Bent Routing Pattern For FPGA

Xibo Sun, Hao Zhou*, Lingli Wang

State Key Laboratory of ASIC and System
Fudan University, Shanghai, China

{*xbsun17,zhouhao,llwang*}*@fudan.edu.cn*

September 9, 2019

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Island-Style FPGA

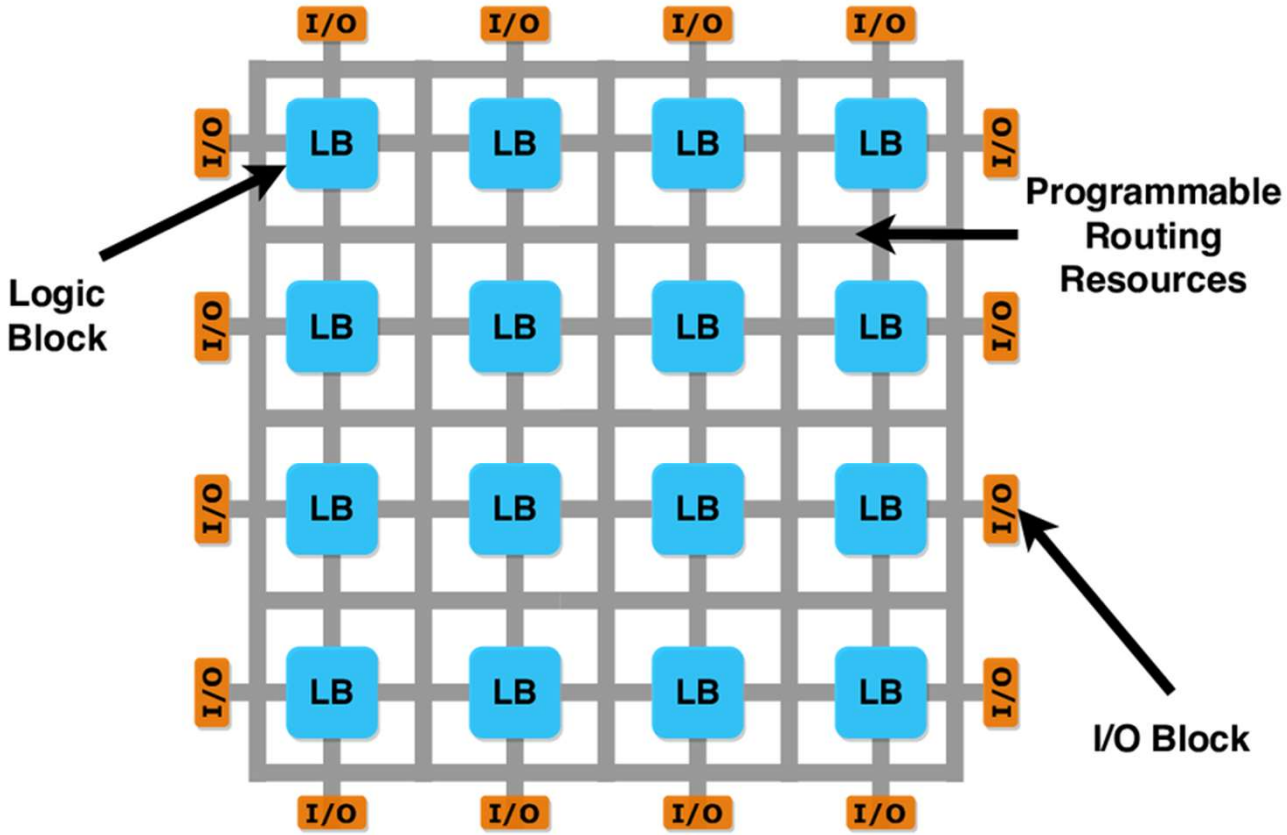
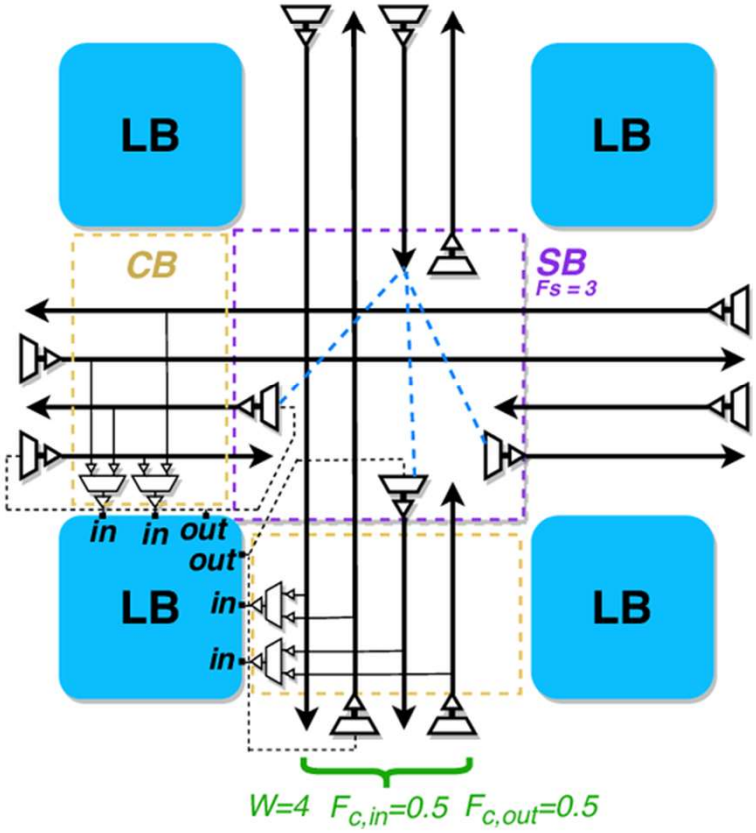


