

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
CARRERA DE MAESTRÍA EN SISTEMAS EMBEBIDOS



MEMORIA DEL TRABAJO FINAL

**Módulos de búsqueda y seguimiento
para receptor GPS sobre FPGA**

Autor:

Esp. Ing. Facundo Larosa

Director:

Dr. Ing. Héctor Lacomí (CITEDEF, UTN FRH)

Codirector:

Ing. Nicolás Álvarez (FIUBA, UNSAM)

Jurados:

Ing. Carlos Beláustegui Goitía (FIUBA)
Ing. Edgardo Comas (CITEDEF, UTN FRBA)
Ing. Juan Manuel Cruz (FIUBA, UTN FRBA)

Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires entre abril y noviembre de 2017.

Resumen

El trabajo planteado abarca el estudio, diseño, simulación y pruebas de un módulo de búsqueda y de un módulo de seguimiento sobre FPGA que serán utilizados a futuro para la implementación de un receptor GPS definido por software. La implementación de sistemas de comunicaciones sobre hardware configurable es una línea de gran interés ya que las técnicas de diseño, desarrollo y pruebas se utilizan en una gran cantidad de aplicaciones y constituyen una rama altamente específica del diseño de sistemas embebidos facilitando el estudio y ensayo de arquitecturas de receptores satelitales.

Este trabajo involucró la puesta en práctica de diferentes áreas de conocimiento estudiadas durante el posgrado: diseño de hardware programable, sistemas de comunicaciones sobre FPGA, control digital, sistemas embebidos de tiempo real, protocolos de comunicaciones, entre otras.

« Aunque tuviera el don de la profecía y conociera todos los misterios y toda la ciencia, aunque tuviera toda la fe, una fe capaz de trasladar montañas, si no tengo amor, no soy nada.» (1 Cor 13:2)

Agradecimientos

Con la escritura de estos agradecimientos llega a su fin un período de dos años de estudio, de reflexión y de grandes desafíos: una verdadera aventura.

Quisiera agradecer a Ariel Lutenberg quién depositó su confianza en mí y posibilitó el cursado del posgrado a través del otorgamiento de una beca.

A mis compañeros, Danilo Zecchin y Pablo Ridolfi, quienes como modernos Virgilio me acompañaron con sus consejos y su apoyo incondicional en los momentos de flaqueza.

A mi director, Héctor Lacomi, quién me aconsejó y corrigió pacientemente el texto de este trabajo con gran dedicación y comprensión. Su ayuda fue inestimable.

A mi familia, pilares silenciosos, no analizaron ecuaciones, razonamientos o diagramas, pero sin su apoyo no podría haber logrado nada.

Tal vez resulte extraño la cita de una epístola paulina acerca de la preeminencia del amor en un texto técnico. Tal vez alguien pueda pensar que está fuera de lugar o que es afectado o pomposo. A mi favor quiero decir que busqué en ella una síntesis de mis sentimientos, unas palabras que reflejen mi ánimo y que permanezcan como un recordatorio de este momento.

Todo permanece en constante cambio: áreas de estudio, enfoques, colegas... hasta el contexto de nuestras labores varía año a año. Así, en el transcurso de nuestras carreras, los profesionales acumulamos técnicas, herramientas, conceptos, abstracciones, esquemas de organización... de forma que es grande la tentación de olvidar lo esencial y confundir los medios con el fin: simplemente nos preparamos para servir mejor a los demás, a la sociedad, a nuestro Pueblo.

Me resulta misterioso que mis profesores me decían que uno tenía que hacer el *sacrificio* de estudiar. La palabra me remitía a dolor, a privación, sentía por ella una profunda resistencia. Pero el significado original del término viene del latín *sacro* (sagrado) y *facere* (hacer): «hacer sagradas las cosas». Pido siempre entonces, que el amor que hace sagradas las cosas, que trasciende toda ciencia, la complete y le de sentido en los demás. Así sea.

Índice general

Resumen	III
1. Introducción General	1
1.1. Introducción	1
1.1.1. Descripción general del sistema GPS	1
1.2. Objetivos	4
2. Introducción Específica	5
2.1. Generación de señal GPS	5
2.1.1. Esquema general	5
2.1.2. Generación y características del código C/A	5
2.1.3. Efecto del medio sobre la señal GPS	10
2.2. Método de búsqueda serial	11
2.2.1. Caso general	15
2.2.2. Caso particular: Código C/A local en fase con código C/A de la señal de entrada	17
2.2.3. Caso particular: Oscilador local con igual frecuencia de la señal de frecuencia intermedia	18
2.2.4. Caso particular: Coincidencia de fase de C/A y desplazamiento de frecuencia de la señal local y la señal de entrada	19
2.3. Diseño del lazo de seguimiento	20
2.3.1. Estimación de la diferencia de fase y frecuencia entre la portadora de la señal de entrada y la señal local	21
2.3.2. Estimación de la diferencia de fase de código C/A entre la señal de entrada y la señal local	22
2.3.3. Controladores de lazo de portadora y de código	24
2.3.4. Estudio y simulación del lazo de portadora	25
Respuesta del sistema frente a escalón de frecuencia	28
Respuesta del sistema frente a escalón de fase	31
2.4. Conclusiones	33
3. Diseño e Implementación	35
3.1. Introducción	35
3.1.1. Consideraciones generales	35
3.1.2. Consideraciones de diseño	36
3.2. Front end	38
3.3. Módulo de adquisición de señales digitales	39
3.3.1. Interfaz	40
3.3.2. Operación	40
3.3.3. Pruebas	41
3.4. Módulo de reproducción de señales digitales	43
3.4.1. Interfaz	43
3.4.2. Operación	44

3.4.3. Pruebas	44
3.5. Generador de código C/A	45
3.5.1. Interfaz	46
3.5.2. Operación	46
3.5.3. Pruebas	46
3.6. Oscilador controlado numéricamente (NCO)	47
3.6.1. Interfaz	47
3.6.2. Operación	47
3.6.3. Pruebas	48
3.7. Módulo de entrada	50
3.7.1. Interfaz	50
3.7.2. Operación	50
3.7.3. Pruebas	51
3.8. Módulo de memoria	52
3.8.1. Interfaz	52
3.8.2. Operación	52
3.8.3. Pruebas	54
3.9. Módulo de generación de señal GPS local	58
3.9.1. Interfaz	58
3.9.2. Operación	58
3.9.3. Pruebas	60
3.10. Módulo de búsqueda	62
3.10.1. Interfaz (núcleo)	62
3.10.2. Operación	63
3.10.3. Pruebas	65
3.10.4. Interfaz (periférico)	67
3.10.5. Operación	67
3.10.6. Pruebas	69
3.11. Módulo de seguimiento	71
3.11.1. Interfaz	71
3.11.2. Operación	71
3.11.3. Pruebas	76
3.11.4. Interfaz (periférico)	78
3.11.5. Operación	78
3.11.6. Pruebas	80
4. Ensayos y Resultados	83
4.1. Validación de algoritmos de búsqueda utilizando señales reales	83
4.1.1. Búsqueda de satélites en vista	84
4.1.2. Evolución de frecuencia y fase de código C/A	87
4.2. Validación de módulo de búsqueda	88
4.3. Validación de algoritmo de seguimiento con señales reales	93
5. Conclusiones	97
5.1. Conclusiones	97
5.2. Trabajo a futuro	98
Bibliografía	99

Índice de figuras

1.1. Segmento de control del sistema GPS [3]	3
1.2. Órbitas de los satélites del sistema GPS (vista superior) en coordenadas ECEF	3
1.3. Órbitas de los satélites del sistema GPS (lateral)	4
2.1. Esquema general de generación de señal GPS donde se indican las frecuencias específicas de cada señal involucrada	6
2.2. Esquema de generación de los códigos de Gold	6
2.3. Autocorrelación del código C/A	9
2.4. Autocorrelación del código C/A (detalle en torno a $n=0$)	9
2.5. Correlación cruzada del código C/A del satélite 1 y 2	10
2.6. Desviación de la frecuencia central por efecto Doppler	11
2.7. Desviación de la frecuencia central por efecto Doppler	11
2.8. Esquema general diagrama de búsqueda	13
2.9. Gráfico de $C[n_{ol}, \Delta f_{dl}]$	17
2.10. Gráfico de $C[n_{ol} = n_0, \Delta f_{dl}]$	18
2.11. Gráfico de $C[n_{ol}, \Delta f_{dl} = \Delta f_d]$	19
2.12. Estructura general del lazo de seguimiento	21
2.13. Valores de correlación normalizados para los códigos <i>Early, Prompt</i> y <i>Late</i> (señal en fase)	23
2.14. Valores de correlación normalizados para los códigos <i>Early, Prompt</i> y <i>Late</i> (señal atrasada)	23
2.15. Estructura en bloques del controlador del lazo de seguimiento	24
2.16. Estructura en bloques del lazo de portadora	25
2.17. Modelo del lazo de portadora en el dominio Z	26
2.18. Estructura general del sistema realimentado	27
2.19. Cambio de la referencia de frecuencia de la señal de entrada	29
2.20. Error de fase instantáneo ($\Delta\phi$)	29
2.21. Correlación del canal en fase (I)	29
2.22. Correlación del canal en cuadratura (Q)	30
2.23. Variación de la frecuencia del oscilador local digital (Δf)	30
2.24. Cambio de la referencia de fase de la señal de entrada	31
2.25. Error de fase instantáneo ($\Delta\phi$)	31
2.26. Correlación del canal en fase (I)	32
2.27. Correlación del canal en cuadratura (Q)	32
2.28. Variación de la frecuencia del oscilador local digital (Δf)	32
3.1. Diagrama en bloques del <i>frontend</i>	35
3.2. Diagrama en bloques general de una FSMD	37
3.3. Diagrama esquemático del circuito del <i>frontend</i>	38
3.4. Placa de circuito impreso (PCB) del <i>frontend</i>	39
3.5. Fotografía del <i>frontend</i> con su antena activa conectado a la placa de desarrollo FPGA	39

3.6. Interfaz del módulo de adquisición de señales digitales	40
3.7. Diagrama en bloques del módulo de adquisición de señales digitales	41
3.8. Señales correspondientes al inicio de una adquisición	42
3.9. Señales correspondientes al proceso de adquisición	42
3.10. Señales correspondientes al fin del proceso de adquisición	43
3.11. Interfaz del módulo de reproducción de señales digitales	43
3.12. Diagrama en bloques del módulo de reproducción de señales digitales	44
3.13. Archivo de datos de la memoria (COE)	45
3.14. Señales del módulo de reproducción (captura del <i>testbench</i>)	45
3.15. Señales del módulo de reproducción (captura del analizador lógico)	45
3.16. Interfaz del módulo de generación de código C/A	46
3.17. Diagrama en bloques del generador de código CA	47
3.18. Generación de código C/A (salida <i>caCode</i> para el satélite 1	47
3.19. Interfaz del NCO	48
3.20. Diagrama en bloques del NCO	49
3.21. Oscilograma de las señales de salida del NCO: seno (superior) y coseno (inferior)	49
3.22. Valores medidos de frecuencia de salida en función del incremento de fase	50
3.23. Interfaz del módulo de entrada	50
3.24. Diagrama en bloques del módulo de entrada	51
3.25. Diagrama de señales del módulo de entrada	52
3.26. Interfaz del módulo de memoria	52
3.27. Diagrama en bloques del módulo de memoria	53
3.28. Comienzo de la prueba de escritura continua	55
3.29. Comienzo del ciclo de grabación	55
3.30. Fin del ciclo de grabación de la memoria 0, comienzo de grabación de la memoria 1	55
3.31. Escritura alternada de las memorias 0 y 1	55
3.32. Fin de escritura de la memoria 1 y comienzo de lectura de memoria 0	56
3.33. Lectura de memoria 0	56
3.34. Escritura alternada de las memorias 0 y 1 en ausencia de nuevo comando de lectura	57
3.35. Lectura continua de la memoria 0 mientras la memoria 1 se está grabando	57
3.36. Detalle del cambio de escritura/lectura de la memoria 1 a la memoria 0	57
3.37. Interfaz del módulo de generación de señal GPS local	58
3.38. Diagrama en bloques del módulo de generación de señales local	59
3.39. Inicio de la grabación de señales réplica	60
3.40. Detalle de la inversión de fase de las señales réplica	61
3.41. Fin de la grabación de las señales réplica	61
3.42. Lectura de las señales réplica (<i>initPhase=0</i>)	62
3.43. Lectura de las señales réplica (<i>initPhase=1024</i>)	62
3.44. Detalle del cambio de dirección de lectura	62
3.45. Interfaz del núcleo del módulo de búsqueda	63
3.46. Diagrama en bloques del módulo de búsqueda	64
3.47. Comienzo de la inicialización del módulo de búsqueda	65
3.48. Fin de la inicialización del módulo de búsqueda	65

