

Zbigniew JACHNA, Ryszard SZPLET, Paweł KWIATKOWSKI, Krzysztof RÓŻYC

WOJSKOWA AKADEMIA TECHNICZNA,  
ul. Gen. Sylwestra Kaliskiego, 00-908 Warszawa

## Procesor kodu do realizacji procedur kalibracyjnych w interpolacyjnym liczniku czasu

Dr inż. Zbigniew JACHNA

Ukończył studia na Wydziale Elektroniki Wojskowej Akademii Technicznej. W roku 2003 uzyskał stopień doktora nauk technicznych. Jest adiunktem w Instytucie Telekomunikacji Wydziału Elektroniki WAT. Jego zainteresowania naukowe to projektowanie układów cyfrowych oraz tworzenie oprogramowania w zakresie precyzyjnej metrologii czasu.



e-mail: zjachna@wat.edu.pl

Mgr inż. Paweł KWIATKOWSKI

Jest absolwentem Wydziału Elektroniki Wojskowej Akademii Technicznej. Kontynuuje naukę na studiach doktoranckich. Jego zainteresowania naukowe dotyczą precyzyjnej metrologii czasu, w szczególności projektowania cyfrowych układów generacji i pomiaru odcinków czasu w programowalnych układach FPGA.



e-mail: pkwiatkowski@wat.edu.pl

Dr hab. inż. Ryszard SZPLET

Jest pracownikiem naukowo-dydaktycznym Wydziału Elektroniki (WEL) Wojskowej Akademii Technicznej (WAT), wykładowcą przedmiotów dotyczących teorii układów cyfrowych oraz projektowania systemów cyfrowych z użyciem układów programowalnych i specjalizowanych. Jego aktywność naukowo-badawcza koncentruje się na opracowywaniu metod i technik precyzyjnego pomiaru i generacji odcinka czasu. Kieruje zespołem badawczym metrologii czasu w Zakładzie Techniki Cyfrowej WEL WAT.



e-mail: rszplet@wat.edu.pl

Mgr inż. Krzysztof RÓŻYC

Ukończył studia na Wydziale Elektroniki Wojskowej Akademii Technicznej. Jest pracownikiem Instytutu Telekomunikacji WAT. Specjalizuje się w projektowaniu i testowaniu urządzeń do pomiaru, generacji oraz dystrybucji odcinków czasu.



e-mail: krozyc@wat.edu.pl

### Streszczenie

W artykule opisano projekt procesora kodu (PK) stanowiącego fragment dwukanałowego precyzyjnego licznika czasu z niezależnymi interpolatorami dwustopniowymi. Projekt został zrealizowany w układzie programowalnym XC6SLX75 (*Xilinx*). Zadaniem układów PK jest wykonywanie kalibracji linii kodujących, w wyniku której następuje aktualizowanie charakterystyk przetwarzania i w efekcie zwiększenie precyzji pomiarowej licznika. Dzięki sprzętowej implementacji algorytmów kalibracyjnych uzyskuje się skrócenie czasu wykonywania kalibracji, zmniejszenie liczby danych przesyłanych do komputera oraz zmniejszenie złożoności oprogramowania sterującego.

**Słowa kluczowe:** układy programowalne, przetworniki czasowo-cyfrowe, interpolacyjne liczniki czasu.

### A code processor for realization of calibration procedures in an interpolating time counter

#### Abstract

In the paper there is presented a design of a code processor (PK) as a part of a 2-channel precise time counter with independent 2-stage interpolators. The project was implemented in Spartan-6 (*Xilinx*) FPGA device. The main task of the PK is calibration of coding lines, resulting in updating transfer characteristics and, as an effect, higher measurement precision of the counter. Thanks to the hardware implementation of calibration algorithms there are achieved: the shorter execution time of calibration procedures, the lower amount of data transferred into the computer and less complex control software. The first simple realization of the PK has been implemented using Spartan-3 device (*Xilinx*) [8]. This paper presents a new, improved realization of the PK whose characteristic is more suited for the newest counters and those to be invented in the future. The use of VHDL language for description of the PK makes it more susceptible to be adapted. This paper consists of description of the counter with advanced architecture of interpolators [7], where 10 independent time coding lines where implemented in each measurement channel. The operating principle of the PK is described based on the following scheme: precise description of code density test realization, the way of forming the transfer characteristic and the results calculations.

