Princípy a základné koncepcie počítačových systémov

(história, základné princípy a technologické súvislosti)

Obsah prednášky

- cieľ predmetu, anotácia a štruktúra
- doporučená literatúra
- číslicový počítač (zadefinovať)
- história vývoja počítačov
- Moorov zákon, CMOS technológia, prognóza vývoja, údaje z Technology roadmap
- základný princíp CMOS logiky
- všeobecná bloková schéma procesora
    - harvardská vs von Neumanova architektúra
    - zbernice (adresová, dátová, riadiaca)
- pojmy paralelizmus, zreťazenie, súbežnosť
- mikroprocesor a jeho využitie v počítačových systémoch
- Amdahlov zákon
- meranie výkonnosti počítačových systémov

Zdroje:

B. Parhami, *Computer Architecture: From Microprocessors to Supercomputers,* Oxford Univ. Press, New York, 2005. (ISBN 0-19-515455-X, 556+xix pages, 300 figures, 491 end-of-chapter problems)
https://www.ece.ucsb.edu/~parhami/text_comp_arch.htm

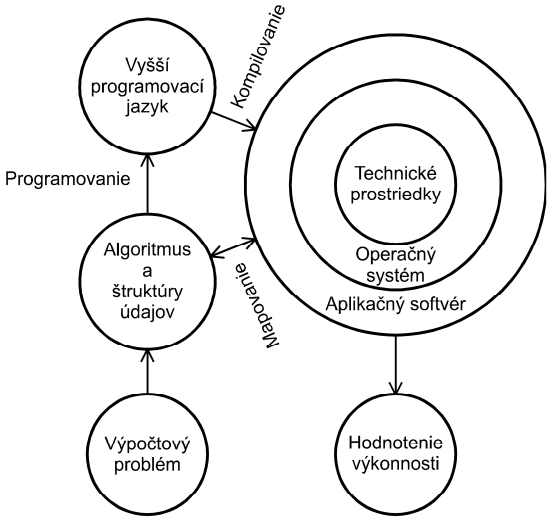
R. Lórencz, J. Hlaváč, T. Zahradnický: *Architektúra počítačových systému*, FIT ČVUT, Praha, 2012. (podklady k prednáškam)

N. Ádám: *Architektúry počítačových systémov*, FEI TUKE, 2017 (podklady k prednáškam, https://moodle.fei.tuke.sk)

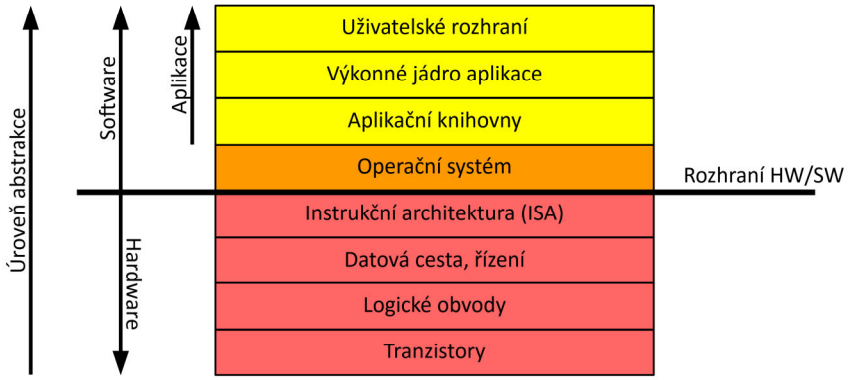
Číslicový počítač (ČP) je zložitý **univerzálny číslicový systém** (automat) určený na **samočinné vykonávanie** postupnosti operácií (výpočtov) nad údajmi zobrazenými **číslicovým kódom**, na základe vopred pripraveného a v **pamäti uloženého programu** (algoritmu).

Systém vytvorený na báze jedného alebo viacerých ČP resp. ich komponentov sa nazýva **počítačový systém (PS).**

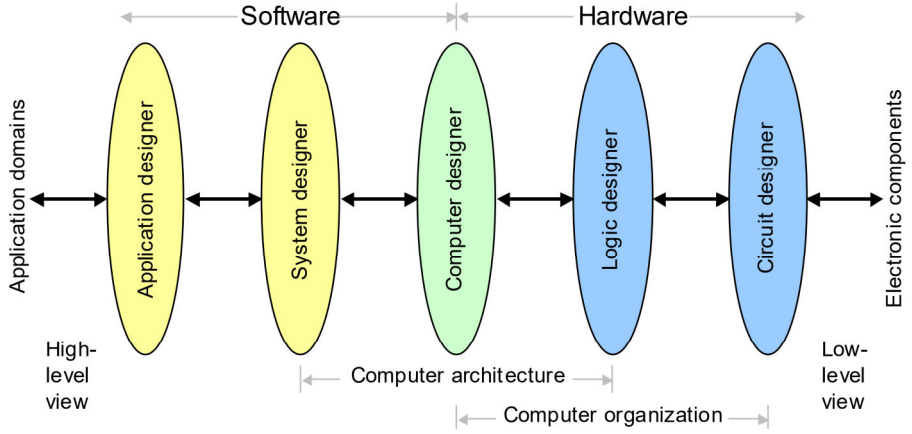
Objekt záujmu v predmete APS – technické prostriedky ČP



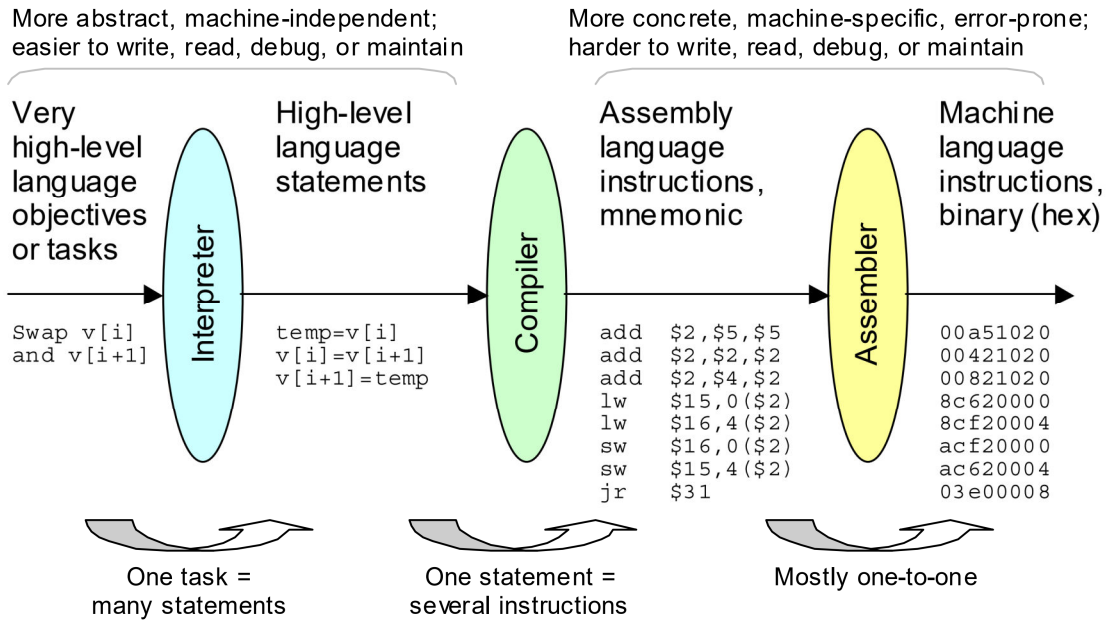
Vrstvy abstrakcie počítača



Členenie vrstiev z pohľadu počítačového inžinierstva



Abstrakcia z pohľadu úrovne programovania



História vývoja počítačov

Vývojové medzníky

- od vytvoření 1. univerzálního elektronického počítače uplynulo více než 50 let
- dnešní PC za 1000 $ jsou výkonnější než počítač z r. 1980 za 1 mil. $
- HW průlom: VLSI technologie a příchod mikroprocesorů (70. léta)
- SW průlom: univerzální na výrobci nezávislé OS (UNIX) a přechod od programování v SOJ k programování ve vyšších jazycích
- nástup RISC (Reduced Instruction Set Computer), důsledek:
  - paralelizmus na úrovni zpracování instrukcí – ILP (Instruction Level Parallelism), tj. proudové zpracování instrukcí, super-skalární architektury atd.
  - používání vnitřních skrytých pamětí (cache)
- průlom v navrhování: vývoj kvantitativního přístupu k návrhu a analýze počítačů, který využívá empirická pozorování, experimentování a simulace

Chronológia

- 60. léta: dominantní velké sálové počítače s aplikacemi jako:
  - zpracování dat ve finanční sféře
  - rozsáhlé vědeckotechnické výpočty
- 70. léta: mikropočítače pro aplikace ve vědeckých laboratořích
- 80. léta: příchod stolních počítačů založených na mikroprocesorech (osobní počítače a pracovní stanice)
  dále se objevují servery a lokální sítě pro větší úlohy s větší pamětí a výkonem
- 90. léta: Internet a WWW technologie
- 00. léta: nelze dále rozumně zvyšovat výkon jednojádrových procesorů $\rightarrow$ prosazují se vícejádrové architektury
- současnost: rozdělení počítačového trhu na 4 oblasti charakterizované rozdílným použitím, požadavky a počítačovou technologií:
  1. osobní, stolní a přenosné počítače
  2. servery a výkonné paralelní počítače a superpočítače
  3. vestavěné a řídící počítače v jednoúčelových zařízeních
  4. nově vyčleněná oblast: mobilní zařízení (tablety, smartphony)
- běžně se používá virtualizace hardwaru
- výkonné grafické procesory použitelné pro obecné výpočty (GPGPU)
- lze si koupit výpočetní výkon nezávisle na hardwaru (cloud computing)
- rychlé a téměř všude dostupné internetové připojení umožňuje přenášet „tradiční“ aplikace do online podoby (SaaS)

