

---

# INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

INGENIERÍA ELECTRÓNICA

## INFORME TÉCNICO DE RESIDENCIA PROFESIONAL

**TEMA:**

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE  
MODULACIÓN DIGITAL DE DATOS MEDIANTE UN FPGA, PARA UN  
ENLACE ÓPTICO

**LUGAR DE REALIZACIÓN:**

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIONES SUPERIOR  
DE ENSENADA

**ASESOR INTERNO:**

DR. JORGE LUIS CAMAS ANZUETO

**ASESOR EXTERNO:**

DR. ARTURO ARVIZU MONDRAGÓN

**PRESENTA:**

MARIO ALBERTO MORALES PALACIOS

**No. DE CONTROL:**

11270484



TUXTLA GUTIÉRREZ, CHIAPAS.

DICIEMBRE / 2015

<b>ÍNDICE</b>	<b>página</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>IV</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>VII</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>JUSTIFICACIÓN .....</b>	<b>3</b>
<b>OBJETIVOS .....</b>	<b>4</b>
<b>OBJETIVO GENERAL.....</b>	<b>4</b>
<b>OBJETIVOS ESPECÍFICOS.....</b>	<b>4</b>
<b>CAPÍTULO 1. CARACTERIZACIÓN DEL ÁREA EN QUE PARTICIPÓ .....</b>	<b>5</b>
<b>1.1. DATOS GENERALES.....</b>	<b>5</b>
<b>1.1.1. Nombre o razón social .....</b>	<b>5</b>
<b>1.1.2. Ubicación.....</b>	<b>5</b>
<b>1.1.3. Giro .....</b>	<b>5</b>
<b>1.1.4. Tamaño.....</b>	<b>5</b>
<b>1.1.5. Rama.....</b>	<b>5</b>
<b>1.2. BREVE RESEÑA HISTÓRICA DE LA EMPRESA.....</b>	<b>6</b>
<b>1.3. MISIÓN, VISIÓN Y OBJETIVOS.....</b>	<b>6</b>
<b>1.3.1. Misión .....</b>	<b>6</b>
<b>1.3.2. Visión.....</b>	<b>7</b>
<b>1.3.3. Objetivos .....</b>	<b>7</b>
<b>1.4. LAYOUT .....</b>	<b>7</b>
<b>1.5. PREMIOS Y CERTIFICACIONES.....</b>	<b>8</b>
<b>1.6. RELACIÓN DE LA EMPRESA CON LA SOCIEDAD .....</b>	<b>8</b>
<b>1.7. DESCRIPCIÓN DEL DEPARTAMENTO O ÁREA DE TRABAJO .....</b>	<b>8</b>
<b>1.7.1. Nombre del departamento .....</b>	<b>8</b>
<b>1.7.2. Líneas de investigación .....</b>	<b>9</b>
<b>1.7.3. Laboratorios.....</b>	<b>9</b>
<b>1.7.4. Posgrados relacionados .....</b>	<b>10</b>
<b>CAPÍTULO 2. PROBLEMAS A RESOLVER .....</b>	<b>11</b>
<b>CAPÍTULO 3. FUNDAMENTO TEÓRICO.....</b>	<b>13</b>
<b>3.1. SISTEMAS DE COMUNICACIONES DIGITALES .....</b>	<b>13</b>
<b>3.1.1. Elementos de un sistema de comunicación óptica .....</b>	<b>13</b>
<b>3.1.2. Canales de Comunicación .....</b>	<b>14</b>
<b>3.1.3. Fibras Ópticas.....</b>	<b>14</b>
<b>3.1.4. Comunicación Óptica por el Espacio Libre.....</b>	<b>16</b>
<b>3.2. FPGAs Y LOS SISTEMAS DIGITALES.....</b>	<b>17</b>
<b>3.3. PROGRAMACIÓN HDL.....</b>	<b>18</b>

ÍNDICE (Continuación)	página
3.3.1. VHDL.....	18
3.3.2. Unidades básicas de diseño.....	19
3.4. DATOS PRBS.....	19
3.5. MODULACIÓN PPM.....	20
<b>CAPÍTULO 4. PROCEDIMIENTOS Y DESCRIPCIÓN DE LAS ACTIVIDADES.....</b>	<b>21</b>
4.1. INTRODUCCIÓN.....	21
4.2. INVESTIGACIÓN.....	23
4.2.1. WLD3343 Controlador de Propósito General para Diodos Laser.....	25
4.2.2. Diodo Laser (980nm Pump Laser Module NORTEL LC92A80-20).....	25
4.2.3. LM14S2 Base Para Diodos Laser Butterfly.....	26
4.2.4. Antena transmisora (THORLABS F810SMA-1550 Air-Spaced Colimators).....	26
4.2.5. Antena receptora (MEADE Telescope LX200-ACF).....	27
4.2.6. Fotodetector (Hamamatsu APD Module C4777-01).....	27
4.2.7. Amplificador (FEMTO DHPV4-100).....	28
4.2.8. DAQ (Adquisición de Datos).....	28
4.2.9. Otros componentes.....	29
4.3. CARACTERIZACIÓN DE LOS FPGAs Y COMPONENTES ÓPTICOS.....	30
4.3.1. Programación del FPGA.....	31
4.3.2. Caracterización de componentes eléctricos y ópticos.....	31
4.4. DISEÑO Y PROGRAMACIÓN.....	32
4.4.1. Generador de pulsos cuadrados (Divisor de frecuencia).....	32
4.4.2. Circuito Antirebote.....	34
4.4.3. Generador De Secuencia Binaria Pseudoaleatoria (PRBS).....	36
4.4.4. Codificador 8-PPM de datos PRBS.....	38
4.5. DESARROLLO EXPERIMENTAL.....	41
4.5.1. Transmisor.....	42
4.5.2. Receptor.....	46
4.6. PRUEBAS EN CAMPO.....	47
4.6.1. Ubicación del transmisor y el receptor.....	47
<b>CAPÍTULO 5. RESULTADOS.....</b>	<b>49</b>
5.1. RESULTADOS DE LA INVESTIGACIÓN TEÓRICA.....	49
5.2. FPGA.....	49
5.2.1. Resultados del Generador de pulsos cuadrados que se implementó.....	50
5.2.2. Resultados del Circuito Anti-rebote implementado.....	50
5.2.3. Generador de datos PRBS.....	51
5.2.4. Para el modulador de datos se obtuvieron los siguientes resultados.....	55
5.3. SISTEMA DE TRANSMISIÓN.....	56
5.3.1. Enlace en campo abierto.....	57

<b>ÍNDICE (Continuación)</b>	<b>página</b>
<b>CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>60</b>
<b>CAPÍTULO 7. COMPETENCIAS DESARROLLADAS Y/O APLICADAS .....</b>	<b>62</b>
<b>FUENTES DE INFORMACIÓN.....</b>	<b>64</b>
<b>ANEXO A “GUÍA PARA LA CREACIÓN DE ARCHIVOS VHDL A TRAVÉS DE XILINX ISE DESIGN SUITE” .</b>	<b>67</b>
<b>XILINX® ISE® DESIGN SUITE .....</b>	<b>67</b>
<b>PANEL DE DISEÑO (DESIGN PANE).....</b>	<b>68</b>
<b>HDL 69</b>	
<b>DESIGN DESCRIPTION .....</b>	<b>71</b>
<b>ANEXO B “TABLA DE LFSR DE GALOIS” .....</b>	<b>74</b>
<b>ANEXO C “PROGRAMAS EN VHDL” .....</b>	<b>75</b>
<b>BLOQUE DIVISOR DE FRECUENCIA .....</b>	<b>75</b>
<b>BLOQUE CIRCUITO ANTI-REBOTE .....</b>	<b>75</b>
<b>PROGRAMA GENERADOR DE DATOS PRBS .....</b>	<b>77</b>

<b>Índice de Figuras</b>	<b>página</b>
Figura 1. Ubicación del CICESE. ....	5
Figura 2. Layout CICESE. ....	7
Figura 3. Laboratorio de Comunicaciones Photonicas del CICESE.....	10
Figura 4. Diagrama que representa los elementos básicos de un sistema de comunicación óptico. ....	13
Figura 5. Enlace por fibra óptica. ....	15
Figura 6. Estructura de la fibra óptica.....	15
Figura 7. Espectro de longitudes de onda por el cual se puede transmitir con fibra óptica. ....	16
Figura 8. Láser verde de 532 nm a 5mw dirigido a una distancia de 30 m. ....	16
Figura 9. Esquema a bloques de pérdidas en un sistema FSO. ....	17
Figura 10. Diagrama que ejemplifica la estructura básica de un programa en VHDL. ....	19
Figura 11. Secuencia de tres palabras 8-PPM.....	20
Figura 12. Diagrama a bloques y breve descripción del desarrollo de actividades durante la residencia profesional. ....	22
Figura 13. Sistema de comunicación óptica a desarrollar. ....	23
Figura 14. Versiones de los FPGAS de la marca XILINX que se tienen en el CICESE.....	24
Figura 15. WLD3343 Controlador de Propósito General para Diodo Laser. ....	25
Figura 16. Laser NORTEL de 980 nm A 200mW. ....	25
Figura 17. Base para Diodos Laser tipo Butterfly.....	26
Figura 18. Colimador con entrada SMA de la marca THORLABS.....	26
Figura 19. Telescopio LX200 de MEADE.....	27
Figura 20. Módulo APD Hamamatsu C4777-01.....	27
Figura 21. Amplificador FEMTO DHPV4-100.....	28
Figura 22. Osciloscopio digital Agilent InfiniiVision DSO-X 3034A.....	28
Figura 23. Conector SMA hembra a la izquierda, Conector SMA macho a la derecha. ....	29
Figura 24. Conector DB9. ....	29
Figura 25. Conector FC/PC.....	29
Figura 26. DIP Switch tipo Tecla 8 Posiciones. ....	30
Figura 27. Software Tera Term para testear FPGAs. ....	30
Figura 28. Configuración “Corriente Constante” del controlador de corriente DWL3343. ....	31

# Índice de Figuras (Continuación)

página

Figura 29. a) Diagrama de flujo del divisor de frecuencia, b) Bloque de entrada y salida del divisor de frecuencia.....	33
Figura 30. Señal de interruptor al rebotar sus contactos internos.....	34
Figura 31. Bloque de entrada y salida del circuito anti-rebote. ....	35
Figura 32. Diagrama de flujo del programa para el circuito anti-rebote. ....	35
Figura 33. Diagrama LFSR para la generación de secuencia binaria pseudoaleatoria de 4 bits. ....	36
Figura 34. Bloque de entrada y salida del generador de datos PRBS. ....	36
Figura 35. Diagrama de flujo del programa del circuito generador de secuencias binarias Pseudoaleatorias, PRBS. ....	37
Figura 36. 8 PPM utiliza un símbolo de 8 ranuras (Slots) por cada tres bits. Se muestra la modulación del mensaje 101001. El orden en que los slots están etiquetadas no es consecucional. Aquí, la etiqueta 7 es asignado a la ranura más a la izquierda de un símbolo y la etiqueta 0 está asignada a la ranura más a la derecha. ....	38
Figura 37. M-ary PPM Modulador. ....	38
Figura 38. Bloque del codificador PPM.....	38
Figura 39. Relación entre la señal del reloj del generador con el reloj del codificador.....	39
Figura 40. Diagrama de Flujo del Modulador 8-PPM.....	40
Figura 41. Registros para la captura de bits. ....	41
Figura 42. Diagrama a bloques del generador de datos PRBS y Modulador PPM. ....	41
Figura 43. FPGA Spartan-6 LX9 MicroBoard. ....	42
Figura 44. Datos PRBS de 4 Bits de resolución (Superior) y Datos PRBS de 8 Bits de resolución (Inferior verde). ....	43
Figura 45. Circuito base para el controlador de corriente. ....	44
Figura 46. Inserción del diodo laser a la base LM14S2. ....	44
Figura 47. Diagrama eléctrico que describe las conexiones en el FPGA LX9 empleado para la transmisión de datos PRBS. ....	45
Figura 48. Fibra conectada al colimador. ....	45
Figura 49. Esquema que ejemplifica la función del telescopio como antena receptora. ....	46
Figura 50. Localización del transmisor (TX) y receptor (RX) en el mapa de Ensenada.....	47

<b>Índice de Figuras (Continuación)</b>	<b>página</b>
Figura 51. Perfil de elevación entre el transmisor y el receptor haciendo uso de herramientas de Google Earth. ....	48
Figura 52. Comparación entre señal del XTAL que presenta un FPGA (Superior) y Señal de reloj que devuelve un generador de pulsos cuadrados implementado en un FPGA (Inferior). ....	50
Figura 53. a) comportamiento de Push-Button empleado antes del implementar el circuito anti rebote, b) comportamiento del mismo Push-Button después de implementar el circuito anti rebote. ....	51
Figura 54. Imagen tomada del osciloscopio que muestra los datos PRBS de 8 bits a una frecuencia de 75 MHz. ....	52
Figura 55. Generador de datos PRBS de 4 bits (amarillo) obtenidos a partir de un reloj de 133MHz (verde) con el FPGA SP605 Spartan 6. ....	53
Figura 56. a) FFT (Fast Fourier Transform) con 5khz de SPAM de una señal PRBS diseñada a 6 bits, b) FFT con 200 Hz de SPAM de la misma señal. ....	54
Figura 57. Resultados de la Modulación PPM. ....	55
Figura 58. Vista superior a), de frente b) y trasera c) del transmisor desarrollado. ....	56
Figura 59. Vista interior a) y exterior b) del lugar donde se instaló el receptor. ....	57
Figura 60. Posición de la antena transmisora, a) y vista de la antena transmisora, b). ....	57
Figura 61. Transmisor, a), Canal, b) y Receptor, c). ....	58
Figura 62. Señal PRBS enviada, a), datos recuperados por el receptor muestreados a 10Msa/s, b), datos recuperados por el receptor muestreados a 50 Msa/s, c). ....	59
Figura A1. Navegador de proyectos (Project Navigator). ....	67
Figura A2. Icono de ISE Design Suite 14.1 ....	70
Figura A3. New Project Wizard – Creando un nuevo proyecto.....	70
Figura A4. New Project Wizard – Propiedades del dispositivo.....	71
Figura A5. New Source Wizard – Asistente de Nueva Fuente. ....	72
Figura A6. Archivo VHDL en el ISE Text Editor. ....	73
Figura B1. LFSR Galois de 8 bits con un ciclo de 255. Este LFSR tiene cuatro “taps” en las posiciones 8, 6, 5 y 4. ....	74

<b>Índice de tablas</b>	<b>página</b>
Tabla 1. FPGAs disponibles en el laboratorio de Comunicaciones Photonicas del CICESE.....	24
Tabla 2. Desempeño en frecuencia de los FPGAs disponibles en el CICESE al implementar el mismo diseño en cada uno de ellos. ....	49
Tabla 3. Recursos utilizados por el FPGA Spartan-6 Microboard para implementar el generador de PRBS 8 de bits. ....	52
Tabla 4. Recursos utilizados por el FPGA Spartan-6 Microboard para codificar datos PRBS. ....	55
Tabla 5. Competencias desarrolladas y/o aplicadas durante la residencia profesional. ....	62
Tabla B1. Registros de desplazamiento con tamaño de ciclo $2^n-1$ .....	74



## INTRODUCCIÓN

Dentro de la gran diversidad de satélites artificiales que se encuentran orbitando en el espacio existe una función que los hace semejantes y es la de establecer comunicación con una estación terrena para que de esta manera se tenga monitoreo y control de lo que está pasando en el satélite. Todos los satélites cuentan con un sistema de señales de radio y antenas receptoras y emisoras de dichas señales para establecer comunicación con la estación terrena, a lo largo de la historia se han utilizado ondas de radio muy potentes por su facilidad de penetrar la atmosfera terrestre, sin embargo hay satélites puestos en órbita baja terrestre que necesitan transmisores menos potentes para transferencia de datos y por esta razón la órbita baja se usa para muchas aplicaciones de comunicación. En el 2004 la NASA dio a conocer su visión futurística a 30 años en la que describe la estructura de las comunicaciones en donde los enlaces ópticos sustituyen a los enlaces con señales de radio (RF). Actualmente la comunicación entre satélites artificiales mediante enlaces FSO (Free Space Optic) ya es posible y están siendo utilizadas.

En la división de Física Aplica del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada), específicamente en el departamento de Electrónica y Telecomunicaciones se hace investigación tanto teórica como experimental en el are de las comunicaciones cuánticas; en conjunto con el área de comunicaciones satélites, se desarrolla el proyecto “Enlace óptico en espacio libre de larga distancia para aplicaciones en comunicaciones ópticas cuánticas satelitales” que trata de establecer la comunicación entre satélites geoestacionarios y estaciones terrenas con la ayuda de la óptica cuántica.

Investigadores del CICESE e INAOE (Instituto Nacional de Astrofísica, Óptica y Electrónica) implementan, caracterizan y prueban enlaces ópticos en espacio libre, estos enlaces con potencia óptica clásicas son un paso previo para la implementación de enlaces cuánticos atmosféricos plantados como una actividad del proyecto antes mencionados.

Este trabajo explica las actividades realizadas durante la residencia profesional para desarrollar uno de los enlaces ópticos en espacio libre utilizando tecnología FPGA (Field Programmable Gate Array, Arreglo de Compuertas Programables en Campo), de ahí el nombre del proyecto. Iniciando con la justificación y objetivos generales y específicos. El capítulo 1 “caracterización del área en que participó” describe el lugar físico en donde se

realizó la residencia destacando aspectos del laboratorio, instrumentación y equipo del CICESE. El capítulo 2 “Problemas a resolver” describe la problemática que se abordó durante el desarrollo del proyecto, El capítulo 3 “Fundamento teórico” contiene las bases teóricas fundamentales de ayuda para desarrollar el proyecto. El capítulo 4 “Procedimiento y descripción de las actividades” como su nombre lo indica se describe detalladamente el procedimiento seguido para cumplir con el objetivo del proyecto, y describe como se ejecutaron las tareas de desarrollo experimental en laboratorio y en campo. El capítulo 5 “Resultados” presenta lo que se obtuvo de hacer las pruebas de campo. En el capítulo 6 “Conclusiones y resultados” se definen conclusiones y recomendaciones acerca del proyecto y de la estancia en el CICESE. El capítulo 7 “competencias desarrolladas o aplicadas” muestra una tabla en donde se describen, conocimientos previos aplicados, conocimiento adquiridos, amplitudes y actitudes, etc. Y en seguida se muestran las fuentes de información consultadas durante el desarrollo del proyecto.

## JUSTIFICACIÓN

Recientes progresos en las comunicaciones ópticas en el espacio libre con detección directa la hacen parecer como una alternativa importante a los enlaces radioeléctricos y/o alámbricos convencionales. Esto es debido a su capacidad potencial para operar a altas velocidades de bit y requiriendo antenas (ópticas) que son mucho más pequeñas que las de radiofrecuencia para una ganancia de antena dada [Willebrand y Ghuman, 2000 citado por Muraoka, 2005].

Este proyecto es derivado de actividades de investigación científica que se hace en el área de las comunicaciones ópticas, en donde se implementan, caracterizan y prueban diferentes enlaces ópticos atmosféricos. Con este proyecto se pretende desarrollar, un sistema de comunicación óptica en espacio libre utilizando la óptica y electrónica disponible en CICESE además de usar tecnología FPGA dentro del sistema de transmisión.

La utilización de un FPGA pretende caracterizar y probar el funcionamiento de este mismo como generador de datos y modulador de datos para posteriormente ser implementado en un sistema de comunicación FSO. Además de que se pretende caracterizar el desempeño de los diversos subsistemas que se han desarrollado en el laboratorio de comunicaciones ópticas del CICESE como parte del proyecto “Enlace óptico en espacio libre de larga distancia para aplicaciones en comunicaciones ópticas cuánticas satelitales”.

# OBJETIVOS

## Objetivo general

Desarrollar un sistema de comunicaciones óptico en espacio libre para transmitir señales digitales PPM (Pulse Position Modulation, Modulación por posición de Pulso) generadas a partir de un FPGA, utilizando un láser de 980 nm y una cadena de datos PRBS (Pseudorandom Binary Sequence, Secuencia Binaria Pseudoaleatoria) de tamaño  $10^{-12}$ , a una velocidad máxima de 200 MBits por segundo.

## Objetivos específicos

- Revisar la teoría de los sistemas de comunicación digital y los diferentes tipos de codificación de datos.
- Revisar y Evaluar las características del FPGA a utilizar.
- Programar el FPGA para la generación de datos pseudoaleatorios (PRBS) de 12 bits.
- Programar el FPGA para la modulación por posición de pulsos de datos PRBS.
- Analizar los datos generados con el FPGA en el dominio del tiempo y la frecuencia.
- Armar el sistema trasmisor de comunicación laser.
- Armar el sistema receptor de comunicación laser.
- Evaluar la modulación de datos en un esquema de transmisión en espacio libre.

# CAPÍTULO 1. CARACTERIZACIÓN DEL ÁREA EN QUE PARTICIPÓ

## 1.1. Datos generales

### 1.1.1. Nombre o razón social

Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE).

### 1.1.2. Ubicación

Carretera Ensenada-Tijuana No. 3918, Zona Playitas, C.P. 22860, Ensenada, B. C. México.

Teléfono: 01(646)175-05-00.

En la Figura 1 se delimita la ubicación geográfica del CICESE.



Figura 1. Ubicación del CICESE.

### 1.1.3. Giro

De servicios.

### 1.1.4. Tamaño

Grande.

### 1.1.5. Rama

Educación.

## **1.2. Breve reseña histórica de la empresa**

El Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) fue la segunda institución creada por el Consejo Nacional de Ciencia y Tecnología (CONACYT) para descentralizar las actividades científicas y tecnológicas en México.

El decreto presidencial de creación del CICESE, publicado el 18 de septiembre de 1973, lo define como un organismo descentralizado de interés público, con personalidad jurídica y patrimonio propios para realizar “investigación científica básica y aplicada inicialmente en los campos de la geofísica, oceanografía física, física e instrumentación, principalmente orientadas a la a la solución de problemas nacionales y en particular a los regionales de la península de Baja California, así como a las actividades docentes en estas áreas de la ciencia en los niveles de maestría y doctorado”.

El contexto que permitió crear al CICESE a principios de los setentas es muy diverso, pero destacan, entre otros aspectos: una política nacional por descentralizar la investigación científica, la presencia de la Escuela Superior [hoy Facultad] de Ciencias Marinas de la Universidad Autónoma de Baja California (UABC) y la cercanía del Scripps Institution of Oceanography (SIO); la decisión, en 1970, de la UNAM de construir el Observatorio Astronómico Nacional en la sierra de San Pedro Mártir; la intensa actividad tectónica y sísmica de la península de Baja California y del golfo de California que justificaba la realización de estudios en las ciencias de la Tierra, y el requerimiento por desarrollar instrumentación electrónica y óptica como apoyo a la UNAM y a la investigación oceanográfica y geofísica del nuevo centro.

## **1.3. Misión, Visión y Objetivos**

### **1.3.1. Misión**

Generar y transferir conocimiento científico de frontera y desarrollar tecnología para contribuir a la solución de problemas regionales, nacionales y universales; así como formar recursos humanos a nivel de posgrado en ciencias biológicas, físicas, de la información, del mar y de la tierra, dentro de un marco de responsabilidad, ética y liderazgo en beneficio de la sociedad.

### 1.3.2. Visión

Ser líderes y referentes a nivel nacional e internacional por la calidad e impacto de nuestras investigaciones y de los recursos humanos que formamos; consolidar las disciplinas emergentes en energías alternativas, desarrollo biomédico, ciencia de datos, cambio climático y sustentabilidad ambiental, liderando proyectos multidisciplinarios de gran envergadura.

### 1.3.3. Objetivos

- Generar conocimiento científico a través de proyectos de investigación en las áreas de especialidad del centro.
- Formar recursos humanos a nivel de maestría y doctorado en las áreas de especialidad del centro a través de programas de posgrado de calidad reconocida.
- Fortalecer la vinculación con los sectores público, privado y social a través de proyectos de investigación y desarrollo, servicios tecnológicos, de consultoría y programas de capacitación.

### 1.4. Layout

En la Figura 2 se muestran las zonas y departamentos que conforman el CICESE.



Figura 2. Layout CICESE.

## **1.5. Premios y certificaciones**

Premio Innovaciones en Redes de Alto Rendimiento en Aplicaciones de Investigación 2012, que otorga la Corporación para Iniciativas en Redes de Educación de California (CENIC), por el uso de redes avanzadas en temas de colaboración fronteriza.

## **1.6. Relación de la empresa con la sociedad**

El CICESE es un Centro Público de Investigación del sistema de centros del Consejo Nacional de Ciencia y Tecnología de México (CONACYT). La institución tiene 40 años de experiencia dedicada a la investigación, formación de recursos humanos de alto nivel y a la vinculación en disciplinas relacionadas con las ciencias del mar, de la tierra, de la vida, de la física y de la información. El CICESE se encuentra localizado en una de las regiones de innovación transfronteriza más dinámicas a nivel mundial, la región Baja California-California integra instituciones educativas y polos de desarrollo aeroespacial, dispositivos médicos y manufactura electrónica avanzada de primer nivel.

Dado que la misión de CICESE es generar conocimiento y tecnología que contribuya a la solución de problemas universales, nacionales y regionales, realizando labores de investigación básica, aplicada e innovación, la inserción y contribución de la Sociedad a la Información es un factor imprescindible en sus actividades.

