General Switch Box Modeling and Optimization for FPGA Routing Architectures

Kejie Ma [#1], Lingli Wang [#2], Xuegong Zhou [#3※], Sheldon X.-D.Tan [*4], Jiarong Tong [#5]

[#] *State Key Lab of ASIC & System, Fudan University, Shanghai, 201203, P.R.China*

[*] *Department of Electrical Engineering, University of California at Riverside, CA, USA*

***Abstract*—This paper explores the FPGA routing architecture based on a new concept of "general switch box (GSB)" to improve the performance of FPGA. Compared with the existing CB/SB routing architecture and CS-box architecture, the proposed GSB architecture has much larger exploration space. Experimental results with MCNC benchmark circuits show that the performance of FPGAs with GSB is about 24.3% better than the CB/SB architecture with the same segment distribution in terms of product of channel width and delay using 0.17% less routing switches for the single wire length. For the two types of wire segments, we propose an architecture with 13.3% performance improvement at the cost of about 0.8% increase in switch number compared to the single wire length GSB architecture.**

I. INTRODUCTION

The demand for Field Programmable Gate Arrays (FPGAs) keeps increasing because of better flexibility, lower amortized non-recurring engineering cost and shorter time to market. Some novel FPGA architectures have been proposed. The design of FPGA interconnects is critical as more than 70% of an FPGA area is dedicated to routing resources.

For the routing architecture, academic research is largely based on the framework using connection block (CB) and switch block (SB) [1][2]. An input or output of a logic block (LB), called pin, can connect to some or all of the adjacent wiring segments via a programmable switch in CB. A SB is simply a set of programmable switches that connects wires in adjacent channels through programmable switches, as shown in Fig. 1(a). Reference [3] proposed a new FPGA architecture using the Connection-Switch Box (CS-Box) by combining the CB and SB. By using the CS-Box, number of switches can be reduced at the cost of increasing the channel widths and circuit delay. Commercial FPGA devices choose better routing architectures such as GRM (general routing matrix) in Xilinx Virtex-5 family to improve the performance [4]. Hence, it is important to study and evaluate modern FPGA routing architectures to further improve FPGA performance.

II. GSB MODELING

It was shown that about 80% of the critical path is inter-LB routing delay [1]. In CB/SB architecture, the critical path typically starts from an LB output, passes through routing switches in CBs and SBs, and terminates at an input of another LB. The CS-Box structure can reduce the number of routing switches. However, only one CS-Box interconnect pattern for both logic pins and pad pins is provided, and such a pattern may not be optimal.

A. Fast Path Connection

With the progress of process technology, it will be wiser to improve the FPGA performance by increasing the routing flexibility than by reducing the number of routing switches as described in [3]. Consequently, "fast path" routings should be considered, as they can connect adjacent LBs directly by less routing switches. We propose a new routing architecture with general switch box (GSB). As shown in Fig. 1 (b), pins of different LBs can be connected with each other through the GSB architecture. GSB is a set of programmable switches that allow some of the wire segments or pins of LB incident to the GSB to be connected with others.

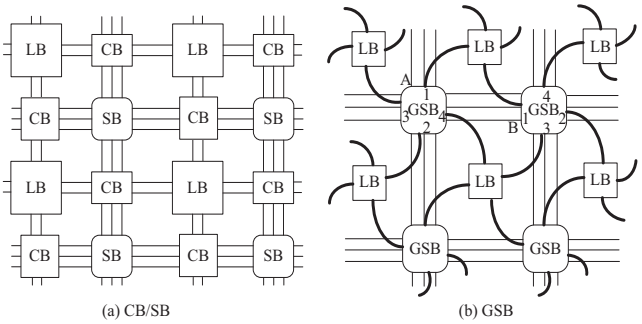


Fig. 1 Comparison of FPGA architectures

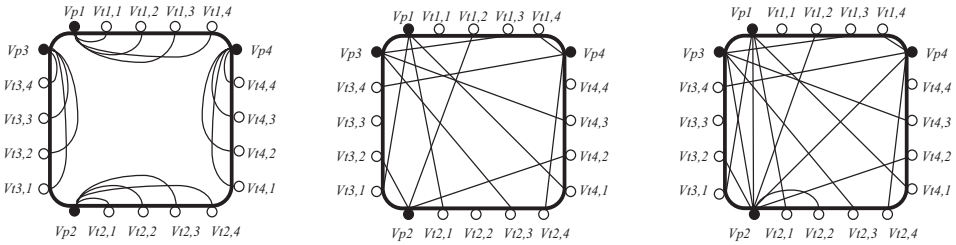
B. Definition of GSB Model

The GSB routing architecture is defined as follows:

