



Digitális Rendszerek és Számítógép Architektúrák

1. előadás: Bevezetés, számítógép generációk

Előadó: Vörösházi Zsolt
Szolgay Péter

Feltételek:

- Könyv: L. Howard Pollard –
Computer Design and Architecture (Prentice-Hall 1990)
- 8 fő fejezet
- Követelmények: 2 évközi ZH lesz
 - 2014. ápr.2 és május 21.
 - kisZH-k illetve nagy beadandó feladatok (plusz pontokért)
- Megajánlott jegy: **mindkét ZH ≥ 4**
- Vizsgára bocsátás feltétele: **mindkét ZH ≥ 2**
- Óralátogatás kötelező
- Vizsga: szóbeli-írásbeli
- **Szigorlati/Államvizsga tárgy!**

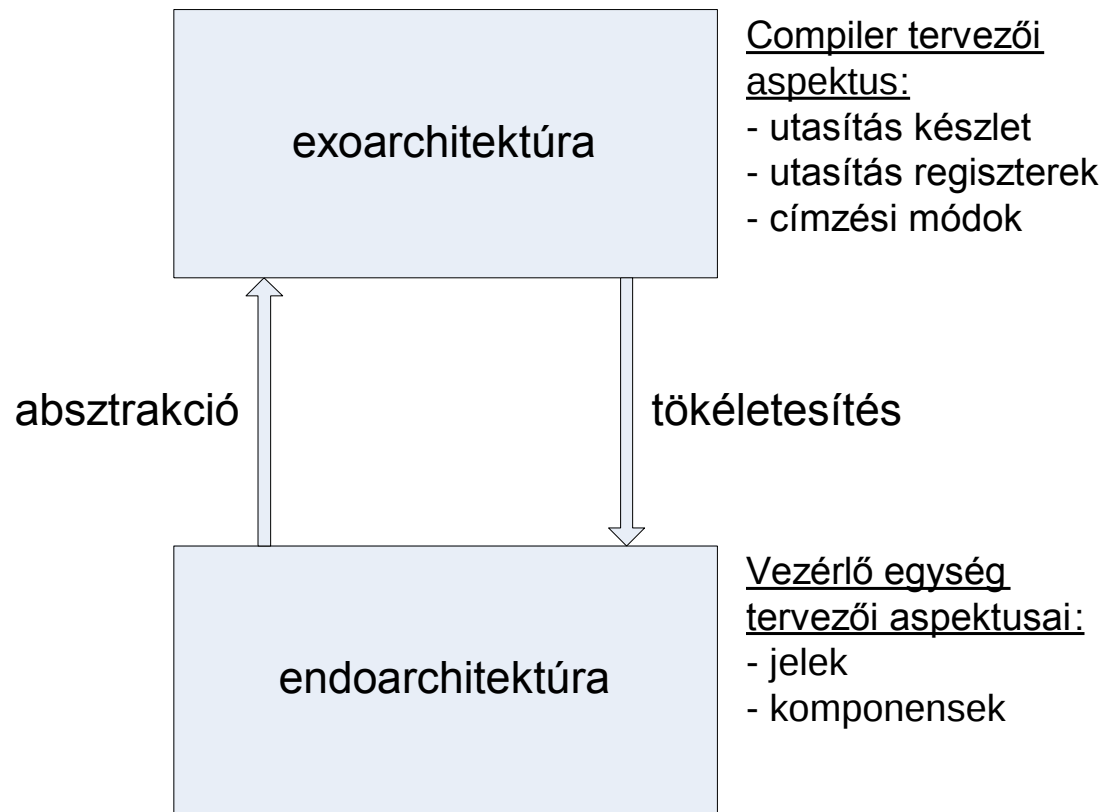
Kapcsolódó jegyzet, segédanyag:

- Angol nyelvű könyv:
<http://www.knt.vein.hu>
→ Oktatás → Tantárgyak → Digitális
Rendszerek és Számítógép Architektúrák
(nappali)
([chapter1/.../8.pdf](#))
 - Bevezetés: Számítógép Generációk ([chapter01.pdf](#))
- Fóliák, óravázlatok .ppt (.pdf)
- Feltöltésük folyamatosan
- Magyar nyelvű szigorlati segédlet

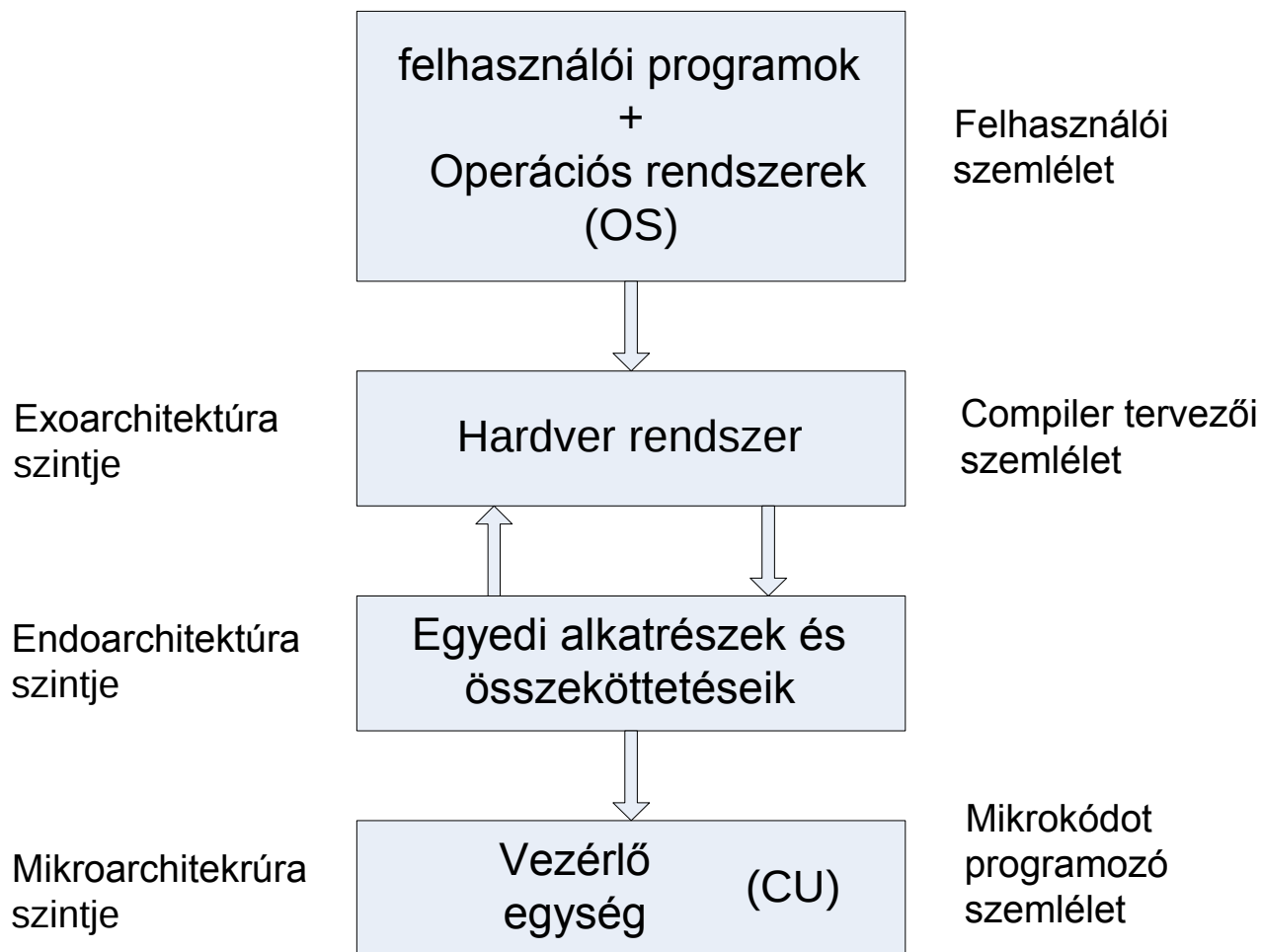
Alapfogalmak:

- **A számítógép architektúra** a hardver egy általános absztrakciója: a hardver struktúráját és viselkedését jelenti más rendszerek egyedi, sajátos tulajdonságaitól eltekintve
- **Architektúrális tulajdonságok** nemcsak a funkcionális elemeket, hanem azok belső felépítését, struktúráját is magába foglalják

Exoarchitektúra – endoarchitektúra:



Számítógép architektúra definíciója:



Számítógépes rendszerekkel szembeni tervezői követelmények:

- Aritmetika megtervezése, algoritmusok, módszerek elemzése, hogy a kívánt eredményt elfogadható időn belül biztosítani tudja
- Utasításkészlet – vezérlés
- A részegységek közötti kapcsolatok / összeköttetések a valós rendszert szemléltetik
- CFG, DFG a főbb komponensek között
- Számítógép és perifériák közötti I/O kommunikációs technikák

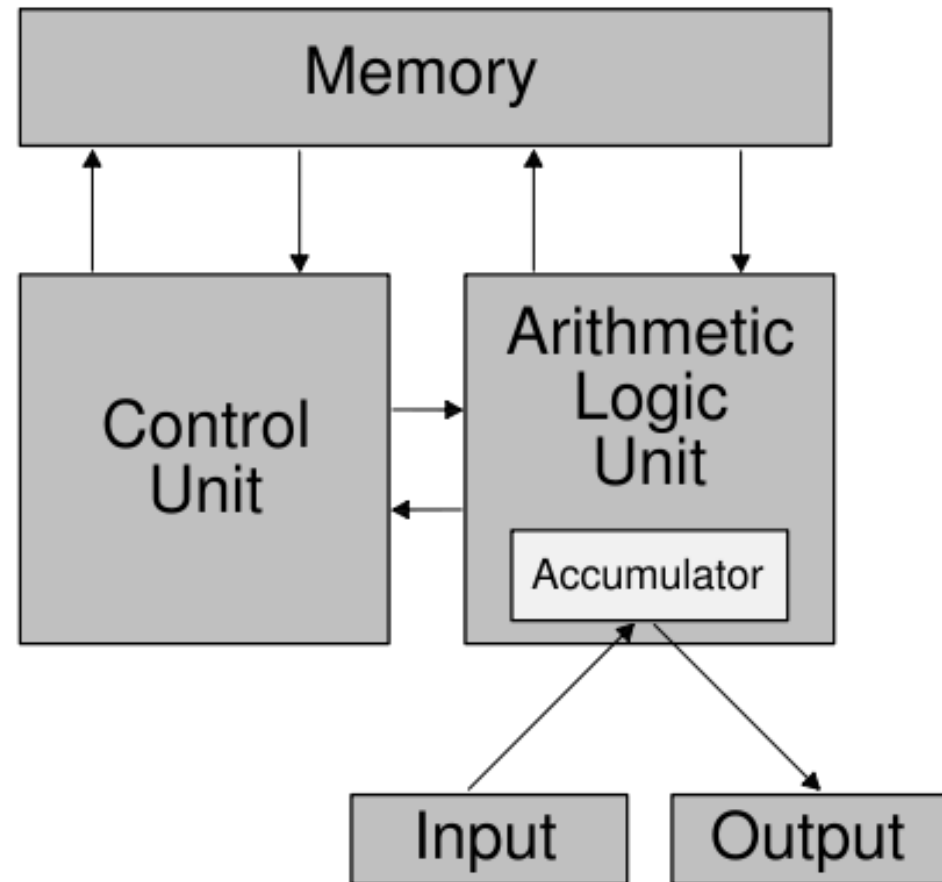


Neumann, Harvard számítógép architektúrák

A.) Neumann architektúra

■ Számítógépes rendszer modell:

- CPU, CU, ALU
- Egyetlen különálló tároló elem (utasítások és adatok részére)
- Univerzális Turing gépet implementál
- „Szekvenciális” architektúra (SISD)



Von Neumann architektúra

- „De Facto” szabvány: „single-memory architecture”. Az *adat-* és *utasítás-*címek a memória (tároló) ugyanazon címtartományára vannak leképezve (mapping). Ilyen típusú pl:
 - EDVAC (Neumann), egyetlenmegoldó tárolt-programú gép
 - Eckert, Mauchly: ENIAC, UNIVAC (University of Pennsylvania) – numerikus integrátor, kalkulátor
 - A mai rendszerek modern mini-, mikro, és mainframe számítógépei is ezt az architektúrát követik.

