Whatever Happened to the Other Turing Machine?

Brian E. Carpenter
Robert W. Doran
The University of Auckland

ACE2012
June 2012

Whatever Happened to the Other Turing Machine?

Brian E. Carpenter
Robert W. Doran
The University of Auckland

ACE2012
June 2012

How we started

- In 1975, we (BEC and RWD) were lowly academics in the CS Department at Massey University, Palmerston North, New Zealand.
  - There wasn't much to do in Palmerston North at the weekend, and more seriously, research funding for CS was hard to get.
  - We were both interested in computer architecture and its history.
- In 1975, computer history was largely American – the computer was the von Neumann machine and everybody knew about ENIAC and EDVAC.
  - Early British computers were viewed as a footnote.

Inputs

- Turing was known as a theoretician and for his work on AI.
- But the first paper on ACE did say “based on an earlier design by A M Turing” (Wilkinson 1954).
- A trade press article by Rex Malik described Turing as “a four in the morning system kicker” - that didn’t sound like a theoretician.
- There were rumours of secret stuff happening during World War II, and there were Brian Randell’s 1972 paper and 1973 book.
- The Ultra secret was blown in 1974.
- In 1972, NPL issued a reprint of Turing’s 1945/46 ACE report; we got hold of it in 1975.

Processing the input

- Reading, in 1975, what Turing wrote in 1945 was an astounding experience.
- Of course, he'd read the EDVAC report by von Neumann, but the ACE proposal went much further.
  - Enchanting style
  - Startling originality
  - A complete design
  - Very detailed

Output:
Computer Journal, **20**(3) 269-279 (1977)

The other Turing machine

B. E. Carpenter* and R. W. Doran†

Department of Computer Science, Massey University, Palmerston North, New Zealand

In a little known report written in 1945, A. M. Turing made a detailed proposal for the construction of a stored program computer. Although sharing some ideas with von Neumann's draft report of the same year, Turing's proposal contained a wide range of novel and formative concepts. These include subroutines, the stack and a micromachine architecture. This paper analyses his report in general terms and in detail, and describes his ideas in modern terms.

(Received October 1975)

*Present address: CERN, 1211 Geneva 23, Switzerland.
†Present address: Amdahl Corporation, Sunnyvale, California 94086, USA.

Volume 20 Number 3

So, here we are 35 years later

- Reading the report* again today, it's *still* enchanting and startling.
- Turing's 48 typed pages synthesised the concepts of
  - a stored-program universal computer
  - a floating-point subroutine library
  - artificial intelligence
  - details such as a hardware bootstrap loader
  - and much else.

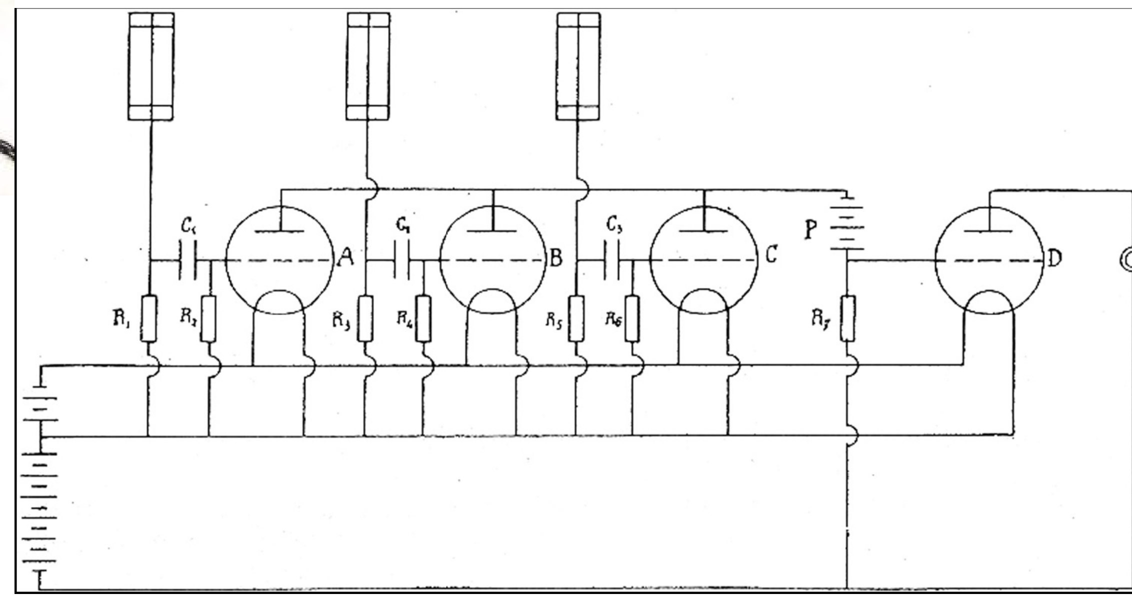
* *Proposals for development in the Mathematics Division of an Automatic Computing Engine (ACE)* (written late 1945, presented to NPL Executive Committee in March 1946)

How was a theoretician able to write such a report in 1945?

