



Ministerstwo Nauki
i Szkolnictwa Wyższego

**Regionalna Inicjatywa Doskonałości w Dyscyplinach
Informatyki, Elektrotechniki, Elektroniki, Automatyki i Robotyki
na Politechnice Częstochowskiej**



Projekt w ramach programu Regionalna Inicjatywa Doskonałości, decyzja nr 020/RID/2018/19

Architektura Systemów Komputerowych

Koprocesor numeryczny oraz jednostki ALU, FPU i VPU: budowa, właściwości i podstawowe operacje

dr hab. inż. Łukasz Szustak, prof. PCz
lszustak@icis.pcz.pl

Katedra Informatyki
Wydział Inżynierii Mechanicznej i Informatyki
Politechnika Częstochowska



Wydział Inżynierii
Mechanicznej
i Informatyki
The Faculty of Mechanical Engineering
and Computer Science



Ministerstwo Nauki
i Szkolnictwa Wyższego

Regionalna Inicjatywa Doskonałości w Dyscyplinach Informatyki, Elektrotechniki, Elektroniki, Automatyki i Robotyki na Politechnice Częstochowskiej



Projekt w ramach programu Regionalna Inicjatywa Doskonałości, decyzja nr 020/RID/2018/19

Zamieszczane materiały służą wyłącznie do celów indywidualnego kształcenia. Nie wyrażam zgody na ich utrwalanie przekazywanie osobom trzecim ani rozpowszechnianie.



dr hab. inż. Łukasz Szustak, prof. PCz
lszustak@icis.pcz.pl

Katedra Informatyki
Wydział Inżynierii Mechanicznej i Informatyki
Politechnika Częstochowska



Wydział Inżynierii
Mechanicznej
i Informatyki
The Faculty of Mechanical Engineering
and Computer Science

Koncepcja modelu data parallelism

SIMD

(Single Instruction, Multiple Data)

Architektura SIMD

- SIMD (Single Instruction, Multiple Data) - przetwarzanych jest wiele danych przez jeden wykonywany program - tzw. komputery wektorowe

A vector processor or array processor is a central processing unit (CPU) that implements an instruction set containing instructions that operate on one-dimensional arrays of data called vectors, compared to the scalar processors, whose instructions operate on single data items.

Architektura SIMD

- SIMD (Single Instruction, Multiple Data) - przetwarzanych jest wiele danych przez jeden wykonywany program - tzw. komputery wektorowe
- Cray 1 (1976) :
 - 8 MB RAM
 - 160 MFLOPS
 - chłodzenie sprężarkowe
 - charakterystyczny kształt litery „C”



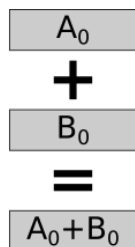
Architektura SIMD

- Współczesne architektury procesorów różnych producentów wspierają wektoryzację, np...:
 - **x86:**
 - SSE 128 bit, AVX(2) 256 bit, AVX-512 512 bit
 - **ARM:**
 - NEON 128 bit
 - **POWER:**
 - AltiVec/VMX/VSX 128 bit, QPX 256 bit
 - **SPARC:**
 - HPC-ACE 128 bit, HPC-ACE2 256 bit

Architektura SIMD

Obliczenia skalarne

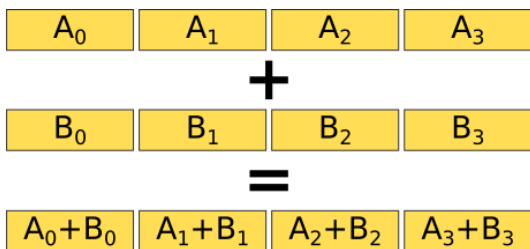
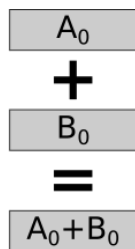
Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



Architektura SIMD

Obliczenia skalarne

Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



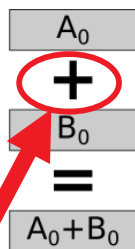
Obliczenia wektorowe

Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

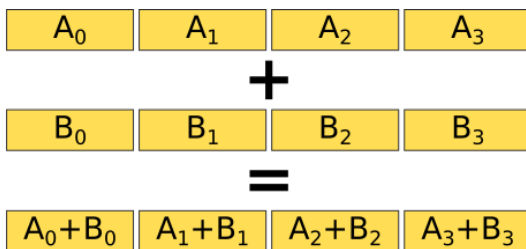
Architektura SIMD

Obliczenia skalarne

Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



**Skalarna
instrukcja
dodawania**



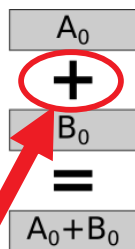
Obliczenia wektorowe

Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

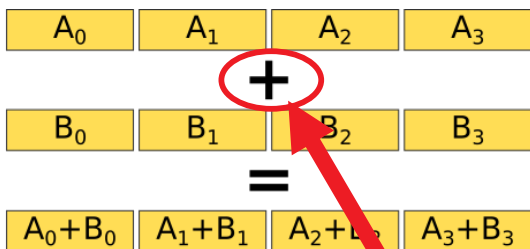
Architektura SIMD

Obliczenia skalarne

Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



**Skalarna
instrukcja
dodawania**



**Wektorowa
instrukcja
dodawania**

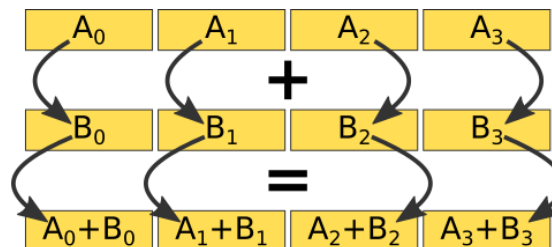
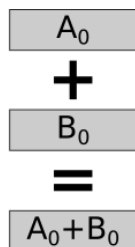
Obliczenia wektorowe

Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

Architektura SIMD

Obliczenia skalarne

Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



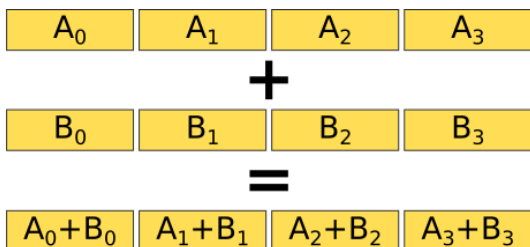
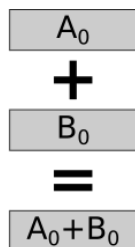
Obliczenia wektorowe

Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

Architektura SIMD

Obliczenia skalarne

Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



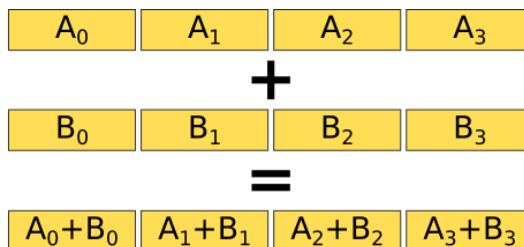
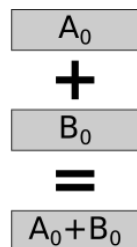
Obliczenia wektorowe

Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

W powyższym przypadku przetwarzanych jest
4x więcej elementów

Architektura SIMD

Obliczenia skalarne
Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



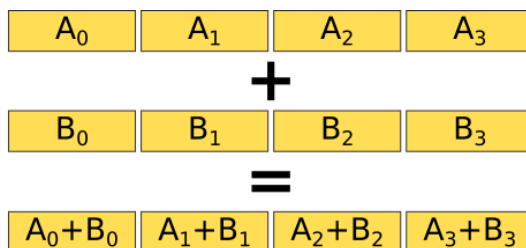
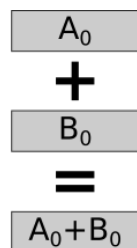
Obliczenia wektorowe
Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

W powyższym przypadku przetwarzanych jest
4x więcej elementów

Zatem, potencjalny zysk wynika z liczby elementów
przetwarzanych równocześnie

Architektura SIMD

Obliczenia skalarne
Pojedynczy element
przetwarzany z pomocą
pojedynczej operacji



Obliczenia wektorowe
Zbiór elementów (wektor)
przetwarzany z pomocą
pojedynczej operacji

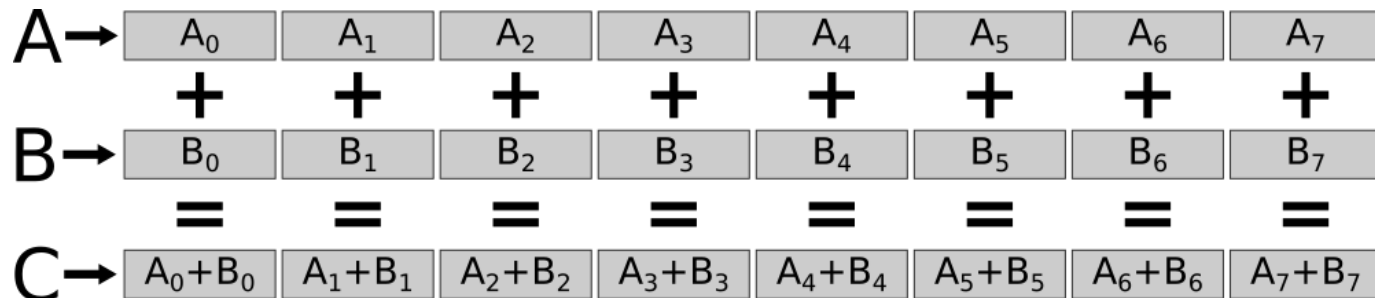
W powyższym przypadku przetwarzanych jest
4x więcej elementów

Zatem, potencjalny zysk wynika z liczby elementów
przetwarzanych równocześnie

Liczba elementów zależy od typu danych oraz rozmiaru rejestrów

Architektura SIMD

Obliczenia Skalarne



Architektura SIMD

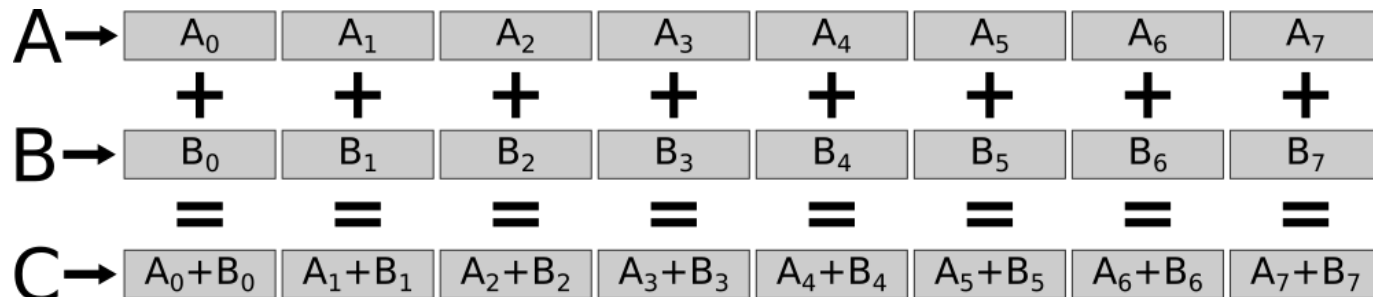
Obliczenia Skalarne

A →	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇
	+	+	+	+	+	+	+	+
B →	B ₀	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇
	=	=	=	=	=	=	=	=
C →	A ₀ +B ₀	A ₁ +B ₁	A ₂ +B ₂	A ₃ +B ₃	A ₄ +B ₄	A ₅ +B ₅	A ₆ +B ₆	A ₇ +B ₇

