Physical design implementation of segmented buses to reduce communication energy

Jin Guo[1,2], Antonis Papanikolaou[1], Pol Marchal[1], Francky Catthoor[1,2]
[1]IMEC v.z.w., Kapeldreef 75, 3001 Leuven, Belgium
[2]Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

Abstract— The amount of energy consumed for interconnecting the IP-blocks is increasing significantly due to the suboptimal scaling of long wires. To limit this energy penalty, segmented buses have gained interest in the architectural community. However, the netlist topology and the physical design stage significantly influence the final communication energy cost. We present in this paper an automated way to implement a netlist consisting of hard macro blocks, which are interconnected with heavily segmented buses in an energy optimal fashion for communication. We optimize the network wires energy dissipation in two separate, but related steps: minimizing the number of segments for active communication paths at the first step (block ordering), followed by the activity aware floorplanning step to minimize the physical length of these segments. Energy gains of up to a factor of 4 are achieved compared to a standard system implementation using a shared bus. Especially, the block ordering step contributes significantly to the network energy optimization process.

I. Introduction

Energy consumption is becoming one of the major optimization targets when designing low-power embedded systems. An important way to reduce the energy cost is to introduce a distributed memory hierarchy [15]. However, due to the increasing number of IP blocks (many dozens), more and longer wires are required to interconnect all of them. The trend toward IP reuse will also push toward the efficient reuse of more hard macro-blocks. Hence, the netlist connecting many hard IP blocks should be efficiently implemented.

The performance and the energy consumption of global wires cannot follow that of the transistors as technology feature sizes scale down. The energy dissipation of those wires is almost not improving anymore, with each new technology node, while transistor capacitance keeps shrinking [2]. As a result, the relative energy consumption of the communication is increasing compared to the computation and the storage system components. The results in [6] show that the energy consumed by the communication network is comparable to the energy consumed by the heavily distributed memories.

At the architectural level a lot of studies exist already for communication energy minimization. Chen et al. [9] have demonstrated how to reduce the energy consumption of the communication by using a segmented bus architecture, which can shut off the unused path via switches. The energy costs of driving the switches for a segmented bus are appreciable lower than the gain obtained from using a segmented bus [16]. The actual wire energy consumption of this architecture, however, depends on the floorplan of the system. The reason is that energy is proportional to the product of wirelength multiplied by the activity of each wire. The activity can be minimized at higher abstraction levels, but the actual length of each interconnect wire is only decided during physical design. Thus, support is needed from the physical design phase in order to achieve minimal communication energy consumption.

Current physical design flows, however, mostly focus on the optimization of area occupation and total wirelength of the design. Total wirelength is probably a good metric to represent wire congestion, but it is not adequate to reflect the energy consumption since different wires can have very different activities.

The use of heavily segmented buses introduces an additional issue to be solved. The number of segments should indeed increase compared to the industrial bridged or segmented bus architectures [19] [18], in order to reduce the energy cost. Then the order in which the blocks are connected to the bus heavily influences the communication energy. For example, a netlist with optimal connectivity can assure the very active paths are always shorter in length than the less active paths during the physical design, which reduces energy consumption intuitively. Hence, such netlist topology problems together with the activity aware floorplanning techniques need to be efficiently coupled to optimize the communication network energy.

In this paper we show how communication energy optimization can be achieved on real application drivers. We couple an energy-optimal communication architecture, connection ordering to an automated energy-aware physical design flow to show that communication energy can be minimized by up to a factor of 4 compared to shared buses architecture with a standard physical design flow. We also stress that the network energy is very sensitive to the netlist topology in terms of block ordering.

II. Related Work

Reducing the communication power dissipation has increasingly become a key concern in SOC design. Many reported approaches have focused on the savings at architecture level. Chen et al. [9] first demonstrated how the segmented buses can improve communication power and critical path delay. However, they evaluated the wire energy based on the estimation of the wire length, without doing the floorplanning. Hence the results lack of real physical level effects. Furthermore, they calculated the wire energy consumption without buffers to neglect the circuit level issues. But buffers contribute significantly to the overall energy consumption and latency.

