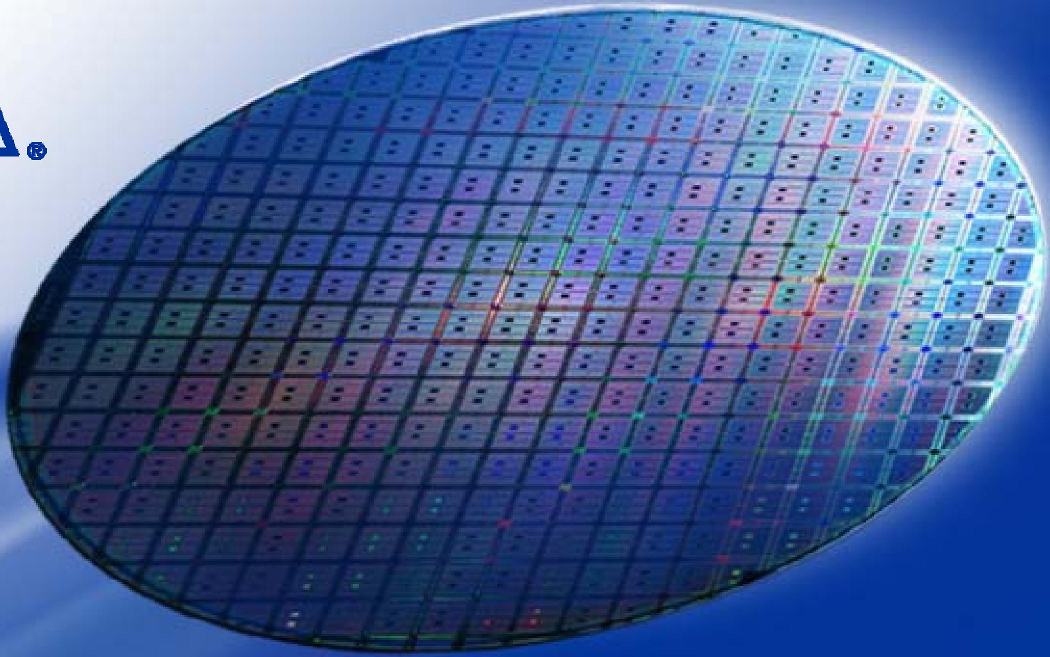


Reconfigurable Antenna Processing with Matrix Decomposition Using FPGA-Based Application-Specific Integrated Processors

Mike Fitton   *Rob Jackson*   *Steve Perry*

Altera European Technology Centre

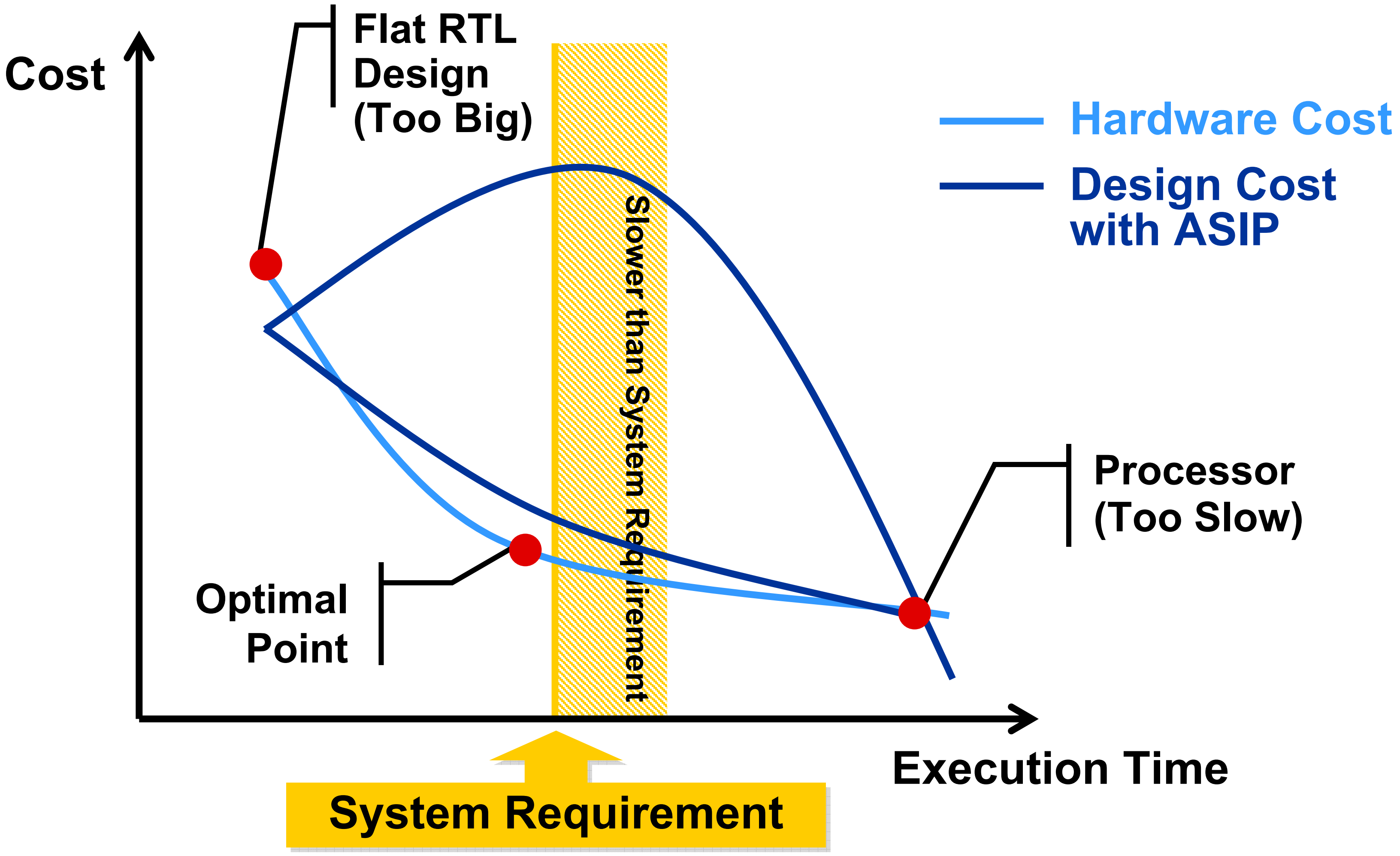
Agenda

- Motivation
- Application-Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

Motivation: System Concept

- Reconfiguration for
  - SDR: Different Standards/Specifications
  - Optimization: Adapt to the Wireless Environment
  - Multiple Input Multiple Output ⇔ Space Time Coding ⇔ Adaptive Antennas
- Provide a Methodology to Give:
  - Reconfiguration
  - Cost/Performance Trade-off
  - Usability: Present a Software-Like Interface

