

JEDNO REŠENJE MATEMATIČKE BIBLIOTEKE ZA PROCESORE ZA OBRADU DIGITALNIH SIGNALA

Dejan Mišković, Micronasnit, Novi Sad; Miroslav Popović, Fakultet tehničkih nauka, Novi Sad;
Dragan Kukulj, Fakultet tehničkih nauka, Novi Sad

Sadržaj - U ovom radu je prikazano jedno rešenje matematičke biblioteke u simboličkom mašinskom jeziku, za digitalne signal procesore (Digital Signal Processors - DSP). Biblioteka je uključena u prevodilac koji je namenjen za prevodenje programa pisanih u "C-like" jeziku. Ciljna platforma je procesor familije MAS35xx, firme Micronas, Freiburg, Nemačka, i namenjen je za obradu digitalnih signala. U biblioteci je podržana većina funkcija koju sadrži matematička biblioteka C programskog jezika, a dodate su i neke funkcije koje nisu u toj biblioteci. Jedna od takvih je funkcija za izračunavanje recipročne vrednosti, odnosno deljenja, koje je standardni operator u C-u. Uslov koji je trebalo zadovoljiti je da tačnost rezultata rada funkcija bude u opsegu koji je određen arhitekturom, ali je podržano i zadavanje tačnosti kao parametra. Algoritmi su napravljeni u aritmetici pokretnog zarez, koja nije podržana arhitekturom, već se emulira određenim funkcijama, koje su definisane od strane prevodioca. Biblioteka je ispitana u okviru prevodioca, a i zasebno, kao assemblerski kod.

1. UVOD

Algoritmi za digitalnu obradu signala zahtevaju obradu velikog obima, tako da je od posebnog značaja da budu efikasni. Postoji više načina za izvođenje izračunavanja, u smislu odabiranja vrste aritmetike. Celobrojna aritmetika je zastupljena u velikom broju algoritama, ali se pored nje koriste i aritmetika pokretnog i nepokretnog zarez. Algoritmi u aritmetici pokretnog zarez su brži i nije potrebno skaliranje kao kod celobrojne aritmetike, ali postoje neke arhitekture koje ne podržavaju aritmetiku pokretnog zarez.

Emulacija aritmetike pokretnog zarez se može izvesti korišćenjem para registara, jednog za mantisu i jednog za eksponent, uz definiciju funkcija koje podržavaju osnovne aritmetičke operacije. Da bi u potpunosti bila podržana aritmetika pokretnog zarez obično se stvara biblioteka matematičkih funkcija za tu aritmetiku. Pošto je u ovom slučaju reč o DSP arhitekturi, neophodno je da aplikacije koje se pišu za nju zadovoljavaju kriterijume u vezi brzine i zauzetosti memorije. Ako ova dva kriterijuma nisu zadovoljena, aplikacija neće biti u stanju da se izvršava na ciljnoj platformi. Zbog toga je potrebno posebno obratiti pažnju na efikasnost algoritma, odnosno broj linija koda i brzinu, tj. broj ciklusa potrebnih za izračunavanje određene vrednosti.

U matematičkoj biblioteci koja je opisana u ovom radu su za neke funkcije, koje zahtevaju više resursa nego ostale funkcije biblioteke, korišćeni izmenjeni standardni algoritmi numeričke matematik, da bi se zadovoljili kriterijumi koji se odnose na brzinu i veličinu koda.

2. OPIS CILJNE ARHITEKTURE

Procesor za obradu digitalnih signala, za koji je biblioteka namenjena, pripada familiji MASX 3500 firme Micronas, Freiburg, Nemačka. Procesor MAS je RISC "Harvard" arhitektura sa protočnom strukturom koja je otvorena za korisnika, podržava aritmetiku nepokretnog zarez, celobrojnu aritmetiku, i kompleksnu aritmetiku. Dužina reči ovog procesora je 20 bita, koliko iznosi i dužina instrukcijske reči. Arhitektura sadrži tri adresne i tri magistrale podataka, 4k reči RAM, 6k reči ROM, arhitekturom podržane petlje (hardware loops), ALU (Arithmetic Logic Unit) i MAC (Multiply And Accumulate) jedinice, i skup ulazno-izlaznih sprežnih podsistema dužine reči od 20 bita.

