Two-level MUX Design and Exploration in FPGA Routing Architecture

Yuhang Shen, Jiadong Qian, Kaichuang Shi, Lingli Wang, Hao Zhou
State Key Laboratory of ASIC and System
Fudan University, Shanghai, China
zhouhao@fudan.edu.cn

***Abstract*—In FPGAs, the programmable interconnect is implemented by multiplexers (MUXes), which have a large impact on the area and delay. In academia, large MUXes are extensively used in intra and inter clusters, resulting in significant FPGA area overhead and load for routing wires. In this paper, we model the interconnect from routing wires and CLB feedbacks to LUT inputs as an input block (IB), and implement the IB and the switch block (SB) with the 2-level MUX topology. Applying the 2-level MUX topology in FPGA routing architecture enables us to explore a larger design space for the area and delay, because the 2-level MUX topology can tradeoff between MUX sizes, connectivity degree, and the input bandwidth. We carefully design a baseline 2-level MUX routing architecture and evaluate it by running place and route experiments with VTR benchmarks. To optimize the baseline 2-level MUX routing architecture, we explore one design parameter at a time by keeping others fixed and perform subsequent explorations based on previous optimal design parameters. The results show that the optimized 2-level MUX routing architecture can achieve 19% shorter critical path delay (CPD) at the cost of 3% area overhead compared to the CB-SB FPGA architecture with 1-level MUX topology.**

I. Introduction

The routing architectures have long been the design bottleneck for Field-Programmable Gate Arrays (FPGAs). In both industry and academia, design metrics for FPGA such as the delay, area, and routability are heavily dependent on routing architectures. For traditional island-style FPGAs [1], the routing architecture usually consists of crossbars inside the cluster logic block (CLB), SBs, and connection blocks (CBs). Past researches [2]–[7] on SB patterns, CB flexibility, and crossbar sparsity reached many insightful architectural conclusions. However, there is not enough attention on the design of detailed routing MUX topology and sizes. In typical academic FPGA routing architecture design, routing MUXes usually come as a result of abstract architectural ideas and CAD algorithms [8]. This kind of top-down MUX design methodology tends to generate wide MUXes and lead to much load for routing wires. Therefore, it is of significance to design routing MUXes from the bottom to up to tightly manipulate the routing architecture.

Reference [9] proposed input interconnect blocks (IIBs) which can route signals on routing wires and CLB feedbacks to LUT inputs. Three types of IIBs were compared by place and route experiments and the results showed that IIBs with 2-level MUX topology can achieve big area savings while maintaining enough routability. This work inspires us that applying the 2-level MUX topology in FPGA routing architecture may enable a much bigger design space for the area and delay without degradation to routability. In this paper, we propose a 2-level MUX routing architecture and investigate the impact of 2-level MUX topology on architectural performances. Our contributions include:

- We model the interconnect from routing wires and CLB feedbacks to LUT inputs as an input block (IB) and apply a well-designed 2-level MUX topology to implement the IB and SB. The input bandwidth of the IB and SB are not strictly constrained and can be changed in a large scale compared to the CB-SB FPGA architecture. To optimize the routing area, we divide the IB and SB into multiple sub-IBs and sub-SBs. The first-level MUXes (L1-MUXes) within each sub-IB (sub-SB) can only connect to second-level MUXes (L2-MUXes) within the same sub-IB (sub-SB).
- We extend the FPGA architecture description file and Routing Resource Graph (RRG) generator in the latest VTR 8 [10], to model the IB and SB with 2-level MUX topology and evaluate it through place and route experiments. The circuitry parameters of 2-level MUXes cannot be generated from COFFE2 [11] because it cannot model the 2-level MUX topology and generate according circuitry. An extension to COFFE2 is augmented to model the 2-level MUX topology and produce optimized architecture parameters by transistor sizing.
- An experimental approach is applied to evaluate the 2-level MUX routing architecture by running place and route experiments with VTR benchmarks. We optimize the baseline architecture in terms of IB input bandwidth, SB input bandwidth, and the number of sub-IBs within the IB. Then the optimized 2-level MUX routing architecture is compared to the CB-SB FPGA architecture, the results show that it achieves about 19% shorter CPD and 18% less segment usage at the cost of 3% area overhead. It can justify that 2-level MUX topology brings large design space for the area, delay, and the routability.

The rest of the paper is organized as follows. Section II gives background and the related work. Section III introduces 2-level MUX routing architecture and section IV discusses the

This work was supported in part by the National Science Foundation of China under Grant No. 61971143.

1946-1488/21/$31.00 ©2021 IEEE
DOI 10.1109/FPL53798.2021.00045

modifications of VPR and COFFE2. In section V, we optimize the 2-level MUX routing architecture and present experimental results compared to the CB-SB FPGA architecture. Section VI concludes the work.

