Universidad Nacional de La Plata

Facultad de Ciencias Exactas

Departamento de Física

Detección y conteo de fotones en experimentos de óptica cuántica

Trabajo de Diploma para la Licenciatura en Física

Matías Rubén Bolaños

Dra. Lorena Rebón
Directora

Dr. Fabián A. Videla
Co-director

Asesor académico
Dr. Raúl Rossignoli

Abril 2021

Laboratorio de Fotónica Integrada - Centro de Investigaciones Ópticas (CIOp)
Información Cuántica y Sistemas Cuánticos de Muchos Cuerpos - Instituto de Física de La Plata (IFLP)

Índice de contenidos

Resumen

A lo largo de este Trabajo de Diploma, se estudiaron las bases fundacionales de la teoría de la información cuántica, y sus potenciales aplicaciones tecnológicas, como lo es la distribución cuántica de claves para encriptación incondicionalmente segura. En particular, se hizo énfasis en los conceptos teóricos de las implementaciones ópticas que utilizan fotones como portadores de la información, en las técnicas experimentales que nos permiten contar con fuentes de fotones individuales, y en el funcionamiento de los elementos ópticos con capacidad de manipular el estado cuántico de los mismos.

Como trabajo de investigación y desarrollo, se diseñó e implementó un circuito capaz de contar fotones detectados en coincidencia, denominado módulo contador de coincidencias, uno de los componentes fundamentales para experimentos y aplicaciones de información cuántica con fotones. Esto fue implementado en una placa de desarrollo de prototipos FPGA, diseñando además una interfaz gráfica capaz de intercambiar información con la FPGA. El mismo permite el conteo de cuentas individuales de cuatro entradas independientes, al mismo tiempo que es capaz de contar coincidencias múltiples de dos, tres y cuatro fotones, dentro de ventanas temporales variables.

Posteriormente, y motivado por su importancia en experimentos de información cuántica, se realizó un estudio de los diferentes tipos de generadores de números aleatorios (RNGs) y pseudoaleatorios, comparando las ventajas y desventajas de cada uno. A partir del conocimiento adquirido, se implementó un circuito capaz de generar números pseudoaleatorios en la misma placa FPGA.

Finalmente, se realizó un estudio de la estadística de foto-detección de distintos tipos de fuentes de luz, bajo las cuales se analizan resultados propios obtenidos experimentalmente. Así mismo, y utilizando el RNG desarrollado, se programó la placa FPGA para simular tres tipos de fuentes: láser, térmica y de fotones individuales (estados de Fock de un fotón). Con la intención de testear tanto el contador de coincidencias como el RNG implementados, se simuló, para cada una de las fuentes, el experimento de Hanbury-Brown y Twiss sin retardos, y se obtuvo la función de correlación de segundo orden $g^{(2)}(0)$ que da cuenta de la estadística de la fuente. Se obtuvieron los resultados esperados en el caso de la fuente láser y de la fuente de fotones individuales, pero debido la relación entre el tiempo de coherencia de la fuente y los tiempos de medida, no se observó diferencia apreciable entre la fuente térmica y la láser.

Capítulo 1

Introducción y motivaciones

La mecánica cuántica está entre las teorías más exitosas del siglo XX ya que ha podido explicar un gran número de fenómenos físicos y ha dado lugar tanto a nuevas interpretaciones como a avances y desarrollos tecnológicos. Desde que fuera establecida alrededor del año 1925, ayudó a entender, entre muchos otros, los espectros atómicos, procesos químicos, características y comportamiento de medios materiales, y el fenómeno de semi-conducción. El desarrollo de esta teoría también dio lugar al nacimiento de nuevas disciplinas, como lo fue la ciencia de la información cuántica. Esta disciplina se encarga de estudiar cómo ocurre el procesamiento de la información cuando la misma se codifica en el estado de un sistema que necesariamente debe ser descrito mediante una teoría cuántica, es decir, que no sigue las leyes de la mecánica clásica. Sus inicios datan de la década del 60, con el análisis de los efectos cuánticos en un sistema de comunicaciones ópticas [1]. Gracias al desarrollo de técnicas para controlar sistemas cuánticos individuales (electrones, fotones, átomos, etc.), en la década del 70 comienza lo que hoy se conoce como 'segunda revolución cuántica', y muchas de las propuestas que se veían como elucubraciones, comienzan a hacerse posibles mostrando un nuevo y enorme campo a explorar. A esto se debe básicamente el crecimiento que experimentó la ciencia de la información cuántica, que se volvió exponencial al ver sus potenciales aplicaciones tecnológicas en comunicación, criptografía, y computación.

Paralelamente, con la potencia de las computadoras personales creciendo exponencialmente cada unos pocos años [2], y el aumento en el porcentaje de la población mundial con acceso a Internet, la información pasó a cumplir un rol fundamental en nuestras vidas, donde cada día se intercambian cantidades inconmensurables de información de todo tipo. Para dar noción de la magnitud de esto, en el año 2020, con 4,57 billones de personas (aproximadamente el 60 % de la población mundial) con acceso a Internet, en solo un minuto de tiempo: los usuarios de WhatsApp intercambiaron 41.666.667 mensajes; los usuarios de Youtube subieron 500 horas de video; y 1.388.889 personas participaron de llamadas/videollamadas online [3]. Uno podría entonces pre-

guntarse ¿qué tiene la ciencia de la información cuántica para aportar a los esquemas de comunicaciones y de computación actual?

En el campo de la computación, como fue propuesto en el año 1965 por Gordon Moore [2], cada 2 años se duplica la cantidad de transistores en un microprocesador. Esto lleva a una miniaturización de los componentes electrónicos en los microprocesadores, por lo que eventualmente se llegará a tener componentes en una escala en la que los fenómenos cuánticos sean relevantes. Esto motivó a pensar qué sucedería al tener un dispositivo cuántico capaz de realizar tareas de cómputo, dando origen a una de las ramas de la ciencia de información cuántica, la computación cuántica. El primer modelo de computadora cuántica fue propuesto en el año 1980 [4]. Sin embargo aún hoy en día, continúa siendo un desafío el desarrollo de una computadora cuántica que pueda resolver problemas relevantes superando las capacidades de las computadoras clásicas. El mayor atractivo de estos dispositivos se encuentra en que, si bien cualquier problema resoluble por una computadora cuántica también lo es por una computadora clásica (y viceversa) [5], existen ciertos problemas que pueden ser resueltos eficientemente por una computadora cuántica y no por una computadora clásica. El más conocido de estos es el algoritmo de factorización de enteros, un algoritmo cuya complejidad crece exponencialmente en computación clásica, mientras que lo hace polinomialmente en una computadora cuántica [6]. Esto último, por ejemplo, pone en peligro gran parte de la seguridad informática actual que se basa en cifrado utilizando números enteros muy grandes, y que funciona bajo la suposición de que un adversario, limitado por las capacidades tecnológicas del momento, tendrá inconvenientes para desencriptar la clave secreta. Así, por ejemplo, aunque sea posible quebrar la clave de encriptación mediante una computadora clásica, ya que la factorización de números es un problema resoluble, esto implica, en ciertos casos, una escala de tiempo inmanejable. Sin embargo, quien tuviera a disposición una computadora cuántica, podría resolver este problema en tiempos considerablemente menores [6]. Esto motivó el origen de otra de las ramas centrales de la ciencia de información cuántica: la criptografía cuántica.

La criptografía cuántica busca la forma de sacar ventaja de las propiedades de sistemas cuánticos para realizar tareas criptográficas. Entre sus aplicaciones más notables, se encuentra la distribución cuántica de claves (QKD), un conjunto de protocolos que utiliza sistemas y canales de comunicación cuánticos para establecer una clave secreta entre dos usuarios autenticados, evitando que una tercera parte no autenticada (un espía) obtenga información de dicha clave, incluso cuando el espía es capaz de escuchar toda comunicación que se establece entre ambos usuarios [7, 8]. En teoría, los protocolos de QKD son inviolables, sin importar las capacidades tecnológicas actuales o futuras de un espía, dado que la comunicación queda intrínsecamente protegida por leyes físicas. Sin embargo, uno de los requisitos para poder realizar experimentalmente QKD, es poder establecer comunicaciones cuánticas a larga distancia, por lo que es necesario

un sistema cuántico capaz de viajar dichas distancias, sin que su estado sea alterado por interacciones con el entorno, siendo este uno de los inconvenientes principales que se presenta al momento de su implementación. Un ejemplo de sistema cuántico que cumple con los requisitos necesarios para realizar comunicaciones cuánticas a larga distancia es el fotón. Los fotones son partículas viajeras que no solo interactúan poco con el entorno, pudiendo viajar largas distancias sin perder coherencia (es decir, sin que su estado se vea afectado por ruido blanco) sino que también pueden ser manipulados a temperatura ambiente, y pueden integrarse fácilmente a las redes de comunicaciones ópticas clásicas ya existentes.

En los últimos años, las tecnologías cuánticas basadas en plataformas fotónicas se han vuelto una de las alternativas más promisorias para el desarrollo de aplicaciones que van desde encriptación segura de mensajes [9], comunicación a larga distancia [10], y simulación de fenómenos complejos [11]. Todas estas aplicaciones y, en general, los experimentos de óptica cuántica requieren de un conjunto de elementos controlables en el laboratorio, como son una fuente de fotones individuales, un generador de números aleatorios, divisores de haz, láminas de onda, detectores capaces de registrar fotones de a uno, y un contador de coincidencias. Es en este contexto que el foco de este Trabajo de Diploma estuvo puesto en el funcionamiento, diseño e implementación de dos de estos elementos: un módulo contador de coincidencias y un generador de números aleatorios. El mismo está organizado en 6 capítulos. En el Capítulo 2, se presentan los fundamentos de la teoría de la información cuántica en torno a tres pilares: el principio de superposición cuántica, el colapso de la función de onda y el teorema de no clonación. Se introducen los conceptos de bit cuántico (qubit), las compuertas cuánticas esenciales, y el fenómeno de entrelazamiento, junto con el formalismo matemático sobre el que se construye toda esta teoría. Luego, se presentan las bases de QKD, explicando el funcionamiento de dos de los protocolos más relevantes. Finalmente, se describe de qué forma todo lo anterior es realizable utilizando fotones como portadores de la información, y se analiza una de las técnicas más utilizadas como fuente de fotones únicos, que alternativamente puede utilizarse como fuente de pares de fotones entrelazados, destacándose la necesidad de un contador de coincidencias. Así mismo, se describe una pseudo-fuente de fotones únicos, y sus limitaciones. El Capítulo 3 se centra en la implementación de un módulo contador de coincidencias en una placa de desarrollo de prototipos FPGA. Este capítulo comienza con una explicación del funcionamiento del módulo junto con el diseño a implementar, seguido de una introducción a la placa y entorno de desarrollo de prototipos utilizada, finalizando con la implementación del dispositivo en dicha placa. En el Capítulo 4, se introduce el concepto de generador de números aleatorios (RNG), explicando las diferencias entre un generador de números pseudoaleatorios, un generador de números verdaderamente aleatorios, y un generador cuántico de números aleatorios. Luego, se realiza una comparación cualitativa entre

ellos, y se introduce el concepto de medidas de aleatoriedad para la determinación de la calidad de un RNG, finalizando con la implementación de un generador de números aleatorios en la placa de desarrollo FPGA. En el Capítulo 5, se describe la estadística de foto-detección de un conjunto de fuentes de fotones (láser, térmica, y de estados de Fock). Se muestran resultados experimentales y un análisis de la estadística obtenida, y cómo determinar experimentalmente la estadística de una fuente utilizando la función de correlación de segundo orden $g^{(2)}(\tau)$. Finalmente, se presenta una implementación en la placa de desarrollo FPGA que permite la simulación de los tres tipos de fuentes mencionados anteriormente. A modo de prueba de funcionamiento, del RNG y del módulo contador de coincidencias desarrollado, se determinó el valor de $g^{(2)}(0)$ para las fuentes simuladas y se comparó con lo esperado teóricamente. Por último, en el capítulo Conclusiones, se presentan las conclusiones del trabajo, acompañadas de las posibles mejoras a los dispositivos implementados, y las futuras aplicaciones que se prevén para los mismos.

Capítulo 2

Información Cuántica

2.1. Fundamentos de la información cuántica

Desde hace algunas décadas la información ha tomado un rol fundamental en nuestras vidas. Cantidades enormes de ella viajan constantemente desde y hacia un teléfono celular o una computadora personal. Para transmitirla y procesarla, la misma debe ser primero codificada en algún sistema físico. En los sistemas informáticos actuales la mínima unidad de información se codifica en un *bit*, el cual es un sistema físico clásico con dos estados posibles representados por los valores 1 y 0. Entre las implementaciones más comunes de un bit se encuentran, por ejemplo, dos estados de un circuito *flip-flop* (para transmisión y procesamiento) o dos niveles de carga almacenada en un capacitor (para almacenamiento).

Con el paso de los años, los componentes electrónicos en los dispositivos de comunicación y computación son fabricados cada vez en menor tamaño. Eventualmente, si esta miniaturización de componentes se mantiene, los sistemas electrónicos llegarán a escalas en las que los fenómenos cuánticos sean relevantes, por lo que surgen las preguntas: ¿es posible transmitir, procesar y almacenar información codificada en un sistema cuántico? Dado que los sistemas cuánticos presentan comportamientos sustancialmente diferentes a los sistemas clásicos, ¿qué sucede con la información cuando el soporte físico tiene un comportamiento cuántico? ¿Es posible hablar de información en forma abstracta mediante una teoría matemática sin tener en cuenta el soporte físico? ¿Qué tiene la física para aportar a la teoría de la información? Éstas fueron algunas de las preguntas que dieron lugar a lo que hoy se conoce como información cuántica y que se presentarán en este capítulo.

2.1.1. El qubit

La ciencia de la información cuántica [5] se encarga de estudiar del procesamiento de la información utilizando sistemas que obedecen las leyes de la mecánica cuántica.

La unidad fundamental en información cuántica, la contraparte del *bit* en información clásica, es el *bit cuántico*, de aquí en más *qubit* por su acrónimo en inglés, *quantum bit*. Un qubit será entonces un sistema cuántico con dos estados posibles, distinguibles mediante mediciones, tales como dos niveles de energía de un ion, la polarización de un fotón, o el espín de un electrón. Debido a la naturaleza cuántica del sistema, un qubit no se comporta de la misma manera que un bit clásico, en particular debido a tres puntos fundamentales: el principio de superposición de estados cuánticos, el colapso de la función de onda y el teorema de no clonación [5, 12–14].

Para describir matemáticamente estos sistemas se utiliza el formalismo del álgebra lineal. Por lo tanto, el estado de un qubit en un dado estado cuántico puede representarse como un vector de dimensión 2 en un espacio complejo conocido como espacio de Hilbert, y representado con $\mathcal{H}$. Los dos estados se denotan $|0\rangle$ y $|1\rangle$, y su representación en vectores columna es

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad , \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.1}$$

Además, a diferencia de lo que sucede para un bit clásico, debido al principio de superposición cuántica un qubit puede estar en un estado $|\psi\rangle$ que sea combinación lineal de $|0\rangle$ y $|1\rangle$, de modo de que su estado más general[1] está dado por

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \tag{2.2}$$

donde α y β son números complejos. De este modo, los estados $|0\rangle$ y $|1\rangle$ forman una base en el espacio $\mathcal{H}$, que se denomina base estándar o base computacional. Sin embargo, debido a que los instrumentos de medición son clásicos, para medidas proyectivas en la base estándar, estos pueden estar solo en el estado $|0\rangle$ o $|1\rangle$, es decir, el resultado de una medida será alguno de los estados de la base, pero no el estado general $|\psi\rangle$. Por lo tanto, si se quisiera medir este estado, es decir, determinar los coeficientes α y β, sería imposible lograrlo en un número finito de mediciones. Al realizar una medida sobre el sistema, la función de onda del qubit 'colapsa' aleatoriamente a uno de los dos estados de la base de medición, de modo que se obtendrá el resultado $|0\rangle$ con probabilidad $|\alpha|^2$, y el resultado $|1\rangle$ con probabilidad $|\beta|^2$. Así, debido a la interpretación probabilística de estos coeficientes, debe cumplirse que

$$|\alpha|^2 + |\beta|^2 = 1, \tag{2.3}$$

[1] Aquí solo nos limitamos al estado *puro* más general, donde es válida la representación del estado como un vector en el espacio de Hilbert. Cuando el estado de un sistema cuántico no está definido, sino que está dado por una mezcla estadística, el formalismo a utilizar es el de la matriz densidad [5].

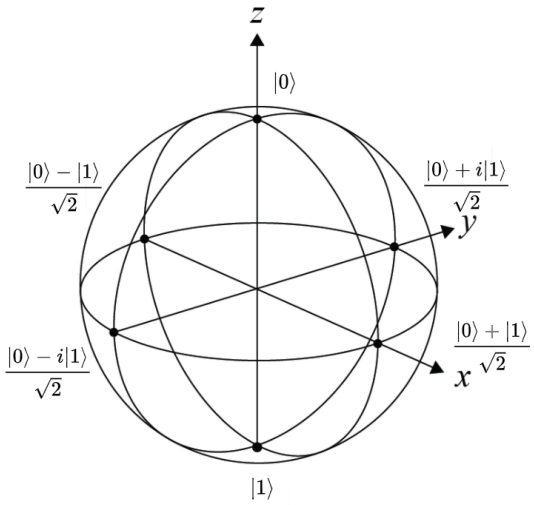


Figura 2.1: Representación gráfica de la esfera de Bloch, con la ubicación usual para los estados de la base computacional $\{|0\rangle, |1\rangle\}$. Alternativamente a la base computacional, otras bases de interés son las dadas por $\left\{\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$ y por $\left\{\frac{|0\rangle+i|1\rangle}{\sqrt{2}}, \frac{|0\rangle-i|1\rangle}{\sqrt{2}}\right\}$, que corresponden a rotaciones de la base computacional en torno al eje y y al eje x respectivamente.

lo que permite representar el estado $|\psi\rangle$ como un vector sobre una esfera de radio unidad, conocida como esfera de Bloch (fig. 2.1).

Se tienen aquí dos grandes diferencias entre un bit y un qubit:

- Mientras que un bit puede tomar solo los valores 0 o 1, un qubit es un vector cuyo extremo pertenece a la superficie de una esfera unitaria (fig. 2.2a).
- No es posible determinar el estado de un qubit a partir de una única medida, aún si las incertezas experimentales fueran despreciables, excepto cuando su estado coincide con alguno de los estados de la base de medición (fig. 2.2b).

Compuertas cuánticas de un qubit

Debido a la representación del estado de un qubit como un vector en un espacio de Hilbert, todas las operaciones para manipular su estado serán operadores que actúan sobre $\mathcal{H}$, representados mediante matrices complejas de 2×2. En analogía al modelo clásico de computación, estos operadores reciben el nombre de *compuertas cuánticas*. La única condición que deben cumplir es que sean unitarios, es decir, dado un operador $\hat{U}$ que representa una compuerta cuántica, debe cumplirse que $\hat{U}\hat{U}^\dagger = \hat{I}$, donde $\hat{I}$ es el operador identidad que deja invariante el estado. Esta condición garantiza que si $|\psi\rangle$ es un vector que representa el estado de un qubit, y por lo tanto de norma 1, $U|\psi\rangle$ es otro vector de dimensión 2 también con norma 1, por lo que puede asociarse sin ambigüedad con el estado de un qubit.

Entre las compuertas cuánticas más utilizadas se encuentran las $\hat{X}$, $\hat{Y}$ y $\hat{Z}$, que representan rotaciones en la esfera de Bloch de 180° respecto a los eje x, y y z respec-

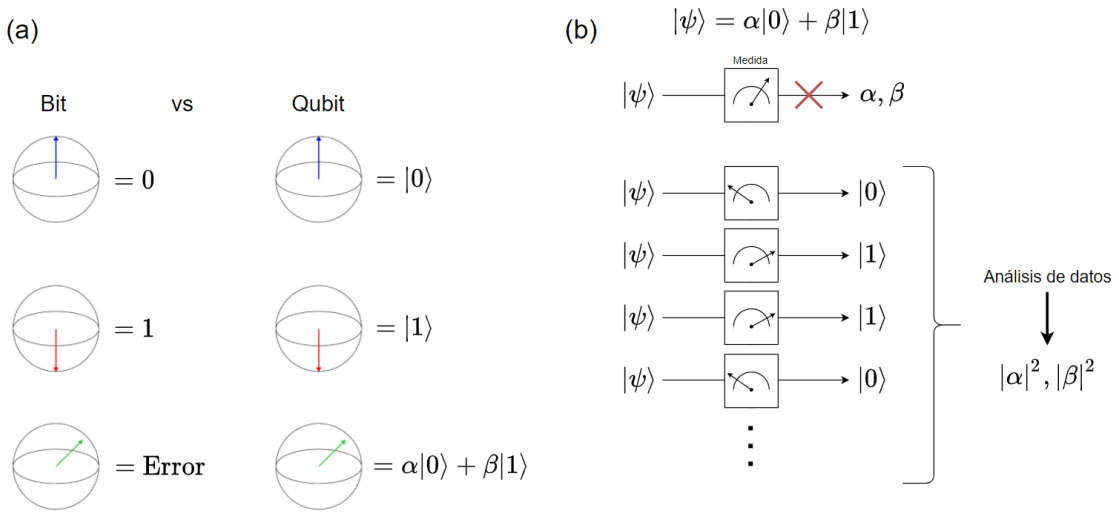


Figura 2.2: (a) Diferencia fundamental entre bits y qubits, donde el bit puede tomar solo dos valores y cualquier valor intermedio es un error, mientras que para el qubit cualquier estado de la forma $\alpha\,|0\rangle + \beta\,|1\rangle$ es válido. (b) Representación esquemática de una medida sobre un estado de un qubit $|\psi\rangle$. Para determinar los coeficientes α y β, imposible con una única medida, es necesario realizar un gran número de medidas al estado, y determinar las probabilidades de obtener cada resultado, $|\alpha|^2$ y $|\beta|^2$.

tivamente, y la compuerta de Hadamard, que representa una rotación en la esfera de Bloch de 180° respecto al eje de 45° en el plano $X - Z$. Estas matrices se definen en la base computacional como

$$\hat{X} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2.4}$$

$$\hat{Y} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{2.5}$$

$$\hat{Z} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.6}$$

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.7}$$

Para representar esquemáticamente la acción de una compuerta sobre un dado estado cuántico, se utilizan diagramas circuitales cuánticos, análogos a su contraparte clásica. En este modelo de circuito, se tienen qubits viajando por *cables cuánticos*, representados con líneas horizontales, y *compuertas cuánticas* que actúan sobre dichos qubits, representadas por bloques rectangulares (fig. 2.3).

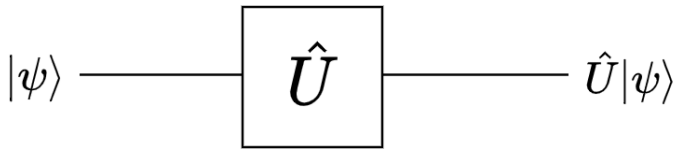


Figura 2.3: Diagrama circuital cuántico, donde al estado $|\psi\rangle$, que viaja por el cable cuántico, se le aplica una compuerta arbitraria $\hat{U}$, y se obtiene al final del circuito el estado $\hat{U}\,|\psi\rangle$

2.1.2. Múltiples qubits

Dado un conjunto de N qubits, donde $\mathcal{H}_i$ denota el espacio de Hilbert asociado al qubit i, el espacio de Hilbert del sistema compuesto viene dado por

$$\mathcal{H} = \bigotimes_{i=1}^{N} \mathcal{H}_i = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes ... \otimes \mathcal{H}_N, \tag{2.8}$$

donde $\otimes$ denota el producto tensorial. De esta manera, suponiendo un estado puro $\{|\phi_i\rangle\}$ base de $\mathcal{H}_i$[2], un estado $|\psi\rangle \in \mathcal{H}$ puede escribirse como

$$|\psi\rangle = \sum_{\phi_1,\phi_2,...,\phi_N} \alpha_{\phi_1,\phi_2,...\phi_N} \, |\phi_1\rangle \otimes |\phi_2\rangle \otimes ... \otimes |\phi_N\rangle \equiv \sum_{\{\phi_i\}} \alpha_{\phi_1,\phi_2,...\phi_N} \, |\phi_1, \phi_2, ...\phi_N\rangle \, . \tag{2.9}$$

Estos estados viven en un espacio de Hilbert de dimensión 2^N, por lo que serán necesarios $2^N - 1$ coeficientes[3] complejos para describir por completo cada estado. Si comparamos esto con un estado de N bits clásicos,

$$a_{N-1} \cdot 10^{N-1} + a_{N-2} \cdot 10^{N-2} + \cdots + a_0 \cdot 10^0, \tag{2.10}$$

obtengo que son necesarios N coeficientes para describir por completo el estado. Esto implica que para representar un estado de 2^N bits, necesito solo N qubits. Este resultado es lo que permite que el tiempo de cómputo de ciertos algoritmos cuánticos sea considerablemente menor a sus contrapartes clásicas [6].

Entrelazamiento cuántico

Uno de los fenómenos cuánticos imposibles de explicar clásicamente es el de entrelazamiento cuántico entre sistemas. Cuando dos o más partículas interactúan de forma tal que el estado de cada una de ellas no puede describirse independientemente del

[2]Por ejemplo, para la base computacional, $\phi_i = 0, 1$.

[3]Con $2^N - 1$ coeficientes definidos, el restante queda definido por la condición de normalización (2.3).

estado de las otras, incluso si están separadas a grandes distancias, se dice que estas partículas están *entrelazadas*. El entrelazamiento introduce un tipo de correlaciones entre sistemas (correlaciones cuánticas) que son irreproducibles e inexplicables mediante las leyes de la mecánica clásica.

El fenómeno de entrelazamiento cuántico es considerado uno de los recursos fundamentales en información cuántica [5]. El aprovechamiento de este nuevo recurso permite, si se utiliza correctamente, la implementación de nuevos algoritmos que serían imposibles, o extremadamente difíciles, de lograr con computación clásica, tales como la teleportación cuántica o el algoritmo de factorización de Shor.

Limitándonos al caso de solo dos qubits, aunque puede fácilmente extenderse a N qubits, reducimos la ecuación (2.9) a

$$|\psi\rangle = \sum_{\phi_1,\phi_2} \alpha_{\phi_1,\phi_2} |\phi_1, \phi_2\rangle \,, \tag{2.11}$$

con $|\phi_1\rangle$ y $|\phi_2\rangle$ no necesariamente en la base computacional. De esta manera, por ejemplo, en la base computacional el estado de dos qubits más general se escribe

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} . \tag{2.12}$$

Los estados de la forma (2.11) pueden clasificarse en estados separables y no separables. Se dice que un estado es separable si $\alpha_{\phi_1,\phi_2} = 1$ para algún par (ϕ_1, ϕ_2), y cero para el resto, es decir, puede escribirse como

$$|\psi\rangle = |\phi_1, \phi_2\rangle \,. \tag{2.13}$$

En contraste, si no existe una base en la cual el estado pueda escribirse como en la ecuación (2.12), se dice que es no separable o *entrelazado*. En particular, existe un tipo de estos estados que se conocen como *máximamente entrelazados*, en los que mientras que el estado conjunto está bien definido, los estados individuales de cada partícula están completamente indefinidos, es decir, no portan ningún tipo de información. En el caso de 2 qubits, un conjunto particular de estos estados, definidos en la base computacional como

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad |\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \tag{2.14}$$

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \qquad |\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \tag{2.15}$$

se conocen como estados de Bell, y cumplen un rol fundamental en gran parte de los algoritmos cuánticos. Uno de estos algoritmos, el más importante para aplicaciones de seguridad criptográfica, es el algoritmo de Shor para factorización de enteros en sus factores primos. Clásicamente, el tiempo necesario para factorizar un número entero crece exponencialmente con el número a factorizar, lo que garantiza la seguridad los protocolos de cifrado basados en números grandes [15]. Sin embargo, en 1994, Peter Shor encontró un algoritmo cuántico que, aprovechando el fenómeno de entrelazamiento, es capaz de factorizar enteros en un tiempo polinomial [6], una mejora considerable respecto a cualquier algoritmo clásico. Esta reducción en el tiempo de factorización presentó (y presenta) una amenaza a cualquier cifrado basado en números grandes, puesto que una computadora cuántica sería capaz de descifrar la clave generada a partir de los factores primos del número en cuestión, quebrantando por completo la seguridad criptográfica actual.