1) *Definition 1:* GSB routing architecture is defined as a directed acyclic graph, $G = (V, E)$, where V and E are a routing node set and an edge set respectively. Similar to the concept of the routing resource graph in the tool of VPR (Versatile Place and Route) [5], both logic pins and routing segments are the members of V, while routing switches are the members of E. If there are n logic pins for a logic block and W routing tracks in a channel, then each logic pin can be represented by a routing node V_{pi}, while $V_{tj,\,k}$ represents the kth track on the jth side of a GSB, $1 \le i \le n$, $1 \le j \le 4$, $1 \le k \le W$.

2) Definition 2: An index number is defined for each side of a GSB. The side, which a routing node belongs to, is always defined as the 1st side, whereas its opposite, right and

978-1-4244-8983-1/10/$26.00 ©2010 IEEE



(a) CB/SB (b) CS-Box (c) GSB

Fig. 2 Comparison of three routing architecture

left sides are indexed as 2nd, 3rd, and 4th sides respectively. For convenience, if we do not specify connections for a specific routing node, then the top side of a GSB is indexed as the 1st side by default.

Fig. 1(b) shows side index numbers of two GSBs where routing nodes "A" and "B" belongs to the top side and left side respectively. Hence, side index numbers of a GSB vary with the position of a routing node.

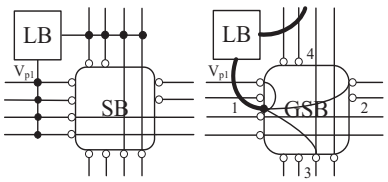
Based on the above definitions, it can be seen that GSB structure is more general than CB/SB and CS-box. Consequently, we have much larger design space to determine better FPGA routing architectures. For example, Fig. 2 (a) shows the CB/SB routing architecture in VPR using the GSB representation, where n = 4 and W = 4. Interconnections between segments, which are denoted by vacant circles, are not shown in Fig. 2 as they are well defined, such as Universal [6], HUSB [7], Disjoint[8], and Wilton[9] architectures.

In Fig. 2 (a), 4 logic pins are distributed evenly on GSB perimeters, shown as 4 filled circles. Fig. 2 (b) shows the CS-Box, where a logic pin can connect with the segments on the sides except the one it belongs to. For example, a logic pin node V_{p2} can connect with 3 segment nodes, $V_{t1,2}$, $V_{t3,2}$, and $V_{t4,2}$. However, the "fast path" is not available in the CS-Box, but they can be added and evaluated in the GSB. In Fig. 2 (c), suppose V_{p2} is an output pin and it can connect to 3 input pins directly, i.e. V_{p1}, V_{p3}, and V_{p4}. Further, in Fig. 2 (c), V_{p2} can connect to $V_{t2,2}$ directly. Hence, CB/SB routing architecture and CS-Box are two special cases of the GSB representation. As a result, the enhanced flexibility of GSB architectures will help to improve not only the routability, but also the performance.

C. GSB Modeling Parameters

GSB architectures can be modeled using the following parameters:

1) F_{c_input}, F_{c_output} and F_{c_pad}: Each is specified as a 4-tuple, ($f_{c1}, f_{c2}, f_{c3}, f_{c4}$), where f_{ci} is the fraction of segments on the ith side of the GSB that an input (output, or IO pad) pin can connect with, $0 \le f_{ci} \le 1, 1 \le i \le 4$. The side indexing for each input (output, or IO pad) pin shown in Fig. 3 (b) is based on definition 2 in section B. Notice that there is no restriction that the sum of those 4 ratio numbers should be 1. For instance, there are some of wire segments passing through the GSB shown in Fig. 3. To achieve equal routability of CB/SB in Fig. 3 (a), the average of f_{ci} should be more than 0.25. In Fig. 3 (a) F_c is 1 in CB/SB architecture which means V_{p1} can connect to 4 wire segments in horizontal channel; however, if F_{c_input} = (0.25, 0.25, 0.25, 0.25) in GSB architecture, V_{p1} can connect to 3 wire segments only because the wire segments which V_{p1} connects to in the 3th side and the 4th side are identical.



(a) CB/SB (b) GSB

Fig. 3 Connections when wire segments passing through SB and GSB

2) $F_{c\_output\_with\_lb\_input}$: This parameter specifies the flexibility of the "fast path" connections, which is the fraction of all the adjacent logic block inputs that an output of the logic block can drive. For example, "$F_{c\_output\_with\_lb\_input}$ = 1" means that any output of a logic block can drive all the inputs of the adjacent logic blocks.