Neumann elvek

- számítógép működését tárolt program vezérli (Turing);
- a vezérlést vezérlés-folyam (control-flow graph - CFG) segítségével lehet leírni; /lásd vezérlő egység tétel!/ Fontos lépés itt az adatút megtervezése.
- a gép belső tárolójában a program utasításai és a végrehajtásukhoz szükséges adatok egyaránt megtalálhatók (**közös utasítás és adattárolás**, a program felülírhatja önmagát – **Neumann architektúra** definíciója);
- az aritmetikai / és logikai műveletek (programutasítások) végrehajtását önálló részegység (ALU) végzi; /lásd ALU-s tétel!/
 - az adatok és programok beolvasására és az eredmények megjelenítésére önálló egységek (IO perifériák) szolgálnak;
- 2-es (bináris) számrendszer alkalmazása.
 - Pl: EDVAC computer, ENIAC stb.

Fix vs. **tárolt** programozhatóság

- Korai számítási eszközök **fix** programmal rendelkeztek (nem tárolt programozható): pl: kalkulátor
 - - Program változtatása: „átvezetékezés”, struktúra újratervezéssel lehetséges csak
 - - Újraprogramozás: folyamat diagram → előterv spec. (papíron) → részletes mérnöki tervek → nehézkes implementáció
- **Tárolt** programozhatóság ötlete:
 - + Utasítás-készlet architektúra (ISA): RICS, CISC
 - + Változtatható *program*: utasítások sorozata
 - + Nagyfokú flexibilitás, *adatot* hasonló módon tárolni, és kezelni (assembler, compiler, automata prog. eszk.)

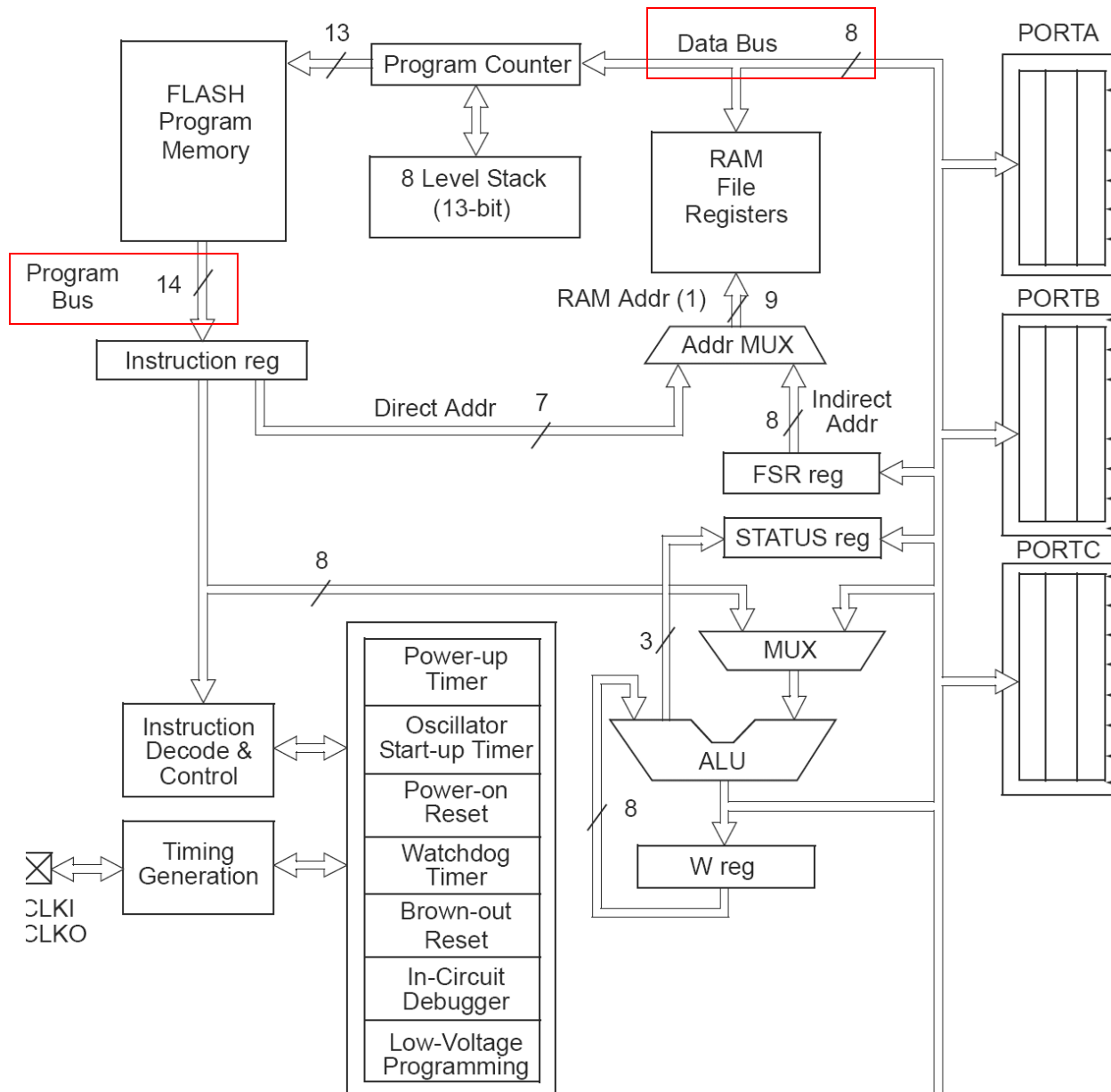
Neumann architektúra hátrányai

- „Önmagát változtató” programok (self-modifying code):
 - Már eleve hibásan megírt program „kárt” okozhat önmagában ill. más programokban is: „malware”=„malfunction”+„sw”.
 - OS szinten: rendszer leállítás
 - Pl., Buffer túlcsordulás: kezelése hozzáféréssel, memória védelemmel
- **Neumann „bottleneck”**: sávszélesség korlát a CPU és memória között (ezért kellett bevezetni a cache memóriát), amely probléma a nagymennyiségű adatok továbbítása során lépett fel.
- A nem-cache alapú Neumann rendszerekben, egyszerre vagy csak adat írás/olvasást, vagy csak az utasítás beolvasását lehet elvégezni (egy buszrendszer!)

B.) Harvard architektúra

- Olyan számítógéprendszer, amelynél a program*utasításokat* és az *adatokat* fizikailag **különálló** memóriában tárolják, és külön buszon érhetők el.
 - Eredet: Harvard MARK I. (relés alapú rdsz.)
 - További példák:
 - Intel Pentium processzor család L1- szintű különálló *adat- és utasítás-cache* memóriája.
 - Ti320 DSP jelfeldolgozó processzorok (RAM, ROM memóriái) - / bővebben a DSP-s főlíákon /
 - Beágyazott (embedded) rendszerek: MicroBlaze, PowerPC (FPGA-n) buszrendszerei, memóriái.
 - Mikrovezérlők (MCU) különálló utasítás-adat buszai és memóriái (PIC-MicroChip, Atmel stb.)

Példa: PIC 14-bites mikrovezérlő



Harvard arch. tulajdonságai

- Nem szükséges a memória (shared) osztott jellegének kialakítása:
 - + Szóhosszúság, időzítés, tervezési technológia, memória címezés kialakítása is különböző lehet.
 - Az utasítás (program) memória gyakran szélesebb mint az adat memória (mivel több utasítás memóriára lehet szükség)
 - Utasításokat a legtöbb rendszer esetében ROM-ban tárolják, míg az adatot írható/olvasható memóriában (pl. RAM-ban).
 - + A számítógép különálló buszrendszere segítségével egyidőben akár egy utasítás beolvasását és adat írását/olvasását is el lehet végezni (cache nélkül is).

„Módosított” Harvard architektúra

- Modern számítógép rendszerekben az utasítás-memória és CPU között olyan közvetlen adatút biztosított, amellyel az *utasítás-szót* is olvasható *adatként* lehet elérni.
 - *Konstans adat* (pl: string, inicializáló érték) utasítás memóriába töltésével a változók számára további helyet spórolunk meg az adatmemóriában
 - Mai modern rendszereknél a Harvard architektúra megnevezés alatt, ezt a módosított változatot értjük.
 - *Gépi* (alacsony) szintű assembly utasítások

Harvard architektúra hátrányai

- Az olyan egychipes rendszereknél (pl. SoC: System On a Chip), ahol egyetlen chipen van implementálva minden funkció, nehézkes lehet a különböző memória technológiák használata az utasítások és adatok kezelésénél. Ezekben az esetekben a Neumann architektúra alkalmazása lehet megfelelőbb.
- A magas szintű nyelveket (pl ANSI C szabvány) sem közvetlenül támogatja (nyelvi konstrukció hiánya az utasítás adatként való elérésére) – assembler szükséges

Harvard – Neumann együttes architektúra megvalósítás

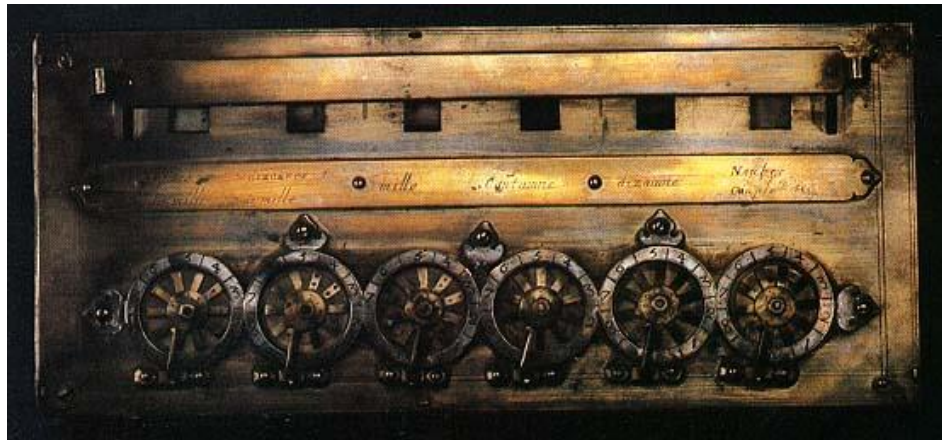
- Mai, nagy teljesítményű rendszereknél a kettőt együtt is lehet, kell alkalmazni.
- Példa: Cache rendszer
 - Programozói szemlélet (Neumann): cache 'miss' esetén a fő memóriából kell kivenni az adatot (cím -> adat)
 - Rendszer, hardver szemlélet (Harvard): a CPU on-chip cache memóriája különálló adat- és utasítás cache blokkokból áll.



Számítógép generációk

Eredet - korai számítási eszközök I:

- 1642: Pascal – mechanikus kalkulátor (+,-)
- 1671: Leibnitz – kalkulátor 4 alapműv.

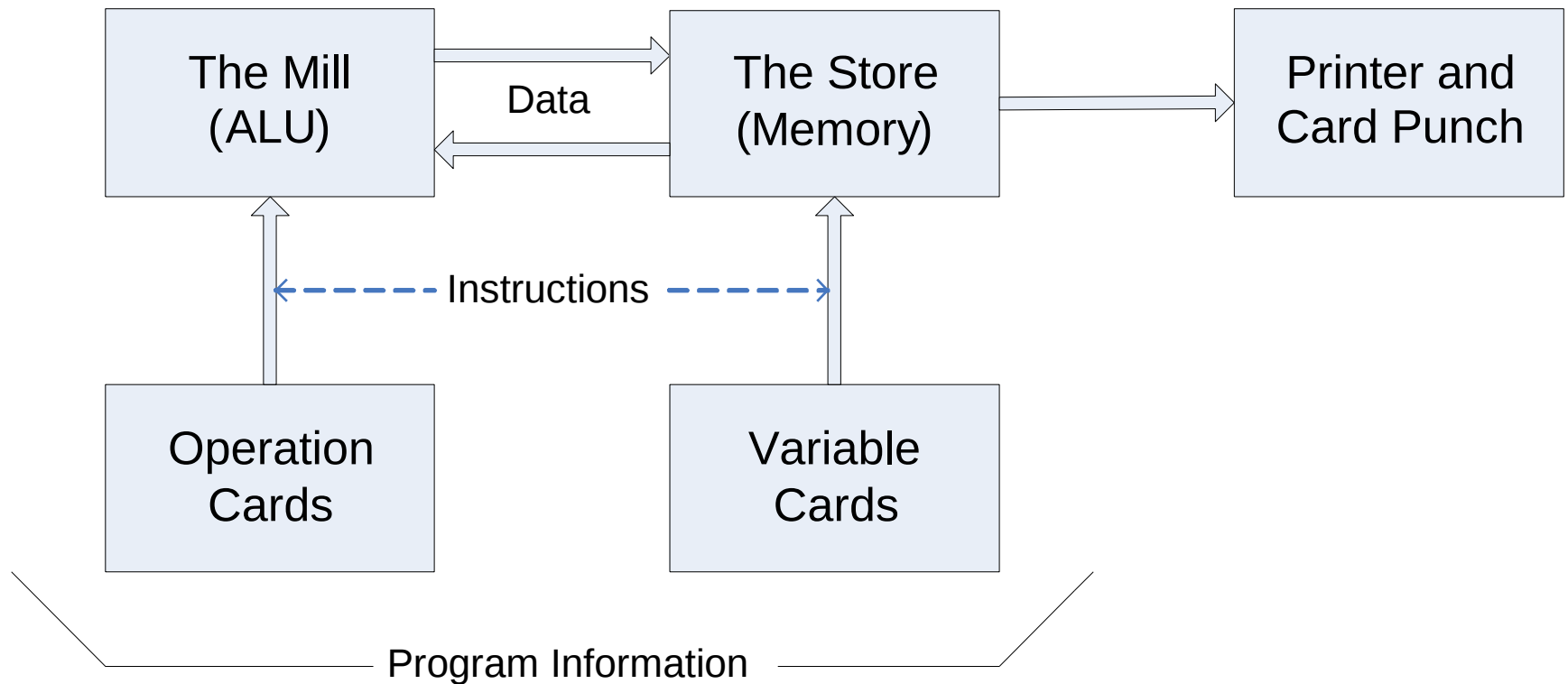


Eredet - korai számítási eszközök I (folyt.)

