Paweł KUBCZAK, Mieczysław JESSA, Łukasz MATUSZEWSKI
POLITECHNIKA POZNAŃSKA, WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI,
ul. Polanka 3, 60-965 Poznań

Generator ciągów losowych wykorzystujący stany metastabilne zaimplementowany w układzie FPGA firmy Xilinx

Mgr inż. Paweł KUBCZAK

Studia ukończył na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej w roku 2013. Kontynuuje naukę na studiach doktoranckich na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Jego zainteresowania naukowe dotyczą cyfrowych generatorów liczb losowych, pomiaru odcinka czasu z dokładnością pikosekundową oraz programowalnych układów cyfrowych.

e-mail: kubczak.pawel@gmail.com

Dr hab. inż. Mieczysław JESSA

Adiunkt na Wydziału Elektroniki i Telekomunikacji Politechniki Poznańskiej, Katedra Systemów Telekomunikacyjnych i Optoelektroniki. Autor lub współautor ponad 120 publikacji, 15 patentów oraz kilkunastu rozwiązań konstrukcyjnych wdrożonych w krajowej sieci telekomunikacyjnej. Kierował ponad trzydziestoma projektami wykonanymi na rzecz podmiotów gospodarczych. Najważniejsze prace dotyczą synchronizacji sieci telekomunikacyjnej, zastosowań zjawiska chaosu oraz losowości i pseudolosowości.

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

Streszczenie

W wysokiej klasy systemach bezpieczeństwa informacji klucze kryptograficzne nie powinny być generowane na zewnątrz systemu, a klucze prywatne, w przeciwieństwie do publicznych, nigdy nie powinny opuścić systemu. Jeśli system bezpieczeństwa jest realizowany w jednym układzie scalonym, klucze powinny być generowane w tym samym układzie. Realizacja generatorów liczb losowych w cyfrowych układach reprogramowalnych jest więc istotnym zagadnieniem. W artykule przedstawiono nową metodę wytwarzania ciągów losowych, opartą o zjawisko metastabilności występujące w układach cyfrowych oraz uwagi na temat sensowności wykorzystania tego fizycznego efektu występującego we współczesnych, powszechnie dostępnych układach cyfrowych.

Słowa kluczowe: metastabilność, generatory ciągów losowych, układy programowalne, kryptografia.

A true random number generator exploiting metastability implemented in Xilinx FPGA

Abstract

The security of cryptographic systems relates mainly to the protection of confidential keys. In high-end information security systems, cryptographic keys should never be generated outside the system and private keys should never leave the system. For the same reason, if the security system is implemented in a single chip (cryptographic system on chip), the keys should be generated inside the same chip. Implementation of random number generators in logic devices, including configurable logic devices, is therefore an important issue. In this paper, we present a new method of generating random digits based on a physical phenomenon occurring in digital circuits. Thus, the proposed generator can be implemented in different Field Programmable Gate Arrays (FPGAs) with other elements of the cryptographic system. If the underlying physical process cannot be controlled, the generator output is unpredictable and/or uncontrollable. The statistical characteristics of TRNGs are closely related to the quality of the entropy source, but also to the randomness extraction method. The statistical quality of the generator was verified with the use of NIST statistical test suite. A discussion of the utility of metastable states for producing random numbers with metastable states in commercially available FPGAs is also presented.

Keywords: metastability, random number generators, field programmable gate arrays, cryptography.

Mgr inż. Łukasz MATUSZEWSKI

Asystent na Wydziale Elektroniki i Telekomunikacji Politechniki Poznańskiej. Ukończył studia na tym samym wydziale w roku 2010. Jego zainteresowania to projektowanie urządzeń z wykorzystaniem układów reprogramowalnych w szczególności generatorów liczb losowych i układów synchronizacji.

