



HAL
open science

Efficient Multilevel Interconnect Topology for Cluster-based Mesh FPGA Architecture

Emna Amouri, Adrien Blanchardon, Roselyne Chotin-Avot, Habib Mehrez,
Zied Marrakchi

► **To cite this version:**

Emna Amouri, Adrien Blanchardon, Roselyne Chotin-Avot, Habib Mehrez, Zied Marrakchi. Efficient Multilevel Interconnect Topology for Cluster-based Mesh FPGA Architecture. ReConFig 2013 - International Conference on Reconfigurable Computing and FPGAs, Dec 2013, Cancun, Mexico. pp.1-6, 10.1109/ReConFig.2013.6732282 . hal-00987368v2

HAL Id: hal-00987368

<https://hal.science/hal-00987368v2>

Submitted on 11 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Multilevel Interconnect Topology for Cluster-based Mesh FPGA Architecture

Emna Amouri, Adrien Blanchardon, Roselyne Chotin-Avot, Habib Mehrez
LIP6 Laboratory, Universite Pierre et Marie Curie
4 Place Jssieu, 75252 Paris, France
Email: emna.amouri@lip6.fr

Zied Marrakchi
Flexras Technologies
153 Boulevard Anatole France
Saint-Denis 93200, France

Abstract—This paper presents an improved cluster-based Mesh architecture. This architecture has a depopulated intra-cluster interconnect, and presents a new hierarchical topology for the switch box which unifies a downward and an upward unidirectional networks. Experimental results of 20 MCNC benchmarks show that density is improved and interconnect area requirement is reduced by 42% compared to the cluster-based VPR architecture.

I. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are an attractive platform thanks to their low cost compared to full custom ASIC design, their short time to market period, and their reprogrammability. The interconnection architecture is one of the major research topics in FPGA design since it occupies up to 90% of the total area and is responsible of large part of the circuit delay.

Mesh FPGAs use clustering to reduce circuit area and to increase circuit speed. In these architectures, several Look-Up-Tables (LUTs) are grouped together, acting as a Configurable Logic Block (CLB). The clustering provides better performances specially for communication and enables to exploit signal sharing among LUTs. There exist different ways to connect signals to the LUTs inputs. In Xilinx Virtex architectures [1], the routing tracks are connected directly to the input muxes. In the VPR [2] and the Altera Stratix [3] architectures, the routing tracks are connected to the input muxes via a connection block. VPR-style interconnect has a sparsely populated connection block and a fully populated intra-cluster crossbar. The fully populated intra-cluster crossbar is simple and ensures a complete local routability, but it takes no advantage of the logical equivalency of LUT inputs and induces a significant area overhead. Lemieux and Lewis [4] proposed to improve the VPR-style interconnect by depopulating the intra-cluster crossbar. With this depopulation, an area saving from 10% et 18% was achieved. All these studies consider the connection block interconnect level and the intra-cluster crossbar separately. In [5], authors investigated joint optimization of both crossbars and proposed a new class of efficient topology. They achieve an area saving of 28%. Nevertheless, they design the cluster local interconnect and the switch box separately. In addition, in the intra-cluster crossbar they optimized only the part connecting external signals to LBs inputs, and they use a full crossbar to connect feedbacks (LBs outputs) to LBs inputs which can be very penalizing. Moreover, they do not consider

the input bandwidth limitation, like in VPR-style interconnect. They use a Rent's parameter equal to 1 which corresponds to the maximal input bandwidth [5]. This latter is the maximal number of distinct external signals allowed to go into a cluster at the same time. In [8] a novel tree based architecture is presented. This architecture contains a depopulated intra-cluster crossbar, and unifies two unidirectional networks: a predictable downward network based on the Butterfly-Fat-Tree topology, and an upward network using hierarchy. Compared to basic VPR Mesh architecture, 56 % of area saving was achieved with the tree architecture [8].

In this work, we propose to improve the cluster-based Mesh FPGA architecture in 4 ways. First, we optimize the intra-cluster interconnect topology by depopulating the intra-cluster full crossbar, using the approach of the tree architecture [8]. The cluster input bandwidth limitation is considered. Second, we use only single-driver interconnect based on unidirectional wires. In fact, it was shown in [6] that single driver interconnect has a good impact on density improvement. Third, we propose a new hierarchical topology for the switch box to connect channel tracks together and to connect inputs and outputs of CLBs to channel tracks. In fact, many studies [7], [8] showed that using hierarchical routing interconnect leads to better density. Finally, we merge the connection block with the switch box.

