Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Školící pracoviště:
Společná laboratoř optiky
UP a Fzú AVČR

BAKALÁŘSKÁ PRÁCE

Generace náhodných čísel prostřednictvím lavinových fotodiod

Vypracoval: **Jan Jašek**
Studijní program: B1701 Fyzika
Studijní obor: 3942R001 Nanotechnologie
Forma studia: Prezenční
Garantující pracoviště: Katedra experimentální fyziky
Vedoucí diplomové práce: Doc. Mgr. Karel Lemr, Ph.D.
Rok: 2017

Prohlášení

Prohlašuji, že jsem předloženou bakalářskou práci vypracoval samostatně pod vedením doc. Mgr. Karla Lemra, Ph.D. a že jsem použil zdrojů, které cituji a uvádím v seznamu použitých pramenů.

V Olomouci dne 9. května 2017

.....

Jan Jašek

Poděkování

Rád bych poděkoval doc. Mgr. Karlu Lemrovi, Ph.D. a Mgr. Antonínu Černochovi, Ph.D. za jejich rady, ochotu a čas, který mi věnovali.

V Olomouci dne 9. května 2017

.....

Jan Jašek

Bibliografická identifikace

Jméno a příjmení autora	Jan Jašek
Název práce	Generace náhodných čísel prostřednictvím lavinových fotodiod
Typ práce	Bakalářská
Garantující pracoviště	Katedra experimentální fyziky
Školící pracoviště	Společná laboratoř optiky Univerzity Palackého a Fyzikálního ústavu Akademie věd České republiky
Vedoucí práce	Doc. Mgr. Karel Lemr, Ph.D.
Rok obhajoby práce	2017
Abstrakt	Náhodná čísla jsou důležitá pro celou řadu aplikací a metody jejich generace jsou v současné době často diskutovaným tématem, zejména z pohledu kryptografie a počítačového zabezpečení. Tato práce si klade za cíl popsat a otestovat možnosti využití lavinových fotodiod operujících v Geigerově režimu pro vytvoření kvantového generátoru pravých náhodných čísel. Hlavní předností našeho generátoru je časově úsporná generace a efektivita zpracování dat, přičemž díky principům kvantové mechaniky by takto generovaná náhodná čísla měla splňovat i vysoké nároky moderních kryptografických aplikací. Jako zdrojové náhodné jevy využíváme měření temných detekcí lavinových fotodiod a měření polarizačního stavu jednotlivých fotonů.
Klíčová slova	Generátory náhodných čísel, temné detekce, lavinové fotodiody, Geigerův režim, Peresův algoritmus, polarizační dělič fotonů
Počet stran	46
Počet příloh	0
Jazyk	Český

Bibliographical identification

Autor's first name and surname	Jan Jašek
Title	Generation of random numbers using avalanche photodiodes
Type of thesis	Bachelor
Garanting department	Department of Experimental Physics
Supervising department	Joint Laboratory of Optics of Palacký University and Institute of Physics of Czech Academy of Science
Supervisor	Doc. Mgr. Karel Lemr, Ph.D.
The year of presentation	2017
Abstract	Random numbers are frequently used in a number of applications and the methods of random number generation are frequently discussed, especially in cryptography and computer security. The goal of this thesis is to describe and test the capabilities of Geiger-mode avalanche photodiodes to generate true random numbers. The main advantage of our generator is time and data processing efficiency while, thanks to the principles of quantum mechanics, our generated random numbers meet even the highest requirements for modern cryptographical applications. As a source of the random events, we use both the avalanche photodiode's dark counts and the single-photons detections.
Keywords	Random number generator, dark counts, avalanche photodiode, Geiger mode, Peres algorithm, polarization beamsplitter
Number of pages	46
Number of appendices	0
Language	Czech

Obsah

Úvod **7**

1 Principy a metody **8**
1.1 Náhodná čísla 8
1.1.1 Generování náhodných čísel 8
1.1.2 Náhodnost a nepředvídatelnost 9
1.2 NIST Statistical Test Suite 11
1.2.1 Nulová hypotéza, kritická hodnota a P-hodnota 11
1.2.2 Interpretace testů 12
1.2.3 Testy NIST STS 13
1.3 Lavinové fotodiody 15
1.4 Temné detekce 16
1.5 Extrahování bitů z neuniformní náhodné sekvence 16
1.5.1 Peresův algoritmus 17
1.6 Shannonova entropie 19
1.7 Polarizace světla 19
1.7.1 Polarizační děliče 20

2 Experimentální část a výsledky **22**
2.1 Popis experimentu a použitá rozhraní 22
2.2 Generace náhodných dat měřením počtu temných detekcí 22
2.2.1 Sudá/Lichá 24
2.2.2 Medián 25
2.2.3 Binární přepis 27
2.2.4 XOR 28
2.2.5 XOR (2x) 28
2.2.6 XOR (3x) 29
2.2.7 Inverzní Box-Mullerova transformace 30
2.3 Generace náhodných dat měřením sledu temných detekcí 34
2.4 Generace náhodných dat měřením polarizace jednotlivých fotonů . 39

Závěr **43**

Úvod

Náhodná čísla jsou důležitá pro mnoho aplikací, jako jsou statistické výzkumy, numerické výpočty, modelování a simulace (např. Monte Carlo), mnohé hry a loterie [1]. Od vzniku výpočetní techniky se pak náhodná čísla začala hojně využívat pro potřeby kryptografických aplikací, jako je generace bezpečnostních klíčů, šifrování, aj. [1–5].

Poptávka po vhodných generátorech náhodných čísel je zejména v oblasti moderní kryptografie velmi vysoká. V dnešní době není problém opatřit si sadu náhodných čísel vygenerovaných například programem v počítači, ale jsou tato čísla doopravdy náhodná? Koneckonců, počítač je jen nástroj, co plní určitou sadu příkazů, a je tedy naprosto předvídatelný. Pokud chceme generovat doopravdy náhodná a nepředvídatelná čísla, je třeba využít fyzikální jevy – ať už klasické nebo kvantové [1, 2].

Ačkoliv metody klasické fyziky nabízejí poměrně široké možnosti k pozorování a měření náhodných jevů (např. atmosférický šum [2], nebo šum elektrického obvodu [6]), ukazuje se, že tyto jevy jen spadají do kategorie chaotických systému, které jsou ve skutečnosti zcela deterministické [1]. Matematický popis těchto jevů je natolik složitý, že i velmi malá odchylka ve vstupních parametrech způsobí naprosto odlišný vývoj [2, 7]. Jevy klasické fyziky se tedy jeví náhodné pouze do té doby, než přesně určíme jejich vstupní parametry [1]. V tu chvíli jsme jev schopni přesně zopakovat, což je například v šifrování samozřejmě nežádoucí.

Přístup klasické fyziky v těchto případech selhává. Nabízí se otázka, jaká je situace z pohledu fyziky kvantové. Kvantové jevy obvykle existují v tzv. superpozici (sloučení více stavů) a až akt měření je donutí „zvolit“ jednu konkrétní hodnotu příslušící určitému stavu. Tyto výsledky nejsou (s výjimkou systémů připravených ve vlastním stavu měřidla) deterministické a nemůžeme je tedy jednoznačně předpovědět ani zopakovat. Můžeme jen spočítat pravděpodobnost, že takový výsledek nastane [8]. Výsledky měření kvantových jevů tedy interpretujeme statisticky a může nám tak posloužit jako ideální zdroj nahodilosti.

Cílem této práce je popsat a otestovat možnosti využití temných detekcí lavinových fotodiod operujících v Geigerově režimu a polarizačního dělení fotonů k vytvoření kvantového generátoru pravých náhodných čísel. Hlavní předností našeho generátoru je časově úsporná generace a efektivita zpracování dat, přičemž díky principům kvantové mechaniky by takto generovaná náhodná čísla měla splňovat i vysoké nároky moderních kryptografických aplikací. Práce je rozdělena do hlavních dvou kapitol, z nichž první pojednává o základních principech náhodnosti i jejím testování a popisuje principy a metody potřebné pro jejich generaci. Kapitola druhá pak rozebírá samotné experimenty, využívající temné detekce lavinové fotodiody a dělení fotonů na polarizačním děliči ke generaci pravých náhodných čísel spolu s algoritmy potřebnými k úpravě naměřených dat a výsledky statistického testování.

Kapitola 1

Principy a metody

1.1 Náhodná čísla

1.1.1 Generování náhodných čísel

Existují dvě možnosti, jak generovat náhodná čísla. První a nejčastěji využívaná možnost zahrnuje použití deterministického matematického algoritmu [9]. Takovýto generátor pak nazveme pseudonáhodný generátor náhodných čísel (PRNG z anglického pseudorandom number generator) [3]. Druhou možností je použít generátor pravých náhodných čísel (TRNG z anglického true random number generator), který využívá fyzikálního jevu – ať už chaotického procesu nebo jevu kvantového (v takovém případě označujeme generátor jako QRNG, kvantový generátor náhodných čísel) [9]. Pro naše potřeby generátory produkují bitový řetězec (sekvenci) náhodných 0 a 1, které mohou být dle potřeby dále rozděleny na menší podřetězce [3].

Pseudonáhodná čísla nejsou klasickými náhodnými čísly, jak už ostatně napovídá jejich jméno. V podstatě jde jen o výstup počítačových algoritmů, které pro generaci náhodných čísel využívají (obvykle) náhodný vstupní parametr nazývaný seed [3]. Algoritmy samotné jsou zcela deterministické a často i veřejné. Proto pokud je odhalen seed, je známá i celá výsledná sekvence, což je poněkud nevhodné například právě pro kryptografické aplikace. Na druhou stranu je pak pomocí PRNG možno generovat náhodná čísla velmi rychle, což může být dostačující předností například při simulacích Monte Carlo [2]. Tyto algoritmy také mají sice velmi dlouhou, ale přesto konečnou periodu, kdy se po určité době začíná generovaná sekvence přesně opakovat.

Čísla generovaná matematickými algoritmy tedy nejsou skutečně náhodná, a tudíž jsou jen pseudonáhodná. Pro dosažení nepředvídatelnosti se pak jako seed často používá výstup z nějakého fyzikálního generátoru náhodných čísel, který tyto nedostatky částečně vyvažuje [3]. Některé PRNG pak ke generaci seedu mohou využívat i lidskou činnost (například doba mezi údery kláves klávesnice [2], pohyb myši na obrazovce [3]). Je také třeba být na pozoru, který algoritmus používáme a kdo jej ve skutečnosti implementoval.

Generátory pravých náhodných čísel vždy využívají za svůj základ fyzikální jevy [10], speciální důraz je pak kladen na jevy kvantové mechaniky. Zájem o tyto generátory náhodných čísel stoupl zejména v posledních letech, kdy se začala stále více projevovat nedostatečná kryptografická bezpečnost spojená s používáním

pseudonáhodných čísel. Ta vyústila v prolomení bezpečnosti několika různých systémů, jako například zabezpečení operačního systému Windows [11], aj. [12].

Odpovědí těmto problémům byla celá řada návrhů generátorů pravých náhodných čísel, včetně těch kvantových. Takovéto generátory jsou postaveny na měření různých fyzikálních jevů, například radioaktivní rozpad [13], měření intenzity dopadajícího osvětlení na mobilním telefonu [12], měření časového intervalu mezi fotony jednofotonového zdroje [9], a další. Tyto generátory se liší zejména rychlostí generace dat, která se pohybuje řádově mezi 10 – 1000 kbit za sekundu, dále pak dobou a účinnosti zpracování nasbíraných dat.

Zamysleme se nyní nad tím, jaké vlastnosti by měl vykazovat ideální generátor pravých náhodných čísel. Ilustrovat takovýto generátor můžeme například pomocí hodů mince, která má strany označené čísly „0“ a „1“. Mince samotná bude samozřejmě dokonale vyvážená (tedy ideální). Co vlastně od takového TRNG požadujeme?

V prvé řadě by výsledná data měla být uniformně rozdělená, tedy každý výsledek je zastoupen se stejnou četností. Pro naši minci to znamená, že po určitém počtu hodů by mělo padnout stejné množství 0 i 1. To by naše mince splnit měla, přece jen je dokonale vyvážená. Druhým požadavkem je nezávislost jednotlivých výsledků. To naše mince splňuje, výsledek předchozího hodu neovlivní další hod ani po libovolném množství hodů. Třetí obecnou podmínkou je, aby ošetřením vstupních parametrů nebyla ovlivněna nepředvídatelnost generátoru. Pro hod naší mince by toto ošetření představovalo přesné změření její polohy, orientace, hybnosti a momentu hybnosti v počátku hodu.

Takováto mince tedy představuje velmi dobrý generátor náhodných čísel. Samozřejmě, i kdybychom takovouto „férovou“ minci dokázali vytvořit, házení mincí pro vytvoření řetězců tvořených miliony bitů by bylo poněkud nepraktické – rychlost generace bitů by byla příliš nízká. TRNG proto obvykle využívají fyzikální jevy, které jsou dostatečně nepředvídatelné (například atmosférický šum nebo šum elektrického obvodu [2, 3]), v případě QRNG pak jevy kvantové (radioaktivní rozpad, foton na polarizačním děliči, aj. [14]), které jsou nepředvídatelné již ze své podstaty [8]. Právě těmito typy generátorů se tato práce zabývá. Zásadní nevýhodou TRNG (případně QRNG) oproti PRNG je nižší rychlost generace [3], v závislosti na použitém fyzikálním jevu pak může být problematický i způsob měření výsledků, případně cena použité experimentální aparatury.

1.1.2 Náhodnost a nepředvídatelnost

Požadovanou vlastností každého generátorů náhodných čísel je nepředvídatelnost výsledků [3]. V případě PRNG to znamená, že pokud je neznámý seed, tak ze znalosti libovolného počtu předchozích hodnot by nemělo být možno dopředu předpovědět další výsledek generátoru. Stejně tak by ve výsledné sekvenci nemělo být možné najít jakýkoliv vzor či vztah mezi jednotlivými bity, který by umožnil seed odhalit [3]. Jak je to ale v případě TRNG a proč je požadovanou vlastností právě nepředvídatelnost?

Jak náhodnost, tak i nepředvídatelnost vyjadřuje naši neschopnost najít v určité sekvenci vzor, který by nám umožnil předpovědět příští výsledek [3]. Zatímco ale nepředvídatelné jevy vykazují vzory jen těžko rozlišitelné, opravdu náhodné jevy nevykazují vzory vůbec žádné. Vztah náhodnosti a nepředvídatelnosti tedy

můžeme popsat takto: skutečně náhodný jev je nepředvídatelný. Neschopnost předvídat jev však ještě nezaručuje jeho náhodnost.

Již v úvodu bylo naznačeno, že jevy klasické fyziky jsou deterministické a tudiž nenáhodné. To je velmi silné tvrzení, na které mezi vědeckou komunitou dodnes nebyl vyvozen jednoznačný názor [2]. V dnešní době je možné jakožto generátor pravých náhodných čísel využít dva možné zdroje – měření kvantových jevů a chaotické systémy [2]. Teoretický rozdíl je značný – výsledky měření kvantových jevů jsou (vyjma systémů připravených ve vlastním stavu měřidla) samy o sobě definovány statistikou interpretací a jejich náhodnost (a tedy i nepředvídatelnost) plyne ze samotných definic kvantové mechaniky [8]. Chaotické systémy pak představují jevy, jejichž matematický model je natolik složitý, že jakákoliv nepřesnost ve vstupních parametrech způsobí dramatickou změnu ve vývoji celého systému [2, 7]. Jako příklad můžeme uvést dvojkyvadlo, nebo vývoj počasí. Přesné ošetření těchto vstupních parametrů zanechá přesně definovaný dynamický systém, který v žádném případě nepředvídatelný není [2, 7].

Z praktického hlediska se díváme na poněkud odlišnou situaci. Přesné vymezení vstupních parametrů by vskutku zanechalo chaotické systémy deterministické, nicméně tohoto vymezení není ve většině reálných situací možno docílit [7]. Jako příklad můžeme využít atmosférický šum, který je dnes jako zdroj náhodných čísel hojně používán [2]. Pro ošetření vstupních parametrů tohoto jevu by pravděpodobně bylo třeba znát rychlost a směr pohybu každé molekuly v atmosféře planety [2]. Tento požadavek je v dnešní době naprosto nereálný, a atmosférický šum tedy můžeme považovat za dostatečný (ne-li naprostý) zdroj náhodnosti. Nepředvídatelnost TRNG tedy můžeme ošetřit použitím vhodného fyzikálního jevu.

