Janusz Jabłoński, Robert DYLEWSKI
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUCJA, Szafrana, 4a, 65-516, Zielona Góra
PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA, Teatralna 25, 66-400 Gorzów Wlkp

Szybkie mnożenie modulo 2^k-1

dr inż. Janusz JABŁOŃSKI
W latach 1994 – 2003 asystent, następnie adiunkt Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Od września 2004 adiunkt Wydziału Matematyki, Informatyki i Ekonometrii UZ. Obecnie prowadzi zajęcia z zakresu Inżynierii Oprogramowania. Posiada, również doświadczenie dydaktyczne w zakresie architektury komputerów oraz logiki programowalnej. Podstawowe badania związane z wykorzystaniem arytmetyki resztowej oraz układów FPGA w akceleracji obliczeń.

e-mail: J.Jablonski@wmie.uz.zgora.pl

dr Robert DYLEWSKI

Doktorat uzyskany w roku 2003, z zakresu metod numerycznych i optymalizacji. Prowadzone zajęcia: programowanie matematyczne, badania operacyjne, podstawy optymalizacji, algorytmy i struktury danych, metody numeryczne. Tematyka badań naukowych: optymalizacja wypukła nieróżniczkowalna, metody rzutowe i zastosowania, własności i zastosowania sieci Petriego.

e-mail: r.dylewski@wmie.uz.zgora.pl

Streszczenie

W artykule przedstawiona została koncepcja mnożenia oparta na reduktorze 4:2. Proponowane rozwiązanie nie korzysta z typowego podejścia optymalizującego opartego na sumatorach CSA w strukturze drzewa Wallec'a. W proponowanym rozwiązaniu multiplikatora modulo 2^k-1 optymalizacja charakterystyki T korzysta z autorskiego rozwiązania reduktora 4:2 oraz eliminowania przypadków szczególnych - nie występujących lub nieosiągalnych wartości.

Słowa kluczowe: CSA, mnożenie modulo, matryca mnożąca,

Fast modulo 2^k-1 multiplication

Abstract

This paper presents the new implementation approach for fast multiplication. The new approach depends on utilization the reducer 4:2 introduced on drawing 4 instead of two levels the CSA [8]. It additionally for improvement timing profile of modulo 2^k - 1 multiplication (in this example $k = 4$), the circuit implemented multiplication schema presented in picture 5 was simplified, because in proposed solution of modulo multiplication the position with weight 2^2 will not step out. The proposed solutions show at picture 5 uses the "bit grouping method" described in equation 3, whose was proposed in [1]. Applying traditional methods leaning on CSA in constructing the Wallace'a tree would not be possible eliminate of level 2^2 because on four bit position are possible different combinations three bits by value "1". Introduced on drawing 4 patern reducer 4:2 as well as to use principle the depending on detecting of non-existent input combinations mighty with success both in multiplication how also in multiplication modulo for arguments larger then 4 bits size. Introduced in summary results show how the large possibilities of utilization proposed method in optimization of the time of multiplication as well as multiplication modulo.

Keywords: multiplication, RNS, modulo, Wallace'a tree

1. Wstęp

Dla opisu danego zjawiska aparatem matematyczny pozwalają również symulować przebieg wielu zjawisk. Obecnie modele matematyczne cyfrowego przetwarzania sygnałów (DSP, ang. *Digital Signal Processing*) [2, 3]. Przetwarzanie obrazów oparte na realizacji transformat, jak ale również potęgowanie metodą iteracyjne w krptografii RSA są formą mnożenia akumulacyjnego (MAC, ang. *Multiply and Accumulate*). Schemat MAC w mnożeniu macierzy korzysta z operacji arytmetycznych dodawania oraz mnożenia [4]. W szczególności wykorzystanie zrównoleglenia, na przykład w sumatorach prefiksowych (PPA, ang. *Paraller Prefix Adder*) [5], prowadzi do przyspieszenia obliczeń. Dla sumatorów prefiksowych złożoność wyrażona charakterystyką T - liczba poziomów logicznych na ścieżce krytycznej, a zatem również czas realizacji dodawania jest funkcją logarytmiczną rozmiaru bitowego argumentów, natomiast złożoność szybkiego mnożenia wyrażana jest funkcja kwadratową - dla prostej implementacji opartej na sekwencyjnym dodawaniu, lub liniową - dla implementacji układów matrycowych, efektywniejsze są rozwiązania specjalizowane oparte na drzewiastej strukturze elementów wykonawczych [1, 6]. Efekt zmniejszenia rozmiarów argumentów oraz zrównoleglenia przetwarzania można osiągnąć między innymi przez wykorzystanie systemu resztowego w reprezentacji liczb (RNS, ang. *Residue Number System*) [1, 2, 3 ,7].