Otra característica fundamental del entrelazamiento cuántico es el hecho de que las partículas entrelazadas seguirán estándolo (siempre y cuando el entorno no perturbe el estado) a pesar de la distancia entre ellas. Supongamos que se tienen dos partículas, preparadas en el estado conjunto $|\beta_{00}\rangle$; si bien conocemos el estado en conjunto de ambas partículas, el estado de cada partícula individual está completamente indefinido. Supongamos ahora que se envía una de las partículas a Marte, en tanto que la otra se queda en la Tierra. Si el envío se logra hacer sin que el estado de las partículas sea perturbado, al hacer una medida sobre la partícula en la Tierra el estado de la partícula en Marte queda completamente definido. Actualmente, el récord de separación entre partículas manteniendo el entrelazamiento fue logrado en 2017 por Yin et al., separando dos qubits entrelazados 1200 km entre sí [10].

Compuertas de múltiples qubits

Análogamente al caso de un qubit, las compuertas de múltiples qubits son operadores que actúan sobre los estados pertenecientes al espacio de Hilbert del sistema compuesto. Suponiendo N qubits, los operadores son matrices complejas de $2^N \times 2^N$. El hecho de que las compuertas actúen sobre múltiples qubits permite la creación de compuertas *controladas*, es decir, que modifican el estado de un qubit i en función del estado de otro qubit j. Un ejemplo usual de estas compuertas, debido a la existencia de una contraparte clásica, es la compuerta llamada *Control-NOT* (CNOT). Esta compuerta invierte el estado del segundo qubit, en el caso que el primer qubit esté en el estado $|1\rangle$, de manera que

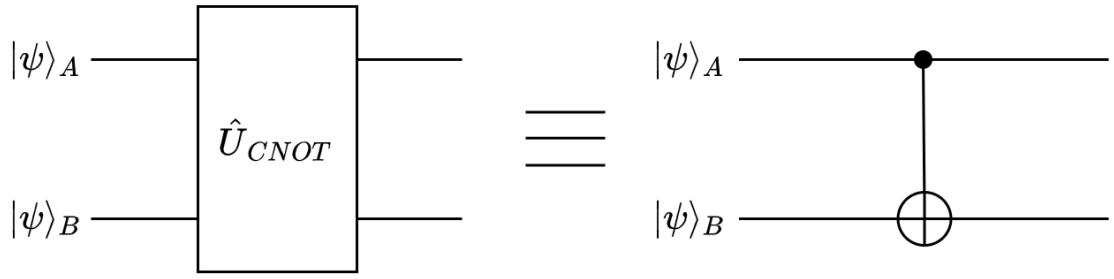


Figura 2.4: Diagrama circuital de la compuerta *Control-NOT*, donde el estado $|\psi\rangle_B$ es *controlado* por el estado $|\psi\rangle_A$.

$$
\begin{aligned}
|00\rangle &\xrightarrow{CNOT} |00\rangle \\
|01\rangle &\xrightarrow{CNOT} |01\rangle \\
|10\rangle &\xrightarrow{CNOT} |11\rangle \\
|11\rangle &\xrightarrow{CNOT} |10\rangle \,.
\end{aligned}
$$

El operador asociado a está compuerta se denota como $\hat{U}_{CNOT}$, y su representación matricial en la base computacional será

$$
\hat{U}_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{2.16}
$$

La representación circuital de esta compuerta es análoga a su contraparte clásica (fig. 2.4). Esta compuerta, junto con la compuerta Hadamard definida en la ecuación (2.7), es utilizada para generar los estados de Bell, de manera que

$$
|\beta_x\rangle = \hat{U}_{CNOT}(\hat{H} \otimes \hat{I})\,|x\rangle \,, \tag{2.17}
$$

con $x = \{00, 01, 10, 11\}$, y $\hat{H} \otimes \hat{I}$ un operador de dos qubits que implica aplicar la compuerta Hadamard al primer qubit, en tanto que no cambia el estado del segundo qubit (fig. 2.5).

Teorema de no clonación

El teorema de no clonación es otro de los pilares en los que se basa una de las áreas de mayor relevancia dentro de la información cuántica, como es la criptografía cuántica. El mismo asegura que es imposible poder crear una copia idéntica de un estado cuántico

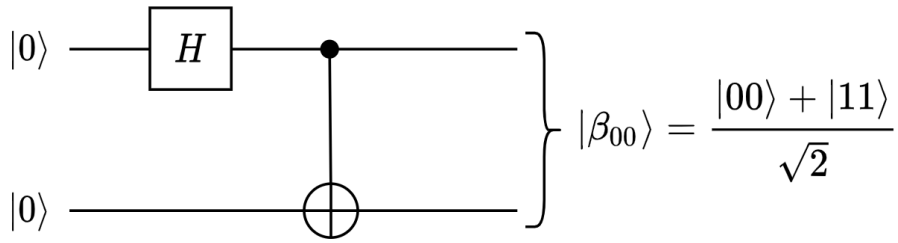


Figura 2.5: Diagrama circuital de la generación del estado de Bell $|\beta_{00}\rangle$ a partir de los estados $|0\rangle$ en ambos qubits y las compuertas Hadamard y *Control-NOT*.

arbitrario desconocido, al menos de forma determinista [4], lo que nos dice que no es posible contar con una 'fotocopiadora cuántica'. En principio este concepto es casi contradictorio a lo acostumbrado, puesto que la información clásica puede copiarse con facilidad. Ejemplos de esto se encuentran en la vida cotidiana constantemente, como es el caso de este trabajo que está leyendo ahora mismo (una copia del original), ya sea en una computadora o impresa en papel.

El teorema de no clonación no previene que *todos* los estados cuánticos sean copiados, si no que establece la imposibilidad de copiar estados cuánticos no ortogonales. Es decir, sean $|\psi\rangle$ y $|\phi\rangle$ dos estados cuánticos no ortogonales, entonces el teorema de no clonación implica que es imposible construir un dispositivo cuántico que, al ingresarle el estado $|\psi\rangle$ o $|\phi\rangle$, obtenga a la salida dos copias del estado de entrada, $|\psi\rangle\,|\psi\rangle$ o $|\phi\rangle\,|\phi\rangle$. Para demostrar este teorema, supongamos que existe un operador $\hat{U}_{copia}$ tal que, para los dos estados mencionados anteriormente, cumple

$$\hat{U}_{copia}\,|\psi, v\rangle = |\psi, \psi\rangle\,, \tag{2.18}$$

$$\hat{U}_{copia}\,|\phi, v\rangle = |\phi, \phi\rangle\,, \tag{2.19}$$

donde $|v\rangle$ es un estado inicial, por ejemplo, vacío. Con estas dos ecuaciones, puedo escribir el producto entre ellas, obteniendo

$$\langle\psi, v|\,\hat{U}^{\dagger}_{copia}\hat{U}_{copia}\,|\phi, v\rangle = \langle\psi, \psi|\phi, \phi\rangle = (\langle\psi|\phi\rangle)^2. \tag{2.20}$$

Teniendo en cuenta que $\hat{U}_{copia}$ debe ser unitario, entonces $\hat{U}^{\dagger}_{copia}\hat{U}_{copia} = \hat{I}$, y el primer término de la ecuación anterior se reduce a

$$\langle\psi, v|\,\hat{U}^{\dagger}_{copia}\hat{U}_{copia}\,|\phi, v\rangle = \langle\psi, v|\phi, v\rangle = \langle\psi|\phi\rangle, \tag{2.21}$$

[4] Es posible realizar una copiadora cuántica probabilística, es decir, un dispositivo copie el estado cuántico con una dada probabilidad [16].

que al reemplazarlo en la ecuación (2.20), se obtiene

$$\langle\psi|\phi\rangle = (\langle\psi|\phi\rangle)^2, \tag{2.22}$$

lo que implica o bien que $\langle\psi|\phi\rangle = 1$, es decir, $|\psi\rangle = |\phi\rangle$, que contradice la hipótesis $|\psi\rangle \neq |\phi\rangle$; o bien $\langle\psi|\phi\rangle = 0$, por lo que $|\psi\rangle$ y $|\phi\rangle$ son ortogonales, y dejan de ser estados arbitrarios, que también contradice la hipótesis. De esta manera, no existe un operador $\hat{U}_{copia}$ universal que me permita realizar una copia de un estado arbitrario $|\psi\rangle$.

2.1.3. Distribución cuántica de claves

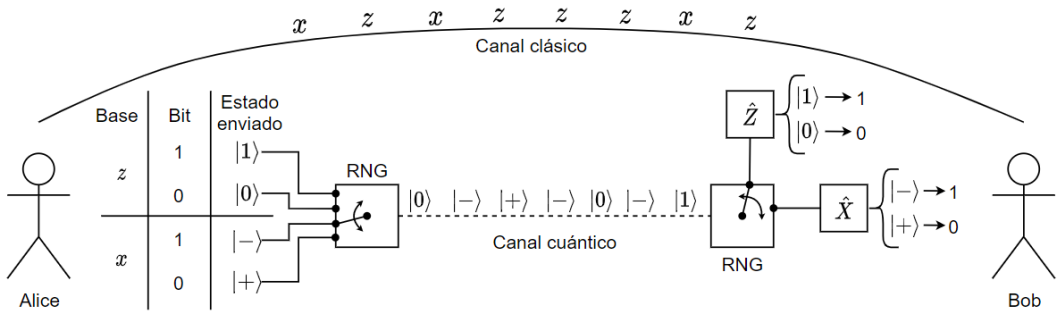
Una de las ramas dentro de la ciencia de información cuántica es la de la criptografía cuántica, que se encarga de estudiar las posibles aplicaciones de la teoría de información cuántica en tareas criptográficas [17]. Entre estas aplicaciones, la más conocida de ellas es la distribución cuántica de claves (QKD por sus siglas en inglés), que permite la generación de claves secretas solo conocidas por los usuarios autenticados.

Una de las características más importantes del protocolo de QKD, es que los usuarios autenticados son capaces de detectar la presencia de un espía intentando obtener la información de la clave, y entonces decidir interrumpir el envío de la clave cuando el canal de comunicación ha sido atacado. Recordando los tres pilares que se mencionaron en las secciones previas, el principio de superposición (sección 2.1.1), el colapso de la función de onda (sección 2.1.1) y el teorema de no clonación (sección 2.1.2), se pueden entender los fundamentos del protocolo de QKD. Por un lado, para obtener información de la clave, el espía necesita medir el estado que obtiene, lo que lo colapsaría a uno de los dos estados de la base en la que el espía esté midiendo, y no podría reenviar el mismo estado que recibió. Por otro lado, suponiendo que el espía quisiera copiar el estado y medir las copias para no perturbar el estado original, entonces el teorema de no clonación prohíbe que esto suceda. Estos dos fenómenos garantizan la seguridad de la clave generada mediante QKD.

Protocolo BB84

Entre los protocolos de QKD, uno de los más comunes es el BB84, denominado así en honor a sus creadores, Bennett y Brassard, y el año de su publicación, 1984 [7]. Este protocolo está pensado para codificar una clave binaria en el estado de qubits. Tendremos dos partes que buscan generar una clave secreta en común: un emisor A (tradicionalmente llamado Alice) y un receptor B (tradicionalmente llamado Bob). Previo a la generación de la clave, Alice y Bob definen dos bases, usualmente:

- la primera de ellas, la base z en la esfera de Bloch, compuesta por los autoestados del operador $\hat{Z}$: $|0\rangle$ y $|1\rangle$,



Transmisión y detección cuántica	Bits aleatorios de Alice	0	1	1	0	1	0	1	0
	Bases de Alice	z	z	x	z	x	x	x	z
	Estados enviados por Alice	$\|0\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$
	Bases de Bob	x	z	x	z	z	z	x	z
	Estados medidos por Bob	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$
Comunicación clásica	Reconciliación		✓	✓	✓			✓	✓
	Clave secreta compartida	-	1	1	0	-	-	1	0

Figura 2.6: Esquema de funcionamiento del protocolo BB84, donde Alice envía una cadena de qubits por un canal cuántico, que Bob mide aleatoriamente en las bases z y x. Bob, por un canal clásico, le comunica a Alice las bases en las que midió, y en un proceso de reconciliación deciden que bits descartan y cuales mantienen. De esta manera, en este ejemplo, Alice y Bob generan la clave compartida '11010'.

- y la segunda, la base x en la esfera de Bloch, compuesta por los autoestados del operador $\hat{X}$: $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}} \equiv |0\rangle_x$ y $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}} \equiv |1\rangle_x$.

Este protocolo fue originalmente pensado para representar cada qubit en el estado de polarización de un fotón, aunque es extensible en principio a cualquier sistema cuántico de dimensión 2. Una vez definidas las bases, el protocolo se realiza en los siguientes pasos:

1. En primera instancia, Alice genera una secuencia aleatoria de 0's y 1's.

2. Alice codifica cada bit en un qubit, $|0\rangle$ o $|+\rangle \equiv |0\rangle_x$ si el bit correspondiente es 0, y $|1\rangle$ o $|-\rangle \equiv |1\rangle_x$ si el bit correspondiente es 1. Para cada bit, Alice elije aleatoriamente en que base lo codifica, y envía el qubit por el canal cuántico.

3. Para cada qubit recibido, Bob decide aleatoriamente en que base medir, x o z. Si para algún qubit Bob elije la misma base que Alice, suponiendo que no hay espías ni impurezas en el canal cuántico, Alice y Bob comparten el mismo bit. Por el otro lado, si Bob elije una base distinta a la de Alice, entonces el bit determinado por Bob no coincide con el bit enviado por Alice. Por ejemplo, si Bob recibe el qubit $|-\rangle$ y lo mide en la base z, es igualmente probable que obtenga los resultados 0 y 1. Esta discrepancia entre Alice y Bob sucederá, en promedio, la mitad de las veces.

4. Bob y Alice comparten, por un canal clásico público, las bases que utilizó cada uno para medir/enviar cada qubit. Es importante notar que comunican solo las bases utilizadas, y no el resultado de las medidas.

5. Alice y Bob eliminan todos los bits que se correspondan con casos en los que usaron diferentes bases, quedándose con el resto. Al final de este procedimiento, Alice y Bob generan una clave secreta compartida (fig. 2.6).

6. Por un canal público, Alice y Bob comparan parte de la clave secreta generada por ambos. A partir de esta comparación, pueden estimar el *ratio de error* R debido a espías o impurezas en el canal cuántico, como el cociente entre las discrepancias en sus resultados y la cantidad total de bits comparados. Si R es demasiado alto, descartan todos los resultados y vuelven a realizar el protocolo desde el comienzo.

Protocolo E91

Otro protocolo de QKD, en el que se aprovecha el fenómeno de entrelazamiento cuántico (sección 2.1.2), es el E91 (también llamado protocolo EPR), denominado así en honor a su creador, Ekert, y su año de publicación, 1991 [8]. A diferencia del protocolo BB84, donde se transmiten y detectan qubits individuales, el protocolo E91 se basa en la transmisión/detección de qubits entrelazados. Para este protocolo, es necesario definir tres bases, a diferencia de las dos necesarias para el protocolo BB84. Para ello, se utilizan las dos ya mencionadas para el protocolo BB84, y se agrega la base y en la esfera de Bloch, compuesta por los autoestados del operador $\hat{Y}$: $|i\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}}$ y $|-i\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}}$. El protocolo se realiza en los siguientes pasos:

1. Se prepara un par de qubits en el estado de Bell $|\beta_{01}\rangle$. De ese par, uno de los qubits es enviado a Alice, y el otro es enviado a Bob. Nótese que no es estrictamente necesaria la presencia de una tercera parte que genere y distribuya el estado entrelazado a compartir, puesto que Alice puede crear el estado de Bell, y mandar un qubit del par a Bob.

2. Para cada qubit recibido, tanto Alice como Bob miden aleatoriamente en una de las tres bases disponibles (x, y o z). Suponiendo que casualmente ambos midan en la misma base (lo cual ocurrirá, en promedio, un 33,3 % de las veces), los resultados que obtenga Bob serán siempre ortogonales a los obtenidos por Alice, debido al entrelazamiento máximo característico de los estados de Bell. Esto implica que si ambos miden en la base x, y Alice obtiene el estado $|+\rangle$, entonces Bob obtendrá el estado $|-\rangle$.

3. Alice y Bob anuncian, por un canal público clásico, las bases que utilizaron para cada medida, descartando todos aquellos resultados en los que no coincidan.

4. Bob aplica una operación de negación sobre todos sus resultados, tal que $0 \to 1$ y $1 \to 0$, generando una clave secreta compartida entre ambos.

Al igual que para el protocolo BB84, Alice y Bob pueden detectar la presencia de un espía o ruido en el canal cuántico, verificando el valor de una magnitud C, definida como

$$C \equiv \langle \hat{X} \otimes \hat{X} \rangle - \langle \hat{X} \otimes \hat{Z} \rangle + \langle \hat{Z} \otimes \hat{X} \rangle + \langle \hat{Z} \otimes \hat{Z} \rangle. \tag{2.23}$$

En ausencia de espías y ruido en el canal cuántico, $|C| = 2\sqrt{2}$ [16], mientras que $|C| < 2\sqrt{2}$ implica la presencia de alguno de estos efectos[5].

2.2. Información cuántica con fotones

Actualmente, en comunicación clásica ya se utiliza luz para codificar información, permitiendo transmitir señales de internet, teléfono o televisión por medio de fibra óptica [19]. Los fotones son sistemas cuánticos, en primera instancia, fácilmente manipulables a temperatura ambiente mediante tecnología estándar (espejos, divisores de haz, lentes, etc). Son partículas viajeras y que interactúan muy poco con el entorno, lo que los hace ideales para el transporte de información a larga distancia. Surge entonces la pregunta: ¿existe algún grado de libertad fotónico que posea las propiedades necesarias para ser considerado un qubit?

2.2.1. Polarización como qubit

De entre los varios grados de libertad de un fotón [20], el más comúnmente utilizado para codificar información es la polarización. Para el caso de ondas electromagnéticas, la polarización es la dirección de oscilación del campo eléctrico, que se da en un plano perpendicular a la dirección de propagación del campo. Para poder definirlo como qubit, es necesario definir dos estados ortogonales de polarización, que actúen como base del espacio de Hilbert asociado. Los más comúnmente utilizados son la polarización horizontal $|H\rangle$ y vertical $|V\rangle$ (fig. 2.7), pero de la misma manera se podrían definir utilizando la polarización circular izquierda $|L\rangle$ y circular derecha $|R\rangle$, o las polarizaciones diagonal $|D\rangle$ y antidiagonal $|A\rangle$. Cada una de estas posibles bases puede expresarse en función del resto; en particular, en función de la base $\{|H\rangle, |V\rangle\}$, se tiene

[5] Este es un caso particular de lo que se conoce como desigualdades de Bell, y suelen utilizarse para medir entrelazamiento entre estados [18].

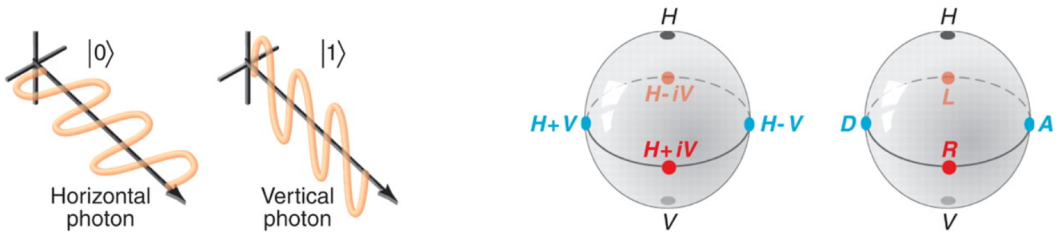


Figura 2.7: A la izquierda, la representación esquemática de los estados de polarización de un fotón $|H\rangle$ y $|V\rangle$. A la derecha, la representación de la esfera de Bloch para estados de polarización, donde se marcan con puntos algunos estados de interés y su denominación.

$$|D\rangle = \frac{|H\rangle + |V\rangle}{\sqrt{2}} \qquad |A\rangle = \frac{|H\rangle - |V\rangle}{\sqrt{2}} \tag{2.24}$$

$$|R\rangle = \frac{|H\rangle + i\,|V\rangle}{\sqrt{2}} \qquad |L\rangle = \frac{|H\rangle - i\,|V\rangle}{\sqrt{2}}. \tag{2.25}$$

De aquí en adelante, asociaremos la base computacional a la base de polarización $\{|H\rangle, |V\rangle\}$, de modo que

$$|H\rangle \equiv |0\rangle \quad \text{y} \quad |V\rangle \equiv |1\rangle .$$

Para poder realizar una implementación real de qubits codificados en polarización es necesario, en primera instancia, poder generar estados de un fotón con una polarización arbitraria. Luego, se debe contar con los dispositivos ópticos necesarios para realizar las compuertas a aplicar y, finalmente, poder realizar una medida proyectiva sobre el qubit para obtener uno de los dos posibles resultados.

Generación de estados de polarización

Suponiendo que se cuenta con una fuente de fotones individuales, que en el caso más general podrían no estar polarizados, es necesario poder generar un estado arbitrario como el de la ecuación (2.2). En primera instancia, un dispositivo óptico que se encarga de gran parte del trabajo es un filtro polarizador lineal (fig. 2.8). Dicho filtro transmite selectivamente fotones con una polarización lineal dada y bloquea el resto, es decir, dada una fuente no polarizada, un polarizador lineal genera estados de la forma

$$|\phi\rangle = \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} = \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix}, \tag{2.26}$$

donde φ es el ángulo entre el eje óptico del polarizador y la horizontal del lugar. Este estado es similar al de la ecuación (2.2), solo que ahora los parámetros son tales que $(\alpha', \beta') \in \mathbb{R}$.

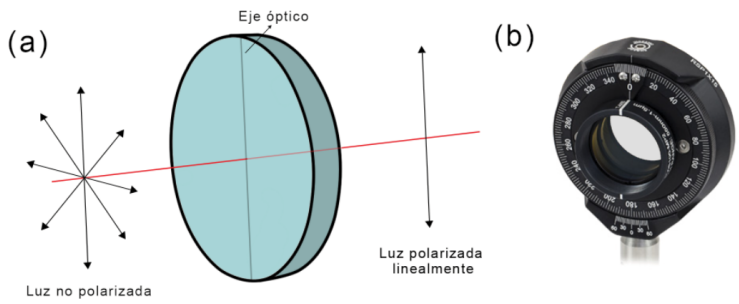


Figura 2.8: (a) Esquema del funcionamiento de un polarizador lineal, donde ingresa luz no polarizada, transmitiendo solo una polarización lineal dada por el eje óptico. (b) Foto real de un polarizador lineal en un montaje rotante (Thorlabs LPMIR100-MP2).

Teniendo estados con polarización definida, para generar estados realmente arbitrarios son necesarios dos dispositivos ópticos: las láminas de media onda y de cuarto de onda (HWP y QWP respectivamente, por sus siglas en inglés) (fig. 2.9). La HWP es un componente óptico que permite cambiar el peso relativo de las componentes horizontal y vertical de la polarización, de modo que, partiendo del estado $|H\rangle$, representa una rotación en torno al eje y en la esfera de Bloch. La acción de la HWP puede representarse como un operador en la base computacional tal que

$$\hat{U}_{HWP}(\theta) = e^{-i\theta\sigma_2} e^{-i\frac{\pi}{2}\sigma_3} e^{i\theta\sigma_2} = -i \begin{pmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{pmatrix}, \tag{2.27}$$

donde θ es el ángulo entre el eje óptico y el vertical, y σ_k, con $k = 1, 2, 3$, representan las matrices de Pauli. De la misma manera, la QWP permite cambiar el peso relativo de las componentes horizontal y vertical de la polarización, y permite agregar una diferencia de fase relativa entre ambas, de modo que partiendo del estado $|H\rangle$, representa una rotación en torno al eje x en la esfera de Bloch. La representación en la base computacional de la lámina de cuarto de onda viene dada por

$$\hat{U}_{QWP}(\theta) = e^{-i\theta\sigma_2} e^{-i\frac{\pi}{4}\sigma_3} e^{i\theta\sigma_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 - i\cos(2\theta) & -i\sin(2\theta) \\ -i\sin(2\theta) & 1 + i\cos(2\theta) \end{pmatrix}, \tag{2.28}$$

donde nuevamente θ representa el ángulo entre el eje óptico y el vertical. Estos dos dispositivos ópticos son suficientes para manipular el estado de polarización mediante rotaciones arbitrarias, permitiendo generar los estados deseados sobre la esfera de Bloch [21]. Para esto, dados los ángulos de Euler α, β, γ de la rotación buscada, el operador de rotación arbitraria viene dado por

$$R(\alpha, \beta, \gamma) = \hat{U}_{QWP}(\gamma)\hat{U}_{HWP}(\beta)\hat{U}_{QWP}(\alpha) = e^{-i\left(\gamma + \frac{3\pi}{4}\right)\sigma_2} e^{i(\alpha - 2\beta + \gamma)\sigma_3} e^{i\left(\alpha - \frac{\pi}{4}\right)\sigma_2}, \tag{2.29}$$

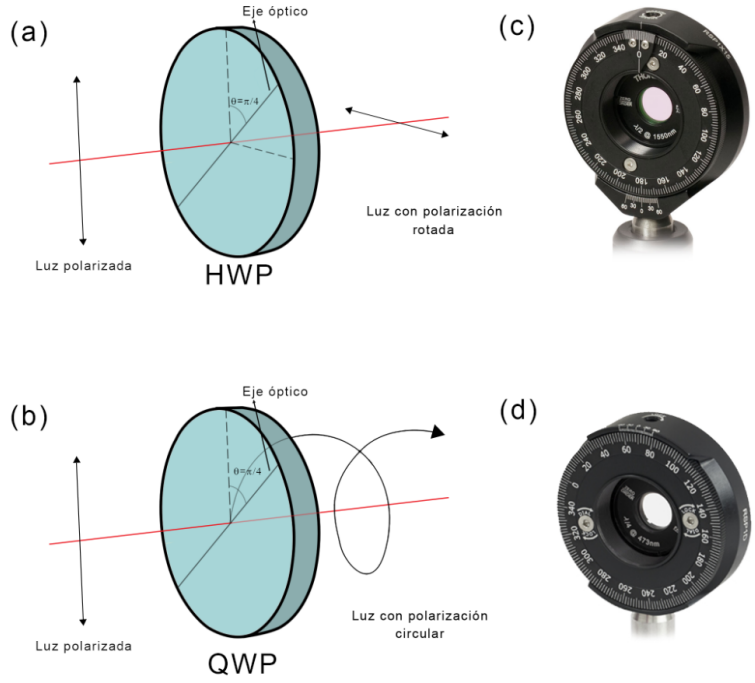


Figura 2.9: (a) Esquema del funcionamiento de una lámina de media onda, donde la polarización de la luz incidente se rota según el ángulo con el eje óptico. (b) Esquema del funcionamiento de una lámina de cuarto de onda, donde se genera polarización elíptica (en particular circular) a partir de una polarización lineal. (c) Foto real de una lámina de media onda en un montaje rotante (Thorlabs WPH05M-1550). (d) Foto real de una lámina de cuarto de onda en un montaje rotante (Thorlabs WPQSM05-473).

que implica 3 rotaciones consecutivas del estado en la esfera de Bloch. Primero se aplica una rotación en torno al eje y de $\frac{\pi}{4} - 2\alpha$, seguido de una rotación en torno al eje z de $4\beta - 2\alpha - 2\gamma$, y finalizando con otra rotación en torno al eje y de $2\gamma + \frac{3\pi}{2}$.