Accessing the Optimal Point



Agenda

- Motivation
- Application-Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

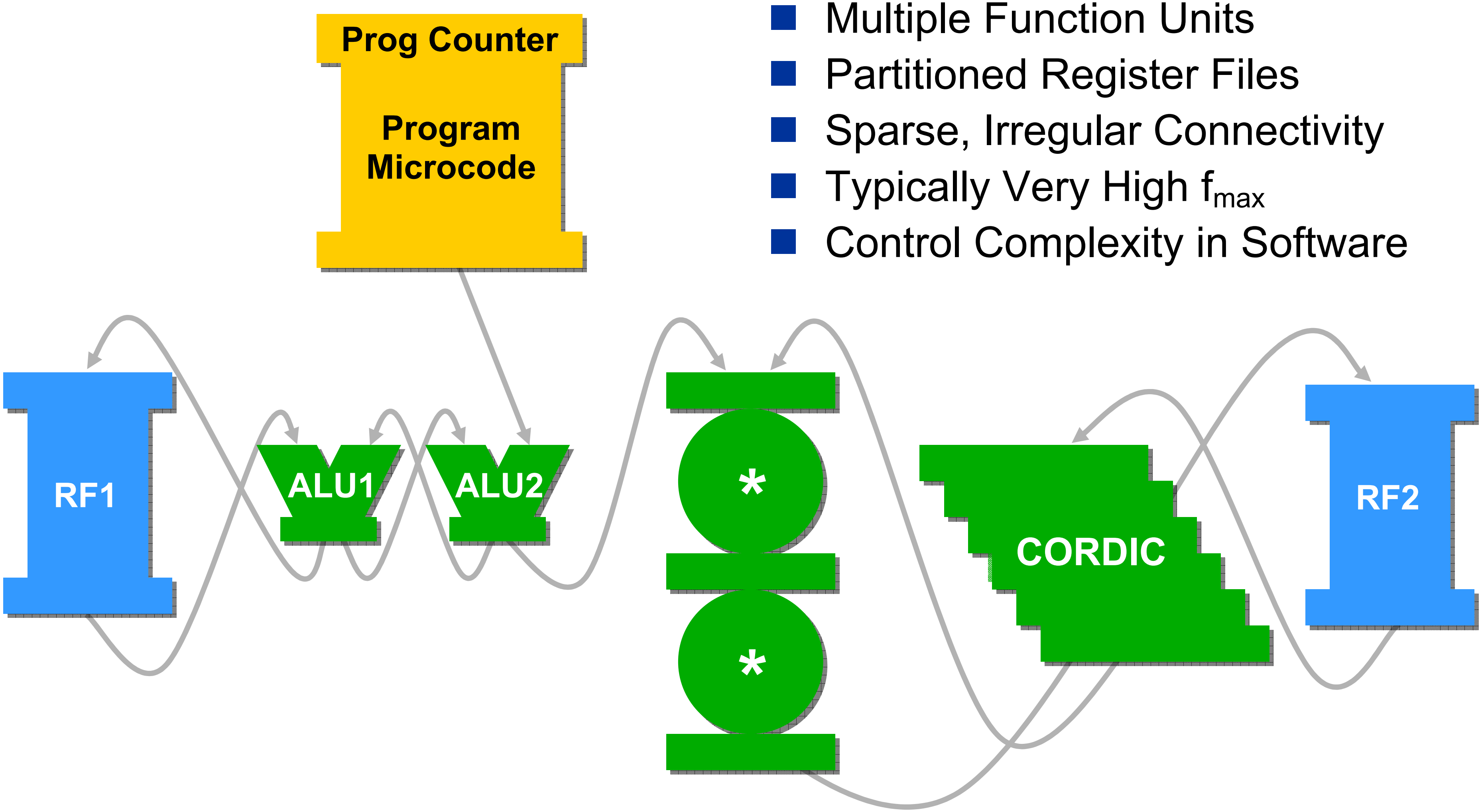
ASIP Approach

- Build Application-Specific Processors
  - Easy to Think of Them as Custom Processors
- Application-Specific Integrated Processor Has
  - Flexible Number of Functional Units
  - Connections Between Function Units as Defined by the Algorithm
  - Program to Allow Flexible (& Dynamic) Control
  - Software-Based Design Flow

