Návrh číslicových systémů (INC)

Otto Fučík

Vysoké učení technické v Brně Fakulta informačních technologií Božetěchova 2, 612 66 Brno



Použitá literatura

N. Frištacký, M. Kolesár, J. Kolenička a J. Hlavatý: "Logické systémy", SNTL Praha, 1986 M. Eysselt: "Logické systémy", SNTL Praha, skriptum VUT v Brně, 1985 J. F. Wakerly: "Digital Design. Principles and Practices", Prentice Hall, ISBN 0-13-769191-2, 2000 V. P. Nelson, H.T.Nagle, B.D.Carroll, J.D.Irwin: "Digital Logic Circuit Analysis & Design", ISBN 0-13-463894-8, 1995 T.L.Floyd: "Digital Fundamentals", Prentice Hall, ISBN 0-13-080850-4, 2000



Administrativa



- Kontrola znalostí
 - Půlsemestrální zkouška (25 bodů)
 - Projekt (20 bodů) zápočet
 - Semestrální zkouška (55 bodů)
 - Min. 5 bodů z projektu podmínka nutná pro získání zápočtu

Pravidla

- Pro získání bodů ze semestrální zkoušky je nutné ji vypracovat tak, aby byla hodnocena nejméně 25 body (z celkem 55 bodů); v opačném případě bude přiděleno 0 bodů
- Pokud bude odhaleno plagiátorství nebo nedovolená spolupráce na projektech, příslušné body nebudou uděleny a bude zváženo zahájení disciplinárního řízení

Návrh číslicových systémů: Historie



- 1822 (Ch. Babbage)
 - Informace mohou být reprezentovány čísly
- 1854 (G. Boole)
 - Matematický aparát umožňující efektivní práci s dvoustavovými (binárními, 0 a 1) funkcemi, výrazy a jejich algebrou
 - Umožňuje systematický návrh a optimalizaci základních stavebních prvků číslicových systémů – tzv. logických obvodů
- 1904 (E. V. Huntington)
 - Rozvinutí a doplnění Booleovy algebry
- 1938 (C. E. Shannon)
 - Využití Booleovy algebry pro návrh log. obvodů (diplomová práce)
 - Použití relé pro realizaci logických operací sepnuto a rozepnuto (0 a 1)

```
1st. Disjunctive Syllogism.
Either X is true, or Y is true (exclusive),
                                              x + y - 2xy = 1
Therefore Y is not true,
Either X is true, or Y is true (not exclusive),
                                               x+y-xy=1
But X is not true,
Therefore Y is true.
  2nd, Constructive Conditional Syllogism.
       If X is true, Y is true,
                                     x(1-y) = 0
       But X is true,
                                        x = 1
      Therefore Y is true.
                                1 - y = 0 or y = 1.
  3rd. Destructive Conditional Syllogism.
         If X is true, Y is true,
                                       x(1-y)=0
        But Y is not true,
        Therefore X is not true.
  4th. Simple Constructive Dilemma, the minor premiss ex-
  If X is true, Y is true,
                                       x(1-y)=0, (41),
  If Z is true, Y is true,
                                       z(1-y)=0, (42),
  But Either X is true, or Z is true, x+z-2xz=1, (43).
  From the equations (41), (42), (43), we have to eliminate
z and z. In whatever way we effect this, the result is
whence it appears that the Proposition Y is true.
```

	whence it appears that the Proposition	on 1 is true.
Anelo	ogsie Batween the Calculus of	Propositions
	and the symbolic Folay And	ilysis
Symbel	Interpretation in relay	Interpretation in the Galmulus of Propositions
х	The circuit X.	The proposition X.
0	The circuit is closed.	The proposition is false.
1	The ciruit is open.	The proposition is true.
x + y	The series connection of circuits X and Y	The proposition Which is true if elther X or Y is true.
XY	The perelled connection of circuits X and Y	The proposition which is true if both X and Y are true.
χı	The circuit which is open when X is closed, and closed when X is open.	The contradictory of proposition X.
	The circuits open and close simultaneously.	Ruch proposition implies the other.

Návrh číslicových systémů: Rozvoj elektroniky



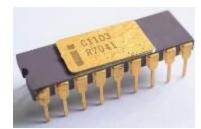
 Efektivní tvorba (rozměry, spolehlivost, příkon) elektronických obvodů pracujících s binárními hodnotami



- 1906 (L. de Forest)
 - Vynález elektronky (zesilování signálů)
- 1947 (W. B. Shockley, J. Bardeen a W. H. Brattain)
 - Vytvoření tranzistoru (rozměry, příkon, cena, spolehlivost)
- 1958 (J. Kilby)
 - Vynález integrovaných obvodů (IO) umožnil umístění mnoha tranzistorů (dnes miliardy) na polovodičovou destičku
- 1966 (R. Dennard)
 - Vynález pamětí DRAM realizace spolehlivých a rychlých elektronických pamětí s velkou kapacitou
- Budoucnost? (nanotechnologie, kvantové jevy...)







Mooreův zákon: Predikce z roku 1965



- Exponenciální růst složitosti čipů
- "Počet tranzistorů, které mohou být umístěny na integrovaný obvod, se při zachování stejné ceny zdvojnásobí zhruba každých 18 měsíců"
- Dnešní technologie výroby umožňují integrovat miliardy tranzistorů v IO
 - Umíme je efektivně využít?

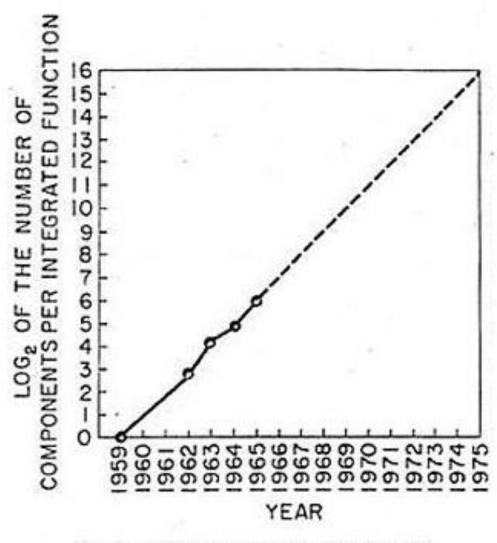


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

 $[Zdroj: http://www.intel.com/pressroom/kits/events/moores_law_40th/index.htm?iid=tech_mooreslaw+body_presskit]$

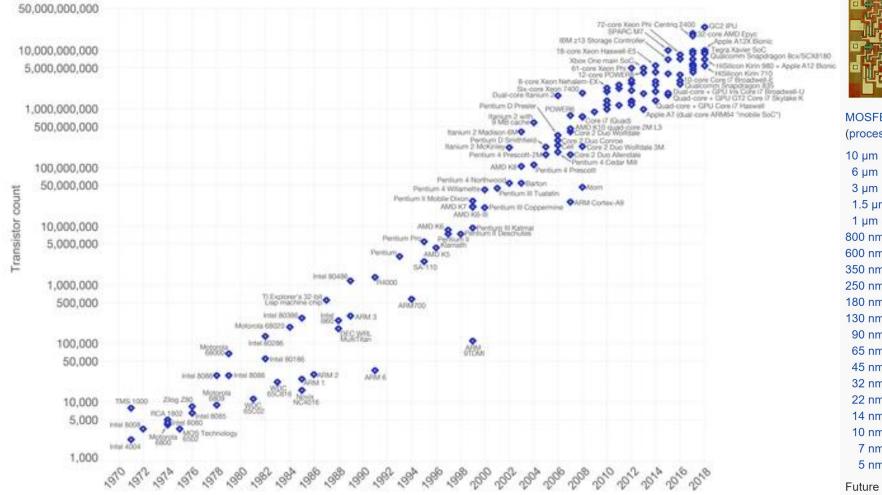
Mooreův zákon: Stav v roce 2018



Moore's Law – The number of transistors on integrated circuit chips (1971-2018)



Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress - such as processing speed or the price of electronic products - are linked to Moore's law.