Section II describes the proposed cluster-based architecture. In section III we present the configuration flow used to place and route netlists on the cluster-based architecture. In the following section, based on the largest MCNC benchmarks implementation, we evaluate architecture routability and we compare its area efficiency to the VPR 5.0 Mesh architecture [13]. Finally, we conclude this paper.

II. ARCHITECTURE OVERVIEW

In this paper, we propose an efficient cluster-based architecture. This architecture contains clusters placed into a regular 2-dimensional grid. Each cluster contains local Logic Blocks (LB) and a depopulated switch block to connect them. Each LB consists of a 4-input Look-Up-Table (LUT) and a Flip-Flop (FF). Cluster is surrounded by a unidirectional routing network. The channel width is varied to fit the implemented netlist, but remains in multiples of 2 [6]. A unidirectional switch Box (SB) with a new hierarchical topology connects different tracks of vertical and horizontal channels together. Cluster inputs and outputs are connected with adjacent routing channels via adjacent Switch Boxes. In [8], authors presented a

tree architecture witch contains a downward network based on the Butterfly-Fat-Tree topology, and an upward network using hierarchy. They showed that the tree architecture reduced the interconnect area by 56% compared to VPR Mesh architecture. However, they noticed that this architecture is penalizing in terms of physical layout generation. It does not support scalability and does not fit with a planar chip structure. In this work, we are inspired by the tree topology. We propose first to use the depopulated intra-cluster interconnect used in the tree architecture. This interconnect contains a depopulated downward crossbar that connects cluster inputs to LBs inputs, and an upward crossbar that connects feedbacks to LBs inputs. In addition, inspired by the hierarchical topology of the tree architecture interconnect, we propose a new hierarchical interconnect of the switch Box. This interconnect is composed of a downward network and an upward network. The cluster-based Mesh architecture with the hierarchical Switch Boxes and the depopulated clusters can be seen as a distributed Tree architecture. The SB and the intra-cluster interconnect form a portion of a Tree interconnect with 3 levels of hierarchy (cf. fig. 3).

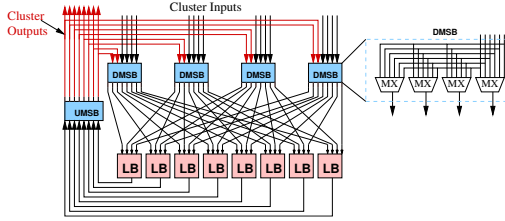


Fig. 1. Cluster Interconnect

A. Cluster Local Interconnect

Each cluster contains a switch block to connect local LBs. Figure 1 shows a cluster with Arity 8, which means that it contains 8 LBs. A switch block is divided into DMSBs and UMSBs (Downward and Upward Mini Switch Blocks). The interconnect unifies two unidirectional networks:

- 1) The downward network is based on the Butterfly Fat-Tree (BFT) topology, where Tree leaves correspond to logic blocks. It connects DMSBs to LBs inputs. As shown in figure 1, the number of DMSBs of the cluster is equal to the number of inputs of LB.

- 2) The upward network connects LBs outputs to cluster outputs and to the cluster DMSBs. The UMSB allows all LBs outputs of a cluster to reach all the DMSBs. Thus LBs positions inside the same cluster, are equivalent.

B. Mesh Routing Interconnect

In the Mesh interconnect we use only single-driver unidirectional wires. Each cluster is surrounded by 4 routing channels which are connected by Switch Boxes (SB). We do not use connection blocks in the Mesh to connect channel tracks to cluster inputs and outputs. The cluster inputs and outputs are equally distributed on the 4 sides and are directly connected to the 4 adjacent Switch Boxes. As shown in figure 2, SB inputs come from the 4 channel tracks and the 4 adjacent clusters outputs. SB outputs are connected to the 4 adjacent

horizontal and vertical channels, and to the 4 adjacent clusters. Thus, each cluster is connected to 8 neighboring clusters through adjacent Switch Boxes. Each SB output is driven by a multiplexer since we use a single-driver based interconnect. The figure 3 shows a detailed view of the interconnect of the SB highlighted in figure 2 and a global view of the 4 adjacent SBs (a,b,c,d). For more clarity, we separate inputs and outputs of the Switch Box. So, adjacent SBs are duplicated in figure 3. The SB interconnect has a hierarchical topology with 2 hierarchical levels, and contains a downward network and an upward network.