2. Resztowa reprezentacja argumentów i arytmetyka modularna

W RNS liczby reprezentowane są jako wektor reszt $X = \{x_0,x_1,..., x_k\}$ będących cyframi obliczanymi jako reszty z dzielenia modulo $x_i = X \bmod m_i$ w dalszych rozważaniach oznaczane jako $x_i = \left|X\right|_{mi}$. Jeżeli moduły m_i są różne i parami względnie pierwsze $NWD(m_i \neq m_j) = 1$ to bazą systemu resztowego jest wektor $B=\{m_0,m_1,...,m_n\}$ dla $i = 0,1,2,...,k$ a zakresem reprezentacji jest $M= \prod m_i$. Arytmetyka w RNS nazywana jest „arytmetyką bez przeniesień” co oznacza, że dodawanie i mnożenie realizowane są niezależnie na poszczególnych składowych wektora reszt reprezentujących argumenty operacji arytmetycznych a wynikiem operacji dodawania i mnożenia $Z_B=X_B \oplus Y_B$ jest wektor reszt $Z_B=\{z_0, z_1,...,z_k\}_B$. Operacje dodawania i mnożenia są realizowane niezależnie w poszczególnych kanałach resztowych $m_i<<X,Y$, więc argumentami działań $\left|x_i \oplus y_i\right|_{mi}$ są znacznie mniejsze liczby i można oczekiwać przyspieszenia obliczeń proporcjonalnego do zmniejszenia rozmiaru argumentów. W mnożeniu zakres wyniku wynosi $n = \log_2 X + \log_2 Y$. Zatem, dla osiągnięcia przyspieszenia obliczeń przez zastosowanie RNS należy uwzględniać kilka przesłanek przy konstruowaniu bazy B systemu:

- ✓ zakres wyników n ≤ M,
- ✓ rozmiary reszt znacząco mniejsze od argumentów,
- ✓ operacje na resztach wykonywane szybciej aniżeli na oryginalnych argumentach,
- ✓ zbliżony rozmiar i czas operacji dla kanałów resztowych.

Zwiększenie liczby kanałów resztowych ma wpływ na rozszerzenie zakresu dynamicznego RNS oraz zwiększenie zrównoleglenia przetwarzania na mniejszych argumentach, jednakże zwiększanie liczby modułów powoduje znaczne dysproporcje w szerokości kanałów resztowych oraz komplikuje proces wyznaczania wartości reszt. Problem wyboru bazy RNS jest tematem wielu opracowań [1, 2]. Można nakreślić podstawowe wskazówki dotyczące kandydatów na moduły m_i wchodzące w skład bazy B systemu resztowego:

- ✓ rozmiary reszt powinny być znacząco mniejsze od rozmiarów argumentów
- ✓ czas T dla operacji modulo m_i mniejszy aniżeli czas operacji na argumentach
- ✓ zbliżony rozmiar i czas operacji (+,*) dla poszczególnych kanałów resztowych.

Praktyczne zastosowania znajdują niektóre zestawy modułów, najczęściej typu n - 1, n, n + 1, gdzie n jest typu $2^{k\pm i}$.