Manipulación de qubits

De la misma manera que para generar estados arbitrarios se aplican rotaciones sobre el estado del qubit, es posible demostrar que toda compuerta de un qubit se puede escribir como una rotación arbitraria y una fase global [5]. Por lo tanto, si el qubit está representado físicamente por el estado de polarización de un fotón, basta con utilizar nuevamente las QWP y HWP para realizar cualquier compuerta de un qubit.

De esta manera, las compuertas básicas de un qubit pueden escribirse en la forma de la ecuación (2.29) tales que:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = iR(\pi/2, \pi/4, \pi/2) = -i\hat{U}_{HWP}(\pi/4), \tag{2.30}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iR(\pi/2, \pi/4, \pi), \tag{2.31}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = -iR(\pi/4, \pi/2, \pi/4), \tag{2.32}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = i\hat{U}_{HWP}(\pi/8). \tag{2.33}$$

Medidas proyectivas

Realizar una medida proyectiva de un estado cuántico $|\phi\rangle$, requiere de realizar la proyección de ese estado sobre los estados usados como base, $\{|b_i\rangle\}$, y finalmente obtener la probabilidad de supervivencia del estado inicial sobre el estado de proyección, $P_{b_i}(\phi)$, dada por

$$P_{b_i}(\phi) = |\langle\phi|b_i\rangle|^2. \tag{2.34}$$

En el caso en que el estado cuántico es un estado de polarización, los dispositivos ópticos que pueden realizar tales proyecciones son los polarizadores lineales (sección 2.2.1) o los divisores de haz por polarización (PBS por sus siglas en inglés). Este último elemento transmite o refleja la luz incidente sobre una interfaz, según su estado de polarización, transmitiendo una dada polarización lineal, y reflejando la polarización ortogonal a esta. De esta manera, si se incide con un estado como el de la ecuación (2.2), y suponiendo que el PBS transmite la componente $|V\rangle$ y refleja $|H\rangle$, el fotón será reflejado con probabilidad $|\alpha|^2$ y transmitido con probabilidad $|\beta|^2$ (fig. 2.10). Si a la salida del PBS se colocan detectores que permiten contar fotones de a uno, la cantidad de cuentas en cada detector (C_1 y C_2) sobre la cantidad de cuentas totales (C_T) permite obtener una estimación de $|\alpha|^2$ y $|\beta|^2$, de modo que

$$|\alpha|^2 \approx \frac{C_1}{C_T} \quad , \quad |\beta|^2 \approx \frac{C_2}{C_T}. \tag{2.35}$$

Para poder medir estados en la base de polarización circular, se reescribe $|\phi\rangle$ en dicha base

$$|\phi\rangle = \alpha\,|H\rangle + \beta\,|V\rangle = \alpha'\,|R\rangle + \beta'\,|L\rangle\,, \tag{2.36}$$

con

$$\alpha' = \frac{\alpha + i\beta}{\sqrt{2}} \quad , \quad \beta' = \frac{\alpha - i\beta}{\sqrt{2}}, \tag{2.37}$$

entonces previo al PBS debo aplicarle una rotación al qubit de manera que obtenga el estado

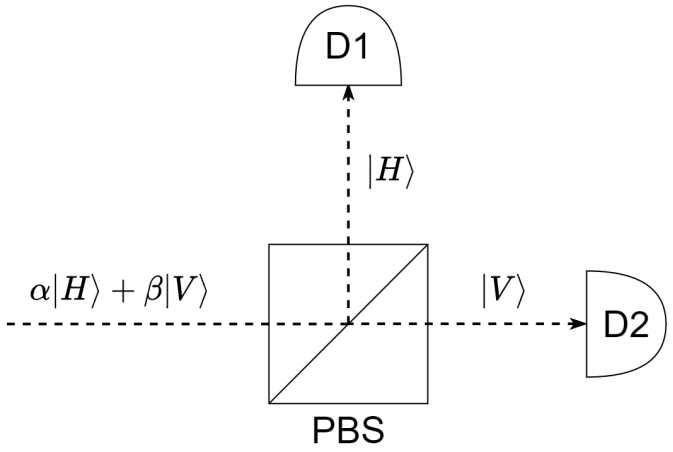


Figura 2.10: Esquema de medida de un estado arbitrario de un qubit codificado en polarización utilizando un divisor de haz por polarización, y dos módulos contadores de fotones.

$$|\phi'\rangle = \alpha' |H\rangle + \beta' |V\rangle , \tag{2.38}$$

sobre el cual puedo realizar una medida proyectiva sobre las polarizaciones lineales. Esto puede realizarse utilizando un arreglo de láminas de onda, tal que

$$\hat{U}_{QWP}(\pi/2)\hat{U}_{HWP}(\pi/2)\hat{U}_{QWP}(\pi/4)\,|\phi\rangle = \frac{1}{2\sqrt{2}} \begin{pmatrix} -1+i & 1+i \\ -1+i & -1-i \end{pmatrix} \begin{pmatrix} \alpha'+\beta' \\ i(\alpha'-\beta') \end{pmatrix} =$$

$$= \frac{-1+i}{\sqrt{2}} \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = e^{\frac{3i\pi}{4}} (\alpha' |H\rangle + \beta' |V\rangle), \tag{2.39}$$

donde la fase global puede ignorarse, y utilizando el PBS, se obtiene el estado $|H\rangle$ con probabilidad $|\alpha'|^2$, y el estado $|V\rangle$ con probabilidad $|\beta'|^2$.

Divisor de haz

El divisor de haz es una herramienta fundamental en experimentos de óptica cuántica, por lo que es necesario describirlo utilizando el formalismo de la mecánica cuántica. En la sección anterior se consideró un divisor de haz por polarización, pero también existen divisores de haces que transmiten o reflejan la luz incidente con cierta probabilidad, independientemente de su polarización. Un divisor de haz funciona como lo muestra la figura 2.11, donde los modos de salida 3 y 4 se expresan en función de los modos de entrada 1 y 2. Clásicamente, suponiendo que los modos 1 y 2 son ondas planas monocromáticas y polarizadas, es suficiente relacionar las amplitudes complejas en el punto de intersección, tal que

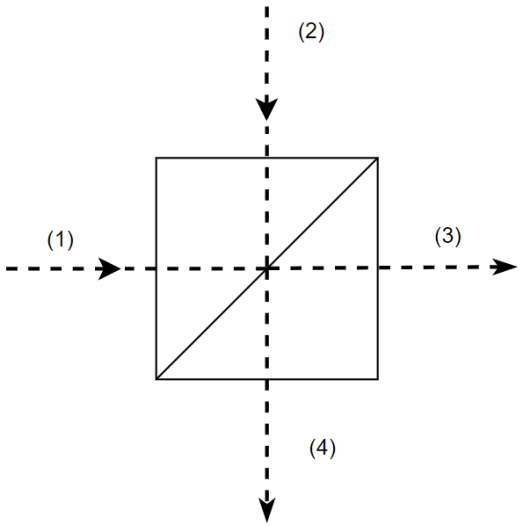


Figura 2.11: Esquema de funcionamiento de un divisor de haz, donde los modos de salida 3 y 4 se calculan en función de los modos de entrada 1 y 2.

$$\begin{aligned} E_3^{(+)} &= rE_1^{(+)} + tE_2^{(+)}, \\ E_4^{(+)} &= tE_1^{(+)} - rE_2^{(+)}, \end{aligned} \tag{2.40}$$

donde $E_i^{(+)}$ es la amplitud del campo eléctrico del modo i en la interfaz del divisor de haz, y t y r representan, respectivamente, la transmitancia y reflectancia del divisor de haz. Suponiendo un divisor de haz sin pérdidas, para garantizar la conservación de energía, debe cumplirse

$$|E_3^{(+)}|^2 + |E_4^{(+)}|^2 = |E_1^{(+)}|^2 \;\; \rightarrow \;\; t^2 + r^2 = 1. \tag{2.41}$$

La acción del divisor de haz puede describirse mediante la acción de una matriz S tal que

$$\begin{pmatrix} E_3^{(+)} \\ E_4^{(+)} \end{pmatrix} = S \begin{pmatrix} E_1^{(+)} \\ E_2^{(+)} \end{pmatrix} \quad \text{con} \;\; S = \begin{pmatrix} r & t \\ t & -r \end{pmatrix}. \tag{2.42}$$

En el caso cuántico, puesto que el divisor de haz no distingue polarización, los estados de entrada y salida se describen con un vector en la base de Fock, $|\psi_{1,2}\rangle$ y $|\psi_{3,4}\rangle$ respectivamente

$$|\psi_{1,2}\rangle = \sum_{i,j}^{\infty} \alpha_{i,j} \, |i\rangle_1 \, |j\rangle_2 \quad y \qquad |\psi_{3,4}\rangle = \sum_{i,j}^{\infty} \beta_{i,j} \, |i\rangle_3 \, |j\rangle_4 \,, \tag{2.43}$$

donde el estado $|i\rangle_1$ representa un estado de i fotones en el modo de entrada 1 (análogo para el modo de entrada 2). Estos estados deberán estar relacionados entre sí mediante una matriz unitaria U, tal que

$$|\psi_{3,4}\rangle = U\,|\psi_{1,2}\rangle\,. \tag{2.44}$$

Esta matriz U así definida es, en la práctica, imposible de calcular en la base de los estados de Fock [14]. Supongamos por ejemplo que ingresa el estado $|\psi_{1,2}\rangle = |n\rangle_1\,|0\rangle_2$, es decir, n fotones en el modo 1, y vacío en el modo 2. Por la conservación del número de fotones, el estado de salida debe tener la forma

$$|\psi_{3,4}\rangle = \alpha_0\,|0\rangle_3\,|n\rangle_4 + \alpha_1\,|1\rangle_3\,|n-1\rangle_4 + \cdots + \alpha_n\,|n\rangle_3\,|0\rangle_4\,, \tag{2.45}$$

es decir, $n+1$ coeficientes para describir la transformación de uno de los estados de entrada más simples posibles[6]. Por suerte, existe una manera de simplificar considerablemente estos cálculos. Supongamos que se quiere calcular el valor medio de un observable $\hat{O}_{3,4}$ en el espacio de salidas, es decir, se quiere obtener

$$\langle\hat{O}_{3,4}\rangle = \langle\psi_{3,4}|\hat{O}_{3,4}|\psi_{3,4}\rangle. \tag{2.46}$$

En general, el estado de salida $|\psi_{3,4}\rangle$ no se conoce, pero sí se conoce $|\psi_{1,2}\rangle$, donde ambos están relacionados por la ecuación (2.44). Esto me permite escribir el valor medio como

$$\langle\hat{O}_{3,4}\rangle = (\langle\psi_{1,2}|\,U^\dagger)\hat{O}_{3,4}(U\,|\psi_{1,2}\rangle) = \langle\psi_{1,2}|\,\hat{O}_{1,2}\,|\psi_{1,2}\rangle\,, \tag{2.47}$$

donde el operador $\hat{O}_{1,2}$ es la expresión del observable $\hat{O}_{3,4}$ en el espacio de entradas. En esta expresión, $\hat{U}$ es un operador unitario de un cuerpo bosónico, que lleva a una transformación unitaria S entre los operadores de destrucción en los modos 1 y 2 y los modos 3 y 4 [14], tal que

$$\begin{pmatrix} \hat{a}_3 \\ \hat{a}_4 \end{pmatrix} = S \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \end{pmatrix}, \tag{2.48}$$

donde el operador $\hat{a}_i$ destruye un fotón en el modo i, y la matriz de transformación S resulta ser igual que para el caso clásico, es decir,

$$S = \begin{pmatrix} r & t \\ t & -r \end{pmatrix}. \tag{2.49}$$

Solo es necesario escribir el operador $\hat{O}_{3,4}$ en función de los operadores $\hat{a}_3$ y $\hat{a}_4$, y mediante la ecuación (2.48) obtener la expresión de dicho operador en función de $\hat{a}_1$ y $\hat{a}_2$.

[6] Para describir la transformación de un estado de entrada general como el de la ecuación (2.43), permitiendo hasta n fotones en cada modo, se necesitan un total de $(n+1)^2$ coeficientes [14].

2.2.2. Fuentes de fotones

La implementación fotónica en las que se basan los algoritmos cuánticos y los protocolos de codificación y decodificación para QKD, supone la existencia de una fuente capaz de emitir fotones individuales a demanda[7]. Así mismo, el desarrollo matemático presentado hasta ahora es válido para estados de un fotón, por lo que, en la práctica, la generación de fotones individuales resulta ser una de las partes cruciales del sistema, dado que de ello depende, por ejemplo, la seguridad de los protocolos de QKD mencionados. Una fuente de fotones individuales ideal debe ser capaz de: emitir un único fotón con 100 % de probabilidad, en tiempos arbitrarios definidos por el usuario, y cada fotón emitido debe ser indistinguible del resto. Para poder definir un estado de un solo fotón utilizamos los estados de Fock, denotando $|n\rangle$ al estado de n fotones[8]. De esta manera, se busca una fuente cuya probabilidad de generar estados $|n \neq 1\rangle$ sea nula.

Existen distintos sistemas capaces de generar fotones individuales, que incluyen centros de color [22], puntos cuánticos [23], iones individuales [24], entre otros [25]. Cada uno de ellos presenta ventajas y desventajas al momento de ser utilizada como una fuente a demanda, y es un tema de investigación en crecimiento. En particular, en este trabajo se estudian las principales características de dos tipos de fuentes de fotones individuales: una basada en un láser de intensidad extremadamente baja, y la otra basada en pares de fotones generados por *conversión paramétrica descendente espontánea* (SPDC por sus siglas en inglés). La primera no es realmente una fuente de fotones individuales, y por eso se la suele mencionar como *pseudo-fuente de fotones individuales*, pero puede considerarse como tal para un número limitado de aplicaciones [25, 26].

Láser atenuado

Dado que, a fin de cuentas, un láser emite pulsos con una cierta cantidad de fotones, resulta lógico preguntarse si es posible atenuar lo suficiente un láser hasta obtener un único fotón por pulso. Para poder responder esta pregunta, es necesario estudiar la estadística de emisión de una fuente láser. Un láser posee una estadística *Poissoniana* [27], lo que se traduce a decir que al emitir un pulso de fotones, la probabilidad de que dicho pulso contenga n fotones, es decir, el estado $|n\rangle$, viene dada por una distribución de Poisson (fig. 2.12), tal que

$$P(N = n) = \frac{e^{-\langle N\rangle}\langle N\rangle^n}{n!}, \tag{2.50}$$

[7] 'a demanda' implica que un fotón puede ser emitido en cualquier tiempo arbitrario elegido por el usuario, es decir, la emisión es determinista.

[8] No confundir los estados de Fock $|0\rangle$ y $|1\rangle$ con los estados de la base computacional. Para eliminar ambigüedades, de aquí en más se aclarará qué estados se están utilizando.

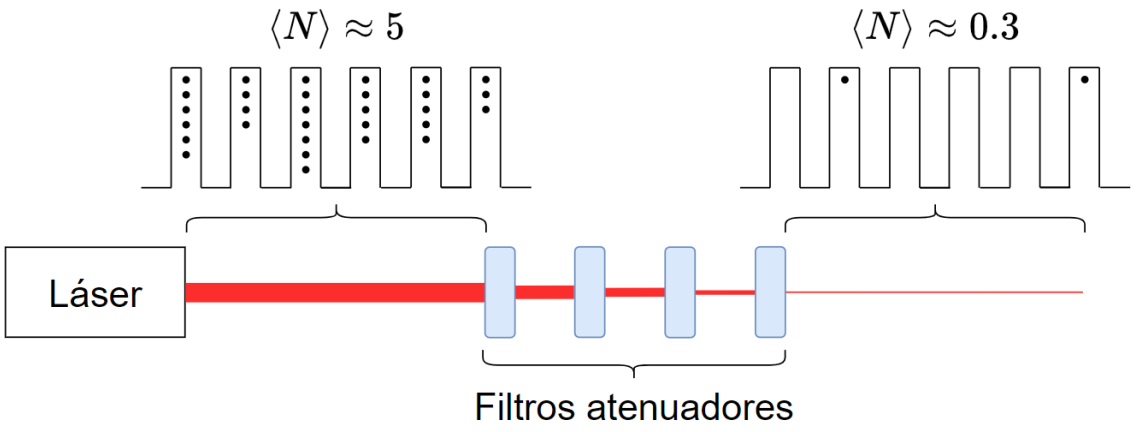


Figura 2.12: Esquema de atenuación de un láser con un conjunto de filtros atenuadores. Previo a los atenuadores, cada pulso láser es emitido con un número aleatorio de fotones, con una media de 5 fotones por pulso. Luego de los filtros, la gran mayoría de los pulsos se encuentran vacíos (con cero fotones), algunos pulsos contienene un fotón, y en menor medida, algunos contienen más de un fotón.

donde $\langle N \rangle$ es la media de fotones emitidos por pulso, relacionada con la potencia de salida del láser. Supongamos por ejemplo que logramos atenuar la potencia del láser (utilizando por ejemplo filtros de intensidad a la salida) lo suficiente como para que, por ejemplo, $\langle N \rangle = 1$, y calculamos las probabilidades de emisión de N fotones por pulso, según la ecuación (2.50), obtenemos

$$\begin{aligned} P(N=0) &\sim 36{,}8\,\%, \\ P(N=1) &\sim 36{,}8\,\%, \\ P(N>1) &\sim 26{,}4\,\%. \end{aligned}$$

Entonces, lo primero que se observa es que, probabilísticamente, gran parte de los pulsos serán pulsos vacíos, es decir, con 0 fotones; por otro lado, la probabilidad de emitir más de un fotón es no nula, y más aún, es no despreciable respecto a la probabilidad de 1 fotón por pulso. Los pulsos vacíos pueden ser descartados post medición, ya que simplemente se pueden conservar todos aquellos que resultaron en una cuenta en el detector. Sin embargo, los pulsos con más de un fotón no pueden distinguirse de los de un fotón mediante el tipo usual de detectores, conocidos como detectores *on-off*[9]. A las fuentes como éstas, donde existe una probabilidad no nula de emitir más de un fotón, se las llama pseudo-fuentes de fotones individuales.

Para ciertas aplicaciones, una pseudo-fuente de fotones individuales es suficiente. Sin embargo, no siempre resulta serlo para aplicaciones de criptografía cuántica [29]. En particular, en el caso del protocolo BB84 de QKD[10] (sección 2.1.3), enviar más

[9]Este tipo de detectores operan en modo Geiger [28], por lo que su salida será un pulso lógico, que es un '0' si no se detecta ningún fotón, o un '1' si se detectan uno o más fotones.

[10]Existe una variante al protocolo BB84, conocido como *Decoy state*, que permite trabajar con

de un fotón por pulso por el canal cuántico permitiría que el espía divida el pulso mediante un divisor de haz, quedándose con n de los N fotones, y reenviando los $N-n$ restantes. Con la ayuda de una memoria cuántica, podría almacenar el estado hasta el momento en que Alice y Bob revelan las bases de medidas y los pulsos útiles para la clave, y entonces medir en la misma base para obtener la clave secreta. De esta manera, un espía puede obtener el estado exacto de cada qubit sin violar el teorema de no clonación, y además como los fotones reenviados no son medidos, su estado no se ve perturbado, por lo que Alice y Bob serán incapaces de determinar la presencia del espía.

Conversión paramétrica espontánea descendente

Uno de los tipos de fuentes más utilizadas para generar fotones individuales son las basadas en el fenómeno de conversión paramétrica espontánea descendente (SPDC por sus siglas en inglés), un fenómeno no lineal de segundo orden en el que un fotón de cierta energía, llamado fotón de bombeo (usualmente llamado *pump*, como en inglés) incide sobre un cristal birrefringente, y con cierta probabilidad (conservando energía y momento lineal) se generan dos fotones de menor energía, llamados señal y heraldo (*signal* e *idler* en inglés respectivamente) (fig. 2.13). Para entender este fenómeno, es necesario primero utilizar el formalismo de la mecánica cuántica para explicar el electromagnetismo, es decir, cuantificar el campo electromagnético. La idea detrás de esta cuantificación es resolver las ecuaciones de Maxwell para el potencial vector $\mathbf{A}$, utilizando el gauge de Coulomb $\vec{\nabla}\cdot\mathbf{A}=0$. Al hacer eso, se encuentra que las ecuaciones para $\mathbf{A}$ son similares a un oscilador armónico, permitiendo hacer el mismo tratamiento para encontrar las variables canónicas conjugadas. De esta manera, se obtiene que

$$\hat{A}(\mathbf{r}) = \sum_l \boldsymbol{\epsilon}_l \sqrt{\frac{\hbar}{2\omega_l V \epsilon_0}} \left(e^{i\mathbf{k}_l\cdot\mathbf{r}}\hat{a}_l + e^{-i\mathbf{k}_l\cdot\mathbf{r}}\hat{a}_l^\dagger \right) = \hat{A}^{(+)}(\mathbf{r}) + \hat{A}^{(-)}(\mathbf{r}) \tag{2.51}$$

donde el índice l representa un modo óptico, y tenemos dos operadores cuánticos $\hat{a}_l$ y $\hat{a}_l^\dagger$, donde $\hat{a}_l$, el operador de aniquilación de fotones, destruye un fotón en el modo l, y $\hat{a}_l^\dagger$, el operador de creación de fotones, crea un fotón en el modo l. De la misma manera, $\boldsymbol{\epsilon}_l$ es el vector de polarización de los fotones en el modo l, ω_l es la frecuencia del modo l, y V es el volumen de cuantización. Como los operadores $\hat{a}_l$ y $\hat{a}_l^\dagger$ son equivalentes a los operadores de creación y destrucción del oscilador armónico, también cumplen que

$$\hat{a}_l^\dagger \left|n_l\right\rangle = \sqrt{n_l+1}\left|n_l+1\right\rangle, \tag{2.52}$$

pseudo-fuentes de fotones individuales con control de seguridad del canal, enviando aleatoriamente pulsos que no contienen información de la clave y con un número de fotones distinto al de los pulsos en los que se codifica la clave, que son usados para detectar la presencia de un espía a partir de la inspección de la atenuación de la señal recibida por Bob [30].

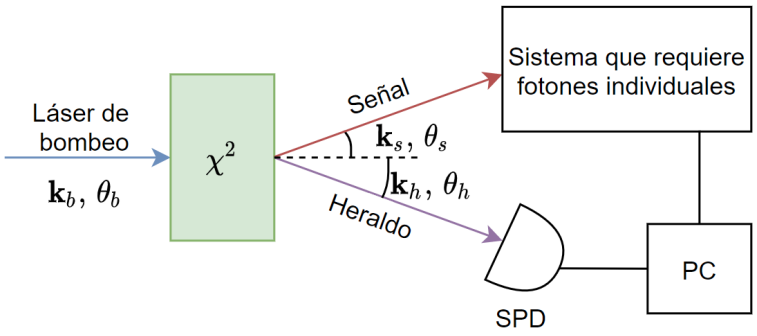


Figura 2.13: Esquema de generación de fotones individuales utilizando conversión paramétrica descendente espontánea. En caso de detectar el fotón heraldo, la PC le indica al sistema que necesita fotones individuales la existencia del fotón señal. La conservación de energía y momento garantizan que $\mathbf{k}_b = \mathbf{k}_h + \mathbf{k}_s$ y $\theta_b = \theta_s - \theta_h$.

$$\hat{a}_l \left|n_l\right\rangle = \sqrt{n_l} \left|n_l - 1\right\rangle , \tag{2.53}$$

donde los estados se encuentran en la base de Fock.

Con este formalismo, es posible explicar la creación espontánea de un fotón de una cierta energía en un dado modo óptico. Es así que un fotón de alta energía puede destruirse, creando en el proceso dos, o más, fotones de menor energía al interactuar con el medio no lineal. La interacción con el medio puede describirse con un Hamiltoniano efectivo tal que

$$\hat{H}_{SPDC} = i\hbar\kappa\, \hat{a}_h^\dagger \hat{a}_s^\dagger \hat{a}_b e^{-i\Delta\mathbf{k}\cdot\mathbf{r}+i\Delta\omega t} + h.c., \tag{2.54}$$

donde $h.c.$ representa el hermítico conjugado[11] (necesario para que $\hat{H}_{SPDC}$ sea hermítico), los índices h, s, b representan los fotones heraldo, señal y bombeo respectivamente, $\Delta\omega = \omega_p - \omega_h - \omega_s$, y κ es una constante de la forma

$$\kappa = \frac{2}{3}\frac{d_{\text{eff}}}{\epsilon_0 V}\sqrt{\frac{\omega_p\omega_h\omega_s}{2\epsilon_0 V}}, \tag{2.55}$$

donde V es el volumen de cuantización y d_{eff} es el índice no lineal efectivo. De esta manera se representa el fenómeno de SPDC en el formalismo de la mecánica cuántica, donde se destruye un fotón de frecuencia ω_p y se crean dos fotones de frecuencias ω_s y ω_h respectivamente.

Si inciden n fotones al medio no lineal, tal que, si se expresa en la base de Fock se tiene inicialmente el estado $|0_s, 0_h, n_b\rangle$. Al reemplazarlo en la ecuación de Schrödinger

[11] Este término representa el fenómeno de generación de suma de frecuencias (SFG por sus siglas en inglés), que en el caso particular $\omega_h = \omega_s$ y $\omega_p = 2\omega_s$ se reduce a generación de segundo armónico (SHG por sus siglas en inglés)

con el Hamiltoniano de interacción, se obtiene que el estado resultante será

$$|\psi(t)\rangle = e^{\frac{-i}{\hbar}\int_0^t \hat{H}_{SPDC}(t')dt'} |0_s, 0_h, n_b\rangle \,. \tag{2.56}$$

Si ahora se realiza una expansión de Taylor sobre la exponencial, se obtiene

$$|\psi(t)\rangle = C_0 \,|0_s, 0_h, n_b\rangle - C_1 \frac{i}{\hbar} \int_0^t \hat{H}_{SPDC}(t')dt' \,|0_s, 0_h, n_b\rangle + \ldots, \tag{2.57}$$

donde $C_0, C_1, \ldots$ son coeficientes que normalizan el estado. Si se aproxima a primer orden y solo se tienen en cuenta los casos en los que la energía se conserva, es decir, $\Delta\omega \approx 0$, el término dentro de la integral resulta en la función delta de Dirac $\delta(t)$, y se puede escribir el estado de la ecuación (2.57) como

$$|\psi(t)\rangle \approx C_0 \,|0_s, 0_h, n_b\rangle - C_1 \kappa e^{-i\Delta\mathbf{k}\cdot\mathbf{r}} \,|1_s, 1_h, n_b - 1\rangle \,. \tag{2.58}$$

Teniendo en cuenta esto, la probabilidad de generar el estado $|1_s, 1_h, n_b - 1\rangle$ es proporcional a κ^2, y agregando términos extra se encuentra que la probabilidad de generar el estado $|2_s, 2_h, n_b - 2\rangle$ es proporcional a $\kappa^4/4$, por lo que solo un pequeño porcentaje de los fotones incidentes serán transformados en un par de fotones convertidos, y mucho menor aún será la probabilidad de generar dos pares de fotones[12]. Estas probabilidades hacen de SPDC un proceso con eficiencia extremadamente baja, siendo una de las implementaciones más eficientes capaz de generar un par de fotones por cada $2{,}5 \times 10^5$ fotones incidentes [32].