A lot of work has been done in the physical design community to optimize chip area and improve circuit performance [10] [11]. To utilize the large impact of physical design

0-7803-9451-8/06/$20.00

on total system energy consumption, researchers have looked into power optimization at the floorplanning stage. Chao et al. [3] have introduced a floorplanner which optimizes the module power consumption and chip area by choosing the specific shape for each module. The wire energy dissipation is neglected. Several other approaches have been introduced to include a low power objective in physical design. Prabhakaran et al. [13] presented a simultaneous scheduling, binding and floorplanning algorithm to minimize interconnect energy dissipation. They contributed to the combination of high-level synthesis and physical design for interconnect power reduction, but not targeting macro blocks.

Jingcao et al. [7] have proposed a new methodology to generate low energy and high performance communication networks at the floorplanning stage. Their methodology is based on a point-to-point connection architecture. This architecture is not suitable for connecting many macro blocks because it can lead to routability and wire congestion problems. Jimenez et al. [12] presented an activity aware placement methodology, which aimed at reducing the power dissipation of macro block-based VLSI design. They mainly introduced how to implement the simulated annealing algorithm efficiently. They did not specify the communication architecture.

To the best of our knowledge, the impact of floorplanning on the energy consumption of heavily segmented buses has not been studied in the past. In sectionV, we will therefore outline an exploration environment for investigating this problem in two steps of optimization. First, we explain the target architecture more in detail.

III. Platform Description

Our target domain is that of application domain specific embedded portable systems. We focus on multimedia and wireless applications. This target implies a number of architectural assumptions that we can exploit. Systems of this domain will follow the System-on-Chip template and they will consist of different tiles, among which one will be the mass storage memory which dominates the system area occupation. In this paper we focus on the implementation of a given tile. The implementation of the complete system can be done in the same manner.

A. Memory Organization

These systems consist of application domain specific processors with their local memories (Fig. 1). The processors can be partitioned into distributed processing elements (PEs). The memories are organized in a hierarchical manner for energy optimal purpose [6]. The local memory layer typically consists of many small memories. This heavily distributed memory organization increases the number of macro blocks, to at least dozens but up to hundreds. And the additional available bandwidth from the memories to the processors increases the number of parallel communication paths among the blocks. Therefore, a hierarchal memory organization results in a more complex communication network architecture and requires novel floorplanning techniques.

The PEs and memories are assumed to be pre-designed IP blocks in order to enable IP reuse and thus limit the system development design efforts. The shapes of the blocks are predefined and assumed to be rectangular. The width and height

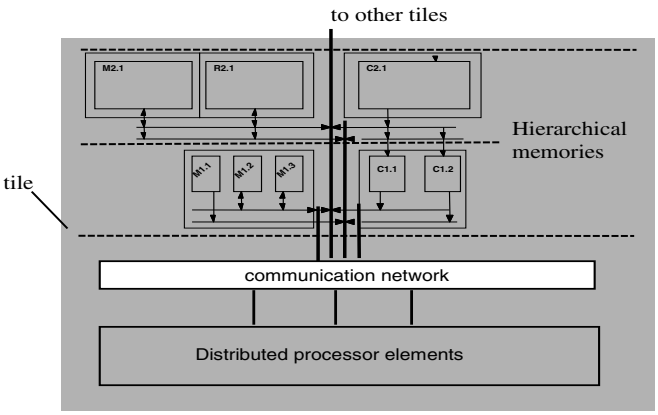

Fig. 1. Memory hierarchy and distributed organization

of the memories are fixed to achieve optimal intra-block energy consumption at the system design phase. During physical design, the physical characteristics of the memories are not changed any more to avoid overruling the already made decisions at higher level.