**Keywords:** programmable device, time-to-digital converters, interpolating time counters.

## 1. Wstęp

Wysoka precyzja pomiarów odcinków czasu, wraz z szerokim zakresem pomiarowym, osiągnięta jest z użyciem liczników interpolacyjnych realizowanych w układach cyfrowych ASIC i FPGA [1-3]. Standardowo, w czasie działania tych urządzeń wyróżnia się trzy etapy: wygrzewania, kalibracji oraz pomiarów. Etap wygrzewania rozpoczyna się po włączeniu zasilania i kończy w chwili osiągnięcia zakładanych parametrów metrologicznych. Występowanie tego etapu jest typowe dla większości precyzyjnych przyrządów pomiarowych i nie stanowi zwykle istotnego ograniczenia funkcjonalności. Następujący po nim etap kalibracji służy do wyznaczenia dokładnej charakterystyki przetwarzania, bez której nie byłoby możliwe uzyskanie odpowiednio wysokiej precyzji pomiarowej. W czasie pracy urządzenia charakterystyka przetwarzania może ulegać zmianie na skutek zmian warunków otoczenia (temperatury, napięcia itp.). Dlatego procedurę kalibracji należy powtarzać, najlepiej z odstępem wyznaczanym wg określonego algorytmu [4]. Można też stosować stabilizację parametrów urządzenia [5] lub korektę charakterystyk przetwarzania w czasie wykonywania pomiarów [6]. Stosowanie każdego z wymienionych rozwiązań jest zwykle związane z właściwościami licznika. W dalszej części artykułu przedstawione zostanie rozwiązanie problemu kalibracji, które zastosowano w dwukanałowym liczniku czasu z niezależnymi interpolatorami dwustopniowymi [7].

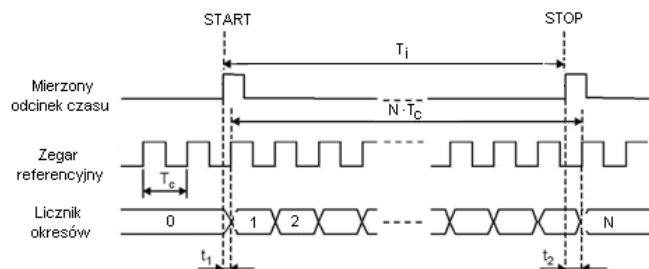
## 2. Interpolacyjny licznik czasu

Zasadę działania interpolacyjnego licznika czasu ilustrują przebiegi czasowe pokazane na rys. 1. Mierzony odcinek czasu  $T_i$ , zawarty między impulsami START i STOP, wyznaczany jest z użyciem trzech krótszych odcinków wg wzoru:

$$T_i = NT_c + t_1 - t_2 \quad (1)$$

Pierwszy odcinek ( $NT_c$ ) dotyczy czasu trwania pełnych okresów zegara referencyjnego zawartego pomiędzy aktywnymi zboczami zegara, obserwowanymi bezpośrednio po wystąpieniu aktywnych

zboczy impulsów START i STOP. Natomiast pozostałe dwa odcinki ( $t_1$  i  $t_2$ ) dotyczą ułamkowych części okresu zegara zawartych pomiędzy impulsami START/STOP i odpowiednimi końcami odcinka  $NT_c$ .

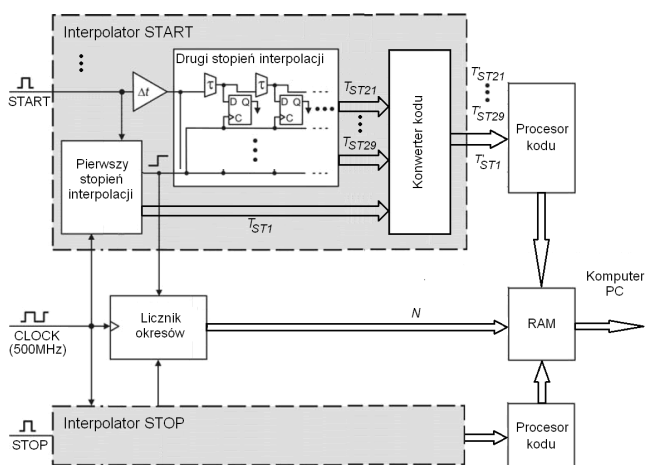


Rys. 1. Idea pomiaru odcinka czasu metodą interpolacyjną  
Fig. 1. The principle of time-interval measurement with use of interpolation method

Wyznaczanie odcinka  $NT_c$  odbywa się na podstawie stanu licznika okresów oraz znanej wartości okresu zegara referencyjnego. Stosując odpowiednio długi licznik można uzyskać praktycznie dowolną wartość zakresu pomiarowego. Jednakże trzeba podkreślić, że wraz ze wzrostem długości mierzonego odcinka czasu rośnie wpływ błędu oszacowania okresu zegara. Dlatego istotne jest stosowanie wysokostabilnych źródeł referencyjnych.

