



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**HARDVÉROVÁ REALIZÁCIA NUMERICKÉHO  
INTEGRÁTORA S METÓDOU VYŠŠIEHO RÁDU**

HARDWARE REALIZATION OF HIGHER ORDER NUMERICAL INTEGRATOR

**SEMESTRÁLNÍ PROJEKT**

TERM PROJECT

**AUTOR PRÁCE**

AUTHOR

**FRANTIŠEK MATEČNÝ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. VÁCLAV ŠÁTEK, Ph.D.**

**BRNO 2018**

## Abstrakt

Práca popisuje numerickú integráciu a riešenie diferenciálnych rovníc pomocou metódy Taylorovej rady v rôznych typoch integrátorov. Ďalej je popísaná aritmetika pevnej a pohyblivej rádovej čiarky. Následne sú predstavené návrhy a spôsob výpočtu paralelných integrátorov s operáciou násobenia a delania v prevedení pevnej a pohyblivej rádovej čiarky.

## Abstract

This work deals with numerical integration and solution of differential equations by the Taylor series in many types of integrators. Next is described floating point and fixed point arithmetic. Subsequently are presented designs and method of calculation of parallel multiplication and division integrators in floating point and fixed point arithmetic. TODO - opraviť!!

## Kľúčové slová

diferenciálna rovnica, numerická integrácia, Taylorova rada, pevná rádová čiarka, pohyblivá rádová čiarka, integrátor

## Keywords

differential equation, numeric integration, Taylor series, fixed point, floating point, integrator

## Citácia

MATEČNÝ, František. *Hardvérová realizácia numerického integrátora s metódou vyššieho rádu*. Brno, 2018. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šátek, Ph.D.

# Hardvérová realizácia numerického integrátora s metódou vyššieho rádu

## Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Václava Šátka, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
František Matečný  
16. mája 2018

## Podakovanie

Chcel by som sa poďakovať svojmu školiteľovi Ing. Václavovi Šátkovi, Ph.D., za odborné vedenie práce, podporu, vecné pripomienky, komentáre a rady k mojej práci. Moje poďakovanie taktiež patrí mojej rodine, priateľke Katke, kamarátom a všetkým ostatným, ktorí ma podporovali počas celej doby môjho štúdia a bez ktorých by táto práca nemohla vzniknúť.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Numerická integrácia</b>	<b>4</b>
2.1	Taylorova rada . . . . .	4
2.2	Eulerova metóda . . . . .	5
2.3	Runge-Kutta . . . . .	5
2.3.1	Runge-Kutta 2. rádu . . . . .	5
2.3.2	Runge-Kutta 4. rádu . . . . .	6
<b>3</b>	<b>Riešenie diferenciálnych rovníc Taylorovou radou</b>	<b>7</b>
3.1	Riešenie diferenciálnej rovnice s operáciou násobenia . . . . .	8
3.2	Riešenie diferenciálnej rovnice s operáciou delenia . . . . .	9
<b>4</b>	<b>Reprezentácia operandov</b>	<b>11</b>
4.1	Pevná rádová čiarka . . . . .	11
4.2	Pohyblivá rádová čiarka . . . . .	12
4.2.1	Súčet a rozdiel . . . . .	14
4.2.2	Násobenie a delenie . . . . .	15
<b>5</b>	<b>Numerické integrátory</b>	<b>16</b>
5.1	Návrh násobiaceho integrátora v pevnej rádovej čiarke . . . . .	17
5.2	Návrh deliaceho integrátora v pevnej rádovej čiarke . . . . .	19
5.3	Návrh jendovstupého integrátora . . . . .	20
5.4	Návrh komponentu pre spracovanie exponentov . . . . .	21
5.5	Návrh násobiaci integrátora v pohyblivej rádovej čiarke . . . . .	23
5.6	Deliaci integrátor v pohyblivej rádovej čiarke . . . . .	24
5.7	Sústava diferenciálnych rovníc . . . . .	25
<b>6</b>	<b>Implementácia integrátorov vo VHDL</b>	<b>27</b>
6.1	Integrátory v pevnej rádovej čiarke . . . . .	27
6.2	Integrátory v pohyblivej rádovej čiarke . . . . .	32
6.3	Sústava integrátorov . . . . .	32
<b>7</b>	<b>Analýza</b>	<b>34</b>
7.1	Porovnanie Taylorovej rady s metódou Runge-Kutta 2. radu . . . . .	34
7.2	Porovnanie Taylorovej rady s metódou Runge-Kutta 4. radu . . . . .	34
<b>8</b>	<b>Záver</b>	<b>35</b>



# Kapitola 1

## Úvod

Hlavným cieľom tejto práce je návrh hardvérových komponentov na riešenie rozsiahlych diferenciálnych rovníc. Diferenciálne rovnice sa väčšinou riešia pomocou numerickej integrácie, a teda s použitím vhodných numerických metód. Hardvérový komponent využívajúci numerickú integráciu sa nazýva numerický integrátor.

V kapitole 2 sú predstavené rôzne numerické metódy - Eulerova metóda, metóda Runge-Kutta a Taylorov rad. Najväčšia pozornosť je venovaná metóde Taylorovej rady, ktorá poskytuje vhodný pomer medzi rýchlosťou a presnosťou [2]. Bližší popis a práca s touto metódou sú uvedené v kapitole 3. Ukážeme si rozdelenie Taylorovej rady na jednotlivé členy a následne úpravu týchto členov tak, aby bolo možné výpočet čo najviac paralelizovať a optimalizovať. Táto úprava bude realizovaná na obyčajných diferenciálnych rovniciach s operáciou násobenia a delenia. Takto upravené a vytvorené rovnice budú následne použité pri návrhu rôznych typov numerických integrátorov.

Kapitola 4 sa zaoberá reprezentáciou operandov v pevnej a v pohyblivej rádovej čiarky. Pri použití pohyblivej rádovej čiarky sú uvedené postupy výpočtu znamienka, exponentu a mantisy na jednoduchých matematických operáciách ako sú sčítanie, odčítanie, násobenie a delenie.

V kapitole 5 sú predstavené a popísané návrhy jednotlivých integrátorov a popis ich činnosti. Podľa rovníc uvedených v kapitole 3, sú navrhnuté paralelné numerické integrátory s operáciou násobenia a delenia. Oba integrátory sú navrhnuté v prevedení pevnej a pohyblivej rádovej čiarky. Operácia delenia je realizovaná pomocou deliacho algoritmu SRT. Bližší popis tohto algoritmu sa nachádza v bakalárskej práci Simulátor procesora s operáciou delenia [3]. Vzájomným zapojením navrhnutých numerických integrátorov je možné riešiť rozsiahle diferenciálne rovnice.

## Kapitola 2

# Numerická integrácia

Diferenciálne rovnice sú matematické rovnice, v ktorých ako premenné vystupujú derivácie funkcií. Najvyššia derivácia v rovnici udáva rád rovnice. Rovnice, ktoré obsahujú derivácie len podľa jednej premennej, sa nazývajú obyčajné diferenciálne rovnice (ODR). Rovnice, ktoré obsahujú derivácie podľa viacerých premenných, sú takzvané parciálne diferenciálne rovnice (PDR). Diferenciálnu rovnicu je možné riešiť analyticky alebo použitím numerickej metódy. Pri väčšine praktických úloh je analytické riešenie veľmi zložité, preto sa používa skôr riešenie numerické. Základným princípom numerického riešenia je diskretizácia premenných, keď spojitú veličinu nahradíme postupnosťou diskrétnych bodov. Pri použití dostatočne hustom rozložení bodov môžeme približne reprezentovať spojitú veličinu. Vzdialenosť medzi dvoma susednými bodmi sa nazýva krok metódy. Numerické metódy pri svojom výpočte používajú niekoľko predchádzajúcich krokov. Podľa počtu týchto krokov rozdeľujeme numerické metódy na metódy jednokrokové a viackrokové. Jednokrokové metódy pri svojom výpočte používajú len jeden predchádzajúci krok, viackrokové využívajú niekoľko predchádzajúcich krokov. Pri numerických metódach je teda potrebné zvoliť počiatočný stav t.j. počiatočnú podmienku riešenej úlohy. Od nej sa následne počíta hodnota funkcie v ďalšom bode [4].

Najhlavnejšími kritériami pri numerických metódach je ich presnosť a rýchlosť. Tie je možné ovplyvniť veľkosťou integračného kroku a rádom integračnej metódy. Pri počítaní numerickými metódami nedostávame teoreticky presné riešenie, ale výsledok konverguje k správne riešeniu, a teda dostávame výsledok s určitou presnosťou. Výsledná chyba výpočtu je súčet lokálnej a akumulovanej chyby. Lokálna chyba zahŕňa chybu numerickej metódy a zaokrúhľovaciu chybu, ktorá môže byť spôsobená typom hardvérovej architektúry, ako napríklad použitím pevnej alebo pohyblivej rádovej čiarky, ktoré sú bližšie popísané v kapitole 4. Akumulovaná chyba je súčtom lokálnych chýb, a teda sa počas výpočtu zvyšuje. Chyba v jednom kroku tak ovplyvňuje výsledky krokov nasledujúcich.

### 2.1 Taylorova rada

Táto numerická metóda je tvorená nekonečným radom, avšak na výpočet sa používa len niekoľko jej členov. Počet použitých členov udáva rád metódy. Čím väčší počet členov použijeme, tým je výsledok presnejší. Počet použitých členov môže byť zadaný fixne, alebo sa môže dynamicky meniť v závislosti od požadovanej presnosti. Presnosť sa počíta z viacerých najvyšších členov, a po dosiahnutí požadovanej presnosti výpočet končí. Nekonečnú Taylorovu radu môžeme zapísať:

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y'''_i + \frac{h^4}{4!}y^{(4)}_i + \dots + \frac{h^n}{n!}y^{(n)}_i, \quad (2.1)$$

kde  $h$  je veľkosť integračného kroku a  $i$  označuje krok diskretizovanej veličiny. Ďalšie popísané metódy sú odvodené od Taylorovej rady.

## 2.2 Eulerova metóda

Najjednoduchšou jednokrokovou metódou pre riešenie obyčajných diferenciálnych rovníc je Eulerova metóda. Je to Taylorova metóda 1. rádu, keďže používa len prvé dva členy Taylorovej rady. Preto je rýchla, ale menej presná. Zapisuje sa nasledovne:

$$y_{i+1} = y_i + hy'_i \quad (2.2)$$

Zvolením dostatočne malého integračného kroku  $h$  môžeme zvýšiť jej presnosť. Čím je však krok menší, tým je výpočet pomalší.

## 2.3 Runge-Kutta

Ďalšou veľmi známou numerickou metódou je Runge-Kutta. Všeobecná schéma tejto metódy má tvar:

$$\begin{aligned} y_{i+1} &= y_i + \sum_{j=1}^r \alpha_j k_j, \quad i = 0, 1, \dots \\ k_1 &= f(x_i, y_i) \\ k_j &= f(x_i + \lambda_j h, y_i + \mu_j h k_{j-1}), \quad j = 2, \dots, r \end{aligned} \quad (2.3)$$

kde  $\alpha_j$ ,  $\lambda_j$  a  $\mu_j$  sú vhodne zvolené konštanty a  $r$  určuje rád metódy. Ako je z uvedených rovníc vidieť, pri výpočte sa používajú medzivýpočty  $k$ , ktorých počet je rovný rádu metódy. Najznámejšie a najčastejšie používané varianty sú Runge-Kutta 2. a 4. rádu, ktoré sú popísané nižšie.

### 2.3.1 Runge-Kutta 2. rádu

Táto metóda je oproti Eulerovej metóde presnejšia, ale pri rovnakej veľkosti integračného kroku vyžaduje viac operácií. Na výpočet používa dva medzivýpočty  $k_1$  a  $k_2$ . Má nasledujúci tvar:

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}h(k_1 + k_2) \\ k_1 &= f(t_i, y_i) \\ k_2 &= f(t_{i+1}, y_i + hk_1) \end{aligned} \quad (2.4)$$



### 2.3.2 Runge-Kutta 4. rádu

Runge-Kutta 4 rádu je najpoužívanější tvar tejto metódy, ktorý môžeme zapísať nasledovne:

$$\begin{aligned}y_{i+1} &= y_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \\k_1 &= f(t_i, y_i) \\k_2 &= f(t_{i+1} + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \\k_3 &= f(t_{i+1} + \frac{1}{2}h, y_i + \frac{1}{2}hk_2) \\k_4 &= f(t_{i+1} + h, y_i + hk_3)\end{aligned}\tag{2.5}$$

Pri výpočte sú použité štyri medzivýpočty, avšak aj napriek tomu má táto metóda dobrý pomer rýchlosti a presnosti.

