

Sebastian PAWLAK

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Zastosowanie układu FPGA do akceleracji obliczeń całki korelacji interwałów RR

Mgr inż. Sebastian PAWLAK

Autor artykułu ukończył studia na kierunku Elektrotechnika, specjalizacji Inżynieria Systemów Komputerowych na Uniwersytecie Zielonogórskim, gdzie pracuje obecnie na stanowisku asystenta. Przygotowuje się także do otwarcia przewodu doktorskiego. Głównym obszarem zainteresowań badawczych Autora jest wykorzystanie układów FPGA do akceleracji obliczeń w systemach przetwarzania obrazów oraz w analizie sygnałów biomedycznych.



e-mail: S.Pawlak@iie.uz.zgora.pl

Streszczenie

Analiza sygnałów biomedycznych jest jednym z ważniejszych obszarów badań naukowych. Istnieje duże zapotrzebowanie na wydajne systemy obliczeniowe, które pozwalałyby na analizę sygnałów biomedycznych wielu pacjentów. W artykule przedstawiono propozycję architektury systemu obliczającego całkę korelacji interwałów międzyuderzeniowych RR, wykorzystującego układ FPGA do akceleracji obliczeń.

Słowa kluczowe: akceleracja obliczeń, całka korelacji, interwał RR, układy FPGA.

FPGA-accelerated calculation of the correlation integral of RR intervals

Abstract

Biomedical signal analysis is a very important field of research. This paper describes architecture of FPGA-accelerated computational system proposed by Author. The main purpose of this system is calculation of the correlation integral of RR intervals in cardiological patients.

Keywords: acceleration of computation, correlation integral, RR interval, FPGA devices.

1. Wstęp

W ostatnich latach daje się zaobserwować znaczny wzrost zainteresowania analizą sygnałów biomedycznych. Wielu naukowców, w liczących się ośrodkach badawczych, podejmuje próby odnajdywania charakterystycznych cech tych sygnałów oraz ich powiązania ze stanem zdrowia pacjentów.

Jednym z podstawowych zabiegów diagnostycznych jest elektrokardiografia (ECG). Szybko rozwijającą się dziedziną badań, bazującą na elektrokardiografii, jest analiza zmienności rytmu serca (HRV), w której jednym z etapów jest obliczanie całki korelacji interwałów międzyuderzeniowych RR [1].

2. Interwał RR

Interwał międzyuderzeniowy RR jest odstępem czasowym pomiędzy szczytami kolejnych załameków R elektrokardiogramu. Przykładowy elektrokardiogram z zaznaczonym interwałem RR przedstawiono na rys. 1.

Typowy elektrokardiogram pozwala zmierzyć czas trwania interwału RR z dokładnością do 1ms.

3. Całka korelacji

W teorii chaosu, całka korelacji jest średnim prawdopodobieństwem, że stany w przestrzeni fazowej, w dwóch różnych chwilach czasu, są sobie bliskie [2]. Całka korelacji wyrażona jest wzorem (1).



Rys. 1. Elektrokardiogram z zaznaczonym interwałem RR

Fig. 1. Electrocardiogram with RR interval

$$C_m(\varepsilon) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \times \sum_{i,j=1}^N \Theta(\varepsilon - \|\vec{x}_i - \vec{x}_j\|), \vec{x}_i \in R^m \quad (1)$$

m – liczba wymiarów przestrzeni fazowej, ε – odległość progowa, N – liczba stanów $\vec{x}_i$, $\|\cdot\|$ – norma euklidesowa, Θ – funkcja skokowa Heaviside'a.

Funkcja skokowa Heaviside'a wyrażona jest wzorem (2).

$$\Theta(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (2)$$

Norma euklidesowa wyrażona jest wzorem (3).

$$\|\vec{x}_i - \vec{x}_j\| = \sqrt{\sum_{k=1}^m (x_{i,k} - x_{j,k})^2} \quad (3)$$

Gdy dostępny jest jedynie szereg czasowy y , wówczas przestrzeń fazowa może być zrekonstruowana za Twierdzeniem Takens'a [3] zgodnie ze wzorem (4).

$$\vec{x}_i = (y_i, y_{i+\tau}, y_{i+2\tau}, \dots, y_{i+(m-1)\tau}) \\ i = 1, \dots, N - (m-1)\tau \quad (4)$$

τ – odstęp czasowy, N – liczba elementów szeregu czasowego.

W praktyce szereg czasowy ma skończoną długość. Wówczas całka korelacji może być przybliżona sumą korelacji (5).

$$C_m(\varepsilon) = \frac{1}{[N - (m-1)\tau][N - (m-1)\tau - 1]} \times \\ \times \sum_{i=1}^{N - (m-1)\tau} \sum_{j=1}^{N - (m-1)\tau} \Theta(\varepsilon - \|\vec{x}_i - \vec{x}_j\|) \quad (5)$$

Można zauważyć, że obliczenie całki korelacji pociąga za sobą liczbę operacji proporcjonalną do kwadratu długości szeregu czasowego.

