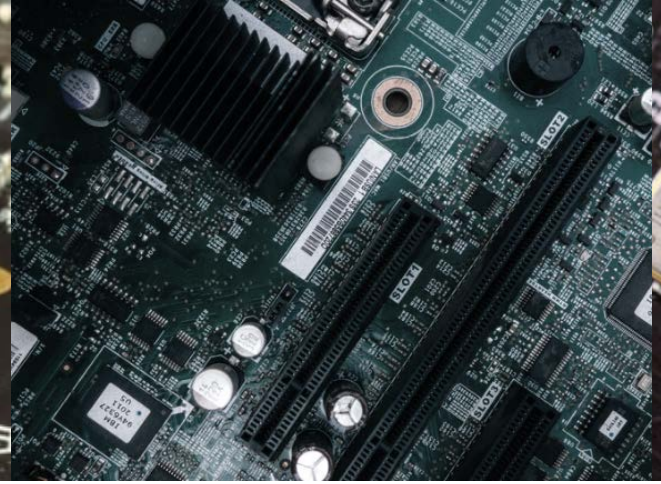
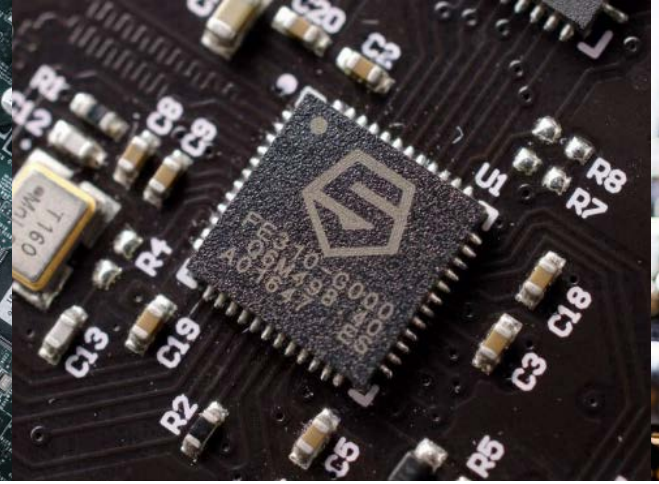
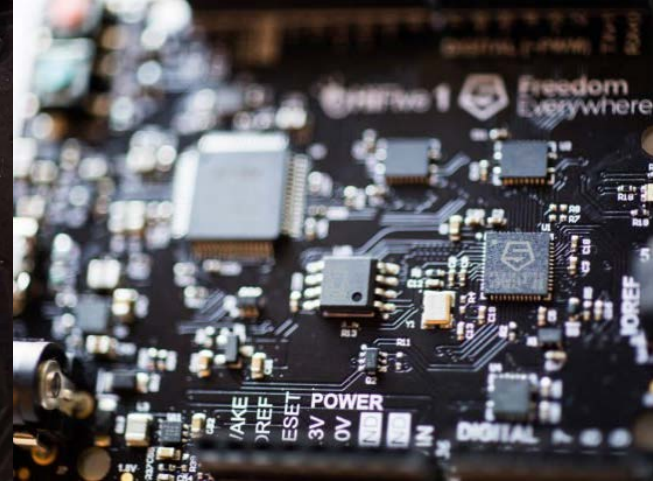
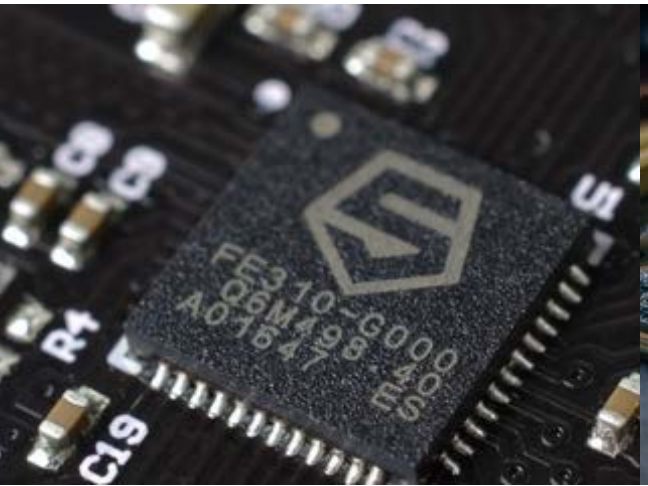
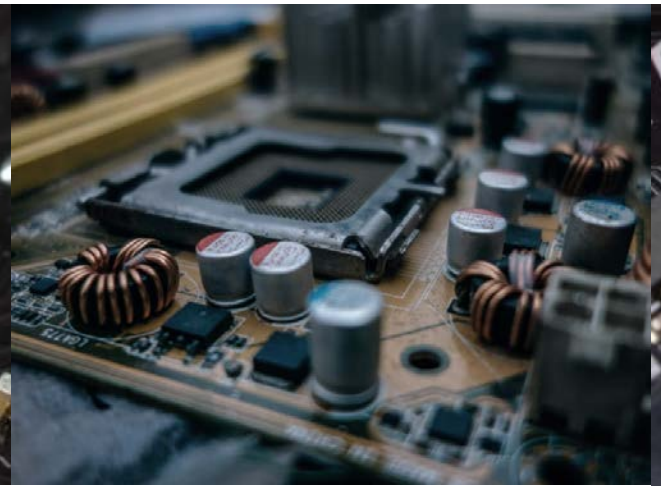
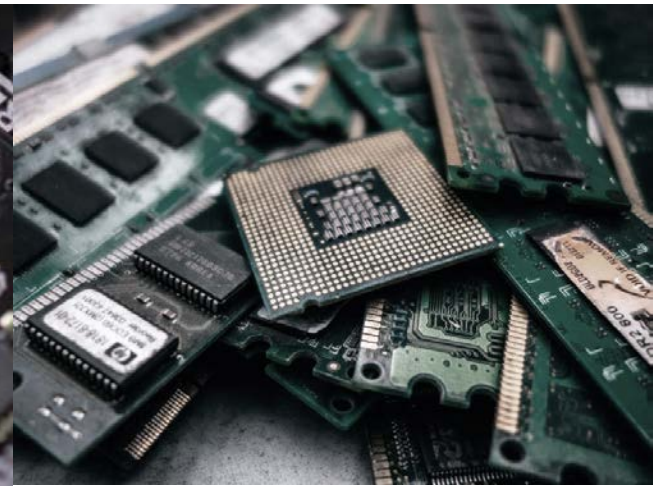





