A Cluster-based Hierarchical Partitioning Approach for Multiple FPGAs

Chunhua Xiao, Zhangqin Huang, Da Li
Beijing University of Technology, Embedded Software and System Institution
Beijing 100022 China
Email: xiaochh@emails.bjut.edu.cn

***Abstract*—Most high performance computing systems are large-scale computing systems, and consist tens of thousands computing nodes with superior capabilities. FPGAs are able to accelerate large scope and complicated computing with flexible configurations. More and more companies and research institutions integrate multi-FPGAs into high performance computing systems to get a better trade-off between high-performance and low power. How to design an effective topology for these integrated multi-FPGAs according different applications has become a key problem in this area. Acluster based architectureand corresponding partitioning approach are proposed in this paper. The proposed hierarchical topology taking full advantages of both traditional metallic lines and emerging interconnections to implement one-hop local communication within the cluster and one-hop global high-speed communication between clusters. The case study proved that the proposed architecture and partitioning approach can implement the fast mapping from the design to real computing system with multi-FPGAs, and accelerate the realization of high performance reconfigurable computing systems.**

***Index Terms*—High performance computing, multi-FPGA architecture, FPGA partitioning, emerging interconnections**

I. Introduction

Most high performance computing systems are large-scale computing systems, and consist tens of thousands computing nodes with superior capabilities [1]. To satisfy the challenges in performance and energy consumption, the mechanisms such as NUCA (Non-uniform cache architecture) and page-recoloring[2][3]are adopted in these large-scale systems[4][5], and the communication pattern tends to non-uniform and heterogeneous. Cluster-based architectures have become the mainstream in the design of high performance computing systems. As shown in Figure 1, up to 80% computing systems adopt the cluster-based architecture, which stands in a monopolistic place in the ranking list [6].

As the advanced requirements for High Performance Computing (HPC), not only the need for reliable and high performance computingincreased, but also the demand forlow cost, low energy consumption and high speedare on the increase. The traditional high performance computing systems usually cost several millions or even up to tens of billions of dollars, and hard to upgrade. With the improvement of CMOS technology, FPGA (Field Programmable Gate Array) is able to accelerate large-scale computing, and dramatically reduce the computing power and cost with flexible configurations. More and more companies and research institutions integrate multi-FPGAs into high performance computing systems to get a better trade-off between high-performance and low power[14][15].The interconnectionnetworksareusually customized for special applications, and varies considerably for different high performance computing systems. How to design an effective topology for these integrated multi-FPGAs according different applications has become a key problem.

The interconnection network is the channels for data flows and communications between the processing nodes in the computing systems, which plays a key role for the performance improvement of HPC. Conventional VLSI mainly adopts metallic lines with low impedance and high conductivity to implement the interconnections of on-chip components. Traditional metallic interconnects face serious transmission delay, bandwidth density issues and power consumption problems asthe expanding scale of circuits. On the other hand, emerging interconnections such as optical interconnects and RF interconnects have become the alternative technologies to replace the traditional RCs [7]-[9]. These emerging interconnections are able to achieve one-hop effective communications between long distance nodes with low power and high bandwidth, and have been widely used to implement the communicationsbetween computing cabinets.The researches to implement the interconnections within computing cabinets, between boards and on chips have been also explored recently[31][32][33][34], and obvious achievements have been obtained. However, they cannot completely replace the traditional RCs in a short time due to the design challenges such as transmission interference and high sensitivity to temperature[10]. The hybrid architectures that composite traditional RCs and emerging interconnections have become an efficient

Manuscript received December 25, 2013; revised March 25, 2014; accepted April9, 2014.

doi:10.4304/jcp.9.9.2173-2180

implementation approach during the technological transition period.

To target the challenges in performance, cost and energy consumptionin large scale communications of HPCs, and exploit the local[11][12] and heterogeneous[13] properties of future large scale communications, we propose a cluster based hierarchical topology and corresponding partitioning approach. The proposed hierarchical topology taking full advantages of both traditional metallic lines and emerging interconnections to implement one-hop local direct communication within the cluster and one-hop global high-speed communication between clusters.The customized interconnection topology can be designed according different applications through the proposed approach and be implemented through the fast mapping from the design to the real high performance computing system with multi-FPGAs.