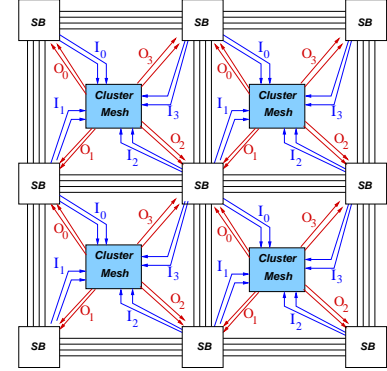


Fig. 2. Cluster-based Mesh architecture

1) *Switch BOX Downward Interconnect:* As described in figure 3, the SB downward interconnect is composed of DMSBs (Downward MSB) placed on 2 hierarchical levels. DMSBs of level $\ell = 2$ connect the inputs (I_a, I_b, I_c, I_d) coming from the adjacent SBs to the outputs (O_a, O_b, O_c, O_d) going to the adjacent SBs. These DMSBs ensure connections between SBs like a disjoint Switch Box. Each level 2 DMSB has inputs, each one coming from one adjacent SB. So, the number of inputs of a DMSB is equal to the number of adjacent SBs. Moreover, the number of DMSBs of $\ell 2$ is equal to the number of inputs coming from one adjacent Switch Box, which is equal to the half of the channel width since we use a unidirectional network. DMSBs of level $\ell 2$ are also connected to DMSBs of level $\ell 1$.

Each DMSB of level $\ell 1$ has inputs coming from DMSBs of level $\ell 2$ and has outputs connected to the adjacent clusters. Thus, SB inputs coming from 4 adjacent SBs are connected to the SB outputs going to the 4 adjacent clusters (O_0, O_1, O_2, O_3) through DMSBs of levels $\ell 1$ and $\ell 2$. As we said before, cluster inputs are equally distributed on the 4 sides. On each side, each input is connected to a level $\ell 1$ DMSB of the adjacent SB.

Thus, we have the following relations:

$$Nb_DMSB(2) = \frac{W}{2} \quad (1)$$

$$Nb_DMSB_inputs(2) = Nb_adj_SBs \quad (2)$$

$$Nb_DMSB(1) = \frac{Nb_In_Cluster}{4} \quad (3)$$

$$Nb_DMSB_inputs(1) = \frac{Nb_DMSB(2)}{Nb_DMSB(1)} \quad (4)$$

where $Nb_DMSB(\ell)$ is the number of DMSBs at level ℓ , $Nb_DMSB_inputs(\ell)$ is the number of inputs of a DMSB at

