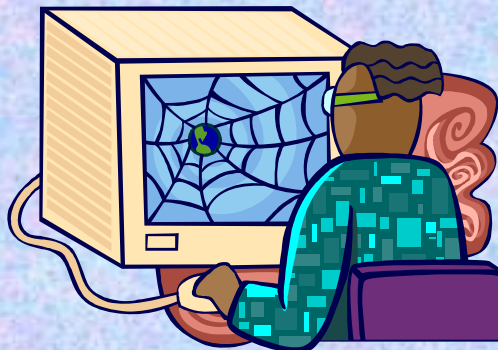


Fejezetek



az Informatika történetéből

**Készítette: Lőcs Gyula**

**Levélcím: 1121 Bp., Eötvös út 47- 53. „F” ép. 10.**

**Telefon: 06-30-919-6009; 06-1-395-95-28**

**E-mail: [gy\\_locs@invitel.hu](mailto:gy_locs@invitel.hu)**

# 1. fejezet

## Babilóniai matematika



Matematikai feladatok és „programok”

## Négyezer éves számtanpéldák — hatvanas számrendszerben

*Babilónia, i.e. 1800-1600, az ú.n. Hamurabbi dinasztia kora*

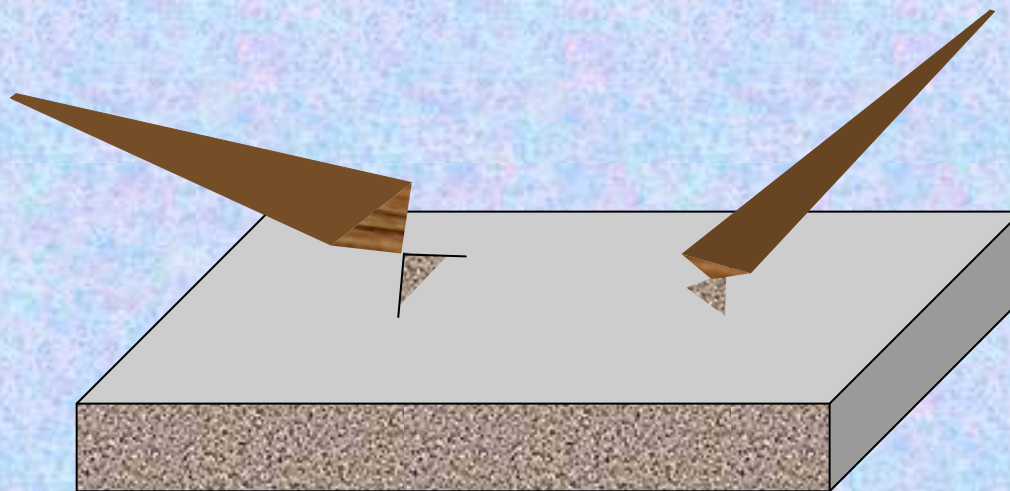
- **Számírás:** ékírásos számjelek
- **Számrendszer:** hatvanas (*sexagézimális*)
- **Forrás:** ékírásos agyagtáblák (*British Museum, Louvre...*)  
*legrégebbi: i.e. kb. 2250*

**Számjelek (az ékírás után szabadon):**

$\nabla = 1$	 = 4	$\nabla \nabla = 11$	$\nabla \nabla \nabla = 30$	$\nabla \nabla = 70$
$\nabla \nabla = 2$	$\cdot$	$\nabla \nabla \nabla = 12$	$\cdot$	$\nabla \nabla \nabla = 80$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	 = 9	$\cdot$	$\cdot$	$\cdot$
$\nabla \nabla \nabla = 3$	$\nabla = 10$	$\nabla \nabla = 20$	$\nabla = 60$	$\nabla \nabla = 120$

**stb.**

## Az ékírás „technikája”



**Jelölésünk:**  $(a_1, a_2, \dots, a_k) = a_1 * 60^i + a_2 * 60^{i-1} + \dots + a_k * 60^{i-k+1}$

*ahol az „i” értékét „fejből” kell tudni*  
*(„kvázi-lebegőpontos” számábrázolás)*

Pl:  $(1,40) = 1 * 60^2 + 40 * 60^1 = 3600 + 2400 = 6000$

vagy:  $= 1 * 60^1 + 40 * 60^0 = 60 + 40 = 100$

vagy:  $= 1 * 60^0 + 40 * 60^{-1} = 1 + 40/60 = 1\frac{2}{3}$

**stb.**

**tehát „hatvanadospont” nincs, és kezdetben zérus sincs!**

- **Szorzás:** nem tudjuk, hogyan végezték. Az „egyszeregy”-et nem ismerték, szorzótáblázataik voltak.

- **Például:**

$$(2,15)*(3,20)=(2*60^1+15*60^0)*(3*60^2+20*60^1)=7*60^3+30*60^2 \\ =(7,30)$$

$$(2,15)=135 \sim (3,20)=12000 \sim 135*12000=1620000$$

*(önkényesen!)*

- **Megfelelő algebrai ismeretek híján mindent „algorithmikusan” írtak le.**



# Reciprok táblázatok

*szerepük: az osztás megkönnyítése*

<u>Szám:</u>		<u>Reciprok:</u>		
<u>Dec.</u>	<u>Sex*</u>	<u>Sex*</u>	<u>Valódi tört</u>	<u>Megj.</u>
1	(1)	(1)	$1/1$	
2	(2)	(30)	$1/2$	
3	(3)	(20)	$1/3$	
4	(4)	(15)	$1/4$	
64	(1,4)	(56,15)	$1/64$	$60^3/64=3375$
80	(1,20)	(45)	$1/80$	$60^2/80=45$

**\*Sex = sexagézimális**

## Reciprok táblázat készítése

*ha  $X$  reciproka  $Y$ , akkor:*

**1 reciproka 1**

**2 reciproka 30**

**3 reciproka 20**

**5 reciproka 12**

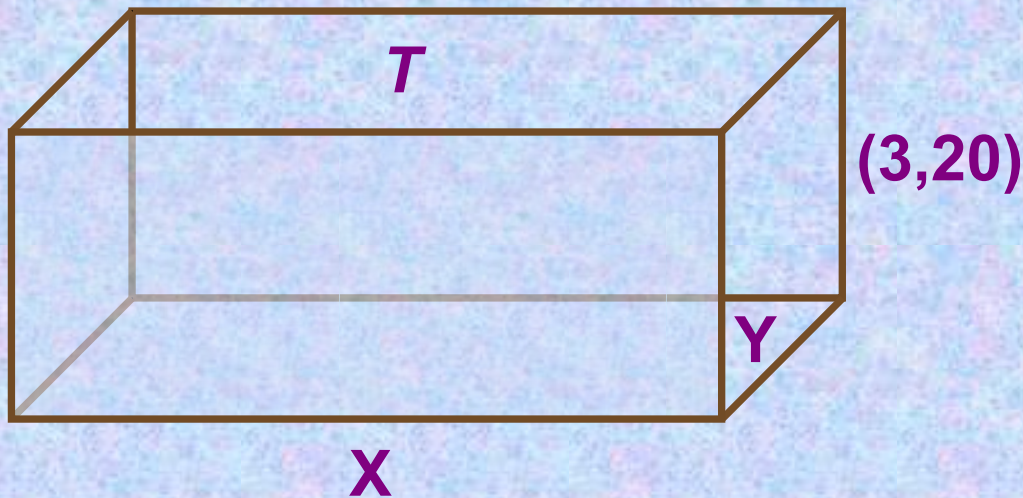
**2X reciproka 30Y**

**3X reciproka 20Y**

**5X reciproka 12Y**

**és így tovább...**

**Egy matematikai példa:** Egy téglatest alakú ciszterna magassága  $(3,20)$ , térfogata  $(27,46,40)$ , hosszúsága pedig  $(50)$ -nel nagyobb, mint a szélessége. Mennyi a hosszúság és a szélesség?



$$V = (27,46,40)$$

$$T = 1/(3,20) * (27,46,40)$$

$$XY = T = (8,20)$$

$$X - Y = (50)$$

**Az előző és a jelen jegyzetoldalon részletezett  
levezetés szerint az**

$$XY=(8,20)$$

$$X-Y= (50)$$

**egyenletrendszer megoldása:**

$$X = \frac{(50)}{2} + \sqrt{\left(\frac{(50)}{2}\right)^2 + (8,20)}$$

$$Y = \frac{(50)}{2} - \sqrt{\left(\frac{(50)}{2}\right)^2 + (8,20)}$$

## Az ókori szöveg *(legalább kétszeres fordításban)*:

Egy ciszternát ástak.

Magassága (3,20) és térfogata (27,46,40).

A hosszúsága (50)-nel nagyobb, mint a szélessége.

Vedd a magasságnak, (3,20)-nak a reciprokát, és (18)-at kapsz.

Szorozd meg ezt a térfogattal, (27,46,40)-nel, és (8,20)-at kapsz.

Vedd az (50) felét, emeld négyzetre és (10,25)-öt kapsz.

Ennek a négyzetgyöke (2,55).

Erről készíts két másolatot (!!), egyet az összeadás és egyet a kivonás számára.

Azt találod, hogy (3,20) a hosszúság és (2,30) a szélesség.

Íme az eljárás.

## Néhány további ismeret a babilóniaiak tárházából

- **Kamatos kamatszámítás**  
(*évenkénti és többévenkénti tőkésítéssel*)
- **Lineáris interpoláció**
- **Négyzetszámok összegének ismerete**
- **2 hatványok összegének ismerete**
- **Szögfüggvények**
- **Algebrai formalizmus híján gondolkodásuk sokkal „algoritmikusabb” volt, mint a későbbieké.**  
**A zérust jóval később fedezték fel (~ i.e. 300)**

# 2. fejezet

## A mechanikus számológépek kora

Schickard, Pascal, Leibniz, Odhner

**Az első (ókortól kezdve használatos)  
mechanikus számolóeszköz: az abakusz**



**a Római Birodalomból...**



**...és Japánból**



## Wilhelm Schickard (1592-1623)

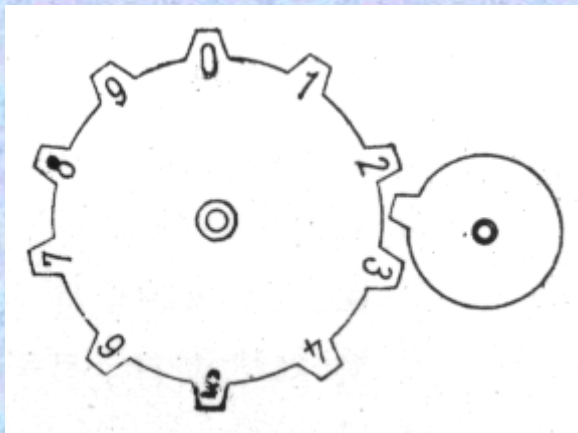


- **1592**-ben született a németországi Herrenbergben
- **1623**-ban készítette el a gépét, amely azonban egy tűzvészben megsemmisült *(a jelenleg múzeumban látható példány az eredeti dokumentációk alapján készült másolat)*
- **1635**-ben halt meg, Tübingen városában, pestisben

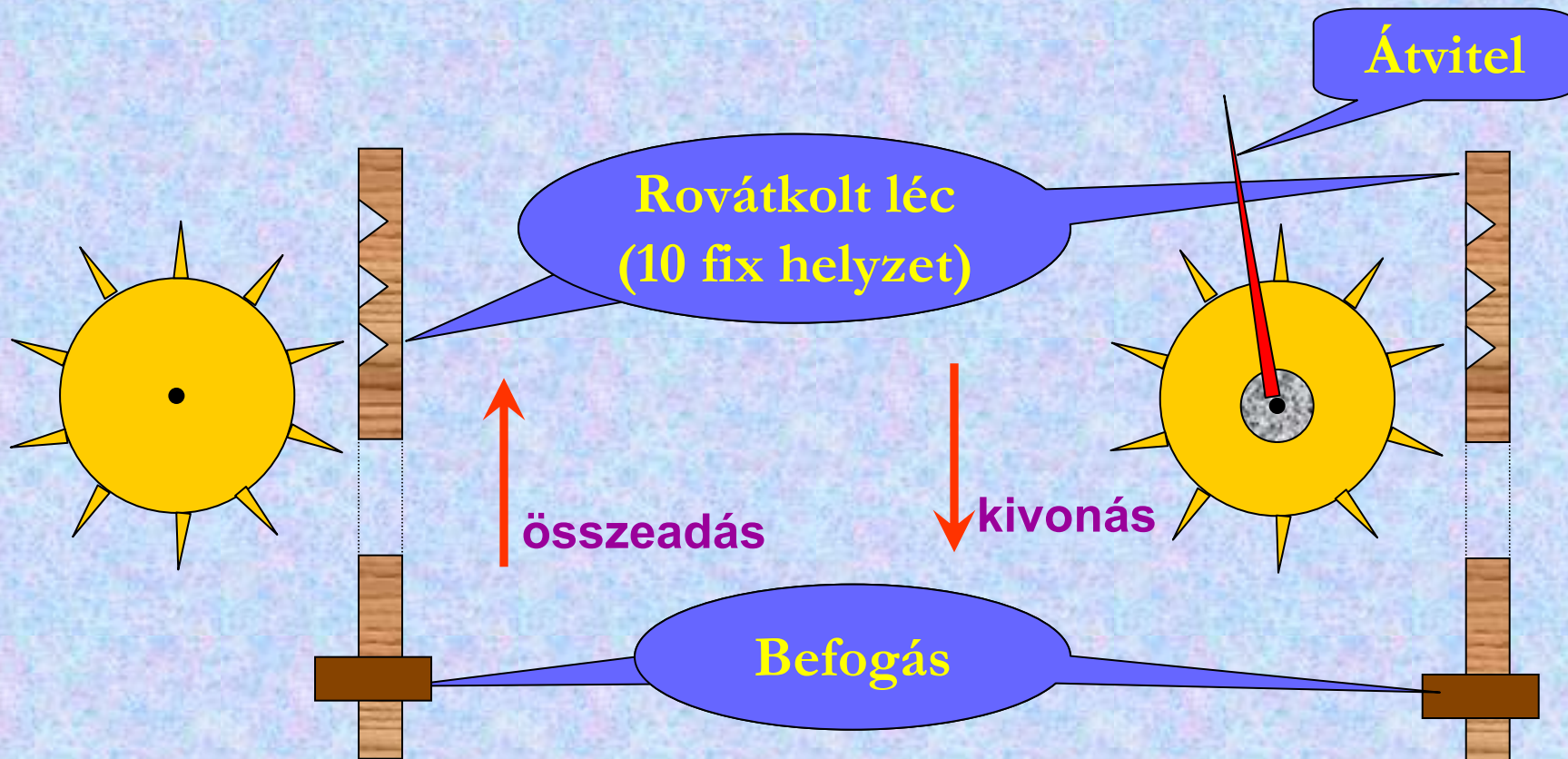
# Schickard számológépe



**A 10-es átvitel elve: fordulatszámoló**



## Alapja: tízállapotú fogaskerék



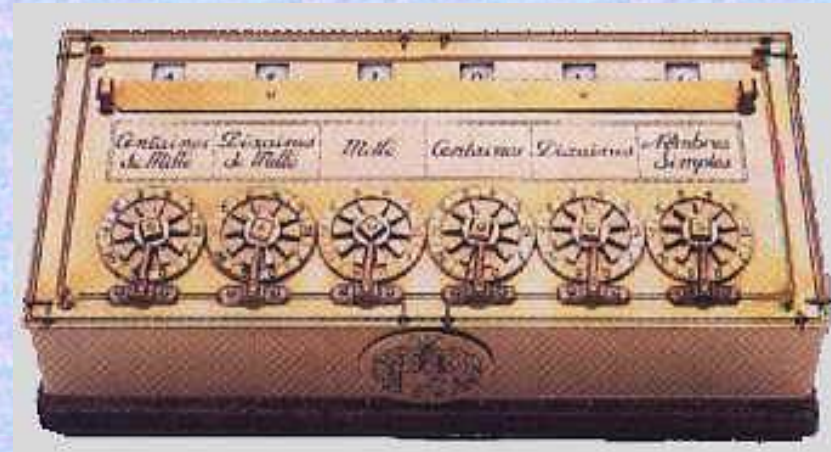
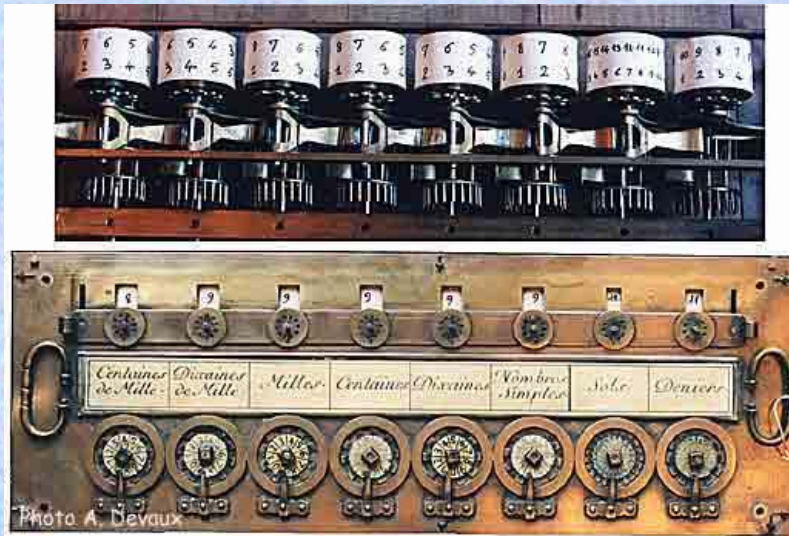
**Átvitel:** 1 fogú fogaskerék, amely meghajtja a szomszédos számkereket a  $9 \rightarrow 0$  vagy  $0 \rightarrow 9$  átmenetkor

## Blaise Pascal (1623-1662)



- 1623-ban született Párizsban
- 1642-ben készítette el a gépét, amellyel édesapja munkáját kívánta segíteni
- 1662-ben halt meg

# Pascal számológépe: a Pascaline



**Shickard és Pascal gépein a számkerekek külön tengelyeken vannak elhelyezve**

## Gottfried Wilhelm von Leibniz (1646-1716)



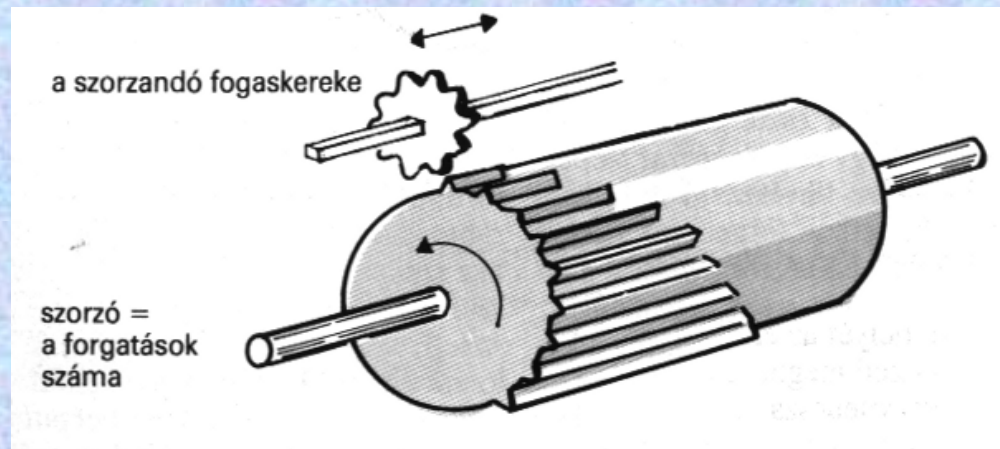
- 1646-ban született Lipcsében
- 1670-es években alkotta meg a gépét, melyben Pascal gépét tökéletesítette, a bordás tengely megalkotásával
- 1716-ban hunyt el Wolfenbüttelben

## Leibniz számológépe



**Nemcsak összeadni és kivonni, hanem szorozni és osztani is tud!**

# Leibniz számológépének „lelke”: a Leibniz-kerék (*bordástengely, különböző hosszúságú bordákkal*)



**A kis kerék, helyzetétől függően, különböző számú bordával érintkezik**

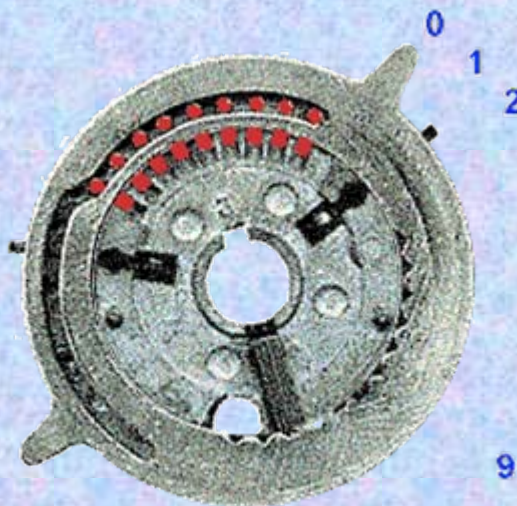


## Willgott Teophil Odhner (1845-1903)



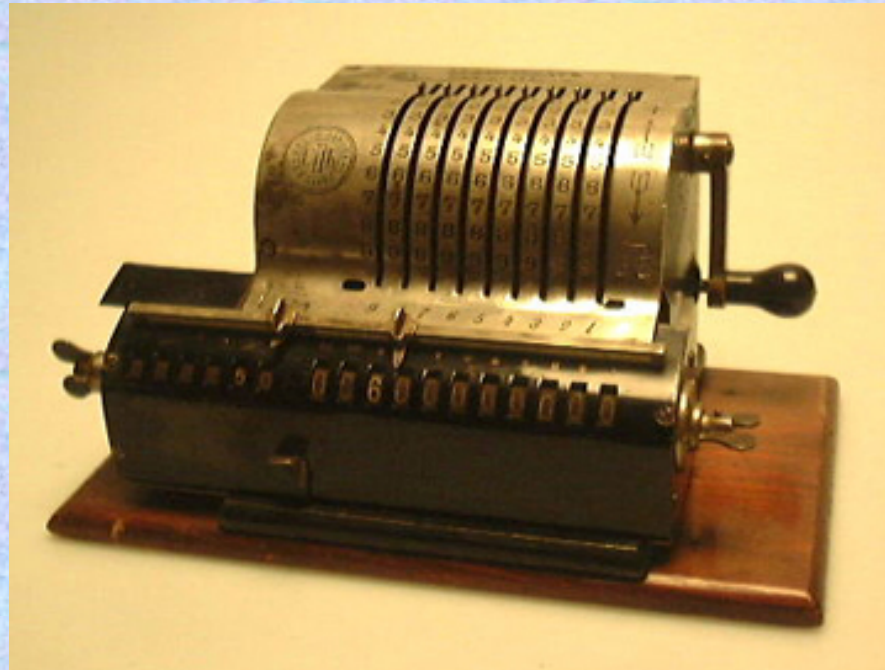
- 1845-ben született a svédországi Dalbyban
- 1874-ben szabadalmaztatta találmányát, melynek fő alkatrésze a Leibniz-kerék továbbfejlesztett változata, az Odhner-kerék. Ezt később több cég is felhasználta (pl. Brunswiga, Facit)
- 1903-ban hunyt el, Szentpéterváron

## Az Odhner-kerék működési elve



**A fogak rugó ellenében  
benyomhatók.  
Mindig annyi számú fog  
marad kívül, amennyit a  
számskálán beállítottunk**

## Egy eredeti Odhner-gép

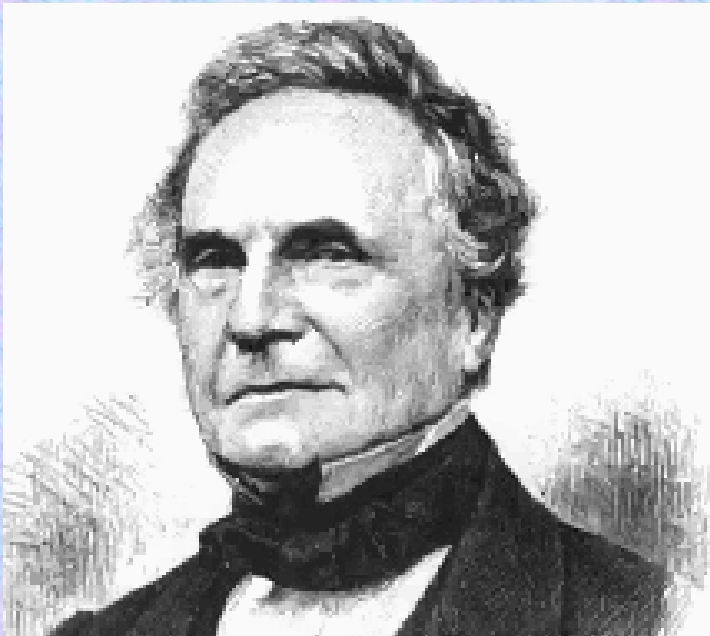


# 3. fejezet

## Az első programvezérelt számítógép

Charles Babbage, Ada Lovelace

## Charles Babbage (1791-1871)



- 1791. dec. 26.-án született. Gyermekkorától matematikai és mechanikai érdeklődésű
- 1817: Londonban matematika professzori állást kap
- kb. 1820: Az Angol Királyi Társaság tagjává választják. Később sok ellentéte lesz a Társasággal. A differencia-gép első elképzései

## A differenciagépről

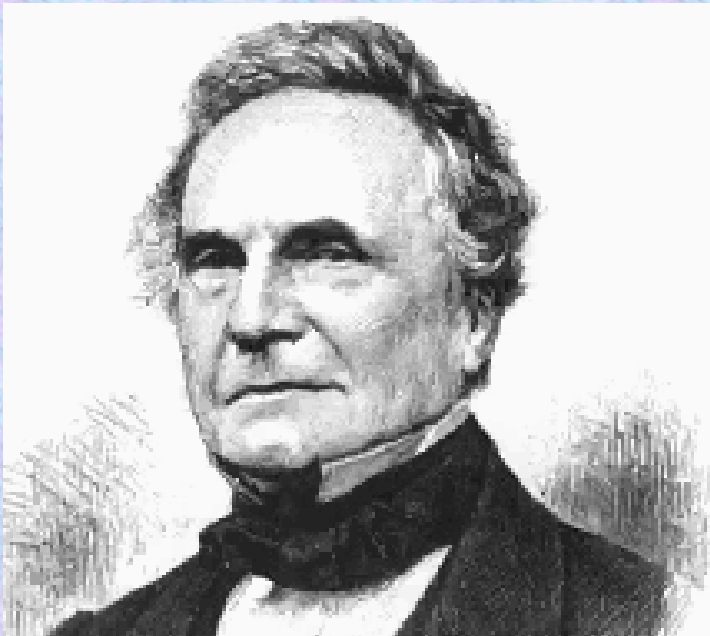
- **Az ötlet feltalálója:** Müller német hadmérnök (1786), aki azonban nem építette meg a gépet.
- **A gép matematikai alapelve:** minden  $n$ -ed fokú polinom  $n$ -edik differenciátáblázata konstans. A polinomok jól használhatók egyéb matematikai függvények közelítésére.
- **Remélt alkalmazási lehetőségek:** navigációs és egyéb célú matematikai táblázatok félig automatikus kiszámítása.

# Egy harmadrendű differenciátáblázat:

$$y = x^3 + 2x^2 - x - 2$$

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
0	-2	2	10	6
1	0	12	16	6
2	12	28	22	6
3	40	50	28	6
4	90	78	34	
5	168	112		
6	280			

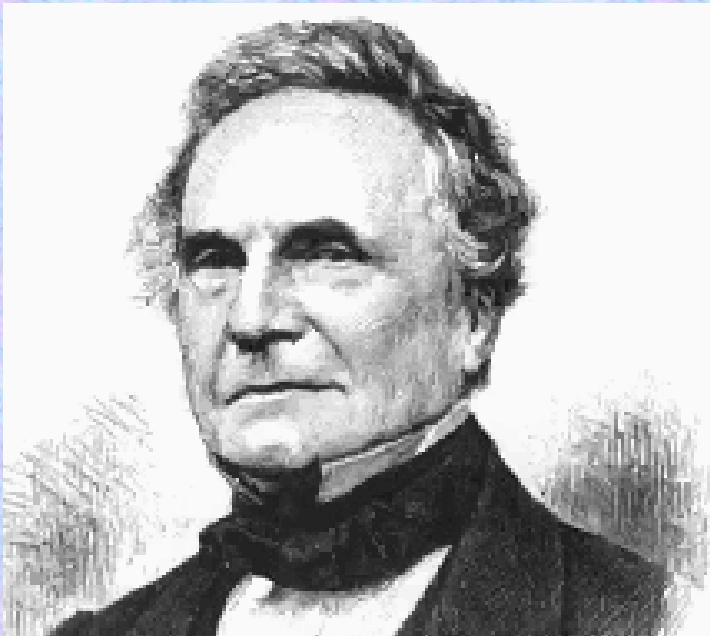
## Charles Babbage (1791-1871)



- **1822**: Elkészül a differenciagép működőképes modellje. Támogatást kér a Királyi Társaságtól
- **1823**: Kap 1500£ előleget; a többi (becslése szerint 3000-5000£) megelőlegezi saját maga, amit a kormány a gép sikeres megépülése esetén visszatérít

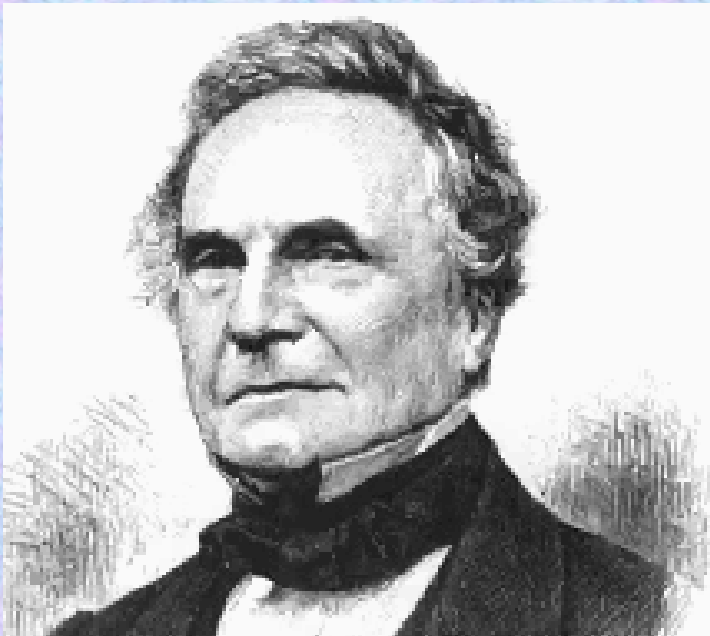


## Charles Babbage (1791-1871)



- **1826**: A Királyi Társaság titkárává jelölik, de Davy (az elnök) megakadályozza megválasztását
- **1828**: A Cambridge-i egyetem matematika professzorává választják (egykor Newton is betöltötte ezt a pozíciót)

## Charles Babbage (1791-1871)



- 1833: Vita, majd később szakítás Clement főmechanikus és Babbage között. Az *analitikai gép* első tervei. A *differentiagép* sikeres demonstrációja
- 1839: Bár soha, egyetlen matematika előadást sem kellett tartania az egyetemen, lemond, hogy csakis a számítástechnikának élhessen

# A differenciagép mintapéldánya



Az  $X^2+X+41$  polinom  
kis  $x$  értékekre csupa  
prímszámot állít elő.

Euler:  $0 \leq X \leq 15$

Escott (1899):  $40 \leq X \leq 89$

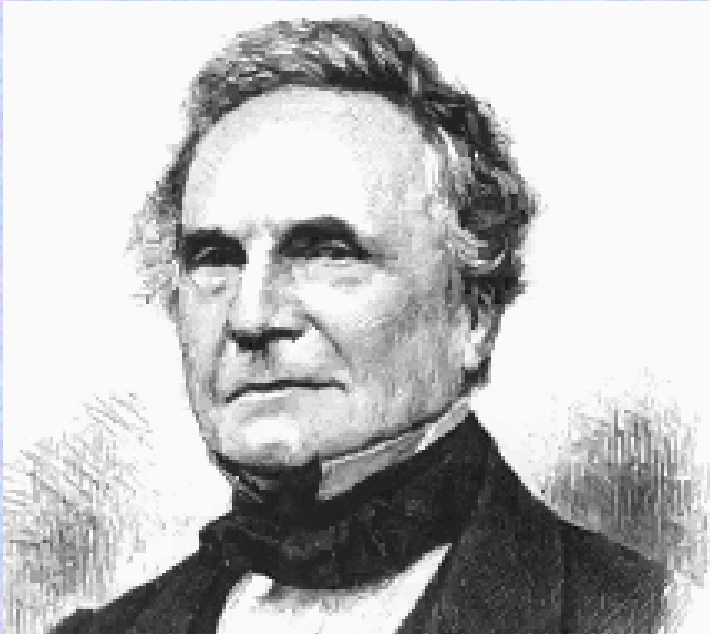
Trost (1959):  $0 \leq X \leq 11000$

értékek közt **4506** prímet  
talált

Babbage 1833-ban 2,5 perc  
alatt gépével (*vagyis annak  
modelljével*) **30** függvény-  
értéket számolt ki

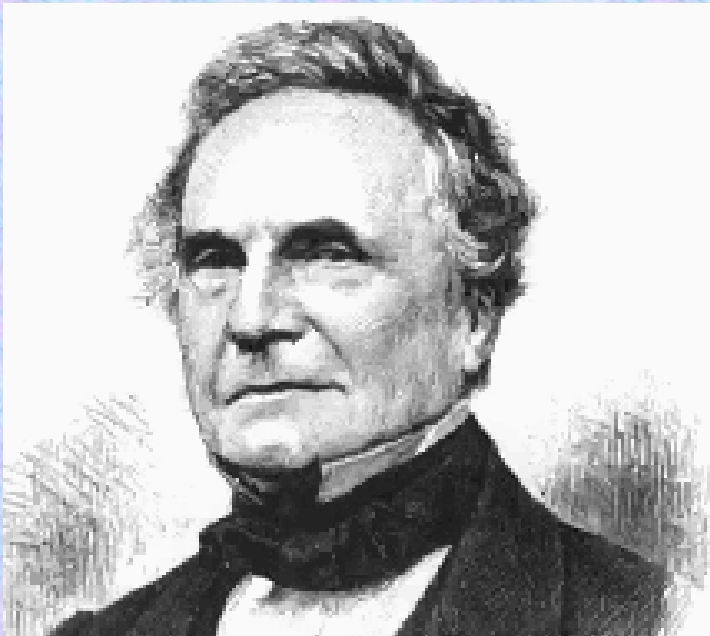
A közös tengelyen, szabadon  
elforduló tárcsák egy-egy  
számjegyet tartalmaznak.

## Charles Babbage (1791-1871)



- 1840: Meghívást kap Olaszországba. Számos szemináriumot tart, melyekről meghívója, Menabrea készít jegyzeteket
- 1842: Megjelenik Menabrea cikke. Ezt A. A. L. angolra fordítja. Végleg megtagadják a támogatást Babbage-től.

## Charles Babbage (1791-1871)



- 1847-49: **Az analitikai gép tervezése mellett, áttervezi a differenciagépet (*D. E. 2*)**
- 1862: **Sikerül elérnie, hogy Londonban — szerencsétlen körülmények között — kiállíthassa**
- 1871: **Halála. A rajzok, szerszámok, stb. fiatalabbik fiára (*Henry Prevost Babbage, 1824-1918*) maradnak**

## A differenciagép történetének vége...

- H. P. Babbage 1879-ben beolvasztatja a Difference Engine megmaradt darabjait. Ugyanezt teszi *Wilkinson is, Clement utóda*
- Differenciagépeket a következő évtizedekben többen is építettek (*pl. Scheutz, Babbage kortársa*). Az 1920-as évektől a könyvelőgépek el tudták látni ezt a feladatot
- H. P. Babbage 1880-ban megépítette az analitikai gép Mill-jét (*ill. annak egy csökkentett modelljét*), majd kiszámította és kinyomtatta a  $\Pi$  első 41 egész számú többszörösét 29 számjegyre

## A differenciagép történetének vége...

- **Babbage születésének 200. évfordulójára, a londoni Science Museum megbízásából, megépítették a *második differenciagépet (D.E.2)* és a hozzá tervezett *nyomtatót*. Mindkettő működőképes.**
- **A gép megépítésénél a Babbage korában elérhető tőrészeket alkalmazták. Ezzel bebizonyosodott, hogy a gép a maga idejében is megépíthető lett volna (*korábban ezt sokan vitatták*).**

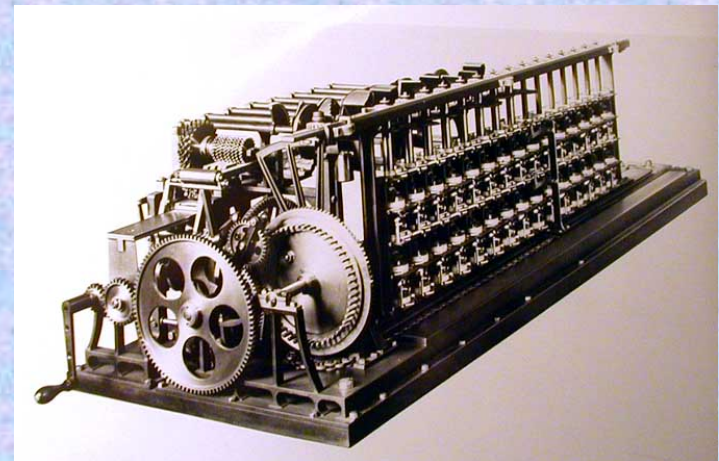
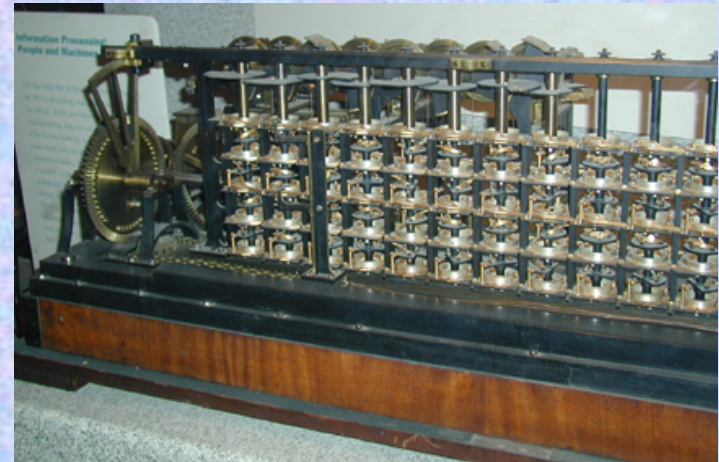
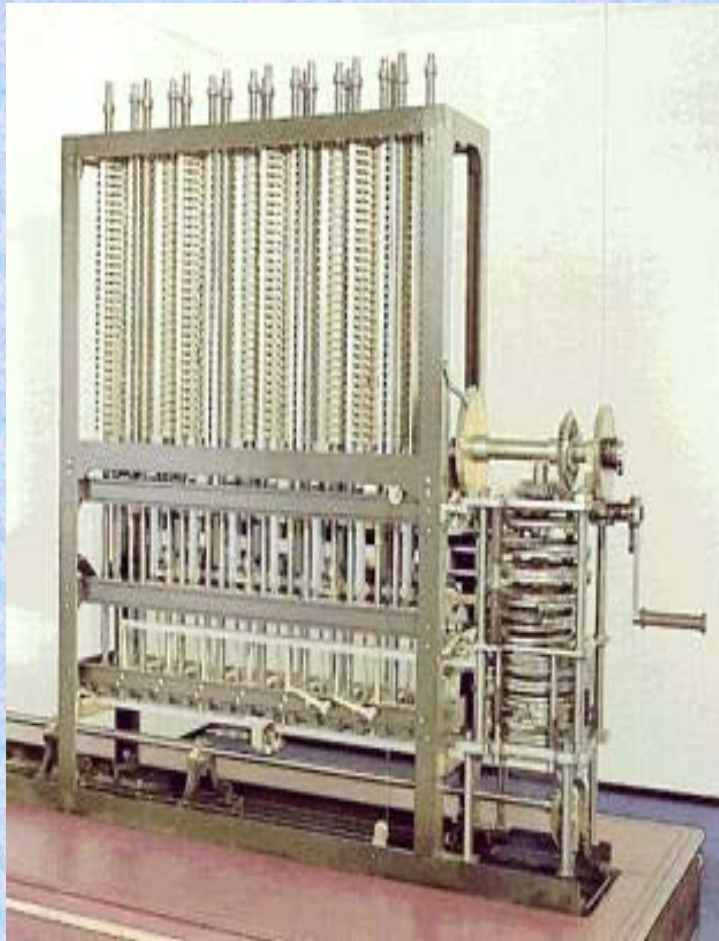
- **1834:** *Dr. Lardner* cikke Babbage gépéről. *George Scheutz (e.: Sejc)* és fia, *Edward* konstrukciója (Svédország)
- **1837:** Babbage-hez hasonló nehézségek. Működő modell készül
- **1851-53:** Sikerül támogatást kapniuk, és megépül a gép. (1854)  
A Svéd Királyi Akadémia részben megtéríti a kiadásait
- **1854:** A gépet Londonba szállítják, és bemutatják. Babbage is elismerően nyilatkozik róla. A gép később Amerikába kerül.



- ~1860: **A brit kormány 1200£-ot ad a „Scheutz-II” gép megépítésére, biztosítási táblázatok számítására (a gép megépült, de nem volt túl megbízható)**  
*[London, Science Museum]*
- ~1875: **Martin Wiberg (Svédország) egy differencia-összeadó segítségével 7 jegyű függvénytáblázatokat számol és publikál**
- 1909: **Hamann (Németország) szerkeszt és használ differencia-összeadót (másodrendű)**  
*(Kézi meghajtás, kb. 6 függvényérték/perc)*
- 1928: **Leslie John Comrie (Új-Zéland) felfedezi, hogy a kereskedelmi könyvelőgépek kis átalakítással differencia-összeadóként is használhatók**

Babbage...

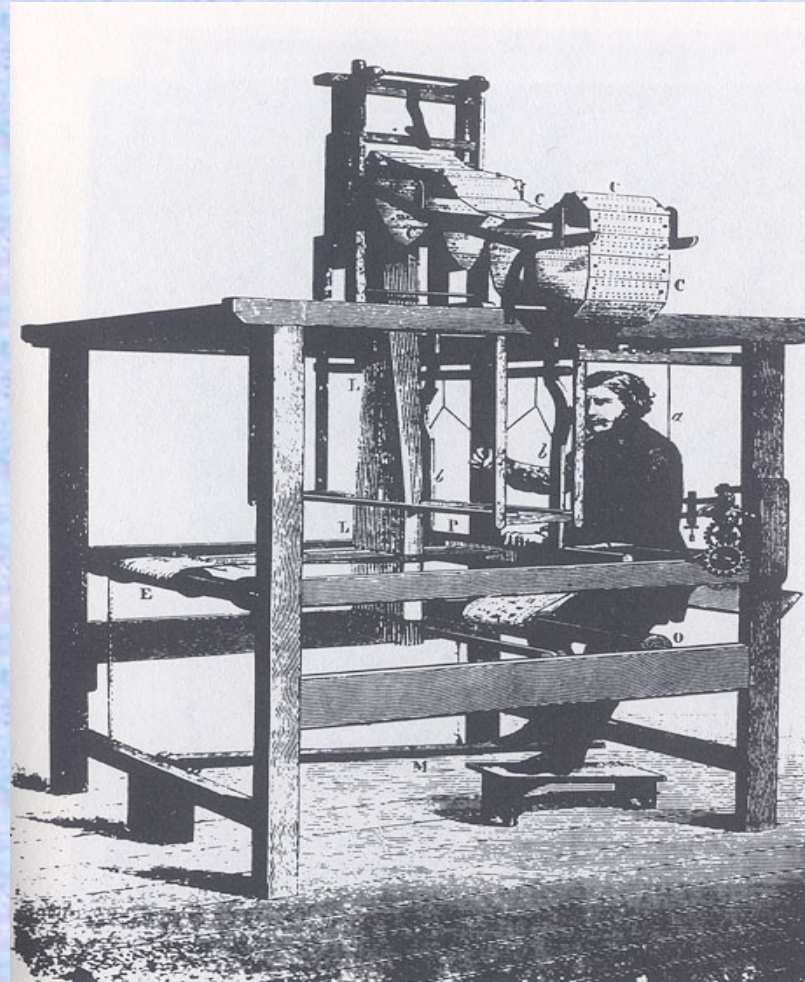
és Scheutz...



...differenciagépe

## Az analitikai gép fő egységei

- **Aritmetikai egység (Mill):**  
a négy alapművelet elvégzésére
- **Tároló (Store):**  
a kiinduló adatok, közbülső-, és végeredmények tárolására
- **Beolvasó egységek (3 db):**  
a különböző kártyatípusok beolvasására
- **Nyomtató egység:**  
az eredmények kinyomtatására
- **Görberajzoló (Curve drawing apparatus):**  
a számítások eredményeinek grafikus megjelenítésére

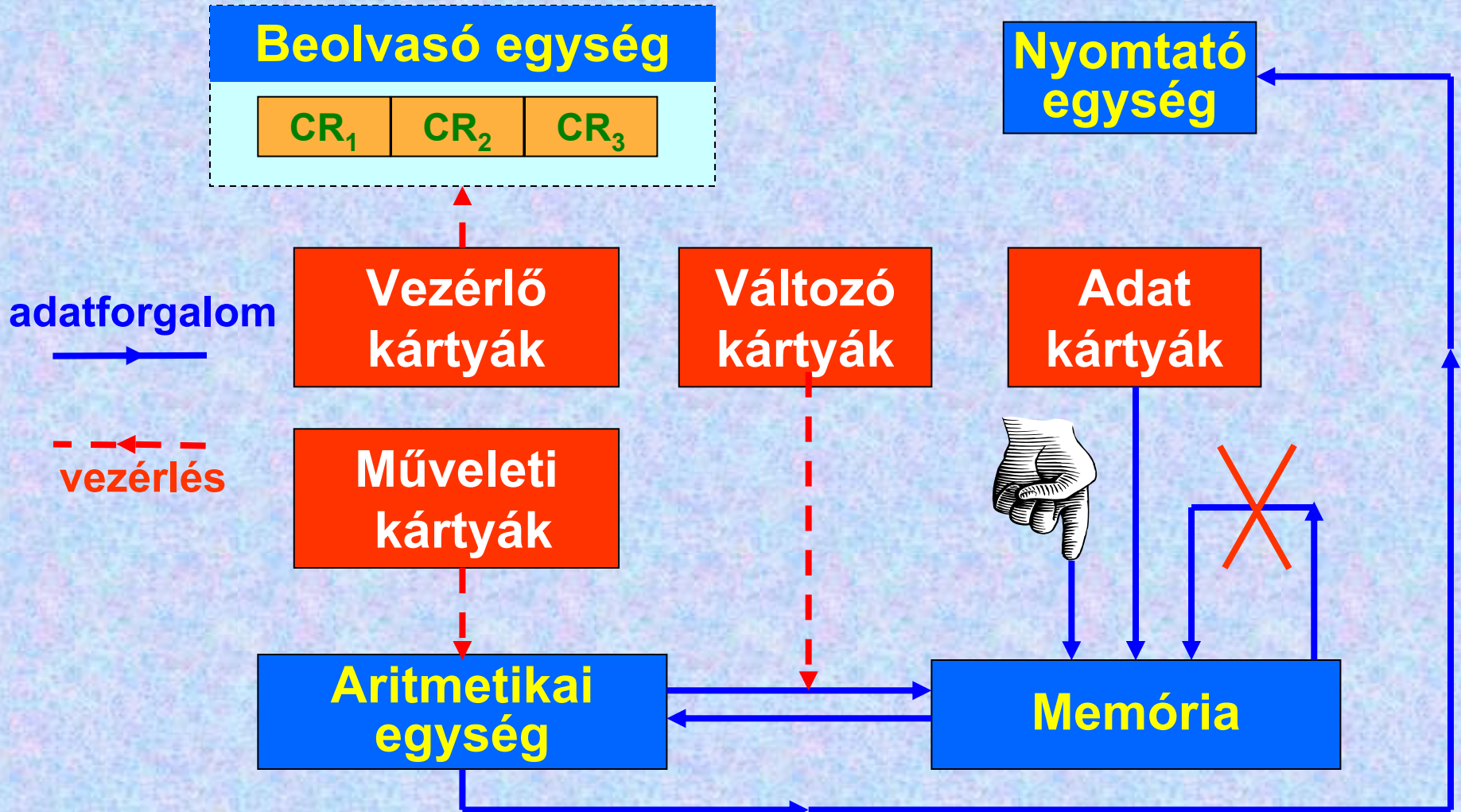


A gép vezérlésére leporelló-szerűen hajtogatott lyukkártyák szolgáltak *(mint Jacquard szövőgépénél)*.

