

Architektura počítačů

01

Martin Sysel

Ing. Adam Malatinský

Obsah předmětu

Přednášky

- Hardware PC
- Periferie
- Architektura
- Princip funkce
- Vývoj

Cvičení

2 hodiny jednou za 14 dní

- Celkem 7 cvičení
- <https://moodle.utb.cz/>
 - vyuka.fai.utb.cz
 - Bakalářské studium
 - Architektura počítačů

Požadavky

- Zápočet
 - Docházka (min 80% - 6 aktivních účastí na cvičení)
 - Odevzdané **všechny** protokoly.
 - Projekt (rozsah min. 7 normostran)
 - Výběr projektu na začátku semestru
 - **Odevzdání do 12. týdne**
- Zkouška
 - Pouze se zápočtem
 - Písemně (základ známky) + 1-2 doplňující otázky

LITERATURA

1. HENNESSY, John L. a David A. PATTERSON. *Computer architecture: a quantitative approach*. 5th ed. Elsevier, 2012. ISBN 978-0-12-383872-8.
2. MUELLER, Scott. *Upgrading and repairing PCs*. 21st ed. Pearson Education, 2013. ISBN 978-0-7897-5000-6.
3. TIŠNOVSKÝ Pavel. *Od logických obvodů k mikroprocesorům*. root.cz, 2008. Dostupné z: <https://www.root.cz/clanky/od-logickych-obvodu-k-mikroprocesorum/>
4. MOORE, Gordon E. Cramming more components onto integrated circuits. *Electronics Magazine*, Volume 38, Number 8, April 19, 1965. Dostupné z: <https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/moores-law-electronics.pdf>
5. PELIKÁN Jaroslav. *Von Neumannovo schéma*. Dostupné z: <https://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/OBSAH.HTML>
6. TIŠNOVSKÝ Pavel. *Jak pracuje počítač?* root.cz, 2008. Dostupné z: <https://www.root.cz/clanky/jak-pracuje-pocitac/>
7. PRÁGER, Milan a SÝKOROVÁ, Irena. Jak počítače počítají. (Czech) [How computers compute]. *Pokroky matematiky, fyziky a astronomie*, vol. 49 (2004), issue 1, pp. 32-45. Dostupné z: <https://dml.cz/handle/10338.dmlcz/141207>
8. IEEE standard for binary floating-point arithmetic. *ANSI/IEEE Std 754-1985*, 1985. Dostupné z: <http://kfe.fjfi.cvut.cz/~klimo/nm/ieee754.pdf>
9. BRANDEJS, Michal. *Mikroprocesory Intel – Pentium* [online]. Brno: Fakulta informatiky, Masarykova univerzita, 2010. Dostupný z WWW: http://www.fi.muni.cz/usr/brandejs/Brandejs_Mikroprocesory_Intel_Pentium_2010.pdf
10. Intel Corporation. *Intel® 64 and IA-32 Architectures Software Developer's Manual: Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D and 4*. May 2020. Dostupné z: <https://software.intel.com/content/dam/develop/public/us/en/documents/325462-sdm-vol-1-2abcd-3abcd.pdf>
11. MALÝ, Martin. *Hradla, volty, jednočipy: Úvod do bastlení*. CZ.NIC, Praha, 2017. ISBN 978-80-88168-26-3. Dostupné z: https://knihy.nic.cz/files/edice/hradla_volty_jednocipy.pdf
12. TIŠNOVSKÝ Pavel. *Učíme trpaslíky počítat*. root.cz, 2008. Dostupné z: <https://www.root.cz/clanky/ucime-trpasliky-pocitat/>
13. PATTERSON, David A., John L. HENNESSY, Perry ALEXANDER, et al. *Computer organization and design: the hardware/software interface*. Fifth edition. Amsterdam: Morgan Kaufmann, [2014]. Morgan Kaufmann series in computer architecture and design. ISBN 978-0-12-407726-3.
14. Intel Corporation. *P6 Family of Processors: Hardware Developer's Manual*. September 1998. Dostupné z: <https://download.intel.com/design/PentiumII/manuals/24400101.pdf>
15. WikiChip. Skylake (client) - Microarchitectures – Intel. On-line, last modified on 20 June 2020. Dostupné z: [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(client\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)).
16. Intel Corporation. *Intel® 64 and IA-32 Architectures Optimization Reference Manual*. On-line, Submitted: April 01, 2018, Last updated: May 19, 2020. Dostupné z: <https://software.intel.com/content/dam/develop/public/us/en/documents/64-ia-32-architectures-optimization-manual.pdf>
17. AMD. *AMD64 Architecture Programmer's Manual: Volumes 1-5*. On-line, Last updated: April, 2020. Dostupné z: <https://www.amd.com/system/files/TechDocs/40332.pdf>

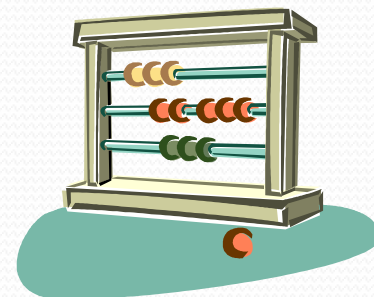
LITERATURA

1. AMD. Developer Guides, Manuals & ISA Documents. On-line, 2020. Dostupné z: <https://developer.amd.com/resources/developer-guides-manuals/>
2. GOOK, Michael. *Hardwarová rozhraní: průvodce programátora*. Brno: Computer Press, 2006. Hardware (Computer Press). ISBN 80-251-1019-2.
3. NVM Express. NVM Express Base Specification. Revision 1.4b, 2020. Dostupné z: https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4b-2020.09.21-Ratified.pdf
4. PCI-SIG. PCI Express M.2 Specification. Revision 1.0, November 1, 2013.
5. VERMA, ANUJ & DAHIYA, PAWAN. (2017). *PCIe BUS: A State-of-the-Art-Review*. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP). 7. 24-28. 10.9790/4200-0704012428. Dostupné z: https://www.researchgate.net/publication/318360748_PCIe_BUS_A_State-of-the-Art-Review
6. SAIF. PCIe Basic. One Convergence. 2017. Dostupné z: <https://www.slideshare.net/SaifuddinKaijar/pcie-basic>
7. Intel Corporation. PHY Interface For the PCI Express, SATA, USB 3.1, Display Port, and USB4 Architectures. Version 6.0, 2020. Dostupné z: <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/phy-interface-pci-express-sata-usb30-architectures-3-1.pdf>
8. USB-IF. Universal Serial Bus Specification. Revision 2.0, 2000. Dostupné z: https://www.usb.org/sites/default/files/usb_20_20190524.zip
9. USB-IF. Universal Serial Bus 3.2 Specification. Revision 1.0, 2017. Dostupné z: https://www.usb.org/sites/default/files/usb_32_20200716_0.zip
10. USB-IF. Universal Serial Bus 4 (USB4™) Specification. Version 1.0, 2020. Dostupné z: https://www.usb.org/sites/default/files/USB4%20Specification_4.zip
11. TANENBAUM, ANDREW S. a TODD AUSTIN. *Structured computer organization*. Sixth Edition. United States of America: Pearson Education, 2013. ISBN 978-0-13-291652-3.
12. STALLINGS, William. *Computer Organization and Architecture: Designing for Performance*. Tenth Edition. United States of America: Pearson Education, 2016. ISBN 978-0-13-410161-3.
13. JOEL EMER, KRSTE ASANOVIC, and ARVIND. 6.823 Computer System Architecture. Fall 2005. Massachusetts Institute of Technology: MIT OpenCourseWare. Dostupné z: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/index.htm>
14. WENTZLAFF DAVID. Computer Architecture. [Princeton University/Course](https://www.princeton.edu/~course/). Dostupné z: <https://www.coursera.org/learn/comparch>.
15. MUTLU ONUR. Introduction to Computer Architecture. [Carnegie Mellon University](https://www.carnegie-mellon.edu/). Dostupné z: <https://course.ece.cmu.edu/~ece447/s13/doku.php?id=schedule>.

