Topology exploration for energy efficient intra-tile communication

With the technology nodes scaling down, the energy consumed by the on-chip intra-tile interconnects is beginning to have a significant impact on the total chip energy. The segmented bus template is an energy efficient architecture style for the on-chip communication between the components. To achieve the minimum energy operation, the netlist topology of the segmented bus should however be optimized accordingly. In this paper we present a strategy for the definition of an energy optimal netlist for segmented buses. An initial floorplanning stage provides information about the eventual lengths of the interconnect wires and a subsequent exploration step defines the optimal topology for the communication architecture. We motivate that a star topology generated using the wire length prediction can be up to a factor 4 more energy efficient compared to standard linear bus topologies.

I. INTRODUCTION

Scaling to Deep Sub-Micron technology nodes is shifting the major energy consumption contribution from the active area to the on-chip interconnect wires, due to the deficient scaling of the wires. Thus the communication networks which are mainly composed of long interconnects can become a significant portion of the total chip energy consumption. Percentages of up to 40% have been reported [1] [2].

Because the interconnect energy is directly influenced by the wire length, which is decided during the placement and routing phases of physical design, many efforts focus on these lower levels of the design trajectory to optimize the interconnect energy. The most popular methods are revolving around the activity aware floorplanning techniques [3]. The wire energy is proportional to the product of the wire length and the activity of the wire. The basic idea of the activity aware floorplanning is to minimize the activity weighted wire length to achieve the globally energy optimal placement and routing.

However it is not enough to only optimize the interconnects at the physical design stage. Predictions about the interconnects at higher abstraction levels can help to limit the search space for the lower level exploration, without removing the optimal solutions. During high-level synthesis, the interconnect energy optimization is normally based on the estimation of the interconnect wirelength [4] [5]. The well known Rent's rule is used for standard cell based designs, which estimates the number of pins of a logic circuit given the average number of pins per cell and the number of cells. Then the average wire length is predicted using techniques such as the hierarchical placement of the cells. The target is to gain a fundamental insight in the average wire length distribution during logic synthesis.

With the increasing design complexity, macro block based design with reuse of IP blocks is becoming more popular in order to reduce the time-to-market. In a macro block based design, the lengths of the interconnects between the macro blocks are easier to predict at a higher abstraction level, be-cause much fewer connections exist than those in standard cell based designs. By pre-placing the macro blocks, the Manhattan distance is an accurate enough estimation for the prediction of the length of the global wires. The accurate estimation of the global wire length makes it possible to evaluate the energy consumption based on the individual interconnect length estimation, as opposed to the statistical nature of Rent's rule. To achieve the wire length estimation, an estimation of the floorplan is required to determine the relative locations of the macro blocks and the distance between any two blocks.

Our design domain focuses on one tile in the System-On-Chip (SOC) system (Fig. 1). The SOC system in our target domain is typically composed of a mass storage memory block and a set of other tiles. The mass storage memory dominates the chip area. The other tiles are clusters which consist of distributed processing elements and a heavily distributed local memory organization. For energy purposes, the number of the local memories can go up to dozens for real life designs [1]. We target the global wires which transfer the data between the memories and the PEs inside the tile. They comprise the intra-tile communication network. We do not consider the local interconnects inside the macro blocks, they are optimized by the designers of the macro blocks.
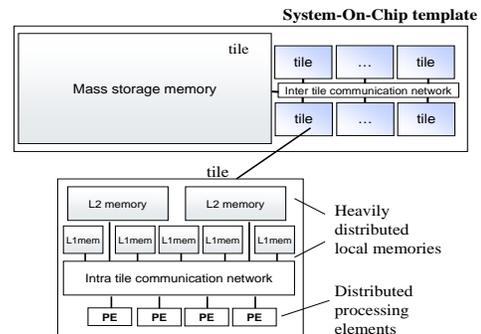