Semiconductor device fabrication



MOSFET scaling (process nodes)

 $10 \, \mu m - 1971$ 6 µm - 1974 3 um - 1977 $1.5 \, \mu m - 1981$ $1 \, \mu m - 1984$ 800 nm - 1987 600 nm - 1990 350 nm - 1993 250 nm - 1996 180 nm - 1999 130 nm - 2001 90 nm - 2003 65 nm - 2005 45 nm - 200732 nm - 2009 22 nm - 2012 14 nm - 2014

5 nm - 2020

3 nm - ~2021 2 nm - ~2024

10 nm - 2016

7 nm - 2018

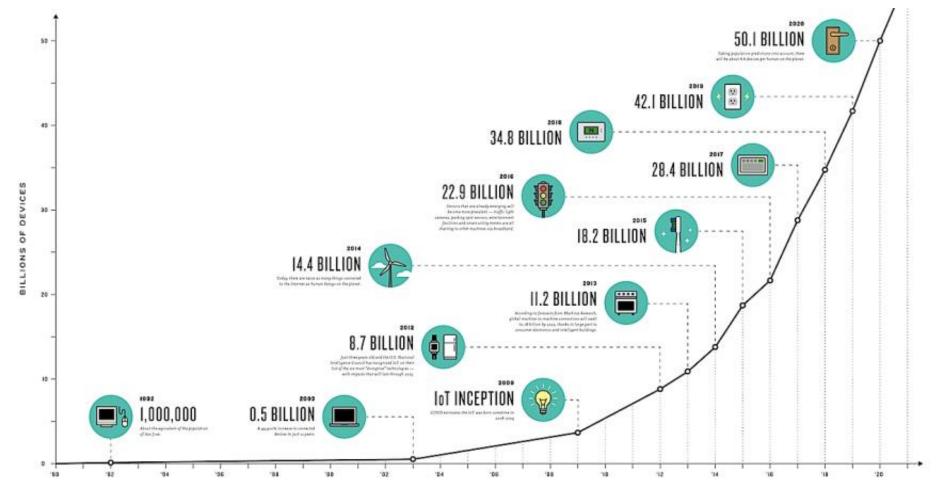
Data source: Wikipedia ihttps://en.wikipedia.org/wiki/Transistor_count) The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

Nejen složitost komponent roste, ale i jejich počet



- Internet věcí (Internet of Things IoT)
 - Propojené vestavěné systémy pro sběr a výměnu dat



[Zdroj: J. Straw: http://idisrupted.com/disrupted-electronics-internet-things-may-create-moores-law-steroids/]

Digitální revoluce



LAW OF ACCELERATING RETURNS

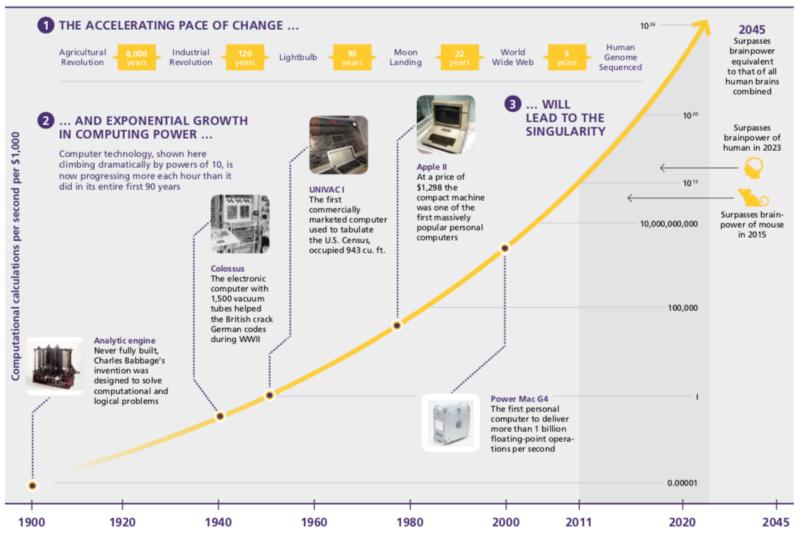
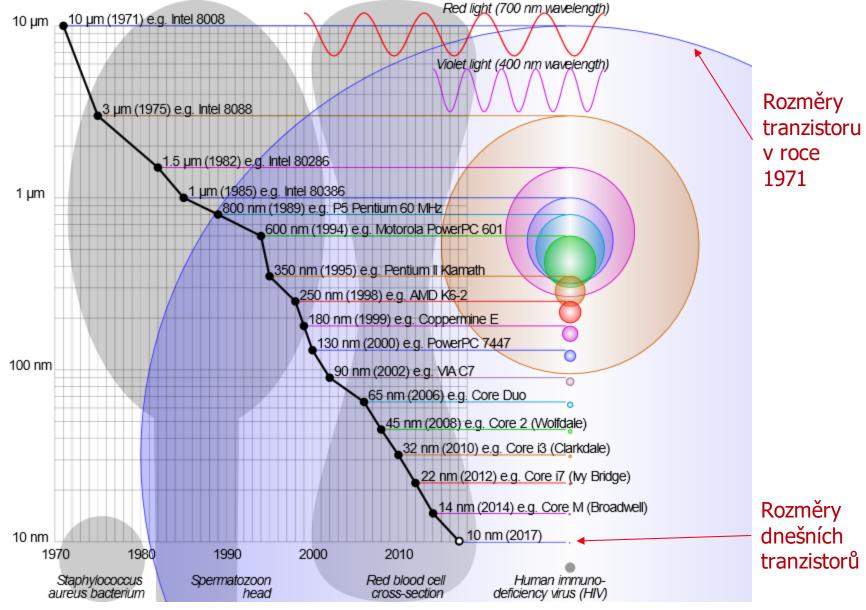


Figure 12: Ray Kurzweil's Law of Accelerating Returns depicts the exponential growth of computer processing power and technology innovations throughout history, and anticipates computers will exceed human intelligence in the future; Source: TIME / Wikipedia

source dhl via @mikequindazzi

Mooreův zákon: Ilustrace zmenšování rozměrů





[Zdroj:By Cmglee - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=16991155]

Důsledky Mooreova zákona

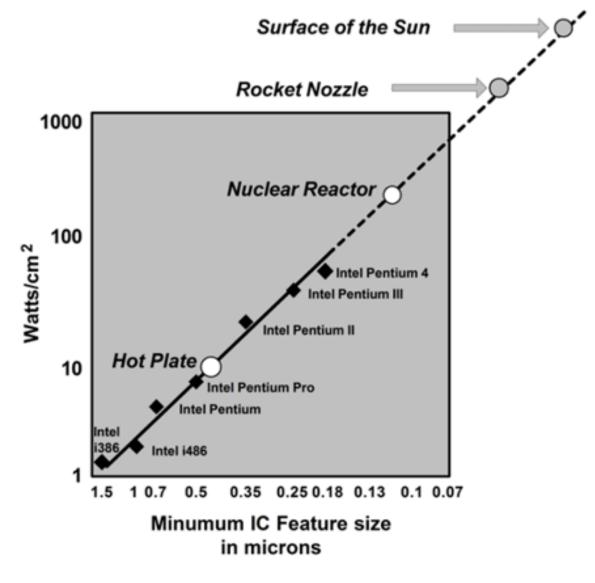


- Roste složitost počet tranzistorů na jednotku plochy se zdvojnásobuje každého cca 1,5 roku
- Roste výkonnost např. v roce 2013 měl běžný laptop srovnatelný výpočetní výkon s nejvýkonnějším počítači poloviny 90tých let
- Klesá energetická náročnost počet výpočtů na kWh se zdvojnásobuje každého cca 1,5 roku
- Klesá cena cena posledních modelů počítačů je téměř konstantní, přičemž jejich výkonnost roste
- Rostou investice do výroby čipů zdvojnásobení každé cca 4 roky

Limity: Využitelná výkonnost a odvod tepla



- Efektivní (užitečná) výkonnost
 - Roste přibližně s druhou odmocninou jejich složitosti (Pollackovo pravidlo)
- Hustota energie (Power Density) roste s výkonností
 - Tepelné ztráty limitují výpočetní výkonnost (odvod tepla)

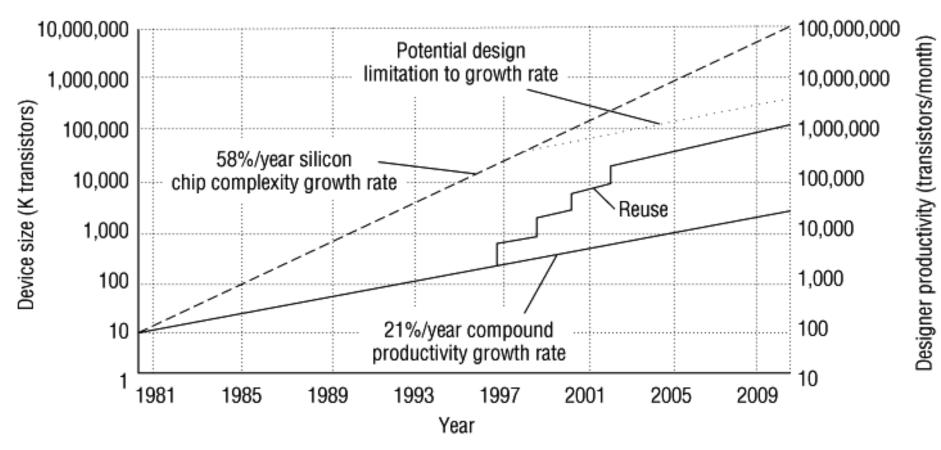


[Zdroj: R. Ronen, A. Mendelson, K. Lai, Shih-Lien Lu, F. Pollack and J. P. Shen, "Coming challenges in microarchitecture and architecture," in Proceedings of the IEEE, vol. 89, no. 3, pp. 325-340, March 2001]

Limity: Produktivita práce při návrhu



- Složitost integrovaných obvodů roste o cca 60 % ročně
- Produktivita práce pouze o cca 20 % ročně



[Zdroj: B. Smith, "Burton Smith's Multithreaded Success Strategy" in IEEE Design & Test of Computers, vol. 16, no. 04, pp. 7-13, 1997]

Číslicové vs. analogové systémy



- Analogové systémy nejsou pro řešení některých úloh použitelné
 - Např. nelineární funkce (šifrování apod.)
- Číslicové systémy lze programovat pro vykonávání libovolného vyčíslitelného algoritmu (omezeno velikostí paměti a dobou výpočtu)
- Informace může být ve fyzicky realizovaném číslicovém systému reprezentována s mnohem větší přesností a ve větším rozsahu hodnot než v analogovém (signál/šum)
 - Přesnost a rozsah hodnot mohou být teoreticky libovolné (omezeno velikostí paměti a dobou výpočtu)
- Činnost analogových obvodů je v praxi výrazněji limitována řadou fyzikálních veličin
 - Šum, teplotní výkyvy, stárnutí součástek apod.

