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

Mike Fitton    Rob Jackson    Steve Perry

Altera European Technology Centre

© 2004 Altera Corporation
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

Cost

Execution Time

System Requirement

Optimal Point

Flat RTL Design (Too Big)

Processor (Too Slow)

Slower than System Requirement

Hardware Cost

Design Cost with ASIP

© 2004 Altera Corporation - Confidential
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 \( f_{\text{max}} \)
- Control Complexity in Software

Prog Counter
Program Microcode

RF1
ALU1
ALU2

RF2
CORDIC

\( \ast \)
\( \ast \)
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

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

Decompose $X$ into $QR$, Where $Q$ is Unitary ($Q Q^T = I$) & $R$ is Upper-Triangular

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

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

\[
\begin{align*}
X_1(4) & \quad X_2(3) & \quad X_3(2) & \quad X_4(1) & \quad z(0) \\
X_1(3) & \quad X_2(2) & \quad X_3(1) & \quad X_4(0) & \quad 0 \\
X_1(2) & \quad X_2(1) & \quad X_3(0) & \quad 0 & \quad 0 \\
X_1(1) & \quad X_2(0) & \quad 0 & \quad 0 & \quad 0 \\
X_1(0) & \quad 0 & \quad 0 & \quad 0 & \quad 0 \\
\end{align*}
\]

Output

Boundary Cell

Internal 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 Within One Time Slot Are Mapped Onto Processors

 Nodes are Divided Into Time Slots When They Will Be Processed
CORDIC as Processor Cell for Real Inputs

Boundary Cell

Vectorize Mode

(R,X_i)

(R',0)

θ_{out}

Internal Cell

Rotate Mode

(R,X_i)

(R',X'_o)

θ_i

θ_{in}

θ_{out}

© 2004 Altera Corporation - Confidential
Complex Operations With CORDIC

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

© 2004 Altera Corporation - Confidential
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

![Graph showing resources and calculation time with LEs and Multipliers]
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)
  - 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)