Utilizando este fenómeno, es posible crear una fuente de fotones individuales aprovechando el hecho que los fotones se crean de a pares. Dado que además de conservarse la energía se debe conservar el impulso lineal total, es decir

$$\mathbf{k}_b = \mathbf{k}_s + \mathbf{k}_h \;\; \text{, con} \quad \Delta\mathbf{k} = \mathbf{k}_b - \mathbf{k}_s - \mathbf{k}_h = 0, \tag{2.59}$$

entonces conociendo $\mathbf{k}_b$, quedan determinadas las direcciones en las que se emite cada fotón. Así, utilizando un sistema como el de la figura 2.13, la detección de un fotón en la dirección $\mathbf{k}_h$ con energía $\omega_h\hbar$ (el heraldo), indica la existencia de un fotón en la dirección $\mathbf{k}_s$ con energía $\omega_s\hbar$ (la señal) que puede utilizarse en lo que se requiera[13]. A esto se le debe agregar una etapa final de correlación temporal, para garantizar que ambos fotones se crearon simultáneamente, es decir, se debe realizar lo que se conoce

[12] Para un bombeo de $\lambda_b = 400$ nm ingresando a un cristal BBO preparado para SPDC tipo II, que genera dos fotones del doble de longitud de onda, la probabilidad de generar un par de fotones convertidos es del orden de 10^{-14} por fotón incidente [31].

[13] También existe la posibilidad que en el proceso de SPDC se generen dos fotones en el pulso heraldo y señal, y no uno en cada uno. Sin embargo, la probabilidad de este proceso es mucho menor comparada con generar solo un par, y para las intensidades en las que se trabaja puede considerarse despreciable.

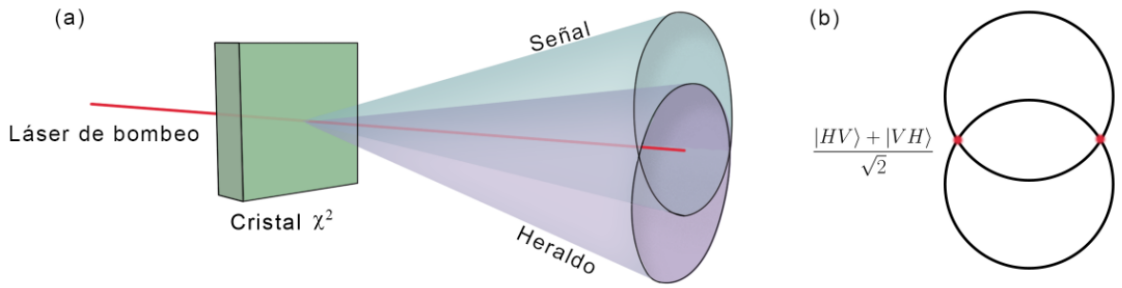


Figura 2.14: (a) Esquema de los conos que limitan la trayectoria de los fotones convertidos por SPDC. (b) Vista de frente de los dos conos, en rojo se marcan los puntos en los cuales se encuentran los estados entrelazados.

como *medida en coincidencia*, que requiere de una etapa de análisis de señales, y es uno de los puntos centrales de este trabajo.

Además de poder crear una fuente de fotones individuales, este fenómeno permite también crear una fuente de fotones entrelazados en polarización. Debido a las condiciones de conservación mencionadas anteriormente, todos los fotones generados por SPDC de una dada frecuencia tienen sus trayectorias limitadas a la superficie de un cono (fig. 2.14a). Un tipo particular de conversión, conocido como tipo II [33], hace que el par de fotones convertidos tengan polarizaciones ortogonales entre sí, por ejemplo $|H\rangle$ para el heraldo y $|V\rangle$ para la señal, o viceversa. Suponiendo que el resto de grados de libertad sean indistinguibles para ambos fotones, en la intersección de los conos se encuentra el estado

$$|\psi\rangle = \frac{|HV\rangle + |VH\rangle}{\sqrt{2}}, \tag{2.60}$$

que es un estado máximamente entrelazado (fig. 2.14b), equivalente al estado de Bell $|\beta_{01}\rangle$. Utilizando este estado, es posible generar el resto de los estados de Bell aplicando las compuertas definidas en la sección 2.2.1 sobre alguno de los qubits, por ejemplo

$$|\beta_{00}\rangle = (\hat{X} \otimes \hat{I})\,|\beta_{00}\rangle = (\hat{I} \otimes \hat{X})\,|\beta_{00}\rangle\,,$$

$$|\beta_{10}\rangle = (\hat{Z} \otimes \hat{X})\,|\beta_{00}\rangle = (\hat{X} \otimes \hat{Z})\,|\beta_{00}\rangle\,,$$

$$|\beta_{11}\rangle = (\hat{Z} \otimes \hat{I})\,|\beta_{00}\rangle = (\hat{I} \otimes \hat{Z})\,|\beta_{00}\rangle\,.$$

Es importante destacar que la caracterización de cualquiera de estos estados requiere de mediciones en coincidencias, dado que individualmente el estado de polarización de cada fotón está completamente indeterminado.

Capítulo 3

Diseño del módulo contador de coincidencias

3.1. Detección en coincidencia

Se denomina detección en coincidencia a la detección simultanea de dos o más eventos. Se dice que dos o más eventos *coinciden* si son detectados en alguna dada ventana temporal, que suele ser menor a la resolución del detector. Esta técnica es ampliamente utilizada, por ejemplo, en experimentos de decaimiento radioactivo, pero cumple un rol fundamental en información cuántica, y en particular en óptica cuántica. La detección en coincidencia de fotones es una herramienta fundamental que permite determinar características no clásicas de las fuentes de luz. En muchos de estos experimentos, es suficiente con detectar coincidencias dobles (solo dos eventos), pero existen otros para los que se necesita detectar coincidencias de mayor orden [34, 35].

3.2. Contador de coincidencias

Se buscó diseñar un sistema capaz de contar la cantidad de veces que un conjunto de señales de entrada coinciden en una ventana de tiempo dada. El sistema propuesto fue basado en los trabajos [36–38], y su diagrama de bloques se muestra en la figura 3.1. El funcionamiento del mismo puede dividirse en los siguientes pasos:

1. Las entradas, que pueden provenir de un detector o un generador de señales, ingresan a un conformador de pulsos, que modifica los pulsos de entrada a un aspecto rectangular en el dominio del tiempo, con ancho y amplitud configurables.

2. Los pulsos conformados ingresan a un detector de señales, que emite un pulso cuando se detecta una coincidencia entre un subconjunto de los canales de entrada. Así, agrupando entradas se disponen de M configuraciones de coincidencias,

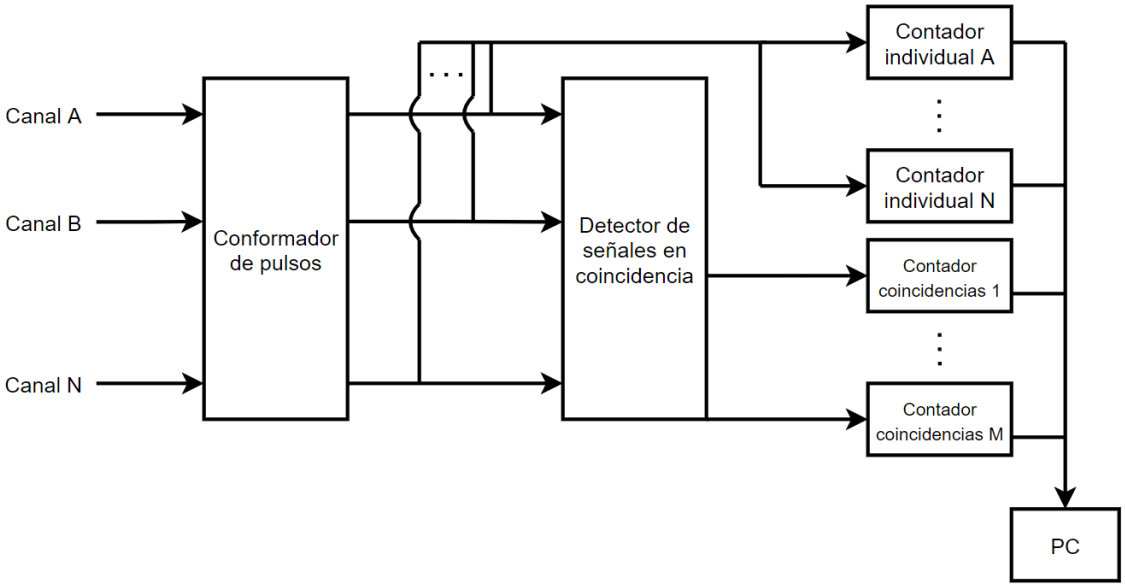


Figura 3.1: Diagrama simplificado del funcionamiento de un módulo contador de coincidencias. Las N señales ingresan por los canales de entrada, son retardadas individualmente, y los pulsos son conformados a un mismo ancho configurable. Estos pulsos se dirigen al detector de señales en coincidencia, que emite pulsos por M salidas al detectar coincidencias entre las señales configuradas para cada una. Los M pulsos de salida, y los N pulsos de entrada conformados son contados en una cantidad dada ventana temporal, y la información se envía hacia una PC.

dando lugar a M salidas, cada una con una configuración asociada.

3. El conteo de pulsos ocurre dentro del lapso de una ventana temporal programable, contando por un lado los pulsos conformados de los N canales, y por otro los pulsos de las M salidas en coincidencia. Esta información es luego enviada a la PC.

3.3. Materiales y métodos

Para implementar el módulo contador de coincidencias, se utilizó una placa para desarrollo de prototipos FPGA (Arreglo de compuertas programables por campos, o *Field Programmable Gate Array* en inglés). Las placas FPGA contienen un arreglo tridimensional de bloques lógicos que pueden ser 'conectados' entre sí, permitiendo que se comuniquen. El contenido de estos bloques puede ser configurado mediante una PC, utilizando un software apropiado conocido en general como entorno integrado para simulación y desarrollo (ISE o IDE por sus siglas en inglés), para realizar desde compuertas lógicas simples (AND, OR, XOR, etc.) hasta funciones más complejas, permitiendo crear circuitos lógicos compuestos de alta complejidad. Entre las características más atractivas de estos dispositivos, se encuentra la capacidad de reprogramar su contenido, permitiendo agregar, remover, o modificar componentes del diseño con facilidad.

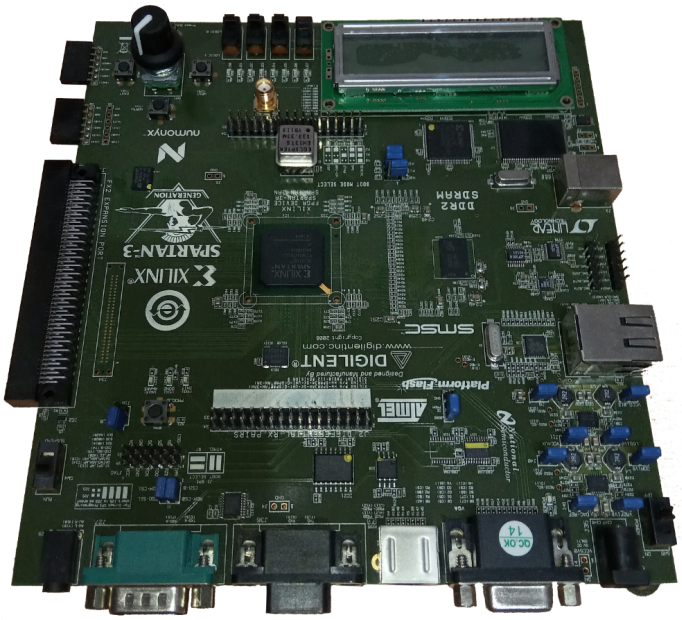


Figura 3.2: Foto real de una placa de desarrollo de prototipos FPGA (Xilinx Spartan-3A XCS3S700A).

3.3.1. Placa de desarrollo de prototipos

Se utilizó una placa modelo Spartan-3A XC3S700A fabricada por Xilinx (fig. 3.2), una de las dos empresas líder en la industria [39]. Los 5 dispositivos pertenecientes a la familia Spartan-3A se caracterizan por su alta cantidad de puertos I/O (de 108 a 502), compatibilidad con 26 estándares de comunicación I/O, la presencia de un modo *suspendido* que permite ahorrar energía de forma flexible y efectiva, y un robusto y económico mecanismo para proteger el dispositivo de ingeniería inversa y clonación [40]. En particular, el modelo XC3S700A posee un reloj de 50 MHz, 700K compuertas lógicas (que equivalen a 13,248 bloques lógicos), 372 puertos I/O, y 360 kb de almacenamiento en memoria RAM.

La arquitectura de la familia Spartan-3A consiste en cinco elementos funcionales fundamentales, que se organizan como muestra la figura 3.3:

- Bloques lógicos configurables (CLBs): Contienen *lookup tables* (LUTs) que permiten tanto implementar lógica como almacenar información en forma de flip-flops o latches.
- Bloques Input/Output (IOBs): Controlan el flujo de datos entre los pines I/O y la lógica interna del dispositivo. Son compatibles con una gran variedad de estándares de señales.
- Bloques RAM: Permiten almacenar información en la forma de bloques de 18 kb.
- Bloques multiplicadores: Permiten realizar el producto entre dos números binarios

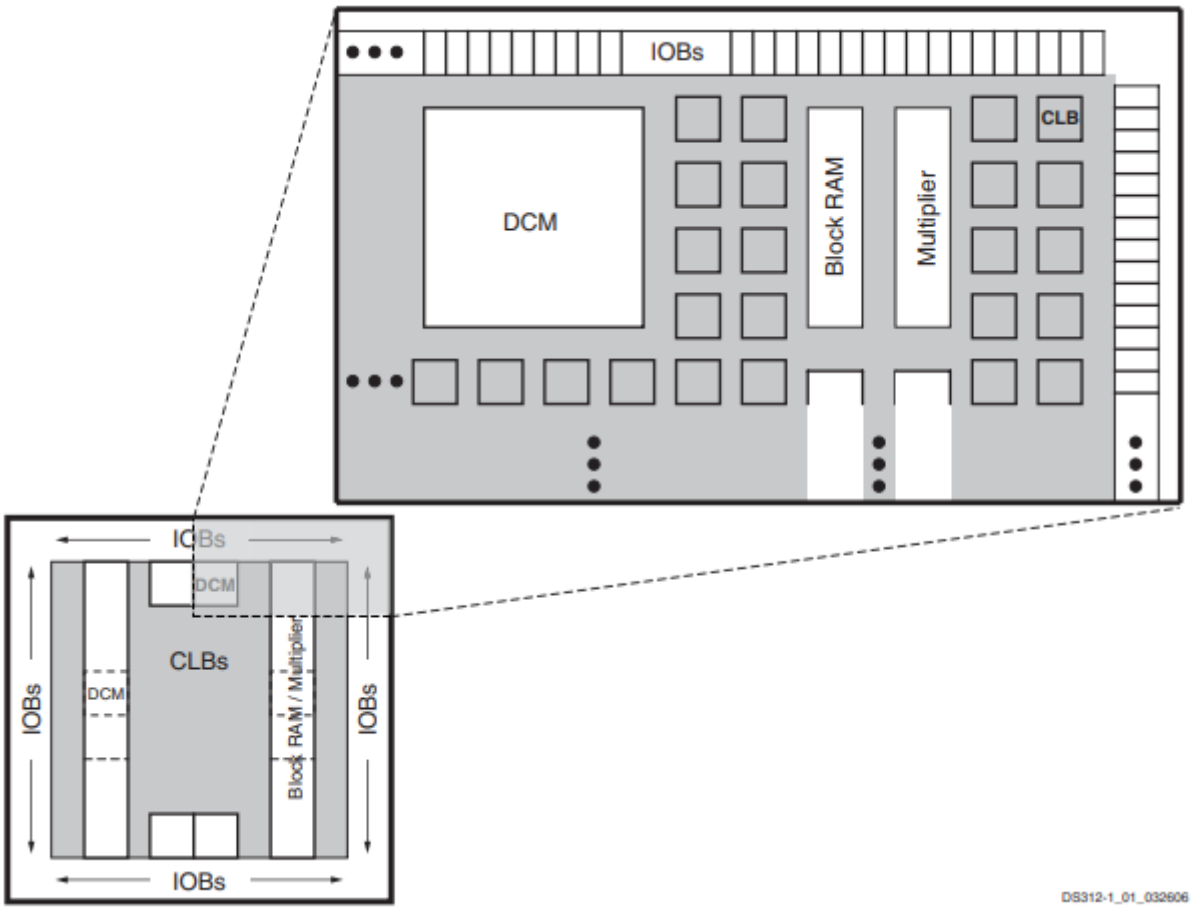


Figura 3.3: Distribución de los bloques fundamentales para la familia de FPGAs Spartan-3A. Los dispositivos poseen dos columnas de bloques RAM, donde cada columna contiene varios bloques RAM de 18 kb. Los DCMs se ubican en el centro, estando dos en la parte superior del dispositivo, y dos en la parte inferior. En particular, el modelo XC3S700A agrega dos DCMs entre las dos columnas de bloques RAM.

de hasta 18 bits.

- Bloques administradores de reloj digital (DCMs): Permiten distribuir, retardar, multiplicar, dividir y aplicar corrimientos de fase a las señales de reloj, de manera completamente digital y auto calibrable.

Para configurar este dispositivo, se utilizaron dos métodos. El primero de ellos consiste en programar los datos de configuración en una memoria PROM externa, que mantiene su contenido al apagar el dispositivo y pueden programarse en la FPGA automáticamente en el encendido, o al presionar un botón. El segundo tiene un carácter menos permanente, y consiste en programar directamente en la FPGA mediante el software de la PC. Este último método tiene como ventaja que permite cambiar la configuración con mayor agilidad, lo que facilita las etapas de desarrollo y depuración, pero también tiene la desventaja que la información se pierde por completo al apagar el dispositivo.

3.3.2. Entorno integrado de desarrollo

Para configurar la FPGA, se utilizó la versión compatible con la familia Spartan-3A del software ISE Design Suite, de Xilinx. Este software permite escribir programas en lenguajes de descripción de hardware (HDL por sus siglas en inglés), tales como VHDL y Verilog, que luego son traducidos al circuito equivalente para su implementación en FPGA. El archivo principal que contiene toda la información recibe el nombre de *proyecto*, y cada archivo en los que se escribe código recibe el nombre de *módulo*.

El software controla todos los aspectos del diseño, síntesis e implementación del circuito. Desde la interfaz del *Project Navigator* (fig. 3.4), es posible acceder a todos los diseños realizados y a las herramientas de implementación, así como todos los archivos y documentos asociados con el proyecto. La interfaz separa la información a mostrar en un número de paneles, entre los más notables se encuentran:

- Panel de vista: Permite seleccionar entre explorar los archivos de los módulos asociados con la síntesis o los asociados con la simulación en el panel de jerarquía.
- Panel de jerarquía: Muestra el nombre de proyecto, el dispositivo FPGA utilizado, los documentos de usuario, y los archivos asociados con lo seleccionado en el panel de vista.
- Panel de procesos: Su contenido cambia según el tipo de archivo seleccionado, y permite acceder a todas las funciones necesarias para definir, ejecutar, y analizar un diseño. Entre sus funciones para un módulo HDL, permite ver el esquema RTL del diseño, revisar si el diseño es sintetizable en FPGA, e implementarlo si es lo que se busca.
- Espacio de trabajo: Permite editar los archivos de diseño, visores, y herramientas de diseño. Esto incluye el editor de texto de ISE, el visor de esquemas, los visor de RTL y tecnología del diseño, etc.
- Panel de consola: Muestra todos los resultados de ejecutar procesos desde el *Project Navigator*. Muestra errores, advertencias, y mensajes del proceso de depuración.

Como fue mencionado, cada parte del sistema se realizó en un archivo de módulo separado, que luego fue agregado como componente de un módulo integrador.

Finalmente, como herramienta de pruebas, el software incluye el paquete *ISE Simulator* (ISim), que permite simular el comportamiento del diseño antes de ser implementado.

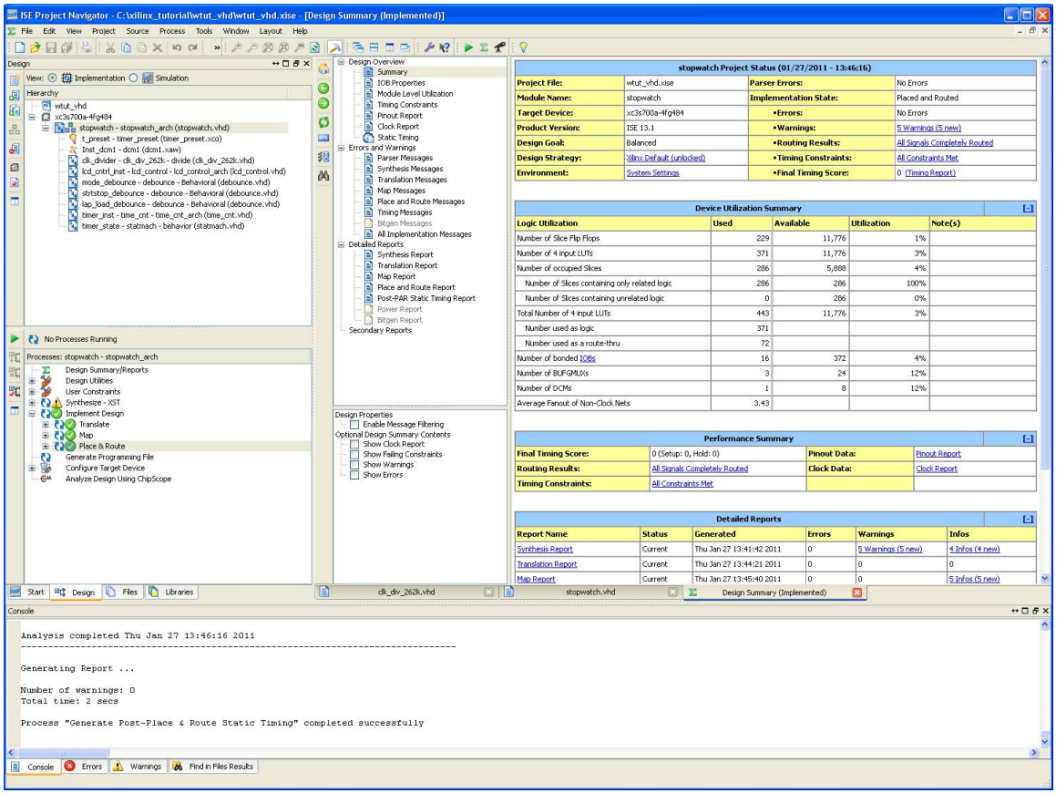


Figura 3.4: Interfaz del *Project Navigator* del software Xilinx ISE. En la parte izquierda, los paneles de vista, jerarquía y procesos, en la derecha el editor de archivos, y en la parte inferior, el panel de consola.

3.4. Implementación en FPGA

Para facilitar la implementación, se dividió el circuito completo en módulos individuales o componentes, que luego se conectaron en un código integrador, obteniendo el circuito total buscado. Los módulos realizados fueron un retardador, un conformador de pulsos, un detector de señales en coincidencia, y un contador. Se implementó el sistema para 4 canales de entrada (A, B, C y D) y 8 señales de salida. Todos los archivos de código se encuentran en [41], requiriendo petición de acceso.

3.4.1. Retardador

El módulo retardador es el encargado de retardar una señal, de manera de poder compensar o generar defasajes temporales en las señales de entrada. Esto se vuelve necesario cuando las señales que arriban al circuito han experimentado una diferencia de caminos y se desea compensarla, o cuando se busca que exista un defasaje particular entre ellas, tal como se necesita para realizar el experimento de Hanbury-Brown y Twiss [42]. La implementación de los retardos en FPGA puede realizarse mediante recursos analógicos, con una cadena de compuertas AND (empleada para ajustes finos), o digitales, utilizando cadenas de flip-flops disparados secuencialmente por flancos de

la señal de reloj (para ajustes gruesos).

La cadena de compuertas AND provoca un retardo resultante, igual a la suma de los retardos individuales inherentes a toda implementación circuital y que depende de parámetros de diseño geométricos y tecnológicos. Estos retardos individuales producidos por cada compuerta, para aplicaciones generales son considerados despreciables, pero encuentran aplicaciones en conteo de fotones. Utilizando un osciloscopio digital RIGOL DS1102E, se midió el retardo producido por una compuerta AND sobre una señal, y se encontró un retardo de ~ 1 ns (fig. 3.5), que coincide con los valores reportados en la bibliografía [43]. Como en general se busca retardar señales hasta decenas de nanosegundos, es necesario sumar decenas de compuertas AND, siendo una estrategia poco económica en cuanto al uso del espacio disponible en la FPGA. Teniendo en cuenta esto último, se decidió implementar los retardos utilizando la señal del reloj. Con este último, en cada flanco de subida del reloj, el circuito revisa el estado de la señal de entrada (0 o 1) y lo almacena en una memoria, que contiene hasta 255 elementos. Para configurar un retardo particular, simplemente se elige la salida como la componente correspondiente al retardo buscado dentro de la memoria, por ejemplo, sin retardo sería la componente 0, un retardo de 20 ns sería la componente 1, y así sucesivamente.

Esta implementación utilizando la señal de reloj no es viable para una aplicación real, debido a que la velocidad máxima del reloj (50 MHz) solo permite aplicar retardos que sean múltiplos de 20 ns. Aún teniendo esa limitación en el mínimo tiempo de retardo aplicable, existe otra limitación ligada a aspectos estructurales del circuito, al tener en cuenta que la lectura de señales a retardar se realiza en los flancos de subida de la señal de reloj. Al ingresar una señal real, los flancos de subida de la señal de entrada y del reloj no estarán necesariamente sincronizados, dando lugar a tiempos de retardos no controlables con precisión (fig. 3.6).

3.4.2. Conformador de pulsos

El módulo conformador de pulsos se encarga de reducir el ancho de un pulso entrante a uno configurable. Para realizar esto, la señal es retardada una cantidad de tiempo configurable Esta señal es luego invertida, y se realiza la operación AND entre esta señal y la original (fig. 3.7). Para implementar el retardo de la señal, con resoluciones de hasta algunos nanosegundos, se utilizó una cadena de compuertas AND como la propuesta en la sección 3.4.1. La máxima longitud de esta cadena fue de 15 compuertas, donde la salida de cada una de las compuertas se conecta a la entrada de un multiplexor (MPX), el cual permite seleccionar sólo una entrada a través de 4 lineas de comando que aseguran 16 combinaciones (fig. 3.7). De esta manera, esta combinación representa un número binario, el cual será igual a la cantidad de compuertas involucradas en la cadena de retardos.

