CPU:

- ALU ✅
- Control Unit (Sends commands to *Memory Management Unit* and the *ALU*, reads from/writes to *RAM Unit*, and writes to *Registers* for temporary program storage)
    - 28 bits of Program Counter command memory
- Registers
- Memory Management Unit
- Clock ✅

RAM:

- Writing ✅
- Reading ✅
- Unit ✅

Program Reader/Counter:

- Pointer (Program Reader/Counter will output what it reads into the *Control Unit*) 🟨
- Set to start of program
- Reset to start of program

Input:

- Input Buffer
    - Reserve a cell that can be typed in and will be read at the *last character in the cell* when Program Counter reaches a `Read Input` instruction
      -> use in OS dev
      -> possibly also game dev?
    - "read input" instruction will keep a copy of the cell before an input to check if an input happened, will update upon reading a new input
        - Add a `Clear Cell` *VBA macro* that clears the *Input Buffer* (the cell where inputs happen) when the clear command is given by the *Program Counter* to the *Command Unit* (*might be cheating*)

Output:

- Screen (Gets inputs from *Control Unit*)

ROM:

- Writing ✅
- Reading ✅
- Unit

Assembly/BF:

- Parser
    - Instruction set
        - `>>` -> Push to stack
        - `<<` -> Pop from stack
        - `%` -> Alias call
            - Jump to memory address of Alias
        - `$` -> System call
            - Pass as a Control Unit command and not a Program Counter command
        - `DPND`
            - Pulls another program into the Registers for use as a function
        - `END`
            - Signal the end of a program
- Compiler
    - eBF instructions -> 4-bit numbers
    - Byte-sized parameters (8-bit numbers)
        - 3 parameters max