



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



Návrh hardwarových komponent

Milan Kolář

Ústav mechatroniky a technické informatiky



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

Projekt ESF CZ.1.07/2.2.00/28.0050
**Modernizace didaktických metod
a inovace výuky technických předmětů.**

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Literatura

- **E-learningový portál: <https://elearning.tul.cz>**
- Pinker, J. – Poupa, M.: Číslicové systémy a jazyk VHDL. BEN, 2006, ISBN 80-7300-198-5.
- Král, J.: Řešené příklady ve VHDL. Hradlová pole FPGA pro začátečníky. BEN, 2010, ISBN 978-80-7300-257-2.
- Šťastný, J.: FPGA prakticky. Realizace číslicových systémů pro programovatelná hradlová pole. BEN, 2010, ISBN: 978-80-7300-261-9.

Rozvrh:

přednáška: Čt 8:50 (A109)

cvičení: Čt 10:40 (A109)



Elektronický systém

Elektronický systém – zařízení zpracovávající data, která mohou být analogová (spojité hodnoty) nebo číslicová (diskrétní hodnoty).

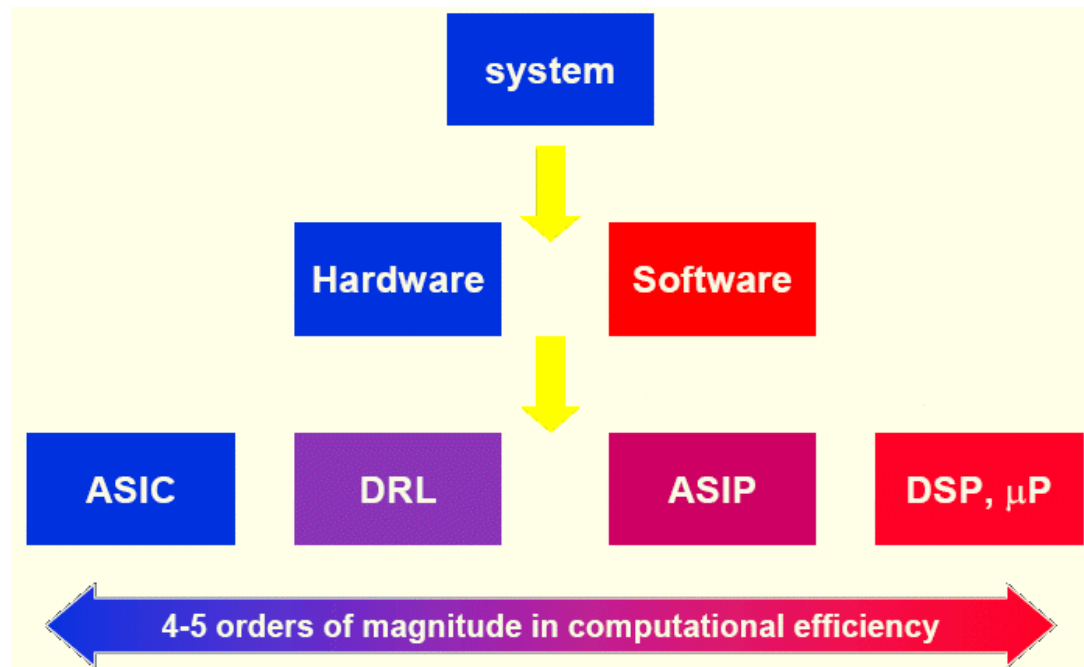
V dnešní době se preferuje **číslíkové** zpracování signálů (informací, dat) jak **hardwarovými**, tak **softwarovými** prostředky.

Číslicové systémy lze v zásadě rozdělit na:

- **aplikačně specifické** - architektura systému je přizpůsobena danému účelu (vyšší výkon, vyšší náklady na vývoj, vhodnější pro větší série, obtížnější změny funkce, obtížnější okopírování);
- např. FPGA, ASIC, CPLD.
- **mikroprocesorové** - univerzální architektura doplněná vhodným programem - řízená instrukcemi (horší dynamické vlastnosti, pružná změna funkce, velká výpočetní kapacita);
- např. μ P, DSP, μ C, PLC.