## Kapitola 3

# Riešenie diferenciálnych rovníc Taylorovou radou

Na riešenie diferenciálnych rovníc je možné upraviť základný tvar Taylorovej rady (2.1) tak, aby sa dali jednotlivé operácie vykonávať paralelne. Prevod jednoduchej obyčajnej diferenciálnej rovnice je prevzatý z [3]. Ďalšie možné zdroje sú [6], [1].

Obyčajná diferenciálna rovnica:

$$y' = y, \quad y(0) = y_0. \quad (3.1)$$

Z tohto vzťahu vyplýva, že:

$$y = y' = y'' = y''' = y^{(4)} = \dots = y^{(n)}. \quad (3.2)$$

Po dosadení do Taylorovej rady (2.1) získame:

$$y_{i+1} = y_i + hy_i + \frac{h^2}{2!}y_i + \frac{h^3}{3!}y_i + \frac{h^4}{4!}y_i + \dots + \frac{h^n}{n!}y_i \quad (3.3)$$

To je možné prepísať na:

$$y_{i+1} = y_i + DY1_i + DY2_i + DY3_i + DY4_i + \dots + DY(N)_i \quad (3.4)$$

kde je význam jednotlivých členov nasledujúci:

$$\begin{aligned} DY1_i &= hy_i \\ DY2_i &= \frac{h^2}{2!}y_i = \frac{h}{2}DY1_i \\ DY3_i &= \frac{h^3}{3!}y_i = \frac{h}{3}DY2_i \\ DY4_i &= \frac{h^4}{4!}y_i = \frac{h}{4}DY3_i \\ &\vdots \end{aligned} \quad (3.5)$$

Všeobecný zápis:

$$DY(N)_i = \frac{h^n}{n!}y_i = \frac{h}{n}DY(N-1)_i$$

Z týchto vzťahov je možné riešiť jednoduché diferenciálne rovnice, ako je tomu [5], [1]. Zo vzťahov taktiež vyplýva, že každý ďalší člen Taylorovej rady je počítaný z predchádzajúceho, čo vedie k zefektívneniu výpočtu, a to hlavne pri vyšších deriváciách, kde je výpočet častokrát zložitý. Podobným postupom je možné upraviť diferenciálne rovnice, ktoré obsahujú operáciu násobenia alebo delenia, a tiež tak zvýšiť efektivitu výpočtu týchto výpočtovo náročnejších operácií.

### 3.1 Riešenie diferenciálnej rovnice s operáciou násobenia

Všeobecný zápis pre diferenciálnu rovnicu s operáciou násobenia je nasledujúci:

$$y' = qr, \quad y(0) = y_0 \quad (3.6)$$

Pre výpočet každej premennej v rovnici je potrebné použiť Taylorovu radu v tvare (3.4). Jednotlivé Taylorove rady vyzerajú nasledovne:

$$y_{i+1} = y_i + DY1_i + DY2_i + DY3_i + DY4_i + \dots + DY(n)_i \quad (3.7)$$

$$q_{i+1} = q_i + DQ1_i + DQ2_i + DQ3_i + DQ4_i + \dots + DQ(n)_i \quad (3.8)$$

$$r_{i+1} = r_i + DR1_i + DR2_i + DR3_i + DR4_i + \dots + DR(n)_i \quad (3.9)$$

Spočítame jednotlivé derivácie rovnice (3.6):

$$\begin{aligned} y' &= qr \\ y'' &= q'r + qr' \\ y''' &= q''r + 2q'r' + qr'' \\ y^{(4)} &= q'''r + 3q''r' + 3q'r'' + qr''' \\ &\vdots \end{aligned}$$

Ako je vidieť, derivácie vytvárajú Pascalov trojuholník a môžeme ich všeobecne zapísať ako:

$$y^{(n+1)} = \sum_{k=0}^n \binom{n}{k} q^{(n-k)} r^{(k)}$$

Z derivácií odvodíme jednotlivé členy Taylorovej rady (2.1), ktoré majú nasledujúci význam:

$$\begin{aligned} \frac{DY1_i}{h} &= DQ0_i DR0_i \\ \frac{DY2_i}{\frac{h^2}{2!}} &= \frac{DQ1_i}{h} DR0_i + DQ0_i \frac{DR1_i}{h} \\ \frac{DY3_i}{\frac{h^3}{3!}} &= \frac{DQ2_i}{\frac{h^2}{2!}} DR0_i + 2 \frac{DQ1_i}{h} \frac{DR1_i}{h} + DQ0_i \frac{DR2_i}{\frac{h^2}{2!}} \\ \frac{DY4_i}{\frac{h^4}{4!}} &= \frac{DQ3_i}{\frac{h^3}{3!}} DR0_i + 3 \frac{DQ2_i}{\frac{h^2}{2!}} \frac{DR1_i}{h} + 3 \frac{DQ1_i}{h} \frac{DR2_i}{\frac{h^2}{2!}} + DQ0_i \frac{DR3_i}{\frac{h^3}{3!}} \\ &\vdots \end{aligned} \quad (3.10)$$

Po úprave dostaneme konečný tvar jednotlivých členov:

$$DY1_i = hq_i r_i \quad (3.11)$$

$$DY2_i = \frac{h}{2}(DQ1_i DR0_i + DQ0_i DR1_i) \quad (3.12)$$

$$DY3_i = \frac{h}{3}(DQ2_i DR0_i + DQ1_i DR1_i + DQ0_i DR2_i) \quad (3.13)$$

$$DY4_i = \frac{h}{4}(DQ3_i DR0_i + DQ2_i DR1_i + DQ1_i DR2_i + DQ0_i DR3_i) \quad (3.14)$$

$\vdots$

Všeobecný zápis:

$$DY(N)_i = \frac{h}{N} \cdot \left( \sum_{k=1}^N DQ(N-k)_i \cdot DR(k-1)_i \right) \quad (3.15)$$

Uvedený prevod vychádza z práce [12]. Jednotlivé rovnice členov Taylorovej rady budú slúžiť pri návrhu a implementácii násobiacich integrátorov.

## 3.2 Riešenie diferenciálnej rovnice s operáciou delenia

Podobne, ako v predchádzajúcom prípade, zapíšme diferenciálnu rovnicu s operáciou delenia nasledovne:

$$y' = \frac{u}{v}, \quad y(0) = y_0 \quad (3.16)$$

Jednotlivé Taylorove rady pre premenné v rovnici vyzerajú nasledovne:

$$y_{i+1} = y_i + DY1_i + DY2_i + DY3_i + DY4_i + \dots + DY(n)_i \quad (3.17)$$

$$u_{i+1} = u_i + DU1_i + DU2_i + DU3_i + DU4_i + \dots + DU(n)_i \quad (3.18)$$

$$v_{i+1} = v_i + DV1_i + DV2_i + DV3_i + DV4_i + \dots + DV(n)_i \quad (3.19)$$

Ďalšie derivácie rovnice (3.16) sú:

$$\begin{aligned} y'' &= \frac{u'v - uv'}{v^2} = \frac{1}{v}(u' - y'v') \\ y''' &= \left( \frac{1}{v}(u' - y'v') \right)' = \frac{1}{v}(u'' - 2y''v' - y'v'') \\ y^{(4)} &= \left( \frac{1}{v}(u'' - 2y''v' - y'v'') \right)' = \frac{1}{v}(u''' - 3y'''v' - 3y''v'' - y'v''') \\ &\vdots \end{aligned} \quad (3.20)$$

Aj tu je vidieť, že jednotlivé derivácie tvoria Pascalov trojuholník. Všeobecný zápis je nasledovný:

$$y^{(n+1)} = \frac{1}{v} \left( u^{(n)} - \left( \sum_{k=1}^n \binom{n}{k} y^{(n-k+1)} v^{(k)} \right) \right)$$

Po dosadení jednotlivých členov  $DV(n)_i$  a  $DU(n)_i$  do derivácií (3.20) dostaneme:

$$\begin{aligned}
\frac{DY1_i}{h} &= \frac{1}{v}u & (3.21) \\
\frac{DY2_i}{\frac{h^2}{2!}} &= \frac{1}{v}\left(\frac{DU1_i}{h} - \frac{DY1_i}{h}\frac{DV1_i}{h}\right) \\
\frac{DY3_i}{\frac{h^3}{3!}} &= \frac{1}{v}\left(\frac{DU2_i}{\frac{h^2}{2!}} - 2\frac{DY2_i}{\frac{h^2}{2!}}\frac{DV1_i}{h} - \frac{DY1_i}{h}\frac{DV2_i}{\frac{h^2}{2!}}\right) \\
\frac{DY4_i}{\frac{h^4}{4!}} &= \frac{1}{v}\left(\frac{DU3_i}{\frac{h^3}{3!}} - 3\frac{DY3_i}{\frac{h^3}{3!}}\frac{DV1_i}{h} - 3\frac{DY2_i}{\frac{h^2}{2!}}\frac{DV2_i}{\frac{h^2}{2!}} - \frac{DY1_i}{h}\frac{DV3_i}{\frac{h^3}{3!}}\right) \\
&\vdots
\end{aligned}$$

Po úprave majú jednotlivé členy Taylorovej rady nasledovný tvar:

$$DY1_i = \frac{1}{v_i}(hu_i) \quad (3.22)$$

$$DY2_i = \frac{1}{2v_i}(DU1_i h - DY1_i DV1_i) \quad (3.23)$$

$$DY3_i = \frac{1}{3v_i}(DU2_i h - 2DY2_i DV1_i - DY1_i DV2_i) \quad (3.24)$$

$$DY4_i = \frac{1}{4v_i}(DU3_i h - 3DY3_i DV1_i - 2DY2_i DV2_i - DY1_i DV3_i) \quad (3.25)$$

$\vdots$

Všeobecný zápis:

$$DY(N)_i = \frac{1}{NDV0_i} \cdot \left( DU(N-1)_i \cdot h - \left( \sum_{k=1}^{N-1} (N-k) \cdot DY(N-k)_i \cdot DV(k)_i \right) \right) \quad (3.26)$$

Vyjadrenie jednotlivých členov Taylorovej rady vychádza z mojej bakalárskej práce [3]. Takto upravené členy sú následne použité na návrh jednotlivých typov integrátorov popísaných v kapitole 5.

## Kapitola 4

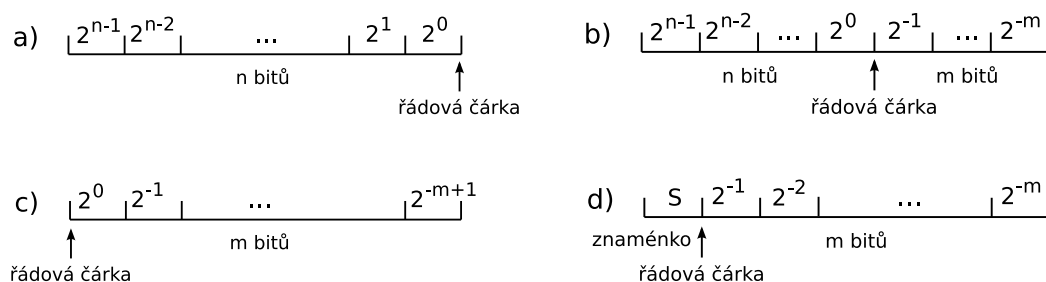
# Reprezentácia operandov

Čísla vo výpočtových systémoch sú reprezentované rôznymi spôsobmi v závislosti od zvolenej aritmetiky. Najpoužívanjšie sú reprezentácie čísiel v pevnej a v pohyblivej rádovej čiarky (ang. fixed point (FX) a floating point (FP)). Obe aritmetiky sú popísané v nasledujúcich podkapitolách. Informácie v tejto kapitole sú čerpané z [9], [10], [1], [7], [8].

