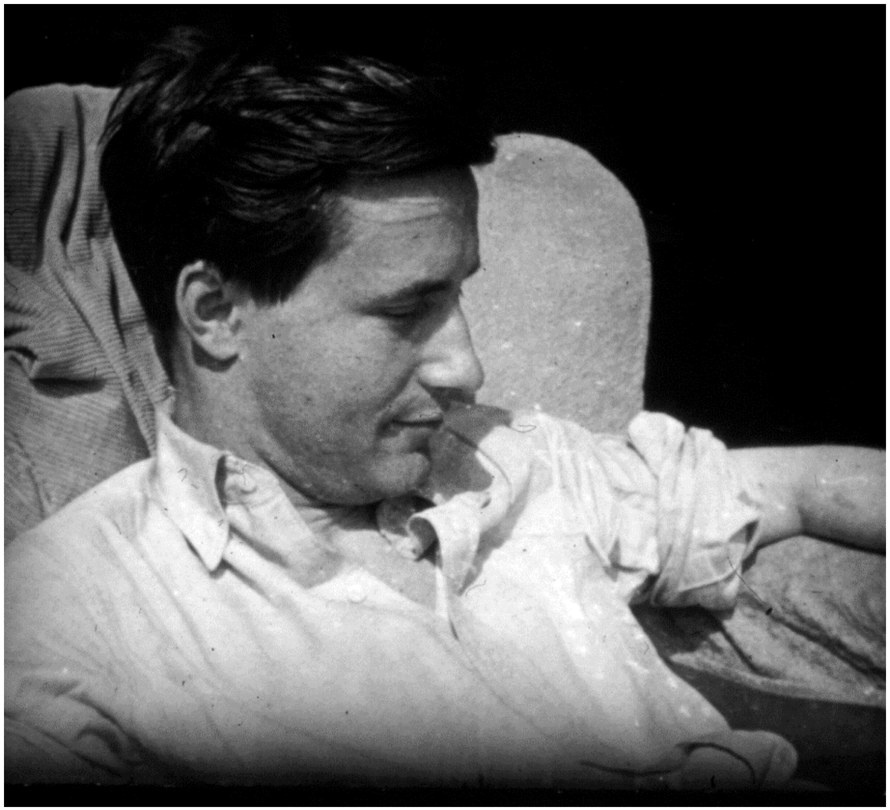
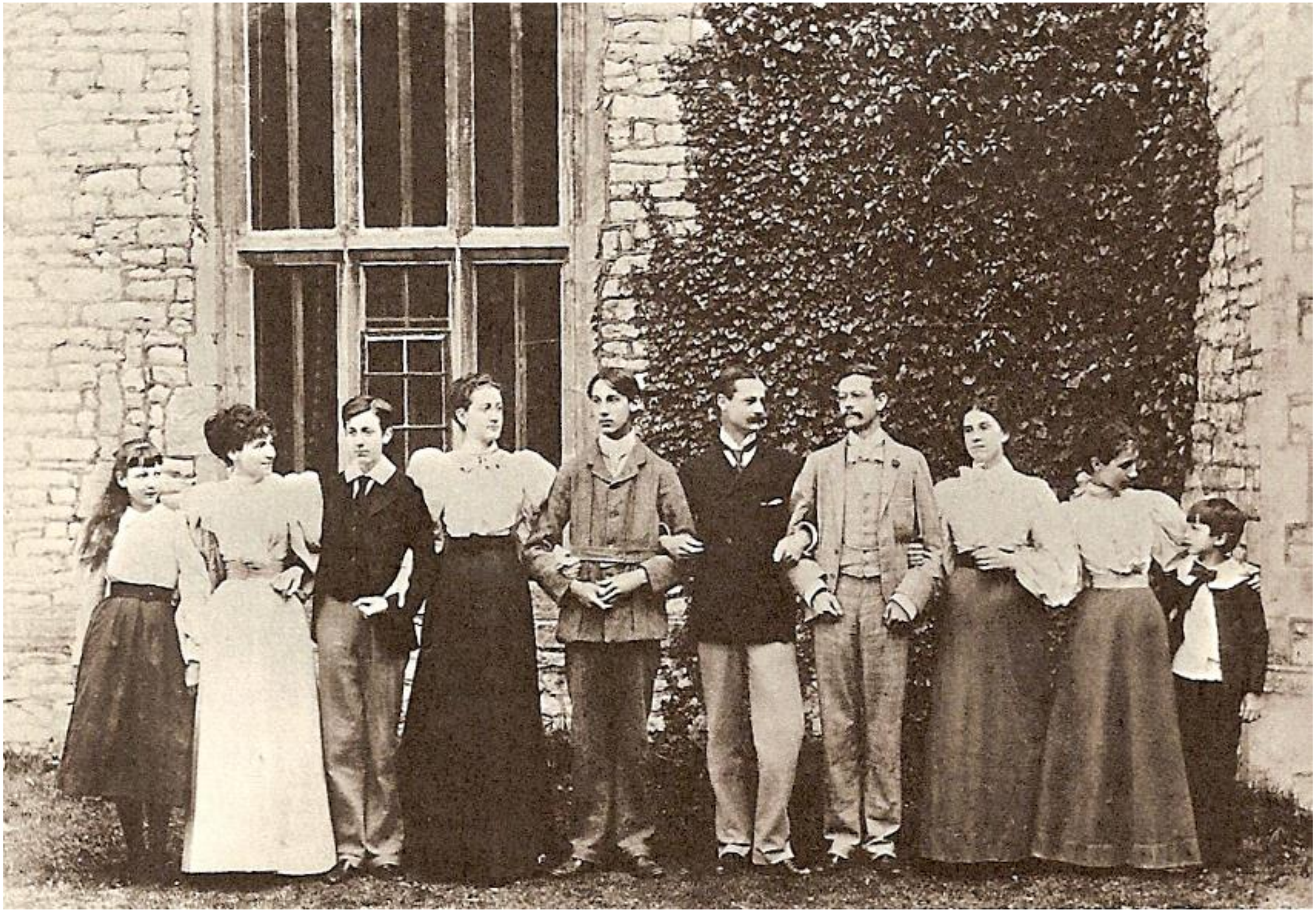
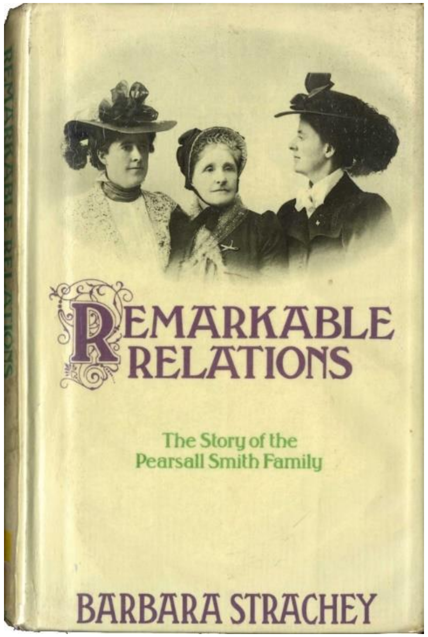
Christopher Strachey 1916 - 1975:
The Bloomsbury Years



