

Úvod do Počítačových Architektur

KIV-FAV

Úvod

Obrázky:

COPYRIGHT MORGAN KAUFMANN PUBLISHERS,
INC. ALL RIGHTS RESERVED ELSEVIER

Přehled úvodní přednášky

- Úvod do „UPA“
- Přehled kurzu
- Organizační záležitosti
 - = = = = = = = = = = =
- Historie výpočetní techniky
- Úrovně abstrakce počítače
- Závěry

Přehled kurzu

- Technologické základy
- Problematika výkonnosti počítačového systému
- ISA - specifická „Instruction Set Architecture“
- Aritmetika - jak navrhnut ALU
- Konstrukce procesoru pro zvolený soubor instrukcí
- „Pipelining“ pro zlepšení výkonu
- Cache, hlavní a virtuální paměť
- I/O
- Paralelní počítače - přehled

Úvod do UPA

- Rychle se měnící oblast:
 - elektronky -> tranzistory -> IC -> VLSI
 - zdvojnásobení každých 1.5 roku (Moorův zákon):
 - *Kapacita paměti*
 - *Rychlosť procesoru (Vlivem pokroku technologie a organizace)*
- Čím se budeme v předmětu UPA zabývat:
 - jak počítače pracují, základy
 - jak analyzovat jejich výkon (popř. jak se to nemá dělat!)
 - problémy ovlivňující moderní procesory (cache, pipeline, paralelismus)

Návaznosti

- Předmět je součástí volného řetězce techničtěji zaměřených předmětů na KIVu:

POT/KIV → UPA/KIV → AČS1/KIV

Počítačová
technika

Úvod do
počítačových
architektur

Architektury
číslicových systémů 1

Proč studovat hardware

Porozumět vývoji, kam počítače směřují

- Nové možnosti otevírají (počítačový) svět
- Dopad reálného světa: bez počítačové architektury → žádné počítače!

Porozumět koncepcím návrhu vyšších úrovní

- Nejlepší systémoví návrháři se vyznají ve všech úrovních
- Hardware, kompilátory, operační systémy, aplikace

Porozumět „výkonu“ počítače

- Psaní sofistikovaného (rychlého) software vyžaduje znalosti hardware

Psaní lepšího software

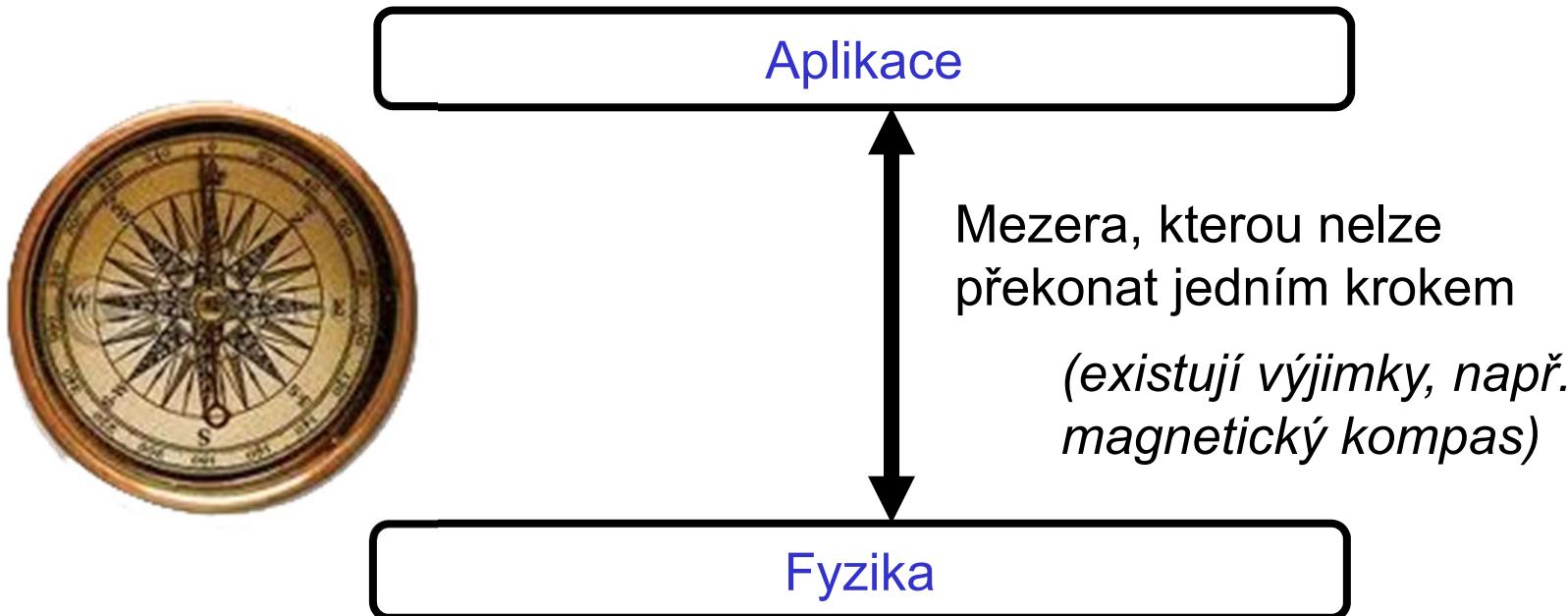
- Nejlepší softwaroví návrháři také rozumějí hardwaru
- Porozumění funkčnosti hardware i jeho omezením

Návrh hardware

- Intel, AMD, IBM, ARM, Qualcomm, Apple, Oracle, NVIDIA, Samsung,

...

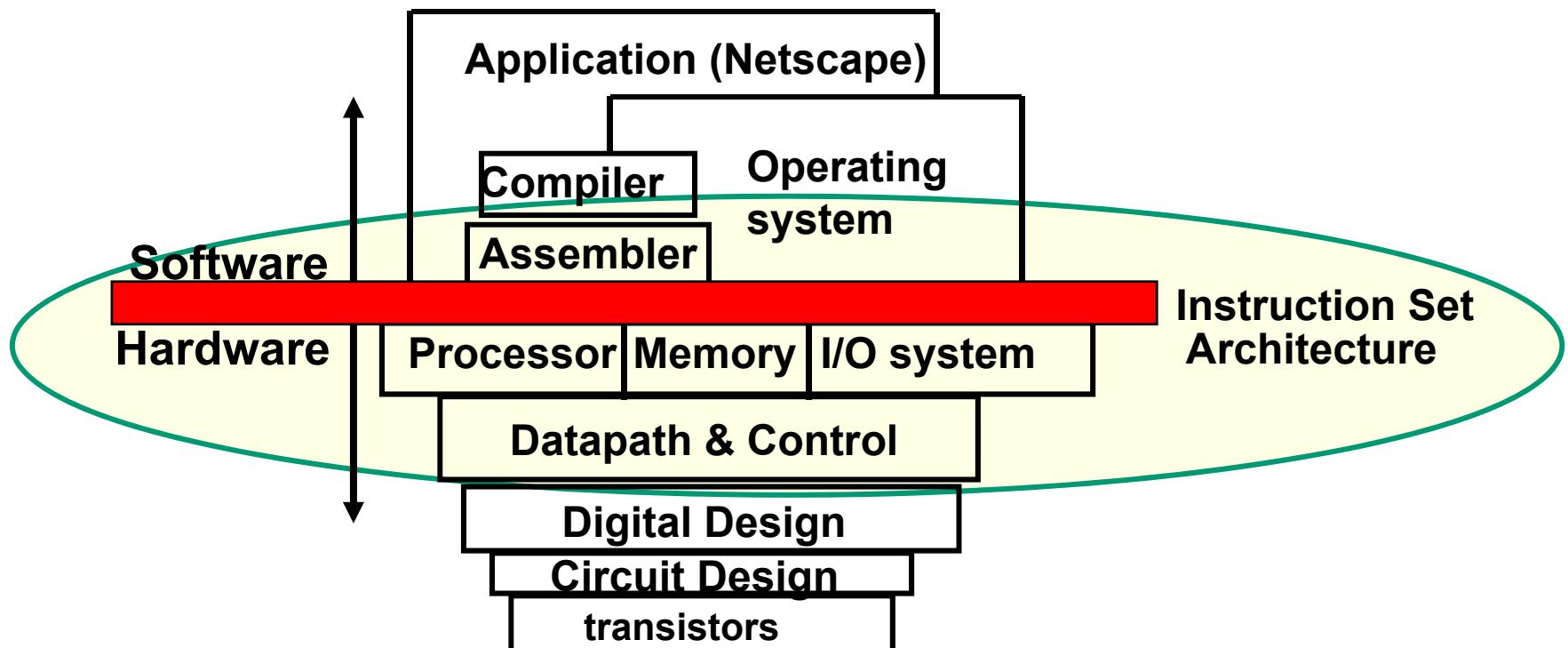
Co je architektura počítače?



V širším pojetí chápeme počítačovou architekturou **návrh úrovni abstrakce**, které dovolují implementovat aplikace (zpracování informace) s efektivním využitím dostupných výrobních technologií.

Výpočetní systém

Kde se budeme pohybovat



Koordinace mnoha *úrovni abstrakce*

O čem bude řeč

- Pět klasických částí počítače (von Neumannova koncepce počítače)
- Data mohou být vše (integer, floating point, znaky): program určuje, o co jde
- Koncepce programu uloženého v paměti: instrukce i data
- Princip lokality, využíván v hierarchickém uspořádání paměti (cache a virtuální paměť)
- Zvyšování výkonu využitím vyšší míry paralelizmu
- Princip abstrakce, používaný v komplexních systémech, budovaných ve „vrstvách“
- Kompilace v. interpretace v jednotlivých vrstvách systému
- Principy a úskalí měření výkonu počítače

Organizační údaje

Přednášky:

- Vlastimil Vavřička, KIV, (vavricka@kiv.zcu.cz)

Cvičení:

- Karel Dudáček, KIV, (dudacek@kiv.zcu.cz)

Termíny zkoušek:

- Zkoušky budou organizovány ve třech termínech během zimního zkouškového období

Organizační detaily

Postup při řešení problémů vztahujících se k předmětu:

- 
1. Konzultace s vedoucím cvičení v rámci výuky
 2. Návštěva na KIVu v době úředních hodin
 3. Mail

Podmínky získání zápočtu z UPA

- Samostatné vypracování zadaných úloh (3 úlohy)
 - o Zadání první úlohy se přidělují při cvičení ve třetím, nejpozději ve čtvrtém týdnu semestru.
 - o Odevzdání (**a akceptování !**) a všech úloh vyučujícímu (vedoucímu cvičení) do konce 13. týdne semestru včetně bude bonifikováno (10 bodů). To představuje 10% bodů z celkového počtu, které lze získat.
- Úspěšné absolvování přehledového testu, který se bude konat v jedenáctém týdnu semestru:
 - o znalosti získané na přednáškách
 - o úlohy obdobné úkolům, probíraným v rámci cvičení
- **Zápočet je možno získat jen do konce zimního zkouškového období**

Požadavky ke zkoušce z UPA

Zkouška – písemná

Podmínky pro přihlášení na zkoušku:

- Zápočet – bez zápočtu nebudou studenti zkoušeni

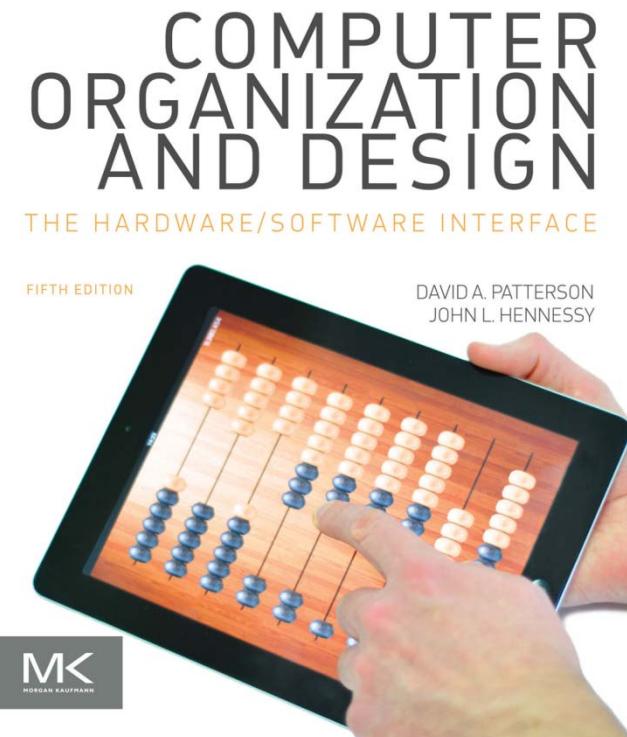
Podmínky pro složení zkoušky:

- Znalost látky v rozsahu přednášek
- Znalost témat (články), která budou určena k nastudování

**!! Poznámky z přednášek berte jen jako průvodce ke zvládnutí problematiky, nikoliv jako hlavní učební texty
a
nevyhýbejte se učebnicím !!**

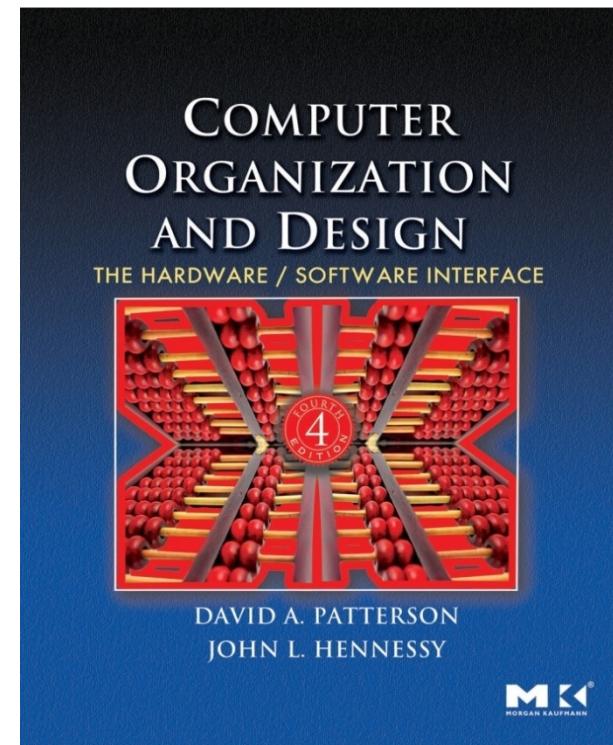
Doporučená literatura

- D. A. Patterson and J. L. Hennessy:
Computer Organization and Design: The
Hardware Software Interface, **5th
Edition**, 2014



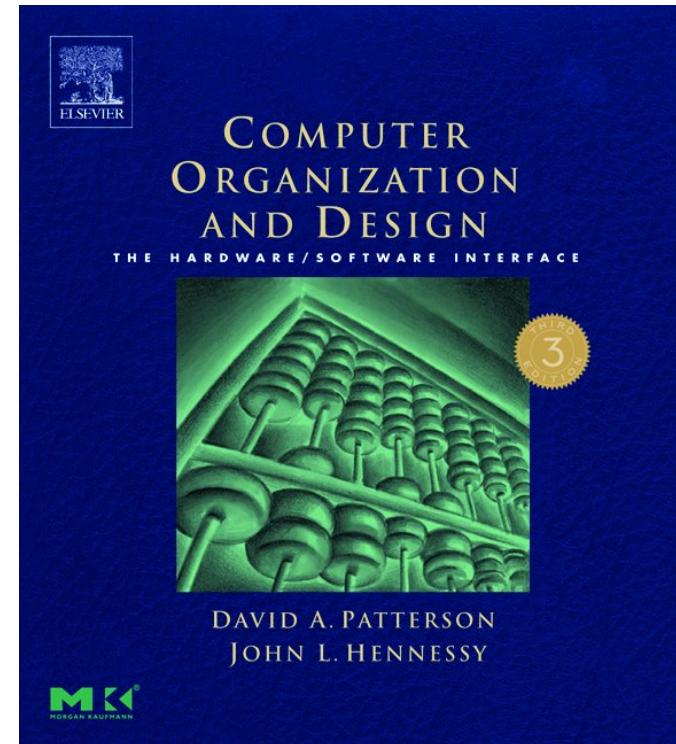
Doporučená literatura

- D. A. Patterson and J. L. Hennessy:
Computer Organization and Design: The
Hardware Software Interface, **4th
Edition**, 2009



Doporučená literatura

- D. A. Patterson and J. L. Hennessy:
Computer Organization and Design: The
Hardware Software Interface, **3rd
Edition**, 2005

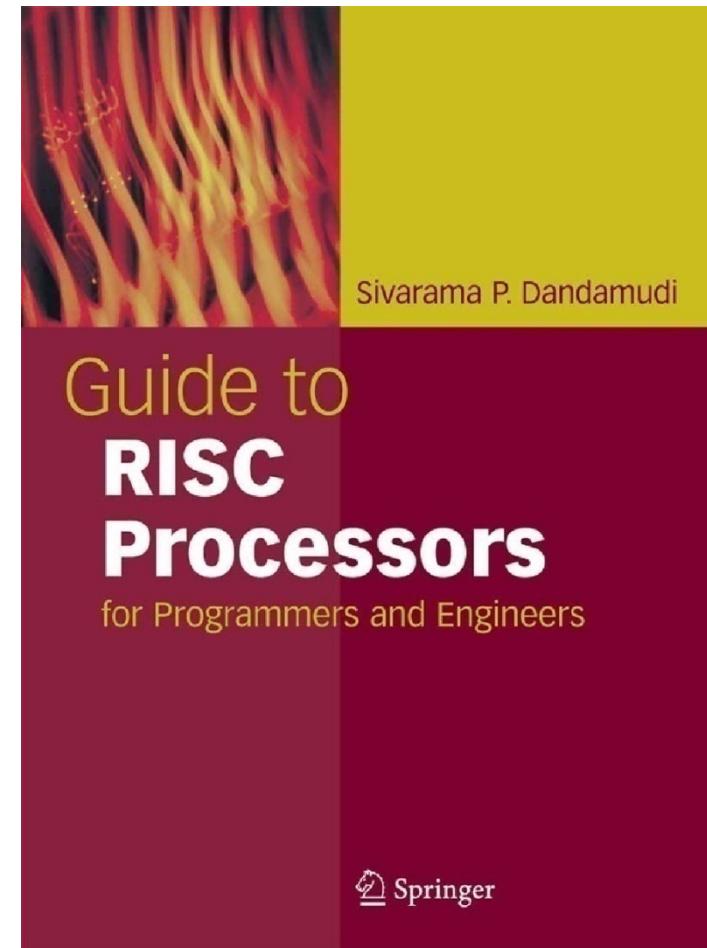


Doporučená literatura (pokr.)

Sivarama P. Dandamudi: Guide to RISC Processors for Programmers and Engineers

© 2005 Springer Science + Business Media, Inc.

ISBN 0-387-21017-2
(elektronické zdroje ZČU !!!)

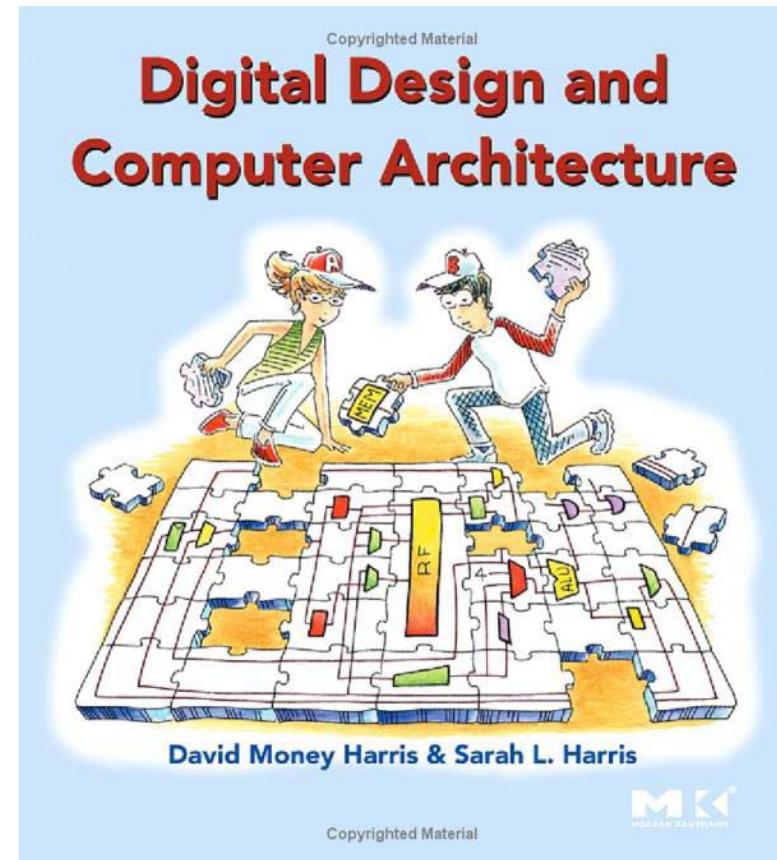


Doporučená literatura (pokr.)

David Harris, Sarah Harris: Digital Design and Computer Architecture, Second Edition

© 2013 Elsevier, Inc.

