ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2013
Seria: TRANSPORT z. 80 Nr kol. 1895

Piotr KAWALEC[1]

ANALIZA METOD GENEROWANIA ZMIENNYCH WEJŚCIOWYCH DLA PROBABILISTYCZNYCH MODELI PROCESÓW TRANSPORTOWYCH

Streszczenie. W artykule przedstawiono analizę podstawowych metod generowania liczb losowych, wskazując na zalety sprzętowej realizacji takich generatorów. Wykorzystując generatory zbudowane na dwóch LFSR-ach opracowano i zbudowano szesnastokanałowy generator liczb pseudolosowych w strukturze FPGA. Zaproponowano wykorzystanie takiego generatora do budowy sprzętowych generatorów liczb losowych o rozkładach wykładniczym i normalnym.

Słowa kluczowe. Liczby losowe, generatory sprzętowe, rozkłady wykładniczy i normalny, HDL, FPGA

ANALYSIS OF METHODS OF GENERATING INPUT VARIABLES FOR PROBABILISTIC MODELS OF TRANSPORT PROCESSES

Summary. The article presents basic methods of random number generation indicating the advantages of hardware solutions. With the use of generators built on two linear feedback shift registers (LFSR), a 16-channel generator of pseudo-random numbers has been designed and constructed within the FPGA structure. On the basis of this multi-channel generator, hardware generators have been designed of a given distribution. The designed generators of exponential and normal distribution have been presented in detail.

Keywords. Random number, hardware generators, exponential and normal distributions, HDL, FPGA

1. WPROWADZENIE

W wielu systemach i procesach transportowych dane wejściowe (parametry strumienia zgłoszeń, struktura rodzajowa strumienia pojazdów itd.) oraz mechanizm ich przetwarzania (czas obsługi, przemieszczanie się pojazdów, rozpływ strumieni pojazdów na skrzyżowaniu itd.) mają charakter losowy. Dlatego też modele symulacyjne systemów i procesów transportowych są budowane jako modele probabilistyczne [1, 6, 9]. Przykładami takich modeli mogą być zarówno makroskopowe, jak i mikroskopowe modele przepływu ruchu,

[1] Politechnika Warszawska, Wydział Transportu, Zakład Sterowania Ruchem, Zespół Sterowania Ruchem Drogowym, ul. Koszykowa 75, 00-662 Warszawa, tel. 22 234 75 85, pka@it.pw.edu.pl

stosowane w symulatorach ruchu drogowego [2], w tym modele wykorzystujące automaty komórkowe [13].

Przy tworzeniu modeli procesów transportowych występuje wiele zmiennych wejściowych o charakterze losowym. Często zmienne te opisywane są wektorami losowymi, np. dla jednoznacznego określenia ruchu pojazdu samochodowego należy podać trzy współrzędne wektora losowego, opisujące: położenie, prędkość i przyśpieszenie pojazdu. Zwykle składowe takich wektorów losowych mają różne rozkłady. W procesie badań symulacyjnych takich modeli, powoduje to konieczność zastosowania wielowymiarowych zmiennych losowych o różnorodnych rozkładach. W procesie badań symulacyjnych stochastycznych modeli procesów transportowych należy zapewnić również niezależność długich ciągów wartości wejściowych wielu zmiennych losowych, co stawia odpowiednie wymagania generatorom takich ciągów.

Spełnienie wymienionych powyżej warunków, dotyczących generowanych losowych danych wejściowych oraz realizacji probabilistycznych algorytmów ich przetwarzania, ma istotny wpływ na poprawność uzyskiwanych wyników modelowania. Dlatego też jakość zastosowanych urządzeń lub procedur generowania zmiennych losowych ma zasadnicze znaczenie dla tworzonych stochastycznych modeli systemów i procesów transportowych. Jeśli w wyniku symulacji chcemy otrzymać wiarygodne estymatory parametrów modelowanych procesów, to należy zwrócić szczególną uwagę na zagadnienia generowania zmiennych losowych o zadanym rozkładzie.

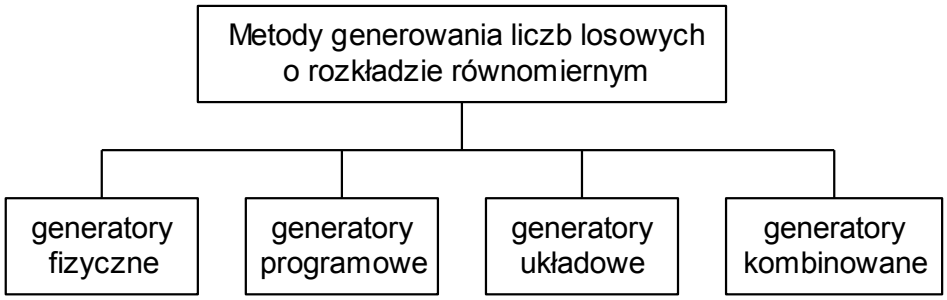
Do generowania zmiennych losowych o zadanym rozkładzie najczęściej wykorzystuje się metody: odwracania dystrybuanty, eliminacji lub superpozycji rozkładów [14]. Każda z nich metod wymaga zastosowania sprawdzonego i rzetelnego generatora niezależnych liczb losowych o rozkładzie równomiernym, a więc szczególną uwagę należy zwrócić na równomierność i niezależność liczb generowanych przez generatory o rozkładzie równomiernym i ich przydatność do tworzenia rozkładów wielowymiarowych oraz na szybkość generowania kolejnych liczb losowych.

Zwykle, jako źródło losowości wykorzystywane są procedury generowania liczb pseudolosowych o rozkładzie równomiernym, nazywane programowymi generatorami liczb pseudolosowych [5, 10, 14]. Ponieważ procedury generowania liczb pseudolosowych zawierają wszystkie języki programowania, to zwykle w procesie modelowania do generowania zmiennych losowych wykorzystywane są właśnie te generatory.

