SiFive U54-MC Coreplex: Multicore, 64-bit Application Processor class RISC-V CPU

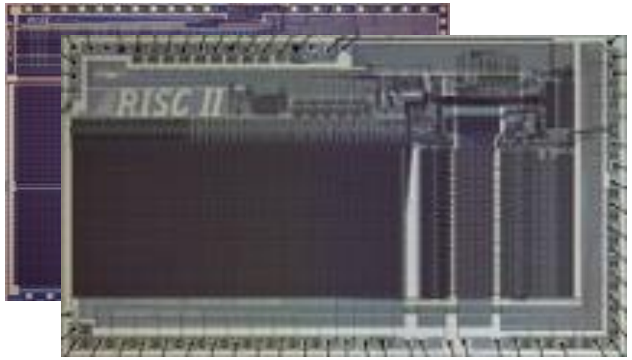
Jack Kang

VP of Product

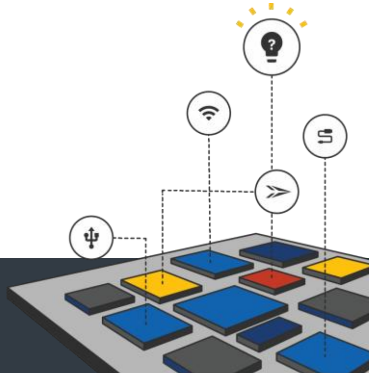
SiFive

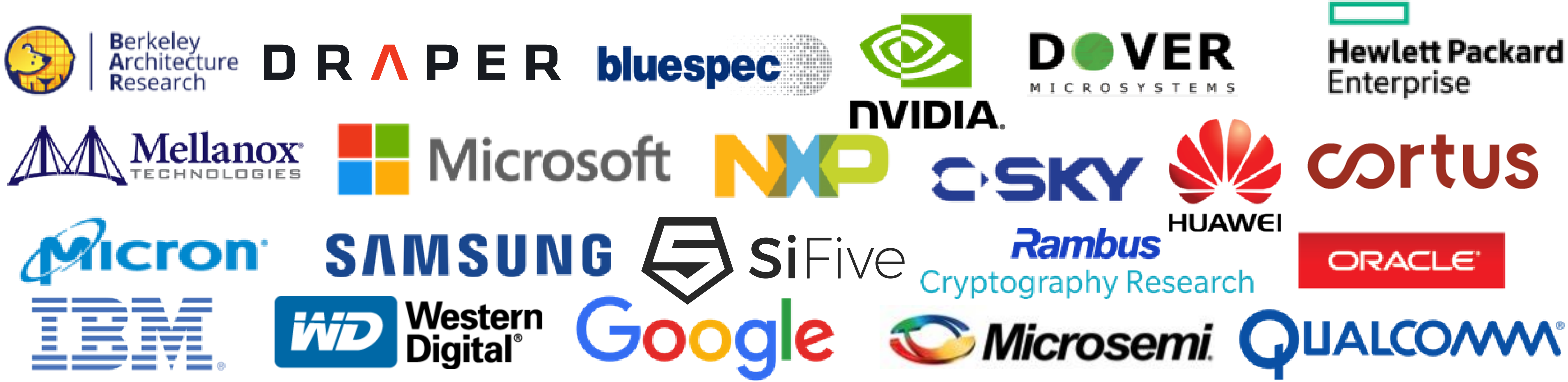
What is RISC-V

RISC-V is a high-quality, license-free, royalty-free ISA

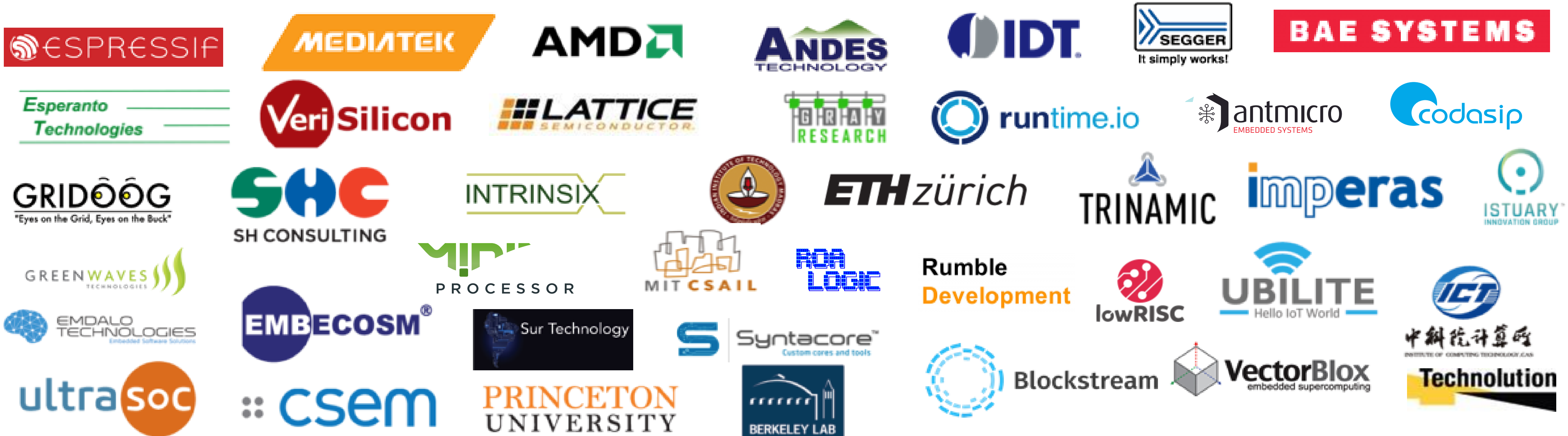


- 5th Generation RISC design from UC Berkeley
- A high-quality, license-free, royalty-free RISC ISA specification
- Standard maintained by non-profit RISC-V Foundation
- Multiple proprietary and open-source core implementations
- Supported by growing software ecosystem
  - binutils/gcc/FreeBSD mainlined, Linux/glibc submitted to upstream
- Appropriate for all levels of computing system, from microcontrollers to supercomputers





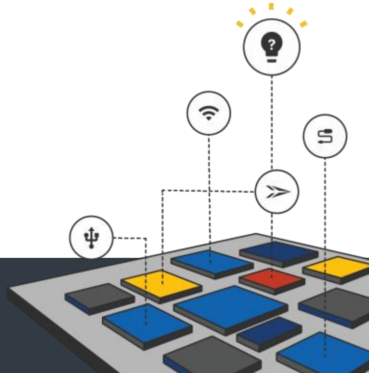
RISC-V Foundation: 65+ Members



How can RISC-V cover all levels of computing?

