Syntacore 32/64bit RISC-V IP product line

Alexander Redkin
Executive director

RISC-V Tokyo Day
September 30 2019

Outline

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

Syntacore introduction

IP company, founding member of RISC-V foundation

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

- Silicon-proven and shipping to customers
- 4+ 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

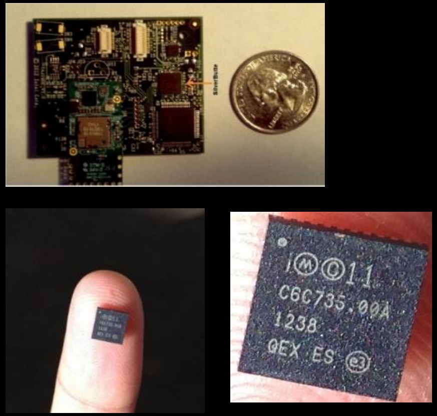
Est 2015 , 30+ EEs

HQ at Cyprus (EU)

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

Team background:

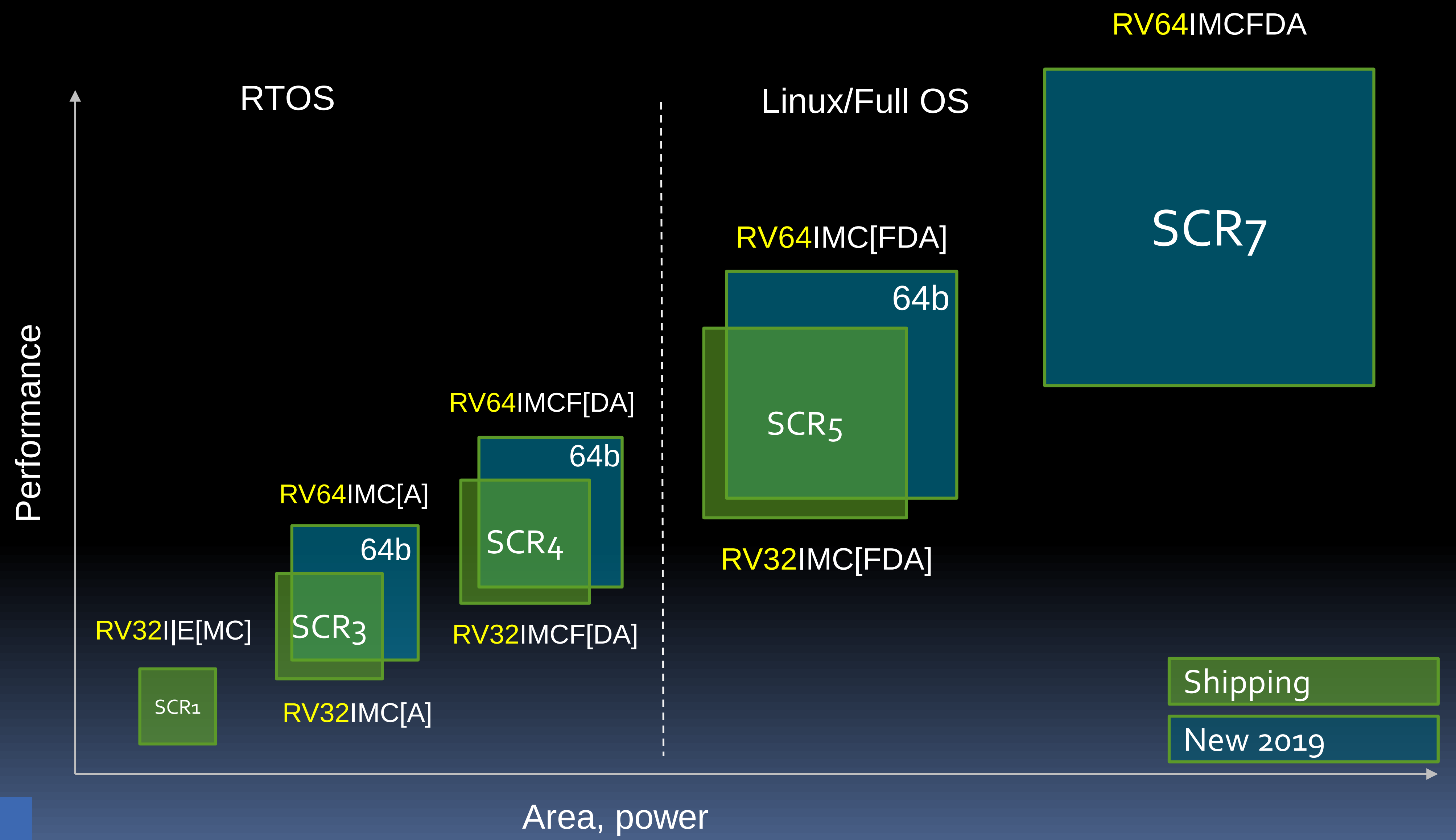
- 10+ years in the corporate R&D (major semi MNC)
- Developed cores and SoC are in the mass productions
- 15+ tapeouts, 180..14nm



Expertise:

