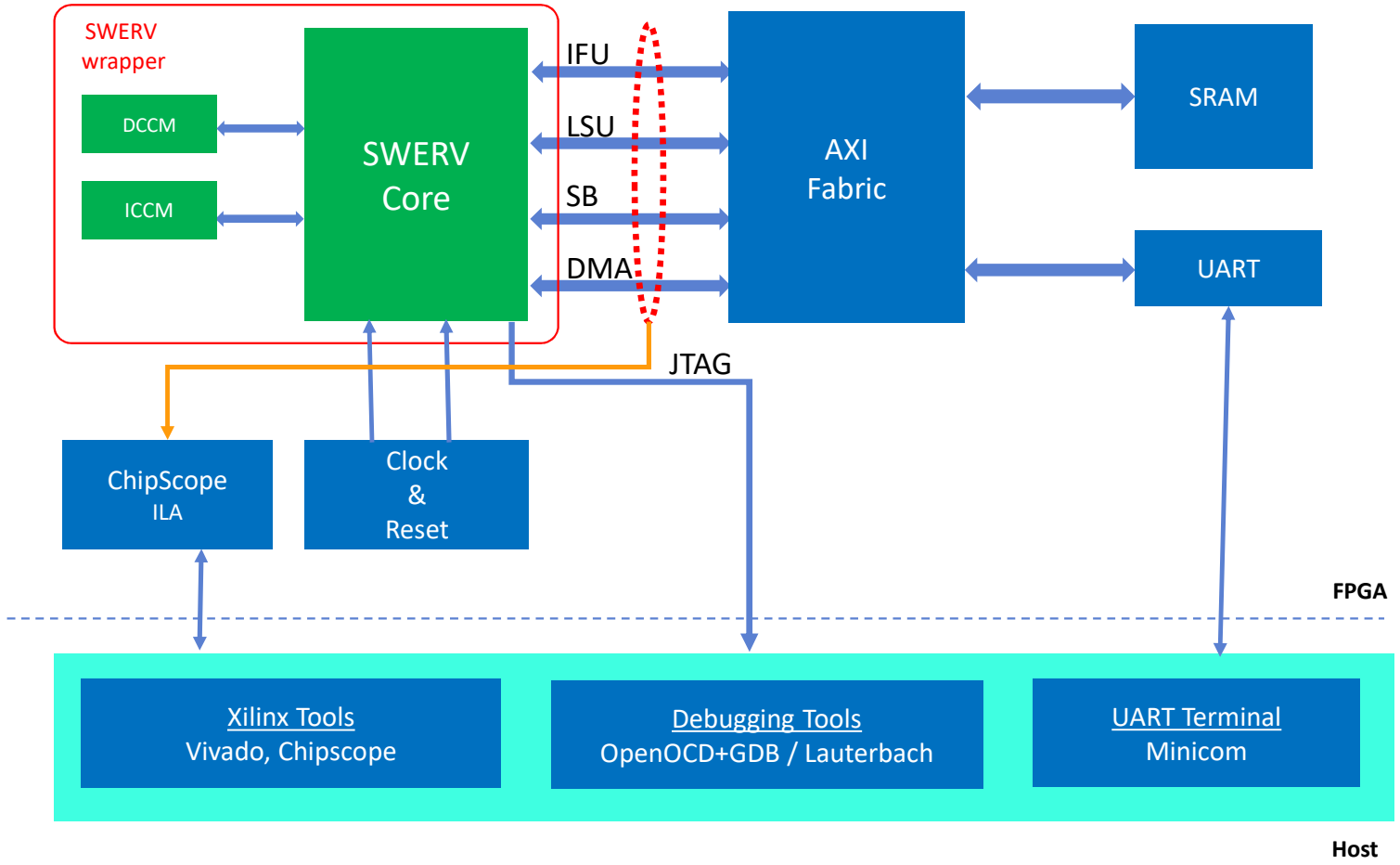
CoreMark Benchmarking for SweRV

SweRV Nexys4 Reference Design Overview



SweRV CPU Configurations

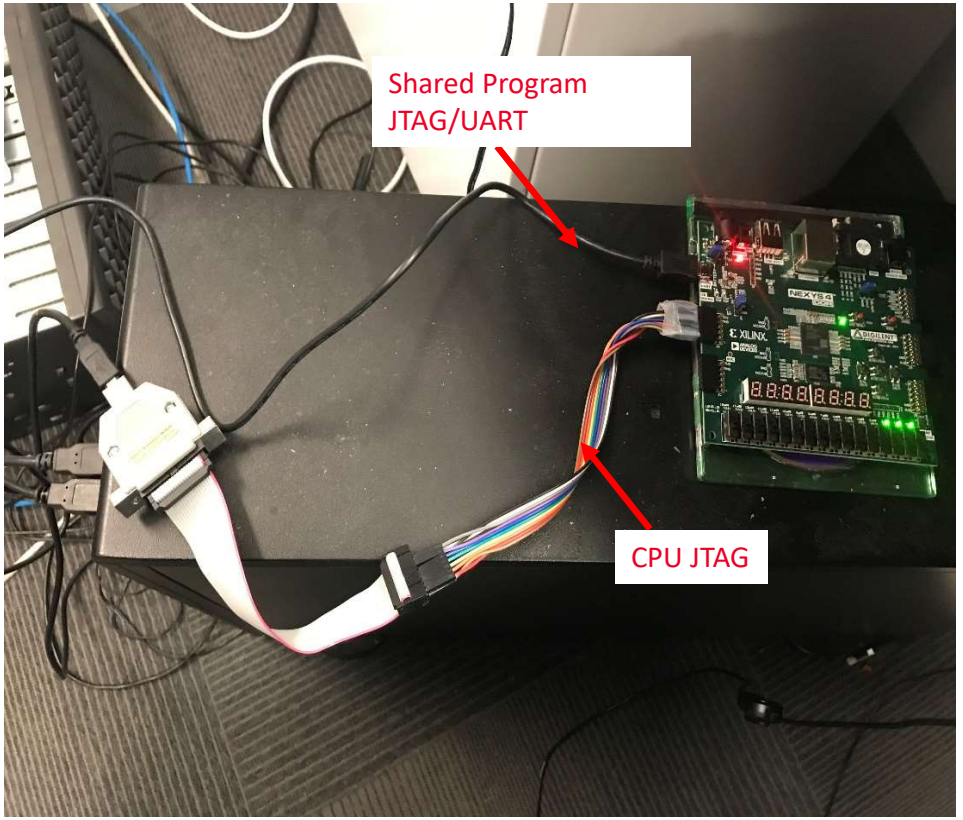
- We added 64 KB ICCM and 64 KB DCCM to keep the code and data for the application
- We set the reset vector to `ICCM start address`
- We increase BTB and BHT size to 512 and 2048 to improve the performance

```
./swerv.config -set reset_vec=0xf0090000 -set=iccm_enable=1
-unset=icache_enable -iccm_region=0xf -iccm_offset=0x90000
-iccm_size=64  -dccm_region=0xf -dccm_offset=0x80000
-dccm_size=64  -btb_size=512 -bht_size=2048
```