Alternatywę dla generatorów programowych mogą stanowić specjalizowane generatory sprzętowe, zwłaszcza realizowane w programowalnych strukturach logicznych FPLD [11]. Specjalizowane generatory umieszczone na kartach PCI mogą być dołączane do komputera, stanowiąc swoisty koprocesor, przeznaczony do generowania liczb losowych oraz pseudolosowych zarówno o rozkładzie równomiernym, jak i dowolnym. Uzasadnieniem takiego podejścia są wyniki przeprowadzonej analizy metod generowania liczb losowych i pseudolosowych.

2. METODY GENEROWANIA LICZB LOSOWYCH I PSEUDOLOSOWYCH

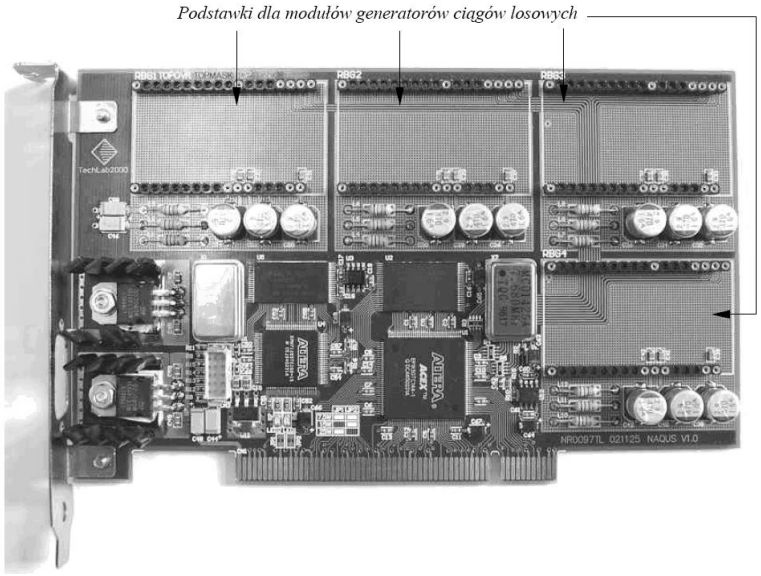
Do generowania zmiennych losowych o rozkładzie równomiernym mogą być wykorzystane różne rodzaje generatorów zarówno liczb losowych, jak i pseudolosowych (rys. 1).



Rys. 1. Klasyfikacja generatorów liczb losowych
Fig. 1. Classification of random number generators

2.1. Generatory fizyczne

Stosowane są do generowania ciągów losowych o rozkładzie dwupunktowym (zerojedynkowym), otrzymywanych ze zliczania fizycznych ciągów losowych lub z dyskretyzacji szumu białego. Przykładami generatorów fizycznych mogą być generatory zerojedynkowych ciągów losowych TL2RBG oraz HGL 98, w których źródłem losowości są szumy cieplne występujące w diodzie Zenera, pozwalające na uzyskanie szybkości generowania binarnych ciągów losowych rzędu setek kbit/s [3]. Umieszczenie kilku takich generatorów na specjalizowanej karcie PCI pozwala na korzystanie w symulacji komputerowej z kilku niezależnych źródeł ciągów losowych bądź na zwiększenie szybkości generowania liczb losowych. Na przykład system generacji ciągów losowych NAQUS pozwala umieścić na karcie PCI do czterech fizycznych generatorów liczb losowych o rozkładzie zerojedynkowym (rys. 2).



Rys. 2. Widok karty NAQUS
Fig. 2. View of a NAQUS board

Generatory te mogą pracować niezależnie, mogą być łączone w pary bądź mogą być wykorzystane do pracy równoległej, czterokrotnie zwiększając szybkość generowania liczb losowych. Cechą charakterystyczną generatorów fizycznych, oprócz stosunkowo niskiej

szybkości uzyskiwania z nich liczb losowych, są problemy z utrzymaniem symetryczności rozkładu zerojedynkowego, będące skutkiem zmian parametrów fizycznych generatora.

Z przedstawionych powodów generatory fizyczne, mimo w pełni losowego charakteru generowanych ciągów, zapewniającego ich nieprzewidywalność, nie znalazły szerszego zastosowania w symulacji komputerowej modeli stochastycznych.

2.2. Generatory programowe

W generatorach programowych liczby pseudolosowe generowane są na podstawie zależności matematycznych, zgodnie z założonym algorytmem produkcji takich liczb. W procedury generowania liczb pseudolosowych wyposażone są powszechnie stosowane języki programowania, przy czym algorytmy działania generatorów nie są najczęściej udostępniane. Jednak, ponieważ procedury generowania liczb pseudolosowych muszą zapewnić dużą szybkość działania generatora przy zastosowaniu prostych obliczeń, to można przyjąć, że są to zwykle generatory liniowe [10, 14].

Generatory liniowe stanowią efektywne źródło losowości, jednak jest to okupione zwykle mierną jakością generowanych liczb losowych. Na przykład w przeprowadzonym badaniu statystycznym jakości generatorów o rozkładzie równomiernym, stosowanych w siedmiu pakietach oprogramowania (Excel 5.0 Pl., Borland Pascal 7.0, Qbasic, Borland C++ 2.0, Quick Pascal 1.0, Ada GNAT 3.07 oraz Object Ada for Windows) odchylenia wartości średniej od wartości oczekiwanej przekraczały 30% [7]. Zatem więc zastosowanie takich generatorów w stochastycznych modelach procesów transportowych nie pozwala na uzyskiwanie miarodajnych ilościowych wyników symulacji. Można, co najwyżej mówić, o modelach probabilistycznych w sensie jakościowym.