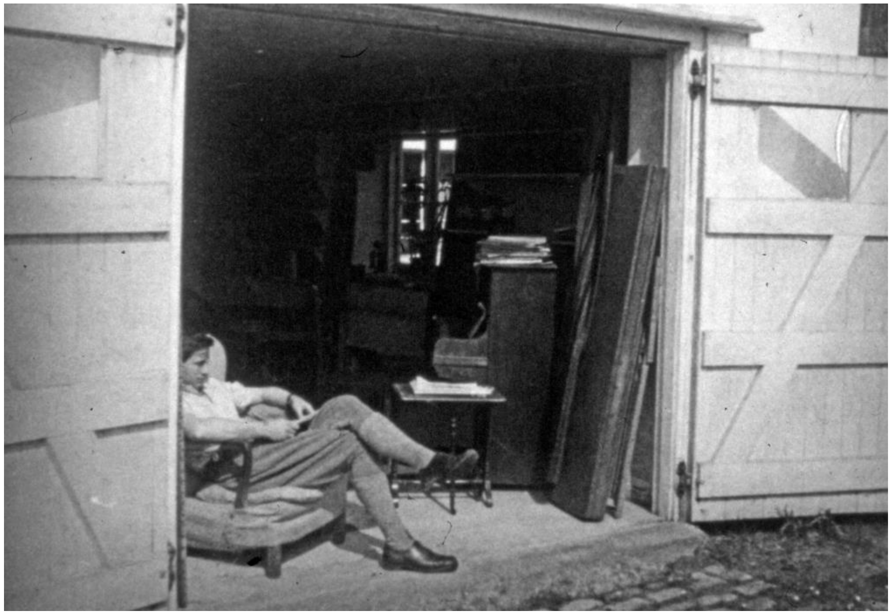


The Strachey Dynasty, c.1893

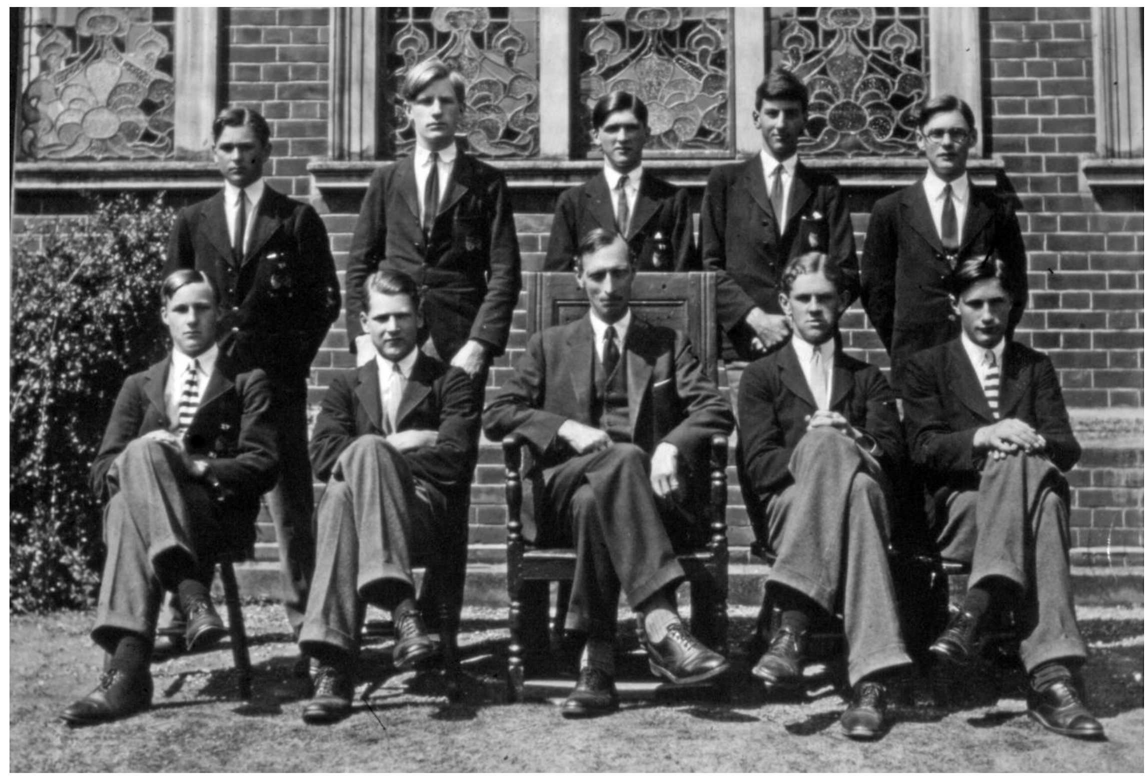


The Pearsall Smith Family, 1894

Parents: Oliver Strachey (1912) and Ray Costelloe (c.1925)



The Mud House, Haselmere, Sussex

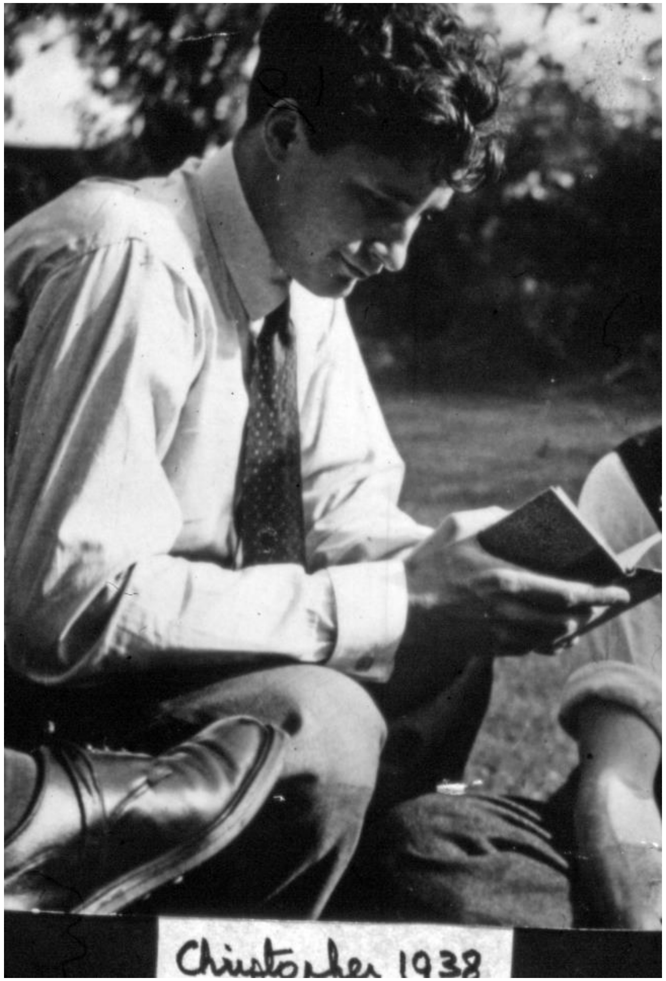


Gresham’s School, Norfolk, 1930-35

Gresham’s School, Norfolk, 1930-35

Kings College, 1936-39

Occupations Before Electronic Computers, 1939-1952



- STC, Illminster, 1939-45
- St Edmunds School, 1946-49
- Harrow School, 1949-52

XIV. *Hahn's Functions* $S_m(\alpha)$ *and* $U_m(\alpha)$.

By C. Strachey and P. J. Wallis *.