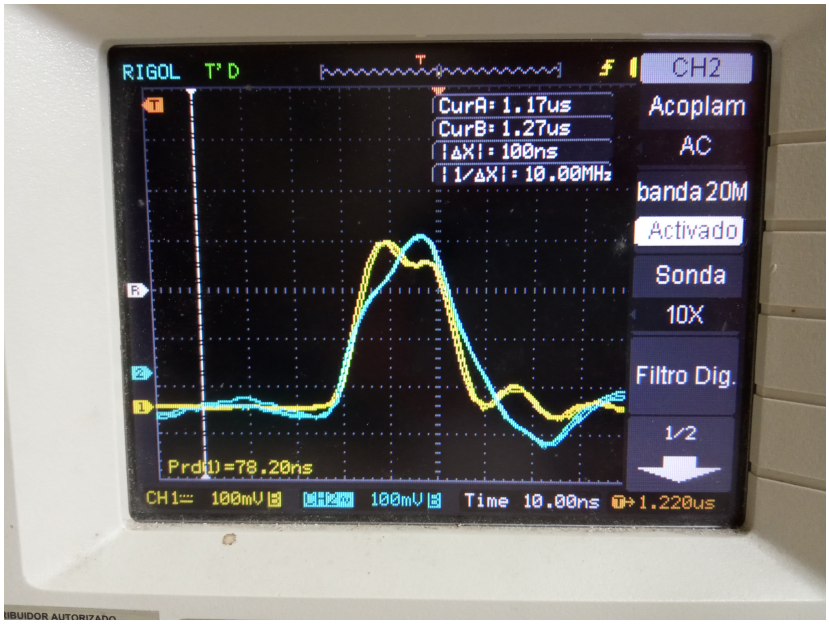


Figura 3.5: Medida del retardo producido por una compuerta AND sobre una señal. La señal amarilla corresponde a un pulso cuadrado de 20 ns generado en la FPGA, mientras que la azul corresponde a esa misma señal luego de pasar por una compuerta AND.

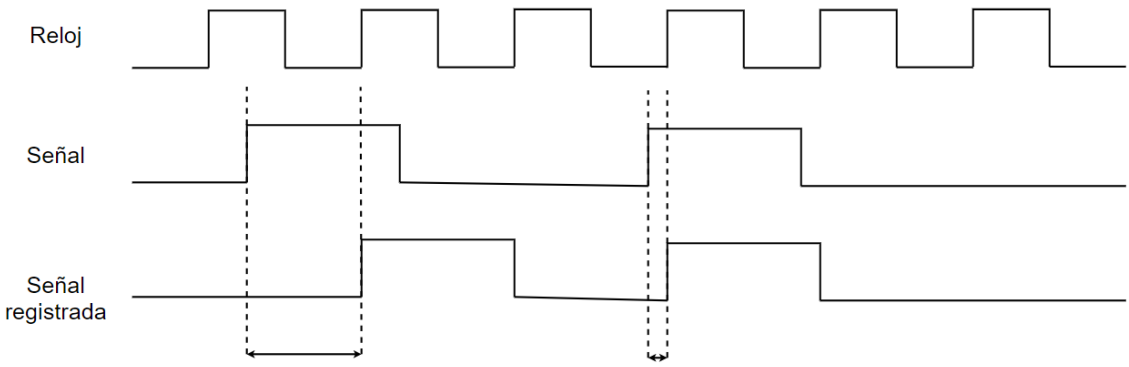


Figura 3.6: Representación de la señal registrada por la memoria al ingresar una señal real, no sincronizada con el reloj. Como el registro ocurre en cada flanco de subida de reloj, se ve que ambos pulsos de entrada son retardados en cantidades distintas, y no controlables, de tiempo.

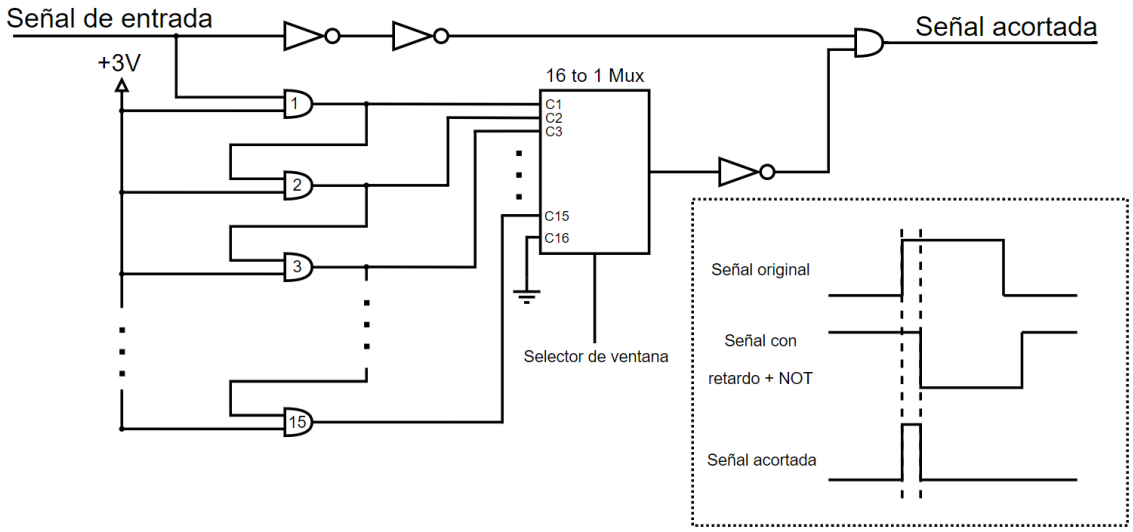


Figura 3.7: Circuito implementado para conformar un pulso de entrada a un ancho configurable. La señal es llevada a una cadena de compuertas AND, donde las entradas a la compuerta i serán una señal de 3V, y la señal saliente de la compuerta $i-1$, donde para la compuerta 1 es la señal original. Las salidas de las 15 compuertas, junto con una señal conectada a tierra, son enviadas a un multiplexor 16 a 1, donde la salida es elegida por una señal de 4 bits. A la señal saliente del multiplexor se le aplica una compuerta NOT, e ingresa a una compuerta AND junto con la señal original luego de aplicársele dos compuertas NOT. Inserto: Funcionamiento del circuito sobre un pulso dado. La señal original es retardada y negada, y al ingresar junto con la señal original a una compuerta AND, se obtiene un pulso con el mismo flanco de subida que la señal original, pero acortado.

3.4.3. Detector de señales en coincidencia

El módulo detector de señales en coincidencia recibe un número de señales, y decide si un conjunto de ellas (seleccionadas por el usuario), se consideran efectivamente coincidencias. Para determinar si dos señales coinciden, se usan ambas señales como entradas a una compuerta AND, obteniendo un pulso saliente en el caso que ambas señales se superpongan. Teniendo en cuenta el método utilizado, se implementó un método para 'apagar' las señales que no se quieran medir en coincidencias. Este problema se resuelve agregando un *switch* para cada señal de entrada previo a su ingreso al módulo, y utilizando una compuerta OR entre el *switch* y la señal de entrada (fig. 3.8). De esta manera, si el *switch* se encuentra encendido, la salida de la compuerta OR será siempre un 1, y al realizar la operación AND sobre todas las señales, la apagada no afectará a la salida de ninguna manera.

3.4.4. Contador

El módulo contador se encarga de registrar la cantidad de pulsos entrantes de manera regular, y transmitir esa información al siguiente módulo. Dicho componente debe poder estar habilitado sólo durante el tiempo transcurrido durante la medida para saber cuando interrumpir la cuenta y enviar la información de la cantidad de

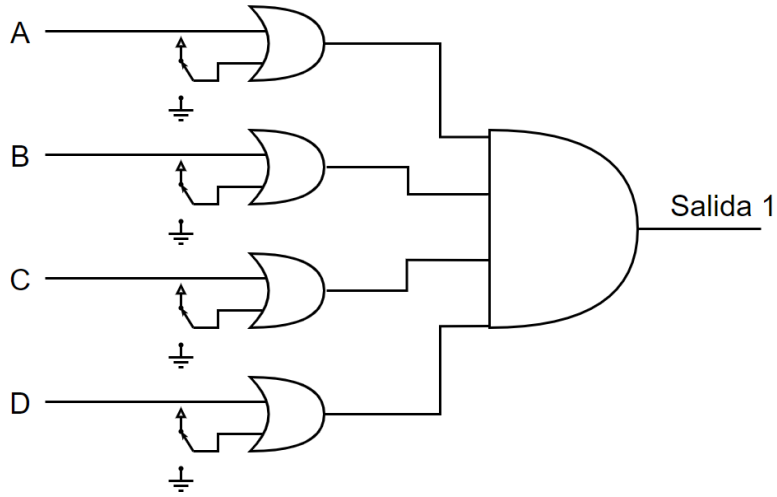


Figura 3.8: Circuito detector de señales en coincidencia para una salida (en particular la salida 1). Cada una de las cuatro señales de los canales de entrada A, B, C y D (ya conformadas), ingresan a una compuerta OR. A través de un conmutador, se puede fijar un '1' a la otra entrada de cada compuerta OR, lo que forzará un '1' lógico a la salida de la compuerta correspondiente, estando en esta configuración inactiva o 'apagada'. Las cuatro salidas de las compuertas OR son enviadas a una compuerta AND, que a su vez emitirá un pulso siempre que las cuatro entradas sean un '1'. Poniendo en '0' lógico por medio del conmutador en las OR que se seleccionen para habilitar, se permitirá que algunas de las cuatro compuertas envíen sus salidas 'copiando' los pulsos de entrada. Solo en el intervalo temporal en que las cuatro salidas de las OR presenten un '1' lógico la compuerta AND presentará un '1' a la salida.

pulsos contados, almacenar en una memoria la cantidad de cuentas registradas hasta el momento, y reiniciar la memoria luego de enviar la información. Para contar los pulsos entrantes, se implementaron variables tipo contador[1] asociadas a cada señal de entrada. De la misma manera, se implementó otra variable tipo contador asociada a la señal de reloj para conocer el tiempo transcurrido, de manera que al llegar a una cantidad de tiempo definida, la información de los pulsos contados se envíe al módulo siguiente y todas las variables tipo contador se reinicien.

3.5. Comunicación con la PC

La placa Spartan-3A XC3S700A cuenta con un puerto de comunicación serie, que permite establecer una comunicación entre la FPGA y la PC, utilizando el protocolo RS-232. Este tipo de comunicación crea bloques de información (usualmente de 1 byte) compuestos por (fig. 3.9):

- Los bits de datos, que son los portadores de la información que se busca transmitir; se puede configurar para que sean entre 5 y 9 bits.

[1]Se denomina variable tipo contador a una variable entera que aumenta en 1 al detectar un flanco de subida de una dada señal.

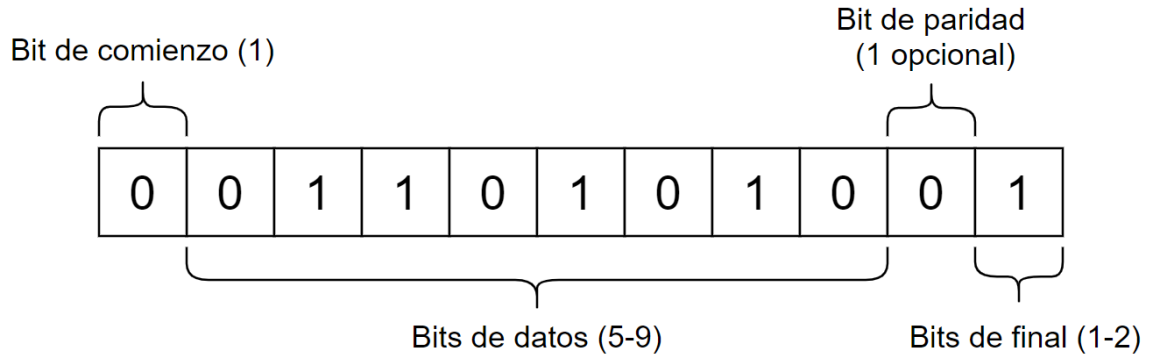


Figura 3.9: Esquema de los diferentes componentes de un bloque de información para comunicaciones, utilizando el protocolo RS-232.

- Los bits de sincronización, que se encargan de marcar el comienzo y el final del bloque de información, y están compuestos por un bit al comienzo, y 1 o 2 al final.
- El bit de paridad, una configuración opcional que permite encontrar errores en la transmisión.

Este protocolo es altamente configurable, permitiendo modificar todos los componentes anteriores y la tasa de *baud*, la cual determina la velocidad de transmisión de información. Esta última se mide en bits por segundo (bps), y puede tomar cualquier valor, siempre y cuando el emisor y el receptor estén configurados de la misma manera.

Para implementar este protocolo en FPGA, se agregó un módulo UART (*universal asynchronous receiver-transmitter*) que cumple la función de intermediario entre el contador de coincidencias y la PC, traduciendo entre la información que se quiere enviar/recibir y el protocolo RS-232 [44]. Este módulo fue configurado con 8 bits de datos, 2 bits de sincronización, ningún bit de paridad, y una tasa de *baud* de 115200 bps.

3.5.1. Interfaz Gráfica

Para facilitar la comunicación entre la FPGA y la PC, se desarrolló una interfaz gráfica en el software Processing 3 (fig. 3.10), que permite al usuario interactuar fácilmente con el contador de coincidencias. La interfaz contiene:

- Un arreglo bidimensional de botones tipo *switch* que permiten configurar los canales a medir en coincidencias. Cada fila del arreglo corresponde a una salida, y las columnas a los 4 canales de entrada, de manera que es posible configurar, para cada salida, que canales se quieren medir en coincidencia.

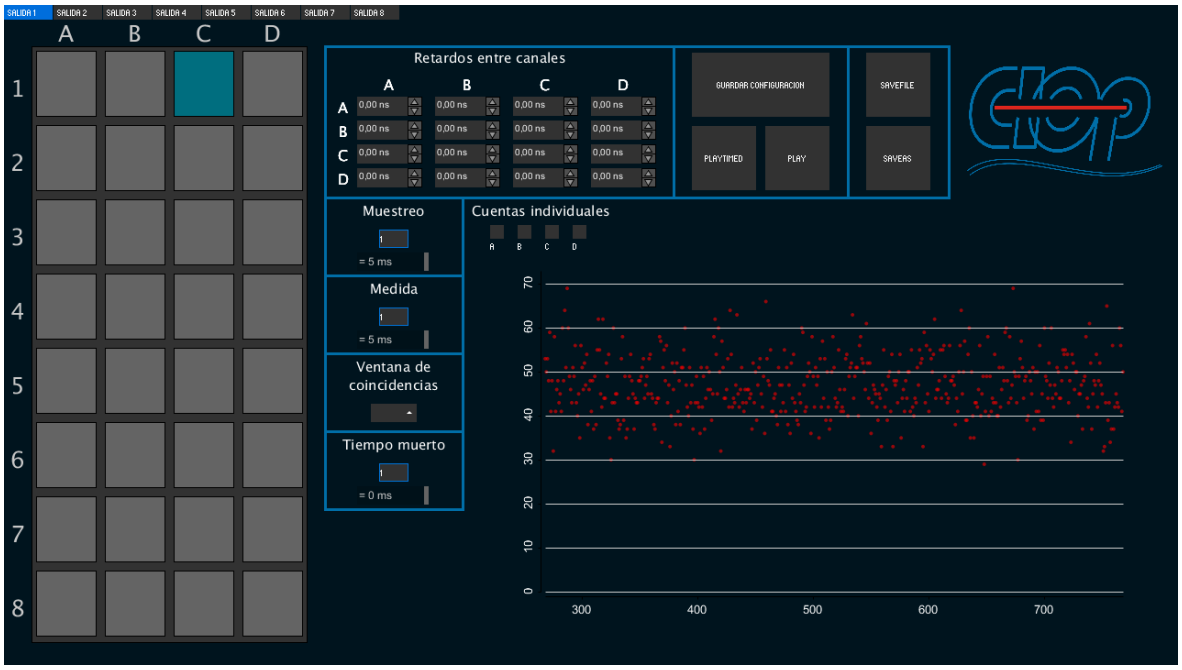


Figura 3.10: Interfaz gráfica para facilitar la comunicación entre la FPGA y la PC. El arreglo de botones de la izquierda permite configurar los canales a medir en coincidencias para cada salida. En la parte inferior derecha se encuentra el gráfico que muestra en tiempo real las cuentas registradas, y alrededor, las cajas de texto que permiten configurar los parámetros de la placa y de medida

- Un arreglo bidimensional cuadrado que tiene a los canales de entrada en sus lados, que permite configurar el tiempo que debe retardarse el canal de la fila respecto al canal de la columna. Está implementado de manera que no se puedan agregar retardos en la diagonal (puesto que cada canal no puede estar retardado consigo mismo), y está en plan de implementarse que la configuración sea automáticamente consistente, de manera que si ingreso un retardo entre los canales A y B, y un retardo entre los canales B y C, el retardo entre los canales A y C se determine automáticamente.

- Un menú desplegable que permite ingresar el tamaño de la ventana de coincidencia, eligiendo la cantidad de compuertas AND que tendrá la cadena de retardos, entre 0 y 15.

- Un gráfico que permite visualizar las cuentas registradas en tiempo real para la salida elegida en las pestañas de la parte superior.

- Cuatro botones que permiten, una vez elegida la salida, superponer al gráfico las cuentas de cada canal individual.

- Cajas de texto que permiten configurar el tiempo de muestreo y el tiempo de medida. El tiempo de muestreo indica cada cuanto tiempo se registran las cuentas, y es configurable en múltiplos enteros de 5 ms, mientras que el tiempo de medida

se refleja en la cantidad total de veces que se registran cuentas, y es configurable en múltiplos enteros del tiempo de muestreo configurado.

Para realizar una medida, una vez enviada la configuración hacia la FPGA, la interfaz cuenta con dos botones, uno de los cuales comienza la medida hasta que se frene, y el otro que mide por el tiempo establecido en el tiempo de medida. Finalmente, permite guardar estas medidas en un archivo de texto. El código de la interfaz se encuentra en [41], requiriendo petición de acceso.

Capítulo 4

Diseño del generador de números aleatorios

4.1. Generadores de números aleatorios

Un generador de números aleatorios (RNG por sus siglas en inglés) es un sistema en hardware y/o software capaz de generar números con la propiedad de que cada posible resultado sea tan probable como el resto, e independiente de factores externos.

La generación de números aleatorios cumple un rol fundamental en la vida cotidiana de mucha gente hoy en día, aún cuando a simple vista no lo parezca. Por ejemplo, la gran mayoría de los algoritmos de cifrado de datos (usuarios, contraseñas, tarjetas de crédito, etc) dependen de la generación de números aleatorios para crear claves seguras. Generar números aleatorios confiables y sin sesgo es de interés en un amplio espectro de aplicaciones, que incluyen criptografía, programación, y simulaciones numéricas, pero cumple un rol fundamental en información cuántica, y en particular en criptografía cuántica [15, 45–47].

4.1.1. Generador de números pseudoaleatorios

Cualquier generador implementado en software recibe el nombre de generador de números pseudoaleatorios (PRNG por sus siglas en inglés). Los PRNGs utilizan algoritmos matemáticos capaces de producir largas cadenas de datos que *parecen* aleatorias a primera vista, pero en realidad son completamente deterministas, periódicas, y determinadas únicamente por su estado inicial. Algunos de los algoritmos más conocidos para generar números pseudoaleatorios incluyen el generador lineal congruencial [48], Mersenne Twister [49] y el generador Blum Blum Shub (BBS) [50].

Para describir el funcionamiento general de un PRNG, tendremos un espacio finito S, donde el PRNG será una función $f : S \to S$. En general, S debe ser un conjunto grande, tal como los números enteros o reales, y f una función fácil de calcular, pero

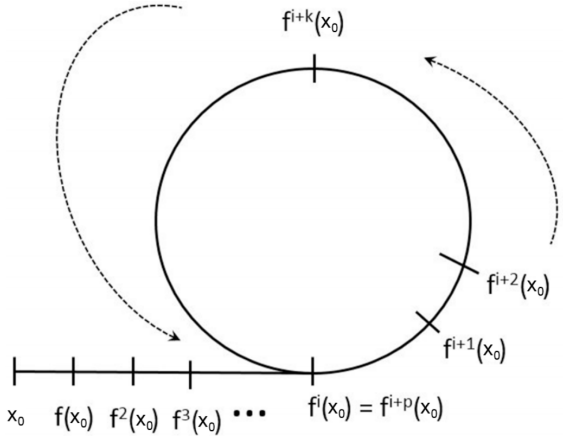


Figura 4.1: Estructura general de la secuencia generada por un PRNG dada una semilla x_0 y un período p.

cuya inversa no debe poder calcularse fácilmente. Se comienza con un dado valor x_0 llamado *semilla*, y se genera la secuencia

$$x_0,\ f(x_0),\ f(f(x_0)),\ f(f(f(x_0))),\ \ldots. \tag{4.1}$$

Definiendo $f^2(x_0) = f(f(x_0))$, $f^3(x_0) = f(f(f(x_0)))$, etc, teniendo en cuenta que S es finito, debe existir un mínimo entero i tal que $f^i(x_0) = f^j(x_0)$ para algún $i < j$. La cantidad $j - i$ recibe el nombre de período de f, y se denota por p. De esta manera, obtenemos una relación periódica en la secuencia tal que $f^q(x_0) = f^{q+tp}(x_0)$ para todo $q \geq i$ y todos los enteros t (fig. 4.1). Debido a que la generación se basa únicamente en métodos aritméticos, estos generadores se destacan por su rapidez, lo que los hace ideales para simulaciones de Monte-Carlo y programación en general.

La naturaleza determinista de los PRNGs, sin embargo, los vuelve una opción no viable para aplicaciones en criptografía. Esto es porque en principio, si un tercero conociese el estado inicial del generador, sería capaz de determinar los números generados en cualquier instante de tiempo. Este fue el caso del sistema operativo Android, donde debido a una mala inicialización de su PRNG en 2013, las aplicaciones que generaban claves mediante la Arquitectura Criptográfica de Java (JCA por sus siglas en inglés) no recibían números aleatorios lo suficientemente buenos. En particular, esto fue la causa de un robo de 5.700 dolares en Bitcoins, que equivalen aproximadamente a 1.425.000 dolares hoy en día [51, 52].

Si bien los números pseudoaleatorios no son realmente aleatorios, existen métodos de aumentar la *aleatoriedad* de un PRNG. Para lograrlo, pueden utilizarse fuentes de entropía externas como los parámetros del generador, como son el horario de la computadora, la posición del cursor, o la actividad del teclado [53, 54].

4.1.2. Generador de números verdaderamente aleatorios

Un generador de números verdaderamente aleatorios (TRNG por sus siglas en inglés), más correctamente llamado generador físico de números aleatorios, genera números aleatorios a partir de algún fenómeno físico. En vez de utilizar algoritmos deterministas, se realizan medidas de algún sistema físico prácticamente impredecible, y el resultado es luego convertido en bits aleatorios. Esta impredecibilidad puede ser causada por grados de libertad incontrolables (por ejemplo, ruido) o sistemas con comportamiento caótico (como la tirada de un dado o una moneda). Por otro lado, existen TRNGs basados en el comportamiento humano como fuente de aleatoriedad. El ejemplo más común de esto último es el generador presente en muchos sistemas operativos tipo Unix, denominado */dev/random*. Este archivo permite el acceso al ruido ambiental recogido de dispositivos y otras fuentes, como pueden ser la actividad del teclado, cursor, entre otros [54].

Una vez obtenidas las medidas sobre un sistema físico, debe realizarse un post-procesamiento en software para acondicionar los números y volverlos acordes a aplicaciones criptográficas. El post-procesado será una función matemática implementada en software que corrige las imperfecciones de la fuente de entropía, tales como sesgo o correlaciones. Un TRNG con una fuente de entropía débil debe tener un post-procesado fuerte para remover las imperfecciones, pero puede terminar escondiendo una falla, como vulnerabilidades. Para realizar un análisis de la entropía generada por un TRNG, es necesario acceder a la información cruda, en vez de a la post-procesada. Un usuario no tiene la capacidad de acceder a esta información, por lo que en general no hay manera para un usuario de detectar fallas o ataques a la fuente de entropía de un TRNG.

Una ventaja fundamental de un TRNG en contraste con un PRNG es la calidad de los números generados. El comportamiento del TRNG es prácticamente impredecible, debido a que muchos procesos físicos reales, si bien pueden ser clásicos y a fin de cuentas deterministas, son demasiado complejos como para que puedan predecirse satisfactoriamente [55]. Sin embargo, estos tipos de generadores suelen ser considerablemente más lentos que un PRNG, debido al tiempo existente entre la detección y el post-procesamiento de los datos.

4.1.3. Generador cuántico de números aleatorios

Un tipo particular de TRNG son los generadores cuánticos de números aleatorios (QRNG por sus siglas en inglés). Este tipo de generadores se basa en procesos cuánticos, que son fundamentalmente probabilísticos, para producir verdadera aleatoriedad. Una diferencia inmediata con un TRNG clásico, es la posibilidad de modelar el comportamiento del QRNG de manera exacta, debido a que el estado del QRNG se comporta según las leyes de la mecánica cuántica. Esto permite calcular analíticamente la pro-

babilidad asociada a cada posible resultado del QRNG en cada instante de tiempo. Sin embargo, por la naturaleza cuántica del sistema, a diferencia de un TRNG basado en un sistema clásico, aún si fuese posible conocer todos los parámetros iniciales del sistema y plantear el modelo que lo describe, sería imposible determinar el resultado con precisión.

Entre los fenómenos cuánticos utilizados para generar números aleatorios se encuentran: el decaimiento nuclear, fotones a través de un divisor de haz, y fluctuaciones en la energía de vacío, entre otros [56]. En la figura 4.2, se muestran cuatro tipos de QRNG basados en detección de fotones individuales. En el caso del divisor de haz, ingresan fotones individuales en el modo 1 de entrada, de modo que ingresa el estado

$$|1\rangle_1 |0\rangle_2 = \hat{a}_1^\dagger |0\rangle_1 |0\rangle_2 = \hat{a}_1^\dagger |0\rangle , \tag{4.2}$$

en la base de Fock. Por lo visto en la sección 2.2.1, el operador de creación en el modo 1 se puede escribir en función de los operadores de creación en los modos de salida 3 y 4, tal que

$$\hat{a}_1^\dagger = r\hat{a}_3^\dagger + t\hat{a}_4^\dagger, \tag{4.3}$$

es decir que, para un divisor de haz 50/50 ($r = t$), se obtiene el estado de salida

$$\hat{a}_1^\dagger |0\rangle = \frac{\hat{a}_3^\dagger + \hat{a}_4^\dagger}{\sqrt{2}} |0\rangle = \frac{|1\rangle_3 |0\rangle_4 + |0\rangle_3 |1\rangle_4}{\sqrt{2}} \tag{4.4}$$

por lo que resulta un 50 % de probabilidades para detectar un fotón en el detector del modo 3, y otro 50 % para detectarlo en el detector del modo 4, generando un bit aleatorio. Si se busca generar más de un bit de información, una posibilidad es utilizar los tiempos de arribo como variable aleatoria uniformemente distribuida, en vez del resultado de contar el número de fotones [57]. Suponiendo una fuente láser, la probabilidad de detectar n fotones en un intervalo de tiempo t viene dada por

$$P\{N(t) = n\} = \frac{\langle n \rangle^n}{n!} e^{-\langle n \rangle} \tag{4.5}$$

donde $N(t)$ es el número de fotones detectados en un tiempo t, $\langle n \rangle = \lambda t$ es el número medio de fotones detectados en tiempo t, y λ caracteriza la intensidad del láser. Comenzando desde $t = 0$, el tiempo de arribo del fotón i se representa como S_i. De esta manera, $\{S_n \leq t\}$ representa la detección del fotón n en un tiempo $t_p \leq t$. Dado que la cantidad de fotones detectados en el tiempo t será mayor a n, se pueden establecer dos eventos equivalentes

$$\{S_n \leq t\} \iff \{N(t) \geq n\}. \tag{4.6}$$

La anterior equivalencia permite escribir la función de distribución de S_n como

$$F_{S_n}(t) = P\{S_n \leq t\} = P\{N(t) \geq n\} = \sum_{i=n}^{\infty} \frac{\langle n \rangle^i}{i!} e^{-\langle n \rangle}. \tag{4.7}$$

La densidad de probabilidad puede obtenerse derivando la expresión anterior [58] para obtener

$$f_{S_n}(t) = \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!}, \tag{4.8}$$

que no es una distribución uniforme, por lo que no es buena idea obtener la información directamente de S_n. Como alternativa, se propone expresar S_n como un múltiplo entero de un período T que representa el tiempo en el que se detecta 1 fotón ($N(T) = 1$), sumado a una fracción del período t_φ, de manera que

$$S_n = q_n \times T + t_\varphi. \tag{4.9}$$

Así, dado un T fijo, t_φ se encuentra en el intervalo $[0, T)$, por lo que cada detección de un fotón ocurre solo una vez en ese intervalo. Con esto puede encontrarse la distribución de probabilidad de t_φ dado que $N(T) = 1$ como

$$F_{t_\varphi}(t) = P\{t_\varphi \leq t | N(T) = 1\} = \frac{P\{t_\varphi \leq t, N(T) = 1\}}{P\{N(T) = 1\}} =$$

$$= \frac{P\{N(t) = 1, N(T) - N(t) = 0\}}{P\{N(T) = 1\}} = \frac{P\{N(t) = 1\} P\{N(T) - N(t) = 0\}}{P\{N(T) = 1\}} =$$

$$= \frac{\lambda t e^{-\lambda t} e^{-\lambda(T-t)}}{\lambda T e^{-\lambda T}} = \frac{t}{T}, \tag{4.10}$$

y puede calcularse la distribución de probabilidad como

$$f_{t_\varphi}(t) = \frac{1}{T} \tag{4.11}$$

que es uniforme en $[0, T)$. Si se quieren obtener números aleatorios entre 0 y N_0, se divide el período T en N_0 intervalos iguales, y se extrae el número aleatorio según el intervalo en el que fue detectado el fotón.

