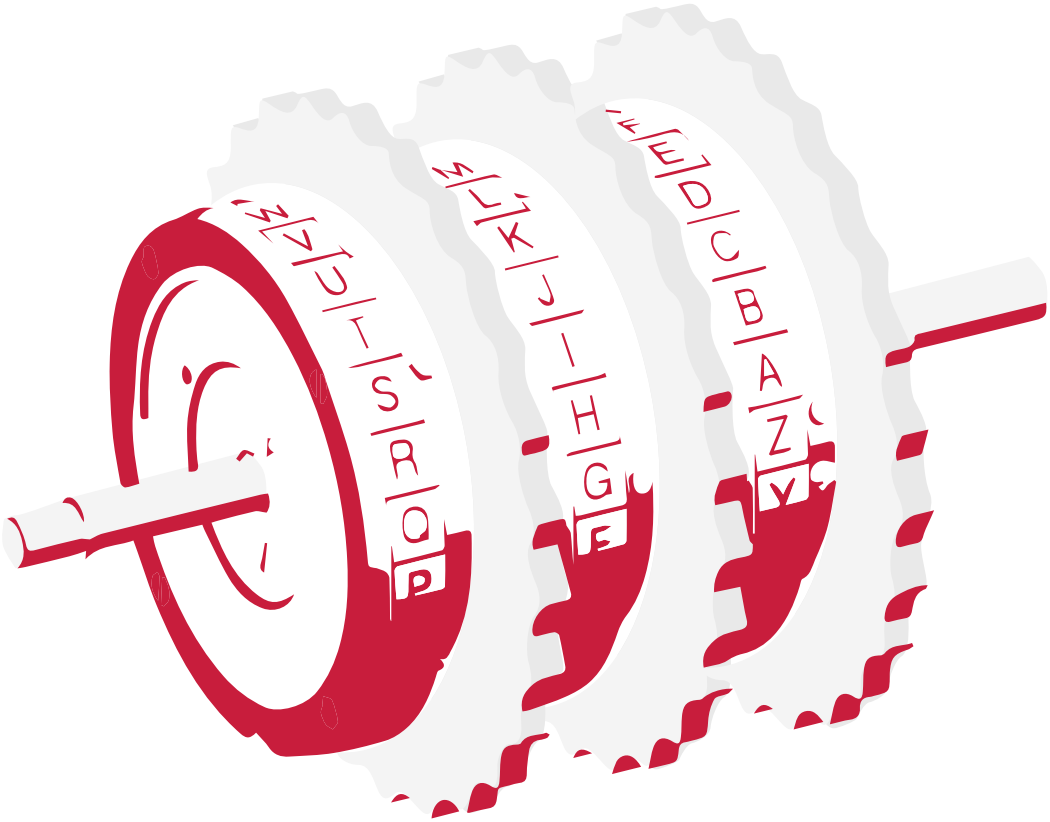
Alan Turing, Colossus and Turing Machines

We're taking a break from our Olde Code series this issue to get some background on a man who was in the news as recently as last year. **Juliet Kemp** looks back at the early works of Alan Turing.



Alan Turing was born in London in 1912, studied mathematics at King's College, Cambridge, and was elected a fellow there in 1935. He worked on the Entscheidungsproblem, then spent two years studying maths and cryptology at Princeton. While there he also built part of an electro-mechanical binary multiplier. This type of machine – a computer, but not a programmable one – was the state of the art in computer hardware at the time. In 1938, Konrad Zuse would complete the Z1, the first mechanical binary programmable computer, in Berlin, although the Z1 was not a general purpose machine as it had no loop capacity.

When WWII began, Turing, who was already working part-time with the Government Code and Cypher School (GC&CS), reported to Bletchley Park to work full time on cryptanalysis. His work on deciphering Enigma, on the Bombes, and his contributions to the development of Colossus, were a hugely valuable part of the history of early computing; they also remained secret until the 1970s.

Turing machines

In 1928, mathematician David Hilbert posed the Entscheidungsproblem (Decision Problem): does an effective procedure exist that would demonstrate whether or not a given mathematical statement is provable from a given set of axioms? In 1931, the Austrian mathematician Kurt Gödel demonstrated that any arithmetic system must be incomplete (that

"A universal Turing machine is one that can compute any computable sequence. Modern computers are all universal Turing machines, and we take this idea for granted now."

is, it is possible to construct a statement that can be neither proved nor disproved), but did not tackle the provability problem.

