

Mieczysław JESSA, Michał JAWORSKI, Krzysztof LANGE

POLITECHNIKA POZNAŃSKA, KATEDRA SYSTEMÓW TELEKOMUNIKACYJNYCH I OPTOELEKTRONIKI

Implementacja programowalnego układu czasowego w układzie Virtex-5**Dr inż. Mieczysław JESSA**

Adiunkt na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Autor lub współautor ponad 90 publikacji, 15 patentów oraz kilkunastu rozwiązań konstrukcyjnych wdrożonych w krajowej sieci telekomunikacyjnej. Kierujący ponad dwudziestoma pracami wykonanymi na rzecz operatorów telekomunikacyjnych. Najważniejsze prace dotyczą synchronizacji sieci, dystrybucji sygnału czasu oraz zastosowań zjawiska chaosu w telekomunikacji i kryptografii.

e-mail: mjessa@et.put.poznan.pl

**Dr inż. Krzysztof LANGE**

Adiunkt na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Autor lub współautor ponad 80 publikacji, 15 patentów oraz rozwiązań konstrukcyjnych wdrożonych w krajowej sieci telekomunikacyjnej. Najważniejsze prace dotyczą tematyki systemów telekomunikacyjnych, a szczególnie badań sygnałów czasu i częstotliwości wzorcowych oraz ich dystrybucji.

e-mail: lange@et.put.poznan.pl

**Mgr inż. Michał JAWORSKI**

Doktorant na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Ukończył studia na tym samym wydziale w roku 2008. Jego zainteresowania to projektowanie urządzeń z wykorzystaniem układów programowalnych oraz nowe technologie internetowe.

e-mail: mjawor@et.put.poznan.pl

**Streszczenie**

W pracy opisujemy implementację programowalnego układu czasowego (PTU) w układzie Virtex-5 (XL5VLX50T). Przedstawiamy zasadę działania PTU, podstawowe tryby pracy oraz przykładowe przebiegi otrzymane w tym układzie.

Słowa kluczowe: faza, częstotliwość, programowanie funkcji, Virtex-5.

Implementation of the programmable time unit in the Virtex-5 circuit**Abstract**

Programmable Time Unit (PTU) offers numerous functions of digital processing of phase, frequency and width of rectangular impulses. The mode of working of PTU and the parameters of the output signal are programmable. The conversion process starts synchronously with the input signal. In this paper we describe an implementation of the programmable time unit (PTU) in Virtex-5 (XL5VLX50T) circuit. The principle of PTU work, basic modes of work and exemplary waveforms obtained in this circuit were presented. The set of applications includes electronics, telecommunications and informatics. The set of functions available in PTU can be divided into two basic groups. The first group contains: programmable delay of an impulse slope, programmable phase shift, pulse position modulation, pulse width modulation, frequency modulation with fixed or modulated pulse width, the generation of programmable number of impulses with programmable delay of generation, and many others. The second group contains functions that require a collaboration of PTU with one or two additional circuits. They are: broad-band phase shift, fast frequency synchronization in the wide range of the input frequencies, adding or multiplication of two numbers, etc. The PTU implemented in Virtex-5 uses less than 1% of its resources. It can also be implemented in cheaper FPGA as a single, low-cost circuit for general purposes.

Keywords: phase, frequency, programmable functions, Virtex-5.

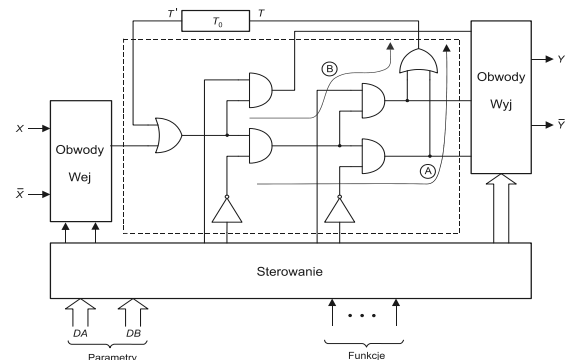
1. Wstęp

Rozwój układów FPGA umożliwia integrację wielu funkcji realizowanych dotychczas przez różne układy czy nawet urządzenia w jednej strukturze sprzętowej. Programowalny układ czasowy PTU (ang. *Programmable Time Unit*) oferuje użytkownikowi

szereg funkcji cyfrowego przetwarzania fazy, częstotliwości oraz wypełnienia przebiegu prostokątnego. Koncepcję układu oraz zbiór podstawowych funkcji opublikowano w pracy [1]. Realizacja techniczna opisanej koncepcji wymagała dotychczas użycia wielu układów scalonych i nie była konkurencyjna w stosunku do rozwiązań, w których pojedynczą funkcję PTU realizowano w postaci oddzielnego układu scalonego. Szeroki dostęp do tanich układów FPGA zmienił tę sytuację.