Číslicové vs. analogové systémy



- Informace reprezentovaná číslicově se lépe ukládá a čte
- Číslicové obvody umožňují realizovat detekci chyb a jejich opravu
- Pro stejnou posloupnost vstupních hodnot produkuje číslicový systém vždy stejné výsledky
 - Např. viz klasická gramofonová deska vs. kompaktní disk
- Návrh číslicových systémů (nazývaných též "logické systémy") pracuje pouze s dvouhodnotovými veličinami a logickými vztahy mezi nimi (Booleova algebra)
 - Číslicové systémy lze navrhovat, analyzovat a realizovat se znalostí relativně jednoduchých principů (náplň tohoto kurzu)
 - Návrh analogových obvodů vyžaduje hlubokou znalost funkce (matematických modelů) použitých součástek

Návrh složitých systémů



- Číslicové systémy
 - Hierarchicky uspořádaná struktura, ve které jednotlivé subsystémy (komponenty) přenáší přes rozhraní informaci pomocí komunikačního média (vodiče)
 - S ohledem na zjednodušení návrhu je výhodné, když komponenty a komunikace mezi nimi je co nejjednodušší => 1bitové hodnoty
 - Proto tvoříme číslicové systémy sestavené z komponent, které pracují a komunikují pouze s 1bitově reprezentovanou informací (nejjednodušší možná forma = jednoduché, spolehlivé, levné...)
 - Umožňují efektivní tvorbu výpočetních strojů (počítače), které zpracovávají binárně kódované informace pomocí logických obvodů (Booleova algebra)

Booleova algebra [G. Boole, 1854]



- 1. Distributivní komplementární svaz
 - Obsahuje alespoň dva prvky
- 2. Šestice (B, +, ', ', 0, 1)
 - B neprázdná množina s alespoň dvěma různými prvky
 - + logický součet (binární operace)
 - logický součin (binární operace)
 - ' komplement (unární operace)
 - 0 nejmenší (nulový) prvek (infimum)
 - 1 největší (jedničkový) prvek (supremum)
 - Definuje množinu prvků, množinu operátorů, axiomy (postuláty) a teorémy (věty)
- Dvouhodnotová Booleova algebra
 - Axiomy a teorémy Booleovy algebry (1854) jsou definovány obecně
 - My se omezíme na algebru, ve které logické proměnné a výsledky logických funkcí mohou nabývat pouze hodnot 0 a 1 (0≠1)

Logické operace



- Základní
 - Logický součet (disjunkce, spojení, sjednocení, OR): $x \lor y$, x + y
 - Logický součin (konjunkce, průsek, průnik, AND): $x \wedge y$, $x \cdot y$
 - Negace (inverze, doplněk, komplement, NOT): χ' , χ , χ , χ
- Shefferova funkce (negace log. součinu, NAND): $x \uparrow y$, $x \cdot y$
- Pierceova funkce (negace log. součtu, NOR): $x \downarrow y$, $\overline{x+y}$
- Exkluzivní log. součet: x ⊕ y, x ≡ y, x · y + x · y
 Pro 2 proměnné též nonekvivalence (součet modulo 2)
- Ekvivalence (totožnost, rovnost, XNOR): $x \Leftrightarrow y, x \equiv y, x \cdot y + x \cdot y$
- Implikace: $x \Rightarrow y, x \rightarrow y, x + y$
- Inhibice (negace implikace) $x \Rightarrow y, x \leftrightarrow y, x \cdot y$
- Kontradikce: výsledkem je konstanta 0 (nezávisle na vstupech)
- Tautologie: výsledek je konstanta 1 (nezávisle na vstupech)
- Logické operace jsou realizovány logickými členy

Princip duality



 Pokud platí nějaké tvrzení, tak platí i duální tvrzení, které vznikne vzájemnou záměnou operací "+" a "·" a prvků 0 a 1

$$0 \rightarrow 1 \quad 1 \rightarrow 0 \quad "+" \rightarrow "\cdot" \quad "\cdot" \rightarrow "+"$$

• Příklad:

$$a + (b \cdot c) = (a+b) \cdot (a+c) \rightarrow a \cdot (b+c) = a \cdot b + a \cdot c$$

- Poznámka
 - Pokud platí jisté tvrzení, není třeba dokazovat tvrzení duálního tvrzení

Logické proměnné, funkce a výrazy



- Logické proměnné x₁, x₂,..., x_n
 - Nabývají hodnot 0 a 1 (logické konstanty)
 - Budeme značit písmeny: A, B, C,..., a, b, c, ..., x₁, x₂, x₃...
 - Poznámka: pro jednodušší zápis budeme logický součin značit jak symboly "X·Y", tak "XY" – pozor na záměnu s proměnnou "XY"
- Logická funkce $f(x_1, x_2, ..., x_n)$ je zobrazení $f: \{0, 1\}^n \rightarrow \{0, 1\}$
 - Hodnota log. funkce je buď 1, nebo 0, v závislosti na hodnotách (0, 1) jednotlivých proměnných x₁, x₂,...,x_n
 - Pro n proměnných (každá má 2 možné hodnoty) máme 2ⁿ možností, jak jim přiřadit hodnoty
 - Existuje celkem 2^{2^n} různých log. funkcí *n* proměnných
- Logický výraz (řetězec symbolů)
 - Obsahuje log. konstanty, log. proměnné a log. operátory
 - Log. výrazy lze upravovat a zjednodušovat s využitím Booleovy algebry – cílem je splnění daných kritérií (cena, rychlost, příkon atd.)

Reprezentace log. funkcí: Pravdivostní tabulka



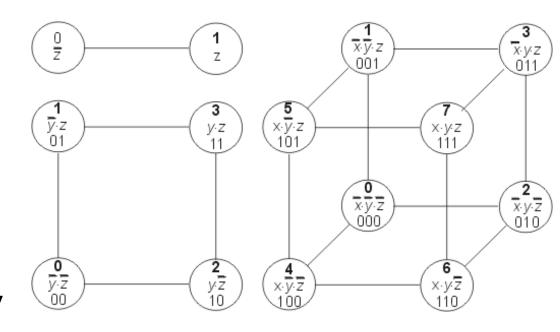
- Normální báze (x, y, z)
 - Proměnným jsou přiřazeny váhy mocnin báze 2
- Log. funkce F(x, y, z)
 - Vstupní stav kombinace vstupních log. proměnných (vstupů)
 - Stavový index s desítkové číslo udávající hodnotu log. stavu
 - Neurčený stav X (anglicky don't care)
 - Stav, kdy není třeba určit, zda při daném vstupním stavu má pravdivostní hodnota funkce F(x, y, z) hodnotu 0, nebo 1
 - Log. funkce určená pro všechny možné vstupní stavy existuje jednoznačné určení pravdivostních hodnot log. funkce F(x, y, z)

stavový index	V	stupi stav		pravdivostní hodnoty		
S	4	2	1	F(x, y, z)		
	X	У	Z			
0	0	0	0	0		
1	0	0	1	1		
2	0	1	0	0		
3	0	1	1	1		
4	1	0	0	0		
5	1	0	1	X		
6	1	1	0	1		
7	1	1	1	X		

Reprezentace log. funkcí: Jednotková krychle



- Neorientovaný graf
- Reprezentace log. funkce N proměnných
 - S více než čtyřmi rozměry se špatně pracuje
- Uzly
 - Reprezentují hodnoty vstupního stavu
 - Počet uzlů je 2^N
- Hrany
 - Spojují uzly, které se liší pouze v jedné proměnné (sousednost)



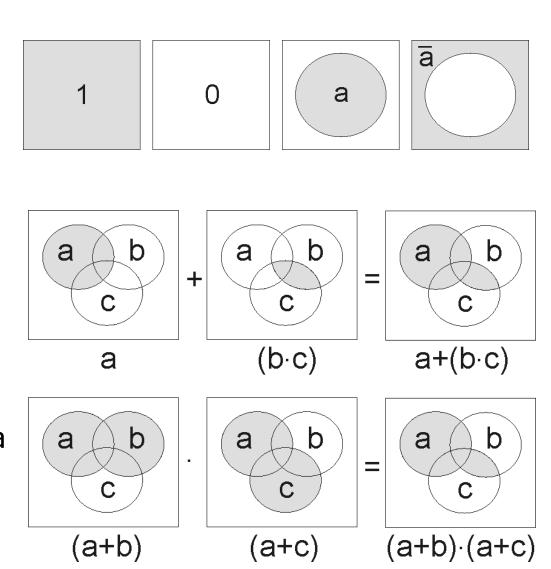
- Význam
 - Proměnnou, ve které se liší uzly krychle, lze eliminovat
 - Např.: pro dvourozměrnou krychli a stavy 3 (binárně 11) a 2 (binárně 10) p<u>la</u>tí:

$$yz + yz = y$$

Reprezentace log. funkcí: Vennovy diagramy



- Prvky množin jsou znázorněny jako uzavřené plochy
- Log. součet = sjednocení příslušných ploch
- Log. součin = průnik příslušných ploch
- Příklad:
 - Reprezentace distributivního zákona pomocí tzv.
 Vennových diagramů



