



**DISEÑO DE UN MÓDULO DETECTOR DE ERRORES  
BASADO EN EL MÉTODO DE COMPROBACIÓN DE  
REDUNDANCIA CÍCLICA (CRC).**

**Autores:**  
Gallardo Luis  
Villarroel Gabriel

Urb. Yuma II, calle No 3. Municipio San Diego  
Teléfono: (0241) 8714240 (máster) – Fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN TELECOMUNICACIONES**

**DISEÑO DE UN MÓDULO DETECTOR DE ERRORES BASADO EN EL  
MÉTODO DE COMPROBACIÓN DE REDUNDANCIA CÍCLICA (CRC).**

**Trabajo de grado presentado como requisito para optar al título de  
INGENIERO EN TELECOMUNICACIONES**

**Autores:**

Gallardo Luis

**C.I:** V-23.649.704

Villarroel Gabriel

**C.I:** V-24.877.109

**Tutor:**

Ing. Rainier Blanco

**C.I:** V-11.556.607

San Diego, Enero de 2019



Universidad José Antonio Páez  
Facultad de Ingeniería

FI-T -002-2018-HCR

Valencia, 06 de Noviembre de 2018.

Ciudadanos:  
Luis Gallardo  
C.I:23.649.704  
Gabriel Villarroel  
C.I: 24.877.109

Presente.-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 01-2018 de fecha 06-11-2018 aprobó el proyecto de trabajo de grado titulado **DISEÑO DE UN MÓDULO DETECTOR DE ERRORES BASADO EN EL MÉTODO DE COMPROBACIÓN DE REDUNDANCIA CÍCLICA (CRC)**. Presentado por usted(es) como requisito para optar al título de Ingeniero en Telecomunicaciones.

Se ratifica la designación del Ing. Rainier Blanco, C.I: 11.556.607 y la Ing. Alicia Yáñez, C.I.: 4.598.880 como Tutores Académicos que lo asesorarán en el desarrollo de este proyecto.

Atentamente,

Prof. Zulay Salcedo  
Decana de la Facultad de Ingeniería



c. c. Coordinación de Pasantías y Trabajo de Grado (1).

ZS/fr

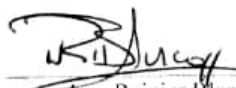


REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN TELECOMUNICACIONES

#### ACEPTACIÓN DEL TUTOR

Quien suscribe, Ingeniero **Rainier Blanco** portador de la cédula de identidad N° **11.556.607**, en mi carácter de tutor del trabajo de grado presentado por el(los) ciudadano(s) **Luis Gallardo** y **Gabriel Villarroel**, portador(es) de la cédula de identidad N° **23.649.704** y N° **24.877.109**, (respectivamente), titulado **DISEÑO DE UN MÓDULO DETECTOR DE ERRORES BASADO EN EL MÉTODO DE COMPROBACIÓN DE REDUNDANCIA CÍCLICA (CRC)**. Presentado como requisito parcial para optar al título de Ingeniero de Telecomunicaciones, acepta la tutoría del mencionado proyecto durante su etapa de desarrollo hasta su elaboración y evaluación; según las condiciones de la Coordinación de Pasantías y Trabajo de Grado de la Facultad de Ingeniería de la Universidad José Antonio Páez y sus correspondientes Reglamentos.

En San Diego, a los 11 días del mes de diciembre del año dos mil dieciocho.

  
Ing. Rainier Blanco  
C.I.: V- 11.556.607

## **AGRADECIMIENTOS**

**Luis Miguel Gallardo**

Inicialmente agradezco a Dios por permitirme lograr todo lo que hoy en día he logrado, a mi familia por apoyarme en cada etapa de mi carrera. A Dani por siempre estar allí conmigo dándome ánimos para seguir adelante. A mi abuelo Miguel por ser mi guía durante toda mi vida.

A mi compañero Gabriel por juntos lograr este trabajo, y a nuestro tutor Rainier Blanco por guiarnos y orientarnos a lo largo de este proyecto.

## **AGRADECIMIENTOS**

**Gabriel Villarroel**

Agradezco a mis padres por siempre apoyarme incondicionalmente en mis metas y sobre todo en esta que ha sido mi carrera universitaria y a toda mi familia que de una u otra forma también lo ha hecho y que hoy ven este logro como uno familiar. También agradezco a mi compañero Luis por un gran trabajo en equipo y a nuestro tutor Rainier Blanco por su gran disposición por ayudarnos a llevar a cabo este trabajo de grado.

## ÍNDICE GENERAL

<b>CONTENIDO</b>	<b>Pp</b>
<b>RESUMEN</b> .....	xi
<b>INTRODUCCIÓN</b> .....	1
 <b>CAPÍTULO</b>	
<b>I EL PROBLEMA</b>	
1.1 Planteamiento del Problema.....	3
1.2 Formulación del Problema.....	5
1.3 Objetivos de la investigación.....	6
1.3.1 Objetivo General .....	6
1.3.2 Objetivos Específicos .....	6
1.4 Justificación del Problema .....	6
1.5 Alcance .....	7
 <b>II MARCO TEÓRICO</b>	
2.1 Antecedentes.....	8
2.2 Bases teóricas.....	9
2.2.1 Transmisión de datos.....	9
2.2.2 Modos de transmisión.....	9
2.2.2.1 Transmisión paralela .....	10
2.2.2.2 Transmisión serial .....	10
2.2.3 Velocidad de transmisión.....	11
2.2.4 Técnicas de detección y corrección de errores.....	11
2.2.5 Detección de errores.....	11
2.2.6 Corrección de errores.....	12
2.2.7 Código ASCII .....	12
2.2.8 Redundancia.....	12
2.2.9 Comprobación de redundancia cíclica (CRC).....	12
2.2.9.1 Estándares CRC.....	15
2.2.9.2 Codificación de bloques.....	16
2.2.10 Lenguaje de programación.....	17
2.2.10.2 Hardware description language (HDL).....	17

2.2.10.3	VHDL.....	17
2.2.10.4	Nexys.....	18
2.3	Definición de términos básicos.....	19

### III MARCO METODOLÓGICO

3.1	Tipo de investigación.....	22
3.2	Diseño de la investigación.....	23
3.3	Nivel de la investigación.....	23
3.4	Población y muestra.....	23
3.5	Técnicas e instrumentos de recolección de datos.....	23
3.6	Fases metodológicas.....	24

### IV RESULTADOS

4.1	FASE I: Estudio la situación actual en cuanto al desempeño de los laboratorios de la materia transmisión de datos de la Universidad José Antonio Páez .....	26
4.1.1	Observación Directa .....	26
4.1.2	Observación Indirecta .....	26
4.1.3	Entrevista .....	27
4.1.4	Observación de puntos críticos .....	27
4.2	FASE II: Análisis del método de comprobación de redundancia cíclica (CRC) en la detección de errores .....	27
4.2.1	Algoritmo para la aplicación del método de comprobación de redundancia cíclica .....	28
4.2.2	Estandarización de la comprobación de redundancia cíclica .....	30
4.2.3	Capacidad para detectar errores .....	31
4.3	FASE III: Realización de un módulo detector de errores basado en el método de comprobación de redundancia cíclica .....	32
4.3.1	Registro de almacenamiento .....	32
4.3.2	Estándares usados en el módulo detector de errores .....	34
4.3.2.1	Estándar 1 .....	34
4.3.2.2	Estándar 2 .....	36
4.3.2.3	Estándar 3 .....	37
4.3.3	Registro de almacenamiento en el receptor .....	39
4.3.4	Diagrama del módulo detector de errores .....	41
4.3.4.1	Palabra dato .....	42
4.3.4.2	Selector .....	42

4.3.4.3	Estándares .....	42
4.3.4.4	Resto transmisor .....	43
4.3.4.5	Resto receptor .....	43
4.3.4.6	Indicador de errores .....	43
4.3.4.7	Letra .....	44
4.4	FASE IV: Creación de un manual de usuario para el módulo detector de errores que explique su funcionamiento y sus especificaciones .....	45
<b>CONCLUSIONES .....</b>		<b>47</b>
<b>RECOMENDACIONES .....</b>		<b>49</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>51</b>
<b>REFERENCIAS ELECTRÓNICAS .....</b>		<b>53</b>
<b>ANEXOS .....</b>		<b>54</b>
A.	Código en VHDL .....	55
B.	Manual de usuario .....	60

## ÍNDICE DE FIGURAS

<b>FIGURA</b>	<b>CONTENIDO</b>	<b>Pp</b>
1	Lista de palabras de datos con su correspondiente palabra código...	14
2	División en módulo 2 de palabra código para obtener su resto.....	15
3	Transmisión no fiable entre emisor y receptor.....	17
4	Dispositivo programable configurable Nexys 2.....	18
5	División para decodificar la palabra dato .....	29
6	Transmisión de mensaje con error .....	31
7	Cálculo del resto de la palabra correcta e incorrecta .....	32
8	Registro .....	33
9	Sumador binario .....	33
10	Registro de almacenamiento del polinomio (Estandar 1) .....	33
11	Registro de almacenamiento del polinomio (Estandar 2) .....	36
12	Registro de almacenamiento del polinomio (Estandar 3) .....	37
13	Diagrama de entradas y salidas del Módulo Detector de Errores ...	41
14	Entradas y salidas mostradas en la tarjeta Nexys 2 .....	41
15	Asignación de botones de la tarjeta Nexys 2 .....	43
16	Diagrama de los leds de un display 7 segmentos .....	44

## ÍNDICE DE TABLAS

<b>TABLA</b>	<b>CONTENIDO</b>	<b>Pp</b>
1	Principales estándares del método CRC.....	16
2	Operaciones binarias en aritmética modular .....	28
3	Ejemplo del cálculo de la palabra código .....	29
4	Ciclos del registro de almacenamiento del estándar 1 .....	36
5	Ciclos del registro de almacenamiento del estándar 2 .....	37
6	Ciclos del registro de almacenamiento del estándar 3 (3 bits más significativos) .....	38
7	Ciclos del registro de almacenamiento del estándar 3 (3 bits menos significativos) .....	38
8	Ciclos del registro de almacenamiento del estándar 1 para decodificación .....	40
9	Caracteres mostrados por el Módulo Detector de Errores .....	44



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN TELECOMUNICACIONES**

**DISEÑO DE UN MÓDULO DETECTOR DE ERRORES BASADO EN EL  
MÉTODO DE COMPROBACIÓN DE REDUNDANCIA CÍCLICA (CRC).**

**Autores:** Gallardo Luis, Villarroel Gabriel

**Tutor:** Ing. Rainier Blanco

**Fecha:** 2018

**RESUMEN**

El método de redundancia cíclica para la detección de errores es el más seguro y usado en las telecomunicaciones. Este permite disminuir considerablemente la cantidad de errores en las transmisiones de datos que se producen en los sistemas o canales de transmisión debido al ruido u otros factores. Con el diseño de un módulo detector de errores que, dada una palabra de datos, aplique las divisiones pertinentes del método en cuestión para crear la palabra código, simule un error aleatorio e indique al usuario si ha habido un error o no, se estará exponiendo el funcionamiento del método de redundancia cíclica lo que servirá como herramienta de apoyo que facilite al estudiantado de la materia “transmisión de datos” de la Universidad José Antonio Páez aprender estos conceptos; tomando en consideración que el laboratorio de dicha materia carece de instrumentos de este tipo y además, la universidad cuenta con dispositivos digitales programables que son una herramienta para la implementación del módulo detector de errores y que a los mismos actualmente no se le da ningún tipo de uso. Queda así expuesta la necesidad de desarrollar este proyecto y la oportunidad para la Universidad José Antonio Páez de proveer a sus estudiantes con mejores herramientas tecnológicas de aprendizaje sin verse en la necesidad de realizar una gran inversión monetaria.

**Descriptor:** comprobación de redundancia cíclica, herramienta educativa, dispositivo programable configurable.

## INTRODUCCIÓN

La presente investigación tiene como principal propósito de estudio el funcionamiento del método de comprobación de redundancia cíclica (CRC) y la problemática que existe en la Universidad José Antonio Páez sobre la falta de herramientas prácticas y cómo se puede mejorar la enseñanza en la materia transmisión de datos a través de la creación de un módulo detector de errores que haga uso del método en estudio.

La comprobación de redundancia cíclica es un tópico clave en el mundo de las telecomunicaciones. Los distintos estándares que existen de este método varían entre sí por la cantidad de bits redundantes que son añadidos a la información que se va a transmitir. Aunque se profundizará mucho más sobre el funcionamiento del método CRC y cómo son añadidos los bits redundantes, no está demás mencionar algunos estándares específicos que son aplicados por una gran cantidad de instituciones tecnológicas en sus sistemas de comunicaciones, como, por ejemplo, la ITU-TS usa el estándar CCITT el cual usa un polinomio de 16 bits, la IBM usa otro estándar de 16 bits llamado CRC-16 en sus protocolos SDLC, e incluso los protocolos de redes LAN usan el estándar de 32 bits CRC-32. No cabe duda que en la industria de las telecomunicaciones se ha hecho un gran esfuerzo en investigar y desarrollar distintos protocolos que reduzcan los errores que suelen ocurrir en las transmisiones.

Para analizar la problemática en la Universidad José Antonio Páez es necesario conocer sus causas principales. Durante los últimos años, debido a la alta inflación que hay en Venezuela, se le ha dificultado muchísimo a la Universidad José Antonio Páez adquirir nuevos equipos para la enseñanza práctica de las carreras de la escuela de ingeniería, viéndose especialmente afectadas las carreras de ingeniería en telecomunicaciones e ingeniería electrónica.

Como estudiantes de ingeniería debemos usar nuestro ingenio para resolver los problemas que se presenten en nuestro campo de experticia. Es por ello que este

proyecto va motivado a darle una herramienta que sirva de apoyo a las clases de transmisión de datos aplicando los conocimientos que se han adquirido durante toda la carrera y con el apoyo de la misma institución para así garantizar su factibilidad.

Primero, en el capítulo I, se hace una descripción de la problemática que motiva a este trabajo de investigación y se plantean los objetivos de la misma. Luego, en el capítulo II, establecemos las bases teóricas que fundamentan este proyecto explicando detalladamente cada aspecto necesario para entender de qué se trata. Posteriormente, en el capítulo III, se muestran los trabajos investigativos que se tomaron como antecedentes, se habla de la población y cómo, en este caso, se corresponde con la muestra, y también se enumeran las fases metodológicas que llevarán el presente proyecto a cabo. Por último, en el capítulo IV, se describen todos los pasos que se llevaron a cabo para crear el módulo detector de errores basado en el método de comprobación de redundancia cíclica y finalmente se presentan las conclusiones y recomendaciones de este proyecto.

# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1 Planteamiento del problema**

La comunicación siempre ha sido un impulsor en el desarrollo de toda sociedad. A lo largo de los años los seres humanos han tenido la necesidad de transmitir ideas, información y mensajes. Siendo esto la razón principal del porqué la necesidad de la comunicación. Con el paso del tiempo, se ha buscado la manera de facilitar la transmisión de estos mensajes, en consecuencia, la tecnología ha entrado a ser protagonista; creando instrumentos de uso cotidiano para las personas.

El mundo de la comunicación ha venido evolucionando a una velocidad abrumadora. Hoy en día existen medios como la radio, el teléfono y el internet que cada día crecen de una manera acelerada y en constante mejora de los canales de comunicación con el fin de que se puedan transmitir señales con una mayor rapidez y mayor capacidad de transmisión de datos.

El internet, no es más que una gran red de redes que, a su vez, permite la interconexión de aún más redes independientes que comparten información entre ellas a través de un protocolo a lo largo de todo el planeta. Existen varios tipos de redes que se clasifican según su extensión. La más común es la LAN (Local Area Network), la cual se emplea en el intercambio de datos y recursos entre ordenadores. Este tipo de red se usa normalmente para la interconexión de equipos que se encuentren en espacios relativamente pequeños, cuya característica principal es establecer interconexión de múltiples nodos, aunque no estén conectados físicamente al ordenador. Debido a que la LAN presenta un inconveniente en cuanto a la cantidad de nodos que pueden ser interconectados, se emplea otro tipo de red como lo es la WAN (Wide Area Network) que consiste en dos o más redes LAN conectadas entre sí. Las grandes compañías y los proveedores de servicio de Internet utilizan

frecuentemente este tipo de redes que, a pesar de su diferencia, están interconectadas entre ellas, utilizando procedimientos distribuidos entre múltiples computadoras o nodos.

No hay duda de la gran importancia que tienen las computadoras en la transmisión de datos, como Herrera (2010) declara: “Los sistemas de transmisión de datos constituyen el apoyo de los sistemas de cómputo para el transporte de la información que manejan” (p.25). Lo que quiere decir, que sin la ayuda de estos sistemas no se hubiese logrado la creación de las redes avanzadas de cómputo de procesamiento distribuido, permitiendo así compartir información y transferir datos a largas distancias, a velocidades sumamente rápidas y en grandes volúmenes; para garantizar el funcionamiento eficiente en la sociedad actual.

En el proceso de transmisión de datos digitales, siempre hay que tomar en consideración la posibilidad de pérdida de información debido a factores como: efectos electromagnéticos sobre el canal de datos, mal acoplamiento entre los terminales de los canales de datos, disipación de niveles de energía, entre otros. En el caso de que un receptor reciba información incorrecta, debe realizarse un proceso de retransmisión de la información. Este proceso de retransmisión, generalmente, no es muy apropiado debido a que genera uso adicional del canal de datos, lo cual conlleva otros inconvenientes.

