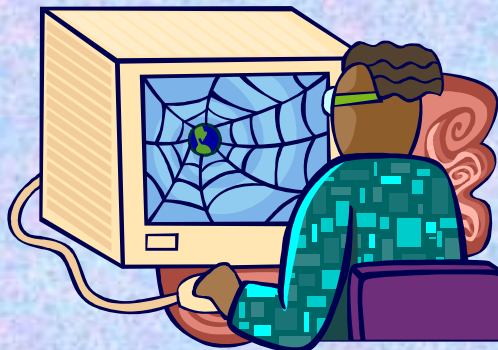


Fejezetek



az Informatika történetéből

11. fejezet

A rendszerszoftware megjelenése a számítástechnikában

Az EDSAC (Electronic Delay Storage Automatic
Computer)

A gép jellemzői

- 17 bites szóhossz
- Két 17 bites szó egy duplaszóvá egyesíthető
- 1024 szavas tároló (32 db 32 szavas „tank”)
- Szorzó regiszter
- „B” regiszter (=indexregiszter, csak az 1951 utáni továbbfejlesztésektől kezdve)
- Egycímű utasításrendszer (32 utasítás)
- Kb. 3000 elektroncső
- 5 csatornás lyukszalag I/O (eltérő input és output kód)

A központi egység regiszterei

- **71 bites akkumulátor (1 db)**
- **Szorzó regiszter**
(2 db, de közülük csak az egyik érhető el programból)
- **Utasítás számláló (Sequence Control)**
- **Utasítás regiszter (Order Tank)**

Utastíásszó alakja

Index bit (1951 után)

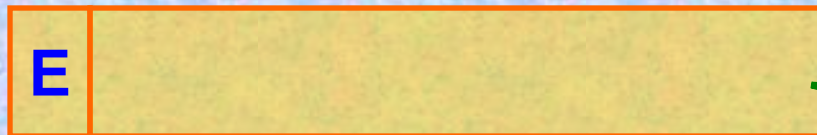


5 bit

10 bit

Spec. indikátor bit

Rövid adatszó:



16 bit

Fixpontos, 16 bit

Hosszú adatszó:



magasabb helyiértékek

alacsonyabb helyiértékek

35 (!!)

„Sandwich” bit

Utasításrendszer

UTASÍTÁS	HATÁSA
A_n	Az n cím tartalma hozzáadódik az akkumulátor tartalmához (<i>fixpontos!</i>)
U_n	Az akkumulátor tartalmának tárolása az n címen; az acc. változatlan
T_n	Mint az U_n , de az acc. törlődik
H_n	Az n cím tartalma beíródik a szorzóregiszterbe
V_n	Az n tartalma és a szorzóregiszter tartalma összeszorozódik, és az eredmény az acc.-hoz adódik

Utasításrendszer

UTASÍTÁS	HATÁSA
$L 2^{n-2}$	Az acc. tartalma n bittel balra lép
$B n$	n értéke beíródik az index- („ B ”) regiszterbe
$I n$	A soron következő ötbites kód beolvasása n -be (2^{-16} skálával!)
$O n$	n legfelső 5 bitjének perforálása (2^{-4} skála!)

Utasításrendszer

UTASÍTÁS	HATÁSA
$E\ n$	Ha az acc. tartalma ≥ 0 , akkor ugrás n -re
$F\ n$	Feltétlen ugrás n -re
$J\ m$	Ugrás m -re, ha az indexregiszter tartalma <u>nem</u> egyenlő 0 -val
$K\ m$	Egy $B\ b$ utasítás elküldése m -be, ahol b az indexregiszter pillanatnyi tartalma (b kimentése)

stb.

Az utasítások kódlaapon látható alakja („külső alak”)

$\langle \text{ut} \rangle ::= \langle \text{kód} \rangle [S] \langle \text{cím} \rangle [\pi] \langle \text{terminátor} \rangle$

$\langle \text{kód} \rangle ::= A | U | T \dots$ (stb.)

$\langle \text{cím} \rangle ::= \langle \text{decimális egész} \rangle | \langle \text{üres} \rangle$

$\langle \text{terminátor} \rangle ::= F | D | \theta | H \dots$ (stb.)

Ahol:

- **Kód**: az utasítás „mnemonikja”
- **Cím**: az utasítás címrésze
- **Terminátor**: az utasítás végjele, amely egyben a cím jellegére is utal
- **S (ha van)**: jelzi, hogy a címrész indexelendő
- **π (ha van)**: utasítás módosító, az indikátor bit=1

Néhány gyakran használt terminátor:

- **F**: a címzés rövid szóra vonatkozik
- **D**: a címzés hosszú szóra vonatkozik
- **θ**: a címzés relatív címre vonatkozik
- **H**: a 45-ös abszolút cím kódja

Példák az utasítások külső formájára:

- ***A 20 F*** → A 20-as cím tartalma+Acc
- ***BS 10 F*** → Az indexregiszter tartalmának növelése 10-zel
- ***TS H*** → Acc tárolása H + B-ben (H=45)

Az EDSAC beindítórutinja

- Az EDSAC operációs rendszer nélküli, mono-programozású gép volt, ezért minden program futtatása előtt „újra kellett indítani”
- Az utasítások átfordítását a külső ábrázolásból belső formára a beindítórutin (=Initial Input Routine) végezte

A beindítórutin funkciói

- **Kód („mnemonik”) helyettesítése a megfelelő bitkombinációval (a bitkombináció megegyezik a mnemonik teletype kódjával)**
- **A cím konvertálása decimálisból binárisba**
- **A terminátor felismerése és értelmezése**
- **A binárisra fordított utasítás eltárolása**
- **A vezérlő kombinációk (=direktívák) értelmezése, végrehajtása**

A beindítórutin a mai assembler és a kezdeti betöltő program („bootstrap loader”) elődjének tekinthető!

Vezérlő kombinációk

- **Vezérlő kombináció:** olyan kód, amely a beindító-rutinnak szóló parancsként értelmezendő
(*későbbi nevén: „assembler/compiler-direktíva”*)

VEZÉRLŐ KOMBINÁCIÓ	JELENTÉSE
G k	A pillanatnyi betöltési cím lesz a továbbiakban a relatív címzés alapja
T q K	A betöltési cím <i>q</i> -ra változik
E q K P F	A szalag beolvasása befejeződik, az akkumulátor törlődik és a betöltött program végrehajtása elindul a <i>q</i> címen

Példa egy főprogramból és három szubrutinból álló program „kiszzerelésére”

VEZÉRLŐ KOMBINÁCIÓ	JELENTÉSE
P Z	(üres szalagrész vége, „figyelemfelhívó”)
T 100 K	(első szubrutin 100-as címtől)
[SUBR 1]	(az első szubrutin utasításai)
T 200 K	(a második szubrutin a 200-as címtől)
[SUBR2]	(a második szubrutin utasításai)

Példa egy főprogramból és három szubrutinból álló program „kiszzerelésére”

VEZÉRLŐ KOMBINÁCIÓ	JELENTÉSE
T 300 K	(a harmadik szubrutin a 300 -as címtől)
[SUBR 3]	(a harmadik szubrutin utasításai)
T 400 K	(a főprogram a 400 -as címtől)
[FŐPROGRAM]	(a főprogram utasításai)
E 400 K P F	(a program elindul a 400 -as címen)

Az EDSAC gép működésével és programozásával
bővebben megismerkedhet az EDSAC szimulátor
segítségével!

MEGJEGYZÉS:

A szimulátor az EDSAC gép átadásának 50. évfordulójára
készült, és pontosan az átadáskori állapotot tükrözi, ezért
a későbbi fejlesztéseket (pl. „B” regiszter)

NEM TARTALMAZZA!

Az EDSAC csak 1-nél kisebb abszolút értékű számokkal tudott közvetlenül számolni, ezért számolásakor „skálafaktorokat” („*normáló tényezőket*”) kellett bevezetni

Pl.:

$$y = \frac{1}{1+x^2} \quad 1 \leq x \leq 10$$

helyett ezt számítjuk:

$$y = \frac{1/256}{1/256 + (x/16)^2}$$

(skálafaktor = 2^{-8})

Szubrutinkönyvtár *(kb. 90 szubrutin)*

- lebegőpontos aritmetika
- komplex aritmetika
- osztás *(nem volt behuzalozott osztási művelet!)*
- interpoláció és inverz interpoláció
- differenciálegyenletek
- numerikus integrálás
- dupla pontosságú *(két duplaszavas)* aritmetika
- matematikai függvények
- I/O

Az EDSAC jellemzői és jelentősége

- Alkotói elsősorban tudományos számításokra szánták
- Az elsőként elkészült, gyakorlatban is használható Neumann-elvű számítógép volt
- Megteremtette a „rendszeresoftware” alapjait
- Használat közben is folyamatosan fejlesztették mind a hardvert, mind a szoftvert
- 1958-ra elkészült az EDSAC 2
(*Sorozatgyártásra nem került*)

Fejezetek



az Informatika történetéből

12. fejezet

**A magasszintű programozási
nyelvek kialakulásának kezdetei**

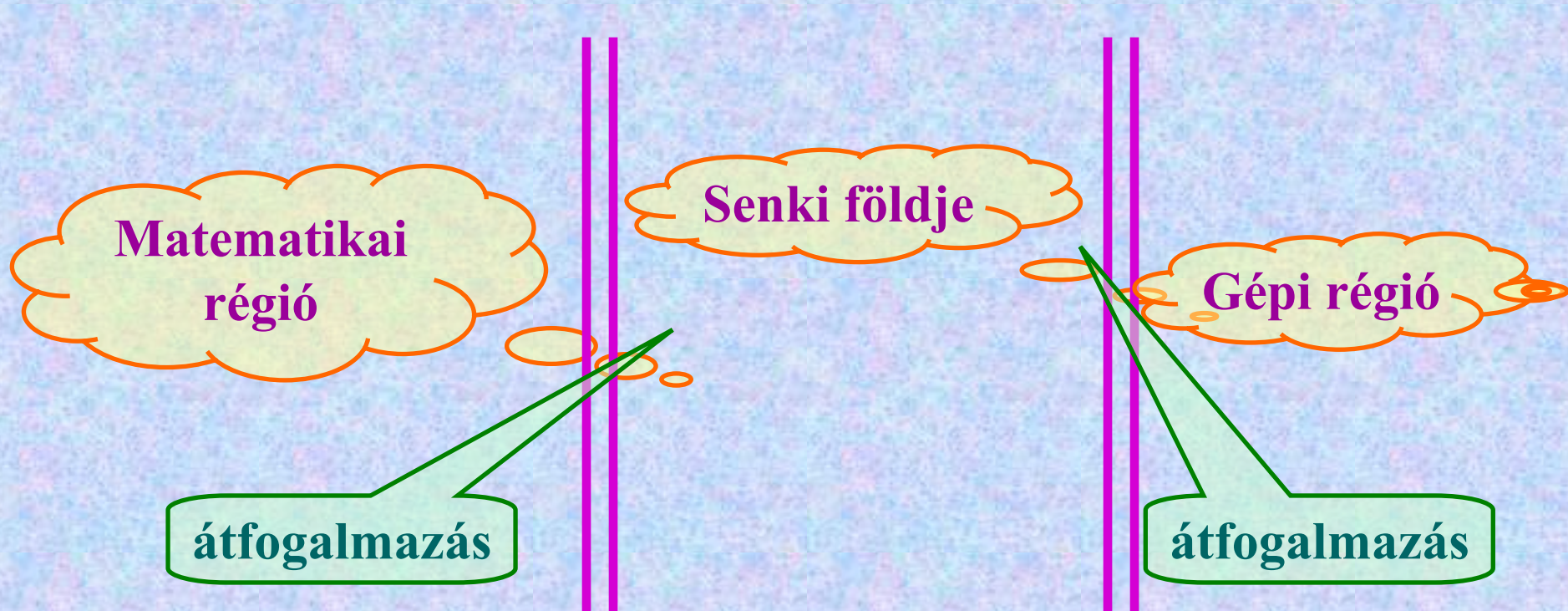
**Plankalkül, Folyamatábra, Böhm nyelve,
Manchester Autokód, ős-FORTRAN**

Magasszintű jelölésrendszerek algoritmusok ábrázolására

- **Motiváció:**

- Szabatos leírási módszer (*szimbolika*)
- Híd a matematikai és a számítástechnikai jelölésrendszer között
- Programozástechnikai segédeszköz
- Gépi program előállítás
(Költség: $\frac{1}{4}$ rész gép, $\frac{3}{4}$ rész programozás+belövés!!)