[Received June 29, 1945.]

1. *Summary and Introduction.*

In a paper [1] on the calculation of fields in certain resonators, Hahn introduced two new functions :

$$-S_m(\alpha)=\sum_{n=1}^{\infty}\frac{m^2\sin^2 n\pi\alpha}{n(m^2-n^2\alpha^2)}$$

and

$$U_m(\alpha)=\sum_{n=1}^{\infty}\frac{m^2 n\sin^2 n\pi\alpha}{\alpha^2\left(n^2-\frac{m^2}{\alpha^2}\right)^2}\quad\text{with } 0<\alpha<1$$

and used these functions to shorten his calculations. Since this time, Hahn's method has been used for certain similarly-shaped resonators and Hahn's two functions usually help to shorten the solution considerably. Hahn himself only gave a small table of $S_m(\alpha)$ and a few values of $U_m(\alpha)$.

In this report closed expressions are derived for the case of α rational, and are used to produce a much more comprehensive table of $S_m(\alpha)$ and a slightly smaller table of $\frac{1}{m}U_m(\alpha)$. In a concluding section integral expressions, power series in α, and asymptotic series in m are given which together facilitate the calculation for values of α not given in the tables.

2. *Closed Expressions for Rational* α.

For many purposes it is only necessary to know the values of Hahn's functions for rational $\alpha(=p/q$, say), and in this case it is comparatively easy to obtain closed expressions for the functions which include those previously given by Goddard [2] as a special case (when $p=1$). For $\alpha=p/q$

$$-S_m(\alpha)=\sum_{n=1}^{\infty}\frac{m^2\sin^2 n\pi\alpha}{n(m^2-n^2\alpha^2)}$$

$$=\sum_{n=1}^{\infty}\frac{m^2\sin^2\frac{np}{q}\pi}{n\left(m^2-\frac{n^2p^2}{q^2}\right)}$$

$$=\sum_{k=1}^{q-1}\frac{\sin^2\frac{kp}{q}\pi}{2q}\sum_{r=0}^{\infty}\left(\frac{2}{r+\frac{k}{q}}-\frac{1}{r+\frac{k}{q}+\frac{m}{p}}-\frac{1}{r+\frac{k}{q}-\frac{m}{p}}\right)$$

by writing $n=rq+k$,

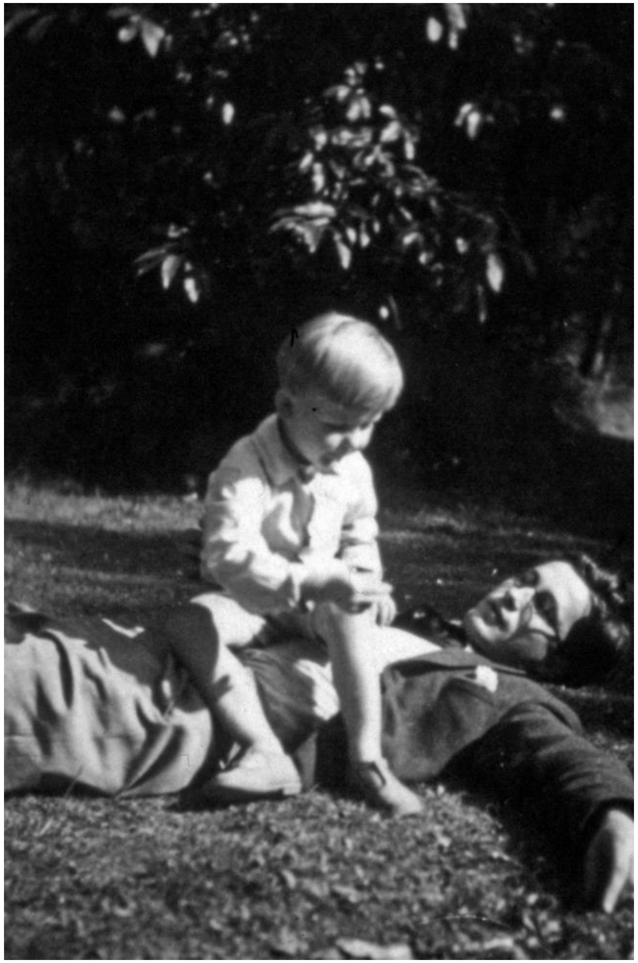
$$=\frac{1}{2q}\sum_{k=1}^{q-1}\left[\psi\left(\frac{k}{q}+\frac{m}{p}\right)+\psi\left(\frac{k}{q}-\frac{m}{p}\right)-2\psi\left(\frac{k}{q}\right)\right]\sin^2 k\pi\alpha, \quad . \quad (2.1)$$

* Communicated by the Authors.

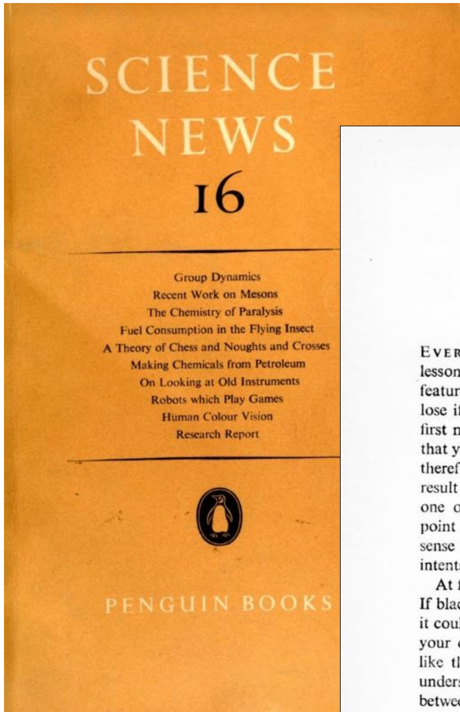
$-S_m(\alpha)$.

