

WSPARCIE SYMULACJI OBIEKTÓW W CZASIE RZECZYWISTYM POPRAZ ZASTOSOWANIE TECHNIKI FPGA

REAL-TIME OBJECT MODELING ACCELERATION USING FPGA BASED SYSTEMS

Szymon Dolecki

Akademia Morska w Gdyni, Morska 81–87, 81-581 Gdynia, Wydział Elektryczny,
Studium Doktoranckie, e-mail: szymon.dolecki.sd@gmail.com

Streszczenie: Prowadząc badania naukowe lub prototypując nowe technologie, niezwykle ważne jest przeprowadzenie możliwie wielu testów przy jak najniższym koszcie i w najkrótszym możliwym czasie. Często jest to niewykonalne bez zastosowania kosztownych urządzeń, np. w czasie projektowania algorytmów regulacji. Przez ostatnie lata dużą popularność zdobywa modelowanie obiektów zainteresowania w czasie rzeczywistym. Wymaga to jednak większej ilości zasobów komputerowych. W artykule wskazano na próby rozwiązania tego problemu poprzez zastosowanie techniki FPGA, a przy użyciu fragmentu równań elektrycznych silnika synchronicznego o magnesach trwałych przedstawiono przykład poprawy szybkości obliczeń, jaką można uzyskać przy jej zastosowaniu.

Słowa kluczowe: symulacja, fpga, silnik elektryczny, implementacja.

Abstract: Scientific work and industrial development of new technologies require high number of tests while minimizing costs and time. It is often impossible to achieve this without costly devices i.e. during control loop hardware design. Through last several years real-time object modeling has increased in popularity. However it is very disadvantageous in terms of processing power requirements. This article points to examples of FPGAs solutions to specified problem. By using fragment of permanent magnet synchronous motor electrical equations there is shown an example of performance increase achievable with this technique.

Keywords: simulation, fpga, electric motor, implementation.

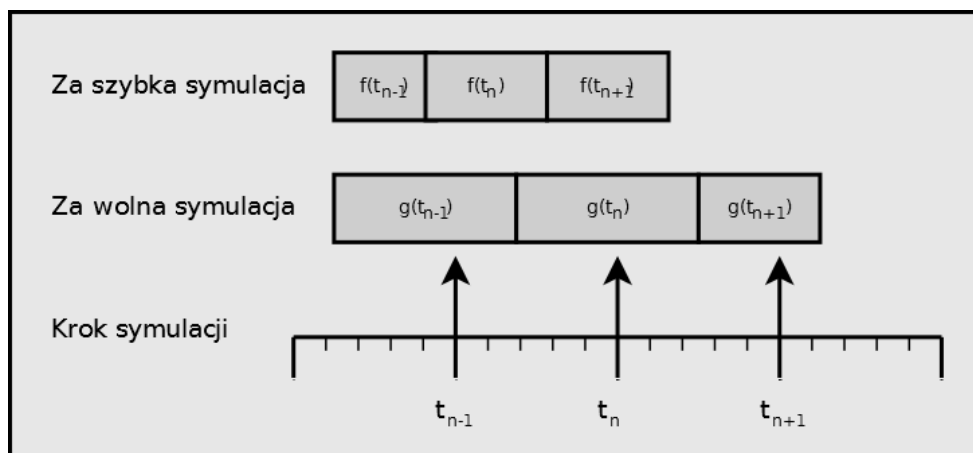
1. WSTĘP

Bardzo ważnym elementem pracy badawczo-rozwojowej jest testowanie i analizowanie nowych algorytmów. Często, w trakcie tego procesu, okazuje się, że układy wchodzące w skład testowanego systemu mogą zachować się niewłaściwie lub nieprzewidywalnie. Sytuacje takie mogą doprowadzić do uszkodzeń lub zniszczeń. Istnieją również przypadki, gdy nie ma bezpośredniego dostępu do

obiektów rzeczywistych. W takich sytuacjach należy zastosować modele symulacyjne. Te, jeśli mają być przetwarzane na komputerach klasy PC, wymagają znacznych ilości zasobów. Okazuje się, że pomimo rozwoju wiele nowoczesnych urządzeń nie pozwala na otrzymywanie wyników w czasie rzeczywistym [Menghal i Laxmi 2012]. Z pomocą przychodzą modele implementowane w technice FPGA.