### 4.1 Pevná rádová čiarka

Pri počítaní v pevnej rádovej čiarky sú čísla reprezentované na  $k$  bitoch v tvare  $n.m$ , kde prvých  $n$  bitov tvorí časť čísla pred desatinnou čiarkou, a zostávajúcich  $m$  bitov tvorí číslo za desatinnou čiarkou. Pozícia desatinnej čiarky je dopredu známa. V závislosti od jej pozície sa používajú rôzne formáty pevnej rádovej čiarky. Tie najpoužívannejšie sú znázornené na obrázku 4.1.

Obrázok 4.1a zobrazuje aritmetiku s nulovým počtom bitov za desatinnou čiarkou a teda sa jedná o celočíselnú aritmetiku na  $n$  bitoch. Vedľajší obrázok 4.1b zobrazuje aritmetiku s  $n$  bitmi pred desatinnou čiarkou a s  $m$  bitmi za desatinnou čiarkou. Ďalšia časť obrázku 4.1c je opakom 4.1a, kde je číslo reprezentované len za desatinnou čiarkou na  $m$  bitoch. Posledný obrázok 4.1d je podobný ako 4.1c, ale s pridaným znamienkovým bitom  $S$ . V tejto práci budem používať FX aritmetiku v tvare 4.1b s pridaním znamienkovým bitom na pozícii MSB (most significant bit).



Obr. 4.1: Rôzne formáty fixed point aritmetiky [1]

Vo fixed point aritmetike sa používajú rôzne kódy, napr. priamy kód, doplnkový kód či inverzný kód. V nasledujúcich kapitolách a pri návrhu integrátorov v pevnej rádovej čiarky budeme používať doplnkový kód.

## 4.2 Pohyblivá rádová čiarka

Čísla uložené v pohyblivej rádovej čiarkke sú tvorené exponentom a mantisou. Všeobecný tvar na získanie hodnoty uloženej vo FP je nasledujúci:

$$X = B^E \cdot M \quad (4.1)$$

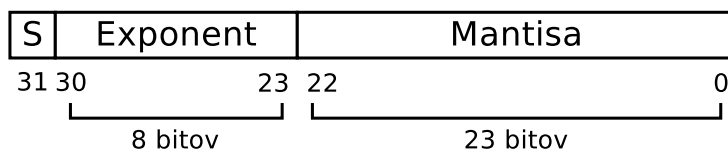
$X$  - výsledná hodnota

$B$  - základ sústavy

$E$  - hodnota exponentu

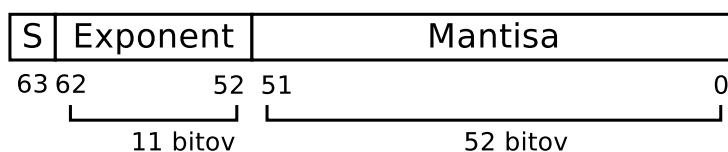
$M$  - mantisa

Zvýšením počtu bitov v exponente  $E$  sa zvýši rozsah hodnôt, ktorý je možný reprezentovať, a zvýšením počtu bitov v mantise  $M$  sa zvýši presnosť uložených čísel. Existuje veľa formátov uloženia čísel v pohyblivej rádovej čiarkke. Najpoužívanější a najrozšírenejší je štandard **IEEE 754**. Definuje vlastný formát uloženia čísel a viaceré formáty s rôznou presnosťou. Najpoužívanějšíe z nich sú formáty čísel s jednoduchou (single) a s dvojitou (double) presnosťou. Čísla s jednoduchou presnosťou sú uložené na 32 bitoch, kde MSB je znamienkový bit  $S$ , ďalších 8 bitov tvorí exponent  $E$ , a zvyšných 23 bitov tvorí mantisu  $M$ .



Obr. 4.2: IEEE 754 formát s jednoduchou presnosťou

Čísla s dvojitou presnosťou sú uložené v rovnakom formáte ako čísla s jednoduchou presnosťou, avšak líšia sa v počte bitov, ktorý je zväčšený na 64, kde MSB je znamienkový bit  $S$ , ďalších 11 bitov tvorí exponent  $E$ , a zvyšných 52 bitov tvorí mantisu  $M$ .



Obr. 4.3: IEEE 754 formát s dvojitou presnosťou

Znamienko  $S$  nadobúda hodnoty 0, čo značí kladné číslo; alebo 1, čo značí záporné číslo. Exponent je uložený v kóde s nepárnym posunutím o hodnotu BIAS. Táto hodnota je zvolená tak, aby uložený exponent bol vždy kladný. Pri jednoduchej presnosti má teda BIAS hodnotu 127 a pri dvojitej presnosti hodnotu 2047.

Hodnota mantisy je uložená v priamom kóde bez znamienka, znížená o hodnotu 1, keďže je tu použitá tzv. normalizácia. Mantis je normalizovaná do tvaru  $1.M$ , kde sa jednotka neukladá – je skrytá, čím sa ušetrí jeden bit. Hodnotu takto uloženého čísla získame zo vzťahu:

$$X_{754} = -1^S \cdot 2^{E-BIAS} \cdot (1, M) \quad (4.2)$$

$X_{754}$  - výsledná hodnota

$BIAS$  - 127 alebo 2047

$E$  - hodnota exponentu

$M$  - mantisa

Štandard IEEE 754 definuje aj špeciálne hodnoty ako kladnú/zápornú nulu, kladné/záporné nekonečno, či hodnotu NaN (not a number). Tieto hodnoty sú uvedené v tabuľke 4.1. Hodnota mantisy v normalizovanom tvare je v intervale  $< 1, 0; 2, 0$ ). Ak tomu tak nie je, ide o tzv. denormalizované číslo, a hodnota exponentu je braná ako  $-126$ . Štandard IEEE 754 definuje aj spôsob vykonávania základných matematických operácií, ktoré sú popísané v sekciách 4.2.1 a 4.2.2.

S (znamienko)	E (exponent)	M (mantisa)	význam
0/1	00000000	nulová hodnota	+/- 0
0/1	00000000	nenulová hodnota	+/- denormalizované číslo
0/1	1 - 254	ľubovoľná hodnota	+/- FP číslo
0/1	11111111	nulová hodnota	+/- $\infty$
0/1	11111111	nenulová hodnota	NaN

Tabuľka 4.1: Štandard IEEE 754 [10]



### 4.2.1 Súčet a rozdiel

Súčet a rozdiel v pohyblivej rádovej čiarke sa počíta podľa vzorcov

$$X + Y = (M_X \cdot 2^{E_X - E_Y} + M_Y) \cdot 2^{E_Y}, \text{ kde } E_X \leq E_Y \quad (4.3)$$

$$X - Y = (M_X \cdot 2^{E_X - E_Y} - M_Y) \cdot 2^{E_Y}, \text{ kde } E_X \leq E_Y \quad (4.4)$$

Postup výpočtu operácie súčtu/rozdielu v pohyblivej rádovej čiarke podľa štandardu IEEE 754 ([11], [13]) je nasledovný:

1. Na začiatku výpočtu sa obe čísla skontrolujú na výskyt špeciálnych hodnôt z tabuľky 4.1. Ak ide o špeciálne číslo, výsledok sa určí podľa tabuľky 4.2. Inak sa pokračuje bodom 2.
2. Vykoná sa porovnanie exponentov. Ak sú exponenty rozdielne, mantisu menšieho čísla posunieme o rozdiel exponentov doprava. Tým docielime rovnosť oboch exponentov. Pri posune je dôležité, aby sme nezabudli na 1, ktorá je skrytá, kvôli normalizácií. Posun sa vykonáva spolu s toutou 1.
3. Následne sa porovnávajú znamienka, a podľa výsledku sa vykoná súčet alebo rozdiel mantisy väčšieho čísla a posunutej mantisy. Ak došlo k pretečeniu, mantisa výsledku sa posunie o jeden bit doprava a hodnota exponentu sa zvýši. Aby bolo možné pretečenie detekovať, je potrebné sčítanie/odčítanie mantís vykonávať na sčítačke o jeden bit väčšej než je veľkosť mantisy (veľkosťou mantisy sa tu myslí počet bitov potrebných na uloženie mantisy aj so skrytou 1, čiže  $|1.M| + 1$ ).
4. Ak je to potrebné, vykoná sa normalizácia. Mantisa sa posunie o potrebný počet bitov doprava resp. doľava, tak, aby bola v tvare  $1.M$ . O daný počet bitov sa exponent zvýši resp. zníži.
5. Na koniec výpočtu sa skontroluje hodnota exponentu. Ak je hodnota maximálna, došlo k pretečeniu výsledku. Ten sa nastaví podľa znamienka na kladné alebo záporné nekonečno. V opačnom prípade, ak je hodnota exponentu minimálna (nulová), došlo k podtečeniu, a výsledok je nastavený podľa znamienka na kladnú alebo zápornú nulu.

Hodnota operandu 1	Hodnota operandu 2	Výsledok sčítania
FP číslo	$+/- \infty$	$+/- \infty$
$+/- \infty$	$+/- \infty$	$+/- \infty$
$+\infty$	$-\infty$	NaN
NaN	ľubovoľná hodnota	NaN

Tabuľka 4.2: Výsledok operácie sčítania so špeciálnymi hodnotami

#### 4.2.2 Násobenie a delenie

Násobenie a delenie v pohyblivej rádovej čiarky sa počíta nasledovne:

$$X \times Y = (M_X \cdot M_Y) \cdot 2^{E_X + E_Y} \quad (4.5)$$

$$X \div Y = (M_X \div M_Y) \cdot 2^{E_X - E_Y} \quad (4.6)$$

Postup výpočtu operácií násobenia a delenia podľa štandardu IEEE 754 ([11], [13]) je nasledovný:

1. Rovnako ako pri sčítaní, aj teraz sa na začiatku výpočtu skontroluje výskyt špeciálnych hodnôt oboch čísel podľa tabuľky 4.1. Ak ide o špeciálne číslo, výsledok sa určí podľa tabuľky 4.3 alebo 4.4. Pri operácii delenia je potrebné kontrolovať nepovolenú operáciu delenie nulou. Pokračuje sa bodom 2.
2. Pri násobení sa hodnota exponentu vypočíta ako súčet exponentov, od ktorého sa odpočíta hodnota *BIAS*. Pri operácii delenia sa hodnota exponentu vypočíta ako rozdiel exponentov, ku ktorému je pripočítaná hodnota *BIAS*.
3. Výsledná mantisa je rovná súčinu resp. podielu mantís. Pri násobení je potrebné použiť násobičku, ktorej bitová šírka sa rovná dvojnásobku počtu bitov mantisy 1.*M*. Ak dôjde k pretečeniu alebo k podtečeniu mantisy, vykoná sa posun mantisy doprava resp. doľava a hodnota exponentu sa zvýši resp. zníži.
4. Pokiaľ je to potrebné, prebehne normalizácia.
5. Na konci výpočtu sa skontroluje hodnota exponentu. Postupuje sa rovnako ako pri operácii súčtu: ak je hodnota exponentu maximálna, nastaví sa podľa znamienka na kladné alebo záporné nekonečno. V opačnom prípade, ak je hodnota exponentu minimálna (nulová), výsledok sa nastaví podľa znamienka na kladnú alebo zápornú nulu.

