

Marek POPLAWSKI, Michał BIAŁKO
POLITECHNIKA KOSZALIŃSKA, WYDZIAŁ ELEKTRONIKI I INFORMATYKI

Implementacja cyfrowego systemu rozmytego w układzie FPGA

Mgr inż. Marek POPLAWSKI

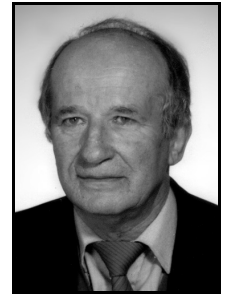
Ukończył studia na Wydziale Elektroniki i Informatyki Politechniki Koszalińskiej (2005). Jest doktorantem w Katedrze Inżynierii Komputerowej Politechniki Koszalińskiej. Jego zainteresowania naukowe obejmują zagadnienia dotyczące systemów rozmytych, systemów wbudowanych a także programowania.



e-mail: milla@przystan.pl

Prof. dr hab. inż. Michał BIAŁKO

Jest pracownikiem naukowym Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej. Jest członkiem rzeczywistym PAN. Otrzymał tytuł doktora h.c. INP de Toulouse. Jego zainteresowania naukowe obejmują zagadnienia dotyczące m.in. układów elektronicznych, sztucznej inteligencji, systemów ekspertowych systemów rozmytych a także algorytmów genetycznych.



e-mail: mibialko@ue.eti.pg.gda.pl

Streszczenie

W artykule przedstawiono praktyczną implementację cyfrowego systemu rozmytego w układzie FPGA. Poszczególne zbiory rozmyte posiadają swój kod, który wykorzystywany jest do adresowania pamięci przechowującej reguły. Wykorzystana technika adresowania znacznie przyspiesza proces obliczeń w porównaniu do szeregowego przetwarzania reguł.

Słowa kluczowe: zbiór rozmyty, logika rozmyta, rozmywanie, wnioskowanie rozmyte, system rozmyty, architektura cyfrowego systemu rozmytego.

Implementation of digital fuzzy system in FPGA circuit

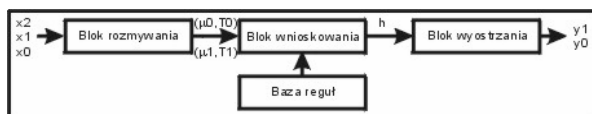
Abstract

The paper presents an implementation of a digital fuzzy system in FPGA circuit. The input fuzzy sets are coded as 3-bit numbers which are used to address the memory containing fuzzy rules. The applied address technique significantly accelerates computation compared to serial rule processing.

Keywords: fuzzy set, fuzzy logic, fuzzification, fuzzy system, digital architecture of fuzzy system.

1. Wprowadzenie

Celem pracy jest przedstawienie sprzętowej implementacji cyfrowego systemu rozmytego [1, 2] (rys. 1) w układzie reprogramowalnym (Spartan3)[3]. Proponowany system składa się z 3 wejść i 2 wyjść, którego ogólną strukturę blokową pokazano na rys. 1. System taki może mieć zastosowanie np. w sterowaniu elektrycznym wózkiem inwalidzkim; dlatego na etapie testowania systemu przyjęto odpowiednie kształty i rozmieszczenie termów zmiennych lingwistycznych jak na rys. 2.

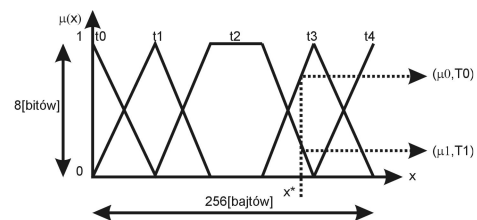


Rys. 1. Ogólny schemat blokowy systemu rozmytego (μ_0, μ_1 – wartości funkcji przynależności T_0, T_1 – kody termów, h – wartości funkcji przynależności termów wyjściowych)

