フォン・ノイマンのEDVAC草稿と二進法

小山俊士　（早稲田大学人間科学学術院）

1　はじめに

本稿は、数学者フォン・ノイマン（1903-1957）が初期コンピュータ開発に携わった際に執筆した文書を分析し、その数学的な側面を理解することを目的としている。これに関しては、現在作業を進めている『コンピュータ理論の起源　第2巻　フォン・ノイマン』（近代科学社）において、『EDVACに関する報告第一草稿』（以下『EDVAC草稿』）、『大規模計算装置の原理について』、『電子計算装置の論理設計に関する予備的議論』の3つの文書の翻訳し、解説を加える予定であるのだが、そのうちの最初の文書に対する解説から抽出し、大幅に加筆した。フォン・ノイマンの文書はコンピューティングの概念を包括的に論じたものであって、技術史の側面からの分析は少なくないのだが[1]、今回は数学的な面に絞って論じていきたい。その中でも二進数の利用に注目し、デジタル装置との相性の良いという理由だけではなく、真空管を使用するという制約条件のもと、二進数の持つ数学的な性質を利用してより単純な回路を作るという意図で採用されたことを明らかにしていく。

---

[1]主な先行研究を上げておく。『EDVAC草稿』の書かれた経緯については、ハーマン・H. ゴールドスタイン『計算機の歴史—パスカルからノイマンまで』末包良太・米口肇・犬伏茂之訳（共立出版、1979年）、Nancy B.Stern, *From ENIAC to UNIVAC,An Appraisal of the Eckert-Mauchly Computers*, Bedford, Massachusetts: Digital Equipment,1981、ウィリアム・アスプレイ『ノイマンとコンピュータの起源』杉山滋郎・吉田晴代訳(産業図書、1995年)、ジョージ・ダイソン『チューリングの大聖堂』吉田三知世訳（早川書房、2013年）等が詳しい。草稿の内容を論じたものとしてはこれらに加えて、W. Aspray and A.Burks,eds., *Papers of John von Neumann on Computing and Computer Theory*, Cambridge Mass.-London: The MIT Press,1987 の3-16ページの Auther Burks の解説、星野力『誰がどうやってコンピュータを創ったのか？』共立出版、1995年、Eloína Peláez, “The Stored-Program Computer: Two Conceptions,” *Social Studies of Science*, Vol.29,No.3, 1999, pp.359-89 などがある。

2 背景

今回取り上げる『EDVAC草稿』は、ペンシルバニア大学電気工学部ムーアスクールにおける1945年6月30日付けの100ページあまりの文書であるが、それに先立って、ムーアスクールではENIAC(Electronic Numerical Integgrator And Computer)の開発が行われていた。ENIACは物理学者モークリーが提案し、陸軍の支援のもと、技術者エッカートが指揮して開発された真空管を用いた高速汎用計算装置であり、公表されたのは1946年2月14日だが、1945年秋頃に稼働していた。ENIACの仕様は何度かの改造で変化しているが、1955年の調査への回答[2]では、真空管17,468本、消費電力174kW、基本システムの費用約$750,000という巨大な装置だった。真空管を素子として用いた高速累算器を20器持っていたことがENIACの特徴なのだが、そのために大きな消費電力、多額の開発費用を要した。真空管は寿命が数千時間と短く、膨大な本数を用いたENIACは、絶えず故障した真空管を交換しながらでなければ動作しなかった。また、処理したい問題に合わせて、累算器の設定スイッチとその間でデータをやりとりするケーブルの配線を変更することで汎用性を持たせていたのだが、その変更のための作業は複雑で時間のかかるものだった。

ムーアスクールではこうしたENIACの欠点を改良した次世代機の検討も行っていたが、その最大の特徴は高速記憶装置（メモリ）を加えることで、累算器を減らして真空管数を削減し、処理の手順をメモリに入れて変更を容易にする点だった。フォン・ノイマンが初めてムーアスクールを訪れた1944年夏には、ENIACは設計を終えて製作中であり完成を待つだけだったが、EDVAC(Electronic Discrete Variable Automatic Computer)と命名された次世代機の検討が行われていたので、その議論へと加わった。

3 『EDVAC草稿』執筆の経緯

フォン・ノイマンは第二次世界大戦中は軍の様々な機関の顧問を務め、兵器開発の際に必要とされる数学の問題を解いていた。特に衝撃波の専門家となっており、マンハッタン計画では爆縮法によって起爆されるプルトニウム爆弾の設計を担当していた。微分方程式を近似的に解くため

[2]Martin H. Weik, *A Survey of Domestic Electronic Digital Computing Systems, BALLISTIC RESEARCH LABORATORIES. REPORT NO. 971*, Aberdeen Proving Ground., Md. , December 1955 (http://ed-thelen.org/comp-hist/BRL.html の index より 、2015年1月31日閲覧).

には膨大な数値計算が必要で、ロスアラモスではIBMのパンチカード装置を使っていたのだが、より高速で汎用性の高い装置を求めて、利用可能な様々な計算装置を調査してした。当時は、継電器（リレー）を用いたコンピュータも作られていたが、それでは計算速度が遅いと判断した。その中で、ムーアスクールで開発中のENIACに注目し、1944年夏からプロジェクトに参加したのだった。1945年秋からは水爆が可能かを確かめるシミュレーションを行うなど、稼働したENIACを様々な軍事研究に利用した。

ムーアスクールではEDVACの検討を行っていたが、フォン・ノイマンが定期的に訪問するようになると、それに合わせて会合を開き、アイデアを検討するようになった。ここで取り上げる『EDVAC草稿』は、その過程でフォン・ノイマンが議論を整理するために執筆したものだった。

『EDVAC草稿』の執筆者はフォン・ノイマンだが、それを陸軍に所属してプロジェクトに参加していた数学者ゴールドスタインが指示して秘書にタイプ打ちさせた。本来はムーアスクールの内部文書だったのだが、複製が作られアメリカ、イギリスの研究者に配布された[3]。配布先で回覧されたりさらに複製が作られたこともあって、その後のコンピュータの発展に大きな影響を与えたとされる。