Generácie počítačov z pohľadu použitej technológie

Generation (begun)	Processor technology	Memory innovations	I/O devices introduced	Dominant look & fell
0 (1600s)	(Electro-) mechanical	Wheel, card	Lever, dial, punched card	Factory equipment
1 (1950s)	Vacuum tube	Magnetic drum	Paper tape, magnetic tape	Hall-size cabinet
2 (1960s)	Transistor	Magnetic core	Drum, printer, text terminal	Room-size mainframe
3 (1970s)	SSI/MSI	RAM/ROM chip	Disk, keyboard, video monitor	Desk-size mini
4 (1980s)	LSI/VLSI	SRAM/DRAM	Network, CD, mouse,sound	Desktop/ laptop micro
5 (1990s)	ULSI/GSI/ WSI, SOC	SDRAM, flash	Sensor/actuator, point/click	Invisible, embedded

Pekne spracovaná historia vývoja počítačov je napr. v skriptách:
https://www.researchgate.net/profile/Jaroslav-Majernik/publication/256442692_Zaklady_informatiky/links/00b7d52299d68f2d7e000000/Zaklady-informatiky.pdf

Najznámejšie omyly v predpovedi vývoja počítačov

"DOS addresses only 1 MB of RAM because we cannot imagine any applications needing more." Microsoft, 1980

"640K ought to be enough for anybody." Bill Gates, 1981

"Computers in the future may weigh no more than 1.5 tons." *Popular Mechanics*

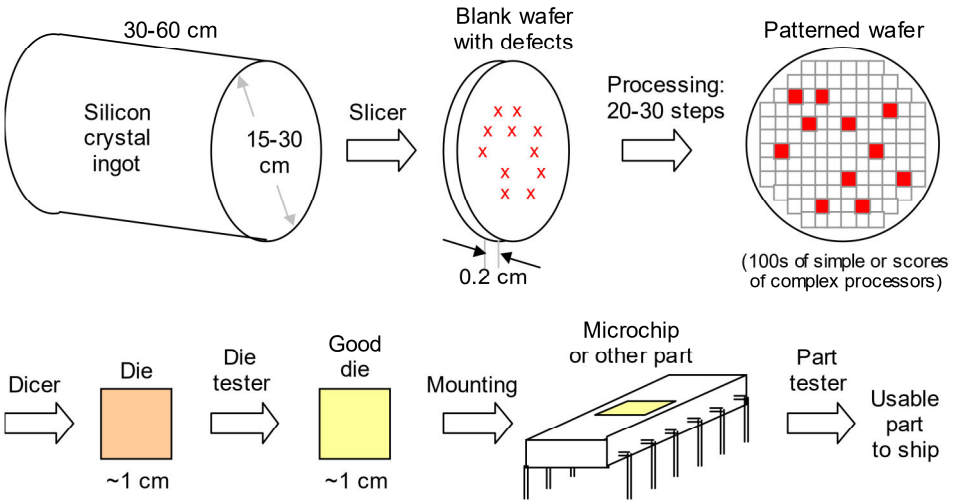
"I think there is a world market for maybe five computers." Thomas Watson, IBM Chairman, 1943

"There is no reason anyone would want a computer in their home." Ken Olsen, DEC founder, 1977

"The 32-bit machine would be an overkill for a personal computer." Sol Libes, *ByteLines*

Technológia výroby -> dominantný vplyv má pokrok vo výrobe polovodičov a integrovaných obvodov (**Moorov zákon**)

Kroky výrobného procesu IO (vrátane CPU a pamätí)



V súčasnosti v **10 nm** technológii na **1 mm^2** je možné umiestniť cca **10^8** tranzistorov

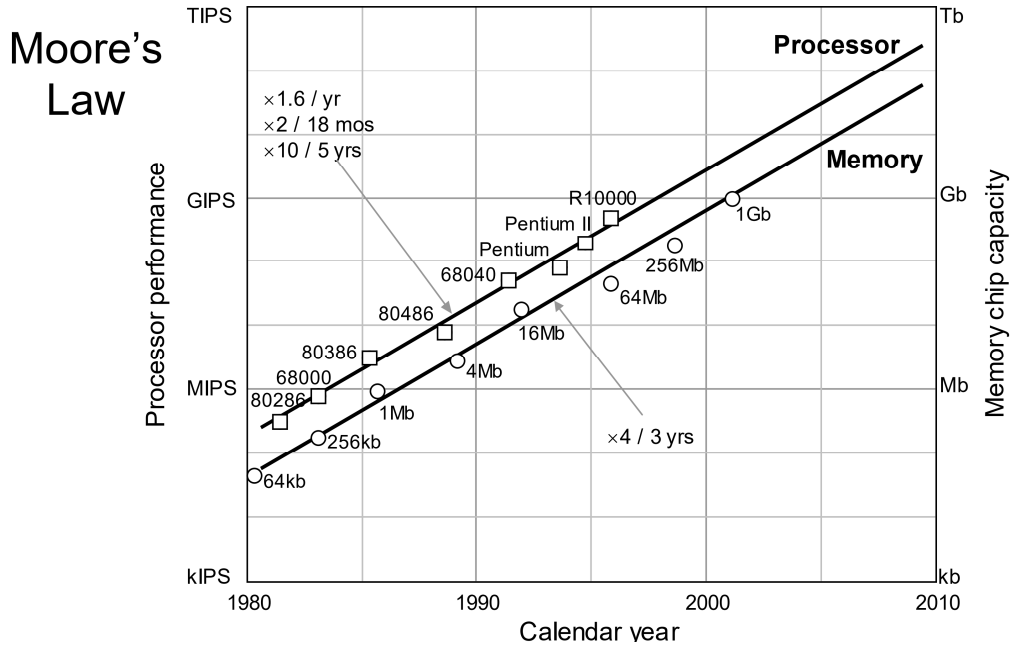
On September 12, 2023, Apple announced the iPhone 15 Pro would feature a 3 nm chip, the A17 Pro.[65]

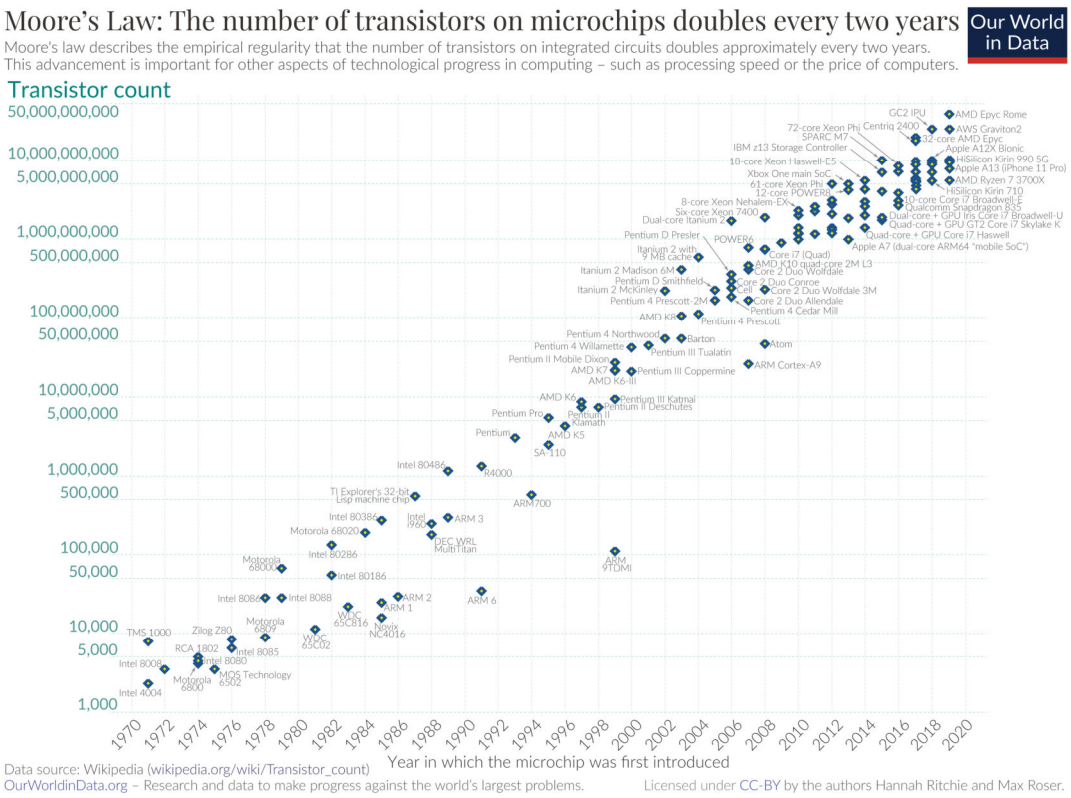
3 nm process nodes [edit]