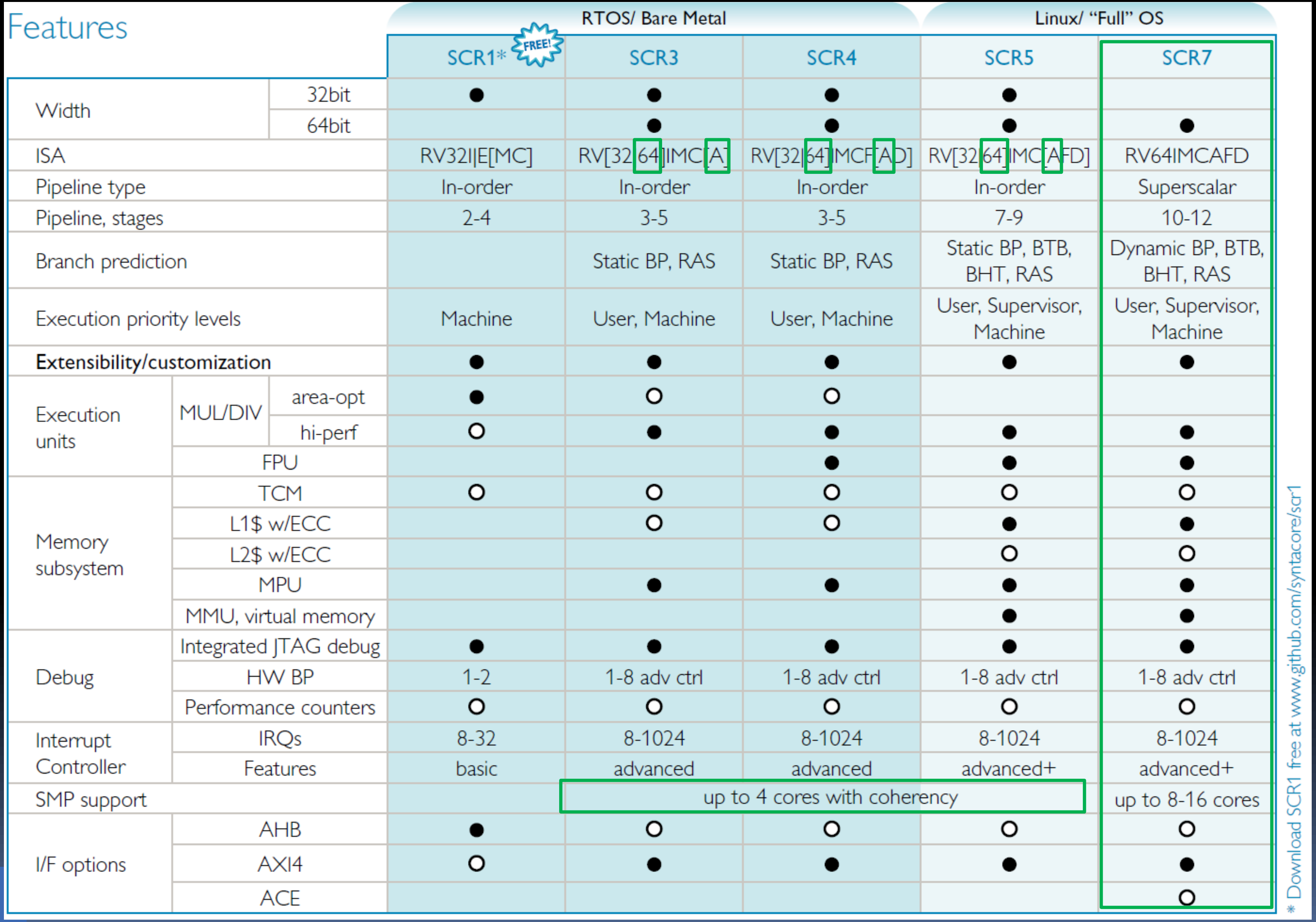
- Low-power and high-performance embedded cores and IP
- ASIP technologies and reconfigurable architectures
- Architectural exploration & workload characterization
- Compiler technologies

SCRx baseline cores 2019



SCRx IP features at glance

Syntacore™
Custom cores and tools



Features			SCR1* FREE!	SCR3	SCR4	SCR5	SCR7
			RTOS/ Bare Metal			Linux/ "Full" OS	
Width		32bit	●	●	●	●	
		64bit		●	●	●	●
ISA			RV32I|E[MC]	RV[32|64]IMC[A]	RV[32|64]IMCF[AD]	RV[32|64]IMC[AFD]	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		○	○	○	○	○
	L1$ w/ECC			○	○	●	●
	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