DTS INSIGHT

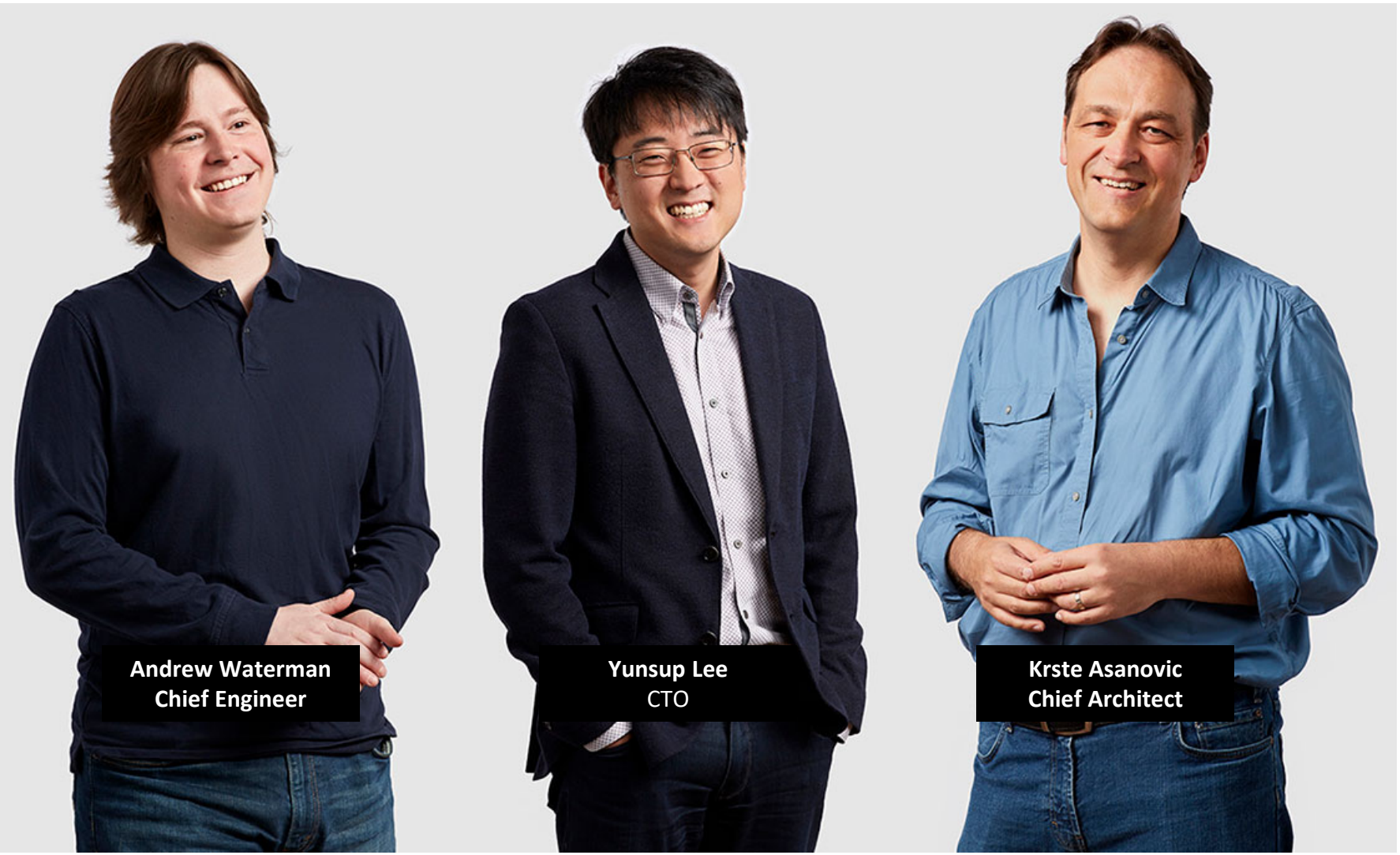
Our insight, your value

SiFive



SiFive Overview



We invented RISC-V

SiFive's founders are the same UC Berkeley professor and PhDs who invented and have been leading the commercial implementation of the RISC-V Instruction Set Architecture (ISA) since 2010

"SiFive Recognized as Most Respected Private Semiconductor Company"

About SiFive

SiFive

Worldwide Presence
15+ Offices
500+ Employees (420+ Engineers)
350+ Tapeouts(OpenFive)
150+ Design Win

World-class expertise
- Inventors of RISC-V
- Ideas to Silicon
- RTL Design & Verification
- FPGA & Emulation
- Physical Design
- Wafer Fabrication
- Board Design
- Full Silicon Validation

SiFive社日本国内代理店 DTSインサイトのご提供領域

ハードウェア/ファームウェア組込などの得意領域を活かし、
お客様のあらゆるニーズにお応えする**One Stop Solution**をご提案いたします

Mainstream
組込みSW開発

- 組込みソフトウェア開発
- ハードウェア開発
  - （回路設計、PCB設計、機構・筐体設計から試作製造、量産）
- デバッガ(ICEなど)
- データモニターツール
- 動的テスト/解析ツール
- ソフトウェア構造分析ツール
- トレーサビリティ管理ツール

New Business
LSIデザインサービス

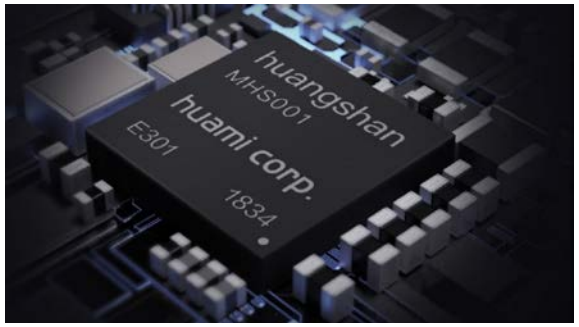
- **LSIデザインサービス**
  - **株式会社SSC様と協業**
- **RISC-V開発ボード環境**
- **RISC-Vデバッガ**
- **FW移行サービス**

プロセッサIP（RISC-V）

RISC-V を題材にスタート

Products Powered by SiFive Cores (Few Examples)

Wearable SoC Huangshan-1

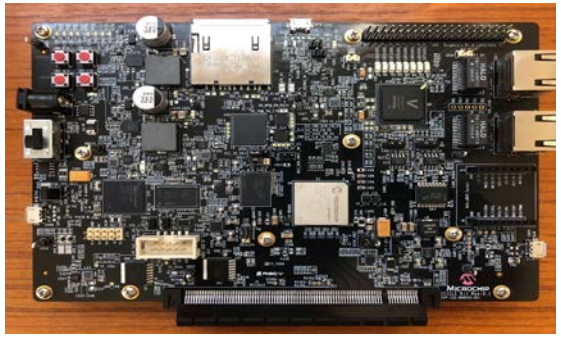


huami

NVMe SSD Controller FC3081

FADU

PolarFire SoC Icicle Kit



MICROCHIP

Rapid adoption of SiFive Core IP from the Edge to the Core

Products Powered by SiFive Cores (Few Examples)

Cayenne family USB-C video interface	TERALYNX Ethernet Switch Silicon	HyperX memory network processor
Synaptics®	Innovium®	Coherent Logix™
		

RISC-V Momentum

ANANDTECH

Home > CPUs

Samsung to Use SiFive RISC-V Cores for SoCs, Automotive, 5G Applications

by Anton Shilov on December 12, 2019 11:00 AM EST

18 Comments | Add A Comment

golem.de
IT-NEWS FÜR PROFIS

SMARTPHONES

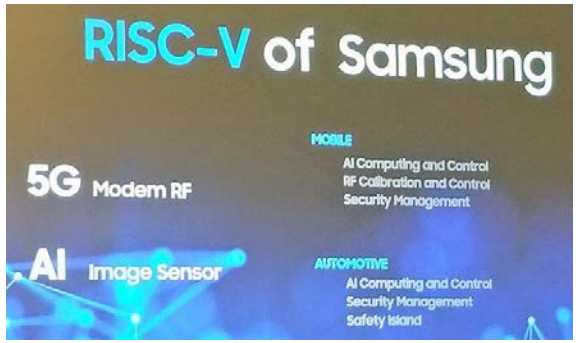
Qualcomm uses RISC-V in Snapdragon chips

