Original scientific paper

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 46, No. 1(2016), 3 – 12

Exploration and optimization of a homogeneous Mesh of Clusters-based FPGA architectures

Sonda Chtourou[1], Emna Amouri[2] , Zied Marrakchi[3], Vinod Pangracious[2], Mohamed Abid[1] and Habib Mehrez[2]

[1]CES Research Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax, Tunisia
[2]LIP6 Laboratory, Universite Pierre et Marie Curie, Paris, France
[3]Flexras Technologies SAS, 102 Avenue Gaston Roussel Romainville, 93230, France

Abstract: This paper presents an efficient interconnect network for Mesh of Clusters (MoC) Field-Programmable Gate Array (FPGA) architecture. Compared to conventional MoC-based FPGA, proposed architecture improves the MoC-based interconnect in 2 ways. First, we optimize the intra-cluster interconnect topology by depopulating the intra-cluster full crossbar. Then, we propose a new multi-levels interconnect for the Switch Box (SB) which unifies a downward and an upward unidirectional networks based on the Butterfly-Fat-Tree (BFT) topology. The comparison with the common MoC-based VPR-Style shows that the proposed MoC-based architecture has better area and power efficiency. To optimize the interconnect flexibility of the proposed MoC-based FPGA, we explored and analysed the effect of different architecture parameters on performance, power consumption and density. Experimental results show that architecture parameters can be tuned and adapted to satisfy different specific applicative constraints. Results also show that cluster size 8 presents the best trade-off.

Keywords: FPGA; Mesh of Clusters FPGA architecture; Computer Aided Design (CAD) tools; Power estimation; Power analysis.

Raziskava in optimizacija homogene mreže FPGA arhitekture na osnovi grozda

Izvleček: Članek opisuje učinkovito povezovalno omrežje za mrežo grozdov FPGA arhitekture. Predlagana rešitev, glede na klasično FPGA strukturo na osnovi MoC, izboljšuje povezovanje na dva načina. Najprej, z razseljevanjem povezav, optimiziramo povezovanje med grozdi. Nato predlagamo novo večnivojsko Switch Box povezovanje, ki poenoti enosmerna omrežja na osnovi Butterfly-Fat-Tree (BFT) topologije. Predlagana topologija izkazuje boljšo izrabo prostora in izkoristek. Za optimizacijo povezav smo analizirali vpliv različnih arhitekturnih parametrov na učinkovitost, porabo in gostoto. Rezultati so pokazali učinkovito nastavljanje arhitekturnih parametrov za različne specifične aplikacije, pri čemer najboljšo rešitev predstavlja grozd velikosti 8.

Ključne besede: FPGA; mreža grozdov FPGA arhitekture; CAD; analiza moči; ocena moči

** Corresponding Author's e-mail: sonda.chtourou@ceslab.org*

1 Introduction

Compared to full custom ASIC design, FPGAs have become one of most attractive platforms since they enable a fast emulation of design alternatives and lead to a faster design cycle. However compared to ASIC, FPGA architectures are still facing serious challenges in term of performance, power and area due to their high programming overhead. To provide the required reconfigurable functionality, homogeneous FPGAs provide a large amount of programmable interconnect resources which occupy 90% of the total FPGA area [1]. Since FPGA area denotes one of main factors which control the manufacturing costs, reducing the silicon area of the programmable routing resources can lead to considerable improvement in manufacturing cost. In addition, FPGAs devices include a large number of prefabricated routing tracks and programmable switches allowing to route different placement solutions. These tracks can be long and consequently dissipate an important amount of energy every time they switch. Furthermore, the capacitance of programmable switches attached to each track is added which further increases

FPGA power consumption. Consequently, power consumption is becoming a major concern for FPGA vendors and customers. FPGA design big challenge is to find a good trade-off between flexibility and performance in terms of power consumption, area and delay.

2 Motivation and problem formulation

Modern Mesh FPGA architectures are based on a clustered architecture where several Look-Up-Tables (LUTs) are grouped together to act as a Configurable Logic Block (CLB). Experiments show that using these architectures allows exploiting signal sharing among LUTs and then improving overall performance of the FPGA [2]. The best characterization to date which reliably estimates interconnect requirements is Rent's Rule [3]. Rent's rule can be applied as follows to MoC-based cluster architecture: $IO = c*k^p$ where IO is the number of inputs/outputs of the cluster, c is the number of inputs/outputs of a Logic Block (LB), k is the cluster arity and p is the Rent's parameter. Intuitively, p quantifies the locality of interconnect requirements. It has small value when most connections are purely local and only few of them come in from the exterior of the cluster. We can distinguish two families of MoC-based FPGAs [4,5,6]: fully populated and depopulation intra-cluster crossbar. A VPR-style interconnect [4] has a sparsely populated Connection Block (CB) and a fully populated intra-cluster crossbar with low Rent's parameter. 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. An improved VPR-style interconnect was proposed by Lemieux and Lewis [5] by depopulating the intra-cluster crossbar. This depopulation achieves an area saving of 18%. However, all these studies consider the CB interconnect level and the intra-cluster crossbar separately. An improved VPR-style topology was proposed by Feng [6]. He investigated joint optimization of CBs and intra-cluster crossbars depopulation while using a high Rent's parameter (p = 1). He achieved an area saving of 28%. However, he optimized only connection of external signals to LB inputs and kept the use of a full crossbar to connect feedbacks (LB outputs) to LB inputs, which can be very penalizing. In addition, he did not experiment neither new Switch Boxes (SBs) topologies nor lower clusters Rent's parameter.

