0.4 A Minimalist's Computing Machine

We are still far from knowledgeable, but we have built enough tools for a most basic computing machine (which is even simpler than ancient computers) to be introduced in this section. The objective of the transition between computing introduced in Section 0.1 and computer here is that, we humans no longer take the heavy duty of computing, but assign it to a surrogate named **computer**. The design behind the scene is to automate the human thinking process of computing effectively (achievable) and efficiently (quick) [1].

Using all the tools introduced so far, we will build a simplest computing machine named **basic counting machine**. The machine is used to do counting on repeated patterns occured more than once. Thus, it can be used to count the number of zebras, lions, humans and many more.

Inside the box of basic counting machine, we have counter (perform plus one task when requested), memory, input, and output. When running the machine, we first use the input to enter existing number (of zebras) before performing the counting task. The input medium can be punch cards, tapes (like the ancient computers) or buttons, keys (like the modern computers). The number will load into memory of the machine. By seeing each new repeated item (1 more zebra), we press the button to the counter once. The counter will increase the number in memory by 1. Once no repeated items have been spotted (no more zebras to be count), we use the output to retrieve number stored in memory and clear the memory. The output medium can be punch cards, tapes (like the ancient computers) or screens, printers (like the modern computers). This basic counting machine works as the simplified version of the first general-purpose electronic computer ever in history (1945), named *ENIAC* - Electronic Numerical Integrator and Computer.

The process of input numbers, press the counter button and derive output is called **program**. As explained in Section 0.3, program is an language of representations. Specifically, program in computer science is an representation language to be understandable by both computing machine operators and the machine itself. Here we can finally define software from hardware, where software is the language, and hardware is the computing machine itself [2].

To perform a small alternation to our basic counting machine, we can store the "counting" operation together with input number in the same memory device. In this way, the machine allows the users to perform more types of computations other than counting without altering the physical buttons/wires [3]. This model is called stored-program model or *von Neumann machine* [4], which is in the designs of every single computer since then (1945) until today.

1. Note that the minimalist's computing machine introduced in this section is still an abstract machine like Turing machine (an hypothetical machine or a mathematical model instead of a real achievable machine like an ancient computer). An achievable toy computer needs more background kownledge and is beyond the scope of this first chapter.↩

2. Precisely, there are a good number of "language" of representations within computing machines (both ancient and modern ones). The division between hardware and software seems to be obvious but actually quite vague. Both hardware and software are general terms in natural language for humans to intuitively classify the physical elements and the programs/instructions of computers.↩

3. If designed properly with registers, program counters and more switches.↩

4. Used in IAS machine built in the Institute for Advanced Study, Princeton. The inventor of stored-program model, though credit to von Neumann, has long been controversial.↩