Broj akumulatora u ovoj arhitekturi je osam, i dužine su 32 bita, s tim da je pri operacijama podržano zadavanje tačnosti.

Memorija se sastoji od dve oblasti, pri čemu postoje četiri adresna generatora za svaku oblast kojima su pridruženi pokazivački registri, modulo registri, inkrement registri i registri bazne adrese. Pri upisivanju u memoriju, podržano je više režima pomoću kojih se opisuje način zaokruživanja. Pokazivači nemaju mogućnost izvođenja operacija adresne aritmetike u samoj instrukciji, već se to radi zadavanjem vrednosti inkrement registra. Zbog toga je potrebno ponekad koristiti akumulatore, što utiče na smanjenje efikasnosti.

Skup registara je takođe podeljen u dve oblasti, gde svaka oblast sadrži 128 registara. U obe oblasti registara postoje registri koji se koriste za posebne namene, kao i registri opšte namene, kojih ima trideset. Među registrima posebne namene su i registri za pristup ulazno-izlaznom sprežnom podsistemu.

Dužina reči ovog procesora određuje tačnost koja je potrebna pri izračunavanjima, i koja iznosi 10^{-6} . Pošto procesor ima i celobrojnu i aritmetiku nepokretnog zarez, pružaju se mogućnosti za korišćenje i jedne i druge aritmetike pri emulaciji aritmetike pokretnog zarez. Ta mogućnost je iskorišćena i rešenje je prikazano u ovom radu.

Važno je napomenuti da MAS arhitektura ima ograničenje pri operaciji pomeranja (shift), koja može biti u opsegu od pomeranja za 23 udesno i 24 ulevo. Ako se pređe ova granica, dobijaju se pogrešne vrednosti u rezultatu.

3. ARITMETIKA POKRETNOG ZAREZA

Broj u pokretnom zarezu je predstavljen pomoću mantise i eksponenta, sa različitim opsezima jednog i drugog, u zavisnosti od standarda. MAS arhitektura podržava sledeće opsege mantise i eksponenta :

- 20-bitni decimalni brojevi koji se koriste za eksponent su u opsegu -524288 do 524287, dok su 32-bitni brojevi istog tipa opsega -134217728.0000

- do 134217727.9375, pošto imaju 4 bita rezervisana za prekoračenje donje granice (*underflow*),
- za predstavu mantise u opsegu od vrednosti nula do 0.5 se koriste brojevi u nepokretnom zarezu, koji imaju tačku između devetnaestog i osamnaestog bita za dvadesetobitne, što znači da im je opseg između 1.0 i 0.999998; kod 32-bitnog brojeva, koji imaju tačku između 23. i 24. bita, je opseg između -256.000 i 255.9999999, pri čemu se 8 bita koristi za prekoračenje gornje granice (*overflow*).

Pošto je biblioteka deo programskog prevodioca, u mogućnosti je da koristi osnovne aritmetičke operacije, podržane od strane prevodioca.

4. PROGRAMSKI PREVODILAC ZA MAS

Programski prevodilac za *MAS* je deo okruženja za razvoj *C-like* programa, u koje spada i *IDE*, kao i komponenta za povezivanje (*linker*).

Da bi se održala konzistentnost aritmetike korišćene u biblioteci sa istom u prevodiocu, biblioteka koristi aritmetičke operacije koje se u prevodiocu odnose na tip realnih brojeva programskog jezika *C*. Međutim, *C-like* jezik predstavlja proširenje programskog jezika *C*, tako da korisnički tipovi imaju posebnu oznaku. U ove tipove se ubraja i tip realnih brojeva, iako je standardni tip, ali na *MAS* arhitekturi nema isti opseg kao tip realnih brojeva u *C* programskom jeziku.

