



Il “programma memorizzato”

Genesi, significato e diffusione di un’idea

Silvio Hénin

La “architettura di von Neumann”, anche detta ‘a programma memorizzato’, la struttura logica di quasi tutti gli attuali computer, viene fatta risalire ad un breve rapporto dattiloscritto del 1945, considerato il documento fondante che diede avvio allo sviluppo dell’informatica moderna. Senza dubbio lo scritto costituì un forte impulso che spinse numerosi ricercatori in tutto il mondo ad intraprendere la costruzione dei primi computer, ma gli eventi di quei primi anni eroici, soprattutto la genesi delle idee fondamentali, la loro evoluzione e la loro diffusione furono più complessi e sfumati di quanto comunemente si affermi in molte storie del calcolo automatico.

Keywords: von Neumann’s architecture, Stored program, John von Neumann, Alan M. Turing, EDVAC, ACE, Early computers

1. Introduzione

Quasi tutta la storiografia sul calcolo automatico suddivide la cronologia degli eventi che portarono all’avvento del computer in due grandi ‘ere’, una *protostoria* che va dall’invenzione delle prime macchine calcolatrici meccaniche del XVII secolo fino alla II Guerra Mondiale, e una *storia moderna* che inizia nel 1945. Qualunque suddivisione della storia in ere è, come si sa, un artificio convenzionale, certamente utile a fini didattici e divulgativi, ma spesso un’approssimazione che difficilmente regge ad



un'attenta analisi. Il confine tra due ere – spesso chiamato 'rivoluzione', anche quando i contemporanei neppure se ne accorsero – viene fatto coincidere con un evento puntiforme e ben identificabile; così la transizione tra Medioevo e Rinascimento è stabilita in concomitanza con la scoperta del Nuovo Continente e la 'rivoluzione' scientifica con la pubblicazione del 'Dialogo' di Galileo o dei 'Principia' di Newton.

Nella storia del calcolo automatico l'evento cardine che viene spesso proposto è la pubblicazione di un breve documento intitolato "First Draft of a Report on the EDVAC" (Prima bozza di un rapporto su EDVAC,¹ da qui in poi semplicemente *First Draft*) datato 30 giugno 1945,² ma probabilmente scritto tra marzo e aprile dello stesso anno, che riporta il nome di John von Neumann come unico autore. Per il vero, non si trattò di una formale pubblicazione a stampa, ma della distribuzione di poche copie ciclostilate³ del dattiloscritto ad un ristretto numero di destinatari, ventiquattro soltanto stando a quanto afferma Herman Goldstine che era all'epoca uno dei principali attori del progetto di EDVAC. L'importanza del documento è enfatizzata dallo stesso Goldstine che sostiene: "Costituì un importante contributo che produsse un profondo impatto non solo su EDVAC, ma servì da modello a tutti gli studi del progetto logico [di un computer]."[1] Altrove si afferma che il rapporto "Esprime qualcosa che Babbage, se non anche Turing, avevano trascurato" e che "Descrisse le proprietà logiche di un vero calcolatore elettronico digitale per uso generale." [2], o anche che "Il computer a 'programma memorizzato' è stato fondamentale per l'informatica quanto il motore a combustione interna lo è stato per i trasporti".[4]. Negli anni 1950 s'iniziò ad identificare la struttura dei computer con quanto espresso nella *First Draft* e ad attribuire questo rapporto unicamente a von Neumann, da qui l'eponimo 'architettura di von Neumann' che restò indelebilmente connesso alla struttura dei moderni computer.

Per quanto limitata, la distribuzione della *First Draft* provocò un'immediata ondata di interesse nell'intero, ancora esiguo, gruppo di addetti ai lavori ed agì da potente stimolo per i primi progetti di calcolatori elettronici programmabili in tutto il mondo. Allo stesso tempo la distribuzione del rapporto diede origine ad una polemica che dopo quasi settant'anni non è ancora del tutto sopita, quella riguardante la paternità dei concetti che vi sono espressi. Da un lato la diatriba contribuì alla dissoluzione del gruppo di lavoro che aveva costruito ENIAC⁴ e che stava progettando il suo successore (EDVAC) all'Università della Pennsylvania, annullando il

¹ EDVAC è acronimo di *Electronic Discrete VAriable Computer* (Calcolatore Elettronico a Variabili Discrete), stando al primo documento pubblicato che ne parla [3]. Nelle varie fonti l'acronimo è anche sciolto come *Electronic Discrete Variable Automatic Computer* o *Electronic Discrete Variable Arithmetic Computer*.

² Anche se la data indicata nel frontespizio è il 30 giugno 1945, alcune copie furono distribuite già il 25 giugno. [1]

³ All'epoca non esistevano ancora le fotocopiatrici e l'unica tecnologia di riproduzione per piccole tirature erano la fotografia e il ciclostile. Quest'ultimo utilizzava una matrice impressa con una comune macchina da scrivere, con cui si producevano al massimo 100-200 copie.

⁴ ENIAC (*Electronic Numerical Integrator And Computer*) fu il primo calcolatore elettronico interamente basato sulla tecnologia della valvole termoioniche (vedi oltre).



primato che l'ateneo aveva conseguito nella tecnologia del calcolo elettronico. Dall'altro, la 'pubblicazione' della *First Draft* originò un effetto ritardato quando divenne uno degli elementi che concorsero all'invalidazione del brevetto di ENIAC nel celebre processo 'Honeywell contro Sperry-Rand'⁵. [5, 6] L'evento del 1945 mostra quindi tutti gli attributi per essere un momento topico nella storia dell'informatica, ma, per gli stessi motivi, anche per meritare un'attenta analisi del suo significato e della sua influenza.

2. Il calcolo automatico nel 1945

Alla fine della II Guerra Mondiale gli strumenti per il calcolo in uso in tutto il mondo erano più numerosi di quanto si potrebbe oggi supporre. Il cavallo da tiro di tutti i centri di calcolo scientifico e statistico, utilizzato da eserciti di 'computer' umani,⁶ era la comune macchina calcolatrice meccanica da tavolo, prodotta in diversi modelli da aziende negli USA e in Germania, come Frieden, Marchant, Monroe e Brunswiga. Al calcolo scientifico erano anche dedicate macchine meccanografiche a schede perforate di Hollerith (IBM) – opportunamente adattate – soprattutto in America ed in Inghilterra. Le esperienze più note sono quelle del *Nautical Almanac* di Greenwich (UK), sotto la guida di John Todd, del *Thomas J. Watson Laboratory* dell'Università di Columbia (USA), diretto da Wallace Eckert,⁷ per la produzione di tavole astronomiche, del *Los Alamos National Laboratory* per i calcoli necessari alla progettazione della prima bomba atomica e del *Ballistic Research Laboratory* di Aberdeen (USA) per le tavole di tiro dell'artiglieria, ma ne esistevano numerose altre. Vi erano poi i calcolatori analogici usati per la soluzione di equazioni differenziali, come i grandi *Differential Analyser* ideati da Vannevar Bush al Massachusetts Institute of Technology (MIT) e presenti in numerosi centri di calcolo negli USA ed in Europa. Altri strumenti analogici specializzati erano incorporati nelle centrali di tiro delle navi da guerra e nei sistemi di guida automatica di navi ed aerei, quasi tutti coperti dal segreto militare. Un caso avveniristico era il calcolatore analogico elettronico che guidava i missili balistici V2 di von Braun.[7] Infine, esistevano i quattro modelli di calcolatori elettromeccanici a relè progettati da George Stibitz e costruiti presso i Bell Laboratories.[8] Tutti questi dispositivi erano solo parzialmente automatici, richiedendo frequenti interventi umani, permettevano solo una limitata programmazione, oppure erano dedicati specificamente ad una particolare funzione, quindi difficilmente o per nulla adattabili alla soluzione di problemi diversi.

⁵ Il brevetto di ENIAC era stato assegnato a John Mauchly ed Presper Eckert e da questi ceduto alla Sperry-Rand, per la quale costituiva un importante vantaggio competitivo. Nel 1973 il giudice Learson attribuì invece la priorità dell'invenzione di un 'calcolatore elettronico' a Vincent Atanasoff che nel 1937-42 aveva costruito una piccola macchina digitale a valvole, lo Atanasoff-Berry Computer (ABC). Lo ABC non era però programmabile, non era completamente automatico e permetteva solo la soluzione di sistemi di equazioni lineari; difficilmente poteva essere definito un 'computer' nel 1973.

⁶ Il termine 'computer' indicava originariamente un uomo o una donna che per professione eseguivano i calcoli numerici, con carta e penna o con l'aiuto di una calcolatrice per la preparazione di tavole matematiche, astronomiche o attuariali.

⁷ Nessuna relazione con J. Presper Eckert del progetto ENIAC.



Adottando la moderna definizione di *computer* come ‘macchina per il calcolo digitale, automatica, programmabile e per uso generale’, nessuno dei dispositivi elencati può meritare tale nome.

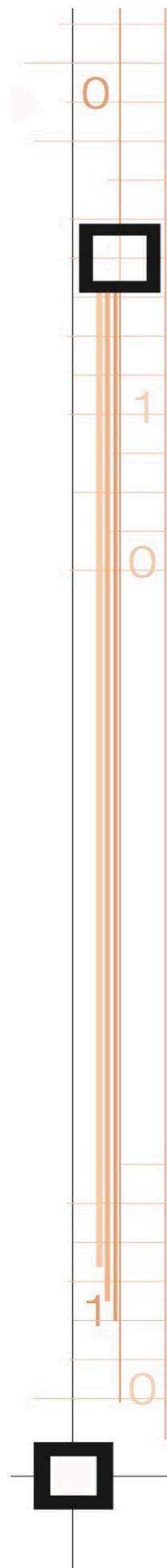
Nell'estate del 1945 esistevano nel mondo solo tre machine che potremmo definire ‘computer’ nell’accezione odierna del termine, una sola delle quali era operativa. Questa era lo Harvard Mark I – noto anche come ASCC (*Automatic Sequence Controlled Calculator*) – ideato da Howard Aiken e costruito dalla IBM, entrato in funzione all’Università di Harvard nel 1944 e che da un anno eseguiva con regolarità calcoli ingegneristici per la Marina degli Stati Uniti e per i laboratori di fisica nucleare di Los Alamos, oltre a produrre tavole matematiche. Secondo Aiken lo ASCC era il vecchio sogno di Charles Babbage⁸ divenuto realtà.[9] Lo ASCC era un'imponente macchina elettromeccanica a relè con notazione decimale il cui programma era registrato su nastri perforati. Un altro calcolatore programmabile si trovava in Germania, battezzato Z4, progettato e costruito dall'ingegner Konrad Zuse. Di dimensioni molto più piccole dello ASCC, lo Z4 era l'evoluzione dei tre modelli sperimentali (Z1, Z2, Z3) costruiti dal 1936 al 1942, poi smontati o distrutti nei bombardamenti di Berlino. La macchina di Zuse era programmabile con nastro perforato, utilizzava la notazione binaria, ma era ancora elettromeccanica,⁹ quindi piuttosto lenta.[10] Nel 1945 lo Z4 si trovava nascosto in un fienile nelle Alpi bavaresi e, oltre a Zuse, ai suoi famigliari ed alcuni collaboratori, nessuno ne conosceva l'esistenza, neppure in Germania. Esistevano infine due progetti in corso di realizzazione: Ad Harvard si stava costruendo il Mark II, [9] successore dello ASCC, mentre ai Bell Laboratories era in produzione il ‘Modello 5’ di George Stibitz, [11] entrambi elettromeccanici e programmabili con nastri perforati. In tutte queste macchine l'accesso alle istruzioni su nastro – che potremmo definire una memoria di sola lettura (ROM) ad accesso sequenziale – rendeva ardua l'implementazione di cicli di elaborazione (*loops*), di sequenze di operazioni utilizzabili in punti diversi del programma (*subroutine*) e di salti a punti diversi del programma (*branching*), incondizionati o condizionati dai risultati precedenti, tutte funzioni essenziali per risolvere problemi matematici di grande complessità in tempi ragionevoli.