Na rys. 2 przedstawiono kod źródłowy funkcji obliczającej całkę korelacji (ANSI C).

```

double
ci(int* rr_ptr, int n, int m, int e, int tau)
{
    int i, j, k;
    int sum;
    int enorm2;
    int e2 = e * e;
    int corrinteg = 0;
    int range = n - (m - 1) * tau - 1;

    for (i = 0; i < range; i++)
    {
        for (j = 0; j < range; j++)
        {
            enorm2 = 0;
            for (k = 0; k < m; k++)
            {
                sum = rr_ptr[i + k * tau] -
                    rr_ptr[j + k * tau];
                enorm2 += sum * sum;
            }
            if (e2 > enorm2) corrinteg++;
        }
    }
    return((double)corrinteg /
        (double)((range + 1) * range));
}

```

Rys. 2. Kod źródłowy funkcji obliczającej całkę korelacji
Fig. 2. Source code of function computing correlation integral

4. Akceleracja obliczeń całki korelacji

Programowa realizacja obliczeń całki korelacji wiąże się z przesłaniem LP_{SOFT} (6) interwałów RR z pamięci przechowującej szeregi czasowe do układu arytmetycznego.

$$LP_{SOFT} = 2mN^2 \quad (6)$$

Jest to związane z faktem, że norma euklidesowa liczona jest dla każdej pary wektorów x_i i x_j , a wektory te składają się z m -interwałów RR.

W tabeli 1 przedstawiono numery elementów wektora interwałów RR, dla kolejnych iteracji pętli, w której obliczana jest całka korelacji (dla $\tau = 4$, $m = 7$).

Tab. 1. Indeksy elementów wektora x_j ($\tau = 4$, $m = 7$)

Tab. 1. Indexes of x_j vector elements ($\tau = 4$, $m = 7$)

j	j+ τ	j+2 τ	j+3 τ	j+4 τ	j+5 τ	j+6 τ	j+7 τ
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31
4	8	12	16	20	24	28	32
5	9	13	17	21	25	29	33
6	10	14	18	22	26	30	34
7	11	15	19	23	27	31	35
8	12	16	20	24	28	32	36
9	13	17	21	25	29	33	37
10	14	18	22	26	30	34	38
11	15	19	23	27	31	35	39
12	16	20	24	28	32	36	40
13	17	21	25	29	33	37	41
14	18	22	26	30	34	38	42

Można zauważyć, że wektor $x_{j+\tau}$ zawiera $(m-1)$ elementów pokrywających się z elementami wektora x_j oraz jeden dodatkowy element $x_{j+(m-1)\tau}$. Dzięki temu, możliwa jest organizacja układu obliczającego funkcję Heaviside'a z różnicy odległości progowej ε i normy euklidesowej wektorów x_i i x_j , jako układ zawierający dwa wielobitowe rejestry przesuwające. Taka organizacja układu arytmetycznego pozwala na zmniejszenie liczby przesłań z pamięci przechowującej wektor interwałów RR do układu obliczającego całkę korelacji, w porównaniu z realizacją programową. Wówczas liczba przesłań LP_{HARD} wyraża się wzorem (7).

$$LP_{HARD} = \tau N^2 \quad (7)$$

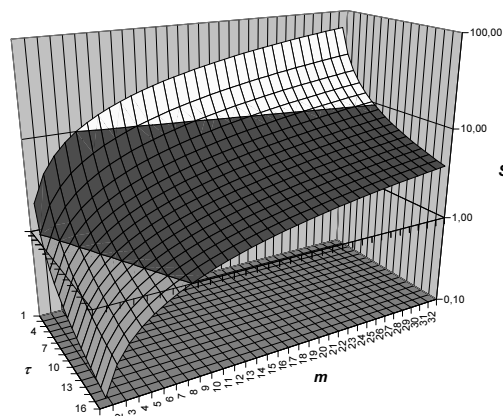
Akceleracja obliczeń S (z ang. *speed-up*) związana z realizacją sprzętową algorytmu obliczania całki korelacji, wyrażona jest poprzez stosunek liczby operacji rozwiązania programowego, do liczby operacji rozwiązania sprzętowego.

Liczba operacji w obliczaniu całki korelacji proporcjonalna jest do liczby przesłań pomiędzy pamięcią, a modułem obliczającym funkcję Heaviside'a z różnicy odległości progowej ε i normy euklidesowej wektorów x_i i x_j .