Figure 1: Architecture of island-style FPGAs

Routing Architecture



LB : **L**ogic **B**lock

CB : **C**onnection **B**lock

SB : **S**witch **B**lock

Fc : **C**onnection Block **F**lexibility

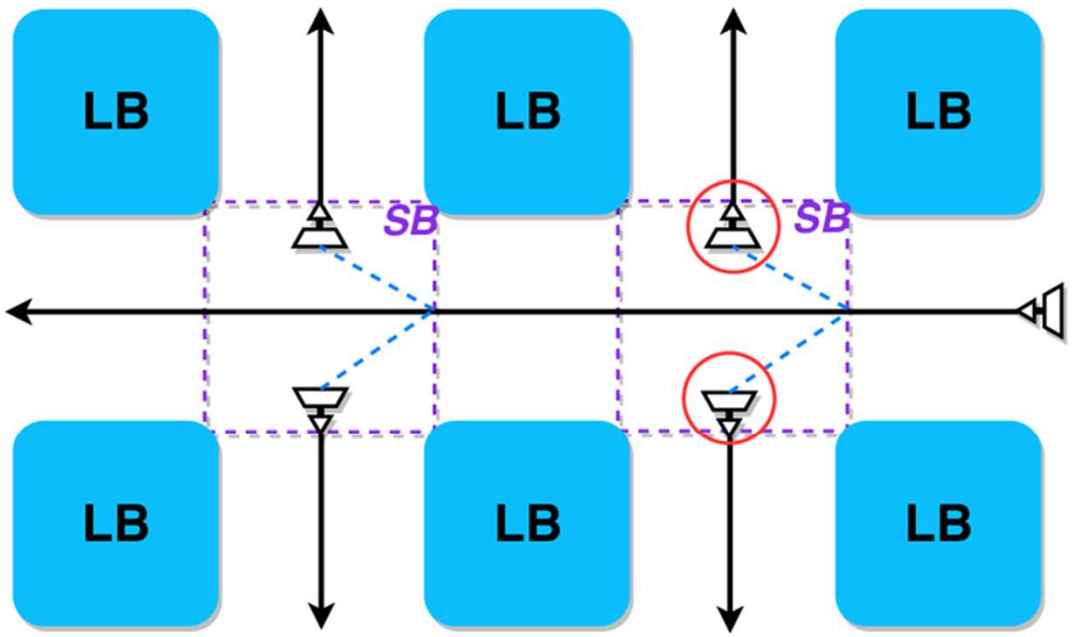
Fs : **S**witch Block **F**lexibility

L : Wire **L**ength

W : Channel **W**idth

The Idea of Bent Routing Pattern

A problem in the architecture with directional wires



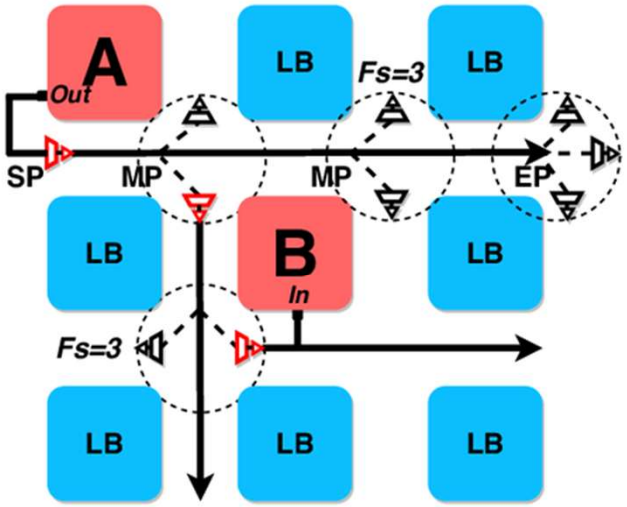
To improve performance:

- In the Straight direction : Use the multi-length wires
- In the direction of Turn : Is that necessary to pass through muxes?

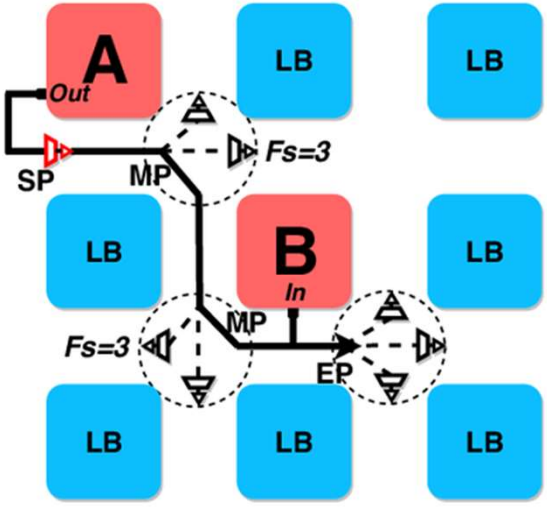
The Idea of Bent Routing Pattern

Comparison of architecture with and without Bent Routing Pattern

SP : Start Point (Driver) **MP** : Middle Point **EP** : End Point



- With Straight wires
- Pass through 3 muxes
- $F_s = 3$



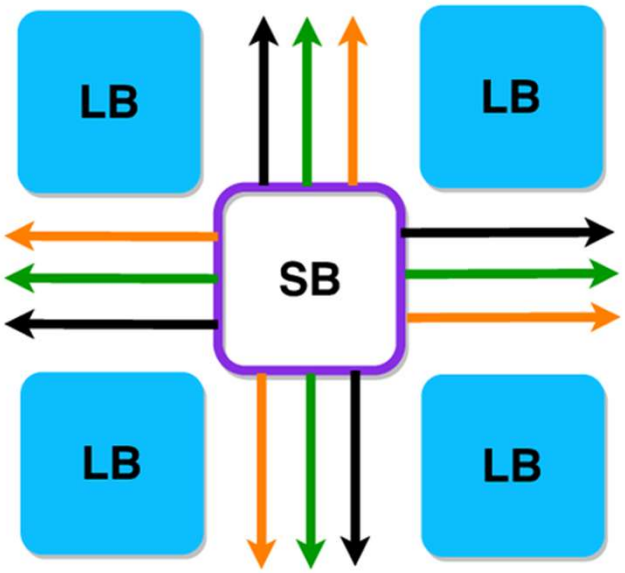
- With Bent wires
- Pass through 1 mux
- $F_s = 3$

