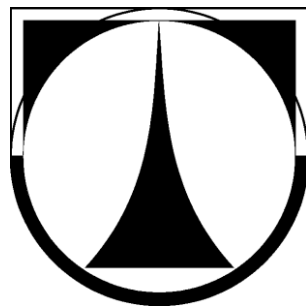


**Technická univerzita v Liberci**

**Fakulta mechatroniky, informatiky a  
mezioborových studií**



**Rekonfigurovatelný systém na FPGA obvodu**

**Autoreferát disertační práce**

**Liberec 2014**

**Ing. Tomáš Drahoňovský**

# Rekonfigurovatelný systém na FPGA obvodu

## Reconfigurable system on an FPGA

### Autoreferát disertační práce

**Studijní program:** P2612 Elektrotechnika a informatika

**Studijní obor:** 2612V045 Technická kybernetika

**Školící pracoviště:** Ústav informačních technologií a elektroniky  
Fakulta mechatroniky, informatiky a mezioborových studií  
Technická univerzita v Liberci  
Studentská 2/1402, 461 17, Liberec

**Autor:** Ing. Tomáš Drahoňovský

**Školitel:** Prof. Ing. Ondřej Novák, CSc.

## **Abstrakt**

Práce se zabývá tvorbou metodiky návrhu rekonfigurovatelného systému na FPGA obvodu. Tato metodika využívá pokročilých technik založených na částečné dynamické rekonfiguraci za účelem optimalizace rekonfigurovatelných systémů z hlediska flexibility, vyžadované paměti, času potřebného pro implementaci návrhu a množství logických zdrojů FPGA obvodu nezbytného pro vytvoření rekonfigurovatelného systému.

V textu jsou představeny základní pojmy z oblastí struktury a konfigurace FPGA obvodů, dále pak základní vlastnosti částečné rekonfigurace, relokace částečných konfiguračních souborů, vyčítání konfigurační paměti FPGA a zapisování dat do interních registrů obvodu.

Jádro práce představuje metodiku návrhu rekonfigurovatelného systému s využitím výše zmíněných technik. Dílčí části této práce jsou ověřeny na různých experimentech. V závěru jsou shrnuty výsledky jednotlivých přístupů a diskutovány přínosy použitých technik.

## **Klíčová slova**

FPGA, částečná dynamická rekonfigurace, vyčítání konfigurační paměti, relokace hardwarových úloh, optimalizace návrhu

## **Abstract**

This work is focused on a methodology of the reconfigurable system design implemented on an FPGA. This methodology uses advanced techniques based on a partial dynamic reconfiguration in order to optimize a reconfigurable system in terms of system's flexibility, memory requirements, implementation time requirements and logic sources consumption.

The text describes basics of the FPGA structure and important features of the dynamic partial reconfiguration, partial bitstream relocation and FPGA's configuration memory readback and FPGA's internal registers states restoration techniques.

The main part of the work presents a design methodology of the reconfigurable system where all mentioned techniques are supported. Individual parts of this work were verified on several applications with different sizes. Conclusion summarizes the results of the different approaches and discussed the benefits of the involved techniques.

## **Key worlds**

FPGA, partial dynamic reconfiguration, configuration memory readback, hardware task relocation, design optimization

## Obsah

Úvod.....	1
1 Obvody FPGA .....	2
1.3 Implementace návrhu do FPGA Xilinx .....	3
2 Částečná dynamická rekonfigurace FPGA .....	4
3 Cíle práce .....	5
4 Navržený rekonfigurovatelný systém .....	6
4.1 Relokace částečných konfiguračních souborů .....	7
4.2 Zpětné vyčítání konfigurační paměti .....	8
4.3 Zápis dat do interních registrů FPGA obvodu .....	9
5 Experimentální výsledky .....	11
5.1 Relokace částečných konfiguračních souborů .....	11
5.1.1 Zhodnocení experimentu .....	15
5.2 Zpětné vyčítání dat z konfigurační paměti .....	16
5.2.1 Zhodnocení experimentu .....	18
5.3 Zápis dat do interních registrů .....	19
5.3.1 Zhodnocení experimentu .....	20
5.4 Test na komplexním systému .....	21
5.4.1 Zhodnocení experimentu .....	25
Závěr .....	26
Seznam literatury .....	28
Vlastní publikace .....	28
Použitá literatura .....	29

# Úvod

Složité číslicové systémy jsou v dnešní době nedílnou součástí celé škály moderních zařízení od spotřební elektroniky přes medicínské přístroje, zařízení pro testování a měření až po aplikace pro vojenskou a leteckou techniku. Při vývoji a výrobě těchto zařízení jsou stále častěji využívána programovatelná hradlová pole FPGA (Field Programmable Gate Array) [18].

Díky možnosti konfigurace obvodu kdykoli během jeho životního cyklu FPGA vykazují velkou flexibilitu. Paralelní zpracování dat a možnost vytvoření hardwarového modulu šitého přímo na míru dané aplikaci zase zaručují vysoký výkon těchto obvodů. Avšak ve chvíli, kdy konkrétní návrh do FPGA umístíme, cílové využití se značně zúží. To je způsobeno tím, že pro každou specifickou úlohu je třeba specifická komponenta a počet takovýchto komponent je omezen velikostí použitého obvodu. Tento fakt je možné minimalizovat, pokud se daný systém navrhne jako rekonfigurovatelný. Díky částečné rekonfiguraci je FPGA obvod schopen provádět funkce, které by v něm nemohly být implementovány současně z důvodu nedostatku logických zdrojů obvodu, a zároveň je možné měnit funkce částí systému bez toho, aby byl jeho zbytek nějak ovlivněn.

Rekonfigurovatelný systém představuje kompromis mezi systémem vytvořeným přímo pro danou aplikaci (aplikačně specifickým), který se vykazuje vysokým výkonem, a univerzálním FPGA systémem (sestavou obsahující hardwarové vybavení pro více typů aplikací) vyznačujícím se vysokou flexibilitou.

Motivací této práce je ověření možností využití pokročilých technik rozšiřujících částečnou dynamickou rekonfiguraci pro optimalizaci návrhu rekonfigurovatelných systémů na FPGA obvodu a zhodnocení možných přínosů využití vytvořené návrhové metodiky.

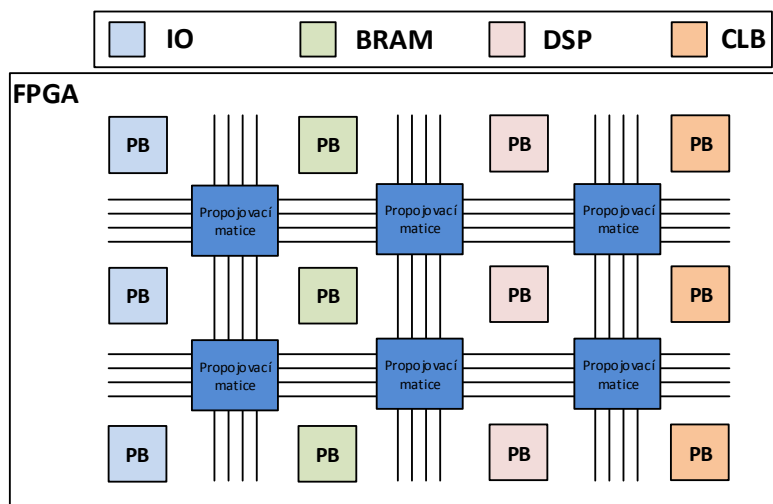
V první kapitole je vysvětlena struktura FPGA obvodů, možnosti jejich konfigurace a postup implementace návrhu. Kapitola dvě popisuje částečnou dynamickou rekonfiguraci a v kapitole tři jsou představeny cíle této disertační práce. Čtvrtá kapitola se zabývá vlastní metodikou návrhu rekonfigurovatelného systému a implementací jednotlivých částí. Pátá kapitola je věnována experimentům. V závěru jsou shrnuty dosažené výsledky a diskutovány přínosy této práce.

# 1 Obvody FPGA

FPGA jsou programovatelné polovodičové součástky určené pro zpracování digitálních signálů s možností konfigurace kdykoliv ve svém životním cyklu. Použití FPGA přináší zkrácení času potřebného pro vývoj aplikace, vysokou flexibilitu a vysoký výkon [20].

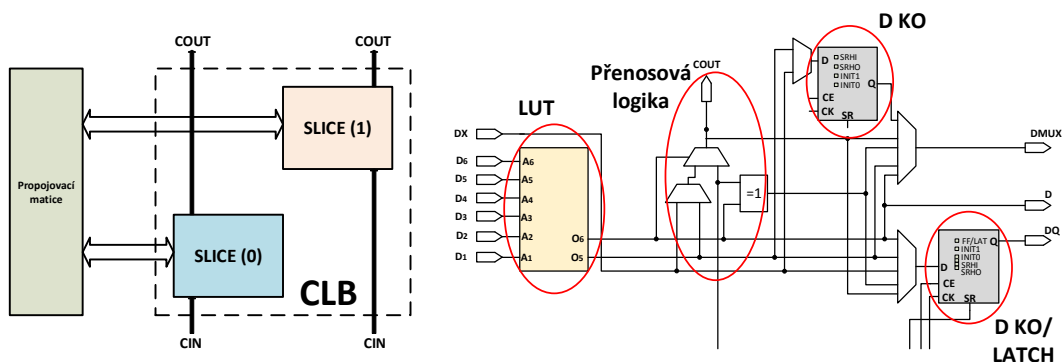
Následující popis se bude týkat obvodů z rodiny Virtex (konkrétně Virtex 6) od společnosti Xilinx. Tyto obvody se vyznačují vysokým výkonem s velkou hustotou vnitřní logiky (vysoký stupeň integrace). Kromě programovatelné logiky mají tyto obvody vestavěny hardwarové bloky pro implementaci složitých systémových funkcí, jako jsou násobičky, DSP (Digital Signal Processing) bloky, paměti RAM a procesorová jádra.

Virtex 6 je obvod vyráběný pomocí 40nm technologie. Je tvořen maticově uspořádanou strukturou různých programovatelných bloků (PB) vzájemně spojených nastavitelnou propojovací maticí. Toto je blokově naznačeno na Obr. 1.



Obr. 1: Principiální blokové schéma FPGA obvodu

Hlavním stavebním prvkem FPGA Virtex jsou tzv. CLB (Configurable Logic Block) bloky. CLB bloky jsou složeny ze dvou nižších funkčních celků (podbloků) nazývaných SLICE. Blokové uspořádání CLB a vnitřní zapojení části SLICE jsou naznačeny na Obr. 2.



Obr. 2: Struktura konfigurovatelného logického bloku

Funkce FPGA obvodu je dána obsahem konfigurační paměti typu SRAM. Konfigurační data jsou nahrávána v podobě konfiguračního souboru takzvaného bitstreamu. Jedná se o sekvenci 32bitových slov, obsahující jak konfigurační data pro dané FPGA, tak příkazy pro řídicí logiku obvodu. Rozlišujeme dva základní typy konfiguračních souborů – úplný a částečný. Úplný obsahuje konfigurační data pro celý FPGA obvod, zatímco částečný umožňuje nastavení pouze určité části obvodu.

Konfigurační data jsou organizována do tzv. konfiguračních rámců (frames). Tento rámec je nejmenší adresovatelný segment konfigurační paměti, ke kterému lze přistupovat. Pozice každého rámce uvnitř obvodu je dána unikátní adresou (frame address). Pokud je prováděna konfigurace více na sebe navazujících rámců současně, může být (při běžném nastavení je) adresa rámce v konfiguračním souboru uvedena pouze jednou a během konfigurace je automaticky inkrementována.