Jak již bylo řečeno, nepředvídatelnost vyjadřuje jen naši neschopnost najít v náhodné sekvenci vzor, který by nám umožnil předpovědět příští výsledek, ne skutečnou náhodnost. Je třeba si proto uvědomit, že například několik století zpátky by pravděpodobně čísla pseudonáhodná byla interpretována jako pravá náhodná čísla, jelikož by bez pomocí výpočetní techniky nebylo možné objevit jejich konečnou opakovací periodu. Z podobného důvodu v dnešní době můžeme jako náhodné jevy interpretovat chaotické systémy - zabraňuje nám v tom jen technologický pokrok. Můžeme tedy říct, že nepředvídatelnost vyjadřuje pouze to, jak náhodně na nás jev působí, a právě to jsme schopni testovat.

Maximální možná přesnost měření se ovšem neustále zvyšuje, a proto jednoho dne i dnes reálně neměřitelné parametry mohou být měřitelné. Oproti tomu jevy kvantové zůstanou (alespoň podle současných teoriích) navždy ve světě statistiky [8], a tedy i ideálním zdrojem skutečné náhodnosti.

Shrňme si myšlenku podkapitoly 1.1.2 do několika bodů:

- Požadovanou (a tedy i testovanou) vlastností generátorů náhodných čísel je nepředvídatelnost výsledných čísel. Tu můžeme zajistit použitím vhodného fyzikálního jevu.
- Nepředvídatelnost nemusí nutně implikovat náhodnost, ovšem skutečně náhodné jevy jsou vždy nepředvídatelné.
- Akt měření jevů kvantové mechaniky je ze své definice skutečně náhodný, a tudíž i nepředvídatelný.

Jelikož tato práce popisuje generování náhodných čísel pomocí měření jevů kvantové optiky, budeme nepředvídatelnost a náhodnost považovat za ekvivalentní.

1.2 NIST Statistical Test Suite

Pro testování náhodnosti výsledků generátorů náhodných čísel je třeba použít vhodný testovací software. V této práci jsme zvolili testovací sadu společnosti NIST. NIST (z anglického National Institute of Standards and Technology) je jednou z nejstarších amerických vědeckých organizací. Její součástí je také divize pro počítačové zabezpečení, která pro potřeby testování generátorů náhodných čísel vhodných pro šifrování vydala statistický testovací balík NIST Statistical Test Suite (STS). Tento testovací balík nám poslouží jako výchozí posuzovací kritérium, zda náš generátor náhodných čísel splňuje nároky náhodnosti pro kryptografické účely. Následující dvě podkapitoly stručně zmíní, jak STS generátory testuje a jaké konkrétní testy využívá. Detailní informace je možno nalézt v manuálu [3].

1.2.1 Nulová hypotéza, kritická hodnota a P-hodnota

Vlastnosti náhodné sekvence můžeme popsat pomocí pravděpodobnosti. Míra náhodnosti sekvence tedy může být charakterizována pomocí statistických pravděpodobnostních testů. Tyto testy obvykle vyhledávají jakousi pravidelnost, vzor, kterým se generátor řídí a odchyluje se tak od náhodnosti. Statistické testy vždy porovnávají dvě hypotézy – nulovou hypotézu H_0 (ta v našem případě říká, že je sekvence náhodná) a alternativní hypotézu H_A (která pro nás naopak říká, že sekvence náhodná není). Výsledkem každého provedeného testu je tedy rozhodnutí, zda byla H_0 přijata, nebo odmítnuta.

Skutečný výsledek testu je nicméně jen určitá pravděpodobnost, čili číslo, a přijetí nebo odmítnutí H_0 tedy závisí jen na naší interpretaci tohoto čísla. V této souvislosti se volí (obvykle velmi nepravděpodobná) kritická hodnota, která je porovnána s výsledkem testu. Pokud tento výsledek kritickou hodnotu překročí, je interpretován jako nenáhodný. Výsledek ovšem náhodný být může, jen pravděpodobnost, že k němu dojde, je velmi malá. To může mít za následek chyby statistické interpretace. Možné výsledky statistického testování shrnuje tabulka 1.1.

Tabulka 1.1: Souhrn možných výsledků statisticých testů. Přeloženo z [3].

	Vyhodnocení	
Skutečná situace	H_0 přijata	H_0 odmítnuta
Náhodná data	Bez chyby	Chyba typu I
Nenáhodná data	Chyba typu II	Bez chyby

Statistická chyba prvního druhu (falešně pozitivní) tedy zahrnuje situaci, kdy je náhodný výsledek interpretován jako nenáhodný (chybně se detekuje nenáhodnost) a H_0 je chybně odmítnuta. Statistická chyba druhého druhu (falešně negativní) pak nastává tehdy, kdy jsou nenáhodná data interpretována jako náhodná, a H_0 je chybně přijata (nenáhodnost se chybně nedetekuje).

Pravděpodobnost výskytu chyby prvního druhu se obvykle nazývá hodnota významnosti a značíme ji α. Hodnotu α můžeme nastavit před proběhnutím testu a její obvyklá hodnota v kryptografii je $\alpha = 0{,}01$. Dále pak ve statistice určujeme tzv. P-hodnotu, kterou definujeme jako pravděpodobnost, že při splnění nulové hypotézy dojde k získání stejného nebo extrémnějšího výsledku, než je výsledek námi vypočtený. Jinými slovy, P-hodnota představuje pravděpodobnost, že by dokonalý generátor náhodných čísel vyprodukoval sekvenci stejně nebo méně nepředvídatelnou, než byla testovaná sekvence. Platí, že pokud P-hodnota $\geq \alpha$, tak je nulová hypotéza přijata. Pokud je naopak P-hodnota $< \alpha$, je nulová hypotéza odmítnuta.

1.2.2 Interpretace testů

NIST STS jako vstup vždy přijme binární datový řetězec, který reprezentuje naměřená data. Tento řetězec vždy rozdělí na uživatelem zvolený počet binárních sekvencí a ty následně testuje jednotlivě. Toto dělení je nutné pro výslednou interpretaci testů. Koneckonců, výsledkem statistického testu pro každou sekvenci je pouze empirická P-hodnota. Na základě těchto P-hodnot je pro každou sekvenci přijata nebo odmítnuta nulová hypotéza.

NIST STS volí pro interpretaci výsledků dva přístupy. Prvním je porovnání výsledného počtu přijatých sekvencí a odmítnutých sekvencí, čímž můžeme ošetřit, zda se v řetězci nevyskytují lokální nenáhodné anomálie, tedy zda zastoupení nepravděpodobných výsledků odpovídá pravděpodobnosti jejich rozložení. Dále pak můžeme porovnat distribuci výsledných P-hodnot a určit tak jejich uniformnost. Pokud by sekvence vykazovaly stále stejnou P-hodnotu, pravděpodobně by generátor vytvářel periodicky stejná data, což je v našem případě nežádoucí.

Minimální poměr prošlých sekvencí ku celkovému počtu sekvencí je dán volbou hodnoty významnosti α, tedy statistickou chybou prvního druhu. α vlastně značí střední hodnotu neprošlých sekvencí a v případě dokonale náhodných čísel by jí pro nekonečný počet měřených sekvencí měl být tento počet roven. Pro konečný počet je ale třeba stanovit určitou nejistotu, která zohlední rozptyl v počtu prošlých sekvencí. Tuto nejistotu označujeme jako interval spolehlivosti I, který můžeme vypočítat dle vzorce (1.1)

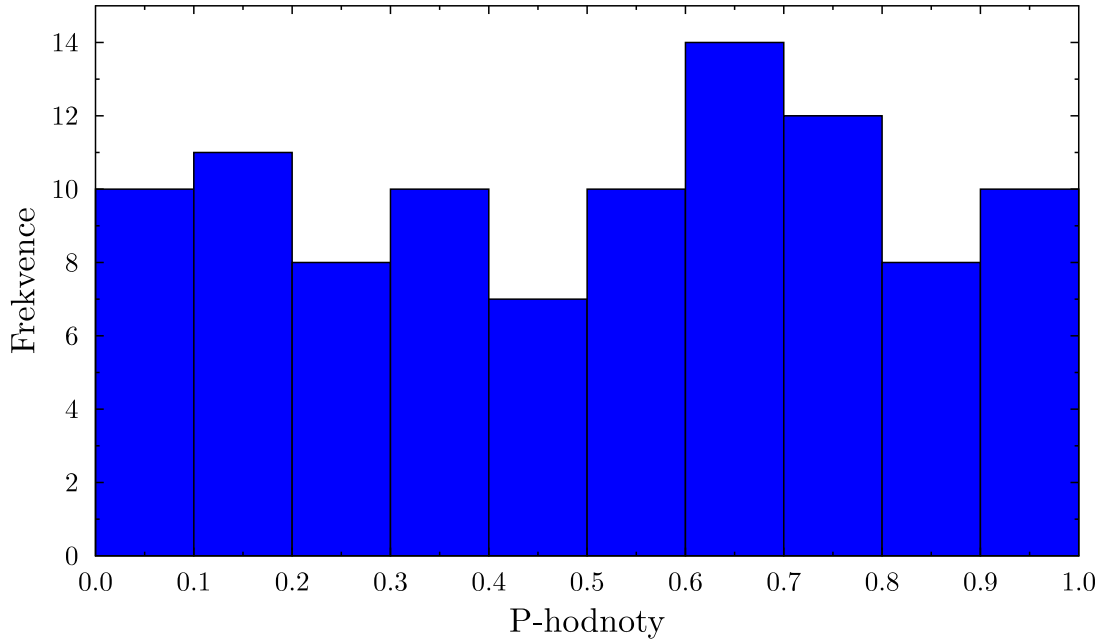
$$I = p \pm 3\sqrt{\frac{p\,(1-p)}{m}}, \tag{1.1}$$

kde m je celkový počet sekvencí, $p = (1 - \alpha)$ a α je zvolená hodnota významnosti. Pokud poměr celkových ku přijatým sekvencím spadá do tohoto intervalu, můžeme data považovat za náhodná. Jako příklad vezměme hodnotu $\alpha = 0{,}01$ a $m = 100$. Po dosazení do vzorce (1.1) dostáváme hodnotu $I = 0{,}99 \pm 0{,}03$. Nejnižší počet přijatých sekvencí je tedy 96/100. Horní hranice výsledku přesahuje 100 %, což je způsobeno tím, že tento interval spolehlivosti byl primárně určen pro počet sekvencí ≥ 1000.

Vyšetřením distribuce ověříme uniformnost výsledných P-hodnot, které mohou ležet v intervalu 0 – 1. Tento interval rozčleníme na 10 menších podintervalů a vytvoříme histogram námi naměřených P-hodnot, jehož příklad můžeme vidět na obrázku 1.1. Matematicky pak můžeme uniformnost P-hodnot zajistit pomocí kritéria χ^2, vyjádřeného vzorcem (1.2)

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(F_i - \frac{S}{10}\right)^2}{\frac{S}{10}}, \tag{1.2}$$

kde F_i je počet P-hodnot v podintervalu i a S je velikost původního binárního řetězce. Dále určíme hodnotu P_T popisující pravděpodobnost obdržení hodnot χ^2 nebo extrémnějších. Dle výchozích parametrů NIST STS splňují P-hodnoty jednotlivých sekvencí uniformnost, pokud $P_T \geq 0{,}0001$. Právě P_T je vypsána pro každý z testů v tabulkách v experimentální části, kde je (v souladu s NIST STS) nazvána jednoduše jako P-hodnota.



Obrázek 1.1: Příklad histogramu P-hodnot, kde osa x představuje jednotlivé podintervaly P-hodnot a osa y vyjadřuje četnost P-hodnot v jednotlivých podintervalech.

1.2.3 Testy NIST STS

NIST STS se skládá z celkem patnácti výchozích testů. Jejich detailní popis, doporučené vstupní sekvence a příklady použití můžeme najít ve zdroji [3].

1. **Frekvenční test** z anglického „Frequency test“. Test kontroluje zda počet bitů 0 a 1 v sekvenci je přibližně stejný, jak by to mělo pro náhodnou sekvenci být.

2. **Blokově frekvenční test** z anglického „Frequency within a block test“. Test rozdělí sekvenci na podsekvence o délce m bitů a testuje, zda je počet bitů 1 přibližně rovný m/2. Pro m = 1 test odpovídá frekvenčnímu testu.

3. **Test běhů** z anglického „Runs test“. Test posuzuje, jestli se v sekvenci nevyskytuje nepřiměřený počet tzv. běhů, tedy nepřerušených řetězců stejných hodnot (např. ’00000’), a to pro různé délky těchto řetězců.

4. **Test nejdelšího běhu** z anglického „Test for the longest run“. Test zkoumá nejdelší běh bitů 1 uvnitř podsekvencí o délce m bitů. Velikost m je nastavena automaticky.

5. **Test binární hodnosti matice** z anglického „Binary matrix rank test“. Test rozdělí bity sekvence do řádků M a sloupců Q a vytvoří z nich matice. Následně zkoumá hodnost těchto matic a testuje tak lineární závislost podsekvencí z původní sekvence.

6. **Test diskrétní Fourierovou transformací** z anglického „Discrete Fourier transform test“. Cílem testu je zkontrolovat, zda se v sekvenci neobjevují periodicky opakující se vzory, a to pomocí kontroly maxim vzniklých diskrétní Fourierovou transformací.

7. **Test nepřekrývajících se šablon** z anglického „Non-overlapping template matching test“. Testuje kolikrát se v sekvenci vyskytuje motiv různých (například iracionálních) čísel. Tato čísla jsou předem definovaná a uložená v tzv. šablonách. Pokud je motiv nalezen, pozice testování v sekvenci se resetuje na první bit za nalezeným motivem.

8. **Test překrývajících se šablon** z anglického „Overlapping template matching test“. Test funguje analogicky k testu nepřekrývajících se šablon, ale pozice testování v sekvenci se neresetuje ani v případě nalezení vzoru, vždy se jen posune o jeden bit dopředu.

9. **Mauerův universální statistický test** z anglického „Mauer's universal statistical test“. Test zkoumá počet bitů mezi souhlasnými motivy a zjišťuje, nakolik je možné sekvenci komprimovat beze ztráty informace. Možnost velké bezztrátové komprese dat poukazuje na odchylky od náhodnosti.

10. **Test lineární komplexity** z anglického „Linear complexity test“. Test zkoumá velikost posuvného registru s lineární zpětnou vazbou (LFSR z anglického „Linear feedback shift register“) a dle této velikosti určuje komplexnost sekvence. Náhodným sekvencím odpovídá dlouhé LFSR.

11. **Sériový test** z anglického „Serial test“. Test porovnává četnost všech nalezených překrývajících se motivů o délce m v celé sekvenci a má za cíl určit, zda počet těchto motivů odpovídá náhodné sekvenci. Náhodné sekvence se vyznačují uniformností, tedy všechny různě dlouhé motivy mají stejnou šanci na výskyt. Pro m = 1 jde opět o frekvenční test.

12. **Test přibližné entropie** z anglického „Approximate entrophy test“. Funguje podobně jako sériový test, ale porovnává četnost překrývajících se bloků o dvou po sobě jdoucích velikostech (m a m + 1) a porovnává je s očekávanými výsledky pro náhodnou sekvenci.

13. **Test kumulativní sumy** z anglického „Cumulative sums test“. Tento test hledá podezřele se opakující zakončení náhodných procházek na daném místě. Program rozdělí sekvenci na různě dlouhé podsekvence a převede bity 0 v sekvenci na znaky -1 a bity 1 na znaky +1. Následně testuje maximální odchylku od náhodné procházky definovanou kumulativní sumou.

