

**DR. SZELEZSÁN JÁNOS**

A számítástechnika  
története

# TARTALOMJEGYZÉK

<b>1. FEJEZET: AZ ABAKUSZTÓL AZ ENIAC-IG</b> .....	<b>4</b>
<b>1.1. AZ ÓKOR ÉS KÖZÉPKOR</b> .....	<b>4</b>
1.1.1. Az ókortól kezdve próbálták gépesíteni a számolást.....	4
1.1.2. A „Napier pálcák” számológép.....	4
1.1.3. Schickard mérnök (1592-1635).....	5
1.1.4. Pascal (1623-1662).....	6
1.1.5. Leibniz (1642-1716).....	7
1.1.6. Müller.....	7
<b>1.2. A SZÁMÍTÁSTECHNIKA ŐSATYJA: CHARLES BABBAGE (1791-1871)</b> .....	<b>7</b>
1.2.1. Élettörténet.....	7
1.2.2. Babbage munkásságának egyéb területei:.....	8
1.2.3. Véges differenciák összeadó gépe (Difference Engine).....	8
1.2.4. Az Analitikai Gép (Analytical Engine).....	10
<b>1.3. ADA AUGUSTA LOVELACE (1815-1852)</b> .....	<b>11</b>
<b>1.4. BABBAGE-TÓL AZ ENIAC-IG</b> .....	<b>13</b>
1.4.1. Differencia-összeadók tervezése.....	13
1.4.2. Kereskedelmi, ügyviteli gépek, adatfeldolgozás: a Hollerith-gépek.....	13
1.4.3. Az Analitikai gép öröksége.....	14
1.4.3.1. <i>H. Aiken (1900-1973, USA)</i> .....	15
1.4.3.2. <i>Németország</i> .....	15
<b>2. FEJEZET: ÚJ KORSZAK, MEGJELENIK AZ ELEKTRONIKA</b> .....	<b>17</b>
<b>2.1. AZ ELSŐ ELEKTRONIKUS SZÁMÍTÓGÉP, AZ ENIAC</b> .....	<b>18</b>
<b>2.2. NEUMANN JÁNOS A SZÁMÍTÓGÉPEK ATYJA</b> .....	<b>19</b>
2.2.1. Rövid életrajza.....	19
2.2.2. Szakmai területek.....	20
2.2.3. Út a számítógéphez.....	21
<b>2.3. AZ EDVAC (ELECTRONIC DISCRETE VARIABLE COMPUTER) GÉP</b> .....	<b>21</b>
<b>2.4. A FIRST DRAFT</b> .....	<b>23</b>
<b>2.5. AZ IAS SZÁMÍTÓGÉP</b> .....	<b>24</b>
<b>2.6. AZ EDSAC (ELECTRONIC DELAY STORAGE AUTOMATIC COMPUTER) GÉP</b> .....	<b>25</b>
<b>2.7. TOVÁBBI „ÖSSZÁMÍTÓGÉPEK”</b> .....	<b>27</b>
2.7.1. Anglia.....	27
2.7.2. Szovjetunió.....	28
2.7.3. Svédország.....	29
2.7.4. Dánia.....	29
2.7.5. Franciaország.....	29
2.7.6. Svájc.....	29
2.7.7. Németország.....	29
2.7.8. Japán.....	30
2.7.9. Ausztria.....	30
2.7.10. Csehszlovákia.....	30
2.7.11. India.....	30
2.7.12. Izrael.....	30
2.7.13. Lengyelország.....	30
2.7.14. NDK.....	30
2.7.15. Románia.....	30
<b>2.8. ELINDUL A SZÁMÍTÓGÉPEK IPARI GYÁRTÁSA</b> .....	<b>30</b>
2.8.1. A UNIVAC – gép.....	30
2.8.2. Az IBM 701.....	31
2.8.3. Más gyártók.....	32

2.9. Az első generációs számítógépek fontosabb jellemzői .....	32
<b>3. FEJEZET: A SZÁMÍTÓGÉPEK 2.-5. GENERÁCIÓJA.....</b>	<b>34</b>
3.1. A SZÁMÍTÓGÉPEK MÁSODIK GENERÁCIÓJA (1959-1965).....	34
3.1.1. A TX-O gép.....	34
3.1.2. A PDP-1.....	34
3.1.3. Az IBM 7090 gép .....	35
3.2. A HARMADIK GENERÁCIÓS SZÁMÍTÓGÉPEK (1965-1975).....	36
3.2.1. Az IBM – 360.....	37
3.2.2. A DEC PDP sorozat .....	37
3.2.3. A jelentősebb harmadik generációs gépek: .....	37
3.2.4. Az ESzR gépek.....	38
3.3. A NEGYEDIK GENERÁCIÓ KORSZAKA .....	38
3.4. Ötödik generációs számítógépek .....	39
<b>4. FEJEZET: A SZÁMÍTÁSTECHNIKA MAGYARORSZÁGON.....</b>	<b>40</b>
4.1. JUHÁSZ-FÉLE LŐELEMKÉPZŐ .....	40
4.2. A HAZAI SZÁMÍTÁSTECHNIKÁT (INFORMATIKÁT) MEGALAPOZÓ SZAKIRODALMAK.....	40
4.3. ADATFELDOLGOZÁS.....	41
4.4. Fejlesztések.....	41
4.5. MTA KIBERNETIKAI KUTATÓCSOPORT (MTA KKCs).....	42
4.5.1 Az M-3 gép jellemzői.....	43
4.5.3. Az első „éles” alkalmazások .....	46
4.6. TELEFONGYÁR.....	47
4.7. VILLENKI (VILLAMOS-ENERGIAI KUTATÓINTÉZET) .....	47
4.8. KFKI (KÖZPONTI FIZIKAI KUTATÓ INTÉZET) .....	47
4.9. EMG (ELEKTROMOS MÉRŐKÉSZÜLÉKEK GYÁRA).....	48
4.10. VASIPARI KUTATÓINTÉZET .....	48
4.11. MÉRÉSTECHNIKAI KÖZPONTI KUTATÓ LABORATÓRIUM .....	48
4.12. Budapesti Műszaki Egyetem (BME) .....	48
4.13. SZÁMÍTÁSTECHNIKAI KÖZPONTOK .....	49
4.13.1. MTA KKCS.....	49
4.13.2. Nehézipari Minisztérium .....	49
4.13.3. Kohó és Gépipari Minisztérium.....	50
4.13.4. Számítástechnikai és Ügyvitel-szervezési Vállalat (SZÜV) .....	50
4.14. ESEMÉNYEK, SZERVEZETEK .....	50
4.15. KÉPZÉSEK.....	50
4.16. A HAZAI SZÁMÍTÓGÉPGYÁRTÁS .....	51
4.16.1. A TPA család .....	51
4.16.2. Az EMG-830-as számítógép .....	51
4.16.3. Az ESzR gépek Magyarországon.....	51
<b>5. FEJEZET: PROGRAMOZÁSI NYELVEK .....</b>	<b>52</b>
5.1. GÉPI KÓD, ASSEMBLER.....	52
5.2. MAGAS SZINTŰ (MAGASSZINTŰ) NYELVEK.....	52
5.2.1. Babbage-től, Ada Lovelacetől a FORTRAN-ig .....	52
5.2.1.1. Babbage, Ada Lovelace terve .....	52
5.2.1.2. Quevedo nyelve .....	52
5.2.1.3. Turing formális nyelve .....	52
5.2.1.4. A Plankalkül .....	52
5.2.1.5. Neumann – Goldstine koncepciója .....	55
5.2.1.6. Burks „közbülső” programnyelve.....	58
5.2.1.7. Brooker Autokódja .....	59
5.2.1.8. H.B.Curry programnyelve.....	60
5.2.1.9. Rutishauser program nyelve.....	61
5.2.1.10. Böhm Compiler-nyelve.....	61
5.2.1.11. A Ljapunov-féle operátor séma.....	62

5.2.1.12. Egy deklaratív nyelv: az ADES .....	64
5.2.1.13. Formula vezérelt számítógép.....	64
5.2.2. Az életben maradt ősök és néhány leszármazott .....	66
5.2.2.1. A FORTRAN 0.....	66
5.2.2.2. A nagy „ősnyelv”, az ALGOL .....	67
5.2.2.3. A COBOL nyelv .....	67
5.2.2.4. A BASIC nyelv.....	69
5.2.2.5. A PL/1 „világnyelv” .....	69
5.2.2.6. LISP deklaratív (funkcionális) nyelv .....	69
5.2.2.7. Az APL .....	69
5.2.2.8. Az RPQ.....	70
5.2.2.9. A PASCAL .....	70
5.2.2.10. A MODULA-2.....	70
5.2.2.11. ADA.....	70
5.2.2.12. A C programnyelv .....	70
5.2.2.13. Feladatorientál programnyelvek .....	70
<b>IRODALOMJEGYZÉK: .....</b>	<b>72</b>

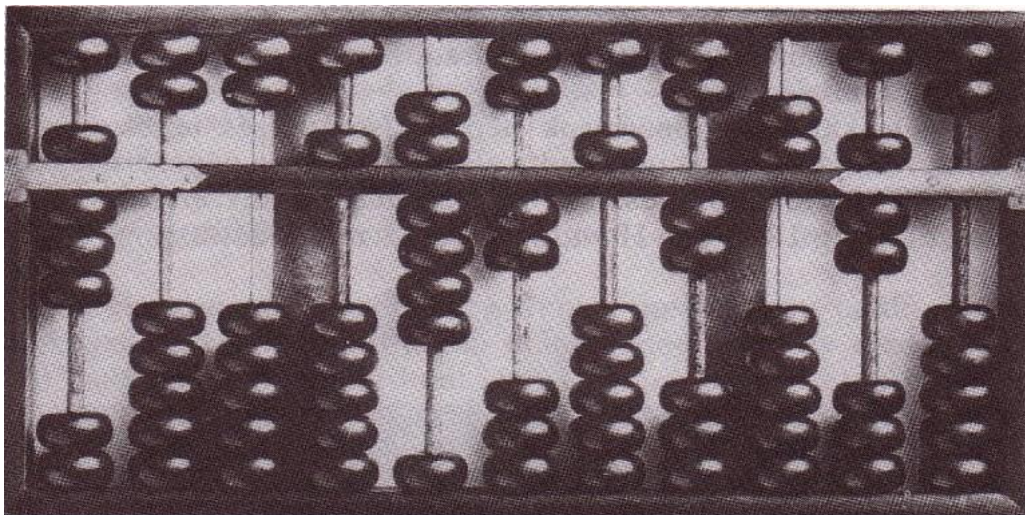
# 1 . F E J E Z E T

## AZ ABAKUSZTÓL AZ ENIAC-IG

### 1.1. AZ ÓKOR ÉS KÖZÉPKOR

#### 1.1.1. Az ókortól kezdve próbálták gépesíteni a számolást

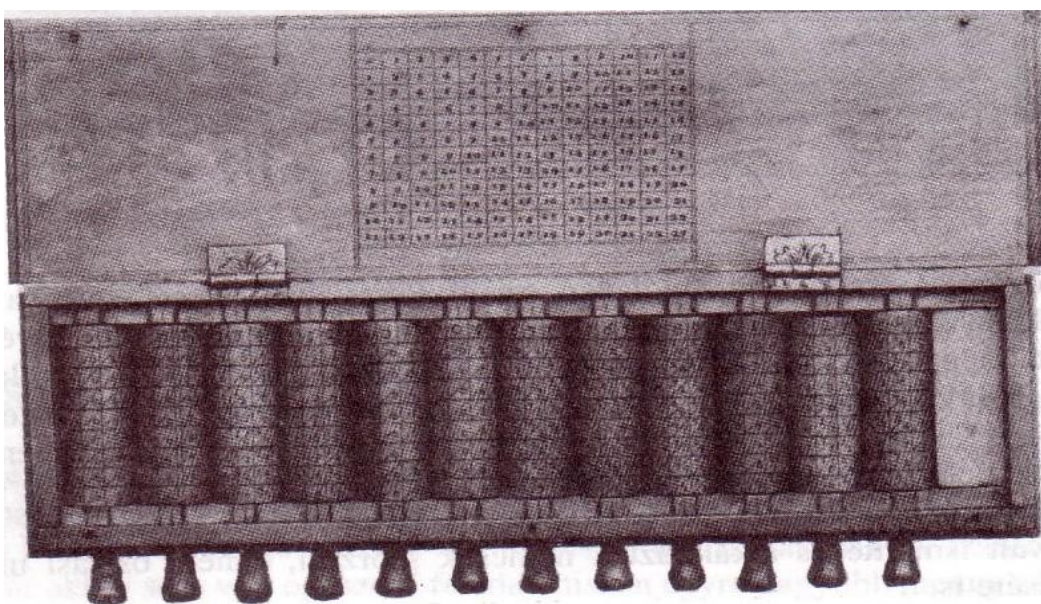
**Abacus:** kavicsok, drótokon, huzalokon kellett tolni a golyókat. Több ezer éve létezett, a XX. század közepéig „élt”.



1.1. ábra: Az Abacus

#### 1.1.2. A „Napier pálcák” számológép

John Napier skót matematikus (a logaritmus elvének kidolgozója) csont és farudacskákból épített számológépet, amellyel nagy számokat is lehetett szorozni, osztani. Nem terjedt el annyira, mint az Abacus.



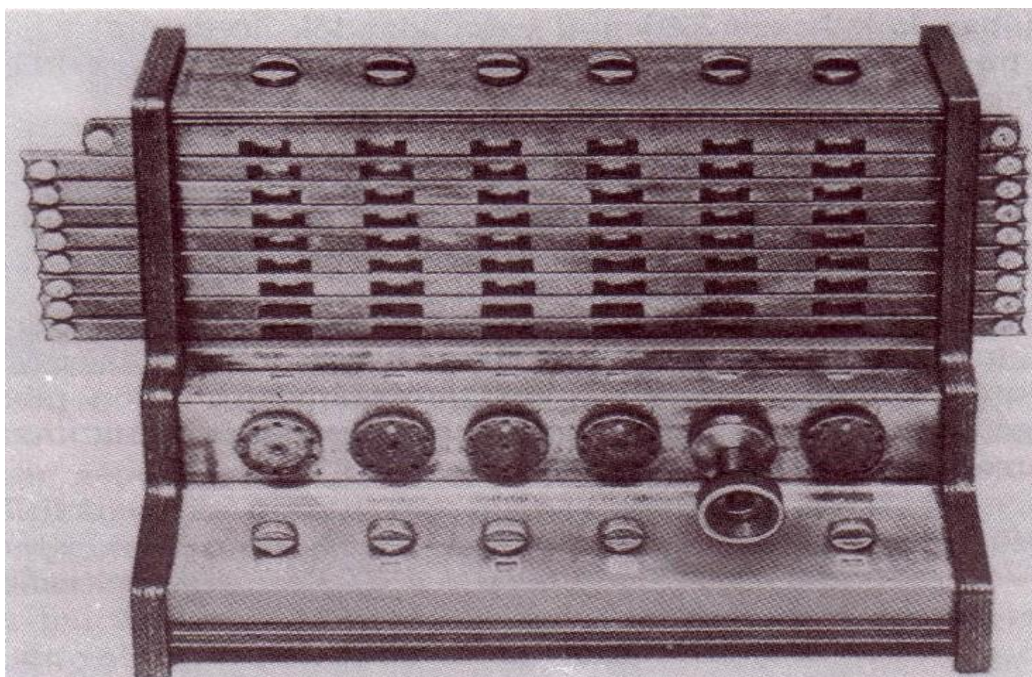
1.2. ábra: Napier pálcák

### 1.1.3. Schickard mérnök (1592-1635)

A XVII. században épített **számológépet**.

Alapja: 10 ágú fogaskerék rovátkolt lécek között.

- fölfelé tolva: összeadás,
- lefelé tolva: kivonás.
- Az átvitelt is ügyesen megoldotta (1 fogú fogaskerék, amely meghajtja a szomszédos számkereket)
- 10-es számrendszer
- Ezt tekintik a későbbi Leibniz kerék ősének



1.3. ábra: Schickard gépe

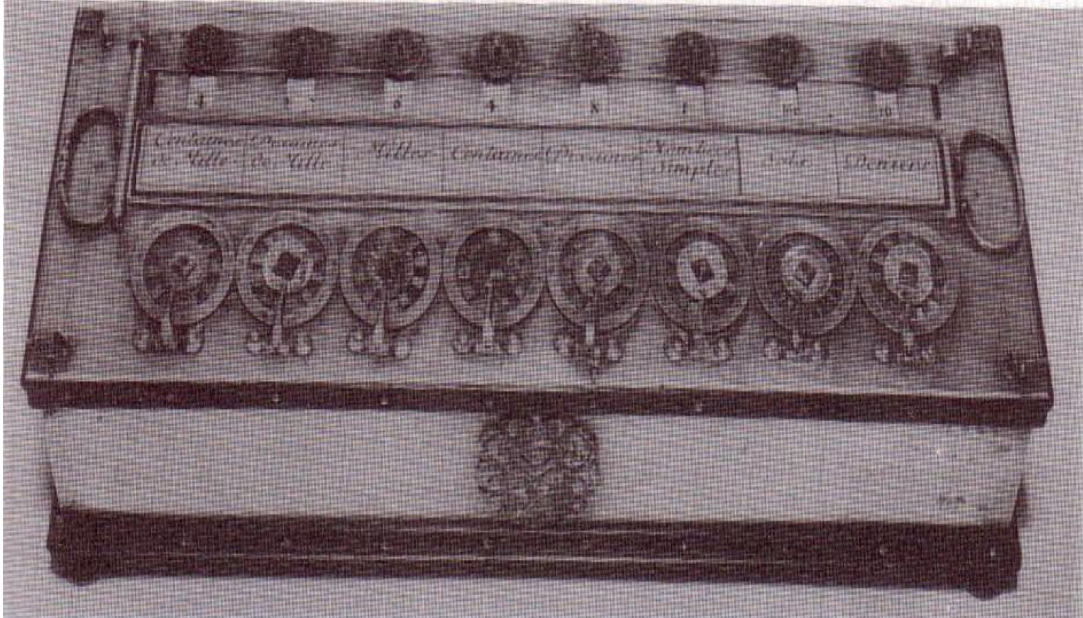


1.4. ábra: Schickard

#### 1.1.4. Pascal (1623-62)

A Pascalin nevű számológép fűződik a nevéhez

Ua. a filozófia, mint Schickardé: 10-es átvitel, de jobb megoldással  
Filozófia jelentőség (Descartes) anyag és szellem közötti kapcsolat



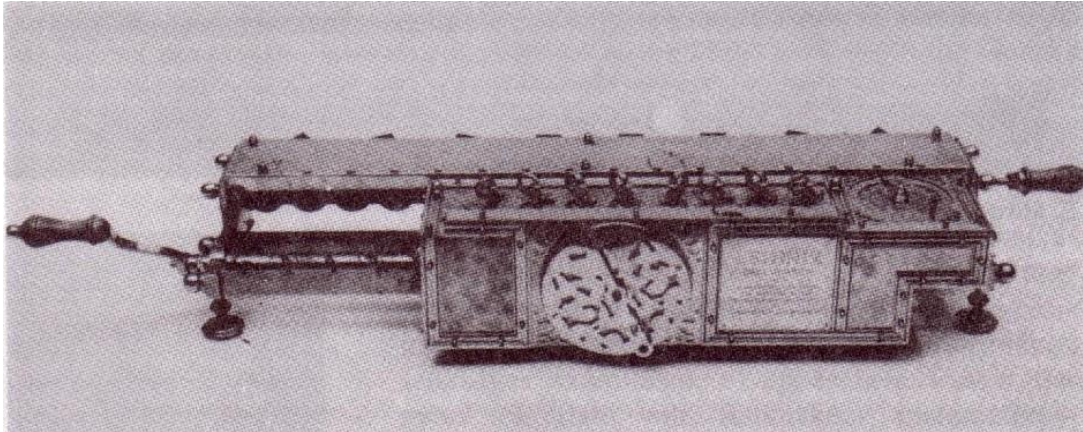
1.5. ábra: Pascalin



1.6. ábra: Pascal

### 1.1.5. Leibniz (1642-1716)

A Pascalin tökéletesítése (Leibniz-kerék)



1.7. ábra: A Leibniz gép



1.8. Leibniz

### 1.1.6. Müller

(XVIII. században) egy differencia összeadó gépet tervezett

- külön tengelyekre szerelt fogaskerek (számkerék) - egyformák
- később közös tengelyen 10 állapotú különböző kerek (mint a kuplung)
- 2 alpművelet

## 1.2. A SZÁMÍTÁSTECHNIKA ŐSATYJA: CHARLES BABBAGE (1791-1871)

### 1.2.1. Élettörténet

- 1810-ben, *Cambridge*-ben (Trinity College) kezdte tanulmányait;
- 1812-ben, Angliában megalapította az *Analitikai Társaságot* (feladata: a modern matematikai eredmények terjesztése);



- 1813-ban átmegy a *Szent Péter College*-be;
- 1814-ben megszerzi a *Bachelor's Degree*-t (diplomát);
- 1817-ben, *Londonban* professzori állást kap;
- 1819-ben *Edinburgh*-ba megy;
- 1820-ban megválasztják a *Royal Society* (Tudományos Akadémia) tagjának;
- 1828-ban visszamegy *Cambridge*-be, kinevezik a *matematikai professzorává*;
- 1829-ben „átcsap mérnöknek”, nekilát a számítógépek tervezésének.

### 1.2.2. Babbage munkásságának egyéb területei:

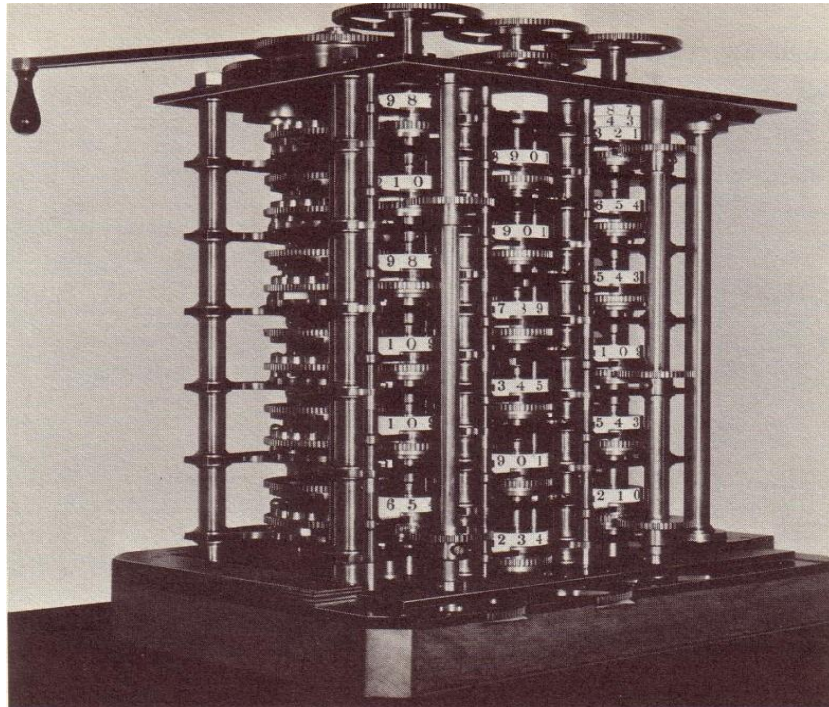
- Függvénytáblázatok készítése (pl.: logaritmustábla);
- Halálózási táblázatok biztosítók számára (évtizedeken keresztül használták);
- Közgazdaságtan: *On the Economy of Manufactureries and Machinery* c. könyvében ipari, gazdasági, termelési folyamatok optimalizálásáról írt;
- Színpadi effektusok létrehozására gépet épített;
- Különbéle automaták (malom-játék);
- Adatkódolás / dekódolás;
- Társadalomtudományi cikkeket is írt;
- Víz alatti úszóhajó (tengeralattjáró) tervezése;
- Világítótorony fényének ki/bekapcsolására automatát készített.