Fig. 1. SOC template for the embedded system

Various communication architectures for the macro blocks in the SOC template have been studied, such as the shared bus, the segmented bus [6] [7] [8], multiple-bus network (based on crossbar structures) and the Network-On-Chip (NOC) [9]. The software controlled segmented bus is a very energy efficient architecture (Fig. 2). It partitions the bus into a large number of segments using switches and the minimal communication paths are used by controlling the switches. Only the segments that are required are activated during each data transfer, thus isolating the activity and the switching to the minimum number of segments necessary. As a result, the communication energy can be significantly reduced, even thought a small area penalty exists due to the insertion of the switches. The switches are asynchronous and software controlled, so the end-to-end transfer over the bus takes a single cycle. The energy consumed by the switch configurations is about an order

of magnitude lower than the energy gain obtained by using the segmented bus [10], so we will not elaborate further on the control plane issues in this paper. Our focus is on the data plane.
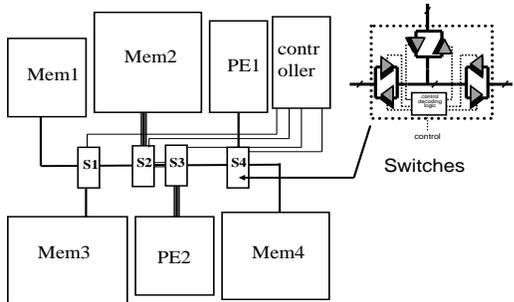

Fig. 2. Segmented bus architecture

II. MOTIVATING EXAMPLE

The netlist topology of the segmented bus describes how the switches are connected to each other. They can be connected in series which results in a linear topology or in a tree-like manner which yields a star topology (see Fig. 3). The energy consumption of each topology is proportional to activity multiplied by wire length for each segment and can vary significantly based on the connectivity. In Fig. 3 we assume all the memories have the same activity α and each segment has the same wire length L. The linear connection has an energy cost of $Cost = \alpha \times (3L + 4L + 5L + 6L) = 18\alpha L$. The star connection only has a cost of $Cost = \alpha \times (3L + 3L + 4L + 4L) = 14\alpha L$. In general, a star topology can minimize the average number of segments each transfer needs to activate. Thus, the total wirelength per transfer will be decreased. Furthermore, it enables the connection of more blocks at a distance of one "hop" (segment) from the processing element (PE), which is good for energy.
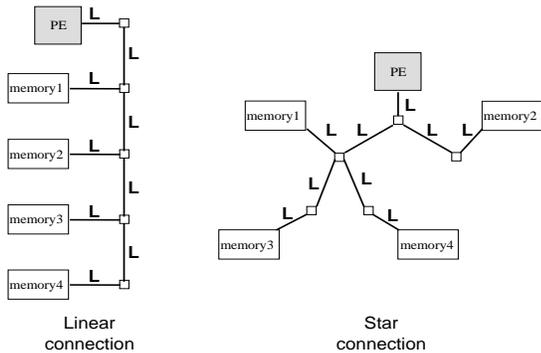

Fig. 3. Netlist decisions influence the communication cost

However in a real design, the memory activity can vary a lot and the wirelengths of the segments are also quite different, because they are determined by the locations of the switches/memories/PEs. To design a globally optimal netlist, we need to consider these two factors simultaneously. This necessitates the introduction of a floorplan estimation step before the connectivity of the netlist is finalized. In Section IV, we will introduce the segmented bus and its netlist topology in more detail and present our methodology to implement the energy optimized segmented bus architecture in real life designs.

III. RELATED WORK

Alternative on-chip communication network topologies have already been exploited. Various bus architectures such as SPIN [11], AMBA bus [12] and FLEXBUS [13] (a dynamical configurable topology performed on AMBA bus) have been designed to improve the system performance. These architectures do not take the physical effects into account and they only perform architecture exploration for performance. Ekpanyapong et al. [14] have introduced a profile-guided micro-architectural floor planner which considers both the impact of the physical wire length and the architecture behavior to reduce the latency of frequent routes inside a processor.

The interconnect energy estimation step has been integrated into high-level synthesis to have an early sight on the global power dissipation. Buyuksahin et al. [4] have presented an approach to estimate the interconnect power during synthesis based on Rent's rule for standard cells with an average error of 25.8%. Lin Zhong et al. [5] have presented a methodology to predict interconnect power consumption in high-level synthesis by taking physical design into account. They floor-planned the RTL datapath units, assumed the data transfer between the datapath units be center-to-center and then estimated the wire length. Jimenez et al. [15] on the other hand, have introduced a placement methodology for power optimization of macro block-based VLSI layouts for a fixed topology, by means of reducing the capacitance of highly active networks.