2. Programowalny układ czasowy

Zasada pracy programowalnego układu czasowego polega na cyklicznym obiegu krótkiego impulsu w dwóch pętlach zawierających ten sam element opóźniający T_0 , programowalną liczbę razy (pętle A i B na rysunku 1).

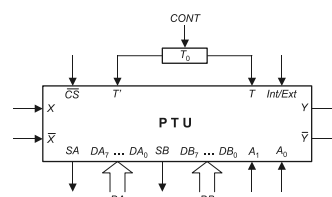


Rys. 1. Schemat blokowy ilustrujący zasadę pracy PTU

Fig. 1. The block diagram illustrating the principle of work of PTU

Poprawna praca układu wymaga, aby opóźnienie T_0 było dłuższe od czasu trwania krótkiego impulsu. Dodanie dodatkowych elementów umożliwia skonstruowanie układu, który potrafi zrealizować wiele funkcji cyfrowego przetwarzania fazy i częstotliwości przebiegu prostokątnego.

Z punktu widzenia użytkownika programowalny układ czasowy może być postrzegany jako pojedynczy układ scalony, którego funkcje oraz parametry są programowane przez podanie kombinacji zer i jedynek na odpowiednie wejścia układu (rys. 2).

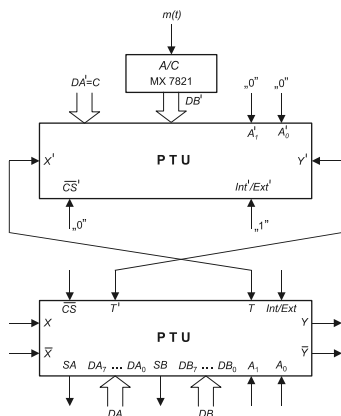


Rys. 2. Programowalny układ czasowy jako pojedynczy układ scalony

Fig. 2. Programmable time unit as a single integrated chip

Na rysunku 2 zastosowano następujące oznaczenia: \overline{CS} – uaktywnienie układu, X , \overline{X} – wejścia, odpowiednio proste i zanegowane, Y , \overline{Y} – wyjścia, odpowiednio proste i zanegowane, A_1 , A_0 – wejścia wyboru trybu pracy PTU, DA , DB – wejścia danych, SA – wyjście synchronizujące dla 8-bitowych danych DA , SB – wyjście synchronizujące dla 8-bitowych danych DB , Int/Ext – włączenie/wyłączenie wewnętrznej linii opóźniającej, T , T' – punkty, do których można dołączyć zewnętrzną linię opóźniającą z boczne impulsu prostokątnego o krótkim czasie trwania, $CONT$ - napięcie analogowe wpływające na wartość opóźnienia T_0 .

Elementem opóźniającym T_0 pokazanym na rysunkach 1 i 2 może być linia opóźniająca zbudowana z elementów układu FPGA, w którym zaimplementowano PTU, zewnętrzna linia opóźniająca lub inny PTU pracujący jako układ opóźniający z boczne sygnału wejściowego (rys. 3). Układ PTU' z rys. 3 wykorzystuje wewnętrzną linię opóźniającą o małym opóźnieniu. Zamiast tej linii można także zastosować linię zewnętrzną o stałym, lecz kalibrowanym opóźnieniu. Najprostsze rozwiązaniem polega na włączeniu parzystej liczby negatorów, których zasilanie zależy od wartości zewnętrznego sygnału kalibrującego.



