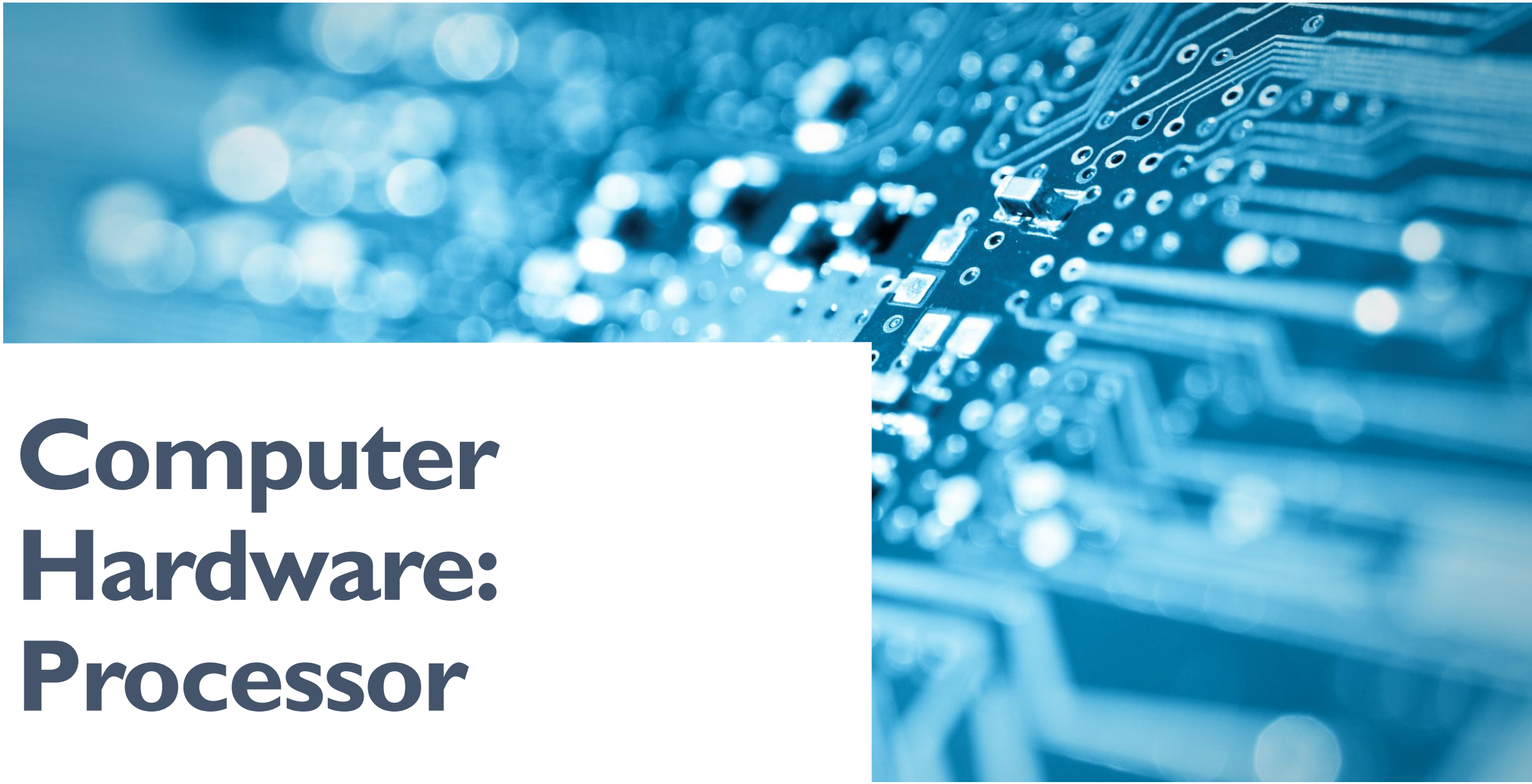
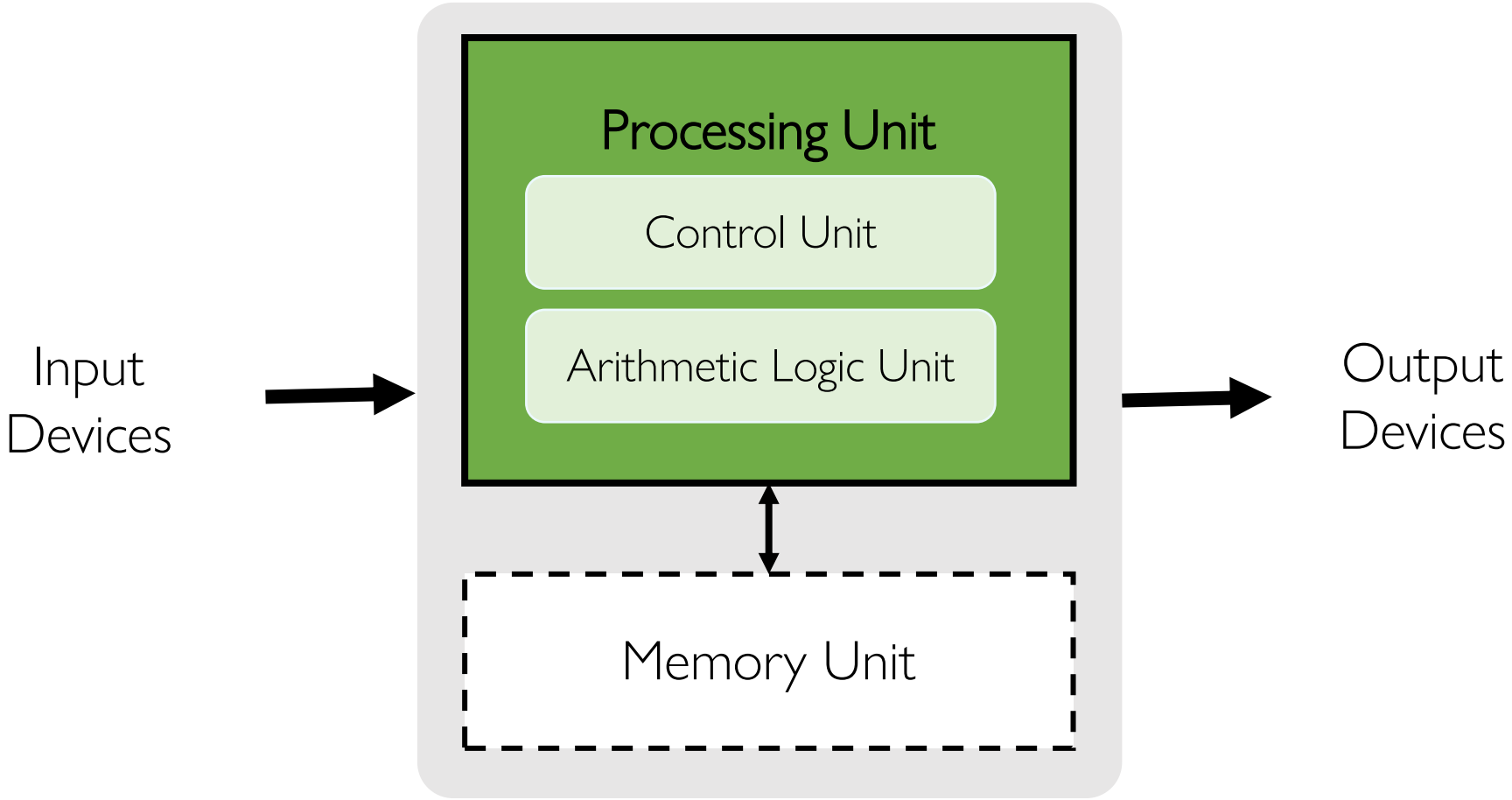