Pomiar odcinków  $t_1$  i  $t_2$  odbywa się z użyciem interpolatorów. Powinny one charakteryzować się odpowiednio wysoką rozdzielczością oraz precyzją. W projekcie licznika, opisanym w [7], wymienione parametry interpolatorów zostały uzyskane przez zastosowanie dwustopniowej interpolacji, gdzie w pierwszym stopniu uzyskuje się dowiązanie aktywnego zbocza impulsu START/STOP do jednej z 4 faz sygnału zegara, natomiast w drugim stopniu następuje zatrzęsienie tegoż impulsu z użyciem dziesięciu równoległych linii kodujących.

Na rys. 2 jest pokazany schemat blokowy 2-kanalowego licznika interpolacyjnego działającego według powyższej zasady.



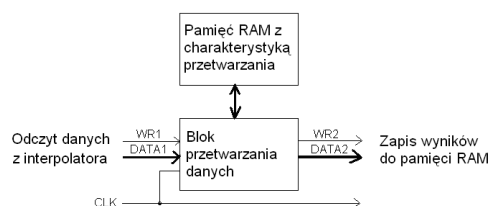
Rys. 2. Schemat blokowy licznika czasu  
Fig. 2. Block diagram of the time counter

Impulsy START i STOP są doprowadzane do osobnych, identycznie zbudowanych, interpolatorów. W interpolatorach wypracowywane są sygnały sterujące licznikiem okresów oraz wielobitowe sygnały stanowiące wynik interpolacji pierwszego ( $T'_{ST1}$ ) oraz drugiego stopnia ( $T'_{ST21}, \dots, T'_{ST29}$ ), czyli razem 644 bity ( $4 + 10 \cdot 64$ ). Następnie, bez straty informacji, wynik jest przekształcany w układzie konwertera kodu do prostszej postaci  $T'_{ST1}, T'_{ST21}, \dots, T'_{ST29}$  ( $2 + 10 \cdot 6 = 62$  bity) przesyłanej dalej do układu procesora kodu.

### 3. Procesor kodu (PK)

Zadaniem układów PK jest identyfikacja charakterystyk przetwarzania interpolatorów na podstawie statystycznego testu gęstości kodu oraz odczyt wyników pomiaru z użyciem zapamiętanej charakterystyki. Układy PK powinny działać w sposób możliwie autonomiczny (bezobsługowy), uaktualniać charakterystyki przetwarzania interpolatorów (najlepiej w tle) oraz udostępniać skorygowane wyniki interpolacji w postaci, która nie wymaga dalszych przeliczeń. Drugim, równie ważnym problemem, jest uniwersalność architektury układu PK, umożliwiającą obsługę zaawansowanych interpolatorów, np. wyposażonych w linie wielokrotne.

Pierwsze, względnie proste, realizacje bloków PK, zostały wprowadzone w scalonym liczniku czasu wykonanym w układzie Spartan-3 (Xilinx) [8]. Ograniczona ilość zasobów logicznych układu FPGA zdecydowała o zaniechaniu dalszego rozwijania tamtej konstrukcji. Jednak zdobyte doświadczenia oraz źródłowy opis projektu PK (w języku VHDL) zostały przeniesione i rozwinięte w układach liczników nowej generacji.



Rys. 3. Uproszczony schemat układu PK  
Fig. 3. Simplified block diagram of the PK

Uproszczony schemat blokowy obrazujący budowę układu PK jest pokazany na rys. 3. Dane wejściowe z interpolatora (DATA1) są wpisywane do bloku przetwarzania danych synchronicznie z sygnałem zegara CLK przy aktywnym poziomie sygnału WR1. Blok przetwarzania danych, jako element przystosowany do przetwarzania potokowego, wystawia na swoim wyjściu dane, stanowiące wartość odcinka czasu odczytaną z charakterystyki interpolatora, przy aktywnym poziomie sygnału WR2. Ważnym aspektem działania układu PK jest dysponowanie aktualną charakterystyką interpolatora, zapisaną w podręcznej pamięci RAM.