Además, se puede presentar distorsión sobre las señales que llevan esa información, que dan origen a múltiples errores. Estas distorsiones en muchos casos se generan por variables externas en el medio de comunicación. Entonces, debido a que estas señales son alteradas en dicho proceso, existe la posibilidad de que la información en el receptor sea diferente a la información en el transmisor. Este problema se soluciona mediante la aplicación de diferentes algoritmos, en los cuales la clave es adicionar redundancia a la cadena de transmisión que permita analizar los datos en dicha cadena. Según Forouzan (2007) dijo que:

Las redes deben ser capaces de transferir datos desde un dispositivo a otro con una exactitud total. Para muchas aplicaciones,

un sistema debe garantizar que los datos recibidos son idénticos a los datos transmitidos. En cualquier momento los datos transmitidos desde un nodo al siguiente pueden ser corrompidos. Muchos factores, incluyendo el ruido de la línea, pueden alterar o eliminar uno o más bits de una unidad de datos determinada. Los sistemas fiables deben tener mecanismos para detectar y corregir tales errores. (p.261)

Queda claro que es inevitable realizar una transmisión de datos ideal, es por ello que se deben emplear métodos para la detección y la corrección de los errores que ocurran en la transmisión, asegurando que el mensaje transmitido llegue al receptor sin ningún tipo de error.

En la Universidad José Antonio Páez, a los estudiantes de la materia transmisión de datos, se les imparte el tema de detección y corrección de errores a través de clases magistrales y con muy poco énfasis experimental ya que los laboratorios de la materia no cuentan con equipos diseñados específicamente para demostrar el funcionamiento de los distintos métodos que existen. De entre todos los métodos de detección de errores, el de redundancia cíclica es el más seguro y usado en las telecomunicaciones como afirma Víctor Alejandro Martínez Prospero (2012): “La verificación de redundancia cíclica es un método de control de integridad de datos de fácil implementación. Es el principal método de detección de errores utilizado en las telecomunicaciones.” Este método permite disminuir considerablemente la cantidad de errores en las transmisiones de datos que se producen en los sistemas o canales de transmisión. Por tanto, debe ser estudiado a profundidad, y, no solo teórica sino también experimentalmente con equipos que no hay en los laboratorios de “transmisión de datos”. Lo que se traduce en una significativa deficiencia en la calidad educativa y un claro punto de mejora para la institución educativa.

## **1.2 Formulación del problema**

Luego de tomar en consideración lo anteriormente planteado, se ha formulado la siguiente interrogante: ¿De qué manera se puede elevar el nivel enseñanza teórico-

práctica de la materia de transmisión de datos de la Universidad José Antonio Páez con respecto al método de detección de errores llamado comprobación de redundancia cíclica (CRC)?

### **1.3 Objetivos de la investigación**

#### **1.3.1 Objetivos General**

Diseñar un módulo detector de errores basado en el método de comprobación de redundancia cíclica (CRC)

#### **1.3.2 Objetivos Específicos**

- Estudiar la situación actual en cuanto a la instrumentación de los laboratorios de la materia Transmisión de Datos de la Universidad José Antonio Páez.
- Analizar el método de comprobación de redundancia cíclica (CRC) para la detección de errores.
- Realizar un módulo detector de errores basado en el método de comprobación de redundancia cíclica.
- Crear un manual de usuario para el módulo detector de errores que explique su funcionamiento y sus especificaciones.

### **1.4 Justificación del problema**

Todo sistema de comunicaciones debe poder transferir exactamente los mismos datos de un transmisor a un receptor. Dichos datos son susceptibles a cambios indeseados debido a factores como el ruido de la línea que pueden alterar los bits del mensaje, y es de gran interés para los ingenieros en telecomunicaciones conocer los mecanismos que detectan y corrigen tales errores.

Este proyecto permite dejarle a la Universidad José Antonio Páez un gran aporte instrumental que permitirá una mejor impartición de la materia “Transmisión de Datos”. Ya que además de ser un gran detector de errores, también ayuda a explicar el método de redundancia cíclica de una manera más interactiva.

Para una institución de educación superior es preponderante estar bien equipada con instrumentos que hagan más expedito el proceso de enseñanza. Aprovisionar los laboratorios de la universidad con equipos destinados a la enseñanza trae grandes

beneficios: le hace más sencilla la instrucción de los temas al facilitador de la materia, permite que los alumnos entiendan mejor los conceptos, aumenta la competitividad no solo de los egresados sino que también de la universidad, además de también elevar su prestigio. Además de eso, que los propios estudiantes de la institución desarrollen los instrumentos de laboratorio también representa un gran ahorro para la universidad.

### **1.5 Alcance**

En este proyecto se busca realizar el diseño e implementación de un módulo que permita exponer el funcionamiento del método de comprobación de redundancia cíclica en una línea de transmisión de datos; simulando los posibles errores que puedan ocurrir en una transmisión, es decir, se modificará el mensaje que vaya a ser enviado para así introducir un error intencionalmente. Siendo esta la limitante principal del proyecto. Además, se busca demostrar el funcionamiento que tiene la detección de errores mediante el método de comprobación de redundancia cíclica CRC para los estudiantes de Ingeniería Electrónica y Telecomunicaciones de la Universidad José Antonio Páez.

## CAPÍTULO II

### MARCO TEÓRICO

En el desarrollo del presente capítulo se analizan y exponen teorías que se encuentran sujetas a la investigación; Arias, F. (2012) define el concepto de la siguiente manera “El marco teórico es la etapa del proceso de investigación en que establecemos y dejamos en claro a la teoría que ordena nuestra investigación, es decir, la teoría que estamos siguiendo como modelo de la realidad que estamos investigando”. (p.1). entendido esto, el marco teórico se adapta a una estructura investigativa para conocer de manera explícita las bases y conceptos de la investigación, permitiendo un fácil manejo del tema.

A continuación, se muestran los antecedentes, posteriormente las bases teóricas que sustentan el presente proyecto, complementado a su vez por la descripción de términos básicos para un adecuado entendimiento y mayor conocimiento del tema.

#### **2.1 Antecedentes de la Investigación**

Como inicio a la investigación se recolectó información de antecedentes similares a los fines educativos de la Universidad José Antonio Páez, como menciona Fabián Borrero (2014), en su trabajo de grado titulado **Desarrollo de un manual de prácticas para el laboratorio de transmisión de datos**, realizado en la Universidad José Antonio Páez, Valencia edo. Carabobo ya se da cuenta de que el laboratorio de transmisión de datos no cuenta con suficientes equipos para que los estudiantes tengan una experiencia más experimental de la materia. Además, también detectó un punto de mejora en las prácticas de laboratorio de la materia lo que lo llevó a desarrollar un manual de prácticas para la misma.

Siguiendo con este mismo orden de ideas, en el trabajo investigativo de Gustavo Durán, Julio F (2014), titulado **Multicodificador Reed-Solomon en software**, realizada en el Instituto Politécnico Nacional de la Ciudad de México, se desarrolló un algoritmo en el lenguaje de programación C++ del método de

corrección de datos Reed-Solomon. El mismo muestra un avance significativo en el establecimiento de nuevas generaciones de comunicaciones móviles basadas en tecnologías de Radio Definido por Software.

Por último el trabajo de investigación de Fernando Serralde (2013), en su tesis **Simular en VHDL la implementación de algoritmos FEC en dispositivos programables configurables**, realizada en la facultad de ingeniería de la Universidad Nacional Autónoma de México pone en uso el lenguaje de especificación VHDL, por tener una gran gama de aplicaciones, para implementar algoritmos FEC (Corrección de errores hacia adelante por su siglas en inglés), un tipo de algoritmo corrector de errores basado en redundancia, en distintos dispositivos programables configurables.

## **2.2 Bases teóricas**

### **2.2.1 Transmisión de Datos**

Cuando nos comunicamos, estamos compartiendo información. Esta compartición puede ser local o remota. Entre los individuos, las comunicaciones locales se producen habitualmente cara a cara, mientras que las comunicaciones remotas tienen lugar a través de la distancia. El término telecomunicación, que incluye telefonía, telegrafía y televisión, significa comunicación a distancia. La palabra datos se refiere a hechos, conceptos e instrucciones presentados en cualquier formato acordado entre las partes que crean y utilizan dichos datos.

La transmisión de datos es el intercambio de datos entre dos dispositivos a través de alguna forma de medio de transmisión, como un cable. Para que la transmisión de datos sea posible, los dispositivos de comunicación deben ser parte de un sistema de comunicación formado por hardware (equipo físico) y software (programas).

### **2.2.2 Modos de Transmisión**

Para la transmisión de datos digitales entre dispositivos, el cableado es de suma importancia para el flujo de datos debido a que dicha transmisión por un enlace se puede llevar a cabo en modo paralelo o en modo serie. En el modo paralelo se envía varios bits con cada pulso de reloj mientras que en el modo serie solamente se envía

un bit con cada pulso de reloj. Mientras que en la transmisión de datos paralela soy existe una única forma, en la transmisión de datos serial se subdivide en 3 clases: síncrona, asíncrona e isócrona.

#### **2.2.2.1 Transmisión Paralela**

Es el conjunto de transmisiones simultáneas de N cantidad de bits. En esta transmisión los bits de un carácter son enviados por su propio cable o línea de envío y un canal o línea nos sirve para indicar la temporización la cual es la que indica cuando comienza y termina el envío de cada carácter. La transmisión paralela funciona enviando simultáneamente bits a través de diferentes canales. Este tipo de transmisión se utiliza con mayor frecuencia para transmitir a distancias más cortas, pero siendo esta transmisión muy costosa.

#### **2.2.2.2 Transmisión Serial**

En la transmisión serial un bit sigue a otro, por lo que solamente se es necesario un canal de comunicación, en lugar de n canales de comunicación para transmitir datos entre dos dispositivos, este método tiene como ventaja principal sobre la transmisión paralela es que, al tener un único canal de comunicación, la transmisión serial reduce el coste de transmisión sobre la paralela. Este tipo de transmisión de datos se subdivide en 3 clases:

otras palabras, esta transmisión se envía un bit detrás de otro, sin bits de inicio/parada o intervalos, la responsabilidad de agrupar los bits es totalmente del receptor.

Transmisión isócrona, garantiza que los datos lleguen a una tasa fija, en aplicaciones donde los retardos desiguales entre tramas no son aceptables y la transmisión síncrona falla, para ello se debe sincronizarse el flujo entero de bits.

### **2.2.3 Velocidad de transmisión**

La velocidad de transmisión de datos mide el tiempo que tarda un host o un servidor en poner en la línea de transmisión el paquete de datos a enviar. El tiempo de transmisión se mide desde el instante en que se pone el primer bit en la línea hasta el último bit del paquete a transmitir. La unidad de medida en el Sistema Internacional sería en bits/segundo (b/s o también bps), o expresado en octetos o bytes (B/s) ya que así puede hacer la transmisión de datos.

### **2.2.4 Técnicas de detección y corrección de errores**

La corrección de errores es más difícil que la detección, en la cual solo se revisa si ha ocurrido un error o no. En cambio, en la corrección de errores, se necesita saber el número exacto de bits que se han dañado y, más importante, su posición dentro del mensaje. Es por ello que en las telecomunicaciones hay dos enfoques al momento de lidiar con este problema. El primero es tratar de corregir el error a pesar de lo difícil que pueda ser, este enfoque se conoce como Forward error correction (FEC). El otro enfoque es llamado Automatic repeat request (ARQ), mucho más sencillo y más usado. En este se detecta el error para luego solicitar al transmisor que vuelva a enviar la información debido a que llegó errada.

### **2.2.5 Detección de errores**

El uso de códigos detectores de errores, permite detectar las fallas de transmisión que pueden ser producidos por varios factores externos como lo son el ruido térmico, interferencia electromagnética, entre otros. Para detectar estos errores, se incluyen bits adicionales, llamados redundantes. De esta manera, el receptor puede

saber si hay un error en la transmisión y pedirle al transmisor que reenvíe la información correcta. Entre los códigos de detección más conocidos están el de comprobación de redundancia cíclica y el de Reed-Solomon.

### **2.2.6 Corrección de errores**

En la corrección de errores es el receptor el que se encarga de encontrar, mediante algún código, el código original enviado. Para esto se puede decir que es necesario añadir aún más bits redundantes en la transmisión. Algunos códigos de corrección de errores son el Hamming y el de paridad.

### **2.2.7 Código ASCII**

ASCII es una sigla para "

" (Código Standard Norteamericano para Intercambio de Información).

Es un código alfa-numérico (letras, símbolos, números y acentos) que representa los caracteres usando una escala decimal del 0 al 127. Estos números decimales son convertidos por la computadora en números binarios para ser posteriormente procesados. Fue propuesto por Robert W. Bemer para que fuese posible que las computadoras de diferentes fabricantes logaran entender los mismos códigos.

### **2.2.8 Redundancia**

El concepto central en la detección de errores es la redundancia, siendo esta la que permite detectar, es necesario enviar algunos bits extra junto con los bits de datos. Estos bits redundantes son añadidos en el emisor y eliminados en el receptor permitiendo al receptor detectar los bits con errores, asegurando una correcta transmisión.

Para una transmisión de datos es esencial garantizar la confiabilidad del mensaje, es por ello que se utilizan distintos tipos de métodos, como lo es el método por comprobación de redundancia cíclica, siendo el método más práctico en las telecomunicaciones, debido a su simplicidad y exactitud.

### **2.2.9 Comprobación de redundancia cíclica (CRC)**

Es un método de detección de errores ampliamente usado en las telecomunicaciones donde se agregan bits redundantes al final de la transmisión. Los

cálculos realizados en el método CRC se basan en el uso de un polinomio característico de grado  $n$ , el cual debe ser conocido tanto por el transmisor como por el receptor. A continuación, se explica el procedimiento tanto en el transmisor como en el receptor para hacer una transmisión aplicando el código de redundancia cíclica.

En el receptor se debe seguir los siguientes pasos:

1. Se divide el polinomio de los datos recibidos por el polinomio generador.
2. Se comprueba que el resto de la división sea igual a cero.
3. Si el resto es igual a cero, no se han producido errores en la transmisión.
4. Si el resto es distinto de cero, se han producido errores, por lo que se solicita una retransmisión.

En el transmisor se deben seguir los siguientes pasos:

1. Obtener el polinomio equivalente al mensaje.
2. Multiplicar por el grado del polinomio generador (equivale agregar tantos ceros como sea el grado del polinomio generador).
3. Dividir el polinomio del mensaje por el generador y obtener el resto.
4. Concatenar el resto al mensaje original y transmitir.

Suponiendo que la figura 1 muestra unos posibles mensajes para codificar y decodificar, vemos que la palabra de datos tiene  $m$  bits (4 para este ejemplo) y la palabra código tiene  $n$  bits (7 para este ejemplo). El tamaño de la palabra de datos aumenta en  $k$  (3 para este ejemplo). El resultado de  $n$  bits, es decir, la palabra código, se introduce en un generador para ser dividida por un número de dígitos (4 para este ejemplo). Cabe destacar que la división que se realiza es en módulo 2. Al obtener el resto de la división, se añade a la palabra de datos para crear el código.

---

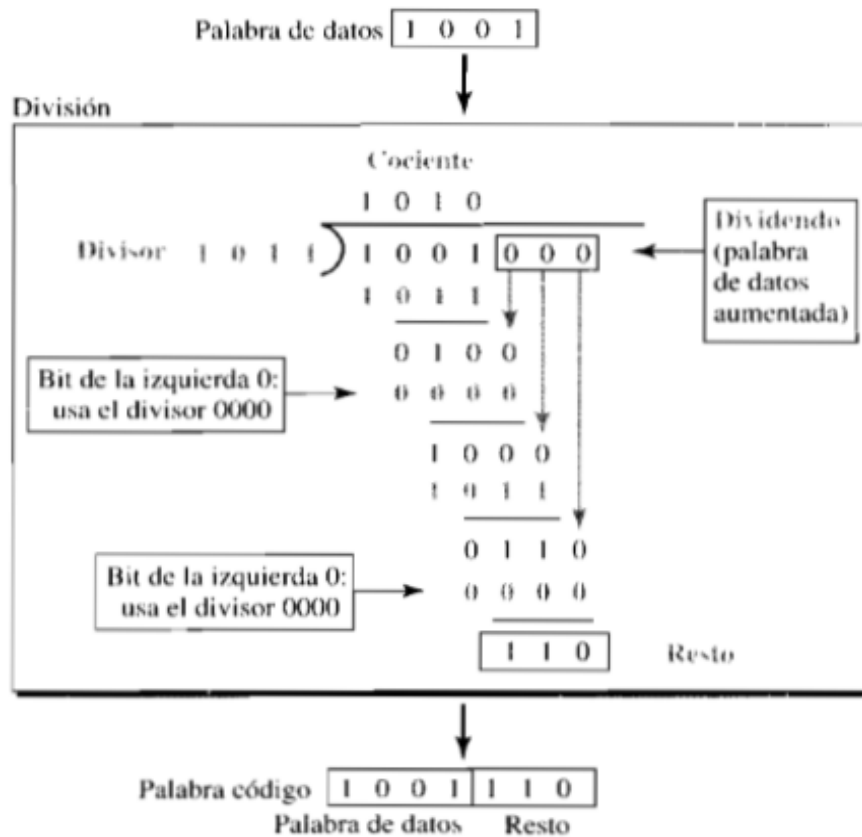
0000	<u>0000000</u>	1000	<u>1000101</u>
0001	<u>0001011</u>	1001	<u>1001110</u>
0010	<u>0010110</u>	1010	<u>1010011</u>
0011	<u>0011101</u>	1011	<u>1011000</u>
0100	<u>0100111</u>	1100	<u>1100010</u>
0101	<u>0101100</u>	1101	<u>1101001</u>
0110	<u>0110001</u>	1110	<u>1110100</u>
0111	<u>0111010</u>	1111	<u>1111111</u>

---

**Figura 1.** Lista de palabras de datos con su correspondiente palabra código

“Transmisión de datos y redes de comunicaciones”, Forouzan (2007).