Hodnota operandu 1	Hodnota operandu 2	Výsledok násobenia
kladné/záporné FP číslo	+/- $\infty$	+/- $\infty$
nula	+/- $\infty$	NaN
+/- $\infty$	+/- $\infty$	+/- $\infty$
NaN	ľubovoľná hodnota	NaN

Tabuľka 4.3: Výsledok operácie násobenia so špeciálnymi hodnotami

Hodnota operandu 1	Hodnota operandu 2	Výsledok násobenia
kladné/záporné FP číslo	+/- 0	+/- $\infty$
0	0	NaN

Tabuľka 4.4: Výsledok operácie delenia so špeciálnymi hodnotami

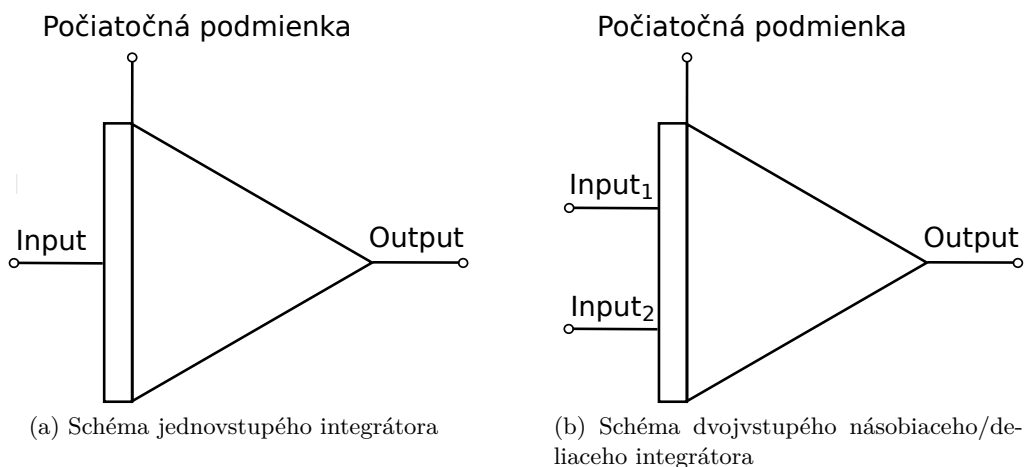
## Kapitola 5

# Numerické integrátory

Numerický integrátor je hardvérový komponent, ktorý slúži na výpočet numerickej integrácie. Podľa spôsobu výpočtu a komunikácie medzi komponentmi integrátora sa numerické integrátory delia na:

- sériovo-sériové integrátory, skrátene sériové integrátory (sériová komunikácia aj výpočet)
- sériovo-paralelné integrátory (sériová komunikácia a paralelný výpočet)
- paralelno-paralelné integrátory, skrátene paralelné integrátory (paralelná komunikácia aj výpočet)

V tejto práci sa budeme zaoberať paralelnými numerickými integrátormi, kvôli ich jednoduchosti a rýchlosti výpočtu. Ďalej môžeme numerické integrátory rozdeliť na jednovstupé a dvojevstupé. Jednovstupý integrátor vykonáva integráciu vstupnej hodnoty a posiela ju na výstup. Schéma tohto typu integrátora je znázornená na obrázku 5.1a a ako z názvu vyplýva obsahuje jeden vstup pre vstupnú hodnotu. Ďalej obsahuje jeden vstup pre počiatočnú podmienku a jeden výstup pre výsledok výpočtu. Dvojevstupové integrátory obsahujú jeden vstup pre počiatočnú podmienku, dva vstupy pre prívod operandov a jeden výstup pre výsledok výpočtu. Schéma integrátora je znázornená na obrázku 5.1b.



Obr. 5.1: Schéma jednovstupého a dvojevstupého integrátora

Tieto integrátory rozdeľujeme podľa použitej operácie na násobiace a deliace integrátory. Násobiaci integrátor vykonáva deriváciu násobenia dvoch vstupných hodnôt. Deliaci integrátor zase deriváciu delenia dvoch vstupných hodnôt. Výsledná hodnota je poslaná na výstup.

V nasledujúcich podkapitolách predstavíme jednotlivé návrhy paralelných násobiacich a paralelných deliacich integrátorov, oba typy v pevnej a v pohyblivej rádovej čiarke. Predstíme taktiež jednovstupý paralelný integrátor v pevnej a pohyblivej rádovej čiarke.

## 5.1 Návrh násobiaceho integrátora v pevnej rádovej čiarke

Násobiaci integrátor počíta rovnicu (3.6) pomocou (3.11) – (3.14). Na základe týchto rovníc bol vytvorený návrh paralelného násobiaceho integrátora, ktorý je na obrázku 5.2. Návrh vychádza z práce V. Závadu [12] a bol upravený a rozšírený o počet registrov  $DR(N - 1)$  a  $DQ(N - 1)$ , ktoré slúžia na ukladanie prichádzajúcich členov. Rozšírený bol aj počet registrov pre uloženie kroku  $h$  integračnej metódy a jeho podielov. Počet týchto registrov je o jeden menší ako rád  $ORD$  použitej Taylorovej metódy ( $N = ORD$ ), keďže pri výpočte sa používajú predchádzajúce členy Taylorovej metódy. Rád metódy je vhodné a potrebné zvoliť vzhľadom na použitú aritmetiku a požadovanú presnosť.

Každý člen  $DY(N - 1)$  obsahuje postupné delenie integračného kroku  $h$ . Tieto hodnoty sú predpočítané a uložené v sade registrov  $h$ . Pre optimalizáciu a ušetrenia miesta však nie je potrebné uložiť všetkých  $N$  hodnôt, stačí uložiť len tie hodnoty, ktorých deliteľ je nepárne číslo. Ostatné hodnoty je možné vypočítať jednoduchým posunom registra doprava, čo je vlastne delenie číslom 2. Je síce potrebné pridať riadice signály pre ovládanie posunu jednotlivých registrov avšak výsledkom je zníženie počtu registrov  $h$  o polovicu. Počet operácií potrebných na výpočet sa nezvýši, keďže posun registrov je možné vykonávať v predstihu a paralelne s inými operáciami.



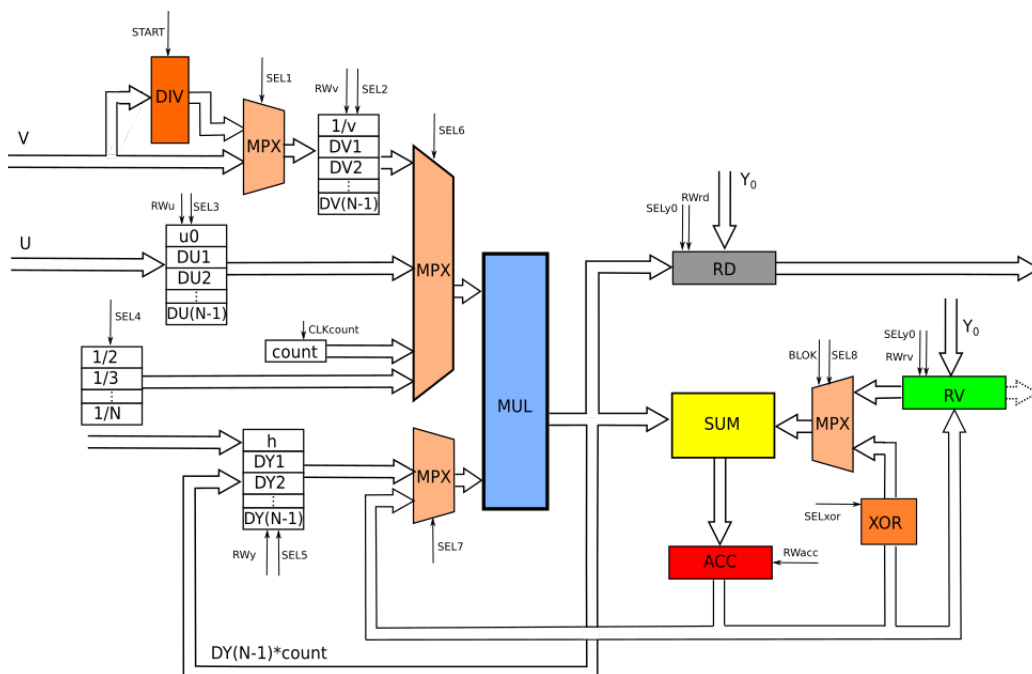
Tabuľka 5.1: Význam skratiek použitých v obrázku 5.2.

18

výpočtu sú vstupné hodnoty ukladané postupne do ďalších registrov. Po uložení sa začne samotný výpočet. Hodnota aktuálne počítaného člena Taylorovej rady sa postupne ukladá do registra *RD* a odtiaľ je privedená na výstup integrátora. Do registra *RV* je ukladá súčet jednotlivých členov Taylorovej rady - čiže celkový výsledok derivácie vstupných hodnôt. Táto hodnota nás bude zaujímať napríklad keď je integrátor "posledný" v zapojení, čiže jeho výstup nie je nikam pripojený a teda nepotrebujeme použiť hodnoty jednotlivých členov.

## 5.2 Návrh deliaceho integrátora v pevnej rádovej čiarke

Tento integrátor počíta rovnicu (3.16) pomocou členov (3.22) - (3.25). Podobne, ako u násobiaceho integrátora, bol z týchto rovníc vytvorený návrh paralelného deliaceho integrátora. Návrh vychádza z mojej bakalárskej práce [3] a bol upravený podobne ako predošlí násobiaci integrátor zvýšením počtu registrov  $DU(N-1)$  a  $DV(N-1)$ . Taktiež sa zvýšil počet registrov  $DY(N-1)$ , ktoré slúžia na uloženie jednotlivých členov Taylorovej rady, keďže na výpočet nasledujúceho člena je použitý predchádzajúci člen. Sada týchto registrov okrem hodnôt  $DY(n-1)$  obsahuje register pre uloženie kroku  $h$  integračnej metódy. Deliaci integrátor, narozdiel od násobiaceho integrátora, nepotrebuje pri výpočte hodnoty podielov kroku numerickej metódy a teda stačí uložiť len jednu hodnotu  $h$ . Deliaci integrátor ale používa sadu registrov, ktorá obsahuje hodnoty  $1/N$ . Podobne ako u násobiaceho integrátora je možné zmenšiť počet týchto registrov na polovicu s využitím aritmetického posunu (operácia *shift*). Register *count* obsahuje *counter*, ktorý sa postupne inkrementuje v každom cykle výpočtu, a s ktorým sa násobí hodnota  $DY(N-1)$ . Vyplýva to z rovníc (3.22) - (3.25) a (3.26), kde pri výpočte každého člena Taylorovej rady sa hodnota  $DY(N-k)$  násobí hodnotou  $N-k$ . Výsledná vynásobená hodnota je uložená opätovne do daného registra  $DY(N-1)$  a znova použitá v ďalšom výpočte. Tým sa ušetrí operácia násobenia v ďalšom cykle výpočtu.



Obr. 5.3: Paralelno-paralelný deliaci integrátor [3].

Skratka	Popis
<b>V, U</b>	vstupné operandy integrátora
<b>1/v, DV1, ... DV(N-1)</b>	sada registrov jednotlivých členov vstupného operandu V
<b>u0, DU1, ... DU(N-1)</b>	sada registrov jednotlivých členov vstupného operandu U
<b>count</b>	čítač cyklov výpočtu - N
<b>1/2, 1/3, ... 1/N</b>	sada registrov s jednotlivými podielmi
<b>h, DY1, ... DY(N-1)</b>	sada registrov s krokom metódy a s vypočítanými členmi
<b>DIV</b>	delička
<b>XOR</b>	invertovanie hodnoty v dvojkovom doplnku (xor +1)
<b>MPX</b>	multiplexor 2-1, 4-1
<b>START</b>	povoľovací signál na spustenie výpočtu
<b>SEL1, SEL6, SEL7, SEL8</b>	riadiace signály multiplexorov
<b>SEL2, SEL3, SEL4, SEL5</b>	riadiace signály registrov
<b>SELxor</b>	signál na invertovanie vstupnej hodnoty
<b>CLKcount</b>	signál na zvýšenie hodnoty čítača
<b>RWv, RWu, RWy</b>	povoľovacie signály sád registrov (READ/WRITE)

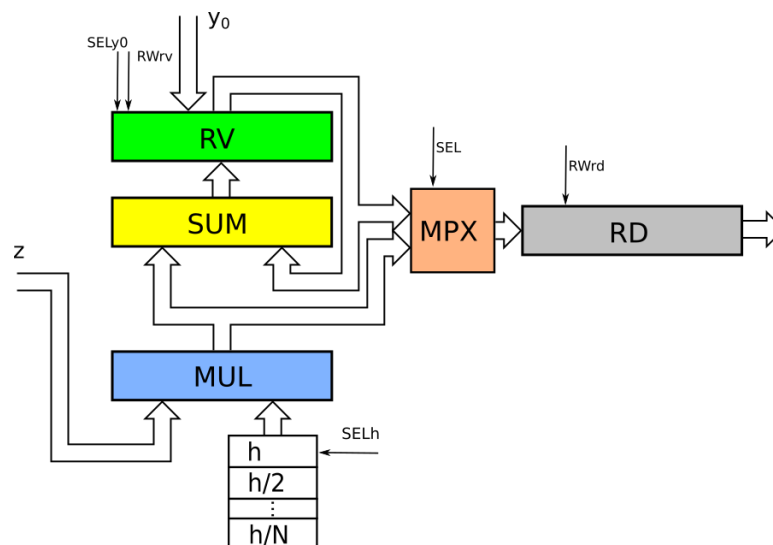
Tabuľka 5.2: Význam skratiek v obrázku 5.3.