In this paper, we propose an improved MoC-based architecture with new hierarchical SB and depopulated intra-cluster interconnect based on the Butterfly-Fat-Tree (BFT) topology with flexible Rent's parameter. Based on analytical method, we identified architecture parameters that control the interconnect flexibility of the proposed MoC-based FPGA. Then, we explored how theses parameters interact in order to balance different trade-offs and satisfy application constraints. A set of CAD tools including metric model to map circuits on the proposed architecture is developed to explore efficiency in terms of power consumption, area and delay. The remaining of this paper is organized as follows. Section 3 presents the proposed MoC-based FPGA architecture. Section 4 presents the used exploration methodology. Section 5 details the experimentation platform and used metric model to estimate power consumption, area and delay. Experimental results are discussed in sections 6 and 7.

3 Proposed MoC-based FPGA architecture overview

Inspired from the Tree topology [7], we propose an improved MoC-based architecture with new hierarchical SBs and depopulated intra-cluster interconnect. This architecture is a mesh of clusters placed into regular 2D grid.

3.1 Cluster architecture

The cluster architecture contains local LBs connected with a depopulated local switch block. Figures 1.a and 1.b illustrate an example of cluster with respectively 8 and 4 LBs. Each LB consists of a 4-input LUT and a Flip-Flop (FF). The depopulated local switch block is divided into Mini Switch Blocks (MSBs). The local interconnect is composed of a downward network and an upward network. The downward network is based on the BFT topology which connects Downward MSBs (DMSBs) outputs to LBs inputs. Each DMSB connects each LB in only one input and hence the number of DMSB is given by equation (1) and the number of DMSB outputs is given by equation (2). The upward network connects LB outputs to an Upward MSB (UMSB) and allows all LBs outputs to reach all DMSBs and cluster outputs. Thus, LBs inside the same cluster are equivalent and their ordering has no impact on routing quality. The number of DMSB inputs is given by equation (3). The number of UMSB inputs and outputs are given respectively by equation (4) and equation (5).

$$Nb\_DMSB(CLB) = Nb\_inputs(LB) \quad (1)$$

$$Nb\_DMSB\_outputs(CLB) = Cluster\_size \quad (2)$$

$$Nb\_DMSB\_inputs(CLB) = \frac{Nb\_inputs(CLB) + Nb\_UMSB\_outputs(CLB)}{Nb\_DMSB(CLB)} \quad (3)$$

$$Nb\_UMSB\_inputs(CLB) = Cluster\_size \quad (4)$$

$$Nb\_UMSB\_outputs(CLB) = Cluster\_size \quad (5)$$

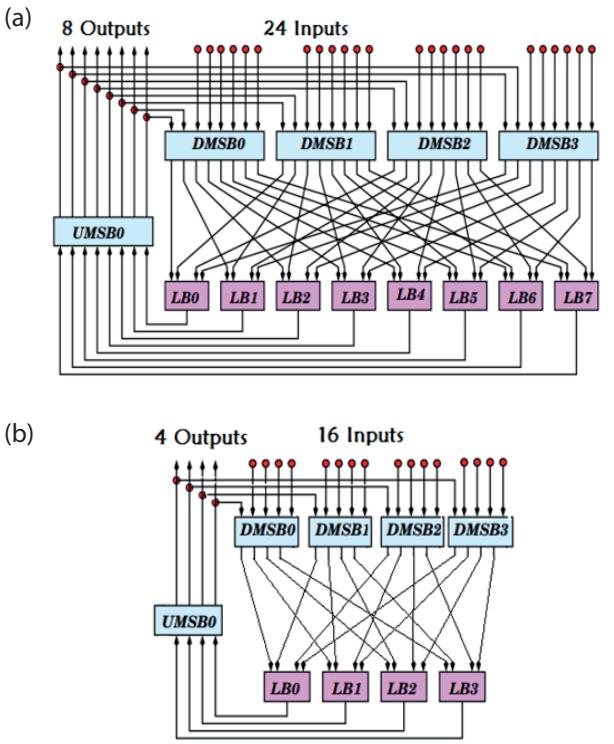

Figure 1: Cluster interconnect with different arity factors. a. Cluster arity 8 (Rent's parameter = 1 and Clusters Inputs = 24). b. Cluster arity 4 (Rent's parameter = 1 and Clusters Inputs = 16).

3.2 Mesh routing interconnect

Mesh routing interconnect of basic VPR MoC-based FPGA architectures use SBs and CBs to assure different interconnections. In fact, SBs are used to assure connection between horizontal and vertical adjacent routing channels and CBs are used to connect channel tracks to cluster inputs and outputs. In the proposed mesh routing interconnect, a new multi-levels interconnect of the SB is proposed to assure connection between horizontal and vertical adjacent routing channels and also between clusters inputs/outputs and adjacent routing channels. As illustrated in Figure 2, each cluster is surrounded by 4 unidirectional routing channels and 4 SBs. Each cluster is connected to 8 neighbouring clusters through adjacent SBs. Figure 3 shows a detailed view of the interconnect of a SB and a global view of the 4 adjacent SBs (Top, Bottom, Right, Left) and the 4 adjacent clusters (A, B, C, D) highlighted in Figure 2. Similarly to cluster, the SB has a multilevel topology including 3 main Boxes organized as follows:

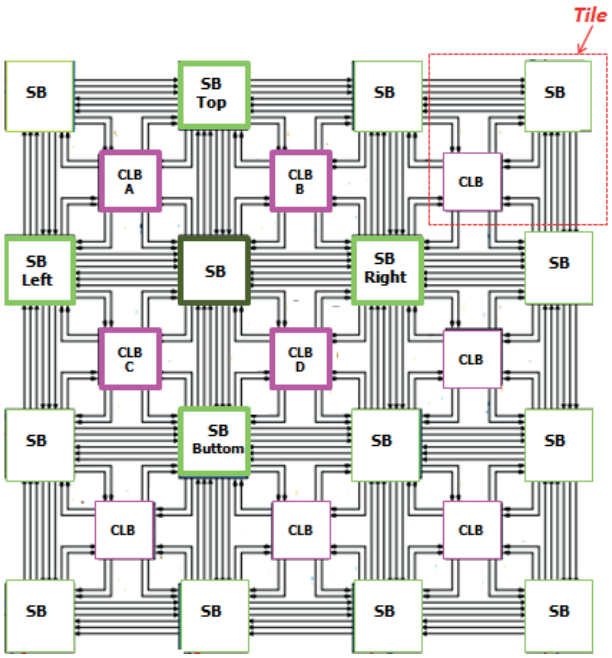

Figure 2: MoC-based FPGA architecture: Unidirectional interconnect.

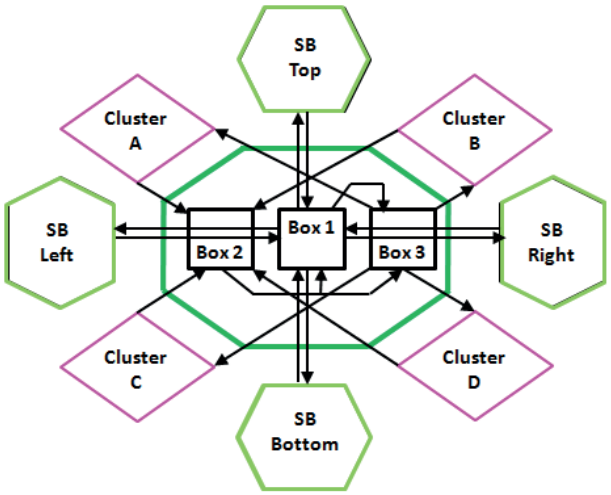

Figure 3: Multilevel SB interconnect.

SB to SB: Each SB is connected to the 4 adjacent SBs using global wires through Box 1. Box 1 is composed of MSB each one drives only one track in each 4 neighbouring channels. This topology is similar to VPR disjoint SB [4]. The number of MSB in Box 1 is given by equation (6):

$$Nb\_MSB(Box\_1) = \frac{Channel\_width}{2} \quad (6)$$

SB to Cluster: Each SB is connected to the 4 neighbouring clusters through 2 interconnect levels. Outputs of MSB located at Box 1 drive MSB located at Box 3 whose outputs drive 1 input of each of the 4 neighbouring clusters. Since the cluster is connected to the 4 neigh-

bouring SBs, the number of MSB located at Box 3 is given by equation (7).

$$Nb\_MSB(Box\_3) = \frac{Nb\_inputs(CLB)}{4} \quad (7)$$

The number of outputs per MSB in Box 3 is given equation (8):

$$Nb\_MSB\_ouputs(Box\_3) = 4 \quad (8)$$

The number of MSB outputs in Box 1 is given by equation (9):

$$Nb\_MSB\_ouputs(Box\_1) = 1 + Nb\_ajd\_SBs = 5 \quad (9)$$

Cluster to Cluster: Each cluster is connected to the neighbouring clusters through 2 interconnect levels. Cluster outputs drive MSB located at Box 2 whose outputs drive MSB located at Box 3. Since the cluster is connected to the 4 neighbouring SBs, the number of MSB located at Box 2 is given by equation (10):

$$Nb\_MSB(Box\_2) = \frac{Nb\_outputs(CLB)}{4} \quad (10)$$

The number of inputs per MSB in Box 2 is given equation (11):

$$Nb\_MSB\_inputs(Box\_2) = 4 \quad (11)$$

The number of inputs per MSB in Box 3 is given equation (12):

$$Nb\_MSB\_inputs(Box\_3) = \frac{2 * Channel\_width}{Nb\_inputs(CLB)} \quad (12)$$

Cluster to SB: Outputs of MSB located at Box 2 drive MSB located at Box 1 whose outputs drive 1 input of each of the 4 neighbouring SBs. Therefore, cluster outputs which drive Box 2 are connected to 4 neighbouring SBs through 2 interconnect levels. The number of outputs per MSB in Box 2 is given by equation (13).

$$Nb\_MSB\_outputs(Box\_2) = \frac{2 * Channel\_width}{Nb\_outputs(CLB)} \quad (13)$$

The number of inputs per MSB in Box 1 is given by equations (14).

$$Nb\_MSB\_inputs(Box\_1) = 1 + Nb\_ajd\_SBs = 5 \quad (14)$$

3.3 Clock network

The clock network is modelled as H-Tree distribution network, similar to the topology used in Xilinx Virtex II Pro [8]. The clock network contains buffers separated by a distance equal to the size of a tile (cluster and SB) in the FPGA (see Figure 2).