Embedded Markets are different from high-performance, application processors

- RISC-V is designed to be modular
  - Base ISA, with standard extensions
  - Custom, “Non-Standard” extensions allowed
- RISC-V is designed to support 32-bit, 64-bit, and 128-bit architectures
- RISC-V provides multiple operating modes (User, Supervisor, Machine)

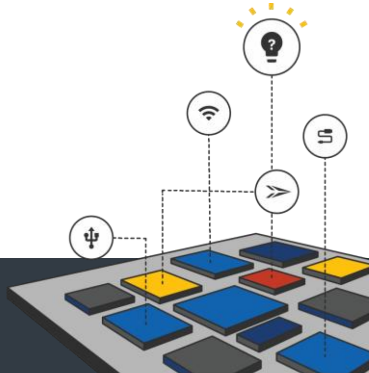


Modular, but standard, Extensions

- Extensions add instructions
- “I” for Integer
  - The only required Extension in a RISC-V implementation
- RISC-V allows for custom, “Non-Standard”, extensions in an implementation
- Putting it all together (examples)
  - RV32I – The most basic RISC-V implementation
  - RV32IMAC – Integer + Multiply + Atomic + Compressed
  - RV64GC – 64bit IMAFDC
  - RV64GCXext – IMAFDC + a non-standard extension
- Binaries that target RV32I can run on any RISC-V system, regardless of extensions

Extension	Description
I	Integer
M	Integer Multiplication and Division
A	Atomics
F	Single-Precision Floating Point
D	Double-Precision Floating Point
G	General Purpose = IMAFD
C	16-bit Compressed Instructions
Non-Standard User-Level Extensions	
Xext	Non-standard extension “ext”

Common RISC-V Standard Extensions
*Not a complete list



RISC-V Modes

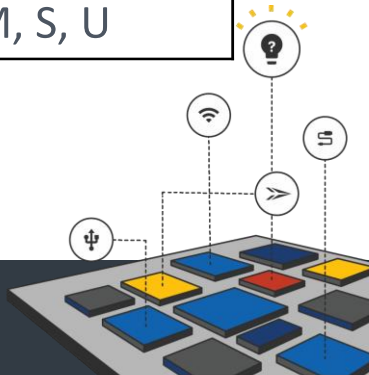
- RISC-V Privileged Specification defines 3 levels of privilege, called Modes
- Machine mode is the highest privileged mode and the only required mode
  - Allows for a range of targeted implementations
- Machine and Supervisor modes each have Control and Status Registers (CSRs)

RISC-V Modes

Level	Name	Abbr.
0	User/Application	U
1	Supervisor	S
	Reserved	
3	Machine	M

Supported Combinations of Modes

Supported Levels	Modes
1	M
2	M, U
3	M, S, U

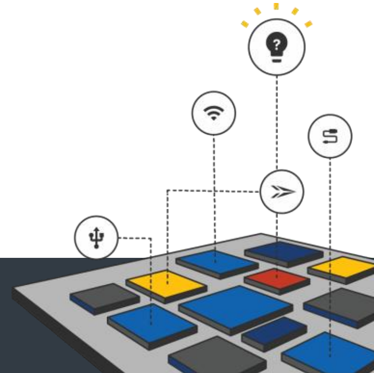


How to measure ISA quality?

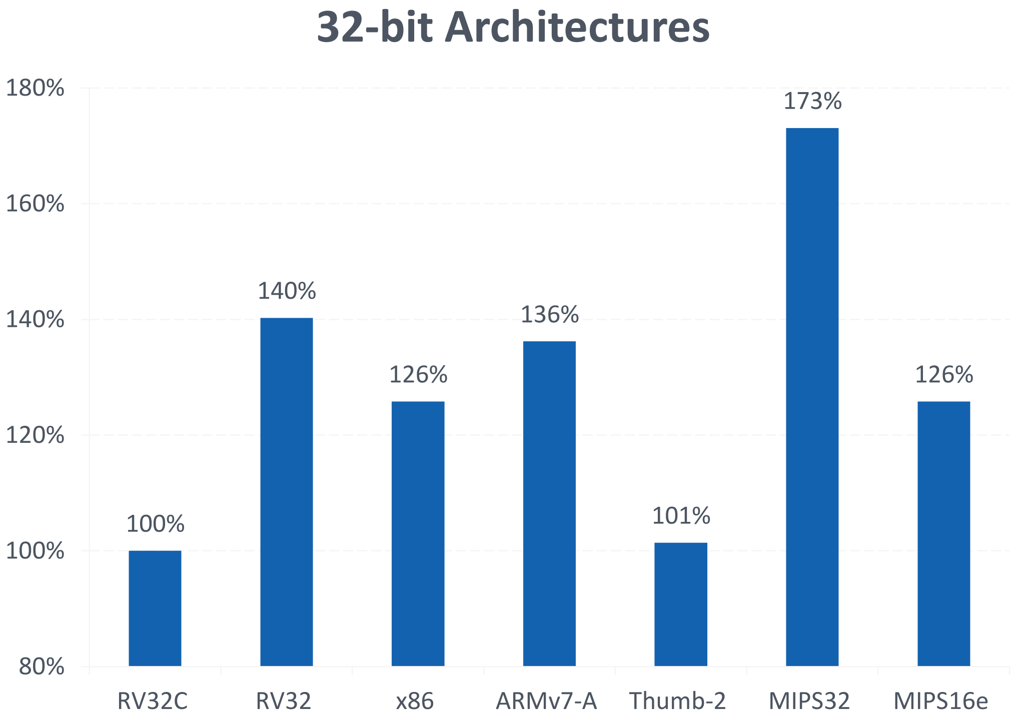
- Static code bytes for program

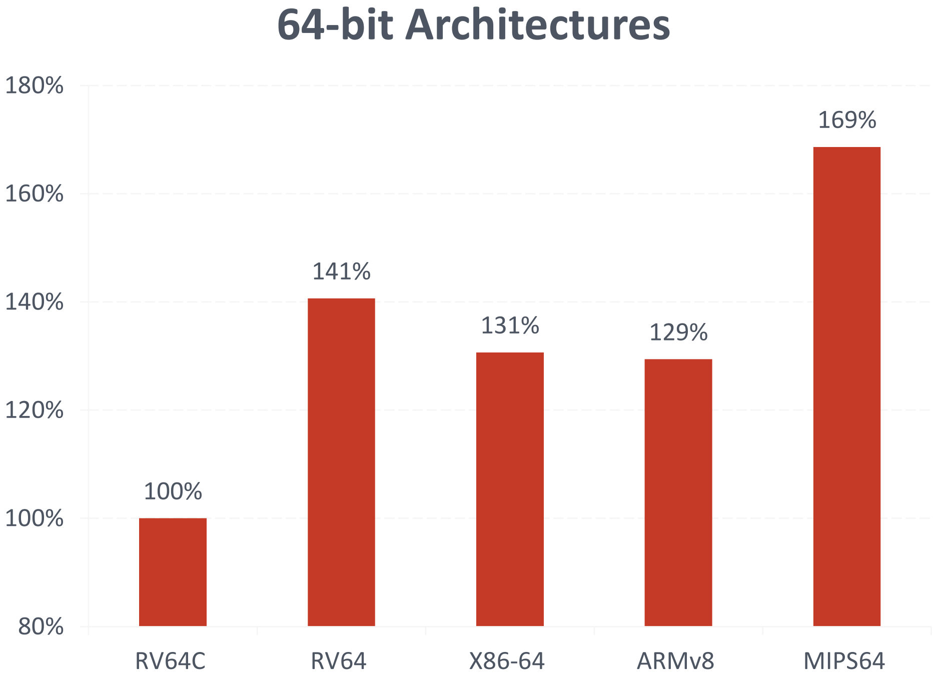
- Dynamic code bytes fetched for execution