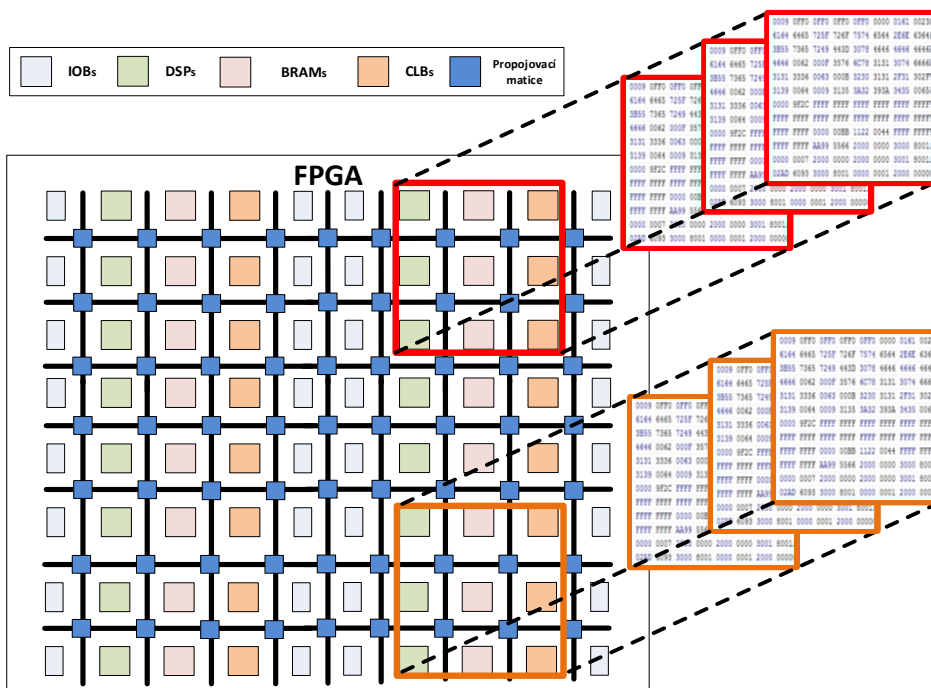
Samotná konfigurace těchto FPGA obvodů je prováděna nahráním konfiguračního souboru do vnitřní paměti obvodu. Tento soubor může být do FPGA stažen z paměti nebo může být nahrán s využitím externího zařízení (mikroprocesor, PC, tester atd.).

### ***1.3 Implementace návrhu do FPGA Xilinx***

Pro získání konfiguračního souboru je nutné, aby návrh popsáný pomocí nějakého HDL (Hardware Description Language) jazyka prošel několika implementačními kroky. Nejprve je třeba návrh (většinou vytvořený v jazyce VHDL, Verilog, případně jako schéma) syntetizovat. Tím dojde k převodu informace o požadovaném chování (behaviorální popis) návrhu na strukturní netlist a optimalizaci návrhu pro dané zařízení. U obvodů Xilinx se syntéza provádí pomocí nástroje XST (Xilinx Synthesis Tool). Dalším krokem je překlad strukturního netlistu nástrojem zvaným NGDBuild. Ten vytvoří tzv. NGD (Native Generic Database) soubor, který obsahuje popis návrhu na úrovni obsažených logických elementů (hradla, LUT elementy, klopné obvody, RAM paměti atd.). Jednotlivé logické elementy jsou dále nástrojem MAP namapované a umístěné do konkrétních logických prvků (CLB, IOB, DSP atd.) obsažených na daném FPGA. Po namapování návrhu jsou tyto komponenty vzájemně propojeny nástrojem PAR (Place And Route). Posledním krokem implementačního řetězce je vytvoření konfiguračního souboru nástrojem BitGen (BITstream GENerator).

## 2 Částečná dynamická rekonfigurace FPGA

Částečná dynamická rekonfigurace (DPR – Dynamic Partial Reconfiguration) je technika umožňující za běhu obvodu měnit některé jeho části, přičemž zbytek obvodu není touto změnou nijak ovlivněn [19]. Tato technika je v poslední době kromě obvodů firmy Xilinx podporována u některých FPGA jiných výrobců (např. Altera Startix V). Obvody firmy Xilinx však mají tuto techniku mnohem propracovanější, ať už po stránce podpory návrhu rekonfigurovatelného systému, nebo z hlediska samotné konfigurace jednotlivých částí obvodu. Z tohoto důvodu jsou v této práci využívány pouze obvody firmy Xilinx.



Obr. 3: Blokově naznačený princip částečné rekonfigurace

Princip DPR je blokově naznačen na Obr. 3. Vnitřní logika FPGA je rozdělena na statickou a rekonfigurovatelnou, někdy též nazývanou dynamickou část. V dynamické části je možné vytvářet rekonfigurovatelné oblasti, kam jsou umísťovány příslušné rekonfigurovatelné moduly. Funkce dané oblasti může být pozměněna nahráním jiného modulu. Nahrání nového modulu je provedeno stažením nového částečného konfiguračního souboru do konfigurační paměti obvodu. DPR v podstatě umožňuje časově multiplexovat části hardwaru na jednom FPGA.

Kromě snížení velikosti, snížení nákladů a zvýšení výkonu částečná rekonfigurace umožňuje nové typy návrhů pro FPGA, které by bez částečné rekonfigurace nebylo možné realizovat [19]. Na druhou stranu použití DPR způsobuje větší složitost návrhu, zvyšuje čas potřebný pro implementaci návrhu, vnáší do systému větší nároky na paměť (paměť pro uložení částečných konfiguračních souborů) a způsobuje navýšení potřebných logických prvků (logic overhead).



### 3 Cíle práce

Cílem této práce je zpracování koncepce návrhové metodiky dynamicky rekonfigurovatelného systému na FPGA obvodu. Podstata navržené metodiky spočívá v reflexi všech výhod částečné dynamické rekonfigurace (vysoký výkon, nízká spotřeba, vysoká flexibilita, vysoká provozuschopnost atd.) a maximální minimalizaci nevýhod této techniky (složitost návrhu, zvýšené nároky na paměť, požadavek na dodatečné hardwarové vybavení obvodu, doba nezbytná pro implementaci návrhu). Pro splnění cíle této práce se předpokládá splnění následujících dílčích cílů:

- analýza stávajících řešení rekonfigurovatelných systémů,
- seznámení s postupem návrhu rekonfigurovatelného systému,
- seznámení s pokročilými technikami založenými na částečné rekonfiguraci,
- návrh a vytvoření metody pro relokaci částečných konfiguračních souborů,
- návrh systémů využívajících zpětné vyčítání konfigurační paměti FPGA obvodu,
- návrh systémů využívajících techniku zapsání dat z konfigurační paměti do interní logiky FPGA,
- implementace zmíněných technik do reálného FPGA systému,
- experimentální ověření funkce a předpokládaných vlastností jednotlivých technik.

Splněním cílů této práce umožníme vytvoření rekonfigurovatelného systému na FPGA obvodu s podporou částečné dynamické rekonfigurace, relokace rekonfigurovatelných modulů, zpětného vyčítání konfigurační paměti a zápisu dat z konfigurační paměti do interní logiky FPGA obvodu. Kombinace těchto technik umožní vytvoření komplexního, univerzálního a vysoce flexibilního systému na FPGA. Jedná se o systém podporující přístup s minimálními nároky na paměť pro uložení částečných konfiguračních souborů a s možností rekonfigurace jednotlivých částí tohoto systému včetně aktuálních hodnot interních registrů obvodu.

## 4 Navržený rekonfigurovatelný systém

S rekonfigurovatelnými systémy na FPGA obvodu se poslední dobou setkáváme stále častěji. Je to způsobeno především větší podporou této techniky v rámci standardních návrhových a implementačních nástrojů jednotlivých výrobců FPGA obvodů. Za nejpropracovanější lze považovat návrhové nástroje firmy Xilinx.

Způsobů využití částečné rekonfigurace existuje celá řada, od systémů určených pro zpracování velkého množství dat v reálném čase přes aplikace zaměřené na testování FPGA obvodů až po systémy s možností tolerance poruch.

Zahrnutí rekonfigurace do návrhu má ve všech případech za následek navýšení potřebné logiky (v závislosti na množství pinů rekonfigurovatelných oblastí) a navýšení potřebné paměti (v závislosti na množství funkcí jednotlivých oblastí).

Využitím techniky relokace hardwarových modulů lze množství potřebných konfiguračních souborů omezit na jeden soubor na každý modul implementovaný v obvodu. Nejmenšího možného počtu potřebných částečných konfiguračních souborů dosáhneme kombinací techniky relokace hardwarových modulů s technikou zpětného vyčítání konfigurační paměti obvodu, kdy není zapotřebí konfigurační data ukládat do systémové paměti.

Dalším neduhem částečné rekonfigurace je navýšení potřebného hardwarového vybavení obvodu, které se projevuje hlavně u systémů, kde mají jednotlivé hardwarové komponenty velký počet vstupních a výstupních pinů, např. v [11]. Zde je ke každému modulu připojeno testovací vybavení. Nebo např. ve [13] a [14], kde je řešena synchronizace jednotlivých modulů jejich fyzickým propojením.

Použití datových sběrnic propojujících jednotlivé moduly zvyšuje náročnost propojení systému a navyšuje režii logických prvků potřebných pro částečnou rekonfiguraci. Dále způsobuje zhoršení časování systému a navyšuje čas potřebný pro jeho implementaci.

Použitím technik vyčítání konfigurační paměti a zápisu dat do interních registrů FPGA obvodu se lze oprostít od nutnosti tohoto fyzického propojení jednotlivých komponent. Prezentovaná metodika návrhu rekonfigurovatelného systému počítá s využitím všech zmíněných technik.

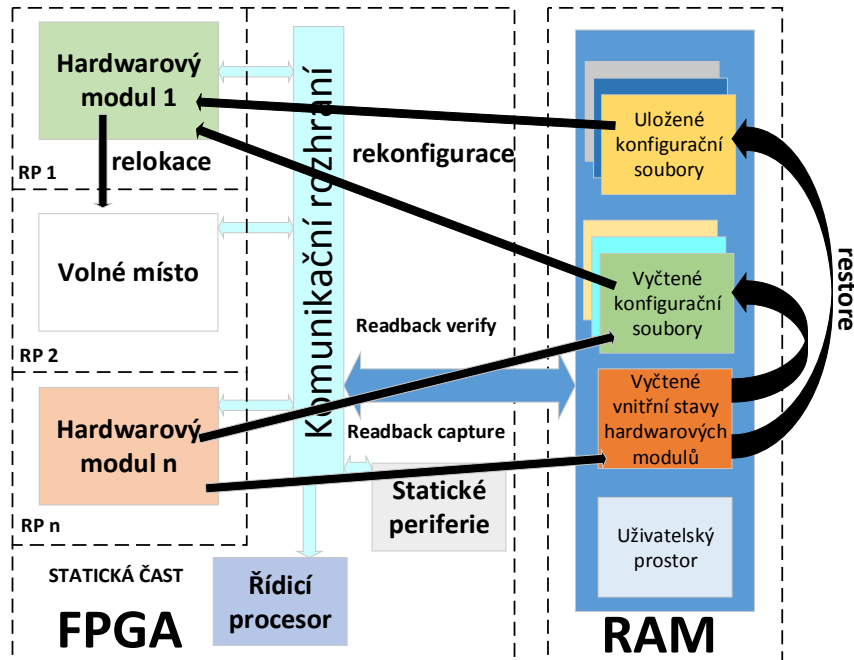
Obr. 4 zobrazuje obecné schéma rekonfigurovatelného systému navrženého s využitím vytvořené metodiky. Návrh je rozdělen na jednu statickou a  $n$  rekonfigurovatelných částí ( $n$  – závisí na velikosti jednotlivých RP a na velikosti použitého obvodu).

Systém má k dispozici různé varianty hardwarových modulů, které jsou uloženy v paměti jako částečné konfigurační soubory nebo mohou být získány z konfigurační paměti obvodu technikou zpětného vyčítání.

Systém je navržen tak, aby využíval techniku DPR doplněnou o možnost relokace rekonfigurovatelných modulů, zpětného vyčítání konfigurační paměti a zápisu dat do interních registrů v obvodu.