Kolejnym ograniczeniem w zastosowaniu generatorów programowych jest trudność uzyskania z nich ciągów liczb losowych wymaganej długości. Stosowane głównie liniowe generatory multiplikatywne o postaci

$$X_{n+1} = cX_n \ (\text{mod}\, M) \text{ gdzie } M \text{ – okres generatora} \tag{1}$$

na komputerze o słowie 32-bitowym pozwalają uzyskać maksymalne ciągi o długości $2^{31} - 1$, co w wielu zastosowaniach może być niewystarczające. Dodatkowym problemem w stochastycznych modelach symulacyjnych są trudności z zastosowaniem liniowych generatorów programowych do generowania wielowymiarowych zmiennych losowych. Stwierdzenie to wynika stąd, że kolejne liczby losowe, otrzymywane z liniowych generatorów programowych, nie mogą być wykorzystywane do generowania współrzędnych wektorów losowych o rozkładzie równomiernym, ponieważ nie są one rozmieszczone równomiernie w k-wymiarowej kostce jednostkowej.

W [12] przedstawiono ilościową miarę nierównomierności generatorów liniowych, z której wynika, że wszystkie liczby, otrzymywane z takiego generatora, są rozmieszczone co najwyżej na

$$\sqrt[k]{k!M} \tag{2}$$

hiperpłaszczyznach, przecinających kostkę jednostkową w k-wymiarowej przestrzeni (gdzie k - wymiar rozkładu, M - okres generowanego ciągu). Z zależności (2) wynika, że w (stosowanych obecnie w komputerach klasy PC) generatorach o zakresie $M = 2^{31} - 1$, dla k = 3, wygenerowane w maksymalnym okresie generatora ponad 700 milionów wektorów losowych o trzech współrzędnych, będzie rozmieszczonych co najwyżej na 2344

płaszczyznach, przecinających sześcian jednostkowy. W przypadku wektorów losowych o większej liczbie współrzędnych nierównomierność jest jeszcze bardziej widoczna (np. dla k = 8, wygenerowane w maksymalnym okresie 268 milionów wektorów losowych będzie rozmieszczonych na co najwyżej 55 hiperpłaszczyznach).

Taka nierównomierność w rozmieszczeniu wektorów losowych utrudnia wykorzystanie liniowych generatorów programowych do generowania wielowymiarowych zmiennych losowych, pozwalając, jako losowe traktować tylko t = ent [m/k] starszych bitów słowa (gdzie m - liczba bitów słowa). Dla stosowanych obecnie komputerów długości słów wynoszą m = 32, a więc dla rozkładów trójwymiarowych, jako losowe możemy traktować tylko dziesięć starszych bitów, w efekcie wykluczone jest uzyskanie satysfakcjonującej dokładności wyników symulacji.

Ponieważ w stochastycznych modelach procesów transportowych konieczne jest zwykle generowanie zmiennych losowych na wielu wejściach modelu, przy wykorzystaniu jednego generatora programowego, powstają problemy z zapewnieniem niezależności generowanych zmiennych losowych oraz uzyskania wymaganego maksymalnego okresu pracy generatora. Ogólnie problemy te mogą być rozwiązywane poprzez zastosowanie większej liczby generatorów oraz zastosowania innych metod generowania liczb losowych. Na przykład generatory wykorzystujące operację odejmowania z pożyczką (SWB) pozwalają, w kombinacji z generatorami liniowymi, otrzymać znacznie dłuższe okresy (np. generator ULTRA pozwala otrzymać okres 10^{344}), jednak wymagają one specjalnej procedury inicjalizacji i są skomplikowane obliczeniowo. Natomiast w generatorach wykorzystujących operację mnożenia z przeniesieniem (MWC) konieczność wielokrotnego wykonywania operacji mnożenia znacznie wydłuża proces generowania liczb [14].

Z przedstawionej analizy generatorów programowych o rozkładzie równomiernym wynika, że w modelach stochastycznych mogą być stosowane generatory o prostej produkcji niskiej jakości liczb pseudolosowych albo generatory o zadowalających parametrach, uzyskiwanych jednak kosztem skomplikowanych procedur obliczeniowych. Procedury generowania liczb pseudolosowych umieszczonych w językach programowania należy zaliczyć do tej pierwszej grupy generatorów. Tak więc do wykorzystania tych generatorów w modelowaniu procesów transportowych należy podchodzić z dużą ostrożnością, jeśli wyniki symulacji chcemy traktować jako miarodajne. Natomiast zarówno zastosowanie wielu generatorów, jak i generatorów wykorzystujących skomplikowane procedury obliczeniowe w istotny sposób wpłynie na efektywność procesu symulacji, zwłaszcza jeśli uwzględnimy konieczność generowania w stochastycznych modelach procesów transportowych wielu danych wejściowych o złożonych, wielowymiarowych rozkładach.

2.3. Generatory układowe i generatory kombinowane

Alternatywą dla generatorów programowych mogą być generatory układowe, w których liczby pseudolosowe są uzyskiwane również zgodnie z zadanym algorytmem, jednak algorytm produkcji liczb realizowany jest w sposób sprzętowy. Dzięki temu generowanie zmiennych losowych nie obciąża procesora, może być realizowane współbieżnie z procesem symulacji, a więc możliwe jest stosowanie złożonych algorytmów generacji bez obniżania efektywności stochastycznych modeli symulacyjnych [9].

Generatory układowe, ze względu na brak zależności między długością słowa i maksymalnym okresem generatora pozwalają w sposób prosty uzyskiwać ciągi pseudolosowe dowolnej długości. Dodatkową zaletą generatorów układowych jest możliwość budowy generatorów wielokanałowych, co pozwala na łatwe generowanie niezależnych wielowymiarowych zmiennych losowych, bowiem każda ze współrzędnych wektora

losowego może być generowana w oddzielnym kanale. Zastosowanie układów specjalizowanych zawierających generatory układowe w stochastycznych modelach symulacyjnych procesów transportowych pozwala na uzyskanie wielu efektywnych źródeł losowości o zadanych parametrach. Zagadnienia związane z analizą i syntezą specjalizowanych układów do generowania liczb losowych zostaną szczegółowo rozpatrzone w rozdziale 3.