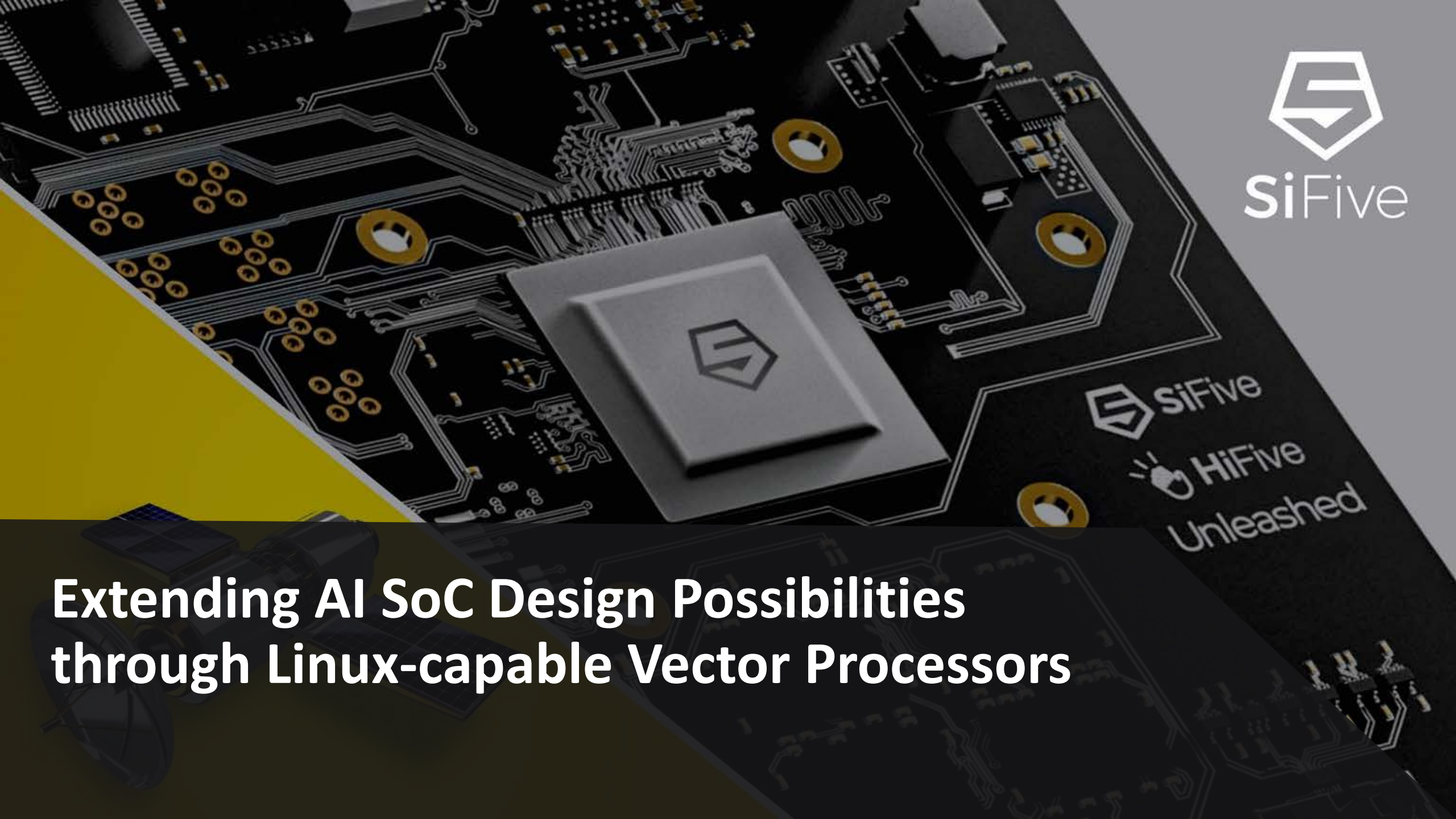
One of the largest SoC developers is now relying on RISC-V : Qualcomm integrates cores with the open instruction set architecture for embedded use in current and future Snapdragon chips.

January 24, 2020, 3:12 p.m., Marc Sauter

腾讯网

New H3C Semiconductor successfully adopts RISC-V U7 multi-core processor from Saifang Technology





Extending AI SoC Design Possibilities through Linux-capable Vector Processors

- **AI SoC Design Hardware and Software Challenges**

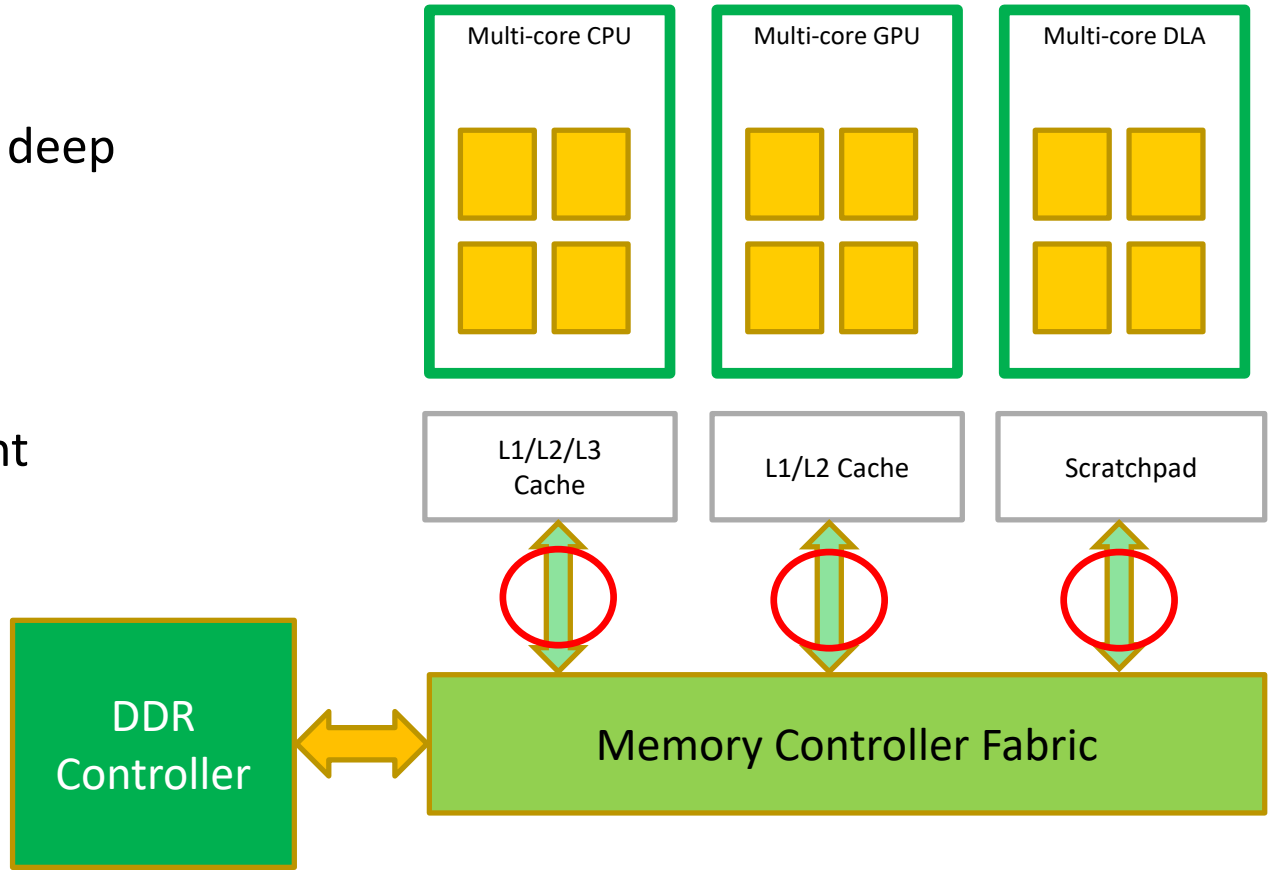
- **SiFive Solution implementing RISC-V with Vectors (RVV)**
  - Enables Efficient SoC Implementation
  - Streamlines Software Development
  - RVV More Efficient than SIMD
  - Autovectorizing C++ Compiler

- **SiFive Intelligence Core IP**
  - VIU7 Series overview
  - VIU7 Microarchitecture
  - Benchmarking

- **Summary**

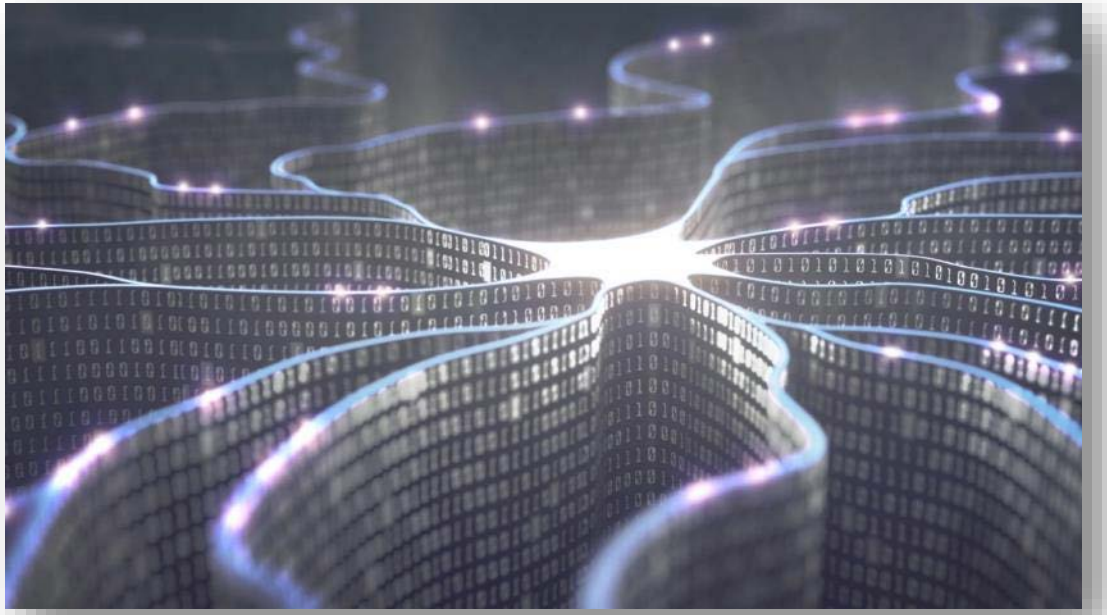
AI SoC Hardware Challenge

- Multiple bandwidth-hungry subsystems
- Multiple proprietary instruction sets for deep learning accelerators
- Poor memory-bandwidth utilization
- Complex memory crossbar
- Power optimization difficult to implement



