RISC-V Architecture and Supercomputing

Thomson Kneeland, Tai-Yin Chong, and David Berstol

CISC - Complex Instruction Set Computer

- Predecessor of RISC
- Capable of multistep operations and addressing modes with single instructions
- Complete task with minimal code and rely on hardware
  - Minimizes compiler work with less assembly code
  - Build task directly into hardware
- Multiclock complex instructions
- High cycles per second
- Memory to memory “load” and “store” incorporated in instructions
- Intel x86 only chip that retains CISC architecture

RISC- Reduced Instruction Set Computer

- Developed in late 70’s: IBM, Stanford, UC Berkeley
- RISC ISA allows the processor to average nearly 1 instruction per cycle
  - **I**nstruction **S**et **A**rchitecture is the interface between software and hardware
- ALU operations between registers only (load/store architecture)
  - In CISC arithmetic, one operand could be in memory and another in a register
  - Large number of registers to minimize interaction with memory
- Pipelining - allows for simultaneous execution of parts
- OS and app programmers can develop code easily
- MIPS is a reduced instruction set developed at Stanford in 1984
  - Lowers compiler to hardware level

RISC V CPU Architecture

- Entered the market in 2014
- Originally designed to support education and computer architecture research
- Allows for efficient implementation by avoiding “overachitecturing”
- Open source and available to public
- Potentially faster and more efficient than x86 and ARM, given good implementation
  - RISC-V Rocket core twice as energy efficient as ARM counterpart Cortex-A5
- Focuses on performance and power efficiency

Implementations of RISC V ISA

- Cloud computers
- Mobile phones
  - Samsung
- Embedded systems
- Supercomputers
- Storage devices
  - Western Digital
- GPUs
  - Nvidia

Migration to RISC V

- Lack of alternative RISC ISA
  - ARM bought out by Softbank, MIPS difficult to use due to intellectual property issues
  - Low competitive RISC ISA to ARM and MIPS until RISC V
- The RISC V foundation allows companies to manage the instruction set as a team
  - Berkeley Architecture Research, Google, Samsung, NVidia, and many more
- Western Digital
  - Recently announced their processors transitioning over to RISC V

Free & Open RISC-V Reference Card ①

Base Integer Instructions: RV32I, RV64I, and RV128I

Category	Name	Fmt	RV32I Base		+RV{64,128}	
Loads	Load Byte	I	LB	rd,rs1,imm		
	Load Halfword	I	LH	rd,rs1,imm		
	Load Word	I	LW	rd,rs1,imm	L{D|Q}	rd,rs1,imm
	Load Byte Unsigned	I	LBU	rd,rs1,imm		
	Load Half Unsigned	I	LHU	rd,rs1,imm	L{W|D}U	rd,rs1,imm
Stores	Store Byte	S	SB	rs1,rs2,imm		
	Store Halfword	S	SH	rs1,rs2,imm		
	Store Word	S	SW	rs1,rs2,imm	S{D|Q}	rs1,rs2,imm
Shifts	Shift Left	R	SLL	rd,rs1,rs2	SLL{W|D}	rd,rs1,rs2
	Shift Left Immediate	I	SLLI	rd,rs1,shamt	SLLI{W|D}	rd,rs1,shamt
	Shift Right	R	SRL	rd,rs1,rs2	SRL{W|D}	rd,rs1,rs2
	Shift Right Immediate	I	SRLI	rd,rs1,shamt	SRLI{W|D}	rd,rs1,shamt
	Shift Right Arithmetic	R	SRA	rd,rs1,rs2	SRA{W|D}	rd,rs1,rs2
	Shift Right Arith Imm	I	SRAI	rd,rs1,shamt	SRAI{W|D}	rd,rs1,shamt
Arithmetic	ADD	R	ADD	rd,rs1,rs2	ADD{W|D}	rd,rs1,rs2
	ADD Immediate	I	ADDI	rd,rs1,imm	ADDI{W|D}	rd,rs1,imm
	SUBtract	R	SUB	rd,rs1,rs2	SUB{W|D}	rd,rs1,rs2
	Load Upper Imm	U	LUI	rd,imm		
	Add Upper Imm to PC	U	AUIPC	rd,imm		
Logical	XOR	R	XOR	rd,rs1,rs2		
	XOR Immediate	I	XORI	rd,rs1,imm		
	OR	R	OR	rd,rs1,rs2		
	OR Immediate	I	ORI	rd,rs1,imm		
	AND	R	AND	rd,rs1,rs2		
	AND Immediate	I	ANDI	rd,rs1,imm		