## Az analitikai gép vezérlésére használt kártyák

- **Változó kártyák** (*Variable cards*):  
a tároló és az aritmetikai egység közti adatátvitelt vezérlik
- **Műveleti kártyák** (*Operation cards*):  
az aritmetikai egység által elvégzendő műveletet határozzák meg. Az előző művelet típusa (*pl. szorzás*) megőrződik, amíg meg nem változtatják
- **Programvezérlő kártyák** (*Combinatorial cards*):  
a feltétlen és feltételes ugrásokat vezérlik
- **Adatkártyák** (*Number cards*):  
az adott számítás adatait tartalmazzák  
(*kézi bevitel is lehetséges*)

# Az analitikai gép elvi vázlatja



## Az aritmetikai egység (*Mill*) felépítése és működési elve

- 2 db 50 digitális jegy hosszúságú bemeneti regiszter (*Ingress Axis*). Az aritmetikai műveletek operandusait tartalmazza
- 1 db 100 digitális jegy hosszúságú kimeneti regiszter (*Egress Axis*)
  - Szorzáskor 100 jegyű számként a szorzatot,
  - Osztáskor a hányadost ill. maradékot tartalmazza
- Az aritmetika *fixpontos*, a tizedespont helyét a programozónak kell nyilvántartania

**1888-ban az analitikai gép felépítéséről és működéséről H. P. Babbage részletes cikket publikált, amelyben apja terveinek részletes elemzése alapján rávilágított számos megvalósítási lehetőségre.**



## Ami hiányzik Babbage terveiből

- „Műhelyrajzok” nincsenek, csak általános vázlatok
- Ha három beolvasó van, ezek hogyan vezérlik egymást (pl. visszalépéskor)?
- Egyetlen program sincs „lyukkártyaszintre” lebontva
- A görberajzoló (plotter) csak elvileg szerepel, terve nincs

**Lásd a 20. és 21. diához fűzött magyarázatokat is!**

# Terminológia Babbage műveinek olvasásához

<b>ANALITIKAI GÉP</b>	<b>MAI SZÁMÍTÓGÉP</b>	<b>MAGYAR MEGFELELŐJE</b>
<b>Accidental sign</b>	<b>Sign bit</b>	<b>Előjel bit</b>
<b>Analyst</b>	<b>Programmer</b>	<b>Programozó</b>
<b>Attendant</b>	<b>Operator</b>	<b>Operátor, gépkezelő</b>
<b>Axis</b>	<b>Buffer register</b>	<b>Puffer regiszter</b>
<b>Barrel</b>	<b>Microcode</b>	<b>Mikroprogram</b>
<b>Card</b>	<b>Instruction</b>	<b>Utásítás</b>
<b>Card, Operation</b>	<b>Arithmetic instruction</b>	<b>Aritm. utasítás</b>

# Terminológia Babbage műveinek olvasásához

<b>ANALITIKAI GÉP</b>	<b>MAI SZÁMÍTÓGÉP</b>	<b>MAGYAR MEGFELELŐJE</b>
<b>Card, Combinatorial</b>	<b>Control transfer (test / jump)</b>	<b>Vezérlésátadó utasítás</b>
<b>Card, Number</b>	<b>Load immediate</b>	<b>Direkt memóriába töltés</b>
<b>Card, Variable</b>	<b>Load / Store</b>	<b>Mill↔Store adatforg.</b>
<b>Carriage</b>	<b>Carry propagation</b>	<b>Átvitel (terjedése)</b>
<b>Curve Drawing Apparatus</b>	<b>Plotter</b>	<b>Plotter</b>
<b>Column (of Rack)</b>	<b>Memory (RAM) cell</b>	<b>Memóriarekesz</b>
<b>Cycle</b>	<b>Loop</b>	<b>Programhurok</b>

# Terminológia Babbage műveinek olvasásához

<b>ANALITIKAI GÉP</b>	<b>MAI SZÁMÍTÓGÉP</b>	<b>MAGYAR MEGFELELŐJE</b>
<b>Mill</b>	<b>Arithm. and Log. Unit</b>	<b>Aritmetikai egység</b>
<b>Rack (of Columns)</b>	<b>RAM</b>	<b>Memória (RAM)</b>
<b>Run up</b>	<b>Overflow or Sign Bit Condition Code</b>	<b>Túlsordulás v. előjelbit jelző</b>
<b>Stepping down</b>	<b>Right shift</b>	<b>Jobbra léptetés</b>
<b>Stepping up</b>	<b>Left shift</b>	<b>Balra léptetés</b>
<b>Store</b>	<b>Memory (RAM) Array</b>	<b>Memória</b>
<b>Turn of the Handle</b>	<b>Clock cycle</b>	<b>Óraciklus</b>

## Ada Lovelace (1815-1852)



- Lord Byron lánya. 1 éves korától anyja nevelteti. Babbage tanácsára matematikát is tanul. Egyik tanítója De Morgan
- 1834: Megismerkedik a differenciagéppel
- 1841: Részletesen tanulmányozza Babbage terveit az analitikai gépről

## Ada Lovelace (1815-1852)



- **1842**: Lefordítja Menabrea cikkét, és Babbage javaslatára megjegyzésekkel egészíti ki
- **1843**: Megjelenik a cikk *(Sajnos, ez az egyetlen publikált műve.)*
- **1852**: Hosszú betegség után meghal

# A történelem első számítógép-programja:

az

$$mx+ny=d$$

$$m'x+n'y=d'$$

kétismeretlenes elsőfokú egyenletrendszer megoldása

Columns on which are inscribed the primitive data	Number of the operations	Cards of the operations		Variable cards			Statement of results
		No. of the Operation-cards	Nature of each operation	Columns acted on by each operation	Columns that receive the result of each operation	Indication of change of value on any column	
${}^1V_0 = m$	1	1	x	${}^1V_0 \times {}^1V_4 = {}^1V_6 \dots\dots$	${}^1V_6 \dots\dots$	$\left. \begin{matrix} {}^1V_0 = {}^1V_0 \\ {}^1V_4 = {}^1V_4 \end{matrix} \right\}$	${}^1V_6 = mn'$
${}^1V_1 = n$	2	"	x	${}^1V_3 \times {}^1V_1 = {}^1V_7 \dots\dots$	${}^1V_7 \dots\dots$	$\left. \begin{matrix} {}^1V_3 = {}^1V_3 \\ {}^1V_1 = {}^1V_1 \end{matrix} \right\}$	${}^1V_7 = m'n$
${}^1V_2 = d$	3	"	x	${}^1V_2 \times {}^1V_4 = {}^1V_8 \dots\dots$	${}^1V_8 \dots\dots$	$\left. \begin{matrix} {}^1V_2 = {}^1V_2 \\ {}^1V_4 = 0V_4 \end{matrix} \right\}$	${}^1V_8 = dn'$
${}^1V_3 = m'$	4	"	x	${}^1V_5 \times {}^1V_1 = {}^1V_9 \dots\dots$	${}^1V_9 \dots\dots$	$\left. \begin{matrix} {}^1V_5 = {}^1V_5 \\ {}^1V_1 = 0V_1 \end{matrix} \right\}$	${}^1V_9 = d'n$
${}^1V_4 = n'$	5	"	x	${}^1V_0 \times {}^1V_5 = {}^1V_{10} \dots\dots$	${}^1V_{10} \dots\dots$	$\left. \begin{matrix} {}^1V_0 = 0V_0 \\ {}^1V_5 = 0V_5 \end{matrix} \right\}$	${}^1V_{10} = d'm$
${}^1V_5 = d'$	6	"	x	${}^1V_2 \times {}^1V_3 = {}^1V_{11} \dots\dots$	${}^1V_{11} \dots\dots$	$\left. \begin{matrix} {}^1V_2 = 0V_2 \\ {}^1V_3 = 0V_3 \end{matrix} \right\}$	${}^1V_{11} = dm'$
	7	2	-	${}^1V_6 - {}^1V_7 = {}^1V_{12} \dots\dots$	${}^1V_{12} \dots\dots$	$\left. \begin{matrix} {}^1V_6 = 0V_6 \\ {}^1V_7 = 0V_7 \end{matrix} \right\}$	${}^1V_{12} = mn' - m'n$
	8	"	-	${}^1V_8 - {}^1V_9 = {}^1V_{13} \dots\dots$	${}^1V_{13} \dots\dots$	$\left. \begin{matrix} {}^1V_8 = 0V_8 \\ {}^1V_9 = 0V_9 \end{matrix} \right\}$	${}^1V_{13} = dn' - d'n$
	9	"	-	${}^1V_{10} - {}^1V_{11} = {}^1V_{14} \dots\dots$	${}^1V_{14} \dots\dots$	$\left. \begin{matrix} {}^1V_{10} = 0V_{10} \\ {}^1V_{11} = 0V_{11} \end{matrix} \right\}$	${}^1V_{14} = d'm - dm'$
	10	3	÷	${}^1V_{13} \div {}^1V_{12} = {}^1V_{15} \dots\dots$	${}^1V_{15} \dots\dots$	$\left. \begin{matrix} {}^1V_{13} = 0V_{13} \\ {}^1V_{12} = {}^1V_{12} \end{matrix} \right\}$	${}^1V_{15} = \frac{dn' - d'n}{mn' - m'n} = x$
	11	"	÷	${}^1V_{14} \div {}^1V_{12} = {}^1V_{16} \dots\dots$	${}^1V_{16} \dots\dots$	$\left. \begin{matrix} {}^1V_{14} = 0V_{14} \\ {}^1V_{12} = {}^1V_{12} \end{matrix} \right\}$	${}^1V_{16} = \frac{d'm - dm'}{mn' - m'n} = y$
1	2	3	4	5	6	7	8

## Ada Lovelace kiegészítései Menabrea cikkéhez (*Terjedelmük mintegy háromszorosan haladja meg az eredeti cikk terjedelmét!*)

- Az analitikai gép tudományos jelentősége. A differencia-összeadó és az analitikai gép összehasonlítása
- Az analitikai gép minden lehetséges (*kiszámítható*) függvény megtestesítője, sőt, lehetőségei ezen túl is terjednek  
(*pl. számítógépes algebra, zenei összhangzattan*)
- A Babbage/Menabrea által, a programok leírására használt (*táblázatos*) jelölésrendszer módosítása, lehetővé téve a változók értékváltozásainak nyomon követését (*ld. az előző diához fűzött jegyzetet!*)



- A program változóinak (V0,V1, ...) definíciója és funkcióik osztályozása (*bemenő adat, közbűlső adat, végeredmény*)
- A programkártyák visszaléptetésének (*program-hurok*) jelentősége
- A számítógép tudományos jelentősége: elvégezhet olyan dolgokat, amelyekre az ember fizikailag nem képes. „Újat” nem hoz létre, de mindent elvégéz, amire programozható

**Ada Lovelace-ről - munkásságának emlékére - nevezték el az ADA programozási nyelvet**

## Az analitikai gép emulátora

- A '90-es évek végén *Java applet* formájában elkészítették az analitikai gép emulátorát
- Az emulátor által utánozott egységek:
  - Store
  - Mill
  - 1 (közös) kártyaolvasó
  - Printer
  - Plotter
- Utasításrendszer: 1 című natív gépi kód

- **Felhasznált dokumentáció: a Babbage és Ada Lovelace által hátrahagyott dokumentumok**
- **A dokumentáció hiányainak kiegészítése a „Maximum Likelihood” („Babbage valószínűleg így gondolta”) elv alapján**
- **Babbage elképzelése: a Programozó (Analyst), a Gépkezelő (Attendant) és a Gép összetartozó, együttműködő rendszert alkot**

## Az emulátor ablakai

**Főablak (3 részből áll):**

**Program ablak**

**(itt látható a futtatandó program)**

**Nyomtató ablak**

**(itt láthatók a program által kiírt eredmények)**

**Napló ablak**

**(nyomkövetési információ)**

## Az emulátor ablakai

**Állapotkijelző ablak (3 részből áll):**



## Az emulátor kivonatos utasításrendszere

UTASÍTÁS	ELNEVEZÉS	JELENTÉS	PÉLDA
<b>L</b> <cím>	<b>Load</b>	<b>Tart. betöltése a Mill-be</b>	<b>L121</b>
<b>S</b> <cím>	<b>Store</b>	<b>Eredmény (maradék) tárolása</b>	<b>S121</b>
<b>S</b> <cím>'	<b>Store Prime</b>	<b>Hányados tárolása</b>	<b>S121'</b>
<b>+, -, ×, ÷</b>	<b>Arith. Op.</b>	<b>Köv. aritmetikai műv.</b>	<b>×</b>
<b>&gt;</b> <lépésszám>	<b>Step Down</b>	<b>Operandus logikai jobbra léptetése</b>	<b>&gt;20</b>
<b>&lt;</b> <lépésszám>	<b>Step Up</b>	<b>Operandus logikai balra léptetése</b>	<b>&lt;20</b>
<b>P</b>	<b>Print</b>	<b>Előző eredmény kinyomtatása</b>	

## Az emulátor kivonatos utasításrendszere

UTASÍTÁS	ELNEVEZÉS	JELENTÉS	PÉLDA
<b>CB?</b> <szám>	<b>Combinatorial Backward</b>	<b>Feltételes visszaugrás</b>	<b>CB?17</b>
<b>CF?</b> <szám>	<b>Combinatorial Forward</b>	<b>Feltételes előre ugrás</b>	<b>CF?17</b>
<b>CB+</b> <szám>	<b>Backward</b>	<b>Feltétlen visszaugrás</b>	<b>CB+17</b>
<b>CF+</b> <szám>	<b>Forward</b>	<b>Feltétlen előre ugrás</b>	<b>CF+17</b>

**A feltételes ugrások a „túlcsordulás vagy előjelváltás” (*Run Up Lever*) állapotától függnnek**

# Egy egyszerű program az emulátorra

## Az első tíz négyzetszám kinyomtatása

V0: még hátralevők+1

V1: az alap

V2: értéke mindig 1

N0 11

N1 1

N2 1

x

L1

L1

P

Az alap növelése 1-gyel

+

L1

L2

S1

A még hátralevők számának csökkentése 1-gyel

-

L0

L2

S0

Vizsgálat: ha  $1-V0 < 0$ ,  
vissza 20 kártyával

L2

L0

CB?20

Akkor lép vissza, ha az első operandus és az eredmény előjele különböző

**Az emulátor behívása**



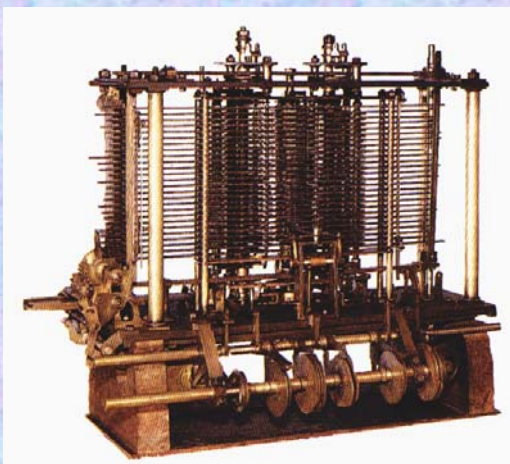
## Mi történt az analitikai gép gondolatával?

*Közel 100 évig — a XX. szd. 30-as éveigig — semmi, bár az analitikai gépben megfogalmazott egyes ötletek tovább éltek (ld. lyukkártya használata, differencia-összeadók, stb.)*

### Howard Aiken (1900-1973)

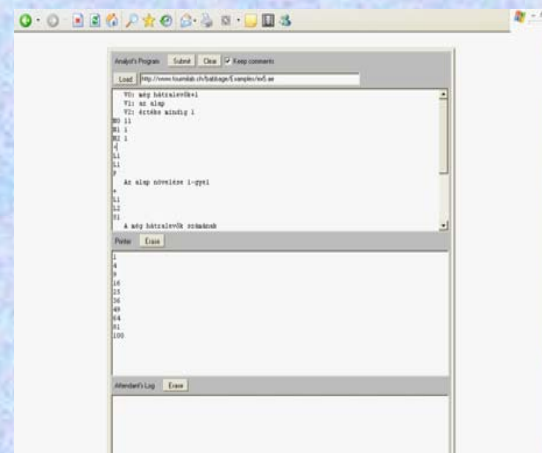
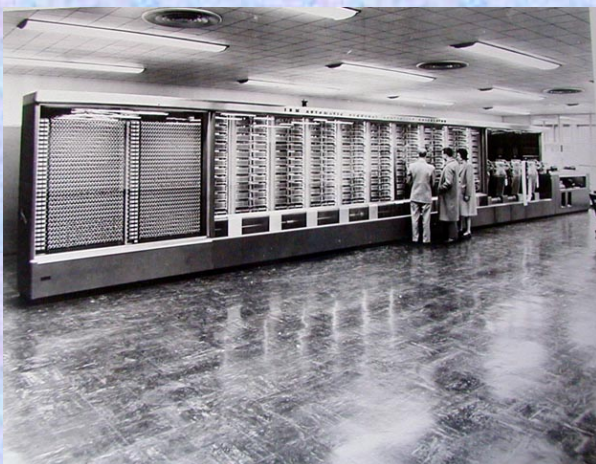
- 1937: Aiken (Fizikai Intézet, Harvard egyetem) javaslatot tesz egy univerzális számológép építésére
- 1939: Közös fejlesztési szerződés a Harvard és az IBM között
- 1943: A gépet működés közben demonstrálják Endicottban, az építés helyén
- 1944 május: Átadás Harvardon

- **Hivatalos elnevezése:**  
„**IBM Automatic Sequence Controlled Calculator**”  
(**ASCC**), vagy későbbi nevén **MARK-I**
- **Működött: 1959-ig**
- **Jellemzői:**
  - **Babbage analitikai gépének lényegében megfelelő, mechanikus működésű, de már elektromos meghajtású gép**
  - **60 regiszterből álló tároló**  
*(23 decimális jegy+előjel)*
  - **72 összeadó regiszter**  
*(23 jegy+előjel)*
  - **2 szorzómű**
  - **lyukszalagos vezérlés**



Az analitikai gép részlete...

...egyesek szerint a teljesen megépült gép kb. futballpálya méretű lett volna...



...a MARK-1...  
LGy.

# Az analitikai és az első számítógépek tulajdonságai

<i>Sz. gép</i>	<i>Üzembeh. éve</i>	<i>Bináris</i>	<i>Elektronikus</i>	<i>Programozható</i>	<i>Turing-teljes</i>
Zuse Z3	1941 máj.	Igen	Nem	Lyukasztott film	Igen (1998)
ABC Computer	1941 nyár	Igen	Igen	Nem	Nem
Colossus	1943 vége 1944 eleje	Igen	Igen	Áthuzalozással	Nem
Mark- I (ASCC)	1944	Nem	Nem	Lyukszalaggal	Nem
ENIAC	1944	Nem	Igen	Áthuzalozással	Igen
	1948	Nem	Igen	Fv. tábl. (ROM)	Igen
Anal.gép	-	Nem	Nem	Igen	Igen

# 4. fejezet

## A lyukkártyás adatfeldolgozás kezdetei

Hermann Hollerith

## Hermann Hollerith (1860-1929)



- 1860-ban született az Egyesült Államok-beli Buffalo-ban
- 1889-ben a szabadalmaztatta a gépét, mellyel az 1890-es népszámlálási adatokat *(kb. 63 millió ember!)* mindössze 1 hónap alatt dolgozta fel

## Hermann Hollerith (1860-1929)



- 1896-ban saját céget alapított (*Tabulating Machine Company*)
- 1911-ben a cég egyesült egy óra- és egy mérleggyárral (*Computing Tabulating Recording Company*)
- 1917: A cég kanadai leányvállalata felvette az *International Business Machines* nevet
- 1924: Az egész cég az *IBM* nevet vette fel
- 1929-ben Washingtonban hunyt el

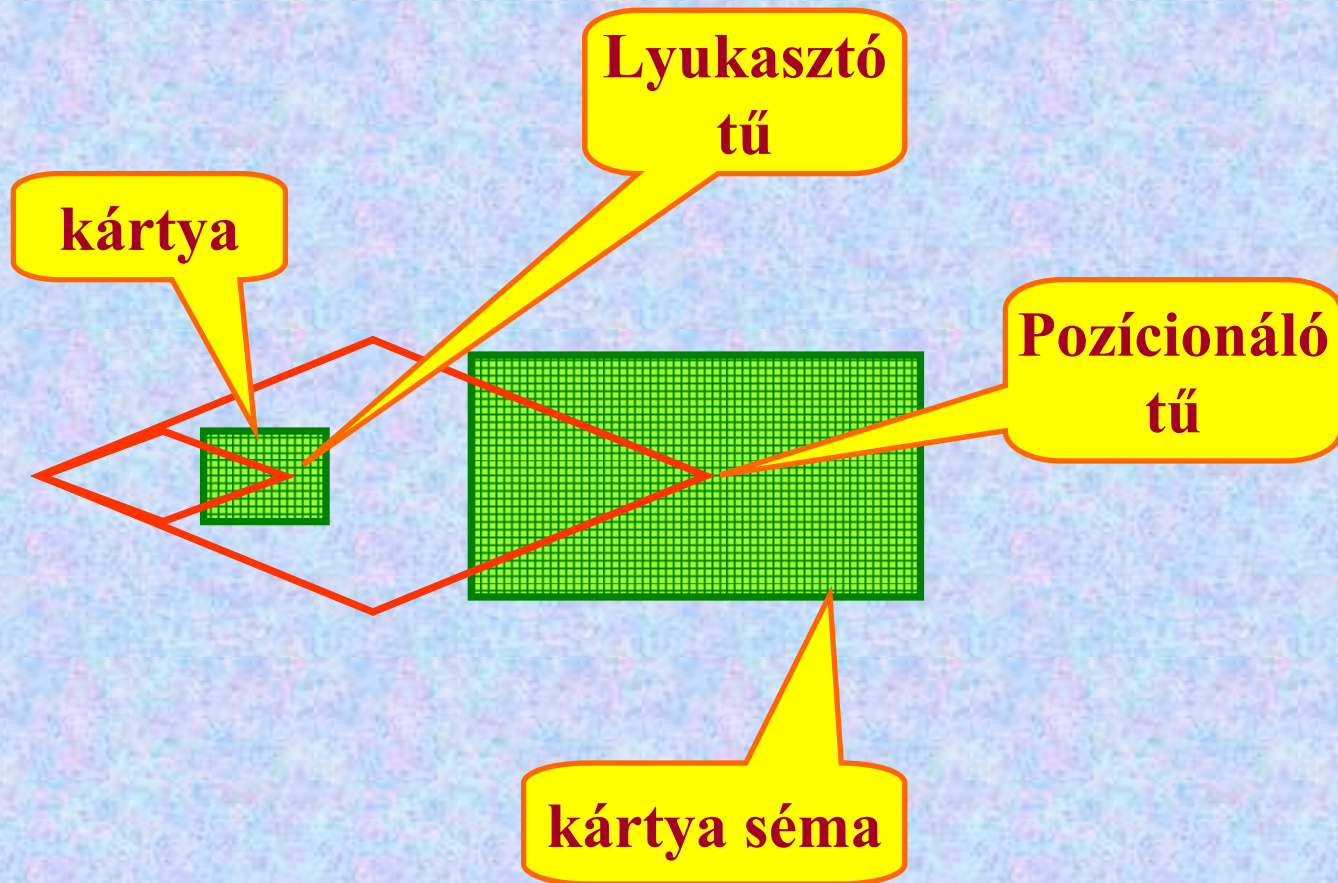
- **Lyukkártyák azonosítása:** körzeti kód, és ezen belül sorszámozás. (*Ellenőrzés lehetősége: tű vagy drót*)
- **Lyukasztás:** karakter pozíció szerint (*vagyis nem karakterkód*)
- **Lyukasztás módszere:** Pantográf-lyukasztó
- **Statisztika készítés:** Dugaszolósos, mechanikus érzékelőkkel letapogatott számlálók
- **Kártyaválogatás:** Relék segítségével felnyitott dobozokba helyezéssel (*kézi berakás, működtetés*)
- **Későbbi fejlemények:** 12 soros és 45 (*később 80*) oszlopos lyukkártya, Hollerith kártyalyukasztó, rendező, tabulátor, automatikus kártyatovábbítás, integrált gépparkok (*1890→1940*)



# Hollerith gépe



## A pantográf-lyukasztó működési elve



## A pantográf-lyukasztó



**A képen a kártyalyukasztó látható, a pozicionáló tűvel**

# Mechanikus lyukkártyarendezés

*(adott kulcsmező szerint)*

**3 jegyű egész kulcsmező esetén:**

	Csop.	Tagjai	Csop.	Tagjai	Csop.	Tagjai
380	0	380	0	104,106,108,208	0	55,99
511	1	511	1	511,913,115	1	104,106,108,115,133
55	2	592	2	—	2	208
108	3	133,913	3	133	3	380
208	4	104	4	—	4	—
133	5	55,115	5	55	5	511,592
592	6	106	6	—	6	—
104	7	—	7	—	7	—
99	8	108,208	8	380	8	—
106	9	99	9	592,99	9	913
115						
913						

**A módszer előnye volt, hogy a számokat nem kellett hozzá nagyság szerint összehasonlítani. Elektronikus gépekre ez a módszer nem lenne gazdaságos!!**

# 5. fejezet

A világ első *Turing-teljes*  
elektronikus számítógépe: az ENIAC

ENIAC= Electronic Numerical Integrator And  
Computer

- **Tervezők:** Presper Eckert, Hermann Goldstine, John Mauchly és sokan mások
- **Tervjavaslat:** 1943. április
- **Kivitelező:** Moore School of Electric Engineering, University of Pennsylvania
- **Beruházó:** Ballistic Research Laboratory (BRL), Aberdeen\*
- **Beruházás kezdete:** 1943. június 1.
- **Befejezés:** 1945. vége
- **Fő cél:** Lövéstani táblázatok számítása
- **Technológia:** elektroncsöves áramkörök  
(*kb. 18 000 elektroncső*)

## Az architektúra főbb jellemzői

- Különálló, fél-autonóm egységek
- Részben elosztott vezérlés
- Külső programozás, dugaszolással
- Különválasztott „numerikus” és „tényleges” programozás
- Lyukkártyás I/O
- Decimális számrendszer
- Párhuzamos működés lehetősége (korlátozottan)
- Még nem Neumann-elvű, de univerzális (Turing-teljes) gép



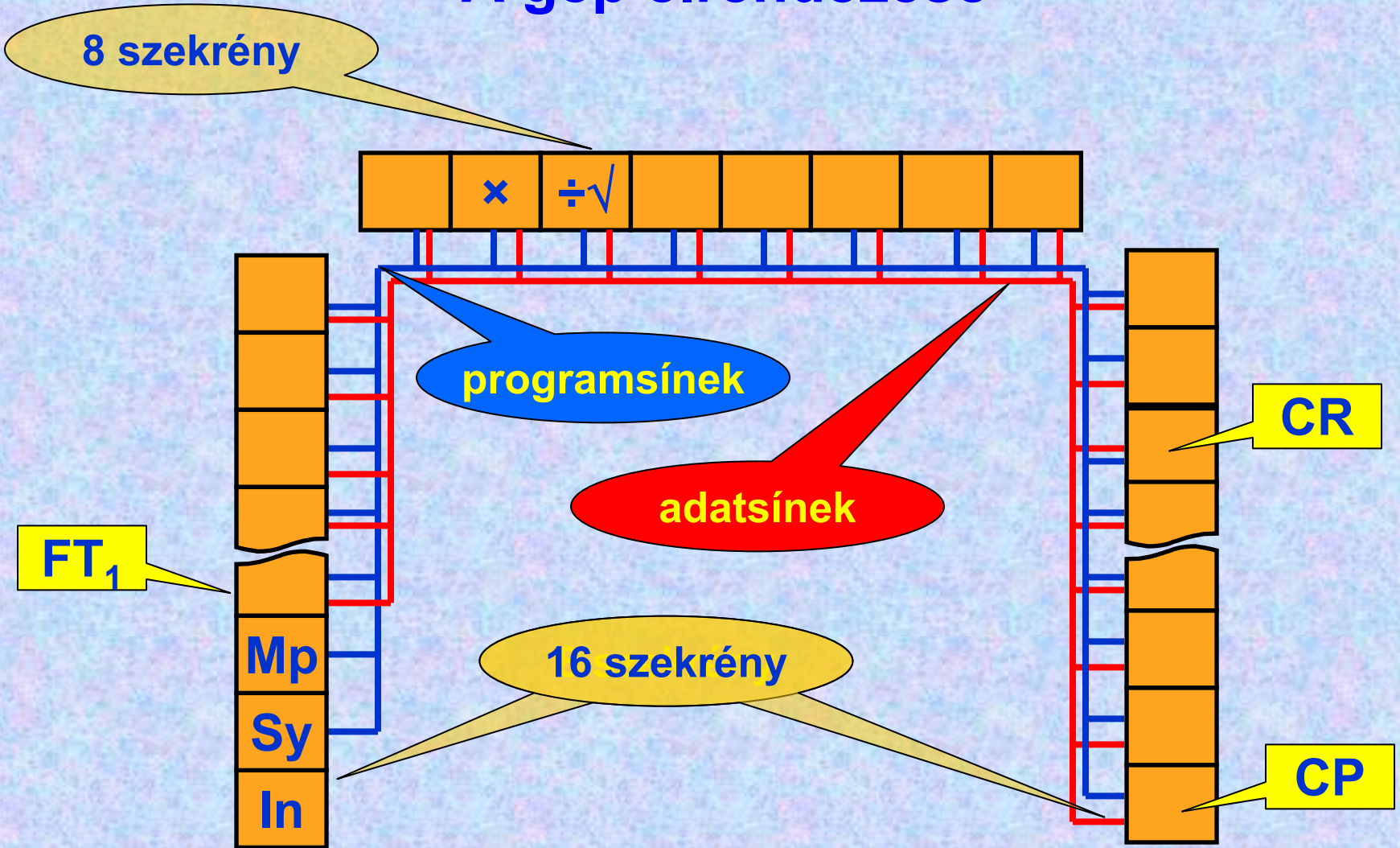
# Funkcionális egységek

- 20 akkumulátor (10 decimális jegy+előjel)
- 3 függvénytáblázat (104-104 fv. érték, egyenként 12-12 dec. számjegy+2 [!!!] előjel)
- 1 szorzóegység
- 1 osztó/gyökvonó egység
- 1 „konstansküldő” (*Constant transmitter*) egység
- Inicializáló egység
- Szinkronjel generátor
- Központi programadó (*Master programmer*)
- 1 kártyaolvasó
- 1 kártyalyukasztó

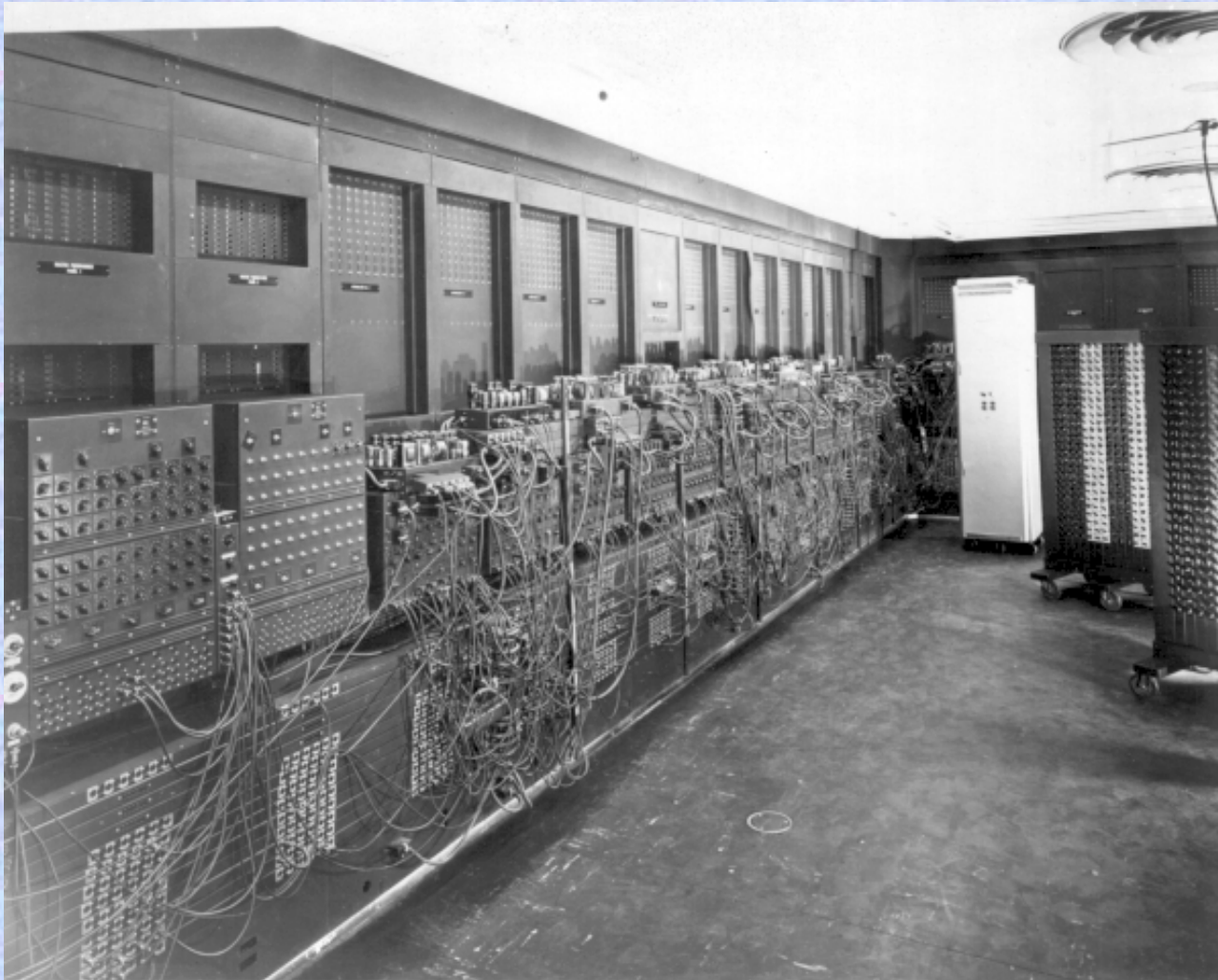
## A főbb egységek funkciói

- **Akkumulátorok:** összeadás és kivonás, operatív tároló
- **Szorzó egység:** szorzás
- **Osztó/gyökvonó egység:** osztás és gyökvonás
- **Konstansküldő egység:** állandóan olvasható puffertároló, amely a kártyáról utoljára beolvasott számokat tárolja, de értéke kézileg is beállítható
- **Központi programadó:** a program logikai vezérlése, ciklusok szervezése
- **Függvénytáblázatok:** táblázatosan adott függvények értékeinek tárolása; később a program tárolására is használták („*Neumann-kód*”)

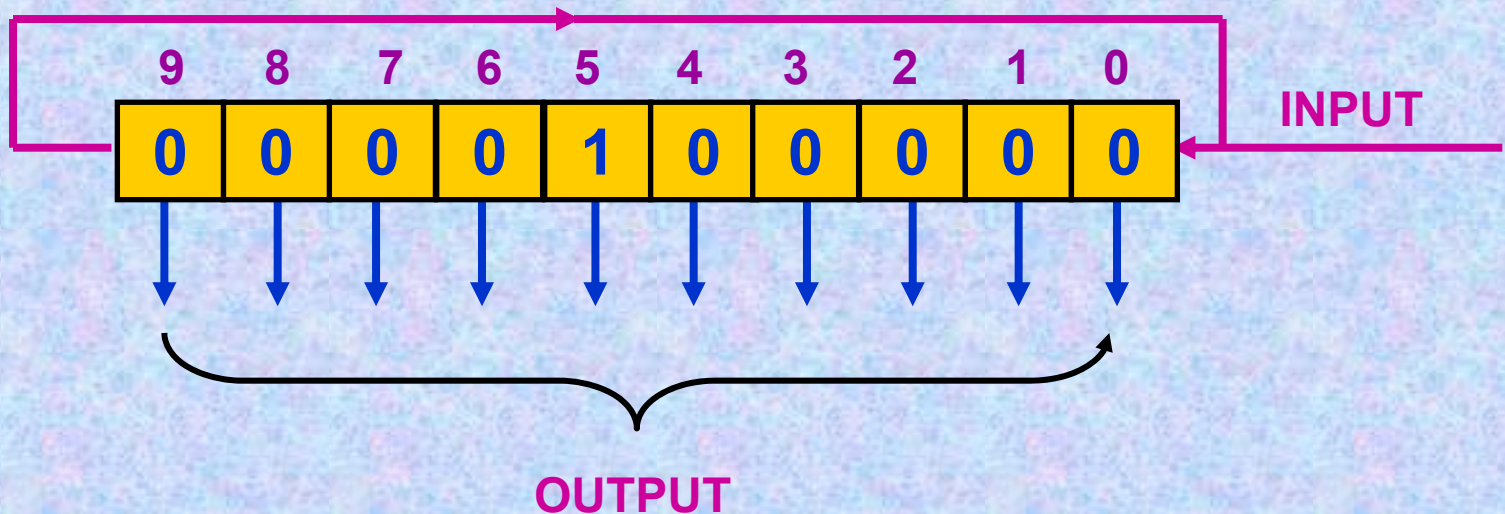
# A gép elrendezése



## Az ENIAC gépterem

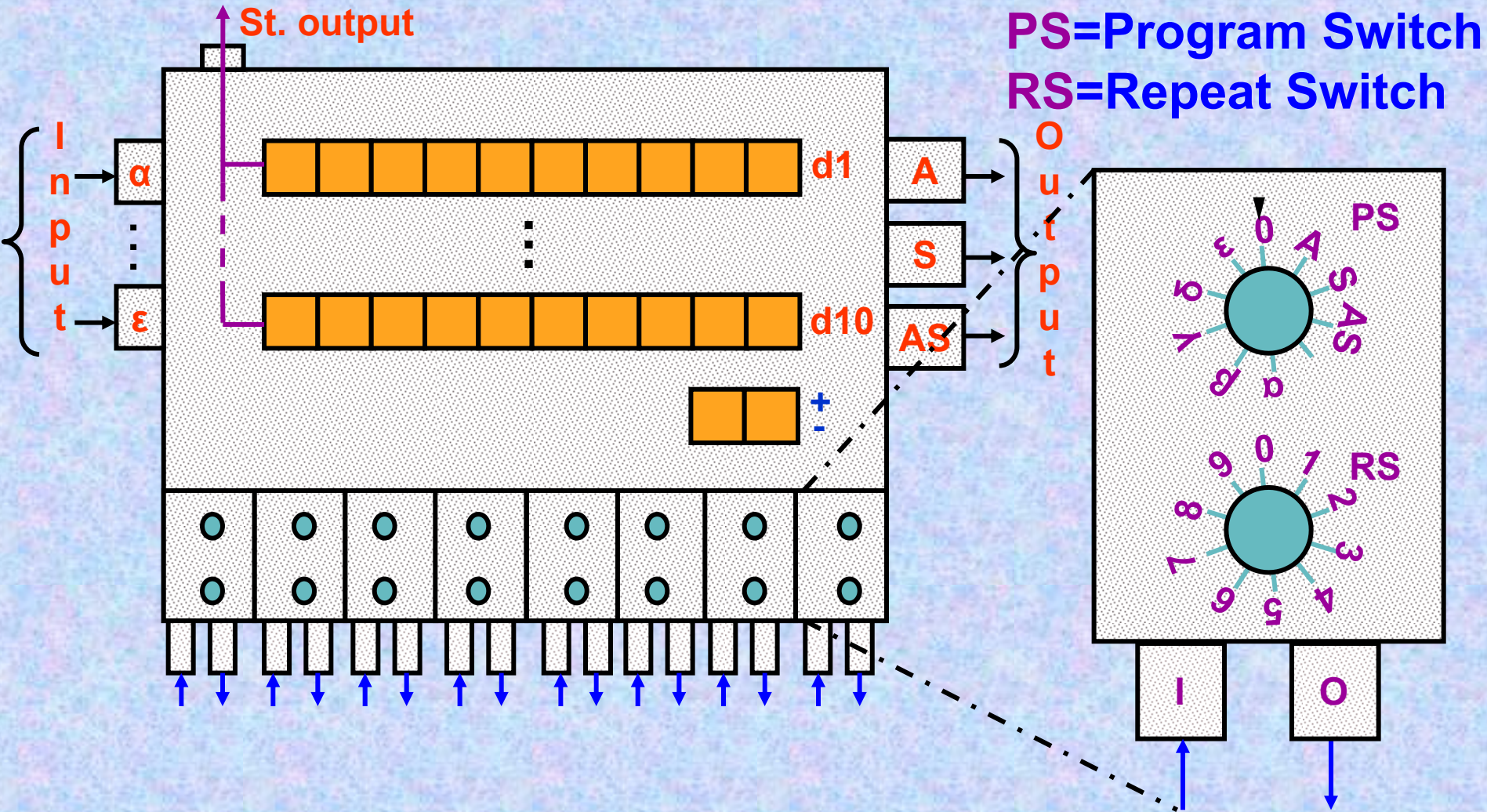


## A legfontosabb alapelem: a decimális gyűrűs számláló

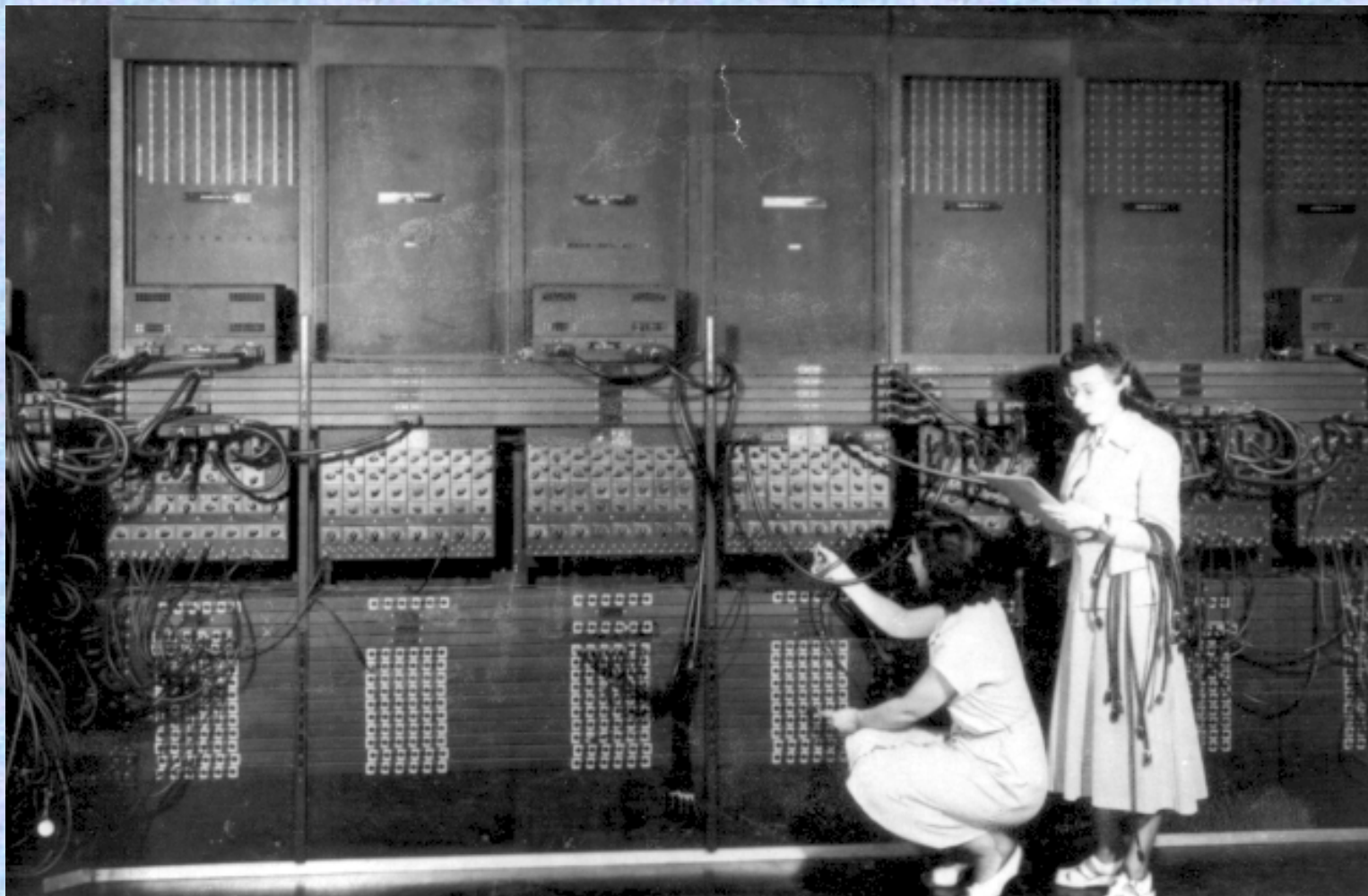


**A ciklikus gyűrűs számláló a már Babbage által is használt számkerék elektronikusan megvalósított megfelelője**

# Az akkumulátorok vázlatos felépítése



## Az akkumulátorok programozása



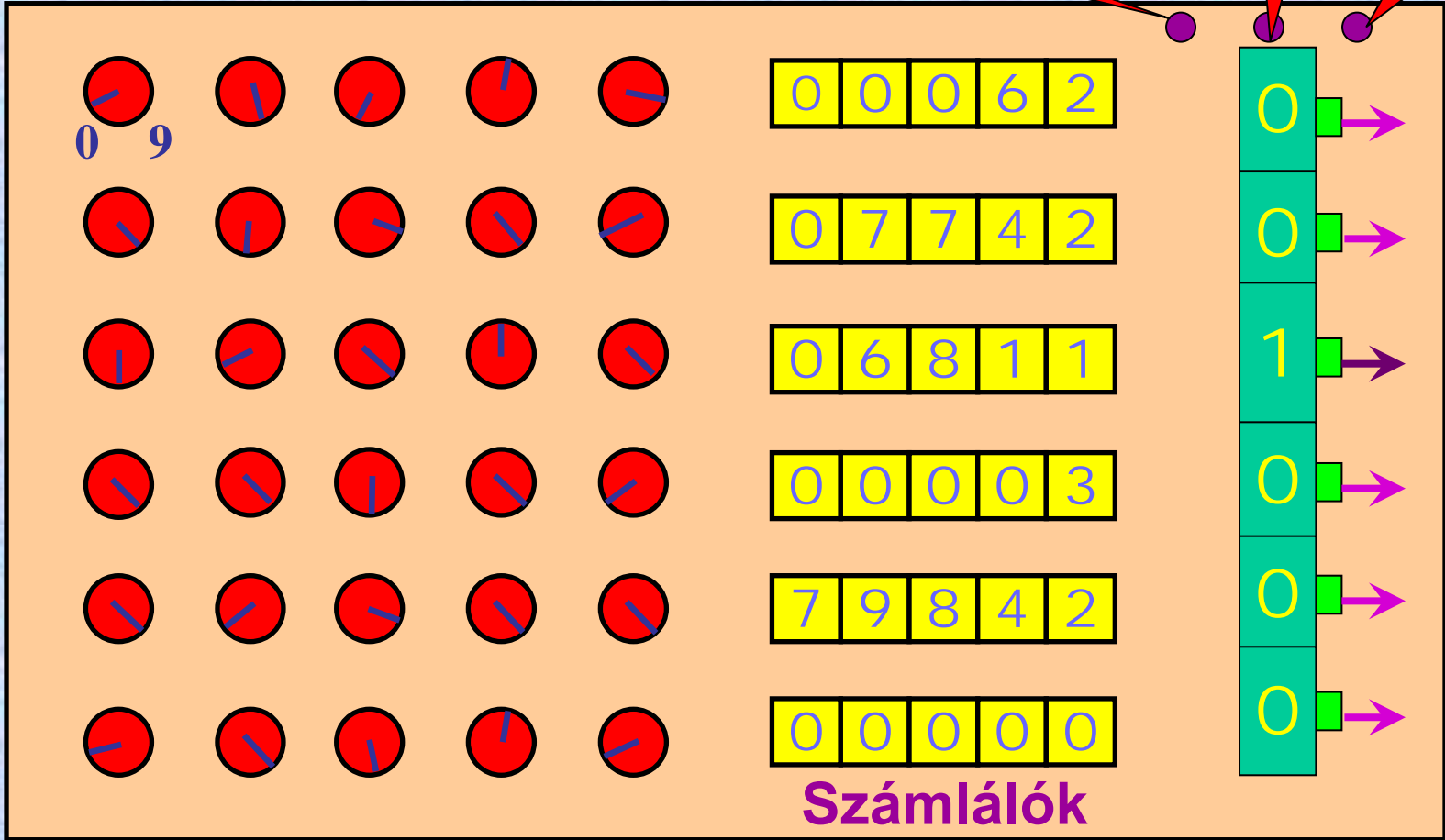
# 10 programadó egység

Tízállású kapcsolók

Léptet

Start

Töröl





## Programozási példa

**Kiszámítandók az egész számok négyzetei és köbei 1-től 100-ig, és minden ötödik szám-hármas ( $n$ ,  $n^2$  és  $n^3$ ) a lyukasztón kinyomtatandó.**

**A program a következő képleteket használja:**

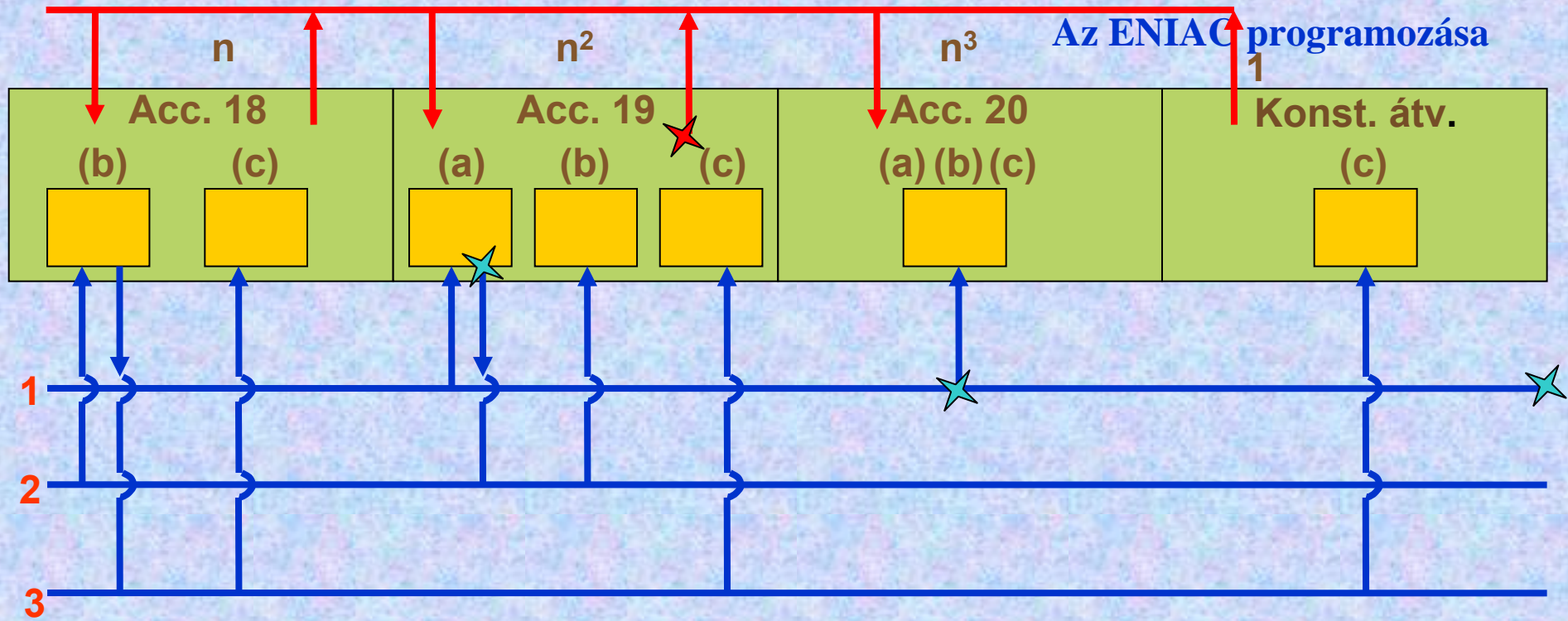
$$(n+1)^2 = n^2 + 2n + 1$$
$$(n+1)^3 = n^3 + 3n^2 + 3n + 1$$

## A bemutatandó program Pascalban *(részlet)*:

*...Begin*

```
Clrscr;  Stepper1:=20;  {Első programadó}
Repeat
  Stepper2:=5;  {Második programadó}
  Repeat
    Acc20:=Acc20+Acc19+Acc19+Acc19;
    Acc20:=Acc20+Acc18+Acc18+Acc18;
    Acc19:=Acc19+Acc18+Acc18;
    Inc (Acc20);  Inc (Acc19);  Inc (Acc18);
    Dec (Stepper2)
  Until Stepper2=0;
  WriteLn (Acc18:10,Acc19:10,Acc20:10);
  Dec (Stepper1)
Until Stepper1=0;
```