4 『EDVAC草稿』の意義

ムーアスクールでのENIACおよびEDVACの開発は軍の支援を受けたプロジェクトであり、第二次世界大戦中は秘密事項だった。アメリカや同盟国のイギリスで計算装置の研究開発を行っていた科学者、技術者のグループはいくつもあったが、ムーアスクール活動について詳しい情報を得ていたものはほとんどいなかったと思われる。『EDVAC草稿』が外部に情報を伝える最初の文書となった。

『EDVAC草稿』は初期コンピュータの歴史において必ずといっていいほどよく言及される文書であり、特に、メモリにプログラムを保存してコンピュータを制御するという「プログラム内蔵」のアイデアを最初に提示したものとされる。さらには、コンピュータ内部での二進法の使用、演算と制御・記憶・入出力という基本的な構成、そして演算装置は記憶装置内の命令を一つずつ実行していく逐次処理方式を合わせて「ノイマ

---

[3]Nancy Stern, *From ENIAC to UNIVAC* によれば原本の送付リストには32名が記されているという。

ン型アーキテクチャ」と呼び、今日のほとんどすべてのコンピュータで使われているのだが、この方式の起源もまたこの『EDVAC草稿』だといわれることがある。実際の歴史を見ていくなら、「ノイマン型アーキテクチャ」がすべてこの『EDVAC草稿』から始まるとすることには無理があり、それをフォン・ノイマンが発明したとするのは誤りである。「ノイマン型アーキテクチャ」に含まれる個々のアイデアに関しては、ムーアスクールの議論の中で提示されたものであり、フォン・ノイマンはそれを整理したに過ぎない。また、後で論じるように『EDVAC草稿』は概念的な検討結果を提示するにとどまっており、この後に出されたムーアスクールの文書や1946年夏にアメリカとイギリスの研究者を招いて開催されたサマースクール[4]で詳細な技術情報が伝えられたことによって、各地の研究者はそれにもとづくコンピュータ開発を始めることができた。フォン・ノイマン自身もその後プリンストン高等研究所でコンピュータ開発を開始し、そこで生まれたアイデアを文書で公開したので、これらの情報が合わさり、1940年代後半にその後のコンピュータの基本となる概念がが形成されていったのだと見るべきであろう。

以上のような経過を考慮に入れつつ、ここではその後の発展とは切り離して、『EDVAC草稿』そのものに表れているフォン・ノイマンの発想を分析することを目的とする。文章中に登場する様々なアイデアは必ずしもフォン・ノイマンによる発明ではないのだが、フォン・ノイマンはそれらにどんな意義を与え、一貫した方針で装置を構成しようとしていたのかという点に注目していきたい。

今回、参照した原典はペンシルバニア大学に保管されているタイプ原稿である[5]。正式に出版されなかった未完成の原稿であり、タイプ・ミスなどの誤りも多数あるため、それらを修正した復刻版がいくつか普及しているのだが[6]、これらの版にも脱落等があるため注意を要する。ここでは原典とそれらの復刻版を参照し、誤りを修正して内容を正確に再現することを意図している。

---

[4]この講義録は、Martin Campbell-Kelly and Michael R.Williams,eds., *The Moore School Lectures—Theory and Techniques for Design of Electronic Digital Computers*, Cambridge,Massachusetts- London: The MIT Press,1985に復刻されている。

[5]インターネットで閲覧できるものとしては、スミソニアン・ライブラリ（http://library.si.edu/digital-library/book/firstdraftofrepo00vonn、2015年1月31日閲覧）に保存されている原稿がある。

[6]例えば、Nancy Stern, *From ENIAC to UNIVAC*に収録されたものや、Michael D. Godfreyが作成し*IEEE Annals of the History of Computing*, vol.15, no.1, 1993に掲載されたもの（http://www.virtualtravelog.net/wp/wp-content/media/2003-08-TheFirstDraft.pdfから入手できる、2015年1月31日閲覧）がある。

5　『EDVAC 草稿』の構成

ここから『EDVAC 草稿』の内容の分析に入る。まず、文章全体の構成を簡単に紹介するが、本文には **1.1** といった章節の番号が付されているので、その章を内容に即してまとめると、

1. **1.1-3.3**　総論
2. **4.1-6.5**　基本となる素子の説明
3. **7.1-11.4**　算術演算のための回路の構成法
4. **12.1-13.5**　記憶の機能の説明
5. **14.1-15.6**　備えるべき命令の一覧

のようになる。

最初に指摘しておきたいのは、この草稿ではきわめて一般性の高い議論を展開しようとしていることである。EDVAC の計画で使用する予定の技術や方式を解説するのではなく、コンピュータに要求される機能は何か、それに対して可能な選択肢として何があるのかを示して、それにもとづいて具体的な設計の指針を示すといった書き方である。フォン・ノイマンはこの草稿において、タイトルである「EDVAC」について報告するのにとどめず、未来の技術の進歩も視野に入れた一般的なコンピュータについて論じているともいえる。

この配列には雑然とした印象を受けるし、著者自身も **3.2** で、「ジグザグ」した論じ方をすると述べているのだが、これが基本となるものがまったくない状態で始めて書かれた文書であるということを念頭に置かねばならない。総論の **2.1-9** では装置の構成する要素の一覧、中央算術部（CA)、中央制御部（CC)、記憶（M)、入出力（I と O）と外部記録媒体（R）を示し、**4.1-3** で素子の概略を行った後に、**7.1-11.4** で四則演算や開平計算を実現する方法を具体的な回路を示しながら論じ、その考察にもとづいて **12.1** 以降で再び各要素に戻って、それらに求められる機能を決めていくという順序である。

通常の教科書であれば、構成要素の機能を順番に説明していき、その後に具体的な使い方として計算方法を示していくという配列にしそうなものだが、それとは逆の論じ方になっている。それぞれの要素がなぜ必要か、なぜその性質にするのかを、具体的な問題を想定しながら、論理

