FOUNDATIONS OF COMPUTING P/T

READING 3 & 4

THE STORED-PROGRAM COMPUTER

The modified extract below is taken from:
http://www.turing.org.uk/turing/scrapbook/computer.html
It is from Alan's Turing's homepage, written by his biographer, Andrew Hodges
---

The Alan Turing Internet Scrapbook

Who Invented the Computer?
Alan Turing's Claim

Scrapbook Index

What is a Computer?

The machine which is sitting in front of you.

The machine which can draw graphics, set up your modem, decipher your PGP, do typography, refresh your screen, monitor your keyboard, manage the performance of all these in synchrony... and do all of these through a single principle: reading programs placed in its storage.

But the meaning of the word has changed in time. In the 1930s and 1940s "a computer" still meant a **person** doing calculations. There is a nice historical example of this usage here. So to indicate a **machine** doing calculations you would say "automatic computer". In the 1960s people still talked about the **digital computer** as opposed to the **analog computer**.

The world's computer industries now make billions out of manufacturing better and better versions of Turing's universal machine. But Alan Turing himself never made a point of saying he was first with the idea. And his earnings were always modest. *Picture from a Japanese comic book of Turing's story.*

But nowadays, I think it is better to reserve the word "computer " for the type of machine which has swept everything else away in its path: the computer on which you are reading this page, the digital computer with "internally stored modifiable program. "

So I wouldn't call Charles Babbage's 1840s Analytical Engine the design for a computer. It didn't incorporate the vital idea which is now exploited by the computer in the modern sense, the idea of storing programs in the same form as data and intermediate working. His machine was designed to store programs on cards, while the working was to be done by mechanical cogs and wheels.

There were other differences — he did not have electronics or even electricity, and he still thought in base-10 arithmetic. But more fundamental is the rigid separation of

instructions and data in Babbage's thought.

Charles Babbage, 1791-1871

- Charles Babbage (Wikipedia)
- Virtual Museum of Computing
- The completion of the Difference Engine in the Science Museum, London
- Home page maintained by his biographer A. Hyman.

A hundred years later, in the early 1940s, electromagnetic relays could be used instead of gearwheels. But no-one had advanced on Babbage's principle. Builders of large calculators might put the program on a roll of punched paper rather than cards, but the idea was the same: you built machinery to do arithmetic, and then you arranged for instructions coded in some other form, stored somewhere else, to make the machinery work.

To see how different this is from a computer, think of what happens when you want a new piece of software. You can download it from a remote source, and it is transmitted by the same means as email or any other form of data. You may apply an UnStuffIt or GZip program to it when it arrives, and this means operating **on** the program you have ordered. For filing, encoding, transmitting, copying, a program is no different from any other kind of data — it is just a sequence of electronic on-or-off states which lives on hard disk or RAM along with everything else.

The people who built big electromechanical calculators in the 1930s and 1940s didn't think of anything like this. I would call their machines near-computers, or pre-computers: they lacked the essential idea.

More on near-computers, war and peace

ENIAC

Even when they turned to electronics, builders of calculators still thought of programs as something quite different from numbers, and stored them in quite a different, inflexible, way. So the ENIAC, started in 1943, was a massive electronic calculating machine, but I would not call it a computer in the modern sense, though some people do. This page shows how it took a square root — incredibly inefficiently.

Colossus

The Colossus was also started in 1943 at Bletchley Park, heart of the British attack on German ciphers (see this Scrapbook page.)

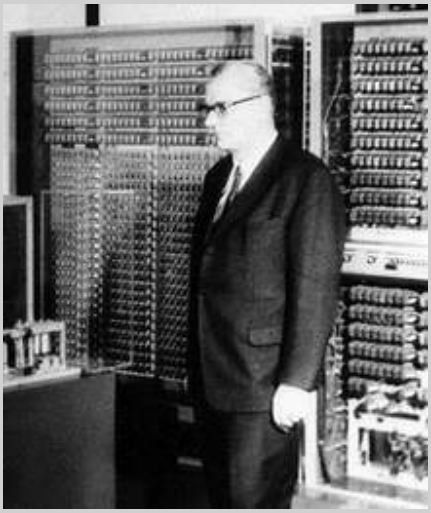
I wouldn't call it a computer either, though some people do: it was a machine specifically for breaking the "Fish" ciphers, although by 1945 the programming had become very sophisticated and flexible.

But the Colossus was crucial in showing Alan Turing the speed and reliability of electronics. It was also ahead of American technology, which only had the comparable ENIAC fully working in 1946, by which time its design was obsolete. (And the Colossus played a part in defeating Nazi Germany by reading Hitler's messages, whilst the ENIAC did nothing in the war effort.)

1996 saw the fiftieth anniversary of the ENIAC. The University of Pennsylvania and the Smithsonian made a great deal of it as the "birth of the Information Age". Vice-President Gore and other dignitaries were involved. Good for them. At Bletchley Park Museum, the Reconstruction of the Colossus had to come from the curator Tony Sale's individual efforts. Americans and Brits do things differently. Some things haven't changed in fifty years.

Zuse's machines

Konrad Zuse, in Germany, quite independently designed mechanical and electromechanical calculators, before and during the war. He didn't use electronics. He still had a program on a paper tape: his machines were still developments of Babbage-like ideas. But he did see the importance of programming and can be credited with a kind of programming language, Plankalkül. It is not difficult to believe that with greater support, he would have gone far ahead with the theory and practice of computing.



Konrad Zuse, 1910-1995, with the Z3.