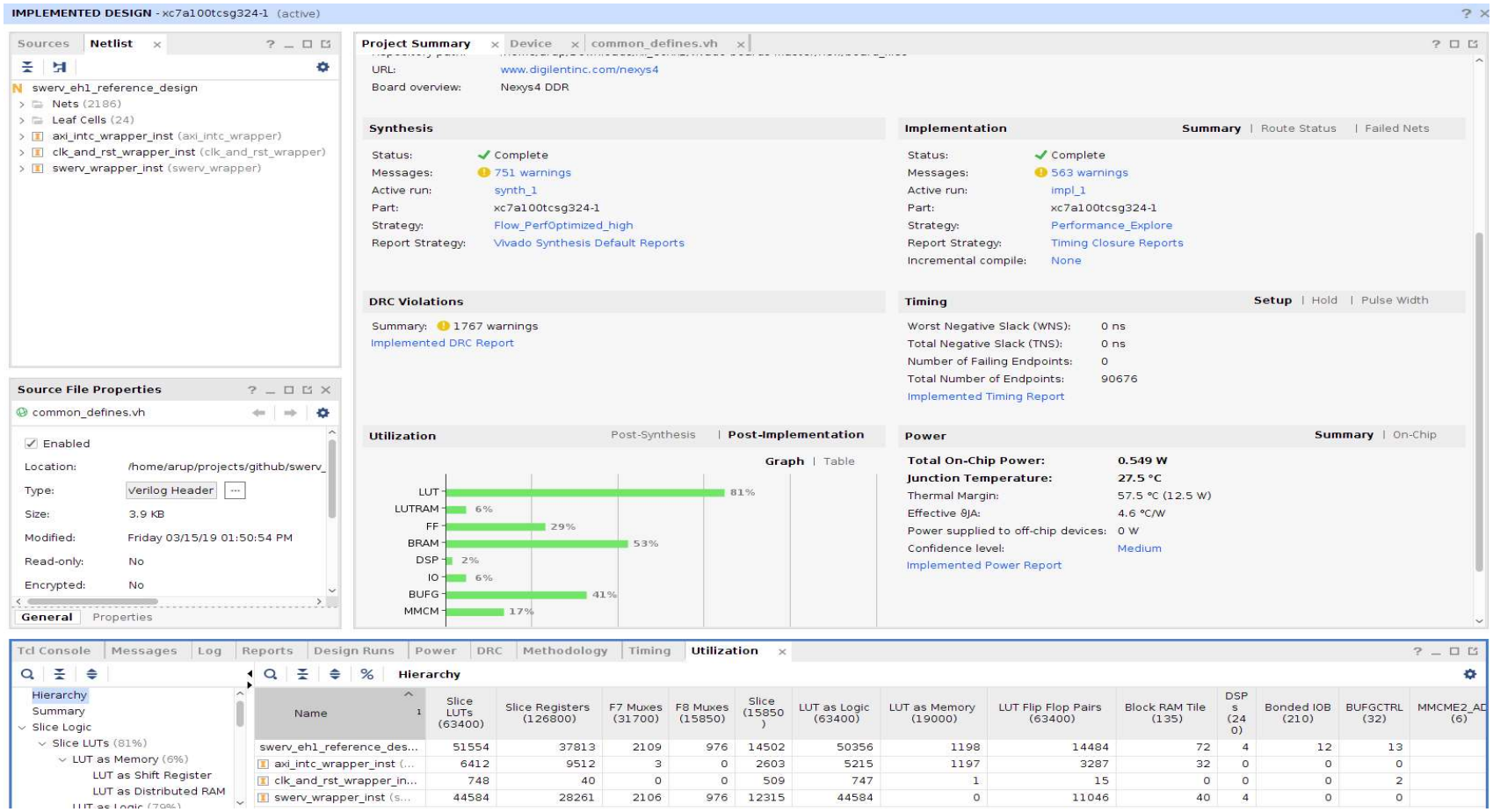
```
swerv: Using target "default"
swerv: target                   = default
swerv: ret_stack_size           = 4
swerv: btb_size                 = 512
swerv: bht_size                 = 2048
swerv: dccm_enable              = 1
swerv: dccm_region              = 0xf
swerv: dccm_offset              = 0x80000
swerv: dccm_size                = 64
swerv: dccm_num_banks           = 8
swerv: iccm_enable              = 1
swerv: iccm_region              = 0xf
swerv: iccm_offset              = 0x90000
swerv: iccm_size                = 64
swerv: iccm_num_banks           = 8
swerv: icache_enable            = 0
swerv: icache_ecc               = 0
swerv: icache_size              = 16
swerv: pic_2cycle               = 0
swerv: pic_region               = 0xf
swerv: pic_offset               = 0x100000
swerv: pic_size                 = 32
swerv: pic_total_int            = 8
swerv: lsu_stbuf_depth          = 8
swerv: lsu_wrbuf_depth          = 4
swerv: dma_buf_depth            = 2
swerv: lsu_num_nbload           = 4
swerv: dec_instbuf_depth        = 4
swerv: opensource               = 1
swerv: no_secondary_alu         = 0
swerv: Set(s) requested : reset_vec=0xf0090000 iccm_enable=1
```