II. Background and Related Work

A. The CB-SB FPGA Routing Architecture

Fig. 1 illustrates the routing paths in a tile of the CB-SB FPGA architecture. The interconnect from routing wires to LUT inputs is composed of CB MUXes and local MUXes. The number of CB MUX is equal to CLB input bandwidth and the size of CB MUX is determined by fc [1] and the channel width. Input bandwidth is the maximal number of distinct routing signals allowed to go into a CLB at the same time. Local MUXes have fan-ins from CB MUX outputs and CLB feedbacks, and the crossbar between CB MUXes and local MUXes is usually fully or half populated. The LUT outputs are routed to SB MUXes through an output crossbar whose connection can be user-specified in the VPR architecture file. The number of SB MUXes is determined by the channel width, and the size of SB MUX is related to fc and fs [1]. SB MUXes are used to connect routing wires to route signals among CLBs. It is the common case that SB MUX, CB MUX, and local MUX have large fan-in sizes that burden routing wires with large loads.

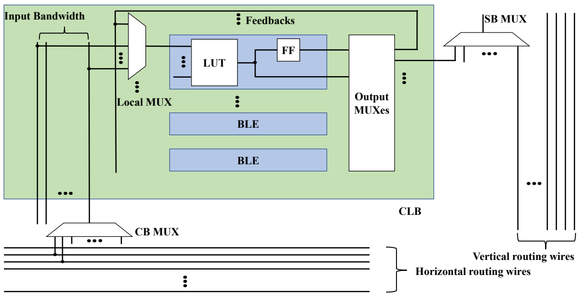

Fig. 1. Routing paths in a tile of the CB-SB architecture.

B. CAD Tools

VTR 8 is the open-source framework used for academic FPGA architecture and CAD algorithm exploration. VPR [12] is part of the VTR project that performs packing, placement, routing, and timing analysis. In this paper, we modify VPR to perform place and route experiments to evaluate the 2-level MUX routing architecture.

COFFE2 is an automated transistor sizing tool for FPGAs which can produce optimized circuitry parameters of the FPGA towards a user-specified optimization object through SPICE simulations. A ready-to-use FPGA architecture file of VPR including circuitry parameters will be exported in the final stage of COFFE2. However, COFFE2 fails to model the 2-level MUX routing architecture, since the optimization strategy in COFFE2 is highly relevant with the CB-SB FPGA architecture. Section IV presents our approach to acquire optimized circuitry parameters of the FPGA architecture we seek to explore.

C. Related Work

Reference [5] questioned that the full crossbar in a cluster has a redundant degree of connectivity and proposed a 50% populated or sparser crossbar to save from 10 to 18% in area without degradation to the CPD. A fixed number of spare inputs are added to the cluster to offset the reduced routability introduced by sparse crossbars. This work justified the redundancy of connectivity in the routing fabric of FPGAs. In this paper, we jointly design LUT input MUXes and cluster input MUXes with even sparser connections to save more area and reduce the load for routing wires, which is also stated as the future work in [5].

Reference [9] proposed three types of IIBs: type-1 IIB is a 1-level crossbar which could be fully populated or sparsely populated, type-2 IIB is a 2-level crossbar with sparsely populated first level and fully populated second level (like VPR-style), type-3 IIB refers to a special kind of depopulated type-2 IIB that can be decomposed into disjoint type-2 IIBs. The results showed that in Actel's flashed-based implementation, 28% of the total routing area saving can be achieved by simply replacing type-1 IIB with type-3 IIB without changing other routing fabric. However, this work didn't take into account the delay changes introduced by IIBs, and the area estimation is based on Actel's flash technology which is not public to academia. Besides, the 2-level MUX topology is not deployed to the SB.

III. Two-level MUX Routing Architecture

A. Two-level MUX Topology

Fig. 2 presents the model of the 2-level MUX topology, which is deployed in both the SB and IB. For one IB or SB, there are M L1-MUXes with fan-in size of S. The input bandwidth of the IB or SB is equal to M which represents the number of distinct signals allowed to go through at the same time. For the IB, the number of L2-MUXes is equal to the number of all LUT input pins inside a CLB since we model CB MUXes (L1-MUX) and local MUXes (L2-MUX) as IB. For the SB, the number of L2-MUXes is equal to the number of routing wires being drived at one SB, assuming the uni-directional routing wires are used. We divide one IB (SB) into many groups, each of which is named as sub-IB (sub-SB). The dotted box in Fig. 2 represents one sub-IB or sub-SB. Within one sub-IB (sub-SB), L1-MUXes can only connect to L2-MUXes in the same sub-IB (sub-SB). The number of sub-IBs (sub-SBs) in one IB (SB) is defined as G. The fan-in pattern P indicates the fan-in composition in terms of routing wires direction and connected switch points.