4 Exploration methodologies

4.1 Analytical model

To identify architecture parameters that control the flexibility of proposed MoC-based FPGA, we evaluate switches requirement. We consider a MoC arranged in a N x N array with k cluster size and W channel width. The total switch number in the MoC-based FPGA is given by equation (15):

$$Nb\_switch = Nb\_switch(SB) * (N+1)^2 + Nb\_switch(CLB) * N \quad (15)$$

Since MSBs are full crossbar, the number of switch per SB is given by equation (16):

$$Nb\_switch(SB) = \sum_{i=1}^{i=3} Nb\_switch(Box\_i) = 16 * W \quad (16)$$

The number of switch per CLB is given by equation (17):

$$Nb\_switch(CLB) = Nb\_switch(DMSBs) + Nb\_switch(UMSB) \quad (17)$$

Since DMSBs and UMSB are full crossbar, the number of switch in DMSBs and UMSB are given respectively by equation (18) and equation (19):

$$Nb\_switch(DMSBs) = Nb\_inputs(CLB) * k + k \quad (18)$$

$$Nb\_switch(UMSB) = k^2 \quad (19)$$

Based on Rent's rule, we have: Nb_inputs (CLB) + Nb_outputs (CLB) = $c*k^p$. Where c is the number of inputs/outputs of a LB and p is the Rent's parameter. As Nb_outputs (CLB) = k, the number of CLB inputs can be modeled by equation (20):

$$Nb\_inputs(CLB) = c * k^P - k \quad (20)$$

As consequence, the total number of switch in the FPGA can be modeled with the equation (21):

$$Nb\_switch = 16 * W * (N+1)^2 + (c * k^{p+1} + k) * N^2 \quad (21)$$

As illustrated in equation (21), the interconnect flexibility of the proposed MoC-based FPGA is controlled by the following architecture parameters: Rent's parameter (p), cluster size (k), channel width (W), LB inputs/outputs (c) and matrix size (N). Rent's parameter and cluster size control local clusters interconnect whereas channel width controls external clusters interconnect.

4.2 Experimental comparison

Rent's rule provides an empirical estimation of switching requirements. Nevertheless, this is not sufficient

since it does not give accurate information about interconnect routability. The best way to verify this point is to implement different benchmark circuits with CAD tools and explore the effect of architecture parameters described in equation (21) on performance. Performance of each solution is evaluated in terms of power consumption, required area and clock frequency.

5 Design and implementation methodology and metric models

We investigate the CAD flow illustrated in Figure 4 to explore proposed architecture. First, the circuit passes though T-VPack tool [9] to achieve the packing phase which consists in grouping N LBs together to form CLBs. Once packing is completed, we use the simulated annealing algorithm [4] to place the CLBs and IOs instances of the circuit on the CLBs and IOs blocks of FPGA. Then, we use the PathFinder [10] to successfully route all nets in a circuit. In our approach, we determine the minimum number of the channel width (Wmin) that routes the circuit, we decrease continuously the tracks number per channel and route the circuit until it fails. Once the Wmin is determined and routing phase is achieved, we estimate resulting power consumption, area and delay with developed models.

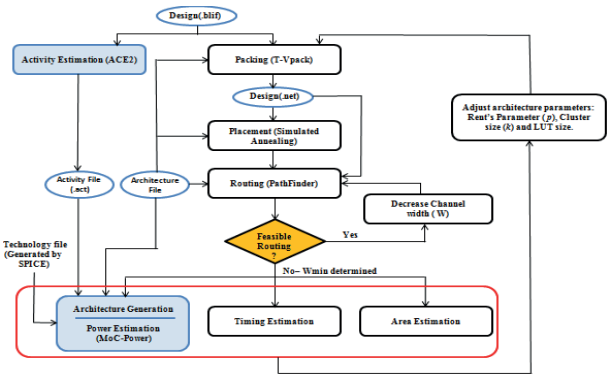

Figure 4: Proposed MoC CAD flow.

To estimate power consumption, we integrated in the proposed CAD flow activity estimator ACE2 and MoC-Power modules. The first module, the activity estimator (ACE2) [11], employs a transition density method to determine the switching density of all nets. The second module, the MoC-Power, estimates the power consumption at transistor level. It incorporates two components: architecture generator and Low-Level Power Estimation. The architecture generator uses the routing resource graph to decompose the entire MoC-based FGPA circuit into low-level components which are inverters, multiplexers and wires using same assumptions of VersaPower model [12]. Then, the Low-Level dynamic and static powers of each component is estimated as defined in [12].

Used area model is based on transistor-counting function consistent with the methodology used in [12] to compute components area. We use routing graph resource to parse all FPGA components (MSBs, LBs and buffers) to accurately compute the total number of transistor in the FPGA. The area is expressed as function of λ which is equal to the half of the minimum distance between source and drain of transistor.

Timing analysis allows evaluating performances of a circuit implemented on a FPGA in term of functional speed. The length of local wires is determined by approximating the size of entities through transistor-counting functions. In addition, proposed model accounts also the effect of the resulting LUT delay as a function of the LUT size to be consistent with experimentation done in [2]. Wire LUT and switch (crossbar, multiplexer) delays are extracted from the SPICE circuit simulator using ST Micros 130nm Technology node.

6 Comparison with basic VPR MoC-based FPGA architectures

In this section, we perform detailed exploration comparison between basic VPR and proposed MoC-based FPGA architectures. In this experimentation, we used largest MCNC [13] and IWLS [14] benchmarks presented in Table 1. The size of these circuits ranges from 1064 to 10437 primitives (LBs, FFs etc). We used VPR7.0 [15] and proposed MoC configuration CAD tools while using same packing options and same place and route algorithms. For both architectures, we determined the smallest architecture implementing every benchmark circuit and we considered homogeneous architecture with $k = 8$, LUT size 4, unidirectional routing network with single length segments and a disjoint switch block. VPR cluster architecture contains 18 inputs and 8 outputs to achieve full logic connectivity [4] and CB population is defined by Fcin and Fcout parameters, where Fcin is routing channel to cluster input switch density and Fcout is cluster output to routing channel density. Fcin and Fcout are chosen equal to respectively 0.5 and 0.25 to be consistent with work presented in [2]. For the proposed MoC-based FPGA, the cluster architecture contains 32 inputs (p=1). Table 1 illustrates comparison results in term of power, area, critical path delay, Wmin and total buffer used to drive global wires. Table 1 shows that the proposed MoC-based architecture can implement all circuits with lower Wmin than VPR architecture. In fact, the maximal Wmin used is equal to 86 and 56 in respectively the VPR and the pro-

posed MoC-based architecture. This induces an average reduction of 24% in total buffer used to drive global wires. Most of the power and area savings (respectively 30% and 32%) obtained in the proposed MoC-based FPGA are due to the reduction in total buffer number. We also note that the power and area gains are achieved at the cost of an increase in critical path delay by an average of 6%.