Like Turing, Zuse was an isolated innovator. But while Turing was taken by the British government into the heart of the Allied war effort, the German government declined Zuse's offer to help with code-breaking machines.

The parallel between Turing and Zuse is explored by Thomas Goldstrasz and Henrik Pantle.

Their work is influenced by the question: was the

computer the offspring of war? They conclude that the war hindered Zuse and in no way helped.

In contrast, there can be no question that Alan Turing's war experience was what made it possible for him to turn his logical ideas into practical electronic machinery. This is a great irony of history which forms the central part of his story. He was the most civilian of people, an Anti-War protester of 1933.

He was very different in character from John von Neumann, who relished association with American military power. But von Neumann was on the winning side in the Second World War, whilst Turing was on the side that scraped through, proud but almost bankrupt.

The Internally Stored Modifiable Program

The breakthrough came through two sources in 1945:

- Alan Turing, on the basis of his own logical theory, and his knowledge of the Colossus.

- the EDVAC report, by John von Neumann, but gathering a great deal from ENIAC engineers Eckert and Mauchly. Download the EDVAC report in pdf form.

They both saw that the programs should be stored in just the same way as data. Simple, in retrospect, but not at all obvious at the time.

John von Neumann, 1903-1957

John von Neumann (originally Hungarian) was a major twentieth-century mathematician with work in many fields unrelated to computers.

- Wikipedia article
- MacTutor mathematical biography
- Tools for Thought, by Howard Rheingold

The EDVAC report became well known and well publicised, and is usually counted as the origin of the computer in the modern sense. It was dated 30 June 1945 — before

Turing's report was written. It bore von Neumann's name alone, denying proper credit to Eckert and Mauchly who had already seen the feasibility of storing instructions internally in mercury delay lines. (This dispute has been revived in the book *ENIAC* by Scott McCartney. This strongly contests the viewpoint put by Herman Goldstine, von Neumann's mathematical colleague, in *The Computer from Pascal to von Neumann.*)

Where did von Neumann get the idea?

He knew Turing at Princeton in 1937-38 (see this Scrapbook Page.) By 1938 he certainly knew about Turing machines. See these source documents on what von Neumann knew of Turing, 1937-39. Many people have wondered how much this knowledge helped him to see how a general purpose computer should be designed.

The logician Martin Davis, who was involved in early computing himself, has written a book *The Universal Computer, The Road from Leibniz to Turing.* Martin Davis is clear that von Neumann gained a great deal from Turing's logical theory.

So who invented the computer?

There are many different views on which aspects of the modern computer are the most central or critical.

- Some people think that it's the idea of using electronics for calculating — in which case another American pioneer, Atanasoff, should be credited.

- Other people say it's getting a computer actually built and working. In that case it's either the tiny prototype at Manchester, (See this Scrapbook Page) or the EDSAC at Cambridge, England (1949), that deserves greatest attention.

But I would say that in 1945 Alan Turing alone grasped everything that was to change computing completely after that date: above all he understood the universality inherent in the stored-program computer. He knew there could be just one machine for all tasks. He did not do so as an isolated dreamer, but as someone who knew about the practicability of large-scale electronics, with hands-on experience. From experience in codebreaking and mathematics he was also vividly aware of the scope of programs that could be run.

The idea of the universal machine was foreign to the world of 1945. Even ten years later, in 1956, the big chief of the electromagnetic relay calculator at Harvard, Howard Aiken, could write:

```
If it should turn out that the basic logics of a machine designed for
the numerical solution of differential equations coincide with the
```

```
logics of a machine intended to make bills for a department store, I
would regard this as the most amazing coincidence that I have ever
encountered.
```

But that is exactly how it has turned out. It **is** amazing, although we now have come to take it for granted. But it follows from the deep principle that Alan Turing saw in 1936: the Universal Turing Machine.

Of course, there have always been lousy predictions about computers.

Computer History on the Web

The Computer Museum History Center has an extensive overview of computer history.

There are many other pages on the history of the computer. Some portals are:

- History of computing (Wikipedia)
- The modern history of computing, Stanford Encyclopedia of Philosophy
- the Virtual Museum of Computing
- International Federation for Information Processing
- the Yahoo listing
- Professor J. A. N. Lee.

But you should watch out — a great deal of published material is highly misleading. An example would be the Turing entry on Professor Lee's site. This writer fails to distinguish the Universal Turing machine from the general Turing machine concept; he believes the Colossus was used by Turing on the Enigma and was 'essentially a bunch of servomotors and metal' and he mentions Alan Turing being gay as if a 'rumour'. But because this was one of the earliest documents on the Web, it has frequently been copied and cited by people who should know better.

Behind these confusions there lies a basic disagreement about whether the computer should be placed in a list of *physical objects* — basically the hardware engineers' viewpoint — or whether it it belongs to the history of *logical, mathematical and scientific ideas,* as logicians, mathematicians and software engineers would see it.

I follow the second viewpoint: the essential point of the stored-program computer is that it is built to implement a logical idea, Turing's idea: the Universal Turing machine of 1936. Turing himself referred to computers (in the modern sense) as 'Practical Universal Computing Machines'.

From Theory to Practice: Alan Turing's ACE

This Scrapbook page has emphasised the importance of Turing's logical theory of the Universal Machine, and its implementation as the computer with internally stored program. But there is more to Turing's claim than this.

He designed his own computer in full detail as soon as the Second World War was over.

What Alan Turing planned in the autumn of 1945 was independent of the EDVAC proposal, and it looked much further ahead.