#### Oszacowanie potrzebnych zasobów logicznych

W bieżącym rozwiązaniu dane na wejściu układu PK reprezentowane są 62-bitowym słowem, na które składa się 2-bitowa dana z pierwszego stopnia interpolatora oraz 10 danych 6-bitowych z drugiego stopnia interpolatora. Do zapamiętania charakterystyki przetwarzania interpolatora przy 12-bitowej reprezentacji danych potrzeba pamięci o rozmiarze  $2^{62} \cdot 12$  bitów dla każdego kanału. Wielkość ta jest niemożliwa do realizacji, dlatego korzysta się z modelu linii ekwiwalentnej [7]. W tym przypadku rozmiar potrzebnej pamięci zmniejsza się do  $10 \cdot 2^8 \cdot 11 = 28160$  bitów. Został on wyznaczony w oparciu o następujące przesłanki:

- 10 bloków pamięci do odwzorowania charakterystyk niezależnych linii kodujących,
- 8-bitowe słowo adresowe stanowiące złożenie słów  $T'_{ST1}, T'_{ST2i}$ , gdzie ( $i$  – numer niezależnej linii kodującej),
- 11-bitowe dane reprezentujące czas na charakterystyce niezależnej linii kodującej,
- 12-bitowe dane reprezentujące czas na wirtualnej charakterystyce ekwiwalentnej otrzymywane są w wyniku przetwarzania danych odczytanych z dziesięciu niezależnych charakterystyk.

#### Oszacowanie błędu wprowadzanego przez PK

Na wyjściu układu PK dane występują w postaci słów 12-bitowych. Zatem zastosowanie układu PK daje oszczędność polegającą nie tylko na sprzętowym wyznaczeniu wyniku, ale również

na zmniejszeniu liczby danych przesyłanych do dalszego przetwarzania w komputerze. 12-bitowa reprezentacja czasu w układzie PK wprowadza błąd  $\varepsilon$ , którego maksymalną wartość, przy okresie zegara  $T_o = 2$  ns, można określić jako połowę rozdzielczości  $r$  tego układu wg wzoru:

$$\varepsilon_{\max} = \frac{r}{2} = \frac{2^{-12} T_o}{2} = 2^{-13} \cdot 2 \text{ ns} < 0.25 \text{ ps} \quad (2)$$

Wartość błędu można więc uznać za mało istotną, gdyż jest ona wielokrotnie mniejsza od precyzji licznika (wg [7] gwarantowana precyzja licznika jest na poziomie 10 ps). Wartość mierzonego odcinka czasu, z uwzględnieniem działania dwóch układów PK, wyznacza się wg następującej zależności:

$$T_i = T_o (N + T_1/2^{12} - T_2/2^{12}), \quad (3)$$

gdzie:  $T_1$  - wynik na wyjściu układu PK toru START,  $T_2$  - wynik na wyjściu układu PK toru STOP,  $N$  - liczba okresów  $T_o$  sygnału zegarowego,  $1/2^{12}$  - rozdzielczość układu PK.

### Procedura kalibracji

Procedura kalibracji składa się z dwóch etapów: etapu tworzenia charakterystyki źródłowej oraz etapu tworzenia charakterystyki przetwarzania.

W pierwszym etapie realizowany jest „test gęstości kodu interpolatora” polegający na wykonaniu odpowiednio dużej liczby pomiarów odcinków czasu generowanych za pomocą losowego generatora impulsów o rozkładzie równomiernym. Wynik, umieszczony w pamięci RAM, tworzy charakterystykę źródłową obrazującą stopień wykorzystania kodów. W wyniku wielu eksperymentów ustalono optymalną liczbę próby kalibracyjnej ( $2^{21}$ ) oraz oszacowano rozmiar niezbędnej pamięci RAM ( $10 \cdot 2^8 \cdot 18$ ). Przy przyjętych ustaleniach średnia liczba używanych kodów interpolatora (w każdej linii niezależnej) wynosi około 100, a średnie wykorzystanie każdego z kodów, niewiele ponad 2000. 13-bitowa reprezentacja mogłaby okazać się wystarczająca, jednak zastosowanie większej, 18-bitowej reprezentacji umożliwia dokładniejszą analizę rozwiązań przejściowych, charakteryzujących się znacznym zwiększeniem stopnia wykorzystania niektórych kodów interpolatora.

Algorytm pierwszego etapu wygląda następująco:

1. Zerowanie pamięci RAM,  $n = 0$ ;
2. Pojedynczy pomiar kalibracyjny odcinka czasu z generatora kalibracyjnego, którego wynikiem jest dana DATA1. Dana ta stanowi adres pamięci odpowiadający poszczególnym kodom interpolatora;
3. Inkrementacja wartości 18-bitowych słów danych DATA2, stanowiących krotność użycia określonych kodów interpolatora;
4.  $n = n + 1$ ;
5. Jeżeli  $n < N$  ( $N$  - wielkość próbki kalibracyjnej), to skocz do 2.
6. Koniec.