In particular, we constrain L1-MUXes in the IB and SB to have small fan-in sizes to improve the input bandwidth and reduce the wire loads. On the one hand, small L1-MUXes will expand the input bandwidth when the number of fan-in sources is fixed. On the other hand, the loads of routing wires

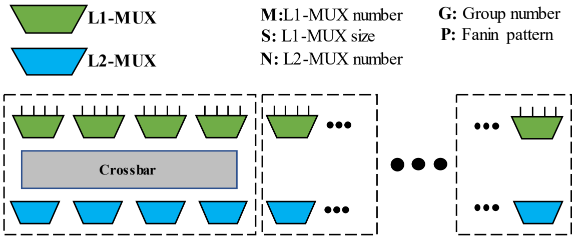

Fig. 2. The model of the 2-level MUX topology.

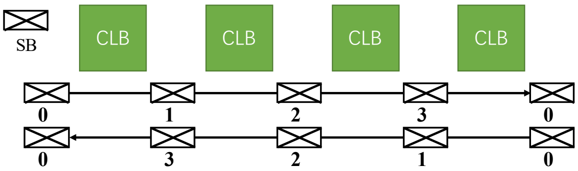

Fig. 3. A pair of uni-directional length-4 wires.

will be relieved when using small L1-MUXes to realize a fixed input bandwidth.

B. IB Design with 2-level MUX Topology

We use uni-directional wires in the routing channel which are proved efficient in the area and delay [13]. Fig. 3 presents a pair of uni-directional wires. There are switch points that are given index 0 at the start or end of a wire and incremented by 1 at each subsequent SB.

Fig. 4 illustrates a design example of the IB with uni-directional length-4 routing wires. The number between an uni-directional wires pair indicates the switch point of the routing wires at the IB. Routing wires with switch point 0 mean non-passing wires to the IB, while ones with switch point non-zero mean passing wires. The L1-MUXes in the IB have fan-ins that come from the routing wires in four routing channels relative to a CLB: North, South, West, East. There are four fan-in patterns in terms of the direction and switch point of the routing wires. For a single L1-MUX, if all its fan-ins come from the single direction and different switch points, we name this fan-in pattern single direction different length (SDDL) as is shown in Fig. 4. Similarly, there are other 3 fan-in patterns: single direction same length (SDSL), different direction different length (DDDL), and different direction same length (DDSL). In addition, the fan-in size of the L1-MUX is constrained to be 5 to accommodate signals from 4 routing channels and CLB feedbacks. IB input bandwidth is a design parameter which needs to be carefully designed to guarantee the routability. In section V, we will try to find the best routing pattern and explore IB input bandwidth.

L2-MUXes have fan-ins from the outputs of the L1-MUXes and have their outputs reached LUT input pins. In Fig. 4, L2-MUXes are presented by programmable points connecting to LUT input pins. All depicted L1-MUXes and L2-MUXes form one sub-IB in which L1-MUXes and L2-MUXes are fully connected. The whole IB is composed of a certain number of sub-IBs. Deciding the number of sub-IBs is also a design problem which will be investigated in section V.

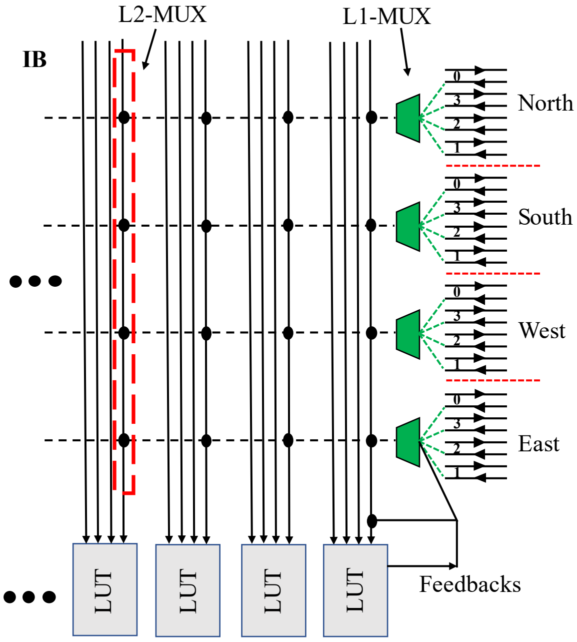

Fig. 4. Design example of the IB with uni-directional length-4 wires.

Besides, L1-MUXes and L2-MUXes in one IB could also route feedback signals from the CLB outputs. We experimentally choose the number of feedback signals to comfortably route all benchmarks. In this paper, feedback signals are only distributed into L1-MUXes of the IB.

C. SB Design with 2-level MUX Topology

Fig. 5 shows the design example of the SB with the 2-level MUX topology. The fan-ins of each L1-MUX are constrained from one direction among North, South, West, and East. This design rule can avoid connections between two routing wires heading opposite directions at the same routing channel. Based on this design principle, the fan-in patterns of the L1-MUXes will have two forms: single direction same length (SDSL) and single direction different length (SDDL). In Fig. 5, there are 4 L1-MUXes, each of which drives 3 L2-MUXes in the other 3 routing channels. These 4 L1-MUXes and 4 L2-MUXes form one sub-SB. The whole SB consists of a fixed number of sub-SBs which equals to the number of routing wires in one routing channel being drived at one SB (assuming horizontal and vertical channels have the same width). When the channel width is chosen, the numbers of L2-MUXes and sub-SBs are determined. The size of L1-MUXes is set to 4 which is effective in area and delay. With above design rules, we will explore SB input bandwidth in section V.