Generatorami kombinowanymi nazywane są generatory będące kombinacją generatorów układowych oraz fizycznych. Tak budowane układy specjalizowane do generowania zmiennych losowych pozwalają uniknąć ograniczeń, charakterystycznych dla generatorów fizycznych i układowych, stanowiąc w ten sposób doskonałe źródło zmiennych zarówno pseudolosowych, jak i losowych. Ogólnie generatory fizyczne, generatory układowe oraz kombinowane będziemy zaliczać do sprzętowych generatorów liczb losowych.

3. GENERATORY UKŁADOWE O ROZKŁADZIE RÓWNOMIERNYM NA LFSR

W specjalizowanych układach cyfrowych, budowanych zarówno z elementów standardowych, jak i w reprogramowalnych strukturach logicznych FPLD, generatory układowe mogą być realizowane na liniowych rejestrach przesuwających LFSR (Linear Feedback Shift Register), opisywanych ogólnie następującym wielomianem charakterystycznym:

$$x^p a_p \oplus x^{p-1} a_{p-1} \oplus, ..., \oplus x^i a_i \oplus, ..., \oplus x a_1 \oplus a_0 , \tag{3}$$

gdzie $\oplus$ jest operacją sumy mod2.

Aby generator zbudowany na takim rejestrze mógł pracować z maksymalnym okresem równym $M = 2^p - 1$ (gdzie p – długość rejestru), wielomian charakterystyczny (3), opisujący działanie rejestru liniowego, musi być nieredukowalnym wielomianem pierwotnym.

W modelowaniu procesów transportowych konieczne jest stosowanie generatorów o wysokim maksymalnym okresie, a więc zastosowanie generatorów, opisywanych ogólną postacią wielomianów charakterystycznych, jest bardzo utrudnione. Problem wyznaczania nieredukowalnych pierwotnych wielomianów charakterystycznych ulega znacznemu uproszczeniu, jeżeli w wyrażeniu (3) tylko dwa współczynniki a_i oraz wyraz wolny a_0 są różne od zera i przyjmują wartość 1. Postać wielomianu charakterystycznego będzie wówczas następująca

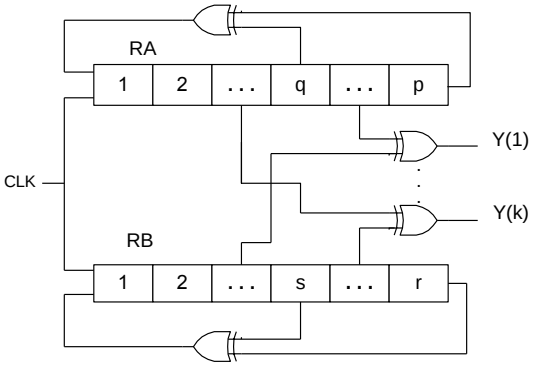
$$x^p \oplus x^q \oplus 1. \tag{4}$$

Dla wielomianu charakterystycznego postaci (4) dostępne są w literaturze w postaci tabel [9, 14] przykładowe wartości parametrów p i q, przy których generowane są ciągi binarne o maksymalnym okresie $M = 2^p - 1$ (zakładamy, że $p > q$).

Najprostszą realizację sprzętową generatora liczb pseudolosowych (GLP) o rozkładzie równomiernym, który produkuje liczby wielobitowe w sposób równoległy, uzyskamy stosując kombinację dwóch przedstawionych poprzednio generatorów na LFSR. Każdy z rejestrów przesuwających RA i RB objęty jest sprzężeniem zwrotnym z dwuwejściowymi elementami logicznymi XOR (rys. 3).

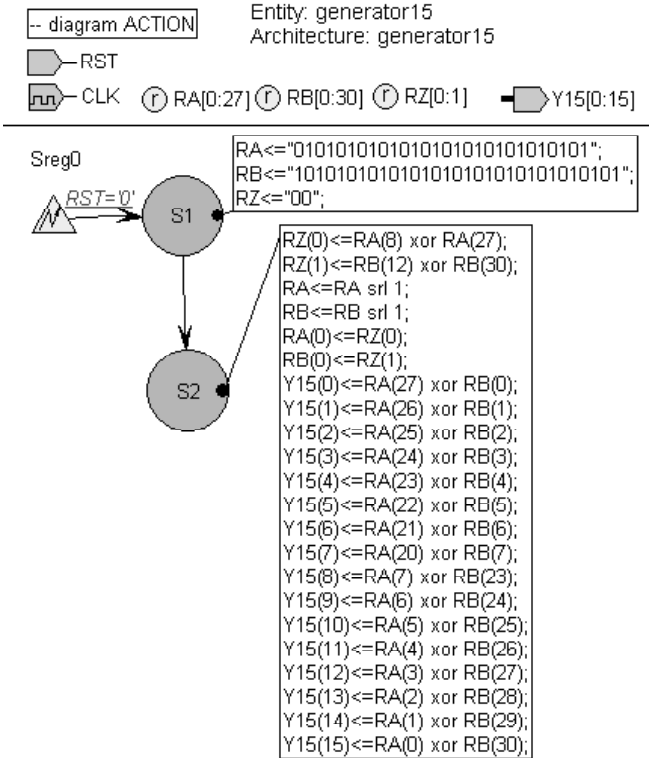
k – bitowa liczba pseudolosowa Y formowana jest w każdym takcie zegarowym clk na wyjściach elementów XOR, których wejścia są w odpowiedni sposób połączone z odczepami rejestrów RA oraz RB ($k < \min(p, r)$). Zastosowanie dwóch rejestrów przesuwających ułatwia

uzyskanie wielobitowych liczb pseudolosowych oraz poprawia własności statystyczne tak zbudowanego generatora.