### 1.2.3. Véges differenciák összeadó gépe (Difference Engine)

Akkoriban rengeteg táblázatot használtak (hajózás, bank, biztosítás), de sok hiba volt bennük. (1834-ben 140 függvény tábla közül kiválasztottak 40-et, ezekben több mint 3700 hiba volt.)

*Véges differenciák módszere*: adott pontokban megadott függvényértékekből közelítőleg közbülső függvényértékeket lehet kiszámítani a táblázatban látható differenciák segítségével. (Newton Leibniz formula ezek alapszik.)

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
$x_1$	$y_1$			
$x_2$	$y_2$	$y_2 - y_1$		
$x_3$	$y_3$	$y_3 - y_2$	$\Delta y_3 - \Delta y_2$	
$\vdots$	$\vdots$			
$x_i$	$J_i$	$y_i - y_{i-1}$	$\Delta y_i - \Delta y_{i-1}$	
$\vdots$	$\vdots$			$\Delta^2 y_i - \Delta^2 y_{i-1}$
$x_n$	$y_n$			



1.9. ábra: A differencia gép



1.10. ábra: Babbage

„Igazi” matematikai gép. Folyamatosan, automatikusan számít ki függvényértékeket.

- 1822-ben elkészül a gép működőképes modellje; £1500 támogatást kap a Royal Society-től egy jobb változat elkészítéséhez;
- 1923-27 között műhely épül;

- 1833-ban vita támadt Babbage és Clement főmechanikus mérnök között, 1833 végén azonban sikeres bemutatót tudtak tartani, a gép 2,5 perc alatt 30 függvényértéket számított ki;
- 1834 elején Babbage és Clement között ismét kiéleződtek a viták, emiatt a végleges példány nem készült el;
- 1862-ben, Londonban kiállítja Difference Engine-t; ismét támogatást kér, de nem kap (£12000 + £8000 a műhelyre).

#### 1.2.4. Az Analitikai Gép (Analytical Engine)

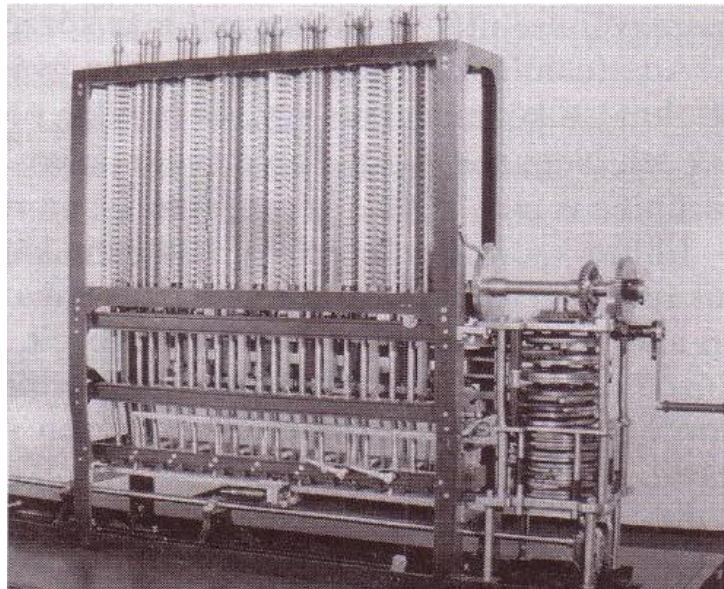
1833-ban Babbage úgy dönt, hogy nem a differencia összeadó befejezésébe fekteti energiáját, hanem új konstrukció, mechanikus, univerzális (külsőleg programozható, digitális) gép tervét készíti el.

##### A gép fő egységei:

1. **Memória** („store”) 1000 db. 50 számjegű decimális szám tárolására,
2. **Aritmetikai egység** („Mill, malom”) négy alpművelet (mind a négy hardveres módon),
3. **Vezérlő egység** (lyukkártya legyen, ami kívülről mozgat);
4. **Beolvasó** („Input”) egység (lyukkártya);
5. **Nyomtató** (lyukkártyás)

##### Sebesség:

- a) Összeadás: 60 / perc (50 jegű számokkal);
- b) Szorzás/osztás: 1 / perc.



1.11. ábra: Az Analitikai gép

**Meghajtás:** ejtősúlyal, vagy gőzgéppel;

- 1838-ban a gép teljes műszaki rajza elkészült;

- 1840-ben Babbage előadó körutat tesz Olaszországban. *Menabrea*, olasz tudós jegyzetbe foglalja Babbage előadását, ezt a jegyzetet *Ada Augusta Lovelace* angolra fordította, kiegészítette.
- 1842-48: folytatja az Analitikai gép építését;
- 1849-52: visszatér a Difference Engine-hez;
- 1852: A kormányzat végleg megszüntet mindennemű támogatást;
- 1871-ben meghal Babbage; a rajzokat és a műhelyt fiára, Henry Prevost Babbage-re hagyja, aki 1879-ben mindent megsemmisít, de 1880-tól kezdve újra elkezdi foglalkozni a Mill-gép építésével. Az 1888-ban a felépített Mill-géppel a  $\pi$ -t 41 számjegy pontossággal kiszámítja;
- 1896: A „Babbage” korszak vége; a Science Museum (London) ad otthont a megmaradt gépeknek, szerszámoknak;
- 1878-ban mások is óriási munkát fektetnek az Analitikai gép megépítésébe (Babbage-től függetlenül); többnyire azt a nézetet vallották, hogy nem nagy gépet kell építeni, hanem kisebb célgépeket.

### 1.3. ADA AUGUSTA LOVELACE (1815-1852)

- A Számítógépes algebra „ősanyja”, az első nő programozó, Byron (angol költő) lánya volt, Babbage javaslatára matematikát kezdett tanulni; De Morgan tanítványa volt;
- 1834-ben bekapcsolódott a Difference Engine munkálataiba is. 1841-ben részletesen tanulmányozni kezdte az Analitikai gépet; 1842-ben a *Menabrea*-cikket angolra fordítja, Babbage tanácsára kibővíti. Az Analitikai gép építését nemcsak szellemileg, de anyagilag is támogatja.



1.12. ábra: Ada Lovelace

### **Kiegészítések a Menabrea-jegyzethez:**

*A Menabrea-cikk 1843-ban jelent meg; Ada Byron „kiegészítése” háromszorosa Menabrea jegyzetének. Főbb gondolatai:*

- Az Analitikai gép tudományos jelentősége (Ada Byron szerint a gépek képesek lesznek zenét szerezni);
- Szimbolikus rendszert javasol a gép működésének, állapotának leírására (Gyakorlatilag az Assembly programozási nyelv őst adja meg; alpműveletek, vezérlésátadás);
- Ciklus (hurok) megvalósítása visszaléptetéssel;
- Felírta egy kétismeretlenes egyenlet megoldóprogramját;
- Bevezette a *munkarekesz* fogalmát (a munkarekesz részeredmények tárolására szolgált);
- Felveti az analitikai gép nem numerikus alkalmazásának lehetőségét (algebrai alkalmazások);
- Filozófiai jelentőség: a számításokat a gép sokkal jobban, hatékonyabban végzi majd el, mint az ember; újat nem alkot a gép, de amire „beprogramozák”, azt elvégzi;
- Az általa alkotott nyelven programot írt a Bernoulli-számok kiszámítására.

### **Fogalmak összefoglalása:**

1. Store (tár);
2. Mill (malom; aritmetikai egység);
3. Munkarekesz;
4. Ciklus;
5. Számítógépes algebra;
6. Intelligens gép.

## 1.4. BABBAGE-TŐL AZ ENIAC-IG

### 1.4.1. Differencia-összeadók tervezése

#### a. Svédország

- I. **George Scheutz** és fia 1834-37-ig egy *differencia-összeadót* tervezett, 1937-ben pedig létrejön egy működő modell is; 1853-ban elkészül az ipari gép, 1854-ben Londonba szállítják; Babbage is látta a gépet; később az USA-ba szállították. 1860-ban a Brit kormány £1200-t adott a gép továbbfejlesztésére, így elkészült a *Scheutz-II*; azonban sok hiba volt benne. Bár a gépet „ipari” méreteken gyártották, mégsem aratott teljes sikert;
- II. Martin Wiberg: Differencia-összeadót készített, mely alkalmas volt hétjegyű függvénytáblázatok készítésére (hajózáshoz és biztosításokhoz).

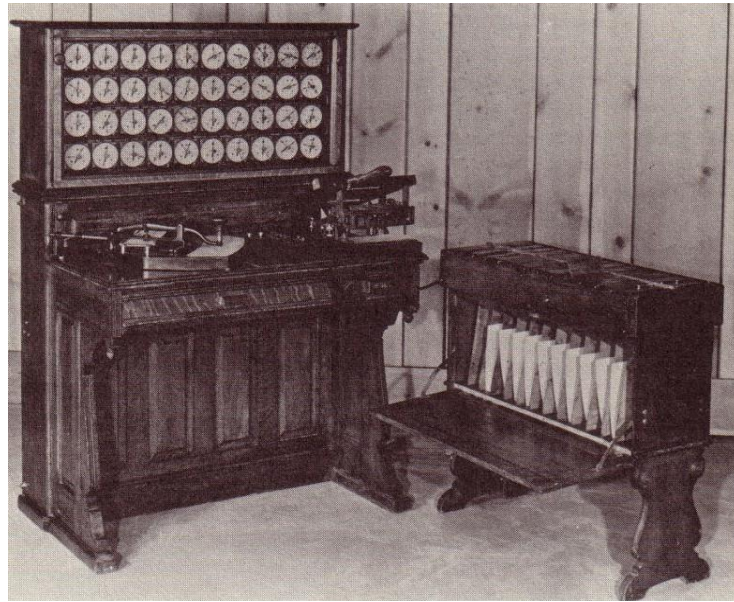
#### b. Németország

- I. **Hamann** (mérnök) készített differencia-összeadó gépet, ez azonban cél-gép volt, csak másodrendű differenciákat használ; bár kézi meghajtással működött, mégis könnyen kezelhető gépről van szó. Egy perc alatt hat függvényértéket tudott kiszámítani;
- II. **Brunsviga-gépek**. Differencia-összeadó gépek elvén működtek; 1928-ban pedig már elektromos Brunsviga-gépeket is gyártottak. Kereskedelmi-üzleti alkalmazások, statisztikai feldolgozások, adatfeldolgozás el-látására voltak alkalmasak.

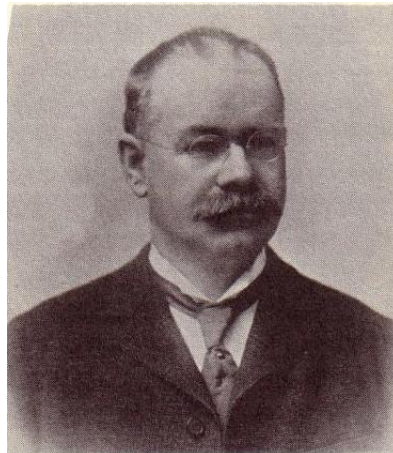
### 1.4.2. Kereskedelmi, üzleti gépek, adatfeldolgozás: a Hollerith-gépek

Hermann Hollerith (1860-1929) az USA-ban, 1890-ben, népszámlálási célra készítette a gépet (pályázatra; 56 000 000 db. lyukkártya feldolgozására kellett alkalmasnak lennie);

- Ötlet: lyukkártya;
- Találmány: *Pantográf* nevű lyukasztógép;
- Működés: ez a gép már elektromos volt, és a kártyák tűkkel való letapogató-ása elvén működött;



1.13. ábra: A Hollerith gép



1.14. ábra: Hollerith

- Programozhatóság: dugaszolással (régii telefonközpontokhoz hasonlóan); kártyaválogatásra is be lehetett programozni. Az 1930-as években már a programozás is lyukkártyával történt;
- Gépparkok: kártyalyukasztó, rendező, *tabulátor*, automatikus kártyatovábbítás;

**Hollerith** 1896-ban saját céget alapít (Tabulating Machine Company);

1911-ben a cég egyesül egy óragyárral és egy mérleggyárral; (neve: Computing Tabulating Recording Company)

1917-ben, Kanadában létrejön egy leányvállalat, amely 1927-től kezdve az *IBM – International Business Machines* nevet veszi fel.

### 1.4.3. Az Analitikai gép öröksége

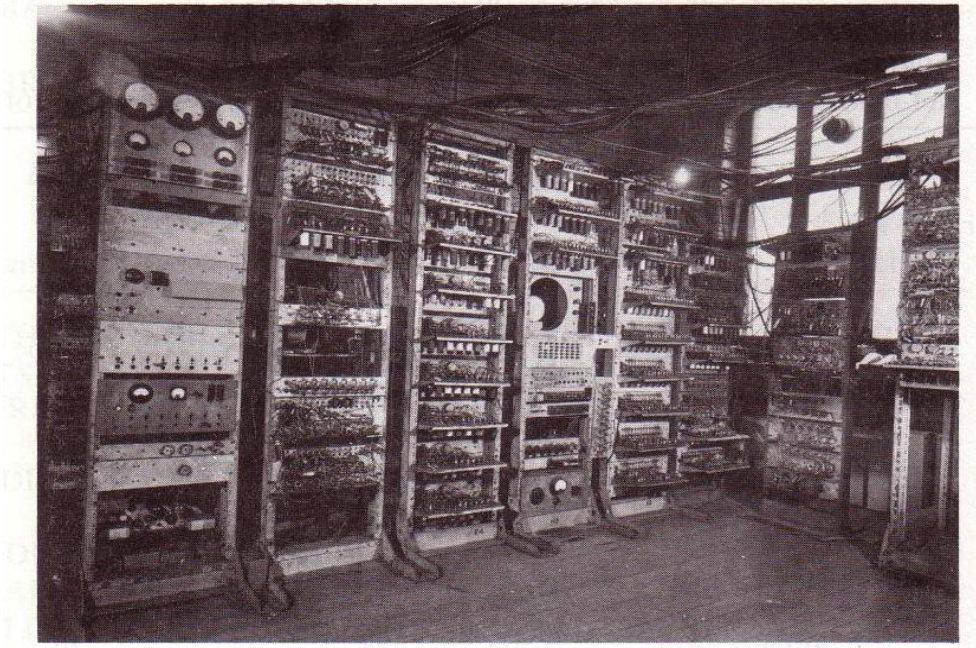
#### 1.4.3.1. H. Aiken (1900-1973, USA)

1937-ben kidolgoz egy tervet univerzális számítógép építésére;

1939: IBM-mel együttműködésben megépül a **MARK-1** gép (*Automatic Sequence Controlled Calculation [ASCC]*);

1943: MARK-1 bemutatása;

1944-1959: a gépet használják.



1.15. ábra: A MARK-1

##### *Jellemzői:*

Mechanikus elektromos meghajtással

60 db regiszteres tároló (23 decimális jegy + előjel)

72 db összeadó regiszter;

2 db szorzóregiszter (vezérlése lyukszaggal);

Hossza: 16 méter; tömeg: 5 tonna; magasság: 2,5 méter; hűtés: napi 2-3 tonna jéggel;

*Utódai:* MARK-II, MARK-III, MARK-IV.

#### 1.4.3.2. Németország

- a) **Konrad Zuse** 1938-ban megépíti a Z1 (ZUSE-1) gépet. Ez a gép teljesen mechanikus, de már bináris;
- b) Z2-es gép: ez már reléekkel működött (Akkoriban ez a csúcstechnológiák egyikének számított);
- c) 1939: Zuse-t besorozzák katonának. Miután elmondja, hogy számítógépek készítésén fáradozik, heteken belül leszerelik, majd állami szolgálatba állítják, hogy a hadseregnek gyártson gépeket;
- d) 1941: Z3-as gép megépítése. Ez volt a világ első, belső programozású gépe;



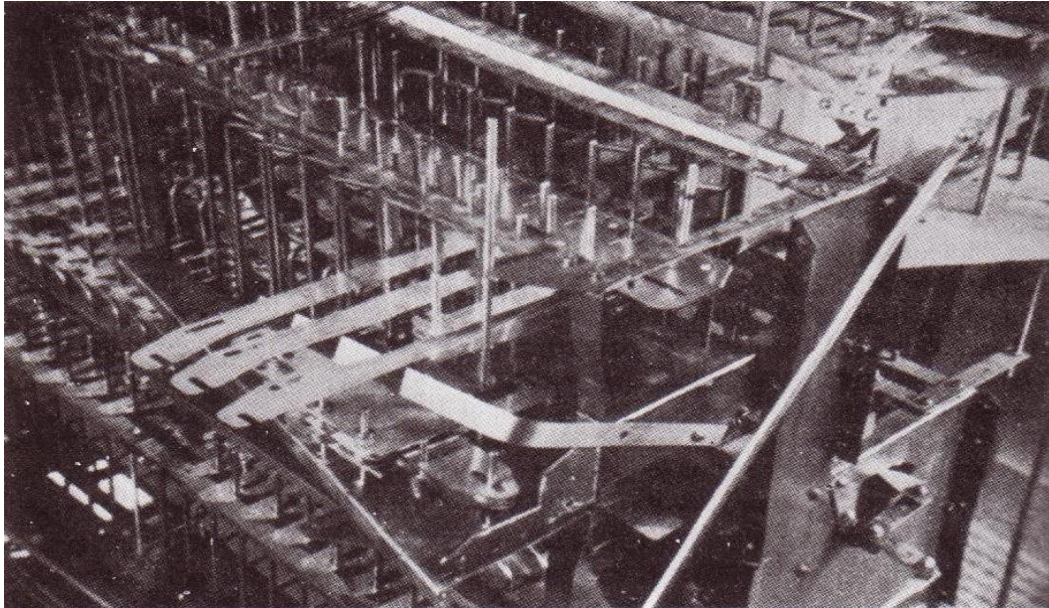
e) 1943: Z4-es gép: Zuse ezt már bunkerben építette meg; ez volt az első igazán elektronikus számítógép;

f) **Zuse jelentősége:**

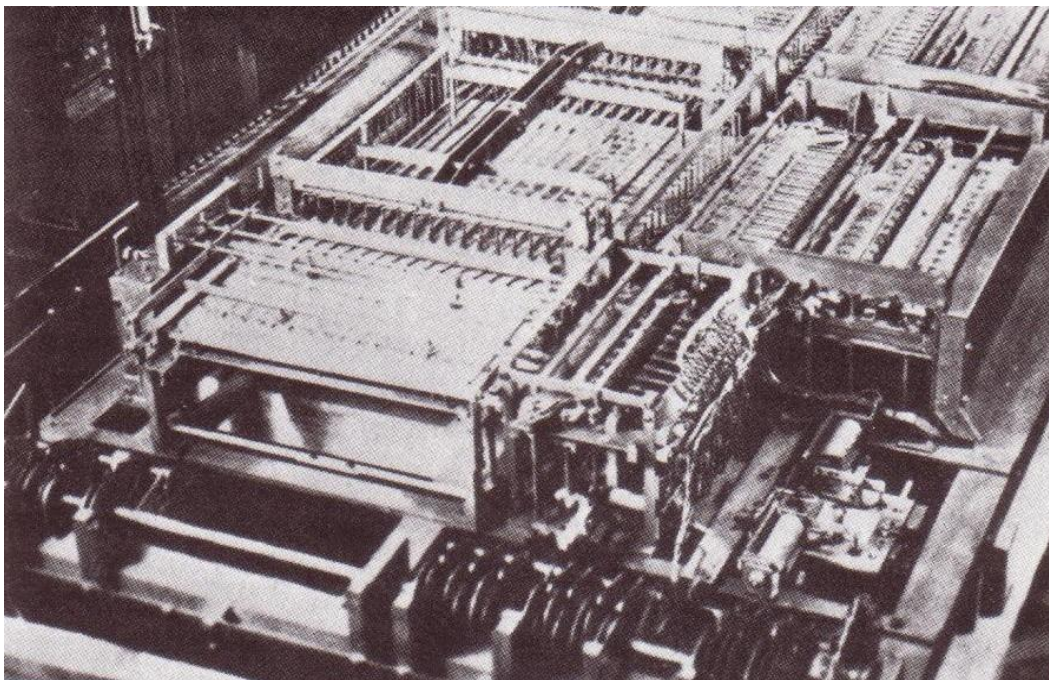
Ténylegesen működő gépeket épített, és ezek bináris számrendszert használtak.

Elsőként vezette be a lebegőpontos számábrázolást.

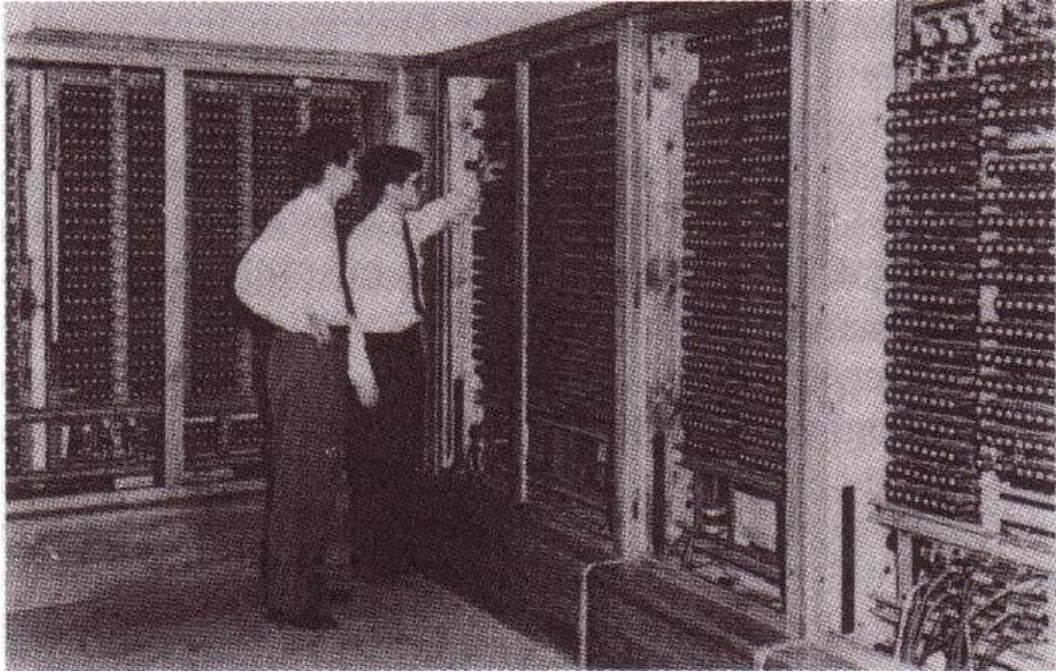
A Z4-hez már programozási nyelvet is kidolgozott (*Plankalkül, lásd később*).