8 operacji
dodawania
z użyciem
8 pojedynczych
elementów tablicy

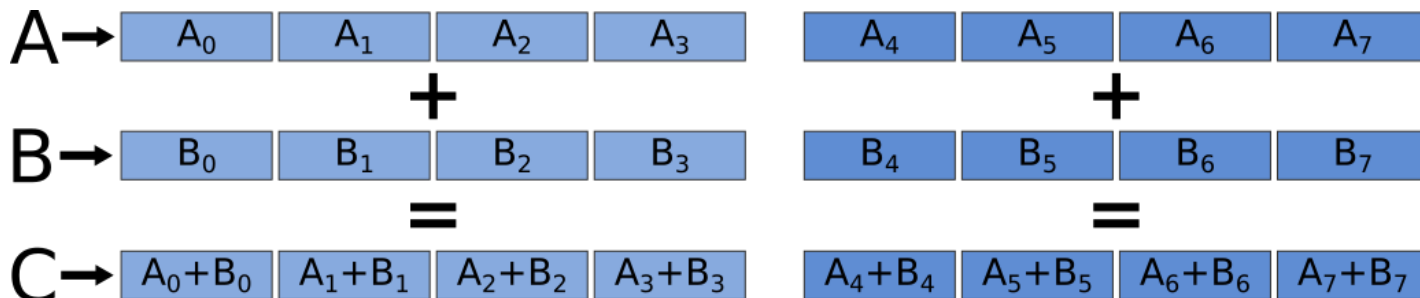
Architektura SIMD

Obliczenia Skalarne



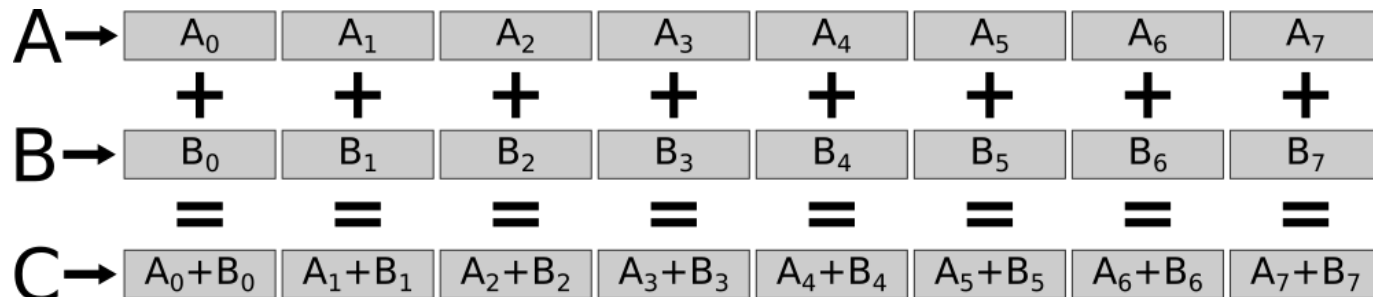
8 operacj
dodawania
z użyciem
8 pojedynczych
elementów tablicy

