

Communication costs in a multi-tiered MPSoC

Marcel D. van de Burgwal

Department of EEMCS

University of Twente

Enschede, The Netherlands

Email: M.D.vandeBurgwal@utwente.nl

Gerard J.M. Smit

Department of EEMCS

University of Twente

Enschede, The Netherlands

Email: G.J.M.Smit@utwente.nl

Abstract—The amount of digital processing required for phased array beamformers is very large. It requires many parallel processors, which can be organized in a multi-tiered structure. Communication costs differ for each of the stages in such an architecture. For example, communication costs from the antenna front-end to the first processing stages is costly because of the amount of connections and data rate. Furthermore there is a trade-off between sequential processing exploiting locality of reference versus exploiting parallelism but adding communication costs. Thus, the optimal architecture depends on the importance that is given to the different measures.

A model is presented to determine the partitioning of a (beamforming) system based on communication costs. It is shown that different solutions can be explored based on the cost model and the incorporated quantitative and qualitative measures. Determining the importance of each measure is subjective to the situation and application. In this work a simple beamforming application is used optimised for energy efficiency.

Index Terms—Phased array, beamforming, multi-tiered, MP-SoC, communication costs, model

I. INTRODUCTION

Within the STW project “CMOS Beamforming Techniques” [1], [2], a multi-tiered multi-processor system-on-chip (MP-SoC) platform is chosen for the digital processing of beamforming. Multiple chips are combined on a board and multiple boards are combined into a system. For this architecture one must determine the granularity at each level; i.e. how many tiles, chips and boards are needed for the processing. A communication cost model is presented to answer these questions.

The basic operation required for phased array processing is described in section II and a multi-tiered architecture for digital processing is proposed in section III. Section IV introduces the cost model that is used for calculating the best match between architecture and application, using a communication analysis model. The proposed approach is compared with related work in section VI and the paper is finalized with a conclusion in section VII.

II. PHASED ARRAY BEAMFORMING

Phased array receivers use multiple antenna elements that pick up radio waves. In case the transmitter and receiver are located at a large distance, the radio waves behave as a wave front when arriving at the receiver. The antenna elements are positioned at a certain distance from each other, such that the wave front arrives at slightly different times at the

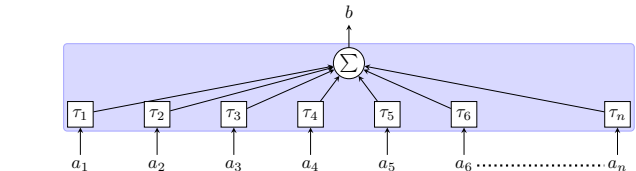


Fig. 1. Beamforming operation

various elements. By compensating for this time delay and combining the signals from all elements, the original signal can be restored and the direction of the transmitter is obtained. In this way, multiple transmitters can be traced at the same moment by applying different delays to a copy of the input signals. However, since the processing of each beam requires identical operations, in this paper we focus on the properties of an application consisting of a single beam.

Next to the combination of input signals, there are some additional operations that are required for a practical system. However, for this paper we will focus on the main operation, which is the beamforming operation. Beamforming can be done in several ways [3]. The time delay can be cancelled mechanically, by adding wires with different lengths between the antenna elements and the summation element, such that an additional delay is introduced for each of the elements. Optical beamformers [4] have been proposed that are based on optical ring resonators with which the antenna signals can be delayed in small steps. Furthermore, several analog beamforming solutions have been developed in the past [5]. Recently, digital beamformers are being considered, as the processing requirements for the time delay and summation are now feasible in CMOS technology [6], [7], [8]. One large advantage of digital beamformers is the support for processing multiple beams simultaneously. The samples received by the antenna array can be reused for each beam, so the front-end design can be done independent of the number of beams. Since beamforming is done fully electronically, the only limitation is the processing capacity of the architecture.

The delay-and-add operation mentioned above can be modeled as an expression tree as shown in figure 1. For this expression tree, the number of antenna elements can be chosen arbitrarily large. Hence, the number of inputs for the adder increases. Depending on the application, a phased array system may consist of hundreds to thousands of antenna elements [3].

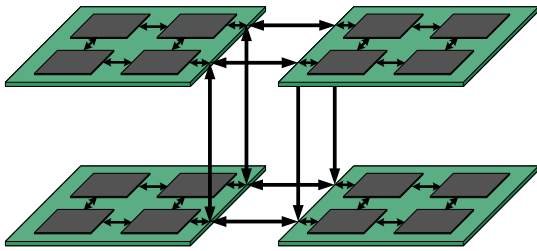


Fig. 2. Hierarchical architecture

In practice, the realization of a single adder with thousands of inputs is not feasible. Therefore, the operation needs to be divided over multiple processing elements that perform the addition. In this paper, we assume a multi-tiered topology on which the application is mapped.