1.16. ábra: A ZUSE-1



1.17. ábra: A ZUSE-3 mechanikus memóriája

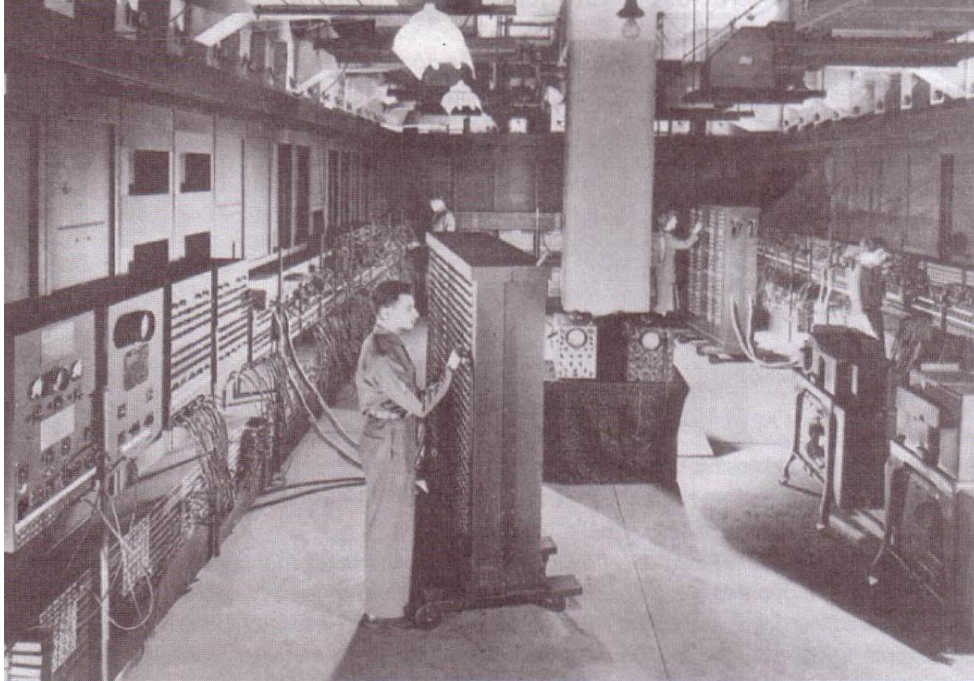


1.18. ábra: A ZUSE-4

## 2 . F E J E Z E T

### ÚJ KORSZAK, MEGJELENIK AZ ELEKTRONIKA

Az elektronika megjelenése a számítógépekben új korszakot nyit. Ha a számítástechnika történetét szakaszokra bontjuk, akkor az elektronika „belépése” az **első generációs** gépek korszakának kezdetét jelenti. Ez a generáció a következő nagy váltásig tart: ez a tranzisztor alkalmazását jelenti a számítógépekben.



2.1. ábra: Az ENIAC gép

## 2.1. AZ ELSŐ ELEKTRONIKUS SZÁMÍTÓGÉP, AZ ENIAC

ENIAC: *Electronic Numerical Integrator and Automatic Computer*. Az USA-ban építették meg. Készítő, tervező: Presper Eckert, John Mauchly (+Hermann Goldstine, Arthur Burks, [Neumann János]);

a) 1943: a tervjavaslat elkészült. Minden terv a *Moore School of Electronic Engineering University of Pennsylvania* intézményében készült el és került kivitelezésre. Beruházó: *Ballistic Research Laboratory of Aberdeen* – (Amerika legnagyobb hadiipari intézménye.) A gép 1945-re készült el.

### b) **Jellemzői:**

Teljesen elektroncsöves áramkörökből felépített gép (18000 db elektroncső).

Különálló, ún. fél-autonóm egységekből állt.

Részben elosztott vezérlés.

Külső programozású gép volt dugaszolással.

Az Input/Output lyukkártyával működött.

A gép decimális számábrázolást használt.

Nem Neumann-elvű gép volt.

### c) **A gép egységei:**

- 20 db akkumulátor (regiszter), 10 db decimális számjegy + előjel;
- 3 db függvénytáblázat volt beleégetve, darabonként 104 függvényértékkel;
- 1 db szorzóegység volt benne;
- 1 db osztó és gyökvonó egységgel rendelkezett;
- 1 db „konstans küldő” egységet is tartalmazott ( $\pi$ ,  $\epsilon$ , stb.);
- Inicializáló egység;
- Központi programadó;
- 1 db kártyaolvasó;

- 1 db kártyalyukasztó.

d) **Tesztfeladatok:**

- Számok négyzeteinek és köbeinek kiszámítása, minden hetedik eredmény kinyomtatása 200-szor.
- Az  $y'=y$  ( $y(x_0)=y_0$ ) differenciálegyenlet megoldása (Euler-módszerrel).

e) **Megjegyzés:**

- Ez a gép már nagyon megbízhatóan (90%) működött;
- Nagyon gyorsan (2,5 év alatt) készült el.

f) **Műszaki adatok:**

- Összeadás: 200  $\mu$ s alatt;
- Szorzás: 2,8 ms alatt I;
- Osztás: 2,4 ms alatt;
- 125 kártya/perc sebességgel vitte be az adatokat;
- A kártyalyukasztó 100 kártyát tudott kilyukasztani 1 perc alatt;
- Fogyasztása 174 kW (egy kisebb település fogyasztása) volt;
- Területe: 180m<sup>2</sup>;
- Helye: Philadelphia, Los Alamos (az atombomba is itt készült) számára dolgozott, majd Aberdeenbe vitték át (1955-ig működött).

g) **Megjegyzés:**

A tervezők 1943-44-től egy új gép tervein dolgoznak; Mauchly, Eckert saját céget alapítanak, amit a Remington Sperry cég felvásárolt; az UNIVAC – gépcsalád itt alakult ki.

## 2.2. NEUMANN JÁNOS A SZÁMÍTÓGÉPEK ATYJA

### 2.2.1. Rövid életrajza

*Neumann János* Budapesten született, bankárcsaládban. Magántanulóként kezdte meg tanulmányait, majd az Evangélikus Fasori Gimnáziumba került. Tanárai: Rácz, Fekete, Kürschák voltak.

1921-ben érettségizik, de ekkor már profi matematikusnak tekintették, akinek több cikke is megjelent neves folyóiratokban.

Két egyetemre is beiratkozik: Pázmány Péter Tudományegyetem Budapest, (matematika) és Zürich, Technische Hochschule (vegyész).

1926-ban 2 doktorátust szerzett; egyiket Budapesten, másikat Zürich-ben.

1929-ben a Hamburgi Egyetemre kerül, matematikusként.

1930-ban látogatást tesz az USA-ban, (Princeton), ahol egy éves szerződést köt, de további két évvel meghosszabbítja; csak 1933-ban tér vissza Európába.

1938-ban visszatér Princetonba, és számos katonai projektben vesz részt.

1954-ben beválasztják az USA Atomenergia Bizottságába.

1955-ben megbetegszik (rákot kap), 1957. február 8-án meghal.



2.2. ábra: Neumann János

### 2.2.2. Szakmai területek

#### a) *Matematika, fizika*

- Algebra és halmazelmélet,
- Valós függvénytan, mértékelmélet,
- Topológia (térelmélet),
- Absztrakt terek, algebrai csoportelmélet,
- Hálóelmélet,
- Numerikus analízis,
- Automaták elmélete,
- Játékelmélet, közgazdaságtan,
- Meteorológia, hidro-és aerodinamika,
- Matematikai-fizika,
- Kvantummechanika.

#### b) *Számítógépek, a számítógépek elméleti oldala*

- 1947: megjelent *Neumann-Goldstine* könyve: *Numerical inverting of matrices of high order*.
- 1948-ban az automaták általános logikai elméletéről tartott előadást, majd ezt utána cikkben is megjelentette.

*Kimondta:* a mesterséges automaták tisztán digitális működésűek lesznek, mert az analóg gép nem tud igazán pontos számításokat végezni; továbbá azt is kimondta, hogy az élő sejtek részben digitálisak, részben analógok.

*Tétel:* Ami egyáltalán formalizálható, az leírható formális neuronok segítségével (A Mc Culloch, Pitts által kidolgozott formális neuronok vizsgálata alapján).

*A Turing-géppel kapcsolatos vizsgálatok.*

*Kérdés:* Lehet-e Önmagát reprodukáló automatát készíteni? *Válasz:* igen, létre lehet hozni egy önmagát reprodukáló, formális automatát.

Három fő komponensből kell állnia:

- I. Leírás (*mikroprogram*).
- II. Az automatának képesnek kell lennie kiolvasni, értelmezni és megérteni saját mikroprogramját.
- III. Az I. – pontot be kell építeni a II. – pontba, majd szabadjára kell engedni a rendszert (II.) → (I.).
- IV. Az I; II; III. – pontot együtt is „be kell tenni” i. – pontba (I. v II. v III.) → I.

– Az automata tehet hibás lépéseket is, ami az élő szervezetekhez hasonlóan mutációt okozhat. Neumann kimondja, hogy a mikroprogramban lévő leírás ugyanolyan, mint az élő sejteknél a genetikai kód. Ezt a munkát A. Burks Neumann halála után 1966-ban megjelentette, *Theory of self-reproducing automata* a címmel.

– Másik kérdés: képes-e egy önmagát reprodukáló automata saját hibáit felismerni, és azokat kijavítani? Neumann bebizonyítja, hogy lehet ilyen automatát készíteni, csak egyetlen baj van vele: egy ilyen automatának végtelen (vagy majdnem végtelen) sok alkatrészről kellene állnia.

Ebből a tételből született a *Probabilistic Logics and the synthesis of reliable organisms from unreliable components* c. dolgozat (1956).

Megjegyzés: A dolgozatban a szerző felveti a *párhuzamos architektúrák* elvét, amely szerint az önmagát reprodukáló automatának párhuzamosan működő komponensekkel kell működnie.

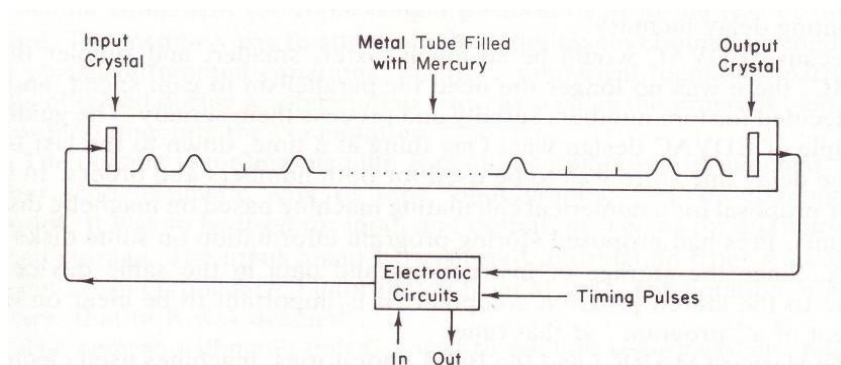
1958: *Bulletin of the American Mathematical Society* kiadta Neumann János munkáit.

### 2.2.3. Út a számítógéphez

- Neumann János az amerikai *Aberdeen* állomáson találkozott először Goldstine-nal, aki beszélt neki az ENIAC-ról; ezután Neumann is becsatlakozott az ENIAC munkálataiba (ekkor a gép már majdnem készen volt, így érthető, hogy ez a gép még nem *Neumann-elvű* volt).
- Még a *Moore School*-ban megbízzák Neumannt, hogy egy csapattal vitassa meg az ENIAC konstrukciós hibáit; ettől kezdve Neumann gyakorlatilag vezető szerepet kapott (Igaz persze az is, hogy Neumann – még mielőtt tudomást szerzett volna az ENIAC létezéséről – már levélben feszegette a Pentagonnak, hogy építeni kellene egy számítógépet). Neumann csatlakozásával új irányt kap a számítógépek tervezése, elindulnak az EDVAC gép munkálatai.

## 2.3. AZ EDVAC (ELECTRONIC DISCRETE VARIABLE COMPUTER) GÉP

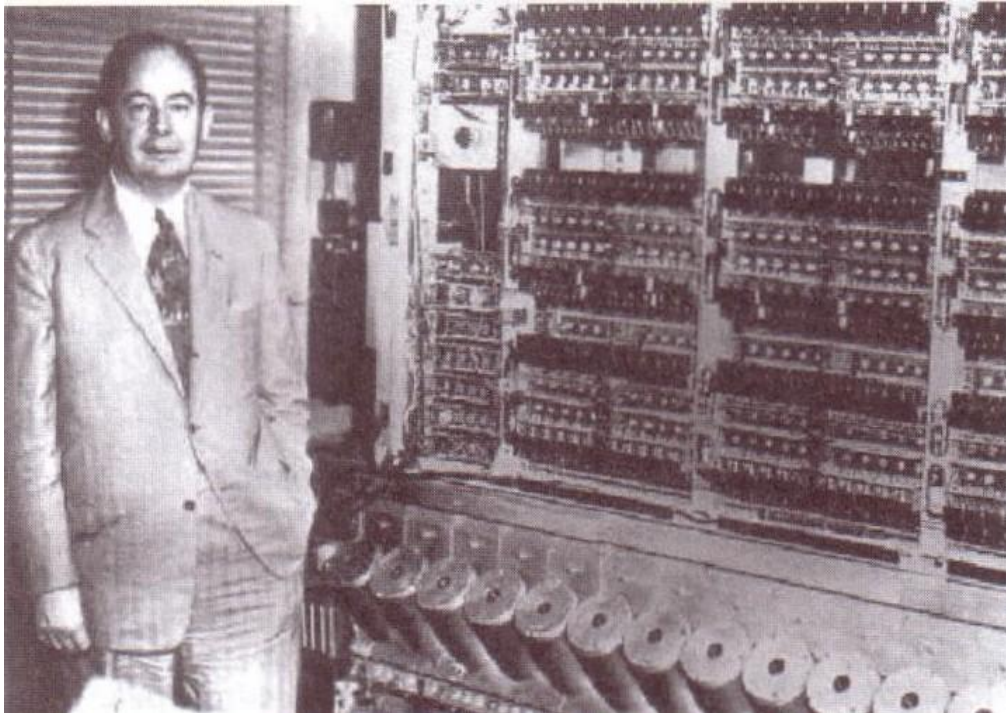
- 1944-ben Eckert és Mauchly - Neumann közreműködésével - új gépet kezdtek tervezni az ENIAC tapasztalatai alapján. Neumann felveti, hogy az ENIAC három tárolója helyett egyetlen tárolót kellene építeni, ami viszont mindent tárol.
- Erre ez egyetlen tárolóra több javaslat is adódott:
  - a) Késleltetett művonal (üvegcsőben higany és kvarc kristályok az adatok itt körforgásban vannak tárolva, nem pedig statikusan).
  - b) Elektrosztatikus (működési elve nagyon hasonló a mai tároló-lemezekéhez, azonban itt az adatot elektromosság rögzíti, nem pedig mágnesesség).
  - c) Mágnesdob (henger, ami több ezer fordulatot tesz meg percenként, felülete galvanizált; ez tárolja az adatokat).



2.3. ábra: Művonalmemória

- Az EDVAC-programozásáról
  - Mauchly könyvet ír: *Preparation of programs for EDVAC-type computers* – ez volt az első, programozási szakkönyv.
  - A szubrutin fogalmának bevezetése (*belépési pont, visszatérési pont, argumentumok helye, eredmény helye*).
  - Neumann és Goldstine már egy *formális nyelvet* is kialakítottak, ez azonban inkább egyfajta blokkdiagramm volt (lásd később)..

- Elvetették az elektrosztatikus és a mágnesdobos tárolás elvét, mert Williams mérnök (Anglia) már korábban készített művonalat, így ez volt a leggazdaságosabb megoldás.



2.4. ábra: Az EDVAC

## 2.4. A FIRST DRAFT

1945-ben Neumann egy összefoglaló jelentést írt a Pentagonnak „*First Draft of a Report on the EDVAC*” – címmel, amelyben összefoglalta azon elveket, melyeket a gép tervezésénél érvényesítettek. Az EDVAC volt az első Neumann-elvű, tisztán elektronikus működésű számítógép, azonban ez a gép sosem működött üzemszerűen.

**A 15 fejezetből álló munka tartalma röviden a következő:**

- Mit értünk **számítógépen** (részletes leírás).
- Mit értünk a **közbülső eredmények** tárolásán.
- Az **utasítás** és **program** fogalmának definiálása.
- Hibák felfedezésének és korrigálásának lehetőségei (Az önmagát reprodukáló automaták elméletének szellemében).

**A rendszer főbb komponensei:**

- Aritmetikai egység.
- Vezérlőegység.
- A program megkülönböztetése (mint utasítások sorozata).
- Egyetlen memória:
  - az aritmetikai részeredmények tárolására,



- a program tárolására,
  - függvénytáblázatok tárolására (úgy gondolta, hogy a függvényeket nem kiszámítani kell, hanem az értékeiket táblázatban tárolni),
  - az utasítások és az adatok azonos módon tárolódjanak,
  - a címezhetőség elvének megvalósítása.
- e) Beviteli egység, mely közvetlenül a memóriába visz be információt (A vezérlőegység megkerülésével).
- f) Külső tároló is kell a gépnek.

**Megvalósítási elvek:**

- a) Kétállapotú elemek (Mc Culloch, Pitts neuron-elmélete), vagyis mesterséges neuronhálózat létrehozása.
- b) *Szinkron* vagy *aszinkron* működésű legyen-e? Mivel az egész felépítését elektroncsövekre akarták alapozni, ezért Neumann a szinkron-elvet javasolta; kimutatta, hogy elektroncsővel meg lehet valósítani a szinkron-működést.

**Tervezői megfontolások:**

- a) Bináris számrendszer a megfelelő, de legyen benne konverziós lehetőség;
- b) *Belső paralelizmussal* (párhuzamossággal) kell működnie a gépnek;
- c) Neumann pontosan leírta az EDVAC utasításrendszerét is; Az EDVAC már *programozható* gép volt.

## 2.5. AZ IAS SZÁMÍTÓGÉP

A háború után Neumann Jánosnak Princetonban, az *Institute of Advanced Study* (IAS) lett a munkahelye.

1946-ban kezdődött el a First Draftban szereplő elvek szerinti gép tervezését és megvalósítását. A gépet az intézet nevéből *IAS*-nek, később *Johny*-nak hívják.

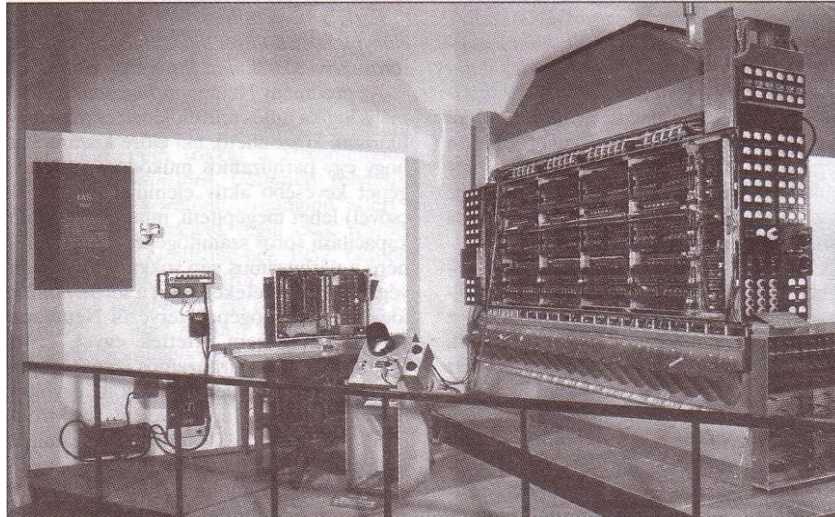
1947-ben elkészült a háttértár a mágneses huzalmemória (a mágneses huzalmemória a higanycsöves memória elvén működött, de itt higany helyett huzal volt benne).

1948-ban elkészült az aritmetikai egység, továbbá sikerült megvalósítani a központi tárolónál a párhuzamos beírást és kiírást.

Neumann eleinte elektrosztatikus memóriával kísérletezett, ami azonban nem vált be; helyette Williams memóriát alkalmaztak. Ez a memória 1024×40 bit tárolására volt alkalmas.

1950-ben elkészült a vezérlőegység.

1951-ben próbaüzemet tartottak, majd 1952-60-ig üzemszerűen működött a gép. A gép az egyetem gépe volt, főleg meteorológiai számításokat végeztek vele. Ez volt az első teljesen Neumann-elvű gép.

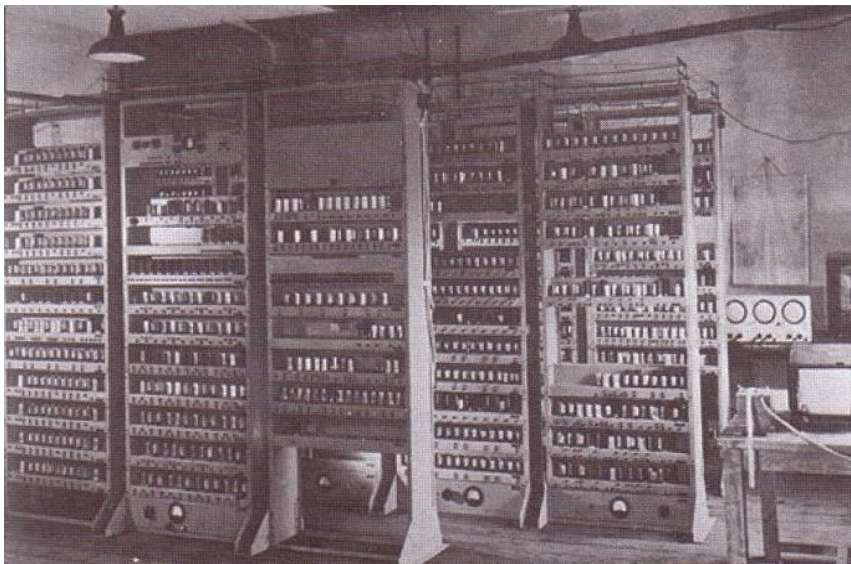


2.5. ábra: Az IAS-gép

## 2.6. AZ EDSAC (*ELECTRONIC DELAY STORAGE AUTOMATIC COMPUTER*) GÉP

A háború befejezése után – 1946 őszén – brit látogatók egy csoportja a Moore Intézetben szervezett nyári iskolába látogatott. Egyikük Maurice Wilkes megismerve a First Draftot, az ENIAC-kal és az EDVAC-kal kapcsolatos tervezési elgondolásokat, visszatérve az angliai Cambridge-be megépítette a világ első tárolt programú számítógépét az EDSAC-ot.

Ez volt az első olyan számítógép, mely vezérlésénél már *rendszerzoftverek* is megjelentek. Októberben megindultak az EDSAC tervezési és fejlesztési munkálatai. 1949-ben a gép már működött, júniusban magáról a gépről tartottak egy nemzetközi konferencia-jellegű bemutatót.



