

Syntacore™
Custom cores and tools



Syntacore open-source and commercial RISC-V solutions

Alexander Redkin
Executive director



Outline

- Company intro
- RISC-V compatible IP
- Customization services

Syntacore introduction

Semiconductor IP company, founding member of RISC-V foundation

Develops and licenses state-of-the-art RISC-V cores

- Immediately available, silicon-proven and **shipping to volume**
- 5+ 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

Company background

Est 2015 , 70+ EEs

HQ at Cyprus (EU)

- R&D offices in St. Petersburg and Moscow (Russia)
- Representatives in APAC, EMEA, US

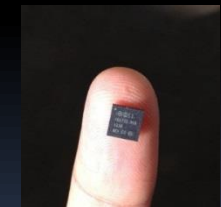
Japan: Syncom Co., LTD

Team background:

- 10+ years in the corporate R&D (major semi MNC)
- Developed cores and SoC are in the mass productions

Expertise:

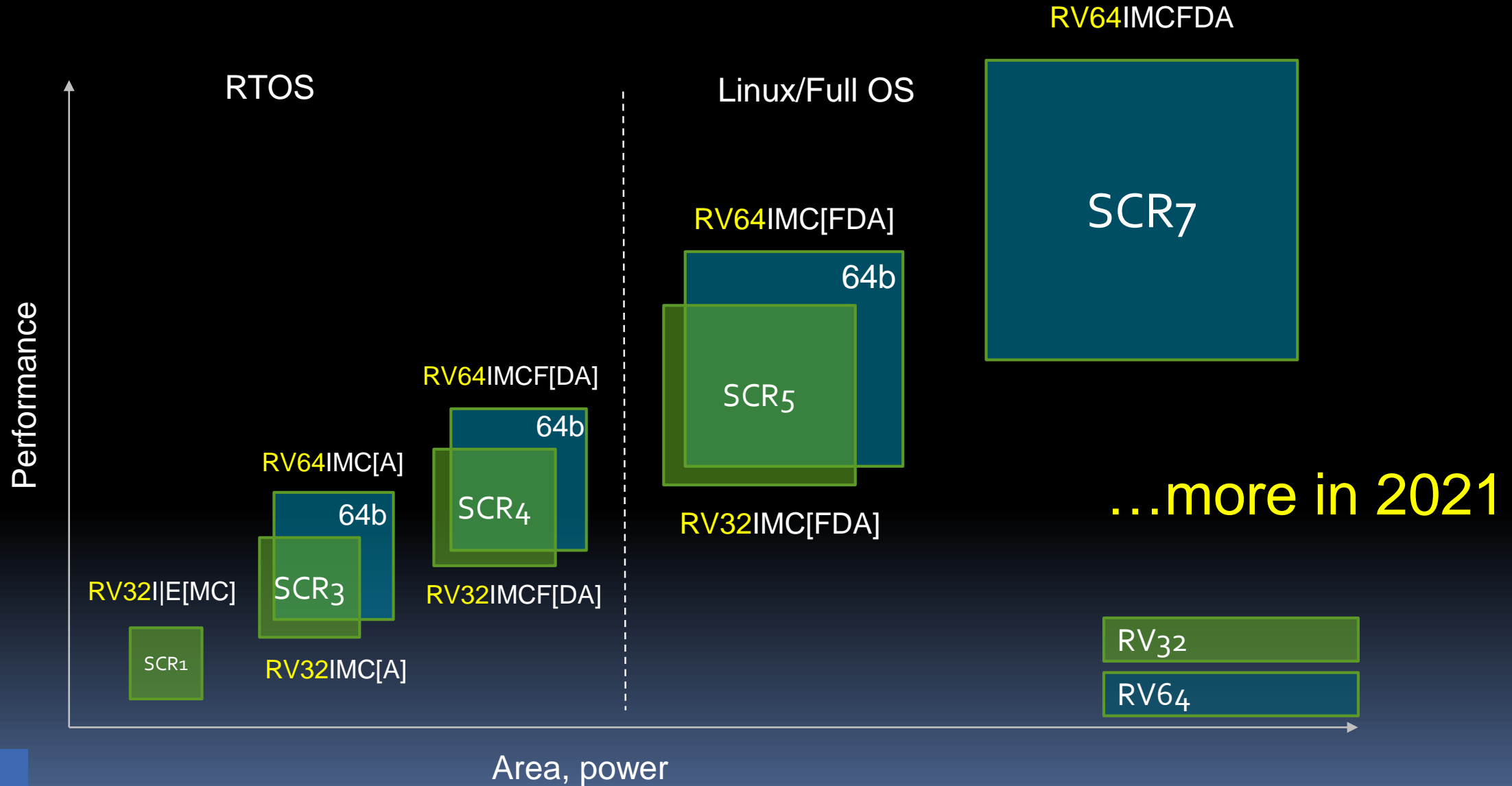
- high-performance and low-power embedded cores and IP
- ASIP technologies and reconfigurable architectures
- Architectural exploration & workload characterization
- Compiler technologies