III. MULTI-TIERED ARCHITECTURES

As mentioned before, the data rate of all antennas is too high to be processed by a single core, so a multi-core architecture is required. However, for a very large number of antennas even a multi-processor System-on-Chip is not sufficient, implying that parallelism is required on a higher level. This can be obtained by putting multiple chips on a board and combining multiple boards in the total system. An example of such a multi-tiered architecture is shown in figure 2.

When designing a multi-processor system, the question arises what topology should be chosen to connect all processing elements. There are many factors that motivate a specific choice of topology, like programmability, efficiency, scalability and so on. Moreover, the design space is bounded by physical limitations like maximum chip size, available I/O bandwidth, maximum number of I/O pins, wiring and transceiver costs, board size et cetera.

Adding 10% more processing elements to a system increases the raw processing power with 10%. However, there are no guarantees that the same utilization can be maintained, which may result in less than 10% additional net processing power. This happens, for example, in case the processing elements share a resource which can only be accessed by one processing element at a time. Arbitration is required to decide who can access the resource. Moreover, if the number of processing elements increases, arbitration may become more difficult and may take more time. Hence, the shared resource can be used less efficiently due to arbitration delays. This typically is the case with communication via a shared medium, i.e. a data bus.

The communication technology at several levels in the architecture differs. For communication between two on-chip processing elements, CMOS wires connected to basic repeaters can be used in parallel to form a high speed bus. However, for communication between two chips, the connection is totally different. Wires on a printed circuit board can be etched on the board to form high speed connections, but the wire routing has to be handled with care for high frequency signals, since different lengths and corners in the wires may cause a different

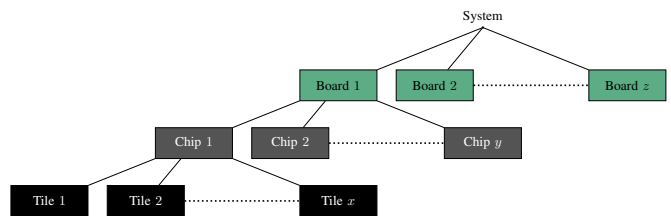


Fig. 3. System architecture in tree structure

frequency response of a transmitted signal. Driving such wires is done with transceivers for off-chip communication. Communication between multiple boards requires flexible wires, robust connectors and low noise transceivers. In practice, optical fibers are used often for communication between boards to ensure the boards are galvanically separated. Hence, for each of the levels, communication is different in terms of energy consumption, complexity and cost price.

IV. COST MODEL

For calculating the best mapping of a beamformer onto a tiled multi-tiered architecture, both the architecture and the application are partitioned to simplify the mapping. The general assumption for mapping is a homogeneous processing architecture (i.e., all processing elements have equal resources and performance). Additionally, the smallest block in the application graph resulting from the partitioning should be small enough to be processed by a single processing element.

A. Architecture description

A typical example of a processing architecture for beamforming is shown in figure 2. It shows a hierarchical tiled architecture that consists of many chips, each located on different boards which are interconnected. A more abstract representation of this example is shown in figure 3, where the architecture is presented in a tree shape. The nodes in the tree show the physical location, e.g. tile 1 is located on chip 1 which is positioned on board 1. Communication between sibling nodes is called *local* if both nodes have the same parent node. Otherwise, the communication is called *global*. Note that *local* communication can exist at multiple levels; thus, the classification *local* or *global* is not related to distance.

The tree structure does not show the topology of the interconnect: the assumption is that there is a connection between siblings, either with a direct link or via another sibling.

B. Application partitioning

As mentioned in section II, the beamforming application has to be partitioned over multiple processing elements. Since the delay-and-sum operation has a very regular structure, there are many candidate partitions. In order to generate each of them, the delay-and-sum operation is first decomposed to the smallest kernels possible, i.e. a set of delay operations and a single sum operation. Next, several kernels are clustered such that the total number of operations in a cluster exactly fits on a single processing element.