A problémamegoldás sematikus folyamata a programozás kezdeti korszakában



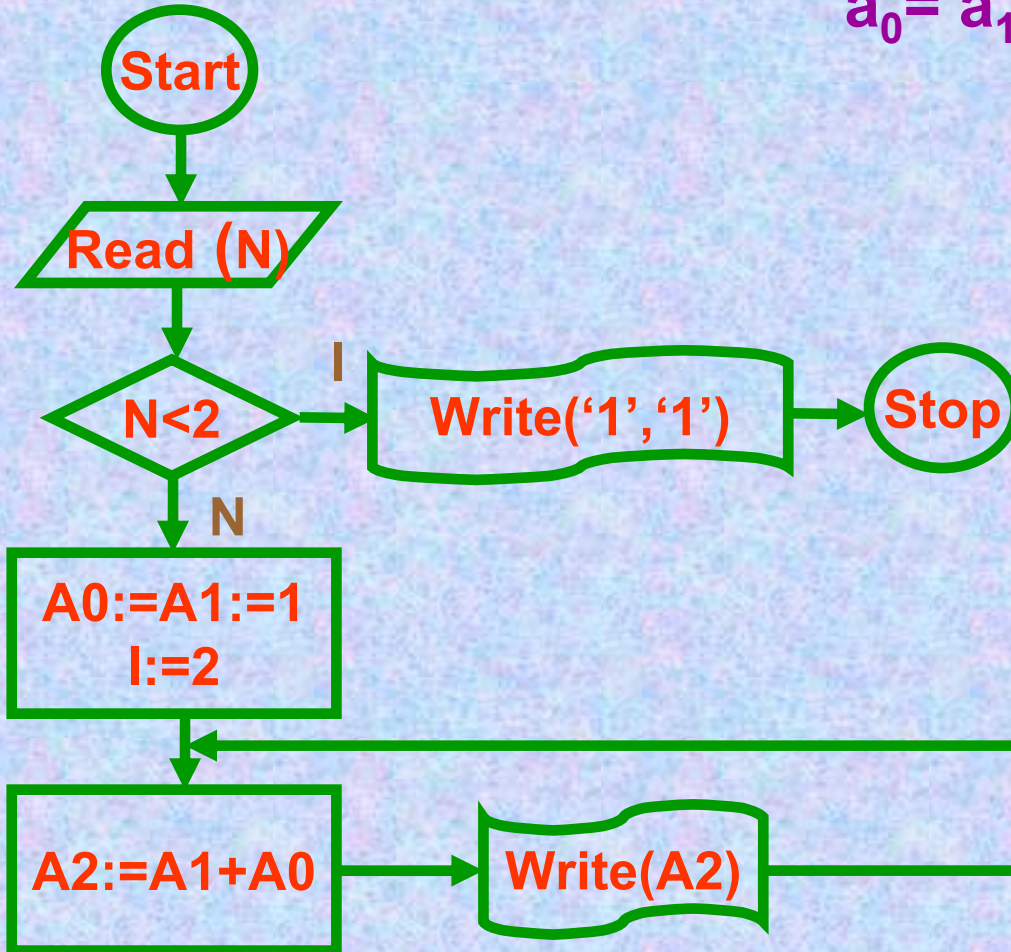
A cél kezdettől fogva:
A „senki földje” minél keskenyebbre szorítása

Neumann és Goldstine megközelítése: a folyamatábra (flow diagram)

- A „senki földjéhez” tartozó szöveg vegyesen tartalmaz a matematikai modellhez és a programhoz kapcsolódó elemeket. Az előbbiek tájékoztató jellegűek, az utóbbiak a gép által elvégzendő műveleteket tartalmazzák. A folyamatábrát a programozónak kell gépi kódba átültetnie.

Egyszerű programozási feladat a Fibonacci-számok generálására

Későbbi folyamatábra:



Matematikai definíció:

$$a_0 = a_1 = 1; a_n = a_{n-1} + a_{n-2} \quad (n = 2, 3, \dots)$$

Jellemzője:

- Csak „számítástechnikai” elemeket tartalmaz
- Minden „doboz” gépi nyelvre fordítható

- **Ötféle „doboz”** (mindegyik négyszögletes)
 - a) Operációs doboz („R”)
elágazás nélküli programrészek jelölésére
 - b) Alternatív doboz („R”)
logikai feltétel szerinti elágazás jelölésére
 - c) Helyettesítő doboz („#”)
híd a matematikai és a gépi jelölésrendszer között
 - d) Állítást tartalmazó doboz („#”)
lényegében kommentár
 - e) A memória állapotát leíró doboz
(szaggatott vonal + négyszögletes doboz)

A Neumann-Goldstine folyamatra tulajdonságai:

- A római számmal („R”) jelzett dobozok azonnal kódolhatók
- Az andrásskereszttel („#”) jelzett, az ábrán lila színű dobozok nem vesznek részt a számításban
- Az ábrán piros színű dobozok a memória állapotát jelzik
- **Kötött változók (*bound variable*):**
 - A program ad nekik értéket, kívülről hozzáférhetetlenek, a feladathoz „kötöttek”
(*pl. az indexek [=ciklusváltozók]*)
- **Szabad változók (*free variable*):**
 - A program be- és kimenő adatai, részeredményei
(*kívülről is hozzáférhetők*)

Jean E. Sammet (1928-)



- 1928 márc. 28.-án született New Yorkban
- 1961-ben az IBM-hez kerül
- 1972-74 között az ACM (első női) elnökhelyettese,
- 1974-76 között elnöke
- 1974-ben megalkotja a FORMAC-ot
- 1988-ban – 27 év után – megválnak az IBM-től

A programozási nyelvek robbanásszerű szaporodása: 1958-1972

- 1960: 72 nyelv...
- 1967: 112 nyelv...
- 1971: 162 nyelv...

...,„használatban” ...

...de ezekből mindössze 10 szerepel mindhárom kimutatásban!

Mikor „keletkezett” egy programozási nyelv?

(Sammet 11 válasza)

1. Amikor az ötlet először felmerült
2. Amikor először leírták
3. Amikor a nyelv előzetes specifikációját közreadták
4. Amikor a nyelv „végleges” specifikációját közreadták
(Mással: amikor eldöntötték, hogy mit valósítanak meg az előzetes specifikációból)
5. Amikor a fordítóprogram kísérleti futásai megkezdődtek
(esetleg a nyelv egy részhalmazára vonatkozóan)
6. Amikor a fordítóprogram teljes egészében megvalósult

Mikor „keletkezett” egy programozási nyelv? (*Sammet 11 válasza*)

7. Amikor nyelvet a fejlesztők éles feladatokon használni kezdték
8. Amikor a nyelvet a fejlesztőkön kívül mások is használni kezdték
9. Amikor a felhasználói kézikönyv elkészült
10. Amikor először publikáltak róla folyóiratban vagy konferencián

(Megjegyzés: ez nem ritkán megelőzi a 3. pontot!)

11. Amikor a nyelv módosításait, kiegészítéseit közzétették

Miért terveznek újabb és újabb nyelveket? (*Sammet 7 válasza*)

1. Új ötlet, új, lefedetlen alkalmazási terület
2. Rossz tapasztalatok egy meglévő nyelvvel
3. Több nyelvből egy $n+1$ -ediket
4. Új lehetőségek beépítésére van szükség, és ez nem oldható meg a meglévő nyelvek kibővítésével
5. Hobbyból
6. Személyes ellenszenv a meglévő és a célnak egyébként megfelelő nyelv(ek) ellen
7. A tervező - tájékozatlansága folytán - azt hiszi, hogy az 1. ill. 2. feltételt kielégítő dolgot talált ki

A számítógép-programozás történeti korszakai

1. **Felfedezés (- 1960)**
2. **Rendszerezés (1960 - 1970)**
3. **Professzionális tevékenység (1970 -)**

1. Más megfogalmazásban:

1. **Kísérletezés**
2. **Matematikai megalapozás**
3. **MéRNÖKI gyakorlat**

2. Még másképpen:

1. **Művészet (esetleg fekete mágia?)**
2. **Szakma**
3. **Ipar**

A magasabb szintű programozás kezdetei

- Kezdetben a bináris, natív gépi kódú programozás dominál
- Utasításrendszer hiányosságainak elfedése egyszerű pszeudokódok (értelmező rendszerek) segítségével (pl. osztás, lebegőpontos és komplex aritmetika, szögfüggvények, stb.)
- Egyszerű átcímző, betöltő programok (pl. az EDSAC beindítórutinja)
- Első kísérletek magasszintű nyelvek létrehozására (autokódok, formulafordítók)
- Az első „nagy” felfedezés: a stack (verem, Samelson & Bauer)

Néhány példa korai magasszintű nyelvekre

Corrado Böhm (1923-) nyelve a Z4 gépre

– A nyelv jellemzői:

- 14 jegyű egész számokon működik
- Csak értékadó utasításai vannak
- Egyszintű indirekt címzés
(↓ *i* jelentése: *i* tartalma, mint cím)
- Műveleti jelek: mint a matematikában, továbbá:

→ értékadás művelete (*balról jobbra*)

∴ pozitív (*logikai*) különbség $x ∴ y = \begin{cases} x - y & \text{ha } x > y \\ 0 & \text{ha } x \leq y \end{cases}$

∩ minimumképzés $x \cap y = \min(x, y)$

- Aritmetikai kifejezésekben vagy teljes zárójelezés, vagy precedencia *(a kettő nem keverhető)*
- Fordítóprogram: egymenetes, saját nyelvén megírva; kb 130 értékadó utasítás, ebből 33 címke ($\pi' \rightarrow X$ alakú, ld. alább!)

A programozás történetében ez volt az első nyelv, amelynek fordítóprogramját magán a nyelven írták meg

π'
 π jelentése: **utasításcím**

$X \rightarrow \pi$ ugorj az X címére (*GOTO X*)

X értéke az utasításslámlálóba kerül, a program végrehajtása innen folytatódik

$\pi' \rightarrow X$ a betöltési cím pillanatnyi értékének elmentése az X rendszerváltozóba, amit a program később ugráscímként használhat. Ezt az utasítást a *betöltő*, és nem a *végrehajtandó* program értelmezi!

Logikai értékek ábrázolása Böhm nyelvén

$$1 \cap (x : y) = \begin{cases} 1 & \text{ha } x > y \\ 0 & \text{ha } x \leq y \end{cases}$$

$$1 \div (x : y) = \begin{cases} 0 & \text{ha } x > y \\ 1 & \text{ha } x \leq y \end{cases}$$

Feltételes ugró utasítás Böhm nyelvén

$\pi' \rightarrow X$

...

$80 \rightarrow p$

...

$\pi' \rightarrow Y$

...

A fenti utasítások után:

$[(1 \cap (100 \div p)) \cdot X] + [(1 \div (100 \div p)) \cdot Y] \rightarrow \pi$

annyi, mint:

IF $p < 100$ Then Goto X Else Goto Y

A Manchester autokód

(R. A. Brooker 1953, Manchester)

A nyelv tulajdonságai:

- „Középszintű” nyelv
(gyakorlatilag háromcímű utasításrendszer)
- Index- ($n1, n2\dots$), és lebegőpontos *(valós)* változók
($v1, v2\dots$)
- Gyorsan, könnyen elsajátítható, és *(értelmező program alatt)* hatékonyan megvalósítható
- Felhasználói szubrutinok nem definiálhatók

Autokódokat később is használtak, bár ezek a magasszinű nyelvek elterjedésével visszaszorultak
(pl. ELLIOTT 803 gépek)

Egyszerű program a 201-as és 301-es címtől elhelyezett vektorok skaláris szorzatának kiszámítására

$n1 = 201$ *{Első vektor kezdőcíme}*
 $n2 = 301$ *{Második vektor kezdőcíme}*
 $v99 = 0$ *{A skaláris szorzat nullázása}*
 $7\ v98 = vn1 \times vn2$ *{A „7” címke; vn1,vn2 indexes változók}*
 $v99 = v99+v98$
 $n1 = n1+1$ *{Index növelése}*
 $n2 = n2+1$
 $j7, 280 \geq n1$ *{Feltételes ugrás „7”-re, ha $n1 \leq 280$ }*

A Pascal-szerű kommentek nem részei a nyelvnek!

