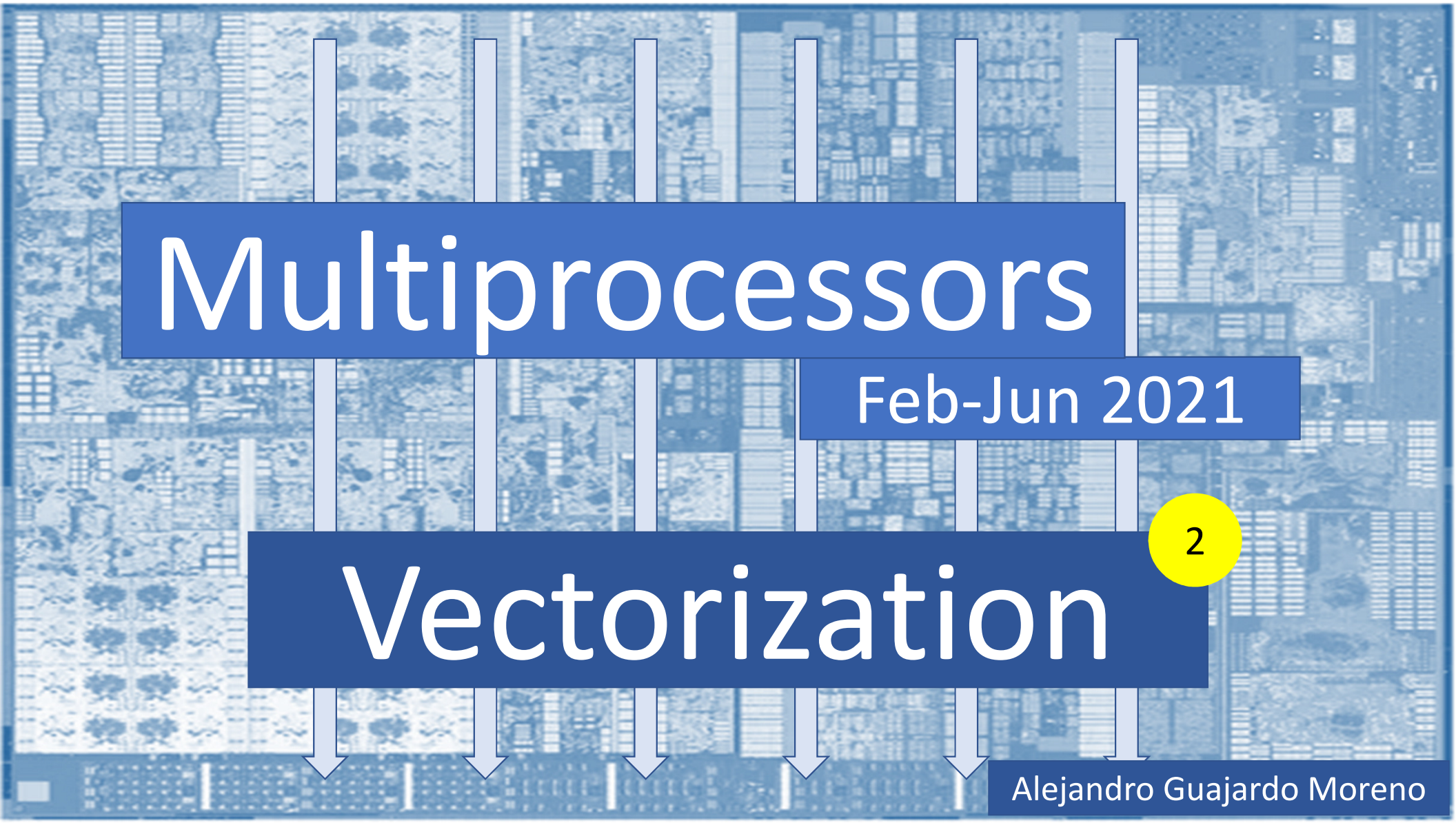
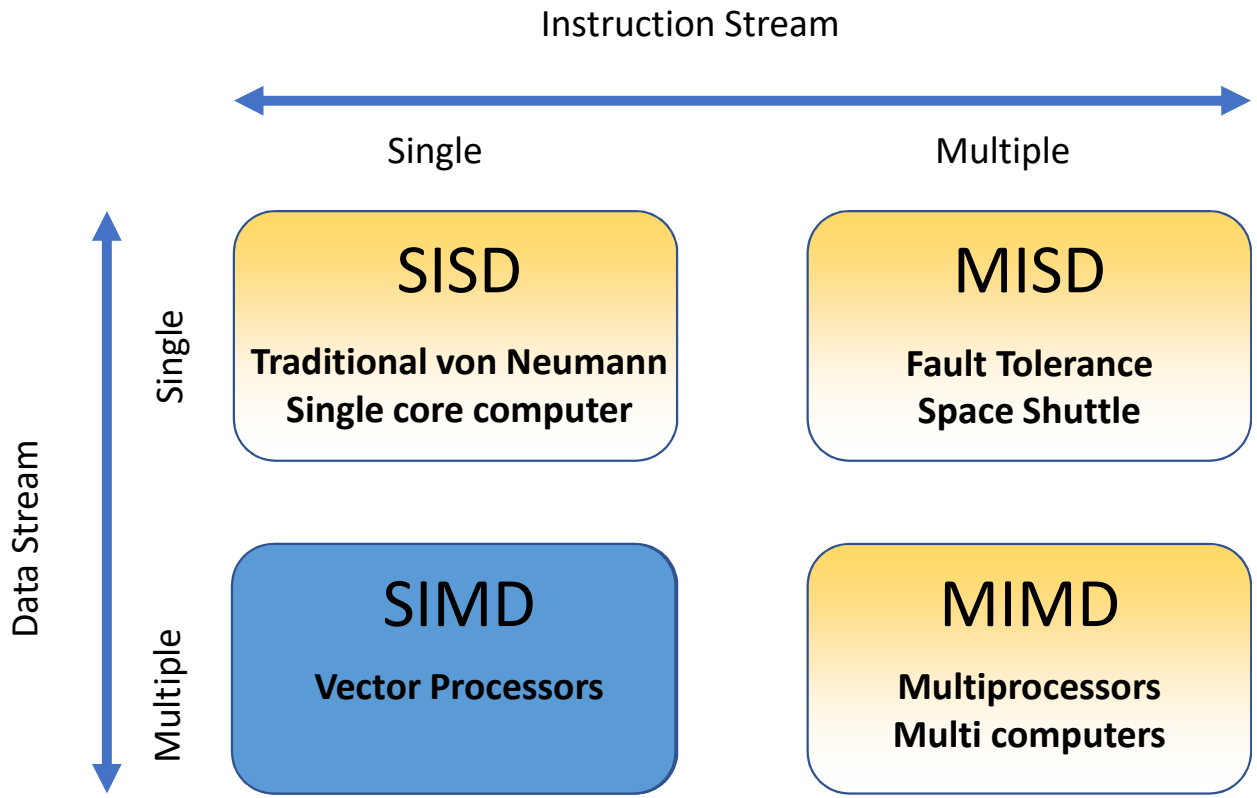

Multiprocessors
Feb-Jun 2021
2
Vectorization
Alejandro Guajardo Moreno

Flynn's Taxonomy



Vectorization Foucs: **SIMD**

SIMD in the real world?

Can you think of an example where SIMD can be used?

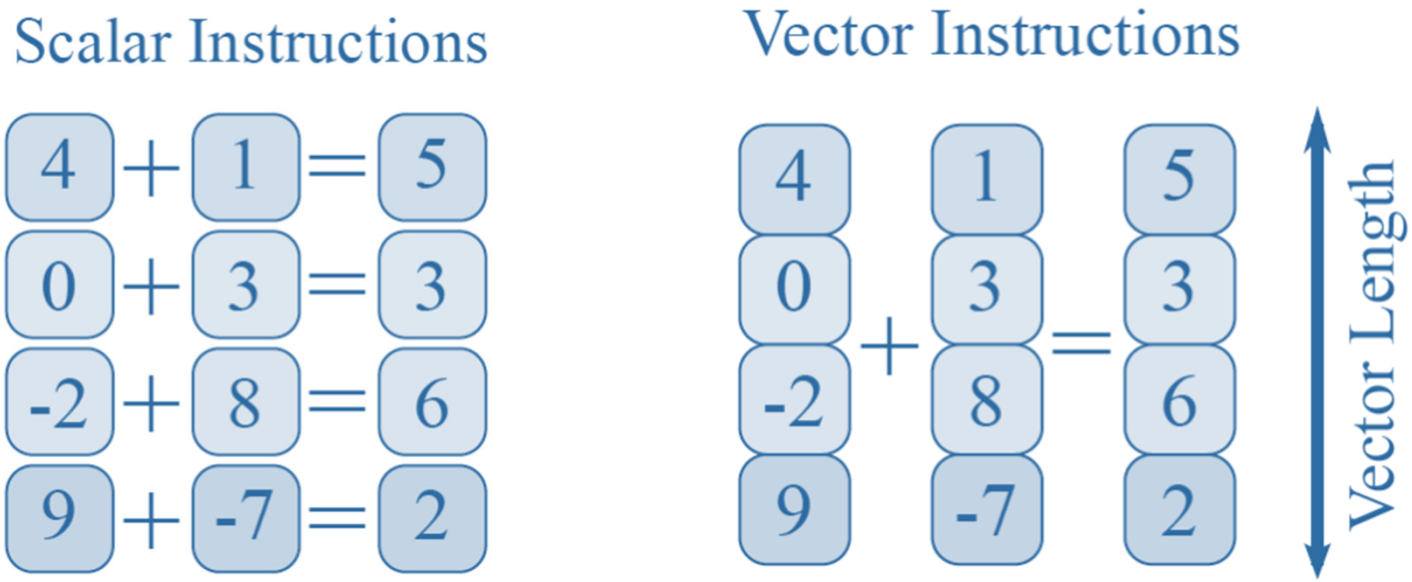
Hint: SAME Instruction over lots and lots of Data.

