

Arquitectura Digital para un Detector de Símbolos ML

**TESIS**

QUE PARA OBTENER EL GRADO DE

**MAESTRO EN CIENCIAS DE LA INGENIERÍA**

PRESENTA

**Aarón Escoboza Villegas**

CIUDAD OBREGÓN, SONORA

SEPTIEMBRE 2021

**Índice**

[CAPÍTULO I. INTRODUCCIÓN 4](#_Toc79514864)

[1.1 Antecedentes 4](#_Toc79514865)

[1.2 Planteamiento del problema 9](#_Toc79514866)

[1.3 Objetivo 10](#_Toc79514867)

[1.4 Hipótesis 10](#_Toc79514868)

[1.5 Justificación 10](#_Toc79514869)

[1.6 Delimitaciones 11](#_Toc79514870)

[1.7 Limitaciones 11](#_Toc79514871)

[CAPÍTULO II. MARCO TEÓRICO 12](#_Toc79514872)

[2.1 El estándar IEEE 802.11p 12](#_Toc79514873)

[2.2 Pilotos en el estándar 802.11p 13](#_Toc79514874)

[2.3 Canal de comunicación V2V 14](#_Toc79514875)

[2.4 Sistemas de comunicación V2V 15](#_Toc79514876)

[2.5 Detección de datos 17](#_Toc79514877)

[**2.5.1 Modelo del receptor** 17](#_Toc79514878)

[**2.5.2 Detector OSIC** 20](#_Toc79514879)

[**2.5.3 Detector QR-ML convencional** 20](#_Toc79514880)

[**2.5.4 Detector V2V Near ML** 22](#_Toc79514881)

[2.6 Modelo algorítmico 24](#_Toc79514882)

[2.6.1 Mapeo algoritmo a arquitectura 24](#_Toc79514883)

[2.6.2 Modelo de oro 24](#_Toc79514884)

[2.6.3 Adecuaciones al algoritmo 25](#_Toc79514885)

[2.6.4 Análisis de punto fijo 25](#_Toc79514886)

[CAPÍTULO III. MÉTODO 29](#_Toc79514887)

[3.1 Sujeto 30](#_Toc79514888)

[3.2 Procedimiento 30](#_Toc79514889)

[3.3 Materiales y Herramientas 32](#_Toc79514890)

[CAPÍTULO IV. DESARROLLO 33](#_Toc79514891)

[4.1 Desarrollo del modelo algoritmo y sistema SISO V2V 33](#_Toc79514892)

[**4.1.1 Modelo de oro** 33](#_Toc79514893)

[**4.1.2 Sistema SISO V2V** 35](#_Toc79514894)

[**4.1.3 Análisis de punto fijo** 37](#_Toc79514895)

[4.2 Arquitectura propuesta 40](#_Toc79514896)

[**4.2.1 Parametrización** 40](#_Toc79514897)

[**4.2.2 Nivel principal** 41](#_Toc79514898)

[4.2.3 Ordenador 42](#_Toc79514899)

[4.2.4 Bloque 2 43](#_Toc79514900)

[4.2.5 Bloque 3 44](#_Toc79514901)

[CAPÍTULO V. RESULTADOS 44](#_Toc79514902)

[5.1 Síntesis 44](#_Toc79514903)

[5.2 Latencia 44](#_Toc79514904)

[5.3 Verificación 44](#_Toc79514905)

[CAPÍTULO VI. CONCLUSIONES 44](#_Toc79514906)

[Bibliografía 45](#_Toc79514907)

# CAPÍTULO I. INTRODUCCIÓN

El presente capítulo presenta una breve descripción sobre el área de investigación del trabajo, se contextualiza y se define el problema a resolver, además, se plantean tanto el objetivo general como los específicos, planteando una posible solución al problema. Por otro lado, se mencionan los beneficios del trabajo resaltandola contribución del mismo, se define el alcance del trabajo así como las limitaciones del mismo que impiden ir más allá del objetivo definido.

## 1.1 Antecedentes

Con el incremento de los vehículos en carretera, el control de tráfico se ha convertido en un gran reto hoy en día. La tasa de vehículos crece exponencialmente diariamente, solo en el 2010 había más de mil millones de vehículos. Con esto, los accidentes aumentaron causando aproximadamente 1.25 millones de muertes alrededor del mundo en 2010 [1].

En Estados Unidos se perdieron 6.9 mil millones de horas en carreteras y 3.1 mil millones adicionales de galones de gasolina debido a embotellamientos, lo cual representó una pérdida anual de 160 mil millones [2]. En Europa alrededor de 40 mil fallecieron y 1.7 millones fueron heridos en accidentes de tráfico [3]. Para tratar de resolver los problemas como los anteriores, las compañías, instituciones académicas, y agencias gubernamentales han incorporado tecnología de comunicación en vehículos y en infraestructura del transporte [4].

Se han integrado las comunicaciones vehículo a vehículo (V2V), vehículo a infraestructura(V2I), vehículo a peatones (V2P) y vehículo a red (V2N), en conjunto denominadas comunicaciones V2X. V2X junto con las capacidades de detección de vehículos provee un soporte para aplicaciones relacionadas con la seguridad, infoentretenimiento para pasajeros y optimización del tráfico de vehículos [4]. En la Figura 1 se muestra las distintas formas de V2X [5].



Figura 1. Una ilustración simple de las comunicaciones V2X [5].

V2V es una tecnología que presenta una comunicación en red para vehículos, de tal manera que puedan estar conectados entre sí. Fue desarrollado y demostrado en 2005 por General Motors y promete una comunicación confiable y fuerte entre usuarios [6]. V2V puede ser usada para mandar advertencias al conductor en situaciones consideradas críticas y de alto riesgo (embotellamiento, cambio de carril) o mandar mensajes de ayuda después de un accidente de tráfico severo [7, 8].

V2V incorpora las siguientes tecnologías relacionadas con la seguridad [9]:

* Sistema de prevención de colisiones.
* Sistema de advertencia al conductor.
* Sistema de estabilidad del vehículo.
* Asistencia automatizada de frenado de emergencia.

Algunas de las aplicaciones son las siguientes [10]:

* Advertencia de colisión frontal.
* Advertencia de no pasar.
* Advertencia de punto ciego.
* Advertencia de cambio de carril.
* Luz de freno de emergencia electrónica.
* Asistente de giro a la izquierda.

Para soportar las comunicaciones V2X se han creado dos soluciones: Comunicaciones dedicadas de rango corto (Dedicated Short Range Communications por sus siglas en inglés DSRC) y tecnologías de redes celulares [2]. DSRC consiste en transceptores de radiocomunicación en los vehículos y unidades colocadas en los laterales de la carretera. Fue desarrollada para cumplir con los requerimientos de un ambiente dinámico como alta movilidad, baja latencia, y un rango considerado bueno de comunicación [8].

DSRC tiene 70 MHz en la banda de 5.9 GHz y es la tecnología por defecto en las aplicaciones de seguridad en V2V porque presenta una latencia menor a otras existentes, se basa en el estándar IEEE 802.11p y se espera que todos los nuevos vehículos la soporten, de acuerdo con la administración nacional de seguridad del tráfico en carreteras (National Highway Traffic Safety Administration, por sus siglas en inglés NHTSA) [11, 12].

Las tecnologías celulares incluyen tanto LTE (Long Term Evolution), 5G como los estándares antiguos. Con respecto a DSRC, estas tecnologías presentan un conjunto de ventajas como: mayor cobertura, infraestructura, seguridad, garantías de QoS (Quality of service) y escalabilidad robusta. Sin embargo, esas ventajas conducen a una arquitectura centralizada, latencia significativa de un extremo a otro, dependencia de la conectividad con la infraestructura y un costo mayor por el uso de la red [13].

Existen algunos retos en las comunicaciones vehiculares principalmente por el comportamiento estocástico del canal. El estándar IEEE 802.11p utiliza OFDM para DSRC, de tal manera que reduce el desvanecimiento que produce el movimiento de las terminales de comunicación y los objetos en el medio ambiente [14]. OFDM es un esquema de modulación multiportadora, los símbolos se transmiten en paralelo en múltiples subportadoras. La técnica reduce el ancho de banda en la subportadora y logra robustez contra el desvanecimiento en frecuencia, debido al retardo de las señales en las diferentes trayectorias de propagación [15].

