VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

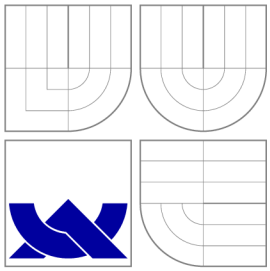
GENERÁTORY NÁHODNÝCH ČÍSEL PRO KRYPTOGRAFII

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

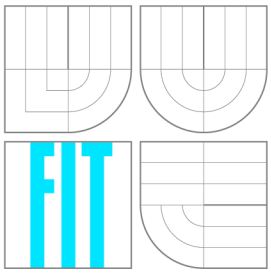
AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV MATĚJÍČEK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

GENERÁTORY NÁHODNÝCH ČÍSEL PRO KRYPTOGRAFII

RANDOM NUMBERS GENERATORS FOR CRYPTOGRAPHY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV MATĚJÍČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Doc.Dr.Ing. PETR HANÁČEK

BRNO 2012

Abstrakt

Obsahem této diplomové práce je návrh a statistické testy dvou různých hardwarových generátorů náhodných čísel. Obsahuje také přehled používaných zdrojů entropie, algoritmů používaných pro korekci odchylky od normálního rozložení a popis používaných statistických testů.

Abstract

The content of this thesis is the design and statistical tests of two different hardware random number generators. It also includes an overview of the sources of entropy, algorithms used to correct deviations from the normal distribution and the description of statistical tests.

Klíčová slova

Generátor náhodných čísel, Entropie, Šum, Statistické testy, Chí kvadrát.

Keywords

Random number generator, Entropy, Noise, Statistical tests, Chi square.

Citace

Jaroslav Matějíček: Generátory náhodných čísel pro kryptografii, diplomová práce, Brno, FIT VUT v Brně, 2012

Generátory náhodných čísel pro kryptografii

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Petra Hanáčka.

.....
Jaroslav Matějíček
23. května 2012

Poděkování

Na tomto místě bych chtěl vyjádřit poděkování Doc. Dr. Ing. Petru Hanáčkovi za vedení a konzultace.

Obsah

Kapitola 1

Úvod

Důležitou, byť mnohdy opomíjenou součástí mnoha systémů jsou generátory náhodných čísel. V mnoha aplikacích, jako například v kryptografii, hazardních hrách i jiných jde přitom o naprosto kritickou součást, jejíž nedostatky mohou mít vážné důsledky pro bezpečnost celého, mnohdy jinak výborně navrženého systému. Případů kdy útok na nevhodně navržený generátor vedl k bezpečnostnímu incidentu lze nalézt mnoho.

Tato práce se věnuje návrhu dvou kryptograficky bezpečných generátorů pravých náhodných čísel a jejich důkladnému testování.

Práce je rozdělena na sedm kapitol. První je věnována úvodu do problematiky a vysvětlení důležitých pojmů.

Druhá kapitola se věnuje různým zdrojům náhody, jejich fyzikální podstatě a hodnocení vhodnosti pro použití v generátoru náhodných čísel.

Třetí kapitola je zaměřena na algoritmy které pomáhají napravovat některé nedostatky fyzických zdrojů entropie.

Čtvrtá kapitola se věnuje problematice testování generátorů. Obsahuje přehled používaných statistických testů a skupin testů.

Pátá a šestá kapitola se věnuje samotnému návrhu generátorů náhodných čísel, popisu jejich funkce a statistickým testům které jsou použity pro ověření jejich funkčnosti a zhodnocení kvality realizovaných generátorů.

Sedmá kapitola obsahuje zhodnocení realizovaných generátorů.

1.1 Náhodná čísla

Náhodná čísla mají v současnosti mnoho uplatnění v praktických i výzkumných oblastech. Náhodnými čísly v širším slova smyslu myslíme posloupnost čísel, jejíž prvky jsou hodnotami náhodné veličiny. Výskyt určitého čísla na určité pozici posloupnosti je náhodný jev a není možno ho s jistotou předvídat, možné je pouze určit pravděpodobnost jeho výskytu. Můžeme si uvést nejvýznamnější oblasti jejich použití:

- Kryptografie - pro kryptografii a její aplikace v informační bezpečnosti jsou náhodná čísla mimořádně důležitá. Jako příklady si můžeme uvést:
  - Použití identifikátorů v autentizačních krocích distribuce klíčů,
  - Generování klíčů relací v symetrickém šifrování,
  - Generování klíčů pro asymetrické šifrování,

  - Generování digitálního podpisu na bázi algoritmu DSA.

- Simulace - Simulací se vytváří většinou zjednodušené podmínky co nejlépe kopírující klíčové vlastnosti původních jevů, abychom mohli zkoumat procesy, které by bylo nákladné, nebezpečné nebo z různých jiných důvodů náročné otestovat v reálných podmínkách. Zejména při simulacích přírodních jevů je kvalitní zdroj náhodných čísel klíčovým pro důvěryhodnou simulaci reality. Simulace jsou dnes nepostradatelné v řadě oblastí, počínaje studiem jaderné fyziky či simulací počasí a konče simulací života a vývoje různých populací či simulacemi v astronomii.

- Vzorkování (náhodný výběr, sampling) - zkoumání vzorku vzorkováním bývá často levnější, rychlejší a někdy dokonce není ani technicky možné zkoumat celý základní soubor. Nejčastěji slouží k získání informací o velkém souboru dat bez nutnosti vynaložení velkého množství nákladů. Mezi základní druhy náhodného výběru patří jednoduchý náhodný výběr, stratifikovaný výběr, systematický výběr a jiné. Vzorkování má široké uplatnění v mnoha oblastech, ať už praktických či vědeckých.

- Metody Monte Carlo - třída algoritmů pro simulaci systémů. Jedná se o stochastické metody používající náhodná čísla. Typicky bývají využívány pro výpočet integrálů, hlavně vícerozměrných, kde běžné metody nejsou efektivní. Základní myšlenka této metody je velmi jednoduchá, chceme určit střední hodnotu veličiny, která je výsledkem náhodného děje. Vytvoří se počítačový model tohoto děje a po proběhnutí dostatečného množství simulací se mohou data zpracovat klasickými statistickými metodami.

- Hry a loterie - V mnoha hrách je třeba simulovat například promíchání karet, hod kostkou či roztáčení rulety. A tato simulace musí odpovídat realitě a tedy být náhodná, aby hráči nemohli předvídat výsledky.

1.2 Generátor náhodných čísel

Generování náhodných čísel má charakter generování posloupností náhodných čísel, na které jsou kladené dva základní požadavky: náhodnost a nepředvídatelnost.

Náhodnost je vlastnost, která vyjadřuje statistické vlastnosti posloupnosti náhodných čísel. Pro posouzení náhodnosti posloupnosti náhodných čísel se používají dvě kritéria:

- Rovnoměrnost - rozdělení vyjadřuje distribuci náhodných čísel v posloupnosti, která by měla být rovnoměrná, což znamená, že četnost výskytu každého náhodného čísla v této posloupnosti by měla být stejná,

- Statistická nezávislost - znamená, že žádné náhodné číslo v posloupnosti není závislé na žádném jiném čísle v této posloupnosti.

Nepředvídatelnost je vlastnost, která vyjadřuje vlastnost posloupnosti náhodných čísel, že sousední čísla v této posloupnosti nejsou předvídatelné, tj. ze znalosti jednoho, resp. více náhodných čísel posloupnosti nelze předpovědět další náhodné číslo této posloupnosti. Ve statisticky nezávislé náhodné posloupnosti žádné náhodné číslo není závislé na jiném čísle v této posloupnosti, proto tato posloupnost má vlastnost nepředvídatelnosti.

1.2.1 Typy generátorů náhodných čísel

Z hlediska způsobu generování náhodných čísel je možné generátory rozdělit do třech kategorií

- nedeterministické,
- deterministické,
- hybridní.

Nedeterministické generátory náhodných čísel

Nedeterministické generátory náhodných čísel, nazývané také jako skutečné nebo pravé generátory se označují zkratkou TRNG. Jako zdroj entropie využívají elementární fyzikální principy které mají náhodný charakter. Nedeterministické generátory se vyznačují narůstáním entropie v čase, ale také pomalejším generováním čísel.

Deterministické generátory náhodných čísel

Deterministické generátory náhodných čísel, nebo také aritmetické generátory se správně označují jako generátory pseudonáhodných čísel. Pro generování výstupní posloupnosti používají algoritmus, který je možno popsat pomocí deterministické funkce a vnitřního stavu generátoru. Protože jde o deterministický algoritmus není výstupní sekvence skutečně náhodná a protože počet vnitřních stavů je konečný je také periodická[10]. Nejčastěji používané jsou lineární kongurentní generátory, posuvné registry s lineární zpětnou vazbou, generátory Blum Blum Shub [3], Micali-Schnorr [16], generátory využívající hashovací funkce, blokové a proudové šifry i jiné.

Hybridní generátory náhodných čísel

Hybridní generátory náhodných čísel jsou zajímavým mixem dvou předchozích typů. Nedeterministický generátor je použit pro určení počátečního stavu a dále je výstup generován deterministickým algoritmem. Mohou být zajímavou variantou například tam, kde nemůžeme uložit vnitřní stav deterministického generátoru a po každém zapnutí zařízení bychom museli použít stejné inicializační hodnoty a zároveň potřebujeme vyšší rychlost generování hodnot.

1.2.2 Bezpečnostní požadavky na generátor

Existuje několik standardů které se týkají bezpečnostních požadavků kladených na generátor náhodných čísel[6][8][1]. Většina těchto požadavků se však týká certifikace konečných produktů a vestavěných systémů. My si uvedeme základní kategorie ve kterých panuje obecná shoda a na které by měl být brán zřetel již při návrhu prototypu.

Zpětná bezpečnost

Jistota že předchozí vygenerované hodnoty nemohou být určeny ze současných nebo budoucích hodnot.

Rozšířená zpětná bezpečnost

Rozšíření předchozího stupně o požadavek, že předchozí vygenerované hodnoty není možno určit ani ze současného vnitřního stavu generátoru.

Dopředná bezpečnost

Jistota, že následující hodnoty nemohou být odvozeny ze současných ani minulých vygenerovaných hodnot.

Rozšířená dopředná bezpečnost

Rozšíření předchozího stupně o požadavek, že následující generované hodnoty nemohou být odvozeny ani z vnitřního stavu generátoru.

Kapitola 2

Zdroje entropie

Zdroj entropie je základní součástí generátoru náhodných čísel. Od jeho kvality se odvíjí kvalita celého generátoru. Jako zdroje entropie používáme fyzikální jevy, které dle současných znalostí fyziky považujeme za náhodné a jejichž vlastnosti můžeme pospat pouze statisticky. Dále se snažíme vybírat takové zdroje entropie aby případný útok na jejich vlastnosti, například podchlazením generátoru byl co nejtěžší a v ideálním případě nemožný.
V této kapitole si popíšeme několik fyzikálních jevů které se dají použít a také často používají právě v generátorech náhodných čísel.

2.1 Šum

Šum je v běžném slova smyslu zvukové nebo hlukové znečištění. V elektronice se používá pro označení nežádoucího rušení které zkresluje užitečný signál. Jeho podstata je z větší části kvantová a proto může být použitý jako dobrý zdroj entropie pro generátor náhodných čísel.

2.1.1 Tepelný šum

Tepelný šum(Johnson-Nyquistuv, Johnsonuv, nebo Nyquistuv šum) je šum generovaný náhodným pohybem elektronů uvnitř vodiče při libovolné teplotě vyšší než 0 K. Je nezávislý na velikosti protékajícího proudu, či na velikosti napětí. Kořen středního kvadrátu šumového napětí na rezistoru je dán vztahem :

$$u_n^2 = 4k_bTR\Delta f \tag{2.1}$$

kde:

$k_b = 1,38 * 10^{-23}$ [J/K] je Boltzmannova konstanta,

T je absolutní teplota v Kelvinech,

R [Ω]je odpor rezistoru,

Δf [Hz] je šířka pásma.

Jak vidíme šumové napětí není příliš vysoké a pro použití ke generování náhodných čísel musíme použít velké zesílení a tím se nám takový generátor stává citlivější k možným vlivům okolí. Také závislost šumového napětí na teplotě není při konstrukci žádoucí.

2.1.2 Výstřelový šum

Výstřelový šum byl poprvé popsán Waltrem Shottkym který studoval proud procházející vakuovou elektronkou [19]. Obecně se vyskytuje všude kde nosiče náboje vznikají a rekombinují nespojitě. Má charakteristiku bílého šumu. Kořen středního kvadrátu výstřelového šumového proudu je dán vztahem :

$$\delta_i = \sqrt{2qI\Delta f} \tag{2.2}$$

kde:

q = $1,602 * 10^{-19}$ [C] je náboj elektronu,

I [A] je stejnosměrný proud procházející přechodem,

Δf [Hz] je šířka pásma.

2.1.3 Generačně-rekombinační šum

Generačně-rekombinační šum je vyvolán náhodnými změnami generační a rekombinační rychlosti nosičů náboje. Vyskytuje se pouze u polovodičových detektorů. Změny počtu elektronů můžeme znamenávat jako signál.

2.1.4 Blikavý šum

Blikavý (plápolavý, růžový) šum se vytváří v důsledku poruch krystalové mřížky a nečistot v polovodiči. Projevuje se především na nižších kmitočtech. Jeho spektrální hustota výkonu klesá směrem k vyšším kmitočtům a to s kmitočtovou závislostí 1/f (proto se někdy také nazývá 1/f šum).

2.2 Radiový šum

Další možností je přijímat radiové signály. Přestože většina rádiových signálů vysílaných lidmi není ani zdaleka náhodná, celkové radiové pozadí je také tvořeno signály přilétajícími z vesmíru. Jejich zdroje je těžko možné považovat za náhodné, ale díky jejich množství je možné je považovat za spolehlivý zdroj náhody.