John Warner Backus (1924- 2007)



1924-ben született,
Philadelphiában

- 1949-ben matematikus diplomát szerez a New York-i Columbia Egyetemen
- 1950-ben belép az IBM-hez
- 1954-ben publikálja a **FORTRAN-t**

John Warner Backus (1924- 2007)



- 1959-ben kidolgozza a BNF-et
- 1977-ben Turing-díjat kap.
Ennek átadási ünnepségén tartja meg „*Can Programming Be Liberated...*” című híres előadását
- 2007 jún.1.-én (halála után) a **FORTRAN** megalkotásáért „*Johnbackus 6830*” néven aszteroidát neveznek el róla

Ős-FORTRAN (FORTRAN0) az IBM 704 gépre

Backus (1954)

```
DIMENSION A(11)
READ A
2 DO 3,8,11 J=1,11
3 I=11-J
  Y=SQRT(ABS(A(I+1)))+5*A(I+1)**3
  IF (400>=Y) 8,4
4 PRINT I,999
  GOTO 2
8 PRINT I,Y
11 STOP
```

A nyelv sajátos tulajdonságai:

- A ciklusutasításnak (*DO*) öt paramétere van: a ciklusmag kezdetének, végének és a folytatás helyének címkéje, valamint a ciklusváltozó kezdő és végértéke
- A későbbi *CONTINUE* utasítás (amely a ciklusmag végét jelezte) nem létezik
- A ciklusmag belsejéből, annak kezdetére való visszaugrást (ma *Loop* vagy *Next*), a *Do* címkéjére való ugrással kellett programozni
- „kétágú” *IF* (relációjelek: =, >, >=)
- A kiírási kép formázása (*FORMAT*) nem lehetséges
- Szubrutinhívási lehetőség és szegmentálás nincs

Fejezetek



az Informatika történetéből

13. fejezet

FORTRAN, ALGOL, BASIC

A két „első generációs” programozási nyelv: a FORTRAN és az ALGOL

- FORTRAN (1954-58):
 - kártyakép-orientált írásmód
 - mellérendeltségi viszonyban álló szubrutinok (*szegmensek*)
 - nem ismer globális változókat, csak közös adatmezőket (**COMMON**) lehet deklarálni
 - a kor technológiai színvonalához képest jó I/O lehetőségek
 - egy adott cég (**IBM**) támogatása áll mögötte

- ALGOL (1958-63):
 - szabad formátumú írásmód
 - alárendeltségi viszonyban álló blokkok és eljárások
 - a hivatkozási nyelvben semmilyen I/O nincs
 - „civil szervezetek” nemzetközi együttműködésével készült, egyik cég sem érzi igazán sajátjának

- Ezek közös „leánynyelve”: a **BASIC** (1963-64):
 - sor orientált írásmód (F)
 - a forrásprogram nem tagolható (*szegmentálható*) (A)
 - nincs blokkstruktúra (F)
 - korlátozott (*később továbbfejlesztett*) I/O lehetőség (F)
 - „nem professzionális” felhasználók támogatása áll mögötte (K)

F=FORTRAN „örökség”

A=ALGOL „örökség”

K=Különleges tulajdonság

A FORTRAN nyelv kifejlesztésének szempontjai és körülményei

- Korábban nem volt általánosan elfogadott „automatikus programozási” rendszer
- A létező „kódkiegészítő” rendszerek hatékonysága nem volt szempont

(csak a lebegőpontos műveletek, mert ezek fordultak elő leggyakrabban)

- Az IBM 704 megjelenésével (~1953) ez a helyzet megváltozott

(beépített lebegőpontos műveletek és indexregiszterek)

- **Cél:** a gépi kódú programozásnál lényegesen *(legalább kétszeresen)* hatékonyabb programozási rendszer kifejlesztése,
- **amely** végrehajtási időben sem marad el lényegesen a gépi kódban írt programoktól.

A project első vezetője: John Backus
*(akinek később az ALGOL-60 kidolgozásában is
döntő szerepe volt)*

Rövid kronológia

- 1953: **Az IBM 704 megjelenése**
- 1954. nov.: **A „FORTRAN0” előzetes specifikációja**
- 1954-1956. II. n.é.: **A fordítóprogram tervezése, megírása**
- 1956. II. n.é.-1957. II. n.é.: **A fordítóprogram „belövése”**
(összesen kb. 25 ember-évnyi munka!)
 - Közben: **a felhasználói dokumentáció elkészítése**
- 1958: **Már 50-80%-os felhasználási arány**

Fortran kézikönyv 1956

Rövid kronológia

- 1958. II n.é.: **FORTRAN-II**
 - szegmensenkénti (*szubrutinonkénti*) fordítási lehetőség, utólagos „összeszerkesztés”
- 1959-60: **FORTRAN-III** (*csak az IBM 704-en*)
 - sok gépfüggő elem került bele, de ezeken kívül lényegében azonos a „végleges” FORTRAN-nal (*FORTRAN-IV, ill. FORTRAN 66*)
- 1977-??: **Későbbi Fortran változatok**

1977 óta „Fortran”
(Nagy kezdő-, és utána kisbetűvel)!

```
C      A FORTRAN1 MAR ISMERI A KOMMENTEKET!      1
C      A SORSZAMOZAS NEM RESZE A PROGRAMNAK!    2
      FUNF(T)=SQRTF(ABSF(T))+5.0*T**3             3
      DIMENSION A(11)                            4
1     FORMAT(6F12.4)                              5
      READ 1,A                                     6
      DO 10 J=1,11                                 7
      I=11-J                                       8
      Y=FUNF(A(I+1))                               9
      IF(400.0-Y)4,8,8                             10
4     PRINT 5,I                                   11
5     FORMAT(I10,10H TUL NAGY!)                   12
      GOTO 10                                     13
8     PRINT 9,I,Y                                 14
9     FORMAT(I10,F12.7)                           15
10    CONTINUE                                    16
      STOP 52525                                  17
```


Az ALGOL nyelv fejlesztésének előzményei, célja, története

- 1955: GAMM (*Gesellschaft für Angewandte Mathematik und Mechanik*) konferencia az „automatikus programozás” témájában
- 1956: GAMM albizottság alakul
- 1957: egyidőben Európában (GAMM) és az USA-ban (ACM – *Association for Computing Machinery*) döntés születik, amely szerint új, egységes programozási nyelvet kell létrehozni
- 1957. október: A GAMM az ACM-hez fordul az erőfeszítések egyesítése érdekében
- 1958. máj. 27.-jún. 2.: Közös konferencia Zürichben; eredményeként létrejön az „Ó-ALGOL” (ALGOL-58)

Célkitűzések

- A nyelv *jelölésrendszere* a lehető legközelebb essen a szokványos matematikai jelölésmódhoz, és minimális magyarázatokkal olvasható legyen
- *Publikációkban* alkalmas legyen numerikus számítási eljárások leírására
- Az új nyelvet nehézség nélkül lehessen *magával a géppel* gépi kódra fordítani

A nyelv előzetes definíciója (Backus [ed.]):
„The syntax and semantics of the proposed international algebraic language of Zurich ACM-GAMM conference”

Az ALGOL 60 végleges változata

- 1958-59: Nyilvános vita a tervezet fölött
- 1959. febr.-: ALGOL Bulletin (*Peter Naur*), CACM algoritmus-, és vitarovat. Számos előkészítő megbeszélés a „nagy végső” konferenciára
- 1960. jan. 11.-16.: ALGOL-60 konferencia, Párizs

A végleges(nek szánt) jelentés (Backus et al):
„Report on the algorithmic language ALGOL-60”
Módosítva: 1962

Az ALGOL 60 által megvalósított nyelvi újdonságok, jelentős gondolatok

- Szilárdan megalapozott *megvalósítási modell*
- Nemzetközi együttműködés keretében dolgozták ki
- A szintaxis leírására formális módszert alkalmaz **(BNF)**
- Lökést ad a *formális nyelvek elméleti kutatásainak*
- Egyesíti magában az algoritmus-definíciós és a programozás-automatizálási szempontokat
- Publikációs nyelv
- Gépfüggetlenség elve
- Blokkstruktúra, azonosítók érvényessége **(hatáskör)**
- Rekurzív eljárások és eljáráshívások
- Számos programozási nyelv alapja, „**kútfő**”

Can Programming Be Liberated...

- Backus a **FORTRAN** kifejlesztéséért 1977-ben Turing-díjat kapott
- A díjátadásakor tartott előadásának címe: „Megszabadítható-e a programozás a Neumann-stílustól?”
- Backus előadásában azokat a programozási nyelveket bírálja, amelyek a Neumann-elvű számítógép „leképezésének” tekinthetők
- Szerinte olyan programozási nyelvekre lenne szükség, amelyek nem a Neumann-elvű számítógépek architektúráján alapulnak

Milyenek a „Neumann-stílusú” programozási nyelvek?

Válasz: amilyen a Neumann-elvű számítógép!

- **Az adatokat a memóriában tárolja**
 - a memória-rekeszeknek a változók felelnek meg
- **A memóriarekeszek tartalma felülírható, változtatható**
 - a program végrehajtása gyakorlatilag a memória állapotváltozásainak sorozatát jelenti
- **Az adatok a memória és az aritmetikai egység között áramlanak**
 - a program folyamatosan címeket és adatokat áramoltat a két egység között

Milyenek a „Neumann-stílusú” programozási nyelvek?

Válasz: amilyen a Neumann-elvű számítógép!

- **A gép a programot általában sorosan hajtja végre**
 - a program végrehajtásának középpontjában az értékadó utasítások állnak
- **A sorrendtől való eltéréseket *(feltételes ill. feltétlen)* ugró utasítások vezérlik**
 - az értékadó utasításokat vezérlő utasítások veszik körül

Backus érvei a Neumann-stílusú nyelvek ellen

- Minden program végső soron az „egyszerre csak egy szót” elvre bontja le a teendőket
- A kifejezések és az utasítások külön világot alkotnak
- A program változóinak *nevük* van, amelyeket *(ha ugyanazt a tevékenységet más elnevezésű változókon kívánjuk végrehajtani)* csak az *eljárások bonyolult apparátusával* lehet felülírni
- Meglevő programokból újabbakat létrehozni nagyon körülményes *(vagy lehetetlen)*
- A számítógépek *(Neumann-elvű)* architektúrája megnehezíti rajtuk *nem Neumann stílusú nyelvek* létrehozását és alkalmazását

Backus javaslata:

funkcionális programozási nyelvek használata

- A funkcionális nyelvek struktúrált adatokra alkalmazott függvények kompozíciójából épülnek fel
- Többnyire nem repetitívek és nem rekurzívak
- Megkönnyítik a programok matematikai tulajdonságainak vizsgálatát, valamint meglévő programokból újabbak előállítását

Újabb 30 év után...

- Backus cikkének megjelenésekor a Neumann-architektúra kb. 30 éves volt. Backus ezután haláláig (2007) a funkcionális programozás kutatásával foglalkozott
- A funkcionális programozásnak jelentős irodalma van, számos, részben vagy teljesen funkcionális nyelv létezik
- Nem Neumann elven alapuló architektúrák kifejlesztése irányában is történtek lépések, de a mai számítógépek alapvetően még mindig Neumann-elvűek

Kemény János (1926-1992)



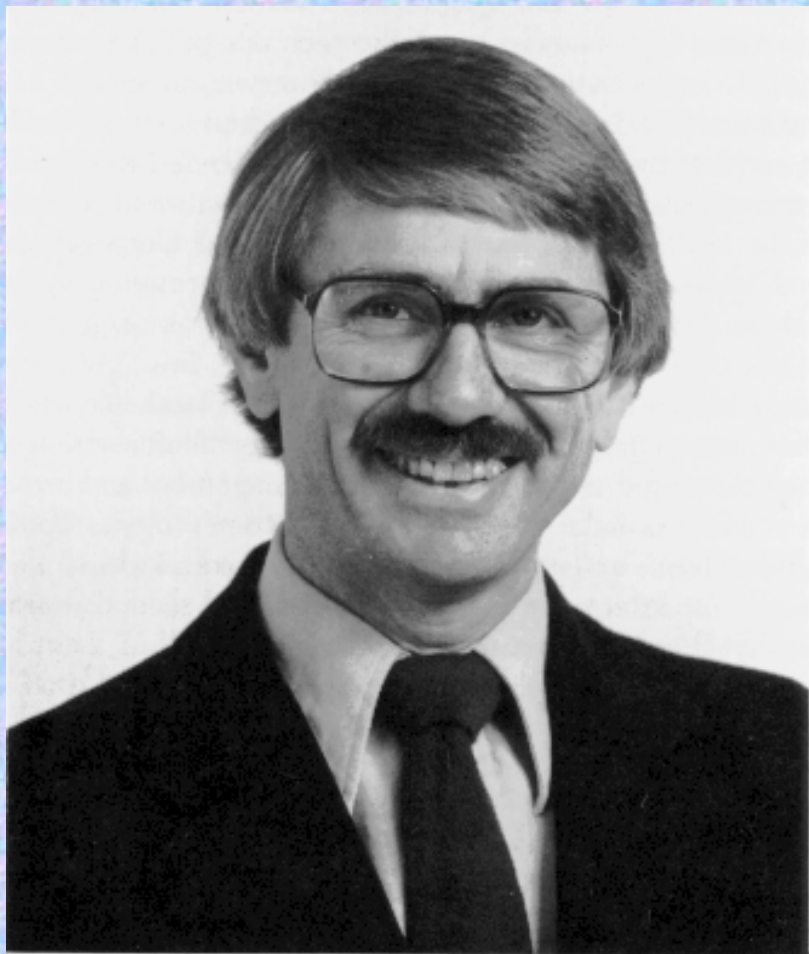
- 1926 máj. 31-én született Budapesten. Apja nagykereskedő
- 1940: az apa a nácik elől New Yorkba menekíti a családot
- 1945: Princetonba, majd Los Alamosba kerül, itt Feynmannel és Neumannal dolgozik együtt

Kemény János (1926-1992)



- 1949: Princetonban megvédi doktorátusát
- 1953: átmegy a Dartmouth College matematikai intézetébe
- 1963-64: *T. Kurtz*-cal együtt megalkotják a **BASIC**-et
- 1979: a Three Mile Island-i reaktorbalesetet kivizsgáló bizottság elnöke
- 1992 dec. 26: **halála**

Thomas Eugene Kurtz (1928-)



- 1928-ban született
- 1956: Princetonban megvédi PhD-jét. Ugyanebben az évben belép Dartmouth-ba
- 1963-64: Kemény Jánossal megalkotják a **BASIC**-et
- 1966-75: a College számítóközpontjának igazgatója
- 1991: a Számítástechnika **Úttörői (Computer Pioneer Award)** díjjal tüntetik ki

A BASIC nyelv kialakulása, tervezési szempontjai

- A BASIC szülőhelye:
 - Dartmouth College
- Alapítva:
 - 1769
- Az intézmény eredeti rendeltetése:
 - „az indián törzsek gyermekeinek írásra, olvasásra és egyéb ismeretekre tanítása”
- Az egyetem jellege:
 - Kevesebb, mint 25% természettudományi hallgató, a többi humán, közgazdász, stb.
- A Basic nyelv kidolgozói:
 - Thomas E. Kurtz és John Kemeny

- Kemény és Kurtz céljai:
 - Számítástechnika oktatása elsősorban humán hallgatóknak
- Dartmouth első számítógépe:
 - 1959, LGP-30 (*4k dobmémória*)
- A szerzők döntése:
 - ALGOL és FORTRAN alapú, de új nyelv kell
- 1964:
 - A Dartmouth C. egy GE-225 gépet kap, melyet később egy GE-265-re cserélnek
- 1964 május 1., hajnali 4 óra:
 - Lefut a világ első BASIC programja

Az első Dartmouth időosztásos BASIC rendszer fontosabb jellemzői

- Hallgatók korlátozás nélküli hozzáférése
- Könnyű elsajátíthatóság
- Elviselhető fordulási idők
- Felhasználóbarátság

A közhiedelmekkel ellentétben, az első BASIC implementáció **fordító-**, és nem **értelmező** programmal működött. Az értelmező program használata később terjedt el.