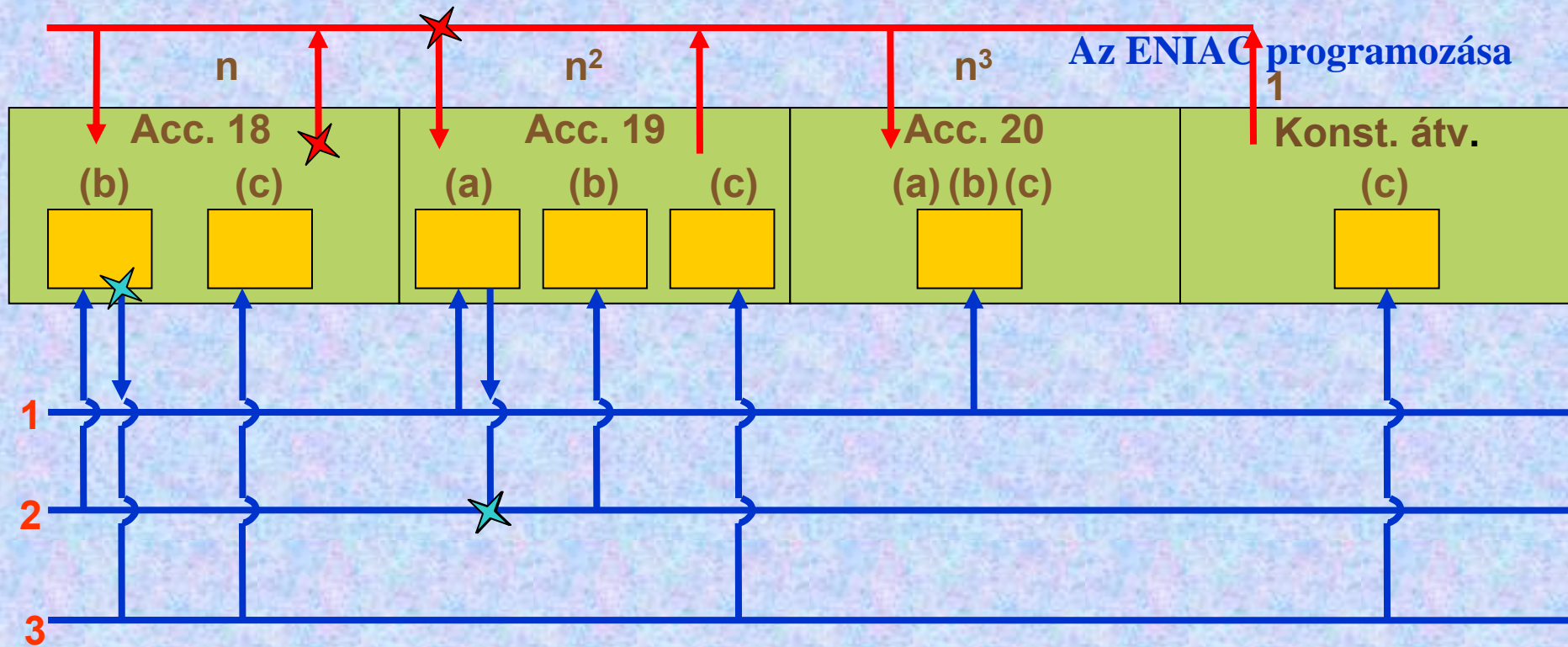
*End...*  
**LGy.**



a) Acc 19 háromszor küld, Acc 20 háromszor fogad

b) Acc 18 háromszor küld, Acc 19 kétszer fogad, Acc 20 háromszor fogad

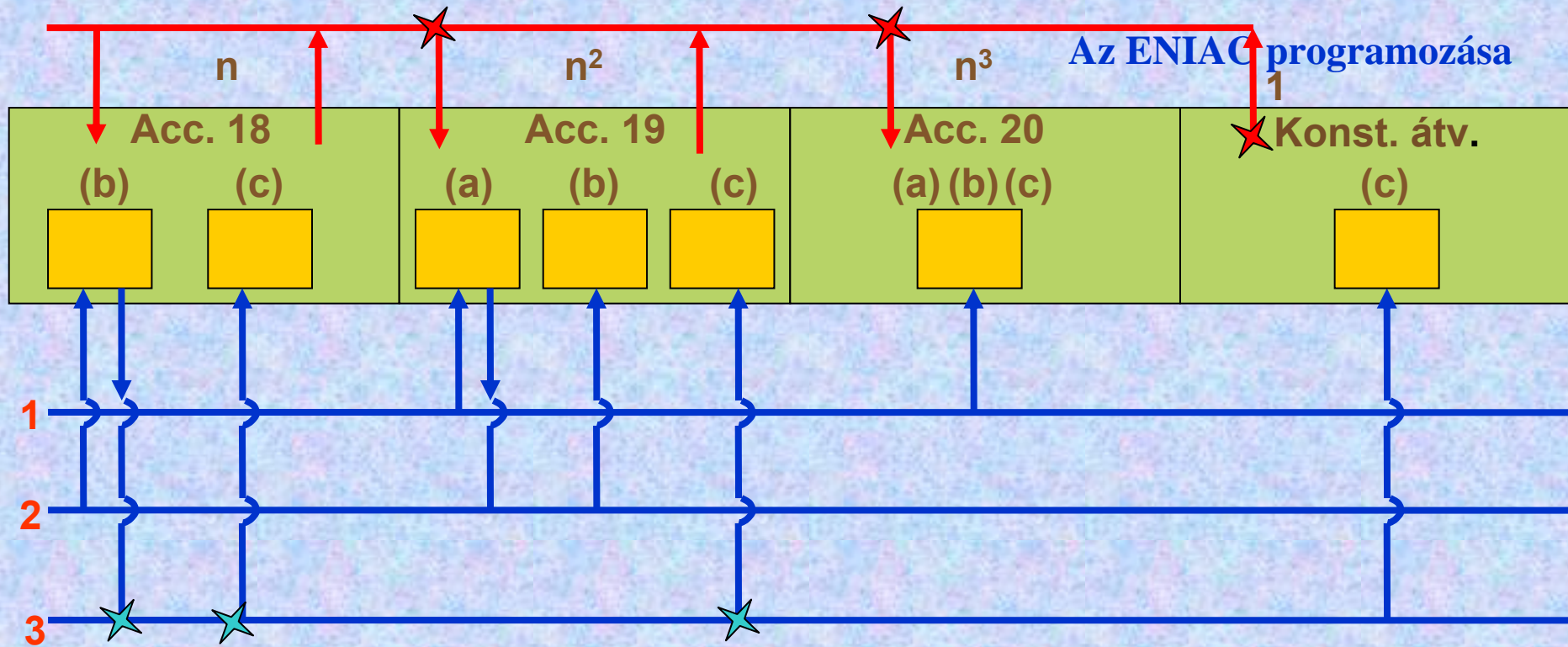
c) Konst. átv. egyszer küld, Acc 18, Acc 19, Acc 20 egyszer fogad



a) Acc 19 háromszor küld, Acc 20 háromszor fogad

b) Acc 18 háromszor küld, Acc 19 kétszer fogad, Acc 20 háromszor fogad

c) Konst. átv. egyszer küld, Acc 18, Acc 19, Acc 20 egyszer fogad

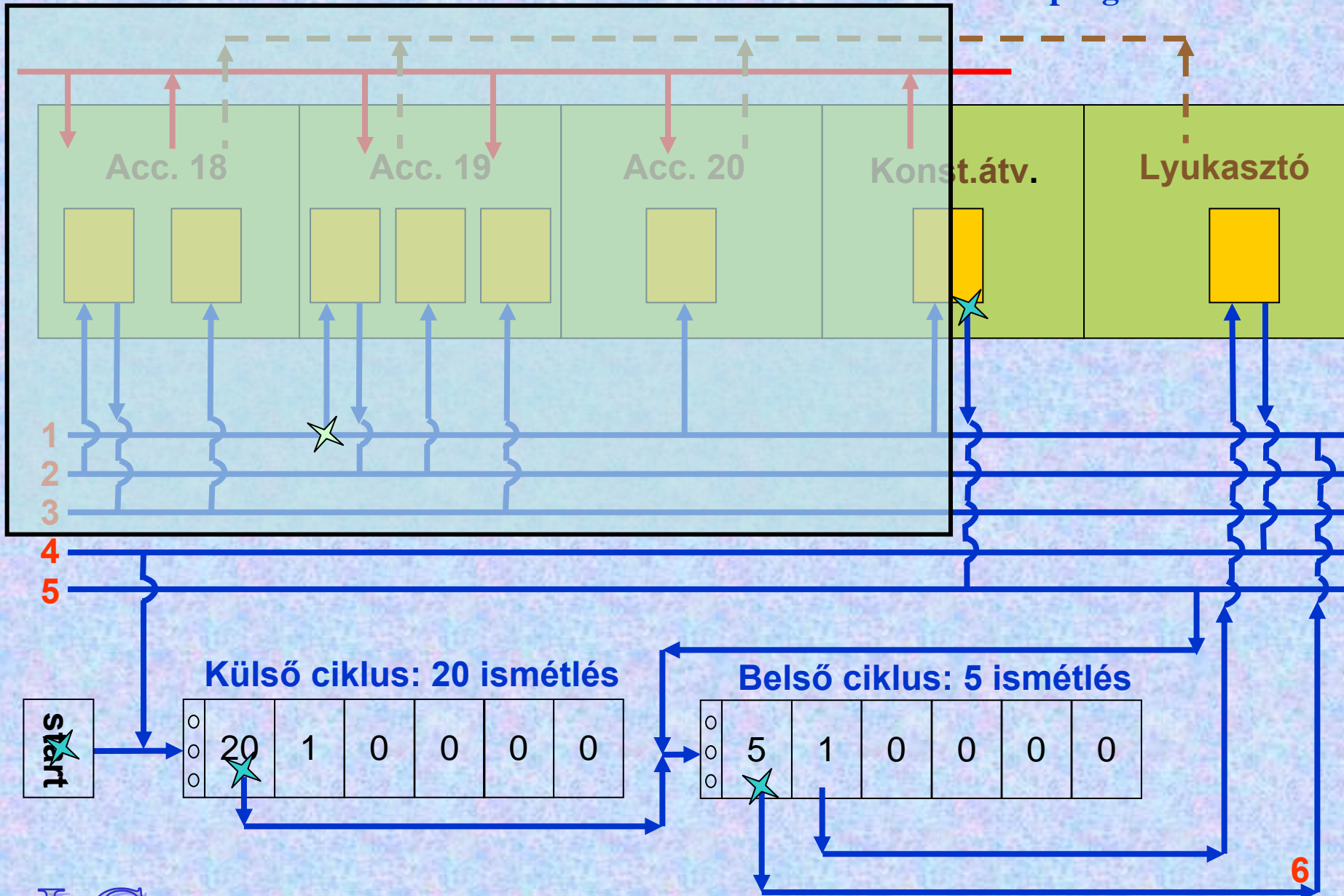


a) Acc 19 háromszor küld, Acc 20 háromszor fogad

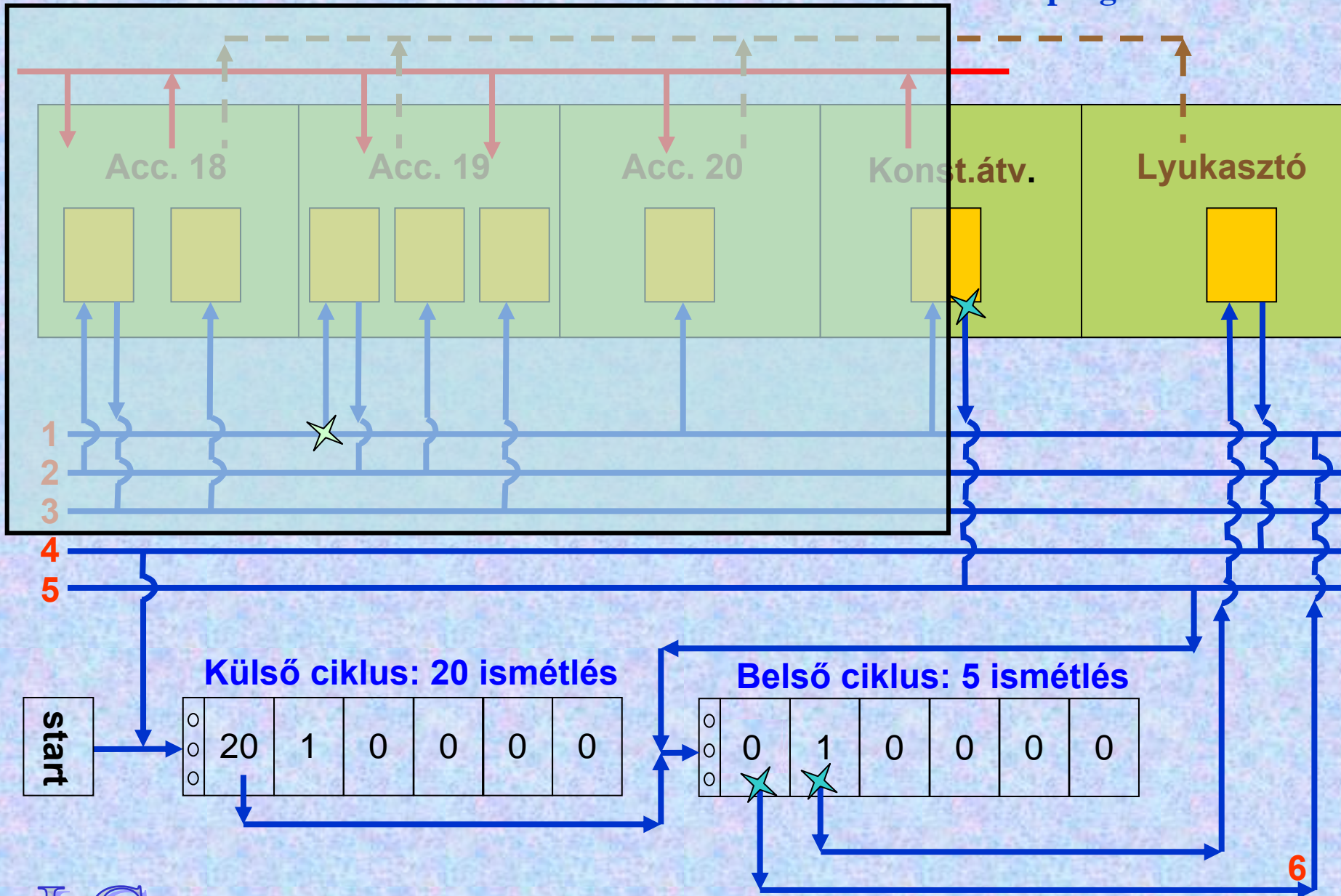
b) Acc 18 háromszor küld, Acc 19 kétszer fogad, Acc 20 háromszor fogad

c) Konst. átv. egyszer küld, Acc 18, Acc 19, Acc 20 egyszer fogad

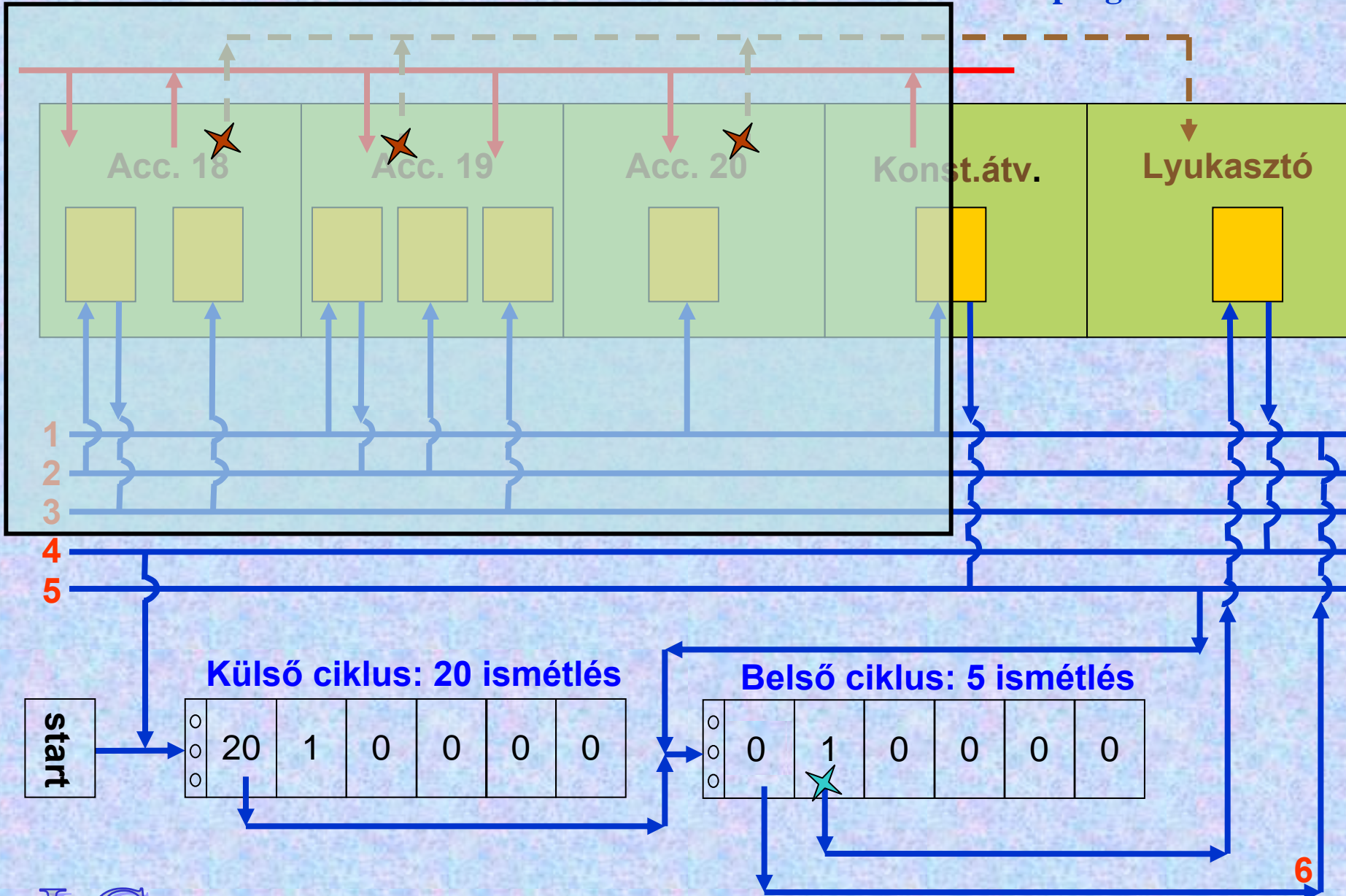
# Az ENIAC programozása



# Az ENIAC programozása

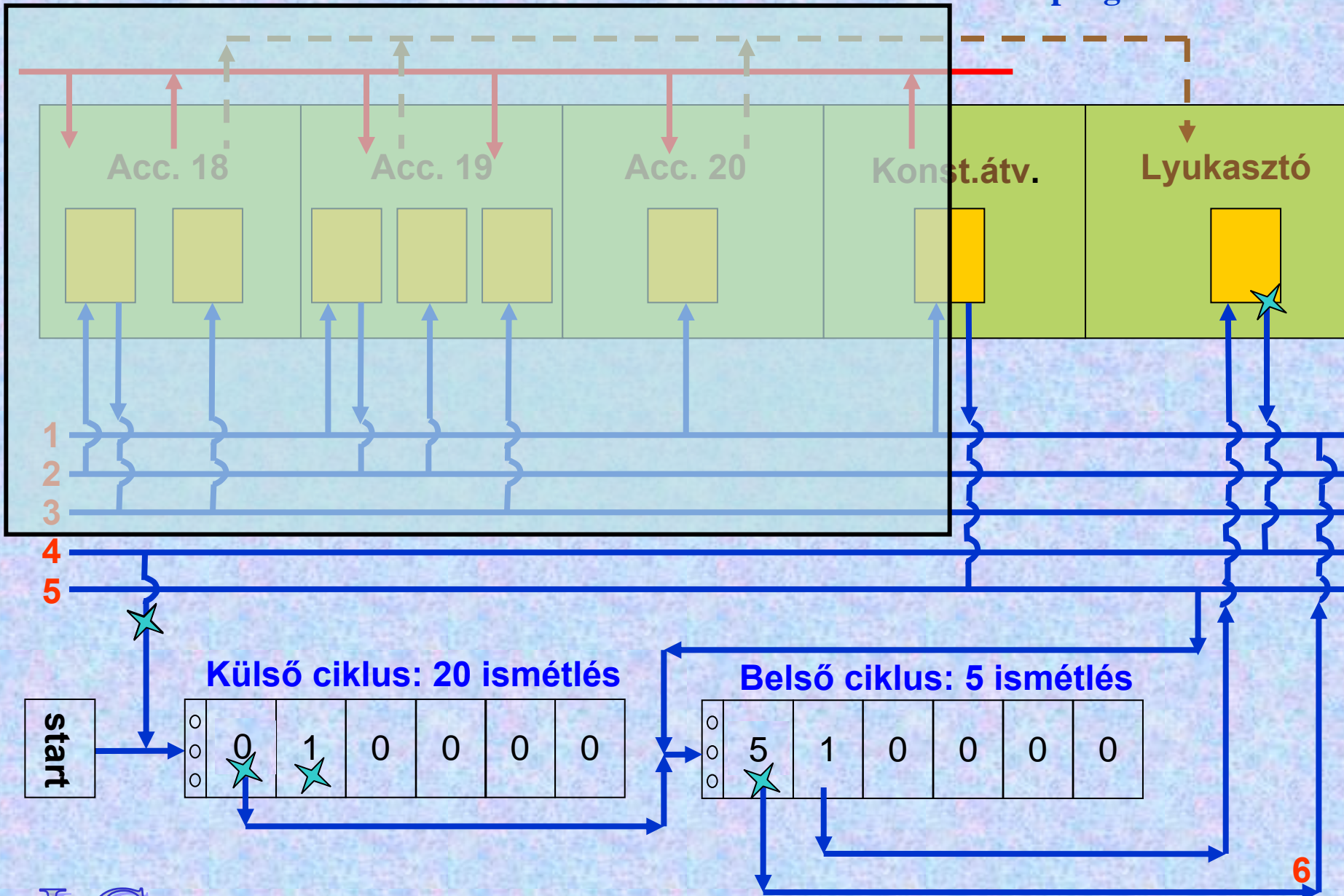


# Az ENIAC programozása





# Az ENIAC programozása



## Az ENIAC-történet vége, és a későbbi fejlemények

- Az ENIAC a semmitől a működőképes gépig  $2\frac{1}{2}$  év alatt jutott el
- Számos tervezői döntést a sietség (II. világháború) motivált
- Megbízhatósága meghaladta a 90%-ot
- Egy évig működött Philadelphiában; ezalatt Los Alamos részére végzett számításokat, majd Aberdeenbe költöztették. Itt működött 1955 október 2.-ig
- Tervezői 1943-44-től kezdve az EDVAC tervein dolgoztak
- Később Mauchly és Eckert saját céget alapított, melyet idővel a Remington felvásárolt, és amely ennek elektronikus számítógép részlege lett = **UNIVAC számítógépek**

# Az ENIAC műszaki adatai

- **Összeadás:** 200  $\mu$ sec
- **Szorzás:** 2,8 msec
- **Osztás:** 2,4 msec
- **Beolvasás:** 125 kártya/perc  
(1 kártya= 8 db 10 jegyű szám + előjelek)
- **Kártyalyukasztás:** 100 kártya/perc
- **Elektroncsövek száma:** 17468 db
- **Energia:** 174 kW
- **Alapterület:** ~180m<sup>2</sup>
- **Kiszolgáló személyzet:** 6 fő/műszak
- **Hatásfok (hasznos idő/össz. idő):**  $\cong 70\%$

# A műveleti idők összehasonlítása

Gép \ Művelet	Összeadás	Szorzás	Osztás
Analitikai gép	1 sec	60 sec	60 sec
	<i>(50 decimális jegyre)</i>		
Z3	0,3 sec	4,5 sec	—
	<i>(22 bitre)</i>		
MARK-1	0,3 sec	5,7 sec	15,3 sec
	<i>(23 decimális jegyre)</i>		
ENIAC	0,2 msec	2,8 msec	6,0 msec
	<i>(10 decimális jegyre)</i>		

# 6. fejezet

**Az első Neumann-elvű  
számítógép: az *EDVAC***

**EDVAC = Electronic Discrete Variable  
Automatic Computer**

## Neumann János (1903-1957)



- 1903. dec. 28. –án született Budapesten. Apja bankár
- 1914-ig magánúton tanul, utána gimnáziumba kerül  
Tanárai:  
*Kürschák, Rátz, Fekete (egyéni tanul)*
- 1921: *érettségizik (ekkor már „profi” matematikusnak számít)*

## Neumann János (1903-1957)



- 1921-23: **Berlinben tanul**
- 1924-26: **Két egyetemre jár**  
Budapest  
*(matematika szak)*  
Zürich:  
Technische Hochschule  
*(vegyésszmérnök, 1924-26)*
- **Kb. egy időben két doktorátust szerez:**  
Budapest: *1926. márc.*,  
Zürich: *1926. okt.*
- 1927: **A berlini egyetemen magántanári állást kap**

## Neumann János (1903-1957)



- 1929: Átmegy a hamburgi egyetemre
- 1930: Látogatást tesz Princetonban, majd 1 éves szerződést kap
- 1938-54: Princetonban dolgozik  
(számos katonai projectben is részt vesz [IAS])
- 1944. szept.: Első látogatása a Moore-School-ban. Csatlakozik a fejlesztői csoporthoz
- 1955: megbetegszik
- 1957 febr. 8.: meghal



# Érdeklődési területek

- **Algebra, halmazelmélet**
- **Valós függvénytan, mértékelmélet**
- **Topológia**
- **Absztrakt terek, csoportelmélet**
- **Hálóelmélet**
- **Folytonos geometriák**
- **Kvantummechanika**
- **Játékelmélet**
- **Meteorológia, hidrodinamika és aerodinamika**
- **Számítástechnika**
- **Automataelmélet**
- **Valószínűségi logika**
- **Matematikai fizika**

## Az EDVAC

- 1944: Eckert és Mauchly új gép tervezésébe kezd.

A tervezők célja az volt, hogy korrigálják az **ENIAC** egyes kevésbé szerencsés megoldásait  
(*pl. a tárolási módok egységesítése*):

– ENIAC: többféle tárolási mód

- akkumulátorok (*elektroncsöves számlálók*)
- függvénytáblázatok  
(*kapcsolókkal beállított „read only” memória*)
- központi programadó (*kapcsolók+számlálók*)
- külső tároló (*lyukkártya*)

## Az EDVAC

- 1945: Neumann jelentése az *EDVAC*-ról.
- 1949: A még nem működőképes gépet Aberdeenbe (*BRL*) szállítják
- 1951-1962: A gép működik, bár az eredeti tervező team felbomlott  
(*Eckert , Mauchly és mások a „versenyszférába” távoztak, Neumann visszatért az IAS-be*)

## MEGJEGYZÉS:

Az EDVAC, abban a formában, ahogy Neumann a Jelentésben leírta, *sohasem épült meg!*

Az, ami elkészült, a Neumann távozása utáni áttervezések következtében *jelentősen különbözik* az általa elképzelt géptől.

<b>JELLEMZŐK</b>	<b>...ahogy Neumann megtervezte...</b>	<b>... és ahogyan megépült...</b>
<b>Időzítés</b>	<b>Szinkron</b>	<b>Szinkron</b>
<b>Működés</b>	<b>Szekvenciális</b>	<b>Szekvenciális</b>
<b>Számrendszer</b>	<b>Bináris</b>	<b>Bináris</b>
<b>Szóhossz (bit)</b>	<b>32</b>	<b>44</b>
<b>Műveleti kód (bit)</b>	<b>3+5</b>	<b>4</b>
<b>Címrész (bit)</b>	<b>13</b>	<b>10</b>
<b>Utasítás/szó</b>	<b>1</b>	<b>1</b>
<b>Aritmetika</b>	<b>Fixpontos</b>	<b>Fixpontos</b>
<b>Számábrázolás</b>	<b>30 bit+előjel</b>	<b>43 bit+előjel</b>
<b>Tároló</b>	<b>8192 szó/32 bit</b>	<b>1024 szó/44 bit</b>
<b>Háttértár</b>	<b>Nincs</b>	<b>Mágnesdob</b>

# Neumann János (1945): Jelentéstervezet az EDVAC gépről (First Draft of a report on the EDVAC) <http://wps.com/projects/EDVAC/index.html>

## 1) Definíciók:

- A számítógép nagy bonyolultságú feladatok elvégzésére alkalmas, bonyolult eszköz
- A feladatok megoldásához szükséges információkat a gépnek – számára felismerhető módon – előre meg kell adni
- A feladat megoldása közben a gép emberi beavatkozás nélkül működik
- A feladat elvégzése után az eredményeket a felhasználó számára meg kell jeleníteni
- A gépnek hibátlanul kell működnie. Képes lehet a gyakran előforduló hibák automatikus felismerésére és

## 2) A gép fő egységei:

- Az aritmetikai műveleteket elvégző egység. Célszerű, hogy az alpműveletek elvégzésére (+, -, ×, ÷) külön részegységek álljanak rendelkezésre, de ez, műszaki megfontolásokból bővíthet vagy szűkülhet.

*Ezt az egységet Neumann CA-val jelöli*

- A feladat megoldásához a felhasználó által megadott utasításokat végrehajtó egység.

*Ezt az egységet Neuman központi vezérlő egységnek nevezi, és CC-vel jelöli. (A CA és CC együttes jelölése Neumannál: C)*

- A feladat megoldásában szereplő adatok tárolására szolgáló memória, melyet Neumann M-mel jelöl. A C és M közti adatforgalmat a gép belső részegységei végzik

- A gép külső kapcsolatainak (*input és output*) működtetéséhez szükséges adathordozó, ennek jele R

- Az információnak az R-ből a gép belső részegységeihez való továbbítását ellátó egység, ennek jele I (*input*).

*(Neumann szerint célszerűbbnek látszik az R-ből mindig az M-be továbbítani az információt, nem közvetlenül a C-be)*

- A számítás eredményeit az R-be továbbító egység, melynek jele O (*output*)

### 3) A tárgyalás menete

- Az anyag bonyolultsága folytán, nem lehetséges a gép egyes egységeit sorrendben, egyetlen menetben áttekinteni. A Jelentés a részegységeket, azok összefüggéseire való tekintettel, több menetben tárgyalja

### 4) A neuronhálózati modell

- A gépet a legcélszerűbb *elektroncsövekből* felépíteni
- Az elektronikus elemekből felépített hálózatok viselkedése jól leírható *McCulloch* és *Pitts* neuronhálózati modelljével
- A modellnek megfelelően, az egyes hálózati elemeknek kétféle állapota lehetséges: *nyugalmi* és *ingerelt*
- Az egyes elemek az őket ért ingerek hatására - adott késleltetéssel – újabb ingerületeket bocsáthatnak ki, amennyiben ezt egy korábban érzékelt inger nem tiltotta le

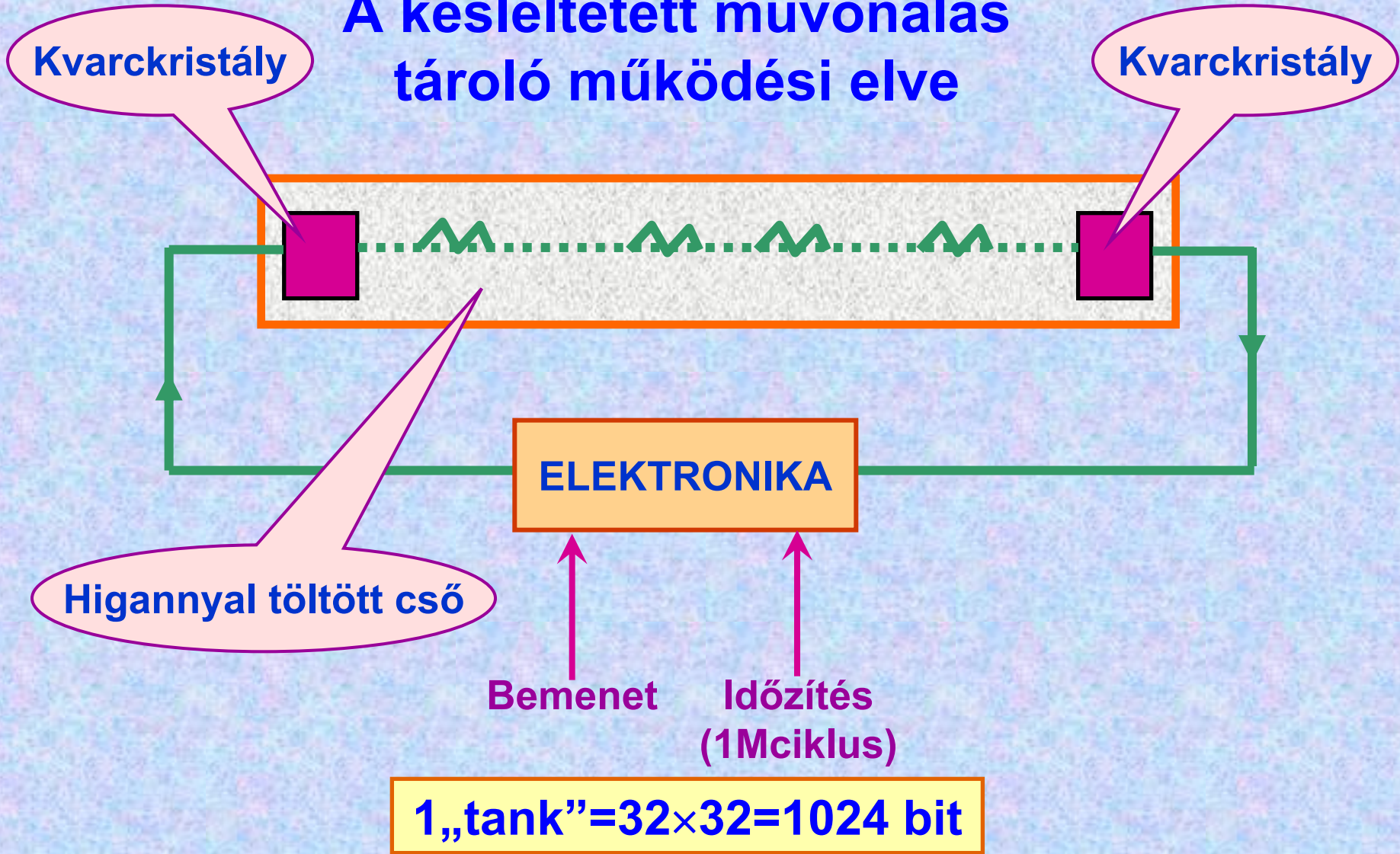


**Neumann JELENTÉSének hatására vált külön a számítógépek logikai és áramköri tervezése.**

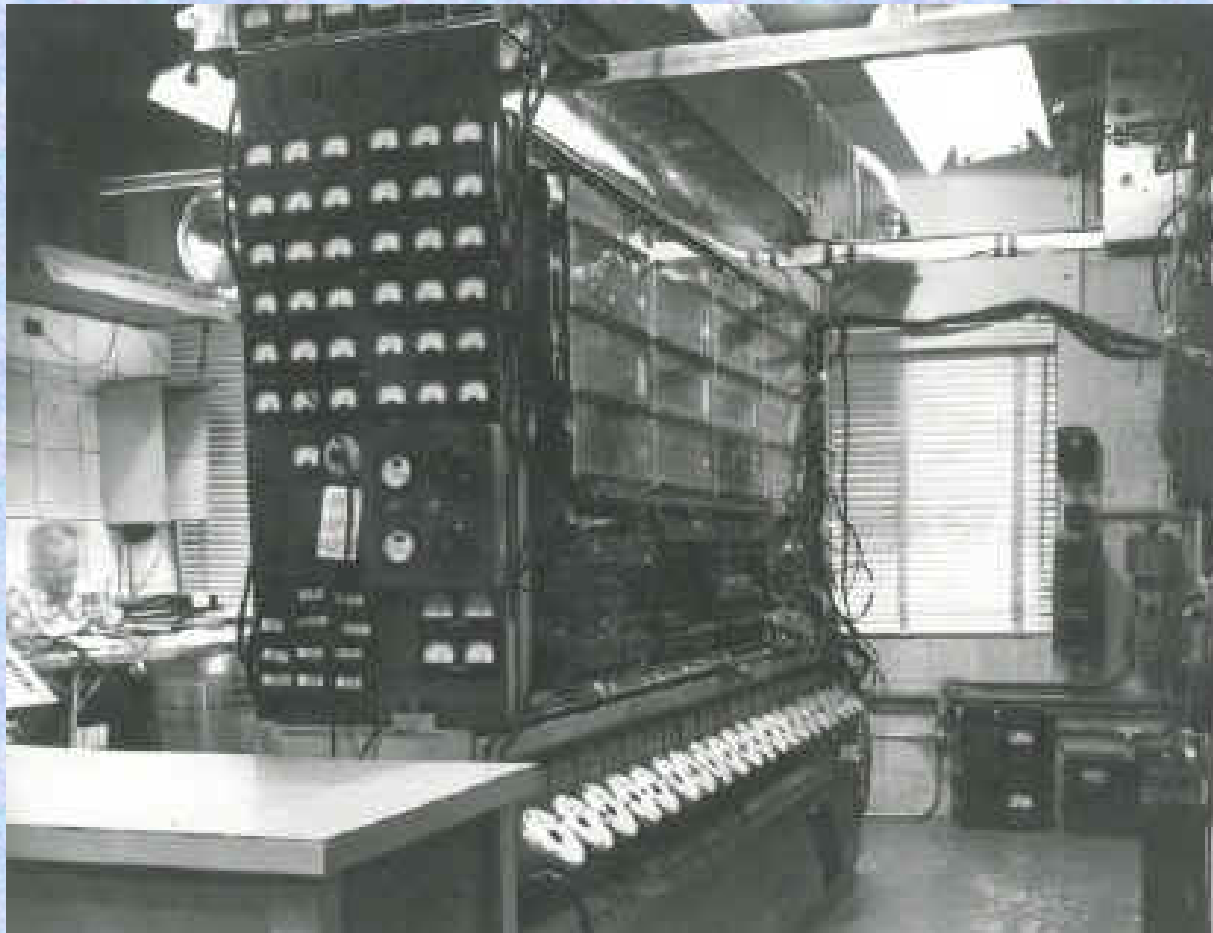
## Alternatívák az adattároló megvalósítására

- **késleltetett művonalas tároló**  
*(ld. a következő animációt)*
- **elektrosztatikus tároló**
  - alapja egy katódsugárcső, amelyre felvihetők az információ bitei. Az egyes biteket a TV működéséhez hasonlóan lehet leolvasni, felvitelük pedig szintén katódsugárral történik  
*(Ez a tárolási mód a későbbiekben nem terjedt el szélesebb körben)*
- **mágneselem v. lemez**
  - ez terjedt el legjobban, a '80-as, '90-es évekig használták

# A késleltetett művonalas tároló működési elve

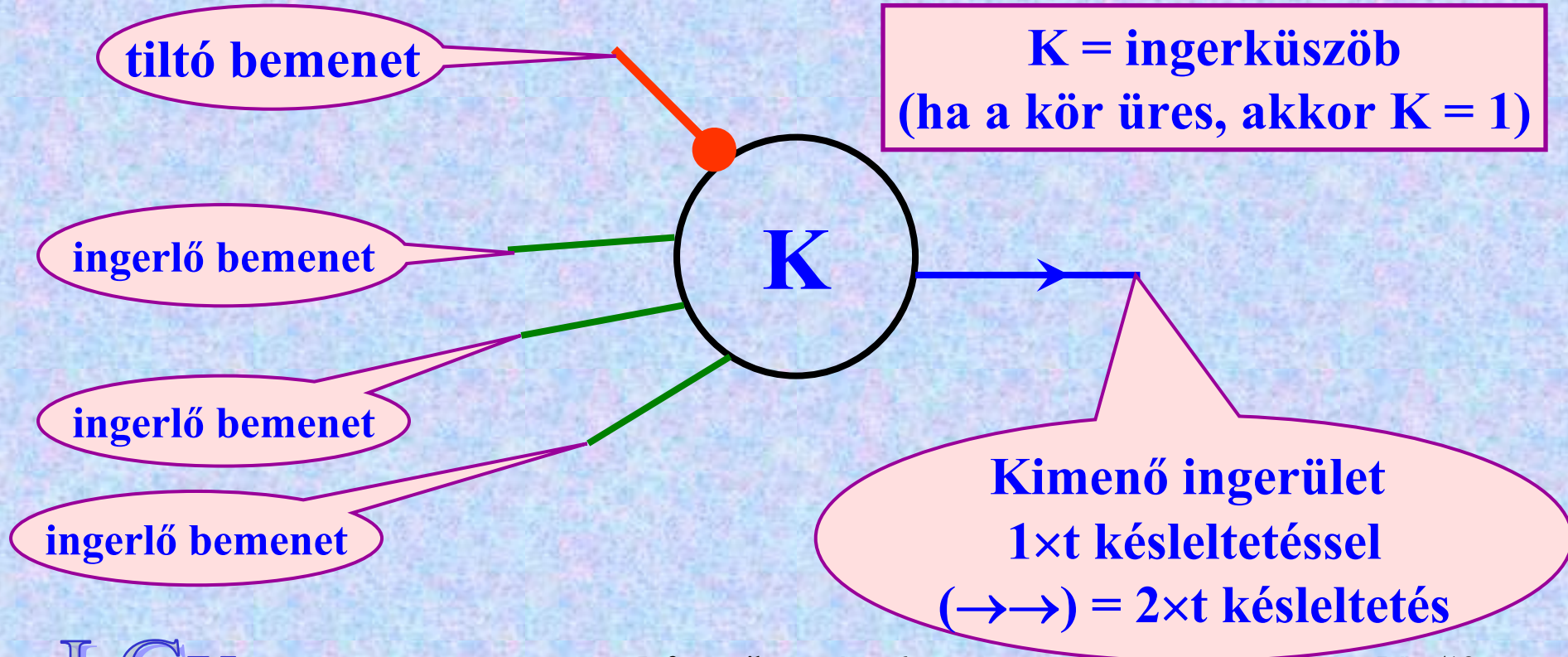


# Az EDVAC a késleltetett művonalas tárolóval

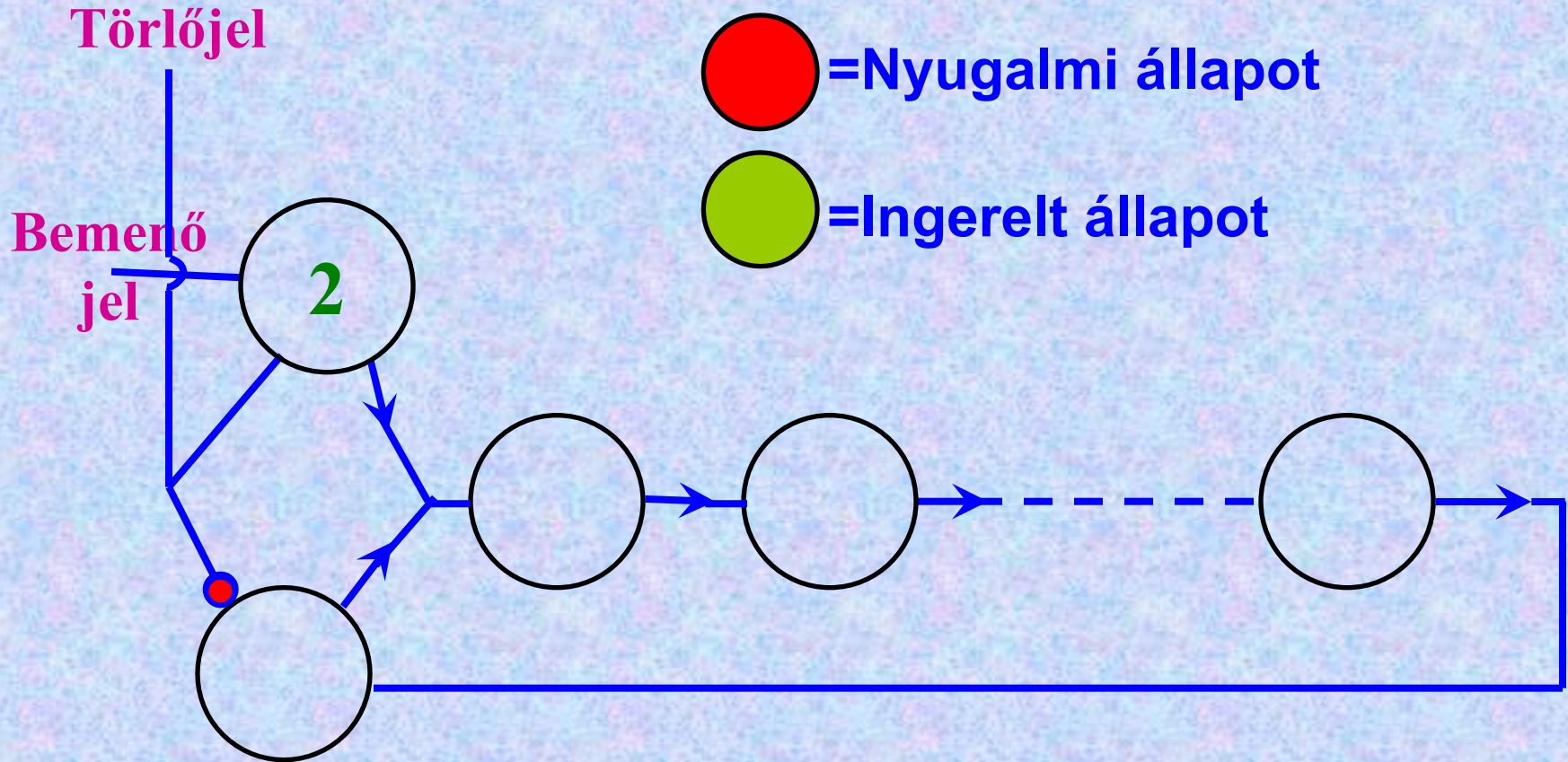


# A neuron-analógia alkalmazása a logikai tervezésben

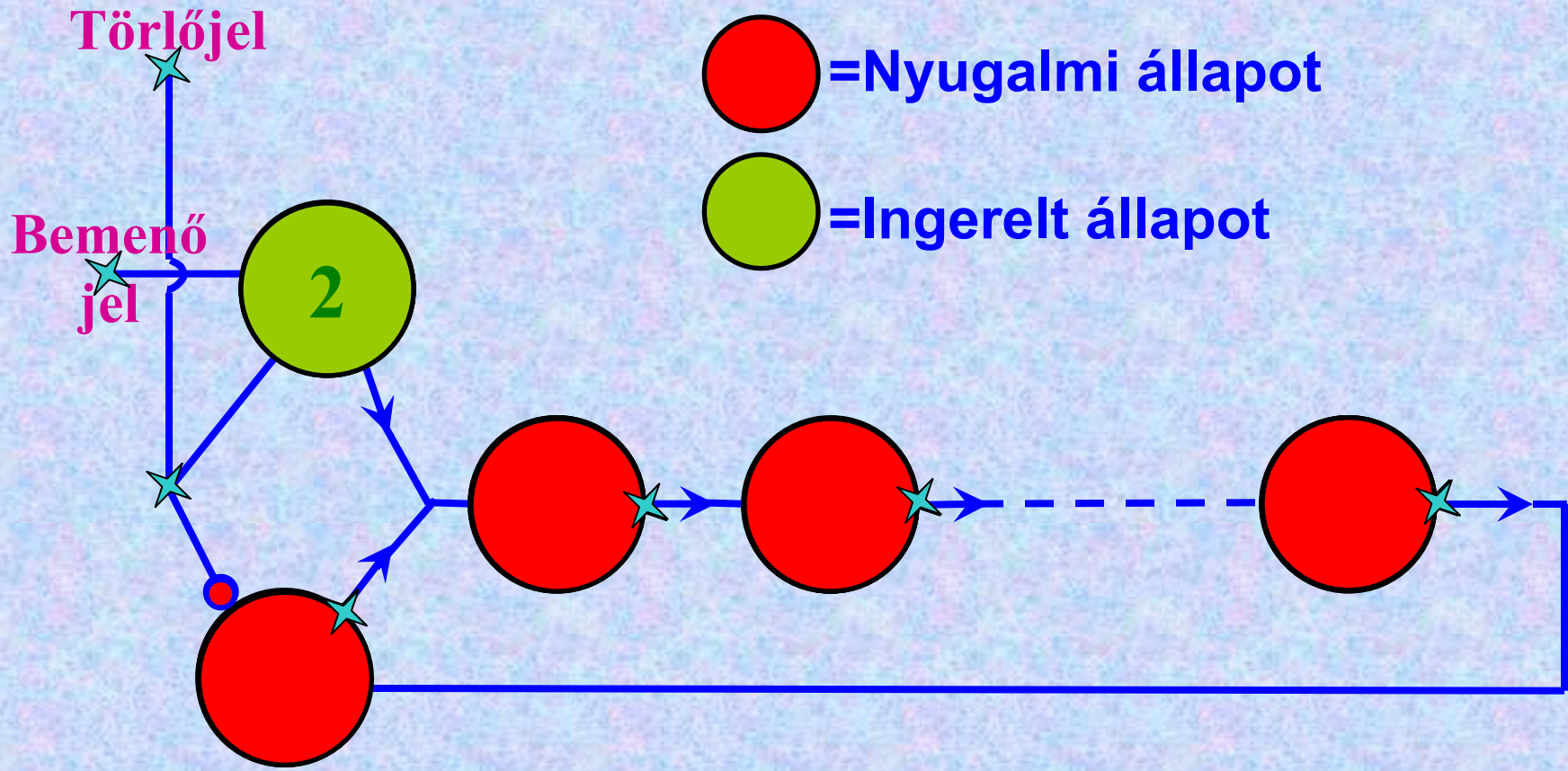
Neuron=kétállapotú elem: nyugalmi/ingerelt



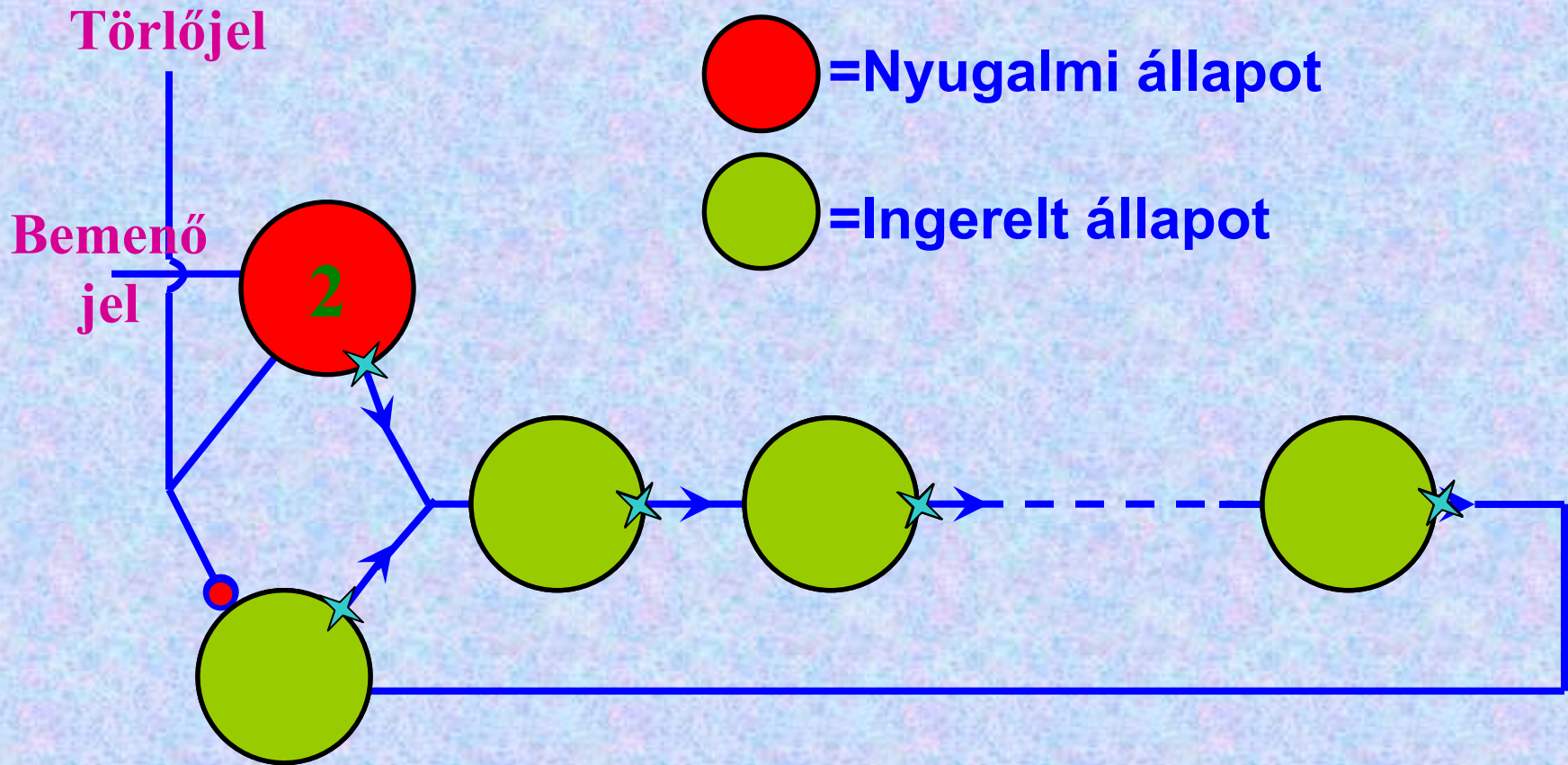
# A késleltetett művonalas tároló működése a neuron analógia szerint



# A késleltetett művonalas tároló működése a neuron analógia szerint



# A késleltetett művonalas tároló működése a neuron analógia szerint

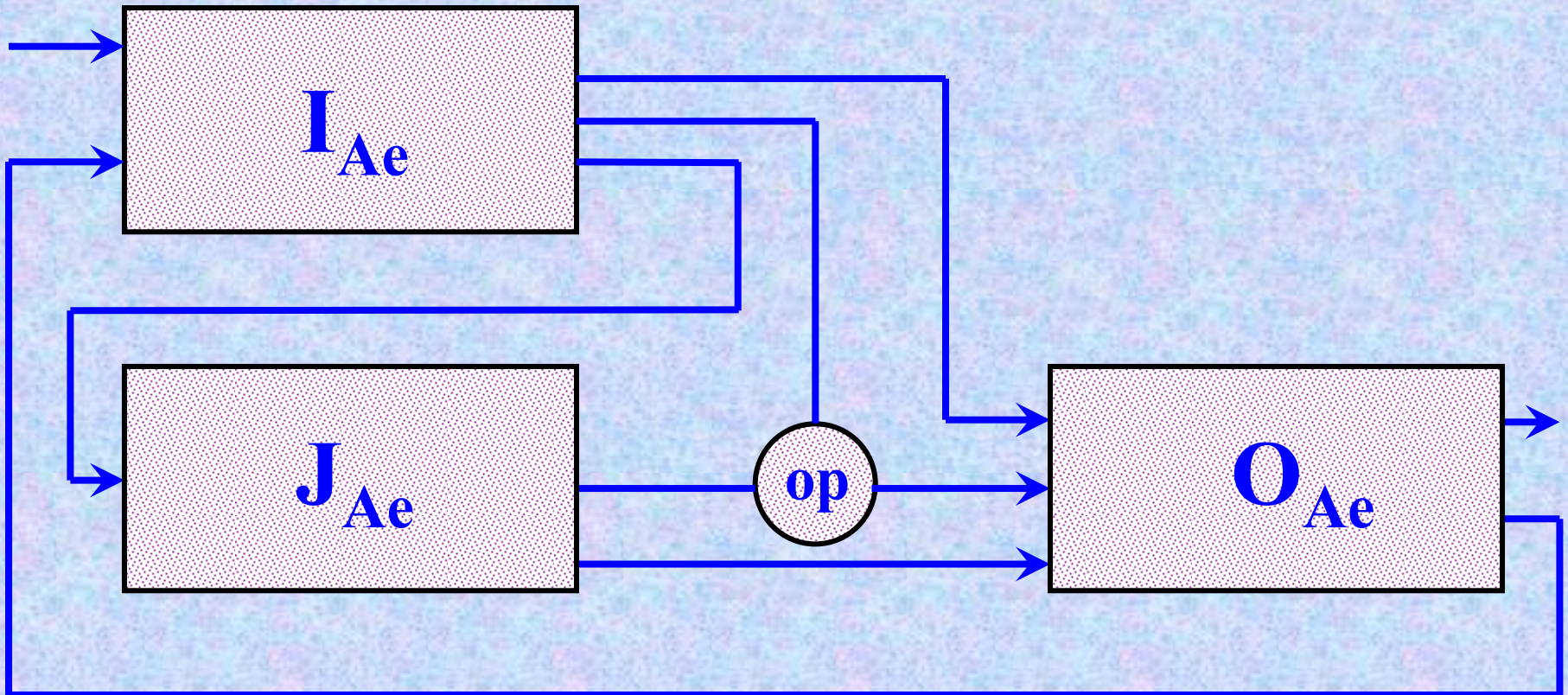




## Fogalom- és jelölés magyarázat

<b>FIRST DRAFT</b>	<b>MAI SZÓHSZNÁLAT</b>
<b>Unit</b>	<b>Byte</b>
<b>Minor cycle</b>	<b>32 bites memóriaszó</b>
<b>Major cycle</b>	<b>32 szavas memóriaegység, szegmens</b>
<b>M[<math>\mu</math>,<math>\rho</math>]</b>	<b>Memóriacímzés: szegmens+offset</b>