的に述べていく書き方をしているのである。こういったところに、数学者としてのフォン・ノイマンの個性が表れているといっていいであろう。

6 プログラム内蔵

『EDVAC草稿』にはいくつかの意外なところがある。例えば、タイトルにある「EDVAC」という装置は、本文中に登場せず、ただコンピューティング・システムとか装置と呼ばれるだけである。ムーアスクールのプロジェクト名としてEDVACはタイトルに残っているものの、本文で行っているのは一般的なコンピュータの考察であることを意味しているといえるかもしれない。また、この文書は世界で初めてプログラム内蔵型コンピュータの概念を明らかにしたものとして有名なのに、本文中に「プログラム」という語も登場しない。今日、広く流布している『EDVAC草稿』についてイメージは、これより後に出された文書で紹介される概念まで混ざったものとなっているのだが、ここではそうした先入観を排して実際の草稿の記述を分析していきたい。

この『EDVAC草稿』の提唱するシステムは次のようなものである。冒頭**1.1**で扱う対象について、「考察の対象は超高速自動デジタル計算システムで、特に論理制御のものの構造」であるという定義が与えられる。さらに「自動計算システムとは、かなりの程度の複雑さをもつ計算をせよという指令を遂行できる装置」であると補足されているが、この「指令(instruction)」または「命令 (order) 」が後のプログラムに相当する。装置はいくつかの演算（四則演算など）機能を備えており、与えられた指令にしたがって「演算の適切な系列」を遂行できる。

コンピュータは、一つの計算を行うだけなく、順序にしたがって多数の演算を行う能力を持つことによって複雑な問題を処理する汎用性を持った装置になる。先行するENIACでは配線とスイッチの設定を変えることで汎用性を与えたのだが、EDVACではその手順を記憶に保存し、自動的に読み出して遂行していくものとするという提案が述べられているのが、**2.4**の記憶の必要性を説明したところである。記憶装置MはEDVACにおいて新たに加えられた構成要素だが、なぜ必要になるのかを(a)から(h)に分類して具体例も交えて説明している。「プログラム内蔵」に相当するのは(b)の「複雑な問題を統制する指令はかなりの量の素材からなるだろうし、... この素材も記憶されねばならない」と述べた箇所で、コンピュータ史を扱う書籍で『EDVAC草稿』に言及したり、抜粋引用する場合のほ

とんどは、この箇所に注目する[7]。しかし、『EDVAC草稿』の中では注意して読まないと見落としてしまいそうな短い記述であり、この「指令」についての説明も与えられていない。フォン・ノイマンが「プログラム」について具体的な考察を示すのは、その後、プリンストン高等研究所に戻って独自のコンピュータ開発プロジェクトを推進する過程で出された文書においてであった[8]。

『EDVAC草稿』の段階での記憶の説明としては、「指令」を保存することよりも、計算過程で一時的に数値を保存することの方に重点が置かれていた。すなわち(a)と(c)から(g)では、解くべき数学の問題によって分類し、それぞれの過程で数値を保存する必要性があることを指摘している。(a)は四則演算、(c)から(g)が各種の微分方程式の近似的解法、(h)が統計や整列（ソート）問題であり、これを執筆した時点でのフォン・ノイマンの関心が衝撃波の計算などで登場する微分方程式に向かっていたことを反映している。(a)については、この後、四則演算のための回路を構成する過程で説明を補う。最後の(h)のような数値計算以外の用途については、少し後にフォン・ノイマンは具体的な検討を行ったことのみを紹介するのにとどめる[9]。

7　算術を統制する原理と二進法

『EDVAC草稿』では、算術を行う回路の構成法を論じる前に、使われる素子と構成のための原理を提示する。フォン・ノイマンは文中で提示される新たな概念を表現するために、生物、特に脳とのアナロジーにもとづく用語を使用している。今日であれば「メモリ」といった言葉は、元々の意味を意識することなく、最初からコンピュータ用語だったと考えてもおかしくない使われ方をしているが、この草稿の時点ではあくまで脳の機能を示す用語を借用したものだった。回路の基本となる部品については、**4.1-3**で生物とのアナロジーを用いて脳細胞をイメージした素子を

---

[7]例えば、月尾嘉男・浜野保樹・武邑光裕『原典メディア環境1851-2000』東京大学出版会2001年、277-279ページ。

[8]"Planning and Coding Problems for an Electronic Computing Instrument," https://ia600506.us.archive.org/11/items/planningcodingof0103inst/planningcodingof0103inst.pdf（高等研究所のアーカイブ、2015年1月31日閲覧）。

[9]Knuth(1970) Donald E.Knuth, "Von Neumann's First Computer Program," *Computing Surveys*, Vol.2,No.4, 1970, pp.247-260、（日本語訳）「Von Neumannの最初のコンピュータ・プログラム(1)」山田真市訳『Bit』Vol.5、No.6、1973年、435-440ページ、「同(2)」Vol.5、No.8、1973年、984-990ページ。

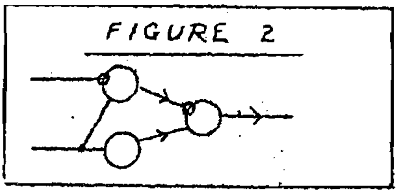

図2 E素子

使用することを説明し、**6.2**で「E素子」として定義される。E素子は2つの状態、すなわち静止と興奮を持ち、繋がれた他の素子からの刺激の入力によって、興奮状態に遷移するという特徴を持つ。E素子は**図**2[10]のようなブロック記号で表され、左側から刺激が入力され、興奮状態になると右側へ刺激を出力するのである。この「E素子」は『EDVAC草稿』のみで使われる用語である。

E素子の状態遷移に要する時間が遅延であってτと表現し、**図**2の線に付された矢印（→）がその左側のE素子の遅延を表している。すべてのE素子の遅延が同時に共通の長さで生じ、刺激の伝播には時間を要しないとすることで、装置全体を同期化させて制御することが容易になる。装置全体を同期化させる方法は、ENIACから採用されていたものだった。真空管を使ったE素子の遅延は、マイクロ秒（10^{-6}秒）という非常に短い時間であり、そのおかげで高速の処理が可能となるのである。ここで使われる素子の図は、神経細胞の機能を簡略化して考察したマカロックとピッツの論文[11]の記法を参考にした独自のものである。フォン・ノイマンはその後のオートマタ理論の記述でもこの記法を使い続けたが、コンピュータの設計においてはその後、使われなくなった。

