

Техничко решење: Софтвер за симулацију мерних система променљиве прецизности

Руководилац пројекта: Владимир Вујичић

Одговорно лице: Владимир Вујичић

Аутори: Мирјана Тробок, Сенка Бајчета, Владимир Вујичић, Платон Совиљ, Бојан Вујичић, Драган Пејић

Развијено: у оквиру пројекта технолошког развоја ТР-32019 и покрајинског пројекта 114-451-2723.

Година: 2011.

Примена: 10.12.2011.

Кратак опис

Развијен је софтвер за симулацију мерних система променљиве прецизности. Трансформација произвољног нумеричког алгоритма за рад у реалном времену у алгоритам са променљивом прецизношћу посебно је значајна због могућности да се такав алгоритам прилагоди резолуцији процесора у циљу повећања брзине рада применом оптималне дужине речи. У последње време приметан је пораст имплементације рачунских алгоритама за рад у реалном времену у програмабилни хардвер (ASIC, FPGA и сл.), где су могуће веће брзине рада. У овим применама метода за трансформацију алгоритма произвољне прецизности у алгоритам са захтеваном прецизношћу доводи до оптимизације величине примењеног хардвера и на крајњу цену уређаја. Софтвер за симулацију мерних система произвољне прецизности је од велике користи за пројектовање оваквог мерног хардвера са уграђеним алгоритмима.

Реализатори:

Факултет техничких наука у Новом Саду

Корисници:

ФТН као произвођач. Могућ је пренос технологије према свим заинтересованим фирмама.

Подтип решења:

Нова метода (М 85)

Стање у свету

Идеја за реализацију алгоритама са променљивом прецизношћу јавила се са појавом сложених алгоритама и великог рачунарског времена потребног да се такви алгоритми изврше. Тада је смањење времена извршавања алгоритма било могуће или упрошћавањем алгоритма, што има своје границе, или применом нумеричких метода за трансформацију математичког алгоритма у рачунарски оптимизован алгоритам који не умањује задату прецизност рада, а омогућава брже извршење на конкретном рачунару коришћењем и познавања предности и ограничења рачунара на коме се алгоритам извршава. Проблем тог времена био је и недостатак стандардизације нумеричких података, па је било потребно прилагођавање алгоритма сваком појединачном рачунару.

Данас је ситуација другачија: начин представљања нумеричких података у рачунарима је стандардизован. Савремени програмски језици користе и операције над битима, тако да се на први поглед стиче утисак да није више потребно прилагођавање алгоритама рачунарима. С друге стране, у развоју модерне инструментације важан простор остављен је за блокове електронског мерења и обраде сигнала. Електронски мерни блок састоји се од А/Д

конвертора (једног или више), а блок за обраду сигнала садржи и један или више процесора. Оба ова блока се пројектују према захтевима мерења – брзине рада, прецизности и тачности (мерне несигурности). У оба случаја постоји широк избор понуђених компоненти. Софтвер за симулацију мерних система променљиве прецизности помаже пројектанту да изабере оптималне компоненте.

Проблем оптималног избора

Данас постоји велика количина софтвера који је написан у различитим програмским језицима и мноштво посебно развијаних хардверских и софтверских платформи. Због тога постоји и велика могућност избора хардвера и софтвера, па се избор оптималне платформе намеће као озбиљан инжењерски и практични проблем.

При развоју инструмента треба предузети следеће кораке:

- избор алгорита (или у општем случају софтвера) који реализује задате функције инструмента. Софтвер је најчешће писан у неком од виших програмских језика.
- овај алгоритам се трансформише у алгоритам са променљивом прецизношћу.

Улазна резолуција података (резолуција А/Д конвертора), резолуција процесорске јединице (INT – целобројна, FXP – са непокретним зарезом или FP - са покретним зарезом) се варира у задатим границама, како би се постигле захтеви прецизности и тачности:

- Д/А конвертора
- А/Д конвертора
- INT, FXP или FP аритметика и одговарајућа прецизност се бирају тако да се не ослабе захтеви тачности и прецизности.

Кад се изабере оптималне резолуције Д/А и А/Д конвертора и INT, FXP или FP аритметика, дефинише се хардвер који задовољава све задате услове.

Проблем може да се реши програмом написаним у MATLAB-у или у C –у коришћењем одговарајућих алата.