BASIC mintaprogram

BASIC

```
10 INPUT "Írja be a nevet: "; U$
20 PRINT "Hello "; U$
30 REM
40 INPUT "Hány csillagot ohajt: "; N
50 S$ = " "
60 FOR I = 1 TO N
70 S$ = S$ + "*"
80 NEXT I
90 PRINT S$
100 REM
110 INPUT "További csillagok (I/N)? "; A$
120 IF LEN(A$) = 0 THEN GOTO 110
130 A$ = LEFT$(A$, 1)
140 IF (A$ = "I") OR (A$ = "i") THEN GOTO 40
150 PRINT "Viszlat";
160 FOR I = 1 TO 200
170 PRINT U$; " ";
180 NEXT I
190 PRINT
```

{U\$ string változó}

{A forrás tagolása}

{N numerikus változó}

{S\$ törlése}

{S\$ feltöltése}

{ karakterekkel}*

{Ciklusmag vége}

{S\$ kiírása}

{A\$ első karaktere}

{Ciklus, ha igen}

{Elköszönés}

{Sorminta}

{Sorminta lezárása}

A BASIC további története

- Kemény és Kurtz 1964 - 1971 között a BASIC nyelv hat, egymásra épülő változatát dolgozta ki
- A nyelv szabványosítása az 1970-es évek közepén kezdődött, amikor már igen sok BASIC változat létezett
- Két ANSI szabvány keletkezett:
 - Minimal BASIC (X3.60-1978)
 - Full BASIC (X3.113-1987)
- A szabványokat később az ISO is átvette (1984 ill. '91).

A BASIC további története

- A BASIC nyelv nem rögtön lett népszerű. Rohamos elterjedése a mini-, majd a személyi számítógépek elterjedésével indult meg. Kurtz szerint a '80-as évek elején már a világ legismertebb programozási nyelve volt
- 1982-ben kísérlet történt egy „univerzális” BASIC nyelv szabványának megalkotására. Ez az irányzat szembement az eredeti tervezési koncepciókkal, és zsákutcának bizonyult
(a tervezet terjedelme 252 oldal volt!)
- A BASIC mai *objektumorientált* utóda a Visual Basic, amely grafikus felület készítését is lehetővé teszi

Fejezetek



az Informatika történetéből

14. fejezet

Tudósaink

Az informatika meghonosítói Magyarországon

Nemes Tihamér (1895-1960)



- 1895. április 29.-én született, Budapesten
- 1929-től a Postakísérleti Állomáson dolgozik
- 1953-tól részt vesz az első magyar színes televízió kifejlesztésében
- 1949-ben egy tanulmányában a számítógépek elve alapján a kétlépéses sakkfeladványok mechanikus megfejtését tárgyalja

Nemes Tihamér (1895-1960)



- Kidolgoz egy logikai gépet, amellyel különböző ok-és okozati kapcsolatok automatikusan felismerhetők
- 1957: A műszaki tudományok doktora
- 1960. márc. 30.-án Budapesten meghal
- 1962: „Kibernetika” c. könyvének megjelenése (*posztumusz*)

Kozma László (1902-1983)



- 1902. nov. 28.-án született, Miskolcon
- 1925: A brünni egyetem ösztöndíjas hallgatója
- 1930: Fejlesztőként dolgozik Antwerpenben, a Bell Telefongyárban
- 1942-ben hazatér
- 1949: Koholt vádak alapján börtönbe zárják
- 1958: A MESZ-1 üzembe helyezése
- 1983 nov. 9.: Budapesten meghal

Kalmár László (1905-1976)



- 1905. márc. 27.-én született,
Alsóbogátpusztán
(Somogy m.)
- Érettségi után a budapesti
Pázmány Péter
Tudományegyetem
matematika-fizika
szakán tanul
- 1927-ben diplomázik

Kalmár László (1905-1976)



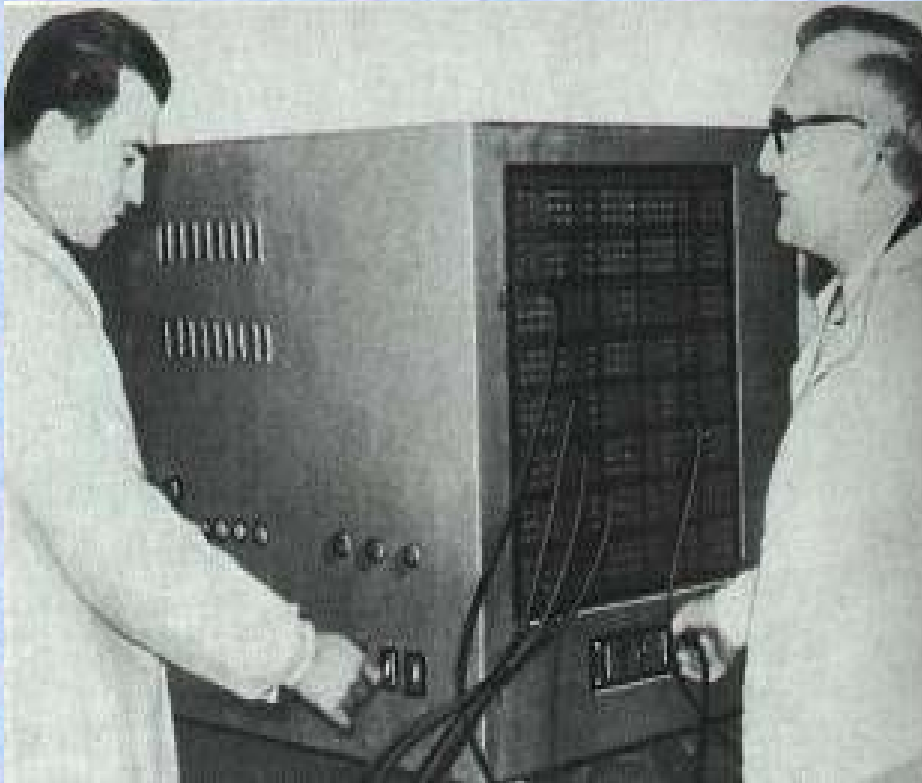
- 1930-ban adjunktus a Bolyai Intézetben
- 1947-ben megörökli *Riesz Frigyes* tanszékét, és egyetemi tanárrá nevezik ki
- 1949-ben a Magyar Tudományos Akadémia levelező,
- 1961-ben pedig rendes tagjai közé választja

Kalmár László (1905-1976)

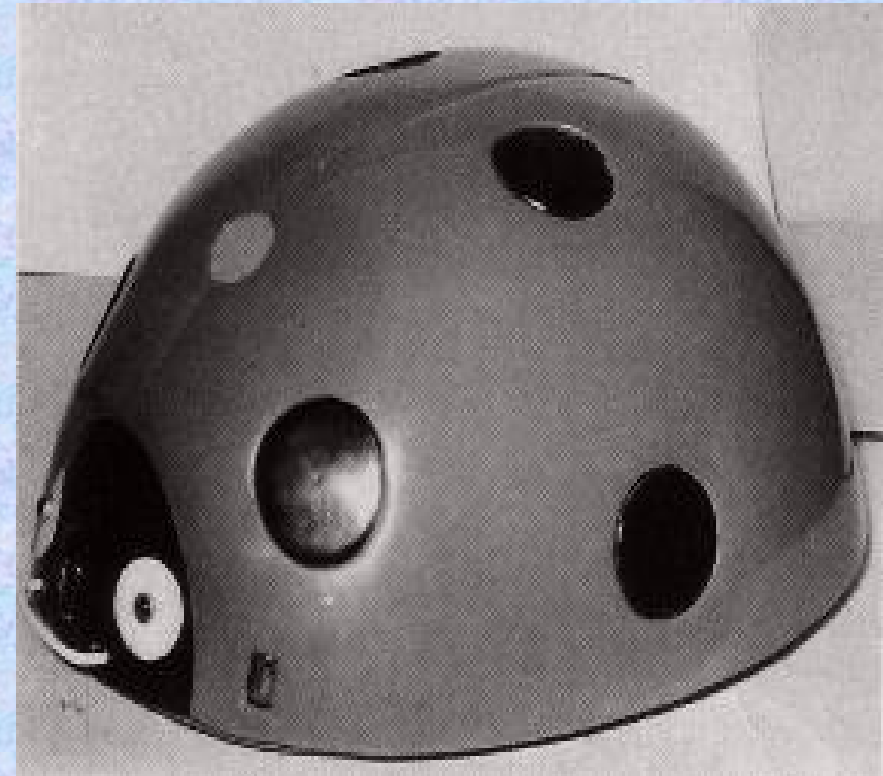


- 1963: **JATE Kiblabor megalakulása**
- 1971: **Bolyai Intézet számítástudományi tanszék vezetője**
- 1976. aug. 2.: **Mátraházán meghal**
- Fő érdeklődési területe: **matematikai logika**

Megvalósult alkotásai



A logikai gép...
*(a képen munkatársával,
Muszka Dániellel)*



**...és a katicabogár, az első
magyar kibernetikai gép**

Nem megvalósított nagy terve: *a formulavezérelt számítógép*

- Formulavezérelt számítógép: „gépi kódja” valamilyen formális nyelv
- Elvi akadály: nincs
- Probléma:
 - a bemeneti nyelv bonyolultsága
 - hordozhatóság

Tarján Rezső (1908-1978)



- 1908 jan. 6-án **született Budapesten**
- Tanulmányait **Bécsben végzi**
- 1953-ban **koholt vádak alapján börtönbe zárják**
- 1956-1958: **a KKCS igazgatóhelyettese**
- 1960-tól **haláláig az OMFB szakértője**
- 1965: **az újonnan megalakult AIOT (A Neumann Társulat őse) elnöke lesz**
- 1978 dec. 21.: **Budapesten meghal**

A kezdetek

A **K**ibernetikai **K**utató **C**soport (*KKCs*)

Az 1950-es évek közepén a számítástechnika terén Magyarországon az általános tájékozatlanság volt jellemző. Tisztázatlan volt pl. a kibernetika fogalma és kapcsolata a számítástechnikával („*kibernetikai gép*”).