14. **Test náhodné procházky** z anglického „Random excursions test“. Série osmi testů. Podobně jako v případě testu kumulativní sumy rozdělí sekvenci na menší podsekvence a převede bity 0 v sekvenci na znaky -1 a bity 1 na

znaky +1. Následně simuluje uzavřené náhodné procházky končící v počátečním bodě a hledá, zdali v příliš mnoho cyklech nedošlo k právě N návštěvám zvoleného místa. Tedy zda místo není navštíveno vždy ve stejném počtu kroků.

15. **Test variací náhodné procházky** z anglického „Random excursions variant test“. Série osmnácti testů. Test funguje analogicky k testu náhodné procházky, ale hledá podezřele často navštěvovaná místa.

1.3 Lavinové fotodiody

Fotodiody jsou fotoelektrické detektory založeny na základě vnitřního fotoelektrického jevu a oproti klasickým polovodičovým diodám jsou upraveny tak, aby na PN přechod dopadalo světlo [15]. Při dopadu fotonu vhodné vlnové délky na PN přechod dochází k předání energie elektronu, který je excitován a získá dostatek energie pro přeskočení do vodivostního pásu [16]. Vzniká tak nosič náboje elektron-díra a pokud je celkový náboj dostatečně silný, dojde k překonání napětí hradlové vrstvy a diodou začne protékat elektrický proud. Toto minimální potřebné napětí U_p se nazývá prahové napětí [15]. Naprázdno v tzv. fotovoltaickém režimu je dioda zapojena v propustném směru a funguje jako zdroj elektrické energie [16]. Pro měření změny intenzity záření dopadajícího na PN přechod se pak dioda obvykle zapojuje ve směru závěrném v tzv. fotovodivostním nebo odporovém režimu, kdy reaguje na změny osvětlení velmi rychle [15, 16].

Pokud jsou příměsi na elektrodách nerovnoměrně rozděleny, dochází také k nerovnoměrnému rozložení elektrické intenzity uvnitř diody. Toto vnitřní elektrické pole pak funguje podobně jako fotonásobič a urychluje vzniklé nosiče náboje elektron-díra [15]. Při vysokém závěrném napětí pak dioda vykazuje mnohonásobně vyšší citlivost, než klasická fotodioda. Fotodiodu v tomto zapojení nazýváme lavinová fotodioda (APD z anglického avalanche photodiode) [16].

Pro detekci jednotlivých fotonů se používá APD v tzv. Gegigerově režimu (G-APD), kdy je přiváděné závěrné napětí U_a nad hranicí průrazného napětí diody U_b [17, 18]. V tomto režimu je APD schopna registrovat i jednotlivé fotony, které způsobí průraz fotodiody a obvodem tak začne protékat elektrický proud [17]. Tyto fotodiody jsou pak s jistou účinností schopné zaregistrovat, zda na PN přechod foton skutečně dopadl a doprovodná elektronika tuto skutečnost vyšle dál ve formě TTL pulsů [19]. Kvantová účinnost fotodiod je závislá na vlnové délce dopadajícího záření a v našem případě činila zhruba 60 %.

Po zaregistrování fotonu musí být závěrné napětí diody okamžitě sníženo na nebo pod úroveň U_b, aby nedošlo ke zničení diody (tzv. „hašení“). Tuto dobu hašení označujeme jako mrtvou dobu detektoru a v našem případě činila t = 50 ns [19]. APD během této doby není schopné detekovat další fotony. Následně dochází k opětovnému zvýšení napětí nad úroveň průrazného napětí diody. Rozlišujeme pasivní hašení a aktivní hašení [18].

Pasivní hašení je způsob zapojení obvodu, kdy přidaná zátěž v podobě resistoru s vysokým odporem způsobí v případě průrazu prudký pokles závěrného napětí přiváděného na fotodiodu. Nově vzniklé nosiče náboje pak již nemají takovou energii a lavinový jev se postupně zastaví [18]. Pasivní hašení chrání fotodiodu před zničením za každých okolností, ale oproti aktivnímu hašení vykazuje vyšší mrtvou dobu detektoru.

Z důvodů vysoké mrtvé doby detektoru bylo pasivní hašení postupně nahrazeno hašením aktivním. V případě aktivního hašení je fotodioda připojená na dodatečný elektronický obvod, který dokáže rozpoznat začínající lavinový jev a na tuto skutečnost rychle reagovat snížením závěrného napětí v obvodu mírně pod hodnotu U_b [20]. Aktivní hašení podstatně snižuje mrtvou dobu detektoru a umožňuje G-APD pracovat s vyšší opakovací frekvencí [18].

1.4 Temné detekce

Popsali jsme si, jak fungují jednofotonové detektory při absorpci světelného záření. Co se ale stane, pokud APD v Geigerově režimu zapneme „naprázdno“, tj. izolovaný od dopadajících fotonů? V ideálním případě by byl počet detekcí za jednotku času nulový, nicméně se ukazuje, že v takto citlivých zařízení ve skutečnosti neustále dochází k dalším detekcím, jež nemají s detekcí fotonů nic společného. Nazýváme je temné detekce (DC z anglického Dark counts) [19] a z hlediska statistiky jde o chyby prvního druhu (falešně pozitivní) [21].

Jak počet, tak i samotná existence temných detekcí je závislá na jiných zdrojích energie, než je světelné záření. Hlavními faktory zde jsou ionizující záření jako například kosmické záření (neutrina, miony, zbytkové radioaktivní záření) a také termální vibrace, jelikož G-APD nepracuje za teploty absolutní nuly. Tyto termální vibrace pak mohou zapříčinit přechod jednoho z elektronů do vodivostního pásu, spuštění lavinového efektu a následně falešné detekce fotonu. Počet temných detekcí je tedy silně závislý na teplotě zařízení [17] a pro potlačení šumu jsou lavinové fotodiody často chlazeny Peltierovým článkem. Jako další faktor ovlivňující počet temných detekcí uveďme kvalitu a čistotu materiálu, ze kterého je fotodioda vyrobena.

Tyto podmínky (koeficienty) tedy silně ovlivňují počet temných detekcí za jednotku času pro každou G-APD [19]. Jelikož jsou ale elektrony subatomární částice a platí pro ně tedy principy kvantové mechaniky, tak ani v případě zajištění konstantních podmínek by podle těchto principů neměl být počet temných detekcí za jednotku času deterministický [8]. Existuje vždy jen určitá pravděpodobnost, že dojde k excitaci elektronu a temná detekce se započítá. Tuto pravděpodobnost, ačkoliv se s časem mění, je možné odhadnout, nicméně temná detekce samotná (a stejně tak celkový počet temných detekcí za jednotku času) se ale nedá s určitostí předpovědět pro konkrétní časový úsek. Je tudíž náhodná a mohla by docela dobře posloužit jako základ našeho generátoru náhodných čísel. Tuto premisu jsme se pokusili experimentálně ověřit.

1.5 Extrahování bitů z neuniformní náhodné sekvence

V experimentální praxi je měření náhodných jevů velmi ovlivněno vybraným způsobem a parametry měření. Často pak dochází k tomu, že měření byť i principiálně náhodného jevu vygeneruje silně neuniformní (tedy předvídatelnou) sekvenci. Tato odchylka od nepředvídatelnosti přitom není způsobena nenáhodností jevu, jen nevhodným způsobem sběru dat, ve kterém nejsou všechny události stejně pravděpodobné. Jednoduchým příkladem může být měření, zda temná detekce nastala nebo nenastala v konkrétním časovém úseku (podkapitola 2.3). Pokud

bude toto časové okno příliš velké, budou temné detekce zaznamenány takřka při každé iteraci. Pokud je příliš malé, nebudou zaznamenány téměř žádné temné detekce. Tuto neuniformnost je obvykle možno odstranit kalibrací měřící aparatury, v našem případě zvolením vhodného časového intervalu, ovšem tento interval je pro každou fotodiodu jiný a navíc nemusí být v čase konstantní.

Pro nastolení uniformnosti v sekvenci je proto vhodnější využít počítačových algoritmů, které dokáží náhodné bity z neuniformní binární náhodné sekvence vyextrahovat [22]. Těchto algoritmů je hned několik a liší se zejména svou účinností extrakce a dobou zpracování. Podmínkou fungování těchto algoritmů je, aby sekvence byla náhodná, jednotlivé bity byly na sobě nezávislé a aby pravděpodobnost obou bitů byla v čase konstantní [23].

Nejjednodušší algoritmus extrahující bity z binárních sekvencí byl navržen von Neumannem [22]. Jeho myšlenka tkví v rozdělení sekvence na dvojice po sobě následujících bitů. Následně pokud máme náhodnou binární sekvenci s pravděpodobností bitu 0 označenou jako p a pravděpodobnost bitu 1 jako q, je výskyt dvou po sobě jdoucích událostí pq stejně pravděpodobný, jako dvojice událostí qp (tedy 01 a 10 má v sekvenci stejnou pravděpodobnost k výskytu). Tyto dvojice událostí tedy můžeme nahradit bity 0 a 1 a vytvořit tak námi hledanou uniformní sekvenci. Stejně ovšem nemůžeme postupovat u po sobě jdoucích událostech pp a qq, jejichž pravděpodobnosti obecně neznáme. Tyto bity tedy musíme zahodit.

Jednoduchost tohoto algoritmu umožňuje velice rychlé zpracování dat, jeho nevýhodou je ovšem účinnost extrakce. Pro $p = q = 0.5$ (tedy pro dokonale uniformní sekvenci) je šance na dvojici událostí $pq = 0{,}25$, tedy 25 %. Celková šance k uskutečnění námi použitelných událostí je tedy v naprosto dokonalém případě $pq + qp = 0{,}5 = 50$ %. Jejich nahrazením bity 0 a 1 ale ztratíme další polovinu dat. Maximální účinnost extrakce von Neumannova algoritmu je tedy pouhých 25 % délky původní sekvence [22]. Účinnosti pro neuniformní sekvence jsou pak ještě podstatně nižší.

Uveďme si ilustrační příklad: Naše výchozí neuniformní sekvence byla vygenerována událostmi „*qqpqqqppqp*“, (jde tedy o binární sekvenci „1101110010“). Aplikací von Neumannova algoritmu dostaneme sekvenci „$\times 0 \times \times 1$“ přičemž $\times$ reprezentuje dvojice bitů, které musíme zahodit. Výsledná sekvence je tedy „01“, jež je skutečně uniformní, ovšem za cenu ztráty 4/5 dat. Znázornění můžeme vidět v tabulce 1.2.

Tabulka 1.2: Příklad použití von Neumannova algoritmu.

von Neumann						
11	01	11	00	10	...	výchozí sekvence
×		×	×		...	odhozené bity
	0			1	...	extrahovaná sekvence

1.5.1 Peresův algoritmus

Existují ovšem i další algoritmy extrahující bity z neuniformní náhodné sekvence, které dosahují daleko vyšší účinnosti extrakce. Právě z tohoto důvodů jsme se v naší práci rozhodli použít algoritmus Peresův [23]. Tento algoritmus je jakousi iterací von Neumannova algoritmu, a umožňuje extrahovat ze sekvence maximální míru Shannonovy entropie H, která je popsána v sekci 1.6. Fungování Peresova

algoritmu (jednu jeho iteraci) si ukážeme v tabulce 1.3 na stejném příkladě jako výše.

Tabulka 1.3: Příklad použití Peresova algoritmu.

iterace Perese						
11	01	11	00	10	...	výchozí sekvence
	0			1	...	extrahovaná sekvence
0	1	0	0	1	...	sekvence x
1		1	0		...	sekvence y

Sekvence x představuje XORování a sekvence y jakýsi doplňkový von Neumannův algoritmus“. Tato část algoritmu tedy hledá v sekvenci dvojice stejných po sobě jdoucích bitů, přičemž dvojici 00 nahradí bitem 0 a dvojici 11 nahradí bitem 1. Sekvence x i sekvence y jsou pak dále každá zvlášť iterovány Peresovým algoritmem jako nové vstupní sekvence. Tímto opakovaným postupem bude konečná extrahovaná sekvence našeho příkladu „0101“. Velikost výsledné uniformní sekvence je tedy oproti použití von Neumannova algoritmu (v tomto případě) dvojnásobná. Naši implementaci Peresova algoritmu můžeme vidět ve zdrojovém kódu 1.1.

```
1 output = ""
2 count = 0
3 def split_string(string):
4     return string[0:len(string):2], string[1:len(string):2]
5 def xor_strings(xs, ys):
6     return "".join(bin(int(x,2)^int(y,2))[2:] for x,y in zip(xs,ys))
7 def peres(inp_str):
8     global output
9     global count
10    if (len(inp_str) % 2 == 1):
11        inp_str = inp_str[0:len(inp_str)-1]
12    if (len(inp_str) >= 2):
13        stringis = ''.join((inp_str))
14        string1,string2 = split_string(stringis)
15        xor_str = xor_strings(string1,string2)
16        neum_str = ""
17        antineum_str = ""
18        for item in range(0, len(string2)):
19            if (int(string1[item]) == 0 and int(string2[item]) == 0):
20                antineum_str = antineum_str + str(0)
21                continue
22            if (int(string1[item]) == 1 and int(string2[item]) == 1):
23                antineum_str = antineum_str + str(1)
24                continue
25            if (int(string1[item]) == 1 and int(string2[item]) == 0):
26                neum_str = neum_str + str(0)
27                continue
28            if (int(string1[item]) == 0 and int(string2[item]) == 1):
29                neum_str = neum_str + str(1)
30                continue
31        output = output + neum_str
32        if(len(xor_str) >= 2):
33            peres(xor_str)
34        if(len(antineum_str) >= 2):
35            peres(antineum_str)
```

Zdrojový kód 1.1: Peresův algoritmus psaný v jazyce Python.

1.6 Shannonova entropie

Entropie S zjednodušeně vyjadřuje, v jakém stavu se systém nachází, a obvykle je definována jako míra neurčitosti systému. Čím je systém náhodnější a neuspořádanější, tím je hodnota entropie vyšší. Obvykle se s entropií setkáváme v termodynamice, kde je popisována termodynamickými zákony [24].

V informatice pak existuje tzv. Shannonova entropie H, která určuje míru neurčitosti (náhodnosti) sekvence znaků v binární zprávě. Sekvence naprosto náhodných čísel by pak měla mít maximální míru entropie. Shannonovu entropii náhodné veličiny je obecně možno vypočítat podle vzorce (1.3) [25]

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 (p_i), \tag{1.3}$$

kde X je naše náhodná veličina nabývající N možných hodnot s pravděpodobnostmi p_i, kde i nabývá hodnot v intervalu $<0;N>$.

Shannonova entropie také vyjadřuje počet bitů potřebných k zakódování řetězce symbolů v závislosti na frekvenci jednotlivých symbolů [25]. Teoretickou hodnotu entropie vstupní i výstupní binární sekvence $\boldsymbol{X}$ je pak možno spočítat vztahem (1.4):

$$H(\boldsymbol{X}) = -N[p \log_2 (p) + (1-p) \log_2 (1-p)], \tag{1.4}$$

kde N je počet bitů v řetězci, p je pravděpodobnost bitu „1“ (tedy počet jedniček dělený počtem bitů), a $(1-p)$ je pak pravděpodobnost bitu „0“ (případně obráceně). Pro dokonale uniformní sekvenci pak platí vztah (1.5)

$$H(\boldsymbol{X}_{p=0.5}) = N, \tag{1.5}$$

tedy hodnota Shannonovy entropie je rovna počtu bitů sekvence.

Výpočet entropie dle rovnice (1.4) nám tedy nejen určí míru náhodnosti dat, ale můžeme také entropii námi naměřených dat porovnat s výstupními daty Peresova algoritmu a určit tak jeho účinnost vztahem (1.6):

$$\eta_e = \frac{H_O}{H_I} \approx \frac{N_O}{H_I}, \tag{1.6}$$

kde H_I je hodnota vstupní Shannonovy informace, H_O je hodnota výstupní Shannonovy informace spočítané dle rovnice (1.4) a N_O je délka sekvence po aplikaci Peresova algoritmu, který zajistí, že se pravděpodobnosti bitu 0 a 1 v sekvenci rovnají. Tato účinnost by nikdy neměla přesáhnout 100%, ale pro $N \to \infty$ by se jí měla limitně blížit [23].