Pasricha et al. [16] have introduced an automated flow to generate values for bus architecture parameters such as bus widths and bus speeds to meet the performance constraints, ignoring energy consumption. They use a floorplanning engine to estimate the wire length, and calculate the delay to check for bus cycle time violations. They use the AMBA bus architecture [12] consisting of a high performance AHB bus and a low bandwidth APB bus. It is very coarsely segmented, in contrast to our segmented bus architecture which is much more heavily segmented. This technique is an incremental performance optimization for the AMBA bus. We exploit the fine grain segmentation of our communication architecture and the topology itself (by changing the netlist connectivity) to find the energy optimal solution.

In the paper of Chen et al. [7], the implementation of the segmented bus to reduce the network energy has been demonstrated. They use the Gomory-Hu method to generate the bus topology at the architecture level. However, they only minimize the number of the traversed bus segments for the data communication, thus only the switching frequency of the wire was considered. The physical properties of the wires are not taken into account. [6] presents a methodology to optimize the on-chip segmented bus by two steps: the linear netlist optimization and the activity aware floorplanning. The netlist is fixed only based on access activity without low level physical information using a linear topology. Our paper is a further extension of [6]. In this paper, we provide a better coupling between the communication topology exploration and the physical design steps in order to achieve further energy optimization.

IV. NETLIST OF THE SEGMENTED BUS ARCHITECTURE

The segmented bus architecture introduces many switches in the netlist. The connectivity between them defines the topology of the communication network. In order to find the topology that optimizes the metrics, such as critical path delay, energy consumption or even area occupation, an exploration step

is needed. Especially, the network energy is closely related to the netlist topology. It decides the communication path for each data transfer. Furthermore, it also has an impact on the activation frequency of each segment for very segmented architectures like the one in Fig. 2. A well designed topology together with an efficient placement and routing, should produce a solution where the short segments are frequently used and the long segments are seldom accessed.

An energy aware netlist for the communication network can be decided at relatively high level without physical/geometrical information. For instance, a linear netlist topology based on the memory activities for the segmented bus can be identified at high level (the left graph in Fig. 4). After the netlist topology decision, the design is mapped to the layout (the right graph in Fig. 4). This netlist decision reduces the number of segments that need to be traversed for the data transfers which have the highest activities. Though the wire length for each segment is not known yet, it is clear that such decision reduces the energy consumption. Such an approach can achieve up to 44.6% of estimated energy gains compared to a netlist ordered randomly [6].

However, this approach has two major drawbacks. Firstly, due to the fact that the linear topology is 1-dimensional, while the layout is 2-dimensional, the mapping step from the netlist to a floorplan is not quite compatible and consistent in the geometrical point of view. As a result, the wires after routing are usually not as short as they can be. Secondly, this approach fixes the netlist without the physical level information, such as wire lengths. Therefore, an accurate enough estimation of the communication energy during the exploration phase of this approach is not possible. Hence the real globally energy efficient solution cannot be found.
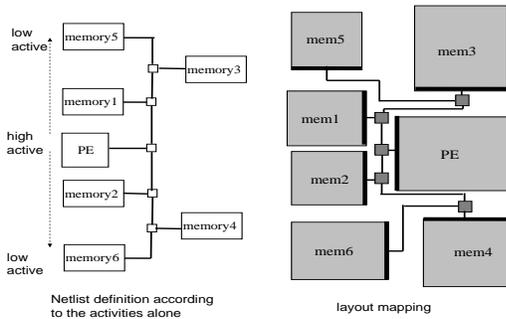

Fig. 4. Linear netlist for segmented bus

In this paper, we propose two extensions to improve the energy efficiency of the previous approach. The netlist is extended from a linear to a tree structure, which fits better the 2-dimensional nature of floorplanning. Furthermore, we introduce a floorplanning estimation step before the final netlist is fixed. This step will provide accurate information about the length of the wires needed to interconnect the blocks and the switches. As presented in Fig. 5, at first, a floorplan is generated according to the blocks' communication activities: the highly active memories are placed close to the PE and the less active memories are placed farther away following an activity aware template for floorplanning. Then, the four-port switches are added into the floorplan and connected between them in an energy optimal manner. The connection topology is a binary tree structure. The PE is the root node and it has two