El decodificador recibe la palabra código y procede a dividirla en módulo 2 por el número establecido antes de empezar las transmisiones de datos como se observa en la figura 2. Al calcular el resto de la división de tamaño (3 para este ejemplo), todos sus dígitos deben ser cero para que no haya error, en caso contrario, hubo un error en la transmisión y el receptor pide al transmisor que envíe de nuevo el mensaje.



**Figura 2.** División en módulo 2 de palabra código para obtener su resto.

Fuente. “Transmisión de datos y redes de comunicaciones”, Forouzan (2007)

### 2.2.9.1 Estándares CRC

Existe una gran cantidad de modelos en el algoritmo de la comprobación de redundancia cíclica regulados por la Unión internacional de telecomunicaciones (UIT); cada uno se diferencia del otro por tener un polinomio característico particular y estos son usados sistemas de transmisión específicos. Algunos de los estándares se muestran en la tabla 1.

**Tabla 1.** Principales estándares del método CRC.

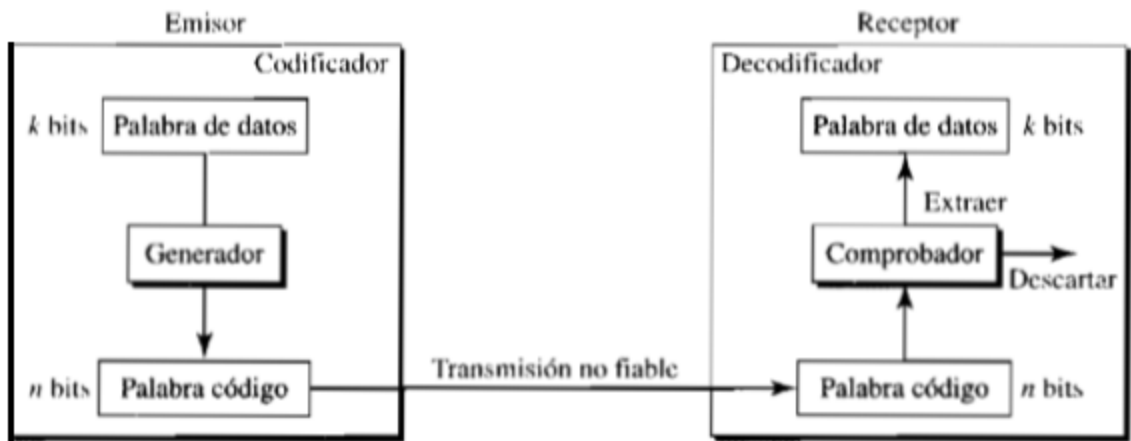
<b>Estándar</b>	<b>Uso</b>	<b>Polinomio característico</b>
CRC-12	Transmisiones de palabras de 6 bits añadiendo otros 6 bits de redundancia	
CRC-16	Transmisiones de palabras de 8 bits añadiendo otros 8 bits de redundancia	
CRC-CCITT	Transmisiones de palabras de 8 bits añadiendo otros 8 bits de redundancia	

Fuente. Gallardo y Villarroel (2018)

La fiabilidad de cada algoritmo depende del estándar (polinomio) adoptado. Para sistemas que requieren de la menor cantidad de errores que el método CRC nos permite usualmente se usa el estándar CRC-32, cuyo polinomio característico es el siguiente:

### 2.2.9.2 Codificación de bloques

Es un tipo de codificación en el que el mensaje a enviar se divide en bits, llamados palabras de datos. Posteriormente, se añaden bits a cada bloque para generar redundancia hasta conseguir la longitud de bits. Los bloques resultantes de bits son llamados palabras código. En la figura 3 se puede apreciar un esquema de este tipo de codificación.



**Figura 3.** Transmisión no fiable entre emisor y receptor

Fuente. "Transmisión de datos y redes de comunicaciones", Forouzan (2007)

### 2.2.10 Lenguaje de programación

El lenguaje de programación es en sí un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

#### 2.2.10.2 (HDL)

Lenguaje de especificación o descripción de hardware por sus siglas en inglés. Es un lenguaje formal usado en el análisis de sistemas, análisis de requerimientos y diseño de sistemas para describir un sistema de una mejor manera que un lenguaje de programación que, a diferencia del primero, se encarga de ejecutar códigos para un sistema.

#### 2.2.10.3 VHDL

El nombre de VHDL proviene de la combinación de otros dos acrónimos: VHSIC ( ) y HDL ( ). Es un lenguaje de descripción de hardware además de un estándar de la IEEE. Cuenta con las ventajas de disponibilidad pública, independencia de

dispositivos y fabricantes y es de diseño jerárquico. Es el lenguaje con el que trabaja el dispositivo programable configurable Nexys 2.

#### 2.2.10.4 Nexys 2

La tarjeta Nexys 2 es un dispositivo programable configurable basado en FPGA Xilinx Spartan 3E (ver figura 4) creado por la empresa americana de dispositivos electrónicos con fines educativos Digilent Inc. Este dispositivo cuenta con un puerto USB 2.0 de alta velocidad, 16 MB de memoria RAM y ROM y varios puertos de entrada y salida lo que le permite ser usado, junto a un computador, como una estación de diseño instrumental portable.



**Figura 4.** Dispositivo programable configurable Nexys 2.

**Fuente.** Diligent Inc. (s/f). Nexys 2 [imagen] Recuperado de:  
[https://reference.digilentinc.com/\\_media/reference/  
programmable-logic/nexys-2/nexys2-0.png](https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-2/nexys2-0.png)

### 2.3 Definición de términos básicos

**Ancho de Banda:** Es la cantidad de datos que se pueden transmitir en una cantidad de tiempo determinada.

**Aritmética de módulo 2:** Es la aritmética usada en los números binarios donde no se toman en cuenta las unidades que se deben llevar al siguiente nivel.

**(ARQ):** Solicitud de repetición automática. Son procedimientos que le requieren al transmisor que vuelva a enviar porciones de información donde se ha detectado que ha ocurrido un error.

**Binario:** Es el sistema de numeración que se caracteriza por los unos y los ceros (1=activado, 0=desactivado).

**Bit:** Es un dígito binario, que toma un valor de 0 ó 1. Son unidades de comunicación y almacenamiento de información en computación.

**Bit más significativo:** Es la posición de bit en un número binario que tiene el mayor valor. A veces se refiere al bit que se encuentra más a la izquierda.

**Capacidad de canal:** Es la velocidad en la que se transmiten los datos en un canal de comunicación. Se expresa en bits por segundo (bps) y depende del ancho de banda, el ruido y la tasa de error binario (BER) permitida.

**Codificación de códigos de línea:** Es el proceso de convertir datos digitales en señales digitales, asumiendo que los datos, en forma de texto, números, imágenes gráficas, audio o voz, se almacenen en la memoria de la computadora en secuencias de bits. Esta codificación convierte una secuencia de bits a una señal digital.

**Códigos de línea:** Son los códigos por los cuales se convierten datos digitales en una señal digital. Suponiendo que estos datos se almacenan en las memorias en forma de secuencias de bits.

**Código binario:** Es un método de representación que usa combinaciones de los números 0 y 1 para producir diferentes números binarios. La mayoría de los dispositivos electrónicos lo emplea debido a que facilita la implementación de la electrónica digital y el álgebra booleana.

**Códigos cíclicos:** Son códigos de bloques lineales que al desplazarse cíclicamente el resultado es otra palabra del código. Por ejemplo, si 1011000 es un código y rota hacia la derecha, entonces 0101100 también es una palabra del código.

**Compuertas lógicas:** Son pequeños circuitos electrónicos diseñados para realizar diversas operaciones lógicas binarias, es decir, operaciones cuyos resultados son unos y ceros. Representan la base de la circuitería digital.

**(FPGA):** Dispositivos programables configurables por sus siglas en inglés, son chips de silicio reprogramables que contiene bloques de lógica cuya interconexión puede ser modificada acorde a las necesidades del usuario mediante un lenguaje de especificación o descripción especializado.

**(FEC):** Corrección de errores hacia adelante por sus siglas en inglés. Es el proceso por el cual el receptor intenta adivinar el mensaje utilizando los bits redundantes, siendo esto posible solo si el número de errores es pequeño.

**Full-duplex:** Es la comunicación que permite al receptor la recepción y transmisión en forma simultánea.

**Half-duplex:** Es la comunicación que sólo permite que una estación reciba mientras la otra transmite.

**Línea de transmisión:** Una guía de ondas es un dispositivo que se usa para transportar eficientemente energía electromagnética y/o información de un equipo de transmisión a otro. Sus parámetros más importantes son su impedancia característica y forma geométrica.

**Multiplexación:** Es el proceso en el que se combinan múltiples corrientes de datos digitales en una señal.

**Octeto:** Es el grupo de 8 bits binarios.

**Protocolo:** Es el conjunto de reglas que rigen la comunicación.

**Red:** Son múltiples computadoras conectadas entre ellas que utilizan un sistema de comunicaciones. Su objetivo es que las computadoras se comuniquen y compartan archivos.

**Red de área extensa (WAN):** Es una Red que abarca un área geográfica más amplia que una red de área local (LAN) sobre redes de comunicaciones públicas.

**Red de área local (LAN):** Hace referencia a una red local o un grupo de redes locales interconectadas que están bajo el mismo control administrativo.

**Red de área metropolitana (MAN):** Es una red que abarca una ciudad, está compuesta por diversos edificios interconectados mediante backbones inalámbricos o de fibra óptica.

**Red de datos:** Es un conjunto de equipos informáticos, software y una variedad de dispositivos electrónicos conectados por un medio físico o inalámbrico por donde se comparte información.

**Simplex:** es un tipo de comunicación que toma lugar únicamente desde el emisor hacia el receptor y no de manera inversa (ej: televisión y radio).

**Tasa de error binario (BER):** Es el número de bits erróneos recibidos respecto al total de bits enviados durante un intervalo especificado de tiempo.

**Transmisión digital:** Para una red de computadoras diseñadas para enviar información desde un punto a otro es necesaria que la señal a enviar sea digitalizada para luego ser transferida por un canal de comunicación en forma de señal eléctrica.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

Este capítulo comprende todo lo referente al tipo, nivel y diseño de la investigación, población y muestra, técnicas e instrumentos de recolección de datos y la validez del instrumento, referente a esto Arias, F. (2012) agrega que: “Un conjunto de pasos, técnicas y procedimientos que se emplean para formular y resolver problemas, este método se basa en la formulación de hipótesis las cuales pueden ser confirmadas o descartadas por medios de investigaciones relacionadas al problema”. (p. 110). Tomando en cuenta esto, se dará a conocer toda la parte metodológica que implica el problema planteado en cuanto la realización de un módulo detector de errores basado en el método de verificación de redundancia cíclica CRC como herramienta de apoyo que permita mejorar la enseñanza de dicho método y equipar el laboratorio de la materia transmisión de datos.

#### **3.1 Tipo de investigación**

De acuerdo con el propósito de la investigación, esta es aplicada y a su vez es un proyecto factible, como dice Labrador y otros (2002) un proyecto factible “consiste en la investigación, elaboración y desarrollo de una propuesta, de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos.” (p. 14).

Al diseñar un módulo detector de errores con un fin educativo se está desarrollando una propuesta viable que efectivamente va a suplir la necesidad que tienen los laboratorios de transmisión de datos de la Universidad José Antonio Páez en cuanto a equipos tecnológicos con los que complementar la información dada en el aula de clases. Quedando así la investigación dentro de los parámetros de un proyecto factible.

### **3.2 Diseño de la investigación**

Según Arias (2012), la investigación de campo “consiste en la recolección de datos directamente de la realidad donde ocurren los hechos, sin manipular o controlar variable alguna”. (p. 94). Por tanto, se corresponde con el diseño de esta investigación cuyo lugar de desarrollo es el laboratorio de transmisión de datos de la Universidad José Antonio Páez de donde los datos serán tomados directamente de la población que será definida más adelante.

### **3.3 Nivel de la investigación**

Hernández, Fernández y Baptista (2010) nos dicen que “los estudios descriptivos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis.” (p. 80). Al estar desarrollando un módulo detector de errores buscamos describir las propiedades, el funcionamiento, ventajas y desventajas del método de comprobación de redundancia cíclica.

### **3.4 Técnicas e instrumentos de recolección de datos**

Méndez (2011) define que “las fuentes y técnicas para recolección de la información como los hechos o documentos a los que acude el investigador y que le permiten tener información” (p.143). De la misma manera Arias (2012), menciona que “las técnicas de recolección de datos son las distintas formas de obtener información” (p.53).

Las técnicas de recolección de datos que fueron utilizadas en la presente investigación son la observación directa y la entrevista. Se utilizará para este fin un cuaderno de notas, que de acuerdo a Cerdá (1991) es una forma de anotar todo lo observado y lo define como “el conjunto de informaciones, datos, expresiones, opiniones hechos, croquis, etc., que pueden constituirse en una valiosa información para la investigación.” (p. 250).

### **3.5 Población y muestra**

Es preciso conocer en primera instancia el término población, Arias, F. (2012) lo explica de esta manera: “La población o en términos más precisos población

objetivo, es un conjunto finito o infinito de elementos con características comunes para las cuales serán extensivas las conclusiones de la investigación”. (p. 81). En este caso, que se cuenta con un tipo de población finita lo cual determina un estudio limitado, donde dicha población es presentado por los estudiantes del octavo semestre en adelante de las Escuela de Ingeniería en Telecomunicaciones y Electrónica de la Universidad José Antonio Páez.

En relación a la población, se define la muestra como un conjunto representativo de carácter limitado que se extrae de la población, según lo argumentado por Wigodski (2010): “Se emplean muestras grandes y algunas veces se recomienda seleccionar de un 30% a un 50% por ciento de la población accesible”. (p. 11). Considerando lo explicado anteriormente, se tomará una población de muestra entre un 30 y un 50 por ciento de estudiantes de la población mencionados anteriormente para representar los resultados, siendo este porcentaje los estudiantes de la materia de Transmisión de Datos de la Universidad José Antonio Páez.

### **3.6 Fases metodológicas**

#### **Fase I: Estudio la situación actual en cuanto a la instrumentación de los laboratorios de la materia Transmisión de Datos de la Universidad José Antonio Páez.**

En esta fase se pondrá en uso todo lo relacionado a la observación directa, la observación de puntos críticos, de las posibles fallas y carencias que tenga el laboratorio de la materia Transmisión de Datos de la Universidad José Antonio Páez. Además, el uso de la entrevista con el facilitador de la materia, para obtener información sobre las deficiencias que puedan existir en dicho laboratorio.

Todo esto con el fin de que, mediante el uso del Módulo Detector de Errores, buscarle una solución efectiva a dicha problemática en la materia de Transmisión de Datos.

#### **Fase II: Análisis del método de comprobación de redundancia cíclica (CRC) en la detección de errores.**

Para esta fase se investigarán e identificarán los fundamentos teóricos, estándares de operación, ventajas y desventajas, técnicas y protocolos por los cuales se rige el método de comprobación de redundancia cíclica. Tal investigación se apoyará en las múltiples bibliografías que existen sobre la materia, en los recursos disponibles de la IEEE y la IUT y en profesores expertos en el tema.

Posteriormente se filtrará toda la información para resumir en los puntos más importantes sobre el método de detección de errores en estudio que serán tomados en cuenta al momento de desarrollar el programa y el módulo detector de errores de manera que sean expuestos en las prácticas de laboratorio de la materia transmisión de datos.

### **Fase III: Realización de un módulo detector de errores basado en el método de comprobación de redundancia cíclica.**

Se usará el dispositivo programable configurable Nexys II ya que la Universidad José Antonio Páez cuenta con varios de estos dispositivos, siendo una clara ventaja económica para esta institución al momento de proveer al laboratorio de transmisión de datos de varios módulos tomando en cuenta la cantidad de alumnos que cursan la materia cada semestre.

La Nexys II trabaja con el lenguaje de descripción VHDL, por lo tanto se investigarán y estudiarán los métodos existentes para realización del código VHDL que pueda ser usado por el dispositivo programable configurable.

### **Fase IV: Creación de un manual de usuario para el módulo detector de errores que explique su funcionamiento y sus especificaciones.**

Luego de haber creado el módulo detector de errores, se creará un manual de usuario que le permita tanto el facilitador de la materia transmisión de datos como a sus estudiantes comprender cómo funciona el módulo y cuáles son sus especificaciones. En el manual se hará énfasis en el hardware y como este hace uso del método CRC y se tocará en menor medida los temas referentes a la programación en VHDL dependiendo de qué tan necesarios sean para el uso a cabalidad del equipo.

## **CAPÍTULO IV**

### **RESULTADOS**

#### **4.1 Fase I: Estudio la situación actual en cuanto al desempeño de los laboratorios de la materia Transmisión de Datos de la Universidad José Antonio Páez.**

Una vez estudiada la situación actual mediante distintos métodos como lo son la observación directa, indirecta y el análisis de los puntos críticos, además de realizar varias entrevistas con el facilitador de la materia para comprender las carencias que tienen los laboratorios de la materia de Transmisión de Datos de la Universidad José Antonio Páez, logramos determinar los aspectos más importantes que desarrollamos en el proyecto.