■ 1823: Babbage

- *Differencia Gép*: véges differencia módszer, ciklusos végrehajtás, automatikusan generált mat. táblákat
- *Analitikus Gép*: mai gépekkel szembetűnő hasonlóság, mat. fgv-ek végrehajtása. MILL – aritmetika: 4 alaplűv ('+' 1sec, '*' 1 min alatt), felt. elágazást is támogatta. Memóriája számoló „korongos”: 1000 db 50 jegyű számot tárolt.

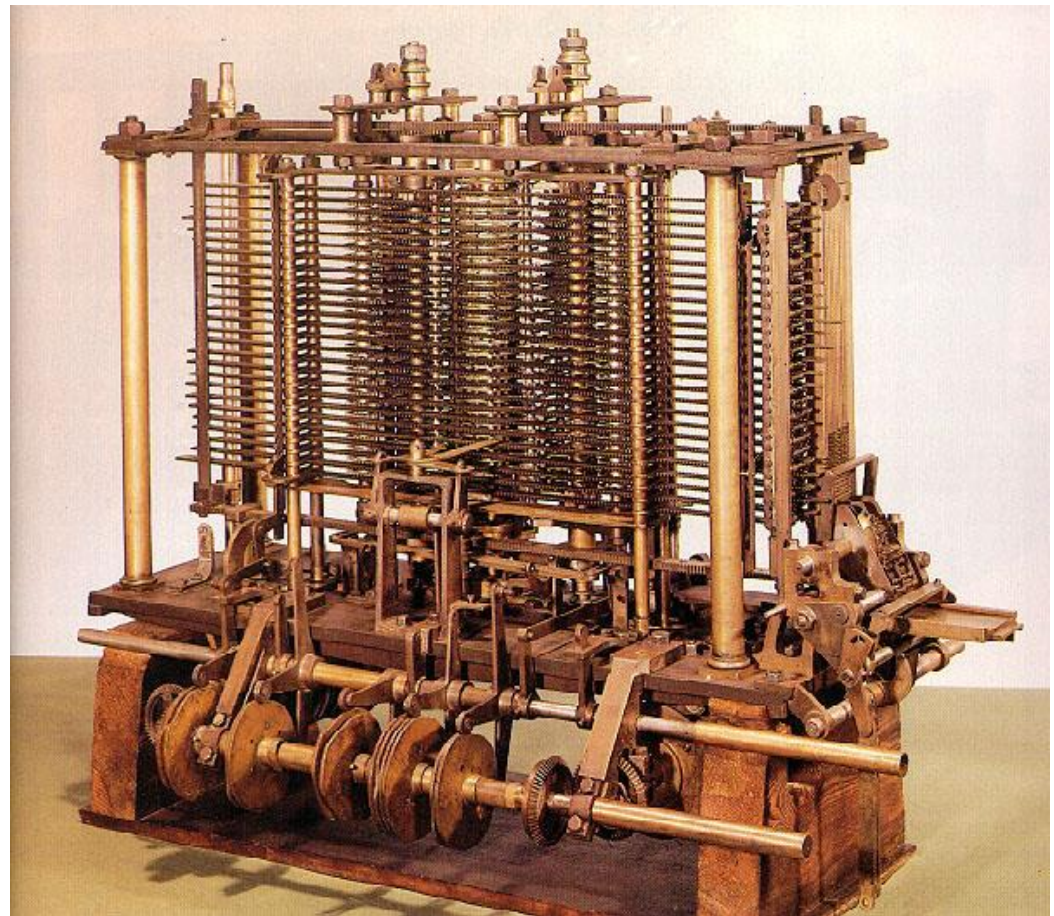
Babbage – Analitikus Gép



Babbage



Differencia gép



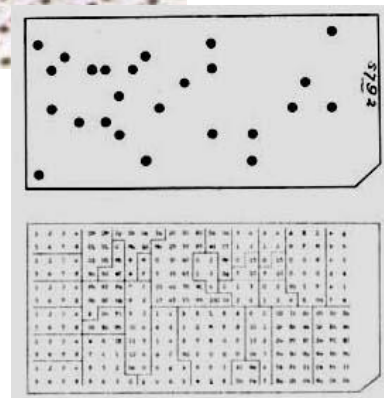
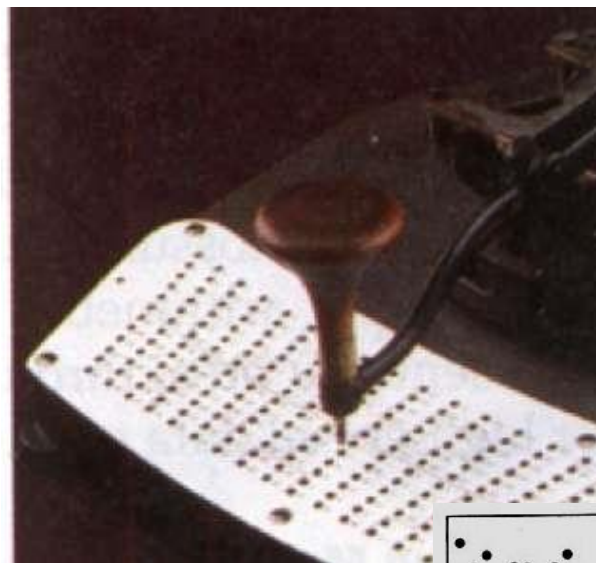
Analitikus gép

Eredet - korai számítási eszközök II (folyt.):

- 1801: Joseph Marie Jacquard: „loom” („szövőszék”) – „lyukkártya szerű” szalag, (számítási folyamat automatizálása)
- 1890: Hollerith – lyukkártya – US népszámlálás adatainak feldolgozására (1911 – IBM)
- 1930: Zuse: elektromechanikus gép
 - Z1: mechanikus relék, 2-es számrendszer!
 - Z3 (1941): első műveleti programvezérelt általános célú gép, lyukszalagos bemet (Neumann elvet követő)
- 1939: Aiken – MARK I (Harvard) relés aritmetika, számoló fogaskerekes tároló. **Harvard architektúra**: különálló program/kód és adatmemória! 72 db 23 jegyű szám

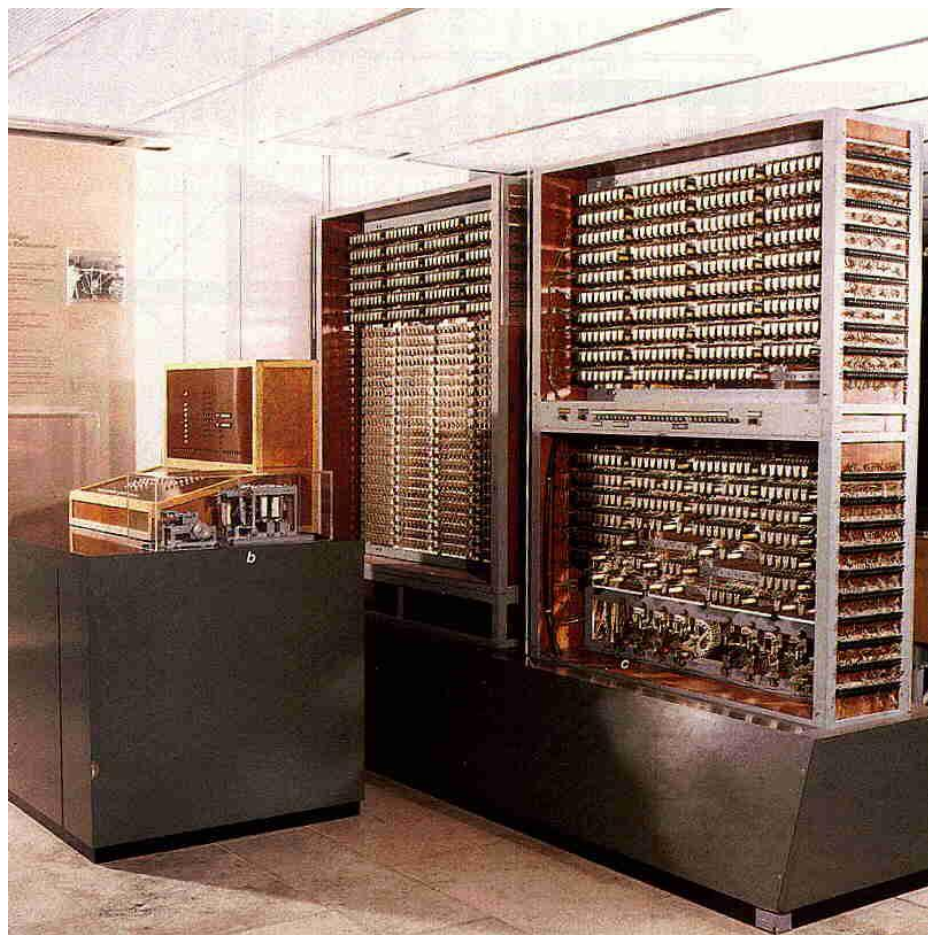
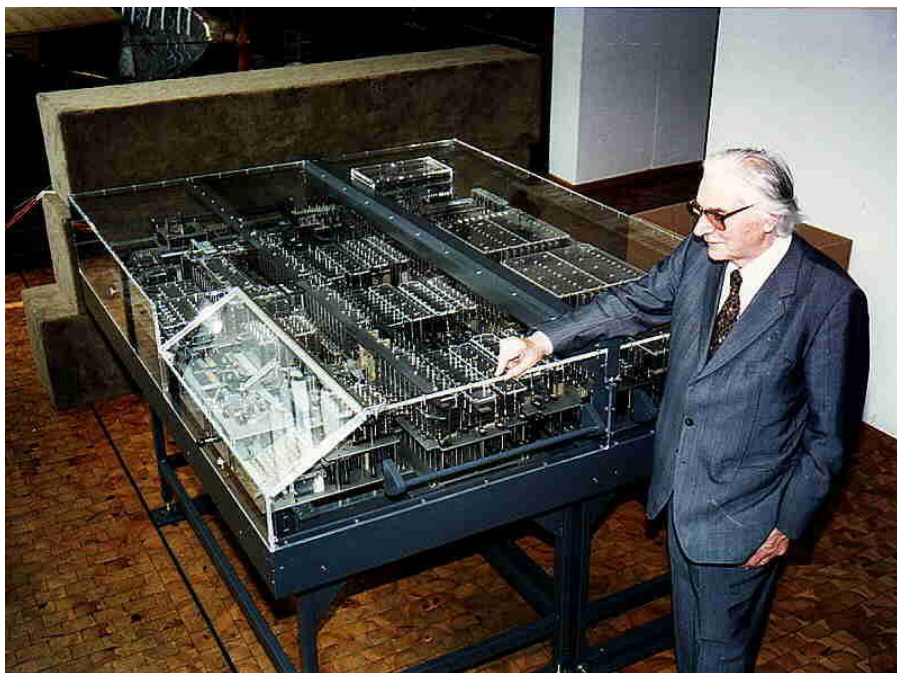