Il terzo computer, quasi terminato e che entrerà in funzione verso la fine del '45, era lo ENIAC (*Electronic Numerical Integrator And Computer*) costruito su progetto del fisico John Mauchly e dell'ingegnere elettronico J. Presper Eckert presso la Moore School of Electrical Engineering dell'Università di Filadelfia. ENIAC era la prima macchina completamente elettronica,¹⁰ tutte le sue funzioni realizzate con valvole termoioniche, salvo i dispositivi d'ingresso ed uscita, soluzione che garantiva una velocità di calcolo circa mille volte maggiore dei dispositivi di Stibitz, Aiken e Zuse. Per contro, la

⁸ A metà del XIX secolo Charles Babbage aveva ideato una ‘Macchina Analitica’ (*Analytical Engine*) interamente meccanica e programmabile con schede perforate, che avrebbe potuto risolvere molti problemi matematici. La macchina non fu mai costruita.

⁹ Solo le unità di calcolo e di controllo dello Z4 erano a relè, la memoria era interamente meccanica.

¹⁰ Con l'eccezione dello ABC di Atanasoff (vedi Nota 5).





macchina usava la notazione decimale e il programma di calcolo era impostato spostando cavi di collegamento e azionando interruttori,¹¹ un processo molto complesso che richiedeva giorni. Le scelte fatte per il progetto di ENIAC sono attribuibili alla necessità di sfruttare al massimo la velocità di calcolo resa possibile dall'uso delle valvole (un'addizione di due numeri di 10 cifre in 0,2 millisecondi) che sarebbe stata inutile se le istruzioni del programma fossero state fornite da nastri o schede (leggere una scheda perforata richiedeva 6-7 secondi).[12] Il disegno di ENIAC era il risultato dell'urgenza di disporre di una macchina veloce per calcolare le tavole di tiro per l'artiglieria; la filosofia di progetto adottata era stata quindi quella della 'forza bruta', con soluzioni tecniche poco eleganti, scarsa attenzione all'ottimizzazione del disegno, e ancor meno a dimensioni e costi.¹² Per produrre ENIAC in tempi brevi il progetto era stato congelato molto presto, prima di poter esaminare soluzioni più semplici e razionali.

Questi computer possono essere definiti 'per uso generale', almeno entro certi limiti; più arduo è considerarli veramente 'automatici'. Uno dei loro principali difetti era infatti l'esigua memoria disponibile, solo 72 accumulatori per numeri di 23 cifre decimali (5500 bit) nel caso dello ASCC e soli 20 accumulatori per 10 cifre decimali (665 bit) nello ENIAC. Ciò rendeva impossibile conservare tutti i risultati intermedi nel caso di calcoli molto complessi ed era quindi necessario registrarli temporaneamente su schede perforate per poi reintrodurli nella macchina e continuare l'elaborazione, operazione lenta ed eseguibile solo da personale che doveva essere continuamente disponibile.

All'elenco dovrebbe essere aggiunto un calcolatore elettronico a valvole che svolgeva un prezioso compito nel Regno Unito, *Colossus*. Questo elaboratore, ideato da Max Newman¹³ e costruito da Tommy Flowers nel 1943-44 era dedicato alla decrittazione dei messaggi tedeschi codificati con la macchina Lorenz SZ40/42 presso il centro di criptoanalisi di Bletchley Park. Colossus era però solo limitatamente programmabile e non per 'uso generale'. [13] Le informazioni su di esso – ne furono costruiti undici esemplari – furono coperte dal massimo grado di segretezza e cominciarono a filtrare solo trent'anni dopo, ma questo non impedì che l'esperienza nell'elaborazione digitale elettronica acquisita da coloro che vi lavorarono fosse trasmessa ai progetti dei primi computer inglesi, permettendo a questi di anticipare gli equivalenti americani. [14]

¹¹ Nel 1948 la programmazione di ENIAC fu trasferita ad una 'memoria interna' più facilmente modificabile, ma la velocità della macchina diminuì di sei volte [H. Neukon, "The Second Life of ENIAC", *IEEE Ann. Hist. Comp.*, 28 (2), 2006, pp. 4-16].

¹² ENIAC costò 487.000 dollari dell'epoca, pari a 30 milioni di Euro odierni. ENIAC fu terminato troppo tardi per contribuire allo sforzo bellico, ma fu inizialmente utilizzato per eseguire i calcoli necessari ai laboratori di Los Alamos per la prima bomba termonucleare a fusione, o bomba H.

¹³ Il matematico Max Newman era stato docente di Alan Turing a Cambridge. Dopo la guerra fu il responsabile della costruzione del primo computer britannico, il Manchester Mark I (vedi oltre).

3. Il calcolo automatico nel 1945

Più di un anno prima che ENIAC fosse terminato, il gruppo di lavoro della *Moore School*, costituito da Mauchly, Eckert e dai matematici Arthur Burks e Herman H. Goldstine,¹⁴ era ben conscio dei limiti della macchina, della scarsa memoria, della lenta e difficile programmazione, delle sue esagerate dimensioni e della scarsa affidabilità: l'intervallo medio tra due guasti ad una qualunque delle 17.000 valvole era solo sei ore e mezza. Il team iniziò quindi a pensare ad un successore di ENIAC, battezzato EDVAC, che superasse questi difetti. Tra le criticità da affrontare le prime in assoluto erano le dimensioni della memoria, la velocità di accesso ai dati e alle istruzioni di programma, e la facilità di programmazione. Mentre la costruzione di ENIAC continuava, ormai affidata ai tecnici, il piccolo gruppo iniziò a tracciare il disegno della nuova macchina. Ci rimangono scarsi documenti che testimoniano i primi sviluppi delle idee, tra questi è noto il rapporto 'Disclosure of Magnetic Calculating Machine', [15] dove Eckert propose l'uso di dischi o tamburi magnetici come forma di memoria ad alta capacità e ad accesso veloce sia per i dati sia per le istruzioni di programma.¹⁵ L'evoluzione del progetto di EDVAC subì però una svolta metodologica quando al gruppo si unì un consulente esterno di grande prestigio: John von Neumann.

Von Neumann (1903-1957) fu un matematico di prima grandezza, uno dei maggiori del 20° secolo. Di origine ungherese e laureato in matematica a Budapest, lavorò con David Hilbert a Gottinga ed ottenne un incarico d'insegnamento all'Università di Berlino. Nel 1930 von Neumann fu chiamato al prestigioso *Institute for Advanced Study (IAS)*¹⁶ di Princeton (USA) dove rimarrà fino alla sua prematura scomparsa. I contributi del matematico esprimono la vastità dei suoi interessi, dalla teoria degli insiemi alla meccanica quantistica, alla teoria dei giochi, a quella degli automi, per citarne solo alcuni. Allo scoppiare della guerra von Neumann fu chiamato come consulente di matematica applicata in numerose iniziative



John von Neumann
(1903 – 1957)

militari, tra cui il Progetto Manhattan per la prima bomba nucleare, e fu in questo periodo che iniziò il suo interesse per i metodi di calcolo automatico: resosi conto che i problemi di fisica, soprattutto di fluidodinamica dei fenomeni esplosivi, richiedevano lunghissimi ed estenuanti calcoli, impensabili per i 'computer' umani, iniziò a cercare soluzioni meccaniche. [16, 17]

¹⁴ Nel 1945 Goldstine era tenente dell'esercito e ufficiale di collegamento tra la Moore School e il Ballistic Research Laboratory di Aberdeen a cui era destinato ENIAC.

¹⁵ La nota di Eckert è datata 29 gennaio 1944. In essa non si specifica se dati e programmi dovessero condividere la stessa memoria.

¹⁶ Lo IAS era un centro dedicato agli aspetti più alti e più astratti della ricerca scientifica. Durante la II Guerra Mondiale, allo IAS risiedevano stabilmente scienziati di primo livello come Kurt Gödel e Albert Einstein.



Tra il 1943 e il '44, su consiglio di Warren Weaver, matematico dello *Office of Scientific Research and Development (OSRD)*¹⁷, von Neumann visitò i principali centri di calcolo scientifico alla ricerca di una tecnologia sufficientemente veloce e potente. Si recò quindi alla Harvard University (Aiken), ai Bell Laboratories (Stibitz), allo Watson Laboratory (W. Eckert) e intraprese anche un viaggio in Inghilterra per esaminare le macchine usate da John Todd al *Nautical Almanac*.¹⁸ Fu durante questo viaggio che von Neumann sviluppò un "interesse morboso per il calcolo", come lui stesso scrisse ad un collega.[18] Le soluzioni esaminate non soddisfacevano il matematico; nessuna di esse era sufficiente per affrontare i sistemi di equazioni differenziali a derivate parziali di cui i fisici di Los Alamos avevano necessità. Agli inizi dell'estate 1945 John von Neumann incontrò casualmente Goldstine¹⁹ che gli parlò di ENIAC. Il matematico ne fu subito interessato e propose di collaborare al progetto EDVAC come consulente esterno. Nei mesi successivi von Neumann iniziò a frequentare le riunioni alla *Moore School*, partecipando attivamente alle discussioni con tutto il peso della sua competenza logica e matematica. Uno dei maggiori pregi del matematico ungherese era quello di saper identificare rapidamente i punti essenziali di un problema, anche se per lui totalmente nuovo, e di riorganizzare l'intero discorso ridefinendone fondamenti e metodologia. Durante i suoi frequenti viaggi in treno, che lo portavano da Princeton ad Aberdeen, a Los Alamos, a Boston e a Washington, von Neumann scrisse alcuni appunti sulle discussioni con il gruppo di EDVAC, riformulandoli secondo le proprie concezioni, e li inviò a Goldstine. Quest'ultimo fece dattilografare il manoscritto²⁰ e, forse affrettatamente, lo fece ciclostilare in più copie, col titolo *First Draft of a Report on the EDVAC* e il solo nome di von Neumann nel frontespizio. [1, 17] Come dice il suo titolo, la *First Draft* non è altro che la trascrizione dattilografata di una 'prima bozza', non destinata alla pubblicazione e a cui non farà mai seguito una versione a stampa completa e corretta. Il documento è lacunoso, mancano le intere sezioni sui dispositivi di ingresso ed uscita, ed è costellato di non pochi errori di trascrizione, soprattutto nei simboli usati e nei rimandi interni tra le sezioni. In tempi recenti il documento è stato pubblicato più volte, ma non sempre in versione integrale e spesso senza una revisione. Solo nel 1993 Michael Godfrey ne ha curato un'edizione opportunamente corretta. [19, 20] Ultimamente sono state riscoperte due copie di una versione differente, forse posteriore a quella diffusa da Goldstine, che comprende due paragrafi

¹⁷ Ente per il coordinamento della ricerca scientifica e tecnologica a fini militari, creato dal presidente Franklin D. Roosevelt nel 1941.

¹⁸ Weaver non aveva menzionato ENIAC, forse perché non ancora terminato o forse perché non credeva che Eckert e Mauchly sarebbero riusciti a portare a termine un progetto così complesso, ambizioso ed innovativo.

¹⁹ Del casuale incontro alla stazione ferroviaria di Aberdeen narrano tutte le storie dell'informatica, ma la vasta rete di contatti di von Neumann gli avrebbe comunque permesso di sapere di ENIAC, prima o poi.