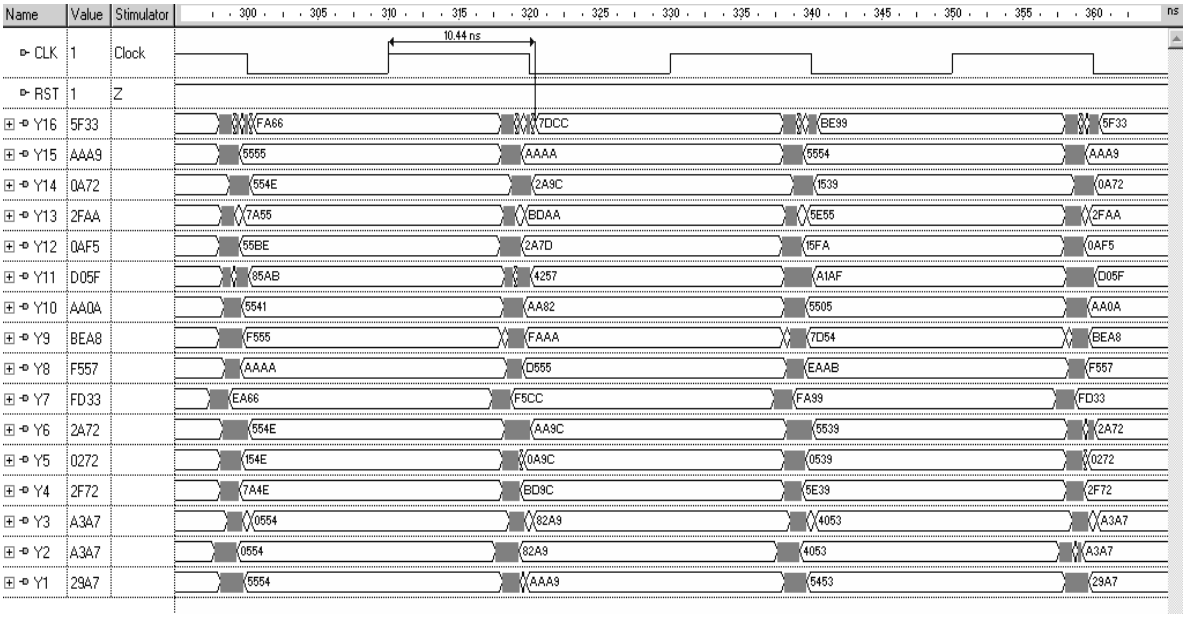
Rys. 3. Schemat generatora na dwóch rejestrach przesuwających
Fig. 3. Diagram of pseudorandom-number generator on two LFSRs

Specyfikacja generatorów w języku VHDL, a następnie ich implementacja w programowalnych strukturach logicznych FPLD pozwala na ich jednoukładową realizację oraz na uzyskanie wysokiej szybkości generowania liczb pseudolosowych. Na rys. 4 pokazano specyfikację przykładowego generatora w postaci grafu przejść automatu skończonego, z wykorzystaniem edytora FSM pakietu Active-HDL. Specyfikację GLP przeprowadzono przyjmując odpowiednie długości LFSR-ów oraz stosując, dla stabilnej pracy generatora, dodatkowy rejestr RZ, w którym przechowywane są wartości sygnałów sprzężeń zwrotnych.



Rys. 4. Specyfikacja generatora w edytorze grafów przejść FSM
Fig. 4. Generator specification by means of FSM editor

Aby było możliwe generowanie wielowymiarowych zmiennych losowych, w analogiczny sposób opracowano 16 szesnastobitowych generatorów, przyjmując dla każdego z nich różne długości rejestrów i wielomiany charakterystyczne (4), zapewniające osiągnięcie maksymalnych okresów. Tak utworzone generatory poddano weryfikacji, testowaniu, a następnie po zaimplementowaniu w układach FPGA przeprowadzono symulację czasową (rys. 5). Dla prototypu generatora zaimplementowanego na stanowisku badawczym Virtual Workbench VW-300 uzyskano częstotliwość pracy powyżej 70 MHz, z wykorzystaniem zasobów układu FPGA XCV300BG352C na poziomie 25%. Maksymalne okresy generatorów w poszczególnych kanałach wyniosły od 12 h do 750 lat, przy generowaniu liczb pseudolosowych w każdym z kanałów z częstotliwością 50 MHz [9].



Rys. 5. Przebiegi symulacji czasowej generatora wielokanałowego
Fig. 5. Timing simulation courses of multi channel generator

4. UKŁADOWE GENERATORY O ROZKŁADZIE WYKŁADNICZYM

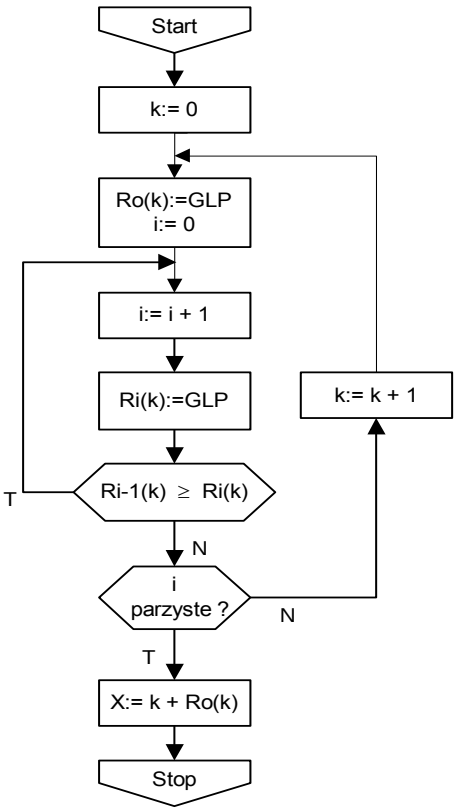
Rozkład wykładniczy jest powszechnie stosowany w analizie niezawodnościowej i eksploatacyjnej urządzeń i systemów technicznych, w tym układów i systemów sterowania [4]. Natomiast w modelowaniu przepływu ruchu rozkładem wykładniczym jest opisywany odstęp miedzy poruszającymi się swobodnie pojazdami w ruchu drogowym, a odstępy między pojazdami w strumieniu zawierającym zarówno pojazdy poruszające się swobodnie, jak i podążające za poprzednikiem są opisywane sumą przesuniętych rozkładów wykładniczych [6].

