ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.8

Impact of Segment Types on Critical Path Delay Characteristics in FPGA Routing

Usha Maddipati[1], Swami Naidu Gummadi[2], Rama Vara Prasad Reddy[3]
M.Tech Student[1], Assistant Professor[2], Associate Professor[3]
Department of Electronics & Communication Engineering
Sasi Institute of Technology & Engineering,Tadepalligudem, India

Abstract:
Hierarchical routing resources play vital role in FPGA routing. Better routability options can be obtained using segmented approach of wires thus enabling routing optimization. Source and sink logic blocks can be connected via wire segments such that the overall wire length and switching transistors inside the switch box can be saved over an extent. This paper presents an experimental approach of study of wire segments types affecting the routability. Widely used segment types are considered for experiments. A routing algorithm containing the cost equation based on routing segment types is proposed. Results show that at least 15% less critical path delay can be achieved tested against 12 largest MCNC benchmark circuits.

Keywords: Hierarchical routing segments; FPGA; Routing algorithm; Wire segments

I. INTRODUCTION

FPGA has its flexibility in routing mechanism mainly due to hierarchical routing resources in which each track can be represented by a group of segments. Such kind of architecture allows the designer to effectively place and route the design as close as possible targeting better delay and area characteristics. Study of such segment dependency can be helpful in understanding the routing strategy while implementing largest and complex logic. Higher number of switches is needed when routing is done with short wire segments thus leading to increased delay. Delay can be improved using longer wire segments but with more area and power as it requires more routing tracks which may results larger capacitances. Therefore, efficient distribution of different types of wire segments is necessary for better delay characteristics. Few earlier works exist for segment-oriented routing approach. The detailed routing algorithm used in [1] with segmented routing channels used less number of wires per channel by maintaining balance between segments usage among long and short connections. A fast routability-driven router for FPGAs targeting short routing times is presented in [2]. Authors in [3] proposed hardware circuit for efficient routing. Using 10% additional routing resources, results showed that there is 10% improvement in the speed when compared to VPR's routability-driven router. Authors in [4] proposed a stochastic approach which uses the tool for optimizing routing architecture (TORCH) software based on the VPR tool. TORCH can be used for optimal use of routing channel segments and switch box patterns based on the architecture parameters of FPGA tested on a set of benchmark designs using average interconnect power-delay product as performance metric. A routability-driven routing algorithm for Hierarchical FPGAs without bend cost factor is presented in [5] and is tested on various MCNC bench mark circuits. Enhancement in speed is achieved by prohibiting nodes so that search space for afterward nets becomes smaller. Authors in [6] investigated the speed performance based on the distribution of segment length in island-style FPGAs. They have presented an approach for determining segment length optimal distribution. They have concentrated only on distribution in which each segment type is assigned with a percentage of total available segment types and the routing algorithm cost function is independent of segments types. A novel FPGA routing method is proposed in [7]. Here the signals are assigned to wire segments groups and compared against assigning to individual wire segments. Although they have achieved better CAD runtime configurations, but critical path delay numbers have not changed. A genetic algorithm approach is proposed in [8] for segmented channel routing. The main objective of the work is to find a congestion-free assignment of nets in the channel tracks having minimum routing cost. Cost function is again independent of segment types. Several other works considered segments as default resources in their respective performance-aware methodologies but no priority is applied for them. In this paper, the main contributions are: (i) analyzing MCNC benchmark circuits [13] on VPR [14] targeting variety of routing segments included in the architecture and their impact on critical path delay. Combination of one or more types of wire segments will be included in the architecture for the study. (ii) Defining a new cost model for VPR routability-driven routing algorithm and study the effect on critical path delay.

