What if Merging Connection and Switch Boxes — an Experimental Revisit on FPGA Architectures

Catherine L. Zhou, Ray C. C. Cheung and Yu-Liang Wu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
Email: {lzhou, cccheung, ylw}@cse.cuhk.edu.hk

***Abstract*— In this paper we proposed a new FPGA architecture using the Connection-Switch Box (CS-Box). It is based on the symmetric-array FPGA architecture and combines the Connection Box and the Switch Box. Two algorithms are designed to build the switches inside CS-Boxes, one for logic pins and the other for pad pins. After theoretical analysis and comparison, we conducted extensive experiments on MCNC benchmark circuits and made comparison between symmetric-array FPGAs and the proposed FPGA architecture on channel widths, circuit delays and switch numbers. By using the CS-Box structure the switch number of the connection boxes can be reduced by up to 11.81% with small penalty of increasing the channel width and the circuit delay by 0.38% and 2.34% on average respectively.**

Keywords: FPGA Architecture, Switch Box, Connection Box

I. Introduction

Field-Programmable Gate Array (FPGA) has been used for over a decade because of its wide applications in digital systems. The conceptual diagram of symmetric-array FPGAs is illustrated in Figure 1 [3]. The architecture of an FPGA has high impact on its routability. Based on the Xilinx FPGA architecture, many research works have addressed on the optimal switch box designs. Currently the most widely used switch box structures are Disjoint, Universal [4] [5] [9], HUSB [6] [7] and Wilton [10]. Their routabilities have been evaluated in [8].

In this paper, we propose a new research area to reduce the hardware resources. The increased flexiblity inside the CS-Box improves the overall chip routability and reduces the switch number of the entire chip. The new FPGA conceptual diagram is shown in Figure 2. It is based on the symmetric-array FPGA architecture. The difference is that the separate C-Box and S-Box structures in the symmetric-array FPGA are combined to form the Connection-Switch Box (CS-Box) structure in the new FPGA. The switch design algorithms we decided try to use as few switches as possible to accomplish routing. We conducted experiments on MCNC benchmark circuits. The results show 11.81% reduction of switch numbers from CS-Box FPGAs to symmetric-array FPGAs together with small penalty of channel widths and circuit delays. As F_{cl} in CS-Box FPGAs increases, the switch number reduction is degraded. And the penalty in channel widths and circuit delays is eliminated.

This paper is organized as follows. Sect. II introduces the connection algorithms in details. Sect. III analyzes the number of the logic-pin-and-track switches in the new FPGA architecture. To achieve a fair evaluation on the proposed architecture, we compare it with the symmetric-array FPGA architecture on different objectives. Sect. IV shows the experimental results with the comparison of channel widths, circuit delays and switch numbers. Sect. V draws our conclusions on this work.

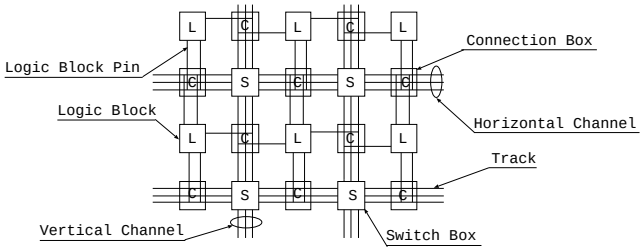

Fig. 1. Symmetric-Array FPGA Architecture

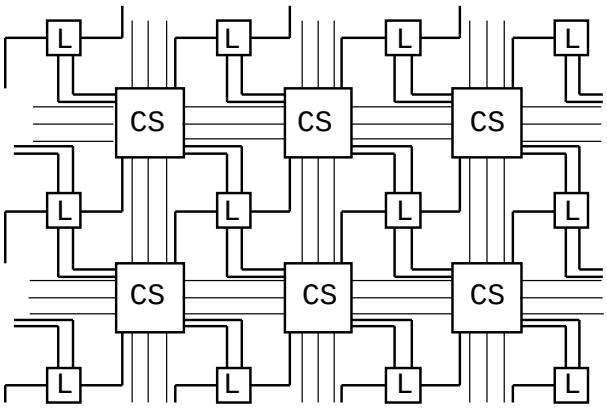

Fig. 2. CS-Box FPGA Architecture

II. Connection-Switch Box Design Algorithms

We designed two algorithms to do the connection between pins and tracks in the FPGA with CS-Boxes. One is for logic pins and the other for pad pins. In this paper W is used as channel width. P indicates the logic (pad) pin number on each logic block (pad). F_{cl} and F_{cp} denote the logic pin connectivity and the pad pin connectivity respectively. According to Figure 2 we determine that each pin connects to the tracks on the other sides except the one it belongs to.

