Adam MILIK, Andrzej PUŁKA
POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI
ul. Akademicka 16, 44-100 Gliwice

Efektywna implementacja algorytmu wyszukiwania wzorców genetycznych

Dr inż. Adam MILIK

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Pracę doktorską obronił w 2003 r. Jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe to układy logiki programowalnej, sterowniki programowalne, modelowanie i synteza złożonych układów sprzętowo-programowych.

e-mail: adam.milik@polsl.pl

Dr inż. Andrzej PUŁKA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Pracę doktorską obronił w 1997 r. Jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe to zastosowanie metod sztucznej inteligencji w elektronice, projektowanie, modelowanie i symulacja układów cyfrowych i mieszanych analogowo-cyfrowych w językach opisu sprzętu, metody projektowania współbieżnego systemów wbudowanych z podziałem na sprzęt i oprogramowanie.

e-mail: andrzej.pulka@polsl.pl

Streszczenie

W artykule zaprezentowano implementację algorytmu obliczającego stopień podobieństwa sekwencji znaków (genów) do zadanego wzorca. Algorytm wywodzi się z biologii obliczeniowej. Rozwiązania programowe wymagają znacznych zasobów sprzętowych oraz czasu. W badaniach nad algorytmem główny nacisk położono na poznanie jego własności i ich wykorzystanie przy implementacji. Pozwoliło to stworzyć bardzo oryginalna implementację zapewniającą niezwykle oszczędne gospodarowanie zasobami w układzie programowalnym jak i uzyskanie bardzo wysokich częstotliwości pracy.

Słowa kluczowe: Programowanie dynamiczne, metody numeryczne, identyfikacja wzorców, rozpoznawanie wzorców, przetwarzanie równoległe, przetwarzanie potokowe.

On efficient implementation of the search algorithm for genome patterns

Abstract

The paper describes implementation of the computation algorithm in modern, complex programmable hardware devices. The presented algorithm originates from computation biology and works on very long chains of symbols which come from reference patterns of the genome. The software solutions in this field are very limited and need large time and space resources. The main research efforts were aimed at investigating the properties of the searching algorithm. Especially, the influence of the penalty values assigned to the mismatch, insertion and deletion on the algorithm was analysed. This allowed obtaining a completely new algorithm offering extremely efficient implementation and exhibiting the outstanding performance. The Virtex 5 FPGA family was considered to be a target family for the searching algorithm based on the dynamic programming idea. The obtained results are very promising and show the dominance of the dedicated platform over the general purpose PC-based systems.

Keywords: dynamic programming, computational methods, pattern identification, pattern recognition, parallel processing, pipeline processing.

1. Wstęp

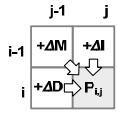
Sekwencja DNA w genomie przechowuje informacje o różnych cechach i predyspozycjach organizmu. Dynamiczny rozwój mikrobiologii oraz biologii molekularnej, który jest obserwowany w kilku ostatnich dekadach przyniósł ogromną liczbę informacji. Odcisnęła ona silny wpływ na nauki związane z przetwarzaniem informacji (głównie nauki matematyczne i informatyczne). Zainicjowało to prowadzenie badań nad nowymi metodami oraz poszukiwanie narzędzi pozwalających na efektywne przetwarzanie i badanie ogromnych zbirów danych biologicznych. Obecnie jednym z głównych celów jest poszukiwanie efektywnych algorytmów wyszukiwania sekwencji genetycznych. Wcześniejsze implementacje programowe dowiodły ich poprawności ale też ujawniły ich ograniczenia czasowe i przestrzenne w przypadku przetwarzania zbiorów danych sięgających miliardów (10^9) symboli. Wskazuje to na konieczność poszukiwania efektywnych metod implementowanych z wykorzystaniem układów sprzętowych. Prezentowana praca jest kontynuacją prowadzonych badań [7]. W pracy przedstawiono najnowsze opracowania dotyczące optymalizacji metody wyszukiwania pozwalającej na uzyskanie interesujących efektów implementacji.

2. Problem dopasowania sekwencji

Problem skrócenia czasu poszukiwania dla bardzo długich sekwencji jest elementem kluczowym wielu badań w tej dziedzinie [1, 2, 4]. Została opracowana znaczna liczba rozwiązań programowych. Koncentrują się na poprawie wydajności obliczeniowej przez zastosowanie różnych modyfikacji algorytmu oryginalnego oraz dostosowania do architektury komputera jak i wykorzystania specyficznych własności jednostki centralnej. Ciągły rozwój komputerów nie powstrzymuje jednak od poszukiwania innych rozwiązań w tej dziedzinie. Rozważany problem może być z powodzeniem rozwiązany przy zastosowaniu dedykowanych sprzętowych układów obliczeniowych. Wiele prac poświecono również implementacji algorytmu poszukującego dopasowania w strukturach sprzętowych [3, 5, 6]. Ciągły rozwój w dziedzinie układów scalonych a w szczególności logicznych układów programowalnych FPGA otwiera nowe możliwości w zakresie implementacji algorytmów w strukturach sprzętowych. Należy podkreślić że w przeciwieństwie do układów o sztywnej logice układy FPGA umożliwiają modyfikację realizowanego zadania. Można je porównać do uniwersalnych maszyn cyfrowych z wymiennym oprogramowaniem. W celu wykonania zadanych obliczeń zostaje wprowadzony określony program do pamięci maszyny, który będzie wykonywany. W układach programowalnych do pamięci (konfiguracyjnej) jest wprowadzana konfiguracja funkcji logicznych oraz połączeń odwzorowująca w sposób programowalny strukturę układu realizującego wybrane zadanie. Złożoność projektowania układów sprzętowych jest czynnikiem zniechęcającym do poszukiwania rozwiązań w obszarze układów programowalnych i rekonfigurowalnych.

3. Algorytm

Niech sekwencja reprezentująca genom będzie dana jako wektor symboli $\boldsymbol{Ref} = \{\boldsymbol{R}_1,\ldots,\boldsymbol{R}_n\}$. Sekwencja zapytania niech będzie dana przez wektor symboli $\boldsymbol{Qry} = \{\boldsymbol{Q}_1,\ldots,\boldsymbol{Q}_m\}$. Najlepsze dopasowanie może zostać znalezione za pomocą algorytmu programowania dynamicznego Smitha-Watermana w sposób iteracyjny. W skrócie algorytm ten będzie nazywany algorytmem SW. W pierwszym kroku funkcja oceny dopasowania przypisuje współczynniki wagowe do każdego węzła reprezentowanego przez komórki macierzy. Wypełnianie macierzy o n+1 kolumnach i m+1 wierszach następuje współczynnikami określonymi na podstawie bezpośredniego sąsiedztwa (rys. 1).



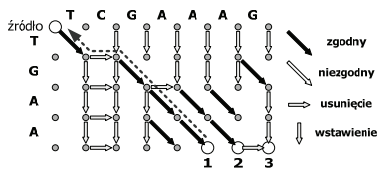
Rys. 1. Schemat wyznaczania wartości P przy użyciu algorytmu SW
Fig. 1. The process of evaluating the P function with SW algorithm

$$P(i,j) = MIN\begin{cases} P(i-1,j-1)+\Delta M & \text{(nie)dopasowanie} \\ P(i,j-1)+\Delta I & \text{wstawienie} \\ P(i-1,j)+\Delta D & \text{pominięcie} \end{cases} \quad (1)$$

Współczynnik ΔM jest zdefiniowany następująco:

$$\Delta M = \begin{cases} 0, & Q_i = R_j \quad \text{dopasowanie} \\ MIS, & Q_i \neq R_j \quad \text{niedopasowanie} \end{cases} \quad (2)$$

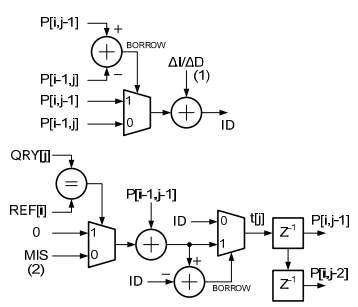
Po wypełnieniu całej macierzy współczynnikami dopasowania rozpoczyna się drugi krok polegający na określeniu najlepiej dopasowanej sekwencji (rys. 2). W kroku tym odnajduje się optymalną ścieżkę od wybranego węzła końcowego do węzła początkowego wybierając drogę z monotonicznie malejącymi współczynnikami wagowymi. W czasie wyznaczania ścieżki powrotnej można określić na podstawie współczynników wagowych istotną liczbę symboli niedopasowanych, wstawionych czy też pominiętych.



Rys. 2. Wyznaczanie współczynników wagowych oraz optymalnego dopasowania sekwencji przy użyciu programowania dynamicznego
Fig. 2. Illustration of the penalty value calculation and optimal cost path search (here sequence alignment) with dynamic programming

Do dalszych rozważań związanych ze sprzętową implementacją pierwszej części algorytmu SW (wyznaczenie współczynników dopasowania) przyjęto następujące wartości współczynników wstawienia ΔI, pominięcia ΔD i dopasowania ΔM:

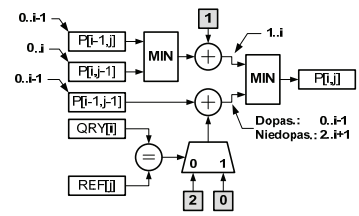
$$PEN = \begin{cases} \Delta M = 2 \\ \Delta I = 1 \\ \Delta D = 1 \end{cases} \quad (3)$$



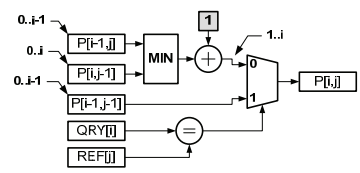
Rys. 3. Schemat blokowy komórki elementarnej SW uzyskanej w wyniku bezpośredniej implementacji
Fig. 3. Block diagram of the SW elementary cell obtained in direct implementation process

Do wyznaczenia dopasowania sekwencji kluczowym jest wyznaczenie współczynników dopasowania dla poszczególnych symboli sekwencji zapytania i sekwencji odniesienia. Początkowe implementacje układu dopasowania SW bazowały na bezpośrednim odwzorowaniu algorytmu w strukturach sprzętowych [7]. Stosując reguły upraszczania działań arytmetycznych można przedstawić schemat blokowy implementacji algorytmu (rys. 3). Użycie podejścia bezpośredniego bez wprowadzenia dodatkowych twierdzeń i obserwacji nie pozwoli osiągnąć dalszej redukcji złożoności układu obliczeniowego dla poszczególnych elementów.

Zmieniając reprezentację układu z postaci blokowej do postaci acyklicznego skierowanego grafu reprezentującego operacje a także dodatkowo opisanego przez zakres zmiennych pokazano na rysunku (rys. 4). Szczegółowa analiza grafu pozwala dostrzec możliwość dalszych optymalizacji. Graf przedstawia w sposób ogólny i-tą komórkę łańcucha. Wykorzystanie zakresu zmiennych pozwoliło usunąć bloki nieużywane, których funkcjonalność została pokryta przez pozostałe bloki. Przykładowo ścieżkę dodającą wartość 2 do zmiennej $P[i\text{-}1,j\text{-}1]$ została pokryta przez ścieżki obliczające wartość ścieżek wstawienia i pominięcia (rys. 5).



Rys. 4. Schemat blokowy komórki elementarnej SW uzyskanej w wyniku bezpośredniej implementacji
Fig. 4. Block diagram of the SW elementary cell obtained in direct implementation process



Rys. 5. Schemat blokowy komórki elementarnej SW uzyskanej w wyniku bezpośredniej implementacji
Fig. 5. Block diagram of the SW elementary cell obtained in direct implementation process

4. Unikalne cechy algorytmu SW

W wyniku działania algorytmu SW przy przyjęciu podanych wartości współczynników wagowych dla wstawienia, pominięcia i niedopasowania uzyskuje się macierz o szczególnych własnościach. Niech początkowe wartości macierzy P przyjmują następujące wartości:

$$\begin{aligned} &\forall_{0<i\le Q_{LEN}} \; P[i,0] = i \\ &\forall_{0<j\le R_{LEN}} \; P[0,j] = 0 \end{aligned} \quad (4)$$

