Michał Trzebiatowski

Zastosowanie procesorów sygnałowych – projekt
Sprawozdanie

Celem projektu było pokazanie wpływu zaokrąglenia obliczeń na dokładność wygenerowania funkcji sinus. Obliczenie wartości sinusa przez maszyny odbywa się z wykorzystaniem wzorów Taylora. Jest to szereg zawierający pewną, określoną ilość wyrazów, która determinuje dokładność odwzorowania funkcji sinus. Wzór szeregu pokazano poniżej:

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \quad = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

Im więcej wyrazów posłuży do obliczenia funkcji, tym większą dokładność można uzyskać. Jednak w zastosowaniach cyfrowych istotną kwestią jest czas obliczeń oraz wielkość zasobów, jaka musi zostać do niego wykorzystana. Dodanie każdego kolejnego wyrazu skutkuje zwiększeniem zapotrzebowania na pamięć a także wydłuża proces obliczenia każdego składnika. Istnieje możliwość ograniczenia obliczenia liczby wyrazów, jednak skutkuje to powstawaniem błędu przybliżenia, który gwałtownie wzrasta wraz z ze zbliżaniem się argumentu do wartości Π/2. W celu zmniejszenia błędów przybliżenia możliwa jest taka modyfikacja współczynników występujących przy kolejnych wyrazach szeregu, która pozwoli na równomierne rozłożenie błędu przybliżenia w całym zakresie argumentów, przy jednoczesnym znacznym zmniejszeniu maksymalnej wartości błędu. Dokonuje się tego za pomocą wielomianów Czebyszewa. Wielomiany te powstają rekurencyjnie i mają następującą postać:

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_k(x) = 2x \cdot T_{k-1}(x) - T_{k-2}(x)$$

Algorytm optymalizacji obliczeń jest następujący:

1) Wybieramy liczbę wyrazów za pomocą których ma zostać obliczona funkcja sinus(np. 2 wyrazy, oznacza to, że pozostaną jedynie argumenty o potęgach 1 oraz 3)
2) Wyznaczamy wielomian Czebyszewa, w którym największa potęga argumentu będzie równa potędze Szeregu Taylora, która została obcięta w celu optymalizacji (w przypadku powyższego przykładu, jest to wartość potęgi równa 5, gdyż pozostawiono argumenty o potęgach 1 oraz 3). Wielomian ten ma postać:

$$\boxed{T_5(x) = 16x^5 - 20x^3 + 5x}$$

3) Przekształcamy uzyskany wielomian tak, aby uzyskać wzór na najwyższą potęgę argumentu (w tym wypadku wzór na x do potęgi 5). Pozostający w tym wzorze wyraz T(x) zostaje odrzucony, jako reszta. Ponieważ wartość wyrazu T jest zawarta w przedziale od -1 do 1 to całkowity błąd przybliżenia jest wyrażony przez wartość współczynnika znajdującego się przy T(x). Wzór ten ma postać:

$$\boxed{x^5 = \frac{1}{16}(20x^3 - 5x + T_5)}$$

4) Podstawiamy uzyskany wzór do szeregu Taylora w miejsce czynnika x do potęgi 5 uzyskując nowe wartości współczynników. Należy pamiętać, że w szeregu Taylora każdy współczynnik ma w mianowniku wartość, będącą silnią potęgi danego

argumentu. W tym wypadku podstawiając do szeregu wzór na x do potęgi 5 należy pamiętać, aby całość podzielić przez wartość silnia z 5.

