Notes

Dr Greig Russell

25 April, 2019

Contents

List of Figures

1 Introduction

2 The modern computer

In information science, there is a constant dance between the applied and theoretical branches of the discipline. The work of one underpins the next advance in the other. While electronic computers were created to meet human needs when those needs exceeded the capacity of individual humans, information science preceded and enabled their development. The focus for advancing the capacity of computers or their application to solving human problems ever since has been about developing the hardware and software together with theoretical paradigms. To understand information there requires an understanding of computers.

Simply put, a computer is the combination of its hardware and software, and its current state of development is an iterative development from the original concepts developed during and before WW2.

2.1 The hardware

In 1945, Von Neumann [1993] first described the components of what he termed "very high speed automatic digital computing system" (section 1.1), and it has become the standard architecture of the computer. The concept has not changed since, even if the component level details have changed dramatically. The aim of this computer, called the EDVAC, was to solve non-linear partial differential equations (Von Neumann [1993], section 1.2).

The current unified concept of a Central Processing Unit (CPU) served two functions. The first was an arithmetic processor to complete basic mathematical operations. For @Von Neumann [1993] this also included more complex functions such as sin, cos, log and square root functions. (section 2.2). The second component was the logic controller (section 2.3). The logic controller was responsible for the orderly execution of a specific set of instructions. The central logic controller was responsible for the orderly operation of the computer as a whole, including hardware components, regardless of any specific program that it was running. It was this latter feature that made this the architecture for a general computer, not dedicated to a specific task like some of the earlier machines (such as the "Colossus" of Bentley Park and the WW2 code breakers fame).

Key to the new design for the EDVAC was the provision of "considerable" memory (section 2.4). Although hilariously small at 34KB by today's standards, such memory allows for the solution to be broken into components for efficient problem solving within a supervising logic tree. It also allowed for interim states of variables to be available between components to be saved (Von Neumann [1993], section 2.4). Von Neumann [1993] also saw that such generalised computers could be useful for statisticians in regards to the managing of large data samples.

The computer will require input devices (Von Neumann [1993], section 2.6), such as punch cards, and recording this information by use by the computer. Note, for this architecture, the recorded information is not available for the computer. Hence, the computer needs a means of moving the data into the working memory or CPU (section 2.7). Equally, a means of moving the results back to be recorded (section 2.8) and then outputting the results in human-readable forms, such as on a teletype (section 2.9)

2.2 The software

Alan Turing outlined the fundamental functional concepts of a computer in his landmark 1937 paper "On computable numbers, with an application to the Entscheidungsproblem" (Turing [1937]). The EDVAC was supposed to enable such functions to occur within a general computer, as Turing described.

Turing's computer was not the focus of the paper. Instead, it was a thought experiment that served as a vehicle to address the real issue the Entscheidungsproblem or "halting" problem. The description has served as the basis for the archetype for what a computer needs to have to be a complete general computer. The term "Turing complete" has been coined to describe any such computer (or language on modern computers

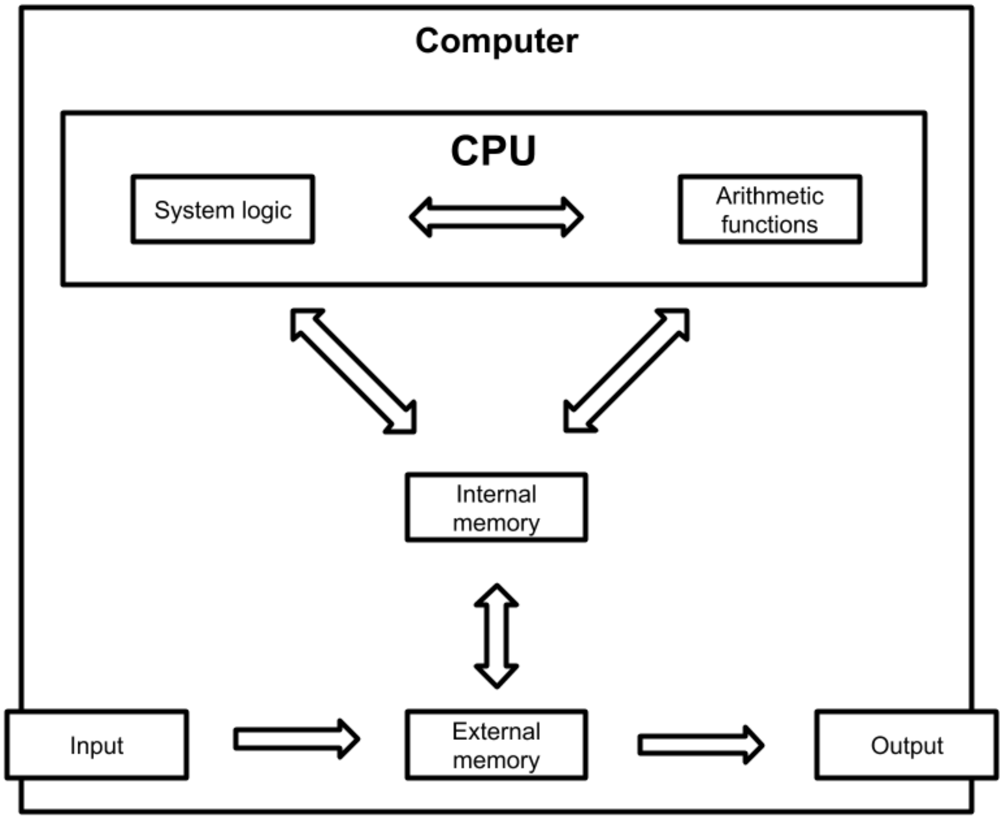

Figure 1: Von Neumann Architecture

with the abstraction of the programming logic from the hardware into the software) that can perform all the functions of a Turing machine.

The components of a Turing machine are the machine itself and paper tape. The former consists of a reader/writer head and a set of possible configurations. The latter consists of a series of symbols. The machine reads a symbol, and in combination with the current state of the machine performs the specified action. This action may be to change the symbol on the tape or move to another place on the tape or even do both (Turing [1937]).

The paper tape describes the initial state of such a machine. When the program has reached its end following a series of transformations, the tape will now describe the effect of the instructions on the original state, namely the answer.

This machine was used by Turing to determine if all such algorithms, which called themselves recursively, would eventually holt. For everyone else, Turing's machine is the blueprint of a computer program, mounted on hardware that follows Von Neumann's architecture.

2.3 The algorithm is the truth

2.4 Turing machine

2.5 Lambda calculus & functional programming

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

3 The arms race

3.1 Moores law

3.2 Hardware, software and OS evolution

3.3 Data still static

4 Collaboration

4.1 Shannon's communication theory

4.2 The internet

4.3 Fake news

5 The ontology of information

5.1 Dretske

5.2 Carnap and Bar Hillel

6 The lost counterfactual

6.1 Lewis

7 Conclusion

Bibliography

Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

John Von Neumann. First draft of a report on the edvac. *IEEE Annals of the History of Computing*, 15(4): 27–75, 1993.