The following two subsections introduce the two algorithms in details.

A. Connection between Logic Pins and Tracks

In this algorithm we regard each CS-Box as a 4-partite graph. The track(s) and the non-global logic pin(s) on one side compose of a disjoint vertex set of the graph. Here we call them track vertices (T-vertex) and pin vertices (P-vertex) respectively. Then the problem of designing switches in the CS-Box can be formulated as designing the edges between P-vertices and T-vertices in the 4-partite graph.

According to the notations at the beginning of Sect. II, P is the total number of P-vertices in the 4-partite graph. W is the number of T-vertices in each disjoint set of the graph. All the P-vertices in the graph are labeled from 1 to P. And the T-vertices in each disjoint set are labeled from 1 to W. We use v_{pi} to denote the ith P-vertex and $v_{ti,j}$ to denote the jth T-vertex of the ith disjoint vertex set. We name such a 4-partite graph as *PT-graph*. Then the switch design problem can be formulated as the following:

Problem Formulation Let G be a PT-graph. S_i and S_j $(1 \leqslant i, j \leqslant 4)$ are disjoint vertex sets of G. v_{pm} is a P-vertex in the graph, where $v_{pm} \in S_i$ and $1 \leqslant m \leqslant P$. A solution to the problem should connect each $v_{pm} \in S_i$ to some T-vertices in S_j, where $i \neq j$. The number of the edges is as small as possible while the number of the connected T-vertices is as large as possible.

Definition *W-PT-graph* is a PT-graph that has W T-vertices in each disjoint vertex set.

We note that the disjoint vertex sets have the same numbers of T-vertices and different numbers of P-vertices. Thus we make i to i matching from the label of P-vertices to those of T-vertices. That is, the ith P-vertex corresponds to the ith T-vertex in each set. Let $H = \{V, E\}$ be a W-PT-graph.

1) Case 1: $W = P$

This is the simplest case considered by this algorithm. The ith P-vertex is connected to the ith T-vertices that are not on the same side with the P-vertex . So the edge set of H is

$$E = \{(v_{pj}, v_{ti,j}) | v_{pj} \notin S_i; 1 \leqslant i \leqslant 4; 1 \leqslant j \leqslant W(P)\} \quad (1)$$

Figure 3 illustrates the 4-PT-graph H whose edges have been settled.

2) Case 2: $W \bmod P = 0$ and $W \neq P$

Here Case 1 is used as the unit case to obtain the solution. Each disjoint vertex set of H can be divided into W/P subsets. And all the subsets can be denoted by

$$\begin{aligned} V'_k &= \{v_{ti,p} | |p-k| \bmod P = 0; 1 \leqslant i \leqslant 4; 1 \leqslant p \leqslant W; \\ &\quad 1 \leqslant k \leqslant P\} \end{aligned} \quad (2)$$

The edges are set between the vertices in V'_k and the kth P-vertex. So the edge set of H is

$$\begin{aligned} E &= \{(v_{pj}, v_{ti,k}) | v_{pj} \notin S_i; |k-j| \bmod P = 0; 1 \leqslant k \leqslant W; \\ &\quad 1 \leqslant j \leqslant P; 1 \leqslant i \leqslant 4\} \end{aligned} \quad (3)$$

