COSMIC C Cross Compiler for Motorola 68300 and 68000 Families

COSMIC's C cross compiler, ***cxcpu32*** *for the Motorola CPU32, CPU32+ and 68000 families of microcontrollers/microprocessors, incorporate over twenty years of innovative design and development effort. In the field since 1980,* ***cxcpu32*** *is field-tested, reliable, and incorporates many features* ***to help ensure your embedded CPU32/68K design meets and exceeds performance specifications.***

The **C Compiler** package for Windows includes: COSMIC integrated development environment (IDEA), optimizing C cross compiler, macro assembler, linker, librarian, object inspector, hex file generator, object format converters, debugging support utilities, run-time libraries and a compiler command driver. The PC compiler package runs under Windows 95/98/ME/NT4/2000 and XP.

Key Features

Supports All CPU32/CPU32+ Family Members and 68000 Processors

ANSI/ISO Implementation

Global and Processor-Specific Optimizations

Optimized Function Calls

Extensions to ANSI for Embedded Systems

C support for Interrupt Handlers

In-Line Assembly

Memory-Mapped I/O in C

Optional 16-bit Integers

Debug Fully Optimized Code

IEEE695 and ELF/DWARF Debug Support

Absolute C and Assembly Listings

Works With All Popular CPU32 In-Circuit Emulators

First Year of Support Service Included

No Charge Upgrades

Microcontroller-Specific Design

cxcpu32 is designed specifically for the Motorola CPU32 family of microcontrollers; it uses a **special code generator and optimizer targeted specifically for the CPU32 and 68K families**. You also get header file support for CPU32 on-chip peripherals, so you can access their memory mapped objects by name.

ANSI C

This implementation conforms with the **ANSI and ISO Standard C** specifications. Standard C is upward compatible with ANSI C but provides additional reliability features and aids for the embedded systems developer.

C Runtime Support

C runtime support consists of a subset of the standard ANSI library, and is provided in pre-built binary and C source form with the binary package. The basic library set includes the support functions required by a typical embedded system application. Support includes:

- Character handling
- Mathematical functions
- Non-local jumps
- Formatted serial input/output
- String handling
- Memory management

The package provides both an **integer-only library** as well as the standard **single and double precision floating point libraries**. This allows you to select the smaller and faster integer-only functions, if your application doesn't require floating point support.

Optimizations

The COSMIC compiler for the Motorola CPU32 family includes global and processor specific optimizations which lead to more compact, faster programs:

- ***cxcpu32*** supports **global optimizations** which allow it to optimize whole C functions as well as C statements,
- Register allocator automatically allocates function locals to machine registers based on usage counts and variable lifetime analysis,
- **Peephole optimizer** further optimizes ***cxcpu32***'s output by replacing inefficient code sequences with optimal code sequences for CPU32 processors,
- Branch shortening logic chooses the smallest possible jump/branch instructions,
- Jump-to-jump and jump-over-jump instructions are eliminated,
- Fast function calls: functions marked with the **@fast** keyword cause the compiler to use multiple **move** instructions rather than **movem** to handle arguments, which is faster,
- Functions defined with the **@regsafe** keyword do not save and restore the processor registers. This is useful for function calls from interrupt service routines or from a real-time kernel,
- Integer and floating point constant expressions are folded at compile time,
- The switch statement is optimized to produce various combinations of jump tables for closely-spaced cases, scan tables for a small group of loosely-spaced cases, or sorted tables for a binary search,
- All integer multiplication, including 32-bit integers, are performed in-line, with optimizations for operands with smaller types,
- Multiplication by powers of two are performed as faster shift instructions,
- Multiplication of constants are optimized to faster shift/add sequences,
- Assignment operations are optimized,
- Unreachable code is eliminated,
- Unused static variable elimination,
- Stack scratch cells are used to save extra stack operations,
- Link/unlink instructions are eliminated where possible in function entry/exit code,
- Compiler option to declare integers 16-bit rather than 32-bit. For CPU32 family members, access to 16-bit integers is faster than 32-bit integers,
- Compiler option to force strict single precision floating point arithmetic