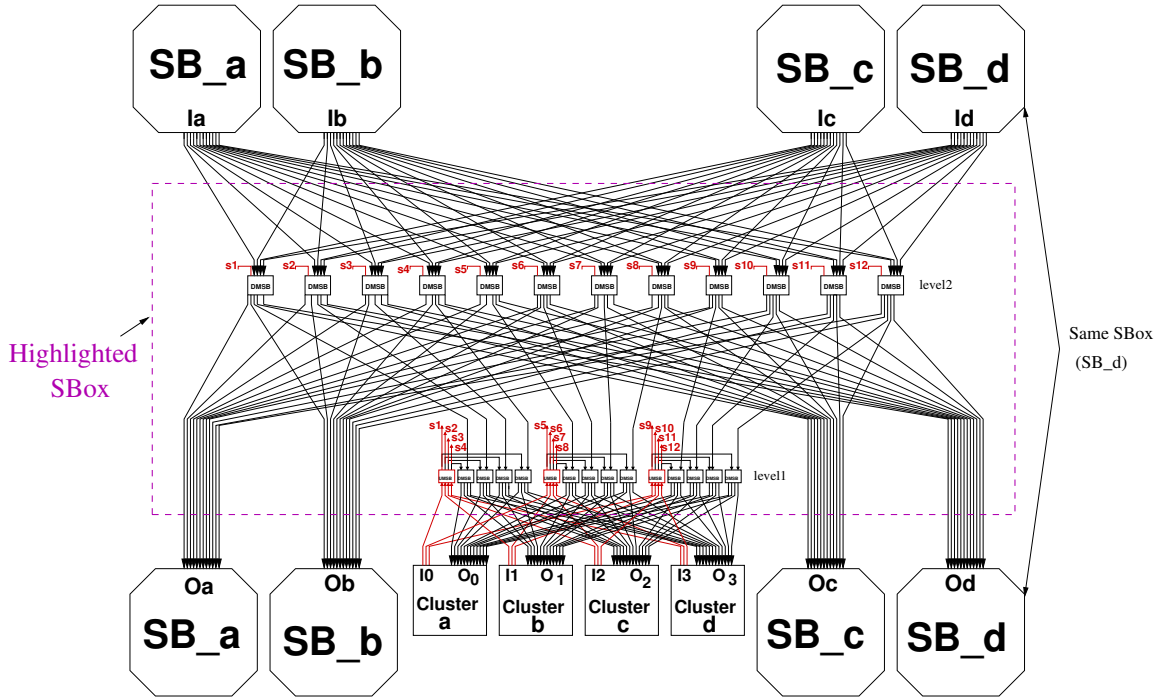


Fig. 3. Switch Box Interconnect

level ℓ , W is the channel width, Nb_adj_SBs is the number of adjacent SBs and $Nb_In_Cluster$ is the cluster inputs number. Numbers of DMSBs of levels 1 and 2 are independent.

2) *Switch BOX Upward Interconnect*: The SB upward interconnect is composed of UMDSBs (Upward MSB) which connect the inputs (I_0, I_1, I_2, I_3) coming from adjacent clusters outputs to the DMSBs of levels ℓ_1 and ℓ_2 . Thus, each cluster is connected to adjacent clusters through UMDSBs and DMSBs of the adjacent SB, and the outputs of a cluster connected to an adjacent SB can reach 4 adjacent routing channels. As shown in figure 3, each Switch Box UMDSB is connected to the outputs of 4 adjacent clusters, one output coming from each adjacent cluster. Since the cluster outputs are equally distributed on the 4 sides, the number of UMDSBs, UMDSB inputs and UMDSB outputs are given by:

$$Nb_UMDSB = \frac{Nb_Out_Cluster}{4} \quad (5)$$

$$Nb_UMDSB_inputs = Nb_adj_clusters \quad (6)$$

$$Nb_UMDSB_outputs = \frac{(Nb_adj_clusters \cdot \frac{Nb_Out_Cluster}{4})}{Nb_UMDSB} \quad (7)$$

where $Nb_Out_Cluster$ is the number of cluster outputs and $Nb_adj_clusters$ is the number of adjacent clusters to the SB.

C. Connection with Outside

Input and output pads are grouped into blocks, and are arranged at the periphery of the architecture. They are connected to the adjacent SBs. Thus, SBs which are placed at the periphery are connected to 2 adjacent clusters and 2 adjacent Input/output blocks. The figure 4 shows connections between pads and the adjacent SB. Each input pad of an input/output block is connected to all UMDSBs of the adjacent SB, and thus can reach the adjacent clusters and can be connected to adjacent routing channels. Output pads of an input/output

block are grouped into a specific cluster. They are connected to all DMSBs of level ℓ_1 of the adjacent SB. In this way they can be reached through different paths.

The number of Input and Output pads in blocks is represented by In_rate and Out_rate respectively.

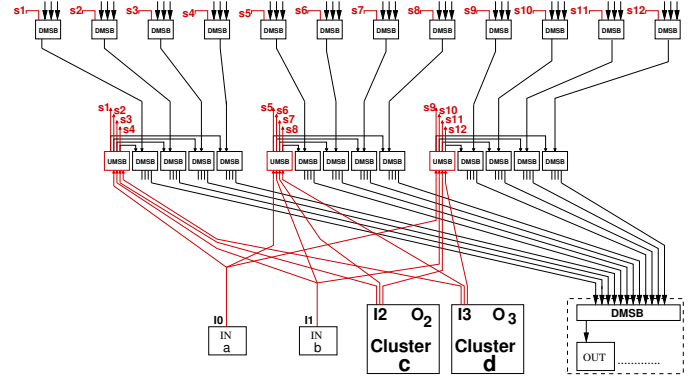


Fig. 4. Connection of peripheral Switch Box with IN/OUT Pads

D. Interconnect Flexibility Control

The number of inputs/outputs of a cluster can be less than or equal to the sum of inputs/outputs required by all the LBs in the cluster. Rent's rule [9] is applied to cluster architecture:

$$IO = c \cdot k^p$$

where k is the cluster arity, c is the number of inputs/outputs of a LB, IO the number of inputs/outputs of the cluster and p is the Rent's parameter. Intuitively, p quantifies the locality of interconnect requirements. If most connections are purely local and only few of them come in from the exterior of a local region, p will be small.