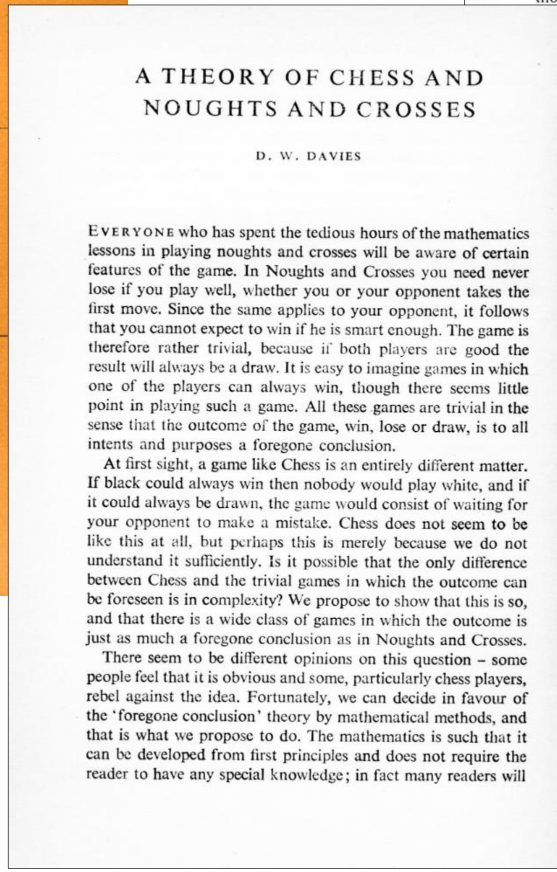
α \ m	1	2	3	4	5	6	7	8	9	10
0	1·21883	1·55718	1·75824	1·90148	2·01277	2·10378	2·18076	2·24746	2·30631	2·35897
0·1	1·21058	1·54893	1·74998	1·89323	2·00452	2·09553	2·17251	2·23921	2·29806	2·35071
0·2	1·18562	1·52387	1·72491	1·86814	1·97943	2·07044	2·14742	2·21412	2·27297	2·32562
0·3	1·14324	1·48104	1·68198	1·82518	1·93645	2·02745	2·10443	2·17113	2·22998	2·28263
0·4	1·08205	1·41855	1·61923	1·76234	1·87357	1·96454	2·04151	2·10820	2·16704	2·21968
0·5	1·00000	1·33333	1·53333	1·67619	1·78730	1·87821	1·95513	2·02180	2·08062	2·13326
0·6	0·89371	1·22019	1·41855	1·56078	1·67159	1·76234	1·83916	1·90576	1·96454	2·01714
0·7	0·75775	1·06998	1·26438	1·40505	1·51510	1·60538	1·68198	1·74836	1·80705	1·85957
0·8	0·58285	0·86499	1·04939	1·18562	1·29335	1·38231	1·45801	1·52387	1·58212	1·63435
0·9	0·35105	0·56366	0·71779	0·83861	0·93700	1·01956	1·09142	1·15546	1·21058	1·26240
1·0	0	0	0	0	0	0	0	0	0	0
0·25	1·16667	1·50476	1·70577	1·84900	1·96028	2·05128	2·12826	2·19497	2·25381	2·30646
0·75	0·67604	0·97604	1·16667	1·30571	1·41493	1·50476	1·58102	1·64725	1·70577	1·75818
$\frac{1}{3}$	1·12500	1·46250	1·66339	1·80657	1·91784	2·00883	2·08580	2·15250	2·21135	2·26400
$\frac{2}{3}$	0·80685	1·12500	1·32114	1·46250	1·57289	1·66339	1·74007	1·80657	1·86529	1·91784

STC, Illminster, 1938-45




St Edmunds School, 1946-49





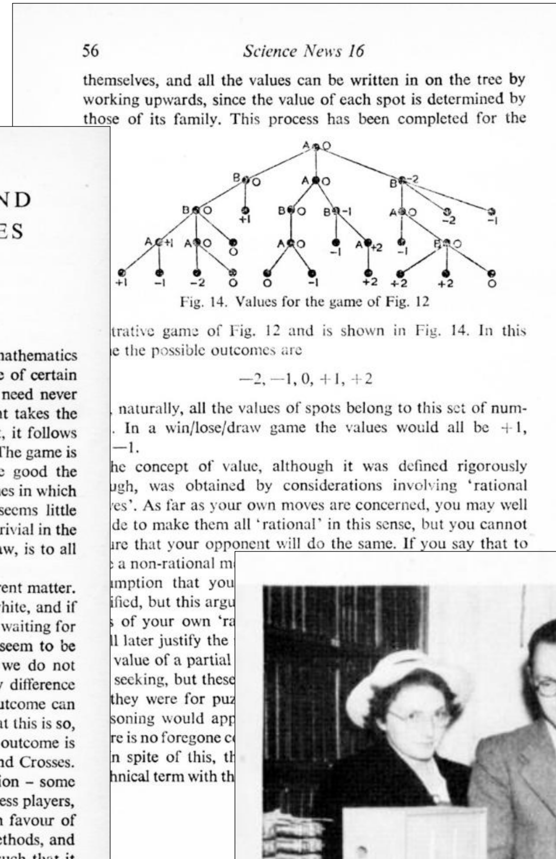
A THEORY OF CHESS AND NOUGHTS AND CROSSES

D. W. DAVIES

EVERYONE who has spent the tedious hours of the mathematics lessons in playing noughts and crosses will be aware of certain features of the game. In Noughts and Crosses you need never lose if you play well, whether you or your opponent takes the first move. Since the same applies to your opponent, it follows that you cannot expect to win if he is smart enough. The game is therefore rather trivial, because if both players are good the result will always be a draw. It is easy to imagine games in which one of the players can always win, though there seems little point in playing such a game. All these games are trivial in the sense that the outcome of the game, win, lose or draw, is to all intents and purposes a foregone conclusion.

At first sight, a game like Chess is an entirely different matter. If black could always win then nobody would play white, and if it could always be drawn, the game would consist of waiting for your opponent to make a mistake. Chess does not seem to be like this at all, but perhaps this is merely because we do not understand it sufficiently. Is it possible that the only difference between Chess and the trivial games in which the outcome can be foreseen is in complexity? We propose to show that this is so, and that there is a wide class of games in which the outcome is just as much a foregone conclusion as in Noughts and Crosses.

