Electrical and Computer Engineering
@ The University of Puerto Rico Mayagüez Campus

TITLE OF RESEARCH WORK: High-Level Partitioning of Discrete Signal Transforms for Distributed Hardware Architectures

Research Description:

Discrete signal transforms (DSTs), such as the discrete Fourier transform (DFT), have numerous applications in a wide spectrum of scientific fields. To attain superior performance, the size and composition of these algorithms frequently requires implementation to architectures involving more than one hardware device (e.g. multiple FPGA platforms). Partitioning is a key component in the design process for those implementations and several mechanisms have been proposed, specifically targeting distributed dedicated-hardware architectures. However, because of the abstraction level at which these methods operate and because they are tailored for general algorithms, they are unable to exploit functional properties of the algorithms as part of the partitioning process. In our research, we study the integration of properties such as symmetry, index mappings, and decomposition rules into automated partitioning strategies for DSTs onto distributed hardware architectures. A high level partitioning methodology is being developed which takes advantage of such properties to improve partition solution quality and exploration time.

WALSAIP GROUP ASSOCIATION: Automated Information Processing Group	
THESIS TITLE: High-Level Partitioning of Discrete Signal Transforms for Distributed Hardware Architectures	
THESIS ADVISOR: Prof. Manuel Jimenez	
INSTITUTION: Electrical and Computer Engineering Dept. University of Puerto Rico at Mayaguez	
PERSONAL WEBSITE: http://www.ece.uprm.edu/~s370549	**NAME OF RESEACH ASSISTANT:** Rafael Arce Nazario

RESEARCH PROJECT OUTCOMES:

Publications:

[ArceFPL06] R.. Arce Nazario, M. Jiménez, D. Rodriguez. “High-level Partitioning of Discrete Signal Transforms for Multi-FPGA Architectures”. 16th International Conference on Field Programmable Logic and Applications. August 2006. Madrid, Spain.

[ArceMWSCAS06] R.. Arce Nazario, M. Jiménez, D. Rodriguez. “Functionally-aware Partitioning of Discrete Signal Transforms for Distributed Hardware Architectures”. 49th Midwest Symposium on Circuits and Systems. August 2006. San Juan, Puerto Rico.

[ArceFCCM06] R.. Arce Nazario, M. Jiménez, D. Rodriguez. “Effects of High-Level Discrete Signal Transform Formulations on Partitioning for Distributed Hardware Architectures”. Extended abstract for poster at the IEEE Symposium on Field-Programmable Custom Computing Machines. April 2006. Napa, California.

[ArceMWSCAS05] R. Arce Nazario, M. Jiménez, D. Rodriguez. “An Assessment Of High-Level Partitioning Techniques For Implementing Discrete Signal Transforms On Distributed Hardware Architectures”. Midwest Symposium on Circuits and Systems. August 2005. Cincinnati, Ohio.

Tools and Applications:

One of the main objectives of our research is the development of a high-level partitioning methodology for Discrete Signal Transforms onto Distribute Hardware Architectures. Fig. 1 shows a conceptual map of our partitioning methodology. The inputs on the top are a DST specified as a Kronecker Products Algebra formulation, parameterized at least by the resolution of its points, and a high-level specification of the target architecture, which includes the number and logic capacity of the devices and their connection topology.

Based on the inputs, a series of heuristics reformulate the transform to expose characteristics that are exploited by the partition/placement process. Reformulation is accomplished by applying algorithmic-level transformations, such as factorization and permutation rules, on the original formulation. The *Kronecker to Graph (KTG)* process converts the algorithmic formulation into a dataflow graph (DFG) whose nodes denote functional primitives, i.e. small computational components that are common throughout the formulation and have been identified as efficient procedures on the target devices. DFG nodes are as fine-grained as deemed necessary by the initial topology/transform heuristic analysis, in an effort to reduce design exploration time. Then, a *partitioning/placement (P/P)* algorithm is run on the DFG, assisted by high-level area and communication estimators. P/P is handled by a communication-channel-aware, *k*-way graph partitioning algorithm. The P/P solution and its cost are used by the *Control Heuristics* to guide the *Formulation Manipulator* to reformulate the DST onto further expressions. Formulation exploration ceases when successive reformulations have produced no considerable gain in partition quality.