3) $F_{c\_output\_with\_pad\_input}$: Similar to $F_{c\_output\_with\_lb\_input}$, it specifies the fraction of adjacent IO pads that an output of the logic block can drive.

III. GSB Exploration

In general, GSB architecture brings great benefits to FPGA routing. In CB/SB architecture, a pin-to-pin net sends a signal through at least two routing switches, since each pin can connect to wire segments at least through a routing switch. In GSB architecture, if this net connects pins of two adjacent logic blocks, only one routing switch is needed and no segments will be consumed in this connection. Fig. 4 shows the difference.

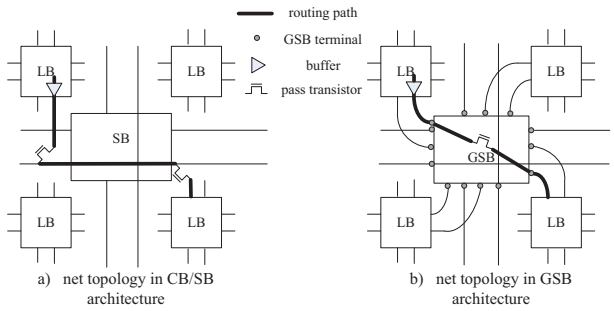


Fig. 4 Pin to pin net topologies in CB/SB and GSB architecture

We can compute two net delays in Fig. 4 by Elmore delay model [1]. Assume that each segment has a metal capacitance C_{seg} and wire connected to an input or output has a metal capacitance C_{pin}; each pass transistor has a parasitic switch capacitance C_{pass}; T_{obuf} and R_{obuf} denote the intrinsic delay and equivalent resistance of the logic block output pin buffer; R_{pass} is the equivalent resistance of each pass transistor.

$$T_{CB/SB} = T_{obuf} + R_{obuf}(C_{seg} + 2C_{pin,CB/SB} + 2C_{pass}) + R_{pass} \times (C_{seg} + C_{pin,CB/SB} + C_{pass}) + R_{pass} \times C_{pin,CB/SB}$$

$$T_{GSB} = T_{obuf} + R_{obuf}(C_{pass} + 2C_{pin,GSB}) + R_{pass} \times C_{pin,GSB}$$

$$T_{CB/SB} - T_{GSB} = R_{obuf}(C_{seg} + 2C_{pin,CB/SB} + C_{pass} - 2C_{pin,GSB}) + R_{pass}(C_{seg} + 2C_{pin,CB/SB} + C_{pass} - C_{pin,GSB})$$

Suppose that $C_{pin,GSB} = 2C_{pin,CB/SB}$, $C_{pin,CB/SB} = 0.1C_{seg}$, and it is easy to learn that $T_{CB/SB} > T_{GSB}$. Therefore path delay in GSB routing architecture can be less than in CB/SB architecture.

The reduction of switches in routing will improve circuit speed. Meanwhile, the routability will be improved since the segment consumption is reduced compared with the routing results of the same circuit in CB/SB architecture. In GSB architecture, pins can connect to segments on four sides. As it is shown in Fig. 4, if a net connects two pins of distant logic blocks, CB/SB architecture has to use more tracks to change routing direction in SB, while GSB architecture can directly connect pins to segments on different sides of GSB to implement this routing. This improvement enhances FPGA routability and reduces the number of routing switches greatly. More unoccupied tracks can be used for circuit implementation.

IV. Experimental Results

To evaluate various GSB architectures efficiently, we perform the following steps:

1) Decide a good distribution of 4 fraction numbers in F_{c_input}, F_{c_output} and F_{c_pad} with wire length 1.
2) Based on the distribution obtained in step 1), we then determine the fraction of "fast path", i.e. value of $F_{c\_output\_with\_lb\_input}$ and $F_{c\_output\_with\_pad\_input}$.
3) Adjust wire length to find the best single wire length based on step 1) and 2).
4) Evaluate more complex architectures that contain two types of wire segments.

Based on the VPR tool, we have added the support for GSB exploration. In our experiments, the iteration number of routing is 30 and the routing algorithm is timing-driven algorithm. Wilton switch topology is applied to make the connections between segments. The experimental results are listed as follows.