²⁰ Formalmente il rapporto avrebbe dovuto essere classificato come 'confidenziale' essendo correlato al contratto tra la Moore School e lo U.S. Army Ordnance Department per la costruzione di EDVAC del giugno 1945, ancora in tempo di guerra. Il manoscritto originale di von Neumann sembra sia andato perduto



precedentemente sconosciuti, ma che poco aggiungono a quanto già si conosceva. [21]

La *First Draft* è organizzata in 15 capitoli, i primi due dedicati alla descrizione generale di un “sistema di calcolo automatico ad altissima velocità” che “esegue istruzioni per effettuare calcoli di elevato ordine di complessità”²¹ e che “una volta ricevute tali istruzioni, [è] capace di eseguirle senza alcun bisogno di ulteriore intervento umano”. Prosegue poi con la definizione degli elementi fondamentali che dovrebbero costituire una tale macchina: 1) Un’unità aritmetica (*Central Arithmetic, CA*) in grado di eseguire almeno le quattro operazioni elementari, ma possibilmente anche altre, come radici e logaritmi; 2) Un’unità di controllo (*Central Control, CC*) che decodifica le istruzioni del programma e ne comanda l’esecuzione, specificando che, se il calcolatore deve essere utilizzabile per ‘uso generale’, si deve distinguere tra CC e le istruzioni stesse (oggi diremmo tra *hardware* e *software*). 3) Una ‘considerevole’ memoria (*Memory, M*) a giustificazione della quale la *First Draft* elenca una serie di problemi matematici che la rendono indispensabile; 4) Una memoria esterna (*Outside Recording Medium, R*), connessa alla memoria interna (*M*) tramite dispositivi di ingresso (*I*) ed uscita (*O*). *R*, che oggi definiremmo come un ibrido tra ‘memoria di massa’ e ‘interfaccia utente’, non è necessariamente ad elevata velocità, ma può essere costituita da nastri e schede perforati o da nastri magnetici. Non sono previste interfacce dirette tra la macchina e l’operatore umano, ma questi può interagire con il computer solo tramite *R*, registrandovi le istruzioni e dati di ingresso e stampando i dati in uscita.²²

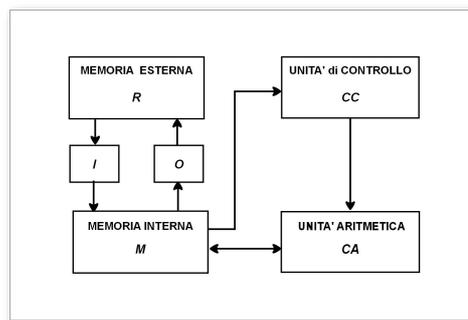
Il documento prosegue con un’analisi dettagliata di come questi quattro elementi devono essere organizzati e connessi tra loro. Qui la *First Draft* si distacca drasticamente da altri progetti precedenti: invece di descrivere i circuiti elettronici, von Neumann adotta una simbologia che astrae dai componenti fisici e dalle loro caratteristiche, concentrandosi sulle funzioni logiche, anche se negli schemi si riferisce prevalentemente all’uso delle linee di ritardo (Riquadro 2). Il matematico prende a prestito la simbologia proposta per la descrizione del cervello biologico e delle sue funzioni. von Neumann aveva iniziato ad interessarsi a questi studi già nel 1941, leggendo un articolo del connazionale Rudolf Ortway [22]; il suo interesse fu poi rinnovato dai lavori del neurofisiologo Warren McCulloch e del matematico Walter Pitts [23] con i quali ebbe frequenti rapporti negli anni successivi.²³ Nella *First Draft* Von Neumann utilizza così la metafora della

²¹ Nella *First Draft* si usano molti termini caduti in disuso nei pochi anni successivi. Ad esempio *order* (ordine) per istruzione, *code* (codice) per programma, *coding* (codifica), per programmazione, *short cycle* (ciclo breve) per parola, *unit* (unità) per bit. Nelle citazioni tradotte useremo sempre la terminologia moderna.

²² L’interattività in tempo reale tra operatore e computer era a quei tempi impensabile e nessuno ne vedeva la necessità.

²³ Assieme a McCulloch e Pitts e a Norbert Wiener, padre della cibernetica, von Neumann fondò la *Teleological Society*, un sodalizio di breve vita che getterà le basi per gli studi sulla teoria degli automi e sull’intelligenza artificiale.

cellula nervosa, il neurone, idealizzato come elemento logico a due stati,²⁴ che chiama *E-element*. Con gli *E-element*, oggi diremmo ‘porte logiche’, costruisce tutte le funzioni dell’unità aritmetica (CA) e dell’unità di controllo (CC), di cui fornisce un’approfondita trattazione. E’ curioso che la *First Draft* non faccia alcun riferimento al lavoro sulla logica dei circuiti a relè scritto nel 1938 da Claude Shannon [26] – il padre della scienza dell’informazione – di cui certamente von Neumann era a conoscenza, [17] eppure l’articolo di Shannon aveva analizzato le relazioni tra logica simbolica di Boole e i circuiti di commutazione a due stati, che diventeranno i componenti dei computer. Nella *First Draft* si stabiliscono altri principi generali, quali la necessità che la notazione dei dati numerici e delle istruzioni sia binaria e non decimale, che tutte le parti siano quanto più semplici possibili per migliorarne la affidabilità, infine l’imperativo che la macchina operi in modo seriale sia nei confronti delle istruzioni, cioè che “non esegua mai due operazioni contemporaneamente” (a differenza di quanto avveniva in ENIAC), sia nell’ambito di ogni singola operazione, cioè operi su “un bit alla volta”, particolare quest’ultimo che sarà adottato solo in pochi dei primissimi computer. Le ultime due sezioni, descrivono il *codice degli ordini*, ciò che oggi chiameremmo ‘set di istruzioni’, la loro codifica e la loro funzione. L’organizzazione del computer in quattro unità funzionalmente distinte non è una novità, ma la si può ritrovare nella Macchina Analitica di Charles Babbage [24, 25] e nelle macchine di Zuse [26]. Non si deve però dimenticare che l’opera di Babbage era conosciuta solo superficialmente e quella di Zuse totalmente ignota.



Schema dell'architettura di von Neumann. Da: *First Draft of a Report on the EDVAC* [19]

4. La memoria, il “programma memorizzato” e il “programma auto-modificabile”

La parte più innovativa della *First Draft* [19] riguarda la memoria (M), la sua realizzazione fisica, le sue funzioni e le sue dimensioni. L’assunto è infatti che la velocità dell’elaborazione elettronica non debba essere vanificata da una memoria troppo lenta o troppo piccola e che le istruzioni di programma devono essere accessibili alla stessa velocità dei dati. E’ quindi implicito nel contesto – anche se non dichiarato – che la memoria debba essere riscrivibile, per poter aggiornare continuamente i dati man mano che il

²⁴ L’ipotesi di McCulloch e Pitts era riduzionista: I neuroni non sono elementi binari e le loro funzioni non sono assimilabili ad operazioni logiche di tipo booleano. Di questo limite von Neumann era ben conscio, come sosterrà negli anni successivi, ad esempio in “*The computer and the Brain*” pubblicato postumo (Yale University Press, 1986). [16, 17]. La metafora del ‘cervello’ è evidente anche nell’uso di parole come ‘organ’ e ‘memory’, che von Neumann adotta a differenza dei predecessori, da Babbage in poi, che usavano ‘store’ (magazzino).



calcolo procede; è altrettanto implicito che le locazioni di memoria debbano essere individualmente indirizzabili, nel moderno senso di memoria 'ad accesso casuale'. Von Neumann valuta in circa 250.000 bit le dimensioni necessarie per i problemi matematici da lui proposti ed esamina le tecnologie al tempo disponibili (Riquadro 2), lasciando ancora aperta la scelta. Il concetto cardine della *First Draft* è che la stessa memoria possa essere utilizzata sia per i dati sia per le istruzioni. Un primo accenno alla duplice funzione appare nell'introduzione, dove si afferma che "Le istruzioni che governano un problema complesso possono costituire un materiale considerevole [...] Questo materiale deve essere ricordato". Più oltre lo scritto continua: "Le varie parti di quest'organo debbono svolgere funzioni alquanto diverse per natura e per scopo [per dati ed istruzioni]", ma "è nondimeno allettante (*tempting*) trattare l'intera memoria come un solo organo e far sì che le sue parti siano intercambiabili quanto più possibile". E' in queste stringate frasi che si esprime il concetto di 'programma memorizzato' (*stored program*), senza però addurre motivazioni di carattere teorico, ma accennando ad una giustificazione pratica: "La capacità [della memoria] per questa finalità [conservare le istruzioni] potrà essere stimata solo dopo che sarà stato deciso l'insieme delle istruzioni e saranno stati formulati alcuni problemi tipici". In altri termini, essendo il computer per 'uso generale', non è possibile conoscere a priori quanta memoria richiederanno i dati e quanta il programma per ogni singolo compito, è perciò conveniente usare lo stesso dispositivo che, di volta in volta, sarà suddiviso tra queste due funzioni, evitando i costosi sprechi che si avrebbero con due memorie separate. Le giustificazioni della proposta del 'programma memorizzato' sono quindi eminentemente pratiche, imposte da costrizioni tecniche ed economiche: una memoria di lettura e scrittura ad accesso rapido, per non rallentare l'elaborazione elettronica, ed una sola memoria per dati e programmi al fine di garantire plasticità alla macchina e limitarne complessità e costo. Per inciso, il termine 'programma memorizzato' è ambiguo, anche in tutte le macchine precedenti il programma era 'memorizzato' in qualche forma, su schede, su nastri o connettendo cavi elettrici come in ENIAC.

Un'altra innovativa caratteristica, che chiameremo 'auto-modificabilità del programma' (*self-modifying program*), è la possibilità che ogni istruzione del programma possa essere trattata come un numero,²⁵ quindi sia assoggettabile ad elaborazione da parte del programma stesso durante la sua esecuzione. Questo concetto, considerato in molte trattazioni come 'inevitabilmente' connesso a quello di 'programma memorizzato, non è espresso in modo esplicito nella *First Draft* ma lo si può evincere dall'ultima sezione, quella che descrive il *set* di istruzioni, senza però che ne venga giustificato l'utilizzo con alcun esempio, Le istruzioni sono codificate come numeri binari, al pari dei dati numerici, ma per discernere i due tipi di informazione si propone che "queste due categorie devono essere distinte l'una dall'altra dai loro rispettivi primi bit". La soluzione proposta nel

²⁵ Anche nei nastri perforati di Zuse ed Aiken e nelle schede di Babbage la presenza o assenza di un foro può essere interpretata come una cifra binaria e quindi l'intera istruzione come un numero.



rapporto, di fatto, introduce una separazione virtuale, non fisica, tra dati e programma. Alcune istruzioni permettono di trasferire un'istruzione nell'unità aritmetica, di processarla come fosse un dato numerico e di riscrivere il risultato nella memoria come nuova istruzione modificata. La modificabilità dell'istruzione, secondo la *First Draft* è però limitata al solo indirizzo di memoria.²⁶

Per la sua stessa natura e per la sua forma, è difficile pensare che la *First Draft* potesse essere stata usata come un progetto, una 'ricetta' da seguire per costruire un computer. Nel 1945 alla *First Draft* seguì un breve rapporto di Presper Eckert che spiegava più chiaramente il concetto di 'stored program' [3] : "EDVAC conserverà le istruzioni operative esattamente nello stesso modo con cui conserva i numeri. La capacità totale della memoria sarà a disposizione dell'operatore per essere allocata, a suo piacimento, parte alle istruzioni, parte alle tavole delle funzioni, parte ai numeri che entrano ed escono in ogni operazione [...] alcune istruzioni possono essere modificate cancellandone una parte ed inserendo qualcos'altro al suo posto. Questo permette [...] di utilizzare le tavole delle funzioni". In altre parole, Eckert spiega come la modifica dell'indirizzo di una istruzione permetta di gestire elenchi ordinati o matrici di dati numerici, situazione che si riscontra comunemente nella soluzione di sistemi di equazioni.

5. Sviluppo e diffusione dell'idea

La distribuzione della *First Draft*, ma non solo questo fatto, causò un certo trambusto alla *Moore School*. [27] Il gruppo di ENIAC ed EDVAC si divise in due correnti e ne seguì una diaspora: Eckert e Mauchly si dimisero dalla *Moore School* per fondare un'impresa per la produzione e commercializzazione di calcolatori elettronici, la EMCC, la prima di questo genere negli USA.²⁷ [27] Von Neumann terminò la consulenza per EDVAC alla *Moore School* e si dedicò al progetto di un nuovo calcolatore da installare allo *Institute for Advanced Study*, progetto che richiese gran parte delle sue energie per convincere i direttori dell'Istituto, poco desiderosi di vedere il tempio della ricerca teorica invaso da tecnici ed ingegneri. Un secondo onere di von Neumann riguardava la raccolta dei fondi necessari; questi furono ottenuti dalle tre forze armate e dall'Ente per l'Energia Nucleare, grazie agli ottimi rapporti che il matematico aveva instaurato con l'*establishment* politico-militare ed ai timori generati dallo spettro della Guerra Fredda. Goldstine e Burks seguirono von Neumann a Princeton.[17] Alla *Moore School* la costruzione di EDVAC fu affidata prima a T. Kite Sharpless, poi a Louis Tabor, infine a Richard L. Snyder e la macchina fu consegnata allo Aberdeen proving Ground nel 1949, ma ancora necessitava di numerosi interventi ed entrò in uso solo nell'ottobre 1951. Come fa notare Williams, "EDVAC non fu costruito come prevedeva il

²⁶ Le istruzioni sono generalmente costituite da un codice che identifica l'operazione da svolgere (op-code) e dagli indirizzi di memoria degli operandi e del risultato. A volte può essere aggiunto l'indirizzo dell'istruzione successiva.