The interconnect flexibility is controlled by 2 parameters: the Rent's parameter and the channel width. Reducing the Rent's parameter and thus the number of inputs in each cluster induces a depopulation in the routing interconnect. For example, when we reduce inputs from 16 ($p=1$) to 8 ($p = 0.63$) for an architecture containing cluster with 4 LBs and 4 outputs, this induces a reduction from 4 to 2 of the number of DMSBs of level ℓ_1 in each Switch Box. In this case, if we consider a 3x3 cluster-based architecture with a channel width of 8, $In_rate = 2$ and $Out_rate = 2$, we get a reduction of the interconnect switches number from 2640 to 2304 (12%) and a reduction of 16% of the interconnect area. There is a strong interaction between the Rent's parameter and the channel width. In fact, if we increase the number of cluster inputs, we increase the routability and thus we can reduce the channel width. The reduction of the channel width induces the decrease in the number of DMSBs of level ℓ_2 in each SBox. By doing so the architecture routability is reduced too. Thus we have to find the best tradeoff between interconnect population and routability.

III. CONFIGURATION FLOW

To explore our architecture, we investigate the following configuration flow:

1) bottom-up clustering: The clustering consists in grouping N LBs together to form logic blocks clusters, meeting the constraint imposed on the number of cluster inputs I . We use the T-VPack tool [10] to achieve the clustering phase .

2) Clusters placement: Our placement tool uses the simulated annealing algorithm [2] to place the CLBs/IOs instances of the netlist on the CLBs/IOs blocks of FPGA. The objective of the placer is to minimize the sum of half-perimeter of the bounding boxes (BBX) of all the nets. The BBX of a net is a minimum rectangle that contains the driver instance and all the receiving instances of a net. The placer performs random swaps between different clusters, and updates the BBX after each move operation incrementally. Then, each LB is assigned to a random position inside its owner cluster, since LBs positions are equivalent.

3) Routing process: Interconnect resources of the architecture are presented by a routing graph with nodes corresponding to wires and CLBs/LBs pins and edges presenting switches. We use the Pathfinder routing algorithm [11] which is an iterative rip up algorithm based on the congestion negotiation. To connect terminals of each net, the router uses the Dijkstra algorithm to find the shortest path (lowest total cost) between a net source node and a net sink node. At the end of an iteration, a resource can be congested because it is used by multiple nets. During the subsequent iterations, the cost of resources is increased. The so-called congestion cost takes into account the number of nets sharing the resource (present congestion), and the congestion history of that resource. Therefore, nets are made to negotiate for the use of routing resources.

IV. EXPERIMENTAL EVALUATION

A. Architecture Optimization

In this section, our objective consists in searching for the most optimized architecture, by varying the netlist Rent's

parameter and the architecture Rent's parameter. The first parameter depends on the inputs/outputs number of a cluster in the netlist. This inputs number is used by the clustering tool. The more the netlist Rent's parameter decreases, the more the cluster inputs/outputs number decreases and the more the number of clusters of the netlist increases. The architecture Rent's parameter was explained in section II-D. It depends on the number of inputs/outputs of a cluster in the FPGA architecture. When we increase the architecture Rent's parameter, we increase the number of cluster inputs. This induces more routability, and can induce a reduction in the channel width. The architecture used here contains clusters with 8 LBs. In table I, we show the average architecture area and the average channel width for 20 MCNC benchmarks obtained with different Netlist and architecture Rent's parameters. We mean by "NR" that some or all circuits are not routable with the corresponding Netlist and architecture Rent's parameters. We calculate the area of the architecture using an estimation model of effective circuit area. The circuit area is the sum of its basic cells areas like SRAMs, buffers and Multiplexers. We use a cell symbolic library [14].

We notice that in all cases, architecture Rent's parameters are larger than Netlist Rent's parameters. This is due to the depopulated switch boxes topology in the cluster. The figure 5 shows the variation of the total architecture interconnect area with architecture Rent's parameter for different Netlist Rent's parameters. Each curve corresponds to a Netlist Rent's parameters. Results correspond to the average interconnect area of all the 20 MCNC circuits. For clarity, we don't show the curve corresponding to the netlist Rent's parameter 0.42. The architecture areas for this netlist Rent's parameter value are much more important than for the other netlist Rent's parameters because the number of clusters (43x43) is much more important. We can see also that the channel width remains the same for many architecture Rent's parameter ($w = 12$) but the architecture area increases. This is due to the increasing number of cluster inputs with the architecture Rent's parameter. On the other hand, we can note that for each Netlist Rent's parameter, there is a reduction in interconnect area until we reach a certain architecture Rent's parameter, from which the increase of architecture Rent's parameter leads to an increase in area. In fact, table I shows that the average channel width decreases when we increase the architecture Rent's parameter, which can explain the decrease of area. Nevertheless, high values of architecture Rent's parameter induces much more DMSBs in Switch Boxes, and the reduction of channel width becomes insufficient to reduce the overall area. On the other hand, we can see that the most optimized architecture corresponds to Netlist Rent's parameter equal to 0.67 and architecture Rent's parameter equal to 0.89, with an average area of $1208 \times 10^6 \lambda^2$.