1.7 Polarizace světla

Elektromagnetické záření tvoří dvě na sebe kolmé vektorové složky. Jde o vektor elektrické intenzity $\boldsymbol{E}$ a vektor magnetické intenzity $\boldsymbol{H}$ [24]. Vektor elektrické intenzity $\boldsymbol{E}$ je přitom vždy kolmý na směr šíření energie vlnění, směr vektoru magnetické intenzity $\boldsymbol{H}$ je potom kolmý jak na směr šíření vlnění, tak na směr vektoru $\boldsymbol{E}$. V izotropním prostředí jak pak směr šíření energie totožný se směrem šíření vlnění. V průběhu šíření elektromagnetické vlny prostorem mění oba

dva vektory svou velikost i směr, ale vždy tvoří rovinu kolmou na směr šíření světla [24]. Tato vlastnost světla se nazývá polarizace a je definovaná trajektorií kmitů vektoru $\boldsymbol{E}$ v určitém bodě. Tedy polarizační stav světla můžeme definovat zkoumáním trajektorie kmitů vektoru $\boldsymbol{E}$. Tato trajektorie může být chaotická, poté mluvíme o tzv. nepolarizovaném světle. Pokud však vykazuje určitou perioditicu, tedy opakuje svůj tvar, je dané světlo polarizované [24]. Speciálním případem polarizovaného světla je pak polarizace lineární. Trajektorie vektoru $\boldsymbol{E}$ v tomto případě odpovídá přímce [16].

Lidské oko nedokáže změny v polarizaci zpozorovat, nicméně je na polarizaci závislá celá řada vlastností záření, jako například index lomu, absorpce a rozptyl záření nebo odraz na rozhraní různých prostředí [16]. Polarizace je možno docílit několika způsoby, a to odrazem, lomem, nebo dvojlomem. Objekty se schopností polarizovat světlo pak obecně nazýváme polarizátory [24]. Polarizaci můžeme zapsat také matematicky, například podle tzv. Jonesova vektoru, jehož obecný tvar pro záření procházející osou $\mathbf{z}$ můžeme vyjádřit pomocí vztahu (1.7) [16]

$$\boldsymbol{E} = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} A_x \exp(i\phi_x) \\ A_y \exp(i\phi_y) \end{bmatrix}, \tag{1.7}$$

kde A_x značí (reálnou) amplitudu a ϕ_x úhel mezi rovinou polarizace a námi zvolenou složkou báze $\mathbf{x}$ (Analogicky pak pro E_y). Tento vztah můžeme také zapsat jako lineární kombinaci bázových vektorů, konkrétně pro lineární polarizaci:

$$\boldsymbol{E} = E_x\mathbf{i} + E_y\mathbf{j} = A_x \begin{bmatrix} \exp(i\phi_x) \\ 0 \end{bmatrix} + A_y \begin{bmatrix} 0 \\ \exp(i\phi_y) \end{bmatrix}. \tag{1.8}$$

kde první člen ze součtu ve vztahu (1.8) představuje světlo polarizované zcela horizontálně, druhý člen pak světlo polarizované zcela vertikálně. Pro lineární polarizaci pod úhlem $\phi_x = \phi_y = \phi = 45°$ (tzv. diagonální polarizace) jsou pak oba vektory E_x a E_y stejně zastoupeny. Po dosazení za $A_x = A_y = A = 1$ pak spolu s normalizační podmínkou $\boldsymbol{E}^2 = 1$ můžeme diagonálně polarizované světlo definovat vztahem (1.9)

$$\boldsymbol{E} = \begin{bmatrix} A\exp(i\phi) \\ A\exp(i\phi) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{1.9}$$

1.7.1 Polarizační děliče

Polarizační dělič obecně dělí fotony optického svazku dle jejich polarizace. Jako možný polarizační dělič můžeme využít polarizační hranol, který je tvořen anizotropním materiálem vykazující různý index lomu (a tedy i různou rychlost šíření světla) pro lineárně polarizované vlnění. Skrz úhlopříčku hranolu je vsazen materiál, který pro jednu složku polarizace způsobuje totální odraz. Druhé složce polarizace ale odpovídá totální odraz při jiném úhlu dopadu a takto polarizované záření pak v drtivé většině projde skrz materiál [16].

Pokud chceme získat co možná nejvyváženější poměr prošlých a odražených fotonů, potřebujeme laserový svazek vedený do polarizačního děliče polarizovat diagonálně. Dosažení přesné lineárně diagonální polarizace v praxi ale není možné. Pro hrubé ovlivnění polarizaci fotonů dopadajících na polarizační dělič se používá polarizační kontroler, pro jemné ladění pak půlvlnná (případně čtvrtvlnná) polarizační destička. Tímto laděním jsme schopni docílit takřka úplné lineárně diagonální polarizace a přiblížit tak poměr odražených a prošlých fotonů na 1:1.

Alternativně je možné fotonový svazek rozdělit pomocí polopropustného zrcátka. Na tomto principu funguje celá řada interferometrů [24], nicméně stejně jako polarizační hranoly ani polopropustná zrcátka nedokáží zaručit rovnoměrné rozložení prošlých a odražených fotonů. Pro ovlivnění odrazivosti (a tedy i propustnosti) polopropustných zrcátek je pak nutné měnit úhel dopadu fotonů na zrcátko a tedy měnit i experimentální uspořádání, což může být vzhledem k přesnosti, s jakou je potřeba světelné svazky navázat do navazovačů, poněkud nepraktické.

Kapitola 2

Experimentální část a výsledky

2.1 Popis experimentu a použitá rozhraní

Jak již bylo zmíněno dříve, cílem našeho experimentu bylo popsat a otestovat několik optických jevů využívajících lavinových fotodiod ke generování náhodných čísel.

Pro měření událostí jsme použili lavinové fotodiody značky Perkin Elmer operující v Geigerově režimu. Tyto fotodiody jsou natolik citlivé, že jsou schopné zaregistrovat dopad i jednotlivých fotonů a v takovém případě vysílají elektronické TTL pulsy, které zaznamenával elektronický systém Ortec. Ten sčítal počet pulsů na počítadle a zároveň komunikoval s počítačem prostřednictvím sériového, případně paralelního portu. Pro řízení samotné komunikace mezi počítačem a elektronikou jsme použili vlastní programy implementované v jazyce C a Python.

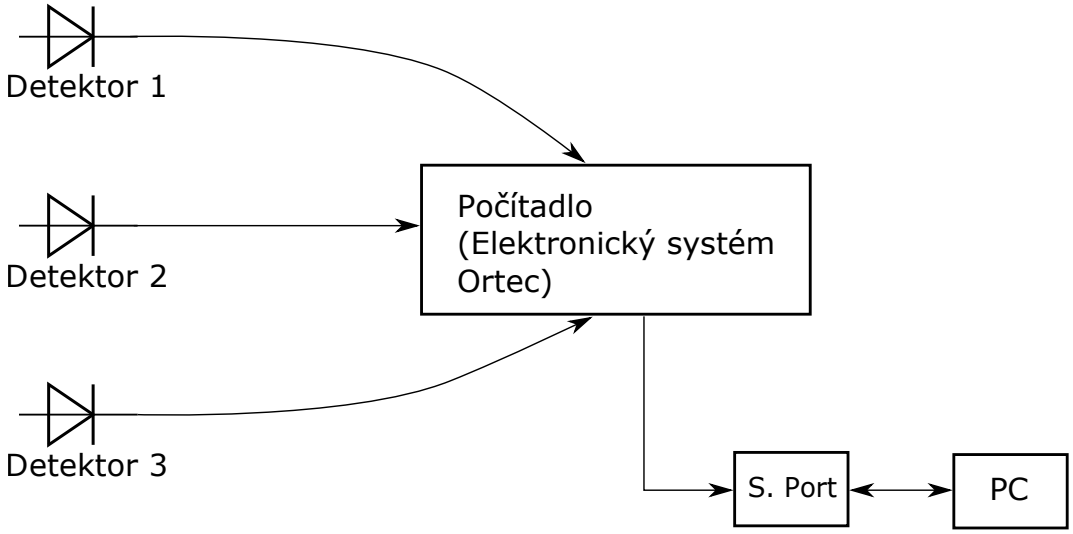
Výsledná data musela být vždy upravena na binární řetězec, aby mohla být otestována pomocí NIST Statistical test suite. Veškeré algoritmy na úpravu dat byly taktéž implementované v jazyce Python. Pro tvorbu grafů a histogramů byl použit program Graphics Layout Engine a grafické rozhraní Pyplot z balíku Matplotlib.

2.2 Generace náhodných dat měřením počtu temných detekcí

Způsobů sběru dat pomocí temných detekcí je více. Tím nejjednodušším je zapnout lavinový jedno-fotonový detektor v Geigerově režimu naprázdno, izolovaný od veškerého dopadajícího světla. Pokud fotodioda zaregistruje temnou detekci, vyšle TTL impuls, který přijme elektronický systém Ortec. Tyto impulsy sčítá na počítadle po stanovenou dobu a výslednou hodnotu posílá na sériový port, který komunikuje s počítačem pomocí programu vytvořeném v jazyce Python. Schéma experimentální sestavy můžeme vidět na obrázku 2.1.

Program pro komunikaci se sériovým portem pracoval v nekonečném cyklu, kdy při každé iteraci zaznamenal počet temných detekcí vždy za 0,1 sekundy a následně byl počet resetován na 0. Výsledné hodnoty jsou ukládány do datového souboru. Opakovací perioda 0,1 sekundy byla nejrychlejší, jakou počítadlo Ortec umožňuje. Při této opakovací periodě generoval náš sběr dat přibližně 7,5 hodnot za sekundu (každá hodnota představuje součet zjištěných detekcí za 0,1

sekund). Rychlosti 10 hodnot za sekundu jsme nedocílili, neboť určitá část času byla využita pro režii na komunikaci systému Ortec s počítačem.

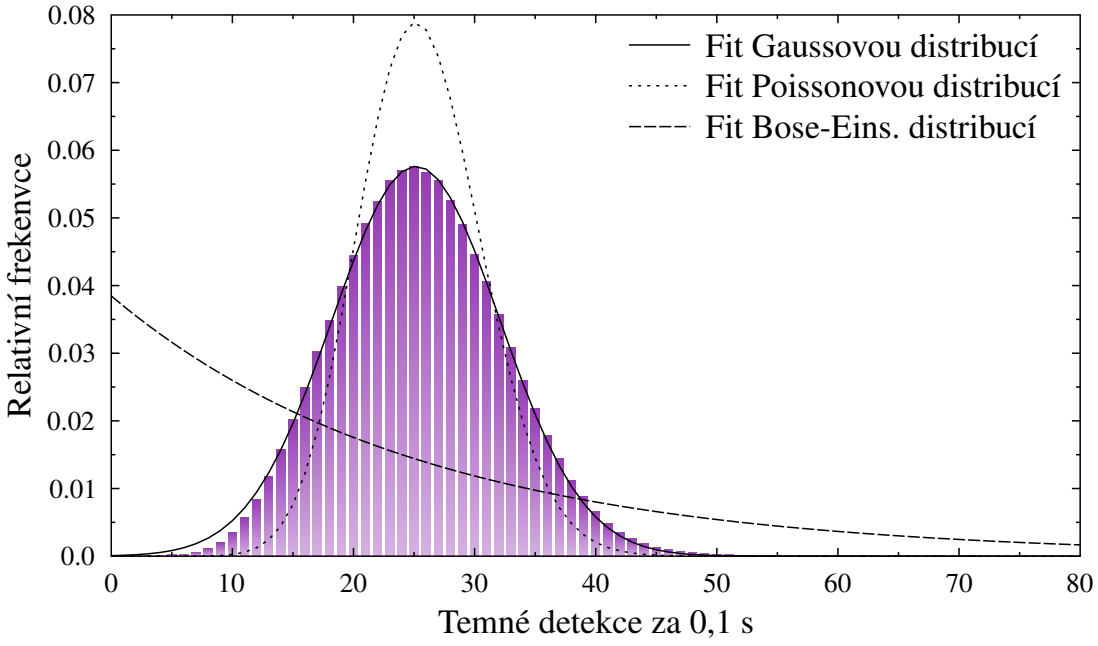


Obrázek 2.1: Schéma experimentální sestavy pro měření počtu temných detekcí dle kapitoly 2.2.

Výsledná data jsme použili k vykreslení histogramu. Následně jsme data fitovali Poissonovou distribucí, Gaussovou distribucí a Bose-Einsteinovou distribucí. Jeden z výsledných histogramů je možné vidět na obrázku 2.2 a ukazuje, že se histogram našich detekcí nejvíce podobá Gaussovské distribuci.

Neexistují přesná kritéria, která by testovací sada NIST pro spolehlivé ověření generátoru náhodných čísel požadovala [3]. Jako minimální doporučená velikost jedné sekvence je alespoň 20 000 bitů, což se odkazuje na dokument FIPS 140-1, který byl ale stažen v roce 2002 a nová kritéria nebyla přesně stanovena [26]. Podobně pak není určen přesný počet sekvencí, na které má být výchozí řetězec rozdělen. Z hlediska spolehlivosti statistického rozložení histogramu P-hodnot je doporučeno alespoň 55 sekvencí [3], spolu s původními 20 000 bity tedy 1,1 milionu bitů. V manuálu NIST STS je ovšem zmíněno, že spolehlivost všech testů roste s délkou sekvencí i jejich počtem [3]. Výsledky testů pro krátké sekvence pak nemusí být vždy spolehlivé a určité testy nebudou provedeny vůbec. Hodnoty použitých parametrů odpovídají výchozímu nastavení a doporučení NIST STS.

Na výsledná data jsme aplikovali několik algoritmů, díky kterým jsme původně Gaussovsky distribuovaná data převedli na uniformně distribuovaná binární data se stejnou pravděpodobností 0 a 1. Tato data pak mohla být otestována pomocí NIST STS. Testovali jsme data ze tří detektorů. V tabulce 2.1 můžeme vidět velikost naměřených dat a průměrnou frekvenci temných detekcí. V těchto případech jsme neměli dostatek dat pro všechny testy sady NIST STS. Některé z těchto testů tedy nemusely fungovat spolehlivě, nebo nebyly vůbec vyhodnoceny [3].



Obrázek 2.2: Histogram námi naměřených dat detektoru 1 společně s fity Gaussovské, Poissonovy a Bose-Einsteinovy distribuce.

Tabulka 2.1: Souhrn velikosti vzorků dat a frekvencí temných detekcí použitých lavinových fotodiod operujících v Geigerově režimu.

	Detektor 1	**Detektor 2**	**Detektor 3**
Množství naměřených dat	2,5 Mil. hodnot	5,2 Mil. hodnot	5,2 Mil. hodnot
Frekvence temných detekcí	254,5 Hz	891,7 Hz	139,5 Hz

Následuje výčet algoritmů, které jsme na data aplikovali. Je třeba poznamenat, že vzhledem k malé velikosti některých vzorků se výsledky testů pro rozdílné parametry mohly lišit. Výsledky testů, které nesplnily požadované kritéria NIST STS, jsou vždy označeny hvězdičkou.

2.2.1 Sudá/Lichá

Jelikož počet temných detekci za jednotku času by měl být náhodný, stejně tak by měl být náhodný i počet sudých a lichých čísel, pokud je histogram dat dostatečně široký. Pro větší množství dat by se oba počty měly přibližně rovnat. Jedním z možných způsobů, jak vytvořit požadovaný binární řetězec, byla tedy jednoduchá náhrada sudých čísel znakem 0 a lichých čísel znakem 1.

Tento jednoduchý algoritmus se ukázal být velmi úspěšným, jak můžeme vidět z výsledků v tabulce 2.2. Pro sekvenci rozdělenou do 100×50000 znaků splnil řetězec všechny testy sady NIST mimo několik subtestů nepřekrývajících se šablon (N-OT z anglického Non-overlapping Templates) a několika testů (univerzální, náhodná procházka a variace náhodné procházky), na jejichž splnění je třeba

větší délka sekvencí. Při upravení vstupních parametrů na například 101 × 50-100 pak byly splněny podmínky všech testů (opět vyjma univerzálního, náhodné procházky a variace náhodné procházky). Tyto odlišnosti pravděpodobně vznikají z důvodu malé velikosti vzorku dat.