AI SoC Software Challenge

- Multiple **proprietary** accelerator instruction sets
- Multiple **proprietary** API's and outdated libraries
- New techniques such as deep compression or Winograd transform not supported
- Memory hierarchy doesn't match new algorithm requirements
- Poor compiler support, requires programming at low level
- Power optimization difficult to implement

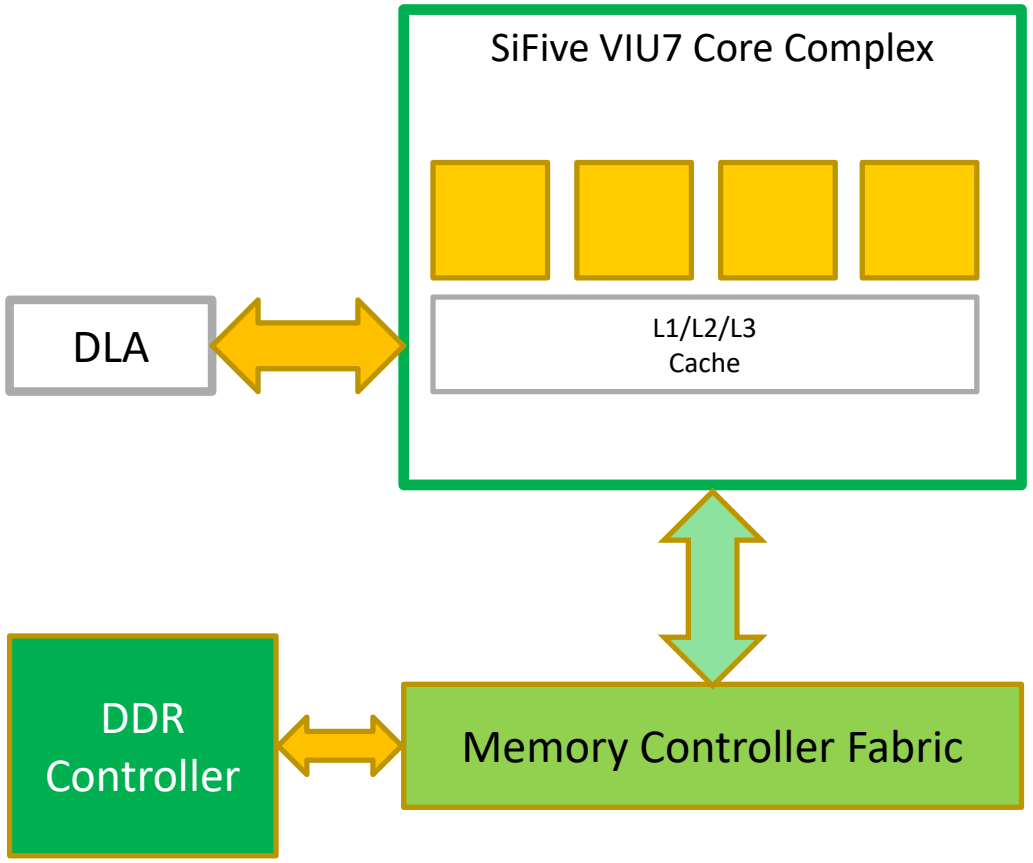


SiFive Solution

- Based on an open industry standard ISA (RVV v1.0) to prevent vendor lock-in and enable rich ecosystem for AI
- **Scalable performance** to meet AI processing requirements from extremely low power to high performance compute applications
- **Multi-core** complex that can integrate Linux capable or Real-time cores with accelerators and scale to TOPs performance
- Implements efficient memory hierarchy that maximizes data reuse
- Single ISA enables simple and efficient programming model that allows tuning algorithms for both performance and low power
- **Comprehensive security** support enabled by SiFive Shield
- SiFive Advanced Trace and Debug Solution

RISC-V with Vectors (RVV) Enables Efficient SoC Implementation

- A GPU or DSP is no longer required for most applications
- A DLA can be **tightly coupled to the RISC-V core** complex for TeraOP class performance
- Bandwidth is utilized more efficiently by **sharing data in the L3 cache**
- Simplified interconnect, area-efficient implementation, and lower power.
- RVV provides a **binary-compatible platform** starting from tiny embedded cores to supercomputer-class multicore clusters



RVV v1.0 Streamlines Software Development

- **Single instruction set architecture** greatly simplifies software development
- **Vector-length-agnostic** architecture allows library and application investment to be **reused** across multiple generations of cores
- **Same binary** runs on tiny low-power cores for wearables and high-performance supercomputer-class cores
- SiFive **Linux-capable VIU7** cores with **48-bit virtual memory support** large datasets and powerful Linux-based frameworks and tools
- **Leverage the open-source** community for application development including exciting projects such as OpenBLAS. Several Deep Learning Frameworks are also under active development, including TensorFlow Lite.
- RVV v1.0 will become the industry standard

RVV More Efficient than SIMD

Even the simple DAXPY function (below) illustrates this*.

```
void daxpy(size_t n, double a, const double x[], double y[])
{
for (size_t i = 0; i < n; i++) {
  y[i] = a*x[i] + y[i];
}
}
```

ISA	*MIPS-32 MSA*	*IA-32 AVX2*	*RV32V*
Instructions (static)	22	29	13
Instructions per Main Loop	7	6	10
Bookkeeping Instructions	15	23	3
Results per Main Loop	2	4	64
Instructions (dynamic n=1000)	3511	1517	163

* https://www.sigarch.org/simd-instructions-considered-harmful/
by David Patterson and Andrew Waterman, Sep 18, 2017

Based on earlier version of RISC-V Vector ISA standard.

Autovectorizing C++ Compiler

```
void saxpy_golden(size_t n, const float a, const float *x, float *y) {
  for (size_t i = 0; i < n; ++i) {
    y[i] = a * x[i] + y[i];
  }
}
```

← **Generic source code**

Inventor and lead developers of LLVM work at SiFive!

