SDCC

Small Device C Compiler

Philipp Klaus Krause

2025-02-02

Table of Contents

Table of Contents

What is SDCC?

- Standard C compiler: ISO C90, ISO C99, ISO C11, ISO C23, ISO C2Y
- Freestanding implementation or part of a hosted implementation
- Supporting tools (assembler, linker, simulator, ...)
- Works on many host systems
- Targets various 8-bit architectures, has some unusual optimizations that make sense for these targets
- Latest release: 4.5.0 (2025-01-28)
- User base: embedded developers and retrocomputing/-gaming enthusiasts
- Also used in downstream projects (z88dk, gbdk, devkitSMS)

Ports

- MCS-51, DS390, STM8, f8, HC08, S08, PDK13, PDK14, PDK15 (PIC14, PIC16)
- MOS 6502, WDC 65C02
- Z80, Z80N, Z180, eZ80, TLCS-90, SM83, Rabbits, R800

Project

- Bug and feature request trackers at SourceForge
- Mailing lists sdcc-user and sdcc-devel
- svn repository (and git mirror) at SourceForge
- Wiki
- Compile farm for nightly regression testing
- Mostly volunteer work, but has received some external support (DFG, Prototype Fund, NLnet, hardware vendors)

Table of Contents

Standard compliance

- struct / union parameters and return types
- Most of ISO C23
- Bits of C2Y

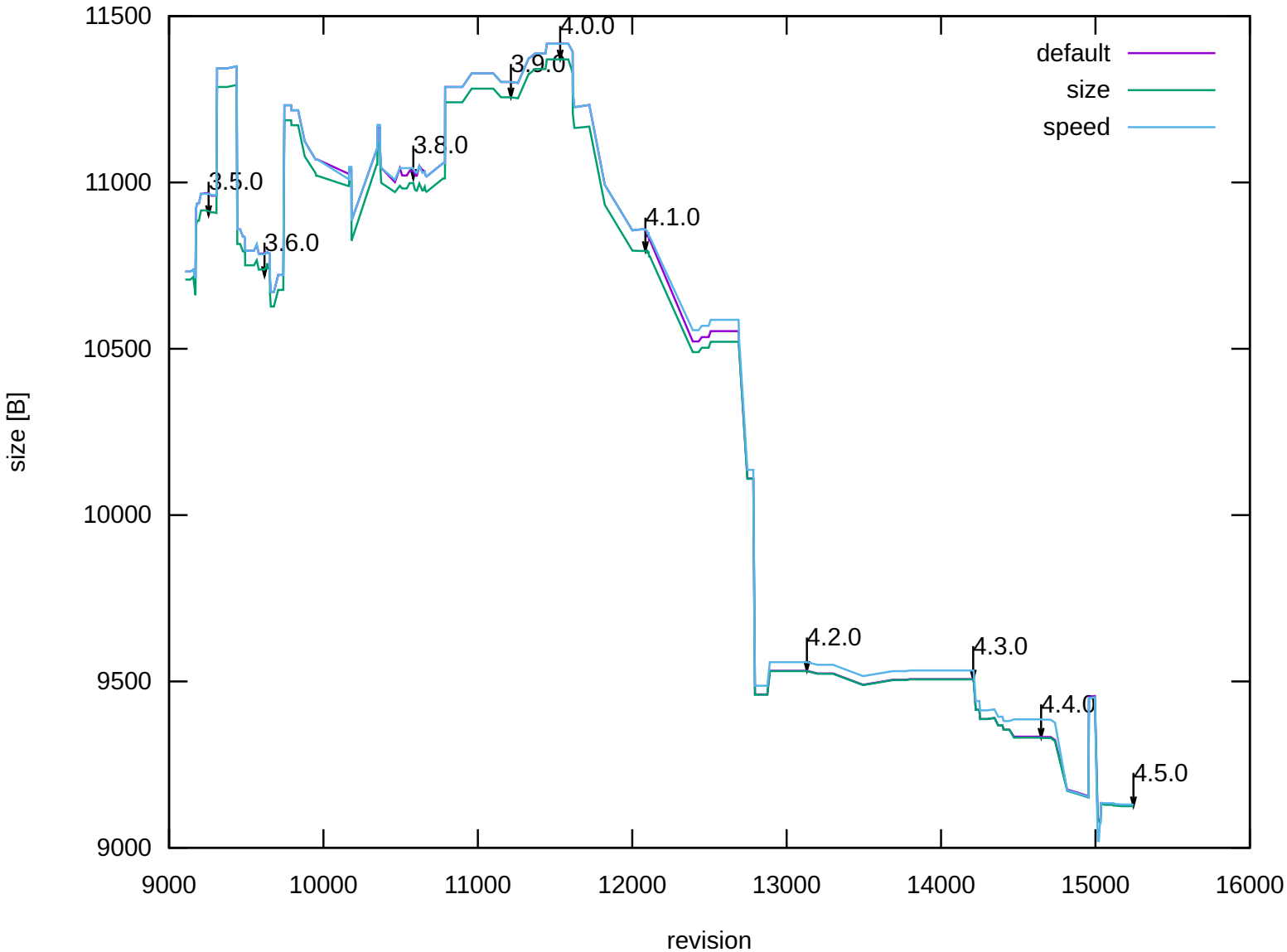
New ports

- Port for WDC 675C02
- Port for R800 (also useful for Z280)

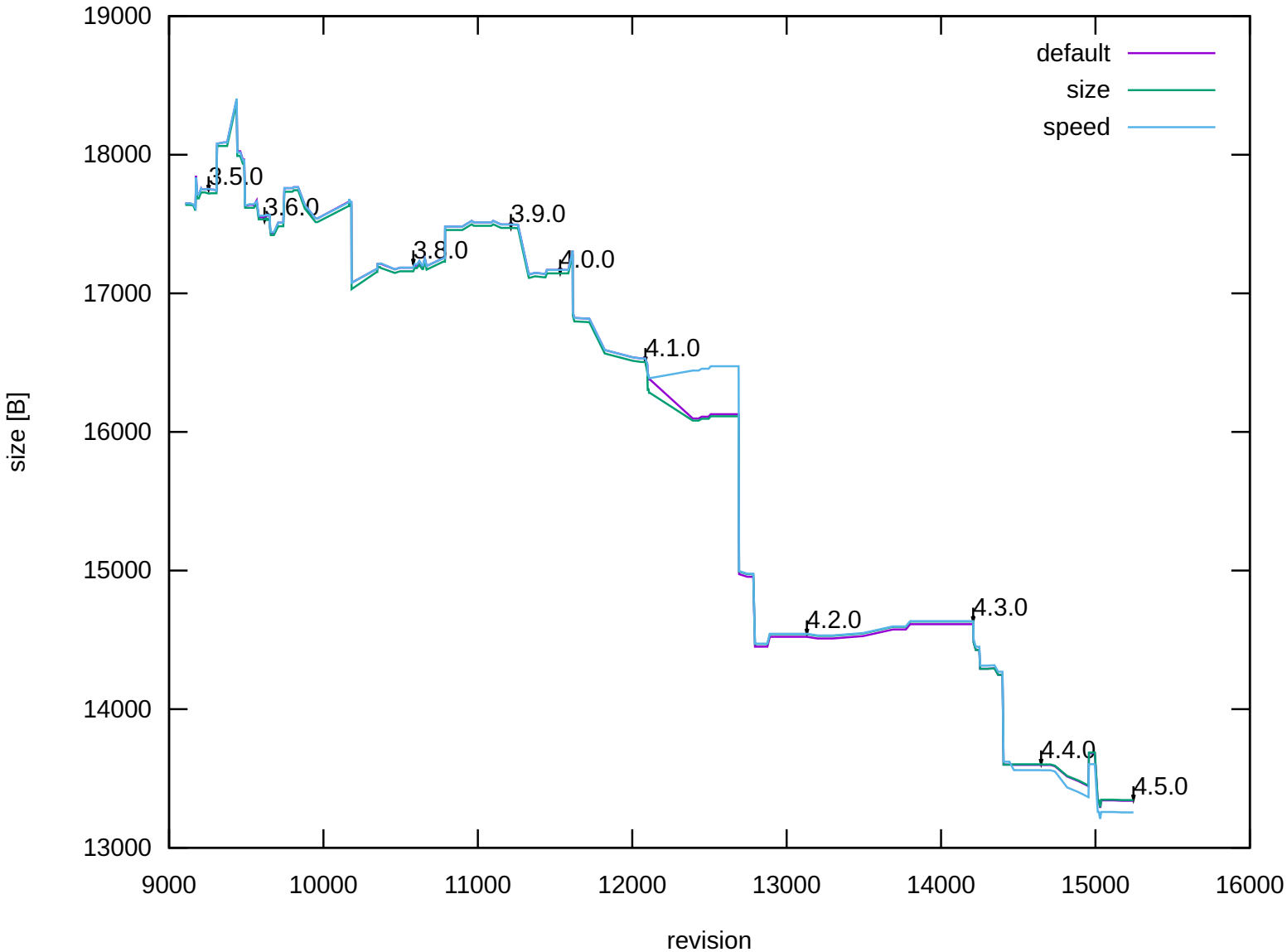
Optimizations

- Bit rotation optimizations
- Generalized constant propagation
- More efficient default calling conventions