Computer Hardware: Processor

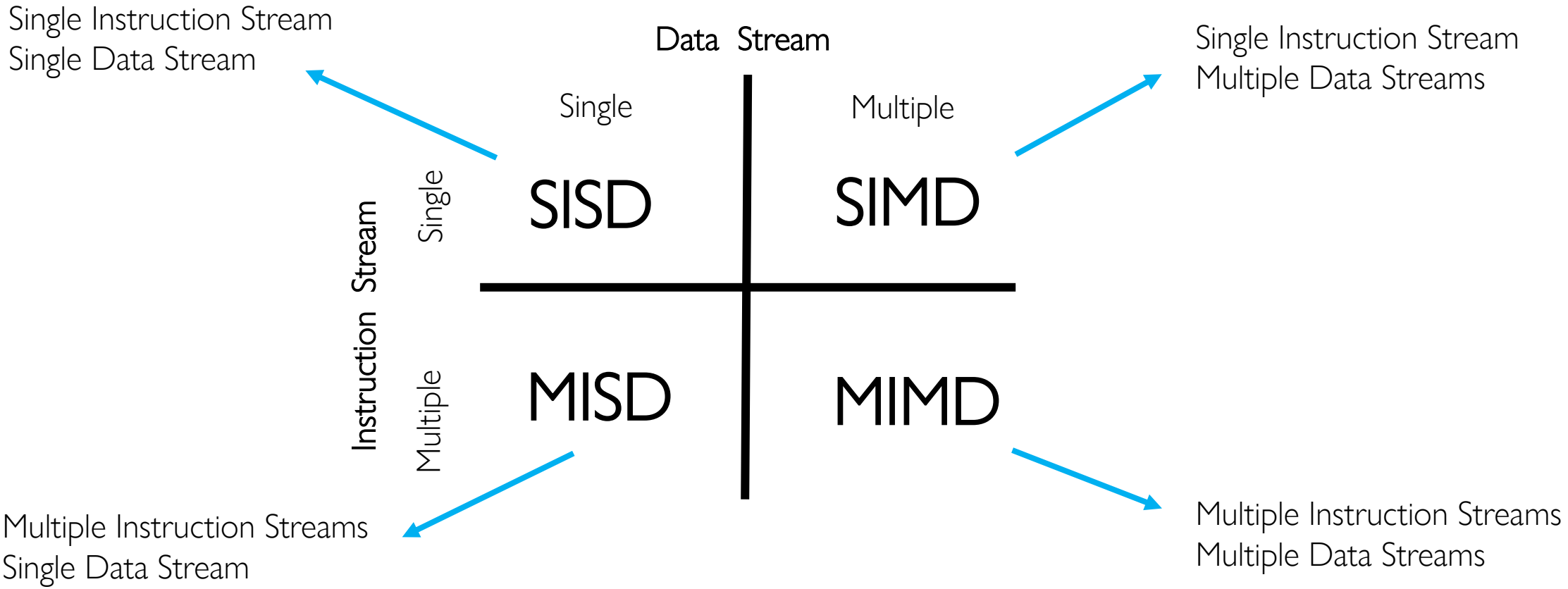
Von Neumann Architecture



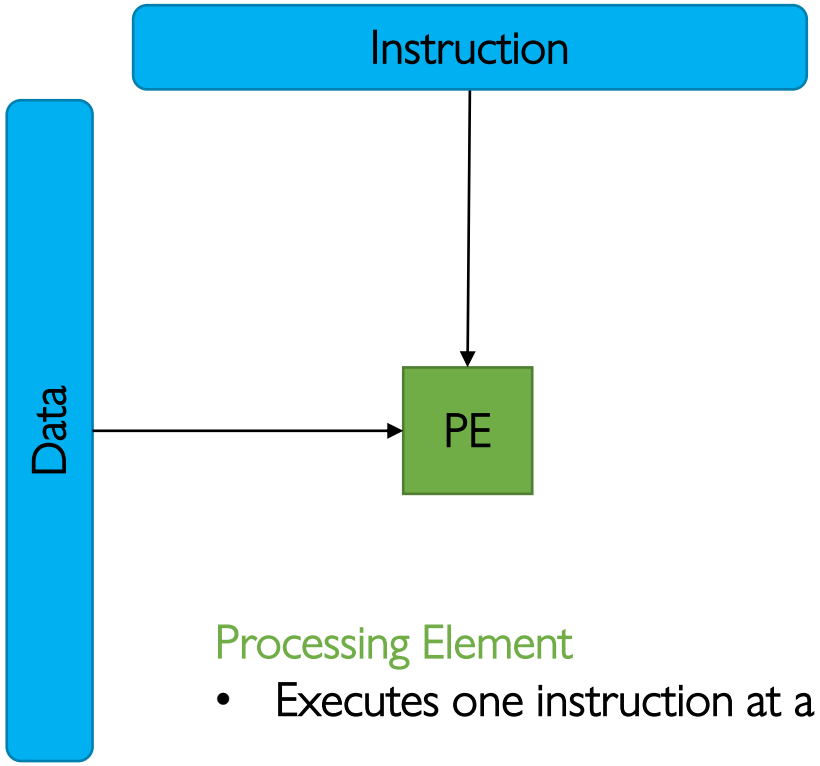
Multitasking

Flynn's Taxonomy

A lot of parallel processors use a hybrid of these taxonomy



Single Instruction Stream Single Data Streams



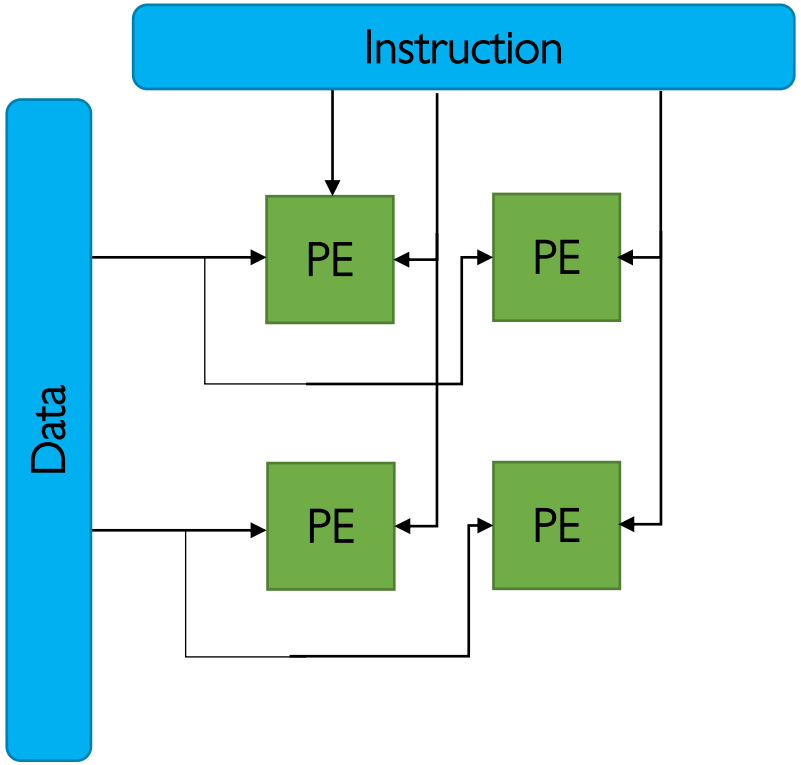
Processing Element

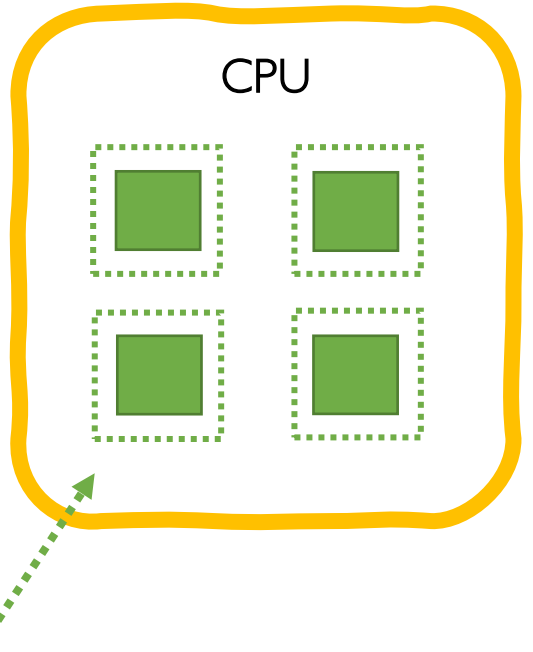
- Executes one instruction at a time

- Also known as a scalar processor
- A standard sequential computer

