* Computer Design
* Instruction Pipelining
* Multiple CPU Architectures
* Co-Processors.

Instruction Cycle

			Cycles
Fetch	MAR ← PC, PC ← PC+1	MDR ← M[MAR], IR ← MDR.	3
Decode			1
Execute	Evaluate address		1
	Fetch operands (3+)		0..
	Execute		1
		min	6 9+ max

(p1)

Average : 6 cycles / instruction.

CPU : 3.0 GHz ⟶ 3×10^9 cycles/second

∴ $\frac{3}{6} \times 10^9$ instructions / second

$= 0.5 \times 10^9$ or 500 million/second.

Assuming ~10 instruction/HLL statement you get 50 million/second.

Speeding up Techniques

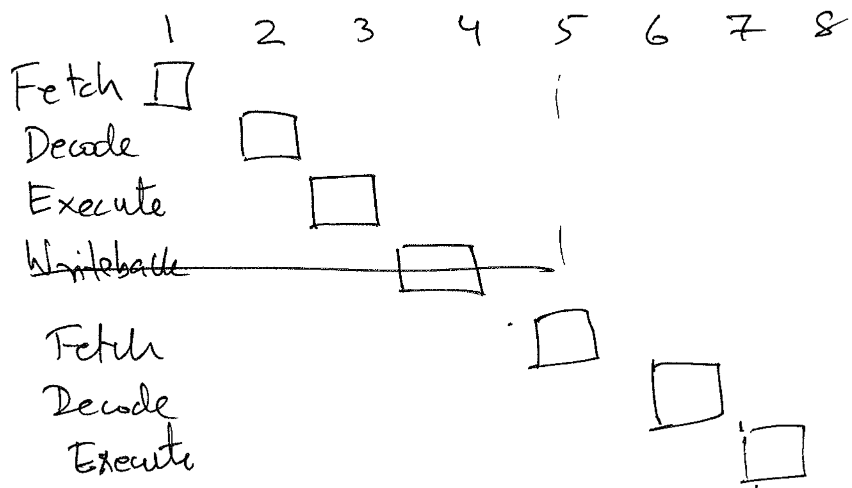
1. Instruction Pipelining
2. Use Multiple CPUs
3. Use co-processors
   - FPU – Floating Point Unit
   - GPU – Graphics Processing Unit.

(P2)

Instruction Piplelining

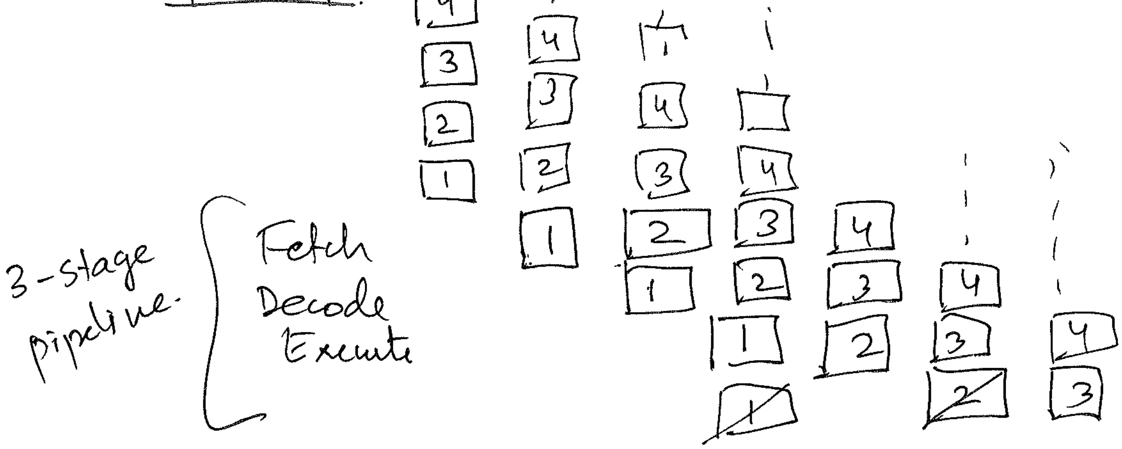
Instructions flow through a cpu in stages.

P3



Pipelining

P4



---

6 cycles @ 4 instructions!

With a 5 stage pipeline CPUs can do 1 instruction/cycle

use up & down tick of clock.

Current CPUs can do 32 instruction/cycle!