Historie

- Abakus

- Vznikl přibližně před 5000 lety.
- Používal se ve starém Řecku a Římě.
 - Dřevěná (hliněná) destička - vkládaly se kamínky („calculi“)



- Logaritmické tabulky

- 1614 - John Napier objevil matematickou metodu
 - násobení a dělení pomocí sčítání a odčítání s využitím logaritmů.
 - Sestaveny první logaritmické tabulky
 - následovalo logaritmické pravítko

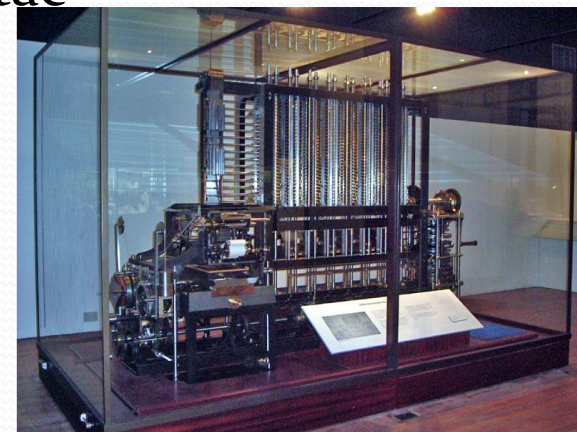
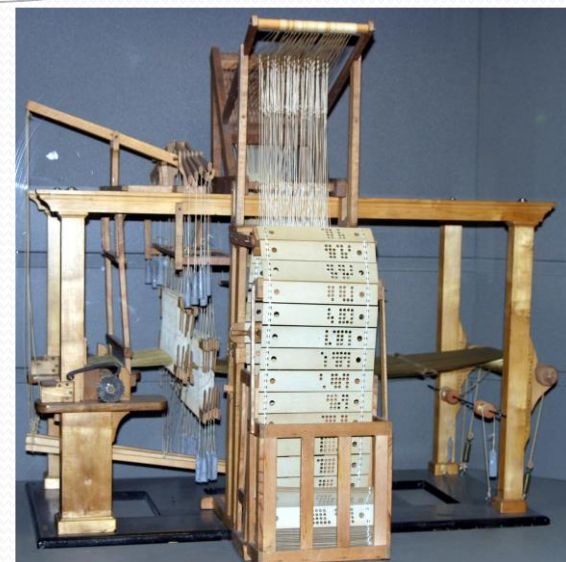
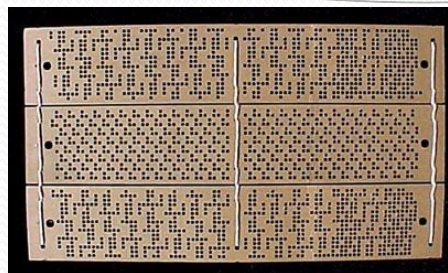
- Mechanické kalkulátory

- 1623 Wilhelm Schickard – počítací hodiny
 - Sčítat a odčítat šesticiferná čísla
 - Využití ozubeného kola
 - Většina byla založena na desítkové soustavě
- G. W. Leibnitz (1646 - 1716)
 - dvojková neboli binární soustava



Historie

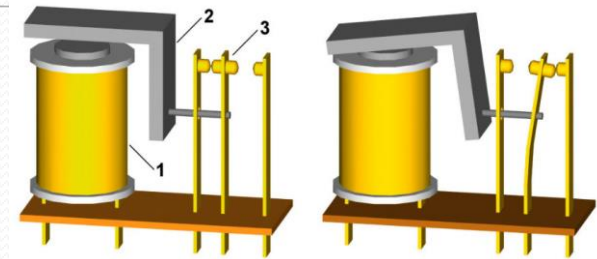
- Děrné štítky (kartičky)
 - 1725 Basile Bouchon
 - Využití pro tkalcovský stav
 - 1801 Joseph-Maria Jacquard
 - možné vyměnit beze změny v mechanice samotného stavu.
Považováno za milník v programovatelnosti strojů
- Programovatelné stroje
 - Cílem je univerzální programovatelný počítač
- Charles Babbage, 1791 – 1871
 - 1822: difference engine
 - aproximace hodnoty polynomů 6. stupně
 - tabulace logaritmů a trigonometrických funkcí



Alan Turing (1912 – 1954)

- zakladatel moderní počítačové vědy
 - formalizace algoritmu a výpočtu
- 1936: abstraktní Turingův stroj
 - abstraktní zařízení pro manipulaci se symboly
 - mohou simulovat libovolný počítač
 - univerzální Turingův stroj
- **Turingův test**
 - Vztah člověka a stroje
 - základy vědy o “umělé inteligenci”
- Test inteligence stroje
 - komunikace prostřednictvím dálkopisu
 - pokud není možno v rozumném čase rozlišit zda odpovídá stroj či člověk, pak stroj “vykazuje znaky inteligence”

Historie – 0. generace



- elektromechanické počítače využívající většinou relé
- 1934 - 1938 Konrád Zuse - **Z₁**
 - Pracoval v dvojkové soustavě s aritmetikou v plovoucí čárce a programem na děrné pásce.
 - Neuměl podmíněné skoky.
 - Elektromechanický s kolíčkovou pamětí na 16 čísel
 - Velmi poruchový
- **Z₂, Z₃**
 - Pracoval s dvojkovou aritmetikou v pohyblivé čárce a prováděl až 50 aritmetických operací za minutu. Délka slova byla 22 bitů, reléová paměť měla kapacitu 64 slov.
 - Počítač byl v roce 1944 zničen při leteckém náletu.
- Po válce poměrně spolehlivý **Z-4** pro univerzitu v Zürichu
- **Z-5** pro Leitzovy optické závody.