- Image Processing
- 3D graphics
- Rendering
- Criptography
- Compression
- Video Processing
  - conversion between video standards and frame rates (NTSC to/from PAL, NTSC to/from HDTV formats)
  - deinterlacing
  - image noise reduction
  - adaptive video compression
  - image enhancement
- And many many MORE!

Vectorization comparison with mundane objects

Vectorization!

What is VECTORIZATION and its relation to SIMD?

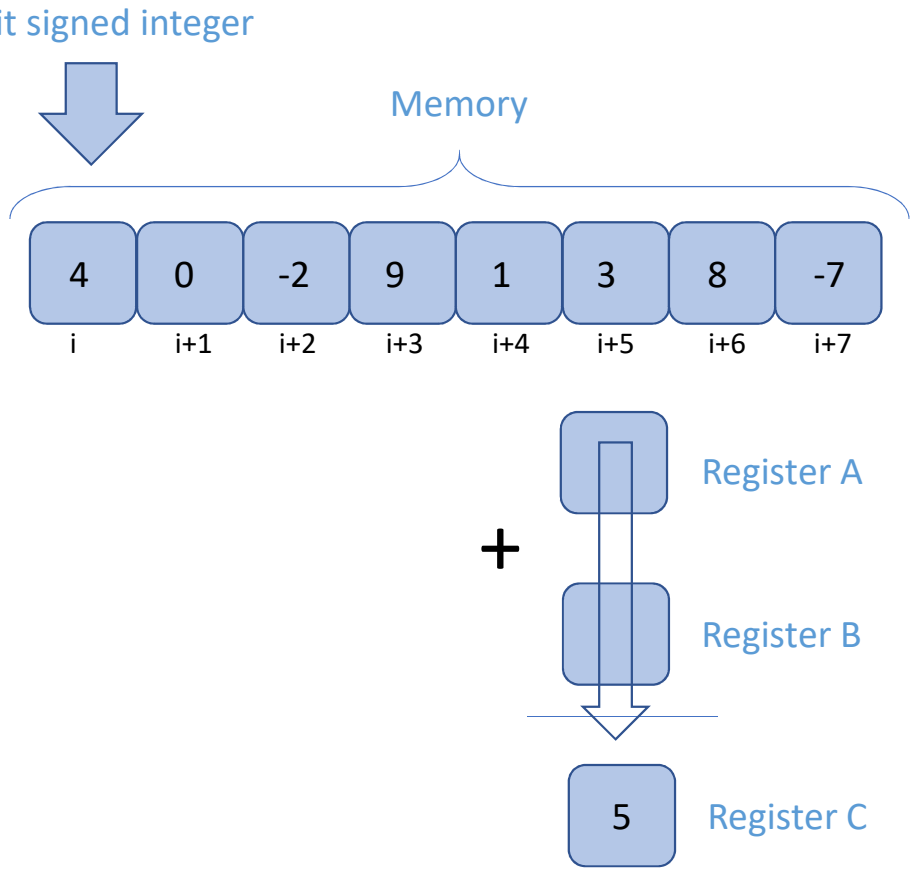


Example… Scalar instructions…

Scalar Instructions

4 + 1 = 5
0 + 3 = 3
-2 + 8 = 6
9 + -7 = 2

1. Load (A)
2. Load (B)
3. C = Sum (A+B)
4. Store (C)



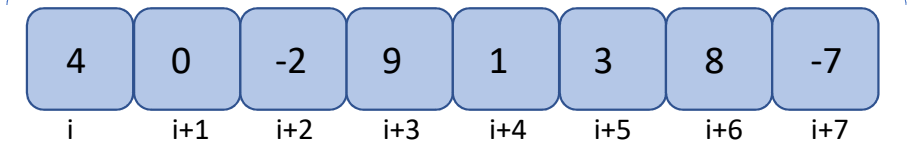
Example… Scalar instructions…

Scalar Instructions

4 + 1 = 5
0 + 3 = 3
-2 + 8 = 6
9 + -7 = 2

32 bit signed integer

Memory

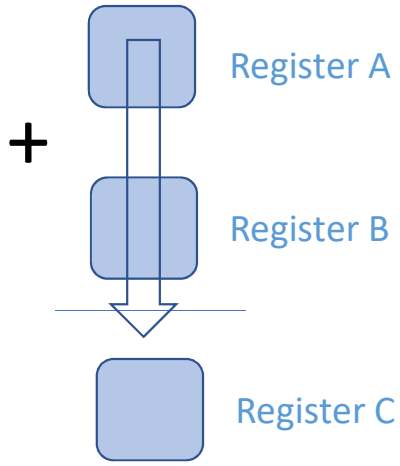


1. Load (A)
2. Load (B)
3. C = Sum (A+B)
4. Store (C)

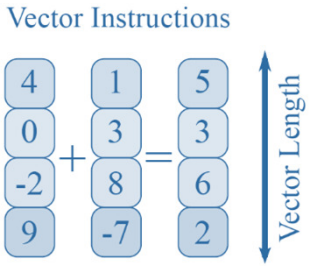
5 - 8. Repeat with 0 + 3 = 3

9 - 12. Repeat with -2 + 8 = 6

13 - 16. Repeat with 9 + -7 = 2

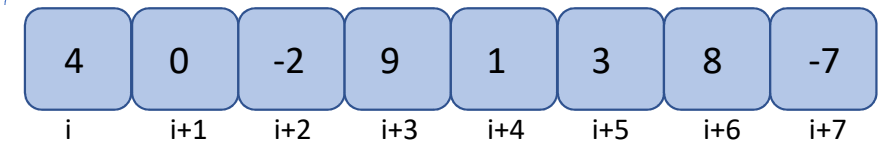


Example… Now using VECTOR instructions…

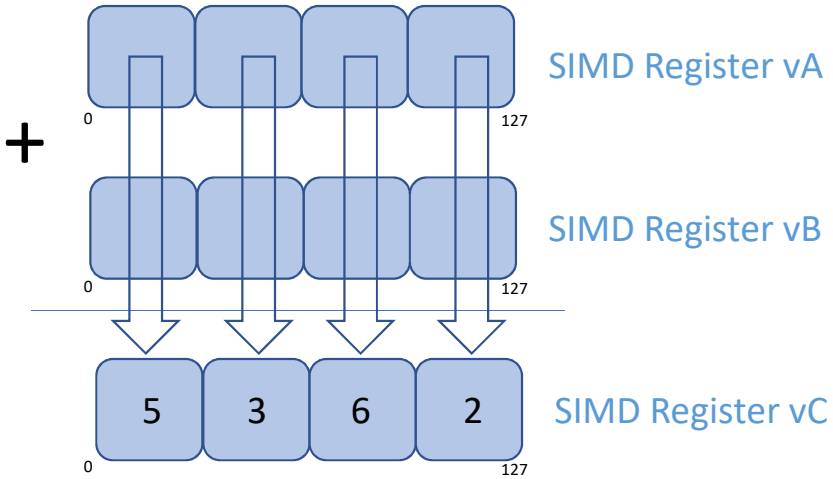


32 bit signed integer

Memory



1. Load (vA)
2. Load (vB)
3. vC = Sum (vA+vB)
4. Store (vC)



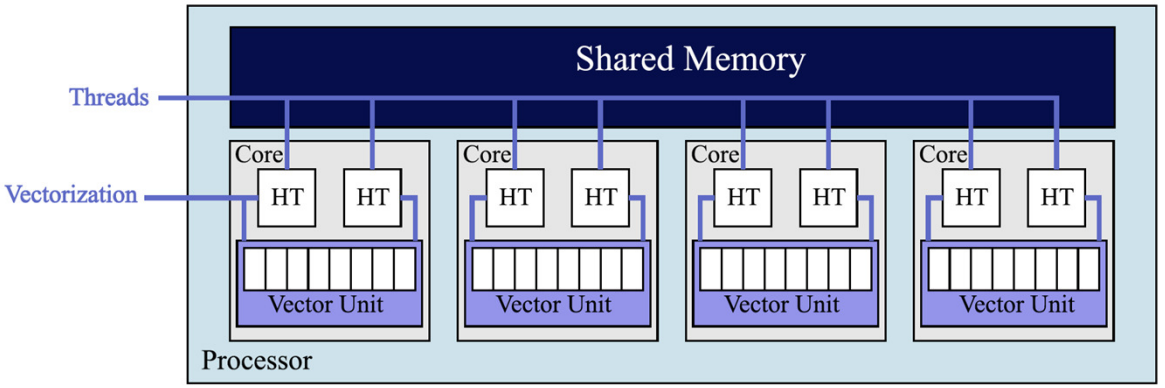
What did you notice?