- Microarchitectural work generated for execution



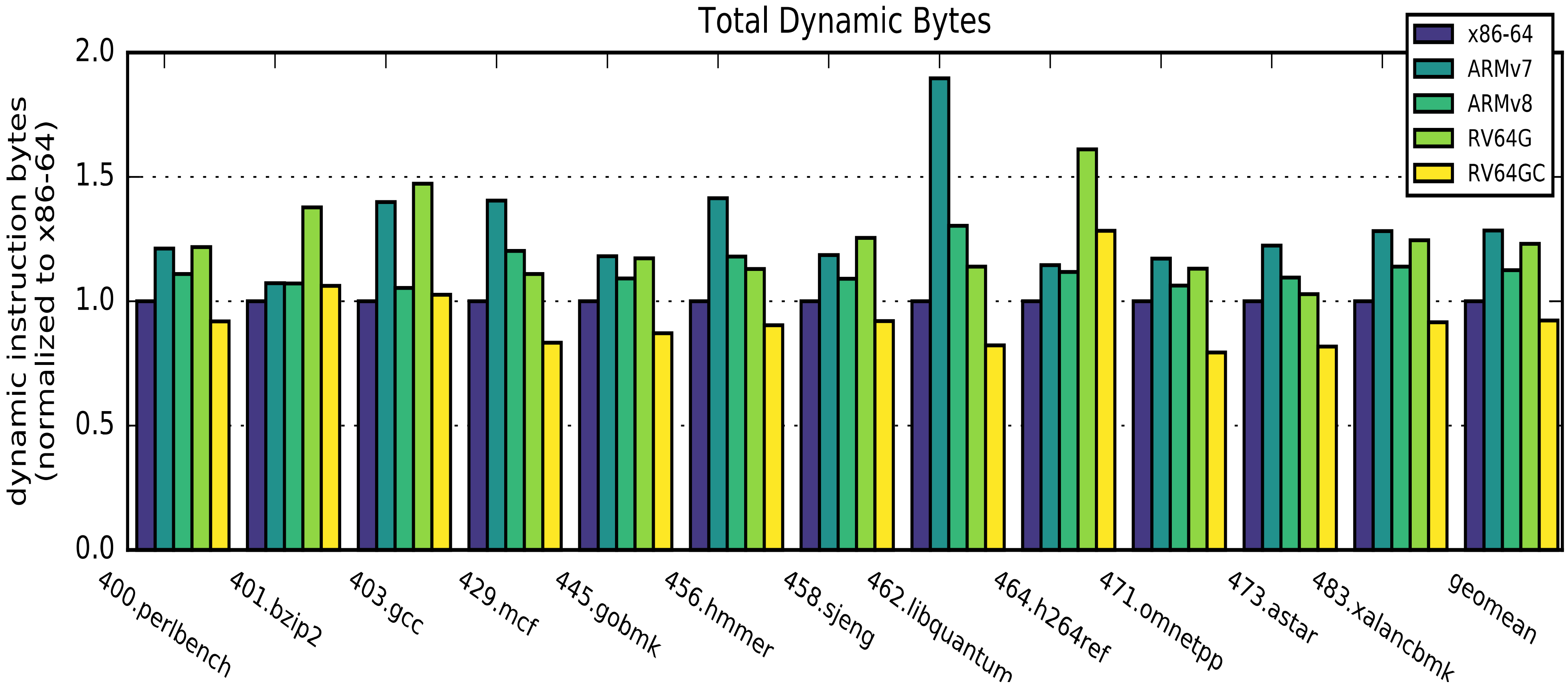
SPECint2006 compressed code size with save/restore optimization (relative to "standard" RVC)





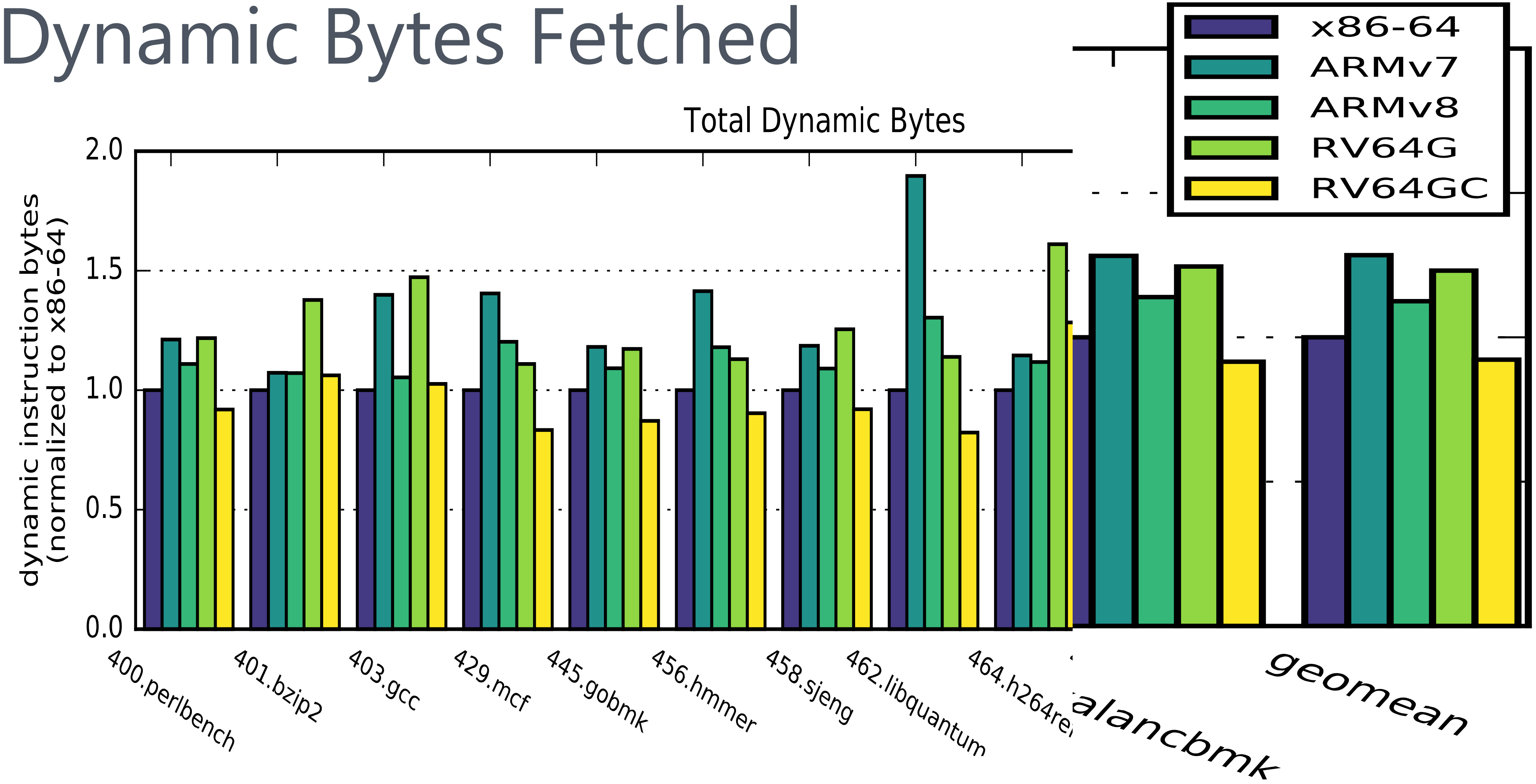
- RISC-V now smallest ISA for 32- and 64-bit addresses
- All results with same GCC compiler and options