2.6. ábra: Az EDSAC gép

### a) A gép jellemzői

- 17 bites szó ( $2 \times 17$  bitből dupla szó),

- 1024 szavas memória,
- 70 bites *Akkumulátor* (Aritmetikai Egység Akkumulátor Regisztere),
- Szorzó/hányados regiszter,
- „B” regiszter: ez volt a világ első indexregisztere, mely lehetővé tette a *ciklus* megvalósítását;  
Az utasítások címezését illetően viták voltak: egycímű vagy kétcímű legyen az utasítás?
- 3000 db. elektroncső,
- Input/Output: 5 csatornás lyukszalag.

1951-ben Wilkes, Wheeler és Gill szerzőktől megjelent egy könyv, amely a *Preparation of Programs for an Electronic Digital Computer* címet viselte. Ez volt a világ első programozási kézikönyve.

### b) A gép utasításrendszere:

Kód	Hatása (magyarázat)
A n	n cím tartalma hozzáadódik az akkumulátor-regiszter tartalmához
U n	Az akkumulátor regiszter tartalma n címre íródik
T n	Ugyanaz, mint az Un, de az akkumulátor törlődik
H n	n cím tartalma beíródik a Szorzó/Hányados regiszterbe
V n	n cím tartalma szorzódik a szorzóregiszter tartalmával, majd hozzáadódik az akkumulátor regiszter tartalmához
$L 2^{n-2}$	Az akkumulátor regiszter tartalmát n bittel balra tolja
B n	n cím tartalma beíródik a "B" regiszterbe
I n	A lyukszalagon soron következő 5 bitet beolvassa az n címre
O n	Az n rekeszben lévő szó első 5 bitjének kiperforálása lyukszalagra
E n	Ha Akk. $\geq 0$ akkor ugrás az n címre
F n	Feltétlen ugrás az n címre
J n	Ha "B" regiszter tartalma nem nulla, akkor ugrás az n címre
K n	Egy "B b" utasítást (pl. B n) helyez el az n címre (lásd*)

\*K n: meg lehetett változtatni B n utasítás címét az indexregiszterben, továbbá  $b=(B)$ , azaz b felvette az indexregiszter tartalmát.

### c) Az utasítások formátuma:

Utasítás:= Műveleti kód Cím, Terminátor (vezérlő hatású karakter; *Flag*). Az utasításokban szereplő címeket és a műveleti kódot betűkkel lehetett jelölni (mindezt *mnemonikus* kódnak hívták). Pl. a 001 0011 gépikód helyett elég az A 3, vagy A b+2 kódot beírni.

d) 1949-ben az utasításokat a *mnemonikus* kódból egy úgynevezett beindító rutinnal (*Initial Input Routine*) hajtották végre, ez volt tehát a világ első fordítóprogramja.

#### Funkciói:

1. Műveleti kód helyettesítése megfelelő bitkombinációval;
2. A rekeszcímek decimálisból binárisba történő fordítása;
3. *Terminátor* értelmezése; relatív címből abszolút címet készített;
4. A binárisan kódolt utasításokat el is tárolta a memóriában;
5. Az ún. *vezérlőkombinációk* értelmezése és végrehajtása (Kizárólag a fordítóprogramra voltak hatással, nem pedig a lefordított programra; pl. G k – re-

latív cím alapja T g k – a betöltési címet g-re változtatta, E g k P F – az éppen beolvasott szalagon befejeződik az olvasás, az Akkumulátor nullázódik, majd a g címen kezdi el végrehajtani a programot).

**f) Utasításszó (a gép belsejében):**

Műveleti kód (6 bit)	B	Címrész	F/D bit
-------------------------	---	---------	---------

**g) Adat:**

E	16 bit	E = Előjel, 1 bit; 16 bites adatrész;
---	--------	--

**h) Dupla szó:**

E	n+1	s	n
Előjel (1 bit)	magasabb helyiértékek		Alacsonyabb helyiértékek

**i) Zárt szubrutin:**

Wheeler nevéhez fűződik.



Hol legyen a *visszatérési cím* eltárolva? *Paraméterek* átadása, tárolása?

Főprogram		Szubrutin
Tárcím	Utasítás	
P	B P (P->B)	q k s
P+1	F q	s (B b)
P+2		s+1 F P+2

**2.7. TOVÁBBI „ÖSSZÁMÍTÓGÉPEK”**

**2.7.1. Anglia**

- a) A Londoni Egyetemen 1947-ben kifejlesztették az **ARC** (Automatic Relay Calculator), majd később a **SEC** (Single Electronic Computer) számítógépet.
- b) 1948-ban, Manchesterben megépítették az első „igazi” univerzális számítógépet, a „Baby”-t. Hivatalos neve: *SSEM (Small Scale Experimental Machine)*;
- c) A teljesség kedvéért itt is megemlíjtük az EDSAC gépet.
- d) 1949-ben, szintén Manchesterben építették meg a MARK-1 gépet.

A MARK-1 gépben szintaktikailag már olyan programok jelentek meg (pl. IF A<0, CI:=CI+1 típusú utasításokkal), amelyeket (erős túlzással) már majdnem

*magas szintű programozási nyelvnek* tekinthetnénk. Az első programot Tom Kilburn írta, aki később a szakma neves képviselője lett. Hamarosan Manchester vált Anglia fő számítógépgyártó központjává.

## 2.7.2. Szovjetunió

A háború alatt analóg számítógépeket gyártottak, mivel a hadiipar ezt erősen igényelte (az analóg gépek nagy részét ballisztikus rakéták vezérlésére tervezték).

1946-ban Sztálin *áltudománynak* minősíti a számítástechnikát (tilos volt a számítógépekkel foglalkozni).

### a) A MESZM gép

1947-ben, Kievből az Elektrotechnikai Intézetben Lebegyev vezetésével elindították a számítástechnikai kutatásokat, hamarosan elkészült a **MESZM** gép.

#### **A gép jellemzői:**

- Kísérleti gépnek építették (ez volt a világ legelső háromcímű gépe),
- 31 adat tárolására volt alkalmas,
- 63 utasítást tudtak tárolni a memóriában (ENIAC típusú tároló),
- a szóhossz 21 bit volt,
- 3800 db elektroncsövet tartalmazott.

**b)** Moszkvában 1948-ban alakult ki a második bázis, ahol megalapították a Mechanikai és Számítástechnikai Intézetet, melynek vezetője Lavrentyev akadémikus lett. 1950-ben itt kezdődött el a **BESZM** nevű gép építése.

#### **A BESZM gép jellemzői:**

- Ez volt az első, gyártásra tervezett, ún. „éles” konstrukció,
- 1024 szavas memóriával rendelkezett,
- háromcímű gép volt,
- 39 bites volt a szóhossz,
- a világon a legelső *műszakilag* is lebegőpontos számábrázolású gép lett,
- 4000 db elektroncsövet tartalmazott,
- eleinte késleltetett művonal memóriája volt, de 1956-ban ezt elektrosztatikus, 60-ban ferritgyűrűs memóriára cserélték,
- bemenetnek *lyukszalagot* használtak,
- 1952-ben elindult a tesztfuttatás.

1956-ban elkészült a **BESZM-2** (a Szovjet Tudományos Akadémián is működött ilyen gép).

**c)** 1952-ben, Belorussziában (Minszkben) elindult az **M** gépcsala gyártása. Ezek kétcímű gépek voltak. 1952-ben: **M-2**, 1956-ban az **M-3**, 1960-ban az **M-20**.

Az első magyarországi gép az M-3, a minszki M-3 gép tervei alapján épült.

d) 1953-ban elindult a **Sztrela** gépcsalád gyártása.

e) 1955-ben, negyedik „vonalként” elkezdődött az **Ural** gépcsalád kifejlesztése. Az Ural gépeket már sorozatban is gyártották, 1960-62-ben már Magyarországon is megjelent az Ural-1, Ural-2 gép.

#### **Megjegyzések:**

1. Bár igen jó tervezésű gépekről beszélhetünk, a technikai lehetőségek elmaradottsága miatt a szovjet gépek megbízhatatlanul működtek.
2. A háromcímes konstrukció elméletileg optimalizálta a memóriacímzést, a gyakorlatban azonban a háromcímes lehetőségek kihasználatlanul maradtak, ami így gazdaságtalan volt és megdrágította a gépeket.
3. További kívánnivalókat hagyott maga után az a tény, hogy az utasításokat kizárólag számok formájában lehetett bevinni a gépekbe.
4. Speciális, nehézkesen kezelhető input, output egységeik voltak a gépeknek.

#### **2.7.3. Svédország**

A Stokholmi Királyi Műszaki Intézetben 1953-ban megépítették a **BESK**-et (**B**inär **E**lektronisk **S**ekvens **K**alkylator), majd 1956-ban a SMIL gépet, amely 1970-ig működött.

#### **2.7.4. Dánia**

A BESK gép alapján azt továbbfejlesztve, 1957-ben létrehozták a **DASK** számítógépet.

#### **2.7.5. Franciaország**

A fejlesztési munkálatok a Blaise Pascal Intézetben folytak. A **BULL**-gyárban 1956-ban építették meg a **GAMMA 3 ET** gépet.

#### **2.7.6. Svájc**

A Technische Hochschulén (Zürich) 1956-ban megépítették az **ERMETH** nevű számítógépet.

#### **2.7.7. Németország**

A II. világháború után a göttingeni Max Planck Fizikai Intézetben és a müncheni, valamint a darmstadti műszaki főiskolán kezdődtek el a számítógépek fejlesztésével kapcsolatos munkálatok.

1. A **DERA**-gép (**D**armstädter **E**lektronische **R**echnenautomat) Darmstadt Műszaki Főiskola, 1946.
2. A **PERM**-gép (**P**rogrammgesteuerte **E**lektrische **R**echenanlage **M**ünchen) München Műszaki Főiskola, 1956.
3. A **G1**, **G2**, **G3** gépek. Göttingeni Egyetem Max Planck Intézet, 1951.
4. 1956-ban az IBM németországi gyárában a Zuse KG cég megkezdte a **Z22** gyártását.
5. **SIEMENS 2002**: 1959-ben már gyártották.

### 2.7.8. Japán

A Fujitsu Company gyáiban 1954-ben megépítették az **E.T.L Mark I.** és **Mark II.** gépeket. (1955-ben a tranzistorizált változatot is.)

### 2.7.9. Ausztria

**URR-1, LRR-1** (jelfogós univerzális számítógépek).

### 2.7.10. Csehszlovákia

**SAPO** (mágnesdobos, jelfogós gép).

### 2.7.11. India

256 szókapacitású, ferritgyűrűs gépet fejlesztettek ki.

### 2.7.12. Izrael

**Weizac** dobmemóriás gép.

### 2.7.13. Lengyelország

**EMAL, EMC-2, ODRA.**

### 2.7.14. NDK

- 1) A Drezdai Egyetemen 1956-ban üzembe helyezték a **D1** számítógépet.
- 2) **R-300** gép (ROBOTRON cég, Drezda)

### 2.7.15. Románia

**CIFA 1, 2, 3.**

## 2.8. ELINDUL A SZÁMÍTÓGÉPEK IPARI GYÁRTÁSA

Az ENIAC, EDVAC, EDSAC, IAS gépek tapasztalatai alapján az 50-es évek elején elindult a számítógépek ipari előállítás.

### 2.8.1. A UNIVAC – gép

Mauchly és Eckert miután kiváltak a Moore School-ból, Eckert-Mauchly Computer Corporation néven céget alapítottak, amely később a Remington Rand Corporation cégbe olvadt bele. Itt 1951. márciusában hozzákezdtek a **UNIVAC (Universal Automatic Computer)** gyártásához. Az első gépet az USA Népszámlálási Hivatala rendelte meg. 1951-54 között 19 példányt értékesítettek. (Egy rendszer ára mintegy 1 Millió dollár volt.)

#### **A gép főbb jellemzői:**

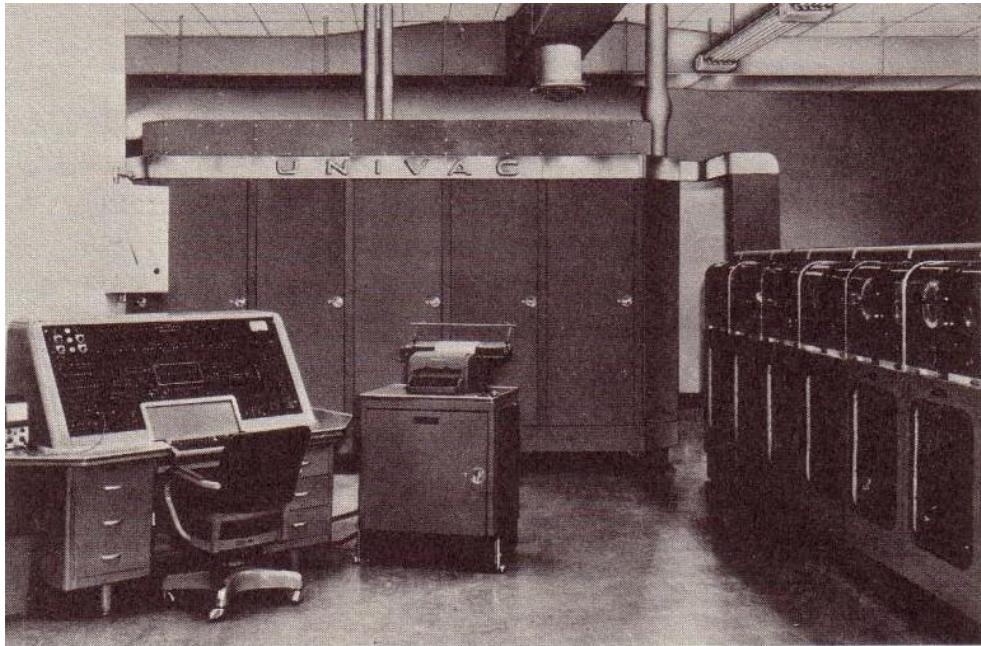
**Szóhossz:** 45 bit (11 decimális számjegy + előjel, ill. 2 utasítás)

**Sebesség:** 465 szorzás/sec.

**Memória:** késleltetett művonal 1000 szó kapacitással + mágnesszalag háttértár (fémszalaggal) 2 Millió karakter tárolására (4 bites karakterrel).

**Központi egység:** 4 általános célú akkumulátor.

**Elektroncsövek száma:** 5000.



2.7. ábra: A UNIVAC I.

### 2.8.2. Az IBM 701

Az **IBM** cég (**I**nternational **B**usiness **M**achines) 1952 májusában jelentette be a UNIVAC -kal ekvivalens tárolt programú számítógép építését. Ugyan a gép nem volt egyszerű másolata a Neumann által kifejlesztett IAS gépnek, de elég pontosan követték az abban megvalósult elveket (Neumann Jánost tanácsadóként alkalmazták). A számítógépet az IBM „elektronikus adatfeldolgozó gép”-nek nevezte (kerülte a computer elnevezést), mert ki akarta emelni a kereskedelmi alkalmazást.

#### **A gép jellemzői:**

*Központi memória:* 3 inch átmérőjű elektroncsövek (Williamus csövek). Egy cső 1024 bitet tárolt, így egy 72 csőből álló memória 2048 36 bites szó tárolására volt alkalmas. Egy szót egyszerre lehetett kiolvasni a memóriából (nem bitenként, mint az UNIVAC művonalas tárolójából).

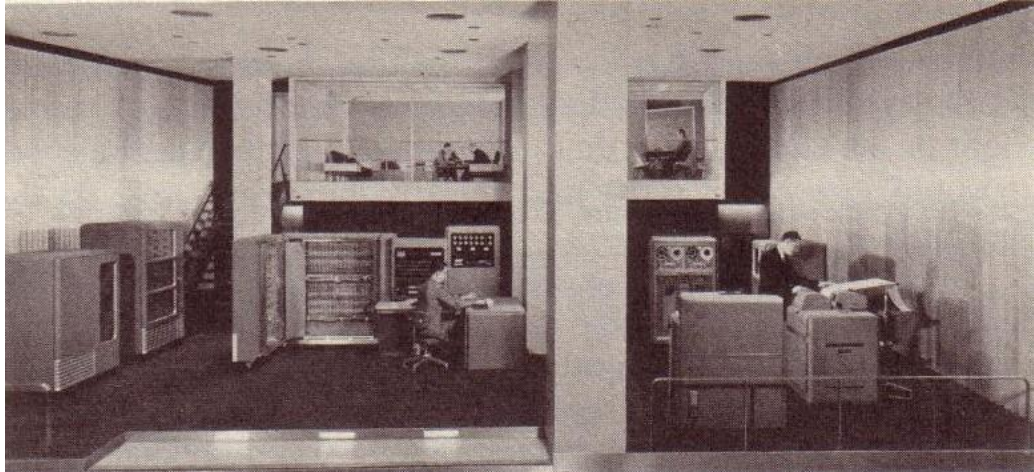
Plasztik szalagos háttérmemóriát építettek a géphez, és a közbülső memóriaként a kettő között egy dobmémória működött.

*Műveleti sebesség:* 2000 szorzás/sec.

1952-ban az IBM bejelentette egy új gép – a 702-es modell – tervét; 1955-ben készült el az első példány. Ebben a gépben már ferritgyűrűs memóriát alkalmaztak.

A 700-as sorozat további tagjai 704, 709. Első generációs gépnek tekinthető az IBM 650-es gép is.





2.8. ábra: Az IBM 701

### 2.8.3. Más gyártók

A fenti „nagyokon” kívül az ötvenes években sok más cég is gyártott első generációs számítógépeket. Ismertebbek:

- RCA Bizmac,
- Burroughs: VDEC, 205,
- Datamatic 1000, LGP-30, G-15,
- ERA 1103,
- CRC-102.

### 2.9. Az első generációs számítógépek fontosabb jellemzői

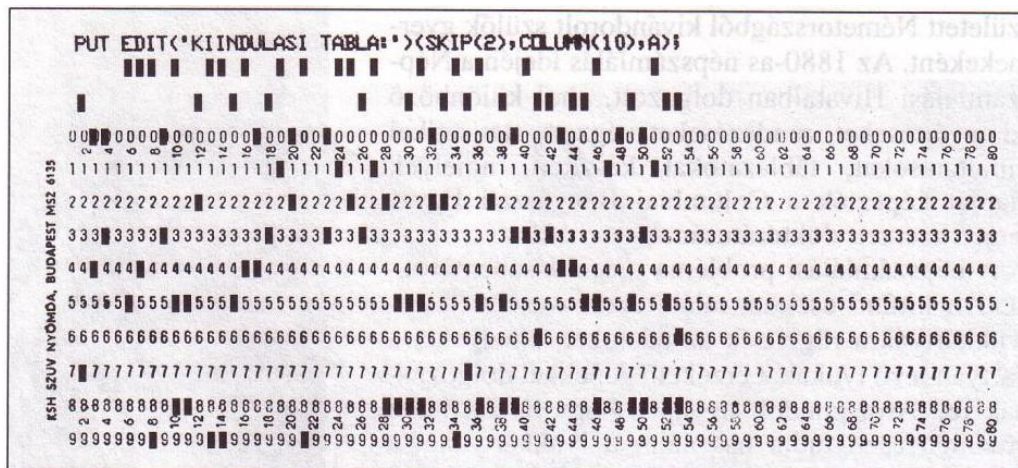
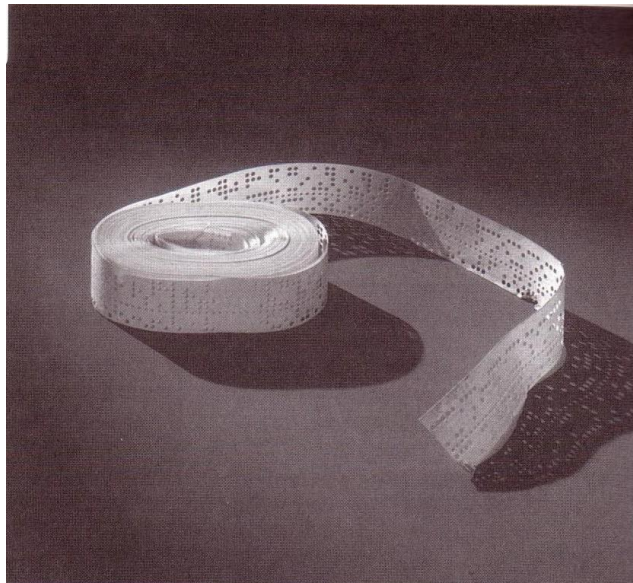
Zárjuk le az első generációs gépek korszakát egy összefoglalással, a fontosabb jellemzők felsorolásával. Ezek:

- elektroncső,
- tárolt program,
- kettes számrendszer,
- mágnesszalagos tárolók,
- mágnesdob memória,



2.9. ábra: Mágnesdob memória

- nagy energiafogyasztás,
- a műveleti sebesség 1000 összeadás/sec alatt,
- nagy méret, -súly (30 tonna),
- papír adathordozójú perifériák,
- gyakori hibák (óránként 1 hiba),
- főleg gépi kódú programozás,
- megjelentek (kísérleti jelleggel) az első magas szintű nyelvek.



2.9. ábra: A lyukszalag és a lyukkártya

## 3 . F E J E Z E T

### A SZÁMÍTÓGÉPEK 2.-5. GENERÁCIÓJA

#### 3.1. A SZÁMÍTÓGÉPEK MÁSODIK GENERÁCIÓJA (1959-1965)

A számítógépek második generációját a *tranzisztor alkalmazása* indította el. (A tranzisztort William Shockley 1947-ben a Bell Laboratóriumban fedezte fel, ezért Nobel díjat kapott.)

##### 3.1.1. A TX-O gép

Az első tranzisztoros áramkörből épült számítógép a TX-O volt, amit a Massachusetts Egyetemen építettek meg 1956-ban. Memóriája 64K (18 bites szavas ferritgyűrűs).



3.1. ábra: A TXO gép

##### 3.1.2. A PDP-1

1959-ben a Digital Equipment Corporation (DEC) a TX-O számos áramköri és architektúrális paraméterét alapul véve megépítette a PDP-1 (Programmed Data Processor) gépet. (Belső ferritgyűrűs memória, 4K szó /18 bit/ 64K-ig bővíthető, 100 000 összeadás/sec.)



3.2. ábra: A PDP-1 gép

### 3.1.3. Az IBM 7090 gép

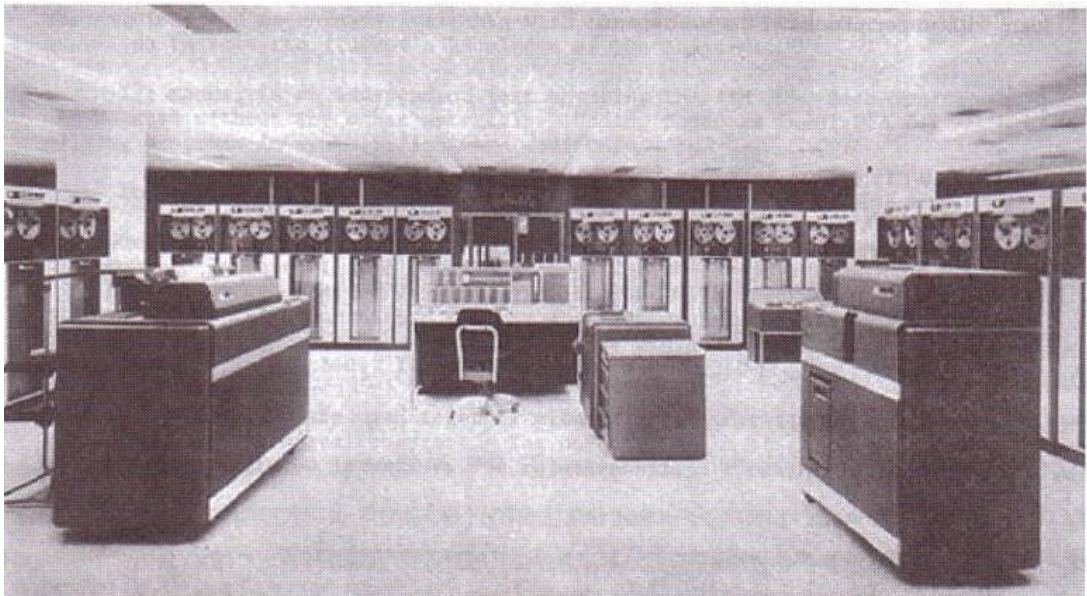
Az IBM is ráállt a tranzisztorizálásra, létrehozták a 7090-es gépet, de amíg a PDP-1 teljesen szakított az elektroncsöves filozófiával, addig az IBM 7090-es voltaképpen a 709-es tranzisztorizált változata volt.