3. Arytmetyka modulo 2^k±1

Argumentami operacji arytmetycznych w RNS są składowe wektora reszt a operacją mającą wpływ na szybkość przetwarzania jest mnożenie modulo w szczególności mnożenie modulo 2^k±1 [6, 7, 9].

Reprezentacją binarną wyniku mnożenia $Z = X \otimes Y$ jest wektor $Z = \{z_{n-1}, z_{n-2}, \ldots, z_k, z_{k-1}, \ldots, z_1, z_0\}$, którego wartość można zapisać również jako złożenie:

$$Z = \sum_{i=0}^{n-1} 2^i z_i = 2^k \sum_{j=0}^{n-k-1} 2^j z_{j+k} + \sum_{i=0}^{k-1} \left(2^i z_i\right). \quad (1)$$

Podstawiając wyznaczenie wyniku z mnożenia reszt modulo 2^k±1 można zrealizować jako:

$$\left|Z\right|_{2^k \pm 1} = \left|\sum_{l=0}^{j-1} 2^{kl} Z_l\right|_{2^k \pm 1} = \left|\sum_{l=0}^{j-1} (\mp)^l Z_l\right|_{2^k \pm 1}. \quad (2)$$

Argumenty mnożenia to reszty modulo 2^k±1, więc wynik to wektor $Z = \{Z_1, Z_0\}$ a resztę można wyznaczyć jako:

$$\left|Z\right|_{2^k-1} = \left|Z_0 + Z_1\right|_{2^k-1}. \quad (3)$$

W kodzie U2 równanie (3) można zastąpić równaniem:

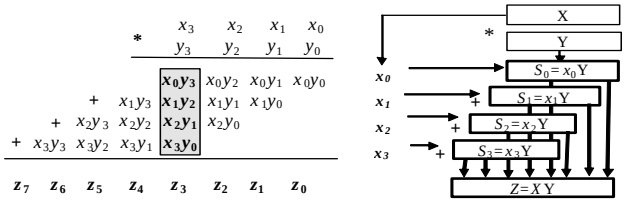
$$\left|Z\right|_{2^k+1} = \left|Z_0 - Z_1\right|_{2^k+1} = \left|Z_0 + \overline{Z}_1 + 1\right|_{2^k+1}, \quad (4)$$

gdzie, $\overline{Z}_1$ oznacza zanegowane wartości bitów Z_1.

Zatem wyznaczanie reszty można wykonać w PPA [6].

4. Efektywne mnożenie

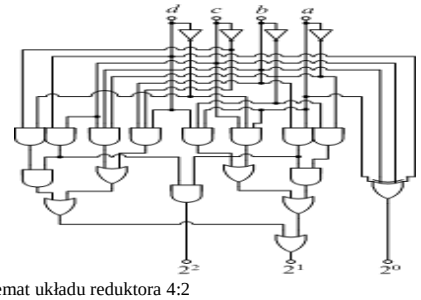
Układy mnożące dzieli się na matrycowe oraz równoległe [1, 2, 6, 9]. Mnożenie matrycowe przedstawione na rysunku. 1 wymaga 3 poziomów CSA a mnożenie równoległe w oparciu o CSA w strukturze drzewa Wallace'a tylko dwu poziomów CSA [8]. Przyjmując, że czas operacji wyrażony charakterystyką T dla CSA wynosi 4 dla 4-ro bitowego mnożenia matrycowego opóźnienie $T_M = 19$ natomiast dla układu równoległego opóźnienie $T_R = 14$.



Rys. 1. Schemat mnożenia
Fig. 1. Multiplication schema

5. Propozycja reduktora 4:2

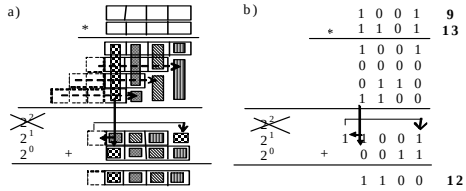
Schemat z rysunku 4 przedstawia implementację układu reduktora 4:2 wraz z wykrywaniem szczególnego przypadku - wyjście o wadze 4. Liczba poziomów logicznych reduktora 4:2 z rysunku 4 T = 5 a dla proponowanych rozwiązań reduktorów 4:2 opartych na dwu poziomach CSA liczba poziomów T = 8.