##### **4.1.1 Observación Directa.**

Primeramente, se observó el entorno en el cual esta se desenvuelve la problemática, para obtener información relevante para el funcionamiento del sistema, la observación se realizó de forma directa, es decir, siendo cursantes de la materia en pudimos observar y tener en cuenta el funcionamiento del mismo. Se seleccionó este tipo de observación ya que la situación planteada en este proyecto, es la falta de material práctico para los laboratorios de la materia de Transmisión de Datos de la Universidad José Antonio Páez

##### **4.1.2 Observación Indirecta.**

Tomando en cuenta la muestra, la cual fue los estudiantes de la materia de Transmisión Datos de la Universidad José Antonio Páez se empleó el uso de otro método de investigación, que fue la observación indirecta, siendo una gran herramienta para el desarrollo del proyecto, porque nos permitió observar el entorno en el cual se desenvuelven.

Se consultó con varios alumnos que ya cursaron la materia, si el desarrollo del mismo sin herramientas prácticas de apoyo, es decir, sin equipos en los laboratorios fue de fácil comprensión, todos nos dieron la misma respuesta, “Logramos entender

la materia, pero en la detección y corrección de errores quedaron dudas”. Se les dio a conocer el proyecto en el cual se está trabajando, y del mismo modo todos afirmaron que con el material práctico se iba a facilitar tanto al profesor de la materia a la hora de explicar el tema de detección de errores, como a los alumnos a la hora de comprender el mismo.

#### **4.1.3 Entrevista**

Conocemos una entrevista como un intercambio de ideas, opiniones mediante una conversación establecida entre dos personas o un grupo de personas, donde en este caso el entrevistado fue el profesor de la materia de Transmisión Datos de la Universidad José Antonio Páez, el cual nos dio a conocer la falta de material práctico que tenía la materia, y la falta que este presentaba a la hora de desarrollar los temas de la materia, especialmente en la detección y corrección de errores.

Esta técnica fue de gran ayuda a la hora de determinar cuáles son los puntos críticos en los cuales nos enfocamos a la hora de este proyecto.

#### **4.1.4 Observación de puntos críticos**

Gracias a la entrevista y los métodos de observación mencionados anteriormente se lograron ver los puntos críticos que tiene la materia de Transmisión Datos de la Universidad José Antonio Páez, entre ellos destacamos los siguientes:

1. Falta de material práctico en los laboratorios, para familiarizarse con los dispositivos que se usan en el ámbito laboral de un ingeniero en telecomunicaciones.
2. Dar a conocer la importancia que tiene el método de redundancia cíclica para las telecomunicaciones, con un dispositivo que logre resaltar el objetivo.

### **4.2 Fase II: Análisis del método de comprobación de redundancia cíclica (CRC) en la detección de errores.**

En esta fase se investigó gran parte de la teoría referente a la detección de errores por el método de comprobación de redundancia cíclica, definiciones, comprobaciones y deducciones matemáticas, métodos de cálculo y capacidad de

detección de errores. Tal información fue indispensable para el diseño y programación del Módulo Detector de Errores.

#### 4.2.1 Algoritmo para la aplicación del método de comprobación de redundancia cíclica

Para crear el módulo detector de errores basado en el método CRC se tomó como punto de partida de la segunda fase de la investigación, el algoritmo enseñado en clases de Transmisión de datos. Comprender este algoritmo fue parte elemental para comprender el funcionamiento de este método.

Como ya se habló en capítulos anteriores, el método CRC está basado en una división y se puede representar con la siguiente expresión matemática:

$$\frac{\text{Palabra dato}}{\text{Polinomio característico}} = \text{Cociente de la división} + \frac{\text{Resto}}{\text{Polinomio característico}}$$

: Palabra dato

: Palabra dato extendida

: Polinomio característico

: Cociente de la división

: Resto

Si se multiplica la expresión por  $\text{Polinomio característico}$  y se resta  $\text{Resto}$  a ambos lados de la igualdad nos queda:

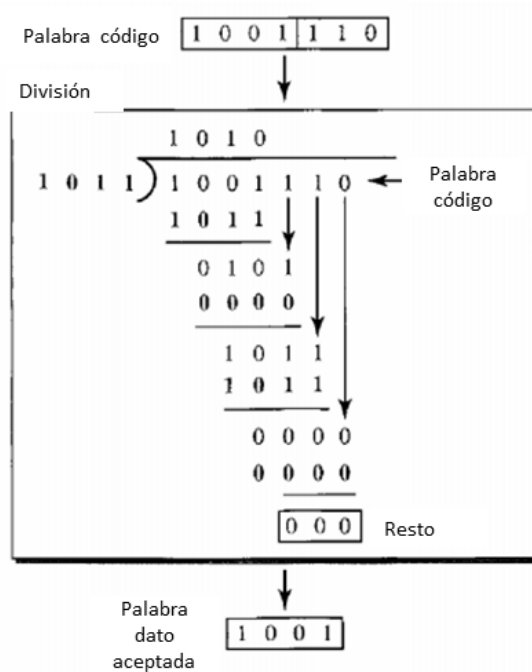
Tal expresión es equivalente a  $\text{Palabra dato extendida} = \text{Cociente de la división} \times \text{Polinomio característico} + \text{Resto}$  debido a que la suma y resta de bit a bit tiene los mismos resultados en la aritmética de módulo 2 como se ve en la tabla 2.

**Tabla 2.** Operaciones binarias en aritmética modular

Suma	Resta	Xor
0 + 0 = 0	0 - 0 = 0	0 xor 0 = 0
0 + 1 = 1	0 - 1 = 1	0 xor 1 = 1
1 + 0 = 1	1 - 0 = 1	1 xor 0 = 1
1 + 1 = 0	1 - 1 = 0	1 xor 1 = 0

Fuente. Gallardo y Villarroel (2018)

El lado izquierdo de la expresión representa la  
 palabra código que codifica el transmisor para enviárselo al receptor. En la figura  
 podemos ver una palabra código siendo decodificada y dando como resultado una  
 palabra correcta.



**Figura 5.** División para decodificar la palabra dato

**Fuente.** “Transmisión de datos y redes de comunicaciones”, Forouzan (2007)

En la tabla 3 vemos cómo, efectivamente, se calcula la palabra código con las expresiones ya presentadas anteriormente.

**Tabla 3.** Ejemplo del cálculo de la palabra código

	1 0 0 1 0 0 0
	0 0 0 0 1 1 0
	1 0 0 1 1 1 0

**Fuente.** Gallardo y Villarroel (2018)

El algoritmo anterior, a pesar de ser útil para entender cómo trabaja el método CRC, fue realmente complicado al momento de crear un código en lenguaje VHDL que replique sus procedimientos. Es por ello que se empleó un circuito digital secuencial llamado “registro de almacenamiento” ya que este emplea una estructura algebraica que hace más sencilla la creación de un circuito digital secuencial detector de errores fácilmente descriptible por el lenguaje VHDL.

#### **4.2.2 Estandarización de la Comprobación de Redundancia Cíclica**

Ya que es imprescindible, como ingenieros, conocer los estándares y las prácticas recomendadas en cada uno de los proyectos de ingeniería en el que estemos involucrados, se estudiaron las distintas instituciones que se encargan de crear manuales y estándares para el mundo de las telecomunicaciones. De esta manera nos aseguramos de incluir información relevante en el momento de crear el Módulo Detector de Errores.

En cuanto a estándares, cada uno de ellos consiste en un polinomio característico distinto y deben estudiarse a fondo debido a que es la parte más importante en la implementación del método CRC. El polinomio debe ser elegido de manera que maximice su capacidad de detectar errores mientras que minimice las probabilidades de colisión. La característica más importante de un polinomio es su longitud porque de ella depende también la longitud del resto de la operación.

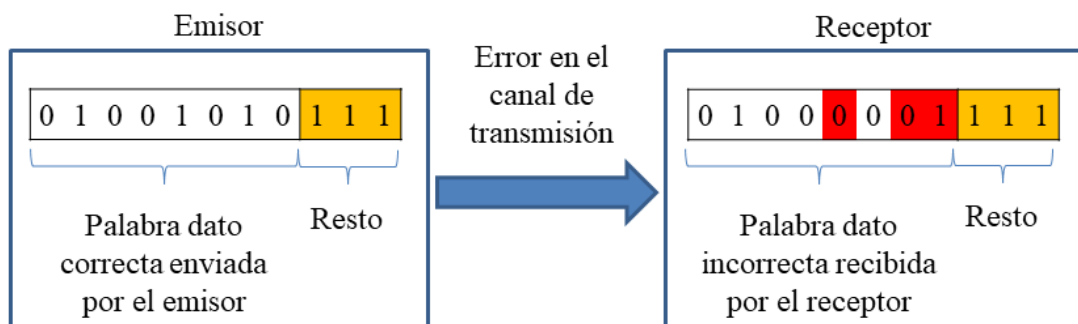
Una gran variedad de estos polinomios han sido incorporados a hojas de recomendaciones y estándares técnicos por varias instituciones ya que un solo polinomio no puede suplir las necesidades de todos los sistemas de comunicaciones. Entre las organizaciones más importantes tenemos a la división de estandarización de la UIT conocida como UIT-T la cual cuenta con una amplia variedad de manuales donde indica cuál polinomio es más efectivo para cada sistema de telecomunicaciones. Además, también está la IEEE (Instituto de Ingeniería Eléctrica y Electrónica) y la ETSI (Instituto Europeo de Normas de Telecomunicaciones) que cumplen con el mismo rol de estandarización de recomendación que el UIT-T pero

cuyas esferas de influencia se encuentran en América del Norte y Europa respectivamente.

### 4.2.3 Capacidad para detectar errores

La capacidad de detección de errores del método CRC depende del polinomio característico que se use. El polinomio error, que denotamos  $E(x)$ , es la diferencia simétrica entre la palabra código recibida en el receptor y la palabra código correcta. Ningún método de detección de errores es infalible, y en cuanto al método CRC, un error no podrá ser detectado si y solo si tal polinomio error es divisible por el polinomio característico.

Para ejemplificar un caso donde el método CRC no puede detectar un error podemos tomar un sistema que utilice el polinomio característico  $x^3 + x + 1$ , cuya representación binaria sería 1011. En la figura 6 representamos una transmisión con errores donde la palabra dato que queremos enviar es 01001010, luego de hacer los cálculos nos queda que la palabra código a enviar sería 01001010111, pero si en el trayecto la palabra dato sufre modificaciones que la cambian, por ejemplo, a 01000001111, el receptor no podría detectar que hubo un error ya que la palabra 01000001111 al igual que 01001010111 (palabra dato original) son múltiplos del polinomio característico como queda demostrado en la figura 7.



**Figura 6.** Transmisión de mensaje con error

Fuente. Gallardo y Villarroel (2018)

0 1 0 0 0 0 0 1 1 1 1   1011	0 1 0 0 1 0 1 0 1 1 1   1011
0 0 0 0	0 0 0 0
1 0 0 0	1 0 0 1
1 0 1 1	1 0 1 1
0 1 1 0	0 1 0 0
0 0 0 0	0 0 0 0
1 1 0 0	1 0 0 1
1 0 1 1	1 0 1 1
1 1 1 1	0 1 0 0
1 0 1 1	0 0 0 0
1 0 0 1	1 0 0 1
1 0 1 1	1 0 1 1
0 1 0 1	0 1 0 1
0 0 0 0	0 0 0 0
1 0 1 1	1 0 1 1
1 0 1 1	1 0 1 1
0 0 0	0 0 0

**Figura 7.** Cálculo del resto de la palabra correcta e incorrecta

**Fuente.** Gallardo y Villarroel (2018)

Conocer ante cuáles errores el método CRC no es muy efectivo es sumamente importante para el ingeniero a cargo de la creación de un sistema detector de errores. Es por ello que se analizaron los puntos débiles del método CRC para así, en la última fase de este proyecto, la creación del manual del módulo detector de errores, incluir este y otros puntos que harán más completa la enseñanza en el aula de clase y en sus laboratorios.

### 4.3 Fase III: Realización de un módulo detector de errores basado en el método de comprobación de redundancia cíclica.

Para la fase III se estudió a fondo los registros de almacenamiento, un tipo de circuito digital que se comprobó sumamente efectivo para el proceso de codificación y decodificación de la comprobación de redundancia cíclica. Se definieron las entradas y salidas, y se creó el código para programar la tarjeta Nexys 2 en el lenguaje VHDL.

#### 4.3.1 Registro de almacenamiento

Una de las razones por las que los códigos cíclicos son tan útiles es porque pueden ser codificados y decodificados eficientemente por un registro de

almacenamiento. El proceso de codificación es más eficiente porque ningún almacenamiento es requerido ya que las palabras código son generadas mientras van desplazándose bit a bit.

Los registros de almacenamiento tienen una lógica secuencial, es decir, sus cálculos no dependen sólo de valores presentes sino de valores pasados también. Dichos circuitos hacen uso de los siguientes componentes:

**1.- Registros:** Elementos capaces de contener un bit. Solo cuentan con una sola entrada y una sola salida como se ve en la figura 8. La salida de cada registro es siempre la misma que la entrada que fue una unidad de tiempo anterior. Circuitualmente, las compuertas flip flops pueden replicar este elemento.

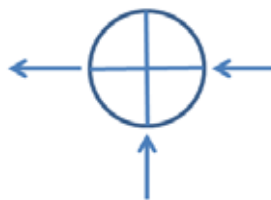


**Figura 8.** Registro

Fuente. Gallardo y Villarroel (2018)

**2.- Señal de reloj o de cambio:** Señal que controla el movimiento y desplazamiento de los bits contenidos en los registros. Luego de cada ciclo de la señal de cambio, la salida de cada etapa toma el valor que la entrada tomó inmediatamente antes de que el nuevo ciclo de la señal iniciara.

**3.- Sumadores binarios:** Estos sumadores son compuertas y tienen dos entradas y una salida como se muestra en la figura 9. La salida de estos sumadores es 1 solo si un número impar de entradas son 1.



**Figura 9.** Sumador binario

Fuente. Gallardo y Villarroel (2018)

Para conseguir una expresión general que nos permitiera calcular el resto de cualquiera palabra de 8 bits a través del método CRC se partió de la palabra genérica de 8 bits. Cada bit de se representa de la siguiente manera: donde denota la posición de cada bit. Dependiendo del tamaño del polinomio característico, se le agrega cierta cantidad de ceros “0” al final de la palabra dato. Si el polinomio tiene bits entonces se agregan bits a la palabra dato. Otra manera de saber cuántos ceros hay que poner al final de la palabra dato es que éstos serán iguales al grado del polinomio característico.

En el Módulo Detector de errores se incluyeron varios estándares los cuales se explican en el siguiente punto cómo se obtuvieron cada una de las expresiones generales para el proceso de codificación que ocurre en el transmisor.

#### **4.3.2 Estándares usados en el Módulo Detector de Errores**

En el Módulo Detector de Errores se incluyeron 3 estándares CRC distintos. Lo que quiere decir que cuenta con 3 distintos polinomios característicos para codificar las palabras datos que se le introduzcan. El principal criterio que se tuvo para elegir estos estándares fue la cantidad de 8 leds con los que cuenta la tarjeta Nexys 2, a partir de esa limitante, el otro criterio importante fue la longitud de cada polinomio; se tomó uno de pequeña longitud y los otros de mediana y larga longitud respecto al de pequeña longitud. Así se logró que el Módulo Detector de Errores tuviera diversidad de estándares y que cada resultado pueda ser mostrado efectivamente a través de sus leds y displays.

##### **4.3.2.1 Estándar 1**

Polinomio característico:

Representación binaria: 1011

Para el estándar 1 se usó la palabra dato genérica siguiente:

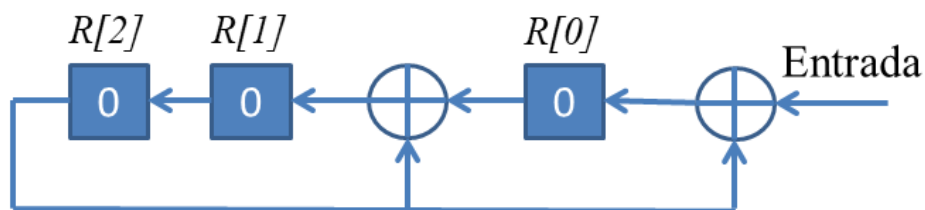
Al ser de grado 3 el polinomio característico se le agregaron tres ceros “0” al final de . El registro de almacenamiento correspondiente a este polinomio característico se muestra en la figura 10. Cada registro representa a uno de los bits del resto . En la tabla 4 se pueden ver todos los ciclos (representados con la letra “C”) hasta llegar a la expresión binaria con la que se calcula cada bit del resto. Cabe mencionar de nuevo que la operación que se aplica entre los bits es la del exclusivo o . También es importante resaltar que para facilitar los cálculos y conseguir expresiones más sencillas se aplicaron las siguientes propiedades de la operación :

1.-

2.-

Como se puede ver en la tabla 4, en un principio todos los registros son cero, luego, se empieza a introducir bit por bit la palabra dato genérica . En el primer ciclo, se realiza la operación entre el bit y el que se encuentra en el registro y el resultado queda registrado en . En el siguiente ciclo, entra al registro el bit y se van corriendo los valores de los registros siguientes y así sucesivamente.

El mismo proceso se aplica para los dos estándares restantes, siendo lo único que cambia la forma de los registros de almacenamiento que dependen del polinomio característico.



**Figura 10.** Registro de almacenamiento del polinomio (estándar 1)

**Fuente.** Gallardo y Villarroel (2018)







Las expresiones de los bits del resto del estándar 3 son las siguientes:

#### **4.3.3 Registro de almacenamiento en el receptor**

Como ya se mencionó, el proceso mostrado anteriormente para cada uno de los estándares CRC usados en el Módulo Detector de Errores es el realizado por el transmisor para codificar los mensajes que envía; sin embargo, el proceso de decodificación realizado en el receptor es exactamente el mismo. Se usan los mismos registros de almacenamiento, salvo que en vez de añadir una cantidad de ceros dependiendo del grado del polinomio característico, se añade el resto calculado por el transmisor.