P5

So our computer w/ 500 million/instruction/second can do 16 billion/second!

500 x 32 = 16000

Issues w/ branching!

② Multiple CPUs

3 concepts – threads, cores, CPU architectures

Threads – a single strand of computation

(P1)

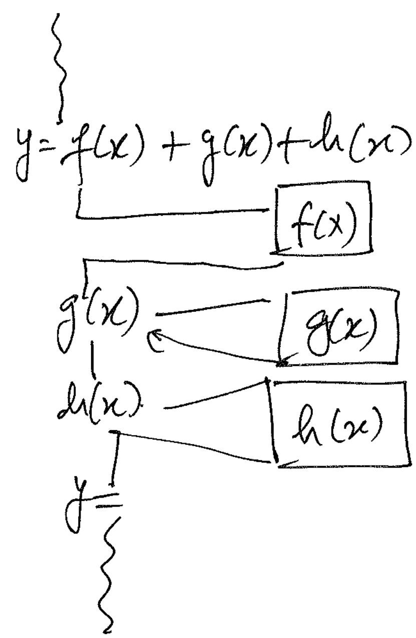
LC-3 Program

START ← single thread (CPU is executing 1 instruction of 1 program)

END HALT

Java/C: $y = f(x) + g(x) + h(x)$

(P2)

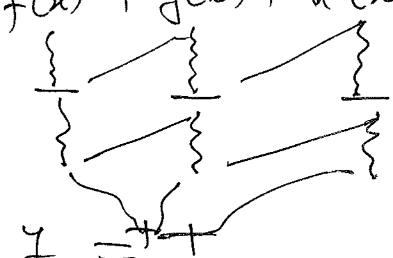


Multiple threads

(P3)

Can do multiple threads on 1 CPU

$y = f(x) + g(x) + h(x)$



CPU uses slicing to allocate some time to each thread.

Still takes the same amount of time as single thread! (actually, a little more) due to context switching.

Another Example

spotify

WebBrowsing + listening to music + ~~down~~

rendering page + File/Image downloading — ~~down~~ buffering

RW — D — B

CPU | RW | D | B | D | RW | D | B | - - - - |

Sometimes also called multiprogramming.
(several threads from diff programs running on a CPU)

P4

With multiple CPUs

We can run multiple programs at any given time.

- Say, we have 8 CPUs (Cores), we can have 8 threads

P5

Core 1	Web browsing	
Core 2	Downloading	also Zoom Backgrounds
Core 3	Spotify.	
Core 4	idle ...	
⋮		

Each core can also be running multiple threads.

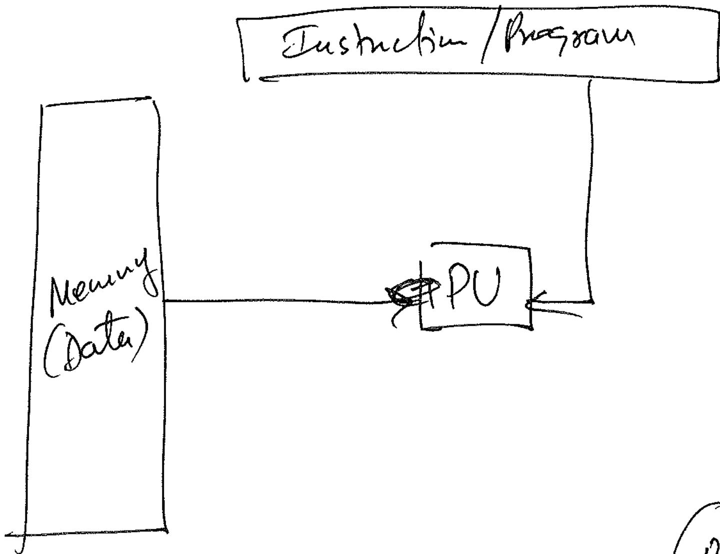
Computer Architecture Possibilities

Flynn's Taxonomy

- SISD – Single Instruction Single Data Stream — LC-3
- SIMD – Single Instruction Multiple Data Stream — Array Processor (NVIDIA)
- MISD – Multiple Instruction Single Data Stream — Rare shuttle flight → fault tolerant
- MIMD – Multiple Instruction Multiple Data Stream → Multi-core processors