Dynamic Bytes Fetched



- RV64GC is lowest overall in dynamic bytes fetched

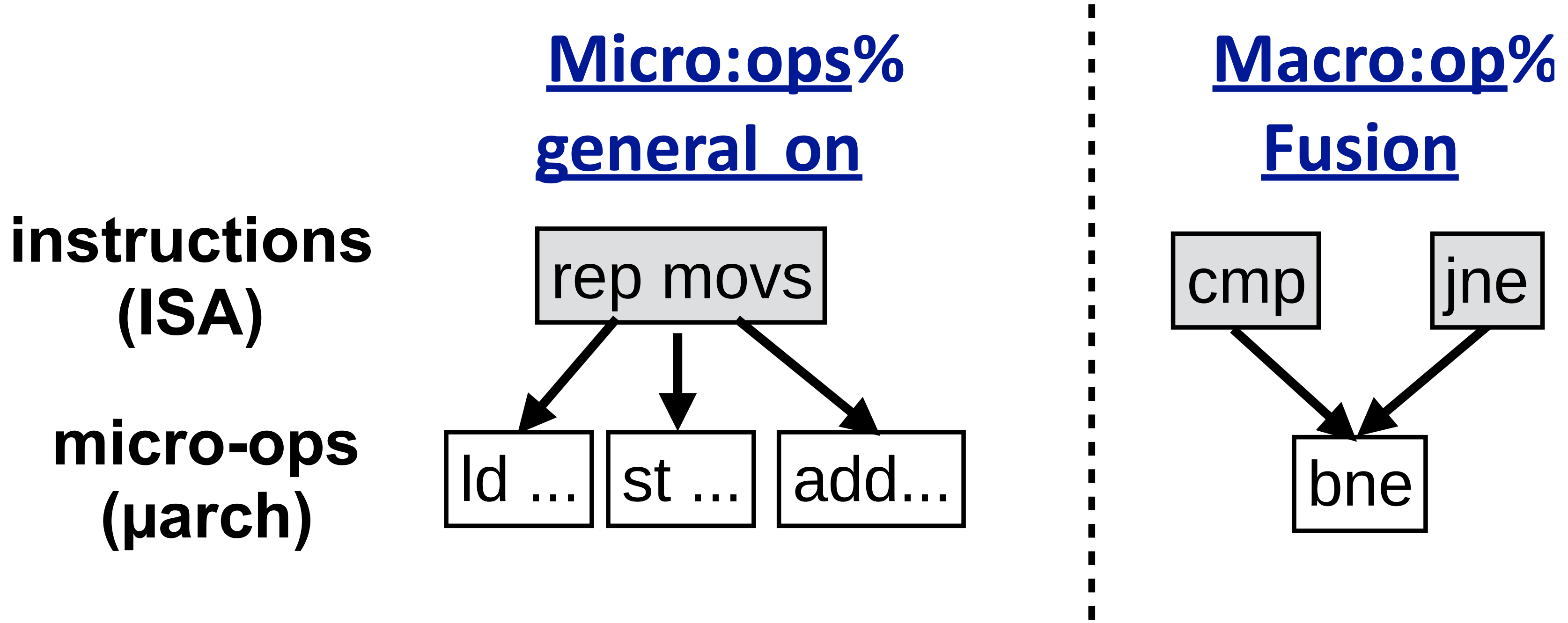
Dynamic Bytes Fetched



- RV64GC is lowest overall in dynamic bytes fetched

Converting Instructions to Microops

Microops are measure of microarchitectural work performed



Micro:ops% general on

Macro:op% Fusion

instructions (ISA)

micro-ops (μarch)

rep movs

ld ...

st ...

add...

