Not Necessarily More Switches More Routability

Yu-Liang Wu
Dept. of Comp. Sci. and Engr.
Chinese Univ. of Hong Kong
Hong Kong, ylw@cse.cuhk.hk

Douglas Chang
Dept of Comp. Sci.
UC Santa Barbara
Santa Barbara, CA 93106

Malgorzata Marek-Sadowska
Dept. of Elec. & Comp. Engr.
UC Santa Barbara
Santa Barbara, CA 93106

Shuji Tsukiyama
Dept. of Elec. & Elect. Engr.
Chuo University
Tokyo, Japan 112

Abstract— It has been observed experimentally that the mapping of global to detailed routing in conventional FPGA routing architecture (2D array) yields unpredictable results. In [8,10,13], a different class of FPGA structures called Greedy Routing Architectures (GRAs), where a locally optimal switch box routing can be extended to an optimal entire chip routing, were investigated. It was shown that GRAs have good mapping properties. An H-tree GRA [10] with W^2+2W switches per switch box (SpSB) and a 2D array GRA [13] with $4W^2$+2W SpSB were proposed (W is the number of tracks in each switch box). Here, we continue this work by introducing an H-tree GRA with $W^2/2$+2W SpSB and a 2D array GRA with $3.5W^2$+2W SpSB. These new GRAs have the same good mapping properties but use fewer switches. We also show a class of FPGA architectures in which the mapping problem remains NP-complete, even with $6(W-1)^2+6W^2$ SpSB. This is close to the maximum number of SpSB which is $6W^2$.

1. Introduction

An FPGA is an array of pre-fabricated functional blocks and wire segments with user-programmability of logic and routing resources. Because of their fast turn-around time and economic manufacturing cost for low volume designs, FPGAs have been a rapidly growing medium for ASIC implementations. A popular FPGA technology is based on Look-Up-Tables (LUTs) and static RAM (SRAM) (e.g., Xilinx [14]). The routing architecture used in this FPGA technology is described in [3,14] and is shown in Fig. 1.

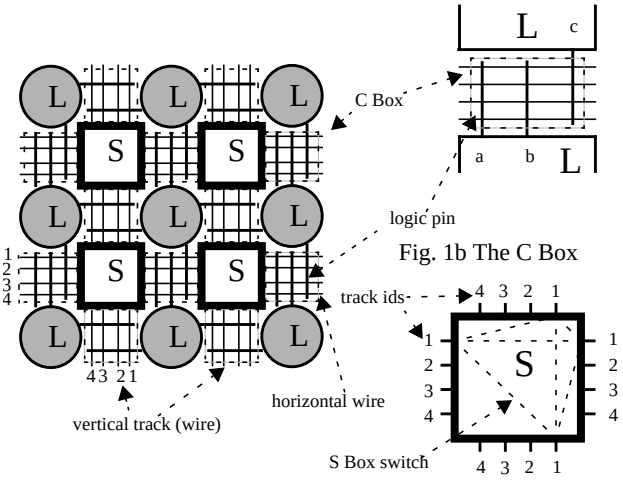

Fig. 1a The Routing Model

Fig. 1b The C Box

Fig. 1c The S Box

Routing is crucial in the FPGA design automation process because the feasibility of a design often depends on the routing resources rather than on the logic gates. The conventional approach to FPGA routing is a 2-step global/detailed routing scheme [3,4]. Thus, the ability to map an arbitrary global routing of all nets to a feasible detailed routing is a critical problem. In this paper, we investigate the following three global to detailed mapping properties imposed by the routing architecture:

Mapping Decision Problem (MDP) (predictable routing): Can the decision, whether an arbitrary global route can be mapped to a feasible detailed route be made in polynomial time?

Constant Mapping Ratio (CMR): Is there a constant ratio bound on the number of tracks required to complete detailed routing over global routing channel density?

Perfect Mapping (PM): Can a detailed routing using the same number of tracks as the global routing be found (even by exhaustive search)?

We will study these problems on two routing architecture, the 2D array routing structure with differing addition of switches, and an H-tree routing structure (Fig. 3). In [13], the mapping problem of a Xilinx style FPGA architecture with minimum switching flexibility was formulated as a 2-D interval packing problem. It was shown to be NP-complete and have no constant mapping ratio bound for any architecture of that style and with the same minimum hardware switching flexibility. In [10], the concept of a greedy routing architecture (GRA) was proposed. A GRA is an architecture with the property that a globally optimal routing of all switch boxes can be extended from a locally optimal routing of a single switch box. Since the locally optimal routing needs to be extended to a global solution, a GRA has a routing order on its switch boxes. The H-tree GRA [10] with W^2+2W switches per switch box (SpSB) has a perfect mapping with a polynomial time algorithm. The routing order for the H-tree is from the root down to the leaves. Another GRA with $4W^2$+2W SpSB was proposed in [13] and is a 2D array routing structure with a snake-like routing order. This snake GRA also has a perfect mapping with a polynomial time algorithm.