LLVM Output →

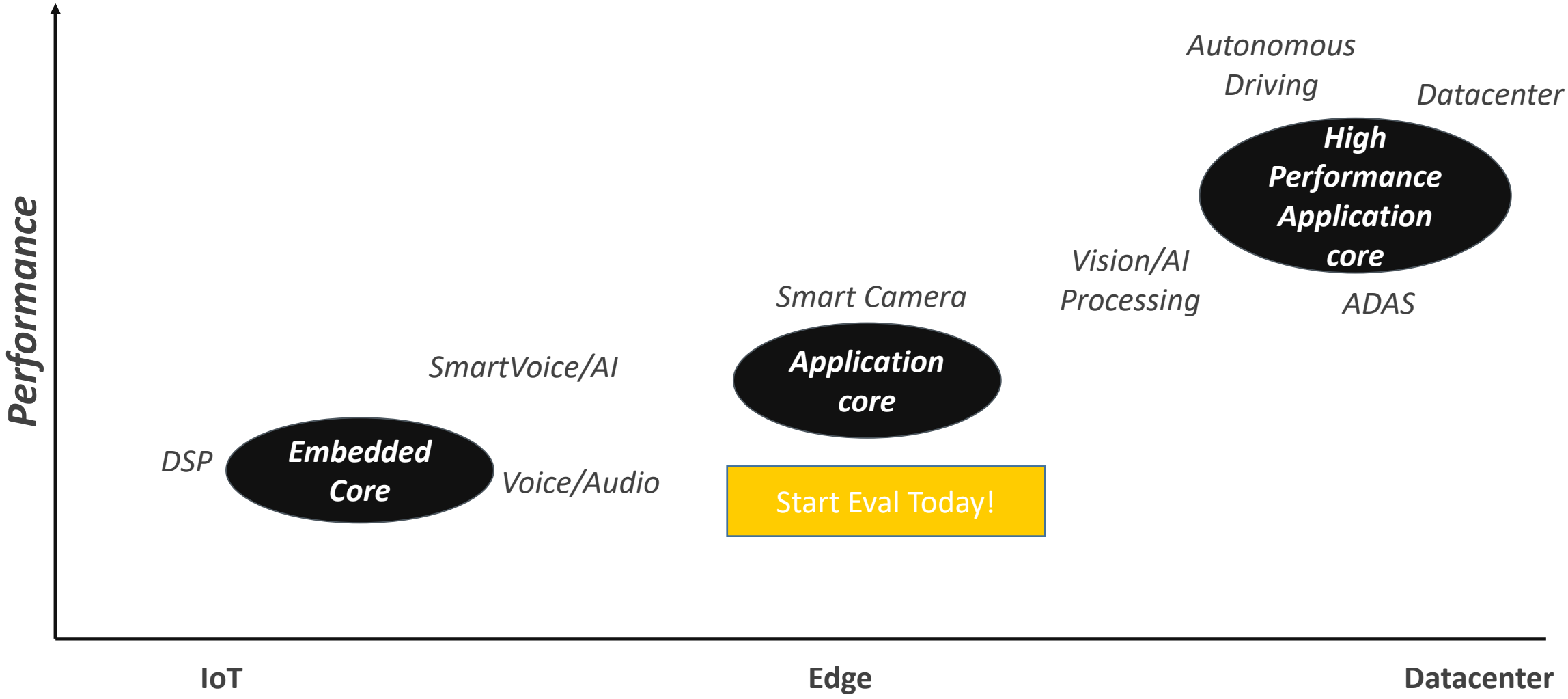
```
saxpy_golden:                   # @saxpy_golden
# %bb.0:                        # %entry
                beqz            a0, .LBB0_3
# %bb.1:                        # %vector.ph
                mv              a3, zero
                vsetvli         a4, zero, e32,m1,tu,mu
                vfmv.v.f        v1, fa0
.LBB0_2:                        # %vector.body
                        # =>This Inner Loop Header: Depth=1
                slli            a4, a3, 2
                add             a6, a1, a4
                sub             a5, a0, a3
                vsetvli         a5, a5, e32,m1,tu,mu
                vle32.v         v2, (a6)
                add             a4, a4, a2
                vle32.v         v3, (a4)
                vfmul.vv        v4, v2, v1
                vfadd.vv        v2, v4, v3
                add             a3, a3, a5
                vse32.v         v2, (a4)
                bne             a3, a0, .LBB0_2
.LBB0_3:                        # %for.cond.cleanup
                ret
```

SiFive is leading the effort to bring state of the art compiler technology to RVV.

Generic C code can be automatically vectorized to take advantage of vector processing.

Further optimization work such as multiply-add fusion is progress. This example is load/ store bound so is fairly optimal.

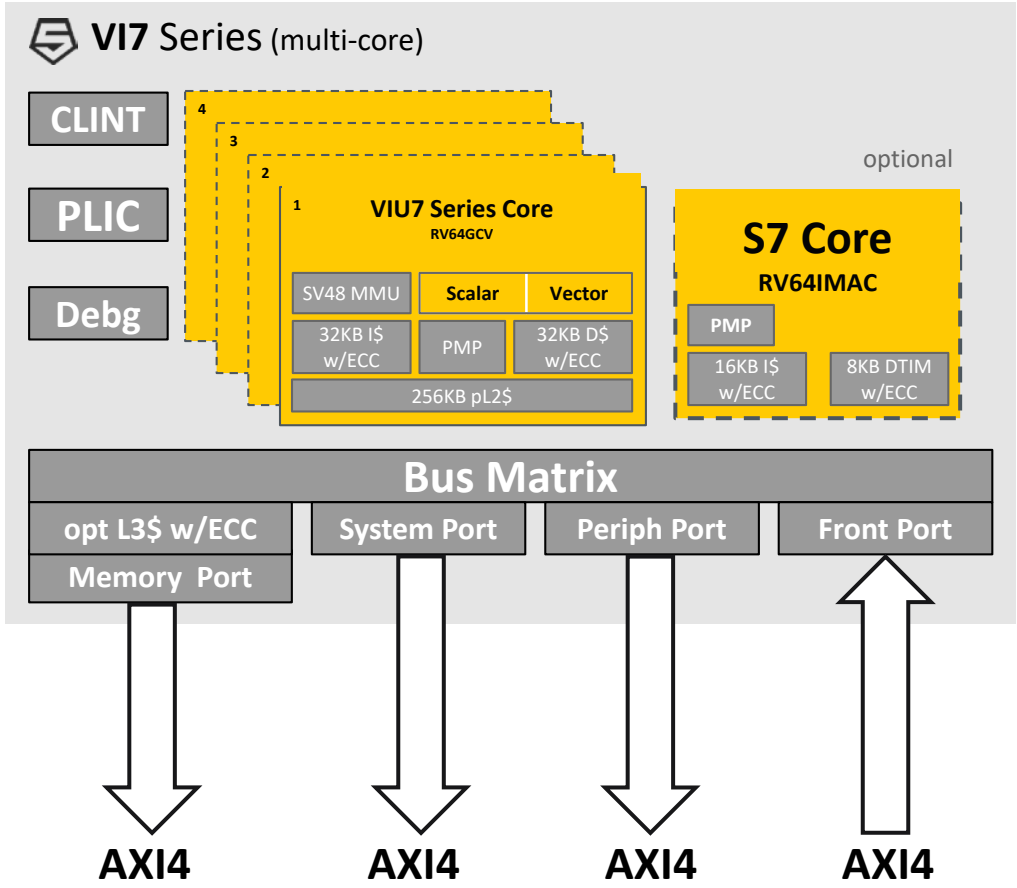
SiFive Intelligence Roadmap



Introducing The SiFive Intelligence VIU7 Series

Key VIU7 Series attributes:

- Base core derived from SiFive U7 series
  - **RV64GCV** ISA with Sv39 or Sv48 support
  - 8-stage **dual-issue** in-order pipeline with decoupled vector unit
  - Machine, Supervisor, and User privilege modes
  - Full TLB-based MMU for Linux-based environments
  - Single and Coherent multi-core IP platform options
- RISC-V Vector ISA extensions v1.0
  - Floating-point, fixed-point, and integer data types
  - Vector computation up to 256b/cycle on 8b to 64b datatypes
- Other key attributes support:
  - Configurable S7 core(s) can support a variety of uses:
    - System boot/monitor, Sensor Hub/Fusion, Security Co-Processor
  - Private L2$ per core, optional shared L3$
  - Up to 16 PMP regions with 4KB granularity
  - Supports SiFive WorldGuard security
  - Leverages existing SiFive Insight Advanced Trace & Debug hardware solutions