Historie - 0. generace

- **1939-1943 - Mark I**

- Financovala firma IBM (International Business Machines).
- Výpočetní laboratoř Hardwardské univerzity v Cambridge.
- Počítač pracoval v desítkové soustavě s pevnou čárkou.
- Dovedl sečíst dvě čísla za 0,3 s, vynásobit je za 6 s a vypočítat např. hodnotu sinus daného úhlu během jedné minuty.
- Během stovky hodin vypočtena konfigurace uranové nálože první atomové pumy

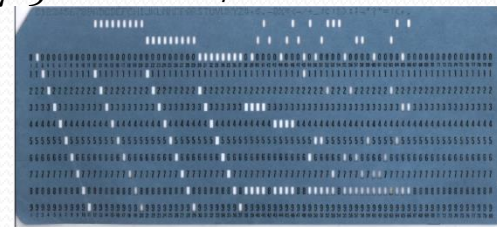
- **Mark II**

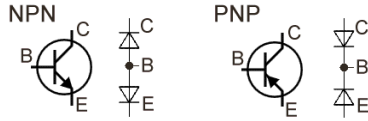
- čistě reléové zařízení.
- Aritmetika pracovala v plovoucí čárce s desítkovými číslicemi, které byly dvojkově kódovány pomocí čtyř relé.
- Operační paměť počítače mohla pojmout až 100 čísel s deseti platnými číslicemi.
- Sčítání již trvalo pouze 0,125 s a násobení průměrně 0,25 s.
- Celý počítač obsahoval přibližně 13 000 relé.

Historie - 1. generace



- Charakteristická použitím elektronek (tzv. elektronika)
 - v menší míře též ještě relé (elektromechanika)
- neefektivní, drahé, vysoký příkon, velká poruchovost a velmi nízká rychlost.
- využity děrné štítky a děrné pásky, řádkové tiskárny.
- Neexistence operačních systémů a programovacích jazyků, později assembler.
- Jeden tým lidí pracoval jako konstruktéři, operátoři i technici
 - úspěchem bylo ukončit výpočet bez poruchy počítače.
- ENIAC, MANIAC, UNIVAC
 - ENIAC - univerzita v Pensylvánii (1944)
 - 18 000 elektronek, 10 000 kondenzátorů, 70000 odporů, 1300 relé, byl chlazen dvěma leteckými motory, umožňoval vytvoření smyčky i podmíněné skoky. 5000 sčítání/s
 - zabíral plochu asi 63 m² a vážil asi 27 tun.
- EDVAC – Bell Laboratories, John von Neumann





1947 - objeven tranzistorový efekt v Bellových laboratořích. William Shockley, John Bardeen a Walter Brattain. 1956 udělena Nobelova cena

Historie - 2. generace

- Použití tranzistorů (tzv. polovodičová elektronika)
 - Zlepšení všech parametrů (zmenšení rozměrů, zvýšení rychlosti a spolehlivosti, snížení energetických nároků).
 - Počátek kolem roku 1959
- Vnitřní paměť 16- 32 kB
- Využívání feritových pamětí
 - Magnetická páska, vývoj magnetických disků
 - Typická páska byla asi 400 metrů dlouhá a 1,5 až 2,5 cm široká obsahovala přibližně 5 MB informací.
- Dávkové zpracování dat
- Operační rychlost 10.000 až 250.000 operací/s
- Počátek využívání
 - operačních systémů
 - jazyka symbolických adres
 - první programovací jazyky (COBOL, FORTRAN, ALGOL)

1958 Jack St. Clair Kilby z Texas Instruments
nápad vyrobit jednodílnou součástku z kousku křemíku
tzv. integrovaný obvod (IO).

Historie - 3. generace

- Použití integrovaných obvodů (polovodičová elektronika)
 - Od roku 1964
- Zrychlení, zlevnění, miniaturizace, modularita
- V této době byl výkon počítače úměrný druhé mocnině jeho ceny.
 - vyplatilo se koupit co nejvýkonnější počítač a prodávat strojový čas.
- Pro ukládání dat se používá magnetický disk
 - 1967 – IBM – první floppy disk
- Operační rychlost až 1.000.000 operací/s
- Vnitřní paměť 0,5 až 2 MB
- Rozvoj operačních systémů, terminálové připojení
 - Multiprogramování
 - jedna uloha čeká na dokončení I/O operace, druhá je zpracovávána procesorem
 - Multitasking
 - Procesy se na procesoru rychle střídají – zdánlivě jsou zpracovávány najednou.



Historie



- **Robert Noyce**

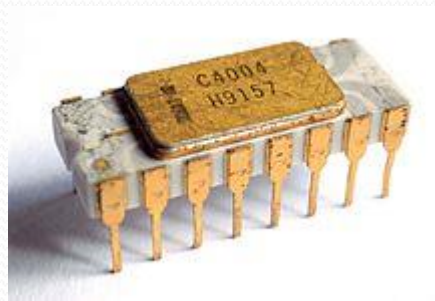
- vyvinul základní prvky technologie hromadné výroby čipů.
- Vyvinul fotomasku a fotolitografii, pasivaci polovodičových povrchů, napařování kovových odporů a kovových spojovacích kontaktů.

- **Gordon Moore**

- 1964 formuloval domněnku, že kapacita integrovaných obvodů se každých 12 až 18 měsíců zdvojnásobí.
- Tento postulát je znám jako tzv. Mooreův zákon
 - Zatím platí.

- **Vyroben první mikroprocesor**

- 1971 - Intel 4004 - 4bitový



Historie - 4. generace



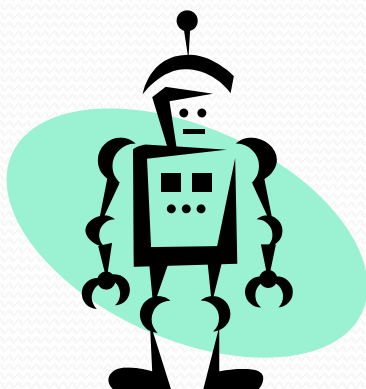
- Začátkem je vyrobení prvního mikroprocesoru
 - obsahují celý procesor v jednom pouzdře
 - dřívější procesory se skládaly z více obvodů
- Miniaturizace integrovaných obvodů
 - zvýšila se spolehlivost, zmenšily rozměry, zvýšila rychlost a kapacita pamětí.
- 1981 – uvedení prvního IBM PC
- Zvolen směr, kterým se dodnes ubíráme
 - Ústup mainframe ve prospěch pracovních stanic

Budoucnost

5. generace

- Umělá inteligence
- Myšlenkové postupy čl.
- Konverzace
- Neuronové sítě
- Kvantové počítače

...