Besides, the SB also routes signals from CLB outputs. We refer to fc in the CB-SB FPGA architecture and assign

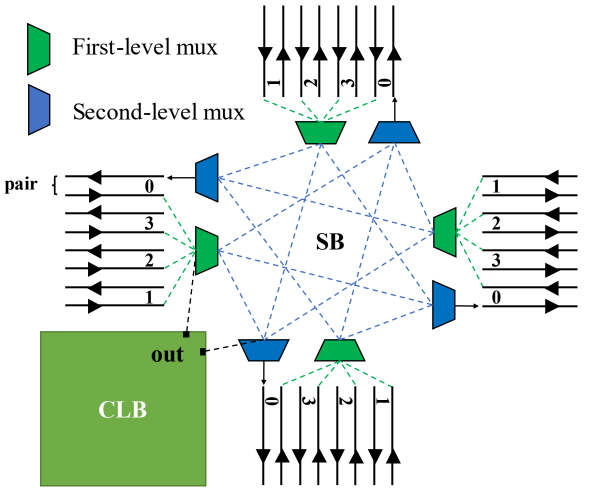

Fig. 5. Design example of SB with uni-directional length-4 wires.

similar number of CLB outputs to connect to the SB. Then these outputs are distributed evenly on the L1-MUXes and L2-MUXes.

D. IB and SB Generation

Algorithm 1 shows the approach to build the SB and IB with the 2-level MUX implementation. The design parameters are M, S, P, N, G of the SB and IB. Besides, the input constraints of the SB and CB must be clearly specified including fan-in names, fan-in numbers, and fan-in categories (for example, routing wires or feedbacks). In particular, the design parameters should be determined properly in order to avoid severely unbalanced design of the 2-level MUX topology. MUX sizes and fan-ins are partitioned as evenly as possible in order not to give any bias to any sub-SB or sub-IB. We believe this partitioning rule is good for diversity of routability and makes the design space manageable. The output is an XML file describing the detailed connection information of the SB and IB in a readable format, and then this generated file will be included into the extended VPR architecture file.

IV. CAD Tools and Parameters

A. RRG Generator

In VPR, RRG describes the routing resources within the FPGA. The presence of the 2-level MUX in the SB and IB changes the way of constructing RRG. To model the SB and IB in RRG, two low-cost segments are added to the original tag $\langle segmentlist \rangle$ in the VPR architecture file representing wires between L1 and L2-MUXes of the SB and IB. These two segments are abstracted as medium vertices in RRG, and programmable edges are used to connect medium vertices to those vertices representative of input and output pins of the CLB, and routing wire segments.

Algorithm 1 Approach to generate IB and SB

```
Input: M, S, P, N, G, and input constraints of the IB and SB.
Output: IB and SB connection pattern
Distribute M L1-MUXes and N L2-MUXes into G groups as evenly as possible.
Calculate the total fan-in number for L1-MUXes according to S*M.
Calculate the total fan-in number for L2-MUXes according to M and the number of signals directly connected to L2-MUXes specified in input constraints; Decide L2-MUX sizes.
Partition S*M fan-ins to G sub-IBs (sub-SBs) and assign them to each L1-MUX based on P.
Create a crossbar within each sub-IB (sub-SB) according to designs rules; Connect signals specified in input constraints to L2-MUXes.
```

B. Architecture Parameters

As discussed in section II-B, COFFE2 is not suitable to generate circuitry parameters of the FPGA architecture that is quite different from the CB-SB architecture. COFFE2 constructs the intra and inter cluster routing circuitry based on CB-SB and uni-directional routing architecture. In this paper, COFFE2 is modified to read in an RRG file exported from the VPR flow representative of the detailed routing information of the 2-level MUX routing architecture. By parsing the RRG file, the fan-in and fanout information of MUXes is extracted, which is then referenced by COFFE2 to build the routing circuitry of the 2-level MUX routing architecture. Since this paper focuses on routing architecture exploration, we don't change the circuitry of LUTs and FFs. One of the main contributions of COFFE2 is the analysis of the effect of wire loading at the interface between logic clusters and routing channels. The interface information of wire loading can be gained by parsing the OPIN and IPIN vertices in the RRG file. Therefore, the circuitry of a tile in the FPGA including the logic and routing circuitry can be modeled and is ready for the sizing optimization.

We leave untouched the Divide-and-Conquer and Inverter Rise-Fall Balancing techniques [11] in COFFE2 to avoid invalid results introduced by algorithm changes. The minimum-width transistor area (MWTA) model used to estimate area is retained, as well as the delay estimation through HSPICE simulations. The wire length estimates between L1-MUXes and L2-MUXes are equal to the length of the layout area of the L2-MUXes calculated by MWTA. The layout of a circuitry block is assumed square. With the above extensions, COFFE2 now can model the 2-level MUX routing architecture and provide the area and delay estimates required in the VPR architecture file.