W związku z powyższym, akceleracją S można wyrazić stosunkiem liczby przesłań LP_{SOFT} do liczby przesłań LP_{HARD} , zgodnie ze wzorem (8).

$$S = \frac{LP_{SOFT}}{LP_{HARD}} = \frac{2mN^2}{\tau N^2} = \frac{2m}{\tau} \quad (8)$$

Dla każdego szeregu czasowego interwałów międzyuderzeniowych RR, konieczne jest obliczenie całki korelacji dla $1 \leq \tau \leq 16$, $1 \leq m \leq 32$. Daje to zestaw 512 całek korelacji dla każdej wartości odległości progowej ε . Na rys. 3 przedstawiono wykres akceleracji S w funkcji parametrów τ i m .



Rys. 3. Wykres akceleracji $S = f(\tau, m)$

Fig. 3. Chart of acceleration $S = f(\tau, m)$

Można zauważyć, że dla pewnych wartości parametrów m i τ , akceleracja S przyjmuje wartości mniejsze od jedności, co oznacza zmniejszenie wydajności w stosunku do rozwiązania programowego.

5. Założenia systemu

System powinien umożliwiać obliczanie całki korelacji na podstawie elektrokardiogramu 24-godzinne. Założono, że jest mało prawdopodobne długotrwałe utrzymywanie się rytmu serca pacjenta powyżej trzech uderzeń na sekundę. W związku z tym maksymalna liczba interwałów RR w ciągu doby została określona jako 2^{18} .

Ponadto założono, że interwał międzyuderzeniowy RR nie będzie dłuższy niż 2^{15} ms, co daje ponad 32s.

W tabeli 2 przedstawiono zakresy poszczególnych parametrów projektowanego systemu.

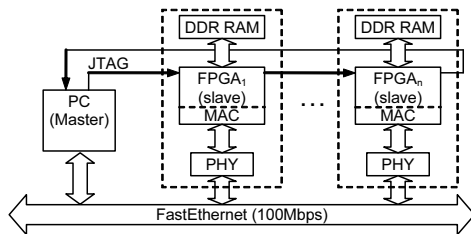
Tab. 2. Zakres parametrów systemu

Tab. 2. Range of system parameters

Parametr	Zakres	
	od	do
N	-	$2^{18} = 262\,144$
m	1	$2^5 = 32$
τ	1	$2^4 = 16$
ε	1	$2^{14} = 16\,384$

6. Architektura systemu

Na rys. 4 przedstawiono architekturę systemu. Projektowany system składa się z komputera PC (węzła nadrzędnego) oraz n -węzłów obliczeniowych (podrzędnych).



Rys. 4. Schemat blokowy systemu
Fig. 4. Block diagram of system

Każdy węzeł podrzędny zawiera układ FPGA, dedykowaną pamięć i kontroler sieci FastEthernet.

Komputer pełni rolę zarządzającą. Do jego zadań należy: - konfiguracja układów FPGA, - ładowanie zestawów szeregów czasowych poszczególnych pacjentów do węzłów obliczeniowych, - pobieranie wyników obliczeń z węzłów obliczeniowych.

Zadaniem węzła podrzędnego jest obliczanie całki korelacji dla załadowanego do niego szeregu czasowego interwałów RR oraz parametrów ε , τ , m . Ponieważ czas obliczania całki korelacji jest znacznie dłuższy od czasu ładowania szeregów czasowych do węzłów obliczeniowych, interfejs komunikacyjny FastEthernet nie stanowi „wąskiego gardła” systemu.

Nowoczesne układy FPGA zawierają w swojej strukturze poza elementami LUT (z ang. Look-at-Table) również dedykowane bloki pamięci, a także sprzętowe układy mnożące – tzw. mnożarki [4, 5]. Przy projektowaniu układu akceleratora obliczeń istotne jest efektywne wykorzystanie zasobów sprzętowych układu FPGA. Duże znaczenie ma pozbycie się operacji wymagających dużej ilości zasobów sprzętowych oraz takich, których czas wykonywania jest długi.

Do budowy systemu wybrany został dostępny autorowi układ FPGA Xilinx Virtex-II Pro (xcvp30-7ff896c). Zawiera on w swojej strukturze między innymi 136 sprzętowych układów mnożących. Operacja pierwiastkowania (przy obliczaniu normy euklidesowej wektorów x_i i x_j) pochłaniała by wiele zasobów sprzętowych, w związku z czym podjęta została decyzja o jej wyeliminowaniu zgodnie ze wzorem (9).