3.49. Inicio de generación de señal GPS local	66
3.50. Inicio de la correlación de señal GPS local	66
3.51. Fin de la correlación de señal GPS local	67
3.52. Interfaz del módulo de búsqueda	67
3.53. Diagrama en bloques del módulo de búsqueda con su interfaz SPI .	68
3.54. Señales de inicialización del periférico de búsqueda	70
3.55. Señales de inicialización completada	70
3.56. Señales de configuración de correlación	70
3.57. Señales de inicio de correlación	71
3.58. Señales de finalización de correlación	71
3.59. Señales de lectura de valores de correlación	71
3.60. Interfaz del módulo de seguimiento	72
3.61. Interfaz del módulo de seguimiento	73
3.62. Valores de correlación normalizados para los códigos Early, Prompt y Late (señal en fase)	74
3.63. Valores de correlación normalizados para los códigos Early, Prompt y Late (señal atrasada)	74
3.64. Diagrama en bloques (énfasis en los bloques utilizados para detec- tar la sincronía del código C/A)	75
3.65. Diagrama en bloques (énfasis en los bloques utilizados para detec- tar la sincronía de la señal de FI)	76
3.66. Inicialización módulo de seguimiento	76
3.67. Sincronización del código C/A	77
3.68. Inicio de la acumulación	77
3.69. Fin de la acumulación	77
3.70. Interfaz del módulo de seguimiento	78
3.71. Diagrama en bloques del módulo de seguimiento con su interfaz SPI	79
3.72. Inicialización de parámetros del periférico de seguimiento	80
3.73. Pulso <i>setStatus</i> que habilita la configuración del periférico	81
3.74. Lectura de los valores de correlación del periférico de seguimiento	81
4.1. Esquema de conexión del <i>frontend</i> con el módulo de adquisición y la memoria SRAM	84
4.2. Fotografía del <i>frontend</i> con su antena activa conectado a la placa de desarrollo FPGA	84
4.3. Matriz de correlación normalizada de una secuencia real (satélite no a la vista)	85
4.4. Matriz de correlación normalizada de una secuencia real (satélite a la vista)	86
4.5. Máximos de correlación para la operación de búsqueda en función del número de satélite	86
4.6. Máximos de correlación (divididos por la media de la matriz) para la operación de búsqueda en función del número de satélite	86
4.7. Matriz de correlación normalizada de una secuencia real (período de integración = 5ms)	87
4.8. Evolución temporal del desplazamiento de la frecuencia central . .	88
4.9. Evolución temporal de la fase del código C/A	88
4.10. Conexión de módulos instanciados sobre la FPGA para validación del módulo de búsqueda	89
4.11. Placa de desarrollo EDU-CIAA-NXP	89
4.12. Esquema de conexión entre placa EDU-CIAA-NXP y placa Nexys 2	90

4.13. Matriz de correlación normalizada calculada por el periférico de búsqueda (señal sintética)	91
4.14. Valores de correlación para fase de código C/A fija y frecuencia variable (señal sintética)	91
4.15. Valores de correlación para frecuencia fija y fase de código C/A variable (señal sintética)	92
4.16. Matriz de correlación normalizada calculada por el periférico de búsqueda (señal real)	92
4.17. Valores de correlación para fase de código C/A fija y frecuencia variable (señal real)	92
4.18. Valores de correlación para frecuencia fija y fase de código C/A variable (señal real)	93
4.19. Estructura general del lazo de seguimiento	93
4.20. Estructura en bloques del controlador del lazo de seguimiento . . .	94
4.21. Estructura en bloques del lazo de portadora	94
4.22. Valores de estimadores <i>Early</i> , <i>Prompt</i> y <i>Late</i> para algoritmo de seguimiento con señales reales	95
4.23. Valor del canal I (en fase) para algoritmo de seguimiento con señales reales	95
4.24. Valor del canal Q (en cuadratura) para algoritmo de seguimiento con señales reales	96
4.25. Valor del canal I (en fase) para algoritmo de seguimiento con señales reales (en negro, se resalta la envolvente de los bits del mensaje e navegación)	96

Índice de Tablas

1.1. Planos orbitales	2
2.1. Salidas	7
3.1. Parámetros del proyecto	36
3.2. Interfaz - Módulo de adquisición de señales digitales	40
3.3. Interfaz - Módulo de reproducción de señales digitales	44
3.4. Interfaz - Módulo Generador de código C/A	46
3.5. Interfaz - NCO	48
3.6. Interfaz - Módulo de entrada	51
3.7. Interfaz - Módulo de memoria	53
3.8. Interfaz - Módulo de generación de señal GPS local	58
3.9. Interfaz - Módulo de búsqueda	63
3.10. Interfaz - Módulo de búsqueda	68
3.11. Interfaz - Módulo de seguimiento	72
3.12. Interfaz - Módulo de seguimiento	78

Capítulo 1

Introducción General

1.1. Introducción

1.1.1. Descripción general del sistema GPS

El sistema de posicionamiento global (GPS, del inglés *Global Positioning System*) es un sistema de navegación satelital creado, operado y mantenido por el gobierno de los Estados Unidos de América (EUA) a través del Departamento de Defensa. La Fuerza Aérea de los Estados Unidos (USAF, del inglés *United States Air Force*) se encarga de la operación del segmento de control a través de estaciones terrenas distribuidas en todo el mundo.

El sistema tiene por objeto proveer servicios de posicionamiento a los usuarios militares de los EUA y sus aliados (denominado PPS, *Precise Positioning Service*), así como servicios de posicionamiento a usuarios civiles de menor resolución (denominado SPS, *Standard Positioning Service*).

Los servicios de uso civil estaban originalmente sometidos a una degradación intencional de la precisión de sus señales conocida como 'disponibilidad selectiva' (SA, del inglés *selective availability*). La disponibilidad selectiva fue eliminada definitivamente del sistema a través de un decreto de la administración Clinton el 1 de mayo del año 2000 [1]. Los nuevos satélites del Bloque III serán puestos en servicio a partir del año 2017 carecen de este mecanismo, reafirmando el compromiso del sistema con los usuarios civiles.

En cuanto a la división de responsabilidades de las diferentes partes del sistema, se puede dividir en tres grandes segmentos: el segmento de control, el segmento satelital y el segmento de usuario.

- **Segmento de control:** se compone de las estaciones terrenas de control, las cuales se encargan de monitorear el estado y posición de los satélites, realizar análisis y enviar comandos de control y datos a la constelación. En la figura 1.1 se pueden observar diferentes estaciones ubicadas a lo largo y ancho del mundo. Las estaciones principales se ubican en bases de la USAF: Colorado Springs (la estación principal), isla Ascensión, isla Diego García, Kwajalein y Hawaii.
- **Segmento satelital:** está compuesto por los satélites operacionales y en reserva de la constelación GPS. Éstos se encuentran ubicados en seis planos orbitales de inclinación aproximadamente constante (en torno a 55°) respecto del ecuador terrestre. Los planos se encuentran separados unos 60°

en azimut. Las órbitas tienen un radio promedio de 26.000 km aproximadamente, poseyendo baja excentricidad (son cuasicirculares) y el período de las mismas es de alrededor de 12 horas (de forma tal que cada 24 horas aproximadamente la constelación completa un período respecto de un observador terrestre). Se requieren al menos 24 satélites en operación para mantener los servicios de posicionamiento con una precisión de acuerdo a los requerimientos del sistema, es decir, 4 satélites por plano orbital. En la figura 1.2 se muestra un gráfico de las órbitas de los satélites del sistema GPS realizado en Octave en base a datos extraídos de la página oficial de la Federación Rusa (<https://www.glonass-iac.ru/en/GLONASS/>).

Las órbitas de los satélites se graficaron utilizando el sistema ECEF (del inglés, *Earth-Centered, Earth-Fixed*), un sistema de coordenadas solidario a la Tierra con origen en el centro terrestre, eje X en la dirección del punto de latitud / longitud $0^\circ, 0^\circ$, eje Z en la dirección del eje de rotación terrestre sentido Norte y eje Y mutuamente perpendicular a los anteriores, formando una terna derecha. Cada color representa un plano orbital según lo que se detalla en la tabla 1.1.

En la figura 1.2 los planos orbitales desplazados 180° se observan superpuestos debido a la perspectiva. En la figura 1.3 se observa desde una perspectiva lateral, permitiendo apreciar los planos de 80° y 260° (separados 180°).

Los satélites radian las diferentes señales del sistema de uso militar y civil. La señal estándar de uso civil se conoce como L1; a través de ella cada satélite envía un mensaje de navegación compuesto por su estado de salud, parámetros orbitales (efemérides) y parámetros de corrección. A través de los parámetros orbitales y de corrección, se puede hallar la posición de cada satélite para, en base a mediciones indirectas de la distancia al mismo, hallar la posición del receptor.

- **Segmento de usuario:** se compone por los receptores que hacen uso de las señales del sistema GPS para su posicionamiento. Un receptor GPS se encarga de recibir y procesar el contenido de las señales desde la recepción de la señal de radiofrecuencia, hasta la obtención del mensaje de navegación y el informe del cálculo de posición al usuario a partir de las ecuaciones de posicionamiento [2].