Overview

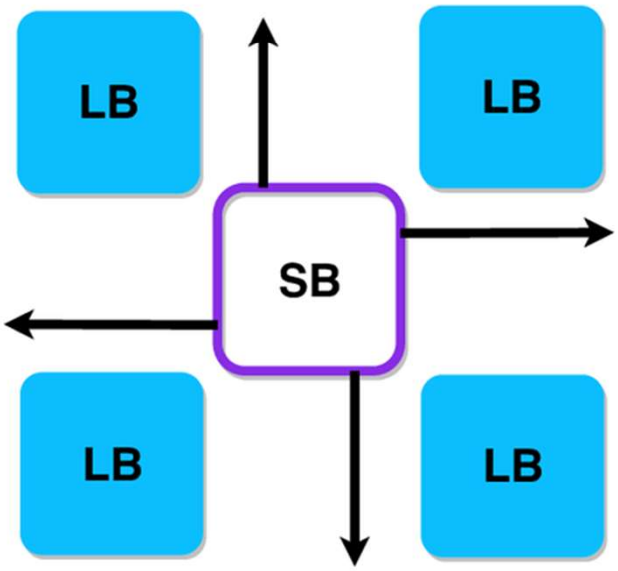
1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Bent Routing Pattern Design

Regularity and Symmetry of Straight Wires



Same Distribution



Symmetrical

Around an SB:

- Same distribution of wire types
- A group of wires is symmetrical

Bent Routing Pattern Design

Regularity and Symmetry of Bent Wires

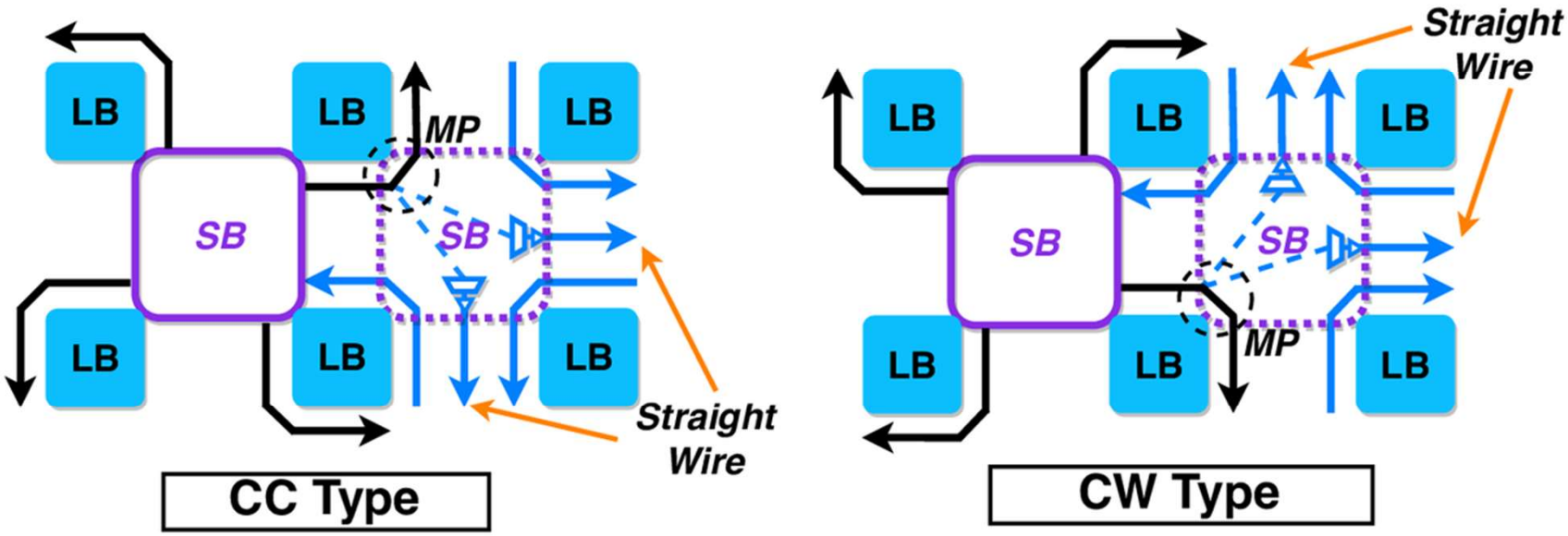


Figure 2: The examples of bent types in the length-2 bent wires

Different two bent types for bent wires:

- **CC Type** : **C**ounter**C**lockwise type
- **CW Type** : **C**lock**W**ise type