Jacquard „szövőgépe”



Hollerith - lyukkártya

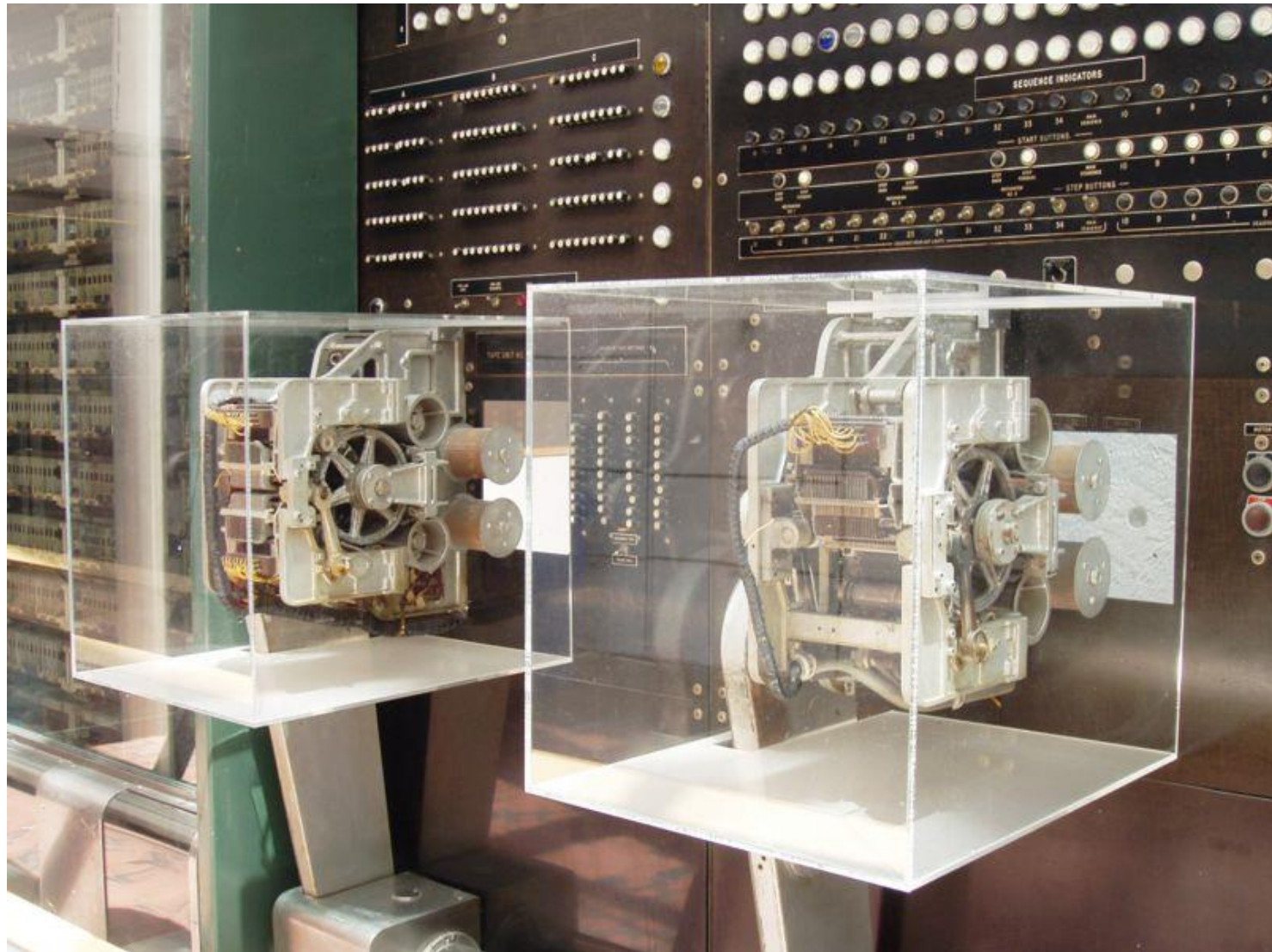
Zuse Z1 és Z3



Harvard MARK I

- Howard H. Aiken (Harvard University) – 1944
- Relés alapú aritmetika, mechanikus, korai szgép rendszer. Korlátozott adattároló képesség. (72 db 23 bites decimális számot tárol)
- ***Harvard architektúra***
 - Lyukszalagon tárolt 24-bites utasítások
 - Elektro-mechanikus fogaskerekes számlálókon tárolt 23 bites adatok
 - Utasítást adatként nem lehetett elérni!
- 4KW disszipáció, 4.5 tonna, 765.000 alkatrész: relék, kapcsolók
- Műveletvégzés: +,-: 1 sec, *: 6 sec, /: 15.3 sec
- Logaritmus, trigonometrikus fgv. számítás: 1 min

MARK I.



Eredet - korai számítási eszközök

III (folyt.):

- 1937: Berry computer (Iowa Egyetem) – John Atanasoff első elektronikus számítógép rendszer
 - egyenlet rdsz.ek Gauss eliminációjára
 - 2-es számrendszer
 - Tárolás: kondenzátoron (mint DRAM-nál)
 - ALU: aritmetikai / logikai szeparáció
 - Részek teljes elkülönítése: memória, I/O perifériák, ALU

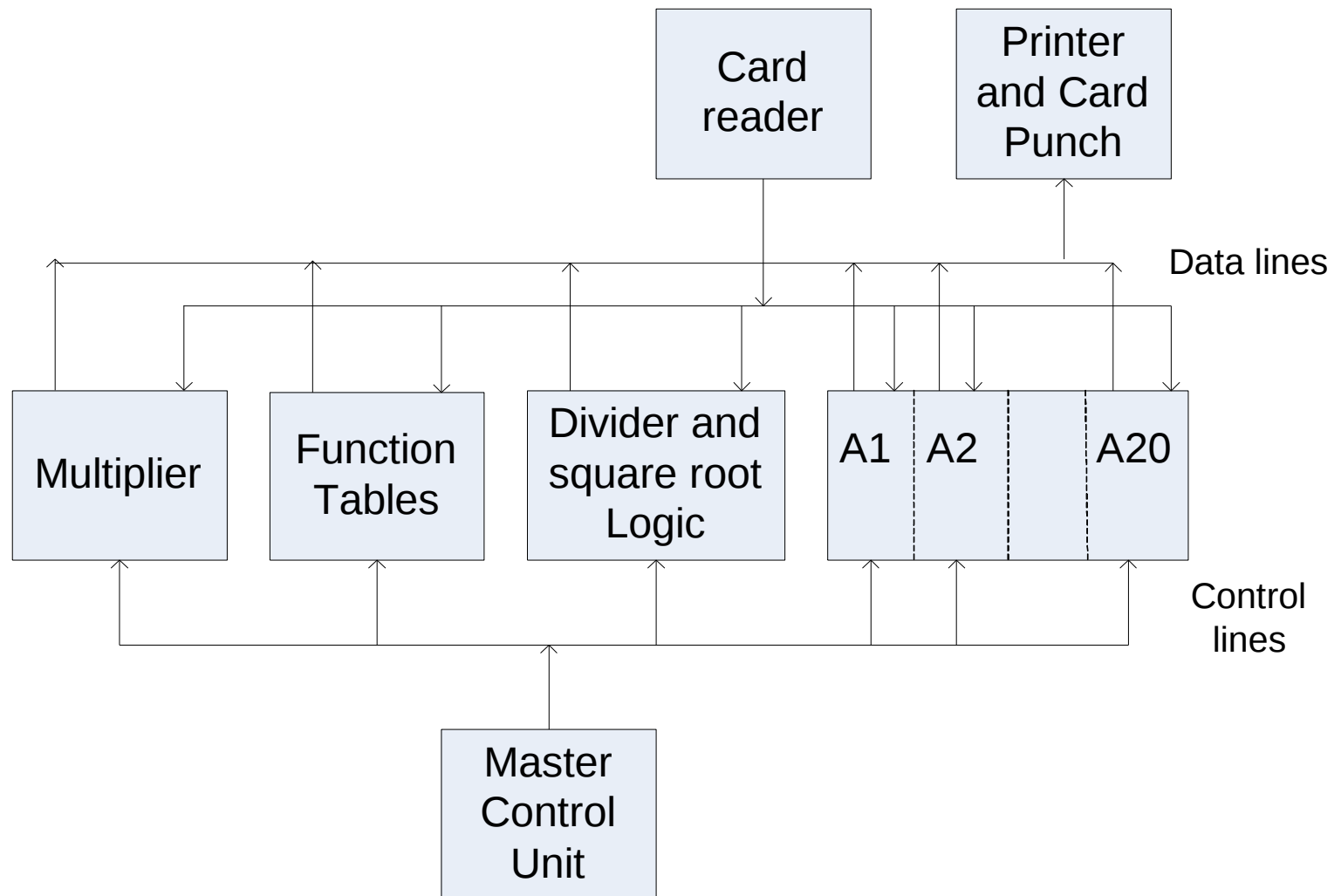
Atanasoff - Berry



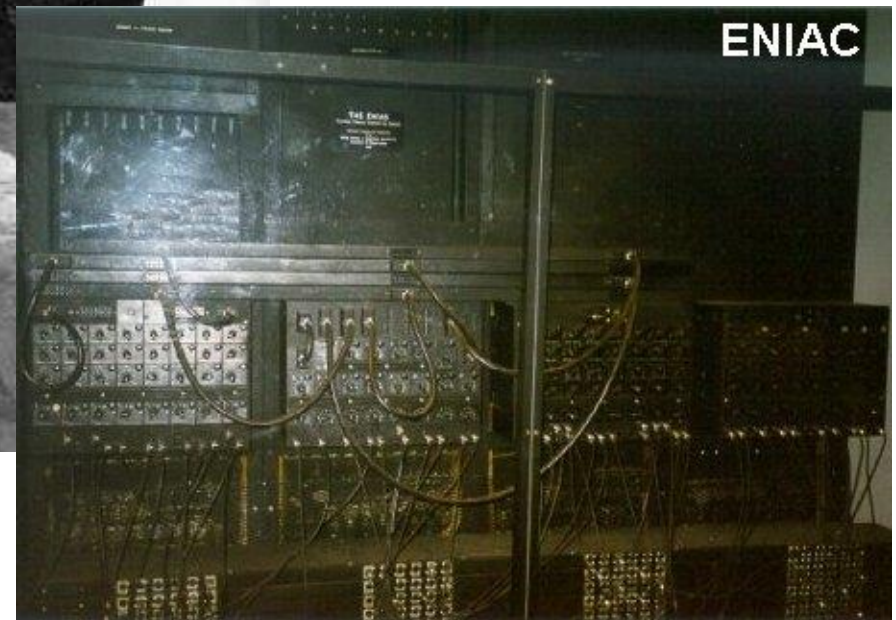
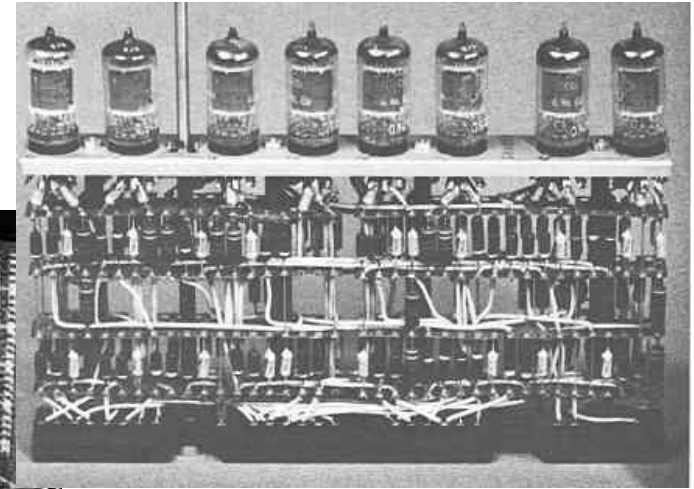
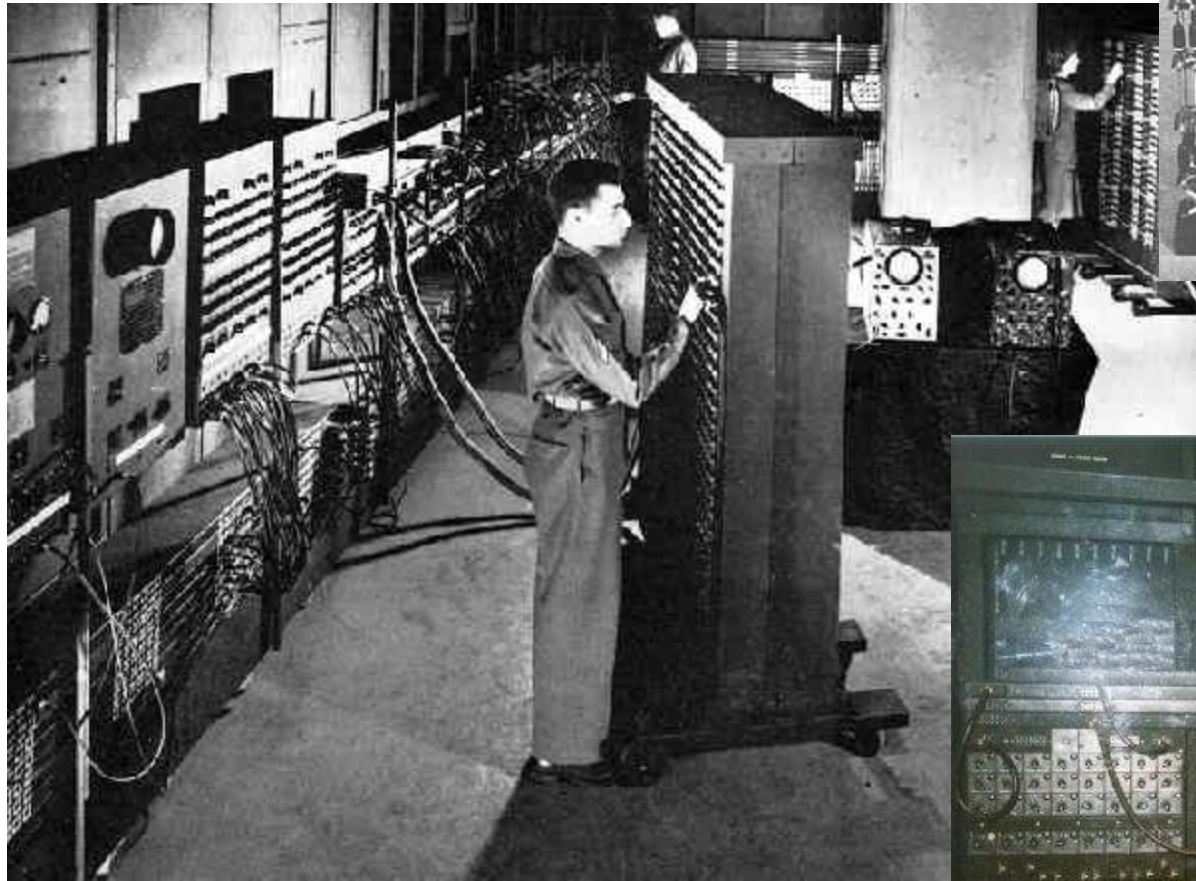
I. Generáció (1952-ig)