There seem to be different opinions on this question – some people feel that it is obvious and some, particularly chess players, rebel against the idea. Fortunately, we can decide in favour of the 'foregone conclusion' theory by mathematical methods, and that is what we propose to do. The mathematics is such that it can be developed from first principles and does not require the reader to have any special knowledge; in fact many readers will



56 *Science News 16*

themselves, and all the values can be written in on the tree by working upwards, since the value of each spot is determined by those of its family. This process has been completed for the

Fig. 14. Values for the game of Fig. 12

[...]trative game of Fig. 12 and is shown in Fig. 14. In this [...]e the possible outcomes are

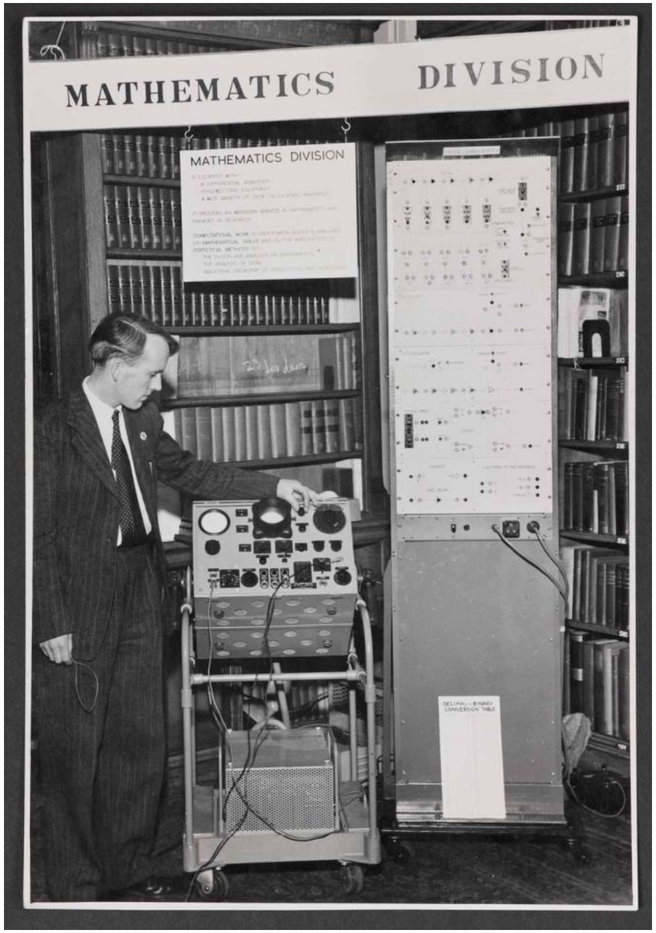
$$-2, -1, 0, +1, +2$$

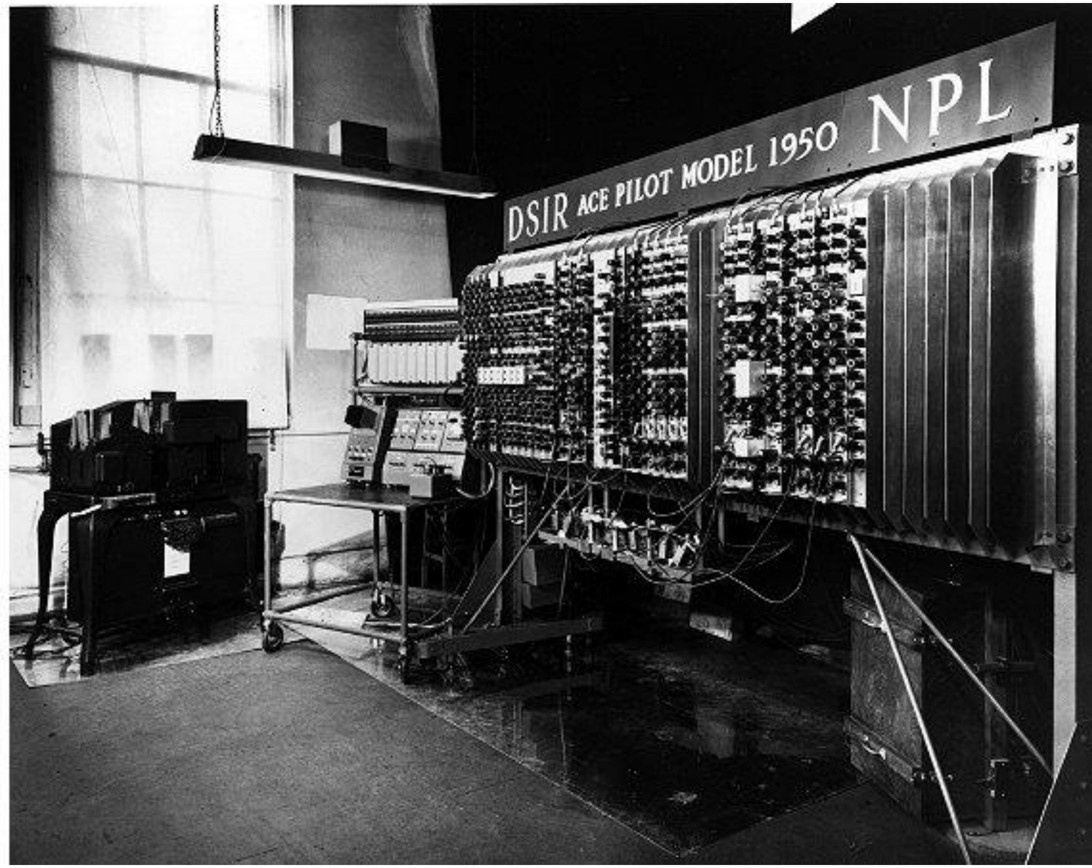
[...], naturally, all the values of spots belong to this set of num[...]. In a win/lose/draw game the values would all be $+1$, [...] -1.

[...]he concept of value, although it was defined rigorously [...]ugh, was obtained by considerations involving 'rational [...]es'. As far as your own moves are concerned, you may well [...]de to make them all 'rational' in this sense, but you cannot [...]re that your opponent will do the same. If you say that to [...] a non-rational m[...] [...]mption that you [...]ified, but this argu[...] [...] of your own 'ra[...] [...]ll later justify the [...] value of a partial [...] seeking, but these [...] they were for puz[...] [...]soning would app[...] [...]re is no foregone c[...] [...]n spite of this, th[...] [...]nical term with th[...]

