CSUN | CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP122

COMP 122

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021

Fall 2022 Rev 8-23-22

Lectures on

Vol 4

Computer Architecture

& ASSEMBLY Programming

By

Dr Jeff Drobman

website → drjeffsoftware.com/classroom.html

email → jeffrey.drobman@csun.edu

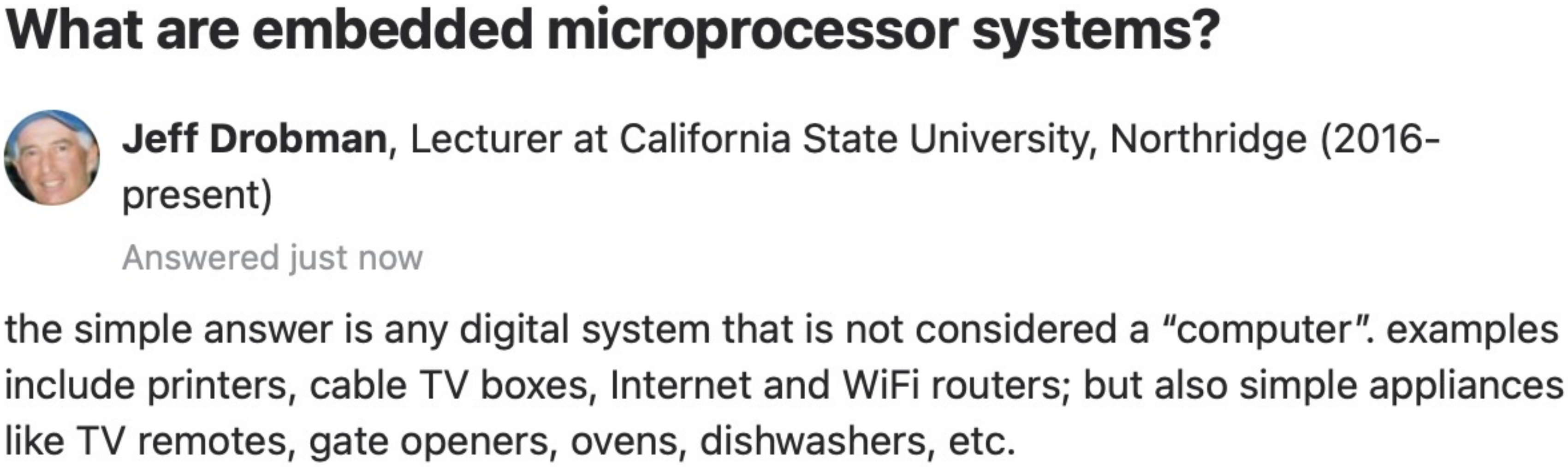
Index

Vol 4

Embedded Control

Embedded Systems

Quora



What are embedded microprocessor systems?

Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

Answered just now

the simple answer is any digital system that is not considered a “computer”. examples include printers, cable TV boxes, Internet and WiFi routers; but also simple appliances like TV remotes, gate openers, ovens, dishwashers, etc.

Embedded Systems

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

An ***embedded*** system is any digital system that is not considered a “computer”. Examples include printers, cable TV boxes, Internet and WiFi routers; but also simple appliances like TV remotes, gate openers, ovens, dishwashers, etc.

What makes embedded applications different? The answer is the software is a ***control program*** that is ***interrupt driven***. the software is mostly written in C due to needing low-level capabilities, some of which is written in ***assembly*** code.

Embedded MCU

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

First, ***microcontrollers*** are almost always used in an embedded application (like a TV or game controller, etc.). They are generally small, **cheap**, low power, low performance.

The whole original concept of a ***microcontroller*** -- starting with the Intel 8048, is a complete SMALL system on a chip: low power and low cost (and low performance). For example, today the **i8051** and the **PIC18F** are widely used at a price **<$1**.

Embedded Software

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

Source code is whatever computer language a programmer writes a program in, including assembly code. ***Machine code*** is an executable binary file -- produced by a compiler, interpreter or assembler (and possibly a linker). I.e., the former is the begining and the latter is the end result.

A sizable difference in applications environment results in different requirements for **OS**. Embedded systems typically use an ***RTOS*** -- Real-Time OS. That OS is designed to quickly perform task swapping in real-time response to external events that use interrupts. VxWorks is one such RTOS, and VRTX was popular decades ago.

Embedded Firmware

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

Embedded code is ANY and all software that controls a device or system, regardless of where it initially is *stored* (disk or ROM or flash). It excludes any "user" software or data. Once stored in ROM, those programs are not changeable, so are called ***firmware***.

Firmware may be updated, though, if stored in flash EEPROM (actually writeable, but slowly). When your cable TV box for example is updated, it is the “firmware” that is updated (so stored in a writable ROM, e.g., EEPROM).

Firmware often runs on a ***microcontroller*** class processor, which comes with on-chip ROM, but not always.

Embedded BIOS

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

Desktop and servers run an OS like Windows or Unix that contains ***BIOS*** (Basic I/O System) code. This is usually "firmware" too, in that it should not be modified.

Gary Kildall created maybe the first BIOS for his CP/M microprocessor OS ca 1975. Intel hired Gary to create their "blue box" development system OS *Isis*, and then he created his own variant he called "CP/M". AMD licensed CP/M for their own development system OS (they foolishly called *AmDOS* – and it was my job to provide tech support for it).

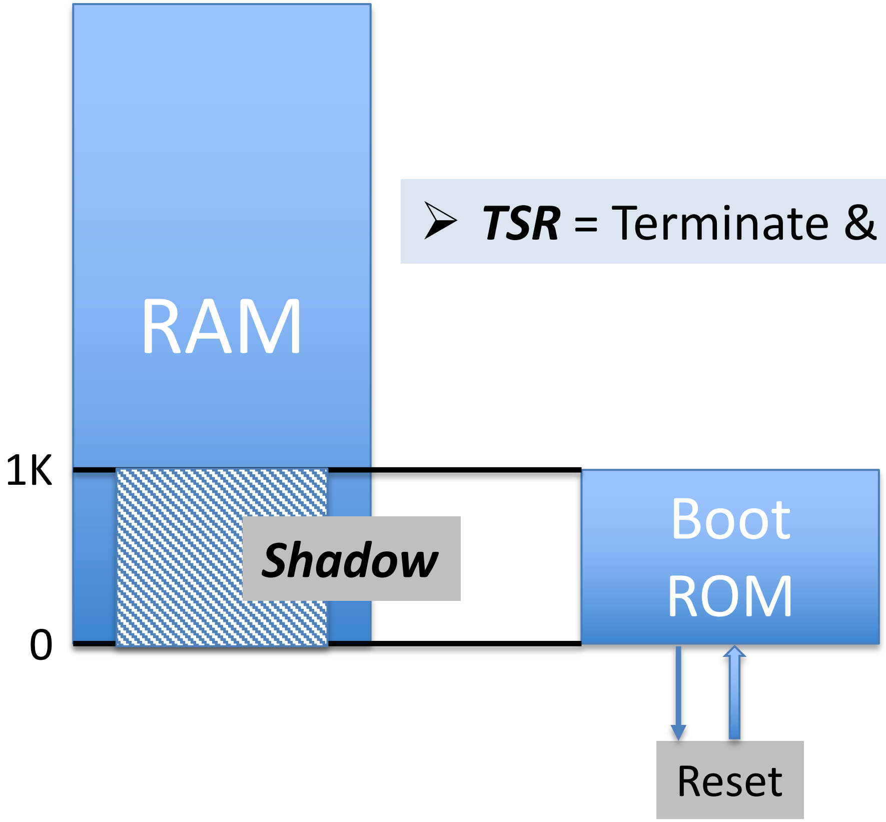
Embedded MT

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

I worked for IDT on their embedded MIPS CPU's (especially the R4600) in the 1990's. MIPS introduced MT in the late 1990's, along with "super-pipelining" (which was not really so special, as it is common now to have 8-11 stages). MIPS eschewed superscalar, even though its AMD 29K and Intel i960 comps used superscalar (in the 1990's). I believe that MT only came into its own since 2000 (as you said), as it is hard to make it effective enough. it started with only 2 threads (per core), but now I see 2-4 threads. hard for me to see how >2 threads helps any, since they have to share the same pipeline.

Shadow Boot ROM

Used on ***small memory*** embedded systems

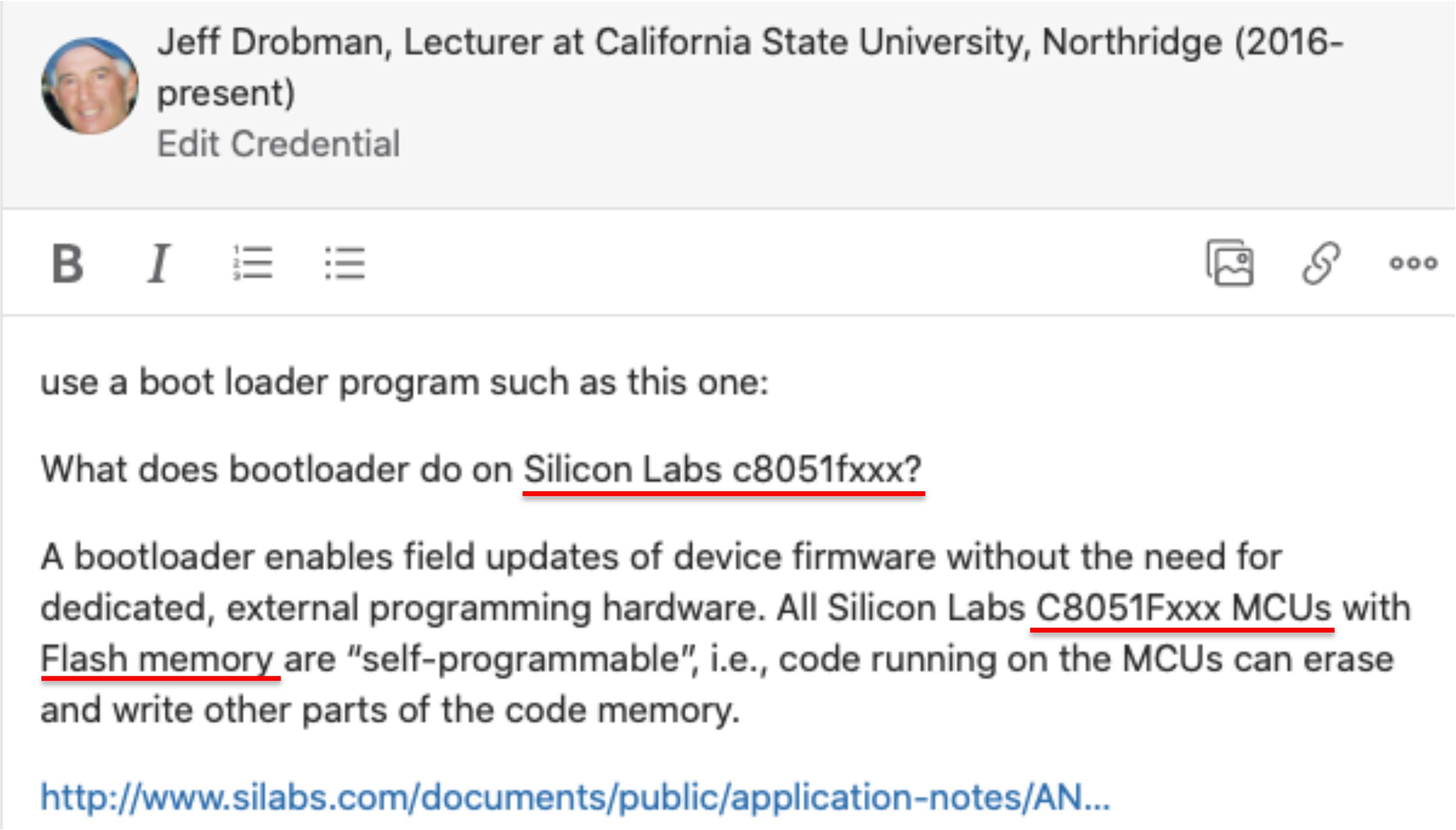


- ***TSR*** = Terminate & Stay Resident

Boot Loader

A bootloader for the 8051 family of microprocessors

The EmbedIT application was developed to be a bootloader for the 8051 family of microprocessors.



Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

Edit Credential

use a boot loader program such as this one:

What does bootloader do on Silicon Labs c8051fxxx?

A bootloader enables field updates of device firmware without the need for dedicated, external programming hardware. All Silicon Labs C8051Fxxx MCUs with Flash memory are "self-programmable", i.e., code running on the MCUs can erase and write other parts of the code memory.

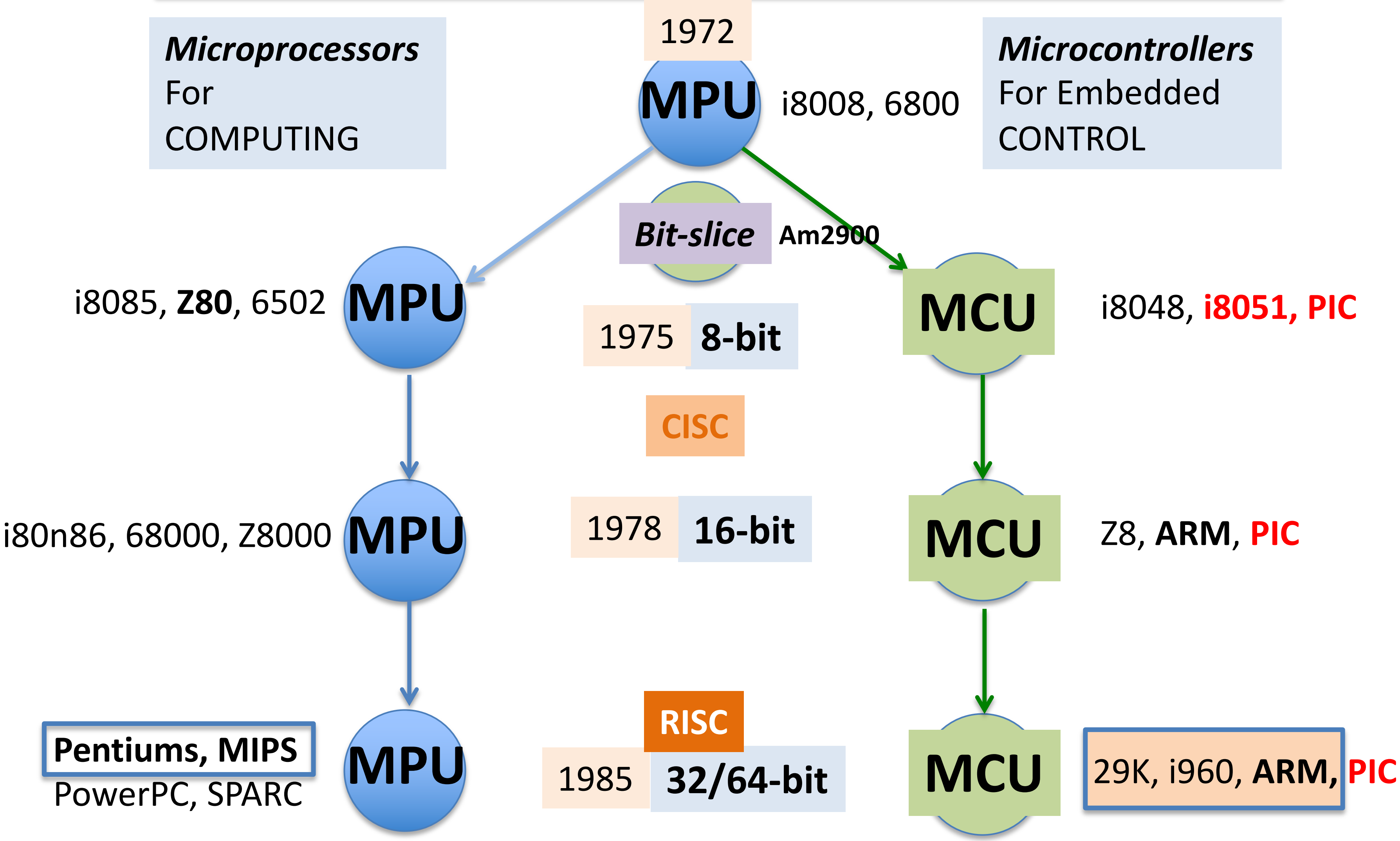
http://www.silabs.com/documents/public/application-notes/AN...

Computer Architecture

Early RISC – Embedded Controllers

- ❑ AMD 29K
- ❑ Intel i960

MPU/MCU Generations



Microprocessors
For
COMPUTING

1972
MPU i8008, 6800

Microcontrollers
For Embedded
CONTROL

Bit-slice **Am2900**

i8085, **Z80**, 6502 MPU

1975 **8-bit**

MCU i8048, **i8051, PIC**

CISC

i80n86, 68000, Z8000 MPU

1978 **16-bit**

MCU Z8, **ARM**, **PIC**

RISC

Pentiums, MIPS
PowerPC, SPARC

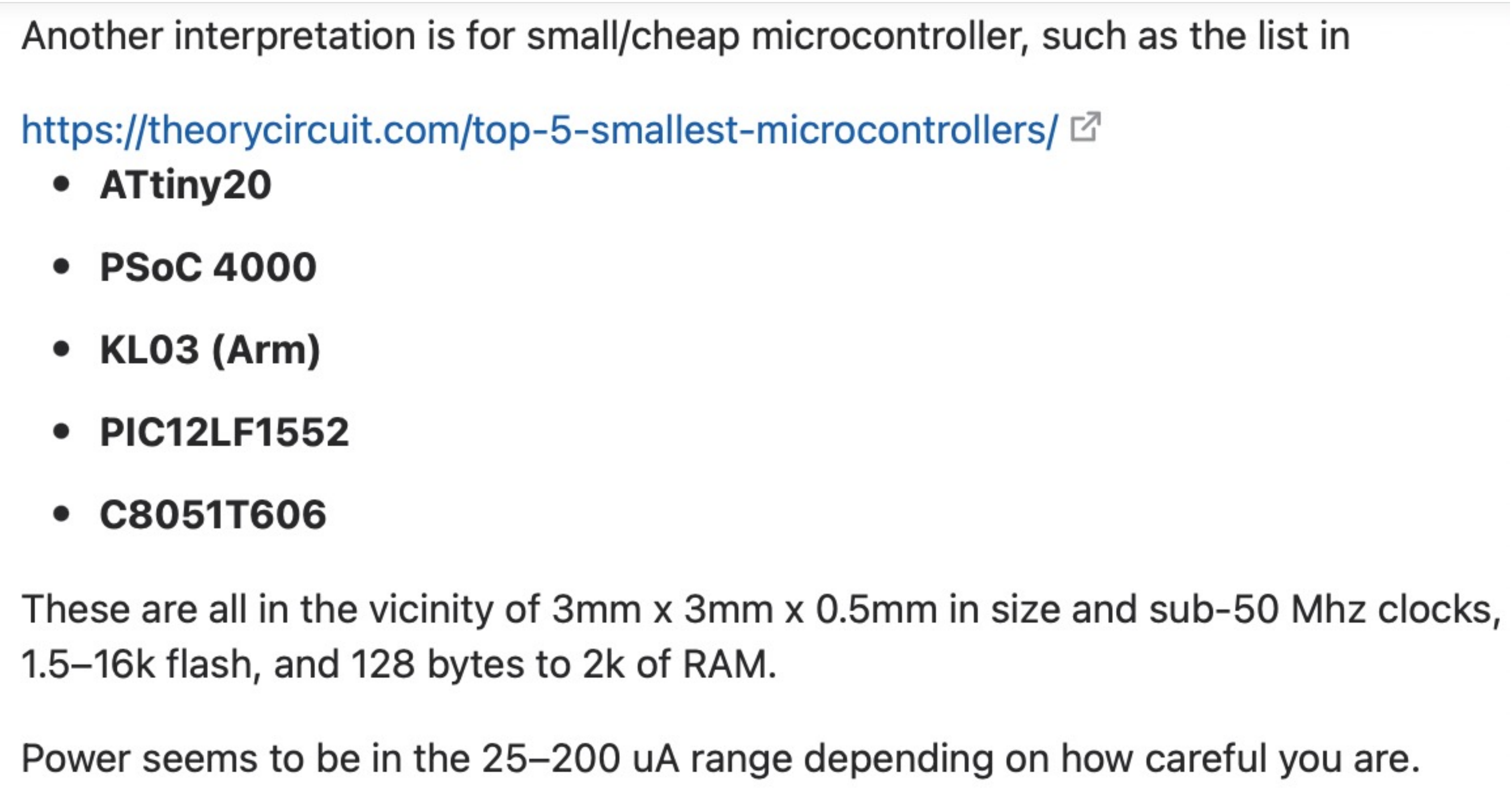
MPU

1985 **32/64-bit**

MCU 29K, i960, **ARM,** **PIC**

Small/Cheap MCU's

Quora



Another interpretation is for small/cheap microcontroller, such as the list in

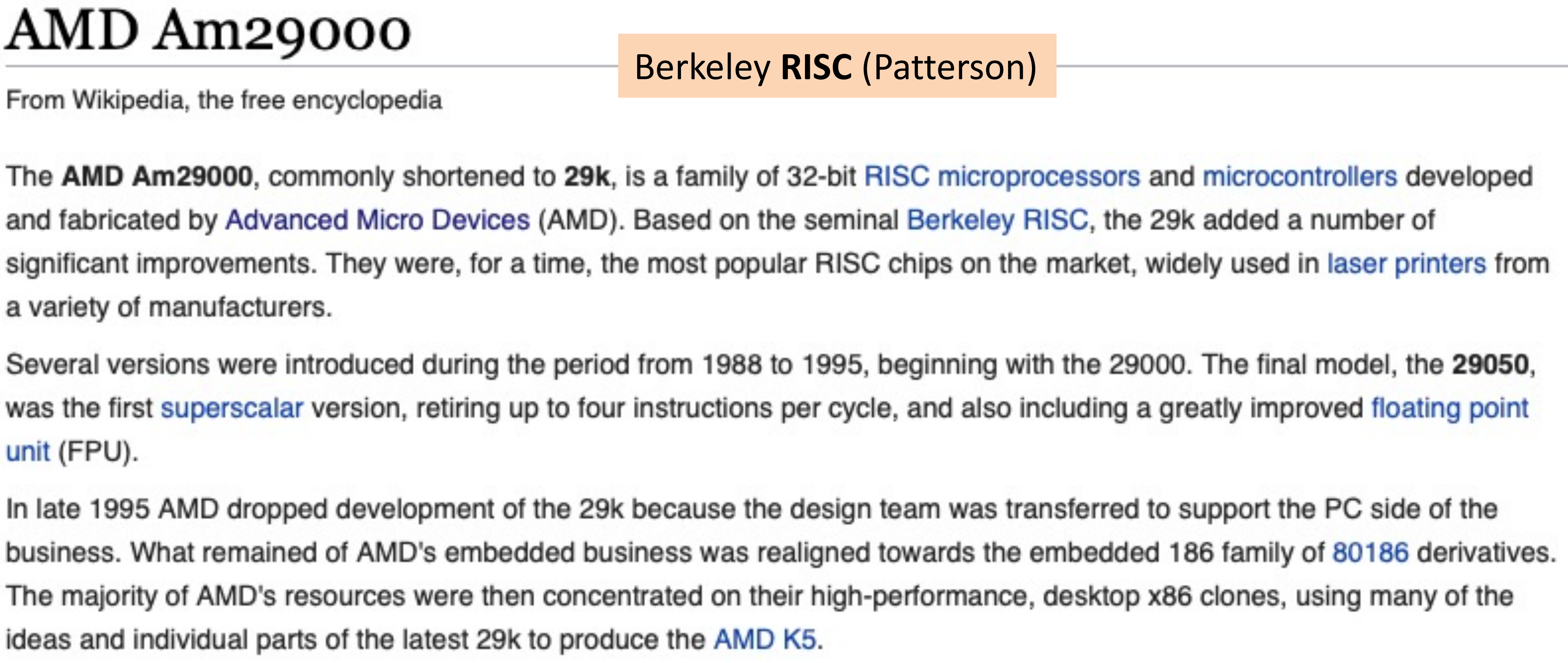
https://theorycircuit.com/top-5-smallest-microcontrollers/

- **ATtiny20**
- **PSoC 4000**
- **KL03 (Arm)**
- **PIC12LF1552**
- **C8051T606**

These are all in the vicinity of 3mm x 3mm x 0.5mm in size and sub-50 Mhz clocks, 1.5–16k flash, and 128 bytes to 2k of RAM.

Power seems to be in the 25–200 uA range depending on how careful you are.

Am29K



AMD Am29000

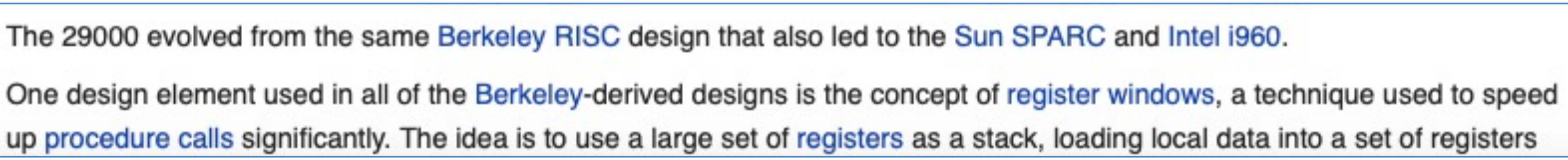
From Wikipedia, the free encyclopedia

The **AMD Am29000**, commonly shortened to **29k**, is a family of 32-bit RISC microprocessors and microcontrollers developed and fabricated by Advanced Micro Devices (AMD). Based on the seminal Berkeley RISC, the 29k added a number of significant improvements. They were, for a time, the most popular RISC chips on the market, widely used in laser printers from a variety of manufacturers.

Several versions were introduced during the period from 1988 to 1995, beginning with the 29000. The final model, the **29050**, was the first superscalar version, retiring up to four instructions per cycle, and also including a greatly improved floating point unit (FPU).

In late 1995 AMD dropped development of the 29k because the design team was transferred to support the PC side of the business. What remained of AMD's embedded business was realigned towards the embedded 186 family of 80186 derivatives. The majority of AMD's resources were then concentrated on their high-performance, desktop x86 clones, using many of the ideas and individual parts of the latest 29k to produce the AMD K5.

Berkeley **RISC** (Patterson)



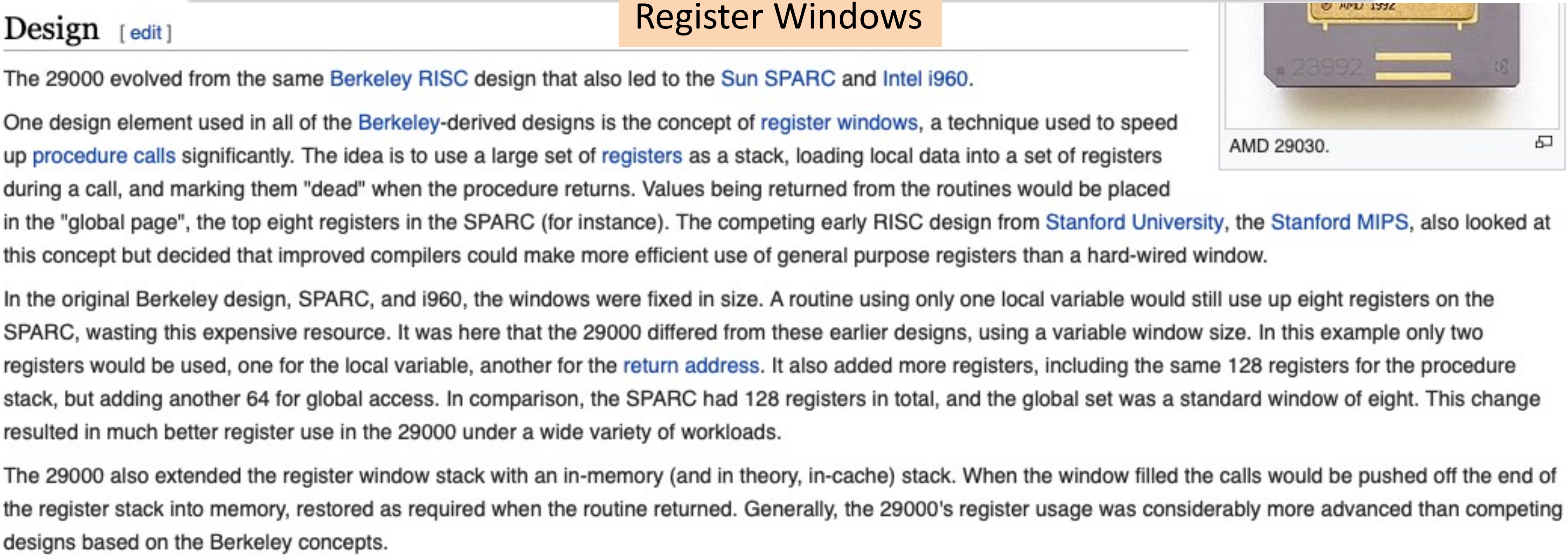
The 29000 evolved from the same Berkeley RISC design that also led to the Sun SPARC and Intel i960.

One design element used in all of the Berkeley-derived designs is the concept of register windows, a technique used to speed up procedure calls significantly. The idea is to use a large set of registers as a stack, loading local data into a set of registers

Register Windows

Am29K

Register Windows



Design [edit]

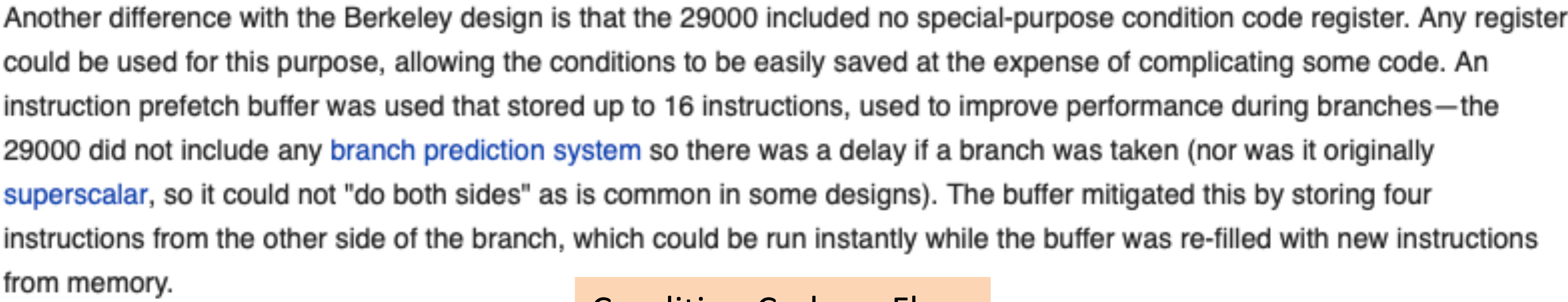
The 29000 evolved from the same Berkeley RISC design that also led to the Sun SPARC and Intel i960.

One design element used in all of the Berkeley-derived designs is the concept of register windows, a technique used to speed up procedure calls significantly. The idea is to use a large set of registers as a stack, loading local data into a set of registers during a call, and marking them "dead" when the procedure returns. Values being returned from the routines would be placed in the "global page", the top eight registers in the SPARC (for instance). The competing early RISC design from Stanford University, the Stanford MIPS, also looked at this concept but decided that improved compilers could make more efficient use of general purpose registers than a hard-wired window.

AMD 29030.

In the original Berkeley design, SPARC, and i960, the windows were fixed in size. A routine using only one local variable would still use up eight registers on the SPARC, wasting this expensive resource. It was here that the 29000 differed from these earlier designs, using a variable window size. In this example only two registers would be used, one for the local variable, another for the return address. It also added more registers, including the same 128 registers for the procedure stack, but adding another 64 for global access. In comparison, the SPARC had 128 registers in total, and the global set was a standard window of eight. This change resulted in much better register use in the 29000 under a wide variety of workloads.

The 29000 also extended the register window stack with an in-memory (and in theory, in-cache) stack. When the window filled the calls would be pushed off the end of the register stack into memory, restored as required when the routine returned. Generally, the 29000's register usage was considerably more advanced than competing designs based on the Berkeley concepts.



Another difference with the Berkeley design is that the 29000 included no special-purpose condition code register. Any register could be used for this purpose, allowing the conditions to be easily saved at the expense of complicating some code. An instruction prefetch buffer was used that stored up to 16 instructions, used to improve performance during branches—the 29000 did not include any branch prediction system so there was a delay if a branch was taken (nor was it originally superscalar, so it could not "do both sides" as is common in some designs). The buffer mitigated this by storing four instructions from the other side of the branch, which could be run instantly while the buffer was re-filled with new instructions from memory.

Condition Codes = Flags

Am29K

AMD 29000 Microprocessor

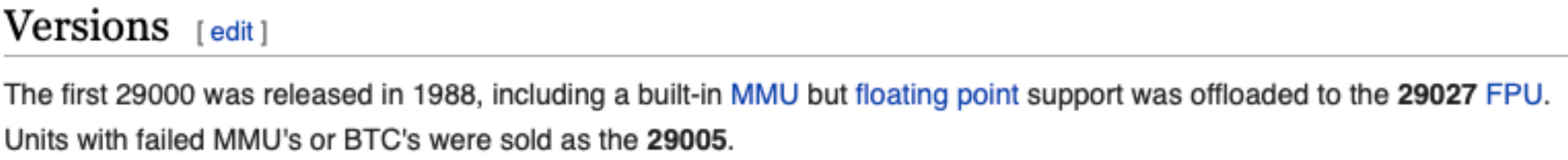
1988

AMD 29030.

1991-2

AMD 29040

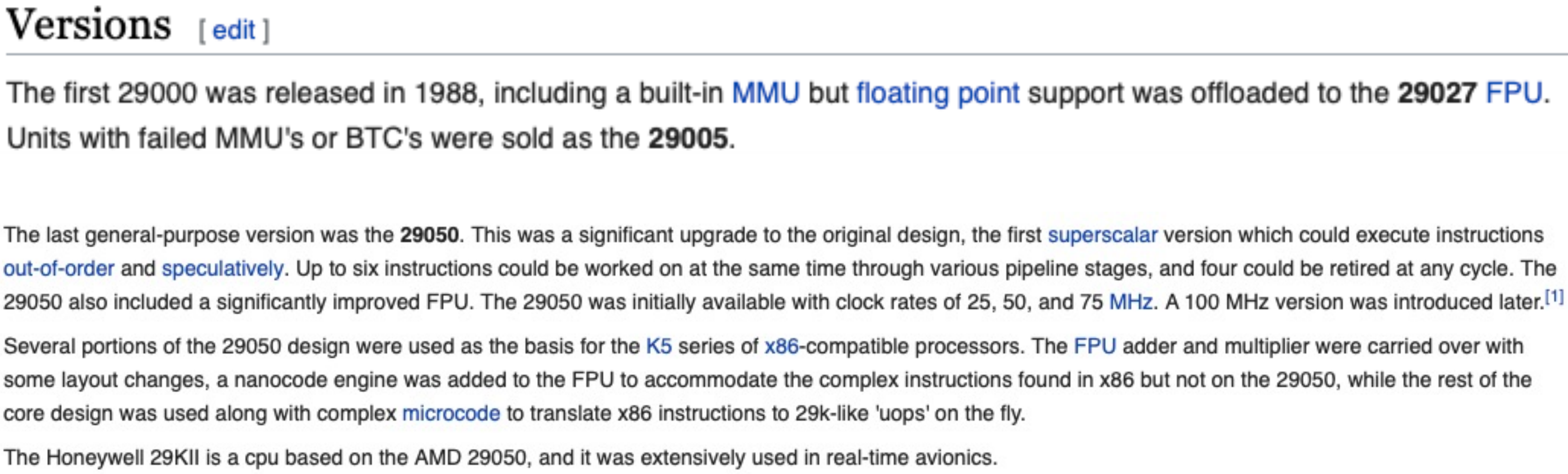
1994-5



Versions [edit]

The first 29000 was released in 1988, including a built-in MMU but floating point support was offloaded to the **29027** FPU. Units with failed MMU's or BTC's were sold as the **29005**.

Am29K



Versions [edit]

The first 29000 was released in 1988, including a built-in MMU but floating point support was offloaded to the **29027** FPU. Units with failed MMU's or BTC's were sold as the **29005**.

The last general-purpose version was the **29050**. This was a significant upgrade to the original design, the first superscalar version which could execute instructions out-of-order and speculatively. Up to six instructions could be worked on at the same time through various pipeline stages, and four could be retired at any cycle. The 29050 also included a significantly improved FPU. The 29050 was initially available with clock rates of 25, 50, and 75 MHz. A 100 MHz version was introduced later.[1]

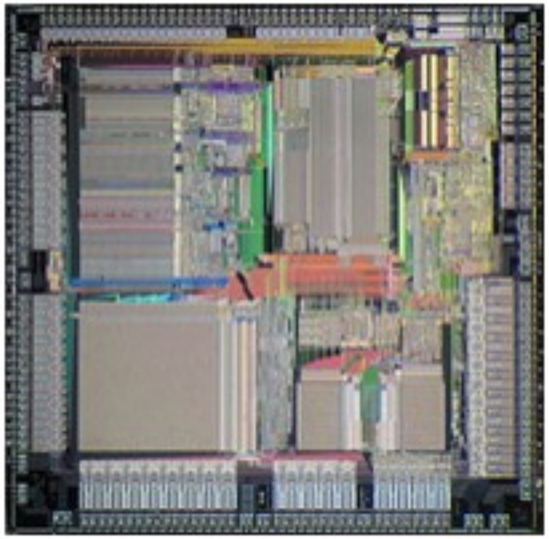
Several portions of the 29050 design were used as the basis for the K5 series of x86-compatible processors. The FPU adder and multiplier were carried over with some layout changes, a nanocode engine was added to the FPU to accommodate the complex instructions found in x86 but not on the 29050, while the rest of the core design was used along with complex microcode to translate x86 instructions to 29k-like 'uops' on the fly.

The Honeywell 29KII is a cpu based on the AMD 29050, and it was extensively used in real-time avionics.

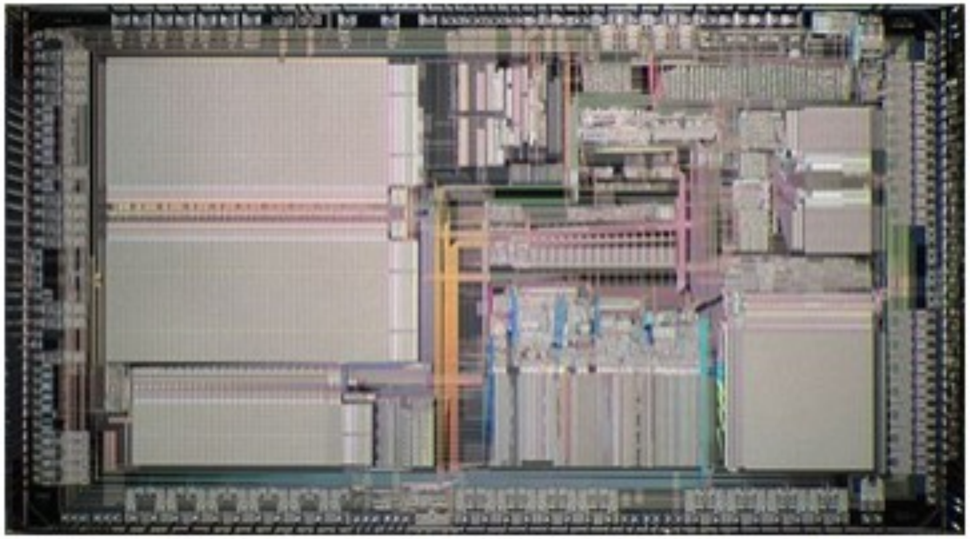
29050 → K5 (x86 Pentium)

Am29K

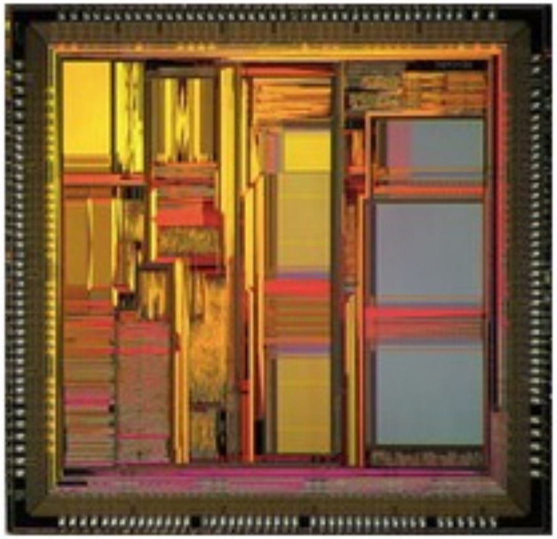
Die photos



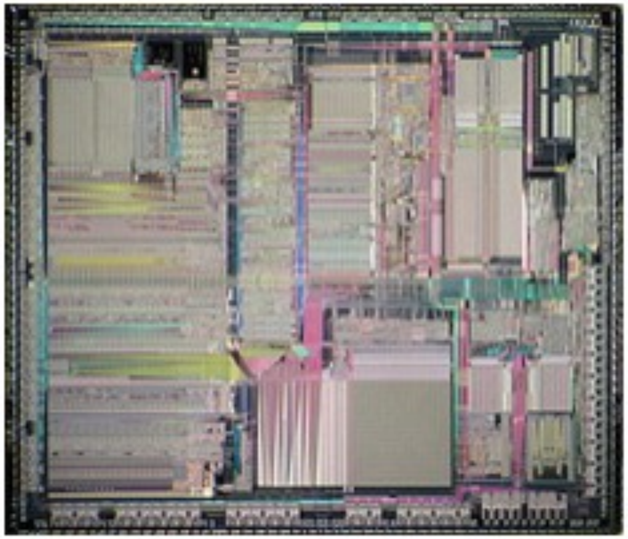
Am29000



Am29030

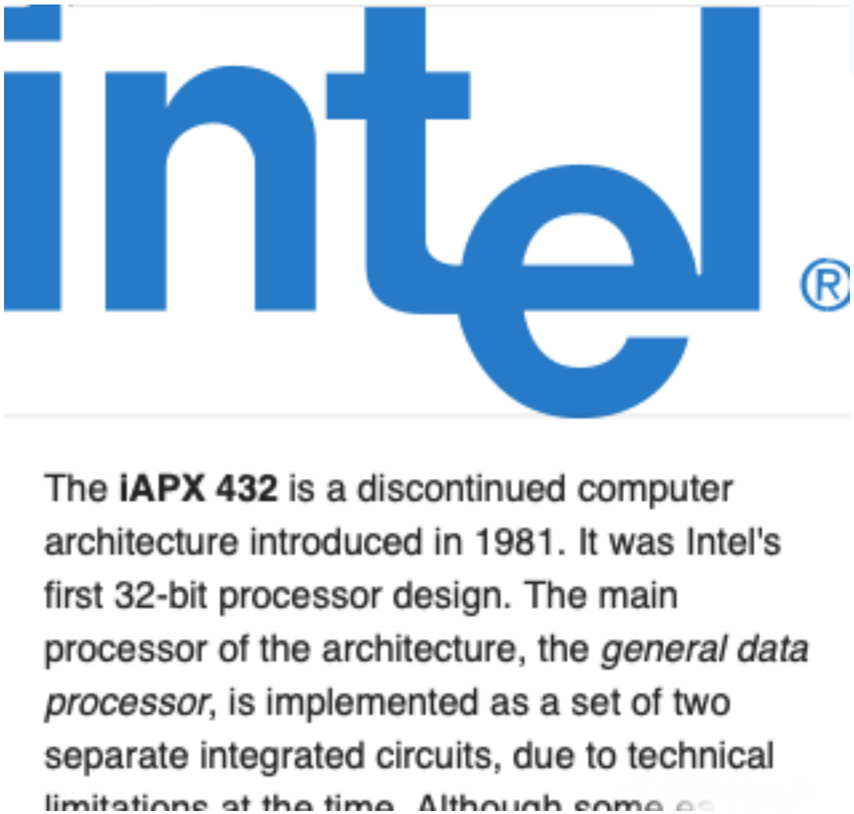


Am29040



Am29050

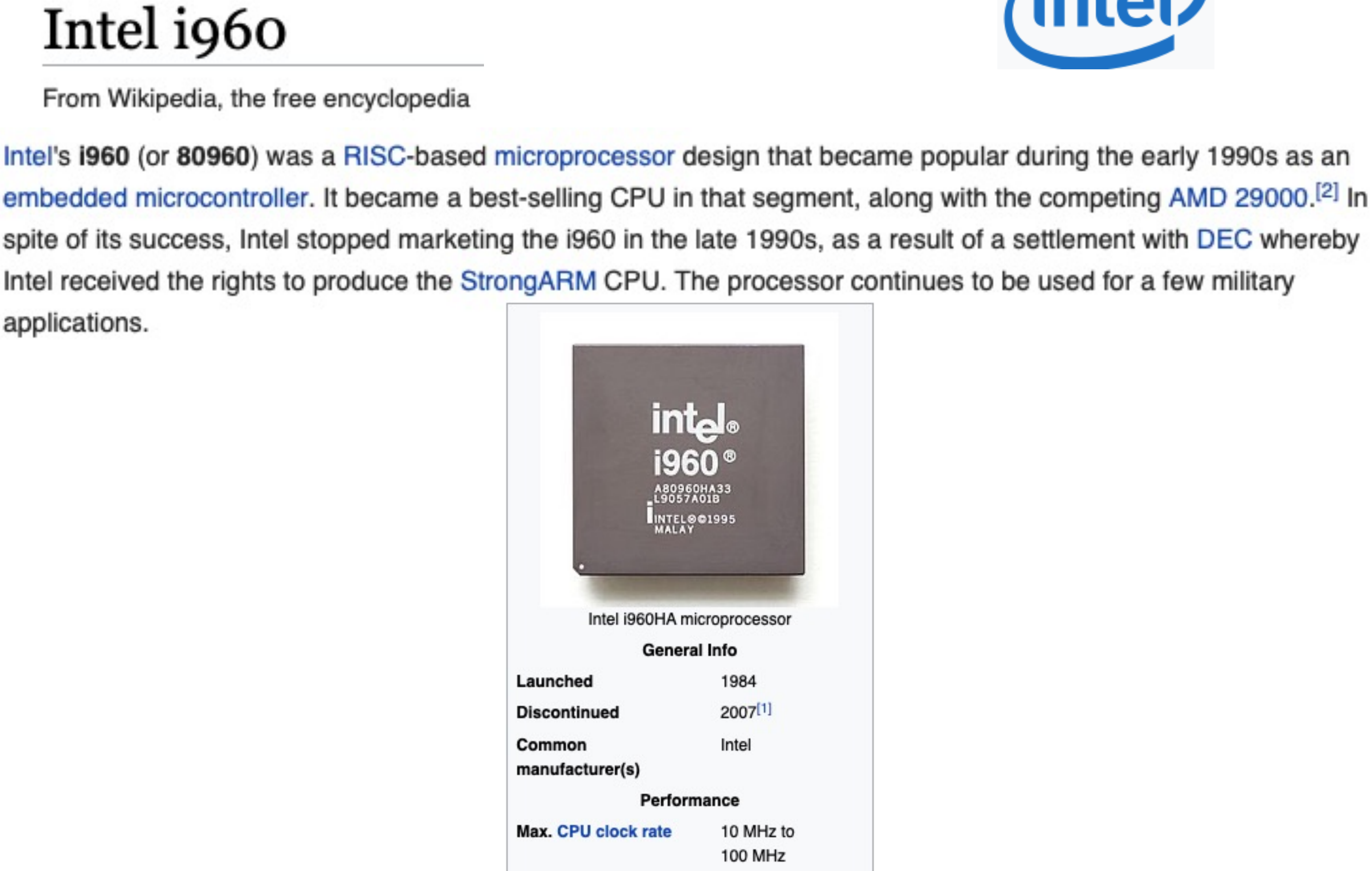
iAPX 432



The **iAPX 432** is a discontinued computer architecture introduced in 1981. It was Intel's first 32-bit processor design. The main processor of the architecture, the *general data processor*, is implemented as a set of two separate integrated circuits, due to technical limitations at the time. Although some e

Forerunner of **i960**

i960



Intel i960

From Wikipedia, the free encyclopedia

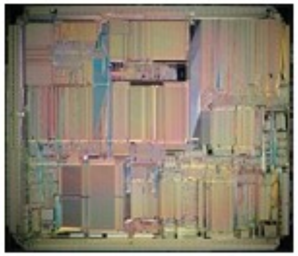
Intel's **i960** (or **80960**) was a RISC-based microprocessor design that became popular during the early 1990s as an embedded microcontroller. It became a best-selling CPU in that segment, along with the competing AMD 29000.[2] In spite of its success, Intel stopped marketing the i960 in the late 1990s, as a result of a settlement with DEC whereby Intel received the rights to produce the StrongARM CPU. The processor continues to be used for a few military applications.

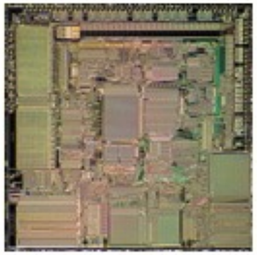
Intel i960HA microprocessor

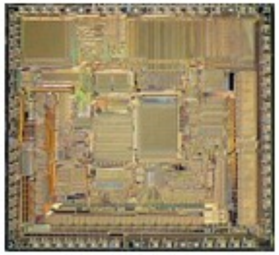
General Info	
Launched	1984
Discontinued	2007[1]
Common manufacturer(s)	Intel
Performance	
Max. CPU clock rate	10 MHz to 100 MHz

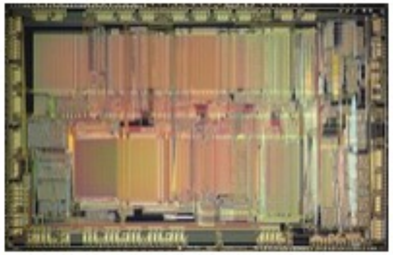
i960

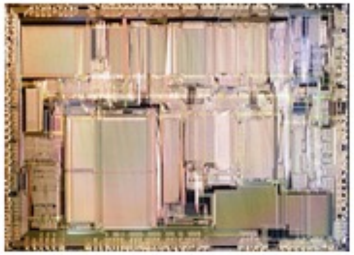
Die photos

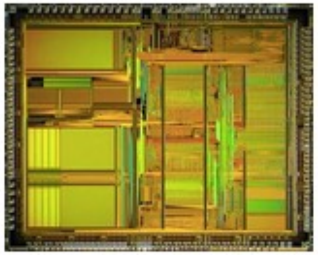

Intel 80960MX


Intel 80960KA


Intel 80960SA

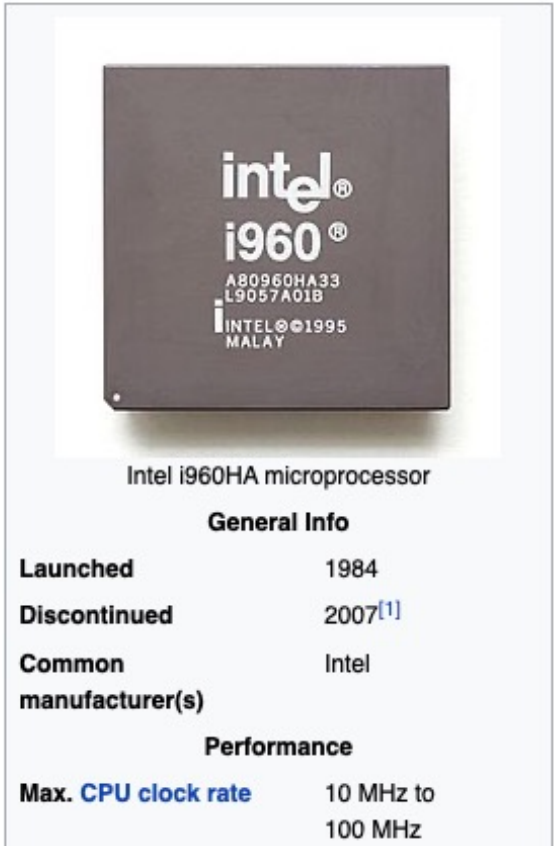

Intel 80960CA


Intel 80960CF


Intel 80960JA

Intel N80960SA (PLCC Package)

Intel GC80960RD66 (BGA Package)


Intel i960HA microprocessor

General Info

Launched	1984
Discontinued	2007[1]
Common manufacturer(s)	Intel

Performance

Max. CPU clock rate	10 MHz to 100 MHz

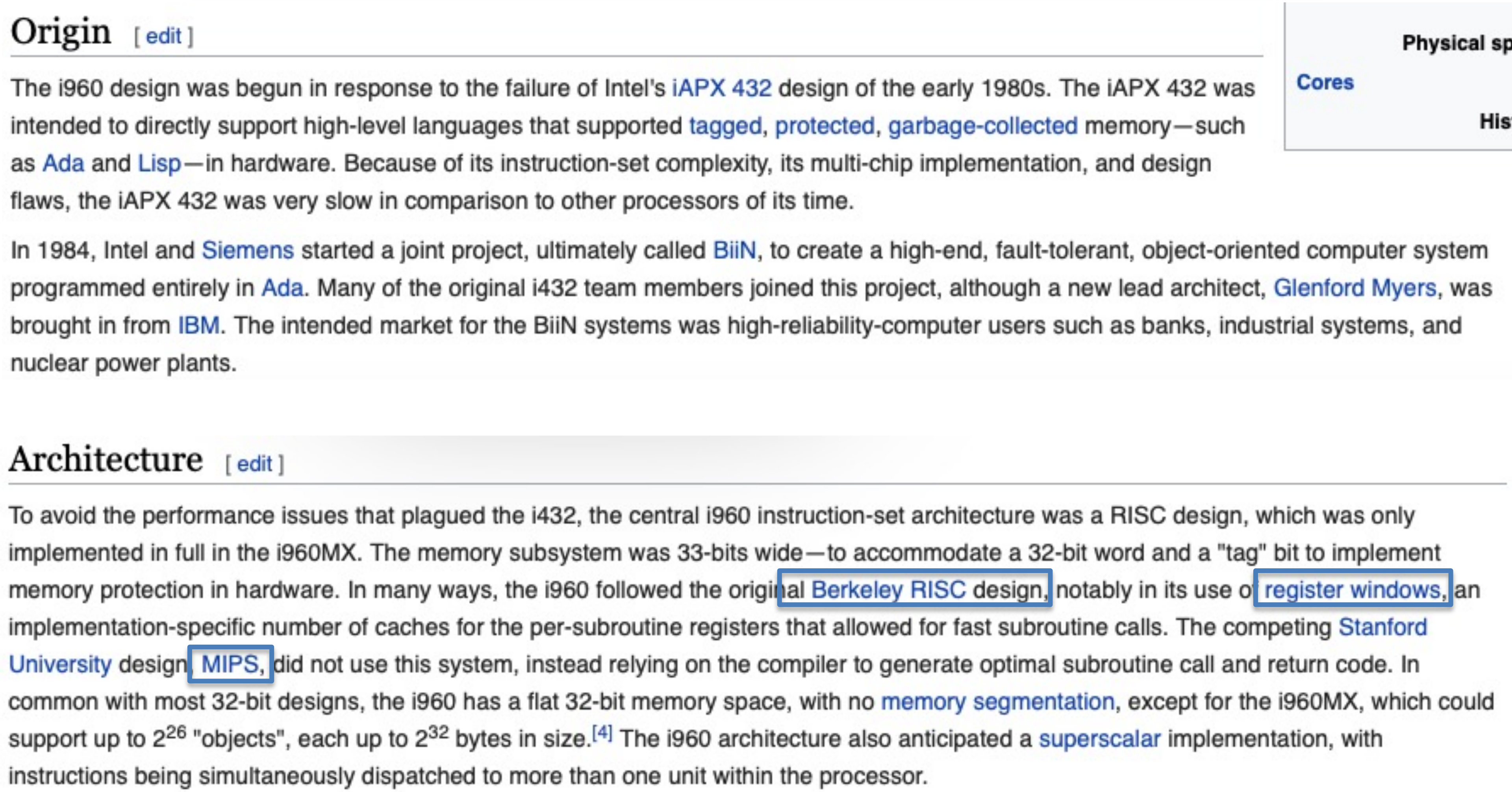
Intel GC80960RN, sSpec: SL3YW, BGA Package

Intel

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

i960

DR JEFF SOFTWARE
INDIE APP DEVELOPER



Origin [edit]

Physical sp
Cores
His

The i960 design was begun in response to the failure of Intel's iAPX 432 design of the early 1980s. The iAPX 432 was intended to directly support high-level languages that supported tagged, protected, garbage-collected memory—such as Ada and Lisp—in hardware. Because of its instruction-set complexity, its multi-chip implementation, and design flaws, the iAPX 432 was very slow in comparison to other processors of its time.

In 1984, Intel and Siemens started a joint project, ultimately called BiiN, to create a high-end, fault-tolerant, object-oriented computer system programmed entirely in Ada. Many of the original i432 team members joined this project, although a new lead architect, Glenford Myers, was brought in from IBM. The intended market for the BiiN systems was high-reliability-computer users such as banks, industrial systems, and nuclear power plants.

Architecture [edit]

To avoid the performance issues that plagued the i432, the central i960 instruction-set architecture was a RISC design, which was only implemented in full in the i960MX. The memory subsystem was 33-bits wide—to accommodate a 32-bit word and a "tag" bit to implement memory protection in hardware. In many ways, the i960 followed the original Berkeley RISC design, notably in its use of register windows, an implementation-specific number of caches for the per-subroutine registers that allowed for fast subroutine calls. The competing Stanford University design, MIPS, did not use this system, instead relying on the compiler to generate optimal subroutine call and return code. In common with most 32-bit designs, the i960 has a flat 32-bit memory space, with no memory segmentation, except for the i960MX, which could support up to 2^{26} "objects", each up to 2^{32} bytes in size.[4] The i960 architecture also anticipated a superscalar implementation, with instructions being simultaneously dispatched to more than one unit within the processor.

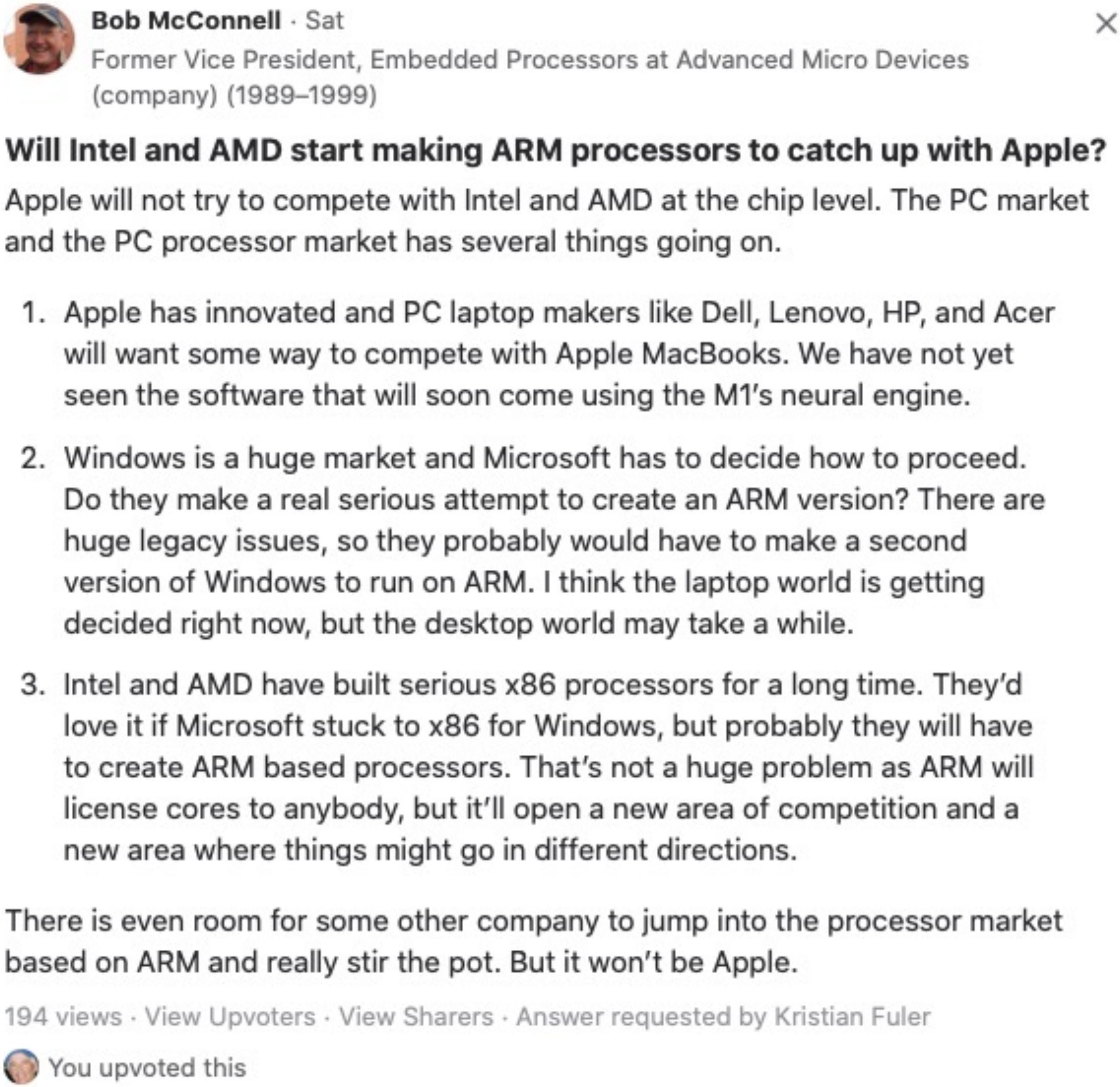
Modern CPU's

Modern CPU's

Apple A & M-series

(ARM v8)

- ➢ See separate slide set "SoC's"



Bob McConnell · Sat
Former Vice President, Embedded Processors at Advanced Micro Devices (company) (1989–1999)

Will Intel and AMD start making ARM processors to catch up with Apple?

Apple will not try to compete with Intel and AMD at the chip level. The PC market and the PC processor market has several things going on.

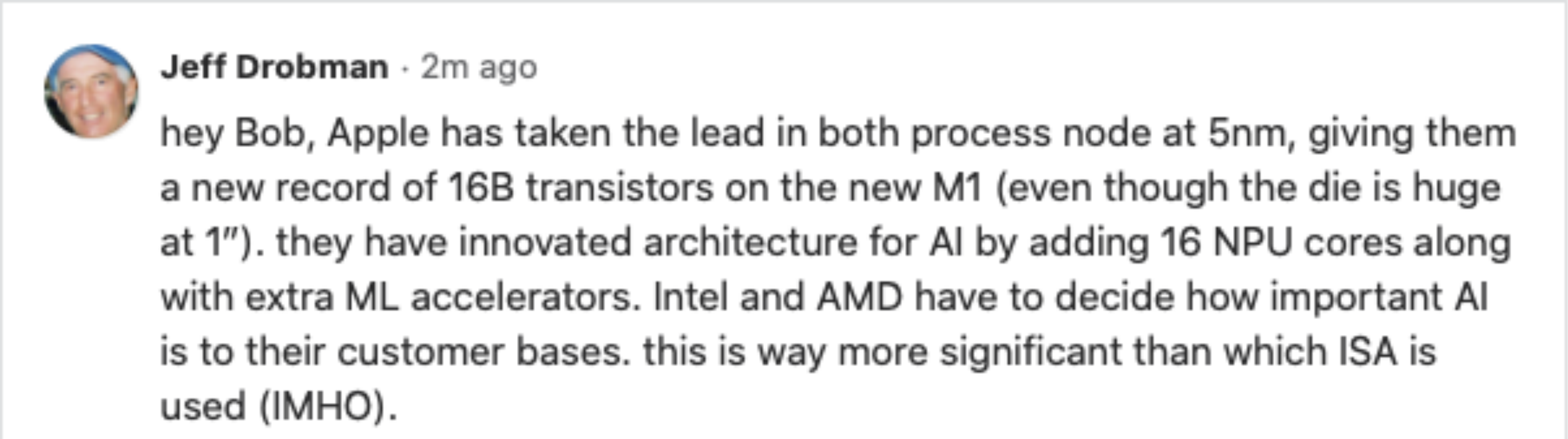
1. Apple has innovated and PC laptop makers like Dell, Lenovo, HP, and Acer will want some way to compete with Apple MacBooks. We have not yet seen the software that will soon come using the M1's neural engine.
2. Windows is a huge market and Microsoft has to decide how to proceed. Do they make a real serious attempt to create an ARM version? There are huge legacy issues, so they probably would have to make a second version of Windows to run on ARM. I think the laptop world is getting decided right now, but the desktop world may take a while.
3. Intel and AMD have built serious x86 processors for a long time. They'd love it if Microsoft stuck to x86 for Windows, but probably they will have to create ARM based processors. That's not a huge problem as ARM will license cores to anybody, but it'll open a new area of competition and a new area where things might go in different directions.

There is even room for some other company to jump into the processor market based on ARM and really stir the pot. But it won't be Apple.