A. F_c Distribution with Wire Length 1

TABLE I shows the results for 3 architectures with the following settings: length of all wire segments is 1; F_{c_input} and F_{c_output} are equal in every architecture; F_{c_pad} = (1, 0, 0, 0); $F_{c\_output\_with\_lb\_input}$ and $F_{c\_output\_with\_pad\_input}$ are 0. We adopt the average value of running 20 MCNC largest benchmark circuits [1] and each average value is compared with the corresponding value on CB/SB architecture using the following formulas to obtain the delta values,

$$\Delta SN = \frac{SN - SN_{CB/SB}}{SN_{CB/SB}} \times 100\%$$

$$\Delta(CW \times CP) = \frac{(CW \times CP) - (CW \times CP)_{CB/SB}}{(CW \times CP)_{CB/SB}} \times 100\%,$$

where "CW" stands for the channel width to route circuits, "CP" for the critical path delay, and "SN" for the number of routing switches. The result is summarized in TABLE I.

Comparing with CB/SB and CS-Box architectures, as shown in TABLE I, the number of switches in GSB architectures increases marginally because of the reduced channel width. In addition, an even distribution of 4 fraction numbers in F_{c_input} and F_{c_output} leads to better performance by considering the product of delay and area: reduction in CW×CP by 7.09% and only few increases in SN by 0.66%.Therefore we chose the best base architecture GSB2. And then analyse the value of $F_{c\_output\_with\_lb\_input}$ and $F_{c\_output\_with\_pad\_input}$.

B. Single Wire Length

In CB/SB routing architecture, length 4 wires achieve the best combination of low delay and high area-efficiency [1] instead of length 1 wires. Consequently, based on an even distribution of F_{c_input} and F_{c_output}, we try to find out a better architecture with settings as follows:

1) there is only one type of wire length in each architecture;
2) F_{c_pad} = (1, 0, 0, 0), $F_{c\_output\_with\_lb\_input}$ = 0.5 and $F_{c\_output\_with\_pad\_input}$ = 0;
3) Each type of wire length tries $F_{c\_input} = F_{c\_output}$ = (0.3, 0.3, 0.3, 0.3), (0.35, 0.35, 0.35, 0.35), (0.4, 0.4, 0.4, 0.4), (0.45, 0.45, 0.45, 0.45), (0.5, 0.5, 0.5, 0.5).

TABLE II illustrates that GSB7 has the best performance and length 4 could be a better choice either in CB/SB or in GSB architecture than other wire length.

C. Two Types of Wire Segment Architecture

Firstly, we compare all results in CB/SB architecture with different combinations of two types of wire lengths, and learn that the architecture with 50% length 1 wires and 50% length 8 wires has the best performance.

Secondly, we try to find out the best GSB architecture with settings as follows: there are two types of wire length in each architecture and every type of wire segment takes 50%; other settings are the same as section IV- B.

TABLE III shows that GSB14 architecture achieves a reduction of 16.7% compared to the best two types of wire segment CB/SB architecture, and only a 7.34% increase in number of switches. While compared to the GSB7 with the single wire length architecture, the performance is 13.3% better with about 0.8% higher switch number.

TABLE I
COMPARISON OF ARCHITECTURES WITH WIRE LENGTH 1
CW: channel width CP: critical path delay SN: number of routing switches

Arch	$F_{c\_input}(F_{c\_output})$	CW	CP(10^{-7}s)	SN	Δ(CW×CP)	Δ(SN)
CB/SB	(1, 0, 0, 0)	9.8	1.53	555393	0	0
CS-Box	(0, 0.25, 0.25, 0.25)	10.1	1.51	542895	14.5%	-2.25%
GSB1	(0, 0.35, 0.35, 0.35)	9.45	1.47	568396	-7.29%	2.34%
GSB2	(0.25, 0.25, 0.25, 0.25)	9.5	1.47	559091	-7.09%	0.66%
GSB3	(0.7, 0.1, 0.1, 0.1)	9.45	1.48	565582	-6.73%	1.83%
GSB4	(0.1, 0.3, 0.3, 0.3)	9.35	1.52	564166	-7.88%	1.58%

TABLE II
COMPARISON OF ARCHITECTURE WITH SINGLE WIRE LENGTH

Arch	Wire Length	$F_{c\_input}(F_{c\_output})$	CW	CP(10^{-8}s)	SN	Δ(CW×CP)	Δ(SN)
CB/SB	4	(1, 0, 0, 0)	13.65	8.07	576917	0	0
GSB6	2	(0.45, 0.45, 0.45, 0.45)	9.35	9.85	577012	-16.5%	0.02%
GSB7	4	(0.5, 0.5, 0.5, 0.5)	10.95	7.62	575956	-24.3%	-0.17%
GSB8	6	(0.45, 0.45, 0.45, 0.45)	13.4	6.86	627253	-16.7%	8.73%
GSB9	8	(0.45, 0.45, 0.45, 0.45)	15.3	6.83	691960	-5.12%	19.9%
GSB10	10	(0.4, 0.4, 0.4, 0.4)	17.8	6.91	741464	11.6%%	28.5%