²⁷ In Germania Zuse aveva già fondato nel 1941 la Zuse Ingenieurbüro und Apparatebaum, dedicata alla produzione e commercializzazione di macchine per il calcolo automatico.



progetto originale, non era particolarmente affidabile,²⁸ ed alla fine fu così tanto modificato che sarebbe stato irricognoscibile a coloro che lo progettarono nel 1945.” [28]

Tra il 1946 e il 1950 si assistette ad un fiorire di incontri, pubblicazioni e contatti personali che contribuirono da una parte ad una maggior comprensione e diffusione dei concetti del ‘programma memorizzato’, dall’altra al sorgere di numerose iniziative tese all’urgente costruzione di computer su entrambe le sponde dell’Atlantico. Il primo di questi eventi²⁹ fu il corso organizzato alla *Moore School* nell’estate del 1946, intitolato “Teoria e tecnica per il progetto di calcolatori elettronici digitali”. [29] Al corso, durato due mesi, parteciparono diciannove docenti e ventotto studenti, provenienti da enti civili e militari, anche dall’Inghilterra. In quella sede Mauchly, Eckert e Sharpless tennero alcune lezioni, tra le quali “Progetto della macchina e codici delle istruzioni” e “Descrizione preliminare di una macchina digitale per il calcolo”; nella prima Mauchly ribadì il significato del ‘programma memorizzato’ con queste parole: “Questo requisito è in parte richiesto dal desiderio di semplificare il progetto e la costruzione della macchina, evitando di dover provvedere diversi tipi di dispositivi per diversi tipi di memoria, e anche per non dover decidere a priori quale sarà la proporzione della memoria da dedicare alle istruzioni e quale ai dati”, aggiungendo che “una ragione ancor più fondamentale è che le istruzioni stesse devono poter essere elaborate da altre istruzioni ... [questa possibilità] è necessaria per poter gestire sequenze di istruzioni memorizzate in differenti parti della memoria, che necessitano [durante l’esecuzione] di modifiche agli indirizzi di memoria associati alle diverse istruzioni.” Mauchly fornisce qui una nuova giustificazione della auto-modificabilità del programma, la gestione delle *subroutine*.³⁰ Al corso delle *Moore School*, seguì un evento ancor più frequentato, il “Simposio sulle grandi macchine per il calcolo digitale”, [30] organizzato da Howard Aiken e tenuto ad Harvard dal 7 al 10 gennaio 1947, dove nuovamente Mauchly divulgò il ‘verbo’ del programma memorizzato. Al simposio furono presentate quaranta relazioni seguite da 168 tra matematici, fisici, ingegneri elettronici, sia civili sia militari, provenienti anche dall’Europa. L’intervento di Mauchly andò ben oltre ai contenuti della *First Draft*, spingendosi ad analizzare l’uso di biblioteche di *subroutine* e preconizzando l’impiego di ‘assemblatori’ per organizzarle nel programma definitivo.

Nel frattempo von Neumann, Goldstine e Burks avviavano il progetto del computer dello IAS. I principi su cui questa macchina si fondava sono descritti in alcuni rapporti interni, quali la “Discussione preliminare sul progetto logico di uno strumento elettronico per il calcolo” del 1946 e “Sui

²⁸ *EDVAC fu modificato più volte tra 1951 e il 1961, ma la sua affidabilità rimase sempre piuttosto bassa, operando correttamente non più di 8 ore al giorno [27].*

²⁹ *Nell’ottobre 1945 si era già tenuta una conferenza al MIT, intitolata Digital Computer Techniques, ma era coperta da segreto militare e la partecipazione fu limitata al solo personale autorizzato e su invito.*

³⁰ *Il concetto di subroutine era già stato preconizzato da Ada Lovelace per la Macchina Analitica di Babbage [Morrison P, Morrison E., Charles Babbage and his Calculating Engines, Dover Pubbl., 1961] ed introdotto da Aiken e Hopper per lo ASCC [Beyer K.W., Grace Hopper and the Invention of the Information Age, MIT Press, 2009]*



principi delle grandi macchine da calcolo”.³¹ [31, 32] I documenti, considerati confidenziali, non furono formalmente pubblicati, ma i concetti esposti circolarono piuttosto liberamente³² nella piccola comunità degli addetti. In entrambi gli scritti si ripropone in modo esplicito il concetto di ‘*stored program*’: “ [Se] le istruzioni sono ridotte a codici numerici e se la macchina può in qualche modo distinguere un numero da un’istruzione, l’organo di memoria può essere usato per immagazzinare sia numeri sia istruzioni.”. Riguardo alla auto-modificabilità del programma, si porta come esempio l’operazione su dati appartenenti ad una lista di valori (una tavola di funzioni o una matrice): “Quindi vi è necessità di un’istruzione che possa sostituire un numero [indirizzo] in un’altra istruzione”, cioè “La sostituzione parziale, in cui la parte di istruzione contenente un indirizzo di memoria – assumendo che le diverse posizioni di memoria siano numerate sequenzialmente - possa essere sostituita da un indirizzo diverso”.[31] “Dobbiamo aggiungere che questa tecnica della sostituzione automatica delle istruzioni, cioè la capacità della macchina di modificare le proprie stesse istruzioni (sotto il controllo di un’altra istruzione) è assolutamente necessaria per una programmazione flessibile. Quindi, se una parte della memoria è usata come ‘tabella di funzioni’, allora il reperimento di un valore della funzione per un certo valore della variabile ottenuto nel corso del calcolo richiede che la macchina stessa debba modificare, o meglio ‘costruire’, il riferimento alla memoria [l’indirizzo] nella istruzione che controlla la localizzazione del dato.”[32] Più oltre si giustifica tale sostituzione anche nel caso di trasferimenti condizionati e non-condizionati a *subroutine*. Diversamente dalla *First Draft*, queste trattazioni erano destinate ad un pubblico di ricercatori e di amministratori civili e militari, certo non digiuni di matematica, ma al loro primo incontro con la nuova tecnologia del calcolo automatico digitale ed elettronico; ciò spiega la maggiore chiarezza ed il ricorso ad esempi presi da applicazioni di immediata utilità. La costruzione del computer dello IAS seguì un lungo e faticoso *iter*, sotto la guida di von Neumann, Goldstine, Burks e Julian Bigelow, costellato da non poche modifiche, che terminò solo nel 1952, sei anni dopo l’inizio del progetto. Molte delle soluzioni adottate sono diverse sia dalla *First Draft*, sia dal definitivo EDVAC, prima tra le quali l’esecuzione delle operazioni in parallelo sull’intera parola di 40 bit, non un bit alla volta come in EDVAC, con notevole aumento di velocità di calcolo.

³¹ All’epoca l’articolo non fu pubblicato e contiene materiale tratto da diversi interventi di von Neumann, in particolare dall’incontro tenutosi il 15 maggio 1946 al *Mathematical Computing Advisory Panel del centro di ricerche della Marina (Office of Naval Research, ONR)*.

³² Della ‘*Discussione preliminare...*’ ne furono distribuite 175 copie.



6. I primi computer a “programma memorizzato” e il contributo di Turing

Nei due-tre anni successivi copie di prima o seconda mano della *First Draft* raggiunsero molti più destinatari dei ventiquattro scelti da Goldstine; così fu anche per i rapporti e le minute delle altre conferenze citate. A ciò si aggiunsero numerosi incontri ufficiali ed informali che molti europei ed americani ebbero presso i centri più avanzati, l'Università di Harvard, la Moore School, i Bell Laboratories, il MIT. Almeno una dozzina di scienziati ed ingegneri britannici si recarono negli USA tra il 1945 e il 1950, sulla scorta degli accordi di collaborazione stabiliti durante la guerra.³³ E' difficile stabilire una relazione causa-effetto, cioè se furono le informazioni su ASCC, ENIAC ed EDVAC a stimolare numerosi progetti di costruzione di grandi calcolatori automatici programmabili o se tale interesse era preesistente, imposto da necessità contingenti create dalla scienza e dalla tecnologia. E' molto probabile che si sia verificato un inestricabile connubio tra le due situazioni, con reciproco effetto di rinforzo. Comunque sia, nella seconda metà degli anni 1940 in molti centri sia negli USA sia nel Regno Unito nacquero contemporaneamente diversi progetti per la costruzione di computer a 'programma memorizzato'.

Quello che viene comunemente considerato il primo³⁴ a giungere a compimento fu quello dell'università di Manchester, voluto e guidato dal matematico Max Newman,³⁵ che era stato docente di Alan Turing ed aveva lavorato su Colossus a Bletchley Park. Newman si appoggiò alle capacità ingegneristiche di Freddie Williams e di Tom Kilburn, entrambi del Telecommunications Research Establishment di Malvern. Il 21 giugno 1948, tre anni dopo la circolazione della *First Draft*, un piccolo prototipo, battezzato SSEM (*Small-Scale Experimental Machine*) o 'Manchester Baby', eseguì con successo un semplicissimo programma 'memorizzato' di 17 istruzioni, citato come il primo caso di 'architettura di von Neumann' in funzione.³⁶ La versione in scala definitiva della macchina, Manchester Mark I, divenne operativa un anno dopo. Il secondo progetto inglese fu lo EDSAC (*Electronic Delay Storage Automatic Calculator*), dovuto a Maurice Wilkes del *Mathematical Laboratory* di Cambridge; questa macchina entrò in funzione il 6 maggio 1949. [33] Il terzo computer a 'programma memorizzato' vide la luce nella lontana Australia, lo CSIRAC (*Council for Scientific and Industrial Research Automatic Computer*) di Trevor Pearcey e Maston Beard, che eseguì il suo primo programma nel novembre 1949 [34].

³³ Ad esempio la *Missione Tizard*, dal nome del suo organizzatore, il chimico Sir Henry Tizard, che gettò le basi di una proficua collaborazione tecnico-scientifica sulle tecnologie militari tra Gran Bretagna e Stati Uniti, quando questi ultimi ancora neutrali.

³⁴ La priorità dei diversi progetti è discutibile, dipende infatti da quale evento si prende in considerazione: un prototipo in scala ridotta, la prima esecuzione di un programma, la consegna al committente, l'entrata in regolare operatività.

³⁵ Max Newman era stato nominato alla cattedra di matematica pura dell'Università di Manchester.

