SDCC

Small Device C Compiler

Philipp Klaus Krause

2025-04-12

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

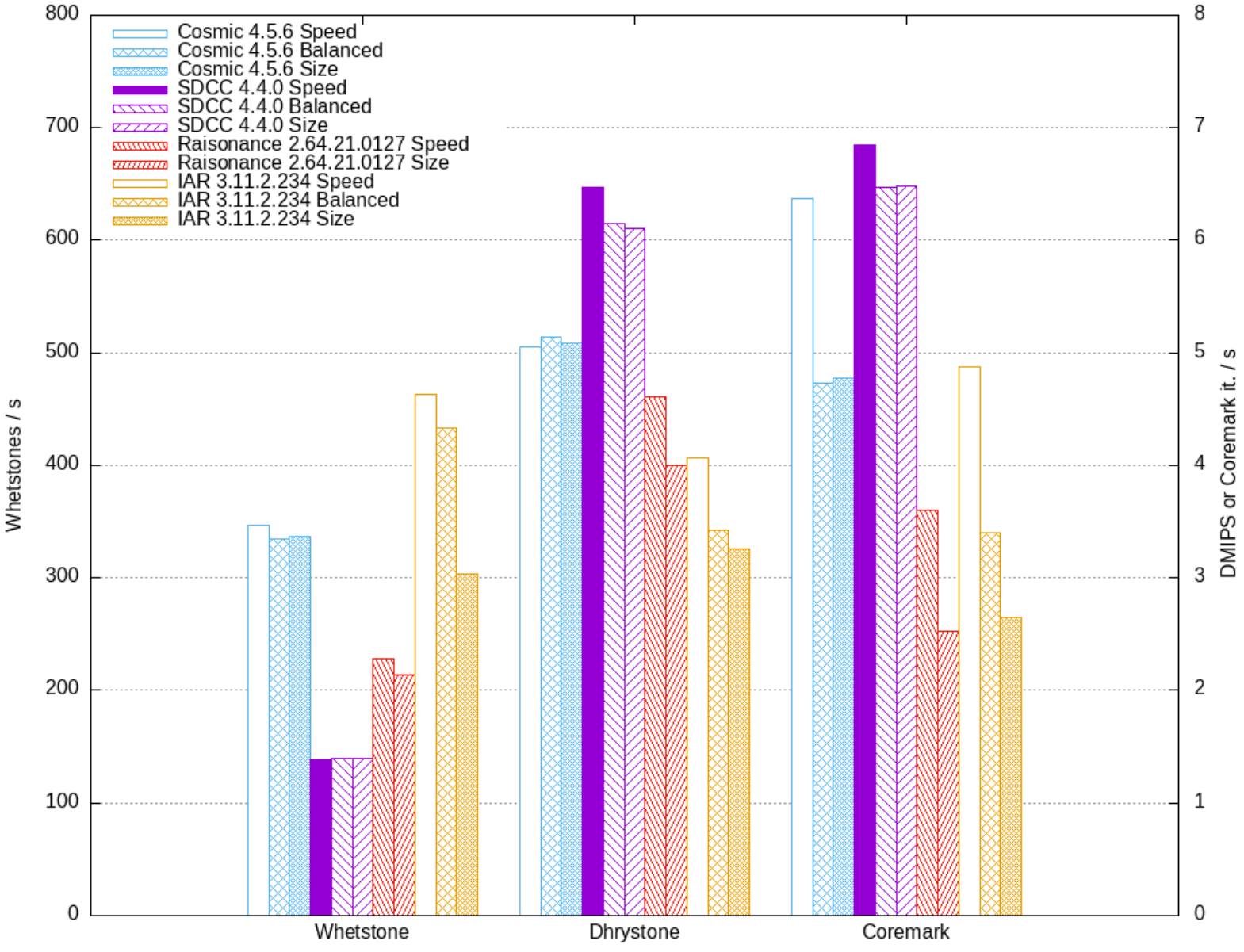
Optimal Register Allocation in Polynomial Time

- Register allocator based on graph-structure theory
- Optimal register allocation in polynomial time
- Flexible through use of cost function
- Provides substantial improvements in code quality
- Slow compilation for targets with many registers
- Compilation speed / code quality trade-off:
  –max-allocs-per-node