Díky relokaci hardwarových modulů je možné využít jednotlivých částečných konfiguračních souborů na více místech v obvodu. S použitím zpětného vyčítání je možné relokovat komponenty získané přímo z konfigurační paměti obvodu. Zápis dat

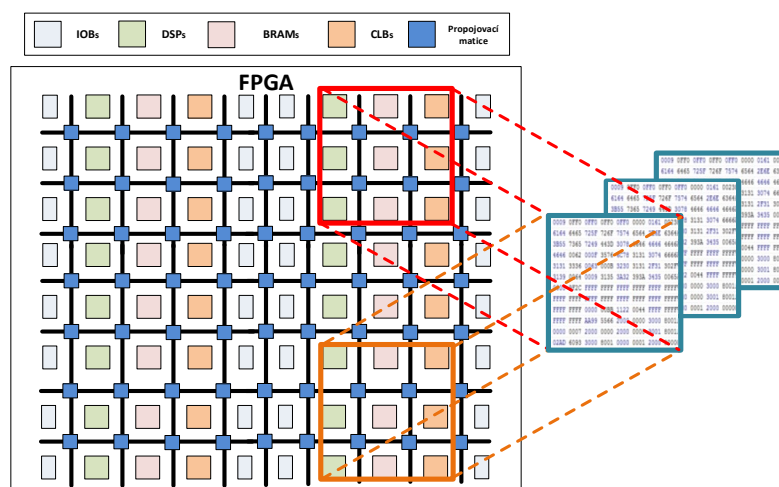
do interních registrů obvodu lze použít např. pro synchronizaci jednotlivých rekonfigurovatelných modulů, vkládání testovacích dat na libovolné místo v obvodu a relokaci jednotlivých komponent včetně jejich vnitřních stavů. Využití zmíněných technik má za následek zvýšení flexibility a snížení paměťových nároků systému.



Obr. 4: Navržený rekonfigurovatelný systém (obecné blokové schéma)

#### 4.1 Relokace částečných konfiguračních souborů

Relokace částečných konfiguračních souborů (PBR – Partial Bitstream Relocation) je technika založená na DPR. Tato technika umožňuje přesouvání jednotlivých hardwarových modulů umístěných původně v jedné části FPGA do jiné části obvodu. Zjednodušeně lze říci, že částečný konfigurační soubor vytvořený pro jednu konkrétní část obvodu je možné použít na více místech. Základní princip PBR techniky je naznačen na Obr. 5.



Obr. 5: Principiální blokové schéma PBR

Podpora techniky PBR je jedním ze základních principů využitých v této práci. Díky PBR lze dosáhnout zvýšení flexibility FPGA návrhu, snížení paměťových nároků systému v porovnání se systémy využívajícími standardní DPR a snížení časových nároků na implementaci návrhu, protože na vytvoření méně PB potřebujeme méně času.

Postup pro vytvoření FPGA návrhu s podporou relokace částečných konfiguračních souborů není v zásadě příliš odlišný od návrhu systému, který je založený na standardní DPR. Během tohoto návrhu však musejí být splněny určité podmínky a jednotlivé oblasti, kam má být daný RM relokován, musejí splňovat jisté parametry.

Tyto návrhové podmínky lze shrnout v několika následujících bodech:

- stejná velikost všech RP,
- stejný počet, typ a rozmístění logických zdrojů v jednotlivých RP,
- žádné statické signály v jednotlivých RP (kromě signálů typu long line),
- žádné rekonfigurovatelné signály ve statické části návrhu,
- stejně vedené vodiče propojující statickou a rekonfigurovatelnou část obvodu.

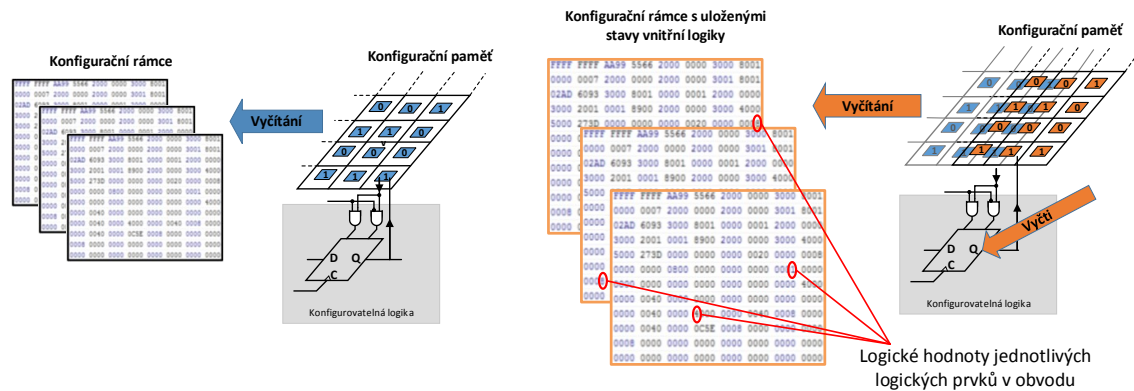
Metodika pro relokaci, která je využita v této práci, je založena na transformaci přístupu využívajícího oddělovacích elementů na přístup využívající oddělovací makra. Tato transformace je realizována pomocí implementačních omezujících podmínek. Pro realizaci systému podporujícího relokaci bylo nezbytné upravit běžný implementační řetězec.

Proces implementace je doplněn o krok, kdy jsou na úrovni VHDL návrhu přidány LUT elementy tvořící polovinu oddělovacího makra. Takovýto doplněný návrh poté projde standardní syntézou. Během překladu nástrojem NGDBUILD jsou do návrhu automaticky vloženy oddělovací elementy. Před spuštěním mapování (MAP) jsou do návrhu vloženy omezující podmínky zapisované do UCF – jedná se o omezující podmínky pro kontrolu velikosti a polohy rekonfigurovatelných oblastí (AREA\_GROUP, RANGE) a omezující podmínky pro kontrolu polohy jednotlivých LUT elementů (LOC, BEL). Dále jsou do UCF souboru vkládány omezující podmínky pro odstranění statických signálů z rekonfigurovatelné části systému (PRIVATE) a naopak (CONTAINED). Jako poslední jsou zapisovány DIRT omezující podmínky. Ač se jedná o podmínky kontrolující propojení, jsou uváděny v UCF souboru, tedy před spuštěním mapování. Po namapování do daného FPGA obvodu jsou do návrhu přidány omezující podmínky pro kontrolu priority (podmínka PRIORITIZE) propojení vodičů spojujících jednotlivé LUT elementy tvořící oddělovací makro.

## **4.2 Zpětné vyčítání konfigurační paměti**

Další technikou, jejíž využití je pro tuto práci nezbytné, je zpětné vyčítání konfigurační paměti. Tato technika umožňuje vyčtení všech dat z interní konfigurační paměti FPGA obvodu. Vyčítání konfigurační paměti rozdělujeme na dva základní typy. Prvním z nich je tzv. Readback Verify, s jehož využitím lze provádět ověření správnosti dat uložených v konfigurační paměti obvodu. Druhým typem je tzv. Readback Capture – tento typ se používá pro vyčítání aktuálního stavu všech paměťových elementů (LUT elementy, klopné obvody, IOB registry, distribuovaná RAM paměť a BRAM paměť) v obvodu.

Takto vyčtená data se využívají při odladování návrhu a funkční verifikaci [17], [20]. Principy obou typů této techniky jsou blokově naznačeny na Obr. 6.



**Obr. 6: Blokové naznačení techniky Readback Verify a Readback Capture**

Pro zavedení techniky zpětného vyčítání konfigurační paměti do systému je třeba nastavit generátor konfiguračních souborů (BitGen) tak, aby vytvořené soubory umožňovaly přístup k jejich datům (nezbytné pro DPR a zpětné vyčítání). Tuto funkci nastavuje parametr security. Dále je třeba povolit zpětné vyčítání konfigurační paměti parametrem readback. Zmíněné parametry generátoru konfiguračních souborů lze zadat v sekci nastavení konfigurace (Set Configuration). Při zápisu např. do příkazové řádky jsou uvozeny pomocí parametru -g. Způsob zápisu je následující:

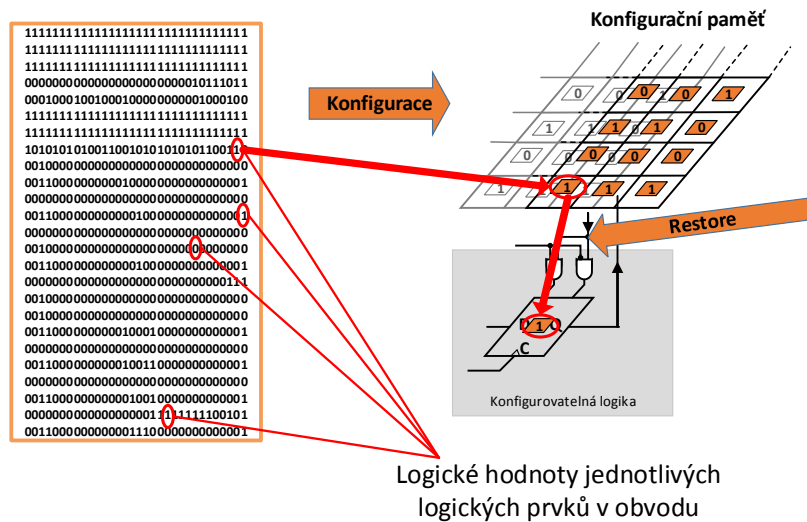
-g Security: none  
-g ReadBack: yes

Typ capture je jakousi nadstavbou typu prvního (verify), kdy jsou vyčtená konfigurační data doplněna o stavy interních registrů, které se ukládají do paměti v přesně definovaných okamžicích, po zadání příkazu vyčti (capture).

Zadání příkazu pro uložení obsahu interních registrů do paměti lze provést zapsáním příkazu capture přímo do řídicího registru obvodu. K tomu, abychom mohli určit, kterému internímu registru odpovídá který bit ve vyčtených datech, se využívá tzv. alokační soubor konfigurační paměti (memory allocation file). V tomto souboru je uložena informace o tom, kde v obvodu (konkrétní SLICE) se daný registr nachází.

### 4.3 Zápis dat do interních registrů FPGA obvodu

Hodnota každého logického prvku schopného uchovat logickou informaci (registr) v obvodu FPGA je v konfigurační paměti obvodu reprezentována jedním bitem (INT0/INT1). Technika zápisu dat, neboli tzv. restore umožňuje přístup, kdy je hodnota těchto bitů reprezentujících vnitřní stavy zapsána do FPGA obvodu. Tento proces je principiálně naznačen na Obr. 7.



**Obr. 7: Princip techniky zápisu dat do interních registrů**

Přepsání hodnot interních registrů se provádí zadáním příkazu: zapiš (restore), kdy dojde k nastavení signálu GSR (Global Set/Reset) a tím k aktualizaci všech bitů (reprezentujících vnitřní hodnoty) v konfigurační paměti.

Zadání příkazu pro zápis je možné provést zapsáním příkazu GRESTORE přímo do řídicího (CMD) registru FPGA obvodu. Vždy po zapsání tohoto příkazu dojde k přepsání vnitřních stavů všech klopných obvodů.

## 5 Experimentální výsledky

Pro účely testování bylo vytvořeno několik rekonfigurovatelných systémů. Při jejich návrhu byl kladen důraz na otestování správné funkce jednotlivých technik, které jsou v popisované metodice využívány, a následně byla otestována možnost jejich kombinace. Jednotlivé experimenty jsou seřazeny chronologicky s postupným vývojem této práce.