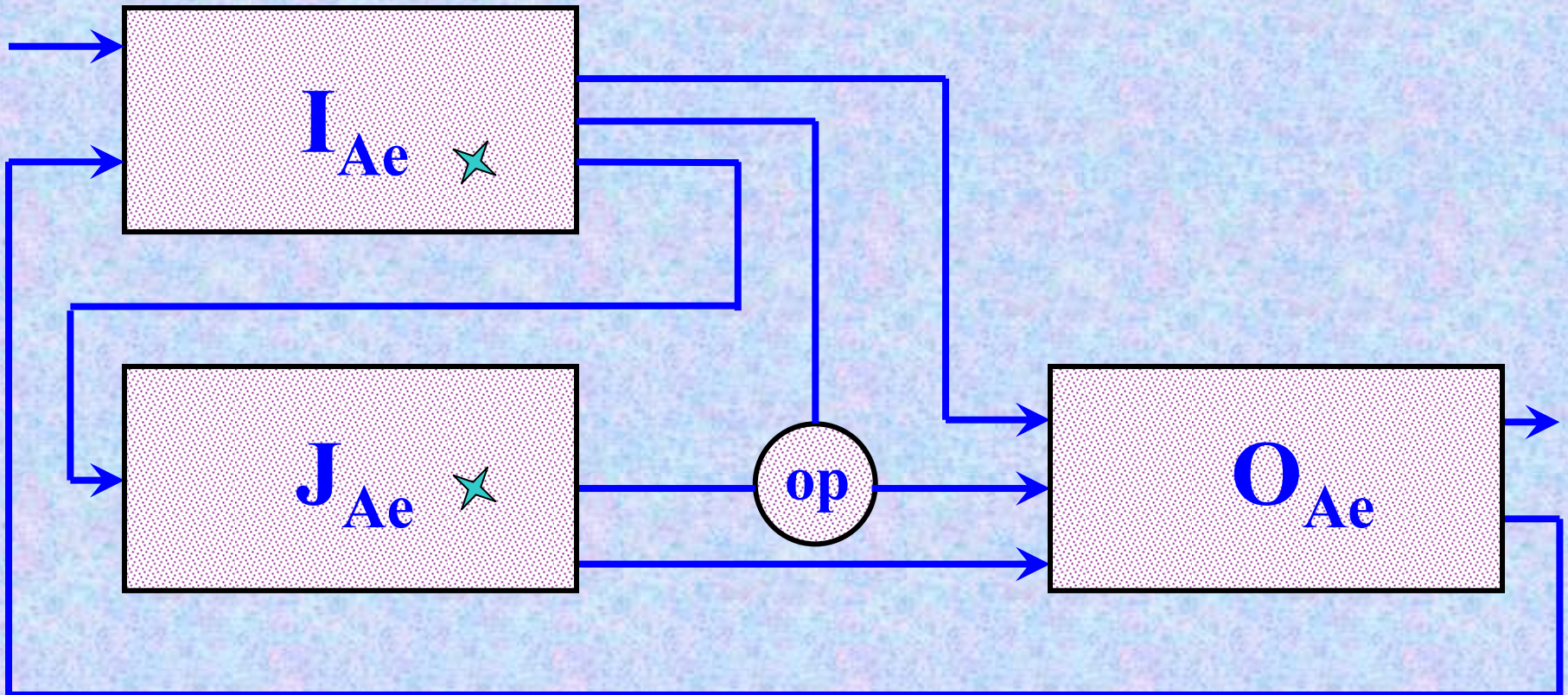
## Az aritmetikai egység



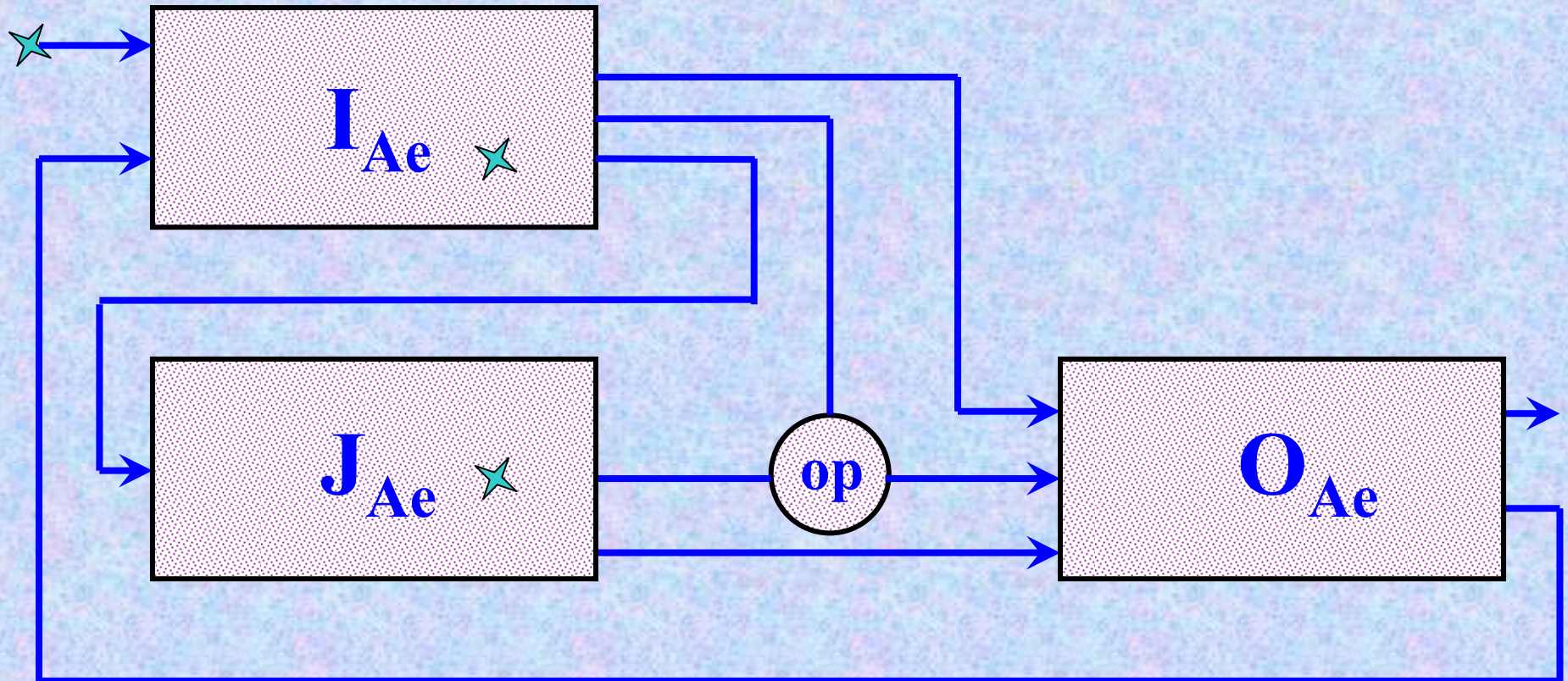
## Az aritmetikai egység funkciói

- Alapműveletek + négyzetgyökvonás
- A két regiszter közül bármelyik tartalmának átküldése az output regiszterbe
- Az output regiszter előjelétől függően, a két regiszter közül valamelyik tartalmának átküldése az output regiszterbe
- Decimális-bináris ill. bináris-decimális konverzió

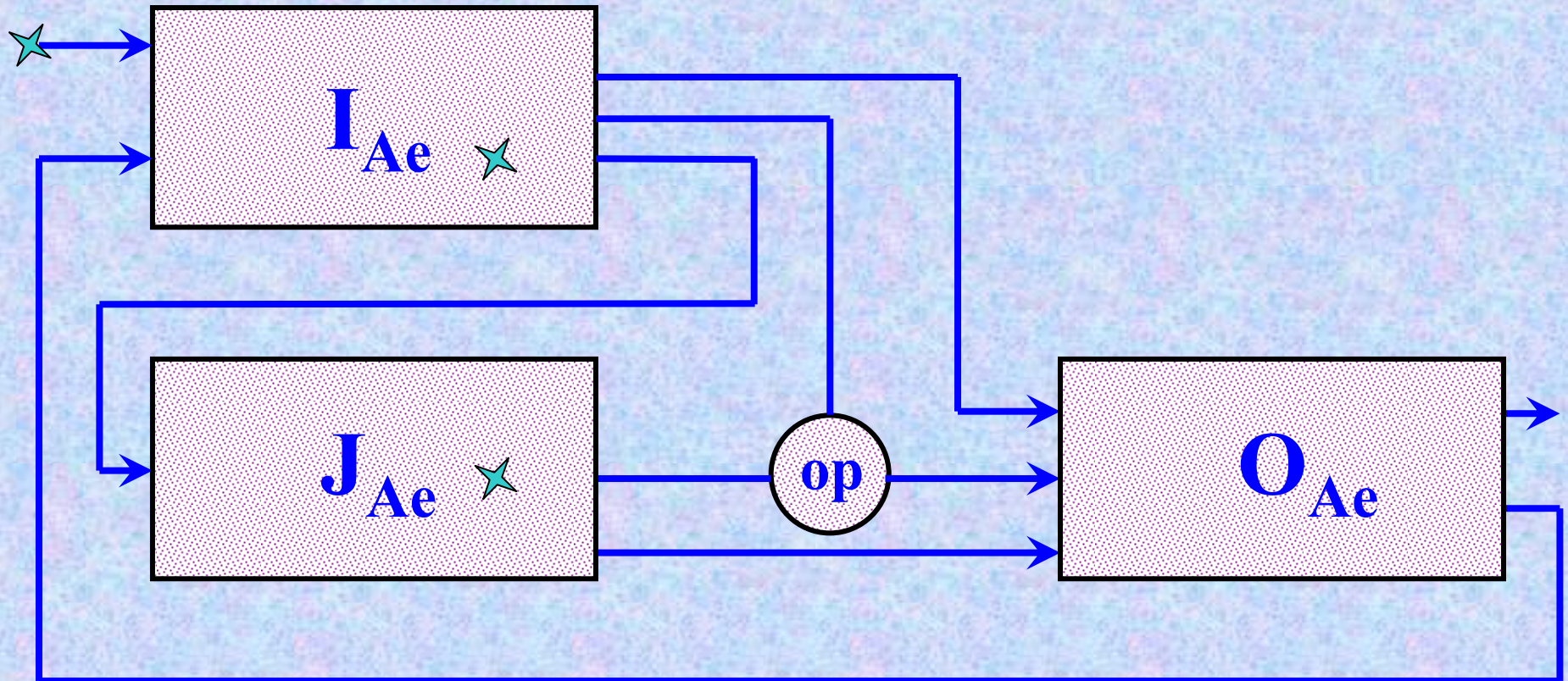
# A kétoperandusú műveletek végrehajtása az aritmetikai egységben



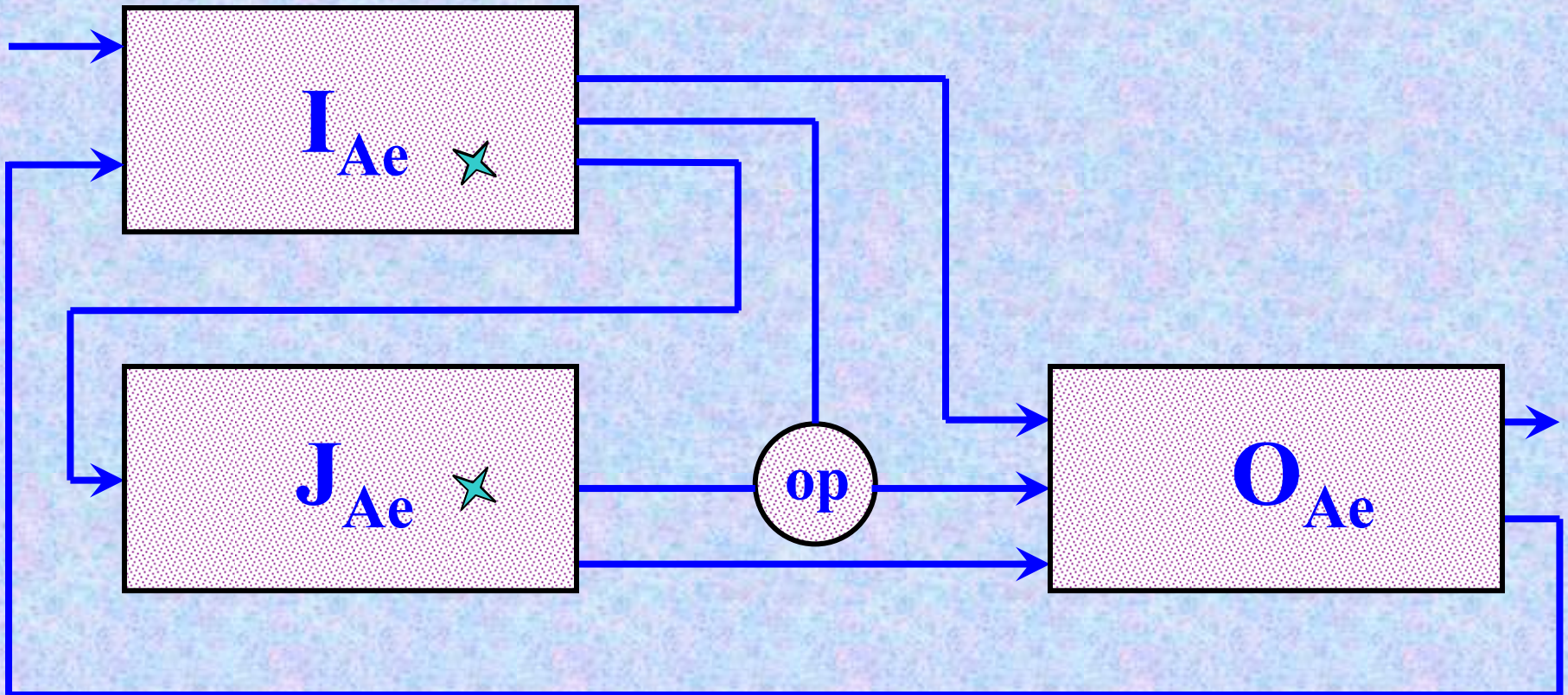
# A kétoperandusú műveletek végrehajtása az aritmetikai egységben



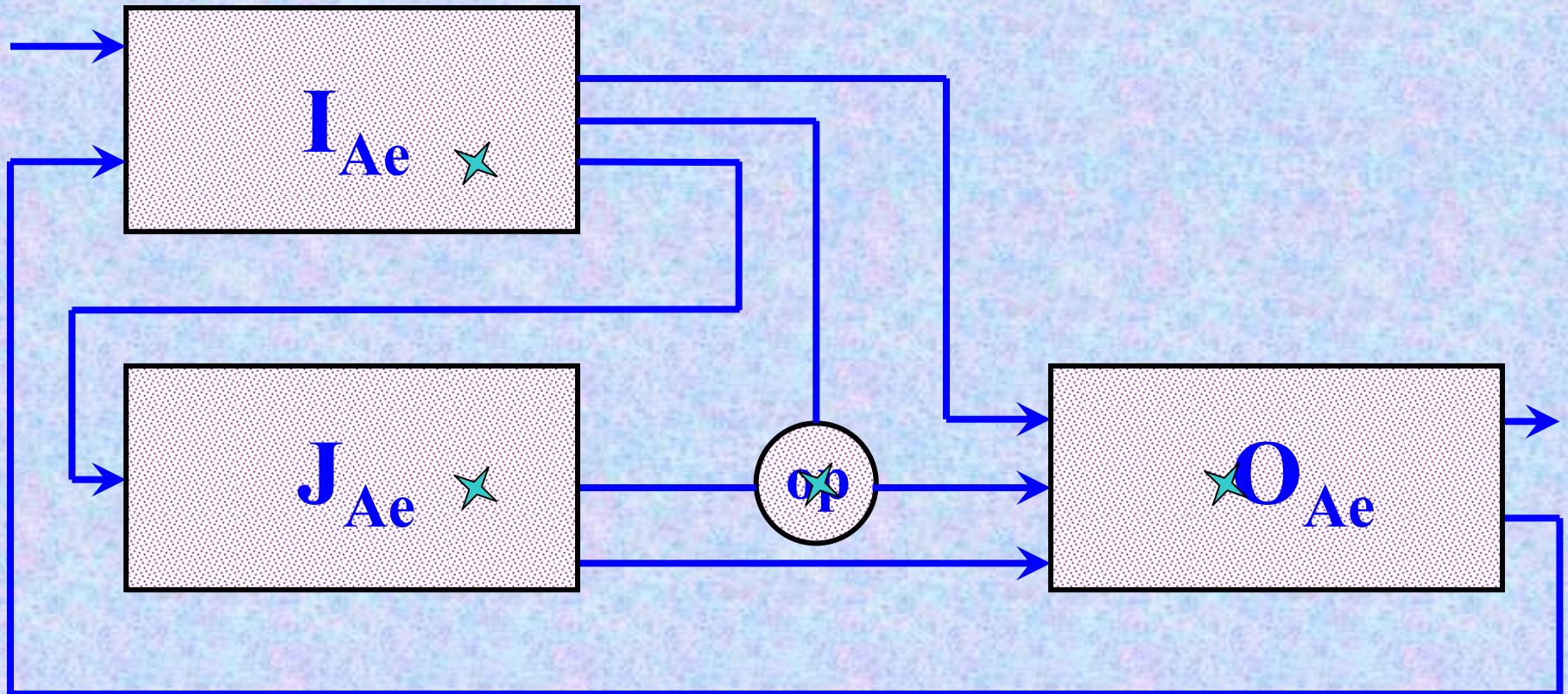
# A kétoperandusú műveletek végrehajtása az aritmetikai egységben



# A kétoperandusú műveletek végrehajtása az aritmetikai egységben



# A kétoperandusú műveletek végrehajtása az aritmetikai egységben

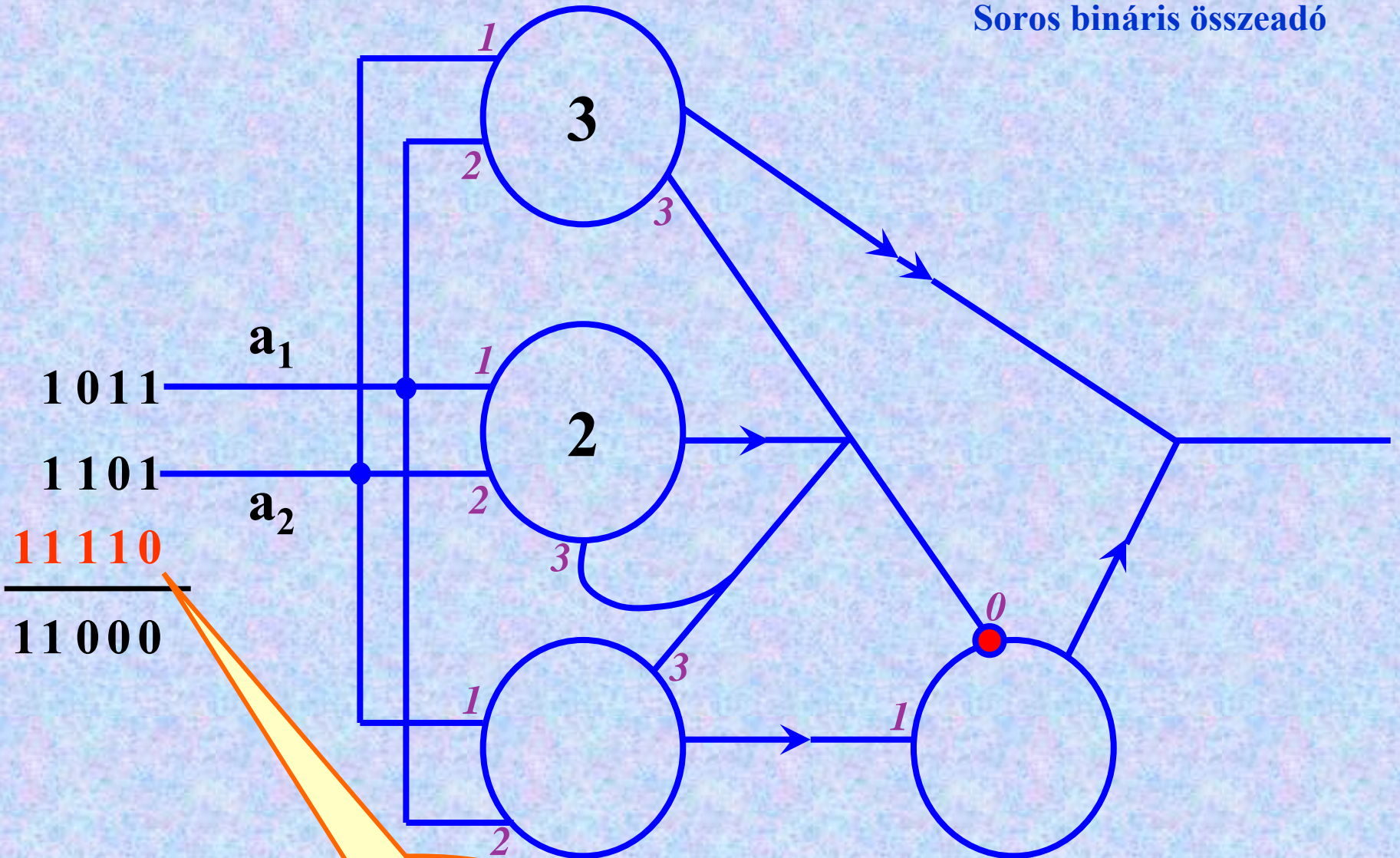




## Soros bináris összeadó

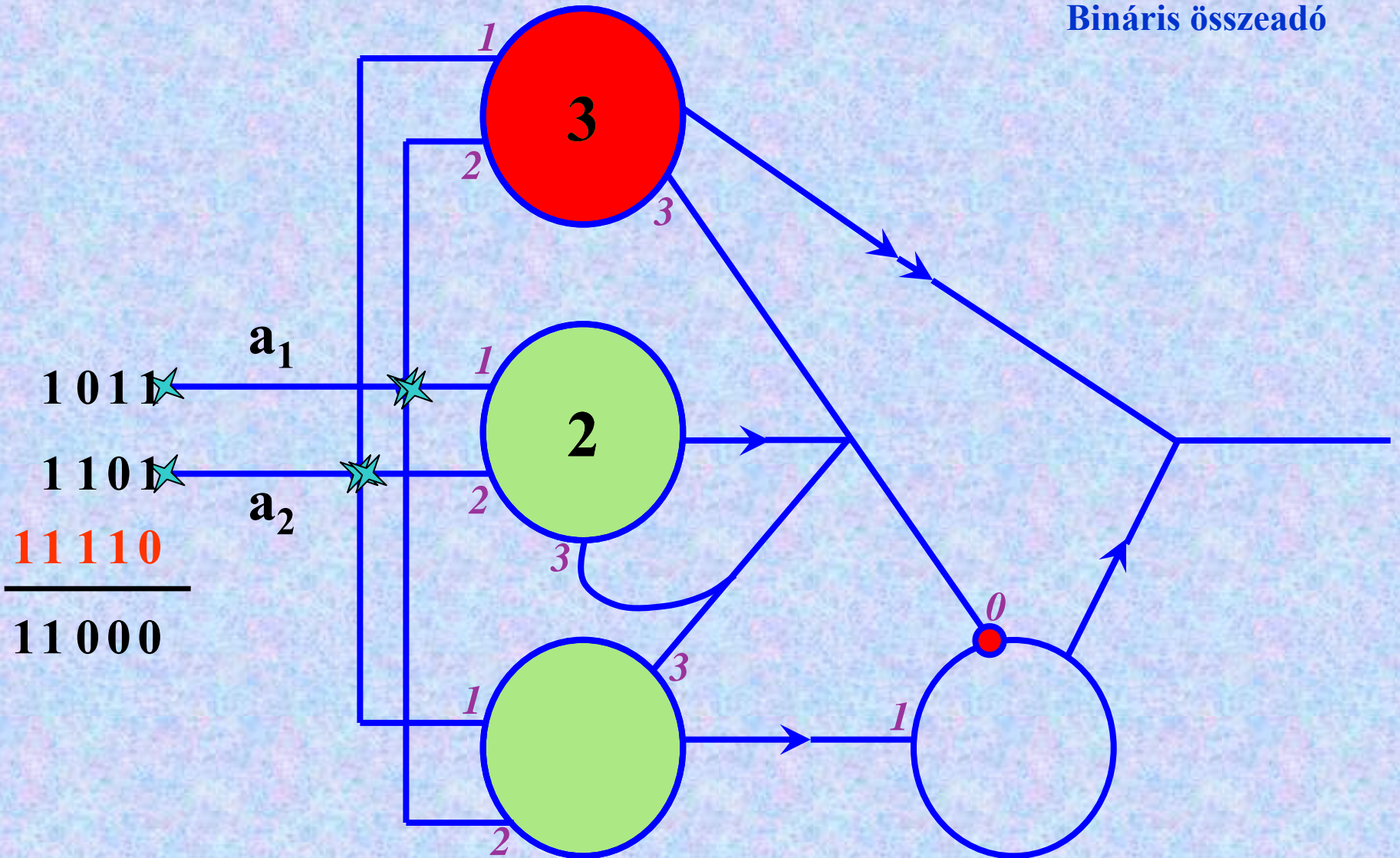
- Funkciója: a gép működésében előforduló összeadási műveletek elvégzése
- Felépítése: négy neuron alkotja, amelyek a következő diákon látható módon vannak összekapcsolva

# Soros bináris összeadó

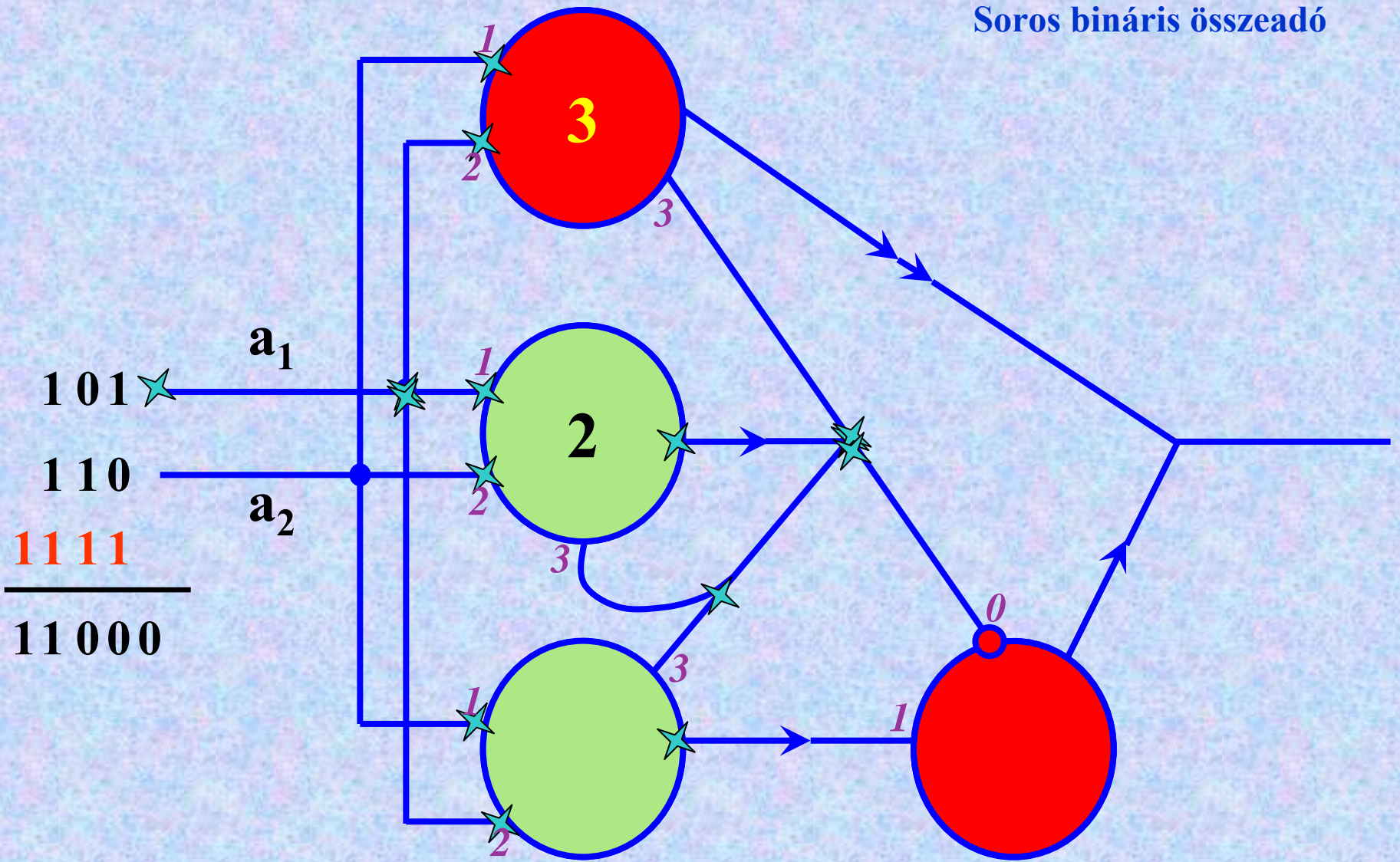


**1 1 1 1 0**  
-----  
**1 1 0 0 0**

**Átvitelek**



1. ütem

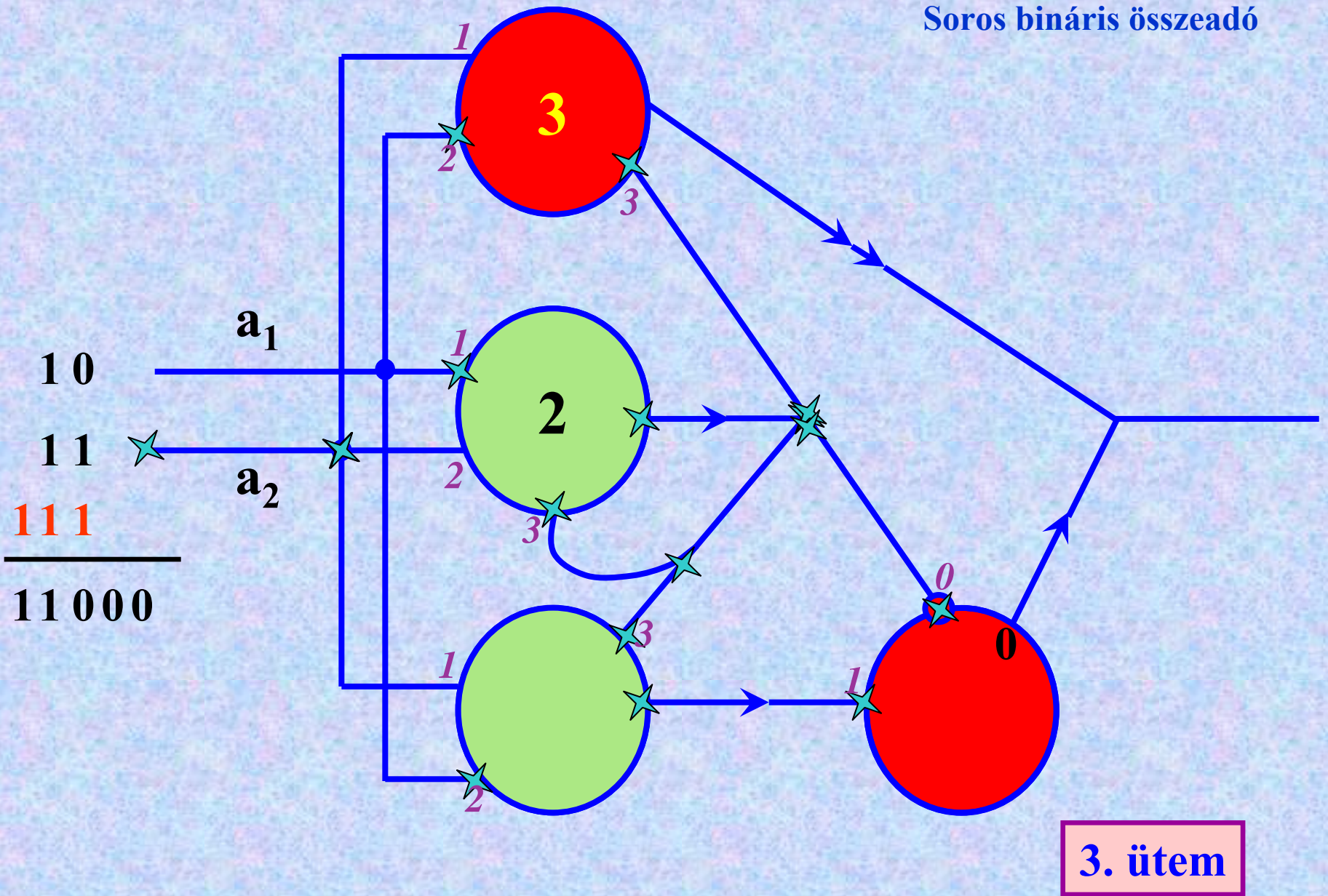


$1\ 0\ 1$   $a_1$   
 $1\ 1\ 0$   $a_2$   
 $1\ 1\ 1\ 1$   


---

 $1\ 1\ 0\ 0\ 0$

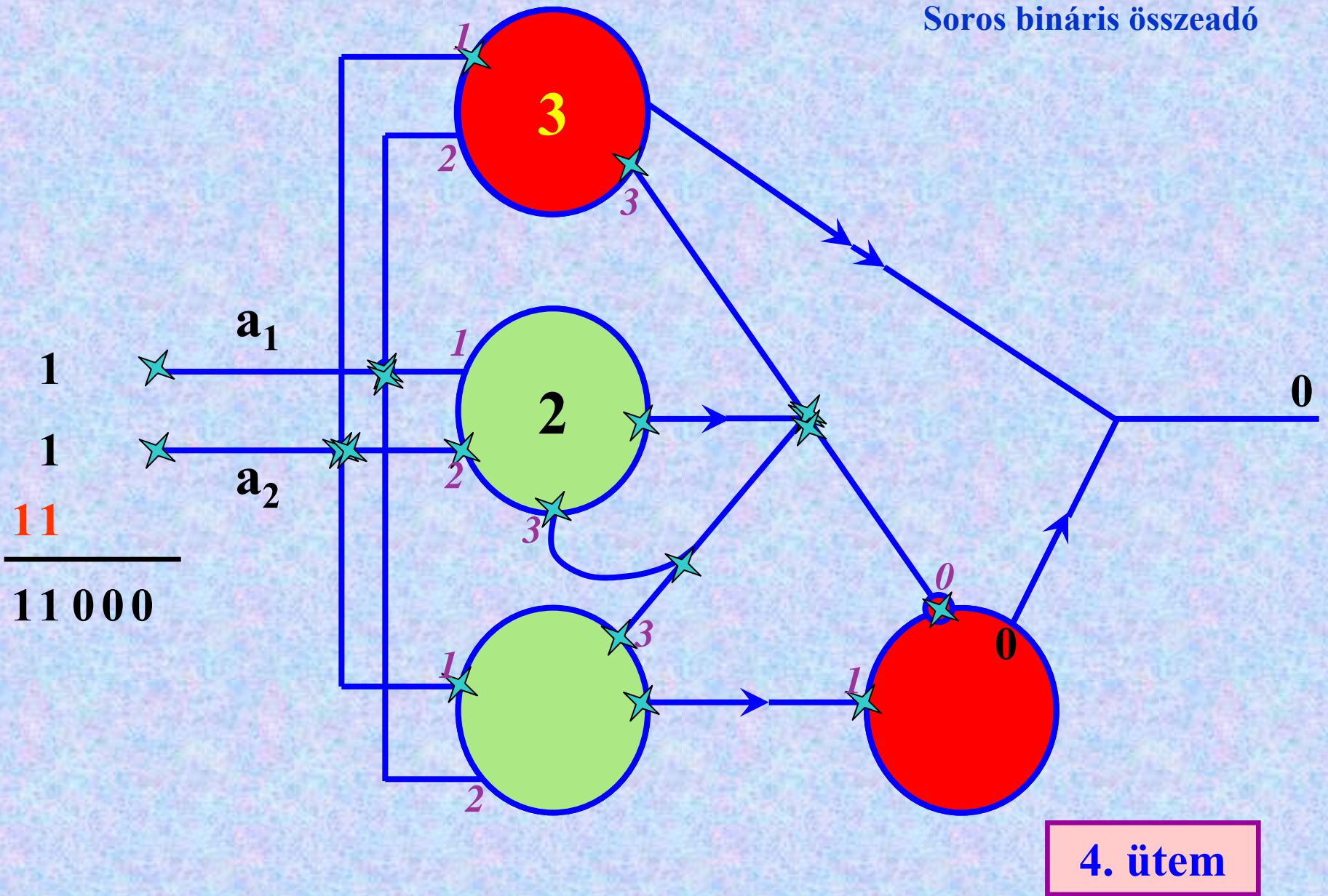
2. ütem

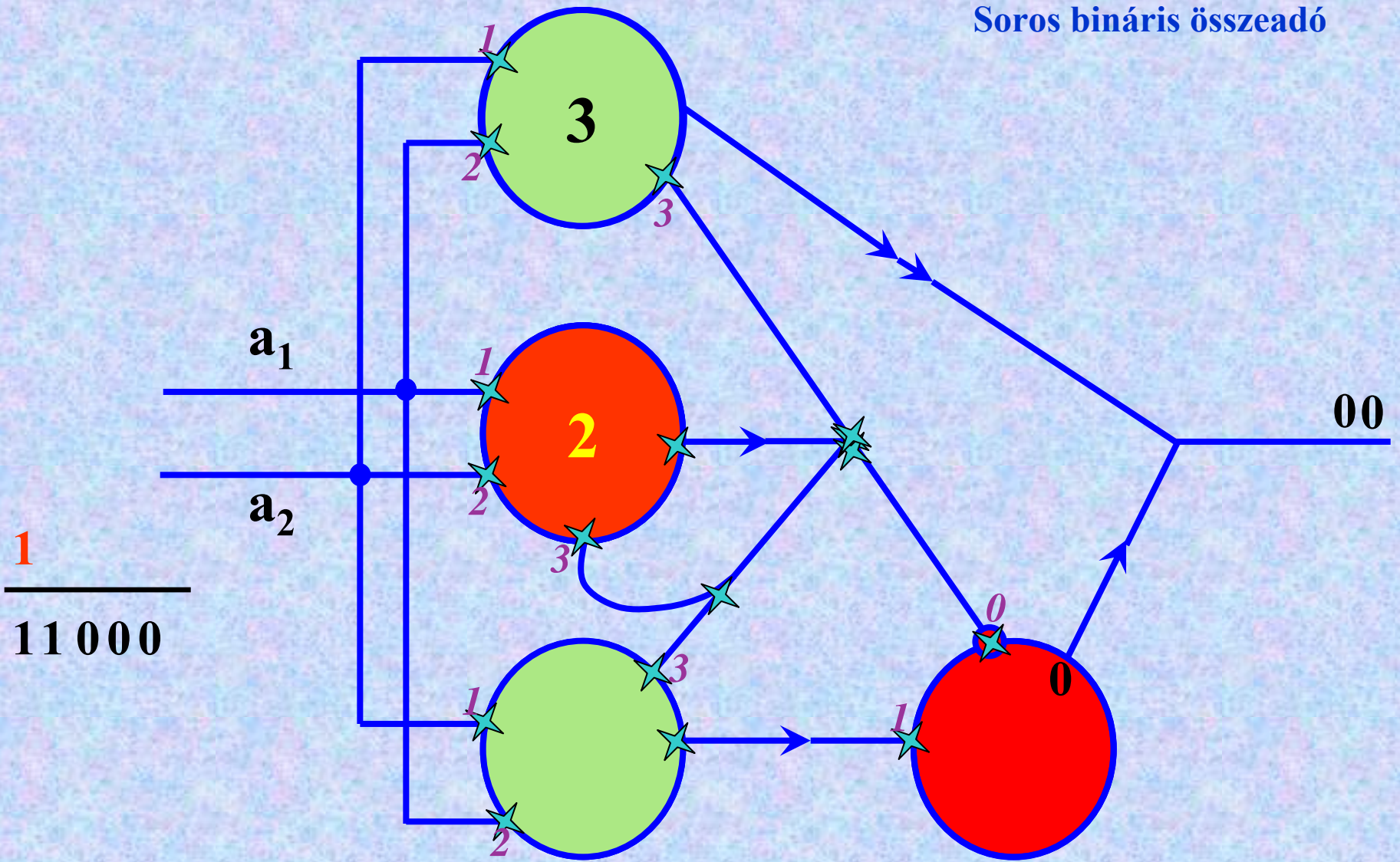


**111**  


---

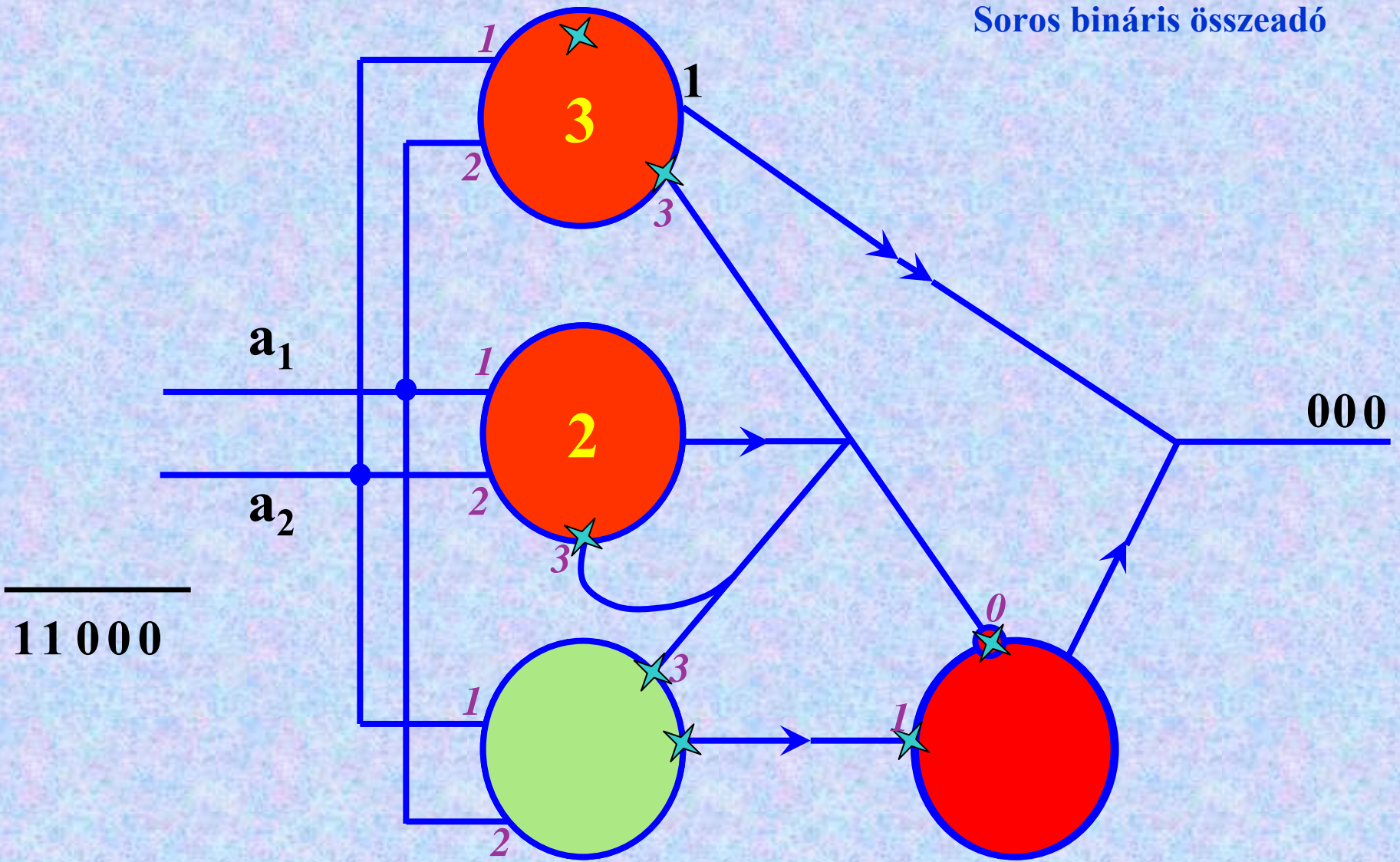
**11000**





1  
-----  
11 000

5. ütem

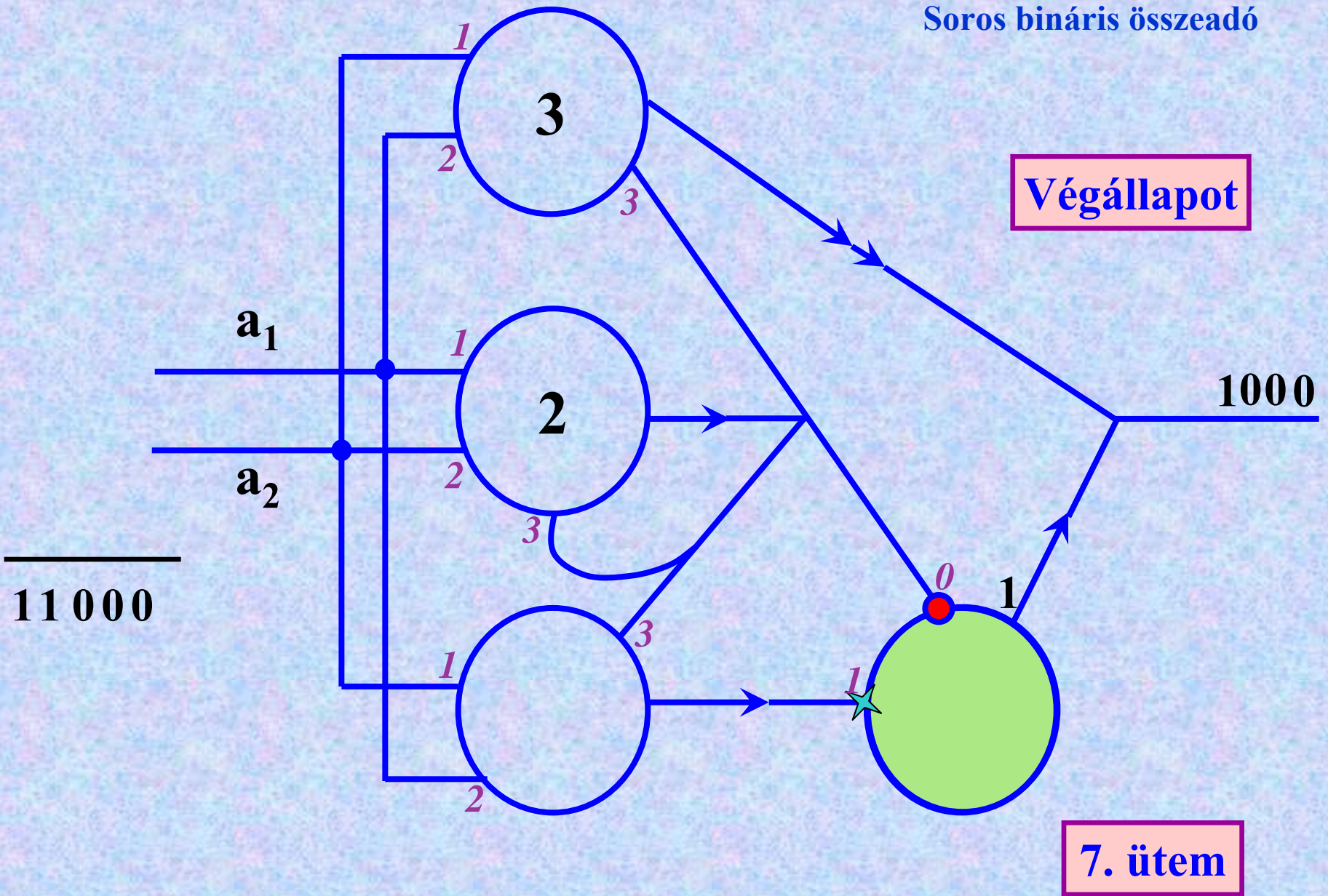


000

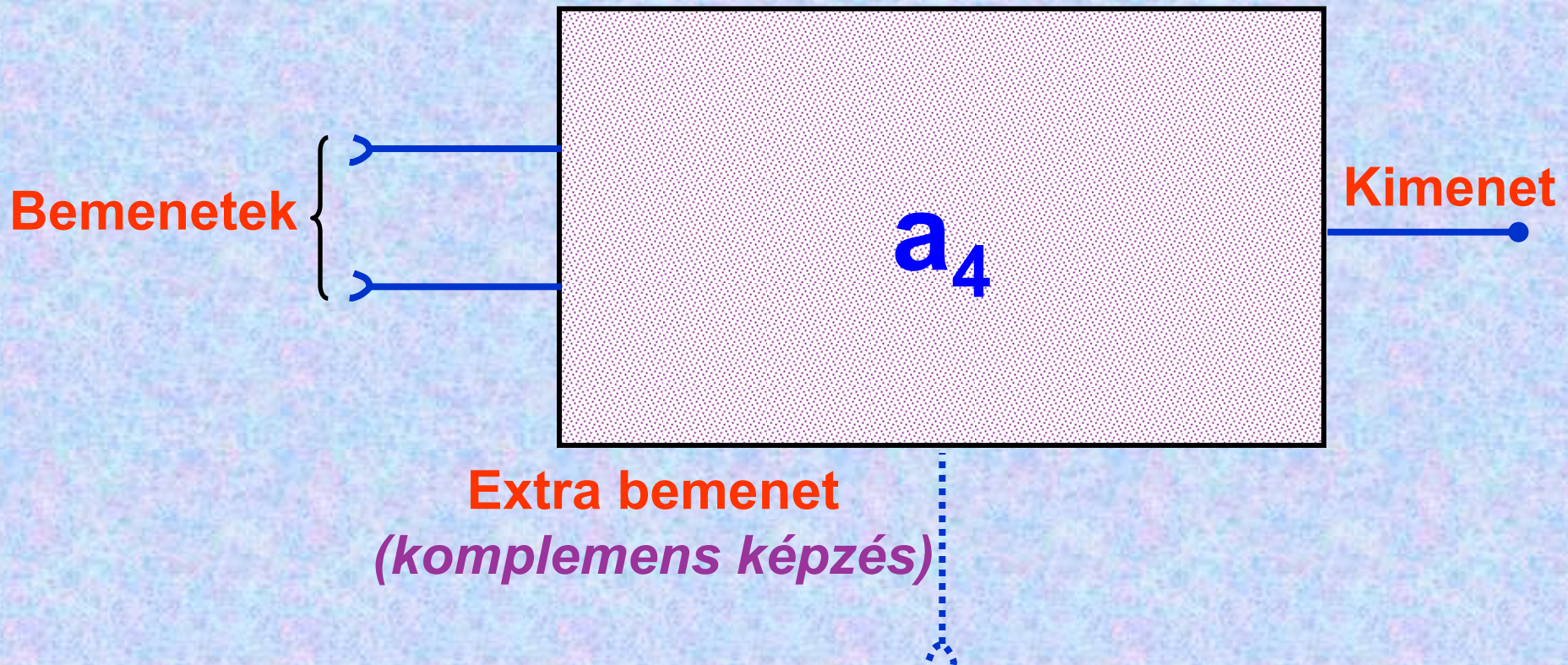
11000

6. ütem





# A bináris összeadó blokk jelölése Neumannnál (*a=aritmetikai blokk*)



# 7. fejezet

## Az informatika kezdetei Németországban

Konrad Zuse élete és munkássága

## Konrad Zuse (1910-1995)

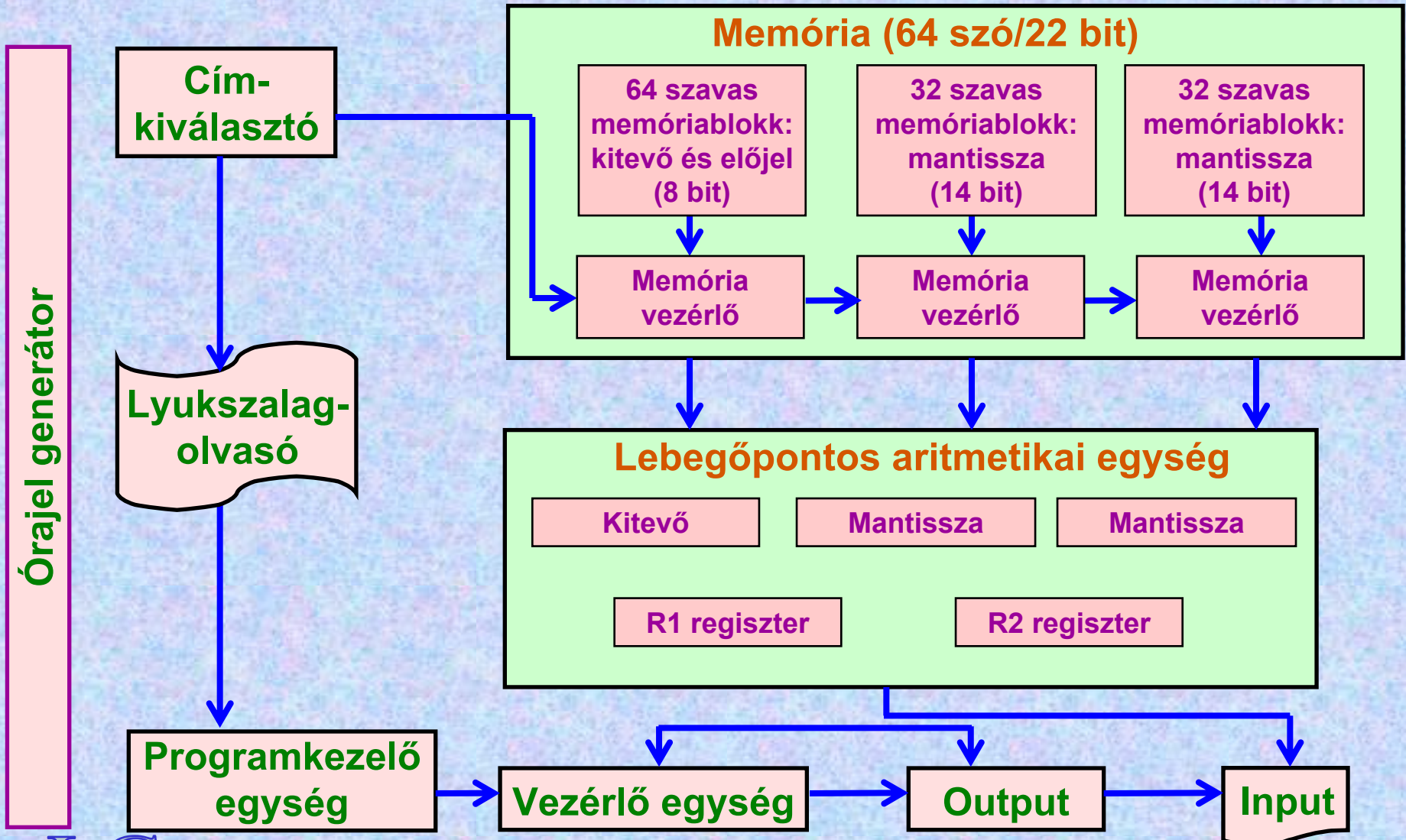


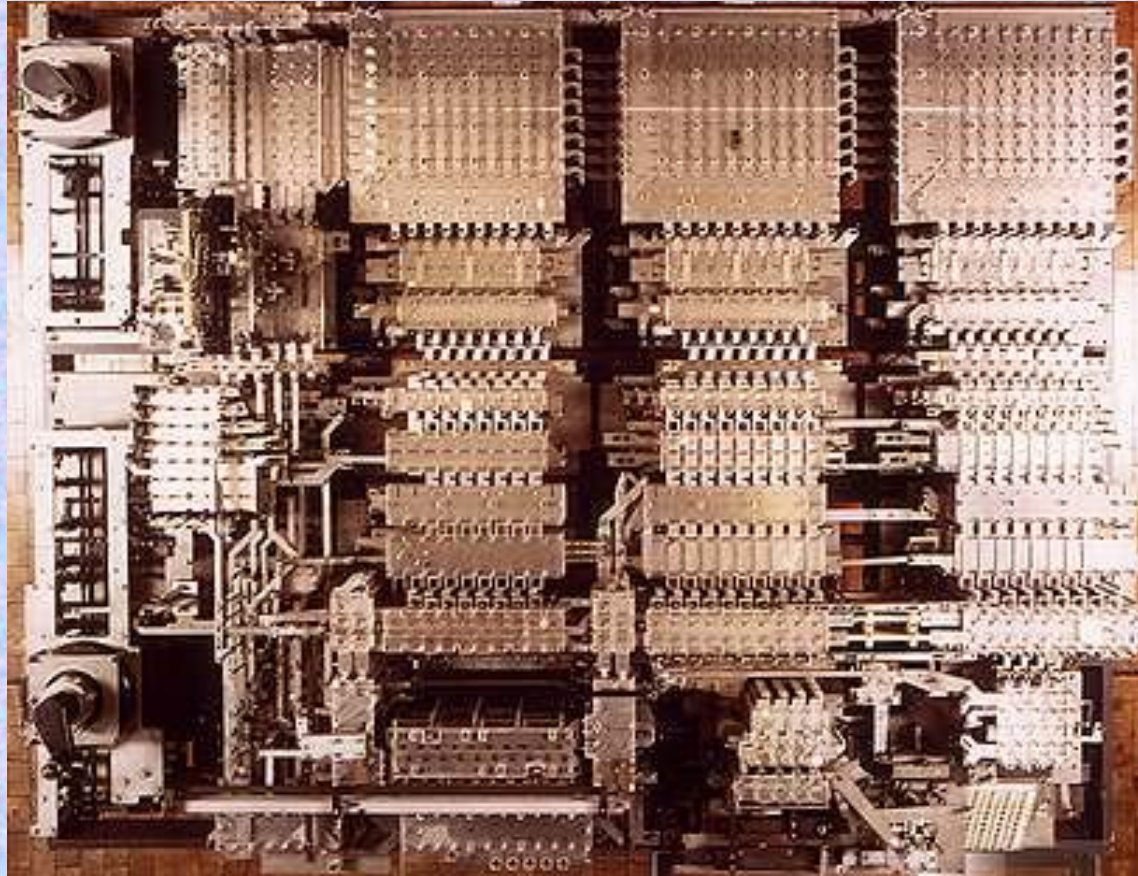
- 1910 jún. 22.: született, *Berlin-Wilmersdorf-ban*
- 1927: *Berlin-Charlottenburg:* ált. mérnöki tanulmányok
- 1935: megszerzi diplomáját, majd a Henschel repülőgépgyárba kerül
- 1934-38: felépíti első számítógépét, a *Z1-et*
- ~1938: felépíti a *Z2-t*, amely hasonló a *Z1-hez*, de az aritmetikai egységben kísérletképpen reléket használ

