



Slovo úvodem

Cíl prezentace

- informovat o praktických zkušenostech s návrhem na FPGA obvodech
- většina obvodů navrhovaných v ASICentru – ASICy, avšak realizovalo se i mnoho FPGA návrhů:
 - prototypy ASICů
 - konverze ASICů do FPGA (emulace)
 - samostatné FPGA aplikace
 - větší digitální obvody (procesory ap.), >200kGates
 - DSP systémy

Předpoklady na posluchače

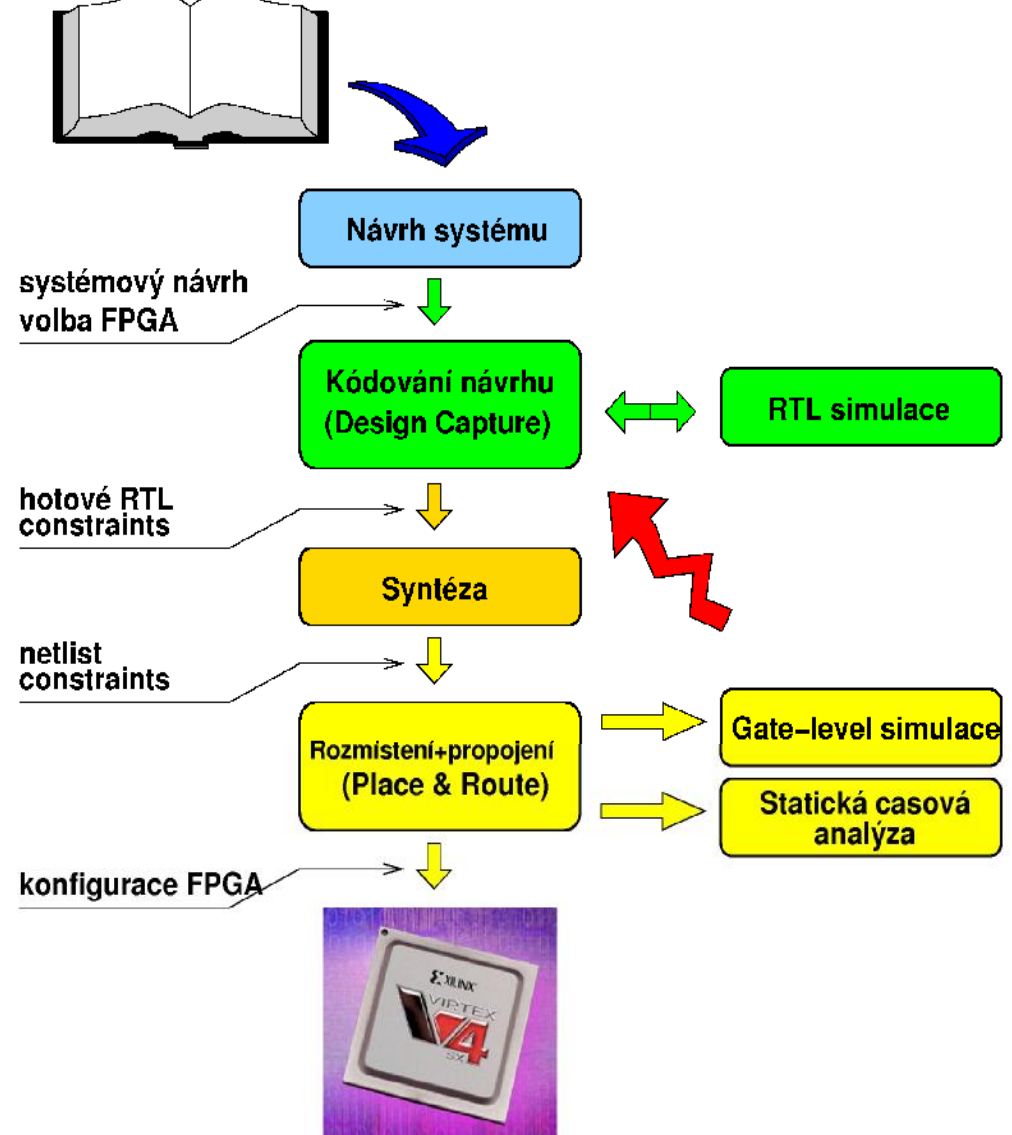
- základní znalost VHDL
- minimální zkušenost s FPGA návrhem