In 1936, Turing wrote "On Computable Numbers with an Application to the Entscheidungsproblem", which showed that no such procedure exists. He used the idea of an "a-machine", now known as a "Turing machine". This is a hypothetical computing device, which reads an infinite tape. At any one moment the machine reads a single symbol from the tape. It may alter that symbol, and the symbol may (combined with the machine's instruction table) affect its behaviour. The tape moves backwards and forwards, so any symbol may eventually be read by the machine. Turing equated the problem of deciding whether a Turing machine halts on a given algorithm to the Entscheidungsproblem. He proved that some processes will never halt; and thus that the answer to the Entscheidungsproblem is 'no'.

A universal Turing machine is one that can compute any computable sequence. Modern computers are all universal Turing machines, and we take this idea for granted now. At the time, however, it was a huge breakthrough, which arguably led to the idea of stored-program computers.

Bombes and Enigma

One of the simplest ciphers is a substitution cipher: each letter is substituted with another letter. This is readily breakable, but it becomes less so if you use a different substitution alphabet for each letter of the message. The Enigma machine, invented at the end of WWI, had a system of rotors which both mechanised this process (making it easier to operate), and increased the number of cipher alphabet options. Each rotor had 26 positions, and they were connected in series, so each letter was transformed multiple times. Further, the rings stepped onwards for every letter (how often the rotors stepped varied, but at least one rotor would step at least once for each letter). This meant that each letter was encrypted with an entirely new cipher from a huge number of options. Finally, a 'plugboard' swapped some pairs of letters before they were output to confuse matters further (though in fact, contrary to expectation, this made breaking it slightly simpler).

Before WWII, the Poles had already had some success with breaking Enigma messages with their bomba cryptologiczna machine. However, as the Germans introduced more rotors and more plugboard settings to their Enigmas, these machines couldn't keep up. The Bletchley Park codebreakers needed to up their game to decrypt Enigma traffic.

Turing was also an accomplished athlete, with a personal best marathon time of 2 hours 46 minutes.

On the shoulders of Polish giants

Turing took the bomba cryptologiczna and improved on it to create the Bombe. A standard British bombe contained rotors to the tune of 36 Enigma equivalents, enabling it to work very rapidly. The bombe input was a crib (a fragment of probable plaintext), which was tested against the ciphertext. The bombe electronically performed various logical deductions based on this, and if a contradiction arose (which it would do with most possible settings), that crib could be discarded. Only a few settings would then be left for the cryptanalysists to look at in more detail. This was an electronic computer in one sense, but it wasn't programmable; it performed just one task. The cribs were obtained by taking advantage of various regularities in the messages transmitted, such as weather reports and message setting information.

Turing's colleague, Gordon Welchman, later implemented the 'diagonal board') improvement. The

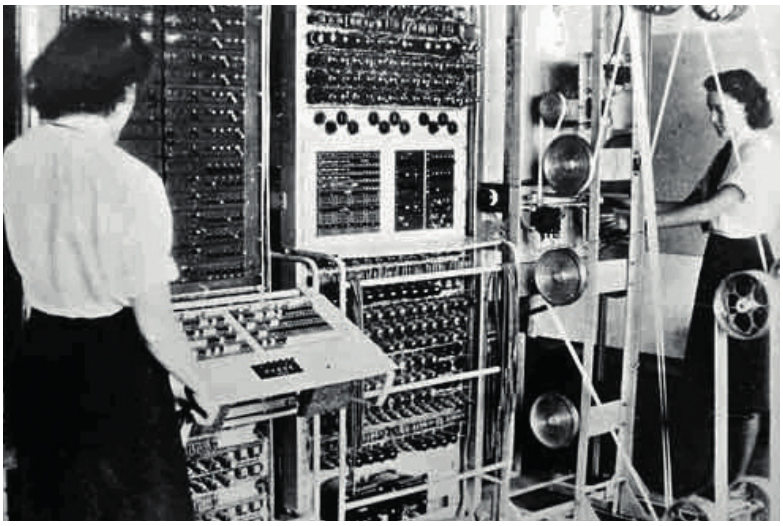
bombes were immensely successful, but initially Turing and his colleagues could not get the resources for more bombes and more people. Eventually they went against military procedure and contacted Churchill directly; resulting in Churchill giving the highest priority to support for the codebreaking team at Bletchley Park.