TABLA 1.1: Satélites por plano orbital

Azimut	Satélites (por número)	Color
20°	8 12 13 14 22	Amarillo
80°	7 23 29 30	Rojo
140°	11 15 24 25 27	Verde
200°	16 18 26 28	Azul
260°	1 2 4 6 20	Violeta
320°	3 5 9 17 19 21 31	Negro

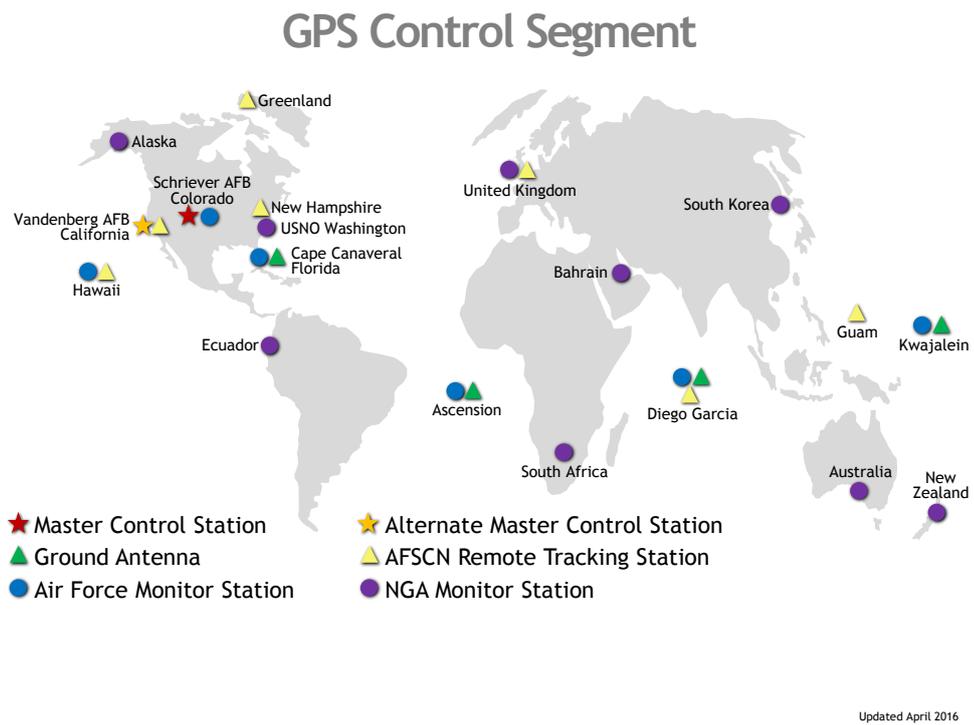


FIGURA 1.1: Segmento de control del sistema GPS [3]

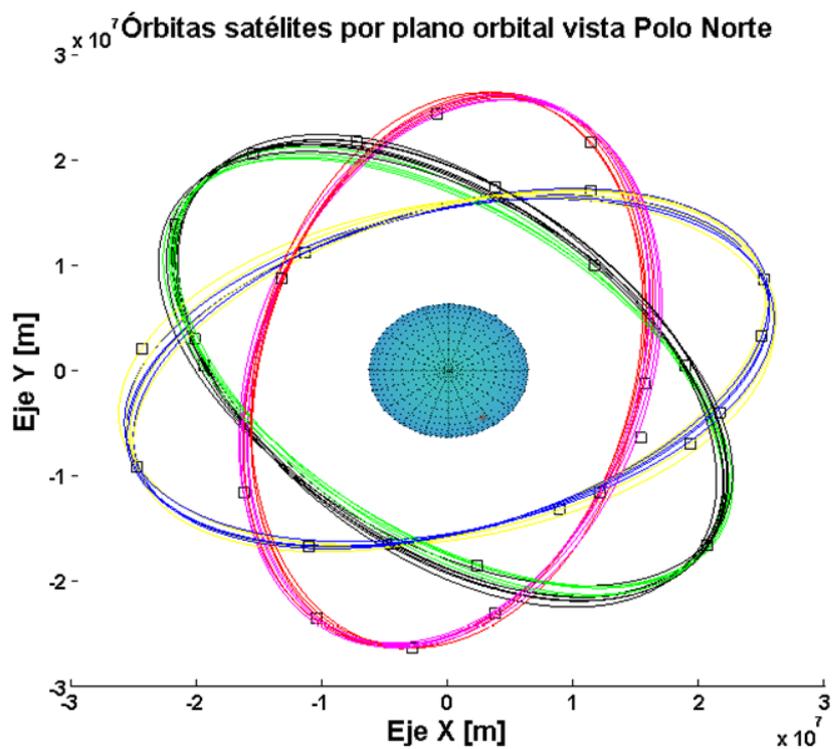


FIGURA 1.2: Órbitas de los satélites del sistema GPS (vista superior) en coordenadas ECEF

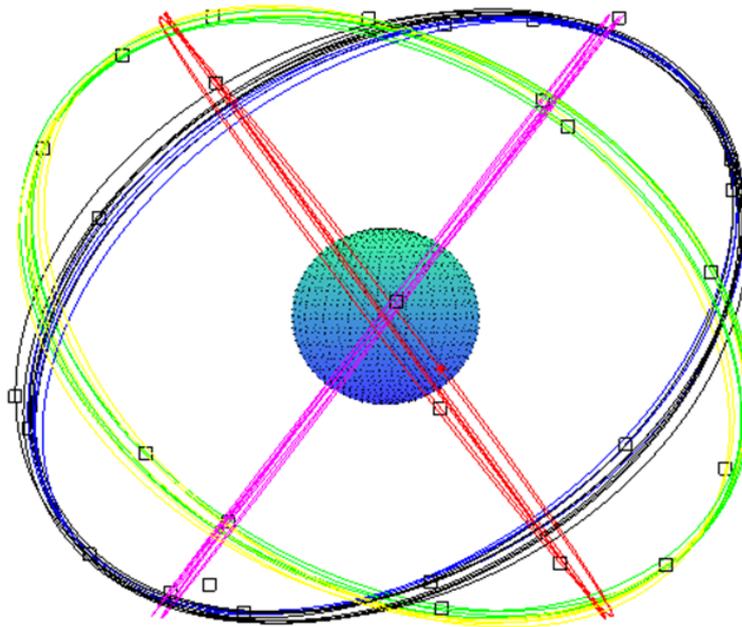


FIGURA 1.3: Órbitas de los satélites del sistema GPS (lateral)

1.2. Objetivos

Se plantearon como objetivos principales de este trabajo el diseño, implementación, simulación y ensayos de módulos digitales destinados a procesar una señal GPS en banda L1. Los módulos se describen en lenguaje VHDL y se realizaron ensayos utilizando señales sintéticas y reales.

Cómo objetivos secundarios se plantearon la familiarización y aprendizaje de técnicas de descripción de circuitos digitales en VHDL, utilización de bancos de pruebas (*testbenches*) a fin de validar el correcto funcionamiento de los circuitos implementados en cuanto al procesamiento de las señales, la implementación de un sistema de control digital para el módulo de seguimiento, el desarrollo de *scripts* para el procesamiento de señales y simulación de diferentes partes del sistema, entre otros.

Los módulos principales son: el módulo de búsqueda (*search*) y el módulo de seguimiento (*tracking*). El módulo de búsqueda tiene por función en base a la señal de entrada detectar si se encuentra presente un determinado satélite (cada satélite está individualizado por un código diferente como se verá en el Capítulo 2), así como parámetros propios de la señal (fase de código C/A y desplazamiento de la frecuencia central).

Estos parámetros se utilizan para alimentar un módulo de seguimiento, el cual tiene por función «seguir» la fase de código C/A y desplazamiento de la frecuencia central para el satélite hallado de forma tal de poder extraer el mensaje de navegación.

Capítulo 2

Introducción Específica

En esta sección se tratan diferentes aspectos de la generación de las señales GPS: generación, propiedades, efecto del medio y detección.

2.1. Generación de señal GPS

2.1.1. Esquema general

Este proyecto está enfocado en el procesamiento de la banda L1 del sistema GPS, que es la banda de posicionamiento de uso estándar civil (denominada en inglés, SPS, Standard Positioning System). Todos los satélites de esta constelación, emiten su mensaje de navegación (de aquí en más, el mensaje) a una única frecuencia de portadora de 1575,42MHz. A fin de poder diferenciar los mensajes de los diferentes satélites, se hace el producto del mensaje con una señal de «ruido pseudoaleatorio», denominada código C/A (del inglés, *coarse acquisition*), el cual es único para cada satélite. Esta técnica se denomina, según se analice como una modulación o un tipo de multiplexación como DSSS (*direct-sequence spread spectrum*) o CDMA (*code division multiplex access*). Los códigos C/A tienen un período de 1023 chips (se utiliza la palabra «chips» para hacer referencia a los estados del código C/A, reservándose el uso de la palabra bits, sólo para el mensaje) cuya tasa de transmisión es de 1,023 Mb/s, siendo su período de 1ms.

En la figura 2.1 se indica un esquema general de generación de la señal, donde se observa en primer lugar la operación de mezcla entre el código C/A y el mensaje de datos, siendo una operación XOR ya que es una una señal digital de un bit. Luego de la combinación anterior, se procede a realizar la mezcla con la señal de portadora, conformando esta última operación una modulación del tipo BPSK (del inglés, *binary phase shift keying*).

2.1.2. Generación y características del código C/A

Los códigos C/A pertenecen a la familia de códigos pseudoaleatorios de Gold [4]. El esquema de generación de estos códigos se muestra en la figura 2.2, donde se observan dos registros de desplazamiento de 10 bits (G1 y G2), los cuales se inicializan con unos y se realimentan tal como se indica. El código C/A se obtiene realizando la operación XOR de la salida del registro G1 y la XOR de dos flip flops del registro de desplazamiento G2, la cual depende del número de satélite. En la tabla 2.1 se indican las salidas que deben tomarse para cada código de satélite.

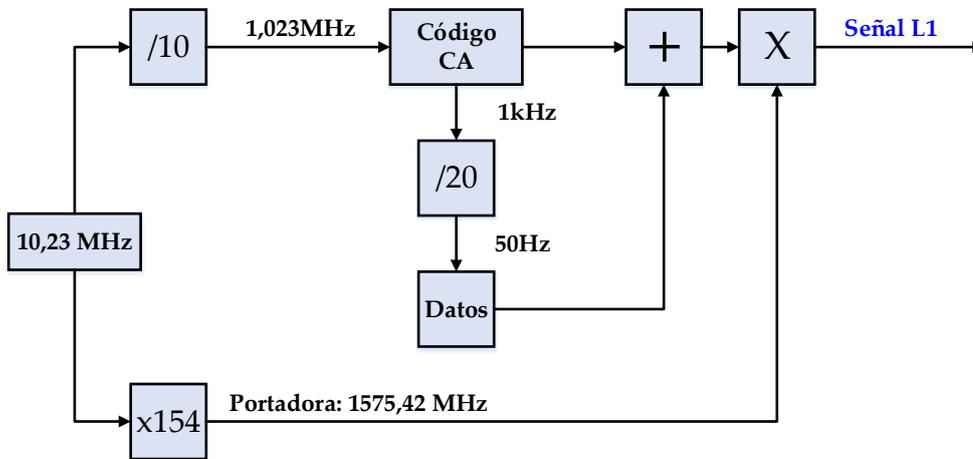


FIGURA 2.1: Esquema general de generación de señal GPS donde se indican las frecuencias específicas de cada señal involucrada

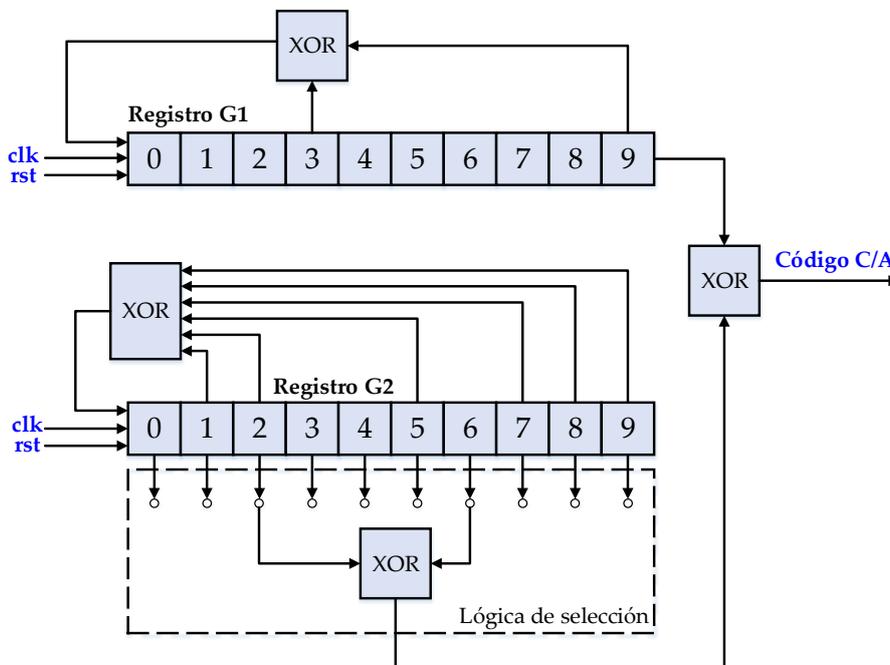


FIGURA 2.2: Esquema de generación de los códigos de Gold

TABLA 2.1: Salidas a tomar según SAT ID

SatID	Selección salidas
1	2 + 6
2	3 + 7
3	4 + 8
4	5 + 9
5	1 + 9
6	2 + 10
7	1 + 8
8	2 + 9
9	3 + 10
10	2 + 3
11	3 + 4
12	5 + 6
13	6 + 7
14	7 + 8
15	8 + 9
16	9 + 10
17	1 + 4
18	2 + 5
19	3 + 6
20	4 + 7
21	5 + 8
22	6 + 9
23	1 + 3
24	4 + 6
25	5 + 7
26	6 + 8
27	7 + 9
28	8 + 10
29	1 + 10
30	1 + 6
31	2 + 7
32	4 + 9

La utilidad de la aplicación de los códigos C/A está dada por las propiedades de la autocorrelación de código C/A de un determinado satélite y la correlación cruzada entre códigos C/A de satélites distintos. Se define el producto de correlación entre dos señales $x[n]$ e $y[n]$ (acotadas en el tiempo, cada una con N cantidad de muestras) como:

$$c_{xy}[n] = x[n] \otimes y[n] \triangleq \sum_{l=0}^{N-1} x^*[l]y[l+n] \quad (2.1)$$

donde:

- $c_{xy}[n]$: Producto de correlación entre $x[n]$ y $y[n]$
- $x[n], y[n]$: Señales a correlacionar
- \otimes : Operador producto de correlación
- $x^*[n]$: $x[n]$ conjugada

En el contexto de este trabajo, se asumirá que el producto de correlación es circular, es decir, que para una señal $x[n]$, acotada a un número de N muestras en el intervalo $n = [0, N - 1]$ se cumple que $x[n] = x[n + N]$. Esta propiedades se identifican usualmente como «aritmética modular de módulo N» [5].

En el caso del código C/A de un satélite i -ésimo, su autocorrelación normalizada ($c_{CAi}[n]$) al valor del máximo de autocorrelación puede expresarse como:

$$c_{CAi}[n] = \frac{1}{M} CA_i[n] \otimes CA_i[n] = \frac{1}{M} \sum_{l=0}^{N-1} CA_i[l] \otimes CA_i[l+n] \quad (2.2)$$

donde:

- $c_{CAi}[n]$: Autocorrelación normalizada
- $CA_i[n]$: Secuencia de código C/A del satélite i -ésimo
- M :Máximo de autocorrelación
- N: Número de muestras

La función de autocorrelación normalizada es máxima para $n = 0$ y comparativamente menor para valores distintos de n . En la figura 2.3, se muestra la función de autocorrelación para el código C/A del satélite 1 ($c_{CA1}[n]$). El tiempo discreto n está escalado para indicar desplazamiento en chips.

En la figura 2.4 se muestra un detalle de la función $c_{CA1}[n]$ (autocorrelación normalizada del satélite 1) en torno a $n = 0$, donde se observa que se desvanece linealmente al desplazarse un chip a derecha o izquierda.

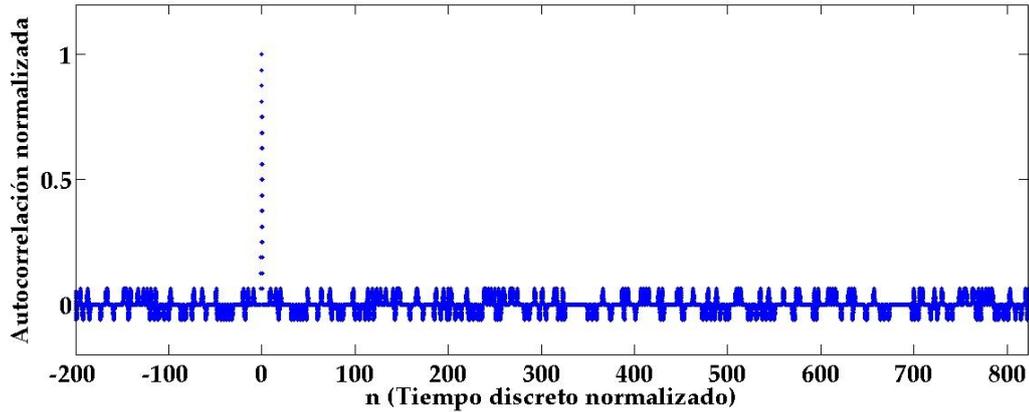


FIGURA 2.3: Autocorrelación del código C/A

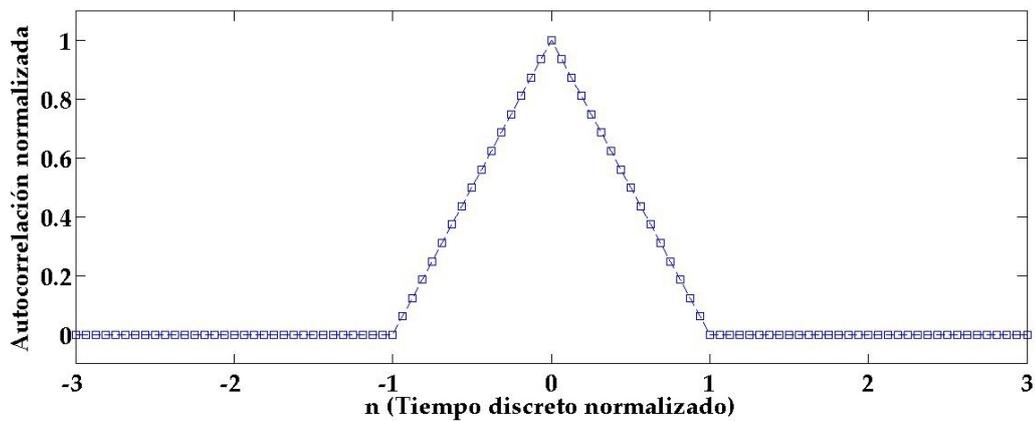


FIGURA 2.4: Autocorrelación del código C/A (detalle en torno a n=0)

Para satélites distintos, la función de correlación cruzada normalizada se expresa como:

$$c_{CA_{ij}}[n] = \frac{1}{M} CA_i[n] \otimes CA_j[n] = \frac{1}{M} \sum_{l=0}^{N-1} CA_i[l] \otimes CA_j[l+n] \quad (2.3)$$

donde:

- $c_{CA_{ij}}[n]$: Correlación normalizada
- $CA_i[n]$: Secuencia de código C/A del satélite i-ésimo
- $CA_j[n]$: Secuencia de código C/A del satélite j-ésimo
- M : Máximo de correlación
- N : Número de muestras

En la figura 2.5 se muestra como ejemplo la correlación para el código C/A del satélite 1 y el 2.

Se observa que para todo valor de n , la correlación no presenta un máximo distintivo. Esta propiedad de los códigos C/A se utiliza para discriminar los mensajes

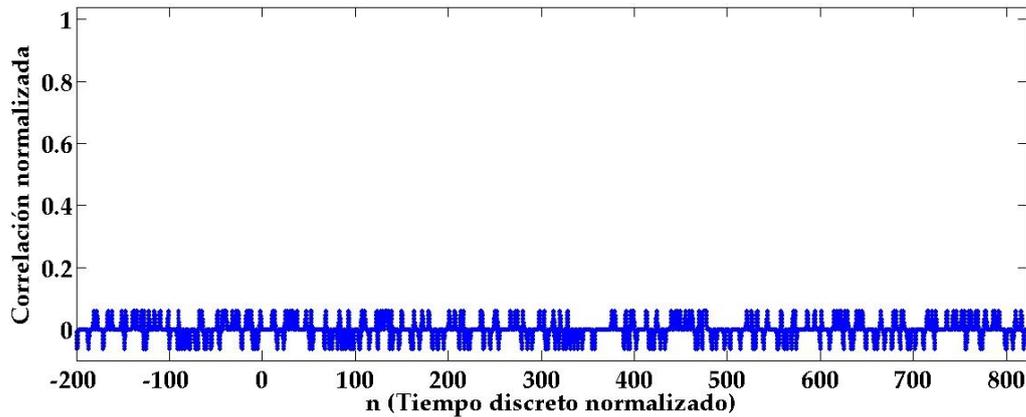


FIGURA 2.5: Correlación cruzada del código C/A del satélite 1 y 2

provenientes de diferentes satélites y para sincronizar la señal de entrada para un satélite determinado de forma local.

2.1.3. Efecto del medio sobre la señal GPS

La señal generada y radiada por los satélites se propaga por el espacio libre, atraviesa las capas de la atmósfera y es recibida por el receptor. Dado que los satélites se encuentran en constante movimiento (se desplazan en órbitas cuasicirculares con un radio medio de 26.000 km aproximadamente), la señal emitida llegará al receptor con una fase desconocida y afectada por efecto Doppler, el cual desplazará la frecuencia central (1575,42 MHz) de acuerdo a velocidad relativa (en el sentido radial) entre el satélite y el receptor.

Ignorando efectos relativistas, se puede describir la variación de frecuencia experimentada en términos de la longitud de onda de la señal vista desde el receptor como:

$$\lambda = \lambda_0 \left(1 + \frac{1}{c} \hat{k} \cdot \vec{v} \right) \quad (2.4)$$

donde:

- λ : Longitud de onda observada por el receptor [m]
- λ_0 : Longitud de onda emitida por el satélite [m]
- c : Velocidad de la luz [m/s]
- \hat{k} : Versor en la dirección de la línea de vista receptor-satélite
- \vec{v} : Velocidad del satélite relativa al receptor

Utilizando MATLAB, se realizaron simulaciones de la cinemática de los satélites respecto de un receptor estacionario relativo a la Tierra. El receptor se ubicó en las coordenadas $-34,603333^\circ S$, $-58,381667^\circ O$, correspondientes al centro de la Ciudad de Buenos Aires. En base a las ecuaciones de posicionamiento indicadas en [6] se discretizó la posición para cada satélite en función del tiempo. Luego, se obtuvo la velocidad instantánea de cada uno de los satélites y se aplicó la ecuación 2.4 a fin de obtener el desplazamiento en frecuencia. Los parámetros orbitales se

obtuvieron de datos reales del *almanac*¹ de los satélites de la constelación GPS, los cuales se muestran en tiempo real en la página oficial de la Federación Rusa (<https://www.glonass-iac.ru/en/GLONASS/>).

En la figura 2.6 se muestra la variación de frecuencia central debido al efecto Doppler para diferentes satélites de la constelación en vista. El lapso simulado abarca seis horas, observándose que el corrimiento de frecuencia se encuentra aproximadamente en el intervalo $\Delta f = \pm 4000 Hz$ respecto de la frecuencia central.

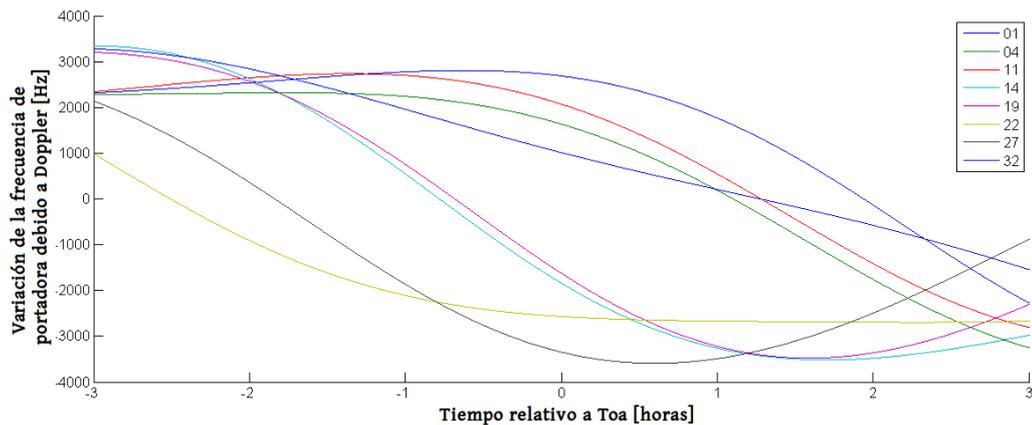


FIGURA 2.6: Desviación de la frecuencia central por efecto Doppler

En la figura 2.7 se muestra la razón de cambio de la frecuencia central respecto del tiempo. Se observa que en todos los casos esta razón es menor a $1 \frac{Hz}{s}$.

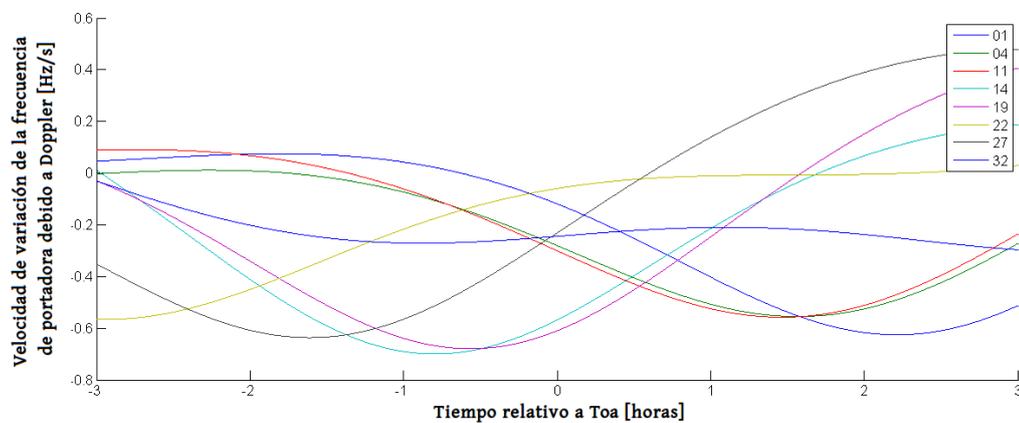


FIGURA 2.7: Desviación de la frecuencia central por efecto Doppler

2.2. Método de búsqueda serial

Los algoritmos de búsqueda se utilizan para hallar parámetros esenciales de la señal GPS de cada satélite, que luego se utilizarán para realizar el seguimiento y

¹El almanac es un juego reducido de efemérides que se utiliza para posicionar los satélites de forma aproximada y facilitar la predicción de su ubicación

poder obtener el mensaje de navegación. De esta manera, el método de búsqueda debe ser capaz de:

- Hallar el valor de la fase del código C/A (n_0)
- Hallar el valor de desplazamiento de la frecuencia central debida al efecto Doppler (f_d)
- Ser insensible a la fase de la portadora (ϕ_0)

El método de búsqueda serial desarrollado permite hallar el valor de la fase del código C/A(n_0) y el valor de desplazamiento de la frecuencia central debido al efecto Doppler(Δf_d). Es un método de los tipos denominados «de fuerza bruta» en el cual se intenta dar con la combinación correcta ($n_0, \Delta f_d$) probando exhaustivamente todas las combinaciones posibles dentro de un dominio determinado en búsqueda del máximo de autocorrelación entre la señal GPS a la entrada del receptor y una réplica generada localmente.

La señal de entrada del sistema (sin considerar ruido) debida a un satélite se puede describir de forma general como:

$$x[n] = A \cdot D[n] \cdot CA_i[n + n_0] \cdot \cos[2\pi(f_i + \Delta f_d)T_s n + \phi_0] \quad (2.5)$$

Donde:

- $x[n]$: Señal GPS en el dominio del tiempo discreto [V]
- n : Variable independiente, tiempo discreto [adimensional]
- n_0 : Fase inicial del código C/A [adimensional]
- A : Amplitud pico de la señal [V]
- $CA_i[n]$: Código C/A del satélite i -ésimo [adimensional]
- $D[n]$: Señal de datos [adimensional]
- f_i : Frecuencia intermedia [Hz]
- Δf_d : Desplazamiento de la frecuencia central debida al efecto Doppler [Hz]
- T_s : Tiempo entre muestras, inversa de la frecuencia de muestreo [s]
- ϕ_0 : Fase inicial de la portadora [adimensional]

La señal de datos $D[n]$ es una señal binaria que modula la fase de la portadora. Siendo su valor -1 o 1 , puede obviarse en el desarrollo matemático de las técnicas que se van a desarrollar incluyendo su efecto directamente en el valor de la fase ϕ_0 .

El esquema general de búsqueda serial se indica en la figura 2.8. La señal de entrada $x[n]$ se multiplica por el código C/A del satélite i -ésimo (la fase inicial de este código C/A es n_{0i}) y luego se divide en dos ramas las cuales se multiplicarán por dos réplicas en cuadratura de la portadora en frecuencia intermedia. Las réplicas seno y coseno están en la frecuencia Ω en torno a la frecuencia intermedia central desplazadas una cantidad Δf_{dl} que se asimila al corrimiento por efecto Doppler.

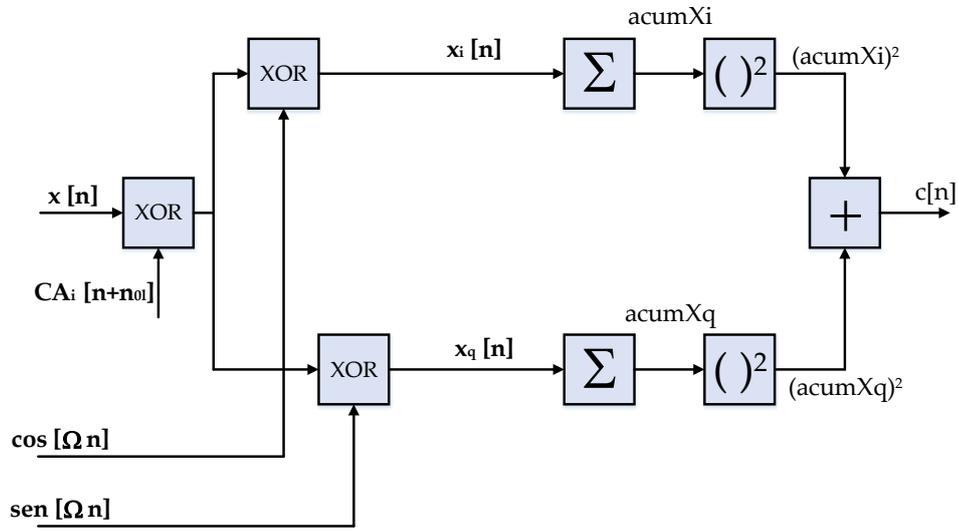


FIGURA 2.8: Esquema general diagrama de búsqueda

A continuación se tratará analíticamente el problema. La señal de entrada $x[n]$ (en el siguiente análisis se ha normalizado la señal para simplificar las ecuaciones, sin pérdida de generalidad) se multiplica con la copia local del código C/A del satélite que se pretende buscar ($CA_j[n]$) y las componentes en cuadratura del oscilador local obteniendo:

$$x_{iCA}[n] = CA_i[n + n_{0i}] \cdot CA_j[n + n_{0j}] \cdot \cos[2\pi(f_i + f_d)T_s n + \phi_0] \cos[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.6)$$

$$x_{qCA}[n] = CA_i[n + n_{0i}] \cdot CA_j[n + n_{0j}] \cdot \cos[2\pi(f_i + f_d)T_s n + \phi_0] \sin[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.7)$$

donde:

- $x_{iCA}[n]$: Señal de entrada multiplicada por copia de código C/A local y oscilador local en fase
- $x_{qCA}[n]$: Señal de entrada multiplicada por copia de código C/A local y oscilador local en cuadratura
- n_{0i} : Desplazamiento de la copia del código C/A local
- Δf_{dl} : Desplazamiento en frecuencia del oscilador local

Utilizando las identidades trigonométricas para el producto de funciones armónicas [7] se obtiene:

$$x_{iCA}[n] = \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \{ \cos[2\pi(2f_i + f_d + \Delta f_{dl})T_s n + \phi_0] + \frac{1}{2} \cdot \cos[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \} \quad (2.8)$$

$$x_{qCA}[n] = \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \{ \sen[2\pi(2f_i + f_d + \Delta f_{dl})T_s n + \phi_0] - \frac{1}{2} \cdot \sen[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \} \quad (2.9)$$

Las señales descriptas en las ecuaciones 2.8 y 2.10 se acumulan durante un período determinado (usualmente la duración de un período C/A equivalente a 1ms) obteniéndose:

$$AcumX_i[n_{ol}, \Delta f_{dl}] = \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \cos[2\pi(2f_i + f_d + \Delta f_{dl})T_s n + \phi_0] \quad (2.10)$$

$$+ \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \cos[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0]$$

$$AcumX_q[n_{ol}, \Delta f_{dl}] = \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \sen[2\pi(2f_i + f_d + \Delta f_{dl})T_s n + \phi_0] \quad (2.11)$$

$$+ \sum_{n=0}^{N-1} -\frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \sen[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \quad (2.12)$$

Los primeros términos de cada una de las ecuaciones 2.11 y 2.12, consisten en una señal centrada en la frecuencia $2f_i$ y tienden a cero al ser integrados ya que los acumuladores tienen una respuesta de tipo pasabajos[8], de tal forma que:

$$AcumX_i[n_{ol}, \Delta f_{dl}] = \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \cos[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \quad (2.13)$$

$$AcumX_q[n_{ol}, \Delta f_{dl}] = - \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n + n_0] \cdot CA_j[n + n_{0l}] \cdot \sen[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \quad (2.14)$$

Finalmente, los valores acumulados se elevan al cuadrado y se suman, obteniéndose:

$$C[n_{ol}, \Delta f_{dl}] = \{AcumX_i[n_{ol}, \Delta f_{dl}]\}^2 + \{AcumX_q[n_{ol}, \Delta f_{dl}]\}^2 \quad (2.15)$$

La ecuación 2.15 representa la ecuación de salida general para el método de búsqueda serial. Éste consiste en ir variando los parámetros n_{ol} y Δf_{dl} con el fin de hallar si un satélite se encuentra presente o no, y de estarlo, su fase relativa de código C/A (n_{ol}) y desplazamiento respecto de la frecuencia central (Δf_{dl}). Como se demostrará, el valor de $C[n_{ol}, \Delta f_{dl}]$ representa un «valor testigo» el cual nos indicará si las condiciones anteriores se cumplen. A continuación, se detalla el caso general de búsqueda en el cual no se introduce ninguna restricción sobre los parámetros n_{ol} y Δf_{dl} . A partir de ese caso, se analizan las implicancias de los casos particulares que surgen de que uno o ambos valores de n_{ol} y Δf_{dl} coincidan con los valores homólogos de la señal de entrada y su importancia para la técnica de búsqueda.