child nodes memory1 and memory2, which satisfy the two requirements to be chosen: high access frequency and adjacent to the root node physically in the floorplan. Each of the child nodes can have at most two of their own child nodes. In case of multi PEs, multi root nodes are made and each of them is corresponding to one of the PEs.
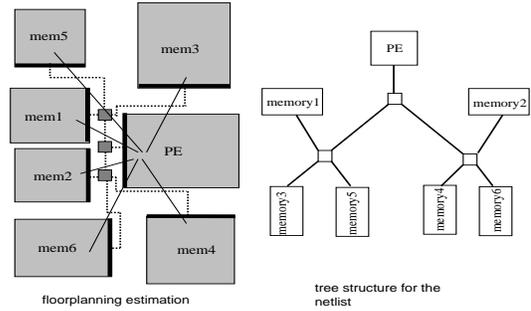

floorplanning estimation   tree structure for the netlist

Fig. 5. Energy optimal tree topology for segmented bus based on activity and layout information

V. THE OPTIMAL COMMUNICATION SPANNING TREE PROBLEM

Finding the optimal communication topology for any optimization criterion in the context of this paper can be tackled using the Optimal Communication Spanning Tree (OCST). The solution to the OCST problem [17] consists of finding a spanning tree that connects all the given nodes and satisfies their communication requirements for a minimum total cost.

A. Problem formulation

Assume a weighted and undirected graph $G = (V, E)$, where $n = |V|$ denotes the number of nodes and $e = |E|$ denotes the number of edges of the graph (Fig. 6). Communication might take place between any of the n nodes, thus communication paths may exist between all the communicating nodes. The amount of data transferred between nodes is specified by an $n \times n$ demand matrix $R = (r_{ij})$, where r_{ij} is the communication activity between node i and node j. $r_{ij} = 0$ if i=j. Additionally, an $n \times n$ distance matrix $W = (w_{ij})$ specifies the distance weights (which correspond to the physical distance between the ports of the two components on the floorplan). They are associated to the edge $e, e \in E$ for each pair of components. The weight w_{ij} is equal to zero, if $i = j$ or if no edge exists between node i and node j in the graph.
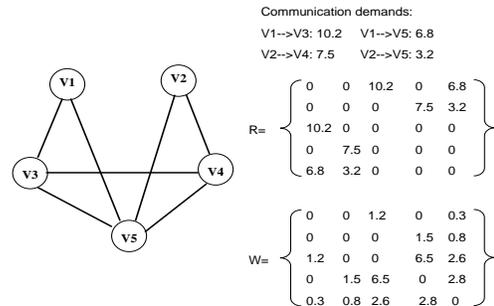

Fig. 6. The weighted and undirected graph

A tree $T = (V, F)$ in which $F \subseteq E$ and $|F| = |V| - 1$ is called a spanning tree of G if it connects all the nodes. In the spanning tree, a unique path from node i to node j exists, which is composed of a set of edges connecting node i

and node j directly or indirectly. This set is defined as the communication path set P_{ij}. The communication cost from node i to node j is the product of the communication distance $\sum_{E_{mn} \subseteq P_{ij}, m,n \in V} W_{mn}$ and the communication demand r_{ij}. The communication cost of the whole spanning tree T is

$$Cost_T = \sum_{i,j \in V} \left(\left(\sum_{E_{mn} \subseteq P_{ij}, m,n \in V} W_{mn} \right) \times r_{ij} \right) \quad (1)$$

T is the optimal communication spanning tree if $Cost_T \leq Cost_{T'}$ for any other spanning tree T'.

Like other constrained spanning tree problems, the OCST problem is NP-hard. In the OCST problem, the weight of one edge (the amount of the communication on this edge) varies for the different tree structures, because the communication of other nodes might also use this edge as the part of the path. The tree structure decides how often this edge is used during the communication. In other words, the weight of the edge is dynamic. As a result, the popular optimization algorithms such as Kruskal's algorithm and Prim's algorithm for fixed weight spanning tree are not appropriate to solve the OCST problem. A large number of evolutionary algorithms using different types of search operators and representations have been proposed for solving the OCST problem [18] [19]. Although the evolutionary algorithms can find a solution very close to the optimal one, they are not time efficient. In general, trade-offs exist between design time and design quality. Especially in modern VLSI design, reducing the time-to-market is a key issue of concern for designers. An efficient approach to obtain the near-optimal solution will be preferred for most of the commercial products' designs. In the following section, a dedicated greedy method will be introduced to generate a topology that is a near-optimal solution for energy based on the binary tree structure efficiently. The quality of the greedy method is evaluated in Section VII.B.