In this paper, we improve on the above results by creating an H-tree GRA with $W^2/2$+2W SpSB and a snake GRA with $3.5W^2$+2W SpSB. Since the H-tree architecture is a type of hierarchical FPGA [5], this indicates that hierarchical FPGAs may be a good choice in situations where quick and/or predictable place and route is needed. Having quick and predictable place and route is becoming increasingly important as increasingly large numbers of FPGAs are used in a single system. For example, in a million gate logic emulation system like Teramac [1], using a commercial FPGA such as the Xilinx 3000 and 4000 would have been impractical due to the time and unpredictibility of place and route. Teramac instead uses a custom FPGA chip (Plasma [2]) that allows quick (3 seconds) place and route. We also show an architecture with $6(W-1)^2$+6 SpSB, in which the MDP is still NP-complete. Since this is

ASP-DAC '97
0-89791-851-7$5.00 © 1997 IEEE

close to the maximum number of SpSB of $6W^2$, this indicates that even the addition of a substantial number of switches may not help the worst case complexity of the MDP.

This paper is organized as follows. In section 2 we introduce basic terminology. In sections 3 and 4 we study the Xilinx routing structure after adding switches in various ways. In section 5 we investigate GRAs. Finally we give conclusions in section 6.

2. Routing Architecture and Terminology

In Fig. 1 we show a simple routing architecture for a two-dimensional array of logic cells. Each logic cell is marked L and can be configured to be a Look-Up-Table (LUT), flip-flop, etc. [3,14]. Prefabricated wire segments run between the cells in vertical and horizontal channels with each wire length spanning an L block. Each routing track within a channel is assigned a track id as shown in Fig. 1. Programmable routing switches are grouped inside the connection (C) boxes and switch (S) boxes (Fig. 1b, 1c). The C boxes contain routing switches that can be programmed to connect logic block pins to wire segments. The S boxes contain switches that allow one wire segment to be connected to another. One switch can connect two wire segments. In Fig. 1c each dashed line is counted as a single switch.

The flexibility of a C block, is defined to be the number of tracks a logic pin can connect to. In this paper, we assume that the flexibility of all C boxes is complete, i.e. each logic pin has a switch to connect to any wire inside the C box containing the logic pin. Therefore, the routing property of the architectures investigated is dependent on the S box structures only. The flexibility of a wire (track) of an S box is the number of other surrounding wire segments this wire has a switch to connect to. For example, consider Fig. 1c. Each wire on one side has 3 possible connections to other sides (only track 1 switches are shown), therefore, the flexibility of each track of this S box is 3. The population of an S box is the number of switches contained in it.

We define a domain as a maximal set of wire segments such that any two segments of the set are connectable through (S box) switches. Let D be the number of domains in a routing architecture, and let D_i represent the i^{th} domain.

The 2-D routing structures can be divided into two major classes: disjoint and non-disjoint architectures. A disjoint routing architecture is a structure in which all S box switch connectable wire segments on the chip are (disjointedly) partitioned into $D > 1$ domains such that each domain D_i contains a particular number ($|D_i|$) of wire segments in each C box , and each wire of a domain has an S box switch connectable to each same domain wire on other neighboring S box sides. We refer to the number of wire segments from a given C box in D_i as the domain capacity of D_i, or $|D_i|$. Thus the number of switches in an S box associated with a domain D_i is $6|Di|^2$. Any other type of routing architecture is a non-disjoint routing architecture. We further divide disjoint routing architectures into two categories, uniform and non-uniform. A disjoint architecture is uniform if all domain capacities are the same; otherwise, it is non-uniform. In a uniform disjoint architecture, the domain capacity is represented by D_c. For example, the Xilinx 4000 [14] like routing architecture has uniform disjoint domains, with $D_c = 1$.

A net is a set of pins that are assigned the same signal and are to be connected by wire segments. A global router decides through which switch boxes and connection boxes each net will pass. A detailed router assigns wire segments and switches to connect each net based on the net path given in the global route.

A channel density resulting from a global routing, W_g, is the maximum number of global routes which run in parallel in any channel. W_d is the minimum number of tracks needed to route the nets given the global routes. Note that the number of pre-fabricated tracks, W, on a chip is fixed, so if $W_d > W$, routing cannot be completed. A mapping ratio (MR) is the value of W_d/W_g.