194 views · View Upvoters · View Sharers · Answer requested by Kristian Fuler

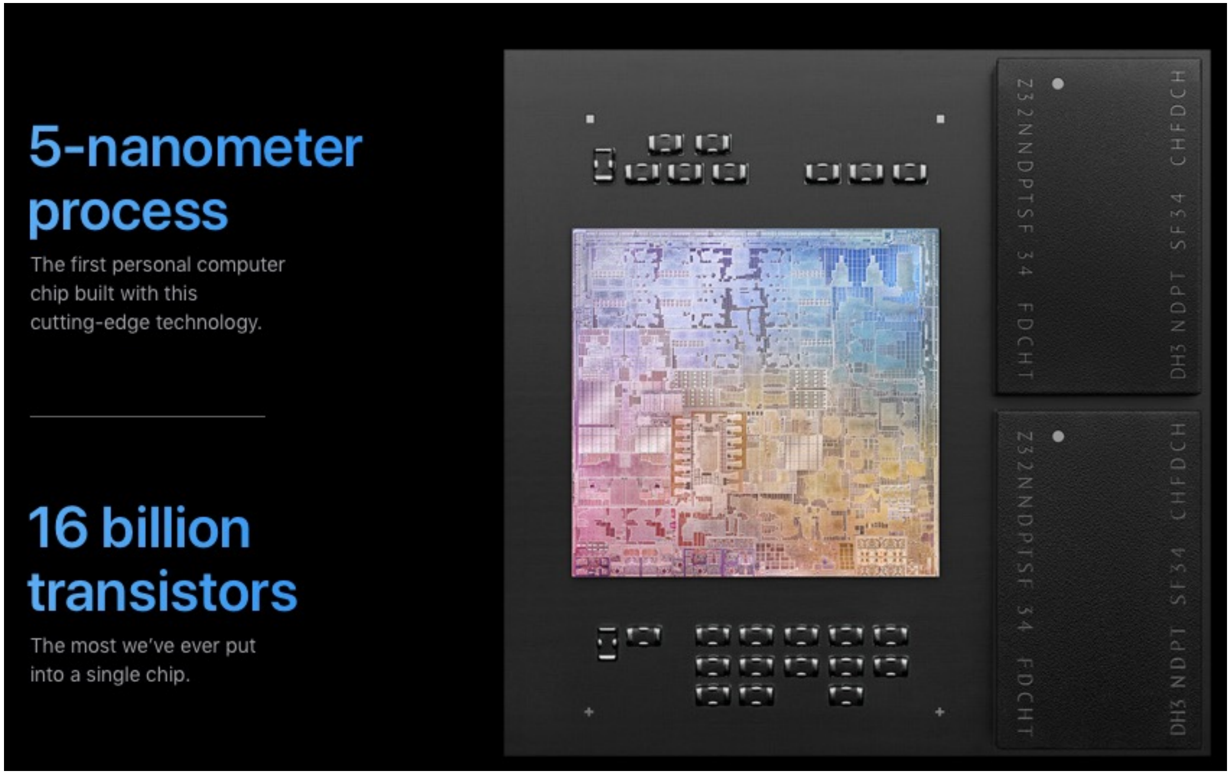
You upvoted this

My Response to Apple



Jeff Drobman · 2m ago

hey Bob, Apple has taken the lead in both process node at 5nm, giving them a new record of 16B transistors on the new M1 (even though the die is huge at 1"). they have innovated architecture for AI by adding 16 NPU cores along with extra ML accelerators. Intel and AMD have to decide how important AI is to their customer bases. this is way more significant than which ISA is used (IMHO).



5-nanometer process

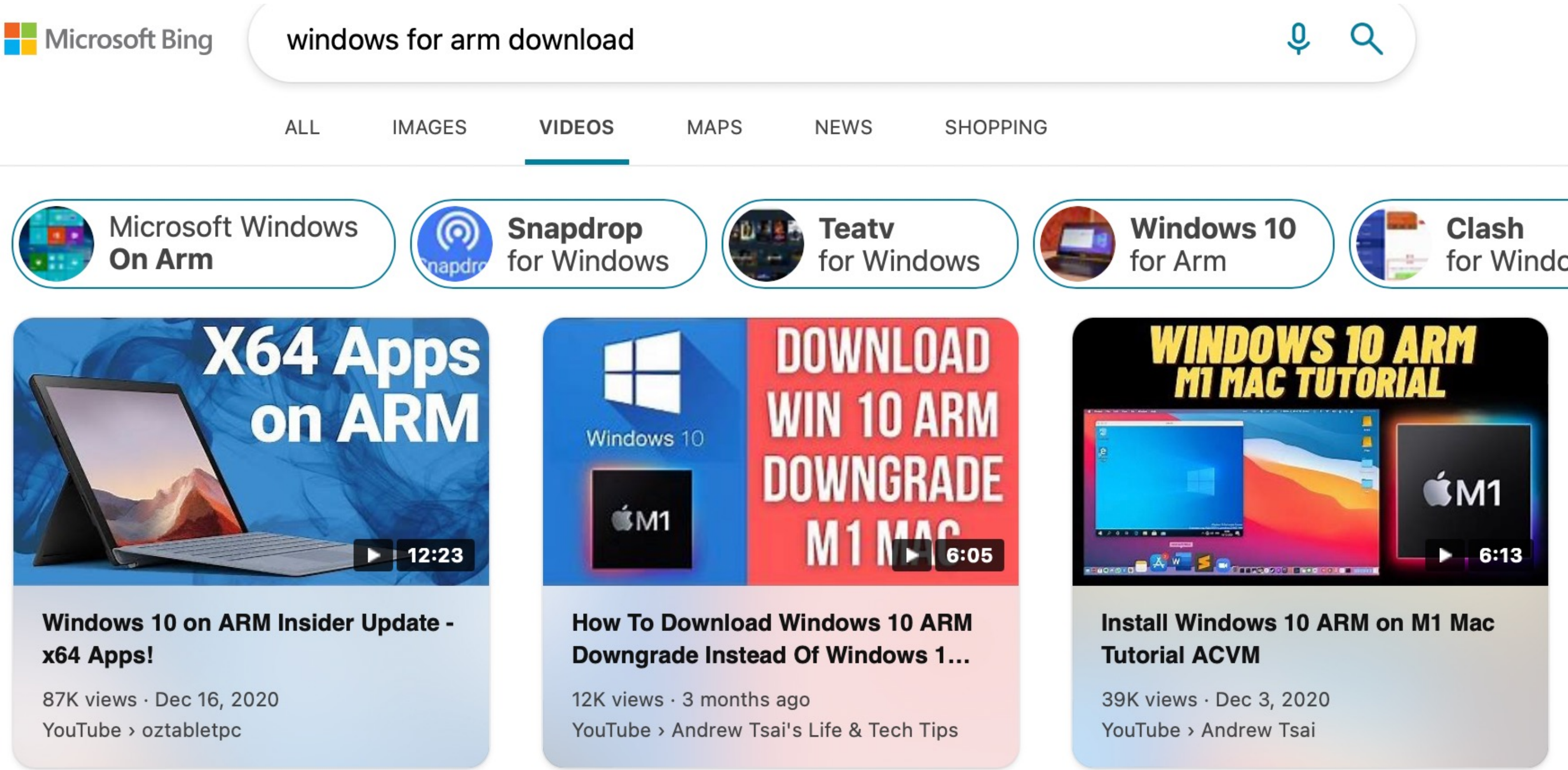
The first personal computer chip built with this cutting-edge technology.

16 billion transistors

The most we've ever put into a single chip.

625 sq mm?
(= 1 sq in)

Windows on ARM M1



Microsoft Bing

windows for arm download

ALL IMAGES **VIDEOS** MAPS NEWS SHOPPING

Microsoft Windows **On Arm**

Snapdrop for Windows

Teatv for Windows

Windows 10 for Arm

Clash for Windo

Windows 10 on ARM Insider Update - x64 Apps!

87K views · Dec 16, 2020

YouTube › oztabletpc

How To Download Windows 10 ARM Downgrade Instead Of Windows 1...

12K views · 3 months ago

YouTube › Andrew Tsai's Life & Tech Tips

Install Windows 10 ARM on M1 Mac Tutorial ACVM

39K views · Dec 3, 2020

YouTube › Andrew Tsai

Modern CPU's

Modern CPU's

Qualcomm

Snapdragon

- ➢ See separate slide set "SoC's"

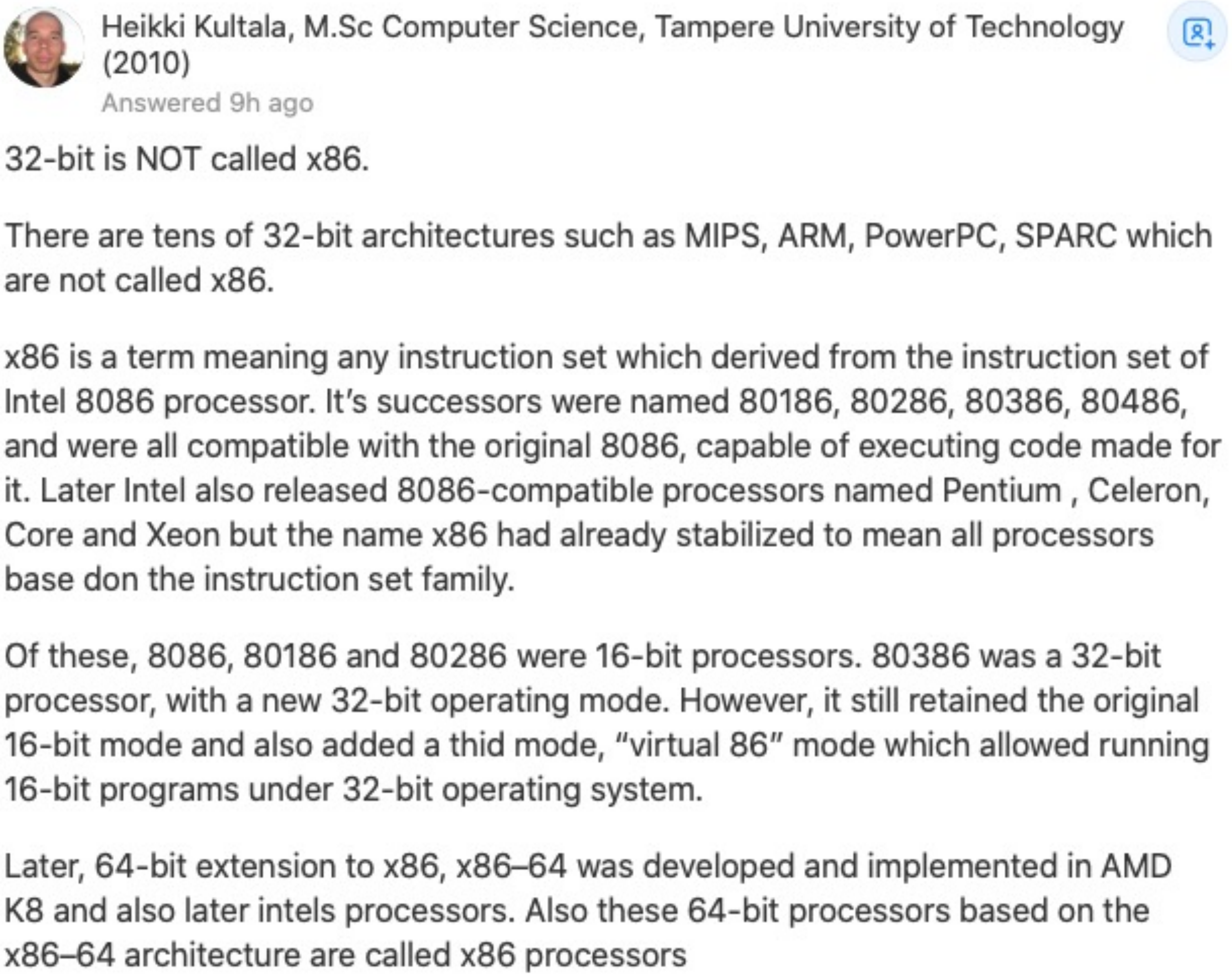
Modern CPU's

Modern CPU's

x86 Pentiums

- See separate slide set "x86"

x86 History



Heikki Kultala, M.Sc Computer Science, Tampere University of Technology (2010)

Answered 9h ago

32-bit is NOT called x86.

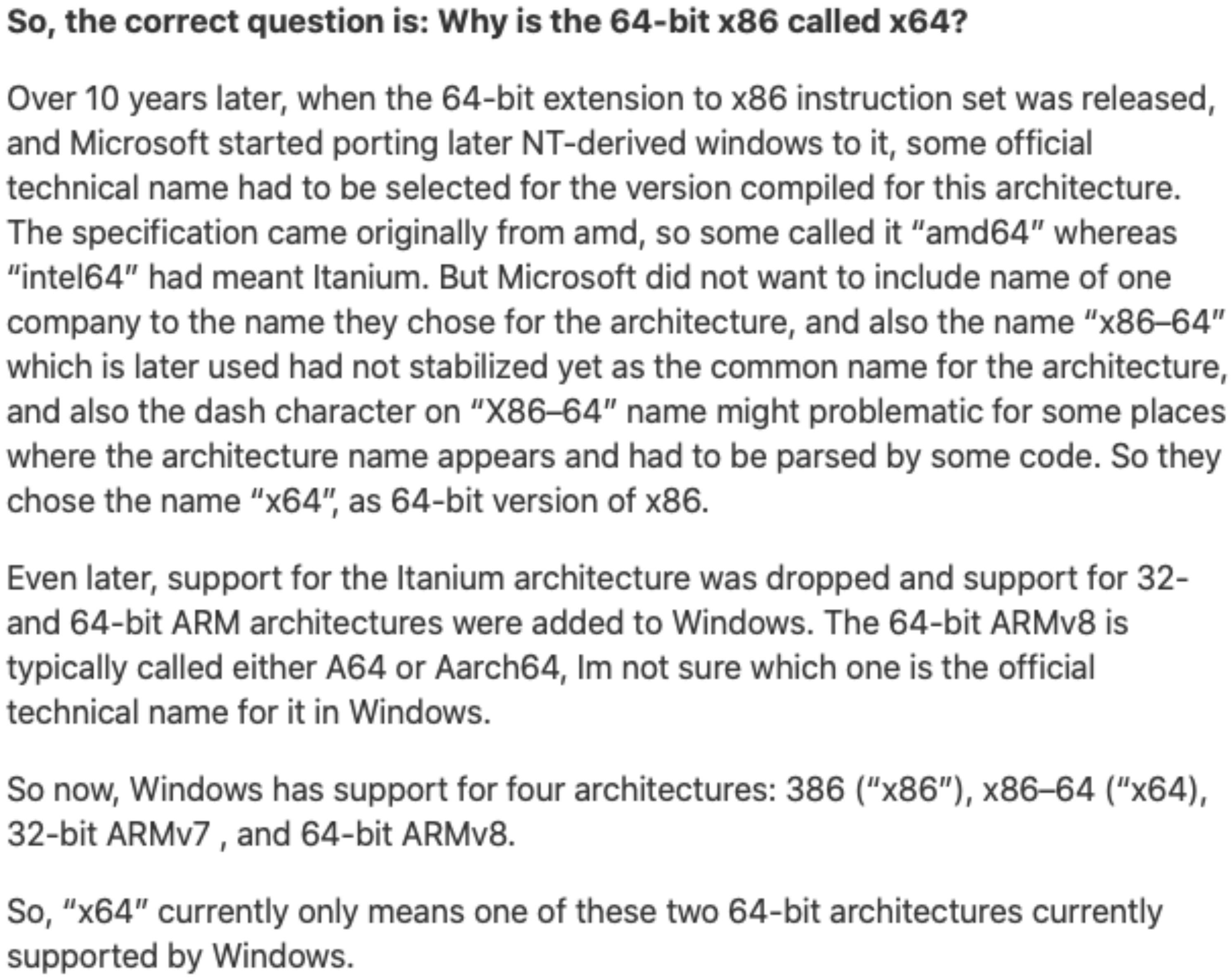
There are tens of 32-bit architectures such as MIPS, ARM, PowerPC, SPARC which are not called x86.

x86 is a term meaning any instruction set which derived from the instruction set of Intel 8086 processor. It's successors were named 80186, 80286, 80386, 80486, and were all compatible with the original 8086, capable of executing code made for it. Later Intel also released 8086-compatible processors named Pentium , Celeron, Core and Xeon but the name x86 had already stabilized to mean all processors base don the instruction set family.

Of these, 8086, 80186 and 80286 were 16-bit processors. 80386 was a 32-bit processor, with a new 32-bit operating mode. However, it still retained the original 16-bit mode and also added a thid mode, "virtual 86" mode which allowed running 16-bit programs under 32-bit operating system.

Later, 64-bit extension to x86, x86–64 was developed and implemented in AMD K8 and also later intels processors. Also these 64-bit processors based on the x86–64 architecture are called x86 processors

x86 History



So, the correct question is: Why is the 64-bit x86 called x64?

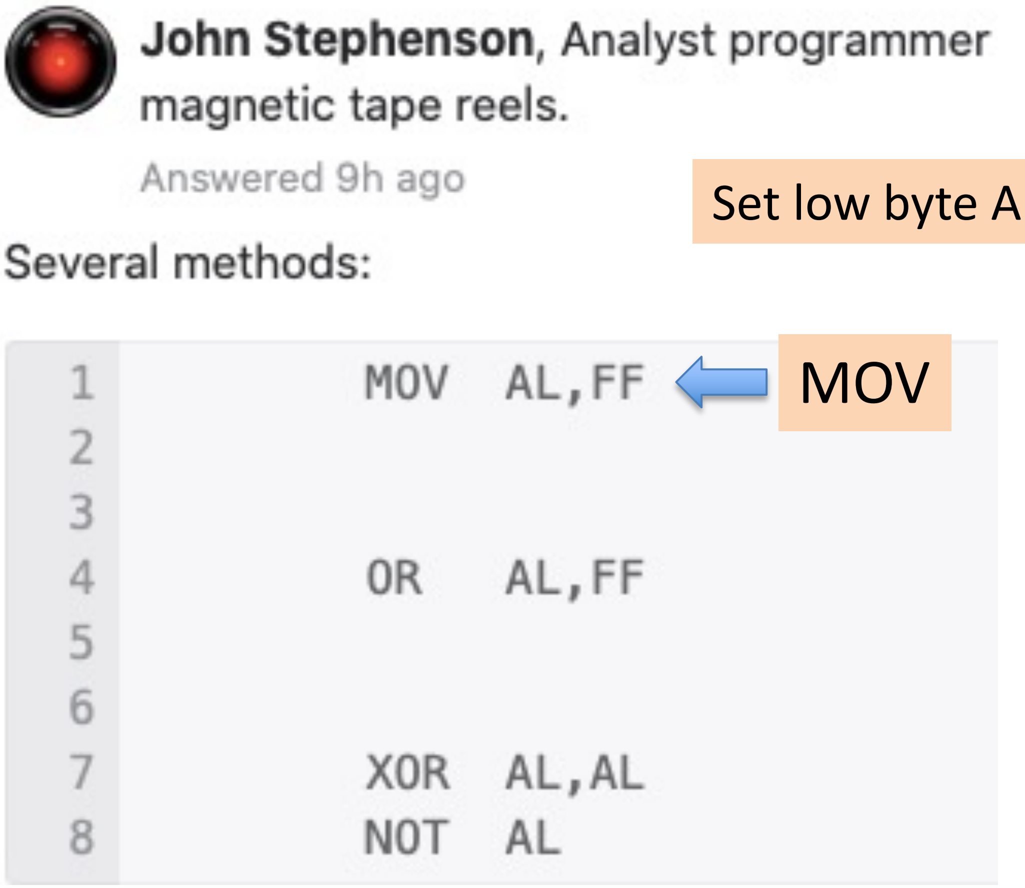
Over 10 years later, when the 64-bit extension to x86 instruction set was released, and Microsoft started porting later NT-derived windows to it, some official technical name had to be selected for the version compiled for this architecture. The specification came originally from amd, so some called it “amd64” whereas “intel64” had meant Itanium. But Microsoft did not want to include name of one company to the name they chose for the architecture, and also the name “x86–64” which is later used had not stabilized yet as the common name for the architecture, and also the dash character on “X86–64” name might problematic for some places where the architecture name appears and had to be parsed by some code. So they chose the name “x64”, as 64-bit version of x86.

Even later, support for the Itanium architecture was dropped and support for 32- and 64-bit ARM architectures were added to Windows. The 64-bit ARMv8 is typically called either A64 or Aarch64, Im not sure which one is the official technical name for it in Windows.

So now, Windows has support for four architectures: 386 (“x86”), x86–64 (“x64), 32-bit ARMv7 , and 64-bit ARMv8.

So, “x64” currently only means one of these two 64-bit architectures currently supported by Windows.

i8080 Code



John Stephenson, Analyst programmer magnetic tape reels.

Answered 9h ago

Several methods:

```
            MOV   AL,FF


            OR    AL,FF


            XOR   AL,AL
            NOT   AL
```

Set low byte A = all 1’s or all 0’s

MOV

x86

The x86 architectures were based on the Intel 8086 microprocessor chip, initially released in 1978.

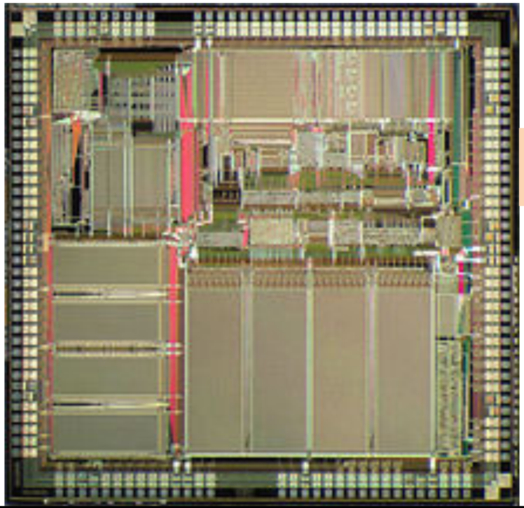
Intel Core 2 Duo – an example of an x86-compatible, 64-bit multicore processor

AMD Athlon (early version) – a technically different but fully compatible x86 implementation

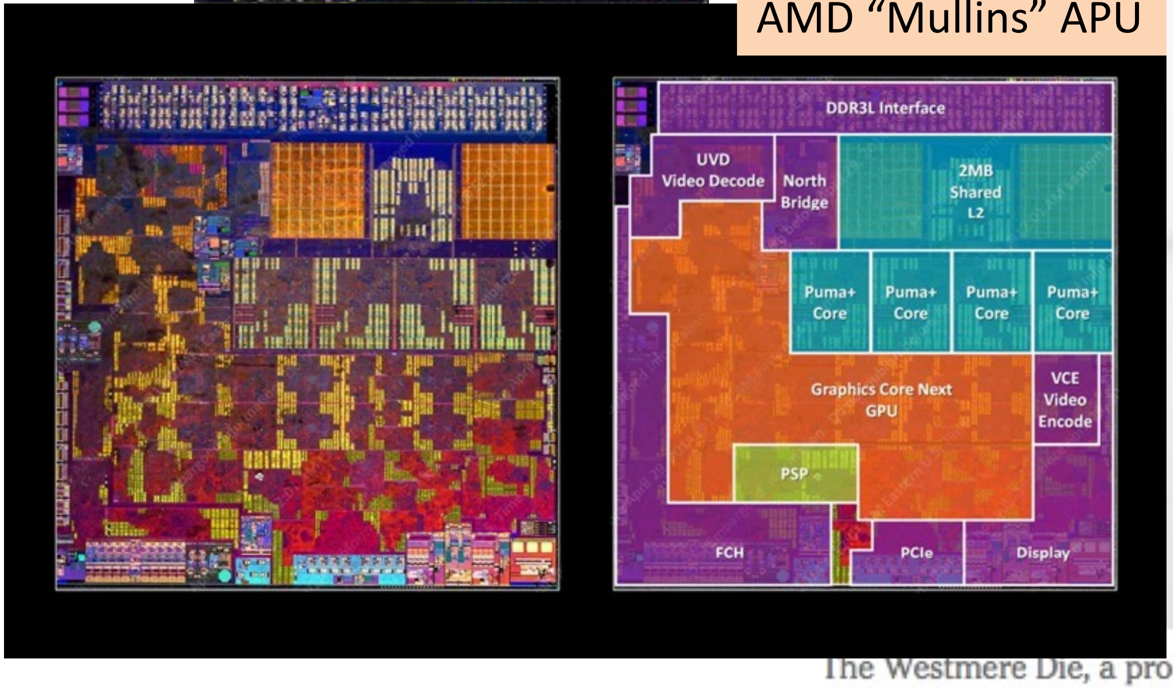
CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

Modern CPUs

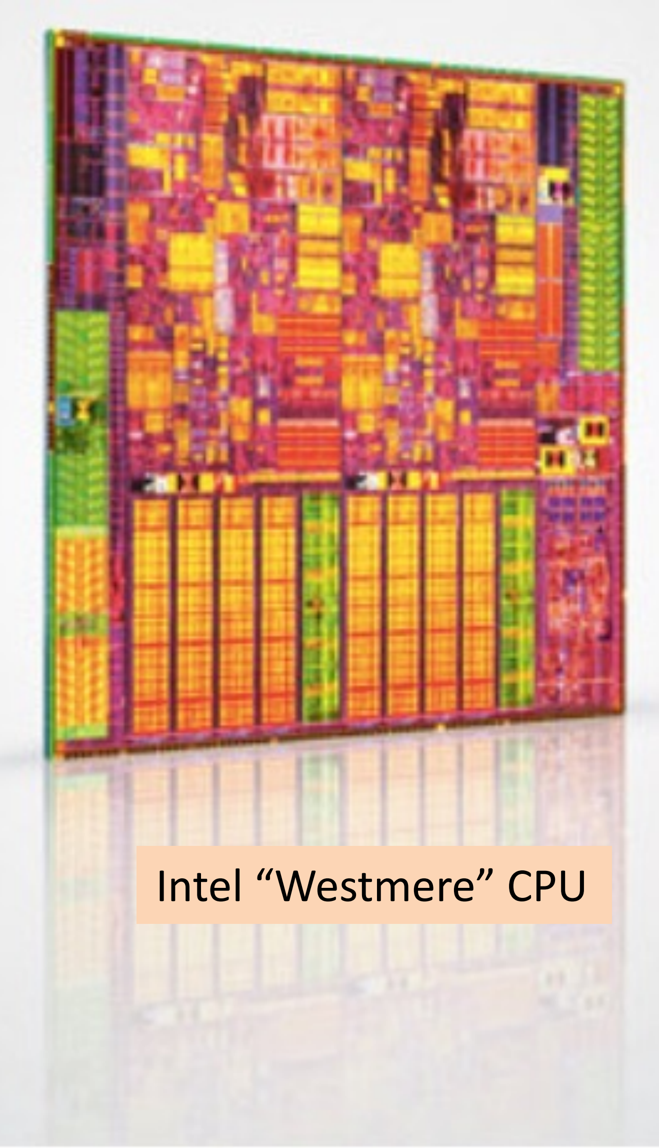
DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021



ARM 610



AMD “Mullins” APU

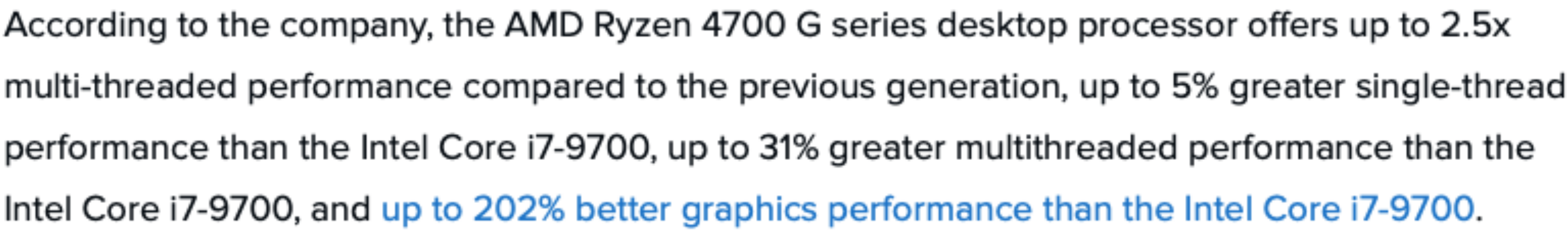


Intel “Westmere” CPU

The Westmere Die, a processor introduced by Intel in 2010. Intel

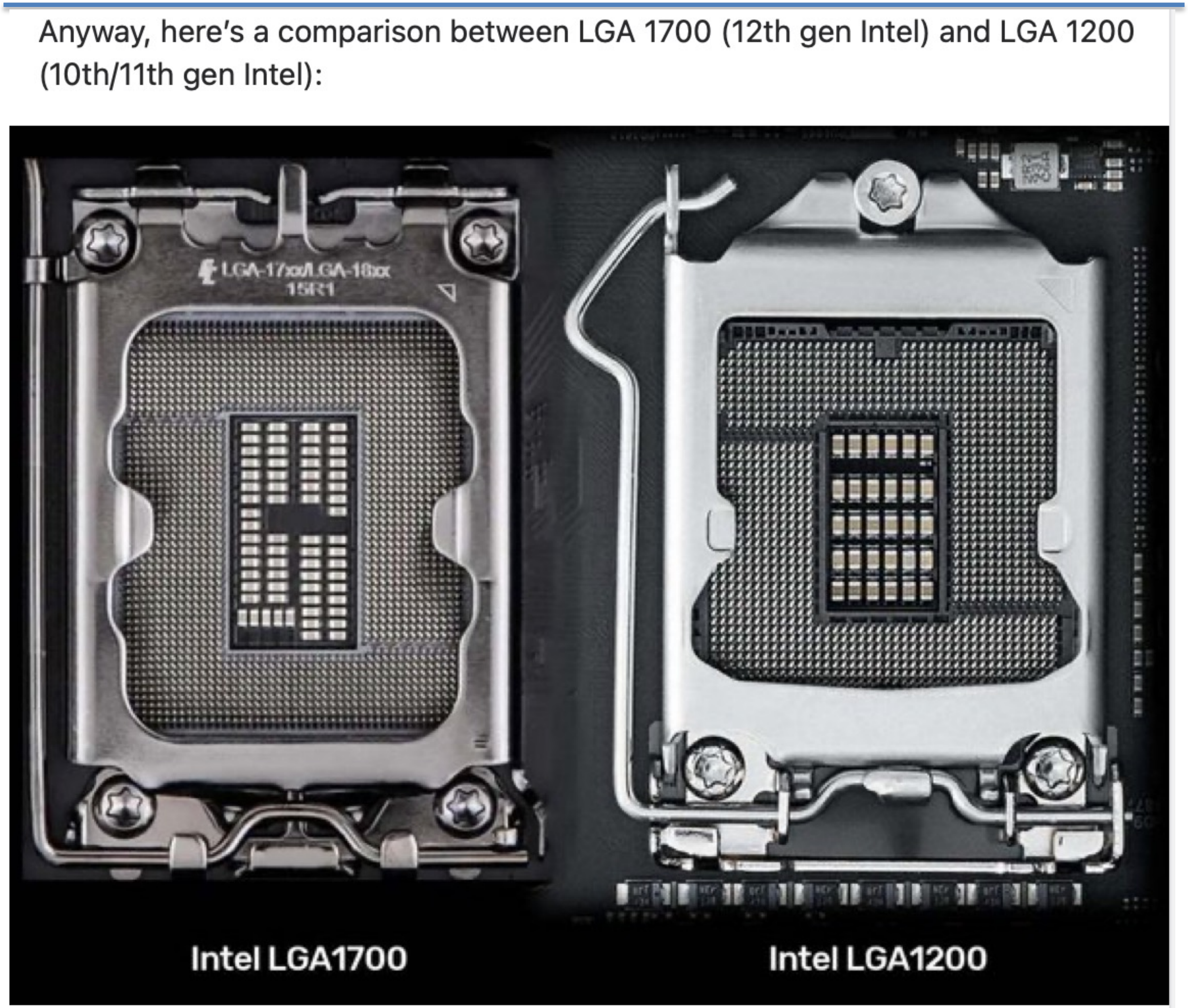
AMD vs Intel: CPU Families

Market Segment	AMD	Intel
Desktop	Ryzen 4K/Athlon 3K	Core i7/i9 (10th gen)
Laptop	Ryzen 4000	Ice Lake
Gaming	Ryzen Threadripper +Radeon	Core Extreme
Server/Workstn	Epyc	Xeon



According to the company, the AMD Ryzen 4700 G series desktop processor offers up to 2.5x multi-threaded performance compared to the previous generation, up to 5% greater single-thread performance than the Intel Core i7-9700, up to 31% greater multithreaded performance than the Intel Core i7-9700, and up to 202% better graphics performance than the Intel Core i7-9700.

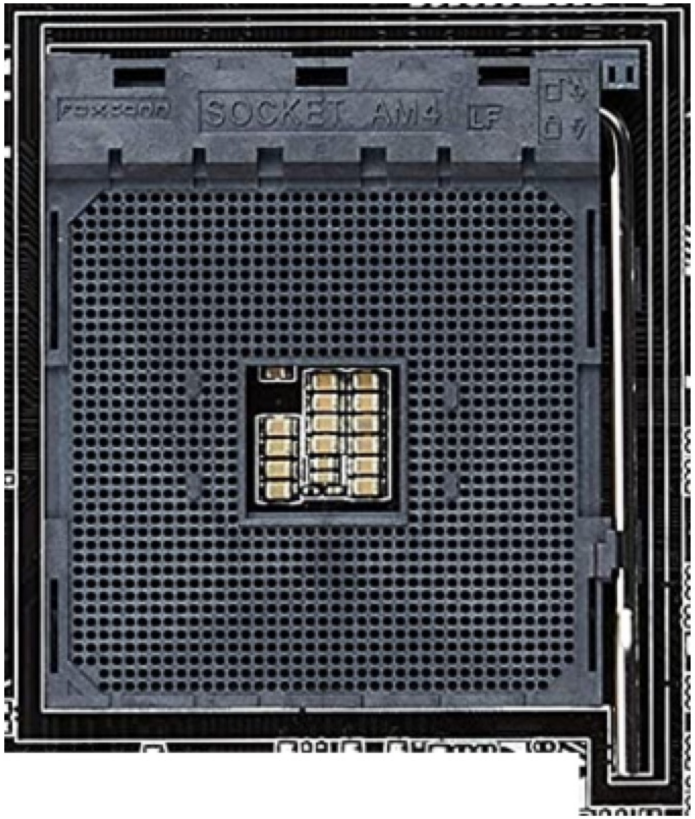
Intel Sockets



Anyway, here's a comparison between LGA 1700 (12th gen Intel) and LGA 1200 (10th/11th gen Intel):

AMD Sockets

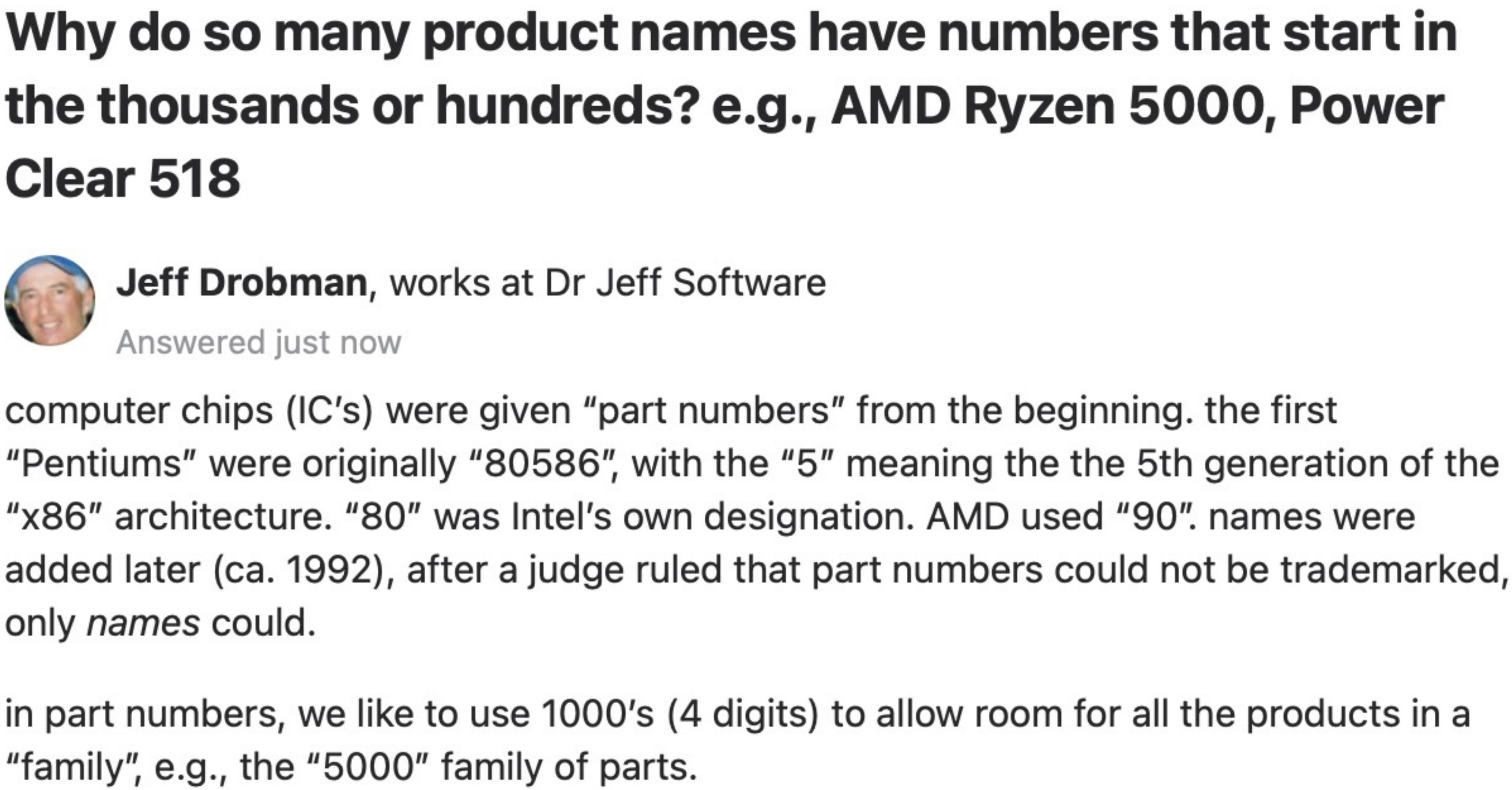
And here's the AM4 socket:



Heck, besides just not having all the pins in the same place, they even use a different *type* of socket — AMD puts all the pins on the processor and you line it up and shove them all down the holes on the motherboard (PGA-style socket) while Intel puts them all on the motherboard and the CPU has a bunch of flat contacts and you apply leveraged force to the CPU to hold it in place (LGA-style socket).

IC Part Numbers

Quora



Why do so many product names have numbers that start in the thousands or hundreds? e.g., AMD Ryzen 5000, Power Clear 518

Jeff Drobman, works at Dr Jeff Software

Answered just now

computer chips (IC’s) were given “part numbers” from the beginning. the first “Pentiums” were originally “80586”, with the “5” meaning the the 5th generation of the “x86” architecture. “80” was Intel’s own designation. AMD used “90”. names were added later (ca. 1992), after a judge ruled that part numbers could not be trademarked, only *names* could.

in part numbers, we like to use 1000’s (4 digits) to allow room for all the products in a “family”, e.g., the “5000” family of parts.

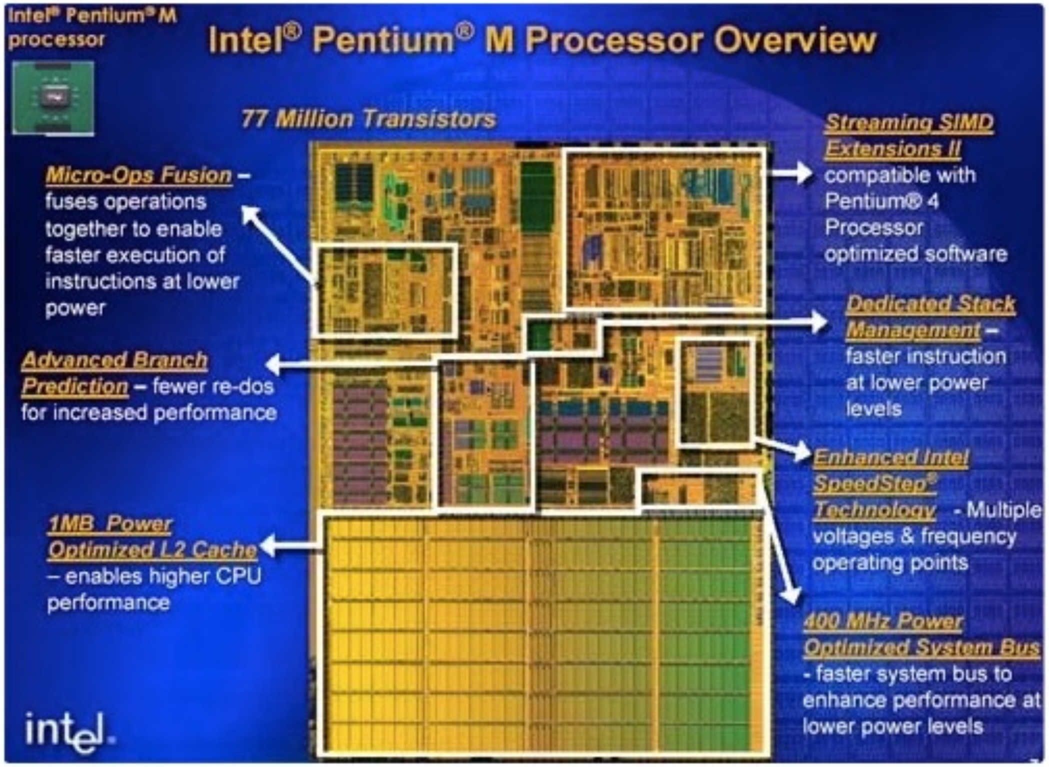
Modern CPU’s

Modern CPU’s

Intel

- See separate slide set “Intel”

Intel Pentium "M"



Intel® Pentium® M processor

Intel® Pentium® M Processor Overview

77 Million Transistors

Micro-Ops Fusion – fuses operations together to enable faster execution of instructions at lower power

Advanced Branch Prediction – fewer re-dos for increased performance

1MB Power Optimized L2 Cache – enables higher CPU performance

Streaming SIMD Extensions II compatible with Pentium® 4 Processor optimized software

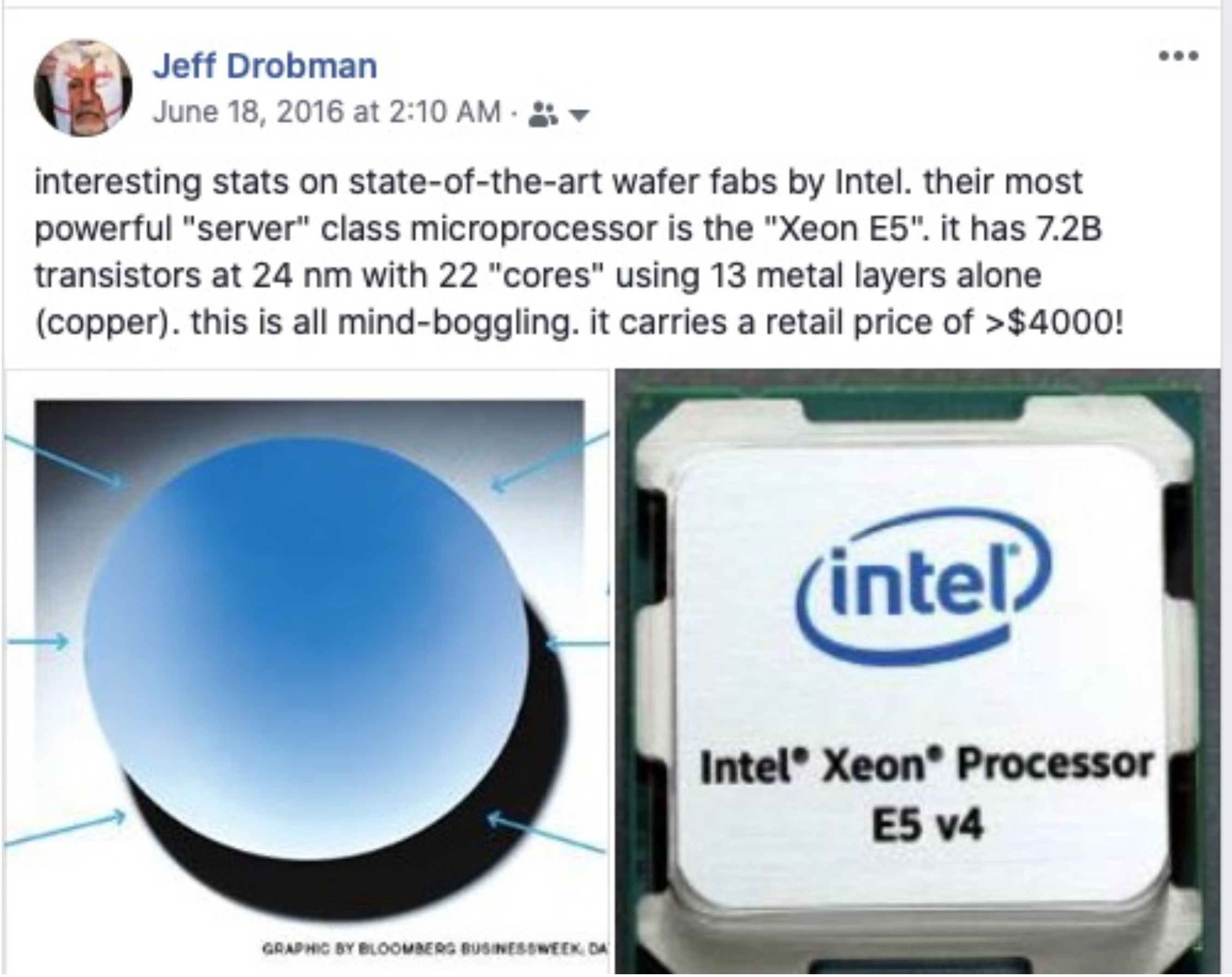
Dedicated Stack Management – faster instruction at lower power levels

Enhanced Intel SpeedStep® Technology - Multiple voltages & frequency operating points

400 MHz Power Optimized System Bus - faster system bus to enhance performance at lower power levels

The arrival of the Pentium M on the market immediately gave Intel back the performance lead in the mobile segment. A second revision, the "Dothan," increased this lead. The Pentium M was so superior to even desktop Pentium 4's that many gamers bought them and ran them on desktop motherboards released for exactly that purpose. Of course, it wasn't long before Intel started contemplating the release of a real desktop version of the Pentium M. When the next-generation Netburst design, called "Tejas," proved to a disaster (engineering samples used 150W at 2.8 GHz), Intel cancelled it and decided to make the Pentium M design the basis of all future Intel releases.

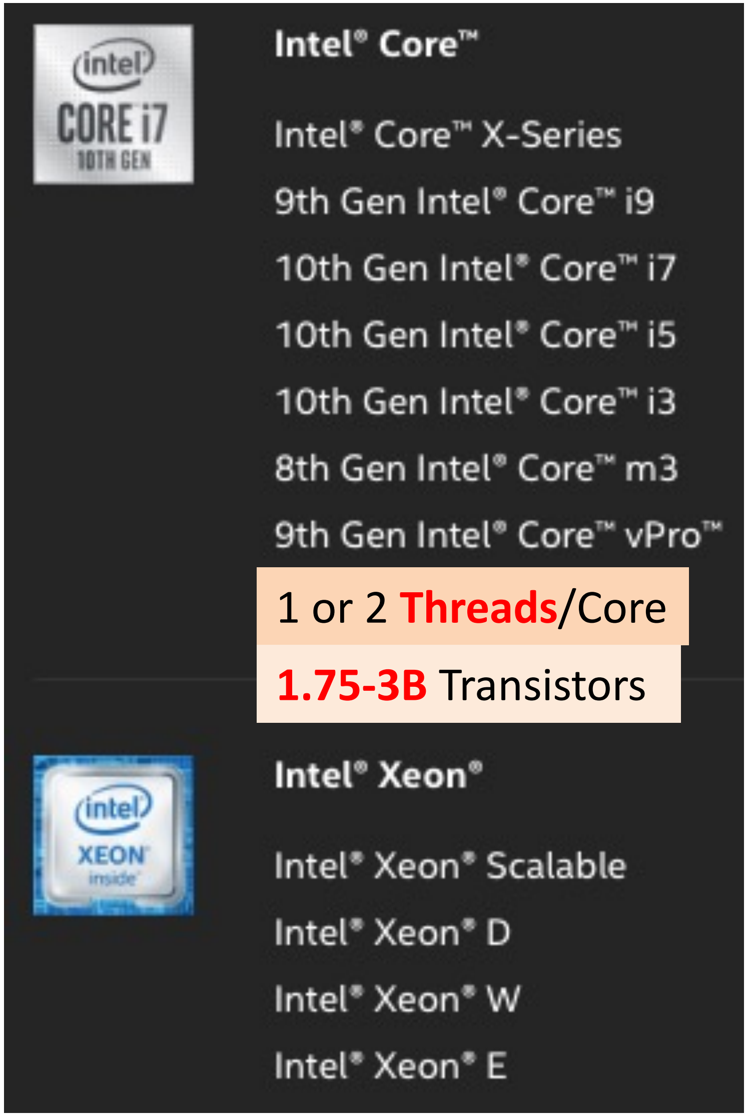
Intel Xeon E5



Jeff Drobman
June 18, 2016 at 2:10 AM ·

interesting stats on state-of-the-art wafer fabs by Intel. their most powerful "server" class microprocessor is the "Xeon E5". it has 7.2B transistors at 24 nm with 22 "cores" using 13 metal layers alone (copper). this is all mind-boggling. it carries a retail price of >$4000!

Intel Core i3/5/7/9 + Xeon



Intel® Core™

Intel® Core™ X-Series

9th Gen Intel® Core™ i9

10th Gen Intel® Core™ i7

10th Gen Intel® Core™ i5

10th Gen Intel® Core™ i3

8th Gen Intel® Core™ m3

9th Gen Intel® Core™ vPro™

1 or 2 **Threads**/Core

1.75-3B Transistors

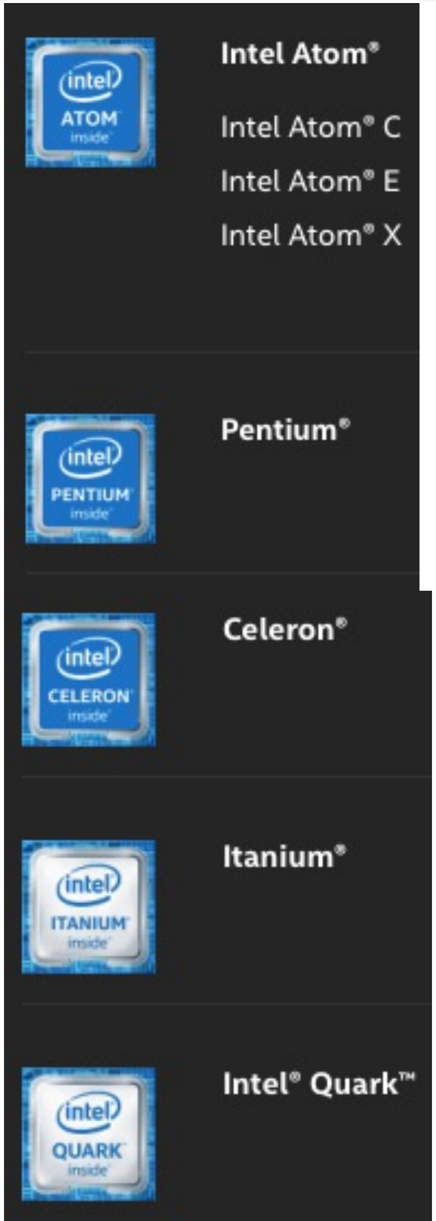
Intel® Xeon®

Intel® Xeon® Scalable

Intel® Xeon® D

Intel® Xeon® W

Intel® Xeon® E



Intel Atom®

Intel Atom® C

Intel Atom® E

Intel Atom® X

Pentium®

Celeron®

Itanium®

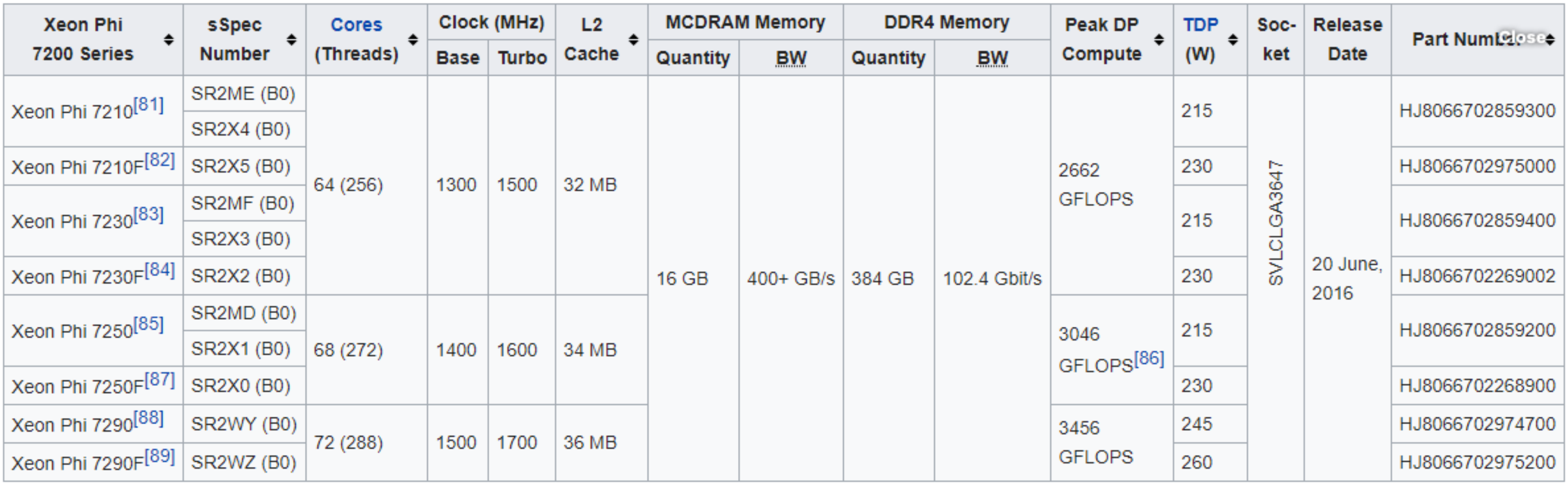
Intel® Quark™

Bottom of an LGA1151 CPU.

Top of an Intel Core i7-6700K (6th Gen).

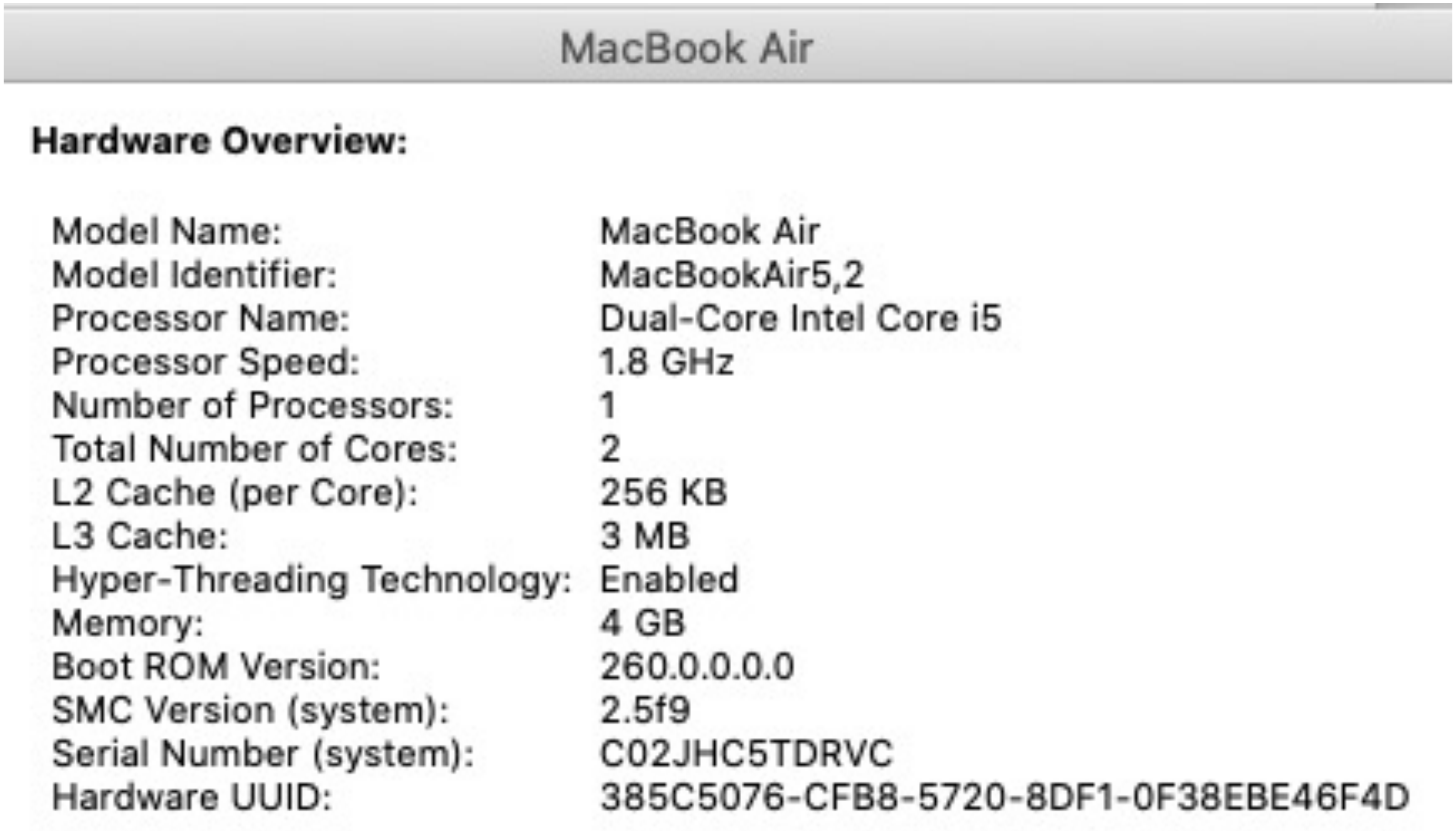
Intel Xeon

>60 Cores



Xeon Phi 7200 Series	sSpec Number	Cores (Threads)	Clock (MHz)		L2 Cache	MCDRAM Memory		DDR4 Memory		Peak DP Compute	TDP (W)	Soc-ket	Release Date	Part NumbClose
			Base	Turbo		Quantity	BW	Quantity	BW					
Xeon Phi 7210[81]	SR2ME (B0)	64 (256)	1300	1500	32 MB	16 GB	400+ GB/s	384 GB	102.4 Gbit/s	2662 GFLOPS	215	SVLCLGA3647	20 June, 2016	HJ8066702859300
	SR2X4 (B0)													
Xeon Phi 7210F[82]	SR2X5 (B0)										230			HJ8066702975000
Xeon Phi 7230[83]	SR2MF (B0)										215			HJ8066702859400
	SR2X3 (B0)													
Xeon Phi 7230F[84]	SR2X2 (B0)										230			HJ8066702269002
Xeon Phi 7250[85]	SR2MD (B0)	68 (272)	1400	1600	34 MB					3046 GFLOPS[86]	215			HJ8066702859200
	SR2X1 (B0)													
Xeon Phi 7250F[87]	SR2X0 (B0)										230			HJ8066702268900
Xeon Phi 7290[88]	SR2WY (B0)	72 (288)	1500	1700	36 MB					3456 GFLOPS	245			HJ8066702974700
Xeon Phi 7290F[89]	SR2WZ (B0)										260			HJ8066702975200

My MacBook



MacBook Air

Hardware Overview:

Model Name:	MacBook Air
Model Identifier:	MacBookAir5,2
Processor Name:	Dual-Core Intel Core i5
Processor Speed:	1.8 GHz
Number of Processors:	1
Total Number of Cores:	2
L2 Cache (per Core):	256 KB
L3 Cache:	3 MB
Hyper-Threading Technology:	Enabled
Memory:	4 GB
Boot ROM Version:	260.0.0.0.0
SMC Version (system):	2.5f9
Serial Number (system):	C02JHC5TDRVC
Hardware UUID:	385C5076-CFB8-5720-8DF1-0F38EBE46F4D

Modern CPU's

Modern CPU's

AMD

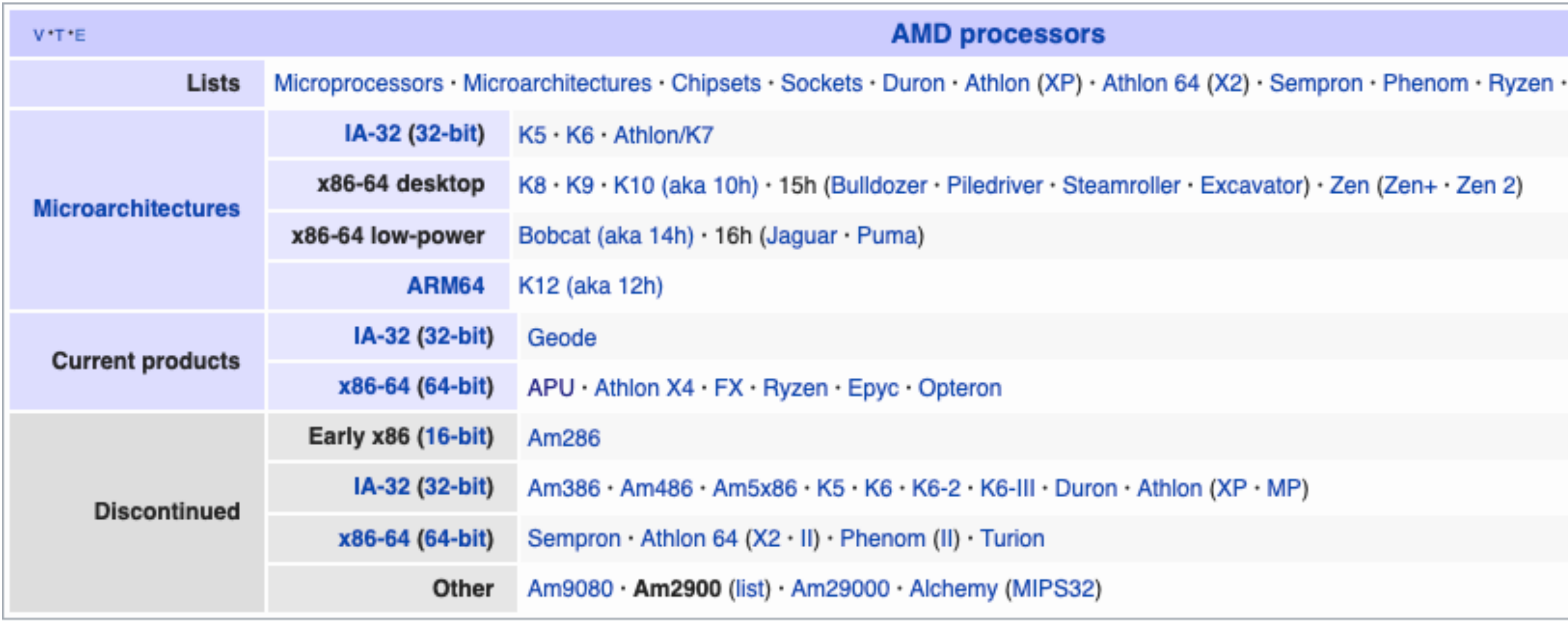
- See separate slide set "AMD"

CSUN
CALIFORNIA
STATE UNIVERSITY
NORTHRIDGE
COMP122

DR JEFF
SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021

AMD CPU’s

AMD x86



V·T·E	AMD processors	
Lists	Microprocessors · Microarchitectures · Chipsets · Sockets · Duron · Athlon (XP) · Athlon 64 (X2) · Sempron · Phenom · Ryzen ·	
Microarchitectures	IA-32 (32-bit)	K5 · K6 · Athlon/K7
	x86-64 desktop	K8 · K9 · K10 (aka 10h) · 15h (Bulldozer · Piledriver · Steamroller · Excavator) · Zen (Zen+ · Zen 2)
	x86-64 low-power	Bobcat (aka 14h) · 16h (Jaguar · Puma)
	ARM64	K12 (aka 12h)
Current products	IA-32 (32-bit)	Geode
	x86-64 (64-bit)	APU · Athlon X4 · FX · Ryzen · Epyc · Opteron
Discontinued	Early x86 (16-bit)	Am286
	IA-32 (32-bit)	Am386 · Am486 · Am5x86 · K5 · K6 · K6-2 · K6-III · Duron · Athlon (XP · MP)
	x86-64 (64-bit)	Sempron · Athlon 64 (X2 · II) · Phenom (II) · Turion
	Other	Am9080 · **Am2900** (list) · Am29000 · Alchemy (MIPS32)

AMD’s Zen

AMD News

AMD's new Zen Processor

AMD Zen

stock news usa (2016)

(Click image to view full size)

AMD made radical alterations to its Zen design while keeping itself distant from an ugly past. The company knew it had to make the changes to become a force to reckon with in the server and PC markets. So when the designers of the chip sat down to map the Zen design, they had two priorities: To boost CPU performance to maximum and to stabilize power efficiency.

According to a company spokesperson, the chips will come with 8 to 32 cores. The 32-core chips may come in the quad-CPU configurations although those details haven’t been finalized yet.

