SPIS TREŚCI

1. Wstęp **5**

2. Arytmetyka rozproszona **6**

2.1. Symulacja układu programowalnego 7

3. Realizacja algorytmu cyfrowego filtrowania **9**

4. ... . **10**

5. ... . **12**

6. Podsumowanie **13**

Literatura **14**

Spis rysunków **15**

Spis tabel **15**

Spis załączników **15**

Tytuł rozdziału głównego – czcionka: Times New Roman, 16 pkt, Bold; wyrównanie: obustronne; wcięcie specjalne: 0,76 cm; akapit odstępy: przed i po: 24 pt; interlinia: 1,5 wiersza.

1. Wstęp

W ostatnich latach można zaobserwować gwałtowny wzrost zastosowań algorytmów cyfrowego przetwarzania sygnałów. Spowodowane jest to zwiększeniem wydajności procesorów DSP (ang. *Digital Signal Processing).* Fakt ten pozwala na implementację coraz to bardziej skomplikowanych algorytmów. Sz programowalne typu FPGA i CPLD. Układy progra programowanie i testowanie zanim ostateczna wersja zostanie wdrożona do produkcji.

Określenie w obcym języku – skrót używanego języka obcego; czcionka: Times New Roman, 12 pkt; kursywa.

…

Celem pracy jest implementacja algorytmów cyfrowego przetwarzania sygnałów w strukturach programowalnych z wykorzystaniem arytmetyki rozproszonej.

…

Praca podzielona jest na cztery rozdziały zasadnicze. W rozdziale drugim przedstawiono arytmetykę rozproszoną w strukturach FPGA i CPLD. Rozdział trzeci został w całości poświęcony …

Akapit - czcionka: Times New Roman, 12 pkt; wyrównanie: obustronne; wcięcie specjalne: pierwszy wiersz 1 cm (stałe w całej pracy); akapity nie dłuższe niż pół strony; interlinia: 1,5 wiersza (stałe w całej pracy).

2. Arytmetyka rozproszona

Arytmetyka rozproszona [2, 3] jest znana od ponad dwóch dekad i była intensywnie używana do konstruowania urządzeń liczących, w czasach gdy nie było mikroprocesorów, a jedynie układy logiczne średniej skali integracji. Arytmetyka rozproszona to sprzętowa realizacja algorytmu obliczeniowego wykonującego mnożenie za pomocą wstępnie obliczonej tablicy i sumowanie wyników kolejnych mnożeń. Taki typ obliczeń jest podstawowym dla wielu funkcji z zakresu Cyfrowego Przetwarzania Sygnałów (CPS). Sama idea algorytmu jest bardzo prosta lec nakładu układów logicznych [6].

Odwołania do literatury - w nawiasach kwadratowych numery pozycji z bibliografii.

…

Obliczenia realizowane za pomocą AR możemy zapisać w postaci równania:

$$y = \sum_{k=1}^{K} A_k X_k \qquad (1)$$

gdzie: y – wynik, X_k – k-ta zmienna, A_k – stały współczynnik dla k-tej zmiennej, K – ilość zmiennych.

Wzory matematyczne – 1 linia odstępu przed i po wzorze, wzór wyśrodkowany, symbole we wzorze Italic.
Numeracja wzorów – w nawiasie okrągłym kolejno zgodnie z umieszczaniem w pracy, wyrównanie: do prawej.

Równanie tego typu, w technice CPS
na lub szybka tran
nus. Wygodną i
alizowana tak aby
ch procesorów).

Oznaczenia do wzorów – wyrównanie: obustronne; wcięcia: z lewej -1 cm, specjalne: wysunięcie 1cm; interlinia: 1,5 wiersza; standardowa czcionka akapitu (Times New Roman, 12 pkt), symbole w oznaczeniach wzorów Italic.

X_k można przedstawić w postaci równania:

$$X_k = -x_{k0} + \sum_{b=1}^{B-1} x_{kb} 2^{-b} \qquad (2)$$

gdzie: x_{k0}, x_{kb} – zmienne binarne mogące przyjąć tylko jedną z wartości 0 lub 1, B – ilość bitów w słowie.

