc. Design Automation Conference*, San Francisco, CA, June 1995.

[3] V. P. Roychowdhury, J. Greene, and A. E. Gamal, "Segmented channel routing," *IEEE Transactions on CAD*, vol. 12, no. 1, pp. 79–95, 1993.

[4] I. S. Gopal, D. Coppersmith, and C. K. Wong, "Optimal wiring of movable terminals," *IEEE Transactions on Computers*, vol. C-32, pp. 845–858, September 1983.

[5] Y. Cai and D. F. Wong, "Optimal channel pin assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, pp. 1413–1424, 1991.

[6] T. W. Her, T. C. Wang, and D. F. Wong, "Performance-driven channel pin assignment algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 849–857, 1995.

[7] Y. Cai and D. F. Wong, "Minimizing channel density by shifting blocks and terminals," in *Proc. International Conference on Computer-Aided Design*, 1991, pp. 524–527.

[8] C. Y. R. Chen and C. Y. Hou, "A pin permutation algorithm for improving over-the-cell channel routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 1030–1037, 1995.

[9] Y. Wang *et al.*, "Optimal Symmetry Detection for OKFDDs," in *Proc. IEEE Midwest Symposium on Circuits and Systems (MWSCAS 2000)*, August 2000.

[10] ——, "Single-faced Boolean Functions and their Minimization," *Computer Journal*, vol. 44, no. 4, pp. 280–291, 2001.

[11] S. Devadas, "Optimal layout via boolean satisfiability," in *Proc. International Conference on Computer-Aided Design*, 1989, pp. 294–297.

[12] R. G. Wood and R. A. Rutenbar, "FPGA routing and routability estimation via Boolean satisfiability," *IEEE Transactions on VLSI Systems*, vol. 6, no. 2, 1998.

[13] G. J. Nam, F. Aloul, K. A. Sakallah, and R. A. Rutenbar, "A comparative study of two Boolean formulations of FPGA detailed routing constraints," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 688–696, 2004.

[14] W. N. N. Hung, X. Song, E. M. Aboulhamid, A. Kennings, and A. Coppola, "Segmented channel routability via satisfiability," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2004, accepted for publication.

[15] S. Gao, K. Thulasiraman, *et al.*, "Homotopic Routing of Multi-terminal Nets with Wire-length Minimization," *Journal of Circuits, Systems and Computers*, vol. 6, pp. 1–14, 1996.

[16] S. Gao and K. Thulasiraman, "Parallel Algorithm for Integrated Floor Planning and Routing," in *Proc. International Conference on High Performance Computing*, December 1995, pp. 457–462.

[17] L. Zhang, C. F. Madigan, M. W. Moskewicz, and S. Malik, "Efficient conflict driven learning in a boolean satisfiability solver," in *Proc. International Conference on Computer-Aided Design*, November 2001, pp. 279–285.

[18] K. Zhu and D. F. Wong, "On Channel Segmentation Design for Row-Based FPGAs," in *Proc. International Conference on Computer-Aided Design*, November 1992.