B. Greedy solution

Starting from the weighted and undirected graph $G = (V, E)$ we divide the node set V into the set P, which contains the root nodes, and the set M containing the rest of the nodes to be connected. Table I illustrates our greedy algorithm. At first, we evaluate the connections from any node X in M to any node Y in P. The evaluation cost function is based on the entries in the demand matrix R and the distance matrix W corresponding to each pair of nodes. The cheapest connection between any node in the two sets is found, assume it connects node $\tilde{X}$ in set M with node $\tilde{Y}$ in set P. Node $\tilde{X}$ is added to set P and deleted from set M. The above step is repeated until set M becomes empty.

In the implementation of the greedy strategy in out design approach, the initial set P is the set of PEs and set M is the set of memories. The evaluation cost function is set as $Cost_{edge} = Comm.\ length/Comm.\ activity$. This corresponds to dividing the entry of the distance matrix with the entry of the demand matrix. By applying this cost function, the memories which are highly active and close to the PEs have the highest priority to be chosen to connect to the processing elements first. This intuitively favors energy optimization since the very active memories will get shorter connections. The constraint of the connection degree (the number of the edges

TABLE I
GREEDY ALGORITHM

Greedy Algorithm:
P = push(root nodes);
M = push(the other nodes);
N = Degree constraint;
Initialization of demand matrix R and distance matrix W
While (M is not empty);
for each node X in M do
for each node Y in P do
if the node degree of Y is less than N
calculate the cost of connecting X to Y:
$Cost_{connection} = W_{XY}/R_{XY}$
end if
end for
keep the minimum cost connection $\tilde{X} to \tilde{Y}$;
end for
add $\tilde{X}$ to P;
increase the node degree of $\tilde{Y}$ by 1;
delete $\tilde{X}$ in M;
end while

one node can have) N is determined by the number of the ports of the switches. In this paper we use the four-port switches, which result in binary tree topologies. This algorithm can also find solution of higher degree, but as we will see in the results, the energy consumption of the binary tree is already very close to the theoretical optimum.

VI. SYSTEMATICAL DESIGN FLOW

This section introduces our methodology of tool support which optimizes the netlist topology for the segmented bus architecture. After the high level synthesis step, the system comprises IP blocks like memories and PEs. The communication characteristics such as the access frequency, the master and slave of each communication are identified in the RTL description. The activities of the communication from the master to the slave are specified as the communication demand matrix $R = (r_{ij})$ mentioned in Section A. The physical characteristics of the macro blocks such as the width and the height are also fixed, which is realistic for our tile domain with many small memories (up to 50) and a small number of customized datapath macros. Before deciding on the topology of the communication architecture, we add an activity aware floorplanning step in order to obtain a realistic estimate of the distance between the various blocks on the final layout, see Fig. 7. The activity information we use is the Master-to-Slave communication frequency. At this stage this abstract representation of the inter-block activity is sufficient. It specifies the frequency that each two blocks need to talk to each other. The public domain floorplanner Parquet [20] is adapted to perform the floorplanning estimation step. It can minimize the activity weighted wire length.

In our approach, we import the Master-to-Slave communication demand matrix $R = (r_{ij})$ into Parquet and let the floorplanner minimize the total activity annotated Manhattan distance from each master to each slave. After this step, the highly active memories are placed close to the PEs as a result of the optimization. With the floorplan we obtain, the distance matrix $W = (w_{ij})$ (see Section V.A) can be constructed by calculating the distance from the master's communication ports to the slave's communication ports. Based on the demand matrix R
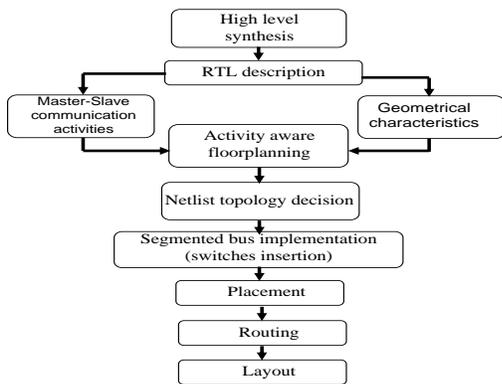
Fig. 7. The framework of the netlist optimization for the low energy segmented bus architecture