Osim osnovnih aritmetičkih operacija, kao što su sabiranje, oduzimanje i množenje, u ovoj biblioteci su korišćeni i operatori koji predstavljaju operacije pretvaranja tipova (*casting*) koje su potrebne pri nekim od koraka u algoritmima.

Sve pomenute operacije i operatori su korišćeni tako što je napisana funkcija u *C-like* jeziku, prevedena na simbolički mašinski jezik, a tako dobijena funkcija je dalje korišćena u biblioteci kao kod koji je ubačen u funkcije (*inline*), a ne kao posebna funkcija, pošto kašnjenje pri skoku kod *MAS* arhitekture iznosi šest ciklusa.

5. SPISAK FUNKCIJA BIBLIOTEKE

U daljem tekstu je dat spisak svih funkcija matematičke biblioteke za *MAS* programski prevodilac :

- eksponencijalna,
- kvadratni koren,
- recipročna vrednost,
- hiperbolički tangens,
- hiperbolički sinus,
- hiperbolički kosinus,
- sinus,
- kosinus,
- tangens,
- arc sinus,
- arc kosinus,
- arc tangens,
- arc tangens 2,
- prirodni logaritam,
- stepen,
- logaritam za osnovu 10,
- zaokruživanje na višu decimalu (*ceil*),

- zaokruživanje na nižu decimalu (*floor*),
- modf,
- fmod.

Algoritmi za neke od funkcija su dati u narednim poglavljima.

5.1. EKSPONENCIJALNA FUNKCIJA

Ova funkcija se koristi i u algoritmima za hiperboličke funkcije, tako da je potrebno da bude napravljena što efikasnije. Algoritam korišćen za ovu funkciju je jedan od najčešće primenjenih algoritama za ovu funkciju u računarskoj tehnici, i koristi *Taylor*-ov, odnosno *MacLaurin*-ov red. Iteracije su date na sledeći način :

$$S_k = S_{k-1} + U_k, \quad U_k = \frac{U_{k-1}}{k} * X, \quad k = 0, 1, 2, \dots$$

$$S_{-1} = 0, \quad S_0 = 1, \quad U_0 = 1,$$

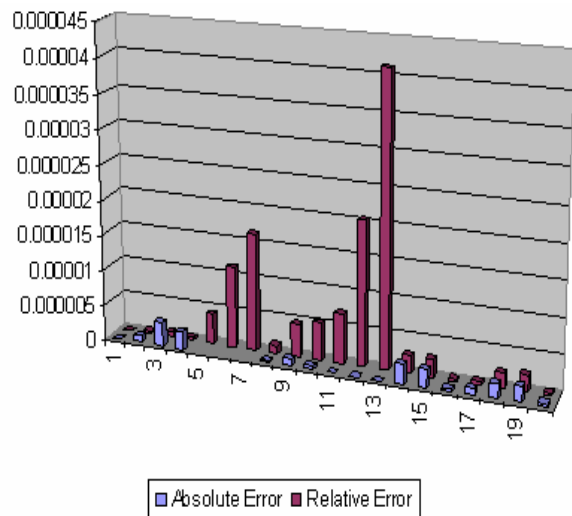
gde je S_k tekuća suma, a U_k element sume za k -ti stepen iteracija. Izračunavanje se prekida u iteraciji koja zadovoljava zadatu tačnost.