II. WIRE SEGMENTS AS ROUTING RESOURCES

In general, FPGAs consist of IO pads, logic and routing resources which is widely represented as island style architecture. Basically it is a combination of logic block (LB), switch block (SB), connection block (CB) and a routing channel. The combination of SB, CB and wire segments normally called as routing resources and provides connectivity using switches which are programmable. Island style architecture contains the routing resources in a mesh-like structure having horizontal and vertical routing channels as shown Fig. 1. These are connected by SBs having flexibility of routing connections. The LBS are connected to channels using CBs. A connection between source and sink LBs can be made using programmable switches in CBs and SBs. As shown in Fig. 1, each LB is connected to channel via CB and can be extended with the help of SB. Wire length is denoted

using LB hops on a FPGA architecture plane. Distance will be increased by one hop whenever a signal passes by a logic block. Wires can be defined into two major types: segmented and non-segmented wires. Former one contains single length (SL), double length (DL) and hex length (HL) wire types and later contains long line (LL) wire types. Segment can be represented by a metal wire which spans over a specific number of LBs and terminate at switch blocks. An example connection between a source LB and a sink LB routed by SL segments is shown in Fig. 2. Interconnection grid is formed by intersection of SL segments at each switch block. DL segments are twice in length compared to SL and passes two LBs before entering a switch block. In similar way, HL passes six logic blocks. Fig. 3 shows an example connection with HL and DL wire types. LL wires form a connection grid passing horizontally/vertically through avoiding all switch blocks which covers entire length/width of the logic blocks array. Long distance net connections can be routed using LL wire types. Each segment type share for any routing channel is decided while defining the FPGA architecture. An example share (6 LLs, 30 HLs, 10 DLs and 4 SLs) in a horizontal routing channel is shown in Fig. 4.

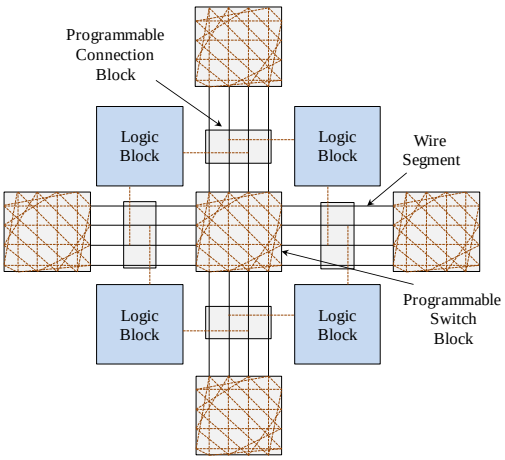

Figure.1.Programmable Connection between Lbs Involving SBS AND CBS

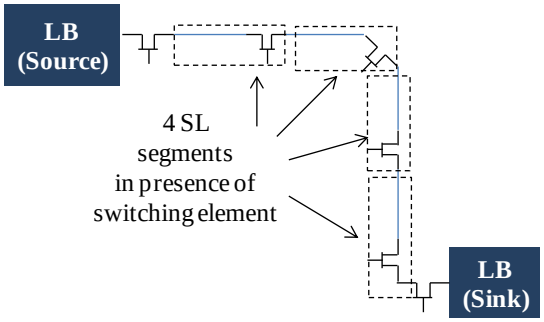

Figure.2. Connection between source and destination lbs using pass transistor

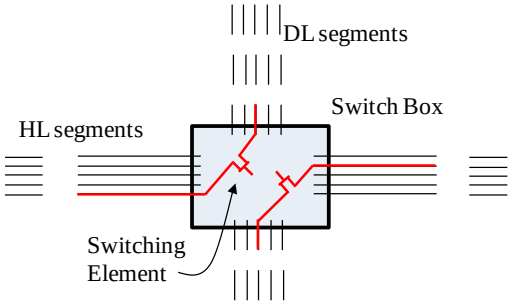

Figure.3. Establishing a connection using different segment types

III. COST FUNCTION BASED ON WIRE SEGMENTS FOR VPR ROUTING

After placement, connections between the LBs are assigned by routing phase. The typical target for a routing phase is optimized critical path delay and also avoiding congestions with in wiring resources. Routing will be done in two stages: global and detailed routing. Global routing will be done based on the circuit without considering the available number and type of wire segments. Whereas in detailed routing connections will be assigned to specific type of wires based on the best delay characteristics. Number of switches need to be turned-on will be determined by the routing phase.