- 1943: **ENIAC**: elektromos numerikus integrátor és kalkulátor (Pennsylvania) Mauchly, Eckert
 - 18000 elektroncső, mechanikus, kapcsolók
 - Gépi szintű programozhatóság, tudományos célokra
 - Összeadás: 3ms
 - 20 ACC reg. – 10 jegyű decimális számra
 - 4 alapl művelet + gyökvonás
 - Kártyaolvasó-író
 - Function table: szükséges konstansok tárolása
 - Neumann elvű: közös program/kód és adat

ENIAC



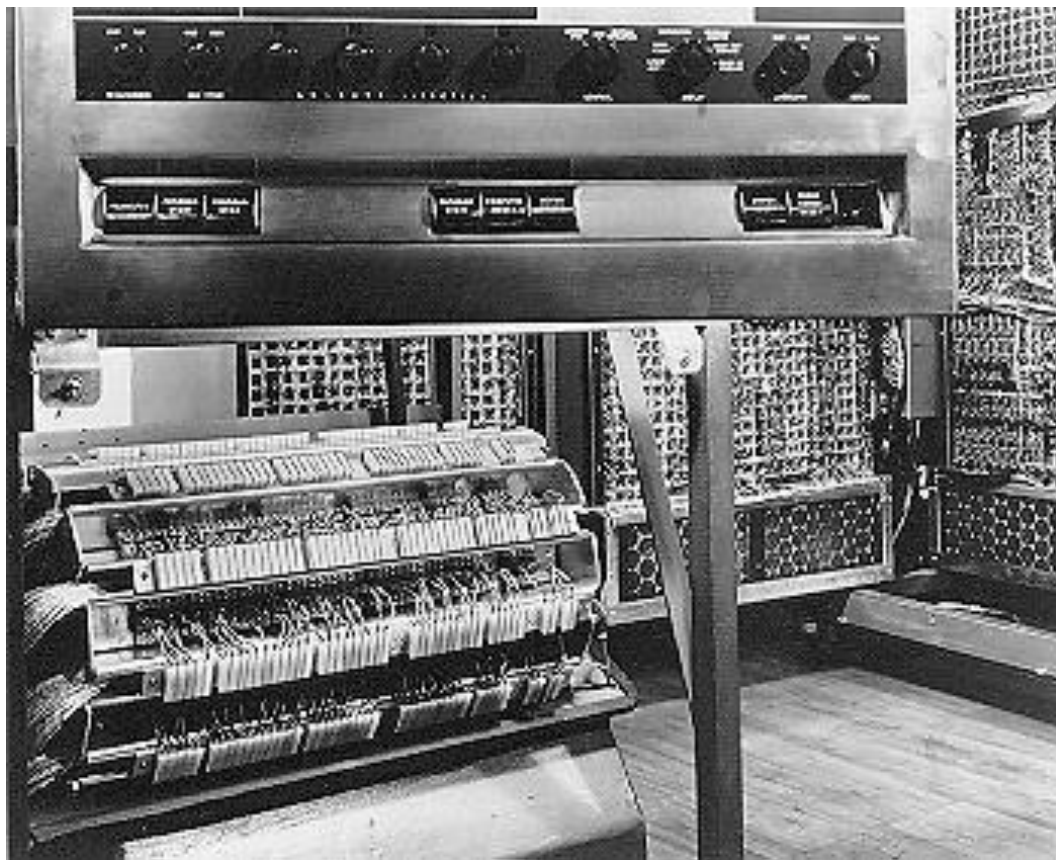
ENIAC



I. Generáció (folyt.):

- 1945: **EDVAC** (Electronic Discrete Variable Computer): egyenletmegoldó elektromos szgép.
 - Neumann János – „**von Neumann architektúra**”
 - Tárolt programozás
 - 2-es számrendszer
 - 1K elsődleges + 20K másodlagos tároló
 - soros műveletvégzés: ALU
 - utasítások: aritmetikai, i/o, feltételes elágazás
 - [EDVAC tanulmány első teljes kivonata \[pdf\]](#)

EDVAC

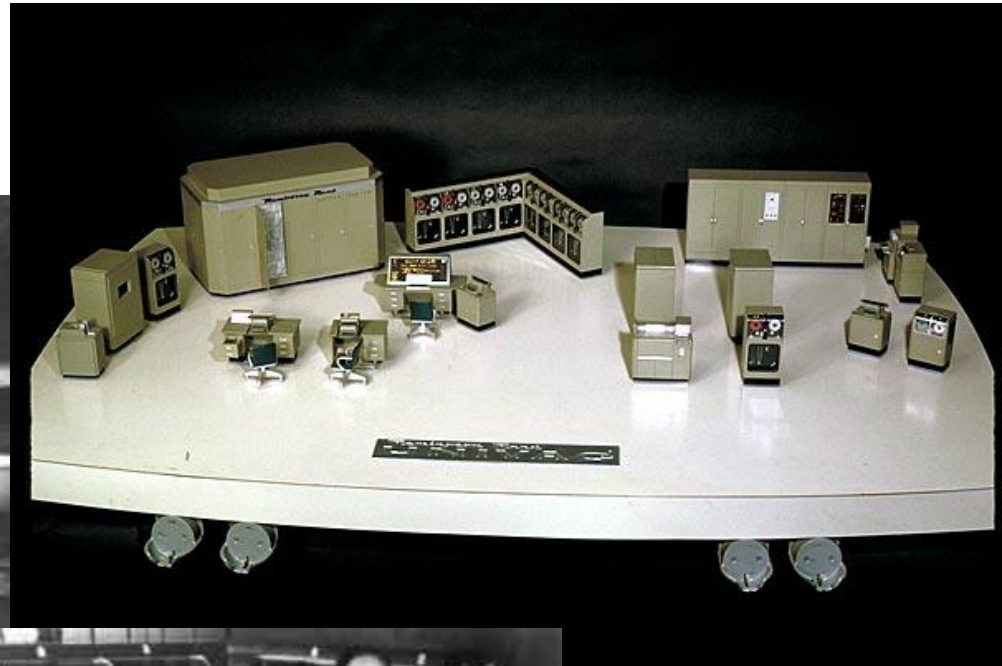
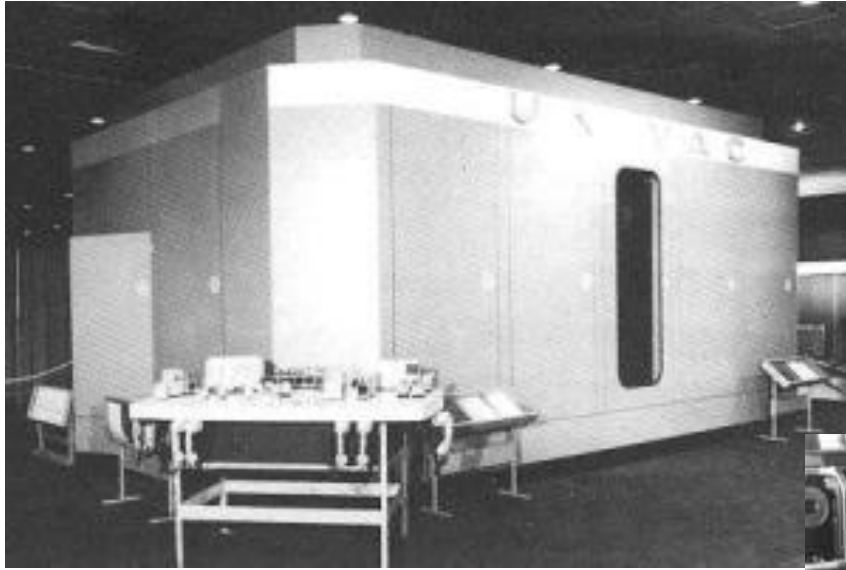


Neumann János

I. Generáció (folyt.):

- 1951: **UNIVAC I** (UNIVersal Automatic Computer I): üzleti/adminisztratív célokra
 - Mauchly, Eckert tervezte
 - 1951-es népszámlálásra, elnökválasztásra
 - 5200 elektroncső, 125KW fogyasztás, 2.25MHz
 - 1000 szavas memória, (12 bites adat: 11 digit + 1 előjelbit, 2x6 bites utasítás formátum)
 - Összeadás: 525μs, szorzás: 2150μs
 - BCD, paritás ell., hiba ell.

