Computers are Factories, and Microprocessors are Tooling.

Definition of Tooling:

* “Tooling” refers to tools expressly developed and used for manufacturing or crafting.
* Tooling is an important subset of technical instruments.
* Technical instruments designed for broadly defined space of intentions
* The designers of tooling do not precisely specify what the tools are intended to do.
* The scope of a given tool’s abilities are well-defined, but it is designed with a flexibility of application. Flexibility is a driving intention.
  + This could mean flexibility of function, and of configurability.
* Like machine tools, computers are tooling.
* While they are marketed as products, each computer is a factory.
  + This goes from workstations to phones to gaming consoles, down to simple programmable logic controllers. Programmable hardware is electronic tooling, possessing a pronounced level of agency and flexibility that differentiates it from other technical artifacts.

Software as a Make Plan:

* Conventional computers are limited by the instruction sets of their processing units, but these are formal constraints rather than practical ones. A Turing-complete system can approximate any function, so a computer is only limited by its outputs and timescale.
* In the space of all possible software, we have an infinitely-extensible make plan.
  + Read back to the MP paper to get some arguments going

Hardware & Agency

* Most matter is programmable, and all tools are (the human swinging the hammer is the software) but tooling has been fashioned expressly to be programmable.
* Computers store and sculpt information. The logic gate is the cutter, memory is the fixture, instructions are the machines, and software is the make plan.
* CPUs don’t require that many degrees of freedom to maintain flexibility. x86 ISA has vastly more operations than RISC-V, and we are seeing a rise in specialized ICs that are designed to fulfill more specific intentions (e.g. tensorrent’s RISC-V hardware).
* The push towards specialization is including a lot of inference-only hardware, interesting in that it has a strictly defined capability to run software (neural networks) that are defined by their flexibility to approximate any function, and have been shown to be Turing-complete without requiring access to external memory.