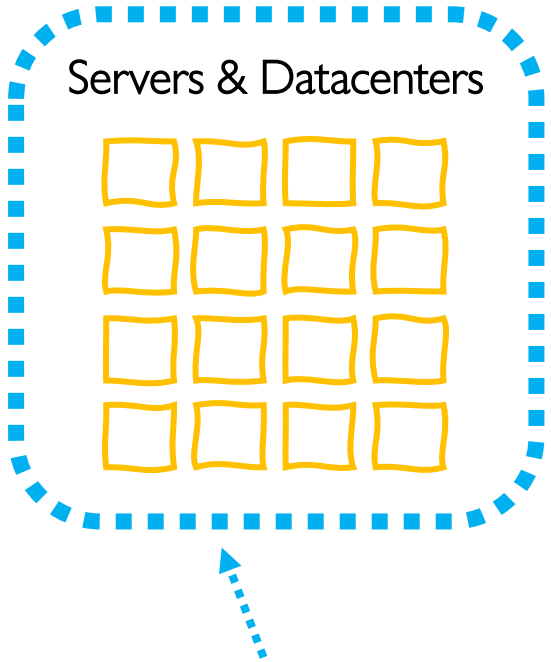
- Found in:
  - Legacy computers
  - Found in some Internet of Things devices
  - Can be used to control another type of processors

Multiple Instruction Stream Multiple Data Streams



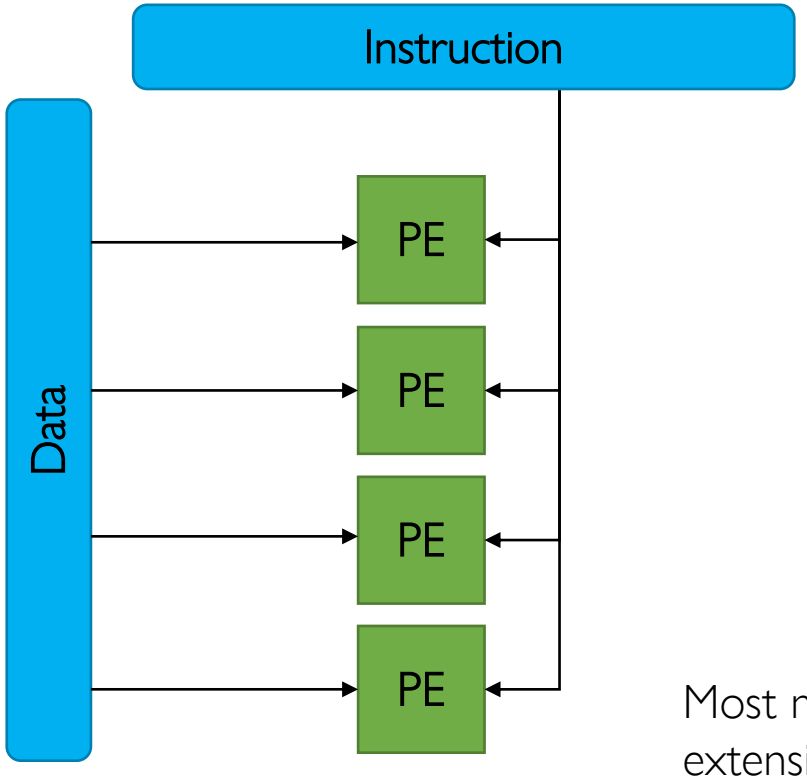


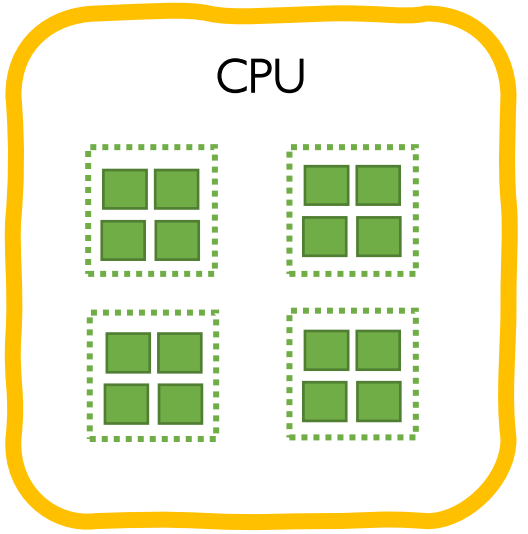
Each CPU core can implement a different instructions on different data

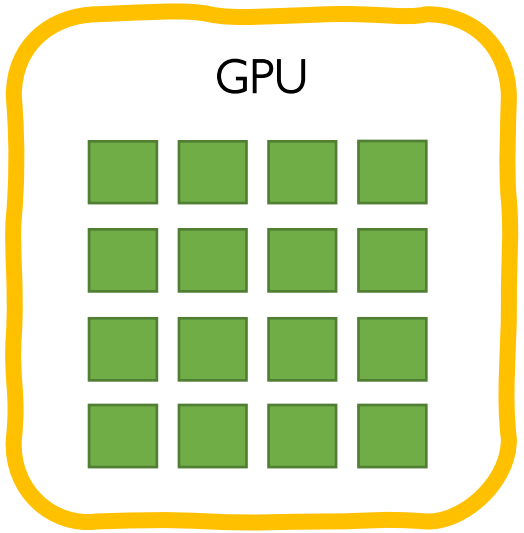


Servers and datacenters typically use multiple cores and multiple computers to run MIMD