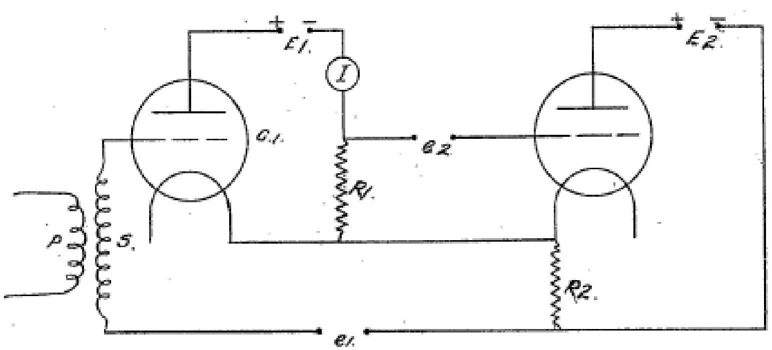
- That’s a trick question: Turing *wasn’t* just a theoretician by 1945.
- His most famous theoretical work was a thought experiment requiring a memory tape and logic circuits.
- He had made two serious attempts to build mathematical machines before the war.
- During the war, he designed information processing machines, witnessed large scale data processing, and built electronics personally.

Was U realistic in 1936?

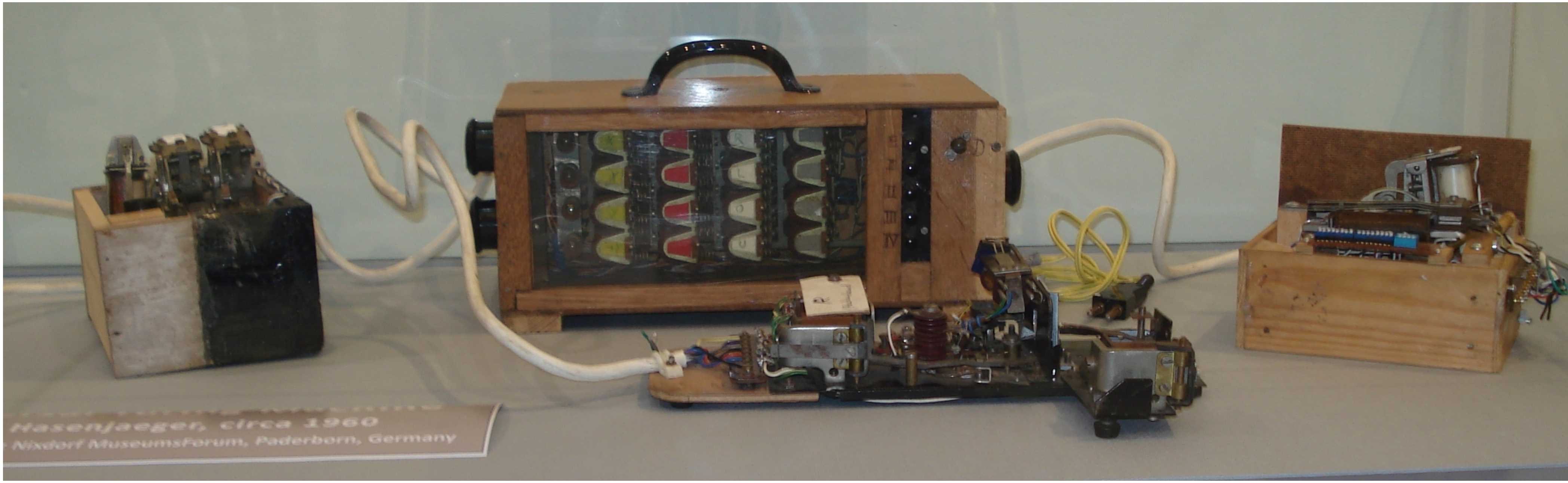
1898 Poulsen “Telegraphone” wire recorder
(image from www.johansoldradios.se)