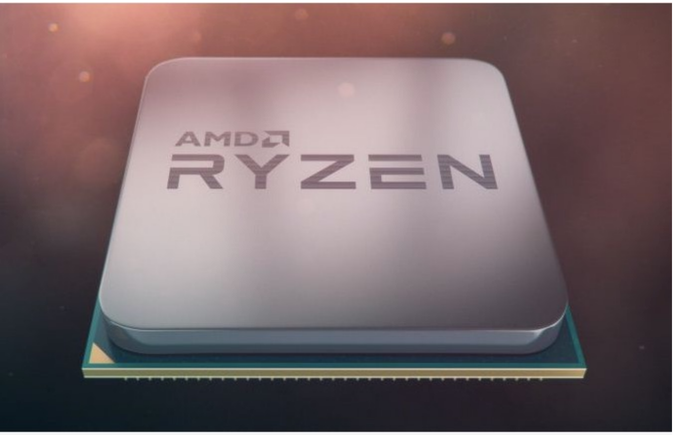
5-19B Transistors

- ❖ CPU performance
- ❖ Power efficiency

8-32 cores

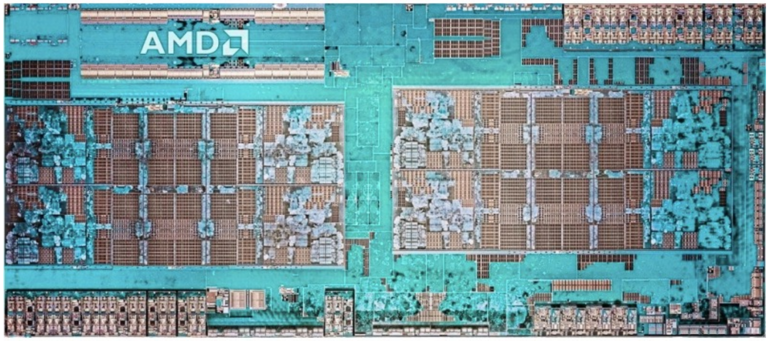
Source: stocknewsusa (2016-08-26)

Inside AMD’s Production Of The Zen CPU



Ryzen 7 will have three CPUs to start, all having eight cores and supporting simultaneous multi-threading:

- *Ryzen 7 1800X: 8C/16T, 3.6 GHz base, 4.0 GHz turbo, 95W, $499*
- *Ryzen 7 1700X: 8C/16T, 3.4 GHz base, 3.8 GHz turbo, 95W, $399*
- *Ryzen 7 1700: 8C/16T, 3.0 GHz base, 3.7 GHz turbo, $329*



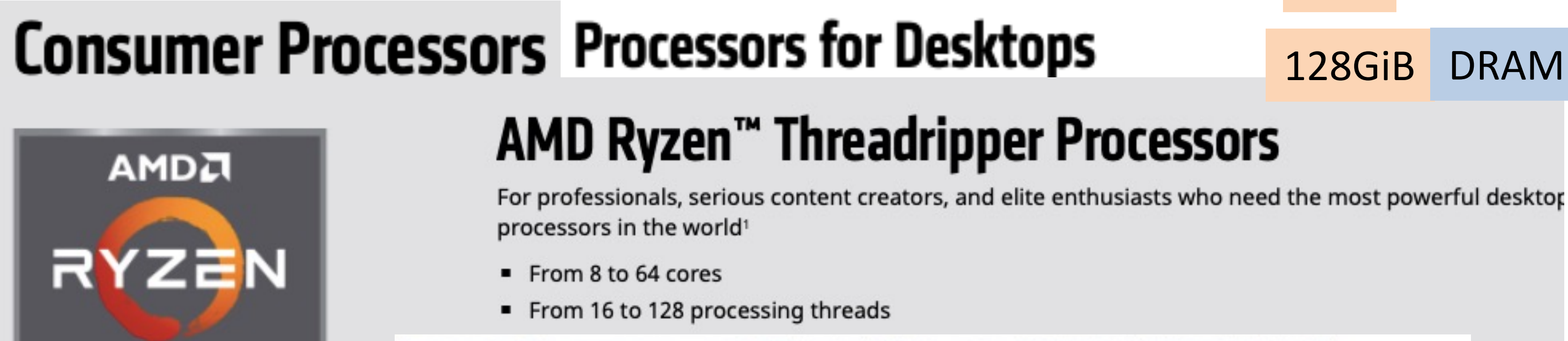
Ryzen

4.8 billion transistors and more than 2,000m of signal wire

AMD

CPU

128GiB DRAM



Consumer Processors | Processors for Desktops

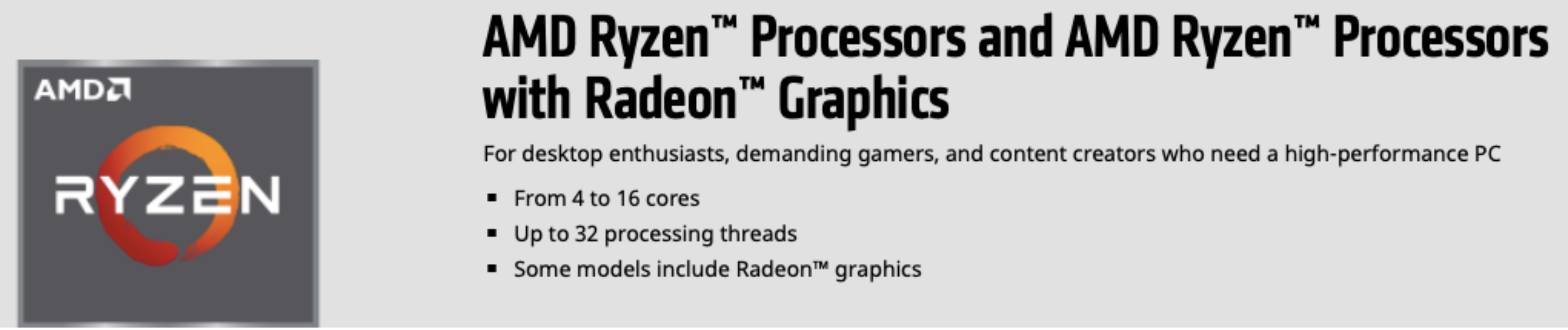
AMD Ryzen™ Threadripper Processors

For professionals, serious content creators, and elite enthusiasts who need the most powerful desktop processors in the world[1]

- From 8 to 64 cores
- From 16 to 128 processing threads

The big old Threadripper in my desktop PC only supports 128GiB DRAM. But processors built for servers, Intel® Xeon® and AMD EPYC, will support 1TB+ DRAM. The EPYC 7742 will support 4TiB DRAM, and the price of the DRAM won't seem so shocking when you learn the price of your 64-core CPU is about $7,500!

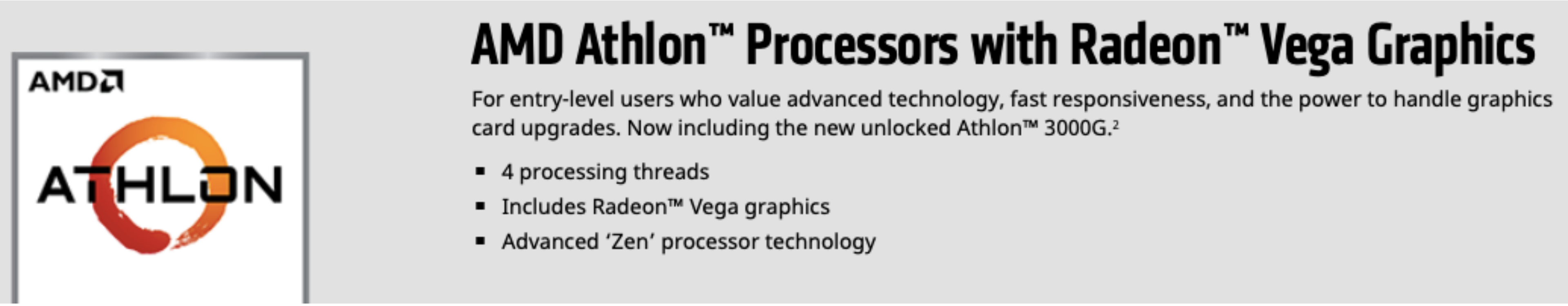
AMD



AMD Ryzen™ Processors and AMD Ryzen™ Processors with Radeon™ Graphics

For desktop enthusiasts, demanding gamers, and content creators who need a high-performance PC

- From 4 to 16 cores
- Up to 32 processing threads
- Some models include Radeon™ graphics



AMD Athlon™ Processors with Radeon™ Vega Graphics

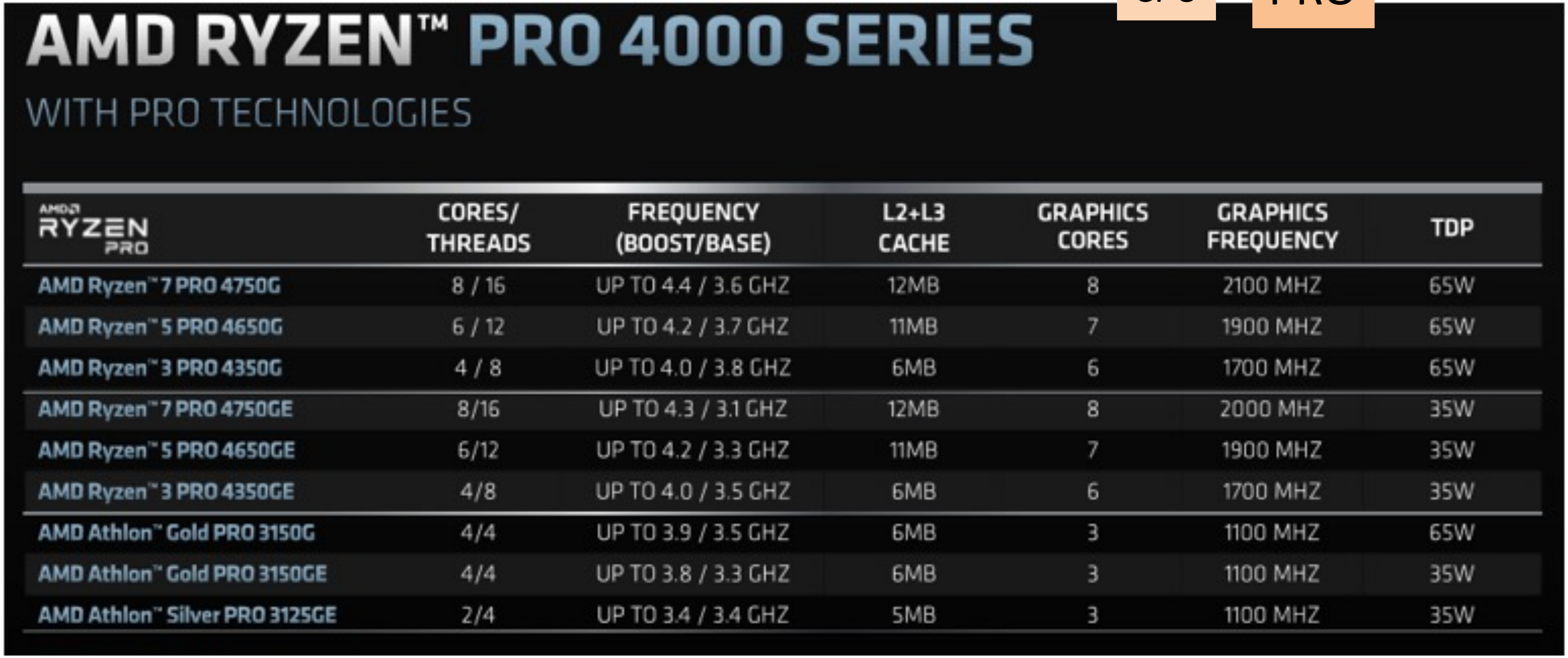
For entry-level users who value advanced technology, fast responsiveness, and the power to handle graphics card upgrades. Now including the new unlocked Athlon™ 3000G.[2]

- 4 processing threads
- Includes Radeon™ Vega graphics
- Advanced 'Zen' processor technology

AMD

CPU

PRO



AMD RYZEN™ PRO 4000 SERIES

WITH PRO TECHNOLOGIES

AMD RYZEN PRO	CORES/ THREADS	FREQUENCY (BOOST/BASE)	L2+L3 CACHE	GRAPHICS CORES	GRAPHICS FREQUENCY	TDP
AMD Ryzen™ 7 PRO 4750G	8 / 16	UP TO 4.4 / 3.6 GHZ	12MB	8	2100 MHZ	65W
AMD Ryzen™ 5 PRO 4650G	6 / 12	UP TO 4.2 / 3.7 GHZ	11MB	7	1900 MHZ	65W
AMD Ryzen™ 3 PRO 4350G	4 / 8	UP TO 4.0 / 3.8 GHZ	6MB	6	1700 MHZ	65W
AMD Ryzen™ 7 PRO 4750GE	8/16	UP TO 4.3 / 3.1 GHZ	12MB	8	2000 MHZ	35W
AMD Ryzen™ 5 PRO 4650GE	6/12	UP TO 4.2 / 3.3 GHZ	11MB	7	1900 MHZ	35W
AMD Ryzen™ 3 PRO 4350GE	4/8	UP TO 4.0 / 3.5 GHZ	6MB	6	1700 MHZ	35W
AMD Athlon™ Gold PRO 3150G	4/4	UP TO 3.9 / 3.5 GHZ	6MB	3	1100 MHZ	65W
AMD Athlon™ Gold PRO 3150GE	4/4	UP TO 3.8 / 3.3 GHZ	6MB	3	1100 MHZ	35W
AMD Athlon™ Silver PRO 3125GE	2/4	UP TO 3.4 / 3.4 GHZ	5MB	3	1100 MHZ	35W

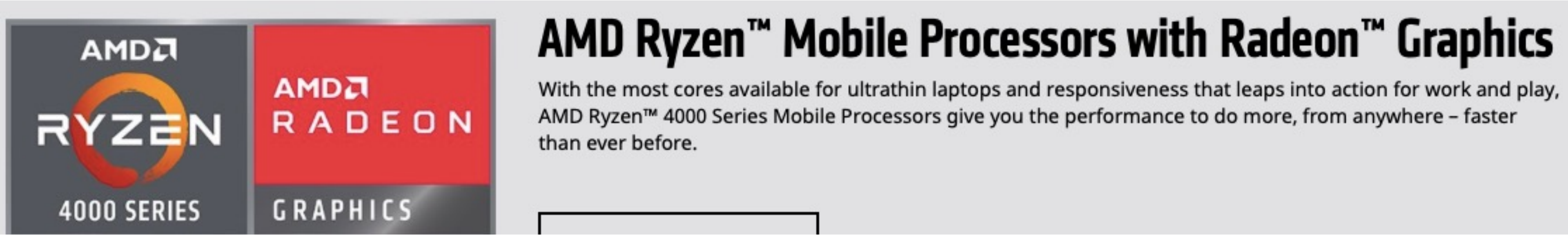
AMD Athlon™ PRO Processors with Radeon™ Vega Graphics

For entry-level users in the workplace

AMD

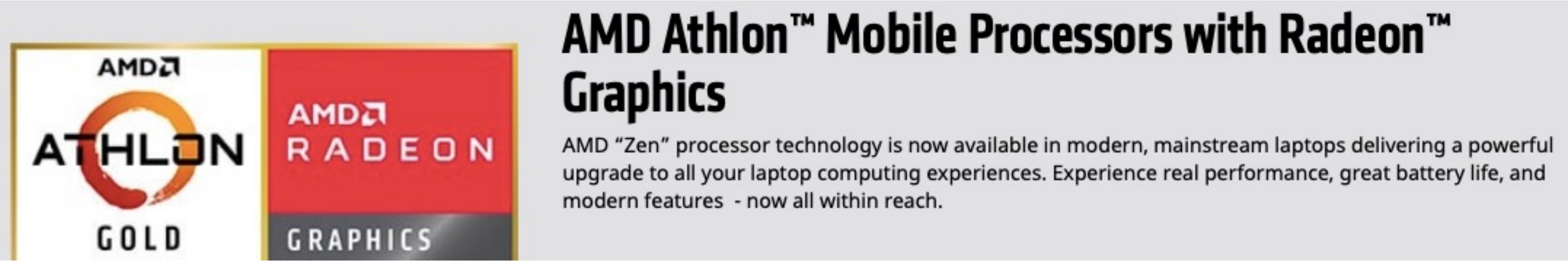
CPU

Mobile Processors



AMD Ryzen™ Mobile Processors with Radeon™ Graphics

With the most cores available for ultrathin laptops and responsiveness that leaps into action for work and play, AMD Ryzen™ 4000 Series Mobile Processors give you the performance to do more, from anywhere – faster than ever before.



AMD Athlon™ Mobile Processors with Radeon™ Graphics

AMD “Zen” processor technology is now available in modern, mainstream laptops delivering a powerful upgrade to all your laptop computing experiences. Experience real performance, great battery life, and modern features - now all within reach.

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

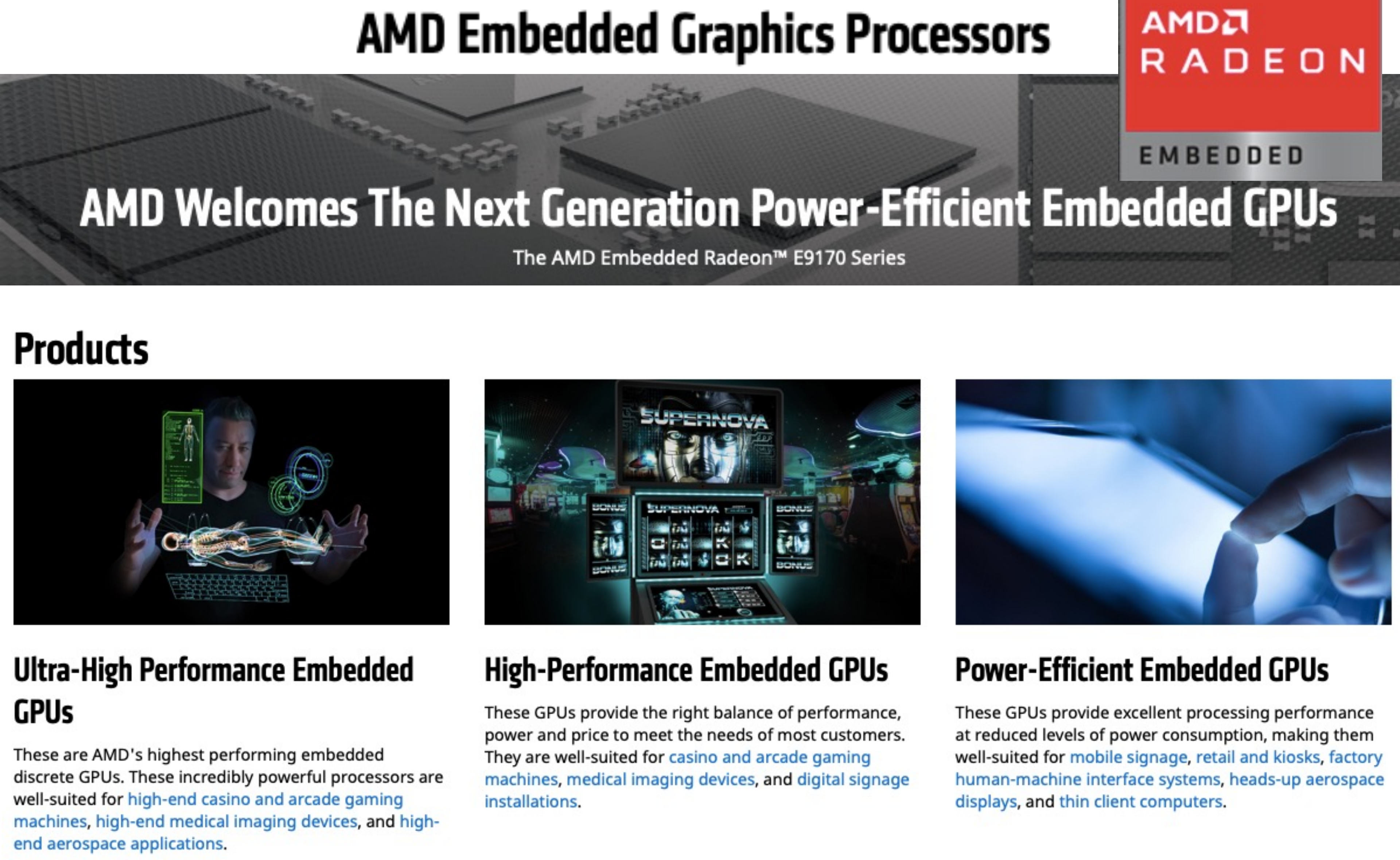
COMP122

AMD

DR JEFF SOFTWARE
INDIE APP DEVELOPER

GPU

AMD RADEON EMBEDDED



AMD Embedded Graphics Processors

AMD Welcomes The Next Generation Power-Efficient Embedded GPUs

The AMD Embedded Radeon™ E9170 Series

Products

Ultra-High Performance Embedded GPUs

These are AMD's highest performing embedded discrete GPUs. These incredibly powerful processors are well-suited for high-end casino and arcade gaming machines, high-end medical imaging devices, and high-end aerospace applications.

High-Performance Embedded GPUs

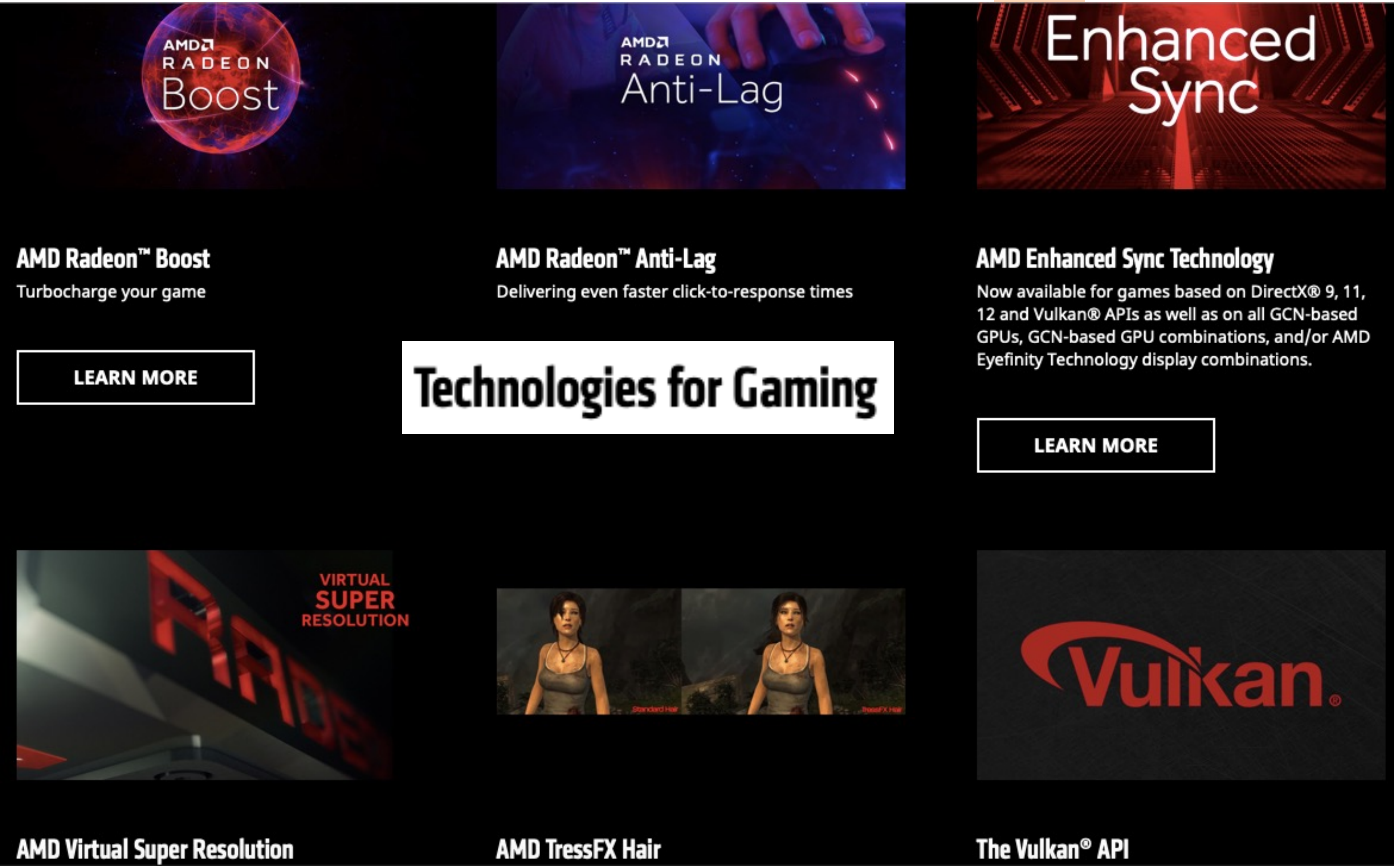
These GPUs provide the right balance of performance, power and price to meet the needs of most customers. They are well-suited for casino and arcade gaming machines, medical imaging devices, and digital signage installations.

Power-Efficient Embedded GPUs

These GPUs provide excellent processing performance at reduced levels of power consumption, making them well-suited for mobile signage, retail and kiosks, factory human-machine interface systems, heads-up aerospace displays, and thin client computers.

AMD

GPU



Technologies for Gaming

AMD Radeon™ Boost

Turbocharge your game

AMD Radeon™ Anti-Lag

Delivering even faster click-to-response times

AMD Enhanced Sync Technology

Now available for games based on DirectX® 9, 11, 12 and Vulkan® APIs as well as on all GCN-based GPUs, GCN-based GPU combinations, and/or AMD Eyefinity Technology display combinations.

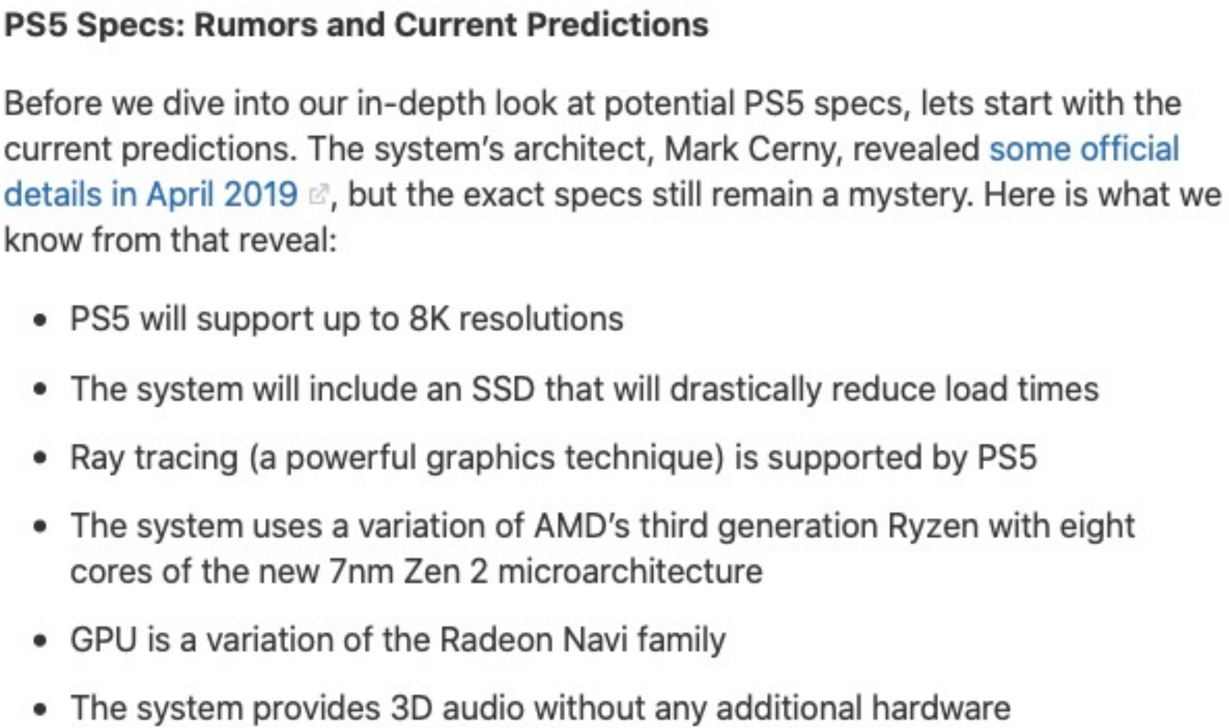
AMD Virtual Super Resolution

AMD TressFX Hair

The Vulkan® API

AMD CPU's in PS5

7nm Ryzen



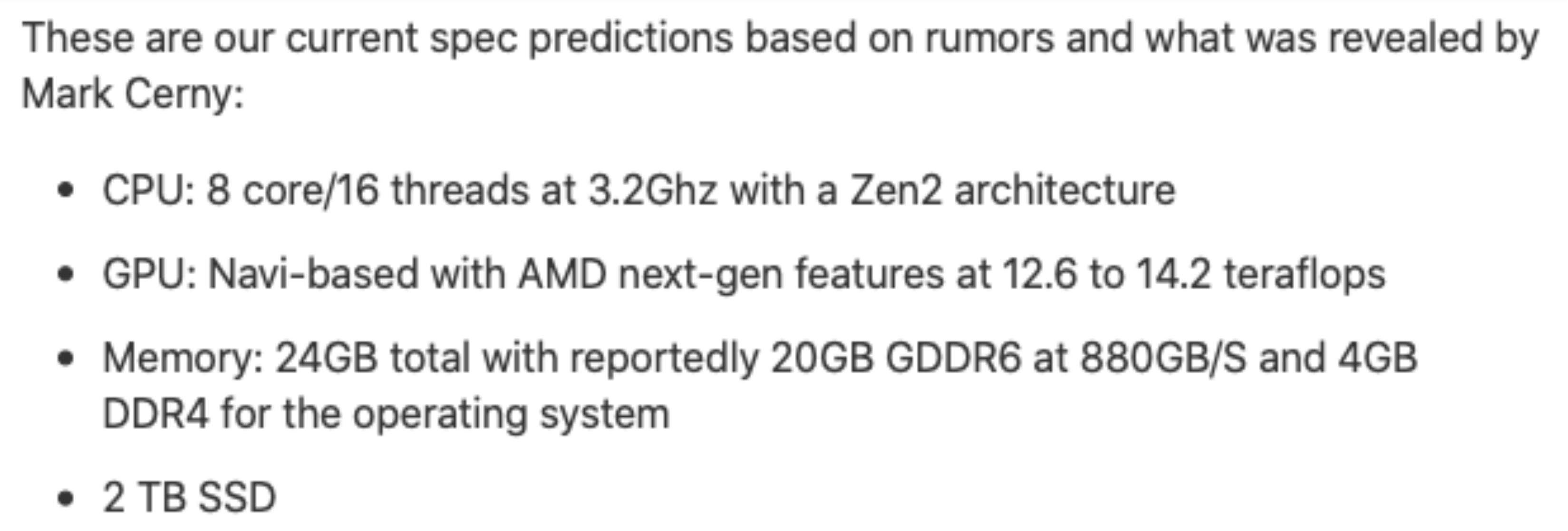
PS5 Specs: Rumors and Current Predictions

Before we dive into our in-depth look at potential PS5 specs, lets start with the current predictions. The system's architect, Mark Cerny, revealed some official details in April 2019, but the exact specs still remain a mystery. Here is what we know from that reveal:

- PS5 will support up to 8K resolutions
- The system will include an SSD that will drastically reduce load times
- Ray tracing (a powerful graphics technique) is supported by PS5
- The system uses a variation of AMD's third generation Ryzen with eight cores of the new 7nm Zen 2 microarchitecture
- GPU is a variation of the Radeon Navi family
- The system provides 3D audio without any additional hardware

AMD CPU’s in PS5

7nm Ryzen



These are our current spec predictions based on rumors and what was revealed by Mark Cerny:

- CPU: 8 core/16 threads at 3.2Ghz with a Zen2 architecture
- GPU: Navi-based with AMD next-gen features at 12.6 to 14.2 teraflops
- Memory: 24GB total with reportedly 20GB GDDR6 at 880GB/S and 4GB DDR4 for the operating system
- 2 TB SSD

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP122

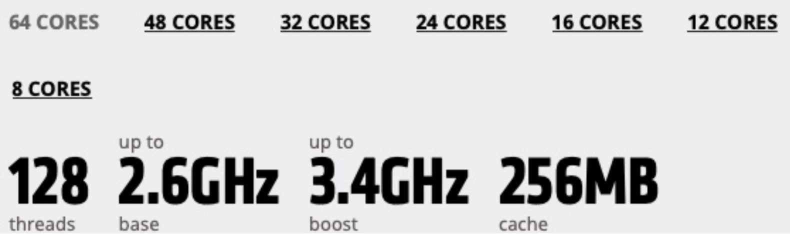
AMD EPYC

AMD

Servers

CPU

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021

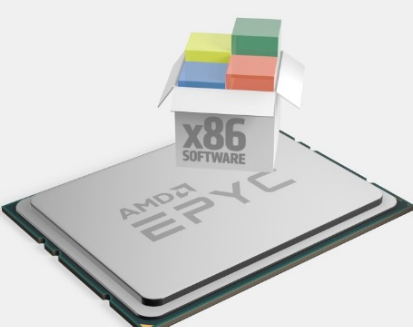


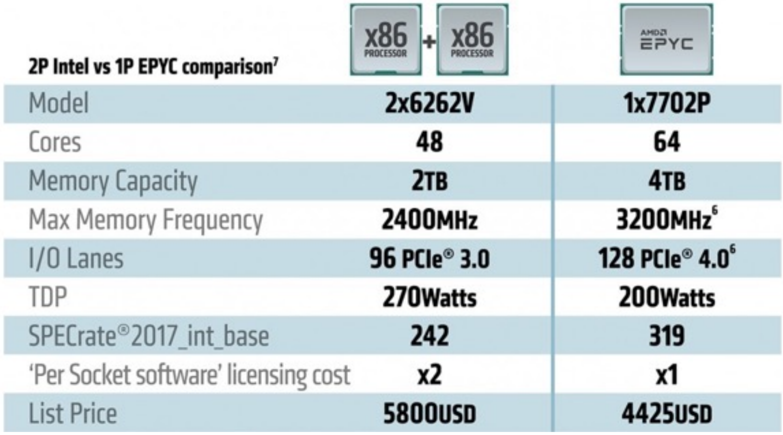
64 CORES   48 CORES   32 CORES   24 CORES   16 CORES   12 CORES

8 CORES

	up to	up to	
128	**2.6GHz**	**3.4GHz**	**256MB**
threads	base	boost	cache

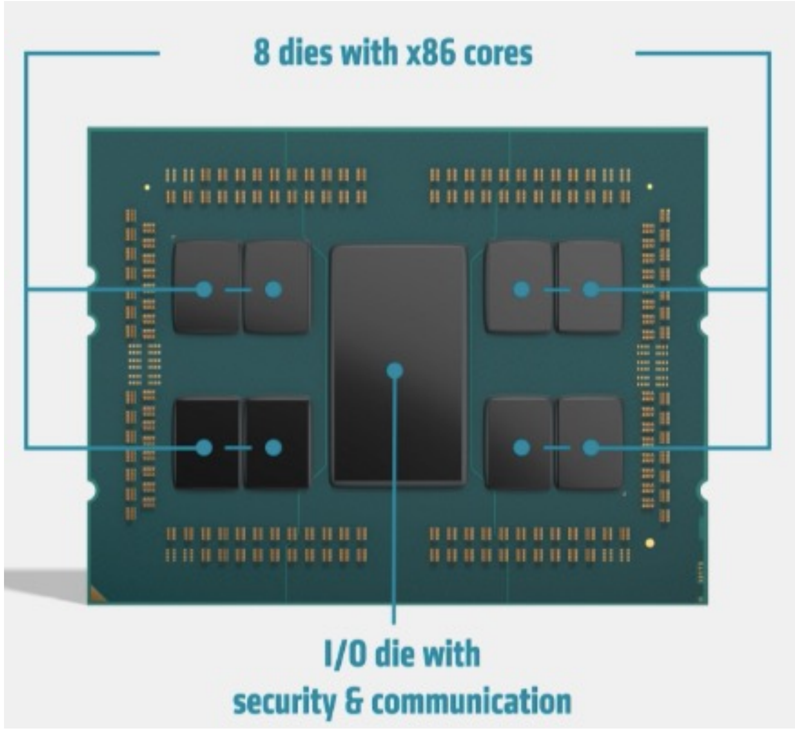
X86 compatibility





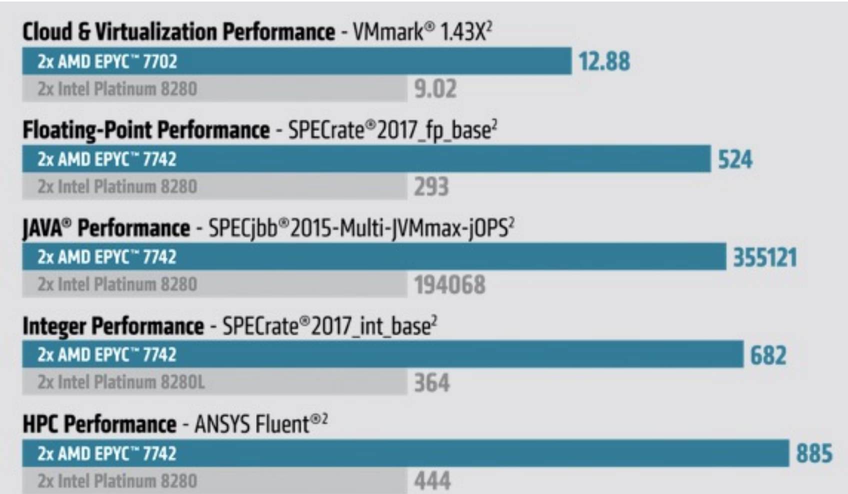
2P Intel vs 1P EPYC comparison[7]

	x86 PROCESSOR + x86 PROCESSOR	AMD EPYC
Model	2x6262V	1x7702P
Cores	48	64
Memory Capacity	2TB	4TB
Max Memory Frequency	2400MHz	3200MHz[6]
I/O Lanes	96 PCIe® 3.0	128 PCIe® 4.0[6]
TDP	270Watts	200Watts
SPECrate®2017_int_base	242	319
'Per Socket software' licensing cost	x2	x1
List Price	5800USD	4425USD



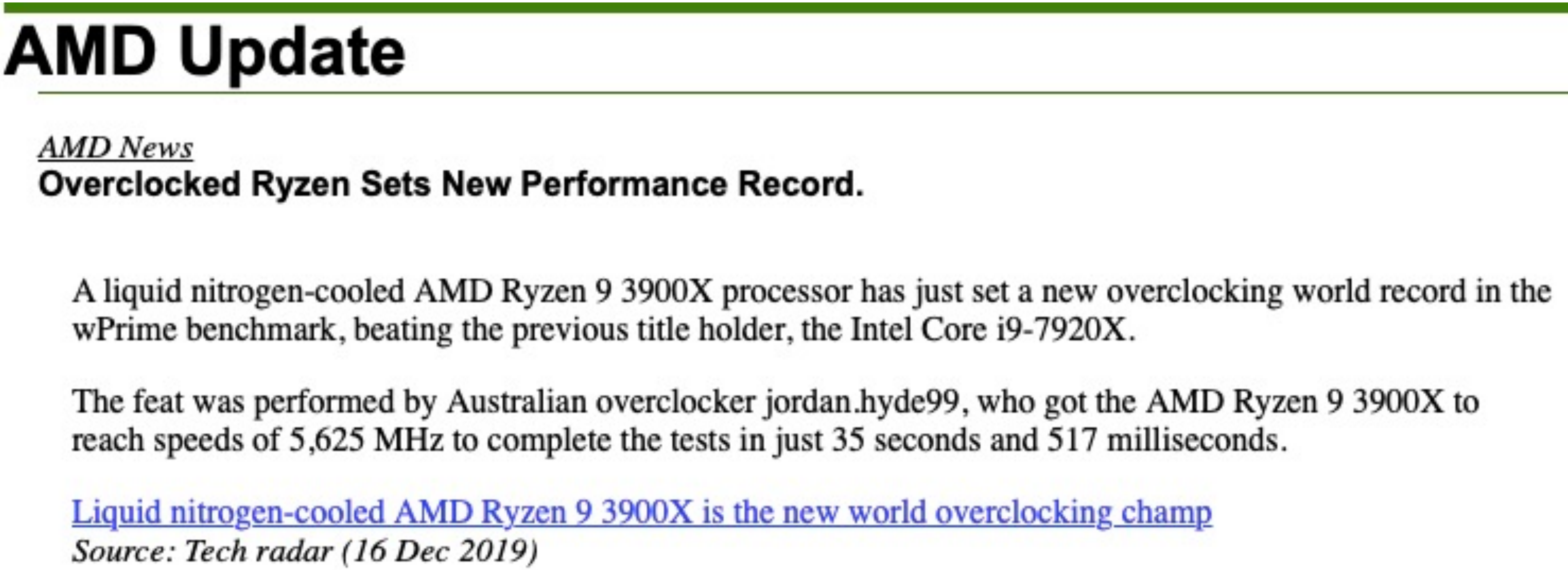
8 dies with x86 cores

I/O die with
security & communication



AMD Over-clocked

AMD Ryzen 9 vs. Intel i9



AMD Update

AMD News

Overclocked Ryzen Sets New Performance Record.

A liquid nitrogen-cooled AMD Ryzen 9 3900X processor has just set a new overclocking world record in the wPrime benchmark, beating the previous title holder, the Intel Core i9-7920X.

The feat was performed by Australian overclocker jordan.hyde99, who got the AMD Ryzen 9 3900X to reach speeds of 5,625 MHz to complete the tests in just 35 seconds and 517 milliseconds.

Liquid nitrogen-cooled AMD Ryzen 9 3900X is the new world overclocking champ

Source: Tech radar (16 Dec 2019)

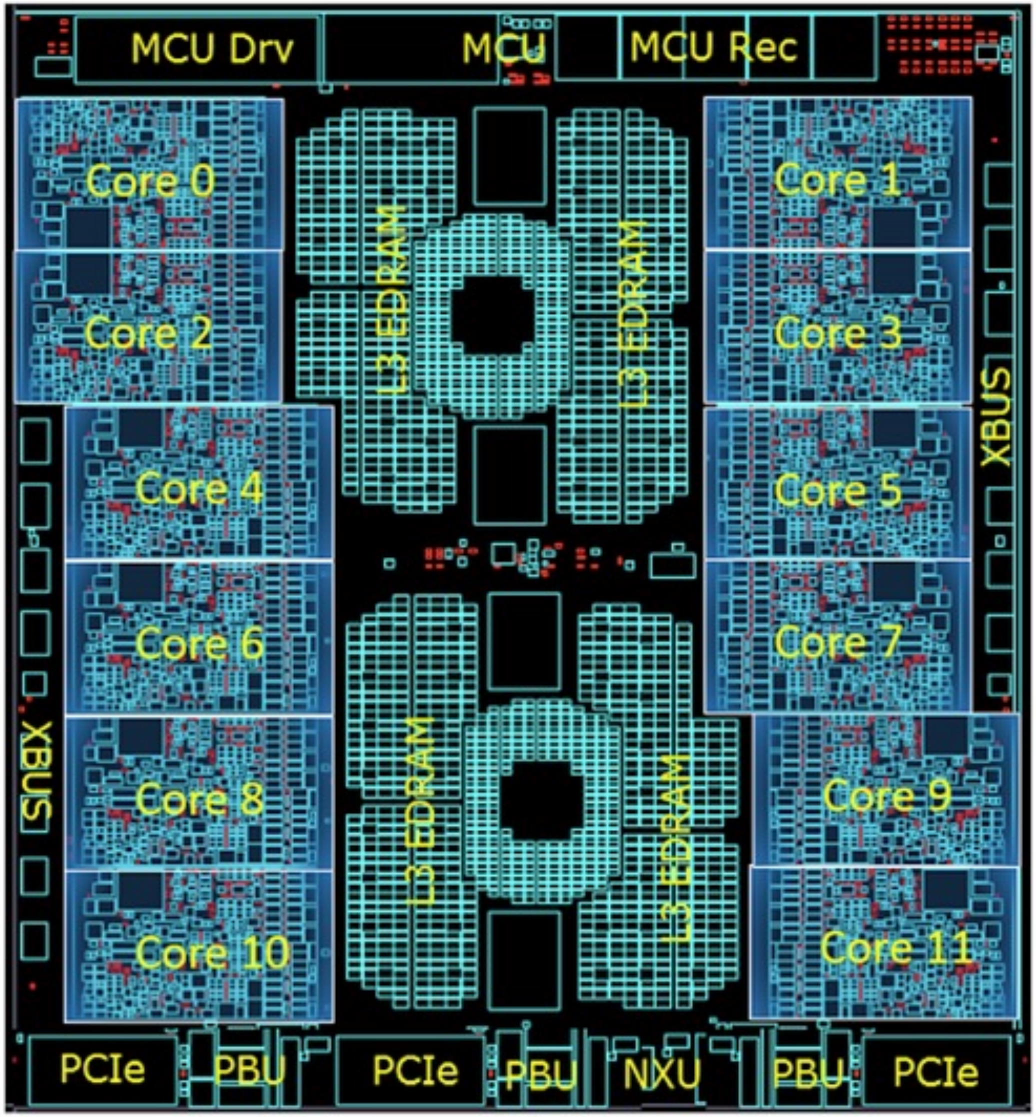
Modern CPU's

HPC

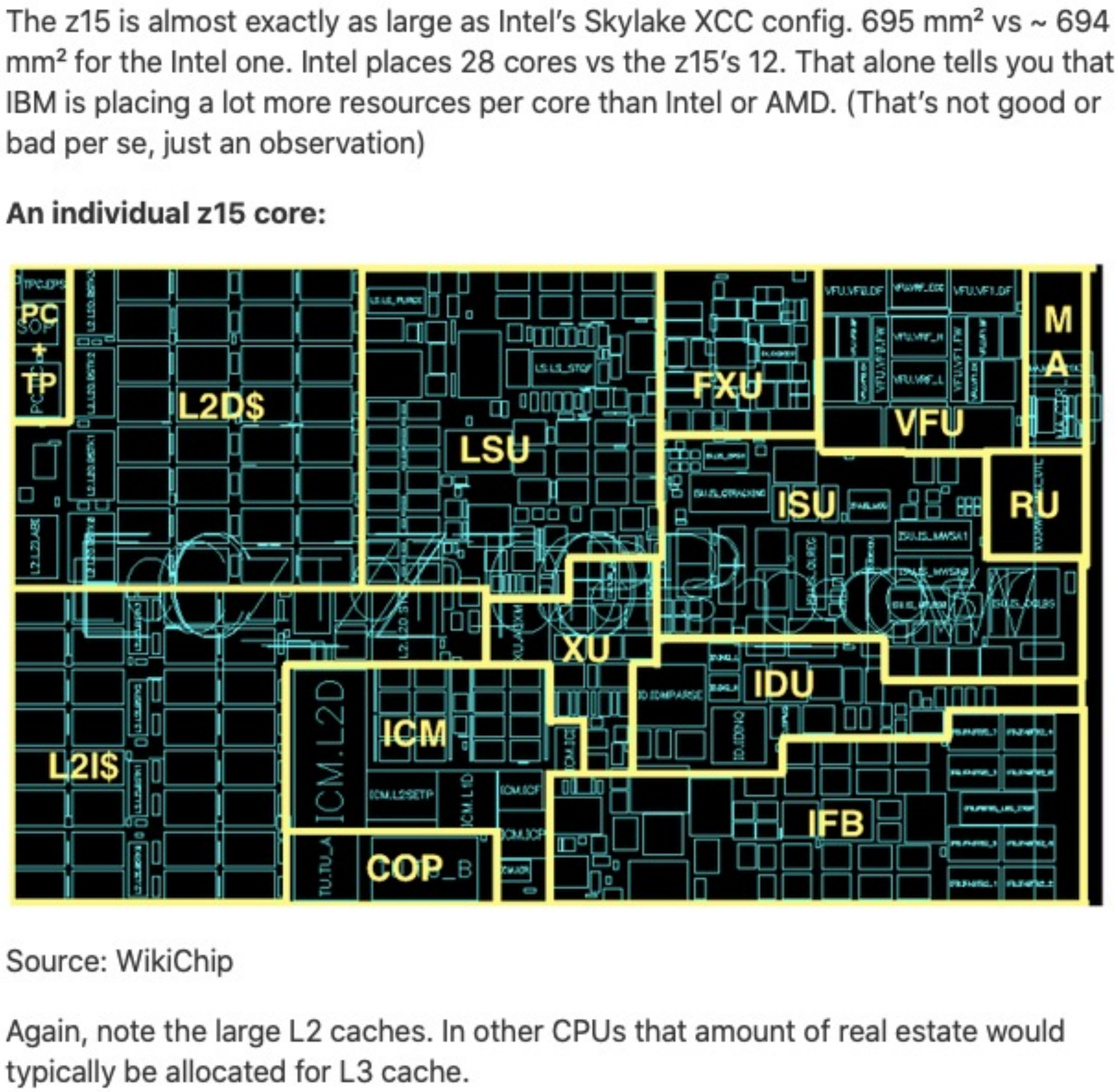
- IBM z15
- Fujitsu Sparc

IBM CPU's: z15

z15 floorplan:



IBM CPU’s: z15



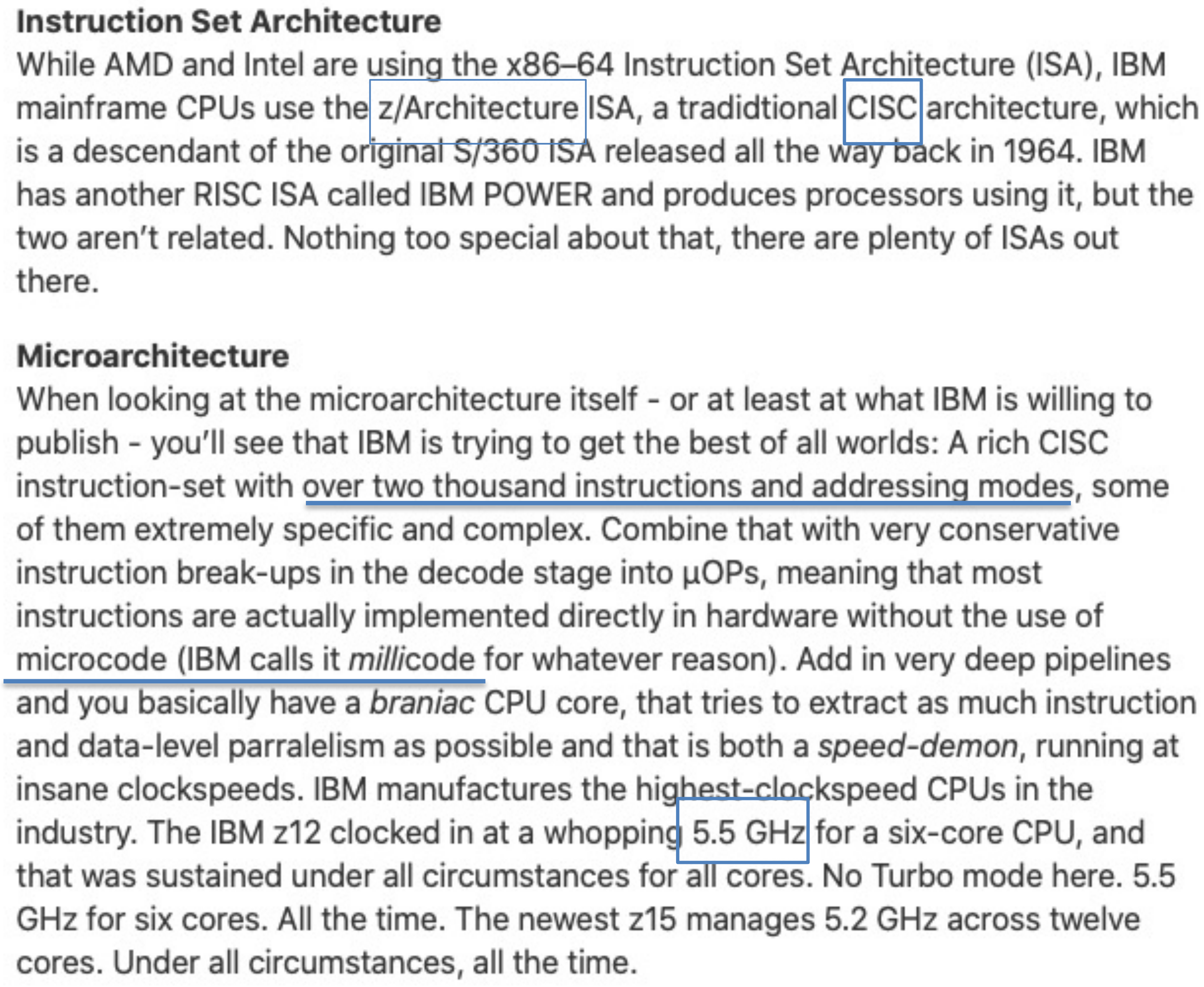
The z15 is almost exactly as large as Intel’s Skylake XCC config. 695 mm² vs ~ 694 mm² for the Intel one. Intel places 28 cores vs the z15’s 12. That alone tells you that IBM is placing a lot more resources per core than Intel or AMD. (That’s not good or bad per se, just an observation)

An individual z15 core:

Source: WikiChip

Again, note the large L2 caches. In other CPUs that amount of real estate would typically be allocated for L3 cache.

IBM CPU’s: ISA



Instruction Set Architecture

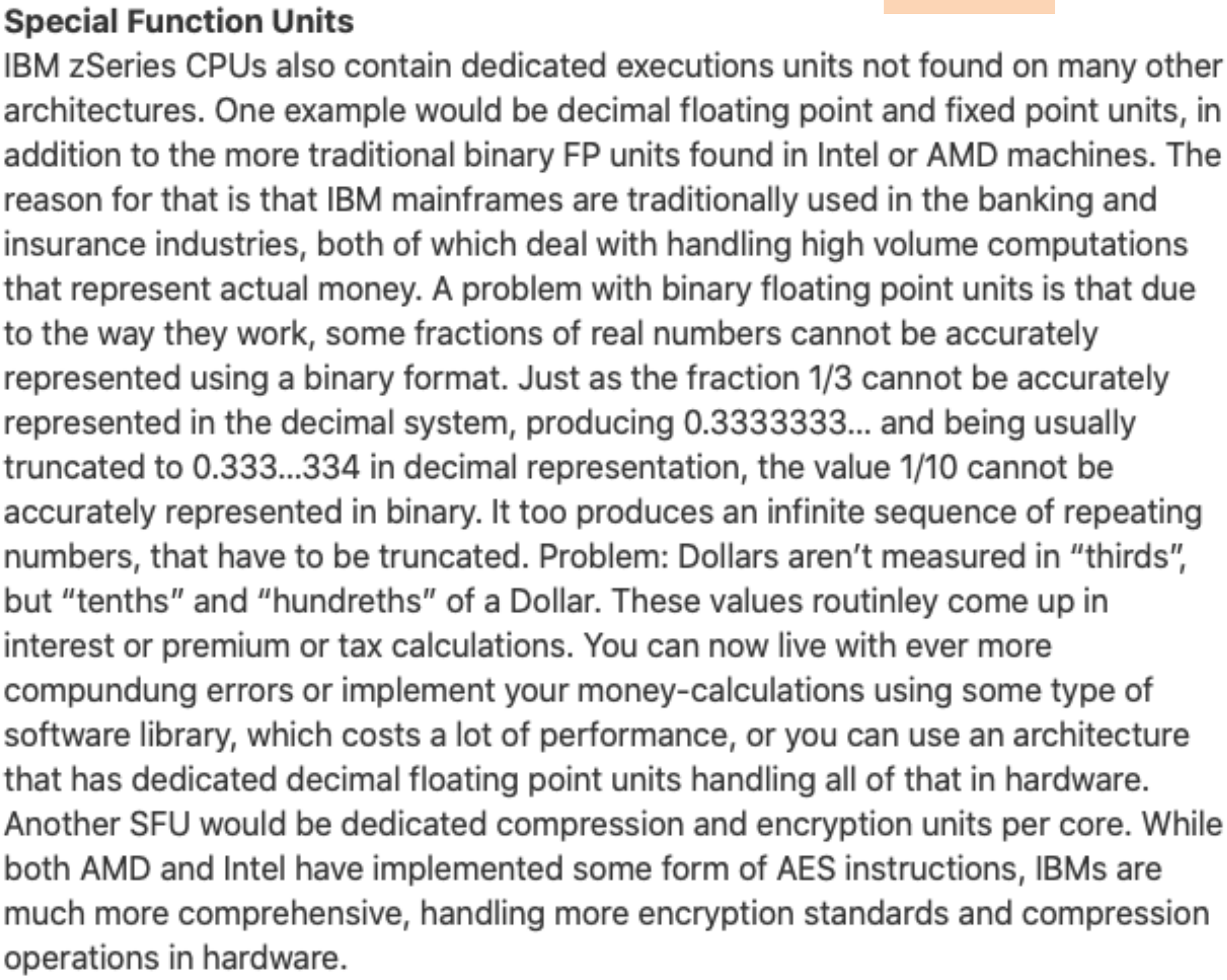
While AMD and Intel are using the x86–64 Instruction Set Architecture (ISA), IBM mainframe CPUs use the z/Architecture ISA, a tradidtional CISC architecture, which is a descendant of the original S/360 ISA released all the way back in 1964. IBM has another RISC ISA called IBM POWER and produces processors using it, but the two aren’t related. Nothing too special about that, there are plenty of ISAs out there.

Microarchitecture

When looking at the microarchitecture itself - or at least at what IBM is willing to publish - you’ll see that IBM is trying to get the best of all worlds: A rich CISC instruction-set with over two thousand instructions and addressing modes, some of them extremely specific and complex. Combine that with very conservative instruction break-ups in the decode stage into µOPs, meaning that most instructions are actually implemented directly in hardware without the use of microcode (IBM calls it *milli*code for whatever reason). Add in very deep pipelines and you basically have a *braniac* CPU core, that tries to extract as much instruction and data-level parralelism as possible and that is both a *speed-demon*, running at insane clockspeeds. IBM manufactures the highest-clockspeed CPUs in the industry. The IBM z12 clocked in at a whopping 5.5 GHz for a six-core CPU, and that was sustained under all circumstances for all cores. No Turbo mode here. 5.5 GHz for six cores. All the time. The newest z15 manages 5.2 GHz across twelve cores. Under all circumstances, all the time.

IBM CPU’s: SFU’s

Decimal

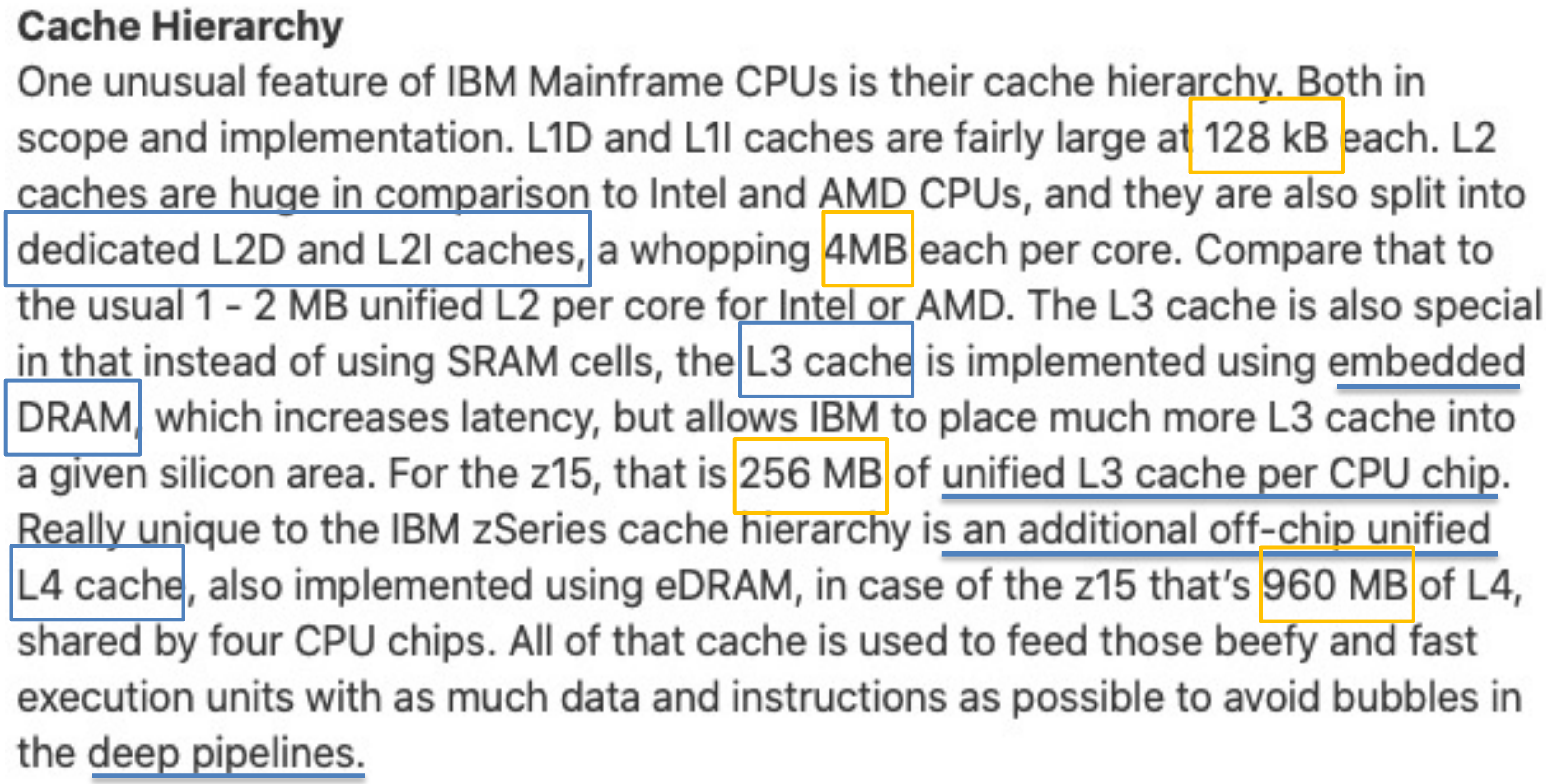


Special Function Units

IBM zSeries CPUs also contain dedicated executions units not found on many other architectures. One example would be decimal floating point and fixed point units, in addition to the more traditional binary FP units found in Intel or AMD machines. The reason for that is that IBM mainframes are traditionally used in the banking and insurance industries, both of which deal with handling high volume computations that represent actual money. A problem with binary floating point units is that due to the way they work, some fractions of real numbers cannot be accurately represented using a binary format. Just as the fraction 1/3 cannot be accurately represented in the decimal system, producing 0.3333333… and being usually truncated to 0.333…334 in decimal representation, the value 1/10 cannot be accurately represented in binary. It too produces an infinite sequence of repeating numbers, that have to be truncated. Problem: Dollars aren’t measured in “thirds”, but “tenths” and “hundreths” of a Dollar. These values routinley come up in interest or premium or tax calculations. You can now live with ever more compundung errors or implement your money-calculations using some type of software library, which costs a lot of performance, or you can use an architecture that has dedicated decimal floating point units handling all of that in hardware. Another SFU would be dedicated compression and encryption units per core. While both AMD and Intel have implemented some form of AES instructions, IBMs are much more comprehensive, handling more encryption standards and compression operations in hardware.

IBM CPU’s: Cache

MLM



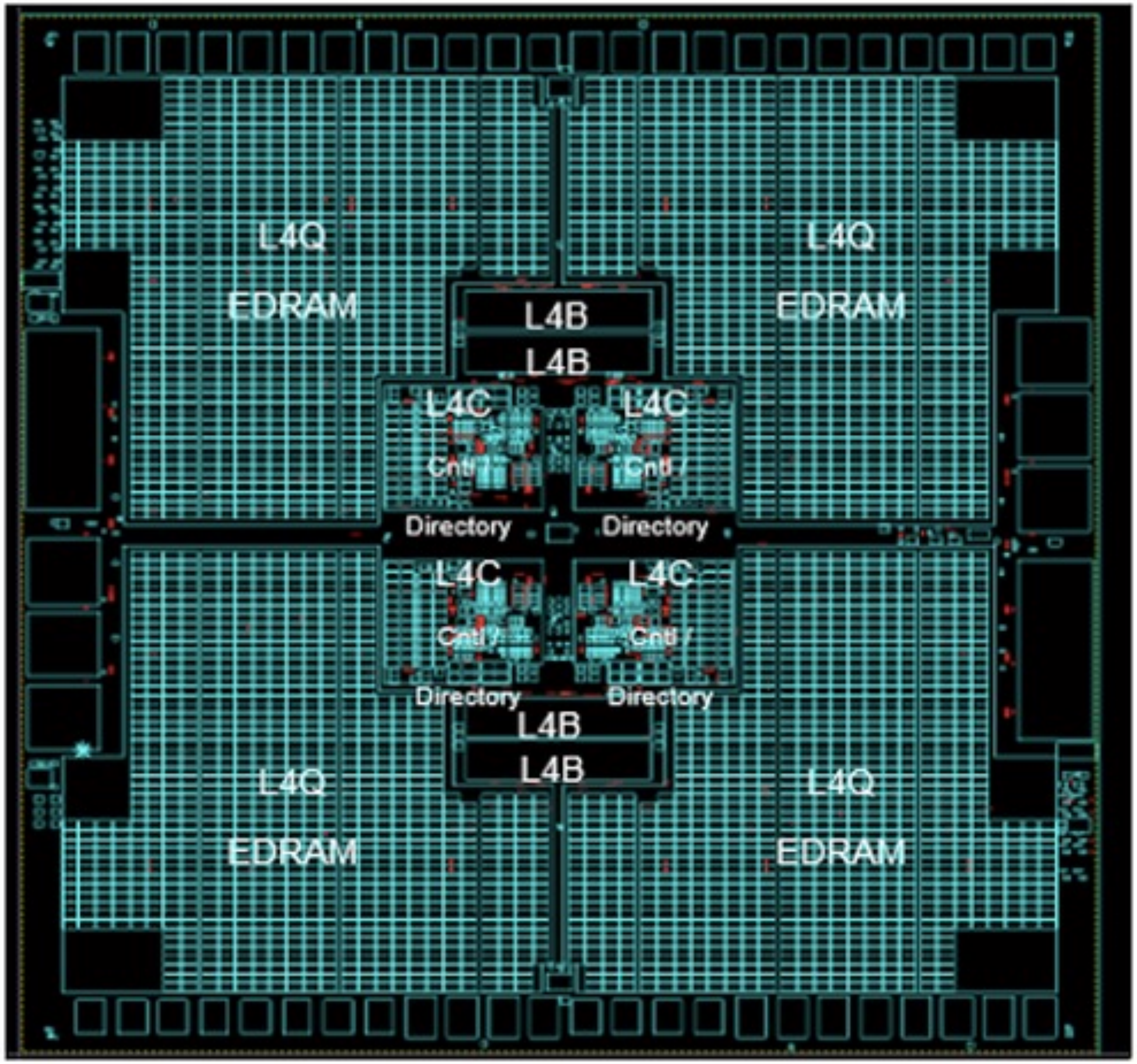
Cache Hierarchy

One unusual feature of IBM Mainframe CPUs is their cache hierarchy. Both in scope and implementation. L1D and L1I caches are fairly large at 128 kB each. L2 caches are huge in comparison to Intel and AMD CPUs, and they are also split into dedicated L2D and L2I caches, a whopping 4MB each per core. Compare that to the usual 1 - 2 MB unified L2 per core for Intel or AMD. The L3 cache is also special in that instead of using SRAM cells, the L3 cache is implemented using embedded DRAM, which increases latency, but allows IBM to place much more L3 cache into a given silicon area. For the z15, that is 256 MB of unified L3 cache per CPU chip. Really unique to the IBM zSeries cache hierarchy is an additional off-chip unified L4 cache, also implemented using eDRAM, in case of the z15 that’s 960 MB of L4, shared by four CPU chips. All of that cache is used to feed those beefy and fast execution units with as much data and instructions as possible to avoid bubbles in the deep pipelines.