### 5.1 Relokace částečných konfiguračních souborů

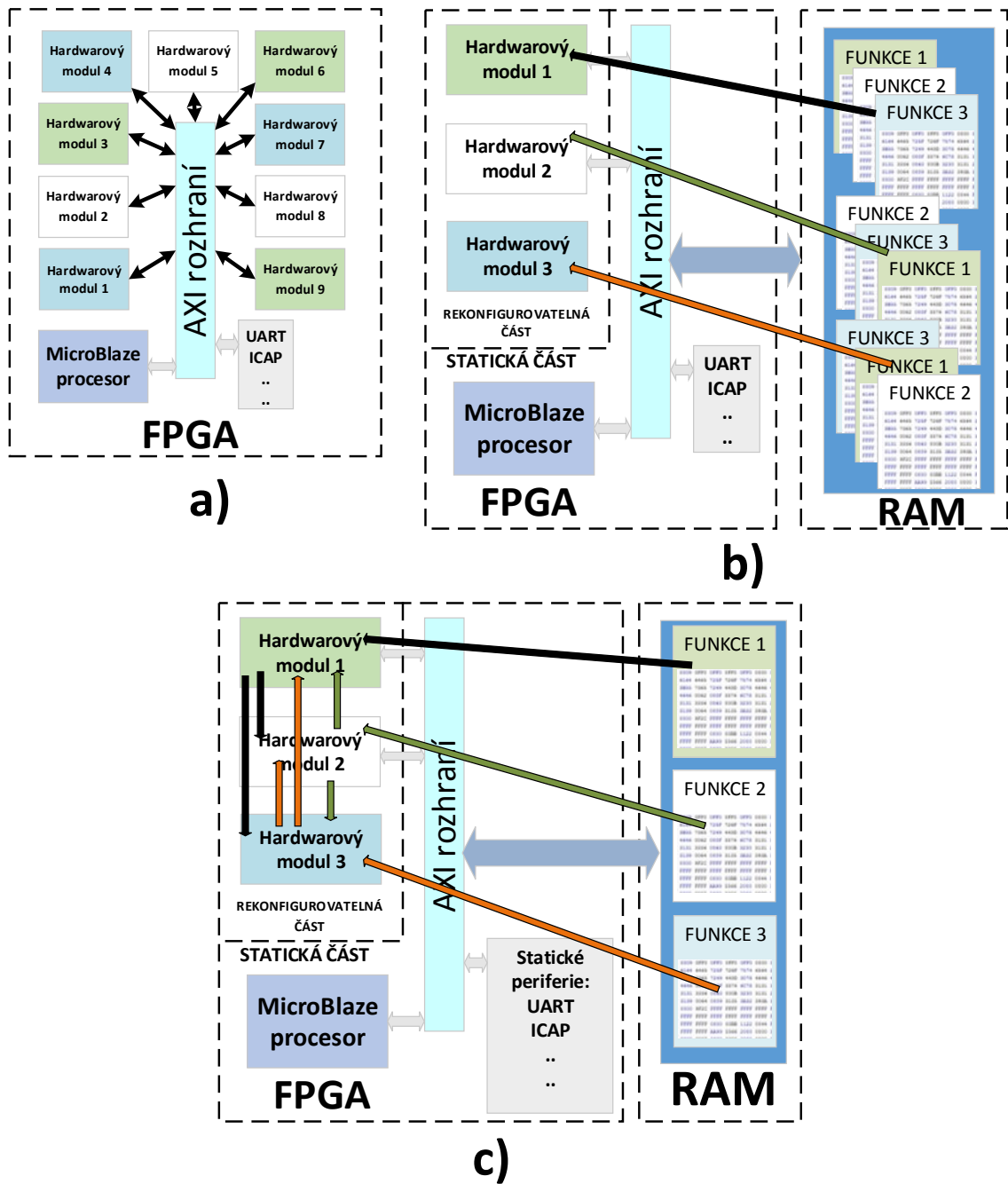
Pro testování funkce a vlastností relokace byl vytvořen systém, na kterém lze demonstrovat výhody a nevýhody této techniky a odhadnout mez její použitelnosti. Mezi použitelnosti je myšleno určení typů systémů, pro které je tato technika vhodná z hlediska spotřeby logických zdrojů, potřebné paměti atd. Při tomto experimentu byl systém podporující relokaci srovnáván se systémem bez jakékoli podpory rekonfigurace a se systémem využívajícím standardní částečnou rekonfiguraci založenou na signálovém rozhraní tvořeném oddělovacími elementy. Všechny tři systémy mají stejnou funkčnost ve smyslu tří aktivních modulů ve stejný okamžik s tím, že tyto moduly mohou vykonávat různé funkce.

Sledovanými parametry je množství logiky potřebné pro vytvoření celého návrhu, množství paměti, které je nezbytné k uložení částečných konfiguračních souborů, a čas potřebný pro implementaci návrhu (doba od začátku implementace až po vytvoření konfiguračních souborů).

Množství potřebné logiky je závislé především na počtu hardwarových modulů, které jsou v systému využívány. Velikost paměti je úměrná počtu a velikosti rekonfigurovatelných modulů a množství funkcí, které mají tyto moduly vykonávat. Čas potřebný pro implementaci návrhu je závislý na celkové velikosti FPGA, na složitosti návrhu a na výpočetním výkonu, s jehož využitím je návrh zpracováván. S množstvím potřebné logiky ještě velmi úzce souvisí potřeba dodatečných logických zdrojů při použití rekonfigurace, která je přímo úměrná počtu vstupních a výstupních pinů jednotlivých modulů. Na každý pin je třeba umístit přemostění mezi statickou a dynamickou částí systému.

Na Obr. 8 jsou vyobrazena bloková schémata tří testovacích systémů. První představuje systém bez jakékoli podpory rekonfigurace (Obr. 8/a), druhý je systém s podporou standardní částečné rekonfigurace (Obr. 8/b) a třetí blokové schéma představuje systém s podporou relokace částečných konfiguračních souborů (Obr. 8/c).

V Tab. 1 je porovnáno celkové množství logiky (LUT elementy) potřebné pro vytvoření výše popsaných systémů s různým počtem pinů u použitých hardwarových modulů. Pro jednoduchost je velikost hardwarových modulů ve všech systémech stejná – 640 LUT elementů = dva rekonfigurovatelné rámce.



Obr. 8: Blokové testovací aplikace

Tab. 1: Přehled potřebných LUT elementů pro jednotlivé typy systémů

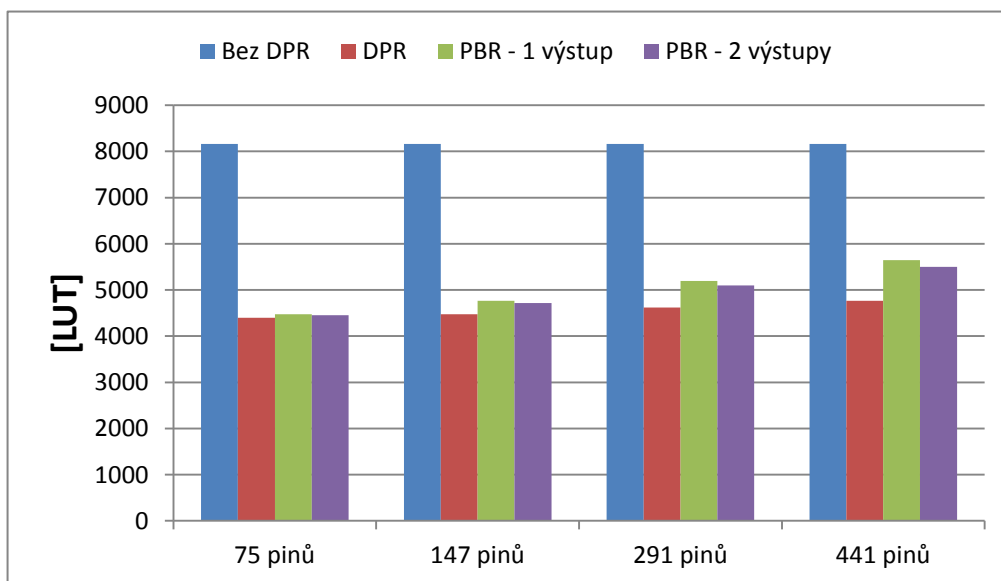
Počet pinů – celkem/1 modul		75/25	147/49	291/97	441/147
<b>Systém bez DPR</b>	logika [LUT]	8165	8165	8165	8165
<b>DPR</b>	logika [LUT]	4400	4472	4616	4766
<b>PBR – LUT jeden výstup</b>	logika [LUT]	4475	4766	5198	5648
<b>PBR – LUT oba výstupy</b>	logika [LUT]	4450	4717	5101	5501



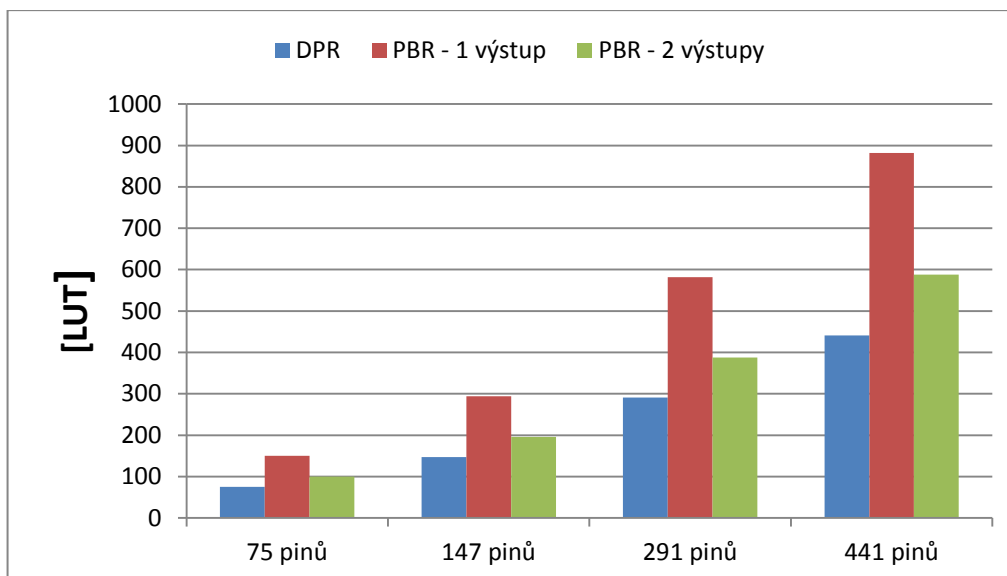
Celkové množství potřebné logiky je závislé na množství dodatečného hardwarového vybavení nezbytného pro vytvoření rekonfigurovatelného systému. Množství dodatečného hardwaru je shrnuto v Tab. 2. Počet pinů je uváděn jako množství pinů všech rekonfigurovatelných komponent v systému / množství pinů jedné komponenty. Počet pinů v prvních třech sloupcích odpovídá komponentám se dvěma n-bitovými vstupy, jedním n-bitovým výstupem a jedním řídicím signálem. Jedná se tedy o komponenty s osmi, šestnácti a 32bitovými vstupy a výstupy. V posledním sloupci je uvedeno množství pinů (147) potřebné pro připojení jedné komponenty na AXI rozhraní. Grafické znázornění celkem spotřebovaných LUT elementů je vyobrazeno na Obr. 9. Grafická závislost postupného navyšování potřebné logiky s narůstajícím počtem pinů je vyobrazena na Obr. 10.