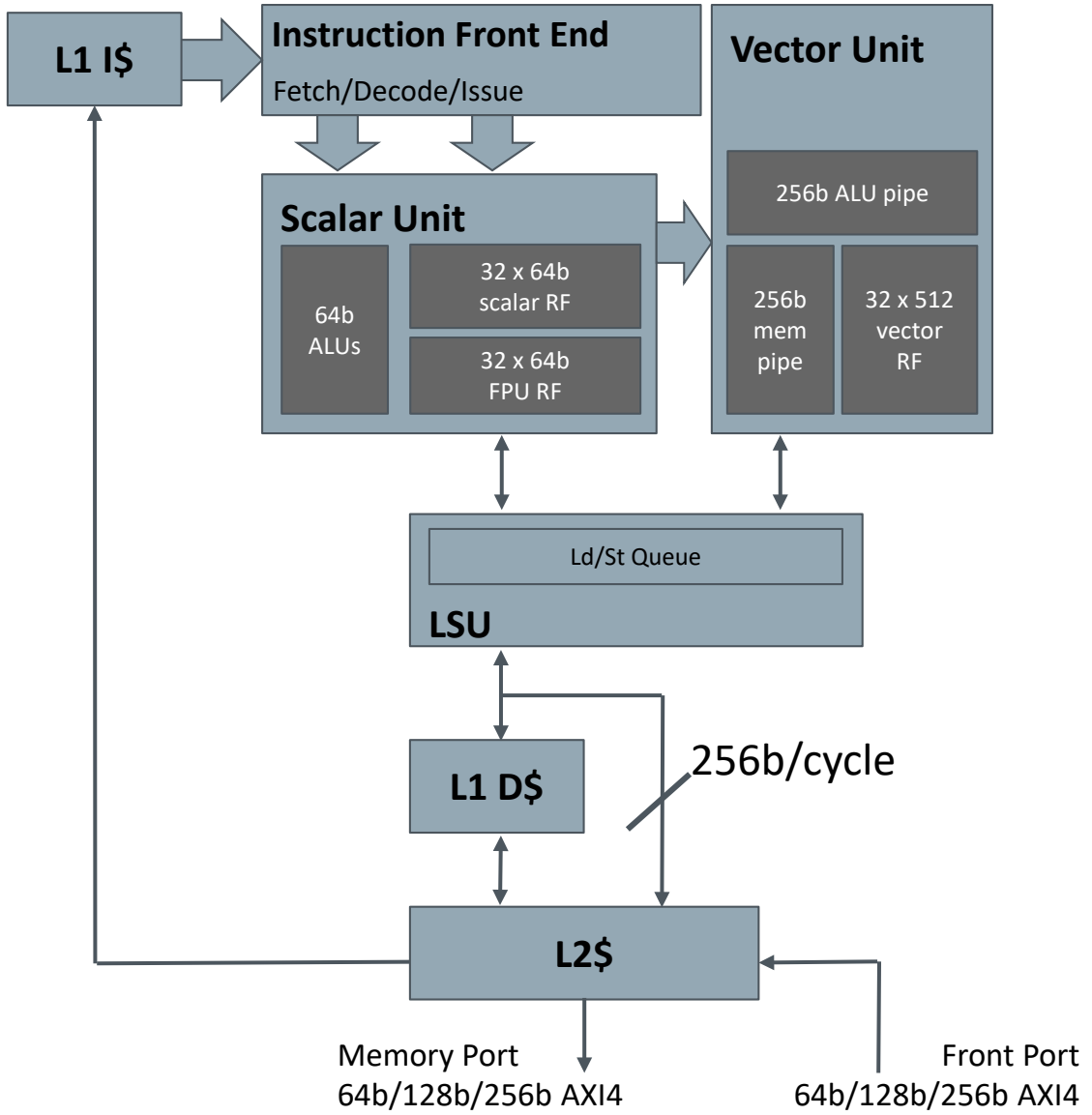


VIU7-256 High-Level Functional View

Single-core view
Dual-issue scalar unit runs concurrently with vector unit

Key vector unit attributes:

- VLEN = 512
- DLEN = 256 (datapath width)
- ELEN = 64
- 2 cycles to complete one 512b vector operation (@LMUL = 1)
- Separate memory and ALU pipelines for concurrent 256b operation
- Chaining of vector operations
- Vector ALU
  - VIU7-256 has a 256b ALU
  - Does 4x64b, 8x32b, 16x16b or 32x8b ops/cycle
- Decoupled Vector loads/stores
- Loads/Stores are 256b/cycle
- Treat L2$ as primary memory, no load-to-use penalty
- Can load from L1$, but initiate L2$ load in parallel to minimize impact of L1$ miss



VIU7 Eval System

Xilinx VCU118 FPGA Platform

- Single and multicore VIU7 support
- 4GB DDR DRAM, SD card and other peripherals
- Linux or Embedded runtime environment
- JTAG debug, Real-time trace and hardware profiling

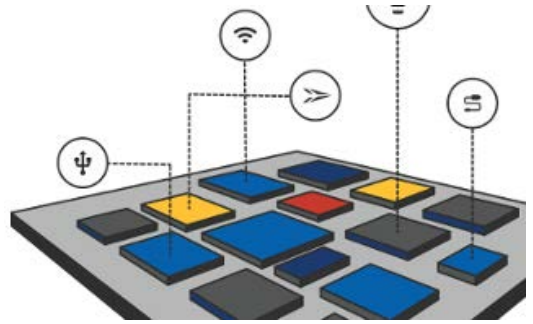
SiFive Freedom Studio Integrated Development Environment