B. Area Efficiency

To evaluate the proposed architecture, we place and route the largest MCNC benchmark circuits, and compare it to the VPR-style clustered Mesh architecture [13]. In both architectures we consider clusters arity 8 and LUT size 4. The VPR architecture uses a unidirectional routing network with single-length segments and a wilton switch block. Each cluster logic block contains I inputs and 8 outputs which are distributed over the cluster sides. We assume that I is equal to 18 inputs

TABLE I. AVERAGE AREA ($\times 10^6 \lambda^2$) AND CHANNEL WIDTH FOR 20 MCNC BENCHMARKS

Netlist's Rent	NxN	Avg. Area ($\times 10^6 \lambda^2$) vs Architecture's Rent									Avg. Channel width vs Architecture's Rent								
		0.42	0.56	0.67	0.83	0.89	0.95	1	1.08	1.16	0.42	0.56	0.67	0.83	0.89	0.95	1	1.08	1.16
0.42	43x43	NR	2477	2955	3327	3349	3587	3680	4017	4476	NR	14	12	12	12	12	12	12	12
0.56	22x22	NR	NR	1542	1507	1449	1529	1650	1704	1830	NR	NR	34	24	22	22	22	22	22
0.67	21x21	NR	NR	NR	1250	1208	1241	1254	1359	1441	NR	NR	NR	34	32	28	28	26	26
0.83	18x18	NR	NR	NR	NR	1245	1234	1292	1335	1346	NR	NR	NR	NR	44	38	34	34	32
0.89	18x18	NR	NR	NR	NR	1349	1300	1240	1220	1282	NR	NR	NR	NR	48	44	38	38	34
0.95	18x18	NR	NR	NR	NR	NR	1243	1241	1239	1277	NR	NR	NR	NR	NR	46	42	42	36

TABLE II. NETLISTS AND ARCHITECTURES CHARACTERISTICS

MCNC benchmarks				VPR Clustered Mesh cluster size 8			New Clustered Mesh cluster size 8		
Circuits Names	LUTs Number	IN Pads	OUT Pads	Arch $N_x \times N_y$	Occup %	Channel Width	Arch $N_x \times N_y$	Occup %	Channel Width
alu4	1522	14	8	15x15	84	50	16x16	74	30
apex2	1878	39	3	16x16	91	50	18x18	72	34
apex4	1262	9	19	13x13	93	46	15x15	70	34
bigkey	1707	263	197	15x15	94	34	15x15	94	24
clma	8383	61	82	33x33	96	72	36x36	80	46
des	1591	256	245	15x15	88	32	16x16	77	24
diffreq	1497	64	39	14x14	95	32	15x15	83	28
dsip	1370	229	197	15x15	76	38	15x15	76	20
elliptic	3604	131	114	22x22	93	58	23x23	85	36
ex1010	4589	10	10	25x25	91	62	28x28	73	36
ex5p	1064	8	63	12x12	92	44	14x14	67	34
frisc	3556	20	116	22x22	91	58	23x23	84	38
misex3	1397	14	14	14x14	89	46	15x15	77	30
pdv	4575	16	40	25x25	91	86	28x28	72	48
s298	1931	4	6	16x16	94	48	17x17	83	26
s38417	6406	29	106	29x29	95	44	30x30	88	30
s38584	6447	39	304	29x29	95	40	29x29	95	34
seq	1750	41	35	15x15	97	50	17x17	75	34
spla	3690	16	46	22x22	95	78	25x25	73	42
tseng	1047	52	122	12x12	90	30	12x12	90	24
Average	2963	66	88	19x19	91	46	21x21	79	32

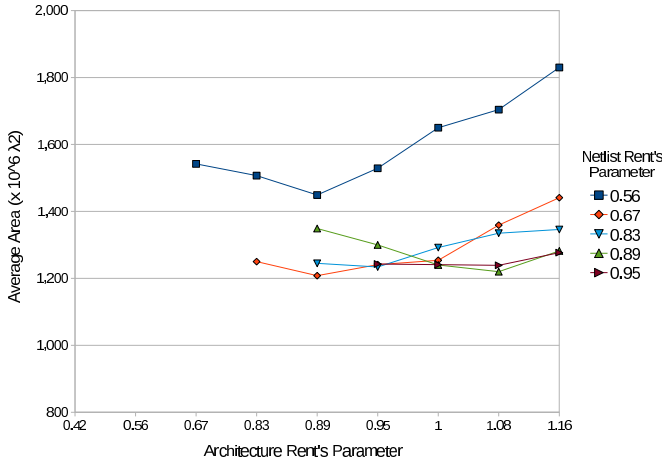


Fig. 5. Variation of Area with the Architecture Rent's parameter for different Netlist Rent's parameters (20 MCNC Benchs Avg.)