**Tab. 2: Přehled dodatečného hardwarového vybavení**

Počet pinů – celkem/1 modul		75/25	147/49	291/97	441/147
<b>Systém bez DPR</b>	režie [LUT]	0	0	0	0
<b>DPR</b>	režie [LUT]	75	147	291	441
<b>PBR – LUT jeden výstup</b>	režie [LUT]	150	294	582	882
<b>PBR – LUT oba výstupy</b>	režie [LUT]	100	196	388	588



**Obr. 9: Grafické znázornění celkové spotřeby logických zdrojů v obvodu**



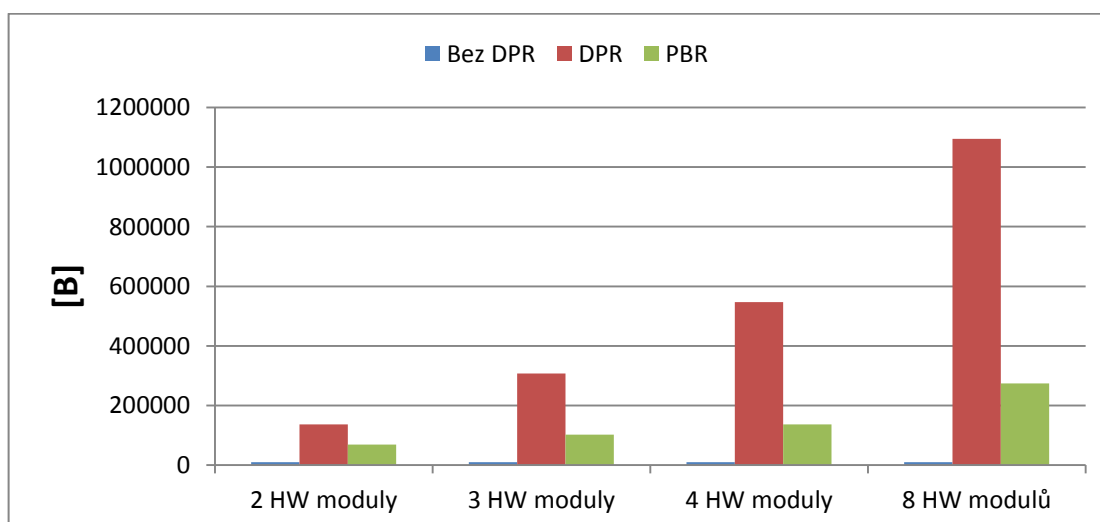
Obr. 10: Grafické znázornění navyšování dodatečných logických prvků v závislosti na počtu pinů u rekonfigurovatelných systémů

V následujících tabulkách je uveden přehled ostatních sledovaných parametrů. Uvedené hodnoty odpovídají systému, ve kterém dochází k rekonfiguraci, potažmo relokaci celých IP komponent připojených na rozhraní AXI (tj. jednotlivé moduly mají 147 pinů).

Tab. 3: Přehled potřebné paměti

Počet HW modulů		2	3	4	8
<b>System bez DPR</b>	paměť [B]	0	0	0	0
<b>DPR</b>	paměť [B]	136 800	307 800	547 200	1 094 400
<b>PBR</b>	paměť [B]	68 400	102 600	136 800	273 600

Tab. 3 shrnuje množství systémové paměti potřebné pro uložení částečných konfiguračních souborů v závislosti na počtu rekonfigurovatelných hardwarových modulů. Graficky je tato závislost zobrazena na Obr. 11.

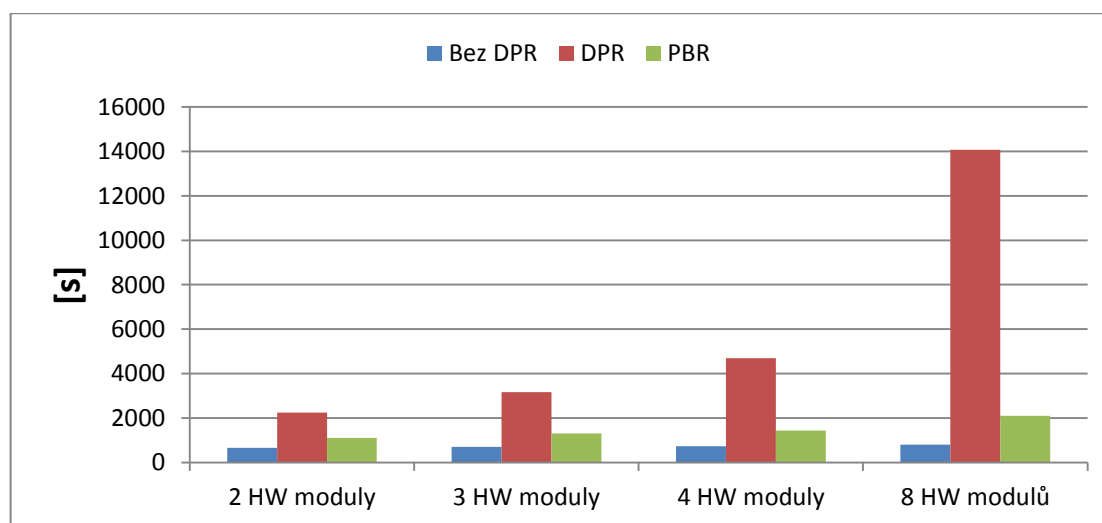


Obr. 11: Grafické znázornění nárůstu potřebné paměti bez a s použitím relokace

Posledním sledovaným parametrem je čas potřebný pro implementaci daného návrhu. Jak již bylo zmíněno výše, do tohoto času není započítána délka přípravy návrhu. Čas implementace zahrnuje pouze dobu běhu implementačních nástrojů. Časové nároky pro jednotlivé typy systémů jsou shrnuty v Tab. 4 a graficky znázorněny na Obr. 12.

**Tab. 4: Přehled časových nároků implementace**

Počet HW modulů		2	3	4	8
<b>Systém bez DPR</b>	čas [s]	657	702	736	806
<b>DPR</b>	čas [s]	2 246	3 165	4 694	14 072
<b>PBR</b>	čas [s]	1 099	1 305	1 439	2 102



**Obr. 12: Grafický přehled času potřebného pro implementaci návrhu**

### 5.1.1 Zhodnocení experimentu

Využití techniky rekonfigurace FPGA přináší snížení počtu logických zdrojů potřebných pro implementaci návrhu oproti systému, kde není rekonfigurace využita (viz Obr. 9).

Použití vytvořené metodiky pro relokaci částečných konfiguračních souborů mírně navyšuje počet potřebného hardwarového vybavení (Obr. 10), avšak oproti standardní částečné rekonfiguraci umožňuje vytvořená metodika značné snížení paměťových nároků (Obr. 11) a notně snižuje délku časového intervalu potřebného pro implementaci návrhu (Obr. 12).

Pro určení použitelnosti popsané metodiky pro relokaci částečných konfiguračních souborů je třeba zvážit, zda je tato technika pro daný konkrétní návrh výhodná či ne.

U návrhů s velkým množstvím malých rekonfigurovatelných oblastí a velkým počtem vstupních a výstupních pinů může použití této techniky způsobit nárůst dodatečného hardwarového vybavení přesahujícího svou velikostí množství logiky potřebné pro implementaci samotných hardwarových modulů. V takovýchto případech není použití relokace příliš vhodné vzhledem k tomu, že u malých modulů je množství

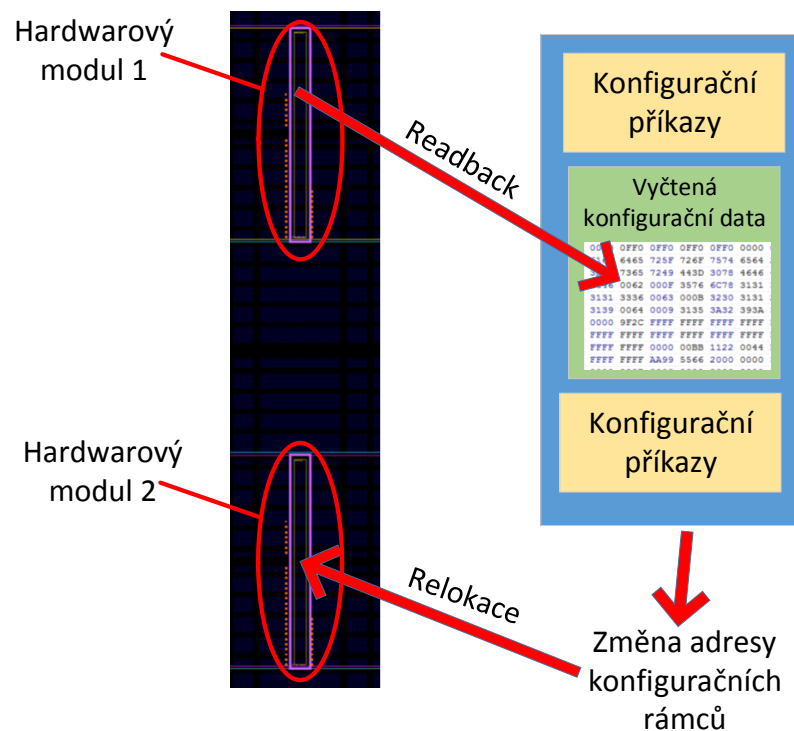
ušetřené systémové paměti neúměrné k množství dodatečné logiky a ke zvýšené složitosti návrhu.

U návrhů, kde jsou rekonfigurovány velké hardwarové moduly, lze považovat dodatečné hardwarové vybavení za zanedbatelné a ostatní výhody relokace částečných konfiguračních souborů převáží zvýšenou složitost návrhu.

## 5.2 Zpětné vyčítání dat z konfigurační paměti

Techniku zpětného vyčítání konfigurační paměti lze použít i pro jiné účely, než pro které byla vytvořena (tj. ověření správnosti konfiguračních dat, funkční verifikace, odladování návrhu). Pokud zkombinujeme techniku relokace s technikou zpětného vyčítání, výsledná platforma může být použita pro snížení objemu konfiguračních dat, které je třeba uchovávat v systémové paměti. Navíc tato kombinace způsobuje další zvýšení flexibility systému, ve kterém je použita.

Moduly, které chceme relokovat (tj. zaměnit funkci jednoho modulu za funkci jiného modulu), jsou do FPGA nahrány již v rámci úplného konfiguračního souboru a pomocí techniky zpětného vyčítání konfigurační paměti můžeme získat všechna data popisující daný modul, změnit adresy konfiguračních rámců a takto upravená data nahrát na jiné místo v obvodu. Z konfigurační paměti FPGA lze vyčíst data pro nastavení veškeré interní logiky a jejího propojení. Informace potřebné pro nastavení řídicích registrů obvodu v konfigurační paměti k dispozici nejsou. Je tedy nezbytné tyto řídicí příkazy uložit do systémové paměti spolu s adresami jednotlivých rekonfigurovatelných oblastí.



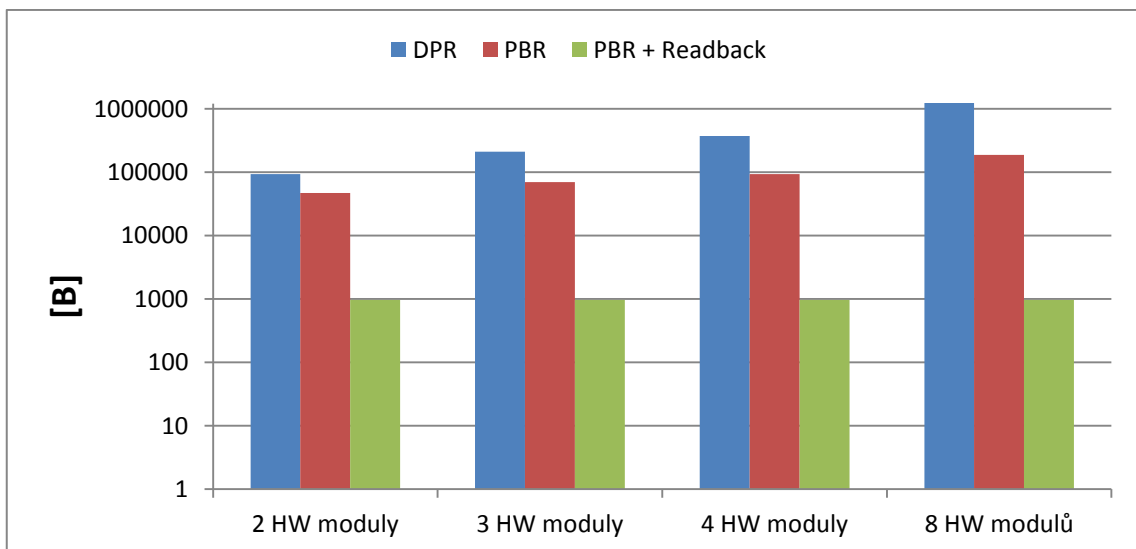
Obr. 13: Principiální schéma testovací aplikace získávající konfigurační data s využitím techniky zpětného vyčítání konfigurační paměti

Využití techniky zpětného vyčítání konfigurační paměti FPGA pro získávání konfiguračních dat bylo otestováno na systému obsahujícím dvě hardwarové komponenty. Pro jednoduchost testovací aplikace byly zvoleny hardwarové moduly zabírající oblast pouhých dvou rekonfigurovatelných rámců. Oba hardwarové moduly obsahují 32bitový čítač, přičemž první modul čítá vzestupně a druhý modul umožňuje čítání sestupně. Princip tohoto testovacího systému je naznačen na Obr. 13. Obě hardwarové komponenty jsou vytvořeny pomocí výše popsané metodiky pro relokaci. Tento fakt má za následek, že lze použít konfigurační data určená pro modul jedna i pro nastavení modulu dva. Popsaná aplikace umožňuje vzájemně přesouvat funkce mezi jednotlivými komponentami bez toho, aby bylo nutné uchovávat konfigurační data trvale v paměti.

Množství paměti potřebné pro trvalé uložení různého počtu hardwarových modulů je uvedeno v Tab. 5 a graficky znázorněno na Obr. 14. Systémová paměť je využita pouze v době, kdy se provádí samotná relokace (vyčtená konfigurační data musí být někde uložena). Trvale je uloženo pouze zanedbatelné množství příkazů pro řídicí registry.

**Tab. 5: Přehled paměti potřebné pro trvalé uložení částečných konfiguračních souborů (vztaženo k HW modulům o velikosti dvou rekonfigurovatelných rámců)**

Počet HW modulů		2	3	4	8
<b>DPR</b>	paměť [B]	93 312	209 952	373 248	1 492 992
<b>PBR</b>	paměť [B]	46 656	69984	93 312	186 624
<b>PBR&amp;Readback</b>	paměť [B]	960	960	960	960



**Obr. 14: Grafické znázornění paměti potřebné pro trvalé uložení částečných konfiguračních souborů**

## 5.2.1 Zhodnocení experimentu

Kombinací techniky relokace částečných konfiguračních souborů s technikou zpětného vyčítání paměti FPGA je možné vytvořit vysoce flexibilní rekonfigurovatelný systém vyžadující minimální množství systémové paměti nutné pro ukládání konfiguračních dat.

Využitelnost výše popsané techniky je závislá na tom, jak rychle je třeba v dané aplikaci relokaci provést. Rychlost vyčítání a následné konfigurace je v tomto případě značně ovlivněna zvoleným přístupem k paměti. Přístup do systémové paměti je obsluhován samotným procesorem bez využití jakéhokoli DMA (Direct Memory Access) nebo jiné možnosti pro dávkové přenosy dat. Výsledná rychlost rekonfigurace (vyčtení z paměti a následný zápis) je tedy hluboko pod udávanou maximální rychlostí konfiguračního rozhraní ICAP.

Čas potřebný pro rekonfiguraci výše popsaným způsobem je uveden v Tab. 6. Pro zajímavost jsou zahrnuty i časy potřebné pro konfiguraci přes rozhraní JTAG a pro případ, kdy jsou konfigurační data uložena v paměti Compact Flash, jejíž rychlost je omezena maximální propustností hardwarové komponenty SysACE. Jednotlivé časové údaje opět odpovídají konfiguraci oblastí o velikosti dvou rekonfigurovatelných rámců.

Tab. 6: Porovnání časů potřebných pro konfiguraci

Způsob konfigurace	Compact Flash	JTAG	Vyčítání + konfigurace		
			Vyčítání	Konfigurace	Vyčítání + konfigurace
Přenosová rychlost [MB/s]	30	66	24	36	-
Množství přenesených dat [B]	186 624	186 624	178 944	186 624	365 568*
Čas potřebný pro konfiguraci [ms]	6,220	2,820	7,456	5,148	12,185

\*množství přenesených dat je téměř dvojnásobné, před konfigurací je třeba data nejprve vyčíst (vyčtená data jsou kratší o nastavení CMD registrů)

Přenosové rychlosti u paměti Compact Flash a rozhraní JTAG jsou uvažovány maximální možné, které jsou uvedeny v oficiální dokumentaci výrobce. Rychlosti vyčítání a zápisu u prezentované techniky byly změřeny pomocí čítače umístěného v návrhu. Tyto rychlosti jsou přímo závislé na velikosti vyrovnávací paměti (paměť typu FIFO (First In First Out)) použité hardwarové komponenty HWICAP. Jedná se o standardní komponentu dostupnou v návrhových nástrojích Xilinx, která je určena pro obsluhu portu ICAP.

Pro dosažení lepší použitelnosti popsané kombinace technik je zapotřebí zvýšit maximální možnou rychlost vyčítání a zápisu dat do konfigurační paměti. A to ideálně až na hodnotu 400 MB/s, což je maximální rychlost ICAP portu udávaná výrobcem. Vyšší rychlosti rekonfigurace lze dosáhnout několika způsoby. Jednou z možností je

doplnění návrhu o komponentu DMA, se kterou lze dosáhnout přenosové rychlosti až 70 MB/s [16]. Další možností je vytvoření vlastního řadiče ICAP portu – tento přístup je popsán např. v [15], kde autoři popisují vlastní komponentu pro správu rekonfigurace s možností dávkových přenosů konfiguračních dat. Uváděná rychlost téměř dosahuje výše avizovaných 400 MB/s. Popsaný přístup je výhodný především v tom, že veškerá agenda spojená s přenosem dat je v režii této komponenty a během rekonfigurace nedochází k nadměrnému zatěžování procesoru. Jiný přístup ke zvýšení rychlosti zápisu a čtení konfiguračních dat je popsán např. v [12]. Zde je využita standardní hardwarová komponenta HWICAP doplněná o vlastní logiku se schopností zápisu datových slov o vyšší bitové šíři a celá komponenta je přetaktována na nejvyšší možný kmitočet. Uváděná maximální přenosová rychlost je 2200 MB/s při přetaktování na frekvenci 550 MHz, což je více než pětinašobek výrobcem doporučené taktovací frekvence 100 MHz.

### **5.3 Zápis dat do interních registrů**

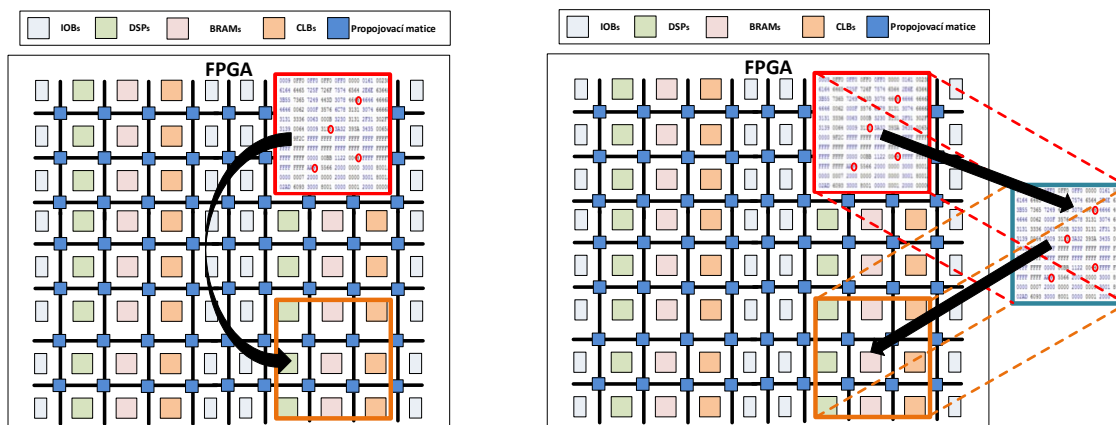
Nastavení příkazu pro zápis (restore) je v této práci realizováno zápisem posloupnosti příkazů do řídicích registrů FPGA. Vzhledem k nedostupnosti jakékoli oficiální dokumentace k technice zápisu dat do interních registrů byla sekvence potřebných příkazů sestavena experimentálně.

Využití techniky zápisu dat do interních registrů společně s technikami výše popsanými (tj. zkombinováním s technikou relokace a technikou zpětného vyčítání) umožňuje vytvoření rekonfigurovatelného systému podporujícího vzájemnou relokaci jednotlivých hardwarových modulů včetně jejich aktuálních vnitřních stavů. Při klasické DPR dochází k nastavení počátečních hodnot (reset) všech zainteresovaných komponent.

Před zahájením procesu přemístování jednotlivých komponent je nejprve nastaven příkaz pro vyčtení hodnot vnitřních registrů (v okamžiku, kdy je žádoucí uložit vnitřní stavy). Poté jsou získána všechna nezbytná data technikou zpětného vyčítání. Tato data jsou uložena do systémové paměti a následně jsou přidána k předuloženým řídicím příkazům, čímž dojde k vytvoření nového konfiguračního souboru. Dalším krokem je změna adresy a data jsou nahrána zpět (na jinou pozici) do systému. Posledním počinem před spuštěním nového modulu je zadání příkazu pro zápis dat do vnitřní logiky. Prvky obvodu, které nemají být při tomto procesu nijak ovlivněny, musí být chráněny pomocí maskovacích rámců. To znamená, že po vyčtení dat je třeba provést maskování zdrojové oblasti, aby nedošlo k jejímu ovlivnění při přepisování vnitřních stavů cílové oblasti. Blokové schéma výše popsaného testovacího systému je vyobrazeno na Obr. 15.







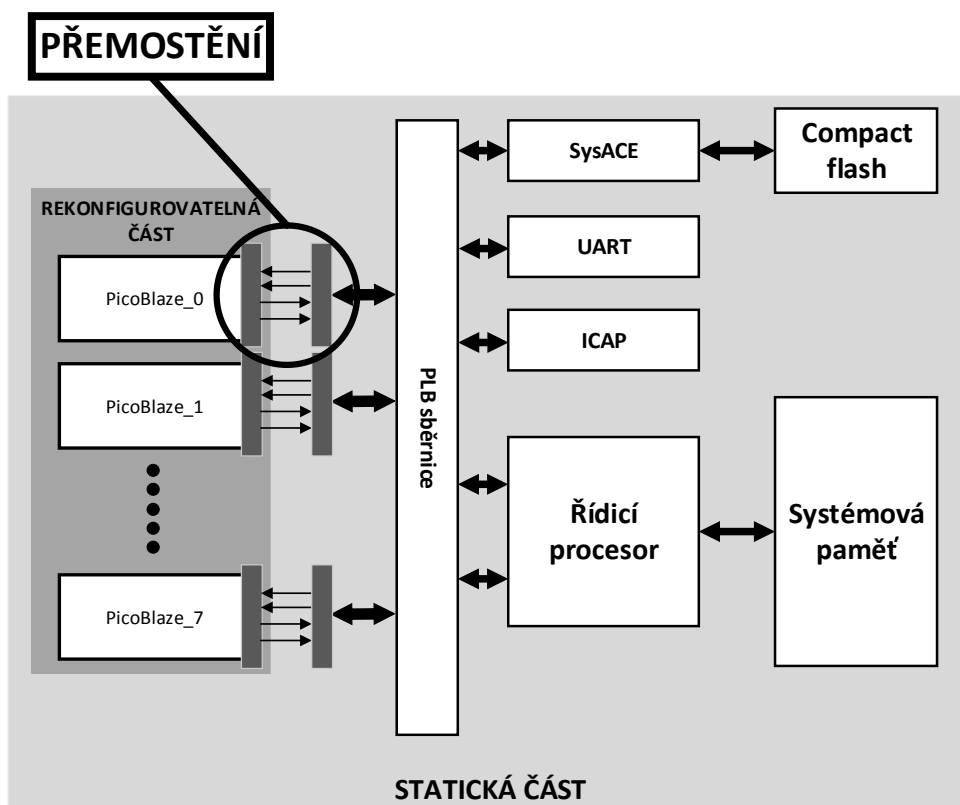
Obr. 16: Různé přístupy relokační hardwarových úloh

## 5.4 Test na komplexním systému

Všechny výše popsané experimenty jsou založeny na využití hardwarových modulů o velikosti dvou rekonfigurovatelných rámců, avšak funkční bloky umístěné uvnitř těchto modulů obsahují pouze jednoduché komponenty (násobička, sčítačka, čítač), které jsou tvořeny z malého počtu logických prvků. V některých případech lze dokonce říci, že navýšení potřebné logiky pro implementaci daného modulu způsobené přemostěním statické a dynamické části systému je vyšší než množství logiky použité pro implementaci samotných hardwarových modulů. Série těchto jednoduchých experimentů byla nezbytná pro ověření funkce dílčích částí práce.