FPGA Design Flow

“Vodopád” návrhu

- softwarové nástroje
- příklady nastavení nástrojů
- základní zásady





Specifikace

Specifikace systému

- úplná
- bezchybná
- bezesporná
- schválená všemi stranami

Plán verifikace

- co a jak má fungovat?
- seznam požadavků na návrh
- priority požadavků – co se musí stihnout?

SW podpora

- textové editory :-)





Výběr vhodného FPGA obvodu

Cílová aplikace?

- **spolehlivost (automotive, space applications)**
- **možnost přeprogramovat zařízení**
- **charakter aplikace – komunikační FPGA, DSP, řídicí, ...**
- **teplotní rozsah**

Další důležité parametry:

- **pracovní hodinová frekvence (technologie, speed grade)**
- **system startup time**
- **spotřeba systému**
- **zabezpečení intelektuálního vlastnictví**

Na velikosti záleží!

- **kolik návrh potřebuje blokové paměti, HW násobiček, FFS**
- **odhad velikosti logiky v hradlech (XAPP059 app. note)**





Systemový návrh

Hierarchická struktura

Top level návrhu

- neobsahuje logiku
- jen vložené bloky

Rozdělení systému do bloků s

- pevně definovanou
 - funkcí
 - rozhraním (signály, protokol)
- každý blok – registrované výstupy
- pomáhá to syntéze i nám
- generátor hodin a resetovacích signálu do zvláštního bloku



Systemový návrh 2

Styl návrhu

- **FPGA – vhodné pro synchronní obvody**
 - **ne!** přepínání hodin
 - **ne!** hradlování hodin
 - **ne!** pokud
 - ❖ zatraceně dobře nevíte, co děláte a proč to děláte
 - ❖ Vaše FPGA to nepodporuje
- **nenavrhovat obvody, které spoléhají na zpoždění, rychlosti logiky,...**
- **vyhnout se asynchronní logice, **ne!** kombinační zpětné vazby**
- **požadavky na časování**
 - **register-to-register**
 - **pad-to-register, register-to-pad, pad-to-pad**

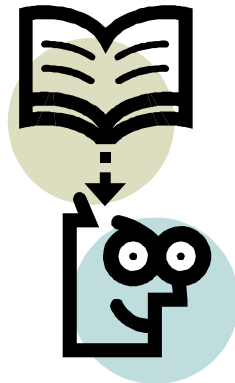


Systemový návrh 3

Reset a asynchronní signály

- **správně synchronizovat**
- **nepoužívat asynchronní set/reset pro běžnou funkci**
- **vždy užít globální reset pro inicializaci**