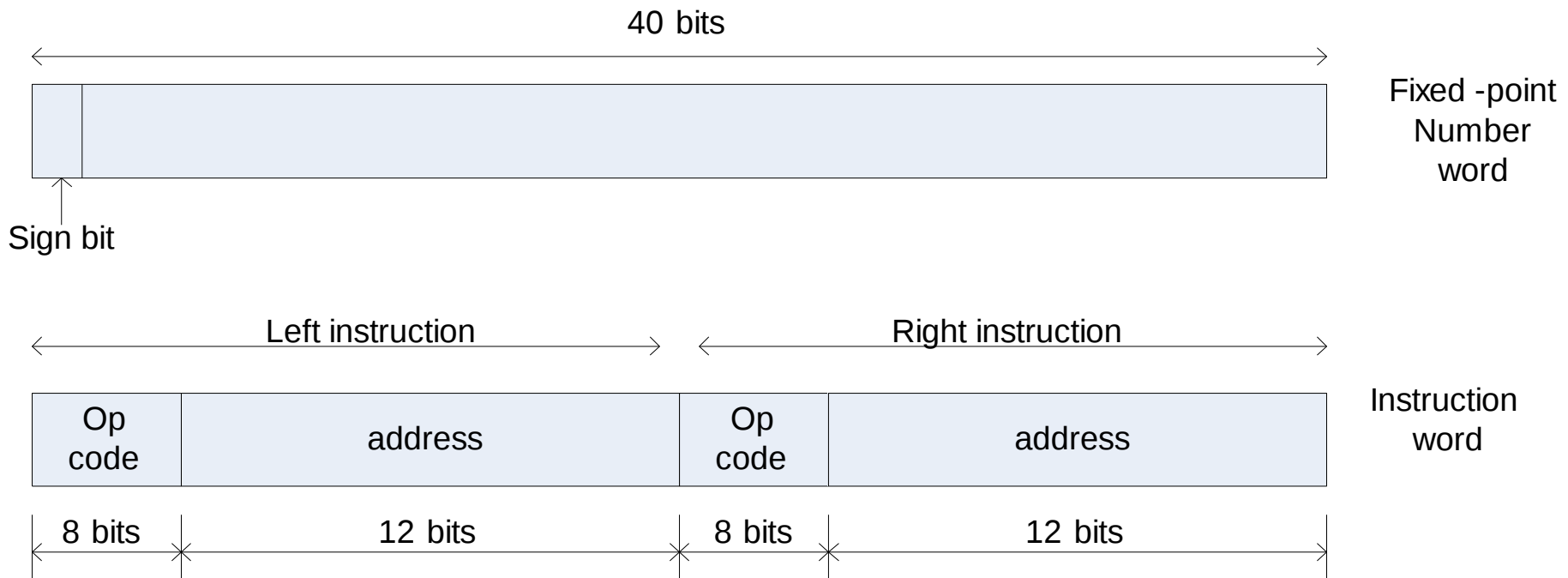
UNIVAC - I



I. Generáció (folyt.):

- 1952: **IAS** (Institute of Advanced Studies) Princeton
 - moduláris felépítés: mem, ALU, CU, I/O, ACC
 - köv. végrehajtható utasítás a memóriában a soron következő helyen van
 - egycímű gép – kisebb utasításhossz, (de ACC műveletek)
 - Mem: $2^{12}=4096$ location
 - párhuzamos feldolgozás!
 - szóhosszúság a feladattípusnak megfelelő numerikus pontosságtól függ
 - Utasítás csoportok: (1.1 táblázat)
 - Adatmozgató, aritmetikai, ugró, feltételes elágazás, címmódosító
 - IAS hátrányai: program struktúráltság – szubrutin hívás (call / return) nem támogatott, nincsenek nemnumerikus adatok

IAS adat és utasításformátum:



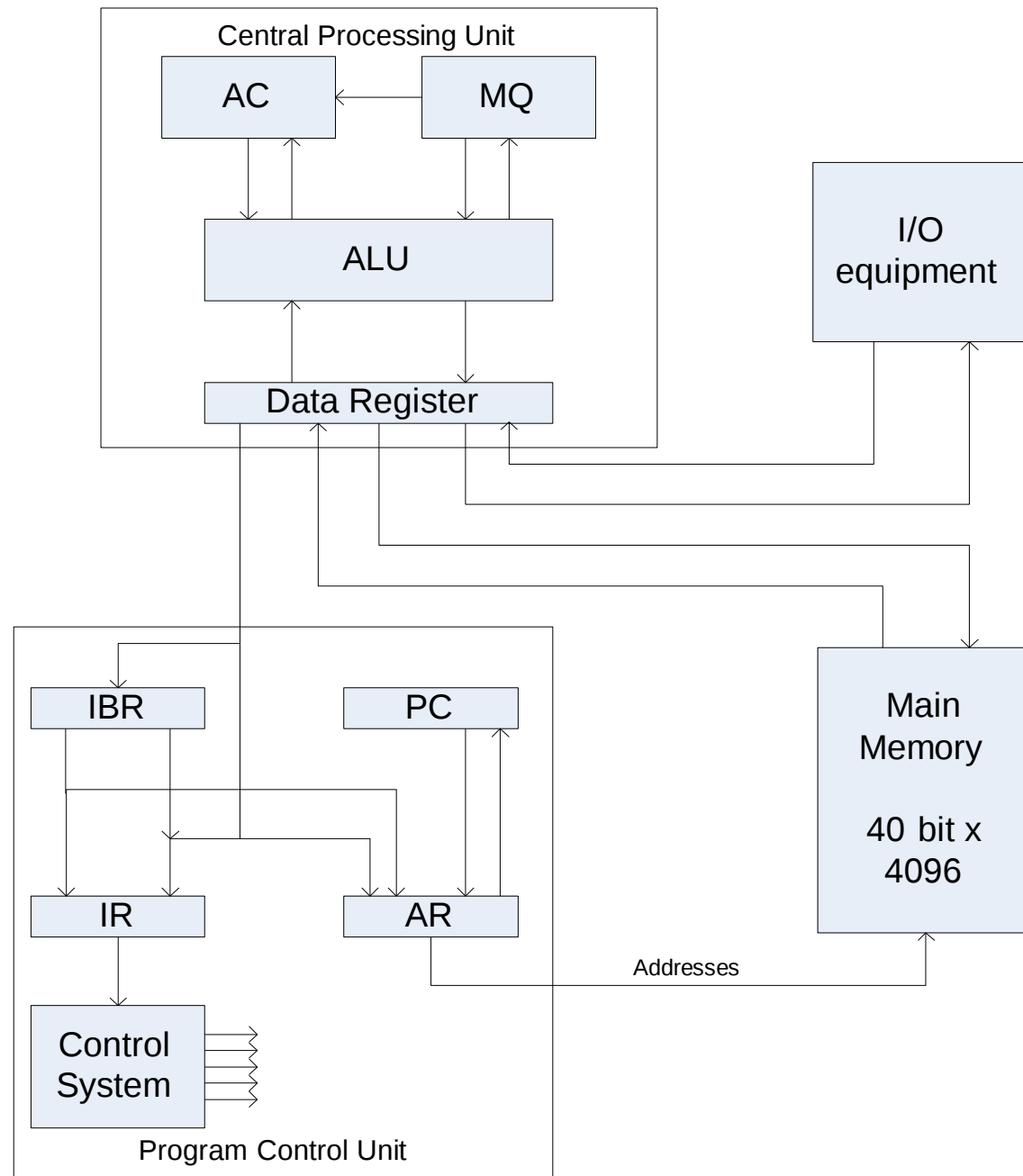
1.1 Táblázat: IAS utasítások

<i>Data transfer instructions</i>	
<i>Instruction</i>	<i>Description</i>
LDA X	Load ACCUMULATOR with value stored at location X.
LDAM X	Load ACCUMULATOR with negative of value stored at location X.
ABS X	Load ACCUMULATOR with absolute value of number stored at location X.
ABSM X	Load ACCUMULATOR with negative of absolute value of number stored at location X.
LDM X	Load MQ register with value stored at location X.
MQA	Load ACCUMULATOR with value stored in MQ register.
STOR X	The value of the ACCUMULATOR is transferred to location X.
<i>Arithmetic instructions</i>	
<i>Instruction</i>	<i>Description</i>
ADD X	Add number stored at location X to ACCUMULATOR.
SUB X	Subtract number stored at location X from ACCUMULATOR.
ADDABS X	Add absolute value of number stored at location X to ACCUMULATOR.
SUBABS X	Subtract absolute value of number stored at location X from ACCUMULATOR.
MULT X	Multiply the number stored in MQ register by value stored in location X, leave 39 most significant bits in ACCUMULATOR, and leave 39 least significant bits in MQ register.
DIV X	Divide value in ACCUMULATOR by value stored at location X; leave remainder in ACCUMULATOR and quotient in MQ register.
LFTSHFT	Multiply the number in the ACCUMULATOR by 2, leaving it there.
RGTSHT	Divide the number in the ACCUMULATOR by 2, leaving it there.

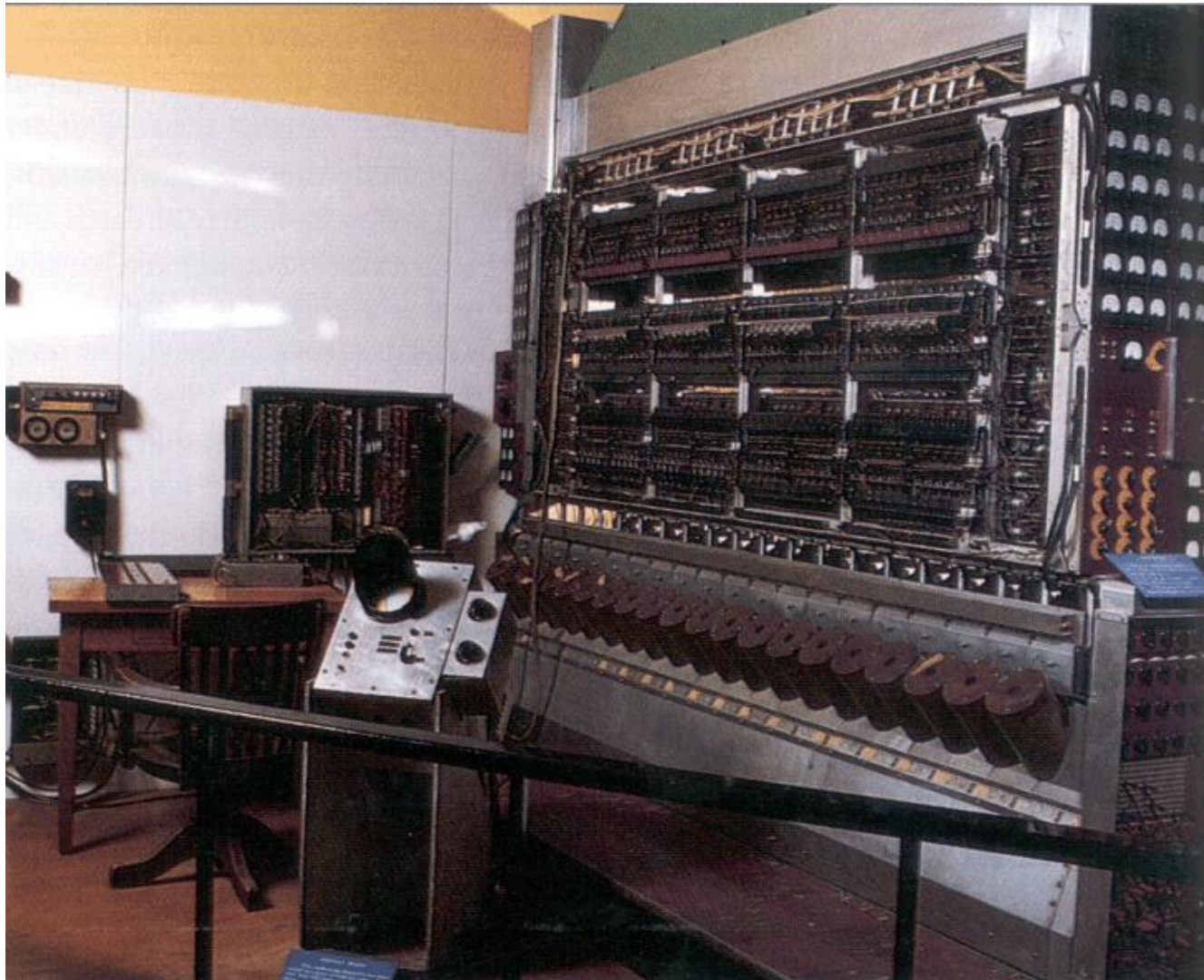
1.1 Táblázat: IAS utasítások (folyt.)

<i>Jump instructions</i>	
<i>Instruction</i>	<i>Description</i>
JMPL X	Next instruction to execute is in most significant half of location X.
JMPR X	Next instruction to execute is in least significant half of location X.
<i>Conditional branch instructions</i>	
<i>Instruction</i>	<i>Description</i>
BRANCHL X	If number in ACCUMULATOR is nonnegative, next instruction to execute is in most significant half of location X.
BRANCHR X	If number in ACCUMULATOR is nonnegative, next instruction to execute is in least significant half of location X.
<i>Address modification instructions</i>	
<i>Instruction</i>	<i>Description</i>
CADRL X	The address bits (12 least significant bits) of the most significant half of location X are replaced with the 12 least significant bits of the ACCUMULATOR.
CADRR X	The address bits (12 least significant bits) of the least significant half of location X are replaced with the 12 least significant bits of the ACCUMULATOR.