LLVM compiler

Vector intrinsics

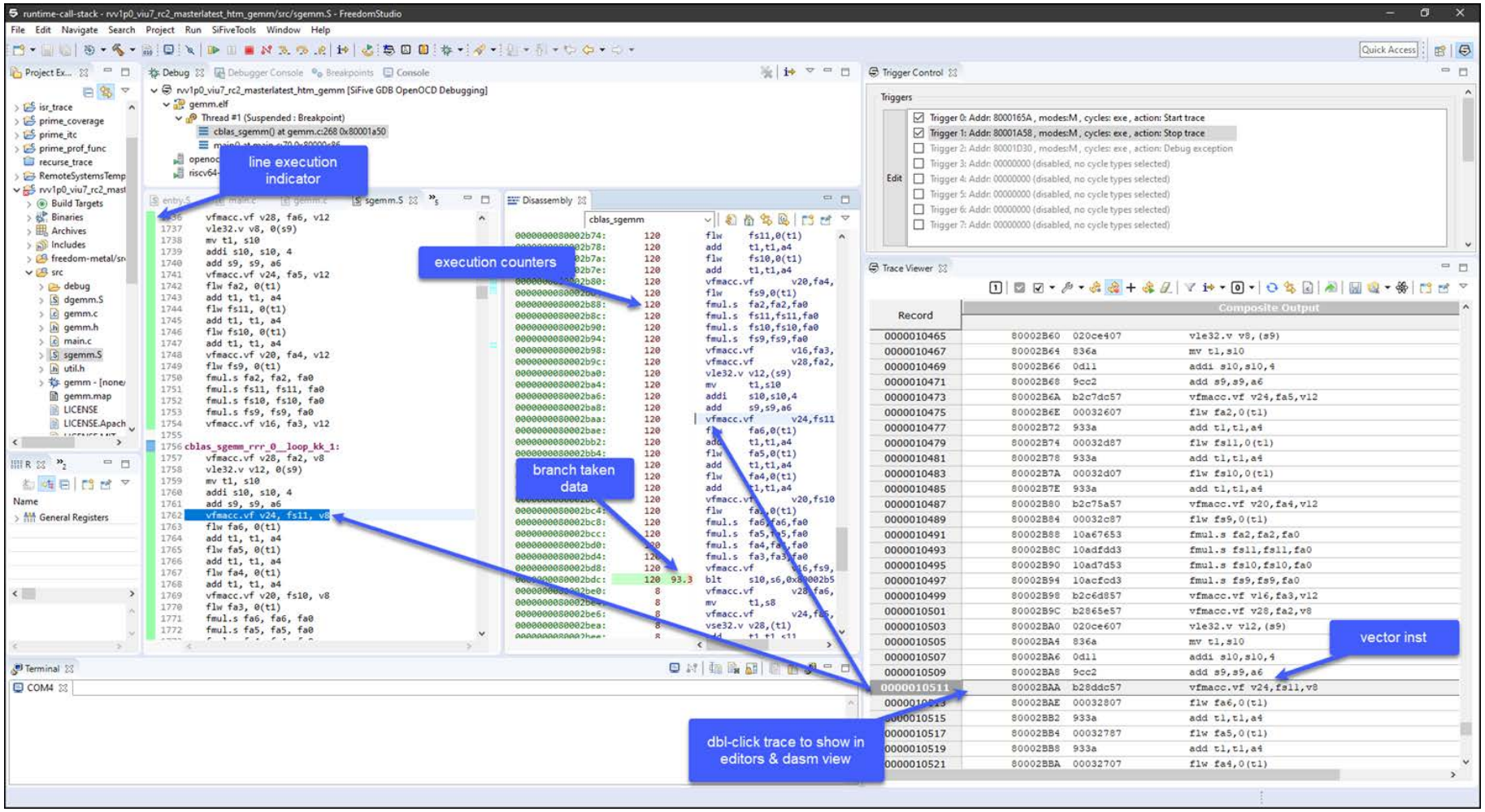
QEMU target

System-C cycle-accurate model

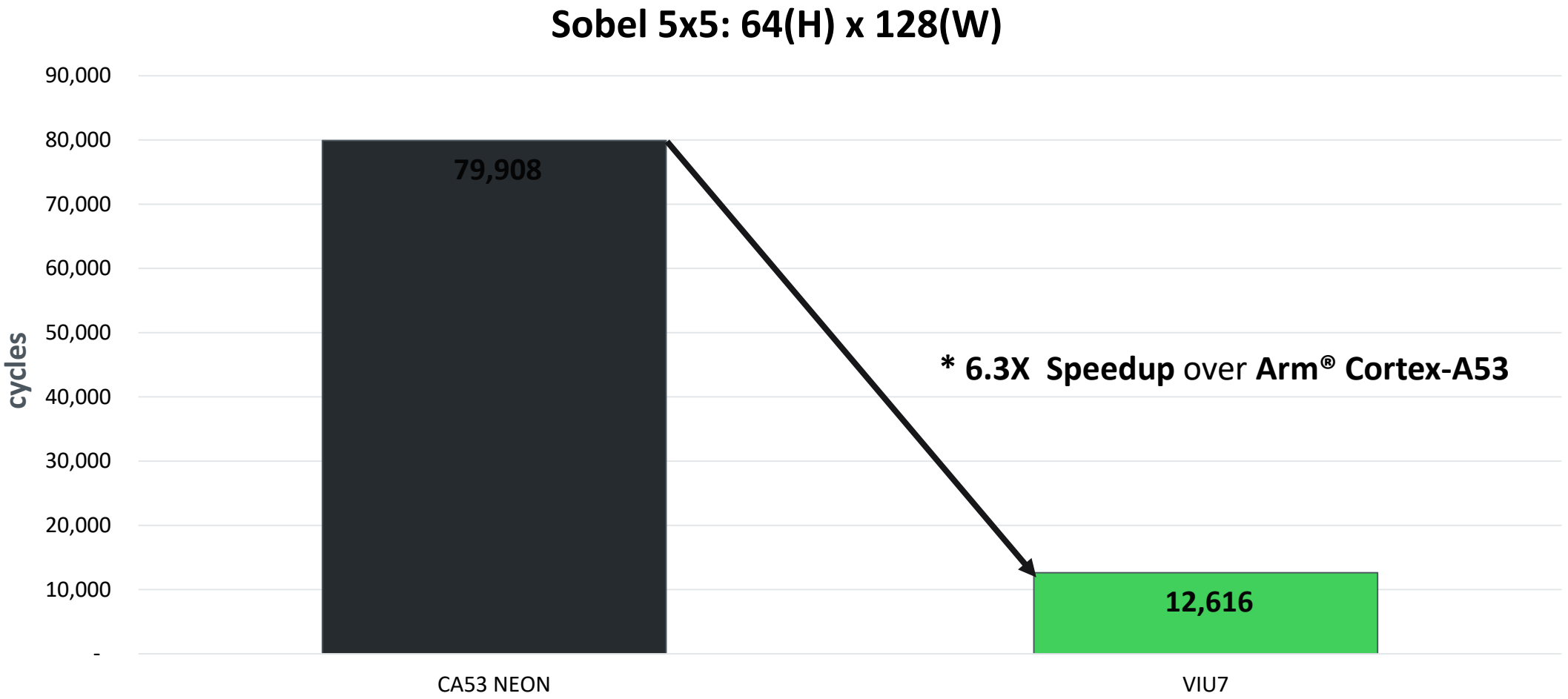


Freedom Studio

SiFive Insight Advanced Trace and Debug

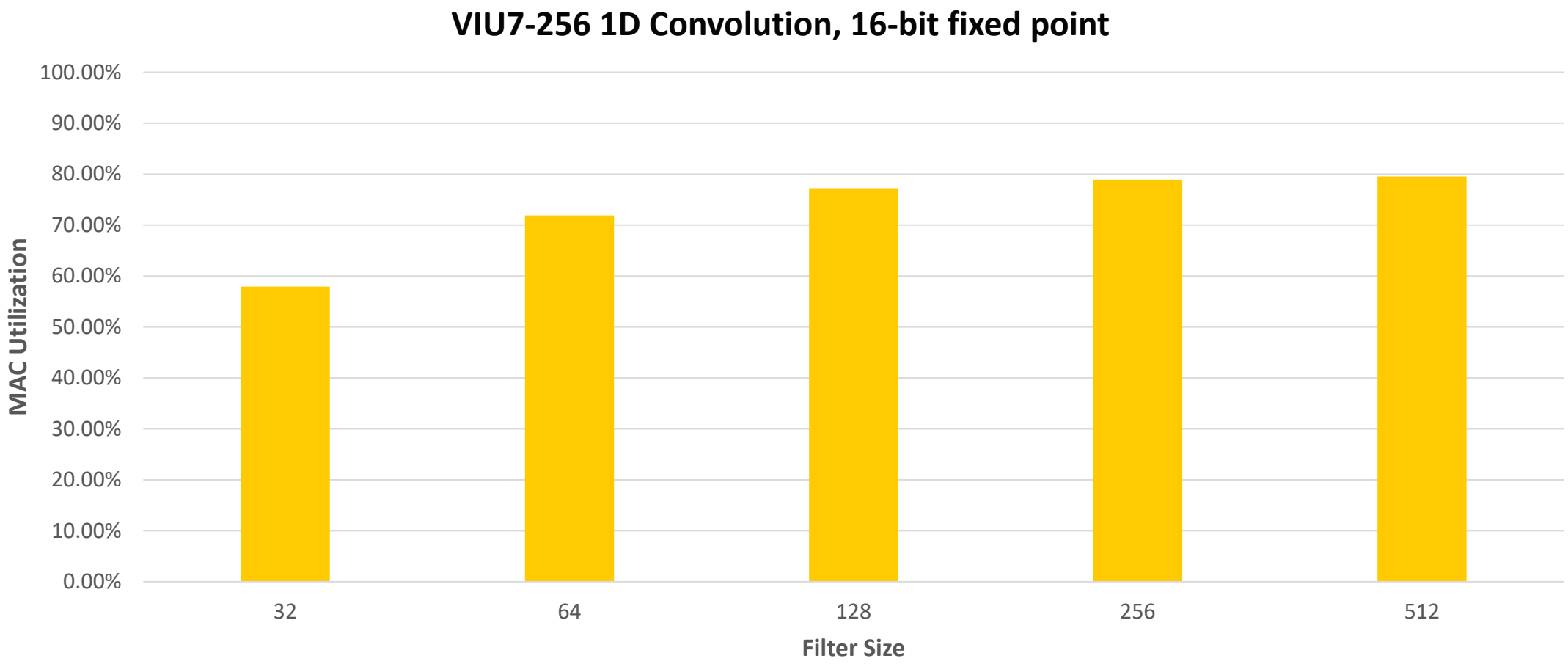


VIU7 Computer Vision Performance



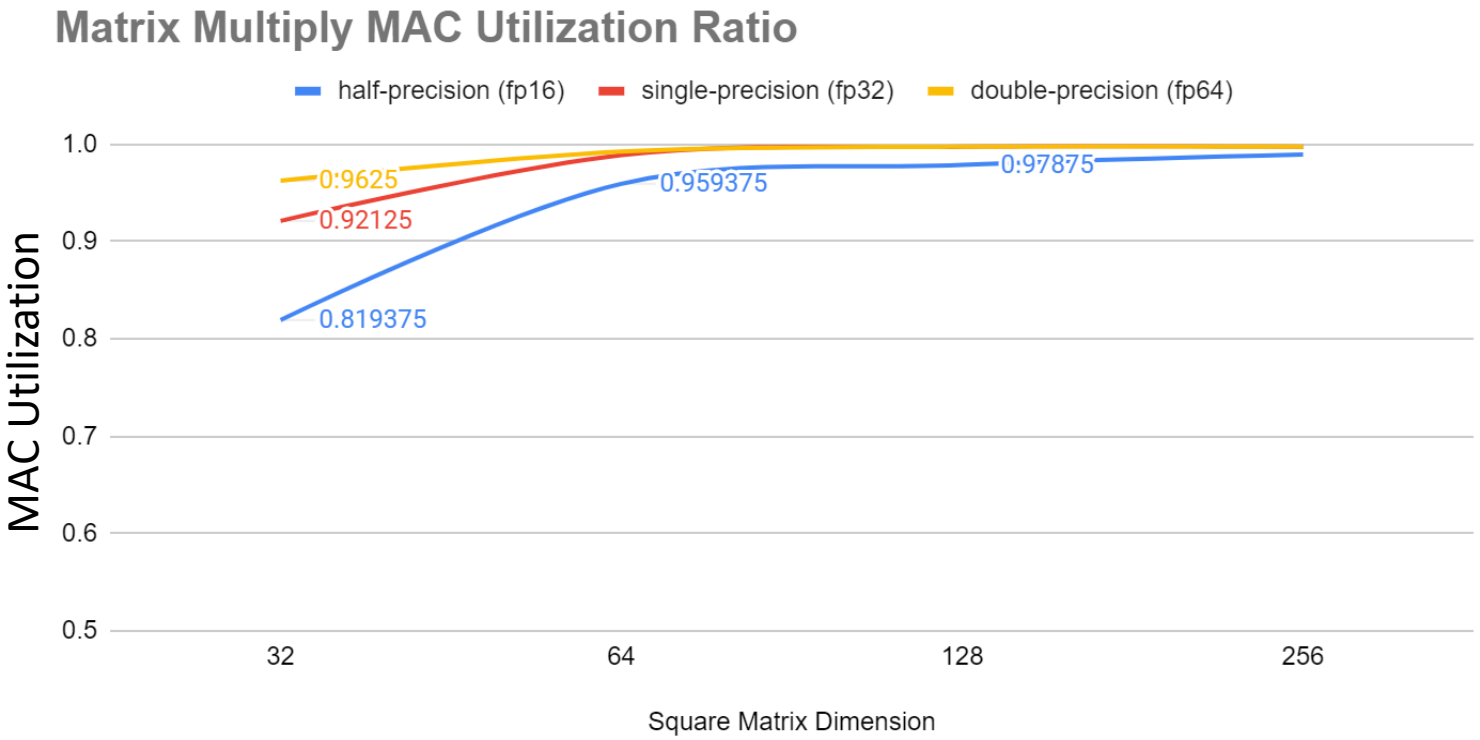
*Performance measured on Raspberry Pi-3

VIU7 Delivers High Utilization for CNN Workloads



VIU7 Delivers High Utilization for Matrix Multiply

- Compute *C* = *A* * *B*, where *A*, *B*, and *C* are *N*-by-*N* matrices
- Matrix entries are randomly sampled from [0.0, 1.0]
- Matrices are stored in row-major order



VIU7 parameters:

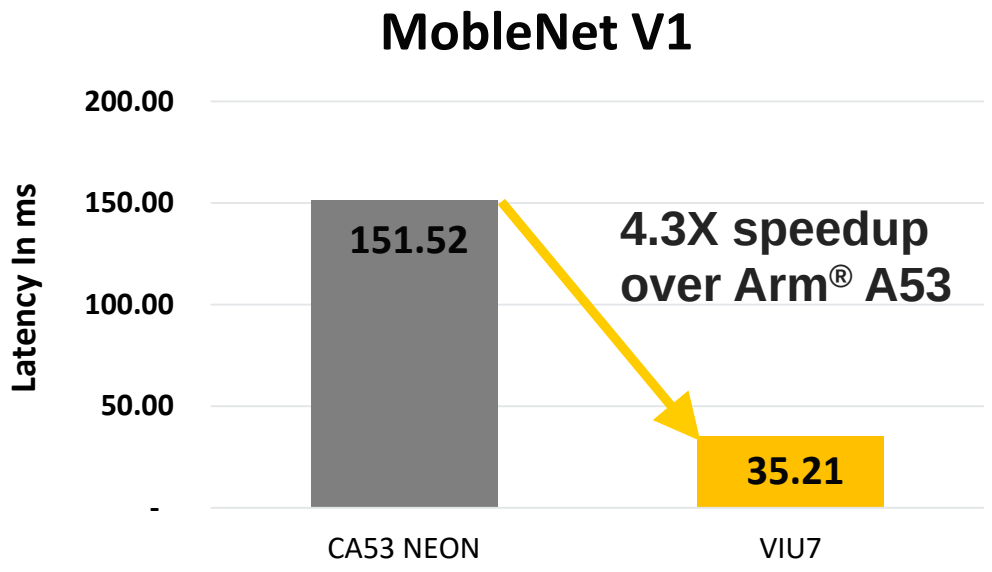
- 32KiB L1 instruction and data caches
- 512KiB L2 cache
- 256b datapath (memory and arithmetic)
- 512b vector length

Benchmarking procedure:

1. L1D$ flushed
2. Code run twice
3. Cycle counts reported for second run

VIU7 Delivers 4X MobileNetV1 Performance vs. Cortex-A53

Mobilenet_f16	CONV_2D	DEPTHWISE_CONV_2D	SOFTMAX	AVERAGE_POOL_2D	TOTAL
VIU7-256 cycles	53,223,073	9,985,008	136,013	24,993	63,376,073
VIU7-256 @1.8 GHz (ms)	29.57	5.55	0.08	0.01	**35.21
% of overall time	84%	16%	0%	0%	
ARM A53 @1.8 GHz (ms)					*151.52



*https://arxiv.org/pdf/1903.05898.pdf
**preliminary performance data measured on FPGA and scaled to 1.8 GHz. Real ASIC performance will depend on memory subsystem design.

The overall network structure of mobilenet v1:

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 2. Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

SiFive VIU7 Executing on FPGA is Faster Than A53!

Input Image

Computing Pioneer
Grace Hopper

```
# ./tflite_rvv/label_image -m ./data/mobilenet_v1/mobilenet_v1_1.0_224.f16.tflit
e -l ./data/mobilenet_v1/labels.txt -i ./data/grace_hopper.bmp -t 1
Loaded model ./data/mobilenet_v1/mobilenet_v1_1.0_224.f16.tflite