2.2.1. Caso general

En el caso general, se evalúa el valor de la función $C[n_{ol}, \Delta f_{dl}]$ (definida en la ecuación 2.15) para un dominio bidimensional compuesto por una cantidad finita de pares $(n_{ol}, \Delta f_{dl})$.

Para tener una idea más cabal sobre el valor que esperamos de la función $C[n_{ol}, \Delta f_{dl}]$:

$$C[n_{ol}, \Delta f_{dl}] = \{AcumX_i[n_{ol}, \Delta f_{dl}]\}^2 + \{AcumX_q[n_{ol}, \Delta f_{dl}]\}^2 \quad (2.15)$$

con:

$$AcumX_i[n_{ol}, \Delta f_{dl}] = \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n+n_0] \cdot CA_j[n+n_{0l}] \cdot \cos[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \quad (2.13)$$

$$AcumX_q[n_{ol}, \Delta f_{dl}] = - \sum_{n=0}^{N-1} \frac{1}{2} \cdot CA_i[n+n_0] \cdot CA_j[n+n_{0l}] \cdot \sen[2\pi(f_d - \Delta f_{dl})T_s n + \phi_0] \quad (2.14)$$

Las ecuaciones 2.13 y 2.14 pueden reescribirse de forma más compacta utilizando el enfoque de señales complejas [9].

Si se considera que la señal de entrada $x[n]$ puede «mezclarse» con un «oscilador local complejo» de la siguiente forma:

$$x_{iq}[n] = x[n] \cdot e^{j2\pi(f_i + \Delta f_{dl})T_s n} \quad (2.16)$$

$$x_{iq}[n] = x_i + jx_q \quad (2.17)$$

donde:

$$x_i = x[n] \cdot \cos[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.18)$$

$$x_q = x[n] \cdot \text{sen}[2\pi(f_d + \Delta f_{dl})T_s n]; \quad (2.19)$$

Así, la señal $x_{iq}[n]$ es la representación compleja de la señal de entrada $x[n]$ desplazada en frecuencia $(f_i + \Delta f_{dl})$ hacia la izquierda. El espectro en frecuencia constará de dos componentes: una en la frecuencia $\Delta f_d - \Delta f_{dl}$ (más cercana al valor de frecuencia 0 o valor DC) y otra ubicada en la frecuencia $2f_i + \Delta f_d + \Delta f_{dl}$ que se extinguirá luego de la integración. La señal $x_{iq}[n]$, usualmente se denomina «señal de banda base compleja» ya que corresponde a la señal en frecuencia intermedia mezclada con el «oscilador local complejo» y filtrada. Si ahora, se define a la señal conjugada de $x_{iq}[n]$ como $x_{iq}^*[n]$, se multiplica por la copia local del código C/A $CA_j[n + n_{ol}]$ y se acumula de la misma forma que se hizo anteriormente, se obtiene:

$$\sum_{n=0}^{N-1} x_{iq}^*[n] CA_j[n + n_{ol}] = \sum_{n=0}^{N-1} x_i[n] CA_j[n + n_{ol}] - j \sum_{n=0}^{N-1} x_q[n] CA_j[n + n_{ol}] \quad (2.20)$$

La sumatoria expresada en la ecuación 2.20 es el producto de correlación entre $x_{iq}[n]$ y la copia local del código C/A, tal como se desprende de comparar con la ecuación 2.1:

$$c[n_{ol}, \Delta f_{dl}] = x[n] \otimes CA_j[n] \triangleq \sum_{l=0}^{N-1} x_{iq}^*[l] CA_j[l + n] \quad (2.21)$$

El módulo al cuadrado de la función $c[n_{ol}, \Delta f_{dl}]$ equivale a:

$$|c[n_{ol}, \Delta f_{dl}]|^2 = \left| \sum_{n=0}^{N-1} x_i[n] CA_j[n + n_{ol}] - j \sum_{n=0}^{N-1} x_q[n] CA_j[n + n_{ol}] \right|^2 \quad (2.22)$$

$$|c[n_{ol}, \Delta f_{dl}]|^2 = \left| \sum_{n=0}^{N-1} x_i[n] CA_j[n + n_{ol}] \right|^2 + \left| \sum_{n=0}^{N-1} x_q[n] CA_j[n + n_{ol}] \right|^2 \quad (2.23)$$

$$|c[n_{ol}, \Delta f_{dl}]|^2 = \left| \sum_{n=0}^{N-1} x_i[n] CA_j[n + n_{ol}] \right|^2 + \left| \sum_{n=0}^{N-1} x_q[n] CA_j[n + n_{ol}] \right|^2 \quad (2.24)$$

$$|c[n_{ol}, \Delta f_{dl}]|^2 = \text{Acum}X_i^2[n_{ol}, \Delta f_{dl}] + \text{Acum}X_q^2[n_{ol}, \Delta f_{dl}] = C[n_{ol}, \Delta f_{dl}] \quad (2.25)$$

de donde se demuestra que los valores hallados según el esquema propuesto en la figura 2.8 equivale a obtener el cuadrado del módulo del producto de correlación entre la señal de banda base compleja y el código C/A del satélite buscado.

A modo de ejemplo, se generó sintéticamente una señal GPS en frecuencia intermedia utilizando MATLAB ($f_i = 4,092MHz$) en base a la bibliografía [10] con un desplazamiento Doppler igual a $\Delta f_d = 1000Hz$ y un código C/A desplazado 800 chips a derecha. La frecuencia de muestreo empleada fue $f_s = 16,368MHz$, es decir 16 muestras por chip. Se empleó la ecuación 2.15 para calcular el valor de la función de salida del módulo de búsqueda en el dominio de $\Delta f_{dl} \in [-4000; 4000]Hz$ y $n_{ol} \in [0; 16367]$. La función de salida se grafica en la figura 2.9.

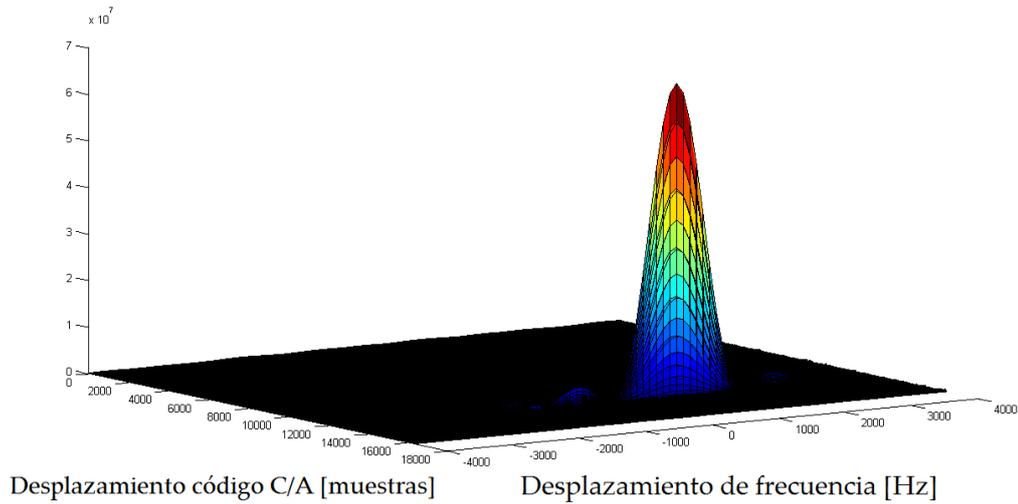


FIGURA 2.9: Gráfico de $C[n_{ol}, \Delta f_{dl}]$

2.2.2. Caso particular: Código C/A local en fase con código C/A de la señal de entrada

En el caso de que la réplica local de código C/A se encuentra en fase respecto del código C/A de la señal de entrada, equivale a asumir la restricción $n_0 = n_{ol}$, de forma tal que la expresión de $C[n_{ol}, \Delta f_{dl}]$ se reduce a partir de la ecuación 2.15 a:

$$C[n_{ol} = n_0, \Delta f_{dl}] = \left\{ \sum_{n=0}^{N-1} 1/2 \cdot \cos[2\pi(\Delta f_d - \Delta f_{dl})T_s n + \phi_0] \right\}^2 + \left\{ \sum_{n=0}^{N-1} 1/2 \cdot \sen[2\pi(\Delta f_d - \Delta f_{dl})T_s n + \phi_0] \right\}^2 \quad (2.26)$$

$$C[n_{ol} = n_0, \Delta f_{dl}] = 1/4 \left| \sum_{n=0}^{N-1} e^{j[2\pi(\Delta f_d - \Delta f_{dl})T_s n + \phi_0]} \right|^2 \quad (2.27)$$

Es decir, en este caso se está integrando en un intervalo de N muestras una señal compleja cuya única componente de frecuencia vale $(\Delta f_d - \Delta f_{dl})$. Si el período de esta señal es igual al intervalo de integración, $C[n_{ol}, \Delta f_{dl}]$ valdrá 0. Así, considerando un período de integración igual a 1ms, a medida que la frecuencia $(\Delta f_d - \Delta f_{dl})$ exceda (en módulo a 1kHz), el valor de $C[n_{ol}, \Delta f_{dl}]$ tenderá a 0.

En la figura 2.10, se grafica el valor de $C[n_{ol}, \Delta f_{dl}]$ para este caso en particular, tomando como parámetro $\Delta f_d = 1000Hz$.