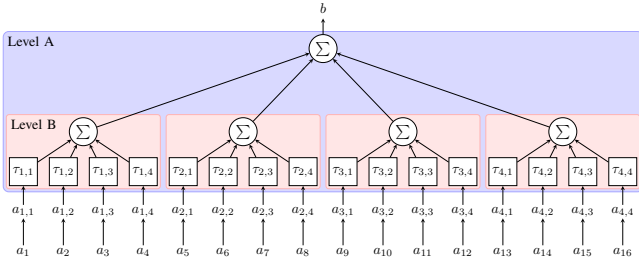


Fig. 4. Tree based beamformer with distributed adders

Given the summation in figure 1 is an associative operation on a set of input samples. Then, it can be rewritten to the sum of the summations of its subsets. In other words:

$$b(t) = \sum_{i=1}^N a_i(t - \tau_i) = \sum_{i=1}^{N/K} \sum_{j=1}^K a_{i,j}(t - \tau_{i,j}) \quad (1)$$

where N is the size of the array of input samples and K is the number elements per subarray. A graphical representation of the beamforming operation is given in figure 4.

In this form, the maximum number of inputs per addition operation is reduced. A reduction of the total amount of delay is also possible. Similarly to the addition operation, we use the associativity of the delay operation. Then, it is possible to split up a delay $\tau = \tau_x + \tau_y$, where $0 \leq \tau_x \leq \tau$ and $0 \leq \tau_y \leq \tau$.

For the i th delay-and-sum operation at level B in figure 4, the smallest delay applied is defined $\hat{\tau}_i = \min(\tau_{i,[1...K]})$. Hence, after delaying all inputs for a time $\hat{\tau}_i$ the delay-and-sum operation can be executed and the result is ready after $\hat{\tau}_i + \tau_{\delta\sigma}$, where $\tau_{\delta\sigma}$ denotes the time required for performing the delay-and-sum operation. Using delay extraction, the delay-and-sum operation is executed first and the result is delayed. Therefore, the result is available after $\tau_{\delta\sigma} + \hat{\tau}_i$, which equals the time of availability of the original value. However, the total amount of delay applied to all inputs and outputs of the delay-and-sum operation is decreased.

If delay extraction is used, the delays applied on the inputs are defined $\tau'_{i,j} = \tau_{i,j} - \hat{\tau}_i$. The total delay applied to the i th subarray of the distributed adder-based beamformer is $\sum_{j=1}^K \tau_{i,j}$, while after delay extraction this is reduced to the sum of the minimum delay $\hat{\tau}_i$ and the new, reduced delays $\sum_{j=1}^K \tau'_{i,j}$. Hence, for the entire array the total delay difference Δ is:

$$\begin{aligned} \Delta &= \sum_{i=1}^{N/K} \left(\hat{\tau}_i + \sum_{j=1}^K \tau'_{i,j} \right) - \sum_{i=1}^{N/K} \sum_{j=1}^K \tau_{i,j} \\ &= \sum_{i=1}^{N/K} \hat{\tau}_i + \sum_{i=1}^{N/K} \sum_{j=1}^K (\tau_{i,j} - \hat{\tau}_i) - \sum_{i=1}^{N/K} \sum_{j=1}^K \tau_{i,j} \\ &= \sum_{i=1}^{N/K} \hat{\tau}_i + \sum_{i=1}^{N/K} \sum_{j=1}^K \tau_{i,j} - \sum_{i=1}^{N/K} \sum_{j=1}^K \hat{\tau}_i - \sum_{i=1}^{N/K} \sum_{j=1}^K \tau_{i,j} \\ &= \sum_{i=1}^{N/K} \hat{\tau}_i - K \sum_{i=1}^{N/K} \hat{\tau}_i = (1 - K) \cdot \sum_{i=1}^{N/K} \hat{\tau}_i \quad (2) \end{aligned}$$

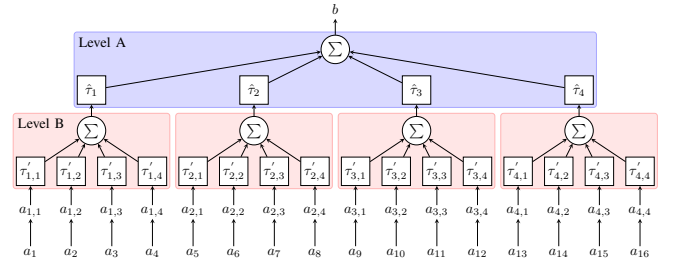


Fig. 5. Tree based beamformer after delay extraction

Thus, for $K > 1$, the difference is negative, which indicates a reduction in buffer area. Figure 5 shows the optimized beamformer using delay extraction. Similar to the optimization in level B, the minimum delay applied in level A can also be extracted and applied after the summation in level A. However, for a beamforming application one of the antenna streams is considered as a reference signal which is not delayed. Therefore, the minimum delay in that subarray in level B is equal to 0, which causes the minimum delay in level A also to be 0. Hence, no more delay can be extracted from level A.