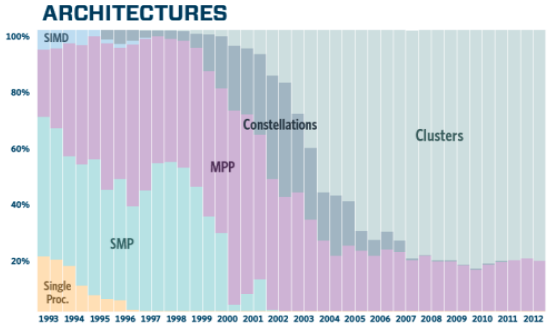


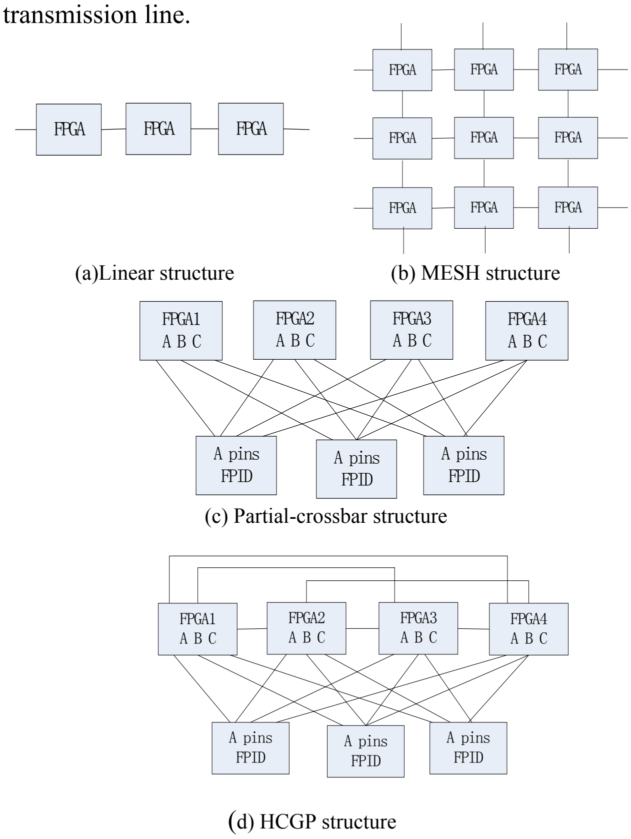
Figure 1 The architecturesdistribution of HPCs in world TOP500[6]

II.The Proposed Topology for Multi-FPGAs Oriented to HPCs

The topology of multi-FPGAs is defined as the connections between the multiple FPGAs. Currently traditional typical topologies include linearstructures[17][18],MESHbased structures[19][20][21][22],partial-crossbar structures[23][24][25] and Hybrid Complete Graph Partial-crossbar (HCGP) structures[26][27][27][29][30] (As shown in figure 2). These interconnection architectures have respective advantages to satisfy different application requirements, but might face serious scalability and routing problems for future large scale computing systems integrated tens to thousands of FPGAs.

As shown in figure 3, our proposed topology is a cluster based hierarchical architecture which consists two tier communication layers: local network (intra-cluster) and global network (inter-cluster). The interconnections within the cluster is based on traditional metallic wired links, while the inter-cluster interconnects are implemented by emerging high speed interconnections such as RF-I and optical interconnection. The local network is completely interconnected and all the nodes in the same cluster can communication each other directly. There is a relay node in each cluster to achieve the interconnection between local communication and global communication. The relay node can be designed with high performance FPGAs to integrate partial computing functions while in charge of the cross-talk communication, or can be implemented with small scale FPGAs to be only responsible for the communications between two layers. If the nodes want to communicate with other nodes in different clusters, they need to transfer the message to the local relay node firstly, then the local relay node transmit the coded message through the high speed RF channel to the relay node of the destination cluster, and finally forwards the message to the destination node by the relay node in destination cluster, so 3 hops needed for inter-cluster communications.

In traditional topologies, the communication between FPGAs that non-directly connected requires routing through intermediate FPGAs. Although these routing algorithms are designed to be simple and efficiency [35][36], communication latency will increase greatly as the computing system scales due to the limited bandwidth and massive routing hops. In our proposed architecture, the FPGAs in the same cluster communicateeach other directly through traditional interconnection, and the FPGAs in different clusters communicate through express RF-I (Radio Frequency Interconnect).The design can be scaled flexibly through the cluster augment with our proposed architecture, and only 2 intermediate hops needed for long distance communications. Besides, the RF-I between clusters divides bandwidth into frequency domains, each becoming a narrow-band signal, which saves power and improves bandwidth efficiency by sending many simultaneous streams of data over a single transmission line.