16 Noughts and Crosses Playing Machine

Harrow School, 1949-52



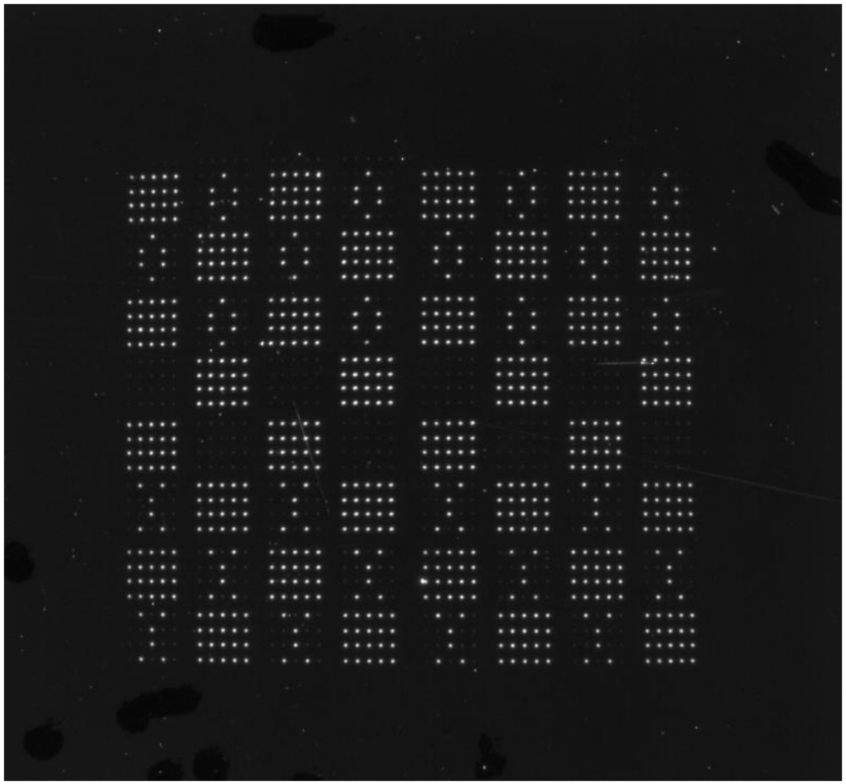


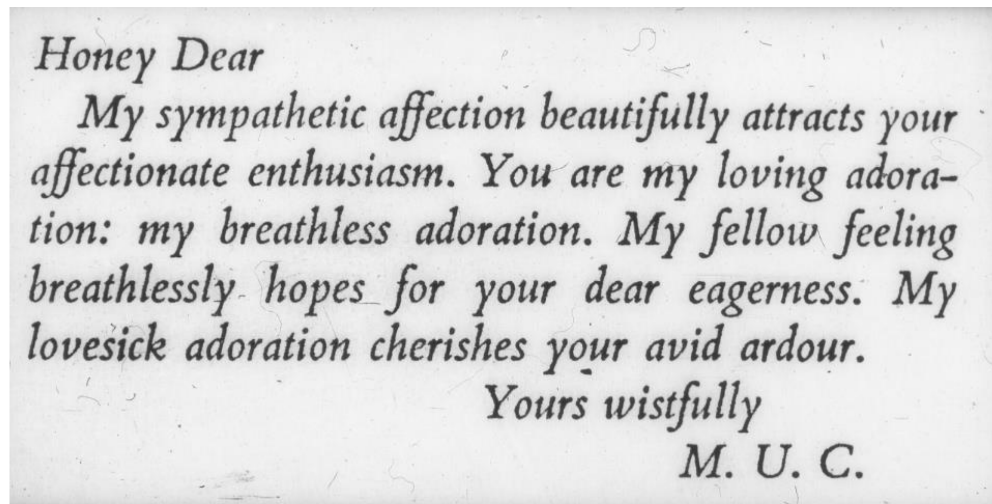
Harrow School, 1949-52

Dawn of Electronic Computers, NRDC 1952-1959

- The Manchester Mark I
- St Lawrence Seaway Calculations
- Computer Design
- Time-Sharing

Lord "Tony" Halsbury, 1908 - 2000





Honey Dear

My sympathetic affection beautifully attracts your affectionate enthusiasm. You are my loving adoration: my breathless adoration. My fellow feeling breathlessly hopes for your dear eagerness. My lovesick adoration cherishes your avid ardour.

Yours wistfully

M. U. C.

The Manchester Mark I, 1952

Backwater calculations for the St. Lawrence Seaway with the first computer in Canada

Scott M. Campbell

Abstract: As the oldest branch of engineering, it is fitting that civil engineering was the first in Canada to make use of modern computing techniques. In the early 1950s, serious planning was underway regarding the St. Lawrence Seaway and Power Project, but before construction could begin, a lengthy series of backwater calculations was required to predict up-river changes to the water profile. It was estimated that these calculations would have taken 20 person-years to complete by hand, but in 1952 and 1953 Ontario Hydro was able to make use of the first electronic computer in Canada – the Ferut at the University of Toronto – to complete in about eight months. These were the first major calculations carried out on any electronic computer in Canada, and helped prove that an all-Canadian navigation route was possible.

Key words: history, computing, backwater, Canada.

Résumé : Étant la plus ancienne branche d'ingénierie, il est normal que le génie civil soit le premier au Canada à utiliser les techniques informatiques modernes de calcul. au début des années 1950, une planification majeure était en cours pour le Projet de Voie maritime et de centrales hydro-électriques du Saint-Laurent mais, avant que la construction ne puisse débuter, une longue série de calculs des retenues a été requise pour prédire les changements en amont de la coupe OU du profil des eaux. Il a été estimé que ces calculs pourraient avoir demandé 20 années-personne à réaliser à la main mais, en 1952 et en 1953, Ontario Hydro a pu utiliser le premier ordinateur électronique au Canada – le Ferut à l'Université de Toronto – pour les compléter en environ huit mois. Ce fut les premiers calculs d'importance effectués sur un ordinateur électronique au Canada, et ils ont aidé à révéler qu'une voie de circulation entièrement canadienne était possible.

Mots-clés : histoire, calcul, retenue, Canada.

[Traduit par la Rédaction]

1. Introduction

There are several book-length histories of the St. Lawrence Seaway, as well as countless articles, reports, and brochures. Most dedicate a significant portion to the social and political history of the waterway prior to its construction, but few explore the engineering challenges in great depth. Several diminish or altogether ignore the associated Power Project and, to my knowledge, none mention the crucial role played by a computer. This article aims to fill several of these gaps and explore how civil engineering — the oldest branch of engineering — came to be the first in Canada to make prominent use of a computer.

The Ferut, the heart of the story, is a first-generation British electronic computer that was installed at the University of Toronto in 1952 (Ferut stood for Ferranti at the University of Toronto). Yet a proper prologue cannot fail to mention the role of the *Manchester Pioneer*, which carried the computer from Manchester to Canada on its maiden voyage across the Atlantic in April 1952, long before construction of the St Lawrence Seaway would begin in 1954. The ship was specially designed to navigate the shallow channels and narrow locks of the St. Lawrence River — with about 30 cm of clearance in some places. It was an inaugural trip marked by fanfare on both sides of the Atlantic, opening the first two-way direct service between Manchester and the Great Lakes. Despite the excitement among the manufacturers and traders on both sides of the ocean, the contents of the *Manchester Pioneer* for her first trip were unremarkable, with one exception. Among the general cargo were 15 boxes containing what one anonymous reporter for the *Globe and Mail* called "an electronic brain" destined for the University of Toronto (Anonymous 1952). It would be the first computer in Canada, and although very few Canadians knew about Ferut, its first major task would be to carry out computations necessary for the completion of the St. Lawrence Seaway and Power Project.

These computations would be an important trial for the new technology: could this extremely expensive computer be put to immediate and practical use? There were only about twenty comparable computers in the world at this point and most were dedicated to military projects, atomic energy simulations, or other similar top secret research. In contrast, the first nontrivial computer program ever written in Canada would carry out a complex set of calculations related to a challenging civil engineering problem. Additionally, the backwater results would eventually play a minor political role when the initially reluctant United States eventually agreed to participate in the St. Lawrence Seaway and Power Project. Unfortunately, for many years the assistance of the computer was unmentioned and this story has been hidden from common knowledge.