RV Privileged Instructions

Category	Name	RV mnemonic
CSR Access	Atomic R/W	CSRRW rd,csr,rs1
	Atomic Read & Set Bit	CSRRS rd,csr,rs1
	Atomic Read & Clear Bit	CSRRC rd,csr,rs1
	Atomic R/W Imm	CSRRWI rd,csr,imm
	Atomic Read & Set Bit Imm	CSRRSI rd,csr,imm
	Atomic Read & Clear Bit Imm	CSRRCI rd,csr,imm
Change Level	Env. Call	ECALL
	Environment Breakpoint	EBREAK
	Environment Return	ERET
Trap Redirect	to Supervisor	MRTS
	Redirect Trap to Hypervisor	MRTH
	Hypervisor Trap to Supervisor	HRTS
Interrupt	Wait for Interrupt	WFI
MMU	Supervisor FENCE	SFENCE.VM rs1

Optional Compressed (16-bit) Instruction Extension: RVC

Category	Name	Fmt	RVC		RVI equivalent
Loads	Load Word	CL	C.LW	rd',rs1',imm	LW rd',rs1',imm*4
	Load Word SP	CI	C.LWSP	rd,imm	LW rd,sp,imm*4
	Load Double	CL	C.LD	rd',rs1',imm	LD rd',rs1',imm*8
	Load Double SP	CI	C.LDSP	rd,imm	LD rd,sp,imm*8
	Load Quad	CL	C.LQ	rd',rs1',imm	LQ rd',rs1',imm*16
	Load Quad SP	CI	C.LQSP	rd,imm	LQ rd,sp,imm*16

Sample RISC V program

Editor Simulator

```
## program that finds value of (2+3)*4, enters in register t4
li t0, 2
li t1, 3
li t2, 4
add t3, t0, t1
mul t4, t2, t3
```

Output to register t4

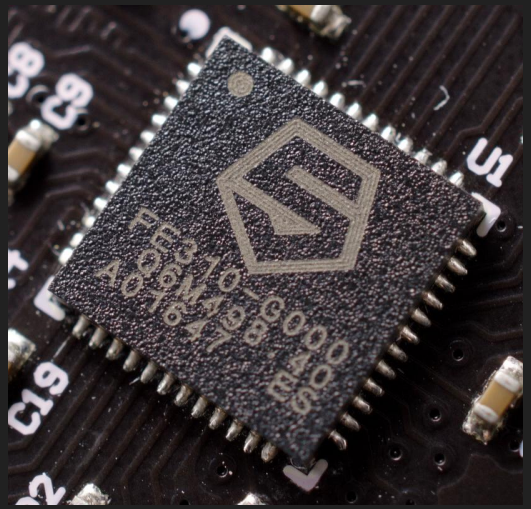
Editor | Simulator

Run | Step | Prev | Reset | Dump

Machine Code	Basic Code	Original Code
0x00200293	addi x5 x0 2	li t0, 2
0x00300313	addi x6 x0 3	li t1, 3
0x00400393	addi x7 x0 4	li t2, 4
0x00628e33	add x28 x5 x6	add t3, t0, t1
0x03c38eb3	mul x29 x7 x28	mul t4, t2, t3

s5 (x21)	0
s6 (x22)	0
s7 (x23)	0
s8 (x24)	0
s9 (x25)	0
s10 (x26)	0
s11 (x27)	0
t3 (x28)	5
t4 (x29)	20
t5	0

