A Combined Hardware/Software Optimization Framework for Signal Representation and Recognition

Anna Gilbert, Yi Wang, University of Michigan
Mary Hall and Melina Demertzi, USC/ISI
Pedro Diniz, Technical University of Lisbon

This research has been partially funded by NSF DDDAS Program.

Motivation

Signal and image recognition

- Largely the domain of experts who eke out the necessary performance under constrained execution environments
- Optimization opportunities:
  - Mathematical: select best transform for feature selection
  - Domain-specific: exploit transform properties to optimize solution
  - General: manage locality and parallelism

This Project

- A systematic approach to constructing image/signal recognition systems
- Exploit signal properties to jointly optimize a computation, from mathematical representation through implementation in hardware

Data driven

- Signal properties guide the optimization strategy

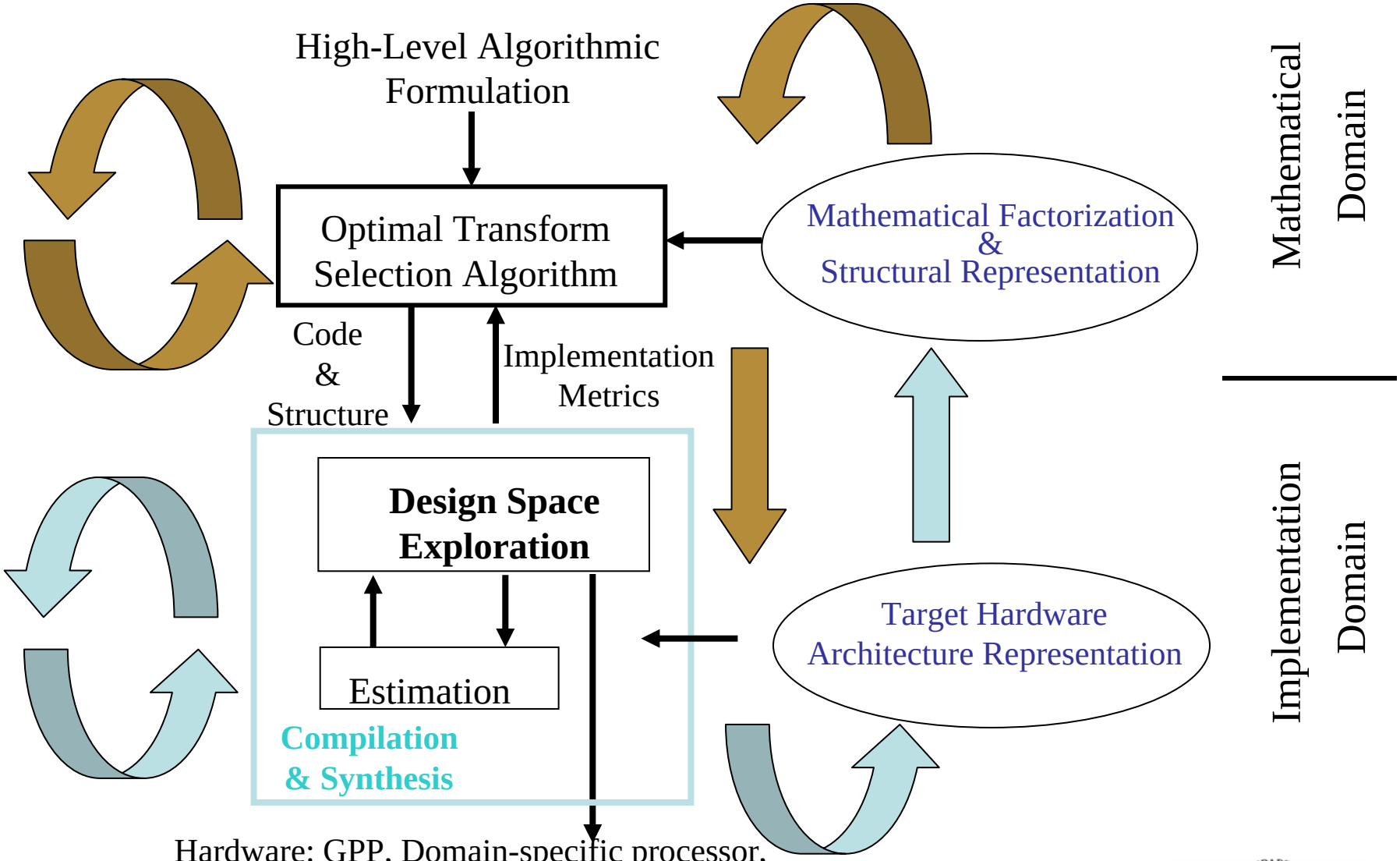
Dynamic

- Algorithm execution will (ultimately) be tailored to the input signal

Outline

- Conceptual diagram
- Example: digit recognition
- Algorithm overview
- Mathematical building blocks
- Optimization strategies
- Preliminary results