Fig. 1. General block diagram of fuzzy system (μ_0, μ_1 – values of membership function, T_0, T_1 – fuzzy set codes, h – values of membership function of output terms)

Wejściowe zmienne lingwistyczne (rys. 2) opisano 5 termami przy pomocy 256 dyskretnych próbek, przy 8-bitowej rozdzielczości. Przy rozmieszczeniu termów (t_0 - t_4), jak na rys. 2, uaktywnianych wartością dyskretnej x^* zmiennej wejściowej odczytywane są wartości funkcji przynależności dwóch zbiorów. W przypadku, gdy jedna z wartości funkcji przynależności przyjmuje wartość zerową ($\mu=0$), to w procesie wnioskowania

zostaje ona pominięta. Zmienne wyjściowe reprezentowane są przez 5 termów w postaci singletonów. Pojedyncza reguła złożona jest z 3 zakodowanych przesłanek, które połączone są operatorami koniunkcji [4]. Dopuszczalna liczba reguł opisująca ten system wynosi $2 * 5^3 = 250$. Operacja wnioskowania przebiega etapowo; najpierw po rozmywaniu (fuzyfikacji) wielkości wejściowych obliczane są stopnie spełnienia (punkty zapłonu) poszczególnych reguł, a później po implikacji dokonywana jest agregacja (sumowanie konkluzji reguł). Proces wyostżania (defuzyfikacji) realizowany jest z wykorzystaniem metody środka ciężkości dla singletonów.



Rys. 2. Przykład wejściowej zmiennej lingwistycznej (t_0 - t_4 – zbiory rozmyte (termy), x^* – przykładowa ostra wartość wejściowa, μ_0, μ_1 – wartości funkcji przynależności aktywowanych zbiorów, T_0, T_1 – kody aktywowanych zbiorów)

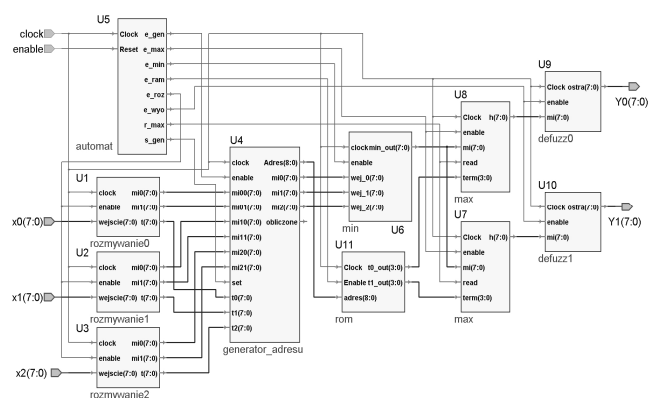
Fig. 2. Example of input linguistic variable (t_0 - t_4 – fuzzy sets, x^* – example of non-fuzzy input variable, μ_0, μ_1 – values of membership functions of activated fuzzy sets, T_0, T_1 – codes of activated sets)

2. Zasada działania

Głównym elementem cyfrowego systemu rozmytego (rys. 3) jest blok „automat”, który realizuje funkcje sterujące. Zadaniem tego bloku jest odpowiednie sekwencyjne dołączanie sygnałów logicznych z wyjść: e_{gen} , e_{max} , e_{min} , e_{ram} , e_{roz} , e_{wyo} , r_{max} , s_{gen} do bloków wykonawczych: „rozmywaniex” (gdzie $x=0,1,2$), „adres_generator”, „min”, „max”, „rom” oraz „defuzzx” (gdzie $x=0,1$). Blok „automat” kontrolowany jest przez taktowany sygnałem zegara, wewnętrzny licznik L_a którego wartości powodują dołączanie wspomnianych wyżej wyjść do odpowiednich bloków inicjując ich działanie.