	Samsung[43][66][67][68]			TSMC[69]					Intel[5]
Process name	3GAE	3GAP	3GAP+	N3	N3E	N3S	N3P	N3X	3
Transistor type	MBCFET			FinFET					
Transistor density (MTr/mm²)	150[67]	195[67]	Unknown	220[46] (theoretical) 183 (A17 Pro)[70]	215.6[71]	Unknown	224.2[72]	224.2[72]	Unknown
SRAM bit-cell size (μm²)	Unknown	Unknown	Unknown	0.0199[59]	0.021[59]	Unknown	Unknown	Unknown	Unknown
Transistor gate pitch (nm)	40	Unknown	Unknown	45[59]	48[71]	Unknown	Unknown	Unknown	Unknown
Interconnect pitch (nm)	32	Unknown	Unknown	Unknown	23[59]	Unknown	Unknown	Unknown	Unknown
Release status	2022 risk production[43] 2022 production[1] 2022 shipping[2]	2024 production	2025 production	2021 risk production 2022 H2 volume production[69][3] 2023 H1 shipping for revenue[73]	2023 H2 production[69]	2024 H1 production[46]	2024 H2 production[62]	2025 production[62]	2023 H2 product manufacturing[5] 2024 fabbing of Xeons[74]

Zdroj: https://en.wikipedia.org/wiki/3_nm_process

Trendy vo vývoji výkonnosti procesorov a kapacity DRAM





Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Zdroj: https://en.wikipedia.org/wiki/Transistor_count

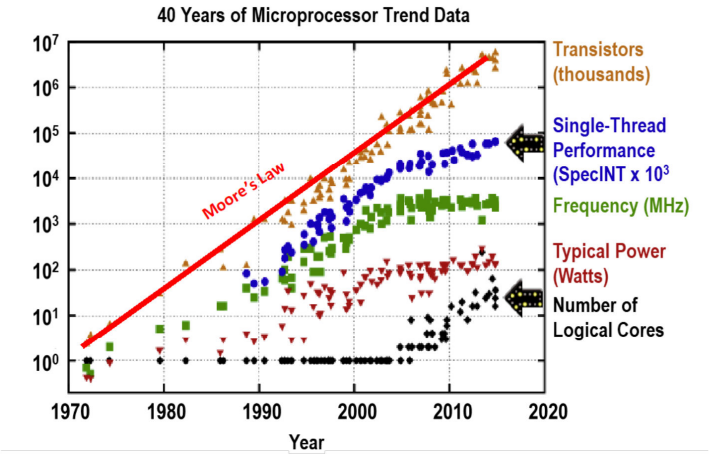


Figure ES39 *Adoption of multicore architecture allowed moderate performance increase in CPU design within power limits*

Prognózy vývoja CMOS technológie (https://irds.ieee.org/editions/2022)

Table ES3 Overall Roadmap Technology Characteristics

2022 IRDS ORTC							
YEAR OF PRODUCTION	*2021*	*2022*	*2025*	*2028*	*2031*	*2034*	*2037*
Logic device technology naming note definition [1a]	G51M29	G48M24	G45M20	G42M16	G40M16T2	G38M16T4	G38M16T6
Logic industry "Node Range" Labeling (nm) [2]	"5"	"3"	"2"	"1.5"	"1.0-eq"	"0.7nm-eq"	"0.5nm-eq"
Fine-pitch 3D integration scheme		Stacking	Stacking	Stacking	3DVLSI	3DVLSI	3DVLSI
Platform device for logic [1b]	FinFET	FinFET LGAA	LGAA	LGAA CFET-SRAM	LGAA-3D CFET-SRAM	LGAA-3D CFET-SRAM	LGAA-3D CFET-SRAM
LOGIC CELL AND FUNCTIONAL FABRIC TARGETS							
Digital block area scaling	1.00	1.00	0.74	0.55	0.26	0.13	0.08
LOGIC DEVICE GROUND RULES							
MPU/SoC M0 1/2 Pitch (nm) [3]	15	12	10	8	8	8	8
Gate length (nm) [4]	17	16	14	12	12	12	12
Lateral GAA (nanosheet) Minimum Thickness (nm)		1	3	3	4	4	4
Number of stacked tiers [5]		1	1	1	2	4	6
Number of stacked nanosheets in one device [5]		1	3	3	4	4	4
LOGIC DEVICE Electrical							
Vdd (V) [6]	0.75	0.70	0.65	0.65	0.60	0.60	0.60
DRAM TECHNOLOGY							
DRAM Min half pitch (nm) [7]	17.5	15.5	13	14	11.5	10	10
DRAM cell size (μm^2) [8]	0.00184	0.00165	0.00118	0.00085	0.00062	0.00044	0.00025
DRAM storage node cell capacitor voltage (V) [9]	0.50	0.45	0.45	0.43	0.4	0.4	0.4
NAND Flash							
Flash 2D NAND Flash uncontacted poly 1/2 pitch – F (nm) 2D [10a,b]	15	15	15	15	15	15	15
Product highest density (3D) (commercialized) [11]	1T	1.3T	2.6T	4T	6T	8T	12T
Flash Product Maximum bits/cell (2D_3D) [12]	2_4	4	5	5	6	6	6
Flash 3D NAND Maximum Number of Memory Layers [13]	64-97	128-192	256-384	384-576	576-768	768-1024	1024-1536
Maximum chip size (mm^2) [14]	140	140	140	140	140	140	140

Table ES1 Overall Roadmap System Characteristics

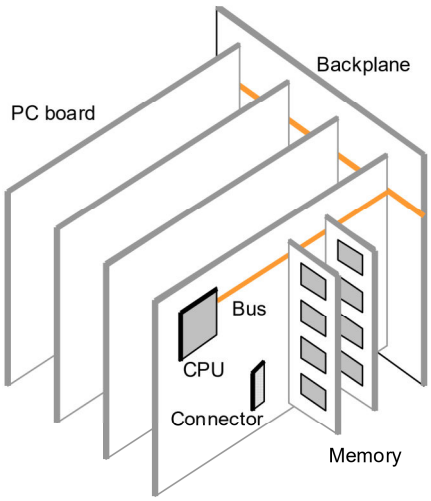
YEAR OF PRODUCTION	2021	2022	2025	2028	2031	2034	2037
Cloud Computing (CC) Latency Sensitive Processors							
Number of Cores per socket (max) [1]	46	64	128	256	384	640	896
Processor Base Frequency (for multiple cores together) [2]	3.20	2.5-3.3	3.0-3.6	3.4-3.7	3.4-3.7	3.4-3.7	3.4-3.7
L1 Data Cache Size (in KB) [3]	38	40	42	42	44	44	44
L1 Instruction Cache Size (in KB) [4]	64	96	128	128	160	160	160
DDR bandwidth (TB/s)	0.2	0.31	0.76	1.02	1.2	1.2	1.2
Number of DDR channels	8	12	12	16	16	16	15
	DDR4	DDR4	DDR5	DDR5	DDR6	DDR6	DDR6
Socket TDP (Watts)	280	300	450	600	600	700	700
SA-OSC Personal Augmentation Table [5] - Focus Drivers Line Items							
# CPU cores	8	8	16	28	32	32	32
# GPU cores	32	32	64	128	256	512	512
Max Freq (GHz)	2.8	2.8	3.0	3.2	3.4	3.4	3.4
5G Maximum Data Rate (Gbps) [6]	5	5	7	10	20	50	70
# Sensors	8	10	12	12	16	16	16
Board Power (mW)	5618	5900	6830	7900	9150	10500	11000
SA IoT Table - Focus Drivers Line Items							
CPUs per device [7]	2	2	4	6	8	8	8
Max CPU Frequency (MHz)	305	310	325	341	360	375	390
Energy Source (B = battery, H = energy harvesting)	B+H	B+H	B+H	B+H	B+H	B+H	B+H
Sensors per device	8	8	12	16	16	16	19
SA CPS Table - Focus Drivers Line Items							
Number of Devices	64	64	128	256	512	512	512
CPUs per Device [7]	8	8	12	12	16	16	16

Table ES4 Systems and Architectures Technology Trends of Latency Sensitive Processers

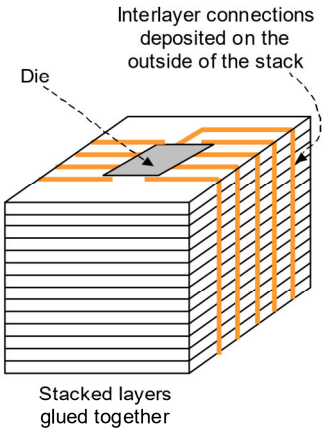
	2019	2022	2024	2026	2028	2030	2032	2034
Number of chiplet per socket	4-8	8	8	8	8-16	8-16	8-16	16-20
Number of Core per chiplet	8	8	8-12	8-16	8-16	16-24	16-32	16-32
Number of Cores per socket (max)	64	64	96	128	256	384	512	640
Processor base frequency (GHz) (for multiple cores together)	2.2-3.0	2.5-3.3	2.8-3.4	3.0-3.5	3.2-3.6	3.4-3.7	3.4-3.7	3.4-3.7
Core total vector length	1024	1024	1024	1024	2048	2048	2048	2048
L1 data cache size (in KB)	36	40	40	42	42	44	44	44
L1 instruction cache size (in KB)	48	96	96	128	128	160	160	160
L2 cache size (in MB)	1	1.5	2	2	2	2.5	2.5	2.5
LLC cache size (in MB)	64-128	64-800	128-1024	256-1536	256-2048	512-4096	512-4096	512-4096
Number of DDR channels	8 (DDR4)	12 (DDR4)	12 (DDR5)	12 (DDR5)	16 (DDR5)	16 (DDR6)	16 (DDR6)	16 (DDR6)
DDR bandwidth (TB/s)	0.20	0.31	0.61	0.76	1.02	1.1	1.2	1.2
DDR size per socket (in TB)	1.0	3.0	4.5	6.0	8.0	10.0	12.0	12.0
Socket max TDP (Watts)	280	300	400	450	600	600	700	700

L1=level 1 cache; LLC=last-level cache; Fabric=PCIe or new fabric (e.g., CXL); TDP=total power dissipation.