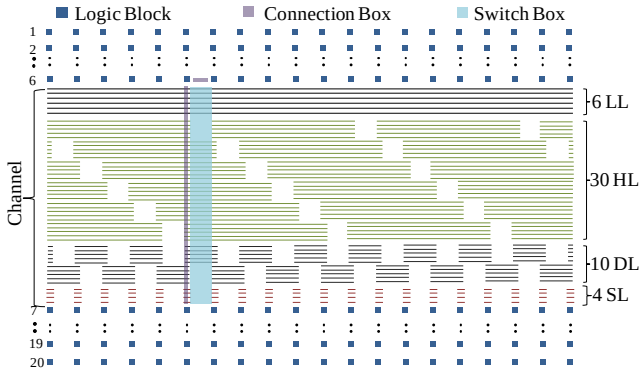

Figure. 4.Typical Horizontal Routing Channel in FPGA [9]

Every routing iteration will rip-up and re-route each circuit net. The foremost iteration in routing targets for less delay which may results to congestion and resources overuse but the problem will be balanced and solved in the next following iterations. Each segment type share in horizontal/vertical routing channel is decided while defining the FPGA architecture. For example, as shown in Fig. 4, the total number of available SL segments in an FPGA of size of 20 x 20 and a routing channel width of 50 tracks for each layer can be obtained as below. The total number of available SL segments in an FPGA can be obtained as shown in Table I. Each layer consists of 19 channels in X-direction, 19 channels in Y-direction and 4 separate channels for each side of the IO banks. Table II gives a typical share of all four types of routing segments in a FPGA with an array size of 20 x 20 and a routing channel width of 50.

Table.1. Number of available sl segments in a given fpga

Parameter	Value
FPGA Array Size	20x20
Routing Channel Width	50
Number of SL tracks/channel	4
Number of SL segment/track	20
Total number of channel for given FPGA	19+19+4 =42
Total number of SL segments for the given FPGA	4*20*42 =3360

Table.2. Routing segments types share of 20 x 20 array size fpga having channel width of 50

Segment Type	Length of segment (in terms of LB hops	Max available tracks/channel	Total number of segments
Single-Length (SL)	1	4	3360
Double-Length (DL)	2	10	4200
Hex-Length (HL)	6	30	3780
Long-Length (LL)	Long	6	252

VPR is an open-source place and route tool intended to architecture and CAD research for island-style FPGA architectures [4]. Such architectures contain programming routing resources along with logic and I/O blocks. Fig. 5 illustrates the implementation process using VPR. It takes a particular logic level design and architecture file along with required specifications of the design which later implements on the given architecture. The output generates all the required parameters starting from the number of interconnect occupied by the nets in circuit to the critical path delay of the implemented circuit. VPR also provides the circuit area in given architecture. Synthesis and technology mapping is done in the initial stage of the implementation. Technology mapped file will be packed into logic clusters specified for the given architecture. VPR will then performs circuit placement and routing. Circuit performance will be determined using timing analysis. VPR uses a routability-driven cost function targeting congestion with minimum value. To achieve best timing numbers the existing cost function is modified with respect to the proposed segment-type parameters.