Extensions to ANSI C

The COSMIC C compiler includes several extensions to the ANSI standard designed specifically for embedded systems programmers. Optional Extensions to the ANSI Standard include:

- In-line assembly, via **_asm()**, to insert assembly instructions directly in your C code to avoid the overhead of calling assembly language subroutines,
- C level support for **interrupt functions** saves volatile registers for handling exceptions and interrupts,
- Char and int-sized bit-fields with the ability to select bit numbering form right-to-left or left-to-right. Char and int-sized bitfields can be mixed in the same source file,
- Absolute Addressing allows you to define functions and data objects in C, to directly reference absolute memory addresses. This feature is useful for handling interrupts and defining memory mapped I/O,
- **@fast** and **@regsafe** for special function calls as described in the previous section,

Additional Compiler Features

- Full C source-level debugging information for use with COSMIC's range of **ZAP** debuggers or debuggers provided by ICE/RTOS vendors,
- All code generated by the compiler using its default options will execute on any member of the CPU32 family (6833x),
- Code is generated as a symbolic assembly language file which is fed to the COSMIC CPU32 assembler,
- Function code, switch tables, and const data can be located separately in ROM. Literal data such as strings and constants are generated into the data section, but can be optionally located in the text section to save RAM space,
- Initialized static data is located separate from uninitialized data or data initialized to zero allowing a fast startup and initialization sequence,
- All function code is reentrant, never self modifying, including structure assignment and function calls, so it can be shared and placed in ROM.
- Floating point numbers are represented in the IEEE Floating Point Standard formats. The compiler supports both software emulation of floating point and the MC68881 coprocessor instructions,
- Single precision software emulation of floating point is also available as an option. The software floating point routines are extremely fast, resulting in good performance, even for systems without MC68881 support,
- In-line generation of MC68881 instructions for add, subtract, multiply and divide operations,

- The compiler creates all its tables dynamically on the heap, allowing large source files to be compiled,
- Identifiers are unique to 127 characters to allow the creation of meaningful identifier names,
- Unused variables can be flagged with an error message,
- Common string manipulation routines are implemented in assembly language for fast execution.

Debugging Utilities

cxcpu32 includes a number of utilities which provide listings for all debug and map file information to allow both host and target C level cross debugging.

Lines utility
Extracts C source line number information from either an object or linked file. This utility prints executable C source line numbers as symbols along with the absolute address of the start of each line.

Clist utility
Prints out the contents of C source files, with line number information and the absolute addresses of the start of each source line after linking.

Prdbg utility
Extracts and prints information on the names, types, storage class, and address (absolute or offset) of program static data and the arguments and automatics belonging to program functions.

CPU32 Assembler

The assembler includes facilities for writing hand coded assembly language. The assembler accepts all of Motorola's defined CPU32 instruction set and UNIX style directives and pseudo-ops. It supports macros, defines, includes, branch optimization, expressions, relocatable arithmetic, cross references and a variety of output files: listing, relocatable object, symbol table and errors. Each line of the listing includes the source line number, location counter contents, instruction code and the actual source code.

Linker

The linker combines relocatable object files created by the assembler, selectively loading from libraries of object files made with the librarian, to create an executable format file. The linker features:

- Flexible user-control over the linking process and selective placement of program sections,
- Multi-segment images can be constructed, with user control over the address for each text, data, and BSS section. The specified addresses can cover the full logical address space of the target processor with up to 255 separate sections. This feature is useful for creating an image which resides in a target memory configuration consisting of scattered areas of ROM and RAM,
- Generation of memory map information to assist debugging,
- Incremental linking to allow programs to be built in stages,
- Symbols can be defined, or aliased, from the command line.

Librarian

The librarian is a development aid which allows you to collect related files into one named library file, for more convenient storage. It provides the functions necessary to build and maintain object module libraries. The linker loads from a library only those modules needed to satisfy outstanding references.

Absolute Hex File Generator

The hex file generator translates executable images produced by the linker to one of several hexadecimal interchange formats for use with most common In-Circuit Emulators and PROM programmers.