3. Mapping Property of Disjoint Architectures

In [13], only the Xilinx style architecture with $D_c = 1$ is analyzed. In [10], it was shown that the mapping decision problem is NP-complete for any uniform disjoint architecture with arbitrary domain capacity when the number of domains is greater than 2. In the case of 2 domains, if $|D_1|=|D_2|=1$ (W=2), the MDP can be solved by a polynomial 2-colorable graph algorithm. However, here we further show that the problem becomes NP-complete if the capacity of any domain is increased. The result is stated in the following theorem.

Theorem 1. The problem [MDP on 2-D non-uniform disjoint routing architecture] is NP-complete for S box topology with D = 2, $|D_1| = W-1$, and $|D_2| = 1$ for any $W > 2$.

Proof: Reduction from the NP-complete [non-negated one-in-three 3SAT] problem. We give the details in the appendix.

Combining the results of this theorem and [10], we conclude that the MDP is NP-complete for a broad range of disjoint architecture (uniform or non-uniform) even with the addition of a large number of switches. This result holds for both 2-pin and multi-pin routing cases. It was also shown in [10] that all 2-D disjoint routing architectures, either uniform or non-uniform, have no constant mapping ratio bound if $W_g > \max(|D_i|)$, where $|D_i|$ is the domain capacity of the i^{th} domain. Consider a non-uniform disjoint routing architecture with just two routing domains, one with capacity W-1 and the other with capacity one. In such a routing architecture, the switch population of a switch box is $6(W-1)^2+6$, which is close to the complete flexibility of $6W^2$; however, neither a polynomial mapping algorithm nor a constant mapping ratio bound is achievable for such an FPGA architecture.

4. Extremal Routing Architectures

There are 6 kinds of side-to-side switch connection relations of an S box as shown in Fig. 2. A one-to-one wire connection mapping of different sides can be represented by a permutation mapping. The identity permutation, I, represents the mapping between wires of the same track id. A complete mapping is one in which a wire on one side can connect to any wire on the other side (requires W^2 switches between the 2 sides).

If only the identity and complete mappings are used in switch box designs (called *extremal switch-block structure*), it was shown [8] that, within this class of architectures, the switch box structure using 3 complete mappings and 3 identity mappings forming a cycle has the lowest number of SpSB ($3W^2+3W$) that can yield a polynomial mapping algorithm. However, since this cyclic extremal architecture has a worst mapping ratio of 3/2, it may take $3(3/2)^2W_g^2+3(3/2)W_g$, which is $6.75W_g^2+4.5W_g$ SpSB to

complete routing cases with global routing density of W_g. We will illustrate more details in Section 5.3.

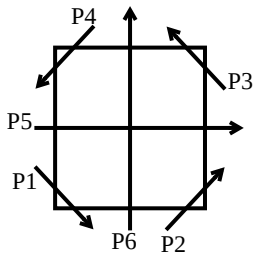

Fig. 2. Side-to-side track mappings of an S Box

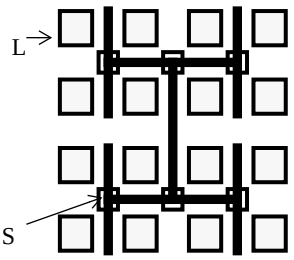

Fig. 3. H-tree architecture

5. Minimizing Greedy Routing Architectures

From the above results we conclude that blindly adding switches to switch boxes will not, in the worst case, make an FPGA more routable. Thus the switches must be added in some intelligent manner so as to ensure the routability of the final chip. In [6] universal switch modules were studied. These are switch boxes that can route any given global routing around the switch box (that satisfy the local routing constraints of this individual switch box). But, this local optimality does not extend across the entire chip. So even though each switch box is locally optimal, that does not guarantee an optimal result across the entire FPGA chip. The authors claim, as most FPGA net lengths are short, universal switch modules can produce good routing results on practical cases.

In the following we will develop GRAs achieving our desired routing goals while using less switches than all other proposed structures [8,13]. We will introduce a minimal 3-way switch box structure, then use it as a basis to construct a 4-way switch box. The 3-way switch box will be used in the H-tree GRA and extended into the 4-way S box used in the 2D GRA.

5.1 H-tree Architecture and 3-way switch box

A GRA in the form of an H-tree (Fig. 3) using a 3-way switch box with W^2+2W switches which is polynomially routable with a perfect mapping was proposed in [10]. The H-tree architecture is a GRA because the 3-way switch boxes are designed to be able to route any global routing with the top side already detailed routed. By routing the H-tree from root down to leaves, each 3-way switch box will always be routable, and thus the entire tree will be routable. In the following, we improve the 3-way switch box by reducing the number of switches to $W^2/2$+2W but maintaining the polynomial routability with a perfect mapping. We furthermore show that the removal of any switch in our $W^2/2$+2W switch box would result in losing the perfect mapping, thus the 3-way switch box has a minimal number of switches. Without loss of generality, we assume W is even in the following discussion.

Fig. 4a shows the structure of a 3-way switch box with W = 4. This new 3-way switch box has an identity mapping between top-right (T_i to R_i) and top-left side-pairs (T_i to L_i), and each left track L_i can be connected to R_{i+1}, R_{i+2}, ..., $R_{i+W/2}$ (subscripts are always mod W, but for clarity, the mod W will be omitted). This 3-way switch box has $W^2/2$+2W switches. Fig. 4b shows a routing example using this 3-way switch box.

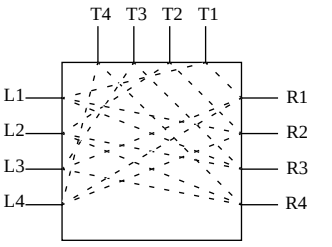

Fig. 4a. The optimal 3-way greedy routing S Box, where the top side is pre-assigned.

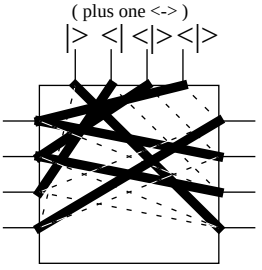

Fig. 4b. A routing example with pre-assigned nets shown on the top side.

The connections between the left and right sides of the switch box can be viewed as a bipartite graph G=(V,E), where each wire, L_i or R_j maps to a vertex vL_i or vR_j in the graph , respectively, and there is an edge between vertex vL_i and vR_j iff there is a switch connecting wire L_i and R_j. Then we define adj(v) to be the set of vertices adjacent to v∈V, and adj(S), S ⊆ V to be $\bigcup_{v \in S} adj(v)$.

Lemma 1. Let G= (V,E) be as described above. If S = {vL_i | i∈{x, x+1, ..., x+m} (mod W) and 1 ≤ i ≤ W}, then the number of vR_js adjacent to S, |adj(S)| = min {W/2+|S|-1, W}. For any other kind of subset of vL_is, S', |adj(S')| ≥ min {W/2+|S'|, W}.

Proof: Consider a subset of vL_is, S'' = {vL_i | i ⊆ {1...W}}. If S'' is of the form S above, i.e. all tracks of S'' are (continuously) consecutive, then let vL_x be the first in the sequence. Since vL_x∈S, by the construction of the switch box {vR_{x+1}, vR_{x+2}, ..., $R_{x+W/2}$} ⊆ adj(S). The addition of a consecutive track (vL_{x+1}) to S will increase the size of adj(S) by one until it reaches the maximum possible size, W. The addition of the first non-consecutive track (if S'' is of the form S' above) will increase the size of adj(S) by at least two (up to maximum size W). The lemma follows.

5.2 One side predetermined 3-way routing problem

A 3-way switch box has incoming and outgoing nets in only 3 directions. To build a greedy routing structure which can propagate an optimal local routing to an optimal entire chip routing, we define the [One-side predetermined 3-way routing problem] as follows:

Instance: A 3-way switch box routing structure, a global route of nets on the 3 surrounding C boxes of the 3-way box, with detailed route of one side predetermined.

Question: Does there exist a valid detailed routing of any global route surrounding the given switch box?

Theorem 2. The 3-way switch box described above and its variations can detail route any 3-way global routing with one side predetermined.

Proof: We will first consider the case where the predetermined side is the top side and the other two sides are the right and left sides (Fig. 4). With small variations, any side can be the predetermined side and the other two sides can be any two of the remain three sides of a 4-way S box. Note that there are 4 kinds of nets surrounding the 3-way switch box: a right-bend net (|>), a left bend net (<|), a horizontal net (<->), and a 3-way net (<|>). Each top

track can have been preassigned (detailed routed) to either a right-bend net, a left-bend net, or a 3-way net.

We first consider the case with only 2 pin nets, (excluding the <|> net). Later we show that the <|> net can be reduced to the 2 pin net case. In the following, we use the matching theorem [9], which states, given a bipartite graph G with vertex partition {S,T}, $|S| \leq |T|$, there exists a complete matching of S iff for every subset $Q \subseteq S$, $|adj(Q)| \geq |Q|$.

To route the 3-way switch box, first the fixed top side is routed. Without loss of generality, assume that there are at least as many <| nets as |> nets to be routed. Each <| (|>) net will occupy a wire L_i (R_j). Thus after routing the top side, a subset of wires on the left and right sides remain unused. To ensure that any possible global routing is detail routable with only W tracks, we must show that the bipartite subgraph including the unused wires and their connections always has a complete matching.