Sk IT

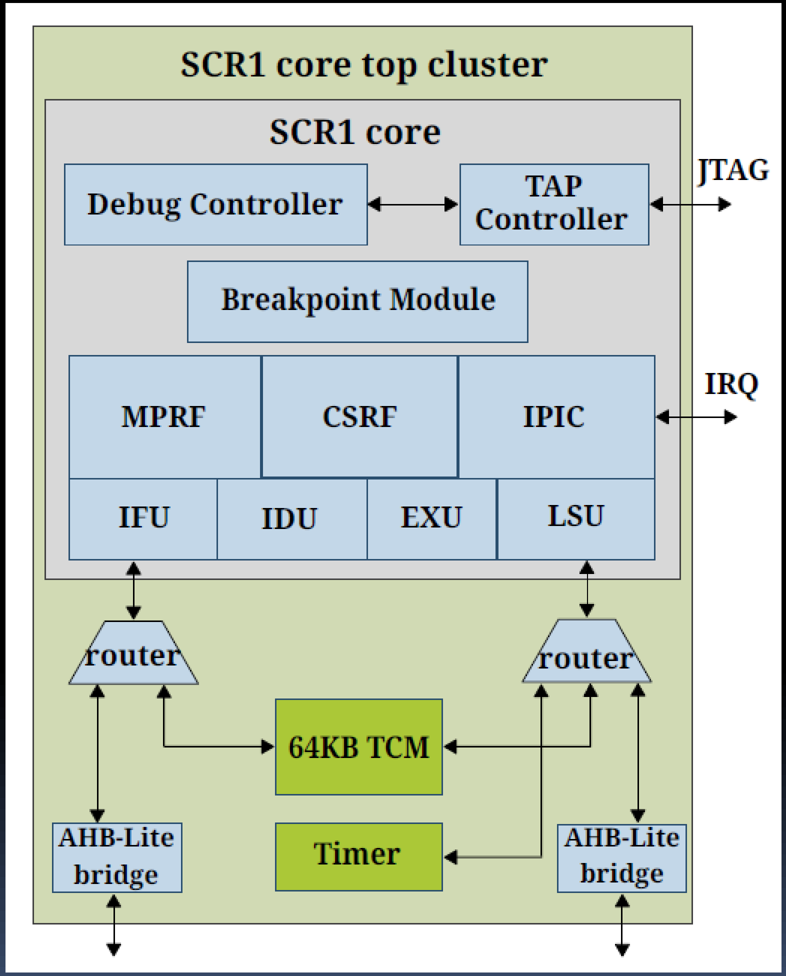
SCR1 overview

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
  - 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)
  - Unrestricted commercial use allowed

Open

- High quality, silicon-proven free MCU IP
- In the top System Verilog Github repos in the world
- 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

Demo

SCR1 SDK

Open

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

Repository content:

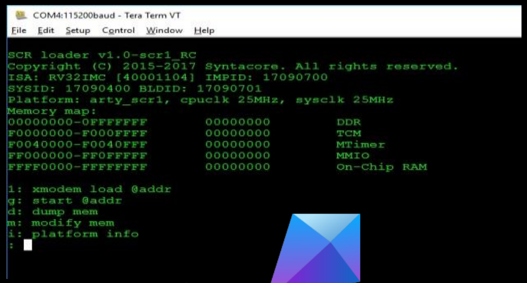
- docs - SDK documentation
- fpga - SCR1 SDK FPGA projects
- images - precompiled binary files
- scr1 - SCR1 core source files
- sw – sample SW projects

Supported platforms:

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

Software:

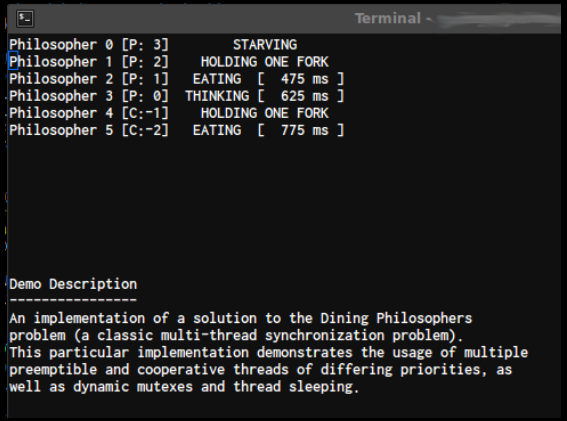
- Bootloader
- Zephyr OS
- Tests/sample apps
- Pre-built GCC-based toolchain (Win/Linux)


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, cpuclk 25MHz, sysclk 25MHz
Memory map:
00000000-0FFFFFFF     00000000        DDR
F0000000-F000FFFF     00000000        TCM
F0040000-F0040FFF     00000000        MTimer
FF000000-FF0FFFFF     00000000        MMIO
FFFF0000-FFFFFFFF     00000000        On-Chip RAM

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

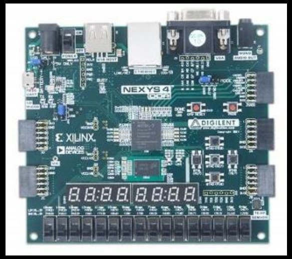
Zephyr™

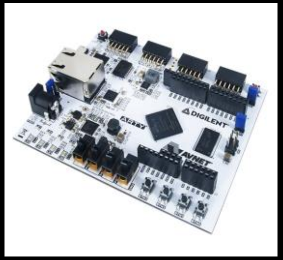


```
Philosopher 0 [P: 3]        STARVING
Philosopher 1 [P: 2]    HOLDING ONE FORK
Philosopher 2 [P: 1]   EATING  [  475 ms ]
Philosopher 3 [P: 0]  THINKING [  625 ms ]
Philosopher 4 [C:-1]    HOLDING ONE FORK
Philosopher 5 [C:-2]   EATING  [  775 ms ]




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.
```





