

CoSy compiler development system

Build your own compilers with CoSy!

CoSy FACTS

- Professional
- State-of-the-art
- Modular design
- Open infrastructure
- Generated components
- Complete
- >250 Plug & play engines
- User extensible
- Fully documented
- Best-in-class compiler track record
- No risk
- Reduced time-to-performance
- Reduced total effort
- Reduced total cost of ownership (TCO)

CoSy is the professional, easy-targetable and highly flexible compiler development system from ACE Associated Compiler Experts, which has been successfully deployed by over 100 industrial customers and partners world-wide, in creating high-quality, high-performance compilers for a broad spectrum of targets, ranging from a 8-bit microcontrollers to CISC, RISC, DSP and 256-bit VLIW processor architectures. Based upon its highly modular design surrounding a generic, extensible intermediate representation (IR) and extensive use of generators, the CoSy environment enables construction of production-quality performance compilers in a highly efficient manner, reducing time-to-market, time-to-performance, as well as development and maintenance costs.

CoSy's dedicated Embedded C/DSP-C language extensions allow the DSP compiler engineer to address the specific characteristics of the target architecture and generate optimal code. In addition, CoSy's configurability and retargetability make it a particularly effective environment for exploration of compiler effects on possible architecture variations, thus enabling true HW/SW co-design.

Why CoSy?

The success of a new processor depends on more than it having the ultimate hardware design. In order for a chip to become truly successful and applied in multiple projects, the associated programming tools need to be of matching quality and performance.

Many in-house compiler developments have failed on the sheer magnitude of such a project. Even more compiler developments that have been based on open-source frameworks, simply could not benefit from the target architecture's specific features, while running into dozens of man-years of development effort.

There is, however, no need to follow that route. With the CoSy compiler development system, one can be absolutely sure that best results can be obtained in a timely and cost-effective manner. CoSy users repeatedly demonstrate the value of this ultimate compiler development environment by constructing best-in-class compilers for regular type, as well as highly specialized processor architectures, with small teams in the range of 1-2 compiler engineers.

CoSy puts the compiler developer in full control, with no restrictions as to extending the IR, adapting standard functionality, or even adding custom features, as necessary to provide optimal support for the specific target architecture. Unburdened with development and maintenance of the compiler framework, compiler engineers working with CoSy can focus their attention on the real task at hand: creating a high-performance compiler that truly deploys the specific features of the target processor. CoSy provides a 200 man-years head start to your compiler development and the advantages of 10-20 man-years of enhancements coming with each release update as you go.

Who use CoSy?

- **Semiconductor** **companies**
for fast and cost-effective development of production-quality compilers for new processor architectures
- **Development tool** **and** **EDA** **companies**
for development of commercial software development tools
- **Architecture** **R&D** **groups**
for performance testing and architecture roadmap exploration
- **Academia** **and** **Research** **Institutions**
for research on compilation techniques and processor architecture design

Packaging & Availability

- **CoSy Base package**
 - ISO/IEC 9899 C front-end
 - Generators for IR, compiler supervisor and back-end
 - Base IR and functor definitions
 - IR management engines for consistency checking, dumping, debugging and visualization
 - A wide range of analysis, optimization, lowering, checking and allocation engines
 - Data flow analysis

- Various prototype target engines
- Code generator generator plus a prototype code generator description
- Code generator feedback mechanism
- Multiple scheduler strategies
- Open, modular register allocator
- Quick start code generator template
- Flexible calling convention support
- Dwarf 1 and 2 object debug format generators
- Target runtime library generator plus various target runtime libraries, incl. libc
- Host independent target support libraries for integer and floating-point arithmetic
- Compiler Trainer, incremental test suite
- The CoSy engineering framework, including configuration management system
- **CoSy Advanced optimizations package**
 - Algebraic optimizations
 - Hardware (zero overhead) loop analysis and optimization engines
 - Advanced loop optimizations, including loop hoisting and loop scalar replacement
 - Use estimate analysis
 - Advanced scheduler support in back-end generator
 - Software pipelining engines
 - Support for advanced inline assembly, including user-defined intrinsic functions
 - Path profiling instrumentation & annotation tools and libraries
 - Static profiling
 - Tuple support
 - Value range analysis
- **CoSy Target optimizations package**
 - Generic predicated execution support
 - Hyperblock scheduling
 - Life range splitting
 - Local spill minimization
 - Out-of-order scheduling
 - Delay slot filling
 - Base binding
- **CoSy Embedded C language package**
 - ISO/IEC 9899, TR18037 Embedded C front-end
 - Target emulation package, including emulation compiler engines and libraries
 - Compiler support library for host independent target fixed-point arithmetic
 - Embedded C specific target runtime libraries
 - Embedded C ready prepared compiler known functions (intrinsics)
 - Embedded C emulation compiler configuration
- **CoSy C++ language package**

- Edison Design Group C++ front-end (ISO/IEC 14882)
- EDG C++ to IR transformation engine
- Prototype compiler driver
- Basic runtime library and include files
- C++ specific engines
- **CoSy Compiler Host packages**
 - Compiler build configuration tools
 - Compiler component objects
 - Available for:
 - x86/Windows
 - x86/Linux
 - Itanium2/Linux
 - Sparc/Solaris

Supported host platforms

The CoSy compiler development system is available in binary and full source versions, for development systems running Solaris and Linux, to generate compilers and cross compilers for a variety of architectures and platforms including Windows.