e-mail: lukasz.matuszewski@et.put.poznan.pl

1. Wstęp

Generatory liczb losowych stanowią jeden z podstawowych elementów używanych w protokołach kryptograficznych. Zastosowanie znajdują w wytwarzaniu kluczy kryptograficznych, wektorów inicjacji, wartości dopełnień itp. [1]. Generatory przeznaczone do zastosowań kryptograficznych muszą spełnić kilka wymogów bezpieczeństwa. Ciągi wyjściowe muszą mieć dobre właściwości statystyczne i być nieprzewidywalne. Są to urządzenia, których działanie jest oparte o zjawiska fizyczne takie jak na przykład szumy termiczny czy efekty kwantowe [2, 3]. Obecnie są implementowane jako układy analogowe, których scalenie z istniejącymi układami programowalnymi w jeden układ jest niemożliwe. Większość współczesnych systemów kryptograficznych to urządzenia cyfrowe. Dlatego oczekuje się, aby generatory ciągów losowych również były konstrukcjami cyfrowymi, dającymi się zintegrować z całym systemem.

W artykule zaproponowano nową metodę wytwarzania ciągów losowych w oparciu o stany metastabilne rozważane w literaturze, łatwą do zaimplementowania w dowolnym układzie FPGA. W odróżnieniu od dotychczasowych rozwiązań, generator posiada wbudowany układ do wykrywania stanów metastabilnych pojawiających się na pojedynczym przerzutniku typu D. Jeżeli taki stan wystąpił to na wyjściu otrzymujemy bit lub bity związane z tym stanem. W badaniach wykorzystano powszechnie dostępny układ Spartan 3 firmy Xilinx [4]. Inną nowością w zaproponowanej metodzie jest pośrednie wykorzystanie zjawiska metastabilności tzn. jego wystąpienie powoduje negację sygnału na wyjściu generatora, które następnie jest równomiernie próbkowane. W badaniach wykorzystano pakiet NIST 800-22 [5]. Ze względu na znacząco mniejsze przepływności otrzymanych strumieni od deklarowanych w literaturze, dokonano także krytycznej analizy rozwiązań istniejących.

2. Opis metastabilności

Jednymi z podstawowych elementów układów cyfrowych są różnego rodzaju przerzutniki. Istnieje wiele ich rodzajów - D, JK, T, zatrzaski (ang. *latch*) itd. Niezależnie od typu i formy, w jakiej przerzutnik występuje, warunkiem jego funkcjonowania zgodnie z tabelą prawdy jest zachowanie określonych relacji czasowych między zmianami sygnałów na wejściach informacyjnych, a zmianami sygnału zegarowego. Typowymi parametrami czasowymi przerzutnika synchronicznego są: czas ustalenia sygnału na wejściu informacyjnym (lub wejściach) przed nadejściem aktywnego zbocza na wejściu zegarowym (ang. *setup time* t_s) i czas przetrzymywania sygnału po aktywnym zboczu na tym wejściu (ang. *hold time* t_h). Wartości t_s i t_h wyznaczają okno czasowe wokół aktywnego zbocza sygnału zegarowego wewnątrz którego stan wejścia informacyjnego powinien być stabilny. Niestety, niemal każde urządzenie cyfrowe ma styczność z nadchodzącymi

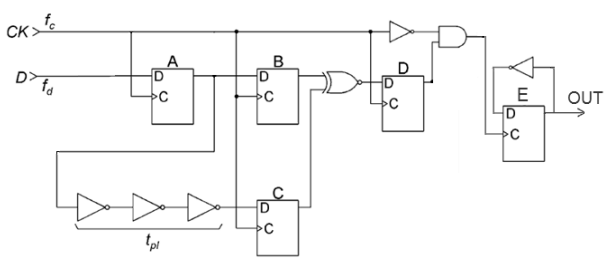
z otoczenia sygnałami, które są asynchroniczne w stosunku do sygnałów zegarowych. Są to sygnały pochodzące np. z zewnętrznych czujników, układów komunikacyjnych, elementów systemu wykorzystujących inne źródło sygnału zegarowego itp. W takich warunkach zapewnienie określonej relacji czasowej między sygnałami informacyjnymi a zegarem jest niemożliwe. Gdy naruszone zostaną parametry t_s, t_h wystąpić może zjawisko metastabilności. Przerzutnik – zamiast zdecydowanie osiągnąć jeden ze stanów: 0 albo 1 – może na pewien czas znaleźć się w stanie równowagi chwiejnej. Wskutek szumów, nierównomierności wzmocnień elementów czynnych itp. zostaje on po pewnym czasie wytrącony z tego stanu pośredniego i osiąga jeden ze stanów stabilnych, jednakże konsekwencje tego "wahania się" mogą być znaczące. Oczywiście, nie każde naruszenie wartości t_s, t_h powoduje wejście przerzutnika w stan metastabilny. Wartości t_s, t_h określają przedział, wewnątrz którego znajduje się – zwykle dość wąskie – okno metastabilności, tj. obszar, w którym zmiana stanu wejścia informacyjnego wywołuje stan metastabilny. Położenie tego okna zmienia się z napięciem, temperaturą i egzemplarzem przerzutnika.