Software Process Brings Productivity vs. RTL

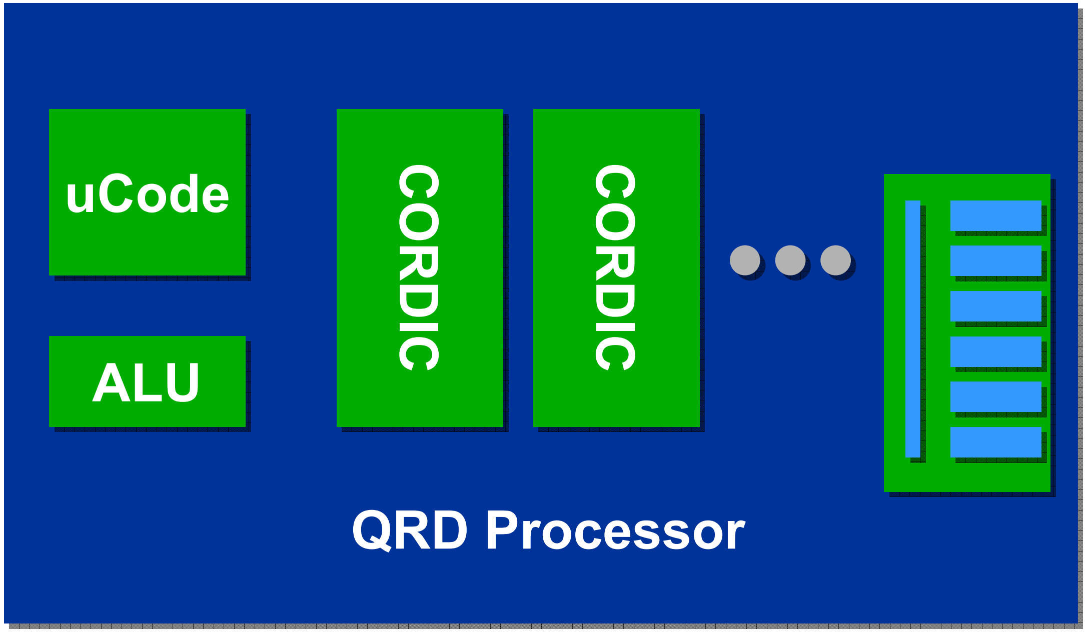
Example ASIP

- Multiple Function Units
- Partitioned Register Files
- Sparse, Irregular Connectivity
- Typically Very High f_{max}
- Control Complexity in Software



Hierarchical Composition

- Processors That Are Built Can Be Used As Functional Units Within Another Processor
- Users Can Add Functional Units to Their Processors



Agenda

- Motivation
- Application-Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

QR Decomposition Overview

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}$$

Given Observation X, Desired Output z, Solve for Optimum Coefficients w

$$\begin{bmatrix} -0.1 & 0.9 & 0.4 \\ -0.5 & 0.3 & -0.8 \\ -0.8 & -0.3 & 0.4 \end{bmatrix}\begin{bmatrix} -8 & -10 & -11 \\ 0 & 0.9 & 1.8 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}$$

Decompose X Into QR, Where Q Is Unitary (Q Q^T=1) & R Is Upper-Triangular

$$\begin{bmatrix} -8 & -10 & -11 \\ 0 & 0.9 & 1.8 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -0.1 & -0.5 & -0.8 \\ 0.9 & 0.3 & -0.3 \\ 0.4 & -0.8 & 0.4 \end{bmatrix}\begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}$$

Solve for w By Rearranging Q to RHS, Where $Q^{-1} = Q^T$

$$Z' = Q^T Z$$

Use Backsubstitution to Solve for w

Adaptive Filter Techniques

- Track Variations in the Data to Give Optimum Filter Coefficients
- Least Mean Squares (LMS) Algorithms
  - Advantage: Simple to Implement
  - Disadvantage: Relatively Slow Convergence Time
- Least Squares (LS) Algorithms
  - Advantage: Faster Rate of Convergence Over LMS
  - Disadvantage: Computationally Intensive
    Can Exhibit Divergence, Due to
    Finite Wordlength

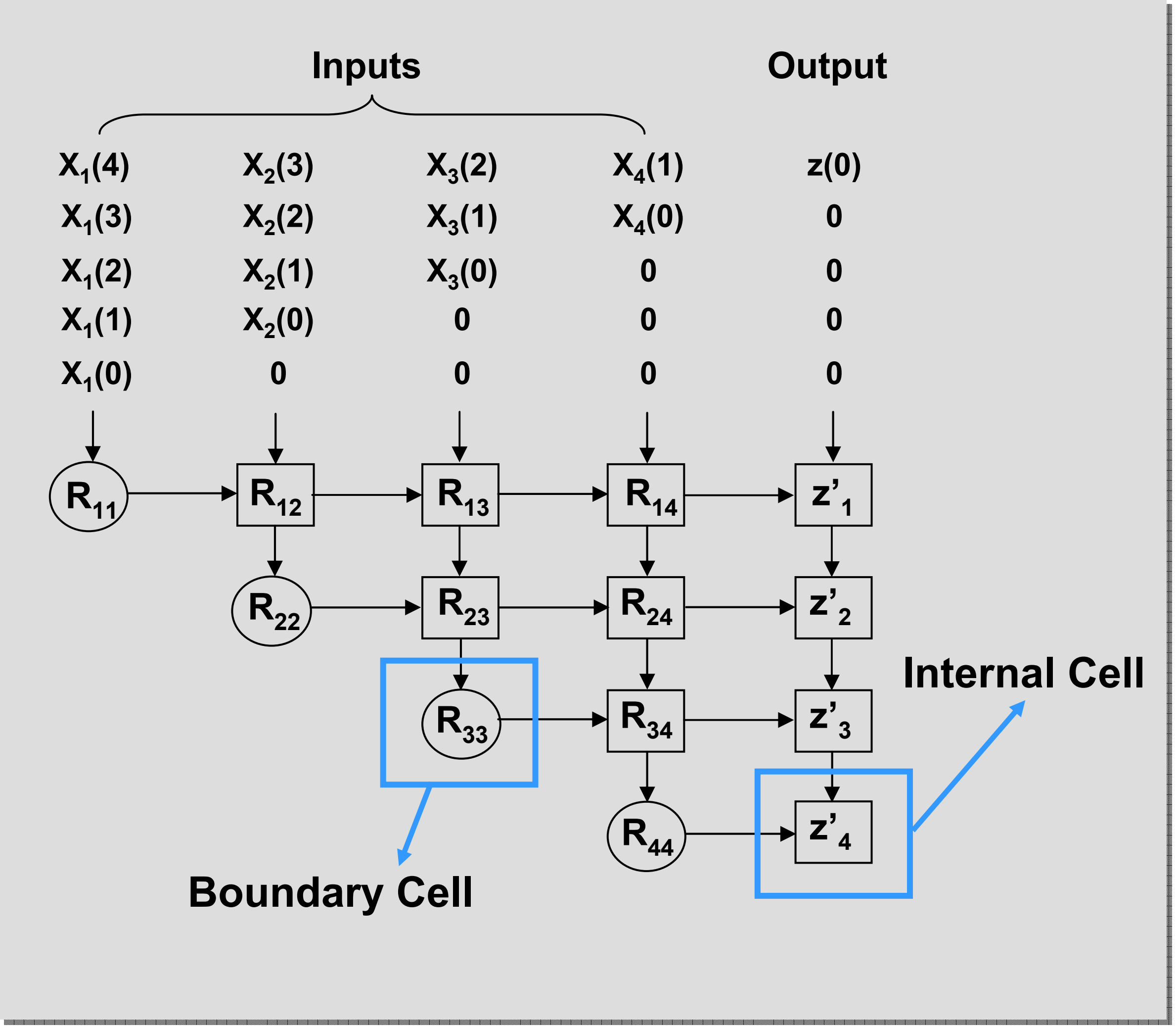
Adaptive Filter Techniques (Cont.)