La velocidad de generación de un QRNG depende ampliamente del proceso utilizado como fuente de entropía. Por ejemplo, uno implementado en laboratorio como los mencionados anteriormente tienen su velocidad de generación limitada por la eficiencia de los detectores y la intensidad del láser utilizado, pero también existen los QRNG comerciales que generan números aleatorios con velocidades en el orden de los

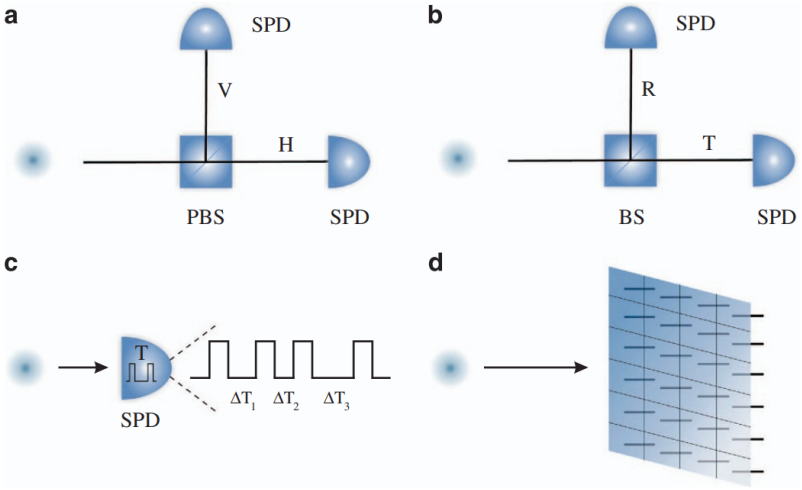


Figura 4.2: Diferentes tipos de QRNG utilizando fotones. (a) Un bit aleatorio se genera determinando, a través de la medida de un detector, el camino tomado por un fotón descrito por el estado $\frac{|H\rangle+|V\rangle}{\sqrt{2}}$, que ingresa a un PBS. (b) Se genera un bit aleatorio determinando el camino tomado por un fotón que ingresa a un divisor de haz simétrico. (c) Se generan bits aleatorios midiendo el intervalo de tiempo Δt entre dos eventos detectados. (d) Se generan bits aleatorios según la posición espacial en la que se detecta al fotón utilizando un arreglo de SPDs. Imagen extraída de [61].

Gbits/s[1]. En un caso real de QRNG, los efectos cuánticos se mezclan con ruidos clásicos, que pueden ser removidos modelando el proceso cuántico en cuestión, como se hizo anteriormente para dos ejemplos particulares [59].

En la actualidad, es posible adquirir QRNGs comerciales fabricados por empresas como QRANGE, ID Quantique, y Quintessence Labs, que garantizan portabilidad y altas velocidades de generación, con precios que rondan entre los 1000 y los 5000 euros. Sin embargo, si el tamaño no es de importancia, es posible realizar implementaciones simples en un laboratorio, que resultan mucho más económicas que su contraparte comercial. En particular, en 2014 se propuso un QRNG utilizando la cámara de un teléfono celular [60].

Finalmente, existen maneras de obtener números generados con un QRNG desde una PC. El primero es un servicio provisto por ANU QRNG [62], que utiliza fluctuaciones de vacío para generar números en tiempo real [63, 64]. El servicio posee ciertas características que ayudan a garantizar la seguridad de los números generados:

- Cada vez que se accede al sitio, los números generados serán nuevos y únicos.
- Si dos o más usuarios acceden al sitio al mismo tiempo, cada uno recibirá distintos números.
- El sitio está encriptado y autentificado para agregar una capa extra de seguridad.

[1]Por ejemplo, la velocidad de generación necesaria para lograr asegurar un teléfono celular es de 1 kbit/s

- El sitio no guarda los números generados en una memoria.

Este generador se puede agregar mediante librerías/paquetes a los lenguajes de programación más utilizados, que incluyen Java, C, C++, Python, Matlab, etc. Además, es posible generar números aleatorios utilizando las computadoras cuánticas provistas por IBMQ. La mayor desventaja de este último es la velocidad de generación. Al utilizar un servicio público con orden de prioridad, dependiendo de la computadora utilizada, los tiempos de espera pueden llegar a durar horas. Un algoritmo para generar números aleatorios con las computadoras de IBMQ se encuentra en [65]. En ambos casos, si bien se garantiza seguridad, el transporte de información se realiza por canales clásicos, por lo que resultan vulnerables a espionaje y sabotaje.

4.1.4. Diferencias

Cada uno de estos tipos de RNGs tiene sus pros y contras, resumidas en la tabla 4.1. La principal diferencia se encuentra en la fuente de aleatoriedad de cada uno. Un PRNG genera números mediante un algoritmo matemático conocido al crear el PRNG, y teniendo en cuenta que la secuencia generada será periódica, la magnitud de dicho período da noción de la aleatoriedad del generador. En contraste, un TRNG utiliza como fuente de entropía fenómenos físicos macroscópicos con condiciones metaestables o caos, tales como la tirada de una moneda, ruido eléctrico o térmico, o *jitters*. En la mayoría de los casos, el proceso utilizado es imposible de monitorear, debido a que plantear el modelo del sistema es en extremo difícil. Esto último deja de ser un problema para un QRNG, cuya aleatoriedad proviene de las medidas realizadas en sistemas que obedezcan las leyes de la mecánica cuántica. A diferencia de un TRNG clásico, la fuente de entropía de un QRNG puede ser descrita por un modelo fundamental, cuyas propiedades y comportamiento son conocidos, y puede probarse la seguridad realizando cálculos analíticos previos.

Otra diferencia importante es la velocidad de generación. En el caso del PRNG, si bien la calidad de los números generados es baja, la velocidad de generación es muy alta, teniendo en cuenta que los algoritmos de generación suelen ser simples de calcular en una computadora. En el caso del TRNG clásico, la velocidad suele ser considerablemente más lenta, debido al tiempo entre la medida del sistema y el post-procesamiento de los datos. Finalmente, las medidas a un sistema cuántico suelen ser lo suficientemente aleatorias como para no necesitar un post-procesamiento, lo que los hace más rápidos que un TRNG clásico, sin embargo la velocidad de generación es altamente dependiente del proceso utilizado como fuente de entropía.

Finalmente, estos generadores se diferencian en la seguridad de la secuencia generada. En el caso de un PRNG, basta con conocer el algoritmo utilizado y el valor inicial para poder determinar la secuencia en su totalidad, volviéndolo la opción más insegura

	PRNG	TRNG	QRNG
Fuente de aleatoriedad	algoritmos matemáticos	caos o ruido clásico	fenómenos cuánticos
Tipo de secuencia generada	determinista	aleatoria o incomprobable	aleatoria
Velocidad de generación	alta	baja	variable
Complejidad de implementación	baja	media	alta
Modelado matemático	simple	extremadamente complejo	simple
Detección de ataques	imposible	imposible o limitado	posible
Seguridad	nula	incomprobable	seguros comprobables

Tabla 4.1: Comparación de las principales características de los distintos tipos de generadores de números aleatorios.

del conjunto. En el caso del TRNG, el hecho de no poder monitorear el proceso encargado de la generación tiene como consecuencia que es imposible probar la seguridad del generador. En particular, un atacante podría modificar la fuente de entropía sin ser detectado, agregando una vulnerabilidad fundamental a la seguridad del sistema. En un QRNG, todas estas preocupaciones desaparecen, puesto que incluso si un atacante modificase la fuente de entropía, bastaría con realizar un modelo matemático de la fuente, y revisar que los números generados sean consistentes con lo calculado analíticamente.

4.2. Medidas de aleatoriedad

La posibilidad de generar números realmente aleatorios trae aparejada la necesidad de evaluar la calidad de esos generadores. Esa evaluación puede efectuarse cuantificando la *aleatoriedad* de una cadena de números. Si bien este concepto en principio suena simple, resulta ser en extremo complejo.

Supongamos, por ejemplo, que se realizan n tiradas de una moneda, y supongamos ahora que el resultado de todas las tiradas fue 'cara'. Uno podría pensar, y con buen argumento, que la moneda posee un sesgo, teniendo en cuenta que la probabilidad de obtener ese resultado es ínfima para n grande (2^{-n}). Sin embargo, lo mismo se puede decir para cualquier otra secuencias de n resultados de 'cara' y 'seca'. Ahora surge la pregunta, ¿cómo es posible determinar si una secuencia es realmente aleatoria? La respuesta corta es que no es posible, sin embargo existen ciertas cantidades y pruebas estandarizadas que permiten estimar la aleatoriedad de una cadena de números. Estos se encargan de buscar la presencia o ausencia de 'patrones', que en caso de ser

detectados, dan indicios de un bajo nivel de aleatoriedad.

4.2.1. Distribución k

Si nos limitamos solo a PRNGs, que ya es sabido no son realmente aleatorios, podemos definir una cantidad que da noción de que tan no-aleatorio es un PRNG. Esta cantidad se conoce como distribución k. Dada una secuencia pseudoaleatoria x_i de enteros de w bits con período p, definimos la función $trunc_v(x)$, que permite obtener los primeros v bits de x. De esta manera, dados p de los k vectores de v bits

$$(trunc_v(x_i), trunc_v(x_{i+1}), ..., trunc_v(x_{i+k-1})) \ (0 \leq i \leq p), \tag{4.12}$$

se dice que la secuencia es k-distribuida con precisión de v-bits si todas las 2^{kv} posibles combinaciones de bits ocurren la misma cantidad de veces en un período, exceptuando por las combinaciones que contienen solo ceros. Cuanto más bajo el valor de k, menos aleatoria será la secuencia. Es importante notar que a priori, esta cantidad mide aleatoriedad sobre *secuencias* y no *generadores*. Para decir que un generador está k-distribuido, todas las secuencias generadas por él deben ser k-distribuidas. Esta medida de aleatoriedad solo funciona con PRNGs, debido a que es necesario conocer el período del generador, inexistente para TRNGs y QRNGs.

4.2.2. Entropía

La entropía de una variable aleatoria X, donde la probabilidad de obtener cada resultado posible x_i es $P(X = x_i) = p_i$ se define como

$$H(X) = -\sum_i p_i \log(p_i), \tag{4.13}$$

donde $H(X)$ recibe el nombre de entropía de Shannon. El logaritmo puede realizarse en cualquier base, pero el más utilizado es base 2, que permite obtener la entropía medida en bits. Otra medida de entropía es por ejemplo la entropía mínima H_∞, que se calcula como el logaritmo de la probabilidad máxima de la distribución, tal que

$$H_\infty = \min_{x_i} \left(-\log(P(X = x_i))\right) = -\log\left(\max_{x_i} P(X = x_i)\right). \tag{4.14}$$

Esta última es de particular interés en seguridad, ya que da noción de la existencia de número aleatorio más probable en la distribución.

Para poder realizar estos cálculos de entropía para un RNG, es necesario conocer la distribución de probabilidad del generador en cuestión. En el caso de un PRNG, para obtener las probabilidades de cada resultado, es necesario generar suficientes números para completar un período del generador, y luego evaluar la frecuencia con la que

aparece cada x_i. El problema de esto es que en general el período de un PRNG es demasiado grande como para poder realizar este proceso en un tiempo accesible. En el caso de un TRNG, la ausencia de un período y la imposibilidad o alta dificultad de modelar matemáticamente la generación, hace imposible también el calculo exacto de la entropía del generador. En ambos de estos casos, sin embargo, puede realizarse una estimación de la entropía, aproximando la probabilidad con la frecuencia de cada resultado x_i en una cadena de longitud n, tal que

$$\hat{p}_i = \frac{n_i}{n}, \tag{4.15}$$

donde n_i es la cantidad de apariciones de x_i, que me permite estimar la entropía como

$$H(X) = -\sum_i \hat{p}_i \log(\hat{p}_i). \tag{4.16}$$

Esta estimación no garantiza aleatoriedad en el generador, si no que es una medida sobre la *secuencia* generada. Esto no es un problema en el caso de un QRNG, donde es posible realizar un modelado matemático y calcular la distribución de probabilidad de cada resultado sin la necesidad de generar ninguna secuencia, lo que permite obtener la entropía del generador fácilmente.

4.2.3. Pruebas de aleatoriedad

Las pruebas estadísticas son uno de los métodos mas comúnmente utilizados para estimar la aleatoriedad de un generador. Estas son diseñadas para probar una *hipótesis nula* ($H0$). En este caso, la hipótesis nula será que 'la secuencia a probar es aleatoria'. Estas pruebas consideran una variable aleatoria cuya función de distribución es conocida. Con la distribución decidida, se calcula un número real entre 0 y 1, llamado *p-value*. Para calcular el *p-value*, primero se plantea una distribución de probabilidad esperada suponiendo $H0$ verdadera. El calculo del *p-value* será, dada una observación O, la probabilidad de obtener dicha observación dado $H0$ verdadera, es decir

$$p = P(O|H0). \tag{4.17}$$

El resultado del $p - value$ puede interpretarse de distintas maneras según el tipo de prueba realizada. En un primer paso, se define un valor critico $\alpha \in [0, 1]$ llamado nivel de significancia. Las pruebas se clasifican, según el criterio de rechazo de H0, en pruebas de una y dos colas. En una prueba de una cola a izquierda, $H0$ se rechaza si $p < \alpha$ (si la prueba se realiza a derecha, $H0$ se rechaza si $p > 1 - \alpha$), mientras que para una prueba de dos colas, $H0$ se rechaza si $p < \alpha$ o $p > 1 - \alpha$ (fig. 4.3)

Supongamos nuevamente el caso de n tiradas de moneda. Realizamos una obser-

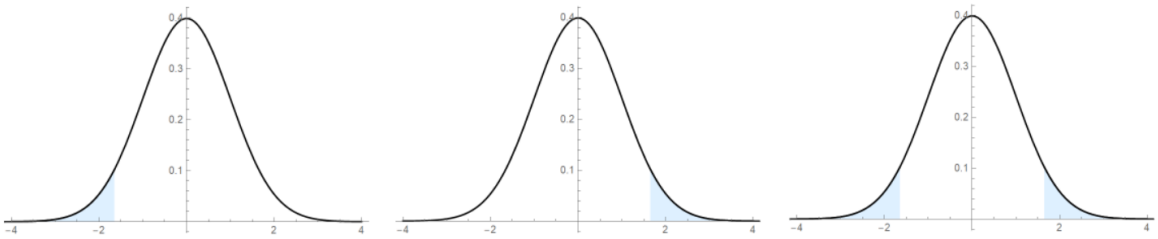


Figura 4.3: Representación gráfica de los valores críticos de $p - value$ para, de izquierda a derecha, una prueba de una cola a izquierda, una prueba de una cola a derecha, y una prueba de dos colas. La zona sombreada representa un nivel de significancia $\alpha = 0{,}05$, tal que cualquier resultado dentro de dichas zonas, rechaza $H0$.

vación O con $n = 50$, obteniendo que 10 de los resultados fueron 'cara'. En el caso de este ejemplo, $H0$ será que la moneda está balanceada con un nivel de significancia $\alpha = 0{,}05$, por lo que la distribución de probabilidad será una distribución binomial, y realizando el calculo:

$$p = P(O|H0) = \sum_{i=0}^{10} \binom{50}{i} 0{,}5^i 0{,}5^{50-i} = \sum_{i=10}^{50} \binom{50}{i} 0{,}5^{50} = 1,19 \times 10^{-5}, \tag{4.18}$$

que es mucho menor al nivel de significancia propuesto, por lo que podemos rechazar $H0$, y decir que la moneda no está balanceada.

El resultado de una prueba de aleatoriedad individual no suele ser suficiente para determinar si una secuencia es o no lo suficientemente aleatoria. Por este motivo, se usan conjuntos de pruebas, llamadas *baterías*, para observar diferentes comportamientos de la secuencia a analizar. Entre las baterías de pruebas más utilizadas actualmente se encuentran la TestU01 [66], PractRand [67], Dieharder [68], y, aunque actualmente desactualizadas, las provistas por NIST (*NIST Statistical Test Suite*) [69], y Diehard [70].

De todas maneras, aún si el conjunto de secuencias generadas por un dado RNG pasan un grupo de pruebas de aleatoriedad, no se garantiza que el RNG utilizado sea realmente aleatorio. Un caso destacable es el generador MaD0, que pasa las baterías de pruebas TestU01, Diehard, y *NIST Statistical Test Suite*, aún siendo un PRNG, por lo que no es realmente aleatorio.

4.3. Implementación en FPGA

Se buscó implementar un RNG en la placa de desarrollo FPGA. A la hora de decidir el tipo de RNG a implementar, se decidió que las ventajas cualitativas brindadas por un TRNG o QRNG no eran lo suficientemente impactantes para una aplicación destinada al testeo del procesamiento de señales, más que a determinar la calidad de los

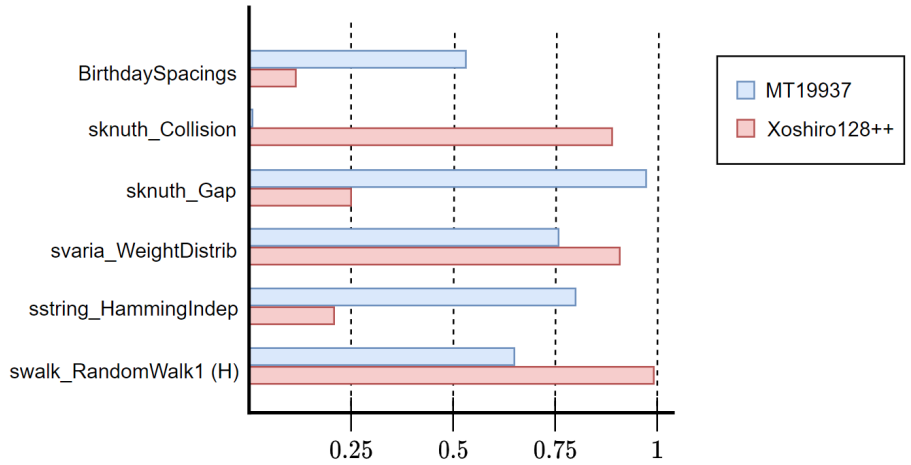


Figura 4.4: Fragmento de los resultados obtenidos para los generadores de números aleatorios MT19937 (en rojo) y Xoshiro128++ (en azul) utilizando la batería de pruebas TestU01.

números aleatorios. Por otra parte, esta elección hace posible controlar la estadística de los números generados. Por esto, se decidió implementar un PRNG con un período lo suficientemente largo como para que las diferencias cuantitativas no sean apreciables. De esta manera, en primera instancia se implementó en FPGA un PRNG basado en el generador Mersenne Twister (en particular la variante MT19937), uno de los más utilizados para propósitos generales en la actualidad [71]. La implementación de este generador requiere una asignación de memoria en bloques RAM, la cual se realiza automáticamente por el entorno de desarrollo. Sin embargo, al implementar múltiples instancias del mismo generador con distinta semilla, la asignación automática de memoria no fue tan precisa, y se perdía información. Teniendo en cuenta esto último, se decidió implementar otro PRNG, esta vez basado en el generador Xoshiro128++, un PRNG de pocos recursos que no requiere asignar memoria, pero con alta velocidad y calidad [72]. A modo de comparación y para dar garantía de su calidad, se utilizó una de las baterías de pruebas mencionadas anteriormente (TestU01) sobre ambos generadores, obteniendo buenos resultados para ambos (fig. 4.4). Los resultados completos se encuentran en el siguiente link [41].

El generador Xoshiro128++ genera números en el rango $[0, 2^{32} - 1]$, y posee un período de $2^{128} - 1$. La implementación de ambos generadores en VHDL fue realizada por J. van Rantwijk, y puede encontrarse en [73]. Esta implementación permite generar un número de 32 bits en cada ciclo de reloj (20 ns), y admite la opción de modificar la semilla durante la generación. Los códigos implementados pueden encontrarse en [41], requiriendo petición de acceso.

Si bien el circuito implementado en FPGA se desarrolla en una PC mediante un código escrito en VHDL, la implementación a fin de cuentas se realiza en hardware, dando lugar a la posibilidad de implementar un TRNG en la FPGA. Dichas implementaciones utilizan anillos osciladores, que consisten en un número impar de compuertas

NOT realimentadas, y se han logrado frecuencias de generación de hasta 300 millones de bits por segundo [74, 75]. Sin embargo, por las razones mencionadas anteriormente, no se implementó un TRNG en FPGA.

Capítulo 5

Estudio de fuentes

5.1. Estadística de fuentes de fotones

En el capítulo 2 se hizo énfasis en la importancia de las fuentes de fotones individuales para aplicaciones de criptografía cuántica. En la práctica, la generación de fotones individuales es una parte crucial de cualquier experimento, por lo que es necesario poder diferenciar este tipo de fuentes del resto. Para hacerlo, se compara el comportamiento de la estadística de las distintas fuentes y se separan, según la relación entre el valor medio del número de fotones y su varianza, en: poissoniana, super-poissoniana, y sub-poissoniana. Los tipos de fuente más representativos de cada estadística son respectivamente las fuentes láser, térmica, y de estados de Fock.

5.1.1. Láser

El término láser proviene del acrónimo 'amplificación de luz por emisión estimulada por radiación' (*Light Amplification by Stimulated Emission of Radiation* en inglés). Es un dispositivo que utiliza la emisión estimulada de radiación en un medio apropiado, para generar un haz de luz cuyas características especiales de monocromaticidad, coherencia y direccionalidad, en el caso de un láser de onda continua, se encuentran perfectamente controladas. Sin embargo, para el caso de un láser pulsado en el orden de los nanosegundos o menores, la noción de monocromaticidad se pierde.

El estado cuántico de la luz emitida por un láser monocromático puede aproximarse como un estado coherente $|\alpha\rangle$ [76], tal que

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{n!} |n\rangle \,, \tag{5.1}$$

donde el coeficiente α determina el valor medio del número de fotones, $\langle \hat{n} \rangle = |\alpha|^2$, y la fase del estado coherente. Entre las propiedades de interés de estos estados se encuentra, en primera instancia, que la probabilidad de 'encontrar' n fotones en el estado $|\alpha\rangle$ viene

dada por

$$P_{láser}(n) = |\langle n|\alpha\rangle|^2 = \frac{e^{-|\alpha|^2}|\alpha|^{2n}}{n!}, \tag{5.2}$$

es decir, una distribución de Poisson. Además, como es de esperarse de una distribución de Poisson, se encuentra que la incerteza en el número de fotones está relacionada con el valor medio de los mismos como

$$(\Delta\hat{n})^2_{láser} = \langle\hat{n}^2\rangle - \langle\hat{n}\rangle^2 = \langle\hat{n}\rangle = |\alpha|^2. \tag{5.3}$$

5.1.2. Radiación Térmica

La radiación térmica es emitida por todo cuerpo con temperatura $T > 0$, mediante la conversión de energía cinética interna a energía electromagnética. Las características de la radiación térmica dependen de las propiedades del emisor, como son su temperatura, geometría, etc [77]. Un estado térmico se describe utilizando radiación de cuerpo negro [78, 79], y es una mezcla estadística incoherente de estados de Fock $|n\rangle$. Para describir estos estados mediante la mecánica cuántica, es necesario utilizar el formalismo de la matriz densidad [5]. Sin embargo, para el análisis que se hará en este trabajo, será suficiente con saber, de manera análoga a lo determinado para la fuente láser, la probabilidad de encontrar n fotones en el estado, que viene dada por la distribución de Bose-Einstein, tal que

$$P_{térmica}(n) = \frac{\langle\hat{n}\rangle^n}{(1+\langle\hat{n}\rangle)^{n+1}}. \tag{5.4}$$

Estos estados tienen la particularidad de que el estado más probable es siempre el vacío, es decir, el estado con $n = 0$. Teniendo en cuenta esta distribución, para luz térmica la incerteza en el número de fotones se relaciona con el valor medio como

$$(\Delta\hat{n})^2_{térmica} = \langle\hat{n}\rangle^2 + \langle\hat{n}\rangle. \tag{5.5}$$

Como $(\Delta\hat{n})^2 > \langle\hat{n}\rangle$, se dice que la luz térmica posee una estadística super-poissoniana.

5.1.3. Estados de Fock

El comportamiento de la luz proveniente de las fuentes mencionadas anteriormente puede describirse mediante una densidad de probabilidad clásica, de modo que la probabilidad de encontrar n fotones está descrita por una función densidad de probabilidad (Poisson para el láser y Bose-Einstein para radiación térmica). Este no es el caso para las fuentes de estados de Fock, es decir, que emiten exactamente un estado

del tipo $|n_0\rangle$. Para estos estados, la probabilidad de encontrar n fotones está descrita por

$$P_{Fock}(n) = \begin{cases} 1, & n = n_0 \\ 0, & n \neq n_0 \end{cases} . \tag{5.6}$$

De la misma manera, para una fuente de n_0 fotones, como el número de fotones está perfectamente definido, se encuentra que

$$(\Delta\hat{n})^2_{Fock} = 0. \tag{5.7}$$

Como $(\Delta\hat{n})^2 < \langle\hat{n}\rangle$, se dice que los estados de Fock poseen una estadística sub-poissoniana.

5.1.4. Estudio experimental de estadística de fuentes

En un trabajo previo, realizado en el Centro de Investigaciones Ópticas para la materia Experimentos Cuánticos II 2019, se caracterizó un tubo fotomultiplicador (PMT) HAMAMATSU HC-135(02) para su utilización en experimentos de óptica cuántica [80]. En dicho trabajo, se utilizó el PMT para realizar un estudio de la estadística de fotodetección para fuentes tanto láser como térmica. Aquí se presentan los resultados experimentales obtenidos que posteriormente sirvieron de modelo de control para las simulaciones realizadas en este Trabajo de Diploma.