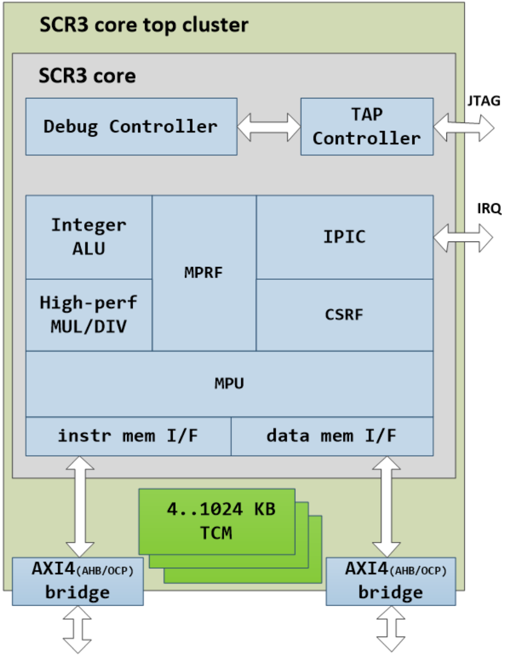
Fully open designs and pre-build images

One of the easiest paths to start with RISC-V

SCR3: 32 or 64 bit

High-performance multicore capable MCU-class core

- RV32I[MCA] or RV64I[MCA] ISA
- Machine and User privilege modes
- Optional MPU (Memory Protection Unit)
- Optional Tightly Coupled Memory (TCM), L1 caches ECC/parity
- 32|64bit AHB or AXI4 external interface
- Optional high-performance or area-optimized MUL/DIV unit
- Integrated IRQ controller and PLIC
- Advanced debug with JTAG i/f
- Multicore configs up to 4 SCRx cores
  - SMP and heterogeneous
  - with memory coherency



			RV32	RV64
Performance*, per MHz	DMIPS	-O2	1.86	1.97
		-best**	2.937	3.27
	Coremark	-best**	3.30	3.40

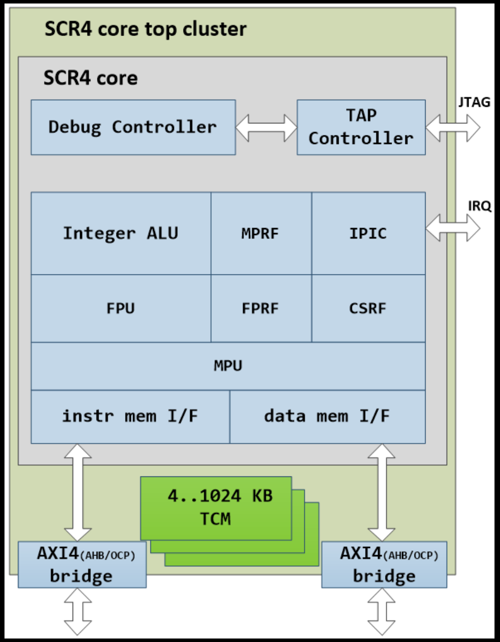
* Dhrystone 2.1, Coremark 1.0, GCC 8.1 BM from TCM

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

SCR4: 32 or 64 bit

High-performance multicore capable MCU core with FPU

- RV32IMCF[DA] or RV64IMCF[DA] ISA
- U- and M-mode
- Configurable advanced BP, fast MUL/DIV
- Integrated IRQ controller and PLIC
- 32|64bit bit AHB or AXI4 external interface
- Optional MPU, TCM, L1 caches w/ECC
- Advanced debug controller with JTAG
- Configurable SP or DP FPU
  - IEEE 754-2008 compliant
- Multicore configs up to 4 SCRx cores
  - SMP and heterogeneous
  - with memory coherency



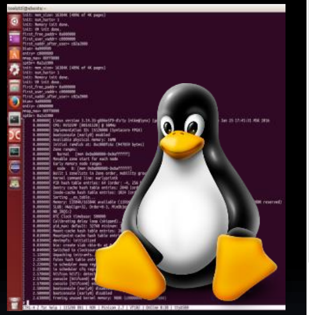
			RV32	RV64
Performance*, per MHz	DMIPS	-O2	1.86	1.97
		-best**	2.96	3.27
	Coremark	-best**	3.30	3.40
	DP Whetstone	-best**	1.22	1.22

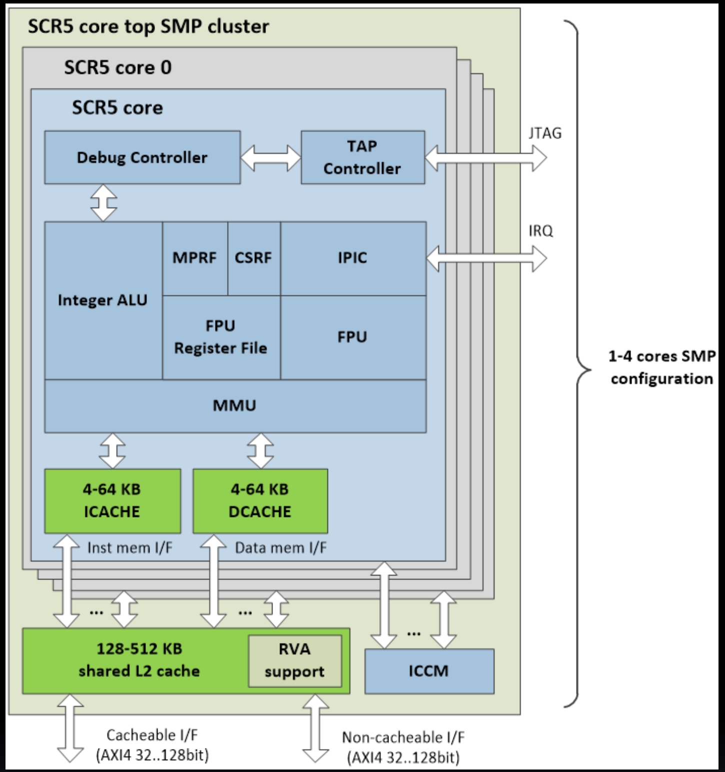
* Dhrystone 2.1, Coremark 1.0, GCC 8.1 BM from TCM
** -O3 -funroll-loops -fpeel-loops -fgcse-sm -fgcse-las -flto