IAS



IAS computer



II. Generáció (1952-63):

- Üzleti célokra (háború vége) IBM
- Tranzisztor! (1940 végétől)
- Csökkenő méret + disszipált telj. / sebesség nő
- Core memóriák – megbízható, gyors
- Lebegő pontos számok, utasítások
- Új módszer az operandus helyének azonosítására
- FORTRAN, ALGOL, COBOL nyelvek
- I/O processzorok: CPU tehermentesítése
- Batch programozás, könyvtári függvények, compilerek

II. Generáció (folyt.):

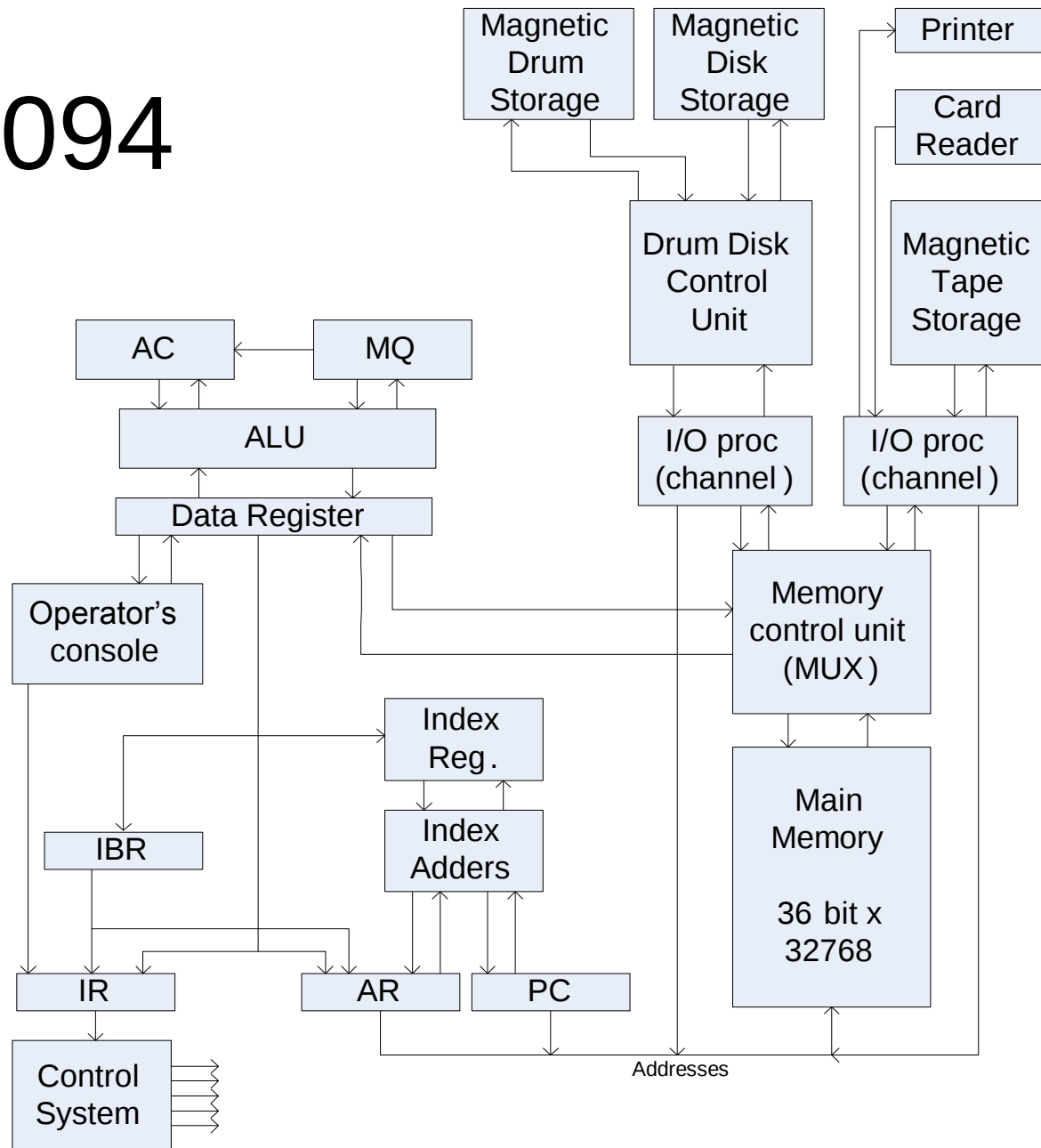
■ IBM 709x

- 36 bites utasítás, műveleti kód (1.1 tábl.)
- egycímű gép ($AR \leftarrow PC + IR$ tartalma)
- 72 bites adatút
- I/O processzorok

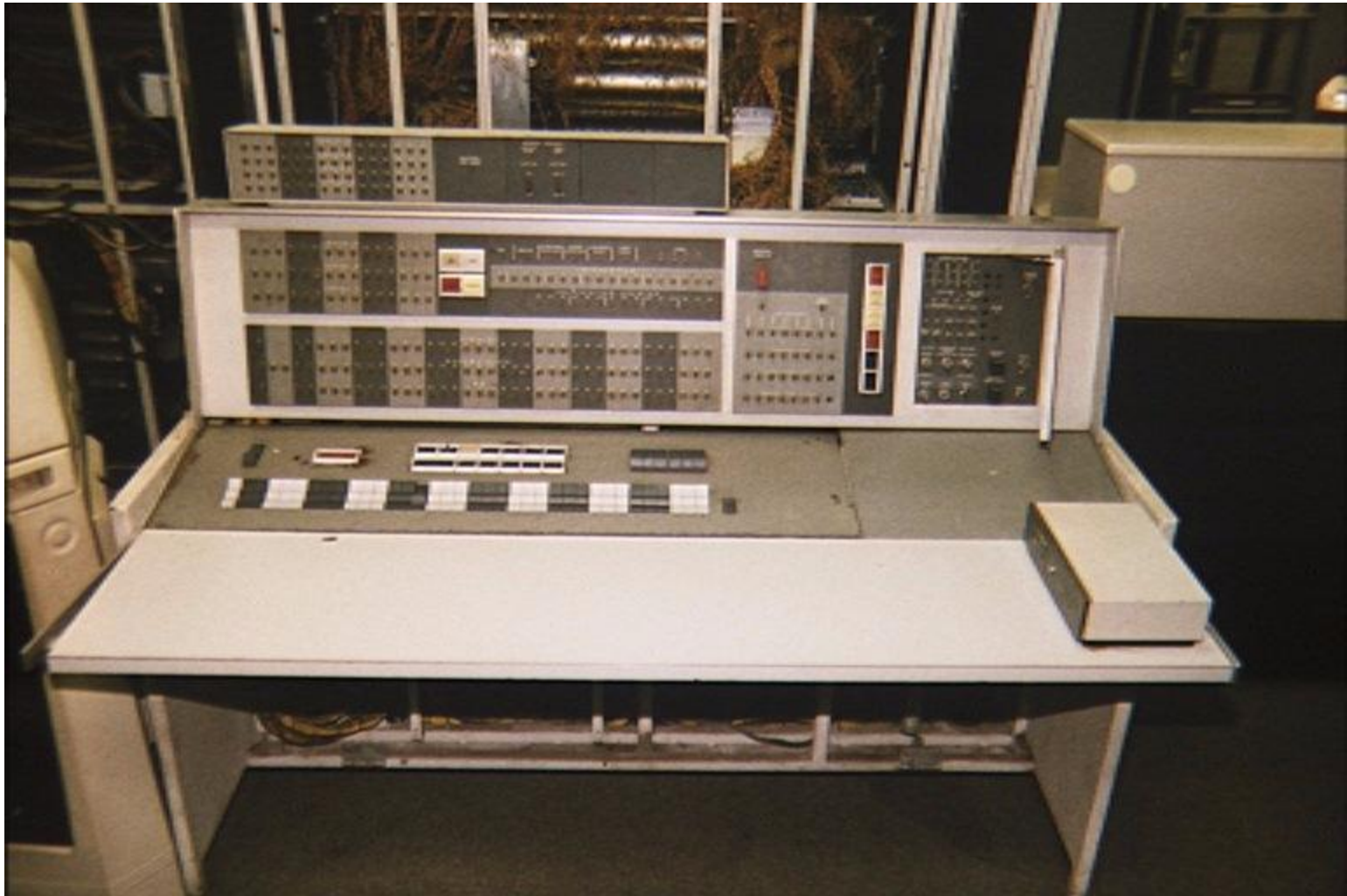
IBM 709x adat- és utasítás formátum:



IBM 7094



IBM 7094



II. Generáció (folyt.):

■ Szuperszámítógépek

- LARC: (Livermore) atom-kutatásokra
- IBM 7030 / Stretch

- **MA (2007. nov.)!:** IBM Blue Gene/L – eServer
(www.top500.org) – Livermore

- 212992 processzoros rendszer (IBM PowerPC 440s)
- 73728 GB memória
- 478200 GFLOPs teljesítmény!

■ FDE – parallelizmus

- átlapolt végrehajtás (látszólagos) - pipeline
- teljesen párhuzamos végrehajtás (több processzor) – CELL BE

IBM Blue Gene/L supercomputer



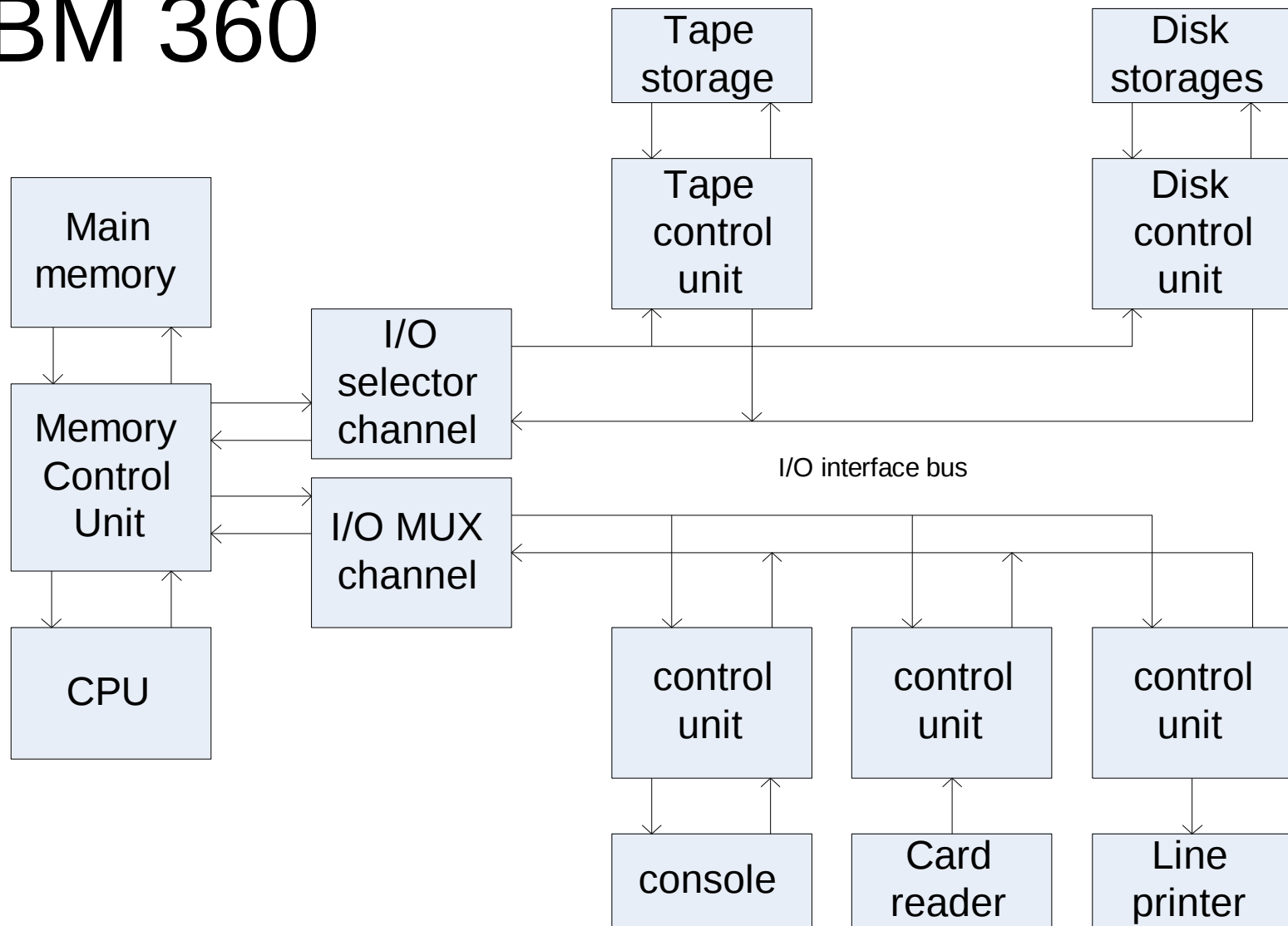
III. Generáció (1962-75):