Bent Routing Pattern Design

Tile-Based Design

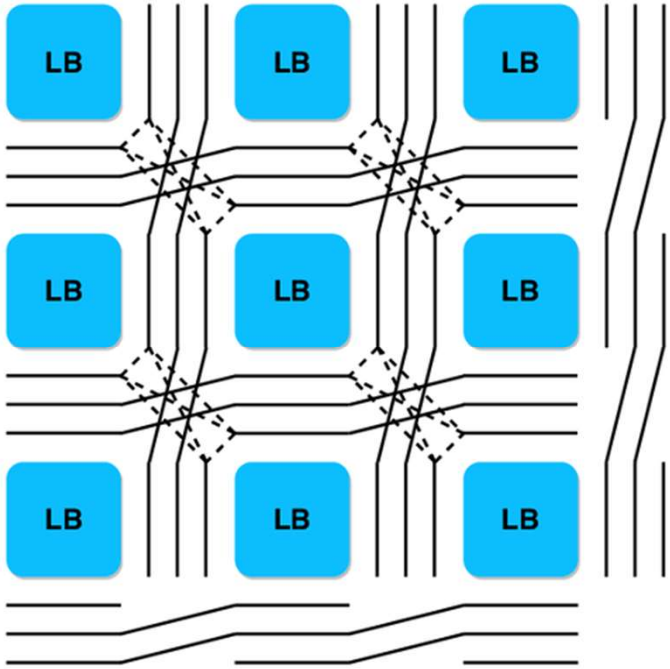


Figure 3: Tile-Based Design[1]

Easy to layout:

- Use a replication of single tile.

Restriction:

- "Start point" of segments should be staggered

[1] V.Betz *et al.*, *Architecture and CAD for deep-submicron FPGAs*. Kluwer Academic Publishers, 1999.

Bent Routing Pattern Design

Staggering for Straight Wires & Bent Wires

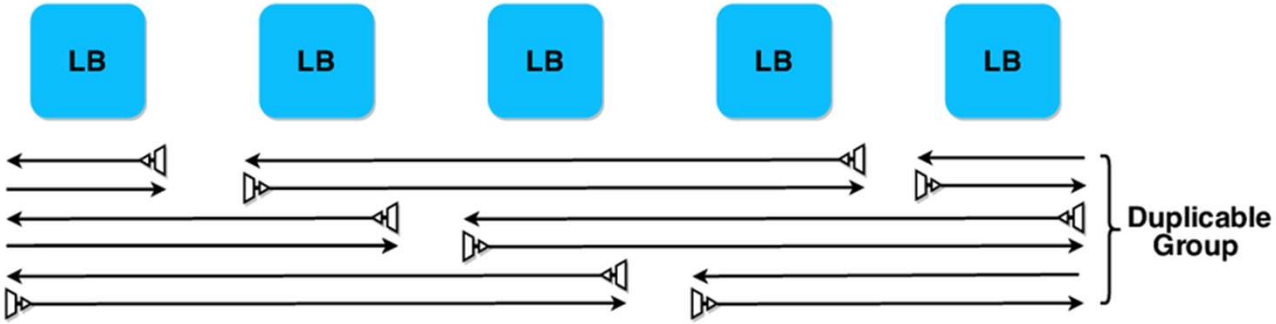


Figure 4: Staggering for Straight Wires

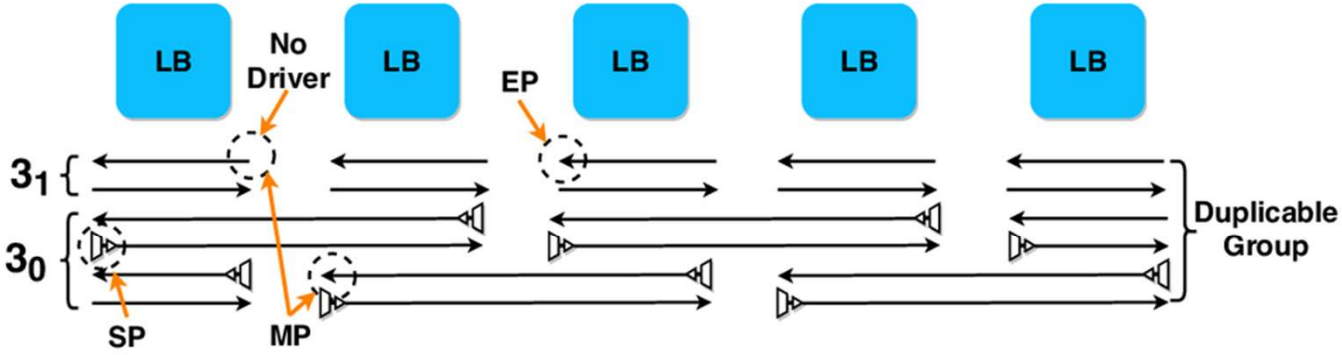


Figure 5: Staggering for Bent Wires

Bent Routing Pattern Design

Extended Architecture File in VTR[1]

```
<segment name="bent_segment"
         length="4" type="unidir"
         Cmetal="1.962e-14"
         Rmetal="101" freq="0.2">
  <mux name="0"/>
  <sb type="pattern"> 1 1 1 1 1 </sb>
  <cb type="pattern"> 1 1 1 1 </cb>
  <bend> ST CC CW </bend>
</segment>
```

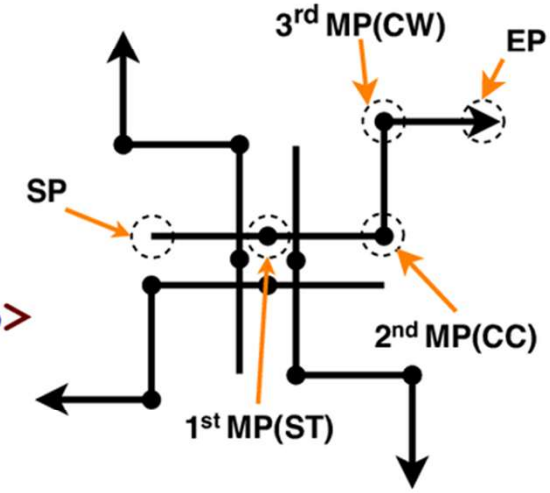


Figure 6: Sample description of length-4 bent wires

- Add the <bend> tag for the bent wires.
- Possible bent types at each MP: {ST,CC,CW}

[1] J. Luu *et al.*, "VTR 7.0: Next Generation Architecture and CAD System for FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1-6:30, Jul. 2014.