Podobne ako pri násobiacom integrátore, aj pri paralelnom deliacom integrátore sú na začiatku výpočtu vynulované registre pomocou signálu *RESET* a nastavené signály multiplexorov *MPX*. Ďalej je do registrov *RD* a *RV* nahraná počiatočná podmienka a do registra *h* je nahraný integračný krok. Do registrov  $1/n$  sú uložené predpočítané konštanty. Po prijatí hodnôt *U* a *V* sa začne výpočet. Hodnota *V* je privedená do deličky *DIV* a spustí sa výpočet  $1/v$  s použitím deliaceho algoritmu *SRT*. Delenie je realizované len raz počas celého výpočtu, keďže ide o veľmi náročnú operáciu. Po skončení operácie delenia je výsledok uložený do sady registrov *DVN* a to konkrétne do registra  $1/v$ . V ďalšom vykle výpočtu je vstupná hodnota *V*, privedená priamo do sady registrov *DVN* bez delenia. Hodnota *U* je uložená do sady registrov *DUN*. Kvôli optimalizácii je paralelne s delením realizovaný výpočet násobenia *uh*. Po skončení výpočtu je výsledok aktuálneho člena uložený do registra *RD* a privedený na výstup deliaceho integrátora. Suma jednotlivých členov je postupne ukladaná v každom cykle do registra *RV*, z ktorého, rovnako ako v násobiacom integrátore, je možné získať hodnotu privedeným na výstup.

### 5.3 Návrh jendovstupého integrátora

Tento integrátor bol navrhnutý podľa rovníc (3.5) na základe ktorých počíta diferenciálnu rovnicu (3.1). Z toho vyplýva, že vykonáva numerickú integráciu vstupnej hodnoty a následne výslednú hodnotu posiela na výstup. Schéma integrátora je na obrázku 5.4 a vychádza z práce J. Opálku [5]. Na naštartovanie výpočtu slúži počiatočná podmienka. Tá je nahraná na začiatku výpočtu do registra *RV*. Následne je cez multiplexor *MPX* privedená a uložená do registra *RD*. V sade registrov *h* je uložený integračný krok metódy a jeho podiely. Rovnako ako vpredchádzajúcich prípadoch je možné počet registrov znížiť na polovicu. Vstupnú hodnotu *Z* nie je potrebné ukladať do registrov, keďže sa nepoužíva pri výpočte ďalších členov. Jednotlivé hodnoty členov sú ukladané do registra *RV* a poslané na výstup integrátora. Ak chceme získať výslednú hodnotu Taylorovej rady rovnice (3.1), stačí priviesť hodnotu z

registra  $RV$  cez multiplexor  $MPX$  do registra  $RD$ , kde je hodnota sprístupnená na výstupe integrátora.



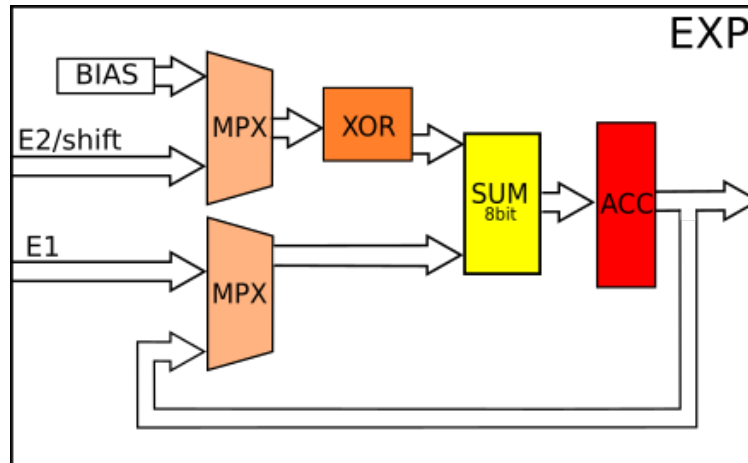
Obr. 5.4: Paralelný jednovstupový integrátor v pohyblivej rádovej čiarke.

## 5.4 Návrh komponentu pre spracovanie exponentov

V pohyblivej rádovej čiarke sú operácie násobenia, delenia a sčítania (odčítania) zložitejšie, nakoľko je k týmto operáciám pridaná aj práca s výpočtom znamienka, exponentu a mantisy. Výpočet exponentov je možné vykonávať v každej z komponent (násobička, delička, sčítačka) samostatne alebo pomocou jednej komponenty zdieľanej medzi týmito komponentmi. Pri zdieľanom prístupe je možné znížiť priestorovú zložitosť zapojenia, ale zvýši sa tým náročnosť riadenie kontroléra a synchronizácia. V deliacom integrátore to môže spôsobiť zdržanie výpočtu, keď sa operácia delenia a násobenia vykonávajú paralelne. Avšak táto situácia nastane len raz na začiatku výpočtu. Predošlé popísané návrhy násobačeho, deliaceho a jednovstupového integrátora je možné použiť ako pri implementácii v pevnej rádovej čiarke tak aj v pohyblivej rádovej čiarke. Integrátory v pohyblivej rádovej čiarke vychádzajúce z popísaných návrhov pracujú s exponentami v každej operácii samostatne. V nasledujúcich sekciách si popíšeme druhú variantu spracovania exponentu so sdielaným komponentom.

Blokové schémy zapojenia integrátorov pracujúcich v pohyblivej rádovej čiarke vychádzajú z predchádzajúcich návrhov, ale boli rozšírené o prácu so znamienkami, s mantisou a s exponentmi. Samotný výpočet exponenta sa deje v zdieľanom komponente *EXP*, ktorého návrh je zobrazený na obrázku 5.5.





Obr. 5.5: EXP - blok pracujúci s exponentmi.

Skratka	Popis
<b>E1</b>	vstupná hodnota väčšieho exponenta
<b>E2/shift</b>	vstupná hodnota menšieho exponenta/hodnota posunu
<b>BIAS</b>	hodnota 127 alebo 2047
<b>SUM (8/11bit)</b>	paralelná 8 alebo 11 bitová sčítačka
<b>INV</b>	invertor

Tabuľka 5.3: Význam skratiek použitých v obrázku 5.5.

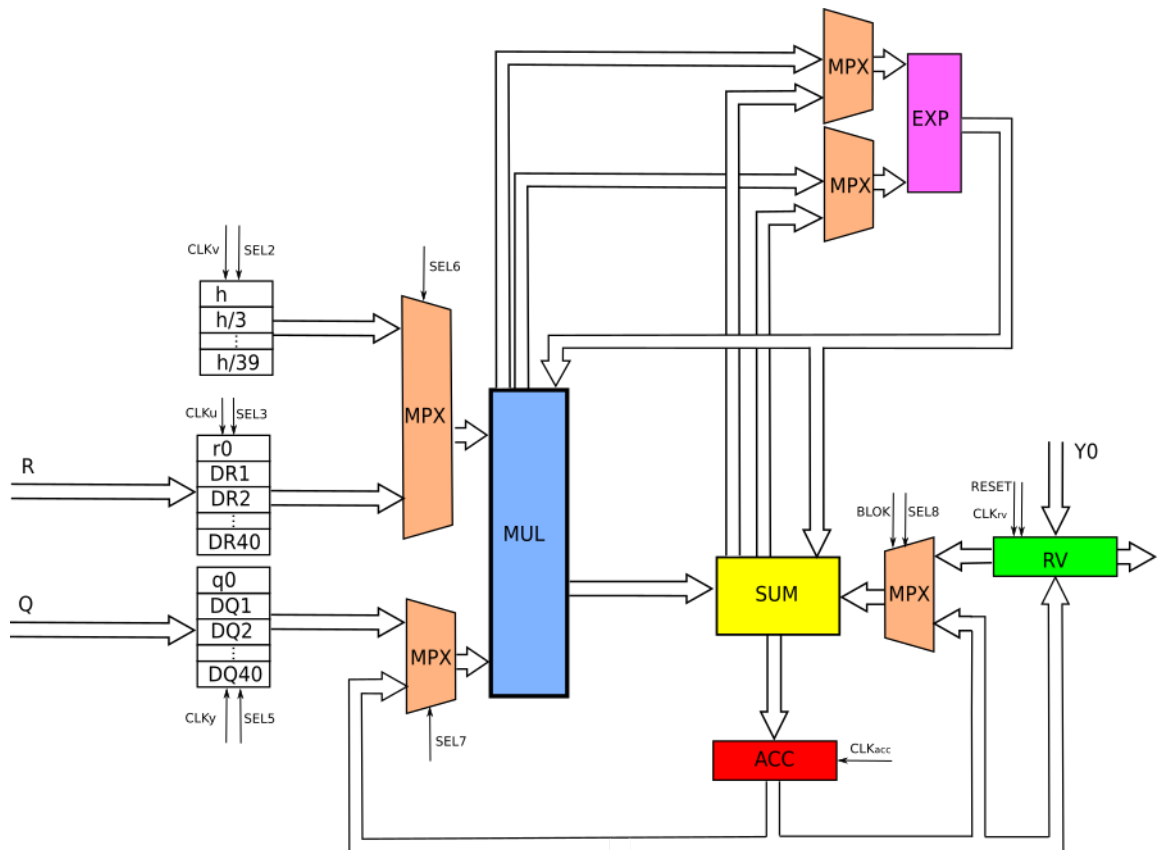
Komponent *EXP* slúži na vykonávanie výpočtov s exponentmi, ktoré sú popísané v sekciách 4.2.1 a 4.2.2. Obsahuje dva vstupy, do ktorých sú privedené jednotlivé exponenty z komponentov *DIV*, *MUL* a *SUM* a jeden výstup. Celý komponent *EXP* pracuje na počte bitov rovnému veľkosti exponenta, ktorá je závislá na použitej aritmetike. Pri jednoduchšej presnosti (32 bitov) na 8 bitoch alebo na 11-bitoch pri dvojitej presnosti (64 bitov). Komponent teda obsahuje buď 8 bitovú alebo 11 bitovú sčítačku a ostatne komponenty v rovnakej bitovej šírke. Pri vykonávaní operácie súčiny alebo rozdielu slúži komponent *EXP* na výpočet rozdielu exponentov. Výsledná hodnota rozdielu je privedená naspäť do sčítačky *SUM* cez register *ACC*. O túto hodnotu je následne v sčítačke posunutá hodnota mantisy menšieho čísla doprava. Výsledná hodnota mantisy je potom súčet/rozdiel mantisy väčšieho čísla a posunutej mantisy. Znamienko a exponent sú rovné hodnote väčšieho čísla z počítaných operandov. Pri operácii delenia alebo násobenia slúži komponent *EXP* na výpočet výsledného exponentu. Podľa typu operácie sa vykoná súčet alebo rozdiel prijatých exponentov. Táto hodnota je uložená v registri *ACC* a privedená naspäť do sčítačky *SUM*<sub>8/11bit</sub> s hodnotou z registra *BIAS*. Vykoná sa súčet alebo rozdiel týchto hodnôt. Získame tak hodnotu nového exponentu, ktorá je privedená naspäť do násobičky *MUL* alebo deličky *DIV*. Ak však je po delení alebo násobení potrebné vykonať posun mantisy hodnota vypočítaného exponentu je privedená z *ACC* cez multiplexor *MPX* naspäť do sčítačky *SUM*<sub>8/11bit</sub> a na vstup *E2/Shift* je privedená hodnota posunu. Vykoná sa rozdiel hodnôt, čím vykonáme posun exponenta. Následne je výsledná hodnota privedená do násobičky *MUL* alebo deličky *DIV* a použitá ako nová hodnota exponentu.