Prawidłowe wykonanie procedury oznacza, że suma wszystkich danych zawartych w pamięci jest równa ustalonej liczbie pomiarów kalibracyjnych  $N$ . Dla potrzeb niniejszej aplikacji ustalono stałą wartość  $N = 2\,097\,151$  (1FFFFh).

W drugim etapie następuje przekształcenie charakterystyki źródłowej w charakterystykę przetwarzania wg algorytmu:

1. Zerowanie słowa adresowego pamięci RAM: DATA1 = 0, zerowanie 21-bitowego rejestru SUM: SUM(20:0) = 0;
2. Odczyt pamięci RAM (słowo DATA2);
3. Wyznaczenie wartości słowa DATA: DATA = DATA2(16:10) & SUM(20:10). Zmiana zawartości rejestru SUM: SUM = SUM + DATA2 (& - operator złożenia);
4. Zapis słowa DATA do pamięci w miejsce wcześniej odczytanej danej DATA2;

5. Inkrementacja adresu: DATA1 = DATA1 + 1;

6. Jeśli DATA1 <= 255 skocz do 2;

7. Koniec.

Po zakończeniu procedury, pamięć RAM zawiera charakterystykę przetwarzania przetwornika. Kolejne adresy pamięci wskazują słowa danych DATA2(10:0), o rosnących wartościach od 0 do 2047, tworzące charakterystykę przetwarzania. Należy dodać, że pozostała część słowa DATA2, tj. DATA2(17:11), zawiera informację o szerokości danego kodu. Tą informację można wykorzystać do testowania prawidłowości działania procedury, gdyż suma wszystkich danych DATA2(17:11) umieszczonych w 8-bitowej przestrzeni adresowej powinna dawać wartość 2047. Do testowania prawidłowości działania PK opracowano także generator testu, który po włączeniu w miejsce interpolatora generuje ustalony strumień danych. Zatem utworzoną z jego użyciem charakterystykę można porównać z oczekiwanym wzorcem.

### 4. Wnioski

Opisany w artykule układ procesora kodu poprawia parametry precyzyjnych liczników czasu działających wg metod interpolacyjnych, przede wszystkim, pod względem przyspieszenia kalibracji i transmisji danych do komputera oraz uproszczenia złożoności oprogramowania użytkowego. Umożliwia również opracowywanie nowych architektur precyzyjnych liczników czasu, np. przystosowanych do pracy wielokanałowej, z pojedynczą lub wielostopniową interpolacją, wyposażonych w rozbudowane układy kalibracji dynamicznej, z wielokrotnymi liniami kodującymi.

*Projekt został sfinansowany ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC-2011/01/B/ST7/03278.*

### 5. Literatura

- [1] Kalisz J.: Review of methods for precise time interval measurements with picoseconds resolution, Metrologia, vol. 41, 2004.
- [2] Szplet R.: Time-to-Digital Converters, Chapter 7 in: Carbone P., Kiaei S., Xu F. (eds), Design, Modeling and Testing of Data Converters, Springer, 2014, pp 211-246.
- [3] Napolitano, P., Moschitta, A., Carbone, P.: A survey on time interval measurement techniques and testing methods, Proc. IEEE International Instrumentation and Measurement Technology Conf., I2MTC 2010, pp 181-186.
- [4] Kalisz J., Pawłowski M., Pełka R.: A method for autocalibration of the interpolation time interval digitiser with picosecond resolution, Journal of Physics E: Scientific Instruments, vol. 18, 1985.
- [5] Rivoir, J.: Fully-digital time-to-digital converter for ATE with autonomous calibration, Proc. IEEE International Test Conference, Santa Clara, United States (2006).
- [6] Rogacki S., Zurbuchen T. H.: A time digitizer for space instrumentation using a field programmable gate array, Review of Scientific Instruments 84, 083107 (2013).
- [7] Szplet R., Jachna Z., Różyk K., Kwiatkowski P.: High-Precise Portable Time Interval / Frequency Counter, 44th Annual Precise Time and Time Interval Systems and Applications Meeting, Reston, 2012.
- [8] Szplet R., Kalisz J., Jachna Z.: A 45 ps time digitizer with two-phase clock and dual-edge two-stage interpolation in Field Programmable Gate Array device, Measurement Science and Technology, 20 (2009). 025108 (11pp).

otrzymano / received: 09.04.2014

przyjęto do druku / accepted: 02.06.2014

artykuł recenzowany / revised paper