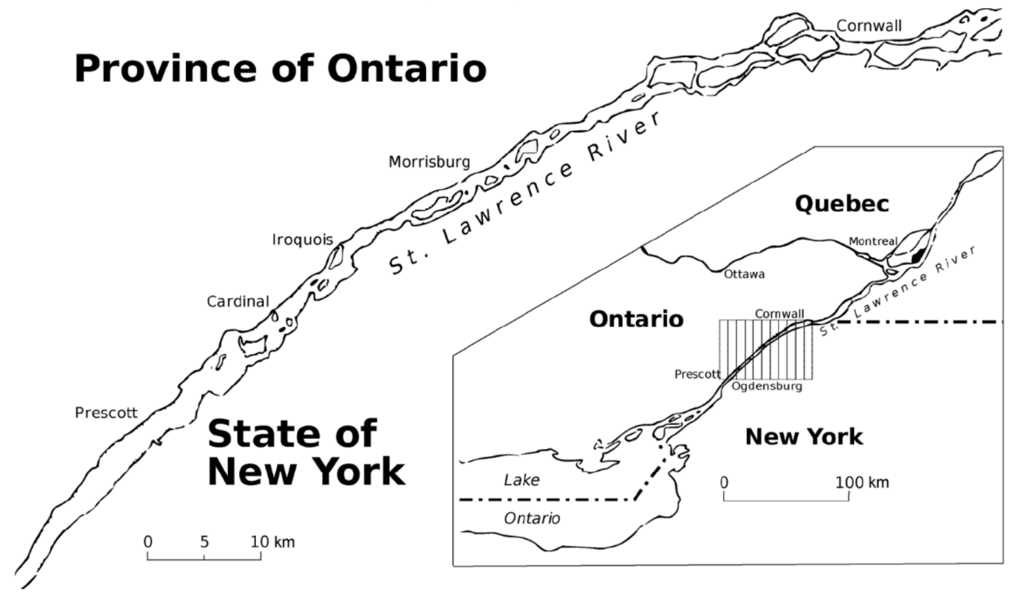
Received 27 August 2008. Revision accepted 25 March 2009. Published on the NRC Research Press Web site at cjce.nrc.ca on 22 July 2009.

S.M. Campbell. Centre for Society, Technology and Values, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sm2campb@uwaterloo.ca).

Written discussion of this article is welcomed and will be received by the Editor until 30 November 2009.

 doi:10.1139/L09-060 Published by NRC Research Press

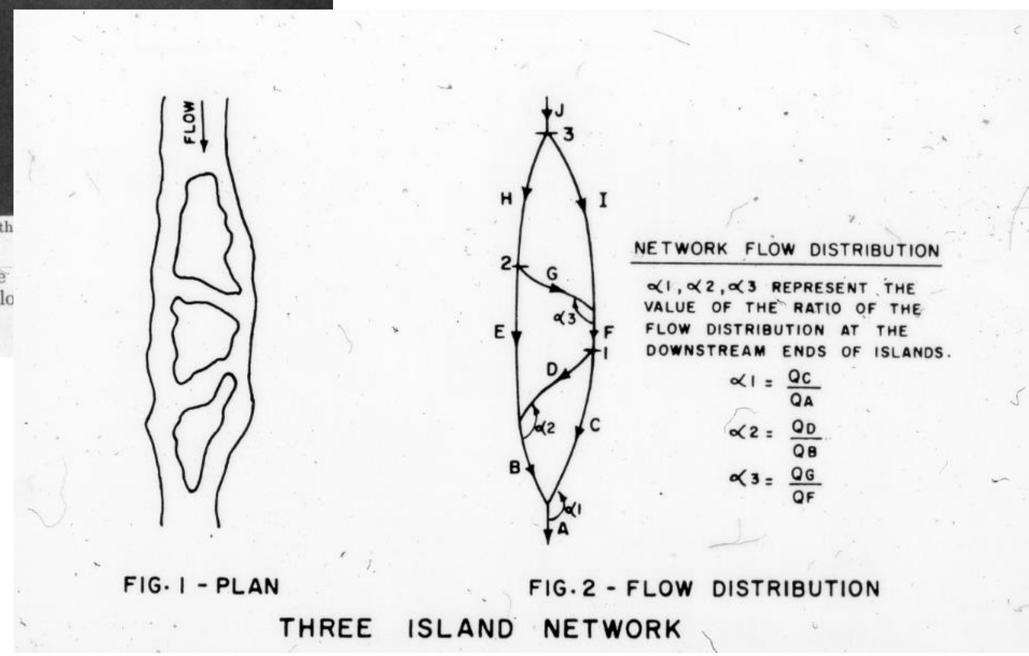
Fig. 1. The backwater calculations necessary to construct the St. Lawrence Seaway and Power Project were limited to a section of the St. Lawrence River between Prescott and Cornwall, Ontario. (Wood 1955).



St Lawrence Seaway Calculations

Power·Auth

MOSES-SAUNDERS ("BIG MO") POWER DAM AS THE WATER CAME UP BEHIND IT
Construction was almost completed on this dam, the largest single structure of the
omposed of steel cells was in the process of being removed. Half of the power dam belo
nd half to the Province of Ontario (right).



St Lawrence Seaway Calculations

Computer Design: Ferranti Pegasus v. English Electric DEUCE

United States Patent Office

3,222,647
Patented Dec. 7, 1965

1

3,222,647

DATA PROCESSING EQUIPMENT

Christopher Strachey, London, England, assignor, by mesne assignments, to International Business Machines Corporation, New York, N.Y., a corporation of New York

Filed Feb. 4, 1960, Ser. No. 6,752
Claims priority, application Great Britain, Feb. 16, 1959, 5,263/59
1 Claim. (Cl. 340—172.5)

This invention relates to electrical data processing equipment and to methods of operation therefor, and more particularly, but not exclusively, to digital computers.

The speed of operation of computers continues to increase and computer designs now contemplated, and by no means impracticable, are of the order of a thousand times faster than the early electrical computers of not much more than ten years ago.

Even with the early computers, directly the idea arose of using them for repetitive, relatively simple programs, the problem arose of introducing and extracting information to be processed at sufficient speed to keep the computer efficiently occupied.

The prime sources of input and the ultimate recorders of output were relatively very slow in operation, and buffer equipment was introduced in which a gear change took place: information was received at one speed and sent out at increased speed to the computer. By using several sources of this type the computer could be kept reasonably occupied.

The speeds of prime sources of information have been much increased, but there has also been a remarkable increase in computer speed, so that the problem of efficient user of computer time continues, and is the more urgent since although the potential increase in capacity is relatively much larger than the corresponding increase in cost, its value depends on the ability to use it effectively.

There are activities in which a computer is, at present, very inefficiently used. One of these is program checking. For many purposes the best method of program checking is for a skilled programmer to sit at the operating console of the machine and to plan his operations according to the results produced by the machine. Unfortunately this method is so grossly wasteful of machine time, even with relatively slow machines, that it is generally not allowed except for a few very special problems. The concept of time sharing between operators makes it possible once more to allow this manual program checking at a special console, without seriously interfering with the amount of machine time available for ordinary computing.

