

Gondolatok Neumann János

***First Draft of a Report on the EDVAC* című,**

1945 júniusában megjelent tanulmányáról

Kovács Győző–Szelecsán János

Ezt a tanulmányt Neumann János írta a második amerikai elektronikus számítógép tervezésével kapcsolatban. A világ számítástechnikai szakemberei ezt a tanulmányt – amit Neumann János szerzőtársak nélkül alkotott és tett közzé – a számítógép-tervezés első és legátfogóbb leírásának tartják, ami lehetővé tette a számítógép iránt érdeklődő tudományos szakembereknek, mérnököknek és matematikusoknak, hogy a negyvenes évek közepén a számítógépekről szóló, minden fontos információhoz hozzájussanak. Talán nem túlzás azt állítani, hogy ez a „First Draft...”, „Vázlat...”, „Jelentés” (mindhárom elnevezést használjuk a következőkben) tette lehetővé, hogy számos egyetemen és kutatóintézetben elkezdődjék a számítógépek tervezése és építése, azt is mondhatnánk, ez volt az a hólabda, ami elindította a számítógépes korszak – azóta is tartó – lavináját.

Írásunkban lényegében a „First Draft...”-ot ismertetjük, de „beágyazzuk”, ugyanis – mielőtt rátérnénk ismertetésére – röviden elmondjuk, hogy milyen fontosabb mérföldkövek voltak a számítástechnika történetében Neumann előtt (az „időszámítás előtt”) és azt is, hogyan jutott el Neumann a „First Draft...” megírásáig.

Írásunk végén – szinte „Függeléként” – a Jelentésben szereplő elveken alapuló ténylegesen megtervezett EDVAC gép utasításrendszerét, valamint az EDVAC környezetében Neumann János közreműködésével létrehozott „magas szintű nyelvet” (3. fejezet) ismertetjük.

1. Út a „First Draft...”-ig, a Neumann János előtti informatikatörténelem

Mérföldkövek a Neumannig vezető úton

Számos kutató feltételezi, hogy az idők homályába vesző őskorban, az őstársadalomban, amikor a számfogalom, a számlálás, a mennyiség fogalma már kialakulóban volt, az ősemberek is használtak számolást megkönnyítő eszközöket, elsősorban a mindig kéznél lévő ujjaikat. Ennek a számolóeszköznek tulajdonítják, hogy az emberiség nagy részénél a tízes számrendszer terjedt el; ha – netán – az emberek nem kétszer öt, hanem kétszer négy ujjal születnek, akkor nagyon valószínű, hogy ma nem tízes, hanem nyolcas számrendszerben végeznénk a műveleteket.

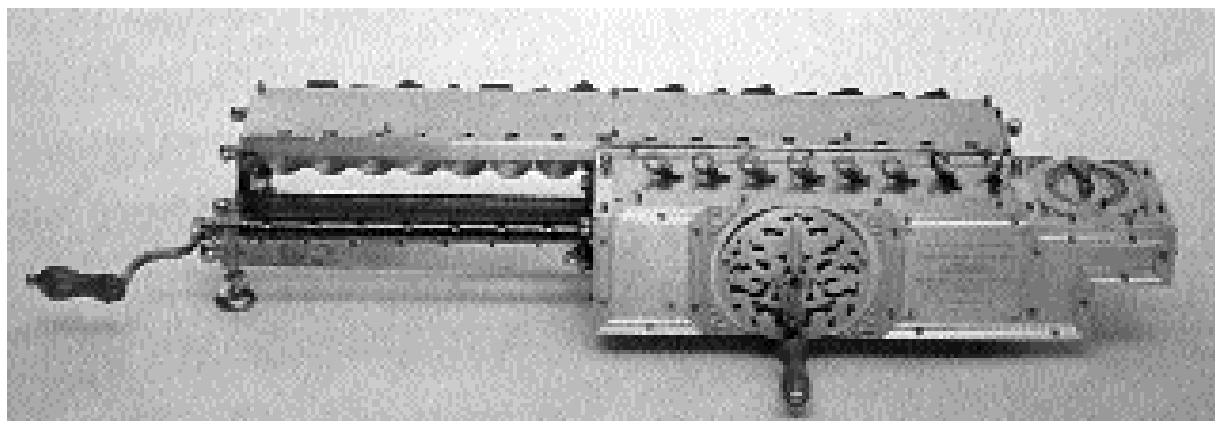
A számolás megkönnyítésének az igénye tehát már a nagyon korai embereknél is megvolt, az ember mindig szeretett volna eszközöket találni számolási feladatainak a megoldására. Előbb kavicsokat (calculus) vett föl a földről, amiket rendbe rakott és ezekkel számolt, azután kitalálja a kavicsokkal való számolás modern változatát, a golyós számológépet,

a számológépeket, felfedezi a matematikát, és hamarosan már meghatározza a matematika törvényszerűségeit is.

Az idő előrehaladtával az ember egyre több matematikai problémát akart megoldani, amihez már nem volt elég a két keze, a golyós számológép is kevés, az sem elég, hogy ismeri az algebra szabályait, egyre csak azon gondolkodik, hogy a számolás műveletét egy számológéppel megkönnyítve, még bonyolultabb feladatokat tudjon megoldani.

Azután felnéz az égre – már az őseink is megnézegette az égitesteket, amikből még isteneket is kreált magának –, azután felfedezi az égi mechanikát (a napszakok és az évszakok változását, a betlehem-i csillagot, a vést hozó üstökösöket, a Nap- és Hold-fogyatkozásokat stb.), sőt hamarosan ki is tudja számolni bizonyos égi jelenségek bekövetkezését. Ehhez viszont számolnia kell, sőt sokat kell számolnia, ami gép nélkül egyre lehetetlenebb.

Így nem csoda, hogy az első kerek számológépeket csillagászok találják fel, Al-Kassi, a szamaritai obszervatórium vezetője (1393–1449), valamint Wilhelm Schickard (1592–1635) Tübingenben, a csillagászat tanára és matematikus. Híres számológép-konstruktőr volt Blaise Pascal (1623–1662), aki ugyan nem volt csillagász, de a számológépét csillagászok is használták. Időben majdnem kortársa volt Gottfried Wilhelm Leibniz (1646–1716), polihisztor, sok mindennel foglalkozott, megalkotta például a számológépéhez a Leibniz-kereket, amit – számológépek konstruálására – még a 20. században is használtak.

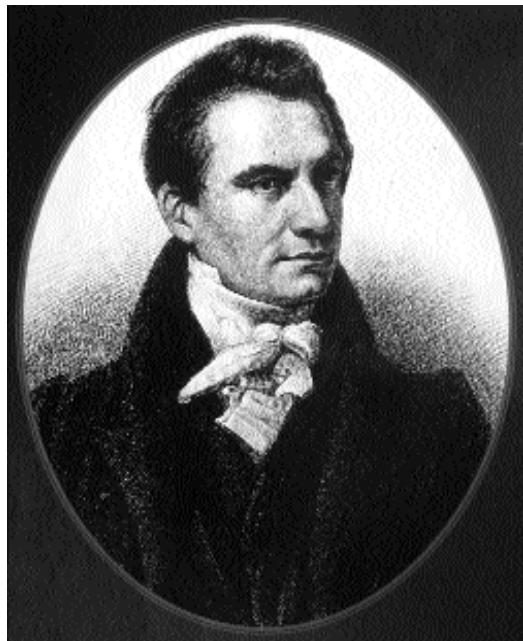


1. ábra. Godfried Wilhelm von Leibniz (1646–1716) matematikus, a mechanikus – négy alpműveletes – számológép megalkotója. A gép legfontosabb alapeleme a lépcsős Leibniz-fogaskerék volt, amit – mechanikus számolóberendezésekben – még a 20. században is használtak. (1670)

Azután Angliában megjelent Charles Babbage (1791–1871), aki ugyancsak közeli kapcsolatban volt a csillagászzal és kora neves csillagászaival, de inkább matematikus volt, és megszállottja a számolás gépesítésének. Anglia ekkor válik tengeri nagyhatalommá, az óceánokon való hajózásban kitüntetett szerepet kap a csillagászat és az időmérés, hiszen a világtengereken csak a pontos idő és az égitestek alapján lehet tájékozódni. A tökéletes hajózási órát 1759-ben John Harrison órásmester alkotja meg, amivel elnyeri a pontos tengeri óra elkészítésére kiírt díjat.

A királyi csillagászok közzé teszik a csillagászati évkönyveket, bennük azokat a táblázatokat, amiknek alapján – megfelelő mérőműszerek (szextáns, oktáns stb.), valamint a már említett pontos óra segítségével – a világ bármelyik pontján pontosan meg lehet határozni, hogy a hajó hol tartózkodik. Egy, a táblázatokat kiszámoló gép megalkotása egyre sürgetőbbé vált,

ugyanis a csillagászati módszerek és a matematikájuk is állandóan fejlődik, ezért a hajósok egyre újabb táblázatokat igényelnek, ezt az egyre nagyobb számítási igényt már csak gépekkel lehet kielégíteni.

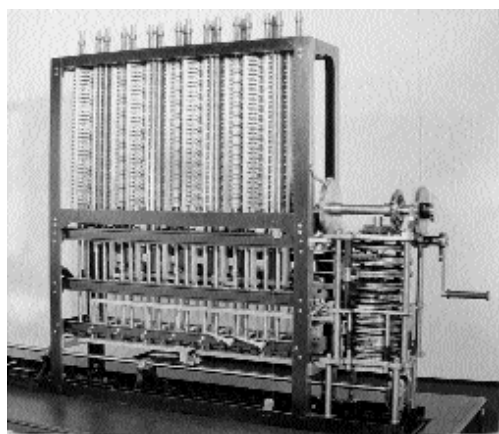


2. ábra. Charles Babbage (1791–1871). Matematikus és csillagász, a Királyi Csillagászati Társaság tagja, az első, programozott, mechanikus számológépek – a Difference Engine és az Analytica

Ekkor lép a színre Charles Babbage, aki célul tűzi ki, hogy a hajózási csillagászati táblázatokat géppel számítsa ki, sőt ki is nyomtassa. A Királyi Csillagászati Társaság már 1820-ban kitünteti a „Gépek matematikai táblázatok kiszámításánál való alkalmazásának tapasztalatai” c. munkájáért, igaz Babbage először még egy gőzgépmeghajtású számológépre gondolt.

A brit kormány 1823-ban dönti el, hogy támogatja Babbage – csillagászati táblázatokat számoló – differenciagépének a megépítését, amit az alkotónak sohasem sikerült tökéletesen befejeznie.

Mint megannyi más zseni – mielőtt a gépet befejezte volna –, új gondolat ébredt benne, kitalálta, hogy az általa kidolgozott elvek és a differenciagép alapján építeni tudna egy általános célú számológépet is, ami nem csak kötött formulák alapján képes számolni, hanem bármilyen formulát – aminek meg lehet írni a *programját*, és azt – *lyukkártyákon* – a gépnek meg lehet adni – ki lehet számolni. A találmányát *analitikus gépnek* nevezte.



3. ábra. Babbage második gépének, az Analytical Engine-nek a működő másolata, amit – az eredeti tervek alapján – születésének a 200. évfordulójára a londoni Science Museumban építettek meg. (1991)

Babbage ezzel feltalálta a programozott mechanikus számítógépet. Nagy szerencséjére csatlakozott hozzá egy fiatal matematikusnő, Ada Byron, Lady Lovelace (1815– 1852), aki azonnal megértette a gép nagyszerűségét. Babbage egyik olaszországi előadásáról Luigi F. Manebrea tábornok (1809– 1896) írt beszámolót, aminek az angol fordítását Lady Lovelace végezte el – az előadásnál háromszor

hosszabb – jegyzetekkel. Babbage zseniális alkotását és a gép programozását a világ ezekből a Lady Lovelace által készített jegyzetekből ismerte meg.

Babbage-nak egyik gépét sem sikerült tökéletesen befejeznie, noha a már működő differenciagépet az 1862-es londoni világiállításán is bemutatták.

Hermann Goldstine könyvéből idézet:

„Egy Pehr Georg Scheutz (1785–1873) nevű svéd nyomdász Babbage gondolataitól indítva megépítette a differenciagépet, és 1854-ben Londonban – Babbage jelentős segítségével – be is mutatta. (...) Scheutz ugyanis 1834-ben Edward fiával együtt egy, az *Edinburgh Review*-ban megjelent cikkben olvasott Babbage differenciagépéről. A cikk felkeltette érdeklődésüket, és hozzáfogtak egy lényegesen módosított, tökéletesített változat megépítéséhez. 1834-ben a gép működőképes volt. (...) 1853-ra Scheutz befejezte a gép tökéletesített változatát, ami 1855-ben a párizsi világkiállításon aranyérmert nyert.”

Babbage Scheutz gépéről – ebben az időben – így írt: „(...) A gép pontos másolatát a Messrs. Dunkin and Co. cég készítette el az angol kormány számára, és jelenleg a Somerset House-ban, az anyakönyvi irattárvezető irodájában használják.”

A hajózási – csillagászati – számológépigény után felmerült a nagytömegű adatok feldolgozásának a szükségessége is, ennek az első lépése a népszámlálási adatoknak a korábbiaknál gyorsabb feldolgozása volt. A sokszereplős történetből Hermann Hollerith (1860–1929) nevét kell megemlítenünk, aki 1889-ben kapott *a népességi statisztikák táblázatba foglalására* szabadalmat, az első Hollerith-gépek segítségével az 1890-es amerikai népszámlálást nagy sikerrel bonyolították le. Amíg a korábbi – 1880-as és a még korábbi – népszámlálások alkalmával az adatok kézi feldolgozása általában a következő népszámlálásig tartott, addig – 1890-ben – az első gépi feldolgozásnál az első adatokat a lyukkártyák összegyűjtése után egy hónappal már közzé lehetett tenni.

Adatfeldolgozás céljából az iparban gyártott Hollerith-gépek megjelentek a nagyvállalatoknál, a közműveknél stb., és minden olyan helyen, ahol sok adatot tároltak és dolgoztak fel. Nem sokkal később a tudományos élet – éppen a csillagászat –, valamint a haditechnika is felfedezte a gépi számolásban rejlő tudományos és katonai lehetőségeket.

Ehhez – természetesen – a haditechnikának is fejlődnie kellett, például a lövészet technikája a tüzérek művészetéből a ballisztika tudományává vált. Azután – az első világháború alatt – megjelentek a hadirepülőgépek, amiket földi léghárítással szerettek volna leküzdeni, így azután kifejlődött a haditechnikai méréstudomány, amit – hogy hatékony eszközöket és módszereket eredményezzen – egyesíteni kellett a katonai számolástechnikával.

Mielőtt elérnénk Neumann Jánoshoz, egy példával mutatjuk be ennek a fejlődésnek egy állomását.

Az ellenséges repülőgépek megsemmisítéséhez a hadseregnek egy olyan eszközre volt szüksége, ami a repülőgépet tűzzel fogadó lövegeket a repülőgép mozgásának, az időjárási viszonyoknak, a lövegek adatainak ismeretében automatikusan célra állítja. Ez a löelemképző. A II. világháború előtt ezt a feladatot a leghatékonyabban egy magyar mérnök, Juhász István (1894–1981) oldotta meg, aki Budapesten kifejlesztette a GAMMA-Juhász löelemképzőt, amit számos ország: például Svédország, Argentína, Kína, Svájc, Hollandia, Norvégia, Finnország és Irán – majd a második világháború után Egyiptom hadseregeiben is hadrendbe állítottak. A löelemképző valójában egy bonyolult mérőrendszerrel összekötött vezérlőrendszer és az ezt irányító, ballisztikus fixmemóriával rendelkező elektromechanikus analóg számítógép volt.

Időben most értünk el a negyvenes évek első éveiben Neumann Jánoshoz és az elektronikus számolóeszközökhöz. Az amerikai hadvezetés, a negyvenes évek elején – látva a németországi eseményeket – biztosra vette, hogy nem maradhat ki az Európában dúló háborúból, ezért igyekezett a háborús részvételre felkészülni. Azt is látták, hogy a harcot az I. világháború után kifejlesztett modern haditechnikai eszközökkel – mozgékony gépesített hadsereggel, modern repülőekkel és tüzérséggel kell megvívnia, ezért elérkezettnek látta az időt, hogy a tudományt – elsősorban a matematikát és a fizikát – is „hadrendbe” állítsa.

Már a harmincas években elkezdődött az európai – főleg zsidó tudósoknak: Einstein, Wigner, Kármán, Oppenheimer, Szilárd, Teller, nem utolsósorban Neumann és mások – a náci megszállás elől Amerikába menekülése, ezek az emberek óriási mértékben megerősítették az ország tudományos és így védelmi kapacitását.

Hamarosan behívták katonai szolgálatra Hermann H. Goldstine-t, a ballisztika tanárát, matematikust, aki azt a feladatot kapta, hogy a Ballisztikai Kutató Laboratóriumban modernizálja – többnyire az első világháborúban készült – tüzérségi és bombázási táblázatokat. Goldstine felmérte a feladatot, és megállapította, hogy az akkor használt matematikai eszközökkel – Bush-analizátor, két jelfogós számítógép (Aiken és Stibitz jelfogós gépei), elektromechanikus asztali számológépekkel stb. – nem lehet a munkát rövid idő alatt elvégezni, egy új elvű, hatékony és nagyságrendekkel gyorsabb számolóeszközt kell keresni vagy kifejleszteni.

Közben elindult a Manhattan-terv, a versenyfutás a német tudománnyal, hogy ki tudja előbb megalkotni és a háború eldöntésére bevetni a minden eddiginél hatékonyabb fegyvert, az atombombát. A tudósok csatasorba álltak, egyik részük Los Alamosban atombombát fejlesztett, a másik részük pedig megalkotta a Pennsylvaniai Egyetem Moore (villamosmérnöki) intézetében az első nagyon gyors számítóeszközt, az ENIAC-ot.

Ezt követően találkozott Hermann Goldstine az aberdeen-i állomás peronján Neumann Jánossal, elindult egy csodálatos együttműködés, ami – talán nem túl nagy szavak – elvezetett az első számítógépekhez és – hosszú távon – az informatikai társadalomba.

Hogyan jutott el Neumann János a First Draftig?

Neumann János „ismerkedik” a számológéppel (Moore School)

Neumann Jánost 1943-ban kinevezték tanácsadónak az atombomba előállítását előkészítő Manhattan-projekthez, egyúttal az aberdeeni Ballisztikai Kutató Laboratóriumba is. Az itt felmerülő numerikus matematikai problémák megoldásához mindinkább szüksége lett volna valamilyen mechanikus gyors számolóeszközre. Ezt többször szóvá is tette. 1944 januárjában levelet írt Warren Weavernek, az Alkalmazott Matematikai Bizottság elnökének. Azt kérdezte, hogy a kormánynak milyen eszközei vannak a számítások elvégzésére. Weaver beszélt neki Aikenről, Stibitzről és Wallace Eckertről. A Moore Schoolt nem említette meg, mert nem tartotta reális kezdeményezésnek.

Pedig Moore Schoolban John Mauchly és J. Presper Eckert vezetésével – John V. Atanasoff elveinek a felhasználásával – ekkor már elkezdtek építeni az ENIAC (*Electronic Numerical Integrator and Computer*) első elektronikus számolóeszközt. Az 1923-ban létrehozott „Moore School of Electrical Engineering” a Pennsylvania Egyetem önálló elektromérnöki részlege, „kara” volt. Írásunkban Neumann Jánosról beszélünk, de az előbbi két név megemlítése nélkül ezt több okból nem tehetjük meg. Ha Neumann Jánost a számítógépek atyjának nevezzük, akkor nem volna

Találkozás a pályaudvaron



4. ábra. A Pennsylvániai Egyetem villamosmérnöki kara, a Moore School, ahol az ENIAC épült. Az ENIAC emléktábla

Tanácsadóként Neumann sokat utazott Los Alamos, Aberdeen és Princeton között. 1944 júniusában az aberdeeni pályaudvar peronján állt, amikor Goldstine csatlakozott hozzá. (Hermann H. Goldstine matematikus – mint láttuk – a hadsereg, a Pentagon képviselőjeként vett részt az ENIAC projektben.) Goldstine azonnal megismerte Neumannt, mert néhány évvel korábban részt vett egy konferencián, ahol az előadók egyike Neumann János volt. „Nyilvánvaló az, hogy Neumann rám emlékezett volna, lehetetlen” – mondja Goldstine. Mindenesetre Goldstine odament Neumannhoz, bemutatkozott.

Neumann János nagyon kedvesen fogadta Goldstine bemutatkozását, és megkérdezte, hogy mi járatban van itt. Goldstine elmondta, hogy az ENIAC projektben vesz részt. Mindjárt el is mesélte Neumann-nak, hogy ez a ENIAC gép 333 szorzást végez másodpercenként. Neumann-nak a szemei felcsillantak. A beszélgetésről Goldstine azt mondja, hogy a találkozás pillanatok alatt egy doktorátusi vizsgává változott, Neumann mindenről kikérdezte az ENIAC géppel kapcsolatban. Goldstine megemlíti azt is, hogy Neumann teljesen megértette, miről van szó, és hogy mit tud az ENIAC.

Látogatás a Moore Schoolban

Néhány nappal később Neumann felhívta Goldstine-t és arra kérte, hogy mutassa be neki Moore Schoolt, ahol az ENIAC gép épült. Mauchly így emlékszik vissza arra az időre. „A látogatásról valamikor augusztus első hetében hallottunk. Mondták, hogy ide jön egy kiváló matematikus, beszélni akar velünk, és könnyen meglehet, hogy nagy segítséget nyújthat nekünk, mivel ő a világ legnagyobb matematikusa. Izgatottan vártam, hogy találkozzam ezzel az emberrel és kiderüljön, hogy miben tud segíteni.”

A látogatás végül szeptember első hetében létre is jött. Goldstine bemutatta a Moore Schoolban dolgozó munkatársakat, és megmutatta az akkumulátorokat (mai kifejezéssel regisztereket). Ez egy kicsivel gyorsabban számolt, mint Neumann János. „Neumann János megnézte a kiállításunkat, és mindenről mindent elmondtunk neki. Utána bementünk egy szobába, és elkezdünk beszélgetni egy tábla előtt. Ismertettük vele a gép terveit, amit éppen építettünk. Neumann világosan és azonnal megértette, hogy ennek a gépnek milyen számítási képessége lesz, és ennek az ENIAC gépnek a Los Alamosban (ahol az atomkutatások folytak) felmerülő problémák megoldásában nagy szerepe lehet. A beszélgetés közben Neumann egészen előre ment, és elkezdett gondolkodni azon, hogyan lehetne a mi terveinkhez képest jobb számológépeket építeni” – mondta Mauchly.