Note that the application partitioning does not modify the total external bandwidth of a beamformer (input and output data rates are not changed), but it lowers internal bandwidth between adjacent levels. Therefore, the internal peak bandwidth is lowered, which considerably relaxes the hardware requirements. By choosing equal number of inputs for the subarrays at each level, the application can be described using homogeneous operations. Intuitively, this can be mapped very well to a homogeneous architecture.

C. Mapping

Depending on the available computational resources per tile, parts of the partitioned beamformer can be mapped. These resources may consist of the number of operations that can be performed in parallel, operational speed, available memory, et cetera.

Since we assume a homogeneous architecture, initially there is no difference between mapping the isolated operation o_1 to tile t_1 or mapping it to tile t_2 . However, there is a difference when operations o_1 and o_2 are to be mapped on tiles t_1 and t_2 , in case communication between these operations is required. Moreover, the mapping can also be constrained by I/O limitations. For example, if a chip has only enough pins for 10 antenna inputs, this may be considered as the mapping bottleneck.

D. Evaluation

The result of the mapping stage is evaluated using a cost function. Such a function may be the estimated total energy consumption, total delay buffer size, resource utilization, et cetera. Upon determination of the cost function, the difference between *local* and *global* communication becomes clear. The result of the beamformer partitioning is a tree structure with more nodes in it. Hence, the total amount of communication between nodes increases. However, the communication within

each partitioned level can be classified *local* and the combination of all processing within one level requires *global* communication. So, after partitioning the amount of *global* communication has decreased.

V. EXAMPLE

Consider the beamformer discussed previously, consisting of 16 antenna inputs ($N = 16$, $K = 4$). Assume the first element is used as reference, i.e. $\tau_1 = 0$, and the delay for each of the next elements linearly increasing with its index, i.e. a_i is delayed with $\tau_i = i - 1$. For the original beamformer, a total of $\sum_{i=1}^4 \sum_{j=1}^4 \tau_{i,j} = 0 + 1 + \dots + 15 = 120$ delays would be needed. Delay extraction results in $\hat{\tau}_1 = 0$, $\hat{\tau}_2 = 4$, $\hat{\tau}_3 = 8$ and $\hat{\tau}_4 = 12$ for the extracted delays and $\tau'_{i,j} = j - 1$ for the new input delays. In total, this equals $0 + 4 + 8 + 12 + 4 * (0 + 1 + 2 + 3) = 48$ delays (see also equation 2).

Assume the architecture consists of tiles with a 4-input adder and for each of the four inputs a delay buffer consisting of 8 positions. For the original beamformer, inputs $a_{[1..4]}$ and $a_{[5..8]}$ each can be mapped on one tile. Due to the delay buffer limitations, inputs $a_{[9..12]}$ and $a_{[13..16]}$ each need to be mapped on two tiles (one tile for applying 8 delay, the other tile for applying the remaining delay and the summation). Furthermore, another tile is required to add the intermediate sums of the four summing tiles. Hence, 7 tiles are required for the entire operation.

Due to the delay reduction as presented in equation 2, each subarray $a_{[1..4]}$, $a_{[5..8]}$, $a_{[9..12]}$ and $a_{[13..16]}$ exactly fits on one tile. The summation of these subarrays, however, cannot entirely be done by another core: the extracted delay $\hat{\tau}_4 = 12$ of subarray $a_{[13..16]}$ is too large to be mapped at once. There are two solutions for this problem: the first solution is to add another tile that is used to apply 8 delay, such that $\hat{\tau}_4$ can be reduced to 4. The other solution is to apply 4 delay at the inputs of subarray $a_{[13..16]}$, i.e. $\tau_{4,j} = j + 3$ and $\hat{\tau}_4 = 8$. In case of the first solution, a total of 6 tiles is required and for the second solution, 5 tiles are sufficient for applying the beamforming operation.

This example shows how a partitioned application can be mapped much more efficiently on hardware blocks. This increases utilization of each of the tiles, while the number of tiles decreases and, therefore, less silicon area is required. Using less tiles also implies having a lower energy consumption, because the tiles are identical.