Az IBM 7090-es már nagyon gyors gép volt: 229 000 összeadás/sec. (50 000 tranzisztort tartalmazott.)

Nem csak az IBM és a DEC állt rá a tranzisztoros gépek gyártására, hanem sok más – már akkor is – „nagyak” számítógépesítő cég is.

*A legismertebb második generációs számítógépek:*

- IBM: 7090, 7094, 1620, 1401

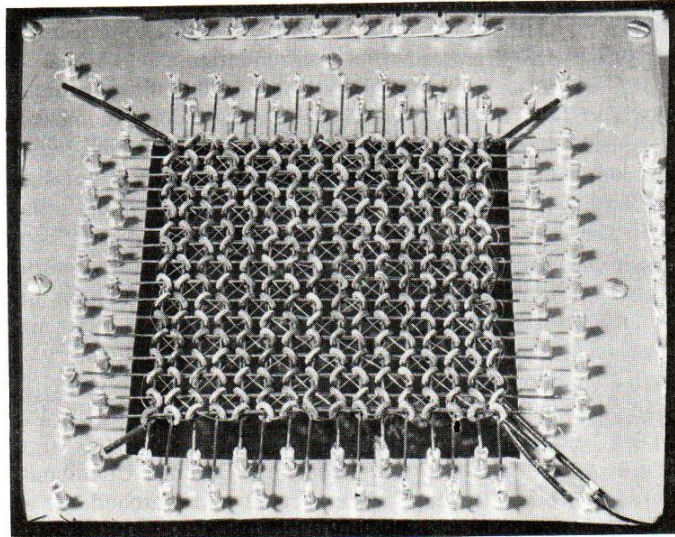


3.3. ábra: az IBM 7090

- RCA 501, 3301
- CDC (Control Data Corporation): CDC 3600, 1604, 7600
- Honeywell: 800
- NEC-1101, 1102
- LARC, Stretch.

*A második generációs számítógépek főbb jellemzői:*

- tranzisztorokból épített (félvezetős) áramkörök,
- a méretek lényegesen csökkentek,
- az áramfogyasztás csökken,
- jóval kisebb volt a hűtési igény,
- ferritmagos memória,



3.4. ábra: Ferritmagos memória

- mágnesszalagos és mágneslemezes külső tárolók,
- a megbízhatóság jelentősen nőtt (4-5 óránként 1 hiba),
- megjelentek a fotocellás input berendezések,
- a műveleti idő millisecundumról mikrosecundumra csökkent (10 000 összszeadás/sec.),
- a gépek ára is meredeken esett (a PDP-1 ára 100 000 dollár),
- megjelentek az operációs rendszerek első változatai (kötegelt feldolgozás),
- megjelentek (kifejlődtek) és elterjedtek a magasszintű nyelvek (FORTRAN, ALGOL, COBOL),
- fontosabb, számítógépeket gyártó cégek: IBM, UNIVAC, Honeywell, Digital Equipruents (DEC), Borroughs, Control Data Corporation (CDC), ICL (Anglia).

### 3.2. A HARMADIK GENERÁCIÓS SZÁMÍTÓGÉPEK (1965-1975)

A számítógépek harmadik generációját az integrált áramkörök (IC-k Integrated Circuit) használata indította. Elindul a „chip”-korszak.

A germánium alapú IC-t Jack Kilby (1958, Texas Instrument) a szilícium alapút pedig Robert Noyce (Fairchild Semiconductor, 1959) fejlesztette ki.

Az IC-eket 1963-ban már tömegesen gyártották; az első IC-eket alkalmazó számítógépek 1964-ben kerültek kereskedelmi forgalomba.

Ez a korszak a közepes intergráltságú (MSI, Medium Scall Intergrated) IC-k korszaka.

Az Intel-cég 1969-ben piacra doja az 1 kilobites RAM chip-et.

A gépek ára jelentősen csökkent, a kereslet nagyon megnőtt (1970 körül több mint 100 000 „nagy gépet” és közel 100 000 „mini számítógépet” üzemeltettek a világon).

### **A korszak számítógépeinek fontosabb jellemzői:**

- jelentős méret- és áramfogyasztás csökkenés,
- megnövekedett megbízhatóság,
- a műveleti sebesség 500 000 összeadás/sec,
- félvezetős memória,
- közvetlen hozzáférésű mágneslemezes háttértárak (5-25 Mbyte kapacitás),
- byte-szervezésű memória,
- input/output processzor,
- több tevékenység párhuzamos végzése,
- grafikus adatok kezelése (digitalizálók, rajzológépek, színes grafikus monitorok),
- a mikroprogramozás megjelenése, elterjedése,
- kifejlesztik a termék jellegű operációs rendszereket, megjelenik a multiprogramozás, az időosztásos (time sharing) rendszerek, amelyek segítségével dialógus rendszerű programfejlesztésre nyílik lehetőség,
- szélesedik a magasszintű programozási nyelvek köre (PL/1, BASIC Pascal stb.); megjelenik az adatbáziskezelés koncepciója. Kialakulnak az alkalmazói szoftverek gyártására a „szoftverházak”, (szoftver gyárak) az ipari szoftvergyártás,
- „elindul” a strukturált programozás filozófiája.

#### **3.2.1. Az IBM – 360**

A harmadik generációs korszakot az IBM S-360-as számítógép családdal (egymásra épülő hat gép) is összeköthetjük. A felhasználók a kisebb teljesítményű géppel kezdhették és az igények szerint változhattak nagyobb, az előbbivel teljesen kompatibilis számítógépekre.

Az IBM – 360-as család 5 milliárd dolláros fejlesztés, beruházás eredménye volt (60 000 új alkalmazott!), 300 szabadalomról született.

#### **3.2.2. A DEC PDP sorozat**

A DEC cég ún. miniszámítógép sorozatra állt rá. Ez a sorozat a második generációs PDP-1-gyel indult, és az 1972-ben gyártott PDP-16-tal végződött. Az 1965-ben készült PDP-8-ből 50 000 db-ot adtak el. A sorozathoz elkészítették a UNIX operációs rendszert.

#### **3.2.3. A jelentősebb harmadik generációs gépek:**

- IBM-360
- DEC: PDP-8, PDP-11,
- CDC: 6000, 7000,
- UNIVAC: 1108, 1110,
- RCA Spectra 70-es család,
- Honeywell H-316, DDP-516,
- ICL (Anglia) gépek.

### 3.2.4. Az ESzR gépek

A harmadik generációs gépek korszakára esik az akkori szocialista országok (KGST) által indított nagy számítástechnikai projekt indítása. Ennek a fő célja, az IBM-360-as gépcsaládhoz hasonló számítógépcsalád kiépítése (**ESzR** = **Egységes Számítástechnikai Rendszer**, oroszul ESz = Egyinnaja Szisztema). A gépcsalád az ESzR/10 – ESzR/60 családból állt. A család egyes tagjait az egyes országok gyártották. Magyarországnak az ESzR/10-es gép „jutott”. Ezekből a gépekből sok példányt gyártottak, a szocialista országokban a nyugati gépek mellett (amelyekre elég erős embargó /NATO korlátozás/ volt érvényben, ezekkel látták el a vállalatokat, bérszámítóközpontokat).

### 3.3. A NEGYEDIK GENERÁCIÓ KORSZAKA

Ezt a korszakot (ami a 70-es évek második felétől kezdődik) a mikroprocesszorok elterjedése, a magasan integrált **VLSI**-vel jelölt (**V**ery **L**arge **S**cale **I**ntegration) elemek megjelenése jellemezte. Szokás ennek a korszaknak az elejét az IBM-370-es számítógép család megjelenésével is összekötni.

Ennek a korszaknak minden eseményét, termékeit felsorolni ezen írás keretében lehetetlen, ezért csak néhány, nagyon fontos jellemzőt, „kilométerkövet” említünk meg. Voltaképpen ennek a korszaknak az eredményeit napjainkban is látjuk, „kis korszakokon” keresztül jutottunk el a mához.

- 10-100 millió művelet/sec,
- szuperszámítógépek (Cray 1, CYBER 205),
- mikroszámítógépek (az első mikrogép az Altair és az Apple),
- személyi számítógépek /PC-k/ (az elsők: Commodore, Altair, IBM, Apple),
- a mikroprocesszorok óriási fejlődése,
- a nagy gépekhez kifejlesztik az ún. munkaállomásokat,
- multiprocesszoros rendszerek, új architektúrák,
- Harvard architektúra (külön tároló az adat és a program számára),
- vektorprocesszor (a vektor elemeinek átlapolása),
- tömbprocesszor (ugyanazon utasítás különböző adatokon),
- szisztolikus processzor (egyforma feldolgozó egységek speciális elrendezése),
- asszociatív processzor (a memória tartalom szerint címezhető),
- data-flow architektúra (utasítás szerinti párhuzamosítás),
- osztott hálózati rendszerek,
- floppy lemezek,
- nyomtatók (mátrix, tintasugaras, lézer),
- merevlemez egységek (Winchester) több Gbyte-os háttértárolók,
- optikai lemezek (CD),
- grafikus perifériák, nagyfelbontású színes monitorok, scannerek, fényce-ruza, egér stb.,
- programgenerátorok,
- CASE (Computer Aided System Engineering) a programfejlesztés elősegítésére,

- számos új („problémaorientált”) nyelv, ill. szoftver (adatbáziskezelők, táblázatkezelők, szövegszerkesztők, folyamatvezérlők stb.), párhuzamos programozás,
- jelentős méretcsökkenés, nagy tárolási sűrűség, csökkenő ár, növekvő megbízhatóság,
- „nagy integráltságú” operációs rendszerek (DOS, UNIX, LINUX stb.),
- létrejönnek a nagy mikroprocesszor gyártó cégek: az Intel, a Motorola, a Texas Instruments stb.,
- létrejön a Microsoft,
- elindul hódító útjára az Internet.

### **3.4. Ötödik generációs számítógépek**

A számítástechnika területén olyan nagy – és olyan folytonos – a fejlődés, hogy nem lehet tudni, hogy most melyik „generációban” vagyunk.

A 90-es évek végéig kialakult technológiák azóta is folytonosan fejlődnek. A 90-es éveket egyfelől a PC-k térhódítása, a hálózatok világméretűvé válása, a nanotechnológiák kifejlesztése (egy mikron = 1000 nanométer; nano = egymilliárdod rész). Az IC gyártásánál „atomi komponenseket” használnak (molekuláris nanotechnológia, nanocsövek).

Létrejön a biochipeket alkalmazó géntechnológia; megjelenik a buborékmemória, holografikus tárolók.



## 4 . F E J E Z E T

### A SZÁMÍTÁSTECHNIKA MAGYARORSZÁGON

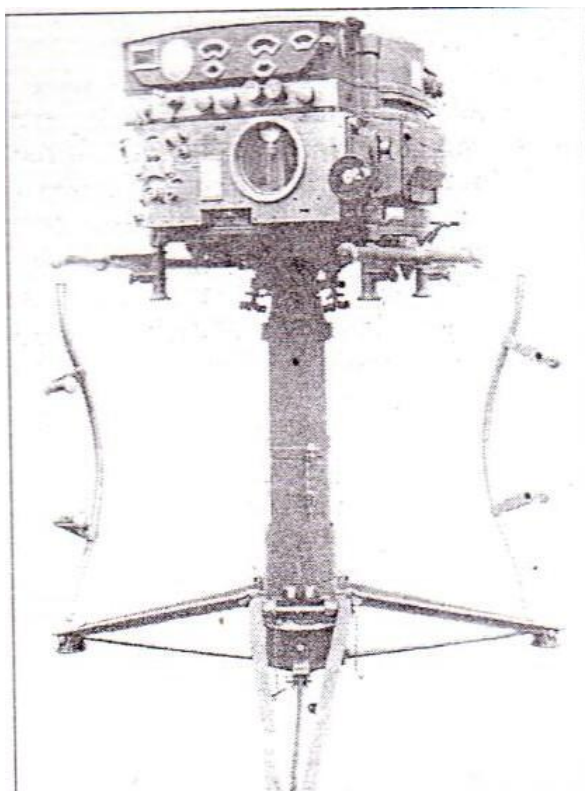
#### 4.1. JUHÁSZ-FÉLE LŐELEMKÉPZŐ

Nem tekinthető szorosan a számítástechnikához kötődőnek, mégis megemlítjük, mint „hungarikum”-ot.

Ballisztikus pályák számításához készült, a GAMMA művekben dolgozták ki.

Analóg, lövegek irányítására használt szerkezet volt.

Juhász ezt a gépet 1920-ban találta fel. Lényegében egy torzított hengerre vitték fel az adatokat, amely ballisztikus pályaként működött. Kb. 80-100 ezer adattal tudott dolgozni, amit egy „paránymérő” nevű eszközzel tapogattak le. Ez a gép voltaképpen *analóg célgépnek* tekintendő.



4.1. ábra: A löelemképző

#### 4.2. A HAZAI SZÁMÍTÁSTECHNIKÁT (INFORMATIKÁT) MEGALAPOZÓ SZAKIRODALMAK

- a) Nemes Tihamér 1947-ben, a Rádiótechnika című újságban viszonylag részletesen ismertette az ENIAC gépet.
- b) Kunfalvi Dezső „Nagyméretű számológépek” címmel 1948-ban, a Természet-tudományi Közlönyben publikált cikket.

- c) Gábor László 1949-ben az Élet és Tudományban „Gondolkodó gépek” címmel írt cikket.
- d) Tarján Rezső „Elektronikus számológépek” címmel a Magyar Tudományos Akadémián tartott előadást 1950-ben.
- e) Frey Tamás „Matematikai gépek” címmel írt 1953-ban cikket a Magyar Technika című folyóiratban.
- f) Rényi Alfréd 1955-56-ban a Szabad Nép című újságban cikksorozatot írt az informatikával kapcsolatban, ami nagy jelentőséggel bírt (akkoriban az informatika áltudománynak számított).

### 4.3. ADATFELDOLGOZÁS

- a) A Központi Statisztikai Hivatal 1930-tól nagy tömegben működtetett lyukkártyás gépeket, 1940-ben pedig már ezen végezték a népszámlálás feldolgozását.
- b) 1953-ban létrehozták a KSH Ügyvitel Gépesítési Felügyeletet, mely a későbbiekben „országos hatósággá” nőtte ki magát. Az ügyvitel-gépesítési oktatás is itt indult el, bölcsője volt a SZÁMOK-nak, amely intézmény jelentős szerepet töltött be a hazai számítástechnikai oktatásban.
- c) A lyukkártyás gépek korszakában, Magyarországon is megjelentek az IBM kirendeltségei és gépei az alábbi helyeken:
  - DIMÁVAG,
  - Elektromos Művek,
  - Magyar Nemzeti Bank,
  - Honvédelmi Minisztérium,
  - Kohó és Gépipari Minisztérium,
  - Gépi Adatfeldolgozó Vállalat (1950-ben hozták létre).

### 4.4. Fejlesztések

- a) 1945-46 között megépítették a „DANADDO” nevű szerkezetet, mely egy fogazott lécekből álló, szivar alakú, kisméretű számológép volt. Érdekessége, hogy – akkoriban rendhagyó módon – negatív számokkal is tudott már számolni. Mérete 42×155×33 milliméter volt. E gép működésének elvi leírását egyébként 1921-ben Zeninváry Lajos adta meg.

#### b) „**Különös kezdeményezés**”

A Gyűjtőfogházban egy javaslatot dolgoztak ki a Magyar Tudományos Akadémiának analóg és digitális számítógépek gyártására 1953-ban (itt több mérnököt tartottak fogva politikai okokból). Legjelentősebb személyek:

- Edényi László,
- Hatvany József,
- Kozma László,
- Tarján Rezső.

- c) 1955-ben megalakult a Magyar Tudományos Akadémia Méréstechnikai és Műszerügyi Intézete.
- Tarján Rezső vezetésével egy osztály a számítógép-memóriákkal és a kibernetikával (akkoriban a kibernetika szóval jelölték a számítástechnikát /informatikát/) foglalkozott, hamarosan kidolgozták a *magnetostriktív*, nikkell, művonalas memóriát. Jelentős szerepet kapott a *ferritmemória* kutatása és a *mágnesdobos* memória fejlesztése is.
  - 1956-ban innen szakadt ki egy csoport, melyből megalakult az MTA Kibernetikai Kutatócsoportja. Ez az intézet lett a hazai számítástechnika bölcsője. (Erről a 4.5. pontban külön szólnunk.)

d) **GYUBER KTSZ (Gyűjtőberendezések Kisipari Termelőszövetkezete)**

Elindult az analóg és digitális számítógépek kutatása és fejlesztése (nyomatékmérleg, himbás olajkutak vezérlésére alkalmas gépek, löelemképzők, továbbá „kisszámítógép”, mely csapágyazott tengelyek fordulatszámát számította ki).

e) **Nemes Tihamér (1845-1960)**

Logikai gépeket készített – általában relés megoldással – melyek logikai kifejezések kiértékelésére voltak alkalmasak.  
Input/output és állapotmodellek megvalósítására több szabadalmat is beadott.

f) **Szeged**

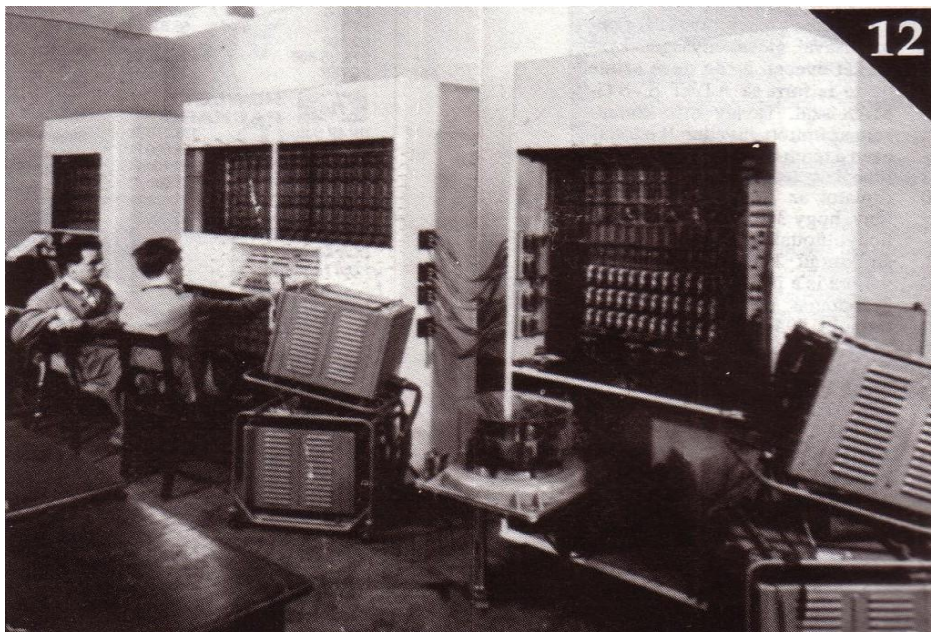
- *Kalmár László* 1956-57-ben megépítette a Szegedi Logikai Gépet (relés gép), mely nyolcváltozós, tetszőleges logikai kifejezést ki tudott értékelni. 1957-ben egy ún. „*Marker-gépet*” készített, amely a konstans értékek mellett már változókat is kezelt.
- Katicabogár-modell (Muszka Dániellel együtt készítette): relék helyett elektroncsövekkel működött, fény- és hangjelet egyaránt képes volt érzékelni.
- 1960-ban megindult a „*programozó matematikusok*” oktatása.

#### 4.5. MTA KIBERNETIKAI KUTATÓCSOPORT (MTA KKCs)

- Ez az intézet volt a hazai számítástechnika bölcsője.
- 1956-ban jött létre, igazgatója Varga Sándor mérnök. Igazából csak 1957 tavaszán kezdett működni (Februárban újjáalakultak, majd átköltöztek a Nádor utca 7-be, ahol később az M-3 számítógépet építették).
- 1957 augusztusának elején alakult meg az a gárda, amelyik nekilátott az M-3 gép építésének. Az M-3 gép szovjet dokumentációk alapján készült, a minszki M „vonal” tagja volt.
- A gép 1960-tól kezdve üzemszerűen működött, de 1966-ban leállították, majd Szegedre került. A gép minden egyes alkatrészét Magyarországon gyártották, néhány önálló fejlesztést is tartalmazott, ilyen volt pl. a mágnesdob.
- 1958-ban az intézmény 70 fővel működött.

#### 4.5.1 Az M-3 gép jellemzői

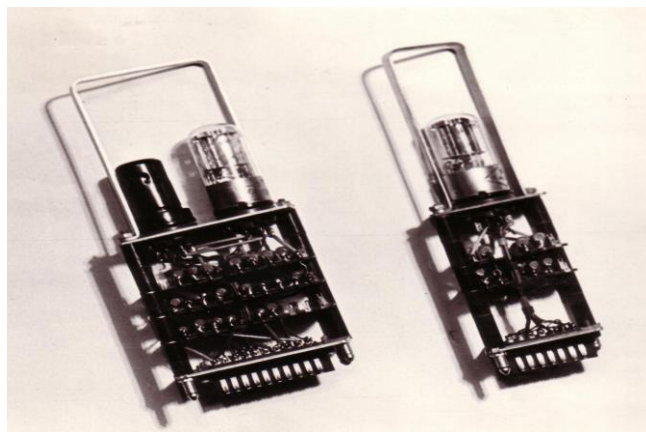
- a) Elektroncsöves gép volt (összesen 2500 elektroncsővel), fogyasztása 10 kW volt, magas hőmérsékleten működött, emiatt nehéz volt hűteni.
- b) Input: lyukszalag, output: *Teletype* (olyasmi, mint az írógép; automatikusan írta ki a szöveget). Bináris gép volt, de nyolcas számrendszerben lehetett a programokat és az adatokat írni.
- c) Memóriája először dobmémória volt (1024 szót tudott tárolni), később (1961-ben) ferritmémória készült a géphez, majd (1962 körül) mágnesszalag-memóriát is építettek hozzá, ez 80 000 szót tudott tárolni, 1 szót 0,7 msec alatt tudott kiolvasni. A gép sebessége: átlagosan 30 művelet / sec, de amikor már ferritmemóriával működött, ez 1000 művelet / sec-ra emelkedett.