Trendy v konštrukcii procesorov, pamätí a iných počítačových komponentov



(a) 2D or 2.5D packaging now common

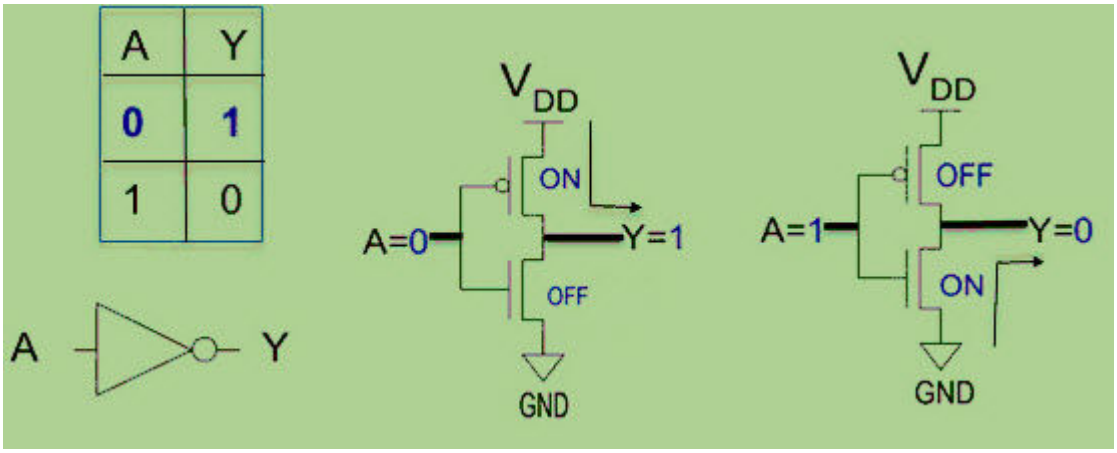


(b) 3D packaging of the future

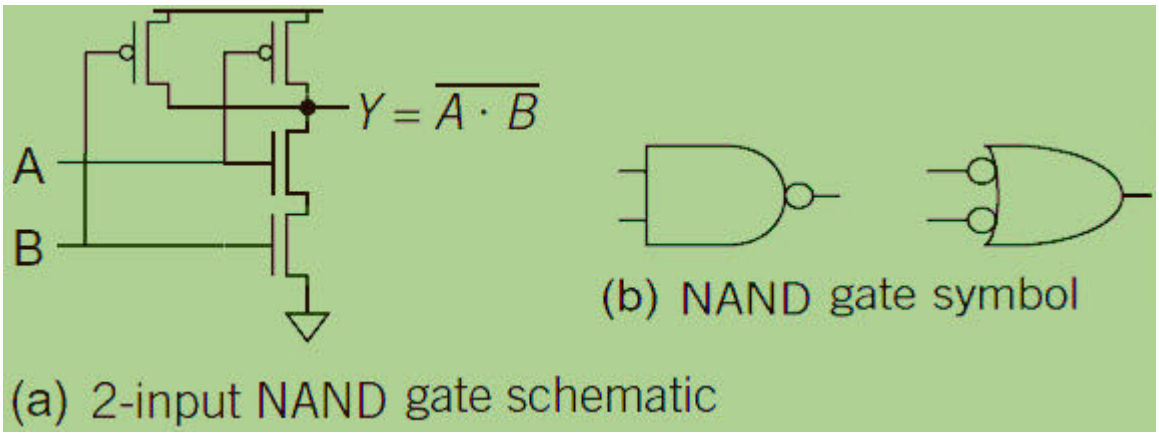
Princíp využitia CMOS technológie pre realizáciu kombinačných logických obvodov

(https://www.elprocus.com/cmos-working-principle-and-applications/)

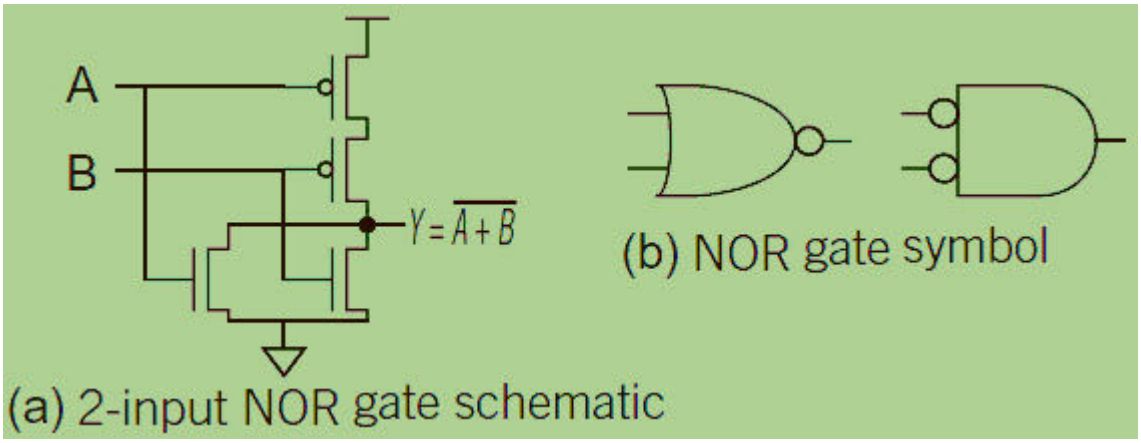
CMOS invertor



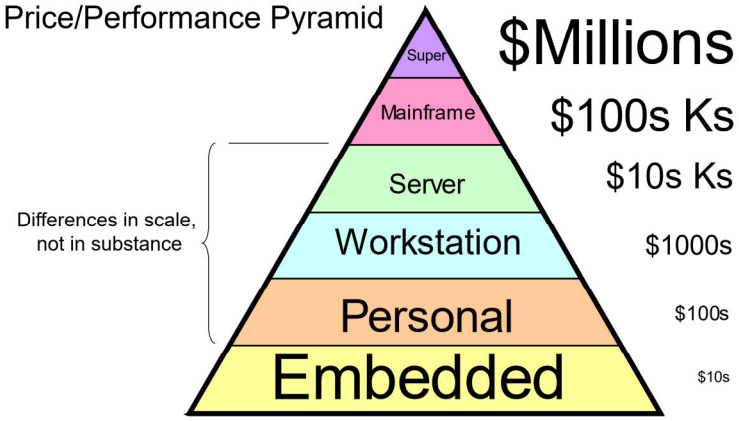
2-vstupové CMOS NAND hradlo



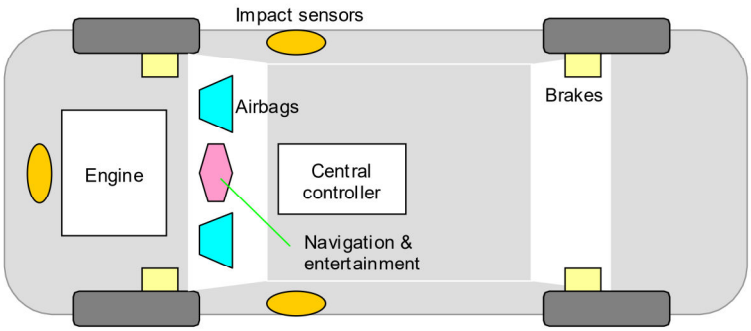
2-vstupové CMOS NOR hradlo



Klasifikácia počítačov z pohľadu výpočtového výkonu a ceny



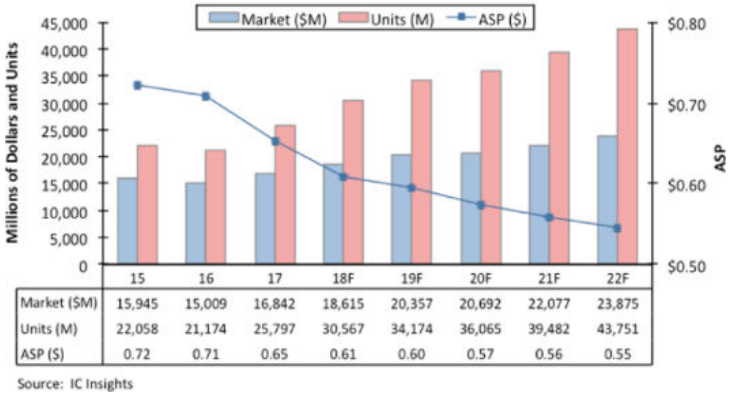
Príklad vstavaného (embedded) počítača



Trh s mikrokontrolérmi (MCU) dosiahol hodnotu 20 miliárd $ v roku 2019

http://www.eenewseurope.com/news/mcu-market-reach-20-billion-2019-0?news_id=110288

A broad uptake of microcontrollers (MCUs) for embedded control, for use with sensors, and with the **Internet of Things (IoT)** is propelling the market forward.