TABLE III
COMPARISON OF ARCHITECTURE WITH TWO TYPES OF WIRE SEGMENTS

Arch	Wire Length	$F_{c\_input}(F_{c\_output})$	CW	CP(10^{-8}s)	SN	Δ(CW×CP)	Δ(SN)
CB/SB	1,8	(1, 0, 0, 0)	11.3	7.69	541075	0	0
GSB11	1,8	(0.4, 0.4, 0.4, 0.4)	9.9	8.01	615179	-8.69%	13.7%
GSB12	1,6	(0.4, 0.4, 0.4, 0.4)	9.65	8.20	608063	-8.84%	12.4%
GSB13	1,4	(0.45, 0.45, 0.45, 0.45)	9.15	8.19	625356	-13.7%	15.6%
GSB14	2,8	(0.45, 0.45, 0.45, 0.45)	10.5	6.89	580805	-16.7%	7.34%
GSB15	2,6	(0.45, 0.45, 0.45, 0.45)	10.4	7.03	575767	-15.8%	6.41%

V. CONCLUSION

In this paper, we have proposed a new FPGA routing architecture based on novel general switching box for design space exploration. We compared the routing models in GSB and CB/SB architecture respectively. Experimental results show that the FPGA performance, in terms of the product of channel width and critical path delay, is 24.3% better than the CB/SB architectures on average with 0.17% less routing switches under the same segment distribution in the case of single wire length. In addition, we obtain a better GSB routing architecture with the parameters $F_{c\_input} = F_{c\_output}$ = (0.45, 0.45, 0.45, 0.45); F_{c_pad} = (1, 0, 0, 0), $F_{c\_output\_with\_lb\_input}$ = 0.5; $F_{c\_output\_with\_pad\_input}$ = 0 in the case of two types of wire segment architecture with 50% length 2 wires and 50% length 8 wires. Our methodology can be used to generate different FPGA routing architectures easily to evaluate the performance of FPGAs.

ACKNOWLEDGMENT

This paper is supported by National 863 Program of China(2009AA012201), Shanghai Pujiang Program 2008，and Science and Technology Commission of Shanghai Municipality Program 08706200101.

※Communication author.

REFERENCES

[1] V. Betz, J. Rose, and A. Marquardt, Architecture and CAD for Deep-Submicron FPGAs. Boston, Kluwer Academic Publishers, 1999, 51-103
[2] Eachempati, S. Nieuwoudt and A. Gayasen, Assessing Carbon Nanotube Bundle Interconnect for Future FPGA Architectures, Design, Automation & Test in Europe(DATE) Conference, 2007, 23 -32
[3] Zhou , C. L., Cheung, R., and Wu, Y.-L. What if Merging Connection and Switch Boxes -- an Experimental Revisit on FPGA Architectures, IEEE International Conference on Communications, Circuits and Systems, 2004, 1295 -1299
[4] Vertex-5Q Family Overview (v2.0), Xilinx , March 22,2010
[5] V. Betz, T. Campbell, and Fang, Wei Mark, VPR and T-VPack User's Manual (Version 5.0), 2008
[6] M.Shyu, G.M.Wu, Y.D. Chang et al, Generic Universal Switch Blocks, IEEE Trans. On Computers, 2000,348-359
[7] H.Fan, J.Liu, Y.L.WU et al, General Models and a Reduction Design Technique for FPGA Switch Box Designs. IEEE Trans. On Computers,2003, 21-30
[8] G. G. Lemieux and S. D. Brown, A Detailed Router for Allocating Wire Segments in Field-Programmable Gate Arrays, Proceedings of the ACM/SIGDA Physical Design Workshop, pp. 215--226 , Lake Arrowhead, CA , April 1993
[9] S. Wilton, Architecture and Algorithms for Field-Programmable Gate Arrays with Embedded Memory, Ph.D. thesis, University of Toronto, 1997
[10] S.Yang, Logic Synthesis and Optimization Benchmarks, Version 3.0, Tech.Report, Microelectronics Center of North Carolina,1991