2.3 Radioaktivní rozpad

Velmi dobrým zdrojem náhody je samovolný rozpad atomových jader v přirozeně radioaktivních prvcích.
Rozpady jader atomu známe dva:

α rozpad

β rozpad

Kdy α rozpad je přeměna izotopů těžkého prvku na lehčí doprovázená emisí α částice a uvolněním energie odpovídající hmotnostnímu úbytku systému.
Obecný zápis přeměny je:

$$^{A}_{Z}X \rightarrow ^{A-4}_{Z-x}Y + ^{4}_{2}He + \Delta E \tag{2.3}$$

X a Y jsou jádra izotopu před a po přeměně

ΔE je energie hmotnostního úbytku systému která je vyzářena při následné deexcitaci ve formě γ záření

β rozpad je radioaktivní přeměna při které se nemění celkový počet nukleonů v jádře. Dle typu rozpadu (β^+, β^-), dochází k vyzáření elektronu nebo pozitronu a k němu příslušejícího elektronového (anti)neutrina.
obecný zápis této přeměny je:

β^- :

$$ {}^A_Z N \rightarrow {}^A_{Z+1} N' + e^- + \overline{v}_e \tag{2.4} $$

β^+ :

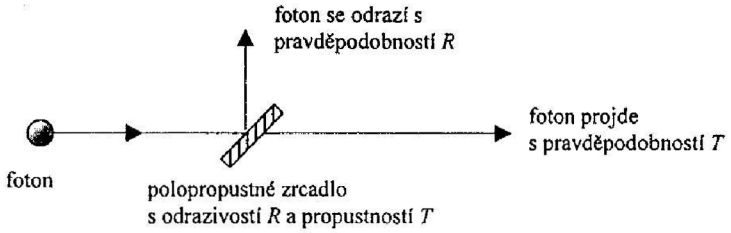
$$ {}^A_Z N \rightarrow {}^A_{Z-1} N' + e^+ + v_e \tag{2.5} $$

Dle klasické fyziky by k rozpadu nikdy nemělo dojít. Rozpad je důsledkem kvantového tunelovacího jevu a jeho rychlost je dána výškou potenciálové bariéry kterou musí částice překonat. Proto podle současných znalostí můžeme říci že přesný čas kdy dojde k rozpadu jádra je nepředvídatelný.
Celkový počet rozpadů bude v čase pochopitelně klesat s tím jak bude ubývat jader která se rozpadnout mohou.

Detekovat můžeme buď primární emitovanou částici, nebo u většiny rozpadů následnou emisi γ částice způsobenou deexcitací atomového jádra. K detekci je pochopitelně potřeba vhodný detektor a to takový který je schopen detekovat jednotlivé částice. Takových detektorů již bylo zkonstruováno více, nejstarším z nich je tzv. Geiger-Mullerova trubice, dále je možno použít scintilační detektory nebo modernější polovodičové detektory, které jsou ale stále velmi drahé. Navíc práce s radioaktivním materiálem je nebezpečná a podléhající přísné kontrole ze strany státních úřadů. Přestože je možno postavit generátor s zářičem s natolik nízkou aktivitou, že jeho držení je legální i bez zvláštního povolení jsou tyto zdroje entropie spíše k vidění na specializovaných pracovištích.

2.4 Dělení světla na děliči svazku

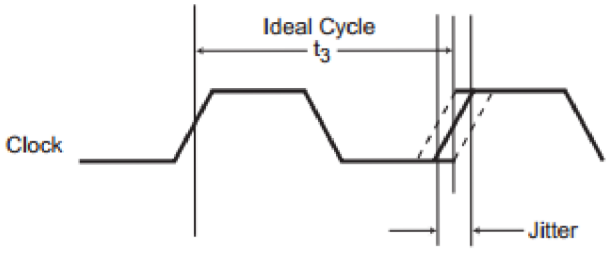
Jedním z elementárních kvantových procesů je dopad světelného kvanta - fotonu na tzv. dělič svazku. Jedná se o zařízení, které se v klasické optice používá k rozdělení jednoho svazku světla na svazky dva. Může jít např. o tzv. polopropustné zrcadlo, existuje však i řada jiných konkrétních realizací tohoto prvku. Zajímavé pro nás je, že snižujeme-li intenzitu světla, začne se projevovat jeho kvantový charakter. Světelná energie se totiž šíří v malých nedělitelných dávkách - fotonech. Dopadne-li jediný foton na dělič svazku, nemůže se rozpůlit; může prostě jen zvolit jednu ze dvou možných cest. Foton ale není kulečníková koule, to, kterou cestou se vydá, je náhodný jev v nejryzejším smyslu [7].



Obrázek 2.1: Dělič svazku fotonů[7]

2.5 Jitter

Jitter je nežádoucí odchylka charakteristik periodického signálu od jejich teoretického průběhu. Ve většině aplikací je silně nežádoucí, ale stejně jako mnoho jiných nežádoucích jevů může být využit jako zdroj entropie. Velmi častým případem je vzorkování rychlého oscilátoru druhým, řádově pomalejším. Kdy při každém cyklu pomalejšího oscilátoru naměříme jiný počet cyklů rychlejšího a tyto rozdíly pak převedeme na binární hodnoty[24].



Obrázek 2.2: Jitter[21]

Kapitola 3

Korekce odchylky od normálního rozložení

Většina zdrojů náhody nemá ideální rozložení. My ale požadujeme rovnoměrné rozložení generovaných hodnot a je poroto nutné sekvenci náhodných čísel ještě dále zpracovat, abychom dosáhli požadovaného rozdělení. V této kapitole si popíšeme několik vybraných algoritmů používaných právě k tomuto účelu.

3.1 Von Neumannův korektor

Von Neumannuv korektor [17] je známá metoda korekce odchylky náhodné posloupnosti od normálního rozdělení. Jde o velmi jednoduchou metodu produkující vyvážený výstup. Korektor zpracovává proud nepřekrývajících se dvojic bitů a generuje výstup následujícím způsobem:

Pokud je vstup dvojice ”00”nebo ”11”je vstup vynechán,

Pokud je vstup dvojice ”01”nebo ”10”je na výstup generován pouze první bit.

Předpokládejme že vstupní posloupnost je zatížena chybou posunutí o velikosti e, to znamená že pro vstupní bit x platí:

$$P_{(x=1)} = \frac{1}{2} + e, P_{(x=0)} = \frac{1}{2} - e \tag{3.1}$$

Tak pro výstupní bit y platí

$$\begin{aligned} P_{(y=1)} &= \frac{P_{(”10”)}}{P_{(”10”nebo”01”)}} \\ &= \frac{(\frac{1}{2}+e)(\frac{1}{2}-e)}{(\frac{1}{2}-e)(\frac{1}{2}+e)+(\frac{1}{2}+e)(\frac{1}{2}-e)} \\ &= \frac{\frac{1}{4}-e^2}{2(\frac{1}{4}-e^2)} \\ &= \frac{1}{2} \end{aligned} \tag{3.2}$$

Takže výstup Von Neumannova korektoru je vyrovnaný.

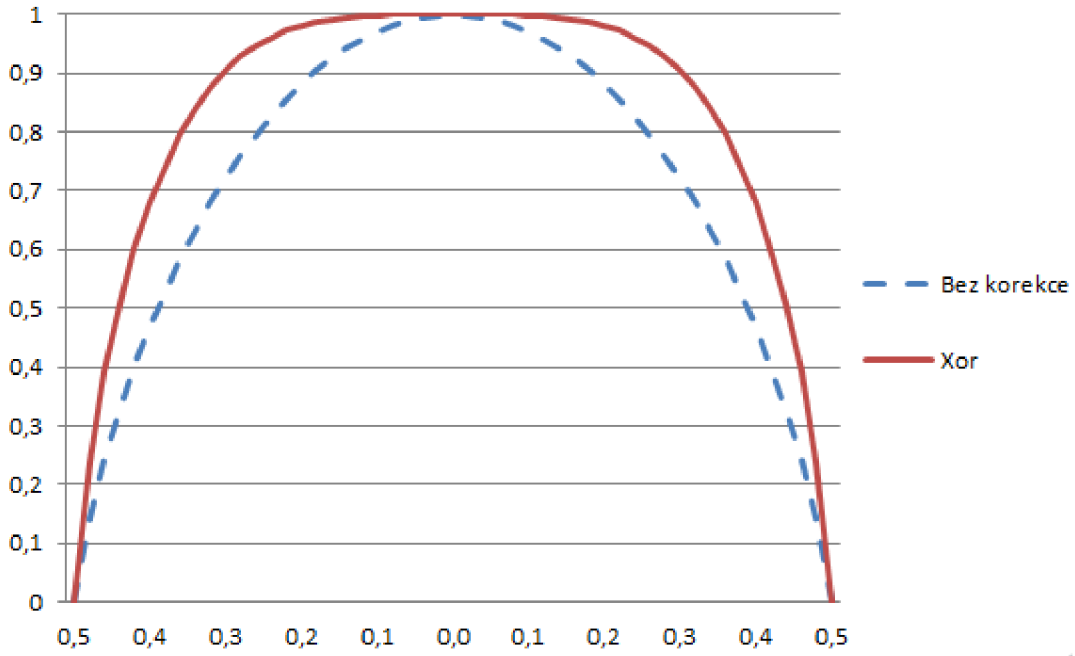
Na druhou stranu efektivita tohoto korektoru bude nízká. Pravděpodobnost dvojice "01"nebo "10"je $2(\frac{1}{2}+e)(\frac{1}{2}+e)$ což je $2(\frac{1}{4}-e^2)$ a navíc každý pár generuje výstup poloviční délky, tudíž poměr vstup/výstup Von Neumannova korektoru je dán vzorcem $\frac{1}{4}-e^2$. Maximální poměr bude tedy $\frac{1}{4}$. To znamená že vstupní proud bitů musí být mnohem vetší než požadovaný výstupní a zároveň se muže stát že delší dobu nebudeme schopni vygenerovat žádný výstupní bit (v případe že na vstupu máme dlouhou sekvenci bez změny).

3.2 XOR korektor

Další velmi jednoduchou metodou korekce je využití funkce XOR pro kompresi vstupu.

$$L(X,Y) = X \oplus Y \tag{3.3}$$

Jak je vidět tento korektor nám sníží rychlost generátoru na polovinu a odchylku sníží na $2e^2$. Jak si ukážeme dále není ani zdaleka optimální variantou, ale je možné ho použít pro srovnání o kolik jsou jiné algoritmy účinnější.



Obrázek 3.1: Závislost entropie na e

3.3 Hashovaci funkce

Hashovaci funkce je deterministický algoritmus který má na vstupu libovolný blok dat a produkuje řetězec pevné délky. Pokud je vstupní řetězec delší než výstupní dochází ke kompresi kterou můžeme využít pro opravu odchylky protože výstup hashovacího algoritmu má blízko k uniformnímu rozložení. Na druhou stranu je velmi těžké určit jaká vlastně odchylka výstupu bude. Přesto se v praxi často používají, nejspíše asi proto že implementace jednotlivých algoritmů jsou velmi rozšířené.

3.4 E Transformace

E transformace (a o něco složitější E' transformace) je metoda korekce navržená trojicí Smile Markovski, Danilo Gligoroski a Ljupco Kocarev v roce 2005[13]. Jde o bijekci z původního řetězce do nového pomoci operací na kvazigrupě.

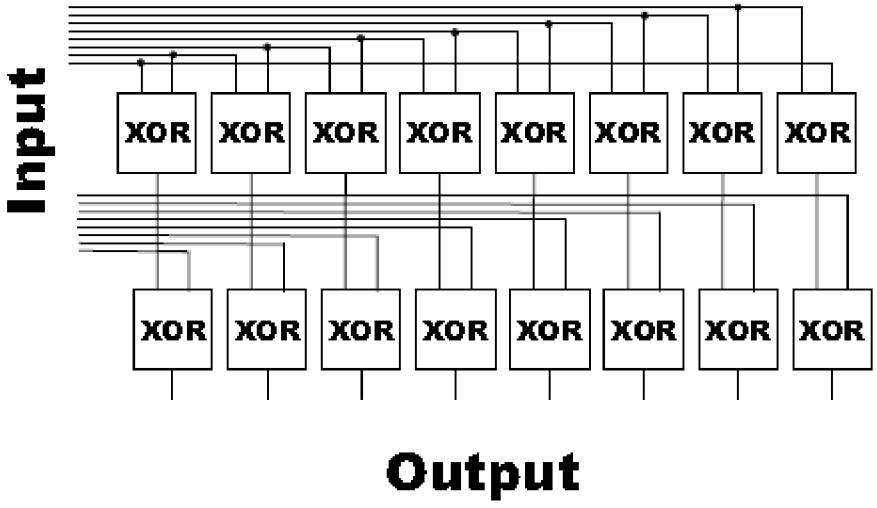
Kvazigrupa A (A, *) konečného uspořádání s je množina A kardinality s operací * na A takovou, že tabulka operací * je Latinský čtverec (tj. že všechny elementy z A se objevují v každém řádku a každém sloupci právě jednou). Mapování $e_{b_0,*}$ mapuje konečný řetězec elementů $a_1, a_2, .., a_n$ z A ko konečného řetězce $b_1, b_2, .., b_n$ tak, že $b_{i+1} = ai + 1 * b_i$ pro i = 0,1,..,n-1. b_0 je nazýván vůdčím prvkem mapování $e_{b_0,*}$. Vůdčí prvek musí být zvolen tak, že platí $b_0 * b_0 \neq b_0$.

Nicméně M. Dichtl ukázal že ne všechny závěry v původní práci jsou správné a že tato metoda korekce není úplně vhodná protože její výstup nemá uniformní rozložení a je dokonce možné odhadovat hodnoty výstupu.[4].

3.5 H korektor

Jde o techniku korekce navrženou Marcusem Dichtlem[4] kterou později generalizoval Patrick Lacharme[11].

Funkce převádí šestnáct vstupních bitů, z nichž každý má odchylku e na osm výstupních. Během procesu dojde u každého bitu k upravení odchylky až na hodnotu $2^4 * e^5$ při kompresi $\frac{1}{2}$ která je stejná jako například u XOR korektoru, který je ale schopen dosáhnout výstupní odchylky pouze $2e^2$
Funkci korektoru můžeme vidět na následujícím obrázku.



Obrázek 3.2: H korektor

Základní korektor lze také pseudokódem zapsat jako:

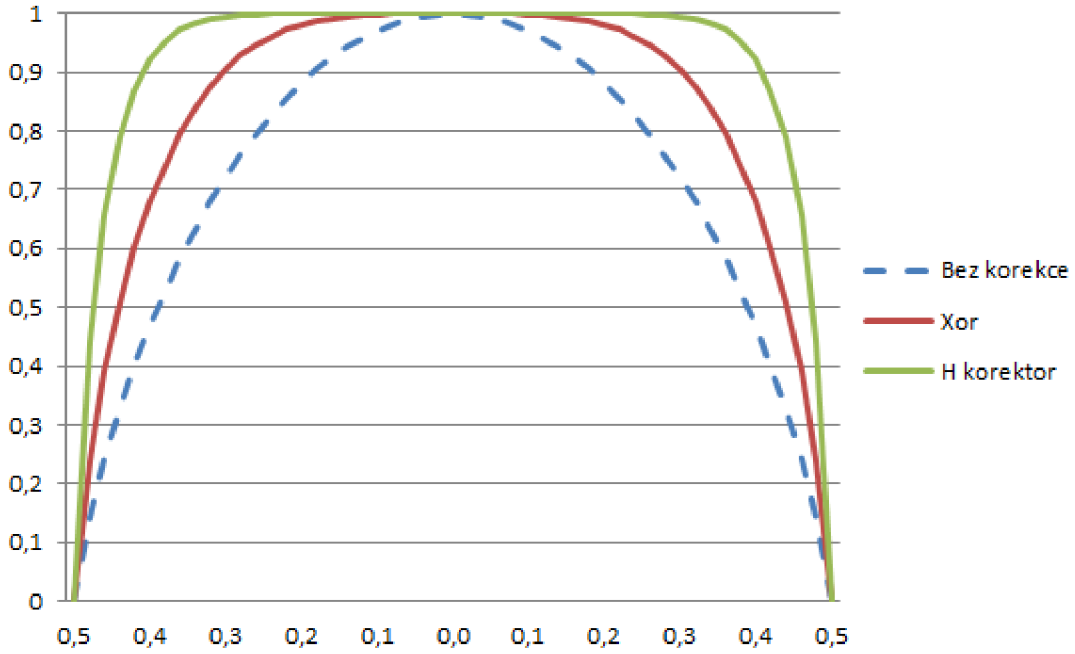
$$L(X, Y) = X \oplus (X \ll 1) \oplus Y \tag{3.4}$$

Odchylka od normálního rozložení po korekci bude $2e^3$

Korektor je možno dále rozšiřovat. Pro jednobitový výstup je nejlepší možnou variantou: $L : GF(2)^8 \times GF(2)^8 \to GF(2)^8$:

$$L(X, Y) = X \oplus (X \ll 1) \oplus (X \ll 2) \oplus (X \ll 4) \oplus Y \tag{3.5}$$

Pro lepší představu o jak velké zlepšení oproti XOR korektoru je na následujícím obrázku zobrazen také.