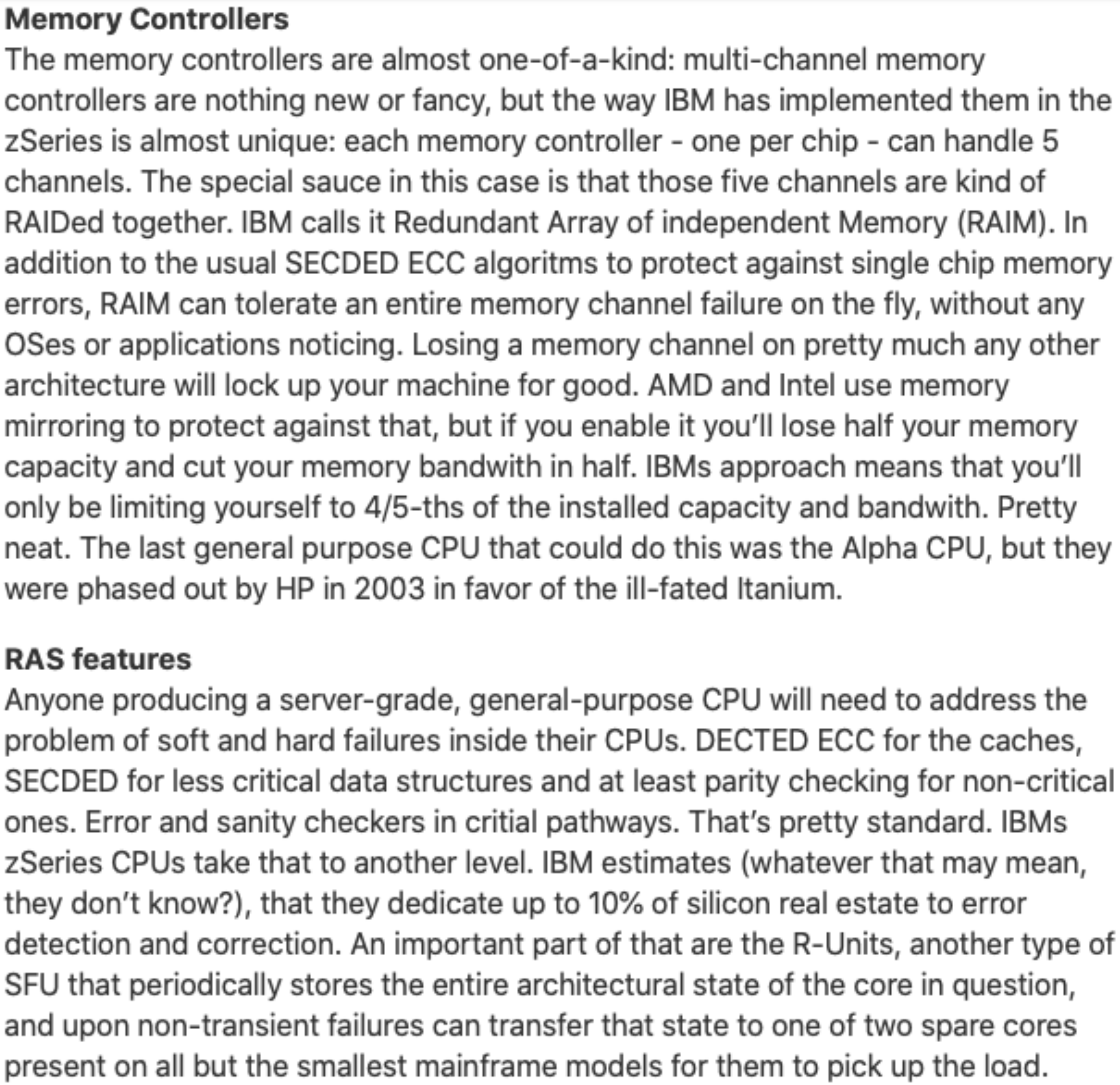
IBM CPU’s: Cache

L4 OFF-CHIP

L4 Cache Chip floorplan:



IBM CPU's: Memory ECC



Memory Controllers

The memory controllers are almost one-of-a-kind: multi-channel memory controllers are nothing new or fancy, but the way IBM has implemented them in the zSeries is almost unique: each memory controller - one per chip - can handle 5 channels. The special sauce in this case is that those five channels are kind of RAIDed together. IBM calls it Redundant Array of independent Memory (RAIM). In addition to the usual SECDED ECC algoritms to protect against single chip memory errors, RAIM can tolerate an entire memory channel failure on the fly, without any OSes or applications noticing. Losing a memory channel on pretty much any other architecture will lock up your machine for good. AMD and Intel use memory mirroring to protect against that, but if you enable it you'll lose half your memory capacity and cut your memory bandwith in half. IBMs approach means that you'll only be limiting yourself to 4/5-ths of the installed capacity and bandwith. Pretty neat. The last general purpose CPU that could do this was the Alpha CPU, but they were phased out by HP in 2003 in favor of the ill-fated Itanium.

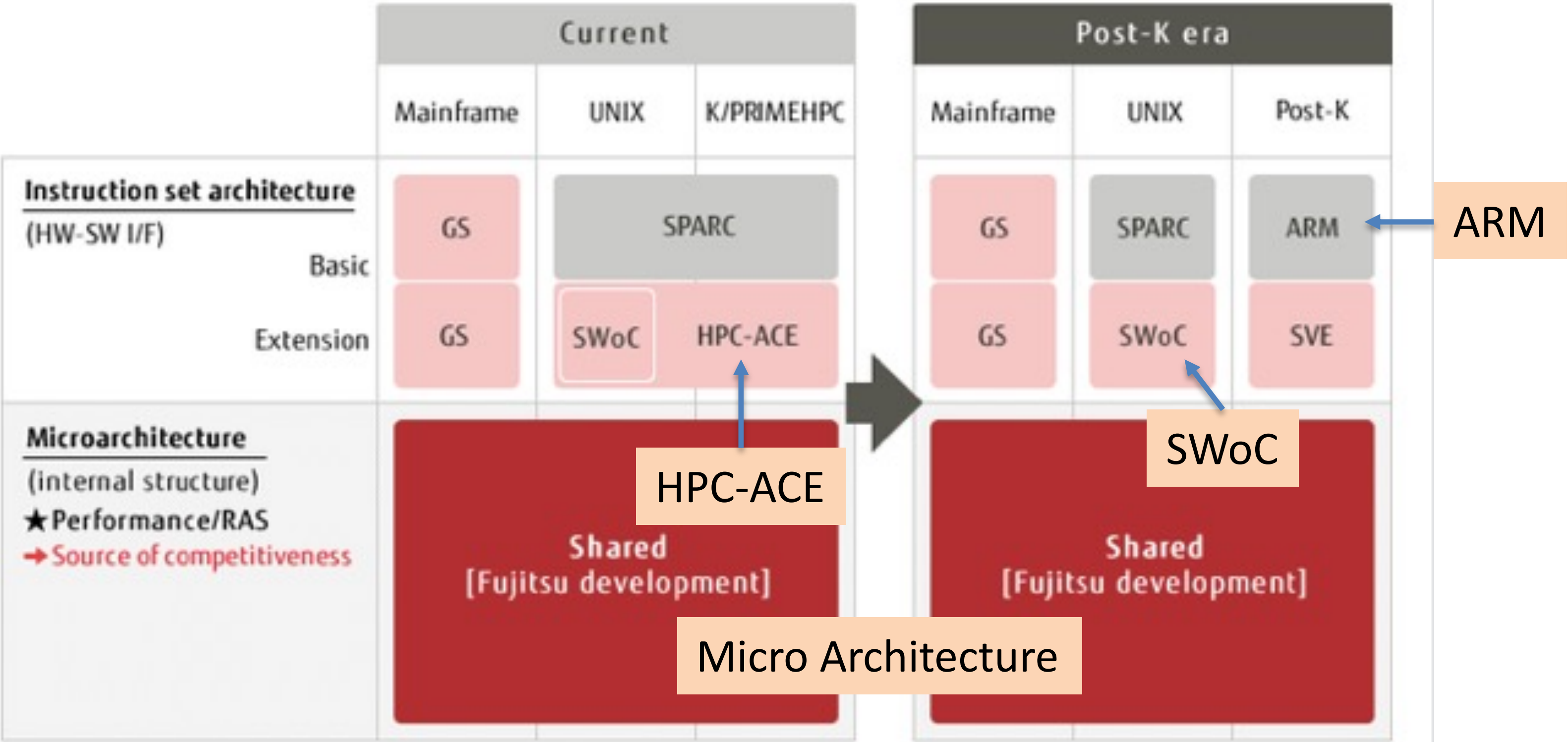
RAS features

Anyone producing a server-grade, general-purpose CPU will need to address the problem of soft and hard failures inside their CPUs. DECTED ECC for the caches, SECDED for less critical data structures and at least parity checking for non-critical ones. Error and sanity checkers in critial pathways. That's pretty standard. IBMs zSeries CPUs take that to another level. IBM estimates (whatever that may mean, they don't know?), that they dedicate up to 10% of silicon real estate to error detection and correction. An important part of that are the R-Units, another type of SFU that periodically stores the entire architectural state of the core in question, and upon non-transient failures can transfer that state to one of two spare cores present on all but the smallest mainframe models for them to pick up the load.

Fujitsu Sparc SoC

Common Microarchitecture

The Microarchitecture of the Post-K and other Fujitsu processors will be uniform.
This is Fujitsu's core competency (design and manufacturing)



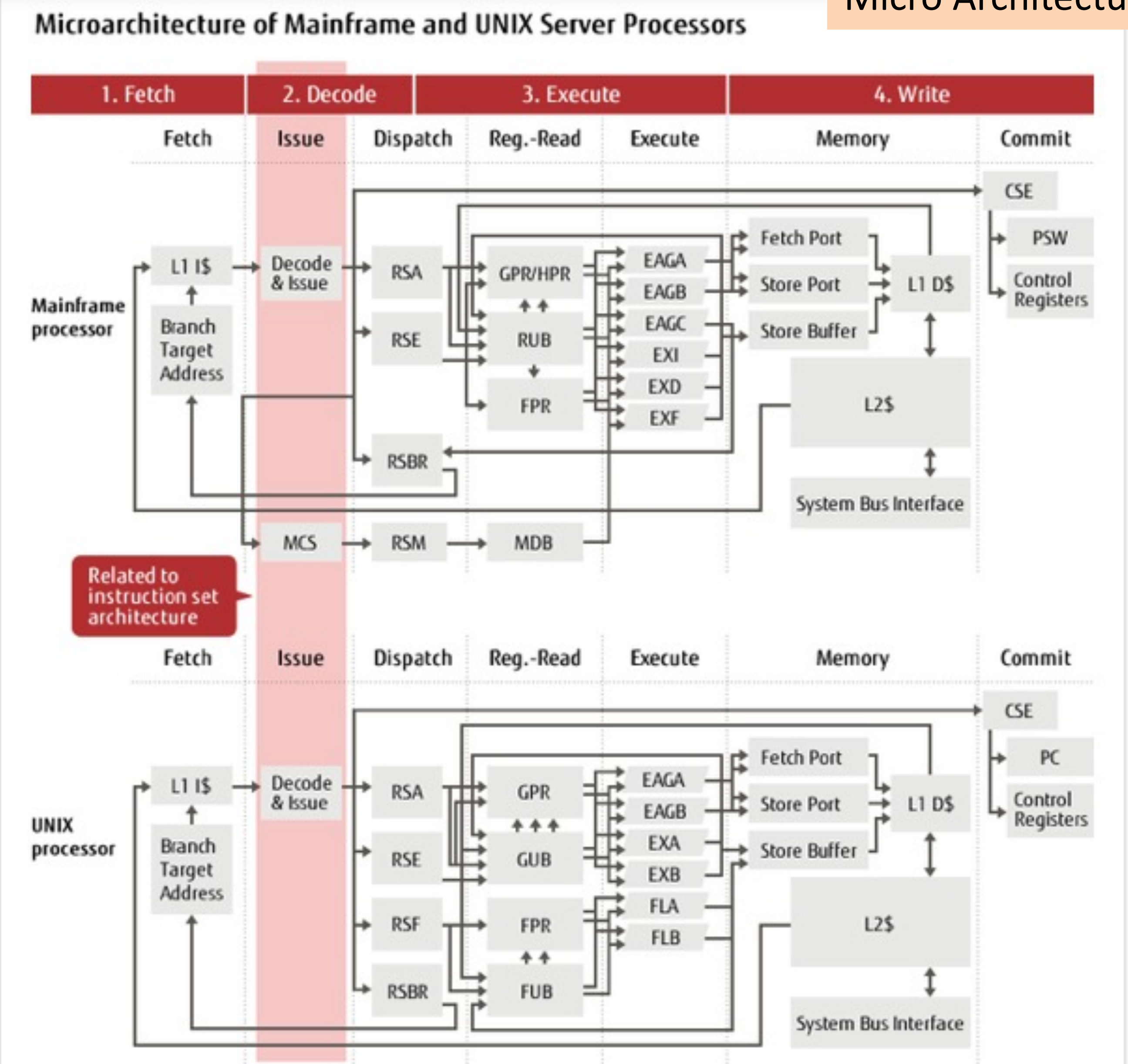
	Current			Post-K era		
	Mainframe	UNIX	K/PRIMEHPC	Mainframe	UNIX	Post-K
Instruction set architecture (HW-SW I/F) Basic	GS	SPARC		GS	SPARC	ARM
Extension	GS	SWoC	HPC-ACE	GS	SWoC	SVE
Microarchitecture (internal structure) ★Performance/RAS ➔Source of competitiveness	Shared [Fujitsu development]			Shared [Fujitsu development]		

SWoC (Software on Chip) : Fujitsu extended instructions for UNIX server
HPC-ACE (High Performance Computing – Arithmetic Computing Extensions) :
Fujitsu extended instructions for supercomputers

Notice how they're sliding in an ARM in place of SPARC, but they're using a common microarchitecture underneath.

Fujitsu Sparc SoC

Micro Architecture



Modern CPU's

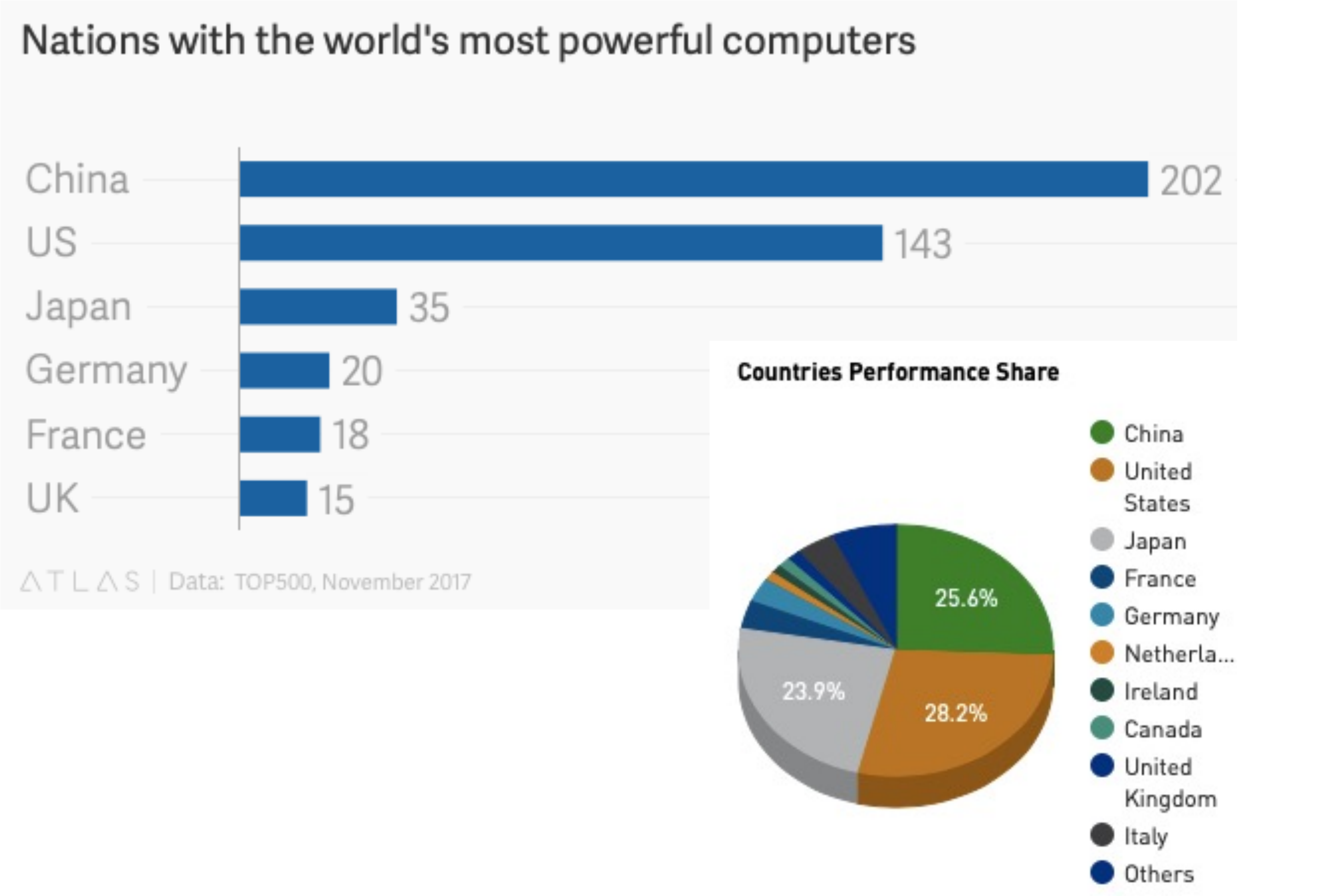
Supers

- Supercomputers
- PEZY SC-2

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

Supercomputers

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021



Nations with the world's most powerful computers

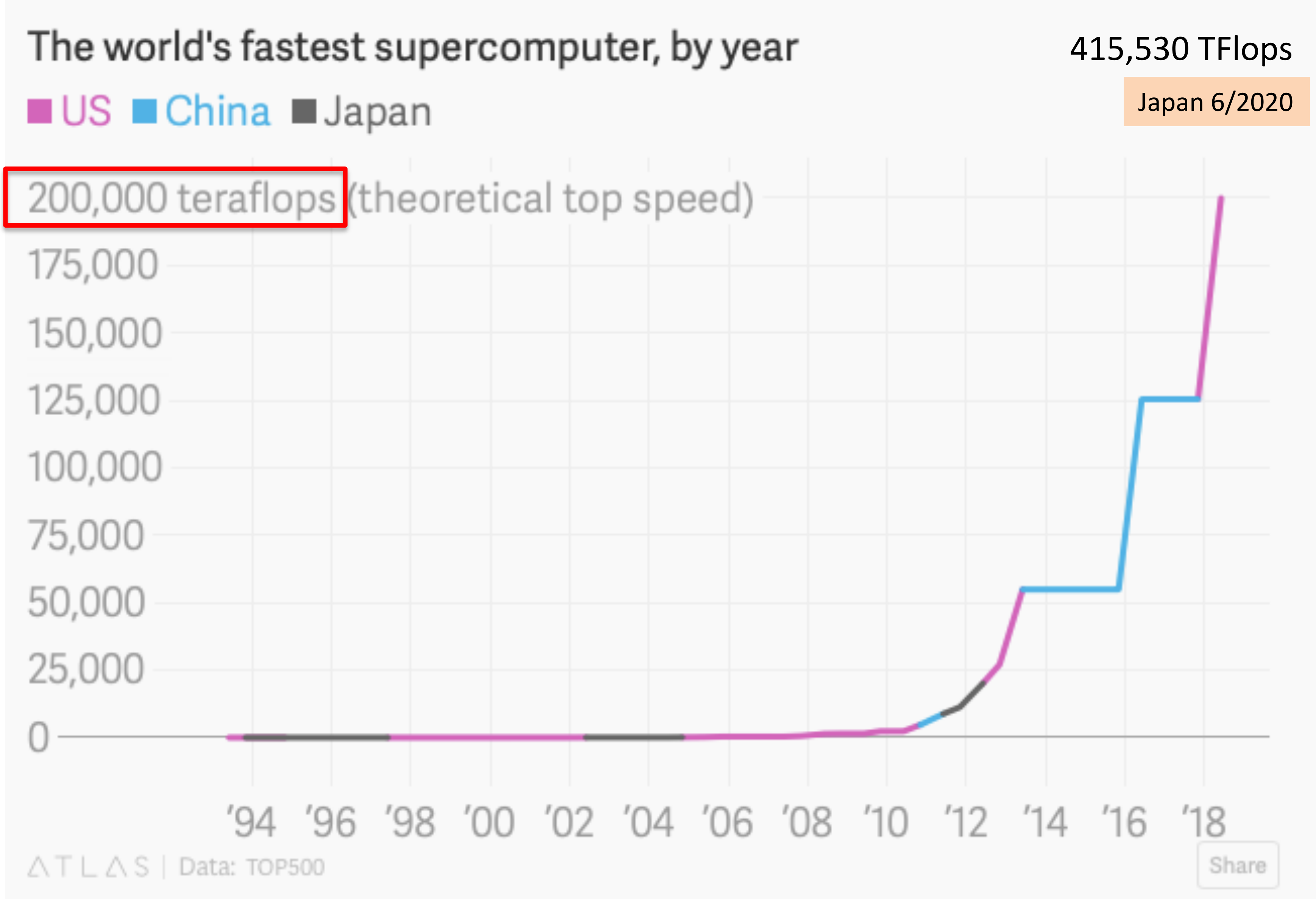
ATLAS | Data: TOP500, November 2017

Countries Performance Share

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

Supercomputers

DR JEFF SOFTWARE
INDIE APP DEVELOPER

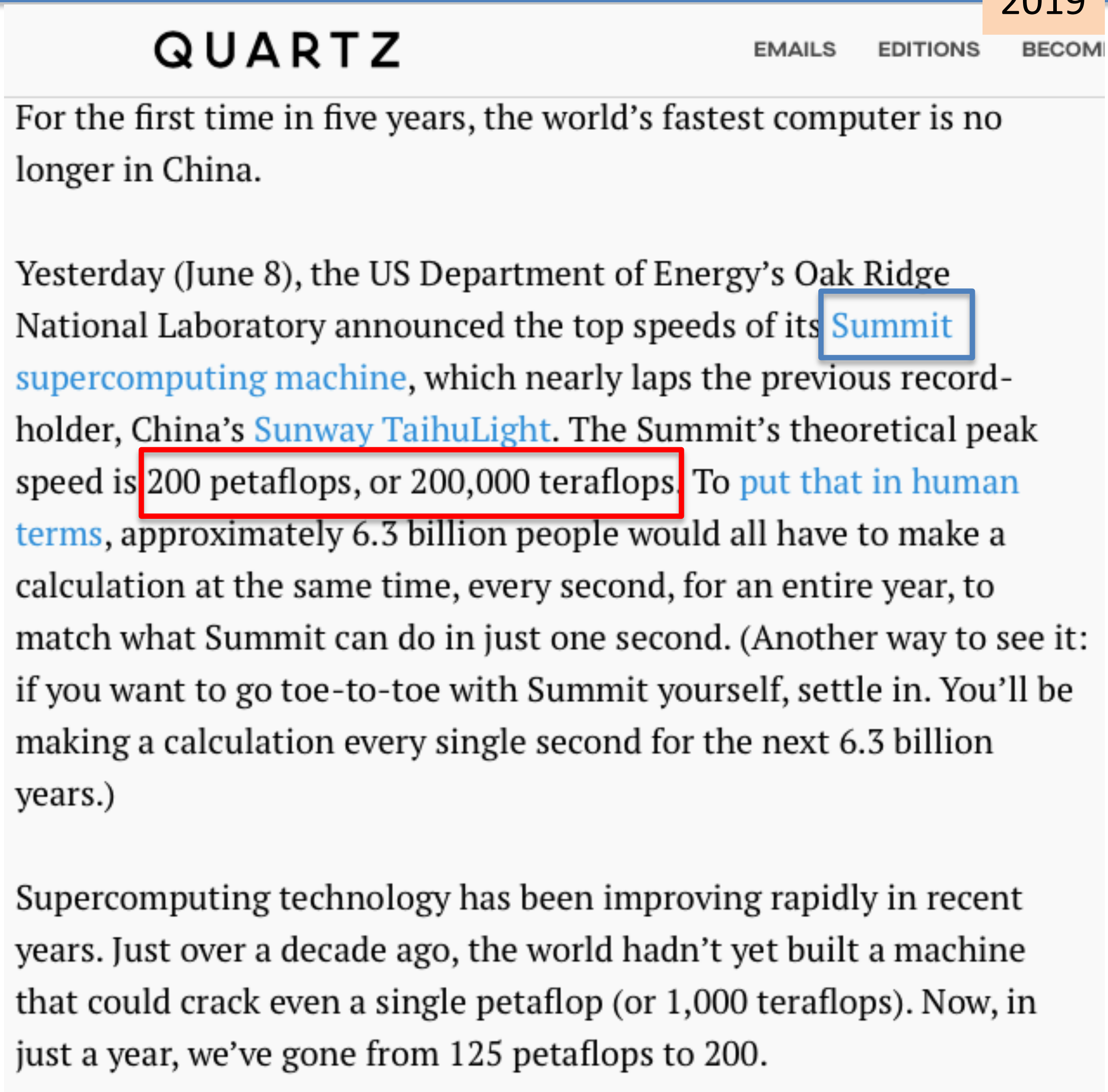


415,530 TFlops

Japan 6/2020

Supercomputers

2019

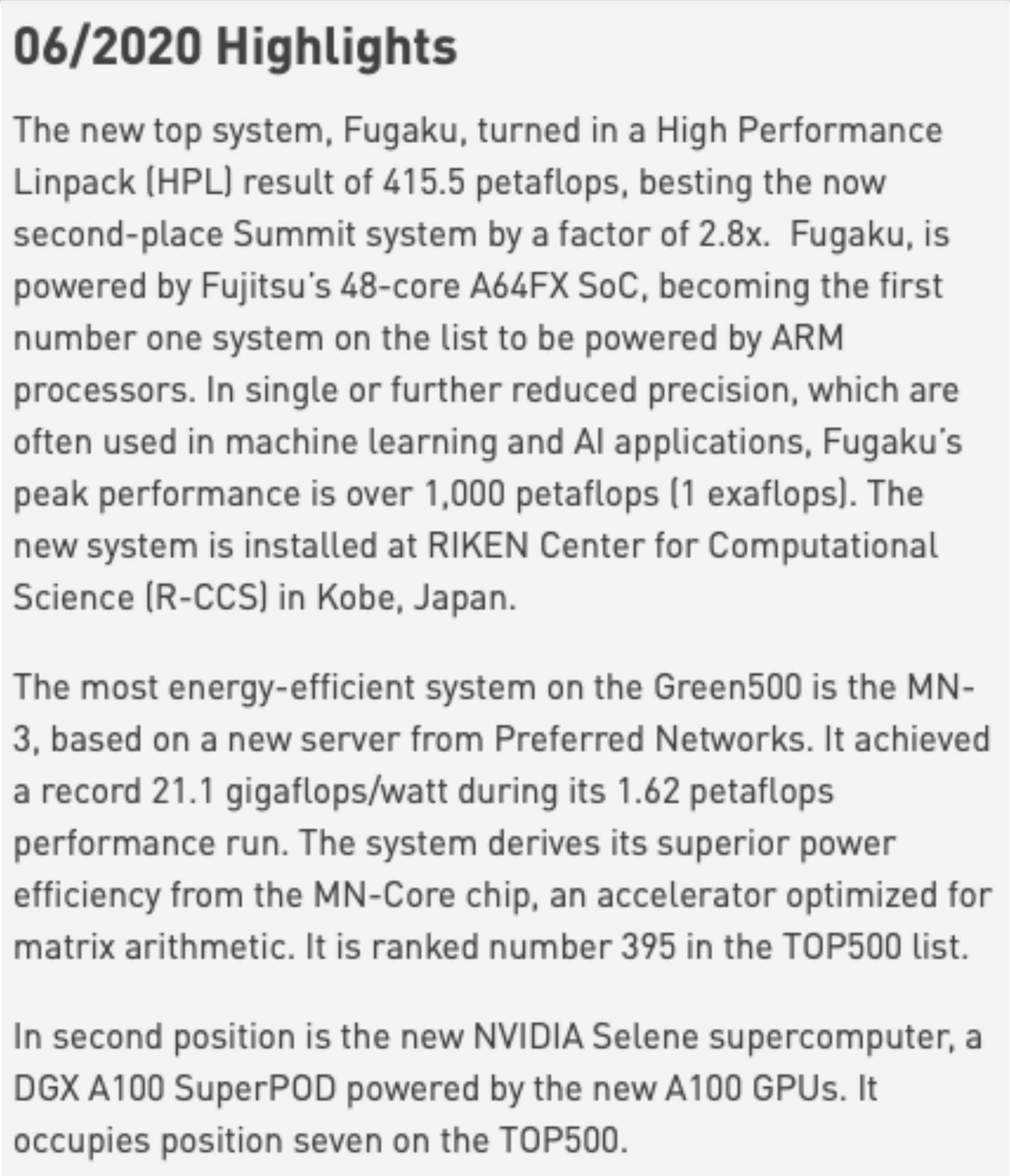


For the first time in five years, the world's fastest computer is no longer in China.

Yesterday (June 8), the US Department of Energy's Oak Ridge National Laboratory announced the top speeds of its Summit supercomputing machine, which nearly laps the previous record-holder, China's Sunway TaihuLight. The Summit's theoretical peak speed is 200 petaflops, or 200,000 teraflops. To put that in human terms, approximately 6.3 billion people would all have to make a calculation at the same time, every second, for an entire year, to match what Summit can do in just one second. (Another way to see it: if you want to go toe-to-toe with Summit yourself, settle in. You'll be making a calculation every single second for the next 6.3 billion years.)

Supercomputing technology has been improving rapidly in recent years. Just over a decade ago, the world hadn't yet built a machine that could crack even a single petaflop (or 1,000 teraflops). Now, in just a year, we've gone from 125 petaflops to 200.

Top500 List: Top 2



06/2020 Highlights

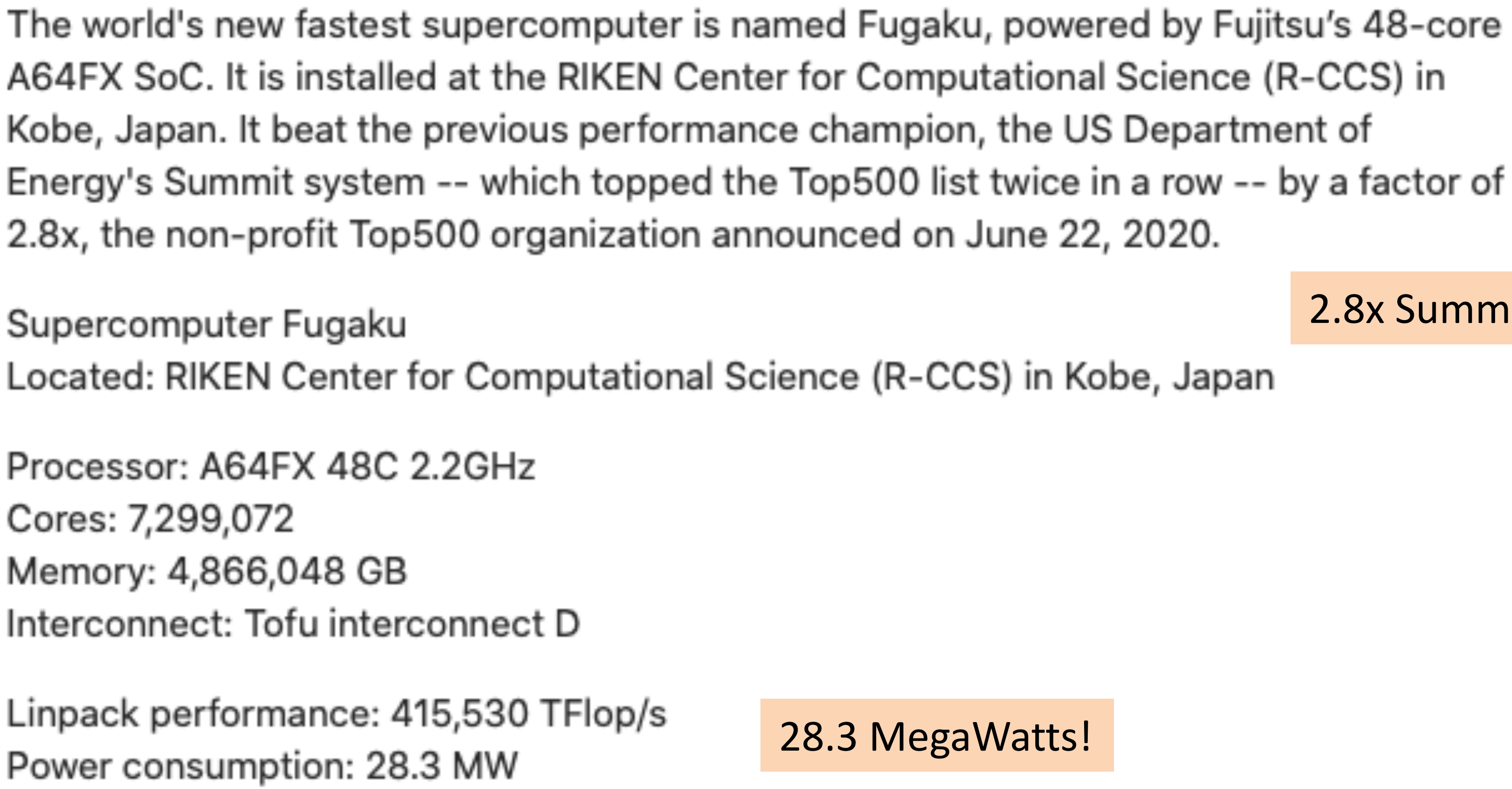
The new top system, Fugaku, turned in a High Performance Linpack (HPL) result of 415.5 petaflops, besting the now second-place Summit system by a factor of 2.8x. Fugaku, is powered by Fujitsu's 48-core A64FX SoC, becoming the first number one system on the list to be powered by ARM processors. In single or further reduced precision, which are often used in machine learning and AI applications, Fugaku's peak performance is over 1,000 petaflops (1 exaflops). The new system is installed at RIKEN Center for Computational Science (R-CCS) in Kobe, Japan.

The most energy-efficient system on the Green500 is the MN-3, based on a new server from Preferred Networks. It achieved a record 21.1 gigaflops/watt during its 1.62 petaflops performance run. The system derives its superior power efficiency from the MN-Core chip, an accelerator optimized for matrix arithmetic. It is ranked number 395 in the TOP500 list.

In second position is the new NVIDIA Selene supercomputer, a DGX A100 SuperPOD powered by the new A100 GPUs. It occupies position seven on the TOP500.

Leapfrog: Fugaku

2020



The world's new fastest supercomputer is named Fugaku, powered by Fujitsu's 48-core A64FX SoC. It is installed at the RIKEN Center for Computational Science (R-CCS) in Kobe, Japan. It beat the previous performance champion, the US Department of Energy's Summit system -- which topped the Top500 list twice in a row -- by a factor of 2.8x, the non-profit Top500 organization announced on June 22, 2020.

Supercomputer Fugaku
Located: RIKEN Center for Computational Science (R-CCS) in Kobe, Japan

Processor: A64FX 48C 2.2GHz
Cores: 7,299,072
Memory: 4,866,048 GB
Interconnect: Tofu interconnect D

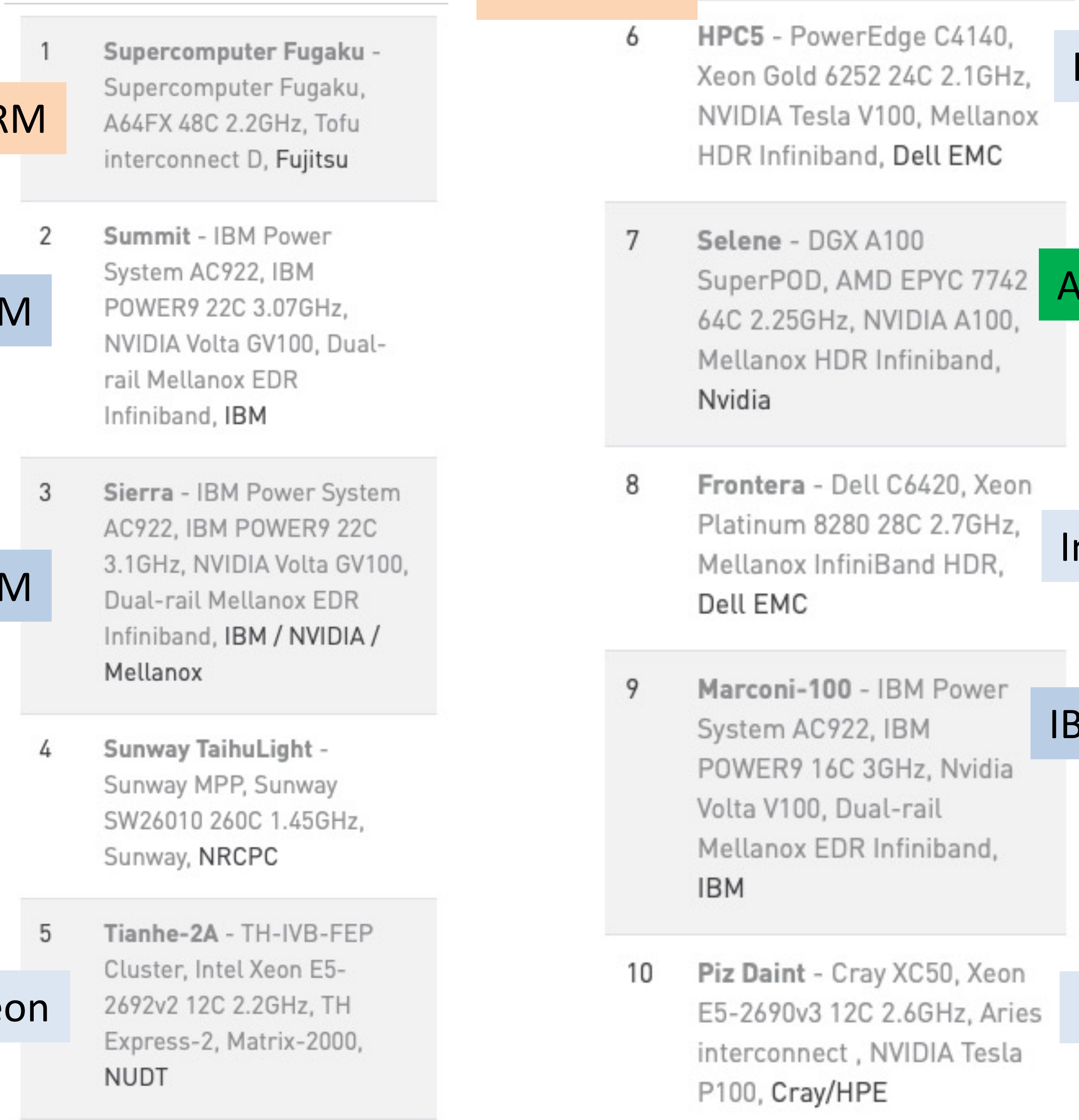
Linpack performance: 415,530 TFlop/s
Power consumption: 28.3 MW

2.8x Summit

28.3 MegaWatts!

Top500 List: Top 10

June 2020



1 **Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu

ARM

2 **Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM

IBM

3 **Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox

IBM

4 **Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC

5 **Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT

Intel Xeon

6 **HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC

Intel Xeon

7 **Selene** - DGX A100 SuperPOD, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia

AMD Epyc

8 **Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR, Dell EMC

Intel Xeon

9 **Marconi-100** - IBM Power System AC922, IBM POWER9 16C 3GHz, Nvidia Volta V100, Dual-rail Mellanox EDR Infiniband, IBM

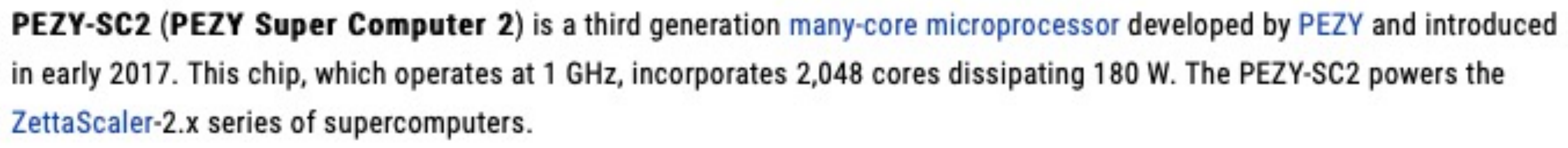
IBM

10 **Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100, Cray/HPE

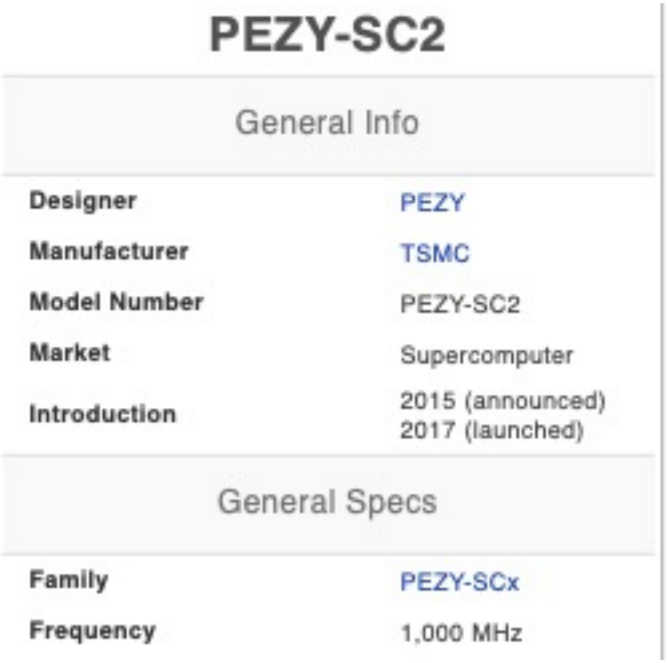
Intel Xeon

PEZY-SC2

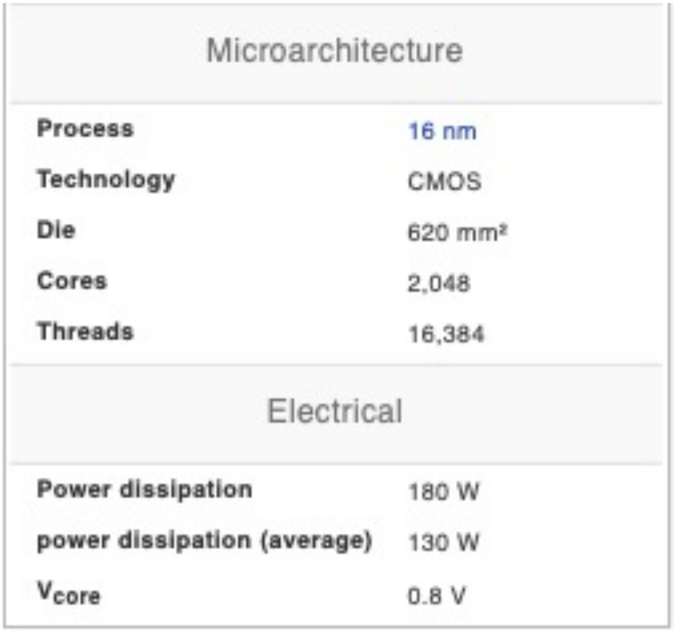
Supercomputer Chip



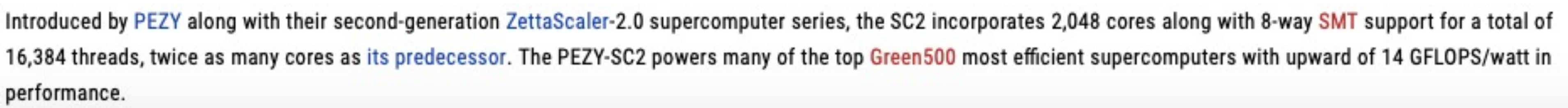
PEZY-SC2 (**PEZY Super Computer 2**) is a third generation many-core microprocessor developed by PEZY and introduced in early 2017. This chip, which operates at 1 GHz, incorporates 2,048 cores dissipating 180 W. The PEZY-SC2 powers the ZettaScaler-2.x series of supercomputers.



PEZY-SC2	
General Info	
Designer	PEZY
Manufacturer	TSMC
Model Number	PEZY-SC2
Market	Supercomputer
Introduction	2015 (announced) 2017 (launched)
General Specs	
Family	PEZY-SCx
Frequency	1,000 MHz



Microarchitecture	
Process	16 nm
Technology	CMOS
Die	620 mm^2
Cores	2,048
Threads	16,384
Electrical	
Power dissipation	180 W
power dissipation (average)	130 W
V_{core}	0.8 V



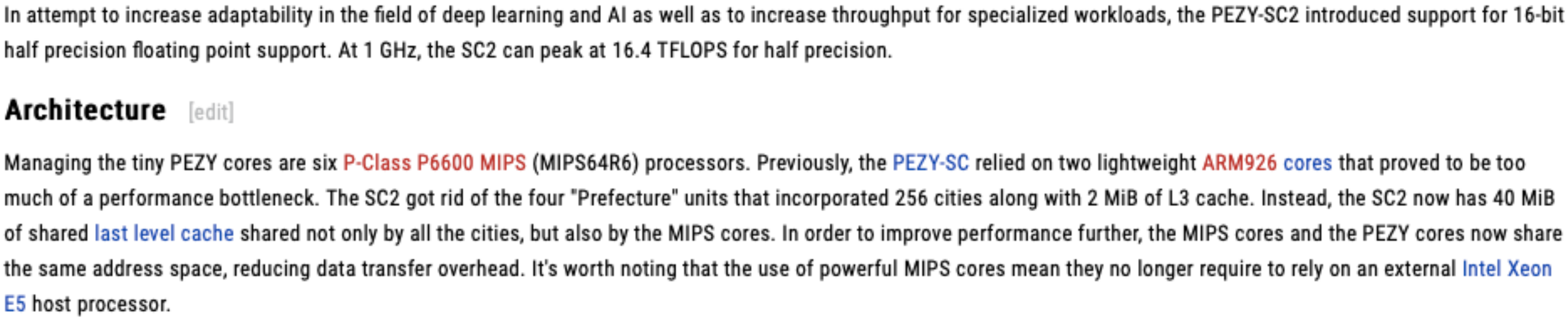
Introduced by PEZY along with their second-generation ZettaScaler-2.0 supercomputer series, the SC2 incorporates 2,048 cores along with 8-way SMT support for a total of 16,384 threads, twice as many cores as its predecessor. The PEZY-SC2 powers many of the top Green500 most efficient supercomputers with upward of 14 GFLOPS/watt in performance.

CSUN
CALIFORNIA
STATE UNIVERSITY
NORTHRIDGE
COMP122

DR JEFF
SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021

PEZY-SC2

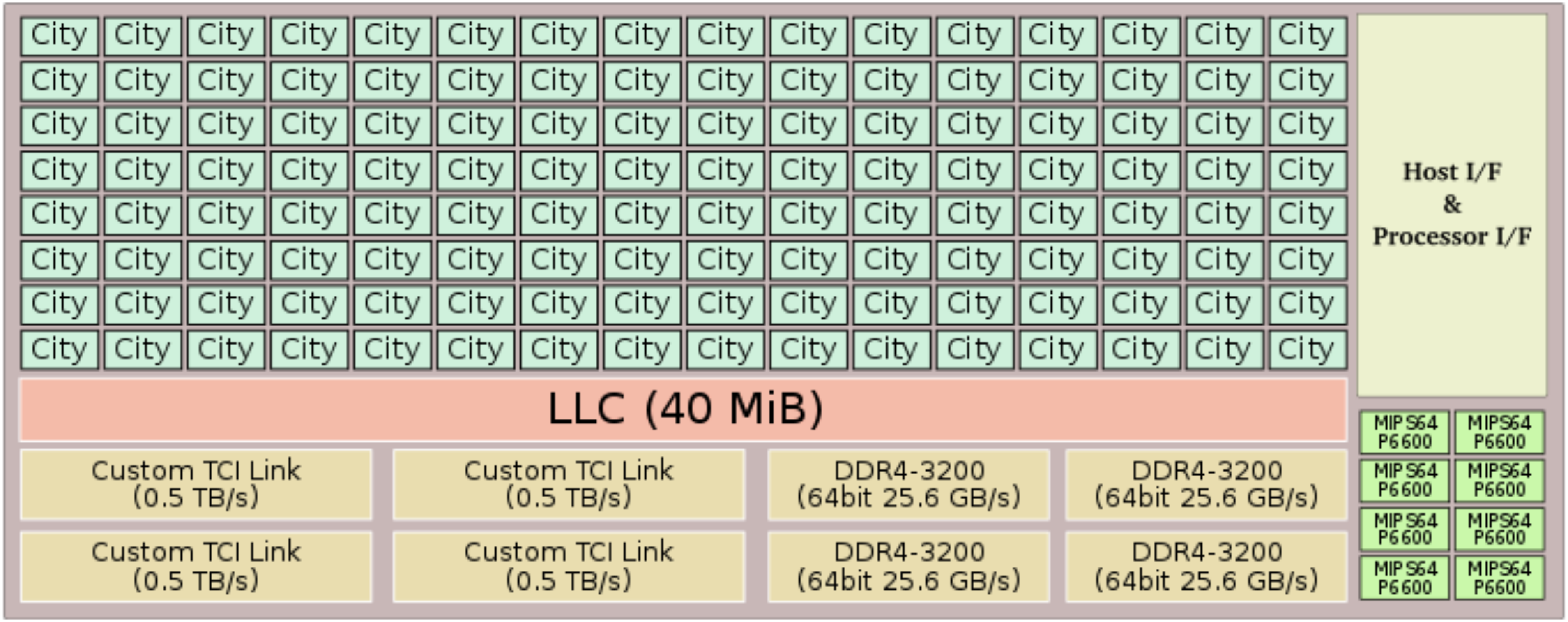
Supercomputer Chip



In attempt to increase adaptability in the field of deep learning and AI as well as to increase throughput for specialized workloads, the PEZY-SC2 introduced support for 16-bit half precision floating point support. At 1 GHz, the SC2 can peak at 16.4 TFLOPS for half precision.

Architecture [edit]

Managing the tiny PEZY cores are six P-Class P6600 MIPS (MIPS64R6) processors. Previously, the PEZY-SC relied on two lightweight ARM926 cores that proved to be too much of a performance bottleneck. The SC2 got rid of the four "Prefecture" units that incorporated 256 cities along with 2 MiB of L3 cache. Instead, the SC2 now has 40 MiB of shared last level cache shared not only by all the cities, but also by the MIPS cores. In order to improve performance further, the MIPS cores and the PEZY cores now share the same address space, reducing data transfer overhead. It's worth noting that the use of powerful MIPS cores mean they no longer require to rely on an external Intel Xeon E5 host processor.



City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City
City City City City City City City City City City City City City City City City

Host I/F
&
Processor I/F

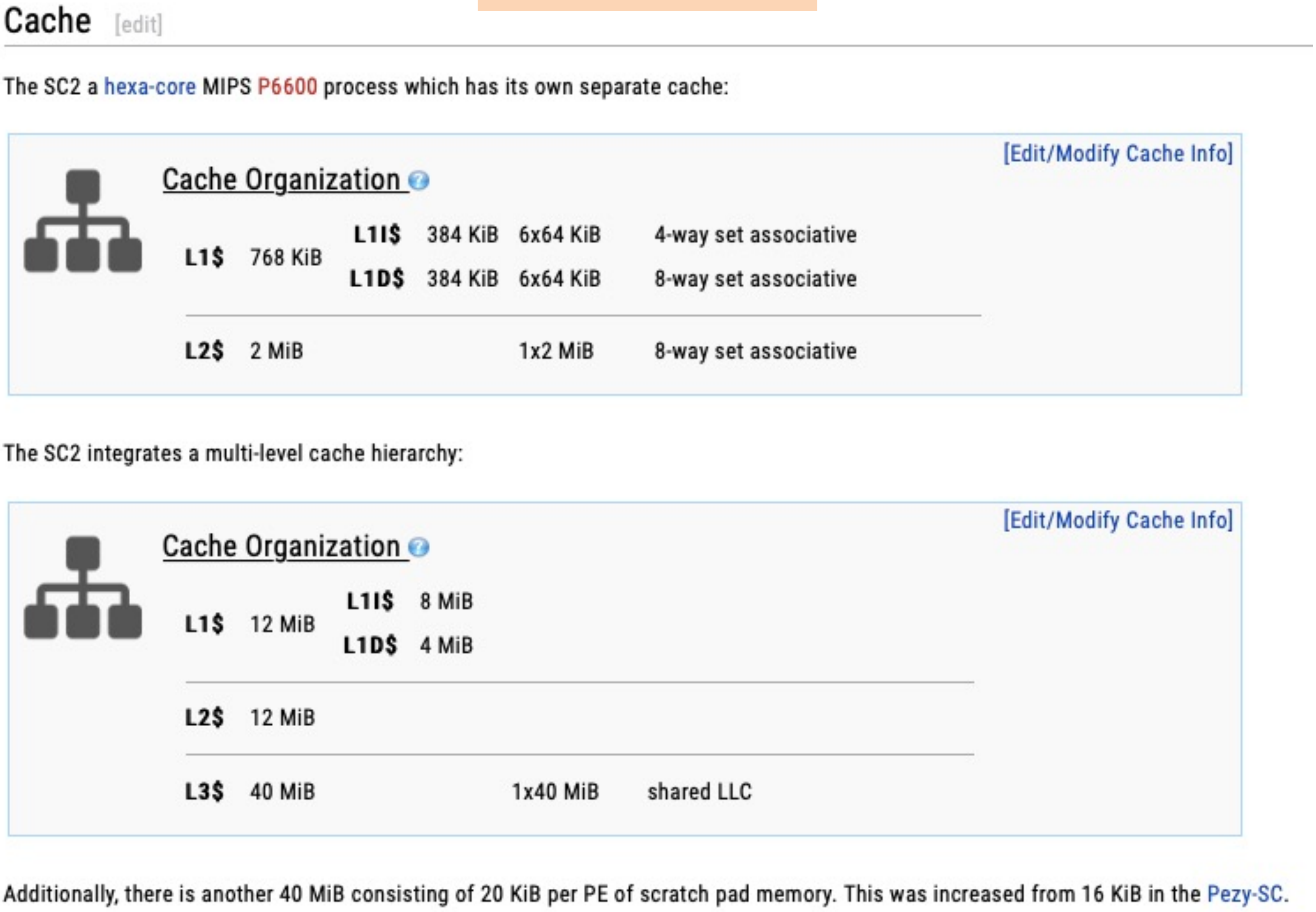
LLC (40 MiB)

MIPS64 P6600 | MIPS64 P6600
MIPS64 P6600 | MIPS64 P6600
MIPS64 P6600 | MIPS64 P6600
MIPS64 P6600 | MIPS64 P6600

Custom TCI Link (0.5 TB/s)
Custom TCI Link (0.5 TB/s)
DDR4-3200 (64bit 25.6 GB/s)
DDR4-3200 (64bit 25.6 GB/s)
Custom TCI Link (0.5 TB/s)
Custom TCI Link (0.5 TB/s)
DDR4-3200 (64bit 25.6 GB/s)
DDR4-3200 (64bit 25.6 GB/s)

PEZY-SC2

Supercomputer Chip



Cache [edit]

The SC2 a hexa-core MIPS P6600 process which has its own separate cache:

[Edit/Modify Cache Info]

Cache Organization

L1$	768 KiB	**L1I$**	384 KiB	6x64 KiB	4-way set associative
		L1D$	384 KiB	6x64 KiB	8-way set associative
L2$	2 MiB			1x2 MiB	8-way set associative

The SC2 integrates a multi-level cache hierarchy:

[Edit/Modify Cache Info]

Cache Organization

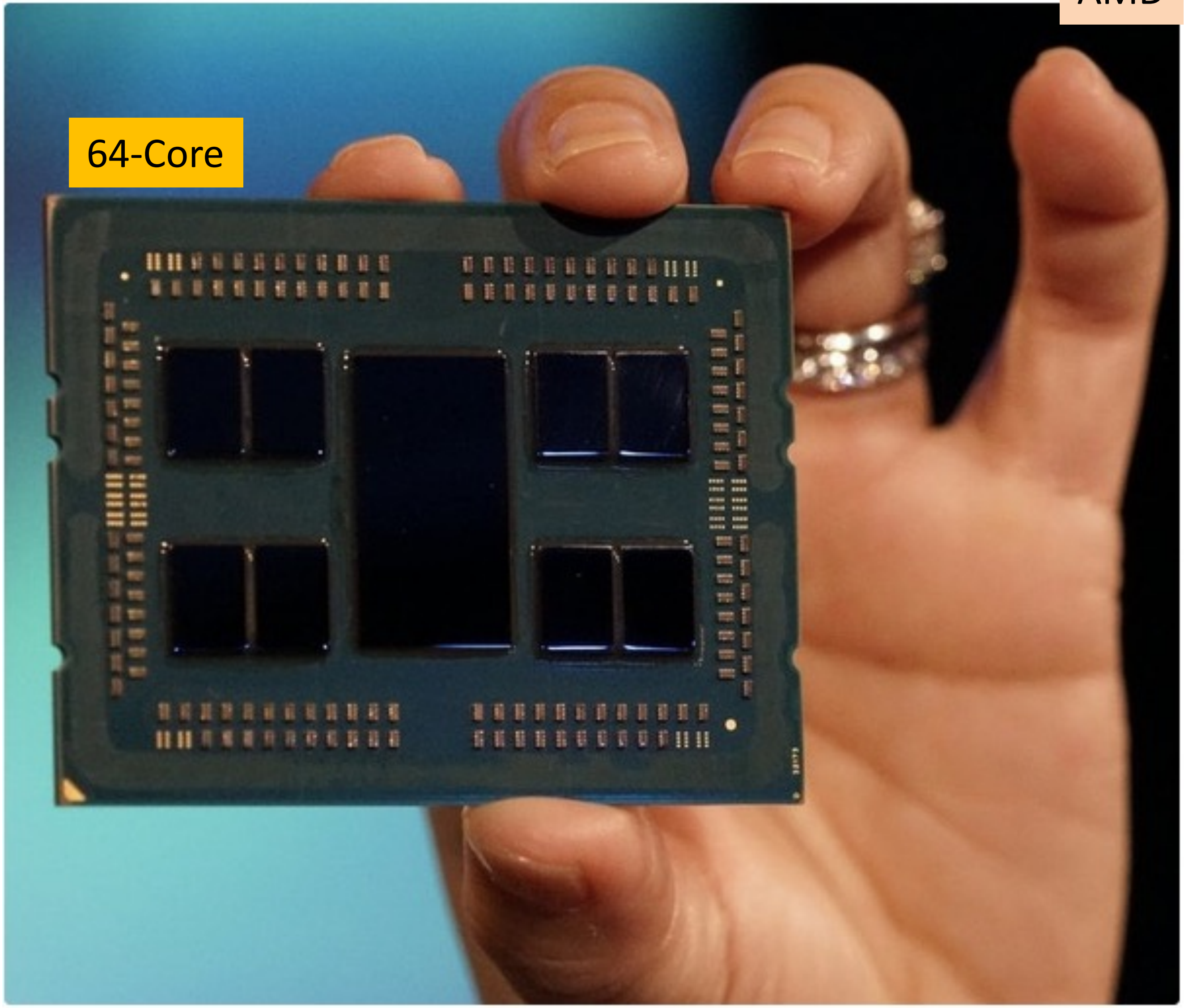
L1$	12 MiB	**L1I$**	8 MiB		
		L1D$	4 MiB		
L2$	12 MiB				
L3$	40 MiB			1x40 MiB	shared LLC

Additionally, there is another 40 MiB consisting of 20 KiB per PE of scratch pad memory. This was increased from 16 KiB in the Pezy-SC.

Macro/Micro Arch

Cores & Threads

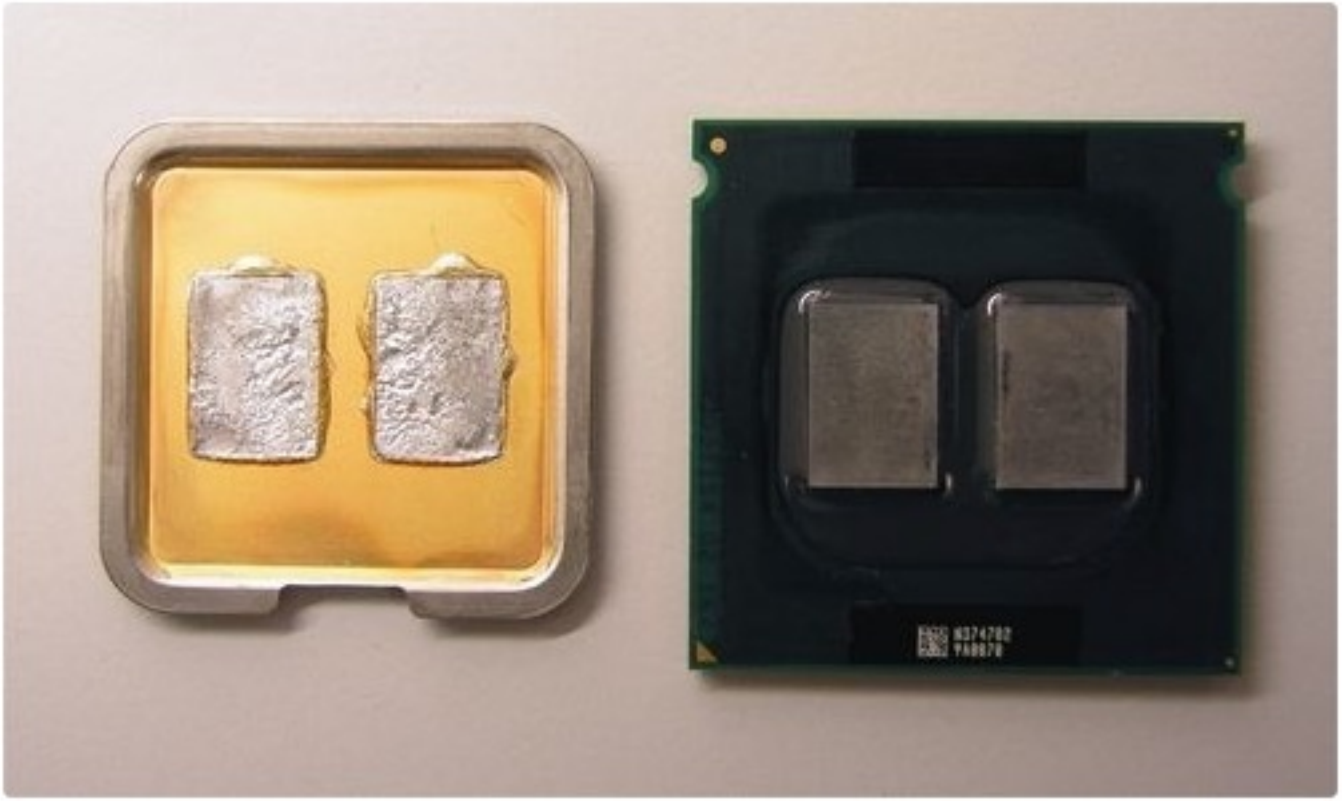
Multi-Cores + L2/L3



Lisa Su proudly shows off her 64-core EPYC monster at CES.