and the distance matrix W, the netlist topology is fixed applying the greedy algorithm introduced in section V.B. The topology can be a linear connection or a binary tree or any other tree structure. It depends on the degree constraints given to the topology generation. After the netlist generation, we add the necessary switches to the floorplan in order to implement the segmented bus architecture. The locations of the switches have an impact on the later routing step, thus influence the final energy consumption. Because the netlist topology decision is based on the distance matrix W, which specifies the distance between the block communication ports, we insert the switches close to each block's communication ports. Such a switch insertion strategy has a limited impact on the distance matrix W, therefore preserves the optimal decision made at the netlist topology generation step. Finally we perform the low level physical design such as the placement and the detailed routing using the commercial tool from MAGMA [21] framework.

The chip area and the wire length for each segment are reported in the MAGMA framework. Using the activity information of each memory from high-level synthesis results, the activity of each segment and the energy consumption of the wires can be calculated. We assume the wires are buffered in a delay optimal way [22]. The energy consumed by the buffers is included in the calculations. We have used the high-level analytical formulas of [22] to estimate energy consumption.

VII. EXPERIMENTAL RESULTS

The experiments shown throughout this section have been performed on two custom designs that implement real-life applications namely the MPEG4 encoder and the Quad-tree Structured Different Pulse Code Modulation (QSDPCM) video encoder.

A. Comparison against state-of-the-art

To assess the impact of using geometry information during communication synthesis and the impact of the topology decisions, we compared four different approaches in Fig. 8. The communication network of the QSDPCM video encoder design is used for this evaluation. We evaluate these four approaches without performing the detailed routing step. The Half Perimeter Wire Length HPWL (Manhattan distance between the ports to be connected) is used as the estimation of the real interconnect length, and the cost is calculated via the equation $Cost = HPWL \times Activity$, which is proportional to the wire energy consumption, neglecting the energy consumed by the buffers.

We use the activity aware floorplanning techniques [20] to optimize the Master-to-Slave connection cost (see Section VI). Four floorplans are tested for each case to make sure that the unpredictability of the simulated annealing approach implemented in the floorplanner does not influence the results. For each floorplan, four netlist topologies are designed. The first topology we use is the linear bus topology only based on the activities of the components (introduced in Fig. 4). The second and the third approach use the design flow illustrated in Fig. 7 to generate the netlist. The difference between these two approaches is that the former uses a linear topology while the latter uses the binary tree structure. The last column in Fig. 8 corresponds to the theoretically minimal energy consumption for the given floorplan. We obtain it by multiplying the Manhattan distance between all the communicating pairs of blocks by the respective activity and then summing for all the pairs of communicating blocks. It can be visualized as an ideal case of point-to-point (P2P) connection scheme, without the area overhead and the wire congestion issues of the real P2P topologies.
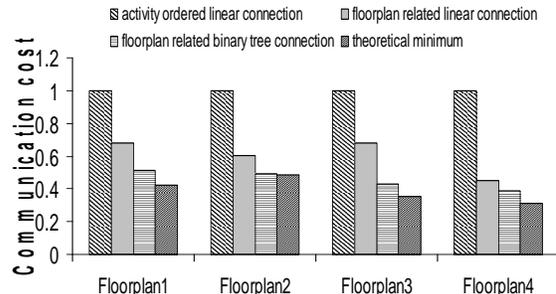

Fig. 8. Communication cost for different netlist topologies at high level

The second approach (linear topology with geometry information) which followed our methodology introduced in Section VI can reduce nearly 40% energy cost in average compared to the first (linear topology without geometry information). Comparing the third approach to the second approaches, the binary tree structure can reduce the energy cost even further compared to the linear connection approach. It incurs an overhead of only about 10% in average compared to the minimum energy reference. These results indicate that our methodology performed on the binary tree topology is a better candidate than the linear topologies for energy optimal communication, if the communication architectures allows to control which parts of the architecture are activated per transfer.

