SCRx Family Of The RISC-V Compatible Processor IP

Alexander Redkin, CEO

www.syntacore.com
info@syntacore.com
Outline

- Company intro
- SCRx product line overview
- Extensibility/customization service
Syntacore introduction

IP company, founding member of RISC-V foundation

Develops and licenses state-of-the-art RISC-V cores
- Initial line is available and shipping to customers
- ~3 years of focused RISC-V development
- Core team comes from 10+ years of highly-relevant background
- SDKs, samples in silicon, full collateral

Full service to specialize CPU IP for customer needs
- One-stop workload-specific customization for 10x improvements
  - with tools/compiler support
- IP hardening at the required library node
- SoC integration and SW migration support
Syntacore background

Company:
- Est 2015
- R&D offices in St.Petersburg and Moscow
  - Representatives in EMEA, APAC
- 25+ employees, hiring

Team background:
- 10+ years in the corporate R&D (major semi MNC)
- Developed cores and SoC are in the mass productions
- 15+ tapeouts, 180..14nm

Expertise:
- Low-power and high-performance embedded cores and IP
- ASIP technologies and reconfigurable architectures
- Architectural exploration & workload characterization
- Compiler technologies
Outline

- **SCR1**: Compact MCU-class open-source core
  - Minimal area configuration is ~11 kGates
  - https://github.com/syntacore/scr1

- **SCR3**: High-performance 32-bit MCU with privilege modes
  - Competitive characteristics

- **SCR4**: 32-bit MCU core with high-performance FPU
  - IEEE 754-2008 compatible

- **SCR5**: Efficient mid-range APU/embedded core
  - 1GHz@28nm, virtual memory, 2-4 cores SMP, Linux

Stable designs, immediately available for evaluation
- SDKs, silicon samples, tools, documentation, support
- All cores are licensed
  - Lead customer’s SoC is deployed in the field in 2018

*Baseline* cores: extensible and customizable
Additions to the family in progress

DAC, June 2018

Copyright © 2018 Syntacore. All trademarks, product, and brand names belong to their respective owners.
# SCRx features at glance

<table>
<thead>
<tr>
<th>Features</th>
<th>SCR1</th>
<th>SCR3</th>
<th>SCR4</th>
<th>SCR5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Width</strong></td>
<td>32-bit</td>
<td>32-bit</td>
<td>32-bit</td>
<td>64-bit</td>
</tr>
<tr>
<td>ISA</td>
<td>RV32iE[IMC]</td>
<td>RV32iMC</td>
<td>RV32iMC[ED]</td>
<td>RV32i[ED]</td>
</tr>
<tr>
<td>Pipeline, stages</td>
<td>2-4</td>
<td>3-5</td>
<td>3-5</td>
<td>7-9</td>
</tr>
<tr>
<td>Branch prediction</td>
<td>Static BP, RAS</td>
<td>Static BP, RAS</td>
<td>Static BP, RAS</td>
<td>Static BP, BTB, BHIT, RAS</td>
</tr>
<tr>
<td>Execution priority levels</td>
<td>Machine</td>
<td>User, Machine</td>
<td>User, Machine</td>
<td>User, Supervisor, Machine</td>
</tr>
<tr>
<td>Extensibility/customization</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Execution units</td>
<td>MUL/DIV</td>
<td>area-opt</td>
<td>hi-perf</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>FPU</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Memory subsystem</td>
<td>TCM</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>L1$i$</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>L2$i$</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>MPU</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>MMU, virtual memory</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Debug</td>
<td>Integrated JTAG debug</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>HW BP</td>
<td>1-2</td>
<td>1-8 adv ctrl</td>
<td>1-8 adv ctrl</td>
</tr>
<tr>
<td></td>
<td>Performance counters</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>IPIC</td>
<td>IRQs</td>
<td>8-32</td>
<td>8-128</td>
<td>8-128</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>basic</td>
<td>advanced</td>
<td>advanced+</td>
</tr>
<tr>
<td>SMP support</td>
<td>OCP</td>
<td>V</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>AHB</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>AXI4</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>

*Download SCR1 free at [https://github.com/syntacore/scr1](https://github.com/syntacore/scr1)*
Compact MCU core for deeply embedded applications

- RV32I|E[MC] ISA
- **<15 kGates** in basic untethered configuration (RV32EC)
- 2 to 4 stages pipeline
- M-mode only
- Optional configurable IPIC
  - 8..32 IRQs
- Optional integrated Debug Controller
  - OpenOCD compatible
- Choices of the optional MUL/DIV unit
  - Area- or performance- optimized
- Open sourced under SHL-license (Apache 2.0 derivative with HW specific)
  - Unrestricted commercial use allowed
- High quality **free** MCU IP
  - Regular updates as specs make progress
- In the top 3 System Verilog Github repos
  - [https://github.com/syntacore/scr1](https://github.com/syntacore/scr1)
- Commercial support offered