The Processors

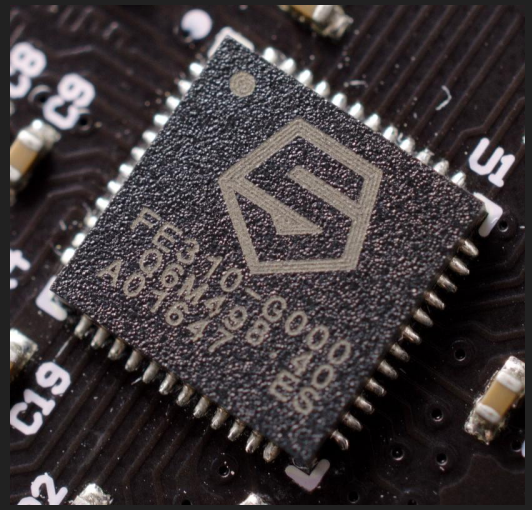


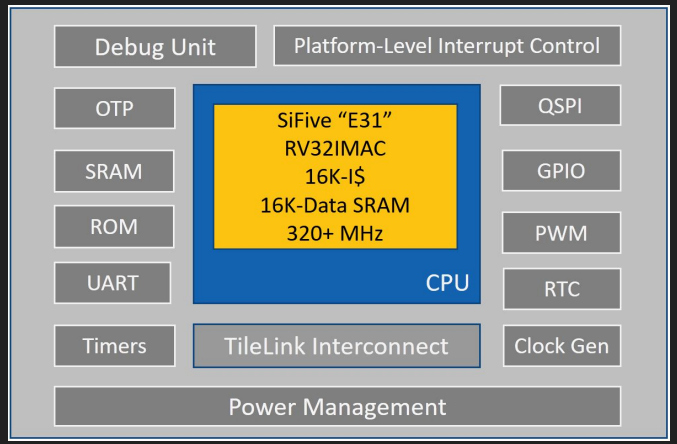
Esperanto Technologies

- Announced future plans to develop energy-efficient computing solutions for AI and Machine Learning applications
  - Based on RISC-V ISA
- Advantage of using RISC-V
  - Over four thousand full 64-bit cores on a single chip
  - Growing software base of operating systems, compilers and applications
  - Ability to deliver world class TeraFlop levels of computing
  - Increased software availability

Freedom E310

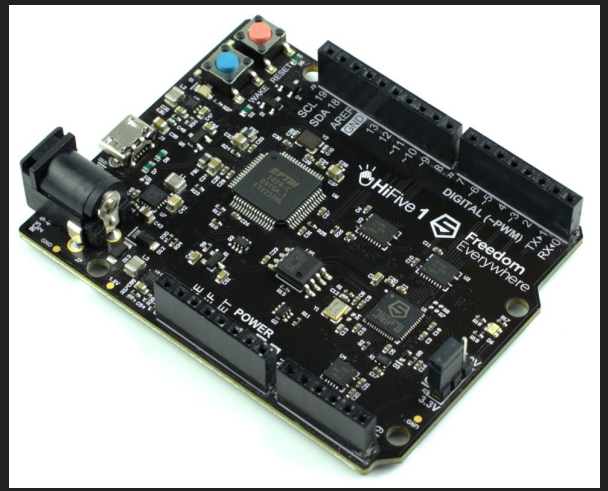
- Industry's first commercially available RISC-V SoC
- Released November 29th, 2016
- Developed by SiFive, startup in Silicon Valley
  - First company to produce a chip implementing RISC-V ISA





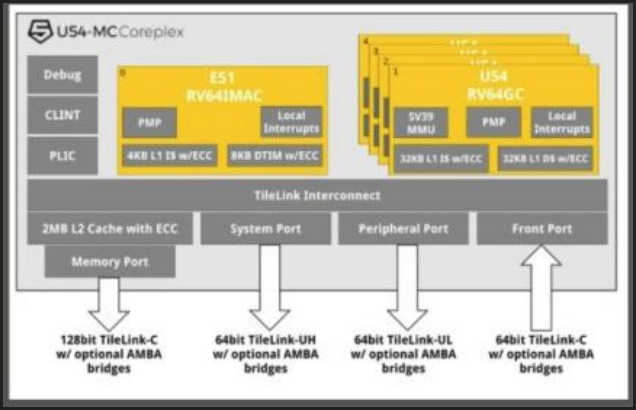
HiFive1

- Arduino-Compatible development kit
- RISC-V based
  - Features the Freedom E310
- Available for $59
- Will allow a variety of designers and system architects to create their own products



U54-MC Coreplex IP

- World’s first 64 bit quad-core RISC-V open source processor in Oct 17
- Created by SiFive
- Brings RISC-V into Linux processing
- Four U54 CPUs and a single E51 CPU; running at 1.5GHz each
- 100 prototypes available at $100,000



U54-MCCoreplex	ARM Cortex-A35
RV64GC M + S + U Mode	ARMv8-A, AArch32, AArch64
16 bit instructions	AArch32 only
Physical Memory Protection (PMP) and MMU	None, MMU only
Real-time capable	Not applicable
E51 Monitor Core	Requires additional IP
Integrated interrupt controller	Requires additional IP