B. Segmented Buses Network

Due to the large number of IP blocks, interconnecting them becomes a critical issue. Shared buses are conventionally used (Fig. 2-a) instead of the point-to-point connections in order to use less communication wires. The main disadvantage of shared buses is that they are power hungry, since every master has to charge the entire bus for communicating data. In Fig. 2-a, the most frequently accessed path is between memory1 and the processor. During each access the entire bus swings, including connections 1-2-3. Traditionally, segmented buses [9] introduce some switches on the buses (see Fig. 2-b). The switches are 3-port uni-cast or multi-cast components implemented using tri-state buffer chains [14]. They can be programmed in order to configure the needed communication paths and shut off the others. For instance, we can avoid those connections 1-2-3 swing in Fig. 2-b by giving the control signal to the switches to shut off the paths. This way, it reduces the capacitance which toggles when communicating and thus reduces the communication energy consumption. In our case, there are dozens of switching connections to many segments, which implies a large potential energy gain there.

Fixing the netlist of the shared buses is straightforward, where all the blocks connect to the same buses without alternative solutions. When it comes to the segmented buses, additional decisions need to be taken to finalize the netlist of the system. Assuming a linear communication topology, the blocks are connected to the bus one by one in a specific order and no star connections exist. The order of the blocks connected to the bus dictates the activation frequency of the various segments. Hence, the network energy of the segmented buses has a strong dependency on the netlist. This step is actually necessary for any topology of segmented buses. In the following sections we propose a simple way to order the blocks on linear topologies.

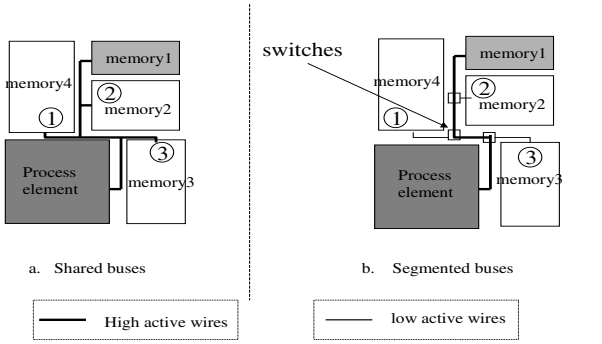

Fig. 2. Segmented buses based on shared connections

IV. Motivational Example

The energy cost of the segmented bus depends directly on the wirelength and the switching activity of each segment. The energy cost can be computed as follows:

$$E = V_{dd}^2 \times C_l \times \sum_i (\alpha_i \times l_i \times bitwidth_i) \qquad (1)$$

We realistically assume that only the global metal layers are used for the implementation of the communication network, so V_{dd} and capacitance per unit of length C_l are constants for a specific technology node. α_i is the number of activations of segment i. l_i is the average wirelength among the wires of a given segment and $bitwidth_i$ is the number of parallel wires of that segment. Conventional physical design flows minimize the total wire length $\sum_i (l_i \times bitwidth_i)$. This reduces wire congestion, but it can lead to energy optimization only when α_i is constant or varies little for the different segments.

In typical embedded applications, the required bandwidth on the different connections from processing elements to memories varies significantly. This translates into large differences in activation frequency among the segments. We have measured a ratio of up to ten between the most active and the least active segments [17]. This large range in activities, means that we cannot neglect it in the optimization cost function. To reduce the energy consumption of the segmented bus, we should minimize the sum of the products of activity times total segment wire length, denoted as $\sum_i (l_i \times bitwidth_i \times \alpha_i)$.

This optimization can be decoupled into two steps: (1) heavily active communication paths should utilize the smallest number of segments for the data transfer (block ordering) and (2) the physical lengths of these segments should be minimized (activity aware floorplanning).