Návrh elektronického systému



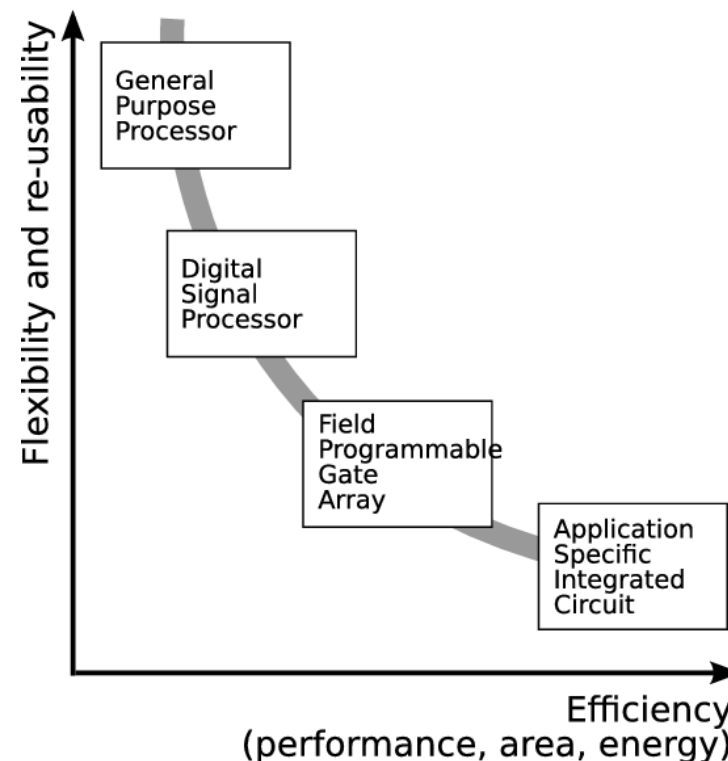
DRL – Dynamically
Reconfigurable Logic

ASIP – Application Specific
Instruction set Processor

Výběr architektury významnou měrou určuje budoucí parametry (výkonnost, cenu, flexibilitu). Současné návrhové systémy umožňují souběžný návrh hardwaru a softwaru (HW/SW co-design).

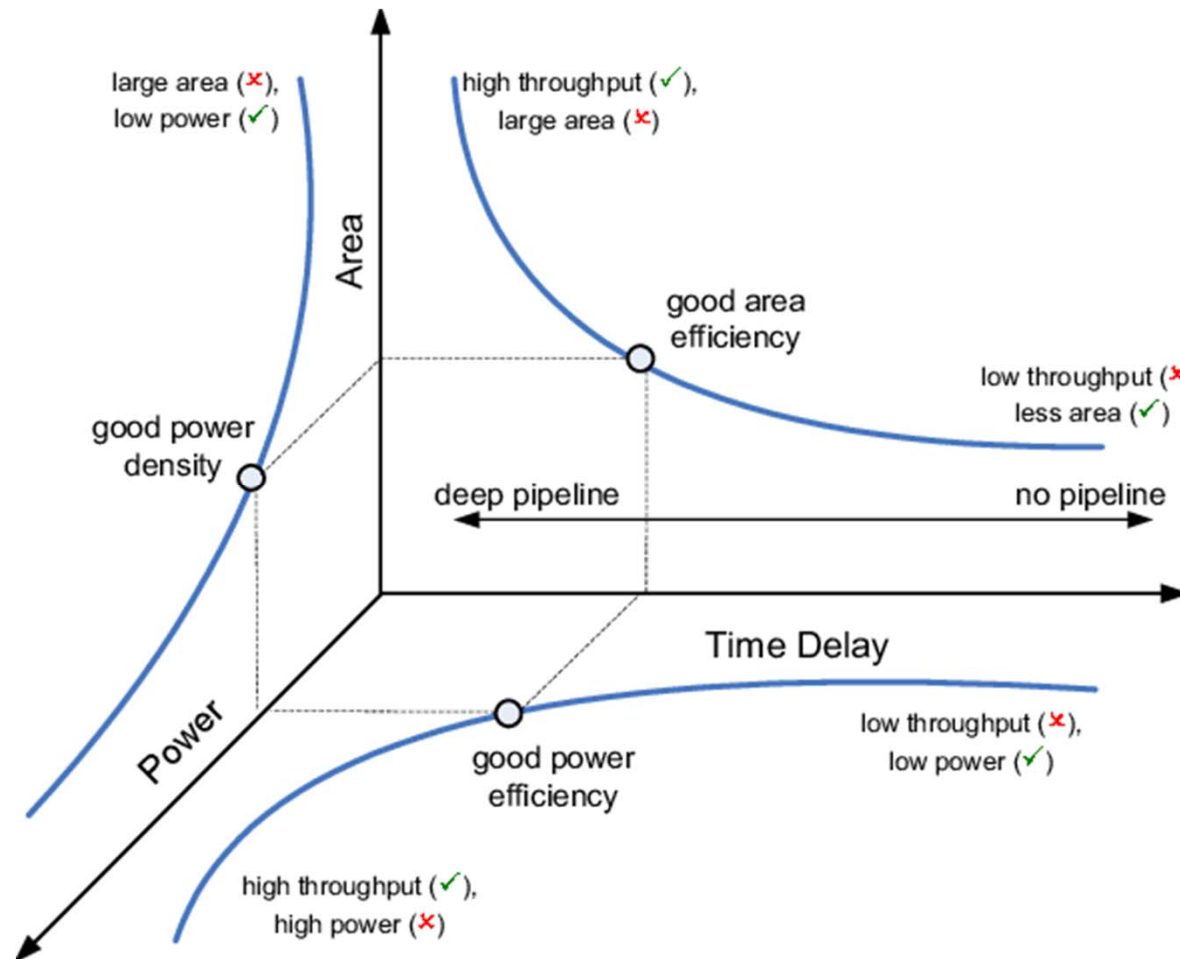
Efektivita vs. flexibilita

Čím bude specializovanější HW pro daný účel, tím bude vyšší energetická efektivita (operací/mW), výpočetní výkonnost i hustota funkcí na čipu.