5.1から二進法を採用する理由を述べていくのだが、その冒頭で扱う素子は小さな変動を切り捨てて「全または無」の二つの状態のみを持つ（すなわちオール・オア・ナッシングの、今日の言葉でいうならデジタルの）装置として使われるべきであることから、機械内部では二進法を用いることが示唆されるとする。現実の真空管は定格通りに機能するとは限らず、設計の際には使用時に電圧が多少変動しても誤動作しないよう余裕を持たせるといった配慮も必要なのだが、その際にデジタル方式を採ることが意味を持つ。

デジタル・コンピュータにおいて二進法体系を使用するという試みは、

---

[10]『EDVAC草稿』には22枚の図が使われている。ここでは必要な図のみを引用するが、図の番号は原典のものである。

[11]W.Pitts and W.S.MacCulloch,“A logical caluculus of the ideas immanent in nervous activity,”*Bull.Math.Biophysics*,Vol.5(1943),pp.115-133。

ENIACの前に真空管を用いた計算機としてアイオワ大学のアタナソフとベリーによって計画されていたABC計算機においても検討されていた。また、リレー回路とブール代数の類似性をシャノンが指摘していた。ここでEDVACに二進法を採用する提案はそのような流れを承けたものではあるが、『EDVAC草稿』では**6.1**でE素子を検討するのに、理想的な手続として素子の電磁的な性質は後回しにして「そうあるべきと意図されるものであるとして取り扱う」と宣言して、真空管回路を実際に運用する際の問題は考察対象から外す。そして、論理的な性質の考察に専念し、そこで二進法の使用が推奨されるとするのである。

ここでは対比のために、十進法を用いたENIACではどのように数を表現していたのかを紹介しよう[12]。ENIACでは真空管をオン・オフ装置として、すなわち電流を通すか通さないかによって情報を表現する装置として使った。三極真空管（陽極－陰極間の電子流をグリッドで制御する真空管）では、グリッドに正電圧を加えると陽極－陰極間に電流が流れるオン（導通状態)、負電圧を加えたときは流れないオフ（不通状態）となる。記憶のためにはフリップ・フロップ回路を用いるが、これは2本の真空管を使って構成され、一方のみが導通状態で、他方が不通状態になるように設計されており、外部からの入力で状態を切り換えることができる（この2状態をA、Bと呼ぶ)。ENIACでは、フリップ・フロップ回路をネオンランプへ接続して、状態Aではランプは消え、状態Bでは点灯することで、操作する人が内部の状態を知りうるようにした。

ENIAC内部での数は、10個のフリップ・フロップ回路を円状つなげたリング・カウンターによって、1個の十進数を表現した。リング・カウンターは、10個のフリップ・フロップ回路のうちの1個だけが状態Bで、残り9個は状態Aになるように設計されている。どのフリップ・フロップ回路がB状態になっているかによって、0から9までの数字を表現するという下表（状態Aを0、Bを1としている）のような方法を用いた。

[12]ここに示した議論や数値については、Anton Glaser, *History of Binary and other Non-decimal Numeration*, Tomash Publishers,1971を参照した。

表現したい数	各フリップ・フロップ回路の状態	
0	00000	00001
1	00000	00010
2	00000	00100
3	00000	01000
4	00000	10000
5	00001	00000
6	00010	00000
7	00100	00000
8	01000	00000
9	10000	00000

リング・カウンターは外部から1個のパルスを受信すると、状態BがAに変わり、その左隣がBに変わる。左端のフリップ・フロップ回路が状態BからAに変わるとき、円状につながった右端がBになるだけでなく、パルスを1個発信することで、繰り上がりを表現できるようになっている。

このようなリング・カウンターを10個並べ、符号を表示するカウンターを加えて、10桁十進数を表す累算器を構成した。累算器の間を結ぶ配線とパルスの伝達を制御するスイッチの設定によって、様々なプログラムでの数値の処理を可能にしたのであった。累算器を計20器設置したが、そのために使用される真空管の数が膨大になり、ENIACの維持管理は困難になった。ここで提案されているEDVACでは、真空管の数を減らすためにフリップ・フロップ回路で作られる記憶装置を少なくし、その代わりに他の記憶装置を利用するべきだとするのだが、もう一つ真空管を節約する方法として、装置内部での数表現を二進数にすることも提案した。例えば、数1984をENIACの方法で表現すると、

$$1984 = 00000\ 00010,\ 10000\ 00000,\ 01000\ 00000,\ 00000\ 10000$$

となるが、各桁ごとに二進数に置き換えるなら、4桁で0から9を表せるので、

$$1984 = 0001\ \ 1001\ \ 1000\ \ 0100$$

となる。さらに1984そのものを二進数に置き換えるなら

1984 = 11111000000

ENIACの方法なら40個のフリップ・フロップ回路が必要なのに対して、各桁ごとなら16個、全体を二進数にするなら11個でよいことになる。

ただし、内部で二進数を使うことの不利な点としては、ENIACのようにフリップ・フロップ回路をネオンランプにつなげて、直接に計算結果を十進数で表示できなくなることがある。だが、ENIACのように内部の状態を目で確認できるようにすることは一見便利なようだが、実際には高速で進行する処理を人間が目で追うことはできないので、結果の出力以外には意味がない。それゆえに装置は自動計算システムなのであって、内部を循環するデータを人が知覚する必要はなく、最終の結果のみを十進法で出力するようにすればいい。二進数を十進数に変換する回路が必要となるのだが、その変換回路による真空管数の増加は数表現において節約される真空管数を上回ることはないだろうという推測のもと、二進数を利用を提案しているのである。

二進法の採用を示唆した後、より一般的な考察として**5.6**で「装置は可能な限り単純であるべきだ、すなわちできるだけ少ない数の素子からなるべきだ」ということを算術演算を統制する原理とすべきだと提唱している。これは『EDVAC草稿』内で、繰り返し「5.6の原理」と言及されており、草稿全体を貫く最も重要な原理だった。1950年代まで主要な素子として使用された真空管の故障頻度の高さは、装置を開発し利用する際の厳しい制約だった。EDVACの記憶装置はENIACの累算器に使われていた真空管を削ることを最大の目的としており、「プログラム内蔵」の実現は、そのための手段であったといってもよい。現在のコンピュータはENIACやEDVACとは桁違いに多くの素子を使用しているのだが、それは1960年代に半導体による集積回路が使えるようになったおかげである。以後のコンピュータ設計においてこの「5.6の原理」は破棄され、代わりに「ムーアの法則」にしたがって素子の数を増やし、より複雑なハードウェアを作る方向での発展が続いている。