SCR5: 32 or 64 bit

Efficient entry-level APU/embedded core

- RV32IMC[AFD] or RV64IMC[AFD] ISA
- Multicore configs up to 4 SCRx cores
  - SMP and heterogeneous
- Advanced BP (BTB/BHT/RAS)
- IRQ controller (integrated and PLIC)
- M-, S- and U-modes
- Virtual memory support, full MMU
- L1, L2 caches with coherency, atomics, ECC
- High performance double-precision FPU
- Linux and FreeBSD support
- 1GHz+ @28nm
- Advanced debug with JTAG i/f





			RV32	RV64
Performance*, per MHz	DMIPS	-O2	1,60	1.70
		-best**	2,48	2.62
	Coremark	-best**	2,83	3.02

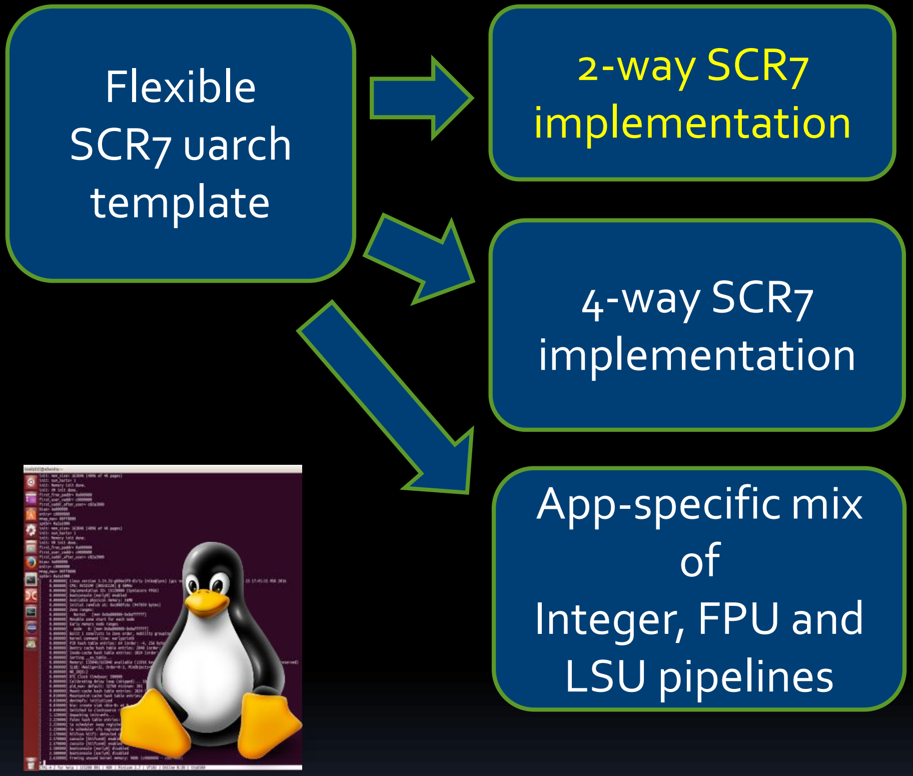
* Dhrystone 2.1, Coremark 1.0, GCC 8.1 BM from TCM

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

RV64 SCR7

Efficient mid-range application core

- RV64GC ISA
- Multicore configs up to 8, later 16 cores
- Flexible uarch template, 10-12 stage pipeline
- Initial SCR7 configuration (Q1’19):
  - Decode and dispatch of 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
- 16-64KB L1, up to 2MB L2 cache with ECC
- 1.2GHz+ @28nm
- Advanced debug with JTAG i/f


Flexible SCR7 uarch template
2-way SCR7 implementation
4-way SCR7 implementation
App-specific mix of Integer, FPU and LSU pipelines