- IC technológia
- 1965. Gordon-Moore tv: Mikro-minimalizáció
- Félvezető memóriák
- Mikroprogramozás (Wilkes 1951)
- Multiprogramozás: „time-sharing”
- Operációs Rendszerek megjelenése
- Pipeline - parallel működés
- Numerikus programozás: vektorműveletek

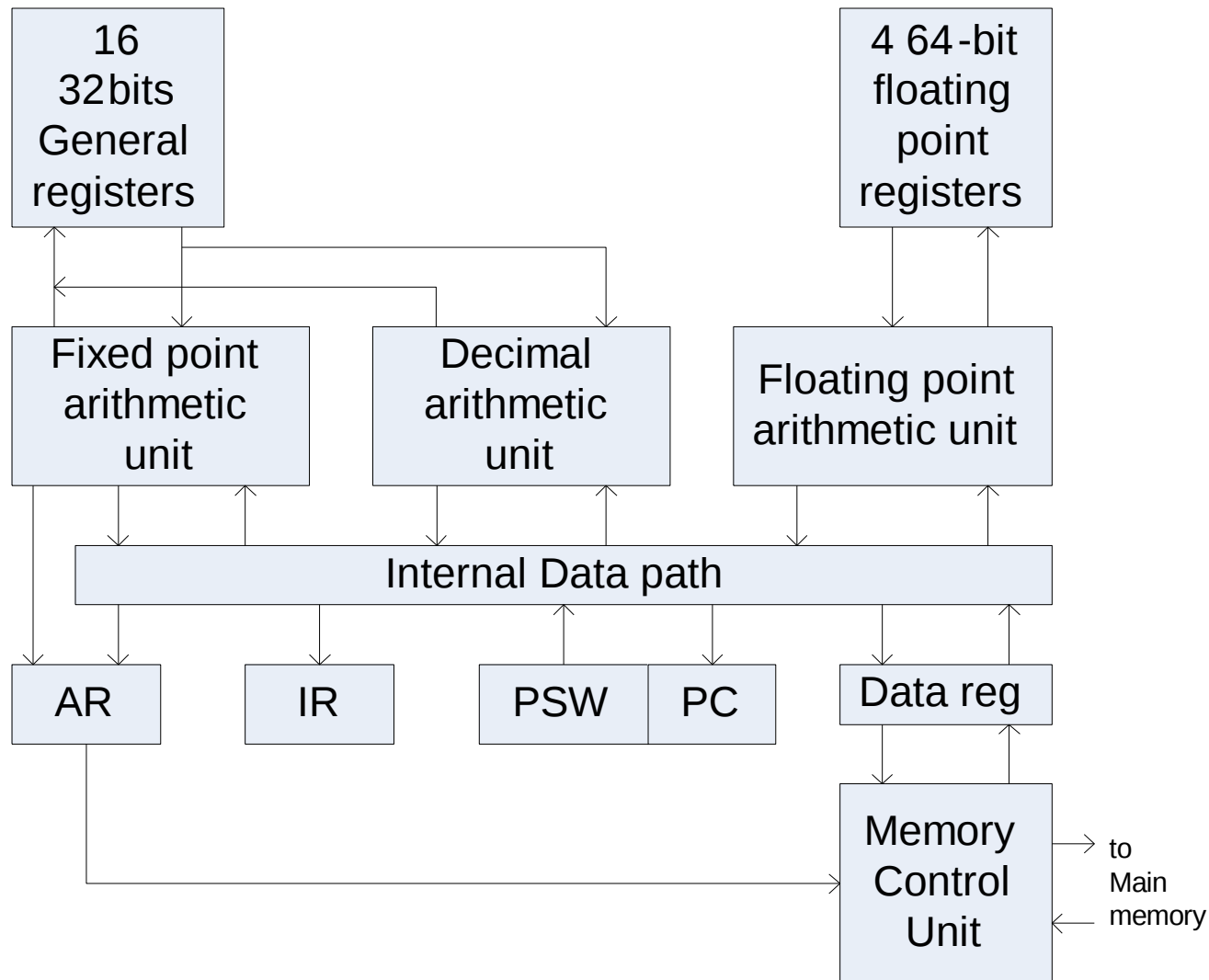
IBM 360

- első sorozatban gyártott (gépcsalád): fogyasztói célok szerinti kategóriák
- azonos utasítás készletek
- I\O csatornák seb. szerint (selector, MUX)
- 32 bites utasítások
- 8x4 bites BCD számjegyeket tárol
- 4x8 bit karakter! tárolására
- Integer / fix-point / floating-point számokat is kezel
- 16 db 32 bites ált.célú regiszter (adatok, címek)
- 4 db 64 bites lebegőpontos műveleti reg.
- Interaktív rendszer
- Virtuális memóriakezelés lehetősége
- PSW: státuszjelző regiszter (flag)

IBM 360



IBM 360 utasítás készlet:



IBM 360



IV. Generáció (1974 - ?):

- IC alapú technológia: komplexitás-méret
- Cache memóriák
- Virtuális memória rendszerek
- SoC: System On a Chip
 - Motorola 68000 – 32bites proc.
 - ALU, Regiszterek, virtuális memória egy chipen
- 4, majd 16 ... megabites memóriák
- PC: személyi számítógépek megjelenése
- Száloptika $\Rightarrow$ hálózatok (INTERNET)



V. Generáció (napjainkban):

- Ember-gép interakció (HCI)
- Felhasználóbarát szemlélet
- Ergonómia
- Mesterséges intelligencia (AI)
- Természetes nyelvi környezet:
fejlesztőeszközök (development tools)



Merre tart a technológia?

1. Roadmap projections for Semiconductor technology (prediction)

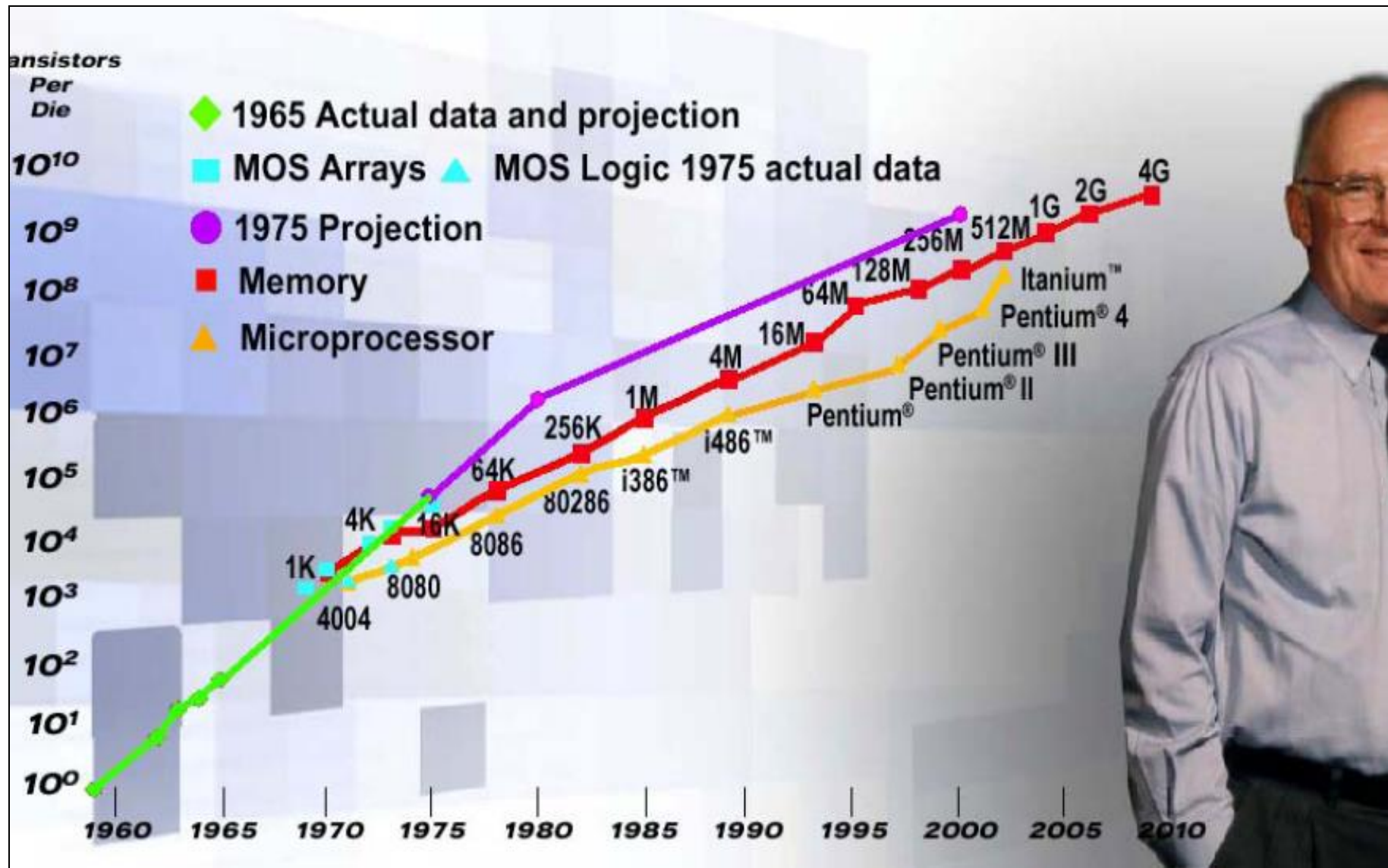
Year	Smallest feature [μm]	Dynamic Ram		Microprocessors			Wiring Levels / chip	I/O /chip
		Chip size [mm ²]	Billions of bits / chip	Chip size [mm ²]	Millions of transistors / cm ²	On-Chip Clock (MHz)		
1995	0.35	190	0,064	250	4	300	4-5	900
1998	0.25	280	0,256	300	7	450	5	1350
2001	0.18	420	1	360	13	600	5-6	2000
2004	0.13	640	4	430	25	800	6	2600
2007	0.09	960	16	520	50	1000	6-7	3600
2010	0.07	1400	64	620	90	1100	7-8	4800

2. NOW and near future:

2004	0.09-0.13					3600	7	
end of 2005	0.065	110	70 Mbit		500	?	8	
2009*	0.03							

*EUV: extrem UV lithographical technique

Moore törvénye



Flash memory

2006-II.	<50 nm	16 Gbit (max 32 GB)		16 milliárd!	Samsung CF NOR technology
2008	40-20 nm	32 Gbit (max 64 GB)			Samsung NAND CF (PRAM t)
szept.08	~20 nm	16 Gbit (max 32 GB)			Samsung SSD PRAM technology

Órajelnövelés helyett Párhuzamosítás!

Többmagos tech.

	Feature	Size	Millions of trans.	Dissipation (TPD/ACP)		
Intel Core2 Duo/Extreme	65nm	143 mm ²	291 millió	2.9 GHz	65-125 W	Conroe
AMD Athlon 64 X2 5200+	90nm SOI	199 mm ²	233 millió	2.6 GHz	65-80 W	Windsor
IBM Power6 (2 magos)	65nm SOI	341 mm ²	700 millió	<5 GHz	>100 W	
Intel Core2 Duo/Extreme	45nm (HKMG)	2x107mm ²	2 x 410 millió	2.6 GHz	130W	Penryn
Intel Core2 Quad (4 mag)	65nm	2x143mm ²	2 x 291 millió	2.6 GHz	130 W	Conroe
AMD Phenom Quad(4 m)	65nm SOI	285 mm ²	463 millió	2.2GHz	95 W	Agena
IBM Cell (8 magos) - PS3	90nm SOI	221 mm ²	231 millió	3.2 GHz	85 W	

Kvantumszámítógép

D-Wave	2007.febr.	sokváltozós feladatokra: biometrika, parametrikus adatbázisok , pénzügyi számítások számára
--------	------------	---

Cell Broadband Engine Processor