ISBN: 978-0-12-394424-5



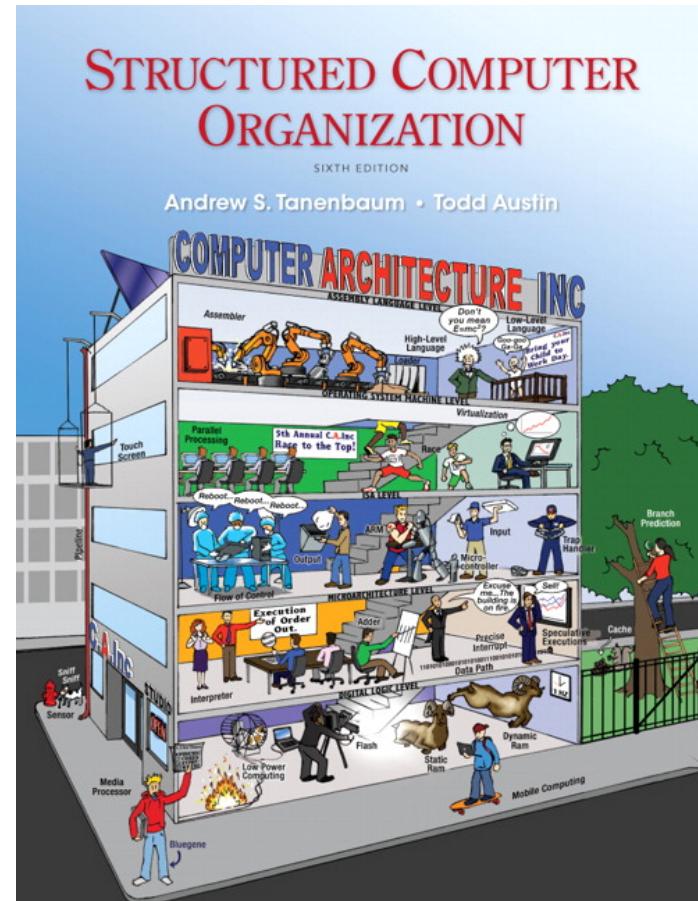
Doporučená literatura (pokr.)

Virgil Bistrițeanu, Illinois Institute of Technology:

Fundamentals of Computer Design
(jednotlivé kapitoly)

Doporučená literatura (pokr.)

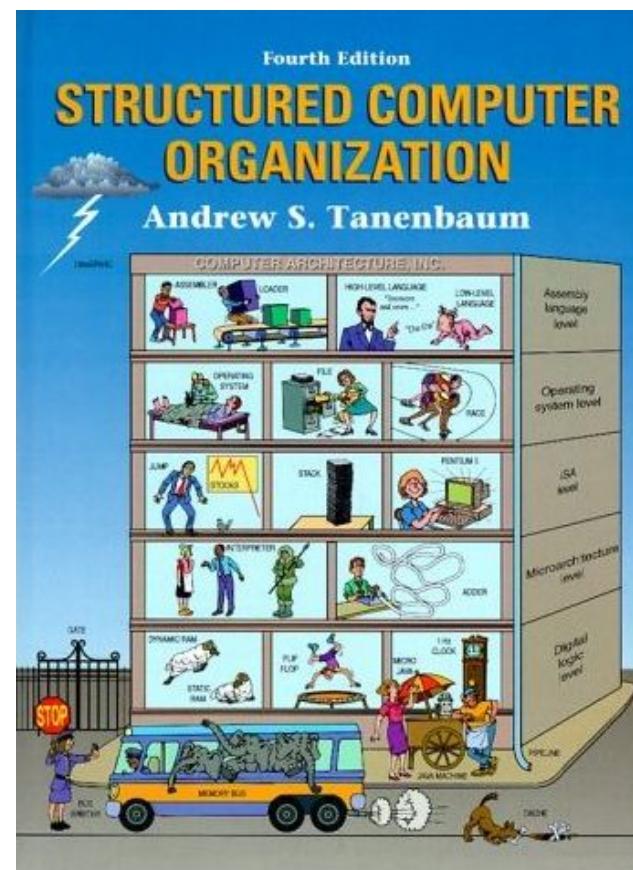
- Tanenbaum, A. S.: Structured Computer Organisation, 6th Edition, Prentice Hall 2012



Doporučená literatura (pokr.)

- Tanenbaum, A. S.: Structured Computer Organisation, 4th Edition, Prentice Hall 1999

(dostupné je i 5. vydání z roku 2005)



Doporučená literatura (pokr.)

Computer Organization and Design Fundamentals

By David Tarnoff - East Tennessee
State University

Volně ke stažení !!
– doplňkové čtení



**Computer Organization
and Design
Fundamentals**
*Examining Computer Hardware
from the Bottom to the Top*



David Tarnoff
Revised First Edition

Doporučená literatura (pokr.)

- **Digital Integrated Circuits**

Second Edition

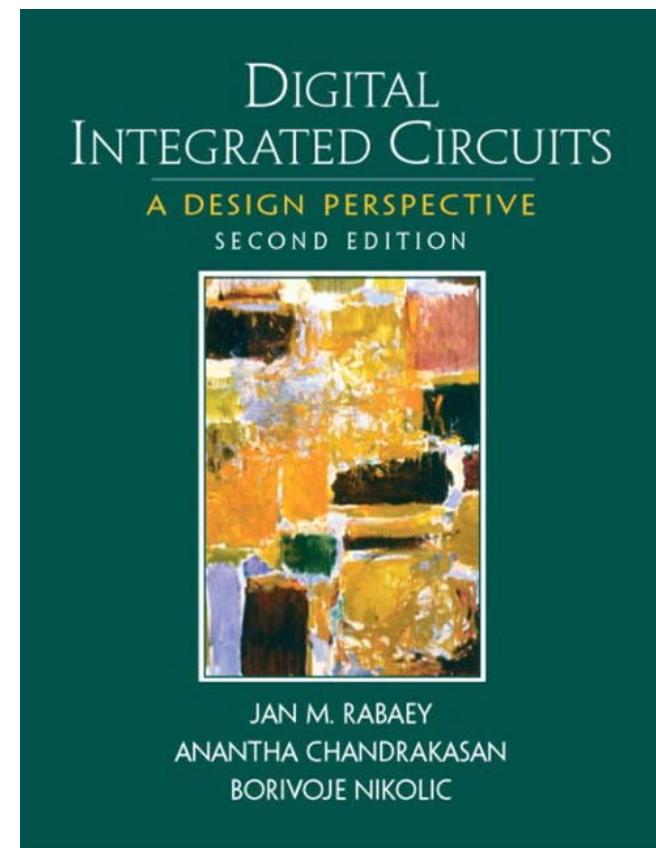
A **Prentice-Hall** publication by

Jan M. Rabaey, Anantha

Chandrakasan, and Borivoje Nikolic

<http://bwrc.eecs.berkeley.edu/IcBook/index.htm>

Doplňení znalostí základů
elektroniky



Doporučená literatura (pokr.)

- J. Douša, A. Pluháček: Úvod do počítačových systémů.
Skripta ČVUT
- Pluháček, A. : Projektování logiky počítače, ČVUT
Praha 2000
- Hlavička, J.: Číslicové počítače II, ČVUT Praha 1997

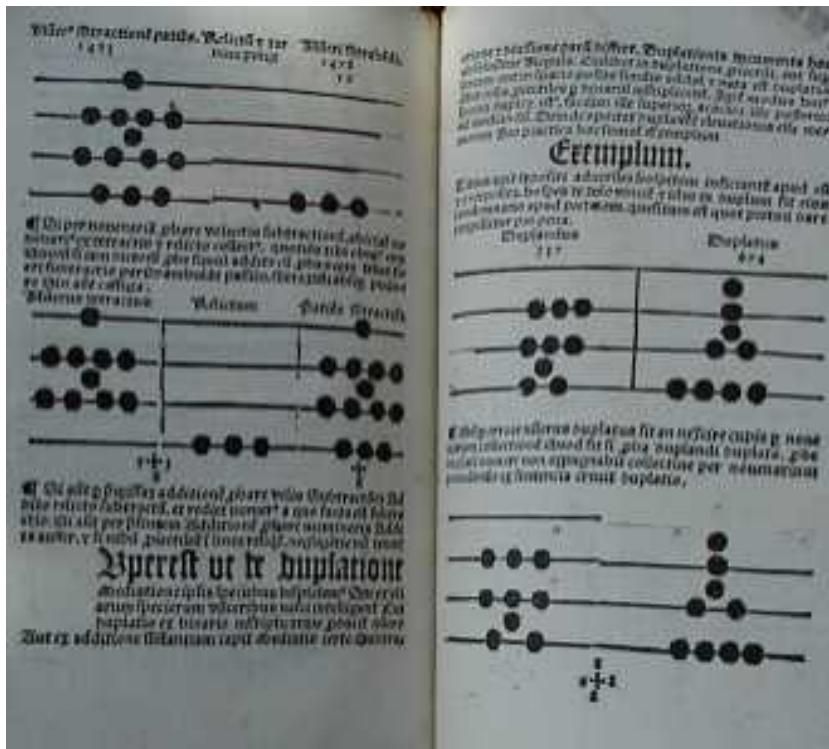
Doplňující údaje

- Možnost absolvovat část studia (obvykle semestr) v zahraničí

Stručná historie výpočetní techniky (1)

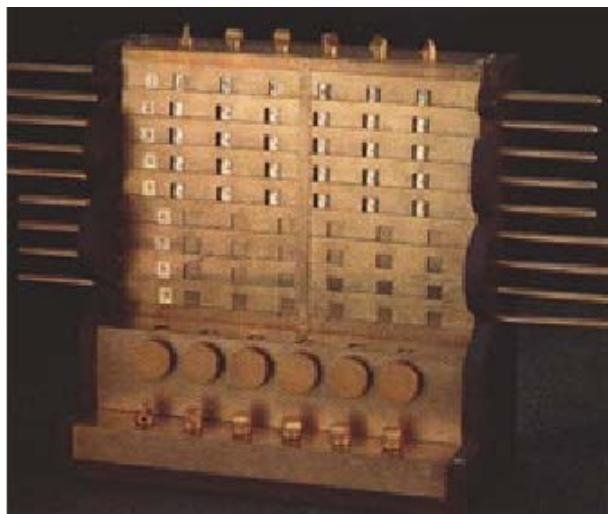
- Abacus - 1100 let př. n. l. Mechanické výpočetní pomůcky. V Číně - " Suan Pan"
- Napierchen Rechenstaebchen - 16. stol.
- Wilhelm Schickardt (1592 - 1635) v roce 1623 návrh počítacího stroje (pro početní úkony +, -, *, /). Stroj se nedochoval, údajně shořel při požáru, zachovaly se pouze nákresy v dopise J. Keplerovi.
- Blaise Pascal (1623 - 1662). V roce 1641 sestrojil do dnešní doby dochovaný sčítací stroj 8-místný Pascaline (Zwinger - Dresden). Celkem bylo vyrobeno více než 50 exemplářů, dochovalo se jen 8. Počítací stroje byly vyrobeny ze dřeva, slonoviny, železa a mědi.

Heinrich Stromer von Auerbach's Algorithmus linealis. Leipzig, 1517.



Tento text je jedním z prvních, které byly vytiskeny v 15. století a popisovaly použití pomůcky abakus. Dřevoryt reprezentuje čítače na řádcích abaku. Popis je věnován scítání, odečítání, násobení a základům aritmetiky.

Wilhelm Schickard



ZS 2015

Rok návrhu
1624

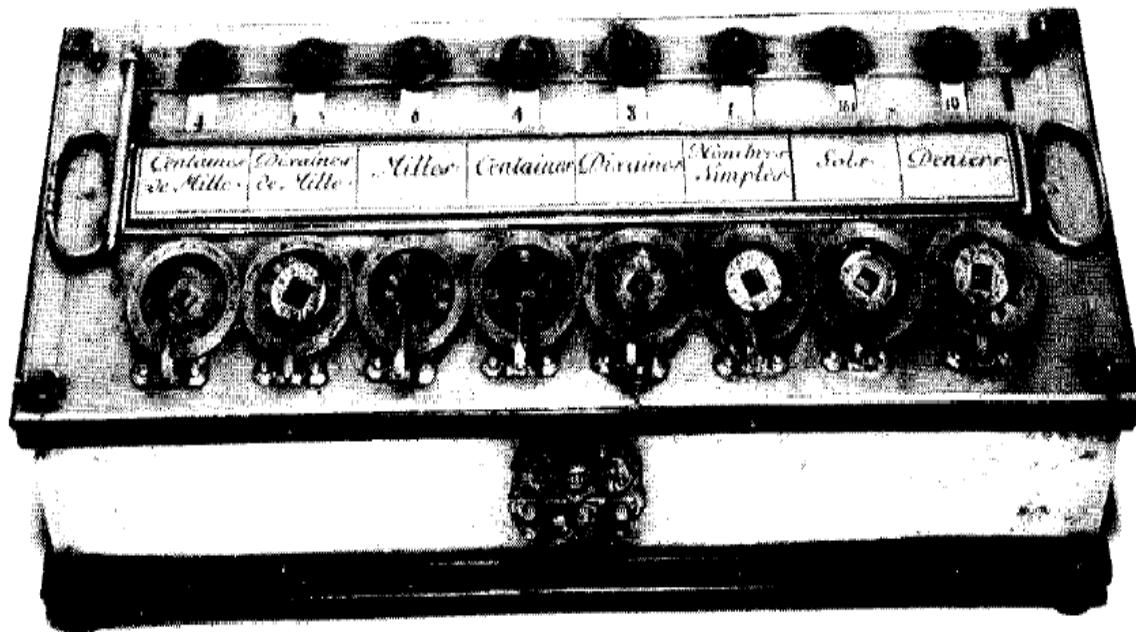


Tübingen, (1592 – 1635)

UPA

28

Blaise Pascal

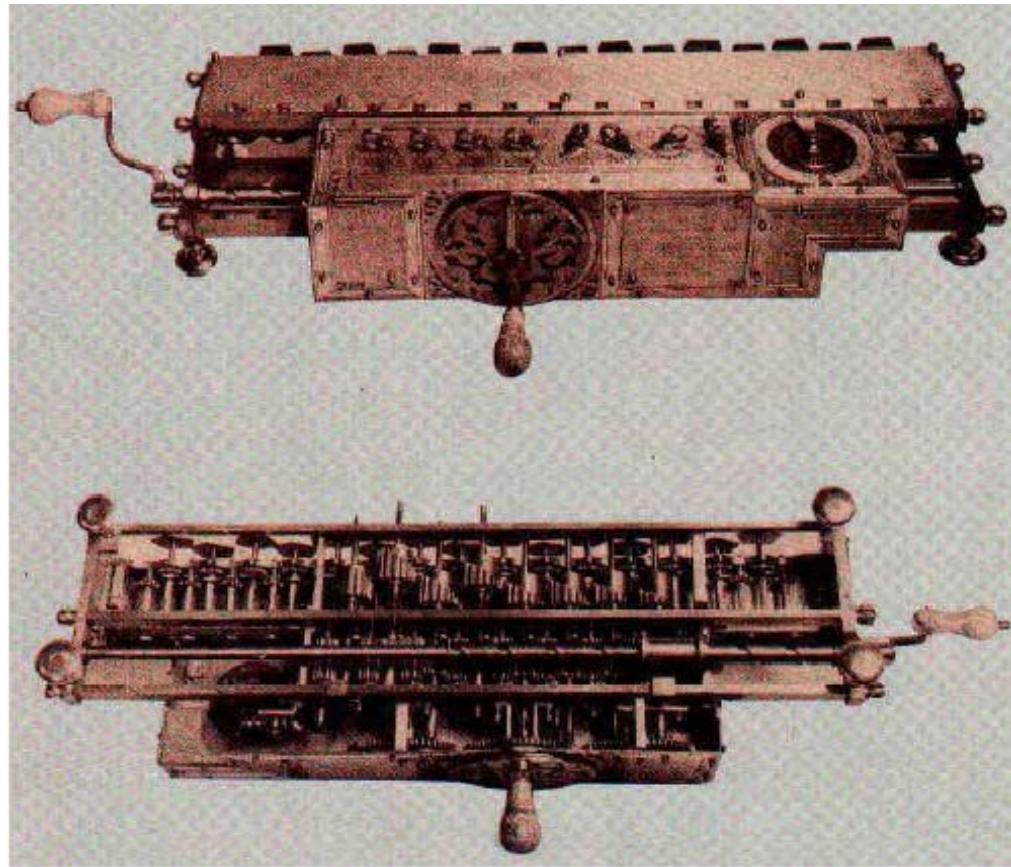


* Clermont-Ferrand, 19. June 1623
+ 1662

Stručná historie výpočetní techniky (2)

- Morland (angličan) - 1660 rozšířil Pascalův stroj o operace násobení a dělení.
- Gottfried Wilhelm Leibnitz (1646 - 1716) zkonstruoval v letech 1671 - 1674 opět dodnes zachovaný počítací stroj, který prováděl všechny čtyři základní početní úkony. Dodnes je uchovaný v muzeu v Hanoveru.
- Seth Partridge - 1650 sestrojil logaritmické pravítko - zástupce analogové techniky.
- Giovanni Poleni z Padovy - pokus o páčkový ruční kalkulační stroj.
- P. M. Hahn – farář z Würtenberku ve spolupráci se švýcarským mechanikem Schusterem sestrojili počítací stroj (+, -, *, /).

Gottfried Wilhelm, Freiherr von Leibnitz

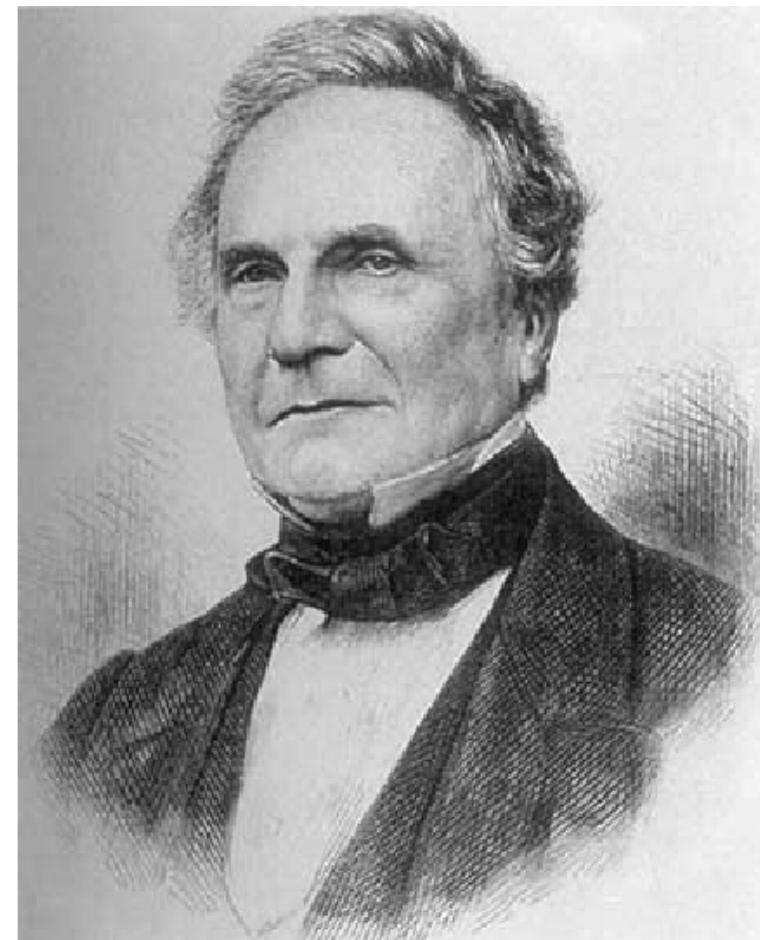
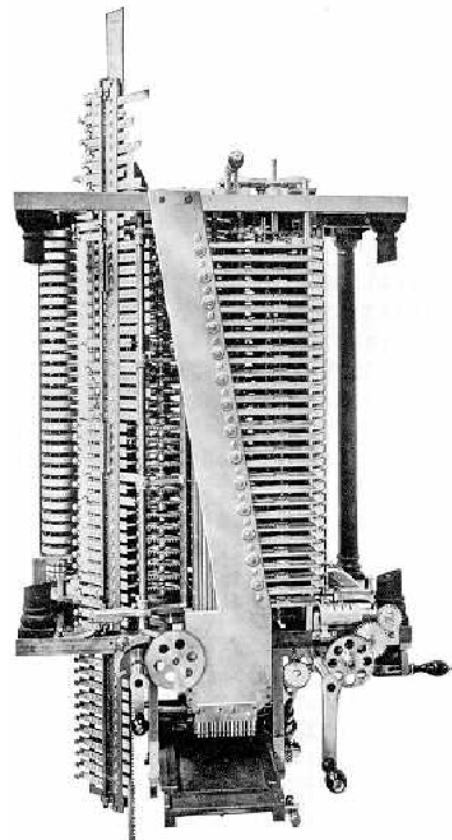


Stručná historie výpočetní techniky (3)

- Jacob Auch 1790 z Vayhingen an der Enz. Stroj zpracovával 8 číslic, prováděl operace +, -, *, /. Jacob Auch pracoval od r. 1798 ve Výmaru jako dvorní mechanik.
- Charles Xavier Thomas (1785 - 1870) zaměstnával mnoho počtářů (byl obchodník). Konstrukce "aritmometru" z r. 1820. Stroj násobil dvě 8 místná čísla za 18 sec. dělení šestnácticiferného čísla osmiferným trvalo 24 sec. Bylo vyrobeno 1500 kusů !!!
- Olivier Thomas - 1821 Paříž - továrna pro sériovou výrobu počítacích strojů, které byly navrženy podle původní Leibnitzovy konstrukce.
- Charles Babage (1792 - 1871) navrhl mechanický počítací automat, "Difference Engine" (1823). K realizaci nedošlo pro chybějící mechanické prvky.

Charles Babbage

Analytical Engine
(1833-1842)



Charles Babbage

- *Difference Engine* 1823
 - *Analytic Engine* 1833
- Vážný předchůdce moderních digitálních počítačů!