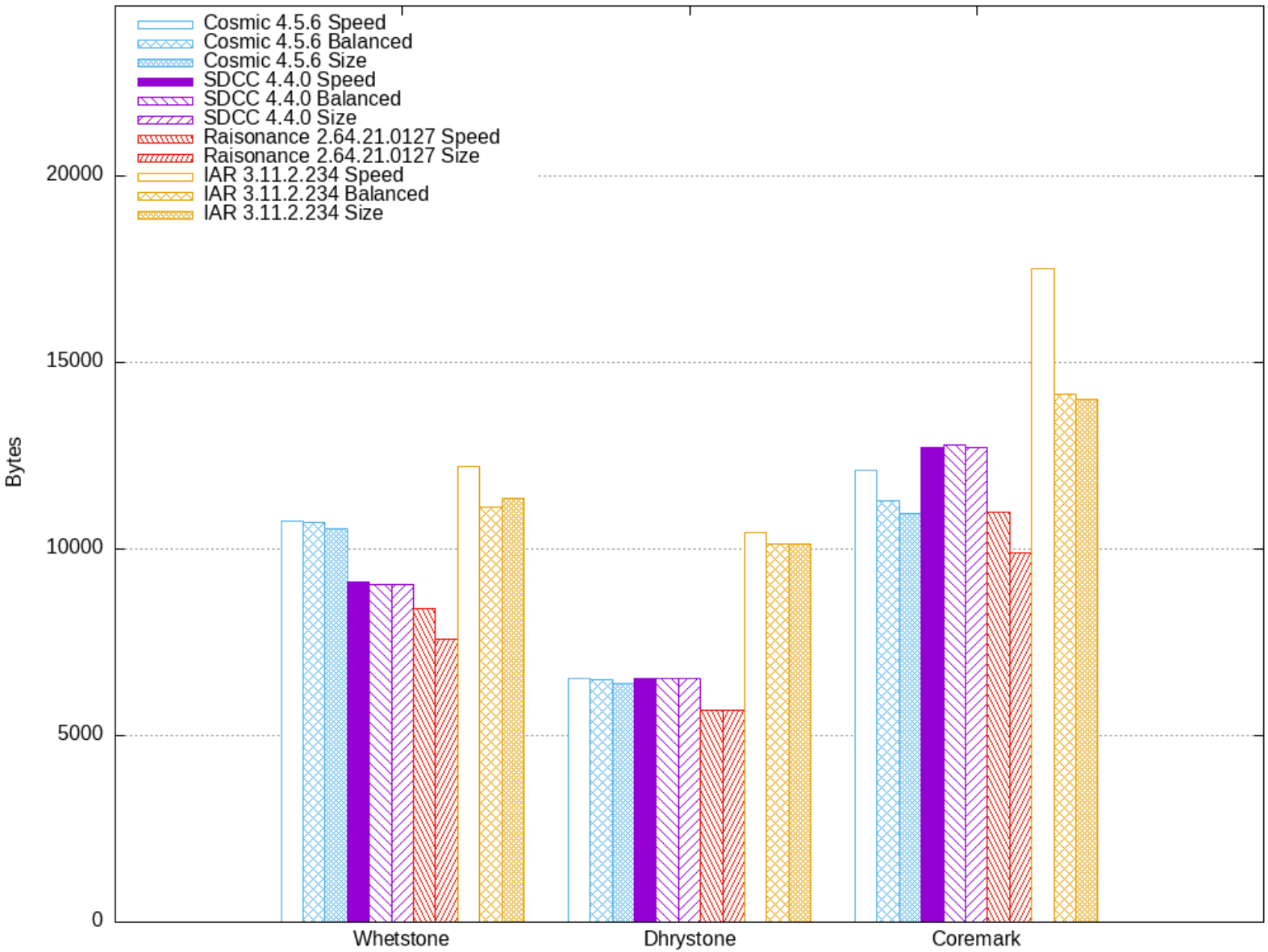
Bytewise Register Allocation and Spilling

- Decide on the storage of variables bytewise
- Decide for each individual byte in a variable whether to store it in memory or a register
- Consider any byte of any register as a possible storage location

SDCC vs. non-free compilers: STM8 Benchmark scores



SDCC vs. non-free compilers: STM8 Code size



Regression testing

- Regression testing of nightly snapshots
- ≈ 32000 tests (thrice as many as 2020) compiled and executed on simulators
- Tests mostly from fixed bugs and from GCC
- Targets architectures: MCS-51, DS390, Z80, Z180, eZ80, Rabbit 2000, Rabbit 2000A, Rabbit 3000A, SM83, TLCS-90, HC08, S08, STM8, PDK14, PDK15.
- Host OS: GNU/Linux, macOS, "Windows" (cross-compiled on GNU/Linux, tested via wine)
- Host architectures: x86, amd64, ppc64, aarch64

TODO

- SDCC needs developers
- Fix SDCC bugs
- Improve debug info, standard compliance, etc
- Improve IDE integration
- Improve hardware interface tools (Easy PDK programmer, free firmware for ST-Link, OpenRabbit, etc)
- Optimizations (in particular link-time elimination of unused objects and functions, and better allocation of spilt variables not on stack)
- Improve MCS-51 support - strategic importance for free software
- Improve Padauk support (PDk13, PDK16)
- Improve ISO C standard compliance (2nd talk today)
- Improve support for high-end 8-bit architectures: Rabbits, eZ80 (3rd talk today)