- Standard **Intel hex** format,
- **Motorola S-record** and S2 record format,
- Tektronix standard and extended hex format,
- Rebiasing of text and data section load addresses.

Absolute C and Assembly Listings

Paginated listings can be produced to assist program understanding. Listings can include original C source code with interspersed assembly code and absolute object code. Optionally, you can include compiler errors and optimizations.

Third Party Debugging Support

You can use ***cxcpu32*** or ***cacpu32*** with **ZAP/SIM, ZAP/BDM and ZAP/ICE** or with the debuggers provided by most popular In-Circuit Emulator manufacturers including AMC, EST, HMI, HP, Nohau and logic analyzers from HP and Tektronix. ***cxcpu32*** also supports several additional debugging formats including **IEEE-695, ELF/DWARF 2.0**, and P&E map file format.

Packaging

All compiler packages are provided on standard CD-ROM with complete on-line user documentation in Adobe PDF format.

The **C Compiler** package for Windows includes: An integrated development environment (IDEA), optimizing C cross compiler, macro assembler, linker, librarian, object inspector, hex file generator, object format converters, debugging support utilities, run-time libraries and a compiler command driver. The PC compiler package runs under Windows 95/98/ME and Windows NT4/2000/XP.

The Windows version also includes integration files for Starbase's popular **CodeWright®** Windows® code and project editor and GNU make utility.
The **C Compiler** package for UNIX includes: *An* optimizing C cross compiler, macro assembler, linker, librarian, object inspector, hex file generator, object format converters, debugging support utilities, run-time libraries and a compiler command driver. The UNIX compiler package is available for SUN Solaris and HP-UX.

Support Services

All COSMIC Software products come with the first year of support included in the price. You will receive a courteous and prompt service from our technical support staff and **you retain control of the severity of the problem** i.e. if it's a problem that is critical to your project we guarantee you a response time of one to three business days depending on the severity of the problem. Service is provided during normal business hours E.S.T. via email, fax or telephone and is unlimited while you have a valid annual support agreement. New releases of the software are provided free of charge to support customers.

Ordering Information

cxcpu32 package product codes are as follows:

Host System	*Product Code*
PC MS Windows Windows 95/98/ME/NT4/2000/XP-	CWS332
SUN SPARC(SunOS/Solaris)	CSS332
HP9000(HP-UX)	CHP332

Orders are shipped within one week of receipt of hard copy purchase order. Call our sales department for license fees and multiple copy discounts.

Other COSMIC Software Products

COSMIC Software products focus on Motorola 8,16 and 32-bit microcontrollers. C compiler/debugger support is available for **68HC05, 68HC08, 6809, 68HC11, 68HC12, 68HC16, 683XX and 680X0**. For more information on the ZAP C and assembler source-level debugger, ask for the ZAP Product Description and demo disk.

For Sales Information please contact:

COSMIC Software USA
COSMIC Software, Inc.
400 West Cummings Park, Suite 6000
Woburn, MA 01801-6512 USA
Phone: (781) 932-2556 Fax: (781) 932-2557
Email: sales@cosmic-us.com
web: www.cosmic-software.com

COSMIC Software France
33 Rue Le Corbusier, Europarc Creteil
94035 Creteil Cedex France
Phone: + 33 4399 5390 Fax: + 33 4399 1483
Email: sales@cosmic.cosmic.fr
web: www.cosmic.fr

COSMIC Software UK
Oakwood House
Wield Road, Medstead
Alton, Hampshire
GU34 5NJ, U.K.
Phone: +44 (0)1420 563498 Fax: +44 (0)1420 561946
Email: *sales@cosmic.co.uk*

COSMIC Software GmbH
Rohrackerstr 68 D-70329 Stuttgart Germany
Tel.+ 49 (0)711 4204062 Fax + 49 (0)711 4204068
Email: sales@cosmic-software.de
web: www.cosmic-software.de

COSMIC Software
Supporting Embedded Innovation since 1983

Digital DNA from Motorola DEVELOPER