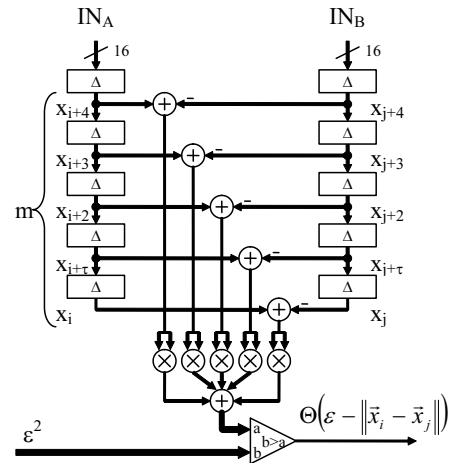
$$\Theta(\varepsilon - \|\tilde{x}_i - \tilde{x}_j\|) = \begin{cases} 0, & \varepsilon - \|\tilde{x}_i - \tilde{x}_j\| \leq 0 \\ 1, & \varepsilon - \|\tilde{x}_i - \tilde{x}_j\| > 0 \end{cases} = \begin{cases} 0, & \varepsilon \leq \|\tilde{x}_i - \tilde{x}_j\| \\ 1, & \varepsilon > \|\tilde{x}_i - \tilde{x}_j\| \end{cases} \quad (9)$$

$$= \begin{cases} 0, & \varepsilon^2 \leq \sum_{k=1}^m (x_{i,k} - x_{j,k})^2 \\ 1, & \varepsilon^2 > \sum_{k=1}^m (x_{i,k} - x_{j,k})^2 \end{cases}$$

Na rys. 5 przedstawiono schemat blokowy modułu akceleratora obliczającego funkcję Heaviside’a z różnicy odległości progowej ε i normy euklidesowej wektorów x_i i x_j .

Układ akceleratora zbudowany jest z zestawu rejestrów wektora x_i , zestawu rejestrów wektora x_j , m -sumatorów 16-bitowych, m -układów mnożących, sumatora $(16 + m)$ -bitowego oraz komparatora.

Działanie akceleratora rozpoczyna się od załadowania do zestawu rejestrów wektora x_i . Następnie uzupełniane są rejestry wektora x_j . Kolejnym krokiem jest obliczenie różnic elementów wektorów x_i i x_j . Otrzymane sumy podnoszone są do kwadratu (z wykorzystaniem sprzętowych układów mnożących).



Rys. 5. Schemat blokowy akceleratora sprzętowego
Fig. 5. Block diagram of hardware accelerator module

Kwadraty różnic podawane są na wejście sumatora, który oblicza kwadrat normy euklidesowej wektorów x_i i x_j . Kolejnym etapem jest jej porównanie z kwadratem odległości progowej ε . Gdy kwadrat odległości progowej ε jest większy od kwadratu normy euklidesowej wektorów x_i i x_j , wówczas jednobitowe wyjście komparatora (wyznaczającego wynik funkcji Heaviside’a) przyjmuje stan „1” logicznej, w przeciwnym wypadku – stan „0” logicznej.

7. Podsumowanie

Zrównoleglenie wykonywania operacji, zastosowanie specjalizowanych bloków takich jak sprzętowe mnożarki oraz optymalizacja dostępu do pamięci, powinny prowadzić do wzrostu wydajności systemu obliczeniowego zbudowanego z wykorzystaniem układów FPGA w stosunku do systemu wykorzystującego procesor ogólnego przeznaczenia. Szczególnie korzystny jest współczynnik liczby wykonywanych operacji do częstotliwości zegara systemu. Mimo to, szybki rozwój procesorów wielordzeniowych oraz nowe procesy technologiczne zmniejszają tę przewagę układów FPGA. Najszybsze układy FPGA - Xilinx Virtex 5 osiągają częstotliwości pracy 550MHz, podczas gdy procesory dawno przekroczyły częstotliwość 3GHz.

Kolejnym etapem badań będzie implementacja systemu na platformie sprzętowej Xilinx Virtex-II Pro Development System (XUPV2P) i weryfikacja jego działania.

8. Literatura

- [1] Carvajal R., Wessel N., Vallverdu M., Caminal P., Voss A.: Correlation dimension analysis of heart rate variability in patients with dilated cardio-myopathy, Computer Methods and Programs in Biomedicine, Issue 78 2003, 133-140.
- [2] Grassberger P., Procaccia I.: Measuring the strangeness of strange attractors, Physica D: Nonlinear Phenomena, Volume 9, Issue 1-2, Elsevier 1983, 189-208.
- [3] Takens F.: Detecting strange attractors in turbulence. Dynamical Systems and Turbulence, Lecture Notes in Mathematics 898, Springer-Verlag, Warwick 1980, 366-381.
- [4] Xilinx: Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet, Xilinx, DS083 (v.4.6) March 5, 2007.
- [5] Xilinx: Virtex-5 Family Overview – LX, LXT and SXT Platforms, Xilinx, DS100 (v.3.0) February 2, 2007.