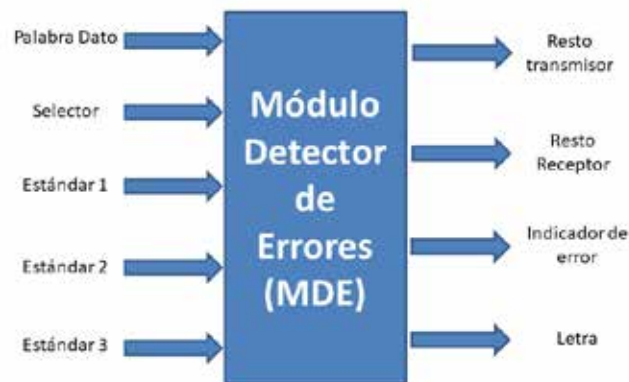
Para calcular la expresión genérica que nos permitirá decodificar la palabra código se agregan bits genéricos del resto que fue añadido a la palabra en el transmisor quedando de la siguiente manera para el caso del estándar 1:

En la tabla 8 se puede ver que para cada ciclo las expresiones son bastante parecidas a cuando se codificó la palabra. Finalmente, para calcular la expresión genérica para la decodificación solo es necesario hacer una operación entre el resto calculado en el transmisor y el resto calculado en el receptor. De ser correcta la transmisión el resultado será cero.



#### 4.3.4 Diagrama del Módulo Detector de Errores

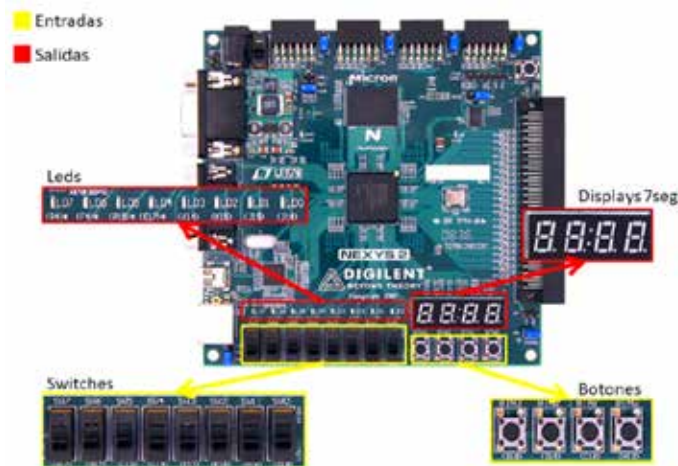
Durante el diseño del Módulo Detector de Errores se buscó aprovechar de la mejor manera todas las entradas y salidas con las que cuenta la tarjeta Nexys 2 de manera de que se pueda introducir la palabra dato y sea mostrado a través de leds y displays información útil para que el usuario pueda comprender cómo funciona este método. En la figura 13 se puede ver el diagrama que representa las entradas y salidas con las que cuenta el Módulo Detector de errores.



**Figura 13.** Diagrama de entradas y salidas del Módulo Detector de Errores

Fuente. Gallardo y Villarroel (2018)

A continuación, en la figura 14, podemos ver la tarjeta Nexys donde se resalta en amarillo las entradas y en rojo las salidas.



**Figura 14.** Entradas y salidas mostradas en la tarjeta Nexys 2

Fuente. Gallardo y Villarroel (2018)

#### **4.3.4.1 Palabra dato**

Es el vector de números binarios que se desea enviar por un sistema de comunicaciones. En el Módulo Detector de Errores se introduce a través de los switches de la tarjeta Nexys 2. El tamaño de la palabra dato es de 8 bits (un bit por cada switch).

#### **4.3.4.2 Selector**

La entrada “selector” es un botón que guarda la palabra que en ese momento se encuentre en los switches como la palabra dato correcta. Cada vez que se pulse el botón selector se guardará una nueva palabra dato “correcta”. Solo se guarda una palabra dato “correcta” a la vez.

#### **4.3.4.3 Estándares**

Los 3 estándares de CRC programados en la tarjeta Nexys fueron asignados a tres de sus cuatro botones. Los polinomios característicos de cada estándar son los siguientes:

##### **Estándar 1**

Nombre: CRC 3

Uso: GSM Mobile Networks

Polinomio característico:

Representación binaria: 1011

##### **Estándar 2**

Nombre: CRC 4

Uso: ITU-T G.704

Polinomio característico:

Representación binaria: 10011

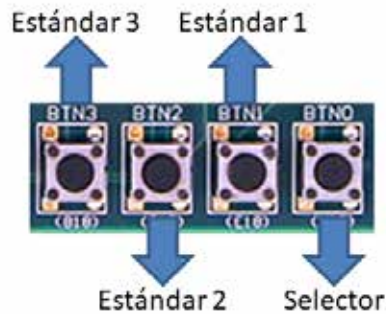
##### **Estándar 3**

Nombre: CRC 6

Uso: GSM Mobile Networks

Polinomio característico:

Representación binaria: 1101111



**Figura 15.** Asignación de botones de la tarjeta Nexys 2

Fuente. Gallardo y Villarroel (2018)

#### 4.3.4.4 Resto transmisor

Esta variable es el polinomio resto que se calcula en el transmisor para codificar la palabra dato. Esta variable se muestra en los leds de la tarjeta Nexys 2 y para ser mostrada se deben presionar simultáneamente el botón selector y el botón del estándar de cual se desee conocer el resto.

#### 4.3.4.5 Resto receptor

A diferencia del resto transmisor, este es el resto que, luego de decodificar la palabra dato, se obtiene en el receptor. Para este resto existen dos posibilidades, la primera es que sea igual a cero, en cuyo caso significa que la palabra transmitida llegó sin errores. La segunda es que el resto sea distinto de cero lo que significa que hay un error en la transmisión. Esta variable se muestra en los leds de la tarjeta Nexys 2 y se muestra al presionar uno de los botones de los estándares.

#### 4.3.4.6 Indicador de error

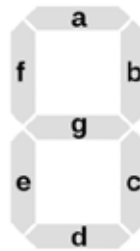
Esta variable indica si la transmisión se hizo sin errores; en tal caso los displays de la tarjeta Nexys 2 mostrarán las letras                    En caso de que haya un error en la transmisión mostrará las letras

Es importante indicar que luego de seleccionar la palabra dato correcta presionando el botón selector, si la misma se cambia a través de los switches, y se

presiona uno de los botones de estándares, los displays indicarán que tal palabra dato es errónea hasta que los switches vuelvan a la posición de la palabra que se seleccionó como correcta.

#### 4.3.4.7 Letra

Para hacer el Módulo Detector de Errores más didáctico se incluyeron algunos caracteres del código ASCII que pudiesen ser representados en los displays de la tarjeta Nexys 2. Viendo la figura 16 se puede darse cuenta que con un display 7 segmentos no se puede mostrar la totalidad de los caracteres del código ASCII, por lo que en la tabla 9 se incluyen todos los caracteres usados junto a su código binario.



**Figura 16.** Diagrama del display 7 segmentos

Fuente. Gallardo y Villarroel (2018)

**Tabla 9.** Caracteres mostrados por el Módulo Detector de Errores

G	f	e	d	c	b	a	Caracter	Código en Binario
1	0	0	0	0	0	0	0	00110000
1	1	1	1	0	0	1	1	00110001
0	1	0	0	1	0	0	2	00110010
0	1	1	0	0	0	0	3	00110011
0	0	1	1	0	0	1	4	00110100
0	0	1	0	0	1	0	5	00110101
0	0	0	0	0	1	1	6	00110110
1	1	1	1	0	0	0	7	00110111
0	0	0	0	0	0	0	8	00111000

0	0	1	1	0	0	0	9	00111001
0	0	0	1	0	0	0	A	01000001
0	0	0	0	0	1	1	b	01100010
1	0	0	0	1	1	0	C	01000011
0	1	0	0	0	0	1	d	01100100
0	0	0	0	1	1	0	E	01000101
0	0	0	1	1	1	0	F	01000110
0	0	1	0	0	0	0	g	01100111
0	0	0	1	0	1	1	h	01101000
1	0	0	1	1	1	1	I	01001001
1	1	1	0	0	0	1	J	01001010
1	0	0	0	1	1	1	L	01001100
0	1	0	1	0	1	1	n	01101110
1	0	0	0	0	0	0	O	01001111
0	0	0	1	1	0	0	P	01010000
0	0	1	1	0	0	0	q	01110001
0	1	0	1	1	1	1	r	01110010
0	0	1	0	0	1	0	S	01010011
1	0	0	0	0	0	1	U	01010101
0	0	0	1	0	0	1	X	01011000
0	0	1	1	0	0	1	y	01111001
0	1	0	0	1	0	0	Z	01011010

Fuente. Gallardo y Villarroel (2018)

#### 4.4 Fase IV: Creación de un manual de usuario para el módulo detector de errores que explique su funcionamiento y sus especificaciones.

Una vez investigado todo lo referente a la detección de errores mediante el método de redundancia cíclica y de haber desarrollado el módulo detector de errores, se vio la necesidad de crear un manual de usuario el cual permita tanto al facilitador

de la materia transmisión de datos como a sus estudiantes comprender el funcionamiento del módulo y cuáles son sus especificaciones.

Este manual parte con una pequeña introducción de lo referente a que es la detección de errores y que es el MDE por las siglas referente a “Módulo Detector de Errores”. Posteriormente se habla de las características generales que tiene este dispositivo, entre ella los componentes que se utilizan para este proyecto, debido a que la Nexys 2 es un FPGA muy amplia y no se utilizan todos los componentes de ella.

En la detección de errores por medio del método CRC existen una gran variedad de estándares, en el manual se explican los estándares mayormente utilizados en las telecomunicaciones que son los tres estándares que utilizamos para el proyecto, esto siempre con el fin de enfocar todo hacia la vida real de las telecomunicaciones donde se utiliza este método para la detección de los errores.

La FPGA Nexys 2 cómo sus siglas lo indican “Field Programmable Gate Array” este dispositivo programable contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurado dependiendo del lenguaje de descripción para ello se empleó VHDL como lenguaje de descripción, este dispositivo es necesario programarlo antes de utilizarlo, es decir, es necesario cargar el proyecto realizado en la fase III. Todo este procedimiento de programar la Nexys 2 se explica en el manual para que cualquier persona que desee utilizar el proyecto tenga las herramientas para hacerlo.

## CONCLUSIONES

Ningún método de detección de errores es completamente infalible y, en el caso de la comprobación de redundancia cíclica, aunque se usen polinomios característicos de gran longitud siempre habrá una pequeña probabilidad de que alguna palabra errada sea tomada como correcta; por lo que constantemente se están implementando y perfeccionando nuevos métodos de detección de errores para minimizar la probabilidad de error.

Es realmente necesaria la modernización de los laboratorios de la Universidad José Antonio Páez para brindarle a los estudiantes de la misma una calidad educativa que esté a la par con las exigencias actuales del campo laboral en el estado Carabobo por lo que se deben considerar inversiones en equipos más actuales y la promoción de proyectos que lleven a cabo la implementación de equipos educativos.

El Módulo Detector de Errores es una excelente herramienta para que los estudiantes de la materia Transmisión de Datos puedan aprender de manera práctica el funcionamiento del método de redundancia cíclica pudiendo ver paso a paso los procesos de codificación y decodificación de una palabra dato tanto en el transmisor como en el receptor.

El Módulo Detector de Errores podrá equipar al laboratorio de Transmisión de Datos que cuenta con muy pocos instrumentos y ninguno que esté dedicado específicamente a la enseñanza del método de detección de redundancia cíclica.

Para los fines de crear un programa que haga el proceso de codificación y decodificación según el método de redundancia cíclica, el uso de registros de almacenamientos es por mucho la mejor manera de calcular todos los bits del polinomio resto que se añaden a la palabra dato para crear la palabra código; tal algoritmo ahora considerablemente muchas líneas de código simplificando la programación.

Para conocer el verdadero impacto del Módulo Detector de Errores en la enseñanza de la materia Transmisión de datos, y en específico en el tema de los

métodos de detección de errores, se debe plantear otro estudio que le haga seguimiento a distintas variables como las calificaciones de los alumnos de Transmisión de datos durante unos varios semestres, promedio de calificaciones de la clase, etc. Tales variables serían medidas a partir de la implementación del Módulo Detector de Errores en los laboratorios de las materias.

## RECOMENDACIONES

- Promover que se sigan creando proyectos que involucren la programación de las tarjetas Nexys 2 para así suplir las necesidades que presentan los laboratorios de la Universidad José Antonio Páez e incluso diseñar herramientas completamente innovadoras que repliquen la gran cantidad de operaciones que se realizan en las telecomunicaciones.

- Crear una versión más avanzada del módulo partiendo del creado en este proyecto que pueda simular una comunicación en la que se comprueben más de una palabra dato e indicar las distintas métricas que se miden en comunicaciones digitales tales como la tasa de error binario (BER) o la velocidad de transmisión.

- Aunque solo se utilizaron los switches, botones, leds y displays de la tarjeta Nexys 2, la misma cuenta con una variedad de puertos que pueden aprenderse a programar para aprovecharlos en la creación de sistemas más elaborados en los que se pueda mostrar una mayor cantidad de datos.

- Incluir dentro del contenido programático de materias como circuitos digitales, transmisión de datos o procesamiento digital de señales la programación del lenguaje de descripción de hardware VHDL y/o Verilog. Dichos lenguajes no solo son utilizados en la programación de la tarjeta Nexys 2 sino también en otras matrices de compuertas programables con fines académicos e incluso profesionales que son de gran relevancia actualmente en empresas electrónicas como Intel e IBM haciendo a los graduados de esta institución más competitivos en el mercado laboral.

- Igualmente se pueden abrir cursos independientes sobre la programación de los lenguajes de descripción de software VHDL y Verilog que serán de gran utilidad para los estudiantes de ingeniería en telecomunicaciones, electrónica, computación y demás carreras afines. Tales cursos podrían incluso atraer a muchos externos ya que no se imparten estos tipos de cursos en otras instituciones educativas del estado Carabobo.

- Se puede desarrollar un programa en cualquier lenguaje de programación o de descripción de hardware que sea capaz de detectar todas las palabras que sean múltiplos de un polinomio característico dado y así estudiar más afondo las probabilidades de error de cada polinomio en específico, e incluso, se pueden estudiar más específicamente cualquier otro aspecto que sea de interés.

- Crear unos lineamientos para todos los técnicos de laboratorios y todos sus usuarios que indique cómo debe ser el uso de las tarjetas Nexys 2 para así preservarlas y que la mayor cantidad de alumnos se pueda beneficiar de ellas cada semestre.

- Crear prácticas y guías de laboratorio con una gran variedad de ejercicios prácticos que sirvan a los profesores para enseñar y evaluar la gran variedad de temas que pueden ser tratados con la tarjeta Nexys 2.

## REFERENCIAS BIBLIOGRÁFICAS

- Arias, F. (2012). **El proyecto de investigación, Introducción a la metodología científica**. Editorial EPISTEME. Caracas Venezuela.
- Borrero, F (2014). **Desarrollo de un manual de prácticas para el laboratorio de transmisión de datos**, tesis de grado, Universidad José Antonio Páez, Valencia edo. Carabobo.
- Cerdá, H. (1991). **Los elementos de la investigación: como reconocerlos, diseñarlos y construirlos**. Colombia: Editorial El Buho.
- Couch, L. (2.008). **Sistemas de comunicación digitales y analógicos**. México: Editorial Pearson Educación.
- Durán, G., Franco, J (2014), titulado **Multicodificador Reed-Solomon en software**, tesis de grado, realizada en el Instituto Politécnico Nacional de la Ciudad de México
- Fernando, S. (2013). **Simular en VHDL la implementación de algoritmos FEC en dispositivos programables reconfigurables**. Tesis de grado. Universidad Nacional Autónoma de México, México.
- Forouzan, B. (2.007). **Transmisión de datos y redes de comunicaciones**. España: Editorial McGraw-Hill.
- Freeman, R. (1.999). **Fundamentals of telecommunications**. Estados Unidos: John Wiley & Sons, Inc.
- Gustavo, D., Franco, J. (2014). **Multicodificador Reed-Solomon en software**. Tesis de grado. Instituto Politécnico Nacional, México.
- Hernández, S., Fernández, C., Baptista, M. (2010). **Metodología de la investigación**. México: Editorial McGraw-Hill.
- Labrador y Otros, (2002). **Metodología**. Valencia: Editorial Clemente.
- Magaña, E., Izkue, E., Prieto., Villadangos, J. (2003). **Comunicaciones y redes de computadores problemas y ejercicios resueltos**. España: Editorial Pearson Educación.
- Mendez, C. (2011). **Metodología: Diseño y desarrollo del proceso de investigación**. México: Editorial McGraw-Hill.

Sabino, C. (1.986). **El proceso de investigación**. Caracas: Panapo.

Serralde, F. (2013), **Simular en VHDL la implementación de algoritmos FEC en dispositivos programables configurables**, tesis de grado, Universidad Nacional Autónoma de México, México.

Stallings, W. (2.008). **Comunicaciones y redes de computadoras**. España: Editorial Pearson Educación.

Tanembaum, A. (2.003). **Redes de computadoras**. México: Editorial Pearson Educación.

Tomasi, W. (2.003). **Sistemas de comunicaciones electrónicas**. México: Editorial Pearson Educación.