V. Experimental Results

A. Baseline Architecture and Methodology

TABLE I lists the parameters of the baseline 2-level MUX routing architecture. We use uni-directional length-4 wires in

TABLE I
BASELINE ARCHITECTURE

Parameters	**Value**
CLB Size	Eight 6-input LUTs
Wire Length	4
Channel Width	160
DSP	36x36 Fracturable Multipliers
Memories	32Kb Block RAMs
Output Connections	160
Feedback Connections	80
Fan-in Patterns	SDSL for SB, SDSL for IB
Sub-SB Numbers	20
Sub-IB Numbers	8
SB Input Bandwidth	100
IB Input Bandwidth	80

the routing channel which is proved area and delay efficient in [14]. The channel width is fixed at 160 and is chosen based on the constraints that all 20 VTR benchmarks [15] can be routed successfully with moderate redundancy and equal to even times that of the routing wire length. Though minimum channel width is the typical indicator to compare FPGA routing architectures, we perform place and route experiments at the fixed channel width due to the fact that the minimum channel width method could hide the changes in segment usage [9]. The output and feedback connections are the numbers of CLB outputs to the SB and IB respectively. These two numbers are experimentally selected to guarantee the successful placement and routing for all the benchmarks. The default fan-in patterns for the SB and IB are both SDSL. There are 20 routing wires being drived in one direction of the SB so the number of sub-SBs is 20 (section III-C). The number of sub-IB is set to 8. The input bandwidth is set to 100 and 80 for the SB and IB respectively. Besides, the fan-in size of L1-MUXes in the SB is fixed at 4 and the one in the IB is set to 5 as mentioned in section III.

Since the IB and SB with arbitrary parameters are too general for analysis, we change only one design parameter at a time while keeping other parameters reasonable. Once the optimal value of a parameter is found, this value is fixed and we proceed to investigate other parameters. As for circuitry parameters of the underlying FPGA subcircuits, extended COFFE2 is called to perform transistor sizing optimization at 22nm technology node [16] when the FPGA circuitry changes significantly in terms of MUX sizes and the load of routing wires. Otherwise, for architectures with minor differences, we will reuse circuitry parameters.

For the exploration of each design parameter, we evaluate the architecture based on not only the CPD and the area, but also the segment usage reported by VPR. It is necessary to observe changes in segment usage of architectures because we route benchmarks at a fixed channel width. For architectures with fixed channel width, less segment usage to route all benchmarks indicates better routability. In the baseline architecture, segment usage exclusively means the utilization of all length-4 routing wires.

TABLE II
FAN-IN PATTERN EXPERIMENTAL RESULTS

Fanin Pattern	**Route Fails**	**CPD(ns)**	**Area(e+6)**	**Segment usage**
(DDDL,SDSL)	0	11.28	133.94	18.60%
(DDSL,SDSL)	0	11.07	133.94	17.89%
(SDDL,SDSL)	0	11.23	133.94	17.88%
(SDSL,SDSL)	**0**	**11.33**	**133.94**	**18.76%**
(DDDL,SDDL)	2	8.05	61.8	17.06%
(DDSL,SDDL)	7	5.20	25.17	11.19%
(SDDL,SDDL)	5	5.53	32.05	13.84%
(SDSL,SDDL)	0	11.18	133.94	18.65%

B. Searching for Fan-in Patterns

We combine 2 fan-in patterns [SDDL, SDSL] of the SB and 4 fan-in patterns [SDDL, SDSL, DDDL, DDSL] of the IB to produce 8 architectures with different fan-in patterns. TABLE II lists the geometric means of the CPD, area, and segment usage of all routed VTR benchmarks for each architecture. For example, (SDSL, SDDL) means SDSL fan-in pattern for the IB and fan-in pattern SDDL for the SB. The experimental result shows that SDDL is not the efficient fan-in pattern for the SB due to many route failures of benchmarks. The best fan-in pattern is (DDSL, SDSL) which achieves an average 2% improvement on the CPD and costs 5% fewer routing wires than the baseline architecture. The area of all architectures which successfully route all benchmarks is the same since the fan-in pattern is irrelevant with the area. In subsequent experiments, we use DDSL fan-in pattern for the IB and SDSL fan-in pattern for the SB.

C. IB Optimization

For the IB, the number of sub-IBs and the input bandwidth are two design parameters to explore. Fig. 6 plots the experimental result of 11 architectures with different number of sub-IBs normalized to the baseline architecture with 8 sub-IBs. As the number of sub-IBs increases, the area decreases and the segment usage increases both in an approximate linear way. But the segment usage changes more obviously than the area. The CPD has the optimal value when the number of sub-IBs is 6. In comprehensive consideration of the CPD, the area, and the segment usage, the best sub-IB number is 6 which has an average 5.8% shorter CPD and consumes 14.7% fewer routing wires with only a 2.4% increment in the area than the baseline.