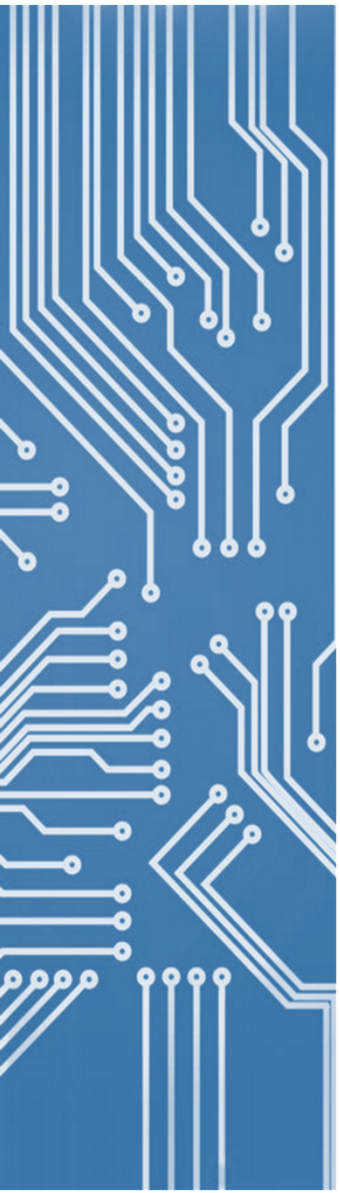
Memory?

- Elements are located next to each other. **Contiguous.**
- Elements are normally and preferable **aligned.**

Registers?

- “Special Looooong Registers”





Components needed for a processor to use vectors.

- With VECTOR Instruction Set Extensions.
- With Special Registers, transistors & logic.

First INTEL® VECTOR Instructions

- 1996
- CISC
- Superscalar
- IA-32 ← 32-bit version of the x86 instruction set architecture

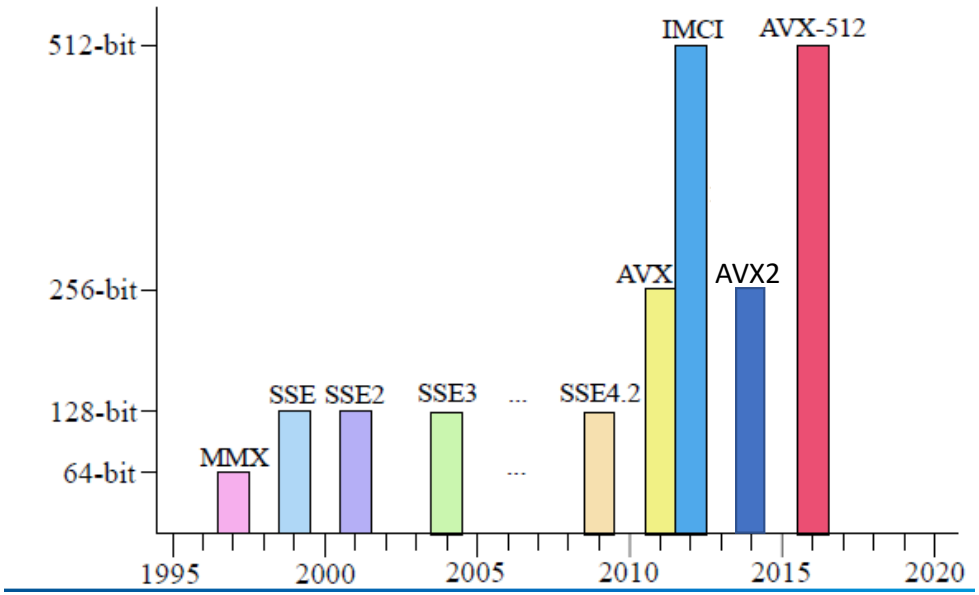
- MMX = SIMD (64 bit)

MMX = Multimedia Extensions

And after…

MMX: Multimedia Extension
SSE: Streaming SIMD Extension
AVX: Advanced Vector Extension

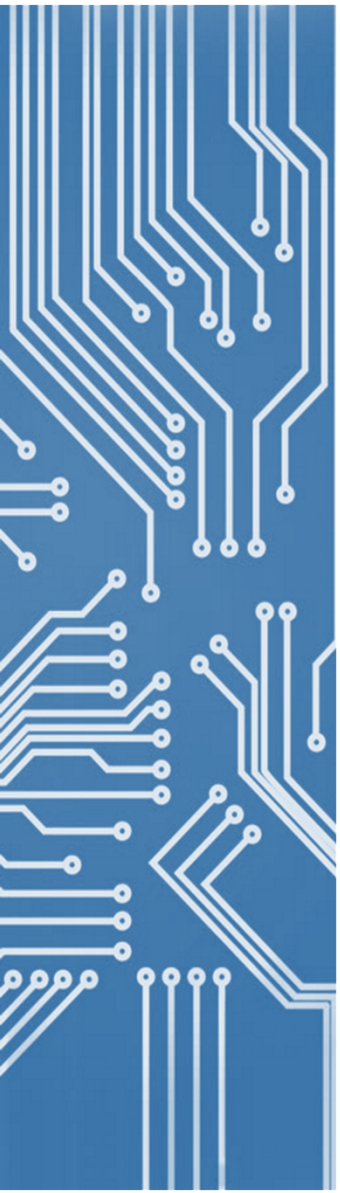
	x86-16	8086
		286
x86-32		386
		486
		Pentium
	MMX	Pentium MMX
	SSE	Pentium III
	SSE2	Pentium 4
	SSE3	Pentium 4E
x86-64/em64t		Pentium 4F
		Core 2 Duo
	SSE4	Penryn
		Core i7 (Nehalem)
	AVX	Sandy Bridge
	AVX2	Haswell
	AVX-512	Skylake X



And after…

Vendor	Name		ν-way	Precision	Introduced with
Intel	SSE		4-way	single	Pentium III
	SSE2	+	2-way	double	Pentium 4
	SSE3				Pentium 4 (Prescott)
	SSSE3				Core Duo
	SSE4				Core2 Extreme (Penryn)
	AVX		8-way	single	Core i7 (Sandybridge)
			4-way	double	
Intel	IPF		2-way	single	Itanium
Intel	LRB		16-way	single	Larrabee
			8-way	double	
AMD	3DNow!		2-way	single	K6
	Enhanced 3DNow!				K7
	3DNow! Professional	+	4-way	single	Athlon XP
	AMD64	+	2-way	double	Opteron
Motorola	AltiVec		4-way	single	MPC 7400 G4
IBM	VMX		4-way	single	PowerPC 970 G5
	SPU	+	2-way	double	Cell BE
IBM	Double FPU		2-way	double	PowerPC 440 FP2

Source: (CS263-2300 ETH)



Teams 3

Use https://ark.intel.com o https://www.amd.com/en/products/specifications to answer the following:

- What SIMD extensions does the i5-8350U processor support?

  Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2

- What is the targeted segment for i5-8350U processor?

  Mobile

- Does the Intel Itanium Processor 9760 has any kind of SIMD extensions?

  No extensions*

- Intel Xeon Platinum 9282, second generation of Intel Xeon Scalable Processors support for SIMD extensions?

  Intel® AVX-512

- Is there an Intel® ATOM processor that supports AVX, AV2 o AVX-512?

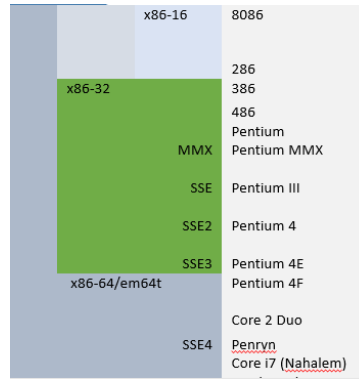
  No

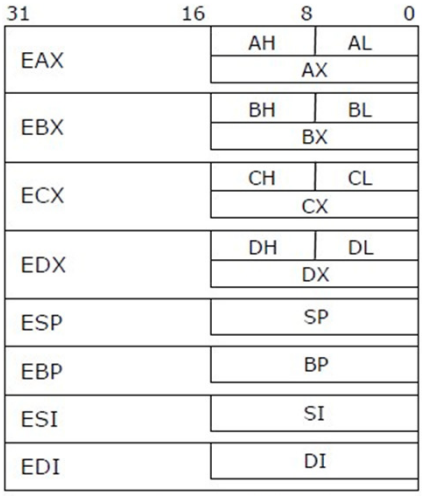
- AMD Ryzen™ Threadripper™ 1950X Processor supports AVX2 and/or AVX-512?