Desde su origen, CICESE ha sido un motor nacional en el desarrollo, aplicación y adopción de las Tecnologías de la Información y la Comunicación (TIC). Desde el surgimiento y penetración del Internet en el entorno académico y tejido social, CICESE ha sido un actor clave, no solo en los aspectos académicos, sino también en el desarrollo de políticas públicas que coadyuvan al tránsito de México hacia la Sociedad de la Información contribuyendo con proyectos que han impactado en la reducción de la brecha digital, generando conocimiento en tecnologías de punta en telecomunicaciones, informática y ramas afines con aplicaciones a la salud, la educación y otras disciplinas socioeconómicas.

## **1.7. Descripción del departamento o área de trabajo**

### **1.7.1. Nombre del departamento**

División Física Aplicada: Departamento de Electrónica y Telecomunicaciones.



En este departamento, se realizan actividades de investigación de excelencia tanto teórica como experimental en las áreas de la electrónica y las telecomunicaciones, para cumplir con los objetivos específicos de generar y aplicar conocimiento a través de la realización de proyectos de investigación básica, aplicada y de desarrollo tecnológico, así como contribuir a la formación de recursos humanos con nivel maestría y doctorado altamente capacitados en las áreas de la especialidad.

La electrónica y las telecomunicaciones, aparte de ser disciplinas prioritarias para el desarrollo del país, muestran una gran dinámica de avance que requiere de la preparación de recursos humanos altamente capacitados que puedan contribuir al dominio de dichas tecnologías con el fin de ofrecerla a los cuadros productivos, docentes, de investigación y de desarrollo en México.

### **1.7.2. Líneas de investigación**

- Altas frecuencias
- Comunicaciones ópticas
- Control
- Instrumentación óptica
- Redes de telecomunicaciones

### **1.7.3. Laboratorios**

- **Comunicaciones Fotónicas**
- Laboratorio de Algoritmos Aplicados a Comunicaciones Móviles
- Laboratorio de Altas Frecuencias
- Laboratorio de Control
- Laboratorio de Control Robusto
- Laboratorio de Procesamiento de Señales para Comunicaciones
- Laboratorio de Robótica
- Laboratorio de Sincronización y Sistemas Complejos
- Taller de Electrónica del Departamento de Electrónica y Telecomunicaciones.

En la Figura 3 se muestra una imagen del laboratorio de Comunicaciones Photonicas que está dentro del edificio de Física Aplicada.



**Figura 3. Laboratorio de Comunicaciones Photonicas del CICESE.**

#### **1.7.4. Posgrados relacionados**

- Electrónica y Telecomunicaciones.

## CAPÍTULO 2. PROBLEMAS A RESOLVER

Si bien es cierto que un sistema de comunicación óptico en espacio libre en una estación terrena no es igual a un sistema de comunicación óptica satelital, sin embargo es posible evaluar características que son susceptibles de reproducir en un sistema de transmisión terrestre, como lo es la recolección de información en el espacio que son presentadas como datos aleatorios debido a la misma naturaleza de la señal que se está monitoreando, el equipo necesario para hacer que se logre la comunicación exitosamente o la atenuación que sufre la señal al ser enviado por el espacio libre debido a los diferentes tipos de ruido, entre otras características que podría presentar un sistema de comunicación óptica satelital. Ahora bien las interrogantes surgen al querer reproducir este sistema, simular la fuente de datos, implementar una tipo de codificación y modulación para los datos y así poder ser enviados por el espacio libre.

Gracias a la investigación que se viene realizando dentro del área de Electrónica y Comunicaciones, se puede encontrar información de mucha ayuda para responder estas interrogantes, es por eso que se sabe de la gran utilidad de los sistemas embebidos en tareas de comunicaciones y procesamiento de señales, de los tipos de modulación y las comunicaciones en espacio libre, y de más información de gran relevancia para este problema.

Sin embargo el hecho de que se sepa que un tipo de modulación funciona para el esquema de comunicación que deseamos implementar, no necesariamente se sabe que esa es la correcta para implementar en nuestro sistema, o puede que sepamos que un FPGA puede procesar y codificar señales pero no sabemos, ¿Qué tipo de FPGA es el adecuado?, ¿A qué velocidad puede procesar las señales?, ¿Cuál es su rango de operación?, y es aquí donde podemos definir específicamente a que problema nos enfrentamos.

¿Se puede desarrollar un sistema de modulación digital de datos mediante un FPGA, para un enlace óptico en espacio libre?

¿Que FPGA es el más adecuado para implementar un sistema de generación y codificación de datos, dentro de un sistema de comunicación óptica?

¿Es posible generar una cadena de datos pseudoaleatoria PRBS de  $10^{12}-1$  a una velocidad de 200 Mbits por segundo a partir de un FPGA?

¿Es posible codificar datos PRBS en un formato 8-PPM a partir de un FPGA?

¿Qué tan eficiente es la transmisión de datos PRBS codificados en un formato 8-PPM en espacio libre en términos de ruido, distorsión por retardo y atenuación?

## CAPÍTULO 3. FUNDAMENTO TEÓRICO

### 3.1. Sistemas de comunicaciones digitales

Un sistema de comunicación digital está compuesto por diferentes elementos empezando con la fuente de información que puede provenir desde el espacio exterior y generalmente se trata de señales analógicas, tal es el caso del audio y el video, estas señales de tipo analógicas son transformadas a señales digitales discretas mediante procesos de muestro ya preestablecidos dentro de la literatura de comunicaciones digitales.

Lo que se consigue a través del muestreo es una serie de datos que por la misma naturaleza de la señal es de comportamiento aleatorio, es decir que no tiene un patrón de repetición característico.

#### 3.1.1. Elementos de un sistema de comunicación óptica

Dentro de los elementos que conforman un sistema de comunicación óptico se destacan los que se muestran en el diagrama de la Figura 4.

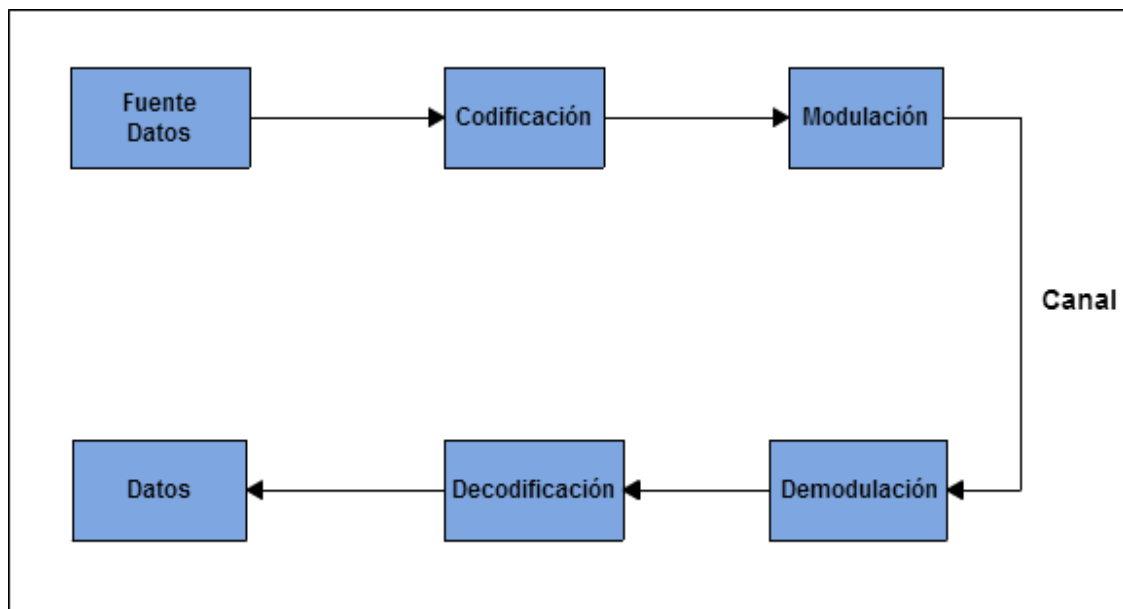


Figura 4. Diagrama que representa los elementos básicos de un sistema de comunicación óptico.

Los datos que se tienen en el bloque de la fuente generalmente pueden provenir del medio exterior, pudiendo ser las señales que recogen un sensor de presión, temperatura, velocidad,

entre otros. Estos datos tienen que ser codificados para su eficiente transmisión por el canal; un tipo de codificación que por sus características resulta idóneo para las comunicaciones ópticas es la codificación PPM de la cual se hablará posteriormente, hasta aquí únicamente se tienen señales eléctricas que son interpretadas como datos que pueden ser 0s y 1s, posteriormente se tienen que convertir en señales ópticas para poder ser transmitidas, y es aquí donde entra en juego la etapa de modulación en donde se convierten y controlan los datos para así poder ser enviados por el canal de transmisión, las etapas de demodulación y decodificación presentes en la parte del receptor son el proceso inverso de sus homólogos [Artes y Pérez, 2012].

### **3.1.2. Canales de Comunicación**

La existencia de un canal es la razón que justifica la existencia de un sistema de comunicaciones, y la forma en que un canal degrada la señal que ponemos en su entrada es la guía para el diseño del sistema de comunicaciones

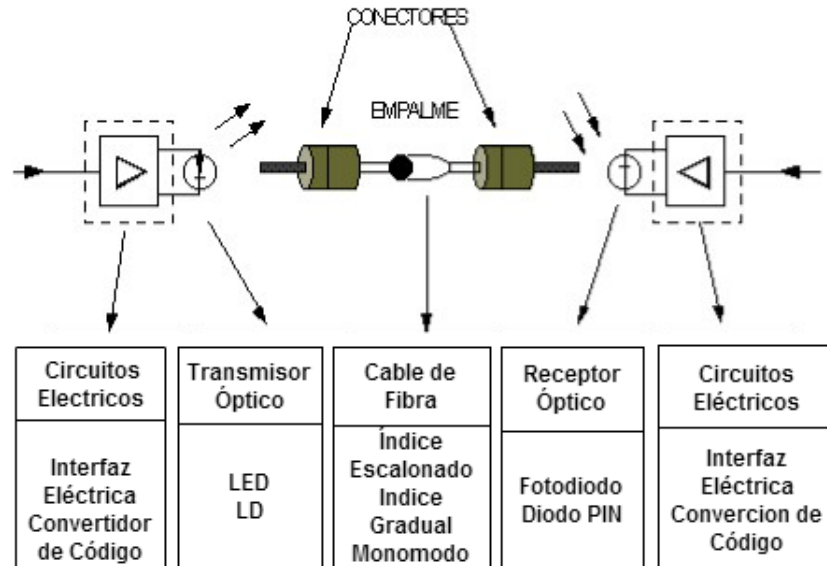
Un primer tipo de canal es aquel que emplea para la transmisión un conductor eléctrico, tal como el par de hilos de cobre que conecta una terminal telefónica con la central, un cable coaxial que conecta un modem de una red de cable con su distribuidor correspondiente, los ocho hilos de un cable de pares trenzados, o los cuatro hilos paralelos de un cable USB (Universal Serial Bus). Los principales mecanismos de degradación que introduce este medio son la atenuación de la señal puesta a su entrada y la adición del denominado ruido térmico.

Un segundo tipo de canal es el denominado canal radioeléctrico o canal radio, que engloba todas aquellas transmisiones electromagnéticas en el espacio libre. Suele dividirse en canal radio fijo (Si tanto la antena transmisora y receptora están en una ubicación física fija) y canal radio móvil (Si al menos una de las antenas está en movimiento).

Los mecanismos de propagación electromagnética son distintos en bandas distintas de frecuencia, pero en espacio libre el efecto es solo una atenuación (Aunque puede ser a frecuencias distintas) [Artes y Pérez, 2012].

### **3.1.3. Fibras Ópticas**

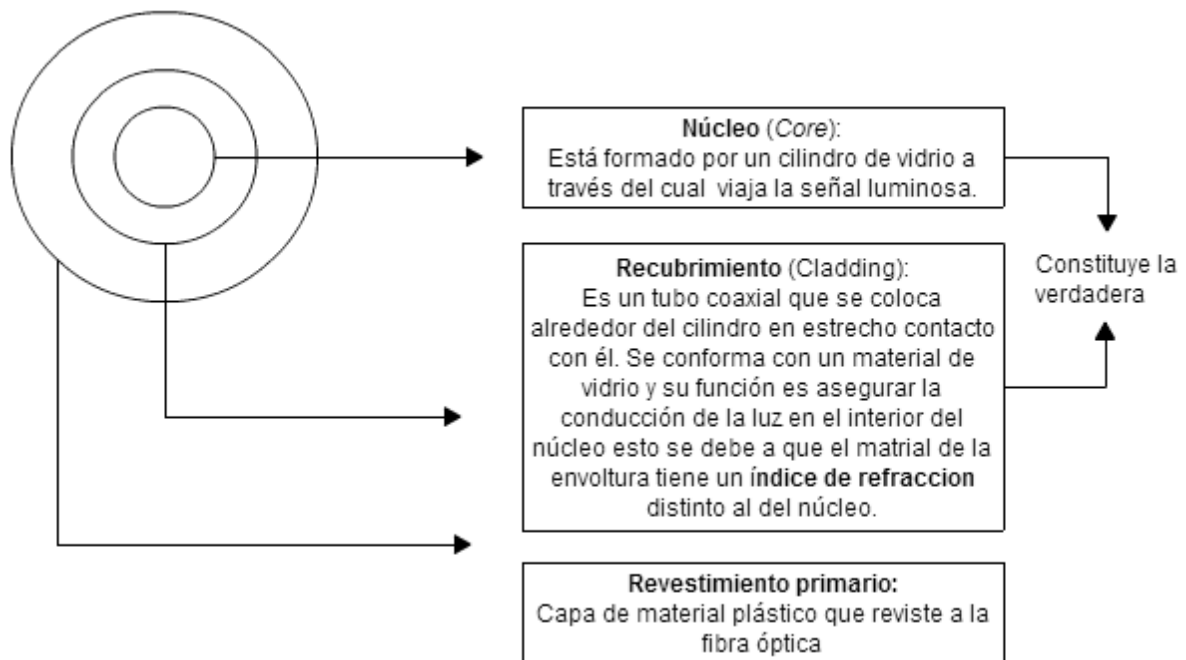
El sistema de fibras ópticas funcionan enviando información por medio de rayos de luz. Para esto se compone de un dispositivo de foto emisor que convierte los impulsos eléctricos en rayos de luz, un canal óptico por donde la luz transita y un dispositivo foto detector que vuelve a transformar la señal luminosa en impulsos eléctricos así como se muestra en la Figura 5.



**Figura 5. Enlace por fibra óptica.**

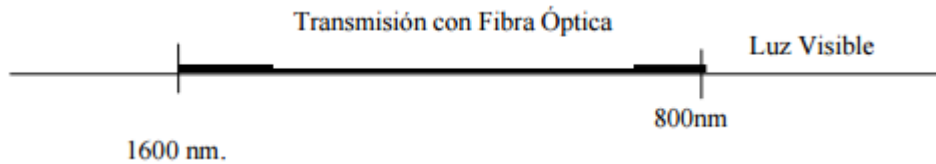
La fibra óptica es una guía de señales ópticas y tiene la particularidad de encaminar la luz, incluso en un recorrido curvilíneo. Estructura:

La fibra óptica se compone de tres partes: Núcleo, Recubrimiento y Revestimiento, Figura 6.



**Figura 6. Estructura de la fibra óptica.**

Para la transmisión por fibra óptica se utilizan las longitudes de onda del infrarrojo, o sea 800 a 1600 nm, siendo los valores más utilizados los de 850, 1300, 1550 nm tal como se ilustra en la Figura 7. Estas longitudes son en donde están las de menores interferencias [Santa, 2012].

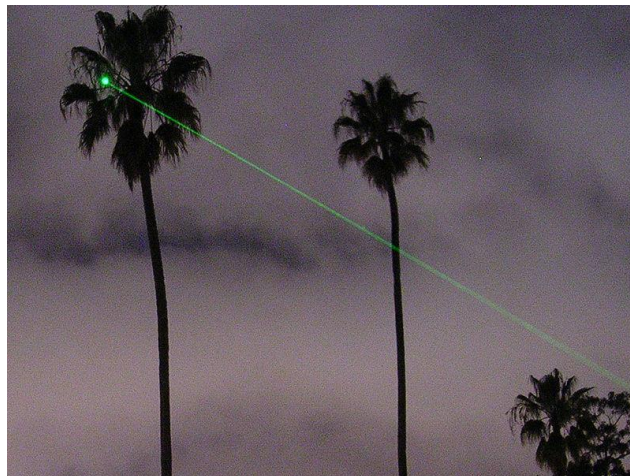


**Figura 7. Espectro de longitudes de onda por el cual se puede transmitir con fibra óptica.**

### 3.1.4. Comunicación Óptica por el Espacio Libre

En telecomunicaciones, la óptica de espacio libre (FSO, siglas en inglés de Free-Space Optical), es una tecnología de comunicación óptica que utiliza la propagación de la luz (visible o infrarroja) en la atmósfera para transmitir información entre dos puntos. Al igual que las redes de fibra óptica, esta tecnología utiliza un diodo emisor de luz o un láser como fuente de transmisión, aunque no necesita que el haz de luz sea guiado a través de cables ópticos. Para recibir la señal, los haces de luz se centran en un lente de recepción conectada a un receptor de alta sensibilidad a través de un cable de fibra óptica.

En la Figura 8 se muestra un láser verde de 532 nm que es dirigido a una distancia de 30m en donde se puede apreciar el ensanchamiento que sufre el haz de luz al propagarse en el espacio [1].



**Figura 8. Láser verde de 532 nm a 5mW dirigido a una distancia de 30 m.**



En la Figura 9 se muestra un esquema a bloques de este tipo de sistemas de comunicación y las pérdidas que este puede sufrir.

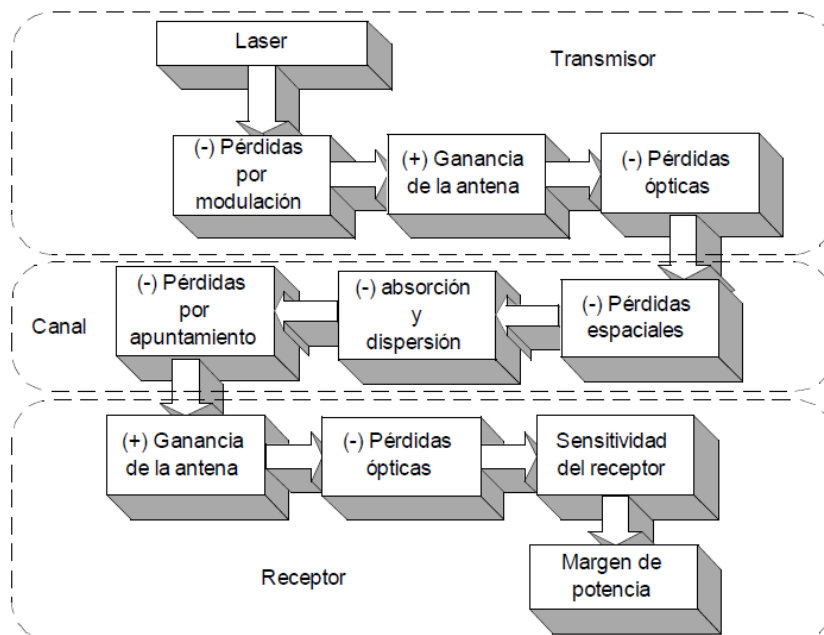


Figura 9. Esquema a bloques de pérdidas en un sistema FSO.<sup>1</sup>

### 3.2. FPGAs y los sistemas digitales

Los FPGAs, facilitan el desarrollo e implementación de sistemas, digitales complejos. Son dispositivos que combinan muchas de las novedades en el diseño de circuitos integrados para la implementación de sistemas digitales, y brindan la posibilidad de ajustarse a necesidades individuales, definidas por el usuario.

Tienen gran impacto dentro de las Telecomunicaciones debido a su alta capacidad lógica y procesamiento de señales, además de funcionalidades que son programables por el usuario [Checa y Velázquez, 2011].

Las aplicaciones típicas son:

- Lógica aleatoria.
- Dispositivos controladores
- Procesamiento digital de señales.

<sup>1</sup> Figura tomada de la Tesis "Comunicaciones Ópticas Coherentes en el Espacio Libre" del autor RAMÓN MURAOKA ESPÍRITU.

- Procesamiento de imagen audio y video.
- Soluciones militares
- Telecomunicaciones
- Redes.
- Sistemas con bloques SRAM (Static Random Access Memory).
- Supercomputadoras.
- Emulación de hardware de computadora.

### **3.3. Programación HDL**

Un lenguaje de descripción de hardware (HDL, Hardware Description Language) es similar a un lenguaje de programación típico, salvo que en HDL sirve para describir hardware en lugar de un programa que la computadora ejecutará. Hay muchos HDL comerciales. Algunos son sujetos a derechos de propiedad pero en este trabajo se utiliza solo un lenguaje que apoyan prácticamente todo los comercios que ofrecen tecnología de hardware digital y oficialmente se respalda como una norma del Instituto De Ingenieros Eléctricos Y Electrónicos (IEEE, Intitute of Electrical and electronics Enginners).

Dos HDL son normas de IEEE: VHDL (Very High Speed Integrated Circuit Hardware Description Language): Lenguaje de descripción de hardware de circuitos integrados de muy alta velocidad) y Verilog VHDL. Ambos lenguajes tiene amplio uso en la industria de los FPGAs [Sánchez, 2014].

#### **3.3.1. VHDL**

Lenguaje de descripción de hardware de circuitos integrados de muy alta velocidad creado por el Departamento de Defensa de Estados Unidos como parte del programa Very High Speed Integrated Circuits a principios de 1980.

Es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales. Así, los lenguajes de descripción de hardware hacen posible una descripción formal de un circuito electrónico, y posibilitan su análisis automático y su simulación.

Es importante saber que este tipo de programación presenta varias características de diseño que lo hacen idóneo para implementar sistemas digitales, rescatando alguna de ellas están las de representar diagramas lógicos, expresiones booleanas o circuitos digitales en pocas palabras está orientado a la descripción y modelado de sistemas digitales.

### 3.3.2. Unidades básicas de diseño

La estructura general de un programa en VHDL está formada por módulos o unidades de diseño, cada uno de ellos compuesto por un conjunto de declaraciones e instrucciones que definen, describen, estructuran, analizan y evalúan el comportamiento de un sistema digital. Existen cinco tipos de unidades de diseño en VHDL: declaración de entidad (entity declaration), arquitectura (architecture), configuración (configuration), declaración del paquete (package declaration) y cuerpo del paquete (package body). En el desarrollo del programa del en VHDL pueden utilizarse o no tres de los cinco módulos, pero dos de ellos (entidad y arquitectura) son indispensables en la estructuración de un programa, en la Figura 10 se da un ejemplo de la estructura de un programa genérico [Maxinez y Alcalá, 2002].

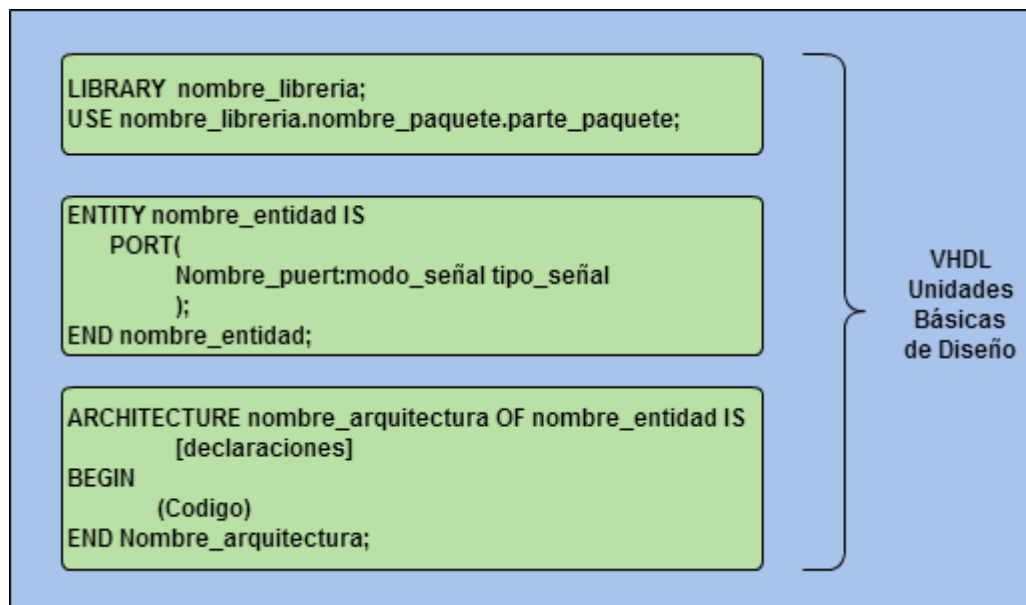


Figura 10. Diagrama que ejemplifica la estructura básica de un programa en VHDL.

### 3.4. Datos PRBS

Un generador de secuencia binaria pseudoaleatoria, SBSA, (En inglés, Pseudo Random Binary Sequence, PRBS) es un circuito que genera una serie de números binarios de n-bits, un número por ciclo de reloj, sin seguir un patrón determinado, pero que se repite luego de  $2n$  ciclos de reloj. En un sistema digital un PRBS se implementa como un registro de desplazamiento de realimentación lineal (en inglés Linear Feedback Shift Register, LFSR). Dicho de esta manera con el término PBRS se describe lo que el circuito hace, mientras que con el término LFSR se describe como el circuito esta implementado [2].