QUANTUM COMPUTING

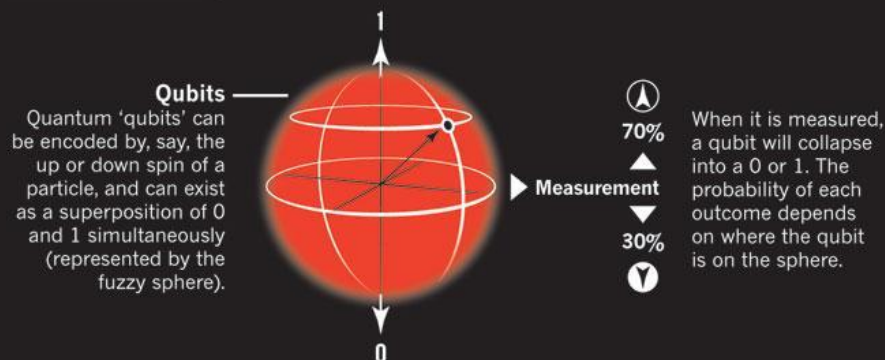
Devices based on subatomic physics could make calculations far faster than conventional machines — if nothing spoils their quantum weirdness.

1. SUPERPOSITION



Bits

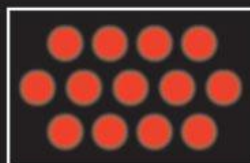
A classical computer encodes information in strings of 'bits', which can take one of two values: 0 or 1.



2. QUANTUM COMPUTATION USING ENTANGLEMENT

Before computation

Data are spread across entangled qubits, which are isolated from the environment.



Perform
computation



After computation

The entangled qubits have processed their information in parallel.

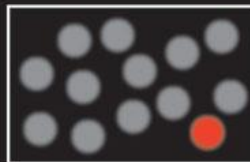


One qubit serves as a spokesman. Taking an average of measurements (0 or 1) over many runs gives the answer.

3. QUANTUM COMPUTATION USING DISCORD

Before computation

Only one qubit is protected from the environment.



Perform
computation



After computation

The other qubits have been exposed to noise and disruption.



Surprisingly, measuring the protected qubit and averaging over many runs still gives the right answer.

Why quantum computation?

The history of computer technology has involved a sequence of changes from one type of physical realisation to another – from gears to relays to transistors to integrated circuits ... and so on. Today's advanced lithographic techniques can create chips with features only a fraction of micron wide. Soon they will yield even smaller parts and inevitably reach a point where logic gates are so small that they are made out of only a handful of atoms.



On the atomic scale matter obeys the rules of quantum mechanics, which are quite different from the classical rules that determine the properties of conventional logic gates. So if computers are to become smaller in the future, new, quantum technology must replace or supplement what we have now. The point is, however, that quantum technology can offer much more than cramming more and more bits onto silicon and multiplying the clock-speed of microprocessors. It can support an entirely new kind of computation with qualitatively new algorithms based on quantum principles!

What are qubits?

From a physical point of view a bit is a physical system which can be prepared in one of the two different states representing two logical values : no or yes, false or true, or simply 0 or 1.

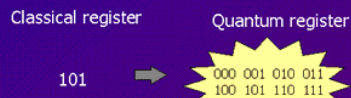
Classical Bit
0 or 1

Quantum Bit
0 or 1 or **01**

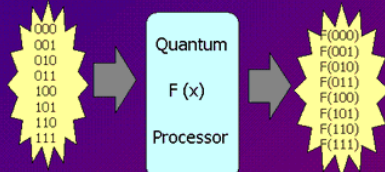
Quantum bits, called qubits, are implemented using quantum mechanical two state systems; these are not confined to their two basic states but can also exist in superpositions: effectively this means that the qubit is both in state 0 and state 1.

What are Quantum Computers ?

Any classical register composed of three bits can store in a given moment of time only one out of eight different numbers. A quantum register composed of three qubits can store in a given moment of time all eight numbers in a quantum superposition.



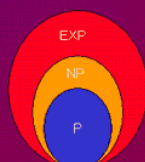
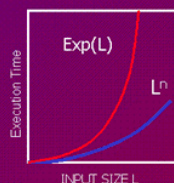
Once the register is prepared in a superposition of different numbers one can perform operations on all of them.



Thus quantum computers can perform many different calculations in parallel: a system with N qubits can perform 2^N calculations at once! This has impact on the execution time and memory required in the process of computation and determines the efficiency of algorithms.

How powerful are quantum computers?

For an algorithm to be efficient, the time it takes to execute the algorithm must increase no faster than a polynomial function of the size of the input. Think about the input size as the total number of bits needed to specify the input to the problem – for example, the number of bits needed to encode the number we want to factorize. If the best algorithm we know for a particular problem has the execution time (viewed as a function of the size of the input) bounded by a polynomial then we say that the problem belongs to class P.



Problems outside class P are known as hard problems. Thus we say, for example, that multiplication is in P whereas factorization is not in P. "Hard" in this case does not mean "impossible to solve" or "non-computable." It means that the physical resources needed to factor a large number scale up such that, for all practical purposes, it can be regarded as intractable. However some quantum algorithms can turn hard mathematical problems into easy ones – factoring being the most striking example so far.

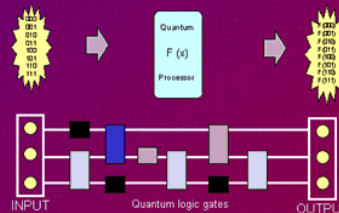


The difficulty of factorisation underpins the security of what are currently the most trusted methods of public key encryption, in particular of the RSA

(Rivest, Shamir and Adelman) system, which is often used to protect electronic bank accounts. Once a quantum factorisation engine (a special-purpose quantum computer for factorising large numbers) is built, all such cryptographic systems will become insecure.

Potential use of quantum factoring for code-breaking purposes has raised the obvious suggestion of building a quantum computer.

How to build quantum computers?



In principle we know how to build a quantum computer; we start with simple quantum logic gates and connect them up into quantum networks. A quantum logic gate, like a classical gate, is a very simple computing device that performs one elementary quantum operation, usually on two qubits, in a given time. Of course, quantum logic gates differ from their classical counterparts in that they can create, and perform operations, on quantum superpositions.

Want to learn more?

Please visit the website cam.qubit.org

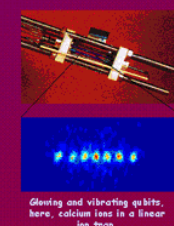
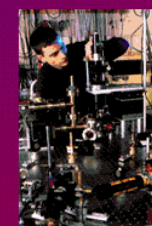
The poster was prepared by Armin Heilmann and Daniel O'Brien from the Centre for Quantum Computation at the University of Cambridge.