## A Z1 jellemzői

- **Aritmetika:** lebegőpontos bináris
- **Memória:** 64 szó (*mechanikus, nem relés!*)
- **Szóhossz:** 22 bit
- **Vezérlés:** lyukszalag
- **Főbb egységei:**
  - Memória
  - Vezérlő egység
  - Aritmetikai egység (*2 regiszter; minden művelet összeadásra van visszavezetve*)
  - Input
  - Output

# A Z1 logikai vázlata





**A Z1 a második világháborúban elpusztult. Ez a kép az 1987-89 között újraépített példányt ábrázolja**

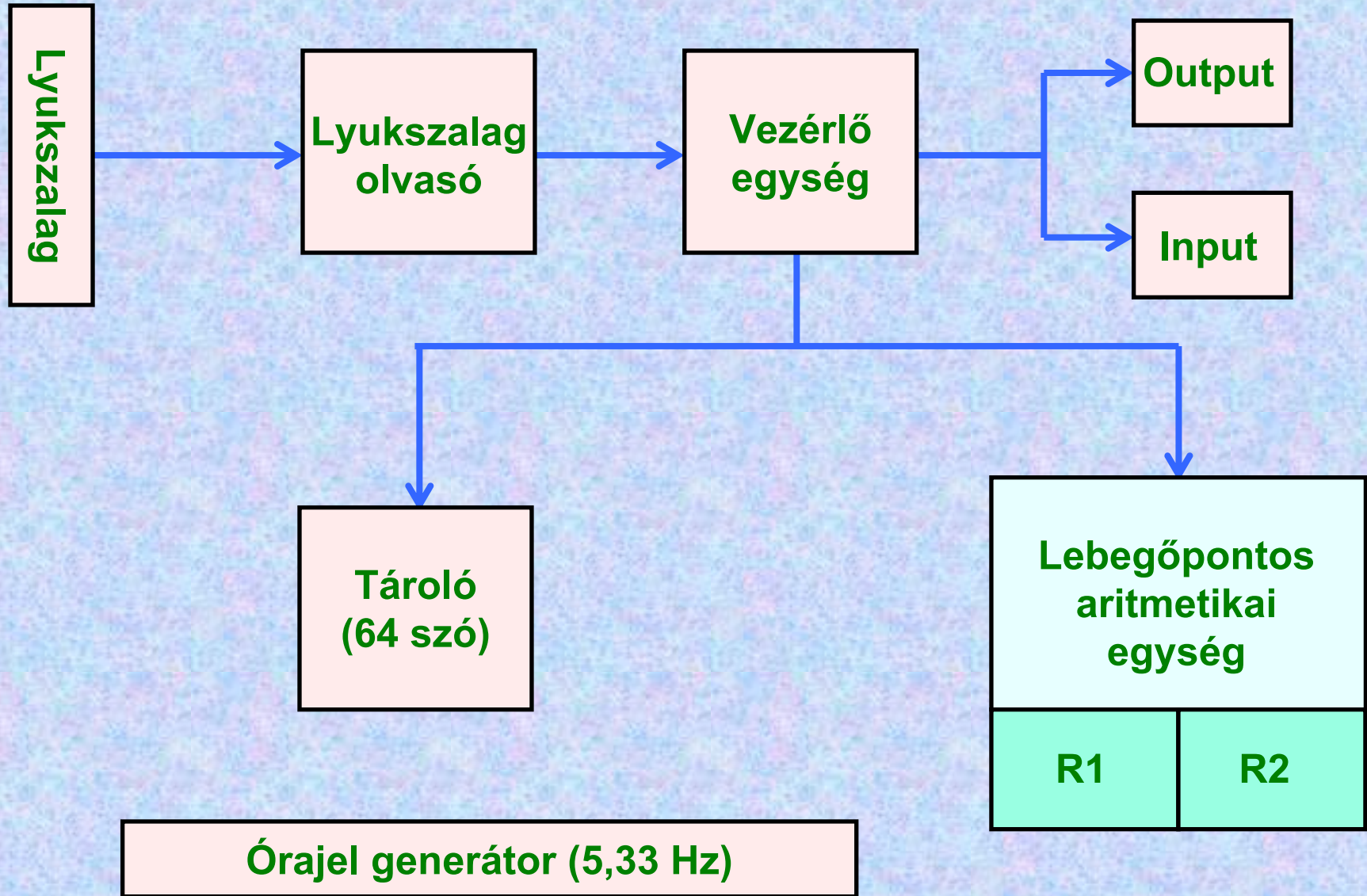
## Konrad Zuse (1910-1995)

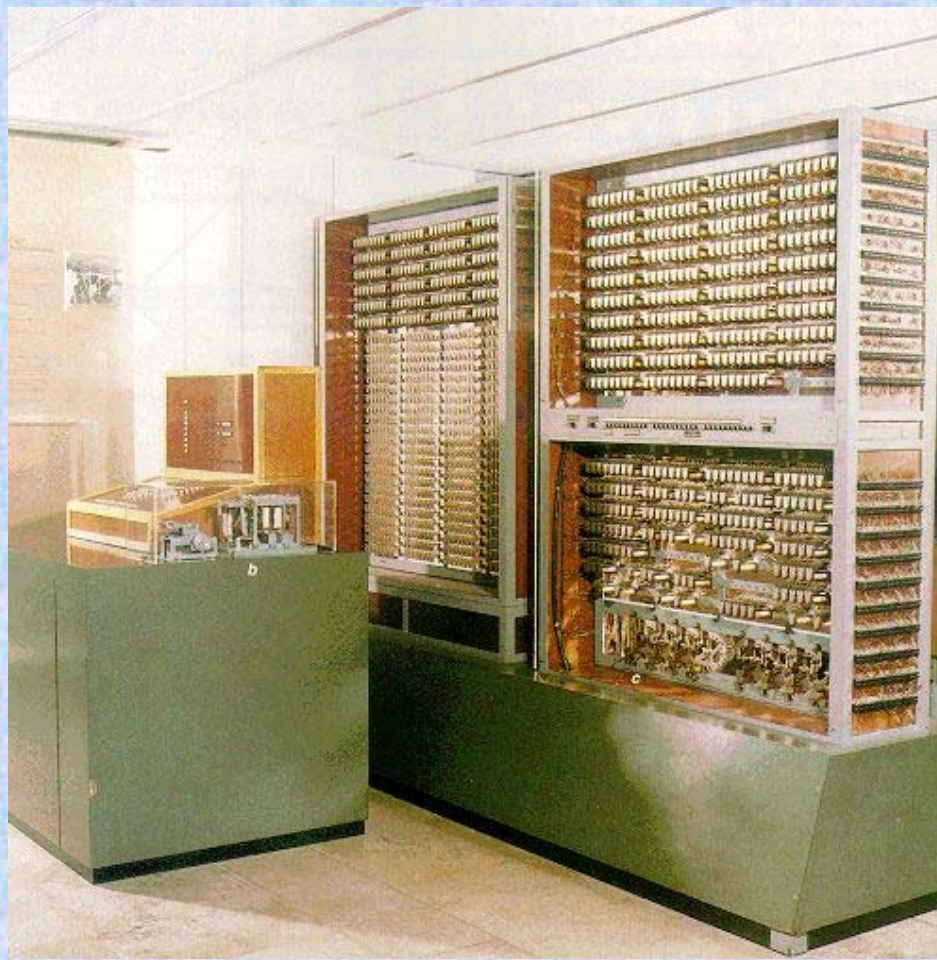


- 1939-41: Egy évi katonai szolgálat után megépíti a Z3- at, amelyet repülőgépek szárnyával kapcsolatos vibrációs számításokra használnak. A gép és dokumentációja szintén megsemmisült a II. világháborúban. Zuse 1960-61 között újraépítette.



## A Z3 logikai vázlat





**Az 1961-ben újraépített Z3 gép**

## A Z3 néhány érdekesebb adata

- **Szóhossz: 22 bit** (*8 bit karakterisztika + 14 bit mantissza*)
- **Tároló: 64 szó** (*jelfogós*)
- **Órafrekvencia: 5,3 Hz**
- **Átlagos műveleti idők (22 bitre):**
  - **Összeadás: 0,7 sec**
  - **Szorzás, osztás: 3 sec**
- **Jelfogók száma: kb. 2000**
- **Súly: kb. 1000 kg**
- **Áramfelvétel: kb. 4 kW**

## Konrad Zuse (1910-1995)

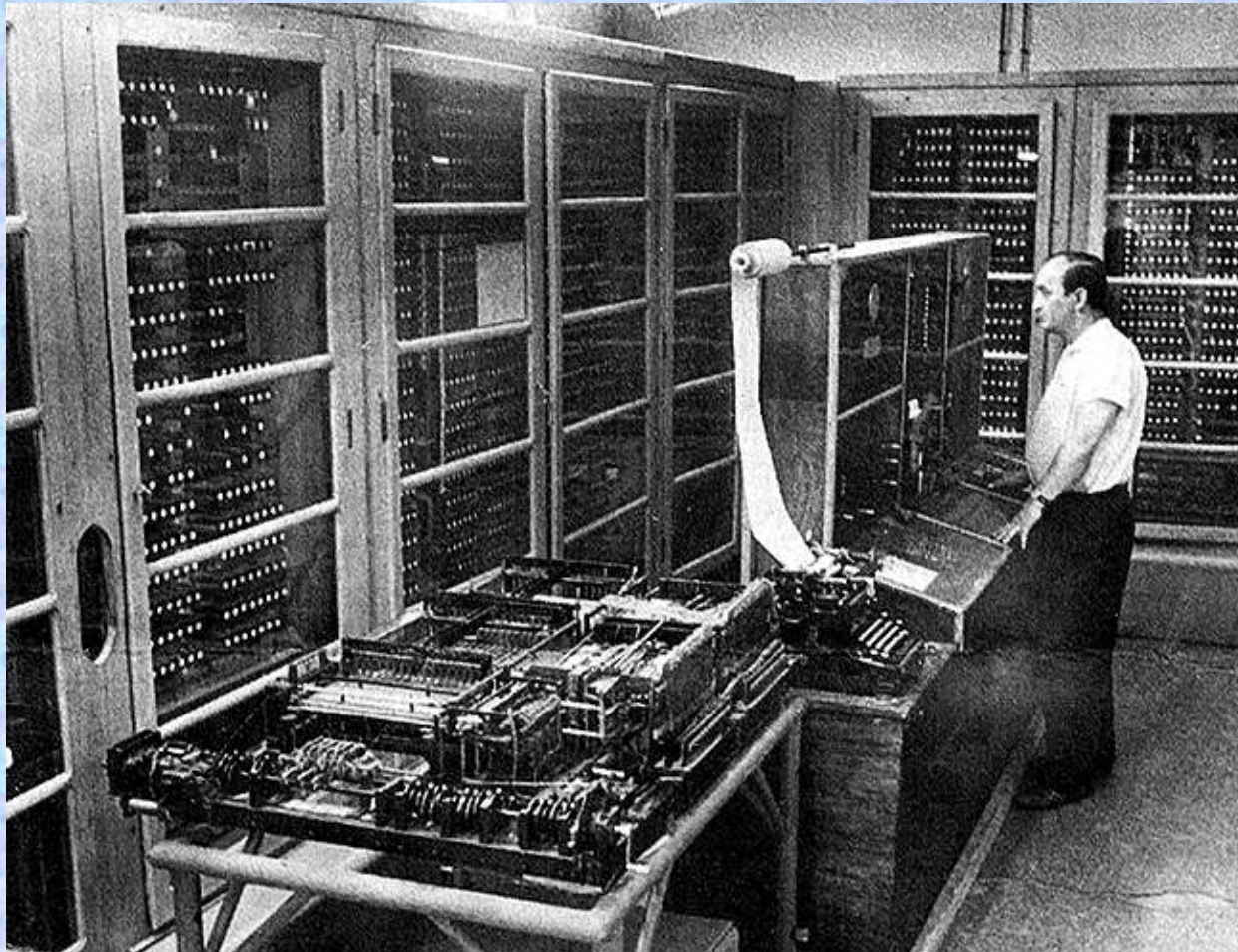


- 1940: Céget alapít, Zuse Apparatebau néven (kb. 20 alkalmazott)
- 1942-45: Megtervezi a Z4 –et, és majdnem befejezi az építését
- 1945 március: A gépet először Göttingenbe, majd - némi kitérővel - Hintersteinbe (Bajorország) menekíti

## Konrad Zuse (1910-1995)



- 1948: Nagyjából helyreállítja a háborúban megsérült gépet
- 1949: Eduard Stiefel bérbeveszi a **Z4**-et a zürichi **ETH** részére
- 1950 július: felújítás és némi továbbfejlesztés után a gépet Zürichben üzembe állítják. Itt 1954-ig működik, majd Franciaországba kerül, ahol 1959-ig használják.



**Az 1950-ben felújított Z4 gép**

## A Z4 néhány érdekes adata

- **Szóhossz:** 32 bit (*7 bit karakterisztika +24 bit mantissza + előjel. Z3: 22 bit*)
- **Tároló:** mechanikus, 64 szó (*tervezett: 500 szó*)
- **Órajel:** 30 Hz (*Z3: 5,3 Hz*)
- **Műveleti sebesség:** átlag 11 szorzás/sec (*Z3: 0,3 /sec*)
- **Input:** decimális billentyűzet és lyukszalag
- **Programelőkészítés:** külön egység programszalagok előállítására (*nem egyszerű szalaglyukasztó!*)
- **Output:** Mercedes írógép
- **Súly:** kb. 1000 kg
- **Áramfelvétel:** kb. 4 kW

## Konrad Zuse (1910-1995)



- 1942-46: a **Z4** tervezésével egyidejűleg kidolgozza a *Plankalkül-t*, az első magasszintű programozási nyelvet



## A Plankalkül jellemzői

- **Cél:** bármilyen (*nemcsak numerikus számítási*) algoritmus leírására alkalmas eszköz létrehozása
- **Megvalósítás:** Zuse kezdetben *nem gondolt* gépi realizációra. Később foglalkozott a gondolattal, de ennek feltételei nem álltak fenn
- **A nyelv sajátos tulajdonságai:**
  - Felhasználó által definiálható adattípusok
  - „kétdimenziós” írásmód
  - kivételkezelés lehetősége programból
  - a változókra vonatkozó logikai állítások használatának lehetősége (*ld. később: Hoare*)

# A Plankalkülben alkalmazott jelölések

## AZONOSÍTÓK

<b>V<sub>0</sub>, V<sub>1</sub>...</b>	<b>Formális paraméterek jelölésére</b> ( <i>Variable</i> )
<b>R<sub>0</sub>, R<sub>1</sub>...</b>	<b>Eredmény, kimenő paraméter</b> ( <i>resultatwert</i> )
<b>Z<sub>0</sub>, Z<sub>1</sub>...</b>	<b>Közbülső eredmény lokális változó</b> ( <i>Zwischenwert</i> )
<b>S<sub>0</sub>, S<sub>1</sub>...</b>	<b>Adattípus, mód</b>
<b>A<sub>0</sub>, A<sub>1</sub>...</b>	<i>(kb. abban az értelemben, ahogy a Pascal használja)</i>
<b>P<sub>1</sub>, P<sub>2</sub>...</b>	<b>Program, szubrutin</b>
<b>P<sub>1</sub>•1, P<sub>1</sub>•2...</b>	<i>(skatulyázás is van)</i>

## ADATTÍPUSOK

$S_0$	Elemi adattípus, bit ( <i>Ja-Neinwert</i> )
$n \times S_0 = S_1 \cdot n$	$n$ bites szó, egész (pl.: $S_1 \cdot 4 = \text{tetrád} = \text{hexadecimális számjegy}$ )
$(S_{i_1}, S_{i_2}, \dots, S_{i_k})$	$k$ komponensű, összetett objektum, rekord, „ $k$ -as” (pl.: $(S_1 \cdot 3, S_1 \cdot 3) = \text{két, } 0 \text{ és } 7 \text{ közötti egészből álló szám, koordináta pár, a sakktabla egy mezeje}$ )

**Minden adattípus az  $S_0$ -ból épül fel!**

# Később bevezetett standard adattípusok:

**A9 = Egész változó**

**AΔ1 (3×S0, 7×S0, 22×S0) = Lebegőpontos változó**



## A Plankalkül sajátos írásmódja

- Egy „programsor” 3-4 szövegsorból áll  
Pl.:  $R_0$  írásmódja:  $R_0$
- Az index a változó neve alá (3. sor) kerül  
Pl.: ha  $V_0$  típusa  $m \times S1 \cdot n$ , akkor ennek  $i$ -edik eleme:  $V_{0i}$
- Az  $i$ -edik komponens  $j$ -edik eleme az  $i \cdot j$  párossal címezhető meg:  $V_{0i \cdot j}$

## A Plankalkül sajátos írásmódja

- Az adatstruktúrák (rekordok) komponensei hasonlóan indexelhetők. Pl.: ha  $V_1$  típusa  $(S1\cdot3, S1\cdot3)$ , akkor ennek komponensei:

$V$		$V$
$1$	<b>és</b>	$1$
$1$		$2$

- A negyedik sorban *kommentként* megadható a változók vagy komponensek típusa Pl.:

$V$		$V$
$0$	<b>és</b>	$1$
$i\cdot j$		$2$
$S0$		$S1\cdot3$

## A Plankalkül sajátos írásmódja

- **Értékadás jele:**  $\Rightarrow$

Pl.:     $Z + 1 \Rightarrow Z$     Jelentése  $Z_1 := Z_0 + 1$ , mindkét  
           0                    1    változó típusa S1•3  
           S1•3                S1•3

- **A  $\Rightarrow$  jel másik alkalmazása:** *eljárások bemenő és kimenő értékeinek specifikálása*

Pl.: P1    |    R(V)  $\Rightarrow$  R  
           V    |    0        0  
           A    |     $\Delta 1$      $\Delta 1$

**Jelentése:** a P1 eljárás bemenő paramétere  $V_0$  ,  
kimenő paramétere  $R_0$  , és mindkettő típusa  $A\Delta 1$

## A Plankalkül sajátos írásmódja

- **A W operátor a programciklus jele**
  - $W(n)$ : ~For  $i:=n$  Downto 0 Do...
  - $W(\forall x(p(x)))$ : A ciklusmag minden olyan  $x$  elemre végrehajtódik, amelyre igaz a  $p(x)$  feltétel
- **$\mu(x)(p(x))$  operátor: a következő,  $p(x)$  feltételnek eleget tevő elem előhívása**
- **Hiányzik az eszköztárból:**
  - If...then...else, goto
- **Megtalálható az eszköztárban:**
  - Kiugrás a ciklusból a magasabb rendű ciklusba, vagy a következő végrehajtási menet elejére



## Részlet egy eredeti Plankalkül programból

$$\begin{array}{l}
 P\ 148 \\
 V \\
 A
 \end{array}
 \left|
 \begin{array}{l}
 R\ (V) \rightarrow R\ 148 \\
 0 \quad 0 \\
 5 \quad 0
 \end{array}
 \right.
 \quad (1)$$

$$\begin{array}{l}
 V \\
 K \\
 A
 \end{array}
 \left|
 \begin{array}{l}
 x \left[ (x \in V) \wedge (x = 10) \right] \Rightarrow Z \\
 0 \\
 4 \left[ \begin{array}{cc} & 1 \\ 5 & 3 \end{array} \right] \quad +
 \end{array}
 \right.
 \quad (2)$$

$$\begin{array}{l}
 V \\
 K \\
 A
 \end{array}
 \left|
 \begin{array}{l}
 (Ex) \left[ (x \in V) \wedge R\ 17\ (Z, x) \wedge (x=0) \vee x \right. \\
 0 \quad 0 \\
 4 \left[ \begin{array}{ccc} 4 & 5 & 2 \end{array} \right] \quad 2 \quad 3 \quad 0
 \end{array}
 \right.
 \quad (3)$$

$$\begin{array}{l}
 V \\
 K \\
 A
 \end{array}
 \left|
 \begin{array}{l}
 \wedge \overline{Ey} \left[ (y \in V) \wedge y \wedge R\ 128\ (v, y, x) \right. \\
 0 \quad 0 \\
 4 \left[ \begin{array}{ccc} & 1.3 & 0 \\ 5 & 0 & 5 \end{array} \right] \quad 2 \quad 2
 \end{array}
 \right.
 \quad (4)$$

## Miért ilyen bonyolult („felhasználóidegen”) a Plankalkül?

- Az informatika történetének első ilyen kísérlete, előzménye nincs
- Nem program *készítési*, hanem program *tervezési* eszköznek készült
- Zuse nem azt nézte, hogy *mi valósítható meg*, (akkoriban még semmi) hanem azt, hogy *mire lenne szükség*

## Konrad Zuse (1910-1995)



- **1949:** Zuse megalapítja a **Zuse KG-t**, Németország első számítógép cégét.  
Első feladat:  
a **Z4** szállításra előkészítése
- **1950:** elkészíti a **Z5-öt**, a **Z4** továbbfejlesztett változatát.  
1 db készül belőle
- **1955-58:** kifejleszti a **Graphomat-ot** (lyukszalag v. kártya vezérlésű síkplotter).  
Ebből kb. 120 db készült

## Konrad Zuse (1910-1995)



- 1957-60: kifejleszti a **Z22-t**. Ez a cég első elektroncsöves, tárolt programú, ferrit-memóriás gépe. Utóda (**1961-62**) a tranzisztoros **Z23**.
- 1962: a cég fokozódó anyagi problémákkal küzd, és Zuse kénytelen eladni. Többszöri tulajdonosváltás után végül a Siemensé lesz.
- 1969: Zuse elhagyja a céget

## Konrad Zuse (1910-1995)



- 1992-95: utolsó konstrukciója: egy változtatható magasságú szélerőmű, amely az optimális szélesebességnek és –irány-nak megfelelően termel energiát. A tervet halála miatt nem tudta befejezni, és az később sem valósult meg.
- 1995.dec. 18: Hünfeldben szívroham következtében meghal.



## A Graphomat



**A Zuse KG gyára Hersfeldben.  
1964-ben kb. 1200 dolgozójuk volt**



**Zuse nem szerette a PC-ket...**



## A ZUSE KG csődjének néhány oka

- Tőkeszegénység, amely részben a háború utáni helyzetből, részben a szponzorok hiányából eredt
- A cég túl gyors növekedése *(a dolgozói létszám kb. 10 év alatt mintegy 30-ról 1200-ra növekedett)*
- A cég „egy lábon állt” *(kizárólag számítógépeket gyártott)*
- A software fejlesztés növekvő költségei
- Az IBM konkurenciája

**1949 és 1969 között a ZUSE KG kb. 250 gépet gyártott**

## Konrad Zuse munkásságának informatika történeti jelentősége

- Először írta le a tárolt program elvét (1936)
- Először alkalmazta a lebegőpontos számábrázolást működő gépen (Z1)
- A számábrázolás jelzőbitjei lehetővé tették a programból történő kivételkezelést
- A Z1 és Z4 mechanikus memóriája a maga idejében műszaki különlegesség volt
- A Z3 volt Európa első számítógépe; kb. 3 évvel előzte meg az ASCC-et (MARK-1)
- A *Plankalkül* volt az informatika történetének első magasszintű nyelve

# 8. fejezet

## Kódtörés a II. világháború előtt és alatt

### 1. Az Enigma és a Bomba

# A rejtjelezés

## Szerepe:

- Üzleti titkok megőrzése és továbbítása a jogosultaknak
- Katonai titkok megőrzése és továbbítása
- Hadműveleti tervek és utasítások továbbítása, stb.

## Legfontosabb eleme:

- A **kulcs**, amelynek segítségével az üzenetet **kódolják és dekódolják**
- Léteznek:
  - Szimmetrikus és
  - Aszimmetrikus kulcsú rejtjelező rendszerek

- **Szimmetrikus kulcsos rendszer**
  - A kapott üzenet a küldéskor használt kulccsal visszafejthető
- **Aszimmetrikus kulcsos rendszer**
  - A rendszernek van egy *nyilvános* és egy *titkos* kulcsa. Az üzenetet a nyilvános kulccsal kódolják, és a titkos kulccsal fejthető vissza.

**Az alábbiakban tárgyalt Enigma gép kódolási rendszere szimmetrikus.**

- **Monoalfabetikus kódolási rendszer**
  - Az üzenet minden karakterét ugyanazon ABC szerint kódolják

Példa (5 betűs ABC-n):

**ABCDE = eredeti ABC**

**DAECB = helyettesítő ABC**

**ABDA = kódolandó szöveg**

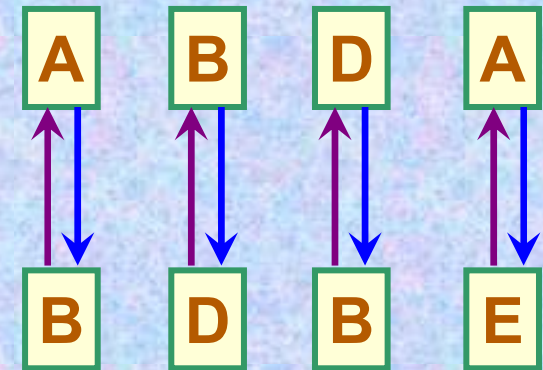
**DACD = kódolt szöveg**

- **Polialfabetikus kódolási rendszer**
  - A helyettesítés során a helyettesítő ABC megváltozik

**Példa (5 betűs ABC-n):**

**Kódtáblázat**  
(*egyben kulcs*)

	A	B	C	D	E
A	A	B	C	D	E
B	B	C	D	E	A
C	C	D	E	A	B
D	D	E	A	B	C
E	E	A	B	C	D

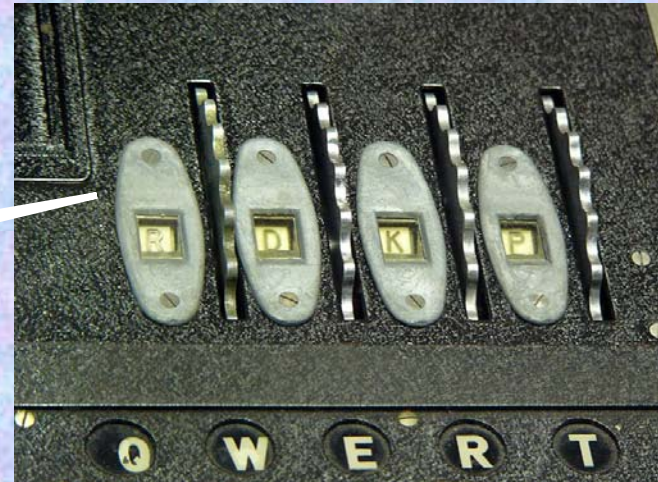


**Kódolt üzenet  
és visszafejtése**

## Az ENIGMA rejtjelezőgép

- **Az ENIGMA gépcsalád:**
  - Feltaláló: *Arthur Scherbius (1918)*
  - Az „ENIGMA” szó jelentése: rejtély
  - A II. világháborúban a németek használták
  - Több változata volt, a különböző fegyvernemek eltérő típust használtak.
- **Fő komponensei:**
  - Rotorok
  - Léptető mechanizmus
  - Tükör
  - Dugaszolótábla
  - Megjelenítő lámpasor





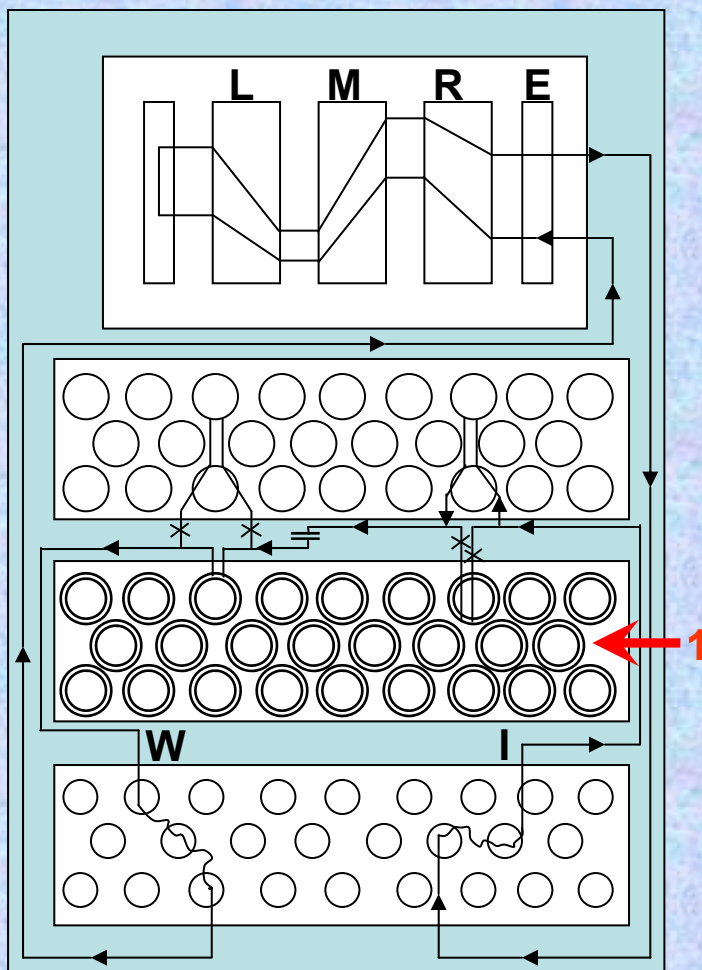
A beállítótárcsák és a hozzájuk tartozó kijelzők

Az ENIGMA külsőre sokban hasonlított egy írógéphez. A képen főként a billentyűzet és a megjelenítő lámpasor látható.



Egy rotor

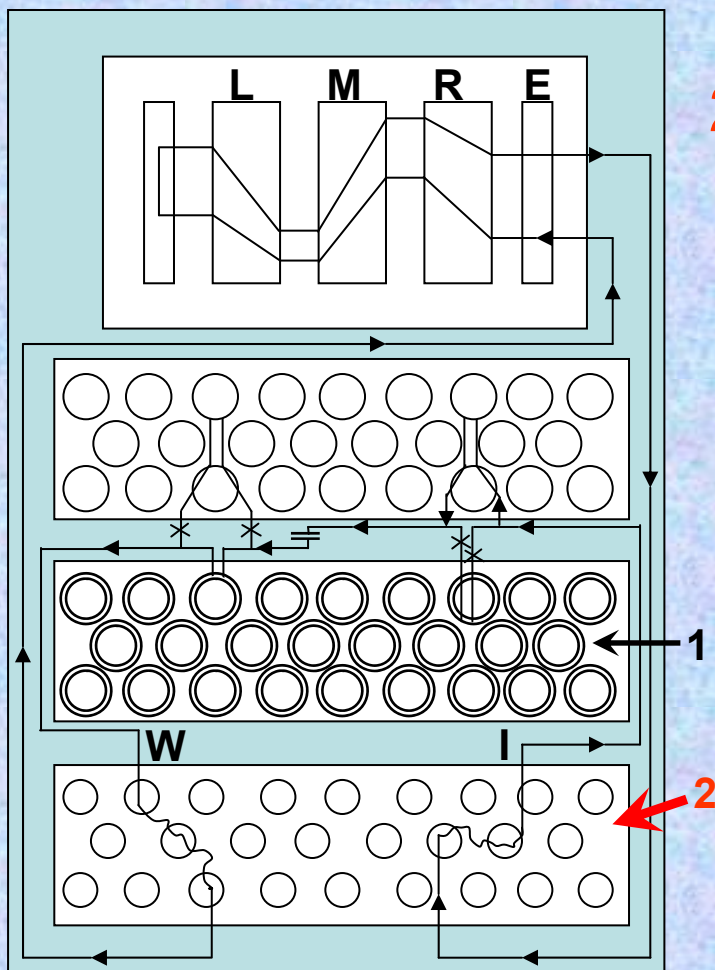
# Az ENIGMA részei



## 1. Billentyűzet

– Itt történik az adatbevitel

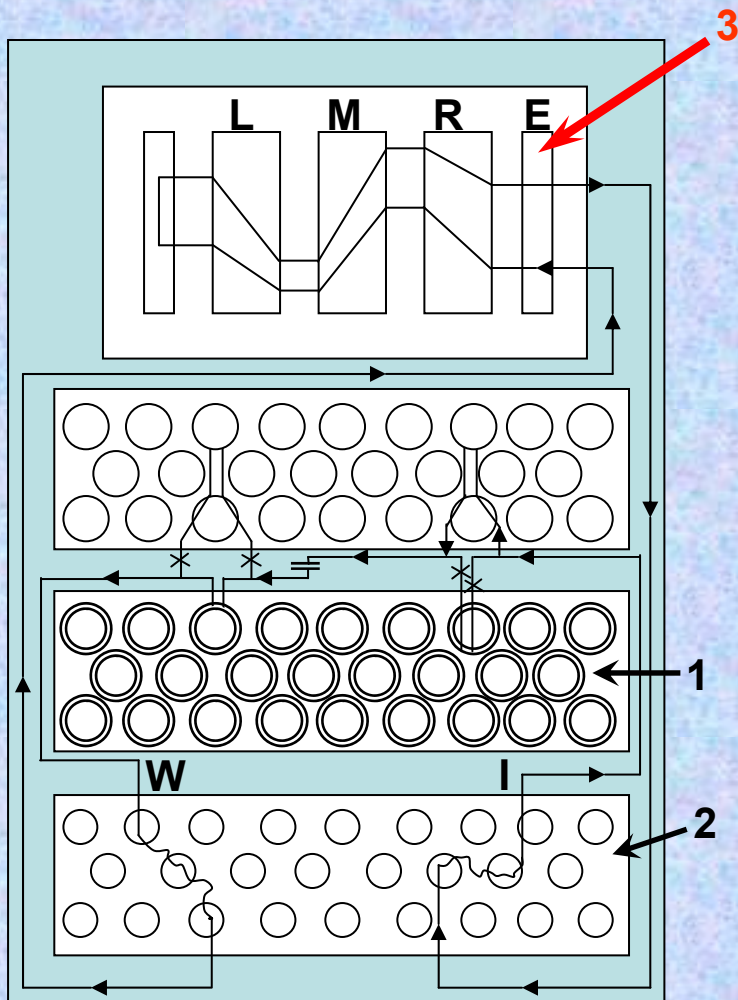
## Az ENIGMA részei



### 2. Dugaszolótábla

- Karaktercseréket hajt végre, melyeket az operátor állít be.

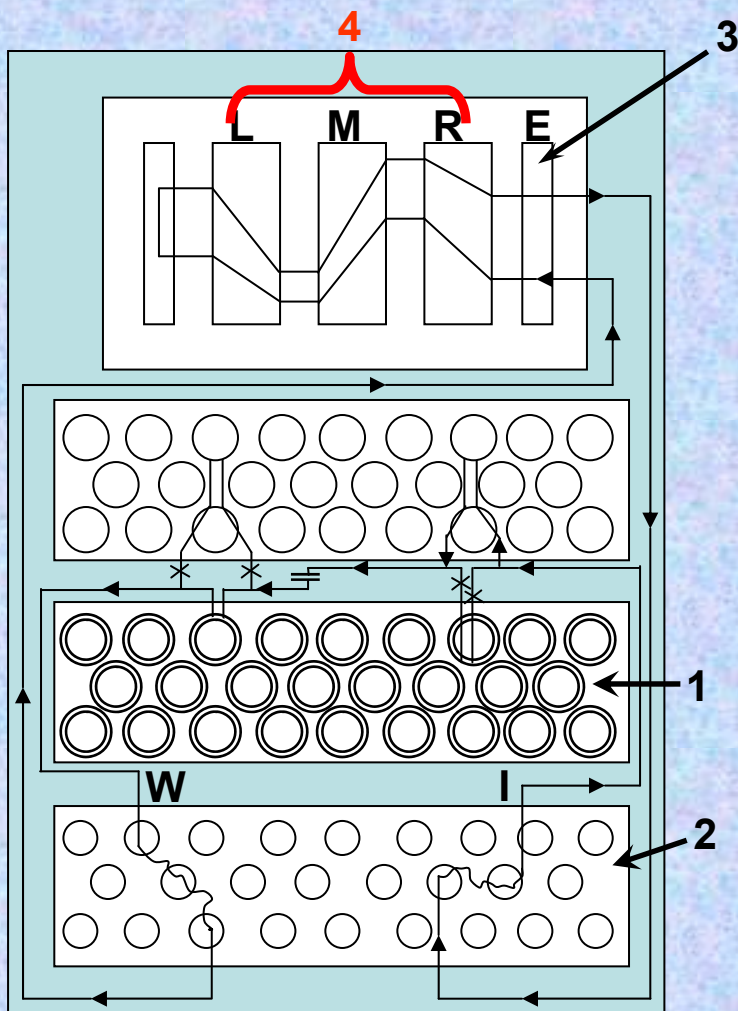
## Az ENIGMA részei



### 3. Belépő ablak

- Az üzenet ezen keresztül érkezik a rotorokhoz, majd keverés után ezen keresztül lép ki.

## Az ENIGMA részei

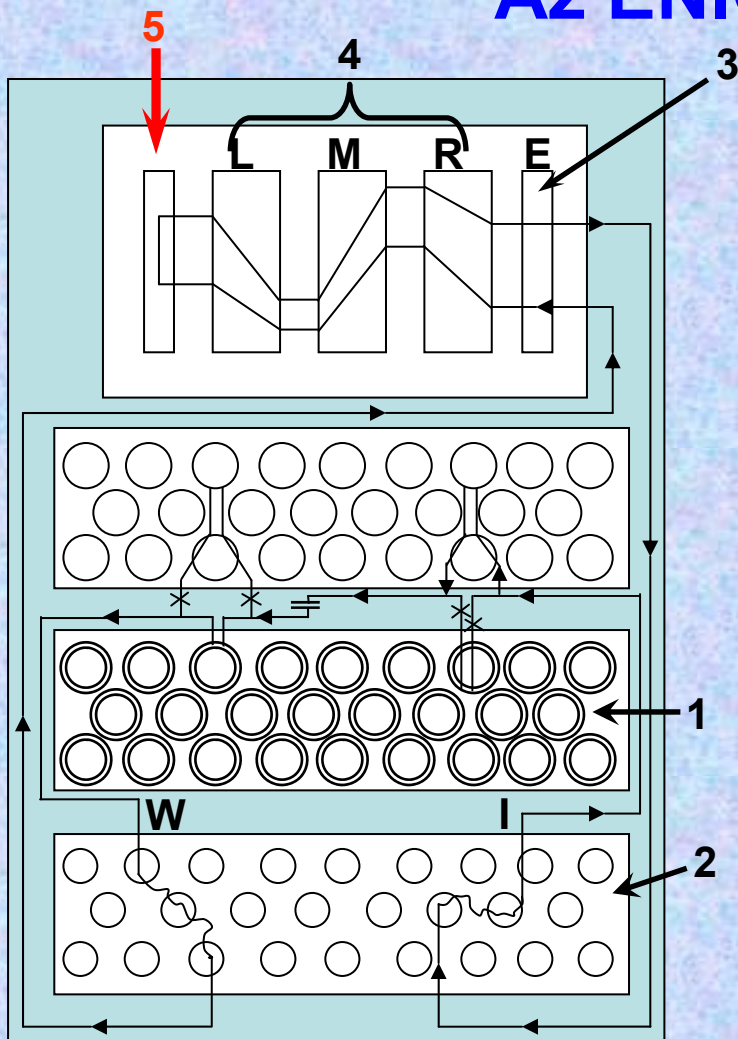


### 4. Rotorok

–Főként ezek végzik a karakterek „keverését”.

- A rotorok része a betűtárcsa (8. dia), amelynek elforgatásával állítható be a kezdő pozíciójuk.

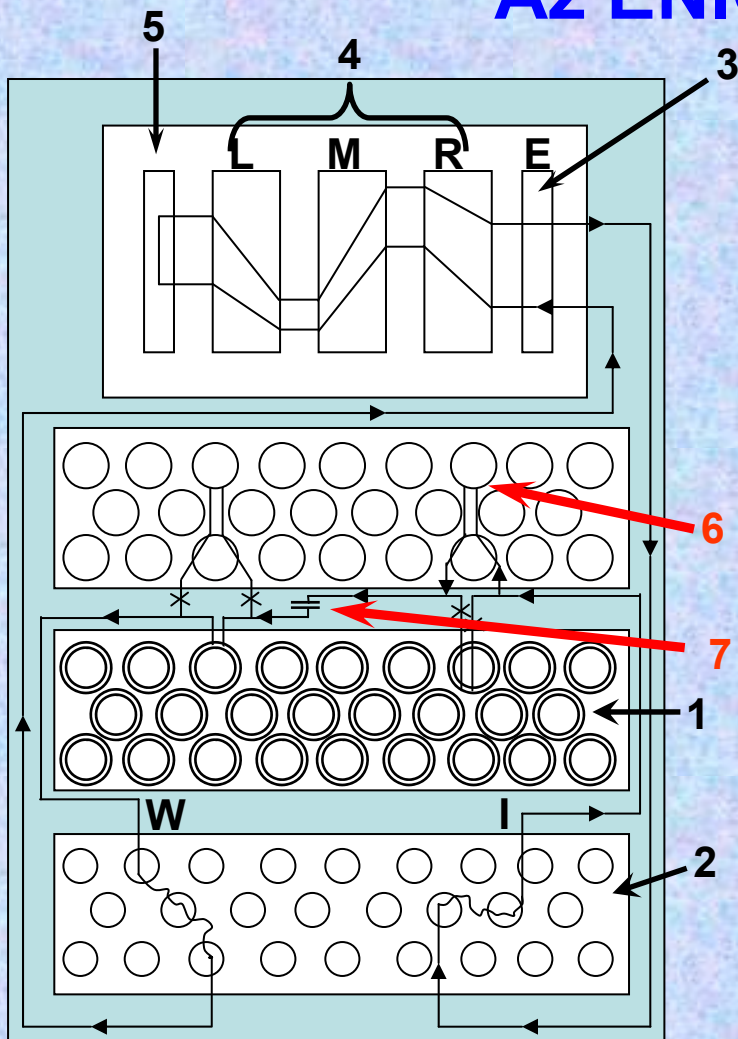
## Az ENIGMA részei



### 5. Tükör

- Belső huzalozása egy fix karaktercserét hajt végre. Az így átkódolt karakter visszakerül a rotorokhoz.

## Az ENIGMA részei

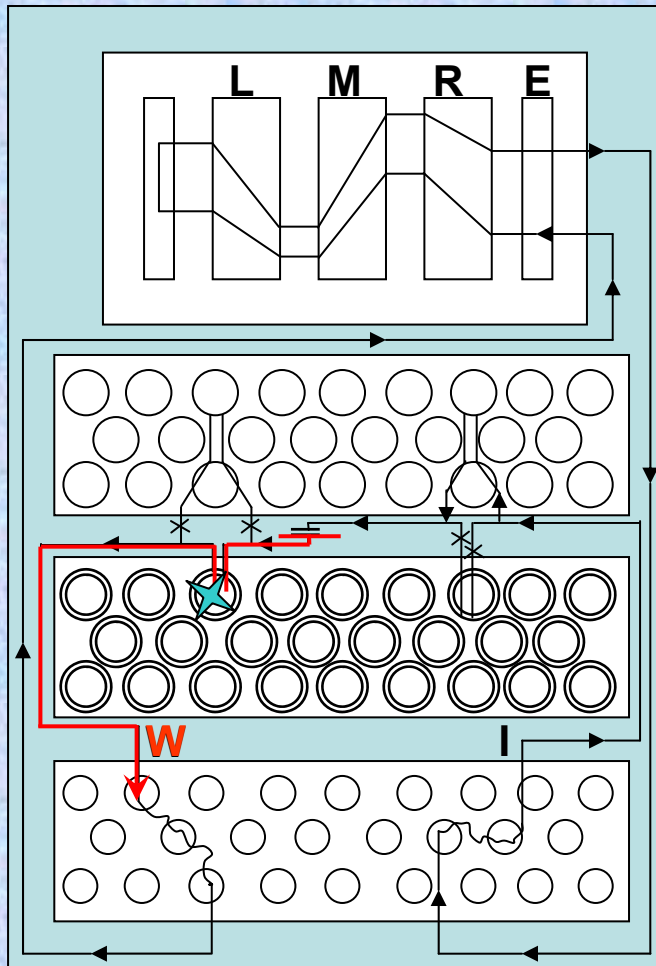


### 6. Megjelenítő lámpasor

- Megmutatja, hogy az eredeti karaktert mivel helyettesítette a rendszer.

### 7. Áramforrás

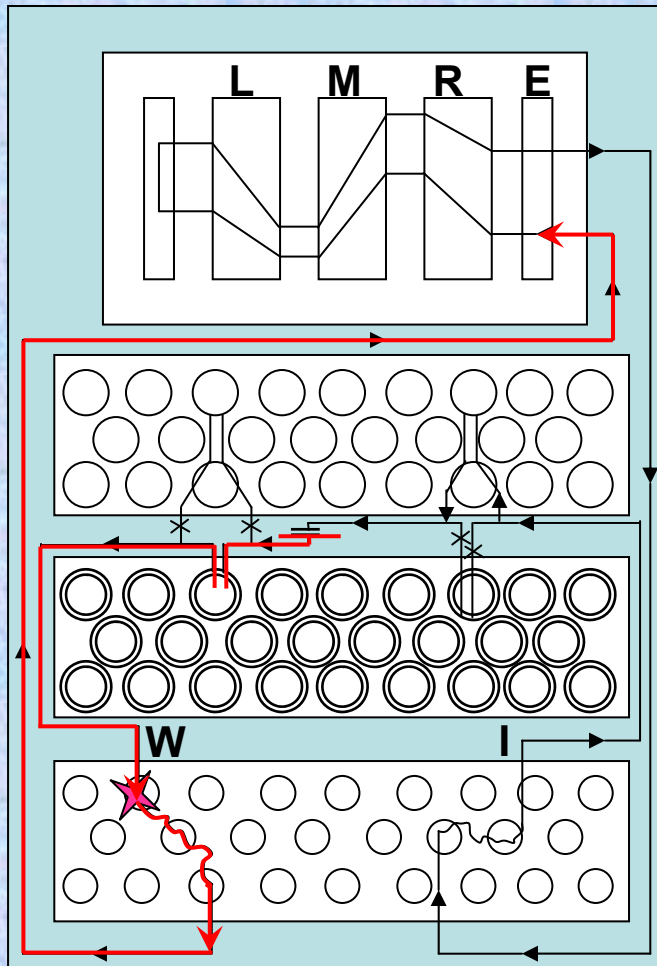
# Az ENIGMA működése



- Az operátor leüti az üzenet betűjét.