Turing also worked specifically on naval Enigma, which he started on "because no one else was interested in it so I could have it all to myself". The chief difficulty here was that the sender enciphered the message settings (the information about which rotor settings the message was encrypted with) twice, once by Enigma and once by hand using 'bigram tables' (tables of letter pairs). Turing deduced how this system worked, but was not able to move further before the Royal Navy got hold of some actual bigram tables.

"Information about Colossus began to emerge in the late 1970s, and GCHQ released a 1945 report on the breaking of the Tunny cipher in 2000."

The Colossus machine was designed by Tommy Flowers to help with the cryptanalysis of the Lorenz cipher. The Lorenz cipher machines were used for high-level wireless traffic between German High Command in Berlin and army commands throughout Europe, during World War II, whereas, as discussed above, the more portable Enigma machines were used for other German army and naval messages. It has often been stated that Turing was involved with the development of Colossus; in fact, this was true only in that his statistical methods were part of its ancestry and the cryptanalysis methods that prompted its building.

Colossus Mark 2 being operated by Dorothy Du Boisson and Elsie Booker. The tape transport is shown on the right of the photo.



Rotor encryption

Lorenz worked in a similar way to Enigma, with multiple moving wheels producing a ciphertext. Similarly, one of the routes in to the cryptanalysis was the 'indicator' (showing the start position of the wheels) sent at the start of the message. Being able to identify when two messages had used the same wheel settings gave the codebreakers access to messages in 'depth', which is crucial in codebreaking.

A colleague of Turing's, John Tiltman, managed to identify the cipher used (the Vernam stream cipher) from studying intercepted ciphertexts, and after this, Bill Tutte worked out the logical structure of the machine from further ciphertext study, without ever seeing a Lorenz machine – a hugely impressive achievement. Tutte and his colleagues correctly determined that the machine had two sets of wheels, chi and psi. The chi wheels all moved on one position with each character. The five psi wheels also all moved together, but at a different rate, controlled by two 'mu' motor wheels. This gave a huge number of machine settings and cipher alphabets, which changed in a complicated way.

Turing's main contribution was a process known as Turingery. This revolved around 'differencing', in which he XORed successive characters to emphasise any points at which the characters moved away from a uniform distribution. The next step was a complex statistical analysis of the ciphertext, in which the cryptanalyst tried out a huge number of possibilities and compared the resulting patterns with one another. Eventually, the chi wheel settings could be deduced, and from there the psi and mu settings.

Turingery was a hand method, and slow. Tutte used it as a basis for his own '1+2 break in'. This required trying all possible combinations of the chi wheels against the ciphertext, and looking for subtle statistical evidence of non-uniformity. This used differencing to amplify the effect, and it worked well -- but it was only practical if it could be automated.

Construction time again

So in 1943 Max Newman, with Frank Morrell and Tommy Flowers from the Post Office Research Station, produced a machine known as 'Heath Robinson', using valves and paper tape. It was Flowers who realised that this could be improved on by building an entirely electronic machine (using an electronic key stream rather than reading off paper tape; although the message was still fed in on tape). Most people argued that this would be far too unreliable to be useful, but, supported by the Controller of Research at the Post Office, Flowers went ahead and built it. It first ran in December 1943 and was operational by early February 1944.

This was, then, Colossus, the first programmable (but not general purpose) digital computer. It had four main parts:

- **Tape transport and reading mechanism** Read the message tape in at 5000 characters per second, using the sprocket holes as a clock signal.
- **Key generation unit** Generated an electronic key (chi) stream.
- **Combining unit** Implemented the logic of the 2+1 method.

- **Counting unit** Counted the dots in the output and printed it out if it was over a given total.

It was so successful that they immediately began building more in place of the Robinsons. The Colossuses, and Colossus II (in operation from June 1944, the week before the Normandy landings), were vital for the remainder of the war effort, but after the war, all evidence of the project, physical and paper notes, were destroyed, for security reasons. Despite this, the number of people who had worked on the project and who went on to work in early computers meant that Colossus and the other Bletchley projects did have a significant indirect impact.