Tabulka 2.2: Výsledky statistického testu NIST STS po úpravě algoritmem sudá/lichá. Velikost každé sekvence pro detektor 1 je 20 100 bitů, pro detektory 2 a 3 je to 50 100 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Uspěl

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	99/101	0,94	100/101	0,97	98/101	0,63
Bloková Frekvence	199/101	0,71	100/101	0,34	101/101	0,05
Kumulativní Suma	99/101	0,61	99/101	0,95	98/101	0,71
Běhy	97/101	0,23	100/101	0,83	98/101	0,27
Nejdelší Běh	98/101	0,41	101/101	0,38	101/101	0,33
Hodnost matice	99/101	0,01	101/101	0,01	101/101	0,001
FFT	98/101	0,13	100/101	0,02	100/101	0,59
Nepřekr. se šablony	99/101	0,49	99/101	0,44	99/101	0,51
Překr. se šablony	100/101	0,63	100/101	0,02	101/101	0,41
Univerzální	-	-	-	-	-	-
Entropie	98/101	0,11	96/101	0,007	100/101	0,12
Náhodná procházka	-	-	-	-	-	-
Var. náhodné proch.	-	-	-	-	-	-
Sériový	100/101	0,67	101/101	0,10	99/101	0,17
Lineární komplexity	100/101	0,38	101/101	0,33	100/101	0,88

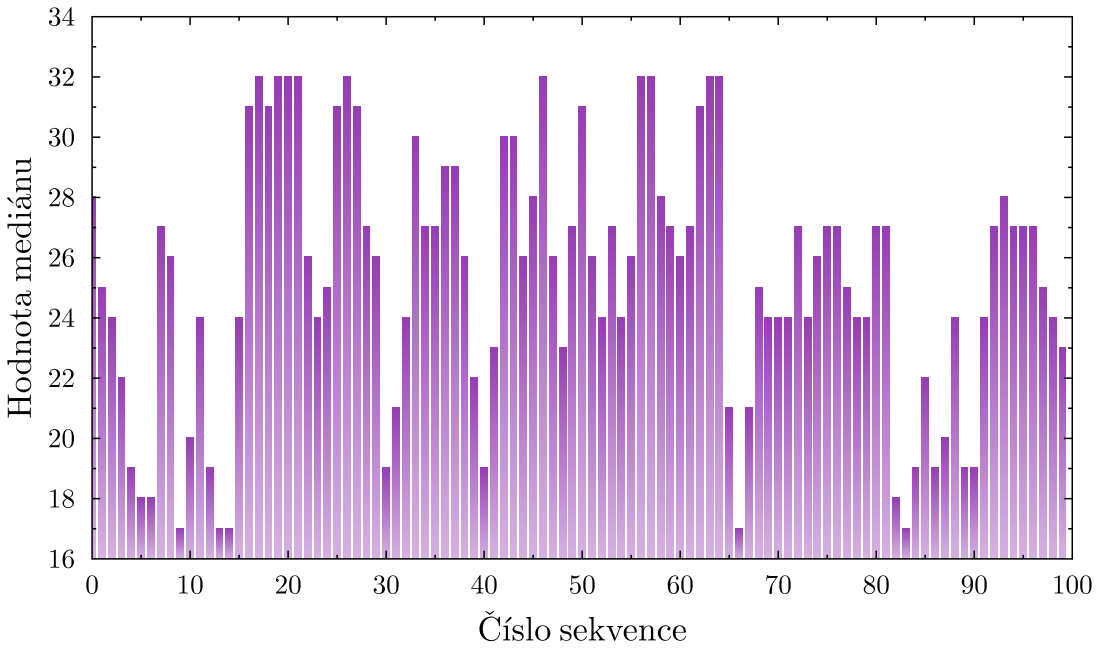
2.2.2 Medián

Medián, jak ze statistiky víme, je hodnota, která dělí data na přesnou polovinu, přičemž platí, že 50% hodnot je větších než medián a 50% hodnot je menších než medián [8]. Medián samotný tedy leží uprostřed. Toho jsme se pokusili využít pro algoritmus, který všechny hodnoty menší než hodnota mediánu nahradí 0 a větší než medián 1. Pro zachování uniformnosti pak samotná čísla odpovídající mediánu musí být zahozena.

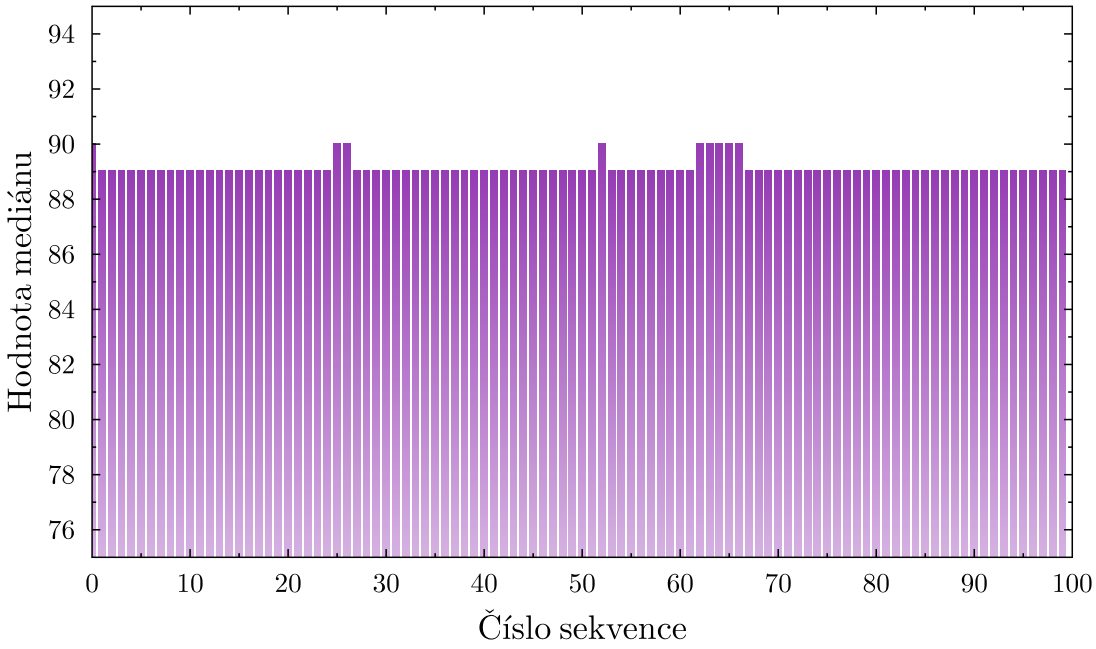
Takto vygenerovaný řetězec testovací sadou vůbec neprošel. Jako hlavní problém zde pravděpodobně byla asymetrická a diskrétní povaha distribuce dat. Kvůli tomuto problému pak nebyla výsledná data uniformní, ale vždy jeden z bitů převažoval, a proto musela být upravena algoritmem, který počty vyrovnával.

Dalším problémem pak byla fluktuace hodnoty mediánu v průběhu času, kterou ilustruje obrázek 2.3. Obrázek nastiňuje mediány pro každou setinu měřené sekvence, přičemž jde jasně vidět, že se medián detekcí v čase neustále měnil. Pro tuto skutečnost jsme nenalezli uspokojivé vysvětlení, jelikož laboratorní podmínky (s výjimkou vlhkosti vzduchu a případných odchylek dopadajícího ionizujícího záření) byly konstantní. Předpokládáme tedy, že hlavní příčinou bylo

opotřebení fotodiody. Pro jiné fotodiody byla fluktuace mediánů mnohem méně dramatická, jak můžeme vidět na obrázku 2.4.



Obrázek 2.3: Fluktuace mediánů temných detekcí lavinové fotodiody 1.



Obrázek 2.4: Fluktuace mediánů temných detekcí lavinové fotodiody 2.

Výsledky statistického testu dat pak můžeme vidět v tabulce 2.3. V těchto testech se projevila i fluktuace hodnoty mediánu, proto můžeme například vidět lepší výsledky pro detektory 2 a 3 oproti detektoru 1, u kterého byla fluktuace nejvíce pozorovatelná.

Tabulka 2.3: Výsledky frekvenčního testu NIST STS po upravení dat pomocí algoritmu Medián. Velikost každé sekvence pro Detektor 1 je 20 000 bitů, pro Detektory 2 a 3 50 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Selhal

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	3*/100	0*	34*/100	0*	6*/100	0*

Fluktuaci mediánu jsme se pokusili vyřešit tak, že jsme algoritmus pro medián i algoritmus vyrovnávající počet 0 a 1 zavolali na každou tísícinu dat zvlášť a zohlednili tak jednotlivé mediány. Výsledná data měla daleko lépe vyvážený počet obou bitů a i v testu NIST STS se osvědčila lépe, ale stále nesplňovala požadovaná statistická kritéria. Problémový byl zejména histogram P-hodnot, jak jde vidět na výsledných P-hodnotách v tabulce 2.4.

Tabulka 2.4: Výsledky frekvenčního a blokově frekvenčního testu NIST STS po vyvážení poměru 0 a 1 z dat upravených dle algoritmu Medián. Velikost každé sekvence pro detektor 1 je 19 000 bitů, pro detektory 2,3 je to 40 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Selhal

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	99/100	0*	98/100	0,002	99/101	0,002
Bloková Frekvence	41*/100	0*	69*/100	0*	69*/101	0*

Jak můžeme vidět, naše implementace vyřešila neúspěch frekvenčního testu a vyrovnalo počty 0 a 1, ovšem periodické zásahy do rozložení sekvence potlačily jeho přirozenou fluktuaci a vytvořily tak opakující se vzory, což se projevilo na neúspěchu například blokově frekvenčního testu.

2.2.3 Binární přepis

Naměřený počet temných detekcí byl zapsán v desítkové soustavě, kterou běžně používáme. Pomocí programu není problém tyto hodnoty přepsat do dvojkové soustavy, kde jsou tvořeny pouze sledem 0 a 1. Tato data pak můžeme zřetězit za sebe a otestovat, ačkoliv kvůli opakujícím se motivům by neměla statistickým testem projít. Na druhou stranu pak značně navýší počet získaných dat a ověří schopnost testovací sady detekovat opakující se sekvence.

Vygenerovaný řetězec podle očekávání vůbec nesplnil kritéria NIST testů, jak lze vidět v tabulce 2.5.

Tabulka 2.5: Výsledky frekvenčního testu NIST STS po přepsání naměřených hodnot do dvojkové soustavy. Velikost každé sekvence pro všechny tři detektory je 50 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Selhal

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	0*/100	0*	0*/100	0*	0*/100	0*

2.2.4 XOR

XOR či XORování (nebo také exkluzivní disjunkce) je bitová operace, která vezme vždy dva znaky z binárního řetězce (případně dvou binárních řetězců), a porovná je – pokud jsou oba znaky 0 nebo 1, zapíše jako výsledek 0, pokud se liší, zapíše jako výsledek 1. Původní data jsou pak zahozena a porovnávají se další dva znaky. V našem případě jsme použili výše zmíněný binárně přepsaný řetězec rozdělený v půli, a následně testovali vždy dva znaky od začátku obou řetězců. Počet dat se zmenší o polovinu, ovšem data se „znáhodní“.

Vygenerovaný řetězec testy NIST nesplnil. Data se po jednom XORování neukázala být dostatečně náhodná. Výsledek frekvenčního testu ukazuje tabulka 2.6.

Tabulka 2.6: Výsledky frekvenčního testu NIST STS aplikaci XORování na data přepsaná do dvojkové soustavy. Velikost každé sekvence pro všechny detektory je 50 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Selhal

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	17*/100	0*	46*/100	0*	0*/100	0*

2.2.5 XOR (2x)

Přístup algoritmu byl identický jako u toho předešlého, ale nově vzniklý řetězec byl XORován ještě jednou. Data tak byla opět více „znáhodněna“, na úkor délky řetězce, který klesl na čtvrtinu binárního přepisu.

Testovaný řetězec splnil kritéria testů velmi dobře, jak můžeme vidět v tabulce 2.7. V závislosti na parametrech se však podobně jako při využití algoritmu sudá/lichá objevily jeden nebo dva N-OT subtesty, jejichž kritéria řetězec nesplňoval. Například při rozdělení sekvence na 101×61000 byly opět splněny všechny testy, vyjma testu univerzálního.

Tabulka 2.7: Výsledky statistického testu NIST STS po aplikaci dvojitého XORování na data přepsaná do dvojkové soustavy. Velikost každé sekvence pro detektor 1 je 25 000 bitů, pro detektor 2 61 000 bitů a pro detektor 3 je to 51 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Uspěl

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	99/100	0,006	98/101	0,75	101/101	0,04
Bloková Frekvence	98/100	0,47	96/101	0,02	101/101	0,69
Kumulativní Suma	99/100	0,21	97/101	0,66	101/101	0,40
Běhy	99/100	0,40	101/101	0,73	101/101	0,63
Nejdelší Běh	99/100	0,19	98/101	0,17	100/101	0,36
Hodnost matice	100/100	0,09	100/101	0,38	99/101	0,47
FFT	99/100	0,03	101/101	0,65	100/101	0,23
Nepřekr. se šablony	98/100	0,49	99/101	0,48	99/101	0,53
Překr. se šablony	99/100	0,19	100/101	0,48	101/101	0,13
Univerzální	-	-	-	-	-	-
Entropie	98/100	0,02	101/101	0,71	100/101	0,15
Náhodná procházka	-	-	-	-	-	-
Var. náhodné proch.	-	-	-	-	-	-
Sériový	98/100	0,93	101/101	0,09	100/101	0,39
Lineární komplexity	96/100	0,92	101/101	0,93	98/101	0,48

2.2.6 XOR (3x)

Trojitým XORem jsme se pokusili ještě navýšit náhodnost dat a odstranit tak jednotlivé nesplněné testy, které se při určitých parametrech objevily u některých dvakrát XORovaných dat. Cenou byla pouhá osmina výstupních dat oproti datům vstupním.

Trojitý XOR se ukázal být jako zbytečný, jelikož výsledný řetězec splňoval testy velmi podobně až hůře, s větším počtem nesplněných testů. Také se opět prokázalo, že malá velikost souboru dat snižuje spolehlivost NIST Statistical test suite. Výsledky testů můžeme najít v tabulce 2.8, kde je vidět, že test bloková frekvence pro detektor 2 nesplnil požadovaná kritéria.

Tabulka 2.8: Výsledky statistického testu NIST STS po aplikaci trojitého XORování na data přepsaná do dvojkové soustavy. Velikost každé sekvence pro detektor 2 je 42 000 bitů a pro detektor 3 28 000 bitů. Množství dat vygenerované detektorem 1 nebylo pro trojité XORování dostatečně velké. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Selhal

Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	-	-	100/101	0,77	98/101	0,81
Bloková Frekvence	-	-	95*/101	0,23	99/101	0,13
Kumulativní Suma	-	-	99/101	0,35	99/101	0,40
Běhy	-	-	100/101	0,03	100/101	0,93
Nejdelší Běh	-	-	101/101	0,003	101/101	0,63
Hodnost matice	-	-	99/101	0,05	100/101	0,06
FFT	-	-	99/101	0,28	100/101	0,20
Nepřekr. se šablony	-	-	99/101	0,45	99/101	0,44
Překr. se šablony	-	-	101/101	0,22	99/101	0,13
Univerzální	-	-	-	-	-	-
Entropie	-	-	97/101	0,01	96/101	0,001
Náhodná procházka	-	-	-	-	-	-
Var. náhodné proch.	-	-	-	-	-	-
Sériový	-	-	99/101	0,36	99/101	0,88
Lineární komplexity	-	-	100/101	0,01	99/101	0,0005

2.2.7 Inverzní Box-Mullerova transformace

Jak jsme již zmínili, rozdělení našich dat se blíží Gaussovské distribuci. Existují také algoritmy, které slouží k převedení uniformní distribuce na distribuci Gaussovskou. Jeden z těchto algoritmů se nazývá Box-Müllerova transformace [27] a my jsme implementovali a využili její inverzní verzi, která by naše data s Gaussovským rozdělením převedla na data rozložená uniformně.