Előzmények

- 1953-54: az MTA felismeri a számítógépek jelentőségét
- 1954 febr. 7.: a KÖMI-401 cég (*Általános Épület- és Géptervező Iroda, Gyűjtőfogház*) ajánlatot tesz az MTA-nak számoló (számító-)gép építésére
- 1954: MTA tájékoztató a számítógép felépítésének adatairól:
 - Kb. 500 elektroncső, 2500-3000 germániumdióda, 3 szabványos telefonkeretre szerelve
 - tervezés: 15-18 hónap
 - építés: 6 hónap
 - olcsóbb (~\$100.000) egy azonos kategóriájú nyugati gépnél

- **1955: az MTA Méréstechnikai és Műszerügyi Intézetében (MÉMI) Számológép Osztály alakul Tarján Rezső vezetésével (memória fejlesztési és kibernetikai kutatások)**
- **1956 szeptember: megalakul a KKCs**
 - igazgató: Varga Sándor
 - helyettes: Tarján Rezső
- **A KKCs feladatai:**
 - számítógépekkel kapcsolatos kutatások
 - kibernetikai kutatások
 - számítógéptudományi ismeretek terjesztése
- **1957: megkezdődik az ország első elektronikus számítógépének (M-3) építése**

Az M-3 jellemzői

- Tervrajzok és alkatrészbazis: szovjet
- Szóhossz: 31 bit
- Utásításrendszer: kétcímű (*2×12 bit*)
- Műveleti kód: 6 bit
- Tároló: mágnesdob, később ferrit, *1024 szó*
- Aritmetika: fixpontos
- Input: 5 csatornás lyukszalag
(*teletype, később fotodiódás lyukszalagolvasó*)
- Output: teletype
- Műveleti sebesség: 30 műv/sec (*a dob lassúsága miatt*)
Ferrittárolóval kb. 1500-ra növekedett
- Átadás: 1959
- Operációs rendszer: nincs

A KKCs tevékenysége

- Gépi numerikus módszerek
- Gazdasági alkalmazások
(a csoport tagja: Kornai János)
- Műszaki alkalmazások
- Matematikai nyelvészet
- Matematikai logika és alkalmazásai
- Számítástechnikai szolgáltatás és tanácsadás

**A pillanatnyilag „súlyponti” téma erősen a pillanatnyi igazgatótól
(és az MTA elvárásaitól) függött!**

A KKCs és az M-3 számítástechnika történeti jelentősége

- Létrejött egy **szakmai műhely**, amely **(az országban először)** képes volt felépíteni és üzemeltetni egy számítógépet
- Kialakult egy **szakembergárda**, amely a feladat kitűzésétől a kódolásig és futtatásig végig tudott vinni különböző feladatokat
- Kialakult egy **felhasználói kör**, amelyben felébredt az igény a számítógépek alkalmazására a legkülönbözőbb műszaki és gazdasági területeken

A KKCs felfutó korszaka után

- Vezetési problémák: Varga Sándor autoriter vezetési módszerei és egyéb hibái kiélezték a helyzetet
(1958: *Tarján távozása*, 1960: *Varga leváltása*)
- Az MTA és KKCs viszonya: a KKCs profilja nem illeszkedett sem az MTA III. (*matematikai és fizikai*), sem a VI. (*műszaki*) osztályának szemléletéhez
- Az MTA koncepciója: alkalmazás fejlesztés, kiegészítő kutatások
- Új számítóközpontok megalakulása, az M-3 elavulása
(1961, KFKI: *Ural I.*, 1962, NIM: *Elliott 803*)

- 1960: a KKCs átalakul, Számítástechnikai Központ néven folytatja működését. Munkatársai egyre gyakrabban kénytelenek külső központok gépeit igénybevenni
- 1965: 2 évi tárolás után egy Ural-2 gép áll üzembe. Egyidejűleg az M-3 Szegedre kerül, oktatási célokra *(működött 1967-ig)*
- **Az Ural-2 problémái *(nemcsak a Központban)*:**
 - gyárilag előállított, de még elektroncsöves gép
 - rendkívül megbízhatatlan
 - még az Ural-1-gyel sem kompatibilis
- 1972: a Központ nyugati gépet kap *(CDC 3300)*
- 1973: összevonják az AKI-val *(Automatizálási Kutató Intézet)*, így jött létre a SZTAKI

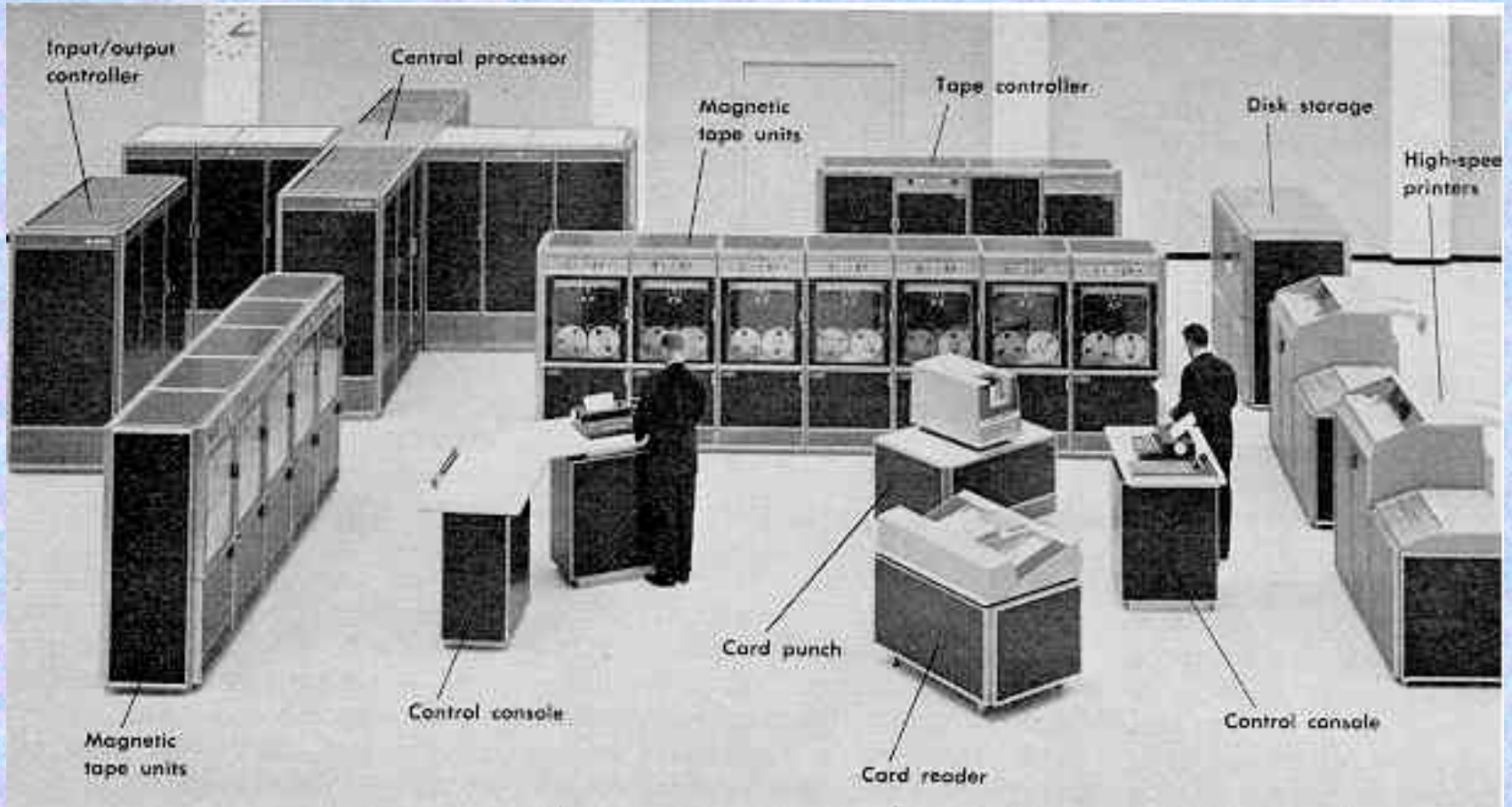
A „nagy” gépek korszaka

Az 1960-70-es évek

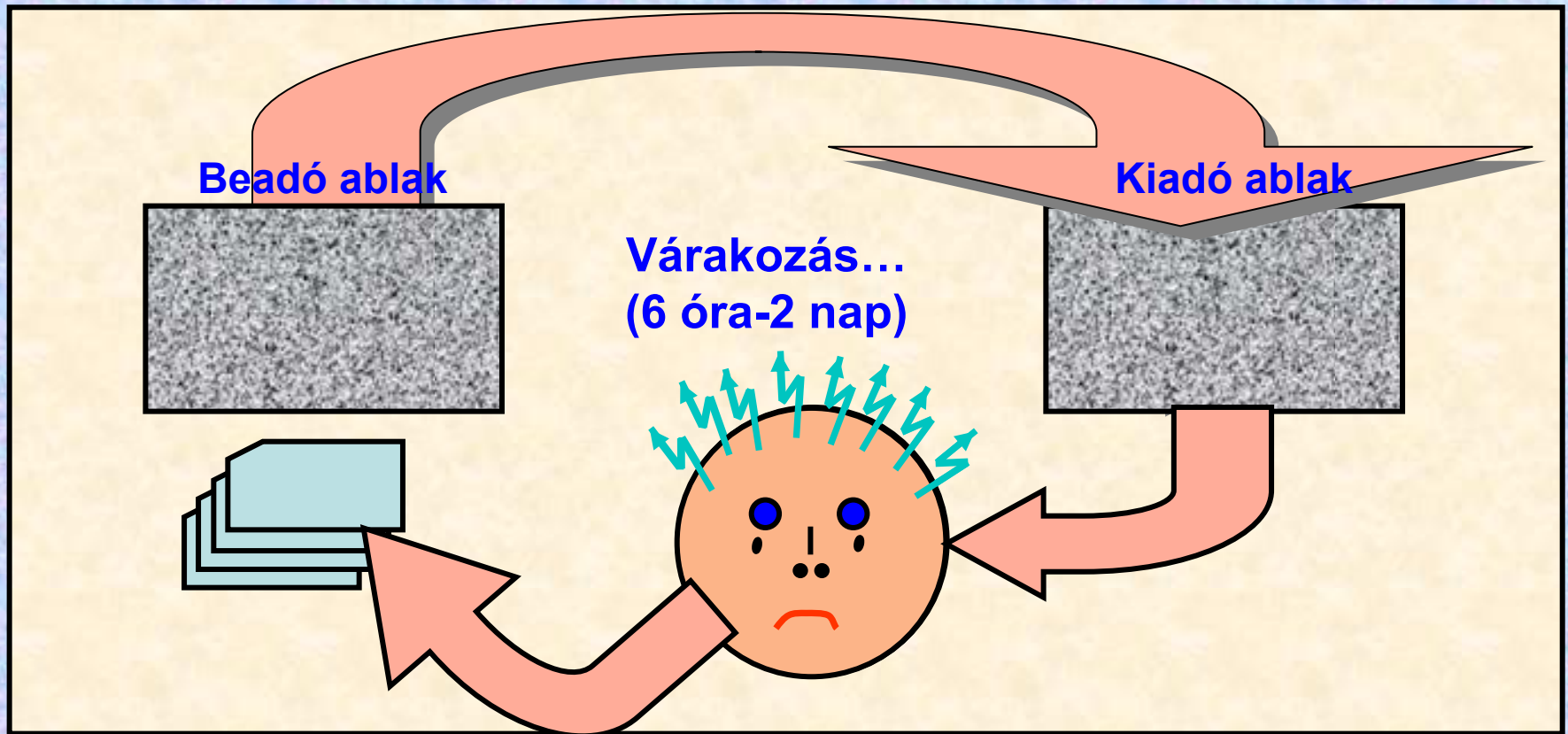
A „mainframe”-ek jellemzői

- Árkategória: néhány százezer – több millió dollár
- Telepítési feltételek:
 - hatalmas gépterem
 - klíma, álpadló, álmennyezet
 - személyzetnek munkaköpeny, -cipő
- Üzem:
 - folyamatos
 - zárt („*closed shop*”=„*kívül tágasabb*”)
 - kötegelt (*batch*) üzemmód
 - munkák fordulási ideje: néhány óra – néhány nap
 - később felhasználói terminálrendszerek

Egy tipikus nyugati gépterem 1967-ből



Batch üzemű számítóközpont



És így tovább...

A magyarországi számítógép „skanzen”

	1967	1968	1969	1970	1971	1972	1973
Típusok száma	23	26	31	40	47	50	54
A gyártó cégek száma	15	15	17	21	21	22	26
Gép összesen	48	65	86	120	161	184	228

A Számítástechnikai Központi Fejlesztési Program (SzKFP)

- 1969 dec.: 6 KGST ország aláírja az **Egységes Számítástechnikai Rendszer (ESZR)** közös fejlesztéséről szóló egyezményt
- 1970: francia licenc alapján a Videoton megkezdi a VT 1010B gyártását
- 1971: a kormány jóváhagyja az SzKFP-t
 - Az ESZR sorozat legkisebb tagjának (**R-10**) fejlesztéséért a Videoton felelős (**gyártás +műszaki kiszolgálás**).
Részt vesz még 7 nagyvállalat és intézmény

- A fő cél az alkalmazás fejlesztés
 - államigazgatási
 - ágazati
 - vállalati szinteken
- A kutatási feladatokat a Számítástechnikai Kutatási Célprogram (SzKCP) részletezi
- Kiemelt feladat az oktatás alap-, közép-, felső- és posztgraduális szinten
- Vállalati számítógép beruházáshoz állami támogatást nem, de preferált hitelkeretet ad

Az ESZR sorozat

- A „skanzen” nem magyar specialitás volt
- Az ESZR program valójában az IBM/360-as, később („*ESZR-II*”) az IBM/370-es gépsorozat lemásolásáról szólt („*reverse engineering*”)
- Az ESZR gépeknél az „R” betű az orosz „*Rjad*” (*sor, sorozat*) szót jelenti
- Az „R” gépek kompatibilis sorozatot alkotnak, az R-10 kivételével
- Az SzKFP jelentősen korlátozta a tőkés számítógép importot (*nemcsak az embargó!*)

Egy „kisebb”
(R30-as)...