Performance*, per MHz	DMIPS	-O2	2.75
		-best**	3.01
	Coremark	-best**	5.00*

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

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

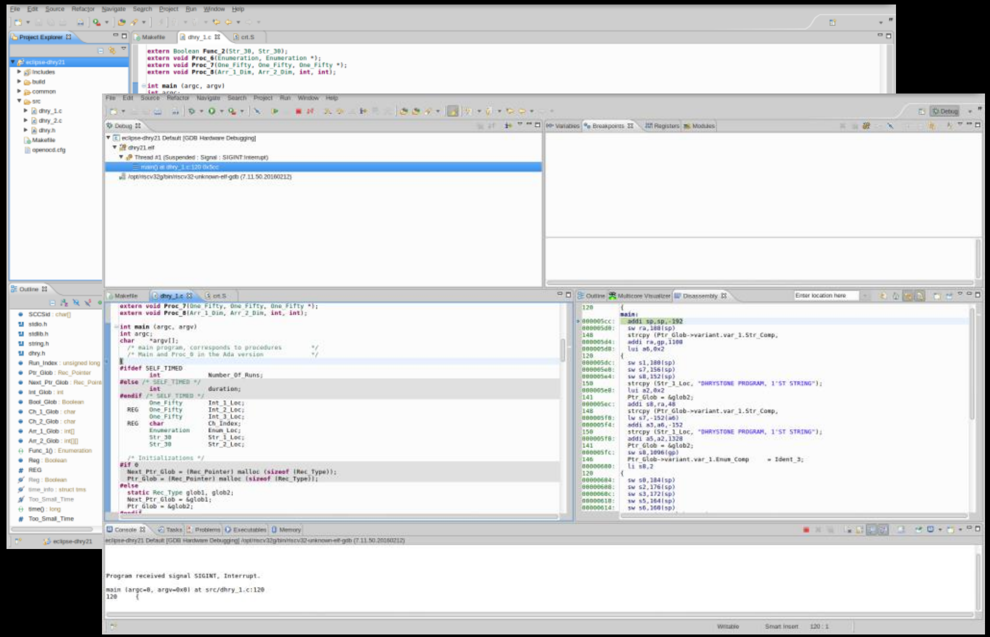
Fully featured SW development suite

Stable IDE in production:

- GCC 8.1
- 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)



eclipse-dhry21 Default [GDB Hardware Debugging]
dhry21.elf
Thread #1 (Suspended : Signal : SIGINT:Interrupt)
main() at dhry_1.c:120 0x5cc
/opt/riscv32g/bin/riscv32-unknown-elf-gdb (7.11.50.20160212)

```
extern void Proc_7(One_Fifty, One_Fifty, One_Fifty *);
extern void Proc_8(Arr_1_Dim, Arr_2_Dim, int, int);

int main (argc, argv)
int argc;
char    *argv[];
  /* main program, corresponds to procedures        */
  /* Main and Proc_0 in the Ada version             */

#ifdef SELF_TIMED
        int             Number_Of_Runs;
#else /* SELF_TIMED */
        int             duration;
#endif /* SELF_TIMED */
        One_Fifty       Int_1_Loc;
  REG   One_Fifty       Int_2_Loc;
        One_Fifty       Int_3_Loc;
  REG   char            Ch_Index;
        Enumeration     Enum_Loc;
        Str_30          Str_1_Loc;
        Str_30          Str_2_Loc;

  /* Initializations */
#if 0
  Next_Ptr_Glob = (Rec_Pointer) malloc (sizeof (Rec_Type));
  Ptr_Glob = (Rec_Pointer) malloc (sizeof (Rec_Type));
#else
  static Rec_Type glob1, glob2;
  Next_Ptr_Glob = &glob1;
  Ptr_Glob = &glob2;
```

```
120         {
            main:
> 000005cc:   addi sp,sp,-192
000005d0:   sw ra,188(sp)
148           strcpy (Ptr_Glob->variant.var_1.Str_Comp,
000005d4:   addi ra,gp,1100
000005d8:   lui a6,0x2
120         {
000005dc:   sw s1,180(sp)
000005e0:   sw s7,156(sp)
000005e4:   sw s8,152(sp)
150           strcpy (Str_1_Loc, "DHRYSTONE PROGRAM, 1'ST STRING");
000005e8:   lui a2,0x2
141           Ptr_Glob = &glob2;
000005ec:   addi s8,ra,48
148           strcpy (Ptr_Glob->variant.var_1.Str_Comp,
000005f0:   lw s7,-152(a6)
000005f4:   addi a3,a6,-152
150           strcpy (Str_1_Loc, "DHRYSTONE PROGRAM, 1'ST STRING");
000005f8:   addi a5,a2,1328
141           Ptr_Glob = &glob2;
000005fc:   sw s8,1096(gp)
146           Ptr_Glob->variant.var_1.Enum_Comp     = Ident_3;
00000600:   li s0,2
120         {
00000604:   sw s0,184(sp)
00000608:   sw s2,176(sp)
0000060c:   sw s3,172(sp)
00000610:   sw s5,164(sp)
00000614:   sw s6,160(sp)
```

eclipse-dhry21 Default [GDB Hardware Debugging] /opt/riscv32g/bin/riscv32-unknown-elf-gdb (7.11.50.20160212)

```
Program received signal SIGINT, Interrupt.

main (argc=0, argv=0x0) at src/dhry_1.c:120
120     {
```

JTAG-based debug solutions:

Supports: Segger J-link, Olimex ARM-USB-OCD family, Digilink JTAG-HS2, Lauterbach trace32, more vendors in 2019

✓ ✓ ✓ ✓

Also available:

- LLVM 5.0
- CompCert 3.1
- 3rd party vendors in 2019

Simulators:

- Qemu
- Spike
- 3rd party vendors

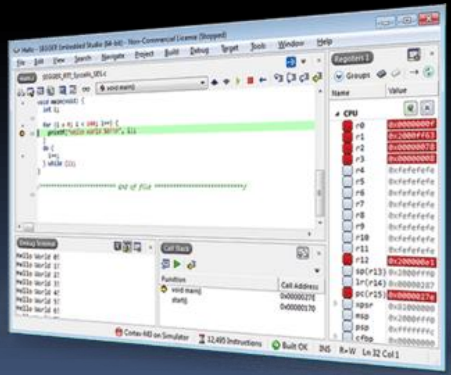
Number of 3rd party tools support SCRx cores

- Lauterbach Trace32

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

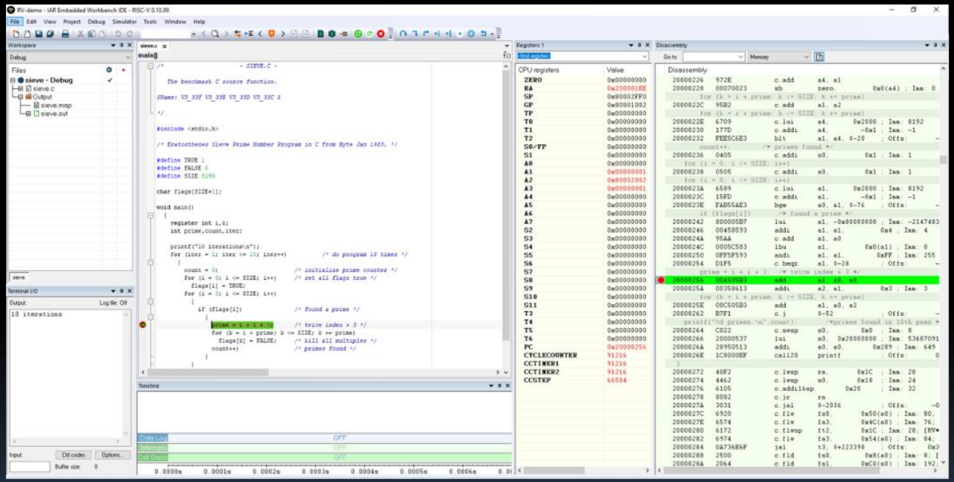
- Segger Embedded Studio

https://wiki.segger.com/Syntacore_SCR1_SDK_Arty



- IAR Embedded Workbench
- NEW!

https://www.iar.com/iar-embedded-workbench/#!?architecture=RISC-V

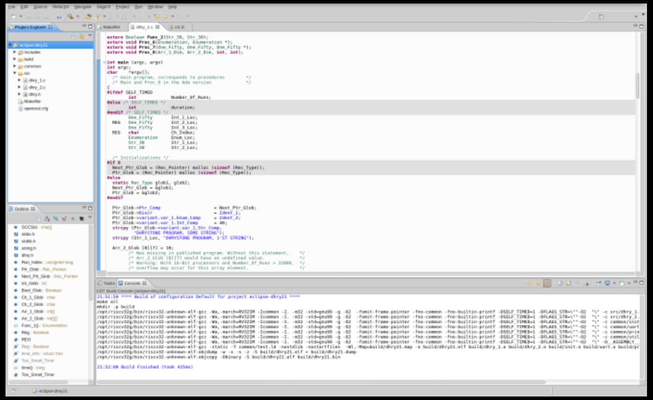


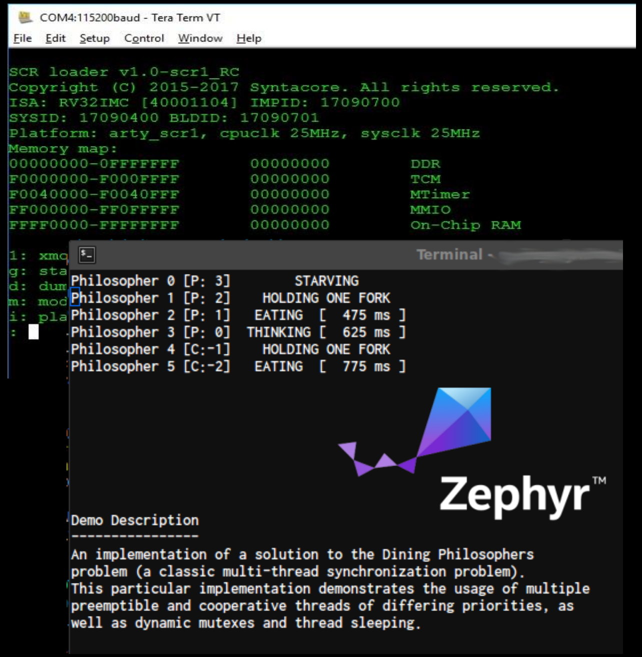
…more in 2019

SCRx SDK

Stable Eclipse/gcc based toolchain with IDE:

- GCC 8.1
- 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, cpuclk 25MHz, sysclk 25MHz
Memory map:
00000000-0FFFFFFF        00000000        DDR
F0000000-F000FFFF        00000000        TCM
F0040000-F0040FFF        00000000        MTimer
FF000000-FF0FFFFF        00000000        MMIO
FFFF0000-FFFFFFFF        00000000        On-Chip RAM