W układowej realizacji generatora o rozkładzie wykładniczym wykorzystano metodę serii monotonicznych J. von Neumanna, polegającą na rejestrowaniu monotonicznych serii w ciągu zmiennych losowych o rozkładzie równomiernym na przedziale (0,1). Rejestrując kolejne serie postaci

$$R_0 \geq R_1 \geq R_2 \geq \ldots \geq R_n < R_{n+1}, \tag{5}$$

numerujemy je kolejno. Numer pierwszej serii, w której n będzie liczbą nieparzystą, przyjmujemy za część całkowitą, natomiast wartość R_0 w tej serii za część ułamkową generowanej liczby losowej o rozkładzie równomiernym [14].

Na schemacie blokowym algorytmu (rys. 6) przedstawiono kolejne kroki generowania liczby losowej o rozkładzie wykładniczym. Po starcie zerowany jest licznik numeru serii k i pobierana jest pierwsza liczba $R_0(k)$ z generatora liczb pseudolosowych GLP o rozkładzie równomiernym. Po pobraniu następnej liczby $R_i(k)$ (gdzie i – numer kolejnej liczby w serii o numerze k) z generatora, następuje porównanie liczb i jeśli pobrana liczba jest mniejsza lub równa od poprzedniej, proces tworzenia serii nierosnących liczb losowych jest kontynuowany, aż do momentu, gdy kolejno pobrana liczba losowa okaże się większa od poprzedniej. W tym przypadku proces rejestrowania serii monotonicznej zostaje zakończony i jeśli i –1 jest liczbą nieparzystą, to na wyjściu tworzymy liczbę losową X, której część całkowitą stanowi numer serii k, natomiast część ułamkową - pierwsza liczba danej serii $R_o(k)$. Natomiast, gdy i – 1 jest liczbą parzystą, rozpoczyna się rejestracja kolejnej serii monotonicznej.

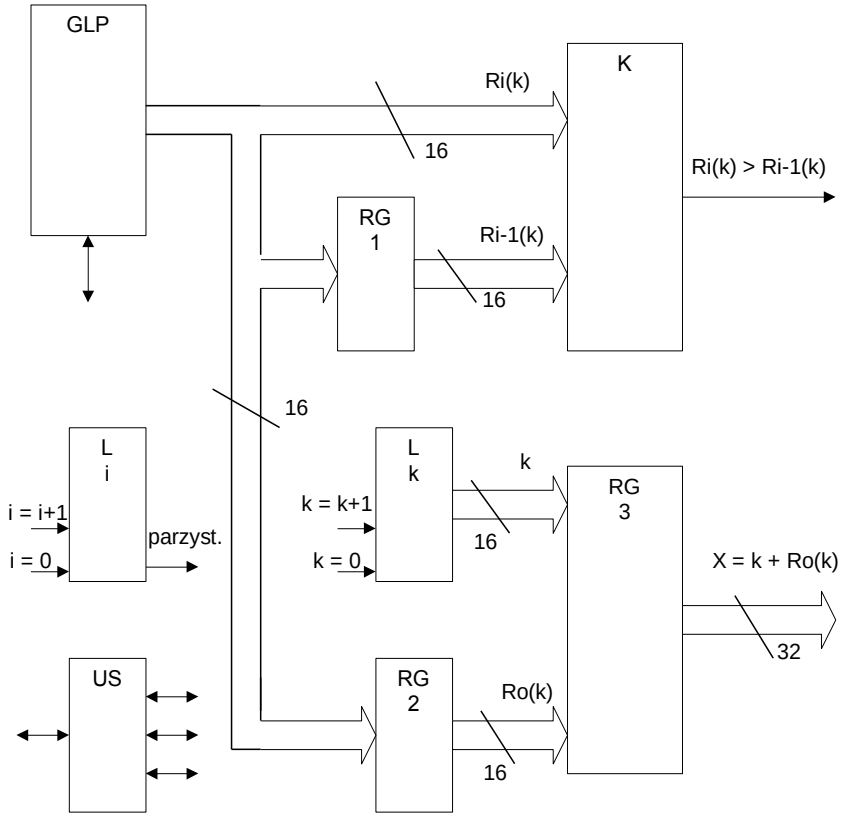


Rys. 6. Algorytm generowania liczb losowych o rozkładzie wykładniczym
Fig. 6. Algorithm operations of pseudorandom-number generation with exponential distribution

W realizacji sprzętowej generatora o rozkładzie wykładniczym zastosowano trzy rejestry równolegle RG1, RG2, RG3, komparator K oraz dwa liczniki Li, Lk (rys. 7). Jako generator liczb pseudolosowych GLP o rozkładzie równomiernym wykorzystano sprzętowy generator liczb pseudolosowych na LFSR-ach. Sygnały sterujące pracą poszczególnych bloków wytwarzane są w układzie sterującym US na podstawie sygnałów zewnętrznych i sygnałów kontrolnych z bloków układu (linie sygnałów sterujących na schemacie zostały pominięte).