  AVX2

IA-32 Architecture

- Datatypes: BYTE, WORD, DWORD

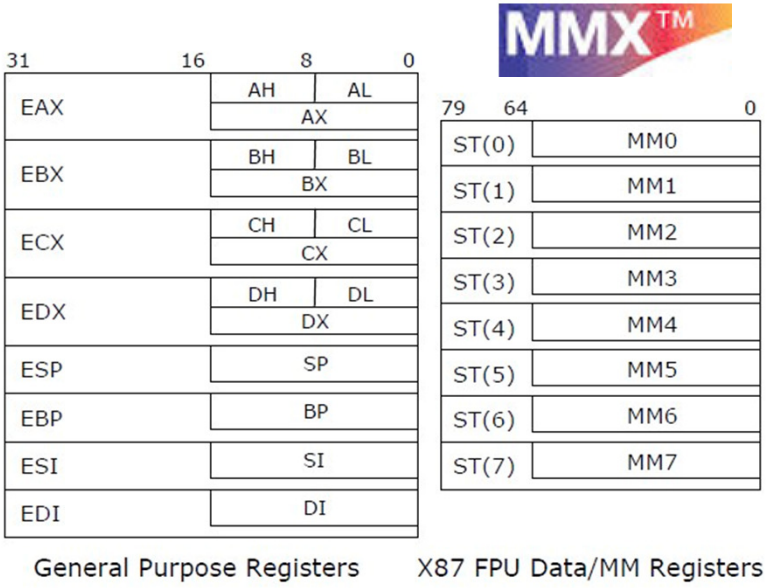


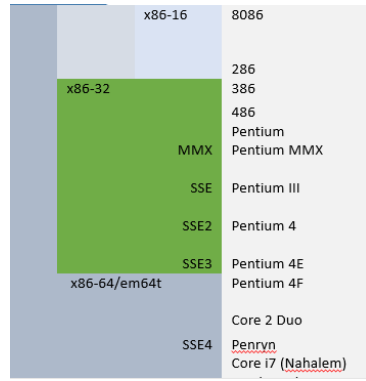


General Purpose Registers

IA-32 Architecture

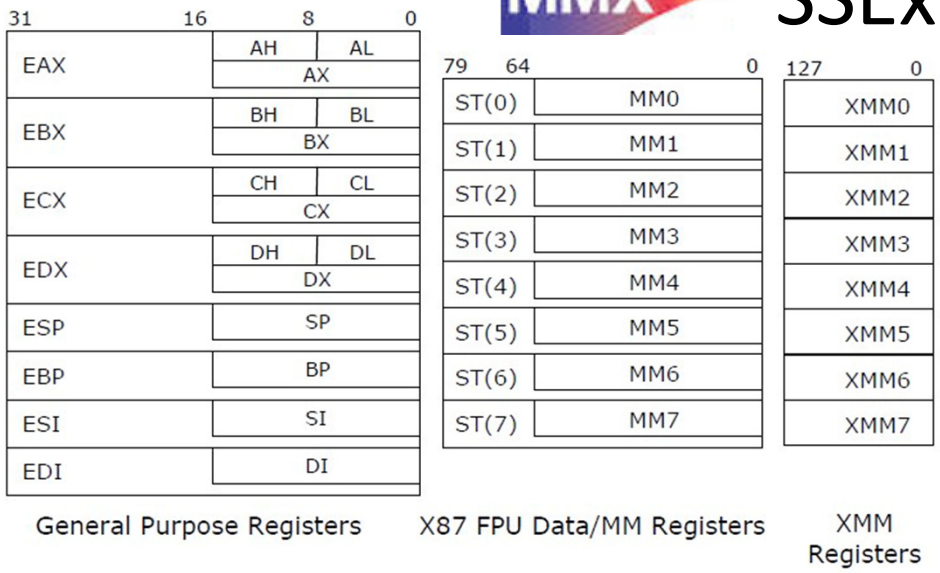
- Datatypes: BYTE, WORD, DWORD y **QWORD**
- The ST(x) registers are accessible as a stack by the FPU, while the MMX registers are accessible by specific instructions.
- For extended precision (80 bits) floating point operations the vectorization is impossible.

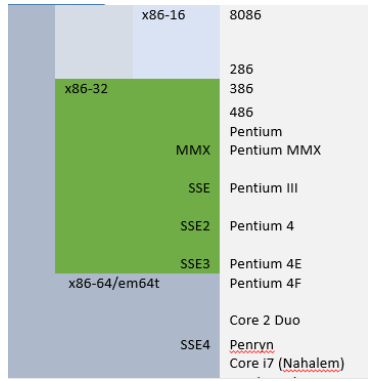


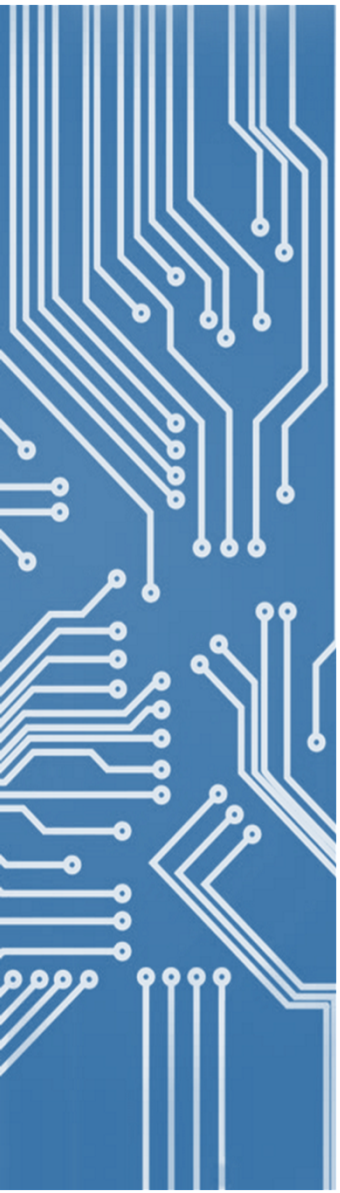


IA-32 Architecture

- Datatypes: BYTE, WORD, DWORD y QWORD
- The ST(x) registers are accessible as a stack by the FPU, while the MMX registers are accessible by specific instructions.
- For extended precision (80 bits) floating point operations the vectorization is impossible.







IA-32 Architecture

When data is vectorial, the processor may use any of two options:

- To use the FPU (x87)
- To use XMM registers

```
float x[4], y[4], z[4];
int i;
…
for (i=0;i<4;i++)
    z[i]=x[i]*y[i];
```

Assembly for Intel SSE

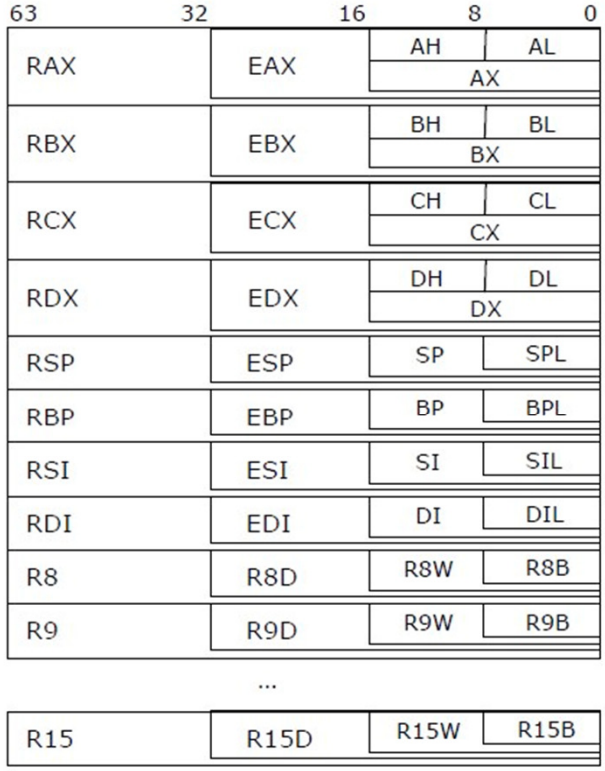
```
movaps   x, %xmm0
mulps    y, %xmm0
movaps   %xmm0, z
```