Aplikace

- Matematické tabulky – astronomie
- Námořní tabulky – mořeplavba

Základy

- Libovolnou spojitou funkci lze approximovat polynomem --- *Weierstrass*

Technologie

- mechanika – ozubené převody, Jacquardské stavy, jednoduché kalkulátory

Difference Engine

Stroj určený k výpočtu matematických tabulek

Weierstrass:

- Libovolnou spojitou funkci lze approximovat polynomem
- Každý polynom lze vypočítat pomocí *diferenčních* tabulek

Příklad:

$$f(n) = n^2 + n + 41$$

$$d_1(n) = f(n) - f(n-1) = 2n$$

$$d_2(n) = d_1(n) - d_1(n-1) = 2$$

$$f(n) = f(n-1) + d_1(n) = f(n-1) + (d_1(n-1) + 2)$$

Vše, co potřebujete, je sčítáčka!

n	0	1	2	3	4
$d_2(n)$			2	2	2
$d_1(n)$		2	4	6	8
f(n)	41	43	47	53	61

Difference Engine

1823

- Babbage' publikoval článek

1834

- Článek si přečetli Scheutz & son ve Švédsku

1842

- Babbage se vzdává myšlenky stroj postavit, pomýslí již na „Analytic Engine“!
- Babbage svůj stroj nikdy nepostavil. V roce 2002 jej sestrojili pracovníci Science Museum in London, další byl potom postaven v museu Mountain View, California.

1855

- Scheutz vystavuje svůj stroj v Paříži na světové výstavě
- Může počítat polynomy až 6. rádu
- *Rychlosť:* 33 až 44 32-ciferných čísel za minutu!

Analytic Engine

1833: Babbageův článek byl publikován

- koncipován během přestávky v práci na „difference engine“

Inspirace: *Jacquardské stavy*

- stavy byly řízeny děrnými štítky
 - Soubor štítků s fixními děrami určoval vzor tkaní ⇒ *program*
 - Stejný soubor štítků může být použit k protahování vláken různých barev ⇒ *čísla*

1871: Babbage umírá

- Stroj zůstává nerealizován.

Stručná historie výpočetní techniky (4)

- Charles Babage (1792 - 1871) navrhl další mechanický počítací automat, kde bylo možno nastavit program. "Analytic Engine" (1833). Využil principu tkacích strojů Jacquarda (1805) - děrné štítky. K realizaci nedošlo pro chybějící mechanické prvky.
- Hermann Hollerith (1860 - 1929). Z roku 1908 pocházejí první děrnoštítkové stroje. Prováděly jednoduché součty. Místo původního čistě mechanického principu byl využit princip elektromechanický. Po r. 1848 musela rodina odejít do Ameriky.
- William Thomson (Lord Kelvin) 1876 objevil princip zpětné vazby.
- Burroughův sčítací stroj, "Čuhelův aritmomet" a "Hlaváčův samočet". Zmínka v českém časopise "Z říše práce a vědy" z roku 1901.

Burrough's calculator

Burroughův sčítací stroj.

Sčítání dlouhých ciferných řad bývá v obchodech a úřadech nejen prací velmi nesnadnou, duchamornou a zdlouhavou, nýbrž stává se velmi často i zdrojem omylk, které pak nesnadno bývá vyhledati a napravit.

K zamezení takových vad sestrojíme jednoduše, aby bylo možné sčítání složitějšího, pojednávat o českých vynálezích tohoto druhu. Bylo to v roč. V. na str. 114, kde jsme popsalí Čuhelův arithmet a Hlaváčův samočet.

Dnes předvádime svým čtenářům nový stroj původu amerického, ježž vynaleze nazval arithmométerem. Zařízen jest v první řadě na sčítání a současně napisování součtu; dá se ho však též užiti k násobení.

Jak obě naše vyobrazení okazují, je to nevelká skřinka, 28 cm široká, 32 cm vysoká a 38 cm dlouhá, která na svém povrchu má 9 řad po 9 klávesách, na zadní stěně pak svítek papíru, na něž stroj sám si vytiskuje součet. Jak na obr. 2. vidíme, označena jest každá řada kláves stojícím číslem, tak že spodní řada má samé jedničky, druhá samé dvojky atd., až nejvyšší řada samé devítky. Nulových kláves vůbec tu není; nuly tisknou se samy na příslušných místech bez našeho přičinění.

Klávesy však neoznačují pouze cifernou hodnotu, která na nich jest napsána, nýbrž zároveň místní hodnotu od setin počínajíc až do milionů, což na hořejší desce pro lepsi pohled označeno jest kovovými lištami. Chceme-li tudíž strojem napsati na příklad číslo 30.278,56, stiskneme postupně 3 na třetím místě (desetitisíce), 2 na pátém místě (sta), 7 na šestém, 8 na sedmém, 5 na osmém a 6 na devátém místě. Nula na čtvrtém místě objeví se již sama. K otisknutí tohoto čísla jest pak ještě zapotřebí zatahnout klíku na pravé straně skříně do předu.

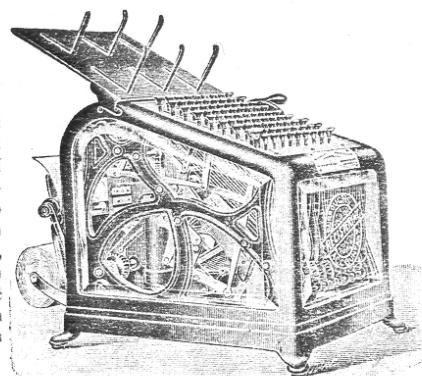
Máme-li totéž číslo několikrát po sobě napsati, stiskneme prázdný kláves na pravé

straně (dole) a otočíme klikou tolíkráte, kolíkráte číslo napsati máme. Této okolnosti užijeme výhodné k násobení. Máme se na př. číslo 4627 násobiti 26, napíšeme strojem toto číslo šestkráte, pak číslo 46270 dvakráte, načež provedeme součet. Na papíře objeví se pak otisk tento:

4627
4627
4627
4627
4627
4627
4627
46270
46270
120302

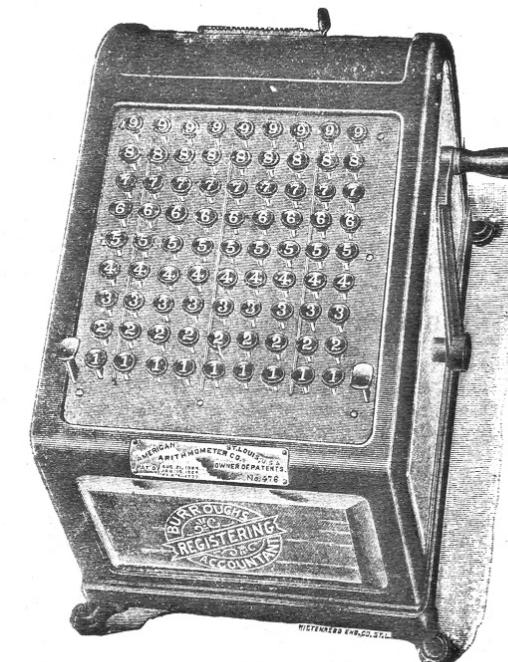
Pro sčítání různých sčítanců vyhnutáme každého sčítance zvlášt a pomocí klíky učiníme otisky, načež otočením klíky nabudeme bezvadného součtu.

Je-li počítání ukončeno, stiskneme prázdný kláves na levo (dole), čímž se celý stroj do původního (nulového) stavu navráti. Tohoto



Obr. 1. Burroughův sčítací stroj od strany.

86



Obr. 2. Burroughův sčítací stroj s hora a od předu.

klávesu lze též užiti k opravám chyb, pokud ještě nejsou vytiskeny.

Avšak Burroughův arithmométer provádí také sečítání složitějšího druhu, jak se to při knihování často vyskytuje.

Máme-li částečný součet ve stroji sečítat, aby mohl na konec s jinými částečnými součty v konečný součet být spojen, provede se celý pohyb klíkou, aniž by se některý z prázdných kláves byl stiskl, po té stiskneme se kláves sečítací (pravý), pošine se klíka ku předu, a dříve než se klíka počne zpět pohybovat, pustí se kláves.

Tím upevněn jest součet ve stroji pro další sečítání.

O výhodnosti a výkonnosti tohoto nového stroje počítacího svědčí výmluvně okolnost, že říšský poštovní úřad v Berlíně má již nyní v užívání 116 takových strojů, ačkoli každý stojí 1500 K.

Stručná historie výpočetní techniky (5)

- Udo Knor 1914 předvedl mechanický diferenciální analyzátor (řešení differenciálních rovnic).
- Prof. Konrad Zuse (1910 - 1995) 1941 Německo - sestavil reléové počítače Z1, Z2 a Z3. Poslední obsahoval 2500 relé.
- Howard Aiken a další 1941-4 USA - konstrukce počítačů Mark I, Mark II. Do vývoje investoval koncern IBM. Postaven z relé, délka 16 m, vysoký 2.5 m.
- John von Neuman (1903 - 1957) 1947 formuloval princip interního programového řízení.

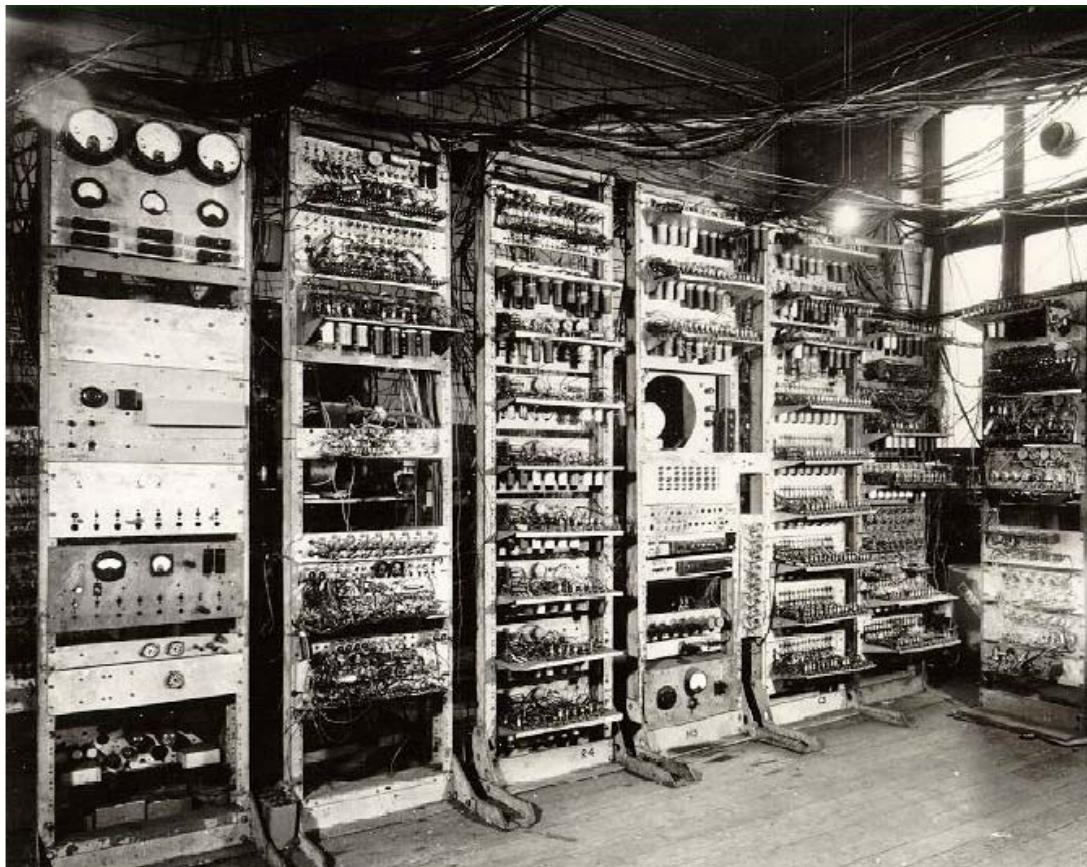
Prof. Konrad Zuse



Plankalkül, erste
Programmiersprache

(*1910 - +1995)

Manchester - Mark 1

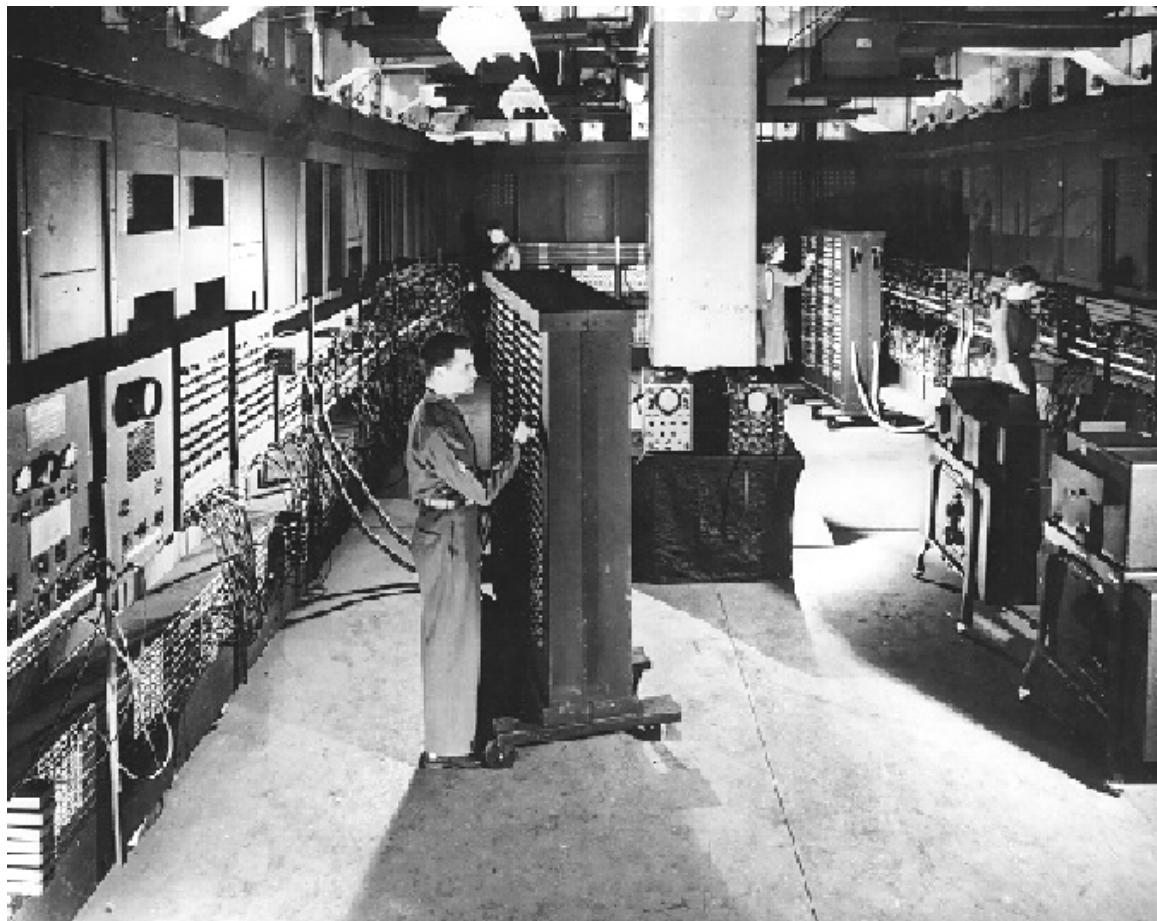


Stručná historie výpočetní techniky (6)

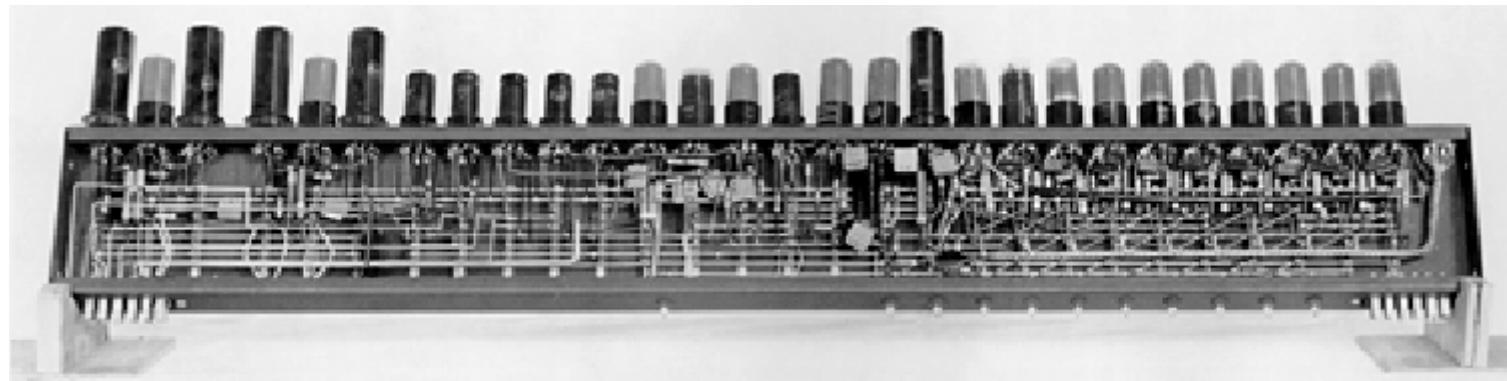
- Eniac
 - Kompletovaný r. 1946 na Moore School of Engineering (University of Pennsylvania) pod vedením Eckerta a Mauchly. Jedná se o první elektronický počítač.
 - Obsahoval: 18000 elektronek a 500 relé.
 - Měl 20 registrů a 312 slov ROM.
 - Určen pro výpočet balistických trajektorií.
 - Rychlosť 5000 op./sec.
 - Aritmetické operace se prováděly sériově, v desítkové soustavě. Každá číslice byla uložena v kruhovém registru, složeném z deseti klopných obvodů.
 - Neprogramoval se, řešená úloha byla pevně propojena.
 - Používal se 10 let.

Von Neuman zde působil jako konzultant a ovlivnil konstruktéry Eckerta a Mauchlyho, aby použili princip interního programového řízení v dalším projektu. V té době strávil léto ve Philadelphii také Wilkes, který postavil pak v roce 1949 na universitě v Cambridge (U.K.) EDSAC.

ENIAC

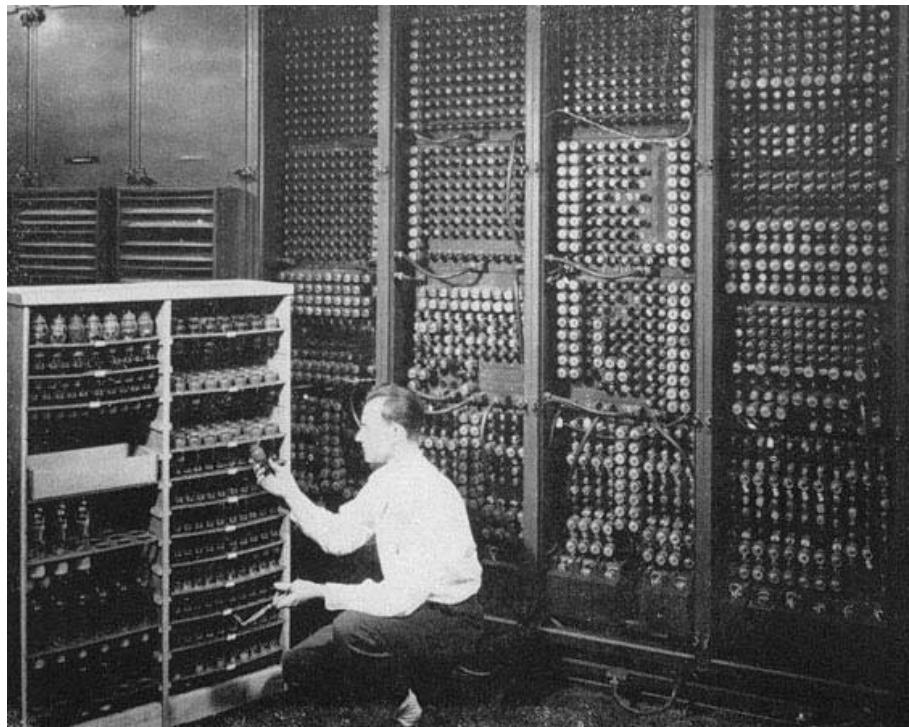


Eniac – jednotka akumulátoru



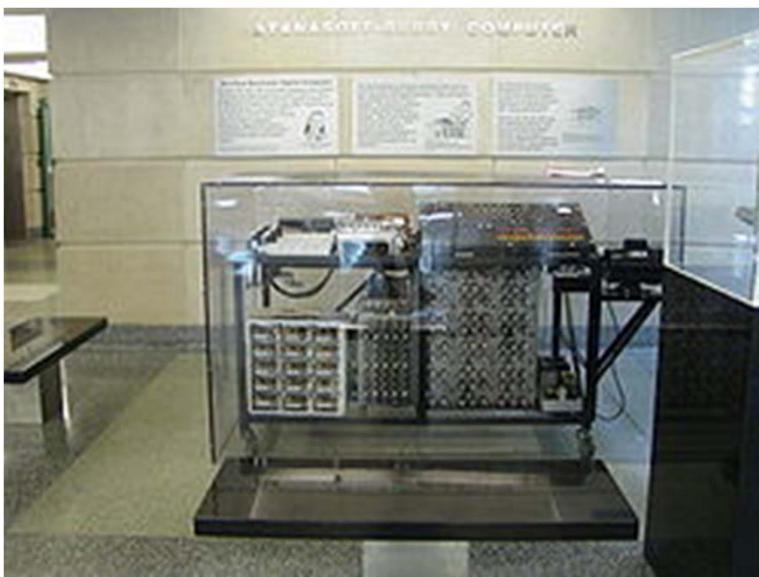
ENIAC

- ENIAC (Electronic Numerical Integrator and Calculator): Jeden z prvních elektronických počítačů pro obecné použití (general purpose)
 - Eckert a Mauchly, ve firmě U. Penn, založené U.S. Army
 - Byl zprovozněn během II. sv. války



Náhrada jedné z 18,000 elektronek

Linear Equation Solver



1930's:

- Atanasoff staví the Linear Equation Solver (Iowa State University)
- Měl 300 elektronek!
- Speciální binární digitální kalkulátor
- Dynamická RAM (hodnoty uloženy na kondenzátorech se zotavením)

Applikace:

- Lineární a integrální diferenciální rovnice

Předchůdce:

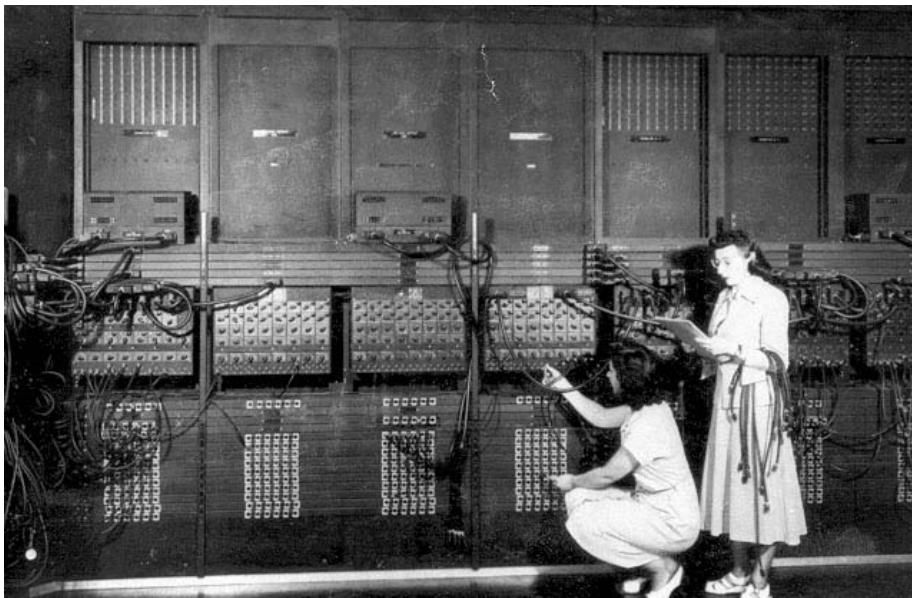
- Vannevar Bush's Differential Analyzer
--- *analogový počítač*

Technologie:

- Elektronky a elektromechanická relé

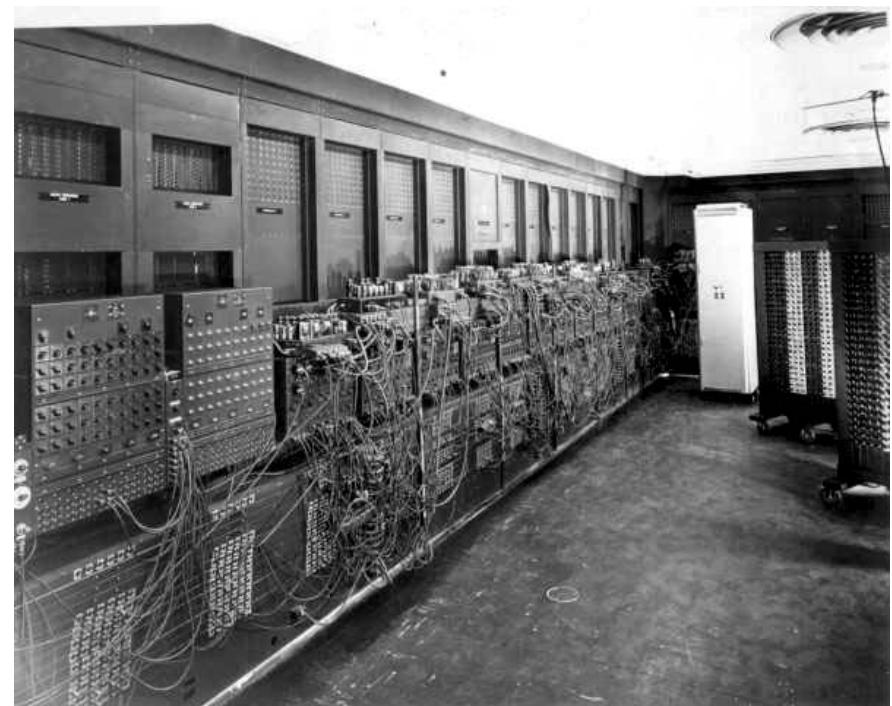
Atanasoff rozhodl, že nevhodnějším režimem výpočtu bude využití elektronické implementace binárních cifer.

ENIAC



Vytváření programu (Wiring a program)

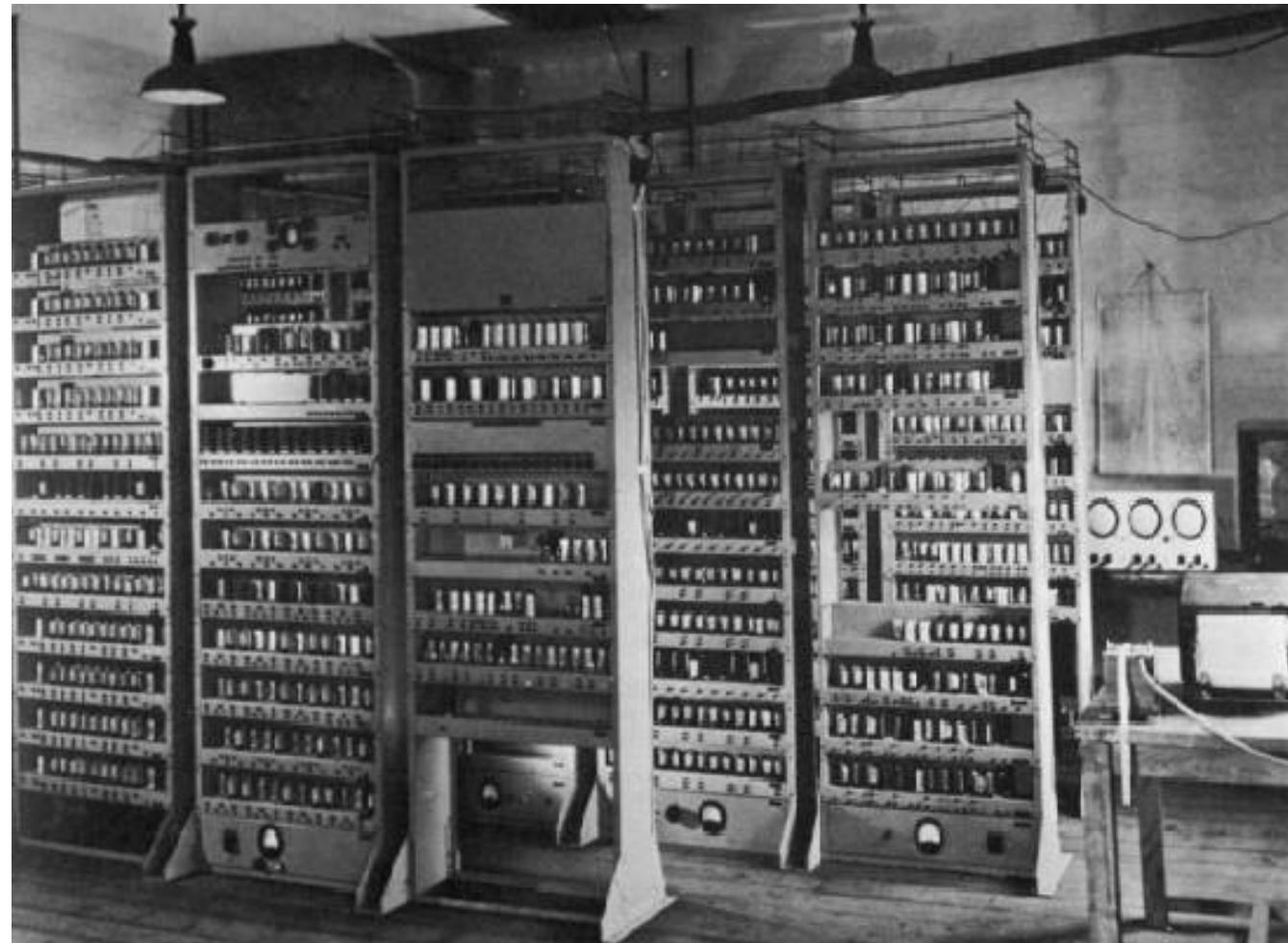
Naprogramovaný počítač
(A wired program)



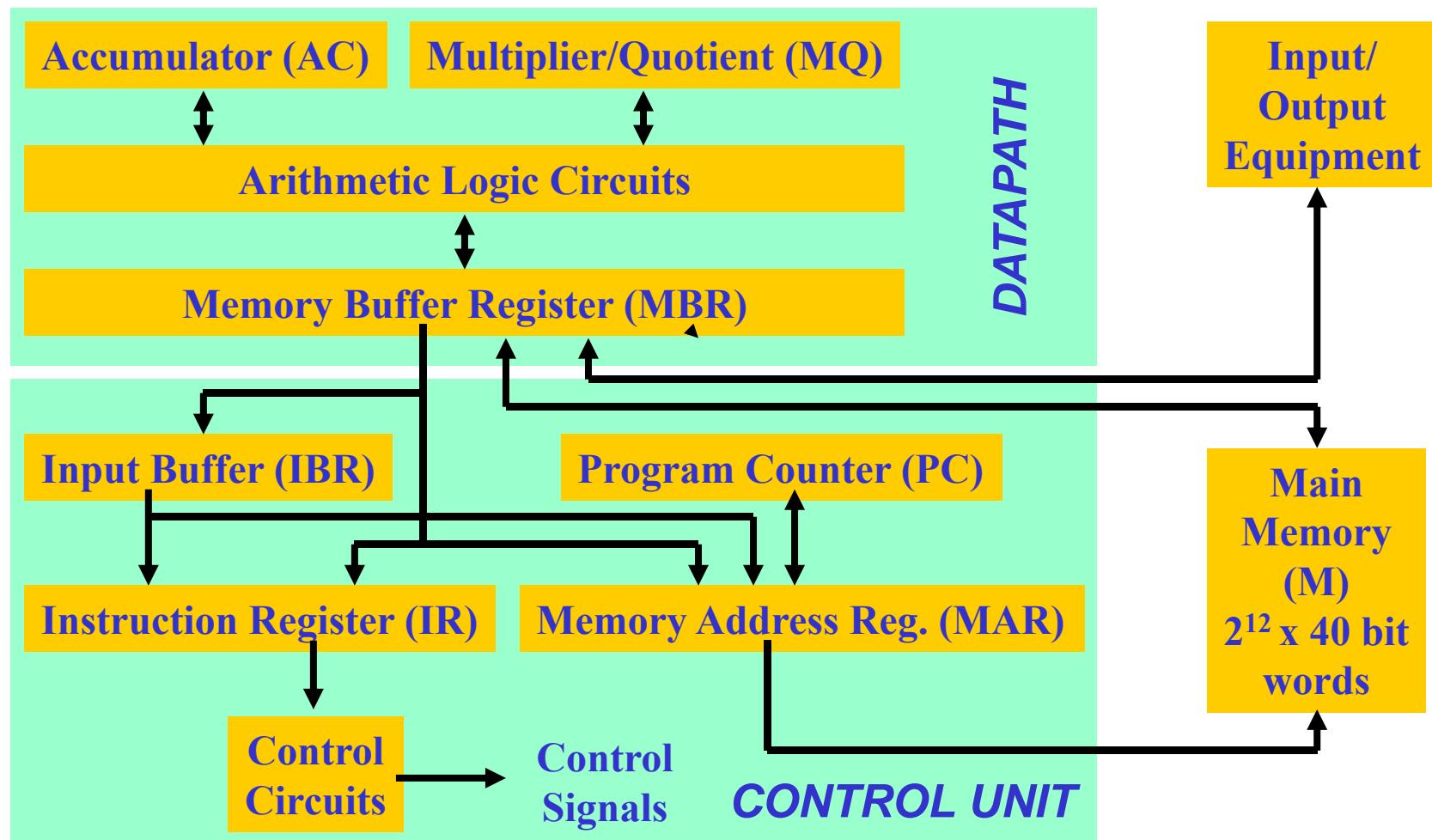
Stručná historie výpočetní techniky (7)

- Edsac
 - První počítač, který pracoval s programem, uchovaným v paměti.
 - Primární paměť měla kapacitu 1024 slov, realizovanou jako rtuťovou zpožďovací linku.
 - Dále bubnovou sekundární paměť o kapacitě 4600 slov.
 - Projekt Edsac byl zastaven, když oba tvůrci odešli ze školy a založili vlastní společnost
- IAS
 - Von Neumann a Goldstein sestavují počítač na Institute for Advanced Study.
 - Stejnojmenný počítač obsahuje známých 5 bloků.
 - Počítač má 64 instrukcí orientovaných na práci s akumulátorem. Paměť měla 4096 slov po 40 bitech.
 - Každé slovo obsahovalo 2 instrukce.

EDSAC, University of Cambridge, UK, 1949

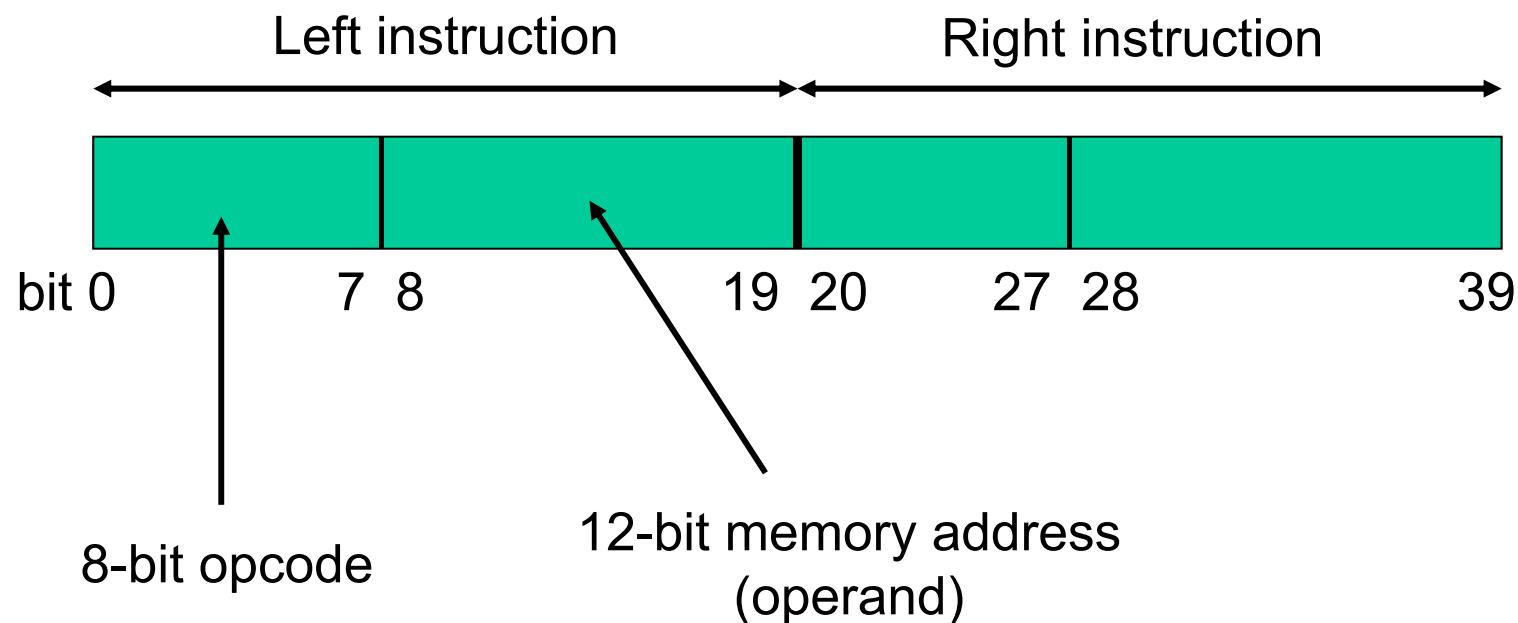


Organizace počítače IAS



Strojní jazyk IAS

40-bitové slovo, obsahující dvě strojní instrukce



Ref: J. P. Hayes, *Computer Architecture and Organization*, New York:
McGraw-Hill, 1978.

Instrukce přenosu dat (IAS)

<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>
• LOAD MQ	00001010	$AC \leftarrow MQ$
• LOAD MQ,M(X)	00001001	$MQ \leftarrow M(X)$
• STOR M(X)	00100001	$M(X) \leftarrow AC$
• LOAD M(X)	00000001	$AC \leftarrow M(X)$
• LOAD - M(x)	00000010	$AC \leftarrow - M(X)$
• LOAD M(X)	00000011	$AC \leftarrow M(X) $
• LOAD - M(X)	00000100	$AC \leftarrow - M(X) $

Instrukce nepodmíněného skoku

<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>
• JUMP M(X,0:19)	00001101	next instruction M(X,0:19)
• JUMP M(X,20:39)	00001110	next instruction M(X,20:39)

Instrukce podmíněného skoku

<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>
• JUMP +M(X,0:19)	00001111	IF AC ≥ 0 , then next instruction
M(X,0:19)	00001110	IF AC ≥ 0 , then next instruction

Aritmetické instrukce IAS

<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>
• ADD M(X)	00000101	$AC \leftarrow AC + M(X)$
• ADD M(X)	00000111	$AC \leftarrow AC + M(X) $
• SUB M(X)	00000110	$AC \leftarrow AC - M(X)$
• SUB M(X)	00001000	$AC \leftarrow AC - M(X) $
• MUL M(X)	00001011	$AC, MQ \leftarrow MQ \times M(X)$
• DIV M(X)	00001100	$MQ, AC \leftarrow MQ / M(X)$
• LSH	00010100	$AC \leftarrow AC \times 2$
• RSH	00010101	$AC \leftarrow AC / 2$

Instrukce modifikace adresy

<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>		
• STOR M(X,8:19)	00010010	M(X,8:19)	←	AC(28:39)
• STOR M(X,28:39)	00010011	M(X,28:39)	←	AC(28:39)

Sčítání dvou čísel na IAS

- Předpokládejme, že čísla jsou uložena v paměti na adresách 100 a 101
- Výsledek – suma se má uložit na adresu 102

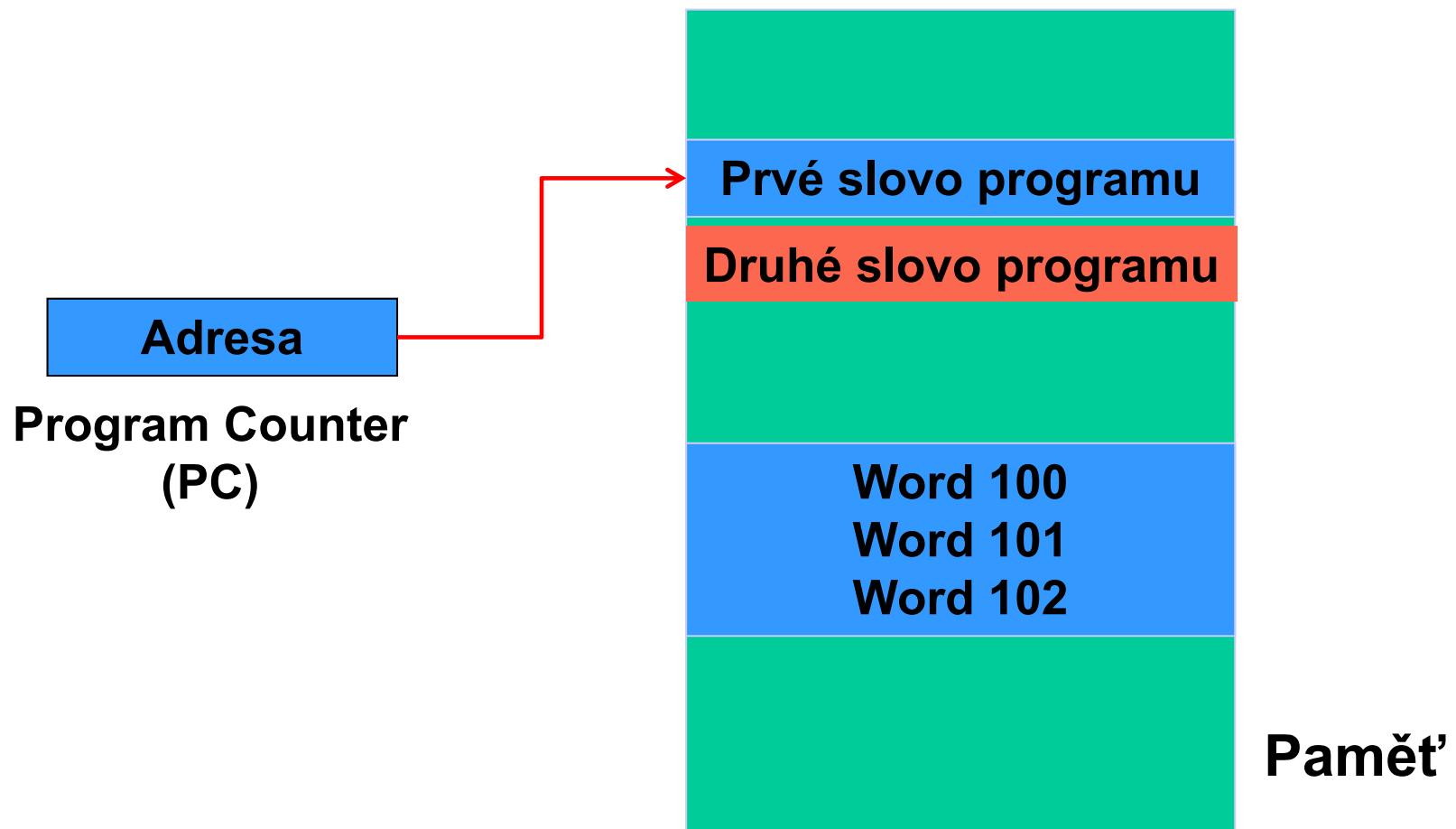
<i>Instruction</i>	<i>Opcode</i>	<i>Description</i>
LOAD M(100)	00000001	AC \leftarrow M(100)
ADD M(101)	00000101	AC \leftarrow AC+M(101)
STOR M(102)	00100001	M(102) \leftarrow AC