³⁶ Alcuni suggeriscono che anche un'originale gigantesca macchina costruita nel 1948 da IBM, lo SSEC (*Selective Sequence Electronic Calculator*) possa essere stata la prima ad incorporare il 'programma memorizzato', ma l'ipotesi è discutibile essendo la memoria elettronica ad accesso rapido di SSEC troppo piccola, solo 8 registri. [Olley A., *Existence Precedes Essence - Meaning of the Stored-Program Concept*, <http://individual.utoronto.ca/fofound/RevisedExistEss1.pdf>]

Negli Stati Uniti gli unici computer terminati prima del 1950 furono, oltre ad EDVAC che fu consegnato al BRL nel 1949, il BINAC (*Binary Automatic Computer*), prodotto dalla neonata azienda di Mauchly e Eckert (EMCC),³⁷ che incontrò numerosi problemi di messa a punto e non divenne mai completamente operativo, [27] e l'unità di calcolo dello Whirlwind I del MIT, progettato da Jay Forrester e Robert Everett, che per dimensioni e prestazioni potrebbe essere considerato un 'supercomputer,' almeno per l'epoca.[35] Questi progetti furono senza dubbio ispirati dalla *First Draft* e, in linea di principio, riconducibili ai concetti espressi nel documento, ma nella pratica furono piuttosto diversi tra loro e tutti diversi dall'originale disegno di EDVAC. Nei primi anni 1950 saranno terminati il SEAC (*Standard Eastern Automatic Computer*) e lo SWAC (*Standard Western Automatic Computer*) del *National Bureau of Standards*, lo Whirlwind I e la macchina di von Neumann allo IAS, preceduta e seguita negli stessi anni una lunga serie di copie realizzate in diverse parti del mondo, grazie alla grande liberalità con cui von Neumann condivideva conoscenze ed esperienze. Neppure questi sono veramente 'cloni' della macchina dello IAS, ma altrettante variazioni del progetto originale.[36] Le idee americane e inglesi filtrarono anche attraverso la cortina di ferro, permettendo all'ucraino Sergei A. Lebedev di costruire il MESM (*Malaya Elektronnaya Schetnaya Mashina*, Piccola macchina calcolatrice elettronica) tra il 1948 e il 1951 sulla sola base di scarse informazioni ricevute dal mondo occidentale, pochi articoli giornalistici divulgativi o brevi riassunti, caso paradigmatico di 'copia dell'idea',³⁸ è infatti assai improbabile che Lebedev abbia avuto accesso alla *First Draft* o ad altri documenti occidentali in quegli anni.[53]

In Inghilterra era in corso un'altra iniziativa, quella del *National Physical Laboratory*.³⁹ A soli due mesi dalla sua stampa una copia della *First Draft* era giunta sulla scrivania di John R. Womerlsey direttore della Divisione di Matematica che da un anno stava immaginando la costruzione di un calcolatore programmabile digitale. Ottenuta l'approvazione e il



Alan Mathison Turing
(1912 – 1954)

³⁷ Secondo Herman Lukoff, uno degli artefici di BINAC, questo computer eseguì il primo programma nel marzo 1949, precedendo quindi Manchester Baby. [Lukoff, H., *From Dits to Bits*, Robotic Press, 1979].

³⁸ Con 'copia dell'idea' o 'diffusione dello stimolo' si intende la situazione in cui un inventore realizza indipendentemente un artefatto basandosi sulla sola consapevolezza che questo è già stato realizzato da altri.

³⁹ Lo NPL è l'ente nazionale britannico di riferimento per gli standard di misura, nonché il maggiore laboratorio di fisica applicata del Regno Unito. Nel 1945 lo NPL era diretto da Charles G. Darwin, nipote di quel Charles R. Darwin che ottant'anni prima aveva proposto la teoria dell'evoluzione.



finanziamento statale, Womersley invitò Turing alla direzione del progetto.[4] Alan Turing (1912-1954) era un matematico di 33 anni che aveva acquisito notorietà per un suo articolo del 1936 intitolato “Sui numeri calcolabili. Con una applicazione al problema della decidibilità”. Per la dimostrazione della sua tesi Turing ricorreva ad una macchina ideale, la ‘Macchina Universale’ (Riquadro 1) che sarà spesso, ma a posteriori, considerata il fondamento teorico del computer per uso generale. Durante la guerra Turing aveva lavorato al centro di crittoanalisi di Bletchley Park riuscendo ad ideare metodi per la decrittazione della macchina tedesca Enigma ed a meccanizzarli con una serie di macchine logiche chiamate ‘Bombe’. Entrambe le esperienze, quella teorica e quella pratica,⁴⁰ oltre alle notevoli capacità di logico-matematico, facevano di Turing il soggetto ideale per intraprendere il progetto dello NPL. [37, 38]

Womersley fornì a Turing una copia della *First Draft* e, prima della fine del 1945, questi presentò allo NPL un lungo rapporto dal titolo “Proposta per un Calcolatore Elettronico” (*Proposed Electronic Calculator*, d’ora in poi semplicemente *Proposed*) [4] che verrà battezzato ACE (*Automatic Computing Engine*).⁴¹ Lo scritto di Turing è molto più particolareggiato della *First Draft*, spingendosi fino alla descrizione dei circuiti elettronici ed al calcolo dei costi – valutazione che non era di sua competenza.⁴² Turing adottò il concetto di programma memorizzato e, con poche modifiche, anche la simbologia ‘neuronal’⁴³ della *First Draft*, citando quest’opera e consigliandone la lettura come testo propedeutico al proprio, ma sviluppando il progetto in modo sostanzialmente diverso. [4] Nello ACE non vi è separazione netta tra memoria ed unità di calcolo; in altri termini, nella memoria di ACE vi sono registri dedicati alle principali operazioni aritmetiche e logiche, ma non vi è un’unità aritmetica centrale. Turing adotta anche il concetto di auto-modificabilità del programma che anzi estende alla modifica dell’intera istruzione, non solo dell’indirizzo dell’operando: “Questa funzione [il salto condizionato] può essere svolta registrando le istruzioni in una forma di memoria riscrivibile, ... Ciò conferisce alla macchina la possibilità di costruire le sue stesse istruzioni.”⁴⁴ Nel rapporto Turing porta un esempio di come sia possibile il ‘ricalcolo’ dell’istruzione nel caso di salto condizionato. L’idea di Turing è quella di una macchina semplificata al massimo, con un set di istruzioni

⁴⁰ Turing, pur essendo un teorico, era appassionato di tecnologia. Già alla fine degli anni '30 pensava di costruire una versione fisica della sua macchina ideale (Riquadro 1). Durante la sua permanenza a Princeton nel 1937-38 costruì un moltiplicatore elettronico [37] e durante la guerra tenne lezioni sulle valvole termoioniche [38].

⁴¹ Si noti che Turing usa il termine ‘engine’, come Babbage, invece di calculator o computer.

⁴² Il costo previsto da Turing era di sole 11.200 £, molto sottostimato rispetto alle 50.000 £ che costò il solo modello pilota (Pilot ACE, vedi oltre).

⁴³ Durante la guerra Turing ebbe modo di incontrare McCulloch. Questi aveva preso spunto da “Sui numeri calcolabili” di Turing per la sua teoria del cervello come macchina logica. [36, 38]. Neppure Turing cita Shannon nonostante gli avesse fatto visita nel 1943 ed avesse discusso con lui su diversi argomenti di reciproco interesse. [37]

⁴⁴ Per il ritorno da una subroutine Turing dedica una locazione della memoria, separata dal programma, per registrare l’indirizzo dell’istruzione immediatamente successiva al ‘ritorno’ dalla subroutine, una specie di stack ‘first-in last-out’, utilizzando due istruzioni apposite che chiama BURY e UNBURY (seppellisci e disseppellisci). [4]



ridotto all'essenziale; egli critica apertamente “la tradizione americana di risolvere le difficoltà per mezzo di grandi strumenti piuttosto che con il pensiero” [39] ed è piuttosto negativo verso progetti di Newman a Manchester e di Wilkes a Cambridge, che seguivano l'architettura di von Neumann in modo più ortodosso. Per la memoria Turing si orientò subito verso la tecnologia considerata più sicura, quella delle linee di ritardo acustiche (Riquadro 2), ma queste non avrebbero garantito l'accesso immediato alle informazioni – per quanto veloce si trattava di una memoria sequenziale, non ad accesso casuale. Volendo mantenere una sufficiente velocità, le istruzioni e i dati dovevano quindi essere registrati nella memoria non in ordine logico, ma disposte in modo tale da essere disponibili immediatamente quando richiesti. La conseguenza era una pesante complicazione del processo di programmazione, ma questo difetto non preoccupava Turing che sosteneva che “[i programmi] dovranno essere costruiti da matematici con esperienza di calcolo e forse con un certa abilità nella soluzione di rompicapo” [40] ACE è quindi tanto differente da EDVAC da postulare una ‘architettura di Turing’.⁴⁵

Tra dicembre 1946 e febbraio 1947 Turing, e il suo assistente Jim Wilkinson, tennero una serie di nove conferenze [4] sull'architettura dei computer⁴⁶ rivolte ad una ventina di partecipanti dal mondo dell'industria elettronica, delle telecomunicazioni e della matematica.⁴⁷ In quest'ambito Turing spiegò ancor più chiaramente la sua visione del computer, dichiarando di essere “più interessato alla possibilità di produrre modelli dell'azione del cervello che alle applicazioni pratiche del calcolo” e spingendosi ad affermare che “... la possibilità di permettere alla macchina di alterare le sue stesse istruzioni produce un meccanismo che può imparare dalla sua stessa esperienza”. [4] Si evince da queste frasi quanto gli obiettivi di Turing divergano da quelli di von Neumann: quest'ultimo era motivato dalla necessità impellente di disporre di un dispositivo pratico per risolvere problemi per la fisica e la tecnologia, mentre Turing desiderava disporre di uno strumento sperimentale su cui provare la sua tesi per una ‘macchina che pensa ed apprende’. [39] La diversità tra i due approcci appare anche dai diversi esempi di utilizzo di un computer che si trovano nella *First Draft* e nel *Proposed* di Turing: Nella prima von Neumann cita solo problemi di matematica applicata, nel secondo Turing aggiunge la soluzione di *puzzle* e il gioco degli scacchi. Per von Neumann il computer è quindi uno strumento per la matematica applicata, per Turing è l'oggetto stesso della ricerca.

Per una curiosa coincidenza la costruzione di ACE subì un destino simile a EDVAC. Turing, preoccupato per i ritardi accumulatisi allo NPL, lasciò il laboratorio nel 1947 per raggiungere Max Newman a Manchester dove contribuì alla programmazione del Manchester Mark I ed ebbe finalmente modo di usare un computer per le sue ricerche, anche se uno molto diverso da ACE. I lavori

⁴⁵ Il ridotto numero di istruzioni fa considerare ACE un precursore della moderna architettura RISC (Reduced Instruction Set).

⁴⁶ Nello stesso periodo anche Max Newman tenne a Manchester una serie di conferenze sul computer. [14]

⁴⁷ Tra di essi anche Maurice Wilkes che costruirà EDSAC a Cambridge.



su ACE rallentarono ulteriormente, affidati successivamente alla responsabilità di Thomas H. Flowers, Harry D. Huskey, H. A. Thomas e infine a Jim Wilkinson e Francis M. Colebrook; solo questi ultimi riuscirono a portare a termine il modello in scala ridotta, detto *Pilot ACE*, che eseguì il suo primo programma il 10 maggio 1950, ma entrò in uso solo un anno dopo. [4] La macchina ridotta, nonostante le piccole dimensioni, si dimostrò più veloce ed efficiente di altri computer coevi. L'architettura del *Pilot ACE* fu adottata in un prodotto della English Electric, il DEUCE, di cui ne furono costruiti 33 esemplari e fu riproposta in altre due piccole macchine commerciali americane degli anni '50, la Bendix G15 e la Packard-Bell PB 250, oltre al il gigantesco MOSAIC del 1952. La versione definitiva di ACE (*Big ACE*) fu terminata solo nel 1958, ma ormai la sua tecnologia era decisamente obsoleta. [4]

7. Conclusioni

I concetti di 'architettura di von Neumann', 'programma memorizzato' e 'programma auto-modificabile' sono spesso presentati nella letteratura come sinonimi o come inestricabilmente connessi l'uno all'altro, caratteristiche indispensabili per la realizzazione di computer programmabili per uso generale, in breve l'unico modo di realizzare in pratica la Macchina Universale di Turing (Riquadro 1). La *First Draft* sarebbe il documento che lo attesta e von Neumann sarebbe quindi l'inventore del moderno computer. A questo proposito, il fisico Stanley Frankel⁴⁸ affermò "Molte persone hanno acclamato von Neumann come 'il padre del computer' (nel senso moderno del termine), ma io sono sicuro che lui non avrebbe mai commesso lo stesso errore. Potrebbe esserne chiamato la 'levatrice', forse, ma dichiarò fermamente a me e ad altri che i concetti fondamentali sono dovuti a Turing, almeno per quanto non anticipato da Babbage e altri." [41]