Table 1: Comparison results: VPR and proposed MoC-based FPGA.

Circuit	Primitives	VPR MoC					Proposed MoC					Gain		
		Power nW	Area E+6(λ)^2	Delay ns	Total Buff	W_{min}	Power nW	Area E+6(λ)^2	Delay ns	Total Buff	W_{min}	Power %	Area %	Buff %
Clma	8416	264	2560	31.51	175032	78	170.3	1593	45.56	116688	52	37	38	33
Elliptic	4726	100.8	970	24.93	52624	60	71.2	650	32.18	44528	44	29	33	15
ex1010	4898	142.2	1368	26.95	75400	70	86.95	810	29.85	57600	48	39	41	24
ex5p	1064	36.1	272	13.55	16224	54	26.39	211	15.45	15600	50	27	22	4
Frisk	1397	110.4	1204	24.24	62744	68	72	666	31.86	46552	46	35	45	26
misex3	4575	46.2	352	13.75	18480	50	31	246	15.73	15960	38	33	30	14
Pdc	1930	170.1	1601	30.29	96200	86	96	886	38.63	67200	56	44	45	30
s298	6406	38.3	345	28.62	19584	34	32.41	295	28.65	17408	32	15	14	11
s38417	6447	168.7	1501	22.80	83520	52	109.8	986	26.35	59160	34	35	34	29
s38584	6537	188.9	1479	17.30	87000	50	126.4	1266	19.57	59160	34	33	14	32
USB_ Funct	5293	127.5	1344	41.51	88400	68	102.0	905	20.68	65000	50	20	33	26
B22_C	10437	429	4632	41.51	291648	62	376.3	3179	48.91	206976	44	12	31	29
AVA2	9378	227.9	2406	41.51	160512	76	158.9	1507	42.34	109824	50	30	37	32
AES_ core	8416	243	2293	41.51	142800	60	177.4	1592	49.46	104720	44	27	31	27
Average												30	32	24

7 Architecture optimization

The flexibility of proposed MoC-based FPGA depends on architectures parameters described in equation (21). The effect of these parameters is not predictable and can only be determined by experimentation. In this section, we explore and analyse how theses parameters interact in order to balance different tradeoffs. Results correspond to interconnect area and the average power consumption and delay of MCNC benchmarks using the biggest FPGA array and Wmin which implement all circuits. Benchmarks are packed, placed and routed through the CAD flow with power, delay and area model described in section 5. For all experimentations, wire capacitances were obtained from the ITRS Interconnect Roamap [16]. The used transistor technology is 130nm PTM models [17]. Table 2 illustrates the variation of the FPGA size with k and also the variation of the Wmin and total buffer obtained with k and p. The Wmin and total buffer decrease when we increase p.

Figure 5 shows the variation of the total power consumption with p for different k. For all k values, the power consumption is reduced until we reach a certain p (the break point p=1) from which the increase of p induces an increase in power consumption. In fact, at this point, the reduction of Wmin with the increase of p becomes insufficient to route circuits (see Table 2). We can also see that Wmin remains the same for many p but the architecture area increases. This is due to the increasing number of cluster inputs with p.

Table 2: FPGA size, channel width and total buffer.

Cluster Size (k)	FPGA Size ($N^x N$)	Channel Width (W_{min}) vs. Rent's parameter (p)						Total Buffer vs. Rent's parameter (p)					
		0.83	0.89	1	1.05	1.09	1.16	0.83	0.89	1	1.05	1.09	1.16
4	49x49	NR	30	30	30	30	30	NR	147000	147000	147000	147000	147000
8	36x36	48	46	44	44	44	44	127872	122544	117216	117216	117216	117216
16	28x28	NR	60	58	58	58	56	NR	97440	94192	94192	94192	90944

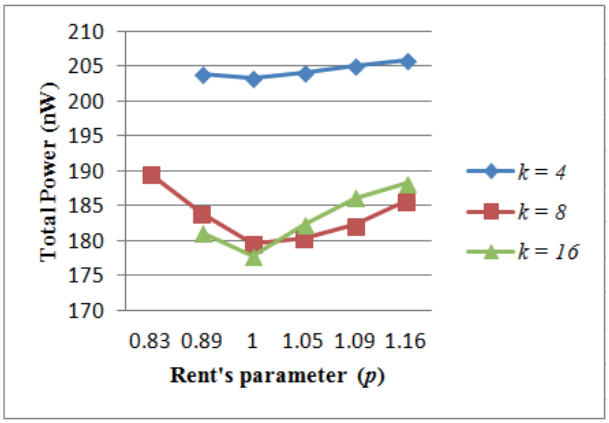

Figure 5: Total power vs. Rent's parameter and cluster size.