Una de las actividades importantes a realizar en cualquier sistema de comunicación, como en un sistema SISO-OFDM para comunicaciones V2V, es la detección de símbolos en el receptor. El detector ML (Maximum-likelihood por sus siglas en inglés ML) presenta la mínima tasa de error de bit. Sin embargo, tiene una dependencia exponencial en la complejidad, por lo tanto, se necesitan algoritmos que alcancen tasas de error similares a la detección ML [16].

Para el desarrollo de tecnología en V2V es necesario la implementación de nuevos algoritmos de detección en plataformas que permitan su construcción física lo más rápido y eficiente posible con el objetivo de medir su viabilidad. El FPGA (Field- Programmable Gate arrays por sus siglas en inglés FPGA) es una tecnología en uso que provee grandes ventajas para implementar algoritmos y prototipos a nivel de hardware [17].

Un FPGA consiste de un arreglo de bloques lógicos programables de diferentes tipos, lógica general, memoria y bloques multiplicadores. Alrededor del arreglo se encuentran los bloques de entrada y salida que permiten la comunicación hacia el mundo exterior como se muestra en la Figura 2.



Figura 2. Estructura básica de un FPGA [18].

Algunos problemas en trabajar al nivel de registro de transferencia (Register Transfer Level, RTL) utilizando lenguajes de descripción de hardware como VHDL o Verilog para su posterior síntesis en un FGPA, radica en que no son tan poderosos como los lenguajes de alto de nivel. Esto provoca códigos largos, incremento en la probabilidad de errores y un gran consumo de tiempo al hacer modificaciones en el diseño. Por los problemas anteriores, las herramientas de síntesis de alto nivel (High-Level Synthesis, HLS) son utilizadas [19].

HLS crea un código en un lenguaje de descripción de hardware tomando la descripción del algoritmo o sistema hecho en un lenguaje de alto nivel como C o C++ y directivas que describen restricciones de la arquitectura [20]. El flujo de diseño se muestra en la Figura 3 [21, 22].



Figura 3. Flujo de diseño en Vivado [1].

## 1.2 Planteamiento del problema

Los sistemas OFDM sufren de un fenómeno llamado interferencia inter-portadora (inter-carrier interference, ICI), el cual se presenta en mayor medida en la comunicación V2V por las altas frecuencias de dispersión Doppler producidas por la alta movilidad en los enlaces vehiculares. El rendimiento general de los sistemas OFDM-V2V es reducido debido a la presencia de ICI provocando que la tarea de detección de datos sea agotadora [23].

La detección de datos es un proceso que conlleva una complejidad computacional importante en el receptor, la factibilidad de cualquier sistema diseñado para funcionar en canales V2V estará comprometida con el grado de complejidad del detector utilizado. El detector Near-ML es de baja complejidad y mejora el desempeño de BER [24], sin embargo, no existen arquitecturas de hardware que permitan su implementación en un FPGA.

De lo anterior, se formula la siguiente pregunta de investigación, ¿será posible diseñar una arquitectura digital que permita la implementación del detector Near-ML y de esta manera obtener sus métricas de desempeño?

## 1.3 Objetivo

Desarrollar una arquitectura de hardware del algoritmo Near-ML para la detección de símbolos en el receptor de un sistema SISO-ODFM en ambientes vehiculares V2V, utilizando herramientas de síntesis de alto nivel y un lenguaje de descripción hardware, con la finalidad de conocer la viabilidad de incorporarlo a los estándares de comunicación actuales.

Objetivos específicos:

* Elegir el algoritmo de ordenamiento más eficiente para ordenar un total de 48 datos.
* Obtener una arquitectura para el receptor del sistema SISO-ODFM en una plataforma HLS y a través del lenguaje de descripción de hardware verilog.
* Validar la funcionalidad del algoritmo, corroborando su correcto funcionamiento a través de instrumentos de medición.
* Evaluar la precisión del algoritmo de detección para el sistema SISO-OFDM. utilizando la relación señal-ruido de cuantificación SQNR (*Signal-to-Quantization-Noise Ratio*).

## 1.4 Hipótesis

Es posible implementar una arquitectura digital de un detector de símbolos bajo el esquema de modulación OFDM y criterio de decisión Near-ML utilizando un FPGA y herramientas de síntesis de diseños en HDL.

## 1.5 Justificación

En la literatura sólo se han reportado resultados de simulación sobre el rendimiento del algoritmo, por lo tanto, no existe una investigación que haya implementado en hardware el algoritmo Near-ML y obtenido métricas de eficiencia. Con esto, se podrá corroborar la factibilidad y la viabilidad de ser introducido en las comunicaciones vehiculares y en los distintos estándares de comunicaciones inalámbricas móviles contemporáneos.

Este trabajo tendrá un impacto en una de las áreas con mayor crecimiento en los últimos años, las comunicaciones vehiculares. Normalmente, los receptores en V2V requieren implementaciones en hardware eficientes, que operen a mayores velocidades de transmisión, alta calidad de servicio y sobre todo de baja complejidad. Una forma de cumplir con lo anterior es implementar mejores y nuevos algoritmos como el que se presenta en esta tesis.

Finalmente, una vez terminado el trabajo se podría continuar con la implementación del algoritmo en un circuito integrado de uso comercial, con el fin de agregarlo a las aplicaciones V2V actuales y emergentes. En consecuencia, la complejidad y el consumo de potencia en los equipos de recepción del vehículo disminuirá.

## 1.6 Delimitaciones

Este trabajo abarca únicamente la implementación del algoritmo en una plataforma de desarrollo de hardware y su correspondiente validación. No se tocará el tema de diseñar y simular el algoritmo debido a que fue un trabajo previo de un ex alumno de maestría.

## 1.7 Limitaciones

Los recursos institucionales como el laboratorio no están disponibles por lo que podría poner en riesgo la etapa de validación física del algoritmo. Otra limitación es el tiempo, una parte del trabajo será realizada en C++ en una plataforma que soporta la síntesis de programas en alto nivel y la siguiente parte en un lenguaje de descripción de hardware.

# CAPÍTULO II. MARCO TEÓRICO

En este capítulo se abordan los conocimientos generales para la correcta comprensión del presente trabajo. Asimismo, se mencionan aquellos trabajos actuales similares al área de estudio planteada en esta investigación.

## 2.1 El estándar IEEE 802.11p

La comunicación V2V fue estandarizada con el nombre de comunicaciones dedicadas a corto alcance DSRC IEEE 802.11p en 2010 [25]. El estándar hace uso de una capa física del estándar IEEE 802.11a diseñado para interiores y baja movilidad, cambiando el ancho de banda de la señal a 10MHz, frecuencia central a 5.9 GHz y manteniendo el esquema de multiplexación por frecuencias ortogonales. Debido a que el estándar 802.11a fue diseñado para entornos internos, los cuales son relativamente estacionarios, el rendimiento de los sistemas V2V no cumplen con el desempeño deseable por su alta movilidad presente entre transmisor y receptor.

Una de las principales diferencias radica en que el preámbulo del 802.11p es distinto del 802.11a en la duración del símbolo con una duplicación de 16 µs a 32 µs. En la Figura 4 se muestra su trama transmitida, la cual consta de un preámbulo de dos símbolos de entrenamiento, un campo de la señal y de datos. La sincronía del sistema se realiza con los primeros 10 símbolos cortos de entrenamiento de 1.6 µs de duración. La estimación del canal y sincronización final se realizan con dos símbolos de entrenamiento largos de 6.4 µs de duración. La última parte de la trama es para el campo de la señal que contiene información sobre longitud de la trama, modulación y esquemas de codificación utilizados en la carga útil.



Figura 4. Trama del estándar 802.11p [2].

## 2.2 Pilotos en el estándar 802.11p

El escenario de alta movilidad en los ambientes V2V producen una respuesta al impulso del canal variante en el tiempo. Por cada símbolo en OFDM que componen cada trama transmitida se presentan variaciones temporales de los coeficientes del canal. Existen 64 subportadoras por símbolo OFDM, incluyendo 48 subportadoras de datos, 4 subportadoras pilotos en los índices -21, -7 , 7 y 21 como se observa en la Figura 5.



Figura 5. Asignación de pilotos en 802.11p [2].

En [2] se demuestra que la asignación de pilotos anteriormente descrita, no es la mejor para la estimación de un canal V2V. Sin embargo, al usar receptores con esquemas de estimación de canal iterativos y detecciones basadas en la técnica de mínimo error cuadrático medio (MMSE) se logran rendimientos aceptables después de la tercera iteración [26].