Obliczenia Wektorowe



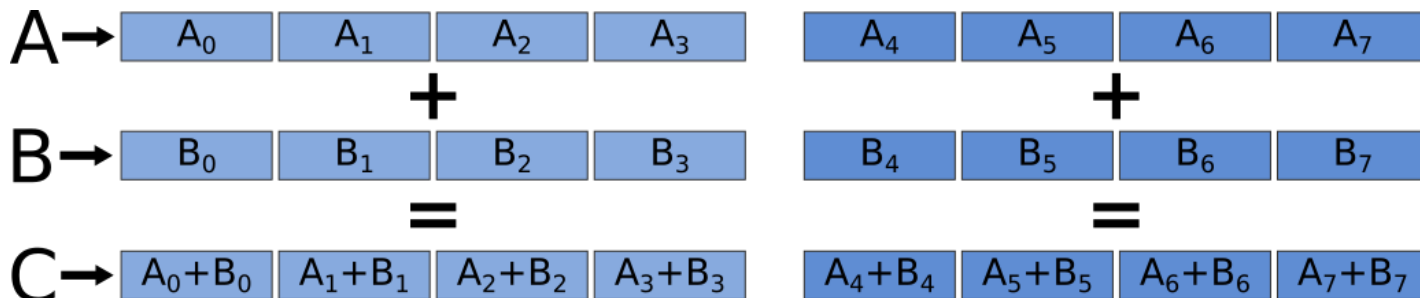
Architektura SIMD

Obliczenia Skalarne



8 operacji
dodawania
z użyciem
8 pojedynczych
elementów tablicy

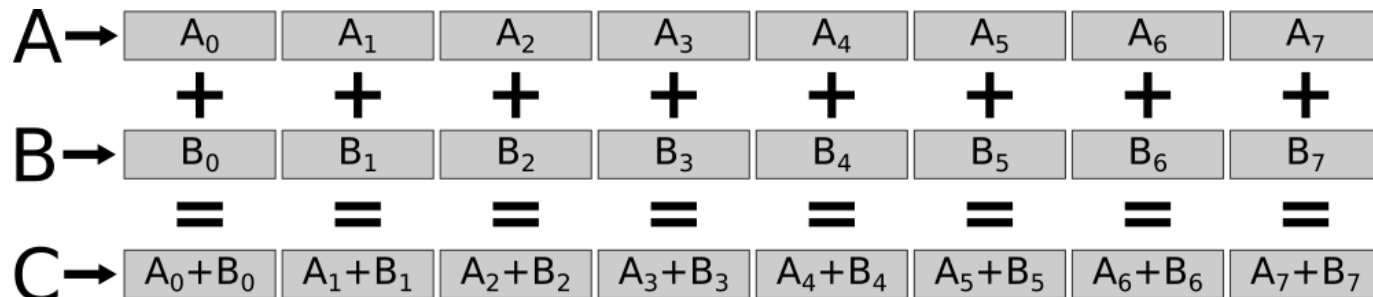
Obliczenia Wektorowe



2 operacje
dodawania
z użyciem 2
czteroelementowych
wektorów

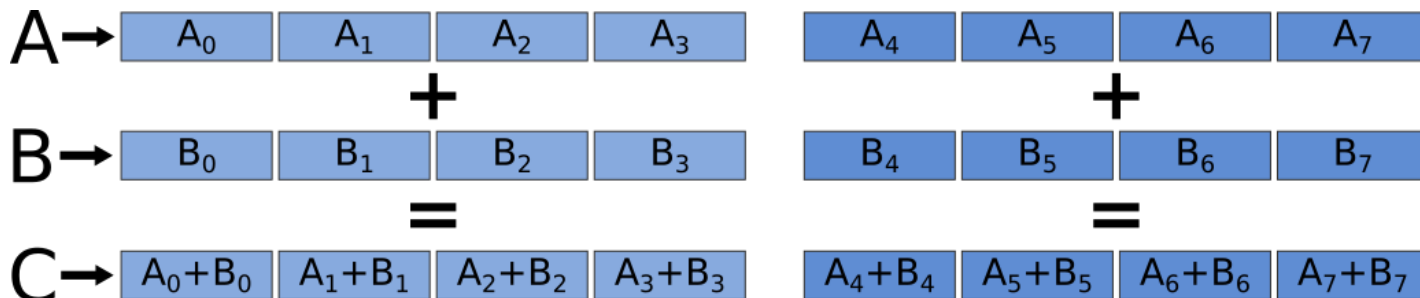
Architektura SIMD

Obliczenia Skalarne



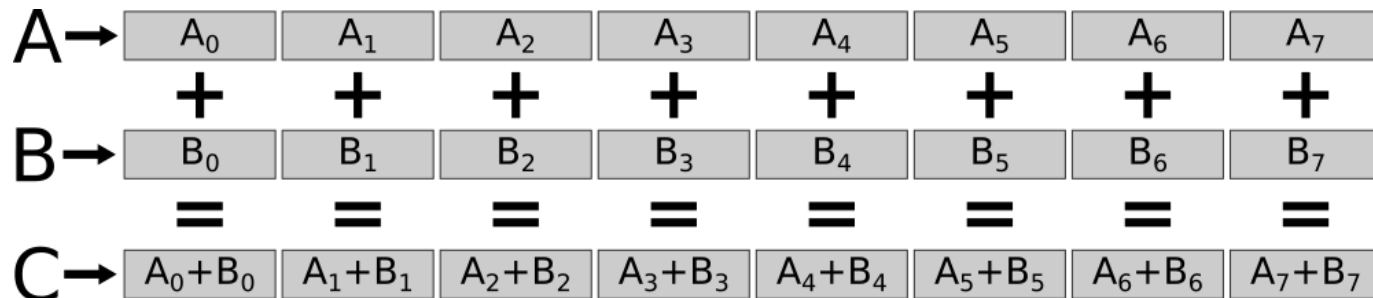
Zakładając, że
operacja dodawania
zajmuje 1 cykl w obu
przypadkach

Obliczenia Wektorowe



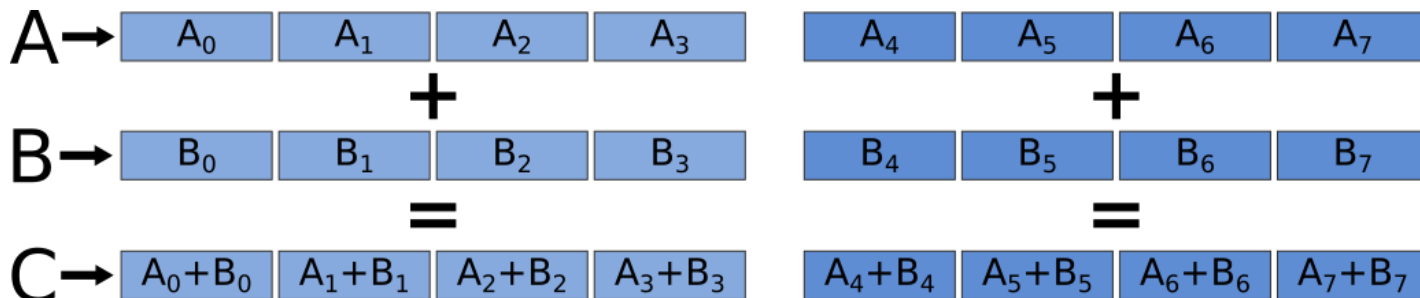
Architektura SIMD

Obliczenia Skalarne



Zakładając, że operacja dodawania zajmuje 1 cykl w obu przypadkach

Obliczenia Wektorowe

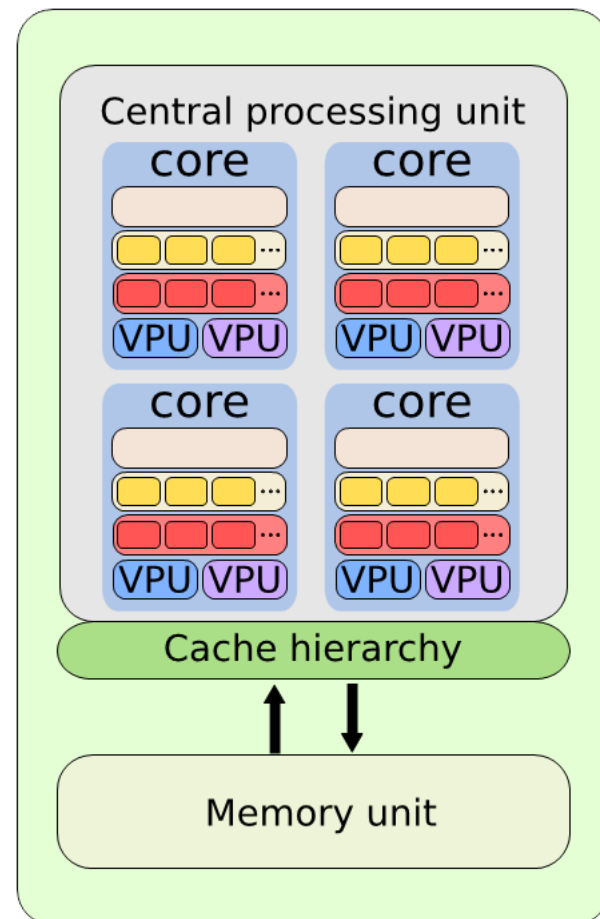


Użycie architektury SIMD umożliwi redukcję liczby cykli z 8 do 2:

4x mniej cykli

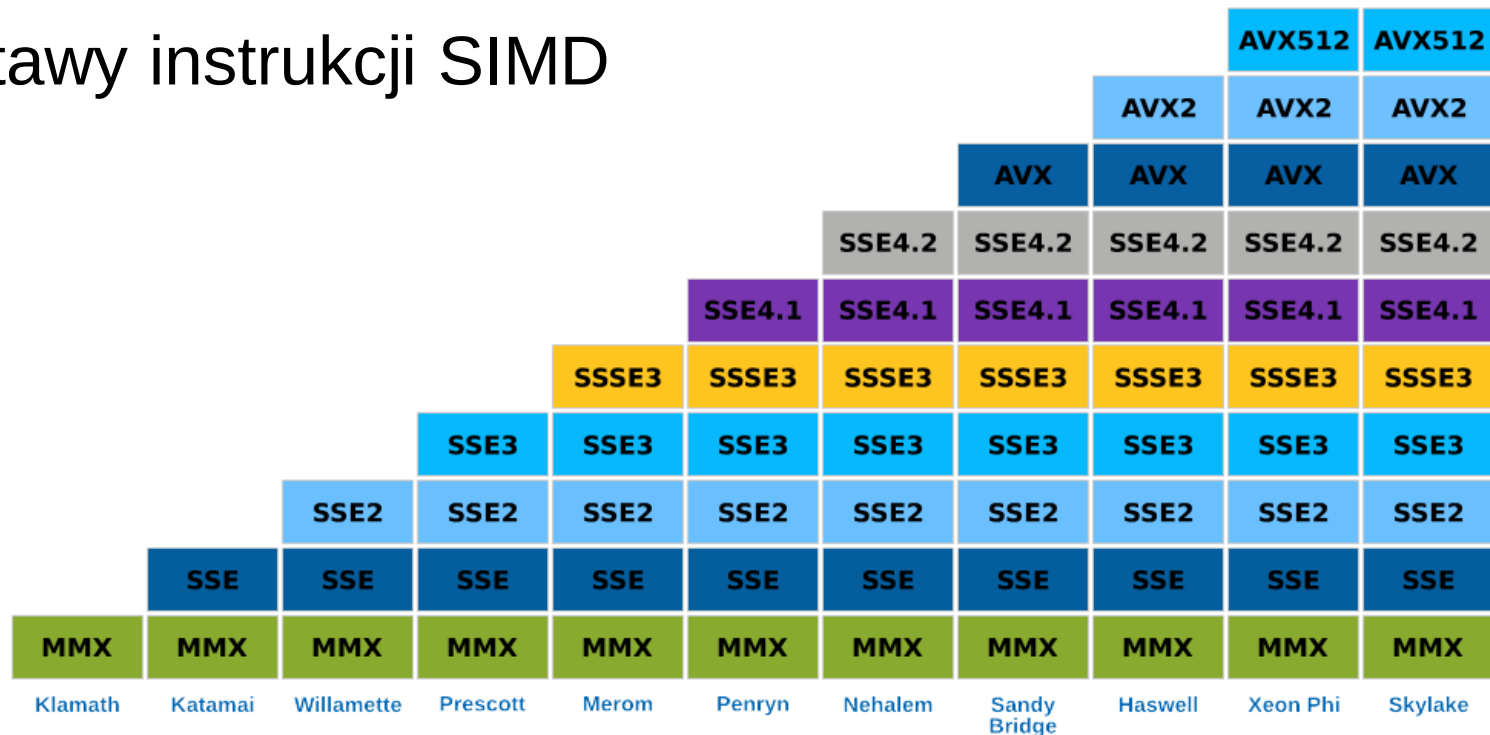
Architektura SIMD

- Obliczenia wektorowe realizowane są z pomocą jednostek wektorowych **VPU**
- W każdym rdzeniu procesora wspierającego SIMD znajdują się co najmniej jedna jednostka VPU
- Każdy rdzeń zawiera zbiór rejestrów dedykowanych do wykonywania instrukcji wektorowe
- Rozmiar rejestrów wektorowych zależy od danej architektury procesora
- Każda jednostka VPU obsługuje zestaw instrukcji (np.. +,-,*,/) dedykowany dla danej architektury procesora