Obrázek 3.3: Závislost entropie na e

Kapitola 4

Testování náhodnosti

Našim úkolem je vytvořit generátor skutečně náhodných čísel. Ale jak poznáme že generovaná čísla jsou opravdu náhodná? Jistá kvantitativní měřítka pro hodnocení nám dává teorie statistiky. Statistických testů můžeme vytvořit libovolně mnoho, popíšeme si ale pouze ty z nich které se ukázaly být nejvíce vypovídající a které jsou vhodné pro výpočet na počítači. Musíme mít ale stále na paměti, že pokud se jistá posloupnost chová jako náhodná vzhledem k testům $T_1, T_2, ..., T_n$, nemužeme s jistotou říci, že neselže v testu T_{n+1}. Každý další test nám ale přesto zvýší důvěru v to že testovaná posloupnost je skutečně náhodná[10].

4.1 Statistické testování náhodnosti

Při testování náhodnosti postupujeme stejně jako při testování statistických hypotéz. Jako H_0, tzv. nulovou hypotézu, označujeme testovanou hypotézu. V našem případě hypotézu, že testovaná sekvence bitů je náhodná. Proti ní stojí H_A, která nulovou hypotézu popírá. Výsledkem každého testu musí být rozhodnutí na základě kterého hypotézu H_0 přijímáme, nebo odmítáme.

K získání tohoto výsledku využijeme tzv. testovou statistiku. Její obor hodnot rozdělíme na dva disjunktní obory - obor zamítnutí a obor přijetí. Hodnoty testové statistiky, které oddělují obor zamítnutí od oboru přijetí se nazývají kritické hodnoty a určují se z teoretického rozložení testové statistiky. Během testu se na základě testované sekvence vypočítá hodnota testové statistiky. Následně je porovnána s kritickou hodnotou a jestliže padne do kritického oboru, je H_0 zamítnuta, v opačném případě zamítnuta není.

Za předpokladu, že testovaná data jsou náhodná, by měla hodnota vypočtené testové statistiky přesáhnout kritickou hodnotu jen s velmi malou pravděpodobností (např. 0,001%). Pokud je kritická hodnota přece jen překročena, je předpoklad nahodilosti testovaných dat přinejmenším sporný.

Při testování statistických hypotéz se můžeme dopustit dvou druhů chyb. Buď zamítneme hypotézu, která je ve skutečnosti pravdivá (tzv. chyba I. druhu), nebo nezamítneme hypotézu která pravdivá není (chyba II. druhu). Pravděpodobnost, že se dopustíme chyby I. druhu označujeme jako hladinu významnosti. Bývá zvolena před začátkem testu. Běžné hodnoty se u statistických testů v kryptografii pohybují mezi 0,01 a 0,0001.

Skutečná situace	H0 se nezamítá	H0 se zamítá
H0 je pravdivá	správné rozhodnutí	chyba I. druhu
H0 je nepravdivá	chyba II. druhu	správné rozhodnutí

V ideálním případě požadujeme aby pravděpodobnost obou chyb byla co nejmenší. Při pevném rozsahu výběrového souboru však platí, že čím je pravděpodobnost chyby I. druhu menší, tím je pravděpodobnost chyby II. druhu větší a naopak. Celý testovací postup ja proto navržený tak, aby při předem dané hladině významnosti zajišťoval minimální pravděpodobnost výskytu chyby II. druhu. Tím je minimalizována pravděpodobnost nezamítnutí sekvence vygenerované špatným generátorem.

Při testování statistické hypotézy postupujeme takto:

1. Formulujeme nulovou a alternativní hypotézu.
2. Zvolíme hladinu významnosti.
3. Určíme kritickou hodnotu.
4. Vypočítáme hodnotu testované statistiky.
5. Pokud hodnota padne do kritického oboru tak H_0 zamítáme. V opačném případě H_0 přijímáme.

Hladina významnosti je v tomto postupu stanovena předem a závěrem je zamítnutí nebo přijetí nulové hypotézy. To nedovoluje uživatelům, kteří mají k dispozici pouze závěry o testované hypotéze, vybrat si svou vlastní hladinu významnosti a udělat tak své vlastní ohodnocení. Proto se zavádí P-hodnota hypotézy, což je nejmenší pozorovaná hladina významnosti, na níž nulová hypotéza muže být zamítnuta. Pokud je P-hodnota rovna 1, jeví se testovaná sekvence jako perfektně náhodná. Pokud je P-hodnota rovna 0, sekvence je naprosto nenáhodná. test hypotézy založený na použití P-hodnoty vypadá takto:

1. Formulujeme nulovou a alternativní hypotézu.
2. Zvolíme hladinu významnosti α.
3. Vypočítáme hodnotu testové statistiky.
4. Určíme (vypočítáme) P-hodnotu.
5. Pokud $P \leq \alpha$, tak H_0 zamítáme. V opačném případě H_0 přijímáme.

Například pokud $\alpha = 0,01$, můžeme očekávat, že 1 sekvence ze 100 bude zamítnuta. $P > 0,01$ znamená, že sekvence bude považována za náhodnou s jistotou $99P \leq 0,01$ znamená, že sekvence bude považována za nenáhodnou s jistotou 99

4.2 Používané testy

V této části si představíme statistické testy, které se používají při testování náhodných sekvencí. Každý test je zaměřen na jiný aspekt testované sekvence a sám o sobě nemá příliš velkou vypovídací hodnotu, pouze pokud sekvence úspěšně projde velkým počtem testů můžeme s větší dávkou jistoty tvrdit že je skutečně náhodná. Postup výpočtu ani implementační detaily většiny testů nebudou v této části rozebírány, jsou popsány v kapitole věnované testování realizovaných generátorů.

Chí kvadrát test

Chí-kvadrát test (χ^2 test) je nejznámějším statistickým testem a je základní metodou používanou v kombinaci s ostatními testy. Umožňuje nám určit míru shody mezi naměřeným a očekávaným výsledkem[10].

1. Obor všech možných hodnot náhodné veličiny se rozdělí na k nepřekrývajících se částí.
2. Pro každou část se stanoví pravděpodobnost p_i, že náhodná veličina nabude hodnoty z i-té části.
3. Provede se N pokusů a zjistí se, kolikrát z těchto pokusů nabyla náhodná veličina hodnoty z k-té části. Tyto četnosti se označí $X_1, X_2, ..., X_k$.
4. Porovnají se očekávané četnosti v jednotlivých částech (Np_i) se skutečnými četnostmi (X_i) pomocí vzorce:

$$\chi^2 = \sum_{i=1}^{k} \frac{(X_i - Np_i)^2}{Np_i} \tag{4.1}$$

dále rozvojem výrazu $(X_i - N_{p_i})^2 = X_i^2 - 2Np_iX_i + N^2p_i^2$ s využitím faktu, že $\sum X_i = n$ a $\sum p_i = 1$ dostaneme vzorec

$$\chi^2 = \frac{1}{n} \sum_{i=1}^{k} \left(\frac{X_i^2}{p_i} \right) - n \tag{4.2}$$

který nám výpočet χ^2 může často usnadnit.

Pokud má testovaná náhodná veličina předpokládané rozdělení, má náhodná veličina χ^2 přibližně rozdělení chí kvadrát. Jestliže bylo rozdělení dáno včetně všech parametrů, je počet stupňů volnosti k-1; jestliže byl některý parametr rozdělení neznámý, snižuje se počet stupňů volnosti za každý neznámý parametr (bylo jej nutno nejprve z dat odhadnout a pak teprve stanovit pravděpodobnosti p_i). Hodnotu veličiny χ^2 porovnáme s kritickou hodnotou příslušného rozdělení chí kvadrát na požadované hladině významnosti.

Monobit

Monobit test je variantou frekvenčního testu pro dvě možné vstupní hodnoty. Je zaměřen na počet jedniček a nul vyskytujících se v celé testované sekvenci. Jeho účelem je zjištění počtu výskytu jedniček a nul, který by mel být pro skutečně náhodnou sekvenci přibližně stejný.

Poker test

Obecný poker test uvažuje n skupin pěti po sobě jdoucích celých čísel pro $0 \leq j \leq n$ a sleduje, kterému z následujících sedmi vzorků odpovídá každá pětice (bez rozlišení pořadí).

Všechny různé: abcde

Jedna dvojice: aabcd

Dvě dvojice: aabbc

Trojice: aaabc

Full house: aaabb

Poker: aaaab

Pětice: aaaaa

Poté provedeme χ^2 test nad počtem pětic zařazených do každé z kategorií. Pro implementaci se lépe hodí jednodušší varianta při které počítáme počet různých hodnot v každé pětici. Toto rozdělení jde stanovat výrazně jednodušeji a výsledný test je skoro stejně dobrý. Ne vždy se ale počítá s pěticí. Obecně můžeme uvažovat n skupin k po sobě jdoucích čísel a zjišťovat počet k-tic s r různými hodnotami.

Test běhů

Který bývá také označován jako Golomb 2 se zaměřuje na zjištění celkového poctu běhů (tj. nepřerušených posloupností identických bitů) v testované sekvenci. Běh délky k je právě k bitu z obou stran ohraničených bity opačné hodnoty. Smyslem testu je určit zdali počet běhů různých délek odpovídá teoretickému rozdělení ve skutečně náhodné sekvenci. Jedná se tedy vlastně o určení, zdali není oscilace hodnot příliš rychlá či příliš pomalá.

Frekvenční test v blocích

Test zkoumá poměr jedniček v M-bitových blocích. Cílem testu je zjistit, zda četnost jedniček v blocích je přibližně M/2. Pokud bychom velikost bloku zvolili M=1, byl by tento test identický s Monobit testem.

Test hodnosti binární matice

Testujeme hodnost disjunktní podmatice celé posloupnosti. Účelem tohoto testu je posoudit lineární závislost mezi subsekvencí a celou testovanou sekvencí.

Test diskrétní Fourierovou transformací (Spektrální test)

Test je zaměřen na výšku spektrálních čar v diskrétní Fourierově transformaci. Cílem je odhalit periodické vlastnosti (tzn. opakující se vzory, které jsou blízko sebe), které by mohly ukazovat na odchylku od náhodnosti. Chyba je detekována, pokud číslo překročí hranici 95%.

Porovnávací test bez překrývání (Aperiodický test)

Testován je počet výskytů předdefinovaných cílových řetězců. Cílem testu je odhalit generátory, které vykazují příliš mnoho výskytů aperiodických vzorů. Používá se m-bitové okno, které hledá určité m-bitové vzory. Pokud není vzor nalezen, okénko se posune o jeden bit. Pokud je vzor nalezen, posune se na bit následující za nalezenou subsekvencí a hledání pokračuje.

Porovnávací test s překrýváním (Periodický test)

Tento test je velmi podobný předchozímu. Rozdílný je pouze algoritmus, pokud je nalezen předdefinovaný vzor. V tom případě se okénko posune pouze o jeden bit a hledání pokračuje.

Maurerův univerzální statistický test

Tento test zkoumá počet bitů mezi odpovídajícími vzory (poměr je vztažen k délce komprimované posloupnosti). Cílem je zjistit, zda je nebo není možné zkomprimovat posloupnost bez ztráty informace. Je-li možné sekvenci výrazně komprimovat, naznačuje to její nenáhodností[15].

Test lineární složitosti

Test je zaměřen na délku zpětnovazebního registru. Cílem je rozhodnout, zda je posloupnost dostatečně složitá, aby mohla být označena za náhodnou. Pro náhodnou sekvenci je charakteristický dlouhý zpětnovazební registr. Naopak krátký zpětnovazební registr naznačuje nenáhodnost.

Sériový test

Test zkoumá frekvenci vzájemného překrývání m-bitových vzorů v rámci testované posloupnosti. Cílem testu je posoudit, zda počet výskytů m-bitových překrývajících se vzorů je přibližně stejný jako u náhodné posloupnosti. Ta je rovnoměrná, tzn. každý m-bitový vzor má stejnou pravděpodobnost výskytu jako jakýkoliv jiný m-bitový vzor. Je-li m=1, je tento test identický s Monobit testem.

Test přibližné entropie

Podobně jako u Sériového testu zkoumá frekvenci vzájemného překrývání m-bitových vzorů. Cílem testu je porovnat tuto frekvenci překrývání bloků dvou za sebou jdoucích délek (m+n+1) s očekávaným výsledkem u náhodné posloupnosti.

Test částečného součtu

Tento test zkoumá maximální odchylku (od nuly) v průběhu náhodné procházky definované jako kumulativní součet hodnot -1 a +1 posloupnosti. Cílem je rozhodnout, zda kumulativní součet částečných posloupností v testované sekvenci je příliš velký nebo příliš malý v porovnání s očekávaným chováním kumulativního součtu náhodné posloupnosti. Tento kumulativní součet může být považován za náhodnou procházku. Ten je pro náhodnou posloupnost blízko nuly. U nenáhodné posloupnosti je procházka velmi vzdálená od nuly.

Test náhodné procházky

Test sleduje počet cyklů majících přesně K vstupů v kumulativním součtu náhodné procházky. Cyklem se rozumí část náhodné procházky začínající a končící nulovým stavem, mezilehlé stavy jsou nenulové. Testujeme, zda získaný výsledek odpovídá situaci, kdy vstupní posloupnost má náhodný charakter. Sledují se počty návratů do osmi nenulových stavů: -4, -3, -2, -1 a 1, 2, 3, 4. Celkem máme tedy vlastně osm různých testů.