B. Optimality of greedy solution

Table II illustrates how far the greedy approach is from the optimal solution for the binary tree topology. We have used an exhaustive search method to enumerate all the possible binary tree structures and find the one which has the lowest associated abstract (energy) cost. This method has been tested on small netlists, because the complexity of the exhaustive search increases exponentially with the increasing number of communicating components. We have used parts of the MPEG4 design. Each part corresponds to an individual kernel of the MPEG4 encoder: Motion Estimation (ME), Motion Compensation (MC), Text Coding (TC), Variable Length Coding (VLC). Each kernel is mapped on a processing element with a local memory organization comprising 6 ∽ 9 memories.

The design flow illustrated in Fig. 7 is used in both cases. In order to compare the two approaches, greedy and optimal, we plug the appropriate algorithm in the Netlist topology decision step. The CPU time for the execution of the two algorithms is provided in Table II. In each experiment for the exhaustive approach, we use the same floorplan that was used for the greedy approach and then enumerate all the legal binary tree structures to find the one with the lowest cost. This step takes a few minutes up to several hours depending on the number of memories in the netlists. To estimate the reported energy consumption, see Table II, we perform the remaining physical design steps using the communication topology decided by the exhaustive and the greedy approaches.

TABLE II
GREEDY APPROACH VS. EXHAUSTIVE APPROACH

| | Exhaustive approach | | Greedy approach | | Energy overhead (%) |
	Energy arbitrary unit	CPU time	Energy arbitrary unit	CPU time	
ME	5.6	258 min.	5.94	1.5 sec.	6.1%
MC	3.52	236 min.	3.68	1.5 sec.	4.6%
TC	1.04	312 min.	1.12	2 sec.	7.3%
VLC	1.37	5 min.	1.42	0.5 sec.	3.8%

The greedy approach has less than 7.3% energy overhead compared to the exhaustive approach which finds the optimal netlist topology for energy. The greedy approach is time efficient, especially in case the number of communicating components is large, like the SOC template which might have dozens of macro blocks. Its energy overhead is small because of the design flow we are using (Fig. 7). The floorplanning estimation step has already placed the highly active memories close to the PEs and the less active memories farther away from the PEs. At the netlist topology decision step, it is intuitive that connecting the high active and adjacent memory first to the PE is mostly probably the best choice for the global energy optimization. Thus the greedy approach of always taking the next local minimum yields results very close to the global optimum.

C. Comparison of binary tree and linear topologies

Finally we have compared our methodology results to the conventional linear netlist for segmented buses on the real-life netlists coming from the high-level synthesis of custom designs for the QSDPCM and MPEG4 applications in Fig. 9. Three approaches are tested. In the first approach, the netlist is fixed based only on the communication activities and area is minimized at the floorplanning stage. The second approach has the same netlist as the first approach, but uses the activity aware floorplanning to optimize the network energy. The third approach follows the proposed design flow (Fig. 7) using the activity aware floorplanning to obtain the floorplan and using the greedy approach to fix the binary tree netlist. It has the same floorplan as the second approach. The normalized energy consumption is calculated after physical design. The activity aware floorplanning approach (the second approach) network energy is around half of the energy of the area optimal solution (the first approach). The tile area overhead is less than 15% in average (Fig. 9). Comparing the second approach with the third approach, nearly an additional factor of 2 for the energy gain is achieved by introducing the binary tree topology. Therefore, by generating the optimal netlist topology, the third approach can improve the network energy significantly

at a reasonable tile area penalty. That penalty is even smaller at global chip level because the area is dominated by the large memory blocks (Fig. 1).
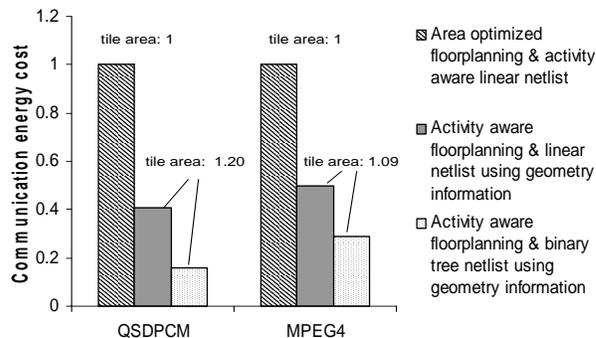

Fig. 9. Energy comparison of linear connection and binary tree connection

VIII. CONCLUSIONS

We have introduced a methodology to design the netlist topology for the implementation of the segmented bus communication architecture. We have demonstrated that a factor of 2 for energy gain is achieved applying our methodology on the linear netlist compared to state-of-the-art implementation of the linear topology. An additional factor of 2 energy gain can be achieved by using a binary tree topology.