Methodology processes were developed using a mixture of in-house, academic and commercial tools:

1. The KTG tool, developed in-house, converts a Kronecker Product Algebra expression to a dataflow graph. It is essentially an expression parser coupled with a graph construction algorithm that obeys the operands and operations provided in the expression.

2. The partition/placement algorithm is an extension of the Kernighan-Lin bipartitioning heuristic expanded to handle *k*-way partitioning to architectures with heterogeneous channel topologies. Several considerations have been implemented in the P/P process to make it more effective in dealing with graph structures such as commonly found in DSTs [ArceMWSCAS06].

3. We used FPGA synthesis tools, such as Synplicity's Synplify and Xilinx ISE, to obtain synthesis results which allowed the development of hardware resource estimation models.

4. The rules for the heuristic control have been developed through a series of experiments in which we have observed the effect of algorithm-level transformations on partition solution quality [ArceMWSCAS06]. We used the KTG and P/P tools, as well as automatically generated formulations, to conduct the experiments.

5. The SPIRAL code generation system (developed by the SPIRAL research group in Carnegie Mellon University) is serving as basis for the Formulation Manipulation capabilities of our approach.

AIP LAB
AUTOMATED INFORMATION PROCESSING LABORATORY
NATIONAL SCIENCE FOUNDATION
ICIS
Institute for Computing and Informatics Studies
WALSAIP

RELATION OF RESEARCH WORK TO WALSAIP PROJECT:

One of WALSAIP's main interests is the processing of signals gathered by physical sensors. Discrete Signal Transforms are an essential part of most signal processing applications. Our research is of importance to the WALSAIP project because it envisions methodologies and tools for the implementation of these transforms to distributed dedicated hardware devices. This, in turn, is significant for the following reasons:

1. DSTs have achieved maximum performance when implemented to dedicated hardware devices, such as ASICs and FPGAs. However, mainly because of their complex design process, these implementations are not commonplace. Automated design tools, such as those proposed in our research, help reduce the effort required to achieve such high-performance implementations.

2. Design trends in microprocessors and embedded systems are moving toward multi-core paradigms. Implementation of algorithms to these distributed systems-on-chip can be aided by methodologies and tools such as ours.

IMAGE REPRESENTATIVE OF RESEARCH WORK:

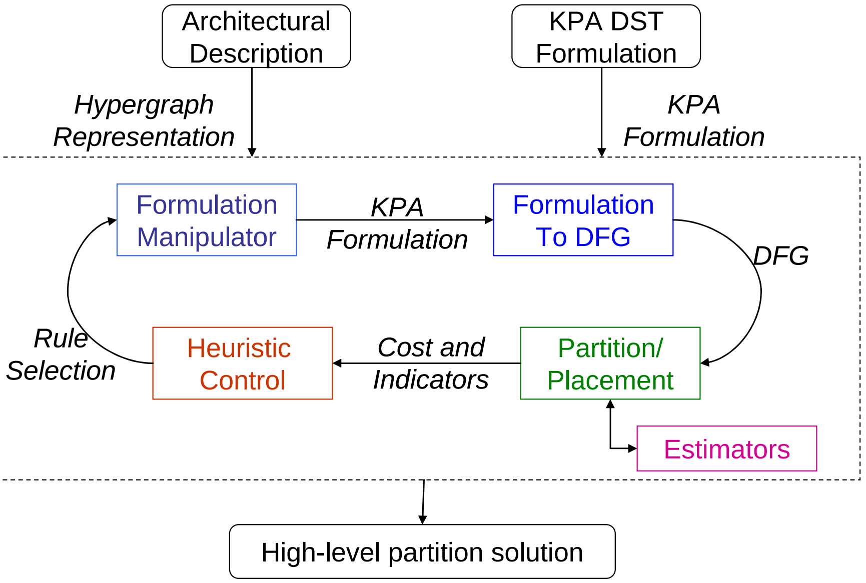

Figure 1 - Block diagram of our proposed high-level partitioning methodology. See full description of this figure in Section "Tools and Applications".