	15	16	17	18F	19F	20F	21F	22F
Market ($M)	15,945	15,009	16,842	18,615	20,357	20,692	22,077	23,875
Units (M)	22,058	21,174	25,797	30,567	34,174	36,065	39,482	43,751
ASP ($)	0.72	0.71	0.65	0.61	0.60	0.57	0.56	0.55

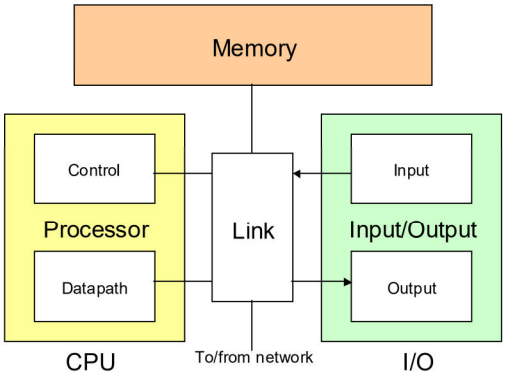
Source: IC Insights

ASP – average system price

The global microcontroller market size was USD 30.63 billion in 2021. The market is projected to reach USD 51.13 billion by 2028 at a CAGR of 7.6%.

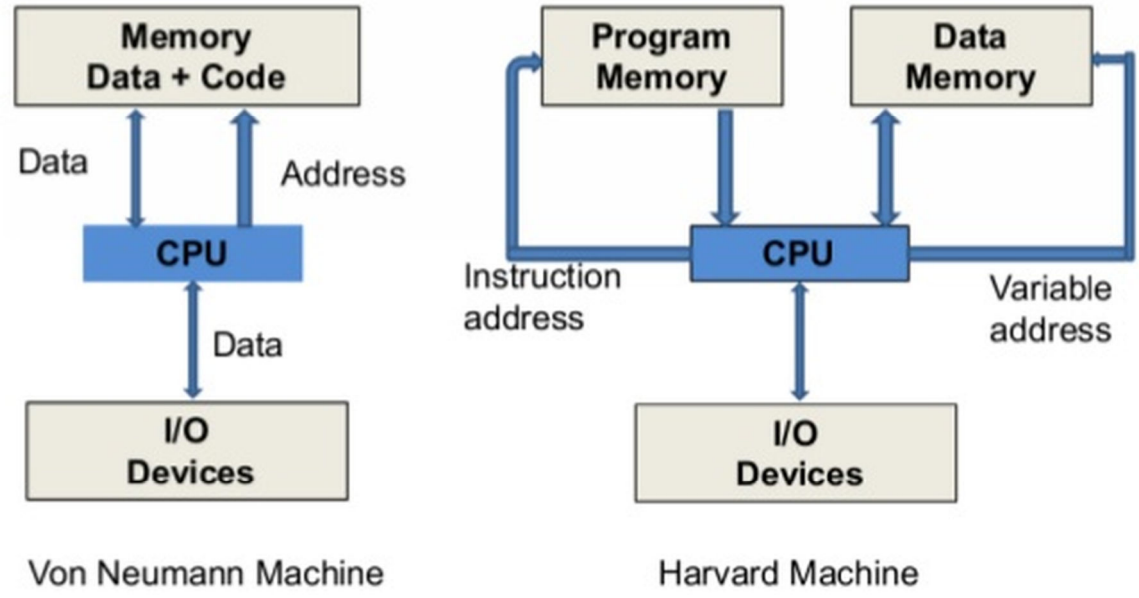
https://www.fortunebusinessinsights.com/microcontroller-market-106430

Subsystémy číslicového počítača



The (three, four, five, or) six main units of a digital computer. Usually, the link unit (a simple bus or a more elaborate network) is not explicitly included in such diagrams.

Harvardska a von Neumanova architektúra



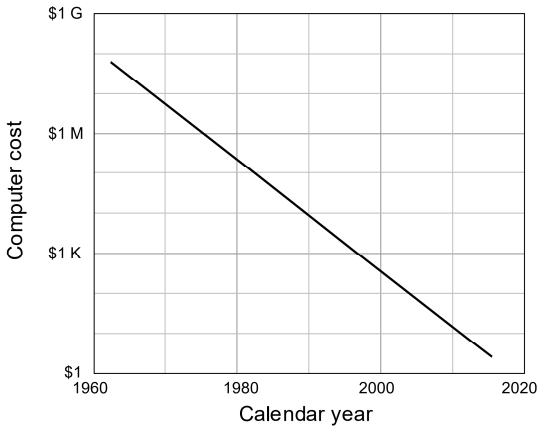
Harvard architecture has separate data and instruction busses, allowing transfers to be performed simultaneously on both busses. A **von Neumann architecture** has only one bus which is used for both data transfers and instruction fetches, and therefore data transfers and instruction fetches must be scheduled - they can not be performed at the same time.

It is possible to have two separate memory systems for a **Harvard architecture**. As long as data and instructions can be fed in at the same time, then it doesn't matter whether it comes from a cache or memory. But there are problems with this. Compilers generally embed data (literal pools) within the code, and it is often also necessary to be able to write to the instruction memory space, for example in the case of self modifying code, or, if an ARM debugger is used, to set software breakpoints in memory. If there are two completely separate, isolated memory systems, this is not possible. There must be some kind of bridge between the memory systems to allow this.

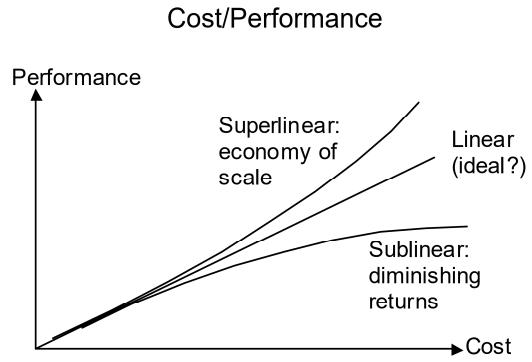
Using a simple, unified memory system together with a Harvard architecture is highly inefficient. Unless it is possible to feed data into both busses at the same time, it might be better to use a von Neumann architecture processor.

Výkonnosť počítačov

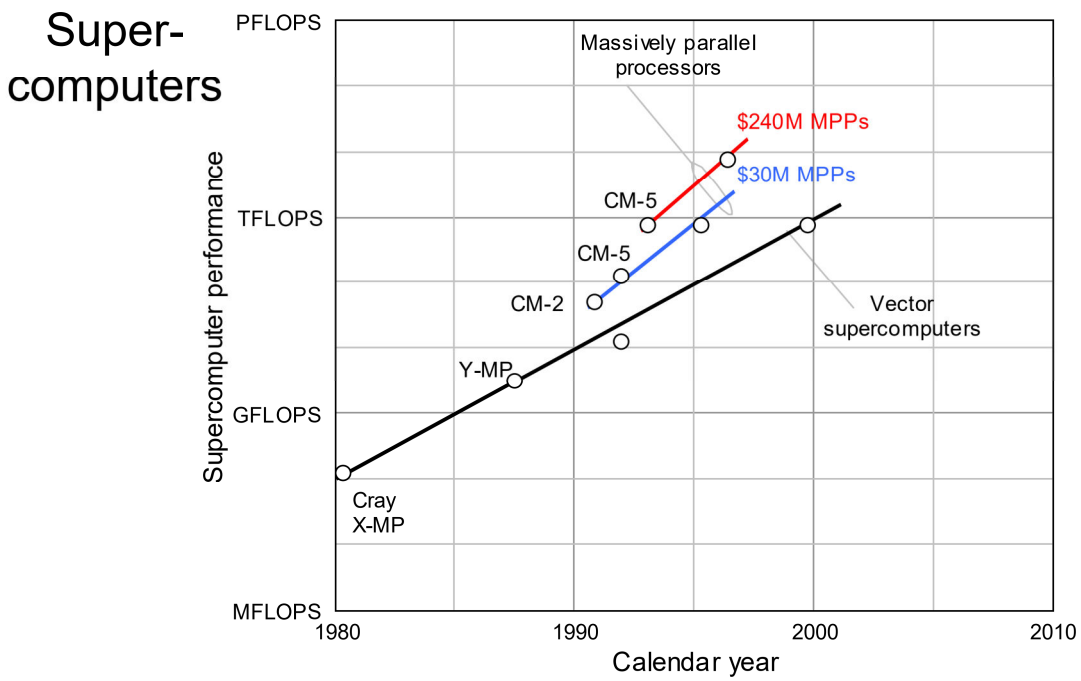
Trend vývoja ceny „počítačov“



Zvyšovanie výkonnosti počítačov v závislosti na cene



Exponenciálny nárast výkonnosti superpočítačov



Výkonnosť nie je vždy najdôležitejšia ...