$(\frac{cluster_size \cdot Lut_size}{2} + 2)$. In fact, it was shown by Betz in [2] that this is sufficient to achieve full logic connectivity. LUTs pins are connected to cluster pins using a full local crossbar. Connection block population is defined by $F_{C_{in}}$ and $F_{C_{out}}$ parameters, where $F_{C_{in}}$ is routing channel to cluster input switch density and $F_{C_{out}}$ is cluster output to the routing channel density. $F_{C_{in}} = 0.5$ and $F_{C_{out}} = 0.25$ are chosen to be consistent with previous work [12].

For both architectures, we determine the smallest architecture

implementing every benchmark circuit. In the case of VPR Clustered Mesh architecture, we use T-VPack to construct clusters and the VPR placer and router (VPR 5.0) [13] to place and route circuits. VPR determines the optimal size as well as the optimal channel width W to place and route each benchmark circuit. In the case of our architecture, we use the configuration flow described in III. In this architecture, routability and switches number depend on three parameters: p (cluster Rent's parameter), N (number of LBs in the architecture), and the channel width W . The Netlist Rent's parameter is fixed to 0.67 and the architecture Rent's parameter is fixed to 0.89 (cluster inputs = 24), which corresponds to parameters of the most optimized architecture. N depends on the cluster arity (8 in this case) and the architecture size $N_x \times N_y$, where N_x and N_y are the array size in terms of logic blocks number. For each benchmark, we vary in both architectures N_x and N_y to find the smallest architecture size, and we search for the minimal channel width to find the architecture with the smallest area that can implement the netlist.

Table II shows the architecture size for each benchmark and the minimal channel width that can implement it. In table III, we observe that the new cluster-based Mesh architecture can implement circuits with lower switches number. In fact, an average of 41% reduction of the switches number is achieved. We achieve a 44% switches reduction in the case of the "tseng" smallest circuit and 40% in the case of the "clma" largest circuit. Thus the new cluster-based interconnect is attractive for both small and large circuits. We compare the areas of both architectures using the estimation model of effective circuit area explained in IV-A. In both architectures we use the same

TABLE III. NEW CLUSTERED MESH VS. CLUSTERED VPR MESH

MCNC	VPR Clustered Mesh Cluster size 8		New Clustered Mesh Cluster size 8		Gain	
	SW $\times 10^3$	Area (λ^2) $\times 10^6$	SW $\times 10^3$	Area (λ^2) $\times 10^6$	SW %	Area %
alu4	390	1246	216	613	44	50
apex2	444	1278	292	828	34	35
apex4	281	812	204	557	27	31
bigkey	332	979	171	486	48	50
clma	2328	6576	1386	3964	40	39
des	325	953	196	556	39	41
diffeq	277	815	183	520	33	36
dsip	349	1023	158	449	54	56
elliptic	915	2606	489	1393	46	46
ex1010	1226	3477	723	2060	41	40
ex5p	235	683	178	505	24	26
frisc	912	2600	505	1436	44	44
misex3	326	943	190	541	41	42
pdc	1491	4166	862	2468	42	40
s298	435	1256	226	642	48	48
s38417	1371	3978	749	2139	45	46
s38584	1312	3830	783	2148	40	43
seq	392	1277	261	742	33	41
spla	1085	3049	633	1803	41	40
tseng	200	588	111	313	44	46
Average	726	2106	425	1208	41	42

TABLE IV. MINIMUM CHANNEL WIDTH FOR VPR MESH (33x33) AND PROPOSED CLUSTERED MESH (36x36) TO IMPLEMENT ALL BENCHS

MCNC benchmarks	VPR Clustered Mesh 33x33	New Cluster-based Mesh 36x36
alu4	50	28
apex2	50	30
apex4	46	30
bigkey	34	22
clma	72	46
des	32	18
diffeq	32	24
dsip	38	20
elliptic	58	32
ex1010	62	36
ex5p	44	32
frisc	58	34
misex3	46	30
pdc	84	44
s298	46	26
s38417	44	28
s38584	40	28
seq	50	32
spla	78	38
tseng	28	22
Max. W	84	46
Max. Switches Nb.	2608	1382
Max. Area	7281	3954

cell symbolic library [14]. As presented in table III, we save 42% in the average area with the new cluster-based architecture compared to VPR clustered Mesh architecture.