Assembly for x87

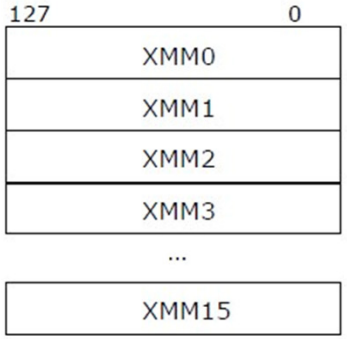
```
flds   x
fmuls  y
flds   4+x
fmuls  4+y
flds   8+x
fmuls  8+y
flds   12+x
fmuls  12+y
fxch   %st(3)
fstps  z
fxch   %st(1)
fstps  4+z
fstps  8+z
fstps  12+z
```

Which one is better in terms of performance/IPC?

x86-64 Architecture

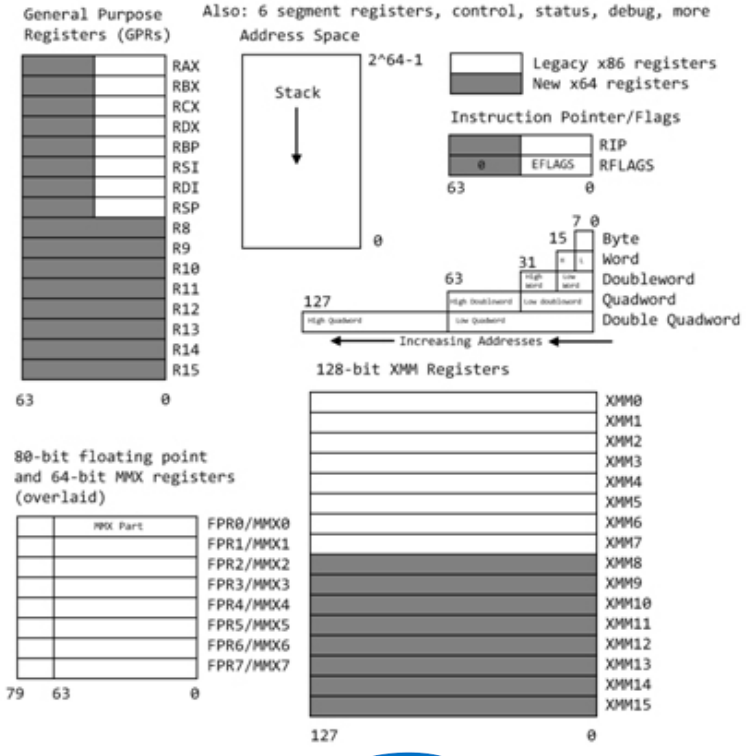


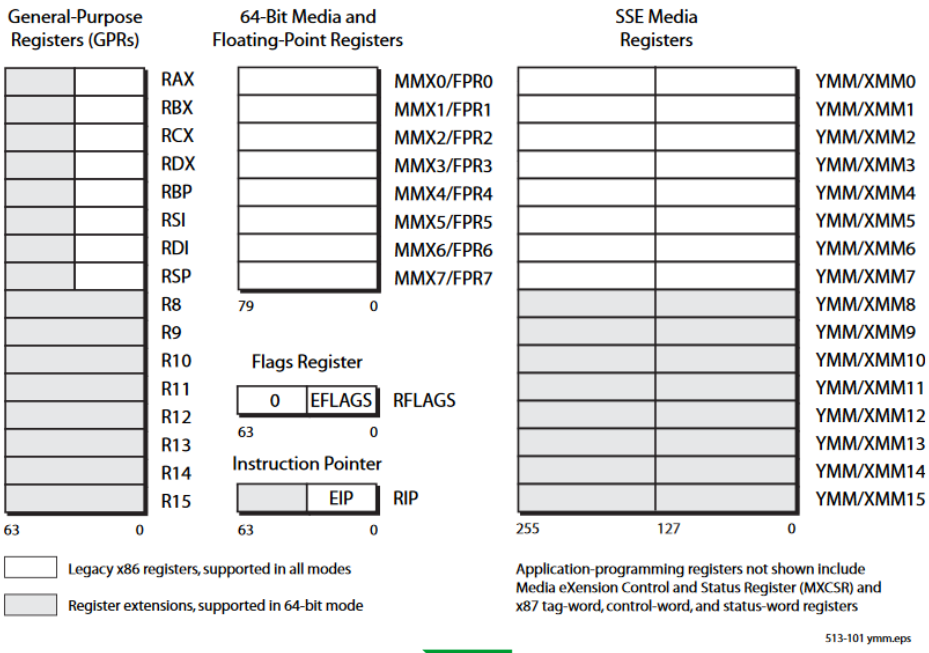
General Purpose Registers



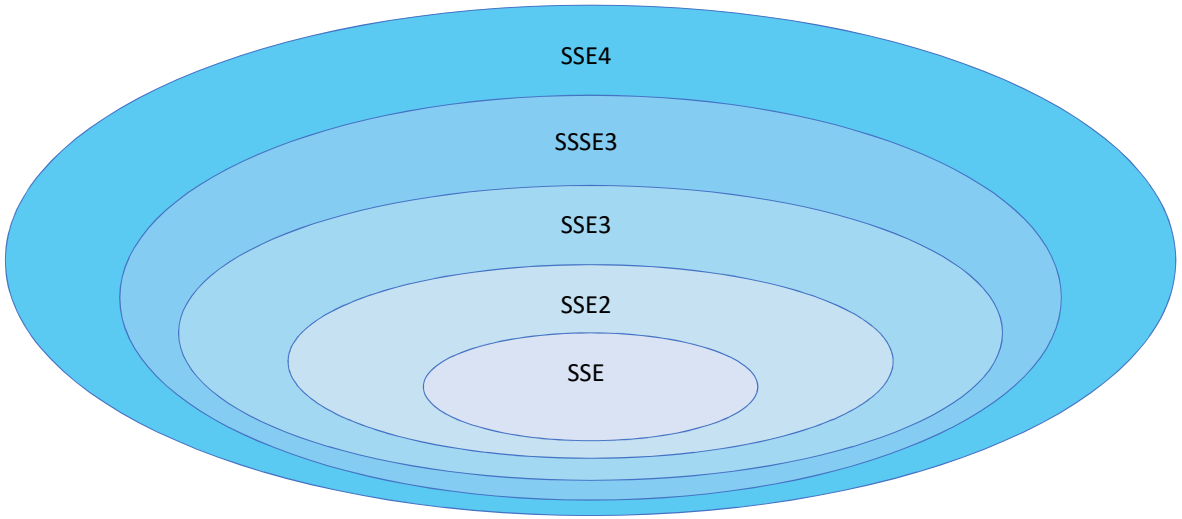
XMM Registers

Intel 64 / AMD x86-64 Full Architectures





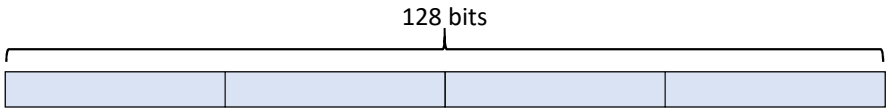
SSE Family



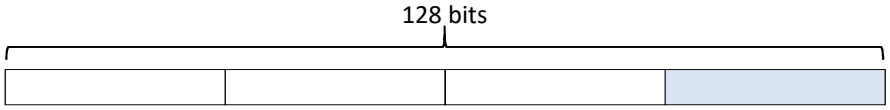
- ✓Each version incorporate new instructions.
- ✓Backward compatibility is guaranteed.
- ✓Each core can execute SSE instructions independently.

Data extensions (SSE)

- Floating point vectors
  - 4-way single precision (DWORD)

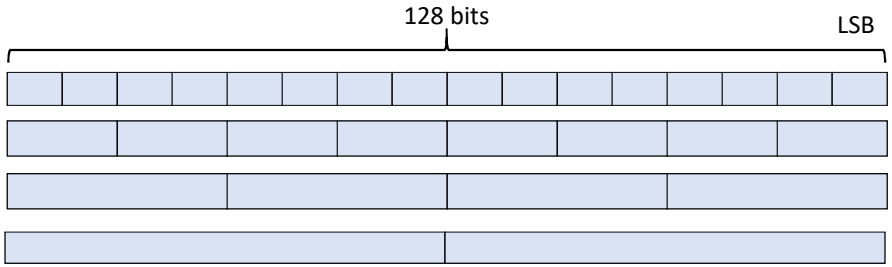


- Floating point scalars
  - Single precision (DWORD)

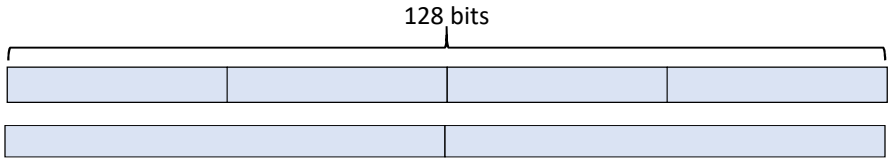


Data extensions (SSE2)