4.3 Skupiny testů

4.3.1 FIPS 140-2

Federální standard zpracování informací 140 je sérií publikací, která popisuje standardy počítačové bezpečnosti úřadů USA. Součástí těchto standardů je soubor požadavků na kryptografické moduly pro použití úřady USA[6]. Ve své poslední revizi určuje pouze obecné bezpečnostní zásady a neobsahuje statistické testy. Pokud ale vyjdeme ze starší verze tohoto dokumentu můžeme si uvést tři statistické testy které obsahoval[5]. Nicméně testy jsou poměrně jednoduché a nemají příliš velkou vypovídací hodnotu.

- Monobit test
- Poker test
- Test Běhů.

4.3.2 DIEHARD

Baterie testů DIEHARD zveřejněná roku 1995 Georgem Marsagliem je souborem patnácti statistických testů převážně z oblasti transformace posloupnosti na jevy známé z reálného života[**?**]. Původním záměrem bylo testování generátorů náhodných čísel používaných v oblasti simulací, pro použití v oblasti kryptografie není nejvhodnější. V novější verzi testu byly přídány další dva testy.

- Narozeninový test,
- GCD test,
- Gorilla,
- Overlapping permutations,
- Hodnoty 31x31 a 32x32 matic,
- Hodnoty 6x8 matic,
- Opičí test na 20-bitových slovech,
- Opičí testy OPSO, OQSO, DNA,
- Počet jedniček v řetězci,
- Počet jedniček ve vybraných bytech,
- Test parkoviště,
- Test minimální vzdálenosti,
- Test náhodných koulí,
- Sqeeze test,
- Test překrývajících se součtů,
- Test běhů,
- Craps test.

4.3.3 NIST

NIST vydal v roce 1999 balík sestávající ze šestnácti statistických testů[18]. Tento balík je veřejně přístupný včetně zdrojových kódů. Rychle po vydání se stal hlavním rivalem balíku DIEHARD a ukázal se jako lepší. Vznikl jako reakce na nedostatek možností testování binárních posloupností se zaměřením na kryptografii. Dnes je v této oblasti prakticky považován sa standard. Samotná implementace obsahuje výběr nejlepších známých testů své doby. Postupem času bylo objeveno několik nedostatků v některých testech které však již byly odstraněny[9]. V poslední verzi tohoto balíku nejsou v současné době známy žádné nedostatky.

Seznam použitých testů:

- Frekvenční test,
- Frekvenční test v blocích,
- Test běhů,
- Test nejdelšího běhu v bloku,
- Test hodnosti binární matice,
- Test diskrétní Fourierovou transformací,
- Porovnávací test bez překrývání,
- Porovnávací test s překrýváním,
- Maurerův univerzální statistický test,
- Test lineární složitosti,
- Sériový test,
- Test přibližné entropie,
- Test částečného součtu,
- Test náhodné procházky,
- Variantní test náhodné procházky.

4.3.4 CRYPT-X

Crypt-X je komerční balík, který není volně dostupný. Jedná se o komplexní softwarový balík, který používá rychlé a efektivní testovací procedury pro proudové šifry, blokové šifry a generátory klíčů. Testy proudových šifer jsou využitelné i pro testování generátorů náhodných čísel.

Testy bitových posloupností:

- Frekvenční test,
- Binary Derivative test,
- Change Point test,

- Test běhů,
- Test složitosti posloupností,
- Test lineární složitosti.

Testy generátorů klíčů:

- Frekvenční test,
- Binary Derivative test,
- Sub-blocks test,
- Test entropie.

Testy blokových šifer:

- Frekvenční test,
- Binary Derivative,
- Sub-blocks,
- Avalanche Criteria test,
- Avalanche Variable test.

4.3.5 AIS 31

V rámci německého systému hodnocení a certifikace je od roku 2001 k dispozici dokument AIS31 - Třídy funkčnosti a metodika hodnocení pro fyzikální generátory náhodných čísel[8]. Tento dokument poskytuje jasná kriteria hodnocení a určuje dvě třídy funkčnosti, P1 pro méně citlivé aplikace a P2 pro kritické aplikace (např. generátory klíčů). Součástí dokumentu je rozbor pozitivních a negativních příkladů a nevylučuje žádné koncepty generátorů. Dává možnost žadateli o certifikaci nastavit alternativní kritéria hodnocení. Při hodnocení generátorů využívá tyto testy:

- Test nespojitosti,
- Monobit test,
- Poker test,
- Test běhů,
- Test nejdelšího běhu,
- Autokorelační test,
- Test uniformity rozložení,
- Porovnávací test pro multinomialní rozložení,
- Test entropie.

První čtyři zmíněné testy jsou bez úprav přebrané z dokumentu FIPS 140-1[5].

4.3.6 TestU01

Zatím nejkomplexnější baterií testů je knihovna TestU01. Jejími tvůrci jsou P. L'Ecuyer a R. Simard a pro nekomerční využití je zdarma[12]. Celkem obsahuje čtyři hlavní části:

- Implementace různých softwarových generátorů,
- Implementace statistických testů,
- Implementace předdefinovaných baterií testů,
- Implementace nástrojů na testovaní celých rodin generátorů.

Baterie testů jsou rozdělené na číselné a binární. Obsahují též vylepšenou baterii testů DIEHARD, celou baterii NIST, program ENT obsahující několik jednoduchých a spíše orientačních testů a knihovnu SPRNG[14], která obsahuje implementaci všech testů popisovaných D. Knuthem[10]. Přestože jde o nejkomplexnější baterii testů ze všech představených, práce s ní není právě jednoduchá. U většiny testů jsou uvedeny pouze základní informace a vzhledem k tomu, že správné nastavení jejich parametrů je naprosto klíčové pro získání relevantních výsledků, bude vhodné počkat než kvality této knihovny prověří čas.

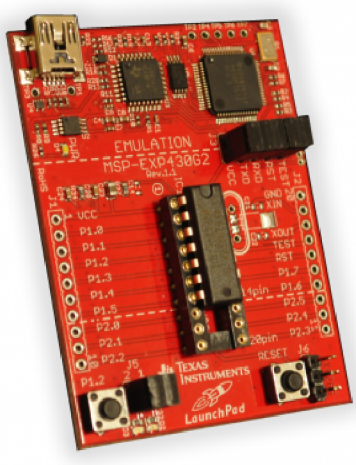
Kapitola 5

Návrh generátorů

Hlavní částí této práce je návrh a praktická realizace dvou různých hardwarových generátorů náhodných čísel a jejich testy. Jako zdroje entropie jsem zvolil v prvním případě jitter na vnitřním oscilátoru procesoru MSP430 G2553 a v druhém případě výstřelový šum na Zenerově diodě. Oba návrhy používají jako základ vývojový kit LanuchPad. V prvním případě bez dalších přídavných zařízení a v druhém případě s přidaným zdrojem entropie vlastní konstrukce.

5.1 Vývojový kit LanuchPad

Vývojový kit LaunchPad je snadno použitelný flash programátor a ladicí nástroj pro vývoj na mikrokontrolérech rodiny MSP430 Value Line. Obsahuje 14-/20-pin DIP cílový procesor v patici s integrovanou možností programování a ladění formou "in-system"prostřednictvím Spy Bi-Wire (2-vodičový JTAG) protokolu[23].

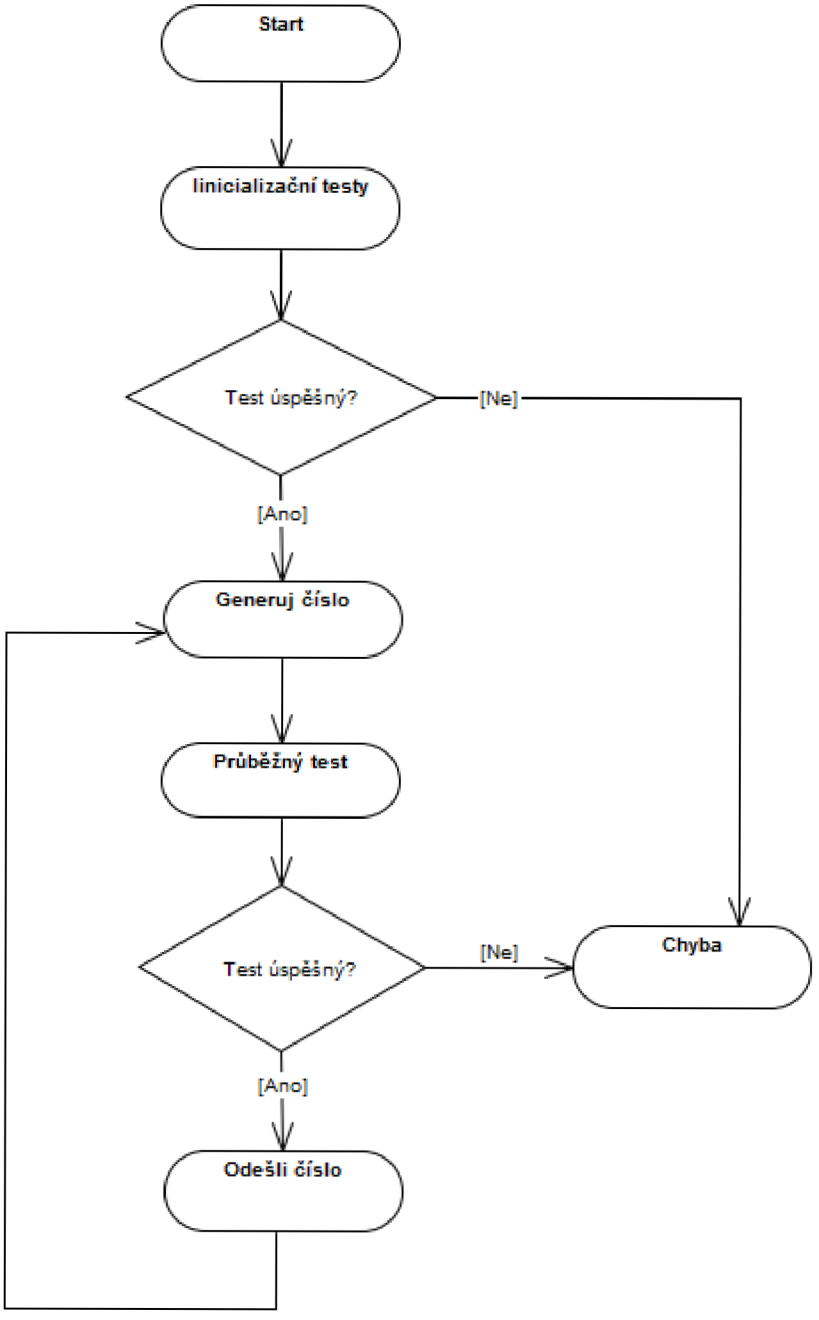


Obrázek 5.1: Vývojový kit LanuchPad[25]

5.2 Princip činnosti

Oba generátory mají podobný princip činnosti, jen každý z nich používá odlišný zdroj entropie. Na následujícím diagramu je znázorněn běh programu generátoru náhodných čísel.

Jak můžeme vidět generátor po zapnutí začne generovat čísla a provede na nich základní statistické testy k ověření správné funkčnosti zdroje entropie. Poté indikuje stav připravenosti a začne generovat čísla která jsou již odesílána na výstup. Před každým odesláním čísla je proveden průběžný test, který ma za úkol odhalit selhání zdroje entropie, nebo případný aktivní útok.



Obrázek 5.2: Stavový diagram generátoru

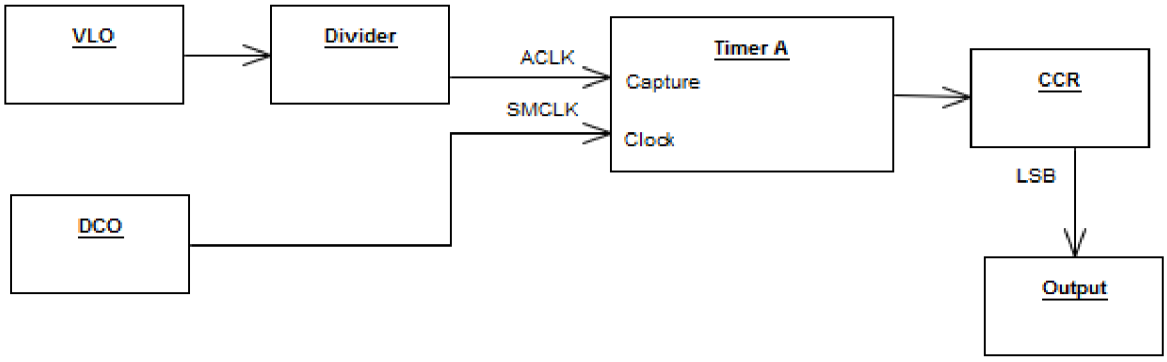
5.3 Návrh generátoru se dvěma oscilátory

Tento návrh generátoru vychází z ukázkového použití vývojového kitu LanuchPad[24]. Využívá toho, že procesor MSP430 má dva nezávislé oscilátory. Nízkofrekvenční oscilátor (VLO) a digitálně řízený oscilátor (DCO), z nich každý má svůj vlastní zdroj času. Rozdíl v časování těchto dvou oscilátorů je využit jako zdroj entropie pro generátor náhodných čísel

Pokud nastavíme oscilátory tak, že v každém cyklu nízkofrekvenčního oscilátoru je velký počet pulsů digitálně řízeného oscilátoru, tak počet těchto pulsů bude sice pokaždé téměř stejný, ale ne identický. Jako zdroj entropie použijeme informaci o tom, zda je počet pulsů sudý, nebo lichý.

V původnímu návrhu se pro zvýšení entropie po každém vygenerovaném bitu změnila hodnota periody oscilátoru, toto řešení není vhodné pokud, jako v našem případě, potřebujeme oscilátory i k jiné činnosti. Při prvních testech se ukázalo, že hodnoty se mění příliš pomalu a generátor má problém úspěšně splnit kritéria testu běhů. Proto jsem zvolil kombinaci dvou různých metod, kdy již při generování každého bitu určíme pět hodnot a vybíráme nadpoloviční většinou. Dále je použitý H korektor pro dosažení lepší shody s normálním rozložením.

Na následujícím obrázku vidíme blokové schéma funkce tohoto generátoru.