L'influenza che la Macchina Universale di Turing potrebbe aver avuto sulle architetture⁴⁹ dei primi computer o sugli scritti di von Neumann e di Turing del 1945 non è però così facilmente identificabile. La memoria delle Macchine di Turing è un nastro infinito ad accesso sequenziale, le singole informazioni non sono individualmente indirizzabili e le 'tavole di istruzioni', l'equivalente del programma, sono costituite solo da operazioni condizionali (Riquadro 1), infine il 'programma' della Macchina Universale non è auto-modificabile; l'architettura di questa macchina ideale è quindi sostanzialmente diversa da quelle di EDVAC e di ACE.[46] Negli anni successivi sia Turing sia von Neumann, ma anche molti altri come Max Newman, iniziarono a riferirsi, negli scritti e nelle conferenze, all'articolo di Turing del 1936 come al fondamento teorico per la realizzazione del computer. Lo stesso von Neumann ne suggerì la lettura ai suoi

⁴⁸ S. Frankel fu il fisico che sviluppò le tecniche di calcolo basate su sistemi meccanografici a schede perforate per l'atomica a Los Alamos e collaborò a lungo con von Neumann. Anche Cuthbert C. Hurd della IBM, che lavorò a lungo con von Neumann, ricorda "Non lo sentii mai reclamare di aver inventato lo stored program." [12]

⁴⁹ Secondo Alan Doran il termine 'architettura' indica la relazione tra la struttura fisica (hardware) della macchina e il vocabolario delle istruzioni (instruction set). [4]



collaboratori, ma come afferma Brian Randell, “è probabile che non saremo mai in grado di stabilire esattamente quale impatto Turing abbia avuto sulla genesi del computer” [41]. “I dettagli di questa storia non confermano l’ipotesi che il computer si sviluppò come prodotto collaterale o applicazione del lavoro teorico di Turing e che gli anni successivi 1946-55 si concentrarono solo su aspetti ingegneristici” [46]. Così si esprime anche Atsushi Akera : “Sebbene il rapporto di von Neumann sia spesso citato come la descrizione ‘logica’ di un moderno computer, il documento non è una trattazione formale di logica matematica. Neppure è principalmente centrato sul concetto di ‘programma memorizzato’. La logica formale è rilevante solo in una piccola parte dell’analisi di von Neumann; per sua stessa ammissione, egli descrive piuttosto la struttura ‘generale’ di una macchina per il calcolo”, e aggiunge che “La *First Draft* fu eminentemente un documento pratico”. [55] Anche M. Priestley afferma che “Nonostante il ricorso ad elementi logici (*E-element*) la logica elementare fu certamente utile ... ma ciò non dimostra una specifica influenza della logica sul progetto di EDVAC”. [46] Si può ipotizzare che la Macchina Universale influì in modo forse inconscio su alcuni dei pionieri del computer nella seconda metà degli anni 1940 – soprattutto fra i matematici, molto meno fra gli ingegneri – ma la testimonianza di tale influenza non trova riscontro in citazioni esplicite in alcuno dei documenti di quegli anni, in nessuno dei quali si tenta una connessione tra “Sui numeri calcolabili” e le architetture proposte. Per contro, è innegabile un punto di contatto tra la Macchina Universale e l’architettura di ACE: in entrambi i casi si tratta di un apporto ‘minimalista’,⁵⁰ la riduzione dell’hardware al minimo indispensabile, seppure a spese di un software più difficile da implementare e di una minore velocità di esecuzione. [4, 37] Una vera teoria matematica del calcolo automatico, che prenderà le mosse dalla logica simbolica (Boole e Shannon) e dalla teoria della computazione (Turing), a cui si aggiungerà la teoria dei linguaggi formali di Noam Chomsky, si svilupperà solo più tardi [45]; “Infatti, l’interesse storiografico del concetto di programma memorizzato si origina in realtà con l’emergere di una scienza teorica del computer, sviluppo che inizierà 20 anni dopo.” [55]

Il primo concetto espresso nella *First Draft*, la separazione tra unità di calcolo, memoria e unità di controllo, abbastanza intuitivo e già messo in atto da altri, è dettato dalla necessità di una struttura semplice in cui sia facilmente identificabile la funzione di ciascuna parte in relazione alle altre. Lo stesso può dirsi per l’uso della notazione binaria. Il secondo concetto, quello del ‘programma memorizzato’, cioè l’uso di una sola memoria ad accesso veloce per immagazzinare dati e programmi, è pensato come l’approccio più pratico per evitare un costoso sovradimensionamento di tale organo. Le ipotesi di precursori del ‘programma memorizzato’, Babbage e Zuse, [42, 43, 44] sono basate su indizi piuttosto tenui e ogni influenza dei presunti anticipatori sulla *First Draft* può essere esclusa, essendo le loro opere poco o per nulla conosciute nel 1945. Eckert e Mauchly sostennero

⁵⁰ L’approccio minimalista fu continuato da altri teorici, tra cui W.L. van der Poel per la macchina ZERO, da H. Proesch e Th. Fromme per la MINIMA e da Wes Clark per il TX-0, uno dei primi computer a transistor del MIT.



a lungo [12] che l'idea di memorizzare le istruzioni assieme ai dati fosse nata alla Moore School prima dell'arrivo di von Neumann, adducendo come prova un documento confidenziale dell'inizio del 1944, [15] ma la breve nota si presta ad interpretazioni discordanti. In realtà, la scelta di due memorie separate per dati e programmi non avrebbe influito sulle prestazioni, posto che la tecnologia adottata per entrambe avesse garantito un accesso sufficientemente veloce al singolo dato o alla singola istruzione in modo casuale. Anche la Macchina Universale di Turing potrebbe funzionare con due nastri separati. Come afferma Mark Priestley, "Fino al 1950 il concetto di *stored program* era visto più come una soluzione tecnica resa necessaria dall'uso dell'elettronica nell'unità di calcolo, e successivamente come caratteristica costruttiva che permetteva una più semplice programmazione, ma non come una proprietà essenziale del computer".[46] Quanto all'obbligatorietà che un computer sia a 'programma memorizzato' perché possa essere considerato 'universale', Charles Petzold asserisce che : "Il primo computer ad essere universale, almeno potenzialmente, fu la macchina chiamata Z3 che Konrad Zuse costruì tra il 1938 e il 1941 ... La Macchina Analitica di Babbage, se fosse stata costruita, sarebbe stata qualificabile come una macchina universale", [47] affermazione ribadita da R. Rojas [48]. Anche lo stesso Turing, in un articolo del 1950, aveva esplicitamente confermato che "la Macchina Analitica [di Babbage] era un computer universale" [49] a dimostrazione della non necessità di un 'programma memorizzato'. [46] Si può concludere quindi che lo '*stored program*' sia una 'condizione sufficiente ma non necessaria' per la realizzazione di una macchina universale.

Il terzo concetto che viene associato alla 'architettura di von Neumann' è quello della auto-modificabilità del programma. Le giustificazioni che vengono portate nei documenti successivi alla *First Draft*, sono tre: 1) possibilità di gestire dati numerici organizzati in liste o matrici , 2) possibilità di cambiare l'indirizzo dell'istruzione successiva (come nel caso di subroutine o salti condizionati), 3) possibilità di eseguire cicli di calcolo. In linea di principio però non è impossibile che una macchina programmata con schede o nastri perforati possa eseguire cicli, salti condizionati e subroutine; assegnando ad ogni istruzione e ad ogni dato un'etichetta indirizzabile dall'unità di controllo, ripetendo più volte le stesse istruzioni nel programma o disponendo di più nastri in parallelo,[46] anche se ciò avverrebbe a grave scapito della velocità di esecuzione. Nella pratica la auto-modificabilità fu adottata solo su poche delle prime macchine [50] e fu presto resa inutile dall'invenzione di registri di memoria temporanea già implementati nel Manchester Mark I del 1949, le c.d. *B-line*, che permettevano di modificare gli indirizzi di memoria senza agire direttamente sulle istruzioni in memoria. Presto la auto-modificabilità sarà considerata una soluzione 'poco elegante' e potenzialmente pericolosa.[51] Infatti, se le istruzioni venissero modificate durante l'esecuzione del programma, ogni nuovo utilizzo dello stesso richiederebbe il loro ripristino alle condizioni iniziali o la reintroduzione in memoria del programma a partire dalla sorgente originale. Oggi gli unici programmi che operano sulle istruzioni, ma di altri programmi e non di se stessi, sono assembleri, compilatori ed



interpreti, strumenti che permettono di tradurre linguaggi di programmazione ad alto livello in codice macchina e che entrarono in uso non prima del 1955. Alle situazioni previste da von Neumann e dai suoi collaboratori per giustificare la auto-modificabilità, Turing aggiunse la possibilità che il computer possa modificare il proprio programma per simulare la capacità di apprendimento, aprendo la strada agli studi sulla intelligenza artificiale,⁵¹ ma i ricercatori in questa disciplina adottarono strumenti diversi, come i simulatori di reti neurali.

La *First Draft* agì certamente da potente stimolo per il gruppo di innovatori che si accingevano a costruire i primi computer, ma ognuno interpretò a modo suo i concetti che vi erano espressi e l'architettura subì una veloce e continua evoluzione nel decennio 1945-55, grazie al parallelo evolversi della tecnologia disponibile che rese presto superate le costrizioni pratiche ed economiche dei primi anni. Le diverse soluzioni adottate furono opera di molti contributi indipendenti, ma strettamente interconnessi, che lentamente confluirono in un'architettura comune e condivisa. Von Neumann era tutt'altro che contrario alla libera interpretazione, asserendo che “[la costruzione di un computer] è un tema nuovo ed è la regola, piuttosto che l'eccezione, che due gruppi che lavorano indipendentemente verso obiettivi simili o anche identici possano arrivare a conclusioni differenti. Non è necessario dire che ciò è altamente desiderabile. L'argomento è così nuovo che è più ragionevole tentare diversi approcci e non puntare tutto sulla stessa soluzione”⁵² Con tutto ciò, non può esservi dubbio che von Neumann fu il primo a razionalizzare l'architettura di un computer, anche se la attinse da idee diverse che circolavano in vari ambienti, idee embrionali che il matematico rielaborò in uno schema semplice ed innovativo, evidenziandone gli elementi fondamentali, e ricorrendo ad una simbologia a cui nessun altro aveva pensato, tanto efficace che fu poi adottata dallo stesso Turing e, con simboli diversi, è usata ancora oggi.⁵³ Si può affermare che von Neumann parafrasò idee già diffuse alla *Moore School* e altrove, riuscendo in una visione più astratta, razionale e meglio formalizzata, oltre che indipendente dalla tecnologia adottata. Più che di una dimostrazione assiomatica della necessità dello ‘*stored program*’, nel caso della *First Draft* e dei documenti immediatamente successivi sarebbe meglio parlare di astrazione e formalizzazione. Non vi è dubbio alcuno che le soluzioni proposte nella *First Draft* furono quelle più ragionevoli per la tecnologia dell'epoca, le uniche che permisero l'effettiva realizzazione di macchine utilizzabili nella pratica, e non può essere un caso se l'architettura di von Neumann continui, dopo quasi settant'anni e con tecnologie affatto diverse, a dominare il mondo dell'informatica. Ma ciò che

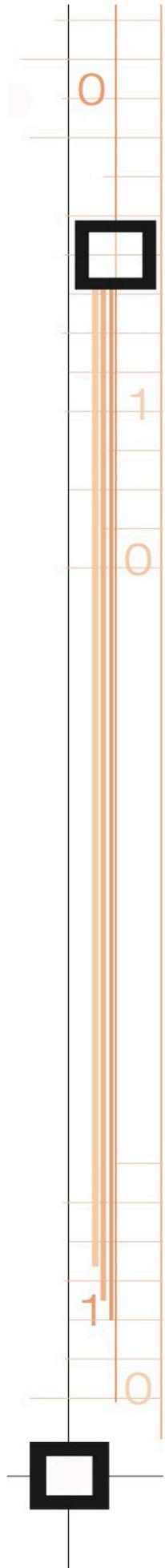
⁵¹ Il termine *Intelligenza Artificiale* sarà coniato negli Stati Uniti in occasione del primo incontro su questa disciplina svoltosi al Dartmouth College nel 1956, ma i primi semi erano stati gettati alle Macy Conference tenute a New York tra il 1947 e il 1953, a cui parteciparono, fra gli altri von Neumann, McCullochs, Pitts e Wiener. Independentemente, Turing proponeva e sviluppava il concetto di ‘*macchina pensante*’, già nel rapporto su ACE del 1945.