Zachowanie się sygnału wyjściowego przerzutnika, który osiągnął stan metastabilny może być rozmaite i zależy od technologii w jakiej jest on wykonany. Metastabilność może objawić się generacją wąskiego impulsu, oscylacjami, wydłużeniem zbocza sygnału wyjściowego lub opóźnieniem zmiany stanu wyjścia [6, 7, 8].

Stan, w jakim znajdzie się przerzutnik po wyjściu ze stanu metastabilnego jest losowy – przerzutnik może zadziałać zgodnie z nowym stanem wejścia informacyjnego lub je zignorować. W większości urządzeń sygnał asynchroniczny jest stabilny przez wiele okresów sygnału zegarowego, zatem metastabilność spowoduje, że zmiana stanu sygnału asynchronicznego wpłynie na urządzenie co najwyżej o jeden takt zegara później. Sytuacja taka zazwyczaj jest akceptowalna. Gorzej przedstawia się sprawa dodatkowego czasu, który wymagany jest dla osiągnięcia stanu stabilnego przez przerzutnik. Może on rzutować na graniczną częstotliwość pracy układu. Czas przebywania przerzutnika w stanie metastabilnym jest zależny od wielu czynników, w tym od cech technologicznych przerzutnika, poziomu szumów w układzie itp. Jest on niedeterministyczny, nie można zatem podać jego granicznej wartości, a jedynie określić jego rozkład, tj. podać prawdopodobieństwo osiągnięcia stanu stabilnego po określonym czasie [9].

3. Implementacja i testy

Schemat proponowanego układu przedstawiono na rysunku 1. Przerzutnik, który jest wprowadzany w stan metastabilny jest oznaczony jako A. Jego wyjście jest prowadzone dwoma równoległymi ścieżkami; jedna droga to bezpośrednie połączenie z przerzutnikiem B, a druga przez linie opóźniającą złożoną z trzech inwerterów do przerzutnika C. Jeżeli wystąpił stan metastabilny, stany wyjść przerzutników B i C będą takie same, więc na bramce XNOR, będzie stan wysoki. Wystąpienie takiego stanu jest rejestrowane przez przerzutnik D, a następnie sygnał wędruje do przerzutnika E, którego wyjście jest negowane, kiedy pojawia się stan metastabilny.



Rys. 1. Schemat generatora
Fig. 1. Schematic diagram of the generator

Układ został zaimplementowany na płytce ewaluacyjnej z wbudowanym układem Spartan 3 firmy Xilinx. Przebieg na wyjściu generatora był badany po podaniu na wejścia D i CK sygnałów o częstotliwości odpowiednio 50 MHz i 1 MHz.

W celu oceny jakości generowanej sekwencji, wyjście układu zostało poddane próbkowaniu równomiernemu, a otrzymane bity przesłano do komputera z wykorzystaniem transmisji szeregowej i układu FTDI. Wygenerowane ciągi (sto ciągów o długości miliona bitów) zostały przetestowane za pomocą baterii testów statystycznych NIST 800-22 [5]. Wyniki testów dla prędkości próbkowania 25 b/s pokazano w tabeli 1.