4.2. ábra: Az M-3 számítógép

#### d) **Regiszterek:**

- Aritmetikai egységben összesen 4 db.
- Utasításszámláló regiszter.
- Szelekciós regiszter:
  - Kétcímű, fixpontos gép (csak abszolútértékben egynél kisebb számokkal tudott számolni).
  - Szóhossz: 31 bit, ahol 1 bit az előjel,  $2^{-1}$ -től  $2^{-30}$ -ig pedig maga a szám.
  - Utasítás: 1 bit előjel, 6 bit műveleti kód, 12-12 bit pedig címek.
  - Alakja: m a b (pl.: 00 0316 0517). A gép mínusz nulla értéket is létrehozott, ami jól jött különféle programozási trükkök alkalmazásánál;
- A „B” regiszter volt az indexregiszter.



4.2. ábra: Az M-3 áramkörei



4.4. ábra: Telex-gép (nyomtat)

e) **Utasítások:**

– **Összeadás:**

A művelet szimbolikus jele	Az utasítás alakjai	Magyarázat
+	00 a b	az <b>a</b> cím tartalma hozzáadódik a <b>b</b> cím tartalmához, és ez az érték bemegy <b>b</b> címre (a továbbiakban ezt így jelöljük: $b:=a'+b'$ )
+,	10 a b	$B:=a'+b'$
↓+	20 a b	$b:=B'+a'$
↓+,	30 a b	$B:=B'+a'$

- *Kivonás* (ugyanaz, mint összeadásnál, azzal a különbséggel, hogy a műveleti kód második jegye 1).
- *Szorzás* (ugyanúgy, mint az eddigiek, de 2 a második jegy).
- *Osztás* (az eddigiekkel analóg módon; a műveleti kód 3).
- *Nyomtatás* (a műveleti kód 40, az eredmény tízes számrendszerben kerül kinyomtatásra).
- *Input 07* ab. A lyukszalagról 1 szó (31 bit) bekerül a b című rekeszbe.
- *Abszolút érték képzése*: a műveleti kód 50.
- *Logikai szorzás*:

$$\Lambda \quad 06 \ a \ b \quad B:=a'\wedge b'$$

Ennek az utasításnak is volt 16, 26, 36 kódú alakja.

- *Ciklus*:

Tipus	Műveleti kód	Magyarázat
+	57 a b	a'+b', és ezt hajtja végre, mint utasítást
↓+	77 a b	a'+B'

(Pl.: a'=00 0512 0316; b'=00 0000 0001; a'+b'=00 0512 0317, b értékét pedig a továbbiakban meg lehetett változtatni)

- *Ugróutasítás* (u, vezérlésátadás); 24 a b. A gép átadja a vezérlést **a**-nak, majd **B** tartalmát beírja **b**-be).
- *Feltételes ugrás* (FU, 34' a b, és a vezérlést **a** kapja meg, ha B'<0, egyébként **b**).
- *Átvitel* (05 a b; **a** tartalma átmegy **b**-be).

#### f) **Példa egy algoritmusra:**

Az  $a$  szám  $\sqrt{a}$  gyökét az alábbi „iterációs” képlettel számítjuk ki

$$\sqrt{a} \approx X_n = \frac{1}{2} \left( \frac{a}{X_{n-1}} + X_{n-1} \right);$$

$X_n$ -t, az új közelítést az előző közelítésből kapjuk. Az ismétlést (iterációt) addig folytatjuk, amíg két szomszédos közelítés egy előre adott számnál abszolútértékben kisebb nem lesz.

Konstansok:

0010:=0.5;

0011:=00..01 ( $2^{-30}$ );

Munkarekeszek:

0020:= $X_0$  ( $\approx 1$ );

0021:= $a$ ;

0022 :=  $\sqrt{a}$ ;  
 0023:=...;  
 0100 A 05 0020 0022 ( $X_0 \Rightarrow 0022$ );  
 0101 A 05 0021 0023 ( $a \Rightarrow 0023$ );  
 0102 : 02 0022 0023  $\left( \frac{a}{X_{n-1}} [\Leftrightarrow X_0] \right)$ ;  
 0103 + 00 0022 0023  $\left( \frac{a}{X_{n-1}} + X_{n-1} \right)$ ;  
 0104  $\times$  03 0010 0023  $\left( \frac{1}{2} \left[ \frac{a}{X_{n-1}} + X_{n-1} \right] \right)$ ;  
 0105 – 01 0023 0022 ( $X_n - X_{n-1}$ );  
 0106 || 55 0000 0022  $\left( X_n - X_{n-1} \right)$ ;  
 0107 – 01 0011 0022;  
 0110 U 34 0113 0111;  
 0111 A 05 0023 0022  $\left( X_{n+1} \Rightarrow 0022 \right)$ ;  
 0112 U 34 0101 0101;  
 0113 34 0504 0504 (végeredményt nyomtató szubrutin);

#### 4.5.3. Az első „éles” alkalmazások

Az alább felsorolt alkalmazásoknak nagy szerepük volt, mert a kételkedőknek (akik sokan voltak) bizonyítani lehetett, hogy a számítógép nem játék, hanem nagyon hasznos eszköz.

- Frey Tamás – Szelezsán János: Az Erzsébet híd merevítő tartóival kapcsolatos számítások (30×30-as nemlineáris egyenletrendszer).
- Krekó Béla – Dömölki Bálint: Szállítási feladat (szállítási költségek minimalizálása;  $Ax \leq b; \min(cx); x = ?$ ; TEFU-nak [másfél óra alatt; 30% megtakarítás] és a Vasúti Tudományos Kutatóintézetnek [2 óra alatt 5% megtakarítás kiszámítása]).
- Buzgó József: Keretszerkezetek számítása (épületeknél; 50 elem, 23 sarokpont vizsgálata).
- Ganczer Sándor – Veidinger László: Sakktábla-szerű termékmérleg (lineáris programozás; az Országos Tervhivatalnak; 40×40-es mátrix inverzének kiszámítása 25 óra alatt).
- Lőcs Gyula: Villamos-hálózattal kapcsolatos feladat (A Villamos Energiaipari Kutatóintézet számára analóg-digitális rendszer kiépítése; kb. 10-15 óra).
- Balatoni János: Optikai rendszerek tervezése (A Gamma Optikai Művek számára; 50-200 óra igénybevétel).
- Rózsa Pál – Veidinger László: Részecskék emulzióban való viselkedésének vizsgálata (KFKI részére; 20×20-as mátrixok invertálása, ami kézzel 8-10 hétig tartott volna, géppel pedig 1,5 órát vett igénybe).

- Szelezsán János: Metán parciális oxidációjához (Magyar Ásványolaj Társaság és a Veszprémi Egyetem megrendelésére; nemlineáris egyenletrendszer, aminek kiszámítása 500-szor gyorsabb volt géppel, mint kézzel).
- Révész Pálné: Regressziószámítás.
- Gergely József: Ipari bordás hőcserélők (ELTE-TTK; 21 db több paramétert tartalmazó képlet kiszámítása; 400-szor volt gyorsabb géppel, mint kézzel).
- Gergely József: Elektronok becsapódása (KFKI megrendelésére; 5000 adat kiértékelése; géppel 200-szor volt gyorsabb, mint kézzel).
- Sándor Ferenc: Tekercselési számítások ([villamosság]; dinamók, transzformátorok; a Klement Gottwald gyár megrendelésére; a kézinél 50-szer gyorsabb).

1961-ben a KKCS neve MTA Számítástechnikai Központ lett. Az M-3 gép építése közben hozzáláttak egy „Budapest” nevű, teljesen magyar gép tervezésének, de ez sikertelen kísérlet volt.

#### 4.6. TELEFONGYÁR

1959-ben elkezdtek gyártani egy EDLA nevű gépet. Kifejlesztettek hozzá egy mágneslemez-memóriát (ami a mai diszkek őséneke tekinthető!), majd később fóliamemóriát is terveztek hozzá, ami a mai floppy-nak felel meg. A gép végül nem készült el.

1963-ban hozzáláttak az EDLA-II gép tervezéséhez, azonban ez a gép sem készült el.

#### 4.7. VILLENKI (VILLAMOS-ENERGIAI KUTATÓINTÉZET)

- a) Analóg számítógépet építettek az országos áramellátás működtetésére (1953-ban ez már egy analóg géppel irányított hálózattá vált).
- b) 1960-ban egy logikai kapcsoló (digitális jellegű) áramkört készítettek. Erre alapozva elkészítették a FÉTIS (Félvezetős Telemechanikai és Irányító Szisztéma) nevű rendszert, mely már célgépnek tekinthető. Ez a gép már *tranzisztorokat* tartalmazott, és 1 kByte memóriával rendelkezett.

#### 4.8. KFKI (KÖZPONTI FIZIKAI KUTATÓ INTÉZET)

- a) 1956-ban már elindultak számítástechnikai jellegű kutatások, amelyet a részecskefizikából felmerülő igény erősen befolyásolt. Első eredményük egy ún. *Sokcsatornás Analizátor* volt, amely a részecskék becsapódását vizsgálta, majd a mérési eredményekkel végzett számítási műveleteket (analóg és digitális módon).
- b) 1958-ban kifejlesztettek egy *fáziseltolódásos memóriát* (speciális tárolóegység), mely sokban hasonlít a *késleltetett művonalhoz*, de higanly helyett fáziseltolással működik.
- c) 1959-ben készítettek egy 256×16 bites *ferritmemóriás analizátort*.



- d) 1963-ban elkészült egy 256 csatornás analizátor, melynek már nevet is adtak: Nukleáris Tárolt-programozású Analizátor (NTA). Ezt a gépet már sorozatban is gyártották az EMG-ben (Elektromos Mérőkészülékek Gyára).
- e) 1965-ben ebből a gépből alakult ki a TPA (Tárolt Programú Analizátor), mely már igazi számítógépnek tekinthető, és teljesen magyar tervezésű gép volt. Ebből a számítógépből 1989-ig összesen 1500 darabot gyártottak (ekkor már az amerikai PDP cég gépeinek minőségi szintjét is elérte). Nagyon elegáns megoldásokkal készült, és viszonylag kényelmesen lehetett használni.

#### **4.9. EMG (ELEKTROMOS MÉRŐKÉSZÜLÉKEK GYÁRA)**

Eleinte csak mérőkészülékeket gyártottak, később már áramkörkutatásokat is végeztek, majd tranzistoros áramköröket gyártottak (ez kezdett egyre fontosabb tevékenységgé válni).

- 1964-ben készítettek egy HUNOR-131 nevű, digitális pénztárgépet – tranzistoros technológiára alapozva.
- Elkészítették az EMG-830 gépet, mely már szintén igazi számítógép volt. (Megemlítendő Klatsmányi mérnök neve.) Ebből a gépből 16 darabot építettek. Itt alkalmaztak először kapcsolóelemként germánium tranzistoros diódát.
- 1970-ben az EMG-ből a Videoton-ba költözött a számítástechnikai profil.

#### **4.10. VASIPARI KUTATÓINTÉZET**

1959-ben kutatásokat végeztek a ferritgyűrűk terén (gyártottak is, melyekből az MTA KKCS is rendelt).

#### **4.11. MÉRÉSTECHNIKAI KÖZPONTI KUTATÓ LABORATÓRIUM**

„Számítástechnikai szorzóművet” készítettek, amely egy olyan áramkör volt, ahol a frekvencia hordozta az információt. Az áramkört szorzási műveletek elvégzésére készítették.

#### **4.12. Budapesti Műszaki Egyetem (BME)**

A BME-n Kozma László professzor vezetésével készült egy egyszerű számítógép. Kozma László már 1938-ban Antwerpen-ben készített egy *elektromechanikus számítógépet* (gyakorlatilag Zuse-vel egyidőben), mely decimálisan számolt; egy telefonközpontban automatikusan végezte a díjszámítási műveleteket. A gyűjtőfogházban lehetőséget kapott arra, hogy megtervezzen – és modellelzen – egy automata telefonközpontot.

Mikor 1956-ban visszakerült az egyetemre, 1956 és 58 között megépítette a MESZ-1 nevű számítógépet:

- A gép *relékkel* (jelfogókkal) működött, összesen 2000 darab relét tartalmazott; a gyökvonás művelete azonban rendhagyó módon áramkörileg volt beégetve a gépbe;

- Egycímű gép volt – nem tárolt programozású – programlapokon (lyukkártyához hasonló eszköz) helyezték el az utasításokat; 1 lapon összesen 45 utasítás fért el;
- Perifériája egy Mercedes-típusú átalakított írógép volt. Tárolója mágnes-huzalos volt, abban keringett az információ;
- 1964-ben Kozma professzor megépített egy speciális számítógépet az MTA részére, mely nyelvstatisztikai műveletek elvégzésére volt hivatott (Petőfi és Ady verseiben gyakran előforduló betűket vizsgáltak; megállapították, hogy Petőfi verseiben az „e” betű dominál).

#### 4.13. SZÁMÍTÁSTECHNIKAI KÖZPONTOK

Az előbbieken felsoroltuk azokat az intézményeket, amelyekben számítástechnikai fejlesztések folytak, számítógépeket építettek.

Az alábbiakban röviden ismertetjük azokat az intézményeket, amelyek kereskedelmi forgalomban kapható számítógépeket beszerelve a hazai számítástechnikai alkalmazások első bázisai voltak.

##### 4.13.1. MTA KKCS

1960-ban a neve Számítástechnikai Központra változott. Gépei: URAL-2, CDC 3300 (1969-ben).

##### 4.13.2. Nehézipari Minisztérium

1961-ben egy ELLIOTT-803B, angol gyártmányú gépet állítottak üzembe.

##### A gép jellemzői:

- Teljesen tranzistoros gép;
- Inputja – saját – *alfanumerikus* lyukszalag-olvasó, mely 500 jel/másodperc sebességgel működött;
- Outputja szalaglyukasztó, mely 100 jel/másodperc sebességgel működött;
- Ferritmemóriás gép, összesen 8102 szó kapacitással + 2 darab mágnesfilm-memóriával;
- Egycímű, fixpontos gép volt, de tartalmazott egy behuzalozott lebegőpontos műveleti lehetőséget is, mely egyáltalán nem elhanyagolható teljesítménnyel működött.
- Az alapgép műveleti sebessége:  
Összeadás/kivonás: 1700 művelet/másodperc;  
Szorzás: 1000 művelet/másodperc;  
Osztas: 80 művelet/másodperc;
- Lebegőpontos műveletek:  
Összeadás/kivonás: 1000 művelet/másodperc;  
Szorzás: 2000 művelet/másodperc;  
Osztas: 100 művelet/másodperc;
- A gép 39 bites szóhosszal rendelkezett. Ez a gép volt az ország első, magas szintű programozási nyelvvel (autokód) ellátott gépe.

#### 4.13.3. Kohó és Gépipari Minisztérium

Egy ELLIOTT-803 gép működött (1962-től);

#### 4.13.4. Számítástechnikai és Ügyvitel-szervezési Vállalat (SZÜV)

1953-ban alakult, akkor még *Hollerith-gépekkel* működött, később GAMMA-2B gépekkel (1961 környékén).

1964-ben kaptak egy GIER nevű, dán „csodagépet”, amelyen már ALGOL nyelven írt programokat lehetett futtatni. Ugyanebben az évben még egy URAL-2 gépet is üzembe helyeztek.

1966-ban beérkezett az ICL-gép, mely már multiprogramozásos gép volt, jó operációs rendszerrel dolgozott, és egyszerre 4 program futott rajta.

**Megjegyzés:** *A SZÜV a KSH-hoz tartozott, és az ország minden megyéjében volt központja.*

#### 4.14. ESEMÉNYEK, SZERVEZETEK

Ebben a pontban megemlítünk néhány olyan eseményt, amelyek a hazai számítástechnikai kultúra elterjesztése szempontjából fontos szerepet játszottak.

- a) Norbert Wiener, „A kibernetika atyja” 1960-ban Budapestre látogatott.
- b) 1960-ban az MTA-n létrejött az Elnökségi Kibernetikai Bizottság (Kalmár L. elnök, Frey Tamás, később Szelezsán János titkár).
- c) **Konferenciák:**
  - 1956-ban rendeztek egy Automataelmélet című konferenciát,
  - 1955-ben Közlekedési Kibernetikai Konferenciát rendeztek Budapesten.
- d) 1962-ben a MTESZ-ben létrejött az Automatizálási Információfeldolgozási és Operációkutatási Társaság (AIOT)  
1967-ben ez a társaság szervezett egy számítástechnikai konferenciát.
- e) 1964-ben – szintén az MTESZ-en belül – létrejött az IKOSZ (Információfeldolgozási, Kibernetikai és Operációkutatási Szakosztály), mely valójában az AIOT-ból szakadt ki, később pedig ebből jött létre a NJSZT (Neumann János Számítógéptudományi Társaság) 1972-ben.

#### 4.15. KÉPZÉSEK

- Az MTA KKCS-ben (M-3), 1959-1962 között számos tanfolyamot szerveztek. Ezeken „képződött ki” az a szakember gárda, amelyik később jelentős szerepet játszott a hazai számítástechnikai kultúrában.
- Az ELTE-n (1961-ben) Békéssy András matematikus hallgatóknak tartott szemináriumot, ahol az ALGOL-60-at tanította.
- 1963-ban Szelezsán János vegyészeknek és fizikusoknak tartott – tanterv szerint – programozási tárgyú előadásokat.
- A Közgazdasági Egyetemen („Közgáz”) a *Tervmatematikus szakon* (1960-ban) programozást és alkalmazást is tanítottak.

- A József Attila Tudományegyetemen (JATE), Szegeden, 1961-ben Kalmár László vezetésével öt hallgató végzett, akik programozási ismereteket is kaptak.
- A Mérnöktovábbképző Intézetben 1962-1963 között számítástechnikai tárgyú tanfolyamokat tartottak.

## **4.16. A HAZAI SZÁMÍTÓGÉPGYÁRTÁS**

### **4.16.1. A TPA család**

A TPA család kifejlesztése és „gyártása” is – mint már említettük – a Központi Fizikai Kutató Intézetben (KFKI) történt. Az „ős -TPA, a TPA-1100 volt”. Mintául a PDP-8-as (DEC) gépet választották. Ebből a gépből mintegy 30 db készült el.

A gépcsalád második lépcsőjét a TPA-i gép jelentette, amely már integrált áramkörös gép volt. 1972-től kezdve a gyártás jelentősen felfutott (1974-ben a fejlesztésben és a gyártásban 400 fő vett részt).

Az igazán sikeres termék a TPA-70-es jelű gép volt. Ennek a gépnek a fejlesztését 1969-ben kezdték el.

A TPA-70-es géppel egyidőben fejlesztették ki és gyártották a TPA-1140-es jelű gépet is, amely a PDP-11 gép koppintásának tekinthető.

1976-ban a TPA-70-es gép gyártását a VILATI (Villamos Automatikai Intézet) vette át.

A 70-es évek közepén a TPA-70-es számítógép iránti kereslet megnövekedett, jelentősebb export is volt belőle (Lengyelország, Szovjetunió, Csehszlovákia, India).

### **4.16.2. Az EMG-830-as számítógép**

Ezt a gépet az Elektronikus Mérőkészülékek Gyárában fejlesztették ki (a Honeywell gépek másolata volt). Összesen 16 db készült a gépből.

### **4.16.3. Az ESzR gépek Magyarországon**

Mint korábban említettük, 1968-ban a szocialista országok a KGST keretében egy Egységes Számítógép Rendszer (ESzR) kifejlesztéséhez fogtak hozzá. Mintául az IBM-360-as családot vették (sok tekintetben másolták is) Magyarországra az R10-es gép kifejlesztése, gyártása jutott. A gépet az 1969-ben létrehozott VIDEOTON Számítástechnikai Vállalat fejlesztette ki, ill. gyártotta, az SZKI (Számítástechnikai Koordinációs Intézet) és a MTA SZTAKI (Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete) közreműködésével.

A fentiekén kívül Magyarországon önálló számítógép fejlesztés, gyártás nem volt.

# 5 . F E J E Z E T

## PROGRAMOZÁSI NYELVEK

### 5.1. GÉPI KÓD, ASSEMBLER

Az első számítógépeket csak gépi kódban lehetett programozni. (Erre példát az M-3 gépnél láttunk.) A gépi kódú programok írását segítették az ún. mnemonikus kódok, a relatív címzés (a rekeszeket  $a+n$  alakban jelölhettük, ahol  $n$  az egész szám, pl.  $a+20$ ,  $b+12$  stb).

Már a korai szakaszban megjelentek az assembler típusú nyelvek is.

Az első, még nem univerzális és nem teljesen elektronikus számítógép, az ENI-AC idején felvetődött, hogy a gépi kód helyett szükség volna egy magasabb szintű nyelvre, legalábbis olyan leírási módra, amelyen az eljárás pontosan, jól leírható, úgy, hogy ennek alapján a gépi kód emberi közreműködéssel, vagy automatikusan előállítható legyen.

### 5.2. MAGAS SZINTŰ (MAGASSZINTŰ) NYELVEK

#### 5.2.1. Babbage-től, Ada Lovelacetől a FORTRAN-ig

##### 5.2.1.1. *Babbage, Ada Lovelace terve*

Már Babbage, és Ada Lovelace az 1800-as évek közepén kísérletezett az-  
zal, hogy a matematikai eljárásokat a gép által is érthető nyelven írják le.

Például a  $W_4 \times W_0 = W_{10}$  három című utasítást úgy vitték volna be a gépbe, hogy az  $\times$  műveleti jelet egy ún. műveleti lyukkártyára, (a kártyát akkor már a Jacquards szövőgépeknél használták), a 4, 0, 10 indexeket egy másik kártyára lyukasztották volna. Babbage és Ada Lovelace ilyen jelekkel a Babbage-gépre felírt egy programot.

##### 5.2.1.2. *Quevedo nyelve*

1914-ben Leonardo Torress Quevedo egy rövid programot írt fel formális nyelven az általa tervezett hipotetikus gépre.

##### 5.2.1.3. *Turing formális nyelve*

Jogosan említjük meg ebben a felsorolásban A. M. Turing angol matematikust is, aki egy univerzális absztrakt számítógépet (automatát, ma úgy hívjuk, hogy Turing-gép) definiált, amelyet voltaképpen a gép nyelvén felírt utasításokkal vezérelhetünk.

##### 5.2.1.4. *A Plankalkül*

A Neumann előtti korból különösen meg kell említeni Zuse *Plankalkül* nevű programnyelvét.

Zuse a Plankalkült egyszerű elemekből elegáns módon építette meg. Elsőként vezette be az adatokra a (típus) deklarációt. A következő módon járt el.

a) Az S0-lal jelölt adattípussal kezdte, amit egy bit jelölésére használt (ahogy a nyelv leírásában szerepel: Ja – Nein – Wert). Ennek értéke „–”, vagy „+” lehet (tehát nem 0 vagy 1). Minden  $\sigma_0, \dots, \sigma_{k-1}$  adattípus csoportból egy  $\langle \sigma_0, \dots, \sigma_{k-1} \rangle$  összetett adattípust lehet szerkeszteni, amelynek elemeire a 0, 1, ..., k–1 indexekkel lehet hivatkozni. A tömböt  $m \times \sigma$  alakban lehet deklarálni, ami m db azonos  $\sigma$  típusú tömbelemet jelent (ily módon többdimenziós tömb is deklarálható). A m jelenthet „ $\hat{m}$ ”-t is, amivel változó hosszúságú listát lehet deklarálni. (Ezt Zuse gráfok, algebrai formulák, a sakkjáték algoritmusainak megadására használta.) Az előbbiekkkel Zuse egy hierarchikus szerkezetű adatot tudott kezelni (fentről indulva eljutott a bit-ig).

b) Az integer változót A9-cel jelölte, a lebegőpontos számot pedig például

$$A\Delta 1 = \langle \times S0,7 \times S0,22 \times S0 \rangle$$

alakban adta meg. Itt az első három biten az előjelet és azt adta meg, hogy a szám valós, komplex vagy zérus. A bitek második csoportja a kitevőt, a harmadik pedig a mantisszát jelenti (7 ill. 20 bit).