W celu pokazania wyników powyższej procedury został stworzony program, który oblicza wartości sinusa. Program wylicza wartość wzorcową korzystając z biblioteki matematycznej. Następnie wylicza przykładową wartość sinusa korzystając bezpośrednio z wzoru na szereg Taylora oraz wartość zoptymalizowaną korzystając ze zmodyfikowanych wzorów uzupełnionych o wielomiany Czebyszewa.
Kod programu został przedstawiony poniżej:

```
#include "ADDS_21161_EzKit.h"
#include <def21161.h>
#include <signal.h>
#include <math.h>


float dm sinus_wzor[1501];
float dm sinus_taylor[1501];
float dm sinus_taylor_opt[1501];

float dm blad[1501];
float dm blad_opt[1501];

int dm wsp[8]={0, 0, 0, 0, 0, 0, 0,0};
float dm wspb[8]={0, 0, 0, 0, 0, 0, 0,0};
float x,a,b,c,d,e,a1,b1,c1,d1,e1,f1;
int i,j=0;
int n=5; // z zakresu 2 do 5



int silnia(int n)
        {
          int i=2, s=1;

          while( i < n+1 )
          {
            s = s * i;
            i = i + 1;
          }
          return s;
        }

void main()
{


        a1=silnia(1);
        b1=silnia(3);
        c1=silnia(5);
```

```
        d1=silnia(7);
        e1=silnia(9);
        f1=silnia(11);
        a=1.0/a1;
        b=1.0/b1;
        c=1.0/c1;
        d=1.0/d1;
        e=1.0/e1;
        for (i = 0; i<=1500 ; i++)
         {
                x = i*0.001;

                sinus_wzor[i] = sinf(x);
                sinus_taylor[i] = 0;
                for (j = 1; j<=n ; j++)
                {
                        sinus_taylor[i] += ((j % 2) ? 1 : -1) * (pow(x,j*2-1))/silnia(j*2-1);
                }
                blad[i]=fabs(sinus_wzor[i]-sinus_taylor[i]);
                if (n==2)
       {
                        sinus_taylor_opt[i]=x-b*pow(x,3)+(20.0*pow(x,3)-5.0*x)/(16.0*c1);

                        blad_opt[i]=fabs(sinus_wzor[i]-sinus_taylor_opt[i]);
                }
                else if (n==3)
                {
               sinus_taylor_opt[i]=x-b*pow(x,3)+c*pow(x,5)-(112.0*pow(x,5)-
56.0*pow(x,3)+7.0*x)/(64.0*d1);
                        blad_opt[i]=fabs(sinus_wzor[i]-sinus_taylor_opt[i]);
                }
                else if (n==4)
                {
               sinus_taylor_opt[i]=x-b*pow(x,3)+c*pow(x,5)-d*pow(x,7)+(576.0*pow(x,7)-
432.0*pow(x,5)+120.0*pow(x,3)-9.0*x)/(256.0*e1);
                        blad_opt[i]=fabs(sinus_wzor[i]-sinus_taylor_opt[i]);
                }
                else if (n==5)
                {
               sinus_taylor_opt[i]=x-b*pow(x,3)+c*pow(x,5)-d*pow(x,7)+e*pow(x,9)-
(2816.0*pow(x,9)-2816*pow(x,7)+1232.0*pow(x,5)-220.0*pow(x,3)+11.0*x)/(1024.0*f1);

                        blad_opt[i]=fabs(sinus_wzor[i]-sinus_taylor_opt[i]);
                }

        }



}
```

Kod oblicza następujące wartości:

1) sinus_wzor – jest to wzorcowa wartość sinusa obliczona z wykorzystaniem biblioteki math
2) sinus_taylor – wartość sinusa obliczona z czystego wzoru Taylora
3) sinus_taylor_opt – zoptymalizowana wartość sinusa
4) blad – jest to wartość pokazująca różnicę między wzorcowym sinusem a sinusem obliczonym ze wzoru Taylora. Blad = sinus_wzor – sinus_taylor
5) blad_opt – jest to wartość pokazująca różnicę między wzorcowym sinusem a sinusem obliczonym ze zmodyfikowanego wzoru Taylora. blad_opt = sinus_wzor – sinus_taylor_opt.

W celu obejrzenia wartości zmiennych należy wykonać następujące czynności:
1) W pierwszej kolejności należy wybrać zadaną ilość wyrazów szeregu Taylora, na podstawie której zostają wykonane wszystkie obliczenia. Wartość tę określa się ustalając zmienną „n” na 2 do 5. Oznacza to, że szereg zostanie przybliżony i zoptymalizowany za pomocą dwóch do pięciu wyrazów. Skompilować projekt – opcja Rebuild projekt.
2) Uruchomić program – opcja Run
3) Program po wykonaniu obliczeń zostanie automatycznie wstrzymany, można obejrzeć wyniki. W tym celu z menu „view” w górnym pasku wybrać Debug Windows -> Plot ->New. W wyświetlonym oknie należy za pomocą przycisku „browse” odnaleźć nazwy zmiennych, które chcemy wykreślić, w polu „count” wpisać wartość 1501 – jest to liczba wyliczonych próbek oraz z rozwijalnego paska Data wybrać „float”. Aby wyświetlić więcej przebiegów na jednym wykresie można dodać kolejny data_set powtarzając ostatnią czynność.
W celu obejrzenia wartości błędów należy na jednym wykresie umieścić przebiegi blad oraz blad_opt. W celu wyświetlenia legendy należy w oknie wykresu kliknąć prawym przyciskiem myszy, wybrać z rozwiniętego menu opcję „modify settings” i w zakładce „General” zaznaczyć opcję „Legend”. Można dodatkowo w zakładce „Font” zwiększyć rozmiar czcionki.

Przestawiony powyżej pogram umożliwia szybki wgląd w zagadnienie dokładności przybliżenia obliczeń. Za jego pomocą można obejrzeć jak wygląda zależność między liczbą wyrazów wykorzystanych do obliczeń a ich dokładnością. Program porównuje obliczone wyrazy do wzorcowej funkcji o dokładności maszynowej (32 bity). Ponadto pokazano sposób optymalizacji obliczeń z wykorzystaniem wielomianów Czebyszewa.