Kvôli prehľadnosti neobsahuje schéma zapojenia znázornenie výpočtu znamienka a rozdelenie operandov na znamienko, exponent a na mantisu, a ich opätovné zloženie. Tieto

operácie sa vykonávajú v komponentoch *DIV*, *MUL* a *SUM*.

## 5.5 Návrh násobiaci integrátora v pohyblivej rádovej čiarke

Návrh paralelného násobiaceho integrátora v pohyblivej rádovej čiarke so zdieľaným komponentom *EXP* je na obrázku 5.2. Čísla v pohyblivej rádovej čiarke môžeme zobrazit presnejšie ako čísla v pevnej rádovej čiarke, čiže pri uložení malých čísel v pohyblivej rádovej čiarke dochádza k menšej zaokrúhľovacej chybe. Z tohto dôvodu je možné počítať Taylorovu radu s použitím väčšieho počtu členov a zvýšiť tak presnosť výpočtu. Integrátory v pohyblivej rádovej čiarke teda môžu počítať viac členov Taylorovej rady a teda počet registrov  $DR(N-1)$  a  $DQ(N-1)$  sa zvýši. Sada registrov  $h/i, i = 2, 3..N$  obsahuje  $N-1$  registrov. Počet týchto registrov je možné znížiť rovnako ako pri integrátoroch v pevnej rádovej čiarke.



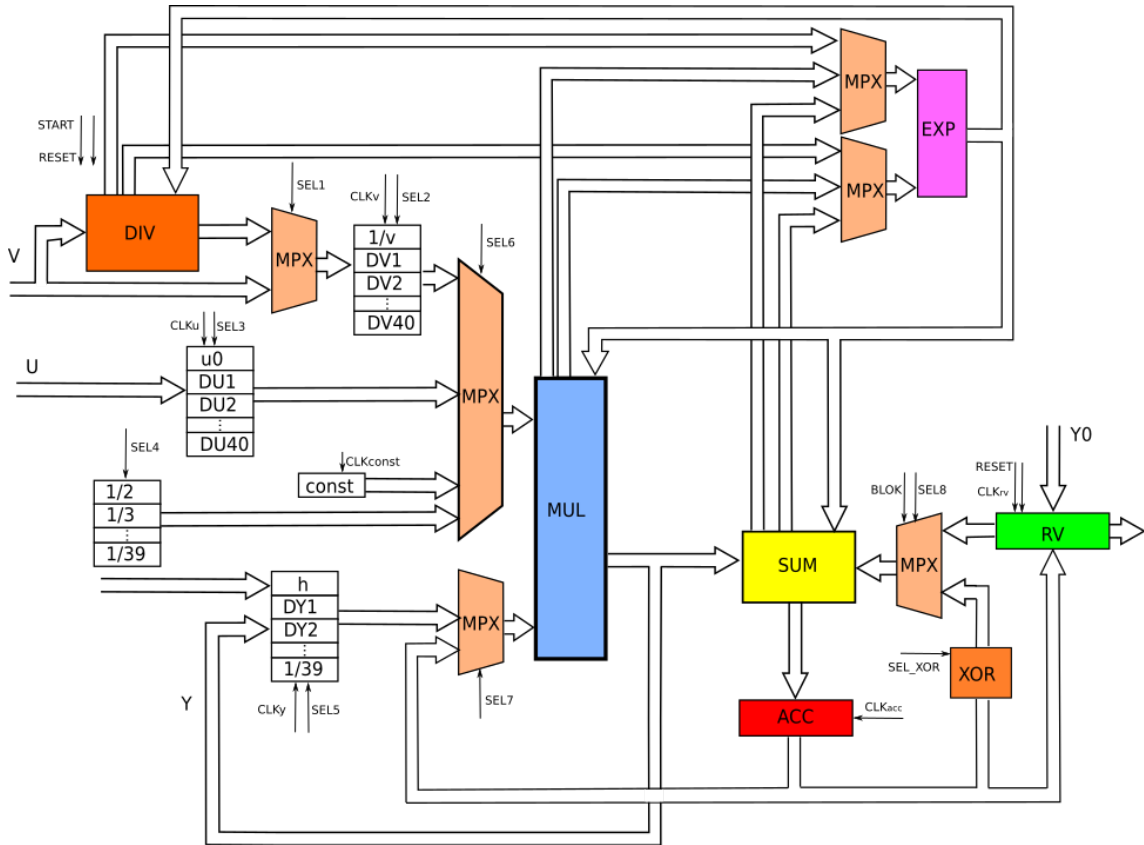
Obr. 5.6: Paralelno-paralelný násobiaci integrátor v pohyblivej rádovej čiarke

Spôsob výpočtu paralelného násobiaceho integrátora v pohyblivej rádovej čiarke so zdieľaným komponentom *EXP* je takmer rovnaký ako pri násobiacom integrátore v pevnej rádovej čiarke. Rozdielne sa vykonávajú operácie násobenia a sčítania. Po privedení hodnôt na vstupy násobičky *MUL* alebo sčítačky *SUM* sú čísla v FP uložené do pomocných registrov v týchto komponentoch. Z týchto registrov sú jednotlivé časti FP čísla roz distribuované do samostatných výpočetných obvodov. V komponente *MUL* sú znamienka privedené ku komponentu *XOR*, ktorý vykonáva nonekvivalenciu. V komponente *SUM* je výsledné zna-

mienko rovné znamienku väčšieho zo vstupných čísel. Exponenty, ako je popísané vyššie, spracúva komponent *EXP*. Mantisy sú privedené do paralelnej násobičky alebo sčítačky. Veľkosť výsledku násobičky je dvojnásobná ako veľkosť mantisy so skrytou jednotkou t.j. 48 bitov pri jednoduchovej presnosti a 106 bitov pri dvojitej presnosti. Veľkosť sčítačky je rovná veľkosti mantisy so skrytou jednotkou t.j. 24 alebo 53 bitov. Výsledné hodnoty jednotlivých častí čísla v pohyblivej rádovej čiarke sú na výstupe spojené do jedného čísla a uložené do výstupného registra daného komponentu (*MUL* alebo *SUM*). Následne sú poskytnuté na jeho výstupe k ďalšiemu výpočtu.

## 5.6 Deliaci integrátor v pohyblivej rádovej čiarke

Paralelný deliaci integrátor v pohyblivej rádovej čiarke so zdieľaným komponentom *EXP* je najzložitejší z navrhnutých integrátorov. Obsahuje všetky spomínané operácie: sčítanie, odčítanie, násobenie a delenie; a všetky vykonáva v pohyblivej rádovej čiarke, kde práca s exponentami je vykonávaná v jednej komponente *EXP*. Operácie násobenia a sčítania sa vykonávajú rovnako ako v paralelnom násobiacom integrátore v pohyblivej rádovej čiarke. Operácia delenia sa vykonáva iba raz za celý výpočet a to paralelne s operáciou násobenia, ako je tomu aj v deliacom integrátore v FX aritmetike. Tu však môže dôjsť ku kolízii v použití komponentu *EXP*. Keďže delenie je náročnou operáciou kde dĺžka výpočtu závisí na použití daného algoritmu. Podľa zvoleného algoritmu je potom potrebné zvoliť, ktorá operácia bude mať prednosť v použití komponentu *EXP*. Po uvoľnení *EXP* násobičkou *MUL* je komponent *EXP* pridelený deličke *DIV* na výpočet exponentov. Ak je použitý zdĺhavejší algoritmus pri výpočte delenia (napr. SRT algoritmus použitý v deliacom integrátore v FX aritmetike), čiže operácia delenia trvá dlhší čas ako operácia násobenia, je vhodnejšie prenechať najprv komponentu *EXP* operáciám násobenia. Ak je deliaci algoritmus veľmi rýchly (napr. ) a čas výpočtu je podobný času výpočtu operácií násobenia, tak potom nezáleží ktorá z komponent bude prvá využívať *EXP*. Po skončení delenia je podiel uložený do registra  $1/v$  a pokračuje sa v ďalšom výpočte. Výpočet prebieha podobne ako pri deliacom integrátore bez komponentu *EXP*. Rozdielne sa vykonávajú operácie delenia, násobenia a sčítania. Ďalší rozdiel je v komponente *XOR*, ktorá bola nahradená komponentom *INV*. Prevrátenie hodnoty čísla sa v pevnej rádovej čiarke použitím doplnkového kódu deje operáciou *XOR* a pričítaním jednotky ako tomu je v predchádzajúcom deliacom integrátore. V pohyblivej rádovej čiarke sa však inverovanie hodnoty čísla vykonáva jednoduchým invertovaním znamienkového bitu *S* vstupného operandu. Toto invertovanie sa vykonáva v spomínanej komponente *INV*.



Obr. 5.7: Paralelno-paralelný deliaci integrátor v pohyblivej rádovej čiarke

## 5.7 Sústava diferenciálnych rovníc

Integrátory, ktoré sme si popísali je možné medzi sebou zapojiť a riešiť tak diferenciálnu rovnicu. Po zapojení integrátorov nebude nutné predpočítavať vstupné hodnoty, ale stačí zadať počiatočné podmienky na naštartovanie výpočtu a všetky ostatné hodnoty sa dopočítajú. Zvoľme konkrétnu diferenciálnu rovnicu s operáciou násobenia v tvare:

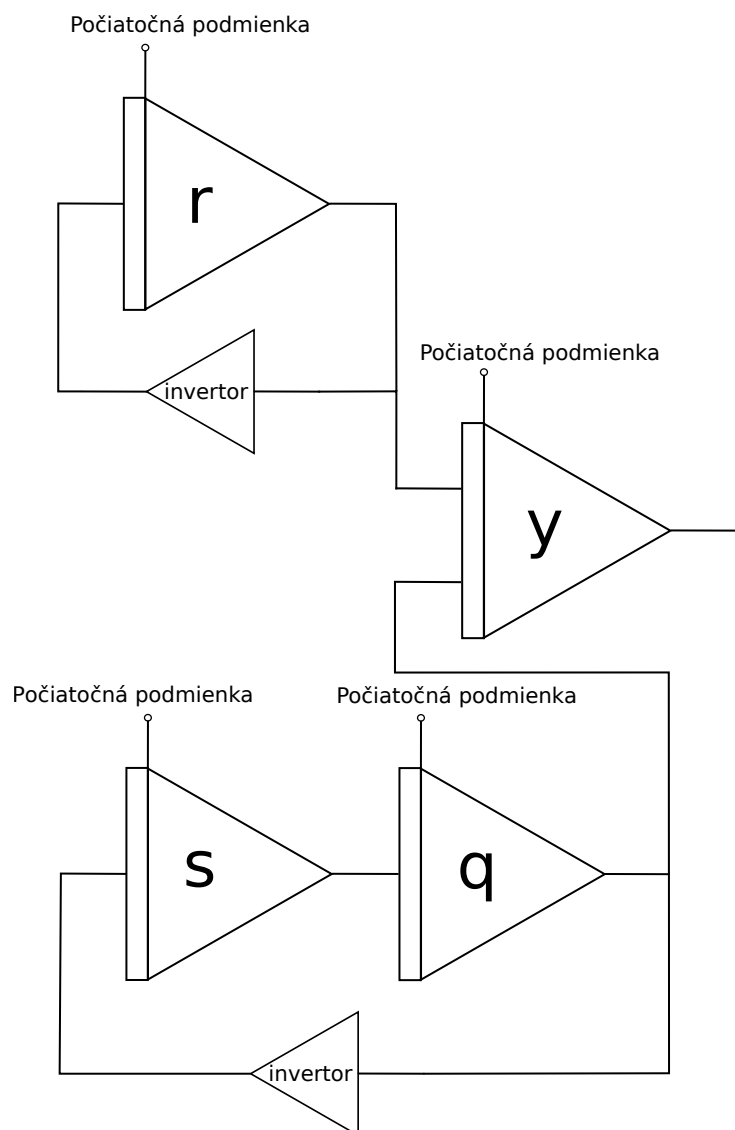
$$y' = \sin(t)e^{-t}, \quad y(0) = 1 \quad (5.1)$$

Rovnicu prevedieme do sústavy diferenciálnych rovníc:

$$\begin{aligned} y' &= qr, & y(0) &= 1 \\ r' &= -r, & r(0) &= 1 \\ q' &= s, & q(0) &= 0 \\ s' &= -q, & s(0) &= 1 \end{aligned} \quad (5.2)$$

Zo sústavy rovníc vytvoríme schému zapojenia integrátorov znázornenú na obrázku 5.8. Schému tvoria tri jednovstupové integrátory  $r$ ,  $q$  a  $s$ , dvojevstupový násobač  $y$  a dva inverotri. Invertor je komponenta, ktorá invertuje prichádzajúcu hodnotu. Integrátor

$r$  a invertor tvoria spolu funkciu  $e^{-t}$ . Integrátory  $r$ ,  $s$  a invertor tvoria spolu funkciu  $\sin(t)$ . Nakoniec sú tieto hodnoty privádzané na vstup násobiaceho integrátora  $y$ , ktorý počíta diferenciálnu rovnicu 5.1. Zo schémy zapojenia je možné následne vytvoriť VHDL kód a vykonať výpočet na FPGA. Takýto postup je vhodný pre rozsiahle diferenciálne rovnice s väčším počtom operácií, ktorých výpočet by bol vykonávaný veľmi často. Vhodnejší spôsob použitia sa ponúka prepojovacia sieť medzi rôznymi integrátormi. Pred výpočtom by sa sieť nakonfigurovala podľa danej diferenciálnej rovnice a výpočet by mohol prebiehať bez nutnosti syntézy. Táto varianta je síce priestorovo náročná, ale ponúka variabilitu výpočtu. V ďalšej kapitole si predstavíme implementáciu popísaných integrátorov a taktiež implementáciu sústavy diferenciálnych rovníc pomocou metódy bez použitia prepojovacej siete.