### 3.5. Modulación PPM

Modulación por posición de pulso (Pulse Position Modulation), esta modulación consiste en codificar una palabra de  $M$  bits, transmitiendo un solo pulso que puede estar en cualquiera de las  $2^M$  posiciones posibles, esto se repite cada  $t$  segundos, por lo cual la tasa de transmisión es de  $M/t$  bits por segundo.

Una de las ventajas de utilizar este tipo de modulación está en que la identificación del símbolo no se realiza teniendo en cuenta referencias externas, sino simplemente comparando el nivel de la señal de entre los diferentes slots temporales de un periodo de símbolo.

En la Figura 11 se muestra una secuencia corta de tres símbolos 8PPM. El dibujo a) representa la señal ideal y b) representa la señal después de haber sufrido los efectos de desvanecimiento y el ruido en el receptor [Álvarez, 2012].

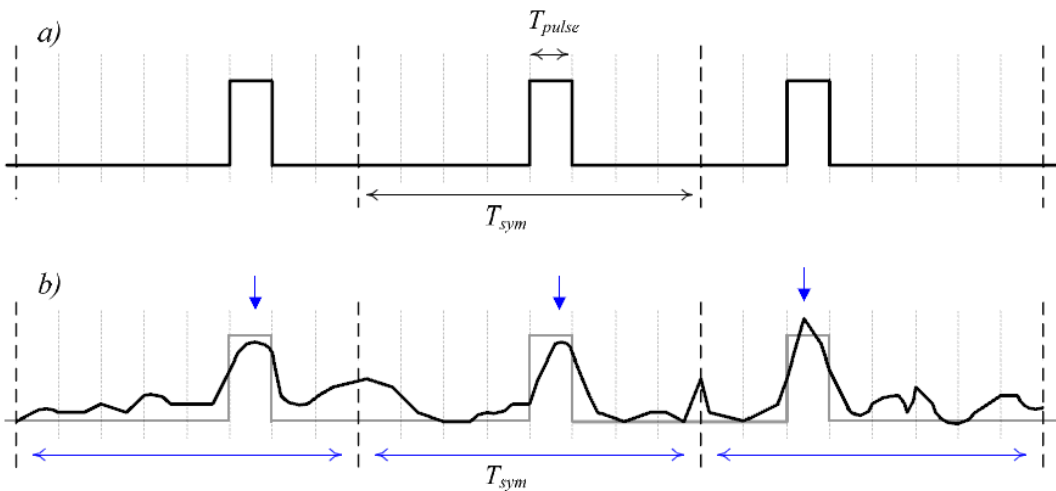


Figura 11. Secuencia de tres palabras 8-PPM.

## **CAPÍTULO 4. PROCEDIMIENTOS Y DESCRIPCIÓN DE LAS ACTIVIDADES**

### **4.1. Introducción**

En esta apartado se describen todas las actividades y procedimientos que se siguieron para desarrollar el proyecto durante la residencia profesional empezando con la Investigación Teórica que trata en recopilar información y datos que den cierta valides al proyecto que está a punto de desarrollarse, lo que se busca con esta primer paso es evitar repetir hipótesis o planteamientos ya trabajados.

Al tratarse del desarrollo de un sistema de comunicación óptico en la que la parte principal es el FPGA es necesario enfatizar en el control y funcionamiento de este, ya que del FPGA depende la generación y codificación de datos que va a ser transmitidos por el canal de comunicación; es por esta razón que la siguiente etapa es la caracterización de los FPGA disponibles.

Luego está la etapa del diseño y programación del generador PRBS en el FPGA adecuado, con las características que se desean y el control sobre estas mismas; en esta misma etapa se diseña y programa la codificación PPM de los datos PRBS en el FPGA.

El desarrollo experimental es donde se construye el sistema de comunicación óptico en un esquema preliminar antes de hacer las pruebas en campo, aquí se implementa el FPGA como parte del transmisor y se hacen el armado de todo el conjunto de componentes que integran el sistema. Para la descripción se dividido en dos secciones el Transmisor y el Receptor. Durante este proceso es posible que surjan nuevos requerimientos que no se tenían contemplados durante la etapa del diseño, es por ello que se tiene que regresar y hacer ajustes en el diseño y proseguir de la misma manera y ver si está bien el diseño o simplemente hay que cambiarlo.

Una vez que se tiene la certeza de que el sistema funciona correctamente a nivel laboratorio, se procede a hacer las pruebas en campo, como el sistema que se está analizando es un sistema de comunicación óptico en espacio libre, esto obliga a hacer una investigación de campo, en donde se verifican las características del medio en donde estarán ubicados el transmisor y el receptor, estas características, pueden ser: posición geográfica, elevación, temperatura, viento, entre otros. Al igual que en el desarrollo experimental en esta etapa es posible que aparezcan nuevos requerimientos para su buen desempeño o problemas en el

caso extremo durante las pruebas de campo y por eso se regresa a solucionar o ajustar parámetros en la etapa del diseño.

En el diagrama de la Figura 12 se ejemplifican las diferentes etapas de todas las actividades que se realizaron.

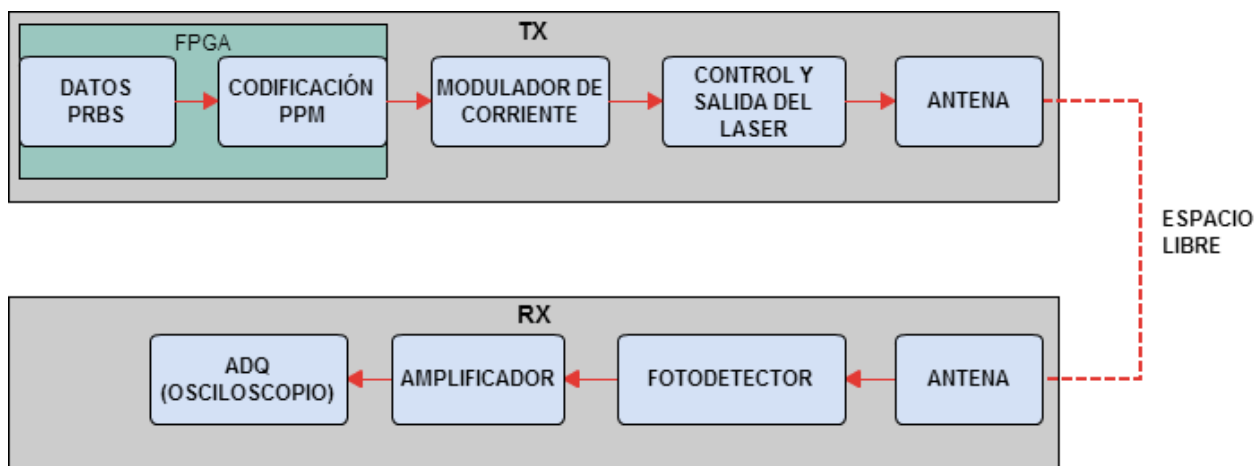


Figura 12. Diagrama a bloques y breve descripción del desarrollo de actividades durante la residencia profesional.

## 4.2. Investigación

En el bloque de la investigación teórica, se hicieron consultas en páginas de internet, artículos publicados sobre el tema, biblioteca del CICESE entre otras. Gracias a esta literatura y el apoyo de investigadores se pudo consolidar la base para comenzar el desarrollo de las actividades, un punto importante en los proyectos de investigación en Física Aplicada, es que la parte teórica y la práctica se van ejecutando y retroalimentando al mismo tiempo, dado que cada fundamento teórico es puesto en práctica y evaluado para verificar su validez.

Anteriormente hicimos referencia a la comunicación óptica por el espacio libre, describiendo sus aplicaciones, y factores limitantes, aquí definiremos un sistema completo de comunicación óptica por el espacio libre al cual se quiere llegar para poder llevar a cabo el desarrollo experimental.



**Figura 13. Sistema de comunicación óptica a desarrollar.**

En la Figura 13 se puede ver un diagrama a bloques del sistema de comunicación que se tiene contemplado realizar, separados en dos grupos, el transmisor TX, y el receptor RX. Cada uno cuenta con elementos dispensables para su funcionamiento, posteriormente se irán definiendo cada uno de estos elementos para su entendimiento y así poder hacer una caracterización de estos.

El primer bloque es el encargado de la generación y codificación de datos PRBS a través de un FPGA y por ello empezamos definiendo los FPGAs a continuación.

En el CICESE se cuenta con tres FPGAs para fines de Investigación, los tres de la misma marca (Xilinx), pero con distintas características, estos son:

- Spartan 6.
- Virtex 6.
- Virtex 7.

En la Figura 14 se muestra el encapsulado de cada uno de los FPGA descritos en el apartado anterior.



**Figura 14. Versiones de los FPGAs de la marca Xilinx que se tienen en el CICESE.**

La siguiente tabla comparativa describe algunas de las características más importantes entre ellos:

**Tabla 1. FPGAs disponibles en el laboratorio de Comunicaciones Photonicas del CICESE.**

Dispositivo	Celdas Lógicas <sup>2</sup>	Compuertas	Arreglo CLB			Block RAM Blocks	
			Ranuras <sup>3</sup>	Flip Flops	Max. Dist. RAM.	18 Kb <sup>4</sup>	Max. Kb
Spartan 6 XC6SLX9	9,152	423,703	1,430	11,440	90	32	576
Spartan 6 XC6SLX45T	43,661	2,021K	6,822	54,576	401	116	2,088
Virtex 6 XC6VLX240T	241,152	12,122K	37,680	301,440	3,650	832	14,976
Virtex 7 XC7VX485T	485,760	24,244K	75,900	607,200	8,175	2,060	37,080

En la Tabla 1 se describen dos FPGA Spartan 6 debido a que Xilinx provee una familia de FPGAs llamada “Spartan-6 Family” donde cada uno de los integrantes tiene distinto nivel de circuitería lógica, lo mismo sucede con las demás versiones de FPGAs; dos de la familia Spartan-6 se tiene disponibles XC6SLX9 y XC6SLX45T.

<sup>2</sup> La capacidad de celdas lógicas de los FPGAs refleja el aumento de la capacidad de celdas lógicas ofrecidas por la nueva arquitectura de 6 Entradas LUT.

<sup>3</sup> Cada rejilla de FPGA contiene cuatro LUTs y ocho flip-flops.

<sup>4</sup> Los bloques de RAMs son fundamentalmente de 18Kb de tamaño.



Hay que destacar que los FPGAs, vienen integrados en sistemas embebidos que ofrecen la posibilidad de probar sus diferentes funciones, llamadas tarjetas de evaluación [3] y es de esta manera que se tienen en el CICESE.

- Spartan-6 LX9 MicroBoard.
- Spartan-6 FPGA SP605 Evaluation Kit.
- Virtex-6 FPGA ML605 Evaluation Kit.
- Virtex-7 FPGA VC707 Evaluation Kit.

#### 4.2.1. WLD3343 Controlador de Propósito General para Diodos Laser

El WLD3343 es un circuito analógico usado para aplicaciones con limitaciones de espacio del diodo laser. El WLD3343 mantiene una corriente constante y precisa para el diodo laser (Constant Current Mode) o una corriente estable para el fotodiodo (Constant Power Mode) a fin de proteger y aislar el diodo laser o fotodiodo, en la Figura 15 se ilustra este controlador.[4]



Figura 15. WLD3343 Controlador de propósito general para diodo laser.

#### 4.2.2. Diodo Laser (980nm Pump Laser Module NORTEL LC92A80-20)

Como se aprecia en la Figura 16 este tipo de láser, necesita de una base tipo mariposa para su control.

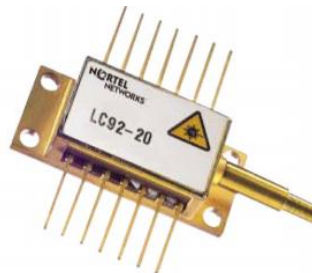


Figura 16. Laser NORTEL de 980 nm a 200mW.

### 4.2.3. LM14S2 Base Para Diodos Laser Butterfly

El LMS14S2 es una base para diodos laser tipo Butterfly (Mariposa) especialmente diseñado para diodos laser butterfly de 14 pines que tiene integrados refrigeradores eléctricos térmicos y sensores termistor. La función de este es la de mantener un control de temperatura para la estabilidad de longitud de onda y sintonización de temperatura del diodo laser así como se muestra en la Figura 17. [5]



Figura 17. Base para Diodos Laser tipo Butterfly.

### 4.2.4. Antena transmisora (THORLABS F810SMA-1550 Air-Spaced Colimators)

Están diseñados para colimar la propagación del haz del láser proveniente de la salida de una fibra óptica. Este colimador tiene a su entrada un SMA (SubMiniature version A) de tipo macho para la entrada de la fibra así como se muestra en la Figura 18 [6].



Figura 18. Colimador con entrada SMA de la marca THORLABS.

#### 4.2.5. Antena receptora (MEADE Telescope LX200-ACF)

MEADE LX200 es una familia de telescopios comerciales producidos por MEADE Instrumentes. El telescopio que se encuentra disponible en el laboratorio de Comunicaciones Photonicas del CICESE es el LX200-ACF de 14 pulgadas representado en la Figura 19 [7].



Figura 19. Telescopio LX200 de MEADE.

#### 4.2.6. Fotodetector (Hamamatsu APD Module C4777-01)

Los módulos APD son Fotodetectores de alta sensibilidad que integran un APD (Avalanche PhotoDiode), un circuito de control de temperatura (bias circuit) y convertidor de corriente a voltaje. La operación de un módulo APD es fácil ya que funciona con sólo suministrar una tensión continua de una fuente externa. Las aplicaciones que se le dan al Módulo C4777-01 son las de medición de fluorescencia, contador de partículas, sensor/monitor [8]. En la Figura 20 se ilustra el APD Hamamatsu.



Figura 20. Módulo APD Hamamatsu C4777-01.

#### 4.2.7. Amplificador (FEMTO DHPV4-100)

En la Figura 21 se muestra el amplificador de alta frecuencia, ganancia variable, ancho de banda de 100 o 200 MHz independiente de la ganancia que esté configurada, rango dinámico de entrada de 10 micro volt a 100 mili volt, utilizado para amplificar la señales de bajo voltaje [9].



Figura 21. Amplificador FEMTO DHPV4-100.

#### 4.2.8. DAQ (Adquisición de Datos)

Para la adquisición de datos se usó un osciloscopio, véase Figura 22, en donde su función consiste en tomar muestras de la señal analógica que está en su entrada, para luego generar datos que puedan ser procesadas por el mismo osciloscopio y al mismo tiempo poder hacer un análisis [10].

Agilent InfiniiVision DSO-X 3034A.

- 4 canales de almacenamiento digital
- Ancho de banda de 350 MHz
- Frecuencia de muestreo entrelazado de 4GSa/s y no entrelazado de 2GSa/s.



Figura 22. Osciloscopio digital Agilent InfiniiVision DSO-X 3034A.

## 4.2.9. Otros componentes

### 4.2.9.1. Conectores SMA

Se llama (*SubMiniature version A*) a un tipo de conector roscado para cable coaxial utilizado en microondas, útil hasta una frecuencia de 33 GHz. En la Figura 23 se ejemplifica este tipo de conector [11].



Figura 23. Conector SMA hembra a la izquierda, Conector SMA macho a la derecha.

### 4.2.9.2. Conectores DB9

El conector DB9 (originalmente DE-9) es un conector analógico de 9 clavijas de la familia de conectores *D-Subminiature* (D-Sub o Sub-D), véase Figura 24. El conector DB9 se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232 (RS-232C) [12].



Figura 24. Conector DB9.

### 4.2.9.3. Acoplador FC/PC FC/APC

EL conector FC mostrado en la Figura 25 es un conector de fibra óptica con un cuerpo roscado, que fue diseñado para su uso en ambientes de alta vibración [13].



Figura 25. Conector FC/PC.

#### 4.2.9.4. DIP Switch

En la Figura 26 se muestra un DIP (Dual In-line Package) Switch de 8 posiciones.

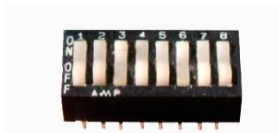
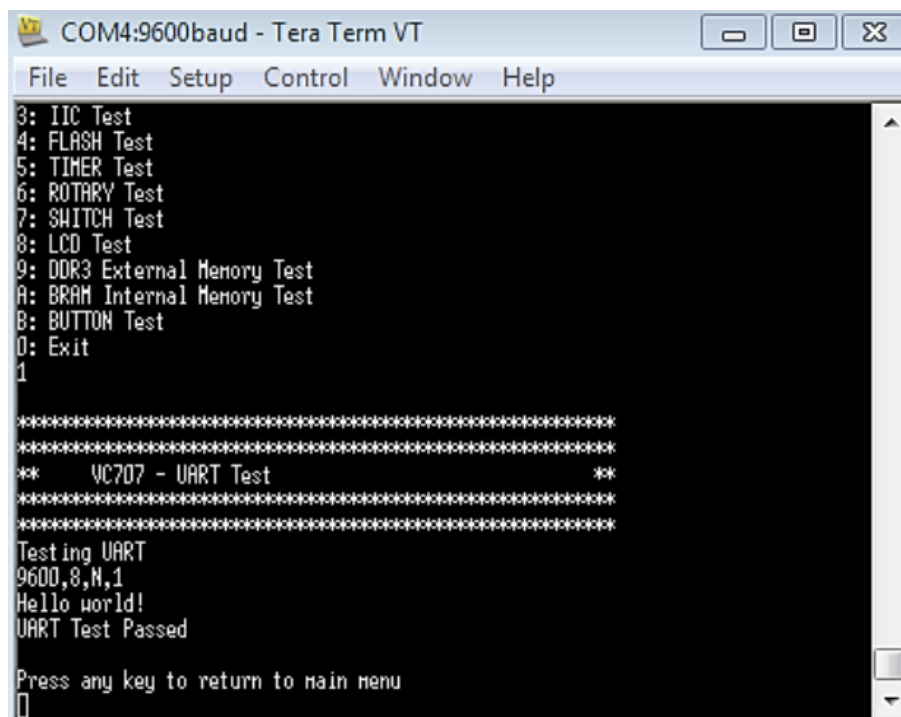


Figura 26. DIP Switch tipo Tecla 8 Posiciones.<sup>5</sup>

En cuanto a la investigación que se hizo para el diseño de los programas, esta se explica a detalle en la etapa de Diseño y Programación.

### 4.3. Caracterización de los FPGAs y Componentes Ópticos

Para poder caracterizar los FPGA, fue necesario interactuar con las herramientas que proporcionan las tarjetas de evaluación, y para eso se programaron los test de evaluación que la misma marca Xilinx proporciona, y con ellos el típico “Hola Mundo” en los sistemas digitales así como se muestra en la Figura 27.



```
COM4:9600baud - Tera Term VT
File Edit Setup Control Window Help
3: IIC Test
4: FLASH Test
5: TIMER Test
6: ROTARY Test
7: SWITCH Test
8: LCD Test
9: DDR3 External Memory Test
A: BRAM Internal Memory Test
B: BUTTON Test
0: Exit
1
*****
** VC707 - UART Test **
*****
Testing UART
9600,8,N,1
Hello world!
UART Test Passed
Press any key to return to main menu
█
```

Figura 27. Software Tera Term para testear FPGAs.

<sup>5</sup> Imagen tomada de [https://es.wikipedia.org/wiki/Interruptor\\_DIP](https://es.wikipedia.org/wiki/Interruptor_DIP)

### 4.3.1. Programación del FPGA

El software que provee Xilinx para la programación de sus tarjetas es presentado con el nombre de Xilinx ISE Design Suite y Xilinx Vivado Design Suite. Principalmente se usó ISE Design Suite durante el diseño y programación de datos PRBS y Codificación PPM de los FPGAs Spartan 6 y Virtex 6, sin embargo también se usó VIVADO para la tarjeta de evaluación “Virtex-7 FPGA VC707”. Para más explicación del Software ISE Design Suite véanse los anexos. Los primeros diseños y programas trataron de leer entradas con DIP Switchs ya integrados en las tarjetas de evaluación del FPGA SP605 y de utilizar los LEDs (Light-Emitting Diode: diodo emisor de luz) y salidas SMA del mismo; esto con la finalidad de familiarizarse con el entorno de desarrollo y uso de FPGAs.

### 4.3.2. Caracterización de componentes eléctricos y ópticos

Un elemento fundamental que se tiene que caracterizar y configurar es el WLD3343 Controlador para el diodo laser, el fabricante ofrece dos formas de configurar el controlador, (corriente constante, o voltaje constante). Esta configuración depende de lo que se quiera controlar, y para esta aplicación los queremos utilizar para controlar un diodo laser, entonces esto nos lleva a revisar las características del diodo laser a emplear.

A fin de asegurar las características de potencia y longitud de onda altamente estables, Nortel recomienda que el dispositivo sea operado aplicando una corriente de 40mA pico a pico a 2 MHz [Santos y Quintero, 2015]. Entonces a partir de estas restricciones se configura el driver WLD3343 de la manera que se muestra en la Figura 28.

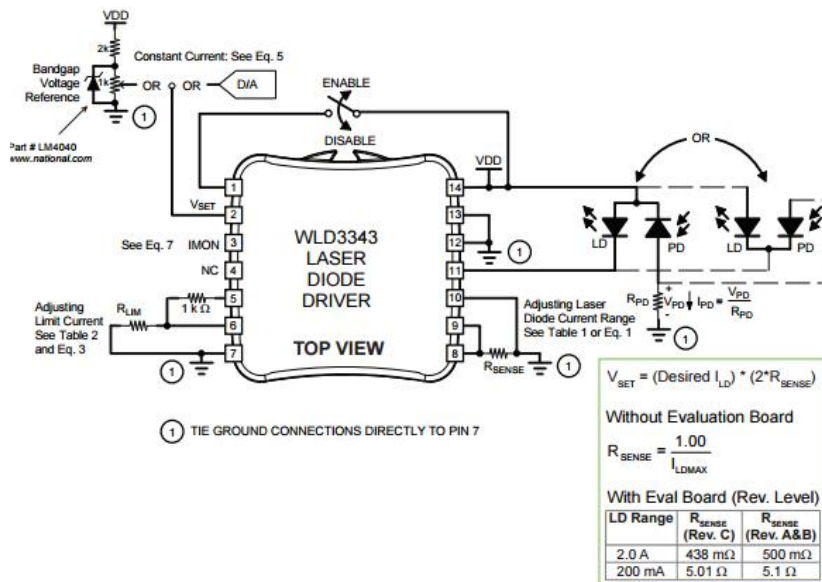


Figura 28. Configuración “Corriente Constante” del controlador de corriente DWL3343.

## 4.4. Diseño y Programación

Una vez que ya se tenía el conocimiento de cómo diseñar y programar un FPGA, se dio paso al diseño y programación de las etapas de generación de datos PRBS y Codificación PPM.

Durante la investigación previa al diseño fue necesario implementar un Generador de pulsos cuadrados, esto debido a que en los sistemas digitales es fundamental disponer de una señal de reloj cuadrada y no como la señal del reloj que proporciona un cristal comercial. Esto es fundamental para la buena sincronización de los sistemas digitales.

### 4.4.1. Generador de pulsos cuadrados (Divisor de frecuencia)

Para la generación de pulsos cuadrados en un FPGA basta con hacer un divisor de frecuencia para el reloj con el que trabaja el FPGA, pudiendo ser este en el *FPGA SP605 Evaluation Kit de Xilinx* el cual viene con un socket de usuario y así poder utilizar CLK's que más convenga. Para el ejemplo utilizaremos un Reloj de 100MHz.

El divisor de frecuencia es un componente simple, cuyo objetivo es reducir la frecuencia de entrada. Este se implementa con la ayuda de un factor de escalamiento y un contador.

Primeramente el factor de escalamiento es la relación entre la frecuencia de entrada y la frecuencia de salida deseada:

$$Escala = \frac{f_{entrada}}{f_{deseada}} \quad (1)$$

Asumiendo que tenemos una frecuencia de 100 MHz y deseamos una salida de 1KHz, tenemos que:

$$Escala = \frac{100MHz}{1KHz} = 100000$$

Por lo tanto el contador para el divisor de frecuencia tiene como función generar la señal de salida de 1KHz cada 100000 ciclos del reloj principal de 100MHz.

Como se muestra en el diagrama de la Figura 29 el bloque del divisor de frecuencia tiene una única entrada para reloj de 100MHz y una salida donde se obtiene la nueva señal de reloj con la frecuencia deseada [14].