Neumann, a tanácsadó

Neumann János ettől kezdve tanácsadóként bekapcsolódott az ENIAC projektbe, és több alkalommal is meglátogatta Moore Schoolt. Lelkes támogatójává vált a Moore Schoolban folyó

munkának. Állandó kapcsolatban állt Mauchly-val és Eckerttel. Neumann különösen a számítógép logikai felépítésének az elemzésében volt nagy segítségükre. A technikai részek abban az időben nem keltették fel Neumann érdeklődését.

Neumann János közreműködése az ENIAC-ban nem csak szakmailag volt jelentős, hanem azzal is, hogy a fejlesztés végén bekapcsolódva, nevének presztízse miatt a tervezők további 105 000 dollárt kaptak a Pentagontól 1944 októberében.

Neumann megjelenésével a korábban szokásos informális beszélgetések helyett napirendekben meghatározott üléseken vitatták meg a problémákat. Neumann János nagyon aktív volt ezeken az üléseken, rengeteg javaslatot tett. Az ENIAC-kal a fő probléma a programozható memória (az adatok tárolásához szükséges szerkezet) volt. Látták, abból a célból, hogy a gép igazán jól használható legyen, szükség van arra, hogy mind az utasításokat, mind pedig az adatokat „meg tudja jegyezni” (tárolni tudja), és hogy az utasítások anélkül lehessenek megváltoztathatók, hogy egy operátornak dugaszolással, billentyűk lenyomásával kellene változtatni a programon (mint az ENIAC esetében).

Eckert és Mauchly közben megkapták az ENIAC-ra a szabadalmat. Gyakorlati szakemberek voltak, és világosan látták a gépeiknek a kereskedelmi lehetőségeit. A Moore Schoolban sokan nem hitték, hogy a számítástechnikából kereskedelmi haszon is származik. Ebbe a csoportba tartozott – érdekes módon – Neumann János is. Meg volt győződve arról, hogy ezek a munkálatok voltaképpen a háború miatt voltak fontosak, vagy esetleg a „tisza tudomány” szempontjából. A dolog szabadalmi része egyáltalán nem érdekelt.

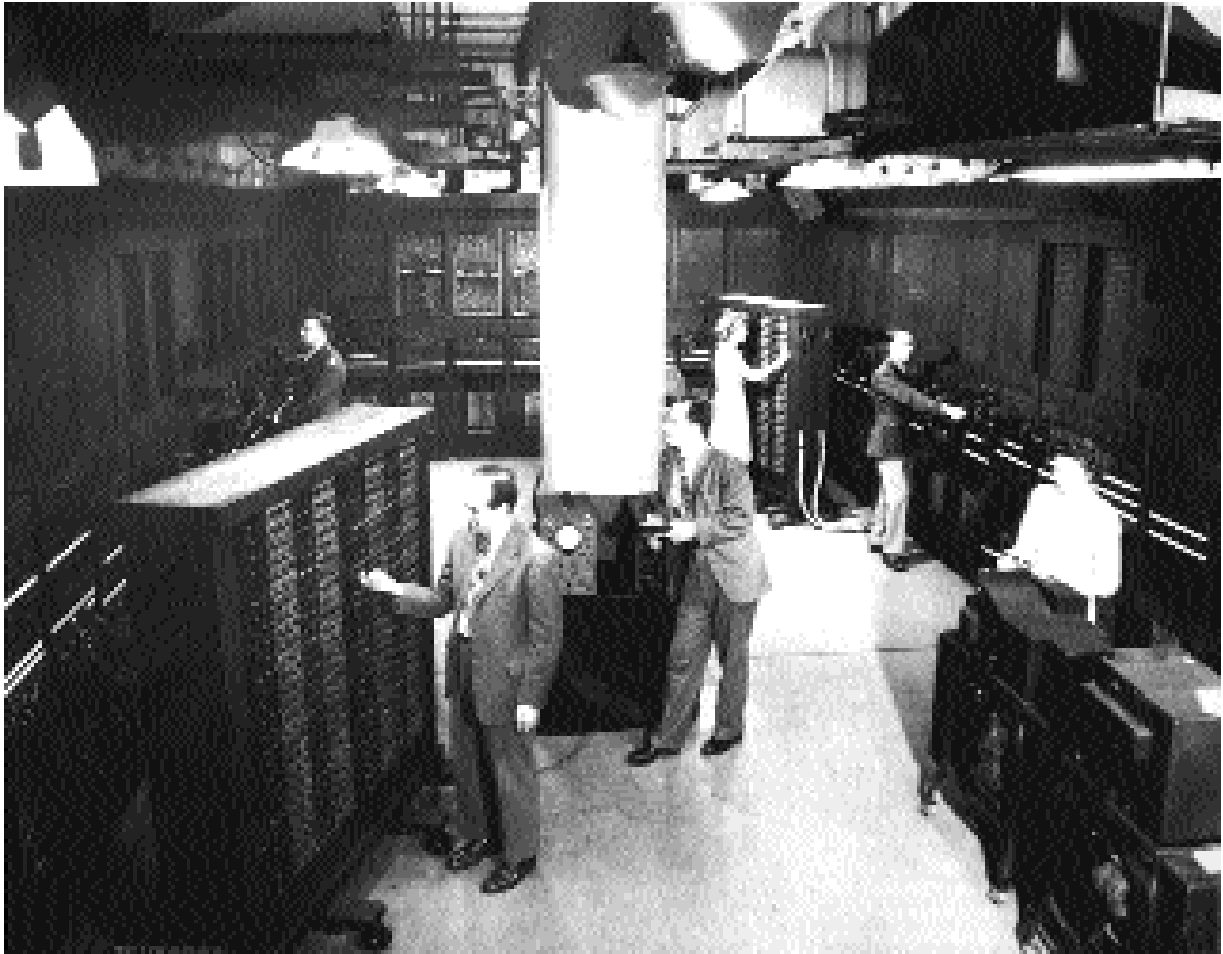
Annak, hogy Neumann János „becsatlakozott” a Moore School Projektbe, két közvetlen hatása is volt. Az első, hogy elismertette a projektet, például a Nemzeti Védelmi Kutatási Bizottsággal, aminek eredményeképpen további támogatást kaptak a projekthez.

A második eredmény az volt, hogy miután Neumann János is érdekelt volt az ENIAC-programban, a Moore Schoolal szerződést kötöttek egy új, sokkal nagyobb teljesítményű számítógépnek, az EDVAC-nak a kifejlesztésére.

Neumann először mint potenciális felhasználó (user) érdeklődött a Moore Schoolban folyó munkálatok iránt, de később már mint tudományos és technikai tanácsadó a projekt egyik főszereplőjévé vált; számos konferencián vett részt, amelyeken javaslatokat fogalmazott meg az EDVAC logikai tervezésével kapcsolatban. Amikor Neumann Los Angelesben volt, állandó levelezésben állt Goldstine-nel, aki informálta őt a projekt haladásáról; Neumann Goldstine közvetítésével juttatta el javaslatait a projektben résztvevőknek.

Röviden az ENIAC gépről

Arthur W. Burks, aki maga is oszlopos tagja volt az ENIAC-ot létrehozó csoportnak, „From ENIAC to the Stored-Program Computer: Two Revolutions in Computers” c. cikkében az ENIAC-kal kapcsolatban a következőket írja: „...két fontos forradalom volt a számítógépek területén. Az első az elektroncsövek alkalmazása, amelyek segítségével gyors, megbízható, nagy teljesítményű általános célú számítógépet lehetett építeni. Ez a fejlesztés John Atanasoff-fal indult, aki egy speciális célú elektronikus számítógépet épített, és az ENIAC-ban kulminált. ... Az ENIAC forradalmi jelentőségű volt: ez volt az első elektronikus, digitális, általános rendeltetésű tudományos számítógép, amely 1000-szer gyorsabban végezte a számításokat, mint elektromechanikus versenytársai. ... A másik forradalom a tárolt programú számítógép létrehozása volt.”



5. ábra. Az ENIAC, a világ első elektronikus számolóberendezése, 1943 és 1946 között épült.

A gép tervezését 1943 áprilisában kezdték, 1943 júniusában már elindult a beruházás, 1945 végén fejezték be.

A fő cél voltaképpen lövéstani számítások elvégzése volt. A gép kb. 18 000 elektroncsövet tartalmazó áramkörökből épült fel.

Főbb jellemzői:

- Különálló, „félautonóm” egységekből állt, a vezérlés részben elosztott volt (az egyes egységekben lévő vezérlést egy „koordináló” vezérlés „kötötte össze”).
- Külső programozású volt, a programokat dugaszolással vitték be a gépbe.
- Decimális számrendszerben dolgozott.
- Lyukkártyás bevittel (input) és kivitellel (output) működött.

Fontosabb egységek:

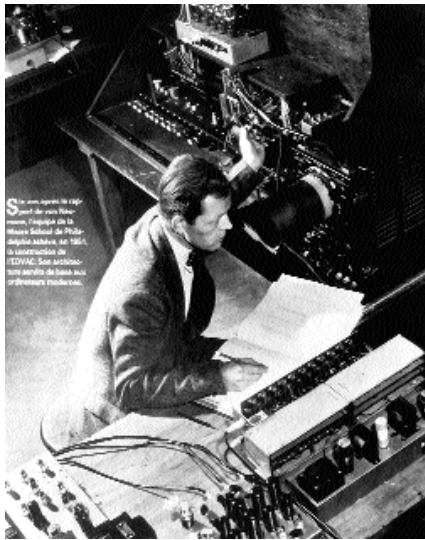
- 10 decimális jegyet + előjelet befogadni képes 20 akkumulátor (összeadó egység)
- 1 szorzóegység
- 1 osztó/gyökvonó egység
- Inicializáló egység, szinkronizátor
- Központi programadó („Master programer” a programhurkok vezérlésére)
- 1 kártyaolvasó, 1 kártyalyukasztó

Műszaki adatai:

- Műveleti idők: összeadás 200 msec, szorzás 2,8 msec, osztás 2,4 msec
- 125 kártyát olvasott be, illetve 100 kártyát lyukasztott percenként
- Energiafelhasználás 174 kW, alapterület 180 m²
- Hatásfok: 70%
- Kiszolgáló személyzet 6 fő/műszak

A gép egy évig működött Philadelphiában (Los Alamos részére végzett számításokat), majd Aberdeenbe telepítették át (itt 1955-ig használták).

Az új gép: az EDVAC



A Moore School megrendelést kapott a hadügyminisztériumtól egy új számítógép kifejlesztésére. Az új gép neve EDVAC (*Electronic Discrete Variable Computer* = Elektronikus diszkrét változójú számítógép).

6. ábra. Az EDVAC, a Moore School-ban tervezett második, de már tárolt programú, soros működésű elektronikus (elektroncsöves) számítógép. (1945)

A „Jelentés”

1944 szeptemberétől Neumann János aktív résztvevője volt az EDVAC projektnek.

1945. június 30-án Neumann Jánostól egy 101 oldalas dokumentum érkezett a Moore Schoolba, Los Alamosból. A címe a következő volt: „First Draft of a Report on the EDVAC” („Az EDVAC-ról szóló jelentés első változata”, a továbbiakban röviden „First Draft...”-nak vagy Jelentésnek hívjuk). Ebben a dokumentumban Neumann János az EDVAC gép teljes leírását adja, összefoglalva mindazon gondolatokat, amelyek Moore Schoolban jöttek létre az új géppel kapcsolatban. Az anyagban leírta a gép logikáját, a gép felépítését. Neumann a gépet gyors működésű, automatikus, digitális számítórendszernek nevezte.

Neumann János elképzelését jelentősen befolyásolta Warren McCulloch neuropszichológus és Walter Pitts matematikus munkássága, akik 1943-ban egy dolgozatot publikáltak, amelyben az emberi agy működését, illetve az agy pszichológiáját úgy írták le, mint egy, az agy által elvégzett logikai kalkulust. Neumann János az elkészített dokumentumban erre hivatkozott, felhasználta McCulloch és Pitts elképzeléseit. Vizsgálta a neuronok és az emberi idegrendszer egyéb területeinek működését, és összehasonlította az automatikus számítógéppel. Ezzel az első lépést tette meg a számítógép „antropomorfizálása” terén; ami önmagában is nagyon jelentős.

Csak Neumann neve szerepel a Jelentésen. Jogos-e?

Goldstine sokszorosította a dokumentumot, szétosztotta Moore Schoolban, majd elküldte a másolatokat sok más, a dolog iránt Amerikában és Nagy-Britanniában érdeklődő tudósnak. A Moore Schoolban, Eckertnél és Mauchly-nál nagy elégedetlenséget váltott ki, amikor kiderült,

hogy milyen széles körben terjedt el a dokumentum. Nagyon haragudtak Golstine-re és voltaképpen Neumannra is. Ennek első oka az volt, hogy a munka szigorúan titkos minősítéssel folyt, és azzal, hogy a „First Draft...”-ot nyilvánosan osztogatták, megszegették a biztonsági előírásokat. A második ok az volt, hogy Mauchly és Eckert szabadalomért folyamodtak az EDVAC-ügyben, és nyilvánvalóan nem volt kétségük afelől, hogy ezt meg is kapják. Attól féltek, hogy a publikáció csökkenti annak esélyét, hogy szabadalmaztathatják az EDVAC-ot.

Az elkeseredettség harmadik, talán a legfontosabb oka az volt, hogy a Neumann-féle dolgozat lényegében semmilyen hivatkozást nem tartalmazott a Moore Schoolban működő többi kutatóval kapcsolatban. Eckert és Mauchly nem tartották helyénvalónak, hogy az általuk elért eredmények Neumann nevéhez fűződnek majd, noha azok némelyike tényleg nem volt Neumannhoz köthető. Tudták, hogy a csoportos kutatásoknál az eredményt a legismertebb névhez kötik, ráadásul annak a nevéhez, aki legelőször publikálja azt.

Eckertnek nem volt lehetősége publikálni a dolgokat, mert az EDVAC gép katonai titkot jelentett. Semmi sem mutat arra, hogy Neumann el akarta orozni mások munkáját, hiszen a dokumentumra ráírta, hogy „First Draft...”, amivel jelezni akarta, hogy a dolgozatot szét kell osztani a többi kutató között, abból a célból, hogy megtegyék a megjegyzéseiket, korrekciójukat, mielőtt a végleges dolgozat létrejön. Goldstine akciója azonban végleges helyzetet teremtett, amelyen már nem lehetett segíteni.

Neumann János a „számítógép atyja”?

Többen azt mondják, akik a számítástechnika korai történetét feldolgozták, hogy a tárolt programú számítógép ötlete voltaképpen nem Neumann Jánostól származik, mert egy ilyen elven működő gép létrehozásának lehetősége a Moore Schoolban már azelőtt felvetődött, mielőtt Neumann János a színre lépett. Eckert már 6 hónappal azelőtt írt erről egy emlékeztetőt, amikor Neumann János egyáltalán nem hallott még a Moore School-projektről. Goldstine ezzel szemben a következőképpen védi meg Neumann Jánost: „A jelentés mesteri analízisét és szintézisét adja mindazon elgondolásoknak, amelyek az EDVAC-kal kapcsolatban 1944 ősztől 1945 tavaszáig megszülettek. Nem minden gondolat származik tőle, de a jelentősek igen. Nyilvánvaló, hogy Neumann János megírva a „Jelentést”, oly módon kristályosította ki a számítógéppel kapcsolatos gondolkodásmódot, ahogy senki más ezt előtte nem tette meg.”

Néhány történetíró azt állítja, hogy a Jelentés címlapján több szerző nevére is hagytak helyet. Tényleg igaz, hogy két címlap készült. Az elsőn, amelyen csak a Jelentés címe és Neumann neve szerepelt, volt hely több szerző részére is. A második, végleges címlapon azonban semmi nem mutatott arra, hogy Neumann János neve mellé más neveknek is kellett volna kerülniük.

„Neumann János hozzájárulása az EDVAC projekthez valóságos és fontos volt, és általában is az első számítógép létrehozásában játszott szerepe nem volt kicsi, de a Jelentés publikálása ezt beárnyékolta” – mondta Goldstine.

Az a körülmény, hogy a „First Draft...”-on csak Neumann neve szerepelt, és ezzel ő lett a tárolt programú számítógépek „atyja”, Moore Schoolban, az EDVAC projekt széteséséhez vezetett. Ehhez még hozzájöttek a szabadalmi problémák is, így együtt egy sor konfrontáció keletkezett Neumann és az EDVAC tervezői teamje között, aminek következtében sokan otthagyták a Moore Schoolt, és más egyetemi intézetekbe mentek át, vagy saját céget alapítottak. Végül is ez azt eredményezte, hogy az EDVAC további tervezése és fejlesztése lényegében

megállt. Egyébként Burks, aki szintén egyik résztvevője volt az EDVAC projektnek, a Neumann jelentéssel kapcsolatban a következőket írja. „Kétségtelen, hogy mások nevét is szerepeltetnie kellett volna. Véleményem szerint elsősorban P. Eckertet és J. Mauchly-t, H. Goldstine-t és engem is be kellett volna írni a jelentés szerzői közé.”

Egyes történetírók azzal kelnek Neumann János védelmére, hogy a jelentést kézírással küldte Moore Schoolba abból a célból, hogy a Hadügyminisztériummal kötött W-670-ORD-4926 számú szerződés teljesítéseként adják be, küldjék el a Pentagon azon osztályára, ahonnan a projektet megrendelték.

2. A First Draft

Az eddig elmondottakkal azt kívántuk bemutatni, hogyan került a korszak nagy (legnagyobb) matematikusa a számítógépek közelébe, hogyan jutott el a „First Draft...” (a „Jelentés”) megírásáig, ami miatt voltaképpen őt tekintik a „számítógépek atyjának”.

A „First Draft...” magyar nyelven még nem jelent meg. Mi nem tűzhetjük ki célul, hogy ezt pótoljuk, de az alábbiakban részletesen ismertetjük fő mondanivalóját. Mondatainkat több helyen akár pontos idézetnek is tekinthetjük, de nem tesszük idézőjelek közé.

Sok helyen megjegyzéseket fűzünk a Neumann-féle filozófiához.

E fejezet íróinak szerencséjük volt: a „First Draft...” keletkezésétől számított alig több mint egy évtizeddel „belecsöppentek” az első hazai számítógép építésének és programozásának munkálataiba, és ettől kezdve saját élményeik alapján növekvő csodálkozással figyelhették, mi lett a Neumann-elv alapján épülő EDVAC-ból. Lenyűgözően meredek fejlődési ív, megjósolhatatlan perspektívával.

Mielőtt rátérnénk a „First Draft...” ismertetésére, elemezzük, milyen előzmények vezettek ahhoz, hogy ez az összefoglaló mű létrejöhessen.

Közvetlen előzmények. Általános áttekintés

Annak érdekében, hogy a „First Draft...” megírásával kapcsolatban az olvasó Neumann János egyedülálló teljesítményét értékelni tudja, röviden összefoglaljuk az olvasó számára érdekes előzményeket, azokat, amelyekből a „First Draft...”-ban szereplő elvek, gondolatok születtek (vagy születhettek). Azt is szeretnénk, ha az olvasó megértené, hogy a „First Draft...” megírásakor Neumann János miért dolgozott teljesen egyedül, a munkatársai nélkül a dolgozaton (bár néhány részletében felhasználta a korábbi munkájukat), illetve miért indult el az EDVAC leírásával egy addig nem járt, a formális logikával jellemzett úton, ami a mai modern számítógépek megtervezésének a technikájához vezetett. Ebből az összefoglalóból – talán – még az is megérthető lesz, hogy miért nem volt szükség a „First Draft...”-on – annak ellenére, hogy a címében „vázlatnak” készült – további munkára és kiegészítésre.

Még valami. Neumann megjegyezte, azért írta meg ezt a jelentést, hogy *„hozzájáruljon a nagysebességű számítógépekkel kapcsolatos ismeretek további fejlődéséhez, továbbá a lehető legkorábban és legkiterjedtebben tárgyalja az e tárggyal kapcsolatos tudományos és műszaki elgondolásokat”*.

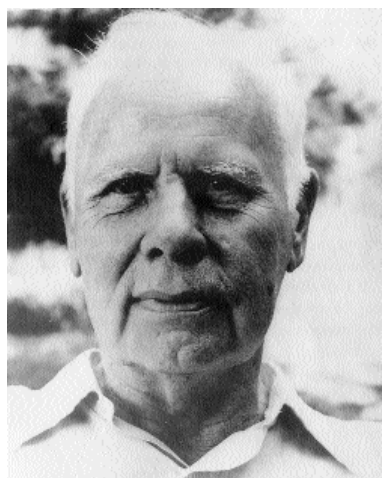
Neumann egyik munkatársa elmondta, hogy Neumann a számítógépet mindig is a tudomány közkincsévé kívánta tenni, azért volt ellene minden szabadalmaztatási kísérletnek,

hogy a tudományos világot ne korlátozza bizonyos dolgoknak egy néhány embernek a birtokában lévő tulajdonjoga. Azt is mondták, hogy a számítógépek fejlesztésének a korai szakaszában Neumann János nem gondolt arra, hogy egyszer majd a számítógépeket gyárakban fogják tömegével előállítani, az volt az elképzelése, hogy a számítógép egy olyan eszköz lesz, amiket különböző egyetemek és kutatóintézetek a maguk képeire és hasonlatosságára, a felhasználandó célra fogják egyedileg előállítani. Ebben az esetben mindenféle szabadalmaztatási eljárás valóban akadályozta volna a számítógépek gyors elterjedését.

A „First Draft...” tárgyalása előtt megvizsgáljuk az EDVAC-ot közvetlenül megelőző számolóberendezések logikai és műszaki felépítését, külön felhívva a figyelmet az azokból átvett megoldásokra és részegységekre, valamint az eltérésekre és az azonosságokra.

– Az EDVAC megtervezése előtt Amerikában szinte alig voltak mérnökök és matematikusok, akik a számolóberendezések tervezésében, a műveletek vezérlésében, egyáltalán a számítógépes elektronikus áramkörök kialakításában tapasztalatokkal rendelkeztek volna.

– Az említett elektromechanikus Hollerith (lyukkártyás) adatfeldolgozó gépek az első elektronikus számítógépek szempontjából azért voltak fontosak, mert John Mauchly és Presper Eckert, az ENIAC tervezői az ENIAC programozásának az áramköreit és módszerét ezektől a gépektől vették át. Már a Hollerith-gépeket is felhasználta Wallace J. Eckert tudományos – csillagászati – számítások elvégzésére, de a gép konstrukciója erre csak némi változtatásokkal volt alkalmas. Wallace J. Eckert munkásságának az óriási jelentőségét abban lehetne összefoglalni, hogy az IBM figyelme az adatfeldolgozás irányából a tudományos számítások felé fordult.



7. ábra. George R. Stibitz, a Bell Laboratórium elektromechanikus, részben automatikus számítógépének, a Model V. jelfogós gépnek a tervezője. (1937)

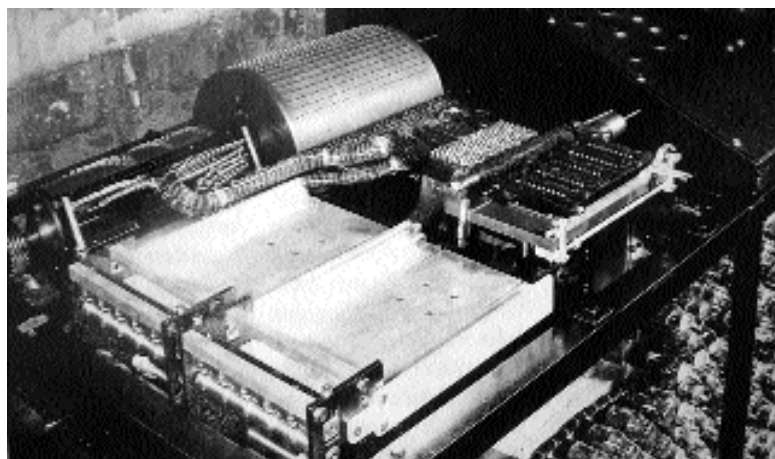
– A számítások gépesítése szempontjából ugyancsak óriási jelentősége volt – a harmincas évek végén megépült – két jelfogós számítógépnek, az egyiket – a MARK I-et – Howard H. Aiken vezetésével a Harvard egyetem és az IBM közösen fejlesztette, a másikat pedig – a Model I–V-öt, a *részben automatikus számológépet* – a Bell Laboratórium, George R. Stibitz irányításával építették meg. Ezek a relés számológépek tulajdonképpen Charles Babbage mechanikus építésű, analitikus gépét másolták le, így – noha az ENIAC-ba szinte semmiféle megoldásuk nem került át – mégis az ENIAC közvetlen „rokonának” voltak tekinthetők.

– Valójában a jelfogós számítógép-technikának már nem volt ideje kifejlődni, mert alig két-három év múlva a számítástechnikában megjelent az elektroncső, ami több nagyságrenddel gyorsabb kapcsolóelem volt, így – az éppen megszületett jelfogós technikát – gyorsan használaton kívül helyezte. Egyébként Aiken egy 1937-es kiadatlan tanulmányában már céloz a számítógépnek az egyik legfontosabb jellemzőjére: „...*(a gép) egy számítást a matematikai műveletek természetes sorrendjében végezzen el*”. Ezt a megjegyzést, közvetve – talán belemagyarázással – úgy is lehetne értelmezni, hogy a gép a matematikai műveletek programját automatikusan hajtja végre.

Ezért írja azt Hermann H. Goldstine a jelfogós számológépekről: *...a Harvard Számítástechnikai Laboratóriumában található gépek önmagukban semmiképpen nem voltak a jövő előfutárai. Egy újszülött tudományág korai időszakában igen sok újszerű ötlet lát*

napvilágot, és ölt testet valamilyen szerkezet alakjában. A tudományos alkalmazás választóvize azután az alkalmatlanokat kiszűri; ez a technológiai analógiája a természetes kiválasztás darwini elméletének, amely szerint a legéletképesebb egyed marad meg. Amikor egy új, fontos gondolat első ízben napvilágot lát, egy sor módosulásra lehet számítani, és ez mindaddig folytatódik, amíg egy új, stabil helyzet nem kristályosodik ki.

– A számolóberendezések fejlődésében ez a módosulás – John Vincent Atanasoff fellépésével – viszonylag gyorsan bekövetkezett. 1937-ben vált világossá Atanasoff számára, hogy elektroncsövekkel – kettes számrendszerben – számolóáramköröket lehet építeni, több nagyságrenddel gyorsabbakat, mint azt elektromechanikus kapcsolóelemekkel bármikor is ki lehetett volna fejleszteni. Atanasoff feltalálta és alkalmazta a kapacitív dobtárolót, amivel – Clifford Berry-vel közösen – egy elektronikus számológépet épített, az ABC-t (Atanasoff Berry Computer), ami – elszigeteltsége miatt – csak közvetve hatott az elektronikus számítógépek tervezésére.



8. ábra. John Vincent Atanasoff, matematikus, az elektronikus számítógép feltalálója. Az első elektroncsöves számítógépet – az ABC-t (Atanasoff Berry Computer) – 1939-ben Clifford Berry fiatal villamosmérnökkel együtt építette meg.

– Amikor Atanasoff ötletét John W. Mauchly meghallotta, azonnal felkereste az Iowa State College-ben, ahol egy hét alatt megismerte az elektroncsöves számítóáramköröket és kikérdezte Atanasoff-ot további elképzeléseiről. Az eredmény Presper J. Eckert villamosmérnökkel közösen született meg: az ENIAC.

Az utókor érdekes vitákat generál az ENIAC-kal kapcsolatban. Az egyik kérdés, hogy az ENIAC, amikor megépült, egy gyors, elektronikus számológép (calculator, mint például a MARK I) volt-e, vagy pedig számítógép (computer), mint például az EDVAC, az EDSAC, illetve az IAS gép? A hazai, de a nemzetközi meghatározás szerint is, számológépnek nevezik azt a számolóberendezést, ami a matematikai és logikai műveleteket el tudja végezni, a programot egy külön tárolóban (például lyukkártyán, lyukszalagon vagy más tárolóeszközön stb.) őrzik, onnan utasításonként olvassák be és hajtják végre. Ha ezt a meghatározást elfogadjuk, akkor az ENIAC *számológép és nem számítógép volt.*

– Az ENIAC tervéről az a Nemzetvédelmi Kutatási Bizottság azt mondta, hogy az nem más, mint a *gyors számológép* program leírásának az elektronikus másolata. Számos szakértő – többek között Goldstine – szerint is az ENIAC valóban egy mechanikus számológépnek volt az elektronikus másolata. Felfedezhetők voltak benne az elektronikusan megépített „tízfogú számológerek” (gyűrűs számlálók), az átvitelképzés, minden akkori mechanikus számológépelemnek az elektronikus megfelelője.

– Goldstine az ENIAC leírásában erről így ír:

Az aritmetikai egységek, az akkumulátorok működésükben nem különböztek túlságosan a kor elektromechanikus, digitális gépeitől. Ezek az elektromechanikus készülékek számlálókerekeket tartalmaztak, amelyek egy elektromos jel hatására egy fokkal elfordultak. Az elektronikus gyűrűs számlálók ezekkel a kerekkel analóg módon működtek. (...) Az ENIAC-ban egy-egy számláló szoroson kapcsolt flip-flopokból állt, amelyeket úgy kapcsoltak össze, hogy a számlálóban mindig csak egy flip-flop volt 1 állapotban, az összes többi 0-ban. Ha a számláló elektromos jelet észlelt, ez a flip-flop visszaállt 0 állapotba, a következő flip-flop pedig 1 állapotba billent át. A számlálót úgy lehetett nullázni, hogy egy előre megadott flip-flopot, amelyet első helyértéknek neveztek, 1 állapotba kellett állítani. Az ENIAC számlálói mind gyűrűs számlálók voltak, ami egyszerűen annyit jelent, hogy az első és az utolsó hely megfelelő összekapcsolása révén a számláló, ha az utolsó állapotban volt, és áramimpulzust kapott, ismét visszatért az első állapotba, azaz az utolsó hely után ismét az első hely következett. Működési elvében tehát egy gyűrűsszámláló nagyon hasonlított a számlálókerekekhez, csak sokkalta gyorsabb volt azoknál. (...)

...az ENIAC decimális gép volt, amely (tízes számrendszerbeli) 10 jegyű, előjeles számokkal tudott dolgozni, minden akkumulátor – azok az egységek, amelyek az összeadás és a kivonás műveletét végezték – 10 darab 10 fokozatú és egy 2 fokozatú gyűrűs számlálót tartalmazott, az utóbbit a szám előjelének a jelzésére. A 10 fokozatú számláló állapotai feleltek meg rendre a 0, 1, ... 9 számjegyeknek. Valamennyi akkumulátorban átviteli áramkörök kötötték össze a számlálókat úgy, hogy ha valamelyik számláló a 9. fokozatról visszaváltott a 0-ra, a következő, tőle balra elhelyezett számláló egy áramimpulzust kapott, jelezve, hogy a maradékátvitel megtörtént.

Pontosan úgy, mint a néhány korszakkal előbbi fogaskerekes – Schickard, Leibniz, Pascal számológépekben. Éppen ezért, az ENIAC-nál sokkal korszerűbb elveken épült fel a korábban megszületett ABC, Atanasoff gépe.

– Az ENIAC egy nagyon gyors, programozható számológép volt, a mechanikus számológépek analógiájára gyűrűs számlálókkal (elektronikus számkerekkel) tízes és nem kettes számrendszerben számolt, a gépet valójában dugaszoló táblákon, Hollerith-rendszerben lehetett programozni. Az ENIAC-ban nyoma sem volt azoknak a logikai áramköröknek, amelyek a modern tárolt programú számítógépeket – így először a technikatörténetben – az EDVAC-ot is jellemezték.

– Az ENIAC vezérlése – enyhén szólva – ugyancsak nagyon bonyolult volt. Ismét Hermann Goldstine-t idézzük:

...hogyan lehetett az ENIAC-ot egy adott feladat elvégzésére utasítani. Ez igen bonyolult vállalkozás volt, és ez a bonyolultság volt az egyik oka annak, hogy a gépből nem készült több. (...) A legtöbb egység önálló programvezérléssel (program control) rendelkezett, amely meg tudta jegyezni, hogy az egységnek egy adott műveletet, például egy összeadást el kell végeznie, és jelezni is tudta, hogy ezzel végzett. Általában több programvezérlésre is szükség volt egy aritmetikai művelet elvégzéséhez. Ha például az 1. akkumulátorban tárolt számot hozzá kellett adni a 2. akkumulátorban tárolthoz, az 1. vezérlést utasítani kellett, hogy vigye át az akkumulátor tartalmát, a 2.-ét pedig, hogy fogadja be. (Zárójelben megjegyezzük, hogy a két akkumulátor között összeköttetést is kellett létesíteni, hogy a számokat valóban át lehessen vinni.)

Ezért mondta Goldstine az egyik előadásában, hogy az ENIAC – talán – legnagyobb hibája a programozás megoldatlansága volt, a programozók egy feladat programját körülbelül három hétig írták, míg a gépen a feladat három perc alatt futott le.

Ennek ellenére, az újzélandi L. J. Comrie (1893–1950) matematikus és csillagász, aki először használta a „tekerős” mechanikus számológépeket és a Hollerith adatfeldolgozó gépeket csillagászati számításokra, az ENIAC-ot látva, nagyon költőien írja le a benyomását:

Tik ... tak ... tik ... tak ...

Három másodperc ... ezer szorzást már elvégzett a gép. Az óra tovább ketyeg ... Egy órán belül egy millióval végez.

Így talán érthető, hogy Neumann János az EDVAC tervezésénél se a műveletvégző egység, sem a vezérlések megtervezésénél az ENIAC tapasztalatait egyáltalán nem használta fel, szinte mindent maga talált ki. Annak egyelőre nincs nyoma, hogy az EDVAC-megoldások kinek jutottak még az eszébe, a munkatársainak-e vagy pedig csak Neumann Jánosnak. Alapos okunk van feltételezni, hogy a „First Draft...” logikájának a megoldásai, különösen a 6–12. fejezetben leírt áramkörök Neumann János fejében születtek. Akkor viszont miért szerepeltetett volna a dolgozatban társszerzőket?

– Az ENIAC alkotóit nem csak az ENIAC vezérlésnek a bonyolultsága, de az ezzel szoros kapcsolatban lévő programozási problémák is izgatták. Goldstine ezzel kapcsolatos korabeli megjegyzései:

1944-re tehát a Moore Intézetben működő csoport tagjai körében általánosan ismertté vált, hogy egy digitális gép számára az utasításokat numerikus formában lehet szalagon tárolni. (...)

Különösen engem (...) foglalkoztatott sokat az ENIAC programozásához használt eszköz (a dugaszolható programtábla. KGy) esetlensége és a tárolóregiszterek kis száma. Lelki szemeink előtt már a ballisztikai egyenletek megoldásánál jóval bonyolultabb számítások elvégzése lebegett.

...a Moore Intézetben sokat beszélgettünk arról, hogyan lehetne az ENIAC működési hiányosságait kijavítani. Ismeretes egy Eckerttől és Mauchlytól származó keltezetlen jelentés, amely egy késleltető művonalnak nevezett, a gép tárkapacitásának bővítésére szolgáló eszközt ír le. (...) A leírt eszköz (...) kulcsfontosságúnak bizonyult a fejlődés következő fázisa során.

Mindez arra utal, hogy a Moore Intézetben folyó elméleti gondolkodás 1944 augusztusának végére már meglehetősen előrehaladott állapotban volt. (...) Ekkor hozakodott elő Eckert azzal a gondolattal, hogy a késleltető művonalat fel lehetne használni információk tárolására. (...) Különös szerencse, hogy éppen, amikor az ötlet fölmerült, lépett színre Neumann János. (...)

John Grist Brainerd, a Moore Intézet igazgatója így ír:

Lehetetlen az ENIAC tárkapacitását olyan mértékben növelni, ... ami a gyakorlati célú nemlineáris parciális differenciálegyenletek kezeléséhez szükséges volna. (...) Jelenleg két olyan elméleti elképzelést ismerünk, amelyet kiindulópontként lehetne kezelni. Az egyik ikonoszkóp csövek esetleges alkalmazása, amelyre nézve dr. Neumann megbeszéléseket folytatott dr. Zworykinnal az RCA Kutató Laboratóriumából, a másik pedig a késleltető művonal felhasználása információk tárolására, amellyel kapcsolatban már vannak tapasztalataim.

Ismét Goldstine:

– Az új gép logikai tervén való munkálkodás pontosan Neumann ízlése szerint való volt. E tevékenység során a formális logika terén folytatott korábbi munkássága döntő szerephez jutott. Mielőtt Neumann színre lépett volna, a Moore Intézetben dolgozó csoport elsősorban a valóban súlyos technológiai problémákra összpontosította a figyelmét, ő viszont a megérkezése után átvette a vezető szerepet a logikai problémák terén. Megkezdődött ily módon a csoport kettéoszlása technológusokra – Eckert és Mauchly – és logikával foglalkozókra – Neumann,

Burks és jómagam. Ez természetes munkamegosztás volt, az ellentétek azonban az idők múltával egyre növekedtek, és végül a csoport kettészakadásához vezettek.

– Így biztosra vehető, hogy a modern számoló és főleg vezérlő, elektronikus áramköröknek a leírását először Neumann János tette közzé a szóban forgó, EDVAC-dolgozatban, a First Draft írásába – megkockáztatható – valószínűleg azért sem vonta bele a társakat: Mauchly-t, Eckert-et, Burks-öt, de még Goldstine-t sem, mert attól félhetett, hogy Mauchly és Eckert az ENIAC irányába vinnék el az EDVAC terveit (ez persze csak spekuláció), még az is feltételezhető, hogy a logikusan felépített gondolatait megzavarták volna a többiek részéről – írás közben – elhangzó hozzászólások.

– Neumann az egyetlen logikus és lehetséges megoldást választotta, az EDVAC terveit, a „First Draft...”-ot miután megbeszélte a munkatársakkal – Los Alamos-ban, az íróasztala mellett, a saját logikájának megfelelően írta meg. Egy ilyen tanulmányt közösen nem is lehetett volna elkészíteni. Azzal az utólagos véleménnyel nem nagyon lehet vitázni, hogy a dolgot Neumann – feltételezhetően – ideiglenes leírásnak szánta, a címében foglalt „Draft” minősítés – talán – éppen ezt mutatja.

– Meg kell jegyezni, hogy az EDVAC-ot megelőző számolóberendezések mindegyike az adatokat sorosan tárolta és a műveleteket sorosan hajtotta végre. Egy korabeli leírásban olvastuk, ennek az volt az oka, hogy a számológépek konstruktőrei a drága aktív elemek nagy részét (előbb jelfogók és kapcsológépek, majd pedig elektroncsövek, sőt tranzisztorok is) „kísérletként” a számológépekből. Úgy okoskodtak, ha egy bináris számológép harmincbites számokkal számol, akkor egy 30 bites műveletvégző áramkört kell megépíteni, amiben ezeket a hosszú számokat egy időben lehet összeadni. A harminc bites műveletvégző áramköröket – a korabeli konstruktőrök szerint – csak sok alkatrészrel lehetett volna megépíteni, így pl. a sok elektroncső miatt romlott volna az áramkörök biztonsága (sok alkatrész, sok hibaforrás), ezen kívül az áramkör sokba is került volna, nem beszélve arról, hogy hosszabban és bonyolultabban lehetett volna az áramköröket kialakítani. Ezért feltehető – amikor Neumann János az EDVAC-leírást készítette – a korabeli elvek szerint, már csak szokásból is, soros számítógép megépítésére gondolhatott.

– Neumann azért is a soros számítógép mellett tette le a voksot, mert akkor fedezték fel és vették alkalmazásba a soros számítógépeknek a már említett egyik legfontosabb elemét, a késleltető művonalat, amivel a legkönnyebben soros számítógépeket lehetett építeni. Ennek a fontos eszköznek a leírásához ismét Hermann Goldstine dolgozatát vesszük segítségül:

– ...a bennünket érdeklő típus az ún. ultrahangos késleltető művonal. (...) Ezek az ultrahangos eszközök úgy működnek, hogy a késleltetni kívánt elektromos jelet ultrahangjellé alakítják át; ezt valamilyen folyadékön keresztül vezetik, majd ismét visszaalakítják elektromos jellé. A késés abból a tényből fakad, hogy a jelek a folyadékön sokkal lassabban haladnak át, mint az elektromosság a vezetéken. Higanyban például egy ilyen jel sebessége 1450 m/sec, míg az elektromos jel sebessége a vezetéken a fény $3 \cdot 10^8$ m/sec sebességéhez áll közel. Ha tehát megfelelően választjuk meg a folyadékot tartalmazó edény hosszát, akkor előre meghatározott nagyságú késleltetés érhető el.

A késleltető művonal, pl. egy higanyal töltött cső, aminek a két végét egy-egy piezoelektromos kvarckristállyal zárták le. A kristály, ha elektromos feszültséget kap, megváltoztatja az alakját, illetve – mechanikus hatásra – a kristályban elektromos feszültség ébred. Ily módon a kristályra vezetett impulzusok hatására a kristály rezegni kezd, ez a folyadékban hanghullámokat kelt, amiket a kimeneti kristály – akár másodpercenként több milliószor – ismét elektromos impulzusokká alakít vissza. Egy 1,45 méter hosszú, higanyal töltött csőben 1 milliszekundumos késleltetést lehetett elérni.



9. ábra. A soros számítógépek memóriája, a késleltető művonal. A képen Isaac Auerbach kezében látható a UNIVAC-ban használt (1946/47) higanyos művonal.

Ha a kimenő és a bemenő oldalt összekötjük, akkor a csőbe vezetett hangminta periodikusan ismétlődik, amíg a veszteségek miatt a jel a csőből el nem tűnik. Ha az energiavesztés pótlására a visszavezetett jeleket felerősítik, akkor máris készen áll a késleltető művonal mint tároló, amiben a bevezetett információt – amíg az erősítés megvan – tárolni lehet.

Ha a jel, amit a bemenő oldalra vezettek, 0,5 mikroszekundum hosszú volt, akkor egy ilyen csőben akár ezer bináris számjegyet is tárolni lehetett.

– A késleltető művonalat az ENIAC-ban alkalmazott tárolóval összehasonlítva azonnal megérthető, hogy Neumann miért fogadta el azonnal az új találmányt, amit Eckert mutatott be az EDVAC tervezőinek, többek között Neumann-nak is. Az ENIAC tárolási kapacitása 20 szó volt, egy-egy bináris számjegy tárolására egy kettős triódával (két félelektroncsővel) épített flip-flopot alkalmaztak. Az EDVAC esetében egy késleltető művonal 1000 bitet tudott tárolni, a művonalon cirkuláltatott bitek energiavesztésének a pótlására elég volt tíznél kevesebb elektroncső, így a művonalas tárolóval a memóriához szükséges elektroncsövek számát – az ENIAC flip-flopos tárolójához viszonyítva – körülbelül a századrészére lehetett csökkenteni. Ezért azután Neumann a tervezésnél az ENIAC 20 szós tárkapacitásával szemben az EDVAC tárkapacitását minimálisan 2000 szóra választotta.

Egy érdekesség, amit éppen itt érdemes megjegyezni. A számítógépek építésénél a legkényesebb elem – az ENIAC-nál és az EDVAC-nál, de még a mai modern PC-nél is – a memória volt. A késleltető művonalak után megjelentek a tárolócsövek (Neumann „igazi” számítógépében, az IAS gépben is ezzel épült a memória), majd a ferritmemóriák, az utóbbiaknak az volt a legnagyobb baja, hogy a memórialapokat kézzel kellett fűzni, a technológia ezért nem volt automatizálható és miniatürizálható sem.

1969-ben az Intel alapítói – Gordon Moore és Bob Noyce – bejelentették az első félvezető alapú, integrált áramköri memóriát, a 3101-es, 64 bites RAM-ot, amiben 1 bit tárolása körülbelül 100-szor többbe került, mint a korabeli, széles körben használt ferritmemóriában. Az Intel piacpolitikáját a szakma értetlenül nézte, hogyan lehet egy ennyire drága termékkel egy kiforrott memóriatechnikával szemben versenyezni? Ez az Intel memória egyébként a valamikori ENIAC memóriára hasonlított, ugyanis egy-egy bitet egy-egy triódapár helyett egy-egy tranzisztorpárból kialakított flip-flop tárolta, amit a szilícium egykristályban tudtak „kifaragni”. Az új és nagyon drága terméket az Intel mégis a piacon tartotta, sőt vásárolták is, ha nem is nagy tömegben, mert a ferritmemóriával szemben a félvezetős RAM kicsi volt, kevés energiát fogyasztott és benne volt az automatizált tömeggyártás lehetősége. Nem kellett hosszú ideig várni, az Intel megalkotta az integrált áramköri tömeggyártási technológiát és a picinyke Intel memóriachip, a ferritmemóriát néhány perc alatt a padlóra küldte. Egy iparág eltűnt, amit felváltott egy másik – az integrált áramköri technika – ami a megjelenése óta uralja a piacot.

– Az EDVAC egyik legnagyobb különbsége – az ENIAC-hoz viszonyítva – az volt, hogy kettes számrendszerben számolt, így Neumann eleve elhagyta a gyűrűs számlálót, azaz az elektronikus fogaskereket. Ezzel a változtatással az egész gépnek az aritmetikai egysége, amit a tanulmány CA-nak nevez, megváltozott. A korabeli irodalomból kitűnik, hogy Neumann

valószínűleg nem ismerte John Vincent Atanasoff munkáját, csak később találkoztak, a találkozással kapcsolatban Atanasoff azt mondta: *Neumann volt az első szakember, akivel a számítógépekről értelmesen lehetett beszélgetni, és akitől sokat tanultam.* Atanasoff 5-6 évvel az EDVAC előtt leírta, sőt John Mauchly-nak el is mondta, hogy elektronikus számolóeszközökbe, amik kétállapotú elemekből – akkor elektroncsövekből – épülnek fel kizárólag a kettes számrendszerű aritmetikai műveleteket lehet beépíteni.

– Neumann úgy gondolta, hogy az EDVAC-ban elég 8 decimális számjegyből (27 bitből) álló számokkal műveleteket végezni, ez *elegendő* – mondta – *számos probléma megoldására.* Neumann tehát a megoldandó feladatokból vezette vissza a megoldásra kerülő feladatban az alapadatok nagyságát. Ennek ellenére – valószínűleg a konstruktőrök szerették a kerek számokat – több korabeli számítógép is 30 bites (+előjel) szavakkal számolt. 30 bites szavak esetében a bináris eredményt előbb nyolcas számrendszerre fordították le, majd utána a nyolcas számrendszerű eredményből számították ki – tízes rendszerben – a végeredményt. Később ez a módszer lényegesen egyszerűsödött.

– Neumann a Jelentésben (7–8. fejezet) részletesen sorba veszi a műveletvégző egységeket, az összeadó (kivonó) egységet, az előjelekkel való műveleteket, a bináris (tizedes) pont meghatározását, a szorzást (külön foglalkozik a szorzás pontosságával, hiszen, ha két 30 bites számot összeszorozunk, akkor egy 60 bites számot kapunk eredményül) és az osztást. Egyébként voltak olyan korabeli számítógépek, amelyek a szorzás, különben elvesző részét, egy külön regiszterben tárolták, ezt a regisztert az osztás pontosságának növelésére is fel tudták használni. Neumann – hasonlóképpen a szorzáshoz – az osztás elvégzésének is megadta a pontos leírását.

– A Jelentésben külön fejezet foglalkozik a bináris (a köznyelvben tizedes) ponttal. Ez azért is lényeges volt, mert a gép pozitív és negatív, egyenél kisebb abszolút értékű, fixpontos számokkal számolt, amiket azután számos első generációs számítógép – éppen a „First Draft...”-ból átvett. Az első hazai számítógép, az M-3 is 30+1 bites szavakkal dolgozott, fixpontos gép volt, az adattartománya pedig a fenti tartományba esett. Ez a korlát részben megkönnyítette, részben megnehezítette a bináris vessző, illetve az eredmény konverziója után a tizedes vessző helyének a meghatározását (9.0 fejezet).

– Neumann az egyéb aritmetikai műveletekkel is foglalkozott, mint például a négyzetgyökvonással, amit a legtöbb korai számítógépben egy szubrutinnal intéztek el, az utasításrendszer négyzetgyökvonást nem tartalmazott. Neumann erre is ad megoldást. Egyébként felteszi a kérdést, hogy milyen műveletek szükségesek a számítógépben? Az alapműveletek ugyanis az összeadás és a kivonás, a többi művelet – a szorzás, az osztás és a négyzetgyökvonás – már visszavezethetők az első két műveletre. De – mondja Neumann – miután a műveletek gátja a viszonylag kicsi memória volt, más útja is van a magasabb műveletek elvégzésének, például a beépített logaritmikus és aritmetikai – mondjuk – szorzótábla. Számos korai számítógéppel lehet találkozni – ilyen volt például Kozma László egyik antwerpeni gépe is, amelyikbe egyszeregytábla volt beépítve, egy másikba pedig egy Wheatstone-hidas (analóg) áramkör, amivel – jó pontossággal – meg lehetett becsülni a magasabb műveletek végeredményét (10.0 fejezet).

– A vázlat – már volt szó róla – nagyon sokat foglalkozik a memóriával, hiszen Neumann az EDVAC számítógéppel az ENIAC-nak több nagy problémáját is meg akarta oldani, ezek közül az egyik a memória elégtelensége volt. A másik a programozás lassúsága, ami ugyancsak a memória problémájára volt visszavezethető.

Az EDVAC leírás kétféle memóriát említ, a már ismertetett késleltető művonalat, illetve az *ikonoszkópot* (*speciális, impulzusokat tároló elektroncső*), ami a későbbi Neumann–Goldstine

IAS-fejlesztésekben főszerepet kapott. A késleltető művonal ugyanis meglehetősen bonyolult módon, főleg soros számítógépekben volt használható, míg az ikonoszkóp inkább – az akkor még nem létező, sőt elvetett – párhuzamosan működő számítógépekben, ehhez az architektúrához a korabeli memóriák közül éppen az ikonoszkóp-memória volt az ideális. Az nyilván a technikatörténetbe való belemagyarázás, hogy Neumann még az EDVAC-riportot írta, de már az IAS számítógépre gondolt, inkább az a valószínű, hogy a tervezéshez szükséges információk összeszedése alkalmával felmerült az ikonoszkóp-memória, amit az EDVAC nem használt. Abban a pillanatban azonban, amikor az IAS gép tervezése elkezdődött, Neumann-nak az ikonoszkóp az eszébe jutott, és a sokkal modernebb – párhuzamos működésű – IAS gépet ezzel a memóriával tervezte és építették meg.

– A „First Draft...” tanulmánynak talán a legnagyobb értéke, hogy Neumann ebben írja le pontosan a később „*Neumann elv*”-nek nevezett *tárolt program* elvet. A „Draft” szinte alig észrevehetően fogalmaz (14.0 fejezet): „Az utasítások az *M*-ből (memória) kerülnek a *CC*-be (central control, központi vezérlőegység), ugyanonnan, ahol a numerikus adatokat is tárolják.” Egyébként csak ennyi és nem több a tárolt program elve.

– Kiegészítésként azt kell még tudni, hogy minden Neumann előtti amerikai gép külön tárolta az adatokat, láttuk az ENIAC-nál flip-flopokban, a jelfogós gépekben jelfogókkal, amikből olyan sebességgel lehetett az adatokat megkapni, mint amilyen sebességgel a műveleteket végrehajtották. Az utasítások tára minden gépben – még Kozma László MESz-1-ében is – külön volt. Az ENIAC-nál dugaszolótablán, más gépekben lyukkártyán, lyukszalagon tárolták az utasításokat, azaz a programot, mert az utasításokat általában lassabban hajtották végre, mint a műveleteket.

Neumann János, amikor az ENIAC nagyon nehézkes programozásán töprengett, jött rá arra, hogy a két tárat, a programtárat és az adattárat egyesíteni kell, annál is inkább, mivel az utasításokat is számokkal kódolták, tehát egy utasítás megjelenésében pontosan olyan volt, mint egy adat és fordítva.

Hermann Goldstine egy hosszú beszélgetés alkalmával elmondta, hogy az elv kitalálása után – amire egyértelműen azt mondta: *Neumann János fejéből pattant ki az isteni szikra* – egymást túllicítálva kezdtek el hangosan gondolkodni. A következőket sorolták fel:

- Ha a közös tárbán együtt van az adat és a program, akkor az adatokat is lehet programnak nézni, és fordítva.

- Ha a programot lehet adatnak nézni, akkor azon műveleteket is lehet végrehajtani.

- Ha a programon műveleteket lehet végrehajtani, akkor egy programmal egy másik programból egy harmadikat lehet létrehozni.

- Ha ez így van, akkor egy számítógép a saját programját módosíthatja – ezt már nem mondták ki: a számítógép önmagát is fejlesztheti (evolúció).

– A tárolt program elve mind az EDVAC-nak, mind az utána következő számítógépeknek – azóta is – a „köbe vésett” törvénye, amin a konstruktőrök – bár számtalanszor megkísérelték – eddig sohasem tudtak túllépni.

– Neumann – ne tévesszük össze a két elnevezést, ezt az üzemmódot ugyanis néha sorosnak mondják, de nem az – a tanulmányban állást foglalt az utasításoknak az egymás utáni (soros vagy inkább sorban való) végrehajtása mellett. Az ENIAC ebben is más volt, ott megengedett volt a több utasításnak az időben egyidejű végrehajtása is.

– Még valamit az ENIAC-ról, ismét egy Hermann Goldstine-idézet:

...Neumann vitába szállt velem: vajon át lehetne-e alakítani az ENIAC-ot egy tárolt programú géppé. Azt javasolta, hogy az ENIAC két függvénytáblája közül az egyiket arra használjuk, hogy ebben tároljuk a program leírásához szükséges utasításokat, és az ENIAC-ba

egyszer s mindenkorra huzalozzuk bele ezeket az utasításokat. Még néhány más trükk segítségével fel lehetett használni az ENIAC különböző egységeit az utasítások végrehajtásának a vezérlésére.

Az ENIAC átalakításához több hónapig tartó fárasztó programozási munkára volt szükség. Minthogy Neumann nagyon türelmetlen volt, hogy mikor lehet az ENIAC-ot az új módon használni Los Alamosban, ezért 1947. június 7-én kinevezte a feleségemet a Laboratórium konzultánsává. Ebben az időben Neumann már így írt: „...nagyon elkötelezett vagyok Adele-nek a leveleiért. Nick és én már az ő új kódjával dolgozunk, és eddig kiválóan találtuk. (...)” Adele Goldstine, az általa kidolgozott rendszert továbbadta Richard Clippingernek, aki akkoriban a Ballisztikai Kutató Laboratórium Számítástechnikai Laboratóriumának a vezetője volt. (...) 1948. szeptember 16-án az új rendszer lefutott az ENIAC-on.” Ez a Goldstine megjegyzés annyit jelent, hogy az ENIAC is, 1947-ben – Neumann és nem Mauchly, valamint Eckert – javaslatára átalakult kvázi tárolt programú géppé, amit – a gép életének második felében – már így használtak.

– Még egy érdekes adalék a tárolt program elvének a kitalálásához.

Egy 1994-es látogatás alkalmával Konrad Zuse egy levelet küldött Hermann Goldstine-nek, akivel sohasem volt alkalma találkozni. A nagyon barátságos hangú levélben Zuse arra kérte Goldstine-t, hogy a publikációiban említse már meg: a tárolt program elvét nem Neumann János, hanem ő maga, Zuse találta ki. 1935 óta, amióta elhatározta, hogy számítógép-fejlesztő lesz, nem is épített más gépet, csak olyanokat, amikben a program és az adattár egybe volt építve. A választ Zuse nem tudta megvárni, mert – sajnos – hamarosan eltávozott az élők sorából.

Valóban Konrad Zuse már nagyon korán alkalmazta a tárolt program elvét, anélkül, hogy azt nagydobra verte volna. Nem is gondolt rá – mondta –, hogy az egy ilyen jelentős elvnek számít. A háború következtében ugyanis nem nagyon hallott az amerikai számítógép-fejlesztésekről. Konrad Zuse egyik tragédiája éppen a háború és a náci uralma volt, akik a német tudomány Németországban élő nagy alakjait elzárták a tudományos világtól. Zuse másik tragédiája az volt, hogy mint németet – noha nem volt náci – a háború után se a keleti, se a nyugati országok tudományos emberei nem nagyon fogadták be köreikbe.

Éppen ezért a tárolt program elvét helyes volna a továbbiakban Neumann–Zuse elvnek nevezni.

Goldstine az EDVAC megoldásainak a szerzőségéről összefoglalóan így ír:

„Vannak bizonyos részek, amelyek szerzősége nyilvánvalóan köthető egyik vagy másik emberhez... arról, hogy egy probléma megoldására föl lehetne használni az akusztikus tartályt, Pres Eckerttől hallottunk először. Más elgondolások esetében a helyzet bonyolultabb volt. Olyan bonyolult, hogy az, aki a gondolatot elsőként felvetette, saját magát beszélte le róla, kétszer vagy akár háromszor is megváltoztatva a véleményét. Sokszor nem az tette a javaslatot, akiben a gondolat elsőként felmerült. Az ilyen esetekben gyakorlatilag lehetetlen a szerzőséget megállapítani.”

– Egy másik idézet Paul Armertől, a számítógép-tudomány egyik vezető alakjától, amit a Datamation 1962. 8. számában tett közzé, amikor az alábbi „Burks–Goldstine–von Neumann: Preliminary Discussions of the Logical Design of an Electronic Computing Instrument” (1946. június 28.) című – az IAS számítógépről szóló – dolgozat megjelent. Ehhez írt Armer bevezetőt:

Ki találta fel tulajdonképpen a tárolt program elvét? Ez talán nem is különösebben érdekes másnak, csak a munkálatokban közvetlenül részt vevőknek, akik számára fontos lehet, hogy kit illet az elismerés. Az elv megszületett, és a későbbi generációk biztosan minden bizonnyal az emberi haladás egyik mérföldkövének fogják tekinteni... A legtöbb joggal az itt idézett dolgozat szerzői, valamint a Pennsylvaniami Egyetemen a J. Presper és John Mauchly

vezetésével működött munkacsoport tagjai követelik maguknak az elsőséget. Nem kétséges, hogy a találmányban másoknak is van részük, köztük nem utolsósorban Babbage-nek...

Az itt idézett dolgozat mindazonáltal a számítástechnika meghatározó jelentőségű dokumentuma. Nem csak a tárolt programú számítógépek tervezésének elveit adja meg, de számos rendkívül bonyolult problémát is előre lát, és zseniális javaslatokat tesz megoldásukra. A dolgozatban leírt gépet (amely IAS gép, Princetoni gép és Neumann gépneveken közismert) meg is építették, azután lemásolták (bár sohasem pontosan), majd a másolatokról is másolatokat készítettek...

E dolgozat megszületésekor az automatikus számolás elve szilárd alapokon állt (már a Harvardon megépített MARK I óta), és annak a hatalmas lépésnek a megtételére is sor került (az ENIAC megalkotásával), amelyet az elektronika alkalmazása jelentett. Annak az ugrásnak a horderejét, amelyet a tudományának akkori helyzetében a dolgozat részletei jelentettek, ma már nehéz tárgyilagosan felmérni...

– Neumann a „First Draft...”-ban szinte egyetlen szót sem szentel a programozásnak, sokkal inkább az EDVAC-nak – elsősorban – a logikai felépítésével foglalkozik. 1945-ben Neumann írt egy rendezőprogramot, de még ekkor sem foglalkoztak a programozásnak és a kódolásnak az alapos elemzésével. Még talán azt a megjegyzést is meg lehet kockáztatni, hogy a programozás – az összámtógépeknél – még nem volt tudomány, csak a hardver fejlesztése számított annak. Mára ez a vélemény alaposan megfordult.

– A programozás tudományának az első alapos elemzésére a „Preliminary Discussion...”-ban, az IAS géppel kapcsolatban került először sor. Ennek első „termése” a *folyamatábraszerű* programábrázolási mód, azt is szokták mondani, hogy mamagas szintű programozási nyelv volt, amit ugyanebben az évben Goldstine – Neumann közreműködésével – dolgozott ki.

Ettől kezdve Goldstine és Neumann számos, a programozással foglalkozó tanulmányt írt, ezek közül az egyik legfontosabb a „*Goldstine–von Neumann: Planning and Coding Problems for an Electronic Computing Instrument*” volt. Erről röviden a későbbiekben szólnunk.

A First Draft részletes ismertetése

Írásunk ezen része azoknak az olvasóknak készült, akik valamivel részletesebben is meg szeretnének ismerkedni a dolgozattal. A Jelentés 15 fejezetből áll. A fejezetek számozását, és a fejezeteken belüli pontok jelölését is az eredeti szerint pontosan követjük.

A Jelentés első hat fejezetét részletesen, szinte szó szerinti fordításban ismertetjük. Ezt azért tesszük, mert ebben a részben Neumann János voltaképpen az alapelveket írja le, egy „előzetes összefoglalót” ad.

A 7–15. fejezeteket kevésbé részletesen ismertetjük, ennek az oka, hogy ezekben a fejezetekben Neumann János „áramköri mélységben” írja le javaslatait, írásunk előírt terjedelme viszont nem teszi lehetővé, hogy ezt mi is megtegyük.

Megjegyzéseinket dőlt betűvel szedjük.

1.0 Definíciók

- 1.1. A Jelentés: nagyon nagy sebességű, automatikus, digitális számítási rendszerekkel (computing system) és ezek logikai vezérlésével foglalkozik.
- 1.2. Egy automatikus számítási rendszer olyan berendezés, amely utasítások végrehajtására képes abból a célból, hogy számításokat végezzen jelentős bonyolultságú problémák megoldására, például nemlineáris parciális differenciálegyenletek numerikus megoldása céljából.

Megjegyzés: ebben a pontban látszik, hogy Neumann a számítógépet teljességében matematikai gépnek tekintette, arra, hogy mire lesznek képesek ezek a gépek a harmadik évezred elejére, talán soha nem is gondolt.

Az utasításokat részletesen meg kell adni. Az utasításoknak tartalmazniuk kell minden numerikus információt, amely a probléma megoldásához szükséges, mégpedig a kezdő és peremfeltételeket, a fix paraméterek értékeit, sőt azokat a függvénytáblákat is, amelyek a probléma leírásában szerepelnek. Az utasításokat kártyán, lyukszalagon, mágnesezett acélszalagon vagy huzalon, filmen stb. kell megadni. A feladatok megoldására szolgáló eljárások megadására kódokat (utasításokat) kell használni, amelyekkel le lehet írni a megoldandó probléma logikai és algebrai lépéseit. A gépnek adott utasításoknak alkalmasnak kell lenniük arra, hogy – további emberi beavatkozás nélkül – teljességgel végrehajthatók legyenek. A műveletek végrehajtása után a gépnek fel kell tudnia jegyezni az eredményeket a fent említett formában. Az eredmények numerikus adatok..

Megjegyzés: Ezekből a mondatokból még jobban látszik, hogy Neumann a számítógépet például differenciálegyenletek megoldására szánta, hiszen neki akkor erre volt szüksége (az atombombával és a ballisztikai rakétákkal kapcsolatban is).

- 1.3. Nyilvánvaló, hogy a berendezésnek általában jóval több numerikus adatot kell produkálnia, mint amit végeredményként várunk. Ez azt jelenti, hogy a numerikus output csak egy része a ténylegesen képződő számoknak.
- 1.4. Természetesen feltesszük, hogy a gép automatikus működése közben hibátlanul végzi el a számításokat, hibátlanul hajtja végre az utasításokat. Az is nyilvánvaló, hogy egy berendezés hibátlan működése csak bizonyos valószínűséggel várható. Egy komplikált gép esetén és ráadásul egy hosszú műveletsorozat elvégzése közben nem reális feltételezni, hogy a hiba keletkezésének valószínűsége elhanyagolható. Abból a célból, hogy a hibákat felismerjük, és korrekciókat tudjunk végrehajtani, mindenképpen szükséges az emberi beavatkozás.

Bizonyos mértékig ezek a hibás működések elkerülhetők, a gép felismerheti a leggyakrabban előforduló hibákat (automatikusan) megjelölve, kiírva ezek létrejöttét és helyét egy kívülről látható jelsorozat segítségével, és megáll, vagy esetleg maga hajtja végre, mégpedig automatikusan, a korrekciókat és folytatja a működését.

Megjegyzés: Ez a probléma évek múlva is foglalkoztatja Neumann Jánost. Elméletileg vizsgálja azt, hogy lehet-e olyan automatát szerkeszteni, amelyik a saját hibáit „felismeri” és kijavítja. Egy érdekes matematikai tételt is kimond és bizonyít.

2.0 A rendszer főbb alkotóelemei

- 1.1. A gép működésének elemzése után kézenfekvőnek látszik, hogy a gépnek az alábbi fő alkotóelemekből kell állnia.
- 1.2. Mivel a létrehozandó berendezés elsősorban számológép, ezért az alapműveletek: összeadás (+), kivonás (-), szorzás (\times), osztás (\div) műveletének elvégzésére kell képesnek lennie, és erre speciális alegységekkel kell rendelkeznie.

Felvetődhet, hogy vajon szükséges-e mind a négy alapművelet, illetve a négy alapművelethez tartozó berendezés. Az is meggondolandó, hogy az olyan műveleteket, mint a $\sqrt{\quad}$, $\sqrt[3]{\quad}$, sgn, $||$, vagy \log_{10} , \log_2 , ln, sin stb. függvényeket és inverzeiket nem kell-e beépíteni. Lehet, hogy éppen szűkíteni kell a műveletek körét, elhagyva az osztást (\div) és akár a szorzást (\times) is. Felvetődhet, hogy sokkal rugalmasabb szerkezet szükséges, például olyan, amely függvénytáblák létrehozására képes, vagy éppen szukcesszív approximációs módszert valósít meg.

Megjegyzés: A szukcesszív approximációval (a fokozatos közelítés módszerével) sok matematikai feladatot lehet közelítőleg (tehát nem pontosan) megoldani, ezért kézenfekvő volt ennek a módszernek a gépbe való beépítése.

Mindenesetre leszögezhető, hogy a szerkezetnek rendelkeznie kell egy központi aritmetikai egységgel és ez alkotja a számológép első specifikus főelemét, amit CA-val (Central Arithmetic) jelölünk.

- 2.3. Szükség van a számológépben egy logikai vezérlőegységre, amely gondoskodik az egymás után következő műveletek végrehajtásáról. A létrehozandó berendezésnek eléggé rugalmasnak, vagyis (amennyire csak lehetséges) általános rendeltetésűnek kell lennie. Különbséget kell tenni a speciális utasítások (amelyekkel egy adott partikuláris problémát lehet megoldani) és az általános vezérlés között, amely végrehajtja az előbbi műveleteket. A vezérlőegységnek az utóbbi a feladata. A vezérlőegység tehát a második fontos eleme a számológépnek, jelöljük röviden CC-vel (Central Control, központi vezérlő).
- 2.4. A harmadik alapegység a memória.

a) Az osztási, illetve szorzási műveletek végrehajtása közben a részeredmények tárolásához mindenképpen szükség van memóriára. Kisebb mértékben ugyan, de az összeadás és a kivonás is igényel memóriát (az átvitelt több pozíción keresztül kell mozgatnunk), és különösen szükséges a memória a négyzetgyök vagy a köbgyök kiszámításánál.

b) Az utasítások is, amelyek a feladat megoldásához szükségesek, szintén egy memóriában tárolandók.

c) Számos (matematikai) probléma megoldásánál bizonyos speciális függvények fontos szerepet játszanak. Ezeket rendszerint táblázat alakjában adjuk meg, máskor pedig analitikus kifejezésekkel. Egyszerűbb és gyorsabb egy fix táblázatból kikeresni az értékeket, mint mindig újból kiszámítani, ahányszor csak szükség van rájuk. Rendszerint elegendő olyan táblázatokat használni, amelyekben 100–200 pontban adjuk meg a függvényértéket, és interpolációt célszerű használni.

Megjegyzés: Az interpolációs módszer lényege: adott pontokban ismert függvényértékekből egy speciális formulával számítjuk ki (közelítőleg) a függvénynek a közbülső pontokban az értékeit.

Lineáris vagy akár kvadratikus interpoláció sok esetben nem elegendő, célszerű inkább köbös vagy negyedrendű interpolációt, akár magasabbat használni. A 2.2.

pontban említett függvényeket hasonló módon kezelhetjük, tehát a függvények értékeit interpolációval számítjuk ki. Sőt a reciprokot (tehát az $\frac{1}{x}$ függvényt) is érdemes lehet ily módon kezelni, így az osztás műveletét szorzásra tudjuk visszavezetni.

Megjegyzés: *Mai szemmel nézve nagyon érdekes, hogy Neumann János ekkor még úgy gondolta, hogy a függvényeket táblázat alakjában kell bevinni, és nem direkt módon – bármely pontban – közelítő formulákkal kiszámítani. Ma a legegyszerűbb zsebszámológép is az összes elemi függvény (\sqrt{x} , $\sin x$, $\cos x$, e^x stb.) értékét egy adott pontban programmal számítja ki (ráadásul elég nagy pontossággal). Természetesen Neumann János jól ismerte azokat a módszereket (formulákat), amelyekkel az elemi függvényeknél „komplikáltabb” függvények értékei jó közelítéssel kiszámíthatók, de úgy látszik, nem gondolt arra, hogy az aritmetikai egység elég gyorsan boldogul a formulák (képletek) kiszámításával.*

d) Differenciálegyenletek megoldásánál a kezdeti, illetve a peremfeltételek nagy numerikus anyagot jelenthetnek, ezeket is meg kell jegyezni, tárolni kell, tehát ehhez is szükséges a memóriaegység.

e)–f) Ezekben a pontokban további differenciálegyenleteket elemez abból a szempontból, hogy megoldásuk folyamán milyen adatokat kell a memóriában megjegyezni.

g) Külön elemzi a szukcesszív approximációknál megjegyzendő adatokat; ezek számára is szükséges a memóriaegység, hiszen az approximációs lépések közbülső eredményeit ideiglenesen ugyan, de meg kell őrizni.

h) Bizonyos statisztikai kísérletekben előforduló rendezési problémákhoz is szükséges memória abból a célból, hogy rögzítsük azon adatokat, amelyekre a rendezést végezzük. Neumann János megemlíti, hogy ezek a nagyon gyors szerkezetek egy érdekes lehetőséget teremtenek majd a statisztikai feladatok megoldásánál.

2.5. Összefoglalva a harmadik egységgel kapcsolatos mondanivalónkat: a berendezéshez mindenképpen kívánatos egy jelentős nagyságú memória. A memória különböző részein természetükben különböző műveleteket kell elvégezni, és jelentősen különbözhetnek rendeltetésükben, mégis a teljes memóriát mint egy egységet kell tekintenünk, biztosítani kell a különböző részek közötti felcserélhetőséget.

Megjegyzés: *Neumann itt voltaképpen arról ír, hogy a memóriában nem kell külön helyet kijelölni az adatoknak, utasításoknak, az induló adatoknak (pl. kezdőfeltétel) és az eredményeknek (részeredményeknek) stb., ezek egységesen tárolhatók.*

A berendezésnek tehát egységes memóriát kell tartalmaznia, ami a harmadik specifikus része a számológépeknek; jelöljük ezt az egységet M-mel.

2.6. Ezek a specifikus (vagyis a rendszerre jellemző) egységek CA, CC (együtt C) és M voltaképpen megfeleltethetők az emberi szervezetben létező, az ember idegrendszerét alkotó asszociatív neuronoknak. Meg kell még vizsgálnunk, mi felel meg az érző (szenzor, afferens) és mi a motoros (mozgató, efferens) idegrostoknak. Voltaképpen az input, ill. az output felel meg a neuronrendszerben lévő szenzor-, illetve motoros elemeknek.

A numerikus (és más) információknak a C és M egységek közötti átvitelét olyan mechanizmusoknak kell megoldaniuk, amelyek ezen egységek részét képezik. Szükséges az is, hogy az eredeti, külső információt is be lehessen vinni a számítógépbe, és a végső információkat, tehát az eredményeket ki lehessen vinni a számítógépből a külvilág

számára. A külső információknak, vagyis a gépbe beviendő adatoknak a létrehozása direkt módon, emberi tevékenység útján történik, gépeléssel, lyukasztással, fotografikus fényimpulzusokkal, amit valamilyen billentyűkkel hozunk létre, vagy valamilyen fémszalag mágnesezése révén. Ezeket statikusan tároljuk.

A számítógépnek tehát rendelkeznie kell azon képességgel, hogy kezelni tudja a bevitelt és a kivitelt (az inputot és az outputot), mégpedig úgy, hogy bizonyos speciális médiumok olvasására, illetve ezekre való írásra legyen képes. Ezeket a médiumokat külső médiumoknak fogjuk hívni és R-rel jelöljük.

- 2.7. A berendezésnek rendelkeznie kell olyan egységekkel is, amelyek képesek a numerikus vagy más információkat átvinni R-ből C-be és M-be. Ezek az egységek a gép inputját képezik. Ezt tekintjük az számológép negyedik specifikus alegységének, és I-vel jelöljük. Jó megoldásnak az látszik, hogy az információkat R-ből (I-n keresztül) M-be visszük, és sohasem direkt módon a C-be.
- 2.8. A számítógépnek rendelkeznie kell egy olyan egységgel is, amely ki tudja vinni (a lényegében csak numerikus) információt a C-ből és az M-ből R-be. Ezek az egységek a gép outputját jelentik, és ez az ötödik alegysége a gépnek, amit O-val jelölünk. Úgy gondoljuk, hogy itt is az átvitelt M-ből O-n keresztül valósítjuk meg, közvetlenül R-be és soha nem C-ből.
- 2.9. Az output információt, amely R-re megy ki, úgy tekintjük, mint a szerkezet által produkált végeredményt, ami a megoldott probléma eredménye. Ezt jól meg kell különböztetni a közbülső eredményektől, amelyek végig M-ben maradnak (amikről a 2.4. pontban beszéltünk).

Ezen a ponton egy fontos kérdés merül fel. A gépben keletkező eredmények természete miatt mindenképpen szükséges, hogy a memóriának, az M-nek legyen olyan képessége is, amit az R át tud venni az M funkcióiból, mivel amellet, hogy az R többé-kevésbé emberileg érzékelhető információt tartalmaz, memóriatulajdonsággal is rendelkezik, és ebben a minőségében kiegészítheti M-et.

3.0. A tárgyalás menete

Ebben a fejezetben Neumann János az előbbi definíciókat felhasználva elmondja, hogy milyen módon fejt ki a továbbiakban az öt alapegységről szóló részletes javaslatait. A fejezetben belül a 3.3.-ban még egyszer visszatér a hibák felfedezésére, lokalizálására és a bizonyos feltételek mellett a lehetséges korrekciókra is; megemlíti, hogy ezt mindenképpen tárgyalni kellene. Azt is mondja, hogy nem tudja ezt a nagyon fontos tárgykört tárgyalni, bár megkísérli, legalábbis érinteni, ahol ez fontosnak látszik majd.

4.0. Elemek, szinkronizáció, neuronanalógia

- 4.1. Minden digitális számológép (gép) diszkrét (egymástól elkülönülő) egyensúlyi állapotokat felvevő, relé típusú elemeket tartalmaz.

Egy ilyen elem két- vagy többállapotú lehet. Ezek lehetnek teljesen egyensúlyi (stabil) állapotok, amelyekben az elem külső hatás nélkül is megmarad az adott

állapotban, miközben valamilyen megfelelő külső hatásra egyik egyensúlyi állapotból képes átmenni egy másik (egyensúlyi) állapotba.

Kétállapotú elem esetén egyik állapotban marad, ha nincs külső hatás, míg a másikba kerül, ha adott külső hatás éri.

Az eddigi számolóberendezések különféle mechanikus vagy elektronikus szerkezeti elemeket használtak: fogaskerekeket, amelyek 10 vagy több állapotot reprezentálnak úgy, hogy akár mechanikusan, akár elektromos impulzus útján más fogaskerekeket mozgatnak egy fokkal arrébb (új állapotba), vagy telefonreléket, amelyek elektromágneses hatással nyitnak vagy zárnak egy elektromos áramkört. Lehetséges az első kettő kombinációja is. Végül természetesen szóba jöhetnek elektroncsövek is, amelyek szintén alkalmasak adott állapotok reprezentálására.

Minden ilyen szerkezet autonóm módon ütemezi magát, a benne lévő elemek szukcesszív ütemezése révén. Lehet az ütemezést egy fix órával is végezni, amely előre definiált periódusokban ad ütemjeleket. Az ütemező lehet pl. egy forgó tengely, vagy egy kristályvezérlésű elektromos oszcillátor. Az elem fogalmát a fenti értelemben használjuk, és a berendezést szinkron vagy aszinkron típusúnak hívjuk attól függően, hogy az ütemezés órával vagy autonóm módon történik.

- 4.2. Megemlítjük, hogy a magasabb rendű élőlények neuronjai az előbbi értelemben elemeknek tekinthetők. „Minden vagy semmi” tulajdonsággal rendelkeznek, azaz két állapotuk van, nyugalmi és ingerületi. A neuronok teljesítik mindazon paramétereket, amit az előbb az úgynevezett elemekről elmondunk. Egy ingerületi állapotban lévő neuron standard hatásokat fogad be, tengelyfonalakon (axonokon) keresztül. Ilyen vonal azonban két különböző módon kapcsolható a következő neuronhoz. Egyfelől egy ingerületi szinapszissal úgy, hogy a hatás ingerületbe hozza a neuront. Másfelől egy úgynevezett inhibitor szinapszissal, amelynek olyan tulajdonsága van, hogy teljes mértékben megvédi az ingerülettől, bármelyik más neuronból is jöjjön az. A neuronnak határozott reakcióideje van (a hatás befogadása és a kimeneti hatás közötti idő), ez az úgynevezett szinaptikus késés.

Követve W. S. McCulloch és W. Pitts dolgozatát, amely 1943-ban jelent meg („*A logical calculus of the ideas immanent in nervous activity*”, Bull. Math. Biophysics Vol 5., (1943)), és ha ugyanúgy, mint ők, elhanyagoljuk a neuron működésének bonyolultabb aspektusait, az ingerküszöböt, az időleges összegezést, a gátlást stb., akkor meg lehet mutatni, hogy az egyszerűsített neuron működése teljességgel utánózható, imitálható egy telefonrelével vagy egy elektroncsővel. Az idegrendszer a szinaptikus késleltetést tekintve valószínűleg aszinkron működésű.

- 4.3. Világos, hogy a nagyon gyors számítási eszközökbe vákuumos elektroncsöveket kell beépíteni. Az elektroncsövekkel nagyon jó, mikroszekundum nagyságrendű reakcióidő (szinaptikus késleltetés) érhető el. Ez olyan teljesítmény, amelyet semmilyen más eszközzel nem lehet megközelíteni. Nyilvánvaló, hogy a tisztán mechanikus eszközök ilyen szempontból nem jönnek számításba, és a telefonrelék reakcióideje is 10 milliszekundum nagyságrendű, vagy talán ennél is nagyobb. Érdekes megjegyezni, hogy az emberi neuron szinaptikus késleltetése 10^{-3} , azaz 1 milliszekundum nagyságrendű.

A további vizsgálatainkban az elektroncsöveket tekintjük a szóban forgó számolóberendezések elemeinek. Megkíséreljük a csövek minden paraméterét elemezni, olyan elektroncsöveket felhasználva, amelyek a kereskedelemben kaphatók. (Ne használjunk nagyon bonyolult elektroncsöveket, és olyanokat sem, amelyek alapvetően

új funkciókkal rendelkeznek.) Végül megemlítjük, hogy a szinkronberendezések lényeges előnyökkel járnak.

5.0. Az aritmetikai műveletek fontosabb elvei, tulajdonságai

5.1. Vizsgáljuk meg a központi aritmetikai egységet, a CA-t.

Az elektroncső – mint egy áramszelep vagy *kapu* – egy „*minden vagy semmi*” típusú szerkezet (vagy legalábbis közelítőleg az), ugyanis a rácsra bocsátott feszültségtől függően vagy átengedi, vagy nem engedi át az áramot. Két triódából vagy egy pentódából ún. *trigger áramkörök* hozhatók létre, amelyeknek két egyensúlyi állapotuk van.

Mivel a számítógépekben az elektroncsöves áramköröknek kell kezelniük a számjegyekből álló számokat, célszerű olyan aritmetikát használni, amelyben a kezelendő számjegyeknek két értéke van. Ez azt jelenti, hogy a bináris számrendszert kell alkalmazni.

Megjegyzés: Láttuk, hogy a korábbi számológépek egy része tízes számrendszeren alapult; például az elektroncsöves ENIAC is (aminek az alapeleme a „számkeréknek” megfelelő, tízes, „elektronikus gyűrűs számolókerék” volt), míg a másik része kettes számrendszerben működött (például John Atanasoff elektroncsöves, ABC számítógépe). A tízes rendszerű elektronikus számológépek építése nyilván azzal függött össze, hogy a mechanikus számológépekben főleg fogaskerekeket (esetleg fogaslécet) használtak.

Neumann János – ahogyan korábban Atanasoff is – észrevette, hogy az elektroncső nagyon alkalmas két stabil állapot megvalósítására, így szinte teljesen természetes, hogy a kettes számrendszeren alapuló aritmetikát kell használni. Neumann John Atanasoff gépét és munkáját csak később ismerte meg, igen jó véleménnyel volt róla.

5.2. A bináris számrendszer alkalmazása jelentősen egyszerűsítheti a szorzási és osztási műveletet is. A bináris aritmetikának egyszerűbb logikai struktúrája van, mint minden másnak (például a tízes számrendszerbelinek).

Megemlítjük, hogy azon numerikus anyag, amit az ember direkt módon kezelni tud, természetesen tízes számrendszerben van felírva. Ez azt jelenti, hogy az R-ben, tehát az inputban és outputban szereplő számok decimálisak. De mindenképpen előnyös a tisztán bináris aritmetikát alkalmazni a CA-ban, és ugyanilyeneket a központi vezérlőegységben, a CC-ben is. Ebből viszont az következik, hogy az M-nek, tehát a memóriának, bináris számokat kell tartalmaznia.

Az előbbieket azt is jelentik, hogy szükségünk van decimális-bináris, illetve bináris-decimális konverzióra, amelyeknek az I-ben (az input részben), illetve az O-ban (az output részben) kell szerepelniük. A konverziókhöz sok aritmetikai művelet szükséges, ezért ezeket a számításokat a leggazdaságosabb a CA-ban elvégezni, és nyilván szerepet kap az I-vel és az O-val kapcsolatban lévő vezérlőegység, a CC is. A konverzióban szereplő aritmetikai műveleteknek teljesen binárisoknak kell lenniük.

5.3. Ezen a ponton egy másik elvi kérdés is felvetődik. Minden olyan létező számológépben, ahol az elem nem elektroncső, az elem reakcióideje elegendően hosszú ahhoz, hogy bizonyos „*teleszkópolást*” lehessen a lépések tekintetében megtenni, mind az összeadási, mind a kivonási, de még inkább a szorzási, illetve osztási műveleteknél.

Elemezzük a dolgot a bináris szorzásnál. Az ésszerű pontosság sok differenciálegyenlet megoldásánál mintegy 8 decimális számjegy, azaz a kerekítési hiba 10–8 alatt van. Ez 2^{-27} -nek felel meg bináris számrendszerben, azaz 27 bináris számjeggyel kell ábrázolnunk a számokat. Így a szorzásnál 27 jegyű a szorzandó, 27 jegyű a szorzó, és a szorzásnál a feladat a 0 és az 1 számjegyek szorzását jelenti, majd utána pozicionálásuk és kombinálásuk szükséges. Ez összesen lényegében 27^2 , azaz 729 lépést jelent, a bináris szorzatoknak az összegyűjtése, valamint kombinálásuk az időt megkettőzheti. Így mintegy 1000–1500 (elemi) lépés szükséges két 27 jegyű bináris szám összeszorzásánál.

Nyilvánvaló, hogy decimális számrendszerben jóval kevesebb, 8^2 , azaz 64 lépés szükséges a szorzás elvégzéséhez, ami szintén duplázandó, tehát körülbelül 100 a lépések száma. Azonban ez a kis szám azzal jár, hogy egy szorzótáblát kell alkalmaznunk, vagy más módon komplikálna a berendezés. Ezen az áron viszont beépíthetünk a gépbe több direkt bináris egységet. A továbbiakban a decimális műveletekkel éppen ezért nem foglalkozunk.

- 5.4. Az 1000–1500 lépésből álló szorzási művelet elvégzése minden nem elektroncsöves berendezésben elfogadhatatlanul lassú. Minden ilyen szerkezetben – kivéve a legújabb speciális reléekkel működőkben – a reakcióidő több mint 10 milliszekundum. Ezekben a gépeken 10–15 másodpercig tartana a két 8 decimális jegyből álló szám szorzása, míg ez 10 másodperc a gyors asztali kalkulátorokban és 6 másodperc a standard IBM-szorzókon.

Hogy elkerüljük ezt a hosszú időt, a műveletek (lépések) teleszkopizálására van szükség, azaz annyi műveletet (lépést) kell elvégezni egyidejűleg, vagyis szimultán módon, amennyit csak lehet. Ez nem egyszerű, de például az összeadásnál és a kivonásnál a megfelelő (az azonos pozícióban álló) számjegyek lépésenként egyszerre kezelhetők, és az átviteli számjegy is egyszerre kezelhető a soron következő pozícióban lévő két számjeggyel. A szorzásnál a részszorzatok szimultán módon képezhetők és pozicionálhatók, hiszen a bináris számrendszerben egy ilyen részszorzat vagy maga a szorzandó (mert 1-gyel szoroztunk), vagy 0. A részösszegek összeadásánál az említett módszer szerint járunk el. A szorzásnál az összeadást gyorsíthatjuk a részösszegek külön, páronkénti összeadásával.

Természetesen, ha a műveletek időtartamát két összeadás egyszerre történő elvégzésével akarjuk felezni, kétszer annyi összeadó szükséges.

A nem elektroncsövekkel működő gépeknél a fenti módszer alkalmazása teljesen indokolt a műveleti idők csökkentése céljából, hiszen itt ez nagyon fontos; és ebben nagy mérnöki tapasztalat halmozódott fel.

Egy valóban univerzális automatikus számolóberendezés, amely a fenti elvek alapján működik, több mint 10 000 elemből áll.

- 5.5. Úgy tűnik, hogy egy elektroncsöves berendezésnél a fentiekkel ellentétes eljárás sokkal ígéretesebb.

Mint a 4.3. pontban már mondtuk, egy nem túl bonyolult elektroncsőnél a reakcióidő egy mikroszekundumnál rövidebb lehet. Ilyen idő mellett egy nem manipulált szorzás elvégzéséhez szükséges időtartam elfogadható: 1000–1500 lépés, azaz 1–1,5 milliszekundum, ez jóval kevesebb, mint bármilyen nem elektroncsöves szerkezet esetén; így nem biztosítható az embernek az input-output végeken történő szinkron (ugyanolyan ütemű) módon való beavatkozási lehetősége (ütemkülönbség miatt).

(Neumann János itt a részletek tekintetében utal egy {} pontra, amit a First Draftban végül is nem dolgozott ki.)

Ami az összeadást, illetve kivonást illeti, ezek jóval gyorsabban végrehajthatók.

Két 27 bináris számjegyű összeadás esetén legfeljebb kétszer 27, azaz mintegy 30–50 lépés szükséges. Ez 0,03–0,05 milliszekundumot tesz ki.

Az osztási művelet időigénye kb. annyi, mint a szorzásé, és a gyökvonásé sem lényegesen több.

- 5.6. Ezen aritmetikai műveletek gyorsítása nem látszik szükségesnek; legalábbis addig nem, amíg teljességgel meg nem ismerjük ezeket a nagyon nagy sebességű szerkezeteket. Kérdéses, hogy jó megoldás-e a teleszkopikus módszerrel történő műveletgyorsítás az ehhez szükséges elemek számának többszörözése árán. A sokkal komplikáltabb (azaz a több elemből álló) elektroncsöves berendezés sokszor sokkal szelesebb tűrést igényelhet az ütemek tekintetében, azaz hosszabb reakcióidő lehet szükséges. A pontos kvantitatív hatásokat nehéz megbecsülni.

Úgy tűnik tehát, helyes az alábbi álláspont: a berendezésnek olyan egyszerűnek kell lennie, amilyen csak lehet, azaz a lehető legkevesebb elemből kell állnia. Ezt úgy tudjuk elérni, hogy nem végzünk el a gépben két műveletet szimultán módon, ha ez jelentősen megnövelné a szükséges elemek számát. A berendezés így megbízhatóbban működik, és az elektroncsövek sokkal gyorsabban (magasabb frekvenciával) hajthatók meg.

- 5.7. Az, hogy a fenti elv alkalmazását milyen határig erőltetjük, függ a mindenkori rendelkezésre álló elektroncsövek fizikai karakterisztikáitól. Előfordulhat, hogy az optimum nem a fenti elv 100%-os alkalmazásánál van, hanem valamilyen kompromisszumos megoldás szükséges. Ez mindig függ majd az elektroncső technika pillanatnyi állásától.

Megjegyezzük, hogy a nagy sebességű digitális számolóberendezésekkel kapcsolatos elképzelés, az erről való gondolkodás mostanáig az előbbiekkal ellenkező irányba mutatott; azaz eddig a szükséges elemek többszöröse árán is a teleszkopikus eljárásokkal való műveletgyorsítást részesített előnyben. Mindenesetre tanulságos volna – annyira, amennyire csak lehet – az ellentétes álláspontot is elemezni.

Megjegyzés: Az 5.3.–5.7. pontokat eléggé részletesen ismertettük. Ennek több oka is van. Az egyik az, hogy ezekből látszik, hogy Neumann János mennyire konkrétan, részletesen elemezte a problémákat. Nagyon sok dolgozatában elvégezte a numerikus számításokat. Az, hogy ez mennyire jellemző Neumann Jánosra, mutatja az alábbi szöveg is. (1947-ben írt dolgozatában szerepel.):

„Megbeszéléseket folytattam az IBM munkatársaival New Yorkban (590 Madison Avenue) a SSEC (Self-Sequencing Electronic Computer = Önműködő Elektronikus Számítógép) ügyében.

Úgy tűnik, hogy ez a gép ebben az esetben néhány hétre bérelhető. Az ára előreláthatólag 300–400 \$ egy ténylegesen felhasznált órára. Az árat illetően az alábbi mondható:

A gép két 14 jegyű decimális számot 20 msec alatt szoroz össze. Ezzel párhuzamosan 20 msec szükséges egy gépi utasítás előhozásához. Úgy gondolom, hogy egy szorzási művelet adminisztrálásához 3–4 utasítás szükséges. Így tehát célszerűnek látszik egy szorzásra 70 msec-ot vagy ellenőrzéssel együtt 140 msec-ot venni. Ez 7 szorzást jelent másodpercenként, azaz 25 000-et óránként. Óránként 350 \$ mellett 1,4 centbe kerül egy szorzás.

Egy ember esetén 10 decimális jegyű szorzás („Friden” vagy „Marchant” asztali számológéppel) 10 másodpercet vesz igénybe. A szorzást kísérő más műveletekre egy 4-

szeres, az ellenőrzésre egy 2-szeres és az emberi gyenge effektivitás, fáradékonyság miatt egy 2-szeres faktort célszerű venni. Ez 160 másodpercet, vagyis 3 percet tesz ki szorzásonként, azaz 20 szorzás esik egy órára.

Nagyon jó számítócsoportoktól szerzett ismereteim alapján ezek a számok nem pesszimisztikus becslésből adódnak. Ez 800 szorzást jelent egy 40 órás héten. 50 \$-t véve egy számolóember heti béreként és 2-szeres faktort az általános költségek miatt, 12,5 centet kapunk egy szorzásra.

Ez azt jelenti, hogy egy SSEC ezen árak mellett $12,5/1,4 = 9$ -szer olcsóbb, mint a számoló-ember.”

A továbbiakban Neumann János kiszámítja, hogy hány embert pótol a számítógép. Figyelembe véve, hogy a számítógép heti 40 órás működése alatt 30–50 százalékos produktív időt jelent, a számítógép 500 embert pótol.

Ezután („Más számítógépek” címmel) a kifejlesztés alatt álló újabb számítógépek teljesítményét és egyéb paramétereit is elemzi.

Úgy látja, hogy az új gépek sebessége 5–50-szer nagyobb lesz, mint a SSEC-gépe. Ezek naponta 16–24 órát üzemelnek, és 5, esetleg 7 napot egy héten. Produktivitásuk több mint 50%-os lesz. Ezekből a számokból azt következteti ki, hogy az új gépek 10 000–100 000 emberrel lesznek ekvivalensek. Neumann János előrejelzése szerint ilyen gépek 1–3 év múlva jelennek meg, digitálisak lesznek, 2000–4000 elektroncsövet tartalmaznak. (A SSEC 12 000 elektroncsövet és 20 000 elektronmechanikus relét tartalmazott, közli Neumann János ugyanitt.)

Elemzi az alkatrészek meghibásodási valószínűségét is. Nyolcóránkénti egy hibás művelet esetén műveletenként 10–12 hiba-valószínűség adódik. Megjegyzi, hogy a telefonrelék, 10–8–10–9 hiba-valószínűséggel működnek, az elektroncsövek ennél rosszabbul. Úgy látja, hogy a 10–12-es számot nem könnyű elérni.

A numerikus megoldási módszer kidolgozása és a stabilitás elemzése után Neumann János pontosan meghatározza a műveletszámokat és megvizsgálja, hogy a különböző számú osztópontok esetén mennyi gépidő szükséges a SSEC-gépen.

„Tehát... 170 000 szorzás tartozik egy feladathoz, azaz 210 emberhét vagy 7 óra a SSEC-gépen; 40%-os hatékonyságot feltételezve a gépen ez 18 óra, azaz 2,2 3 8 órás nap.”

Vagy másutt:

„Egy „nagy” feladat, amelyben az x osztópontok száma 25, a t osztópontoké 120, azaz $m = 25$, $n = 120$, tehát 3000 csomópont esetén

$$3000 \times 16 \text{ sec} = 48\,000 \text{ sec}, 13,3 \text{ órát igényel}”$$

Megjegyzés. Ezt a megjegyzésünket bátortalanul tesszük meg. Több mint 50 évvel a „First Draft...” megjelenése után, a jelenlegi hardverek ismeretében könnyű csodálkozni azon, hogy Neumann János „azon filozofálgat”, vajon érdemes-e elektroncsöves berendezésekben az elemi műveleteket (tehát a bitekkel végzett műveleteket) párhuzamosan végezni (teleszkopizálni), hiszen az input (output) oldalon álló ember úgysem tudja ilyen gyorsan táplálni a gépet. Igaz persze, hogy ha a műveletidők felezése miatt 5 m^3 helyett 10 m^3 -es gépet kellett volna építeni, az jelentős különbség. Ma a chipek világában kétszer annyi chip beépítése lényegében nem jelent problémát. Természetesen enyhén szólva eléggé „elnagyoltan” elemeztük a dolgot, de a lényegét akartuk kiemelni.

6.0. Az E-elemek

6.1. Az 5.0. pontban megvizsgáltuk azokat az alapelveket, amelyek szerint a CA-t működtetni kell. Vizsgáljuk meg most részletesebben technikai oldalról is.

Nyilvánvaló, hogy elemzésünket az alapelemek működésére kell alapoznunk.

Az ideális eljárás az elemek vizsgálatára: úgy tekinteni azokat, amilyenek valójában ti. elektroncsövek. Mivel nagyon sok alternatív lehetőség van az aritmetikai eljárások, logikai vezérlések stb. elrendezésére, ezért hasonlóan sok lehetőség van a különböző típusú, méretű elektroncsövek és más áramköri elemek használatára (például a gyakorlati teljesítmények alapján). Ez azonban nagyon megnehezíti a dolgunkat.

Éppen ezért vizsgálatunkat egy hipotetikus elemre alapozva végezzük, amely úgy működik, mint egy elektroncső – pl. mint egy trióda a hozzá megfelelően választott RLC áramkörrel – de önálló (izolált) egységként elemezhető anélkül, hogy rádiófrekvenciás elektromágneségi vizsgálatot is el kellene végeznünk.

6.2. Az emberi idegsejt, amit az előbbieken már tárgyaltunk, megfelel erre a célra. A továbbiakban ez képezi vizsgálatunk alapját. Pontosán definiálnunk kell azonban ezen elem jellemzőit.

A vizsgálandó elemet E-elemnek fogjuk hívni, és egy O körrel jelöljük. Az O elem ingerületi és inhibitor hatásokat vesz fel, és egy hozzá kapcsolódó tengelyen saját hatásokat bocsát ki; jelöljük ezt a következőképpen: $O-$. A tengely el is ágazhat, ezt így jelöljük: $O-<$, $O-<-$. Az emisszió a tengelyen egy szinaptikus késleltetéssel halad tovább; tegyük fel, hogy ez minden E-elemre ugyanakkora; jelöljük τ -val. A késleltetést a tengelyre rajzolt nyíllal jelöljük, így: $O\rightarrow-$, $O\rightarrow-<$. Ez a nyíl egyúttal a tengely indulópontját és irányát is jelzi.

6.3. Ebben a pontban a szerző elemzi az emberi idegrendszerben a szinaptikus késleltetés és a tengelyen a hatás terjedési sebessége közti különbséget, és összeveti ezt az elektroncsöves áramkörökkel. Megállapítja, hogy az emberi idegrendszerben a τ szinaptikus késleltetésnek nagy a szórása; az E-elemeknél viszont egységes, azonos τ késleltetés feltételezését javasolja abból a célból, hogy *szinkronizálni* lehessen a berendezés különféle egységeinek működését.

Ebből a célból egy központi órát javasol (legjobbnek egy elektromos oszcillátort gondol), amely minden x periódusban egy τ' hosszúságú impulzust bocsát ki.

Megjegyzés. A szerző két helyen is „ígéri” ({} hivatkozással), hogy később ezt részletezi, de ezt nem teszi meg.

6.4. Mindegyik E-elem egy előtte lévő elemtől vagy egy $-O\rightarrow-$ ingerületi szinapszison – $\bullet O\rightarrow-$, vagy egy inhibitor szinapszison keresztül veszi fel a hatást. Mint 4.2.-ben mondtuk, 1, 2, ill. 3 küszöbértékű E-elemekkel foglalkozunk, azaz olyanokkal, amelyek 1, 2, ill. 3 szimultán bejövő hatásra kerülnek ingerületi állapotba.

Használni fogunk dupla, azaz 2τ szinaptikus késleltetésű E-elemeket (jelöljük így: $-O\rightarrow\rightarrow-$) és keverteket is (jelöljük így: $-O\rightarrow-<\rightarrow-$). Így nagyobb flexibilitással csatolhatunk össze egyszerűbb struktúrákat, és ezeket elektroncsöves áramkörökkel meg tudjuk valósítani.

6.5. Ha megvizsgálunk néhány tipikus E-elemet, arra a következtetésre juthatunk, hogy a legtöbb ilyen elem 1-2 elektroncsővel megvalósítható.

7.0. A + és \times műveletek áramkörei

Megjegyzés. Ezt és a későbbi fejezeteket is az előbbiekhöz képest lényegesen kisebb részletességgel ismertetjük.

7.1. Főbb mondanivalók:

A valós számokat bináris számjegyekkel reprezentáljuk (30 bináris jegy elegendő).

A bináris számjegyeket a CA adott pontján és időpontban valamilyen hatás jelenléte (1) vagy hiánya (0) jelenti.

7.2. A különféle E-elemek ábrázolásához különféle *blokkjeleket* használ a szerző (\supset – az input, \bullet – az output).

7.3. A szerző felrajzolja és elemzi az összeadó áramkört.

7.4. A szorzási áramkör lényegesen különbözik az összeadásétól. (A szorzáshoz a részleteredményeket meg kell jegyezni.)

7.5. 7.5.–7.6. Az E-elemek memóriaként is használhatók.

7.6. 7.7.–7.8. A szerző részletesen leírja a szorzási művelet áramkörét.

8.0. A – és az 4 műveletek áramkörei

8.1. A kivonás műveletéhez szükség van az előjelre. Legyen ez a bal első bináris jegy (0 a –, 1 a + előjel).

8.2. Ebben a pontban a szerző megépíti (felrajzolja) a kivonás műveletének áramkörét.

9.0. A bináris pont

9.1. A bináris pont definiálása a szorzás és az osztás műveleténél feltétlenül szükséges.

9.2. Mivel két 30 jegyű szám szorzata 60 jegyű lesz, de az eredmény tárolására 30 jegy áll rendelkezésre, ezért 30 jegy elvész. Megoldódik a probléma, ha a bináris pontot az előjel után helyezzük el, vagyis ha a gépben csak –1 és 1 közé eső számok ábrázolását tesszük lehetővé.

9.2. A fentiek azt jelentik, hogy a számítási feladatot úgy kell transzformálni (2-i-vel való szorzással), hogy a beírt számok és a műveletek eredményei –1 és +1 közé essenek.

Megjegyzés. *Illusztrációképpen arra, hogy Neumann milyen részletességgel „tálatja” mondanivalóját, pontosan lefordítunk néhány mondatot.*

„a) Az összeadást és a kivonást nem lehet elvégezni, ha az eredmény nem –1 és +1 közé esik (hanem –2 és +2 közé).

b) Nem lehet az osztást elvégezni, ha az osztó kisebb (abszolút értékben), mint az osztandó.

Ha ezeket a szabályokat megszegjük, az összeadó, a kivonó, az osztó egység ugyan eredményt ad, de ez nem lesz az összeg, a különbség vagy a hányados.”

9.2.1. Ebben a pontban a szorzás és az osztás kerekítésével foglalkozik a szerző. (Bevezet egy „kerekítési szelep”-et.)

10.0. A gyökvonás áramköre

Más műveletek

10.1.–10.2. A szerző megmutatja, hogy a gyökvonás áramköre nem tér el lényegesen az osztás áramkörétől.

10.3. Miután megszerkesztettük az összeadás, kivonás, szorzás, osztás, gyökvonás műveletének áramkörét, vajon hogyan lehet ezeket integrálni a CA-ban?

Mielőtt erre válaszolnánk, vizsgáljuk meg, feltétlenül fontos-e minden fenti műveletet beépíteni a CA-ba.

Felvetődhet, hogy a logaritmustábla alkalmazásával a szorzás, osztás, gyökvonás műveletét vissza lehet vezetni összeadásra és kivonásra, de egy ilyen táblának legalább 109 értéket kellene tartalmaznia. A következtetés: a szorzás legyen az alapl műveletek között.

Az \div és a $\sqrt{\quad}$ műveletek közelítő formulák alkalmazásával visszavezethetők az összeadás, kivonás, szorzás műveletére. De ha a szorzás hosszú ideig tart, akkor ezek a műveletek is hosszú ideig tartanak.

Következtetés: a $+$, $-$, \times , \div , $\sqrt{\quad}$, műveleteket be kell építeni a CA-ba.

10.4. Milyen más műveleteket célszerű még a $+$, $-$, \times , \div , $\sqrt{\quad}$, műveleteken kívül beépíteni a CA-ba?

A köbgyököt nyilván nem érdemes. Szóba jöhetnek még a logaritmusfüggvény, a trigonometrikus függvény és inverzeik. Ezeket hatványsoraikkal lehetne kiszámítani, de ehhez a gépben meg kellene építeni a megfelelő logikai sémát. Ehelyett célszerűbb a függvénytáblázatok alkalmazása.

11.0. A CA felépítése

A műveletek teljes listája

11.1.–11.2. A CA általában két valós számmal végez műveletet. Az operandusokat az M memóriából kapja, ezek a CA két inputján I_{ca} -n és J_{ca} -n kerülnek be a CA-ba és az eredmény a CA outputján (O_{ca}) keresztül megy át az M memóriába. A memória különböző részei közötti kapcsolat megteremtése felesleges, ez a CA-n keresztül valósulhat meg.

A szerző részletesen leírja a memória és a CA közötti adatforgalmat.

11.3. Ebben a pontban a szerző felveti, hogy szükség van a következő műveletekre is:

- egy szám előjelének kiértékelése,
- két szám közötti reláció megállapítása,
- két lehetséges akció közül az egyik kiválasztására (azaz a számítási folyamatban az elágaztatásra),
- a számoknak kettes számrendszerből 10-esbe és fordítva való konvertálására.

A fentiekre további 5 műveletet javasol. A Neumann-féle gép 10 műveletet tud elvégezni, „utasításrendszere” (Neumann nem így hívja) 10 utasításból áll: +, −, ×, ÷, $\sqrt{\quad}$, i , j , s , bd , db , ahol i a I_{ca} -ból az O_{ca} -ba, j a J_{ca} -ból az O_{ca} -ba való közvetlen átvitel, s a szám előjelének kiértékelését, bd a binárisból decimálisba, db a decimálisból binárisba való konvertálás műveletét jelenti.

12.0. A memória kapacitása. Általános elvek

Megjegyzés. Neumann – az előző fejezetekből is látszik (7.4., 7.5., 7.6., 7.7., 8.3., 10.2., valamint a CA leírásában) – a tanulmányban nagyon sokszor visszatért a memória működésére, ugyanis éppen az EDVAC volt az első amerikai számítógép, amelyik a tárolt programelvének megfelelően egy memóriában tárolta mind az utasításokat, mind pedig a programokat. Ezért nem volt mindegy, hogy Neumann milyen memóriát választ ki, és mekkorára tervezi meg a memória tárolási kapacitását. Az sem volt közömbös, hogy mennyi idő alatt lehet és kell – az adott memóriában – elérni a memóriában tárolt adatokat.

- 12.1. Ebben a fejezetben írja le Neumann, hogy az EDVAC memóriájának egy soros, vagy késleltető elvű, ciklikus típusú memóriát választott, miután – az ENIAC után ez a memória látszott az EDVAC céljára – soros gépről lévén szó – a legalkalmasabbnak.
- 12.2. Valószínűleg Neumann előtt még senki sem gondolta végig – rendkívül érdekes módon ezt a „spekulációt” először nem egy villamosmérnök, hanem egy matematikus tette meg –, hogy milyen körülményeknek kell egy számítóeszköz memóriájának megfelelni, valamint azt sem, hogy az akkor ismert memóriáknak milyen előnyei és hátrányai vannak.

A memória kapacitásának az eszköznek azt a tulajdonságát nevezte, hogy hány bináris számot tud tárolni.

Ami az elemi kapacitás egységet illeti, Neumann már a 7.1.-es fejezetben meghatározta, hogy a gépnek előjeles, 30 bites (azaz 31 bites) bináris számokkal kell a műveleteket végeznie, tehát ilyen hosszúságú bináris számot (később az egységet szónak nevezték, Neumann idejében még nem).

Megjegyzés. Ez a 30-as szinte „kultikus” szám, ami 9 decimális számnak felel meg. Más leírások szerint is, a 30 bites „számolási egység” Neumann-tól ered, ugyanis – már korábban is volt erről szó – Neumann úgy becsülte, hogy körülbelül ez a nagyságrend már alig kezelhető mechanikus számolóeszközökkel, az ilyen pontosságú számításokhoz érdemes elektronikus gépeket használni. (Neumann „legendőnek látta”, ha egy számítógép 27 bites számokkal végez műveleteket, ezt – valószínűleg az ugyancsak ilyen méretű utasítás mérete miatt, később kerekítette fel 30+1 bitre.)

Neumann óriási tekintélyét mutatja, hogy egy néhány évvel későbbi számítógép-konstruktőr elmondta, a gépébe – szinte gondolkodás nélkül – ugyancsak 30 bináris számjegyből való számokkal tervezte meg a műveleteket, még csak nem is gondolt arra, hogy – például 32 bites számokkal számoljon. Eszébe sem jutott, hogy 32 bites aritmetikai egységet építsen, pedig a 32 bites (2 hatványa) bináris szám később nagy könnyebbségeket jelentett volna, a gépnek – például – a szövegfeldolgozásra való felhasználás esetén.

Neumann-nak, miután az EDVAC-ot tárolt programú gépnek tervezte, számolnia kellett a tárolandó utasítással is, ami nem lehetett hosszabb, mint amilyen hosszúak a tárolt adatok voltak. Ezért az EDVAC programozására háromcímű utasításokból álló utasításrendszert hozott létre, ami később zsákutcának bizonyult. Az EDVAC kísérletek alatt Neumann megvizsgálta a programozók szokásait és úgy találta, hogy a háromcímű utasítások bonyolultsága miatt a programozók – általában – az utasításban csak egy címet használtak, így a háromcímű utasításrendszerben rejlő lehetőségeket a programozók nem voltak képesek kihasználni.

Eppen ezért – az ENIAC és az EDVAC után – Neumann harmadik gépe, az IAS gép – noha ugyancsak 39+1 bites számokkal számolt – már egycímű utasításokkal épült, a 40 bitben azonban két egycímű utasítást tároltak. 10 bit a műveleti kód és 10 bit a cím. Ezzel az IAS gép vezette be először a későbbi modern számítógépekben – széles körben használt – egycímű utasításokat.

- 12.3. Ezek után Neumann részletesen megvizsgálta – a 2.4.-ben leírtaknak megfelelően – (a)-tól (h)-ig a különböző típusú műveletek memóriaigényét, a megoldás időszükségletét, ebből kívánt ugyanis következtetni az alkalmazott memória valóságos nagyságára és költségére.

Ebben a fejezetben is látható, hogy Neumann bizonyos részleteket csak jelzett {}, később – valószínűleg – vissza akart ezekre a témákra térni, az oda tartozó további fejtegetések részletesebb kidolgozására. A fejezetet olvasva az embernek a szerzővel kapcsolatban az jut az eszébe, hogy nem is egy matematikus, sokkal inkább egy mérnök „spekulál” így – egy-egy feladattal kapcsolatban – a memória időigényére. Ami egyébként igaz, a legenda szerint Neumann mint vegyészmérnök szokott rá az illetén való mérnöki spekulációra és számításokra. Persze – más fejezeteket (ahol például a formális logikával írja le a különféle műveletek végrehajtását – 7.0., 8.0., 9.0., 10.0.) olvasva, azonnal előbukkan Neumann matematikusi zsenialitása, szemben a kétségkívül erős mérnöki kvalitásaival.

- 12.4. A 12.3. fejezetben felsorolt memóriaigény-számításokat Neumann ebben a fejezetben összegezi, a különféle feladatok megvizsgálása után a memória kapacitását – némi túlbecsléssel – 8000 és 2000 közötti minor ciklusban határozza meg. *Miután 1 minor ciklus kettő az ötödiken egységnyi (unit), így a szükséges memóriakapacitás: 218, illetve 216 egységre jön ki. Neumann azt is megállapítja, hogy a számítógépekben az „üveg nyaka” (bottleneck) a memória. Ha a mai PC-re és az egyre bonyolultabb operációs rendszerekre, valamint alkalmazói programokra gondolunk, ez ma sincs másként.*

Megjegyzés. *Ebben az időben a problémát részben a memória ára okozta, ugyanis a kutatóintézetekben és a gyárakban szinte nem volt memóriaválaszték. Igaz, Atanasoff már évekkel korábban feltalálta a kapacitív dobot, de rajta kívül senki sem használta. A mágneses dob akkor még egyáltalán nem létezett, a Williams-csövet és a ferritmemóriát sem találták még fel. Egyedül a higanyos és a nikkell készletetű művonal jöhetett – memóriaelemként – számításba, talán az EDVAC-kal kapcsolatban ezért is gondolkodott Neumann soros számítógép tervezésében.*

Neumann – a számítások után – kimondja, hogy a számítógépben legalább negyedmillió egységet tartalmazó memóriára volna szükség.

- 12.5. Ennek a pontnak a címe: „Hogyan lehet egy negyedmillió egység kapacitású memóriát építeni?”

Megjegyzés. *Neumannból ismét előbukik a mérnök. Nem csak felméri az igényt, de elgondolkozik azon is, hogy egy ilyen memóriát meg is kell építeni.*

Neumann elgondolkozik és számol. Egy késleltető művonalon 30 bites bináris számokat lehetett tárolni, ami annyit jelentett, hogy negyedmillió számhoz 8000 művonalra lett volna szükség, amit – abban az időben – nem tartott megvalósíthatónak, ezért legfeljebb kb. 65 000 tárolt bináris számra gondolhatott, amit már „csak” 2000 művonalat jelentett volna, ugyanis – abban az időben – ez is hatalmas mérnöki munkát jelentett, ami nagyon nehezen lett volna kivitelezhető.

Megjegyzés. Neumann nem adott megoldást, nem is adhatott, mert nem volt villamosmérnök. A fennmaradt nagyszámú levélből tudható, hogy számos – memória-áramkörökre vonatkozó – kérdést tárgyalt meg Mihály öccsével és még többet Bay Zoltánnal, az utóbbtól leginkább a legújabb elektronikus felfedezésekkel kapcsolatban kért tanácsot: hogyan lehet ezeket az újabb eszközöket memória céljára felhasználni.

Ebben a fejezetben végül is meghatározza annak a memóriának a paramétereit (kapacitás, ciklusidő stb.), amivel a legtöbb feladatot el lehetett látni, miközben azzal nem nagyon törődött, hogy a szükséges memóriát hogyan fogják előállítani. Ez – végül is – a villamosmérnökök gondja maradt.

- 12.6.–12.7. Neumann folytatja a spekulációt, sőt néha bemerészkedik a villamosmérnök konstruktőrök területére is. Tanácsokat ad: hogyan lehet megfelelő mennyiségű elektroncsővel, csökkentett késleltetési idővel, rövidebb szinkronizáló impulzusokkal bizonyos vezérlési, valamint tárolási feladatokat ellátni, és így a kívánt kapacitású memóriát előállítani.

Ezt követően Neumann ismét tovább számol, szinte „hangosan” gondolkodik. Látható módon előtte alaposan tanulmányozta az áramkörök technikáját, az ember arra gondol, hogy Neumann mindezt elsősorban azért tette, mert meg akarta teremteni a számítógép-tervezés módszertanát és nem is azért, hogy egy új elvű, működő számítógépet hozzon létre.

Megjegyzés. Ismerve a később történeteket, az IAS gép tervezését és építését, Neumann valószínűleg itt gondolta végig, hogy a jövőben **nem** EDVAC-szerű, soros számítógépet kell tervezni és építeni, mert a soros gép és a szinkronizált soros memória zsákutca, ami sehová sem vezet. Valószínűleg ezért hagyta abba a „First Draft...” kidolgozása után azonnal, mindazoknak az elveknek – néhány kivétellel – az alkalmazását, amiket a „First Draft...” írása során kidolgozott és tért át az akkor még ismeretlen és sehol sem alkalmazott párhuzamos gépi architektúrára. Az újabb neumann spekuláció azt eredményezte, hogy az IAS gép végül kevesebb elektroncsővel épült meg (így – elektroncsőves gép révén – az üzemeltetés során kevesebb alkatrész tudott elromlani), mint az EDVAC, és körülbelül 30–50-szer gyorsabban hajtotta végre a műveleteket. Ennek az eredménynek a nyomán a fejlesztők szinte azonnal leálltak a soros számítógépek tervezésével, és a kutatóintézetek, valamint a cégek áttértek a párhuzamos számítógépek építésére. Még Eckert és Mauchly is.

- 12.8. Neumann leírja, hogy a memóriaszámítások mind a késleltető művonalas memóriára vonatkoztak, nem zárja ki azonban, hogy később ne találnának ki más, alkalmas memóriákat.

Neumann elsőként az *ikonoszkópra* gondolt, aminek az ernyőjén 2003500, azaz 200 000 pontot lehetett tárolni. Mindehhez egy nagyon bonyolult, speciális elektroncsőre és abban egy elektronsugárra volt szükség, amivel a bináris információt az ernyőre – világító pontok formájában – fel lehetett írni, és onnan ki lehetett olvasni. Ez az „ötlet” azért is érdekes volt, mert addig az ikonoszkópot képfelbontásra és képátvitelre használták, eredetileg nem számítógép-memóriának tervezték. Talán

Neumann látta meg benne először, hogy a képpontok bináris adatok tárolására is alkalmasak.

A fejezeten belül, az (a)-tól (d)-ig pontokban, Neumann a memória számos problémájával foglalkozik, amikre megpróbál részben becsléseket, részben pedig megoldásokat adni. Ezek közül néhányat érdemes felsorolni.

Neumann kiszámolta, hogy például 200 000 pont tárolásához az elektronsugarat milyen pontossággal kell vezérelni, hogy a tárolást biztonságosan meg lehessen oldani. Foglalkozott az információ beírásával és kiolvasásával, a memória sebességéből adódó kérdésekkel, a tárolt információ megcímzésével, de még néhány – az ikonoszkópban megoldandó – technikai problémával is, mint például a memória tárolási biztonsága.

Megjegyzés. *Az ikonoszkópot – szemben a késleltető művonalas memóriával – nem igen volt célszerű soros számítógép memóriájaként alkalmazni, ami például abból is látszik, ahogyan Neumann a kiolvasott információ késleltetésével foglalkozott. Erre azért volt szükség, hogy mind a műveletvégző (CA), mind pedig a vezérlő (CC) egység képes legyen a feladatát végrehajtani. Ezekbe az egységekbe ugyanis az információ sorosan, tehát a bitek egymás után léptek be.*

Az ikonoszkóp – technikailag – tipikusan párhuzamos memória volt, amint később Neumann az IAS számítógépben párhuzamos memóriaként használt fel. Ismét érdemes arra gondolni, nagyon valószínűsíthető, hogy Neumann az ikonoszkóp-memória tanulmányozása során már előre, egy párhuzamos számítógépen gondolkodott (az előregondolkodás egyébként is tulajdonsága volt), tehát az EDVAC-ot tervezte, miközben az IAS gépen is gondolkodott. Ez persze csak spekuláció.

Talán ezért írta Neumann a fejezet végére, hogy ezek után „ésszerűbbnek látszik a késleltető művonalas memória analízisének a folytatása”, de azért nem feledkezett meg az ikonoszkópról.

13.0. A memória (M) szervezése

13.1.–13.5. Ebben a fejezetben a 12.6. és a 12.7. fejezetekben leírtak alapján a memória belső szervezésével foglalkozik, meghatározza azokat a memória működéséhez szükséges logikai elemeket, amiket a rendszer kialakításához meg kell építeni. Műszaki részletekkel nem törődik, inkább az elemeknek, pontosabban a szükségesnek tartott részegységeknek a funkcióit írja le.

Ez a fejezet a következő 14.0. fejezet fordítottjának is felfogható, ugyanis a memória (M) oldaláról mutatja be, hogy a számítógép többi részével (pl. C vagy CC) az M hogyan fog kommunikálni, illetve hogyan kell az M – mai szóval architektúráját, akkor szervezést mondtak – kialakítani, hogy a számítógép más részei által előírt logikai követelményeknek meg tudjon felelni.

A fejezet pontosan leírja nemcsak az egész memória, hanem a részegységek szervezését, valamint lehetséges kapcsolati rendszerét is, amibe – az eredeti leírásban – csak azoknak érdemes mélyen belemenni, akik az EDVAC memóriájának a működését részletesen és mélyen meg akarják ismerni.

Megjegyzés. *A technikatörténettel foglalkozó kutatók a „First Draft...” tanulmányozása közben értik meg igazán, hogy az EDVAC terveit, amit Neumann szinte bit mélységig kidolgozott, miért lehetett a némileg korszerűsített, de az EDVAC*

szervezésével szinte teljesen azonos EDSAC és BINAC számítógépeknél felhasználni. Valószínű – a leírásban sűrűn előforduló, {} jellel kiemelt részek is ezt bizonyítják – Neumann az EDVAC leírásában csak a szerinte leglényegesebb kérdések megoldására koncentrált, így számos részprobléma, amire a fenti jel utal, megoldatlan marad.

Talán nem tiszteletlenség Neumann dolgozatáról feltételezni, hogy a „First Draft...”-ban hibák és ellentmondások is előfordulhattak, aminek a valószínűsége – ismerve Neumann precizitását – nagyon kicsi, de nem kizárható. Ezeket a problémákat az EDSAC tervezésénél Wilkes, míg a BINAC tervezésekor Eckert és Mauchly is föltételezhetően felfedezték és kijavították, amire számos példa volt, például később, az IAS „klónok” tervezése során is.

Nagyon valószínű, amit nemcsak az EDSAC, hanem a BINAC példája is bizonyított, hogy a „First Draft...” nyomán minden számítógép-tervezőnek sokkal könnyebb volt a dolga, Neumann ugyanis elvégezte a munka nagyobbik – logikai – részét, mint azoknak, akik korábban – Neumann munkája előtt – igyekeztek számítógépeket alkotni.

14.0. A CC és az M

Megjegyzés. Ebben a fejezetben a Central Control (CC, központi vezérlőegység) és a memória (M) kapcsolatát elemzi Neumann János, pontosabban azt, hogy a memóriában lévő utasítások hogyan „vezérlik” a központi vezérlőegységet.

14.1. Ebben a pontban a CC elemzésében mélyedünk el. Ehhez az utasításrendszert kell áttekintenünk, mivel a CC feladata ezen utasítások fogadása, értelmezése és más egységekkel való végrehajtása.

Feladatunk tehát a CC-t vezérlő utasításlista létrehozása, azaz a berendezésben használt kód leírása, és ezek matematikai és logikai jelentésének definiálása.

Vizsgáljuk meg a CC-t és főleg kapcsolatát az M-mel.

Az utasítások az M-ből kerülnek a CC-be, ugyanonnan, ahol a numerikus adatokat is tároljuk. Az M tartalma ún. minor-ciklusokból áll, minden minor-ciklusnak egy olyan jelet is kell tartalmaznia, amely azt jelöli, hogy számról vagy utasításról van-e szó.

Az utasítások 4 csoportba oszthatók:

a) csoport: 10 speciális műveletet hajtanak végre (+, −, ×, ÷, három átviteli művelet, két konverziós művelet (decimálisból binárisba, binárisból decimálisba).

b) csoport: a számokat a memória egyik helyéből a másikba viszik át.

c) csoport: ide tartoznak azok az utasítások, amelyek a Cc-nek az M-mel való kapcsolatát változtatják meg: egyik memóriarekeszről egy másikra adják át a vezérlést, hogy onnan vegye a CC a következő végrehajtandó utasítást.

d) csoport: az input és az output berendezések vezérlését végzik.

14.2.–14.5. Ezekben a pontokban pontosan megadja Neumann János, hogy az egyes utasítások hogyan kerüljenek át a memóriából a CC-be és hogyan végezzék a vezérlést. A következőket külön kiemeljük:

A memóriában egymás melletti rekeszekben található utasítások végrehajtását egyszerűen lehet vezérelni. Lenniük kell azonban olyan utasításoknak is, amelyek kivételes esetként arra utasítják a CC-t, hogy egy tetszőleges M-beli pontra (rekeszre) tegyék át a CC kapcsolatát.

Megjegyzés. Ezekben a pontokban azt írja le a szerző, hogy a minor-ciklusokkal hogyan valósulhat meg a vezérlésátadás.

15.0. A kód

- 15.1. A memóriának az előbbi fejezetben történő specifikációja lehetővé teszi, hogy definiáljuk a kódokat, amelyek a CC logikai vezérlését (és ezen keresztül az egész berendezését) valósítják meg.

Az M memóriaegységekből áll, amelyeknek mindegyike egy hatás jelenlétével vagy hiányával jellemezhető. Ezek az egységek az 1, illetve 0 bináris számjegyeket reprezentálják. Ezeket az egységeket 32 egységből álló minor-ciklusokba csoportosíthatjuk; ezek a csoportok szerepelnek majd az alábbi bevezetendő kódokban.

Megjegyzés. Egy minor-ciklus voltaképpen egy rekeszt jelent, azaz a memóriában egy olyan területet, amelynek bináris jegyeiből álló tartalma egy egységet (számot, utasítást) reprezentál. Neumann János a Jelentésben még nem használja a rekesz elnevezést. Mint korábban említettük, Neumann János művonalas memóriára alapozta tervét; ebben például 32 jegyből álló bitsorozat valóban „ciklusban” keringett a művonalban, így tárolta a memória, tehát egy minor-ciklus egy rekesznek felelt meg.

Az egy-egy csoportban szereplő bináris számjegyek, egy $i_0, i_1, i_2, \dots, i_{31}$ sorozatot alkotnak, amelyet $I = (i_0, i_1, i_2, \dots, i_{31}) = (i_v)$ alakban is írhatunk.

A minor-ciklusokat két osztályba sorolhatjuk: a számok, illetve utasítások osztályába. Ezt a két kategóriát az első bittel, tehát i_0 -val különböztetjük meg: jelölje $i_0 = 0$ azt, hogy számról, $i_0 = 1$ pedig azt, hogy utasításról van szó.

- 15.2. Számoknál a 31. egység a bináris számjegyeket, illetve az előjelet reprezentálja.

Az aritmetikai műveletek természetéből kifolyólag (tekintettel az átvitelre) a bináris számjegyeket jobbról balra töltjük fel, azaz $\xi = i_{31}, i_{30}, i_{29}, \dots, i_1$ alakban állnak elő a számok. Az utolsó betöltött jegy i_{31} az előjel, mégpedig $i_{31} = 0$ a +, $i_{31} = 1$ a – előjelet jelenti. A bináris pontot az előjel után helyezük el, vagyis az ábrázolandó ξ számot mod(2)-vel a $(-1, 1)$ intervallumba transzformáljuk, azaz:

$$\xi = i_{31}, i_{30}, i_{29}, \dots, i_1 = \sum_{v=1}^{31} i_v 2^{v-31} \pmod{2}, \quad -1 \leq \xi < 1.$$

(I) Típus	(II) Jelentés						(III) Rövid jel	(IV) Kód- szimbólum
								Minor-ciklus $I = (i_v) =$ $= (i_0 i_1 i_2 \dots i_{31})$
Szám vagy utasítás (γ)	$A \bar{\xi} = i_{31} i_{30} \dots i_1 = \sum_{v=1}^{31} i_v 2^{v-31} \pmod{2};$ $-1 \leq \xi < 1$ számot tárolja. Az i_{31} bit az előjel, 0 a + jelet, 1 a – mínuszjelet jelöli. Ha a CC ehhez a minor-ciklushoz fordul, akkor az utasításként működik, amely ξ -t I_{ca} -ba viszi át. Nem ez történik azonban, ha ez a minor-ciklus közvetlenül egy $w \rightarrow A$, vagy $wh \rightarrow A$ után következik.						$N\xi$	$I_0 = 0$
Utasítás (α)+(δ)	Ezek olyan utasítások, amelyek CA-ban hajtódnak végre. w a 11.4.-ben szereplő művelet valamelyike. Az alábbi táblában a w decimális oszlopban a műveletek decimális, a w.bináris oszlopban bináris alakban, a w oszlopban szimbólumukkal szerepelnek						$w \rightarrow \mu\rho$ vagy $wh \rightarrow \mu\rho$	$I_0 = 1$
Utasítás (α)+(ε)	w. decimális	w. bináris	w w	w. decimális	w. bináris	w	$w \rightarrow f$ vagy $wh \rightarrow f$	
Utasítás (α)+(θ)	0	0000	+	5	0101	i		
	1	0001	–	6	0110	j		
	2	0010	\times	7	0111	s		
	3	0011	\div					
	4	0100	$\sqrt{\quad}$					
				8	1000	db	$w \rightarrow A$ vagy $wh \rightarrow A$	
				9	1001	bd		
Utasítás (α)	A III. oszlopban szereplő jelek jelentései: – h azt jelenti, hogy az eredmény megőrződik az O_{ca} -ban – $\rightarrow \mu\rho$: az eredmény a μ major-ciklus ρ minor-ciklusába kerül átvitelre – $\rightarrow f$: közvetlenül az ε utasítás végrehajtása után az eredmény a végrehajtás alatt álló utasítást követő minor-ciklusba kerül – $\rightarrow A$: az eredmény I_{ca} -ba kerül. – ahol nincsen \rightarrow jel, ott nem történik tárolás (az eredmény megőrzésétől eltekintve)						wh	
Utasítás (β)	A μ major-ciklus ρ minor-ciklusában lévő szám I_{ca} -ba kerül.						$A \leftarrow \mu\rho$	
Utasítás (ζ)	Az utasítás CC-t μ major-ciklus ρ minor-ciklusával köti össze.						$C \leftarrow \mu\rho$	

15.3. Az utasítások 31 bitje, az utasítás hatását adja meg.

Az alábbi utasításcsoportokat definiáljuk.

α) Ide tartoznak a CC azon utasításai, amelyek hatására a CA elvégzi a 14.1.

a) csoportban szereplő 10 utasítást. Jelöljük ezeket sorban a 0,1, 2, ..., 9 számokkal, illetve az ezeknek megfelelő 4 jegyű bináris számokkal.

Megjegyzés: Ebben a pontban Neumann János részletesen elemzi, hogy a CA különböző egységeiben hogyan kell kezelni az operandusokat a műveletek elvégzése után, mikor kell törölni az adott egységet, mikor nem, illetve hogyan lehet megoldani, hogy az előző művelet eredményét a következő műveletben fel lehessen használni (például az xy szorzat eredményét hogyan lehet hozzáadni az előtte elvégzett művelet z eredményéhez, azaz képezni a $z + xy$ összeget).

Annak jelölésére, hogy egy művelet után az O_{ca} -t (az aritmetikai egység eredményregisztere) törölni kell-e vagy nem, a c -vel jelölt bitet kell használni ($c = 0$ törlést, $c = 1$ a törlés elmaradását jelenti).

β) Ebbe a csoportba azok az utasítások tartoznak, amelyek hatásaként a számok M-ből CA-ba mennek át.

γ) Olyan utasításokra is szükség van, amelyek ún. direkt betöltést végeznek, vagyis az utasításszámláló által megcímezett rekeszt követő rekesz tartalmát töltik az I_{ca} -ba.

δ) Szükségesek olyan vezérlőutasítások is, amelyek hatására a számok CA-ból M-be kerülnek.

ϵ) Az ebbe a csoportba tartozó vezérlőutasítások az utasításszámláló által megcímezett rekeszt követő címre adják át a vezérlést.

θ) Az utasítás hatására a számok CC-ből CA-ba kerülnek át (pontosabban, az O_{ca} -ból az I_{ca} -ba).

ζ) Ezek az utasítások a CC és az M kapcsolatát az M egy másik helyén található minor-ciklusára (rekeszre) irányítják át (vezérlésátadás).

η) Ebbe a csoportba az input, output utasítások tartoznak.

15.4. Ebben a pontban a szerző megvizsgálja, hogy a 31 bitből álló utasításoknak milyen legyen a szerkezete.

Megállapítja, hogy kerülni kell több funkció beépítését az utasításba.

15.5. Vannak azonban olyan teendők, amelyeket célszerű összevonni egy utasításba; pl. az (α) típusú utasításoknál a művelettel együtt meghatározható, hogy a művelet eredménye hova kerüljön. Neumann pontos elemzést végez arra nézve, hogy az egyes utasítások által végzendő teendőket hány számjeggyel (bittel) lehet elérni. Azt hozza ki, hogy 32 bináris jeggyel mintegy 50%-os átlagos „hatékonyság” érhető el.

15.6.1. Az előbbi pontokban végzett elemzések alapján Neumann az 1. táblázatot állítja elő.

Utószó a First Drafthoz

Ezzel végére is értünk a First Draft ismertetésének. Reméljük, sikerült a fontosabb gondolatokat kiemelni, érzékeltetni a munka „első vázlat” jellegét és különösen Neumann gondolkodásmódját, elemzési módszerét, a problémák „tálalásának” módját.

3. Egy utasításrendszer, egy „magas szintű” nyelv

Az EDVAC gép utasításrendszere

Az előbbieken (a First Draftban) láttuk, hogy Neumann János miképpen képzelte el a gép utasításrendszerét. Az EDVAC gép tervezett utasításrendszere lényegében követte a Neumann által leírt elveket (a sémát).

Az EDVAC gépben a tervezett utasítások szerkezete az alábbi volt:

műveleti kód 5 bit	cím1 6 bit	cím2 6 bit	cím3 6 bit	vezérlőbitek
-----------------------	---------------	---------------	---------------	--------------

A gép tehát 3 című volt.

A gép tervében az alábbi 15 utasítás szerepelt (**a**, **b**, **g** memóriacímek, **a**, **b**, **g** pedig konstans operandusok).

- 1) **add a b g**: Az **a** és **b** tartalmát összeadja, és az eredményt **g**-be helyezi.
- 2) **sub a b g**: A **b** tartalmából kivonja **a** tartalmát, és az eredményt **g**-be helyezi.
- 3) **mul a b g**: Az **a** és **b** tartalmát összeszorozza, és az eredményt **g**-be helyezi.
- 4) **neg a b g**: Ugyanaz, mint a *mul* utasítás, de **g**-be az eredmény -1 -szeresét teszi.
- 5) **c a b g**: Ha az **a**-ban tárolt szám nagyobb, a **b**-ben tároltnál, akkor a következő utasítást a **g** címről veszi.
- 6) **x a b g**: Ha $a > b$, akkor a következő utasítást a **g** címről veszi.
- 7) **t -- g**: A **g** memóriacímen tárolt utasításra adja át a vezérlést.
- 8) **p a b g**: Az **a**-ban tárolt számot **g** pozícióval balra tolja, és az eredményt **b**-be teszi.
- 9) **q a b g**: Az **a**-ban tárolt számot **g** pozícióval jobbra tolja, és az eredményt **b**-be teszi.
- 10) **i a b g**: A soron következő (eggyel nagyobb című) memóriarekeszben lévő **a b g** utasításban az **a**, **b**, **g** címeket az **a**, **b**, **g** értékekkel megnöveli.
- 11) **e a b g**: Kiemel bizonyos biteket az **a** című rekesz tartalmából, és ezeket elhelyezi a megfelelő pozíciókban a **b** című rekeszben. A szóban forgó biteket a **g** tartalma határozza meg.
- 12) **fn – b g**: **g** számú szót beolvas **b**-be, és az **n** szalagot tovább mozgatja.
- 13) **bn – b g**: Ugyanaz, mint előbb, de a szalagot visszafelé mozgatja.
- 14) **fn a – g**: Kiír **g** szót az **a** rekeszből a szalagra, és a szalagot előre mozgatja.
- 15) **bn a – g**: Ugyanaz, mint előbb, de a szalagot visszafelé mozgatja.

Ez az utasításrendszer a First Draftban szereplő elvek alapján készült ugyan, de egyes történetírók szerint az EDVAC gép nem pontosan eszerint működött.

Neumann és az első „magas szintű” nyelv

Már az EDVAC tervezése közben felmerült, hogy szükség volna olyan nyelvre, amivel a matematikai eljárásokat a gép számára, a gépi kódnál jobb (kényelmesebb) módon lehet leírni.