REFERENCES

[1] "blind review."

[2] "blind review."

[3] H. Jingcao, D. Yangdong, and R.Marculesu, "System-level point-to-point communication synthesis using floorplanning information," in *Proc. ASP-DAC 2002*, Jan. 2002, pp. 573–579.

[4] K.M.Buyuksahin and F. Najm, "High-level power estimation with interconnect effects," in *Proc. ISLPED*, 2000, pp. 197–202.

[5] L. Zhong and N. Jha, "Interconnect-aware high-level synthesis for low power," in *Proc. ICCAD-2002*, 2002, pp. 110–117.

[6] "blind review."

[7] J. Y. Chen, W.B.Jone, J.S.Wang, and T. H.-I.Lu, "Segmented bus design for low-power system," *IEEE Trans. VLSI Systems.*, vol. 7, no. 1, pp. 25–29, Mar. 1999.

[8] W.-B. Jone, J. S. Wang, H. I Lu, I. P. Hsu, and J.-Y. Chen, "Design theory and implementation for low-power segmented bus systems," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 8, pp. 38–54, Jan. 2003.

[9] L. Benini and G. D. Micheli, "Networks on chips:a new soc paradigm," *IEEE computer*, vol. 35, no. 1, pp. 70–79, Jan. 2002.

[10] "blind review."

[11] A. Adriahantenaina, H. Charlery, A. Greiner, L. Mortiez, and C. Zeferino, "Spin: a scalable, packet switched, on-chip micro-network," in *Proc. Design, Automation and Test in Europe Conference and Exhibition, 2003*, 2003, pp. 70–73.

[12] Arm amba bus specification. [Online]. Available: http://www.arm.com/armwww.ns4/html /AMBA?OpenDocument

[13] K. Sekar, K. Lahiri, A. Raghunathan, and S. Dey, "Flexbus: a high-performance system-on-chip communication architecture with a dynamically configurable topology," in *Proc. Design Automation Conference,2005.42nd*, June 2005, pp. 571–574.

[14] M. Ekpanyapong, J. Minz, T. Watewai, H.-H. Lee, and S. K. Lim, "Profile-guided microarchitectural floor planning for deep submicron processor design," *IEEE Trans. Computer-Aided Design*, vol. 25, pp. 1289–1300, July 2006.

[15] M.A.Jimenez and M.Shanblatt, "Integrating a low-power objective into the placement of macro block-based layouts," in *Proc. MWSCAS 2001*, Aug. 2001, pp. 62–65.

[16] S. Pasricha, N. Dutt, E. Bozorgzadeh, and M. Ben-Romdhane, "Floorplan-aware automated synthesis of bus-based communication architectures," in *Proc. Design Automation Conference,2005.42nd*, June 2005, pp. 565–570.

[17] T. Hu, "Optimum communication spanning trees," *SIAM Journal on Computing*, vol. 3, pp. 188–195, Sept. 1974.

[18] Y. Li and Y. Bouchebaba, "A new genetic algorithm for the optimal communication spanning tree problem," in *Proc. Artificial Evolution: Fifth European Conference*, 1999, pp. 162–173.

[19] D. Peleg and E. Reshef, "Deterministic polylog approximation for minimum communication spanning trees," in *Proc. ICALP, Lecture Notes in Computer Science*, 1998, pp. 670–679.

[20] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning through better local search," in *Proc. International Conference On Computer Design (ICCD)*. Austin, 2001, pp. 328–333.

[21] "Blast chip 4.0 user guide," *Magma Design Automation*, pp. 271–351.

[22] J. M. Rabaey, *Digital integrated circuits: a design perspective*. Prentive Hall: Upper Saddle River (N.J.), 2003.