A. Block Ordering

As discussed earlier, we have to deal with up to hundreds of connections, leading to many alternative switches organizations and block ordering. The way the switches are connected to each other introduces the order with which the memories are connected on the buses. The number of segments that need to be traversed to communicate between a given PE and a given memory is determined by the connection ordering of the memories. A good ordering decision results in that the very active transfer paths use a minimal number of segments, which implies low cost. For instance, Fig. 3 illustrates two block ordering decisions. The same architecture is shown comprising two processing elements and 4 memories connected with a different order on the linear communication topology. The different order is the only difference between the two netlists. To illustrate the potential for energy optimization we assume for simplicity in this example that all segments have the same wirelength L and we neglect the wirelengths from memory ports to switches. Some memories communicate to 2 PEs like memory3. The values annotated on top of each memory block in the figure are the number of accesses of each memory to the two PEs individually for executing a given functionality. Furthermore, we assume that the bitwidth is the same everywhere, thus we can neglect it in the qualitative calculations that follow.

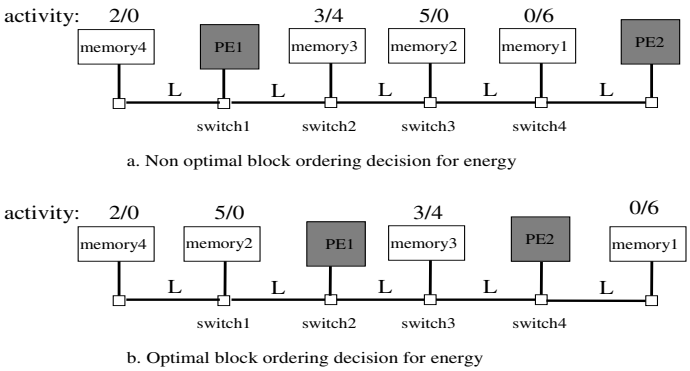

Fig. 3. Block ordering decisions influence communication cost

Considering the energy related communication cost as the product of total wire length from source to sink of communication and activity, the first block ordering decision has a cost of $E = 2\times L+3\times L+5\times 2L+4\times 3L+6\times L = 33L$. The second netlist has a much lower cost, due to a better block ordering. Its cost is $E = 2\times 2L+5\times L+3\times L+4\times L+6\times L = 22L$, a nearly 40% energy reduction. The intuitive principle for energy optimal ordering is that the connection order is made according to the activity order, that is, the most active memories are closer to PE in terms of the number of switches existing in the communication path between these two blocks and vice versa.

B. Activity Aware Floorplanning

Given a netlist with a good block ordering decision, the physical length of each segment should be minimized (step 2) according to their activity in the physical design steps of floorplanning and placement. An activity aware floorplanning technique has been outlined in [8]. The principle is that very active segments should have minimal length, while not so active segments can tolerate a bit longer length. For instance, in Fig. 4, the two floorplans are using the same netlist. The first floorplan does not consider the activity of each segment and aims at minimizing the total wirelength and area occupation. In the second floorplan, we change the positions of the modules such that the wirelength of the most active link is reduced, which should be energy optimal solution for the network. An area penalty is incurred and the total wirelength might be increased a bit. We quantify these overheads in a later section. However, large main on-chip storage memories (not layer1 memories) will dominate the chip area in the embedded systems, thus small area overheads in the area of the individual tiles for

the distributed layer1 memories will be negligible at the level of the entire chip.

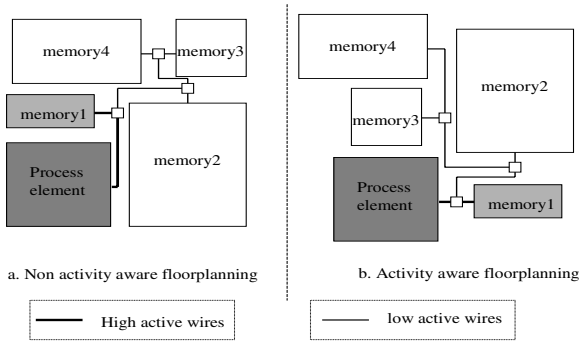

Fig. 4. Floorplanning techniques for segmented buses

In the remainder of this paper we outline an automated methodology, which can produce an optimized implementation of a system consisting of hard macro-blocks interconnected via segmented buses starting from the RTL system description. The main optimization cost is communication energy consumption.