Obrázek 5.3: Blokové schema zdroje entropie

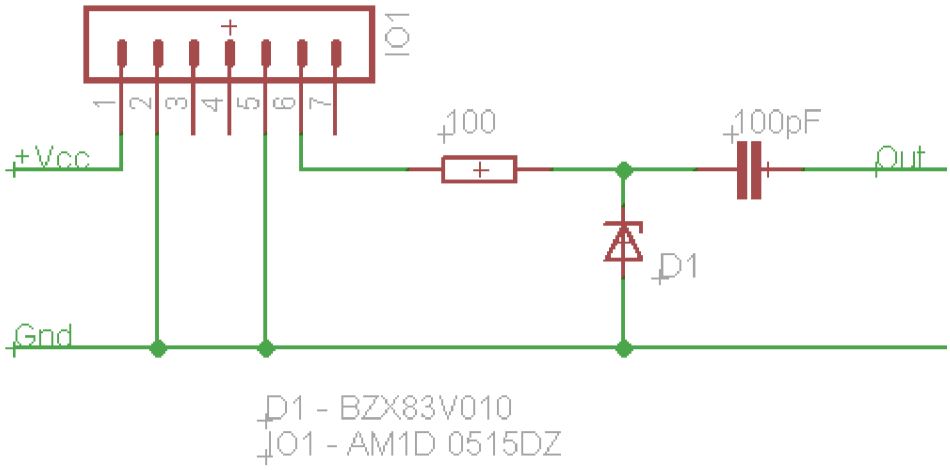
5.4 Návrh generátoru se Zenerovou diodou

Druhým realizovaným generátorem je generátor který používá jako zdroj náhody výstřelový šum na Zenerově diodě. Zenerovu diodu operující v módu závěrného průrazu poprvé navrhl jako zdroj šumu D. E. Sussans[22]. Jak je známo v tomto pracovním módu diody se uplatňují dva různé fyzikální mechanismy. Zenerův jev, který se projevuje hlavně při napětích menších než 6 V a při vyšších napětích tzv. lavinový efekt[20].

Oba tyto jevy vytváří specifický šum, který může být použitý jako zdroj entropie. Jako vhodnější se jeví využití výstřelového šumu produkovaného při funkci v módu lavinového efektu. Jeho hodnoty mohou dosahovat i několika desítek milivolt a jeho podstata je dobře popsána jako kvantová. Vzhledem k tomu že k lavinovému efektu dochází až při napětích vyšších než 6 V a pokud chceme dosáhnout vyšších hodnot šumového napětí, je třeba zvolit Zenerovu diodu se závěrným napětím ještě vyšším. Jako optimální se jeví napětí kolem 10 V. Pokud chceme celý generátor napájet z portu USB, jak je dnes standardem, musíme

použít napěťový násobič.

Pro realizaci jsem stejně jako v předchozím případě zvolil jako základ vývojový kit LanuchPad. Pro tuto realizaci využívám jeho vestavěného AD převodníku na který je připojen zdroj entropie.



Obrázek 5.4: Schema zapojení zdroje entropie

5.5 Inicializační testy

Po spuštění generátoru je proběhne několik jednoduchých testů pro ověření správné funkčnosti generátoru. Testy jsou převzaty z publikace FIPS 140-1 a jejich úspěšné absolvování je podmínkou pro další činnost generátoru[5]. Pokud některý z testů není úspěšný je tento stav signalizován rozsvícením červené diody a na výstup nejsou generována žádná čísla.

Monobit

1. Spočítej počet jedniček ve vzorku 20000 bitů a označ tuto hodnotu jako X.
2. Test je splněný pokud $9725 < X < 10275$.

Poker

1. (a) Rozděl vzorek 20000 bitů na 5000 souvislých čtyřbitových segmentů.
   (b) Spočítej a zapamatuj si počet výskytu všech možných 4-bitových hodnot.
   (c) Označ f(i) jako počet výskytů každé hodnoty i.
2. Vyčísli

$$x = \frac{16}{5000} * \left(\sum_{i=0}^{15} [f(i)]^2 \right) - 5000 \tag{5.1}$$

3. Test je splněný pokud $2,16 \leq X \leq 46,17$.

Run

1. "Run"je definovaný jako souvislá sekvence bitů sestávající pouze z jedniček nebo nul, která je částí testovaného 20000 bitového toku. Výskyty všech úseků "run"všech délek větších nebo rovných 1 se spočítají a uloží.

2. Test je splněný, pokud všechny počty úseků dané délky patří do intervalů daných následující tabulkou. Pro potřeby tohoto testu se všechny úseky s délkou vetší než 6 považují za úsek délky 6.

Délka úseku	Požadovaný interval
1	2315-2685
2	1114-1386
3	527-723
4	240-384
5	103-209
6+	103-209

5.6 Průběžný test

Generátor obsahuje také průběžný test výstupních hodnot. Tento test vychází z doporučení AIS 31[8].

Jedná se o test dobré shody na vzorku 128 bitů, rozdělených do čtyřbitových bloků. Test je neúspěšný pokud hodnota χ^2 překročí 65. V takovém případě generátor přejde do chybového stavu, což signalizuje rozsvícením červené diody a musí být manuálně resetován.

5.7 Sériová komunikace

Oba generátory používají k odeslání vygenerovaných čísel vestavěnou podporu sériového přenosu a emulaci sériové linky která je součástí vývojového kitu.

Kapitola 6

Statistické Testy realizovaných generátorů

Pro testování realizovaných generátorů jsem zvolil baterii testů NIST, protože je v současné době jedním z nejkomplexnějších a nejlépe otestovaných balíků testů. Testovací metodika a použité hraniční hodnoty vychází z doporučení k tomuto balíku[18]. Zaměření každého testu a jeho účel bylo popsáno v předcházejícím textu. V této kapitole si u každého testu si nastíníme jeho výpočet, spočítáme potřebné hodnoty pro testovací vzorky dat a rozhodneme zda vzorek testu vyhověl, nebo ne. Celkový vzorek dat je pro oba generátory shodně 8 000 000 bitů.

6.1 Frekvenční test

Vstup:

n - počet bitů vstupní sekvence,

ϵ_n - n-tý bit vstupní sekvence.

Vyčíslíme:

1. Spočti $S_n = X_1 + X_2 + X_3 + ... + X_n$, $X_n = 2\epsilon_n - 1$
2. Spočti $S_{obs} = \frac{|S_n|}{\sqrt{n}}$
3. Spočti $P = erfc(\frac{S_{obs}}{\sqrt{2}})$
   Kde funkce $erfc(z)$ je doplňková chybová funkce která se počítá následovně:
   $efrc(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{u^2} du$

Test je považován za úspěšný pokud $P \geq 0,01$
Splnění tohoto testu je nutnou podmínkou k tomu, aby vůbec mělo smysl pokračovat v dalších testech.

Výsledky generátorů:

Generátor se dvěma oscilátory

S_n	-344
S_{obs}	-0.000344
P	0.730846
Výsledek	OK

Generátor se Zenerovou diodou

S_n	-692
S_{obs}	-0.000692
P	0.488937
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.2 Frekvenční test v blocích

Vstup:

M - délka každého bloku,

n - délka vstupního řetězce,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Rozděl vstupní sekvenci do $N = \left[\frac{n}{M}\right]$ nepřekrývajících se bloků a zahoď nevyužité bity.
2. Urči poměr π_i jedniček v každém M-bitovém bloku
   $\pi_i = \frac{\sum_{j=1}^{M} \epsilon_{(i-1)M+j}}{M}$, pro $1 \leq i \leq N$
3. Spočti χ^2 statistiku $\chi^2_{(obs)} = 4M \sum_{i=1}^{N} (\pi_i - \frac{1}{2})^2$
4. Spočti $P = igamc(n/2, \chi^2_{(obs)}/2)$
   Kde funkce $igamc(a, x)$ neúplná gamma funkce
   $P(a,x) \equiv \frac{\gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt$
   $\Gamma(a) = \int_0^\infty t^{z-1} e^{-t} dt$
   Kde $P(a,0) = 0$ a $P(a,\infty) = 1$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

χ^2	7816.312500
P	0.484115
Výsledek	OK

Generátor se Zenerovou diodou

χ^2	7677.218750
P	0.859752
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.3 Test běhů

Vstup:

n - délka vstupního řetězce,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Spočítej poměr jedniček π ve vstupním řetězci $\pi = \frac{\sum_j \epsilon_j}{n}$
2. Urči zda je $|\pi - \frac{1}{2} \geq \tau|$ kde $\tau = \frac{2}{\sqrt{n}}$ je hodnota předdefinovaná v kódu.
   Pokud není, prohlaš test za neúspěšný
3. Spočti $V_{n(obs)} = \sum_{k=1}^{n-1} r(k) + 1$, kde $r(k)$=0 pokud $\epsilon_k = \epsilon_{k+1}$, jinak $r(k) = 0$
4. Spočti $P = efrc\left(\frac{|V_{n(obs)} - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

π	0.499828
$V_{n(obs)}$	499633
P	0.463021
Výsledek	OK

Generátor se Zenerovou diodou

π	0.499654
$V_{n(obs)}$	500244
P	0.6252111
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.4 Test nejdelšího běhu v bloku

Vstup:

n - délka vstupního řetězce,

ϵ - vstupní řetězec,

M - Délka každého bloku, kód je přednastaven tak že zvolí hodnotu M dle následující tabulky

Minimální N	M
128	8
6272	128
750000	10^4

N - Počet bloků.

Postup výpočtu:

1. Rozděl vstupní sekvenci do M-bitových bloků
2. Rozděl počty v_i nejdelších běhů v každém bloku do kategorií dle následující tabulky

v_i	M = 8	M=128	$M = 10^4$
v_0	≤ 1	≤ 4	≤ 10
v_1	2	5	11
v_2	3	6	12
v_3	≥ 4	7	13
v_4		8	14
v_5		≥ 9	15
v_6			≥ 16

3. Spočti $\chi^2_{(obs)} = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$
   Hodnoty π_i jsou součástí algoritmu a je možno je dohledat v dokumentaci[18]
   Hodnoty K a N jsou odvozeny z následující tabulky

M	K	N
8	3	16
128	5	49
10^4	6	75

4. Spočti $P = igamc\left(\frac{K}{2}, \frac{\chi_{(obs)^2}}{2}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

χ^2	3.054146
P	0.802021
Výsledek	OK

Generátor se Zenerovou diodou

χ^2	4.597857
P	0.596323
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.5 Test hodnosti binární matice

Vstup:

n - délka vstupního řetězce,

ϵ - vstupní řetězec,

M - Počet řádků matice, použito 32,

Q - počet sloupců matice, použito 32.

Postup výpočtu:

1. Rozděl vstupní sekvenci na nepřekrývající-se M*Q-bitové bloky; existuje $N = \left[\frac{n}{MQ}\right]$ takových bloků. Zbytek vstupu nebude použit. Vytvoř matice tak že každý řádek obsahuje bity vstupního řetězce které po sobě přímo následují.
2. Spočti binární hodnost matice R_l
3. Urči
   F_M jako počet matic kde $R_l = M$
   F_{M-1} jako počet matic kde $r_l = M - 1$
   $N - F_M - F_{M-1}$ bude počet zbývajících matic.
4. Spočti $\chi^2_{obs} = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}$
5. Spočti $P = igamc\left(1, \frac{\chi^2_{(obs)}}{2}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

P_{32}	0.288788
P_{31}	0.577576
P_{30}	0.133636
F_{32}	293
F_{31}	560
F_{30}	123
Počet matic	976
χ^2	0.888072
P	0.641442
Výsledek	OK

Generátor se Zenerovou diodou

P_{32}	0.288788
P_{31}	0.577576
P_{30}	0.133636
F_{32}	298
F_{31}	558
F_{30}	120
Počet matic	976
χ^2	1.8162862
P	0.403272
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.6 Test diskrétní Fourierovou transformací (Spektrální test)

Vstup:

n - délka vstupního řetězce,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Vytvoř sekvenci $X = x_1 + x_2 + x_3 + ... + x_n$, $x_n = 2\epsilon_n - 1$
2. Aplikuj diskrétní Fourierovu transformaci $S = DFT(X)$
3. Spočti $M = modulus(S') \equiv |S'|$, kde S' je podřetězec sestávající z prvních n/2 elementů v S a modulus funkce produkuje sekvenci výšek špiček.
4. Spočti $T = \sqrt{(log\frac{1}{0.05})\, n}$, V náhodném řetězci by 95% hodnot nemělo překročit T
5. Spočti $N_0 = 0.95n/2$. N_0 je teoretický počet špiček které jsou menší než T.
6. Urči N_1 jako skutečný počet špiček které jsou menší než T.
7. Spočti $d = \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/2}}$
8. Spočti $P = efrc\left(\frac{|d|}{\sqrt{2}}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

Percentile	95.004200
N_l	475021
N_o	475000
d	0.192709
P	0.847187
Výsledek	OK

Generátor se Zenerovou diodou

Percentile	94.989000
N_l	474945
N_o	475000
d	-0.504715
P	0.613759
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.7 Porovnávací test bez překrývání (Aperiodický test)

Vstup:

m - Délka vzorku v bitech,

n - délka vstupního řetězce,

ϵ - vstupní řetězec,

B - m-bitová šablona,

M - Délka testovaného podřetězce. Nastaveno na 2^17,

N - počet nezávislých bloků. Nastaveno na 8.

Postup výpočtu:

1. Rozděl vstupní sekvenci na N nezávislých bloků délky M.
2. Spočti W_j jako počet výskytů vzorku B v bloku j. Tak, že pokud testované bity neodpovídají šabloně posuň se o jeden bit, ale poku šabloně odpovídají posuň se o m.
3. Spočti teoretické hodnoty průměru a odchylky.
   $\mu = (M - m + 1)/2^m \; \sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$
4. Spočti $\chi^2_{(obs)} = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}$
5. Spočti $P = igamc\left(\frac{N}{1}, \frac{\sigma^2_{(obs)}}{2}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

Šablona	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	χ^2	P	Výsledek
00000001	228	225	239	272	247	246	232	235	7.080006	0.528026	OK
00000011	210	255	234	238	248	248	229	236	7.404112	0.493724	OK
00000101	242	242	239	268	221	259	223	233	8.182600	0.415840	OK
00000111	244	256	232	227	247	230	249	245	3.447061	0.903259	OK
00001001	249	256	272	242	233	229	256	234	6.534535	0.587580	OK
00001011	237	268	238	272	246	239	226	247	7.633951	0.470018	OK
00001101	219	260	217	258	231	229	252	245	9.640015	0.291206	OK
00001111	251	239	234	216	225	244	237	242	5.881029	0.660557	OK
00010001	248	261	207	255	237	265	262	254	11.438481	0.178070	OK
00010011	252	257	251	255	235	263	241	255	4.070911	0.850670	OK
00010101	237	226	245	206	236	225	239	240	9.780884	0.280743	OK
00010111	227	236	236	268	252	252	207	244	10.581615	0.226550	OK
00011001	243	258	234	250	241	250	254	251	2.202539	0.974164	OK
00011011	220	239	218	252	240	229	239	240	6.956084	0.541378	OK
00011101	225	236	270	251	234	236	249	220	8.146588	0.419284	OK
00011111	260	260	231	245	266	228	243	269	8.624273	0.374976	OK
00100011	240	267	218	260	222	269	257	253	11.979716	0.152111	OK
00100101	266	271	238	247	246	231	238	247	6.219962	0.622608	OK
00100111	267	249	256	269	239	239	264	237	7.647720	0.468616	OK
00101001	232	239	240	241	256	248	247	220	4.009479	0.856267	OK
00101011	261	224	255	225	225	223	248	236	8.756669	0.363237	OK
00101101	247	271	251	265	266	238	271	267	12.604625	0.126197	OK
00101111	249	225	257	229	253	240	205	230	11.058240	0.198421	OK
00110011	237	247	235	236	238	246	252	244	1.319193	0.995319	OK
00110101	239	234	241	239	239	232	243	269	4.059260	0.851738	OK
00110111	225	224	220	264	249	227	245	251	8.951556	0.346399	OK
00111001	226	237	230	235	246	232	235	231	4.525294	0.806897	OK
00111011	235	257	269	234	239	234	233	225	6.730481	0.565971	OK
00111101	250	248	229	221	236	266	233	244	6.276098	0.616335	OK
00111111	271	250	225	236	273	222	255	268	13.557876	0.094042	OK
01000011	238	250	250	243	256	226	253	228	3.881320	0.867669	OK
01000101	241	244	259	243	252	244	264	252	3.183328	0.922331	OK
01000111	256	267	222	248	236	255	274	262	10.867590	0.209315	OK
01001011	229	266	224	243	265	267	226	238	10.331651	0.242516	OK
01001101	259	237	254	255	273	266	230	265	10.317882	0.243420	OK
01001111	237	240	255	252	251	246	261	235	2.825330	0.944836	OK
01010011	251	215	246	255	255	232	253	229	6.736836	0.565274	OK
01010101	247	255	229	236	230	243	232	238	3.417405	0.905504	OK
01010111	239	235	240	228	254	226	224	254	5.571752	0.695078	OK
01011011	222	259	251	243	251	247	251	267	5.869378	0.661861	OK