Let G=(V,E) be as described above. Let G'=(V',E') be the subgraph created from the remaining wires. Let V' = vL' $\cup$ vR', where vL' = {vL_i | wire i is an unused wire on the left side} and vR' = {vR_i | wire i is an unused wire on the right side}. Furthermore let vL = {vL_i | wire i is a used wire on the left side} and vR = {vR_i | wire i is a used wire on the right side}. Then we have the following facts. If $vL_i \in vL$ then $vR_i \notin vR$, also if $vR_i \in vR$ then $vL_i \notin vL$. We call this the mutual exclusive property. For example, if the L2 wire has been used for a <| net, it must be connected to the T2 wire, then the R2 wire must be unused because a |> net connected to R2 must also use the T2 wire.

Since there are at least as many <| nets as |> nets, we must have $|vL| \geq |vR|$ which implies $|vL'| \leq |vR'|$. Since we can have at most W <| and |> nets, we have $|vL| + |vR| \leq W$ which implies $|vR| \leq W/2$. Also $|vL'|+|vL| = W$ which implies $W-|vL'| \geq |vR|$. Thus we have $|vR| \leq \min\{W/2, W- |vL'|\}$

If we take any subset $S \subseteq vL'$ in G' then |adj(S) in G'| = |adj(S) in G - vR| $\geq$ |adj(S) in G| - |vR|. To satisfy the matching theorem we must show that |adj(S) in G'| $\geq$ |S|. There are three possible cases.

° If $|S| \leq W/2$ and S = {vL_i | i$\in$ {x, x+1, ..., y-1, y} (mod W) and $1 \leq i \leq W$} then by lemma 1, |adj(S) in G| = W/2+|S|-1. If $|vR| < W/2$ then the matching theorem is satisfied, but if $|vR| = W/2$, we get |adj(S) in G'| $\geq$ W/2+|S|-1 - W/2 = |S|-1. This does not satisfy the matching theorem. However, looking into the contents of vR we find that $vR_x \in vR$. This follows because $|vR| = W/2$ implies $|vL| = W/2$. Then the mutual exclusive property between vR and vL implies that vR' and vL' must also be mutually exclusive. Thus since $vL_x \in vL'$, we have $vR_x \notin vR'$ and so $vR_x \in vR$. Furthermore vR_x does not belong to adj(S) in G' because $|S| \leq W/2$. Therefore |adj(S) in G'| = |adj(S) in G - vR| = |adj(S) in G| - |vR| + |{vR_x}| = W/2+|S|-1 - W/2 + 1 = |S|. Thus the matching theorem is satisfied.

° If $|S| \leq W/2$ and S is not of the form {vL_i | i$\in$ {x, x+1, ..., y-1, y} (mod W) and $1 \leq i \leq W$}, then by lemma 1, |adj(S) in G| = W/2+|S|, since $|vR| \leq W/2$ we get |adj(S) in G'| $\geq$ W/2+|S|-W/2 = |S|, and the matching theorem is satisfied.

° If $|S| > W/2$ then by lemma 1, |adj(S) in G| = W, since $|vR| \leq W-|vL'|$ we get |adj(S) in G'| $\geq$ W-(W-|vL'|) = |vL'| $\geq$ |S|, since $S \subseteq vL'$. The matching theorem is satisfied.

Now let us consider the routing cases including <|> nets. Such routing instances can be converted to a 2 pin instance as follows. First each <|> net is replaced by a <-> net. Then we add <-> nets to the routing instance until the right or left side of the switch box will be fully used after routing (i.e., number of nets routed through the side will be W). Without loss of generality assume that the left side of the switch box is fully used. Then the above 2 pin analysis shows that this 2 pin routing instance is routable. Furthermore, each <|> net specified (on top side pins) can be connected as a <| net to the left side plus a <-> net. We are assured that the left side wire which the <|> net connects to is connected to a <-> net because the left side of the switch box is fully used. This completes the 3 way connection. Finally the unused <-> nets are removed. We give such a routing example in Fig. 4b.

Note that in the above proof, (also for lemma 1) there was nothing special about the top side having the fixed routing. Thus the above proofs also apply if the right side (left side) of the switch box has the predetermined routing and the identity mapping to the top and left sides (top and right sides). If we also consider rotation of the switch box, we can conclude that we have created a family of switch boxes that solves the [One-side predetermined 3-way routing problem].

Since bipartite matching can be done in polynomial time, we have shown that this 3-way switch box, used in an H-tree architecture can achieve a perfect mapping in polynomial time. Q.E.D.