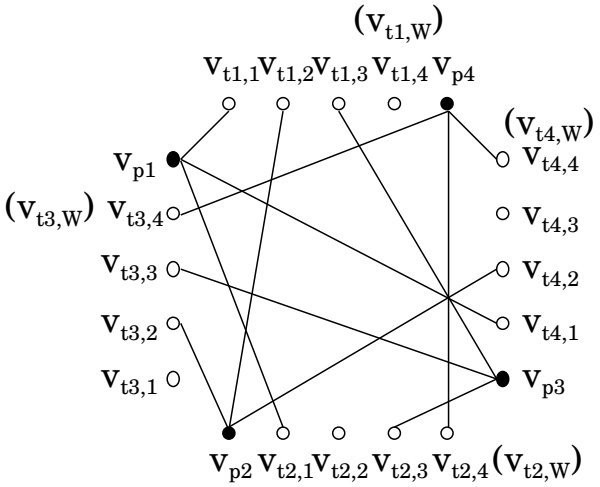

Fig. 3. W-PT-Graph H. Case 1: $W = P$

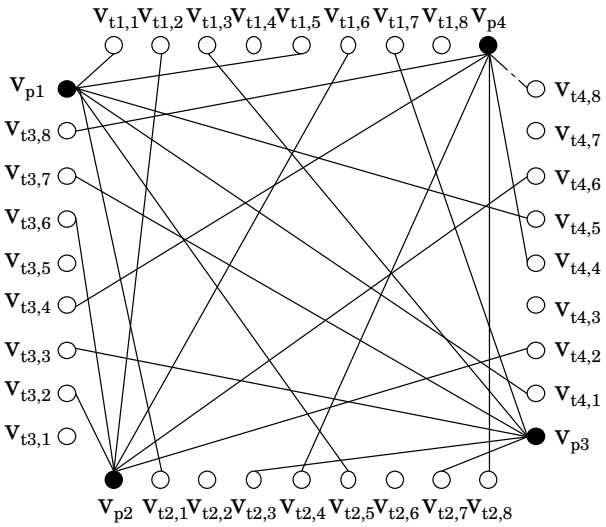

Fig. 4. W-PT-Graph H. Case 2: $W \bmod P = 0$ and $W \neq P$

Figure 4 illustrates the graph H that the edges have been determined.

3) Case 3: $W \bmod P \neq 0$

Let W' be an integer whose value is calculated by Equation (4).

$$W' = W - (W \bmod P) \quad (4)$$

In this case we divide the T-vertices into two groups. The first group can be denoted by the set $\{v_{ti,j} | 1 \leqslant i \leqslant 4$ and $1 \leqslant j \leqslant W'\}$. And the other T-vertices compose of the second group. Now it is clear that the first group corresponds to Case 2. The edges related to them can be determined according to Case 2, i.e.

$$\begin{aligned} E' &= \{(v_{pj}, v_{ti,k}) | v_{pj} \notin S_i; |k-j| \bmod P = 0; 1 \leqslant k \leqslant W'; \\ &\quad 1 \leqslant j \leqslant P; 1 \leqslant i \leqslant 4\} \end{aligned} \quad (5)$$

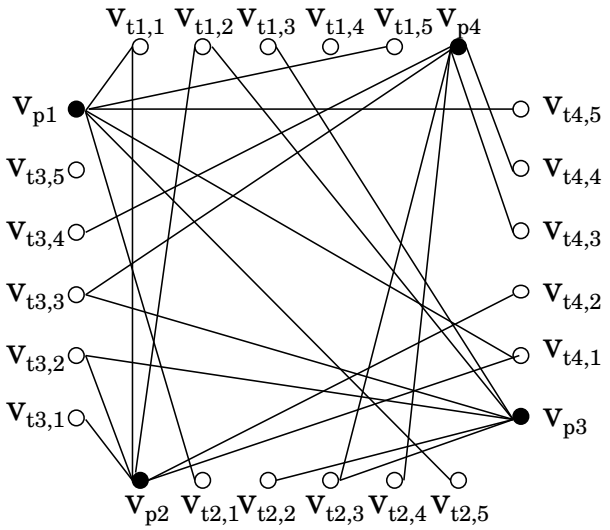

Fig. 5. W-PT-Graph H. Case 3: $W \text{mod} P \neq 0$

For the T-vertices in the second group, we can build the edge set E'' in the way similar to what has been introduced in Case 2 by modifying the labels of the first $P - W + W'$ T-vertices in each disjoint set. That is, for each $v_{ti,j}$ where $1 \leqslant j \leqslant P - W + W'$, relabel it as $v_{ti,j+W}$.Thus,

$$E'' = \{(v_{pj}, v_{ti,k}) | (W + k - j) \text{mod} P = 0; v_{pj} \notin S_i; 1 \leqslant k \leqslant P - W + W'; 1 \leqslant j \leqslant P; 1 \leqslant i \leqslant 4\} \quad (6)$$

Till now we can get the complete edge set E of H by

$$E = E' \cup E'' \quad (7)$$

Figure 5 illustrates the graph H with the edges settled.