….

Podstawiając równanie (2) do równania (1) otrzymamy:

$$y = \sum_{k=1}^{K} A_k \left[-x_{k0} + \sum_{b=1}^{B-1} x_{kb} \, 2^{-b} \right] \qquad (3)$$

Odwołanie do wzoru – numer wzoru ujęty w nawiasie okrągłym.

…

Tak zapisane równanie pozwala na określenie prostego algorytmu jego wyliczenia. Należy zauważyć, że każde wyrażenie w kwadratowym nawiasie ma taki sam skończony zestaw wartości, gdyż zmienne binarne x_{kb} mimo, że należą do różnych słów przyjmują wartość 0 lub 1. Zestaw możliwych wartości to:

- 0 - gdy wszystkie x_{kb} są zerami (wartość S_0=00..00 słowa o długości równej ilości zmiennych w równaniu),
- A_1 - gdy x_{1b} =1 pozostałe są zerami (kolejna wartość S_1=00..01 słowa),
- A_2 - gdy x_{2b} =1 pozostałe są zerami (kolej
- A_1+A_2 - gdy x_{1b} i x_{2b} =1 pozostałe są zera
- itd., aż do $A_1+A_2...+A_K$ - gdy wszystkie x

Wyliczenia, wypunktowanie – zastosowane znaki, ustawienie w stosunku do lewej krawędzi, odstępy muszą być identyczne w całej pracy.

Wystarczy wyliczyć wszystkie sumy wsp isać je do pamięci ROM pod adresami takimi jak odpowiadająca danej sumie kombinacja…

2.1. Symulacja układu programowalnego

Przetestowanie powstałej struktury możliwe
Do tego możemy wykorzystać wbudowane narzęd

Tytuł podrozdziału – czcionka: Times New Roman, 14 pkt, Bold; wyrównanie: obustronne; wcięcie specjalne: 1,02 cm; akapit odstępy: przed i po: 12 pt; interlinia: 1,5 wiersza.

…

Zawartość pamięci układu pojedynczej
kompilacją układu przedstawiono w Tabeli 2.1.

Odwołanie do tabeli – słowo tabela pisane wielką literą.

Tabela 2.1. Zawartość pamięci układu pojedynczej kaskady

Kod wejściowy					Zawartość pamięci	Wyliczone wartości
Input11	a0	a1	a2	a3		
0	0	0	0	0	0	0
0	0	0	0	1	A_4	40h
0	0	0	1	0	A_3	40h
0	0	0	1	1	$A_3 + A_4$	80h
0	0	1	0	0	A_2	
0	0	1	0	1	$A_2 +$	
0	0	1	1	0	$A_2 +$	
0	0	1	1	1	$A_2 + A_3$	
0	1	0	0	0	A_1	
0	1	0	0	1	$A_1 +$	
0	1	0	1	0	$A_1 +$	
0	1	0	1	1	$A_1 + A_3$	
0	1	1	0	0	$A_1 + A_2$	80h
0	1	1	0	1	$A_1 + A_2 + A_4$	120h
0	1	1	1	0	$A_1 + A_2 + A_3$	120h
0	1	1	1	1	$A_1 + A_2 + A_3 + A_4$	160h
1	0	0	0	0	0	0
1	0	0	0	1	$- A_4$	-40h
1	0	0	1	0	$- A_3$	-40h
1	0	0	1	1	$- (A_3 + A_4)$	-80h
1	0	1	0	0	$- A_2$	-40h
1	0	1	0	1	$- (A_2 + A_4)$	-80h
1	0	1	1	0	$- (A_2 + A_3)$	-80h
1	0	1	1	1	$- (A_2 + A_3 + A_4)$	-120h
1	1	0	0	0	$- A_1$	-40h
1	1	0	0	1	$- (A_1 + A_4)$	-80h
1	1	0	1	0	$- (A_1 + A_3)$	-80h
1	1	0	1	1	$- (A_1 + A_3 + A_4)$	-120h
1	1	1	0	0	$- (A_1 + A_2)$	-80h
1	1	1	0	1	$- (A_1 + A_2 + A_4)$	-120h
1	1	1	1	0	$- (A_1 + A_2 + A_3)$	-120h
1	1	1	1	1	$- (A_1 + A_2 + A_3 + A_4)$	-160h

Zasady numeracji tabel – kolejno zgodnie z umieszczaniem w rozdziale.
Tytuł tabeli – nad tabelą, wyśrodkowany, standardowa czcionka akapitu (Times New Roman, 12 pkt).