2. PROBLEM SYMULACJI

Cyfrowa symulacja odbywa się w czasie dyskretnym, o stałej lub zmiennej wartości kroku [Menghal i Laxmi 2012]. Szybkość pracy symulatora, niezależnie od zastosowanego kwantu czasu, zależy przede wszystkim od dostępnych zasobów komputerowych. Skutkuje to faktem, iż symulacja taka może być znacznie szybsza niż rzeczywiste działanie obiektu symulowanego. Tu przykładem może być proces nagrzewania obiektu. Istotniejsza z punktu widzenia tego artykułu jest jednak sytuacja, w której przebieg symulacji jest, nawet nieznacznie, wolniejszy od procesu symulowanego. W obu tych przypadkach nie można takiej symulacji zastosować do testowania rozwiązań, gdyż warunki testowe będą się w sposób znaczny różnić od docelowych. Alternatywę stanowi symulacja w czasie rzeczywistym, której rozdzielczość czasowa musi być zgodna z rozdzielczością systemu badanego. Przy układach o dużej częstotliwości pracy pojawia się problem, gdy zasoby symulatora są niewystarczające, aby zapewnić dane w określonym kwancie czasu.



Rys. 1. Przebieg symulacji zbyt szybkiej i zbyt wolnej względem oczekiwanego kroku

Fig. 1. Faster and slower than expected real time simulation speed

3. AKCELERACJA SPRZĘTOWA

Początkowo symulacje układów w czasie rzeczywistym wykonywano na procesorach sygnałowych DSP, gdzie przyspieszenie zostało uzyskane poprzez zrównoleglenie operacji w wielu rdzeniach procesora [Throckmorton i Wozniak 1994]. Jednakże, wraz z postępem badań, modele stają się coraz bardziej skomplikowane, co pociąga za sobą wzrost wymagań mocy obliczeniowej. W związku z tym na początku XXI wieku podjęto próby zastąpienia dotychczasowych rozwiązań układami FPGA [Zhou i Mei 2005], umożliwiającymi zrównoleglenie wielokrotnie większej liczby operacji niż procesory sygnałowe [Dufour, Cense i Belanger 2014].

4. MODEL MATEMATYCZNY DLA AKCELERACJI

4.1. Silnik synchroniczny o magnesach trwałych

Silnik synchroniczny o magnesach trwałych (ang. *Permanent Magnet Synchronous Motor*) jest maszyną synchroniczną o wzbudzeniu pochodzącym od magnesów trwałych, umieszczonych na rotorze. W typowych konstrukcjach uzwojenie stojana jest trójfazowe, symetryczne i nie posiada wyprowadzonego przewodu zerowego [Zawirski, Deskur i Kaczmarek 2012]. Charakterystyczną cechą ww. maszyny jest sinusoidalny strumień magnetyczny. Jej zlinearyzowany opis matematyczny stanowi doskonały przykład układu dynamicznego, opisywanego równaniami macierzowymi [Krishnan 2010], na którego podstawie pokazano metodę przekształcania równań dla implementacji na układach FPGA. Nie przeprowadzono tu dokładnej analizy i implementacji pełnego modelu silnika synchronicznego, ponieważ dostępna jest w tym zakresie literatura [Dufour, Cacilda i Belanger 2010].

4.2. Model silnika synchronicznego w osiach ABC

Praca w technice FPGA umożliwia efektywne wykorzystanie możliwości przetwarzania równoległego, zwłaszcza w kontekście obliczeń macierzowych. W celu zaprezentowania tych możliwości zastosowano model silnika synchronicznego w osiach stacjonarnych ABC w odniesieniu do modelu w osiach wirujących $dq0$. Takie podejście pozwala na ustalenie parametrów modelu w sposób doświadczalny. Poniżej przedstawiono kilka wykorzystanych dalej wzorów:

$$\frac{dI_{abc}(t)}{dt} = L^{-1} \left(\begin{array}{c} \underline{U}_{abc}(t) - R_{abc} \underline{I}_{abc}(t) - p\omega \psi_f \begin{bmatrix} \sin(\gamma) \\ \sin\left(\gamma - \frac{2\pi}{3}\right) \\ \sin\left(\gamma + \frac{2\pi}{3}\right) \end{bmatrix} \end{array} \right) \quad (1)$$

$$L^{-1} = \frac{L^D}{\det(L)} \quad (2)$$

$$L^D = \begin{bmatrix} L_{bc}^2 - L_b L_c & L_c L_{ab} - L_{ac} L_{bc} & L_b L_{ac} - L_{ab} L_{bc} \\ L_c L_{ab} - L_{ac} L_{bc} & L_{ac}^2 - L_a L_c & L_a L_{bc} - L_{ab} L_{ac} \\ L_b L_{ac} - L_{ab} L_{bc} & L_a L_{bc} - L_{ab} L_{ac} & L_{ab}^2 - L_a L_b \end{bmatrix} \quad (3)$$

$$\det(L) = L_c L_{ab}^2 - 2L_{ab} L_{ac} L_{bc} - L_b L_{ac}^2 + L_a L_{bc}^2 + L_a L_b L_c \quad (4)$$

$$T_e(t) = p\psi_f \begin{bmatrix} -\sin(\gamma) & \sin\left(\gamma + \frac{\pi}{3}\right) & \sin\left(\gamma - \frac{\pi}{3}\right) \end{bmatrix} \underline{I}_{abc}(t) \quad (5)$$

$$\frac{d\omega}{dt} = \frac{T_e(t) - T_{LOAD}}{J} \quad (6)$$

$$\gamma(t) = p \int_0^t \omega(\tau) d\tau \bmod 2\pi \quad (7)$$

gdzie:

$\underline{I}_{abc}(t)$ – wektor prądów fazowych w osiach ABC,

$\underline{U}_{abc}(t)$ – wektor napięć w osiach ABC,

R_{abc} – macierz rezystancji stojana,

L – macierz indukcyjności,

γ – kąt obrotu rotora,

ω – prędkość kątowna rotora,

$T_e(t)$ – moment elektryczny generowany przez silnik,

T_{LOAD} – moment obciążenia,

J – bezwładność wału,

ψ_f^p – strumień magnetyczny magnesów maszyny o p-parach biegunów.

4.3. Dyskretyzacja modelu

Obliczenia w technice FPGA są obliczeniami cyfrowymi, stąd należy dokonać dyskretyzacji modelu. Jako że opis dostępny jest w przestrzeni czasu, można zastosować zależność różniczkową skończoną:

$$\left. \frac{dx(t)}{dt} = \frac{x(t + \Delta h) - x(t)}{\Delta h} \right|_{t=nT_s} \quad (8)$$

gdzie:

- n – numer próbki,
- T_s – okres próbkowania.

4.4. Zrównoleglanie obliczeń macierzowych

Etap ten jest niezwykle istotny w procesie tworzenia modelu dla techniki FPGA. Im więcej obliczeń możliwych jest do wykonania jednoczesnego, tym krótszy jest czas oczekiwania na ostateczny wynik. Dla przykładowego modelu silnika PMSM działania takie pojawiają się w momencie wyznaczania prądów fazowych na podstawie indukcyjności oraz napięć. Problem ten można podzielić na dwa osobne tak, aby pewne obliczenia wykorzystać wielokrotnie:

$$f_{I1}(n+1) = T_s \left(U_a(n+1) - R_a I_a(n) - \psi_f^p \sin(\gamma(n)) \omega(n) \right) \quad (9)$$

$$f_{I2}(n+1) = T_s \left(U_b(n+1) - R_b I_b(n) - \psi_f^p \sin\left(\gamma(n) - \frac{2\pi}{3}\right) \omega(n) \right) \quad (10)$$

$$f_{I3}(n+1) = T_s \left(U_c(n+1) - R_c I_c(n) - \psi_f^p \sin\left(\gamma(n) + \frac{2\pi}{3}\right) \omega(n) \right) \quad (11)$$