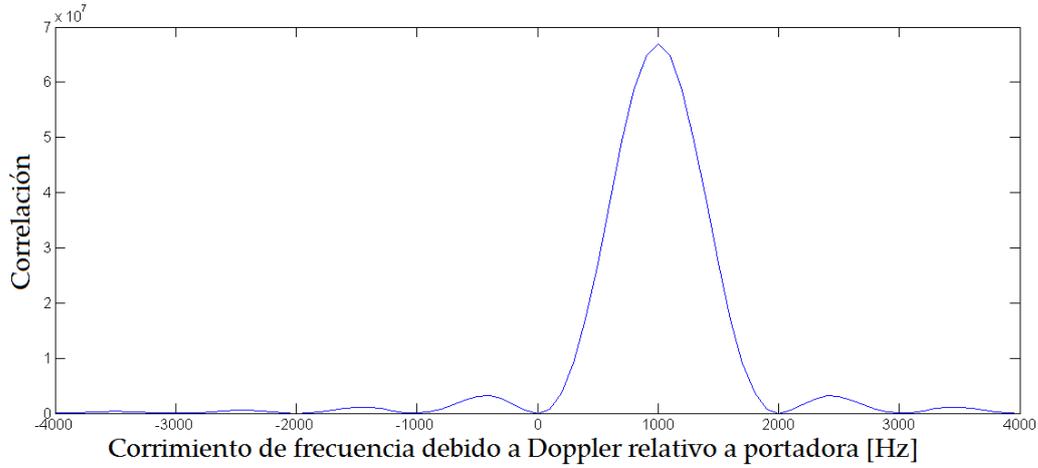


FIGURA 2.10: Gráfico de $C[n_{ol} = n_0, \Delta f_{dl}]$

2.2.3. Caso particular: Oscilador local con igual frecuencia de la señal de frecuencia intermedia

En el caso de que la frecuencia del oscilador local sea igual a la portadora en frecuencia intermedia de la señal de entrada, equivale a aplicar la restricción $\Delta f_d = \Delta f_{dl}$ de forma tal que la expresión de $C[n_{ol}, \Delta f_{dl}]$ a partir de 2.15 se reduce a:

$$C[n_{ol}, \Delta f_{dl} = \Delta f_d] = \left\{ \sum_{n=0}^{N-1} 1/2 CA_i[n + n_0] CA_j[n + n_{0l}] \cos[\phi_0] \right\}^2 + \left\{ \sum_{n=0}^{N-1} 1/2 CA_i[n + n_0] CA_j[n + n_{0l}] \sen[\phi_0] \right\}^2 \quad (2.28)$$

$$C[n_{ol}, \Delta f_{dl} = \Delta f_d] = 1/4 \left\{ \sum_{n=0}^{N-1} CA_i[n + n_0] CA_j[n + n_{0l}] \right\}^2 \quad (2.29)$$

Asumiendo que en la ecuación 2.30, $i = j$ y operando podemos obtener:

$$C[n_{ol}, \Delta f_{dl} = \Delta f_d] = 1/4 \left\{ \sum_{n=0}^{N-1} CA_i[n + n_0] CA_i[n + n_0 + \Delta n] \right\}^2 \quad (2.30)$$

donde se ha sustituido:

$$\Delta n = n_{0l} - n_0 \quad (2.31)$$

siendo Δn la diferencia de fase entre el código C/A de la señal de entrada y el código C/A generado localmente. De acuerdo a la ecuación 2.1, la ecuación 2.30 puede escribirse simplemente como:

$$C[n_{ol}, \Delta f_{dl} = \Delta f_d] = 1/4 c_{CA_i}^2[\Delta n] \quad (2.32)$$

donde:

- $c_{CA_i}[\Delta n]$: Producto de autocorrelación de $CA_i[n]$

Cómo ya se ha indicado en la sección 2.1, la función de autocorrelación del código C/A será máxima para $\Delta n = 0$, es decir, cuando la fase del código C/A local y el de la señal de entrada sean iguales.

En la figura 2.11 se muestra un gráfico a modo de ejemplo, donde el código C/A local y el de la señal de entrada están separados 800 chips. La señal de entrada es una señal ideal generada sintéticamente para realizar la prueba. Para facilitar la interpretación se ha normalizado el eje de abscisas para indicar la variable directamente en chips.

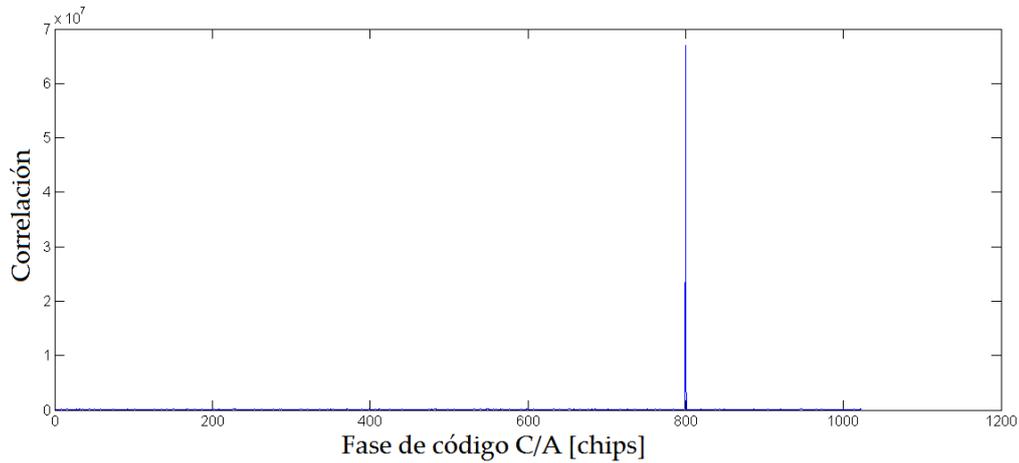


FIGURA 2.11: Gráfico de $C[n_{ol}, \Delta f_{dl} = \Delta f_d]$

2.2.4. Caso particular: Coincidencia de fase de C/A y desplazamiento de frecuencia de la señal local y la señal de entrada

En el caso que se cumplan las condiciones referenciadas en las secciones 2.2.2 y 2.2.3, equivale a la aplicación de las restricciones $n_0 = n_{ol}$ y $\Delta f_d = \Delta f_{dl}$ simultáneamente, de forma tal que la expresión de $C[n_{ol}, \Delta f_{dl}]$ a partir de 2.15 se reduce a:

$$C[n_{ol} = n_0, \Delta f_{dl} = \Delta f_d] = \left(\frac{N}{2}\right)^2 \quad (2.33)$$

En este caso, estamos en presencia del máximo de la función $C[n_{ol}, \Delta f_{dl}]$ cuyo valor será una constante dependiendo solamente de la cantidad de muestras en el intervalo de acumulación considerado (N).

2.3. Diseño del lazo de seguimiento

En esta sección se realizará una síntesis de la estructura y funcionamiento propuestas para el lazo de seguimiento (*tracking*). En términos generales se utilizará como bibliografía de base el capítulo 5 de [11] y otras referencias específicas que se citarán puntualmente.

Como se ha explicado en la sección 2.2 resulta fundamental hallar los valores de frecuencia central y fase de código C/A de la señal del satélite cuyo mensaje se quiere obtener. Este primer paso es necesario para configurar inicialmente la etapa que se encarga de realizar el seguimiento continuo de la señal, es decir, de mantener sincronizada la señal de entrada. Esta operación se realiza generando réplicas locales tanto de la portadora en frecuencia intermedia (FI) como de la secuencia de código C/A del satélite; las cuales deben mantenerse en fase respecto de la señal de entrada. De esta manera, la «mezcla» de la señal de entrada con las réplicas, según lo analizado en la sección 2.2.4, elimina tanto la portadora como la secuencia de código C/A de la señal de entrada dejando disponible el mensaje de navegación. El proceso de seguimiento es esencialmente bidimensional, esto es, tanto la frecuencia de la portadora como la fase del código C/A deben «seguirse» simultáneamente; en caso contrario no será posible obtener el mensaje de navegación. En ese caso, el proceso de búsqueda debe iniciarse nuevamente para obtener una configuración inicial para el algoritmo de seguimiento.

El enfoque utilizado en este trabajo es el de una arquitectura de receptor digital, en la cual un *frontend* se encarga de recibir a través de su antena las señales de GPS, para luego filtrarlas, convertirlas a frecuencia intermedia y digitalizarlas. Luego, la señal digitalizada en FI se procesa a través de módulos digitales específicos que se explicarán detalladamente en el capítulo 3. En la figura 2.12 se observa un diagrama en bloques general del lazo de seguimiento, donde se aprecia que la señal en FI digitalizada $x[n]$ se mezcla con una réplica local de la portadora en FI, en fase y en cuadratura. Luego, cada uno de los productos ($x_I[n]$ y $x_Q[n]$) se vuelve a combinar con tres réplicas del código C/A del satélite: *Early*, *Prompt* y *Late*. Las réplicas representan secuencias idénticas, salvo que las secuencias *Early* y *Late* están adelantadas y atrasadas medio chip, respectivamente, de la secuencia *Prompt*. Las señales resultantes ($x_{IE}[n]$, $x_{IP}[n]$, $x_{IL}[n]$, $x_{QE}[n]$, $x_{QP}[n]$ y $x_{QL}[n]$) se acumulan durante un tiempo equivalente a un múltiplo del período de código C/A (1ms).

Según lo indicado en la ecuación 2.1, la multiplicación y posterior acumulación de dos señales representa su producto de correlación. Así, se obtienen los valores de correlación I_E , I_P , I_L , Q_E , Q_P y Q_L que se utilizarán para controlar los módulos que generan las réplicas locales de la portadora (oscilador digital) y del código C/A (generador de código C/A).

Para analizar el procesamiento de las señales se partirá de la ecuación 2.5 que representa la señal de entrada. Ésta se reproduce aquí por comodidad, conservando la nomenclatura de la sección 2.2:

$$x[n] = A \cdot D[n] \cdot CA_i[n + n_0] \cdot \cos[2\pi(f_i + \Delta f_d)T_s n + \phi_0] \quad (2.5)$$

En primer lugar, se analizarán las implicancias de los valores I_E , I_P , I_L , Q_E , Q_P y Q_L para obtener una estimación de la diferencia de fase y frecuencia entre la

Considerando que $CA_i[n + n_0] \cdot CA_i[n + n_{0l}] = 1$ en esta situación, las ecuaciones 2.34 y 2.35 se reducen a:

$$x_{IP}[n] = \cos[2\pi(f_i + f_d)T_s n + \phi_0] \cos[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.36)$$

$$x_{QP}[n] = \cos[2\pi(f_i + f_d)T_s n + \phi_0] \operatorname{sen}[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.37)$$

Luego de la operación de integración a través de los acumuladores, se puede demostrar que I_P y Q_P valen:

$$I_P[n] = 1/2 \cdot \cos[2\pi(f_d - \Delta f_{dl})T'_s n + \phi_0] \quad (2.38)$$

$$Q_P[n] = -1/2 \cdot \operatorname{sen}[2\pi(f_d - \Delta f_{dl})T'_s n + \phi_0] \quad (2.39)$$

donde T'_s es el intervalo de acumulación elegido.

Analizando 2.38 y 2.39 se observa que las señales $I_P[n]$ y $Q_P[n]$ son armónicas y de frecuencia igual a $f_d - \Delta f_{dl}$, que representa la diferencia de frecuencias entre la portadora de la señal de entrada y la réplica local. Además, ambas poseen una fase inicial ϕ_0 , equivalente al desfase constante entre la portadora de la señal de entrada y la local.

Tomando como base las ecuaciones 2.38 y 2.39 se puede obtener una expresión para la variación de fase entre la portadora de la señal de entrada y la local $\Delta\phi[n]$:

$$\Delta\phi[n] = -\operatorname{atan} \left(\frac{Q_P[n]}{I_P[n]} \right) = \left[2\pi(f_d - \Delta f_{dl})T'_s n + \phi_0 \right]_{(-\pi/2, \pi/2)} \quad (2.40)$$

donde la notación $[\cdot]_{(-\pi/2, \pi/2)}$ significa «acotado entre $-\pi/2$ y $+\pi/2$ ».

En la ecuación 2.40 se observa que la función $\Delta\phi[n]$ varía linealmente con pendiente proporcional a la diferencia de frecuencias $f_d - \Delta f_{dl}$ y ordenada al origen igual a fase inicial ϕ_0 . La función $\Delta\phi[n]$ es un estimador del error de fase de portadora entre la señal de entrada y la réplica local valiendo cero en una situación de perfecta sincronía en fase ($\phi_0 = 0$) y frecuencia ($f_d - \Delta f_{dl} = 0$).

2.3.2. Estimación de la diferencia de fase de código C/A entre la señal de entrada y la señal local

A fin de estimar la diferencia de fase de código C/A entre la señal de entrada y la señal local es preciso tener presente las propiedades de autocorrelación de los códigos C/A explicadas en la sección 2.1.2. En la figura 2.13 se indica el valor de correlación normalizado del código C/A de entrada con las diferentes réplicas (*Early*, *Prompt* y *Late*) teniendo en cuenta que la señal de entrada está en fase con la réplica *Prompt*.