## REFERENCIAS ELECTRÓNICAS

Ecured. (s/f) **Comprobación de redundancia cíclica**, consultado el 6 de junio de 2018 desde:  
[https://www.ecured.cu/Comprobaci%C3%B3n\\_de\\_redundancia\\_c%C3%ADcli%20ca](https://www.ecured.cu/Comprobaci%C3%B3n_de_redundancia_c%C3%ADcli%20ca)

De Matteis, L. (2.016) **Módulo 4 - Detección y corrección de errores (Pt. 1)**, Departamento de ciencias e ingeniería de la computación de la Universidad Nacional del Sur, consultado el 23 de junio de 2018 desde  
<https://cs.uns.edu.ar/~ldm/mypage/data/oc/apuntes/2017-modulo4.pdf>

Diligent Inc. (s/f). **Nexys 2**, imagen, consultado el 2 de julio de 2018 desde:  
[https://reference.digilentinc.com/\\_media/reference/programmable-logic/nexys-2/nexys2-0.png](https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-2/nexys2-0.png)

Detección de errores VRC, LRC, CRC  
[http://teleprospero.blogspot.com/2012/04/deteccion-de-errores-vrc-lrc-crc\\_23.html](http://teleprospero.blogspot.com/2012/04/deteccion-de-errores-vrc-lrc-crc_23.html)

Rouse, M. (2.010) **Cyclic redundancy check, Search Networking**, consultado el 5 de julio de 2018 desde: <https://searchnetworking.techtarget.com/definition/cyclic-redundancy-checking>

Thakur, D. (s/f) **Cyclic redundancy check (CRC)**, Ecomputer Notes, consultado el 6 de junio de 2018 desde:  
<http://ecomputernotes.com/computernetworkingnotes/communication-networks/cyclic-redundancy-check>

Universidad de Hong Kong. (s/f) **Part 2.2 Cyclic redundancy check (CRC) codes**, consultado el 24 de abril de 2018, desde:  
[https://www.eee.hku.hk/~sdma/elec7073/Part2-2-Cyclic%20redundancy%20check%20\(CRC\).pdf](https://www.eee.hku.hk/~sdma/elec7073/Part2-2-Cyclic%20redundancy%20check%20(CRC).pdf)

## **ANEXOS**

## Anexo A. Código en VHDL.

```
crc_encoder2.vhd Thu Dec 13 15:12:59 2018
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity crc_encoder2 is
5      Port ( d : in  STD_LOGIC_VECTOR (7 downto 0);
6            clk : in  STD_LOGIC;
7            estandar1 : in  STD_LOGIC;
8            estandar2 : in  STD_LOGIC;
9            estandar3 : in  STD_LOGIC;
10           resto : in  STD_LOGIC;
11           seg : out  STD_LOGIC_VECTOR (6 downto 0);
12           selec : out  STD_LOGIC_VECTOR (3 downto 0);
13           mostrar : out  STD_LOGIC_VECTOR (7 downto 0));
14
15  end crc_encoder2;
16
17  architecture Behavioral of crc_encoder2 is
18  signal cw1 : STD_LOGIC_VECTOR (10 downto 0) := "000000000000"; --codeword 1
19  signal cw2 : STD_LOGIC_VECTOR (11 downto 0) := "0000000000000"; --codeword 2
20  signal cw3 : STD_LOGIC_VECTOR (13 downto 0) := "000000000000000"; --codeword 3
21  signal r_aux1 : STD_LOGIC_VECTOR (2 downto 0) := "000"; --resto 1
22  signal r_aux2 : STD_LOGIC_VECTOR (3 downto 0) := "0000"; --resto 2
23  signal r_aux3 : STD_LOGIC_VECTOR (5 downto 0) := "000000"; --resto 3
24  signal daux : STD_LOGIC_VECTOR (7 downto 0) := "00000000";
25  signal dmostrar : STD_LOGIC_VECTOR (7 downto 0) := "00000000";
26  signal resto_receptor : STD_LOGIC_VECTOR (5 downto 0) := "000000";
27  signal display_sel: STD_LOGIC_VECTOR(3 downto 0) := "0000";
28
29  begin
30
31      dmostrar <= d;
32
33      -- Resto auxiliar del estandar 1
34      r_aux1(2) <= d(6) xor d(3) xor d(1) xor d(2);
35      r_aux1(1) <= d(7) xor d(2) xor d(0) xor d(5) xor d(1);
36      r_aux1(0) <= d(7) xor d(4) xor d(3) xor d(2) xor d(0);
37
38
39      -- Resto auxiliar del estandar 2
40      r_aux2(3) <= d(2) xor d(3) xor d(5) xor d(7);
41      r_aux2(2) <= d(1) xor d(2) xor d(4) xor d(6) xor d(7);
42      r_aux2(1) <= d(0) xor d(1) xor d(3) xor d(5) xor d(6) xor d(7);
43      r_aux2(0) <= d(0) xor d(3) xor d(4) xor d(6);
44
45      -- Resto auxiliar del estandar 3
46      r_aux3(5) <= d(4) xor d(1) xor d(0);
47      r_aux3(4) <= d(7) xor d(4) xor d(3) xor d(1);
48      r_aux3(3) <= d(7) xor d(6) xor d(3) xor d(2) xor d(0);
49      r_aux3(2) <= d(7) xor d(6) xor d(5) xor d(4) xor d(2) xor d(0);
50      r_aux3(1) <= d(6) xor d(5) xor d(3) xor d(0);
51      r_aux3(0) <= d(5) xor d(2) xor d(1) xor d(0);
52
53  selec <= display_sel;
54
55  process (clk)
56  begin
57      if (clk'event and clk = '1') then
```

```
58
59     --- EMISOR
60
61     if estandar1 = '1' and resto = '1' then
62
63         mostrar(7 downto 3) <= "00000"; --Reset
64         seg <= "1001110";
65
66         mostrar(2) <= r_aux1(2);
67         mostrar(1) <= r_aux1(1);
68         mostrar(0) <= r_aux1(0);
69
70     elsif estandar2 = '1' and resto = '1' then
71
72         mostrar(7 downto 4) <= "0000"; -- Reset
73         seg <= "1001110";
74
75         mostrar(3) <= r_aux2(3);
76         mostrar(2) <= r_aux2(2);
77         mostrar(1) <= r_aux2(1);
78         mostrar(0) <= r_aux2(0);
79
80     elsif estandar3 = '1' and resto = '1' then
81
82         mostrar(7 downto 6) <= "00"; --Reset
83         seg <= "1001110";
84
85         mostrar(5) <= r_aux3(5);
86         mostrar(4) <= r_aux3(4);
87         mostrar(3) <= r_aux3(3);
88         mostrar(2) <= r_aux3(2);
89         mostrar(1) <= r_aux3(1);
90         mostrar(0) <= r_aux3(0);
91
92     -- RECEPTOR
93
94     elsif resto = '1' then
95
96         daux <= d;
97
98     elsif estandar1 = '1' then
99
100         cw1(10 downto 3) <= daux;
101         cw1(2) <= r_aux1(2);
102         cw1(1) <= r_aux1(1);
103         cw1(0) <= r_aux1(0);
104
105         mostrar(7 downto 3) <= "00000"; --reset
106
107         resto_receptor(5 downto 3) <= "000";
108
109         resto_receptor(2) <= cw1(9) xor cw1(6) xor cw1(4) xor cw1(5) xor cw1(2);
110         resto_receptor(1) <= cw1(10) xor cw1(5) xor cw1(3) xor cw1(8) xor cw1(4) xor
111         cw1(1);
112         resto_receptor(0) <= cw1(10) xor cw1(7) xor cw1(6) xor cw1(5) xor cw1(3) xor
113         cw1(0);
```

```
113     mostrar(2 downto 0) <= resto_receptor(2 downto 0);
114
115     if resto_receptor(5 downto 0) = "000000" then --resto igual a 0 entonces PPPP
116
117         display_sel <= "0000";
118         seg <= "0001100";
119
120     else -- Si no, entonces EEEE
121
122         display_sel <= "0000";
123         seg <= "0000110";
124
125     end if;
126
127     elsif estandar2 = '1' then
128
129         cw2(11 downto 4) <= daux;
130
131         cw2(3) <= r_aux2(3);
132         cw2(2) <= r_aux2(2);
133         cw2(1) <= r_aux2(1);
134         cw2(0) <= r_aux2(0);
135
136         mostrar(7 downto 4) <= "0000"; -- Reset
137
138         resto_receptor(5 downto 4) <= "00";
139
140         resto_receptor(3) <= cw2(6) xor cw2(7) xor cw2(9) xor cw2(11) xor cw2(3);
141         resto_receptor(2) <= cw2(5) xor cw2(6) xor cw2(8) xor cw2(10) xor cw2(11) xor
142         cw2(2);
143         resto_receptor(1) <= cw2(4) xor cw2(5) xor cw2(7) xor cw2(9) xor cw2(10) xor
144         cw2(11) xor cw2(1);
145         resto_receptor(0) <= cw2(4) xor cw2(7) xor cw2(8) xor cw2(10) xor cw2(0);
146
147         mostrar(3 downto 0) <= resto_receptor(3 downto 0);
148
149         if resto_receptor(5 downto 0) = "000000" then
150
151             display_sel <= "0000";
152             seg <= "0001100";
153
154         else
155
156             display_sel <= "0000";
157             seg <= "0000110";
158
159         end if;
160
161     elsif estandar3 = '1' then
162
163         cw3(13 downto 6) <= daux;
164
165         cw3(5) <= r_aux3(5);
166         cw3(4) <= r_aux3(4);
167         cw3(3) <= r_aux3(3);
168         cw3(2) <= r_aux3(2);
169         cw3(1) <= r_aux3(1);
```

```
168         cw3(0) <= r_aux3(0);
169
170     mostrar(7 downto 6) <= "00"; --Reset
171
172     resto_receptor(5) <= cw3(10) xor cw3(7) xor cw3(6) xor cw3(5);
173     resto_receptor(4) <= cw3(13) xor cw3(10) xor cw3(9) xor cw3(7) xor cw3(4);
174     resto_receptor(3) <= cw3(13) xor cw3(12) xor cw3(9) xor cw3(8) xor cw3(6) xor
    cw3(3);
175     resto_receptor(2) <= cw3(13) xor cw3(12) xor cw3(11) xor cw3(10) xor cw3(8)
xor cw3(6) xor cw3(2);
176     resto_receptor(1) <= cw3(12) xor cw3(11) xor cw3(9) xor cw3(6) xor cw3(1);
177     resto_receptor(0) <= cw3(11) xor cw3(8) xor cw3(7) xor cw3(6) xor cw3(0);
178
179     mostrar(5 downto 0) <= resto_receptor(5 downto 0);
180
181     if resto_receptor(5 downto 0) = "000000" then
182
183         display_sel <= "0000";
184         seg <= "0001100";
185
186     else
187
188         display_sel <= "0000";
189         seg <= "0000110";
190
191     end if;
192
193     else --Lo que ocurre cuando no se aprieta ningún botón
194
195     mostrar <= dmostrar;
196
197     display_sel <= "1110"; -- display 0
198     case d is
199         when "00110000" =>
200             seg <= "1000000"; -- 0
201         when "00110001" =>
202             seg <= "1111001"; -- 1
203         when "00110010" =>
204             seg <= "0100100"; -- 2
205
206         when "00110011" =>
207             seg <= "0110000"; -- 3
208         when "00110100" =>
209             seg <= "0011001"; -- 4
210         when "00110101" =>
211             seg <= "0010010"; -- 5
212         when "00110110" =>
213             seg <= "0000011"; -- 6
214         when "00110111" =>
215             seg <= "1111000"; -- 7
216         when "00111000" =>
217             seg <= "0000000"; -- 8
218         when "00111001" =>
219             seg <= "0011000"; -- 9
220         when "01000001" =>
221             seg <= "0001000"; -- A
222         when "01100010" =>
```

```
223         seg <= "0000011"; -- b
224     when "01000011" =>
225         seg <= "1000110"; -- C
226     when "01100100" =>
227         seg <= "0100001"; -- d
228     when "01000101" =>
229         seg <= "0000110"; -- E
230     when "01000110" =>
231         seg <= "0001110"; -- F
232     when "01100111" =>
233         seg <= "0010000"; -- g
234     when "01101000" =>
235         seg <= "0001011"; -- h
236     when "01001001" =>
237         seg <= "1001111"; -- I
238     when "01001010" =>
239         seg <= "1110001"; -- J
240
241     when "01001100" =>
242         seg <= "1000111"; -- L
243     when "01101110" =>
244         seg <= "0101011"; -- n
245     when "01001111" =>
246         seg <= "1000000"; -- O
247     when "01010000" =>
248         seg <= "0001100"; -- P
249     when "01110001" =>
250         seg <= "0011000"; -- q
251     when "01110010" =>
252         seg <= "0101111"; -- r
253     when "01010011" =>
254         seg <= "0010010"; -- S
255     when "01010101" =>
256         seg <= "1000001"; -- U
257     when "01011000" =>
258         seg <= "0001001"; -- X
259     when "01111001" =>
260         seg <= "0011001"; -- y
261     when "01011010" =>
262         seg <= "0100100"; -- Z
263     when others =>
264         display_sel <= "0000" ;
265         seg <= "0111111"; -- PPPP
266     end case;
267
268     end if;
269 end if;
270 end process;
271
272 end Behavioral;
273
```

**Anexo B.** Manual de usuario.



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN TELECOMUNICACIONES**

# **Manual de Módulo Detector de Errores**

**Realizado por:  
Gallardo Luis  
Villarroel Gabriel**

## ÍNDICE GENERAL

<b>CONTENIDO</b>	<b>Pp</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
Nexys 2 .....	2
Detección de errores por comprobación de redundancia cíclica .....	2
Diagrama del módulo detector de errores .....	3
Palabra dato .....	4
Botones .....	4
Selector .....	4
Estándares.....	4
Resto transmisor .....	5
Resto receptor .....	6
Indicador de error .....	6
Carácter .....	6
Tabla de Asignación de caracteres según el código ASCII .....	7
Diagilent Adept .....	8
Archivo .bit .....	8
Como programar la tarjeta Nexys 2 .....	8
Ejemplo .....	9

## INTRODUCCIÓN

La detección de errores es uno de los tópicos más importante en el mundo de las telecomunicaciones. Como ya se ha debido estudiar, el método de comprobación de redundancia cíclica es uno de los más usados en campo de las comunicaciones, y es necesario que los estudiantes de ingeniería en telecomunicaciones, electrónica, y demás carreras afines cuenten con los mejores recursos posibles para entender cómo funciona la detección de errores por comprobación de redundancia cíclica (CRC). Es por ello que se ha diseñado este manual de usuario del Módulo Detector de Errores (MDE) para los estudiantes de Transmisión de Datos de la UJAP. y así explicar, de manera teórica-práctica, cómo se ejecuta el método CRC y cómo se comporta ante una palabra código correcta o incorrecta.

El MDE fue desarrollado en la matriz de puertas programables (FPGA por sus siglas en inglés) Nexys 2. Esta tarjeta cuenta con suficientes entradas y salidas para hacer cómoda la experiencia del estudiante en la simulación de errores en un sistema de comunicaciones. Incluso contando con 3 estándares distintos de Comprobación de Redundancia Cíclica; facilitando a los estudiantes, y profesores de materias que incluyan los métodos de corrección de errores en su contenido programático, tanto la enseñanza como el aprendizaje por ambas partes.

## Nexys 2

La tarjeta Nexys2 es una plataforma de desarrollo de circuitos completa y lista para usar, basada en un Xilinx Spartan 3E FPGA. Con un puerto USB2 de alta velocidad incorporado, 16 MB de RAM y ROM, y varios puertos de entrada y salida, convierten a la Nexys2 en una plataforma ideal para sistemas digitales de todo tipo. El puerto USB2 proporciona alimentación de la tarjeta y una interfaz de programación, por lo que la tarjeta Nexys2 se puede usar con una computadora portátil para crear una estación de diseño verdaderamente portátil.

### Detección de errores por comprobación de redundancia cíclica

Es un método de detección de errores ampliamente usado en las telecomunicaciones donde se agregan bits redundantes al final de la transmisión. Los cálculos realizados en el método CRC se basan en el uso de un polinomio característico de grado  $r$ , el cual debe ser conocido tanto por el transmisor como por el receptor.

En la imagen 1 se explica el procedimiento tanto en el transmisor como en el receptor para hacer una transmisión aplicando la comprobación de redundancia cíclica. Se recomienda familiarizarse con cada término mostrado en la imagen 1 ya que se mencionan de manera recurrente en este manual y de ellos dependerá la comprensión del proceso de detección de errores.

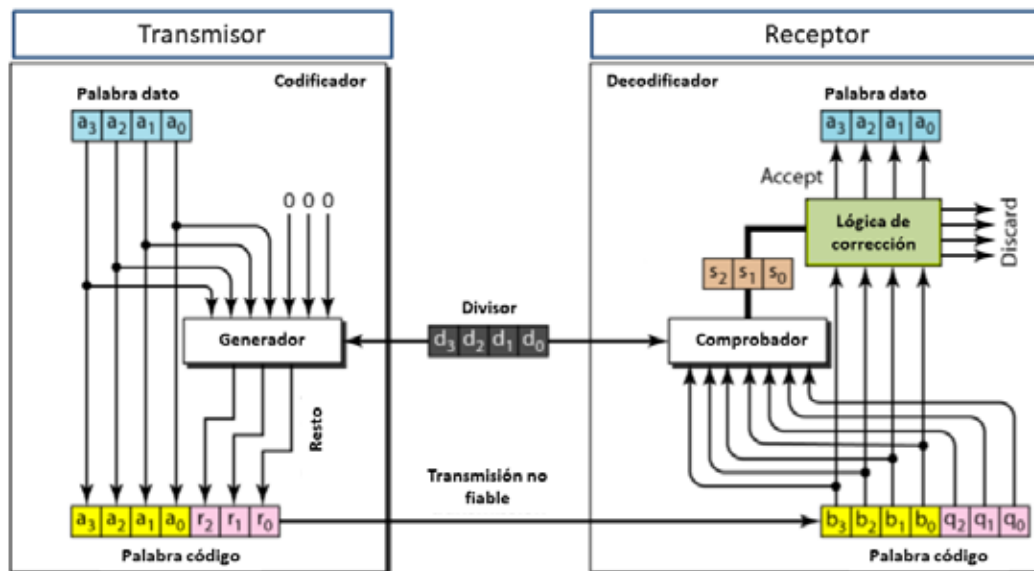


Imagen 1

## Diagrama del Módulo Detector de Errores

En la imagen 2 se puede ver el diagrama que representa las entradas y salidas con las que cuenta el Módulo Detector de errores.