SW podpora pro vkládání návrhu: HDL Designer, ModelSim Designer



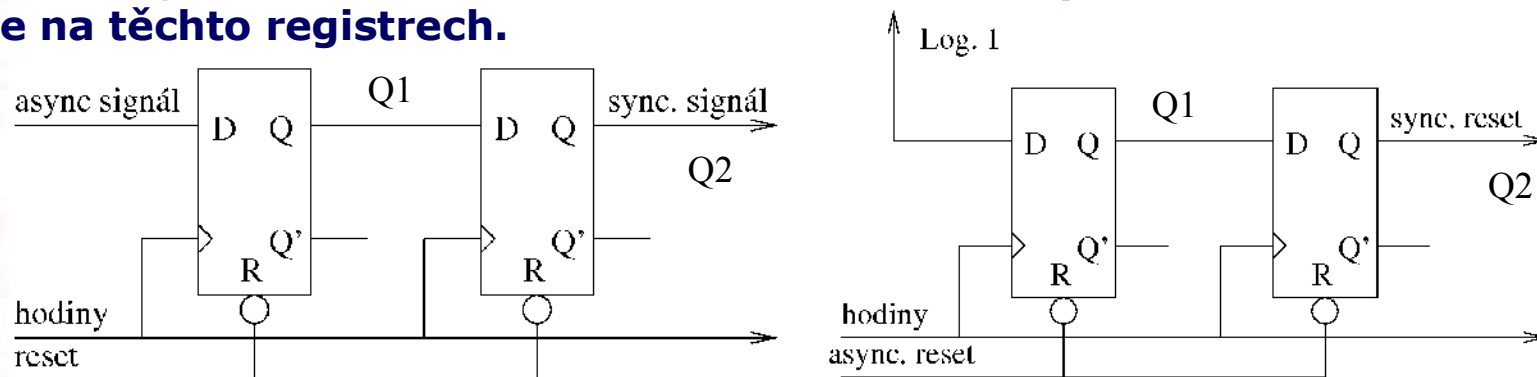


Synchronizace asynchronních signálů a resetu

Nejčastější problémy

- **reset recovery violation**
 - **uvolnění resetu před aktivní hranou hodin**
 - **nutno synchronizovat reset**
 - **řešení: reset synchronizer**
- **setup/hold violation asynchronními signály**
 - **změna asynchronního signálu kolem aktivní hrany hodin**
 - **nutno synchronizovat**
 - **řešení: synchronizer**
 - **i mezidoménový přechod**

Na gate level je nutno zakázat kontrolu dodržení setup, hold a reset recovery time na těchto registrech.





RTL Syntéza

“Konverze popisu systému na vysoké úrovni abstrakce do seznamu logických prvků dané technologie a jejich vzájemného propojení (netlist)”

Vstup: syntetizovatelná podmnožina VHDL

Výstup: netlist

Syntetizují se log. funkce, sekvenční logika, aritmetika, paměti,....

