

System level design flow for low latency sensors

R Woods, J McAllister, R Hasson and S. Fischaber
Programmable Systems and Networking Laboratory, Queen's University of Belfast,
Belfast, BT9 5AH, N. Ireland.

Abstract

There is a considerable interest in using high level representation such as dataflow in the design of complex heterogeneous DSP systems. The key challenge is to implement hardware functionality efficiently by performing optimisations at this dataflow level. The paper outlines the dataflow-based Abhainn design flow and describes some work on the use of the flow in the design of a fixed beamformer.

Keywords: Embedded systems, programmable hardware, DSP implementation

Introduction

DSP systems have often been implemented on multiprocessor platforms comprising RISC and DSP multiprocessors. Increasingly, designers want to add programmable logic capability in the form of field programmable gate arrays (FPGAs) to provide hardware acceleration for computationally intensive or speed critical applications. However, this requires the development of an integrated design environment that allows hardware functionality to be matched to software requirements and effective inter-processor communications to be considered. However, rapid implementation techniques for these heterogeneous systems have been slow to emerge and current design techniques lack the ability to optimize the entire system from a very high level.

Architectural exploration and optimization tools for rapid implementation such as Catapult C [1], Ptolemy [2], PeaCE [3], Compaan [4], and GEDAE [5], are beginning to emerge. Whilst Catapult C captures many of the aspects of "behavioural synthesis", there is no design flow to target heterogeneous platforms and the user is left to develop the

communications interface between separate partitions. GEDAE uses dataflow (DF) to target multiprocessor systems, whilst approaches such as Compaan use predefined IP cores that are connected together using process network models. The Compaan/Laura design flow takes Matlab description (restricted to parameterized static affine nested loop programs) and synthesises it, via a Kahn process network, to hardware and software [4]. The tools give hints at compile time on how to rewrite the algorithm to improve performance [6]. PeaCE [7] is an extension of the Ptolemy tool and uses a dataflow representation to create a globally controlled synchronous architecture and provide memory efficient code. With these two tools, optimizations only work at the network level and are restricted by the use of static intellectual property (IP) cores.

The key issue is that these IP cores are often created using signal flow graphs (SFG) methods and are optimized by various techniques e.g. pipelining, retiming and folding. This restricts the optimizations that can be done at the dataflow graph (DFG) level (as these optimizations were carried out at the SFG level), or requires re-synthesis of the core from a SFG for each

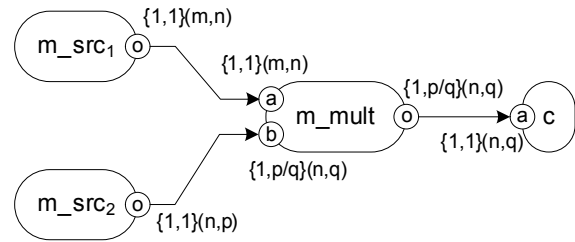
change in the DFG [3] which is counter-productive. This paper outlines a method that allows the core structure to remain adaptable for a range of further optimizations. This will become the methodology for implementing a number of complex DSP systems including hardware for low latency sensors.

Dataflow Basics and Application to Rapid Implementation

The use of dataflow graphs (DFGs) for representation and rapid implementation of DSP systems is a popular and well researched area. A DFG describes a set of functional actors connected via arcs. Tokens are consumed from the connecting arcs at the input ports of an actor when an actor fires. These are transformed and produced on the output ports of the actor. These tokens stream between actors along arcs and can either be scalar, vector, or matrix values. Each port on the actor has a threshold (T), which defines the number of tokens that will be produced or consumed during one firing of the actor. The number of times the actor fires during one iteration of the schedule is defined at the granularity (G). A special type of DF is synchronous dataflow (SDF) in which the schedule can be determined at compile time [7].

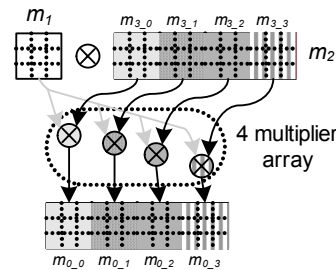
Multidimensional synchronous dataflow (MSDF) has been proposed [8] to use intra-token parallelism to uncover further parallel structures, but this has limited use for rapid implementation and architectural exploration on FPGAs. It is shown [3] that when doing architectural synthesis with pipelined cores the benefits of using MSDF are diminished. A new modelling domain called multidimensional arrayed synchronous dataflow (MASDF) has been proposed [3]. MASDF uses the Processing Graph Method [4] to represent a set of actors and arcs as a family. A MSDF graph is shown for a matrix multiplication example where matrices m_1 (dimensions

(m,n)) and m_2 (dimensions (n,p)) are multiplied together where the value in curly braces represents the token size.