01011101	227	264	253	242	249	259	240	265	6.225258	0.622016	OK
01011111	256	231	249	252	257	241	221	213	8.804331	0.359070	OK
01100101	250	251	237	237	232	253	236	245	2.016126	0.980513	OK
01100111	234	245	230	250	256	242	253	229	3.348559	0.910620	OK
01101011	229	221	239	251	251	244	239	264	5.531503	0.699547	OK
01101101	239	216	216	229	214	258	254	242	12.875772	0.116196	OK
01101111	227	217	244	261	258	221	269	257	11.971243	0.152492	OK
01110101	254	226	250	268	226	245	244	227	7.003746	0.536229	OK
01110111	236	240	247	249	232	226	228	225	5.153380	0.741064	OK
01111011	260	237	231	240	249	238	220	235	5.162913	0.740030	OK
01111101	247	253	236	246	238	249	271	269	6.604440	0.579848	OK
01111111	250	229	244	232	259	244	245	250	2.825330	0.944836	OK
10000011	252	256	244	224	239	258	252	246	3.780699	0.876348	OK
10000111	252	242	255	255	250	241	260	208	8.068210	0.426836	OK
10001011	237	245	270	231	260	241	244	245	4.897062	0.768520	OK
10001111	265	239	207	218	228	246	262	272	16.450463	0.036368	OK
10010011	219	213	260	248	259	251	267	252	11.527451	0.173566	OK
10010111	222	267	231	256	261	246	235	241	7.233586	0.511653	OK
10011011	267	222	246	233	249	238	243	244	5.095126	0.747362	OK
10011111	253	223	248	250	275	238	247	247	6.701883	0.569112	OK
10100011	261	228	245	232	262	213	206	284	21.286622	0.006424	FAILURE
10100111	275	216	257	261	245	246	226	230	11.553931	0.172243	OK
10101011	258	276	243	233	238	248	242	251	6.091803	0.636949	OK
10101111	254	249	253	215	264	221	216	248	11.795421	0.160568	OK
10110011	223	257	250	226	242	252	231	244	5.142789	0.742211	OK
10110111	262	264	263	245	231	243	232	262	7.251591	0.509747	OK
10111011	216	223	241	255	229	265	257	259	10.239504	0.248617	OK
10111111	244	254	270	271	277	247	247	242	10.977743	0.202963	OK
11000111	243	269	237	232	279	250	231	272	12.785743	0.119437	OK
11001111	250	256	215	255	255	249	251	270	8.477049	0.388309	OK
11010111	228	242	263	248	247	249	255	245	3.333730	0.911704	OK
11011111	232	213	231	250	253	229	249	254	7.419999	0.492068	OK
11101111	233	241	248	252	213	225	250	242	6.711416	0.568064	OK
11111111	255	230	273	282	250	240	247	237	11.424712	0.178776	OK
100000000	228	225	239	272	247	246	232	235	7.080006	0.528026	OK
100010000	265	222	234	267	221	241	248	241	8.983331	0.343704	OK
100100000	231	231	247	229	235	227	266	214	9.931286	0.269882	OK
100101000	257	256	223	235	231	226	241	250	5.852431	0.663758	OK
100110000	252	254	228	269	241	257	245	240	5.217990	0.734043	OK
100111000	285	239	242	233	267	251	248	231	10.943850	0.204901	OK
101000000	251	263	247	261	265	256	239	245	5.509261	0.702014	OK
101000100	266	240	234	257	226	235	276	243	9.290489	0.318386	OK
101001000	248	258	239	258	253	227	276	255	8.187896	0.415335	OK
101001100	240	237	224	270	261	263	244	228	8.657107	0.372042	OK
101010000	230	227	243	235	249	248	232	241	3.274417	0.915975	OK
101010100	246	216	244	215	225	223	220	227	14.108643	0.078977	OK
101011000	258	236	280	240	221	254	262	232	11.275370	0.186579	OK
101011100	253	252	220	242	248	254	241	239	3.710794	0.882229	OK

101100000	262	255	252	221	234	238	234	240	5.482781	0.704947	OK
101100100	243	240	260	236	237	270	249	234	5.011452	0.756351	OK
101101000	243	248	252	241	226	234	251	259	3.336908	0.911472	OK
101101100	218	264	251	235	228	239	234	265	8.611563	0.376115	OK
101110000	225	235	219	269	240	241	247	252	7.609590	0.472504	OK
101110100	242	228	227	237	274	232	237	250	7.343739	0.500040	OK
101111000	253	250	247	233	228	242	212	254	6.945492	0.542525	OK
101111100	221	220	234	226	230	253	235	234	8.523652	0.384057	OK
110000000	215	258	232	254	238	260	221	234	9.372045	0.311889	OK
110000010	255	257	247	249	248	225	254	234	3.799764	0.874722	OK
110000100	261	276	230	239	242	251	245	256	7.287603	0.505944	OK
110001000	254	254	244	258	240	245	234	241	2.193007	0.974516	OK
110001010	266	258	275	245	244	223	219	236	11.729752	0.163675	OK
110010000	232	237	231	230	255	225	266	235	6.843812	0.553574	OK
110010010	217	228	258	253	251	251	240	229	6.809919	0.557273	OK
110010100	253	254	211	267	235	220	249	258	11.347393	0.182782	OK
110011000	263	240	246	235	245	248	238	274	5.956229	0.652135	OK
110011010	268	245	262	266	241	266	231	224	10.313645	0.243699	OK
110100000	247	267	238	249	268	269	234	263	9.491731	0.302525	OK
110100010	236	245	247	229	235	250	256	263	3.892971	0.866649	OK
110100100	245	253	251	271	252	260	276	242	9.251300	0.321542	OK
110101000	247	207	245	253	243	236	238	251	6.855463	0.552305	OK
110101010	257	237	258	212	233	228	229	240	8.772556	0.361845	OK
110101100	248	235	260	264	219	250	231	258	7.523798	0.481311	OK
110110000	220	258	255	230	253	226	252	252	6.878764	0.549768	OK
110110010	251	260	251	243	235	238	234	245	2.422846	0.965232	OK
110110100	230	251	239	257	253	248	252	259	3.456594	0.902532	OK
110111000	231	231	241	242	220	233	263	245	6.022957	0.644660	OK
110111010	258	189	242	241	247	247	210	239	18.865365	0.015596	OK
110111100	246	235	247	249	214	239	250	246	4.620619	0.797248	OK
111000000	229	248	226	272	229	269	232	247	9.965179	0.267478	OK
111000010	249	252	235	229	268	250	246	255	4.762547	0.782628	OK
111000100	246	262	271	254	255	232	236	245	6.248560	0.619411	OK
111000110	236	230	231	204	256	238	251	228	10.734135	0.217223	OK
111001000	235	230	240	251	242	254	272	242	5.213753	0.734505	OK
111001010	237	243	211	241	235	252	249	239	5.738041	0.676548	OK
111001100	264	250	236	220	247	229	257	262	7.625478	0.470882	OK
111010000	246	293	215	265	263	241	224	248	18.905613	0.015373	OK
111010010	252	253	245	251	242	241	288	230	9.861381	0.274890	OK
111010100	253	222	261	260	266	237	240	254	7.409407	0.493172	OK
111010110	241	239	262	255	239	253	234	248	2.950312	0.937437	OK
111011000	234	253	271	257	256	228	273	231	10.491585	0.232203	OK
111011010	240	243	242	238	254	250	244	259	1.752393	0.987685	OK
111011100	234	231	239	237	218	251	248	240	4.718062	0.787241	OK
111100000	238	249	247	246	220	255	233	245	3.804001	0.874360	OK
111100010	270	247	280	219	246	234	232	245	12.073982	0.147933	OK
111100100	252	226	246	246	269	261	271	261	9.778766	0.280898	OK
111100110	254	243	234	236	246	235	259	273	5.969999	0.650593	OK

111101000	242	268	238	248	250	233	229	274	8.077742	0.425913	OK
111101010	255	246	245	233	261	246	251	267	4.682051	0.790955	OK
111101100	259	240	249	270	253	217	240	228	8.571314	0.379738	OK
111101110	254	231	240	244	233	230	236	247	2.899472	0.940507	OK
111110000	230	268	237	249	253	243	246	220	6.395784	0.602989	OK
111110010	249	216	262	246	257	265	281	234	13.564231	0.093855	OK
111110100	244	247	235	266	250	210	220	266	11.988189	0.151732	OK
111110110	249	249	266	252	227	232	249	240	4.529531	0.806471	OK
111111000	236	255	257	241	274	232	265	243	7.780116	0.455239	OK
111111010	242	242	234	258	272	249	227	270	8.759846	0.362958	OK
111111100	244	231	266	251	260	228	258	236	6.222081	0.622371	OK
111111110	255	230	273	282	250	240	247	237	11.424712	0.178776	OK

Generátor se Zenerovou diodou

Šablona	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	χ^2	P	Výsledek
000000001	243	255	244	201	250	239	239	269	11.375990	0.181291	OK
000000011	242	271	231	213	234	234	224	251	10.698123	0.219398	OK
000000101	240	245	252	227	269	241	256	242	4.859991	0.772430	OK
000000111	259	267	239	240	242	275	240	273	10.999986	0.201700	OK
000001001	232	250	232	235	239	212	224	259	8.881651	0.352378	OK
000001011	229	269	235	228	246	239	272	235	8.815982	0.358057	OK
000001101	255	261	232	234	217	236	265	212	12.380081	0.135034	OK
000001111	223	280	234	233	255	268	266	259	14.182785	0.077124	OK
000010001	242	225	236	266	255	256	241	214	8.860467	0.354203	OK
000010011	218	230	254	256	209	224	250	240	11.908752	0.155322	OK
000010101	261	231	256	253	228	256	247	256	5.198925	0.736119	OK
000010111	210	238	233	245	225	248	254	240	7.718684	0.461421	OK
000011001	249	259	244	231	226	265	228	260	7.175331	0.517839	OK
000011011	255	223	236	244	239	242	256	224	5.115250	0.745190	OK
000011101	260	228	235	245	225	239	250	233	4.856813	0.772765	OK
000011111	224	294	235	236	254	256	227	222	17.214123	0.027955	OK
000100011	265	225	245	263	251	236	243	240	5.465835	0.706822	OK
000100101	250	236	215	258	242	234	240	230	6.206193	0.624147	OK
000100111	240	216	241	239	227	230	229	222	8.706888	0.367623	OK
000101001	246	236	240	226	239	242	259	248	2.889939	0.941073	OK
000101011	248	231	228	261	249	285	233	251	11.005282	0.201400	OK
000101101	237	268	243	244	288	232	233	230	12.783624	0.119514	OK
000101111	214	236	250	240	218	266	251	222	11.535925	0.173142	OK
000110011	247	254	266	227	230	234	233	254	5.935046	0.654508	OK
000110101	251	244	228	236	230	238	250	211	7.380810	0.496158	OK
000110111	255	245	245	244	247	240	258	239	1.541619	0.991995	OK
000111001	232	240	273	245	255	237	253	248	5.344031	0.720252	OK
000111011	240	242	253	268	215	237	261	212	12.227561	0.141336	OK
000111101	230	258	238	255	257	265	234	252	5.566456	0.695667	OK
000111111	224	260	251	229	257	244	225	243	6.210430	0.623674	OK
001000011	245	242	253	237	219	270	245	249	6.186069	0.626398	OK

001000101	245	241	229	243	262	259	251	241	3.551919	0.895123	OK
001000111	243	219	233	239	252	235	242	247	3.985118	0.858464	OK
001001011	245	249	228	215	234	251	263	245	6.946551	0.542410	OK
001001101	242	257	214	250	240	252	231	251	5.977413	0.649762	OK
001001111	259	218	239	240	232	212	234	244	9.441950	0.306395	OK
001010011	237	227	250	251	234	237	217	267	7.787530	0.454495	OK
001010101	241	246	251	247	237	246	227	242	1.783109	0.986954	OK
001010111	264	221	251	265	248	261	235	246	7.623359	0.471098	OK
001011011	256	254	253	243	262	235	254	251	3.669486	0.885645	OK
001011101	231	244	246	257	253	228	230	249	3.828362	0.872268	OK
001011111	239	221	236	236	238	257	245	230	4.646039	0.794651	OK
001100101	257	231	266	265	231	220	256	234	9.533039	0.299341	OK
001100111	241	279	275	226	246	222	249	250	12.960505	0.113216	OK
001101011	267	212	242	236	239	246	259	252	8.214375	0.412815	OK
001101101	259	237	219	255	253	259	254	241	6.053673	0.641220	OK
001101111	247	272	259	263	263	259	264	244	9.894215	0.272529	OK
001110101	240	242	234	239	255	235	238	247	1.684606	0.989202	OK
001110111	235	218	232	260	229	245	245	223	7.801300	0.453117	OK
001111011	231	234	249	236	259	243	244	264	4.160940	0.842319	OK
001111101	245	248	222	246	247	241	229	214	7.046113	0.531666	OK
001111111	244	243	247	228	254	220	267	251	6.438151	0.598276	OK
010000011	222	260	244	240	232	241	259	227	6.057910	0.640745	OK
010000111	240	252	255	251	238	257	233	240	2.493810	0.962019	OK
010001011	235	232	250	244	269	248	232	254	4.843044	0.774213	OK
010001111	245	231	215	247	250	245	214	256	8.953674	0.346219	OK
010010011	273	241	211	261	246	241	232	235	10.460870	0.234156	OK
010010111	257	247	227	221	234	249	254	239	5.304841	0.724553	OK
010011011	241	248	247	240	230	232	253	246	2.028836	0.980115	OK
010011111	252	239	259	230	250	216	230	252	6.762256	0.562487	OK
010100011	253	252	202	240	231	261	229	264	12.765618	0.120172	OK
010100111	245	238	217	243	244	226	241	250	4.864228	0.771984	OK
010101011	221	274	246	259	248	261	232	234	9.326501	0.315506	OK
010101111	251	236	237	249	253	240	241	224	2.958785	0.936917	OK
010110011	254	258	232	246	242	240	235	234	2.744833	0.949336	OK
010110111	238	255	250	214	278	238	240	262	11.097429	0.196240	OK
010111011	244	236	242	250	236	229	228	259	3.733036	0.880371	OK
010111111	233	218	249	253	251	251	241	231	5.022044	0.755218	OK
011000111	227	235	257	220	270	256	246	262	9.565873	0.296827	OK
011001111	262	263	240	243	240	229	265	255	6.329056	0.610425	OK
011010111	247	223	246	269	239	232	251	266	7.523798	0.481311	OK
011011111	231	264	229	254	246	241	252	251	4.304987	0.828612	OK
011101111	264	226	214	250	214	238	217	236	14.457109	0.070602	OK
011111111	248	234	217	253	253	247	263	255	6.327997	0.610543	OK
100000000	243	255	244	201	250	239	239	269	11.375990	0.181291	OK
100010000	226	256	228	239	233	268	251	272	9.633660	0.291685	OK
100100000	208	254	203	254	210	247	241	229	19.499807	0.012404	OK
100101000	251	250	240	248	203	243	239	228	8.865763	0.353746	OK
100110000	247	262	259	287	230	262	269	247	14.969746	0.059736	OK