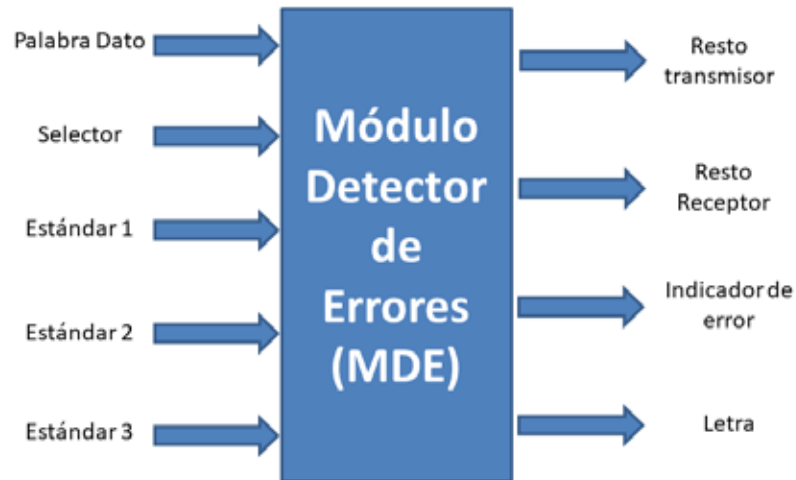


Imagen 2

En la imagen 3 se indica dónde se asignaron las entradas y por donde se muestran las salidas.

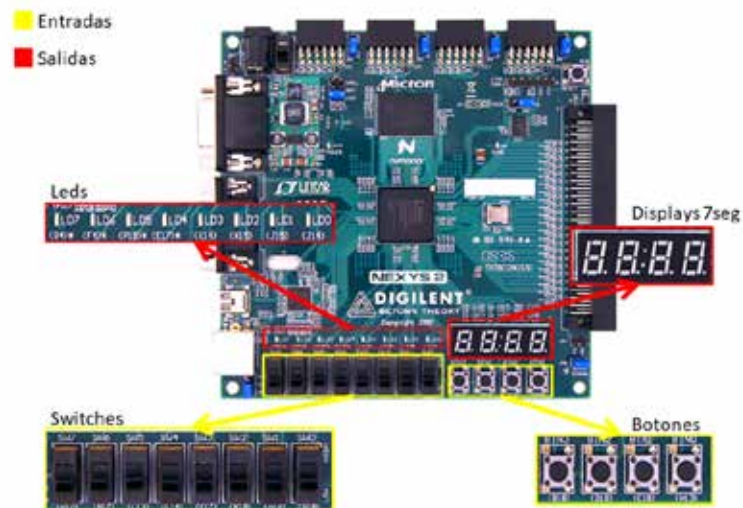


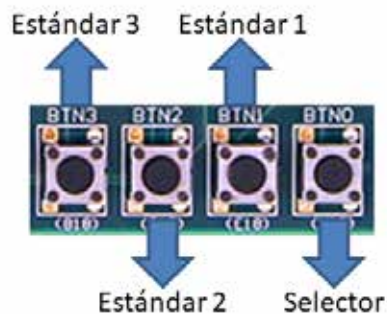
Imagen 3

## Palabra dato

Es el vector de números binarios que se desea enviar por un sistema de comunicaciones. En el MDE se introduce a través de los **switches** de la tarjeta Nexys 2 (Ver **Imagen 3**). El tamaño de cada palabra dato será de 8 bits (un bit por cada switch).

## Botones

La tarjeta Nexys 2 cuenta con 4 botones. El MDE no solo usa cada uno de ellos por separado, sino que también acepta combinaciones entre ellos para mostrar cierta información. En la **imagen 4** se muestra la asignación que se le ha dado a cada botón.



**Imagen 4**

## Selector

Luego de haber elegido una palabra dato a través de los switches, se pulsa el botón Selector (Ver imagen 3) para que este guarde la palabra que en ese momento se encuentre en los switches como la palabra dato correcta. Cada vez que se pulse el botón Selector se guardará una nueva palabra dato “correcta”.

**Nota:** Solo se guarda una palabra dato “correcta” a la vez.

## Estándares

El MDE cuenta con 3 estándares de CRC asignados a tres de sus cuatro botones. Los polinomios característicos de cada estándar son los siguientes:

### **Estándar 1**

Nombre: CRC 3

Uso: GSM Mobile Networks

Polinomio característico:

Representación binaria: 1011

### **Estándar 2**

Nombre: CRC 4

Uso: ITU-T G.704

Polinomio característico:

Representación binaria: 10011

### **Estándar 3**

Nombre: CRC 6

Uso: GSM Mobile Networks

Polinomio característico:

Representación binaria: 1101111

Al presionar uno de estos botones el MDE decodificará la palabra introducida y los leds mostrarán el resto calculado en el receptor. Si el resto es 0, es decir, ningún led se ilumina, significa que no hay error en la transmisión, de lo contrario, hay un error en la transmisión.

También se pueden mostrar a través de los leds el resto con el que el transmisor codifica el mensaje presionando simultáneamente el botón selector y el botón del estándar deseado. Las tres posibles combinaciones de botones son las siguientes:

- 1.- Estándar 1 + Selector
- 2.- Estándar 2 + Selector
- 3.- Estándar 3 + Selector

### **Resto transmisor**

Esta variable es el polinomio resto que se calcula en el transmisor para codificar la palabra dato. Esta variable se muestra en los leds de la tarjeta Nexys 2 y para ser mostrada se deben presionar simultáneamente el botón selector y el botón del estándar de cual se desee conocer el resto.

## **Resto receptor**

A diferencia del resto transmisor, este es el resto que, luego de decodificar la palabra dato, se obtiene en el receptor. Para este resto existen dos posibilidades, la primera es que sea igual a cero, en cuyo caso significa que la palabra transmitida llegó sin errores. La segunda es que el resto sea distinto de cero lo que significa que hay un error en la transmisión. Esta variable se muestra en los leds de la tarjeta Nexys 2 y se muestra al presionar uno de los botones de los estándares.

## **Indicador de error**

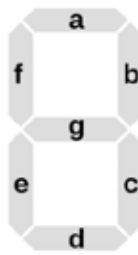
Esta variable indica si la transmisión se hizo sin errores; en tal caso los displays de la tarjeta Nexys 2 mostrarán las letras                      En caso de que haya un error en la transmisión mostrará las letras

Es importante indicar que luego de seleccionar la palabra dato correcta presionando el botón selector, si la misma se cambia a través de los switches, y se presiona uno de los botones de estándares, los displays indicarán que tal palabra dato es errónea hasta que los switches vuelvan a la posición de la palabra que se seleccionó como correcta.

## **Carácter**

El Módulo Detector de Errores muestra algunos caracteres del código ASCII a través de los displays de la tarjeta Nexys 2. En la tabla 1 se incluyen todos los caracteres usados junto a su código binario y los leds del display que se iluminan en cada caso según la imagen 5.

Para cualquier otra combinación de bits diferente a la establecida en la tabla se mostrará en los displays “----”.



**Imagen 5**

**Tabla 1.** Asignación de caracteres según el código ASCII

<b>G</b>	<b>f</b>	<b>e</b>	<b>d</b>	<b>c</b>	<b>b</b>	<b>a</b>	<b>Carácter</b>	<b>Código en Binario</b>
1	0	0	0	0	0	0	0	<b>00110000</b>
1	1	1	1	0	0	1	1	<b>00110001</b>
0	1	0	0	1	0	0	2	<b>00110010</b>
0	1	1	0	0	0	0	3	<b>00110011</b>
0	0	1	1	0	0	1	4	<b>00110100</b>
0	0	1	0	0	1	0	5	<b>00110101</b>
0	0	0	0	0	1	1	6	<b>00110110</b>
1	1	1	1	0	0	0	7	<b>00110111</b>
0	0	0	0	0	0	0	8	<b>00111000</b>
0	0	1	1	0	0	0	9	<b>00111001</b>
0	0	0	1	0	0	0	A	<b>01000001</b>
0	0	0	0	0	1	1	b	<b>01100010</b>
1	0	0	0	1	1	0	C	<b>01000011</b>
0	1	0	0	0	0	1	d	<b>01100100</b>
0	0	0	0	1	1	0	E	<b>01000101</b>
0	0	0	1	1	1	0	F	<b>01000110</b>
0	0	1	0	0	0	0	g	<b>01100111</b>
0	0	0	1	0	1	1	h	<b>01101000</b>
1	0	0	1	1	1	1	I	<b>01001001</b>
1	1	1	0	0	0	1	J	<b>01001010</b>

1	0	0	0	1	1	1	L	01001100
0	1	0	1	0	1	1	n	01101110
1	0	0	0	0	0	0	O	01001111
0	0	0	1	1	0	0	P	01010000
0	0	1	1	0	0	0	q	01110001
0	1	0	1	1	1	1	r	01110010
0	0	1	0	0	1	0	S	01010011
1	0	0	0	0	0	1	U	01010101
0	0	0	1	0	0	1	X	01011000
0	0	1	1	0	0	1	y	01111001
0	1	0	0	1	0	0	Z	01011010

## Digilent Adept

El software gratuito de Digilent Adept es un software gratuito el cual permite a las aplicaciones intercambiar datos con FPGA de Xilinx. Este Software es una solución única y poderosa para todo lo relacionado con la comunicación de tanto las tarjetas Digilent como con una amplia variedad de dispositivos lógicos.

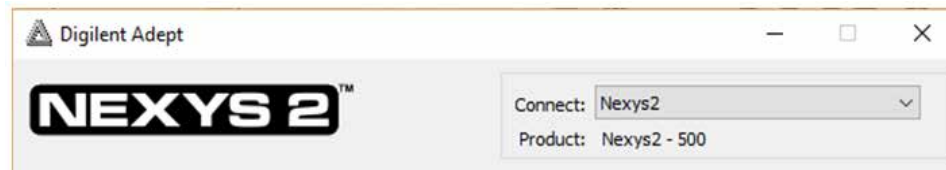
## Archivo .bit

El archivo BIT es uno de archivos de la categoría Archivos CAD. Su nombre completo es Xilinx Bitstream File. El formato de archivo BIT ha sido creado por Xilinx.

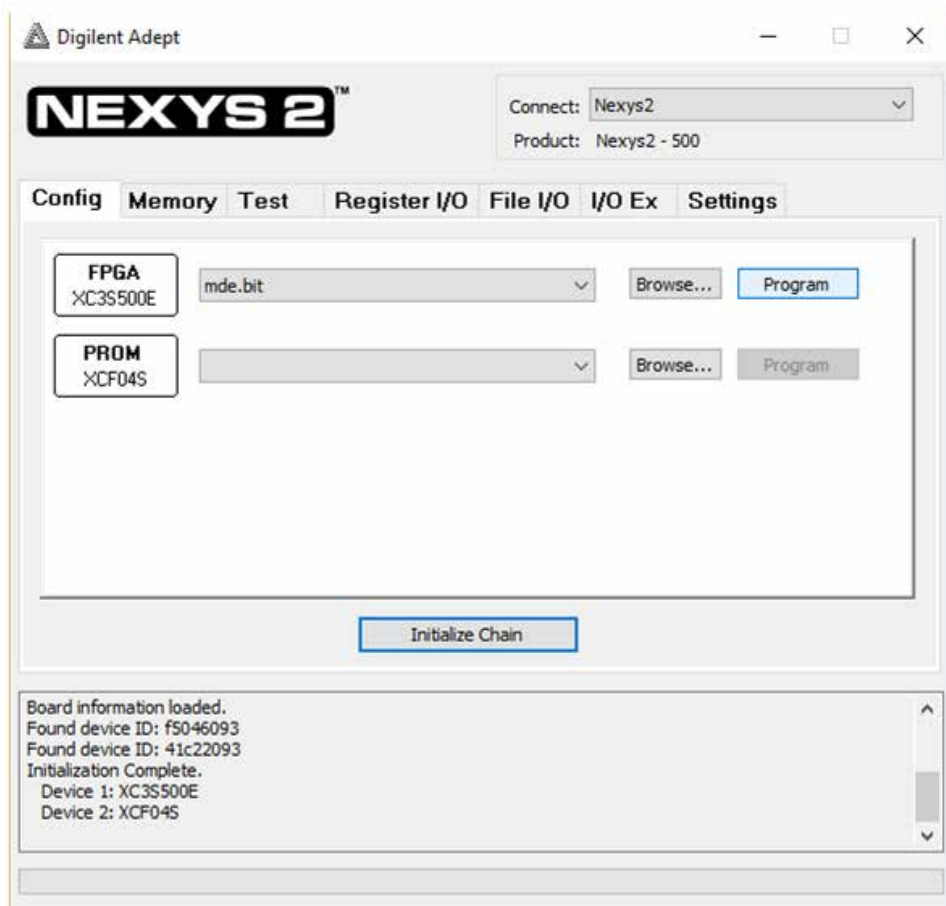
## Como programar la tarjeta Nexys 2

Para la programación de la tarjeta Nexys 2 se debe realizar los siguientes pasos:

1. Se debe tener instalado previamente el programa Adept, con el cual subiremos el archivo .bit a la tarjeta.
2. Conectar la tarjeta Nexys 2 al ordenador, verificar con el Adept en la parte superior derecha que salga el nombre Nexys 2.



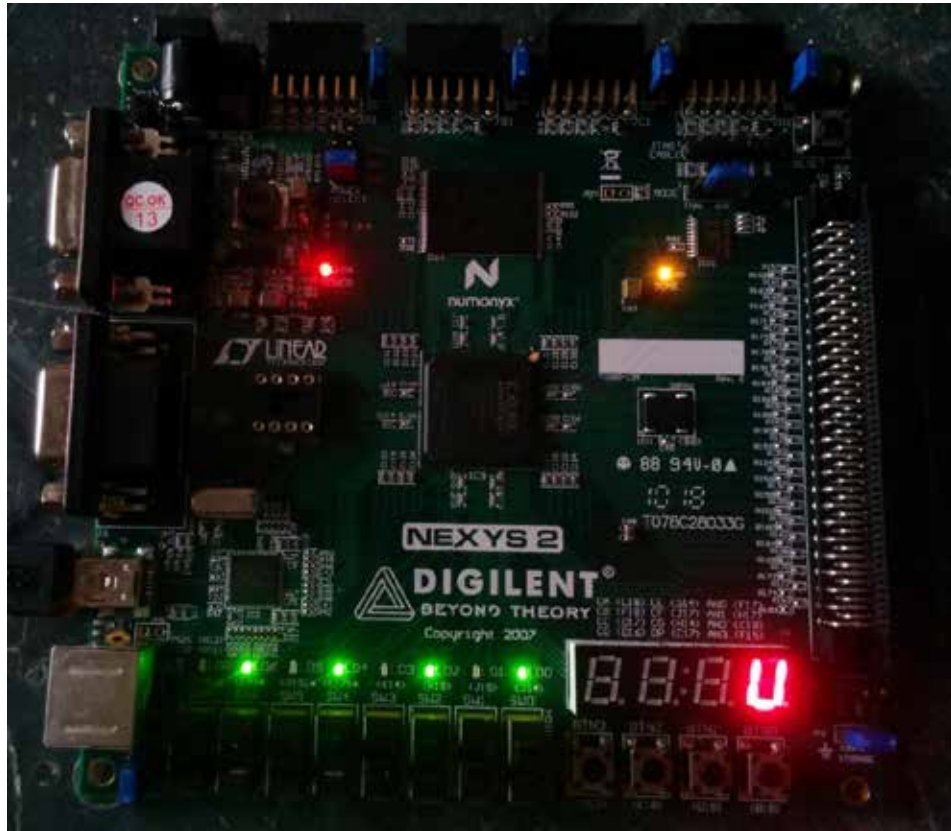
3. Una vez reconocida la tarjeta se le agrega el archivo . bit



4. Finalmente se presiona el botón que dice **Program** para que el archivo se programe en la tarjeta.

### Ejemplo

Una vez programada la tarjeta Nexys 2, se procede a introducir la palabra dato 01010101, la cual tiene como carácter la letra “U” mayúscula. Este carácter se puede observar en el display 7 segmentos siempre y cuando este en la Tabla 1.