Can we build quantum computers?

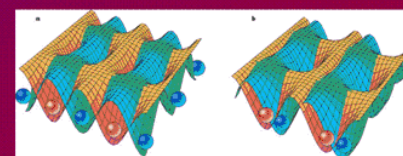
As the number of quantum gates in a network increases, we quickly run into some serious practical problems. The more interacting qubits are involved, the harder it tends to be to engineer the interaction that would display the quantum properties. The more components there are, the more likely it is that quantum information will spread outside the quantum computer and be lost into the environment, thus spoiling the computation. This process is called decoherence. Thus our task is to engineer sub-microscopic systems in which qubits affect each other but not the environment.

What are the most promising technologies?

It is not clear which technology will support quantum computation in future. Today simple quantum logic gates involving two qubits are being realised in laboratories. Current experiments range from trapped ions...

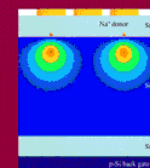


...via atoms in an array of potential wells created by a pattern of crossed laser beams...



...to electrons in semiconductors.

The next decade should bring control over several qubits and, without any doubt, we shall already begin to benefit from our new way of harnessing nature.



UNIVERSITY OF
CAMBRIDGE

Historie - přehled

Generace	Rychlost [operace/s]	Součástky
0.	Jednotky	Relé
1.	100 - 1000	Elektronky
2.	Tisíce	Tranzistory
3.	desetitisíce	IO
4.	Statisíce -> desítky (stovky) milionů	IO – vysoká integrace

Základní pojmy

- **Program**

- Algoritmus zapsaný v programovacím jazyce.
- Na úrovni technického vybavení počítače se jedná o posloupnost instrukcí.

- **Instrukce**

- předpis k provedení nějaké činnosti
 - většinou jednoduché
- realizovatelné přímo technickým vybavením počítače
 - např. přičtení jedničky, uložení hodnoty do paměti apod.

Základní pojmy

- **Firmware**

- programové vybavení, které je součástí hardwaru

- **Řadič (controller)**

- Převádí příkazy v symbolické formě (instrukce) na posloupnost signálů ovládajících připojené zařízení.
 - Jedná se o zařízení, které řídí činnost jiného zařízení.

- **Sběrnice**

- soustava vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače.

Základní pojmy

- **Bit**

- 1 bit (binary digit – dvojková číslice)
 - základní jednotka informace.
 - Poskytuje množství informace potřebné k rozhodnutí mezi dvěma možnostmi.
 - Jednotka bit se označuje **b**
 - Nabývá pouze dvou hodnot – **0**, **1**.

- **Byte**

- jednotka informace, která se označuje **B**
- platí $1\text{ B} = 8\text{ b}$

Základní pojmy

- **Nibble**

- 4 bits = 1 nibble

- **Word**

- 1 W = 2 B = 16 b.

- **Doubleword (DW)**

- 1 DW = 2 W = 4 B = 32 b

- **range**

- $2^8 = 256$ (0-255)
- $2^{16} = 65.536$ (0-65535)
- $2^{32} = 4.294.967.296$ (0 - $2^{32}-1$)
- $2^{64} = 18.446.744.073.709.551.616$ (0 - $2^{64}-1$)

Quadword

8 B = 64 b

$18 \cdot 10^{18}$ trillion/quintillion

ASCII

- American Standard Code for Information Interchange
- Každé z 256 hodnot je přiřazen jeden znak
 - Původní tabulka používala 7 bitů, tedy pouze 128 znaků
 - Rozšířený kód ASCII definuje 256 znaků
 - Přiřazení nemá žádný hlubší smysl
 - 0 – 32 řídicí kódy a speciální znaky
 - Např. End of Transmission (EOT = 04h) – konec přenosu
 - 33 – 127 anglická abeceda (malá, velká písmena), čísla, závorky ...
 - 128 – 255 kódování národních znaků
 - Nejednoznačná, národní abecedy
 - Windows-1250 Microsoft ve Windows (středoevropské jazyky)
 - ISO 8859-2 standard ISO, např. v Linuxu
 - CP852 (Latin2) - kód IBM např. v DOS.

Unicode

- vyvíjen v součinnosti s ISO/IEC 10646
 - mezinárodní univerzální kódovaná znaková sada
- Nejnovější verze (13.0) obsahuje 143,859 znaků
 - Teoreticky umožňuje více než 1M znaků
- Jednoznačnost – libovolná 16bitová (dnes 32bitová) hodnota zastupuje v jakémkoliv kontextu stejný znak
- Několik způsobů reprezentace
 - Např. UTF-8, UTF-16

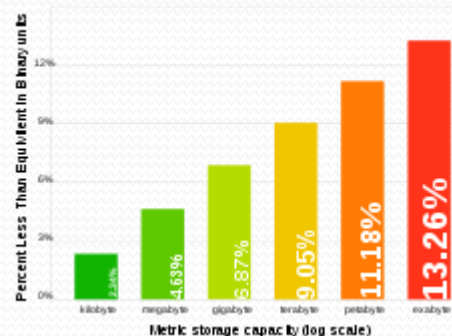
Násobky a užívané předpony

Předpona	Značka	Mocnina	Převod
kilo	k	10^3	1000
mega	M	10^6	1000000
giga	G	10^9	1000000000
tera	T	10^{12}	1000000000000
penta	P	10^{15}	1000000000000000

Předpona	Značka	Mocnina	Převod
kibi	ki	2^{10}	1024
mebi	Mi	2^{20}	1048576
gibi	Gi	2^{30}	1073741824
tebi	Ti	2^{40}	1099511627776
pebi	Pi	2^{50}	1125899906842624

Předpony SI

Comparison of Decimal and Binary Units



Předpony IEC

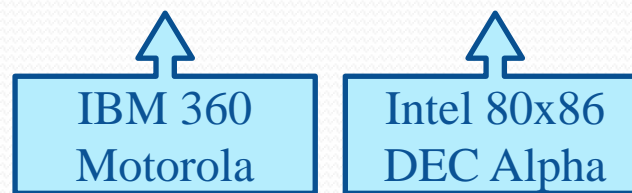
prosinec 1998
dodatek k normě IEC 60027-2

v Česku převzato jako
ČSN IEC 60027-2

Slabiková organizace paměti

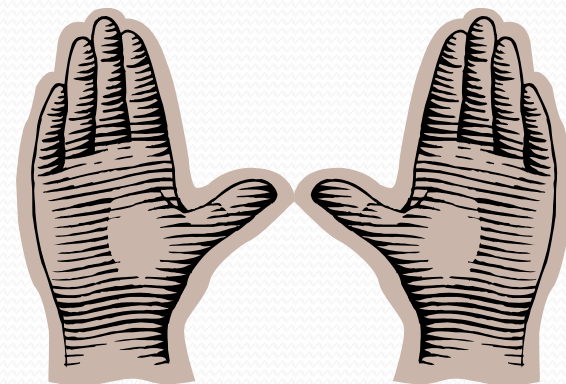
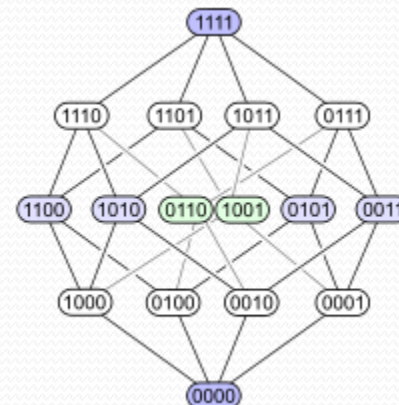
- Od adresy **5678h** má být uložen doubleword **1234ABCD**