Źródło: Opracowanie własne na podstawie: Xilinx "The Role of Distributed Arithmetic in FPGA-based Signal Processing", application note 1996.

Źródło – czcionka: Times New Roman, 12 pkt; kursywa; wyrównanie: obustronne.

3. Realizacja algorytmu cyfrowego filtrowania

…

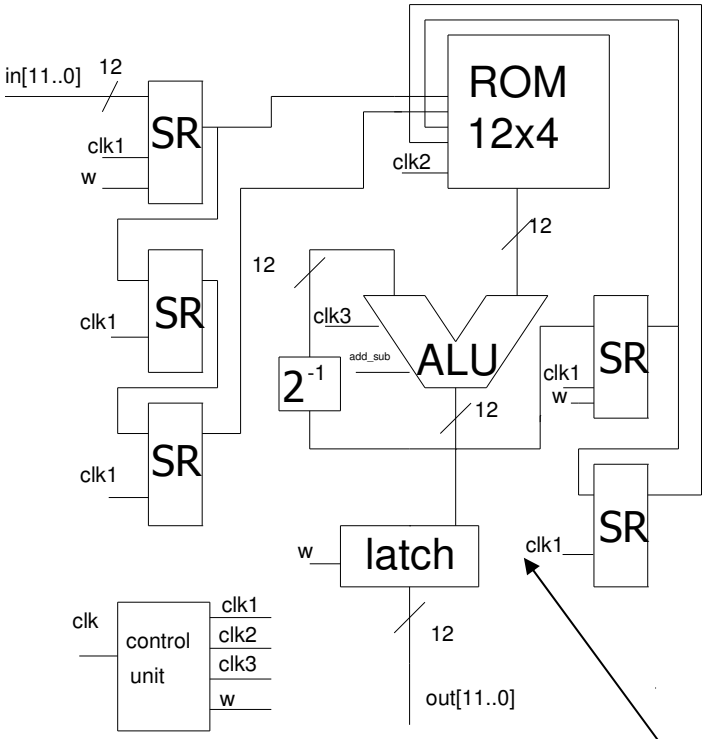
Do realizacji przykładu posłużymy się filtrem 2 rzędu postaci:

$$y(n) = a_0 \cdot x(n) + a_0 \cdot x(n-2) - b_1 \cdot y(n -$$

Odwołanie do rysunku – skrót rys. pisany wielką literą.

…

Schemat blokowy filtru IIR działającego z wykorzystaniem arytmetyki rozproszonej ilustruje Rys. 3.1.



Rys. 3.1. Schemat blokowy filtru

Źródło: Opracowanie własne.

Zasady numeracji rysunków – kolejno zgodnie z umieszczaniem w rozdziale.
Tytuł rysunku – skrót rys., pod rysunkiem, wyrównanie: wyśrodkowanie, standardowa czcionka akapitu (Times New Roman, 12 pkt).

Rysunek – wyrównanie: wyśrodkowanie, należy zwracać uwagę na odpowiednie zakotwiczenie rysunku.

Źródło – czcionka: Times New Roman, 12 pkt; kursywa; wyrównanie: wyśrodkowanie.

4. ...