Some current results

- State-of-the-art RISC-V CPU IP line with competitive features
 - ✓ commercially deployed in SoCs up to 5nm
- Customers in APAC, EMEA, US
 - ✓ References available
- MPWs and full-wafer production at the clients
 - ✓ SoC volumes in x100 000
 - ✓ Project example: 60-core SoC ~700 mm² @ 7nm

SCRx baseline cores



...more in 2021

RV32
RV64



State-of-the art RISC-V CPU IP

Features		RTOS/ Bare Metal			Linux/ "Full" OS	
		SCR1* <small>FREE!</small>	SCR3	SCR4	SCR5	SCR7
Width	32bit	●	●	●	●	●
	64bit		●	●	●	●
ISA		RV32IE[MC]	RV[32 64]IMC[A]	RV[32 64]IMC[F A D]	RV[32 64]IMC[A F D]	RV64IMCAFD
Pipeline type		In-order	In-order	In-order	In-order	Superscalar
Pipeline, stages		2-4	3-5	3-5	7-9	10-12
Branch prediction			Static BP, RAS	Static BP, RAS	Static BP, BTB, BHT, RAS	Dynamic BP, BTB, BHT, RAS
Execution priority levels		Machine	User, Machine	User, Machine	User, Supervisor, Machine	User, Supervisor, Machine
Extensibility/customization		●	●	●	●	●
Execution units	MUL/DIV	area-opt	○	○		
		hi-perf	○	●	●	●
	FPU			●	●	●
Memory subsystem	TCM [w/ECC parity]	○	○	○	○	○
	L1\$ [w/ECC parity]		○	○	●	●
	L2\$ [w/ECC]				○	○
	MPU		●	●	●	●
	MMU, virtual memory				●	●
Debug	Integrated JTAG debug	●	●	●	●	●
	HW BP	1-2	1-8 adv ctrl	1-8 adv ctrl	1-8 adv ctrl	1-8 adv ctrl
	Performance counters	○	○	○	○	○
Interrupt Controller	IRQs	8-32	8-1024	8-1024	8-1024	8-1024
	Features	basic	advanced	advanced	advanced+	advanced+
SMP support			up to 4 cores with coherency			up to 8-16 cores
I/F options	AHB	●	○	○	○	
	AXI4	○	●	●	●	●
	ACE					○

* Download SCR1 free at www.github.com/syntacore/scr1

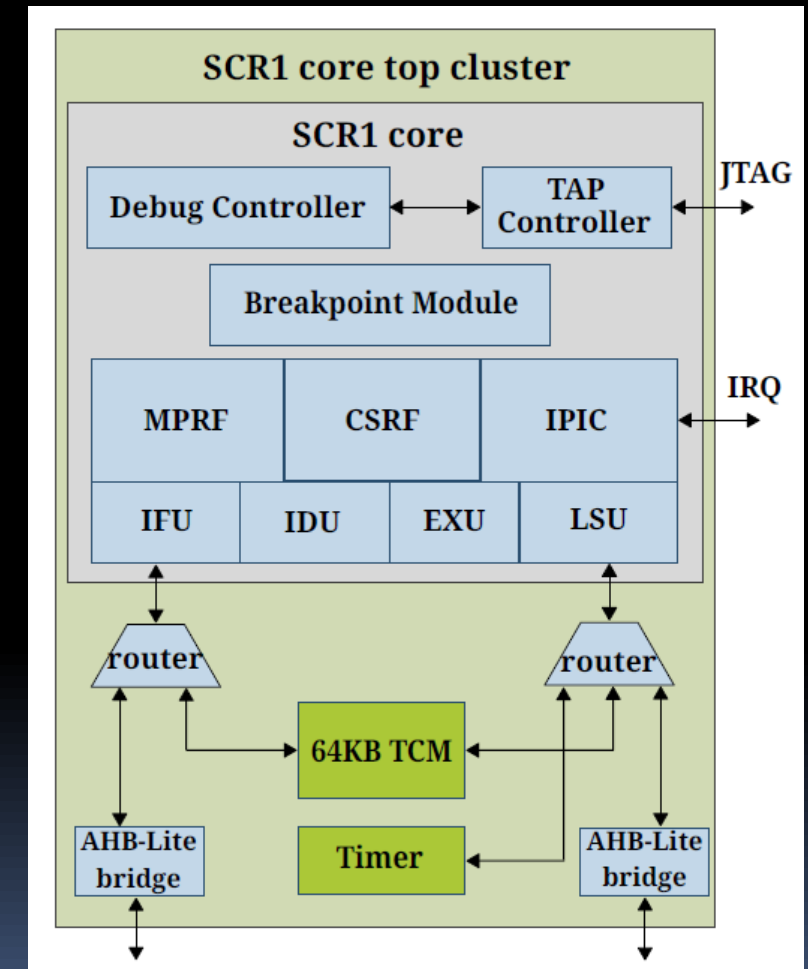
Baseline cores:

- Clean-slate designs in System Verilog
- Configurable and extensible
- 100% compatible with major EDA flows

SCR1 overview

Industry-grade compact MCU core for deeply embedded applications and accelerator control

- RV32I|E[MC] ISA
- 2 to 4 stages pipeline
- M-mode only
- Optional configurable IPIC
- Optional integrated Debug Controller
- Choices of the optional MUL/DIV unit
- Open sourced under SHL (Apache 2.0 derivative) since 2017
 - Unrestricted **commercial use allowed**
- High quality, silicon-proven **free** MCU IP
- In the top System Verilog Github repos in the world
 - <https://github.com/syntacore/scr1>
- Full collateral – TB & verification suite, SDK, specs, SW...
- Best-effort support provided, commercial offered



SCR1 overview cont

Performance*, per MHz	DMIPS	-O2	1.28
		-best**	1.89
	Coremark	-best**	2.95

* Dhrystone 2.1, Coremark 1.0, GCC 8.1 BM from TCM

** -O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flto

Synthesis data:

Minimal RV32EC config: **11** kGates

Default RV32IMC config: **32** kGates

Range 10..40+ kGates

250+ MHz @ tsmc90lp {typical, 1.0V, +25C}

What's new:

- Extensive user guide and quick start collateral
 - works out-of-the-box in all major sims
- Verilator support
- More tests/sample: RISC-V compliance, others
- Taped-out @several companies
- Regular talk at ORCONF
- **Updated and maintained**



SCR1 SDK



Syntacore™
Custom cores and tools



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

Repository content:

- 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 and Nexys 4 (Xilinx)
- Terasic DE10-Lite and Arria V GX starter (Intel)

Software:

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

```
COM4-115200baud - Tera Term VT
File Edit Setup Control Window Help

SCR loader v1.0-scr1_RC
Copyright (C) 2015-2017 Syntacore. All rights reserved.
ISA: RV32IMC (40001104) IMPID: 17090700
SYSID: 17090400 BLDID: 17090701
Platform: arty_scr1, cpucik 25MHz, sysclk 25MHz

Memory map:
00000000-0FFFFFFF 00000000 DDR
F0000000-F000FFFF 00000000 TCM
F0040000-F0040FFF 00000000 MEmory
FF000000-FF00FFFF 00000000 MMIO
FFFFFF00-FFFFFFF 00000000 On-Chip RAM

i: xmodem load @addr
q: start @addr
d: dump mem
m: modify mem
i: platform info
:
```



```
Terminal -
Philosopher 0 [P: 3] STARVING
Philosopher 1 [P: 2] HOLDING ONE FORK
Philosopher 2 [P: 1] EATING [ 475 ms ]
Philosopher 3 [P: 0] THINKING [ 625 ms ]
Philosopher 4 [C:-1] HOLDING ONE FORK
Philosopher 5 [C:-2] EATING [ 775 ms ]

Demo Description
-----
An implementation of a solution to the Dining Philosophers
problem (a classic multi-thread synchronization problem).
This particular implementation demonstrates the usage of multiple
preemptible and cooperative threads of differing priorities, as
well as dynamic mutexes and thread sleeping.
```