### Performance*, per MHz

<table>
<thead>
<tr>
<th></th>
<th>-O2</th>
<th>-best**</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMIPS</td>
<td>1.30</td>
<td>1.73</td>
</tr>
<tr>
<td>Coremark</td>
<td>2.78</td>
<td></td>
</tr>
</tbody>
</table>

* Dhrystone 2.1, Coremark 1.0, GCC 7.1 BM from TCM
** -O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -fto

### Synthesis data:

- Minimal RV32EC config: 11 kGates
- Default RV32IMC config: 32 kGates
- 250+ MHz @ tsmc90lp (typical, 1.0V, +25C)
SCR1 SDK

https://github.com/syntacore/scr1-sdk

Repository contents:
- docs - SDK documentation
- fpga - SCR1 SDK FPGA projects
- images - precompiled binary files
- scr1 - SCR1 core source files
- sw – sample SW projects

Supported platforms:
- Digilent Arty (Xilinx)
- Terasic DE10-Lite (Intel)
- Arria V GX Starter (Intel)

Software:
- Bootloader
- Zephyr OS
- Tests/sample apps
- Pre-built GCC-based toolchain (Win/Linux)
SCR1 support

- Best-effort support provided
  - scr1@syntacore.com
  - academic/research use encouraged

- SLA-based commercial support available
  - Optimal configuration
  - Integration into SoC
  - Integration at client-specific SDK boards
  - Tapeout support at the target node
  - Compiler/development tools
  - Customization service
High-performance MCU-class core with privilege modes

- RV32I, optional M and C extensions
- Machine and User privilege modes
- Optional MPU (Memory Protection Unit)
- Tightly Coupled Memory (TCM) support
  - 4..1024KB
- Optional split L1 caches with ECC
- 32bit AHB or AXI4 external interface
- Optional high-performance or area-optimized MUL/DIV unit
- Optimized for area and power
- Integrated IRQ controller
- Advanced debug with JTAG i/f

**Performance**, per MHz

<table>
<thead>
<tr>
<th></th>
<th>DMIPS</th>
<th>Coremark</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2</td>
<td>1.86</td>
<td>2.937</td>
</tr>
<tr>
<td>-best**</td>
<td></td>
<td>3.30</td>
</tr>
</tbody>
</table>

* Dhrystone 2.1, Coremark 1.0, GCC 7.1 BM from TCM
** -O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flto
MCU-class core with high-performance FPU

- RV32IMCF[D] ISA
- Configurable advanced BP, fast MUL/DIV
- Integrated IRQ controller
- U- and M-mode
- 32- or 64bit bit AHB or AXI4 external interface
- Optional MPU
- Optional configurable TCM, L1 caches
- Advanced debug controller with JTAG i/f
- Configurable SP or DP FPU
  - IEEE 754-2008 compliant

---

Performance*, per MHz

<table>
<thead>
<tr>
<th></th>
<th>DMIPS</th>
<th>Coremark</th>
<th>DP Whetstone</th>
</tr>
</thead>
<tbody>
<tr>
<td>-O2</td>
<td>1.86</td>
<td>2.96</td>
<td>3.30</td>
</tr>
<tr>
<td>-best**</td>
<td>1.86</td>
<td>2.96</td>
<td>3.30</td>
</tr>
<tr>
<td>-O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flto</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Dhrystone 2.1, Coremark 1.0, GCC 7.1 BM from TCM
- ** -O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flto
Efficient mid-range APU/embedded core

- RV32IMC[AFD] ISA, 64bit option
- Single-, dual- and quad-core SMP configurations
- Advanced BP (BTB/BHT/RAS), IRQ controller, JTAG debug
- M-, S- and U-modes
- Virtual memory support, full MMU
- L1, L2 caches with coherency, atomics
- High performance double-precision FPU
- Linux and FreeBSD support
- 1GHz+ @28nm
SW development tools

IDE in production:
- GCC 7.1.1
- GNU Binutils 2.28.0
- Newlib 2.5.0
- GNU GDB 7.11.50
- Open On-Chip Debugger 0.10.0
- Eclipse 4.7.0

Hosts: Linux, Windows
Targets: BM, Linux (beta)