Proces rozmywania (fuzyfikacji) polega na zamianie wejściowej wartości z dziedziny liczb rzeczywistych na wartość μ z dziedziny zbiorów rozmytych. W proponowanym systemie rozmytym proces rozmywania sprowadza się do odczytywania z pamięci systemu, na podstawie 8-bitowej ostrej wartości wejściowej, wartości funkcji przynależności aktywowanych zbiorów rozmytych, a także kodów tych zbiorów. Ze względu na to, że w systemie występują 3 kanały wyjściowe, blok wejściowy został podzielony na trzy niezależne kanały rozmywające zbudowane w oparciu o bloki „rozmywaniex” (gdzie $x=0,1,2$). Pojawienie się wysokiego stanu na wejściu „enable” takiego bloku powoduje rozpoczęcie procesu rozmywania. Na podstawie dyskretnej wartości sygnału wejściowego - „wejście(7:0)”

generowany jest adres przechowujący właściwą dyskretną próbkę funkcji przynależności zapisaną w pamięci bloku „rozmywaniex” (gdzie $x=0,1,2$), przy czym taka próbka zawiera dwie ośmiobitowe wartości funkcji przynależności (μ_0, μ_1) oraz dwa 3 bitowe kody zbiorów rozmytych (T0,T1). W pamięci bloku „rozmywaniex” (gdzie $x=0,1,2$) przechowywane jest łącznie 256 takich próbek opisujących wejściową zmienną lingwistyczną. Odczytanie z pamięci 3 bajtowej wartości próbki oraz jej wystawienie na wyjściach mi0, mi1, t bloku „rozmywaniex” (gdzie $x=0,1,2$) kończy operacje rozmywania.



Rys. 3. Struktura blokowa proponowanego cyfrowego systemu rozmytego
Fig. 3. Block structure of proposed digital fuzzy system

Proces wnioskowania dokonywany jest w blokach „generator_adresu”, „min”, „rom” oraz „max” (rys. 3). Wysoki stan linii sterującej „e_gen” pochodzącej z bloku – automat rozpoczyna proces wnioskowania. Do bloku „generator_adresu” wprowadzane są, poprzez ośmiobitowe wejścia w zakresie tx(2:0) oraz tx(6:4) (gdzie $x=0,1,2$) 3 bitowe kody aktywnych zbiorów rozmytych, oraz przez wejścia mix0, mix1 (gdzie $x=0,1,2$), odpowiadające im funkcje przynależności. Elementem sterującym pracą tego bloku jest 3 bitowy wewnętrzny licznik Lp, którego wartości wag – b2, b1, b0 modyfikowane są sygnałem zegara (wejście clock). Wagi te odpowiadają wejściowym zmiennym lingwistycznym x_2, x_1, x_0 ; przy czym wartości wagi „0” odpowiadają – przyłączeniu wartości μ_0 oraz kodu T0, a „1” odpowiada – przyłączeniu wartości μ_1 oraz kodu T1. Na podstawie wartości wag – b2, b1, b0 następuje sekwencyjne łączenie kodów aktywnych zbiorów rozmytych w zakresie tx(2:0) oraz tx(6:4) (gdzie $x=0,1,2$) tworząc 9 bitowy adres konkluzji (wyjście adres(8:0)) oraz równoległe dołączane są wejścia mix0, mix1 (gdzie $x=0,1,2$) do wyjść mix (gdzie $x=0,1,2$). Dla przykładu wygenerowanie poprzez licznik wartości 001, spowoduje dołączenie kodów z wejść o następującym zakresie t0(2:0), t1(2:0), t2(6:4) do wyjścia adres(8:0) oraz odpowiednio wejść mi00, mi10, mi21 do wyjścia mix (gdzie $x=0,1,2$). Tabela 1 przedstawia kolejne wartości wag licznika Lp – b2, b1, b0 na podstawie których odczytano z wejść bloku „generator_adresu” kody wejściowych zbiorów rozmytych w zakresie tx(2:0) i tx(6:4) (gdzie $x=0,1,2$) oraz odpowiadające im wartości funkcji przynależności z wejść mix0, mix1 (gdzie $x=0,1,2$). Wyniki podane w tabeli 1 otrzymano na podstawie następujących wartości kodów wejściowych zmiennych lingwistycznych oraz ich funkcji przynależności: $x_0 - (T0=100, \mu_0=6; T1=011, \mu_1=94)$, $x_1 - (T0=000, \mu_0=84; T1=001, \mu_1=16)$, $x_2 - (T0=010, \mu_0=100; T1=011, \mu_1=0)$.