Všechny logické funkce 2 proměnných



Log. výrazy v disjunktní formě a pravdivostní tabulka

$$f_{0} = 0 f_{8} = (x_{1} \cdot x_{2}) f_{9} = (\overline{x}_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{10} = (x_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{10} = (x_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{10} = (x_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{11} = (\overline{x}_{1} \cdot \overline{x}_{2} + x_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{12} = (\overline{x}_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) f_{13} = (\overline{x}_{1} \cdot \overline{x}_{2} + \overline{x}_{1} \cdot x_{2} + x_{1} \cdot x_{2}) f_{13} = (x_{1} \cdot \overline{x}_{2} + \overline{x}_{1} \cdot x_{2} + x_{1} \cdot x_{2}) f_{14} = (x_{1} \cdot \overline{x}_{2} + \overline{x}_{1} \cdot x_{2} + x_{1} \cdot x_{2}) f_{15} = (\overline{x}_{1} \cdot \overline{x}_{2} + x_{1} \cdot \overline{x}_{2} + x_{1} \cdot x_{2}) = 1$$

X ₂	X ₁	f ₀	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉	f ₁₀	f ₁₁	f ₁₂	f ₁₃	f ₁₄	f ₁₅
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Všechny logické funkce 2 proměnných



Vennovy diagramy, logické funkce a operace



Kontradikce

$$f_0 = 0$$



Pierceova funkce (NOR)

$$f_1 = (\bar{x}_1 \cdot \bar{x}_2) = \overline{x_1 + x_2} = (x_1 \downarrow x_2)$$



Inhibice

$$f_2 = (x_1 \cdot \overline{x}_2) = (\overline{x_1} \Longrightarrow x_2)$$



Negace (NOT)

$$f_3 = (\bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot \bar{x}_2) = \bar{x}_2$$



Inhibice

$$f_4 = (\overline{x}_1 \cdot x_2) = (\overline{x_2 \Longrightarrow x_1})$$



Negace (NOT)

$$f_5 = (\overline{x}_1 \cdot \overline{x}_2 + \overline{x}_1 \cdot x_2) = \overline{x}_1$$



Exkluzívní součet (XOR)
$$f_6 = (x_1 \cdot \overline{x}_2 + \overline{x}_1 \cdot x_2) = (x_1 \oplus x_2)$$



Shefferova fce (NAND)
$$f_7 = (\bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2) = (\bar{x}_1 \cdot \bar{x}_2) = (x_1 \uparrow x_2)$$

Všechny logické funkce 2 proměnných



Vennovy diagramy, logické funkce a operace

Log. součin (AND)
$$f_8 = (x_1 \cdot x_2)$$

Ekvivalence (XNOR)
$$f_9 = (\bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot x_2) = (x_1 \Leftrightarrow x_2)$$

Identita
$$f_{10} = (x_1 \cdot \overline{x}_2 + x_1 \cdot x_2) = x_1$$

Implikace
$$f_{11} = (\bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot \bar{x}_2 + x_1 \cdot x_2) = (x_1 + \bar{x}_2) = (x_2 \Rightarrow x_1)$$

Identita
$$f_{12} = (\bar{x}_1 \cdot x_2 + x_1 \cdot x_2) = x_2$$

Implikace
$$f_{13} = (\bar{x}_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 + x_1 \cdot x_2) = (\bar{x}_1 + x_2) = (x_1 \Longrightarrow x_2)$$

Log. součet (OR)
$$f_{14} = (x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 + x_1 \cdot x_2) = (x_1 + x_2)$$

Tautologie
$$f_{15} = (\bar{x}_1 \cdot \bar{x}_2 + x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 + x_1 \cdot x_2) = 1$$

Logické členy OR, AND a NOT



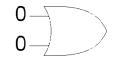
- Definiční tabulka
 - Logický součet

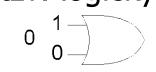
+	0	1
0	0	1
1	1	1

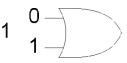
Logický součin

•	0	1
0	0	0
1	0	1

Logické operace se realizují tzv. logickými členy



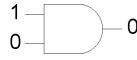






• "·" ... log. člen AND o





- Chování log. členů lze definovat např. pravdivostní tabulkou
 - OR

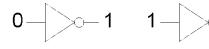
а	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

AND

а	b	a·b
0	0	0
0	1	0
1	0	0
1	1	1

NOT

а	not(a)
0	1
1	0



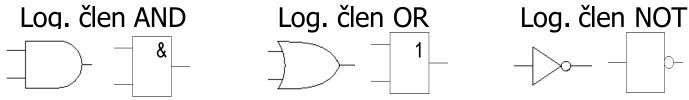
Logické členy



- Pomocí základních log. členů AND, OR a NOT lze realizovat libovolný logický obvod a tedy i číslicový systém (viz dále)
 - Log. funkce AND a OR jsou (s použitím log. funkce NOT) komplementární (lze je vhodným způsobem vzájemně nahradit)
 - Dokonce stačí použít log. členy NAND a NOR pouze se dvěma vstupy (NAND = AND s invertorem na výstupu, nebo NOR = OR s invertorem na výstupu)

Značení

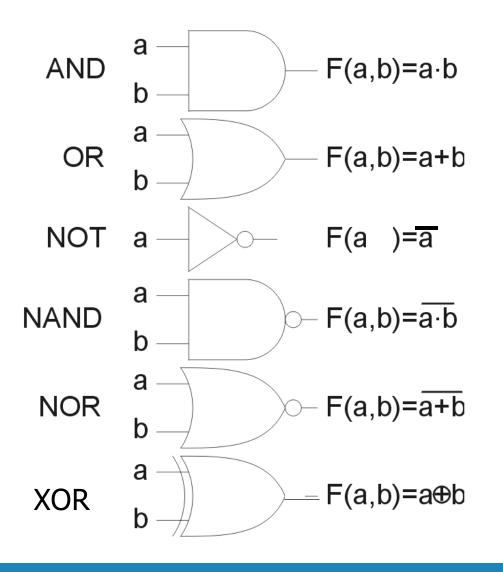
- Čtvercové značky funkce logického členu je označena znaky "&" pro funkci AND, "1" pro funkci OR
- Značky složené z křivek rozšířené ve většině profesionálních systémů pro návrh logických obvodů (každý způsob značení má své výhody a nevýhody)

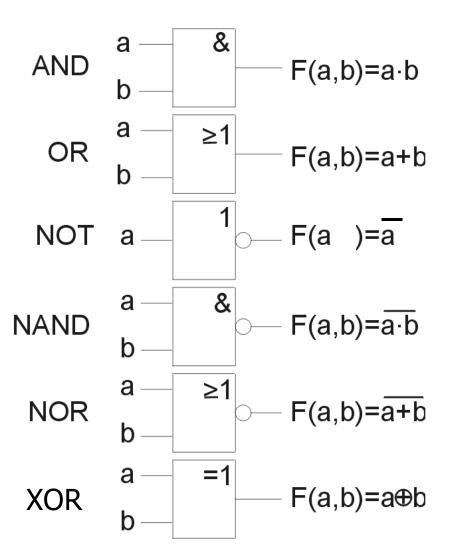


Logické členy: Schematické značky (symboly)



- Standard IEEE/ANSI Std 91-1984
- Standard IEC 60617-12





Logické obvody



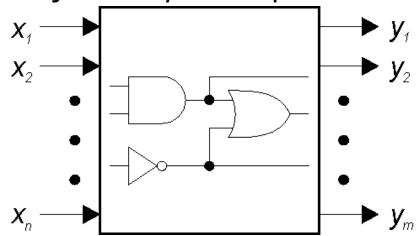
- Číslicové systémy
 - Sestavujeme z obvodů, které navrhujeme za použití Booelovy algebry (práce s logickými výrazy)
 - Proto těmto obvodům říkáme též logické obvody
- Logické obvody dělíme do dvou skupin dle jejich chování
 - Kombinační logické obvody
 - Sekvenční logické obvody
- Kombinační i sekvenční logické obvody
 - Skládají se ze stejných elementárních prvků, tzv. logických členů
- Logické členy
 - Logické členy mají jeden či více vstupů a jeden výstup
 - Hodnota na výstupu log. členu je funkcí hodnot vstupních
 - Log. členy se též nazývají "hradla" (anglicky gate)

Kombinační logické obvody



- Hierarchicky uspořádaná struktura, ve které jednotlivé komponenty zpracovávají a mezi sebou přenáší informaci reprezentované v binární formě (log. úrovně)
 - Každá komponenta má kombinační chování
 - Vstup každé komponenty je připojen pouze k jednomu výstupu předchozí komponenty nebo ke zdroji log. "0" či "1" (nelze spojovat výstupy - není jasné, který výstup je platný)
 - Pozn.: tzv. montážní logické členy tuto podmínku neporušují, viz dále

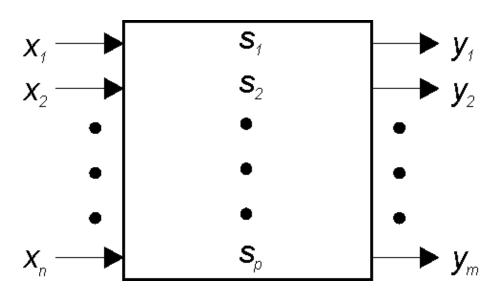
- Struktura neobsahuje cykly (zpětné vazby)
- Funkční a časové chování lze odvodit z funkčního a časového chování jednotlivých komponent



