

КОНФІГУРОВАНА ОБЧИСЛЮВАЛЬНА СИСТЕМА ДЛЯ ВИРІШЕННЯ ЗАДАЧ ЛІНІЙНОЇ АЛГЕБРИ.

Задачі лінійної алгебри все частіше зустрічаються в сучасній цифровій обробці сигналів, наприклад, при розпізнаванні образів, реалізації метода найменших квадратів. Для свого вирішення ці задачі потребують підвищеної точності обчислень. Тому вони, як правило, вирішуються на відносно дорогих мікропроцесорах з плаваючою комою.

Розвиток ПЛІС привів до появи кристалів з великою потенційною обчислювальною потужністю. Так, ПЛІС Xilinx Virtex2 має у своєму складі десятки апаратних блоків множення і блоків пам'яті, причому на один блок множення 18x18 розрядів припадає близько 200 еквівалентних конфігурованих обчислювальних блоків (ЕКЛБ). Але традиційна методика розробки проектів для ПЛІС оперує тільки з цілочисельною арифметикою, або з числами з фіксованою комою і не підтримує високоточні обчислення. А обчислювальні модулі для реалізації в ПЛІС операцій с плаваючою комою мають занадто великі затримки і апаратні витрати. Тому для вирішення задач лінійної алгебри в ПЛІС необхідний пошук нових способів обчислень і алгоритмів, які адаптовані до цього обчислювального середовища.

При вирішенні вказаних задач багатократно використовується операція ділення, яка реалізується з великими апаратними витратами і затримкою, і є основним джерелом похибок обчислень. Існують алгоритми лінійної алгебри, що основані на операції повороту комплексного вектора. В цих алгоритмах, як правило, не використовуються такі операції, як ділення, тому вони успішно виконуються в арифметиці цілих чисел. Так, в [1] описаний приклад реалізації в ПЛІС алгоритма QR- розкладання матриць методом Гівенса, що оснований на поворотах. Але при реалізації цих алгоритмів в нових ПЛІС їхні ресурси блоків множення використовуються нераціонально.

Для мінімізації похибок, затримок і апаратних витрат на операцію ділення пропонується нетрадиційне представлення даних у вигляді дробів. Таке представлення приведено, наприклад, в [2] для цілочисельного вирішення задач з поліномами. При цьому число x представляється двома цілими числами, перше з яких є чисельником n_x , а друге - знаменником d_x дробу, тобто $x = n_x/d_x$. Усі обчислення на перших $N-1$ ітераціях, наприклад, вирішення системи рівнянь, пропонується виконувати з даними у вигляді таких дробів. При цьому множення, ділення і додавання виглядає наступним чином:

$$x^*y = n_x n_y / (d_x d_y); \quad x/y = n_x d_y / (d_x n_y); \quad x+y = (n_x d_y + n_y d_x) / (d_x d_y).$$

У кінці останньої ітерації чисельники дробів діляться на знаменники звичайним діленням з одержанням результатів алгоритму. Таке представлення даних забезпечує як малі похибки обчислень, так і розширений динамічний діапазон в порівнянні з арифметикою цілих чисел, а також простоту реалізації в ПЛІС і високу швидкодію в порівнянні з пристроями з плаваючою комою. При досить великій розрядності дробів (більше 32) слід чекати, що точність обчислень перевищуватиме точність обчислень з плаваючою комою.

При розробці обчислювальної системи (ОС) для вирішення задач лінійної алгебри за базовий був прийнятий алгоритм обернення теплицевої матриці, що описаний в [3]. Перед відображенням в паралельну ОС, алгоритм обчислення матриці, оберненої до теплицевої, було описано і досліджено у вигляді поведінкової моделі на VHDL. При цьому був розроблений пакет `Fract_Lib`, який включає тип `FRACTV`, який представляє дріб з чисельника і знаменника і функції додавання, віднімання, множення і ділення над цим типом, функцію `FRACT_REAL`, що перетворює дріб в число з плаваючою комою а також константи `nil=0/1` і `one=1/1`.