Vezměme dvě náhodná čísla U_1 a U_2, která jsou nezávislá uniformně distribuovaná na intervalu (0, 1) a dvě náhodná čísla Z_1 a Z_2, která jsou nezávislá gaussovsky distribuovaná náhodná čísla [27]. Pak

$$Z_1 = R\cos(\theta) = \sqrt{-2\log U_1}\cos(2\pi U_2) \tag{2.1}$$

a

$$Z_2 = R\sin(\theta) = \sqrt{-2\log U_1}\sin(2\pi U_2), \tag{2.2}$$

kde

$$R^2 = -2\log U_1 \tag{2.3}$$

a

$$\theta = 2\pi U_2. \tag{2.4}$$

Vztahy pro inverzní transformaci můžeme odvodit například podělením rovnice (2.2) rovnicí (2.1), pro θ pak platí:

$$\frac{Z_2}{Z_1} = \frac{\sin(\theta)}{\cos(\theta)} = \tan(\theta) \Rightarrow \theta = \arctan\left(\frac{Z_2}{Z_1}\right). \tag{2.5}$$

Umocněním a sečtením rovnic (2.1) a (2.2) dostáváme:

$$Z_1^2 + Z_2^2 = R^2\left(\cos^2(\theta) + \sin^2(\theta)\right) \Rightarrow R^2 = Z_1^2 + Z_2^2. \tag{2.6}$$

Dosazením rovnice (2.5) do (2.4) a rovnice (2.6) do (2.3) pak pro U_1 a U_2 můžeme psát:

$$U_1 = \exp^{-\frac{R^2}{2}} = \exp^{-\frac{Z_1^2+Z_2^2}{2}}, \tag{2.7}$$

$$U_2 = \frac{\theta}{2\pi} = \frac{\arctan\left(\frac{Z_2}{Z_1}\right)}{2\pi}. \tag{2.8}$$

Naši implementaci inverzní Box-Müllerovy transformace psané v jazyce Python můžeme vidět ve zdrojovém kódu 2.1.

```
1 stred = float(median)
2 sigma = float(std)
3 # Sirka vysledneho uniformniho histogramu
4 n = 4;
5
6 U1m = []
7 U2m = []
8 Z1m = []
9 Z2m = []
10
11 datas = data[0:len(data):2]
12 datal = data[1:len(data):2]
13 # Vytvoreni Gaussovske distribuce
14 for i in range (0,int(len(datas))):
15     Z1 = float(datas[i])
16     Z2 = float(datal[i])
17 # Normalizace
18     Z1 = float((g1-stred)/sigma)
19     Z2 = float((g2-stred)/sigma)
20
21     Z1m.append(g1)
22     Z2m.append(g2)
23 # Vypocet inverzni Box-Mullerovy transformace
24     r2 = numpy.power(g1,2)+numpy.power(g2,2)
25     if (g1 == 0):
26         t = numpy.pi/2
27     else:
28         t = numpy.arctan(g2/g1)
29
30     u1 = numpy.exp(-r2/2)*numpy.power(2,n)
31     u2 = t/2/numpy.pi
32     u2 = (u2+0.25)*2*numpy.power(2,n)
33
34     U1m.append(u1)
35     U2m.append(u2)
```

Zdrojový kód 2.1: Inverzní Box-Müllerova transformace psaná v jazyce Python

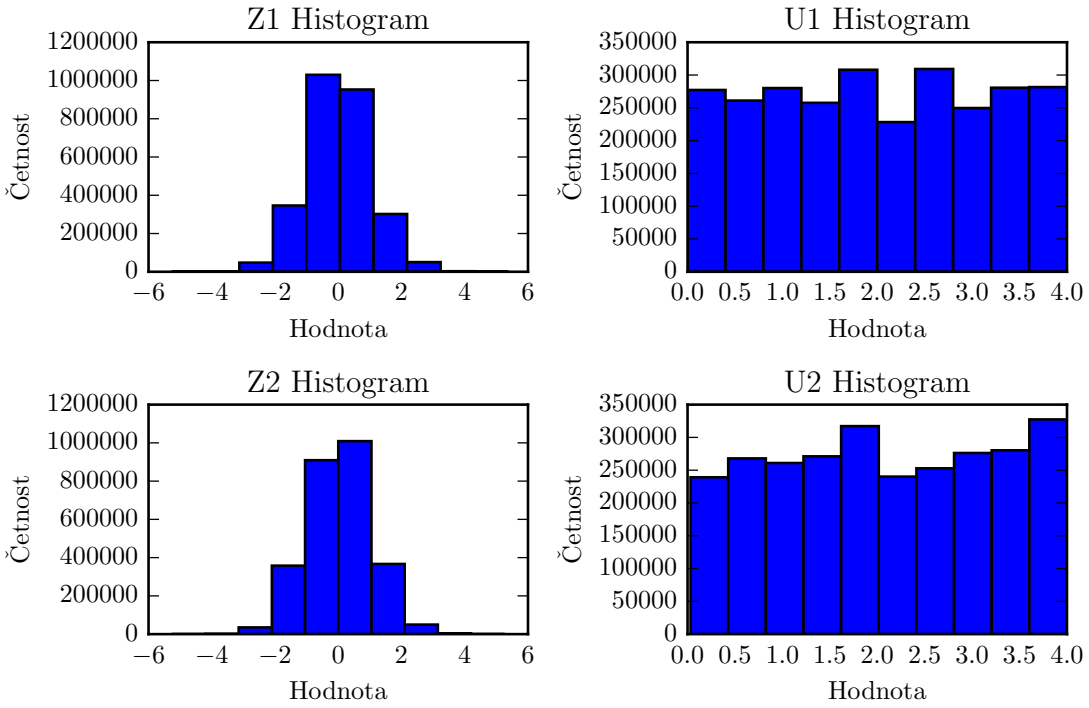
Data vygenerovaná algoritmem neodpovídala požadovanému uniformnímu rozdělení. Uniformnost distribuce můžeme určit pomocí relativní směrodatné odchylky σ, kterou vypočítáme podle rovnice (2.9).

$$\sigma_r = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}}{\overline{x}} \cdot 100, \tag{2.9}$$

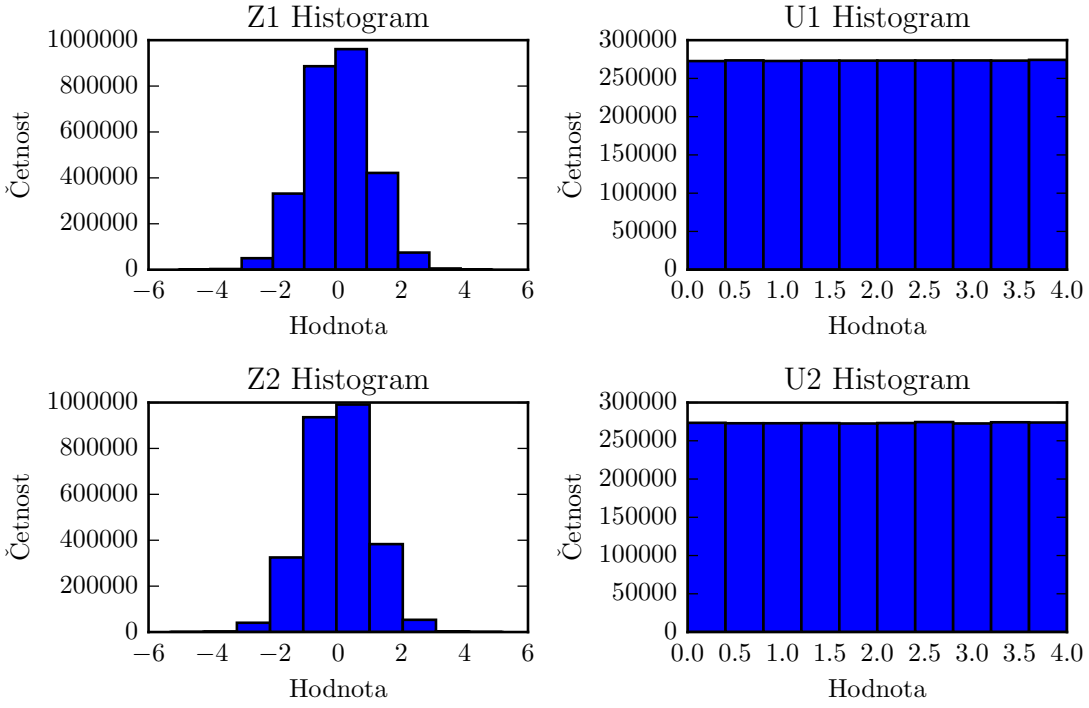
kde n je celkový počet prvků distribuce, x_i je hodnota každého prvku a $\overline{x}$ je střední hodnota distribuce.

Výslednou relativní směrodatnou odchylku jsme určili na $\sigma_{U_1} = 8{,}7$ % a $\sigma_{U_2} = 10{,}2$ % pro detektor 2 a $\sigma_{U_1} = 25{,}7$ % a $\sigma_{U_2} = 26{,}4$ % pro detektor 3. Pro kontrolu algoritmu jsme využili možnosti programovacího jazyku Python a pomocí balíku Numpy vygenerovali odpovídající počet pseudonáhodných dat s Gaussovskou distribucí se stejnou střední hodnotou a variancí, jako měla naše naměřená data. Tato umělá data již uniformnost splňovala, relativní směrodatná odchylka dosahovala maximálně 0,25 %.

Důvodem selhání algoritmu je pravděpodobně fakt, že distribuce našich dat není přesně Gaussovská. NIST STS testy nebyly provedeny vzhledem ke špatné uniformnosti a nevyrovnaném počtu 0 a 1 po přepsání dat do dvojkové soustavy. Uniformnost výsledných dat, která algoritmus vytvořil z námi naměřených dat a z vygenerovaných pseudonáhodných dat s Gaussovskou distribucí, můžeme porovnat na obrázcích 2.5 a 2.6. Na obrázcích můžeme vidět histogramy vstupních Gaussovských dat Z_1 a Z_2 a histogramy výstupních uniformních dat U_1 a U_2.



Obrázek 2.5: Histogram dat Z_1 a Z_2 naměřených detektorem 2 a výsledné uniformní distribuce U_1 a U_2 s relativní směrodatnou odchylkou $\sigma_{U_1} = 8{,}7$ % a $\sigma_{U_2} = 10{,}2$ % po použití inverzní Box-Müllerovy transformace.



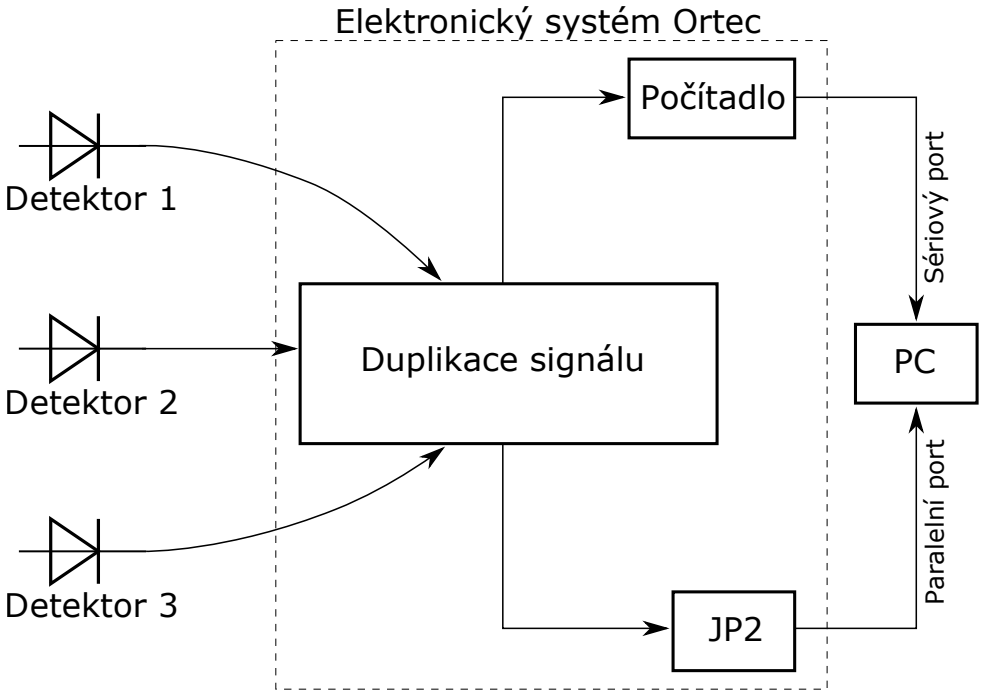
Obrázek 2.6: Histogram vygenerovaných Gaussovských pseudonáhodných dat Z_1 a Z_2 vygenerovaných počítačem a výsledné uniformní distribuce U_1 a U_2 s relativní směrodatnou odchylkou $\sigma_{U_1} = 0{,}21$ % a $\sigma_{U_2} = 0{,}17$ % po použití inverzní Box-Müllerovy transformace.

2.3 Generace náhodných dat měřením sledu temných detekcí

Získávaní dat pomocí počtu temných detekcí za jednotku času vykazuje jednoznačnou nevýhodu – sběr dat je velmi pomalý a hodnoty se pro potřeby NIST STS musí dále upravovat. Pokusili jsme se tedy detektory zapojit tak, aby generovaly přímo binární řetězec.

Detektory v tomto případě mohou mít opakovací periodu mnohokrát vyšší než při využití počítadla, v našem případě jsme takto dosáhli opakovací periody 0,1 milisekund. Pokud v tomto časovém intervalu dioda zaznamená temnou detekci, vyšle TTL puls do klopného obvodu JP2 vyrobeném ve Společné laboratoři optiky. Schéma experimentální sestavy můžeme vidět na obrázku 2.7

Klopný obvod je elektronický obvod, který nabývá právě dva stabilní napěťové stavy – polohu 0 a polohu 1. Přepínám těchto stavů je možné uchovávat informaci. Pokud klopný obvod zaregistruje TTL puls, přepne se do polohy 1 a v té setrvává, dokud není uživatelem resetován. Skrze paralelní port je pak pomocí počítače možné zjistit, v jaké poloze se klopný obvod nachází a případně ho resetovat. Klopný obvod JP2 má čtyři vstupy, čímž umožňuje vést signály až pro čtyři detektory najednou (my jsme současně využili tři).



Obrázek 2.7: Schéma experimentální sestavy pro měření sledu temných detekcí dle kapitoly 2.3.

Program na počítači byl psán v jazyce C a v nekonečném cyklu četl hodnoty napětí na paralelním portu, přičemž při každé iteraci resetoval klopný obvod do polohy 0 s opakovací periodou 0,1 milisekund. Dosáhli jsme tak rychlosti generaci přibližně 4,5 kbit za sekundu. Naši implementaci tohoto programu můžeme vidět ve zdrojovém kódu 2.2 Kopie TTL pulsů byla opět posílaná na počítadlo Ortec, pomocí kterého jsme zároveň zapisovali počet temných detekcí za jednotku času (v tomto případě 0,1 sekundy). Tato data jsme využili pro vykreslení histogramu.

```
1 #define base 0xa000
2 main(int argc, char **argv)
3 {
4   int cekani;
5   int values;
6   char vystup[4] = {'0','0','0','\0'};
7   int maska;
8   int x;
9   ioperm(base,2,1);
10  sscanf(argv[1],"%i",&cekani);
11
12  while (1){
13    values = inb(base+1);
14
15    maska = 64;
16    if ((maska & values) != 0) vystup[0] = '1';
17
18    maska = 128;
19    if ((maska & values) == 0) vystup[1] = '1';
20
21    maska = 32;
22    if ((maska & values) != 0) vystup[2] = '1';
23
24    fprintf(stderr,"%s\n",vystup);
25    for(x=0;x<3;x++) vystup[x] = '0';
26
27    outb(0, base);
28    usleep(0);
29    outb(27,base);
30    usleep(cekani); } }
```

Zdrojový kód 2.2: Zdrojový kód sběru dat lavinových fotodiod pomocí paralelního portu.