Процедура трансформације

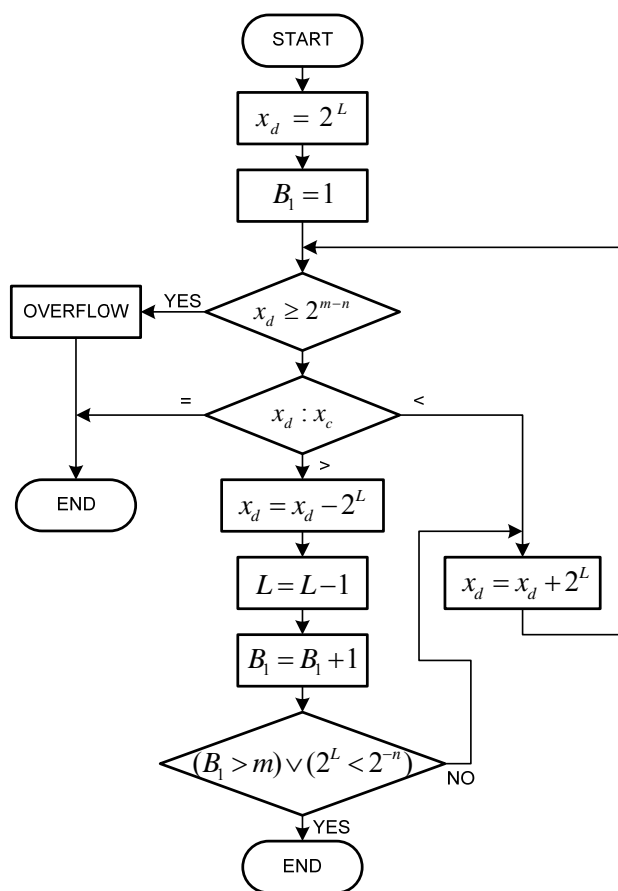
Метода која се користи припада класи симулационих метода. Симулација се извршава на рачунару за рад у покретном зарезу са дугачком мантисом репрезентације података. Горња граница релативне грешке таквих рачунара је:

$$\Gamma_{\max} = \frac{|\Delta x_{\max}|}{x_{\min}} = 2^{1-t}$$

где је t број цифара у мантиси. Кад је $t \geq 32$, подаци могу да се сматрају континуалним у већини случајева.

Проблем пре-детерминисања грешке квантизације је нелинеаран и егзактна математичка анализа је веома компликована, чак и у случају синусоидалног улазног напона и униформног А/Д конвертора. Због тога је једина применљива метода симулација.

Проблем је како симулирати мерни процес у датом инструменту или у мерном систему са истим улазом и избором А/Д и Д/А конвертора и дужине речи нумеричког процесора. Главну улогу у решењу проблема даје алгоритам:



Нека је улазна мерна величина означена са x_c као реални податак представљен са помичним зарезом. Тада, коришћењем горњег алгоритма, φ_d са специфицираним m и $n=1$, x_c се редукује до x_d , податка са помичним зарезом са $m-1$ целобројних цифара. Цифра на 2^{-1} месту користи се за заокруживање. На овај начин алгоритам $\varphi_d(x_c) = x_d$ има улогу симулационог алгоритма за $m-1$ цифарски А/Д конвертор. Такође, ако је потребно да се излазни подаци дају као физичке вредности, алгоритам φ_d може да се користи и као $m-1$ цифарски Д/А конвертор.

Нека су x_1 и x_2 два реална броја са највише m цифара у мантиси, у домену $(a^{m-n}, a^{-n}]$. Нека је \otimes било који од аритметичких оператора у непомићном зарезу, и нека је \odot било који од одговарајућих аритметичких оператора у помичном зарезу. Лако се види да је:

$$x_1 \otimes x_2 = \varphi_d(x_1 \odot x_2)$$

Примена ове једнакости илустрована је следећим примером:

Пример: Ако су a, b, c и d вредности као x_1 или x_2 и ако је z израчунат коришћењем

$$z = ((a \times b) / c) - d$$

тада је z_d дат са:

$$z_d = \varphi_d(\varphi_d(\varphi_d(a \times b) / c) - d)$$

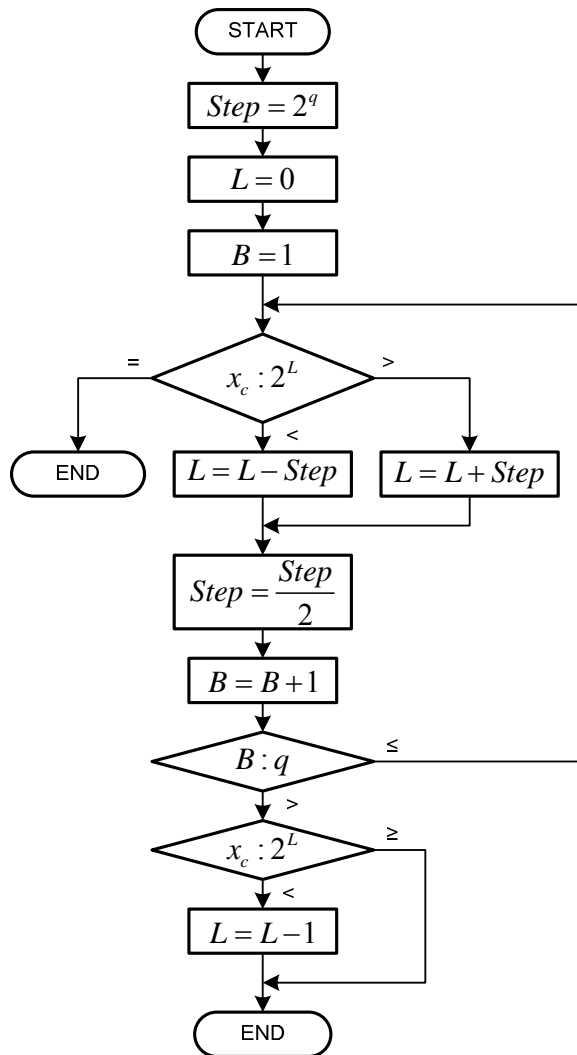
z_d је променљиве прецизности у непомићном зарезу са укупно m цифара, ($1 \leq m \leq t$), и са n цифара после децималне тачке. Ово је основа за развој мерних алгоритама променљиве прецизности.

Посматрајмо уграђене функције, као $\sin(x)$, $\cos(x)$ и сличне. Аргументи ових функција могу да се симулирају на сличан начин, тако да x буде у облику са непокретним зарезом. Једноставно,

$$\sin_d(x) = \varphi_d(\sin(x))$$

и добијамо симулирану уграђену функцију $\sin_d(x)$.

Неопходно је да се размотри и почетна вредност x_d (тј. 2^L), у алгоритму φ_d : сваки x_c и одговарајући c_d припадају истом бројном интервалу $[2^L, 2^{L+1})$. Могуће је конструисати ефикасан алгоритам за претрагу који ће пронаћи доњу границу опсега (тј. 2^L) за свако x_c у том опсегу бројева. Алгоритам је приказан на следећој слици:



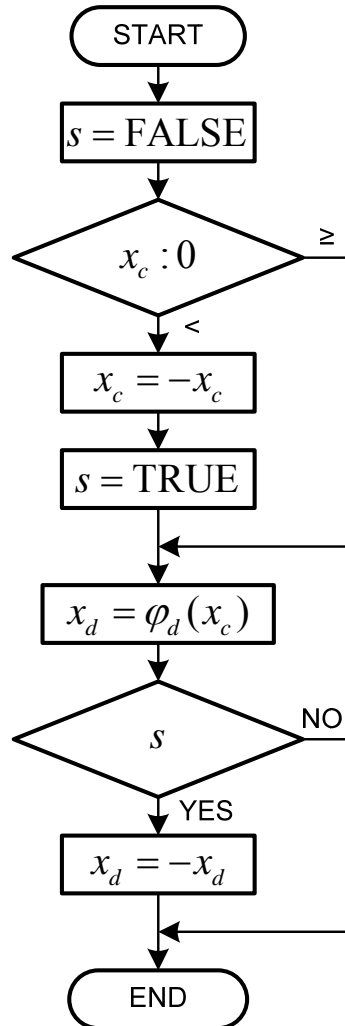
Нека је изабран скуп експонената $\{-2^q, -2^q + 1, \dots, -1, 0, 1, \dots, 2^q - 1\}$. Тада сваки експонент одређеног броја x_c , L , припада том скупу, тј. $L \in \{-2^q, -2^q + 1, \dots, -1, 0, 1, \dots, 2^q - 1\}$

$$\log_2 |x_{MaxExp}| \geq q$$

где је $|MaxExp|$ максимална апсолутна вредност експонента на рачунару на коме се алгоритам извршава. Излазна вредност алгоритма је L . Оно дефинише доњу границу бројног интервала $[2^L, 2^{L+1})$ коме припада x_c :

$$x_c \in [2^L, 2^{L+1})$$

У алгоритму за конверзију φ_d , подразумева се да је број x_c позитиван. Ако је x_c негативан, треба применити алгоритам који је приказан на следећој слици:



Тада φ_d постаје :

$$z_d = \psi_d(\psi_d(\psi_d(a \times b) / c) - d)$$

ψ_d увек обезбеђује не-негативан аргумент за φ_d .