Výsledek syntézy je ovlivněn

- **VHDL kódem (pragmas)**
- **constraints**
- **použitou technologií**

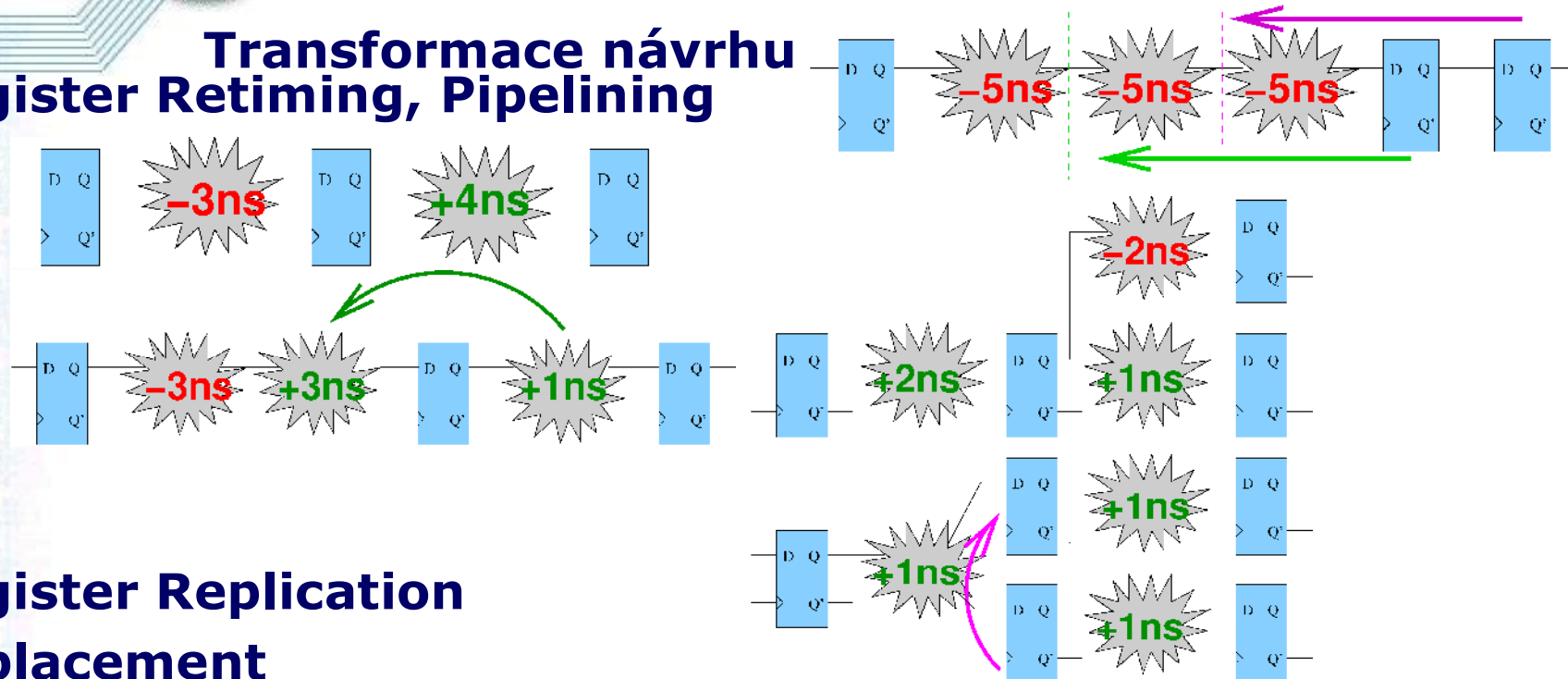
Syntezátor hodnotí výsledek svého snažení podle

- **plochy (minimalizace)**
- **rychlosti (reálné vs. požadované zpoždění - slack)**

a aplikuje na VHDL kód a obvod jisté transformace



Transformace návrhu Register Retiming, Pipelining



Register Replication Replacement

- prvky na kritické cestě jsou přemístěny, aby bylo minimalizováno zpoždění

Resynthesis

- změna rozložení logické funkce do LUTů
- konverze blockRAM do LUTů



Pár rad

Nastavit design constraints

- informace o požadovaných parametrech návrhu
- f_{clk}/T_{clk} systému
- pad-to-register, register-to-register, register-to-pad, pad-to-pad zpoždění

Kontrolovat syntézní logy

- *# Warning: File "memory_debug_rtl.vhd", Line 210: Module work.memory_debug(rtl), Net wr_mem_n: Although this signal is not part of the sensitivity list of this block, it is being read. This may lead to simulation mismatch.*
 - *RTL simulace se může chovat zcela jinak, než FPGA*



Pár rad 2

**# Info: 5205: Module work.clock_controller(stuct), Net timer_speed[0]:
Latch inferred.**

- **vložení hladinového klopného obvodu**
- **velice nebezpečné, pokud to nebyl úmysl!**

```
architecture rtl of logika is
    SIGNAL int : std_logic;
begin
    int <= a OR b;
    something : PROCESS (int,c)
    BEGIN
        IF int='1' THEN
            d <= c;
        ELSE
            d <= '0';
        END IF;
    END PROCESS something;
end rtl;
```

```
architecture rtl of logika is
    SIGNAL int : std_logic;
begin
    int <= a OR b;
    something : PROCESS (int,c)
    BEGIN
        IF int='1' THEN
            d <= c;
        ELSE
            d<= '0';
        END IF;
    END PROCESS something;
end rtl;
```



Place & Route

Implementace netlistu v daném FPGA

- zobrazení na technologii (mapování)
- rozmístění
- propojení

Mapování

- namapování logického netlistu na prvky na FPGA
- optimalizace logiky
- spojení FFs + LUTs do slices/CLB

Rozmístění

- rozmístění jednotlivých prvků po ploše FPGA na základě
 - hustoty vzájemných propojení bloků
 - požadovaných časových parametrů
 - constraint file (LOC, PAD)
- iterativní algoritmus (simulované žíhání)
- randomizovaný proces
- pokaždé to dopadne trochu jinak!
- pokaždé trochu jiná zpoždění, rychlosti, ...