この原理の補足として、装置を単純化するという原理が有効な理由、すなわち単純な装置でも十分な機能を発揮できる理由は、真空管を用いた装置が持つ高速性にあると述べられている。対比として、真空管以前の素子は反応時間が長く、卓上計算機なら乗数あたり10秒、IBM乗算機で6秒かかるという数値が挙げられている。低速の装置で複雑な計算を実行するには、全体での計算時間を短縮する工夫が求められる。そのために、

例えば加算では各桁ごとに足し合わせる計算を同時並行で実行し、その後に繰り上がりの処理をするといった手順が考えられるが、それは素子の数を増やして装置を複雑することを意味しているのである。その一方で、真空管を使った素子を用いた装置では、1つの回路で順番に各桁を足し合わせていっても、1つ1つの計算が速いので、短時間で全体の処理を終えることができる。そこで、計算時間を短縮することより真空管の不安定さを軽減することの方が重要な課題であり、そのためにこの原理を採用するべきだとする。二進法を用いる利点もまた、装置を単純化できることなのであるが、それは次節で説明する。対比を図式化して整理すると、次のようになる。

従来の素子	→遅い	→手順の短縮	→十進法
真空管	→速い	→短縮の必要なし	→二進法
	→信頼性が低い	→本数を減らす	→二進法

8　加算回路

中央算術部CA内に様々な演算を実現するための回路を作るわけだが、**2.2**で特化した回路をどこまで作るべきかという問題提起をしている。半導体を用いるようになった今日では、CPU内に複雑な回路を多数作り込むのが当然のことになっているのだが、真空管の時代には現在とは異なる原理で装置が設計されていた点に注意しなければ、『EDVAC草稿』の意義を理解できない。フォン・ノイマンは真空管の本数を減らすということに強い関心を示し、CAをギリギリまで単純化するための検討を行った。

主要な演算（+, ×, −, ÷, √など）を実現する具体的な方法については、**7.1**から**10.4**で論じているのだが、特定の演算用の回路を示すのではなく、除算回路と同じ部品を使って開平計算もできるといったように、共通の部品を使用して複数の演算を行わせて装置全体を単純化するための考察をしていく。

まず**7.1**で、CA内で標準的な数（二進数で30桁）を表現する方法としては、異なる30箇所に同時に置く方法と、同じ場所で30の連続する時刻に生じる刺激で表現する方法があるとして、どちらを選ぶのかを検討している。後者を使うなら、例えば加算のときに、まずは各桁ごとの和を並列計算で求めて、その後に繰り上がる数を加えるという手順を使うことで、高速計算が可能になる。前者ならば、1桁の加算を少なくとも30回行わねばならなくなるのだが、装置を単純化するという原理に従っ

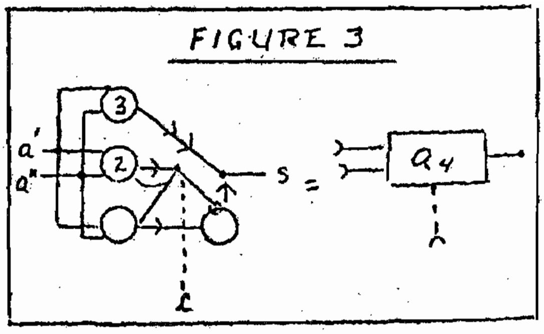

図3　加算回路

て、後者を選択している。まさに逐次的な処理なのだが、この点に関しては30桁を並列に扱う方法もその後の開発の過程で検討されることになる。現在のコンピュータでは、ここでいう意味での並列処理をするのが当然となっている。

7.3で加算のための回路が提案されるが、**図**3が1桁の加算回路である。それぞれが30桁二進数である加数と被加数について、下の桁から順番に、まず各桁の数字を足し、繰り上がりが生じるならそれも加えて、その後上の桁の計算に移るという手順で加算が実現される。

4個のE素子が使われており、下の2個の○は刺激を1個受けると、興奮状態になって刺激を放出する。左側の中間の②の素子は、刺激を2個受けると、興奮状態になって刺激を放出する。左の上の③の素子は、刺激を3個受けると、興奮状態になって刺激を放出する。状態遷移には矢印の個数が表すだけの遅延が生じる。右下のE素子では上側の入力に小さな○が付いているが、これは逆に刺激が興奮状態を抑制するように働くことを意味する。

図3では、a'とa''に加数と被加数を表す刺激が入力され、sに和が出力される。図には示されていないが、下の桁から繰り上がってきた数を表す刺激は②と③に入力される。上の桁への繰り上がりは、②から出力される。

例えば、下の桁からの繰り上がりはなしで、a'とa''に1と1が入る場合を考えてみよう。

a'とa''によって、左の3個のE素子にはそれぞれ2個ずつの刺激が入力される。

すると下の○と中間の②が、興奮状態となって刺激を出力する。

右下の○は、②からの刺激によって抑制されて、刺激を出力しない。その結果、sには刺激が生じず、それは0を表している。同時に、②が興

奮状態になったので、そこからの刺激の出力によって、上の桁への繰り上がりが表される。

このようにして、加算回路は遅延2τで和と繰り上がりを形成するのである。下の桁からの繰り上がり1があり、a'とa''に1と1が入力される場合を考えてみれば、③も興奮状態になるで、sの表す和は1になることが確認できよう。

以降は加算回路を右側のa_4のようなブロック記号を用いて描いていく。一度回路が作られたら、それを簡単な記号へ置き換え、入出力の機能のみを問題とするという論じ方をしているのである。

