

SPRAL: Code Generation for DSP Transforms

Robert Dwyer, Ian M. J. Aberkane, James Aberkane, David Baker, Michael Biles,
Mark D. Brown, Robert Chang, Tony Choudhury, Ian Craig, Thomas Farnsworth, Tony Fries,
Robert G. Johnson, Richard Johnson

David Brown

Abstract—This paper describes the design and implementation of SPRAL, a code generator for digital signal processing (DSP) transforms. SPRAL is a C++ library that takes a high-level description of a DSP transform and generates optimized C code for execution on a target DSP. The high-level description is a C++ class hierarchy that represents the transform and its parameters. SPRAL uses a series of passes to generate the C code, including a pass to generate the C code for the transform, a pass to generate the C code for the parameters, and a pass to generate the C code for the control logic. The generated C code is then compiled and executed on the target DSP. SPRAL is designed to be portable and easy to use, and it is implemented in a way that allows it to be extended to support new DSPs and transforms.

Index Terms—Code generation, DSP transforms, C++ library, C code, digital signal processing, DSP, C++ class hierarchy, C code for the transform, C code for the parameters, C code for the control logic, C code for the target DSP, C code for the compilation, C code for the execution.

1. Introduction

As the demand for digital signal processing (DSP) applications increases, the need for efficient and portable code generation tools for DSP transforms becomes more acute. This paper describes the design and implementation of SPRAL, a code generator for DSP transforms. SPRAL is a C++ library that takes a high-level description of a DSP transform and generates optimized C code for execution on a target DSP. The high-level description is a C++ class hierarchy that represents the transform and its parameters. SPRAL uses a series of passes to generate the C code, including a pass to generate the C code for the transform, a pass to generate the C code for the parameters, and a pass to generate the C code for the control logic. The generated C code is then compiled and executed on the target DSP. SPRAL is designed to be portable and easy to use, and it is implemented in a way that allows it to be extended to support new DSPs and transforms.

¹Copyright © 2005 IEEE. All rights reserved. Manuscript received May 10, 2005.

As the demand for digital signal processing (DSP) applications increases, the need for efficient and portable code generation tools for DSP transforms becomes more acute. This paper describes the design and implementation of SPRAL, a code generator for DSP transforms. SPRAL is a C++ library that takes a high-level description of a DSP transform and generates optimized C code for execution on a target DSP. The high-level description is a C++ class hierarchy that represents the transform and its parameters. SPRAL uses a series of passes to generate the C code, including a pass to generate the C code for the transform, a pass to generate the C code for the parameters, and a pass to generate the C code for the control logic. The generated C code is then compiled and executed on the target DSP. SPRAL is designed to be portable and easy to use, and it is implemented in a way that allows it to be extended to support new DSPs and transforms.

As the demand for digital signal processing (DSP) applications increases, the need for efficient and portable code generation tools for DSP transforms becomes more acute. This paper describes the design and implementation of SPRAL, a code generator for DSP transforms. SPRAL is a C++ library that takes a high-level description of a DSP transform and generates optimized C code for execution on a target DSP. The high-level description is a C++ class hierarchy that represents the transform and its parameters. SPRAL uses a series of passes to generate the C code, including a pass to generate the C code for the transform, a pass to generate the C code for the parameters, and a pass to generate the C code for the control logic. The generated C code is then compiled and executed on the target DSP. SPRAL is designed to be portable and easy to use, and it is implemented in a way that allows it to be extended to support new DSPs and transforms.

As the demand for digital signal processing (DSP) applications increases, the need for efficient and portable code generation tools for DSP transforms becomes more acute. This paper describes the design and implementation of SPRAL, a code generator for DSP transforms. SPRAL is a C++ library that takes a high-level description of a DSP transform and generates optimized C code for execution on a target DSP. The high-level description is a C++ class hierarchy that represents the transform and its parameters. SPRAL uses a series of passes to generate the C code, including a pass to generate the C code for the transform, a pass to generate the C code for the parameters, and a pass to generate the C code for the control logic. The generated C code is then compiled and executed on the target DSP. SPRAL is designed to be portable and easy to use, and it is implemented in a way that allows it to be extended to support new DSPs and transforms.

The authors would like to thank the following people for their contributions: