

Outline: *System Development and Programming with the ADSP-2191*

Course Name:	System Development and Programming with the ADSP-2191
Course Number:	ADST-140
Course Description:	<p>This is a practical course with ‘hands on’ training using the latest software and hardware development tools. First the core elements of the processor, which includes the Computational Units, the Data Address Generators, 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 SPORTS, SPI, UART, Host Port, and External Port, are discussed in detail along with DMA operation between these peripherals and internal memory. System related information such as booting and hardware design issues are also discussed. Hardware development tools, such as evaluation boards and ICE’s are then 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, and ‘C’ programming support.</p>
Goals/Objectives:	<p>The main course objective is to understand the architecture of the ADSP-2191/5/6 DSP’s sufficiently to enable DSP system designers to resolve hardware/software issues with their applications. The ADSP-2195 and ADSP-2196 are memory variants of the ADSP-2191. 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 Introduction/ Course Overview

1.1.1 Who is ADI?

1.1.1.1 DSP at Analog

1.1.2 Goal of Workshop

1.1.3 Course Overview

1.1.4 Course Handouts

1.1.5 ADSP-219x Roadmap

1.1.6 ADSP-2191 DSP Core Features

1.1.7 Modified Harvard Architecture

1.1.8 ADSP-219x Block Diagram

2 Introduction to VisualDSP++

2.1 What is VisualDSP++

2.2 Software Development Flow

2.3 Integrated Development and Debugging Environment (IDDE)

2.4 IDDE Features

2.5 Project Development

2.5.1 Project Development Steps

2.5.2 Project Property Page

2.5.3 Assembler

2.5.4 Compiler

2.5.5 Linker

2.5.6 Loader

2.6 Introduction to the Debugger Front End

2.6.1 Selecting Sessions

2.6.2 User Interface

3 Computational Units

3.1 Binary Number Formats

3.2 Register File

3.2.1 Mode Register

3.2.2 Simulator Exercise

3.3 Arithmetic Logic Unit (ALU)

3.3.1 Features

3.3.2 Instructions

3.3.3 Flags

3.3.4 Simulator Exercise

3.4 Multiplier/ Accumulator (MAC)

3.4.1 Features

3.4.2 Instructions

3.4.3 Flags

3.4.4 Fractional and Integer Math

3.4.5 Simulator Exercise

3.5 Shifter

3.5.1 Features

3.5.2 Instructions

3.5.3 Flags

3.5.4 Simulator Exercise

4 Data Address Generators (DAG)

4.1.1 Load / Store Architecture

4.1.2 Features

4.1.3 Assembly Instructions

4.1.4 Data Moves

4.1.5 Circular Buffering

4.1.6 Bit Reversing

4.1.7 Simulator Exercise

- 5 ADSP-2191 Memory**
 - 5.1.1 Memory Configuration**
 - 5.1.2 Memory Map**
 - 5.1.3 Internal Memory**
 - 5.1.4 Example LDF Memory Section**

- 6 Program Sequencer**
 - 6.1.1 Features**
 - 6.1.2 Assembly Instructions**
 - 6.1.3 Conditional Sequencing**
 - 6.1.4 Instruction Pipe Line**
 - 6.1.5 Branching, Delayed Branching**
 - 6.1.6 Zero Overhead Looping**
 - 6.1.7 Interrupt Handling**
 - 6.1.8 Instruction Cache**

- 7 Program Development**
 - 7.1 IDDE and Projects**
 - 7.2 Assembler**
 - 7.2.1 Features and Overview**
 - 7.2.2 Assembler Expressions**
 - 7.2.3 Assembler Directives**
 - 7.2.4 Definition Files**
 - 7.3 Linker / Linker Description File (LDF)**
 - 7.3.1 Features and Overview**
 - 7.3.2 Example LDF**
 - 7.3.3 LDF Commands**
 - 7.4 Using the Simulator for Code Debug**
 - 7.4.1 Features and Overview**
 - 7.5 Assembly Code Programming Exercise**

- 8 Advanced Instructions**
 - 8.1 Multifunction Instructions**
 - 8.2 Divide Support**
 - 8.3 Multi-precision Math**
 - 8.4 Floating Point and Block Floating Point Support**

9 Flags, Timers, Direct Memory Access (DMA)**9.1 Features/Overview****9.2 I/O addressing****9.3 Programmable Flags****9.4 Timer****9.5 Direct Memory Access (DMA)****9.6 Example Auto Buffer DMA setup****10 Parallel Interfaces****10.1 External Memory Interface (EMI)****10.1.1 Features and Overview****10.2 External Bus Control****10.2.1 Operation****10.2.2 Configuration****10.2.3 Bus Request/Grant Flow****10.3 Host Port Interface (HIP)****10.3.1 Features and Overview****10.3.2 Configuration and DMA****10.3.3 Pin Description****11 Serial Interfaces****11.1 Serial Port****11.1.1 Features****11.1.2 Pin Descriptions****11.1.3 Modes of Operation****11.1.4 Configuration****11.2 Serial Peripheral Interface (SPI)****11.2.1 Features****11.2.2 Pin Description****11.3 UART****11.3.1 Overview****11.3.2 Features****11.3.3 Configuration**

- 12 Reset/Bootloading**
 - 12.1 Resetting the Processor**
 - 12.2 Booting Modes**
 - 12.3 Boot Kernel**
 - 12.4 Boot Stream Format**
 - 12.5 VisualDSP Loader Utility**
 - 12.6 Loader file formats**

- 13 System Design**
 - 13.1 Reference Material**
 - 13.2 Anomaly Sheet**
 - 13.3 Navigating the Datasheet**
 - 13.4 Clocking the ADSP-2191**
 - 13.5 Configuring the PLL**
 - 13.6 Powering the ADSP-2191**
 - 13.7 Power Down Modes**
 - 13.8 Power Consumption**
 - 13.9 JTAG Overview**
 - 13.10 High Speed Design**
 - 13.11 Packaging**
 - 13.12 Part Numbering**

14 C/C++ Compiler

- 14.1 Features**
- 14.2 Configuration**
- 14.3 LDF Related issues**
- 14.4 Register Usage**
- 14.5 Run Time Environment**
 - 14.5.1 Stack and HEAP**
 - 14.5.2 Interrupt Handling**
 - 14.5.3 Assembly Code Interface**
- 14.6 Run-time Library Overview**
- 14.7 C++ capabilities**
- 14.8 C Code Optimization**

15 Hardware Tools

- 15.1 ADSP-2191 EZ-Kit**
- 15.2 Kit Installation**
- 15.3 Emulators**

16 Advanced VisualDSP++ Features

- 16.1 Features Overview**
 - 16.1.1 Analyzing your code**
 - 16.1.2 VDK**
 - 16.1.3 TCL**
 - 16.1.4 Active-X / COM**

17 ADSP-219x versus ADSP-218x key differences

- 17.1 Summary of Differences between ADSP-218x and ADSP-219x**