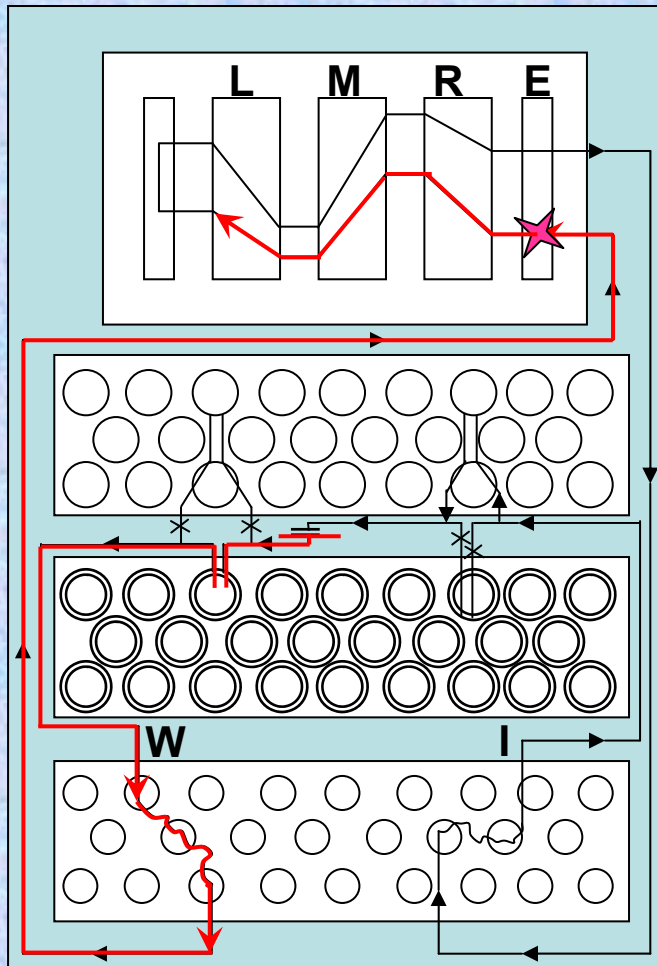


## Az ENIGMA működése



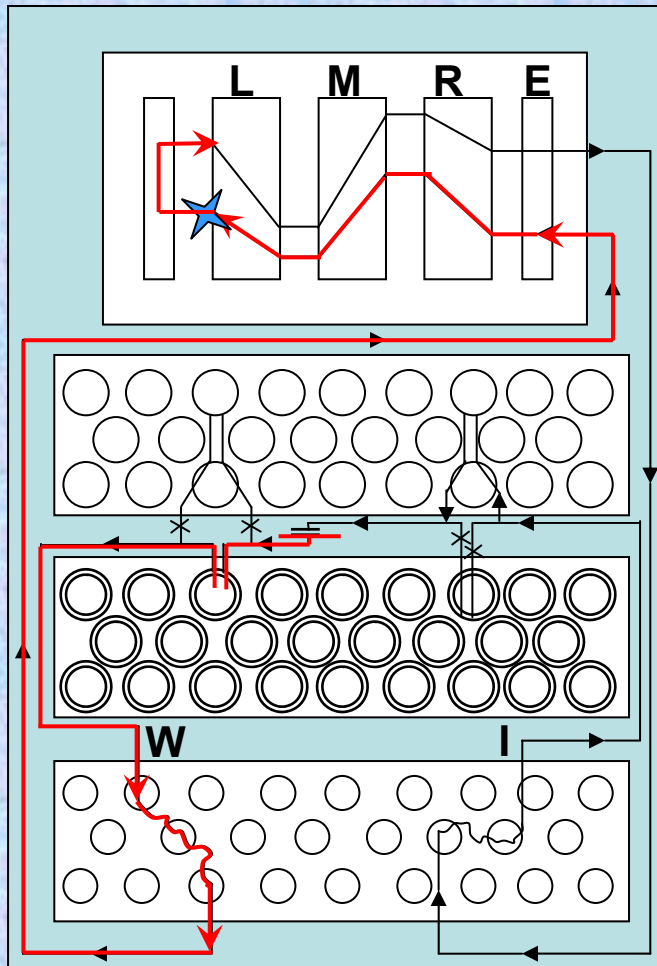
- A betű a dugaszolótábla beállítása szerint megváltozik, majd a belépő tárcsához érkezik.

## Az ENIGMA működése



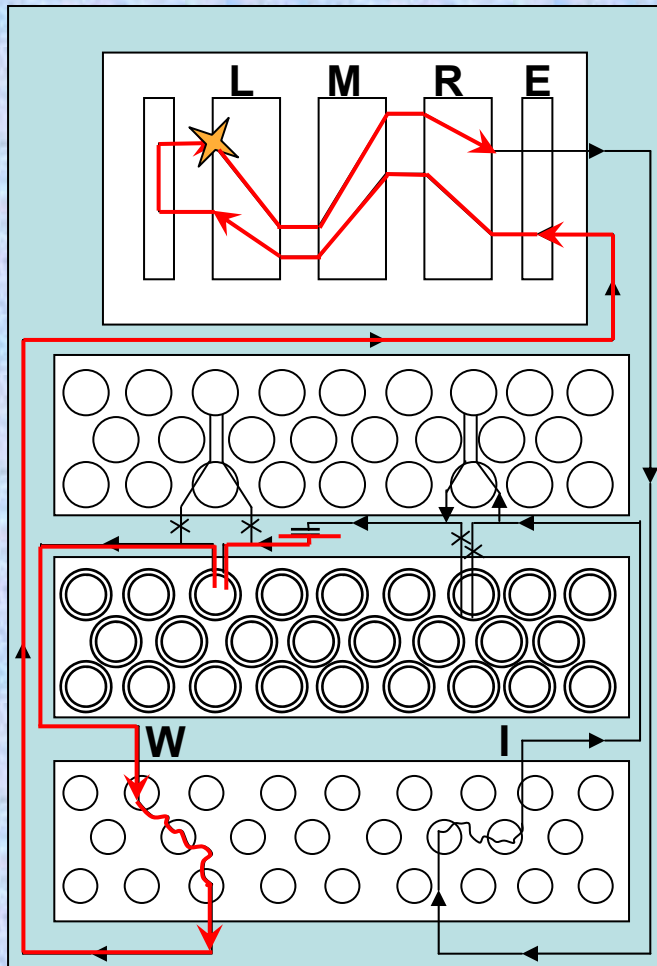
- Mindegyik rotor végrehajt rajta egy-egy helyettesítést, miközben maga a rotor – a léptető mechanizmusnak megfelelően – tovább fordul, vagy helyben marad.

## Az ENIGMA működése



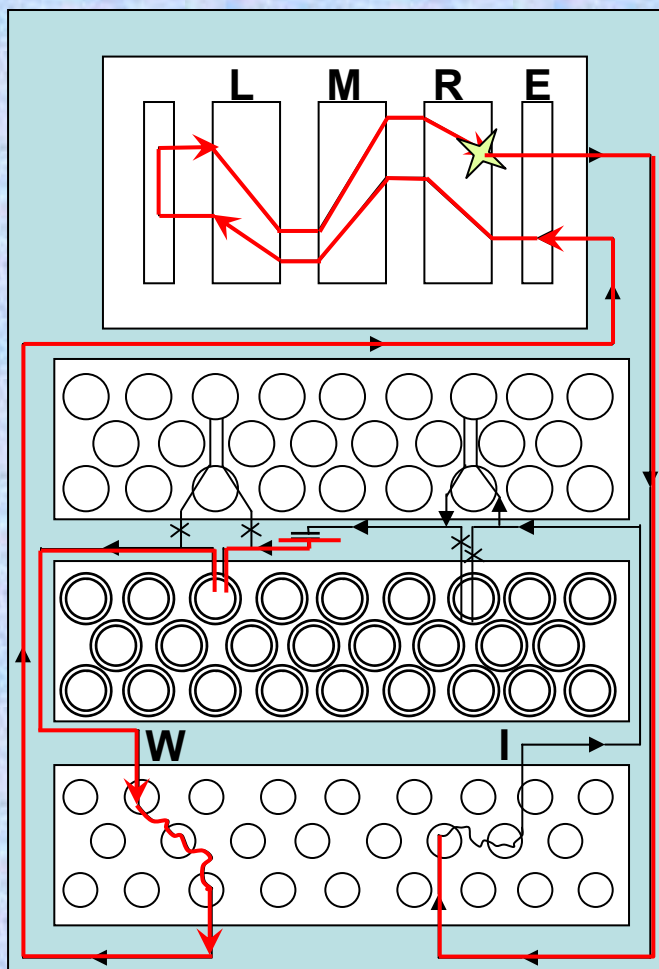
- A betű a tükrőhöz ér, és az abba huzalozott fix helyettesítés szerint megváltozik.

## Az ENIGMA működése



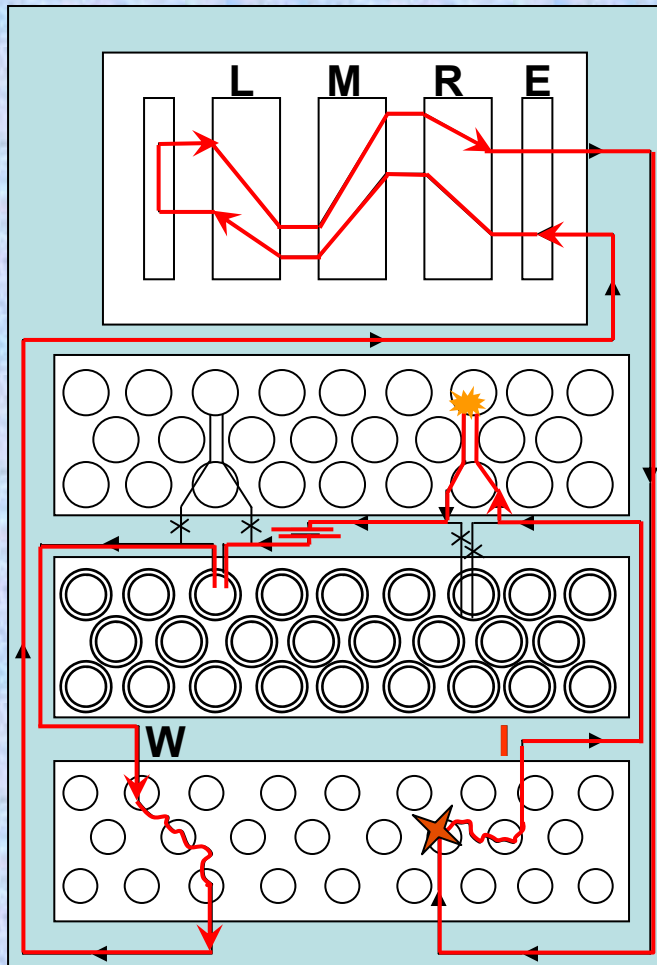
- **Visszalép a rotorokhoz, amelyek mindegyike ismét megváltoztatja.**

## Az ENIGMA működése



- A belépő ablakon keresztül halad, majd a dugaszoló tábla beállításainak megfelelően ismét megváltozik; ez lesz a végső állapota.

## Az ENIGMA működése



- A megjelenítő lámpasoron leolvasható a kódolás eredményeként keletkezett karakter.

A kódolás szimmetriája miatt a kódolt karakter leütése – azonos beállítások mellett – az eredeti karaktert adja vissza.

## Az Enigma bonyolultsága

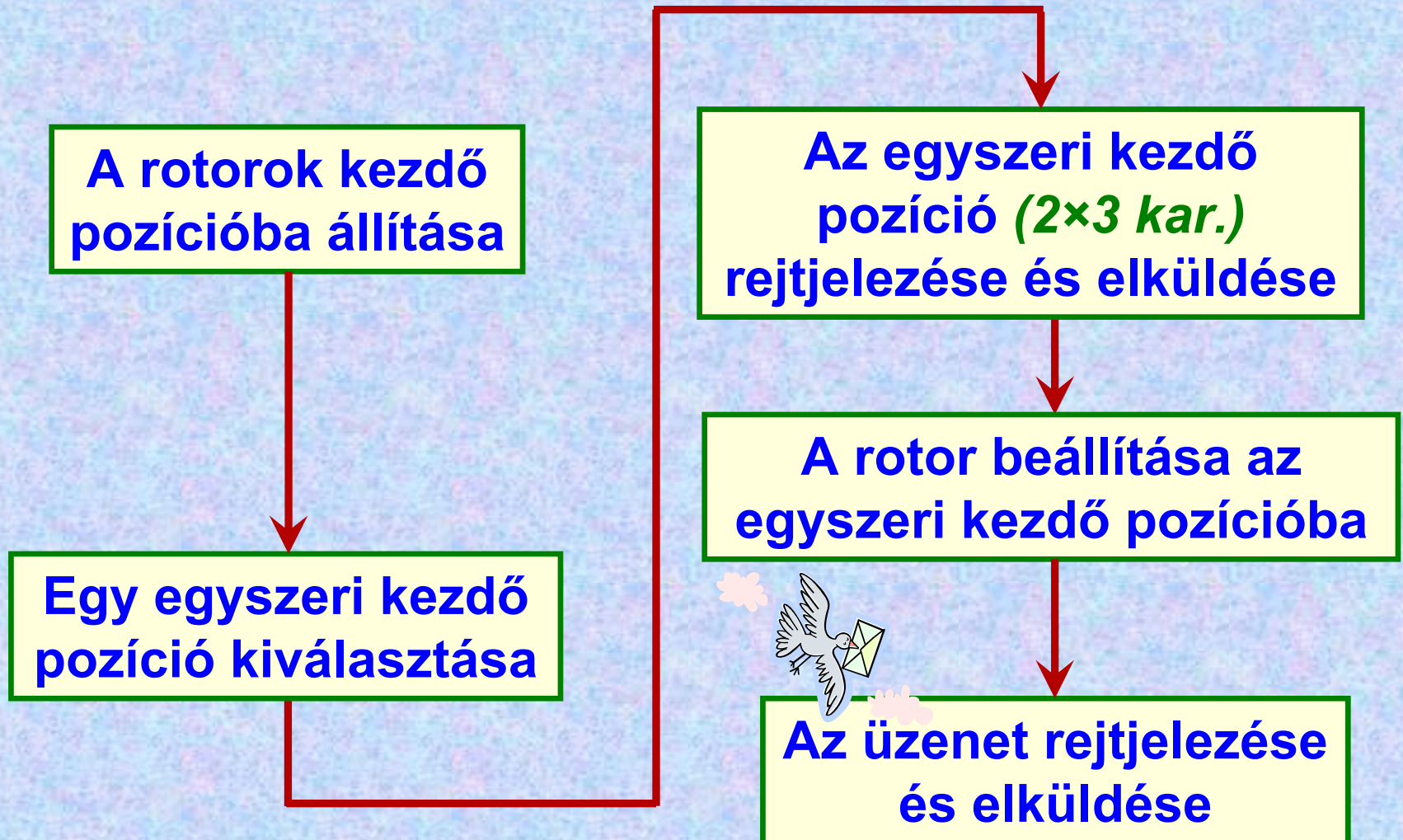
- Rotorok száma: 3 (később 5)
- Lehetséges rotor-sorrendek száma:
  - Kezdetben  $3 \times 2 = 6$
  - Később  $5 \times 4 \times 3 = 60$
- Lehetséges rotor-beállítások száma:  $26^3 = 17576$
- A rotorok lehetséges sorrendjét figyelembe véve:  $6 \times 17576 = 105456$
- A dugaszolótábla variációit figyelembe véve, a gép lehetséges állapotainak száma:  $\sim 1,5 \times 10^{18}$

## A gép beállítása és üzenet küldése

- **A beállítás részei:**
  - A rotorok kiválasztása és elhelyezése
  - A rotorok kezdő pozíciójának beállítása
  - A betűtárcsák beállítása
  - A dugaszolótábla beállítása

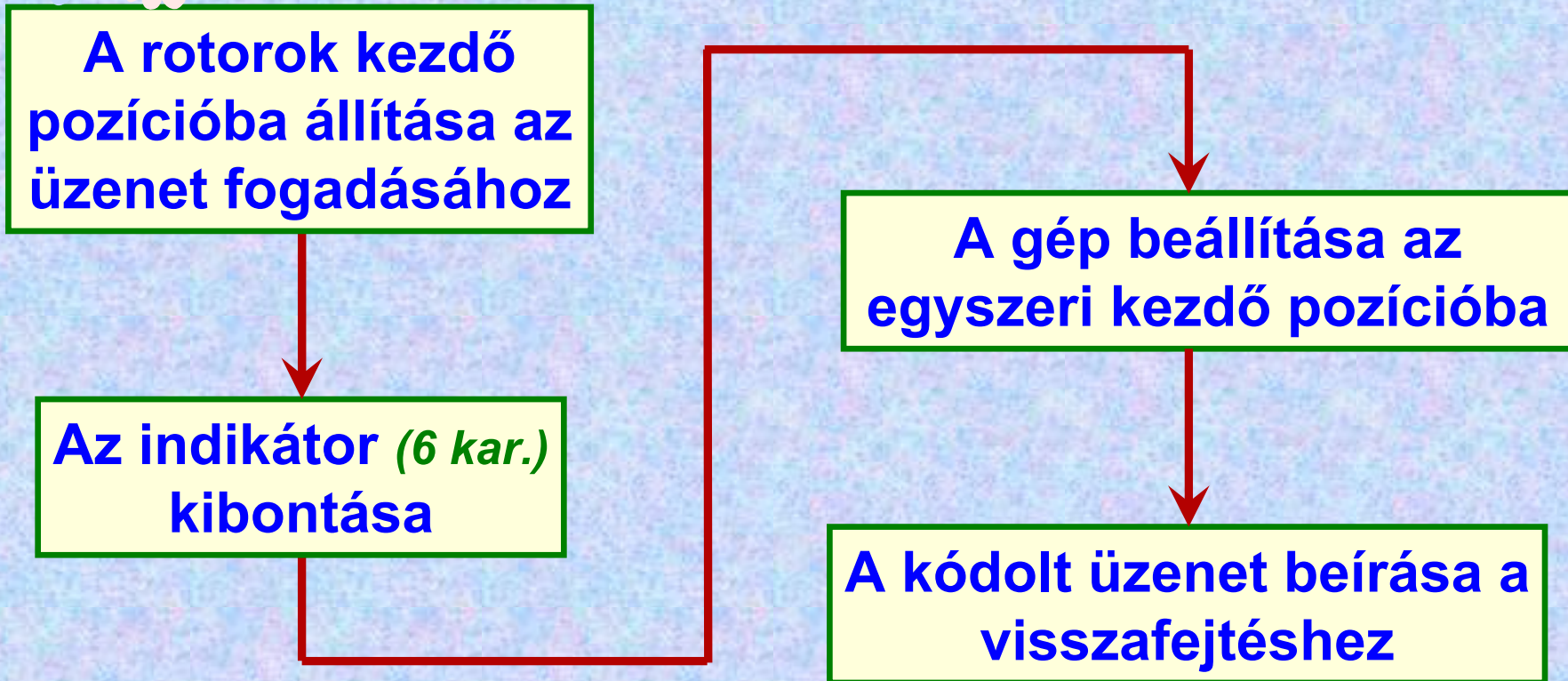


# Az üzenetküldés protokollja (egyszerűbb változat)



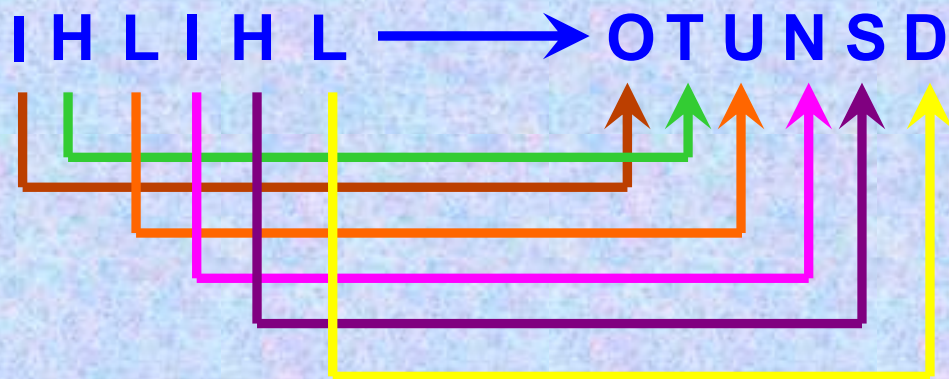


## Az üzenetfogadás protokollja (egyszerűbb változat)

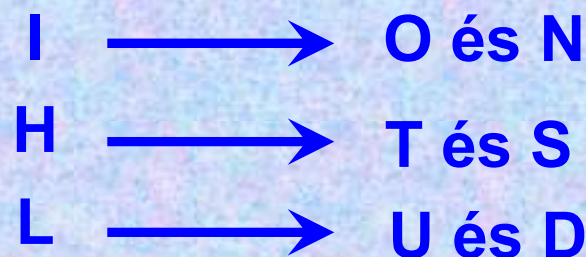


- Az egyszeri kezdő pozíció dupla elküldése az adatvesztés elkerülését szolgálta, de egyben biztonsági részt is jelentett.
- A karakterek leképezéséből következtetni lehetett a kezdő pozícióra *(főként nagyobb számú elfogott üzenet esetén)*.
- Ez adta a kiindulópontot a rotorok belső huzalozásának megfejtéséhez.

Példa a kétszeres elküldésre:



Vagyis:



## A lengyel kódfejtő iroda

- **Alakult: 1919**
- **Első fontos feladata: a szovjet-lengyel háborúban (1919-21) az oroszok rádióüzeneteinek megfejtése**
- **A kódfejtés rendkívül sikeres volt, több száz rádióüzenetet sikerült feltörni**
  - **Az oroszok (ha egyáltalán) a cári rendszerből örökölt kódolási rendszert használták**
  - **A lengyel csapat profi kódfejtőkből állt; a lengyelek üzeneteit az oroszok nem tudták megfejteni**

## A lengyelek gyanút fognak...

- **1928:** egy – tévesen – Lengyelországba kézbesített csomag *(benne egy Enigma)* felhívja magára a titkosszolgálat figyelmét.
- **1932:** Lengyelországban létrehozzák a Kódfejtő Irodát *(Biuro Szyfrów)*. Ennek munkatársa lesz 3 tehetséges, fiatal diplomás: Marian Rejewski *(1905 – 1980)*, Henryk Zygalski *(1906 – 1978)*, és Jerzy Růžicky *(1907 – 1942)*.  
**Feladatuk:** az Enigma kódjának feltörése.
- **1939:** *(még Lengyelország megszállása előtt)* Franciaországba emigrálnak, ahol megosztják információikat a francia és angol elhárítással.
- **1942:** Franciaország német megszállását követően *(több európai országon keresztül)* Angliába emigrálnak, ahol egy lengyel kódfejtő csoport tagjaiként folytatják munkájukat a háború végéig.

## Marian Rejewski



**Felfedezése:** az Enigma belső huzalozásának feltárása.  
A háború után hazatért Lengyelországba, és soha többé nem foglalkozott rejtjelezéssel.

## Henryk Zygalski



**Kidolgozta** a „Zygalsky-kártyákat”, melyeket az Enigma beállításainak kézi megfejtésére használtak.  
A háború után is Angliában maradt, a Surrey-i egyetemen tanított matematikai statisztikát.

## Jerzy Růžicky



**Találmánya:** az ún. „óraszerkezet”, mellyel *(egyes esetekben)* meg lehetett állapítani, hogy melyik a jobb szélső rotor. *(ld. ea. jegyzet!)*  
Fiatalon, hajószerencsétlenségben hunyt el a Földközi-tengeren, még a háború befejezése előtt.

## A napi kezdő pozíciók kiderítését megkönnyítő eszközök

- **Perforált lapok** („Zygalsky-kártyák”)
- **Cyclometer**
- **Ciklus katalógus**
- **„Lengyel bomba”** (*Rejewsky találmánya*)
  - Feladata: a napi kezdő pozíció meghatározása
  - Lényegében 6, sorbakapcsolt Enigmából állt
  - Használatával kb. 2 óra alatt meg lehetett határozni a napi kezdő pozíciót

## Az üzenetküldés protokollja (*bonyolultabb változat*)

Véletlenszerű kezdő  
pozíció ( $K1$ ) és (**3  
karakteres**) üzenetkulcs  
( $K2$ ) kiválasztása

Üzenetkulcs ( $K2$ )  
rejtjelezése a kezdő pozíció  
( $K1$ ) használatával

Üzenet rejtjelezése az  
üzenetkulcs ( $K2$ ), mint kezdő  
pozíció felhasználásával



$K1$ , (a  $K1$  alapján  
rejtjelezett)  $K2$ , és  
az üzenet elküldése



# Az üzenetfogadás protokollja *(bonyolultabb változat)*



**A rejtjelezett K2  
kibontása K1 alapján**



**A rejtjelezett üzenet  
kibontása K2, mint kezdő  
pozíció alapján**

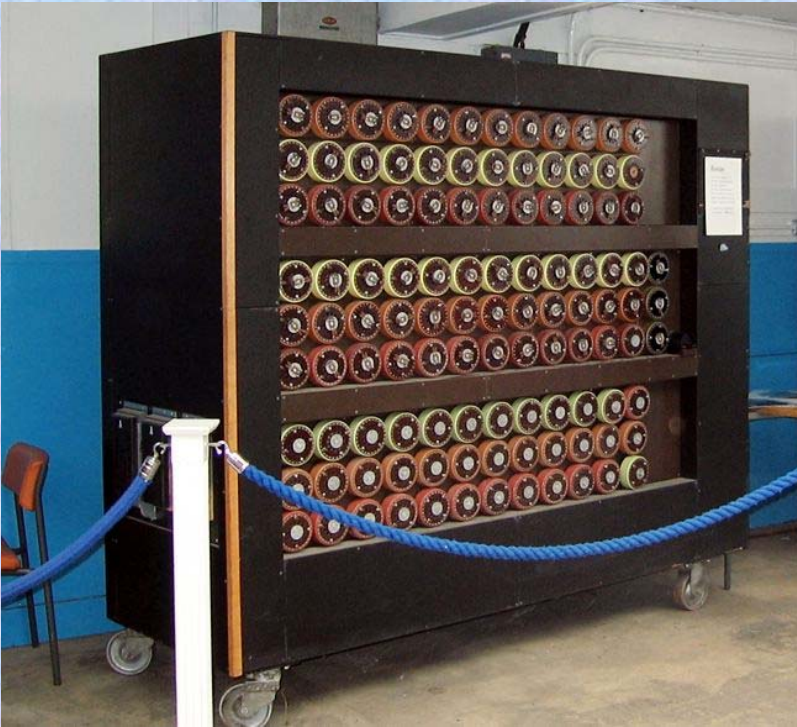
- **A kétszeres rejtjelezés előnyei:**
  - Nincs szükség az egységes kezdő pozíció használatára
  - Minden üzenethez más és más kezdő pozíció tartozik
- **Egyidejűleg megszűnt az üzenetkulcs kettős elküldése**

## A Bletchley park



- **A csodálatos kastély valójában egy szupertitkos létesítményt rejtett, melynek feladata a németek elfogott rádióüzeneteinek minél gyorsabb és hatékonyabb megfejtése volt.**

## A Bomba



- **Alan Turing és Gordon Welchmann találmánya, amely nagyban hozzájárult a háború gyorsabb befejezéséhez.**
- **Segítségével sikerült olyan gépi megoldást találni, amellyel reális időn belül meg lehetett fejteni az Enigma által kódolt üzeneteket.**

## A Bomba fő részei

1. **3 „battery”**: elektromosan szigetelt, de mechanikusan összekapcsolt, egyidejűleg működtethető egységek, melyek a keverőtárcsák sorrendjének cserélgetését helyettesítik.
2. **„Kereszthuzalozás” v. „diagonal board”**: egy 26 db (egyenként 26 erű) kábelből álló vezetékrendszer.
3. **Teszt regiszter**, amely a beállításokra vonatkozó feltételezések helyes, vagy téves voltának jelzésére szolgál.

## A Bomba működési elve

- Az elfogott üzenetekben gyakran találtak olyan szövegrészeket, amelyek megfejtése kézenfekvő, vagy valószínű volt (*pl. címzett, időjárásjelentés, stb.*)
- A feltételezett megfejtést ráillesztve a kódolt szövegre, következtetni lehet arra, hogy az Enigma egy adott beállítása (*rotorok sorrendje és startpozíciója, dugaszolótábla beállításai*) mellett a kódolt szöveg létrejöhet-e?

*Az Enigma sohasem kódol egy karaktert önmagára. Ezért minden olyan feltételezett beállítás, amelyben ilyen előfordul, eleve kiesik.*

## A Bomba működési elve

**Például:**

... P Q W **D** F S B H E Q ... *Kódolt szöveg*  
... R I A **D** O L A N C ... *Megfejtés (?)*

*Nem lehet!*

**De:**

... P Q W **D** F S B H E Q ... *Kódolt szöveg*  
... R I **A** D O L A N C ... *Megfejtés (?)*

*Lehet jó!*

# A Bomba működési elve

Normál Enigma

Nyitott Enigma

Keverő

Nyitott keverő  
vázlata

Tükör

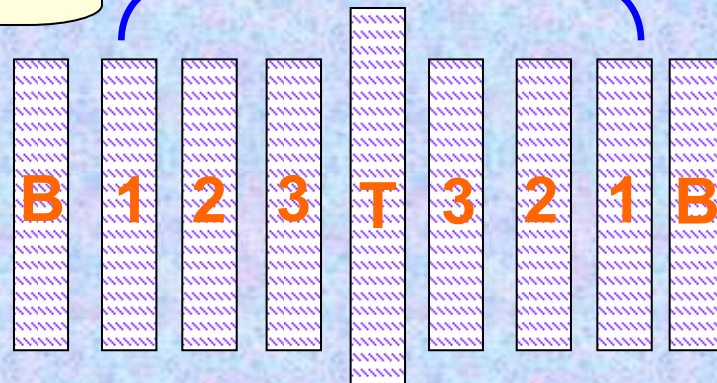
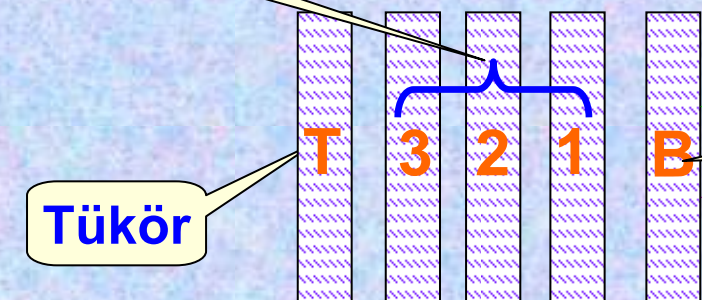
Belépő  
ablak

Áramforrás

Kijelzők

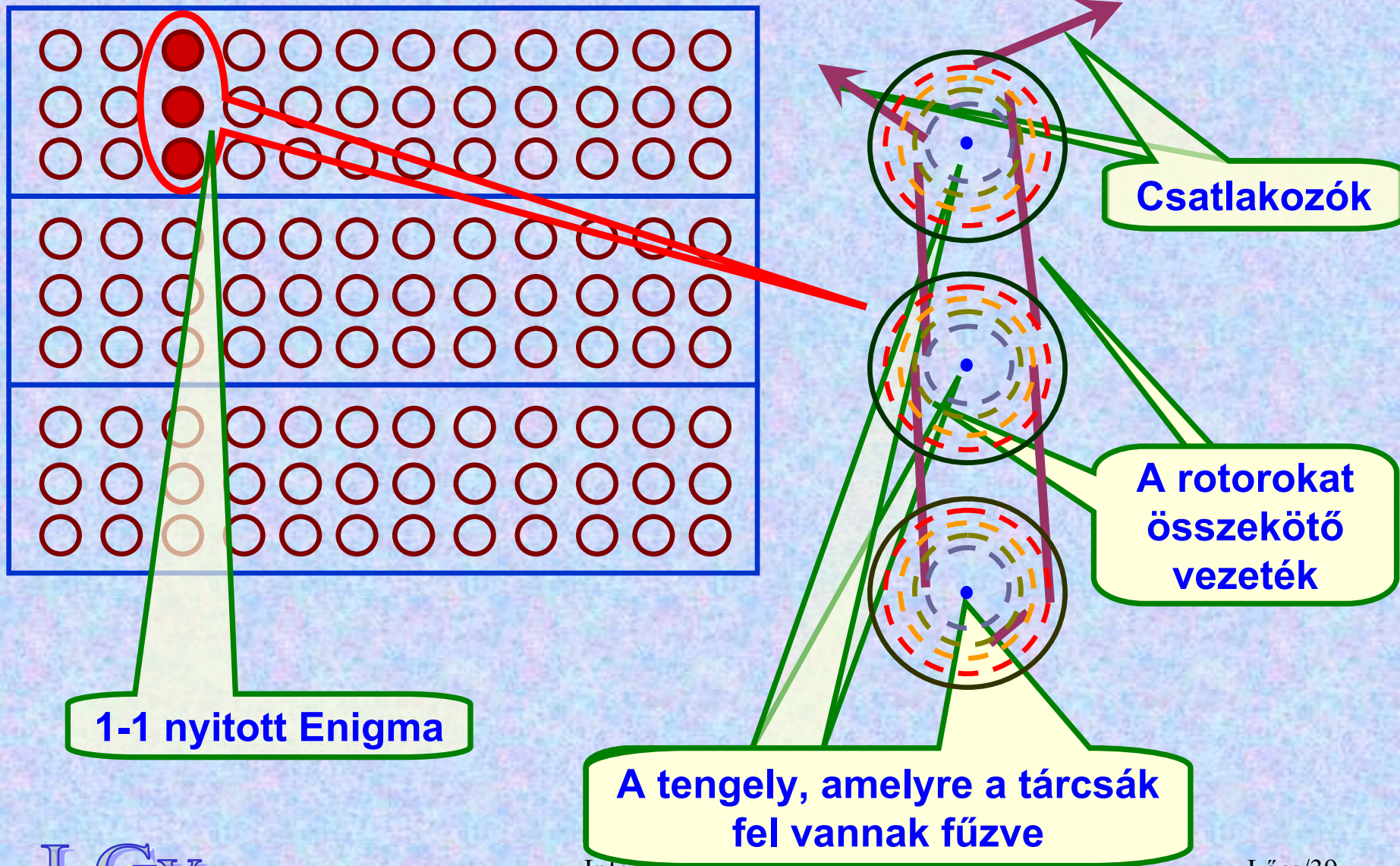
Billentyűzet

Dug. tábla



A „nyitott Enigma” feltalálása lehetővé tette az egyes Enigmák összekapcsolását. A „tükrök” itt már nem megfordítják, hanem átteresztk a jelet.

# A Bomba vázlata

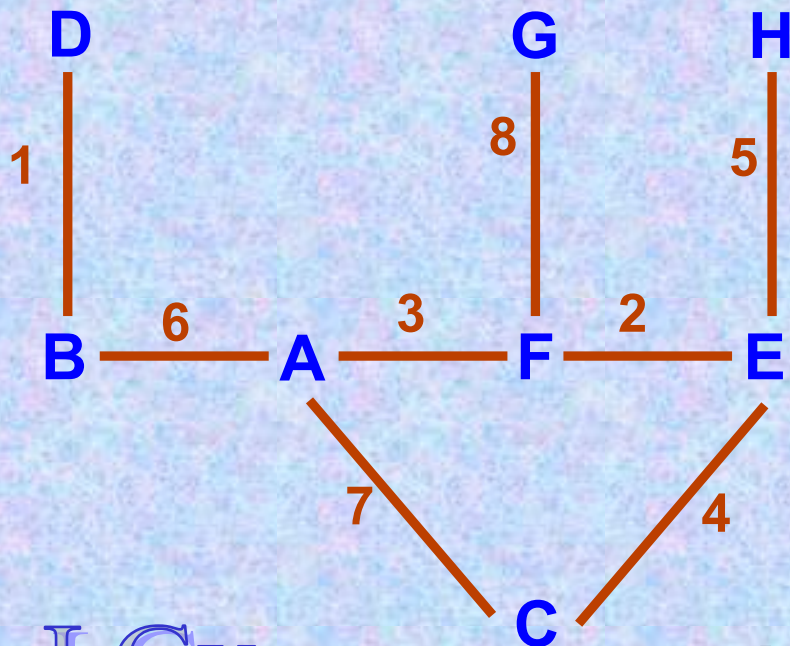




## A Bomba vezetékrendszere

1	2	3	4	5	6	7	8	<i>Sorszám (offset)</i>
B	E	A	C	H	B	A	G	<i>Dekódolt üzenet</i>
D	F	F	E	E	A	C	F	<i>Kódolt üzenet</i>

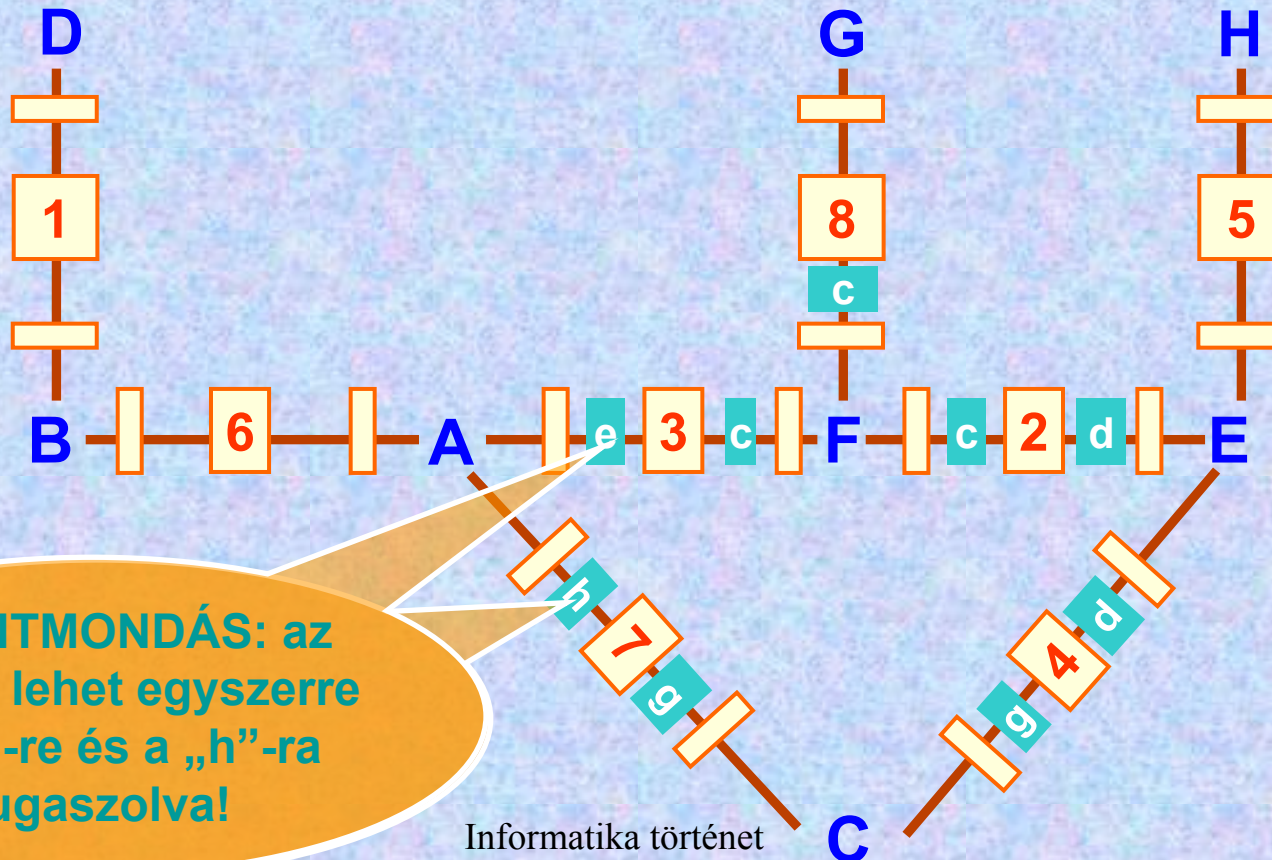
Diagramban („menü”) ábrázolva:



A menü a kódolt üzenet szerkezetét mutatja. Az ábrán látható összetartozó betűknek ( $B \rightarrow D$ ,  $F \rightarrow E$ , stb.) a vezetékrendszerben hídkapcsolatok felelnek meg.

# A Bomba vezetékszere

A menü kiegészítése a dugaszolótábla egy feltételezett (és téves) beállításával, valamint a keverők hatásával



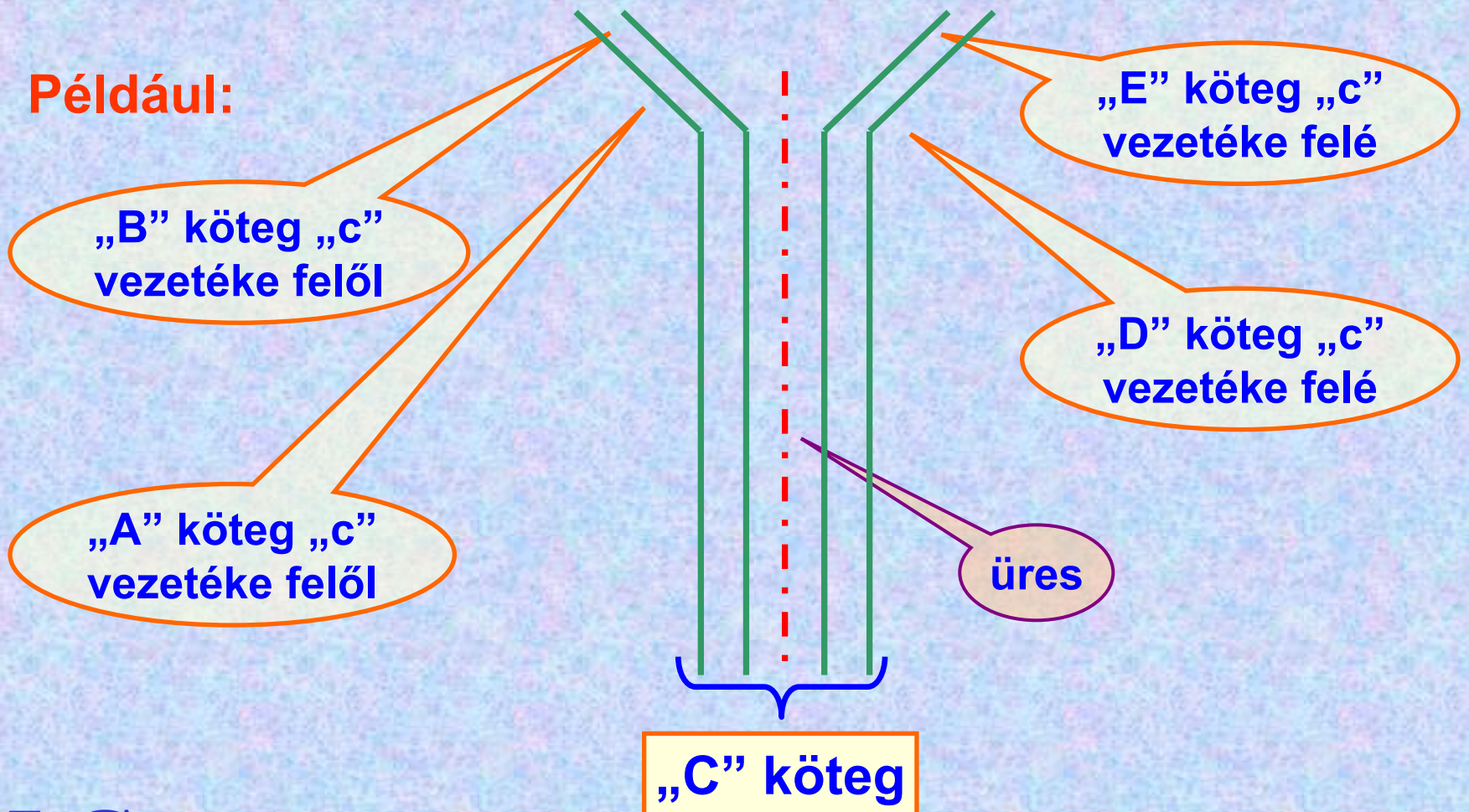
**ELLENTMONDÁS:** az „a” nem lehet egyszerre az „e”-re és a „h”-ra dugaszolva!

## A keresztuzalozás

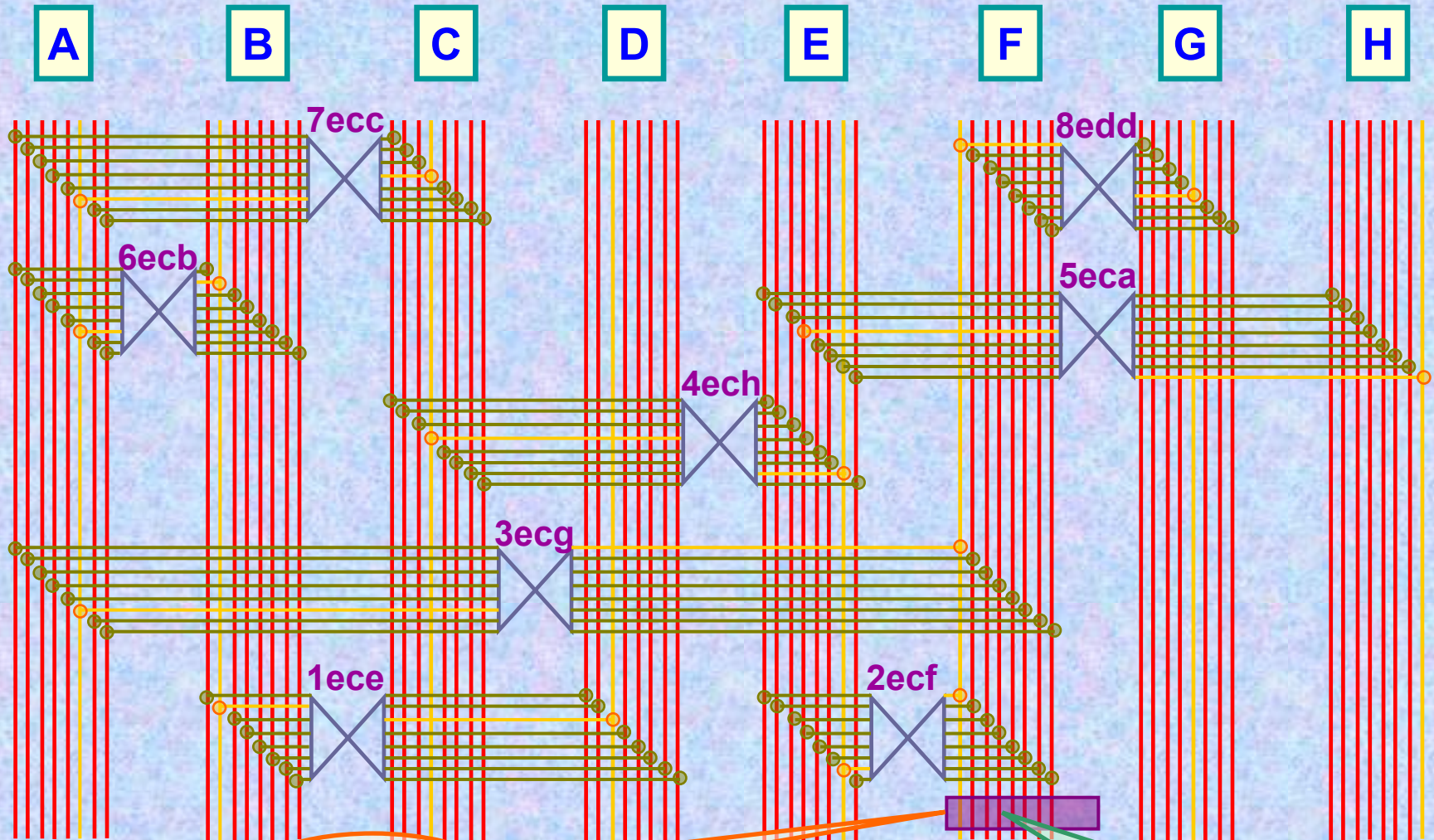
- A keresztuzalozás 26 kábelköteget tartalmaz. Betűjelzésük: **A, B, ..., Z**.
- Az egyes kötegeken belül a betűjelzések: **a, b, ..., z**.
- Mindegyik köteg vele azonos jelű vezetéke „üres”.
- Az egyes kötegekből a nagyobb betűjelzésű köteg felé indul egy-egy vezeték, és az alacsonyabb jelzésűek felől érkezik *(ld. a következő diát!)*.
- Az egyes kötegekből kiinduló vezetékek a többi 25 köteg, velük azonos nevű vezetékével vannak összekötve *(pl.: „P” köteg „q” vezeték → „Q” köteg „p” vezeték)*. Ez nem jelent dupla vezetékezést!
- Az egyes kötegek között hídkapcsolatot létesítenek az egymásnak kölcsönösen megfelelő betűk *(pl. B és E)*.

# A keresztuzalozás

**Például:**

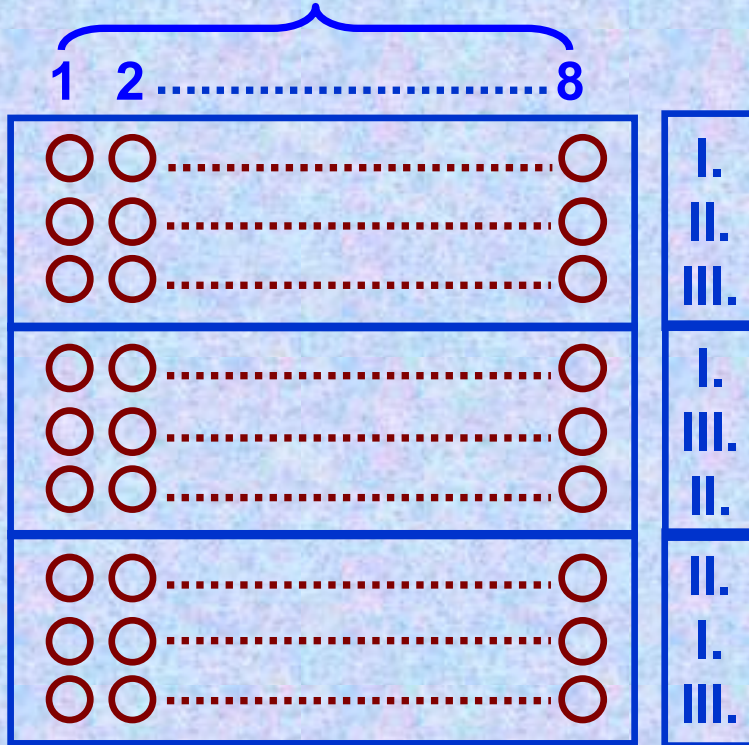


# A keresztuzalozás



## Előkészítés az indításra

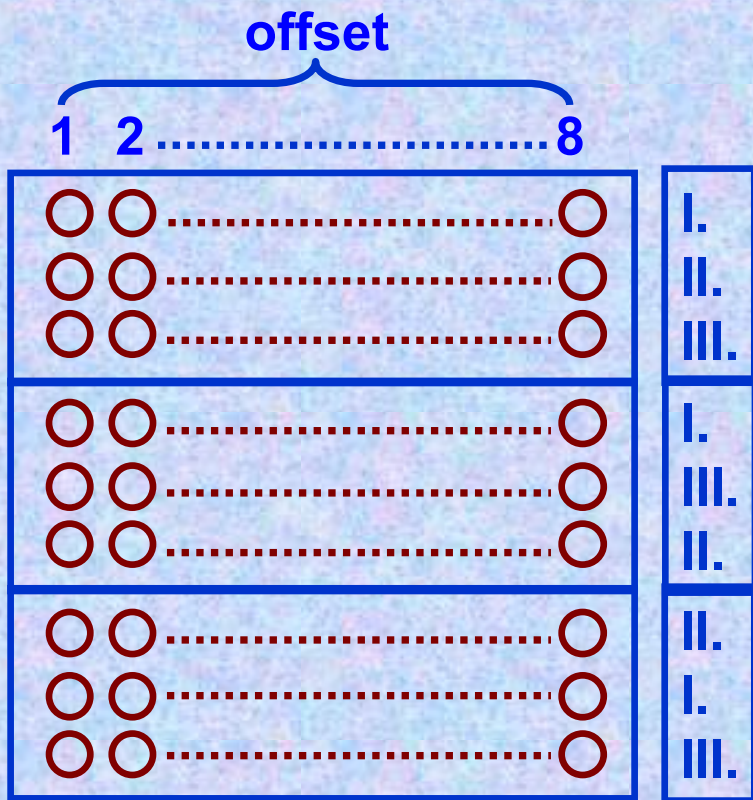
offset



- **A bombán indítás előtt el kell végezni a szükséges beállításokat:**
  - **A használt rotorok kiválasztása (az 5 közül 3, egy kiválasztott sorrendben)**
  - **A dobok felhelyezése a tengelyekre (számuk megegyezik a megfejtett szövegtörredék hosszával).**

**A rotorok sorrendje**

## Előkészítés az indításra



- Az egyes dobok beforgatása a szövegtörödékeknek megfelelő offset pozícióba
- A dobok felhelyezése a 2. és 3. egység (*battery*) tengelyeire. *(Minden adat megegyezik a három egységben, csupán a dobok sorrendje különböző.)*

A rotorok sorrendje

## A teszt regiszter

- **Feladata:**

- Annak eldöntése, hogy egy adott szövegtöredék létrejöhett-e a rotorok sorrendje és beállítása, valamint a dugaszolótábla pillanatnyi beállítása mellett.
- Az eleve kizárható beállítások gyors kiszűrése

- **Lényege:**

- A 26 kábel közül egynek minden vezetéke áthalad a teszt-regiszteren
- A regiszter el tudja dönteni, hogy melyik vezeték van áram alatt, és melyik nincs



## A téves és helyes beállítások kijelzése

- **Ha a vizsgált szövegtöredék *az adott rotorbeállítások mellett nem jöhetett létre*, akkor az áram szétterjed a bomba vezetékhalóján, és a teszt regiszter valamennyi vezetéke áram alá kerül.**

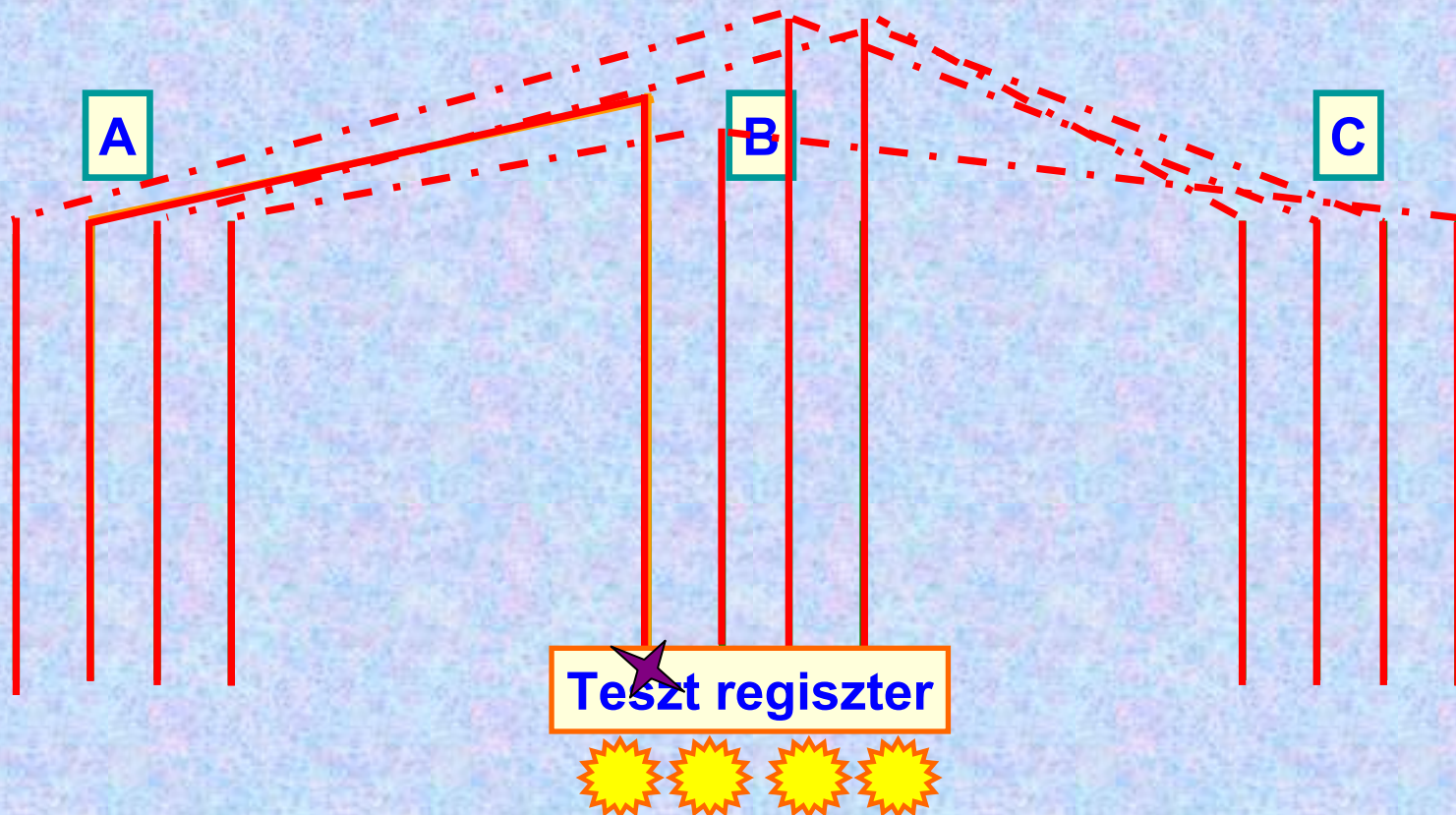
## A téves és helyes beállítások kijelzése

- Ha a rotorbeállítások **helyesek**, akkor két eset lehetséges:
  - Ha a teszt regiszternek egy olyan vezetékét helyeztük áram alá, amelyik nem tartozik a szövegtöredék vezetékeihez, akkor a teszt regiszter 26 vezetéke közül 25 válik „élővé”.
  - Ha ezzel szemben a teszt regiszternek egy olyan vezetékét helyeztük áram alá, amely maga is a szövegtöredékhez tartozik, akkor a 26 vezeték közül csak egy válik „élővé”. Ebben az esetben a gép jelzést ad.