Obr. 5.8: Schéma zapojenia integrátorov [3]

## Kapitola 6

# Implementácia integrátorov vo VHDL

Jednotlivé integrátory predstavené v kapitole 5 boli popísané v jazyku VHDL v prostredí Xilinx ISE a následne odsimulované v nástroji Model Sim požitím *testbenchu*. Následne niektoré z integrátorov boli otestované na programovateľnom hradlovom poli FPGA. Pre priestorovú zložitosť paralelných integrátorov bola snaha implementovať integrátory na rozsiahlejšom čipe FPGA a to konkrétne na VIRTEX-5 s čipom XCS50. Pre prácu s týmto čipom je potrebná plná verzia programu Xilinx ISE s licenciou. Poskytnutie licencie od školy pre študenta však nie je možné. Riešením bolo nakoniec použitie licencie pomocou pripojenia sa na fakultnú sieť cez VPN. Následne bol integrátor implementovaný na VIRTEX-5, avšak skončenie výpočtu bolo možné detekovať len rozsvietením diódy, čo je nepoužiteľný spôsob pre získanie hodnoty výsledku. Preto bolo snahou výsledok zobrazit na LCD displeji. Po pre zložitú prácu a neúspešné sprevádzkovanie displeja, prípadne iných výstupných metód, boli integrátory naprogramované a otestované v FPGA Spartan na prípravku Fitkit. Na Fitkite bola odskúšaný jednovstupový integrátor s 32 bitovou aritmetikou. Tento integrátor bol vybraný kvôli jeho najmenšej priestorovej zložisti. Ostatné integrátory a verzie integrátorov nie je možné nahráť na FPGA použité vo Fitkite. I kvôli tomuto bola snaha otestovania integrátorov na väčšom FPGA Virtex-5.

Jednotlivé časti integrátorov boli popísané samostante a následne vzájomným prepojením bola vytvorená štruktúra prepojenia, ktorá tvorí celý integrátor. Jednotlivé časti sú u každého integrátora sú jednotlivé kroky výpočtu riadené kontrolérom. Ten je tvorení cyklickým konečným automatom. Cyklus vypláva z rovníc (3.15) a (3.26), keďže výpočet jednotlivých členov je podobný, avšak so zväčšujúcim sa počtom operácií. Samozrejme po skončení výpočtu je možné signálom *RESET* prejsť do počiatočného stavu a zadaným novú počiatočnej podmienky začať nový výpočet.

### 6.1 Integrátory v pevnej rádovej čiarke

Násobiaci aj deliaci integrátor v pevnej rádovej čiarke kódujú čísla v dvojkovom kóde. Paralelný dvojjstupový násobiaci integrátor vo fixpoint aritmetike bol implementovaný v dvoch verziách a to na 32 a 64 bitoch. Veľkosť použitej aritmetiky je však možné jednoducho zvážať, keďže celý kód je písaný genericky. Taktiež je možné zmeniť počet cyklov výpočtu, čiže počet počítaných členov Taylorovej rady, avšak v závislosti od počtu cyklov  $N$  je nutné upraviť aj počet registrov  $h/i$ , kde  $i = 2..N$ . Pre 32 i 64 bitovú verziu bola použitá FX

aritmetika s jedným znamienkovým bitom, s 1 bitom pred desatinnou čiarkou a s 30 bitmi respektívne 62 bitmi za desatinou čiarkou. Výpočet počíta s Taylorovou metódou 8. rádu a teda výpočet prebieha až po 8. člen Taylorovej rady t.j. DY8.

Paralelný deliaci integrátor vo fixpoint aritmetike bol implementovaný, podobne ako násobiaci integrátor, v 32 a 64 bitovej verzií. Pri návrhu integrátorov sa kládol dôraz nie len na rýchlosť výpočtu ale aj na priestorovú zložitosť. Z rovnice 3.26 vyplýva, že členy  $DYN$  sú násobené číslom  $N - k$ , kde  $k$  je minimálne 1 a teda najväčšie číslo je  $N - 1$ . Aby bolo možné toto číslo zobraziť bolo potrebné zvoliť vhodnú aritmetiku t.j. vhodný počet čísel pred a za desatinnou čiarkou. Čím väčší počet bitov sa nachádza pred desatinnou čiarkou, tým sa zväčšuje interval zobraziteľných čísel, avšak sa znižuje počet bitov za desatinnou čiarkou. Pri znižovaní počtu čísel za desatinnou čiarkou sa znižuje hustota zobrazenia čísel a v konečnom dôsledku presnosť. V dvojkovom doplnkovom kóde pri 1 bite pred desatinou čiarkou sme schopní vypočítať maximálne dva členy Taylorovej rady. Pri dvoch bitoch sú to 4 členy, pri troch bitoch 8 členov Taylorovej rady a pri piatich bitoch 16 členov Taylorovej rady. Keďže pri použití Taylorovej metódy 4 rádu nemusí byť riešenie dostatočne presné, v 32 bitová verzia počíta s Taylorovou metódou 8. rádu a je teda použitá aritmetika s jedným znamienkovým bitom, s tromi bitmi pred desatinnou čiarkou a zvyšných 28 bitov tvorí časť za desatinnou čiarkou. Vyšší rád metódy nie je použiteľný, keďže hodnoty vyšších členov by boli veľmi malé a teda nezobraziteľné v danej aritmetike. V 64 bitovej verzií bola veľkosť použitých bitov pred desatinnou čiarkou zväčšená o jeden bit. Za desatinnou čiarkou sa potom nachádza 59 bitov. V tejto verzií deliaceho integrátora sme teda schopní vypočítať v danej aritmetike až 16 členov Taylorovej rady.

Stav	SELy0	SEL1	SEL2	SEL3	SELQ	SELR	SELH	RWacc	RWrv	BLOK	
Reset	0	0	0	0	0	0	0	0	0	0	count = 0; countQ = 0;
Init	1	X	X	X	X	X	X	0	1	X	count = 0; countQ = 0;
MulDQxDRx	X	1	0	0	countQ	count-countQ	X	0	0	If countQ=count ? 1 : 0	If count <N ? WriteToAcc : Result
WriteToAcc	X	1	0	0	countQ	count-countQ	X	1	0	If countQ=count ? 1 : 0	If countQ = 0 ? MulAccH_RV : SetupCounter
SetupCounter	X	X	X	X	X	X	X	0	0	X	countQ-;
MulAccH_RV	X	0	1	1	X	X	count	0	0	X	
SaveToAcc	X	0	1	1	X	X	count	1	0	X	countQ++;
SaveToRv	0	0	X	X	X	X	X	0	1	X	count++;
Result	X	X	X	X	X	X	X	0	0	X	If en = 0 ? Result : Init

Tabulka 6.1: Kontrolér násobiaceho integrátora v FX

Stav	SELy0	SEL1	SEL2	SEL3	SELQ	SELR	SELH	RWacc	RWrv	BLOK	
Reset	0	0	0	0	0	0	0	0	0	0	
Init	1	X	X	X	X	X	X	0	1	X	count = 0; countQ = 0;
MulDQxDRx1 MulDQxDRx2	X	1	0	0	countQ	count-countQ	X	0	0	If countQ=count ? 1 : 0	
MulDQxDRx	X	1	0	0	countQ	count-countQ	X	0	0	If countQ=count ? 1 : 0	If count <8 ? WriteToAcc1 : Result
WriteToACC1	X	1	0	0	countQ	count-countQ	X	0	0	If countQ=count ? 1 : 0	
WriteToACC	X	1	0	0	countQ	count-countQ	X	1	0	If countQ=count ? 1 : 0	If countQ = 0 ? MulAccH_RV1 : SetupCounter
SetupCounter	X	X	X	X	X	X	X	0	0	X	countQ-;
MulACCh_RV1 MulACCh_RV2 MulACCh_RV	X	0	1	1	X	X	count	0	0	X	
SaveToAcc1	X	0	1	1	X	X	count	0	0	X	
SaveToAcc	X	0	1	1	X	X	count	1	0	X	countQ++;
SaveToRV	0	0	X	X	X	X	X	0	1	X	count++;
Result	X	X	X	X	X	X	X	X	X	X	If en = 0 ? Result : Init

Tabulka 6.2: Kontrolér násobiaceho integrátora v FP



Stav	SEL2	SEL3	SEL4	SEL5	SEL6	SEL7	SEL8	ClkAcc	ClkY	BLOK	
Reset	000	00	000	000	00	0	0	0	0	0	count = 1; const_int = 1;
Init	000	00	X	000	01	0	X	0	1	1	
MulUH	000	00	X	000	01	0	X	0	0	1	
WaitDiv	000	X	X	X	00	X	X	0	0	X	done = 1 ? SaveDiv : WaitDiv;
SaveDiv	000	X	X	001	00	1	0	0	0	0	
CalculationDY1	000	X	X	001	00	1	0	0	1	0	
WriteToAcc	001	X	X	const_int+1	const_int > 0 ? 10 : 00;	X	X	0	0	1	RWrv = 1; count < 7 ? SetupCounters : Result;
SetupCounters	001	X	X	X	const_int > 0 ? 10 : 00;	0	X	0	0	1	count = 1; const_int > 0 ? MulCountDYx : MulDYxDVx;
MulCountDYx	X	X	X	const_int	10	0	X	0	1	1	
SetupDYxDVx	count	X	X	const_int-count+1	00	0	1	0	0	count = 1 ? 1 : 0;	
MulDYxDVx	count	X	X	const_int-count+1	00	0	1	1	0	count < const_int ? SetupCount : SetupDUxH;	
SetupCount	X	X	X	X	X	X	X	0	0	X	count++;
SetupDUxH	X	const_int	X	000	01	0	1	0	0	0	XOR = 1;
MulDUxH	X	const_int	X	000	01	0	1	1	0	0	XOR = 1;
SetupACCDivX	X	X	const_int-1	X	11	1	X	0	0	1	
MulACCDivX	X	X	const_int-1	X	11	1	X	1	0	1	
SetupCalcDYx	000	X	X	const_int+1	00	1	0	0	0	0	
CalculationDYx	000	X	X	const_int+1	00	1	0	1	1	0	
Result	X		X	X	X	X	X	0	0	X	en = 0 ? Result : Init;