Strojní kód IAS

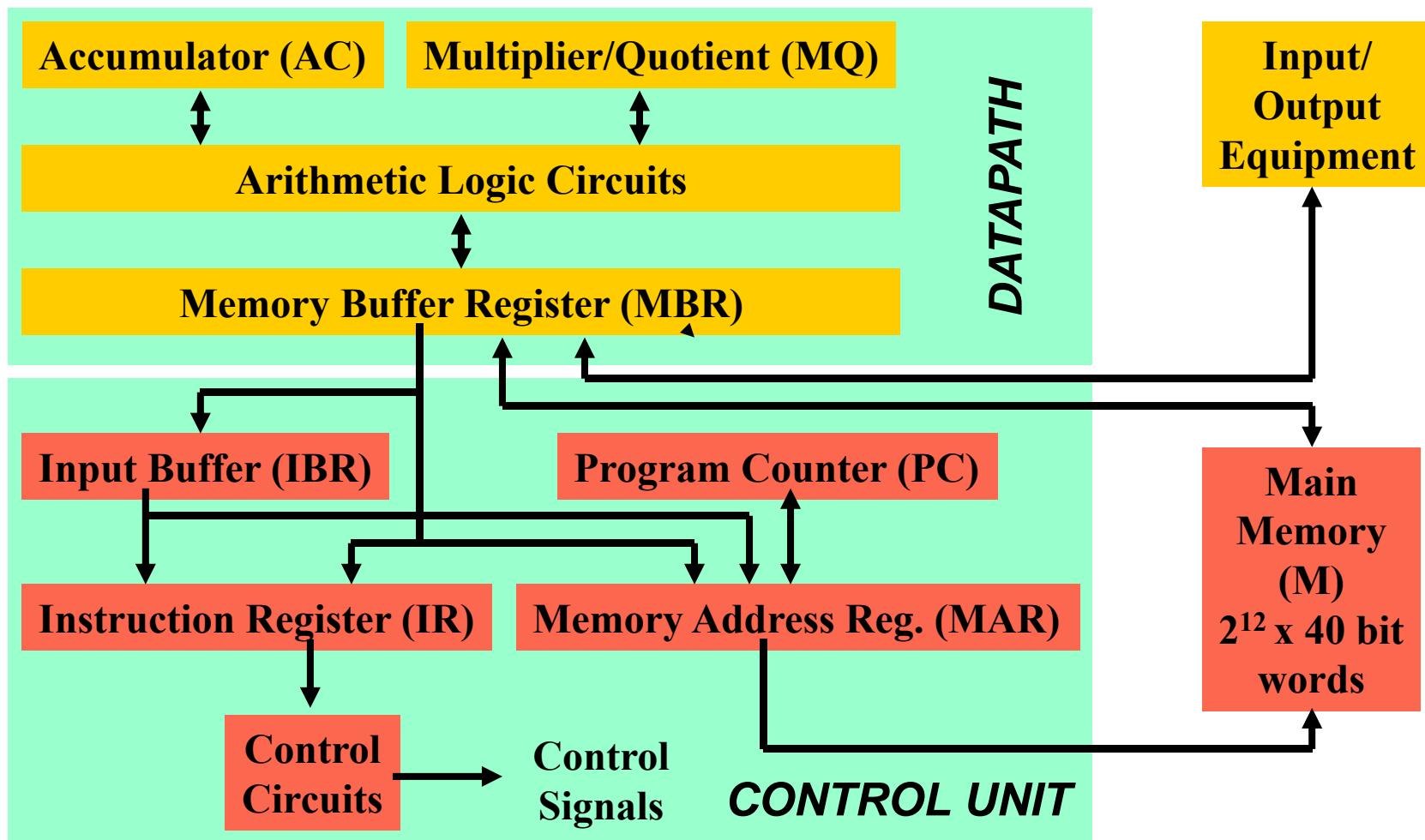
00000001	000001100100	00000101	000001100101
----------	--------------	----------	--------------

00100001	000001100110	00000000	000000000000
----------	--------------	----------	--------------

Program uložený v paměti



Provedení programu



Instrukční cykly IAS

- Strojní kód uložen v paměti na po sobě jdoucích adresách.
- Startovací adresa se uloží do čítače adres instrukcí (PC).
- Start programu: $\text{MAR} \leftarrow \text{PC}$
- Čtení paměti: $\text{IBR} \leftarrow \text{MBR} \leftarrow M(\text{MAR})$, **fetch cycle**
- Levá instrukce se zapíše do IR a adresa 100 do MAR
- Čtení paměti: $\text{AC} \leftarrow M(100)$, **fetch cycle**
- Pravá instrukce se zapíše do IR a adresa 101 do MAR
- Čtení paměti a operace součtu: $\text{AC} \leftarrow \text{AC} + M(101)$,
execution cycle
- $\text{PC} \leftarrow \text{PC} + 1$

Instrukční cykly IAS (pokr.)

- $\text{MAR} \leftarrow \text{PC}$
- Čtení paměti: $\text{IBR} \leftarrow \text{MBR} \leftarrow M(\text{MAR})$, **fetch cycle**
- Levá instrukce se zapíše do IR a adresa 102 do MAR
- $\text{MBR} \leftarrow \text{AC}$
- Operace zápisu do paměti

Stručná historie výpočetní techniky (8)

Rychle se rozbíhají další projekty:

- Illiac - University of Illinois
- Johniac - RAND Corporation
- Maniac - Los Alamos
- Weizac - Weizman Institute in Israel

Stručná historie výpočetní techniky (9)

- Whirlwind - MIT (1947-1951)
 - Slovo 16 bitů, 5 bitů OP.C.
 - 27 instrukcí
 - 2048 slov
 - 20000 op/s., r. 1951 byla paměť na principu paměť. elektronek nahrazena feritovou jádrovou pamětí o kapacitě 2K slov a dobou cyklu 8 us
- !!! MTBF byla 20 minut !!!**

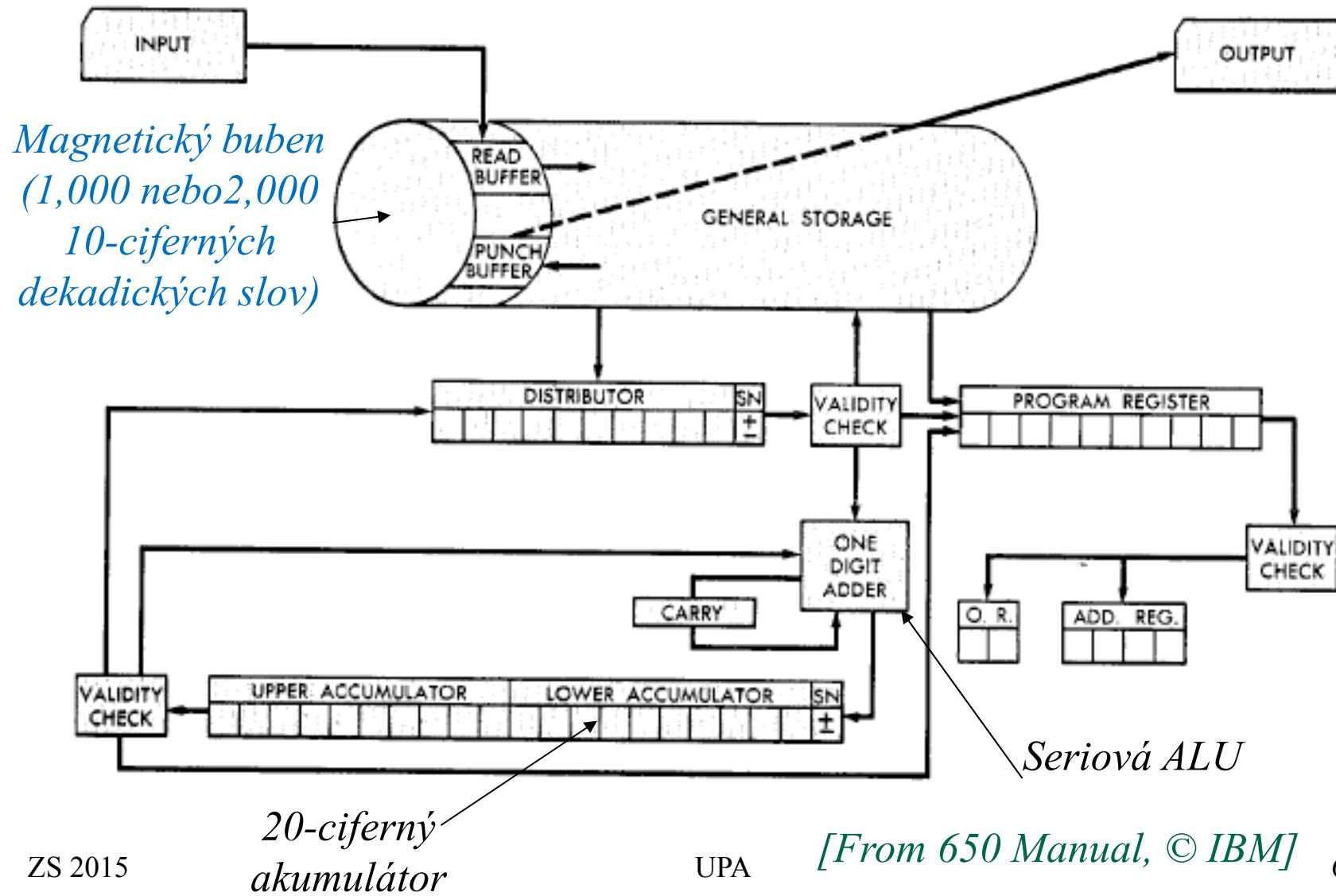
Eckert a Mauchly opustili universitu v Pensylvanii a začali stavět malý binární počítač Binac. Dva procesory, výsledek se porovnával kvůli spolehlivosti. Většinou se neshodly.

Binac (1950) - Nepracoval podle programu v paměti a nikdy ho neodladili.

Stručná historie výpočetní techniky (10)

- Univac I (1951)
 - Magnetická páška 128 char/inch, 1.44 Mbit max.
 - čtení vpřed i vzad
 - používal desítkovou soustavu, 12 cifer
 - instrukce obsahovala 1 adresu, dvě tvořily slovo
 - paměť – rtuťová zpožďovací linka
- Univac II
 - Kompatibilní s Univac I
 - používal feritovou paměť

IBM 650 (1953-4)



IBM 650 z hlediska programátora

Bubnový stroj se 44 instrukcemi

- Instrukce: 60 1234 1009
 - "Přečti obsah buňky 1234 do *distributoru*; dále do *horní části akumulátoru*; vynuluj *dolní část akumulátoru* a pak jdi na adresu 1009 pro další instrukci."

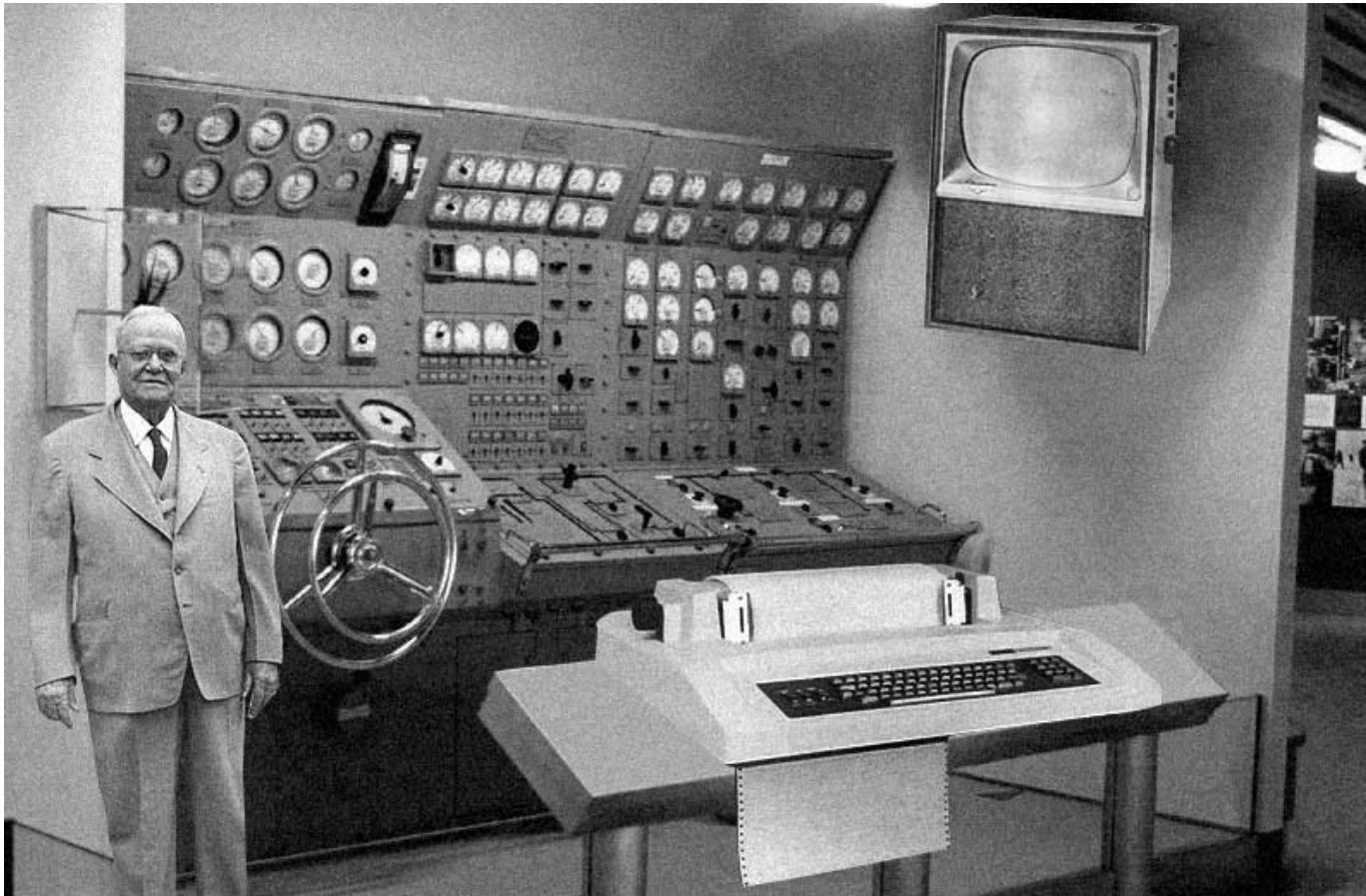
Dobří programátoři optimalizovali umístění instrukcí na bubnu tak, aby se redukovala latence bubnové paměti!



Stručná historie výpočetní techniky (11) - počítače druhé generace

- 1955 to 1964
- Elektronky byly nahrazeny tranzistory
- Magnetické jádrové paměti
- Floating-point aritmetika
- Jazyky vyšší úrovně: ALGOL, COBOL a FORTRAN
- Systémový software: kompilátory, knihovny podprogramů, batch processing
- Příklad: IBM 7094

Představy o PC z let minulých



Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use and only

Problém kompatibility u IBM

Na počátku 60-tých let, měla firma IBM
4 nekompatibilní řady počítačů!

701	→	7094
650	→	7074
702	→	7080
1401	→	7010

Každý systém měl svůj vlastní:

- instrukční soubor
- I/O systém a sekundární paměť:
magnetické pásky, bubny a disky
- asemblery, kompilátory, knihovny, ...
- svoje vlastní zajištění na trhu, ...

⇒ *IBM 360*

IBM 360 : Premisy návrhu

Amdahl, Blaauw and Brooks, 1964

- Návrh musí být podřízen požadavkům na růst a rozvoj následných řad počítačů
- Obecná metoda pro připojování periferií
- Celkový výkon – *počet odpovědí za měsíc* lépe než *počet bitů za mikrosekundu* ⇒ *programovací prostředky*
- Stroj se musí sám kontrolovat, bez manuálních zásahů
- Vestavěná *hardwareová kontrola chyb* a *detekční prostředky* pro zkrácení doby poruchy
- Jednoduché sestavení systémů s redundantními I/O zařízeními, pamětími atd. pro dosažení „*fault tolerance*“
- Řešení některých problémů vyžaduje *floating point* zobrazení delší než 36 bitů

IBM 360: Univerzální registrový (GPR) stroj

- Stav procesoru
 - 16 univerzálních 32-bitových registrů
 - mohou být využity jako indexregistry nebo jako báze
 - registr 0 má zvláštní vlastnosti
 - 4 FP 64-bitové registry
 - Program Status Word (PSW)
 - PC, podmínkové kódy, řídící bity
- 32-bitový stroj se 24-bitovou adresou
 - ale žádná instrukce neobsahovala 24-bitovou adresu!
- Datové formáty
 - 8-bitové byty, 16-bitová půlslova, 32-bitová slova, 64-bitová dvojslova

IBM 360 je příčinou proč má byte dnes 8-bitů!

IBM 360: Počáteční implementace

	<i>Model 30</i>	...	<i>Model 70</i>
<i>Storage</i>	8K - 64 KB		256K - 512 KB
<i>Datapath</i>	8-bit		64-bit
<i>Circuit Delay</i>	30 nsec/level		5 nsec/level
<i>Local Store</i>	Main Store		Transistor Registers
<i>Control Store</i>	Read only 1 μ sec		Conventional circuits

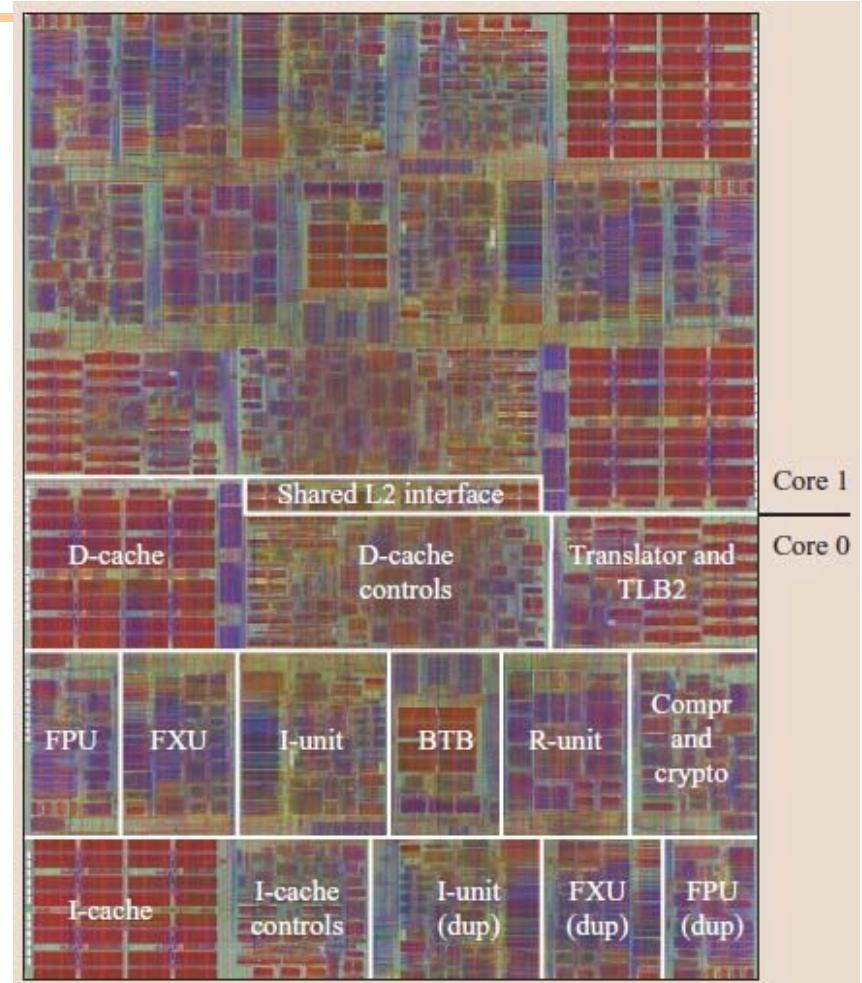
Architektura instrukčního souboru IBM 360 (ISA) úplně překryla rozdíl mezi jednotlivými modely.

Mezník: První pravá ISA navržená jako přenositelný hardware-software interface!

S mírnými modifikacemi přežívá dodnes!