Single Instruction Stream Multiple Data Streams





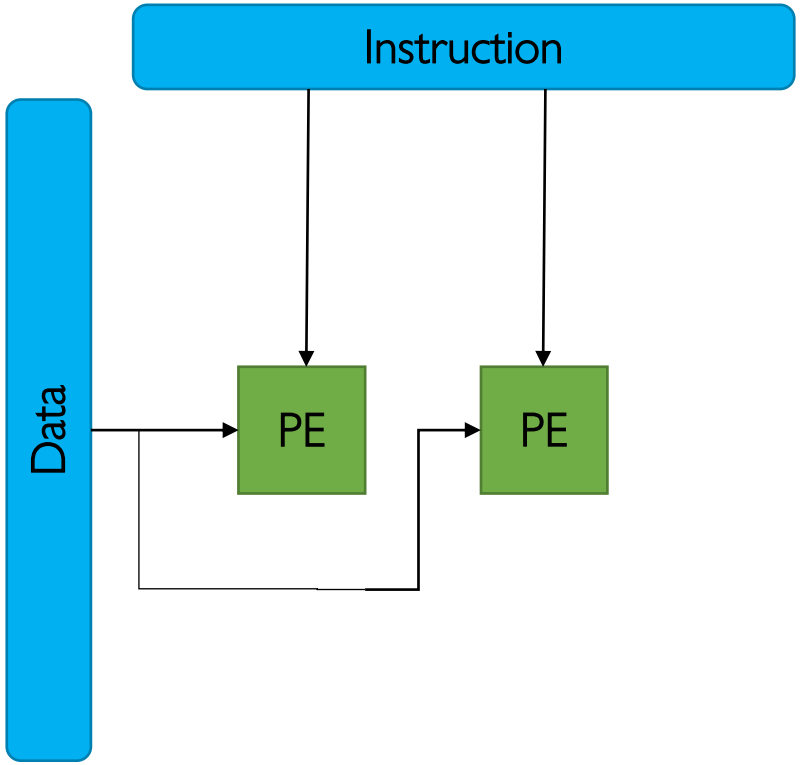


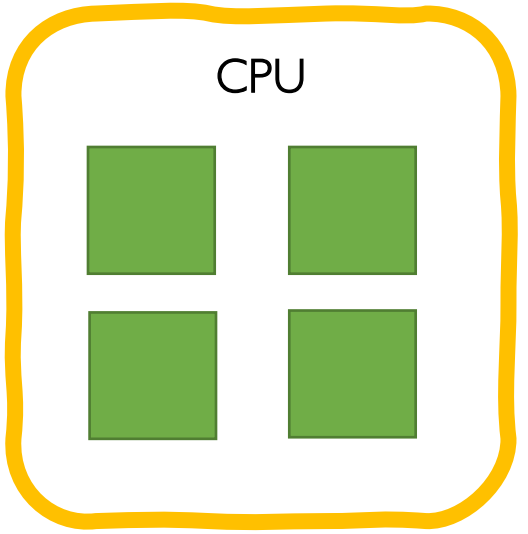
Most modern CPUs now have SIMD extensions to support SIMD in each core