1930 Rossi coincidence (AND) circuit
(L.Bonolis, Am.J.Phys. **79** 1133-1150, 2011)

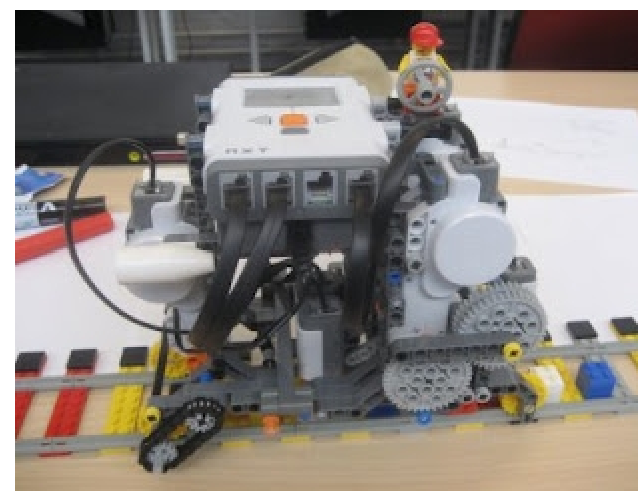


1919 Eccles-Jordan multivibrator trigger (flip-flop) (British patent 148582)



Gisbert Hasenjaeger machine (~1960)

Mike Davey machine (2010)

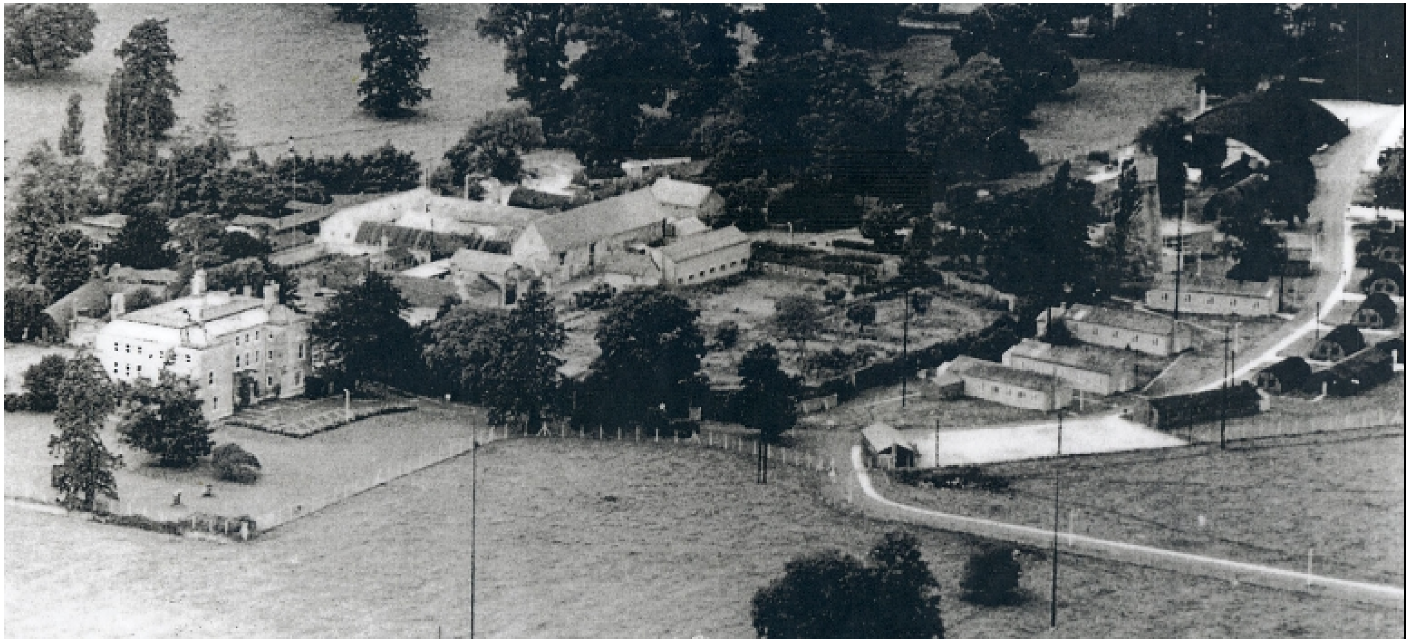


Lego of Doom (2009)

(photo: Wikipedia)

Bletchley Park: Turing designed Bombes and supervised large-scale data processing.

Hanslope Park:Turing designed and built voice encryption electronics



(photo: www.mkheritage.co.uk/hdhs)

Stirrings at the National Physical Laboratory

- The NPL Mathematics Division was approved in late 1944
  - Supported by the Ministry of Supply, Cdr Edward Travis of GCCS, Hartree, and L.J.Comrie, the New Zealander who founded Scientific Computing Service Ltd. in 1938*.
  - The Division's job was to provide and coordinate national facilities for automated computation, including military applications.
- The first head of the Mathematics Division was J.R.Womersley, better known for his later work on fluid dynamics.
  - He'd read *On Computable Numbers* before the war.
  - He visited the U.S.A. in early 1945 and learnt about both ENIAC and plans for EDVAC, just before taking up the job at NPL.