Sekvenční logické obvody



- Hodnoty výstupních proměnných jsou závislé nejen na aktuální kombinaci hodnot vstupních proměnných, ale též na předchozích hodnotách vstupních proměnných a počátečním stavu
 - Říkáme, že sekvenční obvody mají "pamět"
 - Praktická realizace paměti stavu může být různá
- Aktuální hodnoty stavových proměnných
 - Uchovávají veškerou informaci o minulosti, která je potřebná pro stanovení budoucího chování obvodu



Kvantifikace informace (C. Shannon, 1948)



- Entropie (míra neurčitosti) vs. informace (míra určitosti)
 - S rostoucí mírou informace klesá míra entropie
 - Nejjednodušší případ je situace, kdy volíme ze dvou možností
 - Pokud se dozvíme, že jedna ze dvou možností platí, dostaneme elementární množství informace - bit (binary digit – binární číslo)
 - Binární informace je měřena v bitech (b) počet jedniček nebo nul potřebných pro zakódování daných N možností
 - Množství informace pokud máme N možností a nějaký fakt zúží počet možností na M, pak platí, že množství informace I = log₂(N/M) [bitů]
- Příklad
 - Hod mincí: $I_1 = log_2(2/1) = 1 b$
 - Hod dvěma kostkami: $I_2 = log_2(6.6/1) = 5.2 b$

Kódování informace



- Kódování = přidělení jisté reprezentace konkrétní informaci
 - Volba kódu výrazně ovlivňuje vlastnosti implementace příslušného systému (netriviální úloha)
- Problematika zahrnuje
 - Počet potřebných bitů (množství komponent cena a spolehlivost systému)
 - Rychlost manipulace s bity (výkonnost systému)
 - Energetické nároky (změny hodnot odebírají nejvíce energie)
 - Délka (fixní, proměnná)
 - Čísla (bez a se znaménkem, pevná a plovoucí řádová čárka)
 - Komprese (ztrátová, bezztrátová)
 - Šifrování, autorizace
 - Detekce, oprava chyb (redundance, dostupnost)
 - Odolnost proti rušení atd.

Fyzická reprezentace binární informace



- Reprezentace
 - Abstraktní forma pouze hodnoty 0 a 1
 - Reálná forma "nízká" hodnota = 0, "vysoká" hodnota = 1
- Běžně dostupné technologie elektronické obvody
 - Pracují s elektrickými veličinami napětí, fáze, proud atd.
 - Elektrické napětí dnes dominuje (lze snadno generovat i měřit, existuje dlouhodobá zkušenost)
- Hodnota napětí je ovlivněna
 - Nepřesnostmi při jeho generování i měření (reálný svět není diskrétní, ale spojitý)
 - Rušením, výrobními tolerancemi, prostředím (teplota) apod.
- Aby reálný číslicový systém pracoval spolehlivě, musí:
 - Tolerovat určitou chybu ("nízká" a "vysoká" hodnota)
 - Se chovat jako by byl diskrétní (abstrakce 0 a 1)

Reprezentace binární informace: Napětí



- Logické úrovně (pozitivní logika)
 - "Nízká", nula hodnota napětí reprezentující abstraktní "0"
 - "Vysoká", jednička hodnota nap. reprezentující abstraktní "1"
- Fyzická reprezentace logických úrovní
 - Obvody, které spolehlivě (nelze dodržet absolutně) zajistí, že není možno (za normálních podmínek) zaměnit "0" za "1"
- Realizace
 - Obvod má definován "ochranný" interval napětí mezi hodnotami reprezentujícími "0" a "1" (tzv. zakázaná oblast)
 - Napětí v zakázané oblasti není reprezentováno ani jako "0", ani jako "1" = nejsou zde definovány platné logické úrovně
 - Příklad:

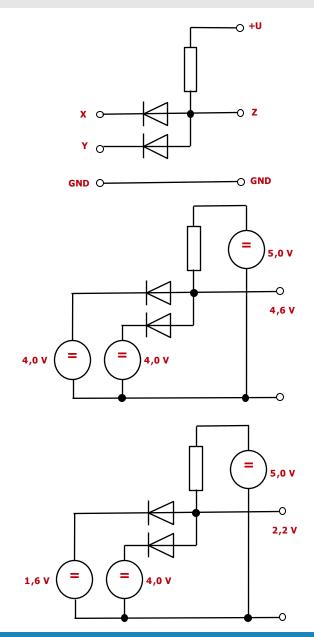


Diodové logické členy



- Realizace hradla AND
- Ilustrace činnosti
 - Oba vstupy jsou na H
 - Výstup je též na úrovni H
- Ilustrace činnosti
 - Jeden vstup je na L
 - Druhý vstup je na H
 - Výstup je na úrovni L
- Pravdivostní tabulka
 - Pomocí hodnot L a H
 - Pomocí hodnot 0 a 1

V_X	V_Y	V_{Z}	Х	Υ	Z
low	low	low	0	0	0
low	high	low	0	1	0
high	low	low	1	0	0
_	high	_	1	1	1

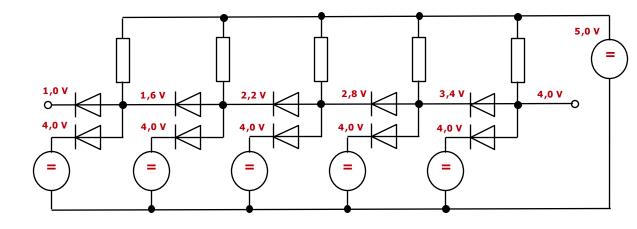


Diodové logické členy



- Tabulka log. úrovní pro napájení 5 V
- Příklad realizace složitějších funkcí pomocí diodových log. členů
 - Kaskáda hradel AND
- Poznámka
 - Díky úbytkům napětí na diodě roste hodnota log. 0 => nelze zapojit více členů za sebe

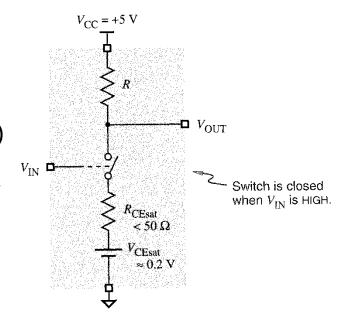
Signal Level	Designation	Binary Logic Value
0–2 volts	LOW	0
2–3 volts	noise margin	undefined
3–5 volts	HIGH	1

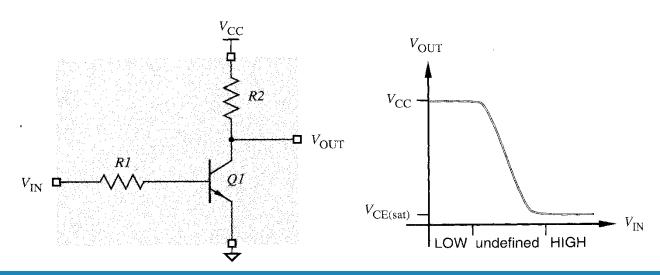


Bipolární tranzistor: Použití při realizaci invertoru



- Náhradní schéma tranzistoru jako spínače
 - R přídavný rezistor (pull-up)
 - R_{CEsat} odpor tranzistoru mezi kolektorem a emitorem při jeho plném otevření (v saturaci)
 - V_{CEsat} zbytkové napětí tranzistoru mezi kolektorem a emitorem při jeho plném otevření (v saturaci)
- Schéma invertoru
 - Přenosová charakteristika





Komunikace binární informace: Rušení



- Informace (logické úrovně) se v číslicovém systému přenáší vodiči
 - Ve vodiči se může indukovat rušivé napětí, které se přičítá k
 platné logické hodnotě => hodnota napětí se může dostat do
 zakázané oblasti a obvod pak nepracuje tak, jak je očekáváno
- Číslicové obvody proto musí být navrženy tak, aby odolávaly rušení z různých vnějších i vnitřních (viz dále) zdrojů
 - Hovoříme o tzv. elektromagnetické kompatibilitě jedna ze základních vlastností všech elektronických zařízení
- Každé elektronické zařízení musí být (dokonce ze zákona) certifikováno tak, aby platilo
 - Nelze jej "zarušit" pracuje správně i při povolené úrovni elektromagnetického rušení daného pracovního prostředí
 - Samo neruší ostatní zařízení nevyzařuje vyšší úrovně elektromagnetického rušení, než je povoleno

Rušení: Odrazy na vedení, odběr proudu



- Odrazy na vedení (vodičích)
 - Při šíření signálů dochází na konci vedení k jejich odrazům a zpětnému šíření k jejich zdroji
 - Elektrický signál se ve vodičích šíří konečnou rychlostí (shora omezeno rychlostí světla)
 - Odražený signál se přičítá k aktuální hodnotě na vedení ovlivňování logických úrovní
 - Vedení je třeba tzv. impedančně přizpůsobit
- Změny odběru proudu obvodem (oproti ustáleným hodnotám)
 - Při změnách logických úrovní (přechody z 0->1 a 1->0)
 - Toto se projeví (díky úbytkům na napájecích rozvodech) změnami hodnot logických úrovní na výstupu
 - Nutno "odfiltrovat" pomocí blokovacích kondenzátorů na rozvodech napájecího napětí