Multi-Cores + L2/L3

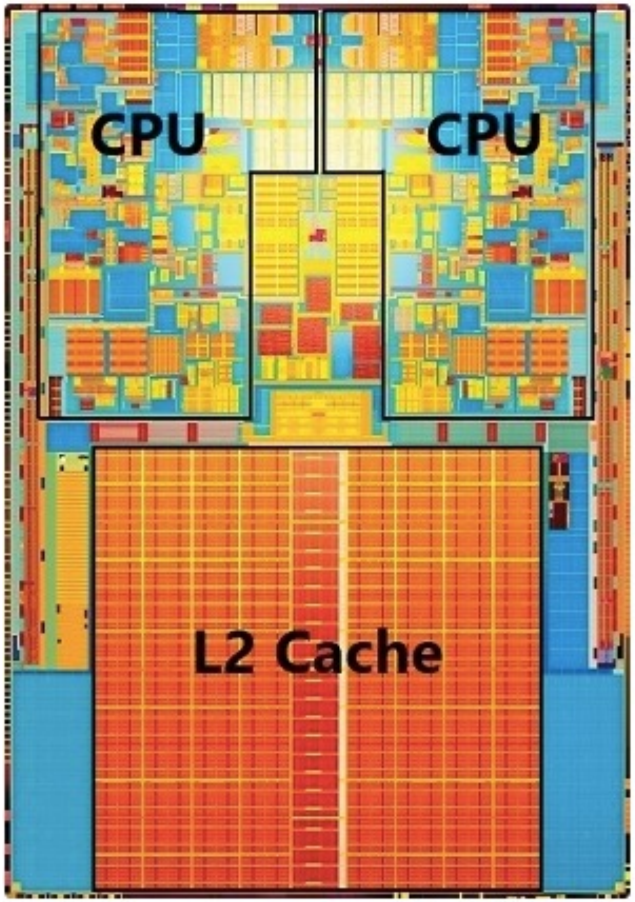
Intel

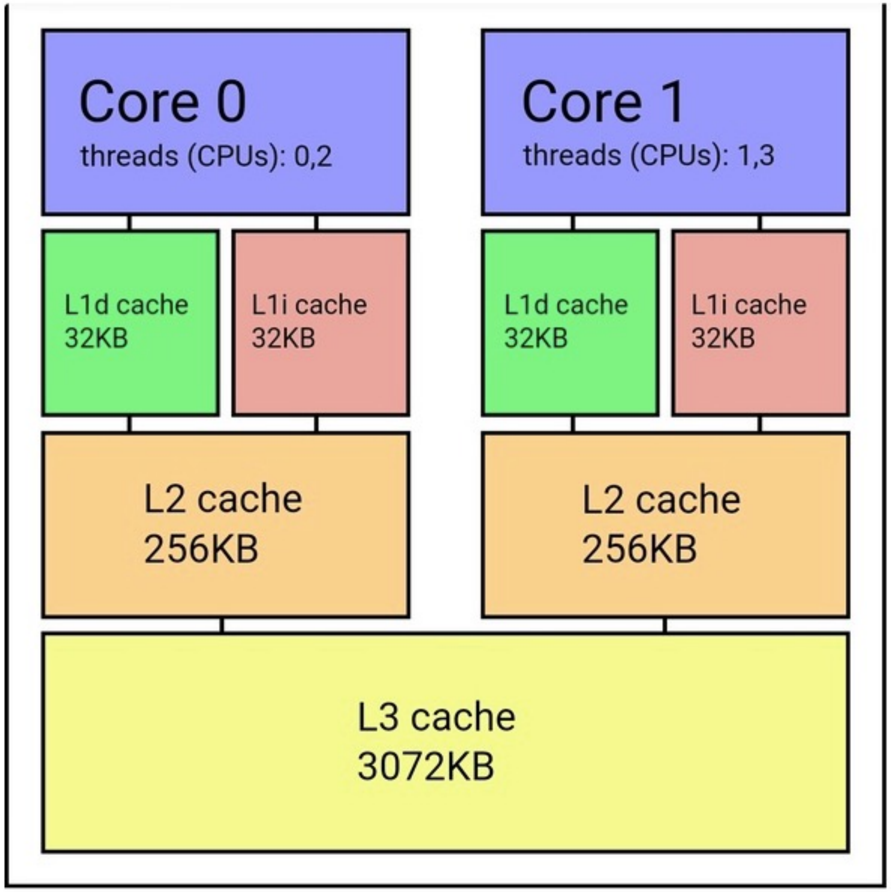


This is a delided Core 2 Quad with the two Core 2 Duo dies.

Multi-Cores + L2/L3

Intel

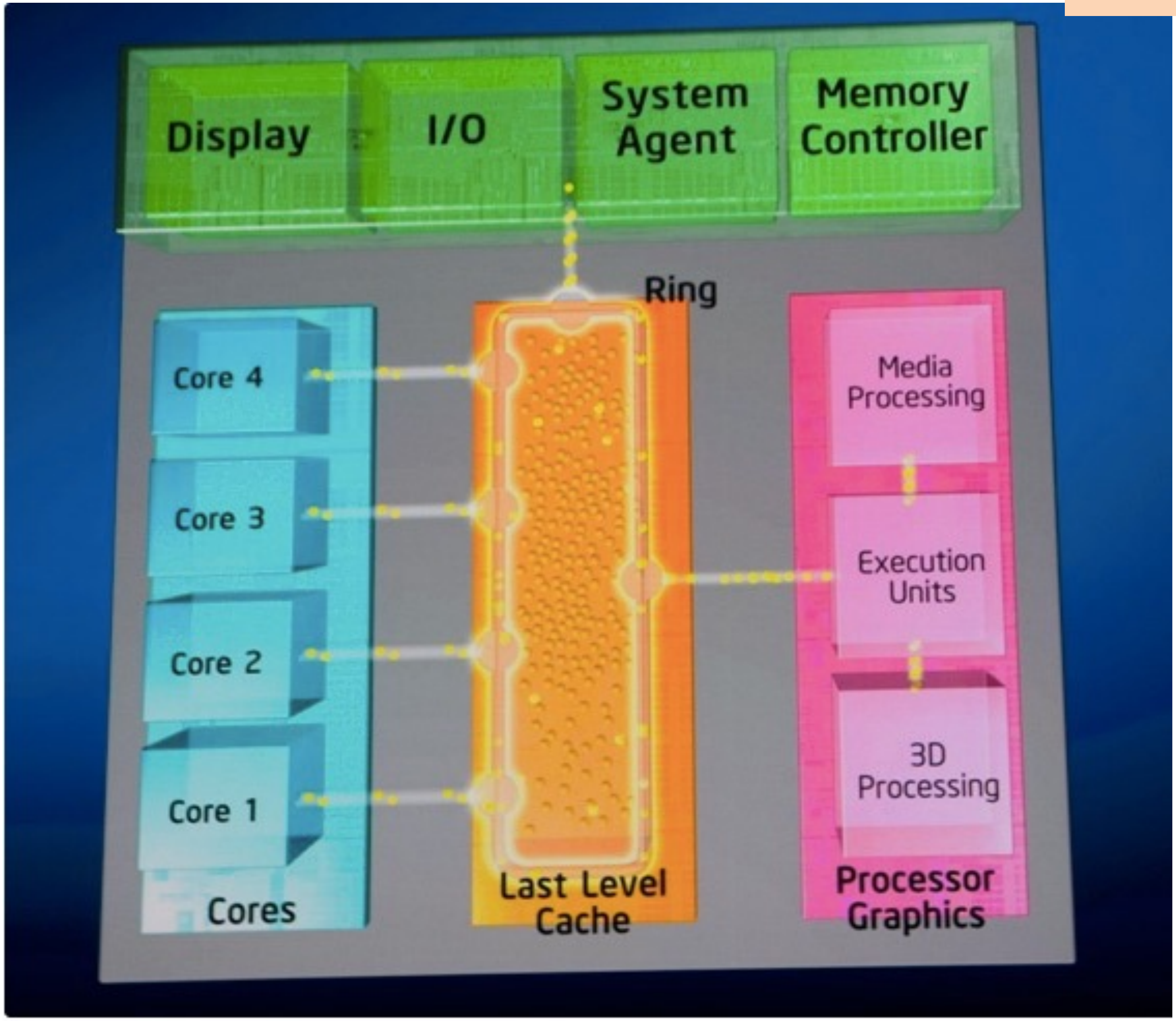




This is a late Core 2 Duo (Wolfdale), note how massive the uncore L2 is compared to the actual CPU cores. (uncore = outside the CPU)

Multi-Cores + L2/L3

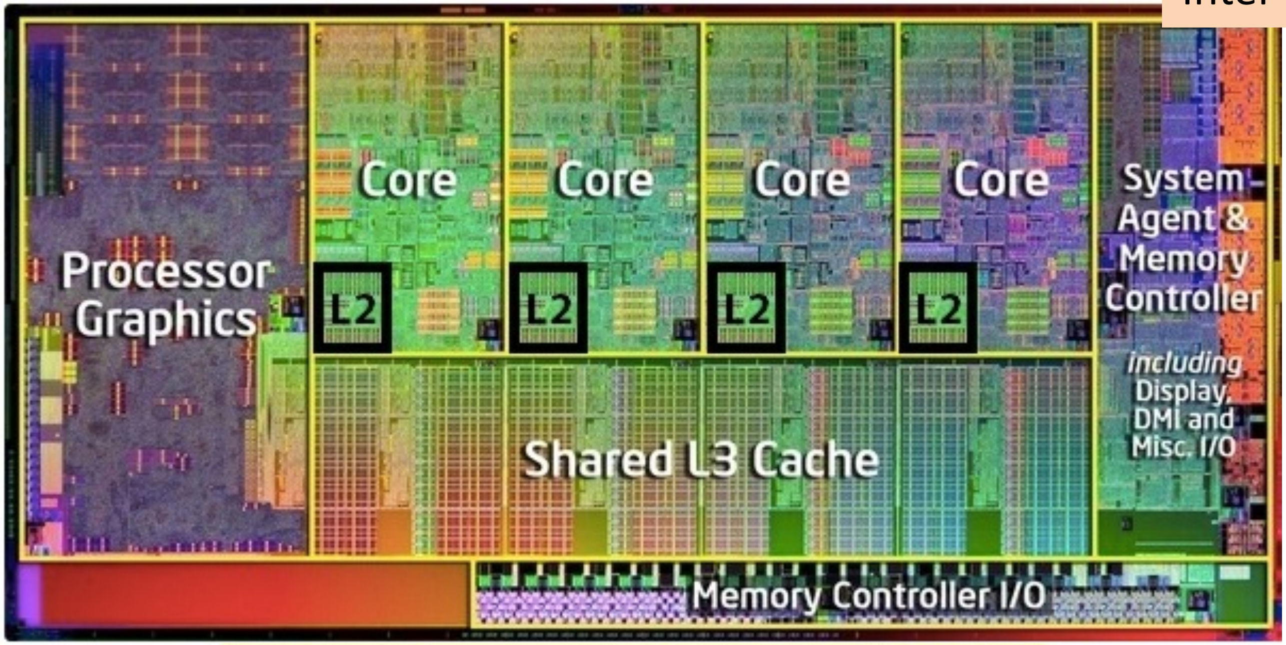
Intel

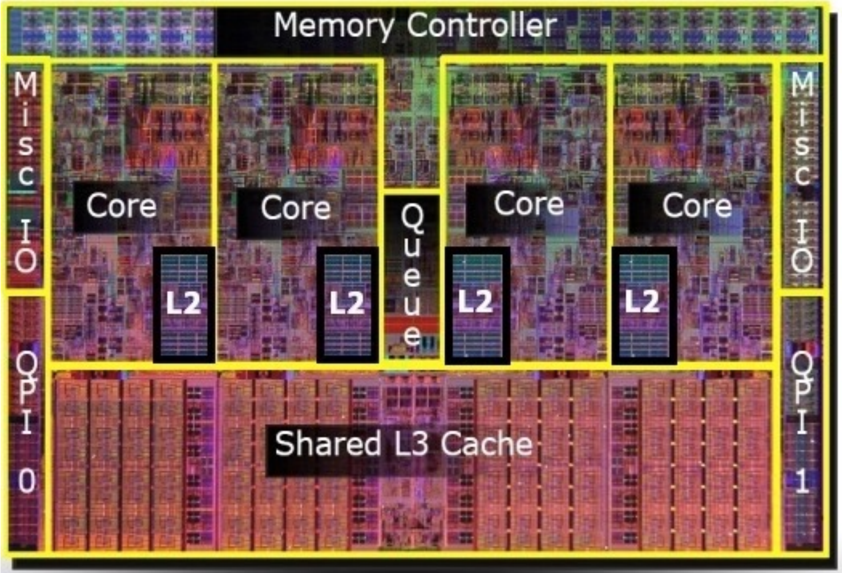


This special 'glue' remains in use today in all non-HEDT Intel CPUs.

Multi-Cores + L2/L3

Intel Core 4

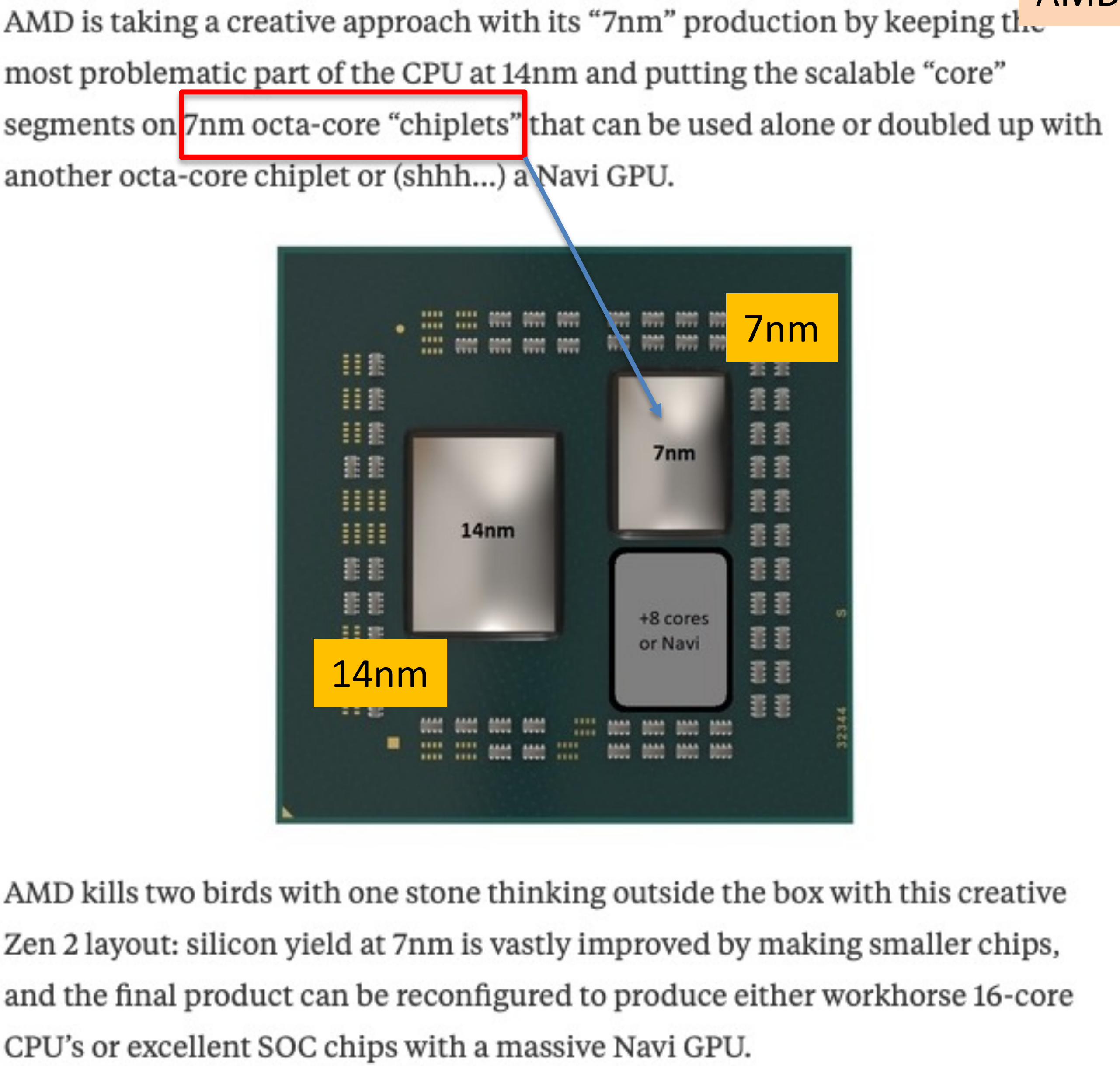




Multi-Cores + L2/L3

AMD

Zen 2

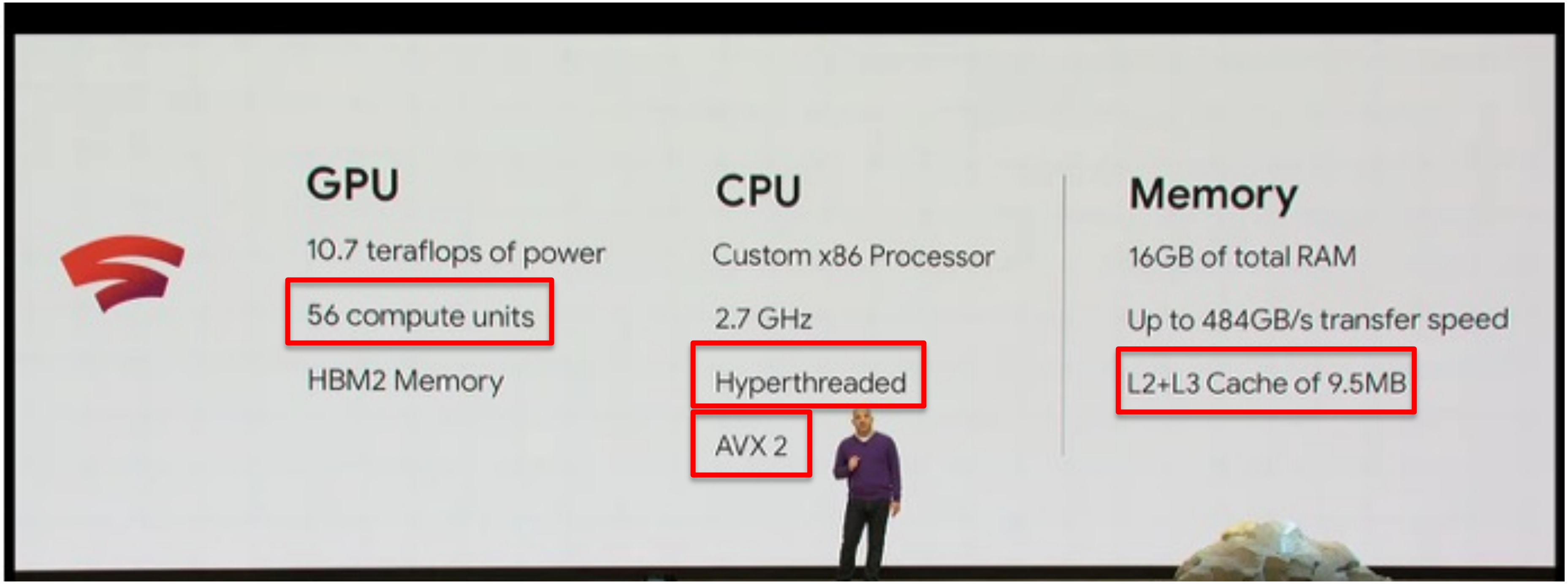


AMD is taking a creative approach with its "7nm" production by keeping the most problematic part of the CPU at 14nm and putting the scalable "core" segments on 7nm octa-core "chiplets" that can be used alone or doubled up with another octa-core chiplet or (shhh…) a Navi GPU.

AMD kills two birds with one stone thinking outside the box with this creative Zen 2 layout: silicon yield at 7nm is vastly improved by making smaller chips, and the final product can be reconfigured to produce either workhorse 16-core CPU's or excellent SOC chips with a massive Navi GPU.

Multi-Cores + L2/L3

Intel



GPU

10.7 teraflops of power

56 compute units

HBM2 Memory

CPU

Custom x86 Processor

2.7 GHz

Hyperthreaded

AVX 2

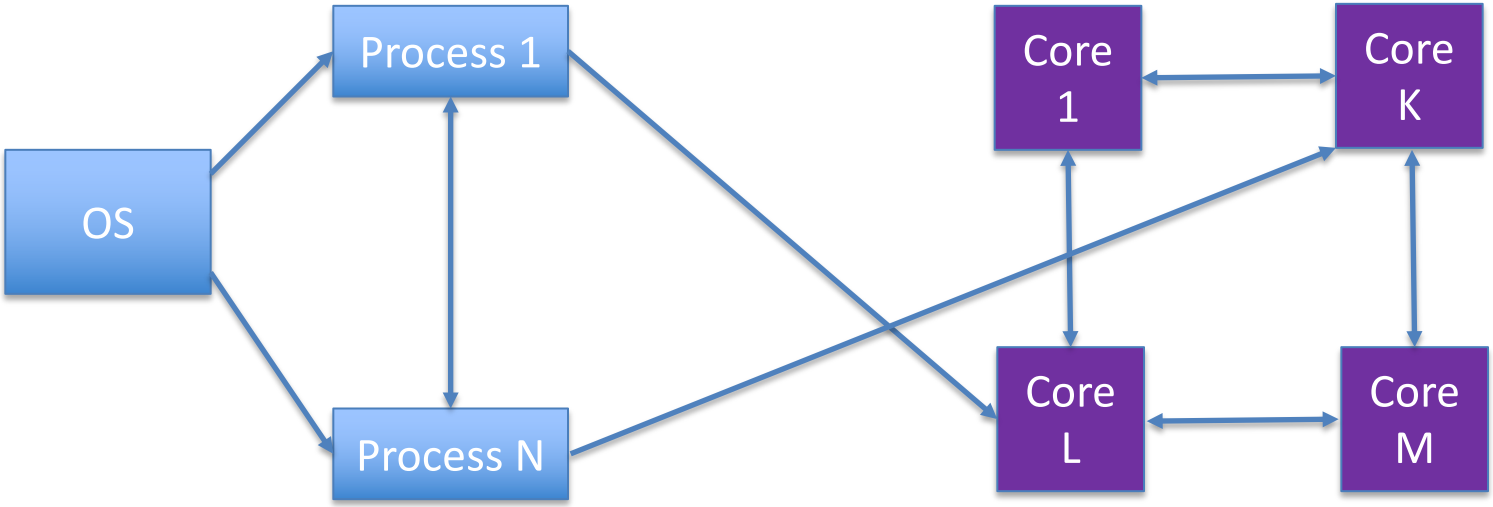
Memory

16GB of total RAM

Up to 484GB/s transfer speed

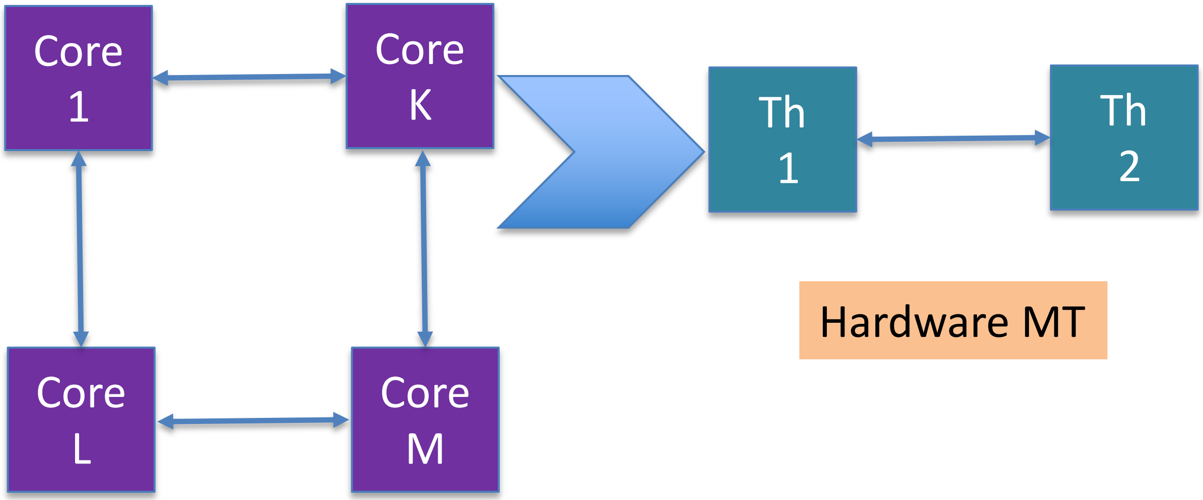
L2+L3 Cache of 9.5MB

Cores



Scheduling

Threads



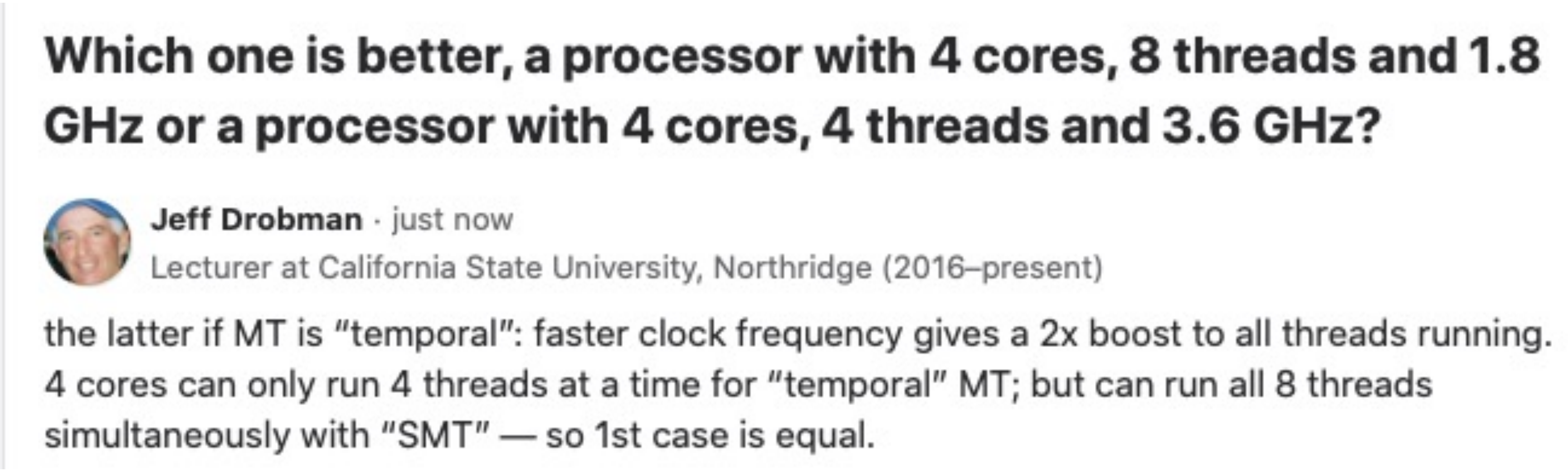
Hardware MT

Assignment

Java

Thread xyz = new Thread( );

Threads



Which one is better, a processor with 4 cores, 8 threads and 1.8 GHz or a processor with 4 cores, 4 threads and 3.6 GHz?

Jeff Drobman · just now
Lecturer at California State University, Northridge (2016–present)

the latter if MT is "temporal": faster clock frequency gives a 2x boost to all threads running. 4 cores can only run 4 threads at a time for "temporal" MT; but can run all 8 threads simultaneously with "SMT" — so 1st case is equal.

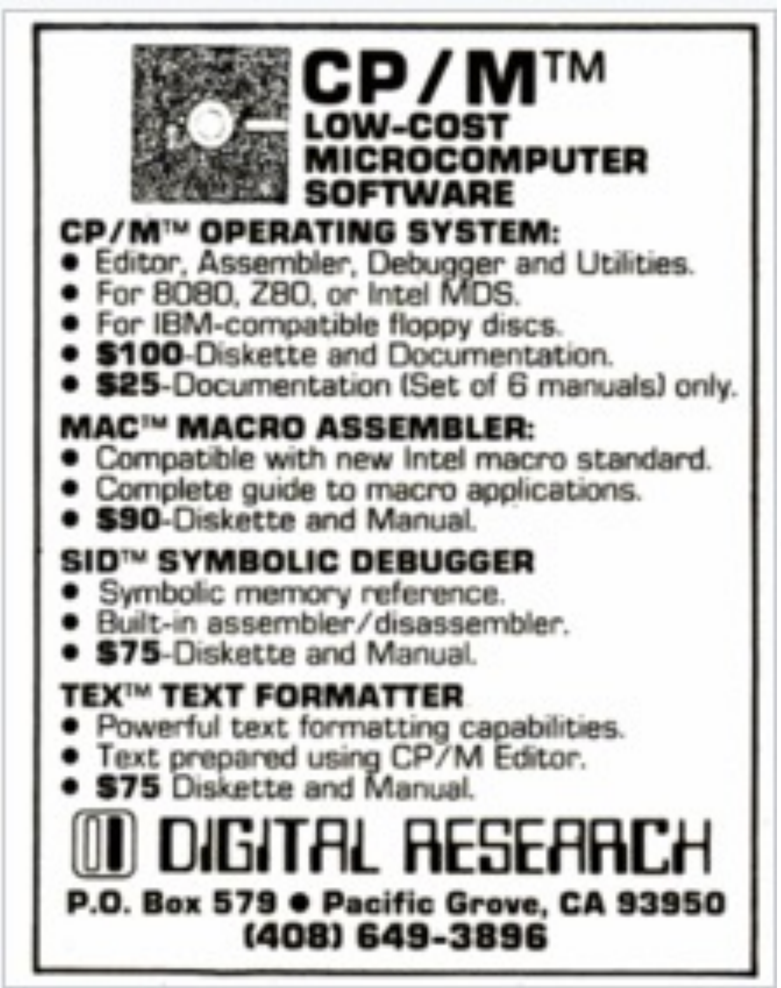
Section

Monitors

Monitors

WIKIPEDIA
The Free Encyclopedia

Control Program/**Monitor**



CP/M™
LOW-COST MICROCOMPUTER SOFTWARE

CP/M™ OPERATING SYSTEM:
- Editor, Assembler, Debugger and Utilities.
- For 8080, Z80, or Intel MDS.
- For IBM-compatible floppy discs.
- **$100**-Diskette and Documentation.
- **$25**-Documentation (Set of 6 manuals) only.

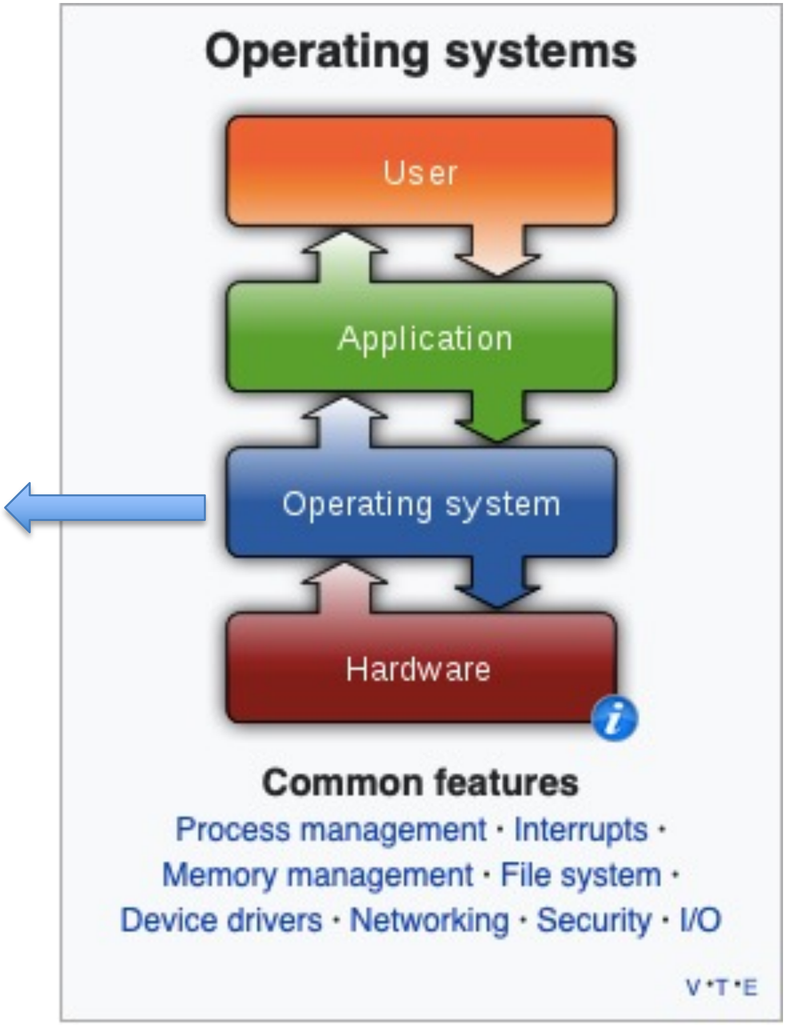
MAC™ MACRO ASSEMBLER:
- Compatible with new Intel macro standard.
- Complete guide to macro applications.
- **$90**-Diskette and Manual.

SID™ SYMBOLIC DEBUGGER
- Symbolic memory reference.
- Built-in assembler/disassembler.
- **$75**-Diskette and Manual.

TEX™ TEXT FORMATTER
- Powerful text formatting capabilities.
- Text prepared using CP/M Editor.
- **$75** Diskette and Manual.

DIGITAL RESEARCH
P.O. Box 579 • Pacific Grove, CA 93950
(408) 649-3896

CP/M advertisement in the 11 December 1978, issue of *InfoWorld* magazine



Operating systems

Common features

Process management · Interrupts · Memory management · File system · Device drivers · Networking · Security · I/O

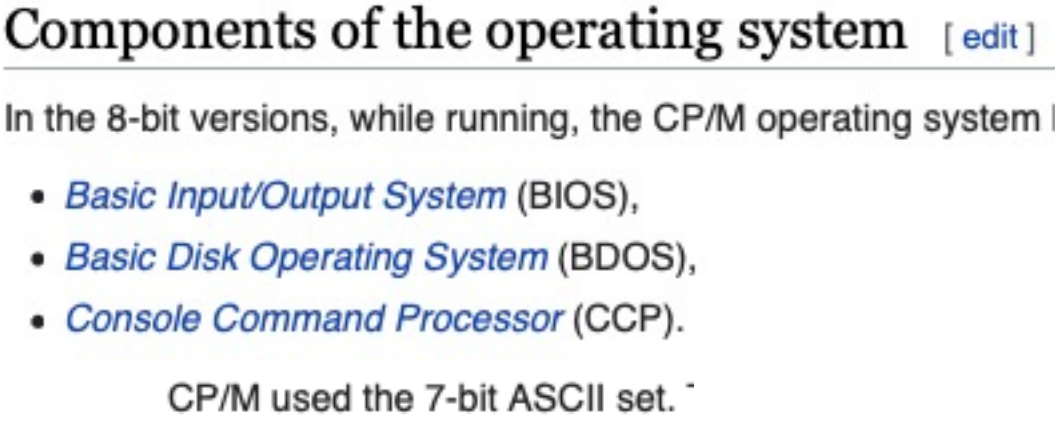
V · T · E

WIKIPEDIA
The Free Encyclopedia

Monitors

Control Program/**Monitor**

CP/M

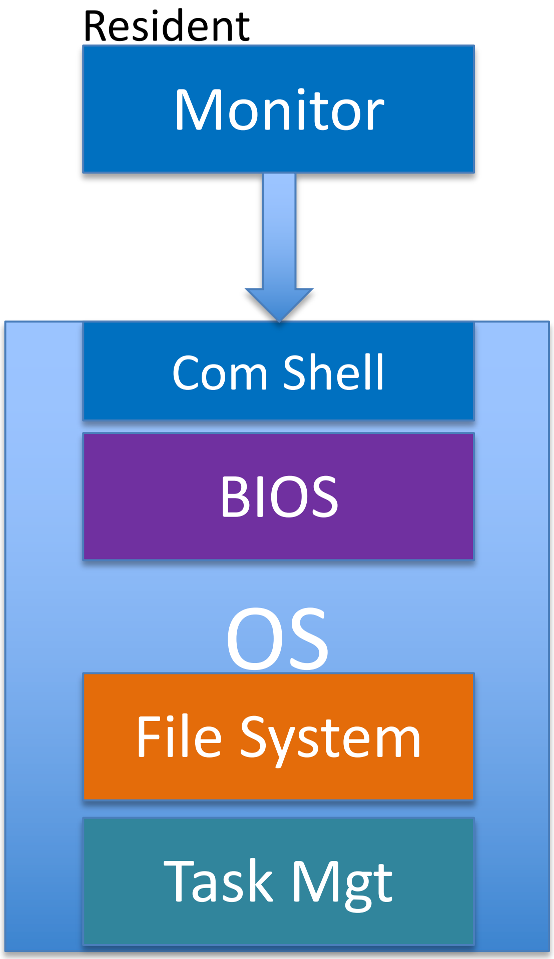


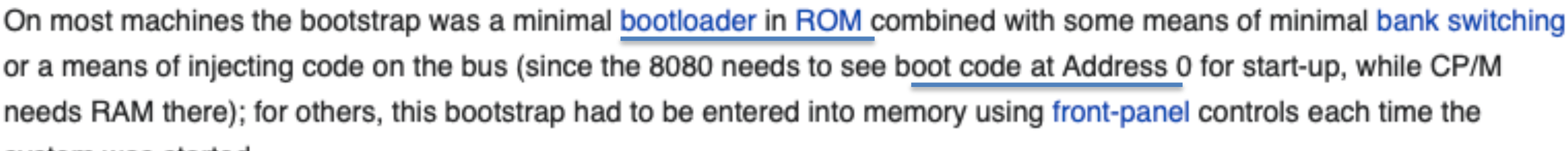
Components of the operating system [edit]

In the 8-bit versions, while running, the CP/M operating system l

- *Basic Input/Output System* (BIOS),
- *Basic Disk Operating System* (BDOS),
- *Console Command Processor* (CCP).

CP/M used the 7-bit ASCII set.

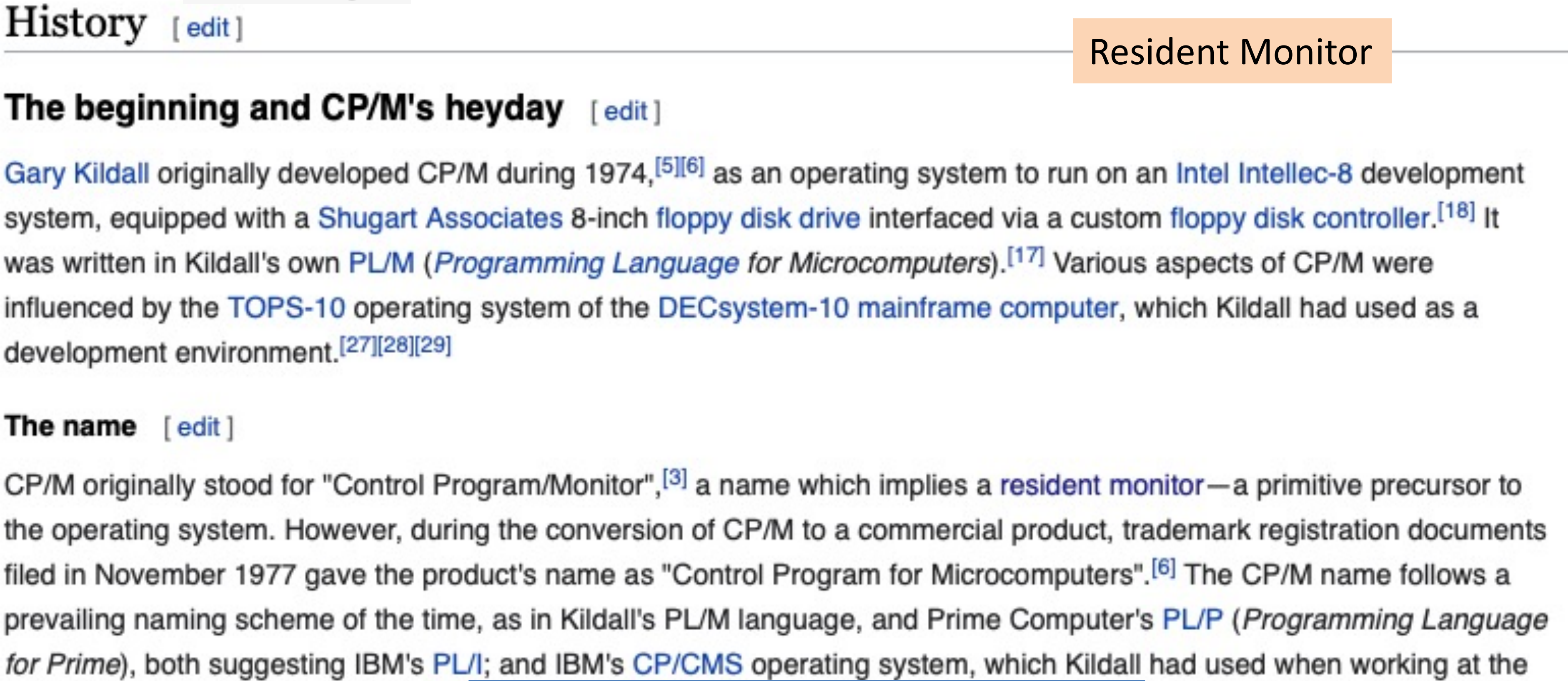




On most machines the bootstrap was a minimal bootloader in ROM combined with some means of minimal bank switching or a means of injecting code on the bus (since the 8080 needs to see boot code at Address 0 for start-up, while CP/M needs RAM there); for others, this bootstrap had to be entered into memory using front-panel controls each time the

Monitors

Resident Monitor



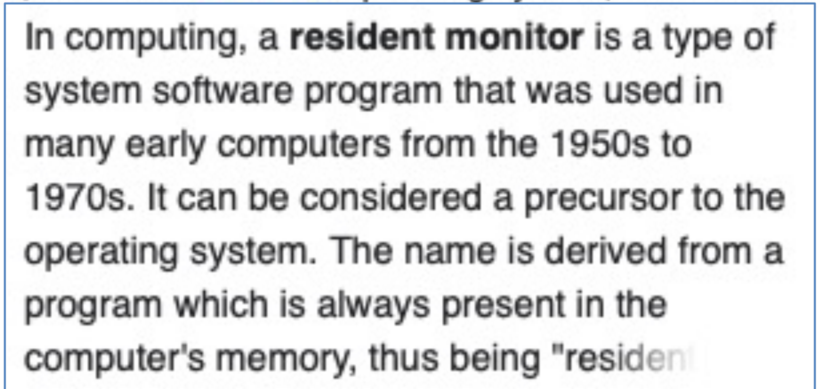
History [edit]

The beginning and CP/M's heyday [edit]

Gary Kildall originally developed CP/M during 1974,[5][6] as an operating system to run on an Intel Intellec-8 development system, equipped with a Shugart Associates 8-inch floppy disk drive interfaced via a custom floppy disk controller.[18] It was written in Kildall's own PL/M (*Programming Language for Microcomputers*).[17] Various aspects of CP/M were influenced by the TOPS-10 operating system of the DECsystem-10 mainframe computer, which Kildall had used as a development environment.[27][28][29]

The name [edit]

CP/M originally stood for "Control Program/Monitor",[3] a name which implies a resident monitor—a primitive precursor to the operating system. However, during the conversion of CP/M to a commercial product, trademark registration documents filed in November 1977 gave the product's name as "Control Program for Microcomputers".[6] The CP/M name follows a prevailing naming scheme of the time, as in Kildall's PL/M language, and Prime Computer's PL/P (*Programming Language for Prime*), both suggesting IBM's PL/I; and IBM's CP/CMS operating system, which Kildall had used when working at the Naval Postgraduate School (NPS).

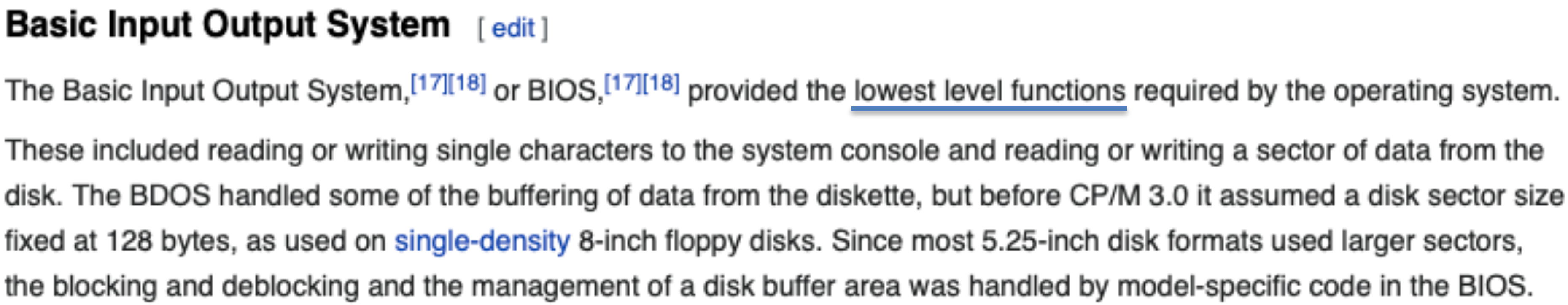


In computing, a **resident monitor** is a type of system software program that was used in many early computers from the 1950s to 1970s. It can be considered a precursor to the operating system. The name is derived from a program which is always present in the computer's memory, thus being "residen

Monitors

Wikipedia
The Free Encyclopedia

BIOS

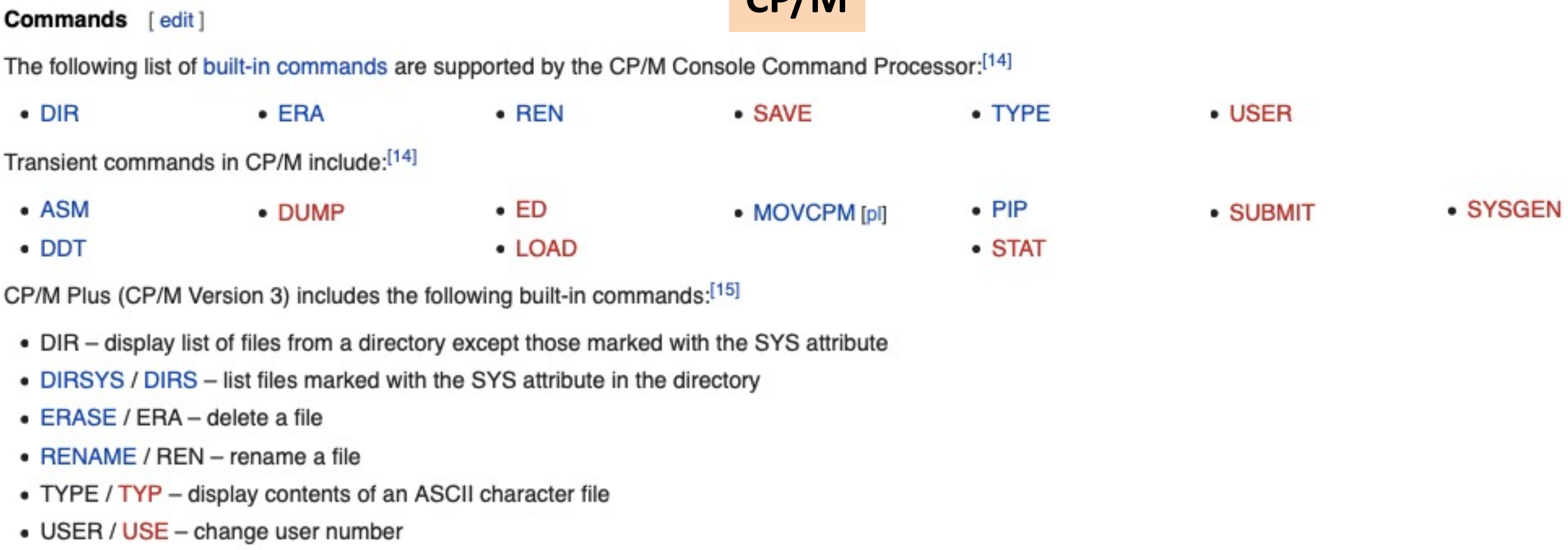


Basic Input Output System [edit]

The Basic Input Output System,[17][18] or BIOS,[17][18] provided the lowest level functions required by the operating system.

These included reading or writing single characters to the system console and reading or writing a sector of data from the disk. The BDOS handled some of the buffering of data from the diskette, but before CP/M 3.0 it assumed a disk sector size fixed at 128 bytes, as used on single-density 8-inch floppy disks. Since most 5.25-inch disk formats used larger sectors, the blocking and deblocking and the management of a disk buffer area was handled by model-specific code in the BIOS.

CP/M



Commands [edit]

The following list of built-in commands are supported by the CP/M Console Command Processor:[14]

- DIR
- ERA
- REN
- SAVE
- TYPE
- USER

Transient commands in CP/M include:[14]

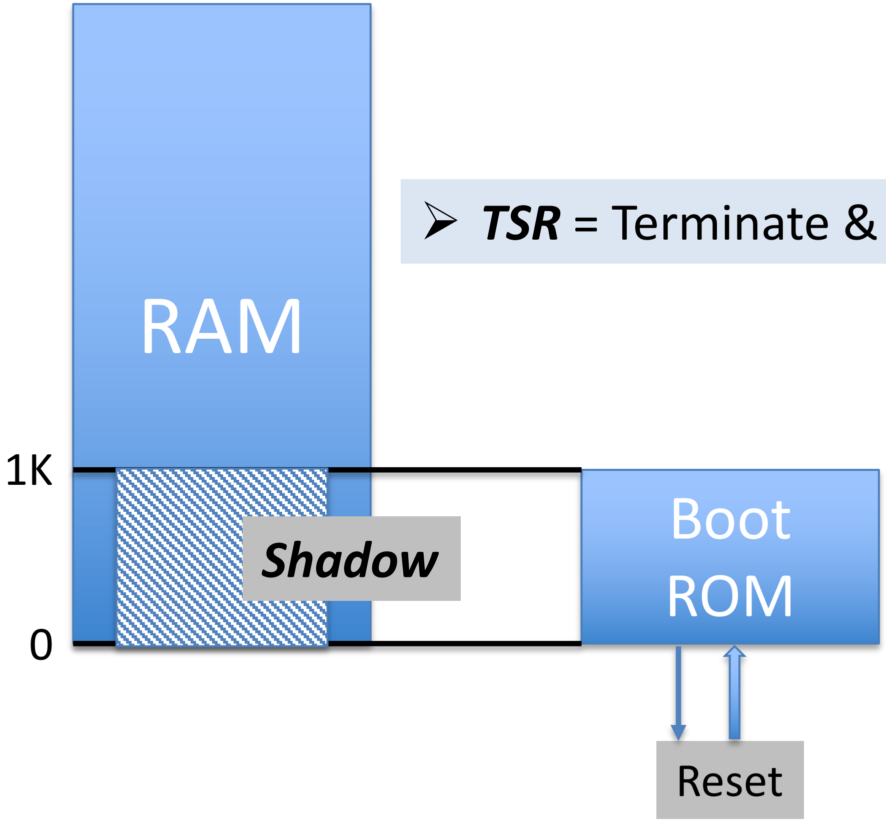
- ASM
- DDT
- DUMP
- ED
- LOAD
- MOVCPM [pl]
- PIP
- STAT
- SUBMIT
- SYSGEN

CP/M Plus (CP/M Version 3) includes the following built-in commands:[15]

- DIR – display list of files from a directory except those marked with the SYS attribute
- DIRSYS / DIRS – list files marked with the SYS attribute in the directory
- ERASE / ERA – delete a file
- RENAME / REN – rename a file
- TYPE / TYP – display contents of an ASCII character file
- USER / USE – change user number

Shadow Boot ROM

Used on ***small memory*** embedded systems



- ***TSR*** = Terminate & Stay Resident

Advanced Material

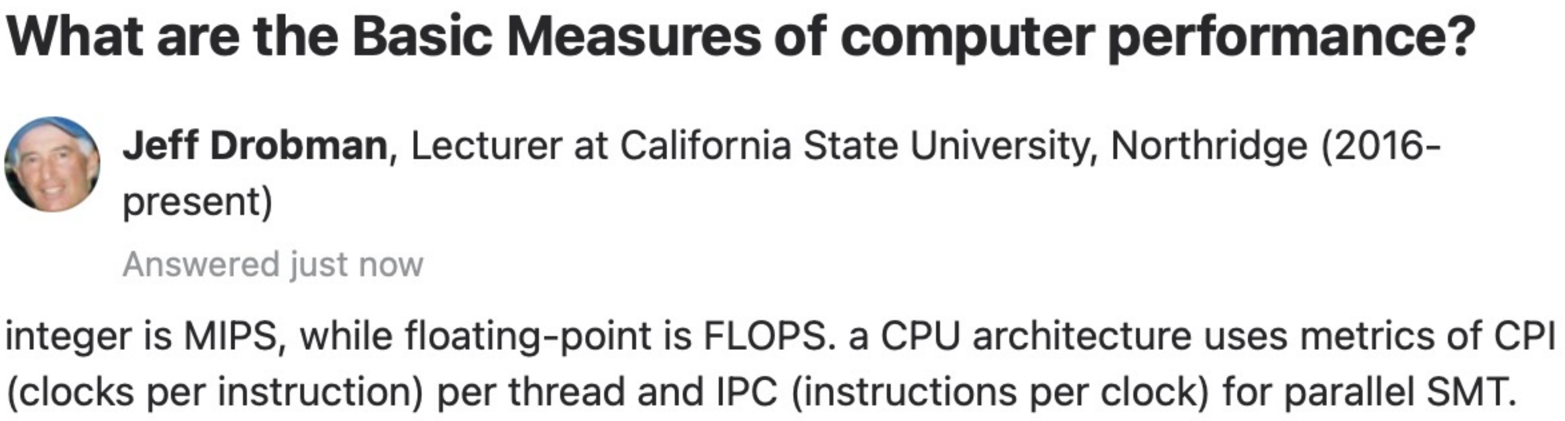
CPU
Performance

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

Quora

MIPS/FLOPS

CPI/IPC

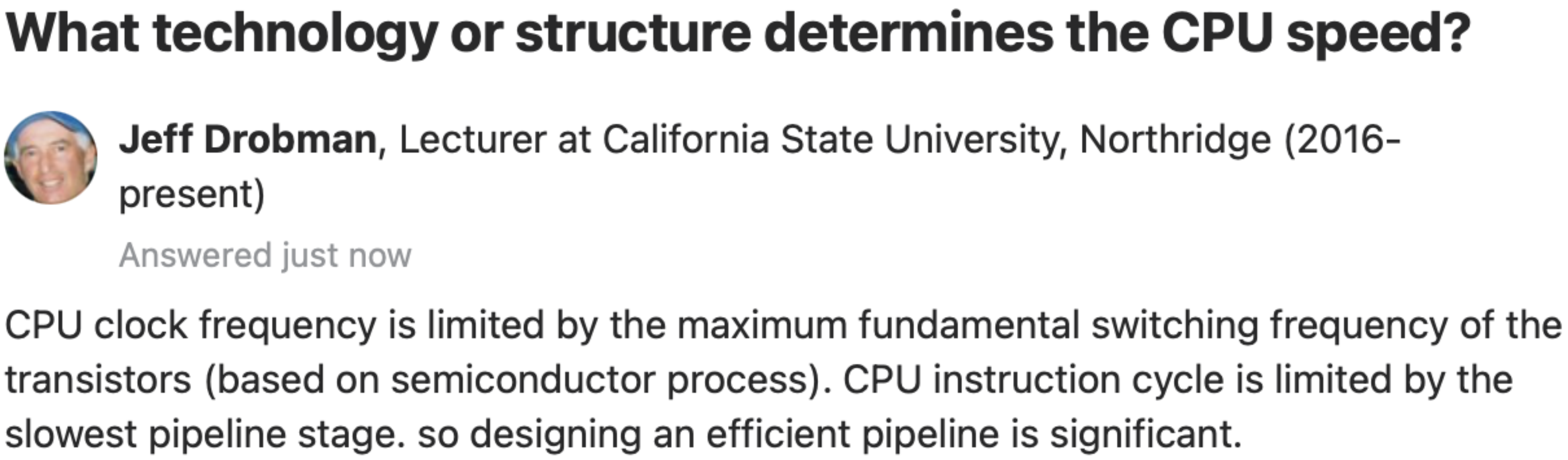


What are the Basic Measures of computer performance?

Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

Answered just now

integer is MIPS, while floating-point is FLOPS. a CPU architecture uses metrics of CPI (clocks per instruction) per thread and IPC (instructions per clock) for parallel SMT.



What technology or structure determines the CPU speed?

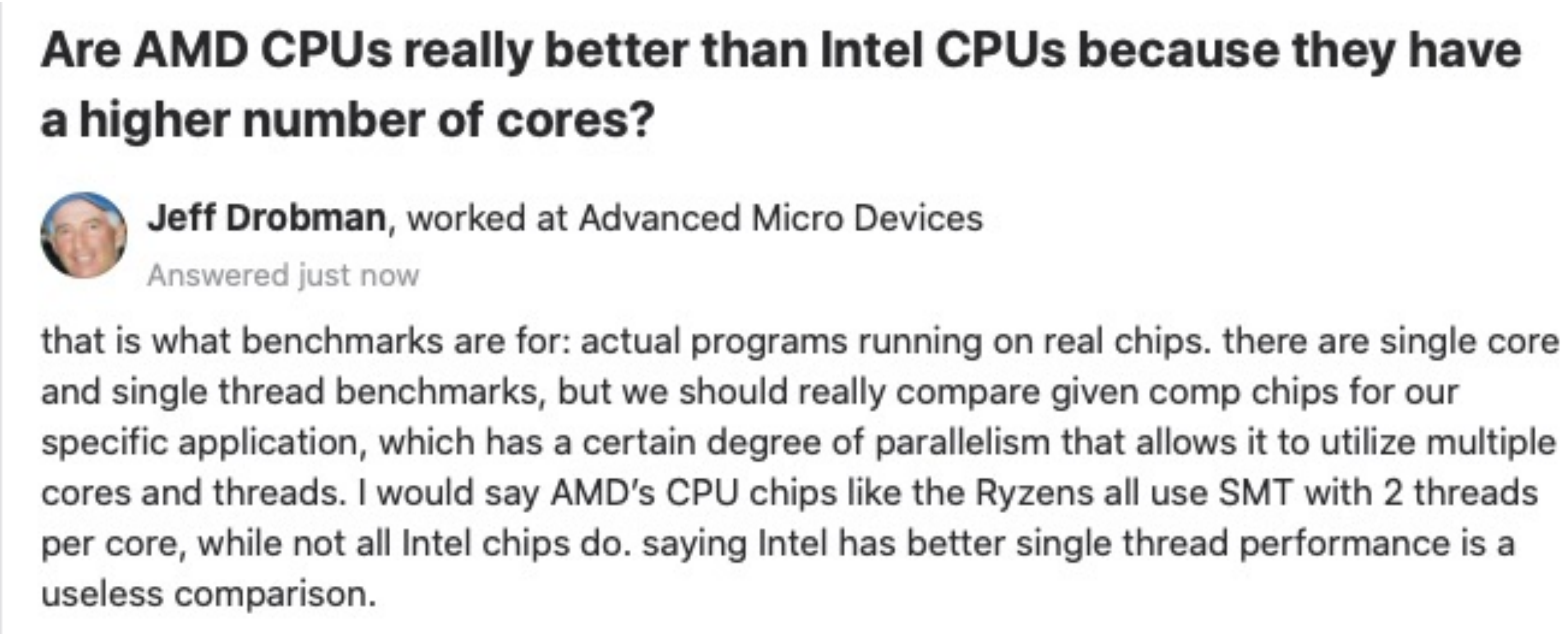
Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

Answered just now

CPU clock frequency is limited by the maximum fundamental switching frequency of the transistors (based on semiconductor process). CPU instruction cycle is limited by the slowest pipeline stage. so designing an efficient pipeline is significant.

AMD vs. Intel CPU's

Cores & Threads



Are AMD CPUs really better than Intel CPUs because they have a higher number of cores?

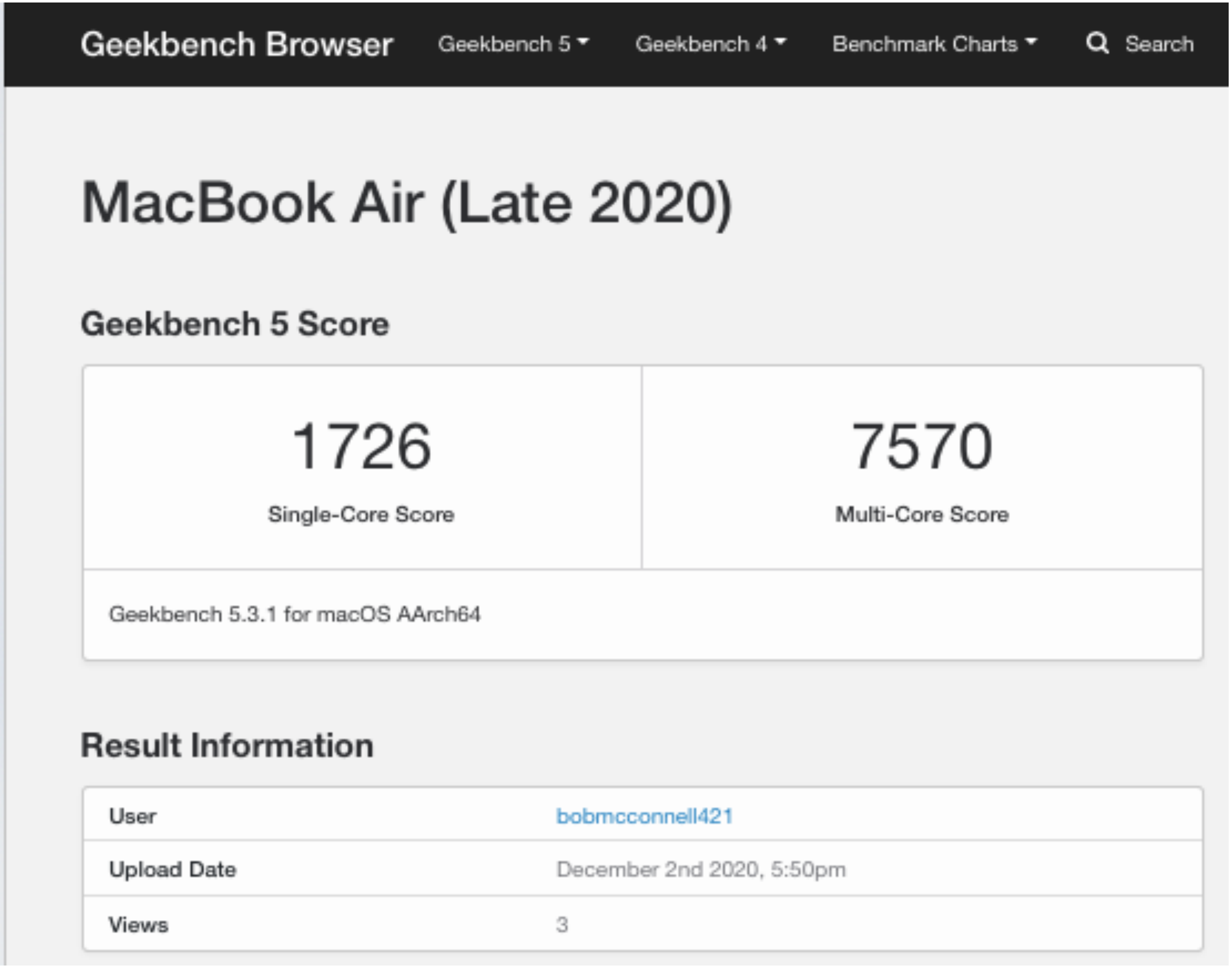
Jeff Drobman, worked at Advanced Micro Devices

Answered just now

that is what benchmarks are for: actual programs running on real chips. there are single core and single thread benchmarks, but we should really compare given comp chips for our specific application, which has a certain degree of parallelism that allows it to utilize multiple cores and threads. I would say AMD's CPU chips like the Ryzens all use SMT with 2 threads per core, while not all Intel chips do. saying Intel has better single thread performance is a useless comparison.

- ❖ Benchmarks
  - ❑ Single Core/Thread
  - ❑ Multi-Core

Geekbench: New MacBook



MacBook Air (Late 2020)

Geekbench 5 Score

1726	7570
Single-Core Score	Multi-Core Score

Geekbench 5.3.1 for macOS AArch64

Result Information

User	bobmcconnell421
Upload Date	December 2nd 2020, 5:50pm
Views	3

CPU Performance

$$\text{Time} = \text{Seconds}/\text{Program} = \frac{\cancel{\text{Instructions}}}{\text{Program}} \times \frac{\cancel{\text{Clock cycles}}}{\cancel{\text{Instruction}}} \times \frac{\text{Seconds}}{\cancel{\text{Clock cycle}}}$$

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

Clock rate = 1/Clock cycle time

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

CPU Performance

CPU performance

- ***faster*** – *Moore's Law* has guided the shrinkage of transistors which has an attendant increase in frequency. we have seen CPU clock frequency go from 1GHz to up to 5GHz (but mostly 2-3 GHz) today.
- ***performance*** of CPU's is measured in "throughput" = clock period (1/freq) * instructions per cycle per (1/CPI) * number of cores (or execution units, i.e., pipelines), or **Perf = N / (f * CPI)**
- we have seen minor (~5-10%) improvements in CPI over the past 10 years, mostly due to hardware assisted out-of-order and speculative execution. number of ***threads*** has gone up, but that also has a limited effect beyond a 2nd thread per core.

> - we have essentially reached the end of the line for *single-core* CPU architecture improvements. Processor *frequencies* have been topping out along with the end of Moore's Law transistor shrinkage.
> - so we are now seeing merely more cores, of both CPU and GPU types – which provide greater opportunity for *parallelism*, which may or may not be usable.