Rys. 3. Wykorzystanie innego PTU jako zewnętrznej linii opóźniającej
Fig. 3. The use of another PTU as an external delay line

Na wejściach DA' i DB' układu PTU' użytkownik może nastawić opóźnienie $T_0 = T_p' + (DA + DB + 2) \cdot (T_{in}' + T_{INT}')$, gdzie T_p' jest pewnym statycznym opóźnieniem wprowadzanym przez obwody wejściowe i wyjściowe PTU', T_{in}' jest opóźnieniem wprowadzanym przez bramki uczestniczące w obiegu impulsu w pętli, a T_{INT}' jest opóźnieniem linii opóźniającej zbudowanej w FPGA. Na rysunku 3 pokazano także sposób, w jaki opóźnienie PTU' (opóźnienie T_0 dla PTU) może być modulowane sygnałem analogowym $m(t)$ w zakresie od $[0, 128(T_{in}' + T_{INT}')$. Jeżeli przyjmiemy, że na wejście DA nie wprowadzamy statycznego opóźnienia, to T_0 może być zmieniane w zakresie od $[T_p', T_p' + 128(T_{in}' + T_{INT}')$. Na podstawie podanych wzorów można zauważyć, że nastawy DA i DB mogą się zmieniać w zakresie od 0 do 127 włącznie. Wybranie $DB > 127$ powoduje, że impuls o standardowym czasie trwania, synchroniczny z przebiegiem wejściowym PTU, nie jest wprowadzany ani do pętli A ani do pętli B. Dla $DA > 127$ lub $DB > 127$ następuje blokowanie niektórych funkcji PTU, przy czym jest to proces synchroniczny z przebiegiem generowanym przez PTU, np. blokada następuje po wygenerowaniu pełnego okresu przebiegu wyjściowego. Asynchroniczne blokowanie pracy PTU można uzyskać zmieniając \overline{CS} z „0” na „1”.

Zbiór zastosowań PTU obejmuje telekomunikację, elektronikę oraz informatykę. Część funkcji PTU jest dostępna natychmiast po podaniu odpowiedniego sygnału na wejścia układu, inne wymaga-

ją użycia dodatkowych układów. Do grupy pierwszej zaliczamy programowalne opóźnienie zbocza impulsu, generowanie impulsu o programowalnym czasie trwania i programowalnym opóźnieniu generacji, generowanie przebiegu o programowalnej częstotliwości i wypełnieniu, generowanie programowalnej liczby impulsów z programowalnym opóźnieniem generacji, cyfrowa lub analogowa (zewnętrzna linia opóźniająca z T_0 kontrolowanym sygnałem analogowym) modulacja czasu trwania impulsu, położenia impulsu, częstotliwości lub wypełnienia przebiegu), generacja ciągu binarnego, szyfrowanie iloczynowe, itp. W grupie drugiej mamy takie funkcje jak szerokopasmowe przesunięcie fazy przebiegu prostokątnego, szerokopasmowy VCO, dodawanie liczb, mnożenie liczb, itp.

3. Implementacja PTU w układzie FPGA

Układ PTU można traktować jako samodzielne urządzenie zaimplementowane w tanim układzie FPGA. Alternatywnie może być wykorzystany jako urządzenie pomocnicze współpracujące z innymi urządzeniami wykonanymi w tej samej strukturze sprzętowej. W tym ostatnim przypadku istotna jest ocena jakości pracy PTU oraz stopnia wykorzystania zasobów potrzebnych dla implementacji PTU. Tę ocenę przeprowadziliśmy dla układu Virtex-5 firmy Xilinx (układ XL5VLX50T).

Podstawowa trudność implementacji programowalnego układu czasowego w FPGA wynika z asynchronicznej pracy wewnętrznych bloków PTU. Układy programowalne nie są optymalizowane dla takich zastosowań, co utrudnia implementację. Na przykład opis bloku generatora krótkich impulsów wymaga zastosowania odpowiednich instrukcji sterujących syntezatorem, które pozwolą zaimplementować go zgodnie z ideą. Bez takiego zabiegu, oprogramowanie w procesie optymalizacji usuwa z projektu istotne elementy. Kaskady negatorów wprowadzające opóźnienie, niezbędne dla idei PTU, są optymalizowane do jednej bramki dla nieparzystej liczby negatorów lub są całkowicie usuwane, gdy liczba negatorów użytych do opóźnienia sygnału jest parzysta. Ponieważ w PTU bardzo ważne są zależności czasowe pomiędzy sygnałami występującymi w różnych miejscach układu oraz wartości opóźnień na poszczególnych ścieżkach przepływu sygnału, istotne znaczenie ma także rozmieszczenie elementów wewnątrz struktury układu FPGA. Optymalny rozkład umożliwi także uzyskanie maksymalnej szybkości pracy. Ścieżka krytyczna dotyczy bloku pętli i jego połączenia z układami liczącymi impulsy, blok sterowania oraz obwody wyjściowe nie są w tym przypadku istotne.