Z80 Dhrystone code size



Z80 Whetstone code size



Z80 Coremark code size

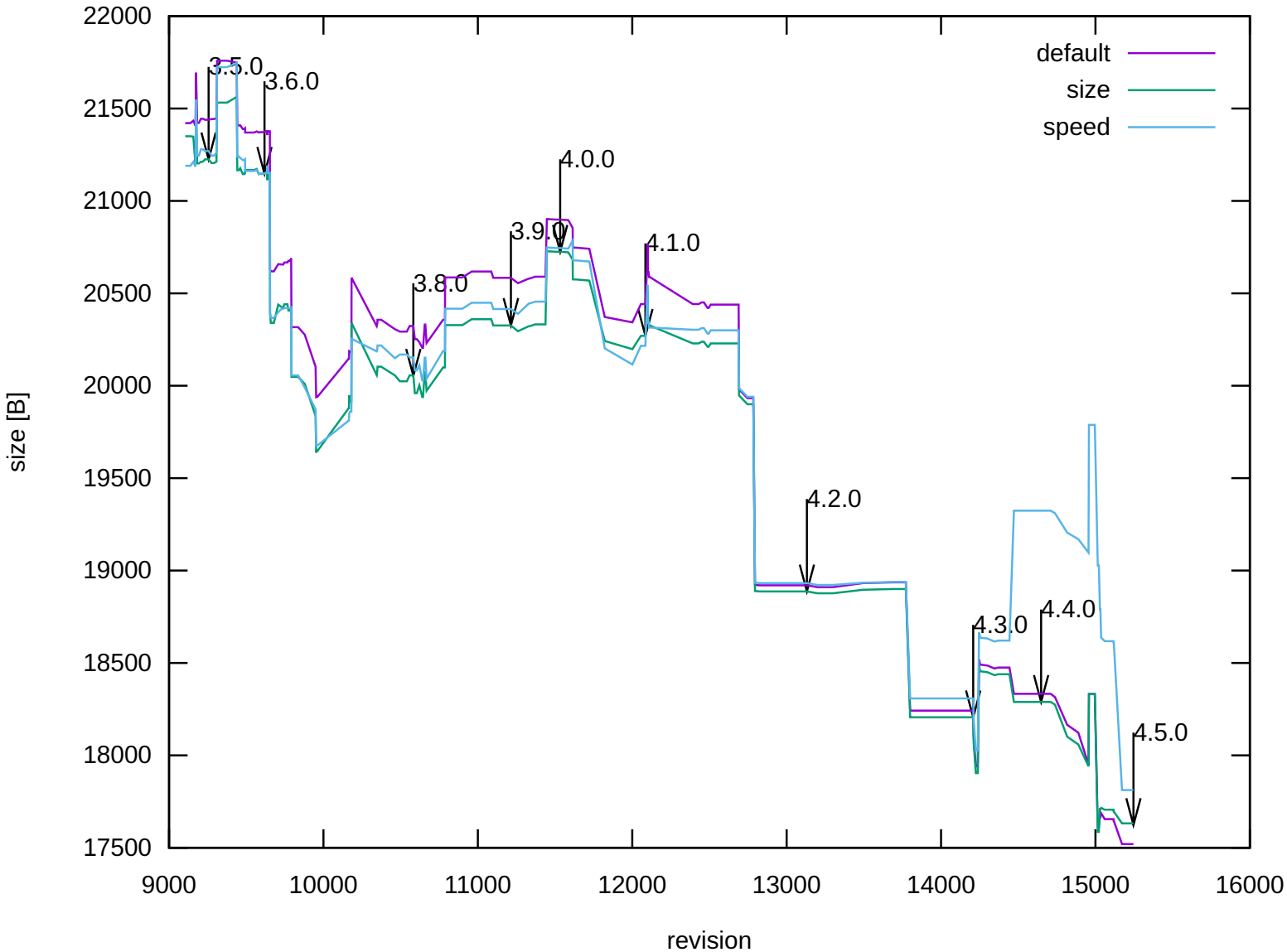


Table of Contents

Standard compliance

- Complete C99-C23 support: compound literals, attributes, atomics, double, long double
- More C2Y support

Ports

- New ports for Rabbit 4000, 5000, 6000
- __far pointers for Rabbit and eZ80 ports
- Pointers to I/O space for Z80-related targets

Optimizations

- Link-time elimination of unused objects and functions
- Efficient allocation of spilt local variables into non-stack memory
- Better heuristics for handling global operands for Z80-related targets