⁵² Lettera di von Neumann a Mina Rees del 1947; citata in: K.C Redmond, T.M. Smith, *Project Whirlwind. The History of a Pioneer Computer*, Digital Press, 1980.

⁵³ Nella sua tesi del 1937 Shannon non giunse a proporre una tale simbologia, ma vi si avvicinò di molto. [26]



più contribuì al suo successo nei primi anni fu che le idee erano ‘nell’aria’ – nello *zeitgeist* (spirito del tempo) come dicono i tedeschi – che la tecnologia necessaria era disponibile, collaudata su ENIAC e Colossus, e, soprattutto, che era sorto ovunque un impellente bisogno di calcolo automatico, ciò che lo storico Brian Winston chiama ‘necessità sociale sopravvenente’ (*supervening social necessity*).[54] Fu soprattutto il libero scambio di idee ed esperienze – un’anomalia nei primi anni della Guerra Fredda, spiegabile solo con la mancata percezione dell’importanza strategica del computer – che permise la nascita del moderno computer. E’ questo il merito da riconoscere a von Neumann e a Turing, veri ‘padrini’ del computer moderno.



Riquadro 1 – La Macchina Universale di Turing in breve.

All'inizio del XX secolo il matematico David Hilbert aveva posto alla comunità dei colleghi una serie di problemi che, se risolti, avrebbero permesso di fondare l'intera matematica presente e futura su un numero limitato di assiomi partendo dai quali qualunque affermazione vera (teorema) avrebbe potuto essere dimostrata. Tra questi vi era il problema detto della 'decidibilità' (*Entscheidungsproblem*) cioè trovare un metodo che permettesse di decidere a priori, con una procedura finita, se un'affermazione matematica vera sarebbe stata dimostrabile. Il problema fu affrontato indipendentemente negli anni '30 da Alonzo Church e da Alan Turing e nel 1936 entrambi ne dimostrarono l'impossibilità, ma con due metodi diversi ed indipendenti. L'approccio originale di Turing ricorre ad una 'macchina' teorica che esegue calcoli (Macchina di Turing, *MT*), costituita da un nastro di lunghezza illimitata suddiviso in caselle in ognuna delle quali può esservi scritto un simbolo di un alfabeto finito (ad esempio 0 o 1) oppure essere vuota. Sul nastro scorre una 'testina' che può leggere il simbolo di una casella, cancellarlo, scrivere un altro simbolo o lasciarla vuota. La testina può muoversi lungo il nastro di una sola casella alla volta, a destra o a sinistra, o restare ferma. Il comportamento della testina è determinato, di volta in volta, da una 'tavola delle istruzioni' in essa contenuta. Ogni istruzione è un'operazione condizionale che dipende dallo 'stato' della testina, anch'esso membro di un insieme finito di stati possibili, e dal simbolo letto, e può essere espressa come, ad esempio: "Se la testina si trova nello stato A e la casella contiene uno '0', scrive un '1' nella casella, si sposta a destra di una posizione ed assume lo stato B". Ogni calcolo necessita di una specifica 'tavola delle istruzioni' e di un proprio nastro, quindi di una diversa *MT*, che può essere assimilata ad un computer destinato a quel solo uso specifico e quindi non programmabile.

Turing immagina poi una Macchina Universale (*MUT*) che può eseguire qualunque calcolo. Per far ciò propone che la tavola delle istruzioni di una *MT* particolare sia codificata come un numero, detto 'descrizione standard'. Questo numero può essere registrato all'inizio del nastro illimitato, sempre usando lo stesso insieme di simboli, ma separato dal resto del nastro da un simbolo speciale. La *MUT* inizia a leggere la descrizione standard e converte questa nella tabella delle istruzioni, poi prosegue con il calcolo desiderato. La *MUT* può, cambiando il nastro, simulare una qualunque *MT* o qualunque altra *MUT* dotata dello stesso nastro, e in questo senso è universale. *MT* ed *MUT* sono macchine ideali e Turing non descrive come e se possano essere costruite. Se si tentasse di realizzarle fisicamente sarebbero estremamente lente, dovendo continuamente percorrere il nastro nei due sensi una casella alla volta; inoltre, salvo che per calcoli banali, la programmazione sarebbe piuttosto ardua. L'analogia tra *MUT* e un computer 'per uso generale' è però evidente: La descrizione standard è l'analogo del programma che viene memorizzato nella memoria interna del computer per poi essere eseguito; cambiando il programma, il computer per uso generale può simulare qualunque qualsiasi altra *MT*, quindi qualunque dispositivo di calcolo. In questi termini si può dire che un computer reale è la realizzazione pratica della *MUT* o, in altri termini, una sua approssimazione 'finita', in quanto la memoria disponibile è finita, anche se potrebbe, in linea di principio, essere estesa indefinitamente. [37, 47]. Le relazioni tra *MUT* e computer reali saranno approfondite negli anni 1950 da Huskell B. Curry, Hao Wang, Edward F. Moore e Claude E. Shannon. Quest'ultimo proverà che una *MUT* a due stati e che opera su due soli simboli, come un computer binario, è universale (Shannon C.E., "A Universal Turing Machine with Two Internal States", in: Shannon C.E., McCarthy J., (a cura di), "Automata Studies", *Annals of Mathematics Studies*, n. 34, 1956).

Riquadro 2 – Memorie dei tempi andati.

Per tutti i primi progetti di computer a 'programma memorizzato' l'elemento più critico da realizzare era la memoria. Questa doveva essere ad accesso veloce, riscrivibile e, possibilmente, ad accesso casuale, ma soprattutto sufficientemente grande – von Neumann prevedeva fino a 250.000 bit e Turing fino a 500.000 (circa 62 KByte, la memoria di un *home computer* degli anni '80). Scartando, per l'eccessiva lentezza, sistemi meccanici o elettromeccanici (relè), la tecnologia più promettente sarebbe stata quella elettronica, immagazzinando ogni bit in un circuito bistabile (*flip-flop*) costituito da due valvole. Questo approccio avrebbe comportato un costo (10 dollari/bit) ed un consumo energetico (4 Watt/bit) spropositati; la memoria prevista nella *First Draft* sarebbe costata 20 milioni di Euro in valuta odierna ed avrebbe richiesto 1 megawatt di potenza elettrica. Nella pratica sia EDVAC sia Pilot ACE furono dotati di memorie molto più piccole, 45.000 e 80.000 bit rispettivamente, ma la soluzione elettronica sarebbe stata comunque troppo costosa.

Fortunatamente lo sviluppo del radar durante la guerra aveva reso disponibili due tecnologie che potevano essere, con non facili aggiustamenti, adattate anche ai computer: Le linee di ritardo acustiche (*acoustic delay line*), inventate da William B. Schokley ai Bell Laboratories, e le memorie elettrostatiche a tubo catodico, sperimentate al Massachusetts Institute of Technology.

Le linee di ritardo erano costituite da tubi rettilinei contenenti mercurio, alle cui estremità erano posti due trasduttori elettroacustici a quarzo. Un impulso a onda quadra (rappresentante un bit) provocava nel trasduttore di ingresso l'emissione di un treno di onde ultrasoniche (15 MHz) che si propagava lungo il mercurio alla velocità del suono in quel mezzo (1450 m/sec) per essere captato dal trasduttore di uscita, riconvertito in onda quadra e amplificato da un apposito circuito, quindi rinviato al quarzo in ingresso. In tal modo un certo numero di bit – da 512 a 2048, a seconda della lunghezza della linea e della durata dell'impulso – restava 'memorizzato' nel mercurio. Un computer poteva essere dotato di più linee di ritardo. Il costo per bit di una memoria a linea di ritardo era di soli 4 centesimi di dollaro,[29] ma questa tecnologia non permetteva un effettivo accesso casuale; per accedere al gruppo di bit (parola) corrispondente ad un dato numerico o una istruzione di programma, occorreva attendere che questo 'uscisse' dalla linea di ritardo. Le linee di ritardo necessitavano inoltre di una temperatura costante ed erano pesanti e voluminose. La memoria a linea di ritardo fu adottata in molti dei primi computer, ad iniziare dallo EDSAC di Cambridge (UK).

La memoria elettrostatica faceva uso di un comune tubo a raggi catodici, come quelli usati nei radar, negli oscilloscopi e nei televisori. Quando il fascio di elettroni colpisce lo schermo fluorescente, oltre ad illuminarlo, vi deposita una piccola carica elettrica puntiforme. Accendendo e spegnendo opportunamente il fascio mentre questo viene indirizzato su punti diversi dello schermo è possibile quindi depositarvi un reticolo bidimensionale di cariche che, con un tecniche diverse e piuttosto complesse, possono essere lette e riconvertite in impulsi digitali. Un tubo da 3-6 pollici può memorizzare 1024-2048 bit di informazioni e ogni bit è individualmente indirizzabile, garantendo, a differenza delle linee di ritardo, un vero accesso casuale ad un costo per bit, all'epoca, di



M.V. Wilkes con la memoria a linee di ritardo di EDSAC.
(Computer Laboratory,
University of Cambridge)

circa 10 centesimi di dollaro.[29] Le cariche sullo schermo si dissipano rapidamente e deve essere prevista una costante azione di ripristino delle stesse (*refresh* di memoria); la messa a punto del sistema era molto critica ed estremamente sensibile ai disturbi elettromagnetici dell'ambiente. Il primo uso di questa tecnologia fu quello adottato da Freddie Williams e Tom Kilburn per il Manchester Mark I (UK), da cui l'eponimo 'Tubo di Williams-Kilburn'. Il computer dello IAS a Princeton fu dotato di 40 tubi di Williams-Kilburn per una capacità totale di 40.096 bit.

Un diverso tipo di memoria elettrostatica fu quello progettato da Jan A. Rajchman alla RCA, battezzato *Selectron*. Il dispositivo, in forma di una grossa valvola termoionica, conteneva numerose cellette (da 256 a 4096) ognuna delle quali poteva registrare un bit caricando un condensatore tramite un fascio di elettroni. Un opportuno sistema di griglie di controllo permetteva di indirizzare il singolo bit, ottenendo quindi una memoria ad accesso casuale. Il dispositivo era di difficile costruzione e costoso, una memoria a *Selectron* di soli 5120 bit, costruita per lo *Air Force Cambridge Research Center*, era costata 750.000 dollari, pari a 145 dollari/bit, cifra difficilmente sostenibile per altri progetti. Il *Selectron* fu adottato solo per il computer JOHNNIAC della RAND, una copia della macchina di von Neumann allo IAS.



Memoria elettrostatica a tubo di Williams-Wilkes del Manchester Baby. (Computer History Museum, Mountain View, CA, USA)

Linee di ritardo e tubi di Williams-Kilburn erano comunque sempre troppo costosi se si voleva disporre di memorie con capacità maggiori di poche migliaia di bit. Una soluzione fu quella di ricorrere ad una memoria ausiliaria meno veloce, sequenziale, ma molto meno costosa (0,2 centesimi/bit), il tamburo magnetico. La tecnologia della registrazione magnetica, inventata nel 1932 dall'austriaco Gustav Tauschek, faceva uso di un cilindro rotante ricoperto da uno strato ferromagnetico su cui erano appoggiate testine che registravano e leggevano i dati digitali lungo piste parallele. L'idea di utilizzare dischi o tamburi magnetici era stata proposta indipendentemente da

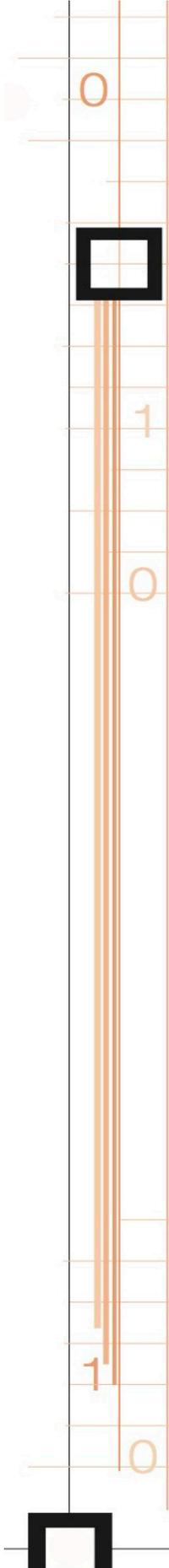
Presper Eckert e da Perry Crawford (MIT) nel 1944, ma la tecnologia non era ancora stata perfezionata ed era più lenta delle linee di ritardo; venne riproposta come memoria secondaria, dove si sarebbero immagazzinati dati ed istruzioni di impiego non immediato, caricandole nella memoria veloce principale solo quando necessari (*swapping*). I primi tamburi magnetici furono sviluppati negli USA alla *Engineering Research Associates* (ERA) alla fine degli anni 1940.