## 2.3 Canal de comunicación V2V

Existen una gran cantidad de investigaciones para clasificar y modelar los canales V2V. Algunas clasificaciones son expuestas en [27, 28, 29, 30, 31, 32, 33]. En estos dos últimos trabajos, se expone la clasificación de escenarios V2V posibles, dentro de los cuales destacan los siguientes: V2V-Autopista con dirección opuesta, V2V-Urbano en avenida principal con dirección opuesta, V2V sub-urbano, V2V en autopista. Lo anterior se resume en la Tabla I, publicada por [9].

**Tabla I. Características de los diversos escenarios V2V**

|  |  |  |  |
| --- | --- | --- | --- |
| **Escenario** | **Velocidad**  **(Km/h)** | **Desplazamiento**  **Doopler (Hz)** | **Retardo**  **máximo (***µ***s)** |
| V2V autopista,  dirección opuesta | 104 | 1000-1003 | 0.3 |
| V2V urbano avenida  principal, dirección opuesta | 32-48 | 300 | 0.5 |
| V2V autopista | 104 | 600-700 | 0.4 |
| V2V urbano avenida principal, misma  dirección | 32-48 | 400-500 | 0.4 |
| V2V suburbano  calle, misma dirección | 32-48 | 300-500 | 0.7 |
| V2V autopista,  misma dirección | 104 | 900-1150 | 0.7 |

Tomando como referencia lo expuesto en los diversos trabajos de la literatura, se concluye que un modelo de canal simple es insuficiente para modelar con precisión el entorno adverso de un canal V2V. Por este motivo, se usan canales estacionarios basados en el modelo TDL (Tapped delay Line) obteniendo la variación presente en los canales V2V. La respuesta al impulso del canal varía temporalmente en el periodo de símbolo OFDM, tal variación en la causante de la interferencia entre portadoras (ICI) degradando el rendimiento de la modulación OFDM [33].

## 2.4 Sistemas de comunicación V2V

En la Figura 6 se muestra la estructura básica de un sistema de comunicación V2V, en la cual se basan gran parte de trabajos presentes en el estado del arte. Las etapas de estimación y ecualización representan dos de los procesos más relevantes durante la recepción. El desempeño general de un sistema V2V va estar estrechamente relacionado al que tengan las dos etapas previamente mencionadas.



Figura 6. Diagrama a bloques de un sistema V2V típico [3].

Con la finalidad de afrontar y mitigar los inconvenientes debido a la movilidad del entorno adverso V2V, se han presentado una gran cantidad de trabajos que abordan la problemática desde diversos enfoques tales como:

* Algoritmos de asignación de pilotos.
* Estimadores de canal en el dominio temporal.
* Estimadores de canal en el dominio frecuencial.
* Detectores lineales.
* Detectores no lineales.

En el estado del arte es conocido que la estimación de parámetros en el receptor se complica en un sistema V2V, debido a la ICI (Interferencia entre portadoras) que afecta a las subportadoras piloto, las cuales son necesarias para la correcta estimación del canal, tal como se describe a detalle en [3] y [11]. Por otro lado, los receptores iterativos tales como [3] y [11] requieren al menos 5 iteraciones para alcanzar un rendimiento BER aceptable. El estimador de canal reportado en [34] es adecuado para operar en canales altamente variantes. Las investigaciones llevadas a cabo en [13] y [14], presentan sistemas con estimadores de datos adecuados para contrarrestar la ICI, logrando un mejor rendimiento que los receptores normalmente usados. Sin embargo, se realizan modificaciones en la capa física del estándar 802.11p, lo cual introduce incompatibilidades.

## 2.5 Detección de datos

Uno de los procesos con mayor grado de complejidad computacional en el receptor es la detección de datos, la factibilidad de un sistema diseñado para funcionar en tiempo real en un canal V2V dependerá principalmente del grado de complejidad del detector utilizado. El detector ML conocido como el detector optimo tiene una complejidad de ) [35]. Se puede apreciar que la complejidad aumenta de forma abrupta con el tamaño de la constelación Ω y número de subportadoras de datos . En consecuencia, no resulta viable implementar tal detector. Actualmente existen una serie de detectores con menor complejidad y un rendimiento parecido al detector ML. Algunos de ellos han sido diseñados para el sistema de espacio-tiempo de laboratorios Bell vertical(Bell Laboratories Layer Space-Time, V-BLAST) [36]. Uno de los más importantes y conocidos es el detector esférico [37].

Detectores no lineales basados en el algoritmo M en conjunto con la descomposición QR de la matriz de canal, ha comenzado a tener un gran auge [38], [39]. Recientemente, una gran contribución dentro de este, los detectores OSIC,QR-ML y V2V Near ML. Los resultados de simulación han comprobado que son adecuados para mitigar la ICI y obtienen desempeños en términos de BER favorables [1].

### **2.5.1 Modelo del receptor**

Es posible representar la señal del -ésimo símbolo que llega al receptor en su representación compleja banda base con la convolución discreta:

|  |  |
| --- | --- |
|  | ( 1 ) |

Donde , , representa la longitud de la respuesta al impulso (CIR), representa la CIR del -th bloque en el instante para un impulso de entrada en las muestras pasadas, el -ésimo símbolo OFDM cuyo tamaño es , por último es el ruido aditivo Gaussiano blanco (additive White Gaussian noise, AWGN) complejo.

La ecuación (1) puede ser vista en su forma matricial como:

|  |  |
| --- | --- |
|  | ( 2 ) |

donde:

,

,

el ruido es de media cero y varianza . Los coeficientes del CIR del canal son representados por la matriz de dimensión de la siguiente forma:

|  |  |
| --- | --- |
|  | ( 3 ) |

donde , representa el operador modulo , la CIR es considerada como cero para . Para obtener el símbolo OFDM en el dominio de la frecuencia (FD) con CP removido, se multiplica la ecuación (2) por la matriz de transformada discreta de Fourier (DFT) normalizada:

|  |  |
| --- | --- |
|  | ( 4 ) |

la multiplicación resulta en:

|  |  |
| --- | --- |
|  | ( 5 ) |

es el símbolo OFDM recibo en el FD y es la DFT del vector de ruido. Utilizando las propiedades de la matriz ortogonal; y , donde representa la matriz identidad de tamaño , entonces (5) puede ser formulada como:

|  |  |
| --- | --- |
|  | ( 6 ) |
|  | ( 7 ) |

donde se compone por el vector de tamaño , símbolos pilotos y símbolos de guarda. La matriz de canal en frecuencia (CFM) contiene información en dominio de la frecuencia del canal y la frecuencia Doopler de la representación circulante y dispersa de la CIR variante en el tiempo.se convierte en una matriz diagonal cuando la propagación Doppler es insignificante, lo cual significa un sistema libre de ICI, sin embargo, los ambientes V2V presentan alta movilidad tanto en el transmisor y receptor, ocasionando una dispersión Doppler significativa. Lo anterior convierte a en una matriz dispersa generando un sistema afectado por la ICI.

Debido a la selectividad del canal V2V, los sistemas OFDM son susceptibles a errores de detección, la potencia local de algunas subportadoras puede ser baja por causa de desvanecimientos profundos en la función de transferencia del canal imposibilitando la detección del dato transmitido en tales soportadoras. La técnica como transformada discreta de Fourier-extendida (Discrete Fourier Transform-Spreading, DFTS) es usada para reducir esta problemática.

Multiplicando la matriz de Fourier F con el vector de símbolo OFDM transmitido es posible incorporar la dispersión DFT de los datos en el modelo de señal (7). Lo anterior, puede ser descrito con la siguiente ecuación:

|  |  |
| --- | --- |
|  | ( 8 ) |

donde es el vector de datos recibido, la CFM de rango reducido , el vector de datos transmitido con recodificación lineal (Linear Precoding, LP), el vector de ruido; cada termino muestreado en las posiciones correspondientes de las subportadoras de datos.