The Turing Bombe rebuild project at Bletchley Park. Photograph by Mike Peel (www.mikepeel.net).

Post-war: the Pilot ACE

After the war, Turing worked on the design of the Automatic Computing Engine (ACE) at the National Physical Laboratory. The resultant paper, in 1946, was the first detailed design of a stored-program computer. The ACE implemented subroutines, and even something called Abbreviated Computer Instructions, which was a sort of programming language.

In the ACE, instead of a CPU, memory locations and temporary stores had specific logical functions associated with them. So transferring two numbers to a particular memory location, for example, added them. To speed up executing, Turing suggested that instructions should be stored at specific locations, with each instruction pointing to the next, in such a way as to optimise instruction access. (Experienced UNIVAC programmers did something similar to get around the limitations of mercury delay line memory – see LV002's tutorial on Grace Hopper and UNIVAC.) He also included a small fast-access memory for storing frequently used numbers or ones that needed to be stored temporarily.

Turing and his team, as well as sketching versions of the ACE, also wrote 'instruction tables'. Their aim was for programmers to be able to select groups of standard instructions and link them together with other cards, and they prepared in detail 'instruction routines' including division, extracting square roots, and logarithms. This echoes the work later done by Grace Hopper on UNIVAC, but sadly in Turing's case, his instructions only ever existed on paper.

Turing knew that what he proposed was feasible, and wanted Tommy Flowers to be involved. However, since no one who hadn't been at Bletchley knew about the Colossus, everyone else thought his proposal far too ambitious, and Flowers wasn't recruited. Instead, they eventually built the smaller Pilot Model ACE. This had 1,450 vacuum tubes and 12 mercury delay lines as its memory (each storing 32 bits – again, see the UNIVAC for more on mercury delay lines). Its clock speed was 1MHz, which at the time it first ran (10 May, 1950) was the fastest in the world, and around 10 times faster than its contemporary, the Manchester Mark 1.

The ACE design was also used for the MOSAIC (1952) which calculated aircraft trajectories from radar data (further information is still classified). The first personal computer (well, arguably; it was a small single-user machine), the G-15, built in 1954, also used ACE principles. The commercial version of the Pilot ACE, the DEUCE, was available from 1955 until 1964, and used various languages including one called GEORGE (1957) which used reverse Polish notation and had a 12-position stack. It was sold with an extensive program library of subroutines – perhaps the descendants of Turing's on-paper versions for his full ACE.

ACE program example

Wilkinson's Progress Report on the Automatic Computing Engine (April 1948) includes this code example to calculate squares and cubes of n by iteration until n = m (m is stored in tank TS1):

```
A1  zeros+ zeros -> TS2 Imm 1, A
A3  zeros+ zeros -> TS3 Imm 1, A
A5  zeros+ zeros -> TS5 Imm 1, A
-----------------------------------
A7  TS2 + TS3 -> TS3 Imm 1, A    i.e. n2 + n -> TS3
A9  TS3 + TS4 -> TS4 Imm 1, A    i.e. n3 + 3(n2 + n)
-> TS4
A13 TS2 + P1  -> TS2 Imm 1, A    i.e. n + 1 -> TS2
A15 TS3 + TS2 -> TS3 Imm 1, A    i.e. (n2 + n) + (n + 1) -> TS3
A17 TS4 + P1  -> TS4 Imm 1, A    i.e. n3 + 3(n2 + n)
+ 1 -> TS4
A19 TS2 != TS1 -> DISC Imm 18, A  If new n = m, return to A7,
else A6
-----------------------------------
A6  END
```

The temporary TS tanks are used for quick access. TS2 holds n, TS3 holds n2, and TS4 holds n3. The first three instructions simply zero the tanks TS2, TS3, and TS4 (so n, n2, and n3 are all, correctly, zero). The following steps calculate the values iteratively, as described; by step A19, TS2, TS3, and TS4 will all hold their new values. (These are all discarded rather than saved, as this code was for demonstration only.) Imm 1 means an immediate transfer with timing number 1, that is, it goes straight to the next instruction. For more detailed information, check out the full paper. And that, dear readers, is where we'll have to leave him for now. We'll get to Turing's work in Manchester in a forthcoming issue of Linux Voice.