V. Physical Design Flow for Energy Optimal Communication

In order to make an automatic physical design implementation for low energy segmented buses architecture interconnecting hard macro blocks, we introduce our approach in Fig. 5, compared to a conventional approach. After the system design step (or high-level synthesis), the system consists of blocks in terms of many memories and a few processing elements. The conventional approach inputs the RTL description into a placer to get the placement and then performs the routing.

For segmented buses, the RTL description should be extended with the information about how the switch blocks are connected. Normally, the switch blocks are much smaller than the memory blocks. It is extremely difficult for most of the current macro cell floorplanning tools to deal with macro blocks of very different sizes. In that case, manual intervention is need. For instance, some extremely large or small blocks are pre-placed manually before automatically floorplanning the other blocks.

Avoiding the manual placement of the switches while keeping the smart strategy made by the floorplanner, we automatically insert the switch blocks after floorplanning, closely to the ports of the memories/PEs. Since the switches are small, the insertion step does not change the relative locations of the large macros, which have been optimized by the floorplanner already. And since the switches are close to the communication ports, the insertion influence on the communication path is slight. Hence the insertion impact on the cost of the activity aware floorplanning is small.

In our approach, we first make the ordering decision according to the memory activities. The ordering decision is necessary to identify the activity of the segments between two switches. Because switches do not appear at floorplanning stage, an intermediate netlist, activity weighted netlist, is produced. This netlist has no switch description, but is annotated with the activities of point-to-point communications between the macros. These activities are equal to the segment activities after the later switches insertion step. By this way, we use the netlist without switches but still provide enough activity information to physical design for energy optimization. The activity weighted netlist is imported to the public domain floorplanner Parquet [4] to get a network energy optimized placement solution. Then switches are added and the placement is slightly adjusted to accommodate them. In parallel, a new RTL netlist which describes all the macro blocks including the switches in the system after switches insertion is fixed. Finally, the placement and the new RTL netlist are imported to the MAGMA [1] environment for routing. We will explain the major steps in detail.

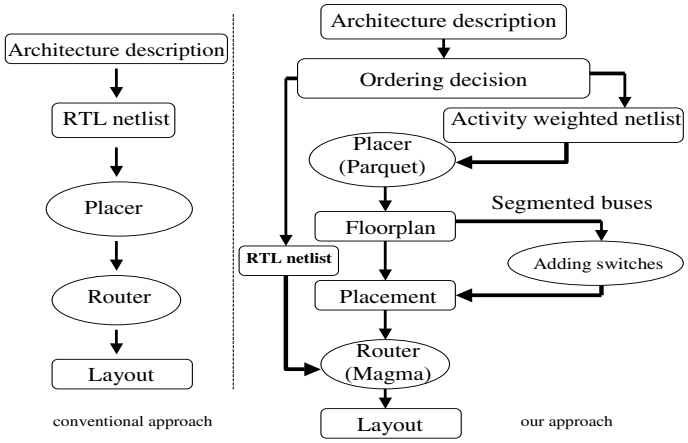

Fig. 5. Exploration methodology fbw-graph

A. Block Ordering Decision

This step aims at minimizing the number of segments those need to be activated for very active communication paths. We currently only deal with linear bus topologies. This is not a severe limitation, since many of the bus-based designs in literature and in industry use linear topologies. This one dimensional topology is already able to illustrate well the gains achieved by the topology decisions, though a two dimensional topology might achieve more gains. Extending the principle to other topologies is future work.