Another activity which makes very inefficient use of a computer is the maintenance and adjustment of the peripheral equipment such as paper tape readers and magnetic tape units. Some of these need a considerable amount of adjustment which can only be done satisfactorily by using the computer. If this part of the maintenance is carried out on a time sharing basis, it should be possible to reduce the total machine time used for maintenance quite considerably.

Several new problems appear as soon as it is contemplated to have several variable programs in the machine at the same time. The most important of these is the necessity of ensuring that the programs do not interfere with each other. This is particularly important, of course, if one of the programs concerned is still under development and so is unpredictable. The solution to this difficulty is to provide for interlocks on the main store so that each program is restricted to altering (and perhaps also to reading) numbers in its own section of

2

the store. This in its turn introduces the problem of altering the interlocks. It is evident that it must be possible to change them when required, or it would be impossible to use the whole machine on a single large problem, and for reasons of speed it is obviously desirable to have them altered by a machine instruction. The problem is to ensure that even if a program obeys a completely unpredictable series of orders, it still shall not be able to alter the interlocks and spoil another program.

The other rather difficult problems are concerned with the best method of program checking on a machine of this sort. The majority of programs (and programmers) are not suitable for the manual checking methods. It is therefore necessary to make some provision for other methods of program checking, and it is likely that there will be a considerable amount of this work. A particular problem which arises in this context is the difficulty of determining when a program under test is in error and has come to a loop stop. If this is not detected rapidly, it can waste a disproportionate amount of computer time.

Even now the state has ben reached, when the attempt to utilize a computer efficiently for the various types of purpose which they are required to serve has led to such complexity of input equipment and output equipment from a number of sources that the computer itself becomes overshadowed in size and cost by its ancillary equipment.

Further, the haphazard and uncoordinated use of computers from a number of different input stations for purposes some of which can be extremely wasteful of computer time poses a problem which becomes more and more acute as the speed of operation, and cost, of computers increase. It is current practice to arrange the computer logic so that a preferred station can break in on a current computation, and cause the current instruction and the state of affairs within the logic in carrying out the instruction to be stored, so freeing the logic for handling a preferential instruction from the preferred station.

However, such a proposal is of very limited application because the need to extend such a thought to cater for the various types of program and the conflicting needs of a number of different stations would complicate and increase the cost of the logic to an undesirable extent.

An object of the present invention is to provide flexible and economic facilities within the computer, which can handle the above problem however complex it becomes.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of the preferred embodiment of the invention, as illustrated in the accompanying drawings.

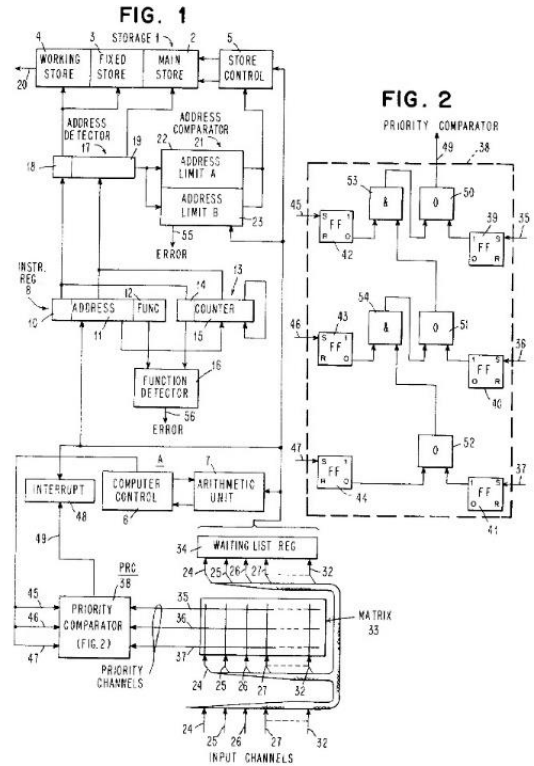
It is proposed to achieve these objects by means of a master program designed to cater automatically for the conflicting demands of a number of stations of different types within a predetermined basic plan incorporated in a coordinating set of instructions which will be called the Director. By means of this program, together with a small fixed amount of additional equipment, all the conflicting interests and diverse requirements can be automatically interwoven into an automatically-coordinated whole which utilizes the computer time on an efficient and economic basis. It will be appreciated that by having a long-running base load program of lowest priority which would come into operation in gaps between programs of higher priority and short duration, the computer can be kept fully occupied: such low priority programs can be routine test programs or program checking from a console.

Each item sent by a station will be treated, so far as transmission is concerned, as a separate message and will be preceded by an "interrupt program signal, which

Dec. 7, 1965 C. STRACHEY 3,222,647

DATA PROCESSING EQUIPMENT

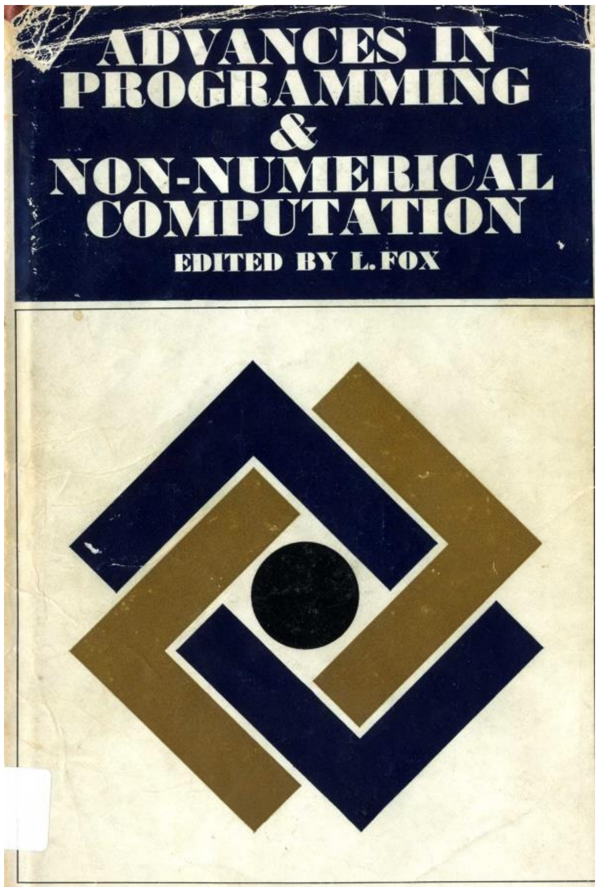
Filed Feb. 4, 1960



Time-Sharing, 1959

Consulting and Cambridge Maths Lab, 1959-1965

- More of the same – but better paid
- Thinking seriously about programming



- CPL, Cambridge 1962-66
- “Advances in Programming”, 1963
- “Towards a Formal Semantics,” 1964
- Formation PRG, Oxford, c.1965
- “Mathematical Theory of Programming,” MIT, 1966

Thinking seriously about programming

Strachey’s last photograph