We can further claim that, in fact, this structure is also the minimum routing structure to achieve the described routing problem.

Theorem 3. The 3-way switch box described above is the minimum structure solving the [One-side predetermined 3-way routing problem].

Proof : It is clear that none of the W top-right switches can be removed since it is the minimum switch set needed to route the case of W |> nets. (Similarly, the top to left W switches is minimal).

It can also be shown that W/2 is the minimum number of horizontal switches needed of each left side track. And any left side track L_x must at least be connectable to a right side track subset Y, s.t. Y = {R_y | $y \neq x$ } and $|Y| \geq W/2$. Otherwise there will be a routing case (with W/2 of each of the |>, <|, and <-> 2-pin nets) unroutable. We will prove this by showing that it is always possible to construct such a unroutable case.

Suppose that adj(L_x) = {R_y | y $\in$ Y, $Y \subseteq \{1, 2, .., W\}$} and $|Y| = W/2 - 1$. Then it is always possible to construct a Y', s. t. x $\in$ Y', $Y \subseteq Y'$, and $|Y'| = W/2$. Let us now assign all T_is, where i $\in$ Y', to be |> nets; all T_js, where j $\in$ X' = {1, 2, .., W} - Y', to be <| nets, and lastly we add W/2 <-> nets. Thus we have W/2 of each of |>, <|, and <-> nets in this routing instance. To route this set of nets, all of the wires (L_i, R_i, T_i where i $\in$ {1, 2, .., W}) must be used. Thus L_x must be used to connect a net. But T_x is used by a |> wire and all connectable right tracks of L_x have been used in connecting |> nets, thus L_x cannot be used to connect any net and this routing instance is not routable. The structure shown in Theorem 2 possesses the minimum number of switches in each of the 3 side-to-side connections. Q.E.D.

It is also interesting to observe that due to the mutual exclusive property of this routing structure, any switch connecting L_i and R_i, $1 \leq i \leq W$, would not contribute in any minimum routing structure

of this routing problem. We can somehow consider such kind of switches as *don't care switches* in this routing problem.

5.3 An Improved 2D array GRA

The above developed 3-way switch box that can route any global routing involving 3 sides can be used to create a 4-way box that can route any global routing of 4 sides by trivially adding a complete mapping from the 4^{th} side to the other 3 sides. This 4-way switch box can be used in a 2D array routing structure. To route such a chip, the first step is a 3-way routing on the 3 C boxes (left, bottom, and right) around each S box and propagating the routing in a row-by-row linear sequence (originally proposed in [13]) across the whole chip.

For example, in Fig. 5a, a 3-way routing is started from the C1 box, which is considered as the predetermined side and the C2 and C3 boxes are considered as the other two sides of the 3-way switch box. Then the process repeats from C3, which is now treated as the predetermined (detailed route fixed) side of S2 box, and propagates the routing to C4 and C5 boxes. The 3-way routing keeps going until it reaches the end of the row (C9 box here). Similarly, the whole process starts all over again from the next row, which starts from C10 box, ..etc. The final step of the routing is to connect the three left over side-to-side connections (Fig. 5b) around each switch box by using the 3 complete mapping connections. This step has to be done on every S box so as to finish the originally assigned 4-way global route. The same routing method can be applied if the routing S boxes are replaced by the extremal routing S boxes, however, as shown in the routing example Fig. 5c, a worst case mapping ratio of 3/2 could be happened.

Based on this approach, perfect routing can be guaranteed and the number of SpSB is $3.5W_g^2+2W_g$, which is less than the previous result of $4W_g^2+2W_g$ [13] and the extremal structure that requires $6.75W_g^2+4.5W_g$ SpSB to complete the worst case routing. Here we express the switch count in terms of W_g to reflect the effect of the worst case mapping ratio.