Conceptual Diagram



Signal Processing and DDDAS

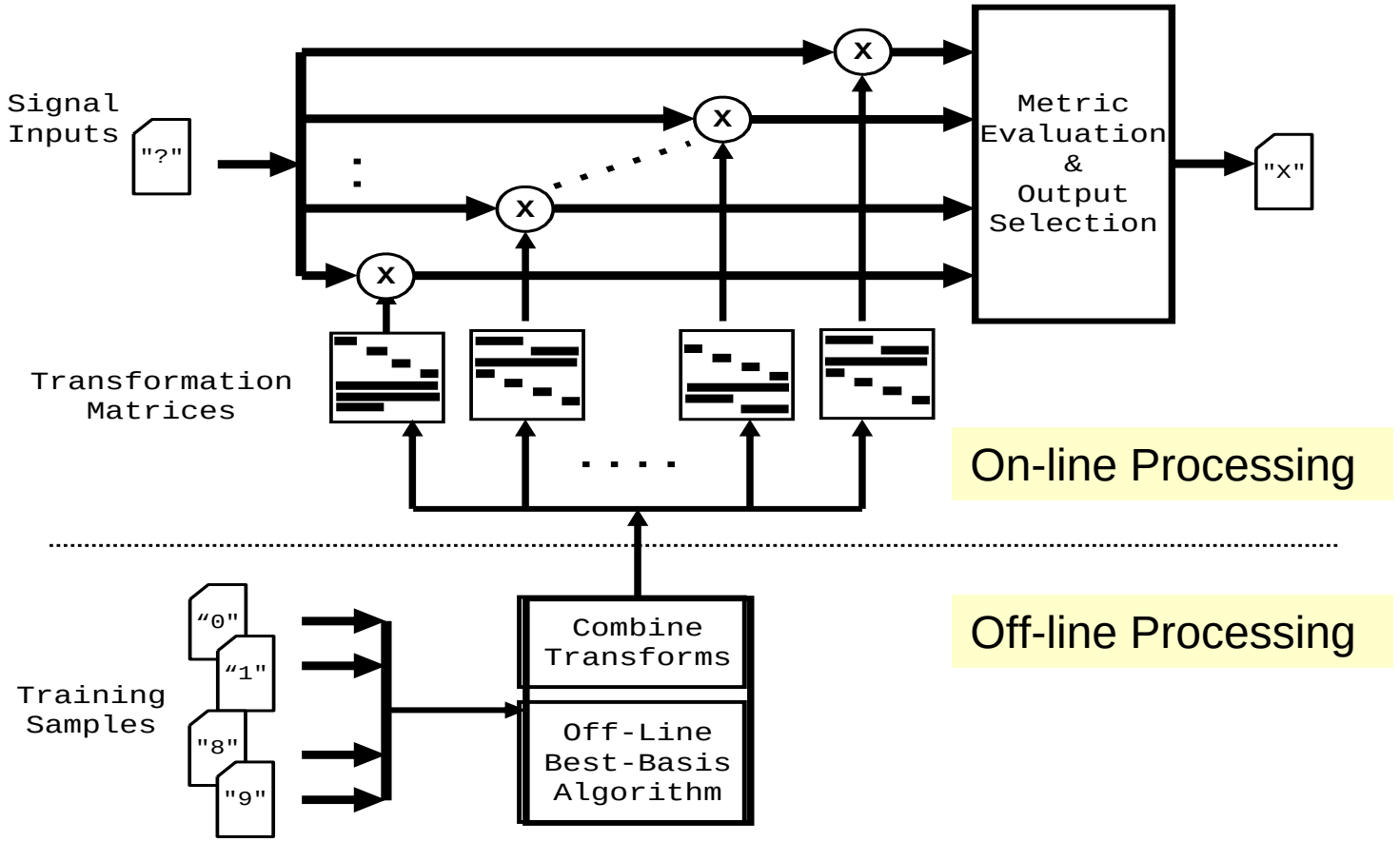
Goal: High-level, retargetable, steerable optimization system

Signal processing characterized by large input data, often streaming. Identify properties, possibly use result to steer application.