- Typically only supports floating point arithmetic

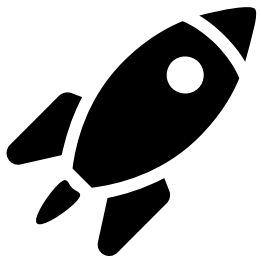
Also known as a vector processor

Multiple Instruction Stream Single Data Streams

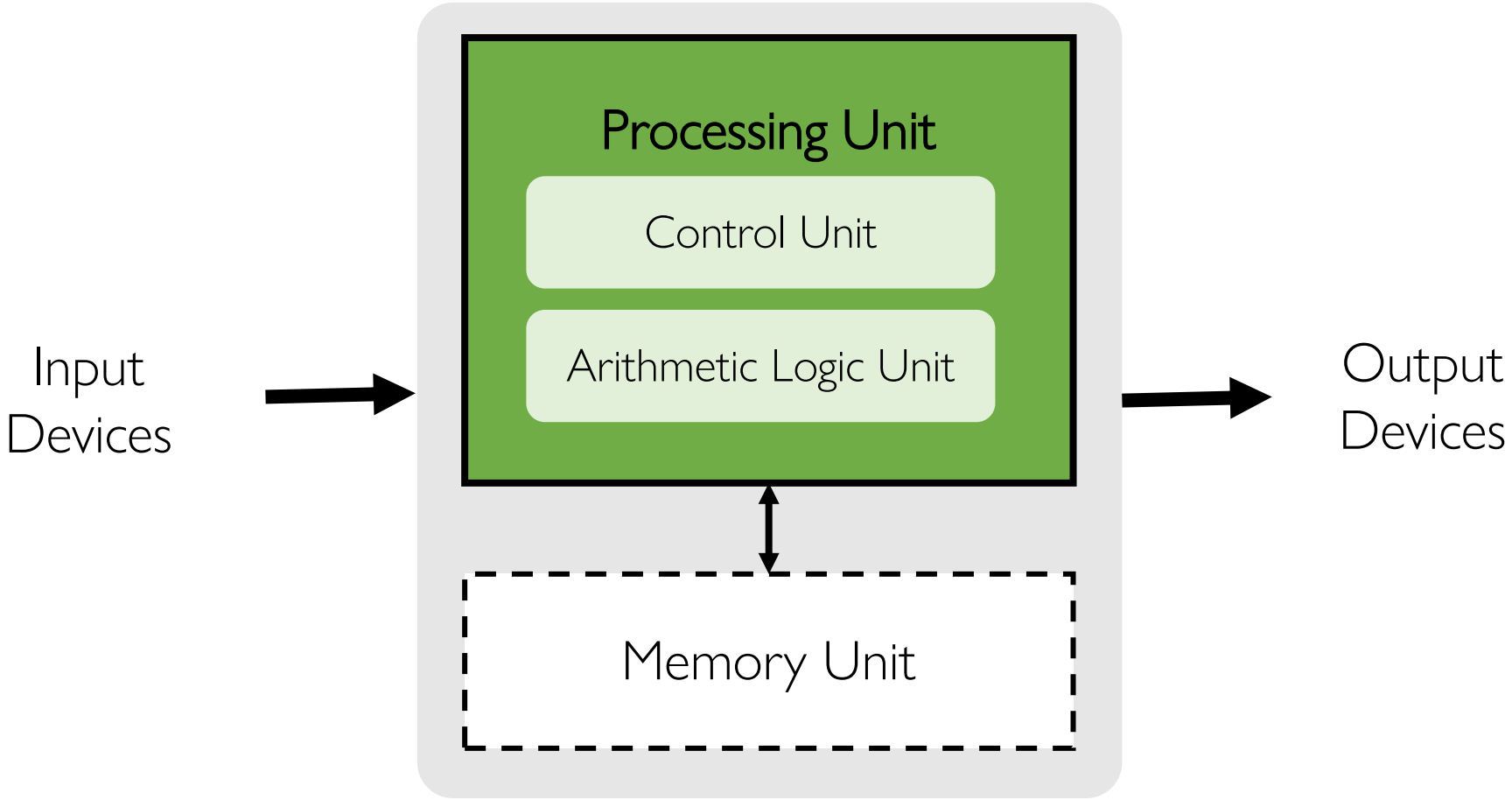




- Very rare
- Famous example is in space shuttle flight control computers

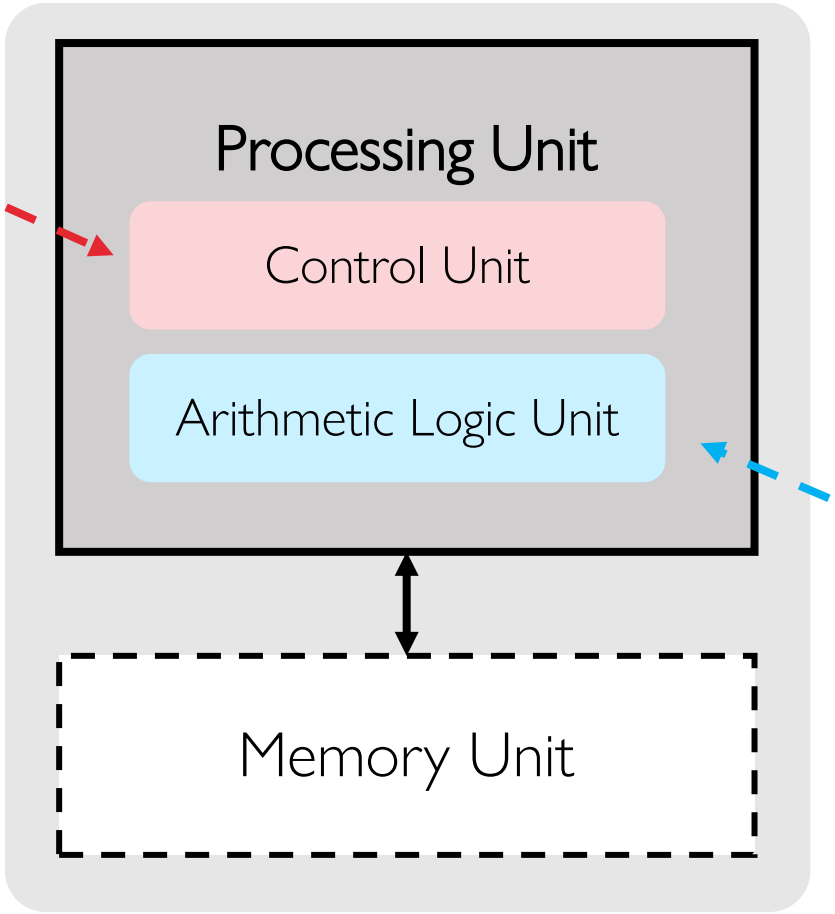


Von Neumann Architecture



Von Neumann Architecture

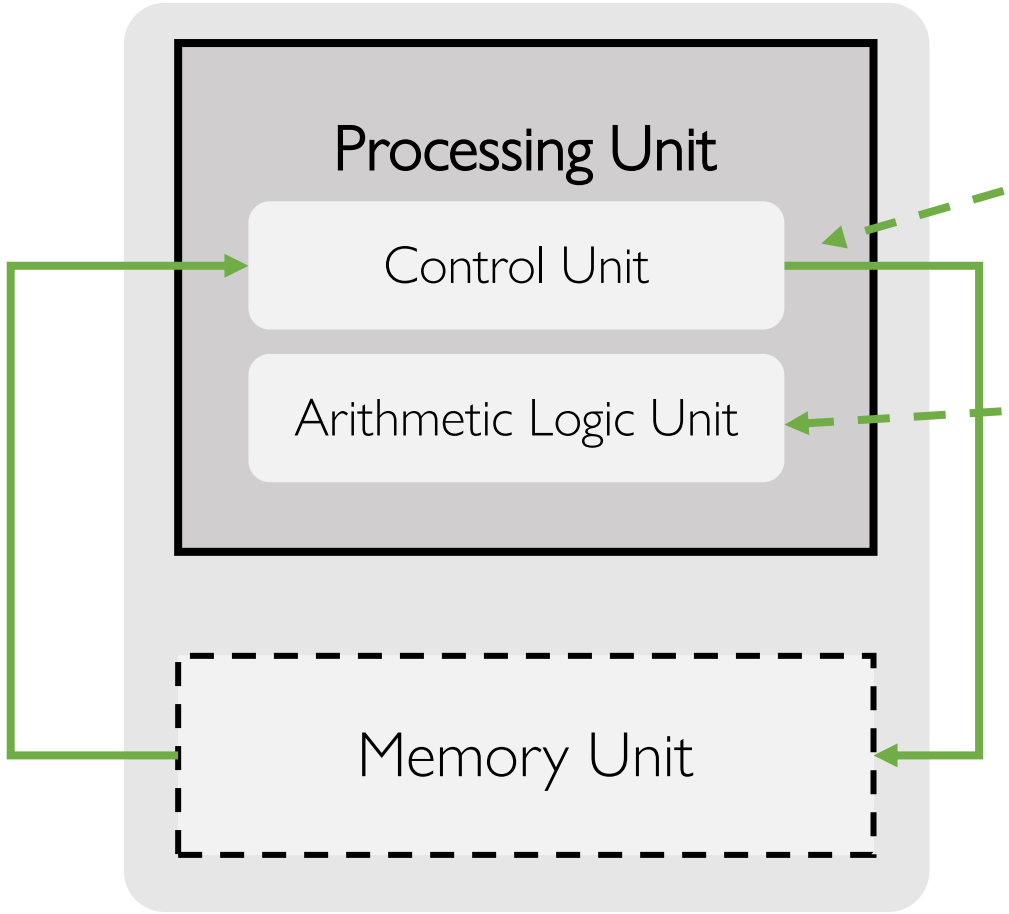
Analyzes instructions and tells other units what to do