Rušení: Rozvody napájecího napětí



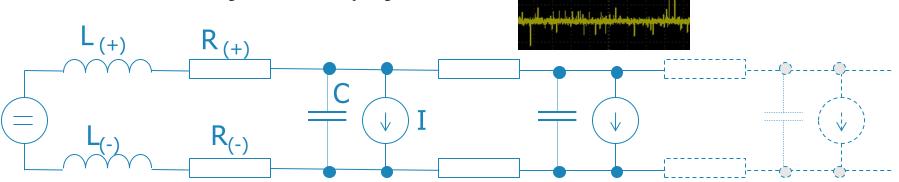
Zdroj

napětí

Náhradní schéma rozvodů napájecího napětí

• L – indukčnost (přívodní vodiče, piny IO)

- R odpor propojovacích vodičů (Cu, Al)
- C parazitní kapacity mezi vodiči
- I odebíraný proud obvodu
- Zdroje rušivého napětí
 - $U_R = I \cdot R$, $U_L = di/dt$
 - LRC tvoří rezonanční obvod
 - Zvlnění napájecího napětí (ripple) je způsobeno poklesy napětí na R a nabíjením a vybíjením LC



R₍₊₎

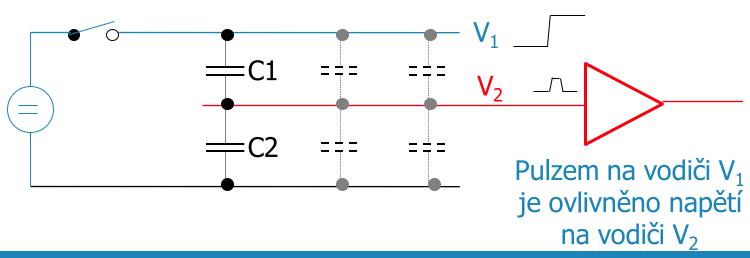
Integrovaný

obvod (IO

Rušení: Přeslechy



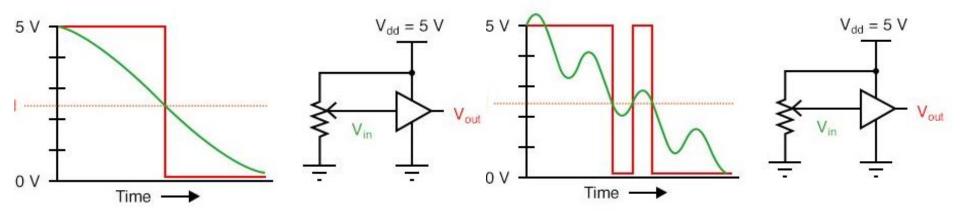
- Mezi dvěma blízkými vodiči dochází vlivem parazitních kapacit k přeslechům (crosstalk)
 - Pokud se na vodiči V₁ skokově změní napětí ΔU₁ (např. změna hodnoty napětí z logické 0 do logické 1), pak se na vodiči V₂ projeví změna napětí
 - $\Delta U_2 = C1/(C1+C2)\cdot \Delta U_1$
- Lze omezit pečlivým vedením vodičů, ne však plně eliminovat

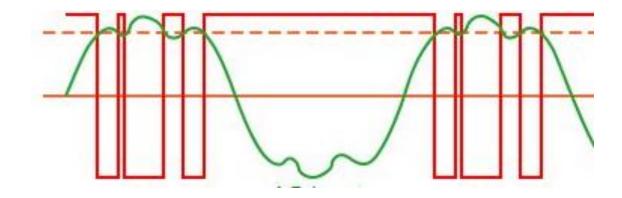


Rušení: Příklad



 Díky rušení se na spoje indukuje napětí, které se přičítá k užitečnému signálu (logickým úrovním)



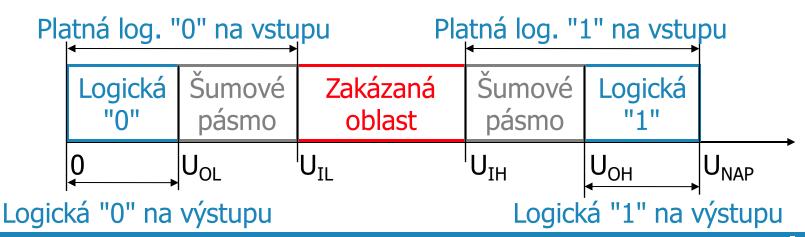


Zdroj: https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/

Rušení: Omezení vlivu



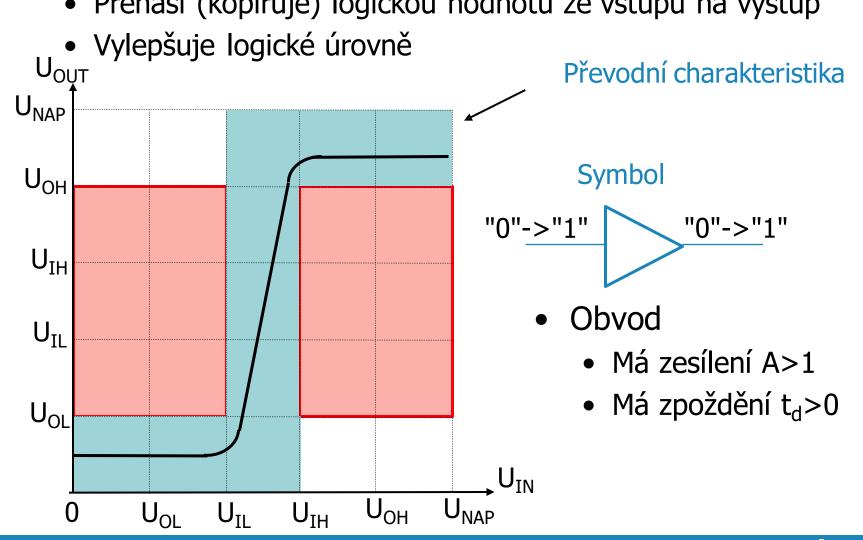
- Šumové pásmo
 - Obvody musí "vylepšovat" hodnoty napěťových úrovní mezi vstupem a výstupem
- Obvody díky tomu akceptují i jistou úroveň rušení
 - Logická "0": napětí v rozsahu 0 až U_{IL}, pásmo akceptovatelného šumu U_{IL}-U_{OL}
 - Logická "1": napětí v rozsahu U_{IH} až U_{NAP} (napájecí napětí), pásmo akceptovatelného šumu U_{OH}-U_{IH}
- Obvod generuje "vylepšené" logické úrovně



Obvod odolný proti rušení: Příklad



- Sledovač (tzv. buffer)
 - Přenáší (kopíruje) logickou hodnotu ze vstupu na výstup

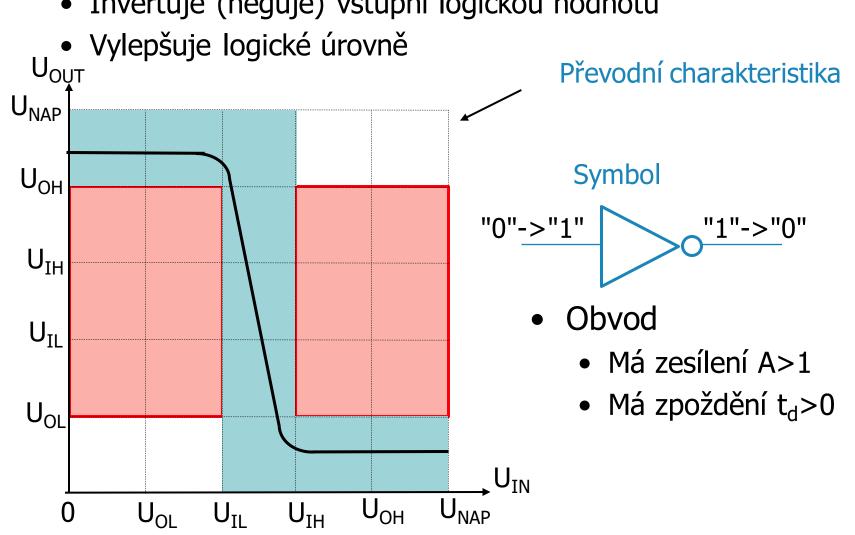


Obvod odolný proti rušení: Příklad



Invertor

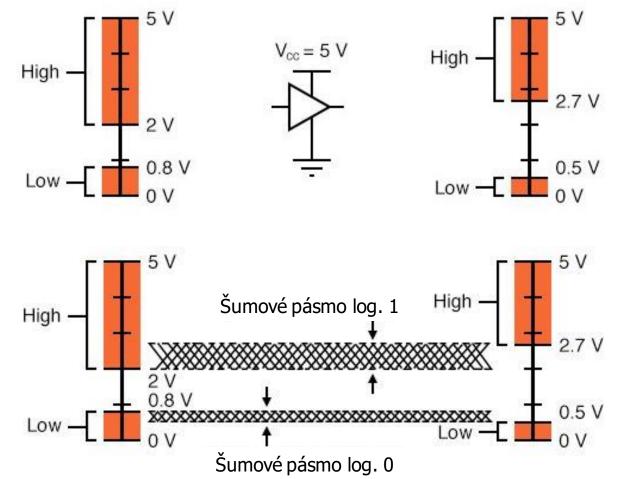
• Invertuje (neguje) vstupní logickou hodnotu