Sformułowanie adresu konkluzji powoduje uruchomienie bloku 8 - bitowej pamięci reguł - „rom” o rozmiarze 512 bajtów, przez blok „automat”. W pamięci tej zapisane są kody konkluzji reguł pod ściśle określonym adresem, którego wartości odpowiadają przesłankom aktywowanych reguł. Odczytanie wskazanej przez sformułowany adres konkluzji powoduje odczytanie dwóch 3 bitowych kodów wyjściowych singletonów odpowiednio dla kanałów y0 oraz y1, których wartości pojawiają się na wyjściach t0 oraz t1 bloku – „rom”.

Tab. 1. Wartości wag licznika Lp bloku „generator_adresu”
Tab. 1. Values of state counter Lp in “generator_adresu” block

b2	b1	b0	mi2	mi1	mi0	Odczytane kody zbiorów			Adres konkluzji
0	0	0	100	84	6	010	000	100	010000100
0	0	1	100	16	94	010	000	011	010000011
0	1	0	100	84	6	010	001	100	010001100
0	1	1	100	16	94	010	001	011	010001011
1	0	0	0	84	6	011	000	100	011000100
1	0	1	0	16	94	011	000	011	011000011
1	1	0	0	84	6	011	001	100	011001100
1	1	1	0	16	94	011	001	011	011001011

Równoległe do operacji odczytu kodów konkluzji reguł z pamięci bloku „rom” inicjowane jest działanie bloku „min” przez blok „automat”. Blok – „min” realizuje operacje typu MIN, która polega na wyborze najmniejszej wartości funkcji przynależności pochodzących z wejść wej_x (gdzie $x=0,1,2$). Wybór tej wartości przebiega w dwóch etapach; najpierw porównywane są między sobą wartości z wejść wej_0, wej_1 a następnie porównywany jest wynik tego działania z wartością z wejścia wej_2. Wynik operacji typu MIN przekazywany jest do wyjścia „min_out” bloku „min”.

Zainicjowanie działania bloków „max” przez blok „automat” powoduje rozpoczęcie operacji typu MAX, która polega na wyborze pomiędzy wartością MIN (blok – „min”), uzyskaną w drodze aktualnych obliczeń, a wartością uzyskaną w poprzedniej iteracji. W ten sposób przebiega operacja częściowej agregacji. Uzyskane w ten sposób wyniki obliczeń tj. wartości funkcji przynależności wyjściowych zbiorów rozmytych (wyjście h) przekazywane są do bloku defuzzx (gdzie $x=0,1$).

Zadaniem bloku „defuzzx” (gdzie $x=0,1$) jest uzyskanie, na podstawie wartości funkcji przynależności zbiorów wyjściowych wyniku, w postaci konkretnej liczby rzeczywistej. W procesie wyostrzenia (defuzyfikacji) wykorzystano metodę środka ciężkości dla singletonów:

$$\tilde{y} = \frac{\sum_{i=1}^n h_i * y_i}{\sum_{i=1}^n h_i} \quad (1)$$

gdzie \tilde{y} - wyostrzona wartość zmiennej wyjściowej, h_i - wartości funkcji przynależności wyjściowych singletonów, y_i - wartości położenia wyjściowych singletonów.