Performs arithmetic (+ - x /) and logic (and, or, xor, not) instructions

Simplified Machine Cycle

1 Fetch the instructions



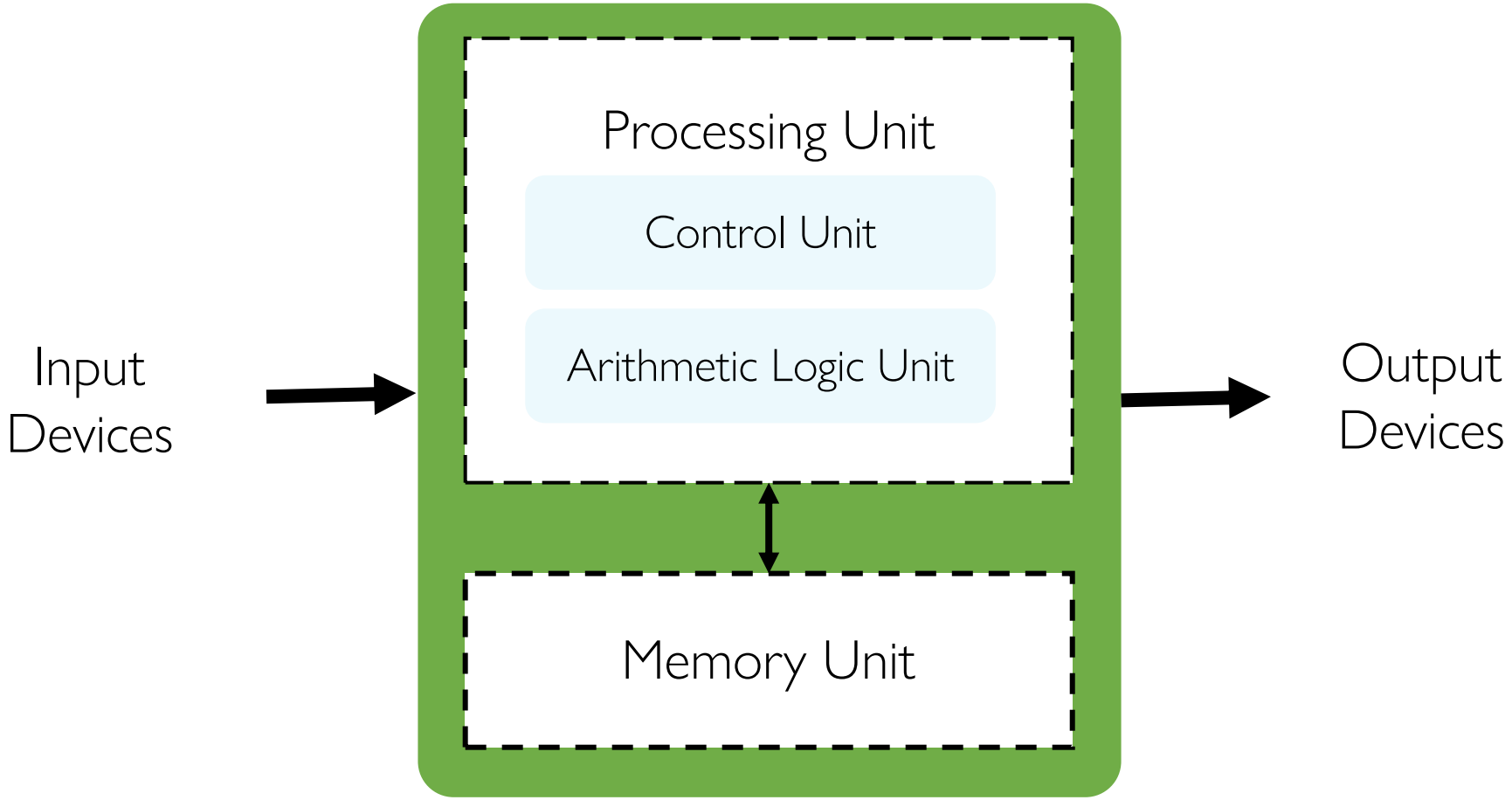
2 Decode the instruction

3 Execute the instruction

4 Store result in the memory

Wrapping Things Up

How we process information today: The Von Neumann Architecture



A hypothetical translation

```
for i=0; i<SIZE; i++
  for j=0; j<SIZE; j++
    for k=0; k<SIZE; k++
        r = r + y[i][k] * z[k][j];
    x[i][j] = r
```

```
LOAD temp1, y[i][k]  # temp1 ← y[i][k]
LOAD r1, y[i][k]     # r1 ← y[i][k]
LOAD temp2, z[i][k]  # temp2 ← z[k][j]
LOAD r2, z[i][k]     # r2 ← z[k][j]
```

```
MULT temp3, temp2, temp1   # temp3 ← temp2 + temp1
MULT r3, r2, r1            # r3 ← r2 + r1
```

```
ADD r, r, temp3    # r ← r + temp3
ADD r4, r4, r3     # r4 ← r4 + r3
```

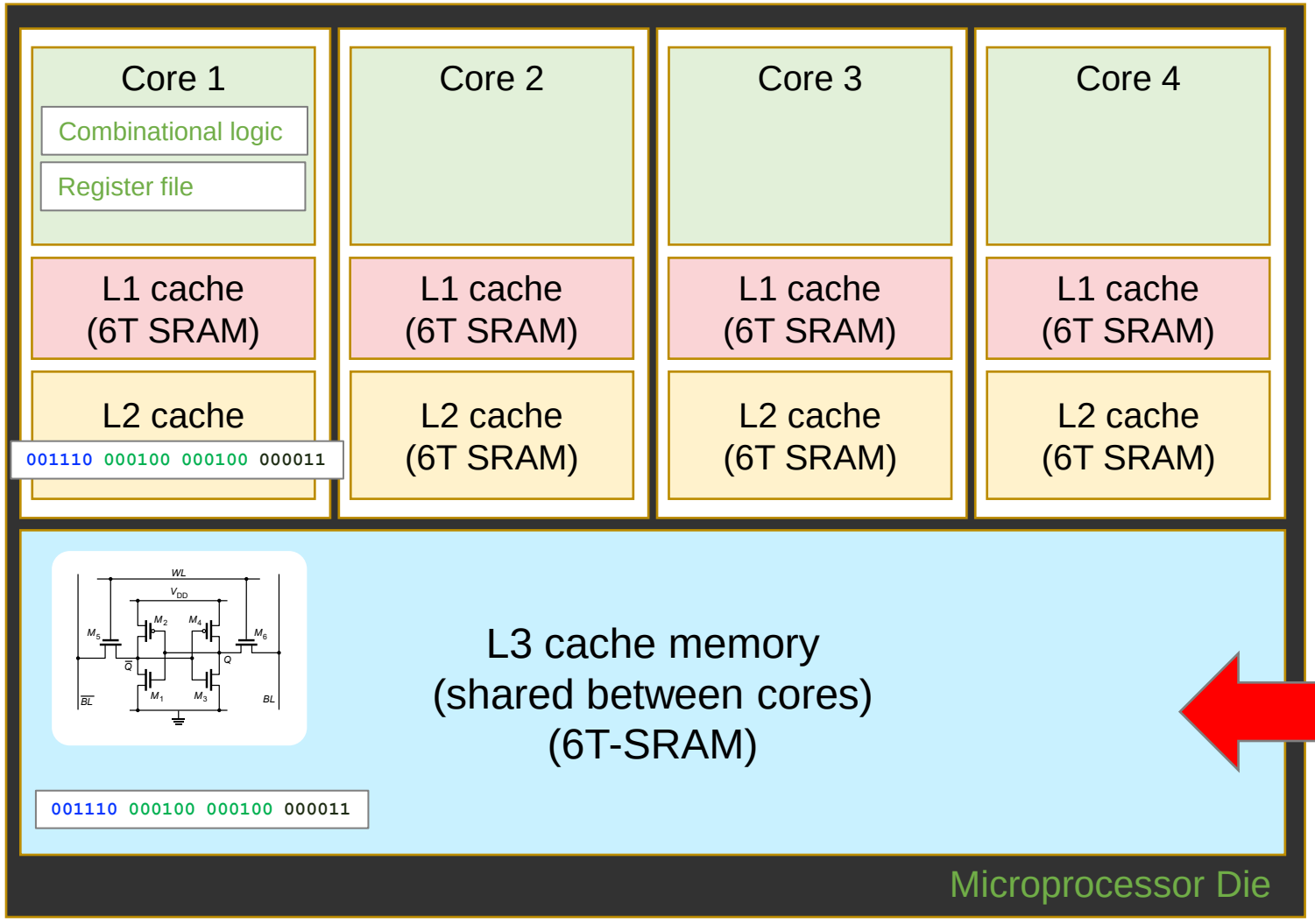
Can define codes for LOAD, MULT and ADD

Assume
LOAD = 111111,
MULT = 110011,
ADD = 001110

stored program becomes

PC	111111	000001	address of y[i][k]	
PC+4	111111	000010	address of z[k][j]	
PC+8	110011	000011	000010	000001
PC+12	001110	000100	001000	000011