### **2.5.2 Detector OSIC**

Utilizando la descomposición V2V Sorted QR y el detector de cancelación de interferencia sucesiva ordenada (Ordering successive interference cancellation, OSIC) es posible implementar un detector subóptimo de baja complejidad, para un sistema multiportadora V2V con LP presenta un excelente desempeño en términos de BER. La descomposición V2V Sorted QR de la matriz de canal consiste en calcular una matriz triangular superior , una matriz ortogonal de norma unitaria y una matriz de permutación , de tal manera que ,donde es la matriz con sus columnas ordenadas de acuerdo a . Se obtiene un modelo reducido de la ecuación (8) :

|  |  |
| --- | --- |
|  | ( 9 ) |
|  | ( 10 ) |
|  | ( 11 ) |

Debido a que es unitaria las estadísticas del ruido no son alteradas. El nuevo modelo representado en la ecuación (11) es adecuado para la aplicación del detector OSIC. La estructura triangular de la matriz permite calulcar el -ésimo elemento de como:

|  |  |
| --- | --- |
|  | (12) |

### **2.5.3 Detector QR-ML convencional**

La detección QR-ML convencional puede ser incorporado al modelo del sistema receptor anteriormente mostrad como:

//definir 3.18 y 3.19

|  |  |
| --- | --- |
|  | ( 3 ) |

Para encontrar el símbolo estimado \_ de la solución ML en () se basa en la construcción de un árbol binario de búsqueda como se muestra en la figura x. Con el fin de calcular (3) se define la métrica de distancia para cada rama del árbol.

//Definir distancia 3.20

La distancia di representa el valor de rama de un nodo s^i que tiene a los nodos …. como sus nodos antecesores. La distancia de un nodo que se encuentra en el k-ésimo nivel y el nodo de la raíz se define como la distancia acumulada , esto es, la suma de todas las distancias de las ramas desde el nodo raíz hasta el nodo indicado. En un nivel , la distancia acomulada se obtiene con la siguiente ecuación.

// Definir ecuación 3.21

Con base a la ecacuin (), la detección obtina del vector s^ ebs la ruta que minimice a (), cuando .



Figura 7. Estructura del árbol de búsqueda bajo el criterio ML.

### **2.5.4 Detector V2V Near ML**

El detector V2V Near ML incorpora el algoritmo adaptativo a la detección QR-ML convencional, el cual realiza ajustes durante el proceso de búsqueda del árbol binario con el fin de reducir la complejidad computacional. Al igual que el detector OSIC hace uso de la descomposición Sorted-QR. Como se muestra en la figura x el símbolo se encuentra en el nodo raíz del árbol, y los nodos que surgen de el a través de las ramas son solución posible para . El algoritmo M consiste en seleccionar en cada nivel del árbol binario de búsqueda un máximo de candidatos para la detección del simbolo del vector s ^ estimado, descartando nodos restantes del nivel actual.



Figura 8. Estructura del árbol de búsqueda bajo el criterio ML con algoritmo M.

Cada rama del árbol de búsqueda contiene una métrica de distancia definida por:

//definir ecuación 3.22

Se selecciona M nodos que mantienen una distancia menor entre cada nodo del nivel y el nodo de la raíz, por lo tanto, al realizar la detección se obtiene M rutas posibles. La distancia correspondiente a cada ruta se define como :

//definir 3.23

la solución S^, esta dada por la ruta que cumpla con:

//definir 3.24

El algoritmo Near ML genera detectores de baja complejidad para valores de pequeños,sin embargo, subóptimos en términos de tasa de error de bit, a medida que el valor de incremente, el desempeño del algoritmo se acerca al del detector ML, con la penalidad del incremento en complejidad computacional.

## 2.6 Modelo algorítmico

En [poner de referencia tesis de Romero] se propone un modelo algoritmo que permite realizar la implementación de un algoritmo de forma eficiente en términos de velocidad o consumo de área. Además, los diseños pueden ser replicados de manera sistemática por cualquier persona lo necesite.

### 2.6.1 Mapeo algoritmo a arquitectura

De manera general, el procedimiento se resume al diagrama de flujo de la Figura 9.



Figura 9. Algoritmo a arquitectura.

### 2.6.2 Modelo de oro

Se genera un modelo que sirva de referencia para el algoritmo a implementar, de tal manera que se pueda evaluar el rendimiento de la implementación. Para lograr esto, se programa el algoritmo bajo estudio empleando de preferencia un lenguaje de alto nivel, en aritmética de punto flotante y con sus ecuaciones originales.

### 2.6.3 Adecuaciones al algoritmo

En este paso se modifica la secuencia de pasos o sus respectivas ecuaciones del algoritmo con el fin de reducir su complejidad sin alterar la funcionalidad general. Es importante recalcar que no existe una metodología para aplicar las adecuaciones del algoritmo, solo el entendimiento profundo y experiencia servirá para identificar las posibles adecuaciones. A continuación, se muestran algunas adecuaciones.

* Transformaciones o cambios de dominios.
* Sistolizacion de operaciones de cálculo intensivo.
* Remplazo de divisiones y multiplicaciones con operaciones como sumas y desplazamientos.
* Simplificación o propiedades matemáticas.
* Factorizaciones o aproximaciones matemáticas.

### 2.6.4 Análisis de punto fijo

Utilizando la representación de punto fijo es posible alcanzar desempeños cercanos a los obtenidos con la representación en punto flotante, por lo cual, esta última es comúnmente descartada en las arquitecturas de procesamiento de señales(PDS). Además, la representación en punto fijo reduce el tiempo de procesamiento y la complejidad de la implementación por ende el consumo de potencia y área de silicio.

El objetivo que se persigue al hacer un análisis de punto fijo es encontrar una longitud y formato de palabra que permita tener un desempeño similar al modelo de oro el cual emplea la representación en punto flotante. Con el parámetro de desempeño SQNR se puede cuantificar el efecto de pasar un algoritmo a punto fijo y de esta manera realizar una comparación con el modelo de referencia.

A continuación, se muestran los pasos a seguir para realizar el análisis de punto fijo:

* Medición del rango dinámico: El rango dinámico de una variable es la diferencia entre el valor máximo y mínimo que puede tomar durante la ejecución del algoritmo. Es importante obtener el rango dinámico para cada variable en punto flotante para después asignar una cantidad de fija de bits con su parte entera y fraccionaria.
* Selección del formato de palabra: La cantidad de bits para el parte entrera (IP) de una variable se relacionada con su rango dinámico con la siguiente expresión:

|  |  |
| --- | --- |
|  | ( 13 ) |

donde y son los valores máximo y mínimo del rango dinámico de la variable en punto flotante. Para la parte fraccionaria se define la resolución de una variable en punto fijo, puede ser expresada como:

|  |  |
| --- | --- |
|  | ( 14 ) |

De 14 se despeja FP que representa la cantidad de bits en la parte fraccionaria para una resolución en específico, se obtiene que:

|  |  |
| --- | --- |
|  | ( 15 ) |

La resolución de la parte fraccionaria debe de ser escogida con base a las necesidades del sistema porque una mayor resolución generara un aumento en la anchura de la palabra (WL).

* Cuantificación de la relación señal a ruido de cuantización (SQNR): Cuando se realiza el proceso de cuantificación de un valor en punto flotante también se genera un error que puede ser representado como ruido agregado al valor original, por lo tanto, el SQNR es el parámetro para medir el nivel de calidad del proceso de cuantificación de una variable en su representación de punto flotante. La mayoría de los algoritmos operan con variables que representan matrices y vectores, por lo tanto, el SQNR se expresa como un promedio:

|  |  |
| --- | --- |
|  | ( 16 ) |

donde es la variable original en punto flotante, es la representación de en punto fijo y es la cantidad de elementos que conforman a la variable. Cuando una variable tiene un SQNR considerado bajo, se ajusta la longitud su longitud de palabra y se sacrifican bits de aquellas palabras cuyo SQNR es considerado alto. El objetivo es conseguir un SQNR en la salida del algoritmo equilibrado con las variables que conforman el algoritmo.

* Efectos del redondeo: Una desventaja que se presenta al pasar un numero en su presentación en punto fijo es que no todos los valores pueden ser representados de manera exacta, el método del redondeo es utilizado para obtener un valor representable a pesar de la perdida de precisión. Existe una gran variedad de tipos de redondeo, los más utilizados son el truncamiento, redondeo hacia cero, redondeo hacia más menos infinito y redondeo al más cercano. El tipo de redondeo a utilizar en el algoritmo tendrá un efecto importante en el desempeño en términos de SQNR, área y velocidad de procesamiento.

# CAPÍTULO III. MÉTODO

El objetivo de este capítulo es describir la metodología empleada para el desarrollo de este trabajo, la cual consta de los sujetos u objeto de estudio, el procedimiento que desglosa las etapas del proyecto en actividades realizadas para lograr el objetivo planteado y por último, se muestra una lista de materiales y herramientas necesarias para el desarrollo del trabajo.