Fully open SDK designs + pre-build images

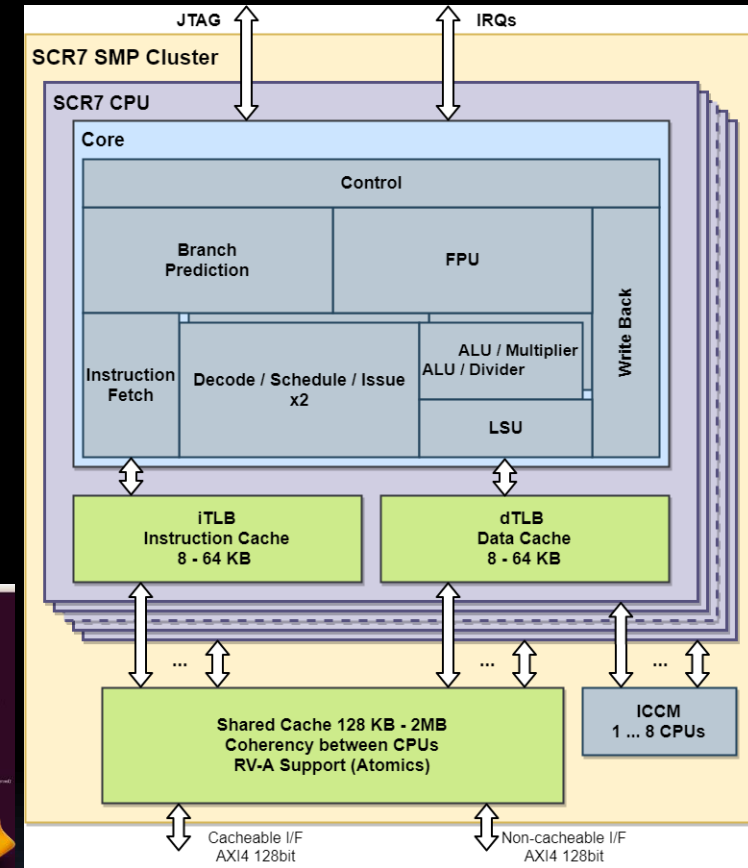
One of the **easiest paths** to start with
RISC-V



RV64 SCR7

Efficient mid-range application core

- **RV64GC** ISA
- SMP up to 8, later 16 cores
- Flexible uarch template, 10-12 stage pipeline
- **Initial SCR7 configuration:**
 - Decode and dispatch up to two instructions per cycle
 - Out-of-order issue of up to four micro-ops
 - Out-of-order completion, in-order retirement
- M-, S- and U-modes
- Virtual memory support, full MMU, Linux
- 16-64KB L1, up to 2MB L2 cache with ECC
- **1.5 GHz+** @28nm
- Advanced debug with JTAG i/f