Po uruchomieniu układu następuje wyzerowanie licznika numeru serii Lk i licznika numeru liczby Li oraz wpisanie pierwszej liczby losowej R_0 z generatora GLP do rejestrów RG1 i RG2 (rys. 7). W następnym takcie, po zwiększeniu zawartości licznika Li o jeden, następuje porównanie w komparatorze K kolejno uzyskanej liczby losowej R_1 z generatora

GLP z liczbą poprzednią. Jeśli R_1 jest mniejsze od R_0, to liczba R_1 zostaje zapisana do rejestru RG1 i proces rejestracji serii monotonicznej jest kontynuowany do momentu wystąpienia sytuacji, gdy kolejno wygenerowana liczba losowa o rozkładzie równomiernym okaże się większa od poprzedniej. Wówczas, na podstawie sygnału z komparatora K układ sterujący US sprawdza czy zawartość licznika Li jest liczbą parzystą. Realizowane jest to poprzez kontrolę wyjścia licznika o najniższej wadze (sygnał parzyst. na schemacie). Zerowa wartość na tym wyjściu odpowiada liczbie parzystej.



Rys. 7. Schemat blokowy generatora
Fig. 7. Diagram of generator

Po stwierdzeniu, że zawartość licznika Li jest liczbą parzystą (oczywiście i – 1 jest liczbą nieparzystą), do rejestru RG3 jest przepisywana zawartość licznika numeru serii Lk oraz zawartość rejestru RG2. W ten sposób w rejestrze RG3 jest formowana liczba losowa o rozkładzie wykładniczym, której część całkowitą stanowi numer aktualnie rejestrowanej serii monotonicznej k (przechowywanej w liczniku Lk), natomiast część ułamkową - pierwsza liczba losowa o rozkładzie równomiernym rejestrowanej serii R_0 (przechowywana w rejestrze RG2). Jeśli natomiast zawartość licznika nie jest liczbą parzystą, zawartość licznika numeru serii Lk zostaje zwiększona o jeden, licznik Li zostaje wyzerowany i poprzez pobranie z generatora liczb losowych o rozkładzie równomiernym GLP pierwszej liczby R_0 następuje proces rejestracji kolejnej serii monotonicznej.

Wykorzystując edytor schematów blokowych BDE, poszczególne bloki generatora o rozkładzie wykładniczym zostały wyspecyfikowane w języku VHDL, a następnie cały generator został syntezowany i zaimplementowany w jednym z kanałów generatora wielokanałowego. Dzięki temu w jednym układzie FPGA mogą być umieszczone nie tylko generatory o rozkładzie równomiernym, lecz również generatory o zadanym rozkładzie,

w których GLP o rozkładzie równomiernym będą wykorzystane, jako źródło losowości, w algorytmach generowania zadanego rozkładu.

Umieszczenie w układzie FPGA XCV300BG352C dodatkowo generatora o rozkładzie wykładniczym zwiększyło wykorzystanie zasobów struktury o 4%. Natomiast struktura programowalna, zawierająca wielokanałowy generator o rozkładzie równomiernym oraz generator o rozkładzie wykładniczym, dalej może pracować z częstotliwością 50 MHz, bowiem dodanie generatora o rozkładzie wykładniczym zmniejszyło maksymalną częstotliwość taktowania układu FPGA o 5 MHz.

Pozostałe niewykorzystane zasoby układów FPGA pozwalają na implementację w każdym z nich innych generatorów o zadanym rozkładzie. Jednak, dla zachowania wysokiej efektywności generowania liczb pseudolosowych o zadanym rozkładzie, należy spośród algorytmów ich generowania wybierać takie, aby nie były one złożone numerycznie i proste do realizacji sprzętowej. Przykładem może być przedstawiona poniżej metoda generowania liczb pseudolosowych o rozkładzie normalnym.

5. UKŁADOWE GENERATORY O ROZKŁADZIE NORMALNYM

W modelowaniu wielu procesów technicznych, w tym procesów transportowych, istotne są zmienne losowe o rozkładzie normalnym. Rozkład normalny mają np.: odstępy między pojazdami w kolumnie, prędkości pojazdów w ruchu swobodnym, losowe błędy pomiarów i obserwacji, losowe zakłócenia w kanałach sterowania i transmisji danych itd. Dodatkowo rozkład normalny stanowi dobre przybliżenie dla wielu innych rozkładów, np. rozkładu dwumianowego.

Uwzględniając wykorzystanie opracowanego, efektywnego, wielokanałowego generatora niezależnych liczb pseudolosowych o rozkładzie równomiernym, w algorytmie generowania liczb pseudolosowych wykorzystano centralne twierdzenie graniczne [4]. Zgodnie z tym twierdzeniem, jeżeli X_1, X_2, … , są niezależnymi zmiennymi losowymi o tym samym rozkładzie, mającym wartość oczekiwaną μ i standardowe odchylenie $\sigma > 0$, to zmienna losowa Z

$$Z = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}} \tag{6}$$

ma rozkład asymptotycznie normalny N(0,1).

Dla rozkładu równomiernego wartość oczekiwana wynosi $E(X) = 0,5$, natomiast wariancja $Var(X) = 1/12$. Podstawiając te wartości do wyrażenia (6), otrzymujemy

$$Z_l = \frac{\sum_{i=1}^{n} X_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \tag{7}$$

Generator pracujący zgodnie z zależnością (7) jest efektywnym źródłem zmiennych losowych o rozkładzie normalnym dla n = 12 („prawo tuzina”) [14], bowiem unika się wówczas konieczności realizacji operacji logarytmowania i dwukrotnego dzielenia [8], a realizacje zmiennej losowej *Z* generowane są zgodnie z zależnością

$$Z_l = \sum_{i=1}^{12} X_i - 6. \tag{8}$$

Mając do dyspozycji szesnastokanałowy generator liczb pseudolosowych o rozkładzie równomiernym możemy, dla uzyskania maksymalnej szybkości generowania liczb pseudolosowych X oraz dodatkowego zapewnienia ich niezależności, pobierać liczby pseudolosowe wygenerowane w jednym takcie zegarowym na wyjściach 12 kanałów generatora wielokanałowego. W sprzętowej realizacji zależności (8), implementując w układzie FPGA kaskadowo 10 sumatorów oraz układ do odejmowania liczb (subtraktor), możemy wygenerować realizację zmiennej losowej Z_l o rozkładzie normalnym, w jednym takcie zegarowym z opóźnieniem równym czasowi propagacji sygnałów w układzie (a więc z opóźnieniem liczonym w nanosekundach). Tak więc sprzętowa realizacja algorytmu generowania liczb o rozkładzie normalnym w układach FPGA pozwala uzyskać efektywne i bardzo szybkie źródło losowości o zadanym rozkładzie.