Place & Route 2

Propojení

- iterativní, randomizovaný algoritmus
- cíl: propojit jednotlivé rozmístěné bloky
- randomizovaný proces, pokaždé trochu jiný výsledek!

P&R proces

- řešení NP těžké optimalizační úlohy randomizovaným algoritmem
- pokaždé trochu jiné časové parametry návrhu
- ruční optimalizace výsledku **velice** obtížná

Nezapomenout nastavit mapování padů

Zkontrolovat DRC report – špatně použité hodinové signály

Výstup – netlist, SDF file

SW podpora

- Xilinx, Altera, ... – výrobce FPGA



Fyzická syntéza

Syntezátor potřebuje znát rychlost obvodu

- zpoždění na buňkách – OK
- zpoždění na spojích – wire load model (fanout)
- zpoždění na spojích až 70% celku
- ASIC – jakž takž
- FPGA – nepredikovatelné výsledky díky P&R procesu
- timing neseďí → předělat a znovu!

Problém: syntéza nezná skutečné parametry po P&R!

Řešení

- syntezátor a P&R pracují v cyklech
- tzv. fyzická syntéza

SW podpora: Precision Physical Synthesis



Simulace

Odladění a ověření správnosti návrhu

- **RTL – bezčasová simulace**
 - **verifikace funkční správnosti**
 - **bez reálných časových parametrů FPGA**
- **post-synthesis simulace**
 - **simulujeme netlist po syntéze**
 - **nemají moc smysl**
 - **když nevíte, co zlobí (chyby v nástrojích)**
 - **nerealistická zpoždění**
- **back-annotated netlist (gate-level simulace)**
 - **simulujeme netlist a SDF po P&R**
 - **verifikace fyzikálních parametrů FPGA**
 - **se znalostí reálných časových parametrů (SIMPRIM)**
 - **problém: ModelSim XE**