Tabulka 6.3: Kontrolér deliaceho integrátora v FX

Stav	SEL2	SEL3	SEL4	SEL5	SEL6	SEL7	SEL8	ClkAcc	ClkY	BLOK	
Reset	000	00	000	000	00	0	0	0	0	0	count = 1; const_int = 1;
Init	000	00	X	000	01	0	X	0	1	1	
MulUH	000	00	X	000	01	0	X	0	0	1	
MulUH1	000	00	X	000	01	0	X	0	0	1	
SaveDiv	000	X	X	001	00	1	0	0	0	0	ClkV = 1;
SaveAcc	000	X	X	001	00	1	0	1	0	0	
CalculationDY1	000	X	X	001	00	1	0	0	1	0	
CalculationDY2	00	X	X	001	00	1	0	0	0	0	
CalculationDY3	00	X	X	001	00	1	0	1	0	0	
WriteToAcc	001	X	X	const_int+1	const_int > 0 ? 10 : 00;	X	X	0	0	1	RWrV = 1; count < (N-1) ? SetupCounters : Result;
SetupCounters	001	X	X	X	const_int > 0 ? 10 : 00;	0	X	0	0	1	count = 1; const_int > 0 ? MulCountDYx : MulDYxDVx;
SetupCountDYx	X	X	X	const_int	10	0	X	0	0	1	
MulCountDYx	X	X	X	const_int	10	0	X	0	1	1	
SaveCountDYx	X	X	X	const_int	10	0	X	0	1	1	
SetupDYxDVx	X	X	X	const_int	10	0	X	0	1	1	
SetupDYxDVx1	count	X	X	const_int-count+1	00	0	1	0	0	count = 1 ? 1 : 0;	
MulDYxDVx	count	X	X	const_int-count+1	00	0	1	1	0	count < const_int ? SetupCount : SetupDUxH;	
SaveDYxDVx	X	X	X	X	X	X	X	0	0	X	count++;
SetupCount	X	X	X	X	X	X	X	0	0	X	
SetupDUxH	X	const_int	X	000	01	0	1	0	0	0	XOR = 1;
MulDUxH	X	const_int	X	000	01	0	1	0	0	0	
SumMulAcc	X	const_int	X	000	01	0	1	0	0	0	
SaveMulAcc	X	const_int	X	000	01	0	1	0	0	0	
SetupACCDivX	X	X	const_int-1	X	11	1	X	1	0	1	
MulACCDivX	X	X	const_int-1	X	11	1	X	0	0	1	
MulACCDivX1	X	X	const_int-1	X	11	1	X	0	0	1	
SumACCDivX	X	X	const_int-1	X	11	1	X	1	0	1	
SaveACCDivX	X	X	const_int-1	X	11	1	X	1	0	1	
SetupCalcDYx	000	X	X	const_int+1	00	1	0	0	0	0	
SetupCalcDYx1	000	X	X	const_int+1	00	1	0	1	1	0	
CalculationDYx	000	X	X	const_int+1	00	1	0	1	1	0	
Result	X	X	X	X	X	X	X	0	0	X	en = 0 ? Result : Init;

Tabulka 6.4: My caption

## 6.2 Integrátory v pohyblivej rádovej čiarke

Jednovstupový integrátor v pohyblivej rádovej čiarke bol implementovaný v 32 a v 64 bitovej verzii. Počet cyklov výpočtu nie je stanovený v implementácii integrátora ako to je u dvojestupových integrotoch, keďže integrátor vykonáva jednoduchú integráciu a pri výpočte nepotrebuje ukladať všetky predchádzajúce členy. Je ovládaný len pomocou signálov RESET a EN. Tie určujú či sa výpočet začína od začiatku alebo sa pokračuje výpočtom ďalšieho člena. Hodnoty kroku  $h$  a jeho podielov nie sú uložené v komponente integrátora, ale sú privádzané z nadradenej komponenty. Dôvodom je redukovanie duplicity registrov. Podrobnejšie vysvetlenie nájdeme v kapitole 6.3.

Stav	SELy0	SEL	SELh	RWrđ	RWrv	
Reset	1	0	0	1	1	
Init	X	X	count	0	0	
MulDYxH	X	X	count	0	0	
SetupToRD	0	1	count	0	0	
SaveToRD	0	1	count	1	0	count++
SaveToRV	X	X	count	0	1	If count < 8 ? Init : Result
Result	X	X	X	0	0	If en = 0 ? Result : Init

Tabuľka 6.5: Kontrolér jednovstupového integrátora v FP

Ďalej bol implementovaný násobiaci a deliaci integrátor v pohyblivej rádovej čiarke a taktiež vo verziách s 32 bitmi a 64 bitmi. Integrátory obsahujú paralelnú FP násobičku, paralelnú FP deličku a paralelnú FP sčítačku. Všetky komponenty boli prevzaté z xx a následne upravené a optimalizované pre daný výpočet. Komponenty vykonávajú operácie sčítanie, násobenie a delenie v troch hodinových taktach, keďže pracujeme v pohyblivej rádovej čiarke. Jednotlivé operácie boli popísané v kapitole xx. Výpočet je teda pomalší oproti variante pracujúcej s pevnou rádovou čiarkou, ale o to presnejší a zároveň sa veľmi zväčšil rozsah zobraziteľných hodnôt.

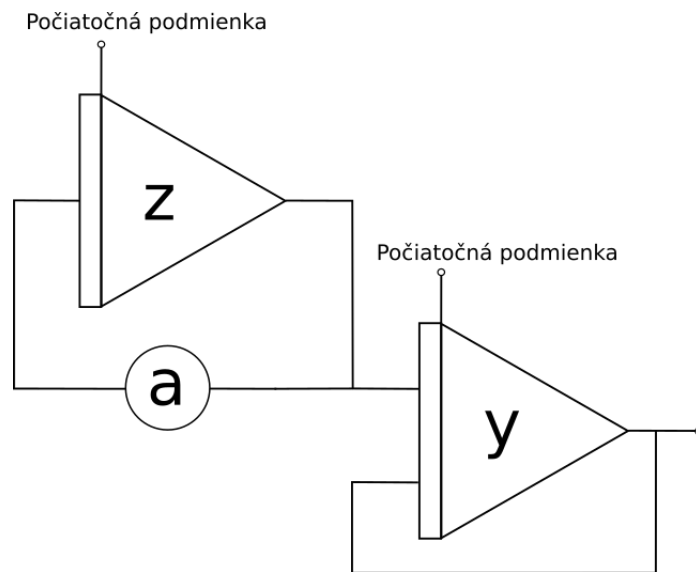
## 6.3 Sústava integrátorov

Podľa schémy zapojenia xx bola implementovaná komponenta počítajúca diferenciálnu rovnicu xx. Boli použité integrátory v pohyblivej rádovej čiarke so 64 bitmi - konkrétne jednovstupový integrátor a násobiaci integrátor. Ďalej bola použitá komponenta invertor. Keďže pracujeme vo FP, invertor vykonáva jednouché invertovanie znamienkového bitu. Každý integrátor je riadnený svojim vlastným kontrolérom, celé zapojenie je potom ovládané hlavným kontrolérom pomocou signálov RESET a EN. Na začiatku výpočtu sú nahrané počiatočné podmienky do integrátorov. Následne je pomocou povolovacieho signálu EN spustený výpočet na všetkých integrátoroch naraz. Všetky integrátory používajú pri výpočte hodnoty  $h$  a  $h/i$ . Pre optimalizáciu priestorovej zložitosti bola použitá iba jedna sada registrov  $h$ . Výber registru  $h$  je riadený pomocou signálu SELH. Jeho hodnota je určená podľa čítača cyklov. Hodnota čítača je inkrementovaná na konci každého cyklu. Pri začatí nového cyklu je táto hodnota použitá pre signál SELH. Keďže výpočet násobiaceho integ-

Stav	SELy0	SEL	RWrđ	RWrđ
Reset	1	0	1	0
Init	X	X	0	1
MulDYxH	X	X	0	0
SetupToRD	0	1	0	0
SaveToRD	0	1	1	0
SaveToRV	X	X	0	1
Result	X	X	0	0
	if ENDyx = 0 ? if en = 0 ? Result : MulDYxh : InitDy			

Tabuľka 6.6: Kontrolér jednovstupového integrátora v FP v zapojení do sústavy

rátora trvá dlhšie, integrátory q, r a s po dokončení výpočtu čakajú na násobiaci integrátor. Ten po dokončení výpočtu aktivuje signál OEN na 1. Kontrolér následne spustí výpočet každého integrátora odznova. Toto sa opakuje až kým výpočet jedného kroku metódy neskončí. Pri výpočte ďalšieho kroku výpočtu sú obnovené počiatočné podmienky a výpočet sa znova vykoná.



Obr. 6.1: Schéma zapojenia integrátorov

## Kapitola 7

# Analýza

- 7.1 Porovanie Taylorovej rady s metódou Runge-Kutta 2. radu
- 7.2 Porovanie Taylorovej rady s metódou Runge-Kutta 4. radu

## Kapitola 8

# Záver

V tejto práci sme sa zaoberali numerickou integráciou pomocou metódy Taylorovej rady. Úpravou jej členov sme získali potrebné rovnice, z ktorých boli vytvorené jednotlivé návrhy integrátorov. Rozšírili sme návrhy paralelných integrátorov s operáciou násobenia a delenia v pevnej rádovej čiarky tak, aby s nimi bolo možné počítať diferenciálne rovnice na 20 členov Taylorovej rady. Ďalej sme tieto integrátory navrhli aj v prevedení pohyblivej rádovej čiarky. Navrhli sme komponent slúžiaci na výpočet exponentu a na jeho úpravu pri normalizácii. Keďže výpočet v aritmetike pohyblivej rádovej čiarky je presnejší, integrátory využívajúce túto aritmetiku sú navrhnuté na riešenie diferenciálnych rovníc až na 40 členov Taylorovej rady.

Ďalším pokračovaním práce je popísanie navrhnutých integrátorov vo VHDL, otestovanie ich funkčnosti na VIRTEX5 a následné analyzovanie ich časovej a priestorovej zložitosti v porovnaní s metódou Runge-Kutta.

# Literatúra

- [1] Kraus, M.: *Paralelní výpočetní architektury založené na numerické integraci*. Disertační práce, FIT VUT v Brně, 2013.
- [2] Kunovský, J.: *Modern Taylor Series Method*. Habilitation work, VUT Brno, 1994.
- [3] Matečný, F.: *Simulátor procesora s operáciou delenia*. Bakalářská práce, FIT VUT v Brně, 2016.
- [4] Milan Kubíček, D. J., Miroslava Dubcová: *Numerické metody a algoritmy*. VŠCHT Praha, 2005, ISBN 80-7080-558-7.
- [5] Opálka, J.: *Automatické řízení výpočtu*. Bakalářská práce, FIT VUT v Brně, 2014.
- [6] Opálka, J.: *Automatické řízení výpočtu ve specializovaném výpočetním systému*. Diplomová práce, FIT VUT v Brně, 2016.
- [7] Sekanina, L.: Operace v FP a iterační algoritmy. , 2017, slajdy k predmetu INP - Návrh počítačových systémů.
- [8] Sekanina, L.: Reprezentace dat. , 2017, slajdy k predmetu INP - Návrh počítačových systémů.
- [9] Tišnovský, P.: Fixed point arithmetic [online].  
<http://www.root.cz/clanky/fixed-point-arithmetic/>, 2006-05-24 [cit. 2017-11-22].
- [10] Tišnovský, P.: Norma IEEE 754 a příbuzní: formáty plovoucí řádové tečky [online].  
<https://www.root.cz/clanky/norma-ieee-754-a-pribuzni-formaty-plovouci-radove-tecky/>, 2006-05-31 [cit. 2017-11-22].
- [11] Tišnovský, P.: Aritmetické operace s hodnotami ve formátu plovoucí řádové čárky [online].  
<https://www.root.cz/clanky/aritmeticke-operace-s-hodnotami-ve-formatu-plovouci-radove-cariky/>, 2006-06-07 [cit. 2018-01-07].
- [12] Závada, V.: *Simulátor procesoru s operací násobení*. Bakalářská práce, FIT VUT v Brně, 2016.
- [13] Čambor, M.: *Elementární procesor v aritmetice pevné a pohyblivé řádové čárky*. Bakalářská práce, FIT VUT v Brně, 2009.