Pošto *MAS* arhitektura ima ograničenja pri operaciji pomeranja, uvedeni su dodatni koraci u algoritmu izračunavanja. Zbog toga, ako je eksponencijalna funkcija data jednačinom :

$$y = e^x,$$

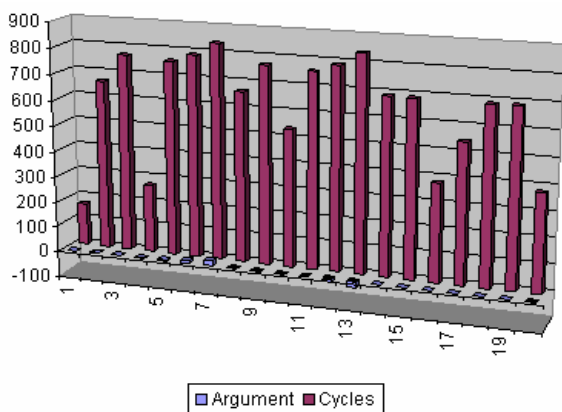
tada je ulazni argument X smanjen, tako da njegova apsolutna vrednost ne može biti veća od dva, pa se nakon izvršenih iteracija radi množenje da bi se dobio konačan rezultat. Ova izmena se radi ako su u pitanju velike vrednosti ulaznog argumenta, dok se za male vrednosti argumenta oduzima argument od vrednosti jedan, što se uzima u obzir pri izračunavanju konačnog rezultata.

Histogram odnosa apsolutne i relativne greške je dat na slici 1, pri čemu vrednosti apscise predstavljaju redne brojeve argumenta :



Slika 1. Apsolutna i relativna greška

Na slici 2 je dat histogram odnosa dužine izvršavanja i ulaznog argumenta, pri čemu apscisa ima isto značenje kao i na prethodnoj slici :



Slika 2. Broj ciklusa pri izvršavanju i vrednost argumenta

5.2. RECIPROČNA VREDNOST

Funkcija je prvo implementirana koristeći *Newton-Rhapson*-ovu metodu, ali se pokazalo da je broj iteracija veliki, a s obzirom da funkcija pripada, odnosno koristi se u osnovnoj aritmetičkoj operaciji deljenja, važno je da njego izvršavanje bude što kraće i da veličina koda bude što manja. Zato je primenjen metod koji koristi *Taylor*-ov, odnosno *MacLaurin*-ov red, sa izmenama koje uključuju korišćenje tabele od 50 vrednosti. Iteracije su date jednačinama :

$$X = 1 + Y$$

$$Y = X - 1$$

$$\frac{1}{1+Y} = (1-Y) * (1+Y^2) * (1+Y^4) * (1+Y^8) \dots$$

gde je X ulazni argument.

Da bi se dobio što manji broj iteracija, određena je vrednost za koju je broj iteracija najmanji. Korišćenjem tabele se skalira, tako da se dobija najmanji broj iteracija, a skaliranje za vraćanje vrednosti u pravi opseg se izvodi i na kraju iteracija, u suprotnom smeru. Tabela se sastoji od 50 elemenata u opsegu 0.5 do 0.99999.

Jedan od posebnih koraka je pristup elementima tabele, koji je izvršen množenjem i oduzimanjem mantise.

Takođe je važno napomenuti da se iteracije izvode samo nad mantisom, a eksponent se uključuje na kraju iteracija, čime se dobija na efikasnosti algoritma.

5.3. ARC SINUS (arcsin)

Funkcija je, kao i prethodno opisane funkcije, načinjena koristeći *MacLaurin*-ov red, a jednačine su u ovom slučaju date sa :

$$\arcsin(X) = X + \frac{X^3}{6} + \frac{3 * X^5}{40} + \frac{15 * X^7}{336} + \dots$$

gde je X ulazni argument. Primećuje se da jednačina sadrži koeficijente, pa je zato potrebno imati tabelu.

U opsegu brojeva od vrednosti nula do 0.5 je moguće koristiti prehodnu jednačinu, dok je za opseg 0.5 do 1 broj iteracija neprihvatljiv. Kao posledica toga se u opsegu 0.5 do 1 koristi sledeća jednačina, koja u sebi sadrži funkciju kvadratnog korena :

gde je X ulazni argument.

$$\arcsin(X) = \frac{\pi}{2} - 2 * \arcsin \sqrt{\frac{1-X}{2}}$$

Pošto se koristi funkcija kvadratnog korena, arcsin će imati istu tačnost kao ta funkcija za opseg vrednosti 0.5 do 1.