cmp

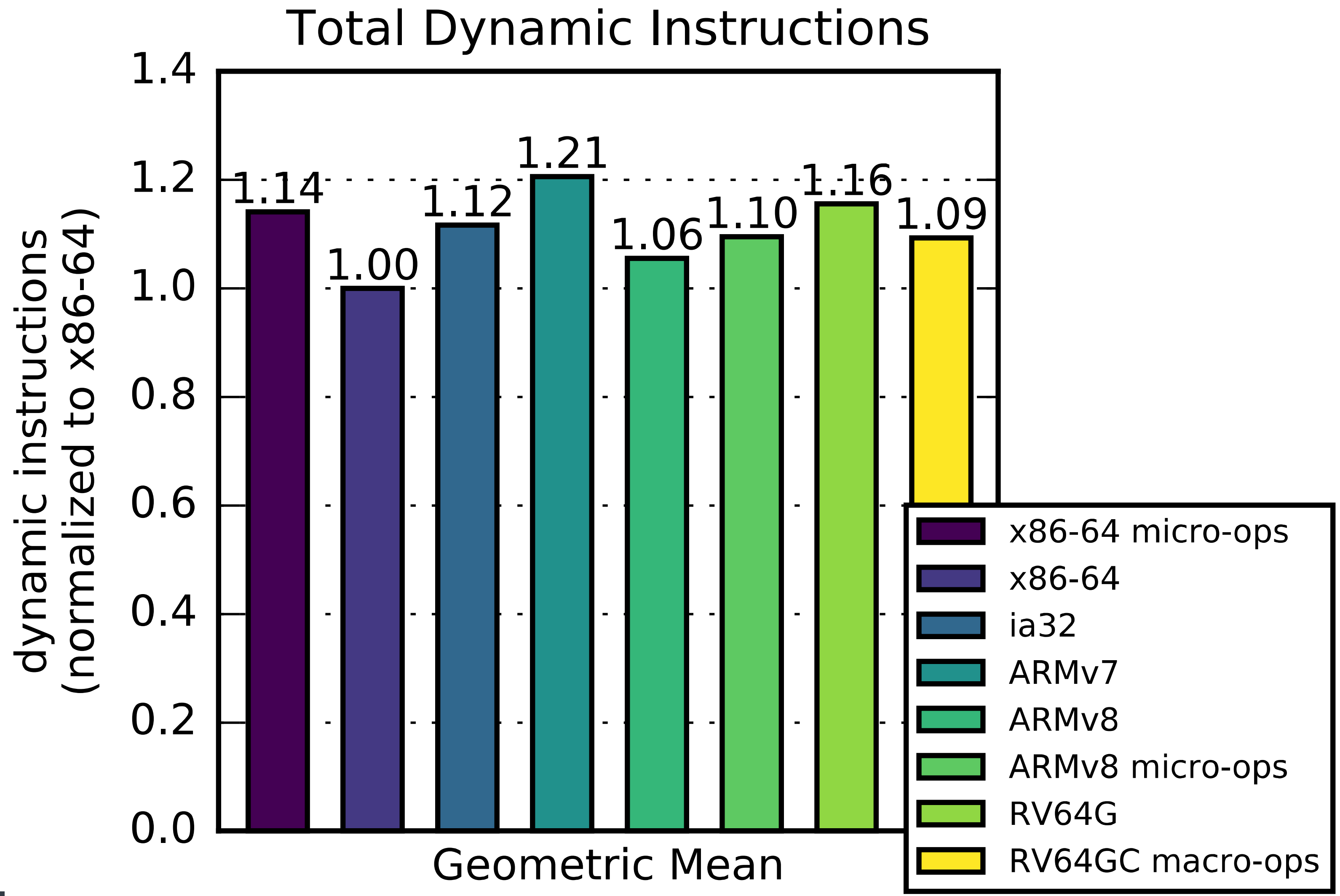
jne

bne

Multiple microinstructions from one macroinstructions

Or one microinstruction from multiple macroinstructions

RISC-V Microarchitectural work generated is less than other architectures

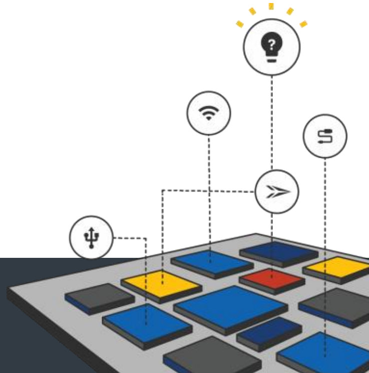


ISA quality -- summary

- Static code bytes for program
  - RISC-V is less

- Dynamic code bytes fetched for execution
  - RISC-V is less

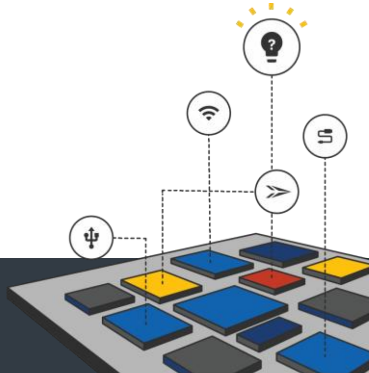
- Microarchitectural work generated for execution
  - RISC-V is less



Ok, but where are the implementations?

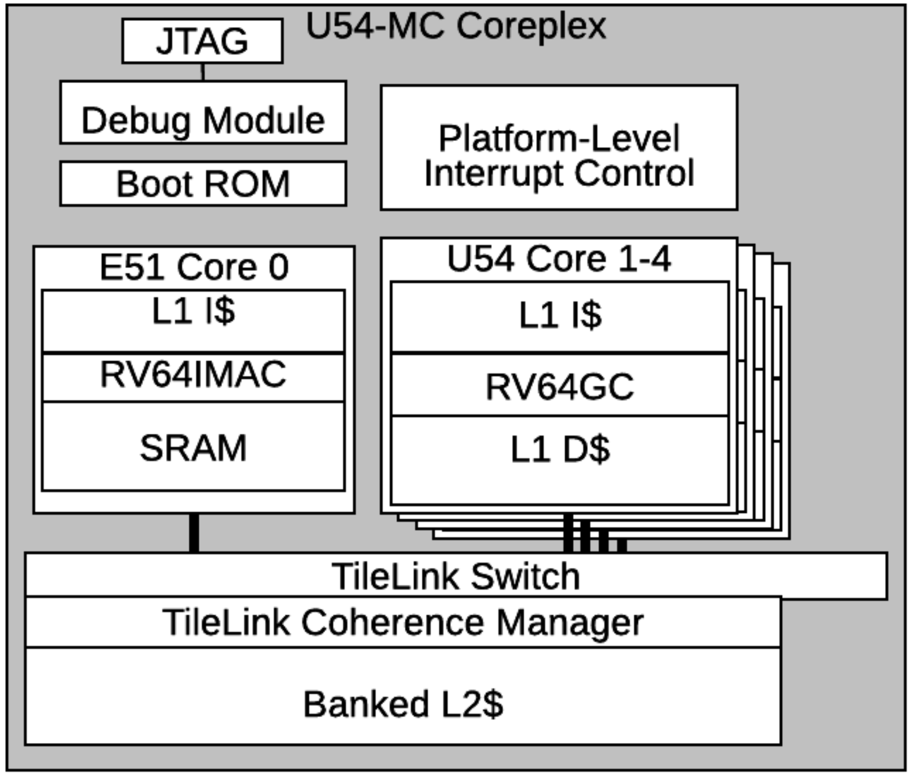
- The ISA might have benefits, but microarchitecture implementations are still very important.

- Up to now, most RISC-V implementations have only been targeted for Embedded Devices, even though the ISA is designed for more complex systems as well

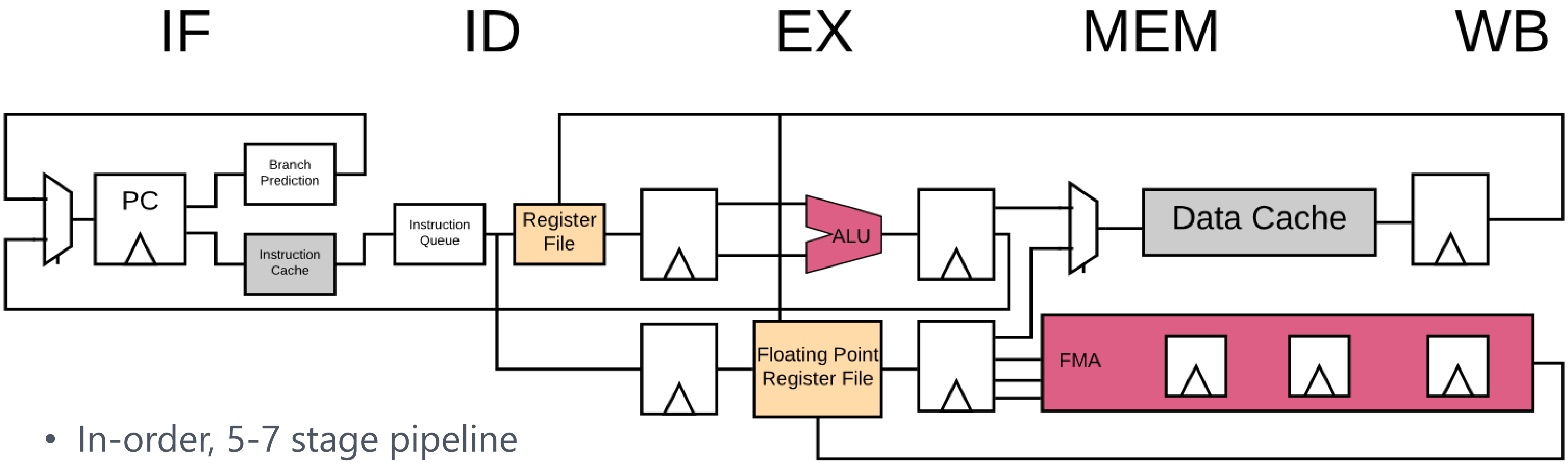


Introducing SiFive U54-MC Coreplex

- Features 4x U54 Cores and 1x E51 Core
- Each U54 core is a 64-bit, 1.5 GHz CPU
- U54 implements RV64GC
  - Hardware multiply/divide
  - Atomic Instructions
  - 16-bit compressed instructions
  - Single and Double Precision Floating Point with Fused Multiply Add
- U54 supports RISC-V privileged modes M, S, and U
- E51 Core is a 64-bit, 1.5GHz CPU “minion core”
  - RV64IMAC
  - RISC-V privileged modes M and U only
- Coherent, 2MB 16-way L2 subsystem
- Standard platform for Linux RISC-V development