1: xmo
g: sta
d: dum
m: mod
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 ]

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.
```

HW platform based on standard FPGA dev.kits

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

User
uide
017-12-14
tacore.com>

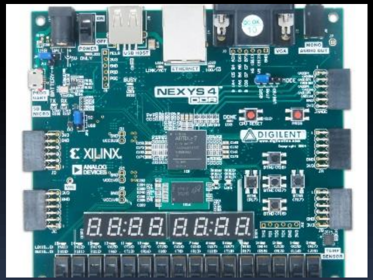
SCRx development tools. Brief installation and usage guide

v0.27, 2017-12-15

© Syntacore <info@syntacore.com>

SW:

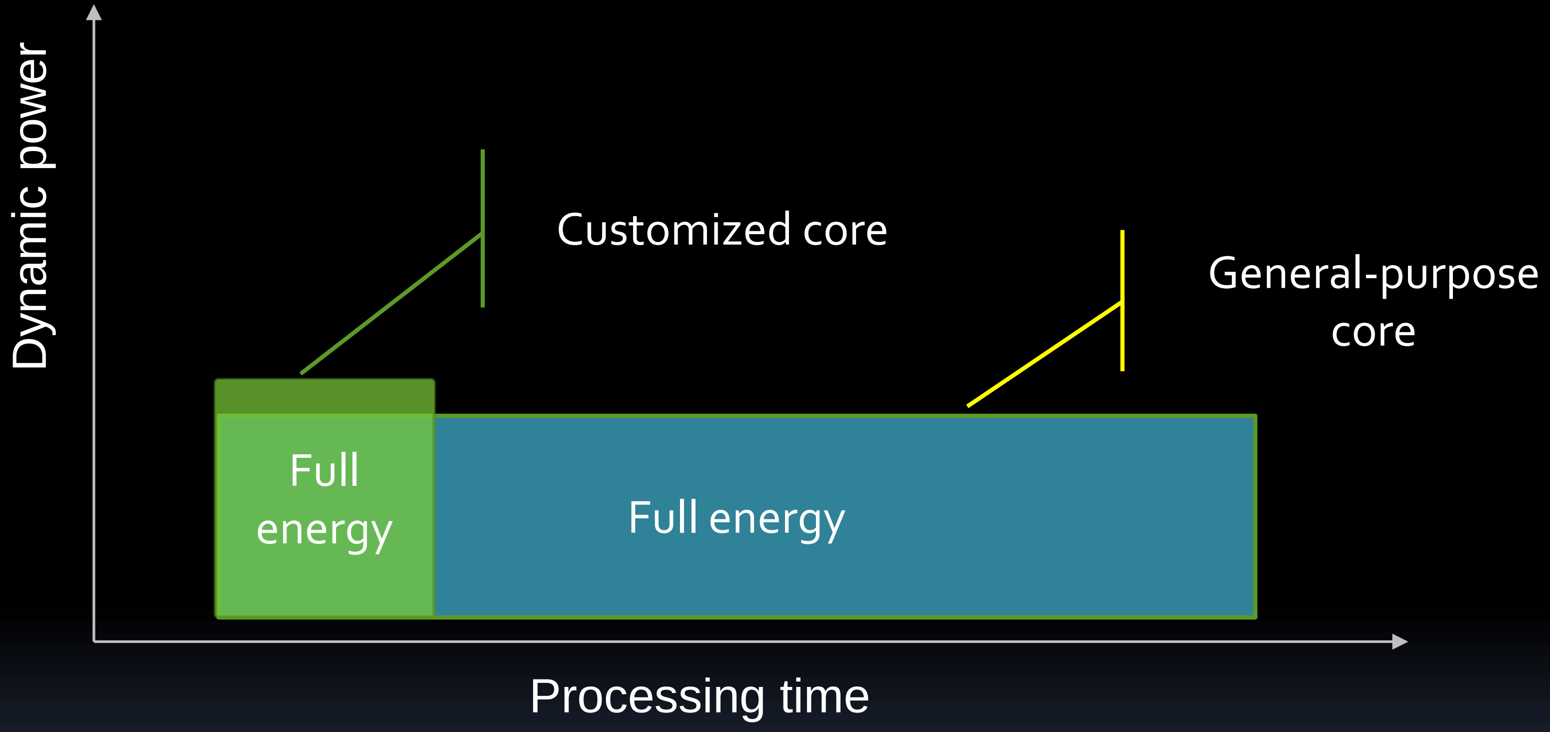
- Bootloader
- OS: Zephyr/FreeRTOS/Linux
- Application samples, tests, benchmarks



https://www.xilinx.com/products/boards-and-kits/1-6olhwl.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

		Encoding throughput, MB/s			Normalized per MHz, MB/s			RV32G + custom speed-up		
Platform	Fmax, MHz	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 evaluation and tapeout

Syntacore™
Custom cores and tools

Summary

- Syntacore offers high-quality RISC-V compatible CPU IP
  - Founding member, fully focused on RISC-V since 2015
  - Silicon-proven and shipping in mass-production
  - Turnkey IP customization services
    - with full tools/compiler support

Visit our booth for SCRx demos, including RISC-V silicon

Sk IT

Syntacore™

Custom cores and tools

Thank you!