Examples of applications:

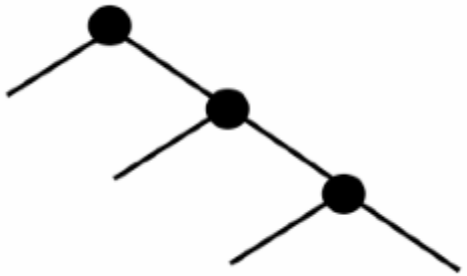
- Communication algorithms
  - Steer wireless protocol choice
- Environmental sensing
  - On-line modeling
- Signal and image recognition (see example)

Example: Digit Recognition



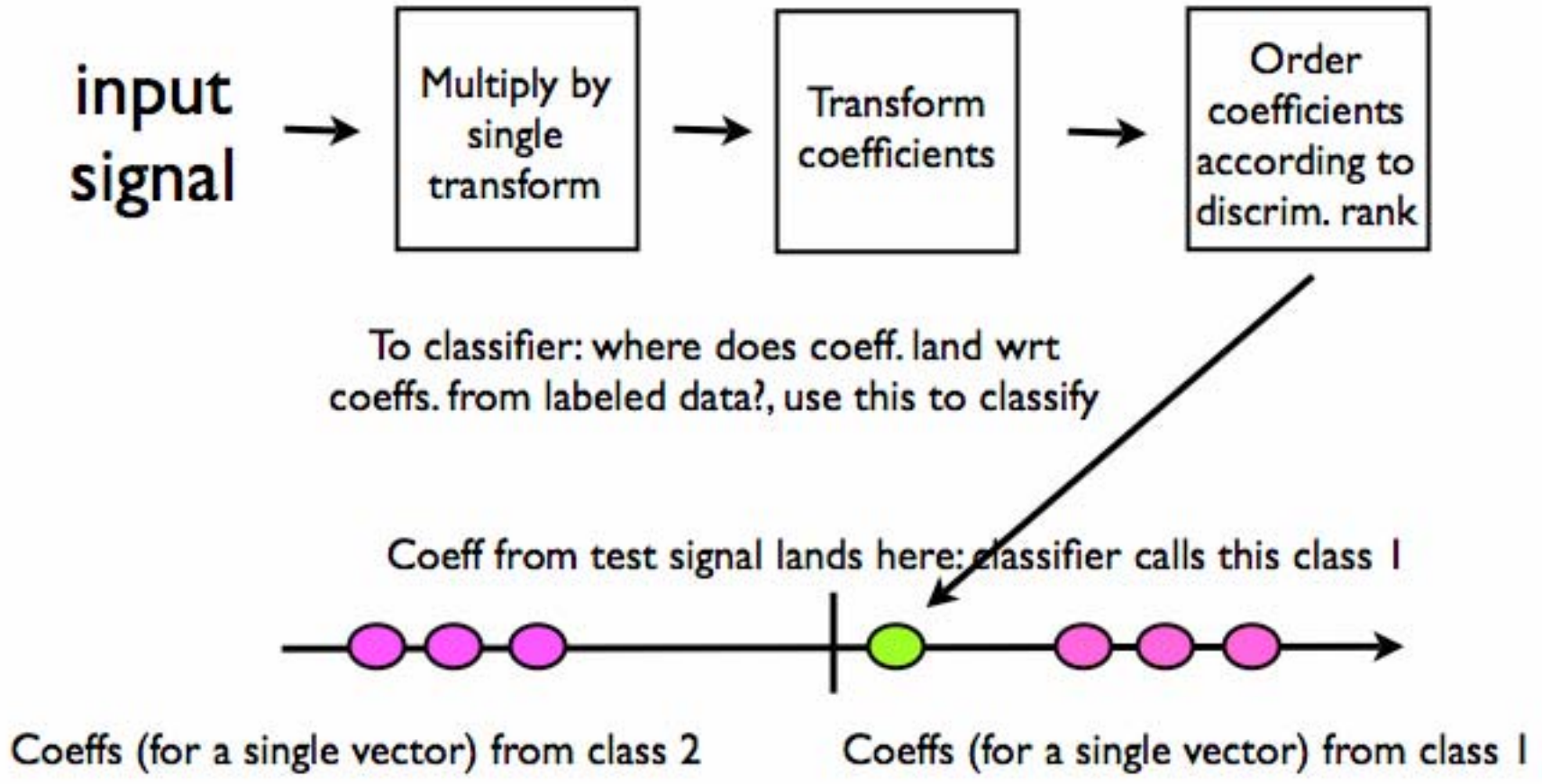
Solution: Decision Tree

- Each node of the tree:
  - "Does the input belong to a certain class?"
- Many decision trees for 11 spoken digits
- Initial solution:
  - Use one transform to detect each class (i.e., digit)
  - Test input one class at a time
  - Define feature vector for class