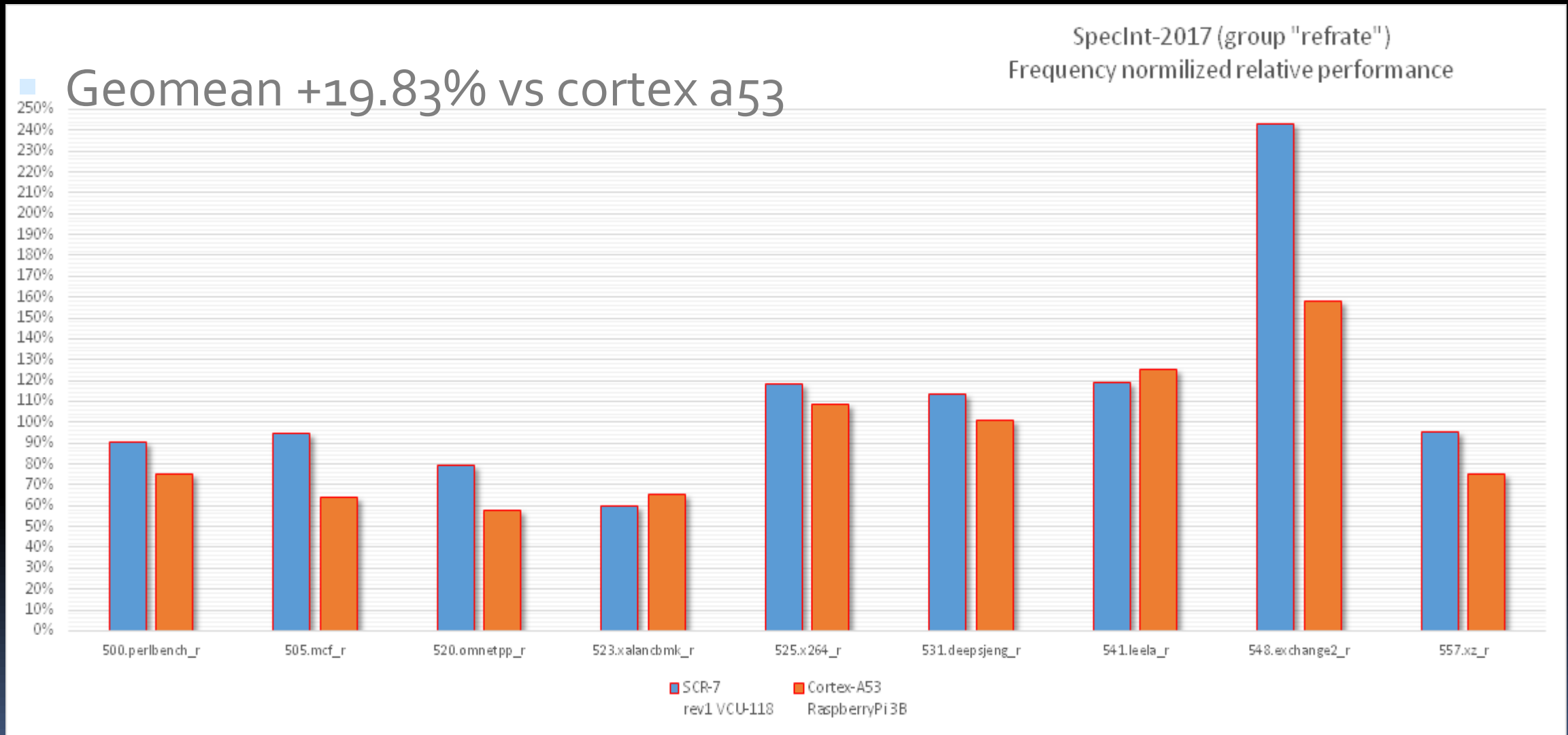


Performance*, per MHz	DMIPS	-O2	3.25
		-best**	3.80
	Coremark	-best**	5.12

* Preliminary data, 2-way implementation, Dhrystone 2.1, Coremark 1.0, GCC 8.1 BM

** O3-funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flt0

SCR7 SpecInt 2017 in HW



Fully featured SW development suite

Stable IDE in production:

- GCC 10.2
- GNU Binutils 2.31.0
- Newlib 3.0
- GNU GDB 8.0.50
- Open On-Chip Debugger 0.10.0
- Eclipse 4.9.0

Hosts: Linux, Windows

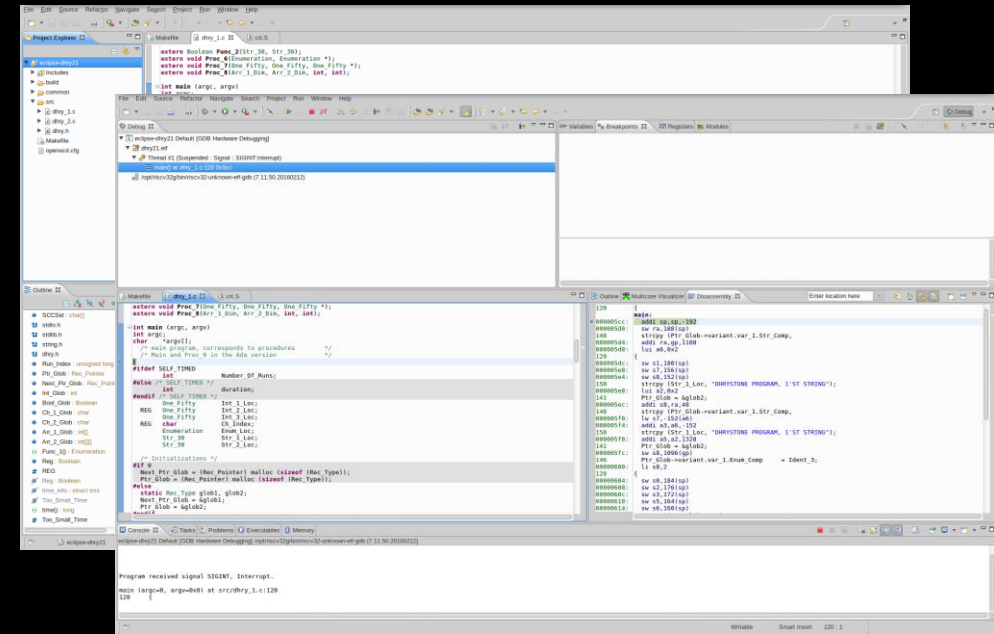
Targets: BM, Linux (beta)