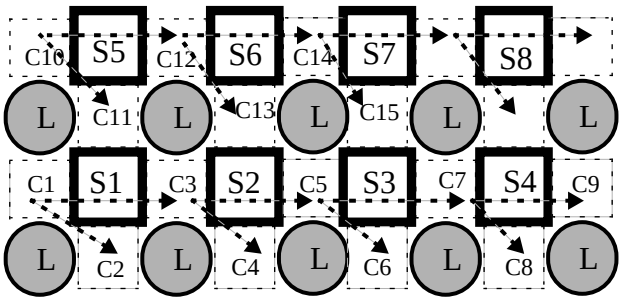

Fig. 5a Routing sequence of the 2D array GRA

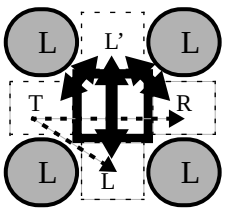

Fig. 5b Final make-up routing of a local S box

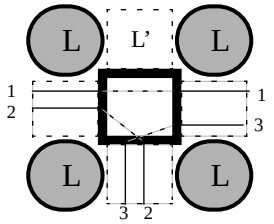

Fig. 5c A cyclic extremal S box routing where the Wd/Wg = 3/2

6. Conclusion

In the past, routing analysis on the Xilinx style FPGA architecture has been based on localized [6] or global-wise but limited to minimum switch flexibility structures [13] or limited to extremal structures [8]. In this paper, we further investigated these fundamental routing problems on the Xilinx style FPGA architecture with increased switching flexibility. Intuitively speaking, more hardware routing switches should bring better global to detailed routing mappability. However, we have found that purely adding switches does not necessarily reduce the complexity of these routing problems and bring better worst case mapping ratio. In particular, for a disjoint routing architecture with switch population of $6(W-1)^2+6W$, neither a polynomial routing mapping algorithm nor a constant mapping ratio bound can be achieved. On an extremal structure, which is non-disjoint, the lowest switch structure yielding a polynomial routing mapping algorithm is $3W^2+3W$. However, due to its worst mapping ratio of 3/2, it requires switch population of $6.75W_g^2+4.5W_g$ to guarantee routing completion for all routing cases with global routing density of W_g. On the other hand, we show that it is possible to achieve polynomial routing and perfect routing with a switch population of $3.5W_g^2+2W_g$ on a GRA. So far as these investigated routing properties are concerned, the GRA seems to be superior to other currently known architecture design styles.

7. Acknowledgments

This work was partially supported by National Science Foundation under grant MIP 9419119, and partially by Xilinx and Actel Corp through the California MICRO program.

Appendix

Theorem A.1. The problem [MDP on 2-D non-uniform disjoint routing architecture] is NP-complete for S box topology with D = 2, $|D_1| = W-1$, and $|D_2| = 1$ for any W > 2.

Proof: To prove this theorem, we first show a proof for the special case W = 3, and then show how it is extended to the general case.

Lemma A.1. The problem [MDP on 2-D non-uniform disjoint routing architecture] is NP-complete for S box topology with D = 2, $|D_1| = 2$, and $|D_2| = 1$.

Proof. This problem is certainly in NP. We will show that the NP-complete [non-negated one-in-three 3SAT] problem [7] is polynomial time reducible to the [MDP on 2-D non-uniform disjoint routing architecture] problem. We first state the [non-negated one-in-three 3SAT] problem:

Input: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$, and no $c \in C$ contains a negated literal.

Query: Is there a value assignment for U such that each clause in C has exactly one true literal?

Let the clauses in C be numbered from 1 to |C|, and let the variables in $U = \{x_1, x_2, x_3, ..., x_{|U|}\}$, where |U| is the number of variables in U. We represent each variables x_i by a net x_i and associate each clause $c_i = \{x_j, x_k, x_l\}$ with a diagonally placed C box through which the nets $\{x_j, x_k, x_l\}$ pass. To complete the

global routing all the segments of the same net are connected using vertical routes. There is a solution to the [non-negated one-in-three 3SAT] if and only if there is a solution for the [MDP on 2-D non-uniform disjoint routing architecture] as follows. If a literal x_i is assigned the value true (false) in a valid non-negated one-in-three 3SAT solution, then the net x_i will be assigned to the domain D_2 (D_1) to form a feasible detailed routing for the routing decision problem. Conversely if net x_i is assigned to D_1 (D_2) in a feasible detailed routing then the variables x_i is assigned a value false (true) to satisfy the boolean formula. This works because the routing domain constraints in the routing instance are the same as the variables value assignment. Q.E.D.

In Fig. 6, we show the corresponding global routing instance of the [non-negated one-in-three 3SAT] instance: $(x_1 + x_2 + x_3)$ $(x_2 + x_4 + x_5)$ $(x_1 + x_3 + x_5)$ for the illustration of the above proof.

This lemma can be easily extended to the theorem by noting that the [Non-negated on-in-K KSAT] problem is NP-complete for any fixed $K \geq 3$.

[Non-negated one-in-K KSAT]

Input: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = K$, and no $c \in C$ contains a negated literal.

Query: Is there a value assignment for U such that each clause in C has exactly one true literal?