- Recursive Least Squares (RLS) Is LS Algorithm
  - QR Decomposition (QRD) Can Be Used to Implement RLS
    - Square-Root Formulation of RLS
    - Advantage: Numerically Stable as Operates on Incoming Data not Time-Averaged Correlation Matrix of Input As Used in Standard RLS
    - Uses Parallel Processing Elements → Suited for Hardware Implementation

Systolic Array Architecture

- Use Systolic Array to Perform QR Decomposition
  - Efficient Hardware Implementation
- Array of Parallel Processing Cells
- Inputs & Outputs Fed Into Different Columns of Array
- 2 Types of Processing Cells
  - Boundary Cell
  - Internal Cell
- Result in Each Cell Forms R Matrix & Z' Matrix Values

Systolic Array Architecture (Cont.)

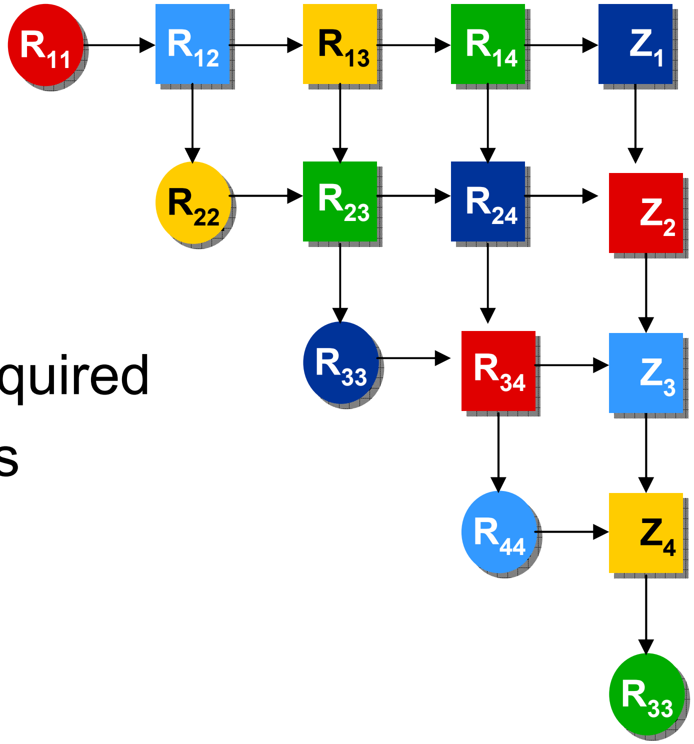

Inputs
Output
$X_1(4)$ $X_2(3)$ $X_3(2)$ $X_4(1)$ $z(0)$
$X_1(3)$ $X_2(2)$ $X_3(1)$ $X_4(0)$ 0
$X_1(2)$ $X_2(1)$ $X_3(0)$ 0 0
$X_1(1)$ $X_2(0)$ 0 0 0
$X_1(0)$ 0 0 0 0
R_{11} R_{12} R_{13} R_{14} z'_1
R_{22} R_{23} R_{24} z'_2
R_{33} R_{34} z'_3
R_{44} z'_4
Internal Cell
Boundary Cell

Agenda

- Motivation
- Application-Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

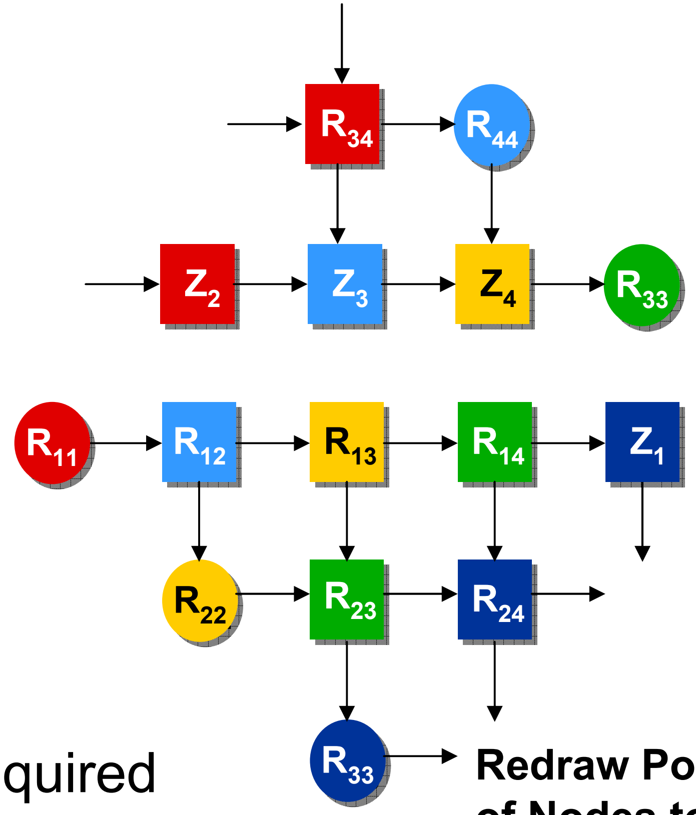
Discrete Mapping

- Method of Mapping Nodes to Limited Hardware Resources
  - One Performs Only Boundary Cell Operations
  - Others Perform Internal Cell Operations
  - Allows Optimizations of Processors
  - Minimum Amount of Memory Required
- Other Resource-Sharing Techniques Possible