Fig. 7 presents the experimental result of architectures with different IB input bandwidth normalized to the baseline with 80 IB input bandwidth. As the input bandwidth increases, the area increases and segment usage decreases, while the CPD doesn't show obvious change from 90 to 140. The saving of segment usage can't offset the area punishment, therefore, IB input bandwidth is maintained at 80 as it achieves the best results in terms of the CPD, area, and segment usage.

D. SB Optimization

For the SB, the numbers of L2-MUXes and sub-SBs are determined by the channel width as clarified in section III

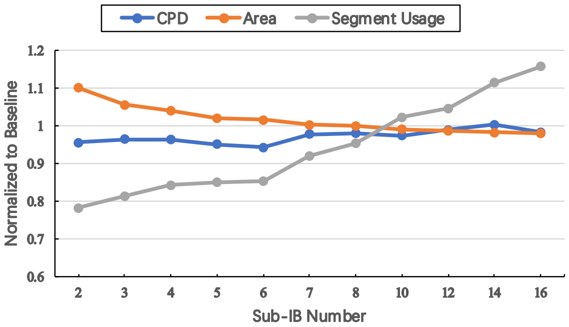

Fig. 6. IB exploration with different number of sub-IBs.

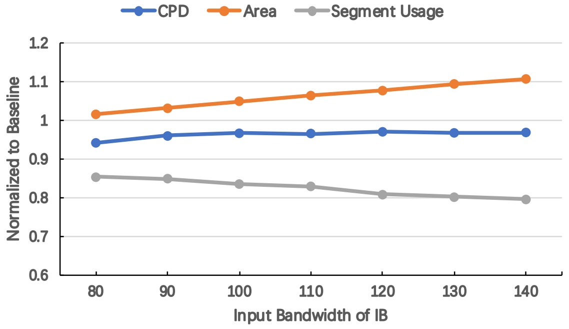

Fig. 7. IB exploration with different input bandwidth.

So we just optimize SB input bandwidth. Fig. 8 shows that the CPD and segment usage are insensitive to the change of SB input bandwidth. However, the area increases in a linear way as the input bandwidth increases. We do not present the results of architectures with less than 100 input bandwidth for the SB owing to some benchmark route failures. Therefore, SB input bandwidth holds at 100 and the conclusion can be drawn that it is not effective to improve architecture performance by adjusting SB input bandwidth for the baseline.

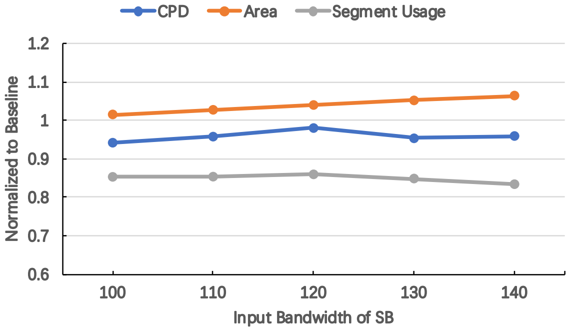

Fig. 8. SB exploration with different input bandwidth.

TABLE III
THE NUMBER AND SIZE INFORMATION OF ROUTING MUXES IN A TILE OF THE CB-SB ARCHITECTURE AND THE PROPOSED ARCHITECTURE

Architecture	MUX Name	Num	Size	Total Switch count
CB-SB	CB MUX	32	16	2432
	Local MUX	48	20	
	SB MUX	80	12	
The Proposed Architecture	IB L1-MUX	80	5	1820
	IB L2-MUX	48	13/14	
	SB L1-MUX	100	4	
	SB L2-MUX	80	4/5	

E. Comparison with CB-SB Architecture

To better illustrate the performance of the optimized 2-level MUX routing architecture, we compare it with the CB-SB architecture. For the CB-SB architecture, circuitry parameters are extracted from COFFE2 at the 22nm technology node with the same optimization objective as that of the baseline architecture. The CLB size, wire length, and the channel width are identical to the baseline architecture, as well as the DSP and memories configurations. The crossbar inside the CLB is 50% populated and the CLB input bandwidth is set to 32 based on the conclusion in [17]. The fc value is set to 0.1 by default.