Kompromisní křivka HW návrhu





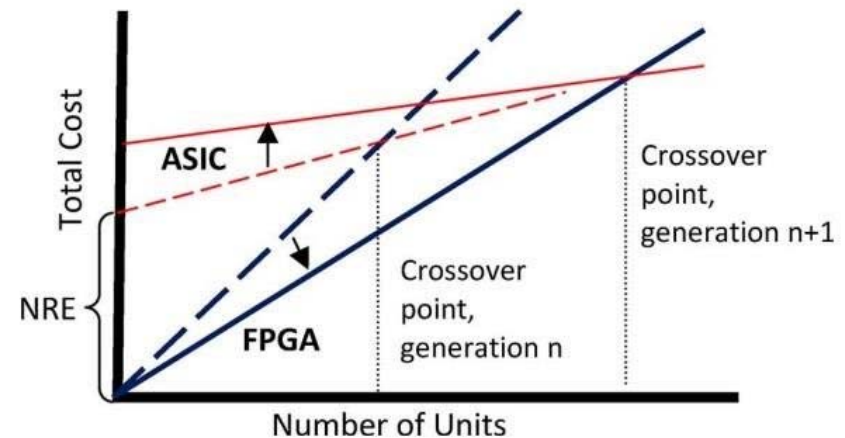
Dělení zakázkových IO

Plně zakázkové (Full Custom)

- podle uživatele se navrhují všechny masky technologického procesu;
- velké náklady na vývoj;
- vhodné pro velké série;
- vyšší výkonnost;
- nižší spotřeba;
- např. ASIC, ASSP, SoC.

Programovatelné (Programmable)

- uživatel sám programuje funkci;
- rychlejší vývoj (uvedení na trh);
- vyšší cena za kus;
- horší dynamické vlastnosti;
- např. PLD, CPLD, FPGA.





Projektování zakázkových obvodů

Rozlišujeme 3 fáze (ASIC):

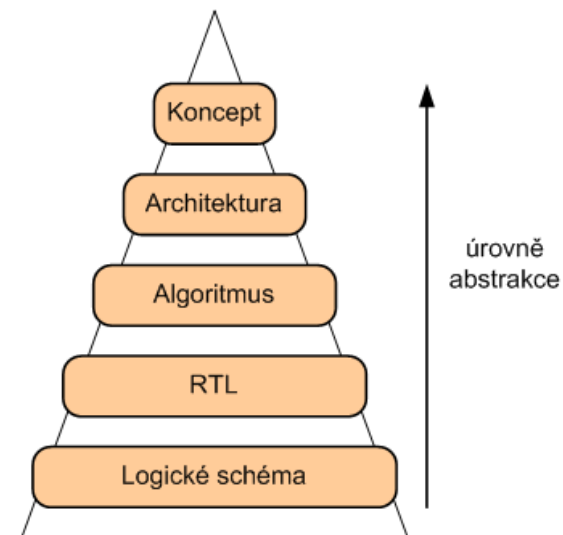
- systémový návrh,
 - návrh masek,
 - technologie výroby.
- nutné návrhové systémy
 - Siemens EDA (Mentor Graphics)
 - Cadence
 - AMD (Vivado)
 - Intel (Quartus)
 - přesun systémového návrhu k uživateli
 - oddělení návrhového a výrobního procesu
 - nutnost přesně definovat rozhraní
 - vzniká skupina systémových inženýrů různých oborů





Systémový návrh

- nejdůležitější fáze, cílem je nalezení obvodové struktury,
- rozčlenění návrhu na úrovně abstrakce,
- smíšená forma vstupního popisu (na různých úrovních) využitím HDL (Hardware Description Language) jazyků,
- syntéza je obecně mnohoznačná úloha (na nižších úrovních abstrakce je automatizovaná),
- ověření správnosti v HDL simulátorech.





Úrovně abstrakce

- **Behaviouristická úroveň**
 - popis chování systému na vyšší úrovni abstrakce,
 - volba algoritmů a architektury (neuvažujeme detaily);
- **Úroveň FB (RTL - Register Transfer Level)**
 - popis tokem dat (stavový diagram)
 - FB: logické operátory, RAM, DSP, multiplexory, čítače, ...
 - Nejčastěji používaná úroveň ve VHDL a Verilog HDL;
- **Úroveň logického schématu**
 - popis na úrovni hradel,
 - jednoznačný přechod na úroveň tranzistorů,
 - často již hardwarově závislé (pro konkrétní FPGA);
- **Úroveň strukturální**
 - vzájemné propojení jednotlivých funkčních bloků či knihoven,
 - omezení možností syntézy či optimalizace.



