An Open-Source SD-Card Host Controller for Linux-Capable RISC-V SoCs

Axel Vanoni[1], Philippe Sauter[1], Paul Scheffler[1], Anton Buchner[1], Micha Wehrli[1], Thomas Benz[1,2], Luca Benini[1,3]
[1]Integrated Systems Laboratory, ETH Zurich [2]lowRISC C.I.C., Cambridge, UK [3]DEI, University of Bologna

1 Motivation

- **Linux-capable open-source RISC-V SoCs** (Basilisk, Cheshire) are now silicon-proven.
- They need fast **non-volatile storage** to boot a Linux image, as root filesystem and for permanent user data.
- SPI and I²C limit throughput; **SD cards** are the de-facto standard embedded-Linux medium.

► ***An open SD host controller is essential for deployment of open RISC-V platforms.***

2 Controller Architecture

- Controlled via **SDHCI v1.0** compatible memory-mapped register interface.
- Independent **command and data channels**
- Data block **includes SRAM or standard cell memory** buffers to guarantee stall-resistant operation.
- Programmable divider derives the SD clock from host.
- SDIOs **6-signal I/O**: clock (CLK), command (CMD) and 4-bit data (DAT[3:0), no additional constraints.

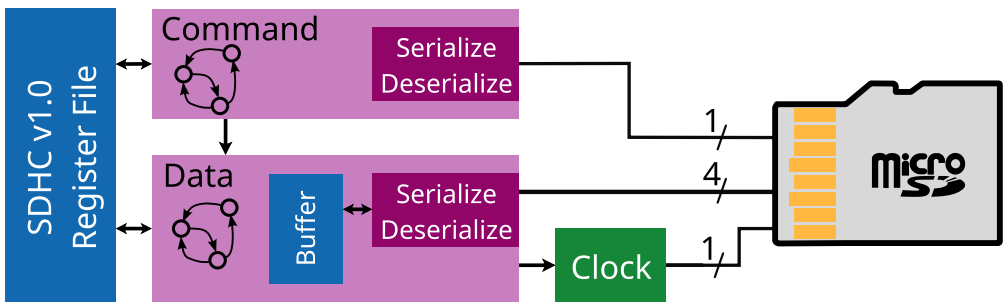


Fig. 1a. SDHC controller architecture

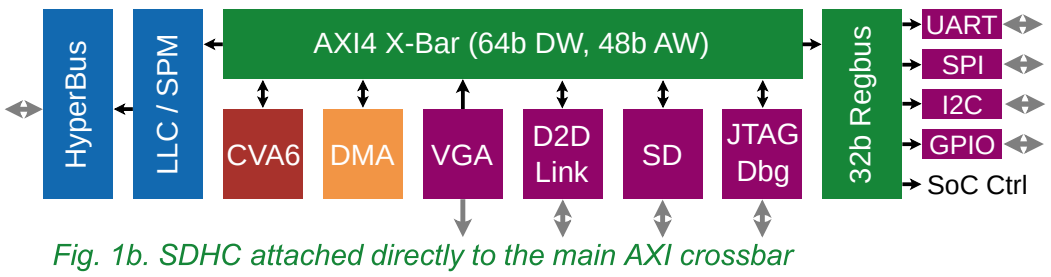


Fig. 1b. SDHC attached directly to the main AXI crossbar

3 Latency-Aware Integration

- Integrated into Linux-capable **Cheshire SoC platform**
- Peripheral connects **directly to the main AXI crossbar**, not the slow register-bus mux.
- Transfers are **latency-sensitive**: CVA6 issues no multiple outstanding requests to registers.
- Read latency cut to **11 cycles**:
  4-byte read every 29 cycles, a write every 9 cycles.