5.4. ARC TANGENS (arctg)

I ova funkcija koristi razvoj u *MacLaurin*-ov red za izračunavanje rezultata, i data je jednačinom :

$$\left[0, \frac{7}{16}\right] \quad a \tan(X) = t - t^3 * (a_1 + t^2 * (a_2 + \dots (a_{10} + t^2 * a_{11}) \dots))$$

$$\left[\frac{7}{16}, \frac{11}{16}\right] \quad a \tan(X) = a \tan\left(\frac{1}{2}\right) + a \tan\left(\frac{t-0.5}{1+t*0.5}\right)$$

$$\left[\frac{11}{16}, \frac{19}{16}\right] \quad a \tan(X) = a \tan(1) + a \tan\left(\frac{t-1}{1+t}\right)$$

$$\left[\frac{19}{16}, \frac{39}{16}\right] \quad a \tan(X) = a \tan\left(\frac{3}{2}\right) + a \tan\left(\frac{t-1.5}{1+t*1.5}\right)$$

$$\left[\frac{39}{16}, \text{inf}\right] \quad a \tan(X) = a \tan(\text{inf}) + a \tan\left(-\frac{1}{t}\right)$$

gde je X ulazni argument.

Pošto jednačina sadrži koeficijente, sledi da je potrebno koristiti tabelu, a u ovom slučaju to je tabela koja ima osam elemenata. Tabela sadrži osam elemenata, jer je to broj elemenata potreban za dostizanje tačnosti koju zahteva *MAS* arhitektura.

5.5. PRIRODNI LOGARITAM (ln)

Vrednosti ove funkcije se izračunavaju koristeći razvoj u red, dat sledećim jednačinama :

$$\ln(X) = m * \ln(2) - 2 * (U_1 + U_2 + \dots + U_n),$$

$$U_k = \frac{\zeta^{2*k-1}}{2*k-1}, \quad k = 1, 2, 3, \dots$$

$$\zeta = \frac{1-z}{1+z}$$

$$X = 2^m * z$$

$$U_n \leq \text{eps}$$

gde je X ulazni argument, U_k je tekuća vrednost zbira elemenata u k -tom koraku, z je mantisa argumenta, m je eksponent argumenta, a *eps* predstavlja tačnost.

6. ZAKLJUČAK

U ovom radu je opisano jedno rešenje matematičke biblioteke za procesore za obradu digitalnih signala. Rešenje je dobijeno u skladu sa ograničenjima ciljne arhitekture, tako da su izbegnuti izuzeci (*exceptions*), uz ispunjenje zahteva koji se odnose na tačnost i pouzdanost.

Dalja poboljšanja bi se mogla odnositi na usavršavanje algoritama u cilju što efikasnijeg

izvršavanja, pri čemu bi predloženo rešenje moglo da posluži za upoređivanje rezultata.

LITERATURA

- [1] Vladimir Kovačević: *Logičko projektovanje računarskih sistema*, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 1993
- [2] B.Kerningham, D.Ritchie: *Programski jezik C*, Savremena administracija, Beograd, 1989
- [3] *MASC development tools*, Micronas Intermetall, 1999
- [4] Vladimir Kovačević: *Programska podrška računarskih sistema*, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, 1989

Abstract – *In this paper a solution of mathematic library for digital signal processors is presented. This library is a part of C-like compiler for DSP MAS35XX processor, made by Micronas, Germany. In the library most of standard functions from C programming language are supported.*

ONE SOLUTION OF MATHEMATIC LIBRARY FOR DIGITAL SIGNAL PROCESORS

Dejan Mišković, Miroslav Popovic, Dragan Kukolj