## Illusztráció:

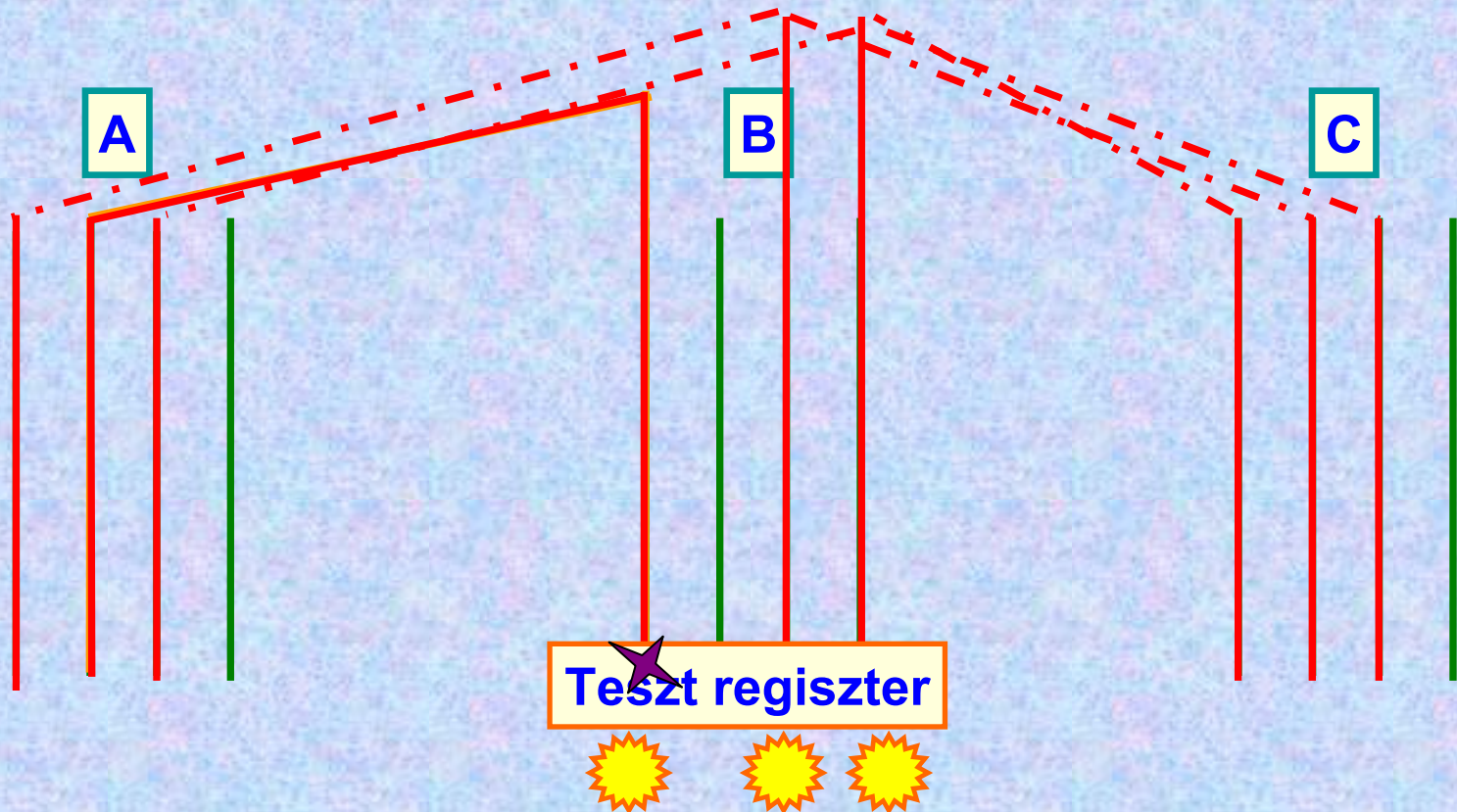
Hibás rotorbeállítás:

*a teszt regiszter minden vezetéke áram alá kerül*



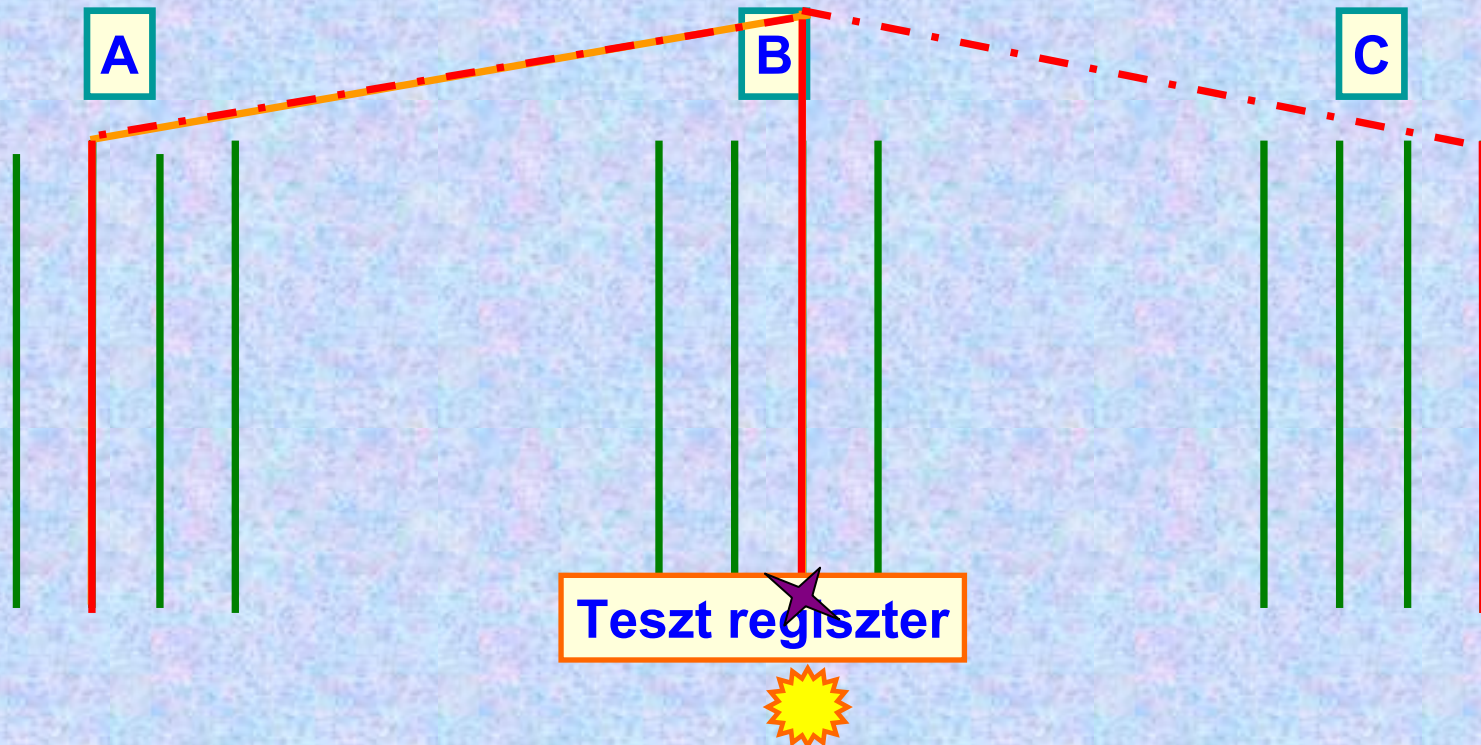
## Illusztráció:

Jó rotorbeállítás, a szövegtöredékhez nem tartozó vezeték:  
*a teszt regiszter 26 vezetékéből 25 kerül áram alá*



## Illusztráció:

Jó rotorbeállítás, a szövegtöredékhez tartozó vezeték:  
a teszt regiszternek 1 vezetéke kerül áram alá



## Epilógus...

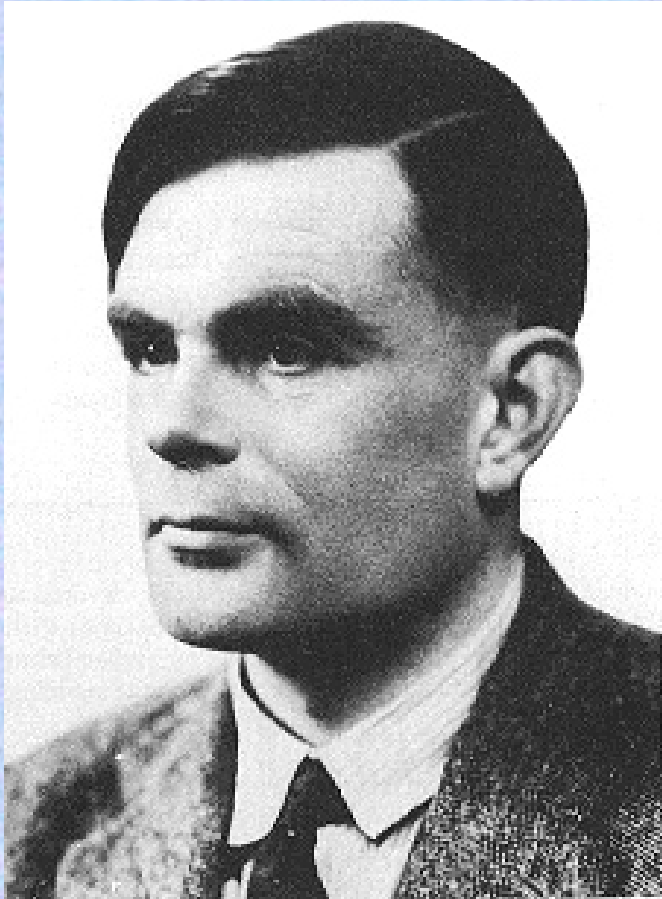
- Az angoloknak a háború végéig sikerült a németek előtt titokban tartaniuk, hogy az üzeneteiket rendszeresen megfejtik.
- Churchill utasítására a háború végén *valamennyi, Angliában fellelhető* bombát megsemmisítették.
- A jelenleg múzeumban látható példány az eredeti tervek alapján készült másolat.
- A kódfejtéssel kapcsolatos államtitkokat csak *kb. 30 évvel a háború után*, a '70-es években oldották fel.
- Bombákat az angolok mellett az amerikaiak is készítettek, ezek azonban némileg különböztek az angol változattól.
- A Bletchley park jelenleg múzeumként látogatható.

# 9. fejezet

## Kódtörés a II. világháború előtt és alatt

### 2. A „Tonhal” projekt

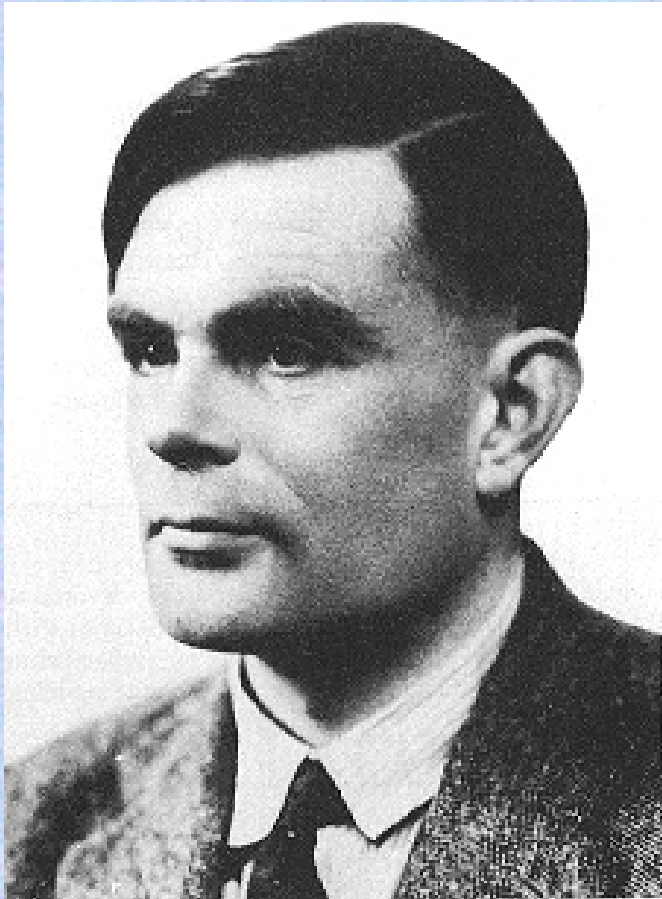
## Alan Mathison Turing (1912 – 1954)



- 1912 jún. 23.-án született Londonban. Kisgyermek korától jeleit adja zsenialitásának
- 1931: a cambridge-i Kings College hallgatója
- 1937: megjelenik híres cikke, amelyben bevezeti az univerzális Turing-gép fogalmát, és újrafogalmazza a Church-tézist

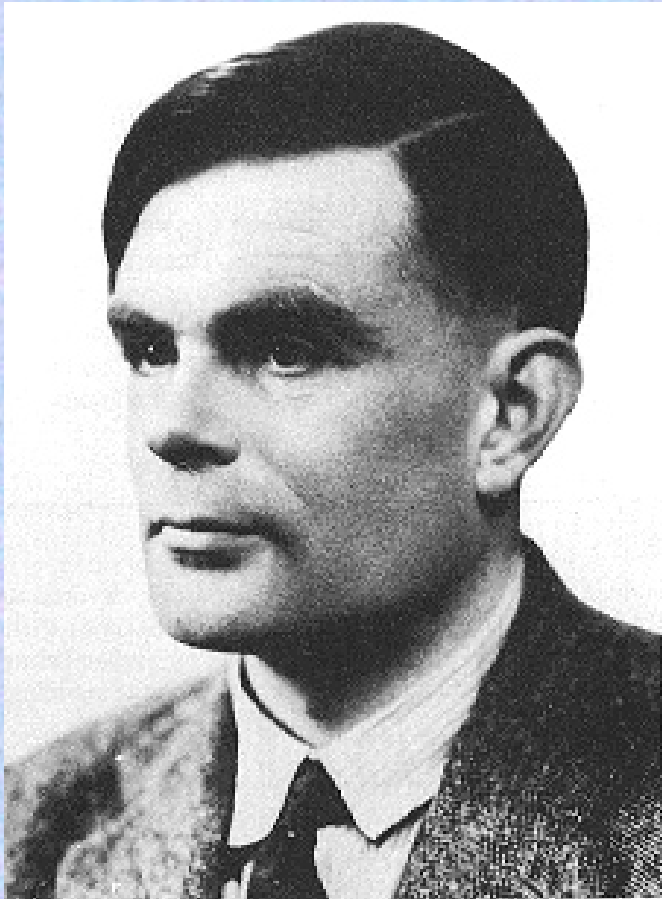


## Alan Mathison Turing (1912 – 1954)



- 1939-45: Bletchley Parkba költözik, ahol a nácik titkosított üzeneteinek megfejtésével foglalkozik.
- 1945-47: meghívják az NPL-be, 1946-ban közzéteszi az ACE tervét.
- 1947-48: visszatér Cambridge-be, ahol neurológiát és fizioológiát tanul. Közben intenzíven atlétizál.
- 1948: tanári állást kap a Manchesteri Egyetemen; itt a Mark-I számítógép szoftverének kifejlesztésén (és közben titokban a GCHQ-nak is) dolgozik.

## Alan Mathison Turing (1912 – 1954)



- 1952: homoszexualitása miatt perbe fogják. Választhat: börtön, vagy „kezelés”. A **GCHQ** számára megbízhatatlanná válik.
- 1954. jún.7.: öngyilkos lesz.
- 2009. szept. 17.: Gordon Brown brit miniszterelnök hivatalosan is bocsánatot kér tőle a „*visszataszító bánásmódért.*”

1966-ban róla nevezték el az informatika legnagyobb kitüntetését, a Turing-díjat

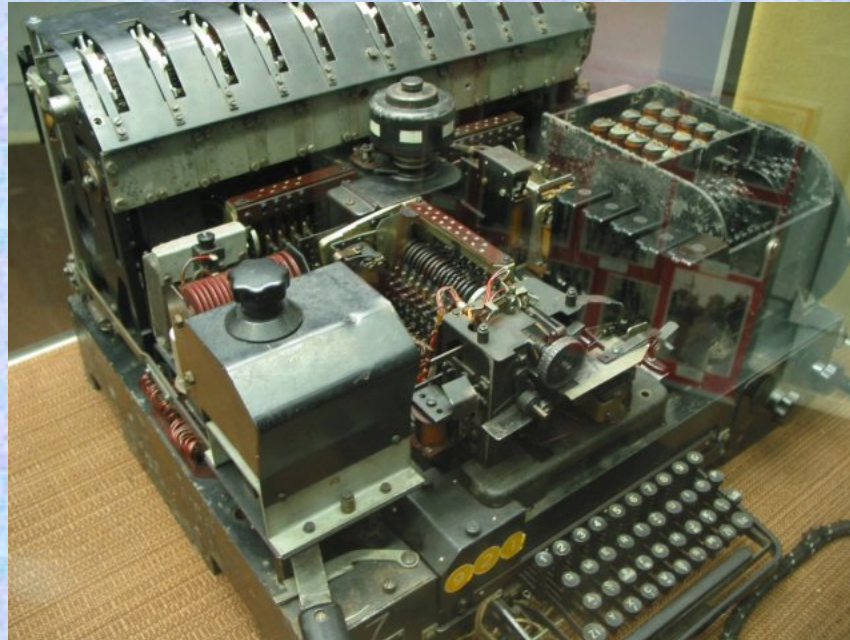
## A Lorenz-kód feltörése

- A Lorenz rejtjelezőgépet a német *legfelső hadvezetés* használta a II. világháború idején.
- A gépet a Lorenzről elnevezett cég gyártotta.
- A kommunikáció az 5 bites *Baudot-kód*-ban történt.
- Az Enigmával ellentétben, a Lorenz-gép *additív* (nem betűhelyettesítéses) kódolást használt.

### A Baudot-kódtáblázat

LETTERS FIGURES	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	4	5	6	7	8	9	0	1	2	3	+	CRASH LINE	FEED	LETTER	FIGURES	SPACE	ALPHABET NOTUS
1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
3	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
4	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
5	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

## A Lorenz rejtjelező gép



## Additív kódolás

- Additív kódolásnál az ötbetűs karakter minden bitjéhez egy keverő bitet (0 v. 1) adunk hozzá, az átvitel nélküli bináris összeadás (XOR) szabályai szerint. Pl:

Kódolatlan betű:	0	1	1	0	0	„I” betű
Keverő bitek:	1	0	0	1	0	„D” betű
-----						
Kódolt betű:	1	1	1	1	0	„K” betű

Bármelyik betűhöz önmagát hozzáadva a **00000** betűt  
(0 karakter) kapjuk!

- A kódolt betűhöz a keverő biteket ismételten hozzáadva, visszacapjuk az eredeti betűt:

Kódolt betű:	1	1	1	1	0	„K” betű
Keverő bitek:	1	0	0	1	0	„D” betű
-----						
Kódolatlan betű:	0	1	1	0	0	„l” betű

- Általában, ha  $N$  a nyílt üzenet,  $R$  a keverésnél használt karaktersorozat, és  $K$  a kódolt üzenet, akkor:

$$K=N+R$$

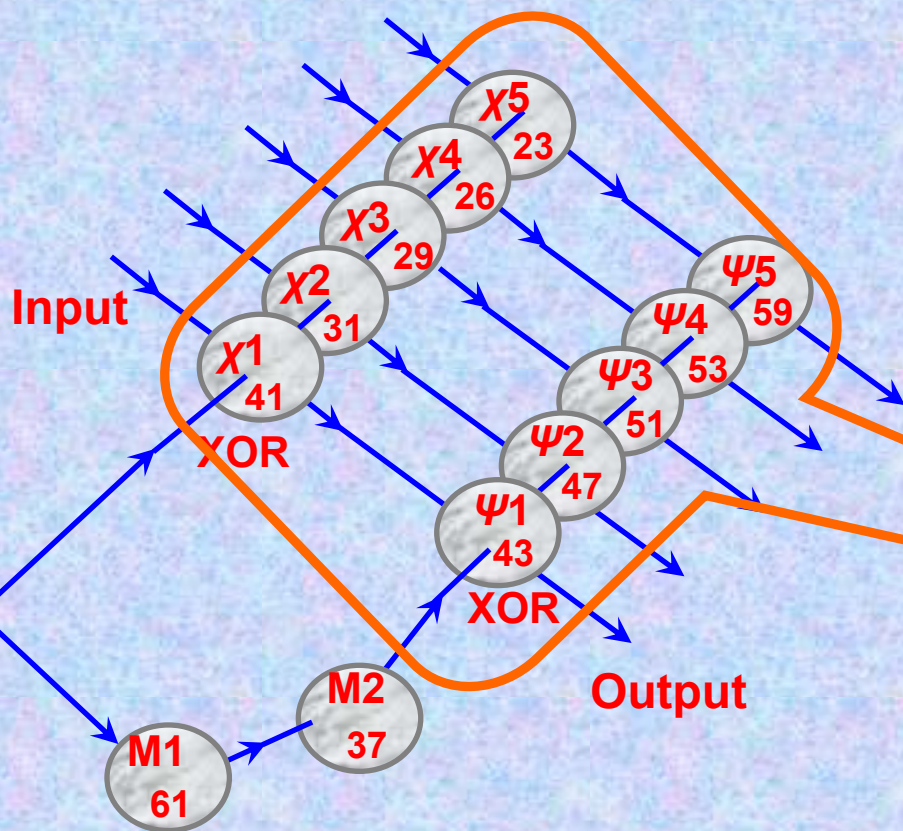
és

$$K+R=N$$

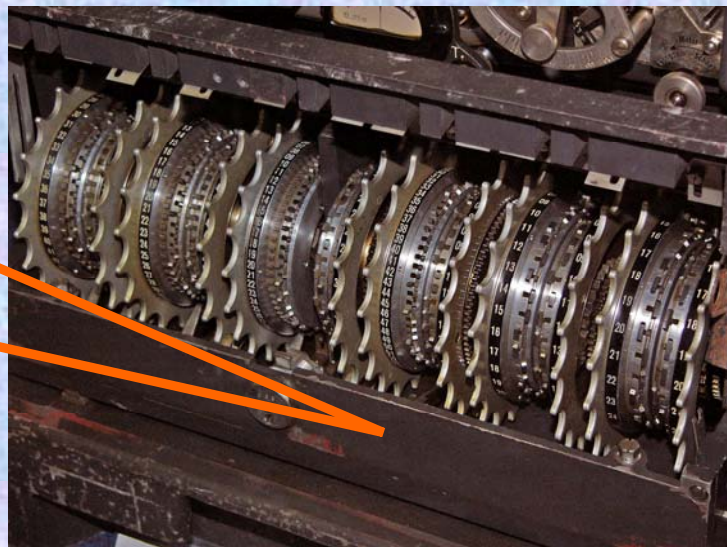
## A rejtjelezett üzenetek feltörésével járó feladatok

- **A Lorenz gép felépítése** (*ellentétben az Enigmával*) kezdetben nem volt ismert.
- **Az egyes üzenetek feltöréséhez ki kellett deríteni a gép kezdeti beállításait** (*ld. Enigma*).
- **Ahhoz, hogy az elfogott üzenet tartalma még releváns legyen, a kódolt üzeneteket a lehető leggyorsabban meg kellett fejteni** (*v.ö. Turing-bomba*).

## A Lorenz-gép logikai vázlata



A tárcsák





## Működési elv

- A gép összesen  $5+5+2=12$  tárcsát tartalmaz ( $\chi$ ,  $\Psi$  és  $M$  tárcsák).
- Funkciójuk:
  - $\chi$  tárcsák: fogadják a kódolatlan szöveg karaktereinek bitjeit, és azokat, a tárcsa beállításától függően, vagy invertálják vagy nem. A tárcsák minden karakter fogadása után továbblépnek.
  - $\Psi$  tárcsák: hasonlóan működnek, de az  $M$  tárcsák vezérlése szerint lépnek, vagy nem lépnek tovább.
  - $M$  tárcsák: vezérlik a  $\Psi$  tárcsákat.

- A 3 tárcsa-csoport működésének eredménye: *a kódolatlan karakterekhez egy-egy pseudo-véletlen karakter hozzákeverése.*
- Ha  $N$  a nyílt üzenet,  $\chi$  és  $\Psi$  a megfelelő tárcsák által generált pseudo-véletlen karakter, és  $K$  a kódolt üzenet, akkor:

$$K=N+\chi+\Psi \text{ (kódolás).}$$

- Az egyenlet mindkét oldalához  $\chi+\Psi$  hozzáadásával:

$$K+\chi+\Psi=N \text{ (visszafejtés).}$$

- A tárcsákba írt számok (41,31, stb.) a teljes körülforduláshoz szükséges lépések számát jelentik. Mindegyik tárcsa indításkor bármelyik állapotba forgatható (Id. Enigma, kezdő pozíció).
- A visszafejtéshez a küldő és a fogadó oldal gépén a beállításoknak azonosaknak kell lenniük.

## Egy hatalmas német baki...

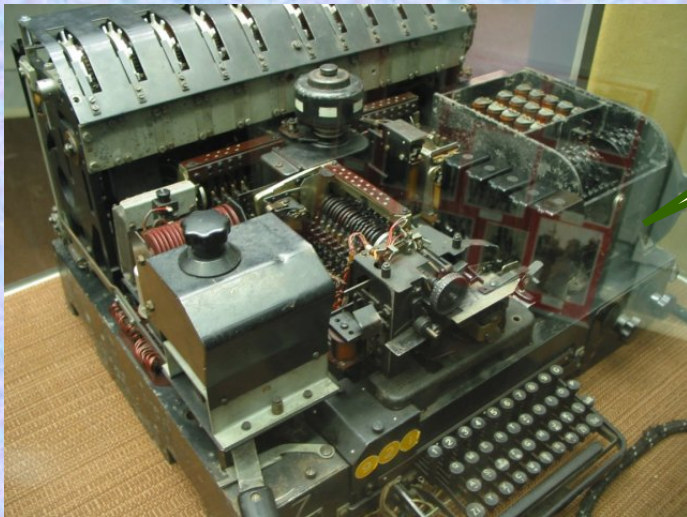
- **1941 augusztusában a németek egy kb. 4000 karakteres üzenetet küldtek, Athénból Bécsbe. Ennek folyamán elképesztő kriptográfiai hibák történtek:**
  - Az üzenet vége felé a fogadó visszajelzett, hogy nem vette az üzenetet, kéri megismételni.
  - A küldő operátor *ugyanazon kezdő pozícióból* elkezdte újra beírni az üzenetet.
  - Azért, hogy kevesebbet kelljen írnia, *az üzenetbe (annak 8. karakterénél, és utána többször is) belemódosított.*

- **A következmény:**
  - Egymás után két, *azonos kezdő pozíciójú és tartalmú, de eltérő szövegű* üzenet keletkezett.
  - A gép mindkét esetben ugyanazokat a keverő karaktereket alkalmazta, különböző karakterekre. Ez lehetővé tette a kódolatlan szövegek és a keverő karakterek szétválasztását.

## A Lorenz-gép szerkezetének megfejtése

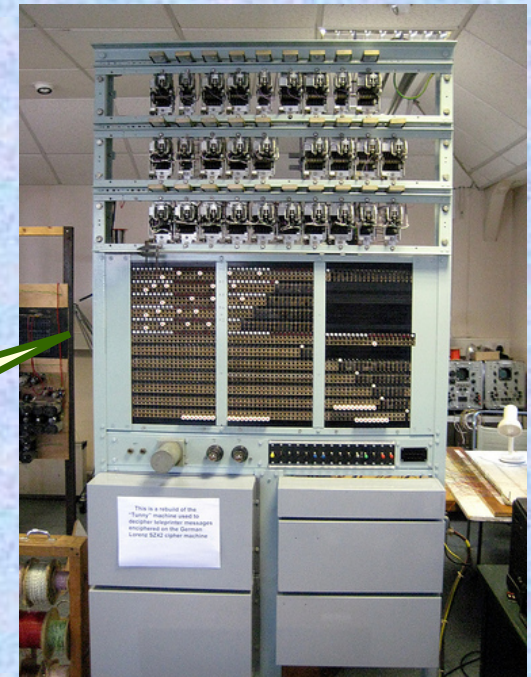
- **A kódfejtők (korábban elfogott üzenetekből) tudták, hogy a németek az adatforgalomban a Baudot-kódot használják.**
- **Mivel a keverés pseudo- (és nem valódi) véletlen karakterekkel történt, ismétlésekre lehetett számítani. A feladat a periódusszámok felderítése volt.**
- **Az első 2 tárcsa ( $\chi_1, \Psi_1$ ) periódusszámát (41 ill. 43) W. T. Tutte fejtette meg. Hamarosan kiderítették a többit is.**

- A Lorenz-gép logikájának és működésének ismeretében (bár „igazi” Lorenz-kódolót sohasem láttak) a szakembereknek sikerült építeniük egy „saját” Lorenz-gépet, melynek a „Tunny” (tonhal) nevet adták.
- A háború után kiderült, hogy az „utángyártott” modell (működését tekintve) azonos, külsejében azonban egyáltalán nem hasonlít az eredetijére.



A német...

...és a brit „kivitel”



## A kezdő pozíció megfejtése

- A gép logikájának megfejtése után bebizonyosodott, hogy a lehetséges beállítások száma kb.  $1,6 \times 10^{19}$ . Ezek szisztematikus kipróbálása kb. *500 millió évig* tartott volna.
- A kódfejtők észrevették, hogy az elfogott üzenetekben a véletlenszerűnél gyakrabban fordulnak elő párosan ismétlődő karakterek. Ennek oka részben a vezérlő karakterek biztonsági okból történő ismétlése volt.

**Példa** (egy német nyelvű fiktív üzenet része):



A kettős karakterek feldolgozására a kódfejtők az eltolás és összeadás módszerét alkalmazták:

9 F L T G R 9 + + K 8 8 R O E M 9 X V I + + L 8 8  
 9 F L T G R 9 + + K 8 8 R O E M 9 X V I + + L 8  
 D 8 4 R T C 8 / H T / Y L B X O B A O X / D F /

Most mindegyik kettős karakter helyén „/” karakter (*null karakter, csupa pont*) áll. Az így keletkező „többlet-pontok” befolyásolják a pontok és keresztek számának arányát.



A pontok és keresztek gyakorisági analízese céljából a kódfejtők *nem* az üzenetek *szavait*, hanem az azonos oszlopban lévő bitek eloszlását vizsgálták („Delta-módszer”):

Példa:

D = X • • X •

8 = X X X X X

4 = • X • • •

1. oszlop

2. oszlop

3. oszlop

4. oszlop

5. oszlop

X

•

•

X

•

X

X

X

X

X

•

X

•

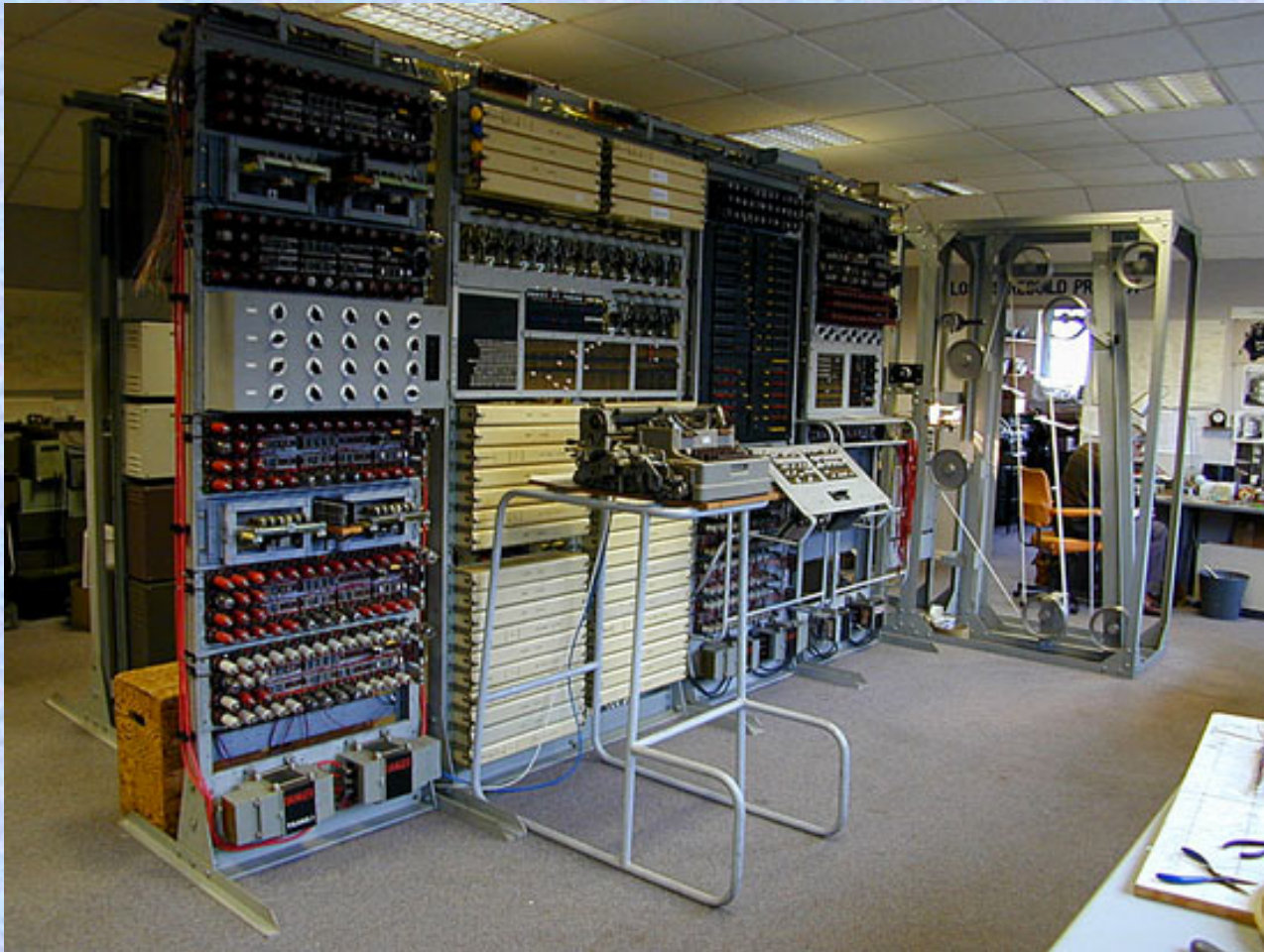
•

•

s.í.t.

- Kezdetben remélték, hogy a statisztikai analízis segítségével *oszloponként* meg tudják határozni a kezdeti beállításokat. Kiderült, hogy ez a németek kódolási szabályai miatt nem lehetséges.
- *Két oszlop együttes analízise* már meghozta a kívánt eredményt: elegendően hosszú bitfolyamok együttes vizsgálatával sikerült megtalálni azokat a kezdeti beállításokat, amelyek mellett az eloszlás szignifikánsan eltér a véletlentől.
- A módszert a *Colossus* gép segítségével automatizálták.

# A Colossus



## A Colossus működése

- Az elfogott üzeneteket tartalmazó végtelenített lyuk-szalagot egy optikai olvasó olvasta be.
- A gép az 1. és 2. oszlop karakterpárjaira alkalmazta azt az általa generált logikai kifejezést, amelynek segítségével ki lehetett mutatni a véletlentől való szignifikáns eltérést.
- Mivel az első 2 oszlop  $41 \times 31$ -féleképpen párosítható, egy **3000** szavas üzenet feldolgozása  $41 \times 31 \times 3000 \sim 3,8$  millió összehasonlítást igényelt. A beolvasás **5000 karakter/perc** sebessége mellett ez **kb. 13 percet vett igénybe**.
- A további oszlopokat gyakran kézzel, vagy több Colossus együttes működtetésével fejtették meg.

## A Colossus története

- **Tommy Flowers mérnök tervezte.**
  - A prototípus (*Mark-1*) 1944. februárjára készült el.
  - A 2., továbbfejlesztett példány (*Mark-2*) 1944. júniusában állt üzembe, éppen a normandiai partraszállás idején.
  - A Mark-1-et hamarosan átépítették, a *Mark-2 tervei szerint*.
  - A háború végéig **10** gépet építettek.
  - 8-at** közvetlenül a háború után szétszereltek.
  - 2** példány **1960-ig** működött, majd azokat is szétszerelték, sőt még *a terveket is megsemmisítették*.

- A Colossus kb. **2 évvel** korábban készült el, mint az **ENIAC**.
- A rendkívül szigorú titkosítás miatt az informatika fejlődésére viszonylag csekély hatása volt.
- Programozható, elektronikus, digitális számítógép volt, de **nem Turing-teljes**.
- **1996-ban (a Bletchley park számára)** egy példányt újraépítettek. Ehhez jelentős segítséget nyújtottak az **USA birtokában lévő** titkos, de a '70-es években felszabadított iratanyagok.

**A Turing-teljesség ebben az időben nem volt szempont, mivel a gépeket rendszerint egy adott feladatkör megoldására építették.**

# 10. fejezet

## Számítógép generációk

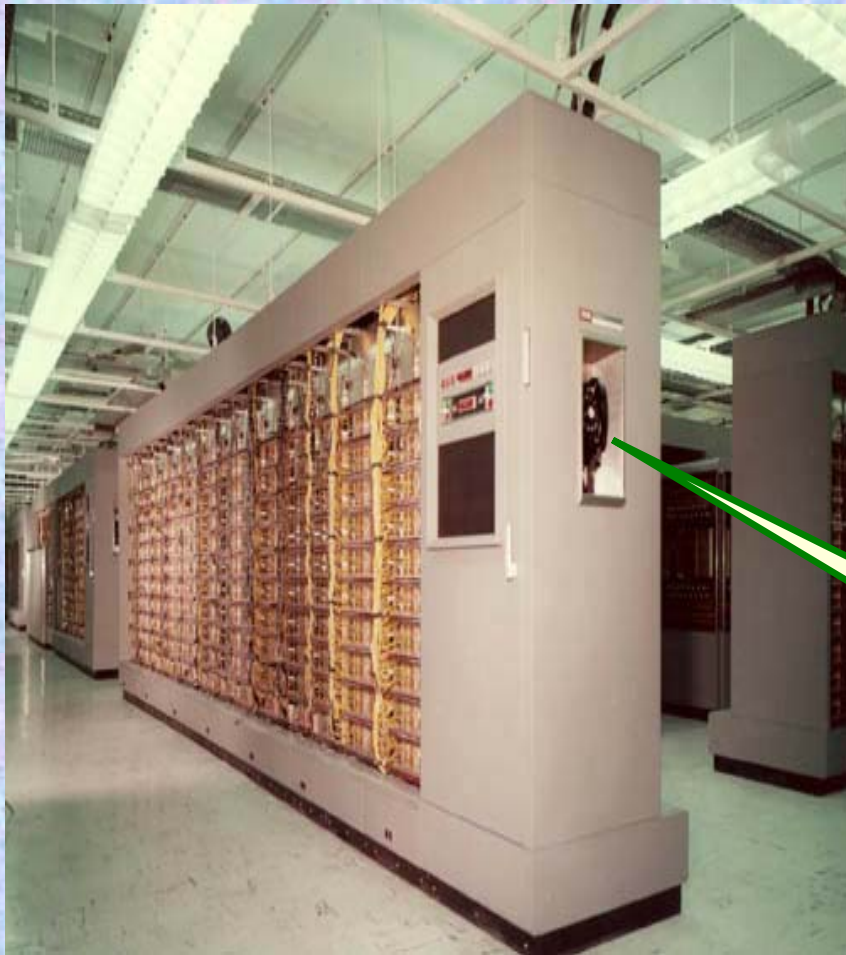
Az ötödik generáció

## A korai számítógépek generációi

- A kezdetektől kb. a 90-es évekig a besorolás alapja az alkalmazott technológia volt:
  - 1. generáció: *elektroncsövek*
  - 2. generáció: *diszkrét tranzisztorok*
  - 3. generáció: *integrált áramkörök*
  - 4. generáció: *mikroprocesszorok*
- A generációváltások hatásai:
  - a fizikai méretek, a területigény, a hőtermelés és az árak csökkenése
  - a műveleti sebesség és a tárhelykapacitás növekedése
  - a felhasználói kör és az alkalmazási területek bővülése



# Néhány példa az első négy generáció gépeire



## 1.: a SAGE földi irányító és léghárító rendszer (USA)

- épült: 1954-1963
- költségei felülmúlták a Manhattan-projectét
- 24 légi irányító és 3 harci központ
- működött 1983-ig
- számítógépek: A/N FSQ-7 (IBM)

Telefon

- kb. 60 000 elektroncső
- 3 megawatt energiafelvétel
- program: kb 500 000 sor

- számos később elterjedt eszköz (modem, egér, interaktív display stb.) használata

## Néhány példa az első négy generáció gépeire



### 2.: ERMA, az első banki alkalmazásokra épült számítógép

- épült: 1950-1955
- naponta 792 000 bizonylat feldolgozására volt képes
- az ilyen típusú gépeket a 70-es évek elejéig használták
- az optikai eszközökkel történő adatfeldolgozás kezdetét jelentette
- Műszaki adatok:
  - energiafelhasználás: 150 kw
  - Súly: kb. 10 tonna

# Néhány példa az első négy generáció gépeire

## 3.: PDP-8, a miniszámítógépek korszakának kezdete

- gyártva: 1965-1990  
(különböző változatokban)
- az első tömegesen  
elterjedt miniszámítógép-  
család
- ára: 16 000 \$
- jellemzői:
  - kb. hűtőszekrény  
méretű
  - légkondicionálást nem  
igényel



## Néhány példa az első négy generáció gépeire



### 4.: **CRAY-1, a szuperszámítógép**

- **tervezője: Seymour Cray (1925-1996)**
- **gyártva: 1976**
- **ára: 8,8 millió \$**
- **szóhossz: 64 bit**
- **központi tároló: 8 Mbyte (1 Mword)**
- **műveleti sebessége: névlegesen 160 mflop/sec**
- **a Cray-1 a maga idejében a világ leggyorsabb számítógépe volt**
- **2007-es adatok szerint, a mai leggyorsabb PC-k sebessége mintegy 130-szorosa a Cray-1 - nek**

# Egy új korszak megalkotásának programja: az ötödik generáció (FGCS)

- A programot Japán indította 1982-ben
- Cél: a „jövő számítógépének” megalkotása, amely:
  - emberi nyelven tud kommunikálni
  - képes az emberihez hasonló intelligens viselkedésre (pl. *orvosi diagnosztika, játékok, szövegfordítás stb.*)
  - architektúrája adat- és tudásbázisokon alapul
  - logikai programozási nyelvet használ
  - működésében jelentősen épít a párhuzamosságra

# Egy új korszak megalkotásának programja: az ötödik generáció (**FGCS**)

- **Költségei:**
  - 450 millió dollár ipari befektetőktől
  - ugyanennyi állami támogatás
- **Ütemterv:**
  - 3 év: előkészítő kutatások
  - 4 év: egyes alrendszerek kifejlesztése
  - 3 év: működő referenciarendszer létrehozása

## Megvalósítás és problémák

- **A megvalósításhoz választott programozási nyelv: PROLOG**
  - a nyelv tulajdonságaiból adódóan fennáll a kombinatorikus robbanás veszélye
  - nem támogatja a párhuzamos programozást, ezért azt „át kellett szabni” a párhuzamosság kezelésére
  - a **LISP** szerencsésebb választás lehetett volna

## Megvalósítás és problémák

- **Műveleti sebesség:**
  - a technológia fejlődése miatt a lényegesen olcsóbb negyedik generációs gépek teljesítménye utolérte, sőt meghaladta az ötödik generációsokét
- **Az Internet megjelenése és elterjedése**
  - már nem volt érdemes nagyobb tudásbázisokat egy számítógépen tárolni, a szükséges információkat a világhálón meg lehetett találni
- **Értékesítési problémák**
  - az elkészült munkaállomásokra nem volt megfelelő piaci kereslet



## A történet folytatódik (?)



Az egyik PIM-gép

- A program keretében öt **PIM** (*Parallel Inference Machine*) és néhány alkalmazás készült el
- 1992: a fejlesztés végetért, bár céljai csak részben teljesültek
- Számos, ebben megfogalmazott elképzelés később a technika részévé vált

## A történet folytatódik (?)



Az egyik PIM-gép

- **Hatására Japánon kívül több ország is indított hasonló kutatás/fejlesztési programot (USA, Anglia, Németország)**
- **Lezárulta után többen már a *hatodik generációt* emlegették. Az erre vonatkozó elképzelések azonban nem álltak össze egységes koncepcióvá**

## Töprengés...

**Sokan vitatják, hogy az ötödik generációs program kudarc volt, vagy csupán megelőzte a korát?**

### Kérdések:

- 1. Észrevesszük-e előre, hogy generációs váltás készül, vagy csak akkor, amikor már folyamatban van, esetleg meg is történt?*
- 2. Kb. 150 év távlatából el tudjuk-e dönteni, hogy Babbage Analitikai gépének terveit kudarcnak, vagy a mai informatika alapjának kell-e tekinteni?*

# 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)

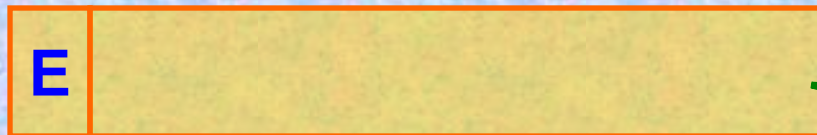


5 bit

10 bit

Spec. indikátor bit

Rövid adatszó:



16 bit

Fixpontos, 16 bit

Hosszú adatszó:



magasabb helyiértékek

alacsonyabb helyiértékek

35 (!! ) bit

„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 $\geq 0$ , 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 kódtagon 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
- **$\pi$  (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 <b>300</b> -as címtől)
[SUBR 3]	(a harmadik szubrutin utasításai)
T 400 K	(a főprogram a <b>400</b> -as címtől)
[FŐPROGRAM]	(a főprogram utasításai)
E 400 K P F	(a program elindul a <b>400</b> -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*)

# 12. fejezet

**A magasszintű programozási  
nyelvek kialakulásának kezdetei**

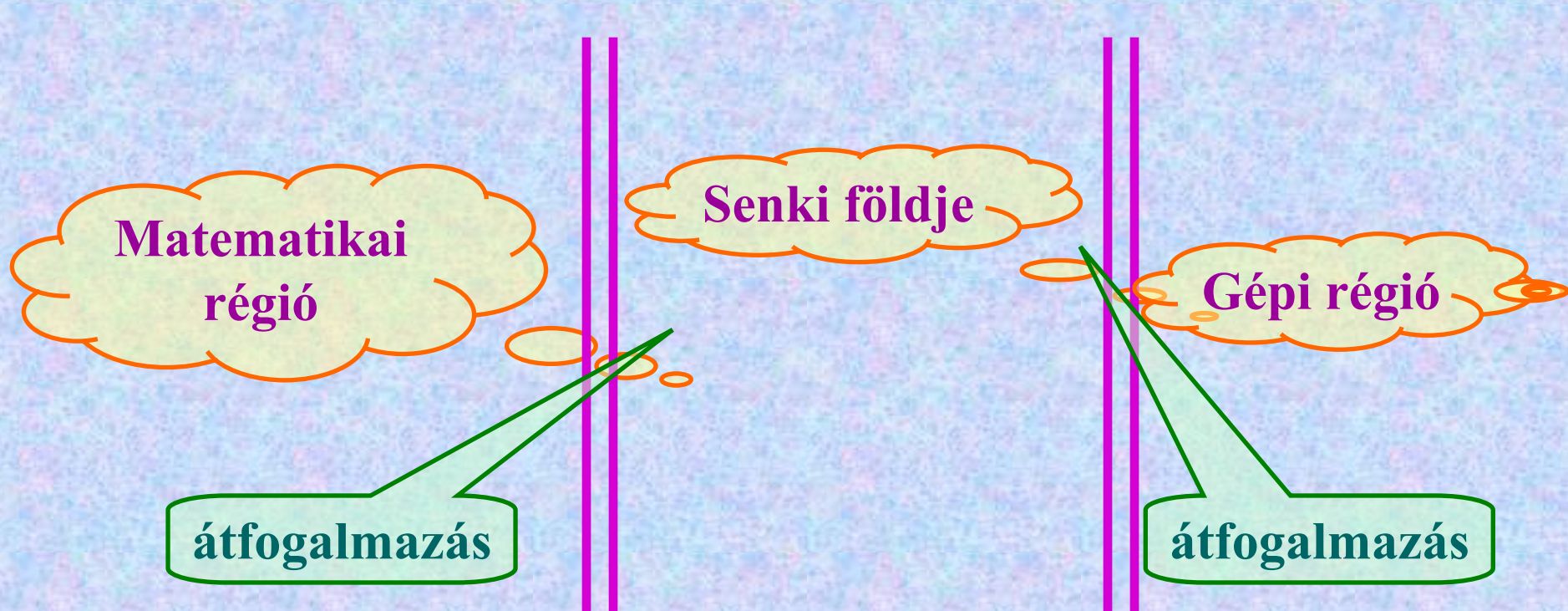
**Plankalkül, Folyamatábra, Böhm nyelve,  
Manchester Autokód, ős-FORTRAN**