## 3.1 Sujeto

El objeto de estudio es un bloque para la detección de símbolos de esquema de transmisión SISO-OFDM para ambientes vehiculares cuya implementación se realiza en un FPGA de la compañía Xilinix. Este trabajo es desarrollado por un alumno de maestría y asesorado por un investigador del Dpto. de Ingeniería Eléctrica y Electrónica.

## 3.2 Procedimiento

En la Figura 10 se muestra el procedimiento a seguir para cumplir los objetivos.



Figura 10. Diagrama de flujo del procedimiento.

**Definición del problema:** Se describen los antecedentes, objetivos, problema a resolver, hipótesis, alcances y limitaciones del proyecto.

**Investigación bibliográfica:** Se revisan documentos como artículos, libros y revistas cuya publicación sea reciente, con el objetivo de conocer el estado de arte de los detectores en sistemas SISO implementados en un PFGA y de esta manera identificar las áreas de oportunidad.

**Determinar el mejor algoritmo de ordenamiento para 48 y 64 datos:** Se implementan los algoritmos más relevantes en Matlab con el fin de simular el ordenamiento de los datos y obtener su rendimiento en términos de tiempo.

**Estudio de técnicas de diseño de hardware:** Se estudian los métodos de diseño e implementación en hardware.

**Modelado del sistema entre el receptor y transmisor:** En MATLAB se modela el sistema completo del transmisor y receptor con la finalidad de comprobar el funcionamiento del detector.

**Conversión de punto flotante a punto fijo:** Después de comprobar el funcionamiento del algoritmo en punto flotante, se convierte a aritmética de precisión finita para simular el comportamiento del algoritmo operando en un sistema digital real con recursos limitados.

**Diseño de la arquitectura de hardware:** La arquitectura propuesta del detector ML se codifica en un lenguaje de descripción de hardware, y se utiliza un simulador basado en software para comprobar el funcionamiento del de la arquitectura diseñada.

**Implementación en FPGA:** Una vez que la simulación de la arquitectura diseñada es exitosa se procede a implementarla en un FPGA.

**Verificación:** Para la verificación se comprueba la frecuencia máxima de operación a la cual la implementación puede manejar, el hardware consumido en el FPGA; Si los resultados no cumplen con las expectativas se regresa a la etapa de diseño de la arquitectura.

**Publicación de resultados:** Cuando la verificación está completada se procede a publicar los resultados en una revista o congreso.

**Escritura del documento final:** Se escribe el documento de tesis para cumplir con el requisito de titulación.

## 3.3 Materiales y Herramientas

A continuación, se muestran los materiales y herramientas necesarios para cumplir con el desarrollo de la investigación.

Herramientas:

* MATLAB
* Vivado 2016.1
* Laptop Acer con procesador Intel core i5-8265U, 8GB DDR4, 256GB PCIe NVMe SSD.

Materiales:

* Tarjeta de desarrollo Nexys 4
* Bases de datos institucionales de revistas internacionales.

# CAPÍTULO IV. DESARROLLO

En este capítulo se realiza el procedimiento para obtener el modelo del algoritmo Near-ML en un lenguaje de alto nivel y el análisis de punto fijo mostrado en la sección 2.6 que permitió dimensionar la longitud de los buses que conforman la arquitectura final. Además, se presenta cada sub bloque digital, la conexión entre ellos y su correspondiente parametrización.

## 4.1 Desarrollo del modelo algorítmico del sistema

El detector Near-ML para comunicaciones vehículo a vehículo mostrado de manera analítica en la sección 2.5.4 fue modelado utilizando la herramienta computacional MATLAB, posteriormente fue incorporado a un sistema completo de comunicaciones SISO V2V en punto flotante. Una vez comprobado su funcionamiento es convertido en su representación en punto fijo para obtener su desempeño en términos de BER.

### **4.1.1 Modelo de oro del detector**

En la Figura 11 y la Figura 12 describen el algoritmo Near-ML a nivel de pseudocodigo propuesto en [2], se trata de un algoritmo recursivo que tiene como entradas la matriz R proveniente de la descomposición QR, el vector que contiene los datos que llegan al receptor, el vector order, el índice M que varía con base al tipo de modulación(QPSK,QAM-16,QAM-64), el tamaño de la constelación y la cantidad de subportadoras de datos de acuerdo con el esquema OFDM.

Las siguientes consideraciones son tomas en el pseudocódigo de la Figura 11 y la Figura 12:

* Variables con formato de negrita y en mayúscula representan matrices.
* Variables con formato de negrita y en minúscula representan vectores.
* Todas las variables son globales.
* El subíndice en las variables indica que se está refiriendo a un vector o una fila de una matriz.

Entrada: **R**, , **order**, **M**, ,

Salida: Símbolo estimado .

detectorNearML(**R**, , **order**, **M**, , )

1. = ∞

2. skip = true

3. **level** = { }

4. **S** = { }

5. =

6.  *=*  para todos los símbolos

7. , en orden ascendente

8. = , en orden ascendente

9. nearml()

10. (**order**) = **sest** (símbolo estimado)

11. devolver

Figura 11. Detector Near ML.

nearml()

1. **for** (k = 1 to ) **do**

2. **if**(skip = true **and** > ) **then**

3. **break**

4.

5. = para todo

6. +

7. = , en orden ascendente

8. **if** (skip = true **and** > ) **then**

9. **break**

10. **if** (n > 1) **then**

11. nearml ()

13. **else**

14. if () **then**

15.

16.

17. **for** () **do**

18.

Figura 12. Algoritmo Near ML.

### **4.1.2 Sistema SISO V2V**

Con el fin de corroborar el funcionamiento del algoritmo mostrado en la sección anterior, fue codificado en Matlab e incorporado a un sistema de comunicaciones de una antena transmisora y una antena receptora simulando su operación en ambientes vehiculares reales. En la Figura 13 se muestra el sistema V2V, cada bloque fue seleccionado con base a las especificaciones del estándar IEEE V2V. A continuación, se describe la funcionalidad de manera general de cada bloque que conforma el sistema.



Figura 13. Sistema SISO V2V.

**Codificador convolucional:** Genera códigos lineales que se utilizan para proteger la información añadiendo redundancia a la secuencia de datos a transmitir. Con esto se mejora el desempeño de la detección y corrección de errores de bits en el receptor.

**Interleaving:** El bloque interleaving de correlaciona la secuencia de bits generadas por el codificador convolucional con el fin de disminuir los errores que se puedan producir con bits por ráfaga.

**Modulador QAM:** Selecciona bloques de bits para realizar un mapeo a un símbolo complejo , por cada segundos, donde es el conjunto de la constelación utilizada basada en el estándar 802.11p.

**IFFT:** El bloque IFFT se encarga de realizar la modulación OFDM por cada bloque de 64 muestras correspondientes a un símbolo.

**Intervalo de guarda:** Se agregan un conjunto finito de ceros a la izquierda y derecha de los datos en un símbolo OFDM, esto permite disminuir la interferencia Inter simbólica de las subportadoras adyacentes.

**Inserción de pilotos:** Se insertan un conjunto de elementos conocidos en cada símbolo OFDM cuya potencia suele ser mayor a las portadoras de datos, lo cual permite estimar el canal altamente variante para posteriormente realizar la etapa de detección.

**Canal V2V:** Se generan matrices especiales que simulan los fenómenos físicos que se pueden añadir la señal de interés al pasar por el canal vehículo a vehículo.

**Ruido:** Se genera un modelo de ruido básico conocido como Ruido Blanco Gaussiano Aditivo para imitar el ruido que se agrega a la señal recibida debido a la electrónica utilizada.

**Remoción del intervalo de guarda y pilotos:** Remueve los intervalos de guarda y los pilos para solo dejar las subportadoras de datos.

**Deinterleaving:** Realiza el desentrelazado de la trama.

**FFT:** Realiza la demodulación OFDM.

**Detector:** Realiza la detección de los símbolos transmitidos utilizando el algoritmo Near-ML.

**Demodulador QAM:** Realiza la demodulación de los símbolos QAM estimados provenientes del detector.

**Decodificador Viterbi:** Realiza la corrección de errores de bits de los datos detectados.

### **4.1.3 Análisis de punto fijo**

Con el fin de obtener el rango dinámico de las variables que componen al algoritmo codificado en MATLAB se ejecutó 1000 veces el script en punto flotante guardando el máximo y el mínimo de cada variable para posteriormente elegir aquella con el valor más grande y el más pequeño, esta última variable determinará la longitud de los buses. En la Figura 14 se muestra el resultado del procedimiento anteriormente descrito.