Also available:
- LLVM 5.0
- CompCert 2.6

Simulators:
- Qemu
- Spike

Debug solution:
Segger J-link, Olimex ARM-USB-OCD family, Digilink JTAG-HS2 supported, Lauterbach – H2’18

DAC, June 2018
SCR3 SDK Example

Integrated IDE/debug solution:
- GCC 7.1.1 (upstream)
- GNU Binutils 2.28.0 (upstream)
- Newlib 2.5.0
- GNU GDB 7.11.50
- Open On-Chip Debugger 0.10.0
- Eclipse IDE 4.7.0

Standard FPGA-based devkit board
- Number of boards supported (Arria V, Artix-7, others)
- Low-cost JTAG probe cables
- Open design

Software:
- First stage bootloader
- Zephyr/FreeRTOS, others – on request
- Sample applications, tests, benchmarks

DAC, June 2018
Integrated IDE/debug solution:
- GCC 7.1.1 (upstream)
- GNU Binutils 2.28.0 (upstream)
- Glibc 2.22
- GNU GDB 7.11.50
- Open On-Chip Debugger 0.10.0
- Eclipse IDE 4.7.0

Standard FPGA-based devkit board
- Number of boards supported (Arria 10, Virtex US+)
- Low-cost JTAG probe cables
- Open design

Software:
- First stage bootloader
- SMP Linux (4.x kernel)
- Sample applications, tests, benchmarks

Evaluation

SCR₁
- Is fully open

SCR₃-4-5
- Full package can be provided under simple evaluation agreement
Extensibility/customization: how it works

Dynamic power

Customized core

Full energy

General-purpose core

Full energy

Processing time
Workload-specific customization

Extensibility features:
- Computational capabilities
  New functions with existing HW
  New Functional Units
- Extended storage
  Mems/RF, addressable or state
  Custom AGU
- I/O ports
- Specialized system behavior
  Standard events processing
  Custom events

Domain examples:
- Computationally intensive algorithms acceleration
- Specialized processors (including DSP)
- High-throughput applications
  Wire Speed Processing/
  DPI/Real-time/Comms
SCRx extensibility example

Custom ISA extension for AES & other crypto kernels acceleration for SCR5

- Data:
  - RV32G – FPGA-based devkit, g++ 5.2.0, Linux 4.6, optimized C++ implementation
  - Rv32G + custom – same + intrinsics
  - Core i7 6800K @ 3.4GHz, g++ 5.4.0, Linux 64, optimized C++ implementation

  60..575x speedup @ modest area increase: 11.7% core, 3.7% at the CPU cluster level

<table>
<thead>
<tr>
<th>Platform</th>
<th>Fmax, MHz</th>
<th>Encoding throughput, MB/s</th>
<th>Normalized per MHz, MB/s</th>
<th>RV32G + custom speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Crypto-1</td>
<td>Crypto-2</td>
<td>AES-128</td>
</tr>
<tr>
<td>RV32G</td>
<td>20</td>
<td>0.025</td>
<td>0.129</td>
<td>0.238</td>
</tr>
<tr>
<td>RV32G + custom</td>
<td>20</td>
<td>14.375</td>
<td>15.188</td>
<td>14.502</td>
</tr>
<tr>
<td>Core i7</td>
<td>3400</td>
<td>79.115</td>
<td>235.343</td>
<td>335.212</td>
</tr>
<tr>
<td>Core i7 + NI</td>
<td>3400</td>
<td></td>
<td></td>
<td>3874.552</td>
</tr>
</tbody>
</table>

Disclaimer: Authors are aware AES allows for more efficient dedicated accelerators designs, used as example algorithm
RISC-V Foundation at DAC

- Visit the RISC-V Foundation booth (West Hall, Level Two at Booth No. 2638). Drinks will be served at the booth on Monday, June 25 and Tuesday, June 26.

- Join the RISC-V Foundation scavenger hunt for a chance to win prizes (visit booth for details).

- Attend the poster presentations from member companies at the booth from Monday, June 25, to Wednesday, June 27. See complete schedule on the last slide.
RISC-V Foundation at DAC

- Join the RISC-V Ecosystem – Reshaping The CPU Landscape workshop on June 24 from 1 p.m. to 4 p.m. PT in Room No. 3018:
  - RISC-V ISA and Foundation Overview – Rick O’Connor, RISC-V Foundation
  - RISC-V – A Diversity of Core and Accelerator Choices – Markus Levy, NXP
  - RISC-V OS Landscape – Palmer Dabbelt, SiFive
  - Designing a custom RISC-V core using Chisel – Alex Badicioiu, NXP