Postupy systémového návrhu

Většinou hierarchický přístup:

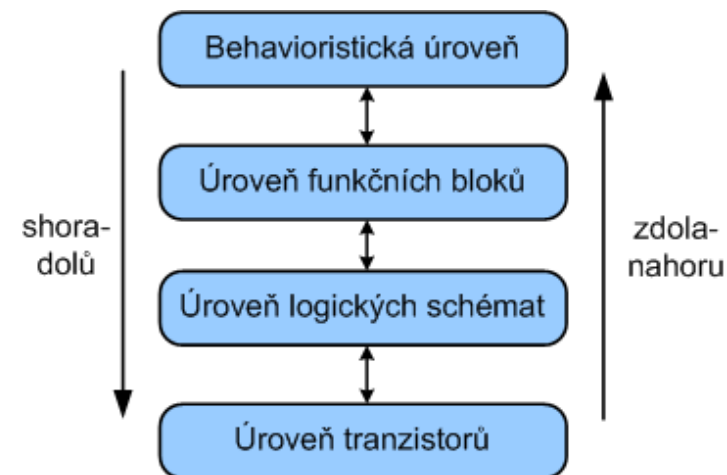
Zdola-nahoru (bottom-up) – systém skládáme z hotových FB

- vhodné pro menší systémy,
- menší nároky na zkušenosti (ověřené FB),
- výsledné řešení méně optimální.

Shora-dolů (top-down) – koncepci rozvádíme do podrobností

- vhodné pro velké systémy,
- dosažení lepšího výsledku,
- náročné na zkušenosti.

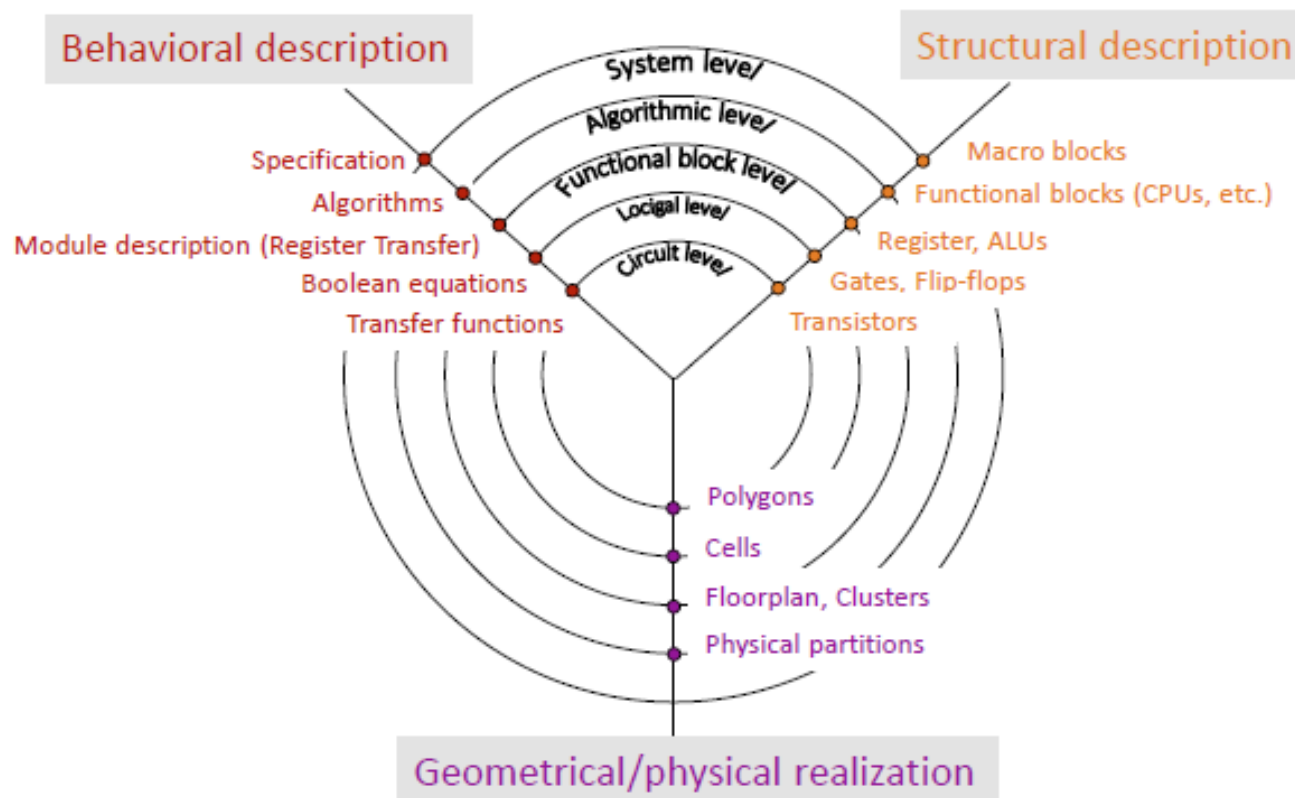
V jednoduchých případech:
model **plochého typu** (flat)