...és egy „nagyobb”,
második sorozatú
(R55-ös) ESZR gép



A KFKI

KFKI = Központi Fizikai Kutató Intézet

A KFKI története dióhéjban

- **1950: a Minisztertanács határozata a KFKI létrehozásáról az MTA keretében**
- **1955: a Minisztertanács határozata a kísérleti atomreaktor megépítéséről**
- **1975: a KFKI Kutatóközponttá alakul. Intézetei:**
 - **Atomenergia Kutató Intézet**
 - **Mérés- és Számítástechnikai Kutató Intézet**
 - **Részecske- és Magfizikai Kutató Intézet**
 - **Szilárdtest Kutató Intézet**
 - **Különbféle kiszolgáló részlegek**
- **1990-1992: különválik az alapkutatással és a számítástechnikai vállalkozásokkal foglalkozó rész (*megalakul a KFKI Rt.*)**

A KFKI jellemzői

- **Nagy anyagi és szellemi koncentráció**
(*max. létszám: kb. 2300 fő, a teljes MTA mintegy fele*)
- **A kísérletek jelentős műszerigénye**
 - A kutatórészlegek körül kísérleti eszközöket előállító „barkácsműhelyek” alakulnak ki
- **A mérések kiértékelésének jelentős számítástechnikai igénye**
 - A kutatórészlegek körül (*kezdetben mechanikus gépekkel felszerelt*) számoló részlegek alakulnak ki
- **Az igények központi kielégítésére jött létre a Számítóközpont, valamint a Mérés- és Számítástechnikai K.I.**

Sándory Mihály elve a „barkácsműhelyek” kialakulásának 5 fázisáról:

1. fázis: Barkácsműhelyek alakulnak, szervezeti egységeként külön-külön

Ciklus

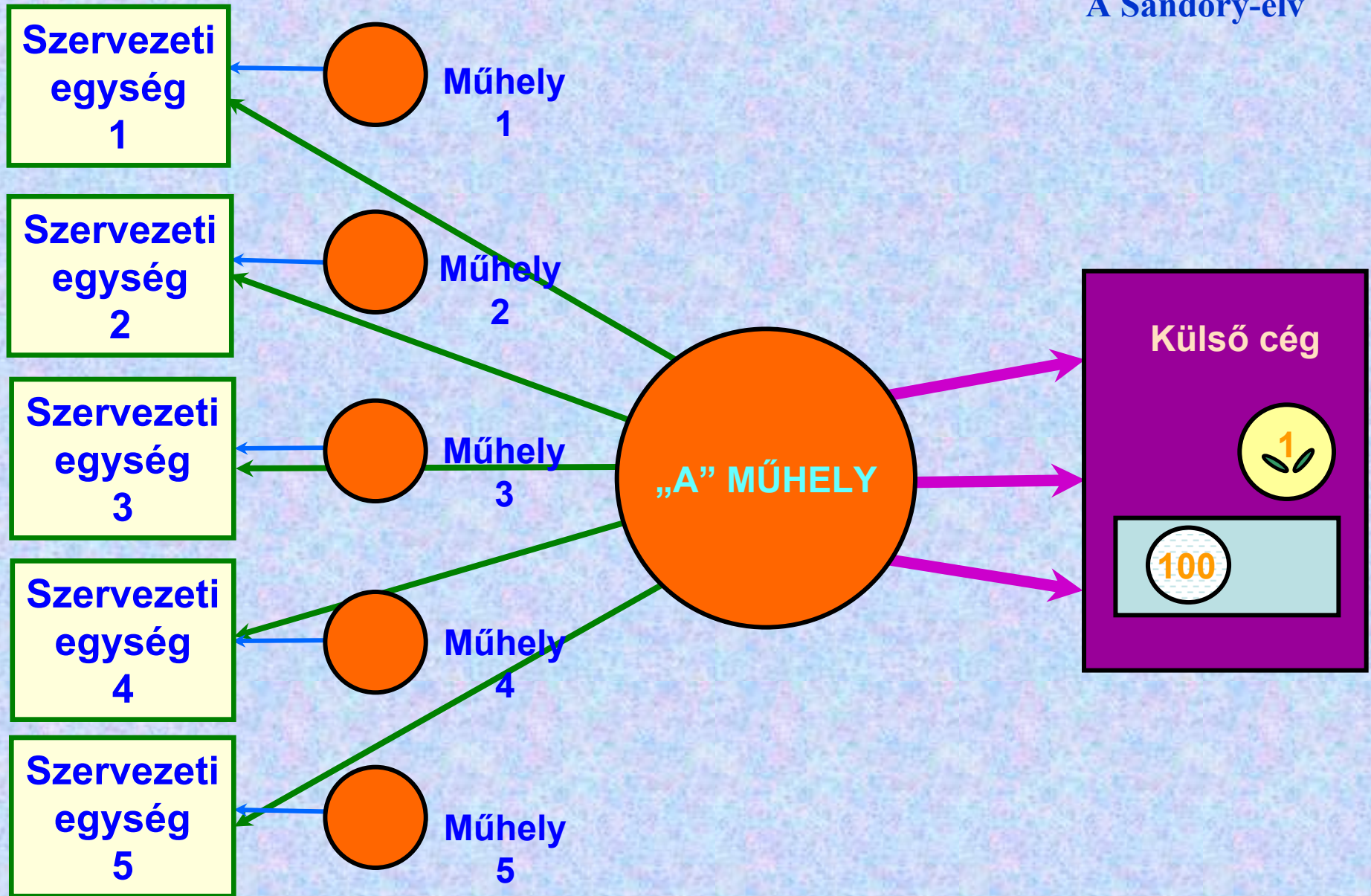
2. fázis: Összevonják őket egyetlen műhellyé

3. fázis: „A” Műhely külső munkákat is vállal

4. fázis: Főleg külső munkákat vállal

5. fázis: Az ilyen módon lefedetlen igények kielégítésére létrejönnek az önálló barkácsműhelyek

Ciklus vége



A KFKI Számítóközpontja

- **1958: a Numerikus Csoport (a későbbi Számítóközpont) megalakulása**
- **1960: az URAL-1 üzembeállítása**
 - szóhossz: 36 bit
 - tároló: mágnesdob
 - műveleti sebesség: 100 műv/sec
 - bemeneti periféria: 35 mm-es, végtelenített perforált film
 - háttértár: 35 mm-es, mágneses bevonatú film
 - csak numerikus (!!) kimenet
- **1964: a KFKI gyakorlatilag kinövi a gépet, külső gépek használata (URAL-2, Gier, Elliott-803)**

Az URAL-I gép tesztelési utasítása 1961-ből

Az URAL számológép ellenőrzése

Ha számolás közben hiba gyanúja merül fel, az alábbi tesztek segítségével ellenőrizhetjük a gép működését:

1. Runge-Kutta feladat

A program és az adatok egyetlen szalagon vannak: a "Runge-Kutta" feliratu dobozban lévő vastagabb szalag.

A pulton: φ blokkolása fel van kapcsolva;
nyomatás: "10-es"
az összes többi kapcsoló le van kapcsolva.

Keszteti indítással beviszünk a programot. Kontroll számolás után megáll és a szumátorba kihozza a kontroll számát:
47 00 13 2552 67.

Indítás: "pusk".

Lefutattunk 10-15 lépést és összehasonlítjuk a program-szalaghoz mellékelte eredményekkel.

Es a teszt-feladat az egész gép működését ellenőrizi /kivéve a mágneszalagot/.

2. Mágnesdob teszt program

A mágnesdob valamennyi memóriarekesztét ellenőrizi.

As "AY, H15 tesztet" feliratu dobozban lévő vékonyabb szalagon van a teszt program és az adatok.

A pulton: valamennyi vezérlés átadó kulcs /1,2,...7/ felkapcsolva;
 φ blokkolása felkapcsolva.

Keszteti indítás.

A beolvasás után kontroll számolásni kezd a gép és ha jó eredményt kap, akkor rátér a dobot ellenőrző program végrehajtására. Mintegy 3 percig tart a program egyszeri lefutása. Ha hiba nem jelentkezik, előlről kezd a program végrehajtását.

Hiba esetén a gép megáll. Újraindításnál annál a programrésznél kezd újra a számolást, ahol a hiba fellépett.

3. Zóna keresés /rövid változat/

A program a "Runge-Kutta" dobozban lévő vékonyabb szalagon van.

A pulton minden kulcs lekapcsolva.

Keszteti indítás.

A beolvasás után a 0005 címél megáll.

- 2 -

Indítás "pusk"-kal.

Készteti keresi a 177. zónát, beolvassa, egy pillanatra megáll, majd újra keresi a 177. zónát, stb. Es a folyamat mintegy 2 sec-es periódussal ismétlődik.

As ismétlődés egyenletességéből megállapíthatjuk, hogy először megtalálja-e mindig a zónát.

4. Kinyomató teszt

A teszt program a "NI" teszt feliratu dobozban lévő vastagabb szalagon van.

Bevitel keszteti indítással.

A beolvasás és kontroll számolás után megáll.

A kontroll szuma az 5672. rekeszben van:
214117 320261.

"10" nyomatás esetén: a 2. és 3. vezérlés átadó kulcs felkapcsolva. Indítás 1565 címél. A gép egyszer végrehajtja az ellenőrző programot és megáll. A kinyomtatott számokat összehasonlítjuk a program szalag mellett található mintával.

"8" nyomatás esetén: az 1. 2. és 3. vezérlés átadó kulcs felkapcsolva. Indítás 1631. címél. A gép egyszer végrehajtja az ellenőrző programot és megáll. A kinyomtatott számokat összehasonlítjuk a program szalag mellett található mintával.

- 1966-67: **Üzembeállítják az ICT 1905-ös gépet**
(*rövid ideig ez volt az ország legnagyobb teljesítményű gépe*)
 - szóhossz: 24 bit
 - tároló: 32 Kszó
 - 8 csatornás lyukszalag bemenet
 - alfanumerikus kimenet (*sornyomtató*)
 - 6 mágnesszalagos egység (*7 csatornás*)
 - 4 program páruzamos futtatására alkalmas op. rendszer (*executiv*)
 - ALGOL, FORTRAN stb. fordítók
 - a batch üzemmódhoz külön programot kellett fejleszteni

- **1973-1988: IBM kompatibilis gépek + BASF**
(1973: R20, 1977: R40, 1986: R45, 1988: BASF 7/61)
- **1978: a merev batch rendszert kiváltó interaktív CEDRUS rendszer üzembeállítása**
(KFKI fejlesztés, TPA-70 front-end)
- **1992: a KFKI átszervezése után az RMKI keretében Hálózati Központként működik tovább**

A Számítóközpont a rendszerváltást megelőző időszakban, a lehetőségekhez képest korszerű eszközökkel tudott működni, de jelentősége a személyi számítógépek elterjedése után csökkent.

A KFKI minigépei, a TPA családok

TPA=Tárolt **P**rogramú **A**datfeldolgozó gép

A minigépek jellemzői

- **Átmenet** (*méretben és időben*) a nagy-, és a személyi számítógépek között
- **Árkatégória:** már kezdettől 1 nagyságrenddel a nagygépek alatt
- **Telepítési feltételek:** nem igényelnek speciális géptermet, helyigényük jóval kisebb
- **Üzem:** kezdetben egy-, később több felhasználós, felhasználóbarát, interaktív
- **Felhasználási terület:** mérési adatgyűjtés, CAD/CAM, folyamatvezérlés, kisebb adatfeldolgozás, front-end processzor stb.
- **Méreteik** az integráltság növekedtével egyre csökkennek

A KFKI minigépeinek fejlesztési korszakai

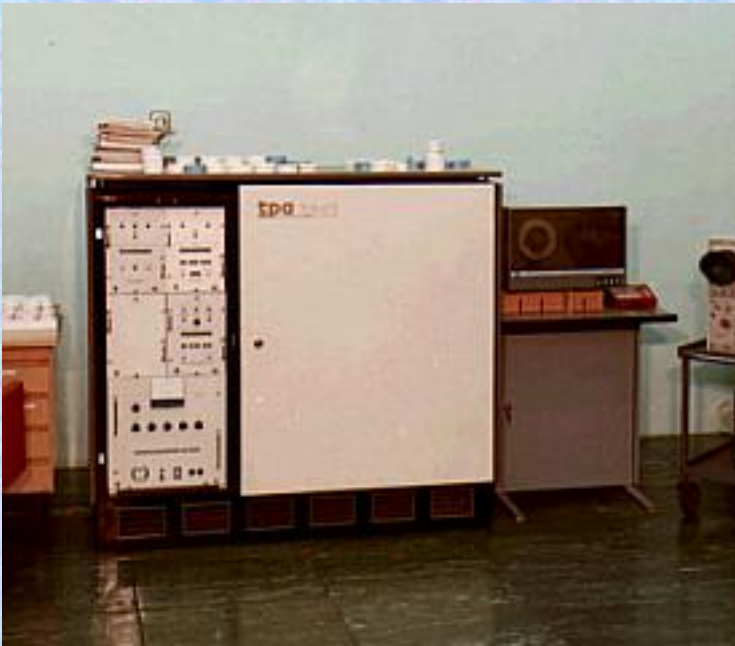
- **12 bites gépek**
(*TPA-1001 család, PDP-8 kompatibilis*)
- **16 bites gépek**
 - TPA 70/25
 - TPA-11 család (*PDP-11 kompatibilis*)
- **32 bites gépek** (*VAX kompatibilis*)

VAX=Virtual Address eXtension

A 12 bites TPA gépcsalád

- 1955-: a kísérleti atomreaktor megépülése után megindul a mérőműszerek (*analizátorok*) építése
- 1960, 1966: az Ural és az ICT gép installálása, üzemeltetése révén további tapasztalatok
- 1965: A DEC (*Digital Equipment Corporation*) cég kihozza a PDP-8 miniszámítógépet (*12 bites szóhossz, egyszintű interrupt rendszer, egyszerű utasításkészlet*)
- 1968: A KFKI elkészíti a PDP-8-cal kompatibilis gépet. Ekkor még TPA (*Tárolt Programú Analizátor*)
- 1969-84: különböző integrált áramkörös változatok

A 12 bites gépcsalád



TPA-1001
az első modell, 1968



TPA Quadro
az utolsó modell, 1984

A 12 bites TPA gépek software ellátottsága

- kezdetben saját fejlesztések, részben szimulátoron
- később az eredeti **DEC** softwarek adaptációja
- az utolsó (*Quadro*) modellen a **CP/M** operációs rendszer is futott

A software kompatibilitásának volt köszönhető a TPA gépek nagy népszerűsége!