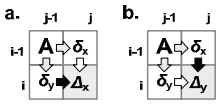
Poszczególne elementy macierzy wykazują następujące własności:

$$\forall_{\substack{0<i\le Q_{LEN} \\ 0<j\le R_{LEN}}} \begin{cases} P[i-1,j-1] \le P[i,j] \le P[i-1,j-1]+2 \\ P[i,j-1]-1 \le P[i,j] \le P[i,j-1]+1 \\ P[i-1,j]-1 \le P[i,j] \le P[i-1,j]+1 \end{cases} \quad (5)$$

Powyższe własności pozwalają na wyznaczenie elementów macierzy podążając wzdłuż kolumn lub wierszy w następujący sposób:

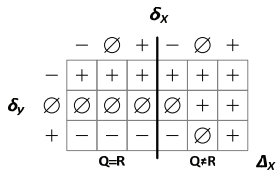
$$\begin{aligned} &\forall_{\substack{0<i\le Q_{LEN} \\ 0<j\le R_{LEN}}} P[i,j] = P[i,j-1] + \Delta x; \text{ gdzie } \Delta x \in \{-1,0,1\} \\ &\forall_{\substack{0<i\le Q_{LEN} \\ 0<j\le R_{LEN}}} P[i,j] = P[i-1,j] + \Delta y; \text{ gdzie } \Delta y \in \{-1,0,1\} \end{aligned} \quad (6)$$

Przedstawione własności pozwalają opisać macierz dopasowania w formie przyrostów w wierszu lub kolumnie. Możliwe jest wypracowanie dwóch schematów wyznaczania wartości przyrostowych na podstawie znanych przyrostów w wierszu (oznaczone jako *x*) i kolumnie (oznaczone jako *y*)

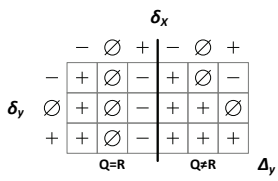


Rys. 6. Schemat blokowy komórki elementarnej SW uzyskanej w wyniku bezpośredniej implementacji
Fig. 6. Block diagram of the SW elementary cell obtained in direct implementation process

Przedstawiony proces obliczania wartości funkcji Δx i Δy reprezentujący przyrost wartości dopasowania wzdłuż wiersza lub kolumny można wykorzystać do obliczenia wartości współczynników macierzy *P*. Zmienne δx i δy opisują przyrosty współczynników w wierszu i kolumnie odpowiednio w odniesieniu do elementu A. Trzecim elementem wpływającym na wartość współczynników przyrostowych jest dopasowanie pomiędzy symbolem sekwencji przeszukiwanej $Q[i]$ oraz sekwencji odniesienia $R[j]$. Tablice prawdy do wyznaczania wartości przyrostowych Δx i Δy z wykorzystaniem kodowania symbolicznego przedstawiono na rysunkach (rys. 7, rys. 8).



Rys. 7. Tablica prawdy funkcji Δx
Fig. 7. The truth table for Δx function



Rys. 8. Tablica prawdy funkcji Δy
Fig. 8. The truth table for Δy function

Warto w tym miejscu zauważyć, że obie tablice dla części związanej z dopasowaniem i niedopasowaniem są symetryczne i spełniają następujące zależności:

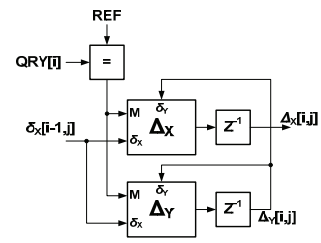
$$\begin{aligned} \Delta x &= f_x(\delta x, \delta y, M) = f_y(\delta y, \delta x, M) \\ \Delta y &= f_y(\delta x, \delta y, M) = f_x(\delta y, \delta x, M) \end{aligned} \tag{7}$$

Współczynniki przyrostowe pozwalają na łatwe wyznaczanie relacji współczynnika dopasowania. Do jakościowej oceny dopasowania sekwencji konieczne jest dokonanie przekształcenia do postaci absolutnej:

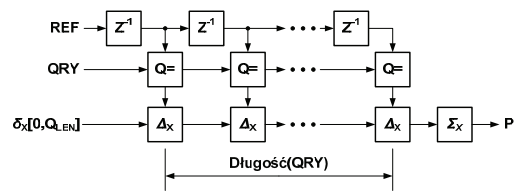
$$\begin{aligned} P[i,j] &= P[i,0] + \sum_{k=1}^{j} \delta x[i,k]; P[i,0] = i \\ P[i,j] &= P[0,j] + \sum_{l=1}^{i} \delta x[l,j]; P[0,j] = 0 \end{aligned} \tag{8}$$

5. Implementacja algorytmu SW

Możliwe są dwie alternatywne metody wyznaczania współczynnika dopasowania przebiegające wzdłuż wiersza lub kolumny. W przypadku wyznaczania położenia najlepiej dopasowanego łańcucha najkorzystniejszym sposobem jest wyznaczanie współczynnika równolegle do sekwencji przeszukiwanej (wierszowej). W prezentowanym rozwiązaniu przeszukiwana sekwencja jest umieszczona w wierszu (oś *X*). Układ będzie wyznaczał dopasowanie na podstawie przyrostu Δx. Schemat blokowy podstawowego modułu ΔSW został przedstawiony na rys. 9. Wypracowuje on wartości Δx oraz Δy. Takie rozwiązanie pozwala na otrzymanie wyniku dopasowania macierzy P poprzez sukcesywne sumowanie współczynnika przyrostowego Δx. Na rys.10 przedstawiono strukturę potokową całego układu. Umożliwia ona przetwarzanie pojedynczego symbolu w czasie jednego taktu sygnału zegarowego. Twierdzenie o zbieżności wartości dopasowania pozwala wyeliminować układ sterowania stosowany w układach przetwarzania potokowego podczas wypełniania potoku danymi.



Rys. 9. Schemat blokowy komórki elementarnej ΔSW opartej na funkcji przyrostu poziomego Δx
Fig. 9. Block diagram of the ΔSW cell based on horizontal growth function Δx



Rys. 10. Schemat blokowy potoku obliczeniowego ΔSW wraz z akumulatorem wartości *P*
Fig. 10. Block diagram of the ΔSW calculation pipe with a *P* value accumulator

Długość sekwencji zapytania określa liczbę połączonych z sobą komórek elementarnych . W odróżnieniu od implementacji opartej na bezpośredniej metodzie algebraicznej, której zakres wartości *P* jest zależny od położenia komórki obliczeniowej, w przypadku systemu przyrostowego w dowolnym miejscu łańcucha przyrostowy zakres wartości dopasowania jest identyczny i niezależny od położenia komórki. Wartość współczynnika *P* zostaje obliczona w układzie akumulacyjnym Σ. Rozmiar bitowy akumulatora jest zależny od długości sekwencji zapytania. Układ akumulacyjny jest implementowany w postaci licznika rewersyjnego.

6. Podsumowanie

Implementacja algorytmów a w szczególności ich efektywne odwzorowanie jest interesującym i wymagającym zadaniem. Języki opisu sprzętu i narzędzia syntezy logicznej mimo ponad 30 letniej tradycji i nieustannego rozwoju nie pokrywają szerokiego spektrum aspektów związanych z implementacją algorytmów. W artykule pokazano ewolucję algorytmu z ogólnej postaci algebraicznej do zoptymalizowanej formy przyrostowej ΔSW. Niezwykła prostota komórki elementarnej pozwala na implementację bardzo długich sekwencji zapytania w pojedynczym układzie. Obok długości sekwencji zapytania istotna jest szybkość przetwarzania sięgającą około 600MHz dla rodziny układów Virtex 5 [8].

Prace nad implementacją algorytmu są kontynuowane. Zespół zamierza rozszerzyć własności funkcjonalne opracowanego rozwiązania a także poddać je weryfikacji wykorzystując rzeczywiste zagadnienia z zakresu genetyki i biologii.