CPU Performance Factors

❖ factors determining

max **instruction cycle** *frequency*

or minimum cycle *time* per *core* or *pipeline* (if *superscalar*):

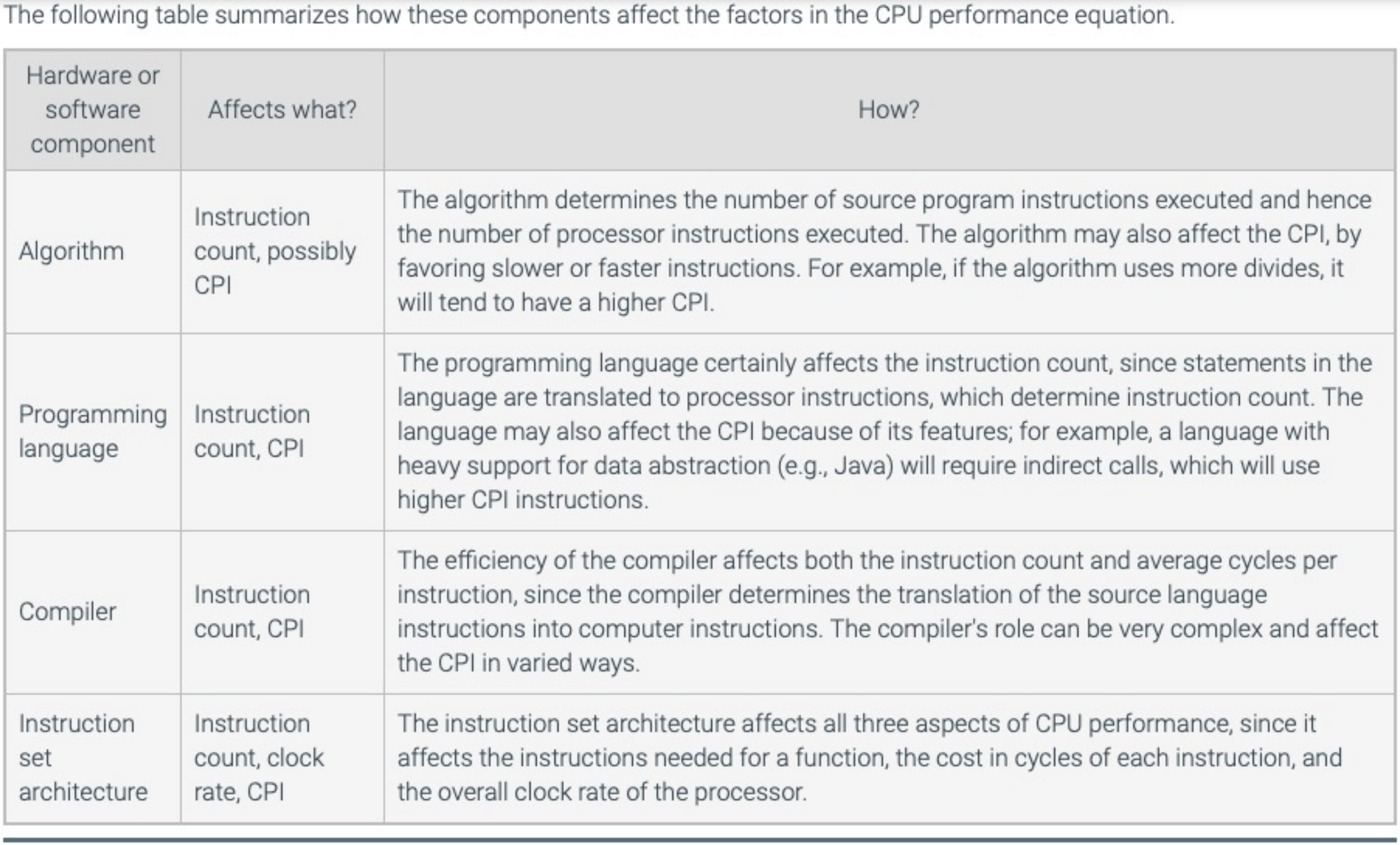
1. **transistor switching** frequency (inverse function of feature sizes, e.g., 7-10 nm)
2. single vs. double **clock phases**
3. instruction **cycle times**: determined by slowest pipeline stage
4. gating pipeline stage = longest **logic gate path** in the "ICU" state machine

CPU Performance Factors

❖ **performance issues** that make the CPU run slower than the max frequency:

1. **cache performance**: miss rates, refill rates, line sizes, coherence and locality of reference for both instructions and data
2. **context switch** rates: events such as system calls, traps, exceptions and interrupts.
3. **branch prediction** performance coupled with **branch rates**
4. number of **cores**
5. number of **threads** per core
6. rate of **multi-cycle instructions** such as integer and floating-point multiply, divide, other floating-point operations such as add, subtract.
7. degree of extractable (usable) **parallelism** in the given code determines effective utilization of cores, threads and co-processors (e.g., floating-point units)

CPU Performance



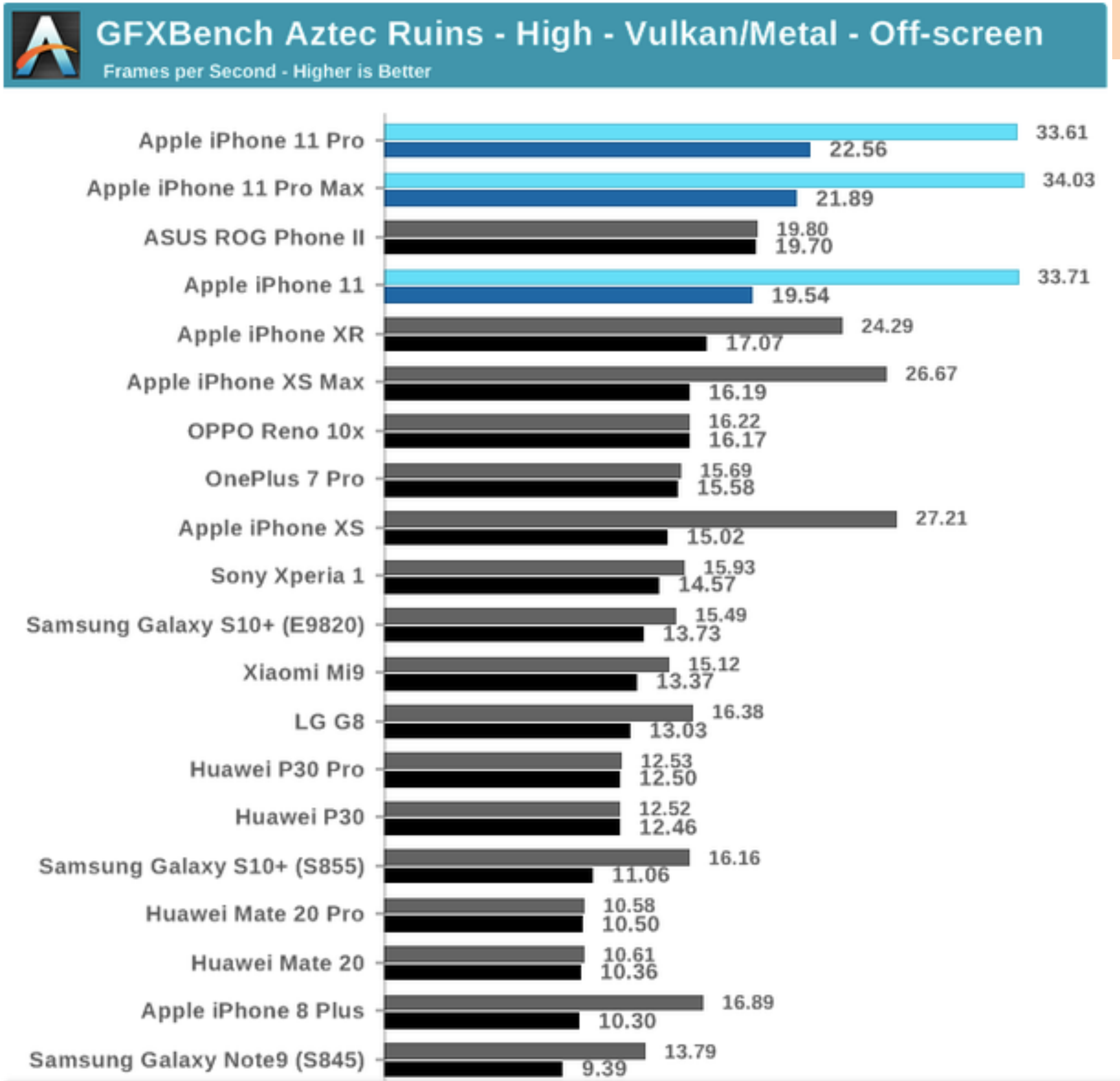
The following table summarizes how these components affect the factors in the CPU performance equation.

Hardware or software component	Affects what?	How?
Algorithm	Instruction count, possibly CPI	The algorithm determines the number of source program instructions executed and hence the number of processor instructions executed. The algorithm may also affect the CPI, by favoring slower or faster instructions. For example, if the algorithm uses more divides, it will tend to have a higher CPI.
Programming language	Instruction count, CPI	The programming language certainly affects the instruction count, since statements in the language are translated to processor instructions, which determine instruction count. The language may also affect the CPI because of its features; for example, a language with heavy support for data abstraction (e.g., Java) will require indirect calls, which will use higher CPI instructions.
Compiler	Instruction count, CPI	The efficiency of the compiler affects both the instruction count and average cycles per instruction, since the compiler determines the translation of the source language instructions into computer instructions. The compiler's role can be very complex and affect the CPI in varied ways.
Instruction set architecture	Instruction count, clock rate, CPI	The instruction set architecture affects all three aspects of CPU performance, since it affects the instructions needed for a function, the cost in cycles of each instruction, and the overall clock rate of the processor.

CPU Performance

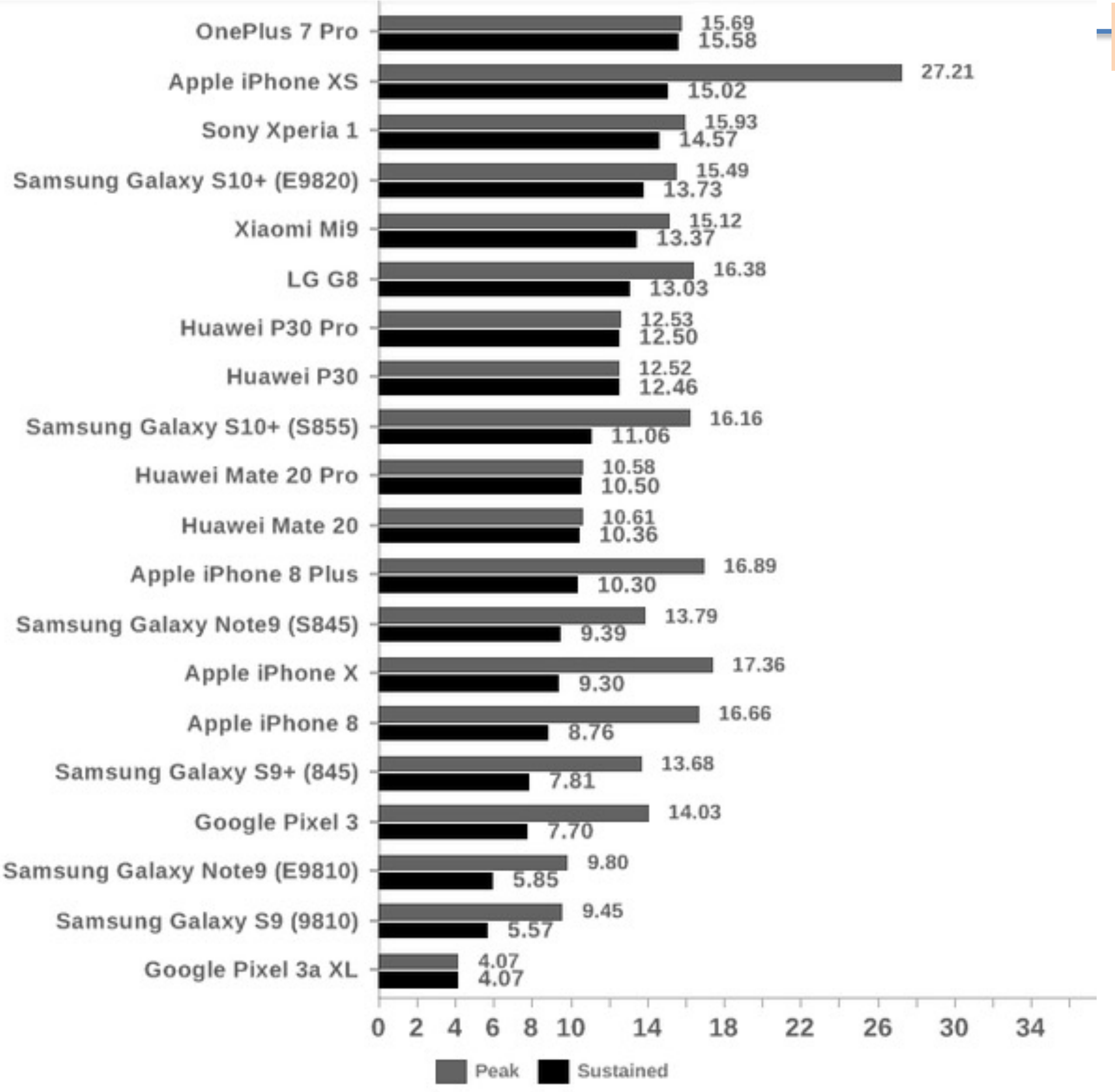
DR JEFF SOFTWARE
INDIE APP DEVELOPER

Frames/sec

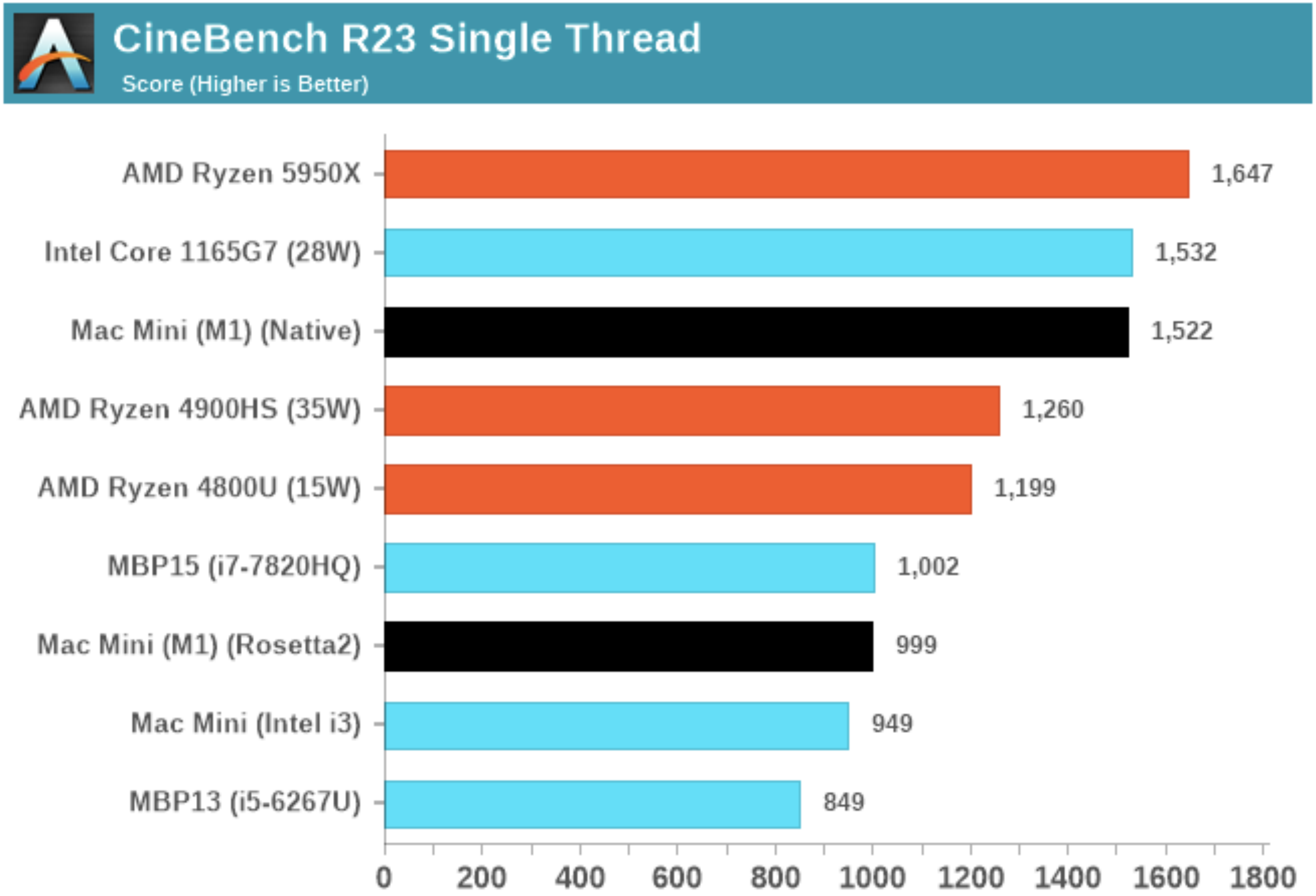


CPU Performance

Frames/sec



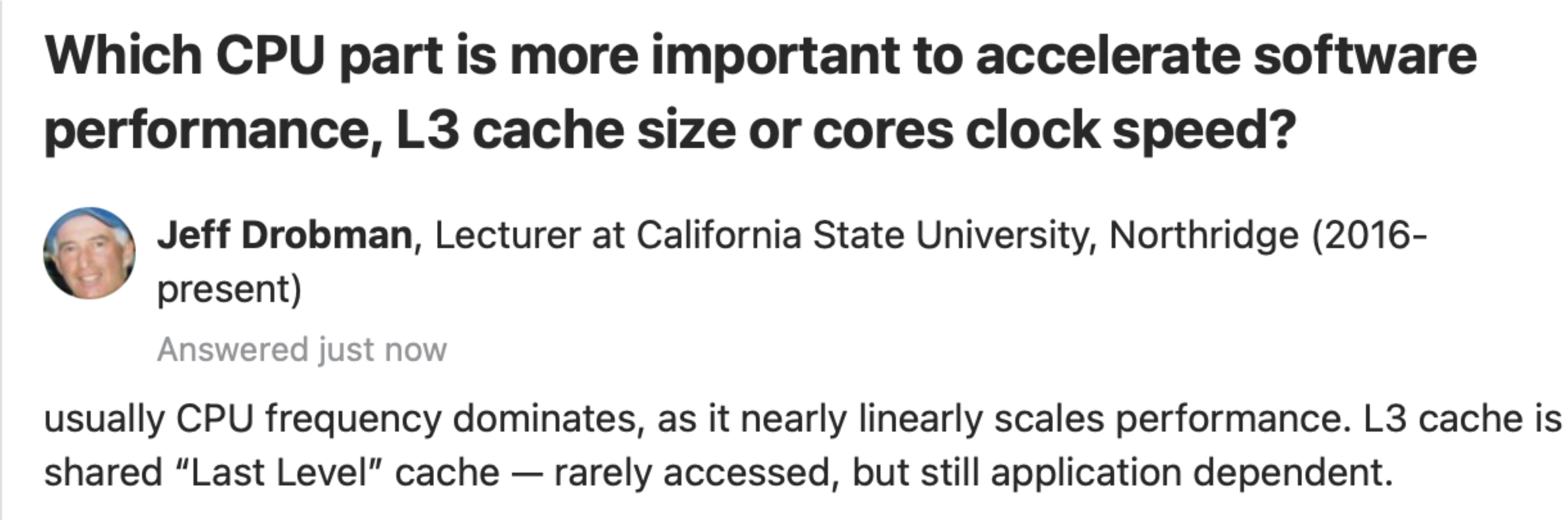
Benchmarks



The AMD Ryzen 5950X is clocked much higher. The M1 runs at 3.2 GHz. The Ryzen runs at 3.4 to 4.9 GHz. Let us use this to compute performance per GHz. For the M1 we get:

Cores vs Cache

Quora



Which CPU part is more important to accelerate software performance, L3 cache size or cores clock speed?

Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

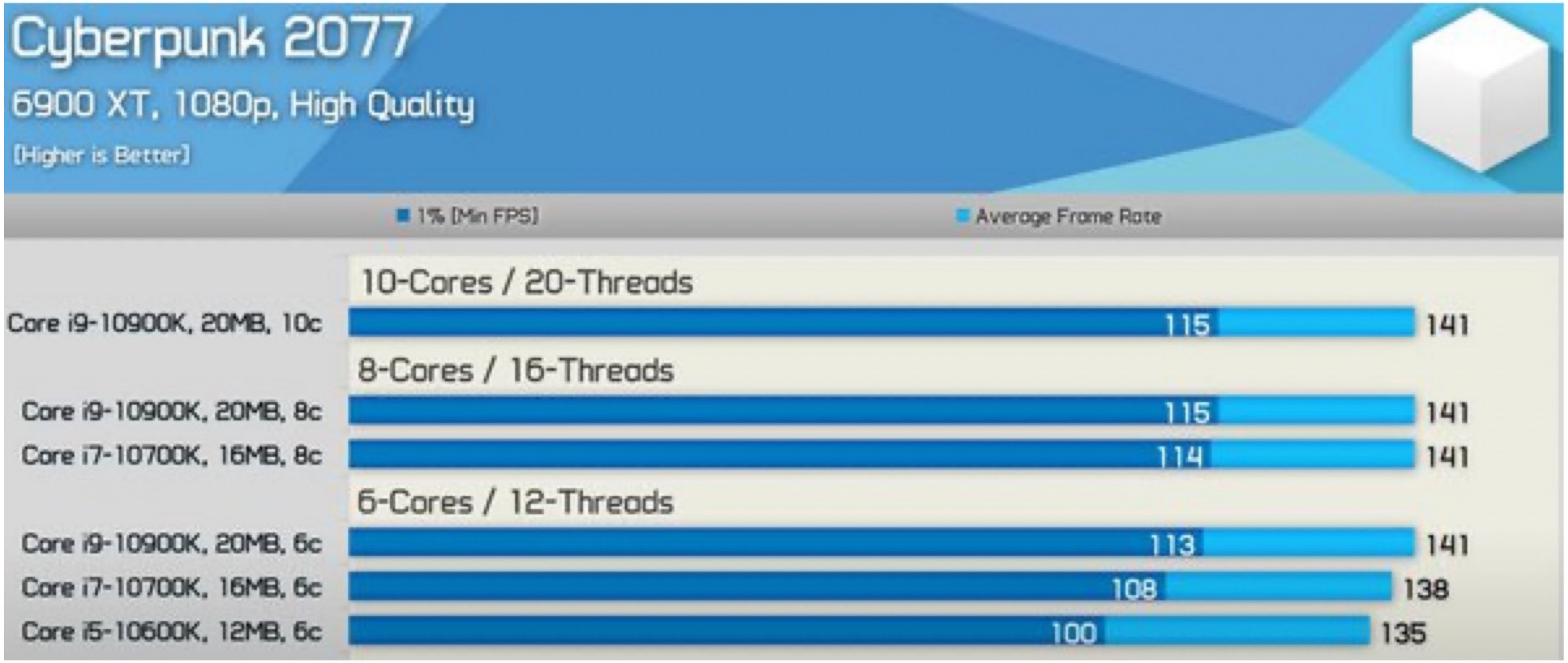
Answered just now

usually CPU frequency dominates, as it nearly linearly scales performance. L3 cache is shared “Last Level” cache — rarely accessed, but still application dependent.

Cores vs Cache

Quora

Some applications need a lot of data and are thus cache-bound, meaning that they will perform best with lots of cache as fetching data from DRAM wastes clock cycles, energy and is slower than SRAM. Others will favour raw single threaded performance and high IPC, this is particularly true for games which scale best with 6–8 high performance cores.



In the above test the Core i9–10900K was setup with 6 and 8 cores alongside the default 10C configuration. The performance from cache scaling pretty much stops

Advanced Material

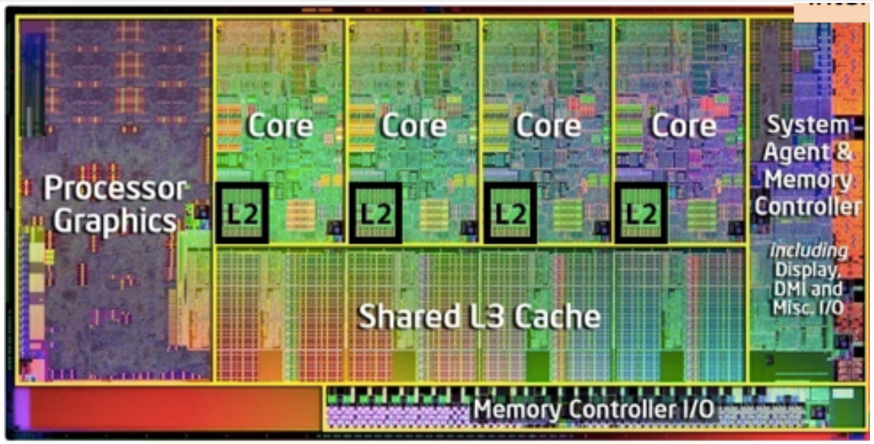
CPU
Parallelism

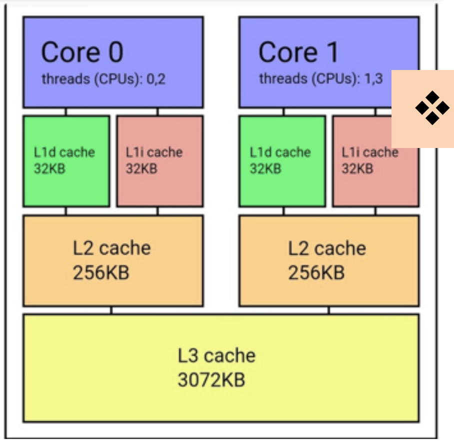
3 Levels of CPU *Architecture*

COMP122/222

❖ Macro

System= *Multi-core SoC*





❖ Floorplan

COMP122

❖ Org (ISA)

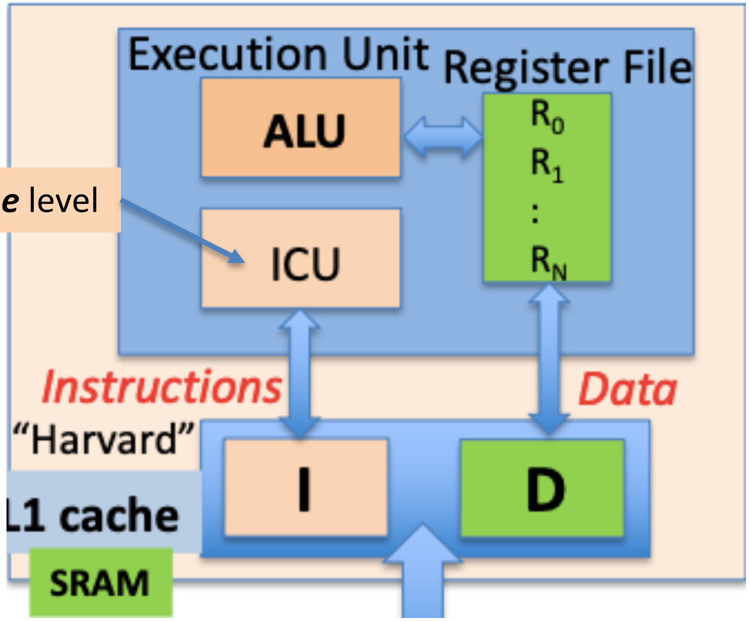
CPU Core internals

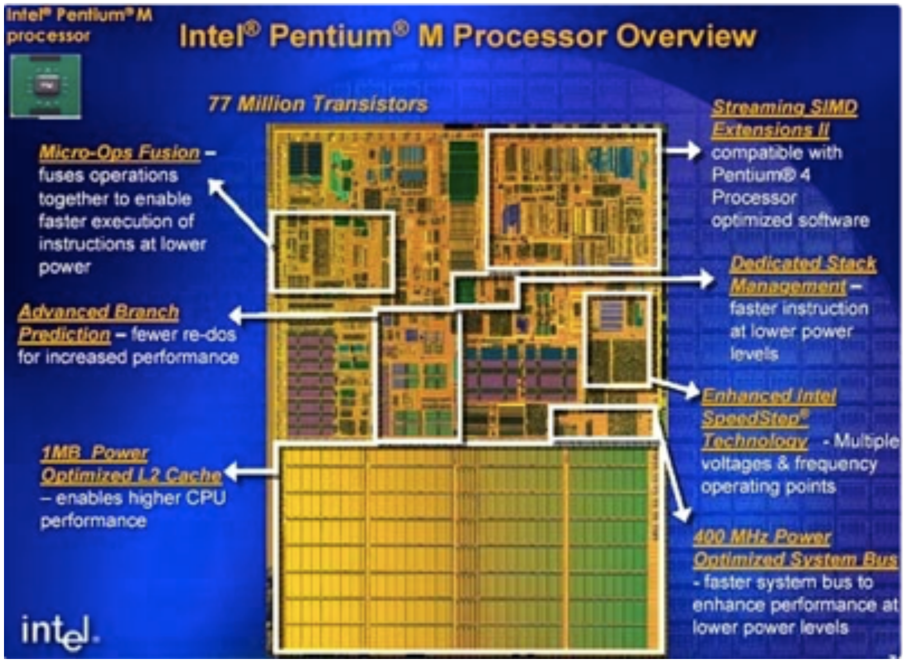
COMP222

❖ Micro

Pipeline level

Pipeline level





Intel® Pentium® M processor

Intel® Pentium® M Processor Overview

77 Million Transistors

Micro-Ops Fusion – fuses operations together to enable faster execution of instructions at lower power

Advanced Branch Prediction – fewer re-dos for increased performance

1MB Power Optimized L2 Cache – enables higher CPU performance

Streaming SIMD Extensions II – compatible with Pentium® 4 Processor optimized software

Dedicated Stack Management – faster instruction at lower power levels

Enhanced Intel SpeedStep® Technology – Multiple voltages & frequency operating points

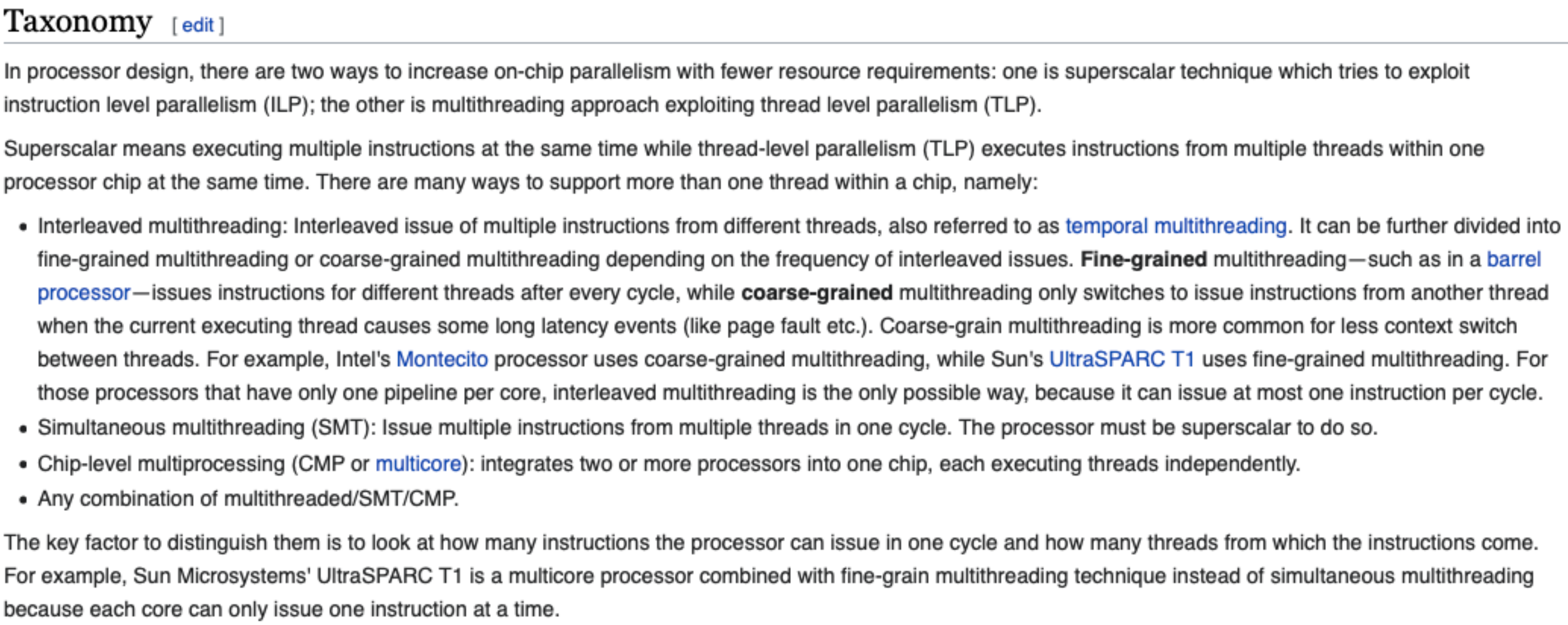
400 MHz Power Optimized System Bus – faster system bus to enhance performance at lower power levels

intel.

ILP/TLP: MT/SMT

Wiki

Instructions vs. Threads



Taxonomy [edit]

In processor design, there are two ways to increase on-chip parallelism with fewer resource requirements: one is superscalar technique which tries to exploit instruction level parallelism (ILP); the other is multithreading approach exploiting thread level parallelism (TLP).

Superscalar means executing multiple instructions at the same time while thread-level parallelism (TLP) executes instructions from multiple threads within one processor chip at the same time. There are many ways to support more than one thread within a chip, namely:

- Interleaved multithreading: Interleaved issue of multiple instructions from different threads, also referred to as temporal multithreading. It can be further divided into fine-grained multithreading or coarse-grained multithreading depending on the frequency of interleaved issues. **Fine-grained** multithreading—such as in a barrel processor—issues instructions for different threads after every cycle, while **coarse-grained** multithreading only switches to issue instructions from another thread when the current executing thread causes some long latency events (like page fault etc.). Coarse-grain multithreading is more common for less context switch between threads. For example, Intel's Montecito processor uses coarse-grained multithreading, while Sun's UltraSPARC T1 uses fine-grained multithreading. For those processors that have only one pipeline per core, interleaved multithreading is the only possible way, because it can issue at most one instruction per cycle.
- Simultaneous multithreading (SMT): Issue multiple instructions from multiple threads in one cycle. The processor must be superscalar to do so.
- Chip-level multiprocessing (CMP or multicore): integrates two or more processors into one chip, each executing threads independently.
- Any combination of multithreaded/SMT/CMP.

The key factor to distinguish them is to look at how many instructions the processor can issue in one cycle and how many threads from which the instructions come. For example, Sun Microsystems' UltraSPARC T1 is a multicore processor combined with fine-grain multithreading technique instead of simultaneous multithreading because each core can only issue one instruction at a time.

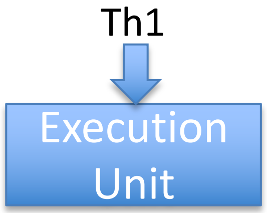
CSUN CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP122

Instruction Level *Parallelism*

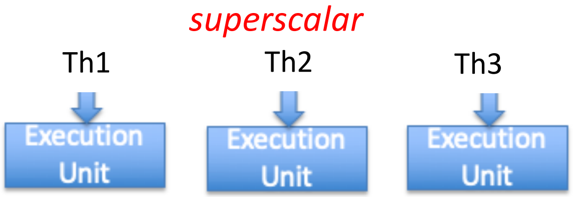
❖ Super- *Pipelining* SISD

- Split some pipeline stages (4-5 → 8-11)
- Faster clock cycle → higher *throughput* (*mips*)
- Affect CPI?



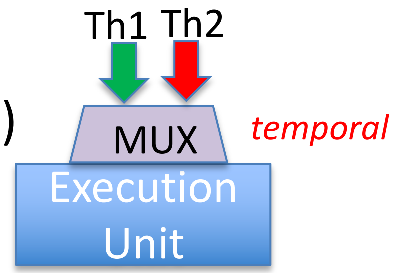
❖ Super- *Scalar* SISD

- Multiple ***Execution Units*** (multi-issue pipelines)
  - each EU = ICU+ALU, with shared GR's
- Hardware + compiler *schedules* instruction streams



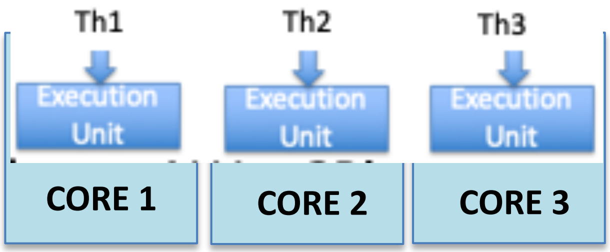
❖ Multi- *Threading* SISD

- Multiple control threads (usually 2, same/dif program)
- Programs can *allocate* code to threads
- Automatic *scheduling* of control threads
- 2 types: *SMT/superscalar* or *temporal* (interleaved: coarse/fine)

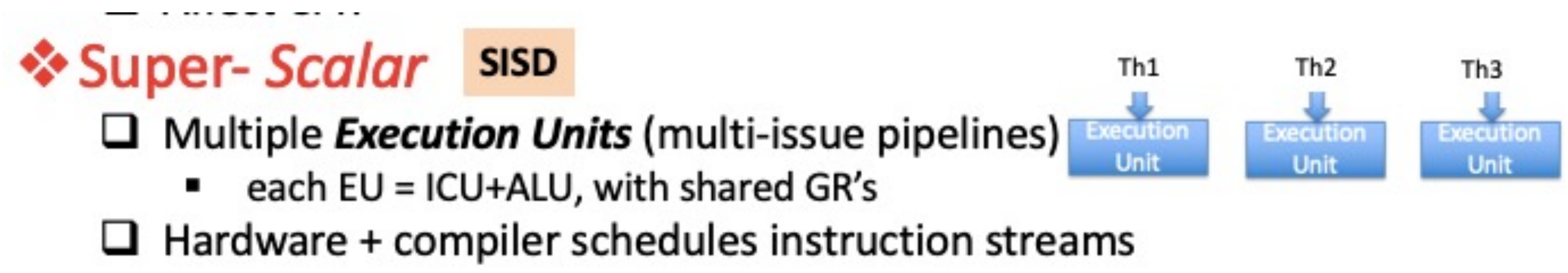


❖ Multi- *Core* MISD

- Classic Parallelism: multiple copies of the CPU
- **Multiple L1/L2 caches** (one set per core)



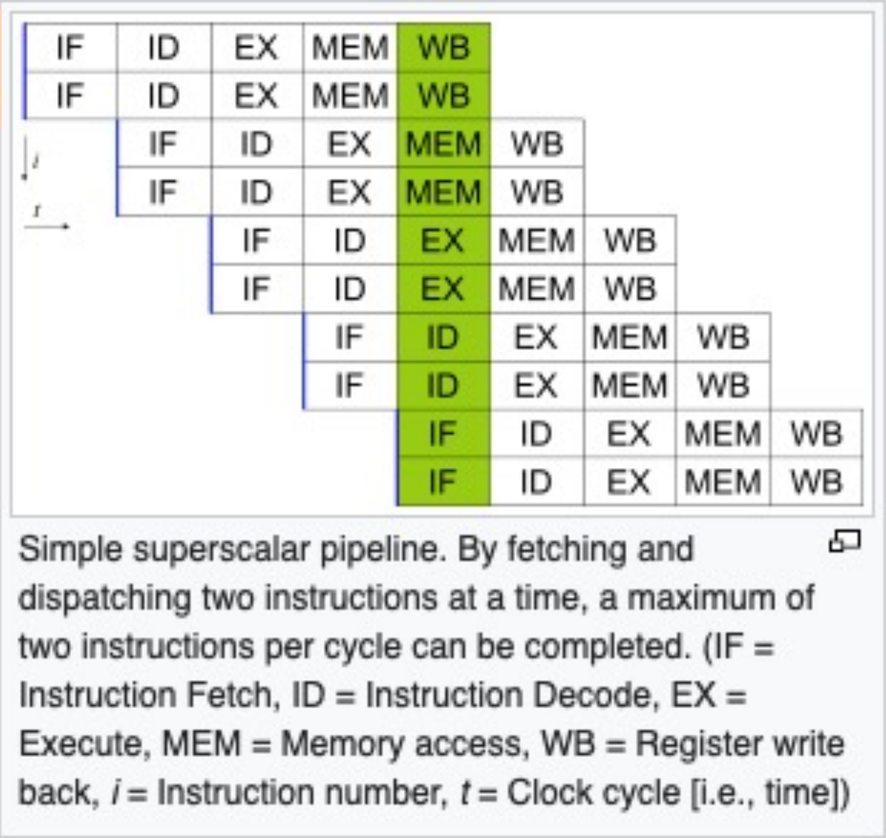
Superscalar



❖ Super- Scalar SISD

- ❑ Multiple ***Execution Units*** (multi-issue pipelines)
  - ▪ each EU = ICU+ALU, with shared GR's
- ❑ Hardware + compiler schedules instruction streams

Th1	EU1	pipeline1
Th2	EU2	pipeline2

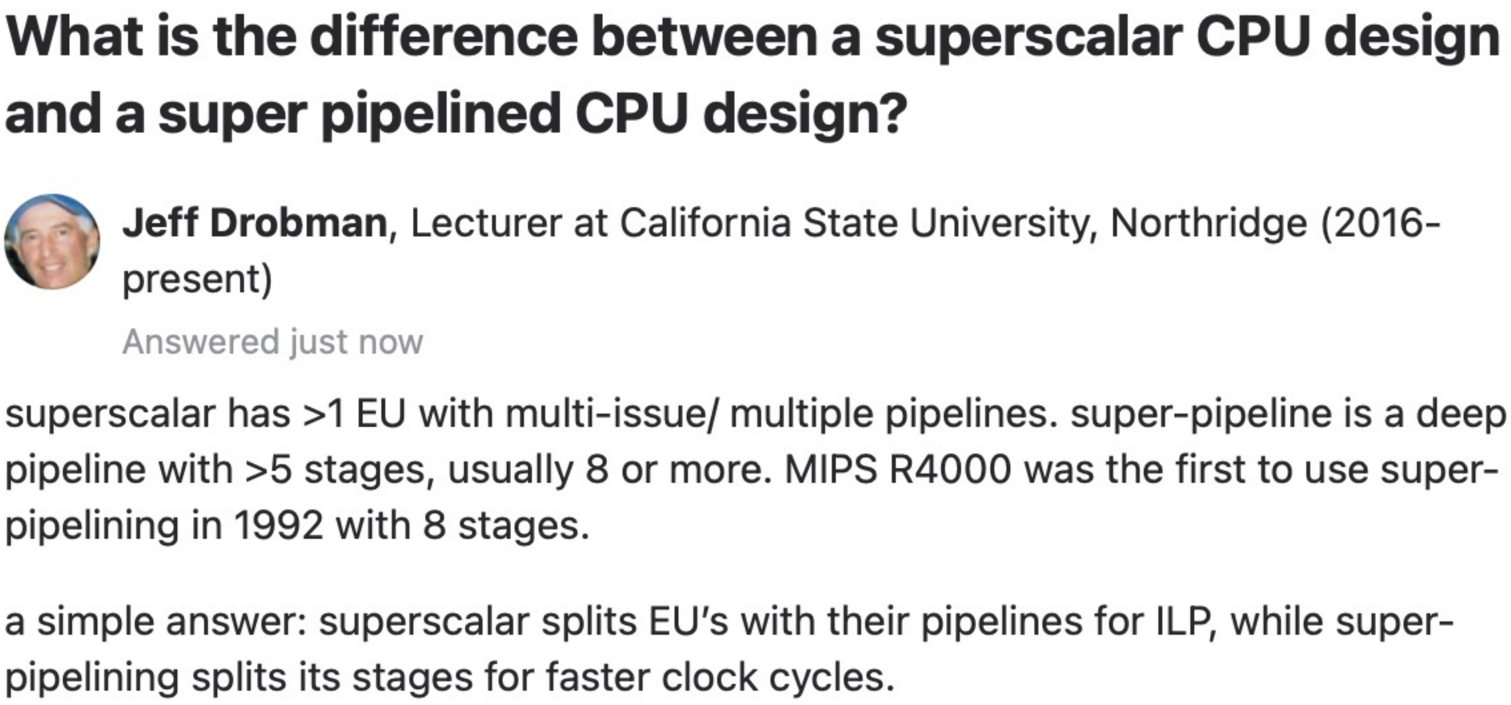


IF	ID	EX	MEM	WB				
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB
				IF	ID	EX	MEM	WB

Simple superscalar pipeline. By fetching and dispatching two instructions at a time, a maximum of two instructions per cycle can be completed. (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back, *i* = Instruction number, *t* = Clock cycle [i.e., time])

Superscalar

-Quora



What is the difference between a superscalar CPU design and a super pipelined CPU design?

Jeff Drobman, Lecturer at California State University, Northridge (2016-present)

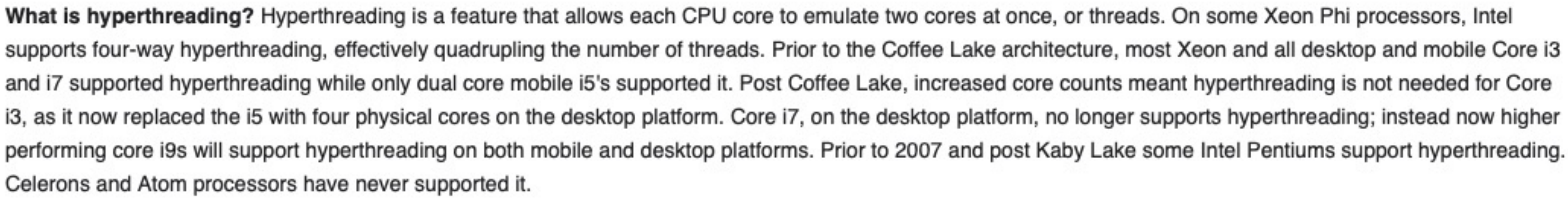
Answered just now

superscalar has >1 EU with multi-issue/ multiple pipelines. super-pipeline is a deep pipeline with >5 stages, usually 8 or more. MIPS R4000 was the first to use super-pipelining in 1992 with 8 stages.

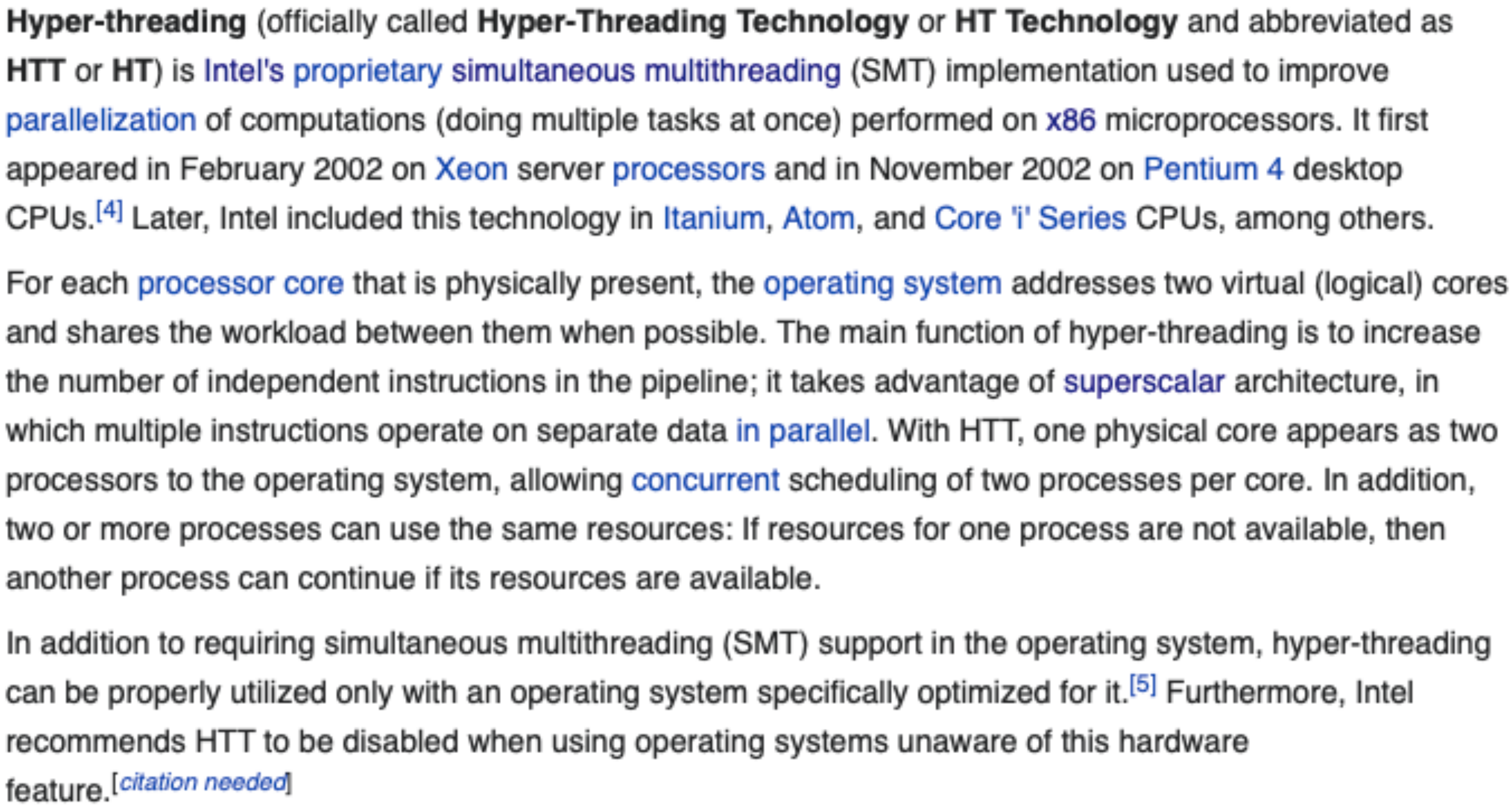
a simple answer: superscalar splits EU's with their pipelines for ILP, while super-pipelining splits its stages for faster clock cycles.

Hyperthreading

WIKIPEDIA
The Free Encyclopedia



What is hyperthreading? Hyperthreading is a feature that allows each CPU core to emulate two cores at once, or threads. On some Xeon Phi processors, Intel supports four-way hyperthreading, effectively quadrupling the number of threads. Prior to the Coffee Lake architecture, most Xeon and all desktop and mobile Core i3 and i7 supported hyperthreading while only dual core mobile i5's supported it. Post Coffee Lake, increased core counts meant hyperthreading is not needed for Core i3, as it now replaced the i5 with four physical cores on the desktop platform. Core i7, on the desktop platform, no longer supports hyperthreading; instead now higher performing core i9s will support hyperthreading on both mobile and desktop platforms. Prior to 2007 and post Kaby Lake some Intel Pentiums support hyperthreading. Celerons and Atom processors have never supported it.



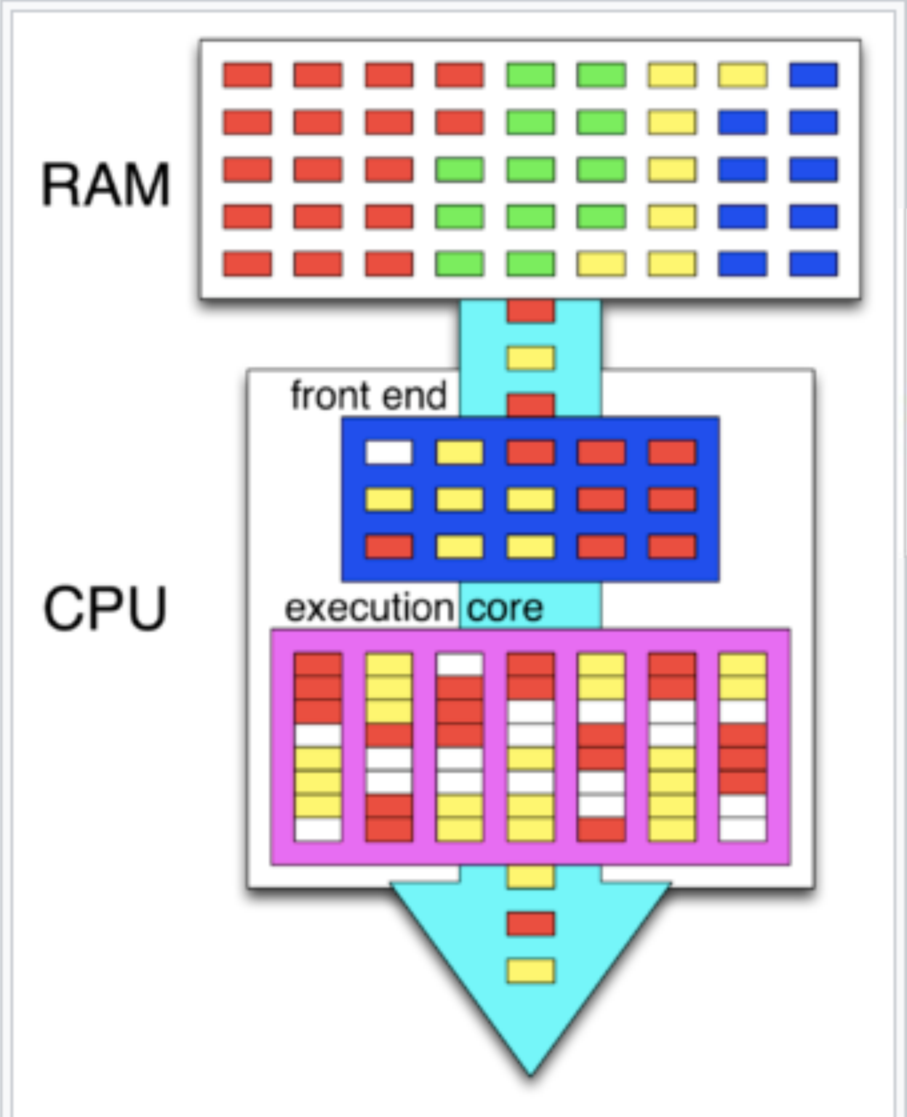
Hyper-threading (officially called **Hyper-Threading Technology** or **HT Technology** and abbreviated as **HTT** or **HT**) is Intel's proprietary simultaneous multithreading (SMT) implementation used to improve parallelization of computations (doing multiple tasks at once) performed on x86 microprocessors. It first appeared in February 2002 on Xeon server processors and in November 2002 on Pentium 4 desktop CPUs.[4] Later, Intel included this technology in Itanium, Atom, and Core 'i' Series CPUs, among others.

For each processor core that is physically present, the operating system addresses two virtual (logical) cores and shares the workload between them when possible. The main function of hyper-threading is to increase the number of independent instructions in the pipeline; it takes advantage of superscalar architecture, in which multiple instructions operate on separate data in parallel. With HTT, one physical core appears as two processors to the operating system, allowing concurrent scheduling of two processes per core. In addition, two or more processes can use the same resources: If resources for one process are not available, then another process can continue if its resources are available.

In addition to requiring simultaneous multithreading (SMT) support in the operating system, hyper-threading can be properly utilized only with an operating system specifically optimized for it.[5] Furthermore, Intel recommends HTT to be disabled when using operating systems unaware of this hardware feature.[*citation needed*]

Hyperthreading

WIKIPEDIA
The Free Encyclopedia



Virtual ***Thread*** Machine

Sits on top of Superscalar Multi-cores

In this high-level depiction of HTT, instructions are fetched from RAM (differently colored boxes represent the instructions of four different processes), decoded and reordered by the front end (white boxes represent pipeline bubbles), and passed to the execution core capable of executing instructions from two different programs during the same clock cycle.[1][2][3]

Intel's proprietary ***HTT***

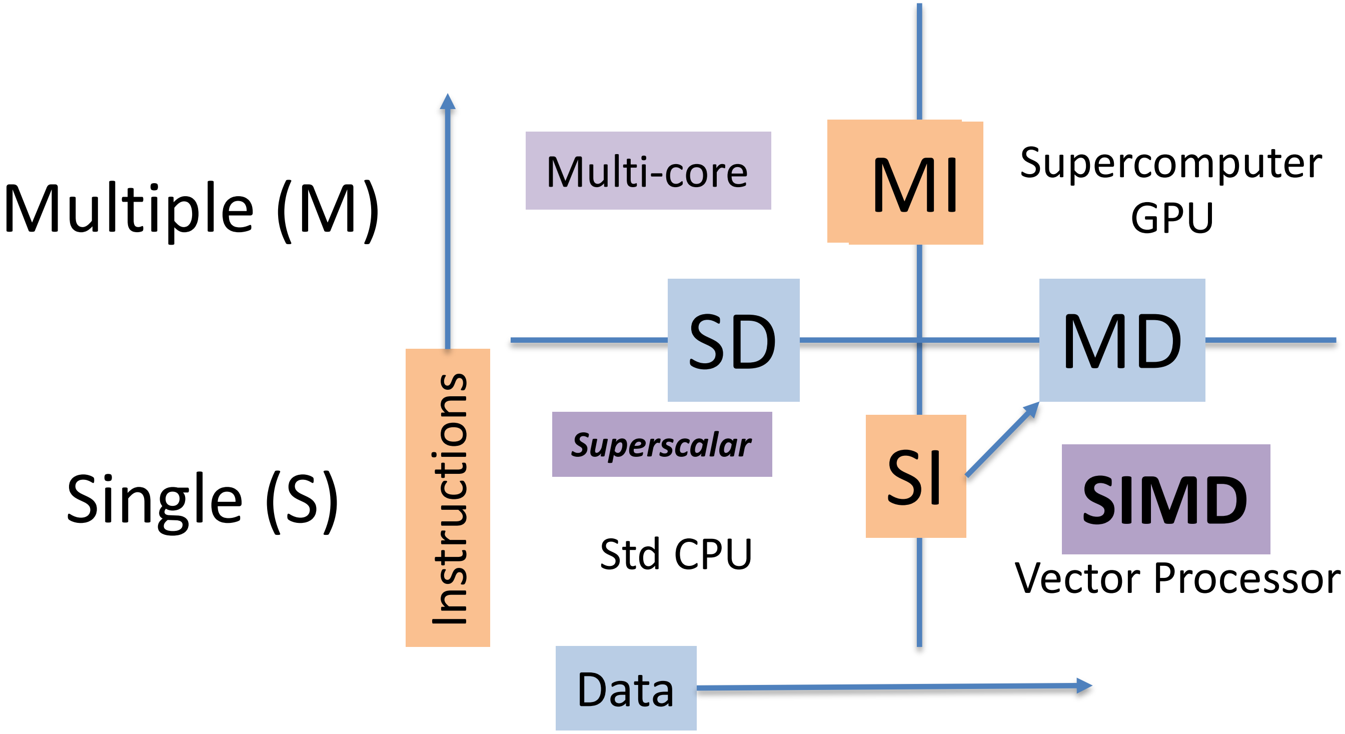
Vector Data Extensions

SIMD/MMX→AVX

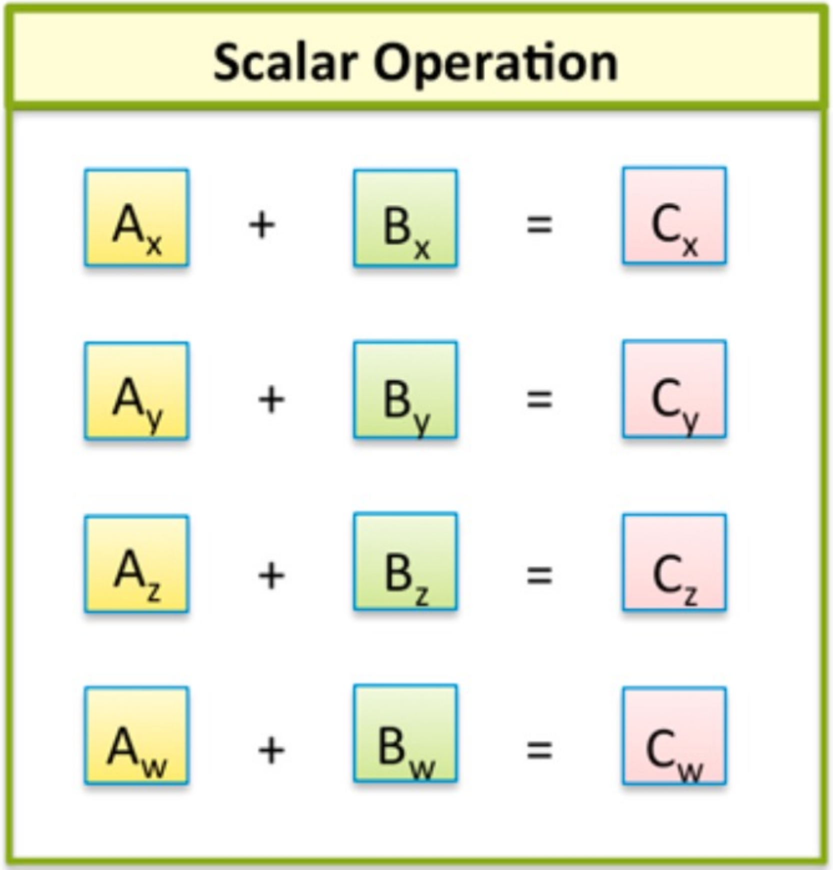
I-D Parallelism: SIMD

Flynn Partition

Michael J. Flynn paper (U Illinois (UIUC), Ca 1969)



SIMD



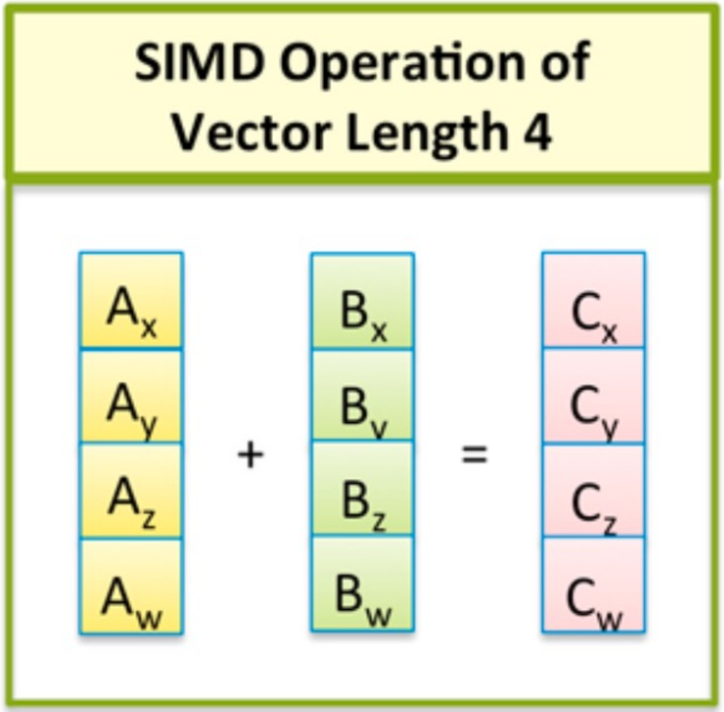
Scalar Operation

$A_x + B_x = C_x$

$A_y + B_y = C_y$

$A_z + B_z = C_z$

$A_w + B_w = C_w$



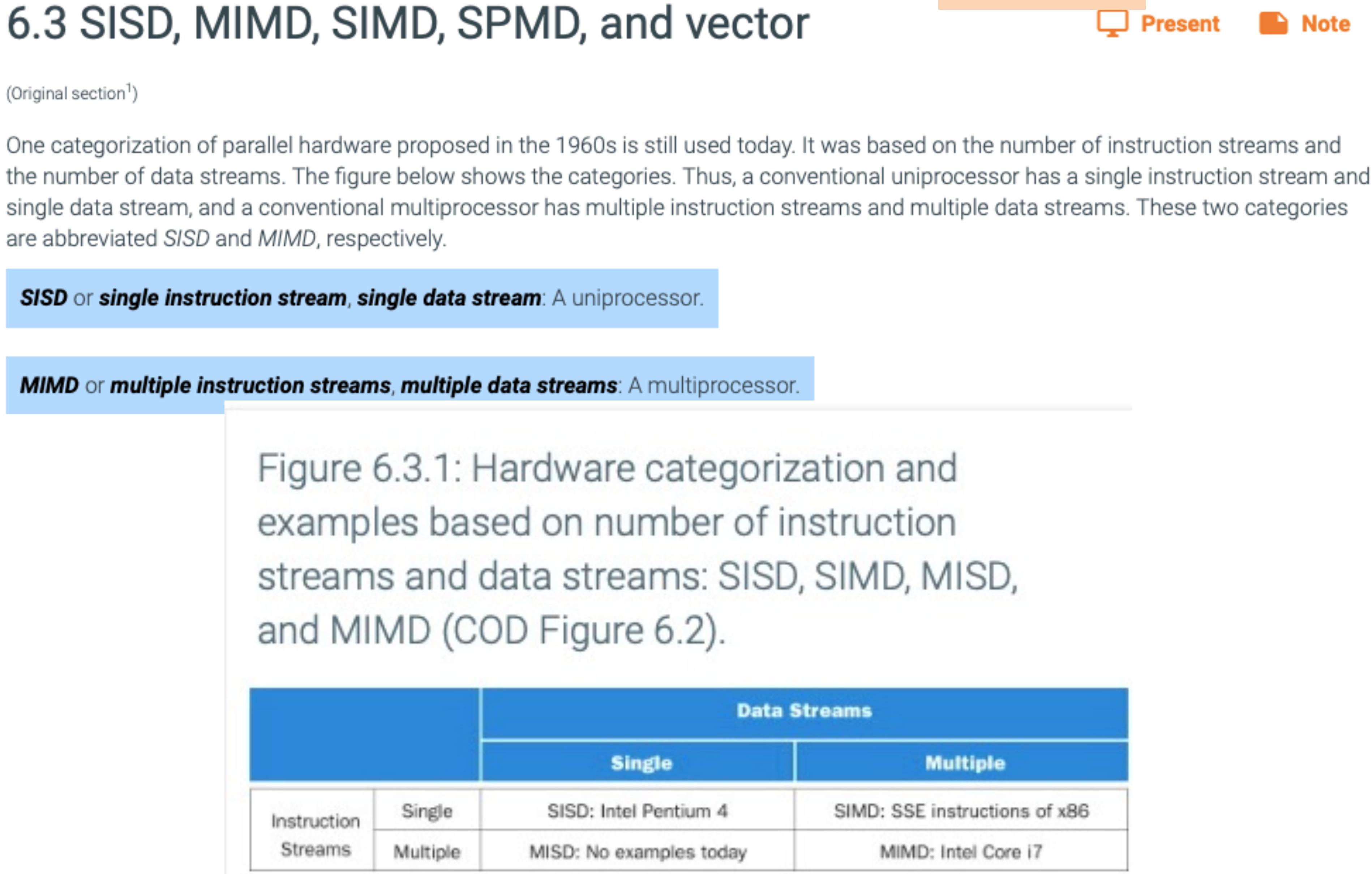
SIMD Operation of Vector Length 4

$$\begin{bmatrix} A_x \\ A_y \\ A_z \\ A_w \end{bmatrix} + \begin{bmatrix} B_x \\ B_y \\ B_z \\ B_w \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \\ C_z \\ C_w \end{bmatrix}$$

Intel® Architecture currently has SIMD operations of vector length 4, 8, 16

SIMD

P&H Ch 6



6.3 SISD, MIMD, SIMD, SPMD, and vector

(Original section[1])

One categorization of parallel hardware proposed in the 1960s is still used today. It was based on the number of instruction streams and the number of data streams. The figure below shows the categories. Thus, a conventional uniprocessor has a single instruction stream and single data stream, and a conventional multiprocessor has multiple instruction streams and multiple data streams. These two categories are abbreviated *SISD* and *MIMD*, respectively.