Egyetlen nehézség volt a Plankalküllel, hogy egy utasítás több sorból állt, mert például az indexet a változó alá kellett írni. A  $R_0, V_0, Z_0$  változókat az

R,	V,	Z
0	0	0

alakban kellett megadni.

c) A változókat három csoportba sorolta:

- outputváltozó, jele  $R_k$  (Resultatwerte)
- inputváltozó, jele  $V_k$  (Variablen)
- közbülső (vagy segéd-) változó, jele  $Z_k$  (Zwischenwerte)

d) A tömbelemeket három sorban kellett felírni:

$$\begin{array}{c} V \\ 0 \\ i \end{array} \quad \text{a } V_0 \text{ tömb } i\text{-edik elemét jelenti.}$$

e) Az értékadás jele  $\Rightarrow$ .

**A mintafeladat:**

Legyen a feladat az  $\langle a_1, a_2, \dots, a_{10} \rangle$  10 elemű tömb elemeire az

$$y_i = \sqrt{|a_i|} + 5a_i^3$$

kifejezés kiszámítása, és amennyiben  $400 - y_i > 0$ , akkor eredményül az  $\langle y_i \rangle$  értékpár, ellenkező esetben az  $\langle 999 \rangle$  értékpár kinyomtatása. A továbbiakban más programnyelveket is ezen a feladaton mutatunk be; nevezzük ezért **Mintafeladatnak**.

**A mintafeladat programja** (5.1. ábra)

- Az 1. sorban egy A2 összetett változót deklarálása történik; ennek első eleme egy integer típusú, második pedig egy lebegőpontos szám.
- A 2-7 sorokban egy  $f(V)$  függvényt adunk meg (a „képlet értéke”)
- A 8-22 sorok a főprogram utasításait tartalmazza, itt a 11-14 sorok egyetlen utasítást jelentenek.
- A „W2(11)” egy iterációt jelent  $i = n-1$ -től 0-ig (0-t beleértve)
- $R1(V)$  az  $R_0$  eredményt jelöli a P1 eljárás x inputjának értékére.
- A program 15-18 sora az **if**  $Z_0 > 400$  **then**  $R_0 \langle 0 - i \rangle := \langle +\infty \rangle$  feltételes utasítást fejezi ki, ahol a  $\infty$  jel a 999 számnak felel meg („túl nagy érték”)
- A  $\rightarrow$  jel a feltételes vezérlés-átadás jele

A fenti néhány mondat feltehetőleg segíti az Olvasót abban, hogy a Zuse féle nyelvet, illetve az azon írt itt látható programot, legalábbis annak lényeges vonásait, megértse.

1	A2 = (A9, AΔ1)	
2	P1	R(V) ⇒ R
3	V	0 0
4	A	Δ1 Δ1
5		$\sqrt{ V } + 5 \times V^3 \Rightarrow R$
6	V	0 0 0
7	A	Δ1 Δ1 Δ1
8	P2	R(V) ⇒ R
9	V	0 0
10	A	11 × Δ1 11 × 2
11		W2(11) R1(V) ⇒ Z
12	V	0 0 0
13	K	i
14	A	Δ1 Δ1
15		$Z > 400 \rightarrow (i, +\infty) \Rightarrow R \left. \begin{array}{l} (10-i) \\ 0 \end{array} \right\}$
16	V	
17	K	
18	A	Δ1 9 2 9
19		$Z > 400 \rightarrow (i, Z) \Rightarrow R \left. \begin{array}{l} (10-i) \\ 0 \end{array} \right\}$
20	V	
21	K	
22	A	Δ1 9 Δ1 2 9

5.1. ábra: A Mintafeladat a Plankalkül nyelven

### 5.2.1.5. Neumann – Goldstine koncepciója

Neumann, aki már az ENIAC gépen is oldott meg feladatokat, cikkeiben kísérletet tesz a program (pontosabban a numerikus eljárás) szimbolikus jelekkel való megadására. Például így jár el a „Statisztikai módszerek a neutrondiffúzió vizsgálatára” c. munkájában (1947), amelyből egy rövid részletet a 2. ábrán láthatunk. E kísérletek nyomán Hermann 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 felesége), Neumann János és Arthur Burks is.

A kidolgozott „nyelv” filozófiája egészen más volt, mint a Zuse által megalkotott nyelvé.

Az előbbi, mint láttuk, nagy súlyt helyezett a típusdeklarációkra, míg Neumannék ezzel egyáltalán nem törődtek.

		Instructions (utasítások)	Explanations (magyarázat)
	1	r of $c_1 - 1$ , see /1/	$r_{i-1}$
	2	r of $c_1$ , see /1/	$r_i$
	3	$/c_3/ ^2$	$s^2$
	4	$/c_2/ ^2$	$r^2$
	5	3 – 4	$s^2 - r^2$
	6	$c_3 \begin{cases} \geq 0 \therefore A \\ < 0 \therefore B \end{cases}$	$s \begin{cases} \geq 0 \therefore A \\ < 0 \therefore B \end{cases}$
Only	7	$/1/ ^2$	$r_{i-1}^2$
For	8	5 + 7	$r_{i-1}^2 + s^2 - r^2$
B	9	$8 \begin{cases} \leq 0 \therefore B' \\ > 0 \therefore B'' \end{cases}$	$r_{i-1}^2 + s^2 - r^2 \begin{cases} \leq 0 \therefore B' \\ > 0 \therefore B'' \end{cases}$

5.2. ábra

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 évtizedeken keresztül a programterveket blokkdiagramokkal ábrázolták.

A Goldstine–Neumann diagram téglalapokból (bokszból, dobozokból) és az ezeket összekötő nyilakból állt. (Lásd az 5.3. ábrát.)



A dobozok négyfélék lehetnek.

- 1) 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 (értékadásokat jelentő) programrészek jelölésére szolgáltak az ilyen típusú dobozok.
- 2) **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.
- 3) **Helyettesítő dobozok**, amelyeket a # jellel jelöltek és ezekben a „→” jelet használták. Az ilyen a 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.)
- 4) 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 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  $2^{-2}$ , stb., 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

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

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  $a/b$  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 „képleten”.

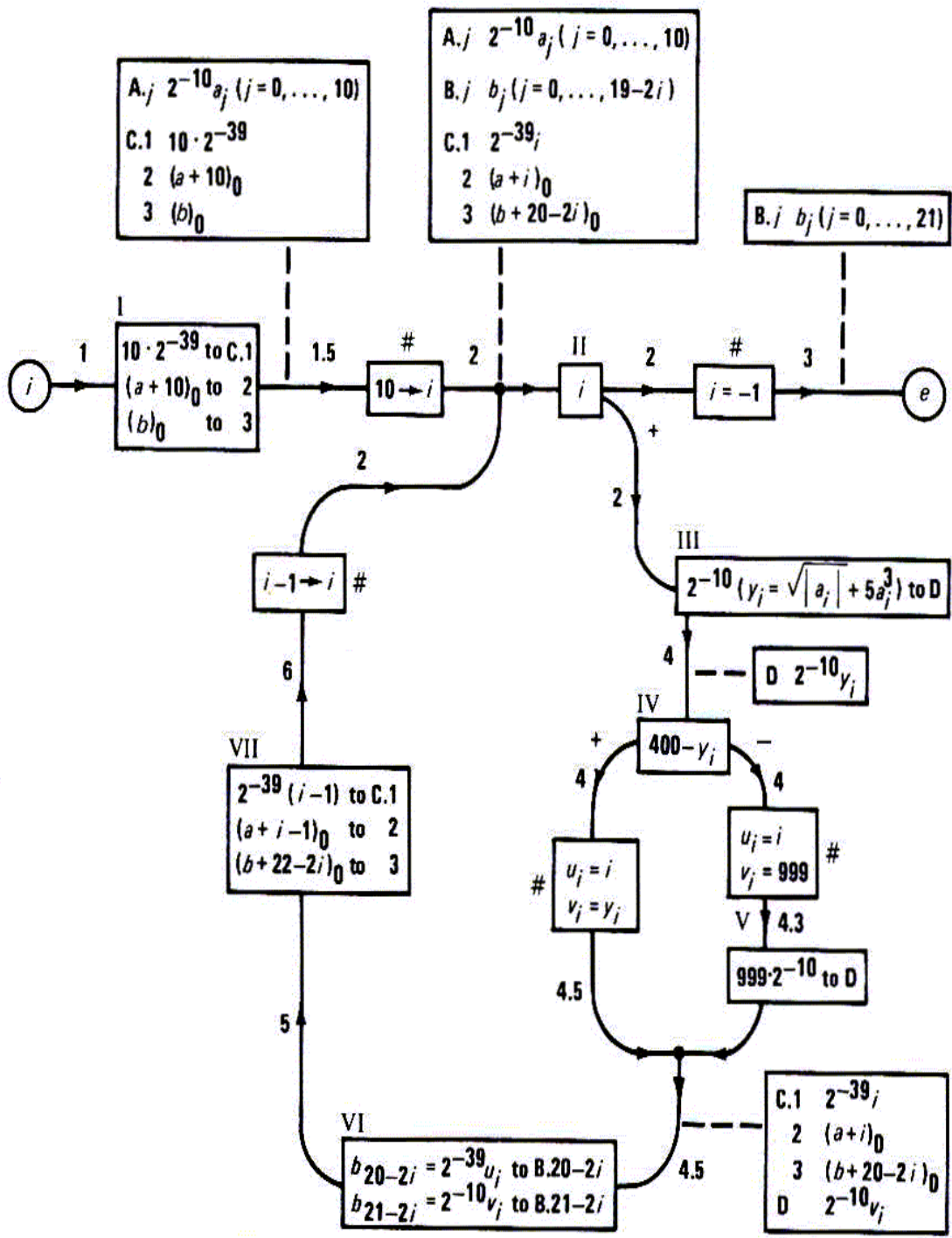
Az ábrán látható táblázatban szerepel még egy érdekes jelölés, például  $\overset{+}{\leftarrow} i$ ,  $\overset{-}{\leftarrow} i$ , stb. Általánosságban  $x_0$  azt jelentette, hogy  $x$  egy memóriacím (egy egész szám), mégpedig úgy, hogy a gépben  $x_0$  a  $2^{-15}x + 2^{-39}x$  alakban van jelen, 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).

Amikor egy folyamatdiagram elkészült, a programozó elkezdhette 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 a 3. ábrán látható folyamatdiagramot, ha mégis, akkor segítünk az-  
zal, hogy elmondjuk, hogy a diagram alapján készült program a **Mintafeladatot** oldja meg.

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



5.3. ábra

### 5.2.1.6. Burks „közbülső” programnyelve

Arthur W. Burks, aki részt vett a Neumann-féle blokkdiagram-nyelv kidolgozásában, 1950-ben a blokkdiagram és a gépi kód közé egy közbülső programnyelvet látott szükségesnek beiktatni.

A nyelvet a **Mintafeladaton** mutatjuk be (lásd az 5.4. ábrát).

A program mellé írt magyarázattal a nyelv lényege érthető.

<b>A z u t a s í t á s o k</b>	<b>M a g y a r á z a t</b>
1.	$10 \rightarrow i$
To 10.	
From 1,35	
10. $A + i \rightarrow 11$	Kiszámítja $a_i$ helyét
11. $\mathbf{A} + i \rightarrow t$	Átviszi $a_i$ -t a memóriába
12. $ t ^{\frac{1}{2}} + 5t^3 \rightarrow y$	$y_i = \sqrt{ a_i } + 5a_i^3$
13. 400, y;20,30	Megvizsgálja a $v_i = y_i$ feltétel teljesülését
To 20 if $y > 400$	
To 30 if $y \leq 400$	
From 13	
20. 999 $\rightarrow y$	$v_i = 999$
To 30	
From 13,20	
30. $\mathbf{B} + 20 - 2i \rightarrow 31$	Kiszámítja $b_{20-2i}$ helyét
31. $i \rightarrow \mathbf{B} + 20 - 2i$	$b_{20-2i} = i$
32. $\mathbf{B} + 20 - 2i + 1 \rightarrow 33$	Kiszámítja $b_{21-2i}$ helyét
33. $y \rightarrow \mathbf{B} + 20 - 2i + 1$	$b_{21-2i} = i$
34. $i + 1 \rightarrow i$	$i \rightarrow i + 1$
35. i,0;40,10	Ismétli a ciklust, amíg $i$ negatív nem lesz
To 40 if $i < 0$	
To 10 if $i \geq 0$	
From 35	
40. F	Megállítja a programot

5.4. ábra

### 5.2.1.7. Brooker Autokódja

Angliában, Manchesterben (mint tudjuk itt volt az angol számítógépek „bölcsője”) R. A. Brooke egy az előzőktől teljesen eltérő, a gépi kódhoz nagyon közel álló utasításokból álló nyelvet dolgozott ki. A már ismert Mintafeladat programja az utasítások mellé írt magyarázatokkal könnyen érthető. (5.5. ábra)

6	$n1$	=	1	$n1$ az 1 értéket veszi fel
1	$vn1$	=	1	$v_{n1}$ -be beolvasás (az input-ról)
	$n1$	=	$n1 + 1$	
	$j1, 11$	≥	$n1$	1-re ugrik, ha $n1 \leq 11$
	$n1$	=	11	
2	$*n2$	=	$n1 - 1$	Kinyomtatja $i = n1 - 1$ -et
	$v12$	=	$vn1$	
	$\beta, v12$	≥	$0 \cdot 0$	
	$v12$	=	$0 \cdot 0 - v12$	$v_{12}$ -be az abszolút értéket teszi
3	$v12$	=	$F1(v12)$	$\mu_{12} = \sqrt{ a_i }$
	$v13$	=	$5 \cdot 0 \otimes vn1$	
	$v13$	=	$vn1 \otimes v13$	
	$v13$	=	$vn1 \otimes v13$	$\mu_{13} = 5a_i^3$
	$v12$	=	$v12 + v13$	$\psi = f \mu_i$
	$j4, v12$	>	$400 \cdot 0$	
	$*v12$	=	$v12$	Kinyomtatja $y$ -t
	$j5$			
4	$*v12$	=	$999 \cdot 0$	Kinyomtatja 999-et
5	$n1$	=	$n1 - 1$	
	$j2, n1$	>	0	Teszteli a ciklust
		H		Megállás
		(j6)		Elindítja a programot

5.5. ábra

### 5.2.1.8. H.B.Curry programnyelve

H.B.Curry az ENIAC gépre hozott létre a Neumann Goldstine blokkdiagramra alapozva egy programnyelvet 1946-ban. Ma ezt „strukturált programozási nyelvnek” neveznék. A **Mintafeladat** programja ezen a nyelven a következő:

$$F(t) = \sqrt[3]{t} + 5t^3 : A$$

$$I = \{i\} \rightarrow L(i) \rightarrow F(i) \rightarrow A : y \rightarrow II \rightarrow \{i\} \rightarrow O_1 \& I_2$$

$$II = \{x = L(i) + 20 - 2i\} \rightarrow \{x\} \rightarrow III \rightarrow \{x = L(i) + 21 - 2i\} \rightarrow \{w\}$$

$$III = \{x > 400\} \rightarrow \{999 : y\} \rightarrow O_1$$

A, a gép akkumulátorát jelenti

Az egyes utasítástípusok jelentése a programban:

- $\{x\} : X$  : Az  $E$  kifejezés kiszámítása és elhelyezése  $X$ -ben
- $\{x = L(i)\} : E$  kiszámítása mint memória cím és ezt helyettesíti  $X$ -ként a soron következő utasításcsoportokban.

- c)  $X \rightarrow Y$ : Az  $Y$  utasításcsoport helyettesítése az  $X$  utasításcsoport első kimenetére.
- d)  $I_j$ : A program  $j$ -edik bemenete (azaz a  $j$ -edik utasítás csoport kezdete).
- e)  $O_j$ : A program  $j$ -edik kimenete.
- f)  $x > y \rightarrow 0_1 \& 0_2$ : ha  $x > y$  menj  $0_1$ -re egyébként  $0_2$ -re.
- g)  $I_{i+1}$ :  $i$  növelése 1-gyel, és ha  $i \geq m$  menj  $0_2$ -re egyébként  $0_1$ -re.

### 5.2.1.9. Rutishauser program nyelve

Rutishauser (Svájc) a Z4-gépre dolgozott ki egy nyelvet (1951). A **Mintafeladat** programja ezen a nyelven a következő volt:

- a) Für  $i = 10(-1)0$
- b)  $a_i \Rightarrow t$
- c)  $(\text{Sqrt Abs } t) + (5 \cdot t \cdot t) \Rightarrow y$
- d)  $\text{Max}(\text{Sgn}(y-400), 0) \Rightarrow h$
- e)  $Z0_i \Rightarrow b_{20 \times 2i}$
- f)  $(h \cdot 999) + ((1-h) \cdot y) \Rightarrow b_{21-2i}$
- g) Ende Index  $i$
- h) Schluss

$a \Rightarrow b$  jelentése:  $a$  elhelyezése  $b$ -be.

### 5.2.1.10. Böhm Compiler-nyelve

Corrado Böhm (Olaszország) 1951-ben dolgozott ki egy nyelvet. A nyelv érdekessége az volt, hogy a Compiler nyelvén definiálta magát a compilert. A **Mintafeladat** programja:

- 1)  $\pi' \rightarrow A$   
 $100 \rightarrow i$   
 $B \rightarrow \pi$
- 2)  $\pi' \rightarrow B$   
 $? \rightarrow \downarrow i$   
 $i+1 \rightarrow i$   
 $[(l \cap (i - 110)) \cdot C] + [(l - (i - 110)) \cdot B] \rightarrow \pi$
- 3)  $\pi' \rightarrow C$   
 $110 \rightarrow i$
- 4)  $\pi' \rightarrow D$   
 $\downarrow i \rightarrow x$   
 $E \rightarrow X$   
 $R \rightarrow \pi$
- 5)  $\pi' \rightarrow E$   
 $r+5 \cdot \downarrow i \cdot \downarrow i \cdot \downarrow i \rightarrow y$

$$[(I \cap (y - 400)) \cdot F] + [(I - (y - 400)) \cdot G] \rightarrow \pi$$

$$\begin{aligned} 6) \quad & \pi' \rightarrow F \\ & i - 100 \rightarrow ? \\ & 999 \rightarrow ? \\ & H \rightarrow \pi \end{aligned}$$

$$\begin{aligned} 7) \quad & \pi' \rightarrow G \\ & i - 100 \rightarrow ? \\ & y \rightarrow ? \\ & H \rightarrow ? \end{aligned}$$

$$\begin{aligned} 8) \quad & \pi' \rightarrow H \\ & i - 1 \rightarrow i \\ & [(I - (100 - i)) \cdot D] + [(I \cap (100 - i)) \cdot \Omega] \rightarrow \pi \end{aligned}$$

### Rövid magyarázat:

1.  $0 \rightarrow$  és a 100-as alapcím beállítása.
2.  $a_i$  beolvasása,  $i$  növelése 1-gyel, és ugrás  $C$ -re ha  $i > 10$ , különben  $B$ -re.
3. a 110-es cím beállítása.
4.  $a_i \rightarrow x$ , és a gyök kiszámítása az  $R$  szubrutinnal és az eredmény  $E$ -ben
5.  $f(a_i)$  kiszámítása és elhelyezése  $y$ -ban. Ha  $y > 400$ , folytatás 6-nál egyébként 7-nél.
6.  $i$  értékének és 999 nyomtatása, ugrás 8-ra.
7.  $i$  és  $y$  nyomtatása.
8.  $i$  növelése 1-gyel és 4-gyel folytatja, ha  $i \geq 0$  ellenkező esetben megállás.

Böhm bevezetett két érdekes műveletet:

a) Az  $-$  művelet a következő „logikai kivonást” jelenti:  $x - y = \begin{cases} x - y & \text{ha } x > y \\ 0 & \text{ha } x \leq y \end{cases}$

b) A  $\cap$  műveleti jel a minimum képzést jelenti, azaz  $x \cap y = \min(x, y)$ . Könnyen látható, hogy

$$1 \cap (i - j) = \begin{cases} 1 & \text{ha } i > j \\ 0 & \text{ha } i \leq j \end{cases}$$

$$1 - (i - j) = \begin{cases} 0 & \text{ha } i > j \\ 1 & \text{ha } i \leq j \end{cases}$$

Ezen műveletek segítségével valósítja meg a vezérlés átadást. A program hetedik sorának jelentése: 3-ra adja a vezérlést, ha  $i > 110$ , 2-re ha  $i \leq 110$ .

#### 5.2.1.11. A Ljapunov-féle operátor séma

Az ötvenes évek elején a Szovjetunióban is elindult az automatikus programozás (programozó programok) területén a fejlesztés. Ljapunov nevéhez

fűződik (1953) az ún. operátor séma kidolgozása, amelyhez több gépre is írtak fordítóprogramot.

Ezen a nyelven a **Mintafeladat** programja:

$$A_1 \lfloor Z_2 A_3 R_4 \lceil A_5 \lceil A_6 R_7 \lfloor A_8 N_9 \lceil A_{10} \lceil A_{11} F_{12} R_{13} \lfloor N_{14}.$$

$A \lceil, \lfloor$  operátorok jelentése: ugorj az  $n$  sorszámú operátorra igaz esetén,  $m$ -re hamis esetén.

$\lceil, \lfloor$  jelentése pedig: az  $i$  sorszámú operátortól ugrottunk ide.

