

DSP Algorithm Design

RASSP E&F Module Number: 23

Copyright 1995-1999 SCRA

All rights reserved. This information is copyrighted by the SCRA, through its Advanced Technology Institute (ATI), and may only be used for non-commercial educational purposes. Any other use of this information without the express written permission of the ATI is prohibited. Certain parts of this work belong to other copyright holders and are used with their permission. All information contained, may be duplicated for non-commercial educational use only provided this copyright notice and the copyright acknowledgements herein are included. No warranty of any kind is provided or implied, nor is any liability accepted regardless of use.

The United States Government holds “Unlimited Rights” in all data contained herein under Contract F33615-94-C-1457. Such data may be liberally reproduced and disseminated by the Government, in whole or in part, without restriction except as follows: Certain parts of this work to other copyright holders and are used with their permission; This information contained herein may be duplicated only for non-commercial educational use. Any vehicle, in which part or all of this data is incorporated into, shall carry this notice .

Module Objectives:

To educate the system designer on the concepts and methods used for the development of algorithms for digital signal processing systems.

Specific Objectives:

Provide information on:

- 1) Introduction to the need for algorithm design methodologies
- 2) Requirements capture
- 3) Fixed point design - a RASSP approach
- 4) HW/SW partitioning overview
- 5) Simulation-based algorithm/functional design
- 6) Network level DSP design
- 7) Link level DSP design

- 8) Signal processing simulators
- 9) PGM/PGSE linking to implementation - Case study
- 10) HW/SW Codesign Overview

Prerequisites:

Prerequisite Modules:

None

Prerequisite Knowledge:

None

Syllabus:

- 1) Introduction (20 Min.)
 - a) Complexity of DSP/Communications systems
 - b) Structure of an application specific system
 - c) Top down design methodologies
 - d) Requirements for algorithm design
- 2) Requirements capture (20 Min.)
 - a) Basic approach to requirements capture
 - b) Primary and secondary requirements
 - c) Testbenches and executable specifications (IRST example)
 - d) RDD-100 mission and capabilities
 - e) Capturing requirements and generating specifications using Statecharts
- 3) Fixed point design - a RASSP approach (20 Min.)
 - a) Motivation for fixed point
 - b) Representations for fixed point
 - c) Examples of fixed point
 - d) Architecture-true and bit-true simulations
 - e) Quickfix environment
 - f) HW/SW design issues
- 4) Simulation-based algorithm/functional design (10 Min.)
 - a) Trends in tools
 - b) Hierarchical approach to DSP design
 - c) Structure of an algorithm design environment
 - d) Functional tasks and input specifications
 - e) Graphical vs language inputs
- 5) Network level DSP design (20 Min.)
 - a) Petrinet and protocol verification
 - b) Examples and modeling methodologies
 - c) Event driven computation

- 6) Link level DSP design (10 Min.)
 - a) Time-driven computation
 - b) Modeling of links and transmission channels
 - c) Execution managers (links to scheduling module)
- 7) Signal processing simulators (30 Min.)
 - a) Block oriented simulators: Khoros, Blosim, SPW, Ptolemy
- 8) PGM/PGSE linking to implementation (50 Min)
 - a) PGM
 - b) Run-time environments
 - c) Examples and demos
 - d) Application development in PGM
 - e) Linking PGM to real-time schedulers
 - f) Examples and demos
- 9) HW/SW Codesign Overview (60 Min.)
 - a) Motivation
 - b) Recent HW/SW codesign methodologies
 - c) RASSP codesign methodology
 - d) VHDL virtual prototyping in RASSP
 - e) Implementation of codesign
 - f) RASSP codesign example
 - g) Conclusions

Audience:

System design engineer, first year graduate student.