Pro demonstraci použitelnosti popisované metodiky byla kombinace popsaných technik odzkoušena na komplexnějším systému. Základem tohoto systému je statický řídicí procesor MicroBlaze s potřebnými perifériemi a osmice rekonfigurovatelných procesorů PicoBlaze.

Popisovaný testovací systém obsahuje ve své statické části procesor MicroBlaze, dále je zde systémová paměť, komunikační rozhraní UART, konfigurační port ICAP a řadič paměti Compact Flash (externí úložiště částečných konfiguračních souborů). Dynamická část tohoto systému je tvořena osmi rekonfigurovatelnými oblastmi pro jednotlivé procesory PicoBlaze. K propojení všech komponent v systému je využita sběrnice PLB (Processor Local Bus). Blokové schéma tohoto systému je vyobrazeno na Obr. 17.



Obr. 17: Blokové schéma testovací aplikace s procesory PicoBlaze

Tato aplikace umožňuje relokaci jednotlivých jader procesoru PicoBlaze včetně přenosu aktuálního vnitřního stavu každého procesoru. Velikost oblastí zabíraná jednotlivými moduly odpovídá šesti rekonfigurovatelným rámcům. Z toho jsou čtyři rámce logiky (SLICES) a dva rámce pro nastavení paměti BRAM (jeden pro nastavení vlastností paměti a jeden pro její obsah). Velikost paměti potřebná pro trvalé uložení konfiguračních souborů, délka času potřebná pro implementaci návrhu a délka času potřebná pro nahrání některého z modulů jsou shrnuty v následujících tabulkách.

Tab. 7: Přehled množství paměti potřebné pro uložení částečných konfiguračních souborů u systémů s různým počtem rekonfigurovatelných procesorů PicoBlaze

Počet modulů		2	4	8
<b>DPR</b>	paměť[B]	202188	808752	3235008
<b>PBR</b>	paměť[B]	101094	202188	404376
<b>PBR&amp;Readback</b>	paměť[B]	960	960	960

Tab. 8: Přehled času potřebného pro implementaci systémů s různým počtem rekonfigurovatelných procesorů PicoBlaze

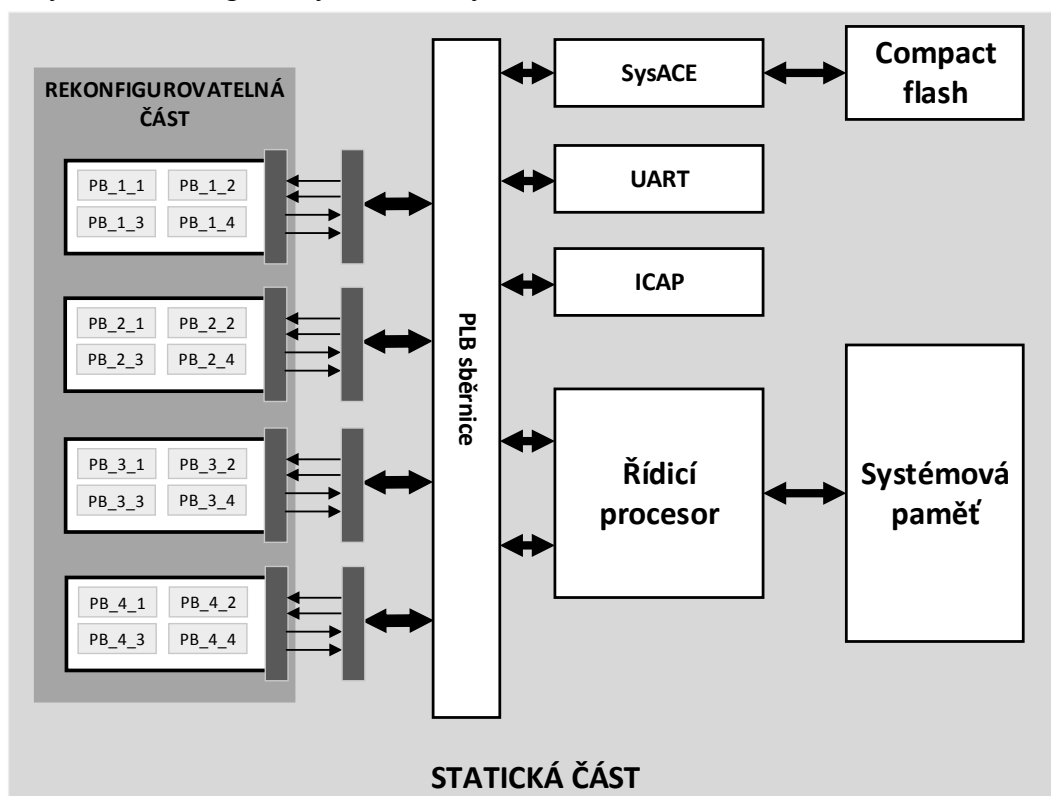
Počet modulů		2	4	8
<b>DPR</b>	čas [s]	3158	21064	82456
<b>PBR</b>	čas [s]	2699	3078	3630
<b>PBR&amp;Readback</b>	čas [s]	2730	3135	3726

Tab. 9: Přehled času potřebného pro rekonfiguraci jednoho modulu s procesorem PicoBlaze

Konfigurace přes ICAP port	Vyčítání + konfigurace		
	Vyčítání	Konfigurace	Celkem
Přenosová rychlost [MB/s]	24	36	-
Množství přenesených dat [B]	56587	57547	114134
Čas potřebný pro rekonfiguraci [ms]	18,862	12,788	31,650

Implementované procesory PicoBlaze využívají pouze jeden blok paměti BRAM, to znamená, že jejich umístění do jednotlivých rekonfigurovatelných oblastí vykazuje značnou neefektivitu využití logických zdrojů obvodu.

Tento nedostatek byl odstraněn mírnou úpravou popsaného systému, kdy namísto osmice rekonfigurovatelných jednotlivých procesorů PicoBlaze nová platforma obsahuje čtyři rekonfigurovatelné komponenty, přičemž každá tato komponenta je složena ze čtyř nezávislých jader procesoru PicoBlaze. Tento modifikovaný systém je blokově naznačen na Obr. 18. Implementační detaily tohoto testovacího systému jsou shrnuty v Tab. 10 a graficky znázorněny na Obr. 19, Obr. 20 a Obr. 21.

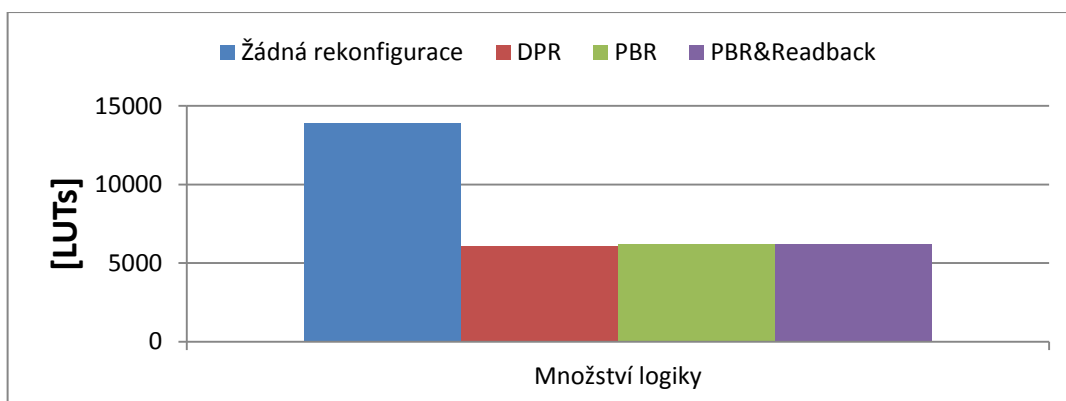


Obr. 18: Blokové schéma testovací modifikované aplikace s procesory PicoBlaze

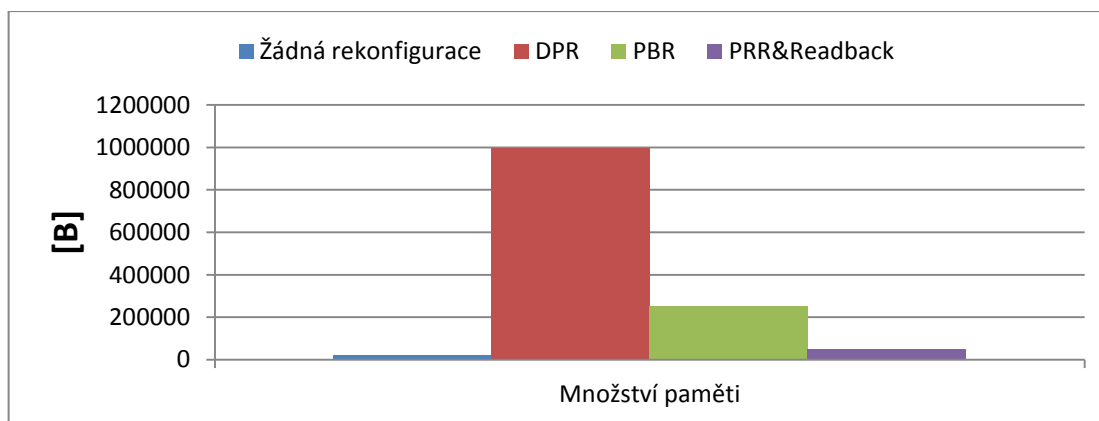
Tab. 10: Přehled základních parametrů testovacího systému

	Logika [LUT]	Paměť [B]	Implementační Čas [s]	Čas rekonfigurace [ms]
Žádná rekonfigurace	13 888	0	830	-
DPR	6 064	997 680	28 640	13,856
PBR	6 192	249 420	1 790	13,856*
PBR&Readback	6 192	960	1 910	34,321

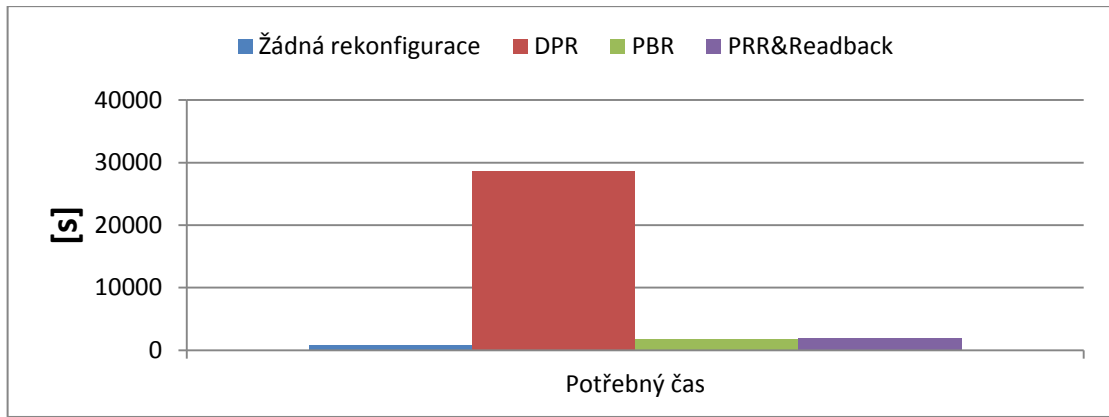
\* Do doby potřebné pro provedení rekonfigurace není započítána doba potřebná pro změnu adresy rámce



Obr. 19: Graficky znázorněné množství logiky nutné pro implementaci testovacího systému se čtyřnásobným procesorem PicoBlaze



Obr. 20: Graficky znázorněné množství paměti potřebné pro trvalé uložení konfiguračních souborů u testovacího systému se čtyřnásobným procesorem PicoBlaze



Obr. 21: Graficky znázorněné množství času potřebného pro běh implementačních nástrojů u testovacího systému se čtyřnásobným procesorem PicoBlaze

### 5.4.1 Zhodnocení experimentu

Popsané experimenty ukazují, že metodika návrhu prezentovaná v této práci je použitelná i na komplexní systémy, kde dochází k rekonfiguraci poměrně složitých hardwarových modulů. V tabulkách jsou porovnány tři typy rekonfigurovatelných systémů, přičemž první umožňuje pouze podporu standardní DPR, druhý podporuje relokaci funkčních modulů a třetí typ je systém s podporou relokace jednotlivých komponent včetně jejich aktuálních vnitřních stavů.