Rozwój architektury SIMD na przykładzie procesorów firmy Intel

- Zestawy instrukcji SIMD



1997

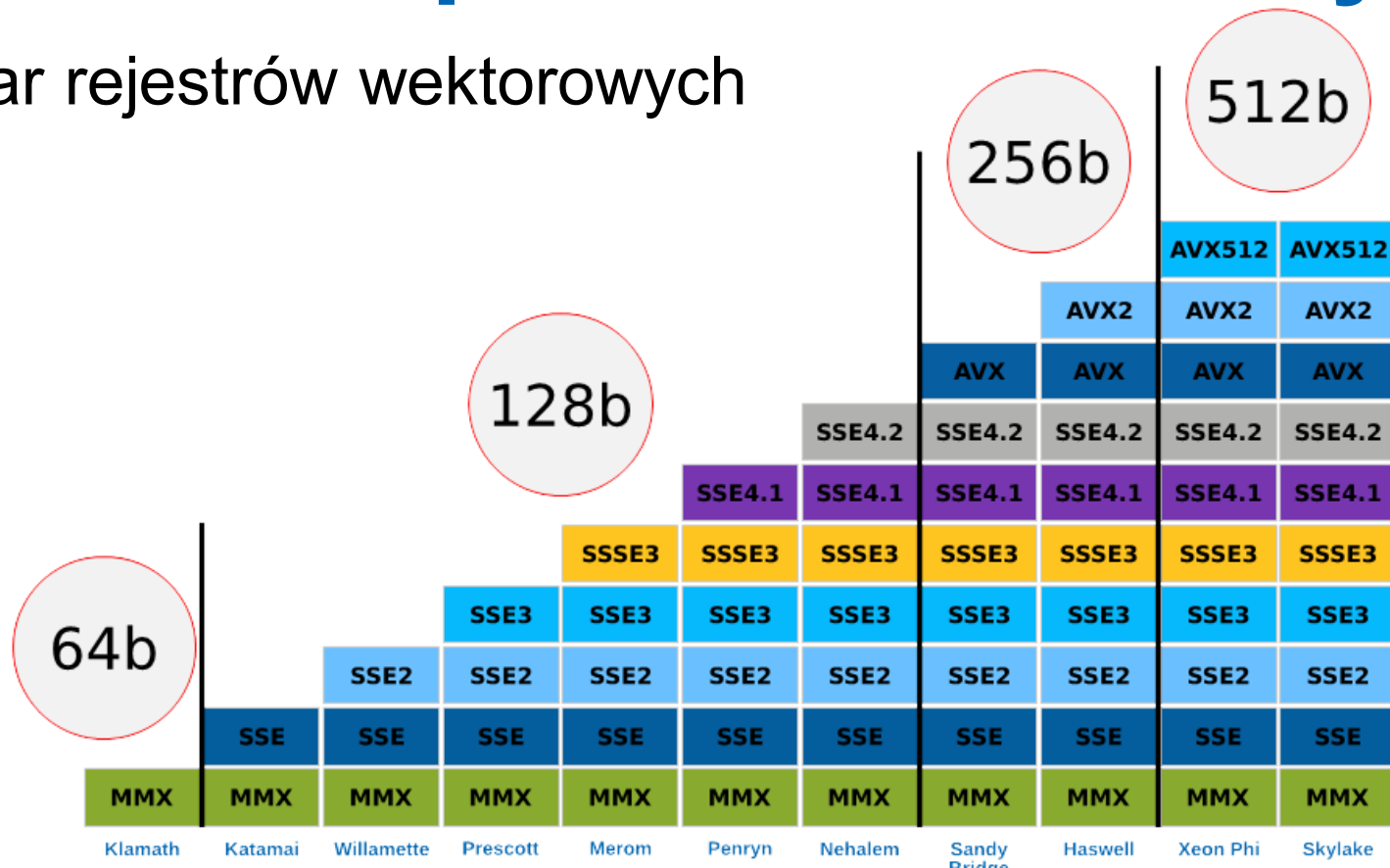
Rozwój Intel SIMD

2016,...

<https://software.intel.com/sites/landingpage/IntrinsicsGuide/>

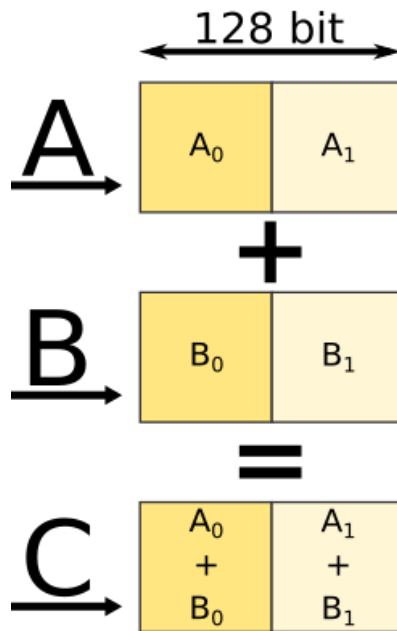
Rozwój architektury SIMD na przykładzie procesorów firmy Intel

- Rozmiar rejestrów wektorowych



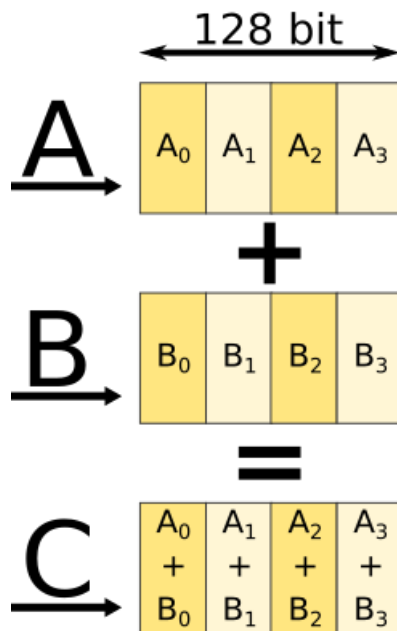
Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **SSE**
- **128** bitowe rejestry „XMM”
- **2 elementy** typu double (64 bity)



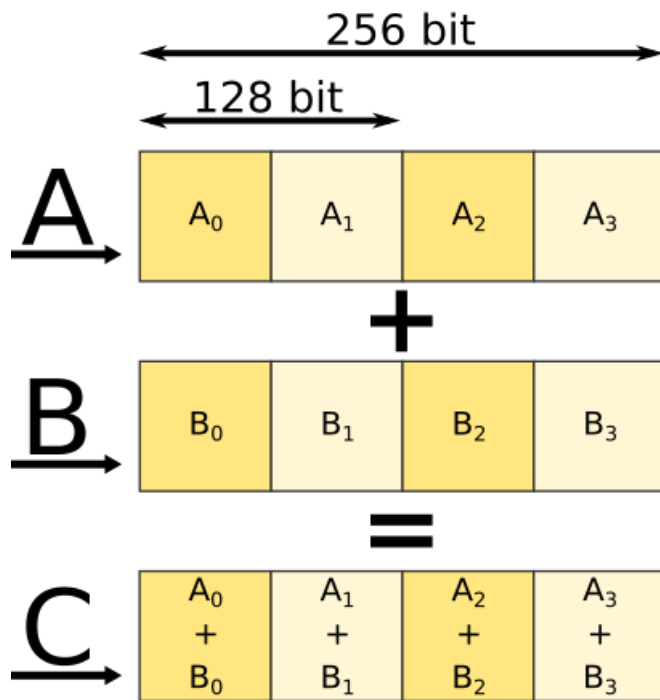
Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **SSE**
- **128** bitowe rejestry „XMM”
- **4 elementy** typu float (32 bity)