W równaniach (9, 10, 11) wprowadzone zostały trzy dodatkowe funkcje, stojące przy odwrotnościach indukcyjności w równaniach prądowych silnika PMSM:

$$I_a(n+1) = I_a(n) + L_{11}^{-1}(\gamma(n)) f_{I1}(n+1) + L_{12}^{-1}(\gamma(n)) f_{I2}(n+1) + L_{13}^{-1}(\gamma(n)) f_{I3}(n+1) \quad (12)$$

$$I_b(n+1) = I_b(n) + L_{21}^{-1}(\gamma(n)) f_{I1}(n+1) + L_{22}^{-1}(\gamma(n)) f_{I2}(n+1) + L_{23}^{-1}(\gamma(n)) f_{I3}(n+1) \quad (13)$$

$$I_c(n+1) = I_c(n) + L_{31}^{-1}(\gamma(n)) f_{I1}(n+1) + L_{32}^{-1}(\gamma(n)) f_{I2}(n+1) + L_{33}^{-1}(\gamma(n)) f_{I3}(n+1) \quad (14)$$

4.5. Test uzyskiwanego przyspieszenia

Na procesorze sygnałowym DSP TMS320C6421 firmy Texas Instruments oraz układzie FPGA Xilinx Spartan-6 SLX75-2 przeprowadzone zostały badania wydajności. Test polegał na przeprowadzeniu 1000 cykli obliczeń przedstawionych wyżej równań (9–14) i określeniu czasu wymaganego do pełnego wykonania.

Tabela 1. Zestawienie wyników testu porównawczego

Table 1. Summary of benchmark test results

	DSP	FPGA
Liczba taktów zegara potrzebnych na wykonanie jednego cyklu operacji	501	1
Częstotliwość taktowania [MHz]	589 824 000	20 097 674
Sumaryczna liczba cykli zegara	501 000	1000
Czas [ms]	0,84940	0,049757
Wydajność względna	1	17,07

To proste badanie wyraźnie pokazuje wzrost wydajności możliwy do osiągnięcia poprzez zastosowanie techniki FPGA. Należy zaznaczyć, iż tak znaczne przyspieszenie obliczeń odbywa się kosztem dużej zajętości logiki wewnątrz układu FPGA (98%), a implementacja wszystkich wymaganych równań zmniejszy ostateczny zysk.

5. PODSUMOWANIE

Technologia FPGA znajduje szersze zastosowanie stosunkowo od niedawna. Dzięki takiemu podejściu możliwe jest znaczne przyspieszenie symulacji, a po zsynchronizowaniu z otoczeniem model może stać się modelem czasu rzeczywistego. Choć istnieją już bardzo dobre rozwiązania, problem stanowi określenie konkretnych sposobów tworzenia modeli i rozwiązywania problemów dotyczących ich implementacji. Bazując na przedstawionym w pracy przykładzie, możliwe jest sprawne projektowanie implementacji modeli dla układów FPGA.

LITERATURA

- Dufour, C., Cacilda, A., Belanger, J., 2010, *Real-Time Simulation Technologies in Education: a Link to Modern Engineering Methods and Practices*, Proceedings of the 11th International Conference on Engineering and Technology Education INTERTECH 2010, Ilheus, Bahia, Brazil.
- Dufour, C., Cense, S., Belanger, J., 2014, *An Induction Machine and Power Electronic Test System on FPGA*, OPAL-RT TECHNOLOGIES, Montreal, Quebec, Canada.
- Krishnan, R., 2010, *Permanent Magnet Synchronous and Brushless DC Motor Drives*, CRC Press.
- Menghal, P.M., Laxmi, A.J., 2012, *Real Time Simulation: Recent Progress & Challenges*, Proceedings of International Conference on Power, Signals, Controls and Computation (EPCSCICON), Thrissur, India, s. 1–6.
- Throckmorton, P.J., Wozniak, L., 1994, *A Generic DSP-Based Real-Time Simulator with Application to Hydrogenerator Speed Controller Development*, IEEE Transactions on Energy Conversion, vol. 9, no. 2, s. 238–242.
- Zawirski, K., Deskur, J., Kaczmarek, T., 2012, *Automatyka napędu elektrycznego*, Wydawnictwo Politechniki Poznańskiej, Poznań.
- Zhou, Y.J., Mei, T.X., 2005, *FPGA Based Real Time Simulation for Electrical Machines*, IFAC World Congress.