Materiales y métodos

Para analizar experimentalmente las características de la fuente láser y su estadística de fotodetección se dispuso de un arreglo como el mostrado en la figura 5.1a. Se utilizó un láser de He-Ne de ThorLabs como fuente de luz, espejos regulables para facilitar la alineación del láser, placas polarizadoras y atenuadores para poder regular la potencia de la luz, y un radiómetro NewPort 880 *Universal Shutter System* para medir dicha potencia. Las medidas se realizaron por un tiempo de 1 minuto. Se realizó una serie de medidas manteniendo una ventana de medición de $T = 10$ ms fija y variando la potencia entre $P_{min} = 5$ pW y $P_{max} = 346$ pW, y otra manteniendo la intensidad fija en $P = 5$ pW y variando la ventana entre $T = 10$ ms y $T = 500$ ms. El estudio de la estadística de fotodetección de una fuente térmica requirió de una modificación a este experimento inicial y consideraciones adicionales.

Una característica de las fuentes térmicas es que el tiempo de coherencia τ_c de las mismas es extremadamente corto (usualmente menor a 1 ps). Para un tiempo de medida $T \gg \tau_c$, la estadística de Bose-Einstein se pierde, y el resultado es idéntico a la de un láser. Para observar la estadística de Bose-Einstein, es necesario un tiempo de

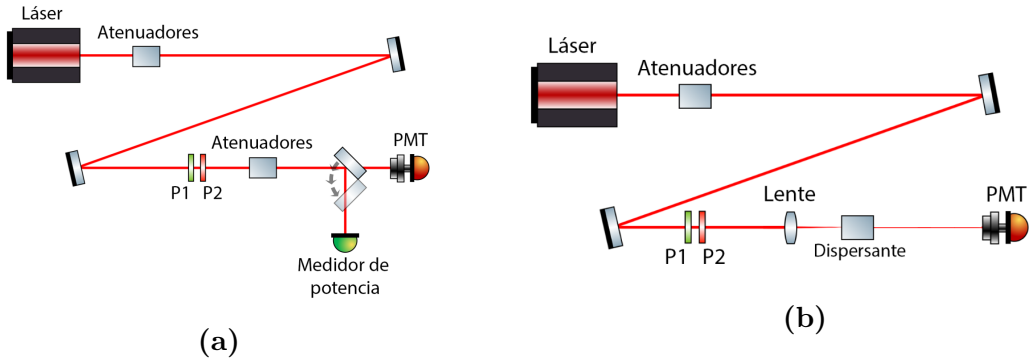


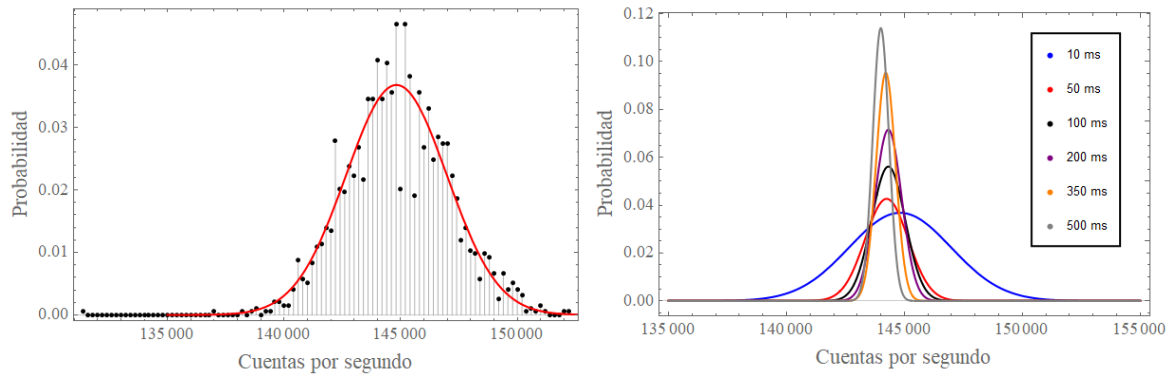
Figura 5.1: Arreglos experimentales utilizados para el análisis de la estadística de una fuente: (a) láser (b) pseudo-térmica. Como dispersante, se utilizaron tanto partículas de látex suspendidas en agua, como un vidrio esmerilado móvil.

medida $T \ll \tau_c$, que por limitaciones del sistema de detección no puede alcanzarse. Sin embargo, es posible crear una fuente pseudo-térmica a partir de un patrón de *speckle* dinámico. Un patrón de *speckle* se logra dispersando luz láser a través de un gran número de centros de dispersión. Al mover estos centros de dispersión, el patrón de *speckle* evoluciona temporalmente, creando la fuente pseudo-térmica. Existen distintos métodos para crear el patrón de *speckle* necesario. Uno de los aquí utilizados consiste en dispersar la luz a través de un recipiente transparente a la longitud de onda del láser, el cual se llena con bolitas de plástico de temaños menores a 1 μm suspendidas en agua [81]. El otro consiste en dispersar la luz a través de un vidrio esmerilado móvil[82]. El primer método es más económico y accesible, mientras que con el segundo método, es posible controlar el tiempo de coherencia regulando la velocidad del vidrio.

Se utilizaron los métodos mencionados anteriormente para generar la fuente 'pseudo-térmica'. Para ambos, se modificó el arreglo de la figura 5.1a agregando una lente para enfocar el láser (fig. 5.1b). Para el primer método, se utilizaron partículas de látex suspendidas en agua dentro de un recipiente de plástico transparente. Las medidas debieron realizarse inmediatamente después de agregar las partículas de látex, debido a que las mismas se disipan rápidamente y se pierde la capacidad de generar un patrón de *speckle*. Para el segundo método se colocó un vidrio esmerilado en el camino del haz, y se utilizó un motor ThorLabs MTS50-Z8 para moverlo en el plano perpendicular al camino del láser a una velocidad de $v = 2,4$ mm/s.

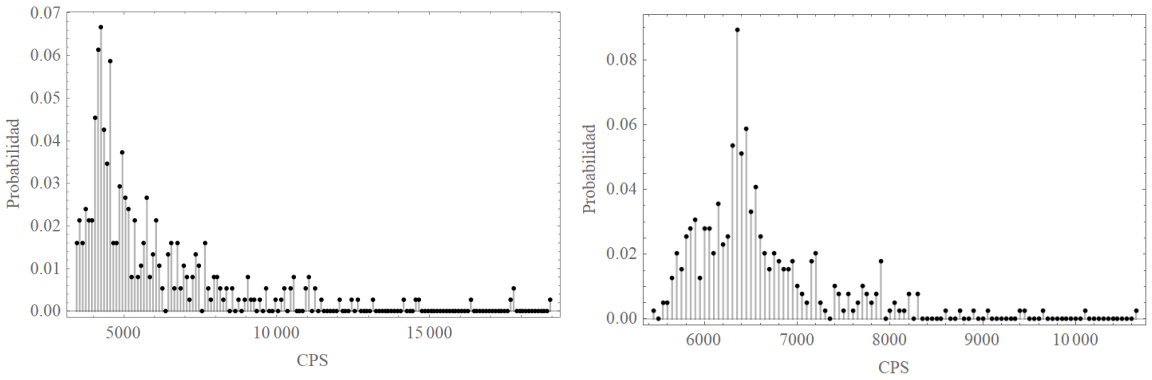
Resultados

Se obtuvieron las distribuciones para conteo de fotones en las distintas fuentes y se las comparó con las esperadas. Las medidas de $P(n)$ para diferentes potencias incidentes se encuentran en el apéndice A. Para las medidas del láser de He-Ne, se obtuvo que la probabilidad de ocurrencia de las cuentas registradas sigue una distribución normal (fig. 5.2a). Esto no contradice lo esperado, puesto que cuando $\langle n \rangle \to \infty$, la distribución



(a) **(b)**

Figura 5.2: (a) $P(n)$ del láser de He-Ne para $P = 5$ pW y $T = 10$ ms. En rojo, el ajuste correspondiente. (b) Ajustes para $P(n)$ para una potencia $P = 5$ pW y $T = 10, 50, 100, 200, 350, 500$ ms.



(a) **(b)**

Figura 5.3: $P(n)$ para una fuente pseudo-térmica generada: (a) suspendiendo partículas de látex en agua. (b) con un vidrio esmerilado moviéndose a $v = 2,4$ mm/s.

Poissoniana tiende a la distribución normal [83]. Además, se observó que a medida que se aumenta T, el ancho de la distribución se reduce (fig. 5.2b).

Para el primer método de creación de fuente pseudo-térmica, se realizaron una serie de medidas modificando T (fig. 5.3a). Se observó que a medida que T aumenta, $P(n)$ deja de comportarse como una distribución de Bose-Einstein y tiende a comportarse como una distribución normal. Los gráficos de $P(n)$ para distintos T se encuentran en el apéndice A. Además, utilizando este método, no hay manera de modificar ni predecir τ_c. Para el segundo método, se realizó una única serie de medidas a potencia fija (fig. 5.3b). Se observó que ninguna de las distribuciones de probabilidad se comportaba como Bose-Einstein. Se cree que esto se debe a que la velocidad del motor utilizado no es la correcta para generar luz con tiempos de coherencia mucho mayores a la ventana de medición. Para obtener la distribución esperada, se debería, o bien disminuir la velocidad del motor, o bien disminuir la ventana de medida, lo cual e ra una limitación al momento en que se realizó el experimento.

5.2. Calidad de una fuente de fotones

Siendo que existen distintos tipos de fuentes de fotones, es necesario definir una cantidad que permita determinar qué tanto se parece la fuente creada a una fuente de fotones individuales ideal. Para esto, se utiliza la función de correlación de segundo orden, $g^{(2)}(0)$.

La función de correlación de segundo orden, definida por Glauber en 1963 [84], permite distinguir entre distintos estados de luz. En particular, permite distinguir si una fuente de fotones posee estadística poissoniana, sub-poissoniana o super-poissoniana. En el caso de estadísticas poissoniana y super-poissoniana, donde la luz puede describirse con un campo electromagnético clásico, la función de correlación de segundo orden $g_C^{(2)}(\tau)$ describe correlaciones entre dos pulsos separados temporalmente un dado τ, y se define como

$$g_C^{(2)}(\tau) = \frac{\langle E^*(t)E^*(t+\tau)E(t+\tau)E(t)\rangle}{\langle E^*(t)E(t)\rangle^2} = \frac{\langle I(t)I(t+\tau)\rangle}{\langle I(t)\rangle^2}, \tag{5.8}$$

donde los promedios se realizan sobre un modo en el tiempo. En el caso particular de separación nula entre los pulsos ($\tau = 0$), la expresión anterior se reduce a

$$g_C^{(2)}(0) = \frac{\langle I(t)^2\rangle}{\langle I(t)\rangle^2}. \tag{5.9}$$

Es posible encontrar una cota a $g^{(2)}(0)$, utilizando la desigualdad

$$\langle I^2(t)\rangle - \langle I(t)\rangle^2 \geq 0, \tag{5.10}$$

$$\frac{\langle I(t)^2\rangle}{\langle I(t)\rangle^2} = g_C^{(2)}(0) \geq 1, \tag{5.11}$$

que implica que para campos clásicos, siempre se cumple que $g_C^{(2)}(0) \geq 1$. Si ahora se quieren pasar esas expresiones al caso cuántico, basta con reemplazar el campo eléctrico $E(t)$ con su correspondiente operador. Este último se calcula a partir de la expresión del operador potencial vector de la ecuación (2.51), tal que, para un dado modo l,

$$\hat{E}_l(\mathbf{r},t) = i\boldsymbol{\epsilon}_l\sqrt{\frac{\hbar\omega_l}{2V\epsilon_0}}\left(e^{i(\mathbf{k}_l\cdot\mathbf{r}-\omega_l t)}\hat{a}_l - e^{-i(\mathbf{k}_l\cdot\mathbf{r}-\omega_l t)}\hat{a}_l^\dagger\right) = \hat{E}_l^{(+)}(\mathbf{r},t) + \hat{E}_l^{(-)}(\mathbf{r},t) \tag{5.12}$$

Al reemplazar esto en la ecuación (5.8) se obtiene la función de correlación de segundo orden en el caso cuántico, $g_Q^{(2)}(\tau)$, tal que

$$g_Q^{(2)}(\mathbf{r},\mathbf{r}',\tau)=\frac{\langle\hat{E}_l^{(-)}(\mathbf{r},t)\hat{E}_l^{(-)}(\mathbf{r}',t+\tau)\hat{E}_l^{(+)}(\mathbf{r}',t+\tau)\hat{E}_l^{(+)}(\mathbf{r},t)\rangle}{\langle\hat{E}_l^{(-)}(\mathbf{r},t)\hat{E}_l^{(+)}(\mathbf{r},t)\rangle\langle\hat{E}_l^{(-)}(\mathbf{r}',t+\tau)\hat{E}_l^{(+)}(\mathbf{r}',t+\tau)\rangle}. \tag{5.13}$$

donde $g_Q^{(2)}(\mathbf{r},\mathbf{r}',\tau)$ representa la probabilidad conjunta de detectar un fotón en $\mathbf{r}$ a tiempo t, y un segundo fotón en $\mathbf{r}'$ a tiempo $t+\tau$. La principal diferencia se encuentra en el hecho de que ahora $\hat{E}_l^{(-)}(\mathbf{r},t)$ es un operador, por lo que no es posible cambiar el orden del producto sin tener en cuenta las propiedades de conmutación de estos operadores.

Al igual que en el caso clásico, podemos tomar el caso particular $\tau=0$, que representa la probabilidad conjunta de detectar un fotón en dos detectores en un dado tiempo t. En este caso, omitiendo las variables $\mathbf{r}$ y $\mathbf{r}'$, la expresión anterior se reduce a

$$g_Q^{(2)}(0)=\frac{\langle\hat{a}_l^\dagger\hat{a}_l^\dagger\hat{a}_l\hat{a}_l\rangle}{\langle\hat{a}_l^\dagger\hat{a}_l\rangle^2}=\frac{\langle\hat{n}(\hat{n}-1)\rangle}{\langle\hat{n}\rangle^2}=1+\frac{(\Delta\hat{n})^2-\langle\hat{n}\rangle}{\langle\hat{n}\rangle^2}, \tag{5.14}$$

donde $\hat{n}$ es el operador número, y cumple $\hat{n}\,|n'\rangle=n'\,|n'\rangle$. Con esta expresión, y utilizando lo encontrado en la sección 5.1, se encuentra que la función de correlación de segundo orden permite distinguir entre los tres tipos de estadística de la luz, tal que

- Fuente láser: $(\Delta\hat{n})^2=\langle\hat{n}\rangle \;\rightarrow\; g_Q^{(2)}(0)=1$.
- Fuente térmica: $(\Delta\hat{n})^2=\langle\hat{n}^2\rangle+\langle n\rangle \;\rightarrow\; g_Q^{(2)}(0)=2$.
- Fuente de estados de Fock $|n\rangle$: $(\Delta\hat{n})^2=0 \;\rightarrow\; g_Q^{(2)}(0)=1-\frac{1}{n}$ para $n\geq 1$.

A partir de esto, se ve que para el caso de estados de Fock, $g_Q^{(2)}(0)<1$, que no cumple la limitación clásica de $g_C^{(2)}(0)\geq 1$. De esta manera, es suficiente con calcular $g_Q^{(2)}(0)$ para determinar la naturaleza de una fuente desconocida de fotones. Una forma de calcularla es en un experimento (fig. 5.4) en donde la emisión de la fuente incide sobre un divisor de haz, tal que, siendo T y R dos detectores que detectan los fotones transmitidos y reflejados respectivamente, se cumple

$$g_{T,R}^{(2)}(0)=\frac{\langle\hat{a}_T^\dagger\hat{a}_R^\dagger\hat{a}_R\hat{a}_T\rangle}{\langle\hat{a}_R^\dagger\hat{a}_R\rangle\langle\hat{a}_T^\dagger\hat{a}_T\rangle}, \tag{5.15}$$

donde $\hat{a}_R$ y $\hat{a}_T$ son los operadores de aniquilación de fotones reflejados y transmitidos, respectivamente. Estos operadores pueden escribirse en función del operador de aniquilación de fotones incidentes, $\hat{a}_I$. En particular, utilizando un divisor de haz 50/50, se obtiene

$$\hat{a}_T=\frac{1}{\sqrt{2}}(\hat{a}_I-\hat{a}_V)\,,\qquad \hat{a}_R=\frac{1}{\sqrt{2}}(\hat{a}_I+\hat{a}_V), \tag{5.16}$$

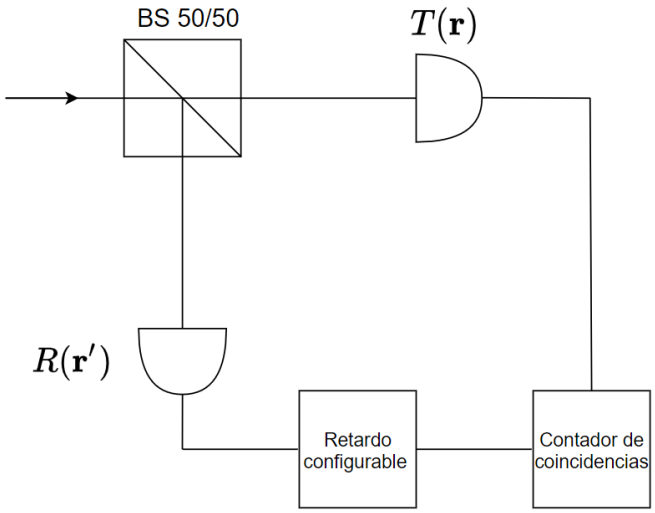


Figura 5.4: Esquema del experimento de Handbury-Brown y Twiss, utilizado para determinar $g^{(2)}(\tau)$ mediante las cuentas en coincidencias en los detectores T y R, donde el pulso medido en R es retardado un tiempo τ configurable.

donde $\hat{a}_V$ representa el operador de aniquilación de la segunda entrada al divisor de haz, por donde ingresa el estado de vacío. Reemplazando en la expresión para $g^{(2)}_{T,R}(0)$, se encuentra que

$$g^{(2)}_{T,R}(0) = \frac{\langle \hat{n}_I(\hat{n}_I - 1)\rangle}{\langle \hat{n}_I\rangle^2}, \tag{5.17}$$

por lo que a partir de las medidas realizadas con un divisor de haz, se puede obtener información de la fuente incidente. Mediante un tratamiento semi-clásico del proceso de fotodetección [85], se obtiene

$$g^{(2)}_{T,R}(\tau) = \frac{P_{TR}(\tau)}{P_T P_R}, \tag{5.18}$$

donde P_T y P_R son las probabilidades de detectar una cuenta en los detectores T y R respectivamente, y $P_{TR}(\tau)$ es la probabilidad de detectar una cuenta en el detector R a tiempo $t+\tau$, habiendo detectado una cuenta en el detector T a tiempo t; todas estas cantidades se calculan en una dada ventana de tiempo Δt. Un proceso fundamental para el cálculo de esta cantidad es poder determinar con precisión las detecciones simultáneas en ambos detectores, es decir, la medición en coincidencia.

5.3. Simulación de fuentes

Para poder tener un sistema controlable y fácilmente configurable que permita, por un lado, probar el funcionamiento del módulo contador de coincidencias con señales realistas, y por otro lado, obtener datos compatibles con un experimento óptico, se

optó por implementar una fuente de fotones simulada en la propia FPGA, de modo que así se pueden generar señales para enviar a los distintos puertos de entrada que son compatibles con una medición realista. La simulación funciona de modo que el sistema decide, aleatoriamente, cada 60 ns si emite o no un pulso. Para ello, se define una probabilidad P_{pulso} que indica la probabilidad de emitir un pulso de fotones. Sabiendo que la fuente emite $\overline{N}$ pulsos por segundo, entonces la probabilidad de emitir viene dada por

$$P_{pulso} = \overline{N} \cdot 60ns. \tag{5.19}$$

Luego, el módulo decide en forma autónoma y aleatoriamente, si el pulso emitido tiene 1 o más fotones. Suponiendo que la fuente emite n fotones por pulso con probabilidad $P_f(n)$, entonces la probabilidad de emitir 1 o más fotones es

$$P_f(n > 0) = \sum_{i=1}^{\infty} P_f(i). \tag{5.20}$$

De esta manera, el algoritmo implementado revisa cada 60 ns si un número generado aleatoriamente entre 0 y 1 es menor a P_{pulso}, y luego si otro número generado aleatoriamente, también entre 0 y 1, es menor a $P_f(n > 0)$, en cuyo caso se emite un pulso (algoritmo 1). Este componente utiliza el generador de números aleatorios Xoshiro128++, descrito en la sección 4.3. El ancho de cada pulso corresponde a 20 ns, en primera instancia, porque se corresponde con el ancho mínimo posible de generar con un reloj de 50 MHz, pero además resulta ser aproximadamente el ancho del pulso generado por un detector de fotones individuales [86].

Algoritmo 1: Pseudocódigo para la simulación de fuente de fotones sin diferenciar cantidad de fotones por pulso.

```
P_pulso = N̄ · 60ns ;
if transcurren 60 ns then
    rng1 ← número aleatorio entre 0 y 1 ;
    rng2 ← número aleatorio entre 0 y 1 ;
    if rng1 < P_pulso then
        if rng2 < P_f(n > 0) then
            emito un pulso de 20 ns
        end
    end
end
```

Para corroborar la calidad de la simulación de la fuente implementada, se obtuvo el número de cuentas para diferentes valores de $\overline{N}$, suponiendo una fuente láser. Para ello, la señal proveniente de la fuente simulada se envía a uno de los canales de entrada del módulo contador de coincidencias, obteniendo las cuentas registradas cada

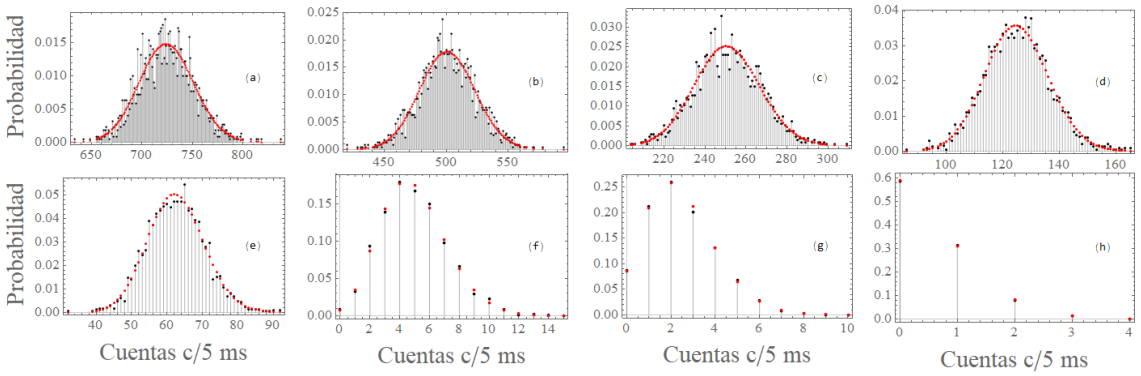


Figura 5.5: Histogramas normalizados de cuentas detectadas cada 5 ms para $\overline{N}$ =: (a) 145000 cps, (b) 100000 cps, (c) 50000 cps, (d) 25000 cps, (e) 12500 cps, (f) 1000 cps, (g) 500 cps, (h) 100 cps. En negro, los valores obtenidos utilizando la fuente láser simulada en FPGA, y en rojo la distribución teórica para dicha fuente.

5 ms mediante el contador de eventos individuales asociado (fig. 5.5). A partir de los resultados obtenidos se construyen los histogramas y se ajustan mediante las curvas teóricas correspondientes a una fuente con las mismas características que la simulada.

Para poder simular medidas de correlación (fig. 5.4) dentro de la FPGA, fue necesario implementar un divisor de haz. En este punto se vuelve necesario distinguir la cantidad de fotones de cada cada pulso emitido por la fuente, por lo que el componente cuenta nuevamente con dos generadores de números aleatorios. El primero de ellos decide la cantidad de fotones en el pulso comparando $P(n)$, con $n = 1, 2, \ldots, 10$[1], con un número generado aleatoriamente. Una vez decidida la cantidad de fotones en el pulso, un segundo generador de números aleatorios decide hacia que salida del divisor de haz envía un pulso (algoritmo 2). Por ejemplo, para un pulso de 2 fotones, el divisor de haz puede enviar ambos hacia una de las salidas, enviando 0 a la otra, o puede enviar 1 fotón para cada salida. Para implementar los dos generadores de números aleatorios, se utilizó el generador Xoshiro128++, modificando la semilla.

5.4. Determinación de $g^{(2)}(0)$

Se buscó determinar la función de correlación de segundo orden para los tres tipos de fuentes mencionados en la sección 5.1. Para ello, utilizando los algoritmos descriptos previamente (sección 5.3), se introdujeron como parámetro a la simulación de fuentes y el divisor de haz las distribuciones de probabilidad para las fuentes láser, térmica, y de estados de Fock, es decir, a partir del RNG se obtiene un número entre 0 y 1 que se compara con $P(n)$, donde $P(n)$ es la probabilidad de n fotones en un pulso para estas

[1]La limitación a pulsos de solo 10 fotones no resulta problemática para los valores de $\langle n \rangle$ utilizados. Por ejemplo, para $\langle n \rangle = 5$, la probabilidad de que un pulso tenga mas de 10 fotones es $P(n > 10) =$ 0,014.

Algoritmo 2: Pseudocódigo para un divisor de haz capaz de distinguir número de fotones por pulso. En primer paso, la variable rng1 decide la cantidad de fotones en el pulso (1 a 10), y luego rng2 decide como distribuir las salidas.

```
if detecto pulso de entrada then
    rng1 ← número aleatorio entre 0 y 1 ;
    rng2 ← número aleatorio entre 0 y 1 ;
    if rng1 < P_f(1)/P_f(n>0) (1 fotón en el pulso) then
        if rng2 < 0.5 then
            salida1 ← 1;
            salida2 ← 0;
        else
            salida1 ← 0;
            salida2 ← 1;
        end
    else if rng1 < P_f(2)/P_f(n>0) (2 fotones en el pulso) then
        if rng2 < 0.25 then
            salida1 ← 1;
            salida2 ← 0;
        else if rng2 < 0.5 then
            salida1 ← 0;
            salida2 ← 1;
        else
            salida1 ← 1;
            salida2 ← 1;
        end
    ⋮
end
```

fuentes. De esta manera, al canal A de entrada del contador de coincidencias ingresa la señal producida por la simulación de la fuente en cuestión. Esta última señal también es dirigida al módulo divisor de haz, donde las salidas del mismo ingresan como entrada a los canales B y C del contador de coincidencias. Se adoptó esta configuración de coincidencias triples en vez de la de coincidencias dobles descrita en la figure 5.4, ya que es la que se suele utilizar para la caracterización de fuentes de un fotón anunciado basadas en el proceso de SPDC [87]. Utilizando la interfaz gráfica en una PC, se configuró el módulo para medir las cuentas en coincidencia para los canales ABC, AB y AC, de manera de poder determinar $g^{(2)}(0)$ como

$$g^{(2)}(0) = \frac{P_{ABC}}{P_{AB}P_{AC}}, \tag{5.21}$$

donde P_{ABC} es la probabilidad de detectar una cuenta en coincidencia entre los canales ABC (análogo para las coincidencias entre AB y AC).