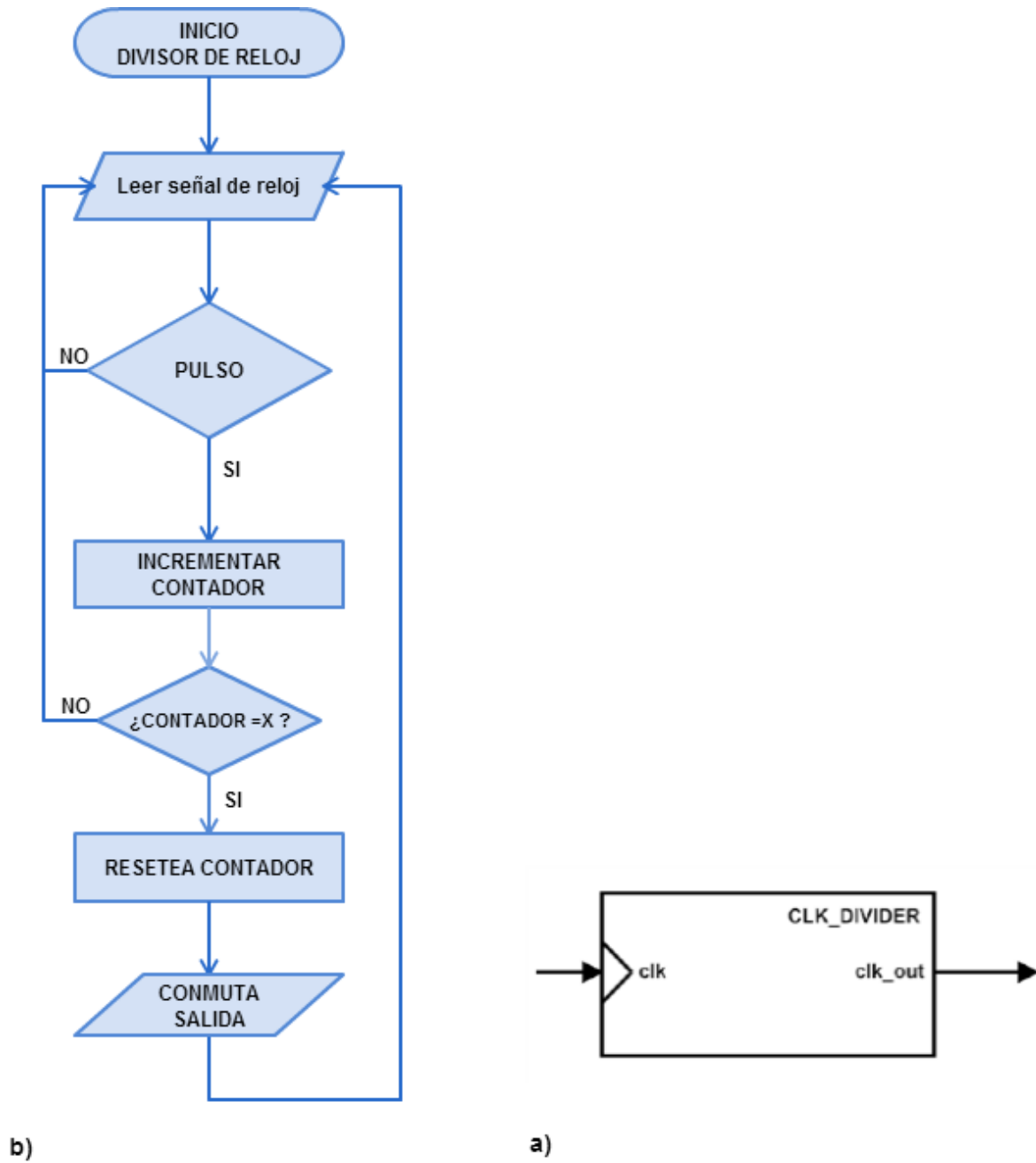
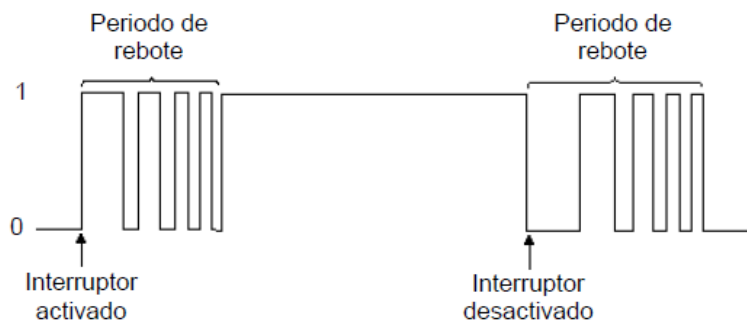


Figura 29. a) Diagrama de flujo del divisor de frecuencia, b) Bloque de entrada y salida del divisor de frecuencia.

#### 4.4.2. Circuito Antirebote

Este bloque no se tenía contemplado en el diseño, pero fue necesario para poder implementar switches y push-buttons de control.

Todos los interruptores mecánicos como los push-buttons y los switches están compuestos internamente por contactos metálicos. Cuando se presiona un interruptor son estos contactos los que cierran el circuito para dejar fluir la corriente. Aunque sean muy pequeños, estos contactos no se liberan de las leyes físicas. Al ser metálicos tienen masa y al ser delgados tienen elasticidad. Son diseñados para abrir y cerrar con rapidez por lo que existe poca resistencia y amortiguación en su movimiento. Por esto al presionar el interruptor, los dos contactos chocan entre si y rebotan entre ellos varias veces antes de llegar a la posición de reposo. Figura 30.

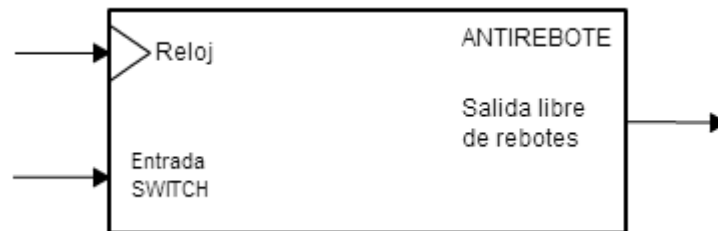


**Figura 30. Señal de interruptor al rebotar sus contactos internos.**

Este comportamiento de rebote genera problemas en los circuitos digitales que trabajan a altas velocidades, ya que puede capturar estos rebotes como valores que considera válidos. Por ejemplo en un contador que se incrementa cada vez que se presiona un interruptor, el valor se incrementa no únicamente cada que se presiona el interruptor sino también cada vez que los contactos rebotan internamente cada vez que los contactos rebotan internamente proporcionando un valor incorrecto del conteo.

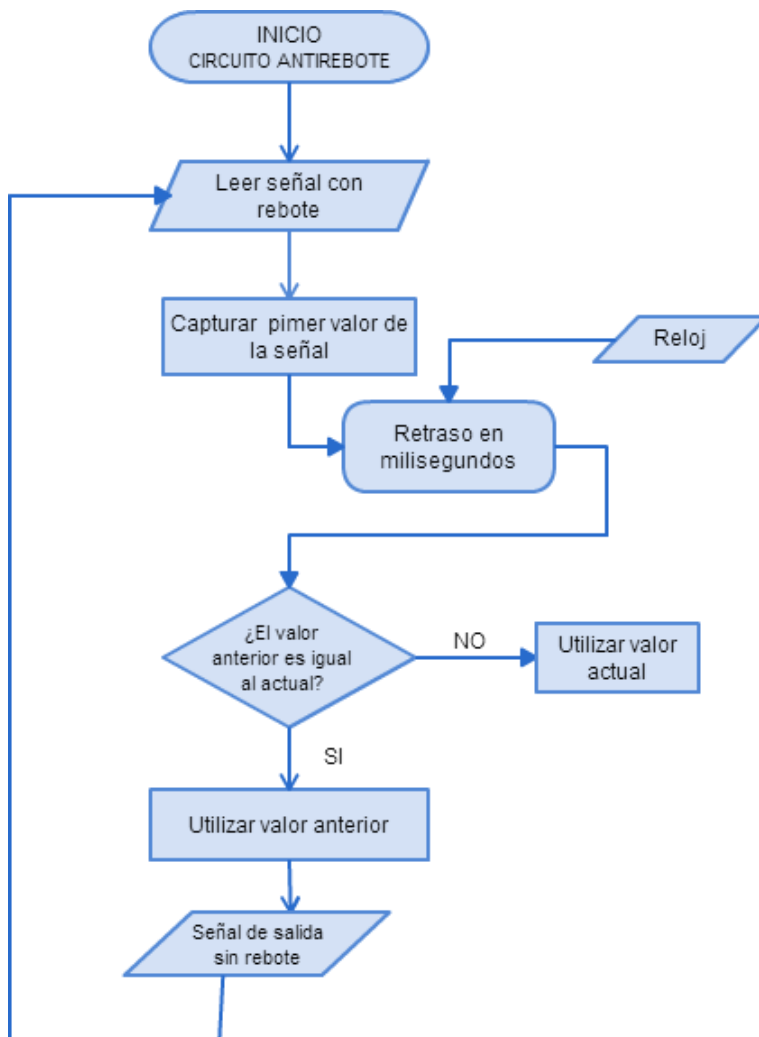
La solución es utilizar un circuito anti-rebote que mantenga la señal del primer flanco durante el tiempo suficiente para ignorar los rebotes, esto se podría hacer vía hardware utilizando un circuito RC, pero al trabajar con un FPGA es factible implementar un sub-bloque anti-rebote dentro del mismo que funcione para cualquier interruptor mecánico que se desee conectar al FPGA. Básicamente es un sub-bloque que reciba a la entrada el pulso del interruptor y asigne a la salida el valor obtenido en el primer flanco de la entrada durante un periodo de tiempo lo suficientemente largo para evitar los rebotes concurrentes, pero lo suficientemente corto para no llegar a ignorar las conmutaciones reales. Como se muestra en la Figura 31 este sub-bloque solo necesita dos entradas, una es la entrada del reloj de sincronización y la otra es la señal

del interruptor al que se le quieren eliminar los rebotes. En la única salida del sub-bloque se obtiene la misma señal que la del interruptor pero libre de rebotes [Hernández, 2008].



**Figura 31. Bloque de entrada y salida del circuito anti-rebote.**

Como se describe en el diagrama de flujo de la Figura 32 el funcionamiento de este programa es sencillo.



**Figura 32. Diagrama de flujo del programa para el circuito anti-rebote.**

#### 4.4.3. Generador De Secuencia Binaria Pseudoaleatoria (PRBS)

Como ya se planteó en el Marco Teórico los LFSR (*Linear Feedback Shift Register*) son ampliamente utilizados para la generación de secuencias pseudoaleatorias de datos binarios, y por ello se implementa en el FPGA un generador de 4 bits así como se describe en la Figura 33:

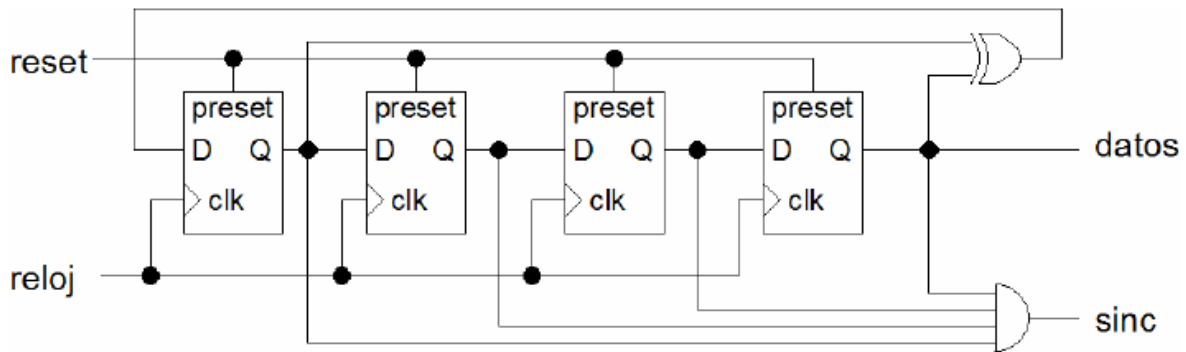


Figura 33. Diagrama LFSR para la generación de secuencia binaria pseudoaleatoria de 4 bits.

La entrada “reloj” Es la que nos proporciona el bloque de Generador de pulsos cuadrados, el reset/preset viene de un pulsador externo y a la salida están los datos, aparte de esto hay una señal adicional que nos sirve para sincronización tal como se muestra en la Figura 34.

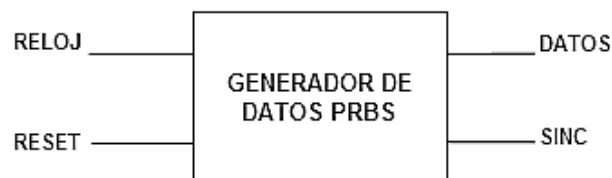
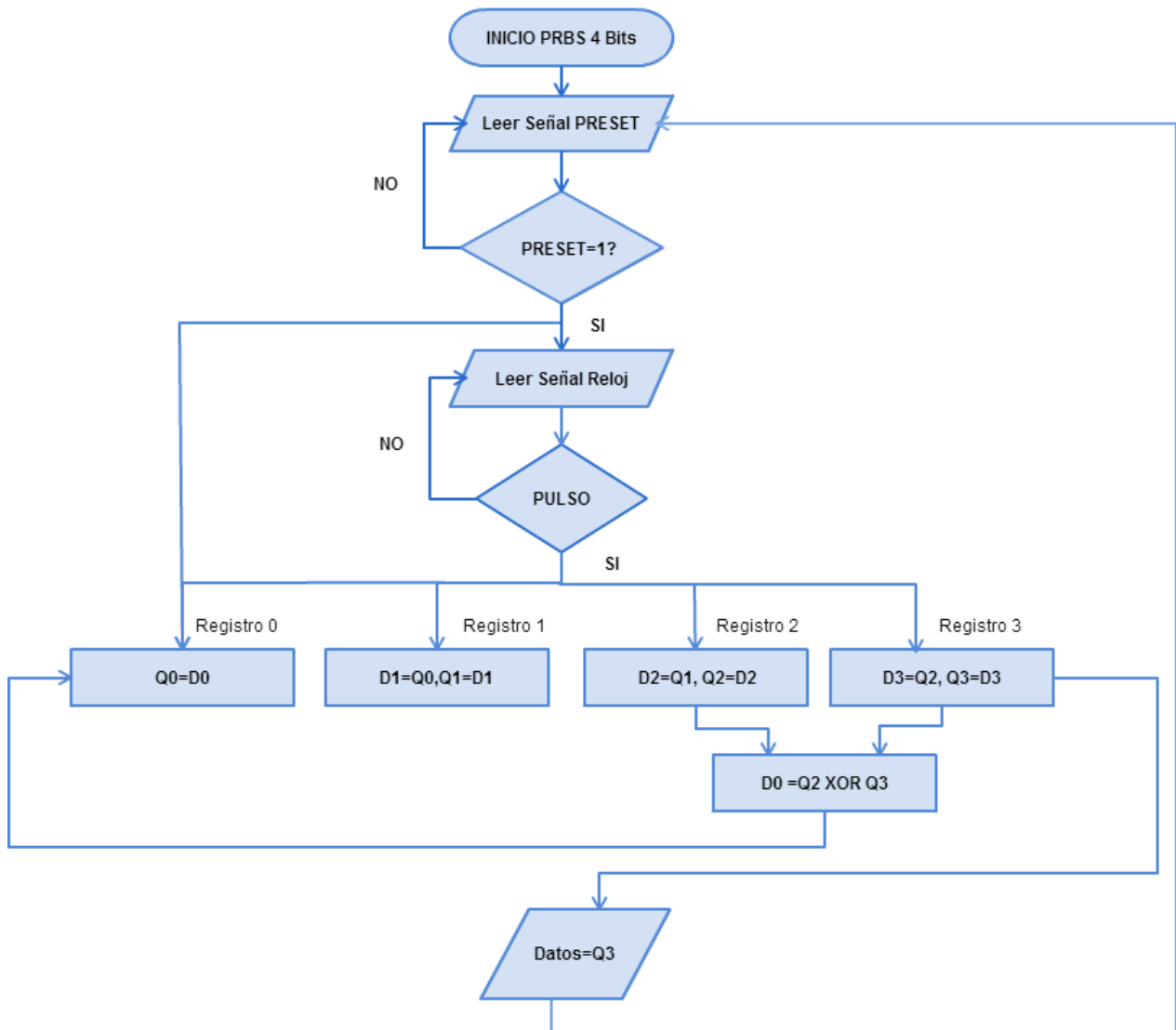


Figura 34. Bloque de entrada y salida del generador de datos PRBS.

En el diagrama de la Figura 35 se describe el funcionamiento del generador PRBS de 4 bits.



**Figura 35. Diagrama de flujo del programa del circuito generador de secuencias binarias Pseudoaleatorias, PRBS.**

Los LFSR (Linear Feedback Shift Register) son métodos para generar secuencias de números pseudoaleatorios utilizando bits que recorren un arreglo de celdas y que relacionan la salida con la entrada utilizando la operación XOR, sin embargo para saber entre que bits se hace esta relación es necesario consultar la Tabla de LFSR de Galois, esta tabla figura en los anexos.

Para que el generador de datos PRBS, sea variable en frecuencia, y en tamaño de palabra, se tuvieron que implementar dos bloques adicionales de código que se describen en los Anexos.

#### 4.4.4. Codificador 8-PPM de datos PRBS

##### Pulse-Position Modulation (PPM)

El PPM óptico se adapta bien a técnicas de modulación de láser existentes ya que requiere potencia media baja, alcanza razonablemente altas eficiencias de información y es resistente a la radiación de fondo.

Una fuente de  $k$ -bits  $U = (U_1, \dots, U_k) \in \{0,1\}^k$  es modulada con  $M$ -ary PPM,  $M=2^k$ , para obtener una señal  $X = (0,0,\dots, 0,1,\dots,0) \in \{0,1\}^M$ , que contiene un solo uno en la posición indicada por la representación binaria de  $U$ .

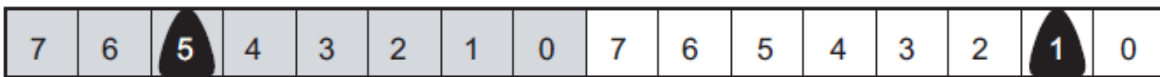


Figura 36. 8 PPM utiliza un símbolo de 8 ranuras (Slots) por cada tres bits. Se muestra la modulación del mensaje 101001. El orden en que los slots están etiquetadas no es consecucional. Aquí, la etiqueta 7 es asignado a la ranura más a la izquierda de un símbolo y la etiqueta 0 está asignada a la ranura más a la derecha.

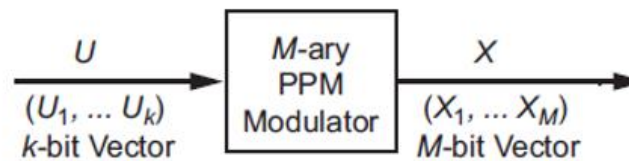


Figura 37. M-ary PPM Modulador.

Para la codificación PPM fue necesario hacer un análisis de cómo se generan los datos PRBS y de esta manera poder codificarlos en un formato PPM, en la Figura 38 se puede ver el bloque de Codificación en donde se tiene tres entradas, reset, datos PRBS y reloj, además de salida donde se obtienen los datos Codificados en formato PPM [Hemmati, 2005].

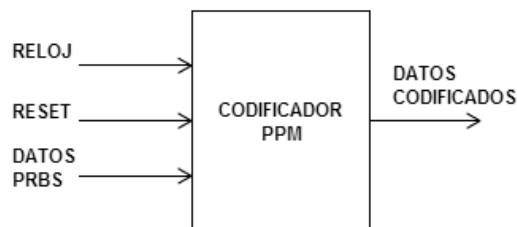
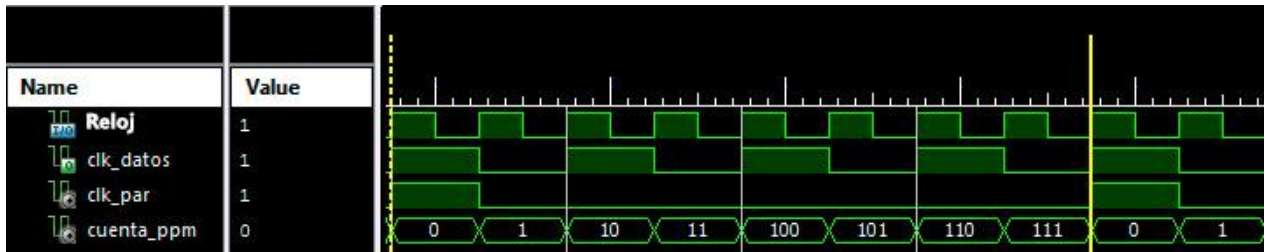


Figura 38. Bloque del codificador PPM.

Durante el diseño del codificador PPM, se descubrió que la señal del reloj del generador de datos PRBS está fuertemente relacionado con el reloj del Codificador PPM esto es para una buena sincronización. En la Figura 39 se describe esta relación.



**Figura 39. Relación entre la señal del reloj del generador con el reloj del codificador.**

Podemos observar que hay un Reloj principal que puede ser de 100MHz, del cual se crean dos relojes secundarios uno para los datos *clk\_datos*, y otro para el PPM, *clk\_par*, cumpliéndose la condición de que cada ciclo del reloj ppm hay cuatro del reloj para los datos y que durante el ciclo del *clk\_par* hayan ocho divisiones, o lo que es lo mismo un conteo de 0 al 7; esto debido a que el bloque codificador ppm debe esperar a que le llegue una cadena de tres bits del generador de datos y convertirlos a su equivalente en código ppm, así pues si le llega tres bits "001" el codificador lanza un alto en el lugar que le corresponde dentro del contador ppm "001". Por esta razón nos damos cuenta de la reducción en la frecuencia de generación de datos PRBS, sin embargo esto no afecta en el proceso de comunicación óptica. Como se verá posteriormente.

En el diagrama de flujo de la Figura 40 se describe el comportamiento del sistema modulador PPM.

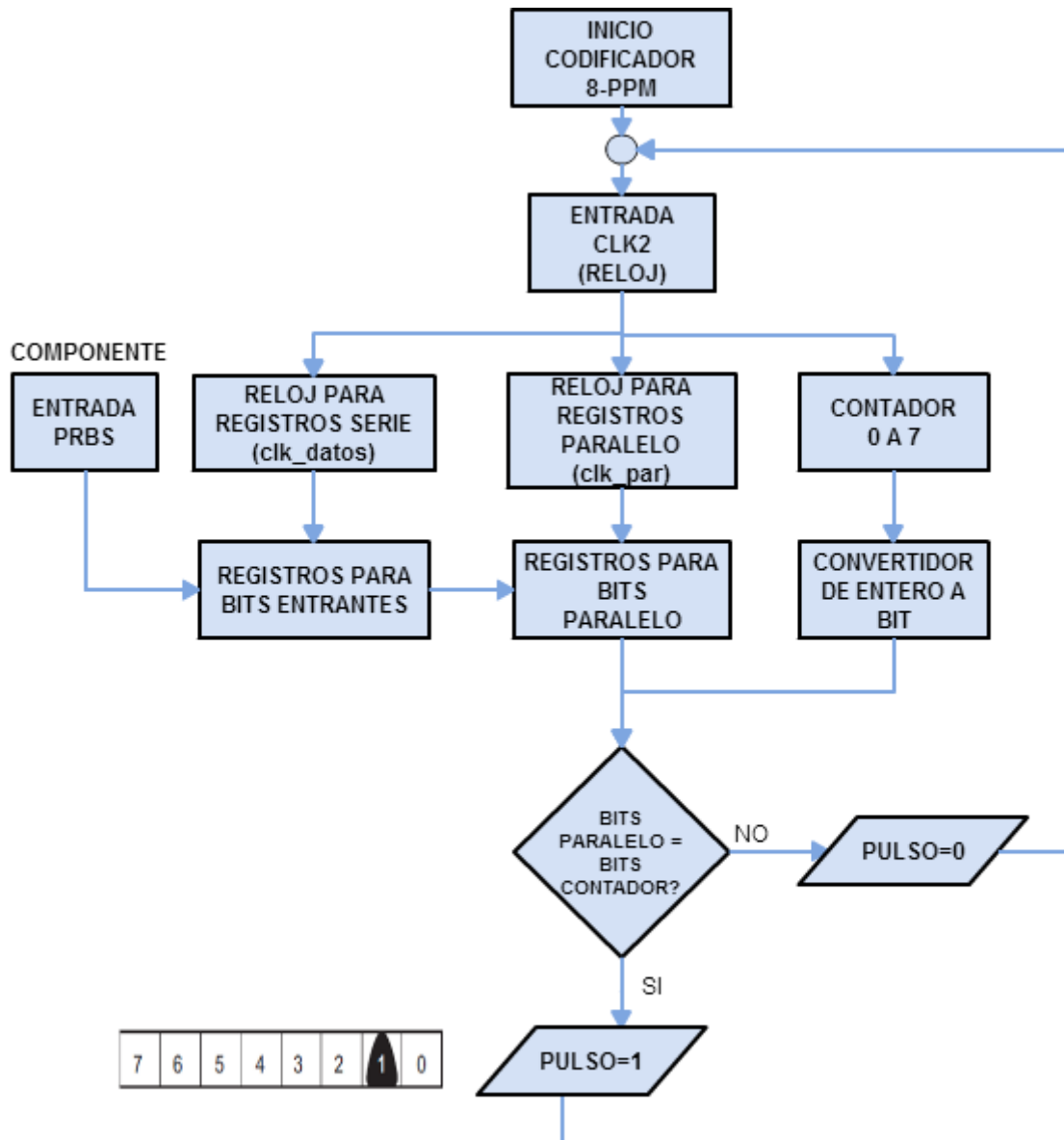
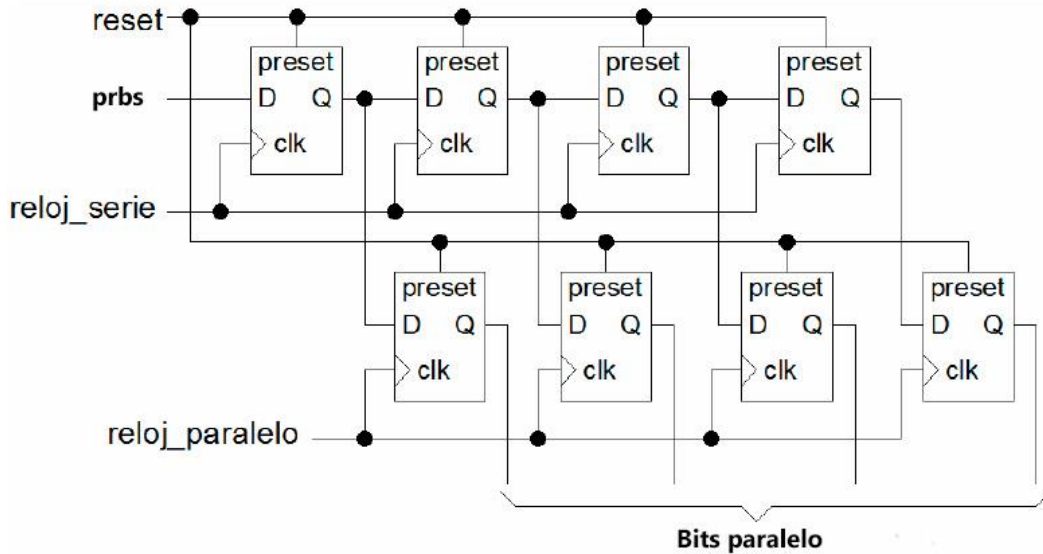


Figura 40. Diagrama de Flujo del Modulador 8-PPM.