Tab. 1. Wyniki testu NIST dla prędkości próbkowania 25 b/s dla 100 ciągów
Tab. 1. The results of NIST tests for sampling speed 25 b/s for 100 sequences

Rodzaj testu	p-wartość	proporcja
Test częstości	0,359555	0,9894
Blokowy test częstości	0,500934	0,9789
Test skumulowanych sum*	0,750985	0,9789
Test ciągów	0,728748	0,9894
Test na najdłuższy ciąg jedynek w bloku	0,103890	0,9894
Test stopnia macierzy binarnej	0,479268	1,0000
Test spektralny FFT	0,260784	1,0000
Test dopasowania nienakładających się wzorców*	0,001522	0,9578
Test dopasowania nakładających się wzorców	0,683283	0,9789
Test uniwersalny Maurera	0,246470	1,0000
Test przybliżonej entropii	0,097224	0,9684
Test błądzenia losowego*	0,014216	0,9696
Test wariancji błądzenia losowego*	0,022503	0,9696
Test serii*	0,291249	0,9894
Test złożoności liniowej	0,545381	0,9789
*Test składa się z kilku testów szczegółowych; w tabeli podano wynik najgorszy		

Dla prędkości wyjściowej 25 b/s układ przechodził wszystkie testy NIST. Niestety wraz ze wzrostem częstotliwości próbkowania układ przestaje przechodzić kolejne testy, co oznacza że entropia na wyjściu generatora nie jest duża. Cały obwód generatora zajmuje kilkadziesiąt komórek logicznych w układzie FPGA. Biorąc pod uwagę, że nowe układy Virtex 6 firmy Xilinx, mają bardzo dużo komórek logicznych, można w takim układzie umieścić np. dziesięć tysięcy takich generatorów pracujących równolegle i teoretycznie uzyskać prędkość wyjściową na poziomie 0,25 Mb/s. Niestety po przeprowadzonych badaniach okazało się, że dla nowych układów uzyskujemy jeszcze mniej bitów z pojedynczego przerzutnika.

Jak widać, możliwe jest stworzenie generatora ciągów prawdziwie losowych opartego o stany metastabilne, lecz jest to trudne w układach dostępnych komercyjnie, a uzyskane prędkości bitowe są bardzo małe. Obecnie producenci sprzętu elektronicznego dążą do minimalizacji prawdopodobieństwa wystąpienia stanu metastabilnego, w celu zwiększenia maksymalnej prędkości układu cyfrowego. Okna metastabilności są coraz węższe, np. dla układów Virtex 7 firmy Xilinx [4] czas narastania sygnału zegarowego wynosi kilkanaście pikosekund, a okno metastabilności jest krótsze od tego czasu. Uzyskanie stanu metastabilnego w konwencjonalnym podejściu, wymaga trafienia w okno metastabilności z dokładnością pikosekundową. Teoretycznie jest to możliwe przy wykorzystaniu układu DLL (ang. *delay-locked loop*) dającego możliwość przesunięcia sygnału o kilka stopni, co przekłada się na opóźnienie sygnału o kilka pikosekund. Jednak automatyczne rozprowadzenie sygnału na płytce układu FPGA wprowadza nanosekundowe opóźnienia.

4. Porównanie generatorów losowych wykorzystujących zjawisko metastabilności

W pracy [10] autorzy wprowadzają pojedynczy przerzutnik D znajdujący się w tak zwanej „otwartej pętli” w stan bliski metastabilności. Wtedy to szum termiczny na wejściu i wyjściu przerzutnika powoduje pojawianie się na wyjściu stanu zero, jeden lub