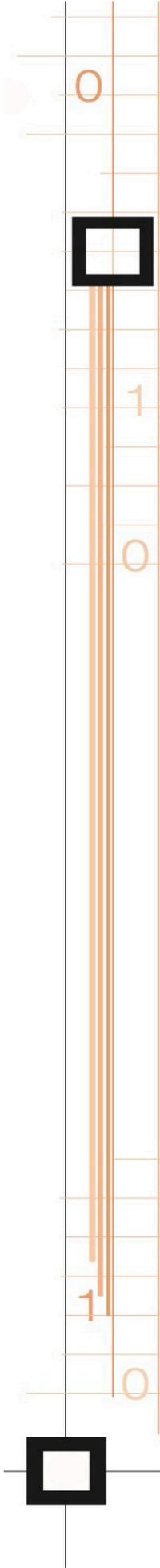
Numerose altre tecnologie furono tentate nei primi anni, cercando di sfruttare i più diversi fenomeni fisici, dalla fotografia alla magnetostriazione, alla fluorescenza, alla ferroelettricità, spesso con risultati deludenti. Per disporre di memorie ad accesso casuale relativamente economiche e più affidabili, bisognerà attendere la tecnologia dei nuclei magnetici, proposti per la prima volta da Presper Eckert nel 1945, su cui, nella seconda metà degli anni '40, lavoravano contemporaneamente Jay Forrester al MIT, Jan Rajchman alla RCA, An Wang ad Harvard, e Mike Haynes all'IBM. La memoria a nuclei magnetici divenne di uso corrente solo nella seconda metà degli anni 1950 e fu soppiantata dai circuiti integrati a semiconduttore nei primi anni '70. Fu quest'ultima tecnologia che permise di diminuire in modo impressionante costo, dimensioni e consumo energetico della memoria, aumentandone al contempo la velocità di accesso, fino alla situazione odierna in cui il costo è sceso a 15 – 20 centesimi di dollaro per miliardo di bit.



Bibliografia

1. Goldstine H. H., *The Computer from Pascal to von Neumann*, Princeton University Press, 1972 .
2. H. Rheingold, *Tools for Thought*, MIT Press, 2000.
3. Eckert J.P. Jr., *Automatic High Speed Computing: A report on the EDVAC*, 30 settembre 1945,
http://archive.computerhistory.org/resources/text/Knuth_Don_X4100/PDF_index/k-8-pdf/k-8-u2736-Report-EDVAC.pdf(visto 5 dicembre 2012)
4. Copeland B. J. (a cura di), *Alan Turing's Automatic Computing Engine*, Oxford University Press, 2005.
5. *ENIAC Patent Trial Collection, 1964 – 1973*, http://www.archives.upenn.edu/faids/upd/eniactrial/upd8_10.html. (visto 2 dicembre 2012)
6. Burks A. R., *Who invented the computer? The Legal Battle that Changed Computing History*, Prometheus Books, 2003.
7. Tomayko J.E., "Helmuth Hoelzer's Fully Electronic Analog Computer", *Ann. Hist. Comp.* **7** (3), 1985, pp. 227-240.
8. Irvine M.M., "Early Digital Computers at Bell Telephones Laboratories", *IEEE Ann. Hist. Comp.*, **23** (3), 2001, pp. 22-42.
9. Cohen J.B., Welch G.W., *Makin' Numbers. Howard Aiken and the Computer*, MIT Press, 1999.
10. Rojas R., "The Architecture of Konrad Zuse's Early Computing Machines", in: Rojas R., Hashagen U. (a cura di), *The First Computers. History and Architecture*, MIT Press, 2002, pp. 237-261
11. Ceruzzi P., *Reckoners. The Prehistory of the Digital Computer, from Relays to the Stored Program Concept, 1935-1945*, Greenwood Press, 1983.
12. Metropolis N., Howlett J., Rota G.-C. (a cura di), *A History of Computing in the Twentieth Century*, Academic Press, 1980.
13. Copeland B.J. (a cura di), *Colossus. The Secrets of Bletchley Park's Codebreaking Computer*, Oxford University Press, 2006.
14. Anderson D., "Was the Manchester Baby conceived at Bletchley Park?", In: *Turing 2004: A celebration of his life and achievements*, University of Manchester, 5 giugno 2004,
http://www.bcs.org/upload/pdf/ewic_tur04_paper3.pdf, (visto 2 dicembre 2012).
15. Eckert J.P. Jr, *Disclosure of Magnetic Calculating Machine* , 29 febbraio 1944.
http://archive.computerhistory.org/resources/text/Knuth_Don_X4100/PDF_index/k-8-pdf/k-8-u2775-Mauchly-letter-plus.pdf. (visto 5 dicembre 2012)
16. Heims S.J., *John von Neumann and Norbert Wiener*, MIT Press 1987.
17. Aspray W., *John von Neumann and the Origins of Modern Computing*, MIT Press, 1992.
18. Riley H.N., *The von Neumann Architecture of Computer Systems*, <http://www.csupomona.edu/~hnriley/www/VonN.html> (visto 5 dicembre 2012)
19. Godfrey M.D. (a cura di), "First Draft of a Report on the EDVAC", *IEEE Annals of the History of Computing*, **15** (4), 1993, pp.27-75,
20. Godfrey M.D., Hendry D.F., "The computer as von Neumann Planned it", *IEEE Ann. Hist. Comp.*, **15** (1), 1993, pp. 11-21.
21. Sugimoto M., "On the original copies of First Draft of a Report on the EDVAC by John von Neumann", *PHS Studies* (Tokio), No. 6, 2012, pp. 83-89.
22. Ortway R., *The study of the Brain as a Switching System*, citato in:[17].
23. McCulloch W.S., Pitts W., "A Logical Calculus of the Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics* Vol 5, pp 115–133.
24. Hyman A., *Charles Babbage. Pioneer of the Computer*, Princeton University Press, 1982.

- 
- 
25. Collier B., *The Little Engines that Could've: The Calculating Machines of Charles Babbage*, Tesi di Dottorato (Ph.D.), Department of History of Science, Harvard University, 1970.
<http://robroy.dyndns.info/collier/index.html> (visto 5 dicembre 2012).
 26. Shannon, C., "A Symbolic Analysis of Relay and Switching Circuits", *Trans. Am. Inst. Electrical Engineers*, 57, 1938, p. 713-723
 27. Stern N., *From ENIAC to UNIVAC. An Appraisal of the Eckert-Mauchly Computers*, Digital Press, 1981.
 28. Williams M.R., "The Origins, Uses and Fates of the EDVAC", *IEEE Ann. Hist. Comp.*, 15(1), 1993, pp. 22-38.
 29. Campbell-Kelly M., Aspray W.F. (a cura di), *The Moore School Lectures. Theory and Techniques for Design of Electronic Digital Computers*, MIT Press, Tomash Publishers, 1985.
 30. Aspray W. F. (a cura di), *Proceedings of a Symposium on Large-Scale Digital Computing Machinery*, MIT Press, Tomash Publishers, 1985.
 31. Burks A.W., Goldstine H.H., Neumann J., "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument", 1946; in: A. H. Taub (a cura di), *Collected Works of John von Neumann*, vol. 5, pp. 34-79, The Macmillan Company, New York, 1963.
 32. Goldstine H.H., von Neumann J., "On the principles of Large-Scale Computing Machines", 1946, in: Aspray W., Burks A., *Papers of John von Neumann on Computing and Computer Theory*, MIT Press, 1987, pp. 317-348.
 33. Lavington S.H., *Early British Computers*, Butterworth-Heinemann, 1980.
 34. McCann, D., Thorne P., *The Last of The First, CSIRAC: Australia's First Computer*, University of Melbourne Computing Science, 2000.
 35. Redmond K.C., Smith T.M., *Project Whirlwind*, Digital Press, 1980.
 36. Bell C.G., Newell A., *Computer Structures, Readings and Examples*, McGraw-Hill 1971.
 37. Hodges A., *Storia di un enigma. Vita di Alan Turing, 1912-1954*, Bollati Boringheri, 1991.
 38. Turing S., *Alan M. Turing*, Cambridge University Press, 2012.
 39. Pelaez E., "The Stored Program Computer: Two Conceptions", *Social Studies of Science*, 29(3), 1999, pp. 359-390.
 40. Copeland B.J., *The Essential Turing*, Clarendon Press, 2004.
 41. Randell B., "On Alan Turing and the Origins of Digital Computers", *Machine Intelligence* 7, 1972, pp. 3-22,
<http://www.cs.ncl.ac.uk/publications/books/papers/126.pdf> (visto 5 dicembre 2012).
 42. Randell, B. (a cura di) *The Origins of Digital Computers: Selected Papers*, Springer-Verlag, 1982.
 43. Rojas R. , Hashagen U. (a cura di), *The First Computers - History and Architectures*, MIT Press, 1999.
 44. Zuse, Konrad. *The Computer - My Life*, Springer-Verlag, 1993.
 45. Mahoney M.F., "Computer Science. The Search for a Mathematical Theory", in: J. Echeverria, A. Ibarra, T. Mormann (a cura di), *The Space of Mathematics* , De Gruyter, 1992.
 46. Priestley M., *A Science of Operations. Machines, Logic and the Invention of Programming*, Springer, 2011.
 47. C. Petzold, *The Annotated Turing*, Wiley, 2008.
 48. R. Rojas, "How to make Zuse's Z3 a Universal Computer", *IEEE Ann. Hist. Comp.*, 20, (3), 1998, pp. 51-54.
 49. Turing A., "Computing Machinery and Intelligence", *Mind*, 49, 1950, pp. 433-460.
 50. Wilkes M.V., *Automatic Digital Computers*, John Wiley & Sons, 1956.

- 
- 
51. P. Ceruzzi, "Nothing New Since von Neumann", in : Rojas R., Hashagen U. (a cura di), *The First Computers. History and Architecture*, MIT Press, 2002
 52. Hodges A., "Alan Turing and the Turing Machine ", in: Herken R., *The Universal Turing Machine. A Half-Century Survey*, Springer, 1994.
 53. Crowe G.F., Goodman S.E., "S.A. Lebedev and the Birth of Soviet Computing", *IEEE Ann. Hist. Comp.* , 16(1), 1994, pp. 4-24.
 54. Winston B., *Media Technology and Society*, Routledge, 2002.
 55. Akera A., *Calculating a natural World. Scientists, Engineers and Computers During the Rise of U.S. Cold War Research*, MIT Press, 2007.

Biografia

Silvio Hénin è uno studioso di storia della tecnologia, in particolare del calcolo automatico. Consulente del Museo Nazionale di Scienza e della Tecnologia 'Leonardo da Vinci' di Milano, collabora con i periodici *Mondo Digitale*, *Le Scienze*, *IEEE Annals of the History of Computing* e ha curato numerose voci del *Dizionario Enciclopedico di Informatica, ICT e Media Digitali* edito da Enciclopedia Treccani. S.H. è socio di AICA (Associazione Italiana per l'Informatica ed il Calcolo Automatico) e di ACM (Association of Computer Manufacturers).

Email: silvio.henin@fastwebnet.it