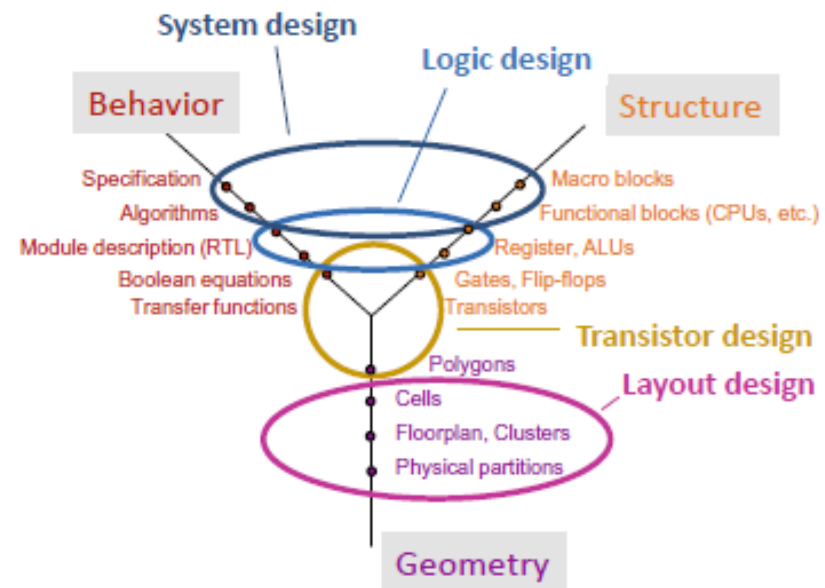
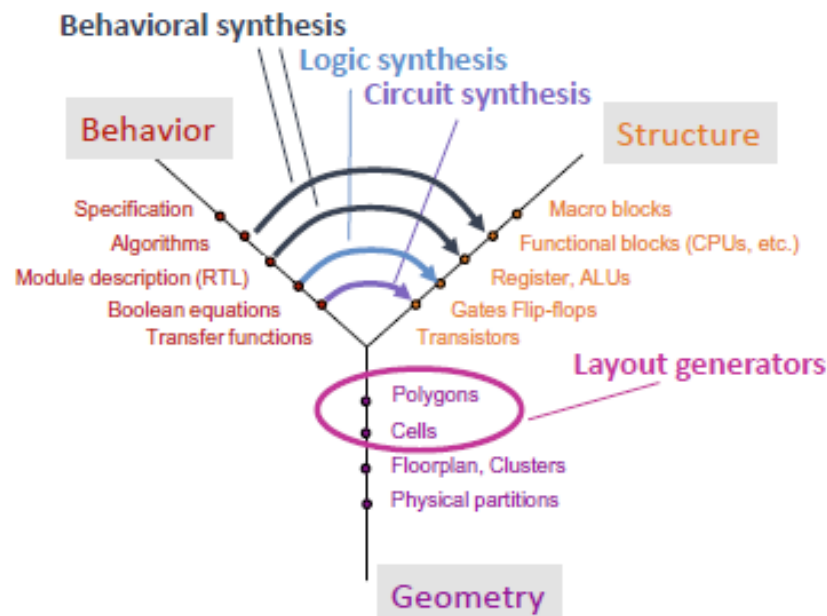
Gajski-Kuhnův Y diagram

Zobrazuje různé perspektivy při návrhu integrovaných číslicových systémů. Vývoj hardwaru je vnímán ve třech doménách s různými stupni abstrakce.



Gajski-Kuhnův Y diagram

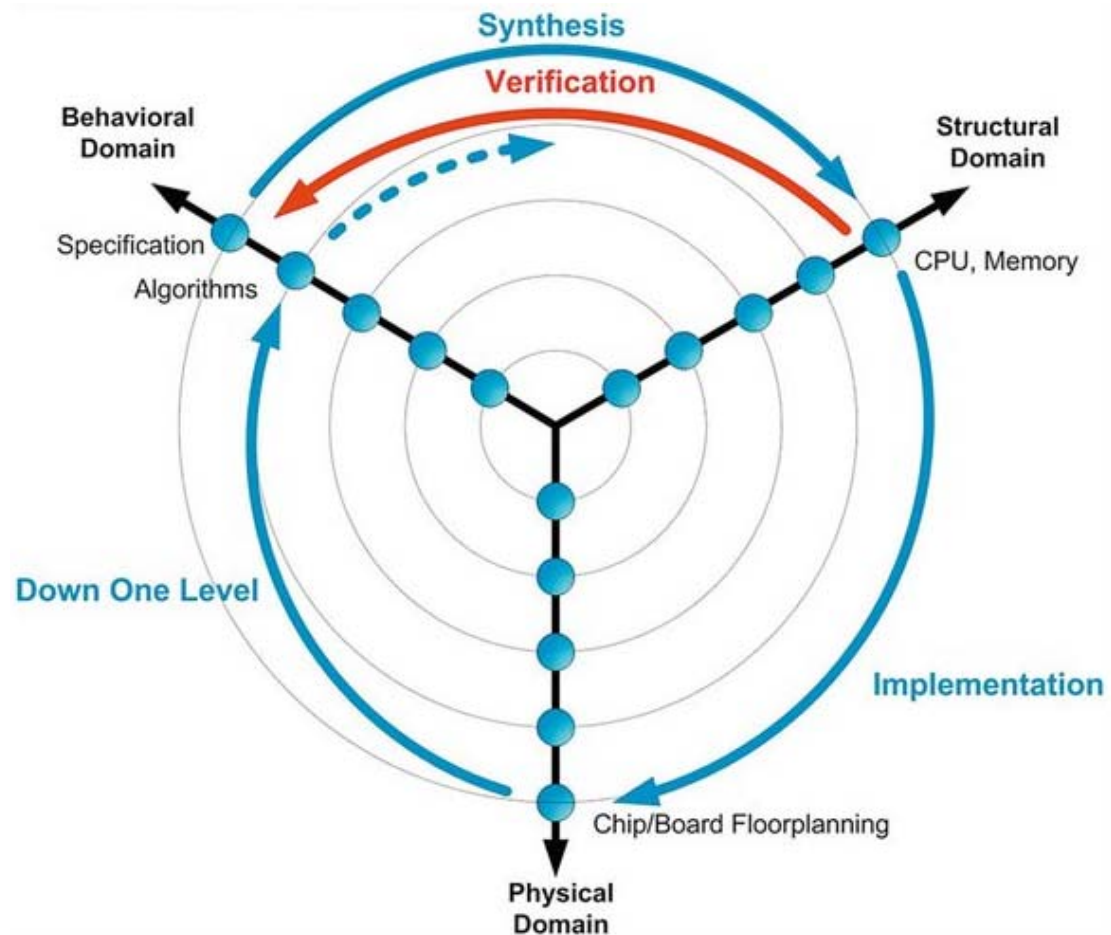
Přechod mezi úrovněmi abstrakce může být manuální nebo automatizovaný (shora-dolů i zdola-nahoru).