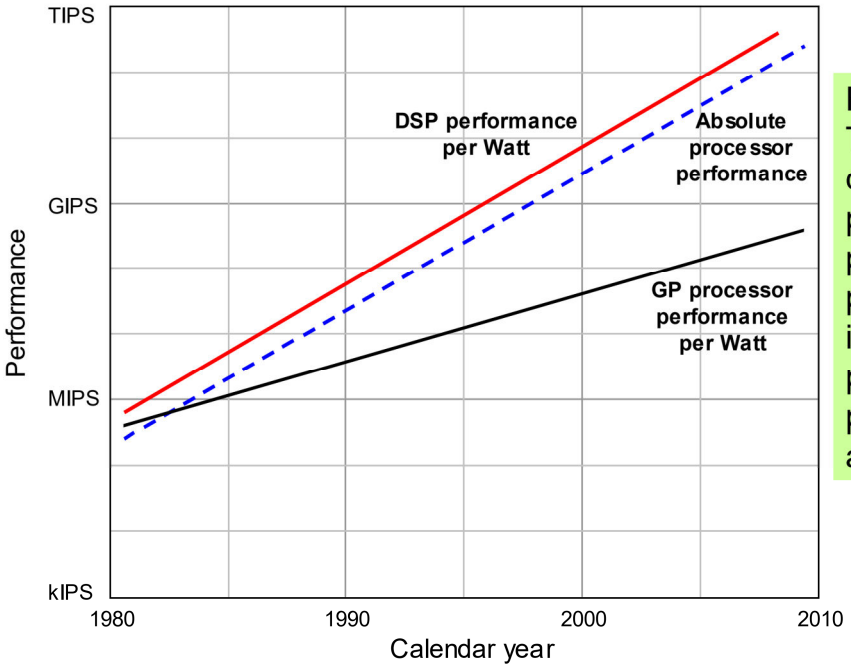
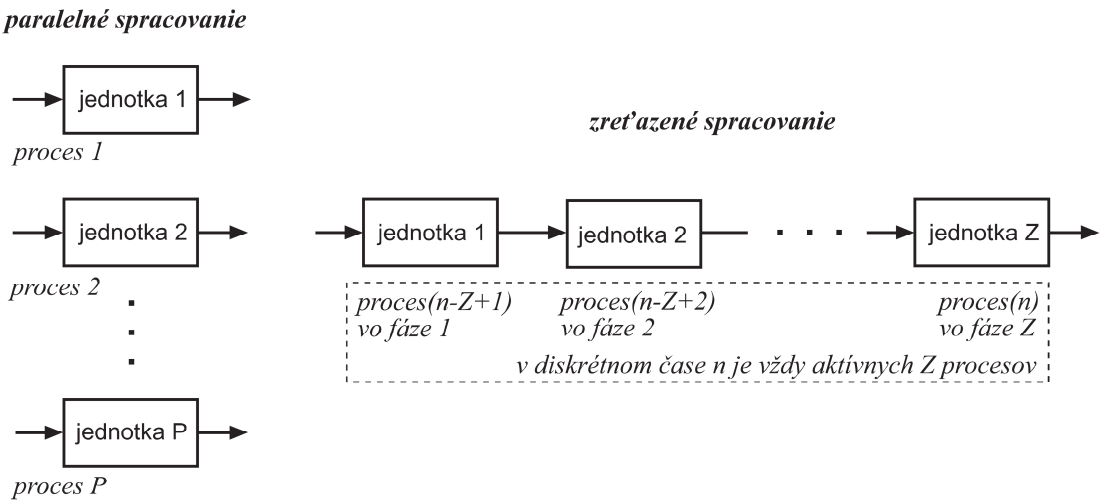


Figure 25.1 Trend in computational performance per watt of power used in general-purpose processors and DSPs.

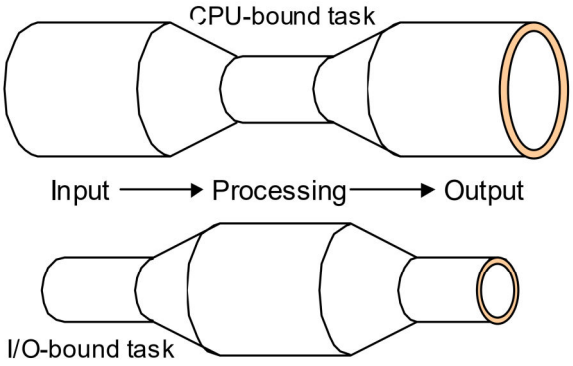
Základné HW spôsoby zvyšovania výkonnosti



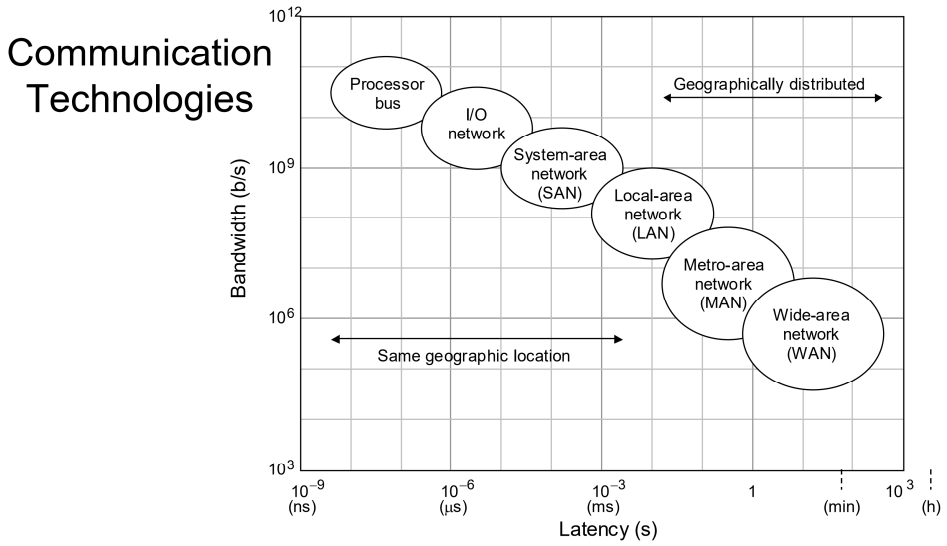
Súbežné (ang. Concurrent) = paralelné - Parallel
alebo
zreťazené (prúdové) - Pipelined

Vstup/výstup a prenos dát

Nevyváženosť CPU výkonnosti a I/O operácií



Latencia a šírka pásma rôznych komunikačných liniek

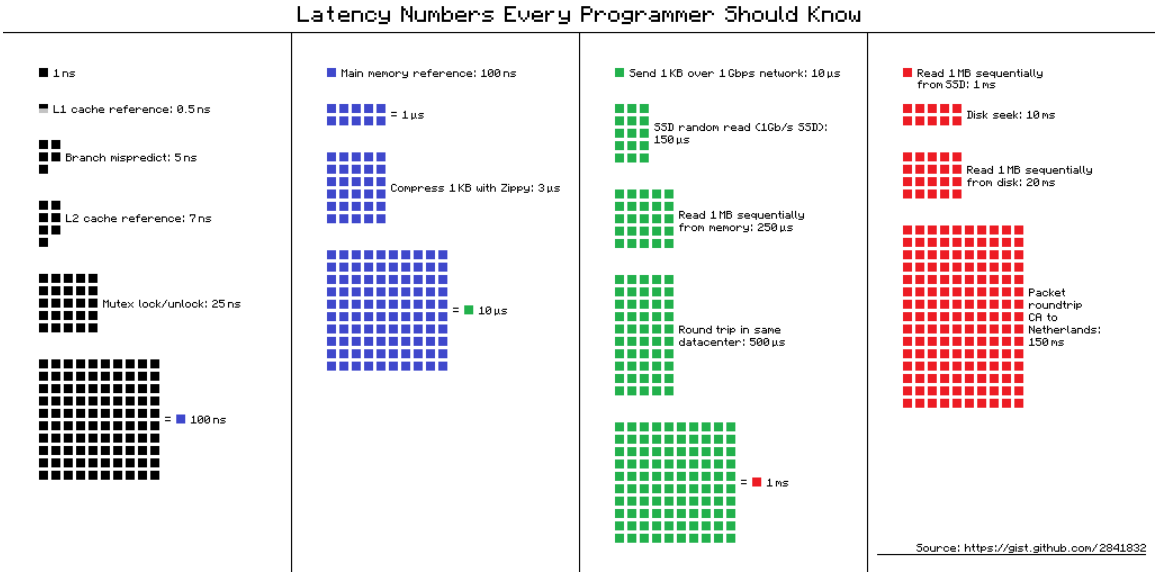


Latencia v počítačovej technike

https://gist.github.com/jboner/2841832#file-latency-txt

```
Latency Comparison Numbers (~2012)
----------------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns
14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns
20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns        3 us
Send 1K bytes over 1 Gbps network       10,000   ns       10 us
Read 4K randomly from SSD*             150,000   ns      150 us
~1GB/sec SSD
Read 1 MB sequentially from memory     250,000   ns      250 us
Round trip within same datacenter      500,000   ns      500 us
Read 1 MB sequentially from SSD*     1,000,000   ns    1,000 us
1 ms  ~1GB/sec SSD, 4X memory
Disk seek                           10,000,000   ns   10,000 us
10 ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20,000 us
20 ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150,000 us
150 ms

Notes
-----
1 ns = 10^-9 seconds
1 us = 10^-6 seconds = 1,000 ns
1 ms = 10^-3 seconds = 1,000 us = 1,000,000 ns
```



Meranie výkonnosti počítačov

Porovnávanie výkonnosti lietadiel: analógia

Aircraft	Passengers	Range (km)	Speed (km/h)	Price ($M)
Airbus A310	250	8 300	895	120
Boeing 747	470	6 700	980	200
Boeing 767	250	12 300	885	120
Boeing 777	375	7 450	980	180
Concorde	130	6 400	2 200	350
DC-8-50	145	14 000	875	80

Speed of sound ≈ 1220 km / h

Rôzne pohľady na výkonnosť (performance)

Performance from the viewpoint of a passenger: Speed

Note, however, that flight time is but one part of total travel time. Also, if the travel distance exceeds the range of a faster plane, a slower plane may be better due to not needing a refueling stop

Performance from the viewpoint of an airline: Throughput

Measured in passenger-km per hour (relevant if ticket price were proportional to distance traveled, which in reality it is not)

Airbus A310	250 × 895 = 0.224 M passenger-km/hr
Boeing 747	470 × 980 = 0.461 M passenger-km/hr
Boeing 767	250 × 885 = 0.221 M passenger-km/hr
Boeing 777	375 × 980 = 0.368 M passenger-km/hr
Concorde	130 × 2200 = 0.286 M passenger-km/hr
DC-8-50	145 × 875 = 0.127 M passenger-km/hr

Performance from the viewpoint of FAA: Safety

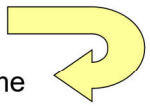
Efektívnosť nákladov: Cost/Performance

Table 4.1 Key characteristics of six passenger aircraft: all figures are approximate; some relate to a specific model/configuration of the aircraft or are averages of cited range of values.