Discrete Mapping

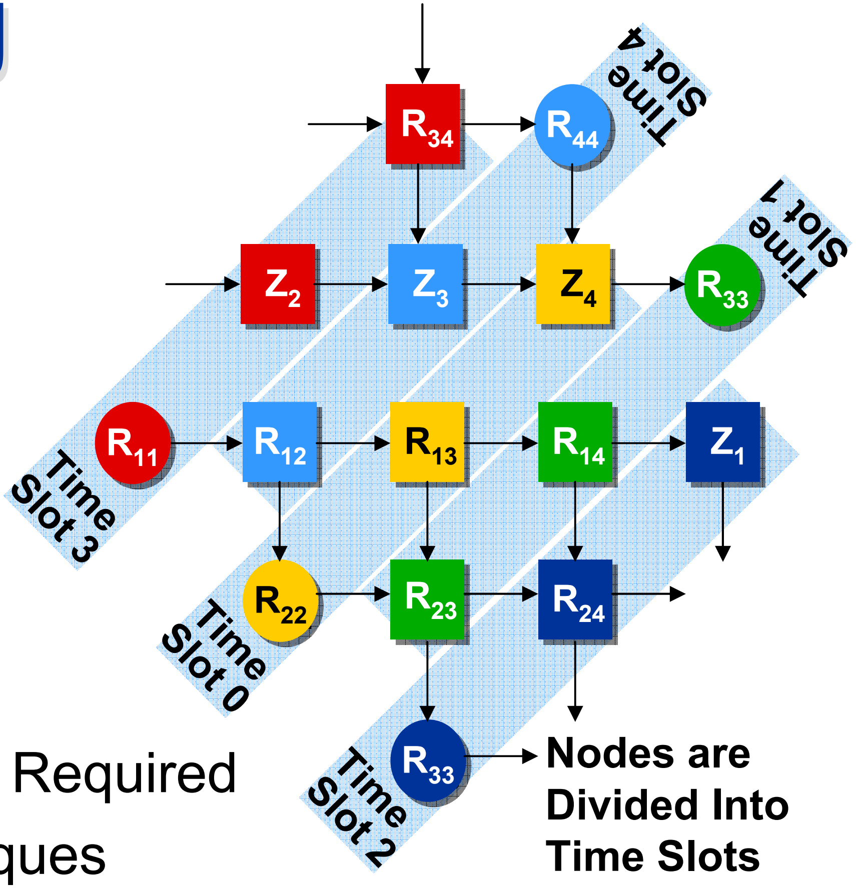
- Method of Mapping Nodes to Limited Hardware Resources
  - One Performs Only Boundary Cell Operations
  - Others Perform Internal Cell Operations
  - Allows Optimizations of Processors
  - Minimum Amount of Memory Required
- Other Resource-Sharing Techniques Possible



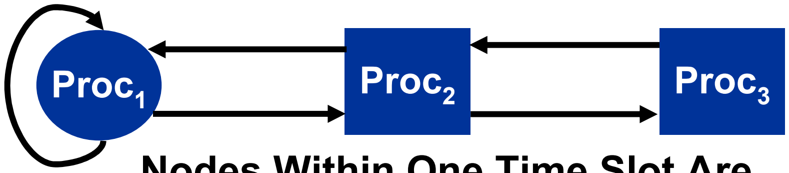
Redraw Position of Nodes to Allow Mapping Onto Sequential Processors

Discrete Mapping

- Method of Mapping Nodes to Limited Hardware Resources
  - One Performs Only Boundary Cell Operations
  - Others Perform Internal Cell Operations
  - Allows Optimizations of Processors
  - Minimum Amount of Memory Required
- Other Resource-Sharing Techniques Possible