*SCS Ltd. is still registered at Companies House as a private company.

A typical computer of the 1930s

Photo of LJ Comrie in *A Computer Perspective*, 1973, credited to Mrs B Atkinson

Civil servants behaving creatively

- Womersley understood the potential of universal automatic computers and was willing to foster unconventional ideas.
- He showed Turing the EDVAC report and hired him as a one-man section of his Division to study the design of an Automatic Computing Engine.
- The result: by the end of 1945, Turing's *Proposals for Development in the Mathematics Division of an Automatic Computing Engine (ACE)* was finished.
- It was presented to the NPL Executive Committee in March 1946, supported by Womersley and Hartree.
- The ACE project was approved by the committee (chaired by NPL Director Sir Charles Darwin, grandson of *the* Charles Darwin).

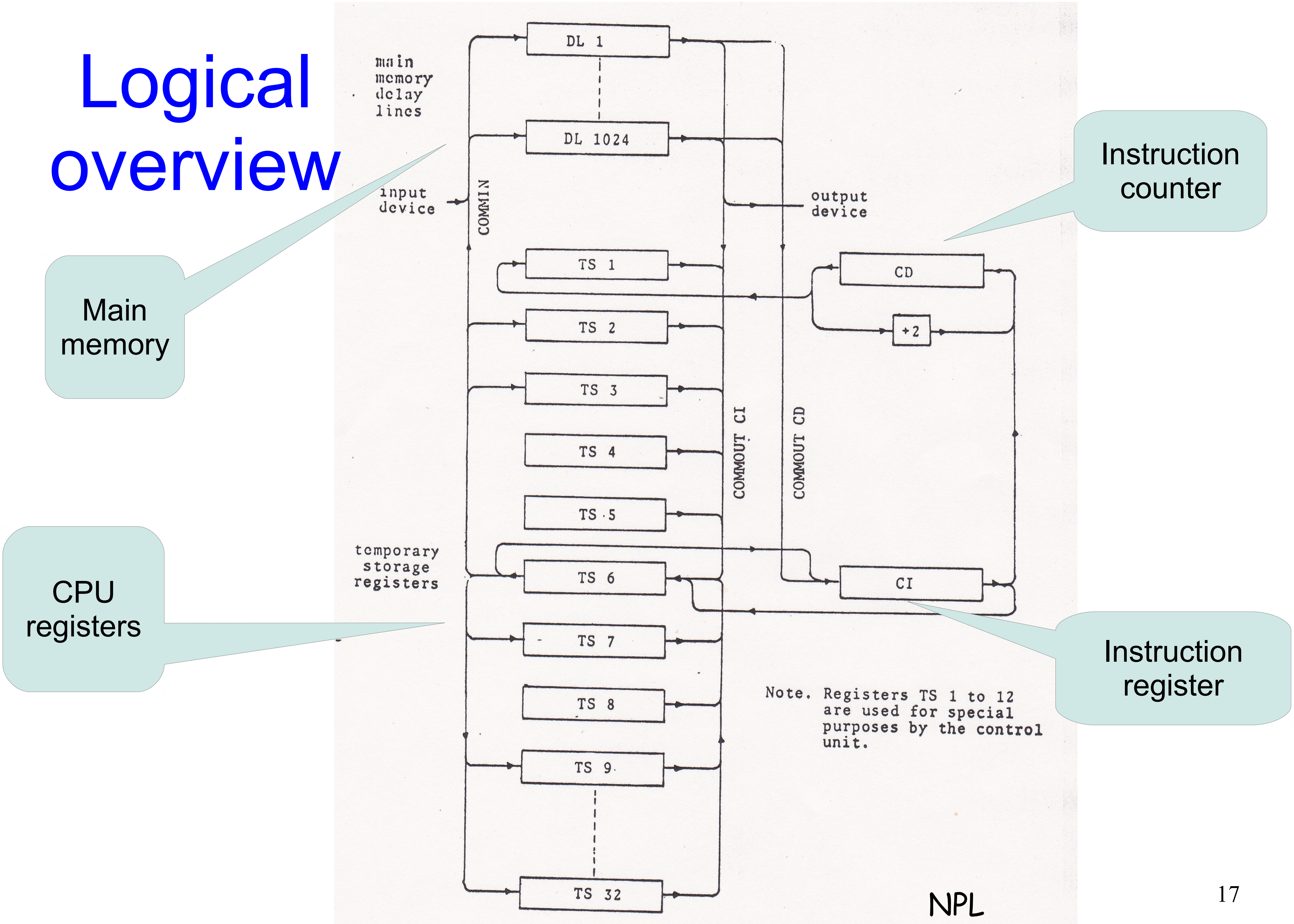
Overview of the ACE report

- An outline of the principles of stored program computers, binary representation, and floating point arithmetic
- A detailed architecture and instruction set
- Detailed logic diagrams, and electronic circuits for logic elements
- Example programs
- A budget estimate of £11,200 (twenty times Turing's annual salary at NPL)