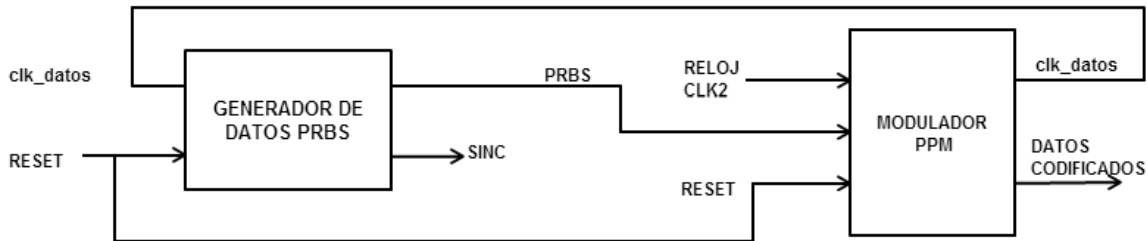
La parte de los registros se puede explicar mejor con la Figura 41.





**Figura 41. Registros para la captura de bits.**

Hasta aquí se diseñó y programó el FPGA Spartan-6 quedando como muestra el diagrama de bloques de la Figura 42.



**Figura 42. Diagrama a bloques del generador de datos PRBS y Modulador PPM.**

Para consultar los programas dirigirse a los Anexos.

## 4.5. Desarrollo Experimental

Esta etapa del desarrollo de actividades consta de todas las pruebas que se hicieron con el FPGA, el armado del transmisor, y el montaje experimental en el laboratorio del transmisor y receptor, para verificar su funcionamiento en un ambiente controlado, para posteriormente hacer las pruebas en campo.

Para mejor explicación del tema, se dividió en dos partes: El Transmisor y El Receptor.

### 4.5.1. Transmisor

De acuerdo a la Figura 13, la parte del transmisor está compuesta por el FPGA, el modulador de corriente, el controlador de corriente junto con el láser, y la antena transmisora, todas estas partes a excepción de la antena, quedaron armadas dentro de una caja de aluminio la cual se adaptó para que fuera lo más portable posible y así poder transportarla con facilidad.

Gracias a las características de tamaño que presenta el FPGA “Spartan-6 LX9 MicroBoard” se optó por usarlo. Véase la Figura 43.



Figura 43. FPGA Spartan-6 LX9 MicroBoard.<sup>6</sup>

Aunque no tiene el mismo número de compuertas y entradas/salida que las demás tarjetas, fue posible implementar el diseño del Generador PRBS, además de las siguientes salidas y entradas de control:

- Salida, Siwtch RESET.
- Salida para los datos PRBS.
- Push-Button, Selector de frecuencia para los datos PRBS.
- Switch para selector de tamaño de pseudoaleatoriedad.
- Salidas para leds indicadores de frecuencia.

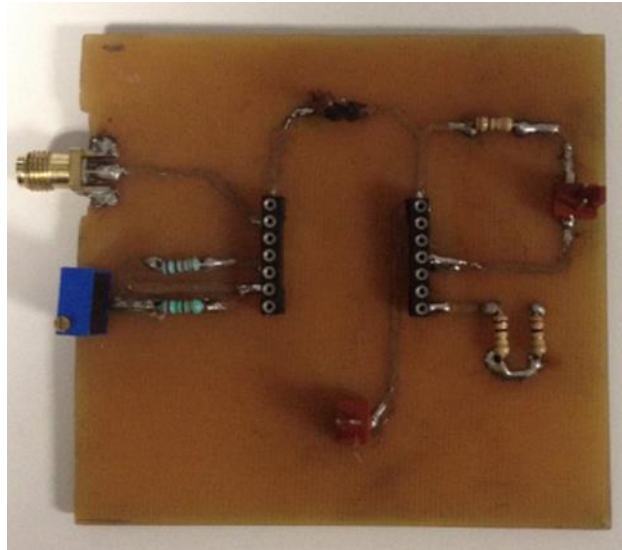
A continuación se muestran las imágenes (Figura 44) de los datos PRBS en el Osciloscopio que describen el funcionamiento de los datos PRBS de 4 bits y 8 bits respectivamente.

<sup>6</sup> Imagen tomada de <http://www.em.avnet.com/en-us/design/drc/Pages/Xilinx-Spartan-6-FPGA-LX9-MicroBoard.aspx>



**Figura 44. Datos PRBS de 4 Bits de resolución (Superior) y Datos PRBS de 8 Bits de resolución (Inferior verde).**

Una vez que se tenían los datos PRBS en el FPGA (variables en frecuencia y tamaño de bits) se prosiguió con el bloque del controlador de corriente, como ya se vio este controlador necesita ser configurado en un circuito. Esta parte del diseño, se hizo anteriormente en los trabajos de investigación del CICESE, a continuación se muestra el circuito resultante del driver.



**Figura 45. Circuito base para el controlador de corriente.**

Como puede observarse en el Figura 45 hay una entrada SMA en donde entra la señal proveniente para los datos y una salida para el siguiente bloque (Control de temperatura).

Esta salida que ya tiene un control de corriente va hacia la base para diodos laser butterfly, como la entrada para el controlador de temperatura es de tipo DB9 fue necesario hacer el conector quedando de la siguiente manera:

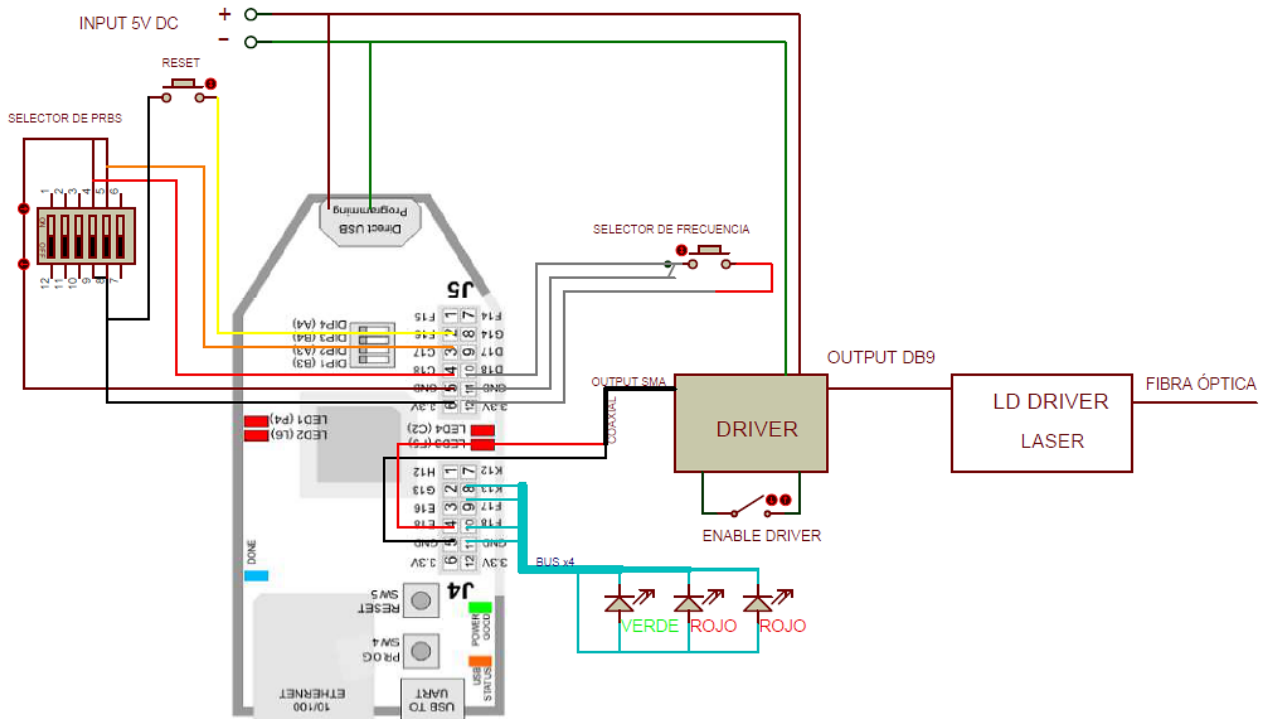
Donde un extremo viene de la salida del controlador de corriente y el otro a la entrada DB9 que tiene el controlador de temperatura. En esta misma base se inserta el diodo laser tal como se muestra en la Figura 46.



**Figura 46. Inserción del diodo laser a la base LM14S2.**

El diodo laser viene conectado a un metro de fibra desnuda mono modo y a la salida de la fibra tiene un conector FC macho.

Aunado a esto también fue necesario implementar en el soporte, un conector USB hembra para la alimentación del FPGA, alimentación y Switch ON-OFF para el controlador de corriente. DIP switch para selección del tamaño de bit de los datos, push-button para la selección de frecuencia, LEDs indicadores de la frecuencia seleccionada, conector FC/PC que une el extremo de fibra del diodo con el extremo de fibra que va a la siguiente etapa (Antena). Las conexiones eléctricas de estos elementos con el FPGA se muestran en la Figura 47.



**Figura 47. Diagrama eléctrico que describe las conexiones en el FPGA LX9 empleado para la transmisión de datos PRBS.**

Para completar la parte del transmisor, a la salida del conector FC/PC de la caja generadora se conecta la antena transmisora, se trata de un colimador, así como se muestra en la Figura 48 y este a su vez se coloca sobre una base para así lograr una estabilidad.



**Figura 48. Fibra conectada al colimador.**

La función del colimador es homogeneizar el haz del láser proveniente de la fibra óptica y proyectarlo idealmente de manera paralela.

### 4.5.2. Receptor

Siguiendo el diagrama del sistema que se desarrolló, los elementos que conforman el receptor son:

- Antena receptora
- Foto detector
- Amplificador
- ADQ (Osciloscopio)

La antena que se emplea para recibir las señales ópticas fue implementada con un telescopio, el LX200 MEADE, que al estar en línea de vista con la antena transmisora se puede recuperar una porción de la potencia que se está transmitiendo, tal y como se muestra en la Figura 49.



**Figura 49. Esquema que ejemplifica la función del telescopio como antena receptora.**

El foto detector se encarga de convertir las señales ópticas que capta el telescopio en señales eléctricas, el foto detector que se emplea es el módulo APD C4777-01 de la marca Hamamatsu, el módulo es sensible a las longitudes de onda de 800 nm y un límite mínimo de tensión de 0.2 pW r.m.s.

La señal detectada por el módulo es de muy baja potencia, y refleja una salida eléctrica de bajo voltaje, precisa entonces amplificar esta señal para poder hacer un análisis mejor y es por ello que se implementa un amplificador de voltaje, variable en ganancia.

Posteriormente la señal se envía hacia el ADQ que adquiere los datos muestreándolos; al tratarse de un osciloscopio como ADQ, este muestrea las señales de su entrada para luego procesar las muestras y desplegarla en pantalla y de este modo poder hacer un análisis de las señales obtenidas.

Hasta aquí hemos explicado todo el desarrollo experimental realizado, esto incluye las pruebas de los datos PRBS y Modulación PPM, así como el desarrollo experimental del sistema de comunicación óptica en espacio libre en el laboratorio, hasta este punto se verificó el correcto funcionamiento de los elementos que integran el sistema de comunicación.

## 4.6. Pruebas en Campo

Uno de los parámetros que define la calidad de un sistema de comunicación es la distancia entre los dos puntos que está comunicando, ya que mientras más grande sea esta distancia, mejor será el sistema de comunicación; está claro que existen muchas limitantes físicas, por ejemplo, la pérdida de potencia del láser al propagarse por el medio, la sensibilidad del fotoreceptor, el camino entre el transmisor y el receptor y esto incluye la curvatura de la tierra al tratarse de un sistema de comunicación que requiere línea de vista, etc.

Además de que al realizar las pruebas en campo se pueden determinar otros requerimientos que no se pueden detectar en el laboratorio.

En este apartado explicamos, el desarrollo de las actividades que se realizaron para hacer las pruebas en campo del sistema de comunicación.

### 4.6.1. Ubicación del transmisor y el receptor

Uno de los requisitos dentro del sistema es la línea de vista, en la Figura 50 se muestra la localización del transmisor (Tx) y el Receptor (Rx) y en la Figura 51 se muestra el perfil de elevación entre el punto Tx y Rx.

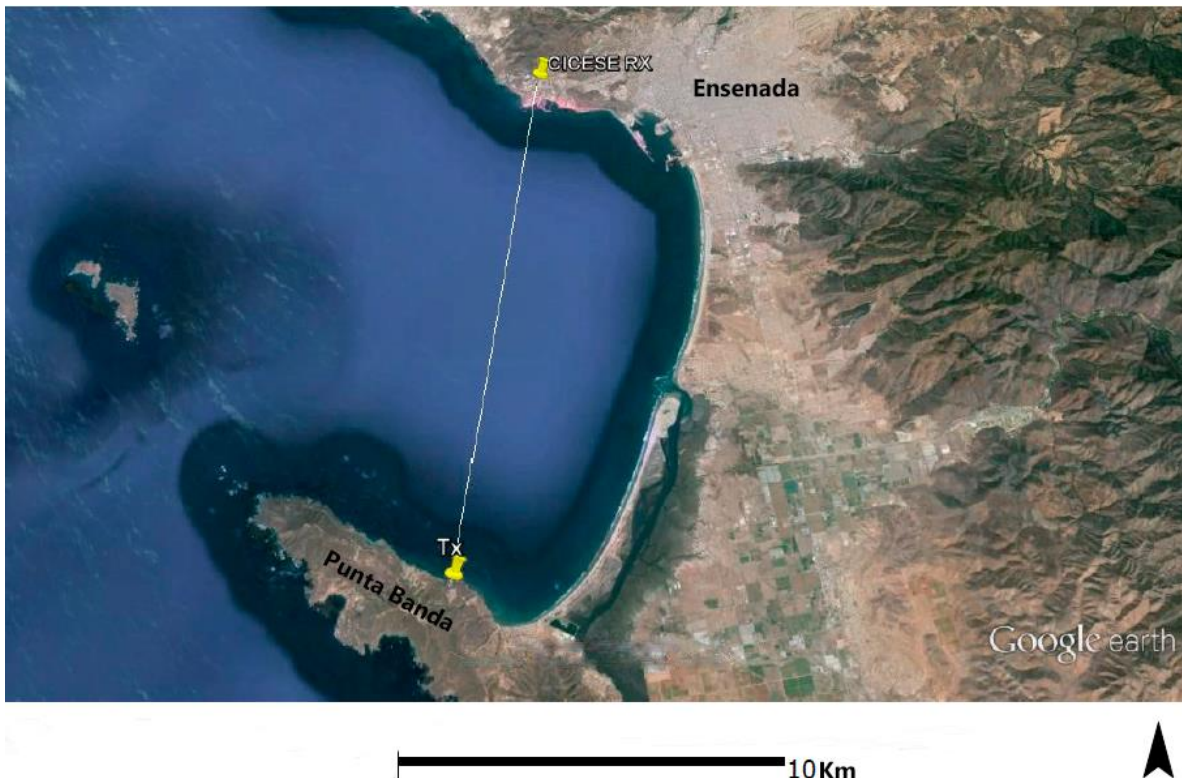


Figura 50. Localización del transmisor (TX) y receptor (RX) en el mapa de Ensenada.



**Figura 51. Perfil de elevación entre el transmisor y el receptor haciendo uso de herramientas de Google Earth.**

Para poder elegir estos dos puntos primero se tuvo que realizar una búsqueda de un punto en la zona de Punta Banda que tuviera línea vista hacia el CICESE y un buen lugar para instalar el Transmisor.

El receptor se instaló en el edificio de física de aplicada, en uno de los cubículos con vista hacia Punta Banda.



## CAPÍTULO 5. RESULTADOS

### 5.1. Resultados de la investigación teórica

La generación de una secuencia pseudoaleatoria de números binarios es muy útil en ciertos ambientes de test y desarrollo, por ejemplo si se quieren hacer pruebas en un canal de comunicación, la manera más eficiente y rápida de simular una fuente de datos es implementando un generador de datos pseudoaleatorios y así obtener resultados como si se tratase de una fuente de datos reales, como lo es el audio o video. La implementación de estos generadores se hace fácil con el uso de circuitos digitales llamados LFSR.

En cuanto a la modulación de datos, el tipo 8-PPM es la más utilizada para establecer enlaces ópticos en el espacio libre, ya que en este escenario suelen haber pocas interferencias por camino múltiple. Se puede demostrar que aun teniendo una fuerte deformación de la señal, la señal puede ser recibida, solo con la condición de que el máximo de la señal en cada periodo de símbolo se mantenga en su slot temporal original.

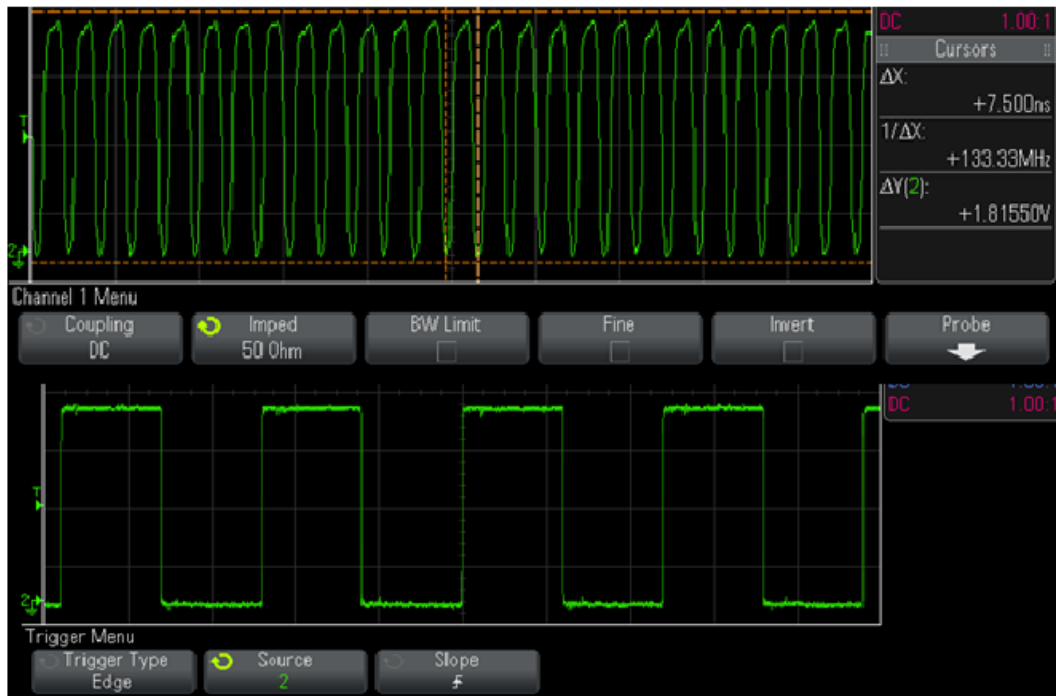
### 5.2. FPGA

Si bien es posible implementar nuestro sistema de generación y codificación en cualquiera de los FPGAs que se tenían disponibles, sin embargo su desempeño era diferente en cada uno, en la Tabla 2 se explican los parámetros que hacían diferente este desempeño además de características esenciales para la implementación y análisis.

**Tabla 2. Desempeño en frecuencia de los FPGAs disponibles en el CICESE al implementar el mismo diseño en cada uno de ellos.**

FPGA	Frecuencia máxima de generación de datos	Frecuencia máxima de codificación de datos	SMA	Dimensiones Ancho x largo x alto
<b>Spartan-6 LX9 MicroBoard</b>	100 MHz	50 MHz	no	4cmx10cmx3cm
<b>Spartan-6 FPGA SP605 Evaluation Kit</b>	133MHz	66 MHz	si	14cmx20cmx4cm
<b>Virtex-6 FPGA ML605 Evaluation Kit</b>	200 MHz	100MHz	si	15cmx25cmx4cm
<b>Virtex-7 FPGA VC707 Evaluation Kit</b>	400 MHz	200 MHz	si	15cmx25cmx4cm

### 5.2.1. Resultados del Generador de pulsos cuadrados que se implementó

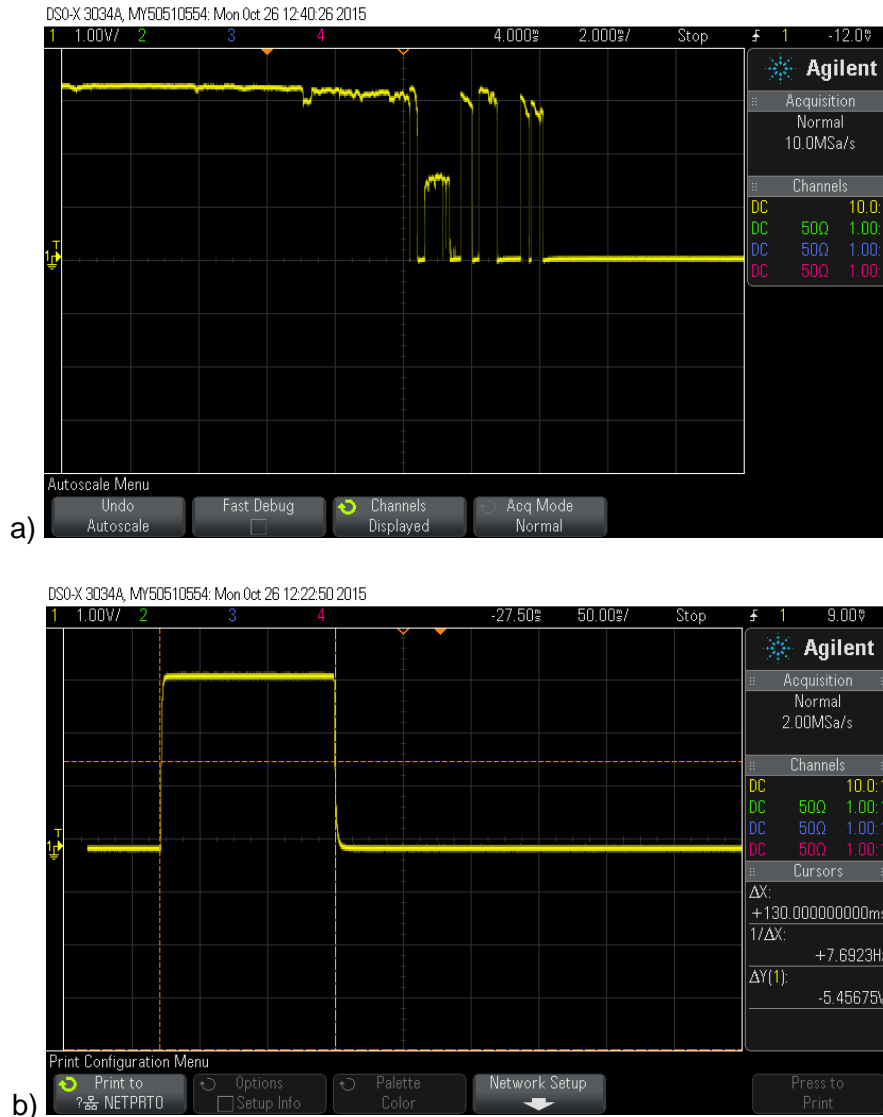


**Figura 52. Comparación entre señal del XTAL que presenta un FPGA (Superior) y Señal de reloj que devuelve un generador de pulsos cuadrados implementado en un FPGA (Inferior).**

El diseño de este generador da la posibilidad de obtener un generador de pulsos cuadrados con ciclo de trabajo del 50% dependiendo del reloj interno del FPGA que se esté utilizando (Figura 52), si por ejemplo se emplean el LX9 Microboard, se tendrá generador de 50MHz, 25 MHz, 12.5 Mhz ..., debido a que su reloj interno es de 100MHz, en cambio si se utiliza el SP605, se tiene la posibilidad de usar un reloj externo que queda a elección del usuario siempre y cuando este dentro de sus límites de operación.

### 5.2.2. Resultados del Circuito Anti-rebote implementado

En la Figura 53 a) se pueden ver los rebotes que produce el push-button antes de alcanzar un valor estable, con la ayuda del osciloscopio fue posible ver el tiempo que tiene que transcurrir para que se alcance esta estabilidad (4ms) y así poder implementar el “reloj” del circuito anti rebote.