A von Neumann Architecture (in a nutshell)

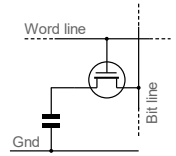


Code

```
for i=0; i<SIZE; i++
  for j=0; j<SIZE; j++
    for k=0; k<SIZE; k++
          r = r + y[i][k] * z[k][j];
    x[i][j] = r
```

Data

```
y[0][0]  y[0][1]  y[0][2]  y[0][3]
y[0][4]  y[0][5]  y[0][6]  y[0][7]
```

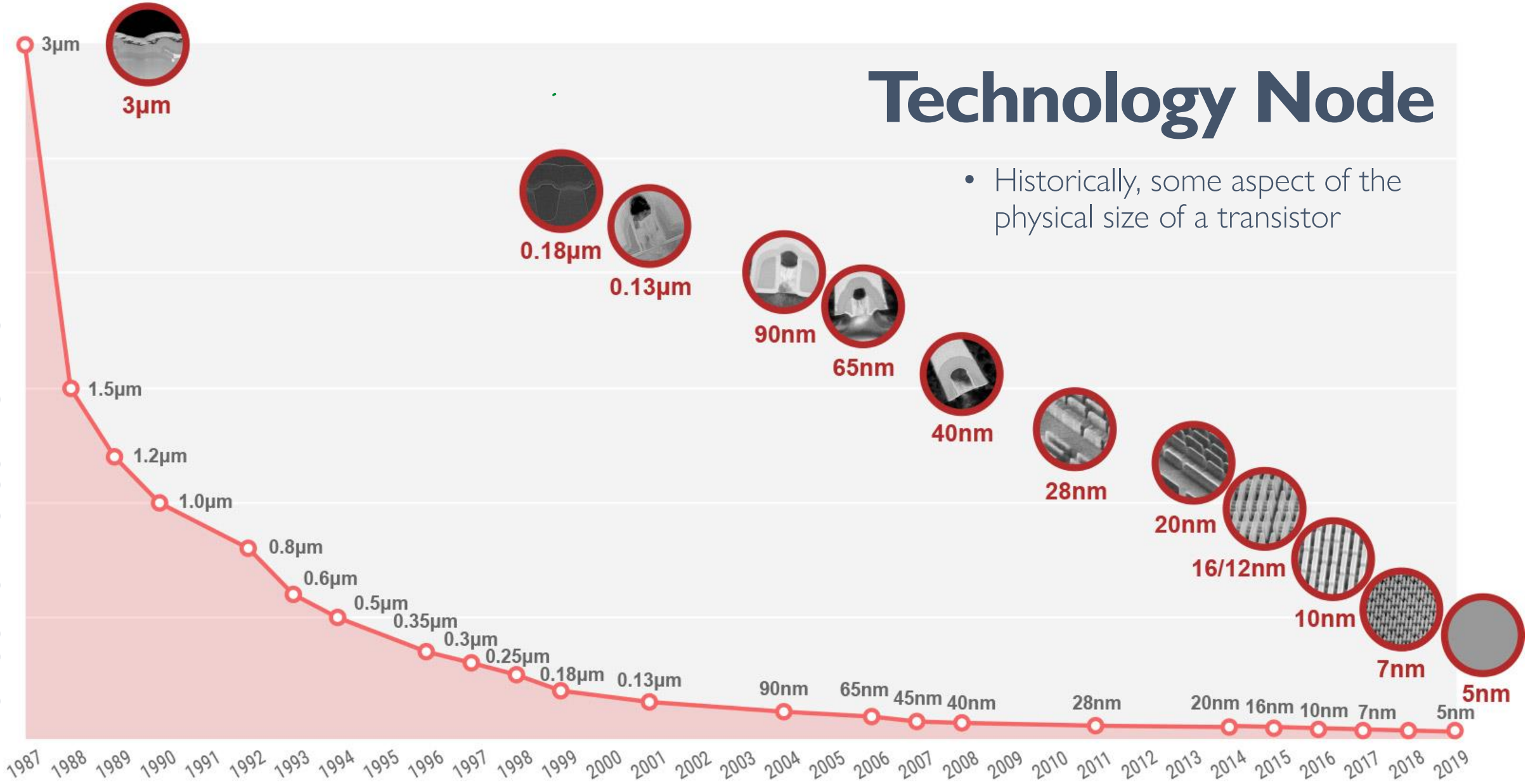


Off-chip memory (1-transistor DRAM)

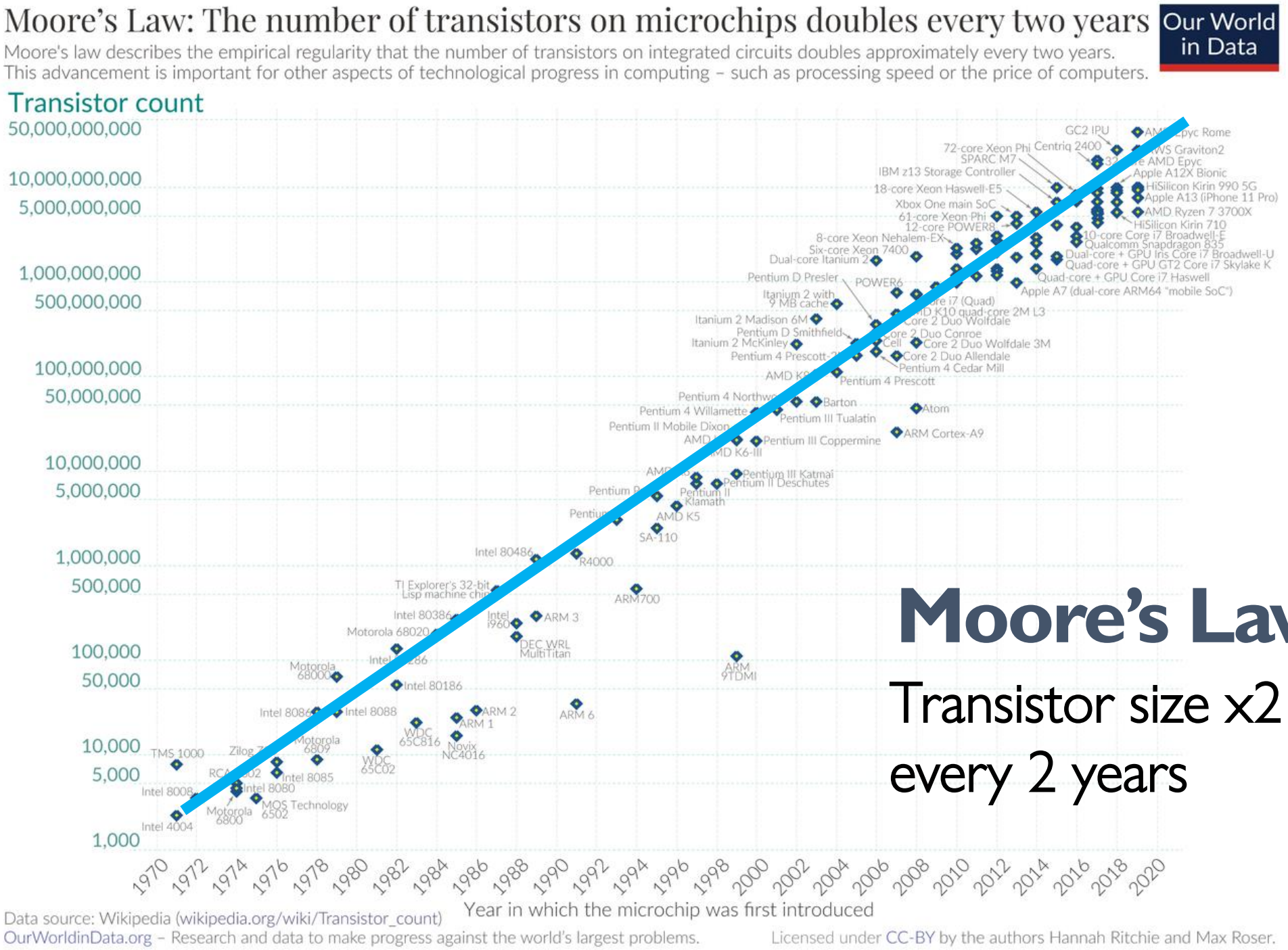
`r + y[i][k] * z[k][j];` | `ADD r4, r4, r3` | `001110 000100 000100 000011`