SW podpora: ModelSim, ModelSimDesigner



Verifikace

Ověření správnosti návrhu

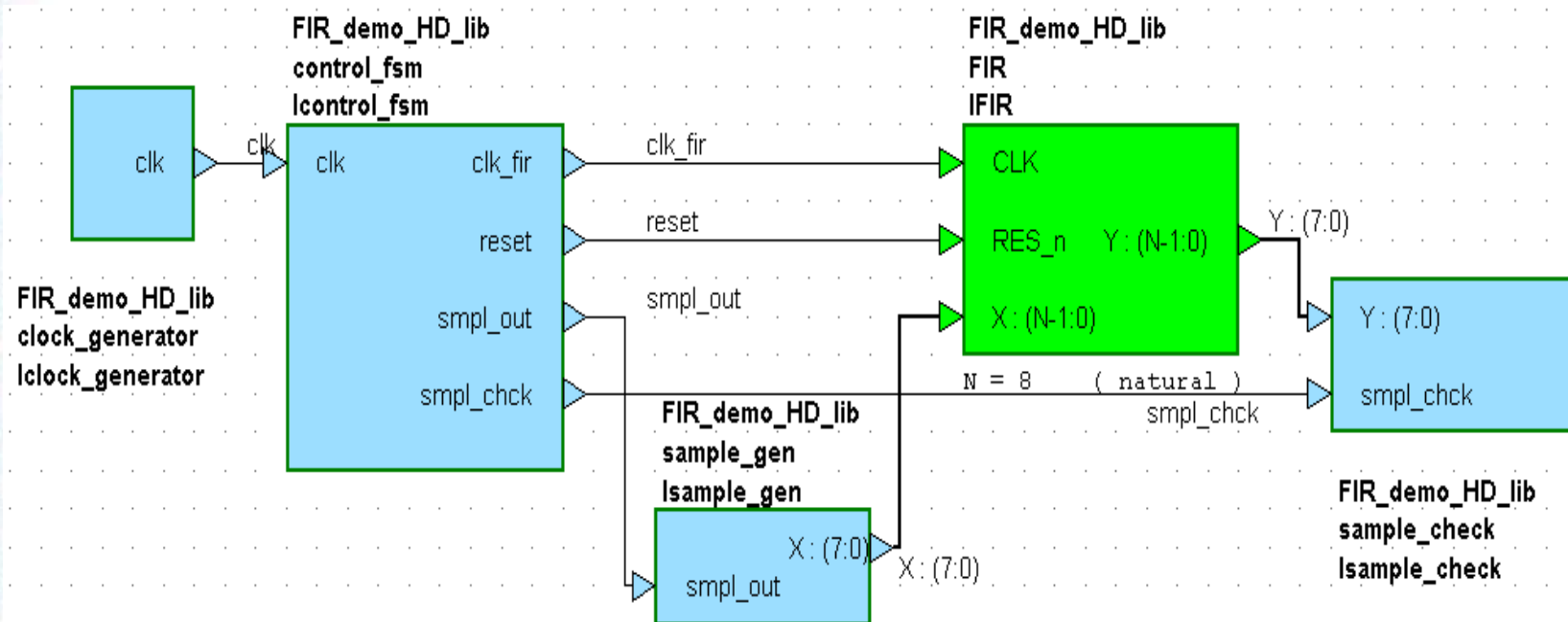
- čím dřív chybu objevíme, tím bude levnější ji odstranit
- základ: je nutno v "tom mít pořádek"
- sepsat seznam požadavků na design
 - "časovač má po resetu hodnotu 11110000"
 - "časovač s každým tiknutím hodin snižuje svůj obsah o jedničku"
 - "dosažení nuly v časovači vygeneruje přerušení"
 - "zápis do registrového pole na adresu 0x23 nastaví registr časovače"
 - ... atd
- verifikace jedině automaticky – yes/no výstup
- nejjednodušší pomocí simulace



Jednoduchý příklad top-levelu verifikačního prostředí

Hierarchický návrh testbenche

- verifikační prostředí
- DUV
- SW podpora: ModelSim Designer, HDL Designer





Měřítko kvality verifikace

Requirement Coverage

- kolik z požadavků verifikačního plánu už testy pokryly?
- dá se změřit (a mělo by být měřeno)

Code Coverage

- kolik % řádků kódu je během testování vykonáno?
- měří simulátor

Toggle Coverage

- kolik % ze všech spojů v designu změní během testů stav?
- měří simulátor

Function Coverage

- kolik % ze všech funkcí systému verifikace testuje?
- obtížně měřitelné

Konečný produkt

- Murphyho zákony.....





Zajímavé odkazy

www.fpgajournal.com **JOURNAL** FPGA and Structured ASIC TRENDS
TECHNOLOGY
TOOLS
TECHNIQUES

www.us.design-reuse.com

www.support.xilinx.com



Design And Reuse
*The Catalyst of Collaborative
IP Based SoC Design*



<http://lsiwww.epfl.ch/LSI2001/teaching/webcourse/toc.html>

DESIGN OF VLSI SYSTEMS

<http://tech-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf>

<http://www.asicentrum.cz>

<http://tech-www.informatik.uni-hamburg.de/vhdl/>

Hamburg VHDL archive





<http://www.asicentrum.cz>
mentor@asicentrum.cz

Copyright ©2005 ASICentrum, s.r.o.