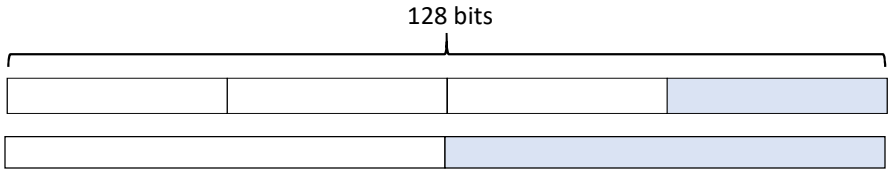
- Integer vectors
  - 16-way bytes (BYTE)
  - 8-way 2 bytes (WORD)
  - 4-way 4 bytes (DWORD)
  - 2-way 8 bytes (QWORD)

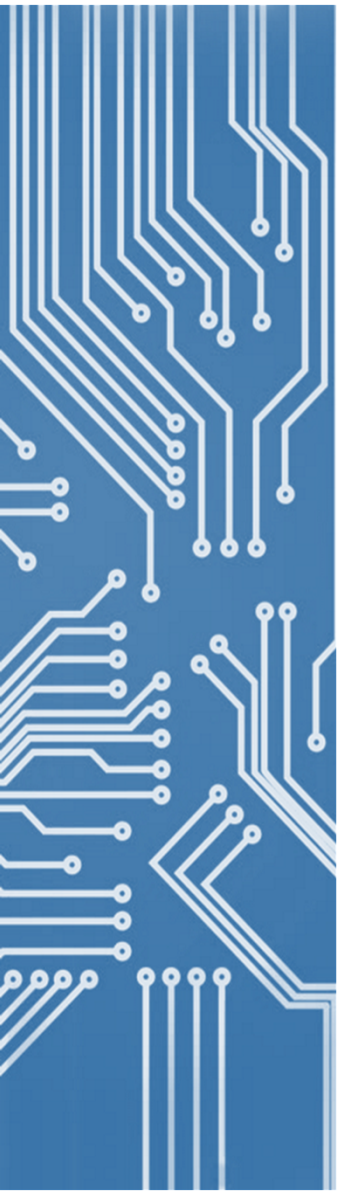


- Floating point vectors
  - 4-way single precision (DWORD)
  - 2-way double precision (QWORD)



- Floating point scalars
  - Single precision (DWORD)
  - Double precision (QWORD)





Data extensions (SSE3)

Focus:

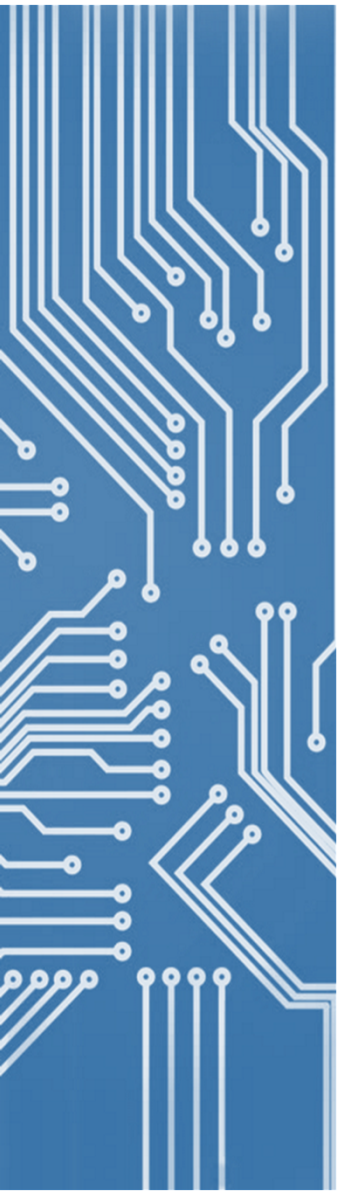
Horizontal work **within** the register.

HADDPD — (Horizontal-Add-Packed-Double)
HADDPS (Horizontal-Add-Packed-Single)
HSUBPD — (Horizontal-Subtract-Packed-Double)
HSUBPS — (Horizontal-Subtract-Packed-Single)

Also:

3D and Digital Signal Processing operations.

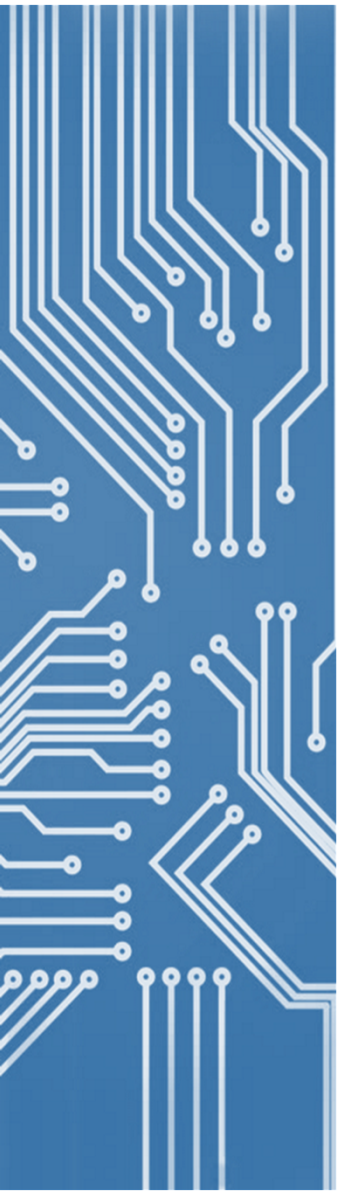
Convert FP to Int avoiding pipeline stall.



Data extensions (SSSE3)

Focus:

- Integers
- Use of either MMX or XMM registers.



Data extensions (SSE4)

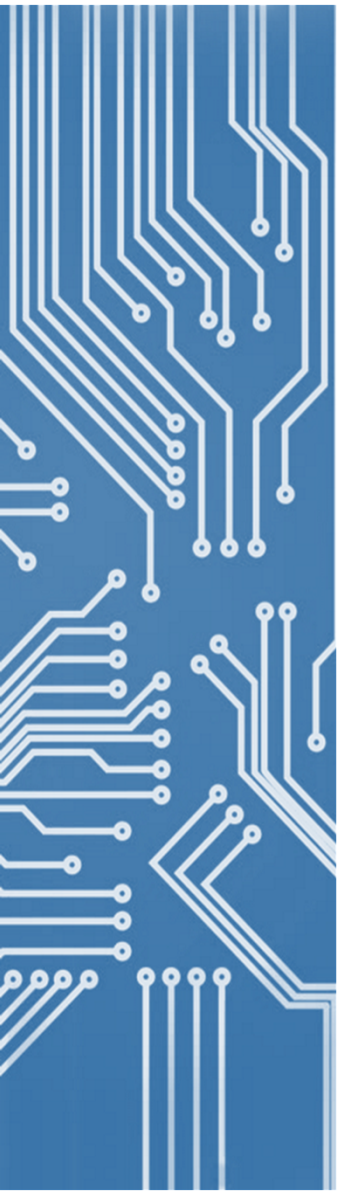
Focus:

Non multimedia.

Dot product.

Absolute, Max/Min, Round

String and Text New Instructions



Advanced Vector eXtensions - AVX

- Width of registers increased from 128 to 256 bits.
- 8 registers in 32 bit mode: YMM0 – YMM7.
- 16 registers in 64 bit mode: YMM0 – YMM15.
- Three-operand SIMD instruction format vs two-operand in SSE.
- The alignment requirement of SIMD memory operands is relaxed.

Advanced Vector eXtensions 2 – AVX2

- Three-operand general-purpose bit manipulation and multiply
- Gather support, enabling vector elements to be loaded from non-contiguous memory locations
- DWORD- and QWORD-granularity any-to-any permutes
- Vector shifts.
- Three-operand fused multiply-accumulate support (**FMA3**)

$a = a \cdot c + b$ $a = b \cdot a + c$ $a = b \cdot c + a$

Note on FMA4…

- Four-operand fused multiply-accumulate support (FMA4)

$$a = b \cdot c + d$$

Only available in:

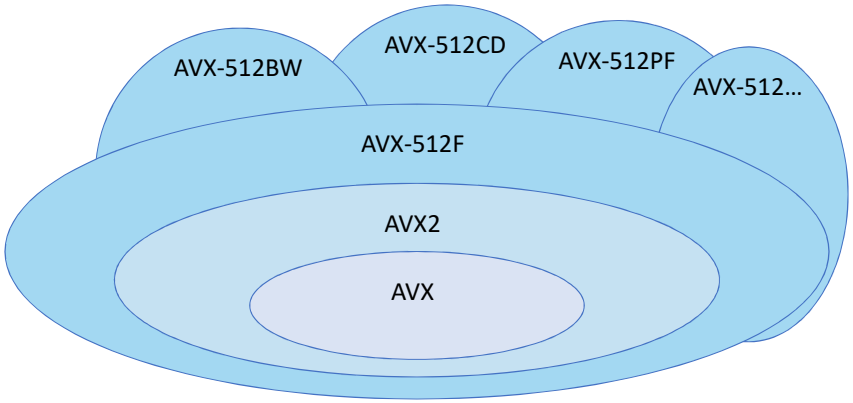
"Heavy equipment" processors

Zen processors*

Advanced Vector eXtensions 512 – AVX-512

- 32 x 512-bit registers (Named: ZMM0 – ZMM31)
- Consists of multiple extensions
- Not meant to be supported by all processors implementing them.
- Only the core extension **AVX-512F (AVX-512 Foundation)** is required by all implementations.

