

# DESIGN AND REALIZATION OF ARITHMETIC OPERATIONS FOR FPGA CIRCUITS

**Luděk Soukup**

Bachelor Degree Programme (3), FEEC BUT  
E-mail: xsouku00@stud.feec.vutbr.cz

Supervised by: Lukáš Fojcik

E-mail: fujcik@feec.vutbr.cz

## ABSTRACT

This project is focused on design and realization of arithmetic operations in digital circuits. The attention is paid to algorithms for addition, subtraction, multiplication and division. Chosen algorithms are described in language VHDL, synthesized, and they are compared in terms of acquired information about computing time and circuit cost.

## 1. ÚVOD

Významným trendem v oblasti výpočetní a řídicí techniky je tvorba stále výkonnějších a obecně dokonalejších systémů. Důležitým prvkem jsou přitom moderní typy obvodů jako FPGA, signálové procesory nebo obvody ASIC. Výpočetní výkon, flexibilita a přijatelná cena představují hlavní pozitiva, která jsou však vykoupena vyšší náročností vývoje.

Příkladem zvýšené náročnosti vývoje může být implementace aritmetických funkcí, které je tento projekt věnován. Zatímco u obvodů FPGA je dána použitým vývojovým nástrojem (dokonalejší systémy ji umožňují přímo), u obvodů ASIC není možná vůbec. Vyžaduje-li tedy aplikace použití matematických operací, je třeba projekt rozšířit o algoritmy realizující příslušné matematické funkce.

V současné době existuje celá řada algoritmů pro takzvanou „počítačovou“ aritmetiku. Liší se jednak náročností realizace, jednak dosažitelným výpočetním výkonem. Při implementaci je nutné brát na zřetel nejen optimalizaci obvodovou, ale také posoudit vhodnost daného algoritmu.

## 2. ROZBOR

Realizace projektu je rozdělena do pěti navazujících částí:

- Definice požadavků – přesné vymezení cílů a rozsahu projektu
- Volba algoritmů – výběr algoritmů s vhodnými vlastnostmi
- VHDL popis – tvorba kódů a jejich verifikace v jazyce VHDL
- Syntéza – získání dat o výsledných strukturách
- Vyhodnocení jednotlivých algoritmů – diskuze využitelnosti algoritmů

### 3. REALIZACE

#### 3.1. DEFINICE POŽADAVKŮ

Hlavním cílem tohoto projektu je vytvořit VHDL popis vhodných algoritmů pro realizaci základních aritmetických operací. Výsledný implementační balík musí obsahovat parametricky realizované struktury (šířka operandů je stanovena při syntéze podle hodnoty příslušného parametru) pro sčítání, odčítání, násobení a dělení v pevné řádové čárce. Algoritmy by měly představovat vhodná řešení jak pro aplikace zaměřené na maximální výpočetní výkon, tak pro aplikace vyžadující nízkou cenu implementované struktury.

#### 3.2. VOLBA ALGORITMŮ

Volba algoritmů proběhla na základě požadavků stanovených v předchozí podkapitole. Jako podklady pro výběr byly použity prameny [1] a [2].

Seznam zvolených algoritmů rozdělených podle funkce:

- Sčítání – Ripple-Carry, Carry-Select, Carry-Lookahead
- Násobení – Ripple-Carry, Booth Radix 4, Carry-Save
- Dělení – Restore, Nonrestore, SRT Radix 4

#### 3.3. VHDL POPIS A SYNTÉZA

Pro tvorbu zdrojových kódů a jejich syntézu bylo využito vývojové prostředí ISE<sup>®</sup> verze 10.1 od firmy Xilinx<sup>®</sup>, která je také výrobcem cílového obvodu Spartan3 XC3S200.

#### 3.4. VÝSLEDKY SYNTÉZY A VYHODNOCENÍ ALGORITMŮ

Jak již bylo v úvodu naznačeno, některá vývojová prostředí jsou schopna automaticky generovat struktury nutné pro implementaci základních aritmetických operací. Jedním z těchto prostředí je také ISE<sup>®</sup>. Struktury generované tímto prostředím byly do projektu také zahrnuty, v dalším textu budou označovány jako AGS.