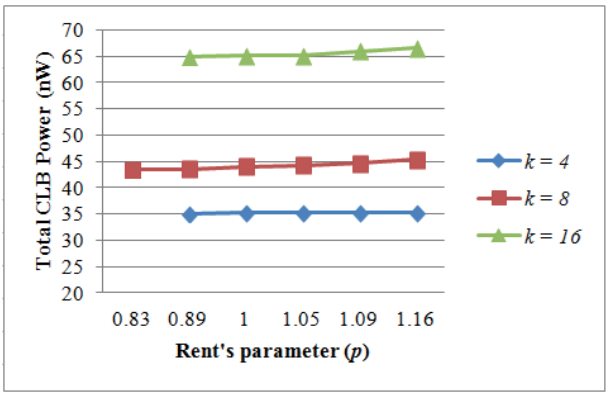

Figure 6: Total CLB power vs. Rent's parameter and cluster size.

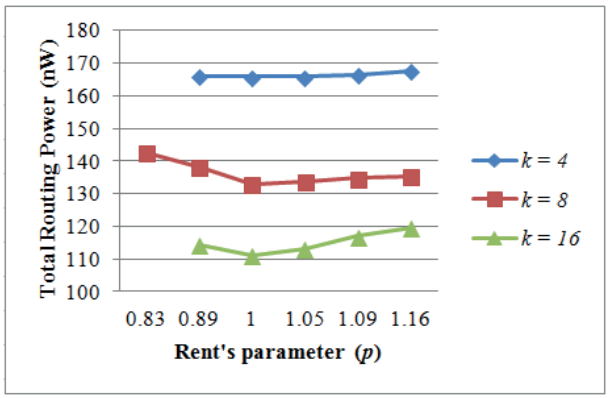

Figure 7: Total routing power vs. Rent's parameter and cluster size.

We notice that the total power consumption is reduced when we increase k. As shown in Figure 5, clusters of size 8 and 16 are more power-efficient than 4. This behaviour is due to the fact that the total number of buffer, used to drive global routing wires, is reduced (see Table 2). In fact, increasing k results in more BLEs being added to a CLB and then circuit can be implemented in smaller FPGA size (see Table 2). In addition, the number of connections routed externally to the CLB is reduced and then the resulting total number of buffer decreases. According to [12], buffers used to drive global routing wires are the major factors behind power dissipation.

To understand why total power consumption of clusters size 8 and 16 are quite close, it is instructive to break out the power components of the data in Figure 5. The total power can be broken into two major parts: the total routing power and CLB powers. Figures 6 and 7 plot respectively the total routing and CLB power with p for different k. By increasing k, we increase CLB multiplexers number to connect LUTs and consequently increase the total CLB power. That is why total CLB power of cluster size 16 is higher than cluster size 8 (see Figure 6). However, with larger k, we can absorb larger number of nets and communication becomes local and then the total number of buffer decreases. Consequently, the total routing power of cluster size 16 is lower than cluster size 8 (see Figure 7). These two opposite effects make the total power of cluster arity 8 and 16 quite close. It is also important to note that with high p, the number of buffer of cluster size 16 increases more rapidly than cluster size 8. Therefore, routing power of cluster size 16 grows more rapidly than cluster size 8 and that is why total power of cluster size 16 becomes higher than cluster size 8 with higher p.

According to results shown in [12], the major source of power dissipation comes from routing resources. This result is also confirmed in our experimentations. In fact, routing power denotes the major factor behind power dissipation in all experimentations done. However, we note that the percentage of routing power drops slightly when we increase k. For example, routing power denotes 80% of the total power for cluster size 4, while it drops to 63% with cluster size 16. Figure 8 plots the variation of the total power of buffers used to

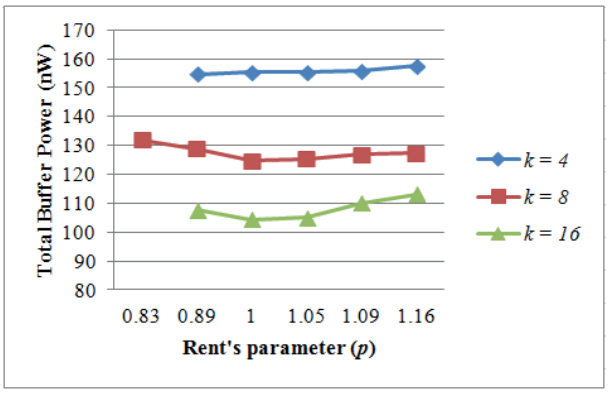

Figure 8: Total buffer power vs. Rent's parameter and cluster size.

drive global routing wires with p for different k. Figure 8 shows that the total buffer power takes up to about 90% of the total routing power. In addition, when we increase k from 8 to 16, the total buffer number is reduced by an average of 20% and then we reduce the total buffer by an average of 17%. All these analyses clearly confirm once again that the number of buffer is the major source of power.

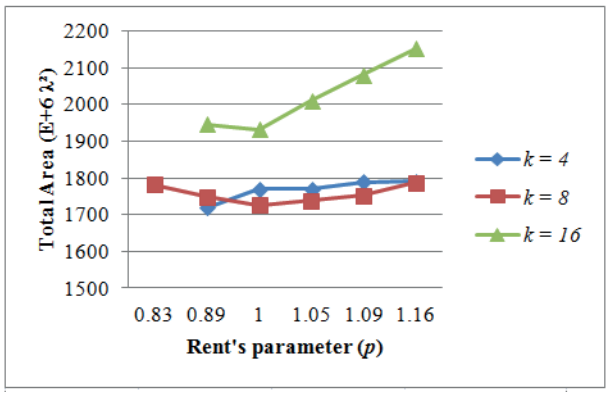

Figure 9: Total area vs. Rent's parameter and cluster size.