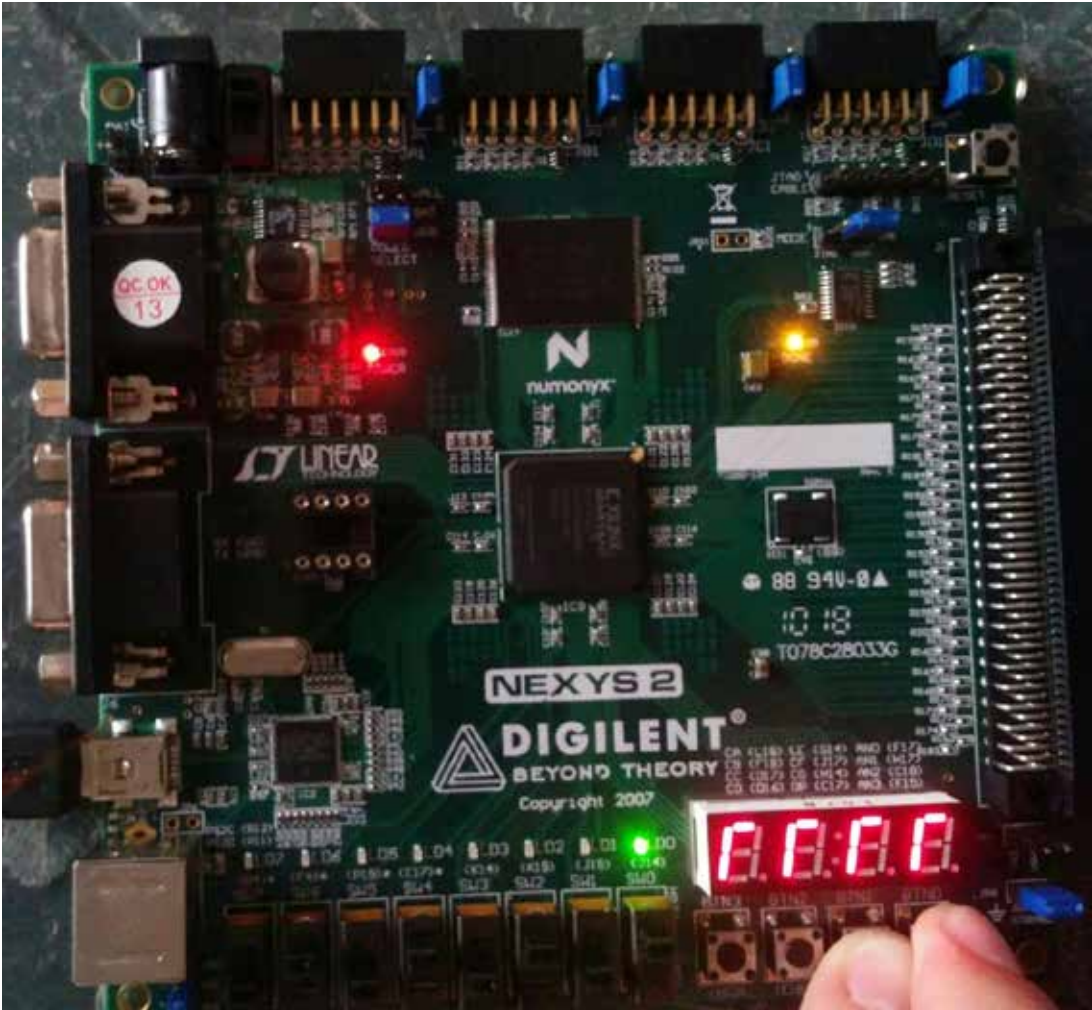


Presionando el botón “Selector”, para establecer la palabra dato como la correcta.

**En el transmisor:**

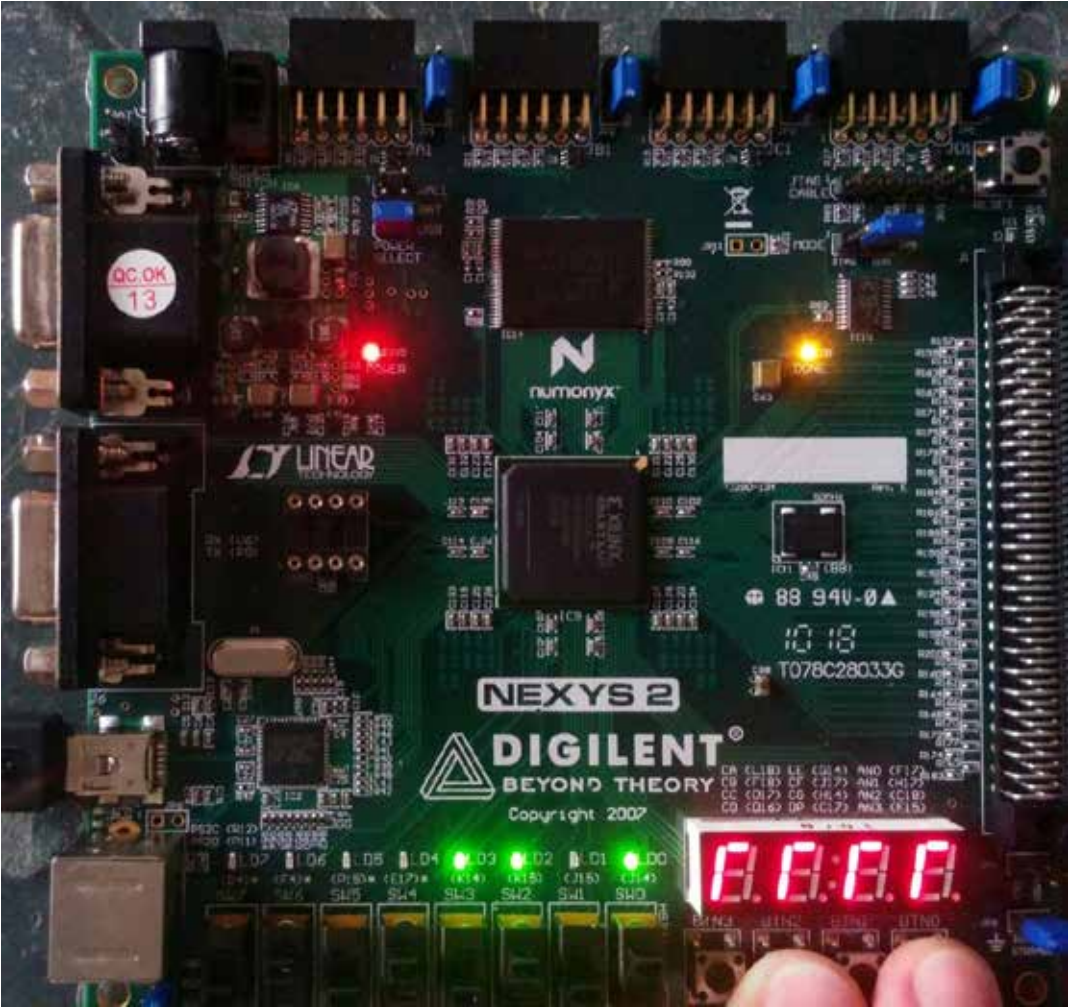
Para conocer el resto que se le debe añadir a la palabra dato, para ser enviada por el transmisor se presionan simultáneamente el botón de “Selector” y cualquiera de los tres estándares:

**Estándar 1**



Se puede observar que el resto es 001

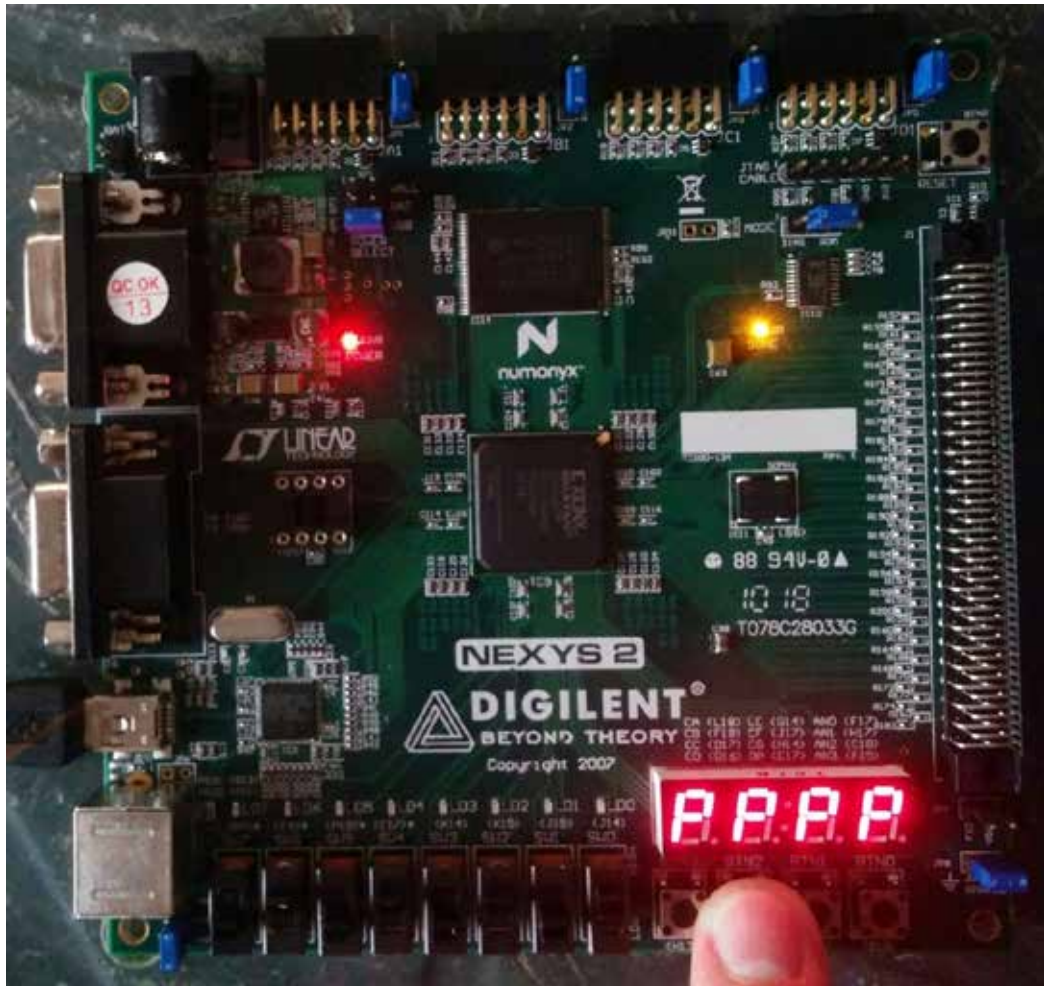
**Estándar 2**



Se puede observar que el resto es 1011



Al presionar cualquier botón de estándar, se puede que los leds están apagados indicando un resto de 0, además se observar en el display señalando que la transmisión se realizó de manera exitosa.

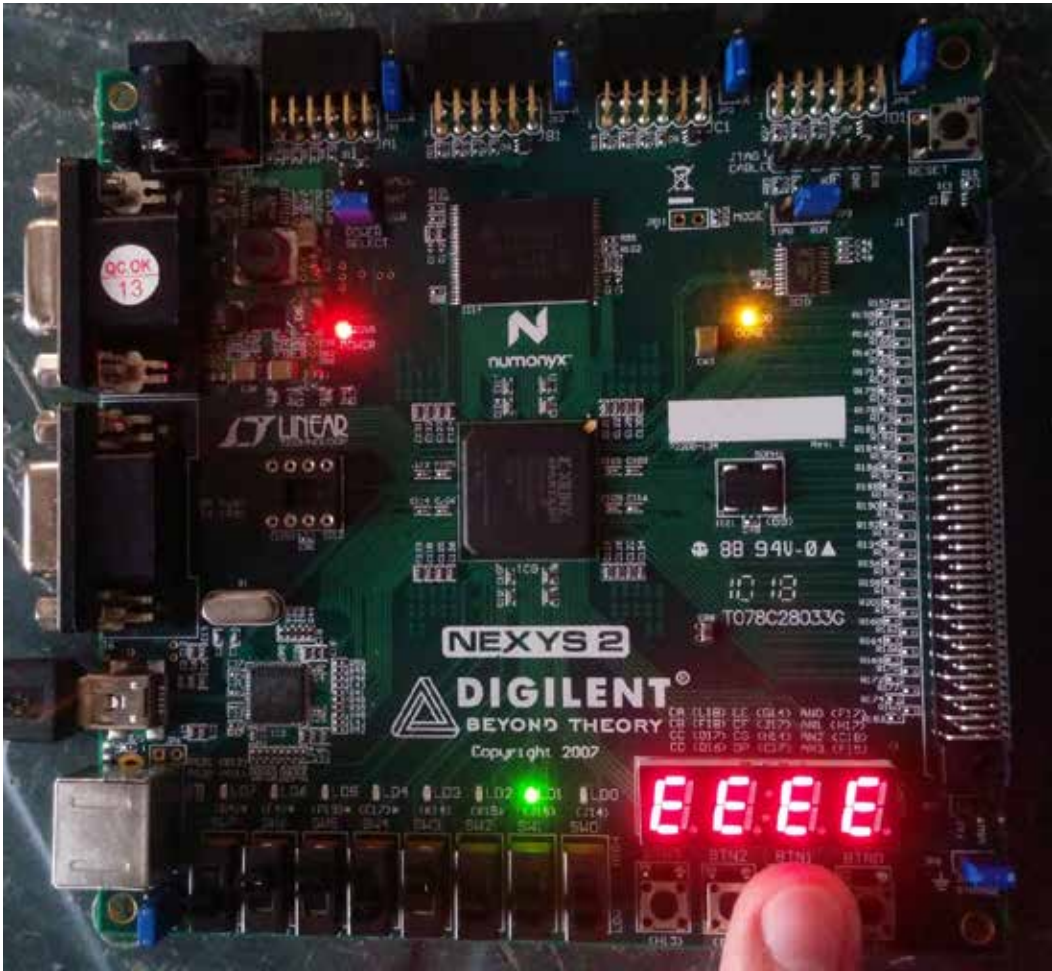


**Error:**

Si se cambia la palabra datos a cualquier otra combinación diferente a la que teníamos anteriormente, en este caso tomaremos 01101000, la cual tiene como carácter la letra “h”, al presionar cualquiera de los botones de estándar podemos

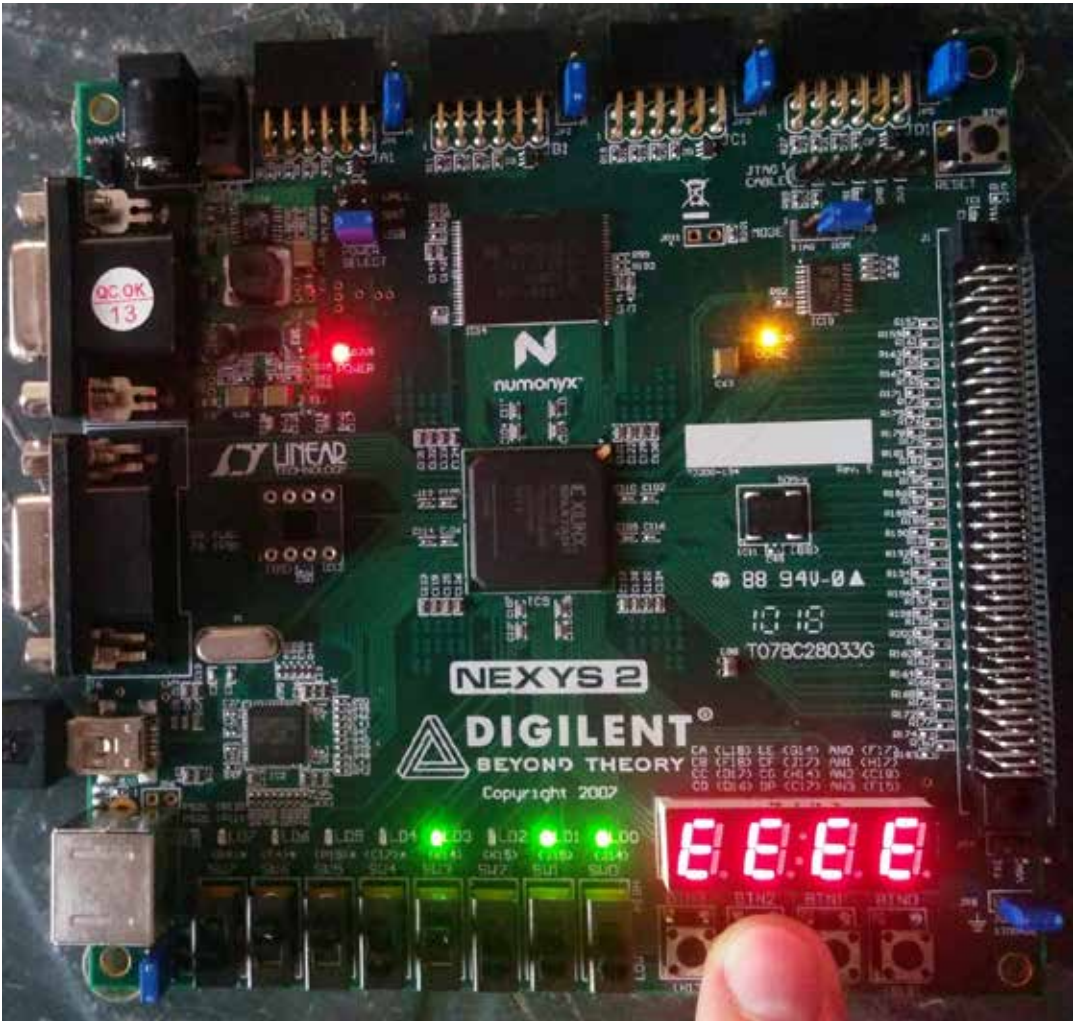
observar que hay un error, indicado en los display con las letras “ ” en la transmisión debido a que la palabra correcta es 01010101, carácter “U”.

**Estándar 1**



Se puede observar que el resto es 010

**Estándar 2**



Se puede observar que el resto es 1011

**Estándar 3**



Se puede observar que el resto es 001011

Como todos estos restos son diferentes de “0” el mensaje no llego de forma correcta y por ende hay un error en la transmisión.