MSDF 2D Matrix Multiplication Graph

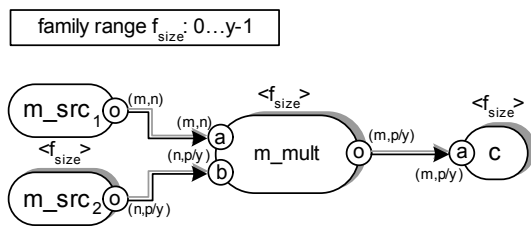
To exploit parallelism, m_2 can be divided into y matrices ($m_{3_0}, m_{3_1}, \dots, m_{3_{y-1}}$) which are then individually multiplied by m_1 as illustrated below for $(m,n) = (3, 3, 12)$ and $y = 4$. The relationship between the number of multipliers and the size of the m_3 matrix can therefore be exploited to trade off the number of actors with the token size, which translates to trading off resource count for throughput on an FPGA.



Parallelisation ($y=4$) for matrix operation

A MASDF graph is given where the families are indicated by the presence of shadows. The family size is shown in the triangular braces above the actor. In this case the designer can control the size of the m_src2 , m_mult , and c families by changing the value of y . It also controls the matrix dimensions on ports $m_src2.o$, $m_mult.b$, $m_mult.o$, and $c.a$. The MASDF domain and associated synthesis method allow the designer to make high level decisions about the hardware being implemented by modifying the family sizes, and hence the number of cores in the implementation as

well as the size of the tokens processed by each actor.



MASDF Matrix Multiplication Graph

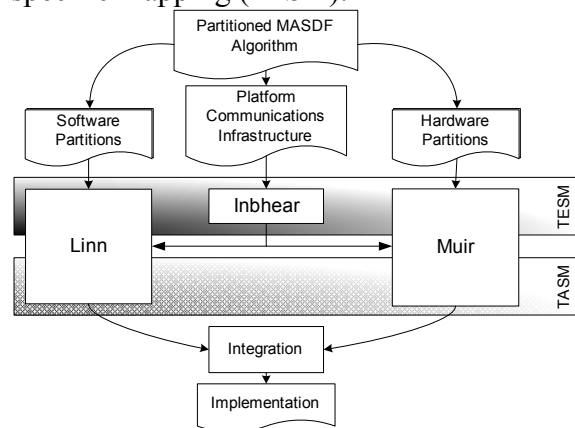
In addition to families of actors and arcs, MASDF also allows families of ports which can be thought higher order token which fixes T at each port to 1 (the family is processed as one large token) or multiple streams of the same token type where T for each port, can be arbitrarily scaled. In the case of a family of four ports representing separate channels, with $T_0=4$ and $T_{1-3}=1$, the actor would take 4 tokens from channel zero, followed by one from each other channel before restarting.

Using MASDF then, each actor can be described by four features: **T**, the threshold at each port; **G**, the granularity of each actor; **X**, is the token dimension at each port and; **S**, the number of independent streams on each port of the actor. An appropriate hardware synthesis method has been developed along with the MASDF domain to allow for rapid implementation and architectural exploration.

Abhainn Flow

The concepts of using MASDF have been captured in the *Abhainn* flow which uses a combination of commercial design tools, but provides an efficient implementation onto programmable hardware. This flow is being developed as part of a major research activity within Queen's University and will be used when developing solutions for the various complex DSP systems encountered in this project.

The MASDF algorithm representation is firstly partitioned between hardware and software. Three processes are involved in the *Abhainn* flow: software synthesis for the software partition (*Linn*); hardware synthesis for the hardware partition (*Muir*); and the inter-processor communications framework (*Inbhear*). They are divided between two distinct stages, technology specific mapping (*TESM*) and target specific mapping (*TASM*).



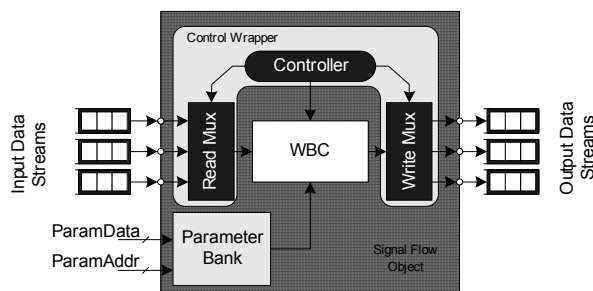
Abhainn Flow Architecture

TESM inserts the specific processor and communications cores into place as well as creating the communications network to allow data transfer between the hardware and software partitions. The TASM takes the implementation architecture from the TASM and translates it to the specific target devices begin used. As software synthesis techniques from dataflow graphs is a well established area [9], the focus is on *Muir* and *Inbhear*.