(a)Linear structure (b) MESH structure

(c) Partial-crossbar structure

(d) HCGP structure

Figure 2 Typical topologies for multi-FPGAs

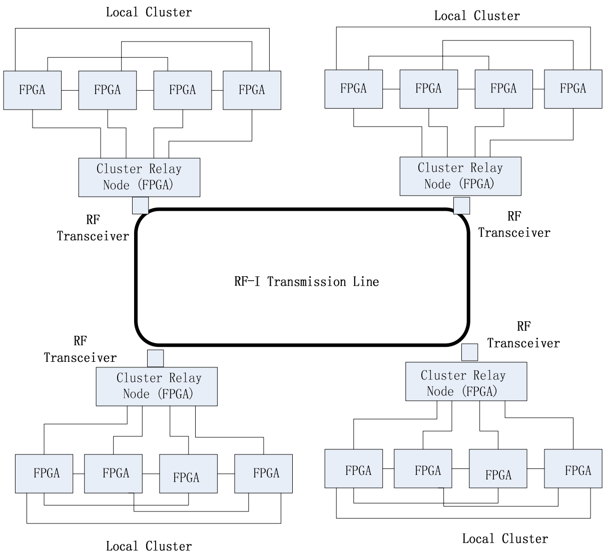


Figure 3 Cluster based hierarchical architecture for multiple FPGAs

III.CLUSTER BASED PARTITIONING APPROACH FOR MULTI-FPGAS

To minimize the average communication latency of the cluster based multi-FPGA system, we propose a partitioning approach based on the former researches[37][41][42][43][44][45] to concentrate the communications within the clusters to minimize the global communication required between clusters. The proposed partition flow is shown in Figure 4, which includes the steps of architecture initialization, partitioning and partition optimization.

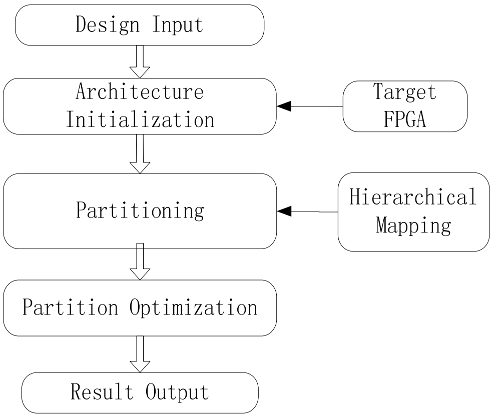


Figure 4 The hierarchicalpartitioning flow for cluster based Multi-FPGAs

A. Architecture Initialization

Before the logic partitioning and mapping, 3 architecture parameters need to be initialized, which are:

1) *Num_FPGAs*:the number of FPGAs needed to map the input design;

2)*Num_Clusters*:the number of clusters of this design;

3) *Size_Clusters:*the size of each cluster (the number of FPGAs in the cluster).

We can get the number of the mapping FPGAs through the comparison of the characteristic parameters between the input design and the target FPGA. We assume the input design is described with hardware described languages (VHDL/Verilog) in this paper. The RTL level description can be obtained through EDA tools to get the characteristic parameters of the design, such as the Configurable Logic Blocks (CLBs), the input/output pins (I/Os) and the Flip-Flops. For a given input design, the required number of target FPGAs is different when the selected mapping FPGA is not in the same size. To simplify the complexity, we assume the target FPGAs are the same size in this work.

To ensure the effective placement and routing, we constrain the capacity of the target FPGA that can be mapped to 75%. If the capacity of the target FPGA is R, the resource that can be mapped for the input design is constraint to 3R/4. We adopt the maximum value (upward rounding) of the comparisons of the characteristic parameters between input design and the constraint target FPGA as the number of target FPGAs needed for the input system. Please refer to the pseudo-code (shown in the algorithm 1) for the calculation details.

The size of clusters is a trade-off between the communication performance and interconnection cost of multi-FPGA systems. If we partition the entire system into small clusters, which means fewerFPGAs in a cluster, the communication performance can be improved greatly due to more communications going through global high-speed channels. But the hardware cost will be increaseddue to more relay nodes needed. On the contrary, if we place more FPGAs in a cluster to reduce the number of clusters of the system, the relay node might become a performance bottleneck if there are too many communications try to go through the global channel.

The interconnection interfaces supported by the target FPGA is another constraint to decided the size of clusters. The high performance FPGAs are usually integrated with high speed communication modules. Taking the Xilinx FPGAs as an example, the Rocket IO modules integrated in Xilinx FPGAs can achieve up to 3.125 Gb/s for high speed interconnections, but the number of Rocket IOs that supported by different series is different from 8 to 24 pairs. So the size of the clusters cannot exceed the number of high-speed communication interfaces supported by the target FPGA. We set the maximum size of the cluster to be m-1 in this work to implement the complete interconnections within the cluster, where, m is the number of high-speed communication interfaces supported by the target FPGA. The size of clusters can be set to be any integer less than m-1 theoretically, but the trade-off between performance and cost needs to be weighted.

Once the number of target FPGAs to be mapped is fixed and the size of clusters is constraint, we can get the number of clusters needed for the input design. The pseudo-code of the architecture initialization is shown in the algorithm 1.

B. Partitioning

To minimize the global communication overhead, we adopt the hierarchical partition approach based on the functional modules of the input design, which means try to partition the modules with the same logic function to the same cluster or the same target FPGA. The design is usually described with multiple interconnected modules if the input design is based on VHDL or Verilog, and each module is consisted with multiple processes and functional sub-modules. We exploit the hierarchy to set the mapping priority. If the module is a top module, it enjoys a higher mapping priority to choose the placement position firstly. For the modules in the same hierarchy, the mapping priority is allocated through the size of the modules, higher priority for bigger module. If the module to be mapped cannot find a reasonable position due to the size problem, the module is decomposed to smaller ones according to functional structure.