Kod programu sortującego całą tablicę po wierszach i kolumnach przedstawia Listing 4.1.

```
void sortuj2 (Tablica2 t)
  {
   int tym[M*N],i,j,k;
   for (i=0;i<M*N;)
         {
          for(j=0;j<N;j++)
             for(k=0;k<M;k++)
                   tym[i++]=t[j][k];
          }

    for (j=0;j<M*N;j++)
        for (k=0;k<M*N;k++)
            if (tym[j]<tym[k])
               zamiana(&tym[j],

    for (i=0;i<M*N;)
        {
        for(j=0;j<N;j++)
            for(k=0;k<M;k++)
                  t[j][k]=tym
        }
    }
```

Przykład kodu źródłowego – wyrównanie: obustronne; wcięcia: z lewej -1,75 cm; interlinia: 1 wiersz; standardowa czcionka akapitu (Courier New, 12 pkt).

Listing 4.1. Sortowanie całej tablicy po wierszach i kolumnach

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity addacc is
  Port ( wej: in std_logic_vector(7 downto 0);
         clk: in std_logic;
         res: in std_logic;
         wyj: inout std_logic_vector(7 downto 0);
         przen: out std_logic );
end addacc;

architecture akumulator of addacc is
begin
  process (clk,res)
  begin
    if res = '1' then
```

```
            wyj <= "00000000";
            przen <= '0';
          elsif (clk = '1' and clk'event) then
            wyj <= wyj+wej;
            if (wyj+wej) > "11111111"then
              przen <= '1';
            else
              przen <= '0';
            end if;
          end if;
        end process;
      end akumulator;
```

Listing 4.2. Moduł 8-bitowego sumatora

5. ...

6. Podsumowanie

Celem pracy było ...

…

Przeprowadzona w pracy analiza, będąca próbą empirycznej weryfikacji zastosowania arytmetyki rozproszonej do realizacji…

W badaniach zastosowano nowe podejście do …

Znaczący wpływ na szybkość pracy wykorzystanych układów programowalnych uzyskano za pomocą…

...

.

Pozycje w literaturze – wyrównanie: obustronne; wcięcia: z lewej -0,5cm, specjalne: wysunięcie 0,63 cm; interlinia: 1,5 wiersza; standardowa czcionka akapitu (Times New Roman, 12 pkt).

Literatura

1. T. Łuba, K. Jasiński, B. Zbierzchowski, *Specjalizowane układy cyfrowe w strukturach PLD i FPG*, Wydawnictwo WKŁ, Warszawa 1997.
2. S. A. White, *Applications of Distributed Arithmetic to Digital Signal Procesing: A Tutorial Review*, „*IEEE ASSP Magazine”,* July 1989, pp. 4-19.
3. Xilinx, *The Role of Distributed Arithmetic in FPGA-based Signal Processing*, application note 1996.
4. M. Omieljanowicz, A. A. Piotrowski, *Procesor potokowy banku cyfrowych filtrów na bazie arytmetyki rozproszonej*, IV Krajowa Konferencja KOWBAN '97, 16 -18 październik 1997,Wrocław WNT '97, s. 341-346.
5. L. Mintzer, *Large FFT's in Single FPGA*, proceedings ICSPAT'98.
6. Altera Corporation. http://www.altera.com, stan z dnia 12.02.2011.
7. …

Wzory odwołań do literatury -
- książki:
I. Nazwisko, *Tytuł*, Wydawca, Miejsce i rok wydania.
- opracowania w publikacjach zwartych:
I. Nazwisko, *Tytuł opracowania*, [w:] *Tytuł książki*, I. Nazwisko (red.), Wydawca, Miejsce i rok wydania.
- artykuły w czasopismach:
I. Nazwisko, *Tytuł,* „Tytuł Czasopisma”, rok, numer.
- strony internetowe:
I. Nazwisko (jeśli autor nie jest znany podajemy nazwę organizacji prowadzącej serwis). Witryna internetowa. http://...., stan z dnia (data odczytu).

Spis rysunków

Rys. 3.1. Schemat blokowy filtru…....……..…..……..…9
…

Spis tabel

Tabela 2.1. Zawartość pamięci układu pojedynczej kaskady…....……..……..… 8
…

Spis załączników

Załącznik 1. Wykaz dostępnych układów programowalnych firmy XILINX (stan na 1 lutego 2011 roku)

Załącznik 2. Kody źródłowe funkcji implementowanych w struktury programowalne
…