Image: SiFive

Nvidia Falcon

- Nvidia GPUS have featured a proprietary microcontroller called “Falcon”
  - Fast Logic Controller
  - Protects hardware from misprogramming/malicious software
  - 32 bit address range
  - No caches until recent Icache
- Next generation microcontroller is being built on RISC-V
  - More capabilities include greater address width and configurable cache
  - Security support, software toolchains and more
  - Backward compatibility opens up to 3rd party programming
  - Can mix and match internally/externally developed cores
- NVidia will use RISC-V in many of its future products

NVidia compared pros and cons

Item	Requirement	ARM A53	ARM A9	ARM R5	RISC-V Rocket	NV RISC-V
Core perf	>2x falcon	Yes	Yes	Yes	Yes	Yes
Area (16ff)	<0.1mm^2	No	No	Yes	Yes	Yes
Security	Yes	TZ	TZ	No	Yes	Yes
TCM	Yes	Yes	No	Yes	No	Yes
L1 I/D $	Yes	Yes	Yes	Yes	Yes	Yes
Addressing	64bit	Yes	No	No	Yes	Yes
Extensible ISA	Yes	No	No	No	Yes	Yes
Safety (ECC/Parity)	Yes	Yes	Yes	Yes	Yes	Yes
Functional Simulation model	Yes	Yes	No	No	No	Yes

NVIDIA

RISC-V surpassed all in criteria

Supercomputers

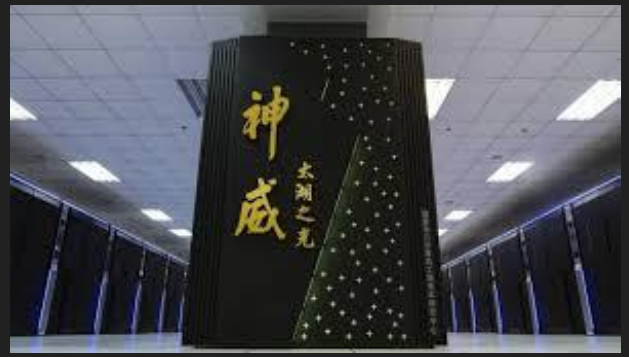
Tsubame 3.01, Japan

Supercomputers and processor architecture

- Massively Parallel Processing (MPP)
  - Each processor has its own memory and operating system
  - All connected and run in parallel for unified system
  - Can all work on same task
- Early models designed by Cray Research in 1960’s
- Computer cluster (local location)
- Distributed computing (BOINC, SETI, etc)
- Can utilize CPUs or a combination of CPUs and GPUS
- Boundaries include power consumption, cooling needs, memory address problems

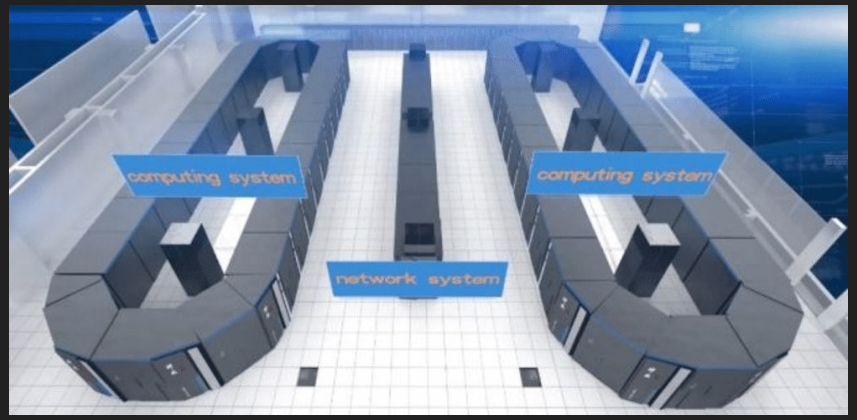
Top Supercomputers 11/13/2017

	NAME	COUNTRY	RMAX PFLOP/S	POWER MW
1	Sunway TaihuLight	China	93.0	15.4
2	Tianhe-2 (Milkyway-2)	China	33.9	17.8
3	Piz Daint	Switzerland	19.6	2.27
4	Gyoukou	Japan	19.1	1.35
5	Titan	USA	17.6	8.2



Sunway Taihu Light

#1 Sunway TaihuLight



Massively Parallel Processor Array

Distributed memory accessed locally;
each processor strictly encapsulated

- Benchmark rating of 93 peta flops
- 40,960 SW26010 64-bit RISC Sunway processors
- Each chip has 256 cores, for a total of 10,649,600 CPU cores in system
- Runs on its own operating system based on Linux
- “Network on a chip” vs traditional cache hierarchy