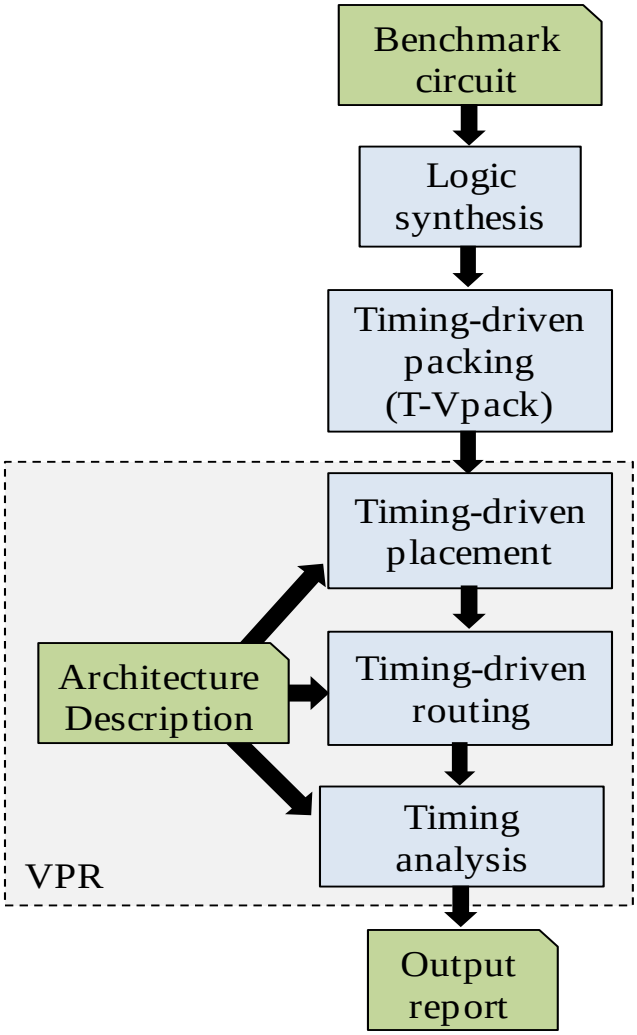

Figure. 5. VPR cad flow for 2d FPGAS

The primary objective of VPR routability-driven algorithm is routing a design successfully with minimum track count and contains a cost function as below:

$$cost_n = bc_n * h_n * p_n + bend_{n,m} \quad (1)$$

Where bc_n is the base cost; h_n is the historical congestion penalty to keep the history of resources earlier cost and stops reusing a channel in later iterations; p_n is the present congestion penalty and its value is the difference between the number of nets using a channel and the number of wires that can be placed on that channel; and $bend_{n,m}$ penalizes the wire when it bends while routing. Percentage delay difference between combinations of variety wire segment types without using proposed cost function Wire segment parameter is included in the overall cost deriving the following function:

$$cost_n = bc_n * h_n * p_n + bend_{n,m} + sgf/wsg_{n,m} \quad (2)$$

Where *sgf* is constant derived from the regression based statistical analysis of benchmark circuits running over variety and combination of segment types and $wsg_{n,m}$ is the cost involved in choosing a variety of wire segments over a particular iteration. A variety of combination of wire segments types effects the parameter $wsg_{n,m}$. From the experiments, we found that large amount of critical path delay can be reduced having a penalty of CAD run-time.

IV. EXPERIMENTAL RESULTS

The proposed algorithm is tested using VPR. The in-built routability-driven routing algorithm is replaced with the proposed segment-oriented routing algorithm. Experiments are carried out on 12 largest MCNC benchmark circuits as shown in Table III. Although the delay characteristics depend on technology parameters as well, we have assumed intuitively the existing 180nm node in VPR. T-V PACK is used to generate VPR style .net file for each benchmark circuit. Then the circuit is placed and routed on 4-LUT based sanitized architecture. We have considered segment types 1, 2, 6 and long line. Study is carried out by running each circuit for several cases like: architecture with one type of segments, two types of segments, three types of segments and four types of segments over all the segment types. Such kind approach is indeed necessary to find the sole effect of individual type on its counterpart. The initial such experiments without cost function based on routing segments motivated us to derive the proposed algorithm based on cost function considering routing segments. Although not all the benchmark circuits responded equally, but majority of the circuits showed a quite good response with optimized delay numbers. Based on the study the proposed cost function is developed and tested.

Table.3.`MCNC benchmark circuit properties

Benchmark	LBs	Nets	Total nets including multi-terminal nets	Input pins	Output pins
alu4	1522	1536	5408	14	8
apex	1878	1916	6692	38	3
bigkey	1707	1936	6313	300	197
clma	8383	8445	30462	63	82
des	1591	1847	6110	256	245
diffeq	1497	1561	5296	65	39
elliptic	3604	3735	12634	132	114
frisc	3556	3576	12772	21	116
pdc	4575	4591	17193	16	40
s298	1931	1935	6951	5	6
s38417	6406	6435	21344	30	106
s38584.1	6447	6485	20840	39	304
seq	1750	1791	6193	41	35
spla	3690	3706	13808	16	46
Average	3467	3535	12287	74	96

Table IV gives the delay numbers without using the segment-based cost function. All the benchmark circuits are ran on a sanitized architecture beginning with only one type wire segment i.e. SL segments. Then a combination of SL-DL followed by SL-DL-HL and SL-DL-HL-LL wire segments types are chosen for circuit implementation. Table V shows the

delay numbers after applying the proposed cost function to the routing algorithm.