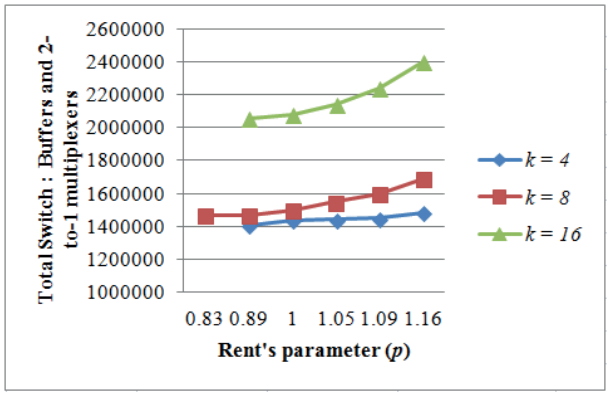

Figure 10: Total switch number vs. Rent's parameter and cluster size.

Figures 9 and 10 illustrate the variation of the total area and number of switch with p for different k. We notice that the total area is increased when we increase k (see Figure 9). Clusters of size 4 and 8 are more area-efficient than 16. In fact, when we increase k, the required multiplexers grow larger and then switches number increases (see Figure 10). Total required switches is a sum of all buffers and 2-to-1 multiplexers used to implement different m-to-1 multiplexers of MSB. Even if the number of buffer decreases with the increase of cluster size, the total number of 2-to-1 multiplexers increases.

To understand why total area of clusters size 4 and 8 are quite close, it is instructive to break out the area components of the data in Figure 9. The total area can be broken into two major parts: the routing area and CLB areas. Figures 11 and 12 plot the total routing and CLB area with p for different cluster size. Since the area depends especially on the number of switch, we extracted also the variation of the total number 2-to-1 multiplexers used in routing and CLB respectively in Figures 13 and 14 with p for different k.

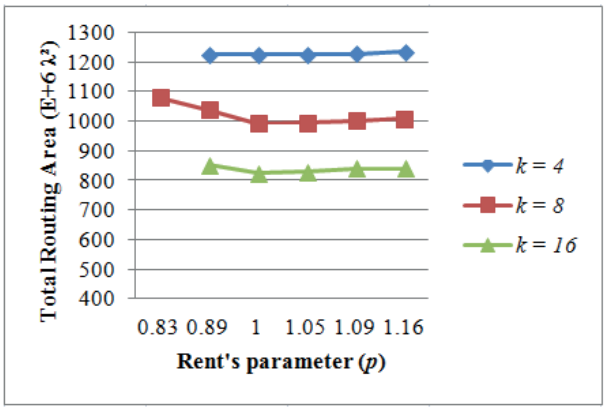

Figure 11: Total routing area vs. Rent's parameter and cluster size.

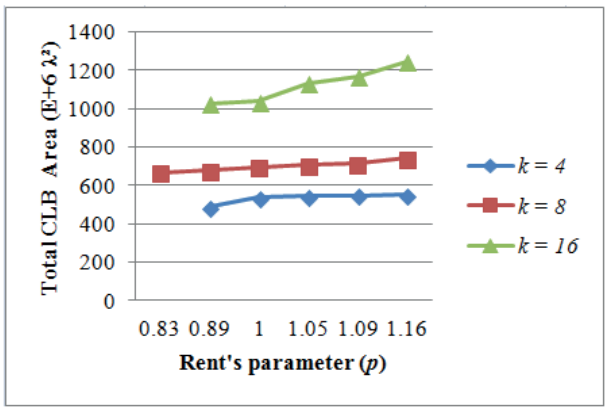

Figure 12: Total CLB area vs. Rent's parameter and cluster size.

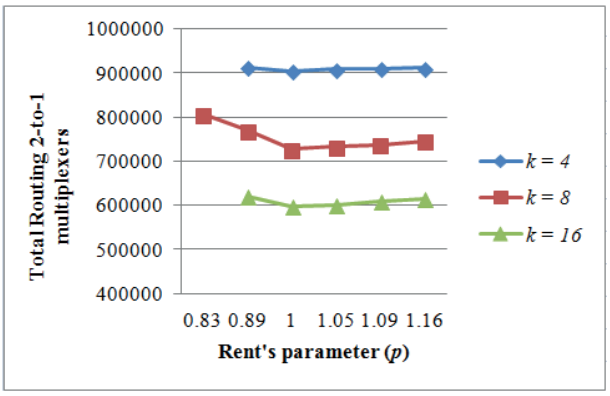

Figure 13: Total routing 2-to-1 multiplexers vs. Rent's parameter and cluster size.

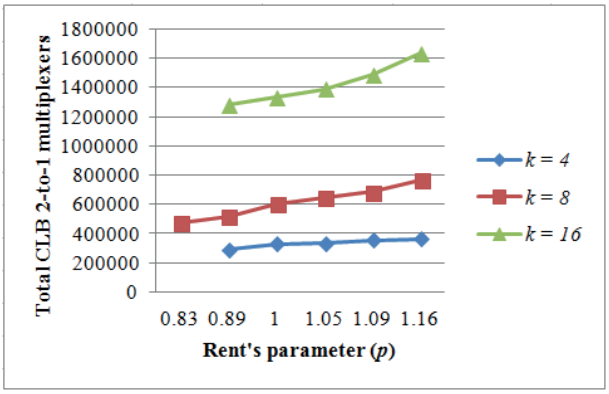

Figure 14: Total CLB 2-to-1 multiplexers vs. Rent's parameter and cluster size.

