NEON Programming & Optimization

Course Description

NEON Programming & Optimization is a two days ARM official course.
The course goes into great depth and provides all necessary know-how to develop software for the NEON coprocessor in Cortex-A processors.

The course covers the coprocessor instruction set, compiler support, SIND programming for vectorization, micro-architecture of NEON, code examples to implement parallel algorithms, code optimizations and performance monitoring.

Learn how to take advantage of NEON to optimize common signal processing functions used in Filtering, Sample Rate Conversion, Audio and Video Codec applications.

At the end of the course the participant will receive a certificate from ARM.

Course Duration

2 days (3 days with labs)

Goals

1. Become familiar with ARMv7-A/v8-A architecturs
2. Become familiar with NEON instruction set
3. Become familiar with NEON micro-architecture
4. Write embedded software for NEON
5. Optimize code in compiler and linker
6. Monitor and analyze NEON performance
7. Implement algorithms with NEON

Target Audience

Software engineers that would like accelerating algorithms for platforms based on Cortex-A processors.

Prerequisites

- ARM Cortex-A architecture knowledge
- C and Assembler
- Experience in developing embedded systems

Course Material

- ARM official course book
- Labs handbook
- DS5 SDK

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

Agenda

Main Topics:

- ARM Architecture Overview
- NEON Introduction
- NEON Instruction Set
- Compiling for NEON
- NEON Micro-Architecture
- NEON Coding Examples
- Benchmarking and Performance Analysis

Day #1

- **ARM Architecture Overview**
  - ARM architectures and processors
  - Programmer's view
  - Extension instruction space
  - ARMv7-A vs ARMv8-A

- **NEON Introduction**
  - What is NEON?
  - NEON compared to v6 SIMD
  - When and why to use NEON
  - Programmer's model
    - NEON registers
    - Data sizes
    - Data types
    - Register and element size
    - Vectors and scalars
    - Specifying data types
    - Extended notation
    - Instruction "shape"
    - Long and narrow operations
    - Instruction "modifiers"
    - Floating point
    - NEON status registers

- **NEON software support**
  - How to use NEON

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il

- What is project Ne10?
- Automatic vectorizing
- Tuning C/C++ code for vectorizing
- Code examples
- Intrinsics

- **NEON Instruction Set**
  - Instruction set overview
  - Data processing instructions
    - Arithmetic
    - Shifts
    - Comparison and selection
    - Bitwise logical
    - Miscellaneous
  - Data move instructions
  - Memory access

Day #2

- **Compiling for NEON**
  - Introduction to compiler support for NEON
  - Writing NEON code using intrinsics
  - Automatic vectorization
  - Turning code for optimal results
    - Loop considerations
    - Pointers and arrays
    - Pointer aliasing
    - restrict
    - Function calls and inlining
  - NEON vectorization example

- **NEON micro-architecture**
  - NEON in Cortex processors
  - Processor pipelines
  - Alignment specifiers
  - ARMv7-A processors pipeline implementation
  - ARMv8-A processors pipeline implementation
  - Instruction timing

- **NEON Coding Examples**
  - Finite impulse response (FIR) filter
  - Matrix multiply
  - YUV to RGB color conversion

- **Benchmarking and Performance Analysis**
  - Motivation for benchmarking
  - Performance Monitoring Hardware: PMU
  - Cycle Accurate Trace: Trace Macrocells
  - Streamline Performance Analysis
  - NEON benchmarking example

P.O.B. 803 Kfar Saba 441080 ISRAEL | P +972-52-5816791 | F +972-77-4702742 | ContactUs@HandsOnTraining.co.il | HandsOnTraining.co.il