Rys. 2. Schemat układu reduktora 4:2
Fig. 2. Schema the 4:2 circuit

6. Mnożenie modulo 2^k-1 oparte na reduktorze 4:2

Wykorzystując równanie (3) dzielenie modulo 2^k-1 można zastąpić dodawaniem Z_0+Z_1. W mnożeniu 9 * 13 = 117 modulo 15 z rysunku 3 wyznaczenie reszty z zostanie sprowadzone do sumy 9 + 3 = 12. Na rysunku 3a przedstawiono wykorzystanie reduktora 4:2 do wyznaczenia liczby "1" dla sumowania wyników częściowych S_i. Dla reszt modulo 2^k-1 czynniki są mniejsze od 2^k - 1, zatem pozycja o wadze "2^2" nie wystąpi.



Rys. 3. Schemat mnożenia modulo 2^4-1.
Fig. 3. Multiplication modulo 2^4-1 schema

W proponowanym schemacie wystąpi jeden poziom redukcji $L_{4:2}$ o charakterystyce $T_{4:2} = 5$, a charakterystyka układu mnożącego wyniesie $T_P = 5 * L_{4:2} + Ts_R = T_P = 5*1 + 9 = 14$. Natomiast dla czterobitowego układu mnożenia modulo 2^k-1 charakterystyka $T_{MM} = 5*1 + Ts_M = 5+ 7 = 12$.

7. Wnioski

Przedstawione rozwiązanie charakteryzuje znaczące zmniejszenie wartości charakterystyki T dla mnożenia w stosunku do rozwiązania matrycowego wynoszące 19/14, natomiast w stosunku do rozwiązania równoległego 17/14. Taki efekt udało się uzyskać dzięki zastosowaniu proponowanego reduktora 4:2 oraz wyeliminowania poziomu "2^2" dla nie występujących wartości w mnożeniu. W tradycyjnym układzie reduktora 4:2 opartym na dwu poziomach CSA redukujących w stosunku 3:2 liczbę jedynek nie udałoby się wyeliminować jednego poziomu CSA, ponieważ nie można przewidzieć w którymz czterech wierszy S_i nie wystąpi wartość "1". Proponowane rozwiązanie można wykorzystać dla mnożenia argumentów większego rozmiaru np. 8 lub 16 bitowych.

8. Literatura

[1] Biernat J., Architektura układów arytmetyki resztowej, Exit, W-w 2007.
[2] Soderstrand M. A., and all, Residue Number System Arithmetic: Modern Applications in Digital Signal Processing, IEEE Press, New York, 1986.
[3] Jabłoński J., Arytmetyka resztowa w cyfrowym przetwarzaniu sygnałów, KNWS '04, Zielona Góra 2004: s. 35--40. ISBN: 83-89712-16-4
[4] Jabłoński J., Arytmetyka resztowa w szyfrowaniu RSA, Przegląd Elektrotechniczny, 2010, nr 9, s. 145--148.
[5] Brent R., Kung H., A regular layout for parallel adders, IEEE Transactions on Computers vol. 31 (1982) 260–264.
[6] Zimmermann R., Efficient VLSI implementation of modulo (2^n±1) addition and multiplication, 14th IEEE Symposium on Computer Arithmetic, Australia -1999.
[7] Ma Y., A simplified architecture for modulo 2^n + 1 multiplication," IEEE Transactions on Computers, Vol. 47, 1998, pp. 333-337.
[8] Wallace C. S., A suggestion for a fast multiplier, IEEE Trans. on Electronic Comp. EC-13(1): 14-17 (196
[9] Wang Z., Jullien G. A., Miller W. C, A new design technique for column compression multipliers, IEEE Trans. Comput. 44 (8) (1995) 962–970.

o / received: 00.00.0000
przyjęto do druku / accepted: 00.00.0000 *artykuł recenzowany*