The system components that are involved in most of the data transfers are obviously the processing elements. They fetch or store all the data from and to the individual memories. Thus, we have developed an approach where we start from the connections of the processing elements on the bus and connect the memories close to them based on their number of accesses. The most frequently accessed memories are connected very close to the processing element they communicate with and vice versa. This ordering localizes most of the communication activity in small areas and few segments need to be activated for most of the transfers. Extending this principle to cover multiple processing elements on a single linear bus is trivial. If more than one bus exists in the netlist, we perform the ordering per bus.

B. Floorplanning and Routing

In this stage, we need to implement the activity-aware floorplanning technique in order to minimize the lengths of the active segments in our macro block based netlist. We use Parquet [4] to obtain the floorplan for each design, which is a macro block placement tool. It can minimize the weighted net length and chip area via using a simulated annealing algorithm.

After the ordering decision, the activity of each net can be determined exactly. The activity is annotated to the net as the weight. Normally, the floorplanner minimizes $w1 \times area + w2 \times weighted\_wire\_length$ to achieve a good balance between area cost and network cost. In order to illustrate how much energy gain we will have compared to the shared buses, we set a low weight for area and a large weight for the weighted wire length in the cost function. Due to the fact that the simulated annealing process takes a long time to converge, we run the process for a set of times, with a reasonable CPU time for each run. The best floorplan is selected.

The floorplan generated by Parquet is not routable since the block edges overlap with their neighbors' edges. So it needs to be changed to a legal placement for routing. In parallel, switches need to be added to implement the segmented buses architecture. Again, the switches blocks and memory/PE blocks can not overlap with each other. We do the switch insertion and block separating simultaneously: starting from the left-bottom block, the neighbor block is moved to the right or to the top until a pre-defined space exists between these two blocks. If switches are needed to be inserted there, further movements are going on until enough space is assured for placing the switches.

The MAGMA BlastFusion chip implementation system was used to check the placement and do the detail routing. Only the global metal layers are used for connecting the memories and PEs. Because the bus is segmented, the length of the segments is rather small and buffering is enough. We target low power embedded system whose clock frequencies is several 100 MHZ but less than 1 GHZ. For these two reasons, no repeaters are needed. Hence, the wires can be routed over the blocks. The wirelength of each segment is measured and chip area is reported by the MAGMA tool suite.

VI. Experimental Results

This section explains the way we performed the calculations for communication energy and critical path length. The results of our experiments are also presented here.

A. Network Energy and Critical Path Length

In order to have a better estimation of the real energy consumption of the buses we take into account the required buffers to drive the wires. The wires are buffered in a delay optimal way [5] and all the calculations are made for the 130nm technology node with data coming from the ITRS roadmap [2].

The total network dynamic energy consumption is the sum of all the individual segments energy consumption. We use the critical path wire length as a delay metric. For shared buses, the critical path wire length is equal to the whole bus length in the case of a single bus or the longest bus length for multiple buses. For segmented buses, the communication path length is the sum of the activated segments wire length. So the critical path length is the maximum sum of segment-lengths for any of the communication paths.

B. Results

First, we performed the design on the motion estimation and the motion compensation kernels of the main profile of an MPEG4 encoder. This is an application specific design consisting of two processing elements and 14 local memories.

We have followed two different combinations of communication architecture and physical design approach: shared buses with minimal total HPWL (Half Perimeter Wire Length), and segmented buses with minimal energy. The shared buses implementation follows the conventional design flow shown in Fig. 5. The RTL netlist is imported into Parquet, which generates one floorplan with minimal HPWL. For the second combination, we adopted activity aware floorplanning combined with a netlist which specifies the energy optimal block order for the segmented buses up front.

TABLE I
THREE DESIGN APPROACHES, MPEG4 ENCODER

Buses architecture	area (mm^2)	energy *1e-3(J)	critical path(m)	total WL(m)
Shared buses Min. HPWL	1.3333	0.481	0.0036	0.032
Segmented buses Min. energy	1.423	0.126	0.0019	0.114

Table I presents the chip area, the network energy consumption, critical path wirelength and total wirelength for these two approaches. Compared to shared buses using the conventional approach, significant energy gain is achieved when we use an optimal netlist and the activity aware floorplanning technique based on segmented buses. This combination reduces the wire energy to around one quarter compared to the minimal HPWL shared buses.