resolved reporter
Use gemm() directly for Conv2D
read_bmp: 398.031 ms
resize(init interpreter): 0.554 ms
resize(fill filter op input data): 447.107 ms
resize(tflite filter op): 697.569 ms
resize(copy filter's result to input buffer): 256 ms
resize: 1482.92 ms
invoked
average time: 2122.12 ms
0.852539: 653 653:military uniform
0.0544739: 907 907:Windsor tie
0.00804138: 466 466:bulletproof vest
0.00698471: 514 514:cornet, horn, trumpet, trump
0.00630951: 543 543:drumstick
#
```

```
Number of nodes executed: 31
============================== Summary by node type ==============================
                     [Node type]          [count]         [avg ms]           [avg %]
                         CONV_2D               15         1643.139           83.826%
                  DEPTHWISE_CONV_2D            13          312.029           15.918%
                         SOFTMAX                1            4.061            0.207%
                   AVERAGE_POOL_2D              1            0.738            0.038%
                         SQUEEZE                1            0.215            0.011%

Timings (microseconds): count=1 curr=1960182
Memory (bytes): count=0
31 nodes observed


#
```

Mobilenet_V1 benchmark running on Linux 5.9-rc4 using VCU118 FPGA platform

Summary

- RISC-V Vectors (RVV) turbocharges Linux application processors to handle demanding AI processing

- Scalable architecture allows **software compatibility** from tiny embedded cores to high-performance multi-core and multi-cluster solutions

- **SiFive Intelligence VIU7 Series**, the **first RVV v1.0-compliant** CPU Core IP
    - SiFive Mix+Match enables Linux and Real-time applications
    - Comprehensive security solutions enabled by SiFive Shield
    - SiFive Advanced Trace and Debug pre-integrated solutions

- Start designing your AI SoC today with **SiFive Intelligence** technology!