Adresa	Big-endian	Little-endian
5678h	12	CD
5679h	34	AB
567Ah	AB	34
567Bh	CD	12



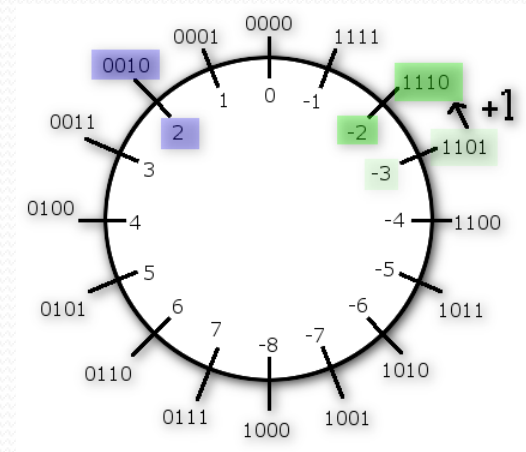
Číselné soustavy

- Určeny bází (základem) z
- Nejčastěji používané soustavy
 - Dvojková (binární) $z=2$
 - Desítková (dekadická) $z=10$
 - Šestnáctková (hexadecimální) $z=16$



Záporná čísla

- Přímý (znaménko a absolutní hodnota)
 - Používá se pro uložení mantisy u IEEE 754 floating point std.
- Aditivní (s posunutou nulou)
 - Přičtená konstanta ke všem číslům
 - Používá se pro uložení exponentu u IEEE 754 floating point std.
- Inverzní
 - Bitová negace
- Doplnkový
 - Bitová negace + přičteme jedničku
 - Jednoduchá implementace operací
 - Používá se pro reprezentaci celých čísel
 - odečtením čísla 00000001 od čísla 00000000 dojde k přetečení,
 - a výsledkem je číslo 11111111.
- Čísla se znaménkem (signed) mají rozsah půlený:
 - např. Integer (16b) $-32768 (-2^{15})$ až $32767 (2^{15}-1)$

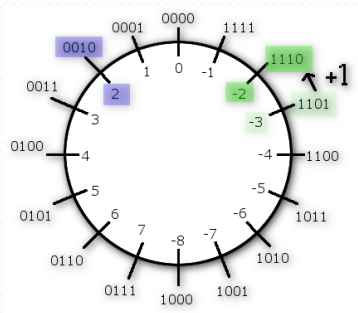


1011	0010
178	(unsigned)
-50	(přímý kód)
50	(aditivní kód)
-77	(inverzní kód)
-78	(doplňkový kód)

Kód:	Bezznaménkový	Přímý	Aditivní	Inverzní	Doplňkový
+15	1111				
+14	1110				
+13	1101				
+12	1100				
+11	1011				
+10	1010				
+9	1001				
+8	1000				
+7	0111	0111	1111	0111	0111
+6	0110	0110	1110	0110	0110
+5	0101	0101	1101	0101	0101
+4	0100	0100	1100	0100	0100
+3	0011	0011	1011	0011	0011
+2	0010	0010	1010	0010	0010
+1	0001	0001	1001	0001	0001
+0	0000	0000	1000	0000	0000
-0		1000		1111	
-1		1001	0111	1110	1111
-2		1010	0110	1101	1110
-3		1011	0101	1100	1101
-4		1100	0100	1011	1100
-5		1101	0011	1010	1011
-6		1110	0010	1001	1010
-7		1111	0001	1000	1001
-8			0000		1000

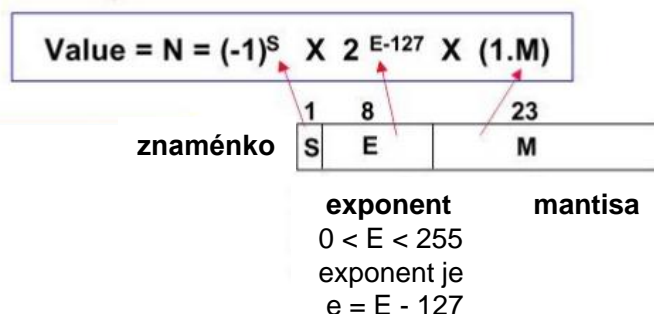
Eight-bit two's complement		
Binary value	Two's complement interpretation	Unsigned interpretation
00000000	0	0
00000001	1	1
⋮	⋮	⋮
01111110	126	126
01111111	127	127
10000000	-128	128
10000001	-127	129
10000010	-126	130
⋮	⋮	⋮
11111110	-2	254
11111111	-1	255

Přetečení rozsahu



Reálná čísla (pohyblivá řádová čárka)

IEEE 754 Floating point standard

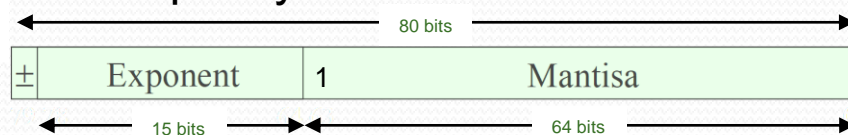


0.15625_{10} je 0.00101_2

$1/8 + 1/32$

$0.00101_2 = 1.01_2 \times 2^{-3}$

Počet platných míst: 18



Double extended precision

Počet platných míst: 16



Double Precision
IEEE 754 Floating-Point Standard

Počet platných míst: 7



Single Precision
IEEE 754 Floating-Point Standard

(IEEE standard for binary floating-point arithmetic, 1985)

převod

$0,1_{10}$ a $0,625_{10}$ převést do dvojkové soustavy:

$$0,1 * 2 = 0,2 \quad 0$$

$$0,2 * 2 = 0,4 \quad 0$$

$$0,4 * 2 = 0,8 \quad 0$$

$$0,8 * 2 = 1,6 \quad 1$$

$$0,6 * 2 = 1,2 \quad 1$$

$$0,2 * 2 = 0,4 \quad 0$$

$$0,625 * 2 = 1,250 \quad 1$$

$$0,250 * 2 = 0,500 \quad 0$$

$$0,500 * 2 = 1,000 \quad 1$$

$$0,625_{10} = 0,101_2$$

$$0,1_{10} \approx 0,0001100110011..._2$$

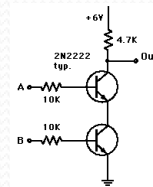
Postup nemusí být konečný!

