www.kaztek.com

System Development and Programming with the ADSP-TS20x Family

Course Name:	System Development and Programming with the ADSP-TS20x (TigerSHARC) Family
Course Number:	WS_TS20x
Course Description:	This is a practical course with 'hands on' training using the latest VisualDSP++ software development tools. First the core elements of the processor, which includes the Computational Units, the Integer ALUs, and the Program Sequencer, are examined in detail along with the relevant assembly code instructions. A number of simulator labs help in understanding operation of the individual elements. Memory configuration (both internal and external) is discussed next. Advanced instructions are presented with a follow on lab on code optimization. The I/O peripherals, which include the Link Ports and External Port, are discussed in detail along with DMA operation between these peripherals and internal memory. This section also deals with system booting and other features including timers and SDRAM controller. Finally, hardware development tools, such as evaluation boards and ICE's are introduced. Throughout the course, the various aspects of the software development process using the latest tools are discussed including setting up and building projects, assembly language programming, code debugging, simulation, tool support for code overlays and shared memory, and 'C' programming support.
Goals/Objectives:	The main course objective is to understand the architecture of the ADSP-TS20x DSP family (including the TS201, TS202, and TS203) sufficiently to enable DSP system designers to resolve hardware/software issues with their applications. Additional goals include gaining a thorough understanding of both assembly language programming and code development (including 'C' programming issues) with the latest software tools
Pre-requisites:	Previous embedded microprocessor background would be an asset (hardware and/or software)
Target Audience:	System Designers needing to make informed decisions on design tradeoffs, Hardware Designers needing to develop external interfaces, and Code Developers needing to know how to get the highest performance from their algorithms
Target Duration:	3.5 days

1 Introduction

1.1 Introductions/ Course Overview

1.1.1 Purpose of the Course

1.1.2 Course Overview

1.1.3 Logistics (breaks, lunch, etc.)

1.1.4 Course Handouts

1.1.5 DSP at Analog

1.1.6 Analog Devices strategy

1.1.7 Signal Processor Portfolio

1.2 Introduction to ADSP-TS201

1.2.1 Characteristics of a Good DSP

1.2.2 ADSP-TS201 Features

2 Introduction to Software Tools (VisualDSP)

2.1 Software Tools Overview

2.1.1 Project development

2.1.2 VisualDSP Overview

2.1.3 Assembler Overview

2.1.4 Linker Overview

2.1.5 Loader Overview

2.1.6 Integrated Development and Debug Environment (IDDE)

2.1.7 VisualDSP Debug Features

3 Computational Block – Data Register

3.1 Registers and Data Types

3.2 Register types - overview

3.3 Register File

3.4 Native data types and data word alignment

3.4.1 Fixed point

3.4.2 Floating point

3.5 Simulator Exercise: registers exercise, basic simulator operation

4 Computational Block – ALU

4.1 Features

4.2 Instructions

4.3 Flags

4.4 Simulator Exercise: ALU operation

5 Computational Block - Multiplier/MAC

5.1 Features

5.2 Instructions

5.3 Flags

5.4 Fractional and integer math

5.5 Simulator Exercise: MAC operation

6 Computational Block - Shifter

6.1 Features

6.2 Instructions

6.3 Flags

6.4 Simulator Exercise: Shifter operation

7 Integer ALU's (IALU)

7.1 Features

7.2 Instructions

7.3 Immediate data move instructions

7.4 Modulo addressing example

7.5 Simulator Exercise: Address Generation operation

8 Memory
8.1 TigerSHARC Memory
8.1.1 Memory Basics
8.1.2 Memory Maps
8.1.3 Internal Architecture
8.2 TigerSHARC Internal Embedded DRAM
8.2.1 Internal Embedded DRAM Architecture
8.2.2 Memory Overhead Considerations
8.2.3 Internal Memory Maps
9 Program Sequencer
9.1 Features
9.2 Instructions
9.3 Instruction pipeline
9.4 Branching
9.5 Looping
9.6 Interrupts
10 Assembly Code Development with VisualDSP++
10.1 Assembler
10.1.1 Assembler Expressions
10.1.2 Assembler Directives
10.1.3 Definition files
10.2 Basic Linker Description File (LDF)
10.2.1 Introduction
10.2.2 Overview
10.2.3 Example LDF File
10.2.4 Example Commands
10.2.5 Expert Linker
10.3 VisualDSP Simulator
10.3.1 Overview
10.3.2 Simulator features

10.3.3 **Simulator Exercise: basic code development exercise**

11 Advanced Instruction Types

11.1 Advanced Math Instructions

11.2 Communications Instructions

11.2.1 Communications Registers

12 Code Optimization

12.1 Parallel Instruction Types and Multifunction Computations

12.2 Optimization techniques

12.3 Simulator Exercise*:* code optimization

13 System On Chip
13.1 SOC Structure
13.1.1 SOC Features
14 DMA Unit
14.1 DMA Architecture
14.2 DMA Features
14.3 DMA modes & examples
14.4 External Port DMA
15 External Port
15.1 Memory Interface
15.2 SDRAM interface
15.3 Shared Bus Multiprocessing (Cluster mode)
15.4 Host Interface
15.5 External Port FIFOs
16 Link Port
16.1 Link Port Features
16.2 Link Port Configuration—DMA & Control
16.3 Link Port Pin Description & Function
17 Booting
17.1 Loader Utility
17.2 Boot Loader Process
17.3 Multiprocessor booting

18 System Design

18.1 Navigating the Datasheet

18.2 Example system configuration

18.3 Design guidelines

18.4 JTAG overview

18.5 ICE emulator header connector

19 Advanced Linker Features

19.1 Features

19.2 Shared Memory

19.3 Multi-Processing

19.4 Software Overlays

20 VisualDSP++ C Compiler

20.1 C Compiler Features and use in Embedded Systems

20.2 Configuring C Compiler via IDDE

20.3 LDF for C Compiler

20.3.1 Stacks

20.4 Run Time Header

20.5 Interrupt Handling with C Code

20.6 Register Usage and Data Types

20.7 Assembly Language Interface

20.8 C Callable Assembly Functions

20.9 C Code Optimization

21 TigerSHARC Hardware Tools

21.1 Hardware Tool Overview with part numbers

21.2 ADSP-TS201 EZ-KIT

21.2.1 Hardware

21.2.2 Software

21.3 In Circuit Emulators

21.3.1 ICE Configurator

Conclusion/Questions