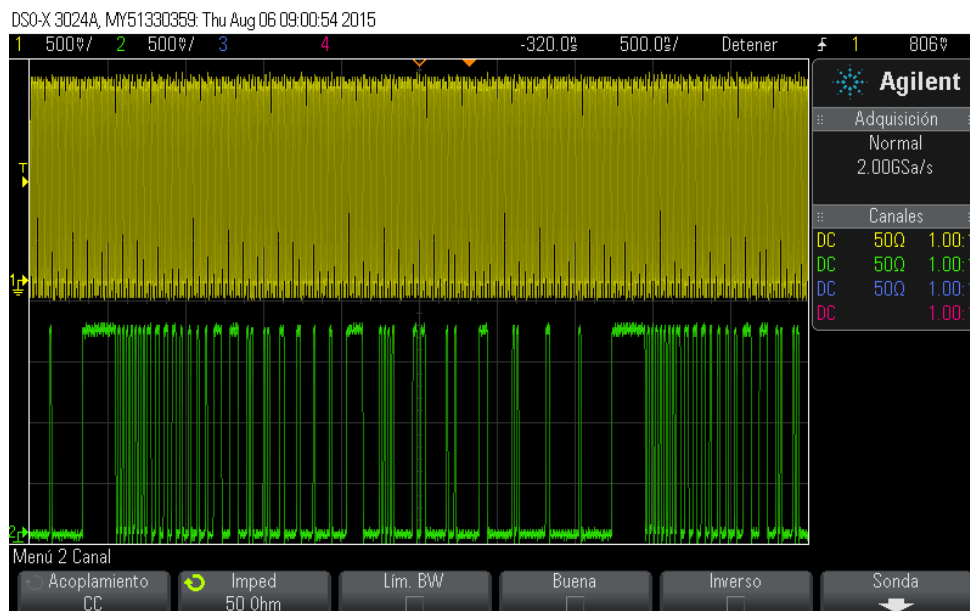


**Figura 53. a) comportamiento de Push-Button empleado antes del implementar el circuito anti rebote, b) comportamiento del mismo Push-Button después de implementar el circuito anti rebote.**

### 5.2.3. Generador de datos PRBS

El FPGA que se utilizó para generar los datos PRBS utilizados en el enlace fue el Spartan-6 LX9 MicroBoard por su tamaño y velocidad del reloj interno de 100MHz. Sin embargo también se hicieron pruebas con en el FPGA Spartan-6 SP605 para alcanzar velocidades en el rango de los 150 MHz, y también el Virtex-6 en el orden de los 200 MHz.

Los resultados obtenidos al implementar el generador de datos PRBS son los siguientes:



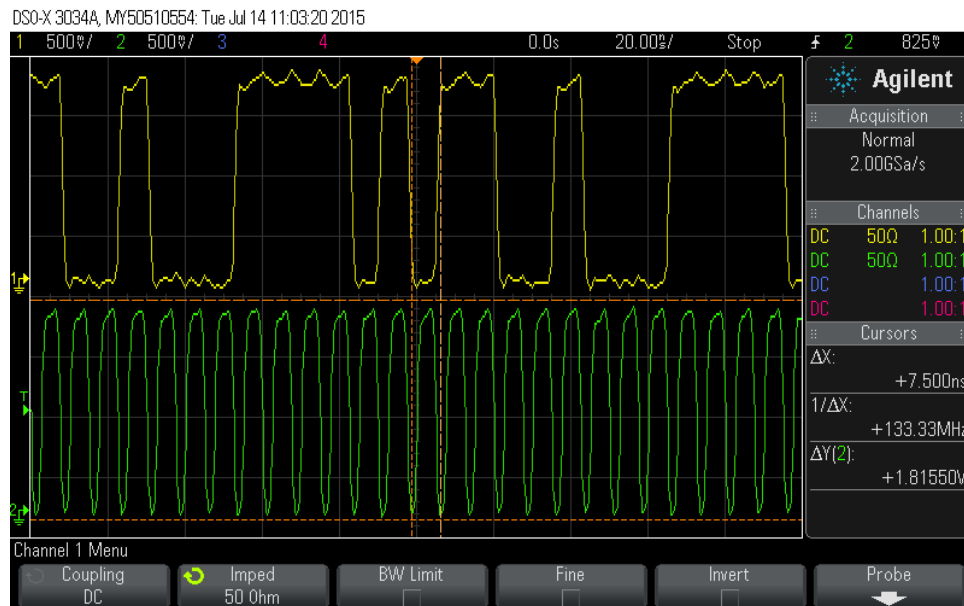
**Figura 54. Imagen tomada del osciloscopio que muestra los datos PRBS de 8 bits a una frecuencia de 75 MHz.**

En la imagen de la Figura 54 se observa el reloj interno del FPGA Microboard LX9 (señal Amarilla) y los datos obtenidos del generador implementado en el mismo FPGA (señal verde), tanto el reloj como el generador están a 75 MHz. Junto a esto también se expone en la Tabla 3 la cantidad de recursos utilizados por el FPGA para implementar el generador.

**Tabla 3. Recursos utilizados por el FPGA Spartan-6 Microboard para implementar el generador de PRBS 8 de bits.**

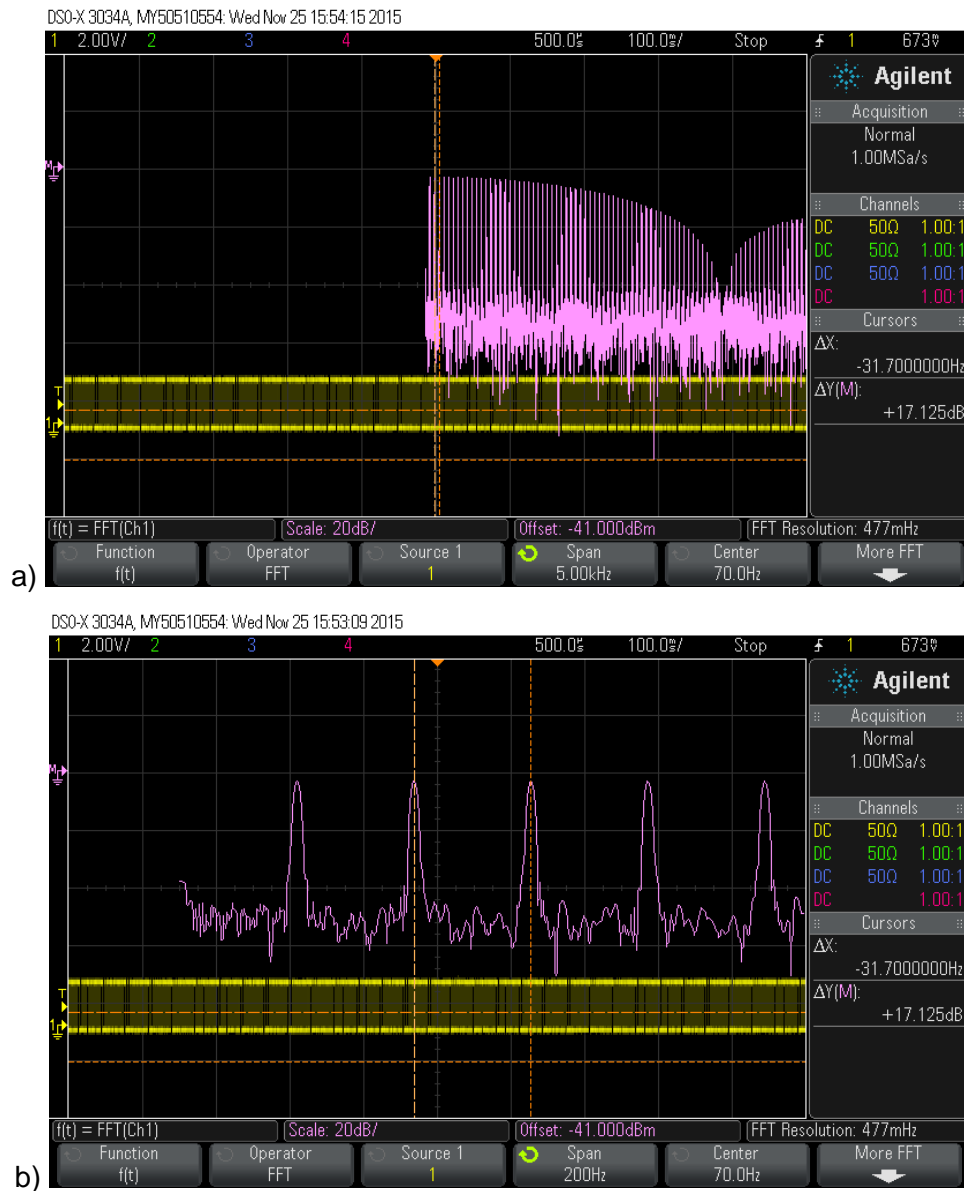
Utilización de partes lógicas	Usado	Disponible	Porcentaje Utilizado
Registros	8	11,440	1%
Flip Flops	8		
LUTs	6	5,720	1%
Usada como lógicas	3	5,720	1%
Pares LUT Flip Flop	6		
Flip Flop sin usar	2	6	33%
LUT sin usar	0	6	0%
Número de pares de LUT-FF usadas totalmente	4	6	66%
Número de registros perdidos para controlar restricciones	0	11,440	0%
Número de IOBs	5	200	2%
Número de LOCed IOBs	4	5	80%

En la Figura 55 se muestra un PRBS de 4 bits a una velocidad de generación de 133 MHz (máxima frecuencia alcanzada por el FPGA Sparta-6 SP605 Evaluation kit). Claramente se puede observar en este ejemplo que si se utiliza el reloj interno del FPGA sin antes hacer un divisor de frecuencia para tener una señal cuadrada, se tendrán esas perturbaciones que no definen de manera adecuada un pulso alto o un bajo o lo que es lo mismo uno “0” o un “1” lógico.



**Figura 55. Generador de datos PRBS de 4 bits (amarillo) obtenidos a partir de un reloj de 133MHz (verde) con el FPGA SP605 Spartan 6.**

Es fácil identificar un generador de 4 bits en el osciloscopio ya que tendremos  $2^4-1$  valores distintos en cada ciclo, es decir 15 valores distintos en cada ciclo, esto es gracias que podemos visualizar los datos en el osciloscopio tal como se muestra en la imagen anterior, sin embargo cuando incrementamos el valor de bits del generador se hace imposible determinar a simple vista esta cantidad de valores distintos. Para poder verificar estos datos es posible hacerlo mediante la transformada rápida de Fourier (FFT) de la señal, que nos despliega todas las componentes en frecuencia que existen en nuestra señal tal y como se muestra en la Figura 56.



**Figura 56. a) FFT (Fast Fourier Transform) con 5kHz de SPAM de una señal PRBS diseñada a 6 bits, b) FFT con 200 Hz de SPAM de la misma señal.**

Al hacer más pequeño el SPAM podremos ver las componentes que se repiten cada ciclo de repetición de los datos, para el ejemplo observamos que se repite cada 31.7Hz y si consideramos la frecuencia a la que se están generando los datos (2 KHz), para este caso, la división de la frecuencia del generador entre el ciclo de repetición de los datos obtendremos el valor adimensional de 63 que es equivalente a  $2^6-1$ , lo que significa que el generador es de 6 bits. De esta manera se analizan y verifican los generadores por encima de 4 bits.

#### 5.2.4. Para el modulador de datos se obtuvieron los siguientes resultados



**Figura 57. Resultados de la Modulación PPM.**

En la imagen de la Figura 57 se observan la codificación en 8-PPM de datos PRBS  $2^6-1$ , con un tamaño de palabra de 160 ns, lo cual equivale a 8 slots de 20 ns, de esta manera se tiene un pulso con duración de 20 ns en la posición que queda determinada según los datos que se están codificando.

La duración del impulso ( $T_p$ ) es de 20 ns porque es a esta velocidad que es posible generar una señal cuadrada con el FPGA LX9 Microboard. Sin embargo con esta velocidad es factible hacer una evaluación en un sistema de transmisión.

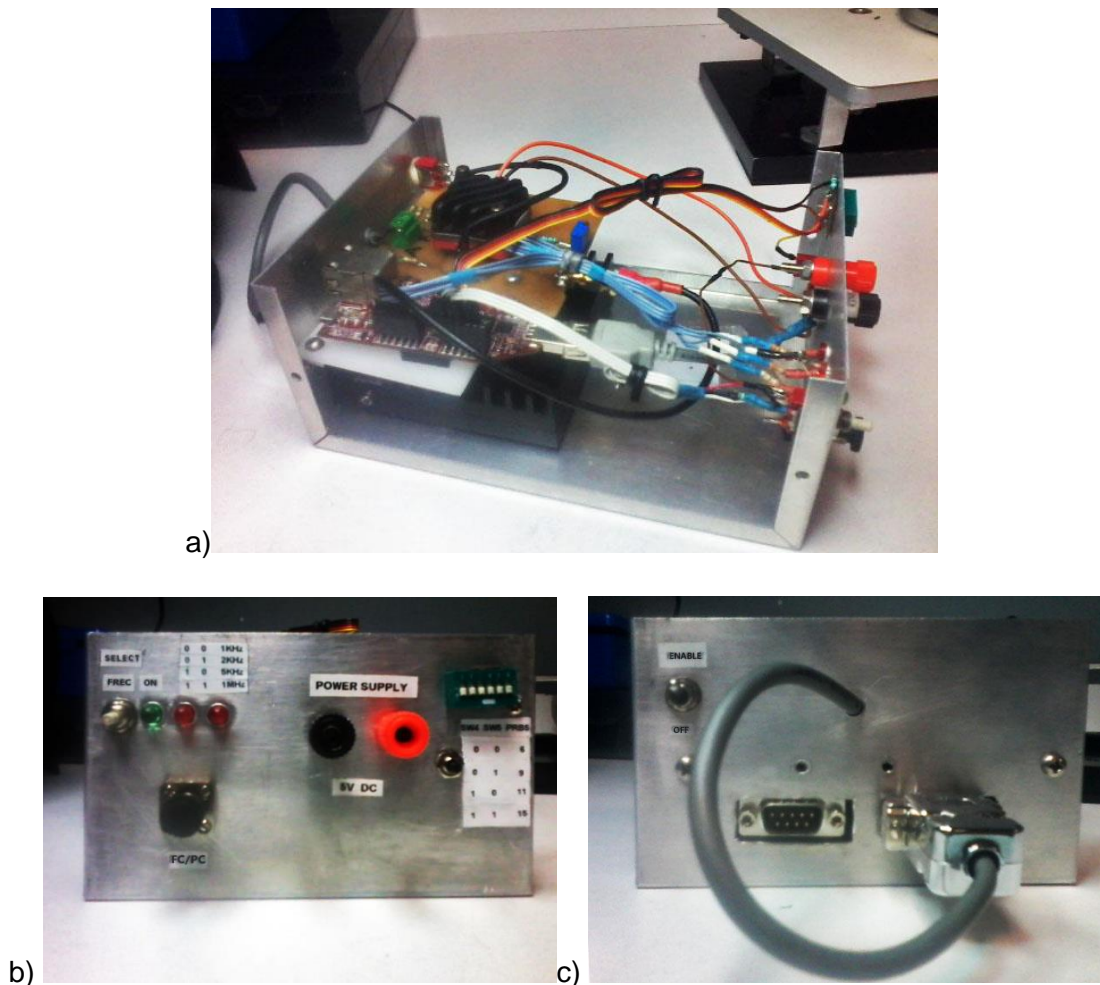
En la Tabla 4 se resumen la cantidad de recursos utilizados por el FPGA Spartan-6 para implementar el codificador 8-PPM.

**Tabla 4. Recursos utilizados por el FPGA Spartan-6 Microboard para codificar datos PRBS.**

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	19	11440	0%
Number of Slice LUTs	10	5720	0%
Number of fully used LUT-FF pairs	9	20	45%
Number of bonded IOBs	9	200	4%
Number of BUFG/BUFGCTRLs	1	16	6%

### 5.3. Sistema de transmisión

Como parte de los resultados se muestran las imágenes del transmisor desarrollado durante las actividades y posteriormente los resultados del enlace. Vease Figura 58.



**Figura 58. Vista superior a), de frente b) y trasera c) del transmisor desarrollado.**

El transmisor funcionó de manera adecuada al generar, controlar, convertir y guiar los datos PRBS por fibra óptica hasta la salida FC/PC, la parte de control quedó determinada por los Switchs de selección de PRBS 6bits, 9bits, 11 bits, y 15 bits, del mismo modo un push-button de selección de frecuencia que eran configurables en un rango de 1 Hz hasta 50 MHz mediante programación, además de indicadores ON/OFF, y frecuencia de datos; de esta manera se hicieron las pruebas de campo. El tamaño y las características que ofrece el Spartan 6 LX9 MicroBoard fueron ideales para implementarlo en el transmisor de una manera compacta, aunque fue necesario implementar alimentación USB 5v DC.



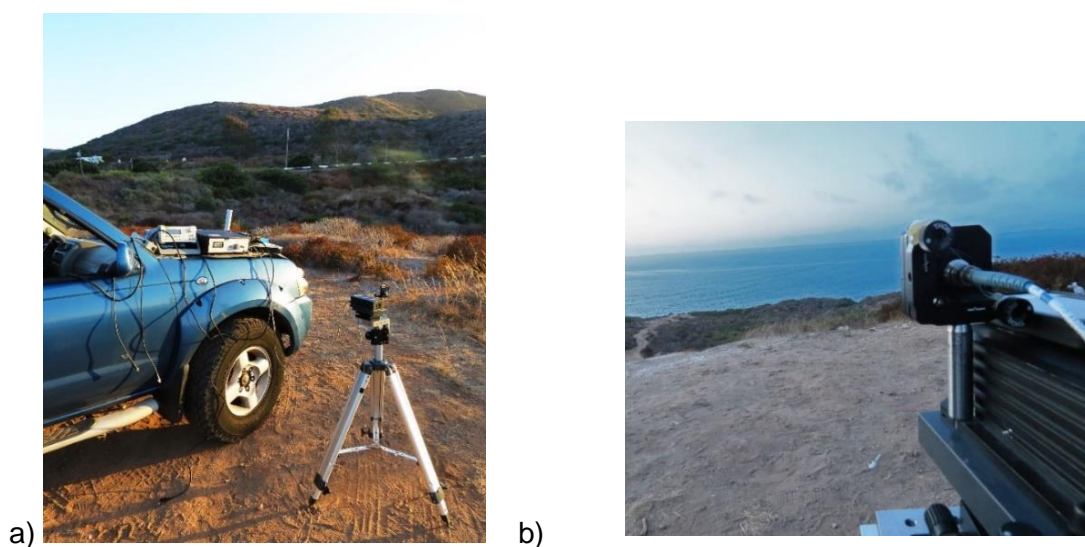
### 5.3.1. Enlace en campo abierto

En las imágenes de la Figura 56 se muestran la localización física del transmisor y receptor. El receptor con vista hacia punta banda.



**Figura 59. Vista interior a) y exterior b) del lugar donde se instaló el receptor.**

Y el transmisor se instaló en una zona de punta banda con vista hacia el CICESE, Figura 60.

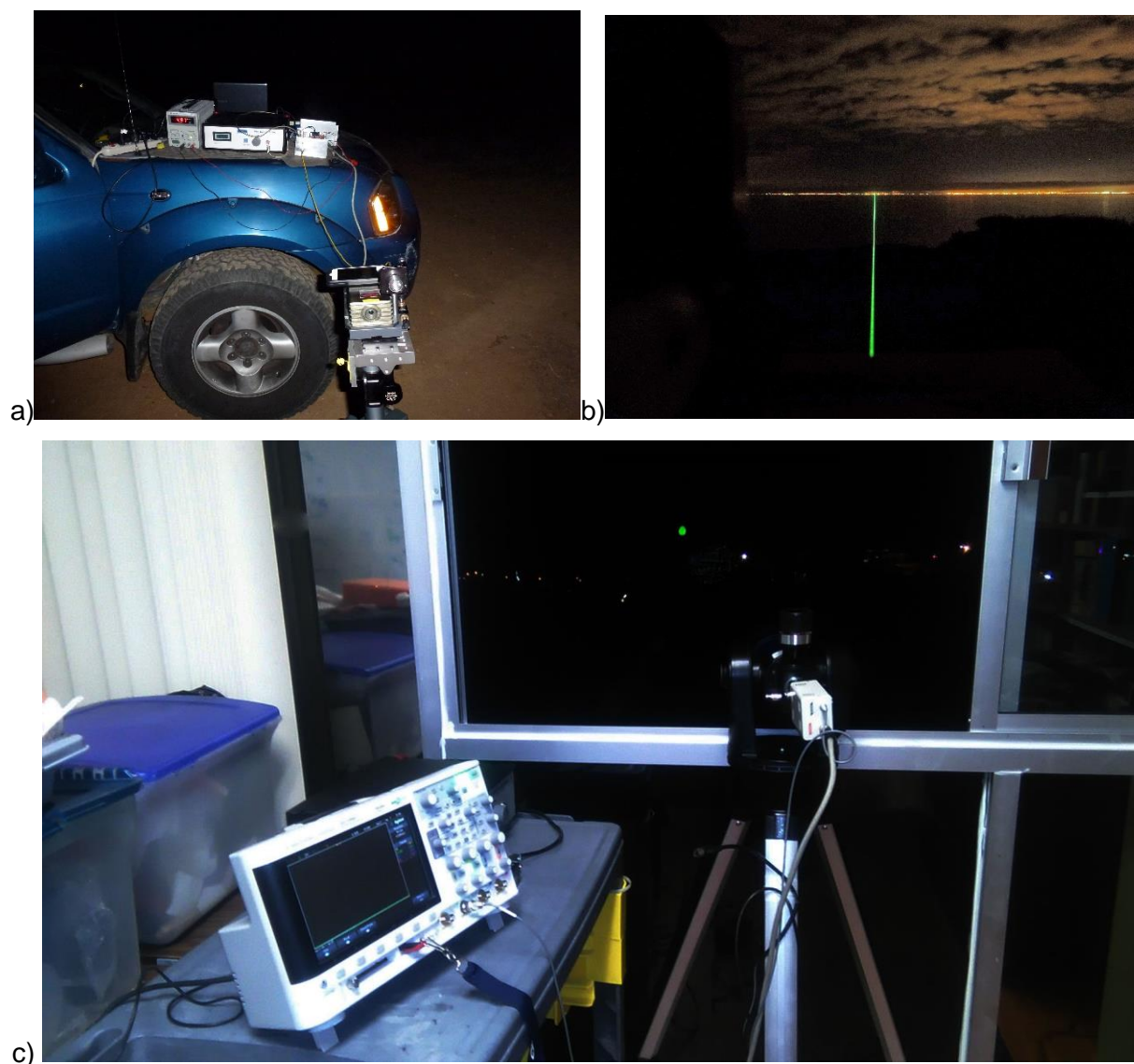


**Figura 60. Posición de la antena transmisora, a) y vista de la antena transmisora, b).**

Para poder alimentar los componentes electrónicos que conforman el transmisor se utilizó un inversor para baterías de autos.

También se utilizó un láser verde como guía de apuntamiento del láser infrarrojo; esto es de mucha utilidad ya que aunque se calculen los ángulos de apuntamiento en muchos casos no se tienen los instrumentos la escala suficiente para poder ajustar exactamente los componentes.

En las siguientes imágenes de la Figura 61 se muestra el transmisor, el canal y el receptor funcionando.



**Figura 61. Transmisor, a), Canal, b) y Receptor, c).**

Obteniéndose los resultados de la Figura 62.

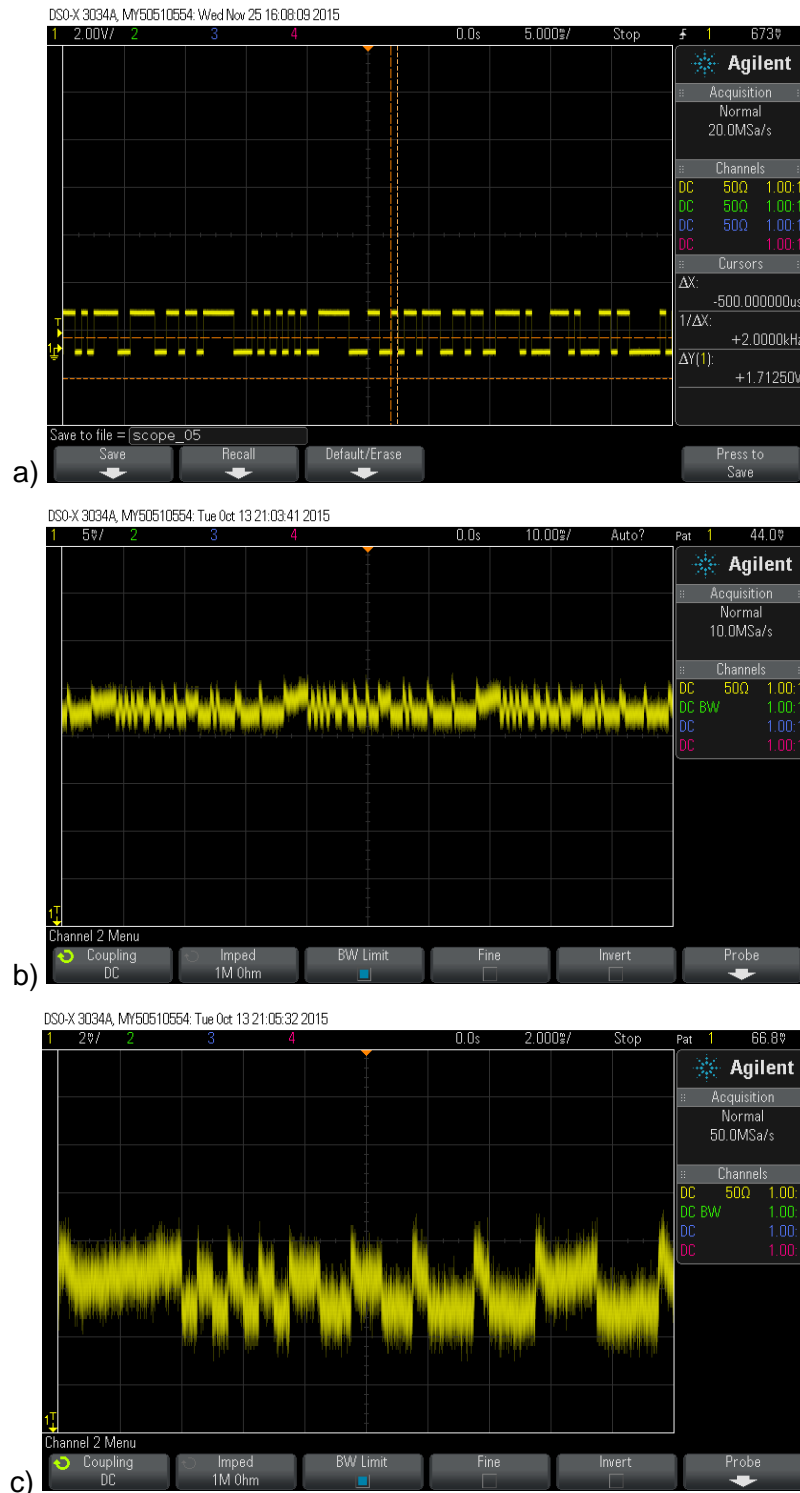


Figura 62. Señal PRBS enviada, a), datos recuperados por el receptor muestreados a 10Msa/s, b), datos recuperados por el receptor muestreados a 50 Msa/s, c).

## CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

Durante el desarrollo de este proyecto se hizo hincapié en la investigación de los sistemas de comunicaciones ópticas y la utilización de tecnología FPGA dentro de estos sistemas, para posteriormente implementar un sistema de comunicación en espacio libre utilizando un FPGA como generador de datos PRBS para simular datos reales. Algo que hay que dejar en claro es que en los proyectos de investigación se tiene que hacer retroalimentación constante en cada una de las etapas de todo el proceso de investigación, de esta manera se rectifican resultados no esperados, o simplemente se adquieren nuevos conocimientos.

En el estudio de los FPGAs, se determinó que en cualquiera de los que se tienen disponibles en el CICESE es posible implementar el sistema generador y codificador de datos, sin embargo como se explicó anteriormente, los resultados varían de acuerdo a las características de cada FPGA.

Para la comprensión de los LFSR (fundamentales para la generación de datos PRBS) fue necesario estudiar teoría de campos finitos ya que a partir de esta matemática se explica su funcionamiento. Una vez que se tuvo el conocimiento necesario para desarrollar el generador de datos PRBS se dio paso al diseño y programación del mismo.

La programación del FPGA para la generación de datos, se alcanzó la velocidad de 130 MHz con el FPGA Spartan-6 SP605 y de 200 MHz con el FPGA Virtex-6 con lo que demostramos la factibilidad de estos dispositivos como generadores de datos PRBS, aunque no se hicieron las pruebas con el Virtex-7 por limitaciones de tiempo, sin embargo al analizar sus características evidentemente se puede implementar el generador de datos PRBS en este FPGA alcanzando velocidades de 400 MHz. En relación al tamaño de bits del PRBS, como se demostró, es fácil implementarlo con un LFSR dado que esta técnica consta de registros de desplazamiento, en donde cada registro representa un bit, implementar un LFSR de 12 bits (12 registros), no representa problema para la alta densidad de circuitería lógica que ofrecen los FPGAs disponibles. Mejor aún se puede implementar PRBS de hasta 128 bits como si se tratase de un PRBS comercial pero para la aplicación que se le dio con una pseudoaleatoriedad de 12 bits o sea  $2^{12}-1=8192$  valores distintos es más que suficiente para ser considerada como una fuente de información real proveniente del medio físico.

Para el análisis y verificación de los datos PRBS se utilizó la herramienta FFT del osciloscopio, pero para un mejor análisis es posible utilizar un analizador de espectros, el cual tiene una mejor resolución, mayor ganancia en dB, y una velocidad de muestreo de 2GSa/s.

En el diseño del modulador PPM se comprendió la importancia de la utilización de este tipo de modulación en los sistemas de comunicación óptica, no obstante para su implementación en un FPGA, es necesario tener un grado de conocimientos en circuitos lógicos bastante desarrollado de lo contrario surgen problemas en la lógica del diseño o problemas de sincronía, aunado a esto

La parte del circuito anti rebote no se tenía contemplada dado que las primeras pruebas del generador PRBS se hicieron con el FPGA Spartan-6 SP605 que incluye en su tarjeta de evaluación salidas SMA, y switch de control ya integrados de mucha ayuda para el análisis del diseño, no obstante al utilizar el FPGA LX9 dentro de lo que conformaría el transmisor, fue necesario su utilización, y se llegó a implementar de manera exitosa en los diseños.

En el desarrollo experimental se integró toda la investigación y la programación hecha en el FPGA, por una parte estaba el Transmisor conformado principalmente por el FPGA que no es más que el resultado de investigación, diseño y programación realizadas en las primeras etapas de este proyecto y por otra parte está el Receptor que dio pauta para analizar el transmisor construido. Ambas partes constituyeron el sistema de transmisión en el espacio libre.

Con el objeto de comprobar la factibilidad del sistema de comunicación se hicieron las pruebas de campo en donde se hizo la instalación del transmisor y receptor con línea de vista a una distancia de 16 Km, y como resultado de estas pruebas se verificó la funcionalidad del transmisor, y la factibilidad de usar el telescopio como antena receptora, resultando un enlace realizado de manera efectiva al poder recuperar los datos transmitidos y de esta manera examinar el correcto funcionamiento de todos y cada uno de los elementos que conforman el sistema de transmisión en espacio libre.

## CAPÍTULO 7. COMPETENCIAS DESARROLLADAS Y/O APLICADAS

**Tabla 5. Competencias desarrolladas y/o aplicadas durante la residencia profesional.**

<b>Competencia desarrollada y/o aplicada</b>	<b>¿Dónde se aplicó?</b>
Instrumentales cognitivas lógicas y analógicas	<ul style="list-style-type: none"> <li>• Al diseñar y programar FPGAs utilizando la lógica de circuitos digitales.</li> <li>• Al diseñar circuitos digitales y plasmarlos en un lenguaje de programación (VHDL).</li> </ul>
Instrumentales metodológicas	<ul style="list-style-type: none"> <li>• Al organizar el tiempo de entrega del proyecto.</li> <li>• Solucionar los distintos problemas que fueron surgiendo durante el desarrollo del proyecto (circuito anti-rebote, sincronía)</li> </ul>
Instrumentales tecnológicas	<ul style="list-style-type: none"> <li>• Al aplicar y mejorar unos de los lenguajes de programación ya adquirido durante la carrera en la materia de “Diseño Digital con VHDL”.</li> <li>• Al utilizar conocimientos ya adquiridos en la materia de “Fibras Ópticas”.</li> <li>• Al utilizar los instrumentos de medición como el analizador de espectros, OTDR, medidor de potencia, etc.</li> <li>• Al diseñar y programar un FPGA y conocer un nuevo entorno de desarrollo como lo es ISE Designe Suite y VIVADO Design Suite.</li> </ul>
Instrumentales lingüísticas	<ul style="list-style-type: none"> <li>• Manejar términos en ingles aplicados en el área de la óptica y comunicaciones.</li> </ul>
Interpersonales individuales	<ul style="list-style-type: none"> <li>• Adaptación al entorno, al tratarse de un lugar de trabajo diferente (laboratorio de las comunicaciones Photonicas) y una</li> </ul>

	ciudad distinta a la de procedencia (Ensenada, B.C.).
Interpersonales sociales	<ul style="list-style-type: none"> <li>• Trabajo en equipo al hacer las pruebas de campo del sistema de comunicación óptica por espacio libre.</li> <li>• Al relacionar con doctores que evidentemente tienen un grado mayor de estudios y una perspectiva diferente de la investigación en México.</li> </ul>
Sistemáticas	<ul style="list-style-type: none"> <li>• Al poder relacionar la teoría de LFSR con la implementación de los mismos en un FPGA.</li> <li>• Al aprender de lo que trata la investigación en los centros de investigación de México.</li> </ul>
Sistemáticas de liderazgo	<ul style="list-style-type: none"> <li>• Al ser asesorado por doctores del CICESE y crear una estimulación intelectual.</li> </ul>

## FUENTES DE INFORMACIÓN

- [Álvarez, 2012]** Álvarez A. O. (2012). *Estudio y diseño de un receptor Mejorado 4-PPM*. Trabajo final de carrera. Universidad Politécnica de Catalunya, Catalunya, España. 7 p.
- [Artes y Pérez, 2012]** Artés R. A.; Pérez G. F., *et al.* (2012). *Comunicaciones Digitales*. España. 3 p.
- [Checa y Velázquez, 2011]** Checa R. E.; Velásquez C. D. (2011). *Implementación de códigos de línea en una tarjeta de entrenamiento basada en un FPGA*. Proyecto de Titulación. Facultad de Ingeniería eléctrica y electrónica, Escuela Politécnica Nacional, Quito, Ecuador. 4-16 pp.
- [Hemmati, 2005]** Hamid Hemmati (2005). *Deep Space Optical Communications*. California Institute of Thechnology. 234 p.
- [Maxinez y Alcalá, 2002]** Maxinez D. G.; Alcalá J. (2002). *VHDL, El arte de programar sistemas digitales*. México: Editorial Continental. 35-37 pp.
- [Sánchez, 2014]** Sánchez G. (2014). *Análisis e implementación de sistemas de modulación digital ASK, FSK, M-PSK y M-QAM mediante la programación de código VHDL utilizando la tecnología FPGA*. Tesis de Grado. Escuela superior politécnica de Chimborazo, Riobamba, Ecuador. 36-38 pp.
- [Santos y Quintero, 2015]** Santos J.; Quintero R. (2015). *Enlace Óptico En Espacio Libre 1M Sdl-5430*. (Reporte de Verano Científico), Centro de Investigación y de Educación Superior de Ensenada, Baja California, México. 10-11pp.



## Referencias en Internet

- [1] What is Free Space Optical Communications? En Harvard Broadband. Consultado en Noviembre 2015. Disponible en:  
<http://web.archive.org/web/20090417063155/http://www.seas.harvard.edu/hbbcl/fsoc.html>
- [2] Generador de secuencia binaria Pseudo Aleatoria. En C7 Tecnología consultado en Septiembre 2015, disponible en: [http://c7t-dl.com/Docs/C7T\\_NT12\\_PRBS\\_LFSR.pdf](http://c7t-dl.com/Docs/C7T_NT12_PRBS_LFSR.pdf)
- [3] Xilinx (2015), FPGAs, En Xilinx. Consultado en Noviembre del 2015. Disponible en:  
<http://www.xilinx.com/products/silicon-devices/fpga.html>
- [4] General Purpose Driver for Laser Diodes WLD3343. En Teamwavelength. Consultado en Septiembre 2015. Disponible en:  
<http://www.teamwavelength.com/downloads/datasheets/wld3343.pdf>
- [5] THORLABS (2015), Butterfly Laser Diode Mounts. En THORLABS. Consultado en Noviembre del 2015. Disponible en:  
[https://www.thorlabs.de/newgrouppage9.cfm?objectgroup\\_ID=1558](https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_ID=1558)
- [6] THORLABS (s.f), DataSheet F810FC-1550 Fiber Collimation Packages. En Thorlabs inc. Consultado en Noviembre del 2015. Disponible en:  
<https://www.thorlabs.com/thorcat/13600/F810FC-1550-AutoCADPDF.pdf>
- [7] LX-200 MEADE Telescope. En MEADE Instruments. Disponible en:  
<http://www.meade.com/products/telescopes/lx200.html>
- [8] HAMAMATSU (2015), APD Modules. En Hamamatsu. Consultado en Noviembre del 2015. Disponible en:  
<http://www.hamamatsu.com/jp/en/product/alpha/A/4162/index.html>
- [9] FEMTO (2015), 100/200 MHz Wideband Voltage Amplifier Series DHPVA. En Femto. Consultado en Noviembre del 2015. Disponible en:  
<http://www.femto.de/en/products/voltage-amplifiers/variable-gain-100-200-mhz-dhpva>

- [10] AGILENT (2015), Agilent InfiniiVision 300 X-Series Oscilloscopes User Guide. En Agilent. Consultado en Noviembre del 2015. Disponible en: [http://web.mit.edu/6.115/www/document/agilent\\_mso-x\\_manual.pdf](http://web.mit.edu/6.115/www/document/agilent_mso-x_manual.pdf)
- [11] SMA (conector) En Wikipedia. Consultado en octubre del 2015. Disponible en [https://es.wikipedia.org/wiki/SMA\\_%28conector%29](https://es.wikipedia.org/wiki/SMA_%28conector%29)
- [12] CCM (2015), Conector DB9. En *KiosKea*. Consultado den Noviembre del 2015. Disponible en: <http://es.ccm.net/contents/176-conector-db9>
- [13] Acoplador Fc/Pc. En Alibaba. Consultado en noviembre del 2015. Disponible en <http://spanish.alibaba.com/product-gs/fc-pc-fiber-optical-square-adapter-fc-upc-sm-optic-coupler-60307652363.html>
- [14] Divisor de frecuencia con VHDL. En Estado Finito. Consultado en agosto del 2015. Disponible en <http://www.estadofinito.com/divisor-frecuencia-vhdl/>

## ANEXO A “GUÍA PARA LA CREACIÓN DE ARCHIVOS VHDL A TRAVÉS DE XILINX ISE DESIGN SUITE”

### Xilinx® ISE® Design Suite

El software que provee el Kit de Evaluación SP605 es presentado con el nombre de Xilinx ISE Design Suite.

ISE Design Suite controla todos los aspectos del flujo del diseño. A través de la interfaz de “Project Navigator”, también se puede acceder a todos los diseños de entrada y herramientas para la implementación de diseños. Además podemos acceder a archivos y documentos asociados con nuestro proyecto.

The screenshot displays the Xilinx ISE Design Suite Project Navigator interface. The main window shows the 'Design Summary' for a project named 'constantes.xise'. The interface is divided into several panels:

- Design Overview:** Contains a table of project status information.
 

gen_datos Project Status (07/27/2015 - 15:31:54)			
Project File:	constantes.xise	Parser Errors:	No Errors
Module Name:	gen_datos	Implementation State:	Programming File Generated
Target Device:	xc6slx45t-3fgg484	Errors:	No Errors
Product Version:	ISE 14.1	Warnings:	No Warnings
Design Goal:	Balanced	Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	All Constraints Met
Environment:	System Settings	Final Timing Score:	0 (Timing Report)
- Device Utilization Summary:** A table showing the utilization of logic resources.
 

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	82	54,576	1%	
Number used as Flip Flops	82			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	158	27,288	1%	
Number used as logic	150	27,288	1%	
Number using O6 output only	88			
Number using O5 output only	60			
Number using O5 and O6	2			
- Warnings:** A list of warning messages at the bottom of the window.
 

```
WARNING:ProjectMgmt - File C:/Users/Byakuya/Documents/Files Verano/FPGA/constantes/medio_ciclo_summary.xml is missing.
WARNING:ProjectMgmt - File C:/Users/Byakuya/Documents/Files Verano/FPGA/constantes/medio_ciclo_usage.xml is missing.
WARNING:ProjectMgmt - File C:/Users/Byakuya/Documents/Files Verano/FPGA/constantes/medio_ciclo_xst.xrpt is missing.
```

Figura A1. Navegador de proyectos (Project Navigator).

Por default, la interfaz del Project Navigator está dividida en 4 paneles, así como se puede observar en la Figura A1. En la parte superior izquierda están los paneles de Start, Design, Files y Libraries, los cuales incluyen despliegue y acceso a los archivos fuente del proyecto así también dan el acceso a manejar procesos para la fuente seleccionada actualmente. El panel Start da un rápido acceso a los proyectos actualmente abiertos, material de referencia frecuentemente utilizado, documentación y tutoriales. En el fondo del Project Navigator se encuentran los paneles de Console, Errors y Warnings. A la derecha esta una interfaz de

múltiples documentos (MDI) ventana referida como “Workspace” espacio de trabajo. El área de Trabajo permite mirar informes del diseño, archivos de texto, esquemáticos y simulaciones.

## **Panel de Diseño (Design Pane)**

El panel de diseño da acceso a los paneles de ver, jerarquía y de procesos (view pane, Hierarchy pane y processes pane respectivamente).

View Pane: El panel Ver tiene disponible dos botones tipo radio para ver los módulos fuente asociados con la implementación o simulación de diseños en el panel de jerarquía.

Hierarchy Pane: El panel de jerarquía se despliega el nombre del proyecto, el dispositivo destino, documentos de usuario y archivos fuente de diseños asociados con el diseño seleccionado.

Cada archivo en el panel de jerarquía tiene un icono asociado. El icono indica el tipo de archivo (Archivo HDL, esquemático, core, o archivo de texto, por ejemplo)

Processes Pane: El panel de procesos puede ser diferente dependiendo del contexto, y cambia de acuerdo al tipo de fuente seleccionado en el panel de Fuentes en el nivel más alto de la fuente del proyecto. Desde el panel de procesos se pueden correr las funciones necesarias para definir, compilar y analizar los diseños. La ventana de procesos da acceso a las siguientes funciones:

- **Design Summary/Reports**

Facilita el acceso a reportes, mensajes y resumen de resultados del diseño.

- **Design Utilities**

Da acceso para la generación del símbolo, ver el historial de líneas de comando, y ver instancias HDL.

- **User Constraints**

Da acceso a editar, direccionar y cronometrar restricciones.

- **Synthesis**

Da acceso para revisar la sintaxis, síntesis, ver el RTL o Tecnología del esquemático, y síntesis de reportes.

- **Implement Design**

Da acceso a las herramientas de implementación y el análisis de post-implementación.

- **Generate Programming File**

Da acceso para la generación del Bitsream (archivo para descargar en el FPGA).

- **Configure Target Device**

Da acceso a las herramientas de configuración para crear archivos de programación y programar el dispositivo.

## **Files Panel**

El Panel de archivos provee una lista ordenable de todos los archivos fuente del proyecto. Las propiedades de cada archivo puede ser vista y modificada haciendo clic derecho sobre el archivo y seleccionando Source Properties.

## **Libraries Panel**

El panel de librerías permite manejar librerías HDL y archivos fuente relacionados. El usuario puede crear, ver, y editar librerías y fuentes asociadas.

## **Console Panel**

La consola provee toda la producción estándar de procesos manejados por el Project Navigator. Este despliega errores, avisos, y mensajes de información.

## **Workspace**

El espacio de trabajo es donde el editor de diseño, vistas, y herramientas de análisis se abren. Estos incluyen el Editor de Texto de ISE, el Editor esquemático, el Editor de restricciones, el visualizador de reportes y resúmenes del diseño, el visualizador de RTL y el analizador del cronometrado

Otras herramientas como lo es el PlanAhead™, (Herramienta para declarar entradas y salidas I/O), ISim (Herramienta de simulación) y el iMPACT (Herramienta de programación del FPGA) se abren en una ventana independiente del Project Navigator cuando es llamado.

## **HDL**

### **VHDL o Verilog**

ISE Design Suite soporta ambos diseños y se pueden aplicar ambos diseños simultáneamente. El usuario debe decidir qué tipo lenguaje HDL desea trabajar.

### **Iniciando ISE Design Suite**

Para iniciar el software hacemos doble clic sobre el icono del Project Navigator.

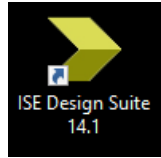


Figura A2. Icono de ISE Design Suite 14.1.

### Creando un Nuevo Proyecto.

Para crear un nuevo proyecto usando New Project Wizard, hacemos lo siguiente:

1. Desde el Project Navigator, seleccionamos **File > New Project**.
2. En el campo de **Location** estará la localización del archivo
3. En el campo de **Name**, introducir el nombre del archivo.
4. Verificar que HDL está seleccionado como el tipo de fuente principal y dar clic en **Next**.

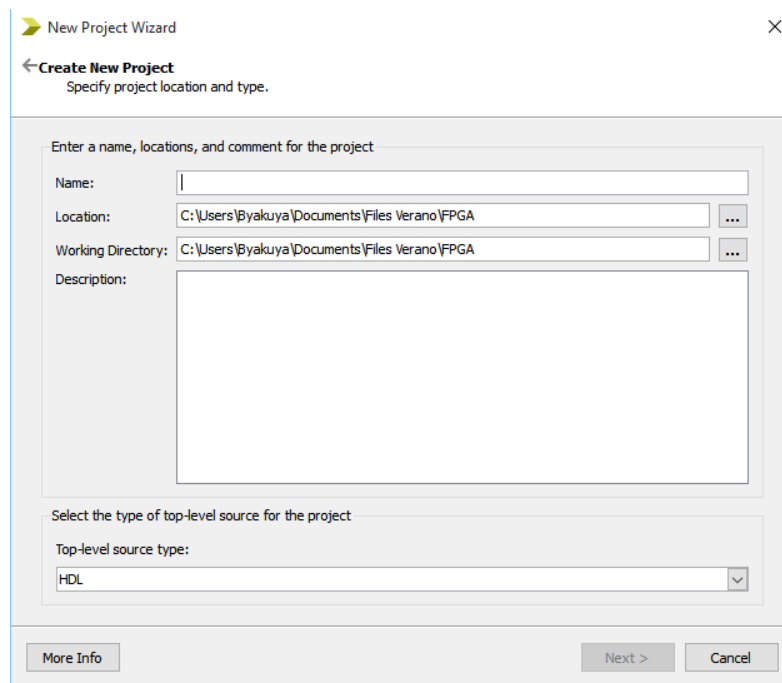
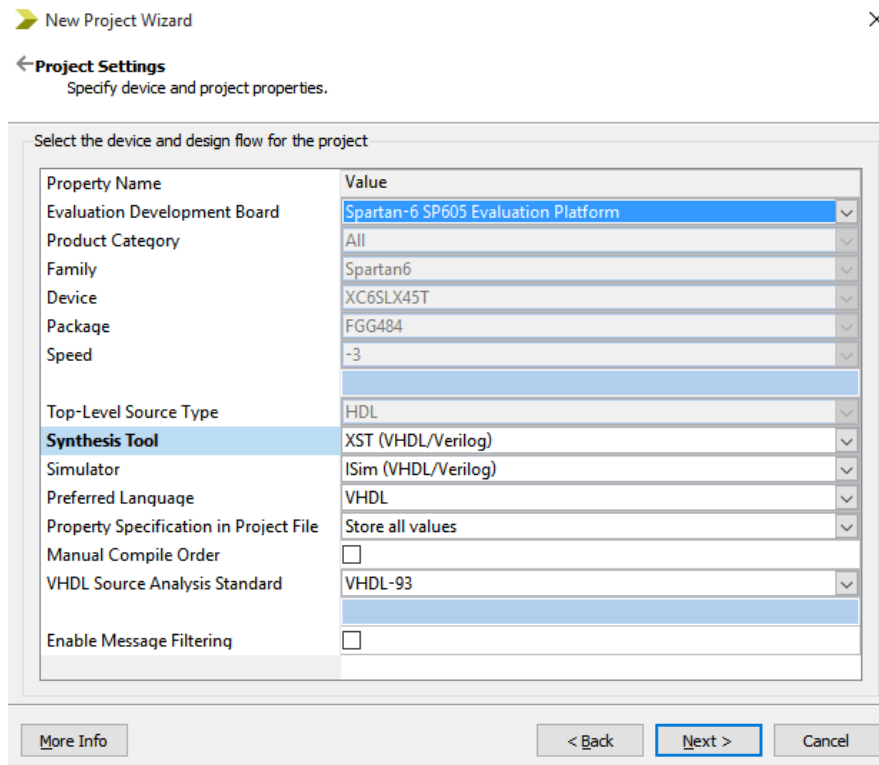


Figura A3. New Project Wizard – Creando un nuevo proyecto.

5. Para el FPGA SP605 seleccionamos los siguientes valores:
  - Evaluation Development Board: **Spartan-6 SP605 Evaluation Platform**. Por default establecerá las demás características de **Family, Device, Package y Speed**.
  - Synthesis Tool: **XST (VHDL/Verilog)**.
  - Simulator: **ISim (VHDL/Verilog)**.

- Preferred Language: VHDL o Verilog dependiendo. Esto determinará el lenguaje por defecto y la generación de todos los archivos HDL. (En nuestro caso VHDL).
6. Clic en **Next**, y luego **Finish** para completar la creación del proyecto.



**Figura A4. New Project Wizard – Propiedades del dispositivo.**

## Design Description

### Creando un Módulo base en HDL

Para crear un módulo desde un código HDL. Con ISE Design Suite, es posible crear fácilmente módulos de DHL usando el ISE Text Editor.