Prototyp PTU zbudowany został w oparciu o układy scalone serii TTL. Implementacja w układzie FPGA nie odzwierciedla fizycznie wersji prototypowej, ponieważ elementami kombinacyjnymi w układach programowalnych nie są bramki, lecz inne struktury, tzw. LUTy (ang. *lookup table*). W przypadku układu Virtex-5 LUTy są 6-wejściowe, co nie ma jednak wpływu na warstwę funkcjonalną PTU. Przykładowy kod, opisujący blok pętli zaznaczony na rysunku 1 linią przerywaną, wygląda następująco:

```
module loops(
  input i_pulse,
  input i_delay,
  input i_N1,
  input i_N2,
  output wire o_delay,
  output wire o_pulsea,
  output wire o_pulseb,
  output wire o_pulsec);
  assign p1 = i_pulse | i_delay;
  assign o_pulsea = p1 & i_N2;
  assign p2 = p1 & ~i_N2;
  assign o_pulseb = p2 & i_N1;
  assign o_pulsec = p2 & ~i_N1;
  assign o_delay = o_pulseb | o_pulsec;
endmodule
```

W procesie projektowym przyjęto założenie o nie stosowaniu żadnych elementów specyficznych dla architektury Virtex-5. W rezultacie otrzymano kod w pełni przenośny na inne układy firmy Xilinx. Zastosowanie do opisu działania PTU języka Verilog pozwala na łatwe i szybkie wprowadzenie zmian, na przykład zwiększenie zakresu liczników. Układ 8-bitowy można szybko i łatwo przekształcić w PTU o praktycznie dowolnej długości słowa danych, na przykład w PTU 32-bitowy. Dzięki temu możemy dostosować PTU do potrzeb urządzenia lub systemu wykonanego w tym samym układzie FPGA. Poziom wykorzystania zasobów układu XL5VLX50T przez 8-bitowy PTU jest mniejszy od 1%.

Na rysunkach 4, 5, 6 i 7 pokazano pracę 8-bitowego PTU zaimplementowanego w układzie XL5VLX50T dla czterech podstawowych trybów pracy: programowalne opóźnienie zbocza impulsu ($A_1="0"$, $A_0="0"$), generowanie impulsu o programowalnym czasie trwania i programowalnym opóźnieniu generacji ($A_1="0"$, $A_0="1"$), generowanie przebiegu o programowalnej częstotliwości lub wypełnieniu ($A_1="1"$, $A_0="0"$), generowanie programowalnej liczby impulsów z programowalnym opóźnieniem generacji ($A_1="1"$, $A_0="1"$). Czas trwania impulsu obiegającego pętlę A i B ustalono na 2ns (wartość minimalna – 0,8ns). Zmierzony czas propagacji T_p przez PTU wynosił $7ns \pm 0,5ns$. Opóźnienie wewnętrzne T_0 dobrano w taki sposób, aby czas obiegu pojedynczej pętli $\tau = T_m + T_{INT}$ był równy $5ns \pm 50ps$ (wartość minimalna – 1ns). Kalibrację czasu obiegu można do pewnego stopnia osiągnąć poprzez odpowiednie prowadzenie ścieżek wewnątrz układu FPGA. Wszystkie wartości liczbowe są obciążone błędem nie większym od 1%.

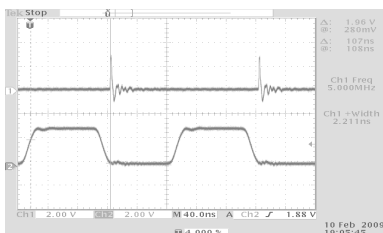
Wartość opóźnienia zbocza impulsu otrzymanego w trybie $A_1="0"$, $A_0="0"$ ze zbocza przebiegu wejściowego (rys. 4) obliczamy ze wzoru

$$\Delta\phi = T_p + (DA + DB + 2) \cdot \tau. \quad (1)$$

Opóźnienie $\Delta\phi$ generacji impulsu prostokątnego o programowalnym czasie trwania ΔT (rys. 5) opisują równania, odpowiednio