SISD or ***single instruction stream***, ***single data stream***: A uniprocessor.

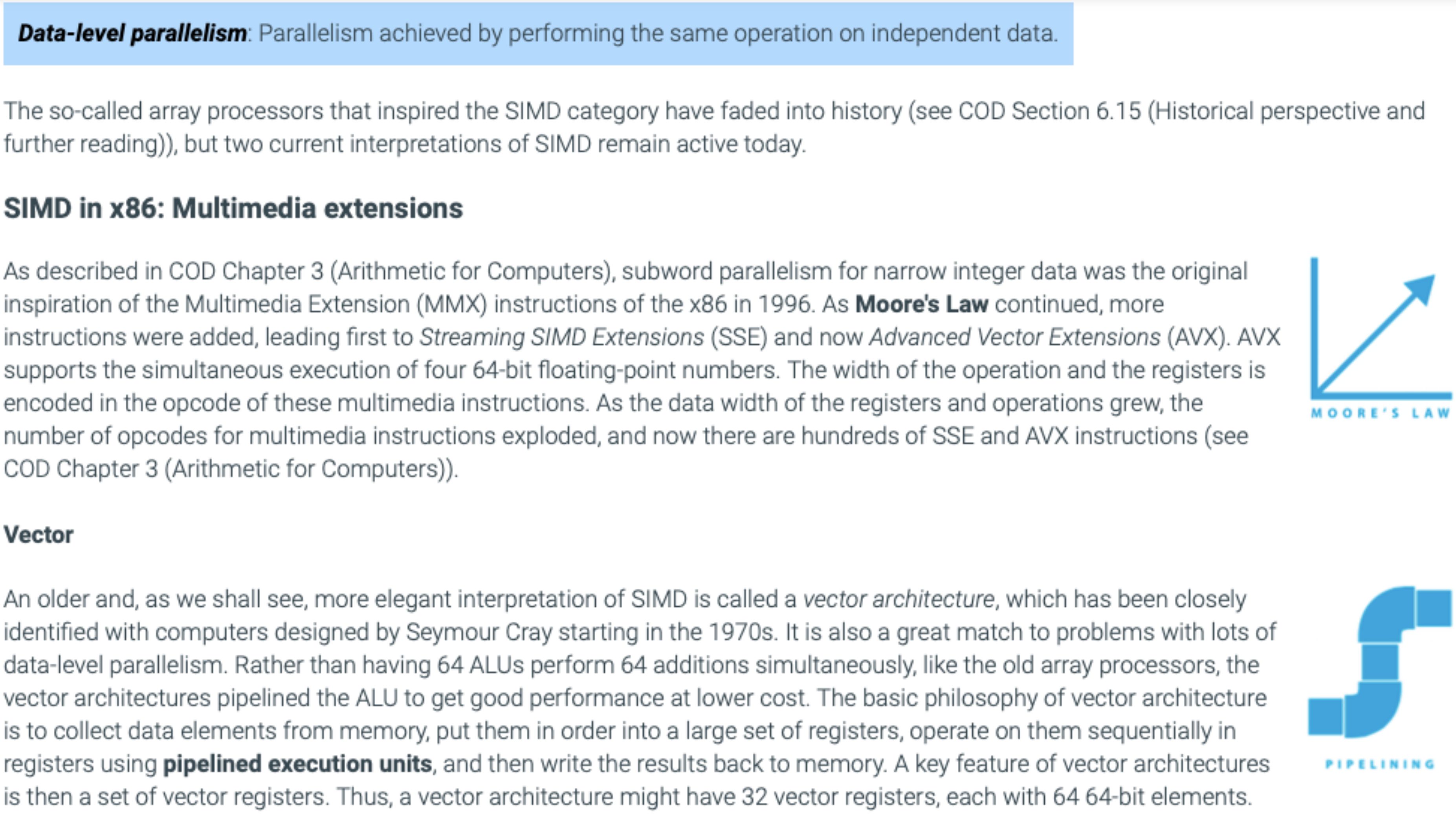
MIMD or ***multiple instruction streams***, ***multiple data streams***: A multiprocessor.

Figure 6.3.1: Hardware categorization and examples based on number of instruction streams and data streams: SISD, SIMD, MISD, and MIMD (COD Figure 6.2).

		Data Streams	
		Single	Multiple
Instruction Streams	Single	SISD: Intel Pentium 4	SIMD: SSE instructions of x86
	Multiple	MISD: No examples today	MIMD: Intel Core i7

SIMD

P&H Ch 6



Data-level parallelism: Parallelism achieved by performing the same operation on independent data.

The so-called array processors that inspired the SIMD category have faded into history (see COD Section 6.15 (Historical perspective and further reading)), but two current interpretations of SIMD remain active today.

SIMD in x86: Multimedia extensions

As described in COD Chapter 3 (Arithmetic for Computers), subword parallelism for narrow integer data was the original inspiration of the Multimedia Extension (MMX) instructions of the x86 in 1996. As **Moore's Law** continued, more instructions were added, leading first to *Streaming SIMD Extensions* (SSE) and now *Advanced Vector Extensions* (AVX). AVX supports the simultaneous execution of four 64-bit floating-point numbers. The width of the operation and the registers is encoded in the opcode of these multimedia instructions. As the data width of the registers and operations grew, the number of opcodes for multimedia instructions exploded, and now there are hundreds of SSE and AVX instructions (see COD Chapter 3 (Arithmetic for Computers)).

Vector

An older and, as we shall see, more elegant interpretation of SIMD is called a *vector architecture*, which has been closely identified with computers designed by Seymour Cray starting in the 1970s. It is also a great match to problems with lots of data-level parallelism. Rather than having 64 ALUs perform 64 additions simultaneously, like the old array processors, the vector architectures pipelined the ALU to get good performance at lower cost. The basic philosophy of vector architecture is to collect data elements from memory, put them in order into a large set of registers, operate on them sequentially in registers using **pipelined execution units**, and then write the results back to memory. A key feature of vector architectures is then a set of vector registers. Thus, a vector architecture might have 32 vector registers, each with 64 64-bit elements.

SIMD ISA Extensions

- **SIMD**
  - **MIPS**
  - **ARM**
- **MMX** = Multi-Media Extensions

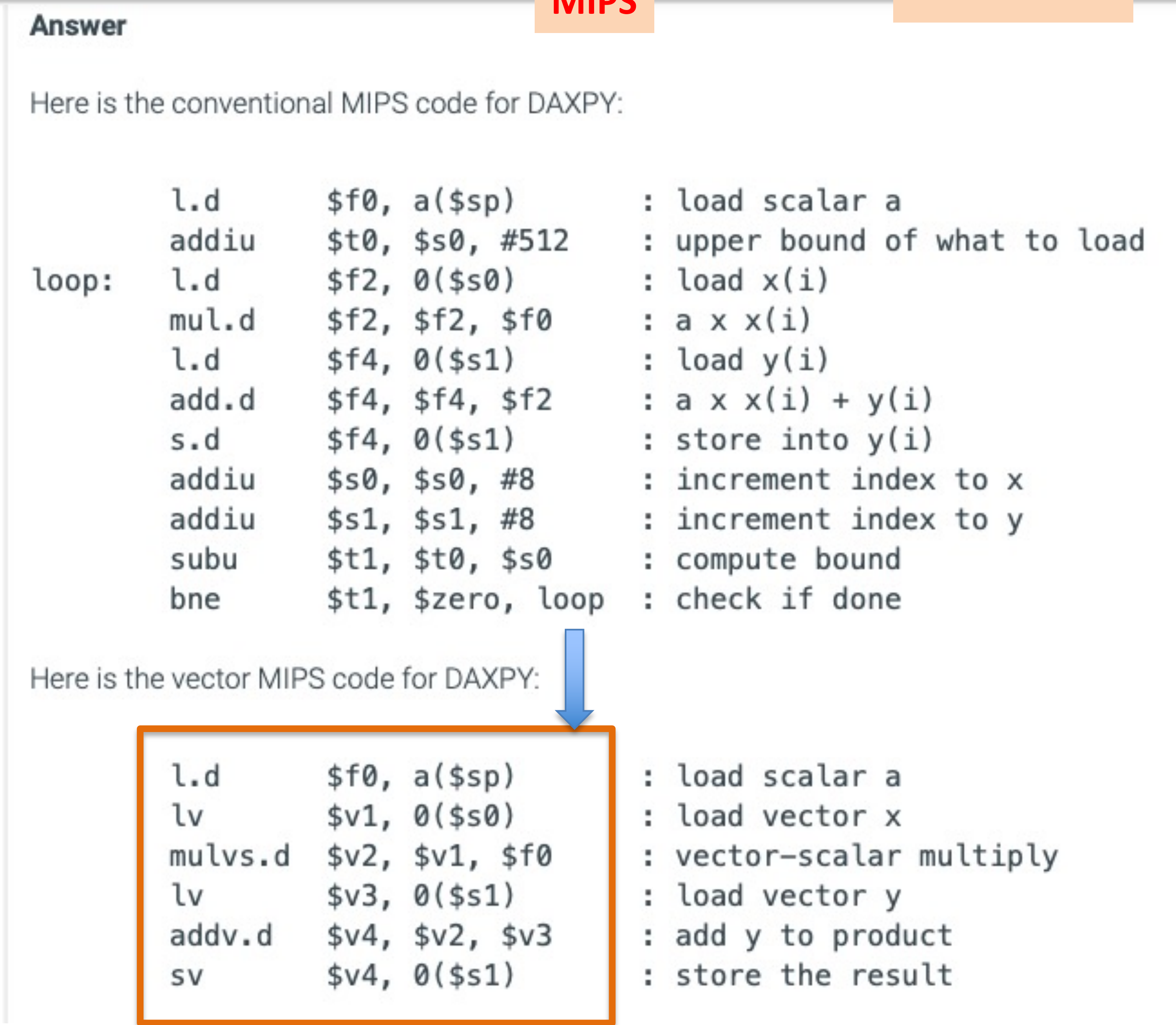
⇩

- **AVX** = Advanced Vector Extensions
  - **Intel**
  - **AMD**

SIMD

MIPS

P&H Ch 6



Answer

Here is the conventional MIPS code for DAXPY:

```
        l.d     $f0, a($sp)         : load scalar a
        addiu   $t0, $s0, #512      : upper bound of what to load
loop:   l.d     $f2, 0($s0)         : load x(i)
        mul.d   $f2, $f2, $f0       : a x x(i)
        l.d     $f4, 0($s1)         : load y(i)
        add.d   $f4, $f4, $f2       : a x x(i) + y(i)
        s.d     $f4, 0($s1)         : store into y(i)
        addiu   $s0, $s0, #8        : increment index to x
        addiu   $s1, $s1, #8        : increment index to y
        subu    $t1, $t0, $s0       : compute bound
        bne     $t1, $zero, loop    : check if done
```

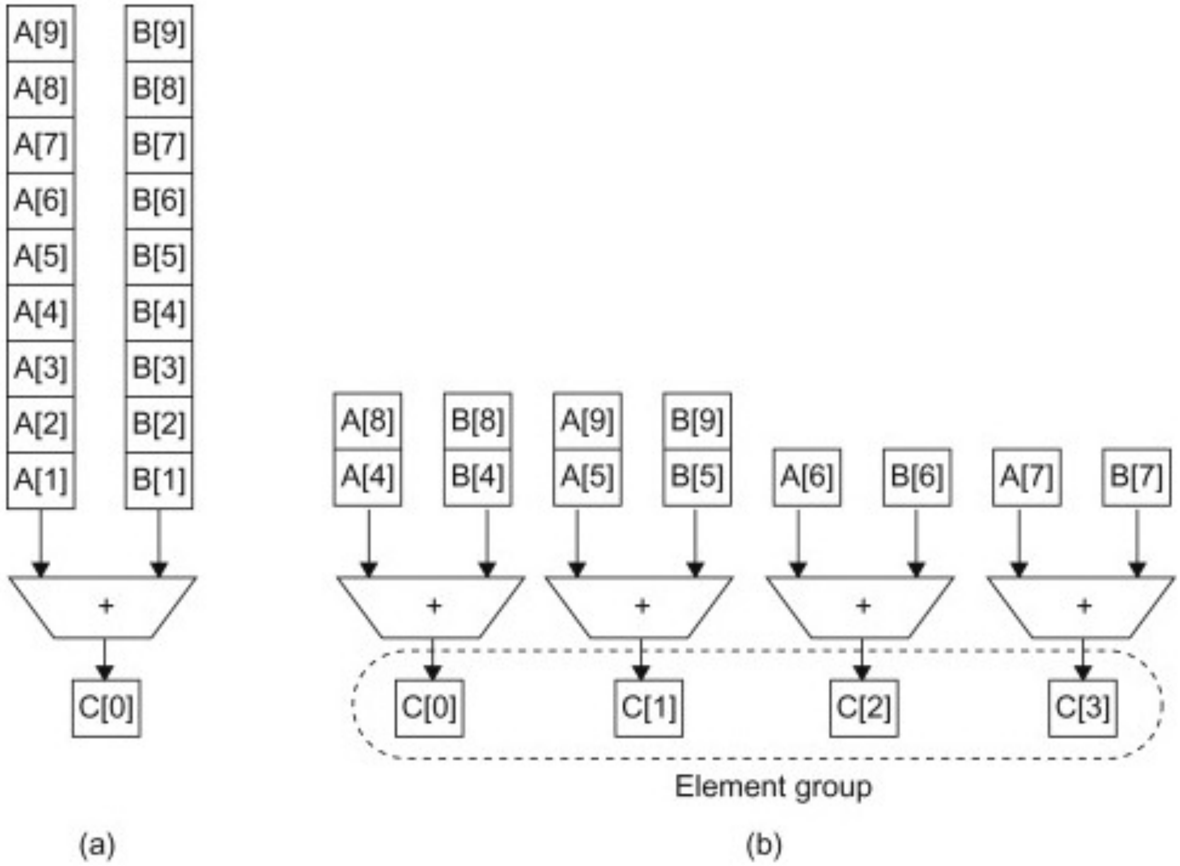
Here is the vector MIPS code for DAXPY:

```
        l.d     $f0, a($sp)         : load scalar a
        lv      $v1, 0($s0)         : load vector x
        mulvs.d $v2, $v1, $f0       : vector-scalar multiply
        lv      $v3, 0($s1)         : load vector y
        addv.d  $v4, $v2, $v3       : add y to product
        sv      $v4, 0($s1)         : store the result
```

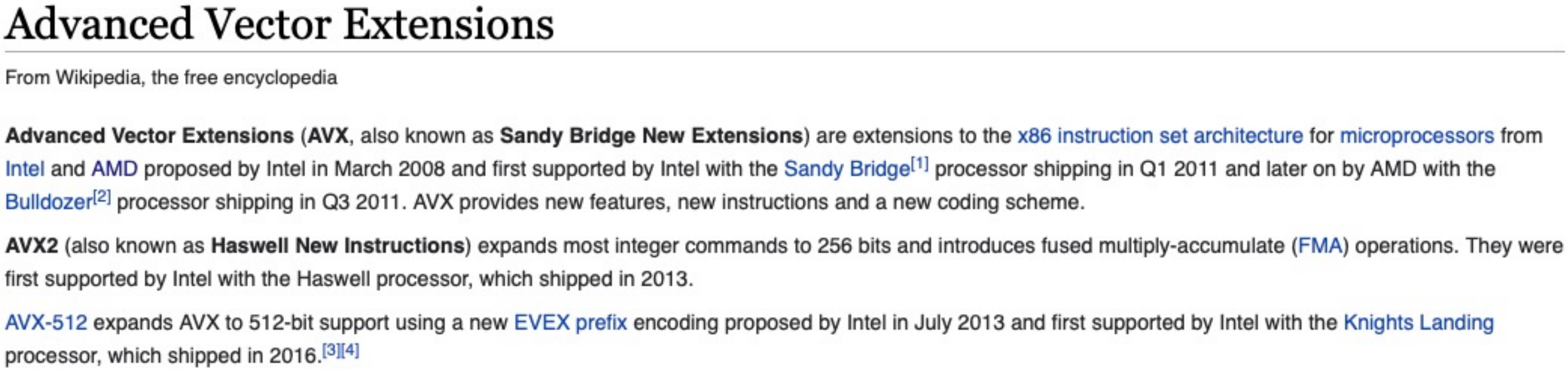
SIMD

Figure 6.3.2: Using multiple functional units to improve the performance of a single vector add instruction, C = A + B (COD Figure 6.3).

The vector processor (a) on the left has a single add pipeline and can complete one addition per cycle. The vector processor (b) on the right has four add pipelines or lanes and can complete four additions per cycle. The elements within a single vector add instruction are interleaved across the four lanes.



Parallel Data: AVX



Advanced Vector Extensions

From Wikipedia, the free encyclopedia

Advanced Vector Extensions (**AVX**, also known as **Sandy Bridge New Extensions**) are extensions to the x86 instruction set architecture for microprocessors from Intel and AMD proposed by Intel in March 2008 and first supported by Intel with the Sandy Bridge[1] processor shipping in Q1 2011 and later on by AMD with the Bulldozer[2] processor shipping in Q3 2011. AVX provides new features, new instructions and a new coding scheme.

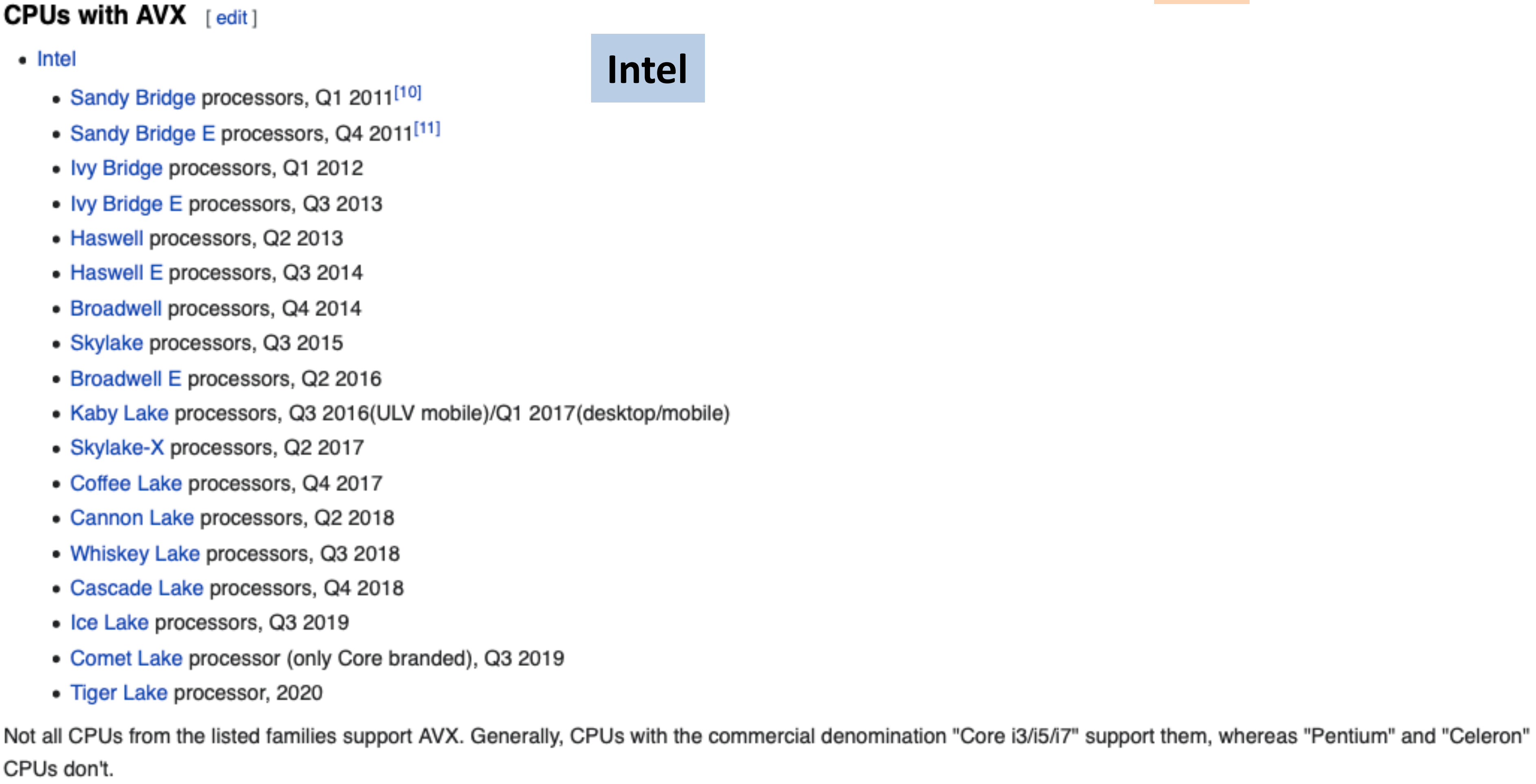
AVX2 (also known as **Haswell New Instructions**) expands most integer commands to 256 bits and introduces fused multiply-accumulate (FMA) operations. They were first supported by Intel with the Haswell processor, which shipped in 2013.

AVX-512 expands AVX to 512-bit support using a new EVEX prefix encoding proposed by Intel in July 2013 and first supported by Intel with the Knights Landing processor, which shipped in 2016.[3][4]

- **MMX** = Multi-Media Extensions
- **AVX** = Advanced Vector Extensions
  - **AVX2**
  - **AVX-512**

Parallel Data: AVX

AVX



CPUs with AVX [edit]

Intel

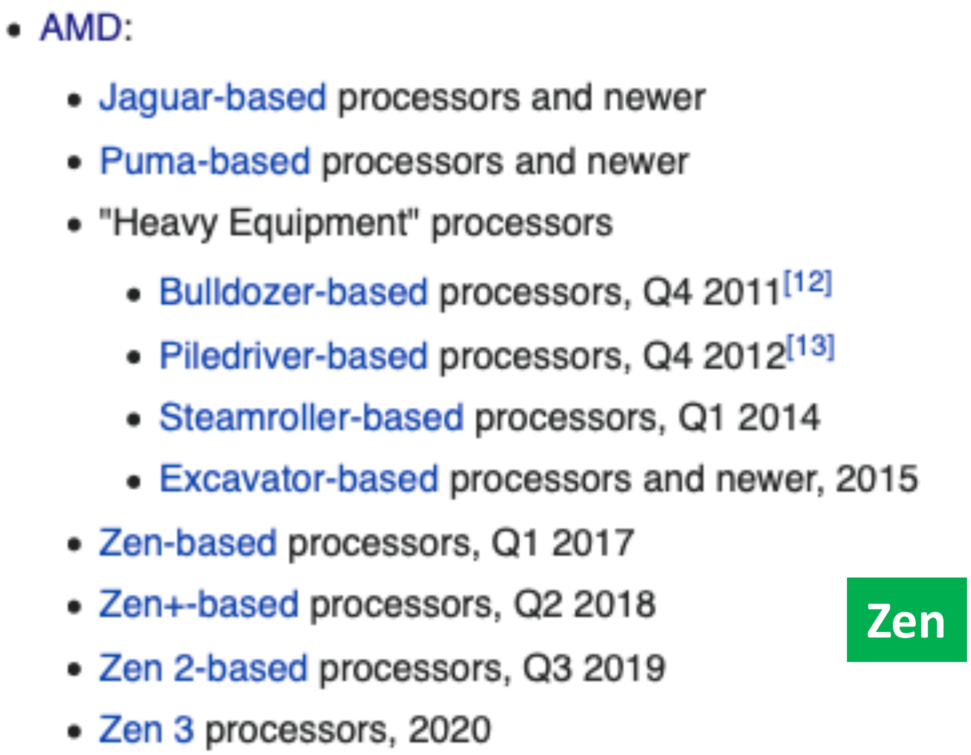
- Intel
  - Sandy Bridge processors, Q1 2011[10]
  - Sandy Bridge E processors, Q4 2011[11]
  - Ivy Bridge processors, Q1 2012
  - Ivy Bridge E processors, Q3 2013
  - Haswell processors, Q2 2013
  - Haswell E processors, Q3 2014
  - Broadwell processors, Q4 2014
  - Skylake processors, Q3 2015
  - Broadwell E processors, Q2 2016
  - Kaby Lake processors, Q3 2016(ULV mobile)/Q1 2017(desktop/mobile)
  - Skylake-X processors, Q2 2017
  - Coffee Lake processors, Q4 2017
  - Cannon Lake processors, Q2 2018
  - Whiskey Lake processors, Q3 2018
  - Cascade Lake processors, Q4 2018
  - Ice Lake processors, Q3 2019
  - Comet Lake processor (only Core branded), Q3 2019
  - Tiger Lake processor, 2020

Not all CPUs from the listed families support AVX. Generally, CPUs with the commercial denomination "Core i3/i5/i7" support them, whereas "Pentium" and "Celeron" CPUs don't.

Parallel Data: AVX

AVX

AMD

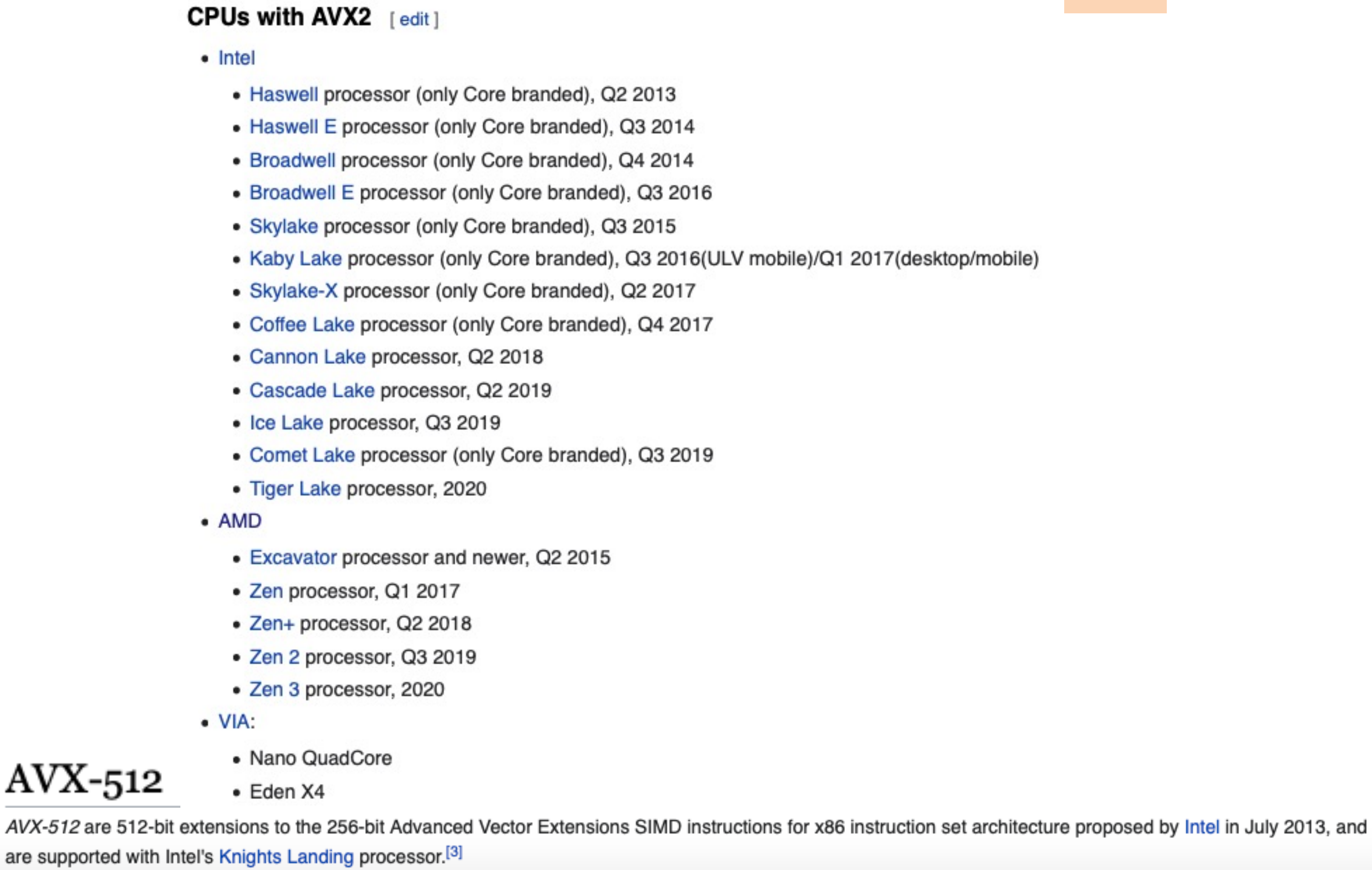


- AMD:
  - Jaguar-based processors and newer
  - Puma-based processors and newer
  - "Heavy Equipment" processors
    - Bulldozer-based processors, Q4 2011[12]
    - Piledriver-based processors, Q4 2012[13]
    - Steamroller-based processors, Q1 2014
    - Excavator-based processors and newer, 2015
  - Zen-based processors, Q1 2017
  - Zen+-based processors, Q2 2018
  - Zen 2-based processors, Q3 2019
  - Zen 3 processors, 2020

Zen

Parallel Data: AVX

AVX2



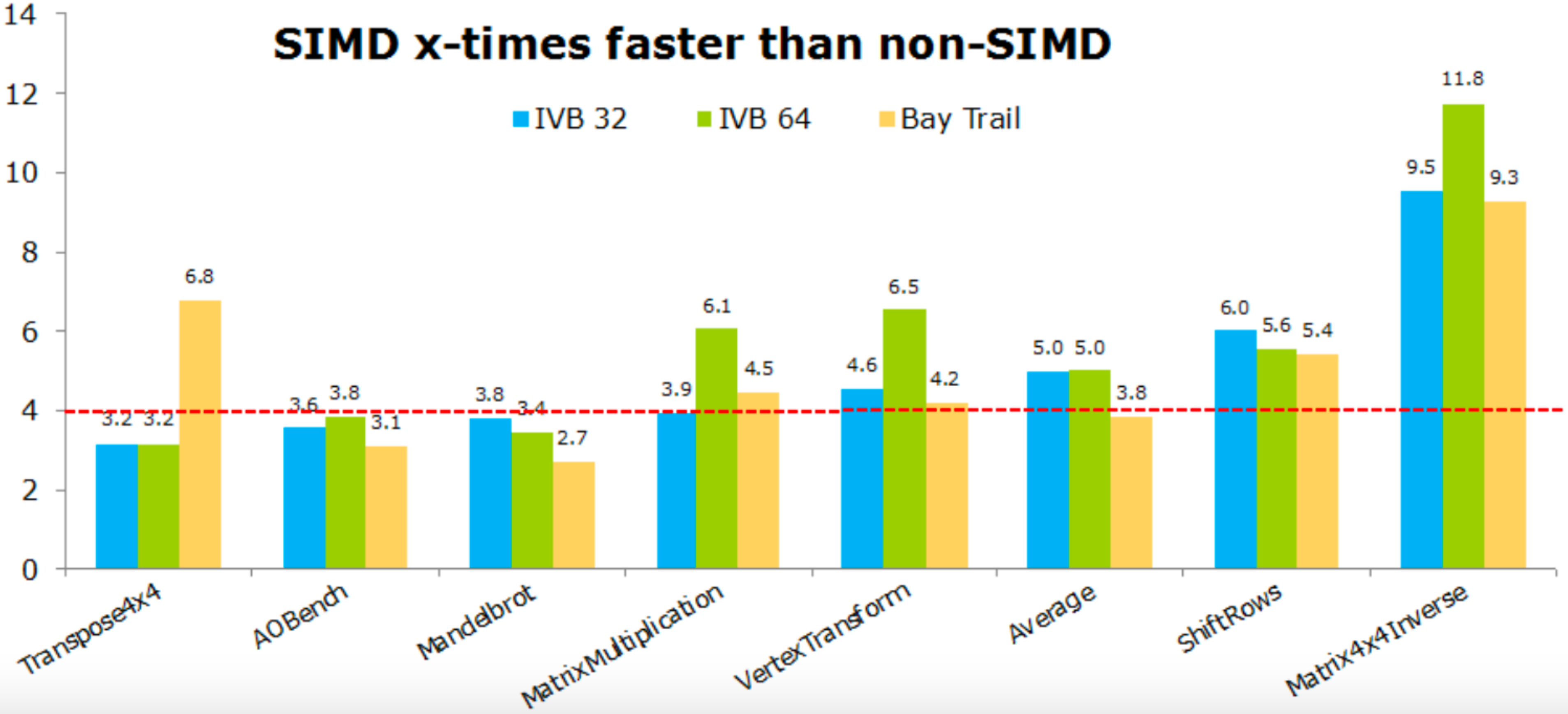
CPUs with AVX2 [edit]

- Intel
  - Haswell processor (only Core branded), Q2 2013
  - Haswell E processor (only Core branded), Q3 2014
  - Broadwell processor (only Core branded), Q4 2014
  - Broadwell E processor (only Core branded), Q3 2016
  - Skylake processor (only Core branded), Q3 2015
  - Kaby Lake processor (only Core branded), Q3 2016(ULV mobile)/Q1 2017(desktop/mobile)
  - Skylake-X processor (only Core branded), Q2 2017
  - Coffee Lake processor (only Core branded), Q4 2017
  - Cannon Lake processor, Q2 2018
  - Cascade Lake processor, Q2 2019
  - Ice Lake processor, Q3 2019
  - Comet Lake processor (only Core branded), Q3 2019
  - Tiger Lake processor, 2020
- AMD
  - Excavator processor and newer, Q2 2015
  - Zen processor, Q1 2017
  - Zen+ processor, Q2 2018
  - Zen 2 processor, Q3 2019
  - Zen 3 processor, 2020
- VIA:
  - Nano QuadCore
  - Eden X4

AVX-512

AVX-512 are 512-bit extensions to the 256-bit Advanced Vector Extensions SIMD instructions for x86 instruction set architecture proposed by Intel in July 2013, and are supported with Intel's Knights Landing processor.[3]

SIMD Benchmark

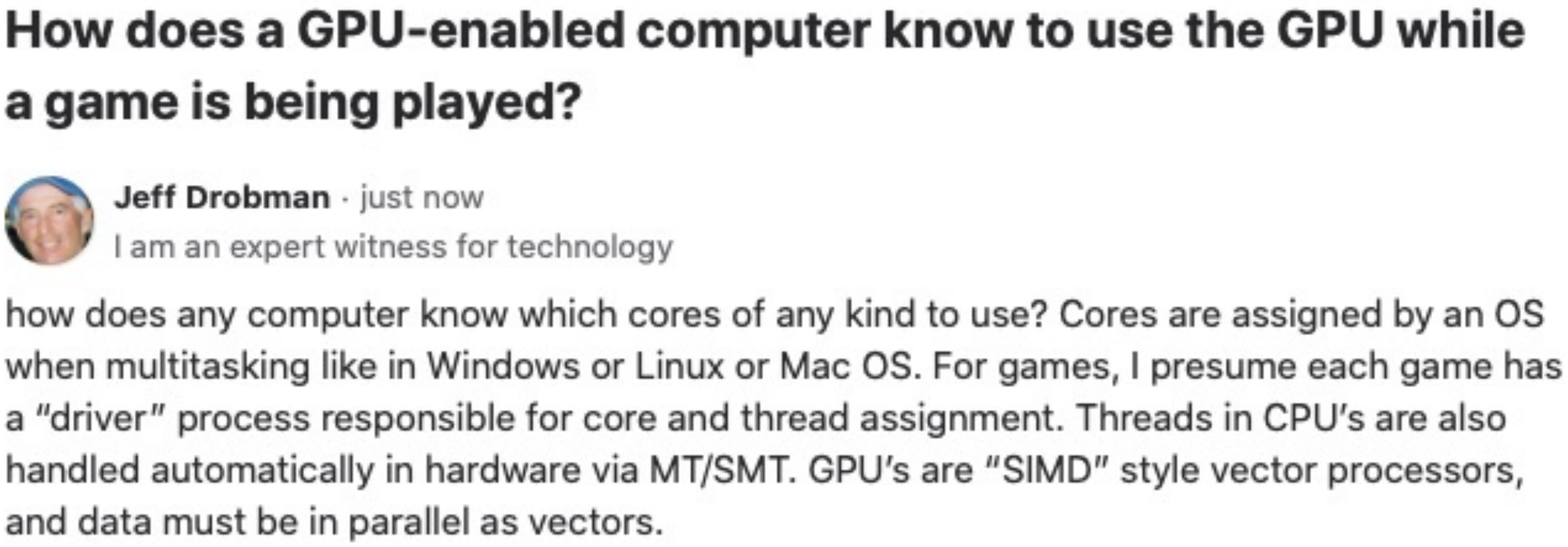


Advanced Material

GPU

- See separate slide set: **GPU**

GPU’s



How does a GPU-enabled computer know to use the GPU while a game is being played?

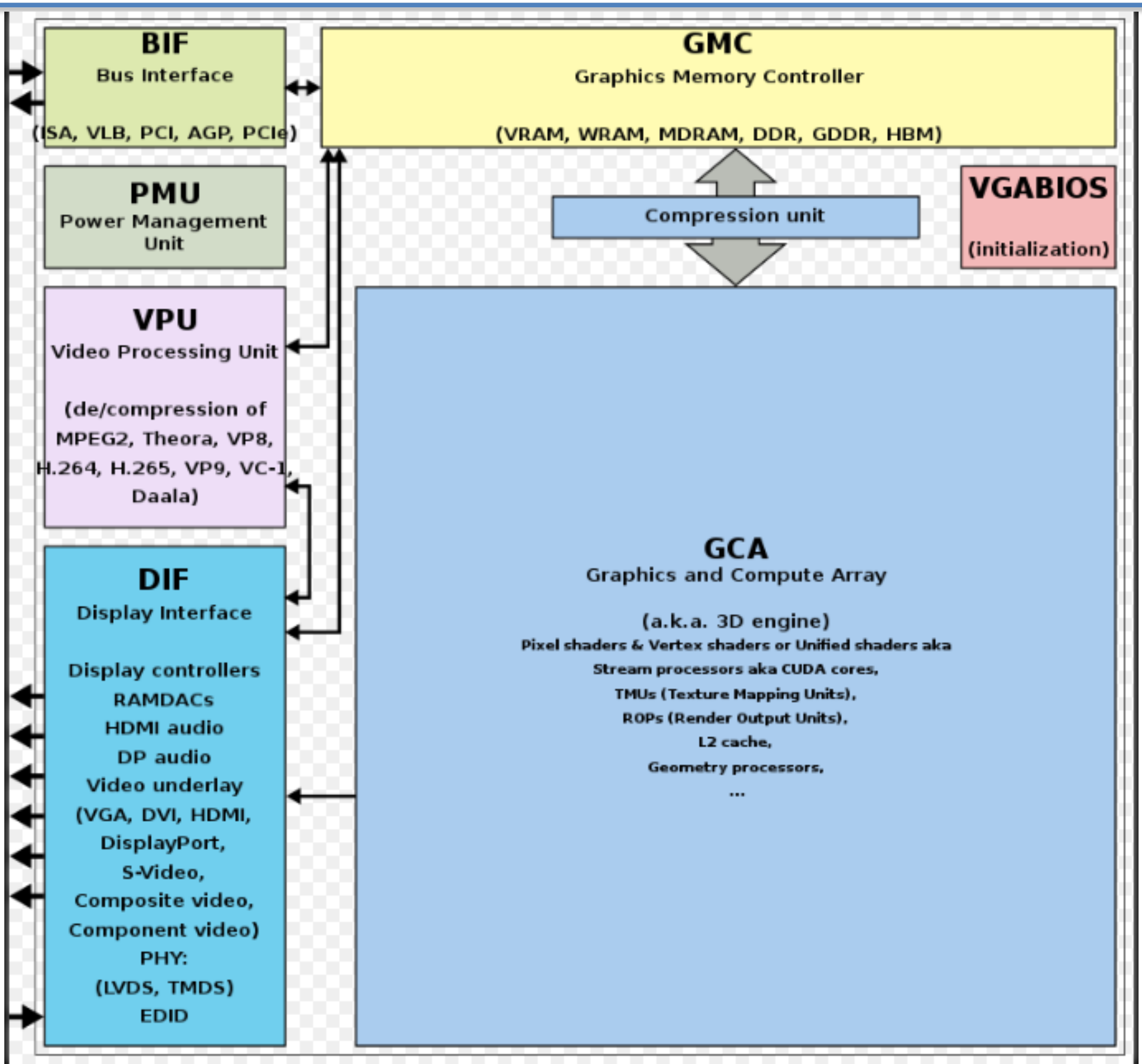
Jeff Drobman · just now
I am an expert witness for technology

how does any computer know which cores of any kind to use? Cores are assigned by an OS when multitasking like in Windows or Linux or Mac OS. For games, I presume each game has a “driver” process responsible for core and thread assignment. Threads in CPU’s are also handled automatically in hardware via MT/SMT. GPU’s are “SIMD” style vector processors, and data must be in parallel as vectors.

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

GPU Architecture

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021



BIF
Bus Interface
(ISA, VLB, PCI, AGP, PCIe)

GMC
Graphics Memory Controller
(VRAM, WRAM, MDRAM, DDR, GDDR, HBM)

PMU
Power Management Unit

Compression unit

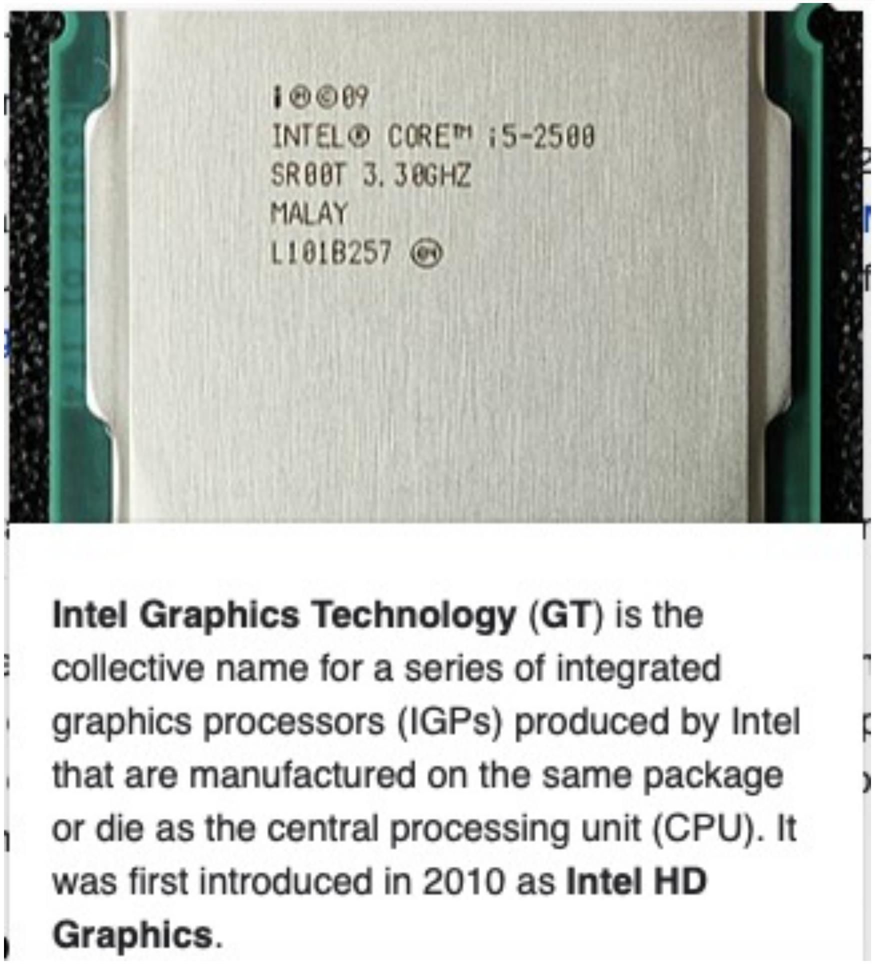
VGABIOS
(initialization)

VPU
Video Processing Unit
(de/compression of MPEG2, Theora, VP8, H.264, H.265, VP9, VC-1, Daala)

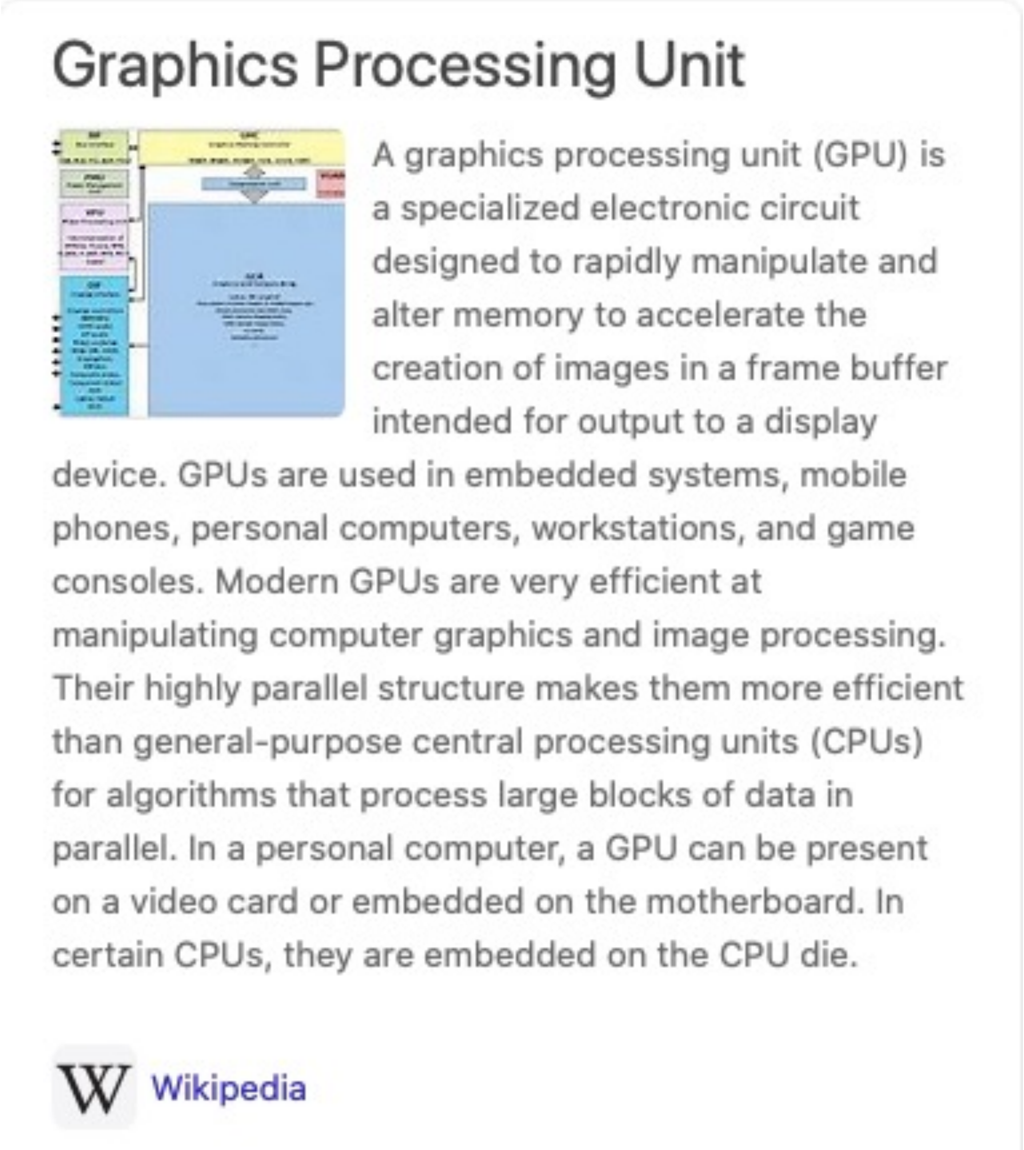
DIF
Display Interface
Display controllers
RAMDACs
HDMI audio
DP audio
Video underlay
(VGA, DVI, HDMI, DisplayPort, S-Video, Composite video, Component video)
PHY:
(LVDS, TMDS)
EDID

GCA
Graphics and Compute Array
(a.k.a. 3D engine)
Pixel shaders & Vertex shaders or Unified shaders aka
Stream processors aka CUDA cores,
TMUs (Texture Mapping Units),
ROPs (Render Output Units),
L2 cache,
Geometry processors,
...

GPU’s



Intel Graphics Technology (**GT**) is the collective name for a series of integrated graphics processors (IGPs) produced by Intel that are manufactured on the same package or die as the central processing unit (CPU). It was first introduced in 2010 as **Intel HD Graphics**.



Graphics Processing Unit

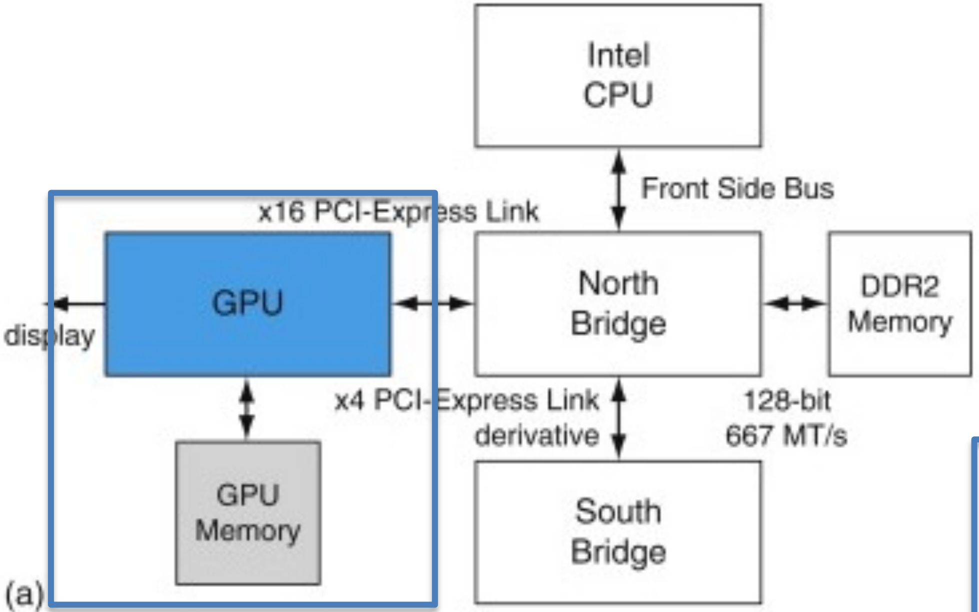
A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing. Their highly parallel structure makes them more efficient than general-purpose central processing units (CPUs) for algorithms that process large blocks of data in parallel. In a personal computer, a GPU can be present on a video card or embedded on the motherboard. In certain CPUs, they are embedded on the CPU die.

W Wikipedia

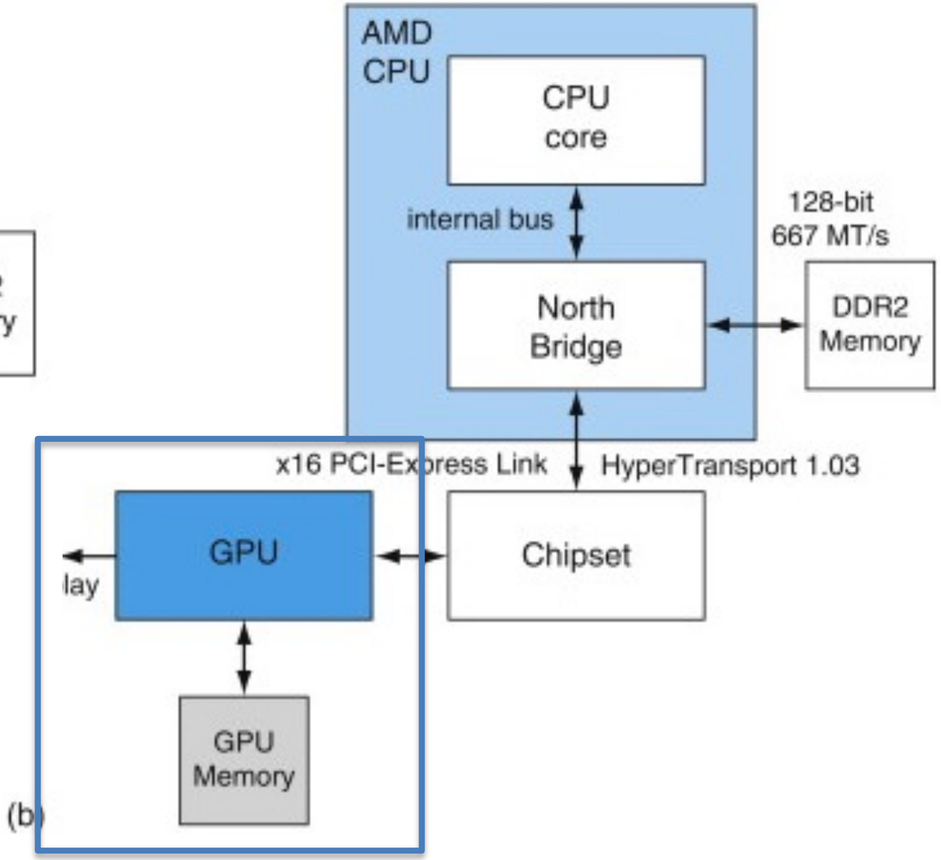
APU = CPU + GPU

Hennessy & Patterson

Intel

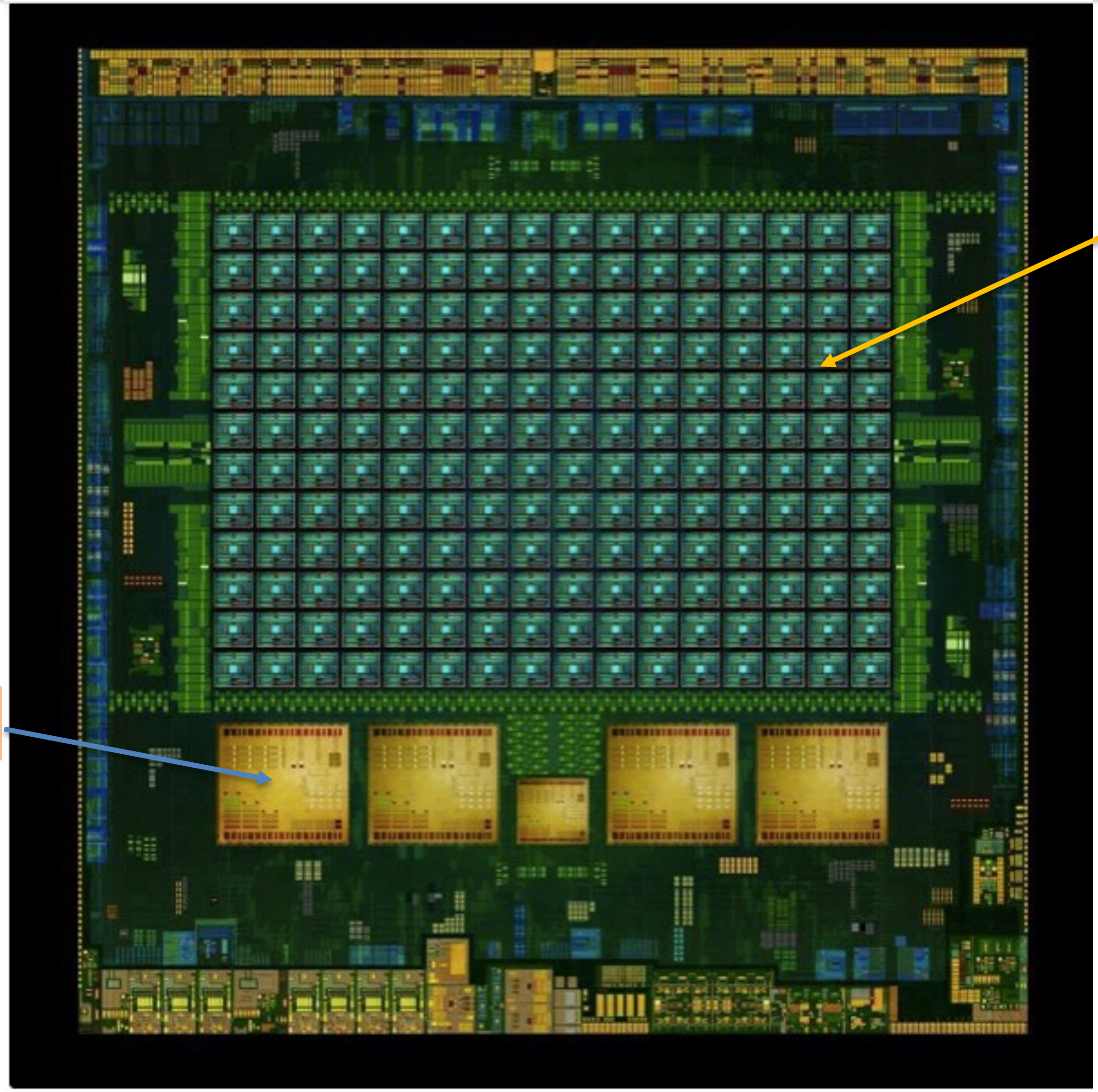


AMD



- See separate slide set "GPU"

SoC = CPU + GPU



GPU cores

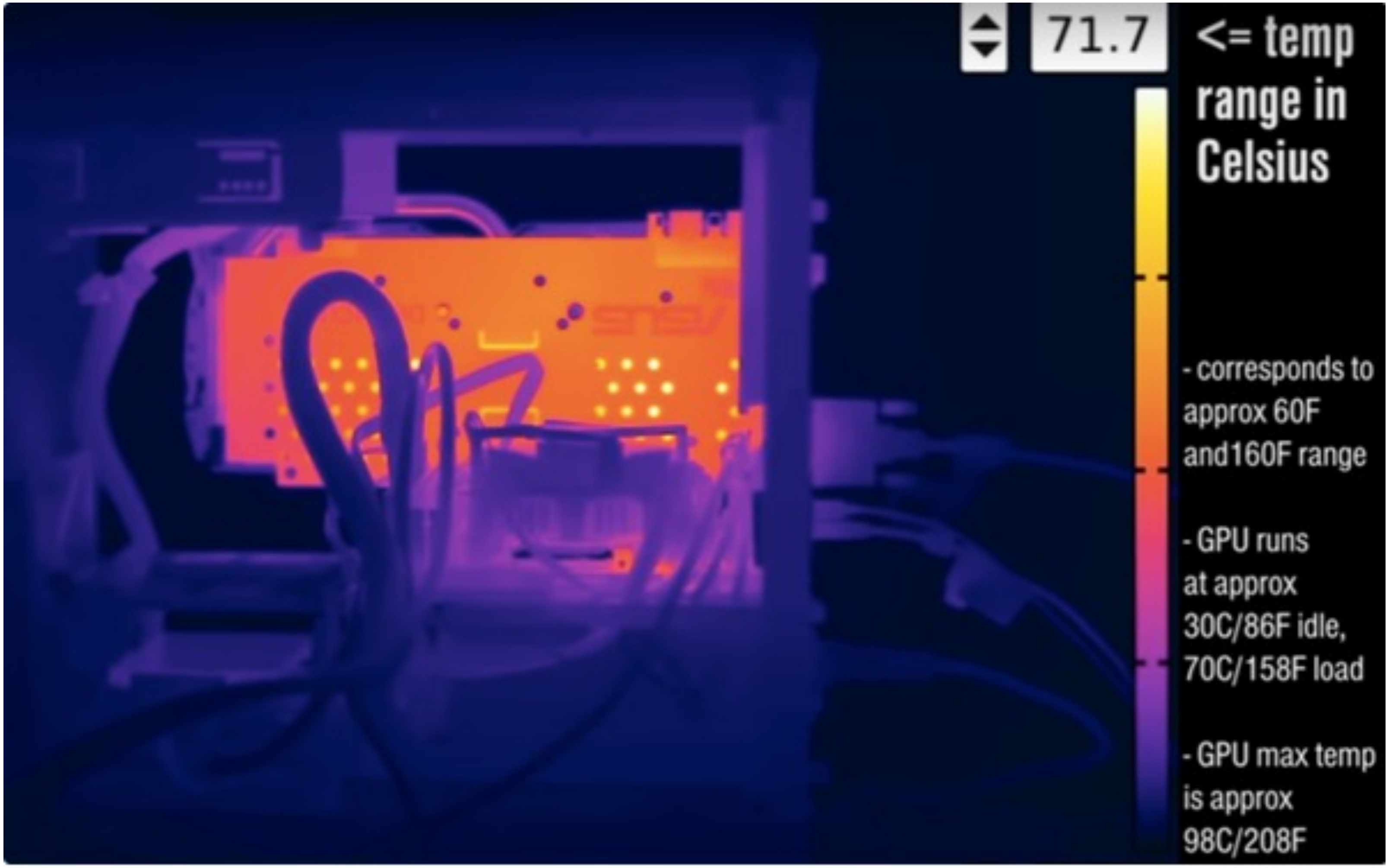
CPU cores

This SOC has four CPU cores (ARM Cortex) and 192 GPU cores (Kepler).

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP122

GPU<CPU Clock Freq

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-2021



<= temp range in Celsius

- corresponds to approx 60F and160F range
- GPU runs at approx 30C/86F idle, 70C/158F load
- GPU max temp is approx 98C/208F

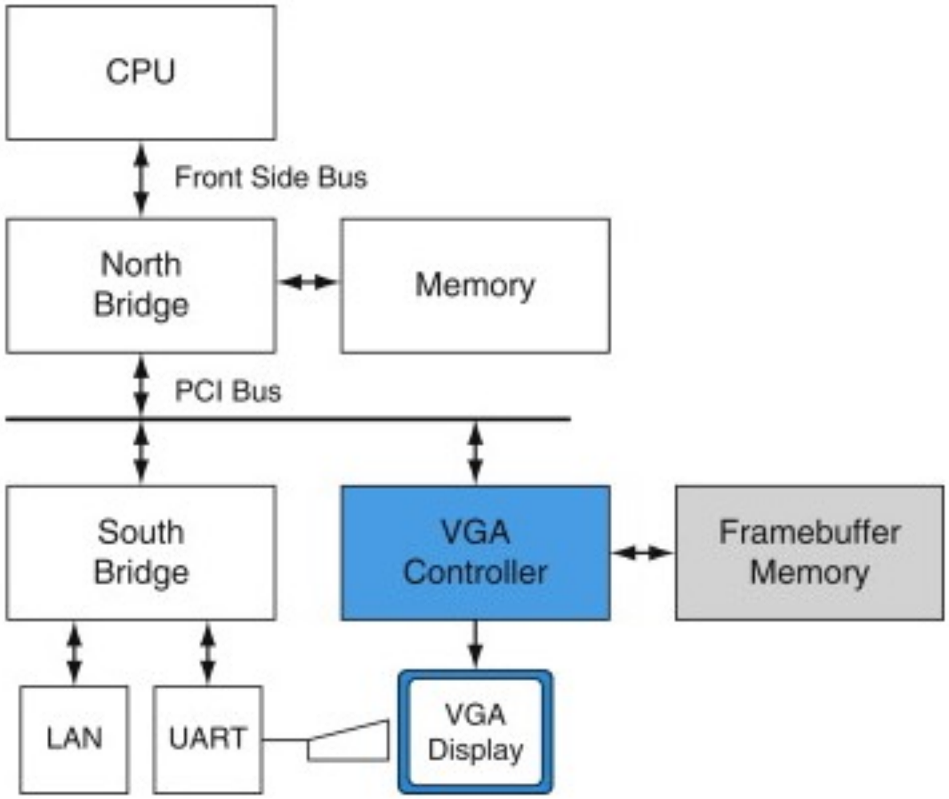
This is an image made with a thermal imaging camera.

GPU

P&H Ch 9

Figure 9.2.1: Historical PC (COD Figure C.2.1).

VGA controller drives graphics display from framebuffer memory.