According the above design, the connectivity of each logic pin can be calculated by

$$F_{cl} = (\lfloor W/P \rfloor + \beta) \times 3 \quad (8)$$

where $\beta = \begin{cases} 0, & \text{if } W \text{mod} P = 0 \\ 1, & \text{otherwise} \end{cases}$.

B. Connection between Pad Pins and Tracks

We propose a separate algorithm for the connection between pad pins and tracks because of the structural difference between pads and logic blocks. Let v_p be the pad vertex (P-vertex) denoting a pad pin. Let v_{xi} (v_{yi}) be the track vertex (T-vertex) denoting a track in the x-channel (y-channel). The switches to be determined are represented by the edges connecting P-vertices and T-vertices.

According to the notations at the beginning of Sect. II, F_{cp} is the connectivity of each pad vertex. In previous work we tried to use the FPGA router VPR [1] [2] to route the benchmark circuits on symmetric-array FPGAs. A fact is that some of the circuits are not able to be routed if F_{cp} equals to some number smaller than W. While all of them are able to be routed when F_{cp} equals to W. Therefore, in our design F_{cp} equals to W. The algorithm separates all the edges adjacent to one P-vertex into two groups. The edges in the first group are adjacent to the T-vertices in the x-channel; and this group is indicated by E_x. The other edges are adjacent to the T-vertices in the y-channel; and the group containing them is indicated by E_y. According to our algorithm, E_x and E_y can be represented by the following two sets,

$$E_x = \begin{cases} \{(v_p, v_{xi}) | i \text{mod} 2 = 0; 1 \leqslant i \leqslant W\}, & \text{if } W \text{mod} 2 = 0 \\ \{(v_p, v_{xi}) | i \text{mod} 2 = 1; 1 \leqslant i \leqslant W\}, & \text{otherwise} \end{cases} \quad (9)$$

$$E_y = \begin{cases} \{(v_p, v_{yi}) | i \text{mod} 2 = 1; 1 \leqslant i \leqslant W\}, & \text{if } W \text{mod} 2 = 0 \\ \{(v_p, v_{yi}) | i \text{mod} 2 = 0; 1 \leqslant i \leqslant W\}, & \text{otherwise} \end{cases} \quad (10)$$

III. Switch Number Comparisons

In this section we discuss the number of the switches connecting logic pins with tracks in the new FPGA architecture. The following variables are defined. W and P have the same meaning as in Sect. II.

- N_{cs} : It is defined for each CS-Box and equals to the number of the switches connecting logic pins and tracks.
- N_x : It is defined for each logic block in symmetric-array FPGAs and equals to the number of the switches connecting logic pins and tracks.
- α : The connection ratio of each logic pin in the symmetric-array FPGA. It indicates how much percent of tracks the logic pin is connected to in one channel.

According to the algorithm introduced in Sect. II,

$$N_{cs} = (\lfloor W/P \rfloor + \beta) \times 3 \times P \quad (11)$$

where, $\beta = \begin{cases} 0, & \text{if } W \text{mod} P = 0 \\ 1, & \text{otherwise} \end{cases}$. In the symmetric-array FPGA,

$$N_x = \alpha \times W \times P \quad (12)$$

Let k be a non-negative integer. And W can be denoted by

$$kP < W \leqslant (k+1)P \quad (13)$$

So there is

$$\lfloor W/P \rfloor + \beta = k + 1 \quad (14)$$

For N_{cs} and N_x,

$$N_{cs} = (k+1) \times 3 \times P \quad (15)$$

$$\alpha \times k \times P^2 < N_x = \alpha \times W \times P \leqslant (k+1) \times \alpha \times P^2 \quad (16)$$

We attempt to find the condition of $N_{cs} \leqslant N_x$. If $\alpha > 3/P$, there is $k < 3/(\alpha \times P - 3)$. Take the example of $P = 5$ and $\alpha = 1.0$, $N_{cs} > N_x$ occurs when $W < 3$.