9 乗算回路

乗算においては、計算の間、乗数と被乗数、および部分積が記憶されねばならない点が加算と異なる。このためには記憶Mでなく、CA内の小さな記憶装置を用いる（**2.2**(a)に示した記憶を指す）。これは今日でいえばCPU内のキャッシュ・メモリであり、水銀遅延線を使った大容量メモリの他に、容量の小さいメモリを作るという発想は、フォン・ノイマンのオリジナルな提案であった。ここでは計算の工夫に過ぎないが、後により複雑なプログラムを考察していく際に、メモリを階層化するというアイデアは重要な意味をもつようになった。この『EDVAC草稿』ではE素子でなく、遅延線[dl]を用いて小さな記憶装置を作ることを考えている。**5.2**から**5.7**で、二進法の乗算を示して、それによって乗算回路が単純化できると論じているので、それを詳しく分析しよう。

科学計算で求められる精度は十進法で8桁と想定しているが、同じ精度の計算を行うには二進法では27桁が必要となる。乗算を行うためには、乗数と被乗数の数字の一つ一つを乗じる。二進法ならば、そのために$27^2 = 729$回の乗算が必要だ。それを集めて、桁を揃えて加える演算が必要なので、全体で必要なのは約2倍の1000 – 1500段階の演算だと推計している。

それに対して、十進法で8桁の数の乗算を行うためには、$8^2 = 64$の約2倍で、約100段階が必要とされるのに過ぎない。つまり、装置が行うべき乗算の回数だけを考えるなら、十進法を使う方がかなり少なくてすむ。だが、十進法の場合は、掛け算表（九九の表）を使う必要があり、そのための装置を追加せねばならない。すなわち、十進法では装置を大きく複雑にするという代償によって、乗算回数を少なくすることができるのだ

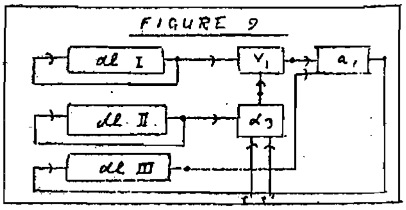

図9　乗算回路

という注意を促す。

それに対して、二進法では回路が単純化できるということを、**7.4**から**7.8**での具体的に乗算の過程の検討を通じて述べている。乗算回路は次の**図**9によって実現される。

まず、乗数を [dl II] へ被乗数を [dl I] へ記憶させる。乗数の一番下の桁の数字を弁別装置 d_3 へ送る。弁別装置は、乗数の数字が1ならば被乗数ともう一つの記憶装置 [dl III] に記憶されている部分積との和を加算回路 a_1 で作り、結果を [dl III] に記憶させる（それまでの部分積と置き換える)。乗数の数字が0のときは何も加えず [dl III] の部分積をそのまま記憶し続ける。これで乗数の一番下の桁に関する処理が終わる。次に、乗数の二番目の桁の数字が1か0かに合わせて同様の手順を行うのだが、その際、被乗数を弁 v_1 で1桁上へシフトして(ようするに、2倍して1桁大きな二進数にして)から加算回路 a_1 へ送るのである。以後同じ手順を乗数の桁数だけ繰り返していけば、[dl III] に積が形成される。

十進法の積では被乗数と乗数の各桁を乗じるのに乗数表を使う必要があるのだが、二進法の積では被乗数を1桁シフトするだけであり、処理が簡単になっていることに注目しておきたい。この点で、演算を実行する際に二進法は大きな利点を持っていることを、フォン・ノイマンは重視していたのである。ここはやや込み入った議論なので、以下に計算の例を示そう。

乗算に必要な計算回数の例

十進法

	2	3	← 被乗数
×	4	1	← 乗数
	2	3	← 3 × 1 ＋ 2 × 1 (1桁シフト)
9	3		← 3 × 4 (1桁シフト) ＋ 2 × 4 (2桁シフト)
9	5	3	

各桁をそれぞれかけ合わせて、位取りして、足し合わせている。必要な

乗算回数は、(被乗数の桁数) × (乗数の桁数) で、この場合は $4 \times 4 = 16$ 回である。

二進法

					1	0	1	1	1	← 被乗数
			×	1	0	1	0	0	1	← 乗数
					1	0	1	1	1	← 乗数が 1 なので、被乗数が部分積になる
								0		← 乗数が 0 なので、部分積も 0
							0			← 乗数が 0
		1	0	1	1	1				← 乗数が 1 (被乗数を 3 桁シフト)
					0					← 乗数が 0
1	0	1	1	1						← 乗数が 1 (被乗数を 5 桁シフト)
1	1	1	0	1	0	1	1	1	1	← [dl III] にこの積が形成される

必要な乗算回数は、二進法の場合は $5 \times 6 = 30$ 回へと増えてしまうのだが、そこで行っている処理は乗数の各桁に合わせて被乗数か 0 を移し、シフトしているだけである。そのために必要な弁別装置と弁を E 素子で構成する方法も説明されているが、それは九九の表のための装置を作るのに比べて単純な回路だと判断するのである。

10　減算、除算、開平回路

8.1-4 では減算と除算の回路網が与えられるが、減算を行うためには負数、除算を行うためには小数を表現せねばならず、まずその議論が行われる。装置で扱う数は **7.1** で 30 桁の二進数だと仮定したのだが、さらに符号のための 1 桁を加える。二進法での小数点について、**8.1** では (31 桁の) 左端から i 番目の数字と $i+1$ 番目の数字の間に入れると一般的な書き方をしているのだが、実際には **9.1** に述べられるように、$i = 1$ のみ, すなわち区間 $[0,2)$ に入る数のみを用いる。そして、数が $[0,1)$ にあるとき、すなわち最左端が 0 のときはそのまま正の数として、数が $[1,2)$ にあるとき、すなわち最左端が 1 のとき補数を用いて負の数を表現するという方式を提案している。補数表現は、今日に至るまでコンピュータで広く使われている。

ここでの定義は、(1 以上の) 数 x は $x' = x - 2$ を意味するというものだ。十進法で例を示すと、1.1 と表現された数は、$1.1 - 2 = -0.9$ を意味するのだと考える。この数を正数に加えて $0.8 + 1.1 = 1.9$ という加算をす

ると、この和は $2-1.9=-0.1$ を意味するので、$0.8-0.9=-0.1$ という減算を行ったのと同じ結果になる。こういう操作が可能になる数学的な理由は、加算と減算は剰余類においても関係を保つからであり、演算回路という視点で見るならば、加算回路を減算に使えるということになる。