Rušení: Šumová odolnost – příklad technologie TTL



 Akceptovatelné vstupní logické úrovně Generované výstupní logické úrovně



[Zdroj: https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/]

Příkon a energie: Nabíjení/vybíjení kondenzátoru



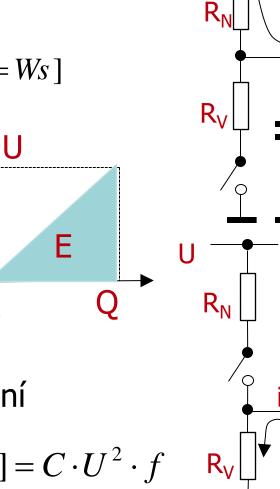
- Náboj $Q = C \cdot U[C] = I \cdot t[As]$
- Energie $E = 1/2 \cdot Q \cdot U = 1/2 \cdot C \cdot U^2 [J = Ws]$
- Nabíjení $E_C = E_{RN} = 1/2 \cdot C \cdot U^2$
- Vybíjení $E_C = E_{RV} = 1/2 \cdot C \cdot U^2$
- Energie nabití/vybití

$$E_{0\to 1\to 0} = E_{RN} + E_{RV} = C \cdot U^2$$

Příkon při periodickém nabíjení/vybíjení

$$P[W] = \frac{E[Ws]}{t[s]} = E[Ws] \cdot f[Hz] = C \cdot U^2 \cdot f$$

• f... frekvence změn 0-1-0 (frekvence hodinového signálu)



Elektrické vlastnosti log. členů



Logické úrovně

• Log. členy jsou konstruovány tak, aby byly za normálních podmínek (teplota, napájecí napětí, rušení atd.) schopny generovat log. úrovně v jistém intervalu hodnot a též rozlišovat log. úrovně v určitém rozmezí hodnot

Odolnost proti rušení (DC Noise Margins)

- Je zajištěna v určitém rozmezí tak, že log. člen je schopen akceptovat větší rozptyl
 vstupních hodnot log. úrovní, než jaký generuje na výstupu
- Rušení může být generováno např. kosmickým zářením, elektromagnetickým polem, kolísáním napájecího napětí apod.

Logický zisk (Fan-Out)

 Počet vstupů log. členů, které můžeme zapojit na výstup daného členu, při kterém jsou ještě zaručeny správné hodnoty log. úrovní pro celý rozsah pracovních podmínek (napájecí napětí, teplota)

Rychlost

- Doba, které je třeba k přechodu signálu ze vstupu na výstup
- Dána dobami přechodů mezi log. úrovněmi a dobou průchodu signálu
- Závisí na konstrukci log. členů, na počtu jiných log. členů zapojených na jeho výstup, na délce vodičů, na konstrukci desky s plošnými spoji atd.

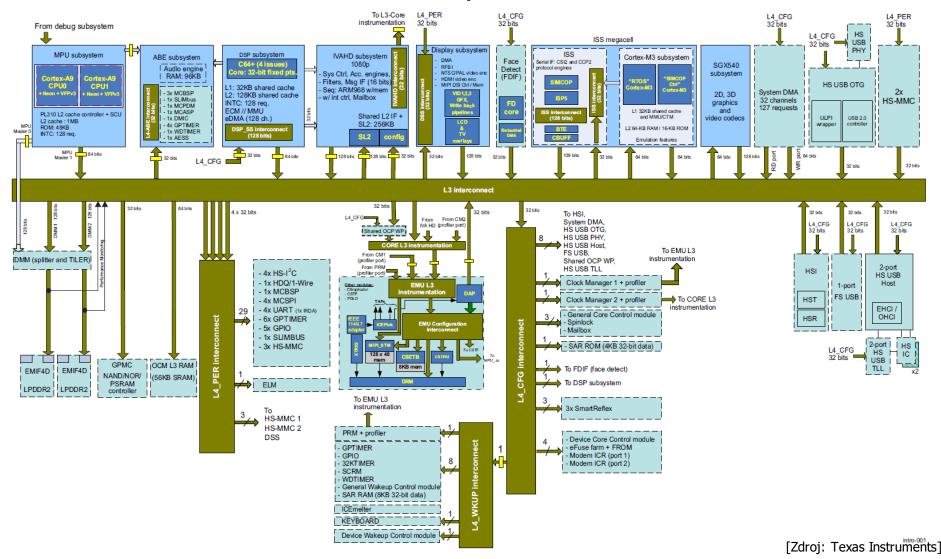
Příkon

 Závisí na konstrukci log. členu, na počtu členů, frekvenci změn log. úrovní, parazitních atd.

Limity: Složitost



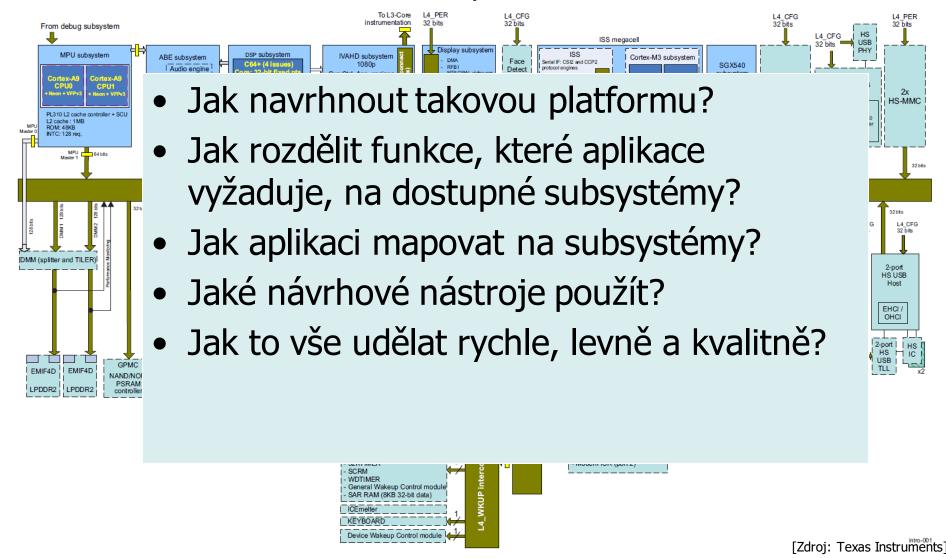
Platforma OMAP4430 - SoC pro mobilní zařízení



Limity: Složitost



Platforma OMAP4430 - SoC pro mobilní zařízení



Návrh složitých systémů



- Složité systémy
 - Jsou sestaveny z jednodušších komponent, které vzájemně komunikují a kooperují a společně utváří chování či vlastnost systému, jež jde nad rámec možností jednotlivých komponent
- Vědecké metody, inženýrský přístup, umění
 - Modelování, optimalizace, verifikace...
 - Vzdělání, zkušenosti, tradice, koncepce...
 - Invence, kreativita, estetika, vize, intuice...
- Kompromisy
 - Hledání kompromisů mezi potřebami uživatelů a možnostmi technologií (cena, výkonnost, spotřeba energie atd.)
- Prostředí
 - Je třeba respektovat životní prostředí a aspekty sociální, etické, zdravotní, bezpečnostní, výrobní, servisní a další

Návrh složitých systémů

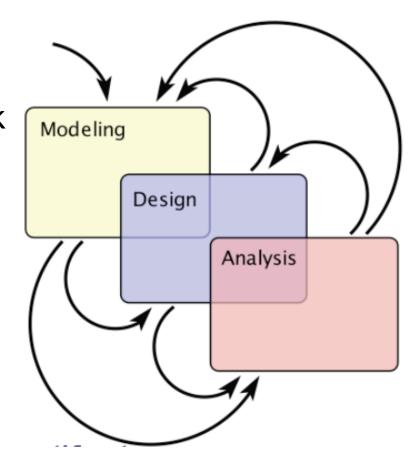


- Techniky návrhu
 - Dekompozice (struktura = komponenty + rozhraní)
 - Hierarchie (snaha o co nejvyšší úroveň abstrakce při návrhu)
 - Vhodná specifikace (popis struktury a chování)
 - Způsob popisu (shora-dolů, zdola-nahoru)
 - Automatizace návrhu (syntéza HW, kompilace SW)
 - Validace (testování, verifikace,...)
- Dekompozice systému a zavedení hierarchie vyžaduje definici rozhraní mezi subsystémy => rozhraní:
 - Umožňuje dekompozici systému a zavedení hierarchie
 - Izoluje jednotlivé subsystémy (komponenty) a různé technologie mezi sebou a umožňuje jejich vzájemné propojení
 - Umožňuje stavbu systému ze standardních komponent
 - Má často delší životnost než samotný systém (viz např. USB)