The input design is mapped to the clusters firstly, and then mapped to the target FPGAs in the clusters. We implement the two tier partition based on the same hierarchical mapping (the pseudo-code is shown in Algorithm1), the steps includes:

1) Allocate the mapping priority according the hierarchy and the size of the modules, higher priority means to choose the placement position firstly.

2）The selected module will be mapped to the position which makes the target set to have minimum remaining resources un-mapped.

3) If the module to be mapped cannot find a reasonable position due to the size problem, the module will be decomposed to smaller ones according the functional structure;

4) If there are new modules generated through the decomposition, re-allocate the priority and mapping all the un-mapped modules with the new generated modules.

```
Procedure Architecture_Initialization($V_{design}$, $FPGA_{target}$) begin
Imprt a HDLs description;
Get the number of total I/Os, FFs, LUTs of the input design $V_{design}$ :Total_LUTs, Total_FFs , Total_IOs
Get the number of total I/Os, FFs, LUTs of the target FPGA $FPGA_{target}$ :FPGA_LUTs, FPGA _FFs , FPGA _IOs
//Compute the number of FPGAs needed for the input design
Num_FPGAs = Ceil ((Total_LUTs)/(FPGA_LUTs), (Total_FFs)/(FPGA_FFs) , (Total_IOs)/(FPGA_IOs))

//Fix the Cluster Parameters:
Cluster_pattern = Num_FPGAs % m;
K = Num_FPGAs /4
if (Cluster_pattern==0)
        Num_Clusters = k
        Size_Clusters = {m,m,m,…m}
else if (Cluster_pattern==1)
        Num_Clusters = k+1
        Size_Clusters = {m,m,m,…m-1}
else if (Cluster_pattern==2)
        Num_Clusters = k+1
        Size_Clusters = {m,m,m,…m-1,m-1}
else if (Cluster_pattern==3)
        Num_Clusters = k+1
        Size_Clusters = {m,m,m,…m-1}
Return Num_Clusters, Size_Clusters
End Procedure
```

Algorithm1. The pseudo-code of the architecture initialization

C.Partition Optimization

After the logic mapping, the optimization algorithms can be adopted to further reduce the interconnection cost. In the previous step, we already tried to map the logical function-modules that are closely interconnected into the same target FPGA or FPGA cluster to get the smaller external interconnection overhead. This step is aimed primarily at the designs with stringent resource requirement. The design that is hierarchical mapped in the previous step can be used as a initialized input to traditional partitioning algorithms[38][39][40] such as KL algorithm and FM algorithm to get a further optimized partition result.

```
Algorithm Hierarchical_Set_Covering(M,C)
// M is the sets of the modules to be mapped；
// C is the mapping target sets；
            Re-organize the order of the sets M by the largest Module to
        the minimal one：V   M;
$V_g$ = ∅; $V_d$ = ∅; $V_{new}$ = ∅;
//$V_{new}$is the set of new generated modules；
// $V_g$  is the set of new generated modules of the selected module；
// $V_d$is the set of the modules that has been de-composed;

While (V≠∅)
$Num_{\_nodes}$ = size of V;
i=0;
while (i<$Num_{\_nodes}$) begin
    {
$V_{current}$= $V_i$
        for all($C_k \in C$)
if（LUTs({$V_{current} \leqslant C_k$}) && FFs({$V_{current} \leqslant C_k$})&& IOs({$V_{current} \leqslant C_k$})）
                score($C_k$,$V_{current}$) =  α Connectivity($C_k$,$V_{current}$) +  β;
            else
score($C_k$,$V_{current}$) = 0;
if all (score($C_k$,$V_{current}$) = 0) then begin
{ {$V_{new}$} = Functional_Module_Decomposition($V_{current,}$ Maximum($C_k$))
                $V_g = V_g \cup \{V_{new}\}$; }
else
{ select the pair of { $C_k$,$V_{current}$ } with the highest score;
                $V_d = V_g \cup V_{current}$;
                $C_k = C_k - V_{current}$; }
}
i++
}
end while;
  $V = V \cup V_g - V_d$;
Re-organize the order of the sets V by the largest Module to the minimal one。
end while;
```

Algorithm 2. The Pseudo-code of the hierarchical mapping

IV.A CASE STUDY OF THE PROPOSED PARTITIONING APPROACH

In this section we use an actual case study and experiment to help understand our proposed approach more clearly.

The case study is a design of the speech recognition implemented by our lab, and described with VHDL. The given input is comprised with 6 top-modules, the characteristic parameters for each module after the synthesis with Synopsys is shown in Table I. The Xilinx 4013E-1 FPGA is adopted as target FPGAs, and we assume to map the input design to the same type of FPGAs for this case study.