We evaluated the segmented buses architecture via our implementation methodology for MPEG4 with two different memory organizations and DAB (Digital Audio Broadcast) receiver with one memory organization in Table II. Similar comparisons of the two designs as in the Table I can be made and the conclusions are the same. In average, the energy optimal implementation of segmented buses can improve communication energy with a factor of 3.1 compared to the shared buses approach.

To assess the impact of the two proposed optimizations in the overall communication energy consumption we have conducted a number of experiments for MPEG4 application. Two different block ordering approaches, a random choice and the optimal choice, were combined with two different floorplanning approaches, a conventional one and the activity aware one. Fig. 6 illustrates the results from the four different experiments. The leftmost bars show the area and energy consumption of a netlist with a random block ordering decision using a conventional floorplanning approach that optimizes area and half-perimeter wirelength with equal weights. The second bar from the left is obtained by combining the same block ordering decision with an activity-aware floorplanning approach. The third set of bars is the result of an optimal block ordering decision (as explained in Section V) and the conventional floorplanning, while the last set of bars are for optimal block ordering and activity-aware floorplanning.

By comparing of the first and third experiments, it is evident that block ordering can have a very significant impact on the resulting communication energy (44.6% reduction). An optimal block ordering decision can by itself reduce the energy, by reducing the number of segments that are activated for very

TABLE II
SHARED BUSES VS. SEGMENTED BUSES FOR DIFFERENT APPLICATIONS

Applications	Memory organization	Communication network	area (mm^2)	energy *1e-5(J)	critical path *1e-3(m)	total WL(m)
DAB	13 memories	Shared buses minimal HPWL	6.567	19.6	2.69	0.105
	3 PEs 3 buses	Segmented buses minimal energy	7.145	5.16	1.51	0.158
MPEG4	9 memories	Shared buses minimal HPWL	5.333	2.08	1.92	0.076
Memory Mapping1	2 PEs 3 buses	Segmented buses minimal energy	5.562	1.16	1.29	0.144
MPEG4	15 memories	Shared buses minimal HPWL	5.193	23.9	2.39	0.101
Memory Mapping2	2 PEs 2 buses	Segmented buses minimal energy	6.107	6.186	1.76	0.149

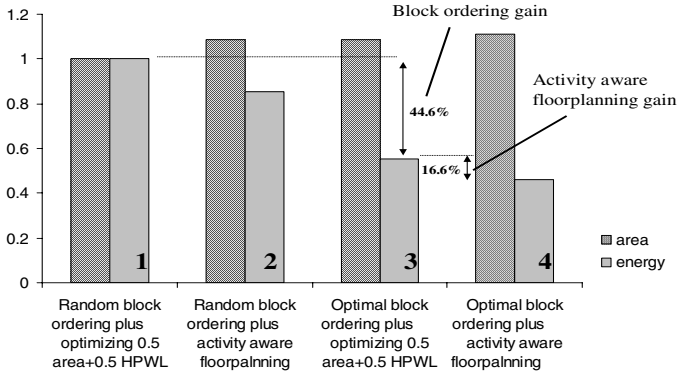

Fig. 6. Impact of block ordering and activity aware fborplanning, MPEG4 application driver

active transfer paths. The comparison of experiments 3 and 4, reveals that impact of activity-aware floorplanning on the final results is relative small (16.6% reduction). Thus, such a floorplanning without an optimized block ordering decision does not provide large energy gains. If the two optimization steps are coupled efficiently, significant energy gains of a factor of 2.16 can be achieved for this design. The area degradation is about 11%. We can conclude that the final network energy cost is much more sensitive to block ordering (netlist topology) than to the activity aware floorplanning techniques.