FPGA Prototype

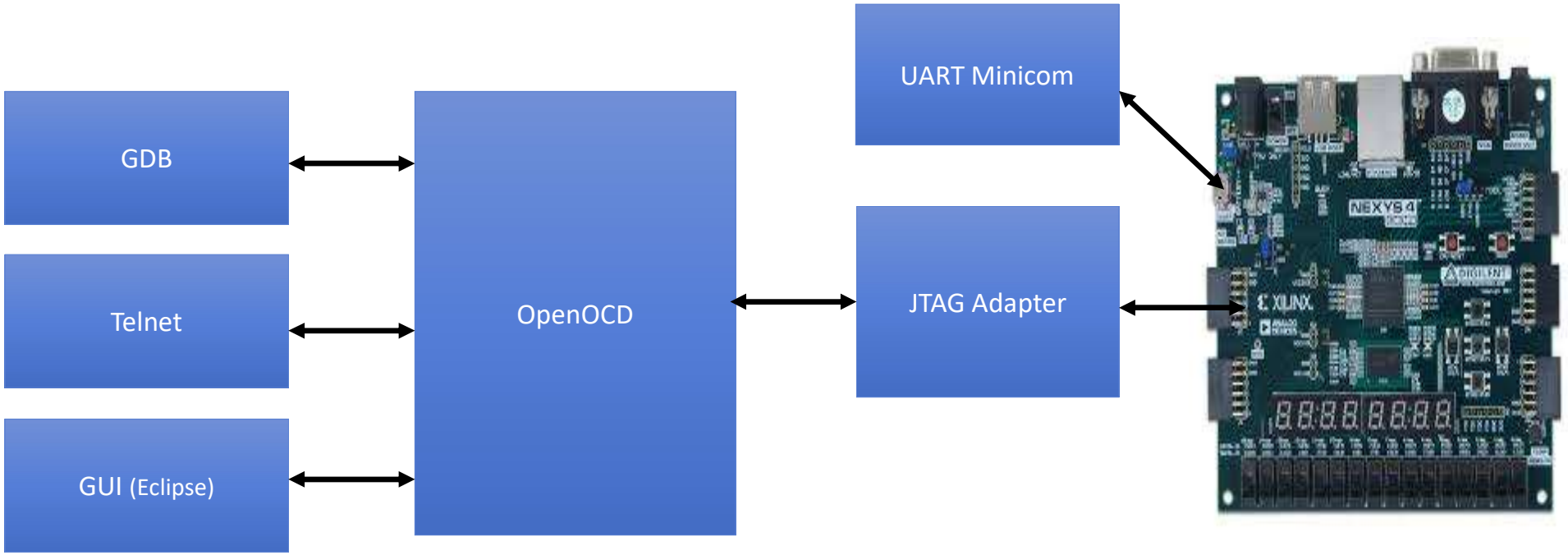
- We build the prototype on Nexys-4 DDR board
  - Artix 7 FPGA (Xilinx part number XC7A100T-1CSG324C)
  - 128MiB DDR2
  - Quad-SPI flash
- System clock runs at 40 MHz
- Shared JTAG/UART port is primarily used to download the bit file on FPGA and UART printfs.
- Olimex CPU JTAG probe is used to download and debug application software
- We used Xilinx Vivado 2018.2 toolchain for our reference design
-



FPGA Resource Utilization

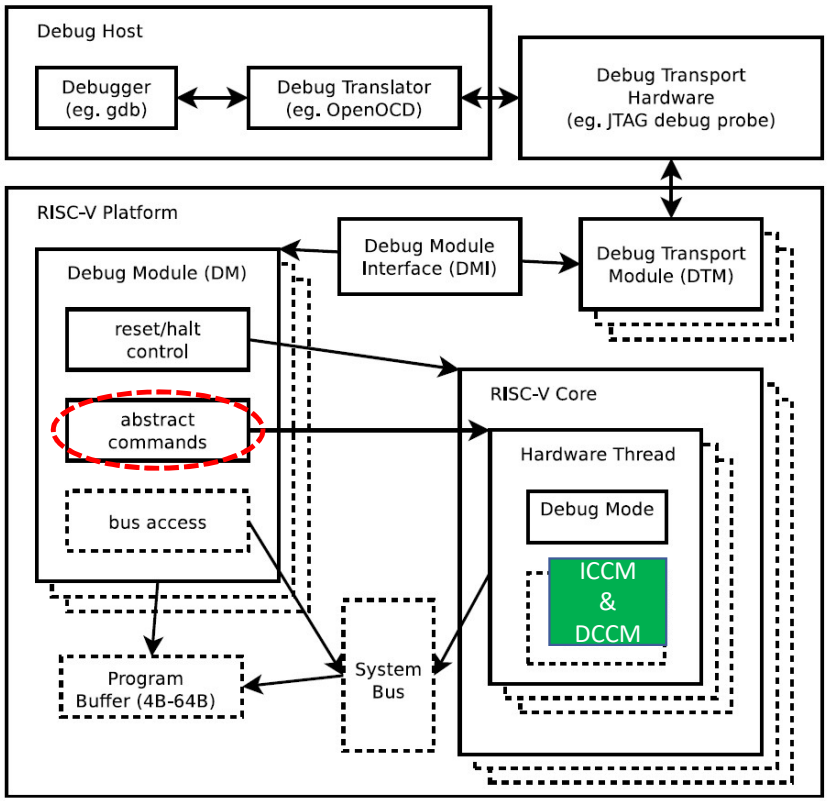


Debugging GDB+OpenOCD and Minicom



OpenOCD Extensions

- **OpenOCD** (Open On-Chip Debugger) is open-source software that interfaces with a hardware debugger's JTAG port
- It provides debugging and in-system programming for embedded target devices
- We enhance OpenOCD to provide access to ICCM and DCCM for loading and executing the code from internal memory
- We get the maximum performance when the code reside in ICCM and data at DCCM
- We added abstract commands (based on RISC-V Debug specification 0.13) to access the ICCM and DCCM (leverage and extend CREX OpenOCD work)