RESEARCH DEMONSTRATION:

As a sample prototype of our research work, we provide a downloadable demo of the Kronecker to Graph (KTG) tool at http://ece.uprm.edu/~s370549/ktg.htm. The tool performs an essential task within our methodology: conversion of a Kronecker Product Algebra expression to a dataflow graph (DFG). KTG's main objective is illustrated in Figure 2. Each DFG node represents a computational entity from the formulation. Documentation and examples for using this tool are provided at: http://ece.uprm.edu/~s370549/KTG_doc.htm.

Although the task of KPA to DFG conversion is not the main focus of our research, we consider the KTG tool as a significant contribution, as it is, to our knowledge, the first publicly available tool to perform such tasks.

$$\left(DFT_2 \otimes I_4\right)\left(I_2 \otimes \left(DFT_2 \otimes I_2\right)\right)\left(I_4 \otimes DFT_2\right)$$

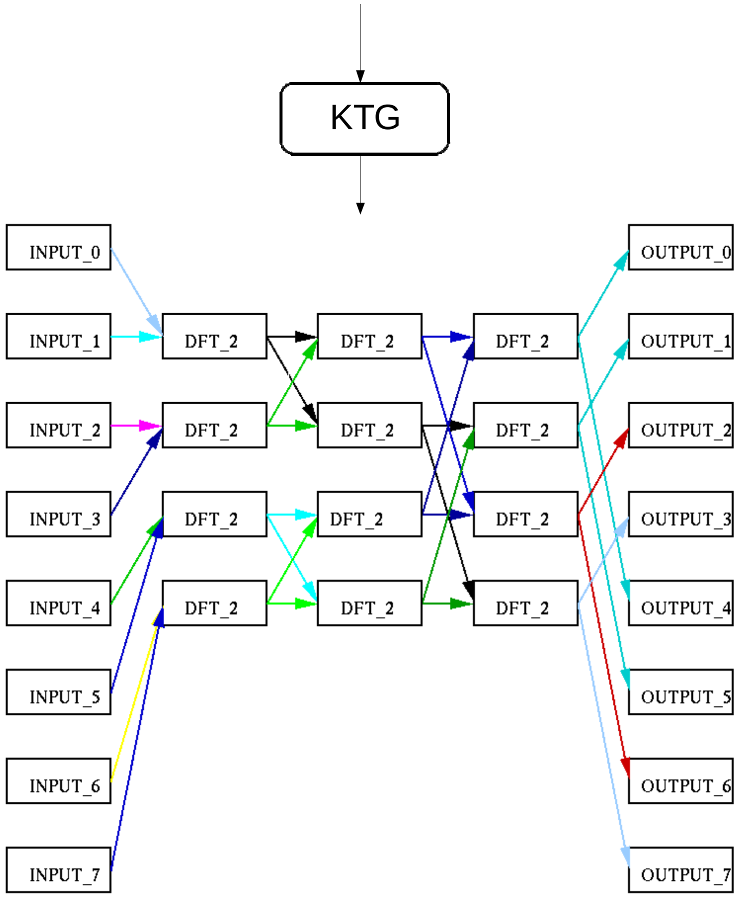

Figure 2 - KPA expression containing typical operands and operations: DFT_2 is a 2-point discrete Fourier Transform, I_n represents an identity matrix size n, ⊗ stands for Kronecker product operation.