Bent Routing Pattern Design

Example of Description of Wire Types

Wire Type	Rate
Length-2 Bent Wire	20%
Length-3 Bent Wire(1)	20%
Length-3 Bent Wire(2)	20%
Length-4 Straight Wire	40%

```
<segment name="bent_2" length="2" ... freq="0.2">
  ...
  <bend> CC </bend>
</segment>
<segment name="bent_3" length="3" ... freq="0.2">
  ...
  <bend> ST CC </bend>
</segment>
<segment name="bent_3_1" length="3" ... freq="0.2">
  ...
  <bend> CC CW </bend>
</segment>
<segment name="straight_4" length="4" ... freq="0.4">
  ...
  <bend> ST ST ST </bend>
</segment>
```

Figure 7: Example of wire types in routing channel

Bent Routing Pattern Design

Modification of RRG Generator in VTR

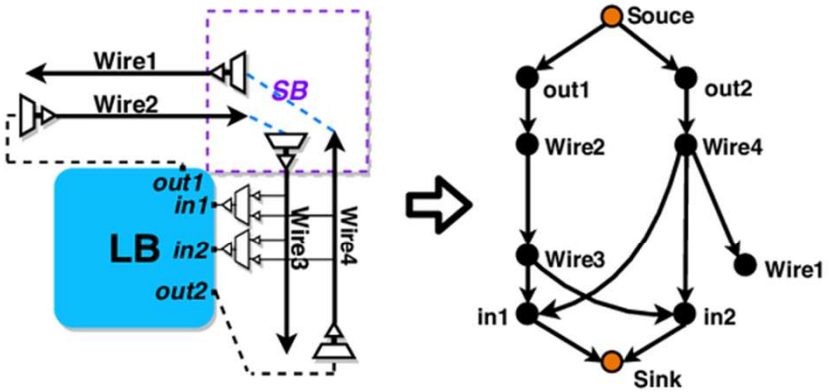


Figure 8: RRG with straight wires

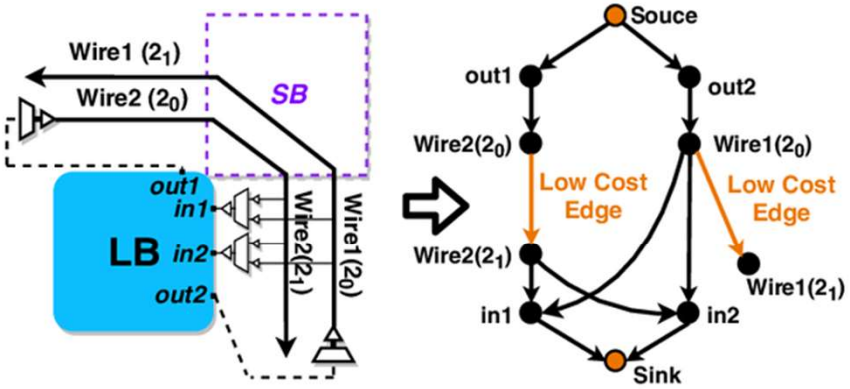


Figure 9: RRG with bent wires

[1] RRG: Routing Resource Graph

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Enhanced Router Lookahead Algorithm[1]

The Cost of Node n

$$Cost(n) = PathCost(s, n) + \alpha \cdot \textbf{ExpectedCost(n,t)}$$

ExpectedCost(n,t):

Related to the smallest number of wire segments to reach the target node

Default Router Lookahead:

- ☺ Single wire type
- ☹ Mixed wire types

Enhanced Router Lookahead[1]:

- ☺ Single wire type
- ☺ Mixed wire types

[1] O. Petelin and V. Betz, "The speed of diversity: Exploring complex FPGA routing topologies for the global metal layer," *in 2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1-10

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Searching Algorithm for Bent Routing Pattern