According to the proposed partition flow, the first step is to initialize the architecture parameters through the characteristic parameters of the input design and the target FPGA constraint. The key parameters of the target FPGA is: 1152 LUTs, 1152 Flip Flops (FFs), 192 I/Os. Considering the feasibility for placement and routing, we reserve 25% available resource. So the mappable capacity of the target FPGA is constraint as: {864 LUTs, 864 FFs, 144 I/O s}. The number of the target FPGAs to be mapped is defined as:

$$\mathrm{Num\_FPGAs}=\mathrm{Max}\left\lceil\frac{\mathrm{Total\_LUTs}}{\mathrm{FPGA\_LUTs}},\frac{\mathrm{Total\_FFs}}{\mathrm{FPGA\_FFs}},\frac{\mathrm{Total\_IOs}}{\mathrm{FPGA\_IOs}}\right\rceil=10$$

TABLE I.
THE CHARACTERISTIC PARAMETERS OF THE TOP MODULES OF THE CASE STUDY

Modules	LUTs	FFs	I/Os	Total Size
Module A	1230	987	100	2317
Module B	54	70	10	134
Module C	2100	1879	220	4199
Module D	3470	2897	330	6697
Module E	954	800	129	1883
Module F	692	751	111	2154
All Modules	8490	7384	1339	

Next we need to fix the size of the clusters. As mentioned before, the cluster size is usually a compromise between system performance and cost. If a cluster contains many FPGAs, the communication pressure is heavy for the relay FPGAs, and the transmission latency will increase, but the cost reduced due to less hardware overhead. On the contrary, the more the number of clusters, the higher performance we can get through the high speed transmission through global channel, but with more cost expense. We constraint the number of clusters as 4 or 3 to get a better trade off for this case study and choose the cluster policy with minimum number of clusters. For example, if Num_FPGAs=12, the possible cluster initialization policies might be {4,4,4} or {3,3,3,3}, the policy of {4,4,4} will be adopted since the number of cluster is only 3, less than the policy of {3,3,3,3}.

So according to the initializationalgorithm described before, the number of clusters is 3 and the size for each cluster is {4, 3, 3} for this case study. The detailed key parameters for each cluster are:

Cluster1, 4 target FPGAs, resources:{ 3456 LUTs, 3456 Flip Flops (FFs), 576 I/Os }；

Cluster2, 3 target FPGAs, resources:{ 2592 LUTs, 2592 Flip Flops (FFs), 432 I/Os }；

Cluster3, 3 target FPGAs, resources:{ 2592 LUTs, 2592 Flip Flops (FFs), 432 I/Os }；

Module D is the largest circuit module in this case study and should be mapped to the clusters firstly. But the size of module D exceeds the amount resource of the target clusters as only 3456 LUTs can be mapped to the largest cluster, cluster 1. So we de-compose the module D to module D1{1470 LUTs, 1024 Flip Flops,130 I/Os} and D2{2000 LUTs, 1873Flip Flops,200 I/Os} according the logical functions.

The priority of the top modules to be mapped is "Module C→ModuleD2→Module D1→Module A→ModuleE → ModuleF → ModuleB" after the decomposition of module D according the proposed approach. The module C enjoys the highest priority and to be the first module to be mapped, so there are 3 choices: Cluster 1, Cluster2 and Cluster 3 for module C. If the module C is mapped to cluster 1, the remaining resource would be {1356 LUTs, 1577 Flip Flops, 356 I/Os} after the mapping. If the module C is mapped to cluster 2 or cluster 3, the remaining resource of cluster 2 and cluster 3 would be {492 LUTs, 713 Flip Flops, 212 I/Os}. ）According to the principle that "the selected module will be mapped to the position which makes the target set to have minimum remaining resources un-mapped", we choose to map the module C to cluster 2, and so the remaining resource after mapping for each clusters are:

Cluster1: { 3456 LUTs, 3456 Flip Flops (FFs), 576 I/Os }；

Cluster2: {492 LUTs, 713Flip Flops (FFs), 212 I/Os }；

Cluster3: { 2592 LUTs, 2592 Flip Flops (FFs), 432 I/Os }；

On the analogy of this, the module D1 and module A are mapped to cluster1. When it is the turn for module E to choose the placement position, the mappable resources of each cluster are：

Cluster1: {756 LUTs, 1445 FFs, 346 I/Os }；

Cluster2: {492 LUTs, 713FFs, 212 I/Os }；

Cluster3: { 592 LUTs,719 FFs, 232 I/Os }；

So the module E need to be decomposed as module E1{554 LUTs, 420Flip Flops,70 I/Os} and E2{400 LUTs, 380Flip Flops,59 I/Os}. Now the modules un-mapped are module E2, module E1, module F and module B, and the priority changed to "Module F→ModuleE1→Module E2→Module B".