AVX Family



- ✓Each version incorporate new instructions.
- ✓Backward compatibility is guaranteed.
- ✓Each core can execute AVX instructions independently.
- ✓But… there is a consideration…

AVX-512 “Problem”…

- ☐ AVX-512
- ☐ AVX-512F
- ☐ AVX-512BW
- ☐ AVX-512CD
- ☐ AVX-512DQ
- ☐ AVX-512ER
- ☐ AVX-512IFMA52
- ☐ AVX-512PF
- ☐ AVX-512VL
- ☐ AVX-512VPOPCNTDQ
- ☐ AVX-512_4FMAPS
- ☐ AVX-512_4VNNIW
- ☐ AVX-512_BF16
- ☐ AVX-512_BITALG
- ☐ AVX-512_VBMI
- ☐ AVX-512_VBMI2
- ☐ AVX-512_VNNI
- ☐ AVX-512_VP2INTERSECT

AVX-512? AVX-512-F?

Intel is introducing several variations of the new AVX-512 instruction support, based on the hardware and market segmentation. Ultimately there is one underlying set of AVX-512 instructions supported, and the different variants add different instructions.

AVX-512-F: F for Foundation
AVX-512-BW: Support for 512-bit Word support
AVX-512-CD: Conflict Detect (loop vectorization with possible conflicts)
AVX-512-DQ: More instructions for double/quad math operations
AVX-512-ER: Exponential and Reciprocal
AVX-512-IFMA: Integer Fused Multiply Add with 52-bit precision
AVX-512-PF: Prefetch Instructions
AVX-512-VBMI: Vector Byte Manipulation Instructions
AVX-512-VL: Foundation plus <512-bit vector length support
AVX-512-4VNNIW: Vector Neural Network Instructions Word (variable precision)
AVX-512-4FMAPS: Fused Multiply Accumulation Packed Single precision

Performance Benefit

- AVX, AVX2, AVX-512… Is it really worth it?

- 512-bit wide vectors
- 32 operand registers
- 8 64b mask registers
- Embedded broadcast
- Embedded rounding

Microarchitecture	Instruction Set	SP FLOPs / cycle	DP FLOPs / cycle
Skylake	Intel® AVX-512 & FMA	64	32
Haswell / Broadwell	Intel AVX2 & FMA	32	16
Sandybridge	Intel AVX (256b)	16	8
Nehalem	SSE (128b)	8	4

Intel AVX-512 Instruction Types	
AVX-512-F	AVX-512 Foundation Instructions
AVX-512-VL	Vector Length Orthogonality : ability to operate on sub-512 vector sizes
AVX-512-BW	512-bit Byte/Word support
AVX-512-DQ	Additional D/Q/SP/DP instructions (converts, transcendental support, etc.)
AVX-512-CD	Conflict Detect : used in vectorizing loops with potential address conflicts

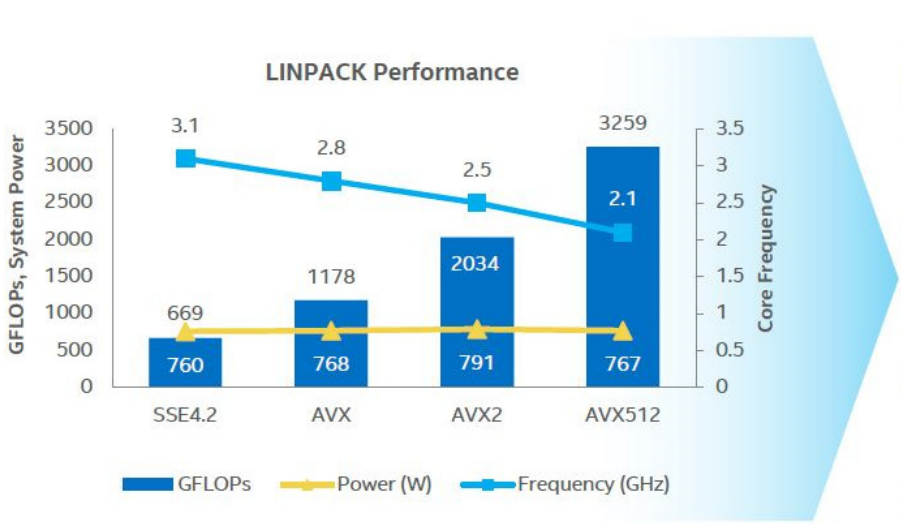
Performance Benefit

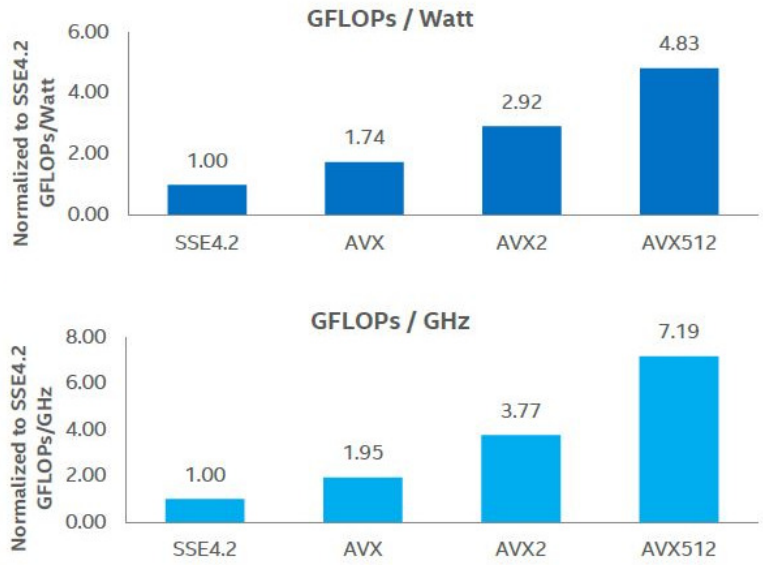
- AVX, AVX2, AVX-512… Is it really worth it?

https://www.extremetech.com/computing/257730-intel-may-deploy-avx-512-upcoming-10nm-cannon-lake-cpus

Performance Benefit

- AVX, AVX2, AVX-512… Is it really worth it?



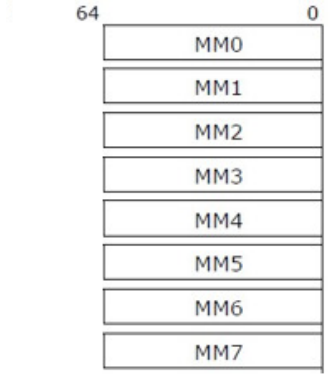


https://www.extremetech.com/computing/257730-intel-may-deploy-avx-512-upcoming-10nm-cannon-lake-cpus