Table.4. Percentage delay difference between combinations of variety wire segment types without using proposed cost function

Circuit	Delay (SL only) T1	Delay (SL & DL only) T2	Delay (SL, DL & HL only) T3	Delay (SL, DL, HL and LL) T4	% Delay difference		
Segment Frequency	SL=1	SL=0.4, DL=0.6	SL=0.3, DL=0.4, HL=0.3	SL=0.2, DL=0.3, HL=0.4, LL=0.2	between T2 and T1	between T3 and T1	between T4 and T1
alu4	2.366	1.725	1.606	1.607	-27.09	-32.12	-32.08
apex2	3.042	2.008	1.987	2.171	-33.99	-34.68	-28.63
apex4	2.577	1.852	1.731	1.705	-28.13	-32.83	-33.84
bigkey	2.101	1.702	1.069	0.918	-18.99	-49.12	-56.31
clma	1.954	1.348	1.294	1.095	-31.01	-33.78	-43.96
des	2.948	2.111	1.419	1.496	-28.39	-51.87	-49.25
diffeq	2.162	1.72	1.688	1.685	-20.44	-21.92	-22.06
dsip	2.188	1.353	0.984	0.955	-38.16	-55.03	-56.35
ex5p	2.267	1.744	1.599	1.866	-23.07	-29.47	-17.69
misex3	2.553	1.677	1.501	1.59	-34.31	-41.21	-37.72
seq	2.786	1.694	1.567	1.626	-39.20	-43.75	-41.64
tseng	1.813	1.641	1.507	1.58	-9.49	-16.88	-12.85
				Average	-27.69	-36.89	-36.03

Table.5. Percentage delay difference between combinations of variety wire segment types using proposed cost function and overall difference between without/with proposed cost function

Circuit	% Delay difference			Difference against without and with proposed cost function		
	between T2 and T1	between T3 and T1	between T4 and T1	T2 and T1	T3 and T1	T4 and T1
alu4	-18.61	-29.16	-30.26	-31.31	-9.22	-5.67
apex2	-21.78	-31.26	-27.06	-35.92	-9.86	-5.49
apex4	-21.02	-27.44	-31.87	-25.28	-16.42	-5.82
bigkey	-10.23	-46.42	-54.14	-46.13	-5.50	-3.85
clma	-20.16	-31.07	-41.61	-35.00	-8.01	-5.35
des	-23.89	-48.33	-46.11	-15.86	-6.82	-6.38
diffeq	-11.46	-18.99	-19.40	-43.94	-13.38	-12.07
dsip	-26.17	-52.06	-55.45	-31.43	-5.39	-1.60
ex5p	-18.20	-26.57	-15.98	-21.11	-9.83	-9.66
misex3	-24.65	-40.36	-34.21	-28.16	-2.05	-9.31
seq	-29.10	-42.36	-39.06	-25.76	-3.19	-6.19
tseng	-7.55	-14.22	-10.36	-20.42	-15.75	-19.39
Average	-19.40	-34.02	-33.79	-30.03	-8.79	-7.56

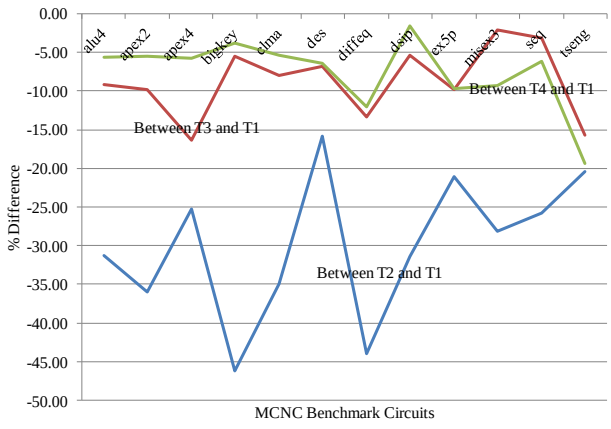

Figure.6. Percentage delay difference between without/with proposed cost function