Byanalog, we can get the mapping result of all the modules as:

Cluster1: { Module D1, Module A, Module F, Module B }；

Cluster2: { Module C, Module E1}；

Cluster3: { Module D2, Module E2}；

The way to map the design from the cluster to target FPGAs is similar as the mapping flow mentioned above. To simplify, we only take the cluster 1 as the example. After the architecture initialization, we know the size of cluster 1 is 4, and the constraints of the target FPGAs of cluster 1are:

FPGA 1: { 864 LUTs, 864 FFs, 144 I/O s }；

FPGA 2: { 864 LUTs, 864 FFs, 144 I/O s }；

FPGA 3: { 864 LUTs, 864 FFs, 144 I/O s }；

FPGA 4: { 864 LUTs, 864 FFs, 144 I/O s }；

According the proposed algorithm, the priority of the modules to map to cluster 1 is: "Module D1→Module A→Module F→Module B", and the mapping result is:

FPGA 1: { Module B, Module D1_1 }；

FPGA 2: { Module A1, Module A2_1}；

FPGA 3: { Module F, Module A2_3}；

FPGA 4：{Module D1_2, Module A2_2}；

To further reduce the interconnection cost, the MP2 algorithm [37]is adopted in this case study to optimize the partition.

To verify the proposed approach, the partition flow is described with High-Level programming language and integrated into the EDA tools (Synopsys Certify-G) through input interfaces for this case study. Experimental result shows that the input design can be partitioned reasonably with a fast speed, and the processes of synthesis, placement and routing can be implemented correctly. We get an average of 23.8% remaining available resource after the implementation of the case design, which means the possibility to relax the constraint of the reserved mapping resource to get a better mapping policy.

V. Conclusion

To target the problems of performance, cost and energy consumption in large scale communications of high performance computing, a cluster based hierarchical topology and corresponding partitioning approachare proposed in this paper. The proposed hierarchical topology taking full advantages of both traditional metallic lines and emerging interconnections to implement one-hop local direct communication within the cluster and one-hop globally high-speed communication between clusters. The customized interconnection topology can be designed according different applications through the proposed partitioning approach to get a better trade-off between performance and cost. The case study proved that the partitioning approach can implement the fast mapping from the design to real high performance computing system with multi-FPGAs, and accelerate the realization of high performance reconfigurable computing systems.

Although the emerging interconnections are advanced in long distance and high bandwidth communications, the introduced extra overhead and cost cannot be ignored for multi-FPGA systems. It is necessary to analyze the performance and energy consumption quantitatively to achieve anoptimum configuration for these emerging effective interconnections. In addition, how to allocate the limited high speed communicationbandwidth and how to solve the traffic congestions are also the important problems to be explored in future works.

Acknowledgment

This work was supported by the Beijing Municipal Natural Science Foundation (No.4122010, 2012.1 - 2014.12) and Beijing innovation platform of Science and technology.

References

[1] ZhouXingming,“High performance computing [J],”*Journal of Chinese Journal of Nature*, vol. 33,no. 5, pp. 249-254, 2011.

[2] N.Hardavellas,M.Ferdman, B.Falsafi,and A.Ailamaki, Reactive NUCA: Near-optimal block placement and replication in distributed caches [C] // in *Proc. of the 36th Annual International Symposium on Computer Architecture*,2009. New York, ACM, 2008, pp. 184–195.

[3] CHO, S. AND JIN, L. Managing distributed, shared L2 caches through OS-level page allocation [C] // Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 39) 2006. Orlando, FL : IEEE/ACM, 2006, pp. 455–468.

[4] QURESHI, M. AND PATT, Y. Utility-based cache partitioning: a low-overhead, high-performance, runtime mechanism to partition shared caches [C] // Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 39) 2006. Orlando, FL : IEEE/ACM, 2006, pp. 423–432.

[5] LEE, H., CHO, S., AND BRUCE R.C. CloudCache: Expanding and shrinking private caches [C] // Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA) 2011. San Antonio, TX, IEEE Computer Society Press, 2011, pp. 219–230.

[6] TOP500 Supercomputing Sites [EB/OL]. [2012-11]. [online] Available: http://www.Top500.org.

[7] M. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y. Qian,“RF/wireless interconnect for inter- and intra-chip communications [J],” *in Proc. of the IEEE*, vol. 89, no. 4, pp. 456–466, 2001.

[8] J. Lin *et al.*“Communication using antennas fabricated in silicon integrated circuits [J],”*Journal of IEEE Solid-State Circuits*, vol. 42, no. 8, pp. 678–1687, 2007.

[9] D. Mizoguchi, Y. Yusof, N. Miura, T. Sakura, and T. Kuroda, A 1.2 gb/s/pin wireless superconnect based on inductive inter-chip signaling (IIS) [C] // Proceedings of the IEEE International Solid-State Circuits 2004. IEEE, 2004: vol.1, 142–517.