加数を補数表現の負数に置き換える、すなわち0と1の間の数 x を $2-x$ に置き換える回路を使えば、それが減数になり、後は加算回路をそのまま使える。この置換を、1を0に、0を1に置き換えるという単純な操作で実現できるのも二進数の利点である。

補数を用いる際の注意点として**9.3**で、加減算では結果が -1 と1の間に入るように、数の大きさを調整する必要があることを述べている。そうなる危険があるときは、あらかじめ加数と被加数に累乗を掛けて小さな数にしておけばよいので、浮動小数点の機能が組み込まれていることが望ましい。だが、そのための回路を作れば装置が複雑になるので、ここでは採否の決定を保留している。浮動小数点の扱いはその後のコンピュータでも判断が分かれ、ハードウェアに組み込むのが当然のことになるのは1960年代以後のことである。

除算回路については**8.3**で説明されているのだが、それは被除数から除数を減じるという計算を、結果が負となる1つ前まで繰り返し、繰り返した回数を商とするというアルゴリズムを用いるものである。二進数を使うので、1回減算を行い、差が正なら商は1、差が負なら商は0となる。次に、差が正のときはその差を1桁左へシフト（2倍した）数が部分剰余となり、差が負のときは被除数を左へシフトした数が部分剰余となって、そこから再び除数を減じるという過程を繰り返していく。この回路では、減算の結果が正か負かにしたがって次の部分剰余が変わるので、そのための弁別装置が必要だが、それは乗算回路に用いた弁別装置にE素子を付け加えることで構成できるとする。

そして、除算回路にさらに必要となる要素として、除数を保存しておく [dl I]、被除数と部分剰余を保存する [dl II] と [dl III]（最初は被除数が入り、その後は計算過程の部分剰余と部分剰余から除数を引いた差とが入る）、商が入る [dl IV]、その間で数をやりとりする弁 ν_1、ν_{-1} が示されている（**図** 13）。

開平回路は**10.1**に示されているが、次の方法を用いる。ここで、数 $x(<1)$ の平方根を二進数で 2^{-n} の桁まで求めたいのだが、その1桁上までを $a_1\cdot 2^{-1}+a_2\cdot 2^{-2}+\cdots+a_n\cdot 2^{-n}=\sum_{i=1}^{n} a_i\cdot 2^{-i}=a\cdot 2^{-n}$ と置いている。そして

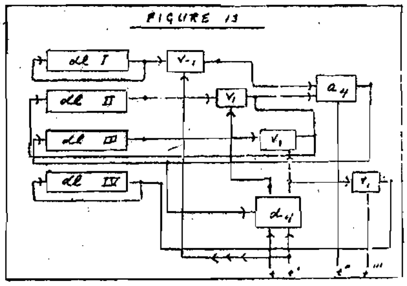

図 13　除算回路

$a_{(n+1)}$ を加えた $(2a+a_{(n+1)})\cdot 2^{-(n+1)}$ の 2 乗が x を超えない最大の $a_{(n+1)}$ の値を探す。だが、ここでは二進数で計算しているので、$x-(2a+1)^2\cdot 2^{-2(n+1)}$ の正負を判定し、負ならば a_{n+1} は 0、非負ならば 1 とするだけでよい。

この一つ前の過程で a を求めるための計算を行っているはずなので、それを $x-a^2\cdot 2^{-2n}=r\cdot 2^{-2n}$ と置いて、r を部分剰余として保存しておく。この部分剰余を 2 桁左へ移動、すなわち 4 倍して、$4a+1$ を引くということは、$4r-(4a+1)=4x\cdot 2^{2n}-4a^2-(4a+1)=\left(x-(2a+1)^2\cdot 2^{-2(n+1)}\right)\cdot 2^{2(n+1)}$ となるので、この正負を判定すれば a_{n+1} が決まる。このようなアルゴリズムを用いることで、開平回路も除算回路と共通の回路で作りうることが示されている。

実際にどの回路を作るべきかについての結論は **10.3** に述べられているのだが、四則演算の回路について整理すると次のようになる。

- 加算回路が基本である。
- 乗算回路は、加算回路＋記憶＋弁別装置＋弁で構成できる。
- 減算回路は、加算回路に補数表現を適用する。
- 除算回路は、乗算回路と減算回路、2 つの数の大小を比較する弁別装置で構成できる。
- 開平回路は、除算回路と加算・減算・桁移動のための時間調整装置で構成できる。

四則と開平算を行う回路の全ては本質的に除算回路の一部として含まれていることから、CA には除算回路を作り、他の演算はその回路に論理制御を加えることで行うと決断するのだが、その前に乗算、除算、開平については、計算によって求めるという方法もあり得るので、それとの

比較をしている。例えば、乗算は対数表と真数表を使えば加算に還元できるし、装置にはそれらの関数表が必要になるのではないかという疑問を与えている。ここでの答は、関数表を十分に高い精度で利用するには補間法が必要で、補間法には乗算が含まれるから、乗算が関数表に先立つ基本的な演算として組み込まれねばならないということである。

除算と開平算についてはやや事情が異なり、漸化式を用いて反復計算をしていけば、高速で商や平方根に収束することも使える。除算の近似値を与える漸化式は次のもので、$u_{n+1} = 2u_n - au_n{}^2 = (2 - au)u$ ある。これについては、$y = x$ と $y = 2x - ax^2$ のグラフを描いてみれば、点列 (u_n, u_{n+1}) がグラフの交点 $\frac{1}{a}$ に収束することが分かるだろう。開平算については、$y = x$ と $y = \frac{3}{2}x - 2ax^3$ の正の交点が $\frac{1}{\sqrt{4a}}$ になることを用いている。

こうした近似計算はできるのだが、「これらの演算過程は多かれ少なかれ含めねばならない論理制御を必要とし、それは些少でない回数の乗算に置き換えられる」が、それは時間の無駄遣いである。それに対して、実際に除算回路を作れば、他の四則演算と開平計算はすべて除算回路の一部として含まれるのだから、CA に除算回路を作った方がよいという判断をしている。

その他の演算——例えば、対数、三角関数とそれらの逆関数など——について **10.4** で言及するが、これらは外部の関数表から補間法によって得ることにすると考えている。それは、これらの関数の計算は開平算のような単純な四則演算で得られるものではなく、級数計算をするよりは関数表を用いる方が容易であると判断する。これらの記述から、論理的な可能性の追求ではなく、装置を簡単にすることと、よく使われる計算を合理的な時間で実行することとのバランスを考えた、実践的な問題が選択の基準となっていたことが確認できる。