Main features of ACE hardware

- Serial 1 MHz machine with 32 bit words
  - That's about 0.03 MIPS
  - "word" in this usage appears to have been coined by Turing
- 32 registers
  - TS = temporary storage, actually a short mercury delay line
- Register-to-register instruction set
  - EDVAC was an accumulator machine
- Only 11 instructions

Logical overview



Main features of ACE software

- Proposed applications ranged from numerical analysis (expected by NPL) to counting butchers, solving jigsaws, and chess (certainly not expected).
- Foresaw relocatable code and something very like assembly language.
- Foresaw a subroutine library, including a stack for nested subroutine calls.

Turing's basic subroutines

```
INDEXIN     TS28 ← M[TS27]
DISCRIM     TS24 ← if TS8 = 0 then TS16 else TS15
PLUSIND     TS27 ← TS27 + 1
TRANS45     TS20,21 ← TS22,23
BURY        M[TS31] ← TS1 + 1; TS31 ← TS31 + 1;
            go to M[TS1]
UNBURY      go to M[TS31 ← TS31 - 1]
MULTIP      TS22,23 ← TS 18,19 x TS20,21
ADD         TS22,23 ← TS 18,19 + TS20,21
BINDEC      Convert TS22,23 to card image in DL11
```

(As transcribed into 1970's notation by Bob Doran and me.)

Summary of formative ideas (1)

1. Binary implementation using standardised electronic logic elements*

2. Complete notation for combinational* and sequential circuits

3. Memory-Control-Arithmetic Unit-I/O architecture*

4. Stored program*

5. Conditional branch instructions (clumsy)*

6. Address mapping (for instruction interleaving)

7. Instruction address register and instruction register

** Also found in EDVAC report. Have not analysed vs Colossus or ideas that Turing may have known from Babbage, Lovelace and Jacquard.*

Summary of formative ideas (2)

8. Multiple fast registers in CPU, for data and addressing; register-register instructions.

9. Microcode; hierarchical architecture

10. Whole-card I/O operations (almost DMA)

11. Complete set of arithmetic, logical and rotate orders

12. Built in error detection and margin tests

13. Floating point arithmetic

14. Hardware bootstrap loader (initial program load)

Summary of formative ideas (3)

15. Subroutine stack
16. Modular programming; subroutine library
17. Documentation standards
18. Link editor; symbolic addresses; programs treated as data
19. Run time systems (I/O conversions; hints of macro expansion)
20. Nonnumerical applications
21. Artificial intelligence

Turing’s Sources

- Some of the ideas derived from the EDVAC report, i.e. from Atanasoff, Eckert and Mauchly via von Neumann. But most of them appeared to us in 1975 to be completely new.
- The stored program concept derived from *On Computable Numbers.*
  - Zuse invented it independently, but in 1945 nobody knew.
- We now know about Turing’s discussions with Shannon during the war, and about his experience at Bletchley and his knowledge of Colossus.
  - Maybe some of these ideas were not brand new in the ACE report in the way they seemed to us in 1975.
  - It remains quite startling to find them all in one place at such an early date.