Derive Transform for Each Class

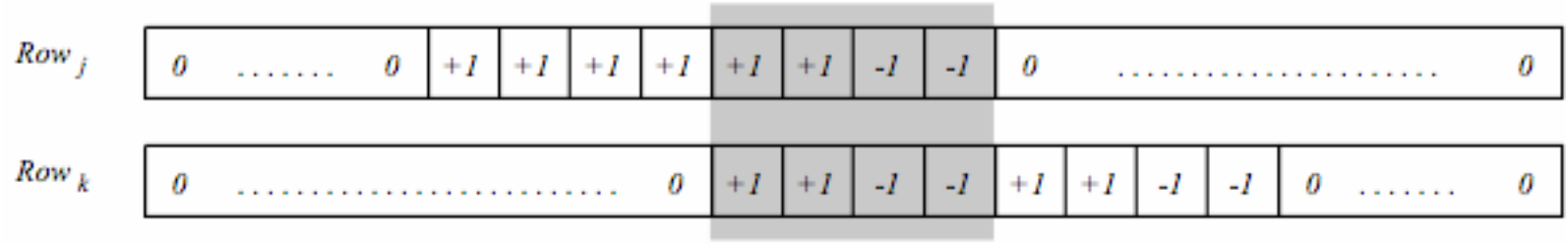
- How to derive transform
  - Let class 1 = {"oh"} and class 2 = {all others}
  - Build a single transform to separate two classes at a time
- Which transform to use?
  - Walsh-Hadamard transform is tailored to each class
- Offline BestBasis Algorithm
  - Search over exponentially many transforms to select transform that best distinguishes two classes
  - Orthonormal basis selected based on discriminatory ability
- Online Recognition Algorithm
  - On input, compute and order coefficients, send to classifier (for each node)


input signal
Multiply by single transform
Transform coefficients
Order coefficients according to discrim. rank
To classifier: where does coeff. land wrt coeffs. from labeled data?, use this to classify
Coeff from test signal lands here: classifier calls this class 1
Coeffs (for a single vector) from class 2
Coeffs (for a single vector) from class 1

Compute optimized Matrix-Vector Multiply

[Transform matrix] X [input signal]

Exploit structure to compress, reduce computation and parallelize.



Reuse partial results.

Family of Orthogonal Transforms

Purpose:

- Each one reveals different structure in speech classes
- Each transform is computed with a different matrix

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Optimization strategy, exploit structure of transform:

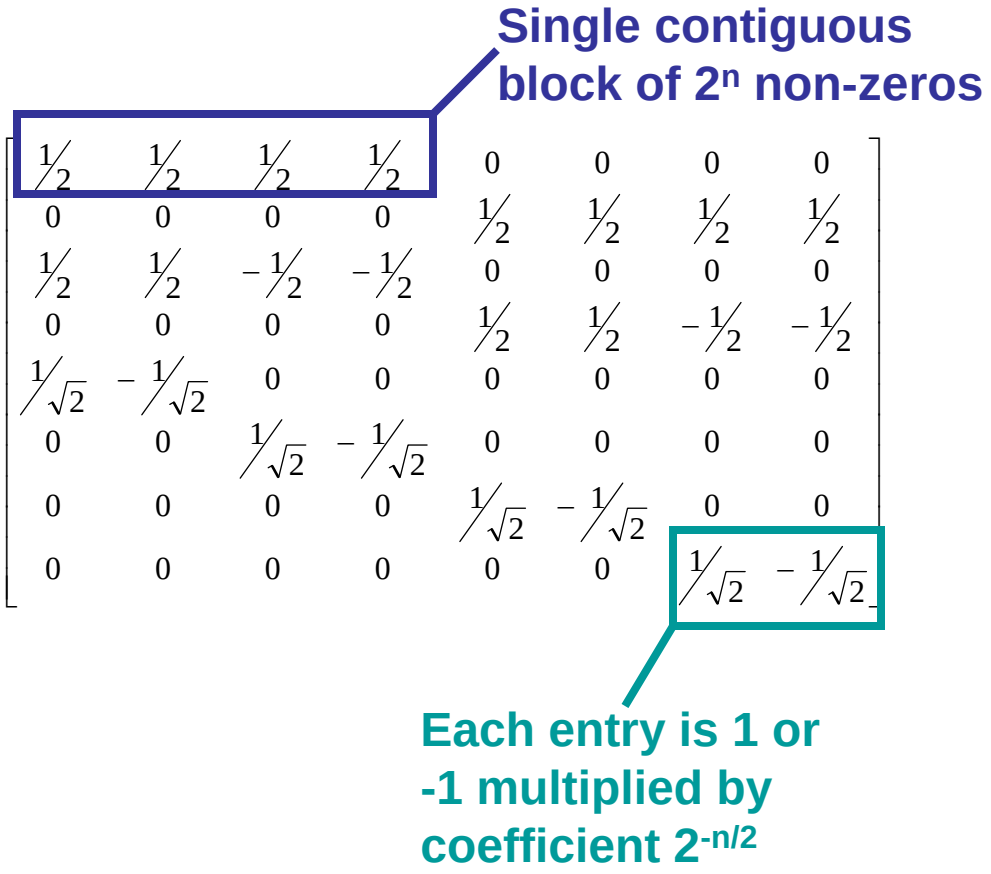
- Compress representation
- Reduce computation