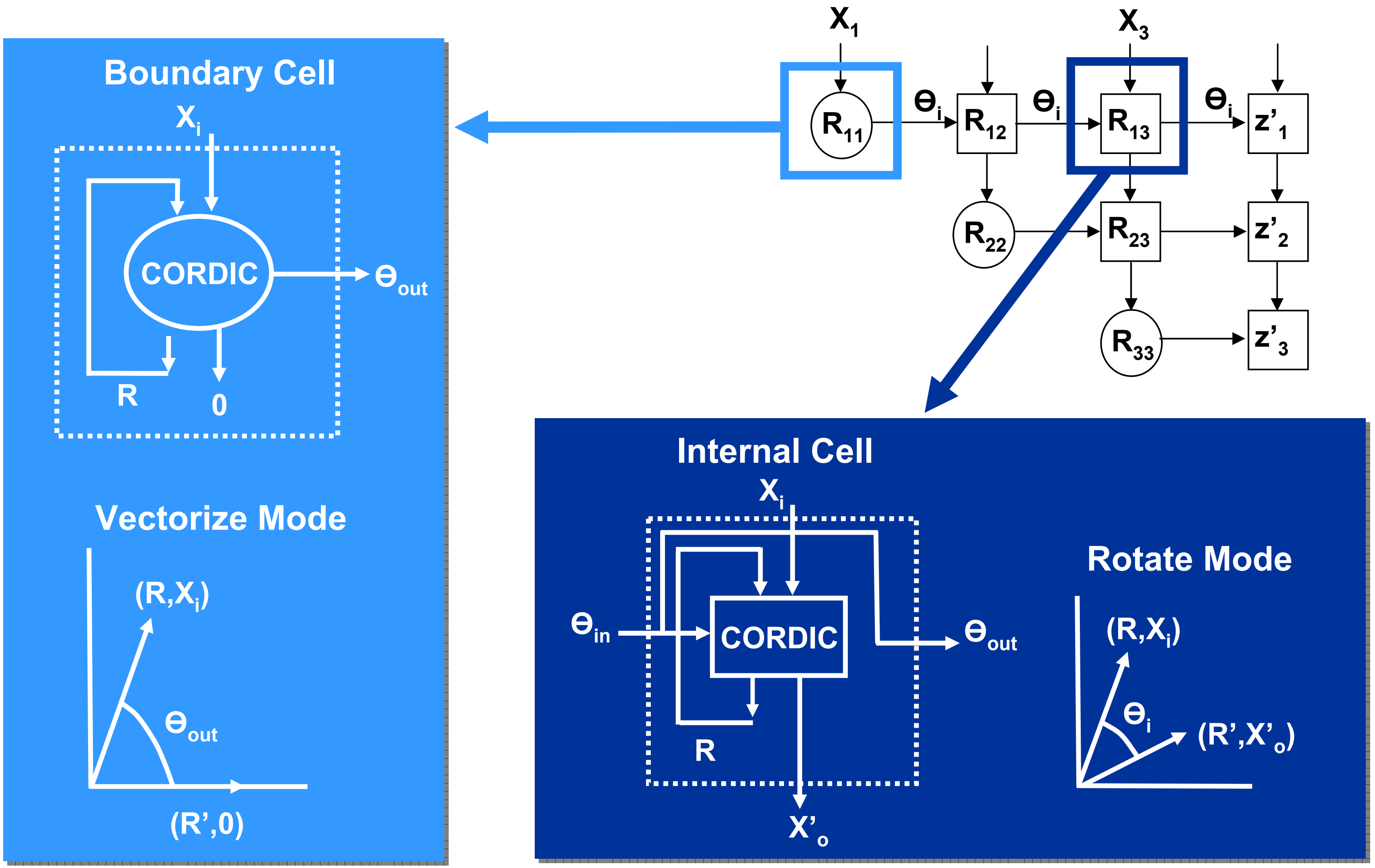


Nodes are Divided Into Time Slots When They Will Be Processed



Nodes Within One Time Slot Are Mapped Onto Processors

CORDIC as Processor Cell for Real Inputs



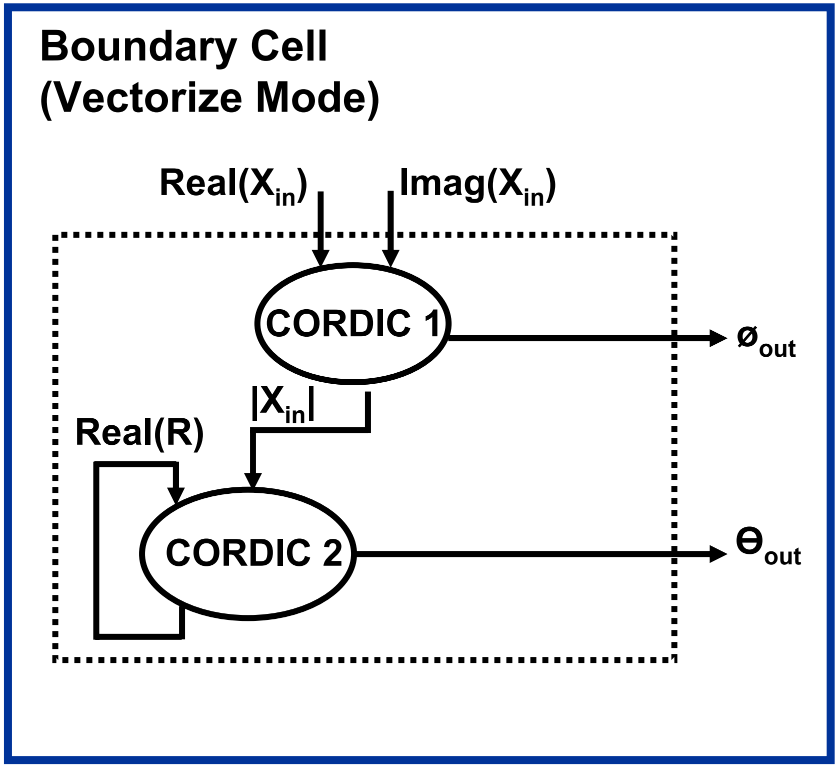
Boundary Cell

Vectorize Mode

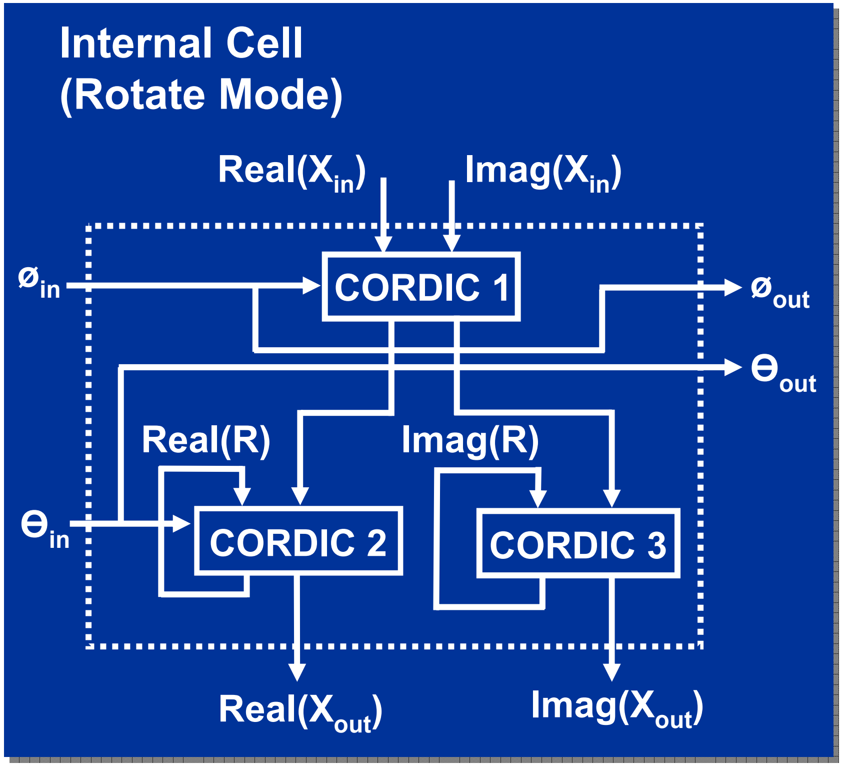
Internal Cell

Rotate Mode

Complex Operations With CORDIC