Obdobou tohoto experimentu je testovací aplikace blokově vyobrazená na Obr. 18, kde došlo ke zvýšení hustoty využití logických zdrojů v jednotlivých rekonfigurovatelných oblastech. Tento fakt vedl mimo jiné ke zvýšení počtu vstupních a výstupních pinů jednotlivých modulů, a tím i ke zvýšení složitosti při mapování a propojování návrhu. Provedený experiment dokládá, že popsaná metodika je použitelná i pro velké funkční moduly (osm rekonfigurovatelných rámců). Časování celého systému bez problému splňuje požadovaných 100 MHz (výsledky implementace uvádějí maximální taktovací frekvenci 130,856 MHz). V Tab. 10 jsou shrnuty základní parametry této platformy, mezi které patří množství potřebné logiky, množství potřebné paměti, čas potřebný pro implementaci a čas potřebný pro rekonfiguraci jednoho hardwarového modulu. V porovnání jsou zahrnuty čtyři typy systémů, prvním je systém bez podpory částečné rekonfigurace, další tři jsou systémy s podporou různých typů rekonfigurace (běžná DPR, PBR, PBR včetně vnitřních stavů hardwarových komponent). Tyto testovací systémy jsou navrženy tak, aby umožňovaly nastavení stejného množství funkcí jednotlivých hardwarových modulů. To znamená, že pokud má daná rekonfigurovatelná platforma čtyři hardwarové moduly, je v případě statického systému zapotřebí vytvořit šestnáct hardwarových modulů, aby bylo dosaženo stejné variability funkcí. U běžné DPR je třeba pro každý z těchto hardwarových modulů vytvořit a uchovávat čtyři konfigurační soubory.

## Závěr

Prezentovaná metodika návrhu je založena na použití několika dílčích technik, které společně umožňují vytvoření komplexního vysoce flexibilního systému na FPGA obvodu.

Se stále se zlepšující podporou částečné rekonfigurace v návrhových nástrojích se tato technika objevuje v celé řadě aplikací. Naneštěstí její použití vykazuje některé aspekty (složitost návrhu, dodatečné navýšení potřebného implementačního času, paměti a logických prvků v obvodu), díky kterým je v některých případech nevýhodná. Tato práce popisuje metodiku návrhu rekonfigurovatelných systémů na FPGA obvodech Xilinx. Prezentovaný postup umožňuje vytvoření optimalizovaného systému z hlediska výše zmíněných negativních parametrů.

Prvním dílčím počinem v rámci této práce bylo zpracování metody pro vytvoření návrhu s podporou relokace částečných konfiguračních souborů. Metodika využívá běžných implementačních nástrojů, tj. Xilinx Design Tools se zakoupenou licencí pro částečnou dynamickou rekonfiguraci. Principiálně je tato metoda založena na transformaci oddělovacích elementů na útvar, který lze definovat jako oddělovací makro.

Standardní oddělovací makro se používalo u starších FPGA obvodů (Virtex 4) a bylo podporováno v nižších verzích návrhových nástrojů (např. Xilinx ISE 9.2). Pro toto makro bylo nutné nejprve definovat správný typ a poté ho manuálně vložit do návrhu. U vytvořené metody je každé oddělovací makro tvořeno z oddělovacího elementu a z jednoho dodatečně přidaného LUT elementu. Volba směru a způsobu propojení těchto LUT elementů je zcela v rukou automatických propojovacích nástrojů. Tento postup vytváření rekonfigurovatelného systému sice navyšuje potřebu dodatečného hardwarového vybavení oproti přístupům využívajícím standardní oddělovací elementy. Avšak při vhodné volbě velikosti hardwarových modulů a množství pinů je toto navýšení potřebné logiky zastíněno ostatními vlastnostmi relokace částečných konfiguračních souborů. Navýšení složitosti návrhu, které je způsobeno nutností manuálně vytvářet jednotlivá oddělovací makra, je kompenzováno automatizací tohoto procesu s využitím sady vytvořených skriptů. Tyto skripty jsou společně s elektronickou verzí tohoto textu umístěny na příloženém CD.

Po rekonfiguraci je třeba nově umístěnou jednotku inicializovat, případně synchronizovat se zbytkem systému. Některé typy FPGA podporují techniku zpětného vyčítání konfigurační paměti (readback), díky které lze z obvodu získat veškerá konfigurační data i s obsahem vnitřních registrů v dané části obvodu. Dále pak bývá podporována technika zápisu dat do interních registrů obvodu (restore), která umožňuje inicializaci vnitřních stavů interních registrů v obvodu. Tyto techniky lze použít pro synchronizaci nových prvků v systému bez nutnosti jejich fyzického propojení. To může vést ke značné úspoře dodatečného hardwarového vybavení oproti systémům, které podporují synchronizaci s využitím datového propojení jednotlivých modulů.

Metodika návrhu prezentovaná v této práci inovativně umožňuje provádět relokaci hardwarových komponent na FPGA obvodech řady Virtex 5 a vyšší. Při práci s těmito

obvody je přemostění mezi statickou a dynamickou částí řešeno pomocí oddělovacích elementů, které za normálních okolností relokaci neumožňují. Přístup vytvořený v rámci této práce nevyžaduje provedení žádných významných zásahů do konfiguračních souborů nebo do strukturních netlistů, jako je tomu např. u nástroje GoAhead, kde dochází k jejich úpravě pomocí jazyka XDL.

Dalším inovativním prvkem je využití částečné rekonfigurace spolu s pokročilými technikami, mezi něž patří relokace hardwarových modulů, zpětné vyčítání konfigurační paměti a zápis dat do interních registrů obvodu. Tato kombinace umožňuje vytvořit systém podporující dynamické přemísťování hardwarových komponent včetně jejich aktuálních stavů. Dále je možné provádět synchronizaci funkčních komponent po jejich rekonfiguraci. Případně je možné měnit obsah libovolného paměťového prvku v obvodu.

Způsob tvorby rekonfigurovatelných systémů prezentovaný v předcházejícím textu celistvě popisuje možnosti využití pokročilých vlastností moderních FPGA obvodů s možností tvorby vysoce flexibilního návrhu splňujícího ty nejvyšší nároky.

# Seznam literatury

## *Vlastní publikace*

### **IEEE publikace:**

- [1] Drahoňovský, T.; Rozkovec, M.; Novák, O.: "A highly flexible reconfigurable system on a Xilinx FPGA," International Conference on ReConfigurable Computing and FPGAs (ReConFig 2014), 8-10 December 2014, ISBN 978-1-4799-5943-3
- [2] Drahoňovský, T.; Rozkovec, M.; Novák, O.: "Relocation of reconfigurable modules on Xilinx FPGA," Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2013 IEEE 16th International Symposium., pp.175 - 180, 8-10 April 2013, IEEE Catalog Number: CFP13DDE-PRT, ISBN 978-1-4673-6133-0
- [3] Cvek, P.; Drahoňovský, T.; Rozkovec, M.: "GNU/Linux and Reconfigurable Multiprocessor FPGA Platform," Electronics, Control, Measurement, Signals and their application in Mechatronics (ECMSM2013), Toulouse, France, June 2013, IEEE Catalog Number: CFP13ECN-USB, ISBN: 978-1-14673-6297-9

### **Ostatní publikace:**

- [4] Drahoňovský, T.; Rozkovec, M.: "Guitar effects implementation in the FPGA circuits," Electronics, Control, Measurement and Signals (ECMS), June 2011, Liberec, Czech republic, Informal proceedings of the ECMS 2011, pp. 80-86, ISBN 978-80-7372-781-9
- [5] Drahoňovský, T.: "Implementace kytarových efektů v obvodu FPGA," 2011, Sdělovací technika, No. 2, February 2011, pp. 4-7, ISSN 0036-9942
- [6] Drahoňovský, T.: "Návrh rekonfigurovatelného víceprocesorového systému na FPGA obvodu," 2012, Počítačové architektury & diagnostika (PAD), September 2012, Milovy, Czech republic, pp. 85-90, ISBN 978-80-01-05106-1
- [7] Drahoňovský, T.: "Využití částečné dynamické rekonfigurace pro testování a zvyšování spolehlivosti FPGA obvodů," 2011, Počítačové architektury & diagnostika (PAD), September 2011, Stará Lesná, Slovakia, pp. 55-60, ISBN 978-80-227-3552-0
- [8] Drahoňovský, T.: "Hardware task relocation on a Xilinx FPGA," 2014, Annual DCPS Evaluation Workshop, November 2014, Cottbus, Germany, pp. 35-38
- [9] Drahoňovský, T.: "Reconfigurable system on Xilinx FPGA with low memory requirements for partial bitstreams storing," 2014, 3<sup>rd</sup> Biannual European – Latin American Summer School on Design, Test and Reliability, April 2014, Frankfurt (Oder), Germany, pp. 115-119
- [10] Drahoňovský, T.: "Víceprocesorový rekonfigurovatelný systém na obvodu FPGA," Počítačové architektury & diagnostika (PAD), September 2013, Klášter Teplá, Czech republic, pp. 75-80, ISBN 978-80-261-0270-0



## **Použitá literatura**

- [11] Bauer, L.; Braun, C.; Imhof, M.E.; Kochte, M.A; Hongyan Zhang; Wunderlich, H.; Henkel, J.: "OTERA: Online test strategies for reliable reconfigurable architectures," *Adaptive Hardware and Systems (AHS), 2012 NASA/ESA*, pp.38,45, 25-28 June 2012
- [12] Hansen, S.G.; Koch, D.; Torresen, J.: "High Speed Partial Run-Time Reconfiguration Using Enhanced ICAP Hard Macro," *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on* , vol., no., pp.174,180, 16-20 May 2011
- [13] Miculka, L.; Straka, M.; Kotasek, Z.: "Methodology for Fault Tolerant System Design Based on FPGA into Limited Redundant Area," *Digital System Design (DSD), 2013 Euromicro Conference on* , vol., no., pp.227,234, 4-6 Sept. 2013
- [14] Straka, M.; Kastil, J.; Kotasek, Z.: "Fault Tolerant Structure for SRAM-Based FPGA via Partial Dynamic Reconfiguration," *Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on* , vol., no., pp.365,372, 1-3 Sept. 2010
- [15] Vipin, K.; Fahmy, S.A, "A high speed open source controller for FPGA Partial Reconfiguration," *Field-Programmable Technology (FPT), 2012 International Conference on* , vol., no., pp.61,66, 10-12 Dec. 2012
- [16] Vipin, K.; Fahmy, S.A, "ZyCAP: Efficient Partial Reconfiguration Management on the Xilinx Zynq," *Embedded Systems Letters, IEEE* , vol.6, no.3, pp.41,44, Sept. 2014
- [17] Virtex FPGA Series Configuration and Readback (xapp 138), [online], [2005-3-11]
- [18] Xilinx Inc.: 7 Series FPGAs Overview (ds180), [online], [2012-5-5]
- [19] Xilinx Inc.: Partial Reconfiguration User Guide (ug702), [online], [2010-10-5]
- [20] Xilinx Virtex-6 FPGA Configuration (ug 360), [online], [2012-9-11]