MAJOR TRENDS:

→ Scaling laws breaking down

→ Energy efficiency issues

SOLUTION:

→ SPECIALIZATION:

* Limit hardware capabilities
* Expose hardware to software using domain-specific interface.

WHY USEFUL FOR DEEP LEARNING:

→ Parallelism

→ Regular memory access

→ Precision not needed

→ Most values are 0.

→ Weights, inputs reused.

DENNARD'S LAW:

Decrease area, voltage of transistor → * Current decreases
* Increases frequency (faster chips)
* Power dissipation decreases
* Power density = $\frac{\text{Power dissipation}}{\text{Area}}$ = Constant.

So, we use same power for a given area, but now that area of transistors is faster!

So for same energy => Faster chips!

This is gone!

Cannot decrease area/size of transistor anymore → ↑ heat

Already 3nm (DNA size range!)

SOLUTION:

1. Increase cores?

Parallel processing with more cores → Faster Chips!

But each core takes E energy. n cores take nE energy. So its not really increased speed with same energy. **Performance / Energy is same!**

2. Specialize

Some generic H/W can be removed. So this can make performance better and decrease energy needs!

REGISTER vs SCRATCHPAD MEMORY vs CACHE vs MEMORY vs DISK

DISKS: Slow, persistent

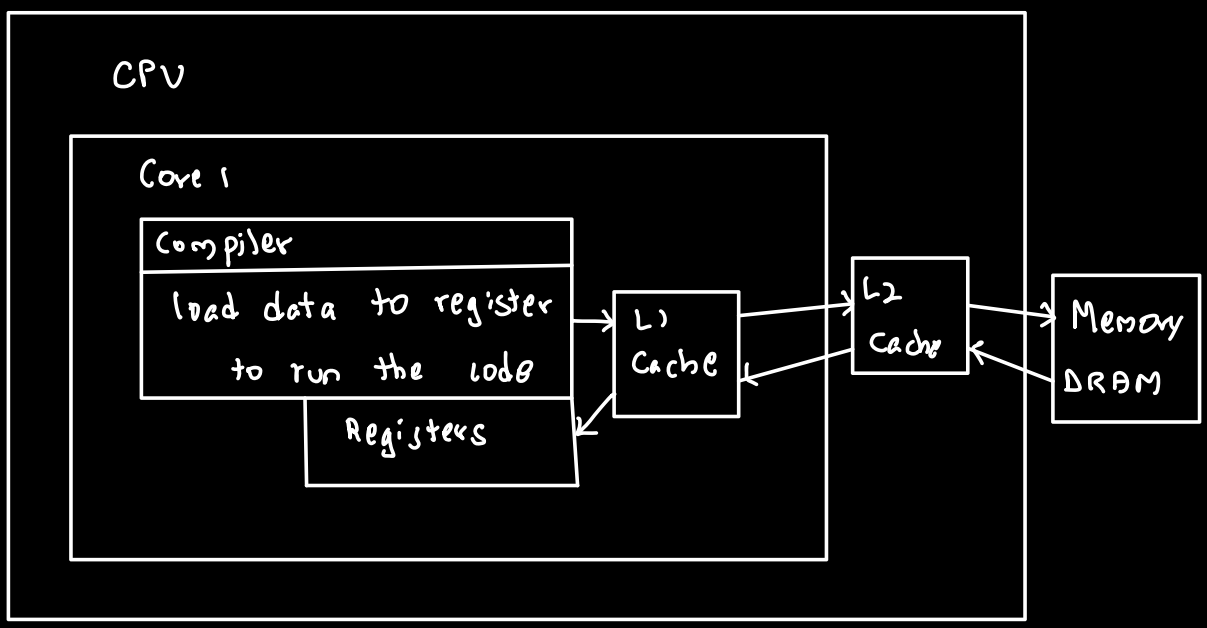
MEMORY: DRAMs

CACHE: SRAMs

Compilers decide what's in the register while H/W decides whats in the cache.

SCRATCHPAD MEMORY:

Similarly, Programmer / Compiler / S/W decides whats in the scratchpad. Other than that it is basically just a cache.


CPU
Core 1
Compiler
load data to register
to run the code
Registers
L1
Cache
L2
Cache
Memory
DRAM

ADVANTAGES OF CACHE:

→ ↓ latency

→ ↓ energy

→ Memory bandwidth is low. Say 8 cores access 1 memory. It cannot take all requests. Cache manages them.