Searching for Optimized Wire Types

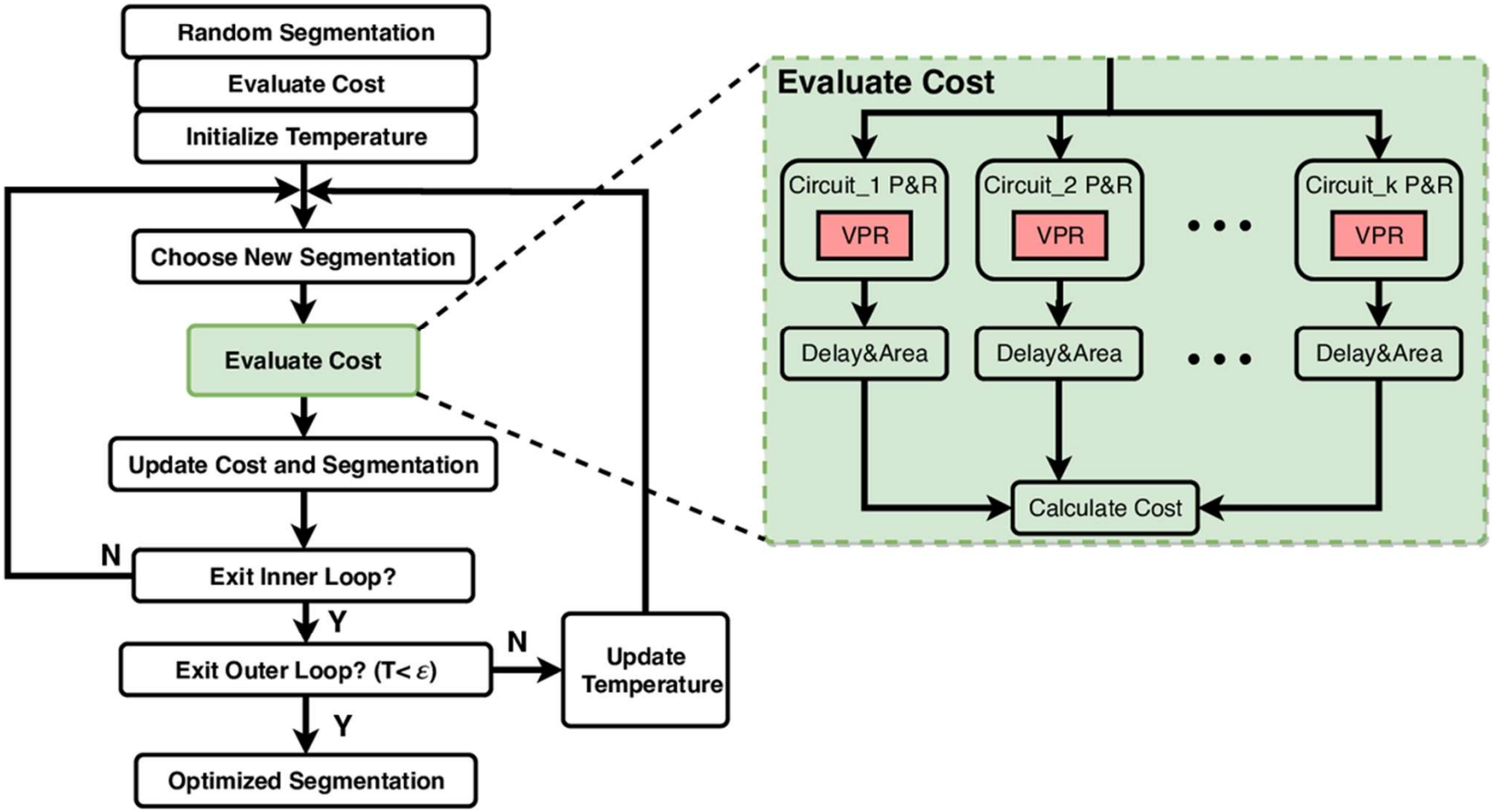
Figure 10: Design wire types in the routing channel

- Different wire types → different area and performance.
- Automaticlly design the wire types[1].

[1] M. Lin and A. El Gamal, "TORCH: A design tool for routing channel segmentation in FPGAs," in *Proceedings of the 16th InternationalACM/SIGDA Symposium on Field Programmable Gate Arrays*, 2008, pp. 131-138.

Searching Algorithm for Bent Routing Pattern

Searching for Optimized Wire Types



Searching Algorithm for Bent Routing Pattern

Searching for Optimized Wire Types

The Cost Function

$$Cost(A) = \frac{1}{k}\sum_{b=1}^{k}\left(\frac{a_{b,M}}{a_{b,base}}\right)^{\alpha} \times \left(\frac{d_{b,M}}{d_{b,base}}\right)^{\beta}$$

- k : Number of circuits
- $a_{b,base}$: Area of baseline architecture
- $a_{b,M}$: Area of architecture with additional bent wires
- $d_{b,base}$: Delay of baseline architecture
- $d_{b,M}$: Delay of architecture with additional bent wires
- α, β : Trade-off exponent, $\alpha, \beta > 0$

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Experimental Result

Architecture Parameters

Table 1: Baseline Architecture Parameters

LB Size	Ten 6-input Fracturable LUTs
LB Input Crossbar	Half-populated
LB Output Crossbar	None
DSP Elements	36x36 Fracturable Multipliers
Memories	32Kb Block RAMs
Channel Width	300
$F_{c,in}$	0.1
$F_{c,out}$	0.1

Table 2: Timing Parameters Extracted From **COFFE 2**[1]

	L2	L3	L4	L5	L6
Total Delay(ps)	93.94	114.4	142.4	174.3	212.1
Mux Pass Gate Area (MWTAs)	1.96	1.74	1.5	1.5	1.5
Buffer Area (MWTAs)	19.64	22.23	25.56	26.21	26.42
Wire Resistance Per Tile (Ω)			259.6		
Wire Capacitance Per Tile (fF)			3.549		

[1] S. Yazdanshenas and V. Betz, "COFFE 2: Automatic Modelling and Optimization of Complex and Heterogeneous FPGA Architectures," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 12, no. 1, pp. 3:1-3:27, Jan. 2019.

Experimental Result

The Result of Searching Algorithm

- **Baseline Architecture**:
  With length-4 straight wires[1]

Table 3: **Baseline Architecture**

Index	Length	Ratio	<Bend> List
0	4	100%	ST ST ST

- **Optimized Architecture**:
  With combination of bent and straight wires

Table 4: **Optimized Architecture**

Index	Length	Ratio	<Bend> List
0	3	19%	CC ST
1	6	36%	ST ST ST ST CC
2	4	45%	ST ST ST

- **Reference Group**:
  Same length and distribution of wires with the optimized architecture, but all the wires are straight

Table 5: **Reference Group**

Index	Length	Ratio	<Bend> List
0	3	19%	ST ST
1	6	36%	ST ST ST ST ST
2	4	45%	ST ST ST

[1] O. Petelin and V. Betz, "The speed of diversity: Exploring complex FPGA routing topologies for the global metal layer," *in 2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1-10

Experimental Result

The Result of Three Architectures

Benchmark set is chosen from VTR.[1]