From the above analysis, the FPGA architecture with CS-Boxes uses fewer logic-pin-to-track switches than the symmetric-array FPGA to acomplish routing. The next section will verify this analysis with extensive experiments.

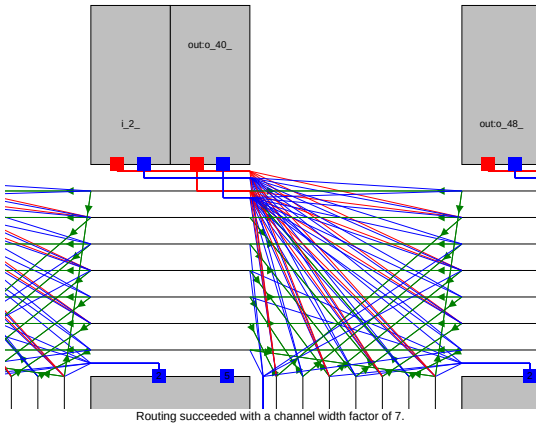

Fig. 6. CS-Box Struction: Pad Pin Connection

IV. Experimental Results

After the theoretical analysis and comparison, we observed the loss and gain of the routability, performance and size on the two types of FPGAs with the help of a 1500 MHz Intel Pentium 4 PC with 512M RAM. The currently best-known router VPR [1] [2] is used to test the routability and efficiency of the CS-Box structure. The iteration number for routing is 100. And the routing algorithm is the Breadth-First Algorithm. The Wilton switch box is applied to make the connection between tracks, i.e., S-Box in symmetric-array FPGAs. We tried different values of F_{cl} to route the benchmark circuits. F'_{cl} is defined as *basic value* and computed by Equation 17.

$$F'_{cl} = (\lfloor W/P \rfloor + \beta) \times 3 \tag{17}$$

where, $\beta = \begin{cases} 0, & \text{if } W \bmod P = 0 \\ 1, & \text{otherwise} \end{cases}$. Then Equation 18 is used to obtain the different F_{cl} values, where, $x = 0, 1, 2, \ldots$.

$$F_{cl} = F'_{cl} + x \times 3 \tag{18}$$

Table I shows the channel width requirements of different benchmark circuits. The circuit delay information is presented in Table II. In Table III we show the number of the logic-pin-and-track switches required in each benchmark circuit. From the tables, applying CS-Boxes can reduce switch numbers with small penalty of circuit delays and channel widths when $F_{cl} = F'_{cl}$ is used in the new FPGA. As F_{cl} of the CS-Box FPGA increases, the reduction of switch numbers becomes less while the penalty on circuit delays and channel widths is eliminated. Figure 6 and 7 shows the switch connections on the proposed design. The entire routing of the e64 benchmark circuit is shown in Figure 8.

V. Conclusions

In this paper we developed a new FPGA architecture, in which the disjoint C-Box and S-Box structures are combined to the CS-Box structure. There are two types of switches in the CS-Box: one of the switch boxes introduced in Sect. I is used to make connection between tracks, and the second type are the switches between pins and tracks. We introduced two algorithms to build the second type of switches. In the theoretical analysis of the switch number, we take the example of $P = 5$ and $\alpha = 1.0$ and find $N_{cs} \leqslant N_x$ when $W \geqslant 3$. The experimental results show 11.81% reduction on average in the number of switches inside the C-Box from the symmetric-array FPGA to the CS-Box FPGA, accompanied by small penalty of channel widths and circuit delays when $F_{cl} = F'_{cl}$ in CS-Box FPGAs. The larger F_{cl} in the new FPGA, the

TABLE I
Channel Width Requirements

FPGA Architecture	Symmetric-Array			with CS-Box Structure	
F_{cl}	W	$0.9W$	$0.8W$	F'_{cl}	$F'_{cl}+1$
Circuit Name	-	-	-	-	-
alu4	10	10	10	10	9
apex2	11	11	11	11	11
apex4	13	12	13	13	13
b9	4	4	4	4	3
bigkey	6	6	6	6	6
dalu	6	6	6	6	6
des	8	7	7	6	5
diffeq	7	8	8	8	7
dsip	7	7	7	7	6
e64	7	7	8	7	7
elliptic	10	11	10	11	10
ex1010	10	10	10	10	10
ex5p	13	13	13	13	13
misex3	10	10	10	11	10
my_adder	4	4	4	4	3
s1423	5	5	6	6	4
s298	8	8	8	8	7
s38417	8	8	7	7	7
s38584.1	7	7	7	7	7
seq	11	11	11	11	11
tseng	7	7	7	6	6
unreg	4	4	4	5	3
Total	176	176	177	177	164
Comparison	*	*	*	+0.57% +0.57% 0	-6.82% -6.82% -7.34%
Average Comparison Result				+0.38%	-6.99%