MMX, SSE, AVX & AVX-512

Summary

- All SIMD Registers…

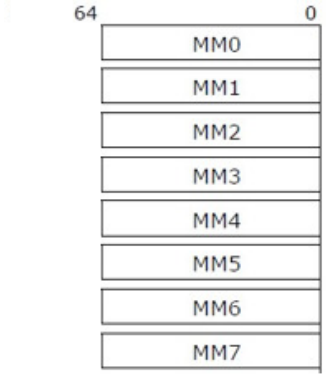


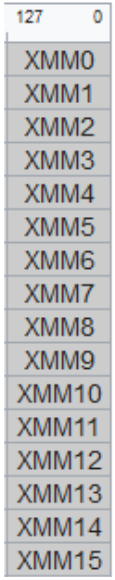
MMX

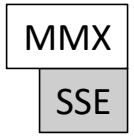
MMX, SSE, AVX & AVX-512

Summary

- All SIMD Registers…



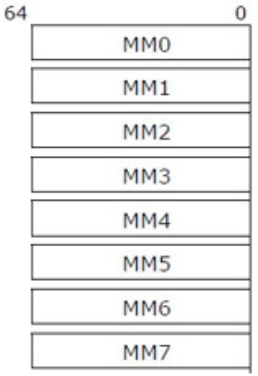


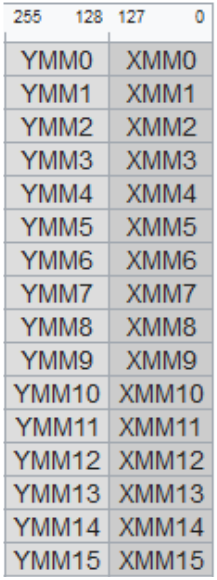


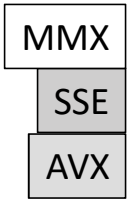
MMX, SSE, AVX & AVX-512

Summary

- All SIMD Registers…



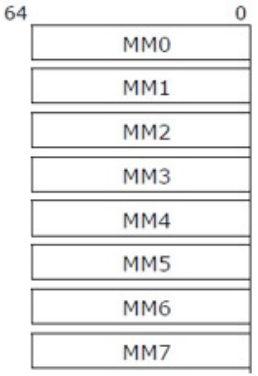




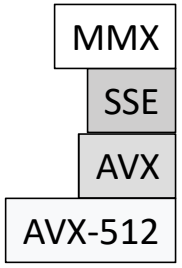
MMX, SSE, AVX & AVX-512

Summary

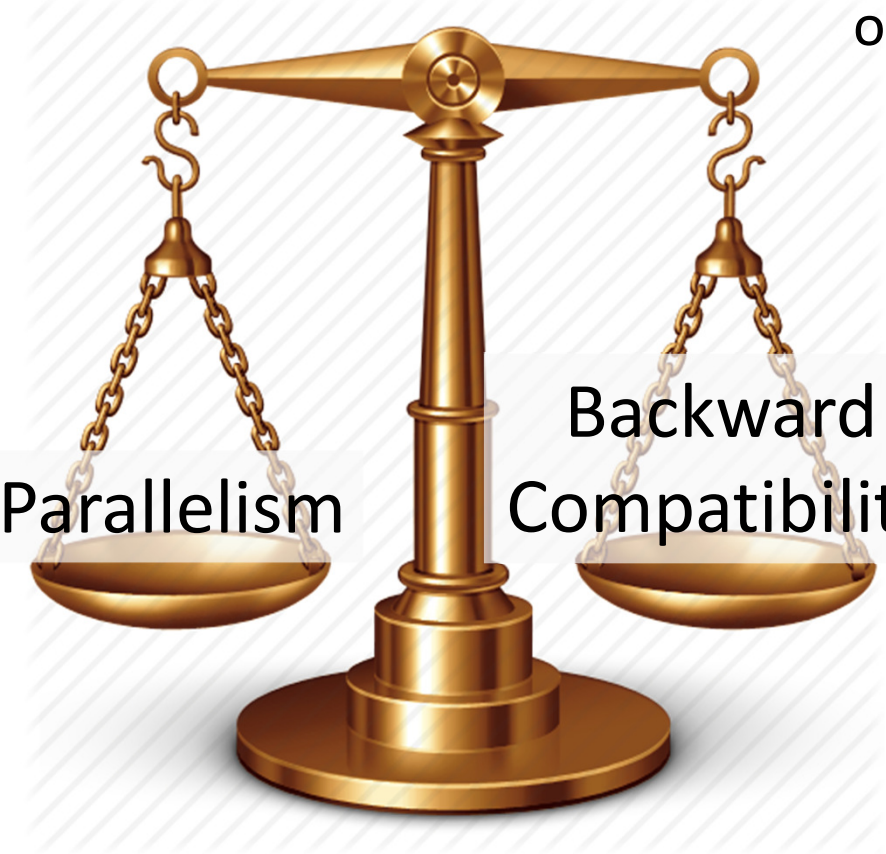
- All SIMD Registers…



511 256	255 128	127 0
ZMM0	YMM0	XMM0
ZMM1	YMM1	XMM1
ZMM2	YMM2	XMM2
ZMM3	YMM3	XMM3
ZMM4	YMM4	XMM4
ZMM5	YMM5	XMM5
ZMM6	YMM6	XMM6
ZMM7	YMM7	XMM7
ZMM8	YMM8	XMM8
ZMM9	YMM9	XMM9
ZMM10	YMM10	XMM10
ZMM11	YMM11	XMM11
ZMM12	YMM12	XMM12
ZMM13	YMM13	XMM13
ZMM14	YMM14	XMM14
ZMM15	YMM15	XMM15
ZMM16	YMM16	XMM16
ZMM17	YMM17	XMM17
ZMM18	YMM18	XMM18
ZMM19	YMM19	XMM19
ZMM20	YMM20	XMM20
ZMM21	YMM21	XMM21
ZMM22	YMM22	XMM22
ZMM23	YMM23	XMM23
ZMM24	YMM24	XMM24
ZMM25	YMM25	XMM25
ZMM26	YMM26	XMM26
ZMM27	YMM27	XMM27
ZMM28	YMM28	XMM28
ZMM29	YMM29	XMM29
ZMM30	YMM30	XMM30
ZMM31	YMM31	XMM31



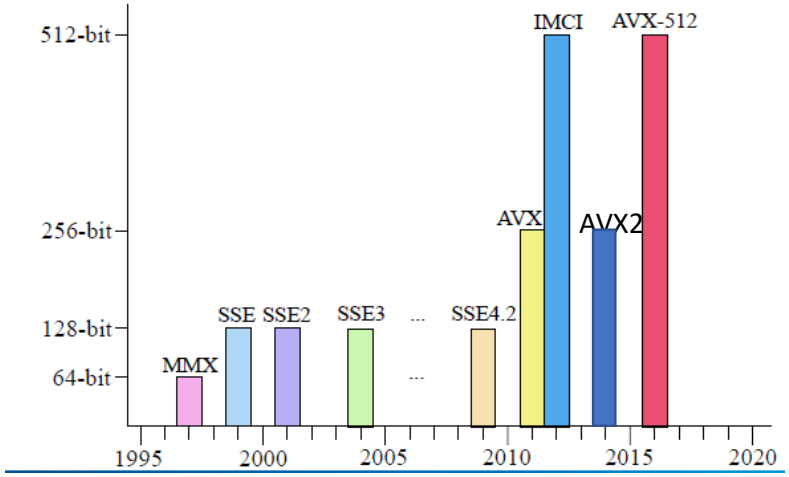
You choose…



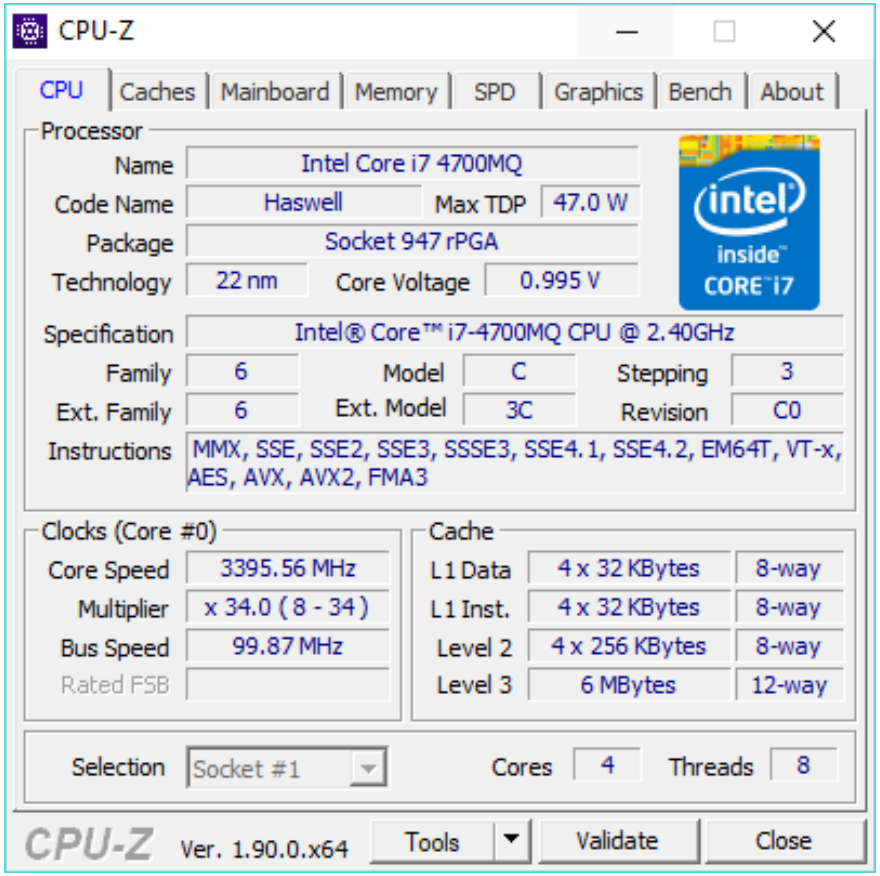
Is it posible to support more than one Extension Instruction Set?

It’s posible:

- More code
- Bigger binaries
- Installation logic



CPU Internals…



CPU-Z

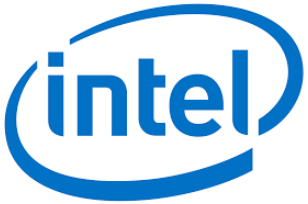
CPUID

https://www.cpuid.com/

Kahoot!

Nickname: Axxxxxx - Nombre

Suggested Material



Introduction to x64 Assembly

AMD64 Technology

For next class…

What Happened to Cyrix Processors? | Nostalgia Nerd

https://www.youtube.com/watch?v=iWGAdoMz1c0

Thank you!

See you next time!