$\langle\hat{n}\rangle$	Fuente de fotones		
	Láser	Térmica	Fock
5	0.992	0.972	-
1	0.631	0.751	0
0.01	0.008	0.015	-

Tabla 5.1: Resultados de $g^{(2)}(0)$ para fuentes láser, térmica y de estados de Fock, con $\langle\hat{n}\rangle = 5$, 1, y 0,01.

Como la probabilidad de emitir n fotones para cada fuente depende de $\langle\hat{n}\rangle$, se realizaron medidas para $\langle\hat{n}\rangle = 5$, 1, y 0,01. Se obtuvo que para la fuente láser, en el caso en que $\langle\hat{n}\rangle = 5$, la función de correlación es muy cercana a 1, lo que se esperaba para este tipo de fuente. A medida que se disminuye $\langle\hat{n}\rangle$, la función de correlación disminuye tendiendo a cero. Si bien esto no es lo que se espera para una fuente láser dado que contradice la deducción de la sección 5.2, hay que remarcar que la simulación realizada supone la emisión de pulsos independientes a intervalos de tiempo regulares. Esto, en el caso de un fotón por pulso, corresponde a lo que se conoce como una fuente con *antibunching*; en cambio para un láser, el intervalo de tiempo entre fotones es aleatorio [27]. Así, aunque la simulación considera un número de fotones en cada pulso de acuerdo a la distribución de probabilidad de fotones en un láser, no se corresponde con la distribución temporal entre fotones en dicha fuente. Como consecuencia, para $\langle\hat{n}\rangle$ pequeños, luego de descartar los pulsos de cero fotones (proceso de post-selección), la estadística simulada se asemeja a la de una fuente de fotones individuales, con la única diferencia de que sigue existiendo la probabilidad de emitir 2 o más fotones por pulso. Para la fuente térmica, no se observó diferencia apreciable con lo que se obtuvo a partir de la simulación de la fuente láser en la estadística de fotodetección. Esto ocurre porque el tiempo de medida (5 ms) es mucho mayor al tiempo en el que la estadística de la ecuación (5.5) tiene efecto (60 ns), lo que hace que la estadística de la fuente térmica sea aproximadamente igual a la de un láser [88]. En el experimento comentado en la sección 5.1.4 se observa este mismo efecto. Por tal motivo, en ese experimento se recurrió a un mecanismo de decoherencia sobre una fuente láser para simular una fuente térmica cuya estadística sea apreciable por el sistema de detección. Para la fuente de estados de Fock, se implementó solo el caso $\langle\hat{n}\rangle = 1$, que equivale a una fuente de fotones individuales. En este caso, teniendo en cuenta que la simulación en FPGA no tiene en cuenta ningún tipo de ruido en los detectores, y una eficiencia de detección del 100 %, se obtuvo el resultado ideal de $g^{(2)}(0) = 0$. Las medidas realizadas para cada fuente y cada $\langle\hat{n}\rangle$ se encuentran disponibles en el siguiente link [41].

Conclusiones y próximos pasos

En este trabajo se estudiaron los fundamentos y posibles aplicaciones tecnológicas de la teoría de información cuántica. En particular, se hizo énfasis en el uso de la polarización (un grado de libertad fotónico) como qubit, mencionando los dispositivos ópticos necesarios para generar, manipular, y medir su estado cuántico. Se presentaron dos protocolos de distribución cuántica de claves, ambos pensados utilizando estados codificados en polarización. Finalmente, se estudiaron dos métodos de creación de fuentes de fotones individuales, destacando la importancia de un módulo contador de coincidencias para fuentes basadas en el fenómeno de conversión paramétrica espontánea descendente.

Se diseñó e implementó un módulo contador de coincidencias en una placa de desarrollo de prototipos FPGA. El contador de coincidencias admite cuatro señales como entrada, y el resultado son ocho señales de salida configurables. La implementación se realizó utilizando un lenguaje de descripción de hardware, VHDL, lo que permite no solo implementarlo en otra placa FPGA sin trabajo extra, si no que también hace al circuito implementado fácilmente escalable para tener más o menos entradas/salidas. Por último, se implementó una interfaz gráfica en una PC capaz de configurar los parámetros del contador de coincidencias, y de recibir los resultados, realizando un gráfico en tiempo real.

Se estudiaron los distintos tipos de generadores de números aleatorios, destacando las ventajas y desventajas de cada uno. Se presentaron las pruebas de aleatoriedad para generadores, que resultan no ser suficientes, puesto que existen generadores de números pseudoaleatorios que superan todas las pruebas sin ser realmente aleatorios. Si bien para aplicaciones de criptografía, se buscan generadores cuánticos de números aleatorios, se implementó en la placa FPGA un generador de números pseudoaleatorios basado en el generador Xoshiro128++.

En última instancia, se realizó un análisis de la estadística de distintos tipos de fuentes, mostrando los resultados experimentales obtenidos de un trabajo previo. A modo de prueba de todo lo implementado en la placa FPGA, se implementaron simulaciones de distintos tipos de fuentes, y se determinó la función de correlación de segundo orden para ellas utilizando el contador de coincidencias. Se obtuvieron los resultados esperados para la fuente de estados de Fock, en tanto que para las fuentes láser

y térmicas, si bien los resultados no reproducen el comportamiento estadístico real, los mismos son consistentes con esta primera simulación, que se ideó en este trabajo como 'aproximación de orden cero' para entender el comportamiento general de las mismas.

En el futuro, se optimizará el código del módulo contador de coincidencias, buscando además solucionar algunos errores esporádicos en la transmisión de información entre la FPGA y la PC. Una vez optimizado, se implementará este diseño permanentemente en una nueva placa no destinada al desarrollo de prototipos, para su posterior utilización en experimentos de óptica cuántica. En particular, se buscará miniaturizar el dispositivo para poder ser añadido a un sistema de comunicaciones cuánticas aeroespaciales.

Apéndice A

Histogramas para fuentes láser y térmica

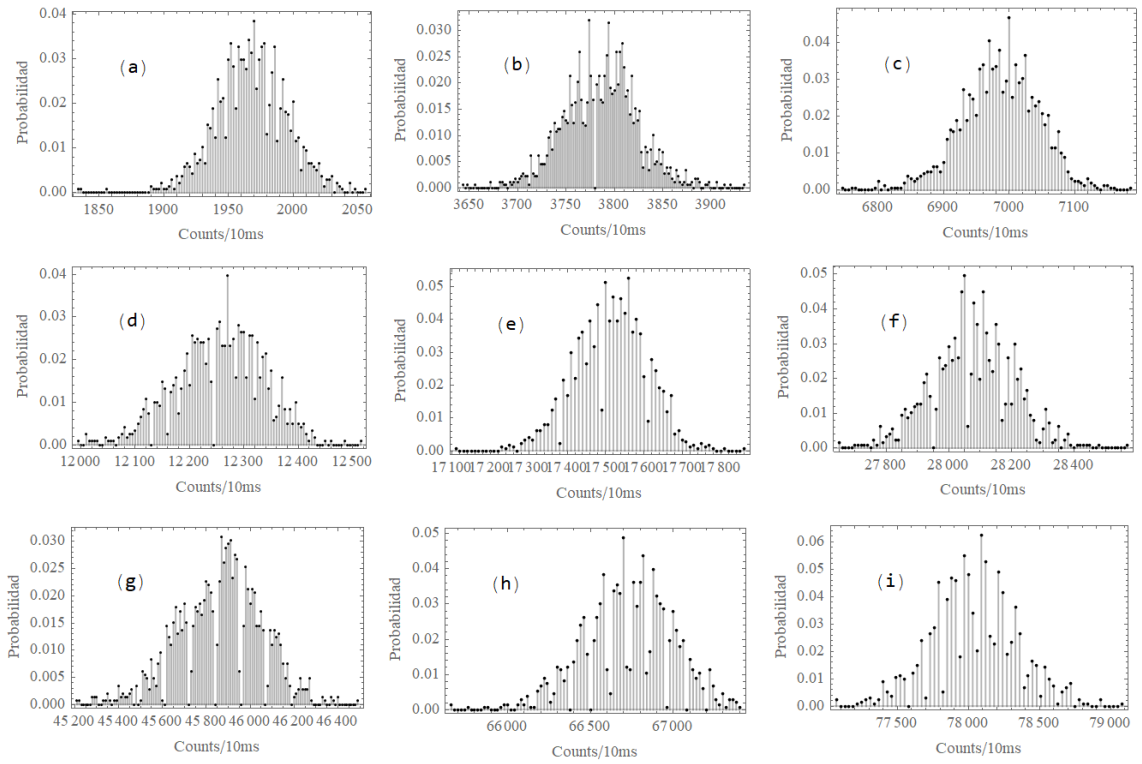


Figura A.1: Histogramas obtenidos a partir de mediciones experimentales. $P(n)$ corresponde a la probabilidad de medir n fotones en una ventana temporal de $T = 10$ ms para un láser de He-Ne con: (a) $P = 8$ mW, (b) $P = 17$ mW, (c) $P = 31$ mW, (d) $P = 52$ mW, (e) $P = 77$ mW, (f) $P = 127$ mW, (g) $P = 200$ mW, (h) $P = 290$ mW, (i) $P = 346$ mW.

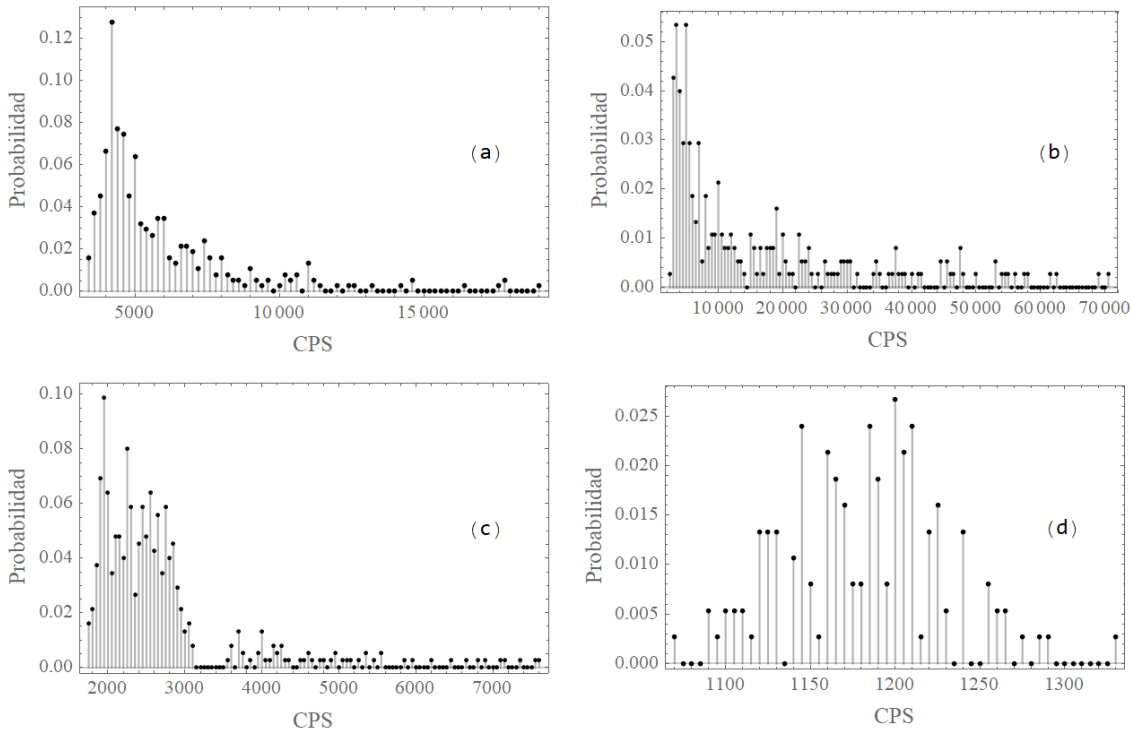


Figura A.2: Histogramas obtenidos a partir de mediciones experimentales. $P(n)$ corresponde a la probabilidad de medir n fotones para el segundo método de creación de fuente pseudo-térmica utilizando $P = 5$ pW y ventanas temporales de: (a) $T = 10$ ms, (b) $T = 50$ ms, (c) $T = 100$ ms, (d) $T = 350$ ms.

Bibliografía

[1] Gordon, J. P. Quantum effects in communications systems. *Proceedings of the IRE*, **50** (9), 1898–1908, 1962. 1

[2] Moore, G. E. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, **86** (1), 82–85, 1998. 1, 2

[3] DOMO. Data Never Sleeps 8.0. URL https://www.domo.com/learn/data-never-sleeps-8. 1

[4] Benioff, P. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of statistical physics*, **22** (5), 563–591, 1980. 2

[5] Nielsen, M. A., Chuang, I. Quantum computation and quantum information, 2002. 2, 5, 6, 10, 20, 58

[6] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, **41** (2), 303–332, 1999. 2, 9, 11

[7] Bennett, C. H., Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint arXiv:2003.06557*, 2020. 2, 14

[8] Ekert, A. K. Quantum cryptography based on bell's theorem. *Physical review letters*, **67** (6), 661, 1991. 2, 16

[9] González, P., Rebón, L., da Silva, T. F., Figueroa, M., Saavedra, C., Curty, M., *et al.* Quantum key distribution with untrusted detectors. *Physical Review A*, **92** (2), 022337, 2015. 3

[10] Yin, J., Cao, Y., Li, Y.-H., Liao, S.-K., Zhang, L., Ren, J.-G., *et al.* Satellite-based entanglement distribution over 1200 kilometers. *Science*, **356** (6343), 1140–1144, 2017. 3, 11

[11] Sparrow, C., Martín-López, E., Maraviglia, N., Neville, A., Harrold, C., Carolan, J., *et al.* Simulating the vibrational quantum dynamics of molecules using photonics. *Nature*, **557** (7707), 660–667, 2018. 3

[12] Peres, A. Quantum theory: concepts and methods, tomo 57. Springer Science & Business Media, 2006. 6

[13] Clearwater, S. H., Williams, C. P. Explorations in quantum computing. Springer Telos, 1998.

[14] Grynberg, G., Aspect, A., Fabre, C. Introduction to quantum optics: from the semi-classical approach to quantized light. Cambridge university press, 2010. 6, 24

[15] Rivest, R. L., Shamir, A., Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, **21** (2), 120–126, 1978. 11, 44

[16] Benenti, G., Casati, G., Rossini, D., Strini, G. Principles of quantum computation and information. World Scientific Publishing Company Pte Limited, 2018. 13, 17

[17] Kollmitzer, C., Pivk, M. Applied quantum cryptography, tomo 797. Springer, 2010. 14

[18] Bell, J. S. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, **1** (3), 195, 1964. 17

[19] Keiser, G. Optical fiber communications. *Wiley encyclopedia of telecommunications*, 2003. 17

[20] Slussarenko, S., Pryde, G. J. Photonic quantum information processing: A concise review. *Applied Physics Reviews*, **6** (4), 041303, 2019. 17

[21] Englert, B.-G., Kurtsiefer, C., Weinfurter, H. Universal unitary gate for single-photon two-qubit states. *Physical Review A*, **63** (3), 032303, 2001. 19

[22] Alléaume, R., Treussart, F., Messin, G., Dumeige, Y., Roch, J.-F., Beveratos, A., *et al.* Experimental open-air quantum key distribution with a single-photon source. *New Journal of physics*, **6** (1), 92, 2004. 25

[23] Kako, S., Santori, C., Hoshino, K., Götzinger, S., Yamamoto, Y., Arakawa, Y. A gallium nitride single-photon source operating at 200 k. *Nature materials*, **5** (11), 887–892, 2006. 25

[24] Maurer, C., Becher, C., Russo, C., Eschner, J., Blatt, R. A single-photon source based on a single ca+ ion. *New journal of physics*, **6** (1), 94, 2004. 25

[25] Eisaman, M. D., Fan, J., Migdall, A., Polyakov, S. V. Invited review article: Single-photon sources and detectors. *Review of scientific instruments*, **82** (7), 071101, 2011. 25

[26] Al-Kathiri, S., Al-Khateeb, W., Hafizulfika, M., Wahiddin, M. R., Saharudin, S. Characterization of mean photon number for key distribution system using faint laser. En: 2008 International Conference on Computer and Communication Engineering, págs. 1237–1242. IEEE, 2008. 25

[27] Fox, M. Quantum optics: an introduction, tomo 15. OUP Oxford, 2006. 25, 68

[28] Di Giuseppe, G., Sergienko, A., Saleh, B., Teich, M. Quantum information and computation. En: Proc. SPIE, tomo 5105, págs. 39–50. 2003. 26

[29] Verma, A., Pathak, A. Which optical processes are suitable to make probabilistic single photon sources for quantum cryptography? En: Optics and Photonics for Information Processing II, tomo 7072, pág. 70720R. International Society for Optics and Photonics, 2008. 26

[30] Lo, H.-K., Ma, X., Chen, K. Decoy state quantum key distribution. *Physical Review Letters*, **94** (23), Jun 2005. URL http://dx.doi.org/10.1103/PhysRevLett.94.230504. 27

[31] Cortiñas, R. G., Iemmi, C., Lorena, R. Detección de fotones gemelos en cristales tipo ii, 2015. 29

[32] Bock, M., Lenhard, A., Chunnilall, C., Becher, C. Highly efficient heralded single-photon source for telecom wavelengths based on a ppln waveguide. *Optics express*, **24** (21), 23992–24001, 2016. 29

[33] Boyd, R. W. Nonlinear optics. Academic press, 2020. 30

[34] Gea-Banacloche, J. Optical realizations of quantum teleportation. *Progress in optics*, **46**, 311–354, 2004. 31

[35] Kok, P., Munro, W. J., Nemoto, K., Ralph, T. C., Dowling, J. P., Milburn, G. J. Linear optical quantum computing with photonic qubits. *Reviews of modern physics*, **79** (1), 135, 2007. 31

[36] Branning, D., Beck, M. An fpga-based module for multiphoton coincidence counting. En: Advanced Photon Counting Techniques VI, tomo 8375, pág. 83750F. International Society for Optics and Photonics, 2012. 31

[37] Park, B. K., Kim, Y.-S., Kwon, O., Han, S.-W., Moon, S. High-performance reconfigurable coincidence counting unit based on a field programmable gate array. *Applied Optics*, **54** (15), 4727–4731, 2015.

[38] Branning, D., Bhandari, S., Beck, M. Low-cost coincidence-counting electronics for undergraduate quantum optics. *American Journal of Physics*, **77** (7), 667–670, 2009. 31

[39] Dillien, P. And the Winner of Best FPGA of 2016 is... *EE Times*, 2017. 33

[40] Xilinx, Inc. Xilinx Spartan-3A FPGA Platform: The World's Lowest-Cost I/O Optimized FPGAs, 2010. 33

[41] Bolaños, M. Photon coincidence counter. https://github.com/bmatiasruben/Deteccion-y-conteo-de-fotones, 2020. 36, 43, 55, 68

[42] Bromberg, Y., Lahini, Y., Small, E., Silberberg, Y. Hanbury-brown and twiss interferometry with interacting photons. *Nature Photonics*, **4**, 08 2010. 36

[43] Mijalli, M. Spartan-3an field programmable gate arrays truncated multipliers delay study. *American Journal of Applied Sciences*, **8**, 554–557, 06 2011. 37

[44] Larson, S. UART (VHDL), 2017. URL https://www.digikey.com/eewiki/pages/viewpage.action?pageId=59507062. 41

[45] Kravitz, D. W. Digital signature algorithm, jul. 27 1993. US Patent 5,231,668. 44

[46] Rivest, R. L., Shamir, A., Adleman, L. M. Cryptographic communications system and method, sep. 20 1983. US Patent 4,405,829.

[47] Hellman, M. E., Diffie, B. W., Merkle, R. C. Cryptographic apparatus and method, abr. 29 1980. US Patent 4,200,770. 44

[48] Lehmer, D. H. Mathematical methods in large-scale computing units. *Annu. Comput. Lab. Harvard Univ.*, **26**, 141–146, 1951. 44

[49] Matsumoto, M., Nishimura, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, **8** (1), ene. 1998. URL https://doi.org/10.1145/272991.272995. 44

[50] Blum, L., Blum, M., Shub, M. Comparison of two pseudo-random number generators. En: Advances in Cryptology, págs. 61–78. Springer, 1983. 44

[51] Markowsky, G. The sad history of random bits. *Journal of Cyber Security and Mobility*, **3** (1), 1–24, 2014. 45

[52] Goodin, D. Google confirms critical android crypto flaw used in $5,700 bitcoin heist. *em Ars Technia, August*, **14**, 2013. 45

[53] Dube, R. R. Hardware-based computer security techniques to defeat hackers: From biometrics to quantum cryptography. John Wiley & Sons, 2008. 45

[54] Torvalds, L. Linux Kernel drivers/char/random.c comment documentation @ 1da177e4, 2005. URL https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/drivers/char/random.c?id=refs/tags/v3.15.6#n52. 45, 46

[55] Born, M. Is classical mechanics in fact deterministic? En: Physics in my Generation, págs. 78–83. Springer, 1969. 46

[56] Herrero-Collantes, M., Garcia-Escartin, J. C. Quantum random number generators. *Rev. Mod. Phys.*, **89**, 015004, Feb 2017. URL https://link.aps.org/doi/10.1103/RevModPhys.89.015004. 47

[57] Yan, Q., Zhao, B., Hua, Z., Liao, Q., Yang, H. High-speed quantum-random number generation by continuous measurement of arrival time of photons. *Review of Scientific Instruments*, **86** (7), 073113, 2015. 47

[58] Gallager, R. G. Stochastic processes: theory for applications. Cambridge University Press, 2013. 48

[59] Ma, X., Xu, F., Xu, H., Tan, X., Qi, B., Lo, H.-K. Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction. *Physical Review A*, **87** (6), 062327, 2013. 49

[60] Sanguinetti, B., Martin, A., Zbinden, H., Gisin, N. Quantum random number generation on a mobile phone. *Physical Review X*, **4** (3), 031056, 2014. 49

[61] Ma, X., Yuan, X., Cao, Z., Qi, B., Zhang, Z. Quantum random number generation. *npj Quantum Information*, **2** (1), 1–9, 2016. 49

[62] Australian National University. Anu qrng. URL https://qrng.anu.edu.au/. 49

[63] Symul, T., Assad, S., Lam, P. K. Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Applied Physics Letters*, **98** (23), 231103, 2011. 49

[64] Haw, J. Y., Assad, S. M., Lance, A. M., Ng, N. H. Y., Sharma, V., Lam, P. K., *et al.* Maximization of extractable randomness in a quantum random-number generator. *Phys. Rev. Applied*, **3**, 054004, May 2015. URL https://link.aps.org/doi/10.1103/PhysRevApplied.3.054004. 49

[65] Ozaner Hansha, J. K. qRNG. URL https://github.com/ozanerhansha/qRNG. 50

[66] L'Ecuyer, P., Simard, R. TestU01: A C Library for Empirical Testing of Random Number Generators. **33** (4), ago. 2007. 54

[67] Doty-Humphrey, C. Practically random: C++ library of statistical tests for rngs. URL http://pracrand.sourceforge.net/. 54

[68] Brown, R. G., Eddelbuettel, D., Bauer, D. Dieharder: A random number test suite. *Open Source software library, under development*, 2013. URL https://webhome.phy.duke.edu/~rgb/General/dieharder.php. 54

[69] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Inf. téc., Booz-allen and hamilton inc mclean va, 2001. 54

[70] Marsaglia, G. Diehard: a battery of tests of randomness. *http://stat.fsu.edu/geo*, 1996. 54

[71] Marsland, S. Machine learning: an algorithmic perspective. CRC press, 2015. 55

[72] Blackman, D., Vigna, S. Scrambled linear pseudorandom number generators. *arXiv preprint arXiv:1805.01407*, 2018. 55

[73] van Rantwijk, J. Pseudo-random number generators in vhdl. URL https://github.com/jorisvr/vhdl_prng. 55

[74] Xu, X., Wang, Y. High speed true random number generator based on fpga. En: 2016 International Conference on Information Systems Engineering (ICISE), págs. 18–21. 2016. 56

[75] Majzoobi, M., Koushanfar, F., Devadas, S. Fpga-based true random number generation using circuit metastability with adaptive feedback control. En: International Workshop on Cryptographic Hardware and Embedded Systems, págs. 17–32. Springer, 2011. 56

[76] Sargent, M., Scully, M., Lamb, W. Laser Physics. Addison-Wesley, 1977. URL https://books.google.com.ar/books?id=DOVrum91zwsC. 57

[77] Blundell, S. J., Blundell, K. M. Concepts in thermal physics. OUP Oxford, 2009. 58

[78] Planck, M. K. E. L. Zur theorie des gesetzes der energieverteilung im normalspectrum. *Verhandl. Dtsc. Phys. Ges.*, **2**, 237, 1900. 58

[79] Planck, M. Ueber das gesetz der energieverteilung im normalspectrum [on the law of distribution of energy in the normal spectrum]. *Verhandlungen Deutsche Physikalische Gesellschaft*, **2**, 202–204. 58

[80] Bolaños, M. R., Pujol, J. M. Caracterización de tubo fotomultiplicador HAMAMATSU HC135-02 y análisis estadístico de fotodetección. URL https://github.com/bmatiasruben/Deteccion-y-conteo-de-fotones/tree/main/Caracterizaci%C%B3n%20HAMAMATSUU. 59

[81] Jakeman, E., Oliver, C. J., Pike, E. R. A measurement of optical linewidth by photon-counting statistics. *Journal of Physics A: General Physics*, **1** (3), 406–408, may 1968. 60

[82] Martienssen, W., Spiller, E. Coherence and fluctuations in light beams. *American Journal of Physics*, **32** (12), 919–926, 1964. 60

[83] Peacock, J. Junion honours astronomical statistics, 2012/2013. 61

[84] Glauber, R. J. The quantum theory of optical coherence. *Physical Review*, **130** (6), 2529, 1963. 62

[85] Thorn, J., Neel, M., Donato, V., Bergreen, G., Davies, R., Beck, M. Observing the quantum behavior of light in an undergraduate laboratory. *American journal of physics*, **72** (9), 1210–1219, 2004. 64

[86] Carabedo, F., Pogi, P. Puesta a punto de un sistema de detección individual de fotones. aplicación al estudio del comportamiento estadístico de una fuente pseudotérmica. *Informe Final Laboratorio 7, Universidad de Buenos Aires.* 65

[87] Grangier, P., Roger, G., Aspect, A. Experimental evidence for a photon anticorrelation effect on a beam splitter: a new light on single-photon interferences. *EPL (Europhysics Letters)*, **1** (4), 173, 1986. 67

[88] Koczyk, P., Wiewior, P., Radzewicz, C. Photon counting statistics—undergraduate experiment. *American Journal of Physics*, **64** (3), 240–245, 1996. 68

Agradecimientos

Al Consejo Interuniversitario Nacional por la beca concedida durante el período agosto 2020 a marzo 2021.

A mi familia, por inculcarme los valores que me hicieron la persona que soy hoy en día, y permitirme y fomentarme a estudiar lo que me gusta todos estos años.

A Nicky, por acompañarme y apoyarme siempre, tanto en las buenas como en las no tan buenas, siendo capaz de sacarme una sonrisa incluso en mis peores días.

A todos mis amigos, tanto de Bahía Blanca, como de La Plata, como de Quilmes, por tantas horas de juntadas, estudio, mate y ocio. En especial, me gustaría agradecer a Esti, Juan, Pau y Poldo, que me acompañaron y brindaron una ayuda gigantesca en esta última etapa, y a Fede y Dani, dos amigos incondicionales de toda la vida.

A mis directores, Lorena y Fabián, por guiarme en este último período de la carrera, y por motivarme a seguir en esta linea de investigación que es la información cuántica.