Jednak w niektórych zastosowaniach, zwłaszcza, jeśli korzysta się z „ogona" rozkładu normalnego, dokładność przybliżenia rozkładu normalnego w generatorach wykorzystujących „prawo tuzina" może być za niska. Jak przedstawiono powyżej, zwiększenie n doprowadzi, w ogólnym przypadku, do znacznego wzrostu złożoności obliczeniowej wyznaczania realizacji zmiennej losowej Z.

Uwzględniając, że operację dzielenia liczby stałoprzecinkowej przez liczbę 2^k można, w realizacji sprzętowej, sprowadzić do przesunięcia dzielnej w prawo o k bitów, założymy, że n będzie przyjmować wartości $n = 12 \cdot 2^{2k}$ (dla k = 0,1,2…). Wówczas, po podstawieniu tej wartości n do (7), otrzymujemy

$$Z_l = \frac{\sum_{i=1}^{12 \cdot 2^{2k}} X_i - 6 \cdot 2^{2k}}{2^k}. \tag{9}$$

Dla $k = 0$ liczba sumowanych zmiennych losowych o rozkładzie równomiernym X wyniesie 12, a więc wyrażenie (9) przekształca się w „prawo tuzina", przyjmując postać (8).

Dla $k = 1$ wyrażenie (9) przyjmie postać

$$Z_l = \frac{\sum_{i=1}^{48} X_i - 24}{2}. \tag{10}$$

Generowanie i przetwarzanie 48 liczb pseudolosowych o rozkładzie równomiernym będzie wykonane w 3 taktach zegarowych, natomiast dzielenie przez 2 sprowadzi się do przesunięcia w prawo o jeden bit wyniku przetwarzania, w celu uzyskania liczby pseudolosowej Z_l o rozkładzie normalnym.

W ten sposób przedstawiony generator pozwala uzyskać zmienne losowe o rozkładzie aproksymującym rozkład normalny z dowolnie wysoką dokładnością. Dzięki wykorzystaniu wielokanałowego generatora o rozkładzie równomiernym oraz realizacji opracowanego generatora w układach FPGA uzyskujemy efektywne źródło losowości o rozkładzie normalnym.

6. PODSUMOWANIE

W wyniku przeprowadzonej analizy metod generowania zmiennych losowych do probabilistycznych modeli procesów transportowych wskazano, że wykorzystanie do tego celu generatorów programowych stwarza wiele problemów. Dotyczą one zarówno efektywności takich modeli, jak i rzetelności uzyskiwanych wyników modelowania. Zastosowanie do tego celu generatorów układowych pozwala odciążyć procesor od generowania zmiennych losowych, dodatkowo zapewniając wyższą równomierność i niezależność generowanych liczb. Natomiast posiadanie efektywnych, wielokanałowych generatorów o rozkładzie równomiernym pozwala budować generatory o zadanym rozkładzie. Zasoby współczesnych układów FPGA pozwalają na rozmieszczanie w nich zarówno generatorów o rozkładzie równomiernym, jak i generatorów o innych rozkładach, wymaganych w modelowaniu np. różnorodnych procesów transportowych.

Bibliografia

1. Adamski A.: Inteligentne systemy transportowe. Wydawnictwa AGH, Kraków 2003.
2. Algers S. i in.: Review of Micro-Simulation Models. SMARTEST Project Deliverable No 3, European Commission DGVII, Brussels 1997.
3. Bednarski J., Kuncewicz W., Iwanicki A.: NAQUS – System generacji ciągów losowych. TechLab2000 Sp. z o.o., Warszawa 2005.
4. Bobrowski D.: Probabilistyka w zastosowaniach technicznych. WNT, Warszawa 1986.
5. Fishman G.S.: Symulacja komputerowa. Pojęcia i metody. PWE, Warszawa 1981.
6. Gaca S., Suchorzewski W., Tracz M.: Inżynieria ruchu drogowego. Teoria i praktyka. WKŁ, Warszawa 2008.
7. Górecka A., Szmit M.: Badanie jakości wybranych programowych generatorów liczb pseudolosowych. Materiały IV Krajowej Konferencji „Komputerowe wspomaganie badań naukowych KOWBAN'97". Wrocław – Świeradów Zdrój 1997, s. 181–186.
8. Jermakow S.M.: Metoda Monte Carlo i zagadnienia pokrewne. PWN, Warszawa 1976.
9. Kawalec P.: Analiza i synteza specjalizowanych układów modelowania i sterowania ruchem w transporcie. Prace Naukowe Transport, z. 68, OWPW, Warszawa 2009.
10. Knut D.E.: The Art of Computer Programming. vol. 2 Seminumerical Algorithms. 3rd ed., Addison-Wesley 2007.
11. Łuba T., Jasiński K., Zbierzchowski B.: Układy specjalizowane w strukturach CPLD i FPGA., WKŁ, Warszawa 1997.
12. Marsaglia G.: Xorshift RNGs. Jurnal of Statistical Software, Vol. 8, Iss.14, 2003.
13. Tampere C., van Arem B.: Traffic flow theory and its applications in automated vehicle control: a review. Intelligent Transportation Systems, Proceedings 2001 IEEE, pp. 391–397.
14. Wieczorkowski R., Zieliński R.: Komputerowe generatory liczb losowych. WNT, Warszawa 1997.