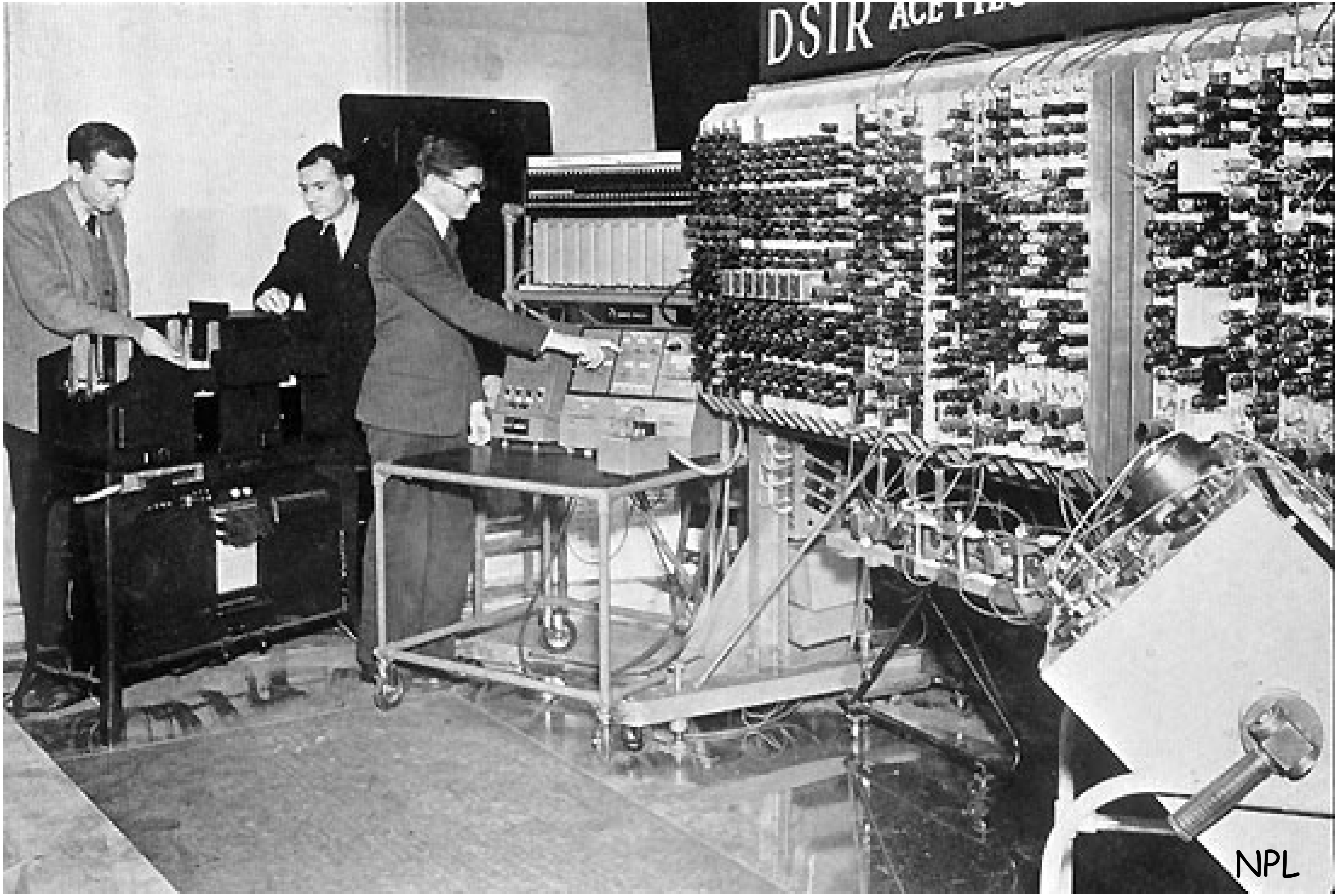
What happened to Turing's design?

- ACE went through many design cycles after the project was approved.
  - The most notorious change was the change from interleaved instructions to optimum coding, where each machine instruction includes the address of its successor.
- Turing the pragmatist was frustrated by slow progress under peacetime civilian conditions.
- He left NPL for Manchester, via a short return to Cambridge.

The ACE family

- Pilot ACE finally ran at NPL in May 1950.
  - The English Electric DEUCE (1955) was directly derived from Pilot ACE.
- The anachronistic full scale ACE (1958)
- The Post Office/RRDE MOSAIC (1953)
- An EMI one-off ‘business machine’ (~1956)
- The Bendix G-15 (1956)
- The Packard-Bell 250 (1960)