4 Driver & the Fence Bottleneck

- SDHCI compliance enables use of **existing driver**, with full support in fewer than **100 lines of code**.
- Bottleneck: CVA6's **fence instruction** (no CMOs) flushes the pipeline and data cache, causing a **~500 cycles** penalty per 32-bit register access.
- Fix: override the driver's register-access callbacks to **skip unnecessary fences**, tailoring it to CVA6.
- A bare-metal driver also enables **booting from SD**, while shrinking the boot-ROM footprint.

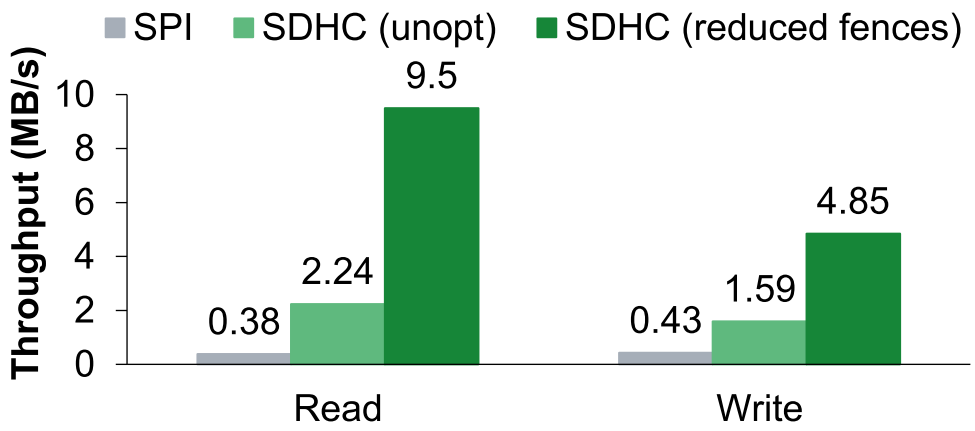


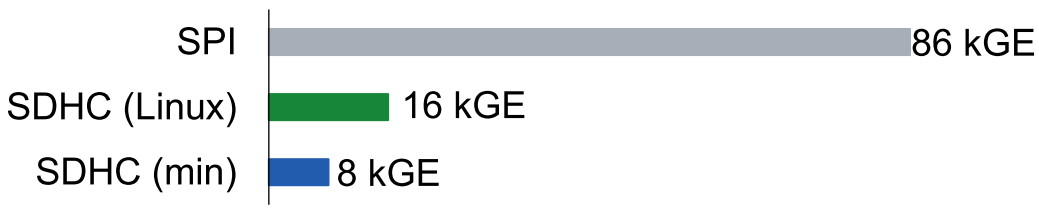
Fig. 2. Linux throughput: skipping fences lifts read 4.2× and write 3×

5 Results & Outlook

- Baremetal the SDHC reaches **11.1 MB/s**, ~8× SPI, near the 12.5 MB/s interface limit.
- Under Linux, skipping fences yields **24.9× read / 11.3× write** throughput compared to previous SPI.
- Compact size of **16 kGE** vs 86 kGE for SPI (**5.4× less**); a non-compliant embedded variant fits in just **8 kGE**.
- Outlook: **CMOs & HPDcache** for lighter fences; **SDHCI v3.0 with DMA** for far higher throughput.

► ***Open-source SDHCI v1.0 compliant RTL release and tapeout this summer***

Controller Area: up to 10× smaller



References

1. Sauter et al. Basilisk: A 34mm² End-to-End Open-Source 64-bit Linux-Capable RISC-V SoC in 130nm BiCMOS. IEEE Hot Chips 37, 2025.
2. Ottaviano et al. Cheshire: A Lightweight, Linux-Capable RISC-V Host Platform for Domain-Specific Accelerator Plug-In. IEEE TCAS-II, 2023.
3. C. Fuguet. HPDcache: Open-Source High-Performance L1 Data Cache for RISC-V Cores. ACM Comput. Frontiers, 2023.

pulp-platform.org pulp-platform @pulp_platform phsauter@iis.ee.ethz.ch

RISC-V Summit Europe 2026