stanu metastabilnego. W układzie opóźnienie jest dobrane tak by sygnał danych był próbkowany w pobliżu połowy napięcia zasilającego. Wtedy szumy termiczne i środowiskowe powodują losową zmianę stanu na wyjściu lub pojawienie się stanu metastabilnego, który nie jest konwertowany na wartość logiczną. Stan ten zmienia się w stan stabilny po pewnym czasie τ, zależnym od szumu termicznego i jest dodatkowym elementem wprowadzającym losowość w układzie. Co ciekawe, autorzy przyznają, że prosta zasada działania układu jest trudna w realizacji z dwóch powodów; po pierwsze układy FPGA są projektowane tak by walczyć z metastabilnością, a po drugie przesłuchy, zmiany temperatury oraz napięcia wpływają na opóźnienia w układzie. Układ zawiera *n* elementów wprowadzających opóźnienia, po jednym dla każdego przerzutnika. Wyjścia pojedynczych układów są dodawane modulo by wykryć zmianę na którymkolwiek przerzutniku. Opóźnienie dla pojedynczego przerzutnika musi być wystarczająco małe (z dokładnością do kilku pikosekund) tak by zapewnić, że wyjście pojedynczego przerzutnika nie pozostanie w stanie '0' lub '1'. Liczba elementów generujących dodatkowo musi być dostatecznie duża, by podczas ataku na generator poprzez zmiany napięcia czy temperatury przynajmniej jeden przerzutnik pozostawał niestabilny. Sygnał zegarowy przed wprowadzeniem na linie danych jest precyzyjnie opóźniany za pomocą potencjometru nie znajdującego się w układzie tak by dobrać jak największy potencjał losowości. Autorzy przyznają, że spełnienie ograniczeń czasowych jest trudne nawet przy precyzyjnym strojeniu potencjometru. W drugiej wersji układu autorzy zastępują przerzutniki D zatrzaskami opartymi na LUT, które nie są zaprojektowane by redukować efekt metastabilności. W tej wersji rozłożenie elementów było ograniczane za pomocą „constraintów", by spełnić wymogi czasowe. W artykule została zbadana jakość generacji z dokładnością do pół obrotu potencjometru. Przed testami statystycznymi NIST została poprawiona częstość występowania zer i jedynek poprzez sumowanie modulo oraz grupę rejestrów LFSR, co oznacza, że po wstępnych testach ciąg musiał mieć bardzo słabe właściwości.

Generator oparty o otwartą pętlę znajdujemy również w artykule [11]. Tutaj autorzy skupiają się na implementacji układu wyłącznie w układzie Virtex 5. Tak jak w innych układach z otwartą pętla różnica opóźnienia pomiędzy sygnałem zegarowym i danych musi być ustawiona z dużą dokładnością, tak aby trafić w okno metastabilności. W prezentowanym układzie linie opóźniające są tworzone dla sygnału zegarowego i sygnału danych, co umożliwia uzyskanie małego opóźnienia różnicowego, zarówno dodatniego jak i ujemnego. W układzie występują elementy generujące opóźnienie w sposób zgrubny, a następnie dokładny. Układ wymaga ręcznego rozmieszczania elementów w FPGA oraz ich manualnego połączenia, co oznacza brak możliwości tworzenia układu na poziomie logiki. Należy więc odwoływać się do fizycznej struktury konkretnego układu FPGA. Dodatkowo należy wprowadzać ograniczenia dotyczące niepożądanej optymalizacji logicznej, fizycznej czy mapowania. Czasami nie wystarcza możliwość rozmieszczenia i połączenia elementów za pomocą edytora, wtedy konieczne jest stworzenie plików odpowiedzialnych za połączenia tzw. hard macro. Ponieważ opóźnienia w układzie są zależne od położenia elementów, dlatego ich rozmieszczenie jest bardzo ważne, niestety nie jest to proste, ponieważ narzędzia do projektowania dążą do zapewnienia jak najlepszych osiągów czasowych a nie zapewnienia równych opóźnień po różnych ścieżkach propagacji sygnału. W układzie element odpowiedzialny za zgrubne opóźnienie jest sterowany w czasie pracy układu, tak aby kompensować nierównomierności w występowaniu zer i jedynek, spowodowane przez zmiany temperatury czy napięcia. Układ dostrajający można znaleźć również w pracach [12, 13]. Odporność wbudowanych elementów na efekty metastabilne w nowych układach jest tak duża, że należy stworzyć własne zatrzaski w logice kombinacyjnej by móc skorzystać z tego fizycznego efektu oraz kontrolować jakość wyjścia w czasie rzeczywistym za pomocą sprzężenia zwrotnego [14].