100111000	247	229	238	253	224	258	256	227	5.868319	0.661979	OK
101000000	250	221	252	244	217	243	251	258	6.813096	0.556926	OK
101000100	237	270	223	227	252	258	225	265	10.658934	0.221783	OK
101001000	254	251	226	261	219	231	261	239	7.933695	0.439974	OK
101001100	272	234	279	261	258	247	251	253	11.470256	0.176450	OK
101010000	244	257	240	223	249	235	252	220	5.847135	0.664350	OK
101010100	247	228	240	235	258	227	243	248	3.688551	0.884074	OK
101011000	235	244	254	242	255	273	248	225	6.431796	0.598982	OK
101011100	229	220	263	274	231	262	249	232	11.532747	0.173300	OK
101100000	241	269	232	224	277	251	252	250	10.189723	0.251962	OK
101100100	267	226	231	239	256	236	223	258	8.033257	0.430228	OK
101101000	262	256	251	251	240	245	238	238	2.744833	0.949336	OK
101101100	244	248	228	245	254	272	264	223	8.437859	0.391908	OK
101110000	232	255	244	241	245	228	236	239	2.661159	0.953792	OK
101110100	225	245	260	275	248	257	226	266	10.844288	0.210679	OK
101111000	236	265	247	232	275	248	260	219	10.628218	0.223667	OK
101111100	257	240	225	256	253	251	230	223	6.191365	0.625806	OK
110000000	237	285	258	243	266	229	240	257	11.885450	0.156388	OK
110000010	238	249	241	237	267	223	258	273	8.971680	0.344691	OK
110000100	226	234	227	249	229	257	247	217	7.993009	0.434153	OK
110001000	228	247	240	250	227	270	250	254	5.993300	0.647982	OK
110001010	236	246	228	232	235	279	227	256	9.364631	0.312476	OK
110010000	213	236	234	251	235	239	236	230	6.607618	0.579497	OK
110010010	259	234	274	221	228	235	252	236	9.415471	0.308468	OK
110010100	241	236	253	247	208	229	262	256	9.139028	0.330704	OK
110011000	241	274	242	258	243	272	268	262	11.723397	0.163978	OK
110011010	284	199	244	209	248	237	244	250	21.015475	0.007106	FAILURE
110100000	235	230	248	229	196	252	224	242	14.040856	0.080707	OK
110100010	241	250	252	239	292	254	212	260	16.125298	0.040621	OK
110100100	252	264	231	276	237	236	261	241	8.713243	0.367061	OK
110101000	237	260	234	218	265	245	232	238	7.239941	0.510980	OK
110101010	227	256	235	247	269	242	248	240	5.004038	0.757144	OK
110101100	242	233	241	221	231	241	248	250	3.831539	0.871994	OK
110110000	257	247	243	233	261	241	246	246	2.544650	0.959616	OK
110110010	247	234	226	235	267	256	245	233	5.555864	0.696843	OK
110110100	258	263	236	268	206	233	265	250	13.694508	0.090084	OK
110111000	224	239	243	215	230	218	231	256	10.490526	0.232270	OK
110111010	230	249	262	245	238	270	233	257	6.525002	0.588636	OK
110111100	248	234	248	227	260	253	246	216	6.571606	0.583476	OK
111000000	252	258	240	232	241	249	247	259	2.887821	0.941199	OK
111000010	250	231	215	238	247	264	241	229	7.347976	0.499596	OK
111000100	208	230	228	240	256	252	230	228	10.354953	0.240991	OK
111000110	263	259	269	247	227	206	245	253	12.840819	0.117445	OK
111001000	201	248	262	246	239	252	223	242	11.595238	0.170197	OK
111001010	233	223	252	261	249	244	243	266	6.017661	0.645253	OK
111001100	231	263	257	261	237	241	282	265	12.328182	0.137151	OK
111010000	225	258	256	226	211	239	242	273	12.666057	0.123868	OK
111010010	265	248	244	267	246	247	225	245	5.729568	0.677495	OK

111010100	270	263	240	236	246	225	244	228	7.363863	0.497931	OK
111010110	212	253	227	237	226	246	243	236	7.855317	0.447729	OK
111011000	258	258	246	244	243	221	236	253	4.530590	0.806365	OK
111011010	267	254	234	252	239	222	243	270	8.354185	0.399659	OK
111011100	207	225	236	232	237	227	238	249	10.008605	0.264422	OK
111100000	264	248	243	238	235	229	254	257	4.338880	0.825328	OK
111100010	227	230	217	225	273	279	206	245	21.601195	0.005711	FAILURE
111100100	266	244	228	235	255	237	242	251	4.417259	0.817653	OK
111100110	242	239	224	256	237	259	261	256	5.400166	0.714074	OK
111101000	229	254	251	221	244	221	243	279	11.272192	0.186748	OK
111101010	267	270	242	253	265	253	242	214	11.450132	0.177475	OK
111101100	255	245	221	234	259	248	240	229	5.246587	0.730925	OK
111101110	235	245	248	228	224	244	229	244	4.206484	0.838030	OK
111110000	237	249	237	251	263	240	248	238	2.535118	0.960073	OK
111110010	253	249	233	242	278	259	244	265	8.623214	0.375071	OK
111110100	236	247	244	220	245	237	233	274	7.304550	0.504159	OK
111110110	236	249	212	226	260	244	206	237	13.585414	0.093232	OK
111111000	221	220	241	246	247	230	223	230	8.403966	0.395036	OK
111111010	244	252	247	233	241	236	249	238	1.402868	0.994206	OK
111111100	238	233	251	241	235	245	255	266	3.809297	0.873906	OK
111111110	248	234	217	253	253	247	263	255	6.327997	0.610543	OK

Závěr
Vidíme že u každého generátoru se několik málo šablon opakuje častěji než by odpovídalo náhodnému rozložení. Vzhledem k velmi malému počtu těchto případů však není důvod sekvenci považovat za nenáhodnou. Pokud by však testovaná sekvence selhala i v některých dalších testech bylo by vhodné otestovat více sekvencí a ujistit se že jde o náhodnou fluktuaci a ne návrhovou chybu.

6.8 Porovnávací test s překrýváním (Periodický test)

Vstup:

m - Délka vzorku v bitech,

n - délka vstupního řetězce,

ϵ - vstupní řetězec,

B - m-bitová šablona,

K - Počet stupňů volnosti. Nastaveno na 5,

M - Délka testovaného podřetězce. Nastaveno na 1023,

N - počet nezávislých bloků. Nastaveno na 968.

Postup výpočtu:

1. Rozděl vstupní sekvenci na N nezávislých bloků délky M.

2. Spočti W_j jako počet výskytů vzorku B v bloku j a ulož počet výskytů B v každém bloku do pole v.

3. Spočti hodnoty λ a η, které budou použity pro výpočet teoretických pravděpodobností π_i odpovídajících tříd v_0. $\lambda = (M - m + 1)/2^m$ $\eta = \lambda/2$

4. Spočti $\chi^2_{(obs)} = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i}$, kde $\pi_0 = 0.367879$, $\pi_1 = 0.183940$, $\pi_2 = 0.137955$, $\pi_3 = 0.099634$, $\pi_4 = 0.069935$ a $\pi_5 = 0.140657$

5. Spočti $P = igamc\left(\frac{5}{2}, \frac{\chi^2_{(obs)}}{2}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

n	1000000
m	9
M	1032
N	968
λ	2.000000
η	1.000000

0	1	2	3	4	≥ 5	χ^2	P	Výsledek
334	191	142	96	61	144	3.966292	0.554279	OK

Generátor se Zenerovou diodou

n	1000000
m	9
M	1032
N	968
λ	2.000000
η	1.000000

0	1	2	3	4	≥ 5	χ^2	P	Výsledek
352	175	134	97	75	135	0.902042	0.970074	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.9 Maurerův univerzální statistický test

Vstup:

L - Délka bloku,

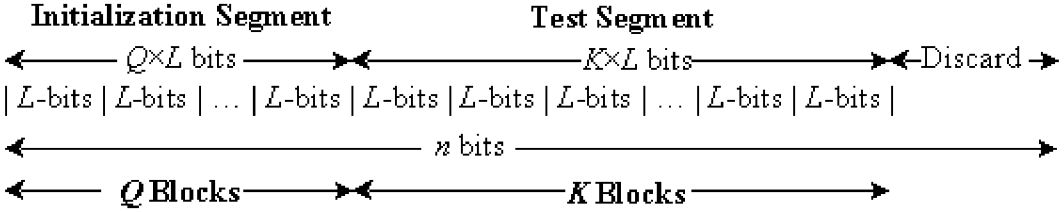
Q - Počet bloků,

m - Délka bitového řetězce,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Vstupní n-bitová sekvence je rozdělena na dvě části. Inicializační segment sestávající z Q L-bitových nepřekrývajících se bloků a testovací segment sestávající z K L-bitových nepřekrývajících se bloků.



Obrázek 6.1: Rozdělení sekvence

2. Pomocí inicializačního segmentu vytvoř tabulku pro každou možnou L-bitové hodnotu (tj. L-bit hodnota je použita jako index do tabulky). Číslo bloku posledního výskytu každého L-bit bloku je uvedeno v tabulce (tj. pro i od 1 do Q, TJ = I, kde j je desítková reprezentace obsah i-té L-bitového bloku)

3. Prozkoumej každý z K bloků v testovacím segmentu a urči počet bloků od posledního výskytu stejného L-bitového bloku (tj. i - Tj). Nahraď hodnoty v tabulce umístěním aktuálního bloku (tj. Tj = i). Přidejte vypočtenou vzdálenost mezi opakováními stejného L-bitového bloku akumulačnímu $\log_2$, součtu všech rozdílů zjištěných v blocích K (tj. sum = sum + log_2(i - T_j)).

4. Spočti $f_n = \frac{1}{k} \sum_{i=Q+1}^{Q+K} \log_2(i - T_j)$, kde T_j je záznam v tabulce odpovídající desítkové reprezentaci obsahu t-tého L=bitového bloku.

5. Spočti P = $efrc\left(|\frac{f_n - expectedValue(L)}{\sqrt{2}\sigma}|\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

L	7
Q	1280
K	141577
sum	877052.460246
σ	0.002768
variance	3.125000
exp value	6.196251
ϕ	6.194880
P	0.620399
Výsledek	OK

Generátor se Zenerovou diodou

L	7
Q	1280
K	141577
sum	876718.190708
σ	0.002768
variance	3.125000
exp value	6.196251
ϕ	6.192518
P	0.177616
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.10 Test lineární složitosti

M - Délka bloku v bitech,

n - Délka řetězce,

ϵ - vstupní řetězec,

K - Počet stupňů volnosti, nastaveno na 6.

Postup výpočtu:

1. Rozděl vstupní sekvenci do N nezávislých M-bitových bloků
2. Pomocí Berlekamp-Massey algoritmu[2] urči lineární složitost L_i každého z N bloků (i = 1, ..., N). L_i je délka nejkratší sekvence lineárních posuvných registrů, které generují všechny bity v bloku i. V každé L_i-bitové posloupnosti, některá kombinace bitů, po sečtení modulo 2, produkuje další bit v pořadí (bit L_i + 1).
3. Spočti teoretickou hodnotu μ
   $\mu = \frac{M}{2} + \frac{(9+(-1)^{M+1})}{36} - \frac{(M/3+2/9)}{2^M}$
4. Pro každý podřetězec spočti hodnotu T_i, kde $T|_i = (-1)^M * (L_i - \mu) = \frac{2}{9}$
5. Zaznamenej T_i hodnoty v_0, ..., V_6 do tabulky C takto:

$T_i \leq -2.5$	zvyš v_0 o jedna
$-2.5 < T_i \leq -1.5$	zvyš v_1 o jedna
$-1.5 < T_i \leq -0.5$	zvyš v_2 o jedna
$-0.5 < T_i \leq 0.5$	zvyš v_3 o jedna
$0.5 < T_i \leq 1.5$	zvyš v_4 o jedna
$1.5 < T_i \leq 2.5$	zvyš v_5 o jedna
$T_i > 2.5$	zvyš v_6 o jedna

6. Spočti $\chi^2_{(obs)} = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$,
   kde $\pi_0 = 0.010417$, $\pi_1 = 0.03125$, $\pi_2 = 0.125$, $\pi_3 = 0.5$, $\pi_4 = 0.25$, $\pi_5 = 0.0625$, $\pi_6 = 0.020833$[18].

7. Spočti $P = igamc\left(\frac{K}{2}, \frac{\chi^2_{(obs)}}{2}\right)$

Test je považován za úspěšný pokud $P \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

C_0	C_1	C_2	C_3	C_4	C_5	C_6	χ^2	P	Výsledek
28	61	263	997	499	111	41	4.681951	0.585203	OK

Generátor se Zenerovou diodou

C_0	C_1	C_2	C_3	C_4	C_5	C_6	χ^2	P	Výsledek
14	58	246	1021	482	144	35	7.73154	0.258438	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.11 Sériový test

M - Délka bloku v bitech,

n - Délka řetězce v bitech,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Vytvoř rozšířenou sekvenci ϵ': Rozšiř sekvenci přidáním prvních $m-1$ bitů na konec sekvence pro různé hodnoty n.

