# High-level Partitioning of Discrete Signal Transforms for Distributed Hardware Architectures

WALSAIP

www.walsaip.uprm.edu

**Rafael Arce-Nazario – Ph.D. Student**
**AIP Group, CISE Program,**
**Email: *rafael.arce@ece.uprm.edu***

**Prof. Manuel Jimenez – Advisor**
**University of Puerto Rico,**
**Mayagüez Campus**

## 1 Abstract

High-level partitioning (HLP) is an essential step in the implementation of algorithms to distributed hardware architectures (DHA), such as multi-FPGA platforms. Fast discrete signal transform algorithms (e.g. FFT, FCT) have a number of decomposition and reformulation properties that offer opportunities for developing semantically guided high-level partitioning schemes that influence their mapping to hardware. Based on this hypothesis, we are conducting research to devise a functionally-aware methodology that uses these types of transformations to provide improved results for the high-level partitioning of discrete signal transforms to DHAs.

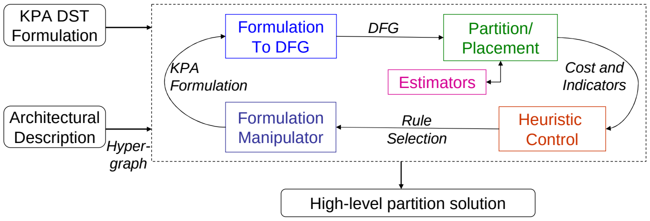
## 2 Motivation and Objectives

- Discrete Signal Transforms (DSTs): major component in today's applications
- DST partition is of interest because:
  - high logic resource utilization  Multi-FPGA / Reconfig. Comp.
  - trend toward multi-core/SoC
- Previous automated schemes treat DSTs in generic way.

- DSTs have properties that can be used to aid High Level partitioning
  - fast algorithms: regularity, factorization rules, recursiveness
  - compact, systematic manipulation at the algorithmic level using Kronecker Products Algebra (KPA).

Hypothesis: Improve DST to multi-FPGA partitioning process by considering DST properties and reformulation.

## 3 Partitioning Methodology



The core of our methodology is an optimization loop that performs exploration in the space of equivalent formulations and partitions using DST-specific transformations.

1. Kronecker to graph (KTG) conversion generates DFG corresponding to (KPA) formulation. Each DFG node is a primitive from the formulation.
2. A Partitioning/placement (P/P) algorithm is run on the DFG, which consults Area/Communication estimators to determine current solution's quality.
3. Indicators output by Partitioning/placement (P/P) are used by a Heuristic Control to chose rule for reformulation, which is performed by the Formulation Manipulator.
4. Process is iterated until no further significant gain is being achieved.

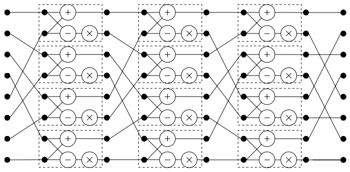
## 4 Tools

**Kronecker to DFG Conversion:**

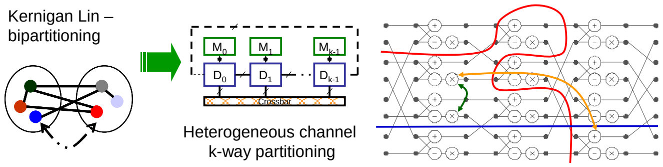
$$F_8 = R_8\left(T_2\left(I_4 \otimes F_2\right)P_{8,4}\right)\left(T_1\left(I_4 \otimes F_2\right)P_{8,4}^{8}\right)\left(T_0\left(I_4 \otimes F_2\right)P_{8,4}\right)$$

Operator matrices: Identity, Transform, Permutation, Unitary, Unitary Transpose, Twiddle

KPA operations: Tensor product (⊗), Direct Sum (⊕ ), Matrix Multiplication



**Partition/Placement:**



Kernigan Lin – bipartitioning

Heterogeneous channel k-way partitioning

Some graph-level considerations
- Linear horizontal rather than random initial partitions
- Avoid schedule-unwise swaps, prefer stage-centered.

## 5 Formulation Exploration

**Objective:** Explore space of equivalent formulations in search for one that better suits the target architecture.

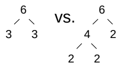
**Challenge:** Combinatorial explosion of the exploration space.

**Approach:** Conducted experiments to assess the impact of transformations on partition quality. Results were used to devise exploration strategy.

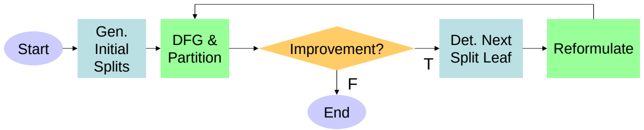
**Experiments:**

- Effect of inter-stage permutations (Pease vs. Cooley-Tukey vs. Stockham ..)
- Effect of operand granularity

  $(F_4 \otimes I_4)T_{16,4}(I_4 \otimes F_4)$ vs. $\left(\left(\left(\left((F_2 \otimes I_2)T_{4,2}(I_2 \otimes F_2)\right)\otimes I_2\right)T_{8,2}(I_4 \otimes F_2)\right)\otimes I_2\right)T_{16,2}(I_8 \otimes F_2)$
- Effect of breakdown strategy
  - Exhaustive generation of split trees for 'small' DFTs.
  - Observation of split tree decisions that lead to 'partition friendly' formulations



**Algorithm:** Greedy heuristic for 'top-down' generation of breakdown strategy.



**Results:**

Results of FFT formulation exploration for various FFT sizes targeting 4 device, ring + crossbar topology.

| | Latency (c-steps) | | | Exploration Time | | |
|---|---|---|---|---|---|---|
| Size | Form. Exp. | SA | Improvement | Form. Exp. | SA | Improvement |
| 512 | 122 | 136 | 10.3% | 50s | 10m 16s | 91.9% |
| 1024 | 235 | 253 | 7.1% | 5m 35s | 3h 16m | 97.2% |
| 2048 | 458 | 528 | 13.3% | 1h 34m | 1d 22h 2m | 96.6% |
| 4096 | 913 | 1047 | 12.8% | 1h 40m | 11d 7h 37m | 99.4% |

## 7 Ongoing Work and Conclusions

The introduction of DST considerations into the graph partitioning heuristics, as well as the algorithmic-level exploration of DSTs, help our methodology obtain improved partitioning results in considerably less time than general purpose partitioning methods.

Currently, the proposed partitioning methodology is being extended to work with discrete cosine transforms (DCT). The extension required development of a Cooley Tukey-like factorization scheme for DCTs, and generated improved results over other existing regular DCT formulations.

## 8 References

R. Arce-Nazario, M. Jimenez, and 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.

R. Arce-Nazario, M. Jimenez, and D. Rodriguez. "High-level Partitioning of Discrete Signal Transforms for Multi-FPGA Architectures". IEEE 16th International Conference on Field Programmable Logic and Applications. August 2006. Madrid, Spain.

Supported By

## Slide 1

**RAN1** Fromulation exploration

-> can lead to exponential explosion
-> mention experiments: perm, gran, split trees
-> observation on split trees, etc..
-> algorithm
-> results
-> references: MWSCAS06, FPL

-> ongoing work..

Rafa, 1/23/2007