Algoritmus		Zpoždění [ns]	Cena [počet elemen- tárních buněk]
Sčítání	AGS	9,758	8
	Ripple-Carry	28,741	18
	Carry-Select	21,325	29
	Carry-Lookahead	18,218	32
Násobení	AGS	11,207	0
	Ripple-Carry	71,513	293
	Booth radix 4	37,796	333
	Carry-Save	33,314	276
Dělení	Res	92,570	283
	NonRes	75,487	161
	SRT radix 4	30,456	162

**Tabulka 1:** Výpočetní zpoždění a cena realizace jednotlivých algoritmů.

Během procesu syntézy byl získán velmi rozsáhlý soubor dat, kdy pro každý algoritmus bylo uvažováno několik šířek operandů, dále bylo použito různých optimalizačních nastavení syntetizátoru. V tabulce 1 jsou uvedeny parametry výsledných struktur pro šířku operandu 16 b a optimalizaci syntetizátoru na minimální výpočetní zpoždění. Tyto hodnoty představují reprezentativní vzorek získaných hodnot.

Sčítání je nejčastěji využívanou aritmetickou operací. Realizované algoritmy představují vhodná řešení pro aplikace vyžadující nízkou cenu (Ripple-Carry), vysoký výpočetní výkon (Carry-Lookahead), případně jakýsi kompromis mezi těmito požadavky (Carry-Select). Jejich parametry jsou však zastíněny strukturou AGS, která dosahuje několikanásobně lepší výpočetní rychlosti při zachování nízké ceny. Příčinou je nedokonalé využití elementárních buněk – slices. Ačkoli tyto buňky obsahují pomocnou aritmetickou logiku, není syntetizátorem využita. K jejímu využití dochází pouze u struktur AGS. Další snížení výpočetního zpoždění poté přináší možnost využít rychlá lokální propojovací pole.

Odčítání bylo do této chvíle poněkud opomíjeno. Důvodem byl prostý fakt, že pro odčítání bývají jen ve výjimečných případech používány samostatné struktury. V naprosté většině případů bývá odčítání realizováno jako přičítání záporného čísla. Dochází tedy ke specifickému využití sčítaček.

Také v případě násobení vyznívají výsledky ve prospěch přímé implementace. Zde je ovšem situace poněkud odlišná. Pro násobení a dělení je totiž využito specializovaných struktur. Jejich počet a přesné parametry jsou závislé na konkrétním obvodu. Tato skutečnost je dokumentována hodnotou 0 ve sloupci cena, neboť tyto specializované struktury nevyužívají elementární buňky.

Přímá realizace dělení (AGS) není bohužel možná. Algoritmy Res – s návratem k nezápornému zbytku a NonRes – bez návratu k nezápornému zbytku byly zařazeny spíše z hlediska ucelenosti projektu. V současné době je nejčastěji využíván algoritmus SRT dosahující vynikajících vlastností.

#### 4. ZÁVĚR

V rámci tohoto projektu byly vybrány vhodné algoritmy pro realizaci základních aritmetických operací v pevné řádové čarce. Algoritmy byly popsány v jazyce VHDL a provedením syntézy byla získána data o rychlosti a ceně výsledných struktur.

Realizované algoritmy představují pro FPGA kvalitativně nižší řešení než struktury vytvořené vývojovým prostředím. Uplatnění však nacházejí při vývoji zákaznických obvodů ASIC. Zdrojové kódy jsou díky použitému jazyku VHDL snadno přenositelné.

#### LITERATURA

- [1] DESCHAMPS, Jean-Pierre, BIOUL, Géry Jean Antoine, SUTTER, Gustavo D. Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems. New Jersey: Wiley, 2006. 556 s. ISBN 978-0-471-68783-2.
- [2] LU, Mi. Arithmetic and logic in computer systems. New Jersey: Willey, 2004. 246 s. ISBN 0-471-46945-9.
- [3] PINKER, Jiří, POUPA, Martin. Číslicové systémy a jazyk VHDL. Praha: BEN, 2006. 352 s. ISBN 80-7300-198-5.