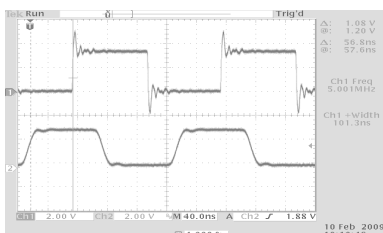
$$\Delta\phi = T_p + (DA + 1) \cdot \tau, \quad (2)$$

$$\Delta T = (DB + 1) \cdot \tau. \quad (3)$$



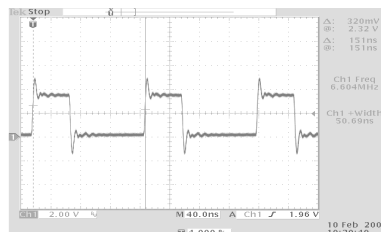
Rys. 4. Programowalne opóźnienie zbocza narastającego impulsu lub przebiegu prostokątnego; $DA=9$, $DB=19$

Fig. 4. Programmable delay of the rising slope of the input pulse or rectangular wave; $DA=9$, $DB=19$



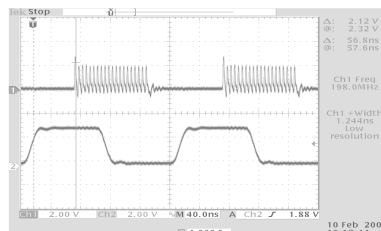
Rys. 5. Generacja impulsu prostokątnego o programowalnym czasie trwania i programowalnym opóźnieniu generacji; $DA=9$, $DB=19$

Fig. 5. The generation of a rectangular pulse with programmable time duration and programmable delay of generation; $DA=9$, $DB=19$



Rys. 6. Generacja przebiegu prostokątnego o programowalnej częstotliwości i wypełnieniu; $DA=9$, $DB=19$

Fig. 6. The generation of a rectangular wave with programmable frequency and pulse width; $DA=9$, $DB=19$



Rys. 7. Generacja programowalnej liczby impulsów o programowalnym opóźnieniu generacji; $DA=9$, $DB=19$

Fig. 7. The generation of a programmable number of pulses with programmable delay of generation; $DA=9$, $DB=19$

Okres T i wypełnienie W przebiegu prostokątnego generowanego przez PTU (rys. 6) obliczamy ze wzoru, odpowiednio

$$T = (DA + DB + 2) \cdot \tau, \quad (4)$$

$$W = (DA + 1) / (DA + DB + 2). \quad (5)$$

Zauważmy, że w PTU oba parametry możemy dobrać niezależnie. Maksymalna częstotliwość generowanego przebiegu wynosi 99,20MHz, co odpowiada czasowi jednego obiegu w pętli równemu 5,040ns. Tę wartość otrzymano dla $DA=DB=0$. Liczba impulsów generowanych w trybie $A_1="1"$, $A_0="1"$ (rys. 7) jest równa $DB+1$, a opóźnienie generacji można obliczyć ze wzoru

$$\Delta\phi = T_p + (DA + 1) \cdot \tau. \quad (6)$$

Zastosowanie zamiast T_0 dodatkowego PTU pracującego w trybie $A_1="0"$, $A_0="1"$ pozwala wpływać na czas trwania impulsów wyjściowych oraz na ich odległość.

4. Podsumowanie

W pracy przedstawiono zasadę pracy, implementację w układzie Virtex-5 oraz podstawowe tryby pracy programowalnego układu czasowego. Cechą charakterystyczną PTU, oprócz integracji wielu funkcji cyfrowego przetwarzania fazy i częstotliwości przebiegu prostokątnego jest to, że przetwarzanie fazy rozpoczyna się synchronicznie z sygnałem wejściowym i może zakończyć się synchronicznie z sygnałem wyjściowym. Lista funkcji, które są dostępne za pomocą PTU obejmuje co najmniej dwadzieścia pozycji [1]. Ograniczony rozmiar artykułu nie pozwala na przedstawienie ich wszystkich. Mamy nadzieję, że pozostałe funkcje będzie można zaprezentować w kolejnych artykułach. Interesujące jest także zaimplementowanie PTU w innych układach programowalnych typu CPLD i FPGA. Koszt pozyskania wielu funkcji PTU może wówczas spaść do kwoty poniżej 1 USD.

5. Literatura

- [1] Jessa M., Lange K.: Programmable Time Unit, Proc. of Europ. Conf. on Circuit Theory and Design, ECCTD'95, Istanbul, 1995, pp. 63-66.
- [2] <http://www.xilinx.com/support/documentation/virtex-5.htm>.