5. Wnioski

Istniejące generatory wykorzystujące zjawisko metastabilności wymagają tworzenia elementów generujących precyzyjne opóźnienia [10 - 14], które bywają elementami zewnętrznymi, nie znajdującymi się w układzie [10, 12] lub programowalnymi o zmiennym opóźnieniu [11, 13]. Dla poprawy właściwości statystycznych stosuje się postprocessing. Użycie postprocessingu może oznaczać, że w istocie tylko małą część bitów spośród otrzymanych zawdzięczamy losowości (stanom metastabilnym), a znaczną zjawiskom deterministycznym. Ponieważ testy statystyczne nie odpowiedzą na pytanie, czy ciąg bitów wytworzyło źródło niedeterministyczne, czy deterministyczne, potrzebne są inne mechanizmy, np. ciągła detekcja stanu metastabilnego i przekazywanie dalej tylko bitów, gdy taki stan zaobserwujemy, czego nie ma w rozwiązaniach opisanych w literaturze.

Podstawowym wnioskiem z pracy jest stwierdzenie, że wykorzystanie seryjnie produkowanych układów FPGA do wytworzenia ciągu losowego w oparciu o stany metastabilne, chociaż możliwe, wydaje się być bezcelowe ze względu na trudności projektowe, technologiczne (brak powtarzalności) i otrzymane małe przepływności wyjściowe. Można sobie jednak wyobrazić stworzenie układu opartego o stany metastabilne w postaci układu ASIC (ang. *Application Specific Integrated Circuit*). Należy się jednak spodziewać dużej wrażliwości takiego układu na czynniki zewnętrzne takie jak temperatura, czy niestabilność napięcia. Dodatkowo wymagane będzie ręczne wykonanie planu layoutu oraz późniejsze trymowanie krzemu w celu uzyskania nieobciążonego wyjścia.

Pracę sfinansowano z projektów 08/83/DSMK/4706 i 08/83/DSPB/4707.

6. Literatura

[1] Vassilev A. Hall T. A.: The importance of entropy to information security, Computer, February 2014, s. 78-81.
[2] Golić J. D.: New methods for digital generation and postproccessing of random data, IEEE Trans., Computers., vol. 55, No 10, 2006, pp. 1217-1229.
[3] Dichtl M. and Golić J. D.: Hight speed true random number generation with logic gates only, Proc., Workshop Cryptograph. Hardware Embeded. Syst. CHES'2007, LNCS 4727, pp. 45-62, 2007.
[4] www.xilinx.com
[5] Rukhin A., Soto J., Nechvatal J., Smid M., Baker E., Leigh S., Levenson M., Vangel M., Banks D., Heckert A., Dray J., VO S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST special publication 800-22, National Institute of Standards and Technology, 2001, USA, Available at: http://csrc.nist.gov/rng
[6] QuickLogic Corp., Metastability report for FPGAs, Appl Note, 1997.
[7] Xilinx Corp., Metastable recovery, Appl Note XAPP 094, 1997.
[8] Lattice Semiconductor Corp., ispLSI/GAL Metastability Report, 1999.
[9] Kalisz J., Z. Jachna.: Metastability tests of flip-flops in programmable digital circuits, Microelectronics Journal, Vol. 37, Issue 2, February 2006, pp. 174–180.
[10] Danger J. L., Guilley S., P. Hoogvorst.: Fast true random generator in FPGAs, Proc. NEWCAS'07, 2007, pp. 506–509.
[11] Lozach F., Ben-Romdhame M., Graba T., Danger J-L.: FPGA Design of an Open-Loop True Random Number Generator, Proc. DSD 2013, pp. 615-622.
[12] Wieczorek P. Z, Golofit K.: Dual-Metastability Time-Competitive True Random Number Generator. IEEE Trans. on Circuits and Systems 61-I(1), Feb. 2014, pp. 134-145.
[13] Majzoobi M., Koushanfar F., Devadas S.: FPGA-based True Random Number Generation Using Circuit Metastability with Adaptive Feedback Control, CHES 2011, pp. 17-32
[14] Lee D., Seo H., Kim H.: Metastability-based Feedback Method for Enhancing FPGA-based TRNG, International Journal of Multimedia and Ubiquitous Engineering; 2014, Vol. 9 Issue 3, p. 235-248.

otrzymano / received: 22.04.2014
przyjęto do druku / accepted: 02.06.2014 *artykuł recenzowany / revised paper*