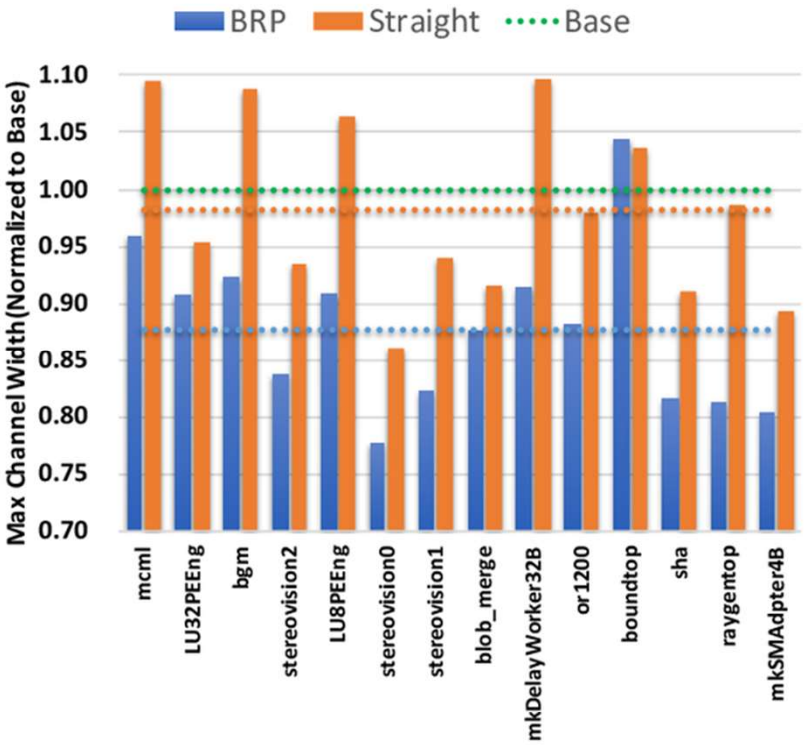


Figure 11: Maximum Channel Width

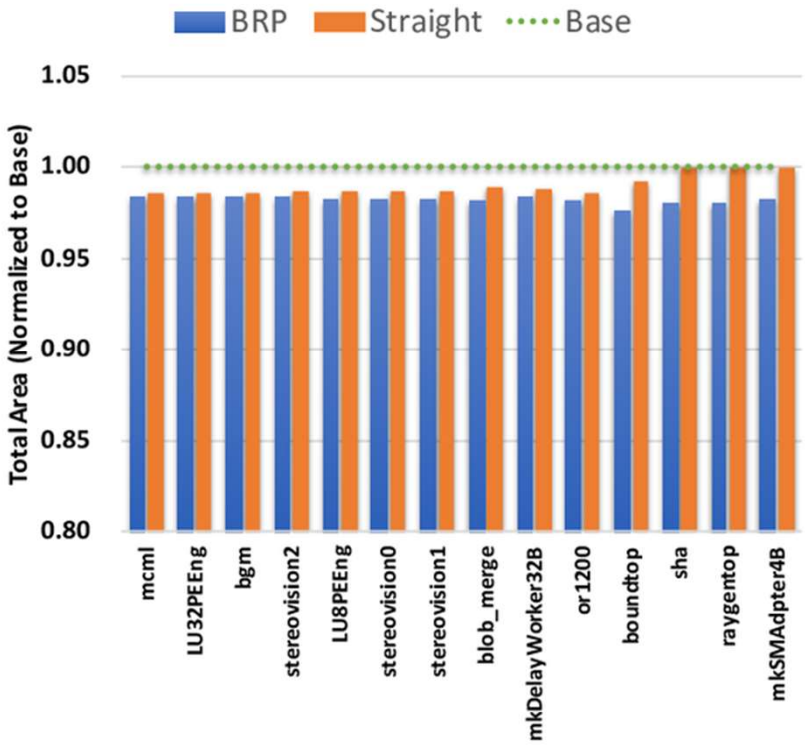


Figure 12: Total Area

[1] J. Luu *et al.*, "VTR 7.0: Next Generation Architecture and CAD System for FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1-6:30, Jul. 2014.

Experimental Result

The Result of Three Architectures

Benchmark set is chosen from VTR.[1]