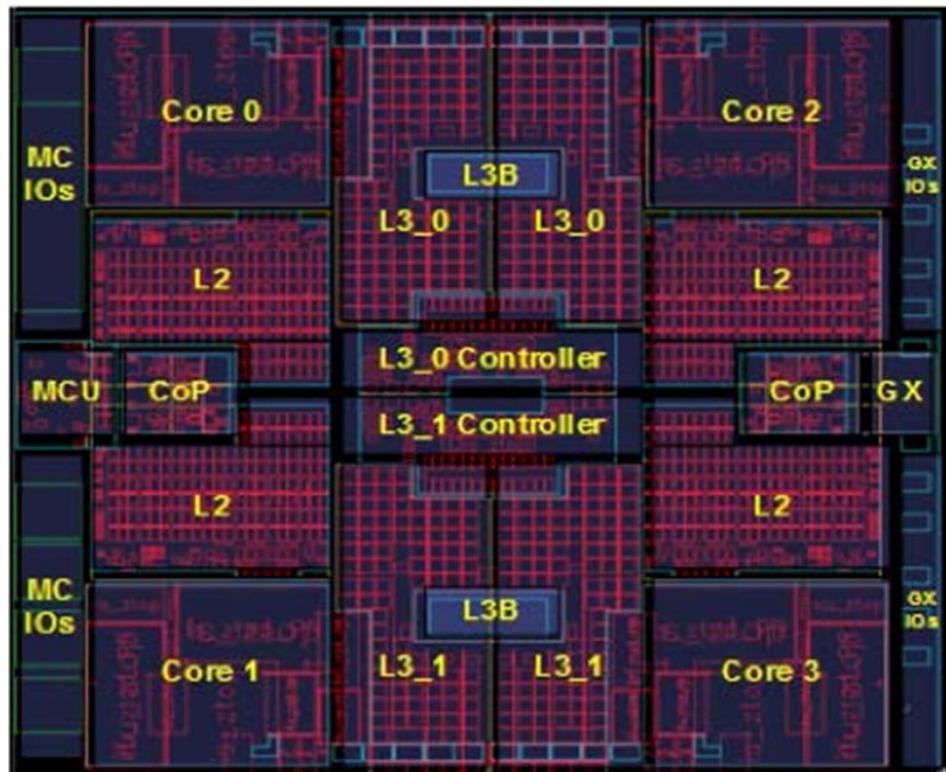
IBM 360: Po 40 letech...Mikroprocesory řady z990

- 64-bitové virtuální adresování
 - originálně u S/360 bylo 24-bitové a u S/370 byla 31-bitová extenze
- Návrh se dvěma jádry
- Dvouproudová superskalární architektura
- 10-stupňová CISC pipeline
- Out-of-order přístupy do paměti
- Redundantní datové cesty
 - každá instrukce se provádí ve dvou paralelních jednotkách a výsledky se porovnávají
- 256KB L1 I-cache, 256KB L1 D-cache na chipu
- 32MB sdílená L2 unifikovaná cache externí
- 512-entry L1 TLB + 4K-entry L2 TLB
 - rozlehlý TLB pro podporu násobných virtuálních strojů
- 8K-entry Branch Target Buffer
 - rozlehlý buffer pro podporu větvení programů
- Až 64 procesorů (48 viditelných zákazníkovi) v jednom stroji
- 1.2 GHz v IBM 130nm SOI CMOS technologii, 55W pro obě jádra



[IBM Journal R&D, 48(3/4), May/July 2004]

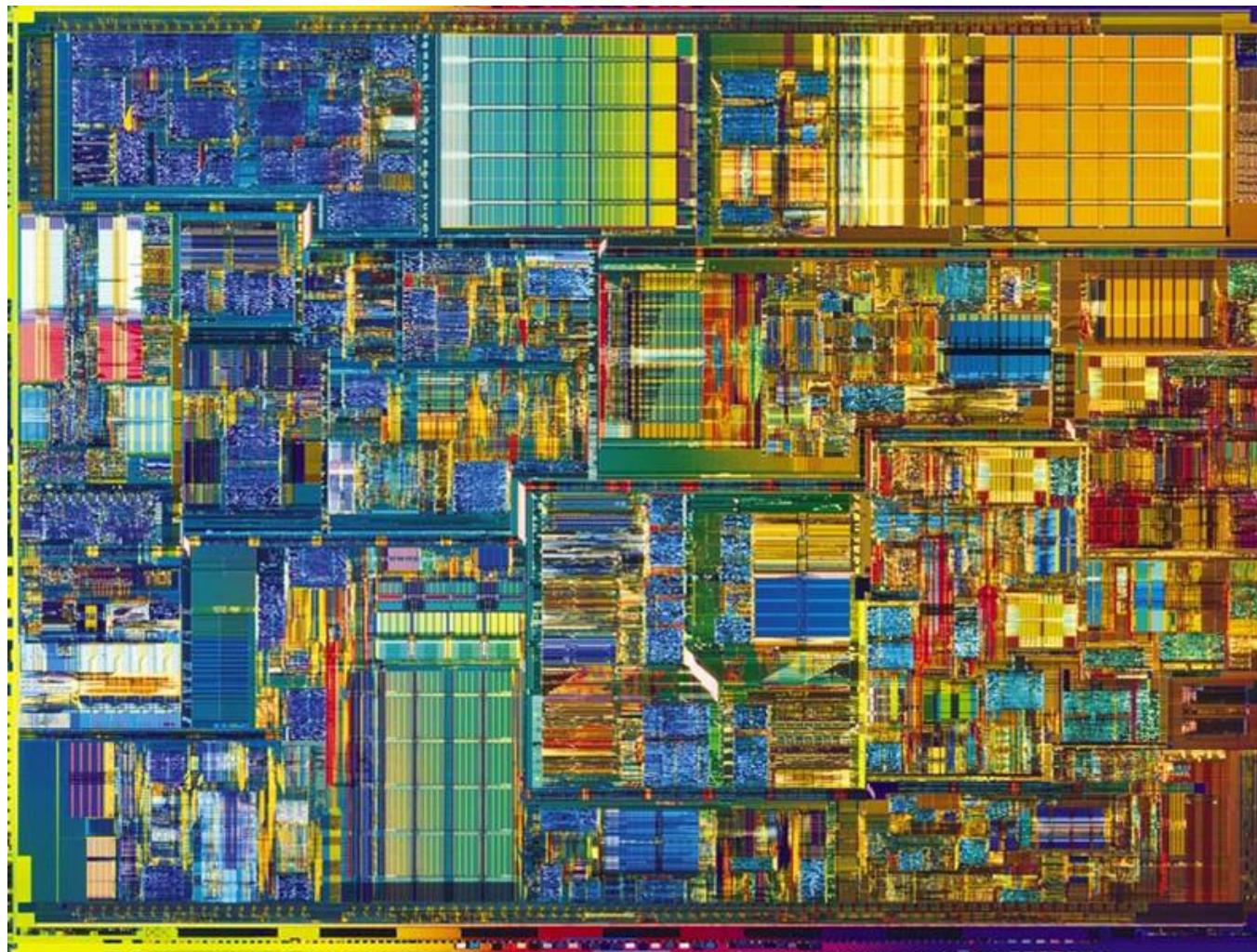
IBM 360: Po 47 letech... Mikroprocesory řady z11



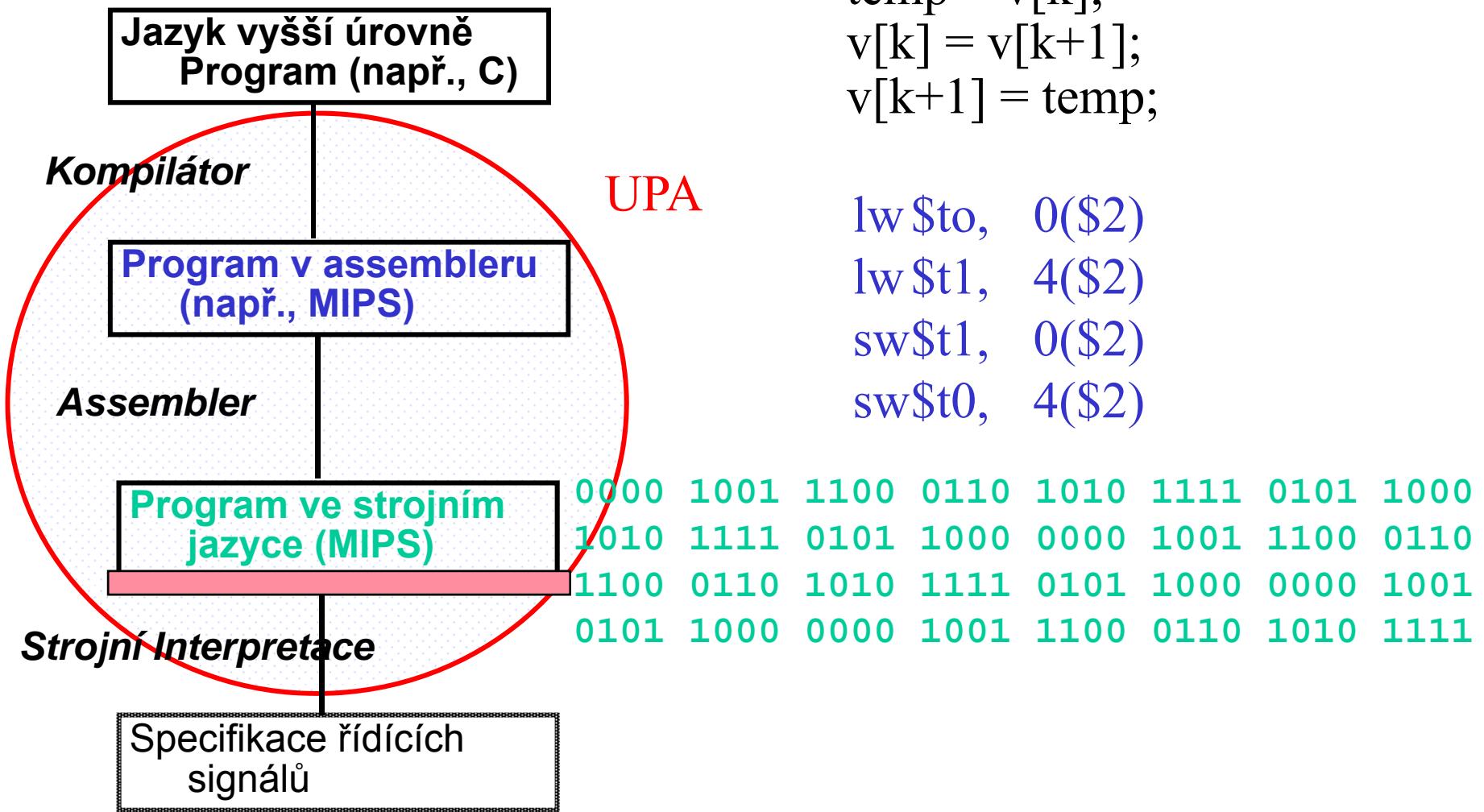
[IBM, HotChips, 2010]

- 5.2 GHz v IBM 45nm PD-SOI CMOS technologii
- 1.4 miliardy tranzistorů na 512 mm^2
- 64-bitové virtuální adresování
 - originalně S/360 měl 24-bitovou adresu, S/370 měl rozšíření na 31-bitů
- Čtyřjádrový návrh
- Three-issue out-of-order superscalar pipeline
- Out-of-order přístupy do paměti
- Redundantní datové cesty
 - every instruction performed in two parallel datapaths and results compared
- 64KB L1 I-cache, 128KB L1 D-cache na čipu
- 1.5MB privátní L2 unifikovaná cache pro každé jádro na čipu
- Na čipu 24MB eDRAM L3 cache
- Scales to 96-core multiprocessor with 768MB of shared L4 eDRAM

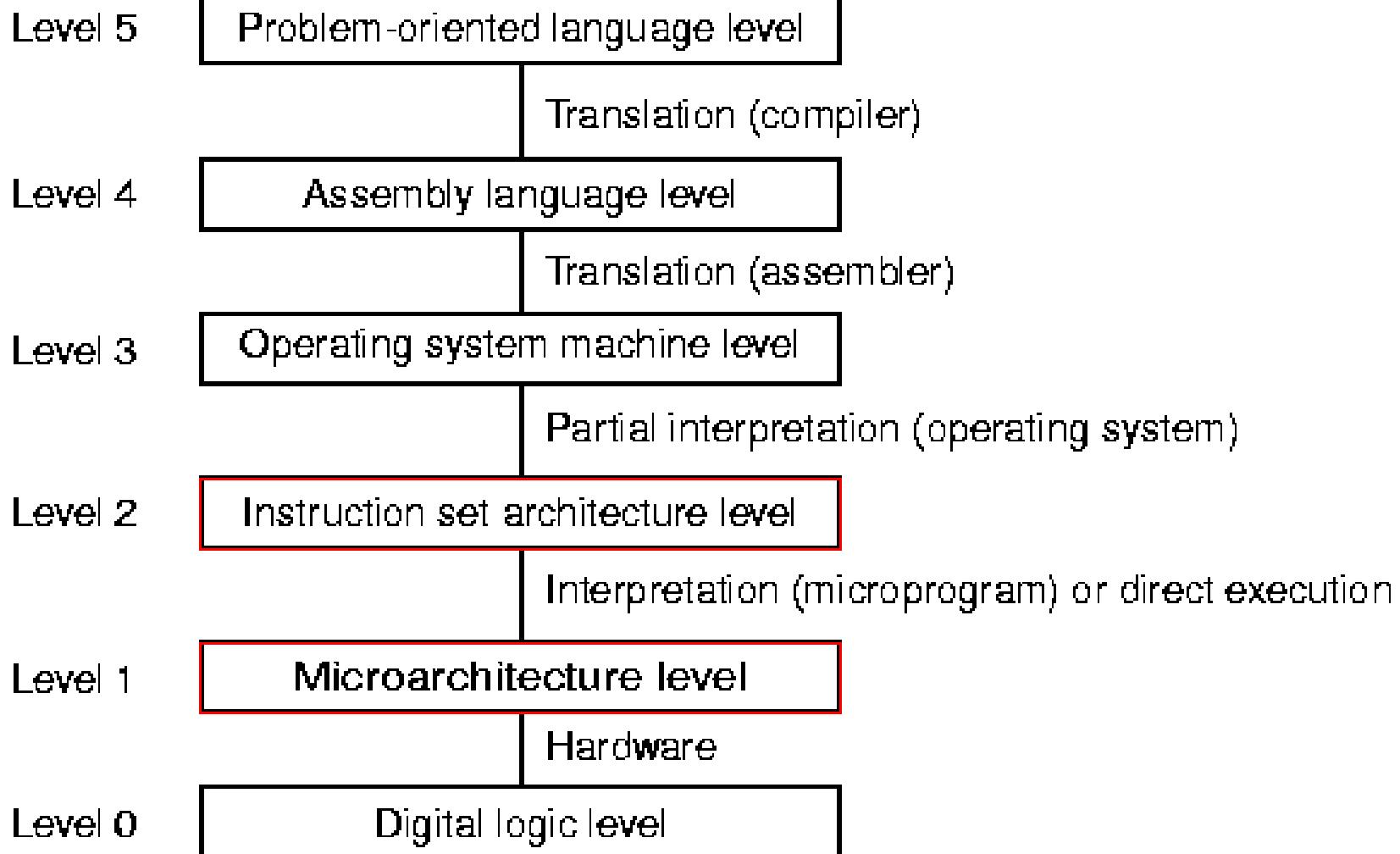
„Nedávná minulost“ u firmy Intel – Pentium 4



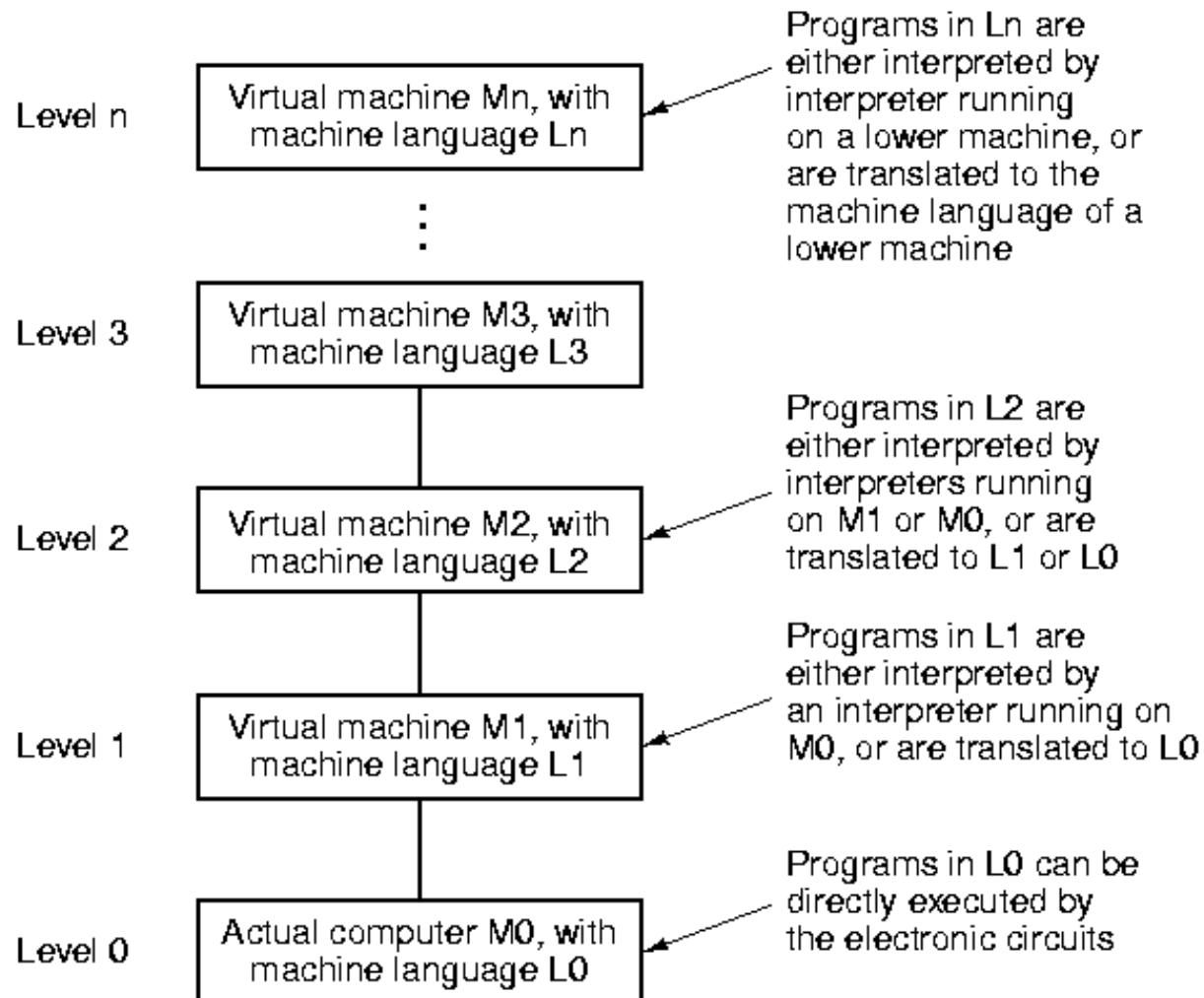
Úrovně reprezentace



Vrstvy počítačového systému



Víceúrovňový stroj



Hierarchie výstavby počítačového systému

1. Samotný hardware
2. Mikroprogramování
- 3. Hardware / software interface**
 - Jednoduchá ISA
 - CISC
 - RISC
 -
4. Operační systém
5. Kompilátory

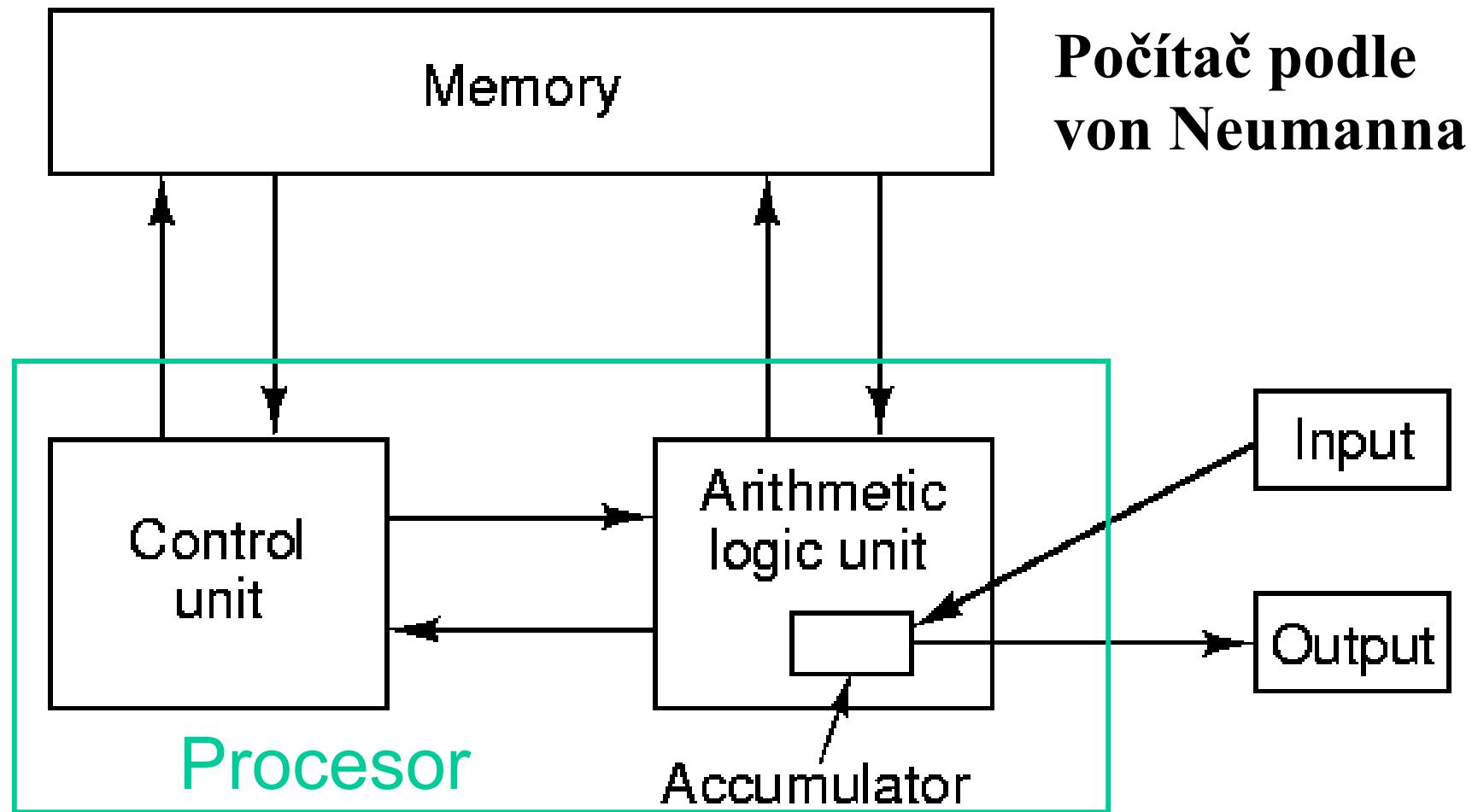
Principy návrhu

CISC versus RISC

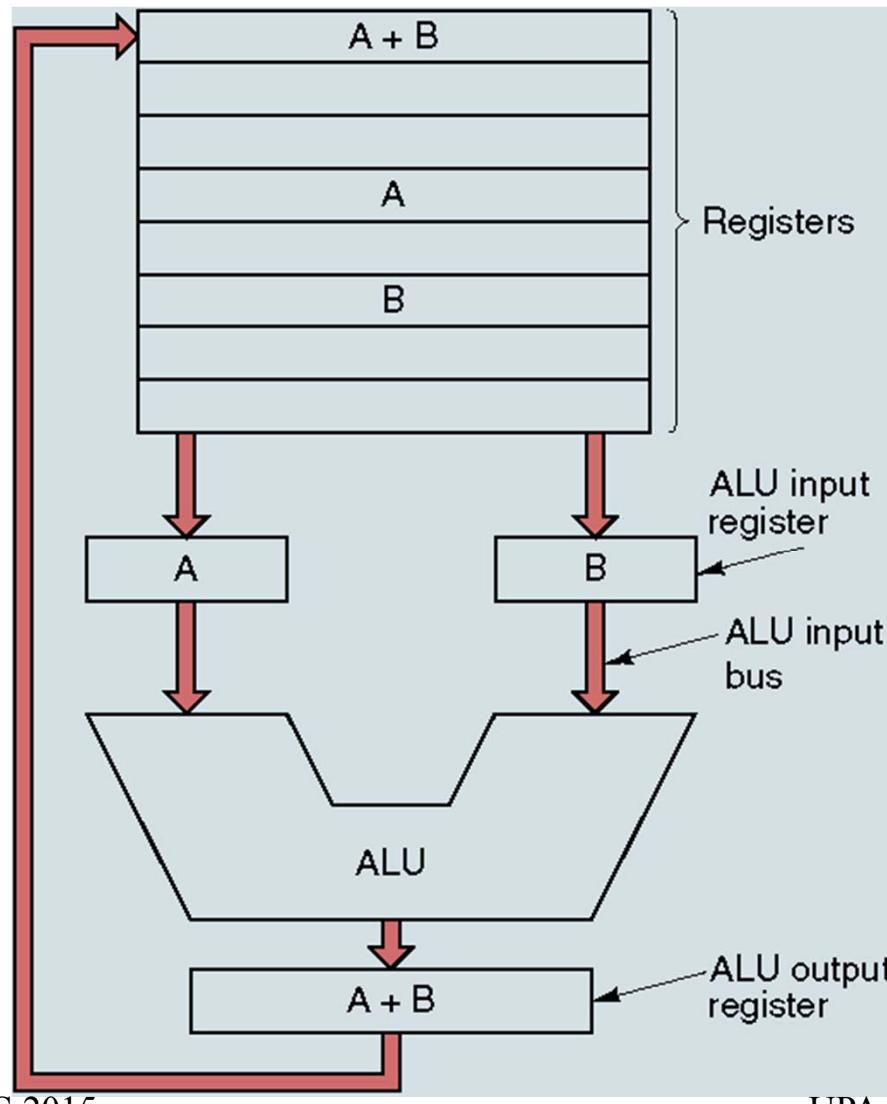
RISC:

- Instrukce jsou přímo prováděny hardwarem
- Maximální průchodnost instrukcí (ILP)
- Jednoduché instrukce (snadné dekódování)
- Přístup do paměti jen instrukcemi load/store
- Velké množství registrů
- Pipelining

Organizace počítače



Datové cesty – současná „klasika“



↔ Memory ↔ I/O

- Registry a ALU
- “Pomocné registry”
 - jen v hardware
 - neobjevují se v programu
- „Jádro procesoru“
 - určuje rychlosť
 - určuje výkon

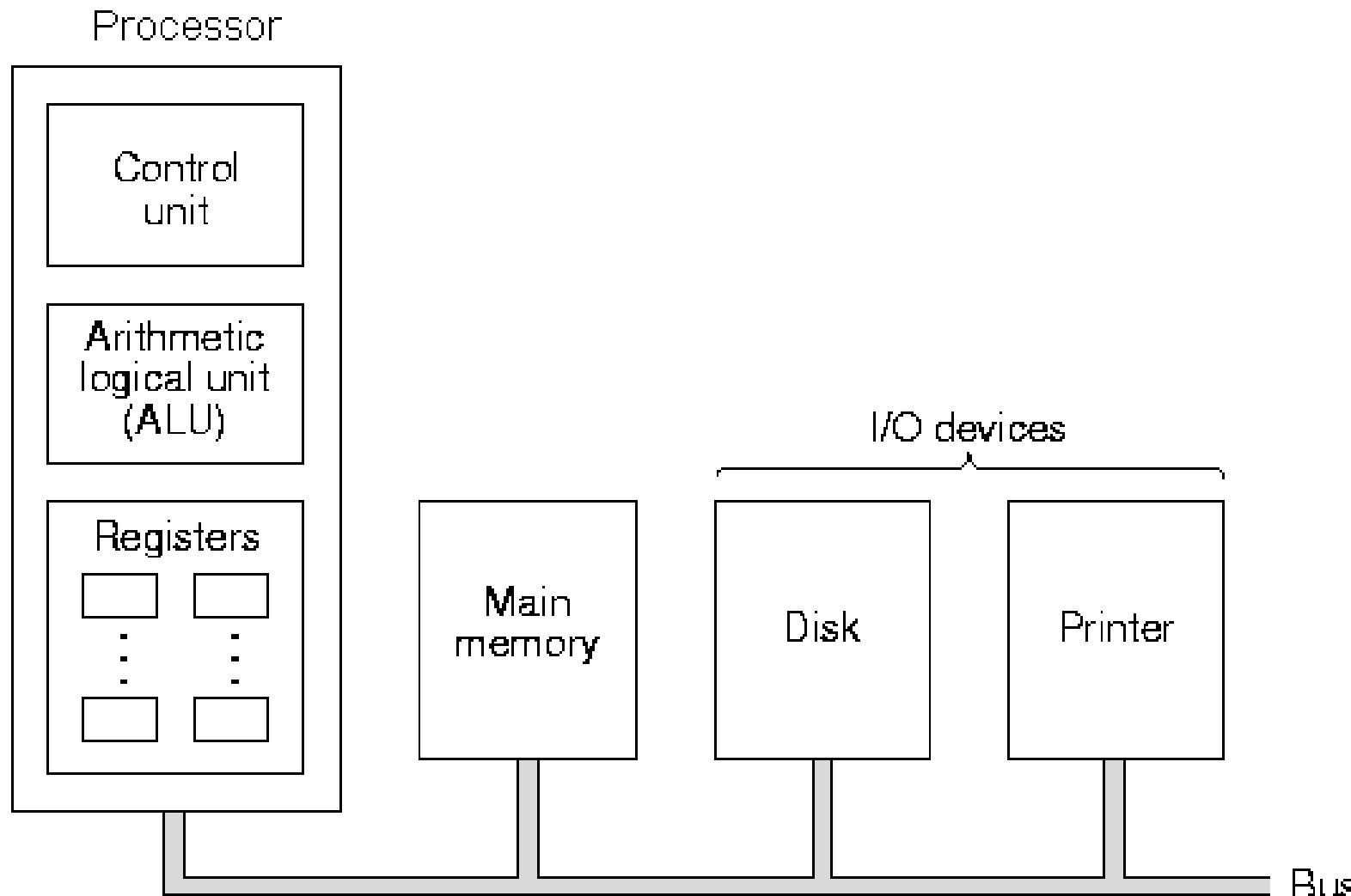
Cyklus zpracování instrukce

Fáze provádění instrukce:

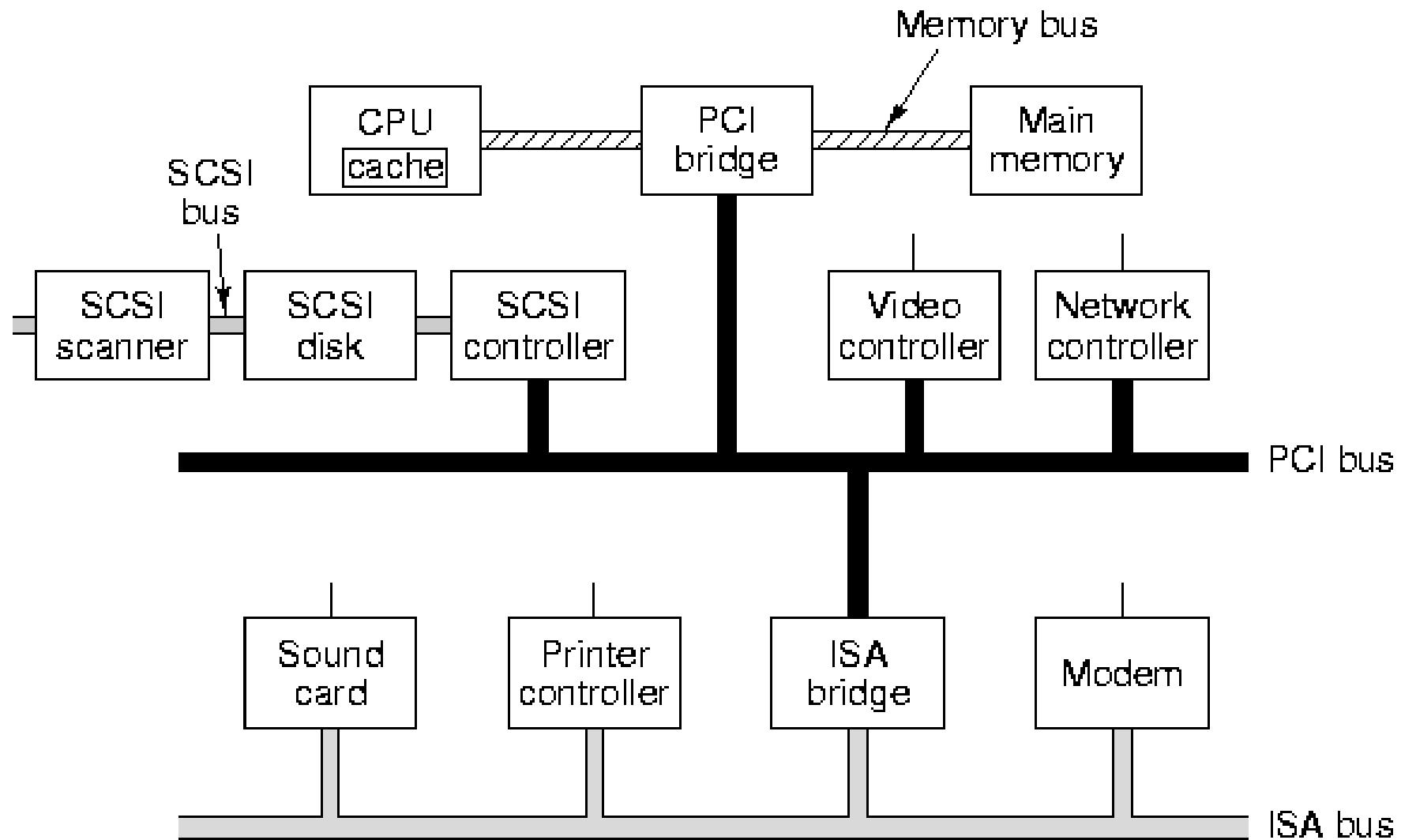
„fetch-decode-execute“

1. Přečtení instrukce z paměti podle adresy v PC do IR
2. Aktualizace PC, aby ukazoval na další instrukci
3. Určení typu instrukce (dekódování) :
 registr/registr nebo registr/paměť ...
4. Výpočet adres operandů (adresní režimy)
5. Načtení operandů do pomocných registrů
6. Provedení vlastní operace prováděcí jednotkou
7. Zápis výsledku do registru(ů), popř. do paměti
8. Pokračování v bodu 1. – zpracování další instrukce

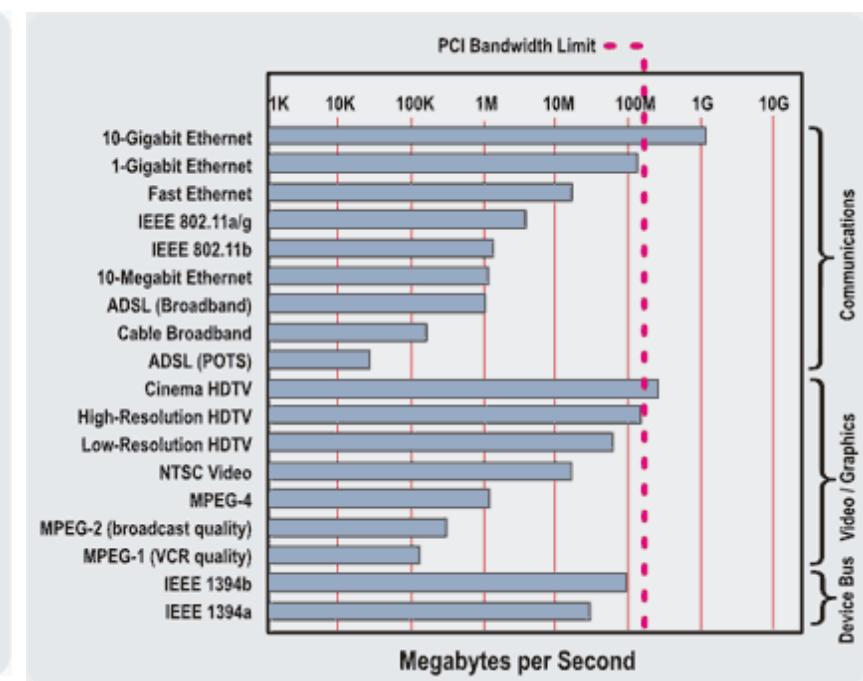
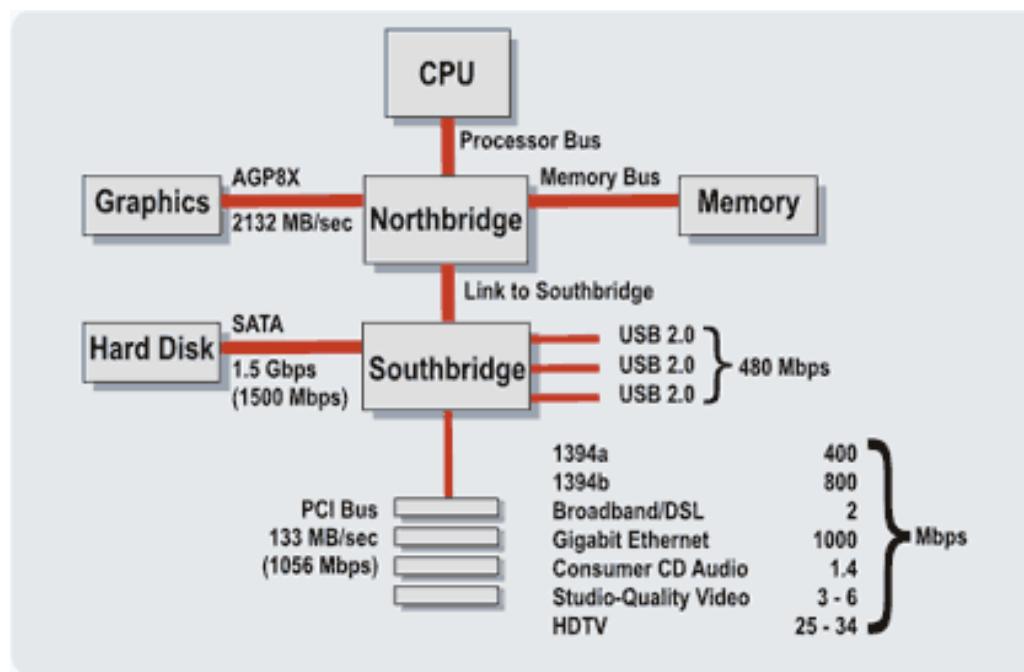
Počítač se sběrnicovou koncepcí



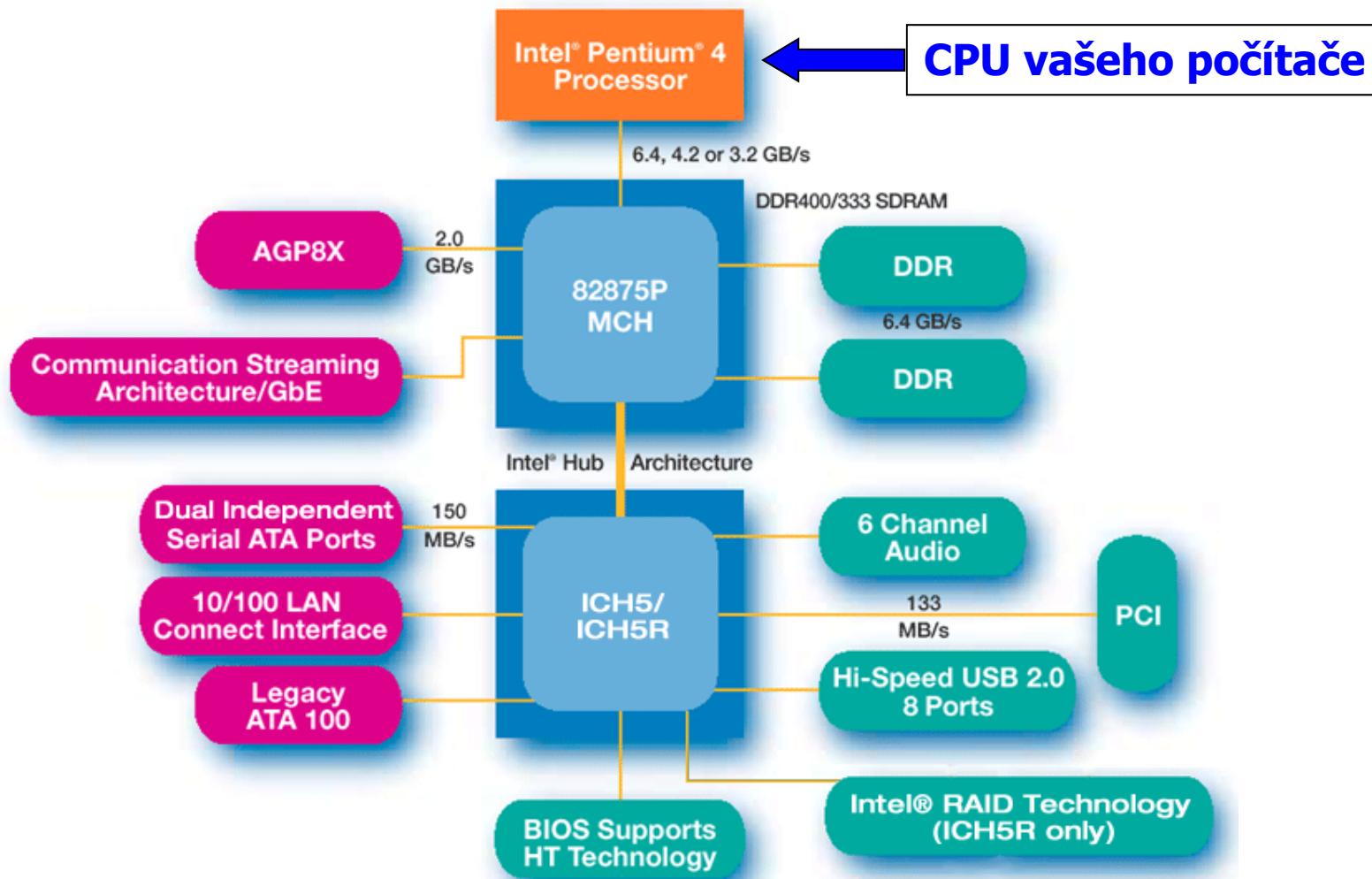
Anatomie PC (kolem r. 1996)



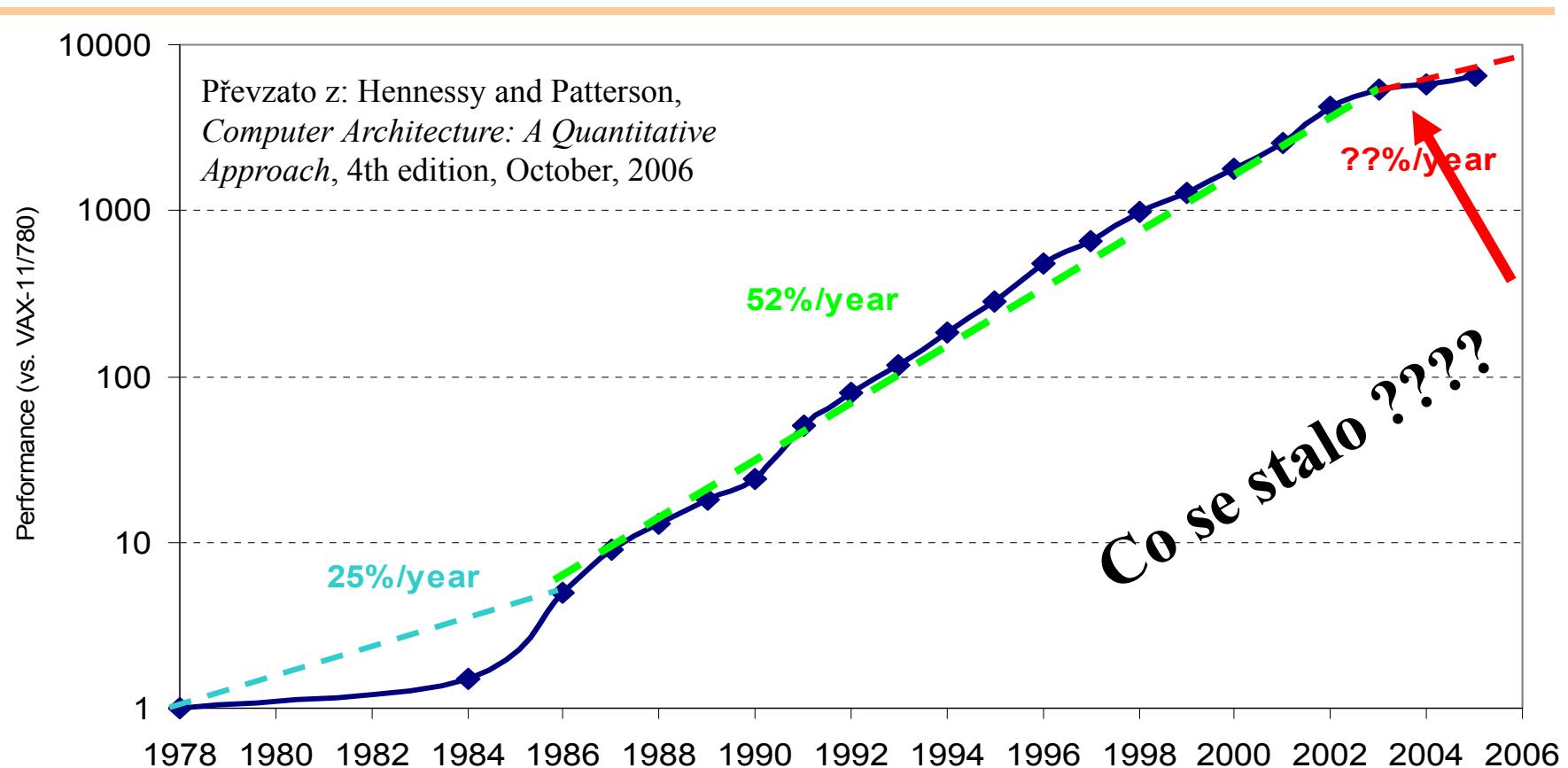
Anatomie PC (kolem r. 2006)