VII. CONCLUSIONS

We propose a novel automated approach that can implement a hard macro block netlist interconnected by segmented buses while minimizing the energy consumed in the communication network. The results show that this approach can reduce the communication network energy consumption by up to a factor of 4 compared to a conventional physical design stage implemented the netlist interconnected with a shared bus. And we present that the energy consumption of the communication network is high sensitive to the netlist topology decisions.

REFERENCES

[1] "Blast Chip 4.0 User Guide", *Magma Design Automation*, Cupertino, CA 95014, pp.271-351, http://www.magma-da.com.

[2] "International technology roadmap for semiconductors 2001 Edition."

[3] Chao Kai-yuan, D. F. Wong, "Floorplanning for low power designs," *IEEE international Symposium on Circuits and Systems*, Vol. 1, pp. 45-48, May 1995.

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

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

[6] A. Papanikolaou, K. Koppenberger, M. Miranda, F. Catthoor, *Memory communication network exploration for low-power distributed memory organisations*, Proc. IEEE Wsh. on Signal Processing Systems (SIPS), Austin TX, IEEE Press, pp.176-181, Oct. 2004.

[7] Hu Jingcao, Deng Yangdong, R.Marculesu "System-level point-to-point communication synthesis using fborplanning information", *ASP-DAC*, pp. 573-579, 2002.

[8] Hua Wang, A. Papanikolaou, M. Miranda, F.Catthoor, "A global bus power optimization methodology for physical design of memory dominated systems by coupling bus segmentation and activity driven block placement", *ASP-DAC*, pp. 759-761, 2004.

[9] J. Y. Chen et al., "Segmented bus design for low-power system", *IEEE Trans. VLSI Syst* pp. 25-29 Mar 1999.

[10] I. Hui-Ru Jiang, Yao-Wen Chang, Jing-Yang Jou, Kai-Yuan Chao, "Simultaneous fborplanning and buffer block planning", *ASP-DAC*, pp. 431-434, 2003.

[11] Hua Xiang, Xiaoping Tang, M. D. F. Wong, "Bus-Driven fborplanning", *ICCAD*, pp. 66-73, 2003.

[12] M. A. Jimenez and M. Shanblatt, "Integrating a low-power objective into the placement of macro block-based layouts", *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems*, Vol 1 , pp. 14-17 Aug 2001.

[13] P. Prabhakaran P. Banerjee, J. Crenshaw, M. Sarrafzadeh, "Simultaneous scheduling, binding and fborplanning for interconnect power optimization", *Proceedings of VLSI Design*, pp. 423 - 427, 1999 .

[14] A.Papanikolaou, F.Starzer, M.Miranda, F.Catthoor, K.De Bosschere, "Architectural and physical design optimizations for effi cient intra-tile communication", *Proc. Intnl. System-on Chip Symp. (SoC)*, Tampere, Finland, pp.-, Nov. 2005.

[15] P. Panda, F. Catthoor, N. Dutt, K. Danckaert, E. Brockmeyer, C. Kulkarni, A.Vandecappelle, P.G.Kjeldsberg, "Data and memory optimizations for embedded systems", *on Design Automation for Embedded Systems (TODAES)*, Vol.6, No.2, pp.142-206, April 2001.

[16] K. Heyrman, A. Papanikolaou, F. Catthoor, P. Veelaert, W. Philips "Energy costs of transporting switch control bits for a segmented bus', *Proceeding of proRISC 2005*.

[17] E. Brockmeyer, J. D'Eer, F. Catthoor, N. Busa, P. Lippens, J. Huisken, *Code transformations for reduced data transfer and storage in low power realization of DAB synchro core* Proc.IEEE Wsh. on Power and Timing Modeling, Optimization and Simulation (PATMOS), Kos, Greece, pp.51-60, Oct. 1999.

[18] STBus specifi cations http://www.stmcu.com/inchtml-pages-STBus_intro.html

[19] ARM AMBA bus specifi cation http://www.arm.com/armwww.ns4/html/AMBA?OpenDocument