A TPA-70

- 1970: a KFKI byte szervezésű, saját architektúrájú gép fejlesztésébe kezd
- 1975: elkészül a gyártható változat
 - *hasonlít* a DEC PDP-11 gépére, de nem sért szabadalmakat
 - nincs adaptálható software rendszer!
 - DEC: kb. 150 fő,
 - KFKI: kb 15 fő alapsoftware-fejlesztő
- 1977: a KFKI eladja a gyártási jogot a VILATI-nak
- Alkalmazások:
 - grafikus display vezérlés (*SzTAKI display*)
 - egyedi alkalmazások
 - front-end processzor (*KFKI, CDC ajánlat!*)



TPA-70 asztali példány

A TPA-11 gépcsalád

- 1975: a KFKI a DEC-vonal (PDP-11 gépek) folytatása mellett dönt
- 1979: elkészül az első PDP-11 kompatibilis gép
- Jellemzők:
 - többféle operációs rendszer
(1 és több felhasználós)
 - BASIC, FORTRAN, COBOL, C, PASCAL, stb.
 - számtalan alkalmazási programcsomag
 - MSZR kompatibilis
(a perifériális egységek szempontjából fontos!)

Az MSZR az ESZR minigépes megfelelője volt



Négy TPA-11-es gép egy sorban...

A 32 bites TPA gépcsalád

- **1983: a KFKI – követve a DEC vonalat – elkezdte a VAX kompatibilis gépek fejlesztését**
- **1987: elkészül a TPA/580**
- **Jellemzői:**
 - VMS operációs rendszer
 - teljes software kompatibilitás
 - „megamini” (*nagy gép, mini kivitelben*)

A „minigép”, mint kategória, napjainkra tartalmatlanná vált!

A TPA-történet vége

- **1990: a rendszerváltás utáni „kiegyezés”:**
 - a KFKI abbahagyja a DEC-klónok gyártását, a DEC pedig „elfelejti” a múltat
 - a KFKI, a DEC és a Számalk közös vállalatot alapít a piaci együttműködés megkönnyítésére
- **1992: a piaci tevékenységet folytató rész (KFKI RT) kiválik a kutatóintézetből**
- **Végül:**
 - a DEC kivásárolja partnerei részét a közös vállalatból
 - a COMPAQ felvásárolja a DEC-et
 - a Hewlett-Packard (**HP**) felvásárolja a COMPAQ-ot

A TPA-korszak jellemzői

- **Saját döntés volt, nem része az országos programoknak (SzKCP)**
- **TPA-k beszerzésére általában nem járt állami támogatás**
- **A periféria-, és alkatrészbázis különbözősége miatt, a létező modellek utánzása is jelentős fejlesztéseket igényelt**
- **A rendszerintegrálás volt a cél, nem a tömeggyártás”**
- **A TPA típusokból a '70-'90 –es években mintegy 1600 darab készült**

Fejezetek



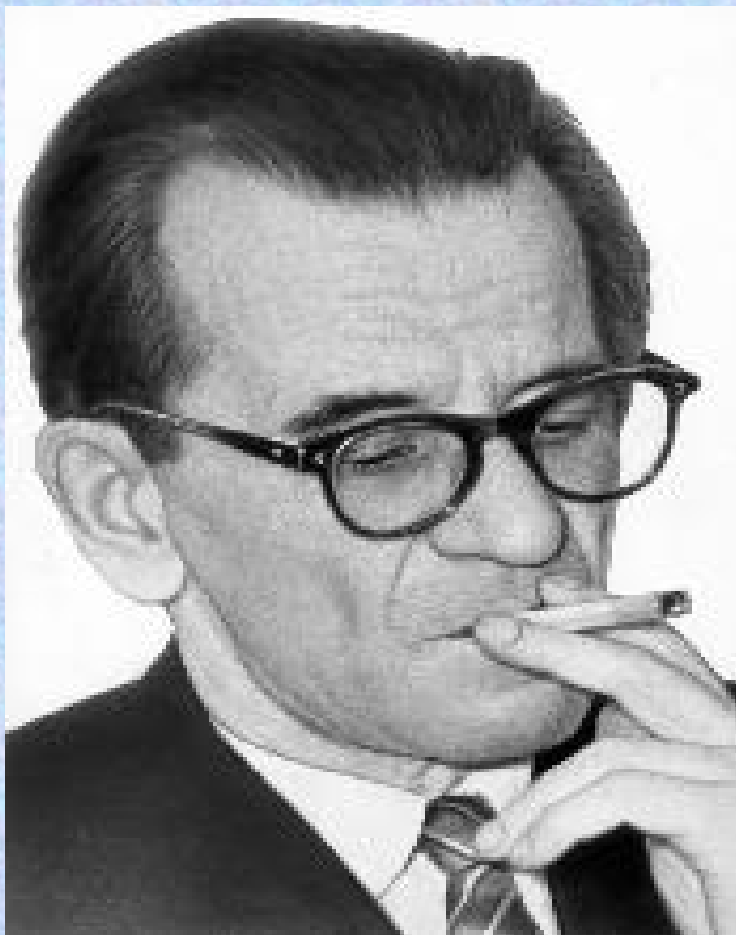
az informatika történetéből

15. fejezet

A számítástechnika története a Szovjetunióban

A MESZM, BESZM és URAL gépek

Szergej Alekszejevics Lebegyev (1902-1974)



- 1902 nov. 2.-án született Nyizsnij Novgorodban
- 1939: megszerzi a kandidátusi fokozatot
- 1946-51: a kijevi Elektromechanikai Intézet igazgatója
- 1950: elkészül a MESZM, a SzU első számítógépe
- 1953: a moszkvai ITMiVT igazgatója. A későbbi BESZM gépek kifejlesztésének irányítója

Az első szovjet számítógép: a MESZM



- **Feltételek: nem a legjobbak!**
 - Általános háború utáni helyzet
 - Sztálini ideológia kibernetika-ellenessége
 - Számítástechnikai publikációk hiánya
 - Nyelvi problémák
- **Következmény: sajátos, belső fejlődés**

Az első szovjet számítógép: a MESZM



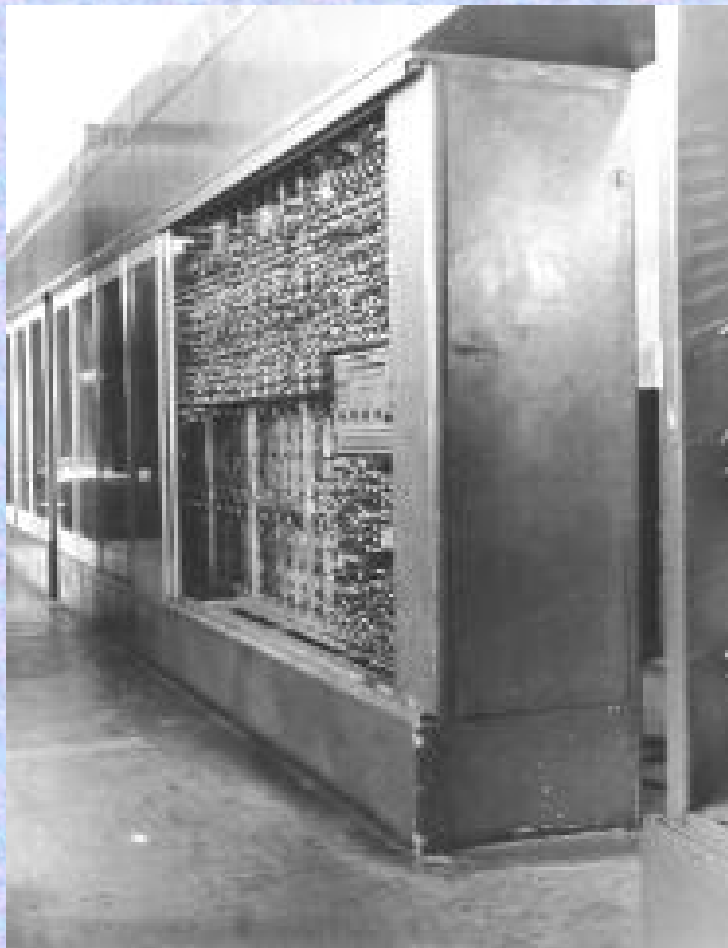
- **MESZM = Malaja Elektronnaja Szcsotnaja Masina
(kicsiny, elektronikus számológép)**
- **Kísérleti („gyakorló”) konstrukció**
- **Cél: minél előbb működőképes modell előállítása**
- **Címzési mód: háromcímű**
- **Szóhossz: 21 bit**

Az első szovjet számítógép: a MESZM



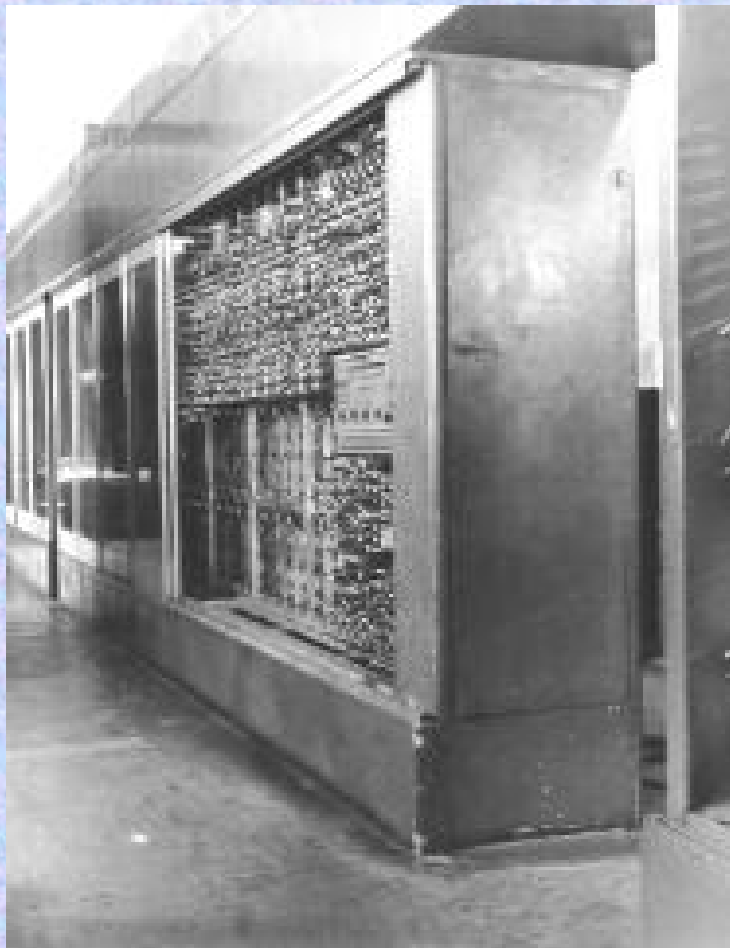
- Tárolható adatok száma: 31
- Tárolható utasítások száma: 63
- Számábrázolás: fixpontos
- Elektroncsövek száma: kb. 3800
- Input periféria: nincs (*kézi input*)
- Tároló: flip-flop-okból felépítve (*ld. ENIAC*)

A BESZM gépek



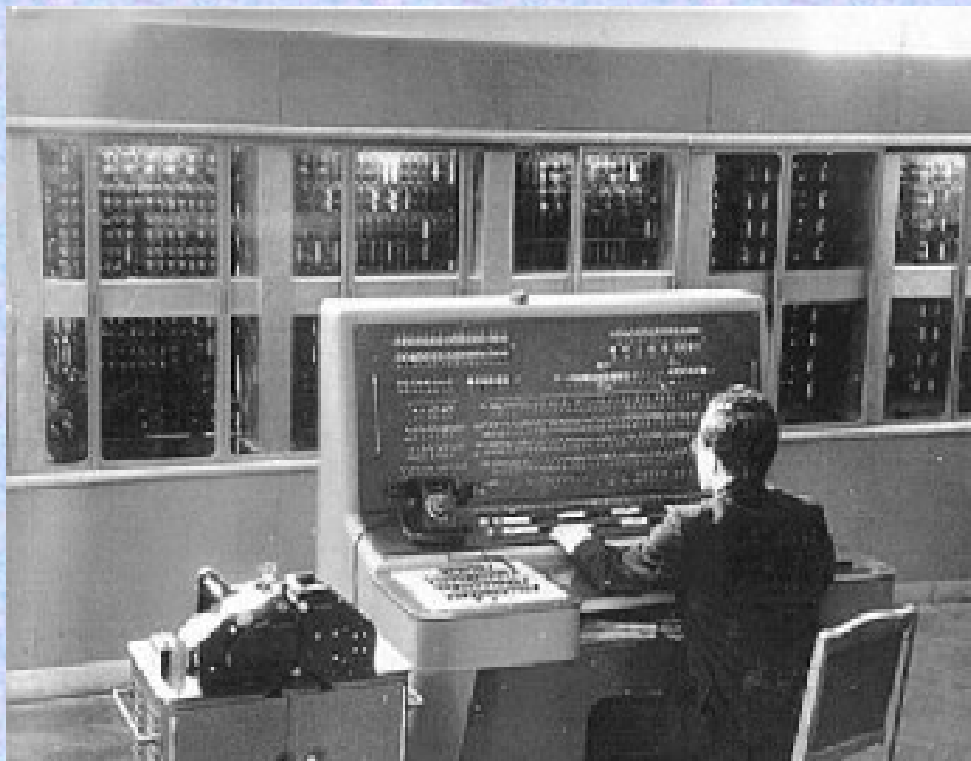
- 1948: Megalakul az ITMiVT
- 1950: Elkezdődik az első „igazi” gép tervezése (későbbi nevén *BESZM-1*)
- 1952: Első tesztfutások
- 1956: Továbbfejlesztett konstrukcióval (*BESZM-2*) sorozatgyártás