"*" indicates the result in the column is used to make comparison

TABLE II
Circuit Delay Comparisons ($e^{-7}ns$)

FPGA Architecture	Symmetric-Array			with CS-Box Structure	
F_{cl}	W	$0.9W$	$0.8W$	F'_{cl}	$F'_{cl}+1$
Circuit Name	-	-	-	-	-
alu4	1.85386	1.56852	1.90095	1.66109	1.66125
apex2	1.607	1.63341	1.3474	1.73219	1.76429
apex4	1.42238	1.43766	1.5797	1.31174	1.26781
b9	0.36672	0.341982	0.365442	0.293081	0.446448
bigkey	1.62792	1.98333	2.40621	1.48558	1.49052
dalu	1.33564	1.22028	1.18171	1.43429	1.10201
des	1.69223	2.04452	1.54254	2.39677	2.09814
diffeq	1.39117	0.977372	1.11989	1.14553	1.07257
dsip	1.20737	1.56772	1.52951	1.78454	1.72232
e64	0.493154	0.43602	0.424841	0.416376	0.431577
elliptic	2.86504	1.96451	2.64456	2.01054	2.52907
ex1010	3.8845	3.35407	3.66273	3.84098	3.12935
ex5p	1.37349	1.22278	1.26679	1.34682	1.32818
misex3	1.71805	1.44106	1.53773	1.68674	1.96567
my_adder	0.534498	0.530612	0.547252	0.596041	0.449367
s1423	0.73654	0.675588	0.651449	0.664029	0.592272
s298	2.70654	2.2548	2.34736	2.68695	2.29187
s38417	1.72278	1.83241	1.79846	2.36803	1.84059
s38584.1	1.72601	2.7448	2.61386	1.98874	2.59543
seq	1.48821	1.82375	1.90447	1.78949	1.54419
tseng	1.5054	1.25099	1.17455	1.18151	1.05066
unreg	1.99059	0.187206	0.1983	1.81475	1.52287
Total	33.457561	32.49339	33.745704	34.002532	32.525871
Comparison	*	*	*	+1.63% +4.64% +0.76%	-2.78% +0.09% -3.61%
Average Comparison Result				+2.34%	-2.10%

"*" indicates the result in the column is used to make comparison

TABLE III

SWITCH NUMBER REQUIREMENTS

FPGA Architecture	Symmetric-Array			with CS-Box Structure	
F_{cl}	W	$0.9W$	$0.8W$	F'_{cl}	$F'_{cl}+1$
Circuit Name	-	-	-	-	-
alu4	80000	72000	64000	48000	72000
apex2	106480	968000	87120	87120	87120
apex4	84240	71280	66096	58320	58320
b9	2420	2420	1936	1815	3630
bigkey	87480	75816	72900	87480	87480
dalu	34680	30056	28900	34680	34680
des	158760	123039	119070	119070	119070
diffeq	53235	54756	47151	45630	68445
dsip	102060	90396	87480	87480	131220
e64	10115	8959	8959	8670	8670
elliptic	186050	186050	148840	167445	167445
ex1010	231200	208080	184960	138720	138720
ex5p	70785	65340	55539	49005	49005
misex3	72200	64980	57760	64980	64980
my_adder	980	980	784	735	1470
s1423	5625	5625	5625	6750	6750
s298	77440	69696	60016	58080	87120
s38417	262440	236196	196830	196830	196830
s38584.1	229635	203391	196830	196830	196830
seq	97020	88220	79380	79380	79380
tseng	38115	33759	32670	32670	32670
unreg	980	980	784	735	1470
Total	1991940	1788799	1603630	1570425	1693305
Comparison with *	*	*	*	-21.16% -12.21% -2.07%	-14.99% -5.34% +5.59%
Average Comparison Result				-11.81%	-4.91%