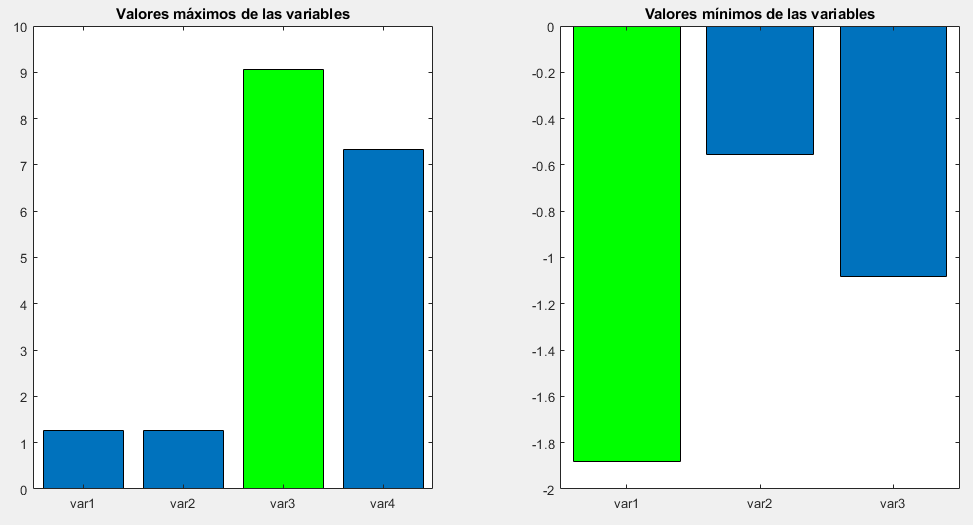


Figura 14. Valores máximos y mínimos de las variables.

Una vez obtenido el valor máximo y el mínimo fueron sustituido en la ecuación ( 13 ) con el fin de obtener los bits de la parte entera que conformará el formato de la palabra como se muestra a continuación:

Se obtiene que para poder representar en punto fijo el valor máximo posible durante la ejecución del algoritmo de detección es necesario por lo menos una parte entera de 6 bits.

Para obtener la cantidad de bits de la parte fraccionaria se usó la ecuación ( 15 ) con una resolución deseada en dB de -60dB en su respectiva conversion en magnitud como se muestra a continuación:

El formato de palabra con base las especificaciones mencionadas en el cálculo de la parte entera y fraccionaria queda de la siguiente forma bits, lo cual es un formato de palabra estándar que facilitará la combinación de los disantos módulos en el diseño de la arquitectura digital.

Con el fin de comprobar el funcionamiento del algoritmo en notación de punto fijo con la anchura previamente calcula se utilizó la herramienta “fixed-point toolbox” de MATLAB, la cual permite convertir variables en punto flotante a objetos que tienen propiedades de punto fijo. En la Figura 15 se muestra las líneas de código que definen el formato de la palaba, la capacidad de almacenar valores negativos y la creación de los objetos “dataFormat” y “dataForma2”. La palabra reservada “numerictype” crea un objeto con las longitudes del formato de palabra que son ingresados como parámetros de la función. La palabra reservada “fimath” devuelve un objeto que contiene características relacionadas con operaciones matemáticas que se hacen entre otros objetos en punto fijo.



Figura 15. Porción de código en MATLAB.

## 4.2 Arquitectura propuesta

En esta sección se describe el módulo principal y los módulos subsecuentes junto con su respectiva parametrización. En el módulo principal se definen los puertos de entrada y salida que servirán te interfaz para la conexión con otros módulos cuando sea integrado en el receptor. La arquitectura se diseñó con una metodología “Top – Down”, la cual consiste en empezar con un módulo principal “TOP” que es el primer nivel de la jerarquía y el más alto. En el segundo nivel de la jerarquía se encuentra sub módulos que componen al módulo principal, de esta misma manera se organizan los módulos del nivel 2 o cualquier otro nivel de la jerarquía.

### **4.2.1 Parametrización**

En la Tabla 1 se muestra los parámetros que se utilizan para personalizar la arquitectura digital. Los parámetros permiten modificar el comportamiento general del detector y obtener distintos comportamientos.

Tabla 1. Parametrización de la arquitectura.

|  |  |
| --- | --- |
| Parámetro | Descripción |
| WL | Longitud de palabra. |
| IP | Cantidad de bits de la parte entera de WL. |
| IF | Cantidad de bits para la parte fraccionaria en WL. |
| NP | Cantidad de subportadoras por cada símbolo OFDM. |
| NV | Cantidad de subportadoras de datos por cada símbolo OFDM. |
| M | Índice de la modulación (BPSK,QAM16,QAM64,QAM264). |

### **4.2.2 Nivel principal**

El nivel principal, también llamado “TOP level” se muestra en la Figura 16 y su respectiva parametrización en la Tabla 2.

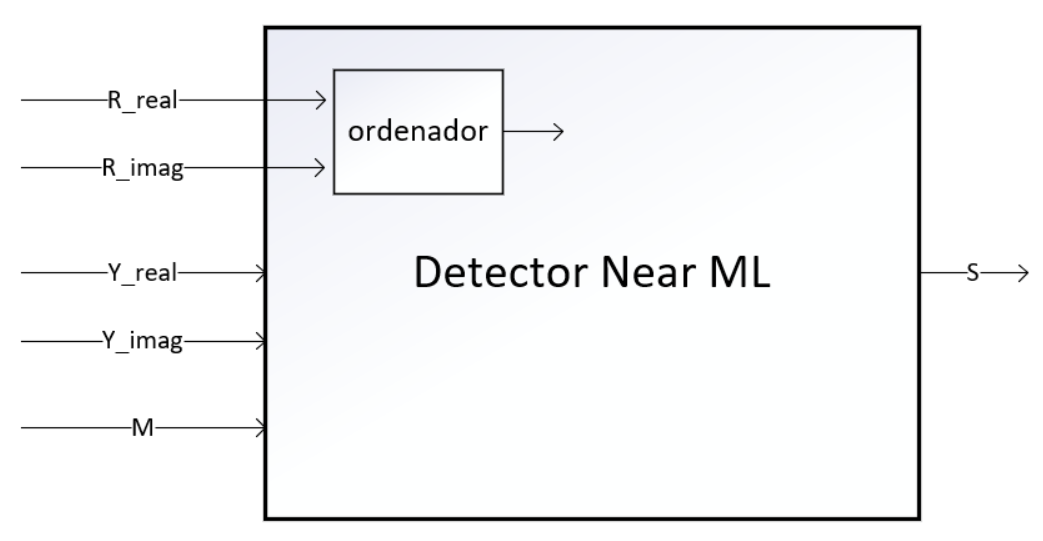


Figura 16.Top level.

Tabla 2. Parametrización del módulo "TOP".

|  |  |  |
| --- | --- | --- |
| Entrada | Longitud en bits | Descripción |
| R\_real | WL | Parte real de la k-ésima  columna de la matriz R. |
| R\_imag | WL | Parte imaginaria de la k-ésima  columna de la matriz R. |
| Y\_real | WL | Parte real del vector recibido. |
| Y\_imag | WL | Parte imaginaria del vector recibido. |
| M | WL | Índice de modulación. |
| Salida | Longitud en bits | Descripción |
| S | Log2(M) | Símbolo detectado |