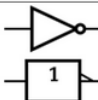
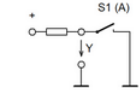
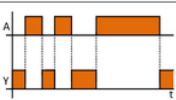

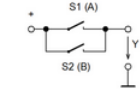
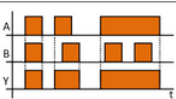
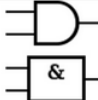
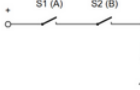
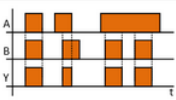

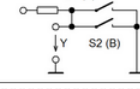
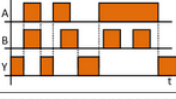
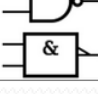
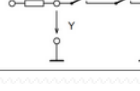



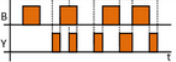
Základ Boolovy algebry

- Činnost moderních počítačů je založená na Boolově algebře
 - logických funkcích a dvojkové číselné soustavě
- *Logické proměnné* mohou nabývat pouze dvou hodnot 0 nebo 1
 - v počítači jsou představovány většinou napětovou úrovní, elektrickým proudem, úrovní magnetizace, ...
- *Logické funkce* mohou nabývat také pouze hodnot 0 nebo 1
 - implementují pomocí *logických členů* neboli *hradel*
 - Na nejnižší úrovni z tranzistorů
 - Logická hradla se sdružují do integrovaných obvodů (IO)
- Z logických členů se při návrhu skládají větší celky, které jsou souhrnně nazývány *logické obvody (kombinační, sekvenční)*

Logické funkce

- Negace (NOT)
- Logický součet (OR)
- Logického součinu (AND)
- Negace logického součtu (NOR)
- Negace logického součinu (NAND)
- Nonekvivalence (XOR)
- Různý počet vstupů
 - 1 až 4, výjimečně 8



Pravdivostní tabulka <table> <tr><th>A</th><th>Y</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	Y	0	1	1	0	Schematická značka 	Realizace pomocí spínačů 	Graf 									
A	Y																	
0	1																	
1	0																	
Pravdivostní tabulka <table> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	Schematická značka 	Realizace pomocí spínačů 	Graf 
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Pravdivostní tabulka <table> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	Schematická značka 	Realizace pomocí spínačů 	Graf 
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
Pravdivostní tabulka <table> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	Schematická značka 	Realizace pomocí spínačů 	Graf 
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Pravdivostní tabulka <table> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	Schematická značka 	Realizace pomocí spínačů 	Graf 
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
Pravdivostní tabulka <table> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	Schematická značka 	Realizace pomocí spínačů 	Graf 
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Von Neumannova architektura

- Ucelená soustava názorů a představ
 - jak by měl počítač fungovat
 - z jakých hlavních částí by se měl skládat
 - co a jak by tyto části měly dělat
 - jak by měly vzájemně spolupracovat ...
- Vnitřní struktura počítače by se neměla nijak měnit v závislosti na zpracovávané úloze
 - Univerzální struktura

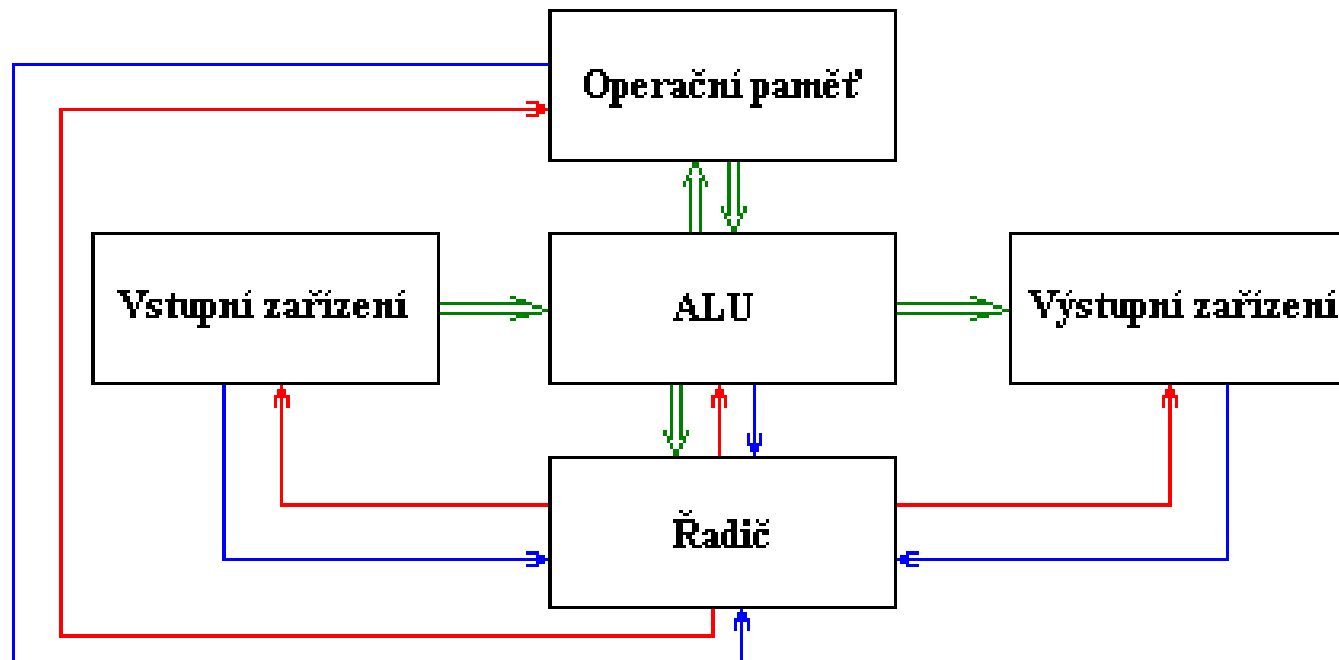
Von Neumannova architektura




- Programy, tak i data jsou ve své podstatě jedno a totéž (posloupnosti nul a jedniček)
 - rozdíl mezi nimi je dán pouze významem
- Programy i jejich data by se měly uchovávat v téže operační paměti.
- Paměť s přímým přístupem
- Sekvenční zpracování