Tento způsob generace dat je mnohem rychlejší než využití počítadla. Čekací doba je zde omezena vlastně jen mrtvou dobou detektoru a rychlostí resetování klopného obvodu, který se po každém zaznamenaném napětí musí resetovat do výchozí hodnoty 0. Problémem ovšem je, že pokud každý detektor vykazuje jiný počet temných detekcí za jednotku času, počet 0 a 1 pro zvolenou opakovací periodu nemůže být nikdy v rovnováze pro všechny tři detektory zároveň. Hledat takovou opakovací periodu, při níž by pravděpodobnost 0 a 1 byla stejná, by i pro jeden detektor bylo značně nepraktické, proto jsme se rozhodli počet 0 a 1 vyvážit jinak.

Pro nastolení rovnováhy 0 a 1 v sekvenci náhodných čísel existuje několik algoritmů. Rozhodli jsme se využít Peresův algoritmus [23], který zachovává maximální hodnotu Shannonovy entropie vstupních dat [25]. Ten značně prodloužil dobu zpracování dat, obzvláště pro detektory s vysokým počtem temných detekcí (a tedy vyrovnanějším počtem 0 a 1), což mohlo být zapříčiněno využitím programovacího jazyka Python.

Je třeba zmínit, že pro Peresův algoritmus jsme při XORování nemohli řetězec rozdělit v půli a XORovat vždy dva znaky jako u předchozích dat – výsledný počet 0 a 1 nebyl dostatečně v rovnováze (lišil se téměř o jedno procento). Tento problém byl pravděpodobně způsoben fluktuacemi 0 a 1 napříč celým souborem dat. Problém jsme vyřešili odlišnou generací podřetězců - jeden podřetězec obsahoval všechny členy se sudým indexem a druhý podřetězec všechny členy s lichým indexem z původního řetězce stejně, jako to ve svém článku navrhl sám Peres [23].

Naměřená data prošla statistickým testováním bez chyby a potvrdila tak myšlenku náhodnosti temných detekcí pro všechny tři detektory, jak můžeme vidět z výsledků v tabulce 2.9. V tomto případě jsme měli dostatek dat pro všechny statistické testy.

Tabulka 2.9: Výsledky statistického testu NIST STS po aplikaci Peresova algoritmu na data získaná měřením sledu temných detekcí. Velikost každé sekvence pro detektor 1 je 1,02 Mbit, pro detektor 2 je to 1,7 Mbit a pro detektor 3 je to 6 Mbit. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testu je 96, pro testy náhodné procházky a variace náhodné procházky je to: 65 pro detektor 1, 60 pro detektor 2 a 80 pro detektor 3. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Uspěl						
Test	**Detektor 1**		**Detektor 2**		**Detektor 3**	
	Poměr	P-hod	Poměr	P-hod	Poměr	P-hod
Frekvence	99/100	0,62	100/100	0,57	98/100	0,92
Bloková Frekvence	99/100	0,78	100/100	0,66	98/100	0,78
Kumulativní Suma	98/100	0,39	99/100	0,29	98/100	0,6
Běhy	100/100	0,97	99/100	0,86	100/100	0,55
Nejdelší Běh	100/100	0,19	99/100	0,46	97/100	0,05
Hodnost matice	100/100	0,64	99/100	0,96	99/100	0,9
FFT	98/100	0,55	98/100	0,80	100/100	0,02
Nepřekr. se šablony	98/100	0,51	98/100	0,50	99/100	0,54
Překr. se šablony	99/100	0,19	100/100	0,08	99/100	0,22
Univerzální	98/100	0,22	100/100	0,002	100/100	0,9
Entropie	100/100	0,28	100/100	0,85	100/100	0,06
Náhodná procházka	68/69	0,21	63/64	0,39	83/84	0,43
Var. náhodné proch.	68/69	0,29	63/64	0,37	82/84	0,44
Sériový	99/100	0,20	99/100	0,70	99/100	0,46
Lineární komplexity	98/100	0,55	98/100	0,21	98/100	0,15

Již jsme zmínili, že doba zpracování Peresovým algoritmem rostla spolu se zvyšující se uniformností dat. Stejně tak ale rostla i výtěžnost extrahování dat, jak můžeme vidět na hodnotách tabulky 2.10, kde je nejvíce dat vyextrahováno právě pro detektor s největší střední hodnotou frekvence temných detekcí. Účinnost extrakce (myšleno jako účinnost extrahování „náhodnosti“ sekvence, nikoliv množství dat) η_e můžeme vypočítat podle vztahu (1.6).

Tabulka kromě množství vstupních a výstupních dat porovnává také účinnost extrakce η_e. Můžeme vidět, že η_e dosahuje nejméně 99,3 %, a tedy že Peresův algoritmus zachoval 99,3 % náhodnosti původní sekvence. Také můžeme vidět, že účinnost extrakce spolu s množstvím výsledných dat roste.

Tabulka 2.10: Porovnání vstupního a výstupního mmnožství dat v závislosti na střední hodnotě frekvence temných detekcí a odpovídající účinnosti extrakce Peresova algoritmu pro všechny tři detektory.

Vstupní hodnoty

	Detektor 1	**Detektor 2**	**Detektor 3**
Množství naměřených dat	1,43 Gbit	1,43 Gbit	1,43 Gbit
Frekvence temných detekcí	107,9 Hz	55,5 Hz	627,5 Hz
Pravděpodobnost bitu 1	1,68 %	0,87 %	9,44 %

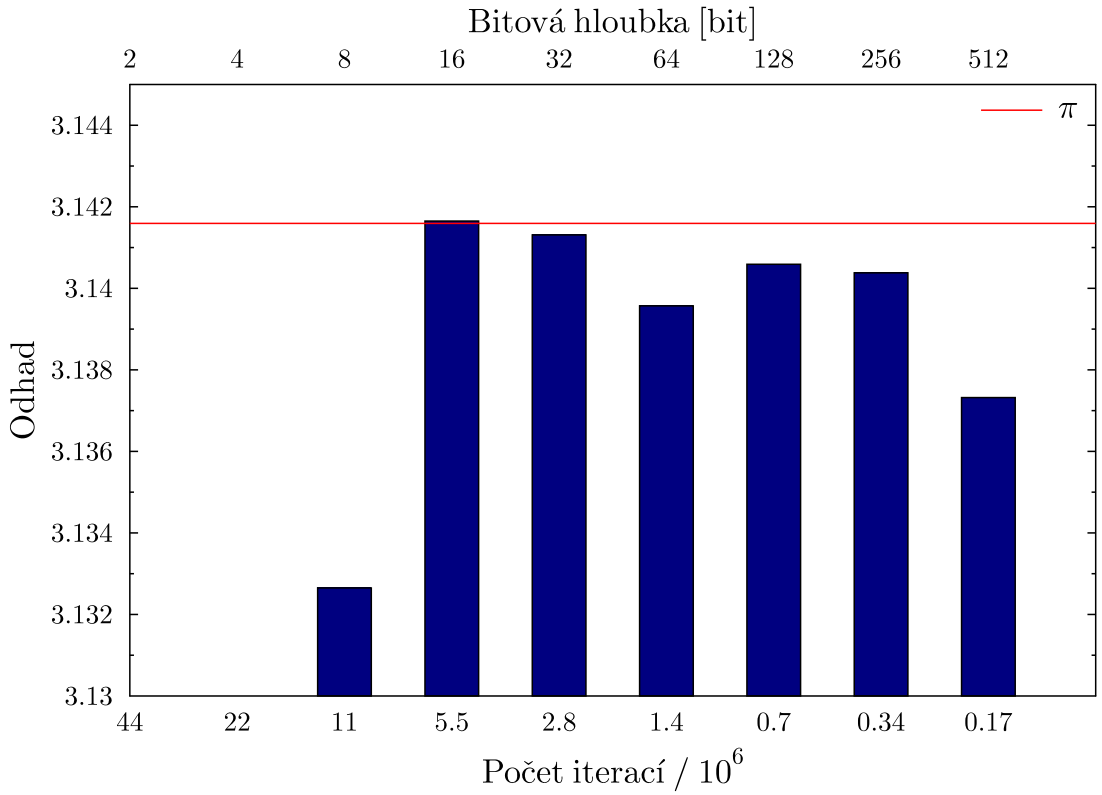
Výstupní hodnoty

	Detektor 1	**Detektor 2**	**Detektor 3**
Množství extrahovaných dat	175 Mbit	102 Mbit	641 Mbit
Účinnost extrakce	99,44 %	99,30 %	99,53 %

Dále můžeme vygenerovaná náhodná čísla otestovat také odhadnutím velikosti čísla π pomocí jednoduché simulace Monte Carlo, jejíž výsledky můžeme vidět v tabulce 2.11. Tato simulace potřebuje ke správnému odhadování obecně velké množství dat, proto jsme jí nevyužili v ostatních experimentech. Přesnost odhadu čísla π závisí na dvou faktorech. Za prvé je pro přesný odhad třeba simulaci mnohokrát iterovat, tedy čím vyšší počet iterací, tím přesnější odhad. Za druhé je pak třeba mít vstupní čísla dostatečně jemně rozlišená. Jelikož je naše vygenerovaná sekvence binární, a π potřebujeme vyjádřit v desítkové soustavě, musíme naše data převést.

Můžeme si ovšem zvolit bitovou hloubku každého převedeného čísla, tedy kolik bitů z binární sekvence bude tvořit jednotlivá převedená desítková čísla. Čím vyšší je tato bitová hloubka, tím větší je interval vygenerovaných desítkových čísel, a tím přesnější by také měl být i výsledný odhad. Cenou za větší bitovou hloubku i za vyšší počet iterací je ovšem i větší spotřeba dat. Můžeme si položit otázku, jaký je optimální poměr těchto dvou faktorů, pokud máme konstantní objem dat. Závislost přesnosti odhadnuté hodnoty π na zvyšující se bitové hloubce (a zároveň snižujícím se počtu iterací) od 2 bitů do 512 bitů při konstantní velikosti dat můžeme vidět na obrázku 2.8.

Z obrázku můžeme vyčíst, že nejméně efektivní jsou vždy krajní hodnoty, tedy pokud zvolíme bitovou hloubku příliš nízkou, nebo příliš vysokou. Výsledné odhady čísla π najdeme v tabulce 2.11. Můžeme vidět, že navzdory očekáváním nebyl nejpřesnější odhad pomocí dat vygenerovaných detektorem 3, u kterého byla využita největší bitová šířka i nejvyšší počet iterací. Odchylka od tabelované hodnoty π je nicméně velmi malá (řádově $1 \cdot 10^{-4}$), proto můžeme odhady považovat vzhledem k použité metodě za přesné.



Obrázek 2.8: Závislost přesnosti odhadnuté hodnoty π na rostoucí bitové hloubce a snižujícím se počtu iterací.

Tabulka 2.11: Výsledky odhadnutí čísla π pomocí simulace Monte Carlo.

Odhad čísla π

	Detektor 1	**Detektor 2**	**Detektor 3**
Bitová hloubka	16 bit	16 bit	64 bit
Počet iterací	$5{,}5{\cdot}10^6$	$3{,}2{\cdot}10^6$	$5{\cdot}10^6$
Odhad čísla π	3,14165	3,1407	3,1418
Chyba odhadu	$5{,}63{\cdot}10^{-5}$	$8.65{\cdot}10^{-4}$	$2{,}5{\cdot}10^{-4}$

2.4 Generace náhodných dat měřením polarizace jednotlivých fotonů

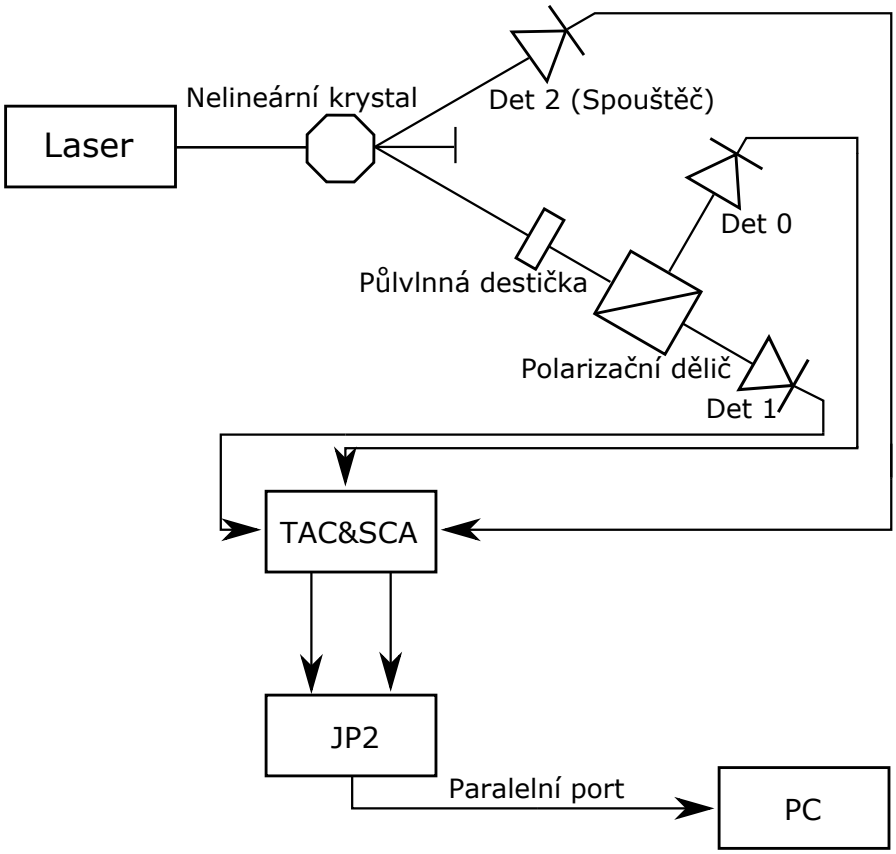
Hlavním účelem lavinové fotodiody v Geigerově režimu je měřit četnost jednotlivých fotonů. Proto jsme jako druhý experiment pro generování náhodných čísel využili vlastností samotného světelného záření, konkrétně jeho polarizace. Sestavili jsme optickou soustavu fungující jako polarizační dělič fotonů, který příchozí fotony dělí dle jejich polarizace do dvou navazovačů světelného svazku. Využili jsme laser Coherent Paladin Nd-YAG s integrovanou generací třetí harmonické o vlnové délce $\lambda = 355$ nm, opakovací frekvenci f = 120 MHz a výkonem zeslabeným na 200 mW. Laserový svazek procházel nelineárním optickým prostředím, a došlo k sestupné frekvenční parametrické konverzi, kdy se z laserového svazku generovaly dva fotony o $\lambda = 710$ nm, které jsme použili v našem experimentu.

Pro toto měření jsme využili tři lavinové fotodiody zapojené v tzv. koincidenčním módu, jehož zapojení můžeme vidět na obrázku 2.9. Detektor 2 je určený jako tzv. spouštěč (z anglického trigger), další dva detektory pak označme jako Detektor 1 a Detektor 0. První ze dvojice vzniklých fotonů navážeme pomocí optického vlákna do polarizačního děliče. Jelikož šíření světla optickým vláknem obecně mění jeho polarizaci v závislosti na ohnutí vlákna, nebyl na výstupu z vlákna foton polarizován diagonálně, jak bychom potřebovali. Této polarizace jsme docílili až laděním pomocí polarizačního kontroléru a půlvlnné destičky natočené pod úhlem přibližně 22,5°. Takto polarizované fotony dále procházejí polarizačním děličem a dělí se přibližně ve stejném poměru – polovina fotonů je navázána do optického kabelu a vedena do detektoru 0, druhá polovina fotonů je vedena do detektoru 1. V případě detekce vysílají obě diody TTL puls dál do obvodu.