VI. RELATED WORK

Kienhuis [9] presented a design approach for mapping applications to multi-processor architectures. This design method, called the “Y-chart approach”, describes an iterative design exploration for systems with many design parameters. In the hierarchical architecture considered in this paper, the number of design parameters is relatively low since a homogeneous architecture is assumed. Hence, a structural and thorough design space exploration is possible.

Bakshi and Gajski [10] proposed a design exploration for implementation of a beamformer. Therefore, they consider

pipelining and serialization to obtain an implementation with a very low latency. However, since this approach can only be used to implement a dedicated beamforming design, it cannot be used in our case where the system architecture consists of processing elements that can be employed dynamically.

Mohanty and Prasanna [11] used a hierarchical design space exploration methodology for mapping the entire processing chain, i.e. all operations from the radio front-end until the beamforming operation, on several architectures. The mapping algorithm is based on optimization heuristics.

In practice, partitioning of a beamformer architecture is often done using common knowledge for deciding how much tiles are to be placed on each level. Because the design space is not explored entirely, good alternatives may be overlooked.

VII. CONCLUSION AND FURTHER WORK

We presented a model for describing both a multi-tiered tiled architecture and a large scale beamforming application. Mapping that application onto a hierarchical tiled architecture requires partitioning and clustering, such that each basic operations can be executed on one tile in the system. The beamforming operation has a very regular structure and is, therefore, a good candidate for mapping onto a homogeneous system.

We evaluated an example beamformer and showed how partitioning can reduce the number of tiles required to perform the operation. Furthermore, the example shows that a slight mismatch between the resources available on a single tile and application resource requirements may result in less utilization. Hence, more tiles may be required to perform the operation, which results in a larger communication bandwidth and a higher energy consumption.

Further work will be done in the field of fault tolerance. One of the many advantages of phased array processing is the fact that errors occurring at one of the elements does not result in a system failure. Moreover, functionality could be relocated to other processing elements. In order to model such fault tolerance, the architecture model needs some more refinement.

ACKNOWLEDGMENT

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs.

REFERENCES

- [1] E. A. M. Klumperink, B. Nauta, A. B. J. Kokkeler, and G. J. M. Smit, “CMOS Beamforming Techniques,” University of Twente, STW Project Proposal, Feb. 2006.
- [2] M. D. van de Burgwal, K. C. Rovers, A. B. J. Kokkeler, G. J. M. Smit, K. S. Garakoui, M. C. M. Soer, E. A. M. Klumperink, and B. Nauta, “CMOS Beamforming Techniques project overview,” *Poster at: Scientific ICT Research Event Netherlands (SIREN)*, Oct. 2007. [Online]. Available: http://www.ictonderzoek.net/3/assets/File/posters/2007_23/2007_23.pdf
- [3] K. C. Rovers, M. D. van de Burgwal, A. B. J. Kokkeler, and G. J. M. Smit, “Rationale for and design of a generic tiled hierarchical phased array beamforming architecture,” in *Workshop on Circuits Systems and Signal Processing (ProRISC)*. Utrecht: Technology Foundation, Nov. 2007, pp. 160–168.

- [4] L. Zhuang, C. G. H. Roeloffzen, R. G. Heideman, A. Borreman, A. Meijerink, and W. van Etten, "Single-chip ring resonator-based 1×8 optical beam forming network in CMOS-compatible waveguide technology," *IEEE Photon. Technol. Lett.*, vol. 19, no. 15, pp. 1130–1132, Aug. 2007.
- [5] M. I. Skolnik, *Introduction to Radar Systems*, 3rd ed. New York, NY, USA: McGraw-Hill, 2001.
- [6] H. L. van Trees, *Optimum array processing*. New York: Wiley-Interscience, 2002, vol. Detection, estimation and modulation theory.
- [7] L. C. Godara, *Smart antennas*. Boca Raton, Fla, USA: CRC Press, Jan. 2004.
- [8] H. J. Visser, *Array and phased array antenna basics*. Chichester: Wiley, Sep. 2005.
- [9] A. C. J. Kienhuis, "Design space exploration of stream-based dataflow architectures: Methods and tools," Ph.D. dissertation, Delft University of Technology, The Netherlands, Jan. 1999. [Online]. Available: <http://repository.tudelft.nl/file/130931/111701>
- [10] S. Bakshi and D. D. Gajski, "Design space exploration for the beam-former system," University of California, Irvine, Tech. Rep. 93-34, Aug. 1993.
- [11] S. Mohanty and V. K. Prasanna, "A hierarchical approach for energy efficient application design using heterogeneous embedded systems," in *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*. New York, NY, USA: ACM, 2003, pp. 243–254.