2. Urči frekvenci všech možných překrývajících se m-bitových bloků, všech možných překrývajících se (m-1)-bitových bloků a všech možných překrývajících se (m-2)-bit bloků.
Nechť $vi_j,..i_m$ udává frekvenci m-bitového vzoru $i_l...im$.
Nechť $vi_j,..i_{m-1}$ udává frekvenci m-bitového vzoru $i_l...im-1$.
Nechť $vi_j,..i_{m-2}$ udává frekvenci m-bitového vzoru $i_l...im-2$.

3. Spočti:
$\psi^2_m = \frac{2^m}{n} \sum_{i_j...i_m} \left(v_{i_j...i_m} - \frac{n}{2^m}\right)^2 = \frac{2^m}{n} \sum_{i_j...i_m} v^2_{i_j...i_m} - n$

$\psi^2_{m-1} = \frac{2^{m-1}}{n} \sum_{i_j...i_{m-1}} \left(v_{i_j...i_{m-1}} - \frac{n}{2^{m-1}}\right)^2 = \frac{2^{m-1}}{n} \sum_{i_j...i_{m-2}} v^2_{i_j...i_{m-1}} - n$

$\psi^2_{m-2} = \frac{2^{m-2}}{n} \sum_{i_j...i_{m-2}} \left(v_{i_j...i_{m-2}} - \frac{n}{2^{m-2}}\right)^2 = \frac{2^{m-2}}{n} \sum_{i_j...i_{m-2}} v^2_{i_j...i_{m-2}} - n$

4. Spočti $\nabla\psi^2_m = \psi^2_m - \psi^2_{m-1}$ a $\nabla^2\psi^2_m = \psi^2_m - 2\psi^2_{m-1} + \psi^2_{m-2}$

5. Spočti $P_1 = igamc(2^{m-2}, \nabla\psi^2_m)$ a $P_2 = igamc(2^{m-3}, \nabla^2\psi^2_m)$

Test je považován za úspěšný pokud $P_1 \geq 0,01$ a $P_2 \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

M	16
n	1000000
ψ_m^2	65906.864128
ψ_{m-1}^2	32947.236864
ψ_{m-2}^2	16582.963200
$\nabla\psi_m^2$	32959.627264
$\nabla^2\psi_m^2$	16595.353600
P_1	0.226720
P_2	0.121753
Výsledek	OK

Generátor se Zenerovou diodou

M	16
n	1000000
ψ_m^2	65675.128832
ψ_{m-1}^2	32783.790080
ψ_{m-2}^2	16279.498752
$\nabla\psi_m^2$	32891.338752
$\nabla^2\psi_m^2$	16387.047424
P_1	0.314267
P_2	0.491816
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.12 Test přibližné entropie

m - Délka bloku, m je délka prvního bloku v testu, m+1 je délka druhého.,

n - Délka celé bitové sekvence,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Vytvoř n překrývajících se m-bitových sekvencí připojením m-1 bitů z začátku sekvence na konec sekvence.
2. Urči počet výskytů možných m-bitových(m+1-bitových) hodnot v C_i^m, kde i je m-bitová hodnota.
3. Spočti $C_i^m = \frac{\#i}{n}$ pro každé i

4. Spočti $\varphi^{(m)} = \sum_{i=0}^{2^n-1} \pi_i \log \pi_i$, kde $\pi_i = C_j^3$ a $j = log_2 i$
5. Opakuj body 1-4 pr $m = m + 1$
6. Spočti $\chi^2 = 2n[log2 - ApEn(m)]$, kde $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$
7. Spočti $P = igamc\left(2^{m-1}, \frac{\chi^2}{2}\right)$

Test je považován za úspěšný pokud $P_1 \geq 0,01$ a $P_2 \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

m	10
n	1000000
$\varphi^{(m)}$	-6.930936
$\varphi^{(m)}$	-7.623571
χ^2	1024.138489
ApEn(m)	0.692635
P	0.492902
Výsledek	OK

Generátor se Zenerovou diodou

m	10
n	1000000
$\varphi^{(m)}$	-6.930969
$\varphi^{(m)}$	-7.623627
χ^2	979.191966
ApEn(m)	0.692658
P	0.838953
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.13 Test částečného součtu

n - Délka řetězce v bitech,

ϵ - vstupní řetězec,

Mode - Přepínač směru (0-dopředu, 1-zpětně).

Postup výpočtu:

1. Vytvoř normalizovanou sekvenci $X = x_1 + x_2 + x_3 + ... + x_n$, $x_i = 2\epsilon_i - 1$

2. Spočti částečné součty S_i:

Mode = 0	Mode = 1
$S_1 = X_1$	$S_1 = X_n$
$S_2 = X_1 = X_2$	$S_2 = XS_n + X_{n-1}$
...	...
$S_n = X_1 + X_2 + ... + X_n$	$S_n = X_n + X_{n-1} + ... + X_1$

3. Spočti $z = max_{1 \leq k \leq n}|S_k|$, kde $max_{1 \leq k \leq n}|S_k|$ je největší z absolutních hodnot částečných součtů

4. Spočti
$P = 1 - \sum_{k=(\frac{-n}{z}+1)/4}^{(\frac{n}{z}-1)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \sum_{k=(\frac{-n}{z}-3)/4}^{(\frac{n}{z}-1)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right]$
kde Φ je Standardní normální distribuční funkce
$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{u^2/2} du$

Test je považován za úspěšný pokud $P_1 \geq 0,01$ a $P_2 \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory
Mode 0:

z	981
P	0.646686
Výsledek	OK

Mode 1:

z	637
P	0.939119
Výsledek	OK

Generátor se Zenerovou diodou
Mode 0:

z	828
P	0.789427
Výsledek	OK

Mode 1:

z	946
P	0.679222
Výsledek	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

6.14 Test náhodné procházky

n - Délka řetězce v bitech,

ϵ - vstupní řetězec.

Postup výpočtu:

1. Vytvoř normalizovanou sekvenci $X = x_1 + x_2 + x_3 + ... + x_n$, $x_i = 2\epsilon_i - 1$
2. Spočti částečné součty S_i:
   $S_1 = X_1$
   $S_2 = X_1 = X_2$
   ...
   $S_n = X_1 + X_2 + ... + X_n$
3. Vytvoř novou sekvenci S' tak, že $S' = 0, S_1, S_2, ..., S_n, 0$
4. Nechť J je celkový počet průchodů nulou v S', kde průchod nulou je nulová hodnota v S' jiná než nula na začátku. J je i počet cyklů v S ', kde cyklus S"je podřetězec S' sestávající z nuly, následované nenulovými hodnotami a končící opět nulou. Koncová nula v jednom cyklu může být počáteční nula dalšího cyklu. počet cyklů v S 'je počet nulových přechodech.

   Je-li J ¡500, přeruš test(podmínka vycházející z vlastností χ^2 testu).
5. V každém cyklu a pro každou nenulovou hodnotu x mající hodnotu $-4 \leq x \leq -1$ a $1 \leq x \leq 4$, spočti počet výskytů v každém cyklu.
6. Pro každý z osmi možných stavů x, vypočti $v_k(x)$ jako celkový počet cyklů, ve kterých se x objeví přesně k-krát, pro k = 0, 1, ..., 5 (všechny počty výskytů větší než 5 jsou uloženy v $v_5(x)$).
7. Pro každý z osmi možných stavů x spočti $\chi^2_{(obs)} = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$,
   kde $\pi_k(x)$ je pravděpodobnost k výskytů x v náhodném rozdělení[18].
8. Pro každé x spočti $P_x = igamc\left(\frac{5}{2}, \frac{\chi^2_{(obs)}}{2}\right)$

Test je považován za úspěšný pokud všechna $P_x \geq 0,01$

Výsledky generátorů:

Generátor se dvěma oscilátory

x	χ^2	P	Výsledek
-4	3.418933	0.635688	ok
-3	4.093627	0.536016	ok
-2	9.284789	0.098230	ok
-1	4.175355	0.524455	ok
1	5.454976	0.362908	ok
2	6.158700	0.291082	ok
3	13.241001	0.021222	ok
4	15.952163	0.006982	fail

Generátor se Zenerovou diodou

x	χ^2	P	Výsledek
-4	2.081409	0.837770	ok
-3	1.097907	0.954289	ok
-2	12.190734	0.032266	ok
-1	0.367825	0.996170	ok
1	1.109943	0.953220	ok
2	2.510804	0.774867	ok
3	5.284034	0.382208	ok
4	11.359925	0.044693	ok

Závěr
V prvním případě nám jedna hodnota překročila stanovenou míru pravděpodobnosti. Vzhledem k tomu, že ostatní výsledky splňují požadovanou míru pravděpodobnosti s rezervou jde o výsledek podezřelý. Není však nutné kvůli němu odmítat celou posloupnost a je potřeba brát ohledy na výsledky dalších testů.

6.15 Variantní test náhodné procházky

Variantní test náhodné procházky je rozšířenou verzí předchozího testu Pro osmnáct možných stavů.

Výsledky generátorů:

Generátor se dvěma oscilátory

x	Počet návštěv	P	Výsledek
-9	485	0.313053	OK
-8	529	0.450433	OK
-7	562	0.579963	OK
-6	588	0.702959	OK
-5	625	0.940257	OK
-4	625	0.932276	OK
-3	596	0.641895	OK
-2	609	0.696955	OK
-1	651	0.612934	OK
1	567	0.063606	OK
2	510	0.045950	OK
3	449	0.020740	OK
4	426	0.027885	OK
5	463	0.111247	OK
6	469	0.164611	OK
7	445	0.142800	OK
8	449	0.181802	OK
9	457	0.230257	OK

Generátor se Zenerovou diodou

x	Počet návštěv	P	Výsledek
-9	1299	0.762408	OK
-8	1356	0.536751	OK
-7	1402	0.357546	OK
-6	1386	0.366413	OK
-5	1306	0.643787	OK
-4	1274	0.778588	OK
-3	1285	0.666050	OK
-2	1290	0.538424	OK
-1	1270	0.507037	OK
1	1230	0.888080	OK
2	1228	0.916798	OK
3	1169	0.540936	OK
4	1045	0.144568	OK
5	956	0.059680	OK
6	1010	0.168811	OK
7	1041	0.274433	OK
8	978	0.178793	OK
9	985	0.219152	OK

Závěr
V obou případech není důvod se domnívat, že by testovaná sekvence nebyla náhodná.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a otestovat dva kryptograficky bezpečné generátory náhodných čísel. Po provedení mnoha statistických testů je možno říci, že oba realizované generátory obstály na výbornou a jsou zcela srovnatelné s běžně dostupnými komerčními generátory.

Literatura

[1] 18031, I.: *Information technology ? Security techniques ? Random bit generation.* ISO, Geneva, Switzerland.

[2] A. J. Menezes, P. C. v. O.; Vanstone, S. A.: *Handbook of Applied Cryptography (Discrete Mathematics and Its Applications).* CRC Press, 1996, ISBN 0849385237.

[3] Blum, L.; Blum, M.; Shub, M.: A simple unpredictable pseudo random number generator. *SIAM J. Comput.*, ročník 15, č. 2, Květen 1986: s. 364–383, ISSN 0097-5397.

[4] Dichtl, M.: Bad and Good Ways of Post-processing Biased Physical Random Numbers. In *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, *Lecture Notes in Computer Science*, ročník 4593, editace A. Biryukov, Springer, 2007, ISBN 978-3-540-74617-1, s. 137–152.

[5] FIPS: *Security Requirements for Cryptographic Modules*, ročník 140-1. 1994.

[6] FIPS: *Security Requirements for Cryptographic Modules*, ročník 140-2. 2001, viii + 61 s.

[7] J. Hrubý, L.: Kvantový šumátor a jeho testování. Technická zpráva, Institute of ComputerScience, Academy of Sciences of the Czech Republic, 2005.

[8] Killmann, W.: *Functionality classes for random number generators.* Bundesamt für Sicherheit in der Informationstechnik, 2011.

[9] Kim, S.-J.; Umeno, K.; Hasegawa, A.: Corrections of the NIST Statistical Test Suite for Randomness. Cryptology ePrint Archive, Report 2004/018, 2004.

[10] Knuth, D. E.: *The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997, ISBN 0201896842.

[11] Lacharme, P.: Fast Software Encryption. kapitola Post-Processing Functions for a Biased Physical Random Number Generator, Berlin, Heidelberg: Springer-Verlag, 2008, ISBN 978-3-540-71038-7, s. 334–342.

[12] L'Ecuyer, P.; Simard, R.: *TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators.* 2002.

[13] Markovski, S.; Gligoroski, D.; Kocarev, L.: Unbiased Random Sequences from Quasigroup String Transformations. In *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, *Lecture Notes in Computer Science*, ročník 3557, editace H. Gilbert; H. Handschuh, Springer, 2005, ISBN 3-540-26541-4, s. 163–180.

[14] Mascagni, M.; Srinivasan, A.: Algorithm 806: SPRNG: a scalable library for pseudorandom number generation. *ACM Trans. Math. Softw.*, ročník 26, č. 3, Září 2000: s. 436–461, ISSN 0098-3500, doi:10.1145/358407.358427.

[15] Maurer, U.: A Universal Statistical Test for Random Bit Generators. *Journal of cryptology*, ročník 5, 1992: s. 89–105.

[16] Micali, S.; Schnorr, C. P.: Perfect Polynomial Random Number Generators. Technická zpráva, Chicago, IL, USA, 1990.

[17] Neumann, J. V.: Various Techniques used in Connection with Random Digits. *National Bureau of Standards Applied Mathematics Series 12*, , č. 12, 1951: s. 36–38.

[18] Rukhin, A.; Soto, J.; Nechvatal, J.; aj.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. 2001.

[19] Schottky, W.: U̇ber spontane Stromschwankungen in verschiedenen Elektrizitätsleitern. *Annalen der Physik*, , č. 57, 1918: s. 541–567.

[20] SOMLO, P. I.: Zener-diode noise generators. *Eletronics letters*, 1975.

[21] Soo, N.: Jitter Measurement Techniques. Technická zpráva, Pericom, 2000.

[22] Sussans, D. E.: Noise calibrator for v.h.f. and u.h.f. field-strength measuring receivers. *Electron. Lett.*, 1967.

[23] Texas Instruments: *MSP-EXP430G2 LaunchPad Experimenter Board User's Guide.*

[24] Westlund, L.: Random Number Generation Using the MSP430. Technická zpráva, Texas Instruments Incorporated, 2006.

[25] WWW stránky: MSP430 LaunchPad (MSP-EXP430G2) - Texas Instruments Embedded Processors Wiki. http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_(MSP-EXP430G2).

Příloha A

Obsah CD

Testované sekvence
vlo.bin - Sekvence z generátoru se dvěma oscilátory
adc.bin - Sekvence z generátoru se zenerovou diodou

Zdrojové kódy pro vývojový kit LanuchPad
vlo.c - Zdrojový kód ke generátoru se dvěma oscilátory
adc.c - Zdrojový kód ke generátoru se zenerovou diodou