Datasheet

VectorC Retargetable Compiler Engine

codeplay
www.codeplay.com

Codeplay's VectorC™is a highly customizable compiler platform for building highly optimizing C and C++ compilers for new processor designs.

VectorC has proven to be an optimal compiler solution for mobile media processors and special purpose accelerators where other compilers such as GCC won't provide the best performance for the features of the processor.

C/C++ Support

The VectorC engine supports C, C++, and is also one of the first commercial compilers to include support for the most useful C++0x features such as 'auto' and delegating constructors.
VectorC can compile source code with multiple proprietary source extensions. It has built-in support for:

- Both Microsoft and GCC inline assembly formats
- Most Microsoft and GNU C/C++ language extensions
- HLSL and OpenCL language extensions can be used inside C/C++ code
- AltiVec extensions can be used inside C/C++ code
- C++ ABI compatible with Visual Studio and GCC

VectorC can easily be configured to support new language extensions, back-ends, and optimizations.

Optimizations

VectorC performs all the standard non-vector optimizations expected of a high-performance compiler, for example loop unrolling, software pipelining, and branch elimination/substitution using predication and conditional moves.

VectorC can also optimize inline assembly. Inline assembly is optimized and scheduled together with instructions generated from C/C++ code.

Supported Processors

- MIPS
- PowerPC
- x86 (SIMD)
- Cell BE SPU
- Ageia PhysX
- PS2VU

Vectorization

VectorC was designed from the outset to exploit vectorization. VectorC can produce fast SIMD vector code from scalar or vector source code on SIMD architectures.

VectorC is especially suited to GPU-like processors.

Special Hardware Support Features

- Efficient code generation for architectures supporting non-byte addressing.
- VLIW support: the scheduler can output more than one instruction per cycle.
- Flexible scheduling, supports stalling and non-stalling architectures, branch delay slot filling
- Support for compiling data and functions across multiple memory spaces

C++ to C Translation

VectorC is capable of generating C code from full C++ using its OutputC backend. Highly optimized C code can be generated from C/C++ code.

OutputC offers a massive advantage when rapid prototyping onto a custom accelerator processor is required. VectorC can compile C++ code to any architecture which has an existing C compiler. The generated C source is fully ANSI C compatible and can be easily configured to use extensions (for example intrinsics) provided by the tool chain for the targeted processor.

Custom Calling Conventions

User-definable calling conventions allow optimal register usage across function calls and to interface with existing assembly code.

Useful Advice Messages

VectorC implements numerous command line options to enforce good coding practices and assist in debugging.

The compiler generates warnings on unsafe pointer casts (casting pointers to structs that are defined later in the translation unit).

In the event of a pointer cast that loses alignment information VectorC attempts to preserve the correct alignment, and issue a warning and advice messages on those casts.

On failed lookups the compiler attempts to suggest alternative lookups to help finding names in complex code.

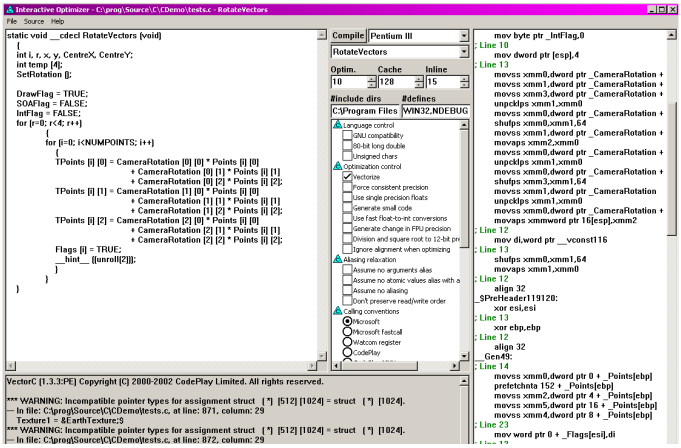
Configurable Feature Enforcement

Command line options are available to instruct the compiler to ban potentially dangerous language features, for example C-style casts in C++.

Interactive Optimizer

The Interactive Optimizer is a graphical front-end for PC-based VectorC compilers provides a side-by-side view of source code and corresponding assembly.

The VectorC Interactive Optimizer provides valuable optimization advice and lets programmers configure different optimizations settings in order to tweak the best performance out of their code. This is especially useful when writing optimal code for new and highly specialized processor architectures.



Tool Chain Integration

VectorC compilers integrate with the GCC tool chain (GCC 4, legacy support for GCC 2.9 is also provided if required). Support for most GCC command line options is provided.

PC-based compilers integrate with Visual Studio. VectorC supports Visual Studio command line options, and generates code compatible with that produced by cl.exe.

Eclipse CDT is also fully supported.

Early Tool Chains

Our VectorC compilers can output assembly ready-code. VectorC can layout global and other data statically in memory, which is ideal for new processors where only an assembler is provided. This also adds optimization opportunities for memory accesses.

Supported Debug Formats

- CodeView (Microsoft Debug Info)
- Dwarf 2.0/3.0
- Stabs

Supported Output Formats

- Assembly
- COFF
- ELF
- C source code

Complete System

VectorC has been developed from the ground up over a period of 10 years and has no third-party product dependencies.

With VectorC, Codeplay can create a complete compiler solution for your processor. VectorC can be rapidly reconfigured enabling us to deliver a prototype C/C++ compiler to customers within a few weeks.