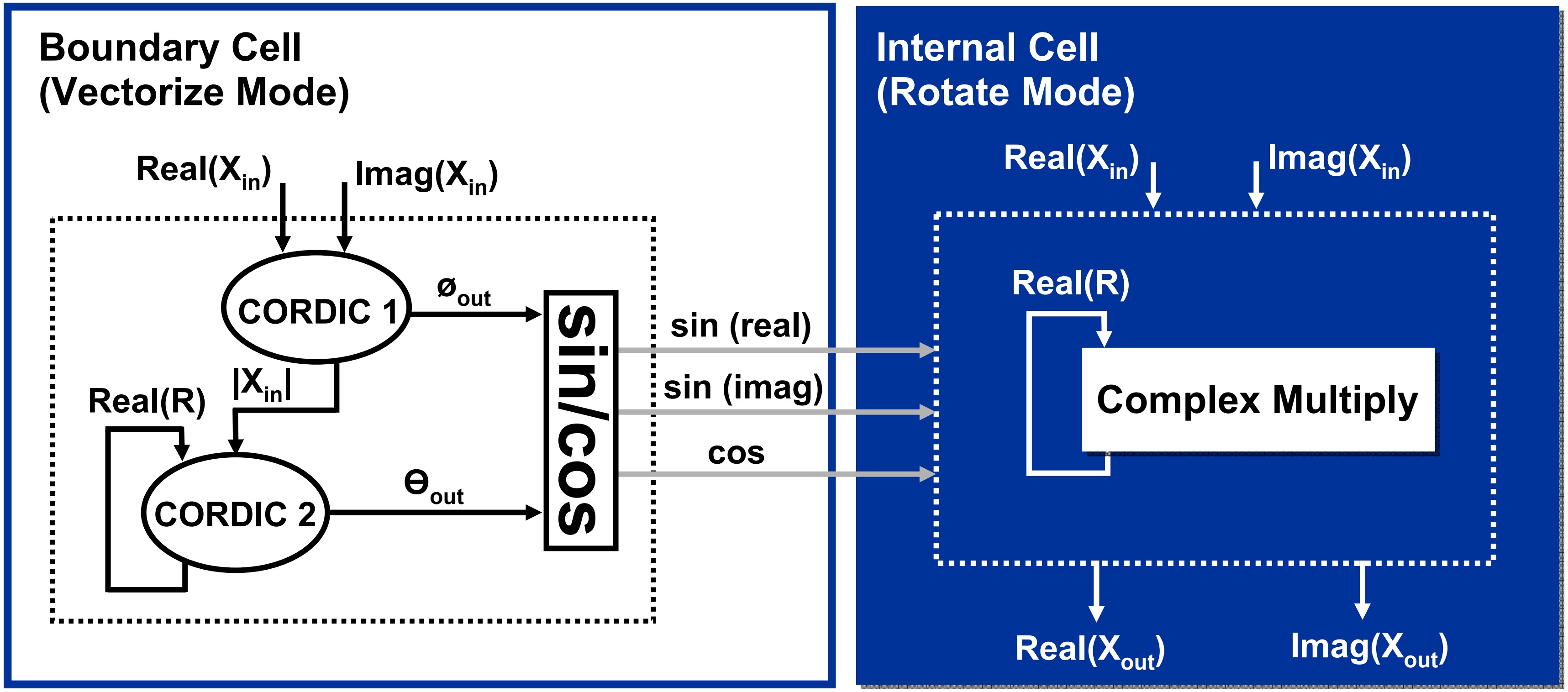
Boundary Cell
(Vectorize Mode)



Internal Cell
(Rotate Mode)

- Time Share Processors Between Nodes
- Time Share Within Processor (e.g., Use a Single CORDIC)

Mixed Cartesian/Polar Processing



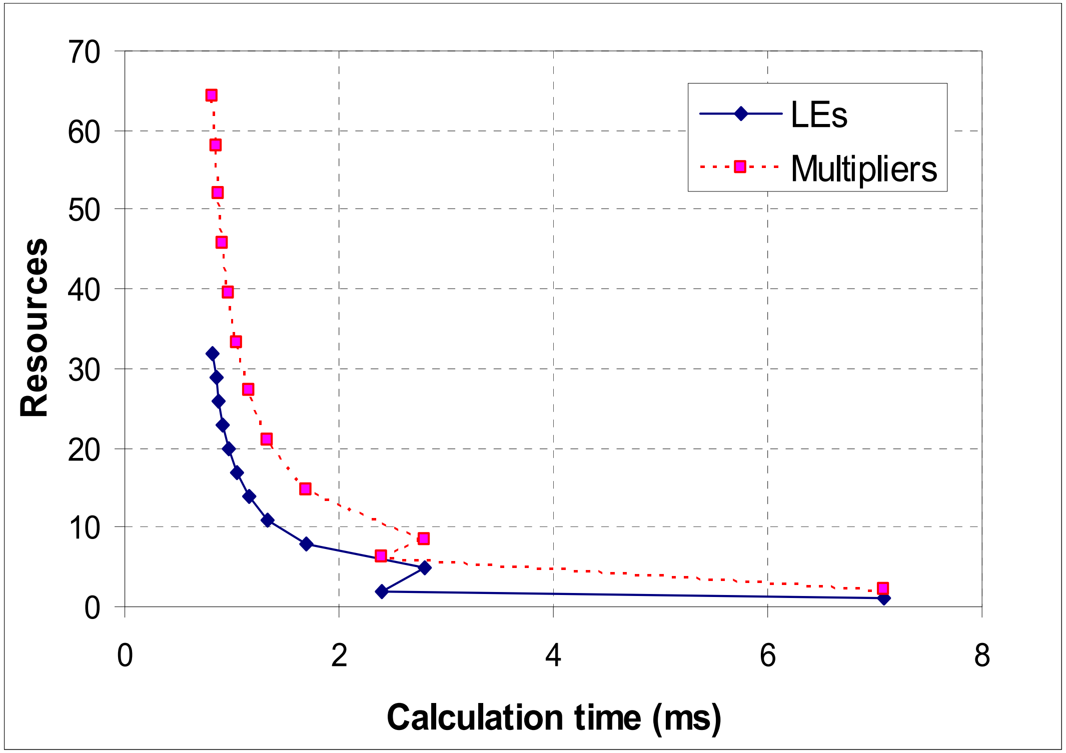
- Different Implementation of Node Processor
  - Transparent to Higher-Level Architecture
- Exploit Hard Multipliers in Internal Cells