Anatomie PC (kolem r. 2006)



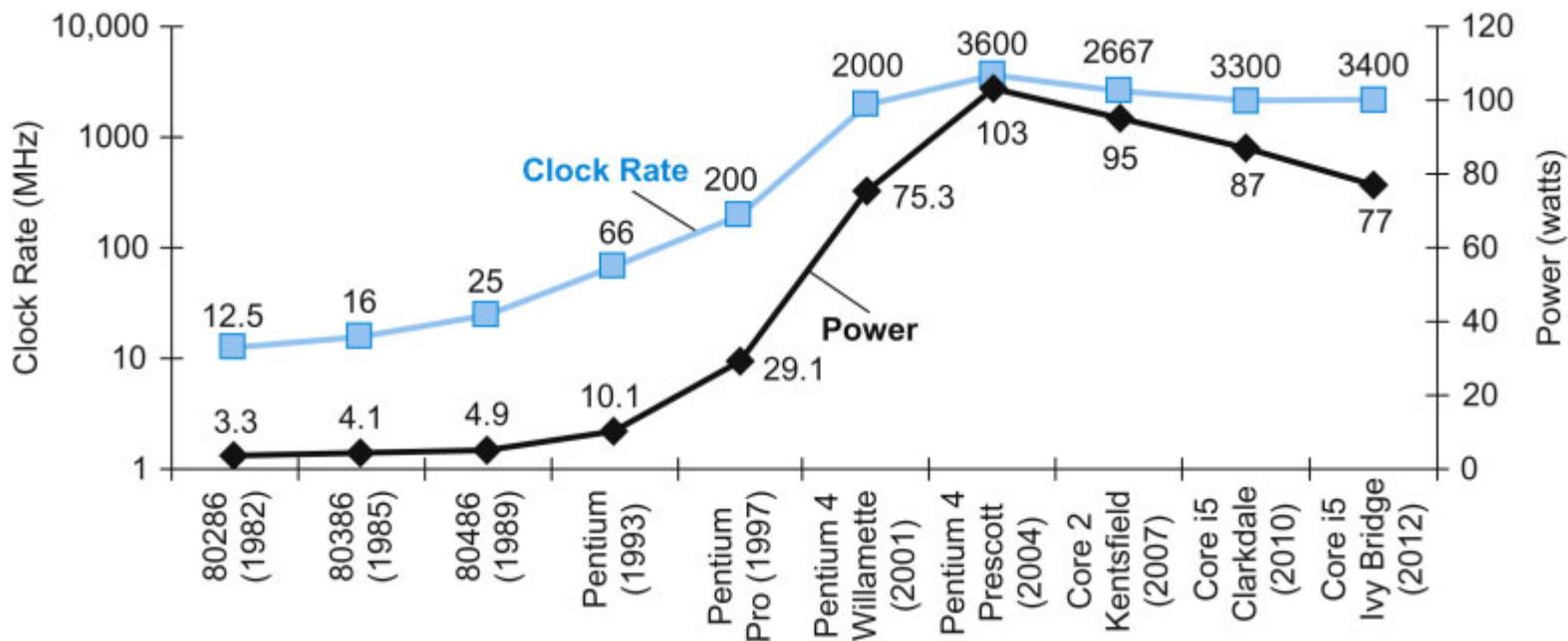
Výkon jednoprocesorového počítače



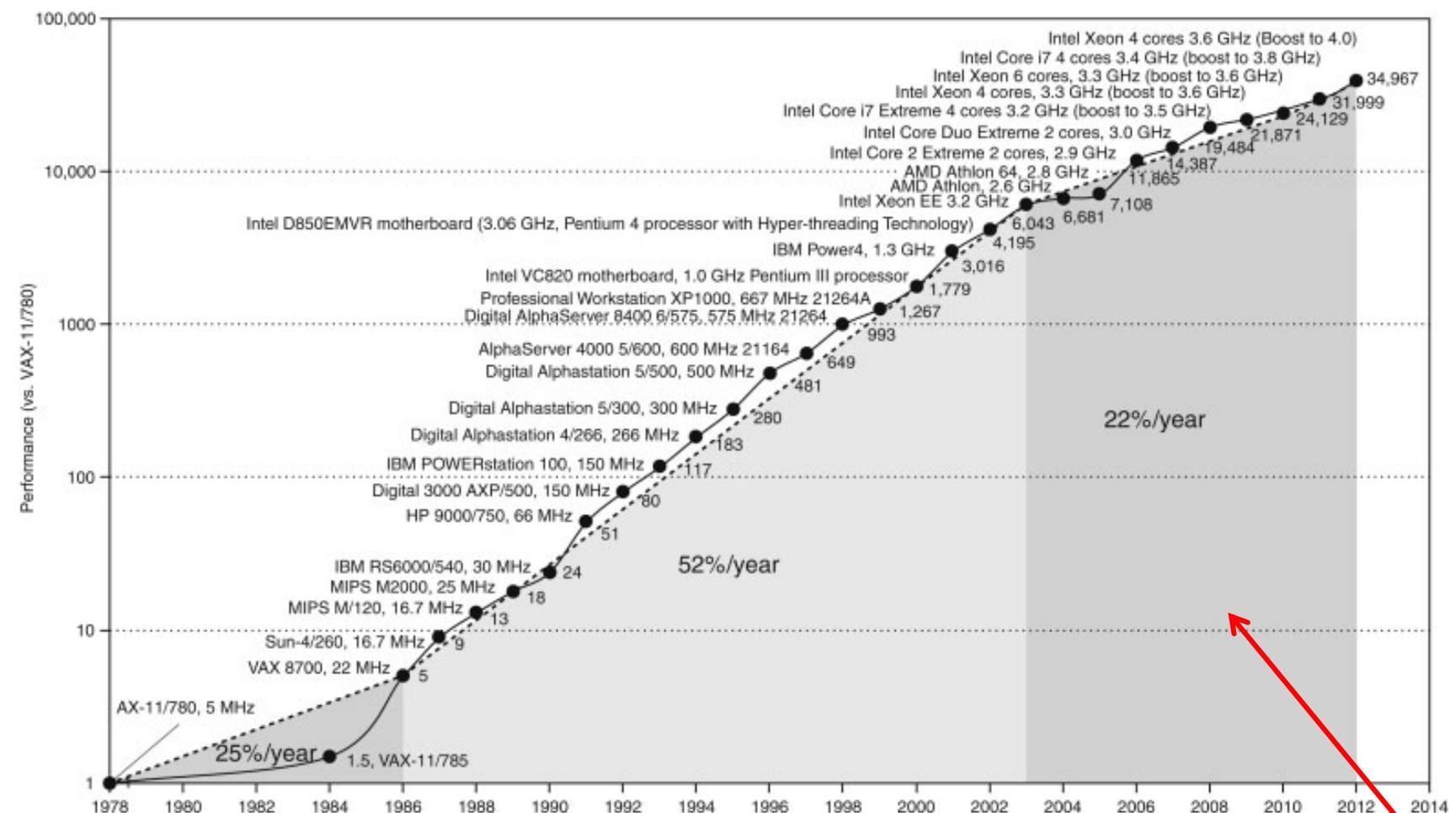
- VAX : 25%/rok od r.1978 do r. 1986
- RISC + x86: 52%/rok od r.1986 do r. 2002
- RISC + x86: ??%/rok od r. 2002 do současnosti

Hodinová frekvence a ztrátový výkon

Procesory firmy Intel:



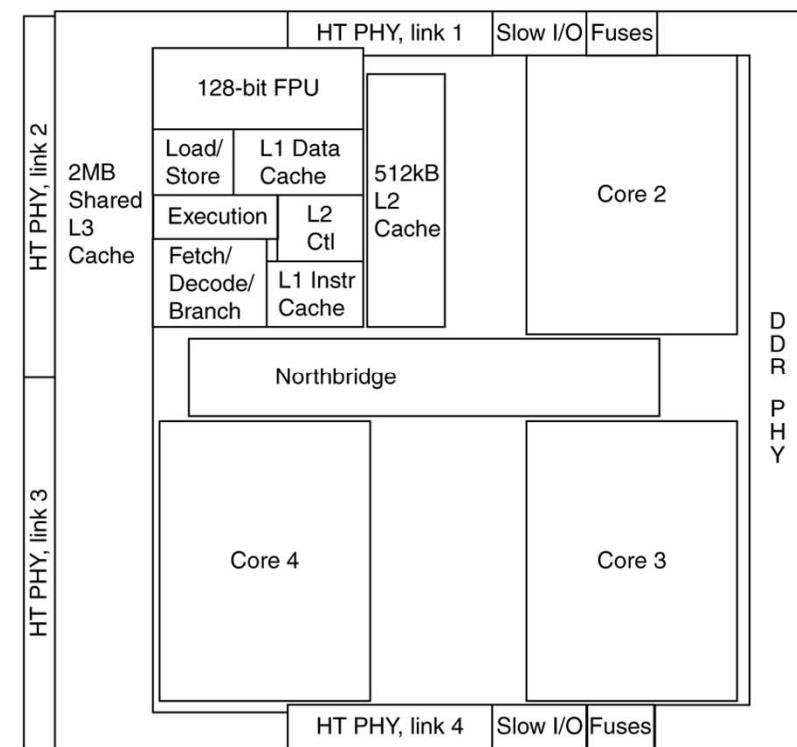
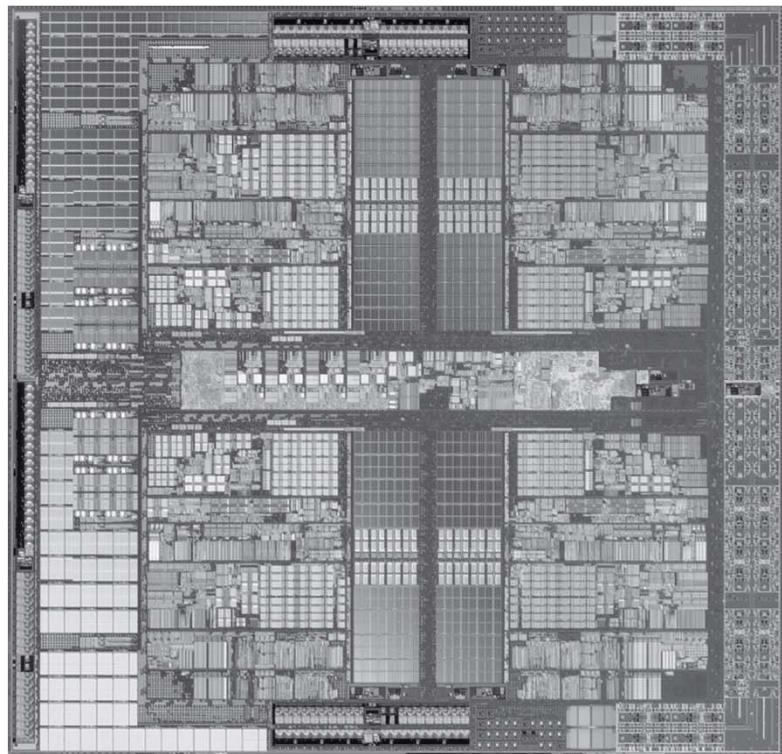
Nárůst výkonu jednoprocessorového počítače



Omezení způsobené ztrátovým výkonem, úrovni parallelizmu a latencí paměti

Vícejádrové procesory

- AMD Barcelona: 4 procesorová jádra (2007)



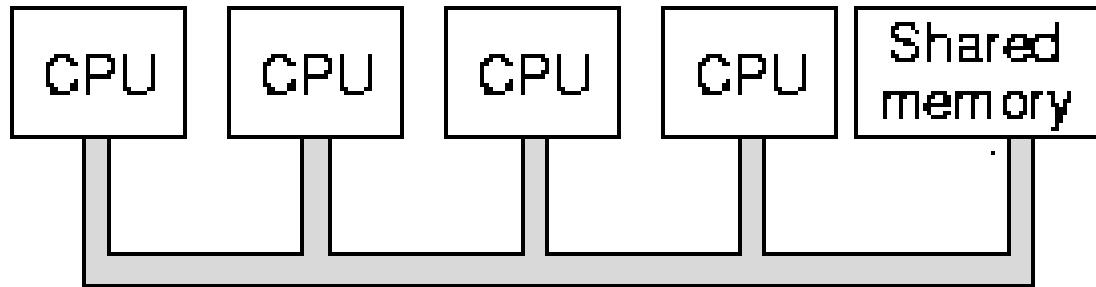
Zvyklosti v počítačové architektuře

- Dříve: Napájení je neomezené, drahý tranzistor
- Nyní: “**Power wall**” výkon je „drahý“, tranzistory nestojí téměř nic
(Na čip lze umístit víc, než je možno „uchladit“)
- Dříve : Dostatečný nárůst paralelismu instrukční úrovně vlivem komplikátorů a inovací (pipelining, superskalární systémy, out-of-order, spekulace, VLIW, ...)
- Nyní : “**ILP wall**” zákon klesající výtěžnosti investice HW pro ILP
- Dříve : Násobení je pomalé, přístup do paměti je rychlý
- Nyní : “**Memory wall**” pomalá paměť, rychlé násobení
(200 cyklů hodin pro přístup do DRAM, 4 hodinové cykly pro násobení)
- Dříve : Výkon jednoprocесоровého systému 2x za 1.5 roku
- Nyní : **Power Wall + ILP Wall + Memory Wall = Brick Wall**
 - Výkon jednoprocесоровého systému nyní 2x za 5(?) let

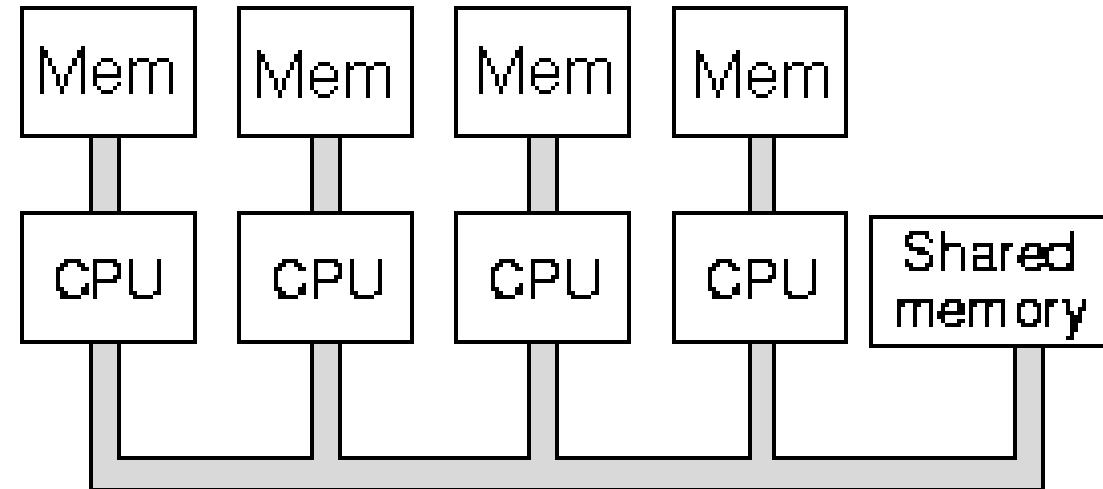
⇒ Viz změny v návrhu čipu: násobná “jádra”
(2x procesorů na čip za ~ 2 roky)
- Více jednodušších procesorů dává efektivněji větší výkon

ILP ... Instruction Level Parallelism

Multiprocesory



Bus



Bus

IBM ASCI White: 8K procesorů, 13 Tflops

Třídy počítačů

- Osobní počítače
 - Navrženy tak, aby poskytovaly dobrý výkon pro jednoho uživatele za nízkou cenu užívajícího obvykle tzv. 3. party software. Zahrnují grafický displej, klávesnici, myš atd.
- Servery
 - Využívány pro zpracování programů více uživatelů současně, typicky přístupné pouze přes síť. Větší důraz je kladen na spolehlivost a (často), bezpečnost
- Superpočítače
 - Třída serverů s vysokou cenou, ale také vysokým výkonem, se stovkami až tisíci procesorů, **terabajty** paměti a **petabajty** úložného prostoru, které se používají pro „high-end“ vědecké a inženýrské aplikace
- „Embedded“ počítače (procesory)
 - Počítač umístěný v „jiném“ zařízení, používaný obvykle pro spouštění jedné předem určené aplikace

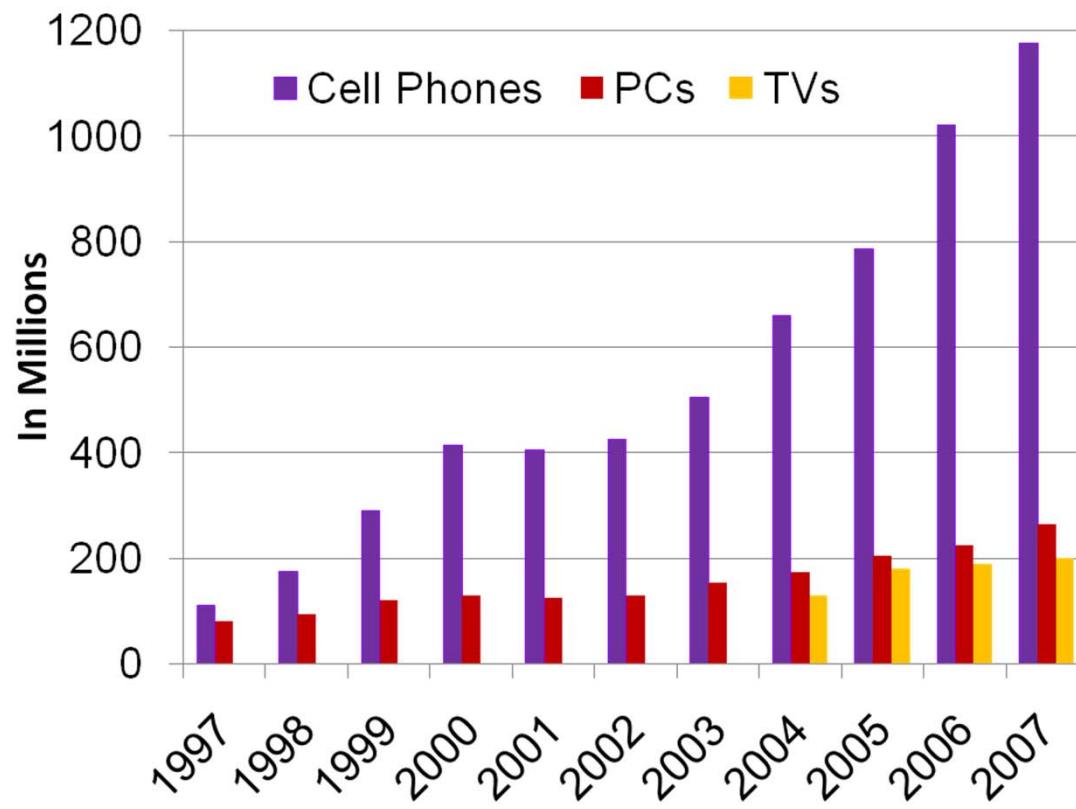
Opakování: Některé základní definice

- Kilobyte – 2^{10} nebo 1024 bytů
- Megabyte – 2^{20} nebo 1048576 bytů
- Gigabyte – 2^{30} nebo 1073741824 bytů
- Terabyte – 2^{40} nebo 1099511627776 bytů
- Petabyte – 2^{50} nebo 1024 terabytů
- Exabyte – 2^{60} nebo 1024 petabytů

Pozn: Rozlišujte KB a kB, MB a mB, atd. !

Nárůst prodeje mobilů („embedded“)

nárůst aplikací třídy „embedded“ >> nárůst PC



- ❑ Kde lze nalézt tzv. „embedded“ procesory?

Charakteristika „embedded“ procesorů

Největší třída počítačů zahrnující nejširší škálu aplikací a výkonů

- Často mají minimální požadavky na výkon. Příklad?
- Často mají přísné omezení na cenu. Příklad?
- Často mají přísné omezení na spotřebu energie. Příklad?
- Často mají nízkou toleranci k selhání. Příklad?

Závěr - opakování

- < 13 týdnů ke studiu základních koncepcí v CS & CE
 - *Principy abstrakce*, použité ke stavbě systému po vrstvách
 - „*Pružná*“ data: program určuje interpretaci obsahu paměti
 - Koncepce *programu v paměti*: instrukce jsou také data
 - Princip *lokality*, využíván v paměťové hierarchii
 - Větší výkon využitím *paralelního zpracování* (pipeline)
 - *Kompilace versus interpretace*
 - Principy a problémy *měření výkonu*