The Pilot ACE in 1950

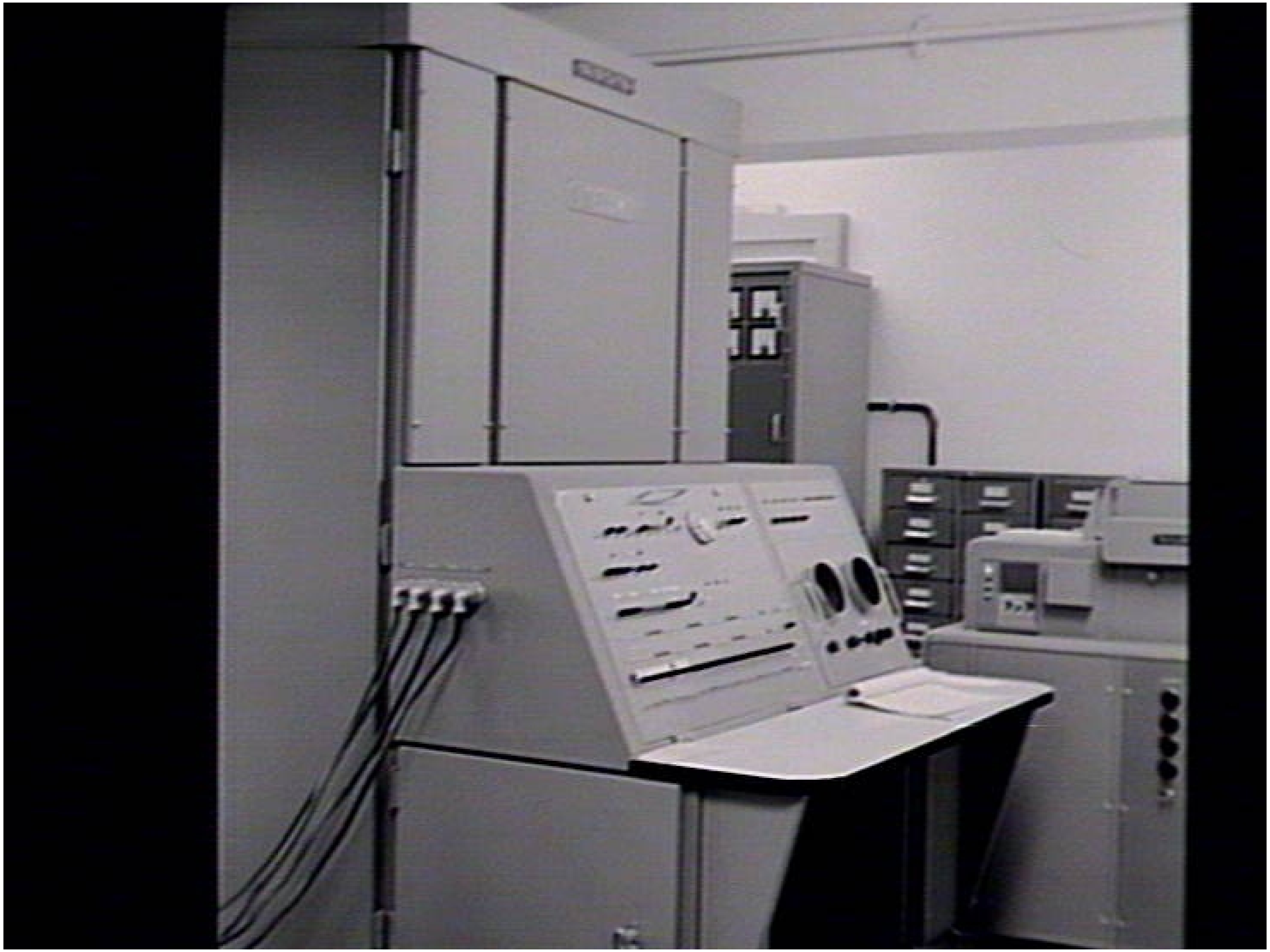


G. G. Allway, E. A. Newman and J. H. Wilkinson.

What the DEUCE?

- Gordon Bell, designer of the PDP-11, used a DEUCE as a Fulbright Scholar in Australia. The PDP-11 was the first widespread register-register machine. Bell wrote: "DEUCE certainly influenced my thinking -- some of it was negative because I was determined not to design a computer that was so difficult and tricky to program."
- We can take that as a comment on optimum coding.
- The Unibus concept in the PDP-11 is very reminiscent of the ACE architecture, with its common input and output lines linking all the central registers and main memory together.
  - However, the Ferranti Pegasus (1956) was also very influential on the PDP-11.

The DEUCE that Gordon Bell used



This DEUCE was at the University of Technology NSW Australia
(© Mitchell Library, State Library of NSW, users.tpg.com.au/eedeuce/)

What happened to Turing's ideas?

- All of them resurfaced; the question is how much of that was re-discovery, and how much was unacknowledged re-use.
- The 1945 ACE report, mimeographed in perhaps 50 or 100 copies, was out of stock by 1948 and vanished from view until 1972.
- The Cambridge team, especially Wilkes, never admitted to much influence by Turing.
  - Stanley Gill, however, worked on Pilot ACE and EDSAC simultaneously.
- In practice, the Manchester team also mainly followed the EDVAC line.

Where did Turing's ideas resurface? (1)

- Address mapping resurfaced in Manchester, most famously in Atlas (1962).
- The instruction counter and instruction register became universal, perhaps because there's really no other way to do it.
- Multiple fast registers in CPU, for data and addressing, became widespread in the 1960s, for example in the IBM 360.
  - The 1949 Manchester B-line, the first index register, was foreshadowed in Turing's INDEXIN routine.
  - Ferranti Pegasus (1956) had multiple registers.

Where did Turing's ideas resurface? (2)

- Microcode; hierarchical architecture
  - presaged in MIT Whirlwind (1947)
  - reappeared most famously in Cambridge (EDSAC2, 1958).
- Whole-card I/O operations (almost DMA).
  - DMA is credited to the NBS DYSEAC (1954)
  - channel I/O is credited to the IBM 709 (1957).