TABLE III lists the number and size information of routing MUXes in one FPGA tile for the CB-SB architecture and the optimized 2-level MUX routing architecture in this paper. In the CB-SB architecture, there are less number of MUXes with wider fan-in size than those in the optimized 2-level MUX routing architecture. This results in more loads for routing wires in the CB-SB architecture. For example, there are 720 loads for all 320 routing wires passing or ending at the SB in CB-SB architecture (240 passing wires with 2 loads, 80 ending wires with 3 loads), while there are only 320 loads for 320 routing wires in the 2-level MUX routing architecture. It is worth noting that the fan-ins of the SB MUX and the SB L1-MUX are part from CLB outputs. Besides, the CLB input bandwidth is limited at 32 in the CB-SB architecture, while there is no limit to the 2-level MUX routing architecture whose CLB input bandwidth is set to 80. The total routing switch count of a tile is 2432 and 1820 for the CB-SB and 2-level MUX routing architecture. This indicates that 2-level MUX routing architecture has a sparser connectivity and introduces less load for routing wires than the CB-SB architecture. Fig. 9 illustrates the routing area breakdown of one FPGA tile for the CB-SB architecture and the optimized 2-level MUX routing architecture, which are reported from COFFE2. The routing area of the 2-level MUX routing architecture is a bit larger than the CB-SB architecture due to the fact that we use more routing MUXes resulting in more SRAM area. Fig. 10 presents the delay of routing MUXes generated from COFFE2. The result shows that the delay through SB L1-MUX and L2-MUX is about 74ps, while the delay through the SB is about 89ps. The delay through IB L1-MUX and L2-MUX is 64ps, which is also less than 118ps through CB MUX and local MUX. The delay savings of the 2-level MUX routing architecture are mainly caused by reduced load for routing wires.

TABLE IV
COMPARISON OF OPTIMIZED 2-LEVEL MUX ROUTING ARCHITECTURE AND THE CB-SB ARCHITECTURE

Benchmark	**CPD(ns)**			**Area(e+6)**			**Segment Usage**		
	CB-SB	***This Paper***	***Ratio***	***CB-SB***	***This Paper***	***Ratio***	***CB-SB***	***This Paper***	***Ratio***
arm_core	11.24	8.80	78.30%	99.3	97.90	98.60%	39.70%	36.00%	90.70%
bgm	11.08	8.35	75.40%	189.9	184.53	97.20%	33.60%	30.40%	90.50%
blob_merge	6.10	4.23	69.30%	48.64	48.87	100.50%	30.50%	23.80%	78.00%
boundtop	1.49	0.94	63.20%	4.92	5.56	113.00%	3.40%	2.70%	79.40%
ch_intrinsics	1.80	1.66	92.10%	4.36	4.94	113.30%	3.50%	2.80%	80.60%
diffeq1	17.69	16.24	91.80%	9.17	10.89	118.80%	14.40%	9.10%	63.30%
diffeq2	13.54	11.96	88.30%	9.17	10.89	118.80%	10.20%	6.50%	64.10%
LU8PEEng	50.03	41.67	83.30%	205.38	207.06	100.80%	32.00%	27.30%	85.30%
LU32PEEng	51.69	38.12	73.70%	690.24	688.70	99.80%	41.30%	35.20%	85.20%
mcml	46.30	36.17	78.10%	644.21	634.20	98.40%	20.30%	19.90%	98.00%
mkDelayWorker32B	4.66	4.74	101.70%	106.52	105.93	99.40%	2.30%	2.00%	86.70%
mkPktMerge	3.41	3.51	102.80%	31.30	32.42	103.60%	5.00%	4.10%	82.10%
mkSMAdapter4B	3.81	3.20	83.90%	15.50	16.11	103.90%	14.90%	10.50%	70.50%
or1200	9.78	7.75	79.30%	28.14	26.69	94.80%	27.40%	22.00%	80.30%
raygentop	3.96	3.95	99.60%	18.17	22.67	124.80%	15.60%	10.70%	68.60%
sha	8.28	6.04	73.00%	19.25	20.10	104.40%	22.90%	16.80%	73.40%
stereovision0	2.20	1.72	78.30%	99.31	100.45	101.10%	13.40%	10.80%	80.60%
stereovision1	4.32	4.09	94.60%	90.80	97.90	107.80%	25.30%	19.60%	77.50%
stereovision2	10.64	9.01	84.60%	324.64	402.13	123.90%	30.00%	25.50%	85.00%
stereovision3	1.54	1.19	77.00%	0.75	1.00	133.30%	6.90%	4.20%	60.80%
Avg.	**-19.01%**			**3.00%**			**-18.51%**		

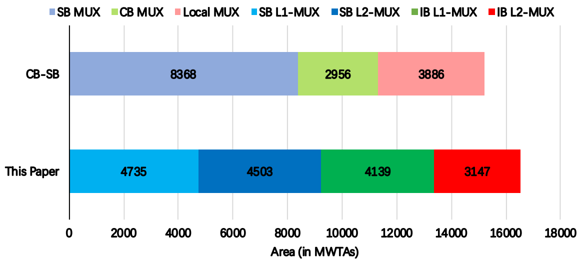

Fig. 9. Routing area breakdown in one FPGA tile.

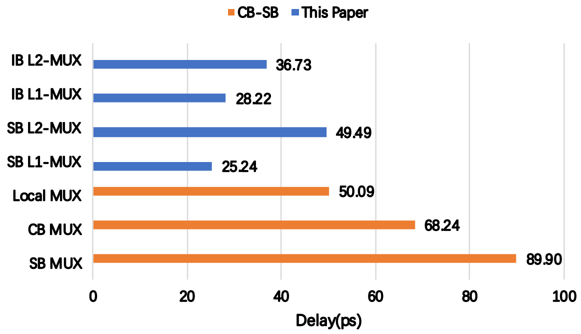

Fig. 10. Delay of routing MUXes.