Table V shows the delay numbers obtained using proposed segment-based cost function. This table also gives the percentage reduction between without and with proposed cost function cases. An illustrative representation is shown in Fig. 6. It is clearly evident from the table that the delay is reduced on an average of 30% for SL to SL-DL, 8% for SL to SL-DL-HL and 7% for SL to SL-DL-HL-LL combinations. It is quite noticeable that the percentage reduction is more for SL to SL-DL and less for other two cases due to the fact that double length segment involvement is more for any particular circuit implementation. It may gives better routability and less number of programming switches. It is identified that on average over 12 MCNC benchmark circuits, around 15% of critical path delay can be reduced.

V. CONCLUSION

Routing segments are having vital role in modern FPGAs. Highlighting such role using proper experimental methodologies will be helpful for developing efficient routing techniques. A novel cost function for FPGA routing based on wire segments is proposed in this work. It is observed from the initial experiments that segment type variation will have significant impact on overall routability as well as the critical path delay. Cost function is examined against the initial experiments and regression based statistical analysis. It is found that at least 15% of critical path delay can be reduced using the proposed work calculated on average over 12 largest MCNC benchmark circuits.

VI. REFERENCES

[1].G. Lemieux, S. Brown, “A Detailed Router for Allocating Wire Segments in FPGAs”, ACMISIGDA Physical Design Workshop, 1993, pp. 215–226.

[2].J. Swartz, V. Betz, and J. Rose, “A Fast Routability-Driven Router for FPGAs”, In 6th International Workshop on Field-Programmable Gate Arrays, Monterey, Ca, Feb. 1998.

[3].Lysecky, R., Vahid, F., & Tan, S. X. D., “ Dynamic FPGA routing for just-in-time FPGA compilation”, In Proceedings - Design Automation Conference , 2004, pp. 954-959.

[4].I. Kuon, R. Tessier and J.Rose, “FPGA Architecture: Survey and Challenges,” Foundations and Trends in Electronic Design Automation, Vol. 2, No. 2, 2007, pp. 180-181.

[5].Xingxing Zhang, Qiang Zhou and Yici Cai, "A fast routability-driven router for Hierarchical FPGAs based on Tabu search," 2009 International Conference on Communications, Circuits and Systems, Milpitas, CA, 2009, pp. 1052-1056.

[6].Ashish Mishra, Nidhi Jayapalan, Harsha Rastogi and Tushar Agrawal, “Impact of Segmentation Distribution on Area and Delay in FPGA Routing Architectures”, in IEEE 3rd International Advance Computing Conference (IACC), 2013.

[7].Marcel Gort and Jason H. Anderson, “Combined Architecture/Algorithm Approach to Fast FPGA Routing”, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 6, June 2013.

[8].Lipo Wang, Lei Zhou and Wen Liu, "FPGA Segmented Channel Routing Using Genetic Algorithms", in IEEE Congress on Evolutionary Computation, 2005.

[9].Krishna Chaitanya Nunna, Farhad Mehdipour and Kazuaki Murakami, "Early stage power management for 3D FPGAs considering hierarchical routing resources", in Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI, 2013.

[10].V. Betz and J. Rose, "Cluster-based logic blocks for FPGAs: area-efficiency vs. input sharing and size", in Custom Integrated Circuits Conference, 551–554, 1997.

[11].S. Wilton, "Architectures and Algorithms for Field-Programmable Gate Arrays with Embedded Memories", PhD thesis, University of Toronto, 1997.

[12].Vaughn Betz, "Architecture and CAD for the Speed and Area Optimization of FPGAs", PhD thesis, University of Toronto, 1998.

[13].VPR-Source code and benchmarks http://www. eecg. toronto.edu/~vaughn/vpr/vpr.html.

[14]. Vaughn Betz, Jonathan Rose, "VPR: A new packing, placement and routing tool for FPGA research", Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications, p.213-222, 1997.