A BESZM gépek



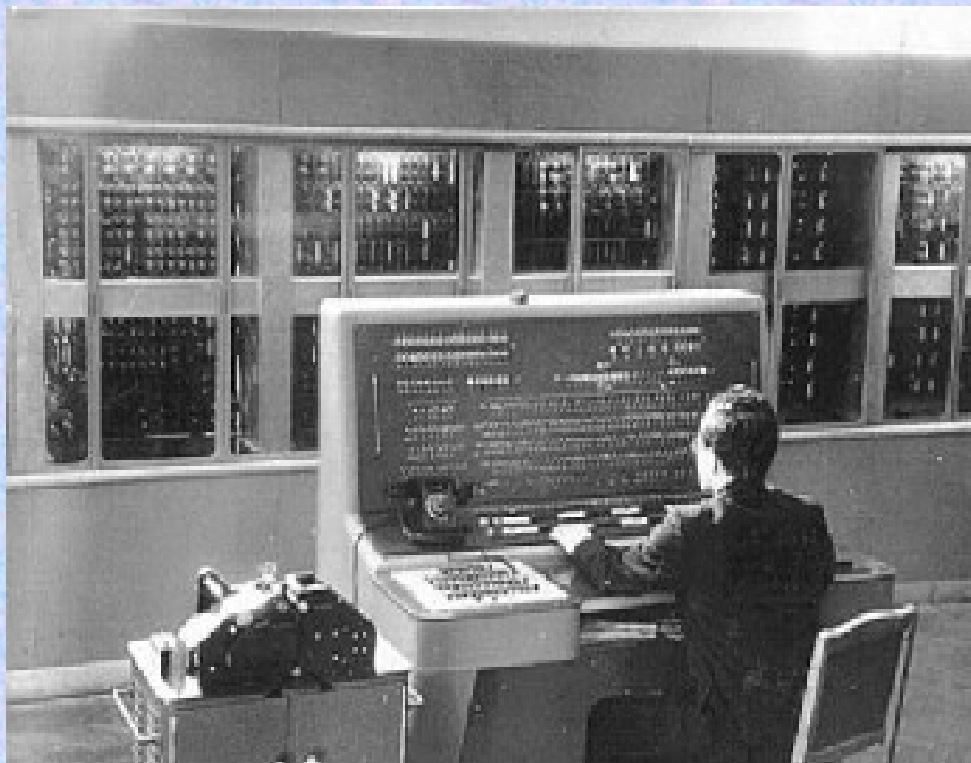
- „Éles” konstrukció
- Címzési mód: háromcímű
- Szóhossz: 39 bit
- Számábrázolás: lebegőpontos
- Tároló: késleltetett művonalas
(melyet 1956-ban elektrosztatikusra, majd 1960-ban ferritre cseréltek)
- Tárolókapacitás: 1K szó
- Elektroncsövek száma: kb. 4000
- Input periféria: lyukszalag olvasó
- Egyéb: egyedi példány
(teljesen „házilagosan” készült)

A BESZM-2



- Az első sorozatgyártású szovjet számítógép
- Címzés, szóhossz, számábrázolás, input: mint elődje
- Tároló: ferrit
(*max. 2047 szó*)
- Elektroncsövek száma: ~4-5000 félvezető dióda
- Output periféria: nyomtató
(*csak numerikus adatok nyomtathatók*)

A BESZM-2



- **Háttértárak:**

- Mágnesdob:
min. 5400 szó

- Mágnesszalag:
min. 40 000
szó/egység
(*max. 4 db.*)

- **Op. rendszer:** nincs

- **Gyártásban:** 1958-62
(*néhány tucat darab*)

A BESZM-6



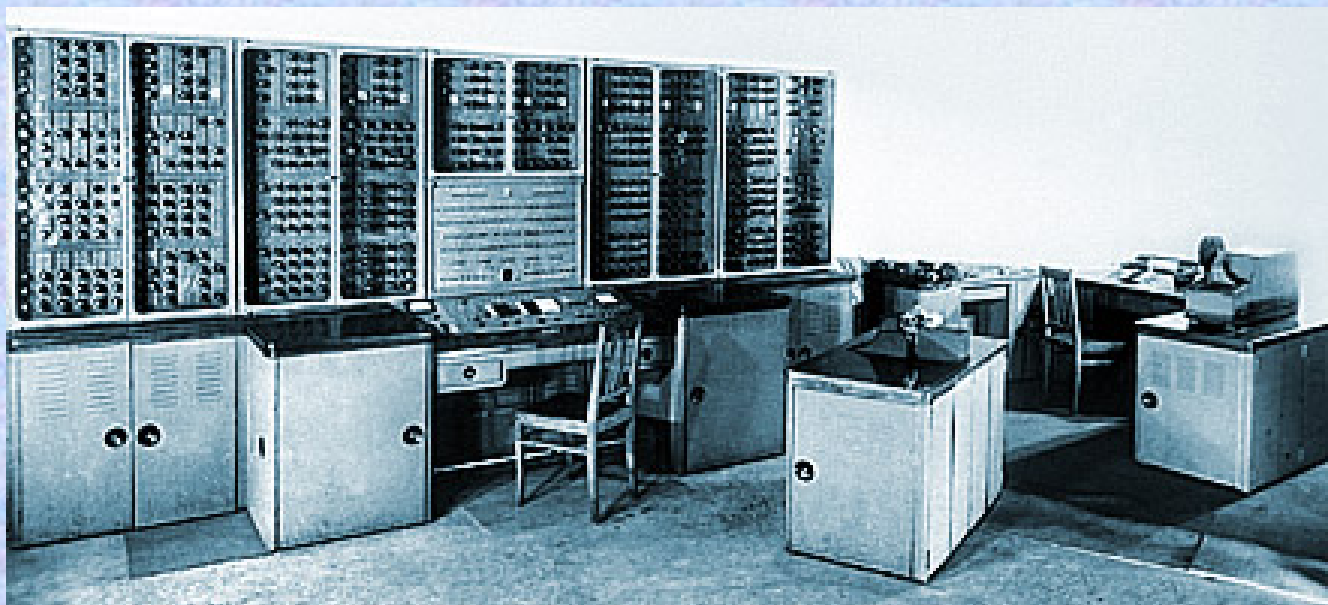
- **Az első multiprogramozásos op. rendszerrel rendelkező, sorozatban gyártott szovjet számítógép**
- **Architektúra: diszkrét tranzistoros áramkörök**
- **Utasításrendszer: egycímű (2 utasítás/szó)**
- **Szóhossz: 48 bit**
- **Számábrázolás: lebegőpontos**

A BESZM-6



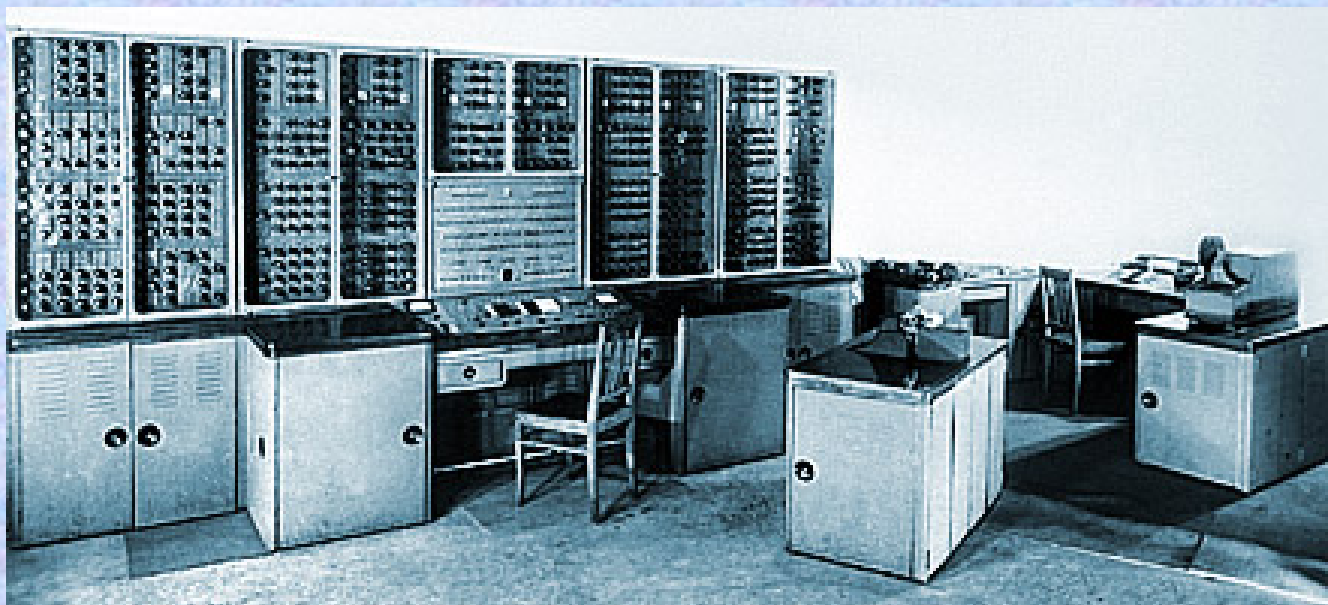
- **Operatív tároló: ferrit**
(32-128K szó)
- **Műveleti sebesség: 1 millió**
műv/sec, 48 bites
lebegőpontos számokon
- **Háttértárak: mágnesdob,**
-lemez, -szalag
- **Fejlesztés kezdete: 1965**
- **Gyártásban: 1967-1987**
(355 db)

Az URAL-1



- Szóhossz: 36 bit (2 utasítás/szó)
- Utasításrendszer: egycímű
- Számábrázolás: fixpontos
- Tároló: mágnesdob (1024 szó)
- Input: végtelenített perforált film
- Output: numerikus nyomtató

Az URAL-1



- **Műveleti sebesség: 100 műv./sec**
- **Háttértár: feketére hívott végtelenített mágnesfilm**
- **Gyártásban: Penza, 1956-1961 (183 db)**
- **Magyarországon installált példányok száma: 2 db**

Az URAL-2



- Szóhossz: 40 bit
- Utasításrendszer: egycímű
- Számábrázolás: fix+lebegőpontos
- Tároló: ferrit (2043 szó)
- Input, Output: mint az URAL-1

Az URAL-2



- **Műveleti sebesség: 5-6000 műv./sec**
- **Háttértár: mágnesdob (*max. 8 db/8192 szó*), mágnesszalag**
- **Gyártásban: Penza, 1959-64 (*139 db*)**
- **Magyarországon installált példányok száma: 2 db**

MEGJEGYZÉSEK:

A Szovjetunióban ebben az időben párhuzamosan számos számítógép fejlesztési program futott
(*BESZM, Ural, Minszk, Razdan...*)

A kifejlesztett gépek gyakran még egy sorozaton belül sem voltak kompatibilisek
(pl. *Ural-1 és Ural-2, Minszk-1 és Minszk-2, stb.*)

A típusválaszték sokszínűségét később korlátozta az *ESZR*, majd az *MSZR* program, amely az amerikai *IBM/360-370* sorozat, ill. a *PDP-11* sorozat gépeinek utánzását tűzte ki célul.

A korai szovjet számítógépek néhány hátrányos tulajdonsága

- **Csak numerikus be- és kimenet**

(Ez konzerválta a gépi kódú programozást, és akadályozta a magasszintű nyelvek elterjedését)

- **A háromcímű utasításrendszerek dominálnak**

(Sok bit kihasználatlanul marad. Nyugati gépeken inkább két egycímű utasítást tettek egy szóba)

- **Nem standard perifériák**

(Perforált film, 14 csatornás, 1" széles mágnesszalag, stb. Kölcsönös inkompatibilitás!)

Programozási eszközök és nyelvek

- Kb. 1960: az első "integrált" sw rendszer az M-20-ra, az ИС (Интерпретирующая Система = értelmező rendszer)
- Lényege: Külső tárolón (*dob*) elhelyezett szubrutinkönyvtár, melynek rutinjai végrehajtáskor szükség szerint töltődnek a tár adott munkaterületére