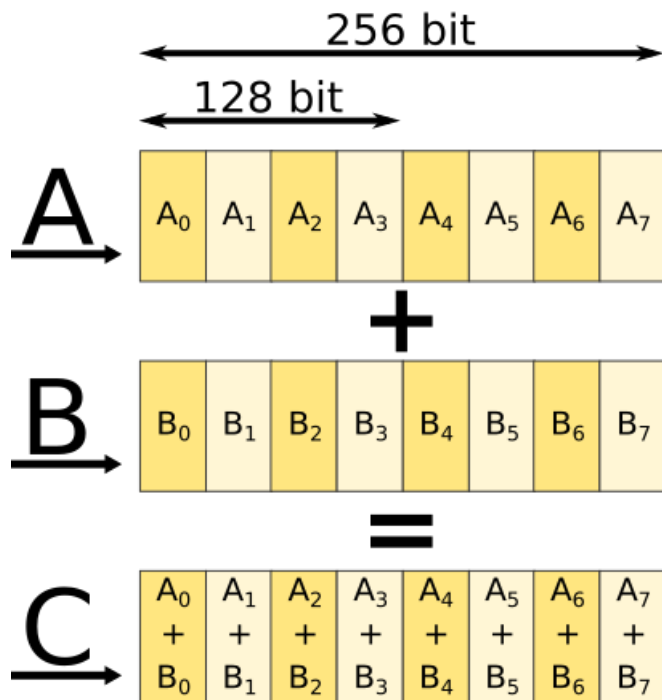
Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **AVX**
- **256** bitowe rejestry „YMM”
- **4 elementy** typu double (64 bity)



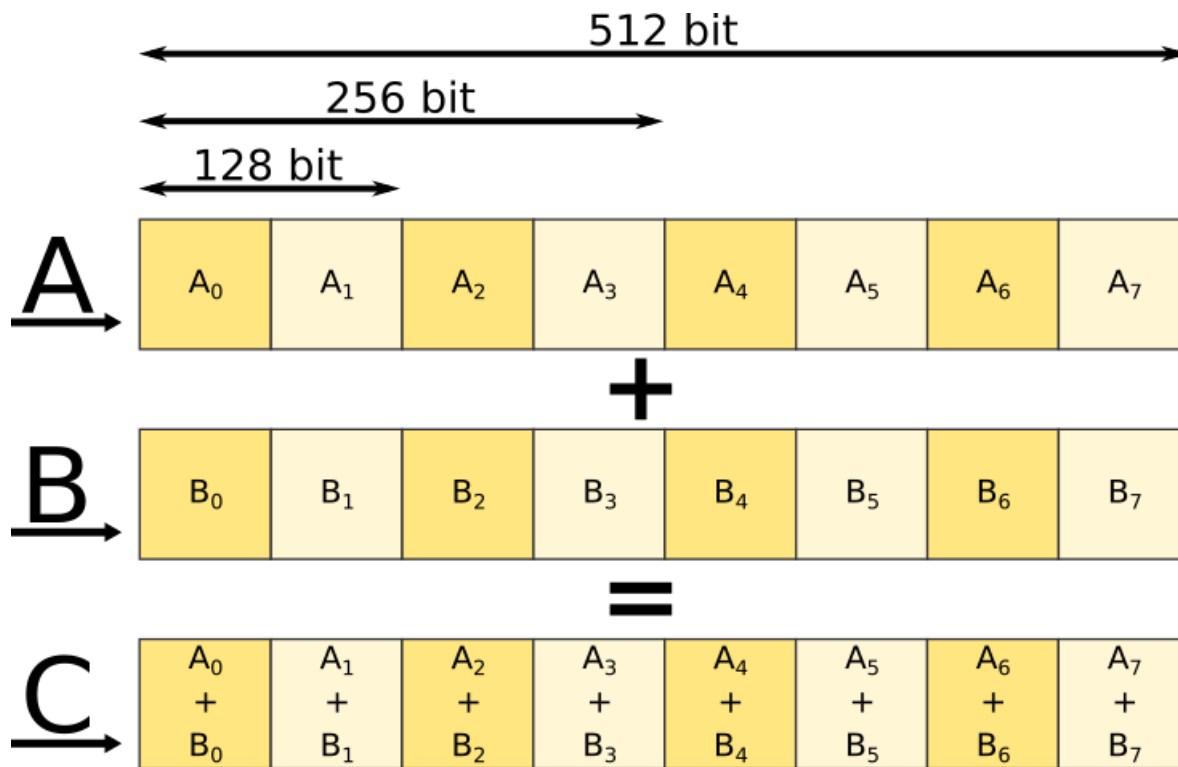
Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **AVX**
- **256** bitowe rejestry „YMM”
- **8 elementów** typu float (32 bity)