Also available:

- LLVM 5.0
- CompCert 3.1
- 3rd party vendors

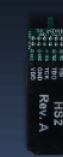
Simulators:

- Qemu
- Spike
- 3rd party vendors



JTAG-based debug solutions:

Supports: Segger J-link, Olimex ARM-USB-OCD family, Digilink JTAG-HS2, more vendors soon



Wide support by 3rd party tools and SW vendors

- Lauterbach Trace32



https://www.lauterbach.com/frames.html?pro/pro__syntacore.html

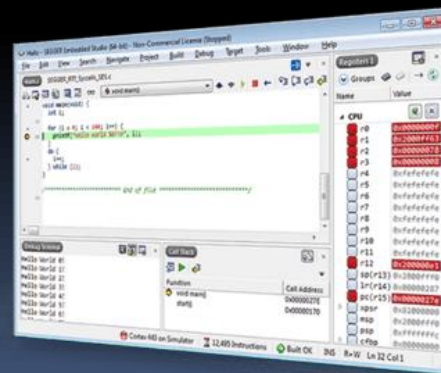


- Segger Embedded Studio

https://wiki.segger.com/Syntacore_SCR1_SDK_Arty



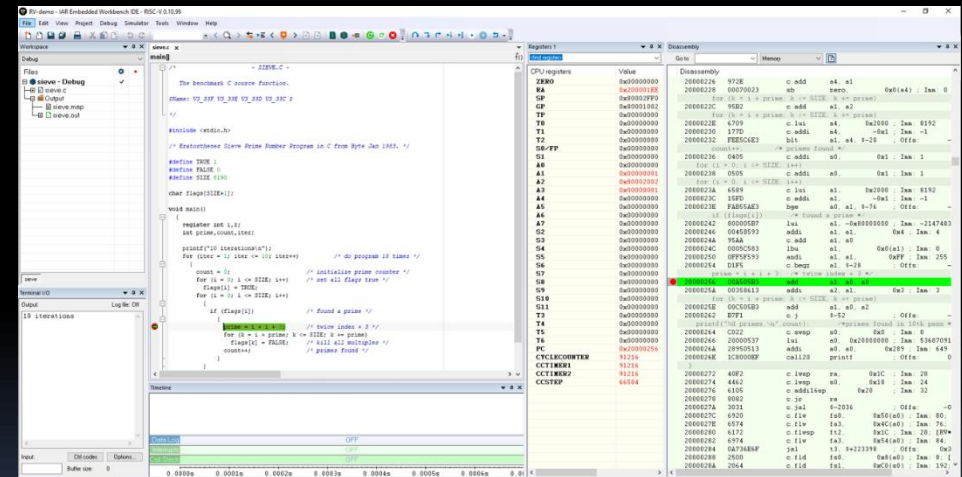
Embedded Studio



- IAR Embedded Workbench

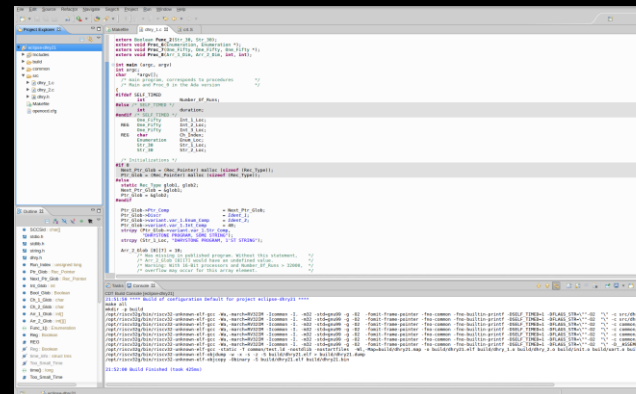


<https://www.iar.com/iar-embedded-workbench/#!?architect>



Stable Eclipse/gcc based toolchain with IDE:

- GCC 10.2
- GNU Binutils 2.31.0
- Newlib 3.0
- GNU GDB 8.0.50
- Open On-Chip Debugger 0.10.0
- Eclipse 4.9.0



```
COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

SCR loader v1.0-scr1 RC
Copyright (C) 2015-2017 Syntacore. All rights reserved.
ISA: RV32IMC [40001104] IMPID: 17090700
SYSID: 17090400 BLDID: 17090701
Platform: arty_scr1, cpucclk 25MHz, sysclk 25MHz
Memory map:
00000000-0FFFFFFF 00000000 DDR
F0000000-F000FFFF 00000000 TCM
F0040000-F0040FFF 00000000 MTimer
FF000000-FFFFFFF 00000000 MMIO
FFFFFF00-FFFFFFF 00000000 On-Chip RAM
```

```
1: xmq
g: sta
g: dta
m: mca
i: pla
:
Philosopher 0 [P: 3] STARVING
Philosopher 1 [P: 2] HOLDING ONE FORK
Philosopher 2 [P: 1] EATING [ 475 ms ]
Philosopher 3 [P: 0] THINKING [ 625 ms ]
Philosopher 4 [C:-1] HOLDING ONE FORK
Philosopher 5 [C:-2] EATING [ 775 ms ]
```



HW platform based on standard FPGA dev.kits

- Multiple boards supported (Altera, Xilinx)
- Low-cost 3rd party JTAG tools
- Open design for easy start

SW:

- Bootloader
- OS: Zephyr/FreeRTOS/Linux
- Application samples, tests, benchmarks