Druhý ze dvojice vzniklých fotonů je navázán přímo do spouštěče, tomuto fotonu říkáme herald. Tyto fotony se šíří po kratší optické dráze, než fotony vedené do polarizačního děliče. Detektor 2 pak při jejich detekování vysílá TTL puls do TAC (Time-to-amplitude converter) převodníku, na kterém dochází k lineárnímu zvyšování napětí až do chvíle, kdy přijme TTL puls z detektoru 0 nebo detektoru 1. V tu chvíli je růst napětí zastaven a konečná hodnota napětí je vyhodnocena jednokanálovým analyzátorem SCA (Single channel analyzer). Ten na základě vhodně zvoleného intervalu amplitudy napětí slouží k odfiltrování detekcí jiných, než těch způsobených detekcí fotonů ze stejného páru. Může jít například o temné detekce, zachycené zbytkové světlo, nebo detekcí dalšího fotonového páru vzniklého v laseru. V takovém případě konečná hodnota napětí nespadá do povoleného intervalu a detekce je zahozena.

Pokud je detekce určená jako platná (tedy způsobená pouze jedním fotonovým párem), je signál určen jako tzv. koincidence detekcí a poslán dál na klopný obvod JP2, který komunikuje s počítačem prostřednictvím paralelního portu. Signál (velikost napětí), který počítač přečte na paralelním portu vždy odpovídá bitům na jeho stavových pinech, kterých je celkem osm. V našem zapojení ovšem aktivně využíváme pouze dva piny odpovídající dvěma koincidencím detekcí (pokud detektor 0 a detektor 2 (případně detektor 1 a detektor 2) současně zaregistrují foton, změní se bit na příslušném pinu na 1, pokud ne, ponechává hodnotu 0), proto máme celkem 2^2 možných výsledných hodnot signálu. Jejich souhrn můžeme vidět v tabulce 2.12. Program na počítači byl psán v jazyce C a v nekonečném cyklu četl napětí na paralelním portu a zapisoval výsledná data. Následně je klopný obvod resetován a celý cyklus se opakuje. Dosažená rychlost

generace byla přibližně 450 bitů za sekundu. Implementaci programu můžeme vidět ve zdrojovém kódu 2.3.



Obrázek 2.9: Schéma experimentální sestavy pro měření polarizace jednotlivých fotonů dle podkapitoly 2.4.

```
1 #define base 0xa000
2 main(int argc, char **argv)
3 {
4   int value;
5   int values;
6   sscanf(argv[1],"%i",&value);
7   ioperm(base,1,1);
8   ioperm(base+1,1,1);
9   while (1){
10    values = inb(base+1);
11    fprintf(stdout,"%d\n",values);
12
13     if (values == 168){
14       fprintf(stdout,"%d\n",1);
15     fprintf(stderr,"%d",1);}
16
17     if (values == 152){
18     fprintf(stdout,"%d\n",0);
19       fprintf(stderr,"%d",0);}
20
21    outb(0, base);
22    usleep(10);
23      outb(255,base);
24    usleep(value);    }    }
```

Zdrojový kód 2.3: Zdrojový kód sběru dat polarizačního děliče pomocí paralelního portu.

Při hodnotě signálu A nebyla detekována žádná koincidence, což může být způsobeno nedokonalým navázáním fotonů do optického vlákna, nebo příliš krátkou opakovací periodou. Tuto hodnotu tedy zahazujeme. Signál B odpovídá koincidenci detektoru 2 s detektorem 0, signál C pak koincidenci detektoru 2 s detektorem 1. Tyto dva případy pak tvoří námi extrahovanou náhodnou sekvenci. Signál D nastává ve chvíli, kdy došlo ke koincidenci všech tří detektorů. Tento případ je také dán nedokonalostí měřící aparatury a důvodů může být více (nelineárním krystalem bylo generováno více, než jen dva fotony; v koincidenčním intervalu se objevila temná detekce aj.), tak nebo tak pro nás nemá relevantní význam a tuto hodnotu tedy také zahazujeme.

Tabulka 2.12: Možné výsledky měření polarizačního děliče.

A	B	C	D	... hodnoty signálu
00	10	01	11	... odpovídající bity
×	0	1	×	... extrahovaná sekvence

Jelikož vyvážení vertikální a horizontální polarizace pomocí půlvlnné destičky není zcela přesné, nebyla zcela uniformní ani naše výsledná sekvence. Opět jsme tedy využili Peresův algoritmus pro nastolení uniformnosti ve vygenerované sekvenci. Výsledky měření takto upravené sekvence byly velmi úspěšné a všechny testy byly splněny, jak můžeme vidět v tabulce 2.13. V tomto experimentu jsme testovali pouze jediné zařízení.

Tabulka 2.13: Výsledky statistického testu NIST STS po aplikaci Peresova algoritmu na data nasbíraná polarizačním děličem fotonů. Velikost každé sekvence je 750 000 bitů. Minimální počet úspěšně vyhodnocených sekvencí pro splnění testů je 96. Minimální P-hodnota úspěšně vyhodnoceného testu je 0,0001. Testy, jež kritéria NIST STS nesplnily, jsou označeny hvězdičkou.

Výsledek testu: Uspěl

Test	**Polarizační dělič**	
	Poměr	P-hod
Frekvence	100/100	0,03
Bloková Frekvence	98/100	0,92
Kumulativní Suma	100/100	0,04
Běhy	99/100	0,20
Nejdelší Běh	98/100	0,35
Hodnost matice	99/100	0,44
FFT	99/100	0,60
Nepřekr. se šablony	99/100	0,48
Překr. se šablony	99/100	0,19
Univerzální	100/100	0,92
Entropie	100/100	0,74
Náhodná procházka	60/61	0,58
Var. náhodné proch.	60/61	0,55
Sériový	100/100	0,75
Lineární komplexity	97/100	0,17

Rychlost sběru dat našeho polarizačního děliče dosahovala přibližně 450 bitů za sekundu a celkem jsme nasbírali 77,9 Mbit dat. Porovnání vstupního a výstupního množství dat po aplikace Peresova algoritmu spolu s účinností extrakce η_e můžeme vidět v tabulce 2.14. V tabulce můžeme také vidět odhad čísla π provedený pomocí simulace Monte Carlo podobně jako v předchozím experimentu.

Tabulka 2.14: Porovnání vstupního a výstupního mmnožství dat nasbíraných pomocí polarizačního děliče, odpovídající účinnost extrakce Peresova algoritmu a odhad čísla π pomocí simulace Monte Carlo.

Vstupní hodnoty	
	Polarizační dělič
Množství naměřených dat	77,9 Mbit
Pravděpodobnost bitu 1	46,82 %
Výstupní hodnoty	
	Polarizační dělič
Množství extrahovaných dat	77,3 Mbit
Účinnost extrakce	99,54 %
Odhad čísla π	
	Polarizační dělič
Bitová hloubka	32 bit
Počet iterací	$1{,}2{\cdot}10^{6}$
Odhad čísla π	3,1409
Chyba odhadu	$7{\cdot}10^{-4}$

Závěr

Cílem této práce bylo navrhnout, sestavit a otestovat úsporný kvantový generátor náhodných čísel využívající měření polarizačního dělení fotonů a temných detekcí lavinových fotodiod operujících v Geigerově režimu. V první kapitole byly zpracovány teoretické poznatky, které se využívají ke generaci náhodných čísel a jejich testování, a také některé principy a metody, jenž jsme při realizování generátoru využili.

Druhá kapitola popisovala tři experimenty zaměřené na generování binárních náhodných sekvencí spolu s jejich výsledky. Požadavkem experimentů bylo vygenerovat náhodnou binární sekvenci, kterou můžeme následně otestovat souborem statistických testů NIST STS.

První dva experimenty využívaly ke generaci náhodných čísel temné detekce lavinové fotodiody. Použili jsme fotodiody značky Perkins Elmer operující v Geigerově režimu a oba experimenty se lišily pouze zapojením experimentální sestavy a způsobem sběru dat. Prvním způsobem bylo jednoduché sčítání počtu temných detekcí za jednotku času pomocí elektronického počítadla Ortec. Maximální opakovací perioda, jakou počítadlo Ortec umožňuje, byla 0,1 sekundy. Rychlost generace v tomto experimentu proto byla velmi nízká, přibližně 7,5 hodnot za sekundu. Takto nasbíraná data byla v desítkové soustavě a měla přibližně Gaussovskou distribuci a bylo třeba je převést na uniformní binární sekvenci. Toho jsme dosáhli použitím několika algoritmů, z nichž nejúspěšnějším bylo několikanásobné xorování. Výsledky statistických testů potvrdily, že temné detekce lavinové fotodiody je možno využít jako efektivní zdroj náhodnosti, nicméně vzhledem k malému množství naměřených dat nemusely být tyto výsledky zcela spolehlivé.

Problémy s nízkou rychlostí detekce a převáděním čísel z desítkové do binární soustavy jsme se v druhém experimentu pokusili vyřešit alternativním zapojením lavinových fotodiod. V tomto případě jsme detekovali sled temných detekcí a pro komunikaci s počítačem jsme využili pouze klopný obvod a paralelní port. Vynecháním sčítací elektroniky jsme dosáhli opakovací periody 0,1 milisekund a značně tak zvýšili rychlost sběru dat, která činila 4,5 kbit/s pro každý detektor, přičemž jsme mohli akivně využívat až čtyři detektory současně. Je třeba zmínit, že rychlost této experimentální sestavy je omezena pouze rychlostí přepínání klopného obvodu, maximální možná rychlost generace dat by pro lavinové fotodiody s vyšším počtem temných detekcí (například fotodiody bez přídavného chlazení) byla ještě mnohokrát vyšší.

Takto generovaná data byla přímo binární, nicméně nebyla uniformně rozložená. Implementovali jsme proto Peresův algoritmus, který slouží k extrahování náhodnosti z binární neuniformně rozložené náhodné sekvence. Výhodou tohoto algoritmu je jeho účinnost – algoritmus zachovává maximální možnou míru Shannonovy entropie H, a tedy i maximální možnou „náhodnosti“ obsažené v sekvenci.

Nevýhodou je dlouhá doba zpracování dat, která ale mohla být částečně způsobena použitím programovacího jazyku Python.

Experiment se ukázal být velmi úspěšný a všechny požadavky statistického balíku NIST STS použitého k testování byly splněny. Účinnost extrakce η_e Shannonovy entropie H dosahovala více než 99,3 % a při použití námi vygenerovaných náhodných čísel pro odhad π pomocí simulace Monte Carlo jsme určili π s přesností na čtyři desetinná místa. Vzhledem k dosaženým výsledkům můžeme konstatovat, že námi navržený generátor náhodných čísel je plně funkční a splňuje požadavky na kryptografickou bezpečnost. Jeho další výhodou je také maximální úspora při zpracování generovaných dat.

Třetí experiment využíval pro generaci náhodných čísel měření polarizace jednotlivých fotonů vzniklých v procesu sestupné frekvenční parametrické konverze při průchodu laserového svazku nelineárním optickým prostředím. Na podobném principu bylo v minulosti postaveno několik generátorů [28, 29], ovšem žádný dle našich vědomostí nedosahoval úspornosti zpracování dat dosažené při použití Peresova algoritmu, který jsme na nasbíraná data opět aplikovali. Náhodná čísla vygenerovaná polarizačním děličem úspěšně splnila všechny test statistického balíku NIST STS, dosažená rychlost generace ale byla podstatně nižší než při měření temných detekcí, a to přibližně 450 bitů za sekundu. Účinnost extrakce Peresova algoritmu η_e pro námi sestrojený polarizační dělič dosahovala více než 99,5 %, a hodnotu π pomocí simulace Monte Carlo jsme určili s chybou $7{\cdot}10^{-4}$.

První experiment zejména potvrdil, že je temné detekce možno využít pro generování náhodných čísel. Dva zbývající experimenty pak představily funkční generátory náhodných čísel, které úspěšně splňují požadavky na generaci kryptograficky bezpečných náhodných čísel.

Literatura

[1] H. Fürst, H. Weier, S. Nauerth, D. G. Marangon, C. Kurtsiefer, and H. Weinfurter, "High speed optical quantum random number generation," *Opt. Express*, vol. 18, no. 12, pp. 13029–13037, 2010.

[2] M. Haahr, "Randomness," 1998. Available at `https://www.random.org/randomness/`.

[3] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. V. M Levenson, N. A. Heckert, and D. L. Banks, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, 2010.

[4] F. X. Wang, C. Wang, W. Chen, S. Wang, F. S. Lv, D. Y. He, Z. Q. Yin, H. W. Li, G. C. Guo, and Z. F. Han, "Robust quantum random number generator based on avalanche photodiodes," *Journal of Lightwave Technology*, vol. 33, no. 15, pp. 3319–3326, 2015.

[5] E. Carter, "The generation and application of random numbers," *Forth Dimensions*, vol. 16, 1994.

[6] S. K. Tawfeeq, "A random number generator based on single-photon avalanche photodiode dark counts," *Journal of Lightwave Technology*, vol. 27, no. 24, pp. 5665–5667, 2009.

[7] S. H. Kellert, *In the Wake of Chaos: Unpredictable Order in Dynamical Systems*. University of Chicago, 1994.

[8] D. J. Griffiths, *Introduction to Quantum Mechanics; 2nd ed.* Upper Saddle River, NJ: Pearson, 2005.

[9] H.-Q. Ma, Y. Xie, and L.-A. Wu, "Random number generation based on the time of arrival of single photons," *Appl. Opt.*, vol. 44, no. 36, pp. 7760–7763, 2005.

[10] C. H. Vincent, "The generation of truly random binary numbers," *Journal of Physics E Scientific Instruments*, vol. 3, pp. 594–598, 1970.

[11] L. Dorrendorf, Z. Gutterman, and B. Pinkas, "Cryptanalysis of the random number generator of the windows operating system," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 1, pp. 10:1–10:32, 2009.

[12] B. Sanguinetti, A. Martin, H. Zbinden, and N. Gisin, "Quantum random number generation on a mobile phone," *Phys. Rev. X*, vol. 4, p. 031056, 2014.

[13] M. Ren, E. Wu, Y. Liang, Y. Jian, G. Wu, and H. Zeng, "Quantum random-number generator based on a photon-number-resolving detector," *Phys. Rev. A*, vol. 83, p. 023820, 2011.

[14] X. Ma, X. Yuan, Z. Cao, B. Qi, and Z. Zhang, "Quantum random number generation," *npj Quantum Information*, vol. 2, p. 16021, 2016.

[15] J. Maťátko, *Elektronika*. SNTL, 1987.

[16] B. E. A. Saleh and M. C. Teich, *Fundamentals of photonics; 2nd ed.* Wiley series in pure and applied optics, New York, NY: Wiley, 2007.

[17] Perkin Elmer Company, *Avalanche photodiode User guide*, 2010.

[18] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa, "Avalanche photodiodes and quenching circuits for single-photon detection," *Appl. Opt.*, vol. 35, no. 12, pp. 1956–1976, 1996.

[19] Perkin Elmer Company, *SPCM-AQR*, 2005.

[20] M. Viterbini, S. Nozzoli, M. Poli, A. Adriani, F. Nozzoli, A. Ottaviano, and S. Ponzo, "Voltage breakdown follower avoids hard thermal constraints in a geiger mode avalanche photodiode," *Appl. Opt.*, vol. 35, no. 27, pp. 5345–5347, 1996.

[21] J. L. D. Roxy Peck, *Statistics: The Exploration & Analysis of Data*. Duxbury, 2011.

[22] J. von Neumann, "Various Techniques Used in Connection with Random Digits," *J. Res. Nat. Bur. Stand.*, vol. 12, pp. 36–38, 1951.

[23] Y. Peres, "Iterating von neumann's procedure for extracting random bits," *Ann. Statist.*, vol. 20, no. 1, pp. 590–597, 1992.

[24] D. Halliday, R. Resnick, and J. Walkers, *Fundamentals of physics; 6th ed.* New York, NY: Wiley, 2001.

[25] C. E. Shannon, "A mathematical theory of communication," vol. 27, pp. 379–423,623–656, 1948.

[26] NIST publication, "Security requirements for cryptographic modules," 1994.

[27] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 29, no. 2, pp. 610–611, 1958.

[28] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A fast and compact quantum random number generator," *Review of Scientific Instruments*, vol. 71, no. 4, pp. 1675–1680, 2000.

[29] A. Ivanova, V. Egorov, S. Chivilikhin, and A. Gleim, "Investigation of quantum random number generation based on space-time division of photons," *Nanosystems: Physics, Chemistry, Mathematics*, vol. 4, pp. 550–554, 2013.