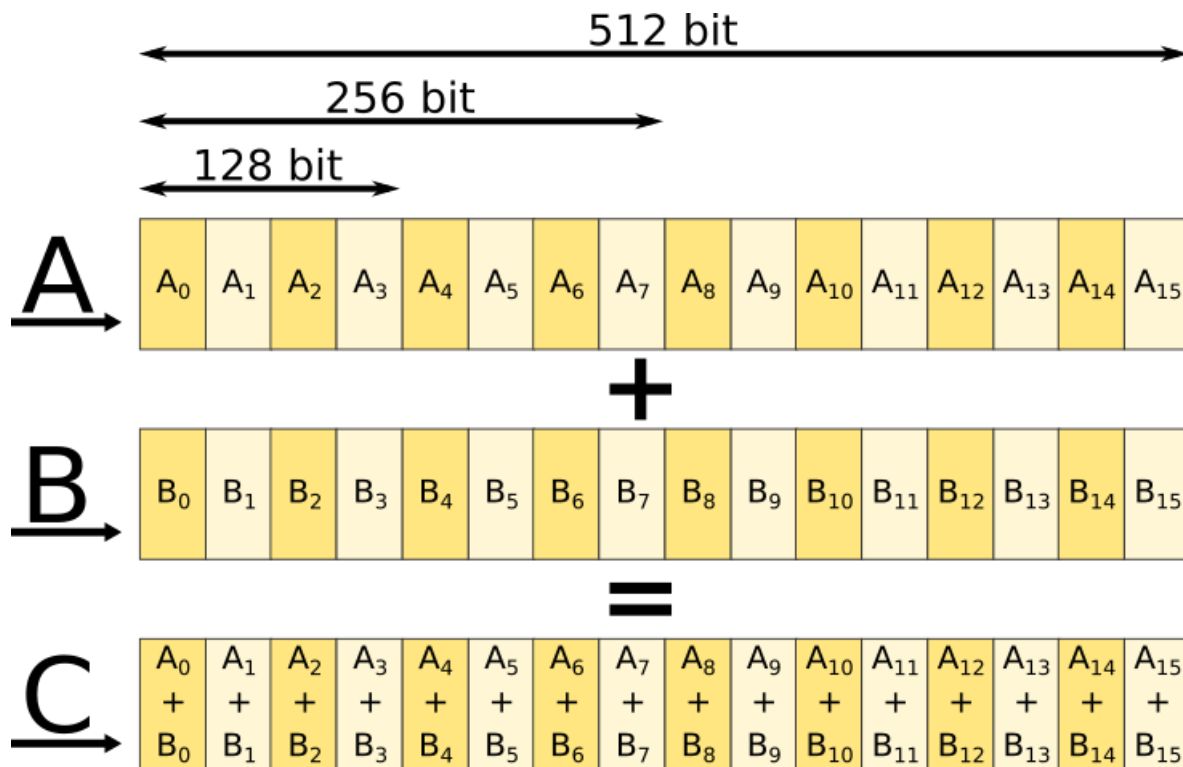
Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **AVX-512**
- **512** bitowe rejestry „ZMM”
- **8 elementów** typu double (64 bity)



Architektura SIMD na przykładzie procesorów firmy Intel

- Zestaw instrukcji **AVX-512**
- **512** bitowe rejestry „ZMM”
- **16 elementów** typu float (32 bity)



Rozwój architektury SIMD na przykładzie procesorów firmy Intel

```
rid@gpu2:~/RID/T5$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          46 bits physical, 48 bits virtual
CPU(s):                 72
On-line CPU(s) list:    0-71
Thread(s) per core:     2
Core(s) per socket:     18
Socket(s):               2
NUMA node(s):           2
Vendor ID:               GenuineIntel
CPU family:              6
Model:                   63
Model name:              Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
Stepping:                 2
CPU MHz:                 1198.211
CPU max MHz:             3600.0000
CPU min MHz:             1200.0000
BogoMIPS:                 4589.45
Virtualization:          VT-x
L1d cache:               32K
L1i cache:               32K
L2 cache:                 256K
L3 cache:                 46080K
NUMA node0 CPU(s):       0-17,36-53
NUMA node1 CPU(s):       18-35,54-71
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx xsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtsc
cp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid
aperfperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cpl
6 xtrn pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
avx_16c rdrand lahf_lm abm cpuid_fault epb invpcid_single pti intel_ppin ssbd ibrs
rpbpb stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bnd
il_1 avx2 smep bmi2 erms invpcid cqm xsaveopt cqm_llc cqm_occup_llc dtherm ida arat
pln pbs md_clear flush_lld
```

- Różne architektury CPU wspierają różne zestawy instrukcji, np..
- Intel Xeon E5-2699v3

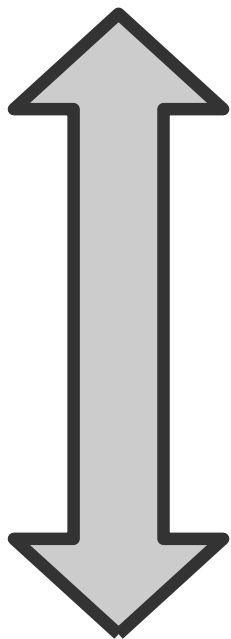
Rozwój architektury SIMD na przykładzie procesorów firmy Intel

```
[lszustak@teslav100 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                72
On-line CPU(s) list:   0-71
Thread(s) per core:    2
Core(s) per socket:    18
Socket(s):              2
NUMA node(s):          2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  85
Model name:             Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz
Stepping:               7
CPU MHz:                999.914
CPU max MHz:           3900.0000
CPU min MHz:           1000.0000
BogoMIPS:               5200.00
Virtualization:         VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              1024K
L3 cache:              25344K
NUMA node0 CPU(s):     0-17,36-53
NUMA node1 CPU(s):     18-35,54-71
Flags:                  fpu vme de pse tsc mce mca mtrr pge m
ca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopol
ogy nonstop tsc_aperf mperf eagerfpu pni pclmulqdq dtcs64 monitor ds_cpl vmx s
mx est tm2 sse3 bdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch
epb cat_l3 cdp_l3 invpcid_single intel_pt ssbd mba ibrs ibpb stibp ibrs_aba
nced tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bml hle avx2
smep bmi2 erms invpcid rtm rcp mpx rdt a avx512f avx512dq dseed adx smap clf
flushopt clwb avx512cd avx512bw avx512vb_1 saveopt xsavec xgetbv1 cqm_llc cqm_o
ccup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts hwp hwp_act_wind
ow hwp_epp hwp_pkg_req pku ospke avx512_vnni and_clear spec_ctrl intel_stibp f
lush_lld arch_capabilities
```

- Różne architektury CPU wspierają różne zestawy instrukcji, np..
- Intel Xeon Gold 6240

Wektoryzacja obliczeń

Łatwe w użyciu



Duża kontrola

Biblioteki

(MKL, OpenBLAS, BLIS, tw, numpy, OpenCV)

Automatyczna wektoryzacja obliczeń
(kompilatory, np.. gcc, intel icc)

Półautomatyczna wektoryzacja obliczeń
(kompilatory + odpowiednie przygotowanie kodu)

Manualna wektoryzacja obliczeń
(instrukcje wbudowane, np.. intrinsics, assembler)

Dziękuję za uwagę



Kontakt:

dr hab. inż. Łukasz Szustak, prof. PCz

lszustak@icis.pcz.pl

Katedra Informatyki

Wydział Inżynierii Mechanicznej i Informatyki

Politechnika Częstochowska