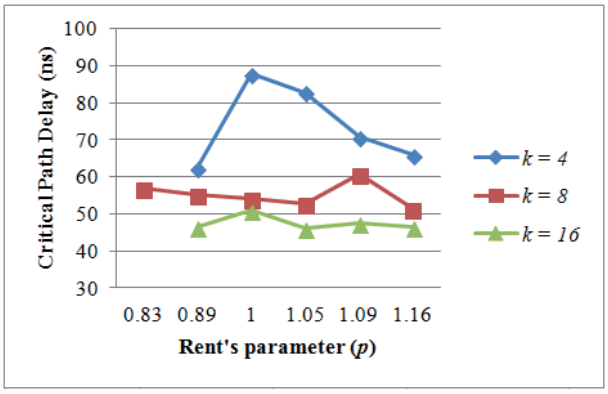

Figure 15: Total critical path delay vs. Rent's parameter and cluster size.

By increasing k, the number of 2-to-1 multiplexers and buffers in SBs decrease (see Figure 13) and then total routing area decreases (see Figure 11). Thus, routing area of cluster size 8 is lower than cluster size 4. However, the required CLB multiplexers grow larger and consequently the bound on area efficiency goes down. As shown in Figure 1, for p=1, in the case of architecture with clusters arity 8, we use 10-to-1 multiplexers within DMSB, 8-to-1 multiplexers within UMSB and 4-to-1 multiplexers to implement LUTs. In total, we use 368 2-to-1 multiplexers to implement the cluster arity 4. With cluster arity 4, we use 5-to-1 multiplexers within DMSB, 4-to-1 multiplexers within UMSB and 4-to-1 multiplexers to implement LUTs. In total, we use 88 2-to-1 multiplexers to implement the cluster arity 8. Therefore, as shown in Figure 14, the total CLB 2-to-1 multiplexers increases when we increase k from 4 to 8 and then total CLB area increases (see Figure 12). These two opposite effects make that clusters 4 and 8 have close resulting number of switch and then total area. Similarly to pervious observations contestations, for all k, the area is reduced until we reach the break point (p=1) from which the increase of p penalizes the resulting area (especially for k=16).

In terms of performance, Figure 15 shows the variation of critical path delay with p for different k. The critical path delay decreases when we increase k. In fact, using larger clusters arity allows reducing external communications and then reduces the number of crossed switch in the critical path delay.

8 Conclusion

In this paper, we presented a new MoC-based FPGA architecture. We showed that the proposed MoC-based architecture has better area and power efficiency than the common MoC VPR-Style. Based on analytical and experimental methods, we showed also that the flexibility of proposed MoC-based architecture is controlled by architecture parameters. The choice of clusters arity must be consistent with the application specifications and constraints. For applications requiring high speed performance and low power dissipation, it is recommended to use clusters with high arity (8-16). If we need to reduce silicon area, using small clusters arities seems to be more efficient. We note from experimentation that cluster size 8 presents the best trade-off between area and power compared to cluster sizes 4 and 16. This is achieved due to the equitable sharing of resource between CLB and routing. As a future work, we plan to add long routing segments which span multiple SBs in every row and column in order to avoid crossing multiple switches to connect clusters, which are not neighbours and to improve the flexibility and routability.

9 References

1. Lin, M., Gamal, A., Lu, Y. and Wong, S. : 'Performance Benefits of Monolithically Stacked 3D FPGA', Proceedings of the ACM/SIGDA 14th ISFPGA NY USA, pp 113-122, 2006.
2. Ahmed, E., Rose, J. : 'The effect of LUT and cluster size on deep-submicron FPGA performance and density', Very Large Scale Integration (VLSI) Systems, 2004, pp 288 - 298.
3. Landman, B., Russo, R. : 'On Pin Versus Block Relationship for Partition of Logic Circuits', IEEE Transactions on Computers, 1971,pp. 1469-1479.
4. Betz, V., Marquardt, A. and J. Rose : 'Architecture and CAD for Deep-Submicron FPGAs', ACL Symposium on Parallel Algorithms and Architectures series, 2000.
5. Lemieux, G., Lewis, D.: 'Design of Interconnection Networks for Programmable Logic, Kluwer Aca-

demic Publishers, Dordrecht, The Netherlands, 2004.
6. Feng. W, Kaptanoglu, S.: 'Designing efficient input interconnect blocks for LUT clusters using counting and entropy', Proc. of FPGA, Monterey, Calif, USA, February 2007, pp. 2332.
7. Marrakchi, Z., Mrabet, H., Farooq, U., Mehrez, H.: 'FPGA Interconnect Topologies Exploration', Int. J. Reconfigure. Comp, 2009.
8. Lamoureux, J., Wilton, S. : 'FPGA Clock Network Architecture: Flexibility vs. Area and Power', FPGA, 2006.
9. Marquart, A., Betz, V., Rose, J. : 'Using cluster-based logic block and timing-driven packing to improve FPGA speed and density', International symposium on FPGA, Monterey,1999, pp. 3746.
10. McMurchie, L. : 'Pathfinder: A Negotiation-Based Performance-Driven Router for FPGAs', Proc. FPGA, 1995.
11. Lamoureux, J. , Wilton, S. : 'Activity Estimation for Field Programmable Gate Arrays', in FPL, 2006.
12. Goeders. J, Wilton, S. : 'VersaPower: Power Estimation for Diverse FPGA Architectures', International Conference on Field-Programmable Technology (FPT), 2012.
http://er.cs.ucla.edu/benchmarks/ibm-place.
http://iwls.org/iwls2005/benchmarks.html.
13. Rose, J., et al. : 'The VTR Project: Architecture and CAD for FPGAs from Verilog to Routing', FPGA, 2012.
14. International Technology Roadmap for Semiconductors, 'Interconnect,' 2007.
15. Weste, N., Harris, D. : 'CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition). Addison Wesley, 2010.

Arrived: 09. 06. 2015
Accepted: 09. 02. 2016