CoreMark

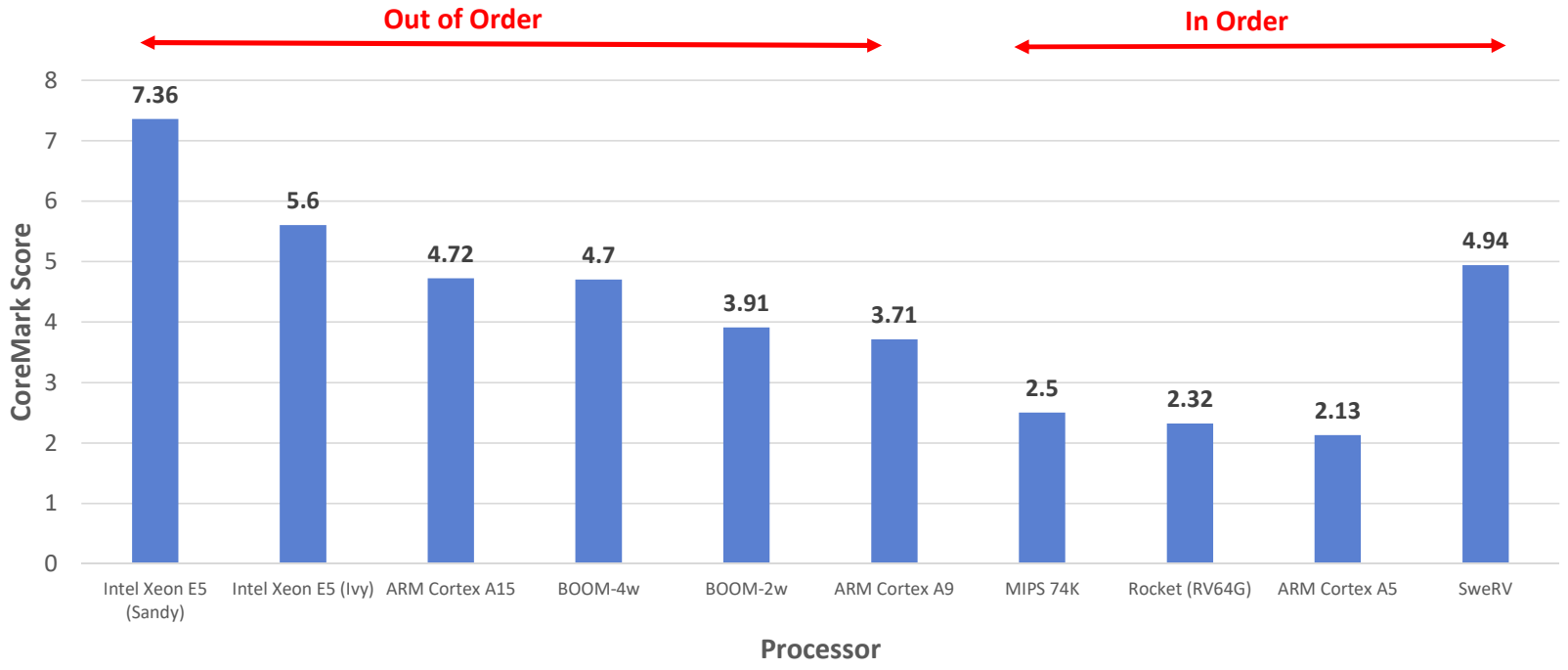
https://www.eembc.org/coremark

- CoreMark benchmark measures the performance of CPUs used in embedded systems
- It is becoming the industry standard for the embedded CPU benchmarking
- The code is written in C and contains implementation of following algorithms
  - Linked list processing (find and sort)
  - Matrix manipulation (common matrix operations)
  - State machine (determine if an input stream contains valid numbers)
  - CRC (cyclic redundancy check)
- It is designed to run on devices from 8-bit microcontrollers to 64-bit microprocessors
- All code used within the timed portion of the benchmark is part of the benchmark itself (no library calls)
- We compiled the CoreMark benchmark using RISCV 32-bit toolchain with GCC version 7.2.0

Note: CoreMark number was lower with latest version GCC 8.2.0. I got CoreMark score 4.84. It looks like GCC 7.2 generates more optimized code. We achieved 4.94.

CoreMark Results - 4.94

Score =CoreMark/MHz (SweRV ran 1000 iterations)



Source: https://www.anandtech.com/show/13964/western-digitals-riscv-swerv-core-released-for-free

Western Digital®