### **4.2.3 Norma euclidiana**

Dado un numero complejo con parte real a y parte imaginaria b, el bloque de la Figura 17 realiza la norma euclidiana evitando realizar la raíz cuadrada, por lo tanto, las salidas del bloque son valores al cuadrado. La operación que se realiza se puede definir como:

|  |  |
| --- | --- |
|  | (17) |

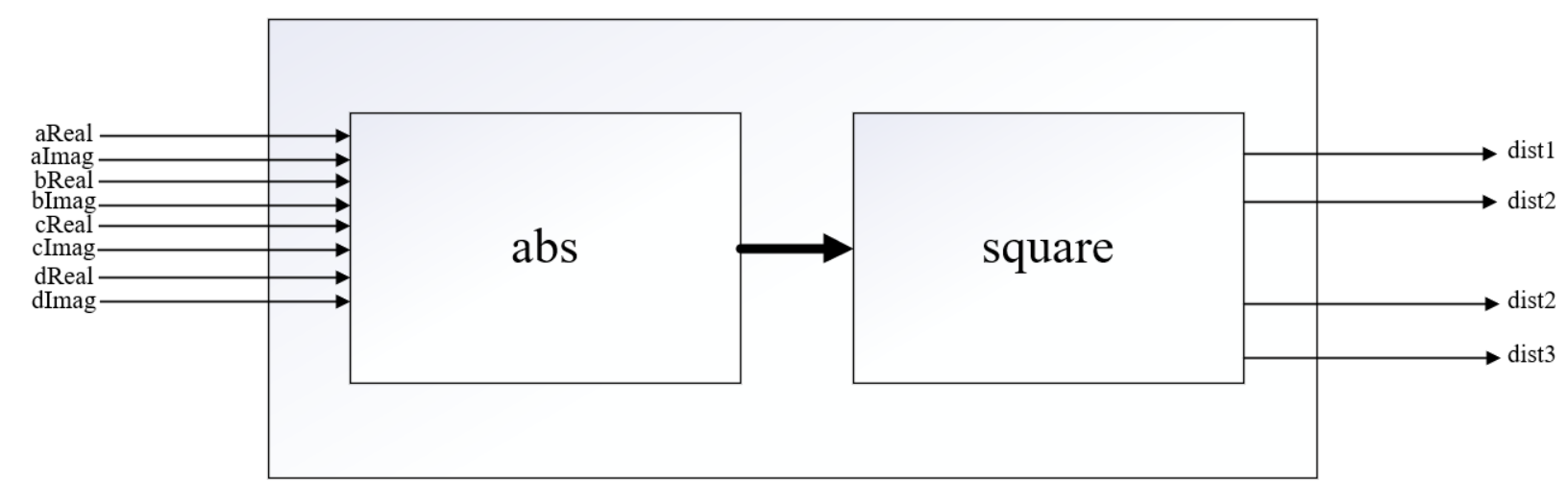


Figura 17. Bloque dist2.

El bloque de la Figura 17 se compone de los siguientes elementos:

* **abs:** Realiza el absoluto de los 4 elementos complejos dados como entrada.
* **square:** Eleva al cuadrado la salida del bloque abs.

Tabla 3. Entradas y salidas del bloque dist2.

|  |  |  |
| --- | --- | --- |
| Entrada | Tamaño (bits) | Descripción |
| aReal,aImag,bReal,bImag  cReal,cImag,dReal,dImag | WL | Parte real e imaginaria de números complejos |
| Salida | Tamaño (bits) | Descripción |
| dist1,dist2,dist3,dist4 | WL | Distancia al cuadrado |

### **4.2.3 Ordenador**

En la Figura 18 se muestra el bloque sorter el cual se encarga de construir conjuntos ordenados de 4, 16, 64 y 256 con cuatro elementos desordenados que son dados como entradas. Una vez se tienen los conjuntos ordenados se seleccionan las cuatro distancias más pequeñas y son multiplexadas a la salida del bloque. La ejecución comienza cuando el sub bloque “Control unit” detecta que la señal “start” se encuentra en un estado lógico “1”, entonces coordina al sub bloque “Mege unit” para formar los conjuntos anteriormente mencionados.

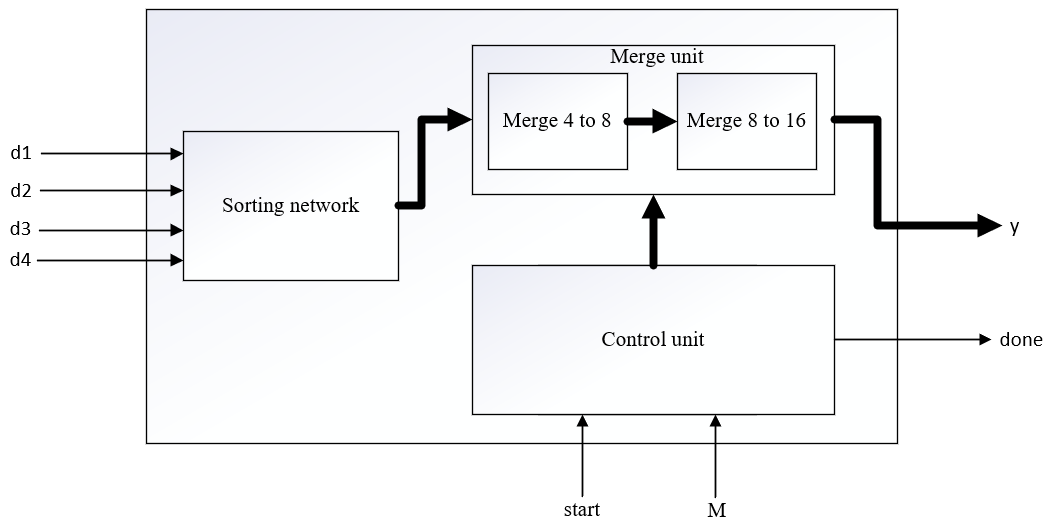


Figura 18. Bloque sorter.

En la Figura 18, se pueden identificar los siguientes componentes:

* **Sorting network:** Circuito combinacional que realiza el ordenamiento de 4 elementos.
* **Merge unit:** Combina conjuntos de 4 elementos para generar 16, 64 o 256 distancias dependiendo del índice de modulación M.
* **Control unit:** Genera líneas de control para el sub bloque Merge unit.

Tabla 4. Entradas y salidas del bloque sorter.

|  |  |  |
| --- | --- | --- |
| Entrada | Tamaño(bits) | Descripción |
| d1,d2,d3,d4 | WL | Distancias a ordenar |
| start | 1 | Señal para empezar |
| M | 2 | Índice de la modulación |
| Salida | Tamaño(bits) | Descripción |
| done | 1 | Bandera que indica la finalización del ordenamiento de datos |
| y | 4\*WIDTH | Elementos ordenados |

# CAPÍTULO V. RESULTADOS

## 5.1 Síntesis

## 5.2 Latencia

## 5.3 Verificación

# CAPÍTULO VI. CONCLUSIONES

En este trabajo, se propuso, diseñó y sintetizó una arquitectura digital implementada en el FPGA Artix-7 del algoritmo de detección de símbolos Near-ML para un sistema de comunicaciones SISO-OFDM en ambientes V2V. Actualmente, no existen implementaciones del algoritmo para operar en sistemas V2V, por lo cual la principal aportación es ir más allá del modelo algorítmico y obtener una arquitectura implementable, así como su consumo de recursos y latencia.

Cada módulo que conforma al detector fue presentado a detalle, describiendo sus entradas, salidas y la relación con otros módulos. Se obtuvo que la implementación en punto fijo tiene un rendimiento similar al algoritmo en punto flotante y la arquitectura no consume una gran cantidad de recursos de hardware, por lo que el bloque detector puede ser incorporado a los receptores de los estandartes actuales sin ningún aumento de complejidad y recursos consumidos importante.

Como trabajo futuro se propone realizar mejoras a nivel algorítmico e implementar optimizaciones y técnicas de paralelismo en cada bloque de hardware con el fin de incrementar el desempeño en términos de BER y latencia del detector.

# Bibliografía

[1] A. D. and S. S. S. Kulkarni, “Proposed framework for V2V communication using Li-Fi technology,” *2017 Int. Conf. Circuits, Control. Commun. (CCUBE), Bangalore,* pp. 187–190, 2017.

[2] H. A. O. and W. Z. K. Abboud, “Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9457–9470, 2016.

[3] B. H. and H. D. S. J. Lianghai, A. Weinand, “Multi-RATs Support to Improve V2X Communication,” *2018 IEEE Wirel. Commun. Netw. Conf. (WCNC), Barcelona*, pp. 1–6, 2018.

[4] Y. Q. and R. Q. H. S. Gyawali, S. Xu, “Challenges and Solutions for Cellular based V2X Communications,” *IEEE Commun. Surv. Tutorials*, 2020.

[5] H.Rosier, “The details of V2X communication.” https://www.intelligent-mobility-xperience.com/the-details-of-v2x-communication-a-872631/.

[6] G. K. and F. G. K. Eshteiwi, K. Ben Fredj, “Performance analysis of peer-to-peer V2V wireless communications in the presence of interference,” *2017 IEEE 28th Annu. Int. Symp. Pers. Indoor, Mob. Radio Commun. (PIMRC), Montr.*, pp. 1–6, 2017.

[7] T. B. and A. Z. D. Boehmlaender, S. Hasirlioglu, V. Yano, C. Lauerer, “Advantages in Crash Severity Prediction Using Vehicle to Vehicle Communication,” *IEEE Int. Conf. Dependable Syst. Networks Work. Rio Janeiro*, pp. 112–117, 2015.