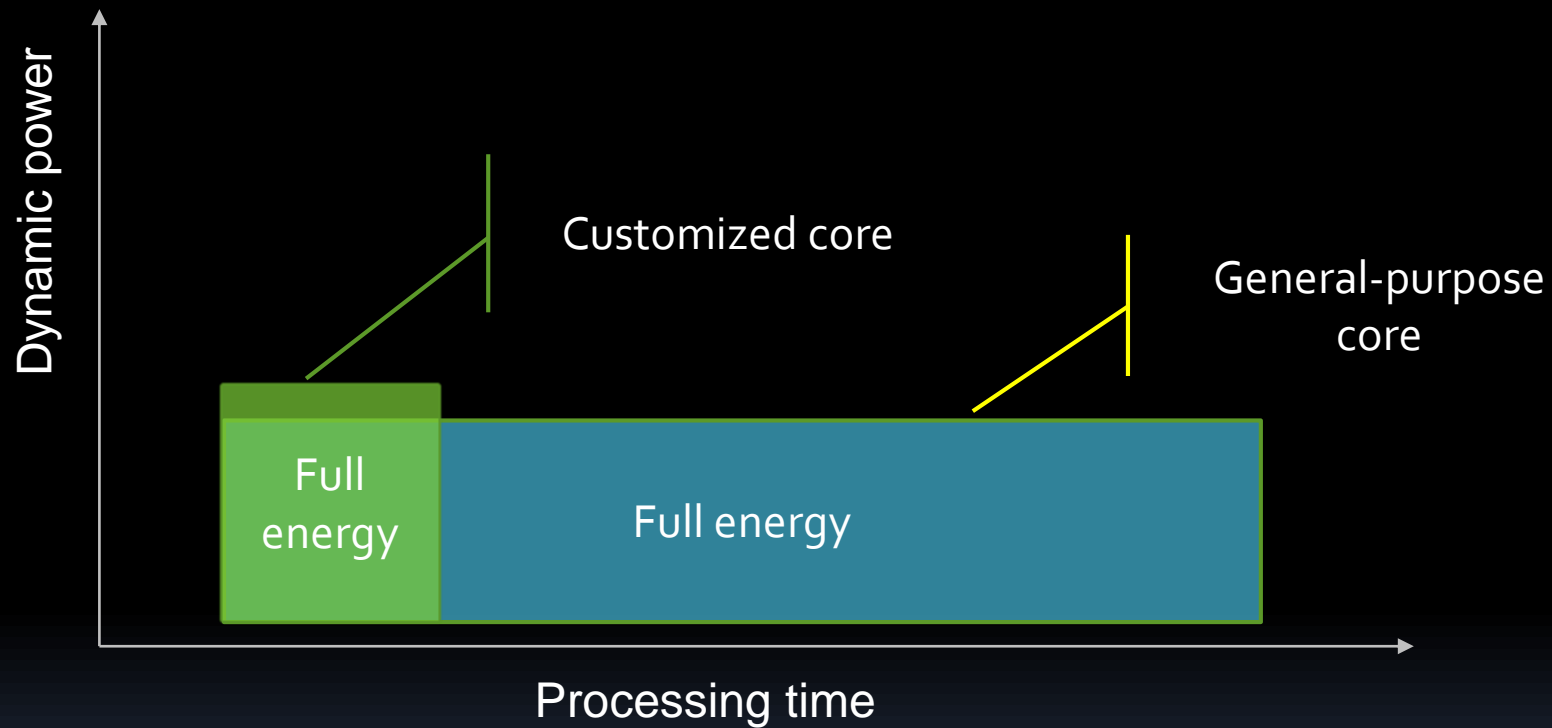


<https://www.xilinx.com/products/boards-and-kits/1-60lhw.html>



https://www.altera.com/products/boards_and_kits/dev-kits/altera/kit-arria-v-starter.html

Extensibility/customization: how it works



Workload-specific customization

Extensibility features:

- Computational capabilities
 - New functions using 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

Details
in paper
@EW2018
conference

Platform	Fmax, MHz	Encoding throughput, MB/s			Normalized per MHz, MB/s			RV32G + custom speed-up		
		Crypto-1	Crypto-2	AES-128	Crypto-1	Crypto-2	AES-128			
RV32G	20	0.025	0.129	0.238	0.00125	0.00645	0.0119	575.00	117.74	60.93
RV32G + custom	20	14.375	15.188	14.502	0.71875	0.7594	0.7251			
Core i7	3400	79.115	235.343	335.212	0.02327	0.06922	0.09859	30.89	10.97	7.35
Core i7 + NI	3400			3874.552			1.13957			0.64

Disclaimer: Authors are aware AES allows for more efficient dedicated accelerators designs, used as example algorithm



Getting access/evaluation

SCR₁

- Is fully open: <https://github.com/syntacore/scr1> and <https://github.com/syntacore/scr1-sdk>
- SHL-licensed with unrestricted commercial use allowed
 - Commercial SLA-based support is available

SCR 3|4|5|7

- Full package* access is available after simple evaluation agreement

For more info: evaluation@syntacore.com

(*) sufficient for simulation and synthesis

IP collateral (what is included)



Standard core package (SCR3)

- RISC-V compatible core
 - RV[32|64]IMC[A] ISA
 - RTL (encrypted for evaluation stage), suitable for simulation and synthesis
 - Netlist for the required FPGA devices (Xilinx/Altera)
- Simulation and verification environment
 - Testbench, Integration verification environment
 - Architectural and compliance tests suites (pre- and post-si)
- Synthesis support harness
 - sample scripts, SDC/timing constraints for the required flow
- Reference instantiation examples (for AHB and AXI sockets)
- Back-end support @ required process node (PDK access to be provided)
 - Full cycle: synthesis, floor-planning, netlist verification, PaR/CTS/timing closure, DRC, FEV, DFT)
- Support for 1 tapeout up to a year is included

Tools (pre-built & sources)

- GCC based toolchain
 - compiler, debugger, linker, functional simulator, binutils, newlib, openocd
- Eclipse-based IDE (Linux, Windows)

FPGA-based SDK

- Sample FPGA project (open design)
- pre-build FPGA and SW images

SW:

- First stage bootloader (SC-BL)
- ZephyrOS /FreeRTOS for the SDK board, including BSP
- Application samples for BM env (tests)

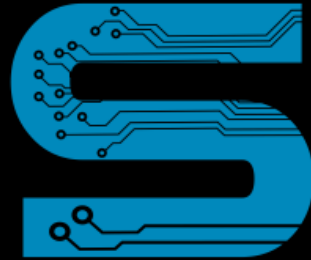
Documentation

- SCR_x user manual (quick-start/integration guide)
- SCR_x EAS (External architecture specification)
- SCR_x ISM (Instruction set manual)
- SCR_x SDK guide
- Integration verification environment guide
- Tools guide (IDE & CLI)



Summary

- Syntacore offers high-quality RISC-V compatible CPU IP
 - Founding member, fully focused on RISC-V since 2015
 - Silicon-proven and shipping in full-wafer production
 - Turnkey IP customization services
 - with full tools/compiler support
- Local contact in Japan: Syncom Co., LTD
 - Mr. Katsuhiro Katayama katayama@synkom.co.jp



Syntacore™
Custom cores and tools

info@syntacore.com

Thank you!