Von Neumannova architektura

- **Operační paměť**
 - slouží k uchování zpracovávaného programu, zpracovávaných dat a výsledků výpočtu
- **ALU - Arithmetic-logic Unit**
 - Jednotka provádějící veškeré aritmetické výpočty a logické operace.
 - Obsahuje sčítačky, násobičky (pro aritmetické výpočty) a komparátory (pro porovnávání)
- **Řadič**
 - Řídící jednotka, která řídí činnost všech částí počítače.
 - Řízení je prováděno pomocí **řídících signálů**, které jsou zasílány jednotlivým modulům.
 - Reakce na řídící signály, stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí **stavových hlášení**
- **Vstupní zařízení**
 - Zařízení určená pro vstup programu a dat.
- **Výstupní zařízení**
 - Zařízení určená pro výstup výsledků, které program zpracoval

Von Neumannova architektura



-  Tok dat
-  Řídicí signály řadiče
-  Stavová hlášení řadiči

Princip činnosti podle von Neumanna

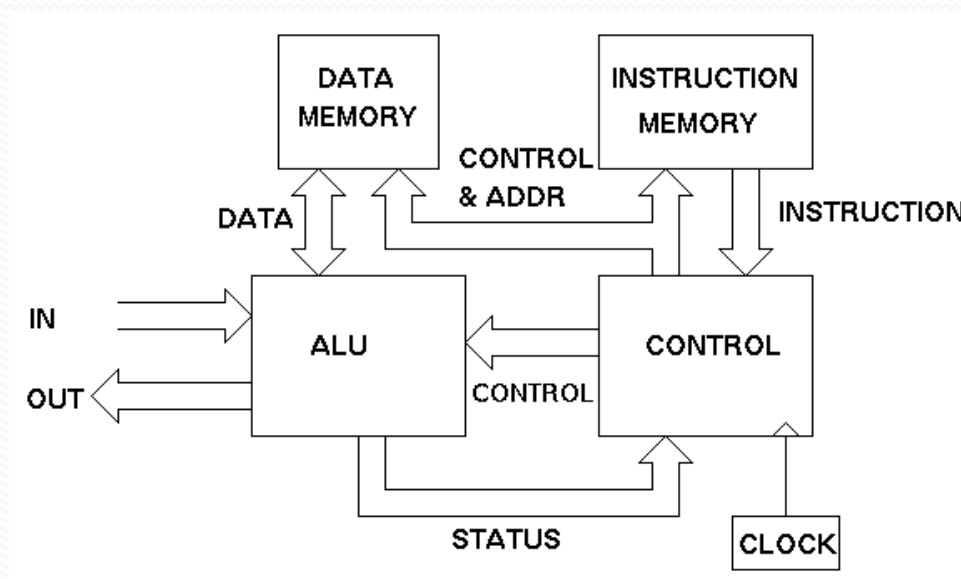
- Do operační paměti se pomocí vstupních zařízení přes ALU umístí program, který bude provádět výpočet.
- Stejným způsobem se do operační paměti umístí data, která bude program zpracovávat
- Proběhne vlastní výpočet, jehož jednotlivé kroky provádí ALU. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače.
 - Mezivýsledky výpočtu jsou ukládány do operační paměti.
- Po skončení výpočtu jsou výsledky poslány přes ALU na výstupní zařízení.

Odlišnosti od von Neumanna

- Podle von Neumannova schématu počítač pracuje vždy nad jedním programem.
 - Toto vede k velmi špatnému využití strojového času.
 - Dnes je obvyklé, že počítač zpracovává paralelně více procesů zároveň - tzv. multitasking
- Počítač může disponovat i více než jedním procesorem
- Existují vstupně / výstupní zařízení (I/O devices), která umožňují jak vstup, tak výstup dat
- Program se do paměti nemusí zavést celý, ale je možné zavést pouze jeho část
 - ostatní lze části zavádět až v případě potřeby

Harvardská architektura

- Fyzicky oddělená paměť pro instrukce a data
 - Paměti mohou být naprosto odlišné
 - Dvojí paměť umožňuje paralelní přístup k oběma paměťem
 - Program je často umístěn v ROM (nelze modifikovat)



von Neumann X Harvardská architektura

- Moderní počítače spojují obě architektury.
 - Celý počítač se chová jako von Neumannova architektura.
 - Načítá data i program z hlavní paměti.
 - Uvnitř procesoru je použita harvardská architektura.
 - Paměť L1 cache se dělí na paměť instrukcí a paměť pro data.

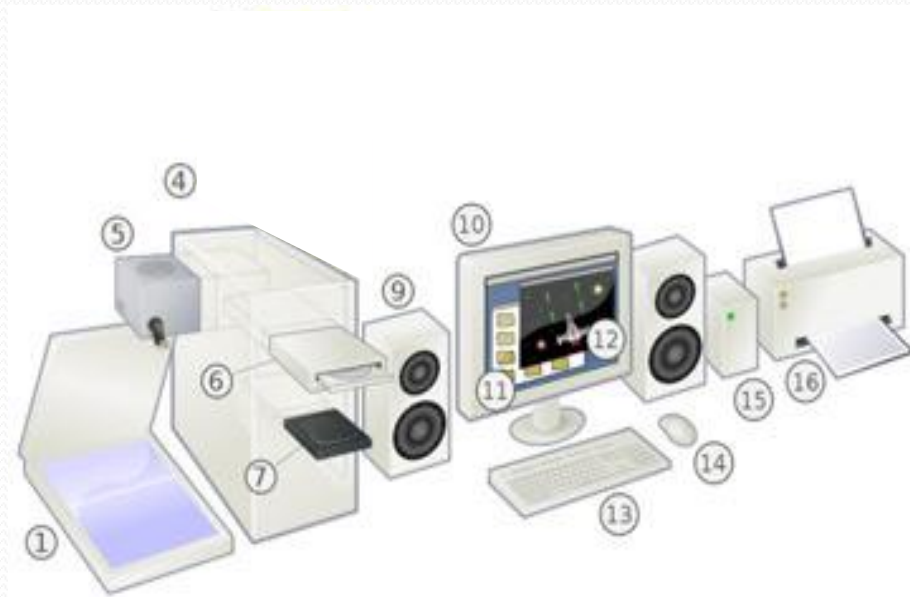
Konfigurace počítače

- Základní část
 - Základní deska
 - Procesor
 - Paměti
 - Mechaniky pružných disků
 - Pevné disky
 - Mechanika CD, DVD, Blu-ray
 - Videokarta (grafická karta)
 - Zvuková karta
 - Síťová karta
 - Další zařízení



Konfigurace počítače

- Periferie
 - Monitor
 - Tiskárna
 - Klávesnice
 - Myš
 - Další zařízení



Konkrétní vybavení daného počítače se liší podle potřeb uživatele a způsobu využití