Sprzętowa realizacja procesu wyostrzenia (defuzyfikacji) przebiega równoległe dla dwóch kanałów wyjściowych w blokach defuzzx (gdzie $x=0,1$). Proces wyostrzenia (defuzyfikacji) rozpoczyna wystawienie wysokiego stanu na wejściu „enable” przez blok „automat”. Kolejne wartości zegara powodują przepisywanie z bloku „max” wartości funkcji przynależności poszczególnych singletonów wyjściowych. Wartości te mnożone są z wartością położenia singletona przy czym jednocześnie wykonywana jest częściowa operacja sumowania zarówno licznika jak i mianownika (wzór 1). Wykonanie 5 iteracji powoduje zakończenie procesu obliczania sum częściowych, po czym następuje operacja dzielenia. Całkowitym zakończeniem procesu wyostrzenia (defuzyfikacji), jest uzyskanie wartości \tilde{y} na wyjściach y0, y1.

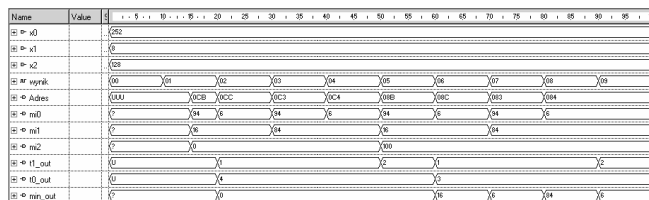
3. Podsumowanie

Proponowany cyfrowy system rozmyty opisano przy pomocy języka opisu sprzętu –VHDL, a następnie zasymulowano jego działania w środowisku Activ- Hdl firmy Aldec. Testowanie proponowanego systemu rozmytego przeprowadzono w oparciu o W tym celu w pamięci systemu zapisano wartości funkcji przynależności zbiorów rozmytych a także tablicę reguł. Następnie wprowadzono testowe (ostre) wartości wejściowych zmiennych lingwistycznych x_0, x_1, x_2 (tab. 2).

Na podstawie testowych wartości wejściowych (tab. 2) przeprowadzono symulację procesu wnioskowania proponowanego cyfrowego systemu rozmytego (rys. 4).

Tab. 2. Przykładowe ostre oraz rozmyte wartości zmiennych wejściowych x0, x1, x2
 Tab. 2. Examples of non-fuzzy and fuzzy values of input variables x0, x1, x2

Wej. zmienna lingwistyczna	Ostra wartość wejściowa	Rozmyte wartości wejściowe		Kody aktywowanych zbiorów	
		μ_0	μ_1	T0	T1
x0	252	6	94	100	011
x1	8	84	16	000	001
x2	128	100	0	010	011



Rys. 4. Fragment wyników symulacji bloku wnioskowania proponowanego cyfrowego systemu rozmytego

Fig. 4. Part of simulation process in rule inference block of the proposed fuzzy system

Znaczenie zmiennych użytych w symulacji (rys. 4) jest następujące: „x0”, „x1”, „x2” – wejściowe zmienne lingwistyczne, „wynik” – kolejne stany licznika La bloku „automat”, „adres” – adresy konkluzji wygenerowane przez blok „generator_adresu”, „mi0”, „mi1”, „mi2” – wartości funkcji przynależności wygenerowane przez blok „generator_adresu”, „t0_out, t1_out” – odczytane kody konkluzji z bloku „rom”, „min_out” – wynik działania bloku „min”. Wyniki symulacji potwierdziły poprawne działanie bloków: „generator_adresu”, „min”, „rom” oraz „max”.