Резултати – мерење 64 хармоника у дистрибутивној мрежи

Према Европској норми EN50160, која дефинише квалитет напона електричне дистрибутивне, најсложенији параметар који се мери је хармонијско изобличење (THD). Према овој норми фактор хармонијског изобличења мери се у периоду не дужем од 10 min. Јака конкуренција на тржишту довела је до тога да неки произвођачи праве уређаје који мере овај параметар три пута у секунди.

Применом софтвера за симулацију мерних система променљиве прецизности на овај случај дефинишемо кораке:

1. корак: Изабран је алгоритам Radix 2 за рачунање брзе Фуријеова трансформације (FFT), писан у програмском језику DELPHI, који се користи више година и који је проверен и ради поуздано.

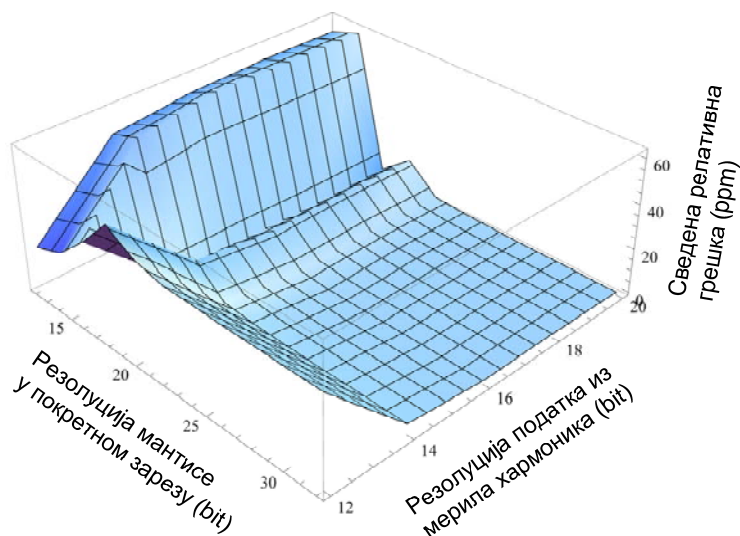
2. корак: Применом изложене методе овај алгоритам је конвертован у алгоритам са променљивом прецизношћу.

3. корак: Симулирани су случајеви $\psi_{AD} = \psi_d(m, 1)$ и $\psi_{FP} = \psi_d(m, m)$. Очигледно је да они представљају посебне случајеве оператора који су приказани алгоритмима који су описани у овом техничком решењу.

4. корак: Бира се улазни тригонометријски полином за тестирање, у овом случају:

$$y(t) = 10 + \sum_{i=1}^{63} [(64 - i) \cos i\omega t + (64 - i) \sin i\omega t]$$

На следећој слици графиком је приказан утицај резолуције A/D конвертора, $m_{AD} \in \{12, 14, 16\}$, и процесора у покретном зарезу, $m_{FP} \in \{12, 14, 16, 18, 20, 22, 24, 26, 28\}$, на модуо средње квадратне грешке за појединачне хармонике.



У основи, треба решити инверзни проблем: узимајући у обзир високе захтеве у погледу брзине и тачности које треба испунити, треба да проверимо да ли нека од широко распрострањених хардверско-софтверских платформи задовољава услове.

5. корак: Дискусија.

За процесор са оптималном резолуцијом (CELL процесор) од $m_{FP} = 24$, горња слика показује да је за FFT у 256 тачака (двоструко већа учестаност одмеравања, 64 хармоника), оптимална резолуција процесора не угрожава тачност мерења било којег од три симулирана А/Д конвертора. Показано је да је довољно да резолуција примењеног процесора у покретном зарезу буде 4 бита виша од резолуције примењеног А/Д конвертора да процесор не утиче на тачност мерења хармоника. Ово је веома важан резултат.

Софтвер за симулацију мерних система променљиве прецизности развијен је на Факултету техничких наука у Новом Саду, у оквиру текућег пројекта бр. ТР-32019 код Министарства за науку и технолошки развој Републике Србије и и покрајинског пројекта 114-451-2723.

Штампано – Децембар 2011.