Az első ilyen kísérletet nem Amerikában tették meg, hanem Európában. Németországban Konrad Zuse 1945-ben kidolgozott egy Plankalkül nevű nyelvet, ami lényegében a Hilbert-féle ítéletkalkuluson alapult. A kéziratot csak 1972-ben publikálták, bár rövid ismertetőt 1948-ban és 1959-ben Zuse már közölt.

Az óceán túlsó partján is foglalkoztak a gépi kódnál magasabb szintű programozási nyelvek kidolgozásával. Goldstine kezdte el elemezni, hogyan lehetne olyan módon leírni az eljárásokat – akkor még csak a matematikai (numerikus) eljárásokra gondoltak –, hogy azt a „kódoló”, aki a gép nyelvére, gépi kódra írja át az eljárást, könnyen meg tudja ezt tenni, és az ember számára is jól áttekinthető legyen. A feladat megoldásához csatlakozott Adele Goldstine (Goldstine első felesége), Neumann János és Arthur Burks is.

A kidolgozott „nyelv” filozófiája egészen más volt, mint a Zuse által javasolté. Az előbbi nagy súlyt helyezett a típusdeklarációkra, míg Neumannék ezzel egyáltalán nem törődtek.

A Neumannék által kidolgozott „nyelv” voltaképpen a blokkdiagramok őse volt (ők „flow-diagramnak”, később „flowchartnak” nevezték).

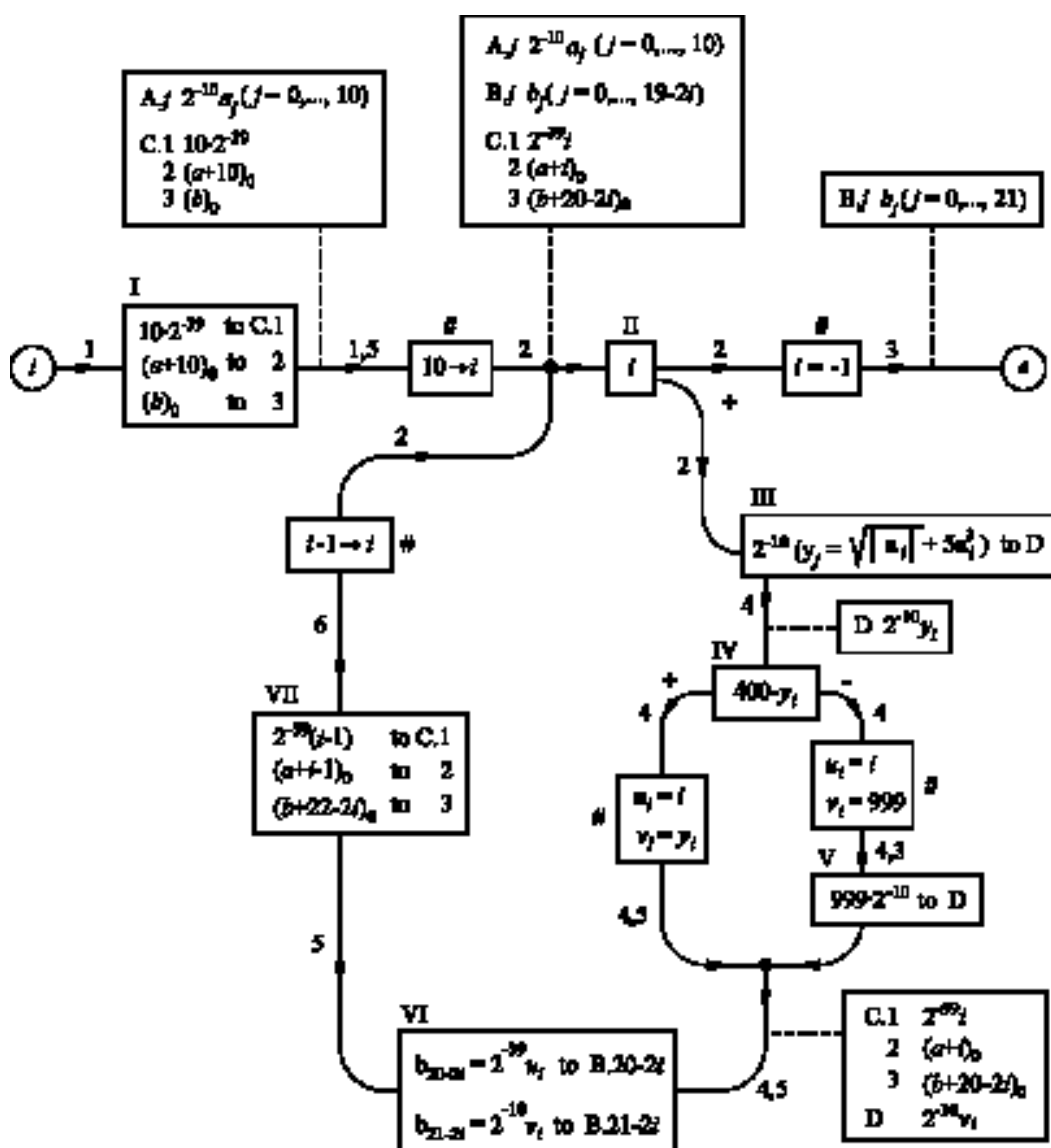
A projekt 1946-ban indult, és 1947-re lettek kész vele. Folyóiratokban ugyan nem publikálták (kéziratban „Planning and Coding Problems for an Electronic Computing Instrument” címet viselte), de gyorsan elterjedt, és a később létrejött programnyelvek mellett is hosszú évtizedeken keresztül a programterveket blokkdiagramokkal ábrázolták.

A Goldstine–Neumann-diagram téglalapokból (bokszo**kból**, dobozokból) és az ezeket összekötő nyilakból állt. (Lásd a 10. ábrát.)

A dobozok négyfélék lehetnek.

a) Római számokkal jelölt *operációs dobozok*, amelyekben a memóriában végzendő átviteleket jelölték, azaz a memóriarekeszek (akkor még nem így nevezték) változásait adták meg. Voltaképpen az elágazás nélküli (értékdadásokat jelentő) programrészek jelölésére szolgáltak az ilyen típusú dobozok.

b) *Alternatív dobozok* (szintén római számokkal jelölték), amelyeknek + illetve – jellel jelölt kimenetük van. Ezek a dobozok vezérlésátadást jelentettek: a dobozban szereplő mennyiség előjelétől függően jelölték ki az eljárás további irányát, tehát logikai feltételek szerinti elágazást adtak meg vele.



10. ábra. Az első magas szintű (grafikus) nyelv, amit Neumann János, Adele és Hermann Goldstine dolgoztak ki: a programok leírására és gyors áttekintésére használt folyamatábra. (1947)

c) *Helyettesítő dobozok*, amelyeket a # jellel jelöltek és ezekben a „ \rightarrow ” jelet használták. Az ilyen dobozokban nem a gép által elvégzendő műveleteket jelölték. Például az „ $i - 1 \rightarrow i$ ” tartalmú doboz azt jelenti, hogy azt i (index) $i - 1$ -re változik. Ma azt mondanánk, hogy i ciklusváltozó értéke 1-gyel csökken. (Neumannék még nem vezették be a mai értelemben vett ciklus fogalmát.)

d) Állítást tartalmazó dobozok (állításdobozok), amelyeket szintén # szimbólummal jelöltek, amelyekben a külső jelölések és a vezérlés kurrens állapotát adták meg. Az ábrán három állításdobozt látunk. Az egyikben az „ $i = -1$ ” állítás szerepel, a másik kettőben az, hogy az u_i, v_i („változók”) kimenetek, ezen a ponton értékeket vesznek fel. (Mai szemmel ezt jogosabban hívhatnák „beállítási” dobozoknak, mert az i, u_i, v_i változók értékeit állítják be. Azt is mondhatnánk, hogy ezek kommentárok.)

A helyettesítő és az állításdobozok nem jelentenek a számítógép által elvégzendő műveleteket.

Az ábrán szereplő dobozokban sok helyen látunk 2^{-n} -nel való szorzásokat. Ez azzal függ össze, hogy – mint a First Draftban is láttuk – az akkori gépek [az EDVAC is] fixpontosak voltak, mégpedig a bináris pont [vessző] úgy volt elhelyezve a memóriában binárisan ábrázolt x számban, hogy $-1 < x < 1$ teljesüljön, vagyis a gépben csak egynél abszolút értékben kisebb számok voltak ábrázolhatók, azaz a szám legmagasabb helyiértéke 2^{-1} , a következő 2^{-2} i-t, az utolsó 2^{-n} volt. (n a szám hosszát, azaz a bitek számát jelenti, EDVAC-nál ez 40 volt.) Például a memóriában ábrázolt 101101 szám ($n = 6$) az

$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6}$$

mennyiséget, vagyis a

$$0.101101$$

számot jelenti.

A fixpontos gép komoly gondot jelentett a programozónak. A megoldandó feladatot úgy kellett transzformálni, hogy minden szám – az input adatok, a közbülső részeredmények, a végeredmény – abszolút értékben egynél kisebb legyen. Például az $\frac{a}{b}$ hányados kiszámításánál előbb meg kellett vizsgálni (a programban), hogy $a < b$ teljesül-e; ha nem, akkor a -t el kellett osztani (annak idején úgy mondtuk, hogy normálni kell) egy olyan c számmal, hogy $\frac{a}{c} < b$ teljesüljön. Természetesen a c -vel való osztás következményeit végig kellett vinni a teljes „képletben”.

Az ábrán látható táblázatban szerepel még egy érdekes jelölés, például $(a+i)_0$, $(b)_0$ stb. Általánosságban x_0 azt jelentette, hogy x egy memóriacím (egy egész szám), mégpedig úgy, hogy a gépben x_0 a $2^{-15}x + 2^{-39}x$ alakban van jelen, tehát egy olyan bináris szót jelöl, amelyben x kétszer szerepel: a 9–20 és a 29–40 pozíciókon (balról számítva). Az ilyen x_0 típusú számokat utasítások módosítására használták; egy 40 bites szó mindkét felét lehetett módosítani vele (pl. hozzá lehetett adni ezt a számot egy utasításhoz, miáltal az utasításban szereplő címek módosultak).

Megjegyzés. Az EDVAC-kal kapcsolatos fenti paraméterek (a szám hossza, a cím két részre bontása) nem a First Draftban leírt EDVAC, hanem feltehetőleg inkább az EDVAC után épült IAS gép adatai.

Amikor egy folyamatdiagram elkészült, a programozó elkezdhetette a „dobozok” gépi kódra történő leképezését (ezt „statikus kódolásnak” nevezték).

A folyamatábra csak egy eszköz volt, ami megkönnyítette a program írását, de nagy szükség volt a programozó problémamegoldó képességére, a gépi kódú utasítások ismeretére. Nem gondoljuk, hogy az Olvasónak érdemes volna dobozról dobozra menve elemezni az ábrán látható folyamatdiagramot, ha mégis, úgy segítünk azzal, ha megmondjuk, hogy az ennek alapján elkészítendő program a 10 elemű $(a_1, a_2, \dots, a_{10})$ tömb elemeire kiszámítja az

|

kifejezés értékét, és amennyiben $400 - y_i > 0$, akkor rögzíti (például kinyomatás céljából) az (i, y_i) , ellenkező esetben pedig az $(i, 999)$ értékpárt.

A Goldstine–Neumann folyamatdiagrammal kapcsolatban megemlítjük, hogy ebben a szubrutin fogalma nem szerepel (tehát a szubrutin jelölésére nem tesznek javaslatot).

4. Epilógus

A „*First Draft...*” – Neumann János számítástechnikai alkotásai közül az első – amivel nem csak Amerikában, de mindenütt a világon elismerést szerzett mind a Pennsylvania Egyetemen, mind az ott dolgozó kutatóknak és természetesen saját magának is. **Feltételezhető, hogy ennek a tanulmánynak és persze későbbi számítástechnikai alkotásainak köszönhetően lett Neumann János világhírű matematikusból a „számítógépek atyja”.**

A számítástechnika történetében viszonylag kevés olyan alkotás található – ilyen a „*First Draft...*” is –, ami a megjelenése után több mint 50 évvel pontosan úgy érvényes, mint ahogyan a megjelenésekor volt. A „Vázlat” megítélésében az sem lehet közömbös, hogy a bírálók legfeljebb azért szóltak hozzá a tanulmányhoz, mert kimaradtak a névsorból, azért senki sem, hogy ahhoz valamit hozzátegyen, netán abból elvegyen. Annak ellenére, hogy a címben benne van, hogy „draft”, ez a vázlat a maga valóságában tökéletes, egy költő azt mondhatná: „Olyan, mint egy csiszolt, tiszta gyémánt. Nincs benne hiba.”

Korábban már említettük, hogy az EDVAC szerzősége körüli viták, a szabadalmaztatással kapcsolatos torzsalkodások oda vezettek, hogy Presper Eckert és John Mauchly szakítottak nemcsak Neumann Jánossal és Hermann Goldstine-nel, hanem a Pennsylvania Egyetemen is.

Az ENIAC-ot a kormány – hivatalosan – 1946. június 30-án vette át a Pennsylvania Egyetemtől (PENN). A háborús évek alatt az Egyetem az ENIAC fejlesztésében kifáradt, ezért a gépet szíves örömet visszaadták a hadseregnek (egy púppal kevesebb – gondolták az egyetemen). Az ENIAC-ot 1946. november 9-én kikapcsolták és leszerelték, majd Aberdeenbe, a Ballisztikai Kutató Laboratóriumba szállították. A gép újraélesztése 9 hónapig tartott, 1947. július 23-án állították ismét üzembe.

Az Egyetem nem volt korrekt sem Mauchlyval, sem Eckerttel, sem Goldstine-nel, sem Neumann-nal szemben, ugyanis az ENIAC után számítógép-tervezéssel és – egy ideig – magával a számítógép-tudománnyal sem foglalkoztak. Az ENIAC alkotói azt is nagyon sérelmezték, hogy – a fenti négy kiváló szakembernek – katedrát sem ajánlottak fel az egyetemen. Az addig – a számítógép-tudományban – vezető pozícióban lévő egyetem számítástechnikai fejlesztései leépültek, csak évekkel később kezdtek el újra a számítástechnikai konferenciákat szervezni, amivel megpróbálták a régi híruket visszaszerezni. A számítógép-tudományban a vezető szerepet a princetoni IAS, a UNIVAC cég és számos más egyetem, nem utolsósorban az IBM vette át, amelyek folytatták és továbbfejlesztették Neumann János és társainak munkáját.

1996-ban az Egyetem óriási ünnepséget tartott az ENIAC születésének az 50. évfordulóján. Az ENIAC építői közül, akik még éltek, ott voltak. A díszvendég Hermann Goldstine volt. Többen nehezményezték, hogy a megemlékezésen szinte egyedül Goldstine beszélt Neumann-nak, az ENIAC tökéletesítésében játszott szerepéről, Atanasoff érdemeiről, az egyetemi szónokok, de a kormány képviselői erről nem szóltak, kizárólag Eckert és Mauchly kétségtelen érdemeit emlegették.



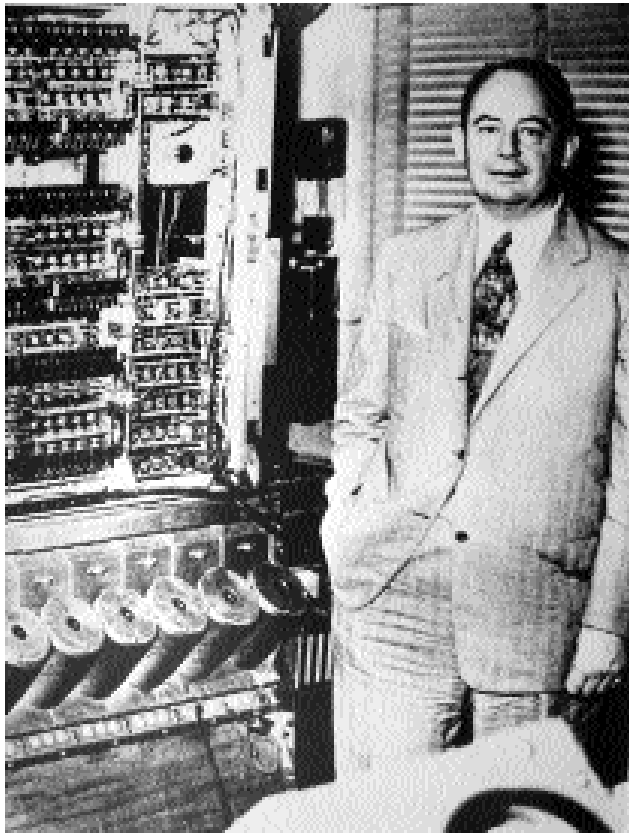
11. ábra. A princetoni IAS (Institute for Advanced Study) központi épülete.



12. ábra. Az ENIAC születésének az 50. évfordulója Philadelphiában. A képen Hermann Goldstine és Neumann János titkárnője. (1996)

Eckert és Mauchly még azt sem várták meg, hogy az ENIAC számológépből kvázi tárolt programú számítógéppé alakuljon át, úgy hagyták el az egyetemet, és megalakították Philadelphiában a saját közös vállalatukat, az Electronic Control Co.-t. Ezután hozzáfogtak az EDVAC terveit követő, ugyancsak higanyos késleltetésű művonalakkal dolgozó, BINAC számítógép tervezéséhez és gyártásához. Egyébként Amerikában ez volt az első számítógép, ami az ENIAC után épült. Ezt követően megalapították az „Eckert–Mauchly Computer Corporation”-t, megtervezték és megépítették a UNIVAC-nak nevezett gépet, amit 1951-re fejeztek be. Ezzel hamarosan elkezdődött az amerikai számítógépipar szédületes fejlődése.

Neumann sem akarta a számítógépek fejlesztését befejezni, mert már akkor ott motoszkálhatott benne az IAS gép terve.



13. ábra. Az elkészült IAS gép. Előtte az alkotó, Neumann János. (1952)

Az IAS gépben Neumann és Goldstine – minden korábbi géphez képest – forradalmi változásokat vezettek be. Miután ez az írás az EDVAC-ról és nem az IAS gépről szól, ezeket a változásokat – Goldstine könyvéből vett idézetekkel – csak nagyon röviden foglaljuk össze:

– Az IAS gépet Burks–Goldstine–von Neumann: „Preliminary Discussion of the Logical Design of an Electronic Computing Instrument” című alapvető munkája írta le.

– A tanulmányban a szerzők, a számítástechnika tudományának az egyik legfontosabb döntését hozták meg: az IAS-ben *nem* az EDVAC-nál alkalmazott soros tárolási módot alkalmazzák, hanem a *párhuzamos* szervezési módot, „*aminek az eredménye egy ún. párhuzamos működésű gép (parallel machine) lett, szemben az EDVAC-kal, amelynek a szervezése soros volt... (...) E két rendszer közötti fő különbség az összeadás elvégzésének a*

módjában rejlik; a párhuzamos működésű gépben a megfelelő számjegyek párpait egyidejűleg adjuk össze, míg egy soros szervezésű gépben e számpárok összeadása egymás után történik...

– Eleinte az IAS gép tervezőiben még élt az alkatrészeket „feleslegesen pazarló” műszaki megoldástól való félelem, amit Neumann oszlatott el, aki kiszámolta, hogy a párhuzamos működésű gép – a látszat ellenére, miután a vezérlőegysége egyszerűbb, mint a soros gépeké – kevesebb alkatrészből felépíthető, mint a soros számítógépek. A körülbelül egyforma kapacitású párhuzamos IAS gép összesen 2000, míg a soros EDVAC 3000 csövet tartalmazott. A Neumann-gép – a párhuzamos működés miatt – 600 mikroszekundum alatt végzett el egy szorzást, míg az EDVAC 3,0 milliszekundum alatt. Az IAS gép az ikonoszkóp memóriájába 25 mikroszekundum alatt helyezett el egy szót, az EDVAC pedig – a soros memóriába – 200 mikroszekundum alatt.

– A két gép között teljesítménykülönbség volt, ezért – az IAS megszületése után – soros számítógépeket gyakorlatilag alig építettek. Ettől kezdve az IAS gép volt az etalon, világszerte és főleg Amerika-szerte ezt a gépet másolták. Az IAS gép közeli rokonai (mai kifejezéssel klónjai): a JOHNNIAC, az ORDVAC, az ILLIAC, de még a UNIVAC 1100-as és az IBM 700-as és 7000-es szériái, ezenkívül még más – párhuzamos, külföldi – gépek is voltak, például a már említett első hazai elektronikus számítógép az M-3-as is.

– A gép struktúráját illetően úgy határoztak, hogy a gépben ábrázolt szám 39 bitből és egy előjelbitből fog állni.

– Az IAS gép az utasításrendszerekben is forradalmi változást hozott, ez a „forradalom” még ma is tart. Az EDVAC háromcímű gép volt, az oka egy korábban elterjedt hiedelem volt, a tervezők azt mondták, minél több című egy gép, annál gyorsabban számol. Azután – amint ezt a tanulmány magyarázatában már említettük – Neumann rájött arra, hogy egy programban a két- vagy háromcímű utasítások második vagy harmadik címét nem használják, egy utasításba legtöbbször csak egy címet írnak, így az utasítások címeinek egy része az esetek többségében kihasználatlan marad. Ezért az IAS gépet eleve – először a számítástechnika történetében – egycíműre tervezték, egy 40 bites szóban így két utasítást (10 bit a művelet, 10 bit pedig a cím kódolására – lásd: *H. Goldstine: A számítógép Pascaltól Neumannig*) tudtak elhelyezni. Azóta is csak egycímű gépeket építenek.

A háború befejezése után – 1946 őszén – brit látogatók nagy létszámú csoportja érkezett a Moore Intézetbe, közöttük Maurice Wilkes is, aki kézbe kapta a „First Draft...”-ot, és megismerhette az ENIAC-kal és az EDVAC-kal kapcsolatos tervezési elgondolásokat – közöttük a tárolt program elvét is. Ezután visszatért az angliai Cambridge-be, ahol megépítette a világ első tárolt programú számítógépét, az EDSAC-ot (Electronic Delay Storage Automatic Computer), ami előbb készült el, mint az IAS, Neumann János számítógépe.

Jöttek látogatók Svédországból, aminek nyomán megépült a BESK, a dánok Svédországból és Angliából vették át a tapasztalatokat, így épült meg a DASK (Danish BESK). A szovjetek a Moore Intézetből postán kérték el a számítógépek építésének megkezdéséhez szükséges leírásokat, Goldstine – a már említett munkájában is leírta – hogy meg is kapták. Valószínűleg ennek nyomán épültek meg az első szovjet számítógépek, a MESzM és a BESzM, majd később egész sor különféle architektúrájú és teljesítményű számítógép az ország minden részén.

Neumann János alapvető munkája, akaratának megfelelően, elterjedt az egész világon.