National Research Center in Wuxi China

- Chinese built 64-bit RISC processor, mystery architecture

#2 Tianhe-2 (TH-2)

- Previous fastest supercomputer, also Sunway, completed at end of 2013
  - 34 petaflop speed in Guangzhou
  - Team of 1,300 scientists and engineers to bring to fruition
  - 3,120,000 cores
  - Chinese built 64-bit RISC processor, mystery architecture
  - CPU processor memory = 1.34 PiB Pebibytes
  - State sponsored
  - 33,860 trillion calculations/second
  - Power consumption: 17,800 kW
  - Cost over $385 million
  - Occupies 7,750 square ft

#5 Titan

- Oak Ridge Laboratories in Oak Ridge, Tennessee
  - Sponsored by DOE and NOAA
  - Built by American supercomputer manufacturer Cray
  - Benchmark Rating of 17.59 PetaFLOPS
  - Was #1 in 2012 until overtaken by Tianhe-2
  - Includes GPUs and CPUs
    - 18,688 AMD Opteron 16 core CPUs
    - 18,688 Nvidia Tesla GPUs
  - $97 million to build
  - Scientific computing, climate models
    Cryptographic analysis

The future: IBM “Summit”

- In development at Oak Ridge and expected to be operational in 2018
- Uses IBM Power9 CPU and NVIDIA Volta GPU
  - Volta: 100 Teraflops/s deep learning
  - 21 billion transistors
  - CUDA and Tensor Cores: optimized for Deep Learning/AI
  - Should top Titan’s performance by 5-10X
  - >40 TeraFlops
  - Total system memory >10 PebiBytes
  - Will only require 4,600 processors vs Titan’s 18,688

IBM “Sierra”

- Being built for Livermore for assessment of nuclear weapon science/calculations
- Also IBM built
- Uses IBM Power 9 processors and NVidia’s Volta Graphics CPU
  - Will operate at 5-7X the present Sequoia supercomputer installed
  - 125 PetaFlop/s peak
  - 5 times more power efficient

Summary

- Overall, the industry is moving forward in terms of energy efficiency, size, and computing power
- Parallel Processing is the order of the day for complex calculations and modelling
- RISC and CISC technologies have actually been converging
- RISC-V could usher in increased efficiency for the future; especially as it is available for public utilization
- RISC-V and open source paves the way for increased collaboration for research without being limited by patents and corporate control/licensing

Appendix: Loop: Sum of Fibonacci Numbers (MIPS)

```
        la $a0, msg # output message to console
        li $v0, 4
        syscall

        li $v0, 5 #Read user input integer
        syscall

        move $t6, $v0 # store user input n
        sub $t5,$t6, 1 #subtract one for iteration

loop:
        bgt  $t3, $t5, exit  # loop up to user input number
        add $t2, $t0, $t1    # add previous two terms and store
        add $t4, $t4, $t2    # add new term to final sum
        move $t0, $t1        # store two terms for next iteration
        move $t1, $t2
        addi $t3, $t3, 1     # increment counter
        j loop

exit:
        la $a0, msg1 # output message to console
        li $v0, 4
        syscall

        move $a0, $t6  # output user input n to console
        li $v0, 1
        syscall

        la $a0, msg2 # output second part of message
        li $v0, 4
        syscall

        move $a0, $t4   # print final sum to console
        li $v0, 1
        syscall
```

Output

Mars Messages | Run I/O

Clear

```
Enter the number of Fibonacci numbers you would like to find the sum of: 1
The sum of the first 1 Fibonacci numbers is : 1
-- program is finished running --

Enter the number of Fibonacci numbers you would like to find the sum of: 4
The sum of the first 4 Fibonacci numbers is : 7
-- program is finished running --

Enter the number of Fibonacci numbers you would like to find the sum of: 11
The sum of the first 11 Fibonacci numbers is : 232
-- program is finished running --
```

Register to Memory (MIPS)

```
#Tai-Yin Chong
#CSIT 230
#PROJECT
#PROBLEM 3
        .data
Num0:   .word 3
Num1:   .word 4
msg0:   .asciiz "The sum of the two integers is: "
        .text
        .globl main
main:
        la $s0, Num0
        la $s1, Num1

        la $a0, msg0
        li $v0, 4
        syscall

        lw $t0, 0($s0)
        lw $t1, 0($s1)
        add $a0, $t0, $t1
        li $v0, 1
        syscall

        li $v0, 10
        syscall
```