Теплицева матриця A загального вигляду задається двома векторами:

```
type ARRv is array(natural range <>)of FRACTV;
signal Af:ARRv(0 to nn);
signal Bf:ARRv(0 to nn);
```

де nn - порядок матриці, Af - перша строка, Bf -остання строка матриці A .

Алгоритм обчислення оберненої матриці полягає в обчисленні її першого Xf і останнього Yf стовпця, так як решту стовпців нескладно обчислити через ці стовпці. Масиви проміжних результатів були объявлені як:

```
signal Ff,Gf,Rf,Sf,Tf:arrv(0 to nn):=(others=>nil);
```

Алгоритм був записаний як наступний процес:

```
FRACT_INV: process(clk,rst)
    variable f,g,r,s,t:FRACTV ;
begin
    if rst='1' then
        Xf(0)<=one/Af(0); Xf(0)<=one/Af(0);
    elsif rising_edge(clk)then
        f:=nil; g:=nil;
        for i in 0 to k-1 loop
            f:=f+Af(k-i)*Xf(i);
            g:=g+Bf(i+1)*Yf(i);
        end loop;
        r:=one/(one-f*g); s:=-r*f; t:=-r*g;
        Xf(0)<=Xf(0)*r;
        Yf(0)<=Xf(0)*t;
        Xf(k)<=Yf(k-1)*s;
        Yf(k)<=Yf(k-1)*r;
        for i in 1 to k-1 loop
            Xf(i)<=Xf(i)*r+Yf(i-1)*s;
            Yf(i)<=Xf(i)*t+Yf(i-1)*r;
        end loop;
```

```

        Ff<=f;Gf<=g;Rf<=r;Sf<=s;Tf<=t;
        k<=k+1;
    end if;
end process;

```

Процес обчислення починається по сигналу `rst` і закінчується через `nn` тактів синхросерії `clk`.

Для обчислення задач з теплицевими матрицями широко розповсюджені алгоритми Левінсона і Дарбіна [3]. Але ці алгоритми відрізняються нерегулярністю своїх графів залежностей і тому вони погано розпаралелюються. Приведений алгоритм відрізняється регулярністю і тому він добре відображається на структуру ОС у вигляді лінійки процесорних елементів (ПЕ), в якій ПЕ взаємодіють локально. Після відображення k -ті елементи векторів $Af, Bf, Ff, Gf, Rf, Sf, Tf, Xf$ і Yf знаходяться в k -му ПЕ.

Арифметика дробів була використана в експериментальній паралельній ОС для вирішення задач з теплицевими матрицями. Паралельна ОС має кільцеву структуру, яка може бути змінена на будь-яку структуру, завдяки реконфігурації ПЛІС. ПЕ взаємодіють між собою через буфери даних і механізм переривань з високою швидкістю.

За основу процесорного елемента паралельної ОС було взяте ядро мікропроцесора RISC_ST, що описане в [4]. Це ядро складається з базового ядра і апаратного прискорювача. Апаратний прискорювач реалізує розширення команд з арифметикою дробів та інших операцій, що часто використовуються при цифровій обробці сигналів. При цьому виконання множення і ділення дробів триває один такт, а їхнє додавання – два такта. Перед записом в пам'ять чисельник і знаменник результату нормалізуються шляхом зсуву на 0, 1 або 2 розряди вліво. Загальний об'єм обладнання одного ПЕ при реалізації даного алгоритму складає 380 ЕКЛБ, 2 блока множення і 2 блока ОЗП, а тактова частота досягає 80 МГц при конфігуруванні в ПЛІС Xilinx Virtex2.

Методика розробки застосування для паралельної ОС складається з трьох етапів. На першому етапі розробляються на мові асемблера і транслюються програми для ПЕ за допомогою асемблера процесора RISC_ST, що зкоригований для трансляції додаткових команд. Одержані програми відлагоджуються і тестуються на моделі ОС за допомогою системи Active HDL. При трансляції невикористані коди команд ПЕ і адреси даних і переривань фіксуються асемблером.