"*" indicates the result in the column is used to make comparison

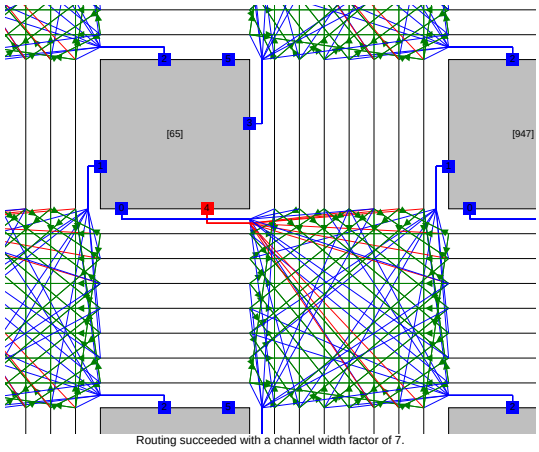

Fig. 7. CS-Box Struction: Logic Pin Connection

smaller the reduction of switch numbers. But the penalty of the channel widths and circuit delays is eliminated.

REFERENCES

[1] V. Betz, J. Rose: VPR: A New Packing, Placement and Routing Tool for FPGA Research. In: Proc. International Workshop on Field Programmable Logic and Applications (1997) 213-222

[2] V. Betz, J. Rose, Alexander, M.: Architecture and CAD for Deep-Submicron FPGAs. Kluwer Academic Publishers, Boston (1999)

[3] S. D. Brown, R. J. Francis, J. Rose, Z. G. Vranesic: Field-Programmable Gate Arrays. Kluwer Academic Publishers, Boston (1992)

[4] Y. W. Chang, D. F. Wong, C. K. Wong: Universal Switch Modules for FPGA Design. ACM Trans. on Design Automation of Electronic Systems **1(1)** (January 1996) 80-101

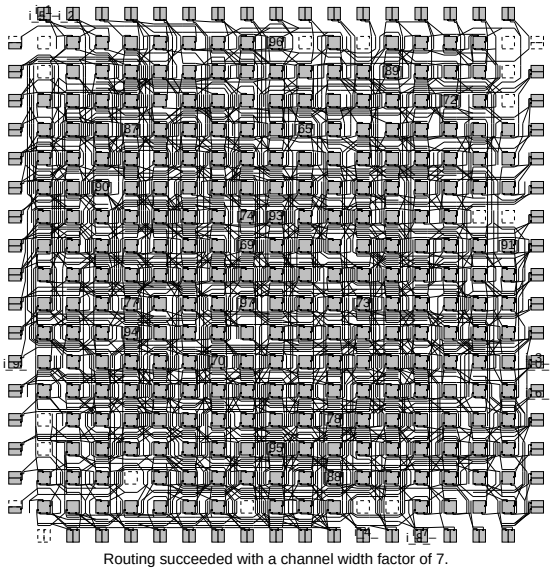

Fig. 8. Routing Result of *e*64 by Using CS-Boxes, W=7

[5] H. Fan, J. Liu, Y.-L. Wu: Combinatorial Routing Analysis and Design of Universal Switch Blocks. In: Proc. Conference on Asia South Pacific Design Automation (January 2001) 641-644

[6] H. Fan, J. Liu, Y.-L. Wu: General Models and a Reduction Design Techinique for FPGA Switch Box Designs. IEEE Trans. on Computers **52(1)** (January 2003) 21-30

[7] H. Fan, J. Liu, Y.-L. Wu, C.-C. Cheung: On Optimum Switch Box Designs for 2-D FPGAs. In: Proc. IEEE/ACM Design Automation Conference (DAC'2001) (June 2001) 203-208

[8] H. Fan, J. Liu, Y.-L. Wu, C.-C. Cheung: On Optimal Hyper Universal and Rearrangeable Switch Box Designs. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems **22(12)** (2004) 1637-1649

[9] M. Shyu, G.-M. Wu, Y.-D. Chang, Y.-W. Chang: Generic Universal Switch Blocks. IEEE Trans. on Computers **49(4)** (April 2000) 348-359

[10] S.J.E. Wilton: Architecture and Algorithms for Field-Programmable Gate Arrays with Embedded Memory. PhD thesis, University of Toronto (1997)