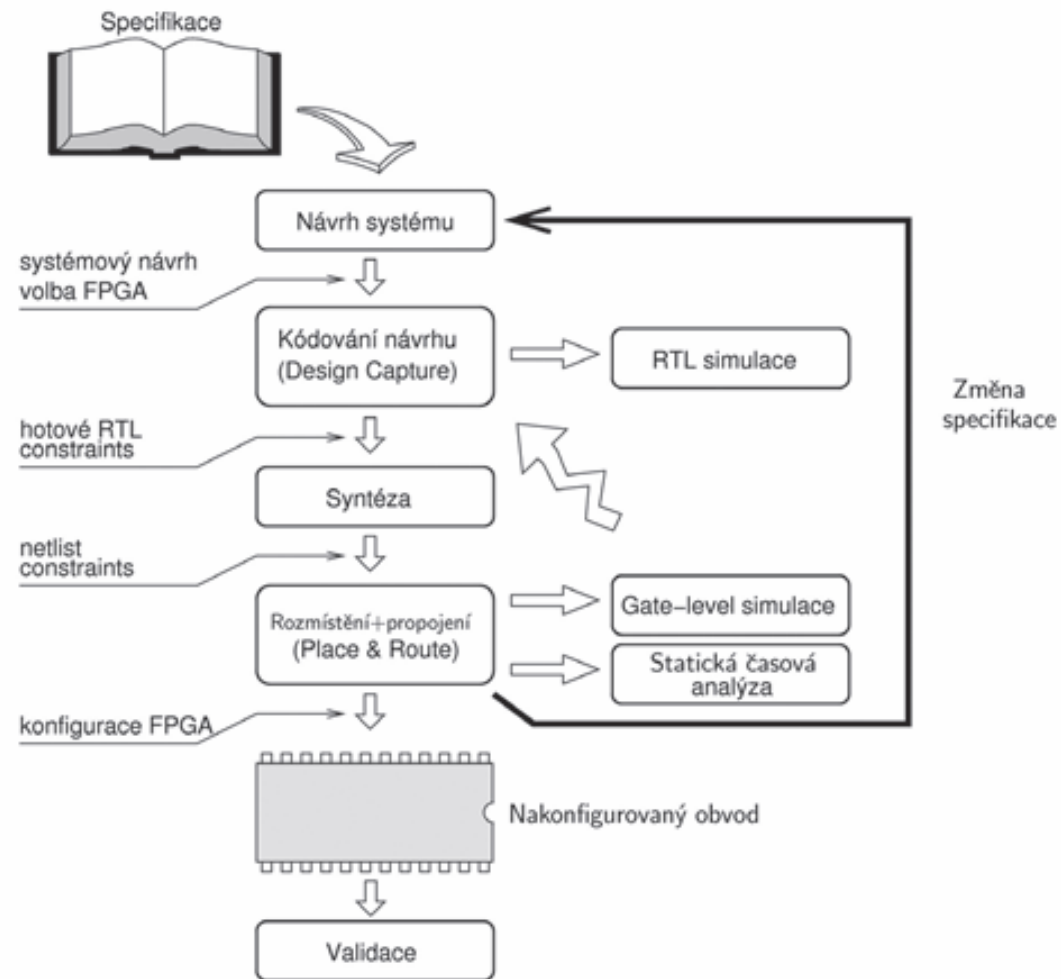


Gajski-Kuhnův Y diagram

Ilustrace typického návrhu shora dolů (spirála)