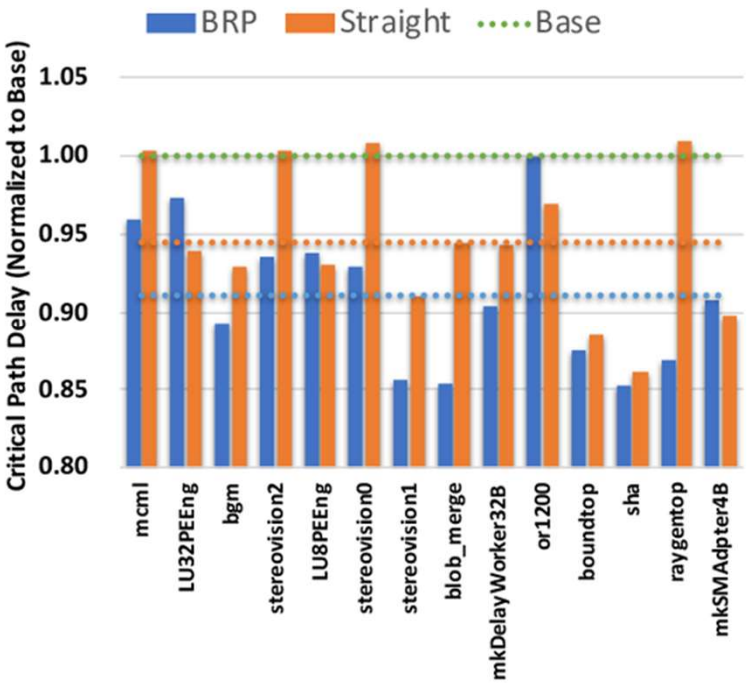


Figure 13: Critical Path Delay

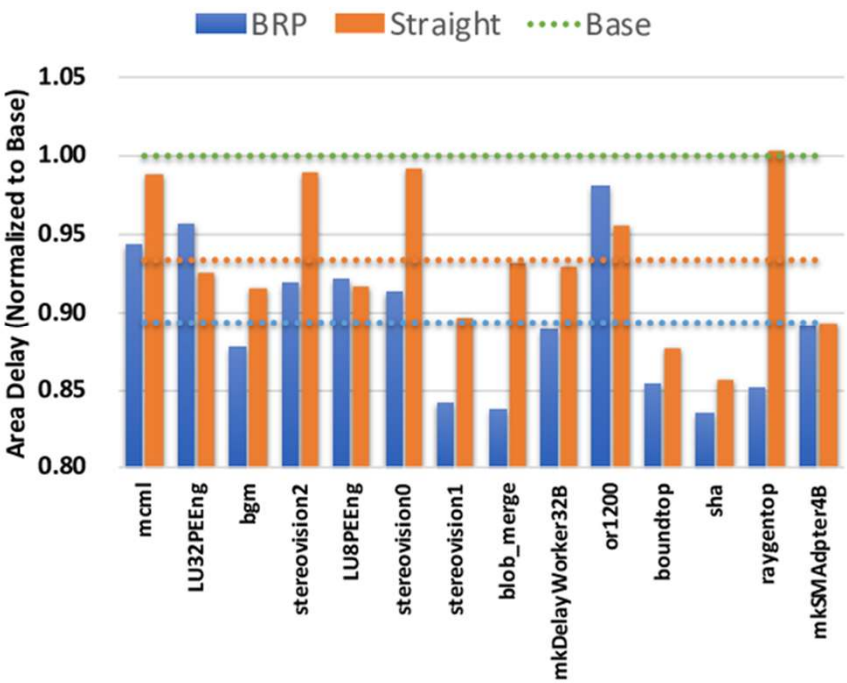


Figure 14: Area-Delay Product

[1] J. Luu *et al.*, "VTR 7.0: Next Generation Architecture and CAD System for FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1-6:30, Jul. 2014.

Overview

1. Introduction
2. Bent Routing Pattern Design
3. Enhanced Router Lookahead Algorithm
4. Searching Algorithm for Bent Routing Pattern
5. Experimental Result
6. Conclusion & Future Work

Conclusion & Future Work

Conclusion

- Bent routing pattern is proposed and evaluated by extended VTR.
- An simulated annealing based searching algorithm is developed to search the optimized architecture.
- The architecture with some bent wires have more than 10% savings in the area-delay product, compared to the baseline architecture.
- The bent routing pattern increases the diversity of the wire types in the routing channel.

Future Work

- Since the speed and quality of CAD tool limit the architecture exploration, we will seek for more efficient algorithms.

The End

Backup Slides

Bent Routing Pattern Design

Problem : Imbalance of bent wires in the fringe SB

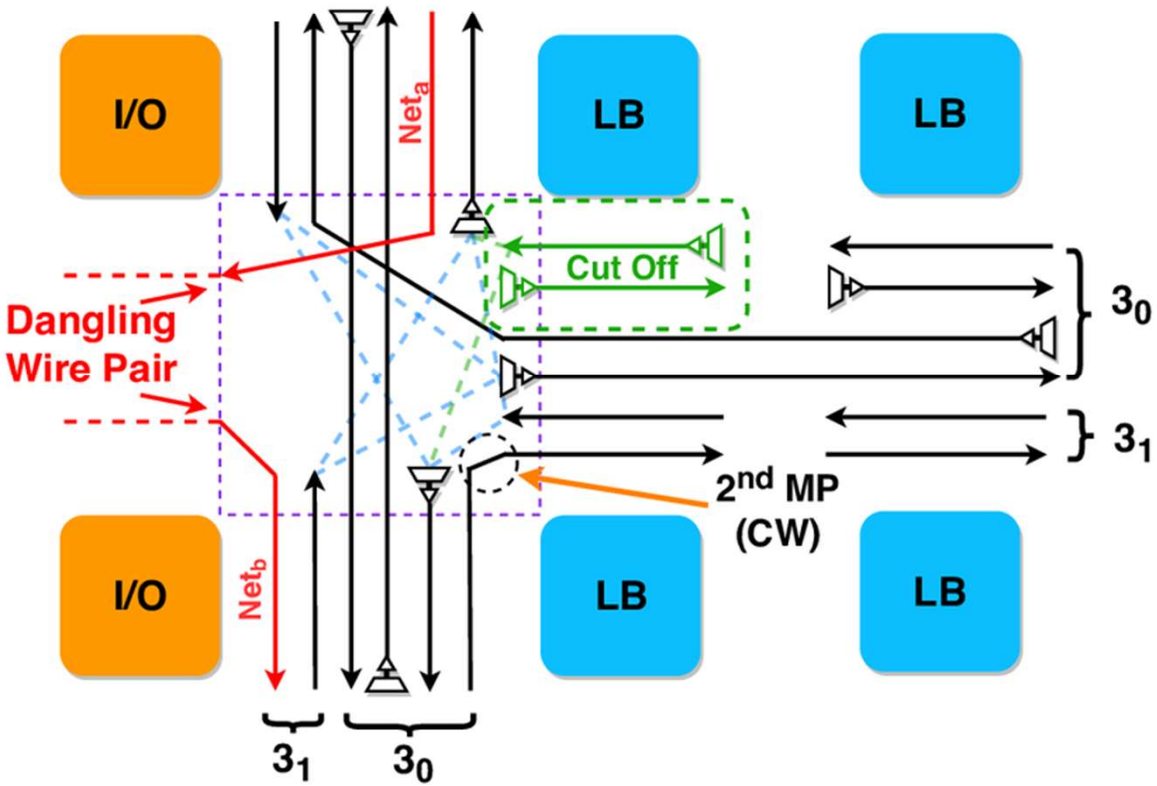


Figure 15: Length-3 bent wires in the leftmost SB

Solution:
Connect incoming and outgoing dangling wire pairs.
(e.g. $Net_a \rightarrow Net_b$)