

Course Name:	System Development and Programming with the ADSP-TS101 (TigerSHARC)
Course Number:	ADST-160
Course Description:	<p>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.</p>
Goals/Objectives:	<p>The main course objective is to understand the architecture of the ADSP-TS101 DSP 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</p>
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-TS101

- 1.2.1 Characteristics of a Good DSP**
- 1.2.2 ADSP-TS101 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 SRAM

8.2.1 Internal SRAM 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

10.3 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 Communications Instructions

11.2 Reciprocal and Divide

11.3 Reciprocal Square Root and Square Root

12 Code Optimization

12.1 Parallel Instruction Types and Multifunction Computations

12.2 Optimization techniques

12.3 Simulator Exercise: code optimization

13 I/O Processor

13.1 IOP Structure

13.1.1 I/O Processor Features

13.2 DMA Unit

13.2.1 DMA Architecture

13.2.2 DMA Features

13.2.3 DMA modes & examples

13.2.4 External Port DMA

13.3 External Port

13.3.1 Memory Interface

13.3.2 Shared Bus Multiprocessing (Cluster mode)

13.3.3 Host Interface

13.4 SDRAM interface

13.5 Link Port

13.5.1 Link Port Features

13.5.2 Link Port Configuration—DMA & Control

13.5.3 Link Port Pin Description & Function

13.6 Timers

14 Booting

14.1 Loader Utility

14.2 Boot Loader Process

14.3 Multiprocessor booting

15 System Design

15.1 Navigating the Datasheet

15.2 Example system configuration

15.3 Memory Timing

15.4 Design guidelines

15.5 JTAG overview

15.6 ICE emulator header connector

16 Advanced LDF Features

16.1 Features

16.2 Shared Memory

16.3 Multi-Processing

16.4 Software Overlays

17 VisualDSP++ C Compiler

17.1 C Compiler Features and use in Embedded Systems

17.2 Configuring C Compiler via IDDE

17.3 LDF for C Compiler

17.3.1 Stacks

17.4 Run Time Header

17.5 Interrupt Handling with C Code

17.6 Register Usage and Data Types

17.7 Assembly Language Interface

17.8 C Callable Assembly Functions

17.9 C Code Optimization

18 TigerSHARC Hardware Tools

18.1 Hardware Tool Overview with part numbers

18.2 ADSP-TS101 EZ-KIT

18.2.1 Hardware

18.2.2 Software

18.3 In Circuit Emulators

18.3.1 ICE Configurator

Conclusion/Questions