The case for $K \geq 3$ can be proved inductively by showing a reduction of the [Non-negated one-in-K KSAT] problem to an equivalent [Non-negated one-in-(K+1) (K+1)SAT]. The basic idea of converting an instance C (collection of clauses) of KSAT to that of (K+1) SAT is to introduce a new variable z1 that must be assigned a false value to each clause in C. This is done by adding such a new variable z1 and by adding to C three new clauses $\{y_1, y_2, .. , y_k, z_1\}$, $\{y_1, y_2, .. ,y_k, z_2\}$ and $\{y_1, y_2, .. , y_{k-1}, z_1, z_2\}$ consisting of new variables y_1, y_2, ..., y_k, and z_2. Given K = W, the reduction to our problem [MDP on 2-D non-uniform disjoint routing architecture] with $|D_1| = W-1$ and $|D_2| = 1$ is the same as in Lemma A.1. Q.E.D.

References:

[1] R. Amerson, R. Carter, W. Culbertson, P. Kuekes, and G. Snider, “Teramac - Configurable Custom Computing,” FPGAs for Custom Computing Machines, 1995.

[2] R. Amerson, R. Carter, W. Culbertson, P. Kuekes, and G. Snider, “Plasma: An FPGA for Million Gate Systems,” International Symposium on FPGAs, pp. 10-16, 1996.

[3]S. Brown, R. Francis, J. Rose, and Z. G. Vranesic, “Field Programmable Gate Arrays,” Kluwer Academic Publishers, 1992.

[4] S. Brown, J. Rose, and Z. G. Vranesic, “A Detailed Router for Field-Programmable Gate Arrays”, IEEE Trans. on CAD, V11:5. pp. 620-628, May 1992.

[5] V.C. Chan and D. M. Lewis, “Area-Speed Tradeoffs for Hierarchical Field-Programmable Gate Arrays,” International Symposium on FPGAs, pp. 51-57, 1996.

[6] Y.W. Chang, D.F. Wong, and C.K. Wong, “Universal Switch Modules for FPGA Design,” ACM Trans. on Design Automation, pp. 80-101, Jan. 96.

[7] M. Garey and D. Johnson, “Computers and Intractability - a Guide to the Theory of NP-Completeness,” W.H. Freeman and Company, 1979.

[8] Y. Takashima, A. Takahashi, Y Kajitani, “Detailed-Routability of FPGAs with Extremal Switch-Block Structures,” Euro. Design & Test, pp. 160-164, 1996.

[9] S.G. Williamson, “Combinatorics for Computer Science,” CS Press, 1985.

[10] Y.L. Wu, and D. Chang, “On the NP-completeness of Regular 2-D FPGA Routing Architectures and a Novel Solution,” Proc. Int’l Conf. on CAD, pp. 362-366, 1994.

[11] Y.L. Wu and M. Marek-Sadowska, “Graph Based Analysis of FPGA Routing,” EURO-DAC with EURO-VHDL, pp. 104-109, 1993.

[12] Y.L. Wu, S. Tsukiyama, and M. Marek-Sadowska, “On Computational Complexity of a Detailed Routing Problem in Two-Dimensional FPGAs,” 4th Great Lakes Symp. on VLSI, pp. 70-75, 1994.

[13] Y.L. Wu, S. Tsukiyama, and M. Marek-Sadowska, “Graph Based Analysis of 2-D FPGA Routing,” IEEE Transactions on CAD., pp. 33-44, Jan. 1996.

[14] “The Programmable Logic Data Book,” Xilinx, 1994.

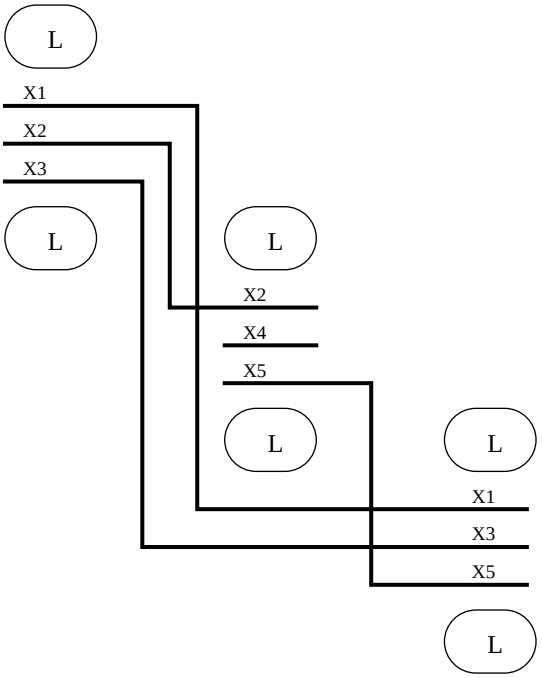

Fig. 6 An example global route (Wg = 3, N = 5) corresponding to the clauses: (X1+X2+X3)(X2+X4+X5)(X1+X3+X5)