[10] ITRS Sites [EB/OL]. [2012].[Online] Available:http://www.itrs.net/Links/2012ITRS/Home2012.htm

[11] Tutsch D, Malek M. “Comparison of network-on-chip topologies for multicore systems considering multicast and local traffic [C]” in *Proc. of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SimuTools 2009).* Brussels: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009: Article No. 2

[12] Tutsch D, Lüdtke D. “Chip multiprocessor traffic models providing consistent multicast and spatial distributions [J],”*SIMULATION*, vol. 84, no. 2-3, pp. 61–74, 2008.

[13] Wang Wei. Studies on Key Technologies of Network-on-Chip Interconnection for Very Large Scale Chip-multiprocessors [D]. Beijing: Tsinghua University. Department of Computer Science and Technology, 2010.

[14] Baxter R, *et al.*“High performance reconfigurable computing -- the view from Edinburgh [C]” in*Proc. of 2nd NASA/ ESA on Adaptive Hardware and Systems.* Edinburgh: ACM, 2007, pp. 279-373.

[15] El Ghazawi T, El Araby E, Huang M, *et al.* “The promise of high performance reconfigurable computing [J],”*Computer*, vol. 41, no. 2,pp. 69-75, 2008.

[16] Ma Lei. The Research on Optical Interconnection Technologies Oriented HPC [D]. Changsha: National University of Defense Technology. Department of Computer Science and Technology, 2007.

[17] D. E. Van Den Bout, *et al.* “Anyboard: an FPGA-Based reconfigurable system [J],”*IEEE Design and Test of Computers*, vol. 9,no. 3, pp. 21-30, 1992.

[18] M. Gokhale*et al*. "Building and using a highly parallel programmable logic array [J]."*IEEE Computer*,vol. 24, no. 1, pp. 81-89,1991.

[19] S. Walters. "Computer-aided prototyping for ASIC-based systems [J]."*IEEE Design and Test of Computers*, vol. 8, no. 2, pp. 4-10, 1992.

[20] J. E. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, and P. Boucard. "Programmable active memories: reconfigurable systems come of age [J]."*IEEE Transactions on VLSI,*vol.4, no. 91, pp. 56-69, 1996.

[21] S. Hauck, G. Boriello, C. Ebeling. "Mesh routing topologies for multi-FPGA systems [C]" in P*roc. of IEEE International Conference on VLSI in Computers and Processors,* Cambridge, MA: IEEE, 1994, pp. 170-177.

[22] S. Hauck, Multi-FPGA Systems [D], Washington: University of Washington, Department of Computer Science and Engineering, 1995.

[23] M. Butts, J. Batcheller, and J. Varghese. "An efficient logic emulation system [J]."*IEEE Transactions on Very Large Scale Integration Systems*, vol. 1, no. 2, pp. 138-141, 1992.

[24] M. Slimane-Kadi, D. Brasen, and G. Saucier. A Fast-FPGA Prototyping System That Uses Inexpensive High Performance FPIC [C] // ACM/SIGDA Workshop on Field-Programmable Gate Arrays, 1994.

[25] D. M. Lewis, D. R. Galloway, M. Van Ierssel, J. Rose, and P. Chow. "The transmogrifier-2: A 1 million gate rapid prototyping system [J]."*IEEE Transactions on Very Large Scale Integration Systems,*vol. 6, no. 2,pp. 188-198, 1998.

[26] Mohammed A. S. Khalid. Routing Architecture and Layout Synthesis for Multi-FPGA Systems [D]. Toronto: University of Toronto. Department of Electrical and Computer Engineering, 1999.

[27] Jain,Sushil Chandra; Kumar,Anshul; Kumar,Shashi. "Hybrid multi-FPGA board evaluation by permitting limited multi-hop routing [J],"*Design Automation for Embedded Systems*. vol. 8, no. 4, pp. 309-326, 2003.

[28] Khalid, Mohammed A. S. Scalability evaluation of a hybrid routing architecture for multi-FPGA systems [C] // Proceedings of the International Conference on Microelectronics, ICM 2006. Dhahran,pp. 162-165.

[29] Khalid, Mohammed A.S. Hybrid complete-graph partial-crossbar routing architecture for multi-FPGA systems [C] //ACM/SIGDA International Symposium on Field Programmable Gate Arrays – FPGA 2008. Toronto: University of Toronto, 1998, pp. 45-54.

[30] Mohammed A. S. Khalid and Jonathan Rose. "A Novel and efficient routing architecture for multi-FPGA Systems [J],"*IEEE Transactions on VLSI Systems*, vol. 8, no. 1, pp. 30 – 39, 2000.

[31] A. Joshi, C. Batten, Y.-J. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. "Silicon-photonic CLOS networks for global on-chip communication [C]," in *Proc. of IEEE International symposium of Network-on-Chip*. San Diego, CA: IEEE, 2009, pp. 124–133.