SiFive U54 Core Pipeline Diagram

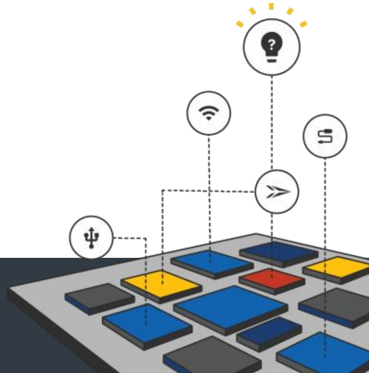


- In-order, 5-7 stage pipeline
- Advanced Branch Prediction
  - 30-entry BTB
  - 256-entry BHT
- MMU Support
  - 32-entry fully associative ITLB and DTLB (each)
  - 128-entry direct-mapped unified L2 TLB

- High performance memory subsystem
  - 32KB 8-way Instruction Cache
  - 32KB 8-way Data Cache
  - 2MB 16-way shared L2 Cache
- Supports Latest RISC-V specifications

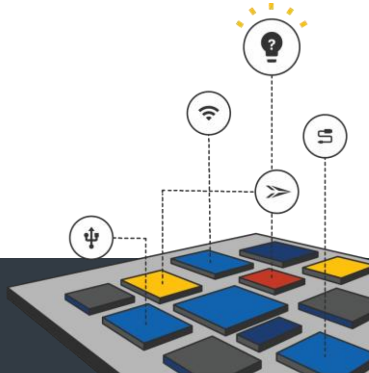
U54 Specifications

- Speed:
    - TSMC 28nm HPC:
        - Typical: 1.5 GHz, 0.9V, 25C
        - Fast/Fast: 2.6 GHz, 0.99V, 125C
        - Slow/Slow: 960 MHz, 0.81V, -40C
    - Standard cell, 12-track library
- Area:
    - Single U54 Core-only Area: 0.234 mm^2
    - Single U54 Coreplex Area: 0.538 mm^2
        - Includes 32KB/32KB L1 Cache
- Performance:
    - 1.7 DMIPS/MHz
    - 2.75 CoreMark/MHz



RISC-V Software Update

- Binutils is stable & mainlined as of 2.28 release (March 2017)
- GCC stable & mainlined as of 7.1 release (May 2017)
- LLVM in process of being submitted upstream
- Linux is in process of being submitted upstream
  - glibc targeting next release in Feb 2018
- newlib upstreamed and will be released in 2.6.0 (Dec 2017)



Robust Ecosystem developing for RISC-V tools

Free, Open-Source Tools

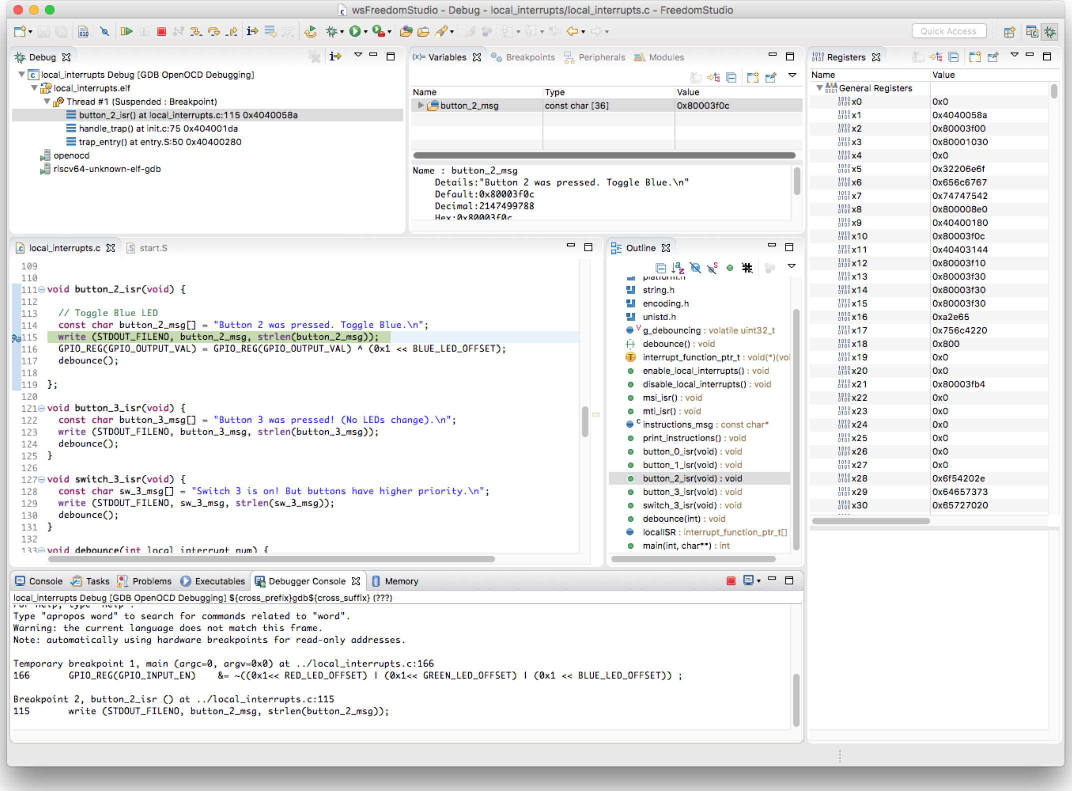
FreedomStudio

Freedom Studio is the fastest way to get started programming your SiFive hardware. Freedom Studio is built on top of the popular Eclipse IDE and packaged with a prebuilt toolchain and example projects from the Freedom E SDK. Freedom Studio is compatible with all SiFive products including the HiFive1, the Freedom E300 Arty FPGA Dev Kit, and more.