GPU

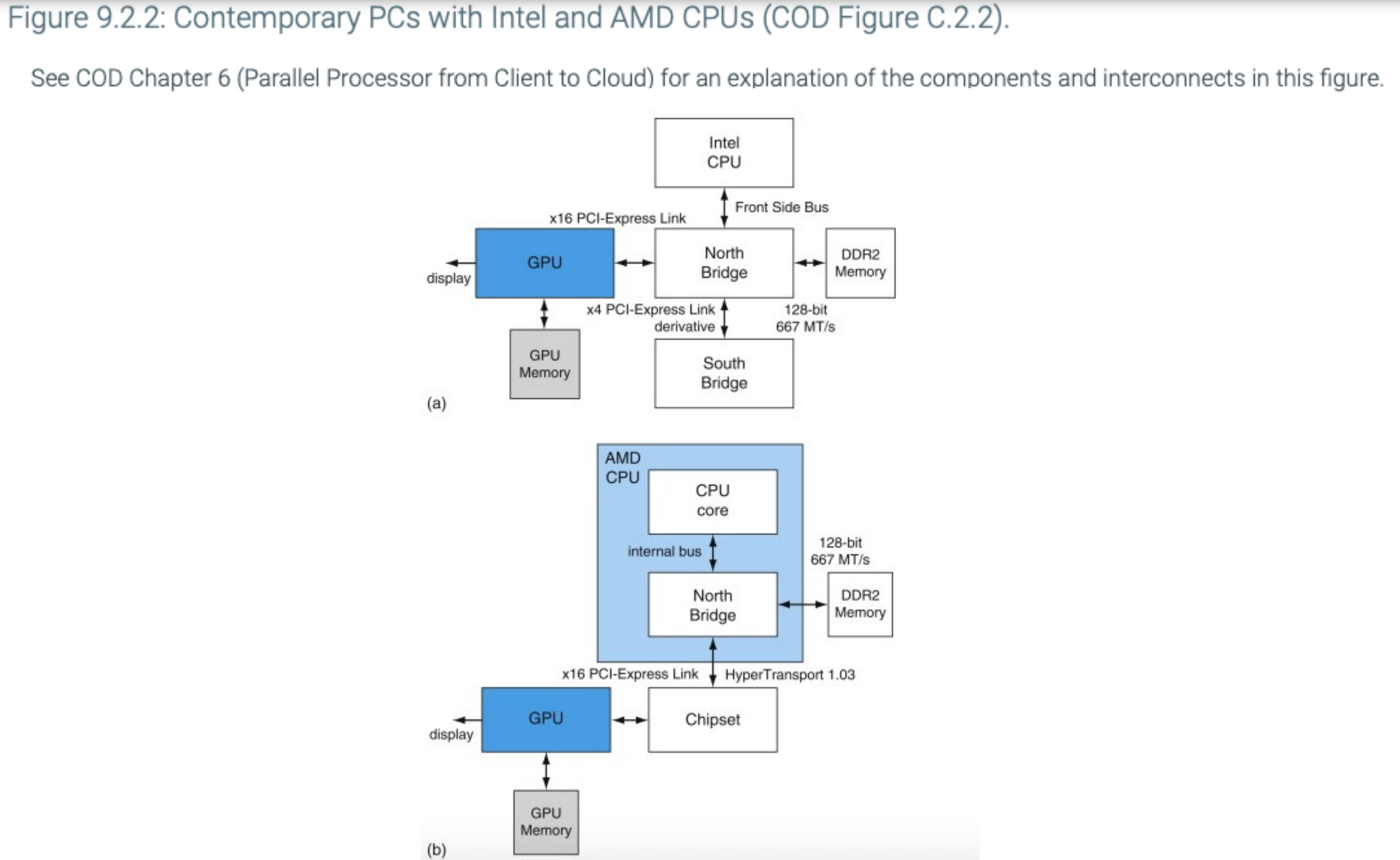
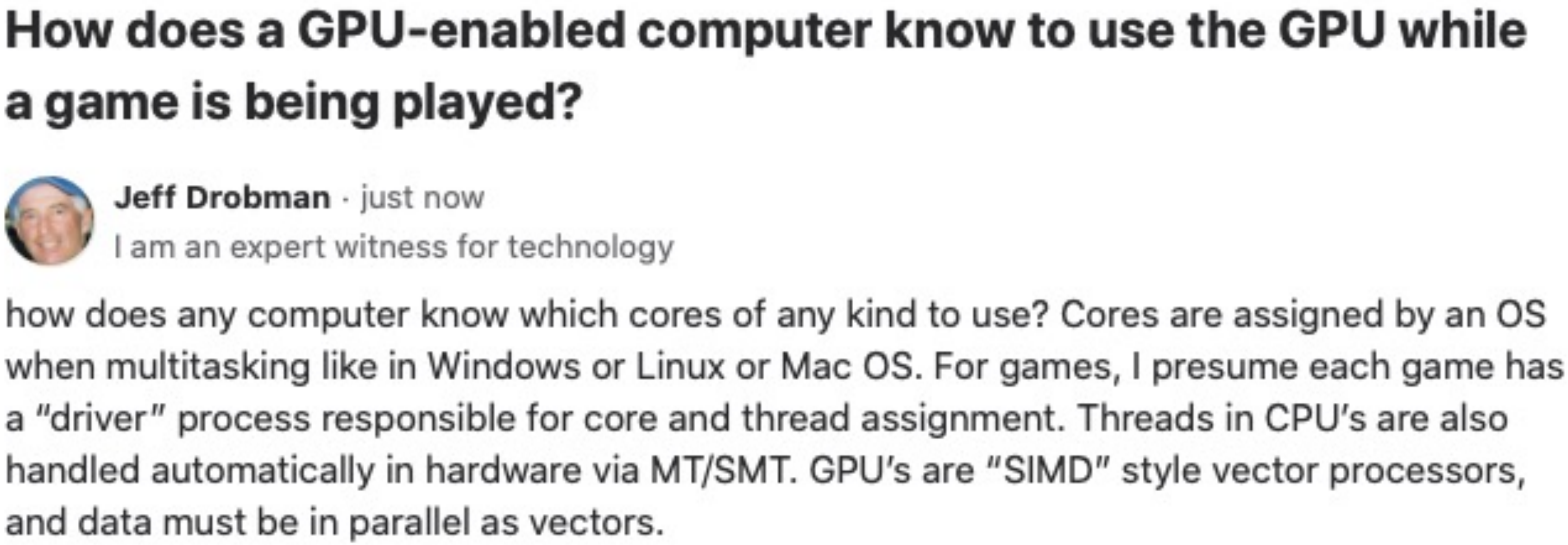


Figure 9.2.2: Contemporary PCs with Intel and AMD CPUs (COD Figure C.2.2).

See COD Chapter 6 (Parallel Processor from Client to Cloud) for an explanation of the components and interconnects in this figure.

GPU's



How does a GPU-enabled computer know to use the GPU while a game is being played?

Jeff Drobman · just now
I am an expert witness for technology

how does any computer know which cores of any kind to use? Cores are assigned by an OS when multitasking like in Windows or Linux or Mac OS. For games, I presume each game has a "driver" process responsible for core and thread assignment. Threads in CPU's are also handled automatically in hardware via MT/SMT. GPU's are "SIMD" style vector processors, and data must be in parallel as vectors.

GPU

Graphics logical pipeline

The graphics logical pipeline is described in COD Section C.3 (Programming GPUs). The figure below illustrates the major processing stages, and highlights the important programmable stages (vertex, geometry, and pixel shader stages).

Figure 9.2.3: Graphics logical pipeline (COD Figure C.2.3).

Programmable graphics shader stages are blue, and fixed-function blocks are white.

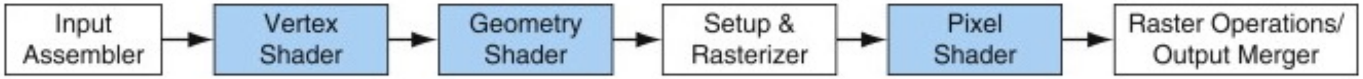
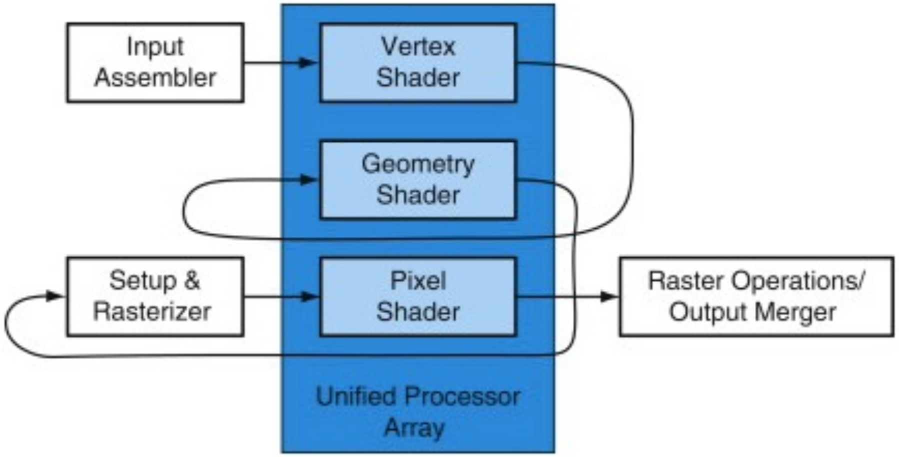


Figure 9.2.4: Logical pipeline mapped to physical processors (COD Figure C.2.4).

The programmable shader stages execute on the array of unified processors, and the logical graphics pipeline dataflow recirculates through the processors.

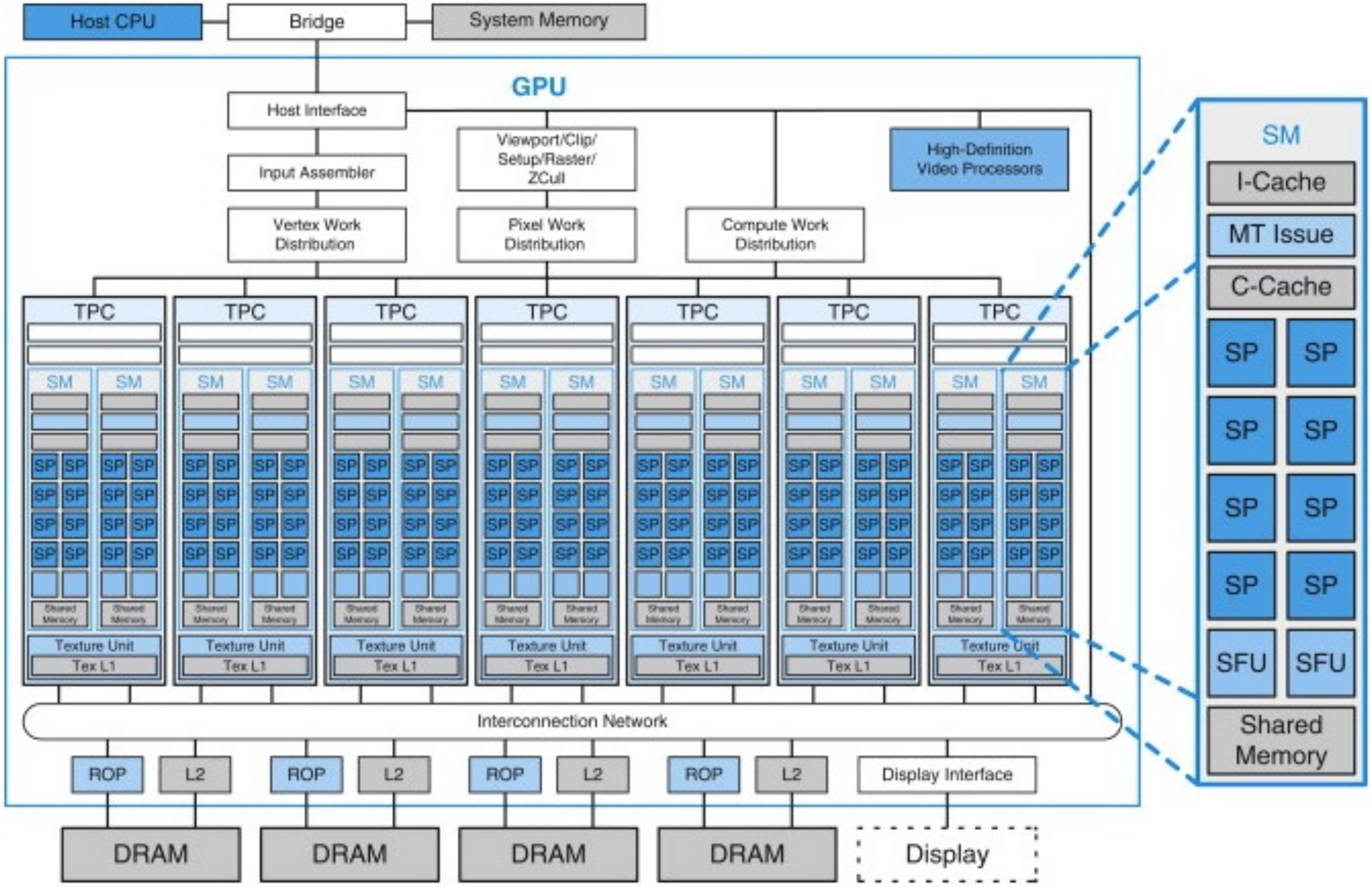


GPU

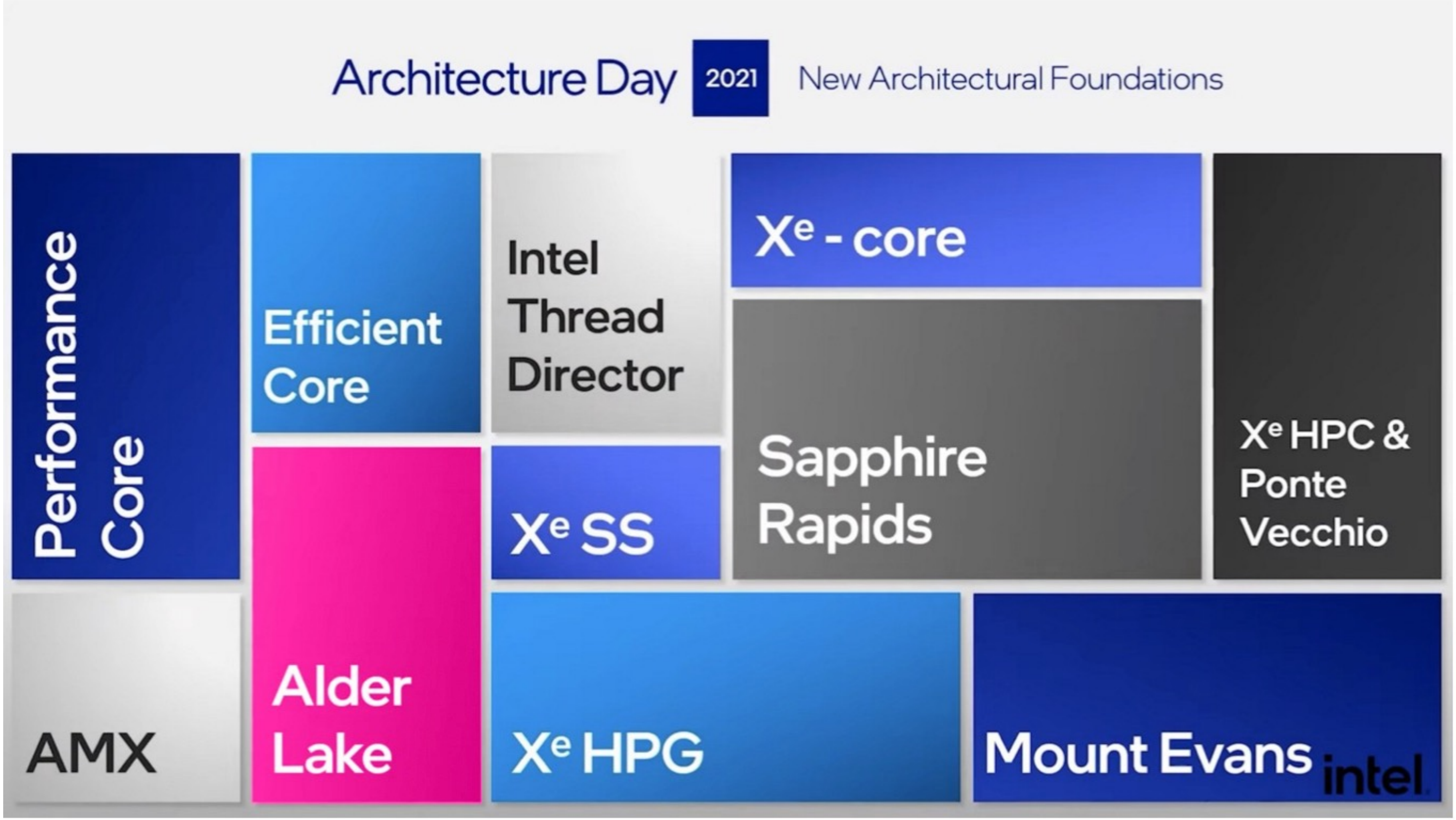
P&H Ch 9

Figure 9.2.5: Basic unified GPU architecture (COD Figure C.2.5).

Example GPU with 112 streaming processor (SP) cores organized in 14 streaming multiprocessors (SMs); the cores are highly multithreaded. It has the basic Tesla architecture of an NVIDIA GeForce 8800. The processors connect with four 64-bit-wide DRAM partitions via an interconnection network. Each SM has eight SP cores, two special function units (SFUs), instruction and constant caches, a multithreaded instruction unit, and a shared memory.



Intel Architecture Day



Architecture Day 2021 New Architectural Foundations

Performance Core

Efficient Core

Intel Thread Director

X^e - core

X^e HPC & Ponte Vecchio

X^e SS

Sapphire Rapids

AMX

Alder Lake

X^e HPG

Mount Evans

intel

Source: Intel Architecture Day (highlight added by author)

Appendix

Wafer Fab

1968	**Intel** founded
1987	**TSMC** founded (foundry)
2009	❖ AMD spins off fabs to **Global Foundries** (owner Abu Dhabi)

(see separate slide set ***Chips & Fabs***)

Appendix

MIX/MMIX

Stanford Prof. Donald Knuth

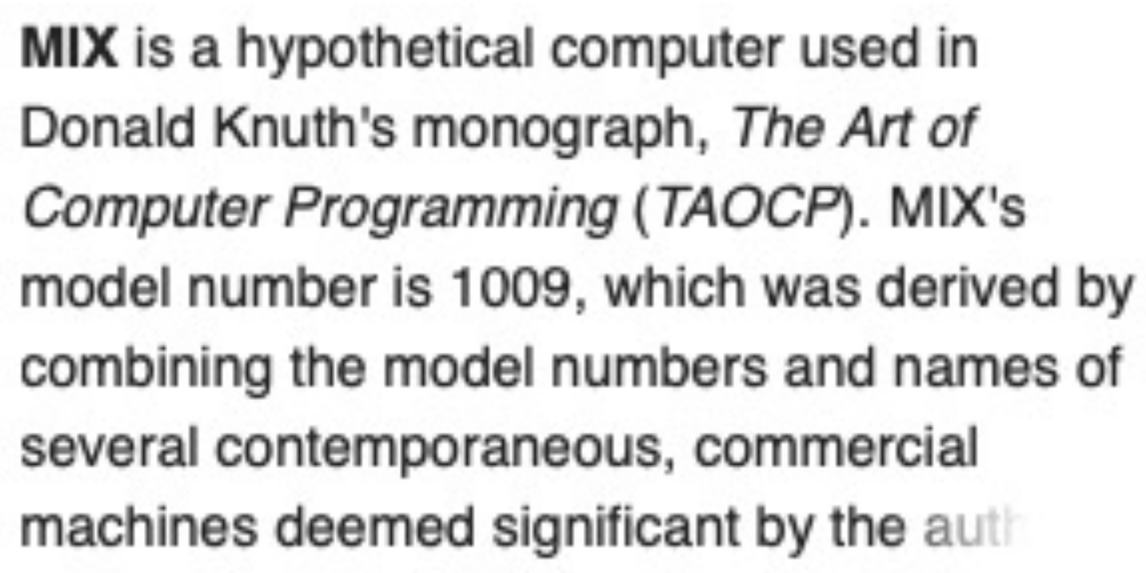
Artificial (Pedagogical) **ISA**

Donald Knuth

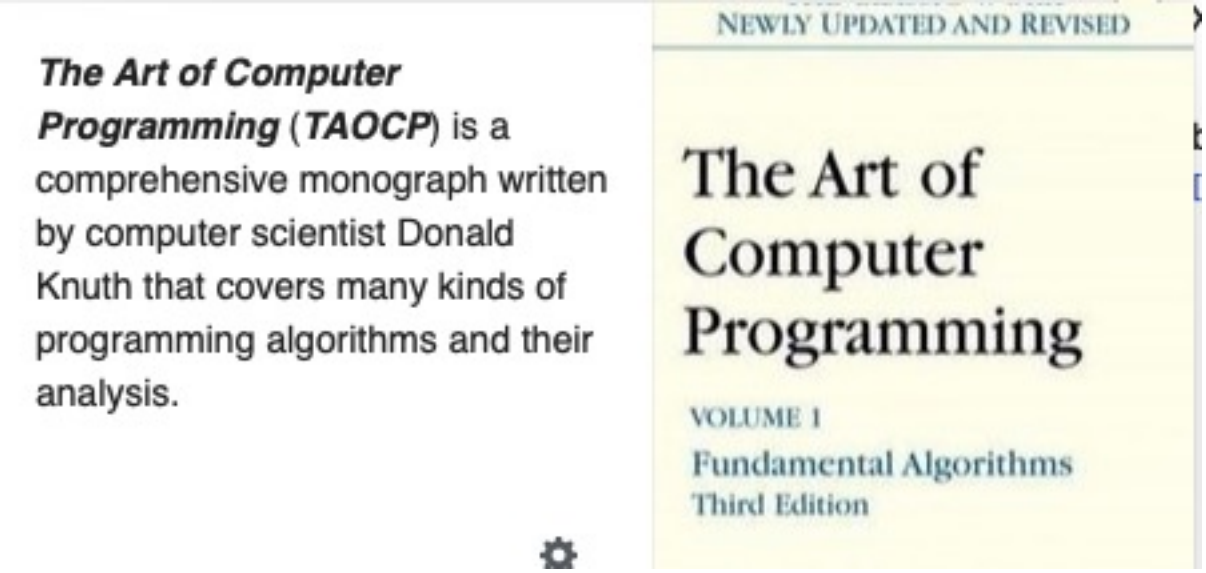
ForMemRS

Knuth in 2005

MIX



MIX is a hypothetical computer used in Donald Knuth's monograph, *The Art of Computer Programming* (*TAOCP*). MIX's model number is 1009, which was derived by combining the model numbers and names of several contemporaneous, commercial machines deemed significant by the aut



The Art of Computer Programming (***TAOCP***) is a comprehensive monograph written by computer scientist Donald Knuth that covers many kinds of programming algorithms and their analysis.

NEWLY UPDATED AND REVISED

The Art of Computer Programming

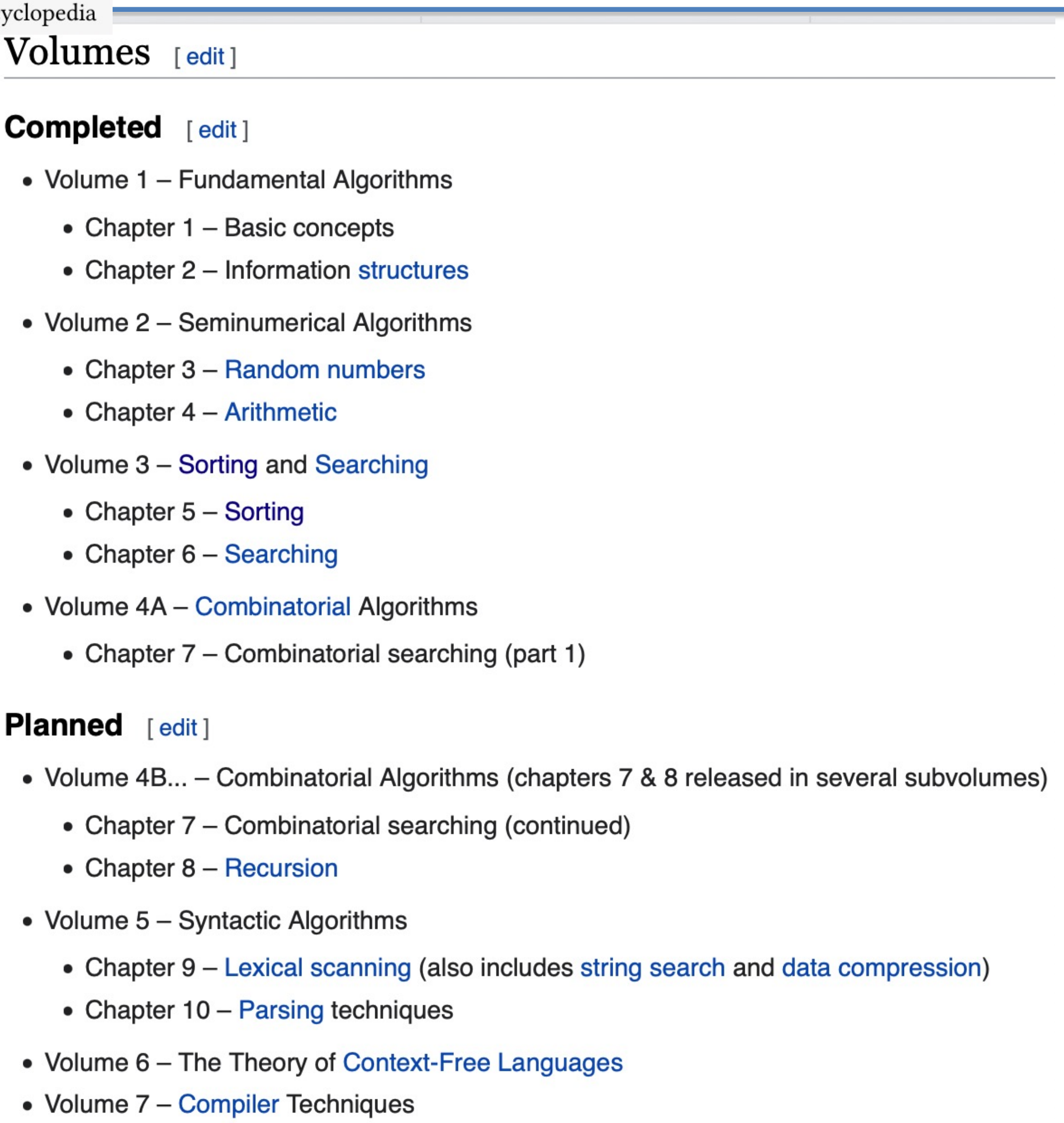
VOLUME 1

Fundamental Algorithms

Third Edition

WIKIPEDIA
The Free Encyclopedia

TAOCP Volumes



Volumes [edit]

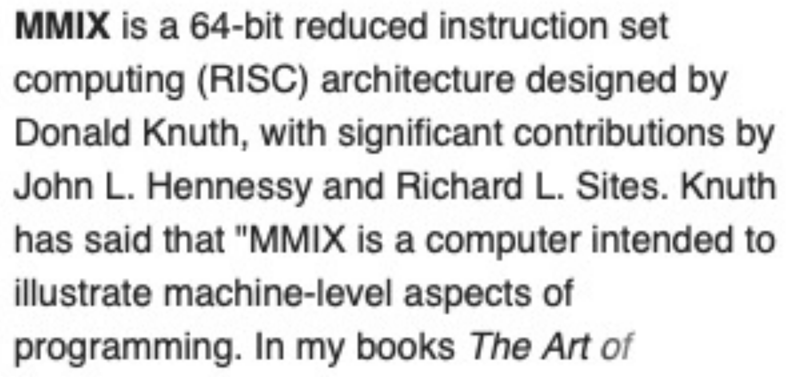
Completed [edit]

- Volume 1 – Fundamental Algorithms
  - Chapter 1 – Basic concepts
  - Chapter 2 – Information structures
- Volume 2 – Seminumerical Algorithms
  - Chapter 3 – Random numbers
  - Chapter 4 – Arithmetic
- Volume 3 – Sorting and Searching
  - Chapter 5 – Sorting
  - Chapter 6 – Searching
- Volume 4A – Combinatorial Algorithms
  - Chapter 7 – Combinatorial searching (part 1)

Planned [edit]

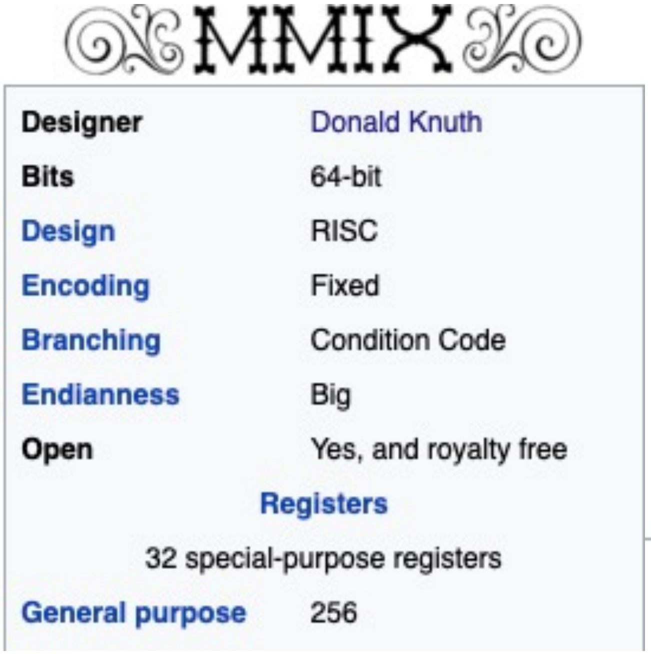
- Volume 4B... – Combinatorial Algorithms (chapters 7 & 8 released in several subvolumes)
  - Chapter 7 – Combinatorial searching (continued)
  - Chapter 8 – Recursion
- Volume 5 – Syntactic Algorithms
  - Chapter 9 – Lexical scanning (also includes string search and data compression)
  - Chapter 10 – Parsing techniques
- Volume 6 – The Theory of Context-Free Languages
- Volume 7 – Compiler Techniques

MIX



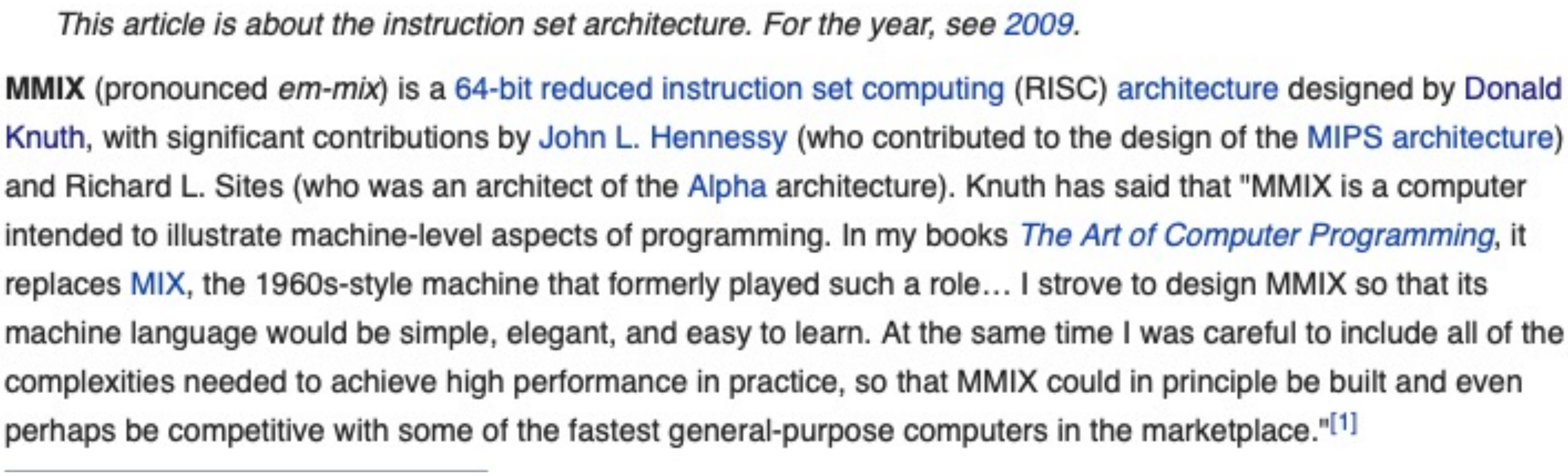
MMIX is a 64-bit reduced instruction set computing (RISC) architecture designed by Donald Knuth, with significant contributions by John L. Hennessy and Richard L. Sites. Knuth has said that "MMIX is a computer intended to illustrate machine-level aspects of programming. In my books *The Art of*

= 64-bit RISC Update to original CISC **MIX**



MMIX

Designer	Donald Knuth
Bits	64-bit
Design	RISC
Encoding	Fixed
Branching	Condition Code
Endianness	Big
Open	Yes, and royalty free
Registers	
32 special-purpose registers	
General purpose	256



MMIX

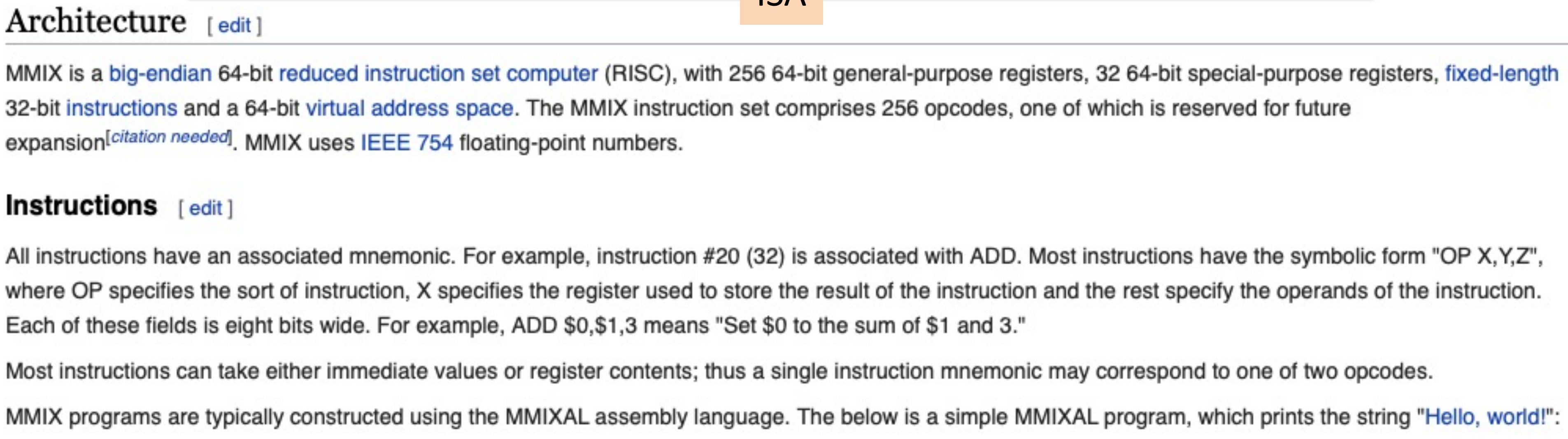
From Wikipedia, the free encyclopedia

This article is about the instruction set architecture. For the year, see 2009.

MMIX (pronounced *em-mix*) is a 64-bit reduced instruction set computing (RISC) architecture designed by Donald Knuth, with significant contributions by John L. Hennessy (who contributed to the design of the MIPS architecture) and Richard L. Sites (who was an architect of the Alpha architecture). Knuth has said that "MMIX is a computer intended to illustrate machine-level aspects of programming. In my books *The Art of Computer Programming*, it replaces MIX, the 1960s-style machine that formerly played such a role… I strove to design MMIX so that its machine language would be simple, elegant, and easy to learn. At the same time I was careful to include all of the complexities needed to achieve high performance in practice, so that MMIX could in principle be built and even perhaps be competitive with some of the fastest general-purpose computers in the marketplace."[1]

MMIX

ISA



Architecture [edit]

MMIX is a big-endian 64-bit reduced instruction set computer (RISC), with 256 64-bit general-purpose registers, 32 64-bit special-purpose registers, fixed-length 32-bit instructions and a 64-bit virtual address space. The MMIX instruction set comprises 256 opcodes, one of which is reserved for future expansion[citation needed]. MMIX uses IEEE 754 floating-point numbers.

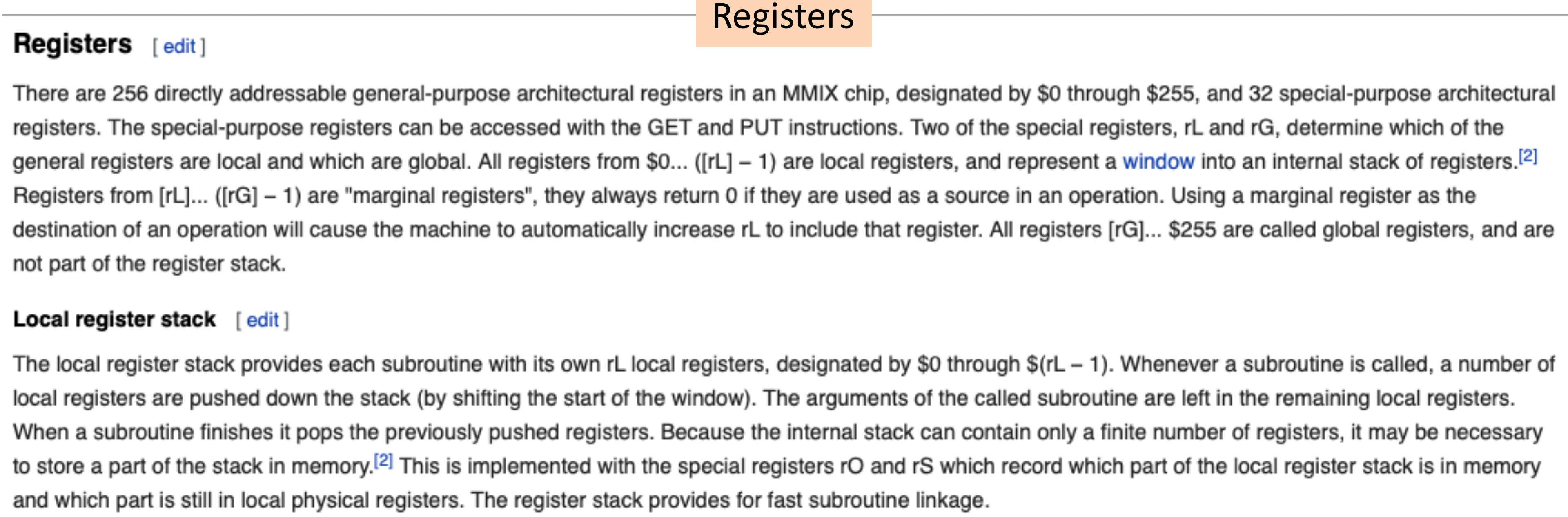
Instructions [edit]

All instructions have an associated mnemonic. For example, instruction #20 (32) is associated with ADD. Most instructions have the symbolic form "OP X,Y,Z", where OP specifies the sort of instruction, X specifies the register used to store the result of the instruction and the rest specify the operands of the instruction. Each of these fields is eight bits wide. For example, ADD $0,$1,3 means "Set $0 to the sum of $1 and 3."

Most instructions can take either immediate values or register contents; thus a single instruction mnemonic may correspond to one of two opcodes.

MMIX programs are typically constructed using the MMIXAL assembly language. The below is a simple MMIXAL program, which prints the string "Hello, world!":

Registers



Registers [edit]

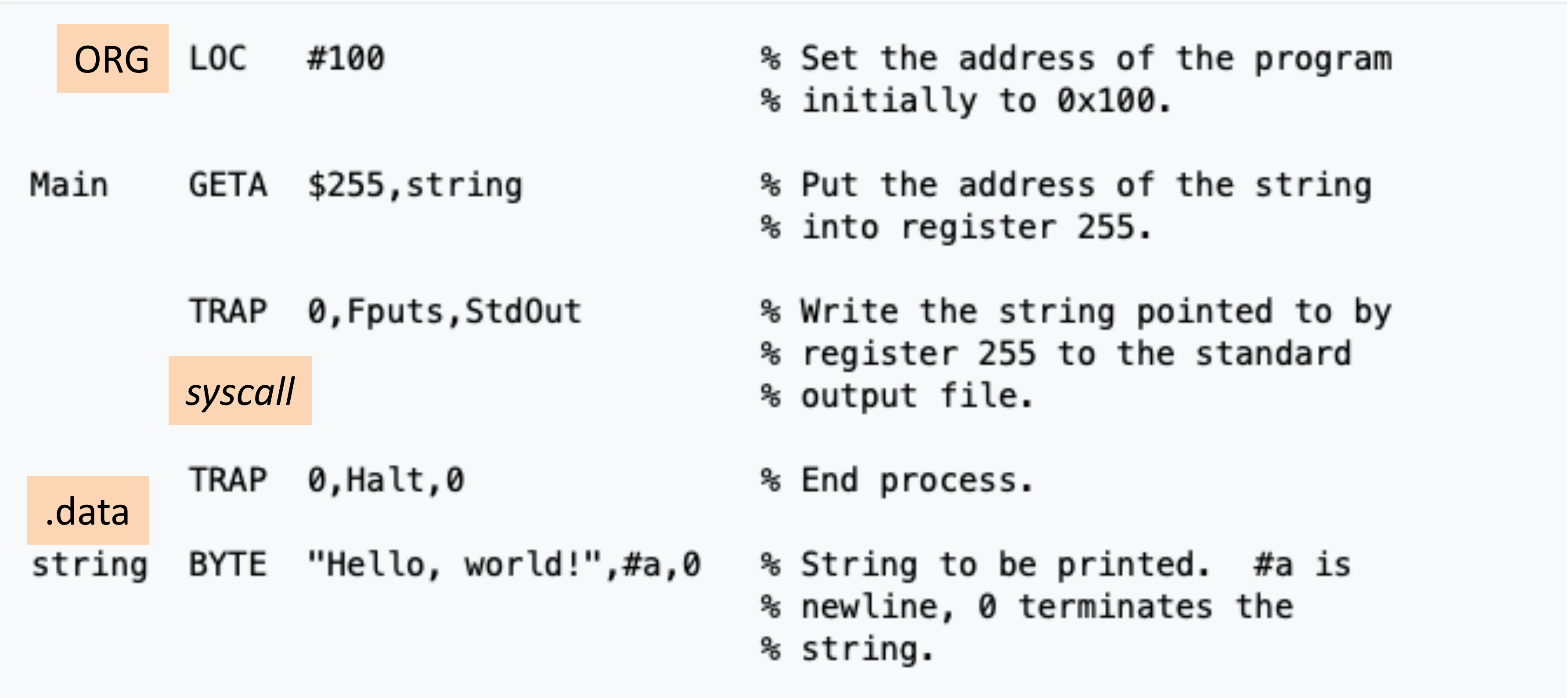
There are 256 directly addressable general-purpose architectural registers in an MMIX chip, designated by $0 through $255, and 32 special-purpose architectural registers. The special-purpose registers can be accessed with the GET and PUT instructions. Two of the special registers, rL and rG, determine which of the general registers are local and which are global. All registers from $0... ([rL] – 1) are local registers, and represent a window into an internal stack of registers.[2] Registers from [rL]... ([rG] – 1) are "marginal registers", they always return 0 if they are used as a source in an operation. Using a marginal register as the destination of an operation will cause the machine to automatically increase rL to include that register. All registers [rG]... $255 are called global registers, and are not part of the register stack.

Local register stack [edit]

The local register stack provides each subroutine with its own rL local registers, designated by $0 through $(rL – 1). Whenever a subroutine is called, a number of local registers are pushed down the stack (by shifting the start of the window). The arguments of the called subroutine are left in the remaining local registers. When a subroutine finishes it pops the previously pushed registers. Because the internal stack can contain only a finite number of registers, it may be necessary to store a part of the stack in memory.[2] This is implemented with the special registers rO and rS which record which part of the local register stack is in memory and which part is still in local physical registers. The register stack provides for fast subroutine linkage.

MMIX

Hello World!



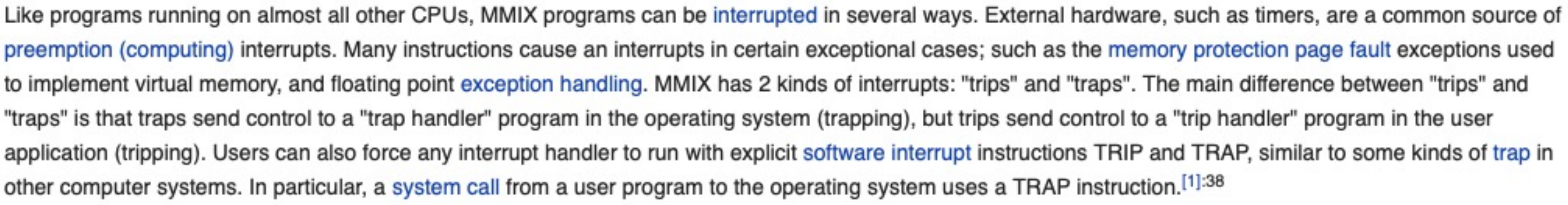
```
         LOC    #100                   % Set the address of the program
                                       % initially to 0x100.

Main     GETA   $255,string            % Put the address of the string
                                       % into register 255.

         TRAP   0,Fputs,StdOut         % Write the string pointed to by
                                       % register 255 to the standard
                                       % output file.

         TRAP   0,Halt,0               % End process.

string   BYTE   "Hello, world!",#a,0   % String to be printed.  #a is
                                       % newline, 0 terminates the
                                       % string.
```

ORG

syscall

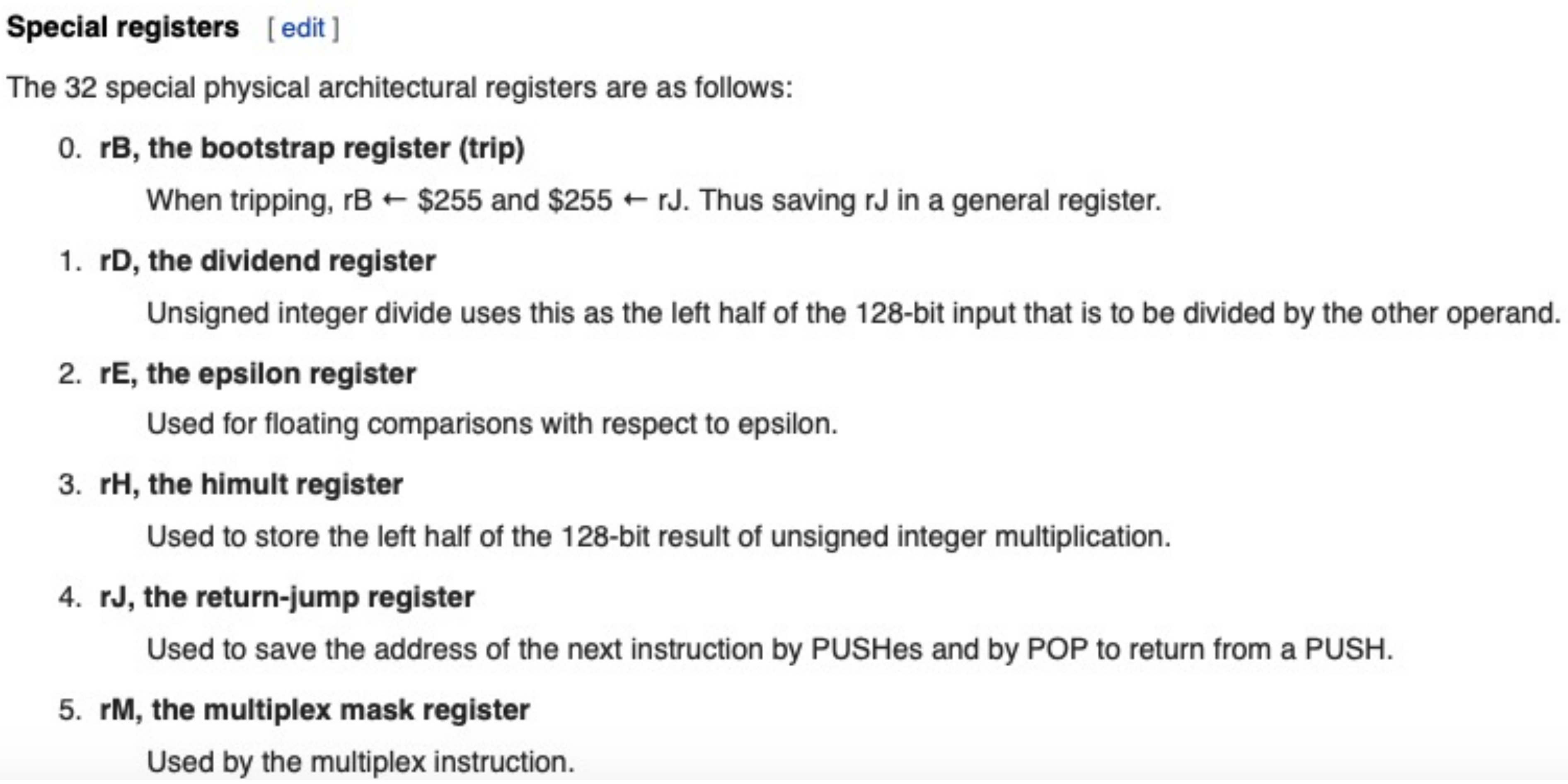
.data

Trips, Traps & Interrupts



Like programs running on almost all other CPUs, MMIX programs can be interrupted in several ways. External hardware, such as timers, are a common source of preemption (computing) interrupts. Many instructions cause an interrupts in certain exceptional cases; such as the memory protection page fault exceptions used to implement virtual memory, and floating point exception handling. MMIX has 2 kinds of interrupts: "trips" and "traps". The main difference between "trips" and "traps" is that traps send control to a "trap handler" program in the operating system (trapping), but trips send control to a "trip handler" program in the user application (tripping). Users can also force any interrupt handler to run with explicit software interrupt instructions TRIP and TRAP, similar to some kinds of trap in other computer systems. In particular, a system call from a user program to the operating system uses a TRAP instruction.[1]:38

MMIX



Special registers [edit]

The 32 special physical architectural registers are as follows:

0. **rB, the bootstrap register (trip)**

   When tripping, rB ← $255 and $255 ← rJ. Thus saving rJ in a general register.

1. **rD, the dividend register**

   Unsigned integer divide uses this as the left half of the 128-bit input that is to be divided by the other operand.

2. **rE, the epsilon register**

   Used for floating comparisons with respect to epsilon.

3. **rH, the himult register**

   Used to store the left half of the 128-bit result of unsigned integer multiplication.

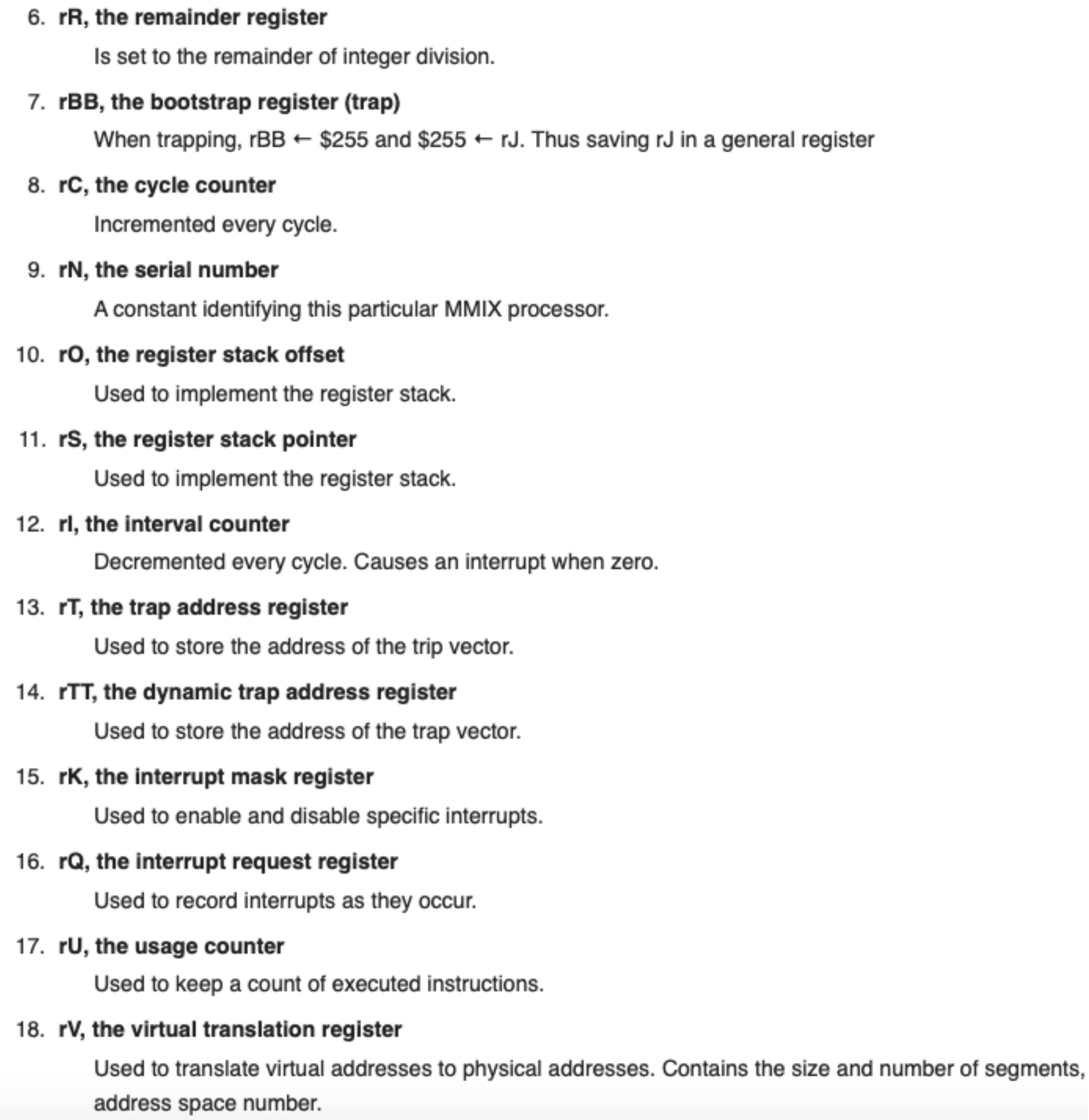
4. **rJ, the return-jump register**

   Used to save the address of the next instruction by PUSHes and by POP to return from a PUSH.

5. **rM, the multiplex mask register**

   Used by the multiplex instruction.

MMIX



6. **rR, the remainder register**

   Is set to the remainder of integer division.

7. **rBB, the bootstrap register (trap)**

   When trapping, rBB ← $255 and $255 ← rJ. Thus saving rJ in a general register

8. **rC, the cycle counter**

   Incremented every cycle.

9. **rN, the serial number**

   A constant identifying this particular MMIX processor.

10. **rO, the register stack offset**

    Used to implement the register stack.

11. **rS, the register stack pointer**

    Used to implement the register stack.

12. **rI, the interval counter**

    Decremented every cycle. Causes an interrupt when zero.

13. **rT, the trap address register**

    Used to store the address of the trip vector.

14. **rTT, the dynamic trap address register**

    Used to store the address of the trap vector.

15. **rK, the interrupt mask register**

    Used to enable and disable specific interrupts.

16. **rQ, the interrupt request register**

    Used to record interrupts as they occur.

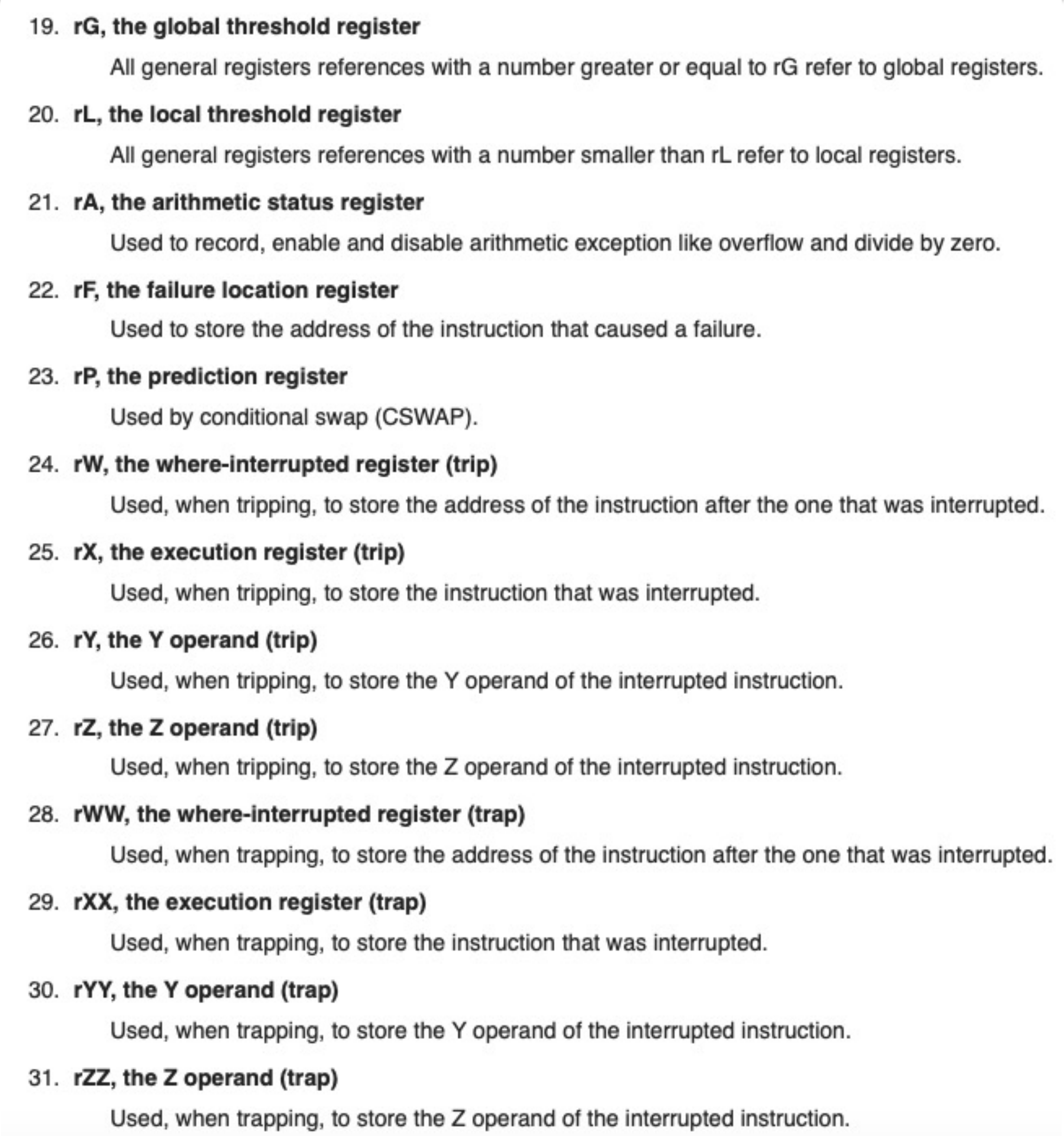
17. **rU, the usage counter**

    Used to keep a count of executed instructions.

18. **rV, the virtual translation register**

    Used to translate virtual addresses to physical addresses. Contains the size and number of segments, address space number.

MMIX



19. **rG, the global threshold register**

    All general registers references with a number greater or equal to rG refer to global registers.

20. **rL, the local threshold register**

    All general registers references with a number smaller than rL refer to local registers.

21. **rA, the arithmetic status register**

    Used to record, enable and disable arithmetic exception like overflow and divide by zero.

22. **rF, the failure location register**

    Used to store the address of the instruction that caused a failure.

23. **rP, the prediction register**

    Used by conditional swap (CSWAP).

24. **rW, the where-interrupted register (trip)**

    Used, when tripping, to store the address of the instruction after the one that was interrupted.

25. **rX, the execution register (trip)**

    Used, when tripping, to store the instruction that was interrupted.

26. **rY, the Y operand (trip)**

    Used, when tripping, to store the Y operand of the interrupted instruction.

27. **rZ, the Z operand (trip)**

    Used, when tripping, to store the Z operand of the interrupted instruction.

28. **rWW, the where-interrupted register (trap)**

    Used, when trapping, to store the address of the instruction after the one that was interrupted.

29. **rXX, the execution register (trap)**

    Used, when trapping, to store the instruction that was interrupted.

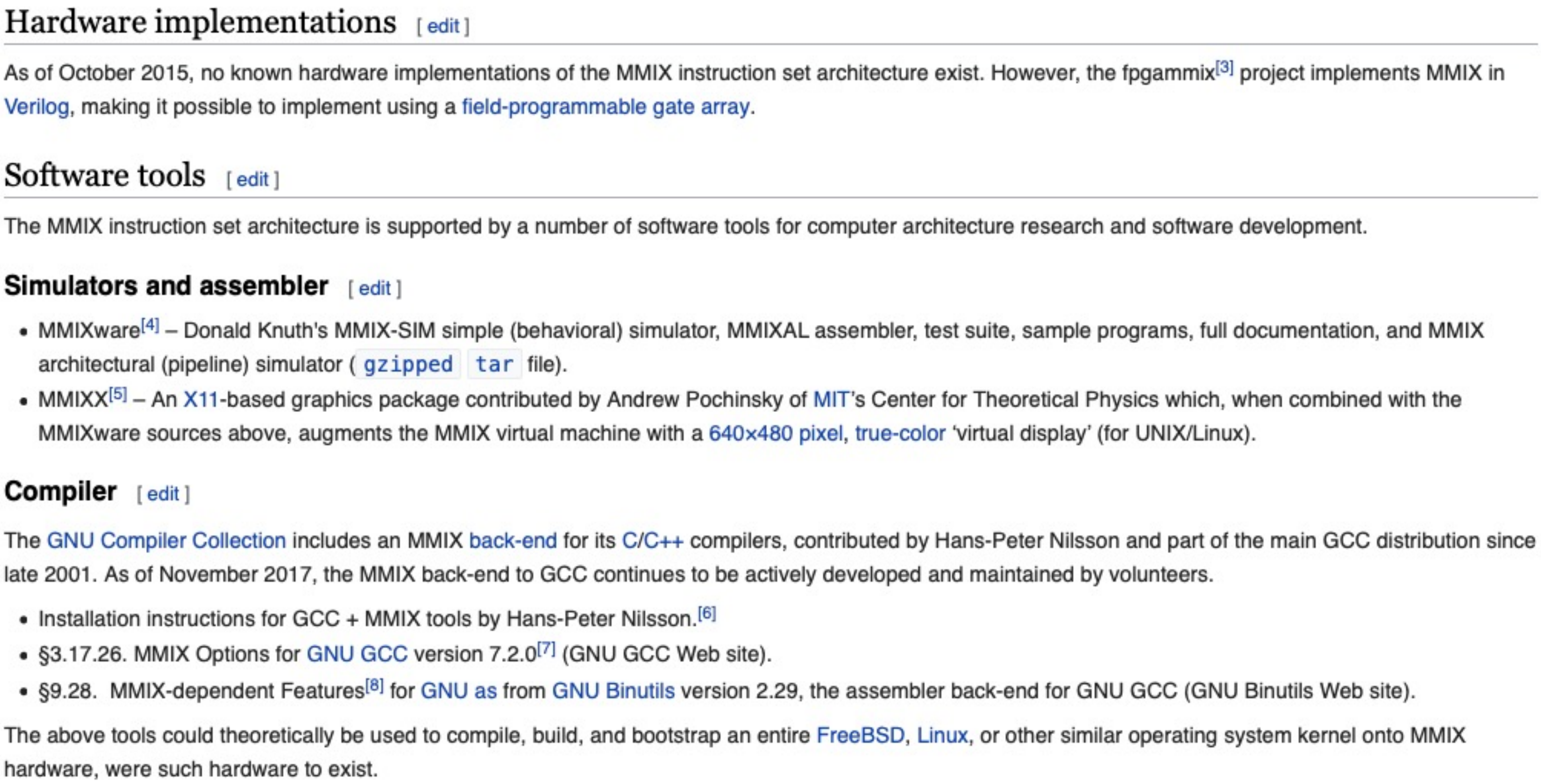
30. **rYY, the Y operand (trap)**

    Used, when trapping, to store the Y operand of the interrupted instruction.

31. **rZZ, the Z operand (trap)**

    Used, when trapping, to store the Z operand of the interrupted instruction.

MMIX



Hardware implementations [edit]

As of October 2015, no known hardware implementations of the MMIX instruction set architecture exist. However, the fpgammix[3] project implements MMIX in Verilog, making it possible to implement using a field-programmable gate array.

Software tools [edit]

The MMIX instruction set architecture is supported by a number of software tools for computer architecture research and software development.

Simulators and assembler [edit]

- MMIXware[4] – Donald Knuth's MMIX-SIM simple (behavioral) simulator, MMIXAL assembler, test suite, sample programs, full documentation, and MMIX architectural (pipeline) simulator (`gzipped` `tar` file).
- MMIXX[5] – An X11-based graphics package contributed by Andrew Pochinsky of MIT's Center for Theoretical Physics which, when combined with the MMIXware sources above, augments the MMIX virtual machine with a 640×480 pixel, true-color 'virtual display' (for UNIX/Linux).

Compiler [edit]

The GNU Compiler Collection includes an MMIX back-end for its C/C++ compilers, contributed by Hans-Peter Nilsson and part of the main GCC distribution since late 2001. As of November 2017, the MMIX back-end to GCC continues to be actively developed and maintained by volunteers.

- Installation instructions for GCC + MMIX tools by Hans-Peter Nilsson.[6]
- §3.17.26. MMIX Options for GNU GCC version 7.2.0[7] (GNU GCC Web site).
- §9.28. MMIX-dependent Features[8] for GNU as from GNU Binutils version 2.29, the assembler back-end for GNU GCC (GNU Binutils Web site).

The above tools could theoretically be used to compile, build, and bootstrap an entire FreeBSD, Linux, or other similar operating system kernel onto MMIX hardware, were such hardware to exist.