Modelování, návrh, analýza



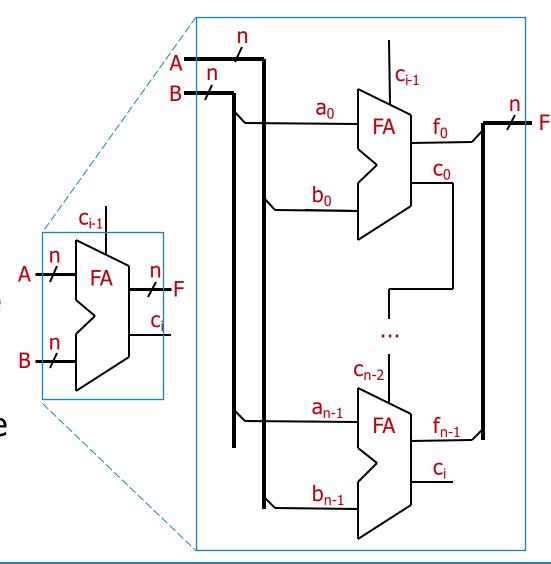
- Modelování specifikuje, co systém dělá
 - Proces získávání hlubší znalosti o systému jeho imitací
- Návrh (design) specifikuje, jak systém pracuje
 - Strukturovaná tvorba výrobku
- Analýza specifikuje, proč systém dělá to, co dělá
 - Proces získávání hlubší znalosti o systému jeho podrobným zkoumáním



Popis systémů: Hierarchie



- Pro zjednodušení návrhu je výhodné používat hierarchický popis systémů
- Hierarchie zapouzdřuje (seskupuje) jednodušší obvody do složitějších celků (komponent), se kterými se lépe pracuje
- Příklad:
 N jednobitových
 úplných sčítaček (FA) je
 seskupeno do jedné N bitové úplné sčítačky



Popis systémů: chování - struktura



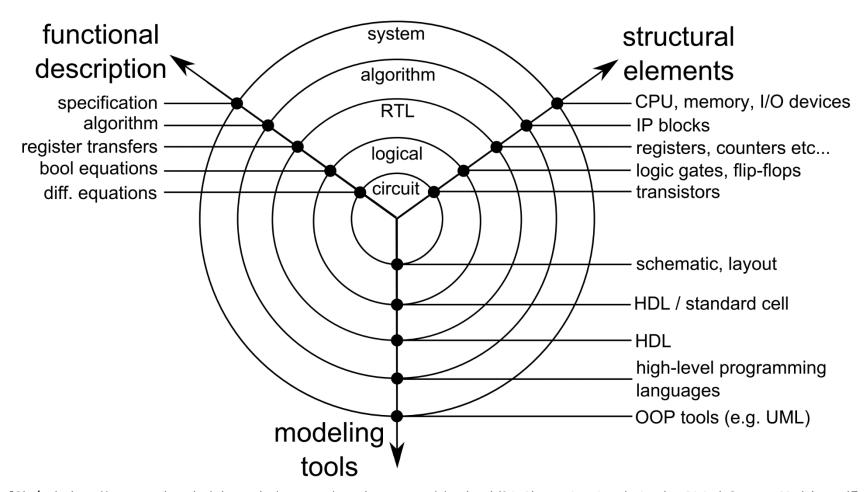
- Popis chování
 - Zápis algoritmu, který má obvod vykonávat
 - Matematický výraz
 - Programovací jazyk
 - Vývojový diagram
 - Tabulka
 - Časový diagram, atd.
- Specifikace toho, co má systém dělat (ne toho, jak bude implementován)
 - Tvůrčí činnost, kterou lze jen omezeně automatizovat

- Popis struktury
 - Definice a propojení jednotlivých prvků systému
 - Schéma
 - Programovací jazyk, atd.
- Specifikace toho, jak bude systém implementován (ne toho, co bude dělat)
 - Jednotlivé komponenty, ze kterých se systém skládá
 - Všechny signály, kterými se přenáší informace jak mezi komponentami, tak okolím
 - Tok informace v systému
 - Lze automatizovat

Popis systémů: Úrovně abstrakce



 Popis chování (behaviorální, funkční), popis struktury, modelovací nástroje



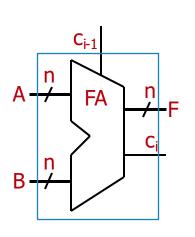
 $[Obr\'{a}zek: \ http://www.eet.bme.hu/\sim horvathp/contents/aramkortervezes/eloadasok/01_Abstraction_Levels_in_the_Digital_System_Modeling.pdf]$

Popis systémů: Úrovně abstrakce



- Nižší úrovně abstrakce nesou více informací o struktuře výsledného obvodu, vyšší popisují jeho chování
 - Přechody z vyšší do nižší úrovně abstrakce = kroky návrhu
- Příklad úplná N-bitová sčítačka složena z 1bitových

 Popsána algebraickými výrazy (popis chování) a jako logická síť (popis struktury)



Úroveň popisu	Úroveň abstrakce	Množství detailů
Systém	Nejvyšší	Nejméně
Přenosy mezi registry	1	
Log. obvody a členy		
Tranzistory		+
Polovodičová podložka	Nejnižší	Nejvíce

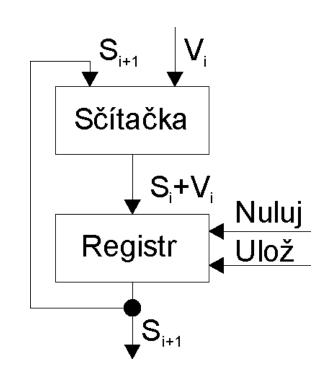
 $F=(A+B)\mod(2^{n}-1)$

 $c_{n-1} = ((A+B) > (2^{n}-1))$

Úroveň popisu: Přenosy dat mezi registry (RTL)



- Příklad součet členů posloupnosti $S_0 \leftarrow 0$;
- Popis chování programovacím jazykem
- $S_0 \leftarrow 0;$ vynulujregistr $\underline{for} \ i = 1 \ \underline{to} \ N \ \underline{do};$ provšechna V_i $(S_{i+1} \leftarrow S_i + V_i);$ akumuluj
- Popis struktury graficky schématem
 - Funkční jednotka sčítačka, provádí součet dvou čísel (mezivýsledek plus další člen posloupnosti)
 - Registr S (paměť) uchovává (akumuluje) jednotlivé mezivýsledky
 - Sčítačka a registr jsou propojeny signály
 - Registr je na počátku nejprve signálem Nuluj vynulován a dále si postupně, na povel signálu Ulož, pamatuje novou hodnotu výsledku



Různé úrovně a způsoby popisu systému: Příklady

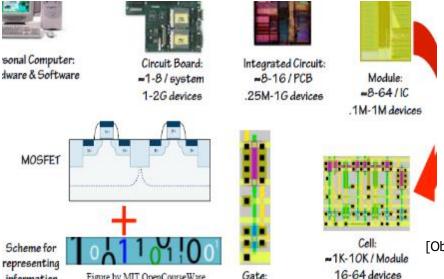


- Osobní počítač
- Deska s plošnými spoji
- Integrovaný obvod
- Komponenta
- Logická buňka

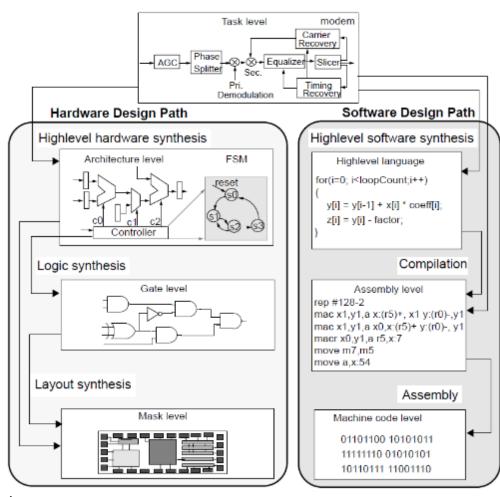
Figure by MIT OpenCourseWare.

information

- Hradlo
- **Tranzistor**
- Reprezentace informace (0, 1)



Modem – SW a HW



[Obrázek: A. P. Kalavade: System-Level Codesign of Mixed Hardware-Software Systems, University of California, Berkeley, Technical Report No. UCB/ERL M95/88, 1995]

Postup při návrhu číslicových systémů



- Shora-dolů (anglicky top-down)
 - Nejprve definujeme chování systému a postupně konkretizujeme jednotlivé části – od obecného po podrobný popis struktury (podklad pro výrobu)
- Zdola-nahoru (anglicky bottom-up)
 - Vycházíme z komponent, které máme k dispozici
 - Postupně z nich skládáme jednotlivé bloky, z nichž postupně vybudujeme celý systém
- Oba přístupy je nezbytné vhodně kombinovat
 - Při návrhu algoritmu pro řešení daného problému je snahou postupovat co nejvíce "shora-dolů"
 - V praxi je však třeba též postupovat "zdola-nahoru" pro optimalizaci návrhu (cena, příkon apod.) je třeba znát cílovou technologii, ve které bude systém fyzicky realizován, a využívat její komponenty

Návrh = algoritmus + syntéza



- Návrh = tvorba algoritmu a jeho implementace (syntéza)
- Algoritmus
 - Intuitivně postup, který nás dovede k řešení úlohy
 - Formálně "Přesně definovaná konečná posloupnost příkazů (kroků), jejichž prováděním pro každé přípustné vstupní hodnoty získáme po konečném počtu kroků odpovídající výstupní hodnoty" [z kurzu Základy programování]
- Syntéza
 - Z formálního popisu algoritmu je třeba vytvořit výpočetní strukturu (logický obvod), která jej implementuje
 - Tvorba vhodné struktury se dnes provádí do značné míry automatizovaně
- Existuje mnoho výpočetních struktur, které implementují daný algoritmus (vícenásobná realizace)
 - Jejich vlastnosti určují rychlost výpočtu, příkon, rozměry, atd.