$$\begin{bmatrix} \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \end{bmatrix}$$

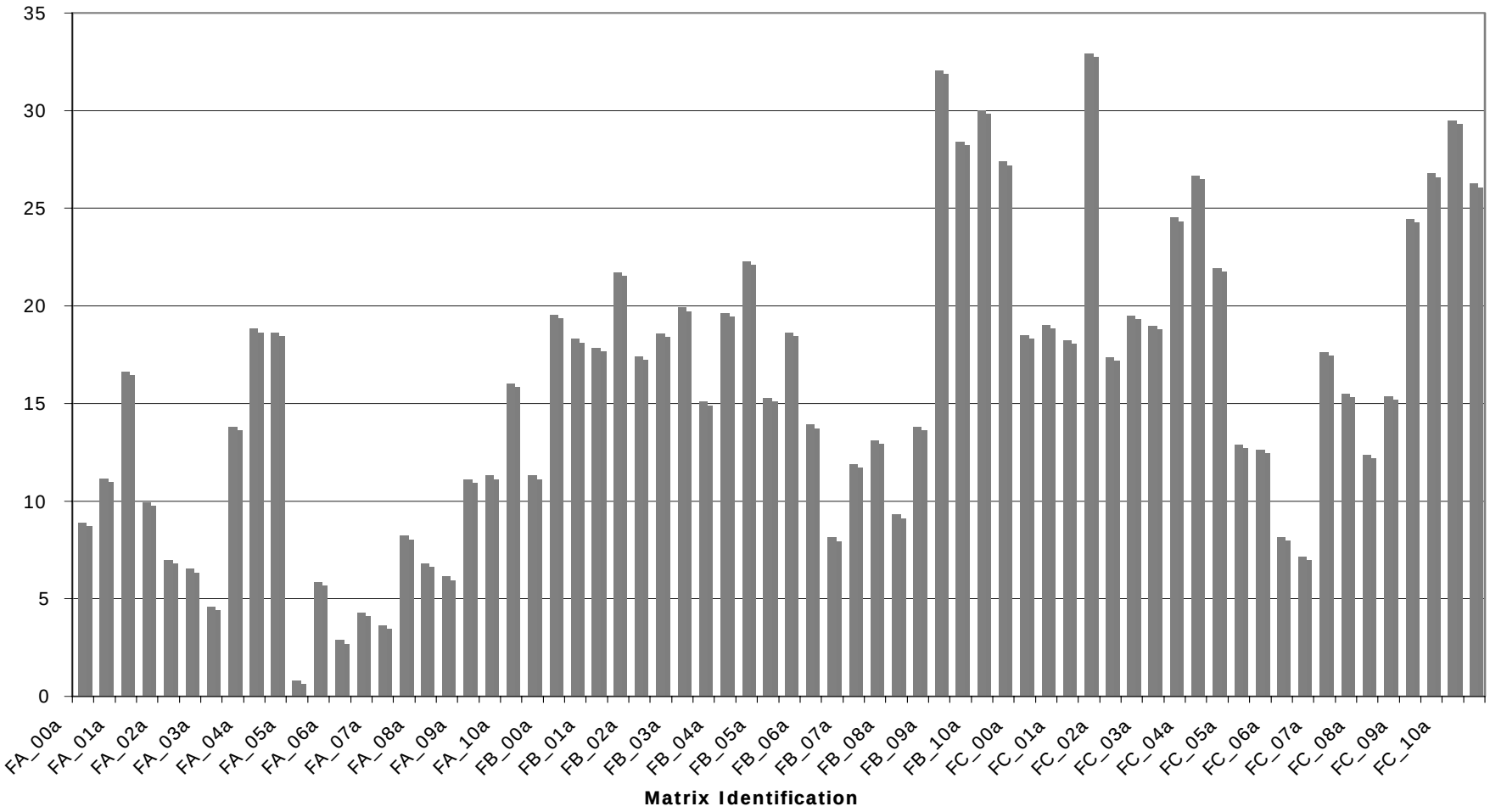
Optimization: Compressed Representation

Characterization of each row:

- **n** where 2^n is the block length and $2^{-n/2}$ is the coefficient for the block
- **index**, the position of the start of the block
- **B(index,index+2^n-1)**, a set of 2^n bits with each bit representing the sign of the corresponding nonzero element



Nonzero Elements in Matrix



Female voices (A, B and C) from TIDIGITS corpus.
Average number of nonzero elements in transform matrices = **10%**

UNIVERSITY OF MICHIGAN

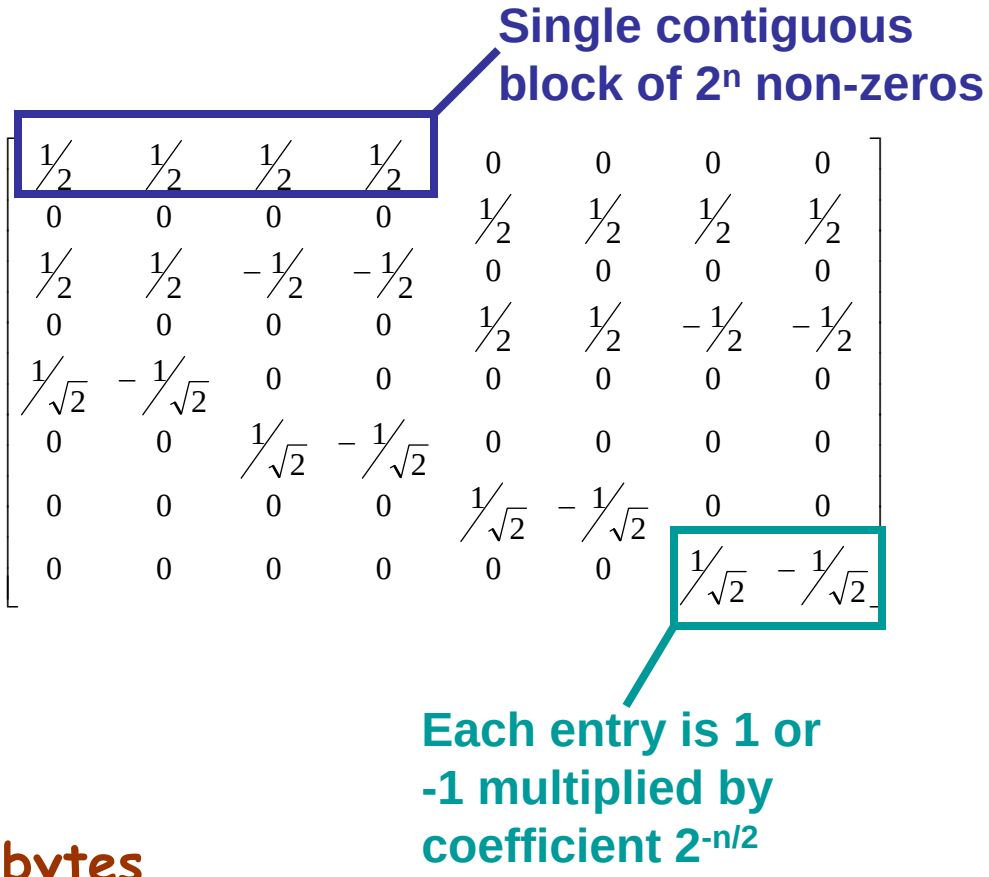
USC Viterbi School of Engineering

UTL

Optimization: Compressed Representation

Characterization of each row:

- **n** where 2^n is the block length and $2^{-n/2}$ is the coefficient for the block
- **index**, the position of the start of the block
- **B(index,index+2^n-1)**, a set of 2^n bits with each bit representing the sign of the corresponding nonzero element



Optimization:

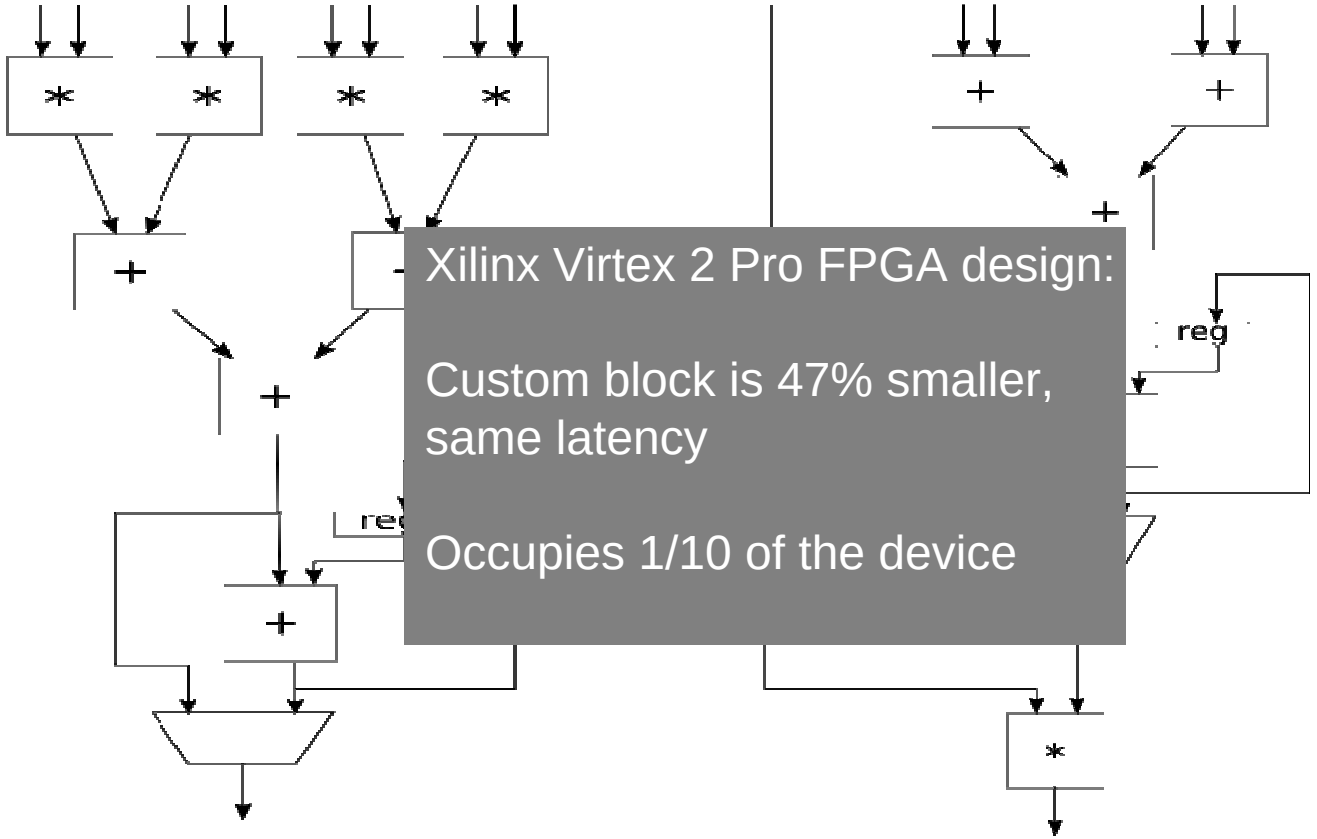
4096x4096 32-bit matrices = **60Mbytes**

OR Per row: **12 bits** for **n** and **index**, up to **4096 bits** for **B** = **2Mbytes**

OR **10% populated** = **200 Kbytes**

Optimization: Hardware Structure

Sign bit from matrix used to negate some inputs



Xilinx Virtex 2 Pro FPGA design:

Custom block is 47% smaller, same latency

Occupies 1/10 of the device

Sum of 4 input signal elements

2^{n-2} accumulations

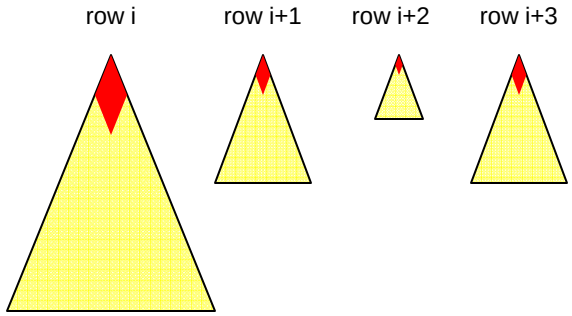
Multiply by coefficient from lookup

Standard 4x4 MV multiply block

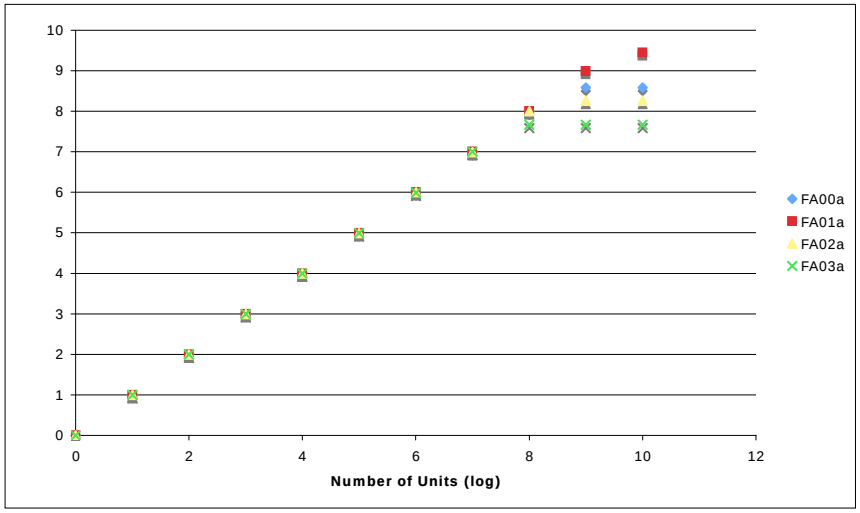
Custom 4x4 block

Optimization: Parallelism & Scheduling

- Opportunities:
  - One non-zero block, one Matrix-Vector multiplication task
  - Schedule larger blocks first
  - For each non-zero block, pipeline independent additions in the computation



- Results:
  - Lots of instruction-level and task-level parallelism for 4K matrices
  - Nearly perfect speedups up to 256 4 input units (simulated)

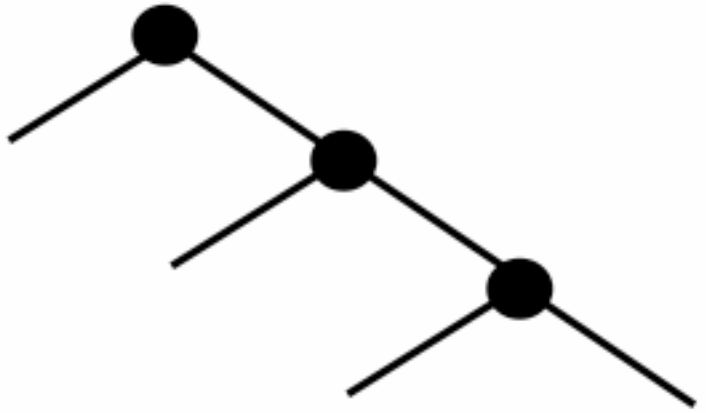


Early Project Goals

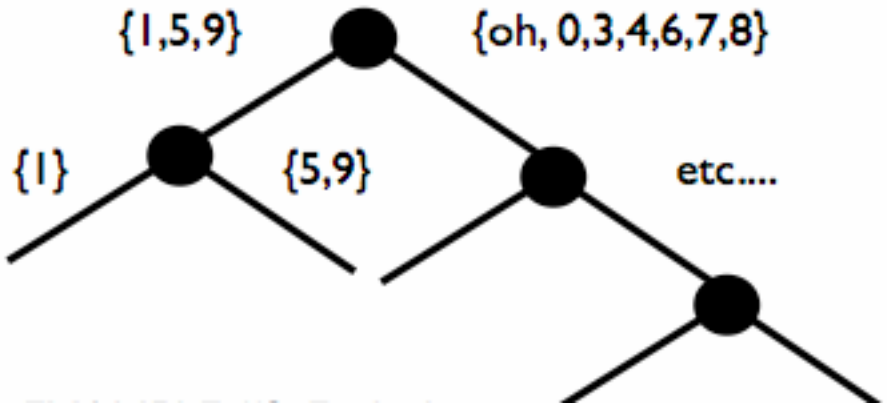
- Completed
  - Define joint optimization opportunities
  - Design signal recognition system
  - Nearly complete implementation (finish this summer)
- Next steps
  - Validation and evaluation across different optimization strategies
  - Decision tree algorithm (add "dynamic" part)
  - High-level representation

Future Work:
Optimize over decision trees

EXAMPLE 2: Each decision separates one class from the remaining classes---3 decisions to classify 4 classes



EXAMPLE 1: One decision results in 7 outputs



EXAMPLE 3: Each decision separates different classes into different subsets of classes; very general procedure

Take-Away Bullet

- Customize signal processing task for DDDAS requirements from a high-level specification and joint optimization framework