### Usando el New Sourse Wizard y el ISE Text Editor

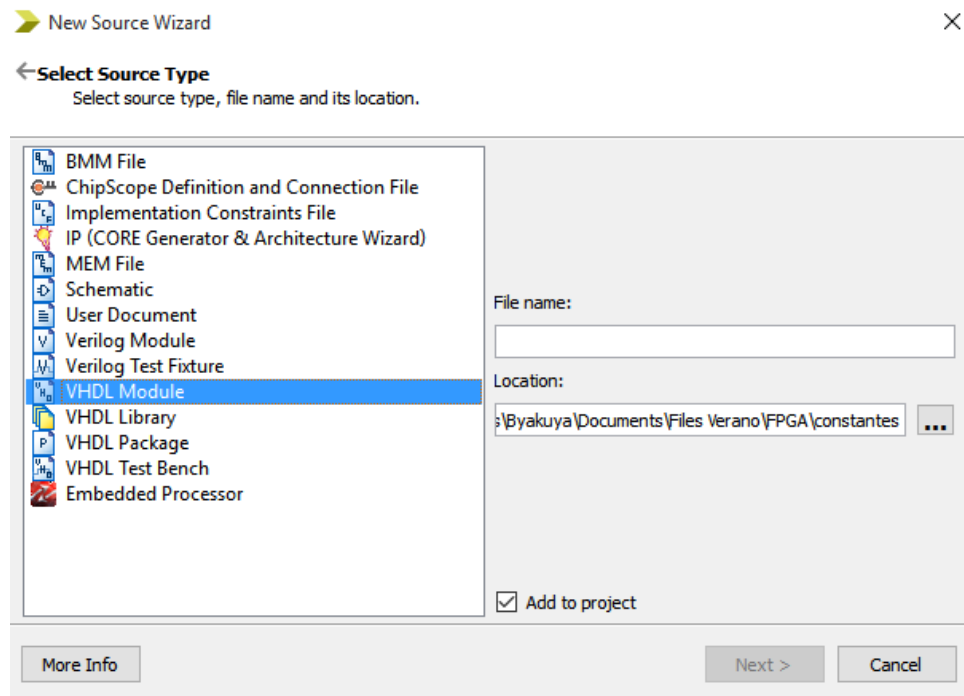
En esta sección, se creará un archivo usando el New Sourse Wizard, especificando el nombre y los puertos I/O del componente. EL archivo HDL resultante entonces modificable en el ISE Design Suite.

Para crear el archivo fuente, hacemos lo siguiente:

1. Seleccionamos **Project > New Source**.

El New Source Type se abre en el cuál tu especificas el tipo de fuente que quieres crear.

2. En la página de Select Source Type, seleccionamos **VHDL Module** o **Verilog Module**.
3. En File Name, escribimos el nombre que queremos para el modulo.



**Figura A5. New Source Wizard – Asistente de Nueva Fuente.**

4. Clic en Next.
5. En la página de Define Module, se designan las entradas, salidas o entradas-salidas del Módulo (Opcional).
6. Clic Next para ver un resumen de todo el modulo que se está creando.
7. Clic en Finish para abrir un archivo DHL vacío en el ISE Text Editor.

A continuación un ejemplo de un archivo VHDL.



```
1 -----
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date:    14:10:52 08/28/2015
6 -- Design Name:
7 -- Module Name:    Report - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --|
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32
33 entity Report is
34 end Report;
35
36 architecture Behavioral of Report is
37 begin
38
39
40 end Behavioral;
```

Figura A6. Archivo VHDL en el ISE Text Editor.

## ANEXO B “TABLA DE LFSR DE GALOIS”

### Tabla de LFSRs

Roy Ward, Tim Molteno

Octubre 26, 2007

Aquí hay una tabla de los “taps” (bits para hacer la relación XOR) del LFSR para conseguir el máximo ciclo de pseudoaleatoriedad. La numeración empieza de  $n \dots 1$  siendo  $n$  el bit de entrada y 1 el bit de salida. La Figura B1 muestra un LFSR de 8 bits con máximo ciclo. LFSR-2 se refiere a la utilización de 2 “taps” como realimentación, LFSR-4 se refiere a 4 “taps” como realimentación. Un espacio en blanco indica que no existe solución.

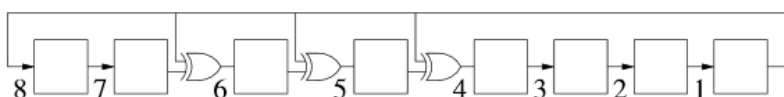


Figura B1. LFSR Galois de 8 bits con un ciclo de 255. Este LFSR tiene cuatro “taps” en las posiciones 8, 6, 5 y 4.

Tabla B1. Registros de desplazamiento con tamaño de ciclo  $2^n - 1$ .

$n$	LFSR-2	LFSR-4	$n$	LFSR-2	LFSR-4	$n$	LFSR-2	LFSR-4
2	2, 1		24		24, 23, 21, 20	46		46, 40, 39, 38
3	3, 2		25	25, 22	25, 24, 23, 22	47	47, 42	47, 46, 43, 42
4	4, 3		26		26, 25, 24, 20	48		48, 44, 41, 39
5	5, 3	5, 4, 3, 2	27		27, 26, 25, 22	49	49, 40	49, 45, 44, 43
6	6, 5	6, 5, 3, 2	28	28, 25	28, 27, 24, 22	50		50, 48, 47, 46
7	7, 6	7, 6, 5, 4	29	29, 27	29, 28, 27, 25	51		51, 50, 48, 45
8		8, 6, 5, 4	30		30, 29, 26, 24	52	52, 49	52, 51, 49, 46
9	9, 5	9, 8, 6, 5	31	31, 28	31, 30, 29, 28	53		53, 52, 51, 47
10	10, 7	10, 9, 7, 6	32		32, 30, 26, 25	54		54, 51, 48, 46
11	11, 9	11, 10, 9, 7	33	33, 20	33, 32, 29, 27	55	55, 31	55, 54, 53, 49
12		12, 11, 8, 6	34		34, 31, 30, 26	56		56, 54, 52, 49
13		13, 12, 10, 9	35	35, 33	35, 34, 28, 27	57	57, 50	57, 55, 54, 52
14		14, 13, 11, 9	36	36, 25	36, 35, 29, 28	58	58, 39	58, 57, 53, 52
15	15, 14	15, 14, 13, 11	37		37, 36, 33, 31	59		59, 57, 55, 52
16		16, 14, 13, 11	38		38, 37, 33, 32	60	60, 59	60, 58, 56, 55
17	17, 14	17, 16, 15, 14	39	39, 35	39, 38, 35, 32	61		61, 60, 59, 56
18	18, 11	18, 17, 16, 13	40		40, 37, 36, 35	62		62, 59, 57, 56
19		19, 18, 17, 14	41	41, 38	41, 40, 39, 38	63	63, 62	63, 62, 59, 58
20	20, 17	20, 19, 16, 14	42		42, 40, 37, 35	64		64, 63, 61, 60
21	21, 19	21, 20, 19, 16	43		43, 42, 38, 37	65	65, 47	65, 64, 62, 61
22	22, 21	22, 19, 18, 17	44		44, 42, 39, 38	66		66, 60, 58, 57
23	23, 18	23, 22, 20, 18	45		45, 44, 42, 41	67		67, 66, 65, 62

## ANEXO C “PROGRAMAS EN VHDL”

### Bloque divisor de frecuencia

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa generador de pulsos cuadrados
5  -- frecuencia de reloj/2
6  library IEEE;
7  use IEEE.STD_LOGIC_1164.ALL;
8
9  entity Pulse is
10     port (USER_SMA_GPIO_N,USER_SMA_GPIO_P: inout std_logic:= '0'; -- Salidas SMA
11           USER_CLOCK :      inout  std_logic); -- Entrada Reloj del FPGA
12 end Pulse;
13
14 architecture Behavioral of Pulse is
15     -- components--
16     component base_de_tiempo1 is
17     port
18         (USER_CLOCK :      in  std_logic; -- Declaración de componente que hace la
19          USER_SMA_GPIO_N,USER_SMA_GPIO_P: inout std_logic:= '0'); -- división de frecuencia
20     end component;
21     -- signals--
22
23
24 begin
25
26     BT: base_de_tiempo1 port map
27         (USER_CLOCK,USER_SMA_GPIO_N,USER_SMA_GPIO_P); -- Llamada a divisor de frecuencia
28
29 end Behavioral;

```

### Bloque circuito anti-rebote

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa anti-rebote para interruptores
5
6  library IEEE;
7  use IEEE.STD_LOGIC_1164.ALL;
8  use ieee.std_logic_unsigned.all;
9
10
11 entity anti is
12
13     port (   clk: in std_logic; -- la frecuencia de clk tiene que ser el
14            swin: in std_logic; -- el inverso del tiempo que duran los
15            debounced_out: inout std_logic); -- rebotes no desados del interruptor
16
17 end anti;
18
19 architecture Behavioral of anti is
20     --señales
21     signal currents,nexts : std_logic_vector (1 downto 0);
22     signal conteo: std_logic_vector (7 downto 0) := "00000000";
23     signal pulso : std_logic;

```

```

24
25 begin
26 --lógica combinacional
27 nexts <= "00" when swin = '1' else
28     "01" when currentns = "00" else "10";
29
30 pulso <= '0' when currentns = "10" else '1';
31 debounced_out <= pulso;
32
33 -- lógica secuencial
34 process (clk,pulso)
35 begin
36     if (pulso'event and pulso = '1') then
37         conteo <= conteo +1;
38     end if;
39
40     if clk'event and clk = '1' then
41         currentns <= nexts;
42     end if;
43 end process;
44 end Behavioral;

```

### Componente base de tiempo para circuito Anti-rebote

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa base de tiempo para componenete anti-rebote
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  entity BaseAnti is
8
9      port (
10         clk: in std_logic;
11         reset : in std_logic;
12         clk3: inout std_logic
13     );
14 end BaseAnti;
15
16 architecture Behavioral of BaseAnti is
17 signal count2: integer :=1;
18
19 begin
20     process (clk)
21     begin
22         if (clk'event and clk = '1') then --- 100Mhz/400000 = 250
23             count2 <= count2 +1; --- inverso de 250 = 4 ms
24             if (count2=400000) then
25                 clk3 <= not clk3;
26                 count2 <=1;
27             elsif (count2>400001) then
28                 count2 <=1;
29
30             end if;
31         end if;
32     end process;
33 end Behavioral;

```

Tanto el divisor de frecuencia y el circuito anti-rebote sirven como componentes para integrar otros programas.

## Programa generador de datos PRBS

### Paquete de constantes a utilizar en el programa principal de PRBS

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Definición de Constantes a utilizar en Programa generador de PRBS
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.all;
7
8  package Constantes is
9      constant N : positive:=6;      -- Tamaño de LFSR
10     constant N2: positive:=9;      --
11     constant N3: positive:=11;     --
12     constant N4: positive:=15;     --
13     function and_vector (vector : in std_logic_vector(0 to N-1)) return std_logic;
14 end Constantes;
15 package body Constantes is
16
17 function and_vector (vector : in std_logic_vector (0 to N-1)) return std_logic is
18     variable resultado : std_logic;
19     begin
20         resultado := vector(0);
21         for I in 1 to N-1 loop
22             resultado := vector (I) and resultado;
23         end loop;
24         return resultado;
25     end and_vector;
26
27 function and_vector2 (vector2 : in std_logic_vector (0 to N2-1)) return std_logic is
28     variable resultado2 : std_logic;
29
30     begin
31         resultado2 := vector2(0);
32         for I in 1 to N2-1 loop
33             resultado2 := vector2 (I) and resultado2;
34         end loop;
35         return resultado2;
36     end and_vector2;
37
38 function and_vector3 (vector3 : in std_logic_vector (0 to N3-1)) return std_logic is
39     variable resultado3 : std_logic;
40     begin
41         resultado3 := vector3(0);
42         for I in 1 to N3-1 loop
43             resultado3 := vector3 (I) and resultado3;
44         end loop;
45         return resultado3;
46     end and_vector3;
47
48 function and_vector4 (vector4 : in std_logic_vector (0 to N4-1)) return std_logic is
49     variable resultado4 : std_logic;
50     begin
51         resultado4 := vector4(0);
52         for I in 1 to N4-1 loop
53             resultado4 := vector4 (I) and resultado4;
54         end loop;
55         return resultado4;
56     end and_vector4;
57 end Constantes;

```

## Componente Registro

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa registro con Flip Flop D
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9  entity Registros is
10   port (clk2,reset,D : in std_logic;
11         Q : out std_logic);
12 end Registros;
13 architecture Behavioral of Registros is
14 begin
15     process (clk2,reset)
16     begin
17         if reset='1' then
18             Q <= '1';
19         elsif clk2'event and clk2='1' then
20             Q <= D;
21         end if;
22     end process;
23 end Behavioral;

```

## Componente “Base Tiempo” de control de PRBS

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa/componente Base de tiempo
5  -- En este programa se lleva a cabo la etapa de divisor de frecuencia y control de PRBS
6  library IEEE;
7  use IEEE.STD_LOGIC_1164.ALL;
8  use ieee.std_logic_unsigned.all;
9
10
11 entity Base_Tiempo is
12   port (
13     clk: in std_logic;
14     reset,swin : in std_logic;
15     S1,S2,S3: out std_logic;
16     clk2,clk3,debounced_out : inout std_logic;
17     RST: inout std_logic:='0'
18   );
19 end Base_Tiempo;
20
21 architecture Behavioral of Base_Tiempo is
22 -- señales --
23 signal count: integer :=1;
24
25 signal currents,nexts : std_logic_vector (1 downto 0);
26 signal conteo,conteo2 : std_logic_vector (7 downto 0) := "00000000";
27 signal pulso : std_logic;
28 signal y: integer:=1;
29 signal indicadores: std_logic_vector (2 downto 0);
30 -- proceso --
31 begin
32
33     nexts <= "00" when swin = '1' else -- Anti-rebote para push-button
34         "01" when currents = "00" else "10"; -- selector de frecuencia
35

```

```

36 pulso <= '0' when currents = "10" else '1';
37 debounced_out <= pulso;
38
39 process (clk3,reset, pulso,conteo)
40 begin
41   if reset = '1' then
42     conteo <= "00000000";
43   elsif (pulso'event and pulso = '1') then
44     conteo <= conteo +1;
45   end if;
46   if clk3'event and clk3 = '1' then
47     currents <= nexts;
48   end if;
49 end process;
50
51 Process (reset,conteo2,debounced_out)
52 begin
53   if reset='1' then
54     conteo2<= "00000000";
55   elsif (debounced_out'event and debounced_out='1') then
56     conteo2<=conteo2+1;
57   end if;
58
59   if (conteo2 = "00000101" ) then
60     conteo2 <= "00000000";
61   end if;
62 end process;
63
64 with conteo2 select                                -- Lógica combinacional para seleccionar frecuencia
65 y <= 0 when "00000000",
66     50000 when "00000001",
67     25000 when "00000010",
68     10000 when "00000011",
69     50 when "00000100",
70     0 when others;
71
72
73 with conteo2 select                                -- Lógica combinacional para indicadores de frec.
74 indicadores <= "000" when "00000000",
75               "100" when "00000001",
76               "101" when "00000010",
77               "110" when "00000011",
78               "111" when "00000100",
79               "000" when others;
80
81 (S1,S2,S3) <= indicadores;
82
83 with y select
84 RST <= '1' when 0,
85      '0' when others;
86
87 process (clk,y)                                    -- Logica secuencial para divisor de frec.
88 begin
89   if y=1 then
90     count <=1;
91   elsif y>1 then
92     if (clk'event and clk = '1') then
93       count <= count +1;
94       if (count=y) then
95         clk2 <= not clk2;
96         count <=1;
97       elsif (count>y) then
98         count <=1;
99       end if;
100     end if;
101   end if;
102 end process;
103 end Behavioral;

```

## Programa principal “generador de datos PRBS”

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa principal que une todos los componentes
5  -- para poder implementar el "generador de datos PRBS"
6  library IEEE;
7  use IEEE.STD_LOGIC_1164.ALL;
8  use IEEE.STD_LOGIC_ARITH.ALL;
9  use IEEE.STD_LOGIC_UNSIGNED.ALL;
10 use work.Constantes.All;
11
12 entity Pseudoaleatorios is
13     generic (Nreg      : positive := N;           -- Uso de paquetes de constantes
14             Nreg2     : positive :=N2;
15             Nreg3     : positive :=N3;
16             Nreg4     : positive :=N4);
17     port (reset,clk,swin : in  STD_LOGIC;
18          clk2,clk3 : inout STD_LOGIC;
19          RST,sw2,sw3: inout STD_LOGIC;
20          clk_datos, reloj,S1,S2,S3,datos,datos1,
21          datos2,datos3,datos4,datos5,sinc,sw33,sw22 : inout STD_LOGIC);
22 end Pseudoaleatorios;
23
24 architecture Behavioral of Pseudoaleatorios is
25     -- Componentes--
26     component Base_Tiempo is                       -- Uso de Componente "Base tiempo"
27     port (
28         clk: in std_logic;
29         reset,swin : in std_logic;
30         S1,S2,S3: out std_logic;
31         debounced_out,clk2,clk3 : inout std_logic;
32         RST: inout std_logic:='0'
33     );
34 end component;
35
36     component registros is                         -- uso componente Registros
37     port (clk2, reset,D : in std_logic;
38          Q               : out std_logic);
39 end component;
40
41     component BaseAnti is                         -- Uso componente base de tiempo para anti-rebote
42     port (
43         clk: in std_logic;
44         reset : in std_logic;
45         clk3: inout std_logic
46     );
47 end component;
48     -- Señales --
49     signal sig_xor : std_logic;
50     signal sig_xor2 : std_logic;
51     signal sig_xor3 : std_logic;
52     signal sig_xor4 : std_logic;
53     signal Q_int   : std_logic_vector(0 to Nreg-1);
54     signal Q_int2  : std_logic_vector(0 to Nreg2-1);
55     signal Q_int3  : std_logic_vector(0 to Nreg3-1);
56     signal Q_int4  : std_logic_vector(0 to Nreg4-1);
57     signal longitud: std_logic_vector (0 to 1);
58
59 begin
60     Generador_datos: for I in 0 to Nreg-1 generate           --LFSR de 6 bits
61         Reg00: if (I=0) generate
62             Reg0: Registros port map (clk2, reset, sig_xor, Q_int(0));
63         end generate;
64         Regs : if I>0 generate
65             Reg: Registros port map (clk2, reset, Q_int(I-1), Q_int(I));
66         end generate;

```



```

67     end generate;
68
69     Generador_datos2: for I in 0 to Nreg2-1 generate           -- LFSR de 9 bits
70         Reg000: if (I=0) generate
71             Reg00:  Registros port map (clk2, reset, sig_xor2, Q_int2(0));
72         end generate;
73         Regs2 : if I>0 generate
74             Reg2:  Registros port map (clk2, reset, Q_int2(I-1), Q_int2(I));
75         end generate;
76     end generate;
77
78     Generador_datos3: for I in 0 to Nreg3-1 generate           -- LFSR de 11 bits
79         Reg0000: if (I=0) generate
80             Reg000:  Registros port map (clk2, reset, sig_xor3, Q_int3(0));
81         end generate;
82         Regs3 : if I>0 generate
83             Reg3:  Registros port map (clk2, reset, Q_int3(I-1), Q_int3(I));
84         end generate;
85     end generate;
86
87     Generador_datos4: for I in 0 to Nreg4-1 generate           -- LFSR de 15 bits
88         Reg00000: if (I=0) generate
89             Reg0000:  Registros port map (clk2, reset, sig_xor4, Q_int4(0));
90         end generate;
91         Regs4 : if I>0 generate
92             Reg4:  Registros port map (clk2, reset, Q_int4(I-1), Q_int4(I));
93         end generate;
94     end generate;
95
96     BT1: Base_Tiempo port map (clk2 => clk2,clk3=>clk3, RST => RST,    -- Unión de Componentes
97                               clk => clk,reset=>reset,swin=>swin,
98                               S1 =>S1,S2=>S2,S3=>S3);
99
100    BTA: BaseAnti port map (clk3=>clk3, clk=>clk,reset=>reset);
101
102    debugsele: debunse sele port map (clk3=>clk3,reset=>reset,
103                                     sw2=>sw2,sw3=>sw3,sw22=>sw22,
104                                     sw33=>sw33);
105
106    datos1 <= Q_Int(Nreg-1) ;
107    datos2 <= Q_Int2(Nreg2-1);
108    datos3 <= Q_Int3(Nreg3-1);
109    datos4 <= Q_Int4(Nreg4-1);
110    sig_xor <= Q_int(1) xor Q_int(2) xor Q_int(4) xor Q_int(5);
111    sig_xor2 <= Q_int2(4) xor Q_int2(5) xor Q_int2(7) xor Q_int2(8);
112    sig_xor3 <= Q_int3(6) xor Q_int3(8) xor Q_int3(9) xor Q_int3(10);
113    sig_xor4 <= Q_int4(10) xor Q_int4(12) xor Q_int4(13) xor Q_int4(14);
114
115    sinc <= and_vector(Q_int);           --logica secuencial para selección de longitud PRBS
116    longitud <= (sw33,sw22);
117    with longitud select
118    datos <= datos1 when "11",
119            datos2 when "10",
120            datos3 when "01",
121            datos4 when "00";
122
123    process (RST,datos)
124    begin
125        if RST ='1' then
126            datos5 <= '0';
127        else
128            datos5 <=datos;
129        end if;
130    end process;
131 end Behavioral;

```

## Programa principal/ Componente Modulador PPM

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa principal para modulador PPM
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9  use work.Constantes.ALL;
10
11  entity ppm is
12      port (clk2, reset,datos: inout STD_LOGIC;      -- entrada de PRBS,reloj del FPGA y RESET
13            clk_datos,salida_ppm : out STD_LOGIC;    -- salida PPM
14            datos_par_sal:inout STD_LOGIC_VECTOR(2 downto 0));
15  end ppm;
16
17  architecture Behavioral of ppm is
18  --Señales--
19  signal datos_par,datos_serie : std_logic_vector(2 downto 0);
20  signal clk_datos_int, clk_par : std_logic;
21
22  component Registros is                                -- uso del componente registro
23  port (clk_datos,reset,D : in std_logic;
24        Q : out std_logic);
25  end component;
26  signal cuenta_ppm : natural range 0 to 7 :=0;
27
28  begin
29  Gen_relojes: process (clk2,reset)                    -- Generador de Relojes
30  variable cuenta : natural range 0 to 8:=0;
31  begin
32      if reset ='1' then
33          cuenta :=0;
34          clk_datos_int <='1';
35          clk_par <='1';
36
37      elsif clk2'event and clk2='1' then
38          cuenta := cuenta+1;
39          if cuenta = 8 then
40              cuenta:=0;
41              clk_datos_int <='1';
42              clk_par <='1';                                -- Reloj para registro paralelo
43          else
44              clk_par <='0';
45              if ((cuenta=0 ) or (cuenta=2) or (cuenta=4) or (cuenta=6) or (cuenta=8)) then
46                  clk_datos_int <='1';                    -- Reloj para registros serie
47              else
48                  clk_datos_int <='0';
49              end if;
50          end if;
51      end process Gen_relojes;
52
53  clk_datos <=clk_datos_int;
54
55  Registro_paralelo: for I in 0 to 2 generate          --
56  RegM1: if I=0 generate                                -- registros para bits entrantes
57  Reg_M1: Registros port map
58  (clk_datos_int, reset,datos,datos_serie(0));
59  end generate;
60  RegM: if I>0 generate
61  Reg_in: Registros port map
62  (clk_datos_int, reset,datos_serie(I-1), datos_serie(I));
63  end generate;
64  Reg_par: Registros port map                          -- Registro para bits paralelo
65  (clk_par,reset, datos_serie(I), datos_par(I));
66  end generate;

```

```

67
68   Gen_ppm: process (clk2, reset)          -- Contador de 0-7
69     begin
70
71       if reset ='1' then
72         cuenta_ppm <= 0;
73       elsif clk2'event and clk2='1' then
74         if cuenta_ppm = 7 then
75           cuenta_ppm <=0;
76         else
77           cuenta_ppm <= cuenta_ppm +1;
78         end if;
79       end if;
80     end process Gen_ppm;
81
82     salida_ppm <='1' when
83       (std_logic_vector (conv_unsigned (cuenta_ppm,3))= datos_par) -- Convertidor y comparador
84       else '0'; -- de entero a bit
85 datos_par_sal <= datos_par; -- salida para datos modulados en 8-PPM
86 end Behavioral;

```

## Programa que une el Generador de datos PRBS y el Modulador PPM

```

1  -- Mario A. Morales Palacios 11270484
2  -- Instituto Tecnológico de Tuxtla Gutiérrez
3  -- Departamento de Eléctrica y Electrónica
4  -- Programa principal que une el generador y el Modulador
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9  use work.Constantes.ALL;
10
11 entity sistema is
12 Port (  clk2,clk,reset : inout STD_LOGIC;
13        datos,sinc,salida_ppm : inout STD_LOGIC;
14        datos_par_sal : inout std_logic_vector(2 downto 0));
15 end sistema;
16
17 architecture Behavioral of sistema is
18
19 component BTPPM is          -- Componente Base de Tiempo
20   port (
21     clk: in std_logic;
22     reset : inout std_logic;
23     clk2: inout std_logic
24   );
25 end component;
26
27 component Pseudoaleatorios is -- Componente Pseudoaleatorios que se encarga
28   generic (Nreg      : positive := N); -- de generar datos PRBS
29   port (reset,clk_datos : inout STD_LOGIC;
30         datos,sinc : inout STD_LOGIC);
31 end component;
32
33 component ppm is
34   Port (  clk2,reset,datos : inout STD_LOGIC; -- Componente Modulador
35         clk_datos,salida_ppm : out STD_LOGIC;
36         datos_par_sal : inout std_logic_vector(2 downto 0));
37 end component ppm;
38
39 signal datos_int,clk_datos,clk_ppm : std_logic;
40
41 begin
42   Gen_datos0: Pseudoaleatorios port map (clk_datos=> clk_datos, -- Unión de Componentes
43                                         reset=>reset,datos=>datos,sinc=>sinc);
44
45   Gen_ppm: ppm port map (clk2=>clk2,reset=>reset,datos=>datos,clk_datos=>clk_datos,
46                       salida_ppm=>salida_ppm,datos_par_sal=>datos_par_sal);
47
48   BTMOD: BTPPM port map (clk=>clk,reset =>reset, clk2=>clk2);
49 end Behavioral;

```