На другому етапі генерується опис ПЕ і паралельної ОС в цілому на VHDL, в якому відсутні невикористані вузли і елементи пам'яті моделі ПЕ. Завдяки цьому одержуються мінімізовані апаратні витрати кожного ПЕ. Кожна зтрансльована програма вкладається в ОЗП програми відповідного ПЕ. На третьому етапі всі моделі ПЕ підключаються до VHDL – проекту, який транслюється в файл конфігурації цільової мікросхеми ПЛІС.

Паралельна ОС з $N=10$ ПЕ відображається в ПЛІС Xilinx XCV2-1000. За допомогою цієї ОС обернена матриця розмірності $N=10$ (точніше – її перший і останній стовбці) знаходиться за 1.23 мкс., не рахуючи ввід-вивід

начальних даних і результатів. Таким чином, середня продуктивність ОС досягає 172 млн. операцій в сек., таких як складання, множення і ділення, які еквівалентні операціям з плаваючою комою.

Для приймання рішення про доцільність використання розробленої ОС була виконана оцінка похибок обчислень в ній. На рис. показані графіки залежності похибки результатів обчислень від розмірності матриці N і від розрядності даних n .

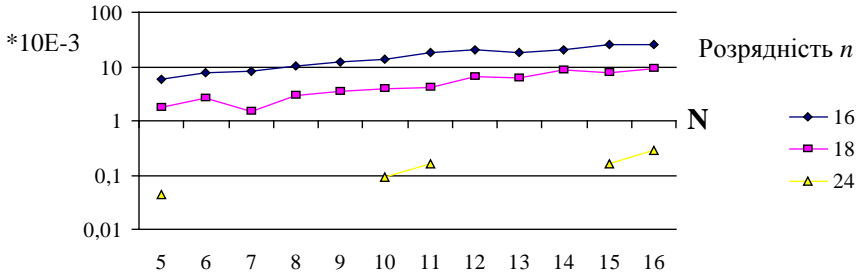


Рис. Залежність похибки оберненої матриці від розрядності даних і порядку матриці

Кожне дане для графіків одержано при оберненні 500 випадкових не вироджених теплицевих матриць і осередненні похибок результатів. Завдяки підвищенні до $n=18$ розрядності блоків множення ПЛІС, одержується похибка обчислень близько 0,5%, що майже в 4 рази менше, ніж при реалізації цього алгоритму в 16-розрядному сигнальному мікропроцесорі. Такої точності досить для вирішення багатьох задач цифрової обробки сигналів, таких як адаптивна фільтрація, лінійне передбачення і т.і.

Застосовуючи запропоновану методику розробки проекту для конфігурованої паралельної ОС можна швидко розробити конфігурацію для ПЛІС і досягти балансу апаратних витрат в ній, так що на один ПЕ припадатиме близько 200-500 ЕКЛБ. Так, ОС, що складається з 80 таких ПЕ, може бути зконфігурована в ПЛІС Xilinx XCV2-8000 і буде мати середню продуктивність до 2000 млн.оп. в сек. при реалізації задач лінійної алгебри.

1. Виноградов Ю.Н., Шевченко Ю.А., Сергиенко А.М. Конфигурируемые компьютеры: состояние и перспективы. *Материалы международной научно-технической конференции "International Active-VHDL Conference" (16 окт. 2001)*, -Харьков: -2001. -с. 5-9.
2. Кнут Д. Искусство программирования. Т.2. -М.:Мир. -1979. -556с.
3. Воеводин В.В., Тыртышников Е.Е. Вычислительные процессы с теплицевыми матрицами. - М.: Наука. -1987.- 320 с.
4. Сергиенко А.М. VHDL для проектирования вычислительных устройств. - Киев: DiaSoft. -2003. - 208 с.