[32] H. Gu, J. Xu, and W. Zhang. "A low-power fat tree-based optical network-on-chip for multiprocessor system-on-chip [C]" in*Proc. of Design, Automation & Test in Europe Conference& Exhibition*. Nice, France, 2009, pp. 3–8.

[33] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. Popovic, H. Li, H. Smith, J. Hoyt, F. Kartner, R. Ram, V. Stojanovic, and K. Asanovic. "Building manycore processor-to-dram networks with monolithic silicon photonics [C],"in *Proc. of 16th IEEE Symposium on High Performance Interconnects* 2008. Stanford, CA: IEEE, 2008, pp. 21–30.

[34] Xiao Chunhua, M-C.Frank Chang, Jason Cong, Michael Gill, Huang Zhangqin, Liu Chunyue, Glenn Reinman, Wu Hao. "Stream arbitration: towards efficient bandwidth utilization for emerging on-chip interconnects," *ACM Transactions on Architecture and Code Optimization, Article*,no. 60, vol. 9, no. 4, January 2013.

[35] Jain, S.C., Kumar, A., Kumar, S."Multi-hop routing of multi-terminal nets for evaluation of hybrid multi-FPGA boards [C]," in*Proc. of IEEE International Conference on Field-Programmable Technology*. IEEE, 2002, pp. 298–301.

[36] S.C.Jain,S.Kumar,A.Kumar. "Evaluation of various routing architectures for multi-FPGA boards [C]."*Thirteenth International Conference on VLSI Design 2000*. Calcutta, 2000, pp. 262 – 267.

[37] Nam-Sung Woo, Jaeseok Kim, "An efficient method of partitioning circuits for multiple-FPGA implementation [C]," in *Proc. of the 30th Conference on Design Automation 1993*. Dallas, Texas: ACM/IEEE, 1993, pp. 202 – 207.

[38] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs [J]."*Bell system technical journal*, vol. 49, pp. 291-307, 1970.

[39] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions [C]," in *Proc. of Design Automarion Conference 1982.* Las Vegas, NY: IEEE, 1982, pp. 175-181.

[40] RanieriBaraglia, RaffaelePerego, J. Ignacio Hidalgo, Juan Lanchares and Francisco Tirado, "A parallel compact genetic algorithm for multi-FPGA partitioning [C]," in*Proc. of Ninth Euromicro Workshop on Parallel and Distributed Processing 2001*. Los Alamitos, California: IEEE, 2001, pp. 113-120.

[41] Wen-Jong Fang, Wu, A.C.H.,"A hierarchical functional structuring and partitioning approach for multiple-FPGAimplementations[J]."*IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 16, no. 10,pp. 1188 – 1195, 1997.

[42] Wen-Jong Fang, Wu, A.C.H.,"Integrating HDL synthesis and partitioning for multi-FPGA designs[J]."*IEEE design & test of computers,*vol. 15, no. 2, pp. 65-72,1998.

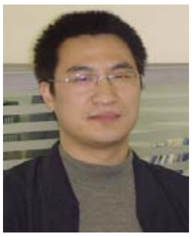
[43] K.Roy-Neogi,C.Sechen,"Multiple FPGA partitioning with performance optimization[C],"in*Proceedings of the Third International ACM Symposium on Field-Programmable Gate Arrays, 1995*, pp. 146-152 .

[44] Wen-Jong Fang, Wu, A.C.H.,"Multi-way fpga partitioning by fully exploiting design hierarchy [J]."*ACM Transactions on Design Automation of Electronic Systems(TODAES),*vol. 5, no. 1, pp. 34-50, 2000.

[45] H.Selvaraj,P.Sapiecha, N.Dhavilikar, "Partitioning of large HDL ASIC designs into multiple FPFA [C]"// in *Proc. of Fourth International Conference on Computational Intelligence and Multimedia Applications,*Yokusika City, 2001, pp. 411-415.

Chunhua Xiao received her B.S. in Electronic Information Engineering from Shijiazhuang Tiedao University, Hebei Province, China, in 2007, and her M.S. in Computer Science from Beijing University of Technology, Beijing, China, in 2010. She is currently a PhD student in Department of Computer Science and Technology, Beijing University of Technology. Her research interests include embedded system co-design, Multi-processor system-on-chip, and Network-on-Chip.

Zhangqin Huang received his B.S., M.S., and PhD in Computer Science from Xi'an Jiaotong University, China, in 1986, 1989 and 2000, respectively. He is currently the Deputy Director of the Embedded Software and Systems Institute (ESSI), Beijing University of Technology (BJUT), China. His current research interests include co-design for embedded software and hardware, human–computer interaction based on internet, Multi-processor system-on-chip, mass data storage, and network information security.



Da Li received his B.S., M.S., and PhD in Computer Science from Xi'an Jiaotong University, China, in 2002, 2006 and 2012, respectively. He is currently a instructor of Embedded Software and Systems Institute (ESSI), Beijing University of Technology (BJUT). His research interests include embedded FPGA system design and multi-core processors.