[8] T. Bey and G. Tewolde, “Evaluation of DSRC and LTE for V2X,” *IEEE 9th Annu. Comput. Commun. Work. Conf. (CCWC), Las Vegas, NV, USA*, pp. 1032–1035, 2019.

[9] H. P. Dai Nguyen and R. Zoltán, “The Current Security Challenges of Vehicle Communication in the Future Transportation System,” *IEEE 16th Int. Symp. Intell. Syst. Informatics (SISY), Subotica*, pp. 161–166, 2018.

[10] S. D. Gregory , Yoon Rebecca, Fikentscher Joshua , Doyle Charlene and J. P. Lukuc Mike,Simons Jim, Wang Jing Harding, “Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application,” *Tech Rep. DOT HS 812 014, U.S. Dep. Transp. United States,* 2014.

[11] R. Sabouni and R. M. Hafez, “Performance of DSRC for V2V communications in urban and highway environments,” *IEEE Can. Conf. Electr. Comput. Eng. (CCECE), Montr.*, pp. 1–5, 5245.

[12] A. D. and S. A. J. Thota, N. F. Abdullah, “V2V for Vehicular Safety Applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2571–2585, 2020.

[13] K. O. and S. I. Z. MacHardy, A. Khan, “V2X Access Technologies: Regulation, Research, and Remaining Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 20, pp. 1858–1877, 2018.

[14] Y. P. and K. K. M. S. Hossen, M. Bang, “Performance analysis of an OFDM-based method for V2X communication,” *2014 Sixth Int. Conf. Ubiquitous Futur. Networks (ICUFN), Shanghai*, pp. 238–242, 2014.

[15] T. Maehata et al., “DSRC using OFDM for roadside-vehicle communication system,” *VTC2000-Spring. 2000 IEEE 51st Veh. Technol. Conf. Proc. (Cat. No.00CH37026), Tokyo, Japan*, vol. 1, pp. 148–152, 2000.

[16] R. M. and T. U. Atsushi Okawado, “Near ML detection using Dijkstra’s algorithm with bounded list size over MIMO channels,” *2008 IEEE Int. Symp. Inf. Theory, Toronto*, pp. 2022–2025, 2008.

[17] S. D. and M. Z. V. Kiray, “Improving Digital Electronics Education with FPGA technology, PBL and Micro Learning methods,” *Proc. 2013 IEEE Int. Conf. Teaching, Assess. Learn. Eng. (TALE), Bali*, pp. 445–448, 2013.

[18] Ian Kuon; Russell Tessier; Jonathan Rose, “FPGA Architecture: Survey and Challenges,” 2008.

[19] K. Georgopoulos et al., “An evaluation of vivado HLS for efficient system design,” *2016 Int. Symp. ELMAR, Zadar*, pp. 195–199, 2016.

[20] D. L. and F. M. D. O’Loughlin, A. Coffey, F. Callaly, “Xilinx Vivado High Level Synthesis: Case studies,” *25th IET Irish Signals Syst. Conf. 2014 2014 China-irel. Int. Conf. Inf. Commun. Technol. (ISSC 2014/CIICT 2014), Limerick*, pp. 352–356, 2014.

[21] “ug871-vivado-high-Ievel-synthesis-tutorial.” .

[22] “wp416-Vivado-Design-Suite.” .

[23] J. A. Del Puerto-Flores, J. Cortez, C. A. Gutiérrez, C. Del Valle-Soto, R. Velázquez and L. J. Valdivia, “Performance of MRC Detection in OFDM System with Virtual Carriers over V2V Channels,” *2019 IEEE 39th Cent. Am. Panama Conv. (CONCAPAN XXXIX), Guatemala City, Guatemala*, pp. 1–6, 2019.

[24] Hector Eduardo Aldrete Vidrio, “Sistema de comunicación multiportadora para el estándar 802.11p utilizando precodificación frecuencial y cancelación no lineal de interferencia,” 2019.

[25] “IEEE Draft Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) specif.” .

[26] T. Zemen, L. Bernadó, N. Czink, and A. F. Molisch, “Iterative time-variant channel estimation for 802.11p using generalized discrete prolate spheroidal sequences,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 1222–1233, 2012, doi: 10.1109/TVT.2012.2185526.

[27] N. M. M. Channels, “An Adaptive Geometry-Based Stochastic Model for,” *IEEE Trans. Wirel. Commun.*, vol. 8, no. 9, pp. 4824–4835, 2009.

[28] C. X. Wang, X. Cheng, and D. Laurenson, “Vehicle-to-vehicle channel modeling and measurements: Recent advances and future challenges,” *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 96–103, 2009, doi: 10.1109/MCOM.2009.5307472.

[29] X. Cheng *et al.*, “An improved parameter computation method for a MIMO V2V rayleigh fading channel simulator under non-isotropic scattering environments,” *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 265–268, 2013, doi: 10.1109/LCOMM.2013.011113.121535.

[30] X. Cheng *et al.*, “Cooperative MIMO channel modeling and multi-link spatial correlation properties,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 388–396, 2012, doi: 10.1109/JSAC.2012.120218.

[31] I. Sen and D. W. Matolak, “Vehicle-vehicle channel models for the 5-GHz band,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 235–245, 2008, doi: 10.1109/TITS.2008.922881.

[32] G. Acosta-Marum and M. A. Ingram, “Six time- and frequency- selective empirical channel models for vehicular wireless LANs,” *IEEE Veh. Technol. Mag.*, vol. 2, pp. 4–11, 2007.

[33] X. Cheng, Q. Yao, M. Wen, C. X. Wang, L. Y. Song, and B. L. Jiao, “Wideband channel modeling and intercarrier interference cancellation for Vehicle-to-Vehicle communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 434–448, 2013, doi: 10.1109/JSAC.2013.SUP.0513039.

[34] F. Pena-Campos, R. Carrasco-Alvarez, O. Longoria-Gandara, and R. Parra-Michel, “Estimation of fast time-varying channels in OFDM systems using two-dimensional prolate,” *IEEE Trans. Wirel. Commun.*, vol. 12, no. 2, pp. 898–907, 2013, doi: 10.1109/TWC.2013.010413.120624.

[35] M. O. Damen, H. El Gamal, and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, 2003, doi: 10.1109/TIT.2003.817444.

[36] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K. D. Kammeyer, “Efficient algorithm for decoding layered space-time codes,” *Electron. Lett.*, vol. 37, no. 22, pp. 1348–1350, 2001, doi: 10.1049/el:20010899.

[37] S. Statistics, H. Vikalo, B. Hassibi, and H. Vikalo, “On the Sphere-Decoding Algorithm I .,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, 2005.

[38] H. Moroga, T. Yamamoto, and F. Adachi, “Iterative overlap QRM-ML block detection for single-carrier MIMO transmission without CP insertion,” *Wirel. Pers. Commun.*, vol. 74, no. 4, pp. 1163–1177, 2014, doi: 10.1007/s11277-013-1570-5.

[39] K. Temma, T. Yamamoto, and F. Adachi, “Improved 2-step QRM-ML block signal detection for single-carrier transmission,” *IEEE Veh. Technol. Conf.*, no. 0, 2011, doi: 10.1109/VETECF.2011.6093118.

[40] C. C. Alatriste, Gabriel Torres, Olvera, “Modelo de Predicción de LEE,” pp. 25–43, 2007, [Online]. Available: https://tesis.ipn.mx/bitstream/handle/123456789/3523/MODELODEPREDICCION.pdf?sequence=1&isAllowed=y.

[41] J. A. Del Puerto-Flores, R. Parra-Michel, F. Peña-Campos, J. Cortez, and E. Romero-Aguirre, “Evaluation of OFDM Systems with Virtual Carriers over V2V Channels,” *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2018*, no. 1, pp. 882–886, 2019, doi: 10.1109/IEMCON.2018.8615092.

[42] J. A. Del Puerto-Flores, L. C. Yllescas, R. Parra-Michel, F. Pena-Campos, and J. Cortez, “Performance Evaluation of Turbo Decoding in DFTS-OFDM Systems over V2V Channel,” *Proc. - 2018 10th IEEE Latin-American Conf. Commun. LATINCOM 2018*, 2019, doi: 10.1109/LATINCOM.2018.8613202.

[43] D. A. Ibarra and Ciudad Obregón, “Arquitectura en hardware de un detector OSIC para receptor SISO V2V,” 2020.