One major limitation of current approaches is the usage of existing IP cores, as many of these cores have a fixed interface and a fixed internal architecture (albeit parameterisable). This limits the efficient usage in a high level system (dataflow) flow. By making the core more programmable here termed *white box* IP cores and adding suitable wrappers, system efficiency and performance can be improved.

In *Abhainn*, hardware cores, termed signal flow objects (SFOs) are divided into three pieces: the control wrapper, the white box component (WBC), and the parameter bank. The control wrapper implements the cyclic schedule to multiplex data into and out of the WBC and the WBC executes the actor functionality. The parameter banks holds run-time parameters e.g. filter weights.

The control wrapper is generated to insure proper data switching in and out of the WBC. The parameter bank is generated in RAM with a particular address map that is passed to the host controller. This comes directly from the dataflow graph description and represents the design process for making the core usable directly and efficiently in a system flow.



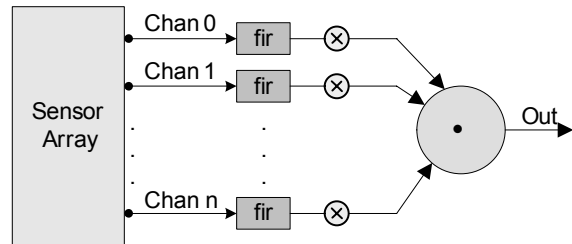
SFO Structure

The flow also addresses the inter-process communications network addressed communications between the hardware blocks and the software. The process is outlined in detail [10] and the next section describes the design of a fixed beamformer and shows how these concepts are applied.

Fixed beamformer system

Beamforming provides an effective and versatile method of spatial filtering for radar, sonar, biomedical and comms. applications. A beamformer is typically used with an array of sensors which are positioned at different locations so that they are able to “listen” for a received signal by taking spatial samples of the received propagating wave fields. The fixed

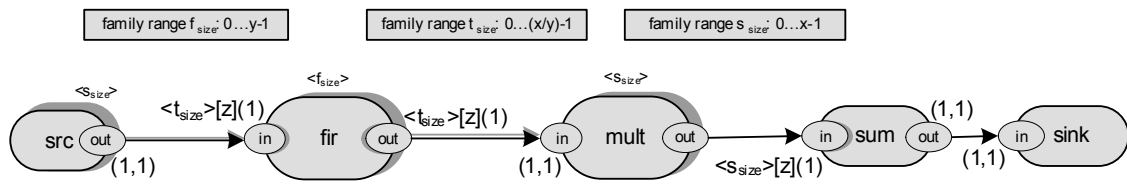
beamformer can determine from what direction a particular signal has come from in a noisy environment by attenuating the noise and interference coming from other directions. It involves FIR filtering and summing the outputs of an array of sensor elements (see below). For a particular target, the average power at the output of the beamformer is maximised when the beamformer is steered toward the target.



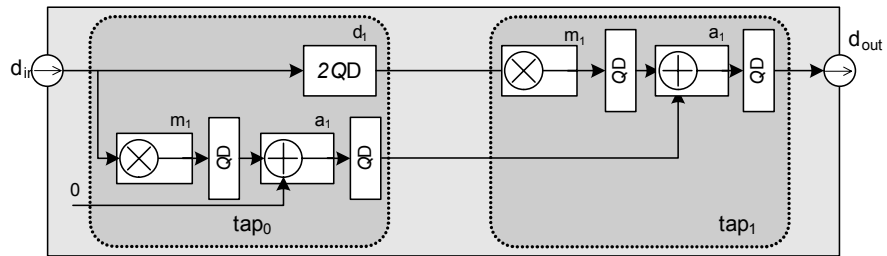
Fixed beamformer block diagram

In the MASDF graph for the beamformer, the multidimensional tokens are thought of as multiple independent streams of scalars namely channels. The MASDF graph allows choices such as the number of channels, the sequence of the channels (*S* exploration), and the number of filters (combined with the number of channels to define *X*). These choices allow design decisions and exploration to be made at a very high level, e.g. trading off throughput for a reduction in area by hardware sharing the filters or processing a single channel for multiple cycles instead of interleaving each channel (block processing).

