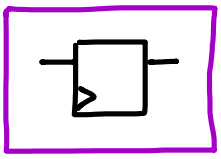
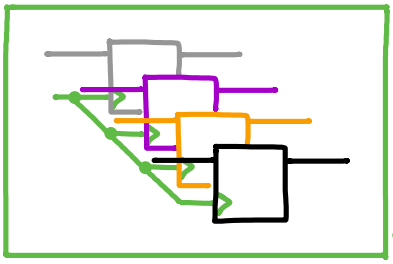
ISA

What is this:



? → a flip flop! Its the smallest form of storage in a computer (stores 1 bit)

What about this?



→ its a register!

→ the size of a register determines how many bits it holds

← this register is 4 bits

What is a register? → a quick access local variable

→ register files are small so look↑ time is short

Each ISA has a unique number of registers & register size

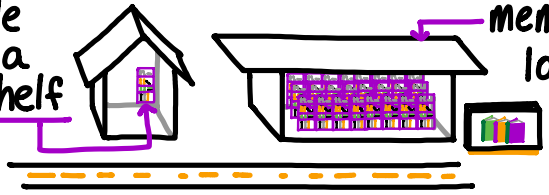
e.g. MIPS ← 32 registers @ 32 bits each ← how many bytes are these?

Tiny ISA ← 16 registers @ 16 bits each

This information is stored in the register file.

A register file is one component of ISA ("macroarchitecture")

a register file is like a home bookshelf



memory is like a library longer to find a book than your home book-shelf

---

What is an ISA?

→ Instruction Set Architecture → an ISA defines the instruction set (the native commands)

→ a well defined hardware/software interface

→ functional definition of storage locations & operations

What is an **instruction set**?

A set of **assembly language** commands that can be computed by the processor

→ how humans interact with 1's & 0's

→ human readable representation

An **assembler** translates assembly to machine code (1's & 0's)

machine language is the binary decoding of assembly instructions

→ often dislayed in Hex

machine code	assembly code	e.g.
x18C4	ADD R4, R3, R4	

→ macro vs micro architecture

what vs how

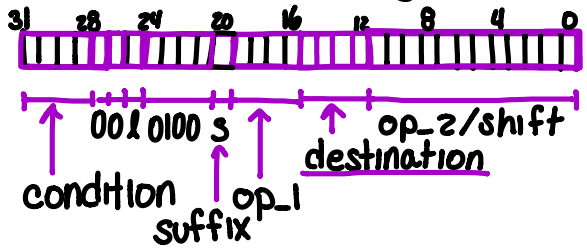
assembly command example

ADD<suffix> <dest>, <op1>, <op 2>

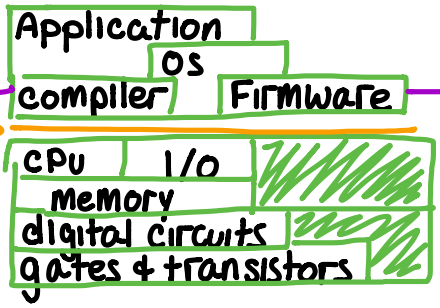
32 bit instruction breaks down as follows:

dest = op 1 + op 2

e.g. ARM has 32 bit instr size



instruction size is also unique to an ISA



Program Compilation

→ compiler translates high level code (c, c++, Java, c#) to assembly

→ compiler optimizes

→ compiler is a program

Instruction Execution Model

- a computer is a finite state machine
  - registers - few but fast
  - memory - lots but slower
  - program counter - next instruction to execute
- a computer executes instructions
- basic process
  1. fetch
  2. decode
  3. read
  4. execute
  5. write
  6. next instr

program is just data in memory