To find the architecture that can implement all benchmarks, we place all circuits in the biggest array for both architectures (33x33 for the VPR Mesh and 36x36 for the proposed architecture), and we search for the minimal channel width that can implement each circuit. In table IV, we show the channel width for 20 MCNC benchmarks placed and routed in the biggest array. We can see that maximum channel width used is equal to 84 and 46 in the VPR clustered Mesh and in the proposed architecture respectively. So, the proposed architecture can implements all circuits with a gain of 45% in area compared to VPR clustered Mesh.

V. CONCLUSION

In this paper, we proposed an efficient cluster-based Mesh architecture. The cluster has a depopulated intra-cluster interconnect, which unifies a downward network to connect external inputs and feedbacks to LBs inputs, and an upward network to connect LBs outputs to external interconnect. The switch box interconnect has a new hierarchical topology and contains two unidirectional networks to connect channel tracks together and to connect CLB inputs and outputs to channel tracks. Based on the largest MCNC benchmark implementation, we showed that this architecture has better area efficiency than the VPR Style clustered Mesh. In fact, the area is decreased by 42%. On the other hand, compared to the Tree architecture [8] from which the proposed architecture is inspired, the total area is increased by 14%. This increase is compensated by the clustered Mesh layout generation simplicity compared to the Tree interconnect.

In a future work, we aim at studying the impact of the LUT size and the cluster size on architecture density and performances. We plan also to develop the layout of the architecture.

ACKNOWLEDGMENT

This work is a part of the project Robust FPGA ANR 11 INS-02, funded by The French National Research Agency, The Pole Systematic and The Pole Minalogic.

REFERENCES

- [1] Xilinx Corp, <http://www.xilinx.com>.
- [2] V. Betz, A. Marquardt and J. Rose, *Architecture and CAD for deep-submicron fpgas*, Kluwer Academic Publishers, 1999.
- [3] D. Lewis et al. *The StraiXTM routing and logic architecture*, Proc. of FPGA '03, pp. 12-20, Monterey, Calif, USA, February 2003.
- [4] G. Lemieux and D. Lewis, *Design of Interconnection Networks for Programmable Logic*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
- [5] W. Feng and S. Kaptanoglu, *Designing efficient input interconnect blocks for LUT clusters using counting and entropy*, Proc. of FPGA '07, pp. 2332, Monterey, Calif, USA, February 2007.
- [6] G. Lemieux et al., *Directional and Single-Driver Wires in FPGA Interconnect*, ICFPT 2004, Brisbane, Australia, pp. 41-48.
- [7] W. Tsu, et al, *HSRA: High Speed, Hierarchical Synchronous Reconfigurable Array*, Proc. of FPGA, pp. 125-134, February 1999.
- [8] Z. Marrakchi et al., *FPGA Interconnect Topologies Exploration*, Int. J. Reconfig. Comp, vol. 2009, 2009.
- [9] B. Landman and R. Russo, *On Pin Versus Block Relationship for Partition of Logic Circuits*, IEEE Transactions on Computers, vol. 20, pp. 1469-1479, 1971.
- [10] A. Marquardt, V. Betz, and J. Rose, *Using cluster-based logic block and timing-driven packing to improve FPGA speed and density*, International symposium on FPGA, Monterey, pp. 3746, 1999.
- [11] L. McMurchie and C. Ebeling, *Pathfinder: A Negotiation-Based Performance-Driven Router for FPGAs*, Proc. FPGA 95, 1995.
- [12] E. Ahmed and J. Rose, *The effect of LUT and cluster size on deep-submicron FPGA performance and density*, Proc. of FPGA, pp. 312, 2000.
- [13] J. Luu et al., *VPR 5.0: FPGA CAD and architecture exploration tools with single-driver routing, heterogeneity and process scaling*, ACM Trans. Reconfigurable Technol. Syst., vol. 4, pp. 32-55, Dec. 2011.
- [14] A. Greiner and F. Pecheux, *Alliance: A complete set of CAD tools for teaching VLSI design*, Eurochip Workshop, 1992.