Az  $A_i$  (aritmetikai)  $R_i$  (relációs) operátorokat külön kell specifikálni. A feladatnak megfelelően:

$R_4 \cdot p_1; 6, 5$	ha $p_1$ igaz, ugrás 6-ra, egyébként 5-re
$R_7 \cdot p_2; 8, 10$	ha $p_2$ igaz, ugrás 8-ra, egyébként 10-re
$R_{13} \cdot p_3; 14, 2$	ha $p_3$ igaz, ugrás 14-re, egyébként 2-re
$p_1 \cdot c_3 < v_2$	$[0 < x]$
$p_2 \cdot c_4 < v_3$	$[400 < y]$
$p_3 \cdot v_6 < c_3$	$[i < 0]$
$A_1 \cdot c_6 = v_6$	$[10 \Rightarrow i]$
$A_3 \cdot v_1 = v_2$	$[(a_1 \Rightarrow) x]$
$A_5 \cdot c_3 - v_2 = v_2$	$[0 - x \Rightarrow x]$
$A_6 \cdot \sqrt{v_2} + c_5 \cdot v_1 \cdot v_1 \cdot v_1 = v_3$	$[\sqrt{x} + (5 \cdot a_1 \cdot a_i \cdot a_i) \Rightarrow y]$
$A_8 \cdot v_6 = v_4, c_2 = v_5$	$[i \Rightarrow b_i, 999 \Rightarrow c_i]$
$A_{10} \cdot v_6 = v_4, v_3 = v_5$	$[i \Rightarrow b_i, y \Rightarrow c_i]$
$A_{11} \cdot v_6 - c_1 = v_6$	$[i-1 \Rightarrow i]$
$Z_2 \cdot v_1; 3, 6$	[ $a_i$ átvitele speciális rekeszekbe a 3-6 operátoroknál]
$F_{12} \cdot v_6; 2, 10$	[az $i$ -től függő címek módosítása a 2-10 operátorokban]
$N_9 \cdot BP 11$	ugrás a 11. operátorra
$N_{14} STOP$	megállás

Konstansok:  $c_1 \cdot 0,1 \cdot 10^1$   
 $c_2 \cdot 0,999 \cdot 10^3$   
 $c_3 \cdot 0$   
 $c_4 \cdot 0,4 \cdot 10^3$   
 $c_5 \cdot 0,5 \cdot 10^1$

Munkarekeszek:  $v_1 \cdot 130$   $a_i$  kezdőcíme  
 $v_2 \cdot 131$   $x$  címe  
 $v_3 \cdot 132$   $y$  címe  
 $v_4 \cdot 133$   $b$  kezdőcíme  
 $v_5 \cdot 134$   $c_i$  kezdőcíme  
 $v_6 \cdot 154$   $i$  címe



### 5.2.1.12. Egy deklaratív nyelv: az ADES

A Neumann elven (címezlehetőség elve) alapuló nyelveket *imperatív nyelveknek* nevezzük (a program „utasításokból”, „parancsokból” áll). A programozási nyelvek történetében egy másik vonal is kialakult a *deklaratív nyelvek*é. Ezek a rekurzív függvényeken alapulnak. A deklaratív nyelveknél a programozó a változók közötti relációkat írja (állításokat fogalmaz meg) fel, anélkül, hogy explicite megadná a kiértékelés sorrendjét.

Első ilyen nyelv volt az ADES, amelyet 1956-ban E.K. Blum dolgozott ki. Illusztrációként bemutatjuk a Mintafeladat programját ADES nyelven.

1.  $a_{011} : q_{011}$ ,
2.  $f_{50} = a \sqrt{\text{abs } c_1 \dots 5 c_1 c_1 c_1}$ ,
3.  $d_{12} b_1 = r_0$ ,
4.  $d_{22} b_2 = \leq b_3 400, b_3, 999$
5.  $b_3 = f_{50} a_0 r_0$ ,
6.  $r_0 = -10q_0$ ,
7.  $-0q_0 10b_0 = f_0 b_1 b_2$ .

Röviden elemezzük a programot:

1. **sor:** az  $a_0$  input tömb 11 elemű, és a  $q_0$  indexszimbólum a 11 értéket veszi fel.
2. **sor:** az  $f_{50}$  segédfüggvényt definiálja, (a Mintafeladat képletét) ún. Lukasziewicz zárójelmentes formában (szokták lengyel formának is nevezni).
3. **sor:** azt „állítja”, hogy a  $b_1$  változó azonos az  $r_0$  indexváltozóval;  $d_{12}$  azt jelenti, hogy outputként ez egy pár első eleme.
4. **sor:** egyrészt a  $b_2$ -t definiálja, mint egy pár második komponensét, másrészt a „ha  $b_3 \leq 400$ , akkor  $b_3$  egyébként 999” feltételt *állítja*.
5. **sor:** azt tartalmazza, hogy  $b_3$  az  $f_{50}$  függvény értékét veszi fel az  $a_0$ ,  $r_0$ -dik elemére.
6. **sor:**  $r_0$  értéke  $10 \cdot q_0$ .
7. **sor:** a teljes programfolyamatot adja meg:  $b_1$ -et és  $b_2$ -t a  $q_0 = 0, 1, \dots, 10$  értékekre kell „kiszámítani”.

### 5.2.1.13. Formula vezérelt számítógép

Az ötvenes évek közepén K.Samelson és F.L.Bauer felvetették, hogy olyan számítógépet kell kiépíteni, amely egyből „megérti” a formulákat; tehát nem kell a magas szintű nyelvet lefordítani gépi kódra, vagyis az algebrai formula maga a gép „nyelve”. 1955-ben meg is alkottak egy ilyen nyelvet (1957-ben szabadalmaztatták is).

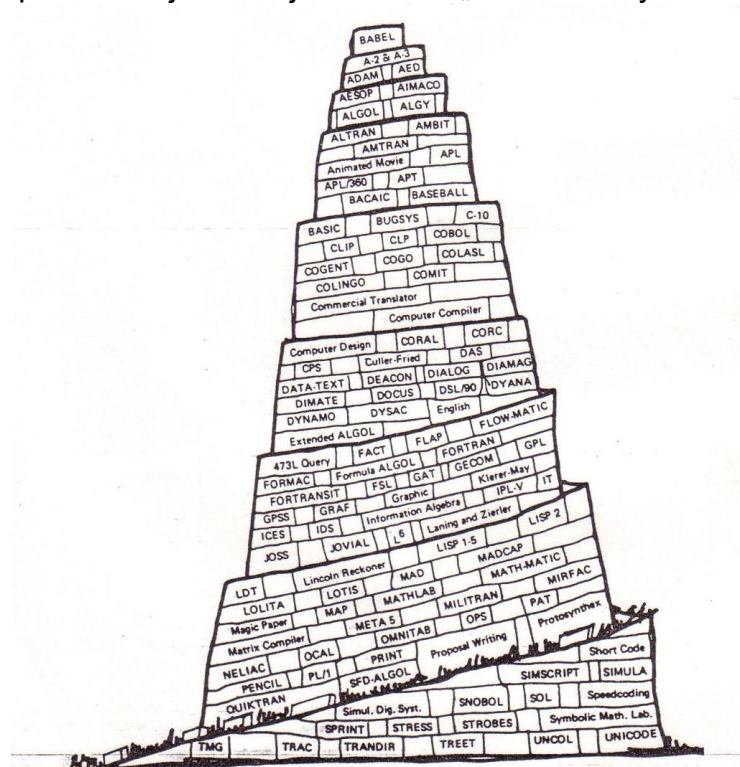
Ezen a nyelven a **Mintafeladat** „programja” a következő módon adható meg:

1	◇	0000.00000000 ⇒ a ↓11↑	2-12-ig adatok bevitele
2		2.27 ⇒ a ↓1↑	↓↑ indexjel
3		3.328 ⇒ a ↓2↑	
	...		
12		5.28764 ⇒ a ↓11↑	
13		10 ⇒ i	
14	44*	a ↓ i + 1 ↑ ⇒ t	
15		$\sqrt{Bt + 5 \cdot t \cdot tt} \Rightarrow y$	
16		i = □□ ⇒ i	
17		y > 400 → 77*	
18		y = □□□.□□□ ⇒ y	□..□ az output specifikációja
19		→ 88*	tizedespont kijelölésével
20	77*	999 = □□□ ⇒ y	
21	88*	i - 1 ⇒ i	
22		i > -1 → 44*	

### A programozási nyelvek robbanásszerű szaporodása

- 1960-ban 72,
- 1967-ben 117,
- 1971-ben 162 nyelv volt használatban.

Ezek nagy része rövid életű volt, kísérleteknek lehet tekinteni őket. 1972-ben csak 10 olyan nyelv volt, amelyik mindhárom fenti számban szerepelt. A nyelvek szaporodását jól mutatja az alábbi „Bábel-torony”:



5.6. ábra: A programozási nyelvek Bábel-tornya

## 5.2.2. Az életben maradt ősök és néhány leszármazott

### 5.2.2.1. A FORTRAN 0

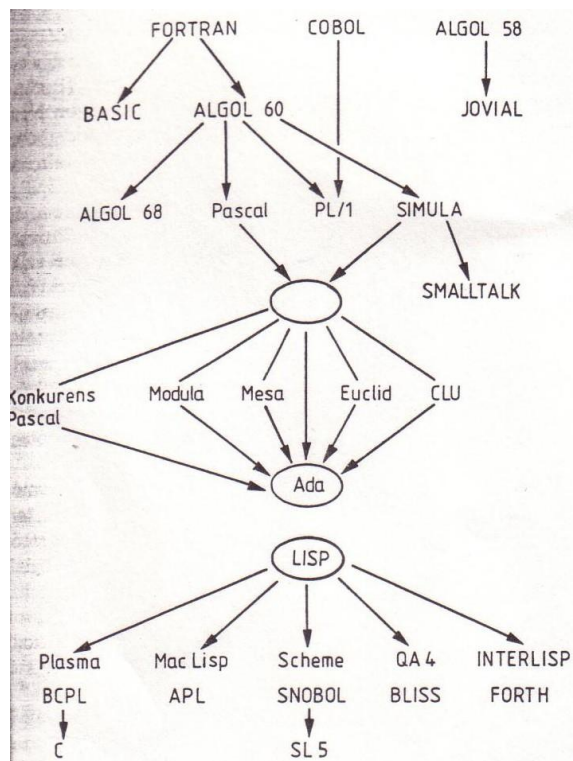
Felsorolásunkban eljutottunk a FORTRAN-hoz (FORMula TRANslating system = Formula fordító rendszer), az első „igazi” programnyelvhez. Az ötvenes évek nyelvei közül ez az egyetlen, amely még ma is „él” (persze számos korszerűsítésen ment keresztül).

A nyelvet az IBM-nél John Backus vezetésével dolgozták ki 1954-56 között.

Ez a nyelv már ipari jellegű volt abban az értelemben, hogy gépfüggetlennek tervezték.

Álljon itt Mintafeladatunk FORTRAN programja is. Azok számára, akik valamilyen programnyelvet ismernek, nyilvánvaló, hogy semmilyen magyarázat itt sem szükséges, hiszen a későbbi nyelvek utasításai „felfedezhetők” benne.

```
1      DIMENSION A(11)
2      READ A
3      DO 3,8,11, J=1,11
4      3  I=11-J
5      Y=SQRT(ABS(A(I+1)))+5*A(I+1)**3
6      IF (400 >= Y) 8,4
7      4  PRINT I,999.
8      GO TO 2
9      8  PRINT I,Y
10     11 STOP
```



5.7. ábra: Ősök és leszármazottak

### 5.2.2.2. A nagy „ősnyelv”, az ALGOL

Az ötvenes évek második felében egy nemzetközi projekt indult abból a célból, hogy létrehozzanak egy standard algoritmikus nyelvet. 1955-ben Darmstadtban, a Gesellschaft für Angewandte Mathematik und Mechanik (Alkalmazott Matematika és Mechanikai Társaság) égisze alatt tudósok egy csoportja indította el a folyamatot, majd csatlakoztak hozzájuk amerikai számítástechnikusok is.

1958 nyarán Zürichben egy bizottság (tagjai: J.W.Backus, F.L.Bauer, H.Bottenbruch, C.Katz, A.J. Perlis, H.Rutishauer, K.Samelson, J.H.Wegstein) az összegyűjtött javaslatok alapján „jóváhagyta” a nyelv alapjait, amelyet 1960-ig végleges formába öntöttek. A nyelvet először **IAL**-nek (International Algorithmic Language), majd később **ALGOL**-nak (**A**lgorithmic Language) nevezték el.

Az ALGOL nyelv tekinthető „minden nyelvek őségének”. Magát a nyelvet (szintaktikáját) egy metanyelven írták le (igen nagy precízséggel). Az ALGOL nyelv nem lett világnyelv. Ennek sok oka volt (többek között az, hogy egy nemzetközi tudós társaság és nem az IBM cég állt mögötte).

Illusztrációképpen álljon itt a **Mintafeladat** ALGOL-programja:

```
procedure Mintafeladat (a[ ] = : b[ ] ;  
array (a[0:10], b[0:21]);  
comment A program a Mintafeladatot számítja ki.  
begin for i : = 10(-1)0;  
begin y : = f(a[i]);  
      f(t):=sqrt(abs(t) + 5 · t ↑ 3 ↓ ;  
      if (y>400); y := 999;  
      b [20-20·i] : = i;  
      b [21-2·i]:= y  
end;  
return;  
integer (i)  
end Mintafeladat
```

### 5.2.2.3. A COBOL nyelv

Az előbbieken ismertetett nyelveket lényegében mind matematikai algoritmusok leírására hozták létre. Már az ötvenes évek végén felvetődött, hogy szükség volna egy tipikusan adatfeldolgozásra (könyvelés, raktárnyilvántartás, banki- kereskedelmi feladatok stb.) alkalmas nyelvre. Egy ilyen nyelv kidolgozására 1958-ban egy **CODASYL** (**C**onference on **D**ata **S**ystems Language) nevű bizottságot hoztak létre. A bizottság munkájának eredményeként megszületett a **COBOL** (**C**ommon **B**usiness **O**riented Language) nyelv.

1961-ben az IBM 1401-es gépre el is készült az első fordítóprogram.

A COBOL „lelke” nem a **változó** volt, hanem a mezőkből álló **rekord**. Egy COBOL programban a rekordok leírása (deklarálása) tette ki a nagyobbik részt, a procedurális rész egyszerű szerkezetű volt. (Aritmetikai, logikai kifejezéseket nem tudott kezelni.)

Illusztrációképpen bemutatunk egy COBOL programrészletet:

```
00001 010 IDENTIFICATION DIVISION.
00002 020 PROGRAM-ID. MINTA.
00003 030 AUTHOR. NAGY K.
00004 040 REMARKS. A PROGRAM EGY SPORTSZER-HANGSZER-JATEK
      : KERESKEDELMI VALLALAT HAVI ELADASAIROL KESZIT KIMUTATAST.
00022 220 01 ELADAS-REKORD.
00023 230 02 ARU-KOD.
00024 240 03 ELSO-KAR PIC 9.
00025 250 03 FILLER PIC 9(7).
00026 260 02 MEGNEVEZES PIC X(32).
00027 270 02 EGYSEGAR PIC 9(16)V99.
00028 280 02 MENNYISEG PIC 9(4).
00029 290 02 FILLER PIC C(28).
      :
00035 350 WORKING-STORAGE SECTION
00036 360 77 GYUJTO PIC 9(17)V99 VALUE 0.
00037 370 77 SZORZAT PIC 9(17)V99.
00038 380 77 HASONLITO PIC 9.
00039 390 77 SORSZAMLALO PIC 99 VALUE 25.
      :
00045 450 02 FILLER PIC X(52) VALUE SPACES.
00046 460 02 FILLER PIC X(28) VALUE „KIMUTATAS A HAVI ELADASOKROL”.
00047 470 02 FILLER PIC X(53) VALU SPACES.
      :
00066 660 PROCEDURE DIVISION.
00067 670 KEZDES.
00068 680 OPEN INPUT ELADASOK, OUTPUT KIMUTATAS.
00069 690 ELSO-KARTYA.
00070 700 READ ELADASOK AT END GO TO BEFEJEZES.
00071 710 MENTES.
00072 720 MOVE ELSO-KAR TO HASONLITO.
00073 730 FELDOLGOZAS.
00074 740 MULTIPLY MENNYISEG BY EGYSEGAR GIVING SZORZAT.
00075 750 ADD SZORZAT TO GYUJTO.
00076 760 MOVE AU-KOD TO ARU-KOD-2.
00077 770 MOVE MEGNEVEZES TO MEGNEVEZES-2.
00078 780 MOVE MENNYISEG TO MENNYISEG-2.
00079 790 MOVE SZORZAT TO ERTEK.
00080 800 PERFORM NYOMTATO-RUTIN
      :
00105 1050 IRATAS
00106 1060 MOVE GYUJTO TO OSSZEG.
```

```
00107 1070 WRITE SOR FROM OSSZEG-SOR AFTER 3.  
00108 1080 MOVE 25 TO SORSZAMLALO
```

#### 5.2.2.4. A BASIC nyelv

Ez a nyelv, amely voltaképpen a FORTRAN egyszerűsített változata, 1963-ra készült el. (Alkotói: a magyar származású Kemény János és Thomas Kurtz).

Az alkotók a programozás oktatására szánták. Később a nyelv az első személyi számítógép generáció nyelve lett. Az évtizedek során jelentős átalakításokon keresztül eljutott a Visual BASIC-ig.

#### 5.2.2.5. A PL/1 „világnyelv”

A 60-as évek közepén az IBM cég (amely ekkor már „diktált” a számítógép piacon) egy új „óriási nyelv” létrehozását szorgalmazta. Ez lett a **PL/1** (Programming Language/1).

A nyelv a FORTRAN, ALGOL, COBOL nyelvek szerencsés ötvöze. A FORTRAN-ból a formulakezelést, az ALGOL-ból a vezérlési szerkezetet (eljáráselv), a COBOL-ból az adatszerkezetet (rekord) vette át.

A PL/1 az informatika története folyamán a legelterjedtebb nyelv volt. (Számos ember kereste a kenyerét PL/1 programok írásával.)

Illusztrációképpen szolgáljon az alábbi programrészlet:

```
P: PROCEDURE (A) RETURNS (BIT(1));  
  DECLARE A(**), J;  
  DO J = 1 TO HBOUND(A,1);  
    IF ALL(A(J,*)¬=0) THEN RETURN (0);  
  END;  
  DO J = 1 TO HBOUND(A,2);  
    IF ALL(A(*)¬=0) THEN RETURN (0);  
  END;  
  RETURN ('1'B);  
END;
```

#### 5.2.2.6. LISP deklaratív (funkcionális) nyelv

A **LISP** (List Processing Language) nyelv az ADES nyelvhez hasonlóan nem imperatív, hanem deklaratív nyelv, azaz: a program elemei nem utasítások, hanem állítások (predikátumok). John McCarthy (Stanford Egyetem, USA) fejlesztette ki 1960-ban. A nyelv alapeleme az atom, adatszerkezete pedig egy lineáris lista.

Fontos attribútuma a nyelvnek a rekurzió. A nyelv egyik alkalmazási területe a mesterséges intelligencia.

#### 5.2.2.7. Az APL

Az **APL** (A Programming Language) nyelv az első olyan nyelvnek tekinthető, amellyel interaktív programfejlesztés volt végezhető. Valósidejű nyelv is, folyamatvezérlések leírására (programozására) is jól használható. Az IBM-nél Kenneth Iverson dolgozta ki 1962-ben.

#### **5.2.2.8. Az RPQ**

A RPQ egy „listageneráló nyelv” volt. Könnyen meg lehetett szerkeszteni vele - különösen a COBOL, PL/1 nyelveken írt programokból – az output táblázatokat, listákat. Fontos tulajdonsága volt, hogy speciális (kész) elemekből lehet összeállítani a programot (és annak eredményét a táblát, listát).

#### **5.2.2.9. A PASCAL**

A PASCAL nyelvet Niklaus Wirth (Zürich) fejlesztette ki a 70-es évek elején. A nyelvben sok új programozási filozófiát érvényesített (felülről lefelé programozás módszere, moduláris programstruktúra stb.).

A PASCAL nyelven jól oktatható a programozás.

#### **5.2.2.10. A MODULA-2**

A PASCAL-ból fejlesztették ki, a modularitás elvét erősítették, fontos eleme a rekurzió és a párhuzamosságot lehetővé tevő vezérlőszerkezet.

#### **5.2.2.11. ADA**

A PASCAL-hoz hasonló, erősen moduláris nyelv. Az USA Védelmi Minisztériuma számára készült a 70-es évek közepén. Fontos jellemzői a multitasking, megszakítás-kezelés, taskok közötti szinkronizáció, kommunikációs interfész.

#### **5.2.2.12. A C programnyelv**

A C nyelv „rendszerprogramozási nyelv”, operációs rendszerek eljárásainak, fordítóprogramok leírására tervezték. A nyelv megalkotója Dennis Ritchie (1972).

#### **5.2.2.13. Feladatorientál programnyelvek**

Már a programozási nyelvek őskorában felvetődött, hogy az általános (univerzális) nyelvek mellett szükség van ún. célnyelvekre, amelyeken egy-egy szakterület problémái programozhatók. Sok ilyen nyelvet alkottak.

### 5.2.2.14. Nyelvgenerációk

Az alábbi ábrán egy áttekintést adunk a magasszintű nyelvek generációjáról.

II. generációs nyelvek:

nyelv	év	vez.terv.	elődjei	célja
FORTRAN	1954-57	J.Backus (IBM)	-	tudományos szám.
ALGOL 60	1958-60	-	-	tudományos szám.
COBOL	1959-60	-	-	adatfeldolgozás
LISP	1956-62	J.Mc Carthy (MIT)	-	szimbolikus számítások
JOVIAL	1963	C.J.Shaw	ALGOL 60	rendszerprogramozás
SNOBOL	1962-66	R.Griswold (Bell Lab)	-	stringfeldolgozás
PL/I	1963-64	IBM	FORTRAN ALGOL 60 COBOL	általános célú
ALGOL-W	1966	N.Wirth (ETH Zurich)	ALGOL 60	általános célú
CORAL 66	1966		ALGOL 60	real-time

III. generációs nyelvek

nyelv	év	vez.terv.	elődjei	célja
SIMULA 67	1967	O.-J. Dahl Norw.Comp.C.	ALGOL 60	általános és szimulációs célra
ALGOL 68	1963-68	A. van Wijngaarden	ALGOL 60	általános célú
BCPL	1967-69	M.Richards	-	rendszerprogramozás
Bliss	1971	W.A.Wulf	ALGOL 68	rendszerprogramozás
Pascal	1971	N.Wirth (ETH Zurich)	ALGOL 60	általános és oktatási célra
C	1974	D.Ritchie Bell Lab	ALGOL 68 BCPL	rendszerprogramozás
Conc. Pascal	1975	P.Brinch Hansen	Pascal	párhuzamos progr.

IV. generációs nyelvek

nyelv	év	vez.terv.	elődjei	célja
Mesa	1974	(Xerox PARC)	Pascal SIMULA 67	rendszerprogramozás
ALPHARD	1976	W.Wulf,M.Shaw, R.L.London	-	általános célú
CLU	1974-77 (MIT)	B.Liskov	SIMULA 67	rendszerprogramozás
Euclid	1977	-	Pascal	rendszerprogramok helyességének biz.
Modula-2	1977	N.Wirth (ETH Zurich)	Pascal	rendszerprogramozás, real-time
Ada	1979	J.Ichbiah (CII Honeywell Bull)	Pascal SIMULA 67	általános célú, alkalmazó rendsz., real-time alk.



## Irodalomjegyzék:

1. Buzás Ottó (szerk.) Az e-kommunikáció kultúrája (2006)
2. Ceruzzi E. Paul: A History of Modern Computing (MIT Press, 1998)
3. Kovács Győző, Szelezsán János: Gondolatok Neumann János „First Draft of a Report on the EDVAC” című tanulmányáról (Nemzeti Tankönyvkiadó, 2003)
4. Metropolis N., Howlett, J., Rotar Cian-Carlo: A History of Computing in the Twentieth Century (Academic Press, New York, 1980)
5. Raffai Mária: Az informatika fél évszázada (Springer, 1997)
6. Shurkin Joel: Engines of the Mind (W.W. Norton, New York, 1984)
7. Szelezsán János: A programozás őskora (Természet Világa, Neumann Emlékszám, 2003)