TABLE IV lists the comparison of the CB-SB architecture and the optimized 2-level MUX routing architecture in terms of the CPD, area, and segment usage of all benchmarks. Results show that it can achieve 19% shorter CPD and consume 18% less routing wires on average than the CB-SB architecture at the cost of 3% area overhead. The CPD saving and the area punishment are predictable by referring to the Fig. 9 and Fig. 10. Besides, the 2-level MUX routing architecture has better routability than the CB-SB architecture by expanding the IB and SB input bandwidth. In [9], 2-level IIBs achieve the area saving at the cost of decreased routability, but this work did not perform delay analysis. In this paper, we tradeoff the delay and the area, while improving the routability.

VI. CONCLUSION AND DISCUSSION

In this paper, we propose a 2-level MUX routing architecture based on tailored MUX design. We deploy the 2-level MUX topology in the IB and SB, then optimize in terms of fan-in patterns, sub-IB number, IB input bandwidth, and SB input bandwidth. Compared to the CB-SB architecture, the optimized 2-level MUX routing architecture can achieve 19% improvement on the CPD and reduce the segment usage by 18% at the cost of 3% area overhead. The experimental results show that applying 2-level MUX topology enables a large design space for the delay, area, and routability. We conduct experiments based on some constraints such as L1-MUXes fan-in sizes in the IB and SB. Besides, the experimental results depend on CAD algorithms. In the future, we intend to explore more design parameters such as the fan-in size of L1-MUXes and determine the interaction between different design parameters. The CAD algorithms will be enhanced to better adapt to the 2-level MUX routing architecture.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China under Grant No. 61971143. Besides, We appreciate the authors in [9] which inspire us, and the open source work of VTR [10] and COFFE2 [11] which enable us to perform this research.

References

[1] V. Betz, J. Rose, and A. Marquardt, "Architecture and CAD for deep-submicron FPGAs," 1999.

[2] Y.-W. Chang, D. Wong, and C.-K. Wong, "Universal switch modules for FPGA design," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 1, no. 1, pp. 80–101, 1996.

[3] S. J. Wilton, "Architectures and algorithms for field-programmable gate arrays with embedded memory," Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Toronto, 1997.

[4] G. G. Lemieux, S. D. Brown, and D. Vranesic, "On two-step routing for FPGAs," in *Proceedings of the international symposium on Physical design*, 1997, pp. 60–66.

[5] G. Lemieux and D. Lewis, "Using sparse crossbars within LUT," in *Proceedings of the ACM/SIGDA ninth international symposium on Field programmable gate arrays*, 2001, pp. 59–68.

[6] X. Sun, H. Zhou, and L. Wang, "Bent routing pattern for FPGA," in *29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 9–16.

[7] K. Shi, H. Zhou, X. Zhou, and L. Wang, "GIB: A novel unidirectional interconnection architecture for FPGA," in *International Conference on Field-Programmable Technology (ICFPT)*. IEEE, 2020, pp. 174–181.

[8] M. B. Petersen, S. Nikolić, and M. Stojilović, "Netcracker: A peek into the routing architecture of xilinx 7-series FPGAs," in *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2021, pp. 11–22.

[9] W. Feng and S. Kaptanoglu, "Designing efficient input interconnect blocks for LUT clusters using counting and entropy," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 1, no. 1, pp. 1–28, 2008.

[10] K. E. Murray, O. Petelin, S. Zhong, J. M. Wang, M. Eldafrawy, J.-P. Legault, E. Sha, A. G. Graham, J. Wu, M. J. Walker *et al.*, "VTR 8: High-performance CAD and customizable FPGA architecture modelling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 2, pp. 1–55, 2020.

[11] S. Yazdanshenas and V. Betz, "COFFE 2: Automatic modelling and optimization of complex and heterogeneous FPGA architectures," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 1, pp. 1–27, 2019.

[12] J. Luu, I. Kuon, P. Jamieson, T. Campbell, A. Ye, W. M. Fang, K. Kent, and J. Rose, "VPR 5.0: FPGA CAD and architecture exploration tools with single-driver routing, heterogeneity and process scaling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 4, no. 4, pp. 1–23, 2011.

[13] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect," in *International Conference on Field-Programmable Technology*. IEEE, 2004, pp. 41–48.

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

[15] J. Luu, J. Goeders, M. Wainberg, A. Somerville, T. Yu, K. Nasartschuk, M. Nasr, S. Wang, T. Liu, N. Ahmed *et al.*, "VTR 7.0: Next generation architecture and CAD system for FPGAs," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 2, pp. 1–30, 2014.

[16] "Predictive technology model," Website, 2012, http://ptm.asu.edu/.

[17] G. Zgheib and P. Ienne, "Evaluating FPGA clusters under wide ranges of design parameters," in *27th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2017, pp. 1–8.