Decode-Remember

- The opcode specifies the instruction format in which the instruction is encoded.

- After decoding the binary code of the instruction, the processor knows what instruction and parameters it is dealing with.

- Example **`ADDI`**:
    - Immediate instructions contain bits that are interpreted as numerical value that is retrieved using shifting operations.
    - 5 bit are enough to encode 32 registers.

immediate	rs1	funct3	rd	opcode
12 bit	5 bit	3 bit	5 bit	7 bit

Execute

- The execution of every RISC-U instruction has a well-defined effect. It changes the state of the machine only at a specific location involving little data.

- At most two registers or one register and one memory location are modified by an instruction.

  - Every instruction modifies the **`PC`**.

  - Instruction which modify **data** (another register or memory location) have trivial control flow (`PC` to next instruction).

  - Control-flow instructions have a more sophisticated **control flow**, that is, the change the `PC` using relative or absolute addressing.

Decode-Remember

- The opcode specifies the instruction format in which the instruction is encoded.

- After decoding the binary code of the instruction, the processor knows what instruction and parameters it is dealing with.

- Example **`ADDI`**:
  - Immediate instructions contain bits that are interpreted as numerical value that is retrieved using shifting operations.
  - 5 bit are enough to encode 32 registers.

immediate	rs1	funct3	rd	opcode
12 bit	5 bit	3 bit	5 bit	7 bit

Decoding Instructions

- Each instruction can be uniquely identified by certain parts of the 32 bits called **opcode**, **funct3** and **funct7** code.

- When the instruction is known, the meaning of the remaining bits of the instruction becomes clear.