W drugim etapie testowania proponowanego systemu rozmytego zaprogramowano układ Spartan 3 znajdujący się na płycie testowej[5]. Wyniki pomiarów ponownie potwierdziły

prawidłowe działanie systemu, którego czas przetwarzania wyniósł $t_p=0,6[\mu s]$ przy częstotliwości zegara taktującego 50[Mhz]. Dodatkowo na podstawie prostych obliczeń określono szybkość działania proponowanego systemu rozmytego tj. liczbę reguł przetwarzanych w jednostce czasu. W proponowanym systemie liczba ta wynosiła $2*8$ reguł (przy $t_p=0,6\mu s$), co daje szybkość przetwarzania $2*13,3$ [MFLIPS] (ang. „mega fuzzy logic inferences per second”).

Wykorzystanie techniki adresowania [6] w porównaniu do szeregowego przetwarzania reguł w procesie wnioskowania, pozwala bardzo szybko wybrać tylko aktywne reguły z bazy reguł. W tej technice istnieje konieczność umieszczania poszczególnych reguł, a także ich konkluzji, pod ściśle określonym adresem w pamięci systemu co powoduje zwiększenie nakładu pracy podczas przygotowywania odpowiedniej bazy reguł.

4. Literatura

- [1] R.R. Yager, D.Filev, Podstawy modelowania i sterowania rozmytego, WNT, Warszawa 1995.
- [2] M. J. Patyra, J. L. Grantner, „Hardware implementaions of digital fuzzy logic controller”, Elsevier Information Science 113 (1999) pp. 19-54.
- [3] Xilinx DS099 Spartan-3 Complete data sheet.
- [4] M. Białko, „Sztuczna inteligencja i elementy hybrydowych systemów rozmytych”, Wydawnictwo Uczelniane PK Koszalin 2005.
- [5] Altium LiveDesign – dokumentacja “LiveDesign_Eval_Board_Tech_Ref_Manual.pdf”
- [6] D. Falchieri, A. Gabrielli, E. Gandolfi, M. Masetti, “Very fast VLSI fuzzy processor: 2 inputs loutput”, Fuzzy Sets & Systems, November 2002, Volume/Issue 132/2 pp. 261-272.

Artykuł recenzowany

INFORMACJE

Studia Podyplomowe

Wydział Elektryczny Politechniki Śląskiej w Gliwicach, Instytut Metrologii, Elektroniki i Automatyki ogłasza nabór na Dwusemestralne Zaoczne Studia Podyplomowe

Sieci Komputerowe i Systemy Telekomunikacyjne (SKST)

Cel Studiów

Celem studiów jest przekazanie wiedzy teoretycznej i umiejętności praktycznych w zakresie: budowy bezpiecznych i wydajnych sieci komputerowych, konfiguracji i eksploatacji sieci komputerowych ze szczególnym uwzględnieniem sieci korporacyjnych, diagnostyki i pomiarów w sieciach komputerowych.

Zajęcia prowadzone są na Wydziale Elektrycznym Politechniki Śląskiej w Gliwicach, w systemie zaocznym w każdą sobotę lub co drugi weekend (opcja do wyboru), przez dwa semestry. Planowany termin rozpoczęcia: 1 października 2007 roku (pod warunkiem skompletowania odpowiedniej liczby kandydatów).

Warunki przyjęcia na studia:

1. Na studia mogą być przyjęte osoby posiadające dyplom magistra lub inżyniera, posiadające podstawową wiedzę z zakresu telekomunikacji i sieci komputerowych.
2. Kandydaci odbywają rozmowę kwalifikacyjną we wrześniu 2007 r.
3. Warunkiem uruchomienia studiów jest przyjęcie odpowiedniej liczby Kandydatów na podstawie złożonych dokumentów.

Organizator studiów:

Instytut Metrologii, Elektroniki i Automatyki Politechniki Śląskiej, 44-100 Gliwice, ul. Akademicka 10, tel. 032 237 12 41, fax: 032 237 20 34, e-mail: re2@polsl.pl lub agnieszka.skorkowska@polsl.pl, http://www.wega.elekt.polsl.gliwice.pl

Kierownik studiów:

Dr hab. inż. Lesław TOPÓR-KAMIŃSKI, prof. Pol. Śl.