Aircraft	Passen-gers	Range (km)	Speed (km/h)	Price ($M)	Throughput (M P km/hr)	Cost / Performance
					Larger values better	Smaller values better
A310	250	8 300	895	120	0.224	536
B 747	470	6 700	980	200	0.461	434
B 767	250	12 300	885	120	0.221	543
B 777	375	7 450	980	180	0.368	489
Concorde	130	6 400	2 200	350	0.286	1224
DC-8-50	145	14 000	875	80	0.127	630

Súvislosť výkonnosti (Performance) a zrýchlenia (Speedup)

Performance = 1 / Execution time



is simplified to

Performance = 1 / CPU execution time

(Performance of M_1) / (Performance of M_2) = Speedup of M_1 over M_2
= (Execution time of M_2) / (Execution time M_1)

Terminology: M_1 is *x* times **as fast as** M_2 (e.g., 1.5 times as fast)
M_1 is 100(*x* – 1)% **faster than** M_2 (e.g., 50% faster)

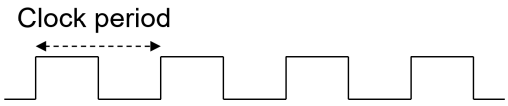
CPU time = Instructions × (Cycles per instruction) × (Secs per cycle)
= Instructions × CPI / (Clock rate)

Instruction count, CPI, and clock rate are not completely independent, so improving one by a given factor may not lead to overall execution time improvement by the same factor.

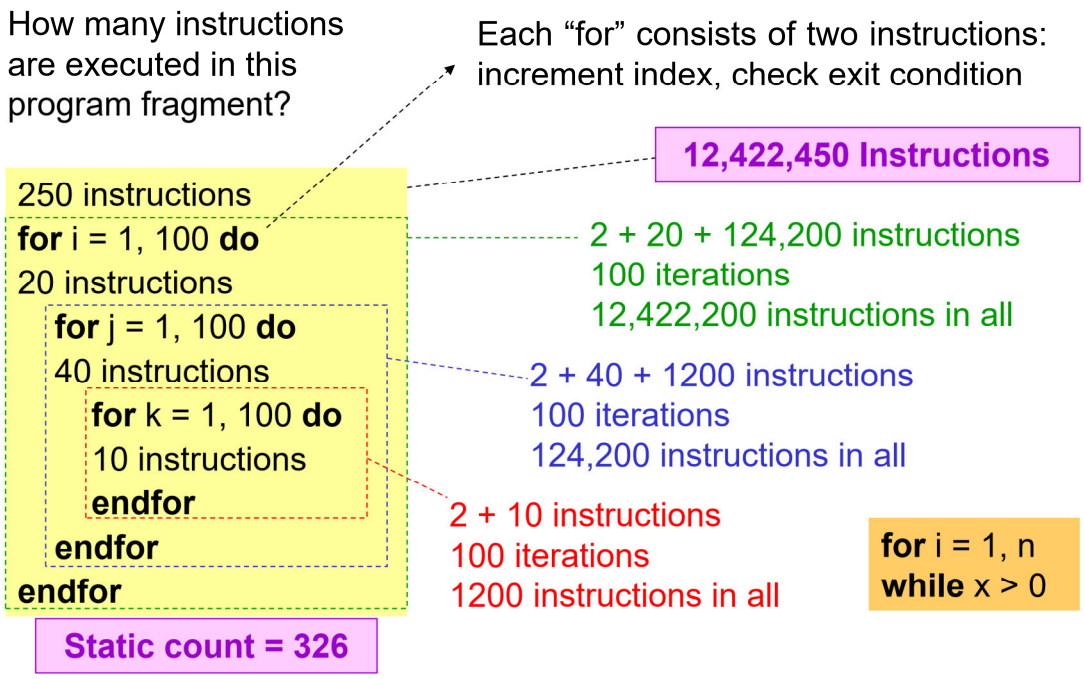
Určenie hodnoty „CPU Time“

CPU time = Instructions × (Cycles per instruction) × (Secs per cycle)
= Instructions × Average CPI / (Clock rate)

Instructions:	Number of instructions executed, not number of instructions in our program (dynamic count)
Average CPI:	Is calculated based on the dynamic instruction mix and knowledge of how many clock cycles are needed to execute various instructions (or instruction classes)
Clock rate:	1 GHz = 10^9 cycles / s (cycle time 10^{-9} s = 1 ns) 200 MHz = 200 × 10^6 cycles / s (cycle time = 5 ns)



Dynamický počet inštrukcií – príklad



Meranie výkonnosti – MIPS, MFLOPS

Definice měření výkonnosti jednotkami MIPS

$$MIPS = \frac{IC}{T_{CPU} \times 10^6} = \frac{IC}{IC \times CPI \times T_{CLK} \times 10^6} = \frac{f_{CLK}}{CPI \times 10^6} = [10^6 \text{ instr.}/s]$$

MIPS závisí na:

- ISA a programu, neboť CPI závisí na programu, mixu instrukcí

Důsledek:

- výkonnost vyjádřena v jednotkách MIPS je závislá na programu, i když se tato závislost často neuvádí

Měření výkonnosti MFLOPS – totéž pro FP instrukce

- k vyjádření špičkové výkonnosti superpočítačů, reálná výkonnost může být značně nižší

Výpočet CPI (Clocks per Instruction) a IPS (Instructions per Second)

Consider two implementations M_1 (600 MHz) and M_2 (500 MHz) of an instruction set containing three classes of instructions:

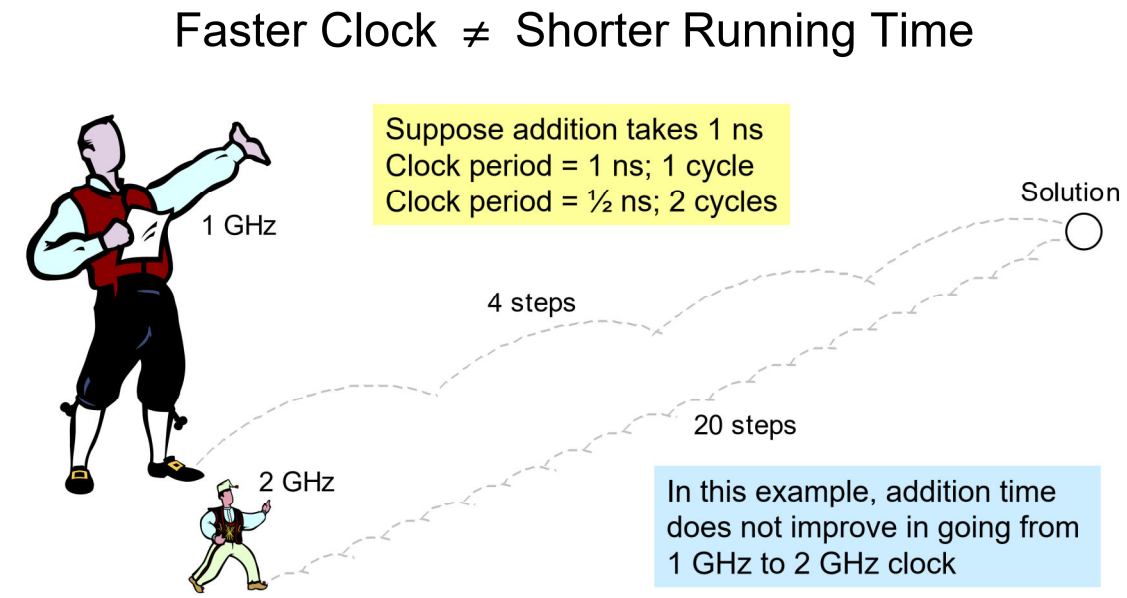
Class	CPI for M_1	CPI for M_2	Comments
F	5.0	4.0	Floating-point
I	2.0	3.8	Integer arithmetic
N	2.4	2.0	Nonarithmetic

a. What are the peak performances of M_1 and M_2 in MIPS?
b. If 50% of instructions executed are class-N, with the rest divided equally among F and I, which machine is faster? By what factor?