11　丸めの手順

『EDVAC 草稿』では、標準的な数を 30 桁の二進数とするが、加算では繰り上がりが生じ、乗算では桁をシフトするため、和や積が 30 桁に収まるとは限らない。一般に、30 桁の二進数の積は 60 桁になるが、装置で扱える 30 桁の標準的な数にするための近似が必要がある。**9.4** では、数を丸める、すなわち 31 桁以上の数を適切に近似する方法を論じている。

よく使われる「四捨五入」は、1 から 4 の 4 つの場合に切り捨て、5 か

ら9の5つの場合に切り上げを行うため（0のときはどちらも行わないと考える）、丸めた結果の方が少し大きくなってしまうという問題がある。実験の際に観測した結果を1回だけ丸めるくらいなら大きな問題は生じないが、コンピュータでは丸めた結果を次の入力して出力を再び丸めるということを繰り返していくと、偏りの累積が深刻な問題を生じるのではないかという点を危惧をしているのである。

それを避けるためにガウスの方法を採用するのだが、十進法でのガウスの手続は、31桁目が5のとき、30桁目が奇数のときは切り上げ、偶数のときは切り捨てるという規則によって、丸めた結果の偏りが生じないように工夫したものである。このガウスの方法をそのまま二進法に適用すると、次のようになるとする。

30桁と31桁の数字	丸めた結果（30桁目の数字）
00	0
01	0
10	1
11	10

4行目は、30桁目は0で、29桁目に（繰り上がった）1を加えるという意味である。丸める前後の数字の平均値を十進法で現すと、丸める前は、$(0+1+2+3)\div 4=1.5$であり、丸めた後も、$(0+0+2+4)\div 4=1.5$であるので、偏りは生じていない。

これに対して『EDVAC草稿』ではモークリーの提案した次の方法を推奨するのだが、こちらは繰り上がりが生じない分だけ回路が簡単になるとする。

30桁と31桁の数字	丸めた結果（30桁目の数字）
00	0
01	1
10	1
11	1

丸める前後の数字の平均値を十進法で現すと次のようになる。丸める前は、$(0+1+2+3)\div 4=1.5$であり、丸めた後は、$(0+2+2+2)\div 4=1.5$である。いずれの手続でも、丸める前の数と丸めた結果の数の平均値は同じで、偏りは生じていないので、装置が単純になるモークリーの方法

を選ぶべきだと述べているのである。このように数値計算で生じうる問題をていねいに検討しているのも、『EDVAC草稿』の特徴である。

現在のコンピュータではこの偏りはあまり深刻な問題だとは考えられていないようで、たいていは、はみ出した桁（ここでは31桁目以降）を切り捨てるという単純な処理を行い、精度が重視されるときには倍精度等の桁を増やすプログラムを用いる方法で対処している。ここで検討された手順はその後に与えた直接的な影響は小さいと思われるが、コンピュータに関する最初の文書として、数学的に考えうる様々な問題を考慮に入れていたのである。

12　『EDVAC草稿』以後

本稿での考察は算術回路までで終えるが、『EDVAC草稿』では**11.4**で算術回路を検討を終えた後、微分方程式の近似解等を求めるために必要な演算を制御し、中間の結果を保存するのにどれだけの記憶容量が必要かを検討する。そして一連の制御のために必要な命令を数え上げ、符号体系を整理していくという展開になっている。

1945年6月の『EDVAC草稿』の配付に続いて、ENIACの公表、ペンシルバニア大学のサマースクールでの講義等を通じてムーアスクールでの研究成果が伝えられ、アメリカとイギリスのいくつかの場所でそれらのアイデアにもとづくコンピュータ開発が始まる。フォン・ノイマン自身も、自らが籍を置いていたプリンストン高等研究所にスタッフを集め、コンピュータ開発を始めたのだが、その過程で今回紹介したアイデアについて様々な改良が加えられ、その成果は文書を公刊されていった。『EDVAC草稿』から始まる一連の情報が、現代のコンピュータの基本的なアイデアをもたらしたのである。

13　まとめ

ここまで見てきたように『EDVAC草稿』は、E素子などの電子的部品、記憶に使う遅延線については抽象的に機能を示すだけにとどめ、電子工学的な側面に関する言及はしなかった。フォン・ノイマンが数学者、論理学者として、自らが得意な分野について論じたといえるが、これは演算素子が真空管から半導体に変わり、記憶装置も遅延線、フェライトコア、半導体と変わっていったとしても通用する、普遍的な議論となっていた。

本稿ではあまり論じなかったが、二進法はその算術によってブール代数が実現できるといった論理的な面からも、デジタル・コンピュータにとって都合のよい方式である。

その一方で、算術回路の構成法を具体的に検討する過程では、論理的な可能性を追究するだけなく、装置を単純にすることと、よく使われる計算を合理的な時間で実行することとのバランスを考えた、実践的な問題を選択基準としていた。それは真空管の寿命や遅延線でのデータの逐次的な保存といった、直接には考察対象としていない装置の特性を考慮し、当時の技術の制約に対応したものだった。『EDVAC 草稿』で二進法を採用したのも、真空管を使用するという制約に対応し、算術演算を単純な回路で実現できるという実践的な理由があった。

フォン・ノイマンがコンピュータの設計において二進法を採用するべきだと提唱したことには、以上のようにいくつかの理由があったのだが、『EDVAC 草稿』においては利用する予定の素子の制約に対応した実践的な理由が強いように読める。実際に、フォン・ノイマンはプリンストン高等研究所でのコンピュータでは記憶装置を遅延線からブラウン管に変えた結果、数の逐次的な保存を並列的な保存となり、ここで紹介した算術回路の構成なども変更されていく。そして、1960 年代に素子が真空管から半導体に変わった後には、『EDVAC 草稿』で強調された装置を単純化するという原理は意味を持たなくなる。「ノイマン型アーキテクチャ」と呼ばれる方式に関しても、その詳細や意義は時代とともに変化していくのである。