Freedom Studio v20170616 (beta2)

- Windows ⬇
- macOS ⬇
- Linux ⬇

FREEDOM STUDIO MANUAL



```
local_interrupts Debug [GDB OpenOCD Debugging]
  local_interrupts.elf
    Thread #1 (Suspended : Breakpoint)
      button_2_isr() at local_interrupts.c:115 0x4040058a
      handle_trap() at init.c:75 0x404001da
      trap_entry() at entry.S:50 0x40400280
  openocd
  riscv64-unknown-elf-gdb
```

Name	Type	Value
button_2_msg	const char [36]	0x80003f0c

```
Name : button_2_msg
    Details:"Button 2 was pressed. Toggle Blue.\n"
    Default:0x80003f0c
    Decimal:2147499788
    Hex:0x80003f0c
```

Name	Value
General Registers	
x0	0x0
x1	0x4040058a
x2	0x80003f00
x3	0x80001030
x4	0x0
x5	0x32206e6f
x6	0x656c6767
x7	0x74747542
x8	0x800008e0
x9	0x40400180
x10	0x80003f0c
x11	0x40403144
x12	0x80003f10
x13	0x80003f30
x14	0x80003f30
x15	0x80003f30
x16	0xa2e65
x17	0x756c4220
x18	0x800
x19	0x0
x20	0x0
x21	0x80003fb4
x22	0x0
x23	0x0
x24	0x0
x25	0x0
x26	0x0
x27	0x0
x28	0x6f54202e
x29	0x64657373
x30	0x65727020

```c


void button_2_isr(void) {

  // Toggle Blue LED
  const char button_2_msg[] = "Button 2 was pressed. Toggle Blue.\n";
  write (STDOUT_FILENO, button_2_msg, strlen(button_2_msg));
  GPIO_REG(GPIO_OUTPUT_VAL) = GPIO_REG(GPIO_OUTPUT_VAL) ^ (0x1 << BLUE_LED_OFFSET);
  debounce();

};

void button_3_isr(void) {
  const char button_3_msg[] = "Button 3 was pressed! (No LEDs change).\n";
  write (STDOUT_FILENO, button_3_msg, strlen(button_3_msg));
  debounce();
}

void switch_3_isr(void) {
  const char sw_3_msg[] = "Switch 3 is on! But buttons have higher priority.\n";
  write (STDOUT_FILENO, sw_3_msg, strlen(sw_3_msg));
  debounce();
}

void debounce(int local_interrupt_num) {
```

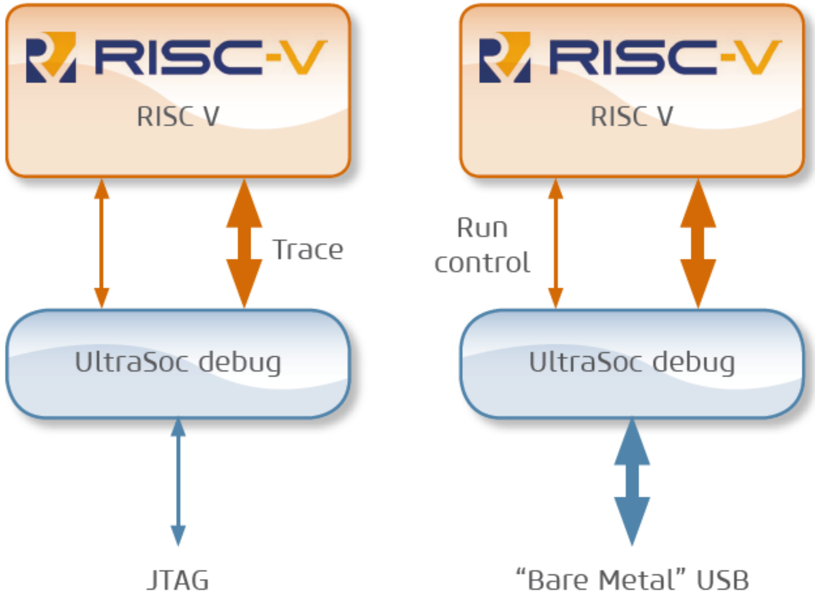
```
platform.h
string.h
encoding.h
unistd.h
g_debouncing : volatile uint32_t
debounce() : void
interrupt_function_ptr_t : void(*)(voi
enable_local_interrupts() : void
disable_local_interrupts() : void
msi_isr() : void
mti_isr() : void
instructions_msg : const char*
print_instructions() : void
button_0_isr(void) : void
button_1_isr(void) : void
button_2_isr(void) : void
button_3_isr(void) : void
switch_3_isr(void) : void
debounce(int) : void
localISR : interrupt_function_ptr_t[]
main(int, char**) : int
```

```
local_interrupts Debug [GDB OpenOCD Debugging] ${cross_prefix}gdb${cross_suffix} (???)
For help, type "help".
Type "apropos word" to search for commands related to "word".
Warning: the current language does not match this frame.
Note: automatically using hardware breakpoints for read-only addresses.

Temporary breakpoint 1, main (argc=0, argv=0x0) at ../local_interrupts.c:166
166       GPIO_REG(GPIO_INPUT_EN)    &= ~((0x1<< RED_LED_OFFSET) | (0x1<< GREEN_LED_OFFSET) | (0x1 << BLUE_LED_OFFSET)) ;

Breakpoint 2, button_2_isr () at ../local_interrupts.c:115
115       write (STDOUT_FILENO, button_2_msg, strlen(button_2_msg));
```

Announced Commercial Debug and Tools

UltraSoC

ultrasoc for RISC-V



At-a-glance

- Run control: halt, run, single step, break point
- Processor trace
- JTAG and cJTAG
- "Bare metal" USB
- Other connectivity options
- Standards compliant
- IDE support: Eclipse / GDB, 3rd party solutions as they emerge
- Non-intrusive, wire-speed
- Reduces post-silicon bring-up and debug burden