Solution

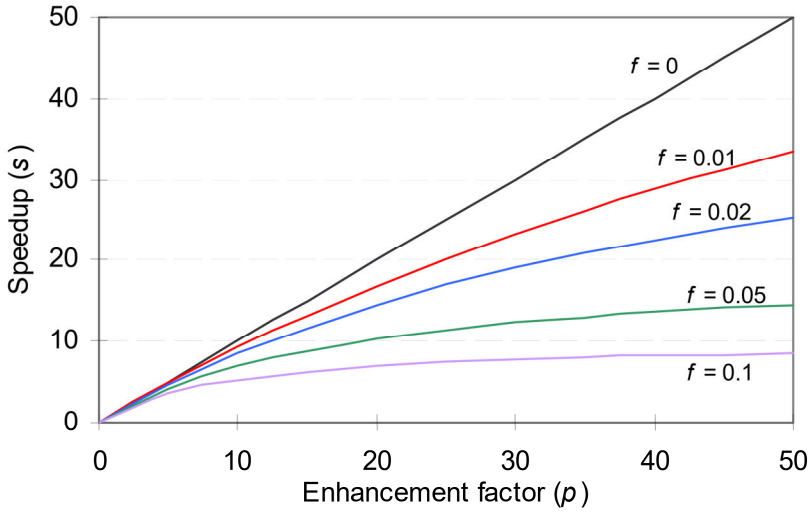
a. Peak MIPS for M_1 = 600 / 2.0 = 300; for M_2 = 500 / 2.0 = 250
b. Average CPI for M_1 = 5.0 / 4 + 2.0 / 4 + 2.4 / 2 = 2.95;
for M_2 = 4.0 / 4 + 3.8 / 4 + 2.0 / 2 = 2.95 $\rightarrow$ M_1 is faster; factor 1.2

Frekvencia hodiny a doba vykonávania



Faster steps do not necessarily mean shorter travel time!

Zvýšenie výkonnosti – Amdahlov zákon



f = fraction unaffected

p = speedup of the rest

$$s = \frac{1}{f + (1-f)/p}$$

$$\leq min(p, 1/f)$$

Amdahl's law: speedup achieved if a fraction f of a task is unaffected and the remaining 1 – f part runs p times as fast

Porovnanie úrovne zvýšenia výkonnosti CPU - Príklad

A processor spends 30% of its time on flp addition, 25% on flp mult, and 10% on flp division. Evaluate the following enhancements, each costing the same to implement:

a. Redesign of the flp adder to make it twice as fast.
b. Redesign of the flp multiplier to make it three times as fast.
c. Redesign the flp divider to make it 10 times as fast.

Solution

a. Adder redesign speedup = 1 / [0.7 + 0.3 / 2] = 1.18
b. Multiplier redesign speedup = 1 / [0.75 + 0.25 / 3] = 1.20
c. Divider redesign speedup = 1 / [0.9 + 0.1 / 10] = 1.10

What if both the adder and the multiplier are redesigned?

Příklad 2

Pro FP výpočty v počítačové grafice se hodně používá operace odmocniny FPSQRT a výkonnost procesorů pro grafiku je na jejím efektivním provádění silně závislá.
Předpokládejme, že FPSQRT odpovídá 20% a všechny FP instrukce odpovídají 50% doby výpočtu kritické zkušební úlohy pro grafiku.
Úkolem je rozhodnout, zda je výhodnější:

1. 10 x zrychlit provádění instrukce FPSQRT, nebo
2. 1.6 x zrychlit provádění všech FP instrukcí?

Řešení:

1. $S_{FPSQRT} = \dfrac{1}{(1-0.2)+\dfrac{0.2}{10}} = \dfrac{1}{0.82} = 1.22$
2. $S_{FP} = \dfrac{1}{(1-0.5)+\dfrac{0.5}{1.6}} = \dfrac{1}{0.8125} = 1.23$ ← výhodnější řešení

Zovšeobecnený Amdahlov zákon

Original running time of a program = 1 = $f_1 + f_2 + \ldots + f_k$

New running time after the fraction f_i is speeded up by a factor p_i

$$\frac{f_1}{p_1} + \frac{f_2}{p_2} + \ldots + \frac{f_k}{p_k}$$

Speedup formula

$$S = \frac{1}{\frac{f_1}{p_1} + \frac{f_2}{p_2} + \ldots + \frac{f_k}{p_k}}$$

If a particular fraction is slowed down rather than speeded up, use $s_j f_j$ instead of f_j / p_j, where $s_j > 1$ is the slowdown factor

Výber vhodného testovania (benchmark) – Príklad

You are an engineer at Outtel, a start-up aspiring to compete with Intel via its new processor design that outperforms the latest Intel processor by a factor of 2.5 on floating-point instructions. This level of performance was achieved by design compromises that led to a 20% increase in the execution time of all other instructions. You are in charge of choosing benchmarks that would showcase Outtel's performance edge.

a. What is the minimum required fraction *f* of time spent on floating-point instructions in a program on the Intel processor to show a speedup of 2 or better for Outtel?

Solution

a. We use a generalized form of Amdahl's formula in which a fraction *f* is speeded up by a given factor (2.5) and the rest is slowed down by another factor (1.2): $1 / [1.2(1 - f) + f / 2.5] \geq 2 \Rightarrow f \geq 0.875$

Skúšobné úlohy (Benchmarks)

Pro měření a vyhodnocení výkonností počítače se používají zkušební úlohy (programy), které se dělí do několika základních skupin:

1. Reálné aplikace: překladače C, Word, Photoshop atd. Zde existuje problém přenositelnosti, tj. závislosti na OS nebo překladači.
2. Upravené aplikace: reálné aplikace jsou stavebními bloky pro zkušební úlohy. Důvodem modifikace je buď zlepšit přenositelnost aplikace, nebo zaměření na určitou část výkonnosti systému.
3. Jádra (kernels): malé klíčové části reálných aplikací (např. Linpack). Nezahrnují vliv paměťového systému (vejdou se do vnitřních skrytých pamětí). Jsou pro uživatele nedostupné.
4. „Toy“ zkušební úlohy: typicky 10 – 100 řádkové programy (Eratosthenovo síto, Quicksort, atd.). Nezahrnují vliv paměťového systému (vejdou se do vnitřních skrytých pamětí). Uživatel dopředu pozná výstup.
5. Umělé zkušební úlohy: podobné filozofii jader. Zkouší nalézt průměrné hodnoty výskytu operací a operandů v rozsáhlých programech (např. Whetstone a Dhrystone).

Eliminace „slabin“ jedné zkušební úlohy jinou vedla k vytvoření sad zkušebních úloh určených pro různé aplikační oblasti.

Uznávané sady zkušebních úloh vydává organizace SPEC (Standard Performance Evaluation Corporation), zahrnují například:

1. Zkušební úlohy zaměřené na CPU
   - SPEC89, SPEC92, SPEC95, SPEC CPU 2000, SPEC CPU 2006
   - celočíselné výopočty (CINT2006), FP výpočty (CFP2006)
   - reálné výpočetní úlohy modifikované pro přenositelnost a minimalizování I/O
2. Zkušební úlohy zaměřené na grafiku
   - SPECviewperf: OpenGL, hlásí počet snímků/s a další údaje
   - SPECapc: výkon různých 3D grafických aplikací (LightWave, 3ds Max, Maya, ...)

3. Zkušební úlohy pro servery
   - CPU: úlohy zaměřené na CPU (SPEC CPU 2006), ale v režimu měření propustnosti (SPECint_rate) – spouští se zpravidla 1 úloha na každý procesor resp. jádro
   - úlohy pro vysoce paralelní systémy: SPEC MPI2007, SPEC OMP2001
   - Java: SPECjbb2005, SPECjEnterprise2010, ...
   - spotřeba: SPECpower_ssj2008
   - síťový souborový systém: SPECsfs2008 – měří propustnost a dobu odezvy serverů, které zpřístupňují soubory pomocí protokolů NFS a CIFS
   - virtualizace: SPECvirt_sc2010

Kromě zkušebních úloh SPEC také stojí za zmínku:

- zkušební úlohy TPC (Transaction Process Council): určené pro transakční zpracování a databáze, měří výkonnost v transakcích za sekundu
- Zkušební úlohy pro vestavěné počítačové systémy (Embedded benchmarks)
  - nová třída zkušebních úloh, hodnotí se také spotřeba, cena atd.
  - nejlépe standardizovanými zkušebními úlohami se jeví sada EDN Embedded Microprocessor Benchmark Consortium (EEMBC – vysloveno „embassy“), obsahující 5 podskupin:
    1. automobilové/průmyslové
    2. zákaznické
    3. síťové
    4. automatizační pro kancelářskou práci
    5. telekomunikační

Zkušební úlohy pro OS MS Windows

- Business Winstone
  - aplikace sady „office“ (Microsoft, Corel, WordPerfect)
  - skripty simulující uživatele přepínajícího a spouštějícího množství aplikací
- CC Winstone
  - práce s větším objemem dat (Photoshop, Premiere a různé audio editovací programy)
  - skripty simulující uživatele spouštějícího skupinu smíšených aplikací
- Winbench
  - skripty spouštějící testy výkonnosti CPU video systémů, disků atd.
  - používají se jádra se zaměřením na každý podsystém

LINPACK – [MFLOPS] FP zkušební úloha

- řešení SLR ($\mathbf{Ax} = \mathbf{b},\ hod(\mathbf{A}) = 100$)
- mnoho FP operací, ale použito jen několik typů
- vzhledem k tomu, že operace $y_i = y_i + a \cdot x_i$ jsou prováděny nejdelší dobu, může se projevit silný vliv i malé, instrukční vnitřní skryté paměti
- data jsou rozmístěna ve velkém prostoru
- vhodná jako zkušební úloha pro vektorové a paralelní superpočítače
- verze LINPACK:
  - single/double
  - rolled/unrolled