Technology Node

- Historically, some aspect of the physical size of a transistor



https://www.tsmc.com/english/dedicatedFoundry/technology/logic


Moore's Law: The number of transistors on microchips doubles every two years
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.
Our World in Data
Transistor count
50,000,000,000
10,000,000,000
5,000,000,000
1,000,000,000
500,000,000
100,000,000
50,000,000
10,000,000
5,000,000
1,000,000
500,000
100,000
50,000
10,000
5,000
1,000
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
Year in which the microchip was first introduced
Intel 4004
Intel 8008
Intel 8080
Motorola 6800
MOS Technology 6502
TMS 1000
Intel 8085
Motorola 6809
WDC 65C02
Intel 8088
Intel 8086
Intel 80186
Motorola 68000
Intel 80286
Motorola 68020
Intel 80386
Intel i960
ARM 3
DEC WRL MultiTitan
ARM 1
ARM 2
WDC 65C816
Novix NC4016
ARM 6
ARM700
ARM 9TDMI
Intel 80486
R4000
TI Explorer's 32-bit Lisp machine chip
AMD K5
SA-110
Pentium
Pentium II Klamath
Pentium II Deschutes
Pentium III Katmai
Pentium II Mobile Dixon
AMD K6-III
Pentium III Coppermine
Pentium 4 Willamette
Pentium 4 Northwood
Barton
Pentium III Tualatin
Atom
ARM Cortex-A9
Pentium 4 Prescott
Pentium 4 Cedar Mill
Core 2 Duo Allendale
Core 2 Duo Wolfdale 3M
Core 2 Duo Conroe
Core 2 Duo Wolfdale
AMD K10 quad-core 2M L3
Core i7 (Quad)
Itanium 2 with 9 MB cache
Itanium 2 Madison 6M
Pentium D Smithfield
Itanium 2 McKinley
Pentium 4 Prescott-2
Cell
POWER6
Pentium D Presler
Dual-core Itanium 2
8-core Xeon Nehalem-EX
Six-core Xeon 7400
12-core POWER8
61-core Xeon Phi
Xbox One main SoC
18-core Xeon Haswell-E5
IBM z13 Storage Controller
72-core Xeon Phi
SPARC M7
GC2 IPU
Centriq 2400
AWS Graviton2
AMD Epyc Rome
AMD Epyc
Apple A12X Bionic
HiSilicon Kirin 990 5G
Apple A13 (iPhone 11 Pro)
AMD Ryzen 7 3700X
HiSilicon Kirin 710
10-core Core i7 Broadwell-E
Qualcomm Snapdragon 835
Dual-core + GPU Iris Core i7 Broadwell-U
Quad-core + GPU GT2 Core i7 Skylake K
Quad-core + GPU Core i7 Haswell
Apple A7 (dual-core ARM64 "mobile SoC")
Moore's Law
Transistor size x2
every 2 years
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Commercial Success due to:

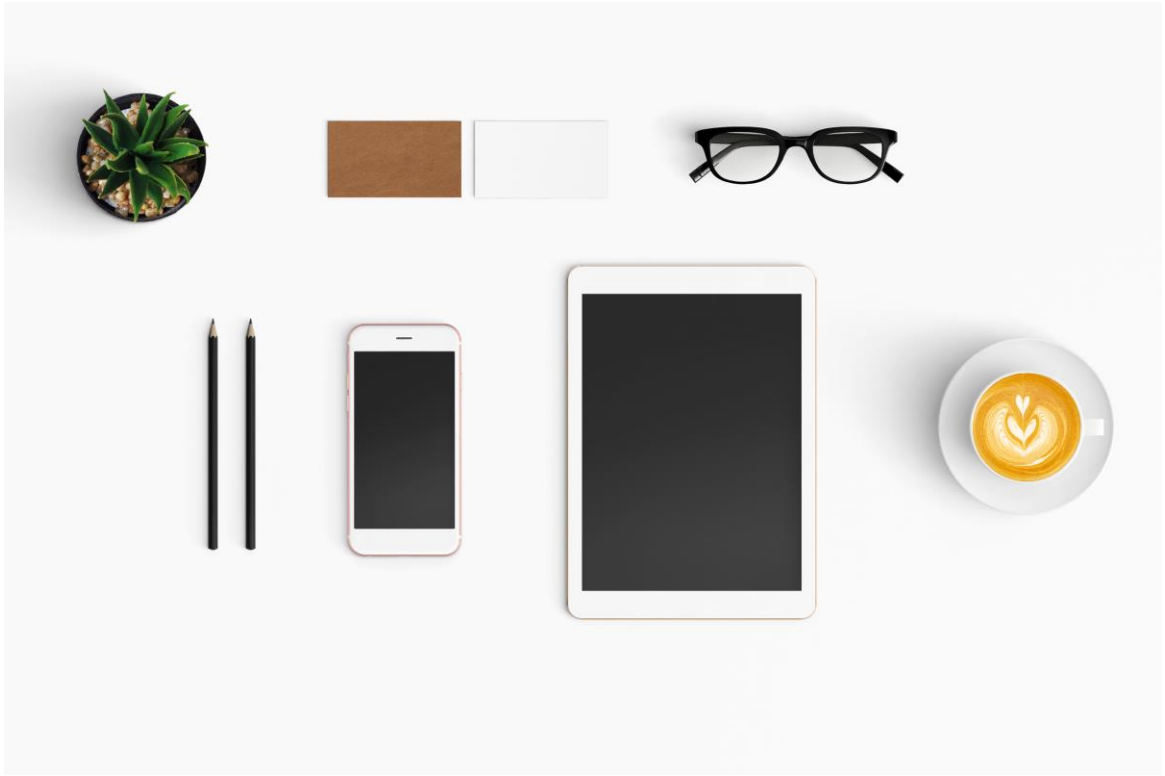
- Elimination of assembly language
- Creation of a standardized operating system

RISC: Reduced Instruction Set Architecture

- Instruction Level Parallelism
- Caches

Effects

- Availability
- New classes of computers
- Software development
- Multimedia applications



Ann Franchesca Laguna

Canvas

ann.laguna@dlsu.edu.ph

THANK YOU

Ann Franchesca Laguna

Copyright

- These slides are created by the following people:
  - Ann Franchesca Laguna