7. Literatura

[1] Smith T.F. and Waterman M.S.: Identification of Common Molecular Sub-sequences, Journal of Molecular Biology 147 (1981), pp. 195–197.
[2] Gusfield D.: Algorithms on strings, trees and sequences, Cambridge University Press, 1997.
[3] Yamaguchi Y., Maruyama T.: High Speed Homology Search with FPGAs, Proceedings of Pacific Symposium on Biocomputing 2002, pp. 271–282.
[4] Zhang F., Qiao X., Liu Z.: A Parallel Smith-Waterman Algorithm Based on Divide and Conquer, Proceedings of Fifth International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'02), 2002.
[5] Benkrid K., Liu Y., Benkrid A.: High Performance Biosequence Database Scanning Using FPGAs, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP Honolulu, Hawaii, USA 2007, pp. 361–364.
[6] Phoophakdee B. and Zaki M.J. (2007): Genome-scale disk-based suffix tree indexing", Proceedings of 2007 ACM SIGMOD international conference on Management of data, Beijing, China. pp. 833–844.
[7] Milik A., Pułka A.: The Reconfigurable Hardware Accelerator for Searching Genome Patterns, Proceedings of the IFAC Workshop on Programmable Devices and Embedded Systems PDeS 2009, Roznov pod Radhostem, Czech Rep. Feb.10-12, 2009, pp. 33–38.
[8] Xilinx, Virtex-5 User Guide, Xilinx, 2007.

otrzymano / received: 15.10.2010
przyjęto do druku / accepted: 01.12.2010 *artykuł recenzowany*

INFORMACJE

XVII MIĘDZYNARODOWE SEMINARIUM METROLOGÓW

„Metody i Technika Przetwarzania Sygnałów w Pomiarach Fizycznych"
Gdańsk – Karlskrona, 20 – 22 października 2011

Z przyjemnością informujemy, że w dniach 20 – 22 października 2011r po dwóch latach przerwy planujemy zorganizować kolejne XVII Międzynarodowe Seminarium Metrologów *„Metody i Technika Przetwarzania Sygnałów w Pomiarach Fizycznych"*. Celem seminarium jest prezentacja prac naukowo-badawczych dotyczących obszernej i ważnej tematyki - przetwarzania sygnałów w szeroko rozumianych pomiarach fizycznych. W seminarium tradycyjnie uczestniczą przedstawiciele instytucji naukowych z Polski i zagranicy, a także specjaliści praktycy nastawieni na aplikacje inżynierskie w zakładach przemysłowych.

Zapraszamy do czynnego udziału w MSM'2011 i zgłaszania referatów. Oczekujemy, że kolejne seminarium będzie ważnym wydarzeniem stymulującym rozwój prac naukowo-badawczych z zakresu przetwarzania sygnałów, a także integrującym środowisko metrologów.

Tematyka Seminarium:

1. Czujniki i przetworniki pomiarowe
2. Przetwarzanie sygnałów
3. Systemy informacyjno-pomiarowe
4. Dydaktyka metrologii

Patronat:

Komitet Metrologii i Aparatury Naukowej PAN

JM Rektor Politechniki Rzeszowskiej

Prezydent Miasta Rzeszowa

Organizatorzy:

Politechnika Rzeszowska
Katedra Metrologii i Systemów Diagnostycznych

Lviv Polytechnic National University
Department of Information Measuring Technology

Politechnika Gdańska
Katedra Metrologii i Systemów Informacyjnych

Blekinge Institute Of Technology
School Of Engineering Karlskrona (Szwecja)

Adres do korespondencji:

MSM'2011
Politechnika Rzeszowska
Wydział Elektrotechniki i Informatyki
Katedra Metrologii i Systemów Diagnostycznych
ul. W. Pola 2, 35-959 Rzeszów
tel: (+48) 17-865-1438, 17-865-1575, 17-865-1231
fax: (+48) 17-865-1575
e-mail: kmisd@prz.edu.pl

Szczegółowe informacje dotyczące programu, terminów i publikacji prac dostępne są na stronie internetowej MSM'2011:

http://www.prz.edu.pl/msm11

Przewodniczący Komitetu Naukowego
prof. dr hab. inż. Bohdan Stadnyk
Lviv Polytechnic National University

Przewodniczący Komitetu Organizacyjnego
dr hab. inż. Adam Kowalczyk, prof. PRz
Politechnika Rzeszowska