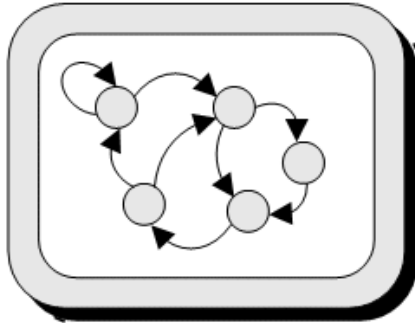


Postup návrhu FPGA



Formy popisu

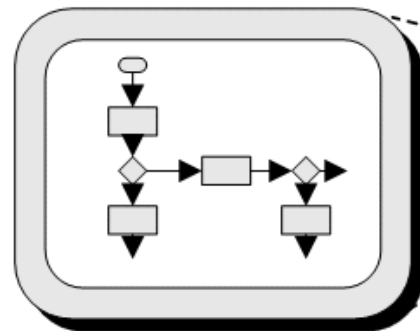
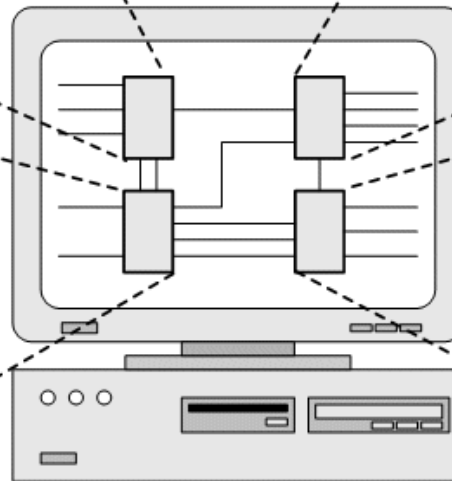
Graphical State Diagram



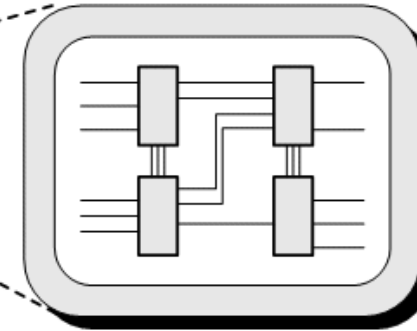
Textual HDL

```
When clock rises  
If (s == 0)  
then y = (a & b) | c;  
else y = c & !(d ^ e);
```

Top-level
block-level
schematic



Graphical Flowchart



Block-level schematic



HDL jazyky

HDL – programovací jazyky popisující v textové formě chování elektronických obvodů (převážně číslicových), popis na různých úrovních abstrakce. Nepopisují funkci sekvenčně, ale paralelně (distribuované bloky). Jazyky lze použít i pro simulaci a verifikaci.

Návrhový systém nemusí být schopen celý popis v HDL jazyku přeložit do netlistu (implementuje jen tzv. **syntetizovatelnou podmnožinu** HDL). Vlastnosti HDL jazyka, které slouží k řízení simulace a verifikace funkce, tvoří **simulační podmnožinu** HDL.



HDL jazyky - přehled

VHDL (Very High Speed Integrated Circuits **HDL**) – standardizován v roce 1987 (IEEE Std. 1076-1987), v současnosti nepoužívanější pro popis FPGA (zejména v Evropě);

Verilog HDL – první standard v roce 1995 (IEEE Std. 1364-1995), rozšířený zejména pro ASIC (v Americe a Asii), podobný jazyku C.

Nové jazyky vznikají zejména z důvodu dosažení větší abstrakce od hardwaru, kladou důraz na popis na úrovni algoritmů (nejsou koncipovány pro přímý popis HW – nejsou to HDL jazyky). Syntéza buď přímo do cílové technologie nebo přes RTL v jazycích VHDL nebo Verilog (označuje se **HLS** – High Level Synthesis).