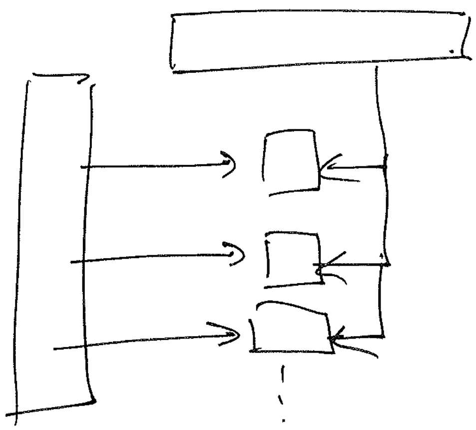
P1

SISD



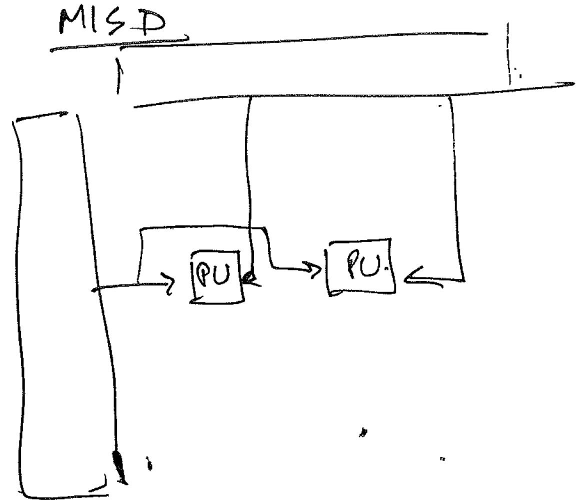
P2

SIMD



Array Processors
GPUs.

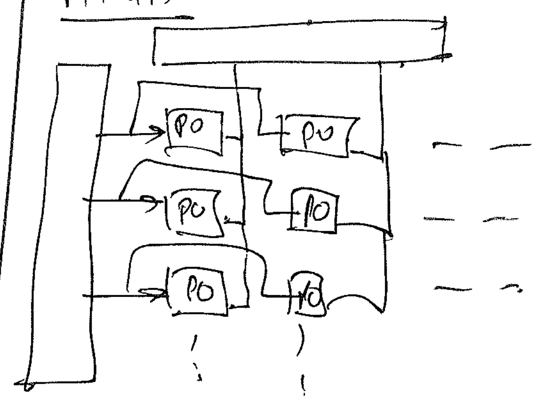
MISD



P3

Rare:
Used in fault tolerant systems: like space shuttle flight controller.

MIMD



Multicore processors.

P4

Example: Intel Core i7-9700 CPU @ 3.GHz 8 cores
~~each core~~ · RAM 32 GB max 128 GB
64-bit

At 32 instructions/cycle on 3.0 GHz CPU
$3 \times 32 \times 10^9$ instructions/second. = 96 billion ips
Times 8 ~ 1 trillion/second.
NO!

Better measurement of computing power is done using performance benchmark tasks.
e.g. Linpack
~~100 × 100~~ solve 1000 linear equations
use 1000 × 1000 matrix of floating pt. values
typically addition + multiplication only.

FLOPS – floating point operations/second.
KFLOPS – kilo 10^3
MFLOPS – Mega (10^6)
GFLOPS – Giga (10^9)
TFlops – Tera (10^{12})
PFLOPS – Peta (10^{15})
EFLOPS – Exa (10^{18})

June 2023
FRONTIER
@ Oakridge National Labs in Tennessee
has 8,699,904 cores
gives ~ 3EFLOPS!!

(3) Use Co-Processors

FPU – Floating Point Unit.

- separate processor that handles FP arithmetic.

GPU – SIMD – NVIDIA!

③ **Example**: Why we need many algorithms

sort 10,000 000 number.

Computer A	Computer B
Speed: 10^{10} instr/sec	Speed: 10^7 instruction/sec

Insertion Sort: $O(n^2)$

$\sim 2n^2$ instructions

$$\frac{2 \times (10^7)^2}{10^{10}} \simeq 20{,}000 \text{ seconds}$$

~ 5.5 hours

$$\frac{2 * (10^7)^2}{10^7} = 2 \times 10^7 s$$

~ 6000 hrs

Quick sort: $O(n \log n)$

$\sim 50 \log n$

$$\frac{50 \log(10^7)}{10^{10}}. < 2 \text{ seconds!}$$

$$\frac{50 \times 10^7 \times \log 10^7}{10^7} = 1163 \text{ secs}$$

< 20 min.