

# Game development for the ColecoVision and Sega 8-bit systems

Developing for Z80-based video game systems using modern  
free tools

Philipp Klaus Krause

February 4, 2018

# Table of Contents

1 Consoles

2 SDCC

3 Library

4 Tools

5 Summary

# Table of Contents

1 Consoles

2 SDCC

3 Library

4 Tools

5 Summary

- 3.5 Mhz Z80
- 1 KB RAM
- TI TMS99xx graphics
- 16 KB VRAM
- TI SN76489A sound
- Readonly cartridges up to 32 KB

# ColecoVision Peripherals

- Controllers with joystick, 2-4 fire buttons, keypad
- Optional roller controller (trackball)
- Expansion module #1: Atari adapter
- Expansion module #2: Steering wheel
- Expansion module #3: Adam computer
- Expansion module #3: Super Game Module

- Similar hardware to the ColecoVision
- But better cartridge port pinout

## Mark III / Master System

- 3.5 Mhz Z80
- 8 KB RAM
- Custom graphics
- 16 KB VRAM
- TI SN76489A sound

# Table of Contents

1 Consoles

**2 SDCC**

3 Library

4 Tools

5 Summary



# What is SDCC?

- Standard C compiler (ANSI C89, ISO C90, ISO C99, ISO C11)
- Freestanding implementation or part of a hosted implementation
- Supporting tools (assembler, linker, simulator, ...)
- Works on many host systems (GNU/Linux, Windows, Mac OS, Solaris, NetBSD, FreeBSD, OpenBSD, ...)
- Targets various 8-bit architectures (MCS51, DS80C390, Z80, Z180, Rabbit 2000, Rabbit 3000A, GBZ80, TLCS-90, HC08, S08, STM8, PIC)
- Has some unusual optimizations that make sense for these targets (in particular in register allocation)

# 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
- But slow for architectures with many registers
- Compilation speed / code quality trade-off:
  - max-allocs-per-node

# Regression testing

- Regression testing of nightly snapshots
- $\approx$  10000 tests compiled and executed on simulator
- Tests mostly from fixed bugs and from GCC
- Targets architectures: MCS-51, DS390, Z80, Z180, GBZ80, Rabbit 2000, Rabbit 3000A, TLCS-90, HC08, S08, STM8
- Host OS: GNU/Linux, Windows, MacOS
- Host architectures: i386, x86\_64, ppc, arm

- Uses LLVM C front- and backend to produce C code to be compiled with SDCC
- Code compiled with LLVM+SDCC can be mixed with C code compiled with SDCC
- Allows languages other than C
- Enables high-level optimizations
- Experimental, many issues remaining

- scc compiler based on Small-C
- zsdcc fork of SDCC.
- Emphasis on a large set of asm-written libraries for various Z80-based systems

# Table of Contents

1 Consoles

2 SDCC

**3 Library**

4 Tools

5 Summary

- Thin hardware abstraction layer
- Graphics, sound, input, ...
- Supports all ColecoVision peripherals
- Supports basics of Sega 8-bit systems

- Provides common functionality for games
- Graphics, music, compression, fixed-point math, ...



# Table of Contents

1 Consoles

2 SDCC

3 Library

**4 Tools**

5 Summary

# Tools

- Graphics
- Music
- Compression

# Table of Contents

1 Consoles

2 SDCC

3 Library

4 Tools

**5 Summary**

# Summary

- SDCC, a modern compiler targets the Z80
- ColecoVision and Sega 8-bit systems have similarities making it easy to write portable games
- This is supported by the libcv and libcvu libraries