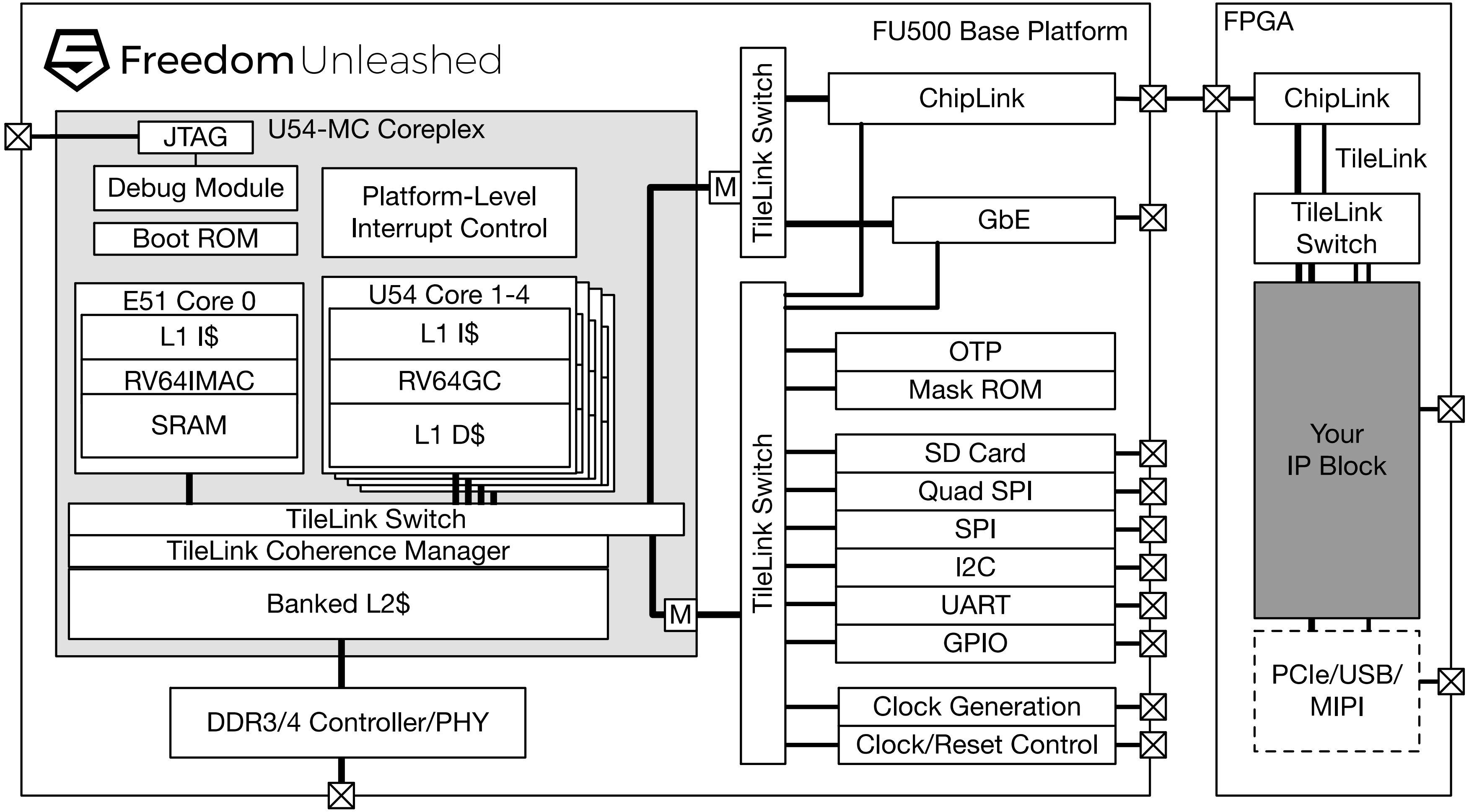
Segger

SiFive

J-Link Probe Support

Freedom U500 Base Platform Block Diagram

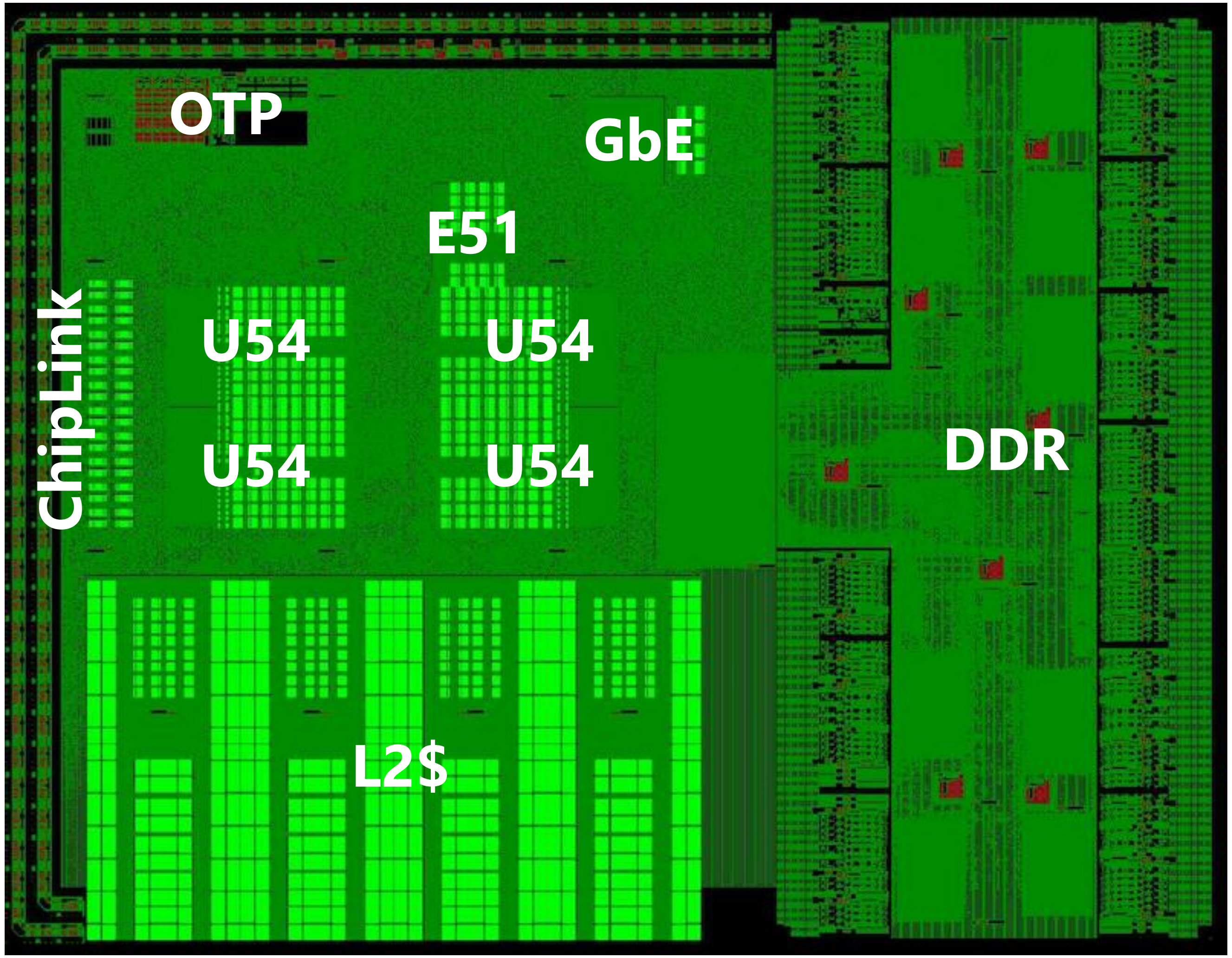
TSMC 28nm Chip for Rapid Customization of the Freedom Unleashed Platform



- U54-MC Coreplex
  - Single- and Double-precision floating-point support
  - Banked L2$ with directory-based cache-coherence
  - Modern OS support
- ChipLink
  - Serialized Chip-to-Chip TileLink Interconnect
- GbE
- Peripherals
- DDR3/4

Freedom U500 Base Platform Chip

~30mm^2 in TSMC 28nm



- 250M transistors
- 1.5 GHz+ SiFive E51/U54 CPU
  - 1x E51: 16KB L1I$ and 8KB DTIM
  - 4x U54: 32KB L1I$ and 32KB L1D$
  - ECC support
- Banked 2MB L2$
  - ECC support
- TSMC 28HPC
- FCBGA package
- Development board available in Q1 2018

Summary

- RISC-V ISA is designed for all computing devices
- RISC-V ISA has many benefits, but microarchitectural implementations still matter
- U54-MC is a advanced, multicore system, with Virtual Memory Support and M+S+U privileged mode, enabling Linux and other advanced Operating System
- Development board with Freedom U500 available Q1 2018
- Lead Customers already licensed U54-MC Coreplex
- U54-MC Coreplex available as Soft IP for evaluation now