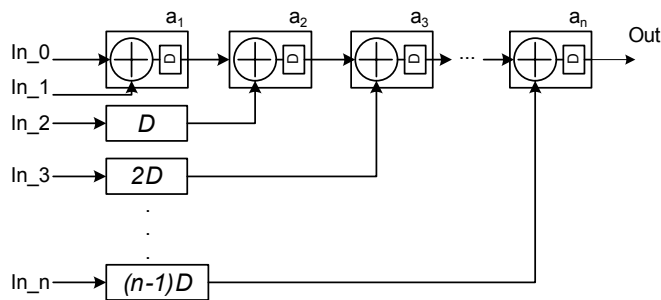
The WBC for a two tap FIR filter is shown. As multiple channels can be implemented on a filter, delays are needed to store the intermediate values for each channel and are scaled by the number of channels (*Q*) implemented in each filter. The sum block in the MASDF sums the channels. An adder chain was used here for simplicity, although an adder tree would be equally viable. Once again, the delays scale to the number of input ports to ensure that the correct samples are summed together.



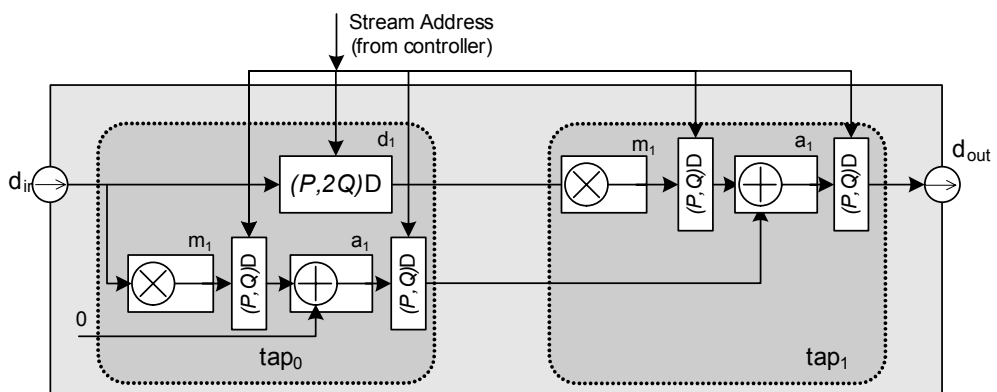
Fixed beamformer MASDF Graph



WBC for FIR filter of fixed beamformer



WBC for Summation of fixed beamformer



Pipelined Scalable FIR Filter

For high order S operations, multiple samples from a channel are taken in a row (block processing) [10] which means that the delays in the filters need to be scaled vertically by a factor of P . This now gives arrays of delays with dimensions (P,Q) and provides a filter that can be used for a wide range of applications by changing the number of channels and the sequence that the channels are processed. When implementing the filters on FPGAs, these arrays can be simply mapped into distributed RAM components with the address being determined by the channel state of the controller.

Currently a complex multi-channel beamformer design is being implemented on a Xilinx Virtex-II Pro XC2VP125. Various design configurations are being explored by changing the parameters for the design. Some of the results will be highlighted during the presentation.

Conclusions

The paper has described *Abhainn*, a high level design flow based on dataflow representation which allows core optimization to be carried out from a dataflow level. In this instance, we have made the core more flexible by making the internal delays more programmable. The paper shows how the concept is applied to the design of a fixed beamformer.

References

- 1 Mentor Graphics, "Catapult C Datasheet", www.mentor.com.
- 2 Hylands, C, et al., 2003, Tech. Mem. UCB/ERL M03/25, University of California at Berkeley.
- 3 Oh, H, and Ha, S, 2004, J. of VLSI Sig. Proc. Syst. **37**, no. 1, 41-51.
- 4 Stefanov, T et al., 2004, *Proc. DATE*, 340-345.
- 5 B.K. Madahar, et al, 2003, EURASIP J. on Appl. Sig. Proc. **6**, 580-593.
- 6 McAllister J. et al., 2004, Proc. IEEE W/shop on Sig. Proc. Sys, 164-169.
- 7 Lee, EA and Messerschmitt DG, 1987, Proc. IEEE **75**, no. 9, 1235-1245.
8. Murphy, PK, and Lee EA, 2002, IEEE Trans. Sig. Proc. **50**, no. 8, 2064-2079.
9. Bhattacharyya, SS, Murthy, PK, Lee, EA, 1996, *Software Synthesis from Dataflow Graphs*, Kluwer Acad. Publishers.
10. McAllister, J, 2004, PhD Diss., Queens University Belfast.

Acknowledgements

The work reported in this paper was funded by the Electro-Magnetic Remote Sensing (EMRS) Defence Technology Centre, established by the UK Ministry of Defence and the Engineering and Physical Science Research Council through a CASE award. The DTC is run by a consortium of SELEX Sensors and Airborne Systems, Thales Defence, Roke Manor Research and Filtronic.