- Members of the RISC-V Foundation are participating in additional sessions including:
  - Core Choices: How To Navigate The Brave New World Of IP (Panel)
  - A New Golden Age For Computer Architecture: Domain Specific Accelerators And Open RISC-V (Keynote)
  - PULP-HD: Accelerating Brain-Inspired High-Dimensional Computing on a Parallel Ultra-Low Power Platform (Research Reviewed)
  - Computing Minus Moore’s Law = ?!?! (Panel)

- To learn more, please visit: https://riscv.org/2018/05/risc-v-at-design-automation-conference-dac/.
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Speaker(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monday, June 25, 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 a.m. PT</td>
<td>RISC-V ISA &amp; Foundation Overview</td>
<td>Rick O’Connor, RISC-V Foundation</td>
</tr>
<tr>
<td>Noon PT</td>
<td>From Lab to Fab: An IP Story</td>
<td>Drew Barbier, SiFive</td>
</tr>
<tr>
<td>1 p.m. PT</td>
<td>Panel: Meet The RISC-V Members at DAC 2018</td>
<td></td>
</tr>
<tr>
<td>2 p.m. PT</td>
<td>Fueling the RISC-V Ecosystem With Microsemi’s Mi-V Programmable Solutions</td>
<td>Ted Marena, Microsemi</td>
</tr>
<tr>
<td>3 p.m. PT</td>
<td>Machine Learning With RISC-V</td>
<td>Filip Blagojevic, Western Digital</td>
</tr>
<tr>
<td>4 p.m. PT</td>
<td>It’s Not Just the Core, It’s the System: Processor Trace in a Holistic World</td>
<td>Randy Fish, UltraSoC Technologies</td>
</tr>
<tr>
<td>4:30 p.m. PT</td>
<td>RISC-V Virtual Platforms, Simulators and Software Tools</td>
<td>Simon Davidmann, Imperas Software</td>
</tr>
<tr>
<td>5 p.m. PT</td>
<td>Enabling Innovation in Embedded and Enterprise Data-Centric Architectures; Networking Event; Daily Prize Draw</td>
<td>Zvonimir Bandic, Western Digital</td>
</tr>
<tr>
<td><strong>Tuesday, June 26, 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 a.m. PT</td>
<td>RISC-V ISA &amp; Foundation Overview</td>
<td>Rick O’Connor, RISC-V Foundation</td>
</tr>
<tr>
<td>Noon PT</td>
<td>It’s Not Just the Core, It’s the System: Processor Trace in a Holistic World</td>
<td>Randy Fish, UltraSoC Technologies</td>
</tr>
<tr>
<td>1 p.m. PT</td>
<td>Panel: The Key Role For The Commercial Software Tools Ecosystem For RISC-V</td>
<td></td>
</tr>
<tr>
<td>2 p.m. PT</td>
<td>RISC-V Support for Persistent Memory Systems</td>
<td>Matheus Ogleari, Western Digital</td>
</tr>
<tr>
<td>3 p.m. PT</td>
<td>RISC-V Virtual Platforms, Simulators and Software Tools</td>
<td>Simon Davidmann, Imperas Software</td>
</tr>
<tr>
<td>4 p.m. PT</td>
<td>Introducing the Latest RISC-V Core IP Series</td>
<td>Drew Barbier, SiFive</td>
</tr>
<tr>
<td>4:30 p.m. PT</td>
<td>SCRx Family of the RISC-V Compatible Processor IP</td>
<td>Alexander Redkin, Syntacore</td>
</tr>
<tr>
<td>5 p.m. PT</td>
<td>Keynote on Vision and History of RISC-V; Networking Event; Daily Prize Draw</td>
<td>Yunusup Lee, SiFive</td>
</tr>
<tr>
<td><strong>Wednesday, June 27, 2018</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 a.m. PT</td>
<td>Fueling The RISC-V Ecosystem With Microsemi’s Mi-V Programmable Solutions</td>
<td>Ted Marena, Microsemi Corporation</td>
</tr>
<tr>
<td>Noon PT</td>
<td>SCRx family of the RISC-V Compatible Processor IP</td>
<td>Alexander Redkin, Syntacore</td>
</tr>
<tr>
<td>1 p.m. PT</td>
<td>Panel: New Markets and Applications for RISC-V</td>
<td></td>
</tr>
<tr>
<td>2 p.m. PT</td>
<td>Panel: Meet the RISC-V Foundation Board Of Directors</td>
<td></td>
</tr>
<tr>
<td>3 p.m. PT</td>
<td>RISC-V ISA &amp; Foundation Overview; Daily Prize Draw</td>
<td>Rick O’Connor, RISC-V Foundation</td>
</tr>
</tbody>
</table>