Implementation of Back Substitution

- Soft Nios® II Processor:
  - Flexibility in Changing R Matrix Size
  - 5,623 Cycles for 20 Coefficients
- Dedicated Hardware
  - Faster than Processor Solution
  - 728 Cycles for 20 Coefficients
- Application-Specific Integrated Processor
  - Can Optimize for Speed or Size
  - Between 460 & 1,144 Cycles for 20 Coefficients

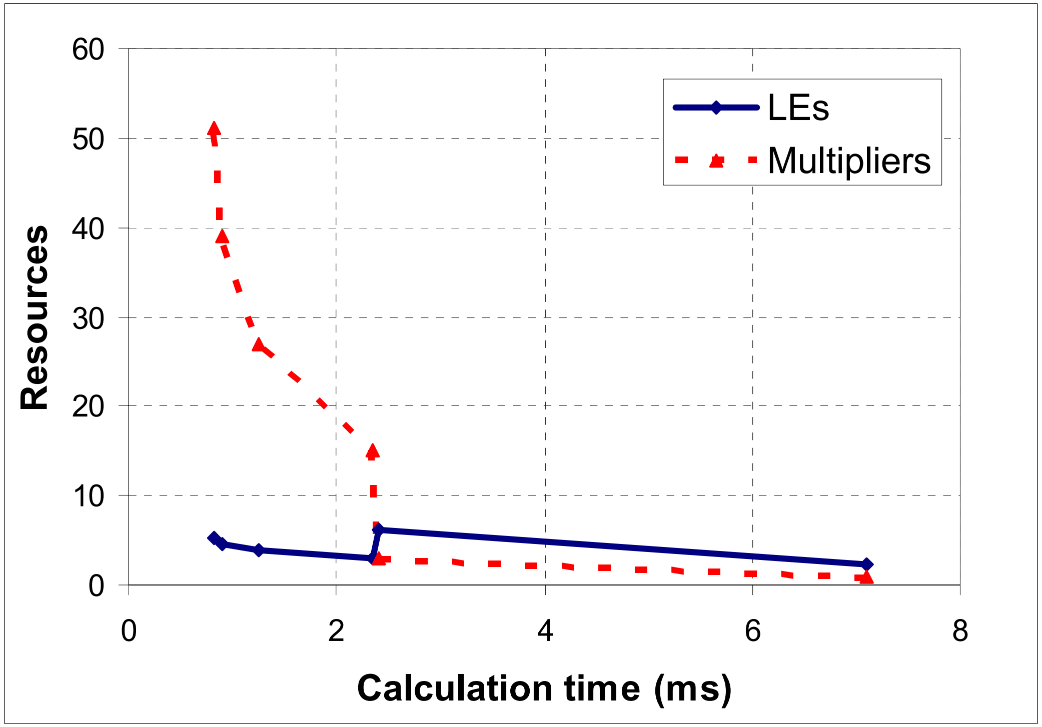
Results & Resource Utilization (i)

- 20 Coefficients, 18 bit, with 2,048 Iterations
- Full CORDIC Implementation Shown Here
  - Multipliers to Remove CORDIC Scaling



Results & Resource Utilization (ii)

- Similar Top-Level Architecture With Mixed Polar & Cartesian Processing
- Higher Multiplier Utilisation, Less Logic
  - Design Space Exploration Allows Speed & Size Trade-Off



Agenda

- Motivation
- Application-Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

Reconfiguration & ASIPs

- ASIPs Use a Program
  - Sequential Instruction-Level Parallelism
  - Can be Quickly & Easily Reprogrammed by Changing the Contents of Program Memory
- Conventional Hardware Description
  - Behavior Encoded in a Number of Parallel Processes
  - Reconfiguration Requires new FPGA Image - Literally Rewiring the Device
- ASIPs Can Combine Flexibility of DSPs with Processing Power of FPGAs
- Migrate to Structured ASIC (e.g. Altera Hardcopy Structured ASIC)
  - Does Not Compromise the Reconfigurability

SDR Aspects

- Different Levels of Reconfiguration
- Parameterization
  - Dynamically Change Update Rate, Size, Adaptation, etc.
  - Change the Cost-Performance Trade-Off
- Algorithmic Reconfiguration
  - MIMO for Multipath-Rich Channels & High Data Rates
  - Space-Time Coding for Robustness
  - Adaptive Beamforming to Increase Signal-to-Noise Ratio
- Application Reconfiguration
  - QRD for Polynomial-Based Digital Predistortion
  - QRD for Channel Estimation
  - QRD for Antenna Beamforming

Agenda

- Motivation
- Application Specific Integrated Processors (ASIPs)
- QR Decomposition & QRD-RLS
  - Introduction
  - Systolic Array
- ASIP Implementation of QRD-RLS
  - Architecture
  - Results
- SDR Aspects
- Summary

Summary

- Methodology for Constructing Customized Application-Specific Processors on FPGA
  - Used Here for QRD-Decomposition-Based RLS
  - Appropriate Algorithm for Many Wireless Applications
- Exploit ASIP Methodology
  - Encapsulation & Abstraction
  - Time Multiplexing on Different Levels
- Efficient, Scalable Design Produced
- Straightforward to Reconfigure & Parameterize
- ASIPs Remain Reprogrammable if FPGA Design is Converted to ASIC (e.g., HardCopy Structured ASIC)