

Memory

Main Memory (RAM)

- Running programs are stored in **RAM**
- Cache memory is stored in **SRAM** (*static*)
- Main memory is stored in **DRAM** (*dynamic*)

Computer memory is broken down into four types, each with a purpose...

```
flowchart LR
    subgraph Memory
        Heap
        Stack
        Global
        Text
    end
    Heap --> dynamic/runtime --> fragmentation
    Stack --> runtime_functions --> overflow
    Global --> static_vars
    Text --> Code
```

Frames

The **Static Text/Code** region of memory is allocated into regions called **frames** where methods and their local variables are stored. These are accessed by method pointers from the **stack**.

```
flowchart RL
    methodPointer1 --> frame1
    subgraph Frames In Text Region
        subgraph frame1
            arg1_1
            arg1_2
        end
        subgraph frame2
            arg2_1
            arg2_2
            arg2_3
        end
    end
    subgraph Stack
        methodPointer1
    end
```

HDL (*Hardware Description Language*)

- Verilog

Processors

Instruction Set Architectures

- Intel x86
- ARM (licensed -> NVIDIA)
- RISC-V (UCB)

Execution Flow

1. PC (*program counter*) **points** to an instruction in memory
2. processor **fetches** the instruction
3. processor **decodes** the instruction
4. processor **executes** the instruction
5. processor **updates** the PC

flowchart LR

```
subgraph Processor_Instruction_Types
    Load
    Store
    Operate
    Jump
end
```

Memory Hierarchy

- **Register**
 - smallest unit of memory
- **Cache**
 - highly reused memory
 - Has levels L1, L2, L3...
- **RAM**
 - main memory. Good speed and storage
- **Local Storage** / SSD
- **Remote Storage** / HDD

Memory is too tricky to interact with directly, *Virtual Memory* gives an abstraction.