Where did Turing’s ideas resurface? (3)

- Complete set of arithmetic, logical and rotate orders (probably suggested by cryptanalysis requirements).
  - Logical instructions in Manchester Mark I (1949)
  - The IBM 701 had SHIFT and AND instructions (1952).
- Built in error detection and margin tests
  - Turing knew about this need from Bletchley Park experience (or from T. Flowers?).
  - Other builders of thermionic valve computers had to learn it the hard way.

Where did Turing’s ideas resurface? (4)

- Floating point arithmetic (in electronics)
  - Appeared in Manchester MEG, the prototype Ferranti Mercury, and the IBM 704 (1954)
  - Known conceptually since 1914
  - Already seen in electromechanical machines (Zuse, 1938; Harvard Mark II, 1944; Stibitz Model V, 1945)
- Bootstrap loader
  - Appeared in IBM 701 (1952)

Where did Turing's ideas resurface? (5)

- Modular programming; subroutine library.
  - Reinvented by Grace Hopper (1951-2) and by Wilkes, Wheeler and Gill (1951).
- Subroutine stack
  - Recursive calls in LISP (McCarthy,1958), Algol (Dijkstra et al, 1960).
  - Appeared in hardware in English Electric KDF9 (1960), Burroughs B5000 (1961), Atlas (1962).

Where did Turing's ideas resurface? (6)

- Link editor; symbolic addresses; programs treated as data.
  - EDSAC had instruction mnemonics (1949).
  - Turing contributed to the Manchester software effort, but the first real Autocode solution was due to Tony Brooker (1954).
- Documentation standards
  - Usually credited to Grace Hopper around 1952, but Turing recognised the need 3 years before any stored program machine was built.

Where did Turing's ideas resurface? (7)

- Run time systems (I/O conversions etc.).
  - Became universal, and another remarkable Turingesque foresight in 1945.
- Nonnumerical applications
  - Presumably inspired by cryptanalysis, but Turing couldn't mention that in writing.
- Artificial intelligence
  - At least Turing got credit for this one (along with Shannon, with whom he had discussed it during the war).

Turing’s sociological foresight (1)

- At Bletchley Park, Turing saw what was needed to manage a data-intensive operation.
  - “One of our difficulties will be the maintenance of an appropriate discipline, so that we do not lose track of what we are doing. We shall need a number of efficient librarian types to keep us in order... I have already mentioned that ACE will do the work of about 10,000 [human] computers.” (1947)

Turing’s sociological foresight (2)

- He also foresaw what systems programmers would be like, with their mystique and gibberish.

“The masters [programmers] are liable to get replaced because as soon as any technique becomes at all stereotyped it becomes possible to devise a system of instruction tables which will enable the electronic computer to do it for itself. It may happen however that the masters will refuse to do this. They may be unwilling to let their jobs be stolen from them in this way. In that case they would surround the whole of their work with mystery and make excuses, couched in well chosen gibberish, whenever any dangerous suggestions were made.” (1947)

How much credit should Turing get?

- It’s clear that Turing was the first to work seriously on a general-purpose computer design in the UK, in late 1945, and that he showed remarkable foresight.
- The community of pioneers in the UK and the US was relatively small until 1954 when Turing died; we can assume that word of mouth had a significant effect, and that ideas were not always properly credited.
- We know that apart from NPL, there was some direct influence in Manchester but little influence in Cambridge.
- It’s impossible to tell, today, to what extent Turing’s amazing foresights were passed on indirectly to other pioneers, or to what extent they were simply rediscovered when their time came.

NPL

Sources and further reading

- *Alan Turing: The Enigma,* Andrew Hodges, Burnett Books, 1983.
- *The other Turing machine,* B.E. Carpenter & R.W. Doran, Computer J. **20** (1977) 269-279.
- *A.M.Turing's ACE Report of 1946 and other papers*, Brian E. Carpenter and Robert W. Doran (editors), Vol. 10 in the Charles Babbage Institute Reprint Series, MIT Press, 1986.
- *Early British computers: the story of vintage computers and the people who built them*, Simon H. Lavington, Manchester University Press, 1980.
- *Innovating for Failure: Government Policy and the Early British Computer Industry*, John Hendry, MIT Press, 1989.
- *Alan Turing's Automatic Computing Engine*, B. Jack Copeland (ed.), Oxford, 2005.
- *The Essential Turing*, B. Jack Copeland, Oxford, 2004.
- *Alan Turing and his Contemporaries*, Simon Lavington (ed.), BCS, 2012.
- *A Survey and Taxonomy of I/O Systems*, Mark Smotherman, last updated July 2006, www.cs.clemson.edu/~mark/io_hist.html
- *The Manchester Computer: A Revised History (Pts 1 & 2),* B. Jack Copeland, Annals Hist. Comp. **33**(1), (2011) 4-21 & 22-37.
- Wikipedia and Google, of course.