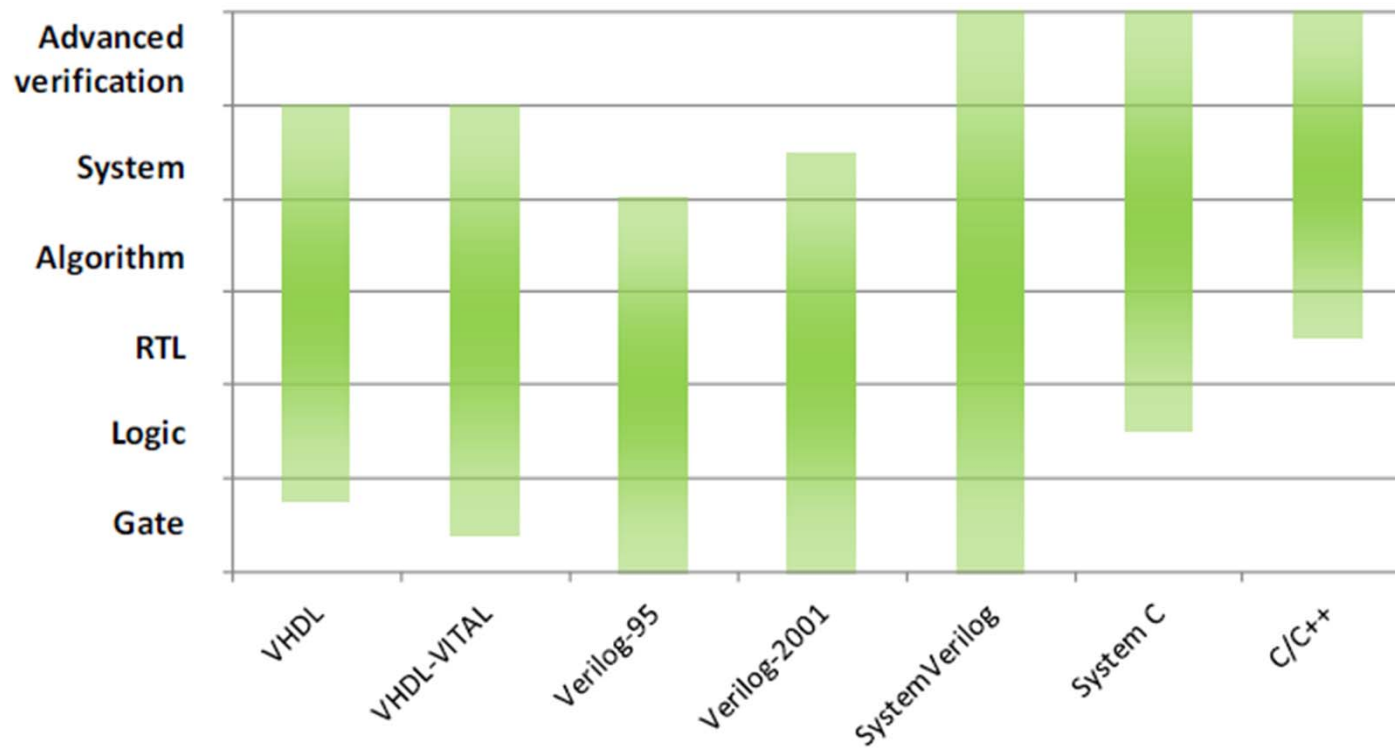
SystemVerilog – první standard IEEE Std 1800-2005, v roce 2009 sloučeny Verilog a SystemVerilog do IEEE Std. 1800-2009;

SystemC – standard IEEE Std 1666-2005, vychází z objektového jazyka C++.



Úroveň abstrakce program. jazyků

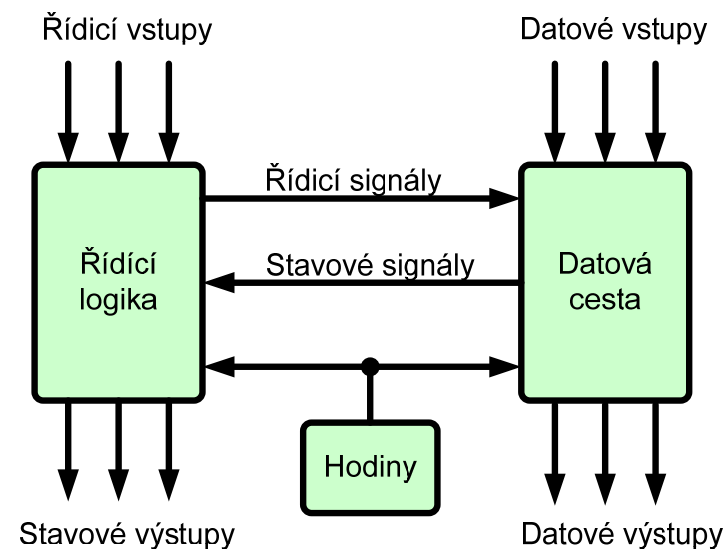
Přibližná úroveň abstrakce dosažitelná v různých programovacích jazycích vhodných pro popis číslicových systémů





Typické blokové schéma systému

- oddělení zpracování dat od řídicích obvodů (obecně rozdělení do několika relativně samostatných bloků)

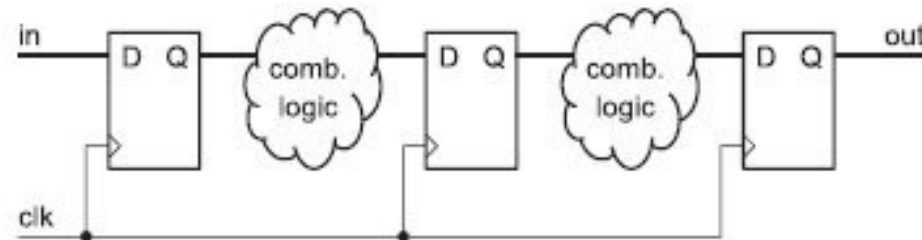




Návrh datové části

Datová část se obecně skládá z:

- výpočetních jednotek (ALU, MAC, ...),
- registrů pro uchovávání proměnných (paměti, zásobníky),
- komunikačních datových sběrnic.



V datové části řešíme algoritmizaci (obecně dvěma krajními implementacemi – sériovou nebo paralelní).

Návrh řídicí části

Řídicí část řešena:

- a) pamětí s mikroprogramem (u velkých řadičů neúměrně roste potřebná paměť)
- b) stavovým automatem (Moore, Mealy)