## 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**

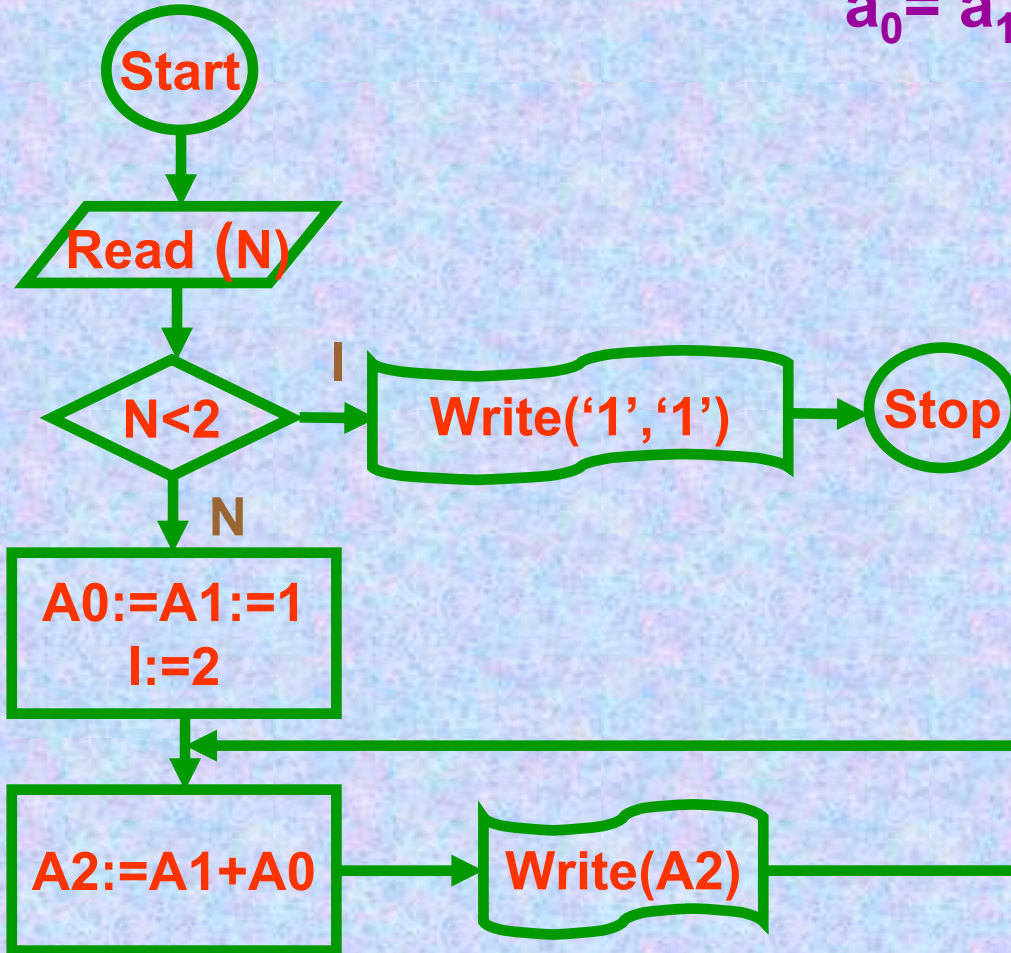


## Neumann és Goldstine megközelítése: a folyamatábra (flow diagram)

- 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:**

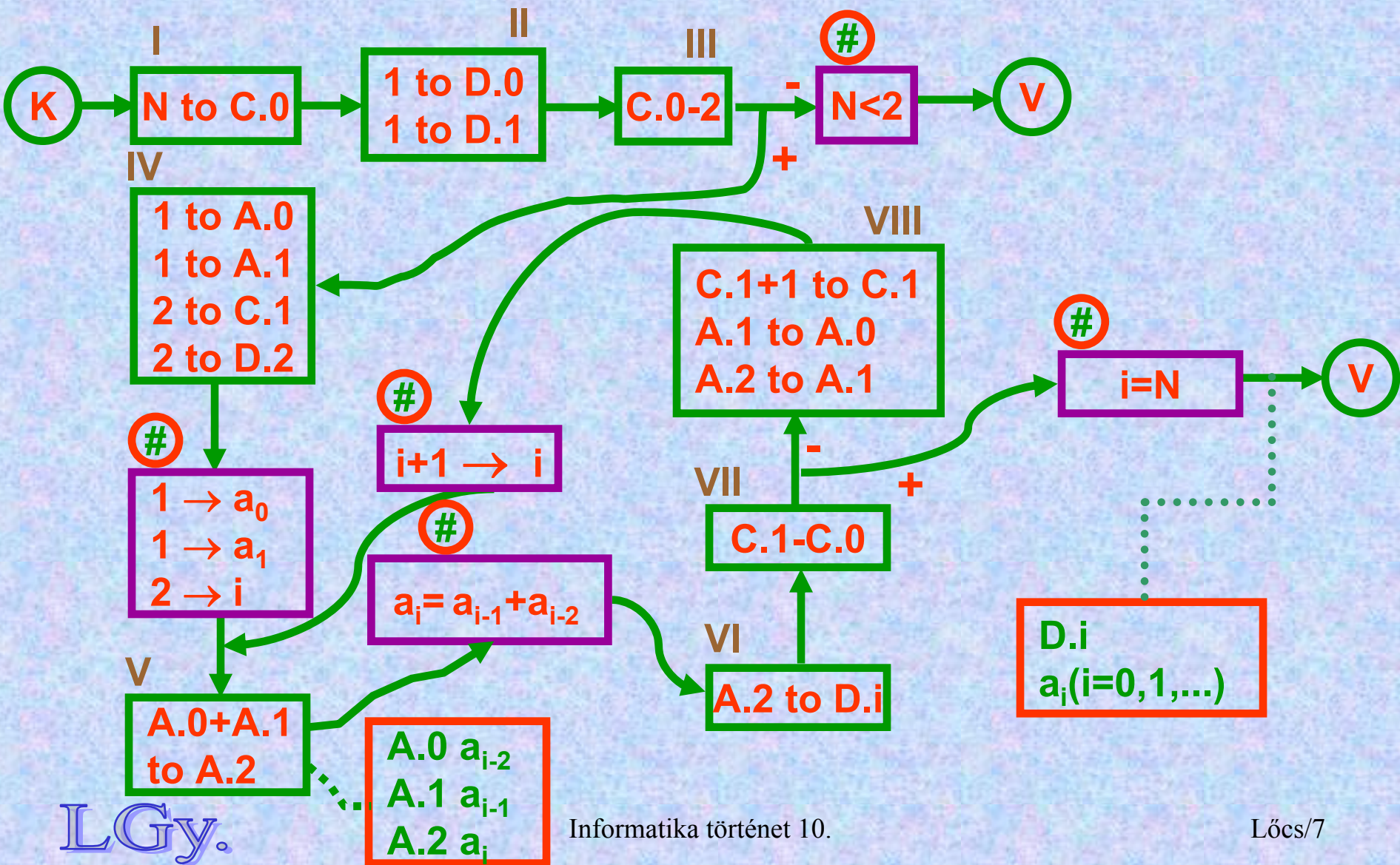
$$a_0 = a_1 = 1; a_n = a_{n-1} + a_{n-2} \quad (n = 2, 3, \dots)$$

**Jellemzője:**

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

# Egyszerű programozási feladat a Fibonacci-számok generálására

## Neumann és Goldstine stílusában:



- **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 andráskereszttel („#”) 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 hozzáférhető*k)

## 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 – megválik 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**

$\pi'$   
 $\pi$  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}*  
 $7\ v98 = v_{n1} \times v_{n2}$     *{A „7” címke;  $v_{n1}, v_{n2}$  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

# 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)



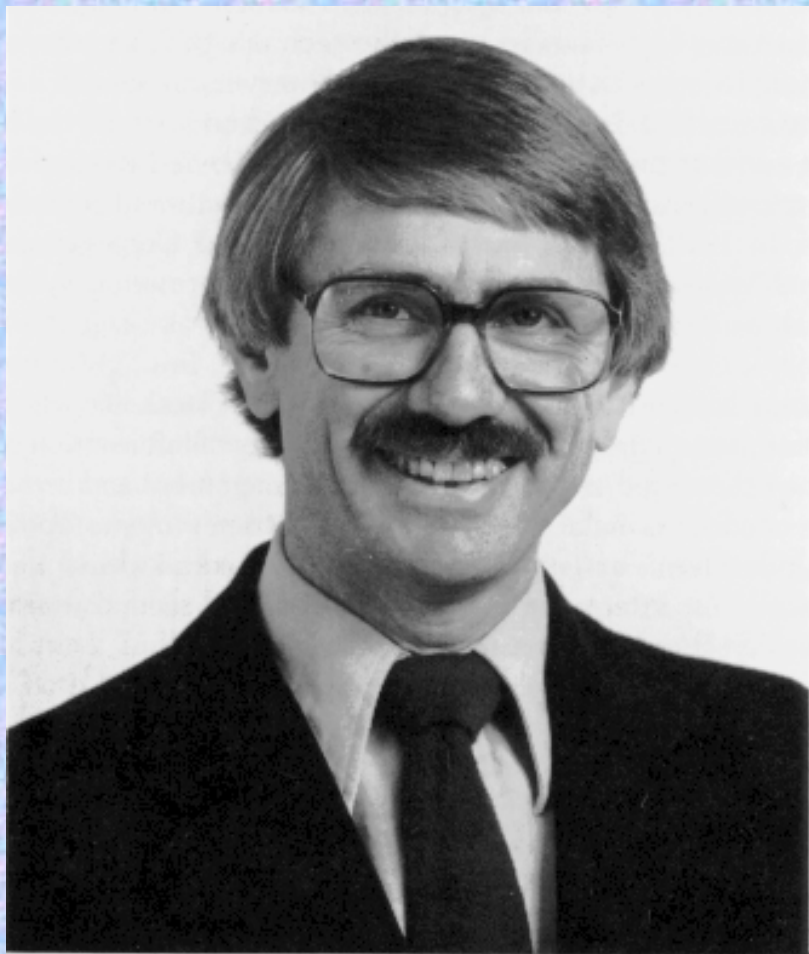
- 1926 máj. 31-én született Budapesten.  
Apja nagykereskedő
- 1940: az apa a nácik elől New Yorkba menekíti a családot
- 1945: Princetonba, majd Los Alamosba kerül, itt Feynmannel és Neumannal dolgozik együtt

## 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 dobmemória*)
- 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

# 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)



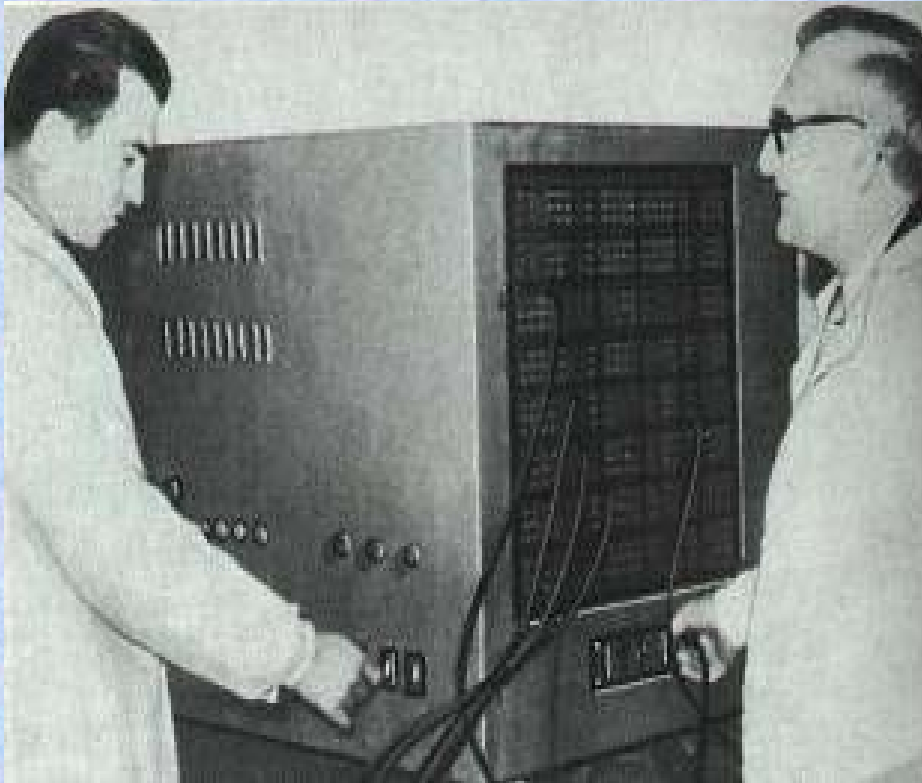
- 1930-ban adjunktus a Bolyai Intézetben
- 1947-ben megörökli *Riesz Frigyes* tanszékét, és egyetemi tanárrá nevezik ki
- 1949-ben a Magyar Tudományos Akadémia levelező,
- 1961-ben pedig rendes tagjai közé választja

## Kalmár László (1905-1976)

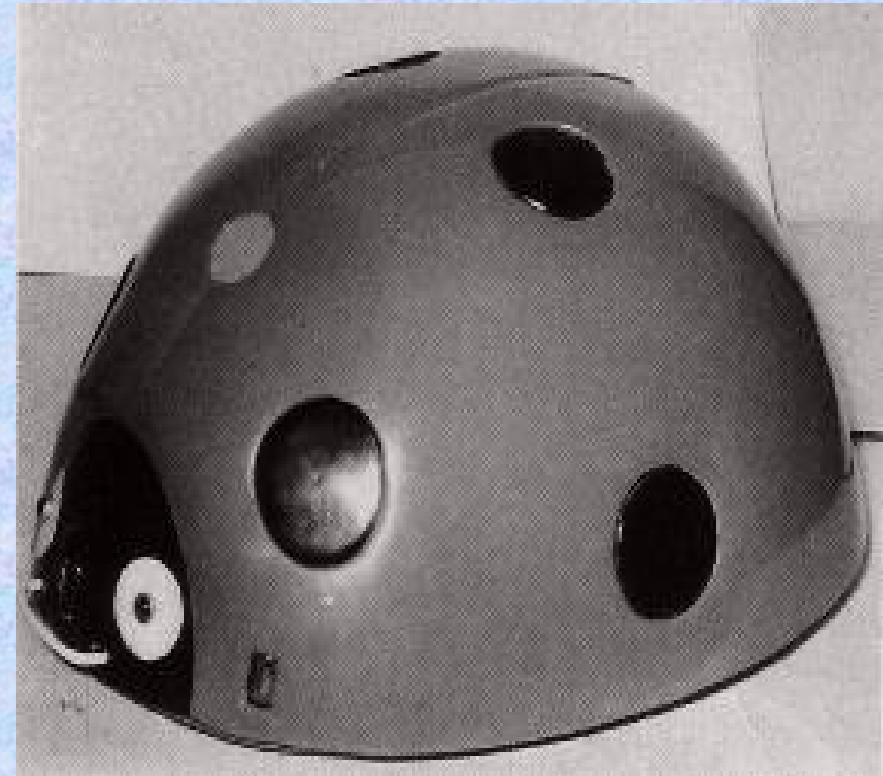


- 1963: **JATE Kiblabor megalakulása**
- 1971: **Bolyai Intézet számítástudományi tanszék vezetője**
- 1976. aug. 2.: **Mátraházán meghal**
- Fő érdeklődési területe: **matematikai logika**

## 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 alkatrészbázis: szovjet
- Szóhossz: 31 bit
- Utasításrendszer: 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

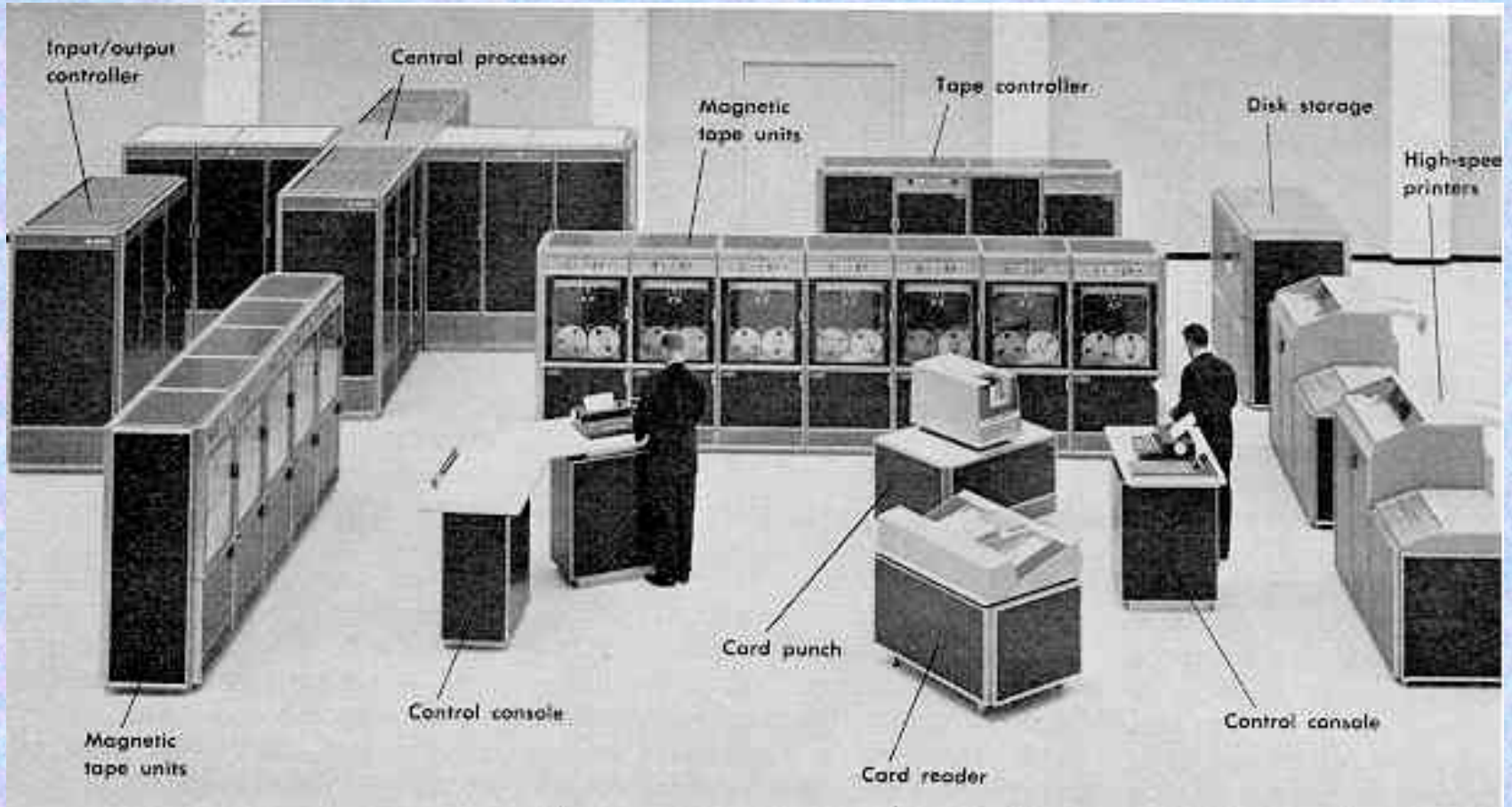
**Az 1960-70-es évek**

## A „mainframe”-ek jellemzői

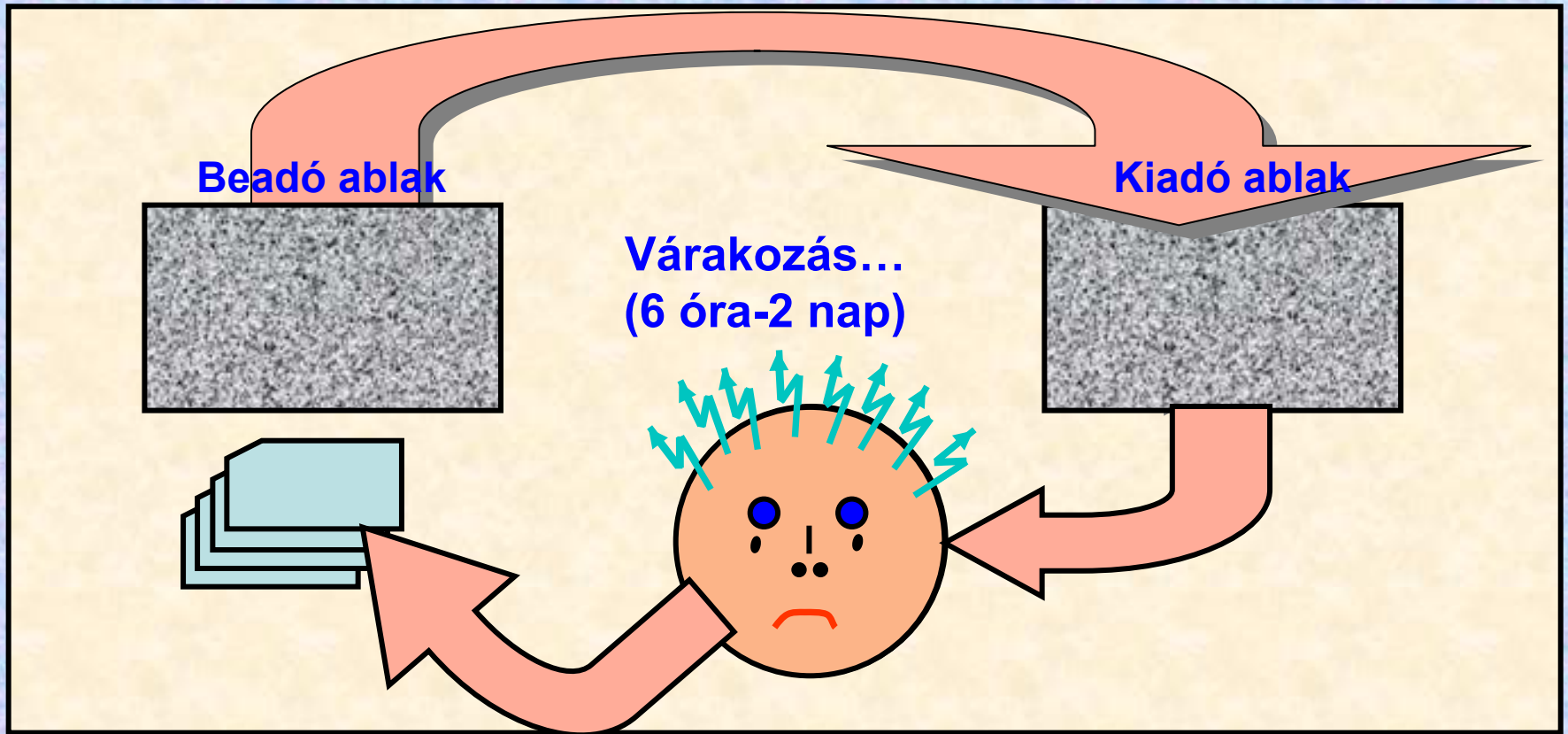
- Árkategória: néhány százezer – több millió dollár
- Telepítési feltételek:
  - hatalmas gépterem
  - klíma, álpadló, álmennyezet
  - személyzetnek munkaköpeny, -cipő
- Üzem:
  - folyamatos
  - zárt („*closed shop*”=„*kívül tágasabb*”)
  - kötegelt (*batch*) üzemmód
  - munkák fordulási ideje: néhány 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
<b>Típusok száma</b>	23	26	31	40	47	50	54
<b>A gyártó cégek száma</b>	15	15	17	21	21	22	26
<b>Gép összesen</b>	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**
  - **Különbéféle kiszolgáló részlegek**
- **1990-1992: különválnak 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

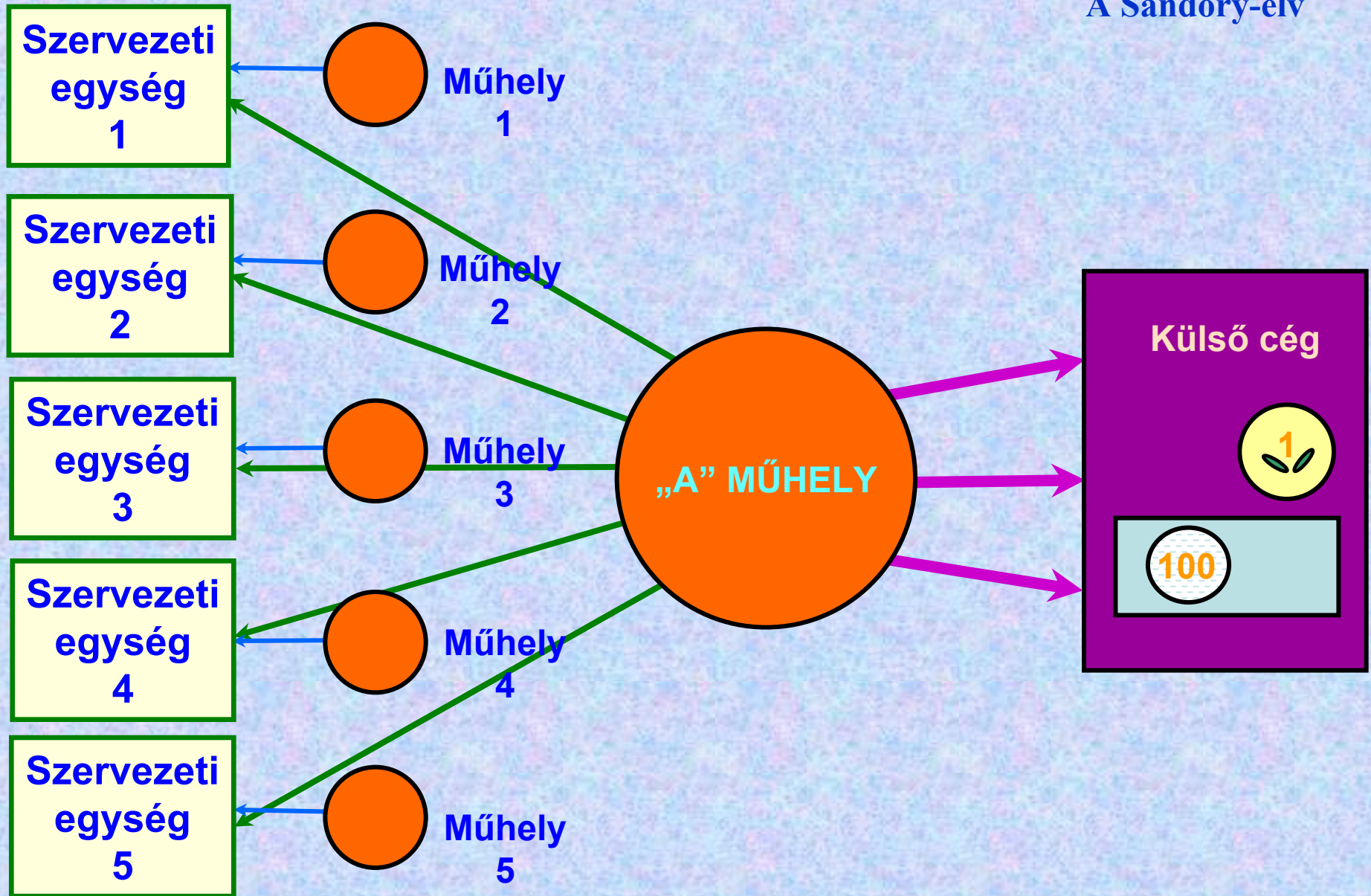
2. fázis: Összevonják őket egyetlen műhellyé

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

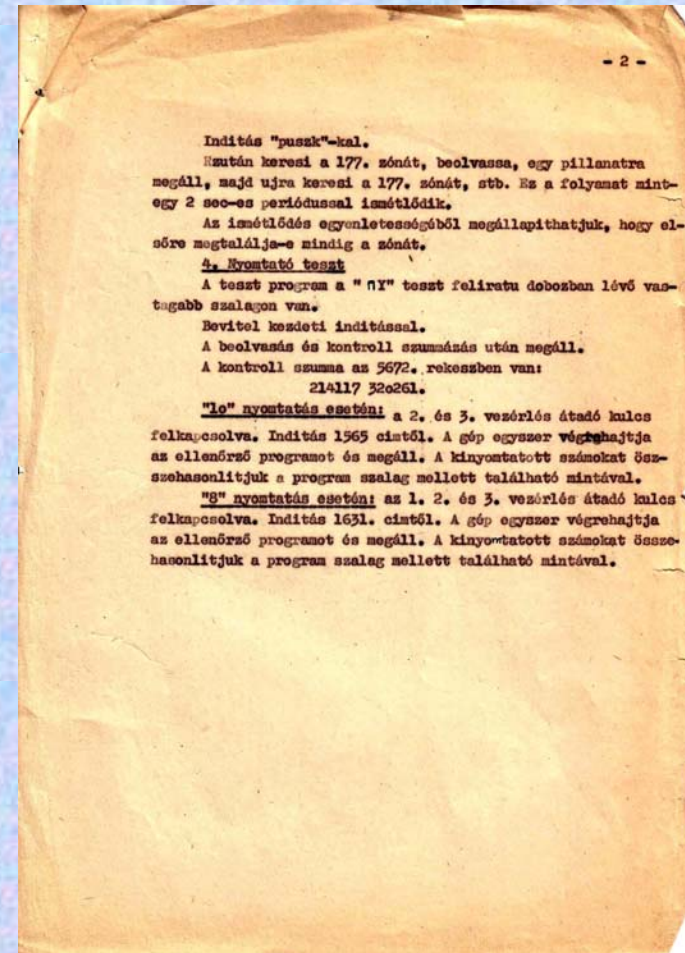
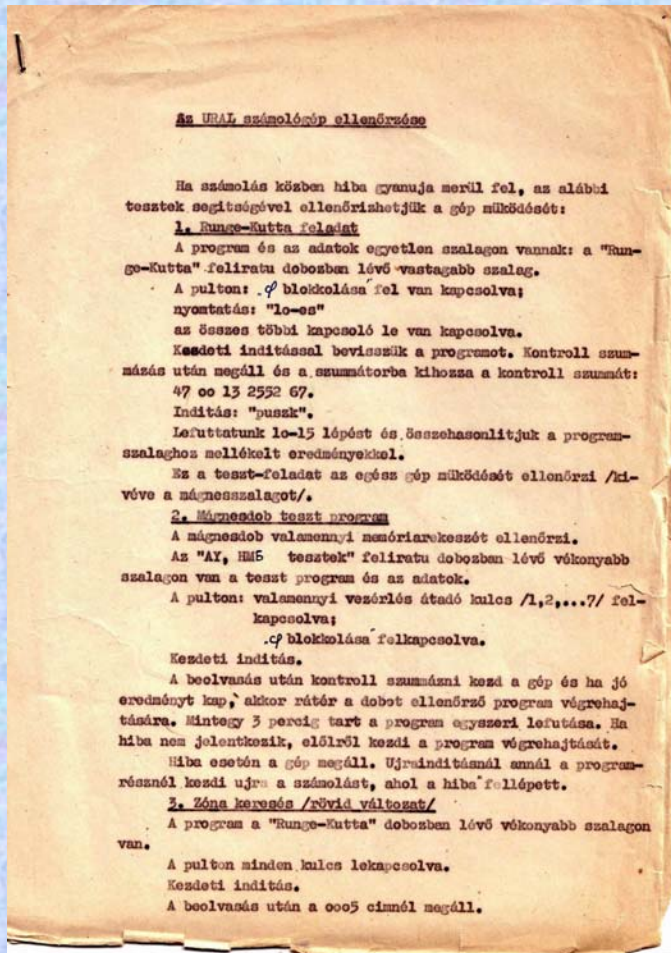
### Ciklus vége



## 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



- **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=T**árolt **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

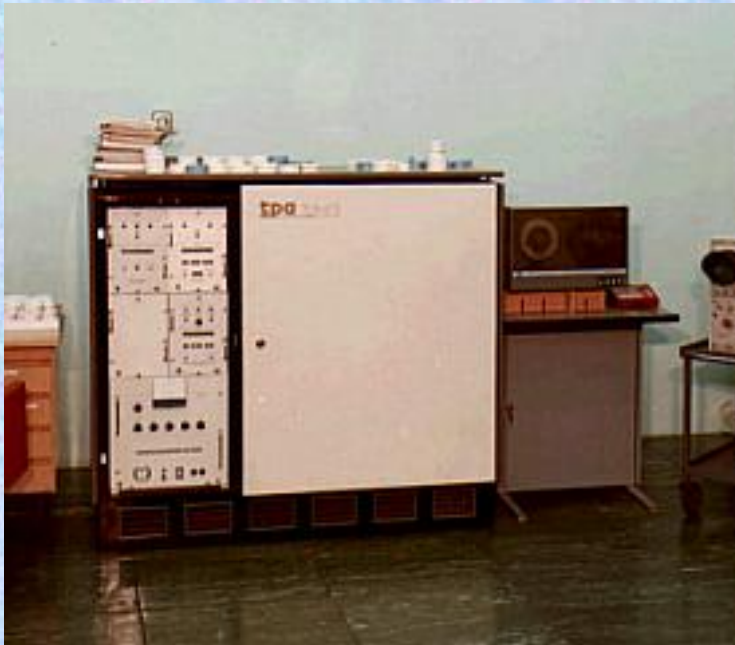
- **12 bites gépek**  
(*TPA-1001 család, PDP-8 kompatibilis*)
- **16 bites gépek**
  - TPA 70/25
  - TPA-11 család (*PDP-11 kompatibilis*)
- **32 bites gépek** (*VAX kompatibilis*)

**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**

# 15. fejezet

## A számítástechnika története a Szovjetunióban

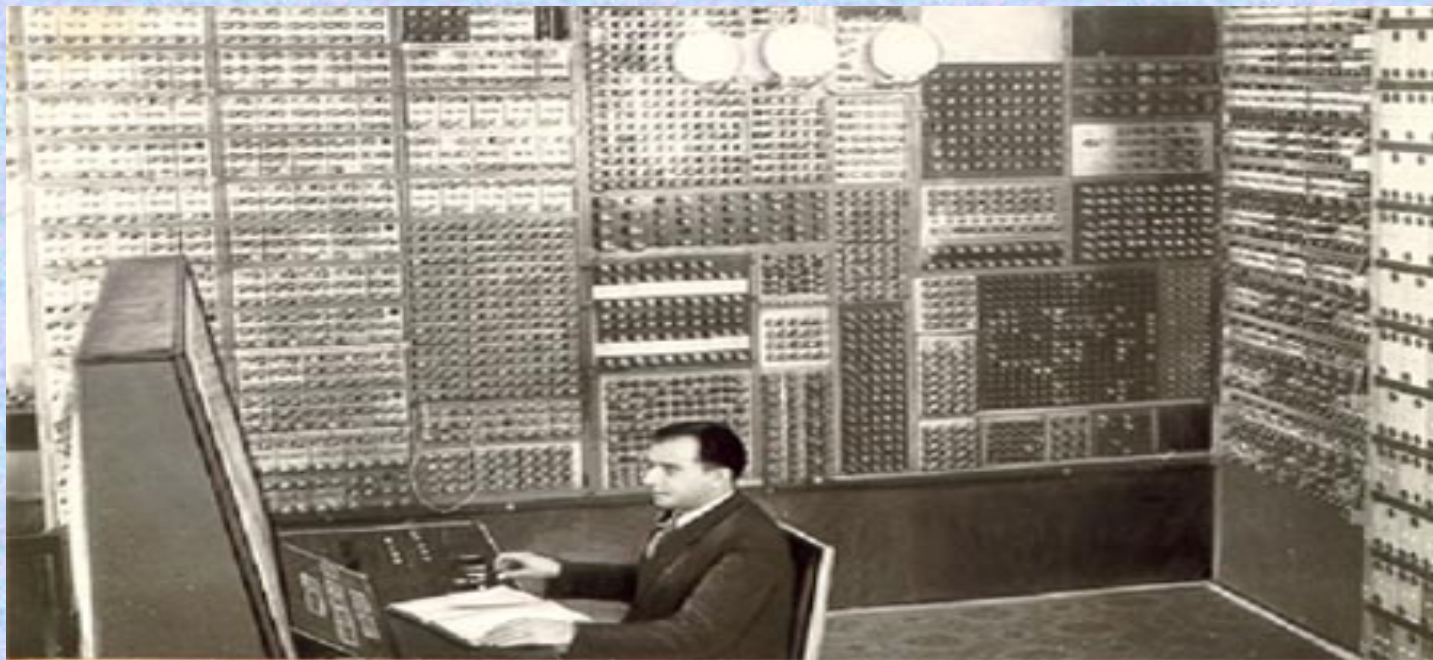
A MESZM, BESZM és URAL gépek

## 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



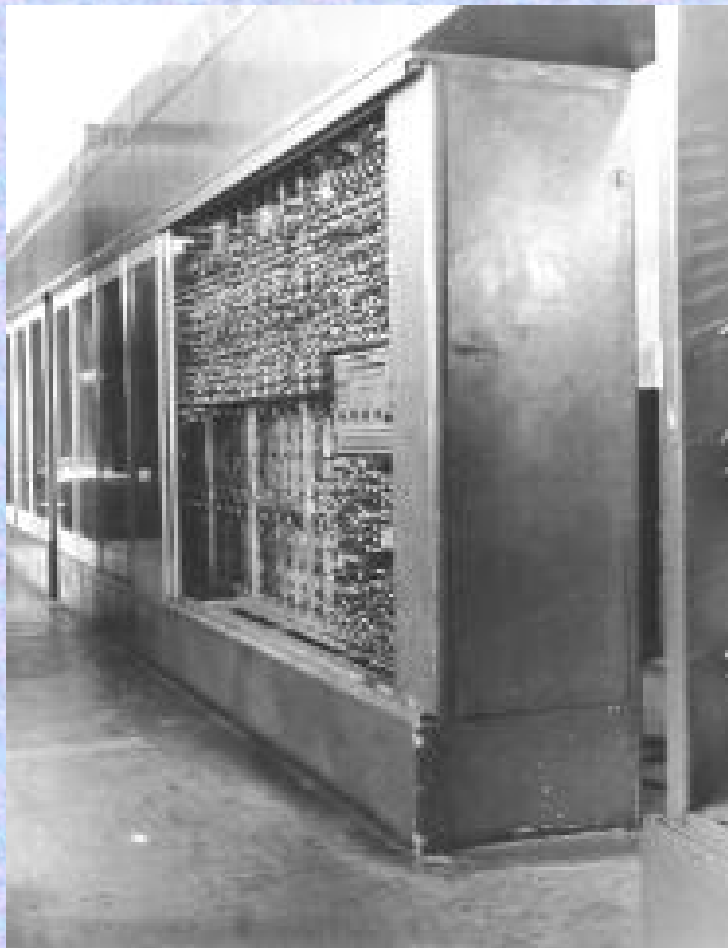
- **MESZM = Malaja Elektronnaja Szcsotnaja Masina  
(kicsiny, elektronikus számológép)**
- **Kísérleti („gyakorló”) konstrukció**
- **Cél: minél előbb működőképes modell előállítása**
- **Címzési mód: háromcímű**
- **Szóhossz: 21 bit**

# Az első szovjet számítógép: a MESZM



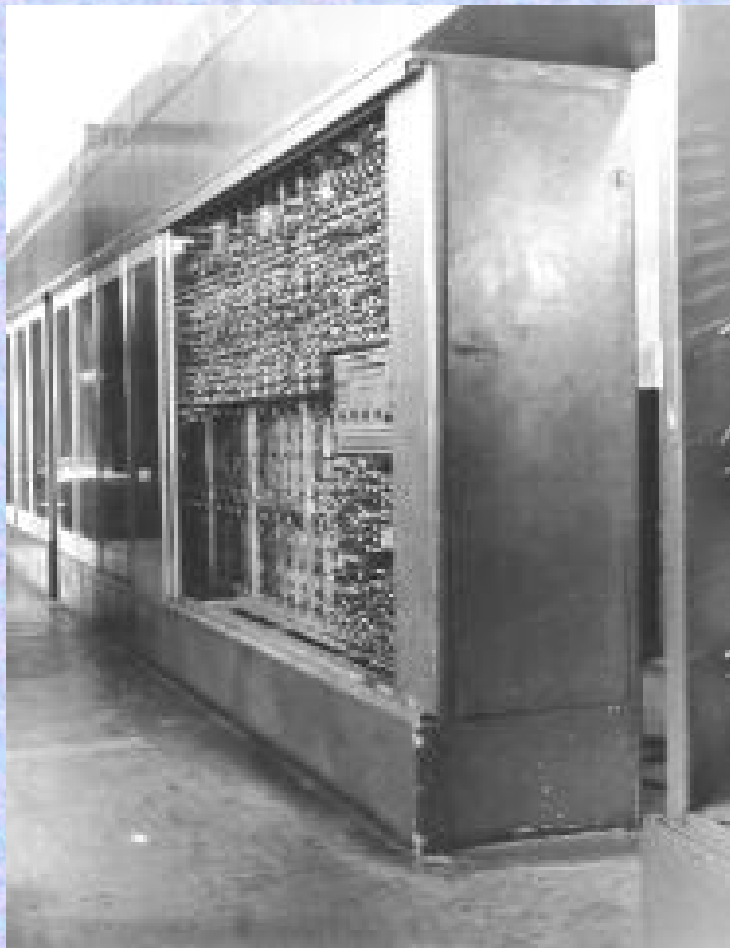
- Tárolható adatok száma: 31
- Tárolható utasítások száma: 63
- Számábrázolás: fixpontos
- Elektroncsövek száma: kb. 3800
- Input periféria: nincs (*kézi input*)
- Tároló: flip-flop-okból felépítve (*ld. ENIAC*)

## A BESZM gépek



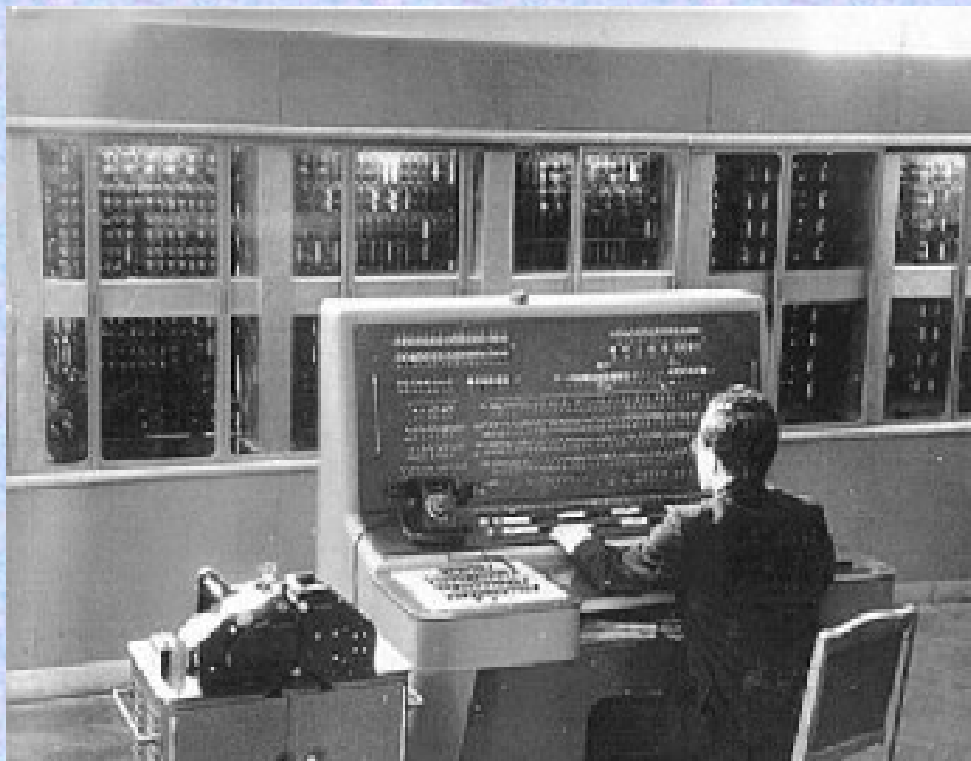
- 1948: **Megalakul az ITMiVT**
- 1950: **Elkezdődik az első „igazi” gép tervezése (későbbi nevén *BESZM-1*)**
- 1952: **Első tesztfutások**
- 1956: **Továbbfejlesztett konstrukcióval (*BESZM-2*) sorozatgyártás**

## A BESZM gépek



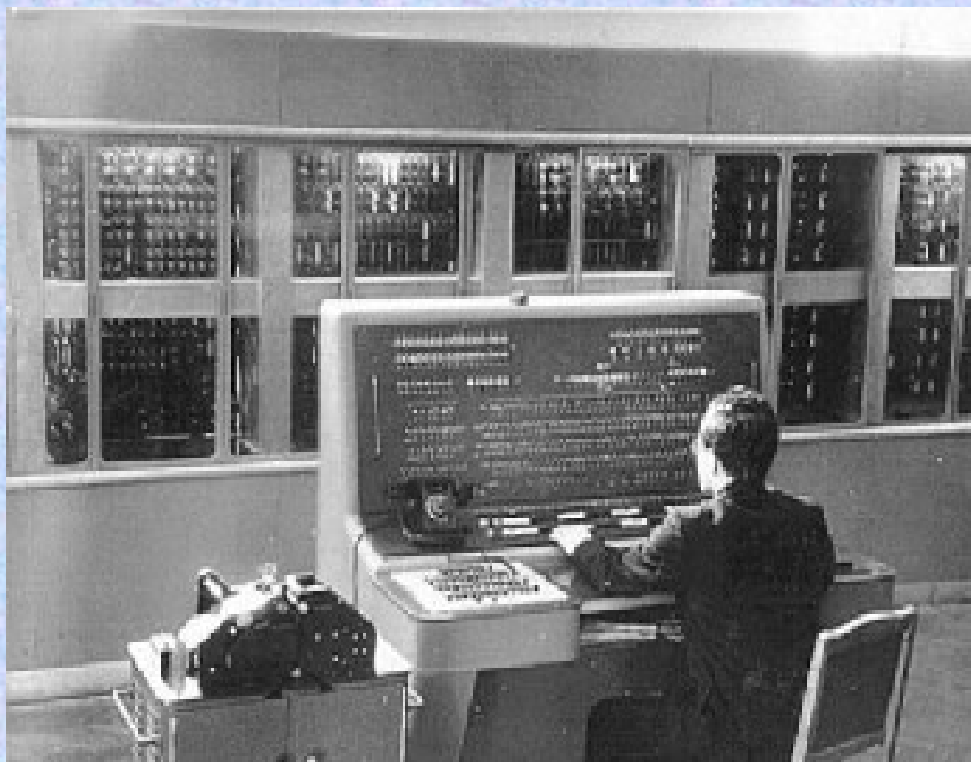
- „Éles” konstrukció
- Címzési mód: háromcímű
- Szóhossz: 39 bit
- Számábrázolás: lebegőpontos
- Tároló: késleltetett művonalas  
(melyet 1956-ban elektrosztatikusra, majd 1960-ban ferritre cseréltek)
- Tárolókapacitás: 1K szó
- Elektroncsövek száma: kb. 4000
- Input periféria: lyukszalag olvasó
- Egyéb: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

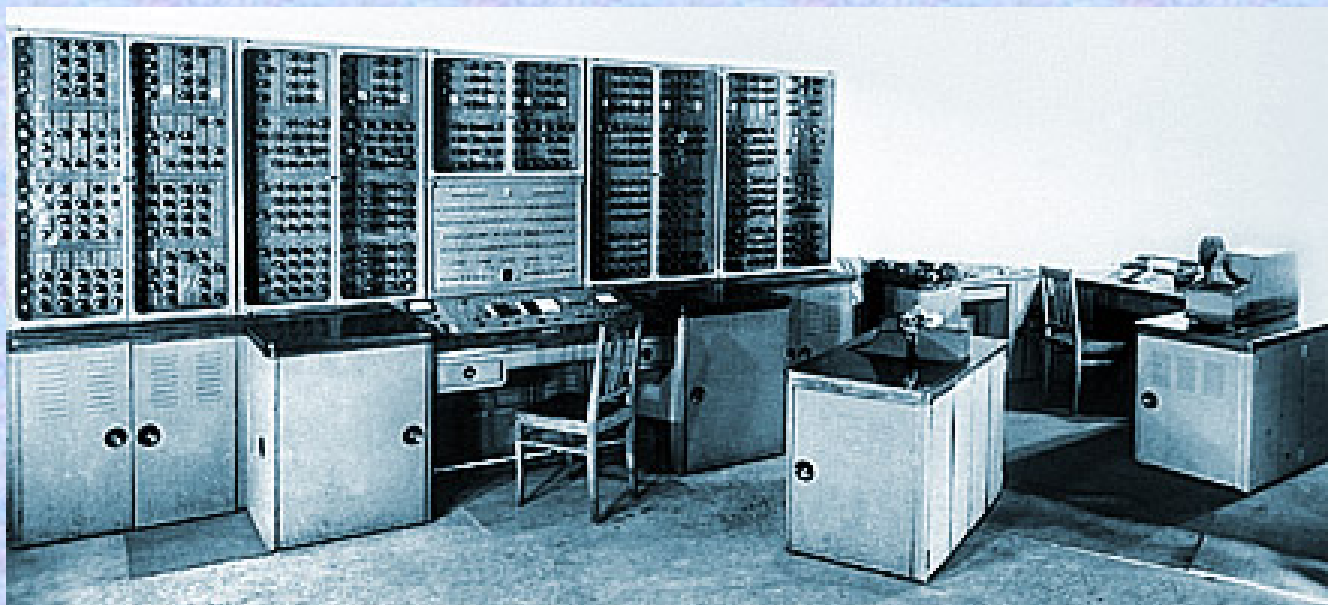
## A BESZM-6



- **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)*

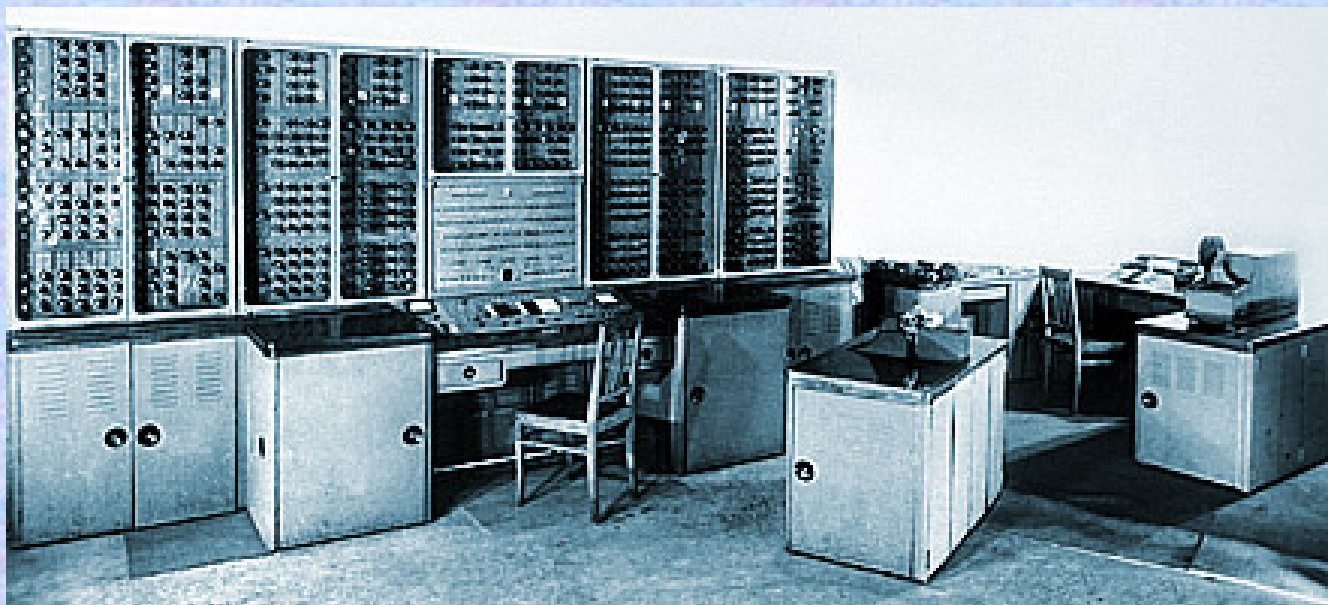


## Az URAL-1



- 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**  
(**И**нтерпретация **С**истема = értelmező rendszer)
- Lényege: **Külső tárolón (*dob*) elhelyezett szubrutinkönyvtár**, melynek rutinjai végrehajtáskor szükség szerint töltődnek a tár adott munkaterületére