Se observa de la figura 2.13 que el valor de la correlación para la réplica *Prompt* es máximo, mientras que para las réplicas *Early* y *Late* es igual y coincidente con la mitad del valor máximo. En esta condición, el generador de código C/A y la

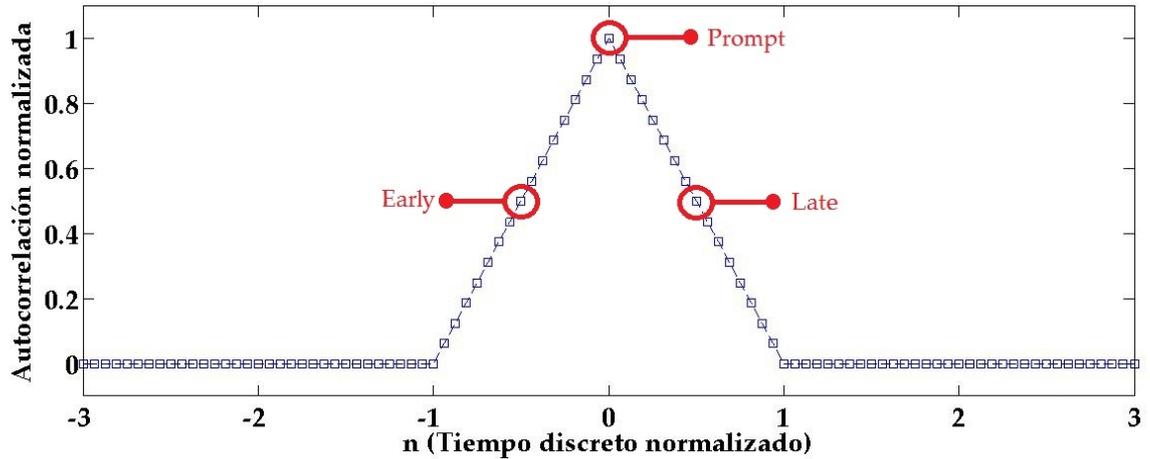


FIGURA 2.13: Valores de correlación normalizados para los códigos *Early*, *Prompt* y *Late* (señal en fase)

señal de entrada se encuentran en perfecta sincronía. Si se supone que la señal de entrada se atrasa respecto de la local, el valor de correlación de la señal de entrada con la réplica *Late* aumentará de valor, y simultáneamente, la correlación de la réplica *Early* disminuirá tal como se indica en la figura 2.14.

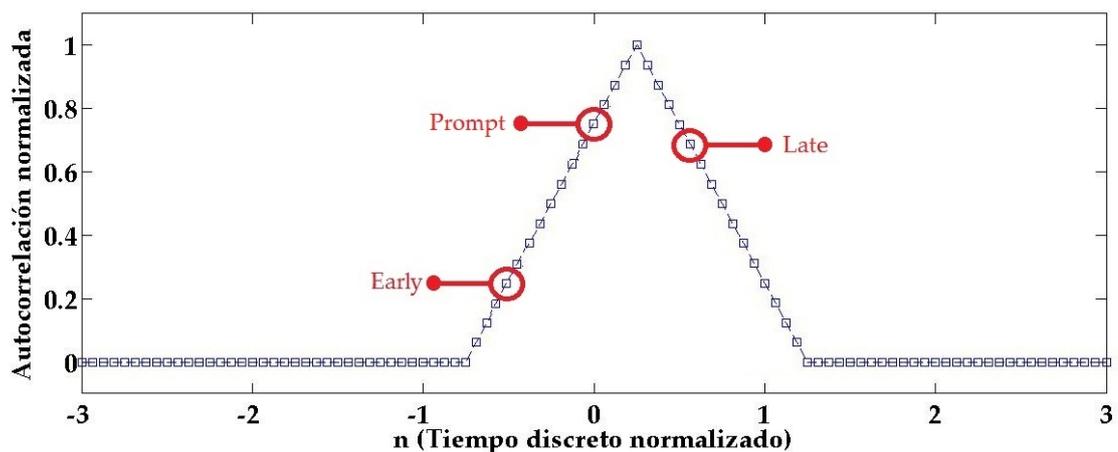


FIGURA 2.14: Valores de correlación normalizados para los códigos *Early*, *Prompt* y *Late* (señal atrasada)

Así se puede inferir cualitativamente que: la diferencia entre los valores de correlación relativos a la réplica *Early* y la réplica *Late* se puede utilizar como estimador de la diferencia de fase de código C/A entre la señal de entrada y la réplica local. Existen diversas formas de formular esta diferencia, las cuales se explican en detalle en la referencia[11], pero en este trabajo se eligió emplear el discriminador expresado en la ecuación 2.41 el cual representa una buena solución de compromiso entre la performance del estimador y su complejidad de implementación en términos de costo computacional.

$$\Delta n = \frac{1}{2} \frac{E - L}{E + L} \quad (2.41)$$

donde:

- Δn : Diferencia fase de código C/A entre señal de entrada y réplica *Prompt*
- $E = \sqrt{I_E^2 + Q_E^2}$: Correlación réplica *Early*
- $L = \sqrt{I_L^2 + Q_L^2}$: Correlación réplica *Late*

Como se aprecia en la figura 2.15 los valores E y L se obtienen a partir de los bloques detectores de envolvente a partir de las entradas I_E , Q_E , I_L y Q_L .

2.3.3. Controladores de lazo de portadora y de código

En esta sección se resume la estructura de los controladores a partir de las ecuaciones 2.40 y 2.41.

En base a las consideraciones desarrolladas en la sección anterior, se puede definir una estructura del controlador del lazo de seguimiento como se indica en la figura 2.15 el cual se implementa dentro del bloque denominado «sistema de control» de la figura 2.12.

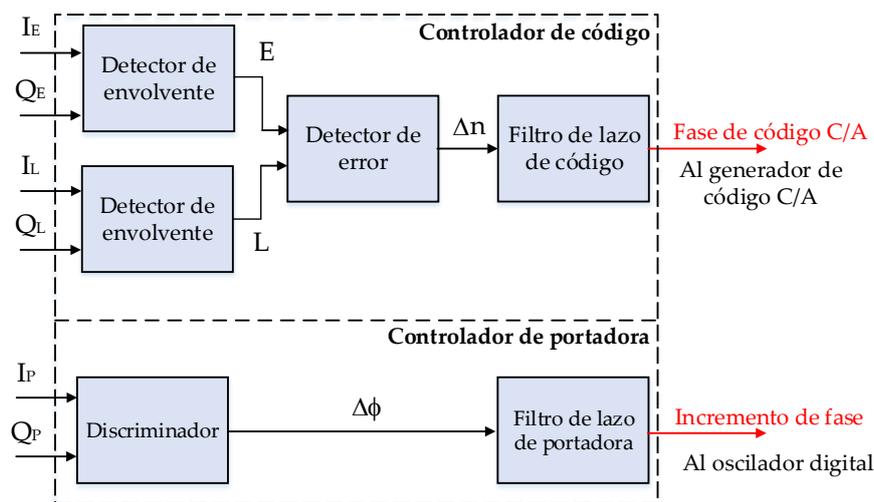


FIGURA 2.15: Estructura en bloques del controlador del lazo de seguimiento

El controlador consta de dos partes:

- **Controlador de portadora:** Recibe a su entrada los valores de I_P y Q_P y por medio de la ecuación 2.40 genera una estimación del error de fase entre la señal de entrada y la portadora local $\Delta\phi$. Este error se procesa mediante el bloque denominado «filtro de lazo de portadora» que genera una señal de control que establece el valor de frecuencia del oscilador digital local.
- **Controlador de código:** Recibe a su entrada los valores I_E , Q_E , I_L y Q_L . Los bloques denominados «detector de envolvente» realizan las operaciones:
 - $E = \sqrt{I_E^2 + Q_E^2}$
 - $L = \sqrt{I_L^2 + Q_L^2}$

que luego se utilizan para obtener el error de fase a través del «detector de error» por medio de la expresión para el error de fase del código C/A (ecuación 2.41).

2.3.4. Estudio y simulación del lazo de portadora

La simulación del lazo de portadora es relevante por los siguientes motivos:

- tener un modelo de comparación con el que contrastar el módulo de seguimiento desarrollado.
- estudiar el desempeño del sistema en diferentes condiciones: niveles de ruido, variación de los parámetros de la señal de entrada, entre otros.
- estudiar modificaciones en la estructura del lazo, en particular, de los discriminadores y del filtro de lazo.
- profundizar el conocimiento de las señales y las operaciones involucradas en el modelo.

El lazo de portadora se puede modelar en bloques como se presenta en la figura 2.16. La señal de entrada $x[n]$ se multiplica por las réplicas de la portadora local, seno y coseno, para obtener las señales $x_I[n]$ y $x_Q[n]$. Estas señales se acumulan un período de tiempo T'_s resultando las señales $I[n]$ y $Q[n]$, a partir de las cuales se obtiene una expresión para el error de fase $\Delta\phi[n]$ a partir de 2.40. Finalmente, el error de fase $\Delta\phi[n]$ se procesa por medio de un filtro para obtener la señal de control $\Delta f[n]$ la cual se emplea para variar la frecuencia del oscilador digital.

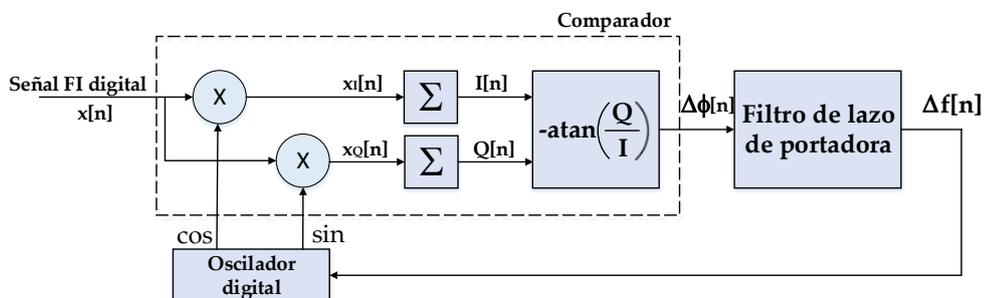


FIGURA 2.16: Estructura en bloques del lazo de portadora

En el esquema planteado (figura 2.16) el bloque «comparador» simultáneamente obtiene una expresión del error de fase y a la vez actúa como convertidor entre dos dominios temporales: a la entrada la tasa de muestreo depende de la frecuencia de operación del conversor analógico digital (T_s) mientras que la salida presenta una tasa distinta que depende del período de acumulación (T'_s). Si se toma como ejemplo un período de integración de $1ms$, la frecuencia de muestreo de $\Delta\phi[n]$ será $1kHz$, mientras que la señal de entrada, como se verá, se muestreará a una frecuencia de $16,368MHz$. La diferencia entre ambos tiempos de muestreo permite formas de implementación distintas para cada bloque. Por ejemplo, los bloques de generación de señales digitales, mezcla y acumulación suelen concretarse por medio de lógica digital programable por su mayor exigencia de velocidad; los bloques del discriminador y el filtro del lazo pueden llevarse a cabo utilizando un microprocesador.

Desde el punto de vista del análisis de señales y sistemas, el lazo de portadora se puede modelar como se indica en la figura 2.17.

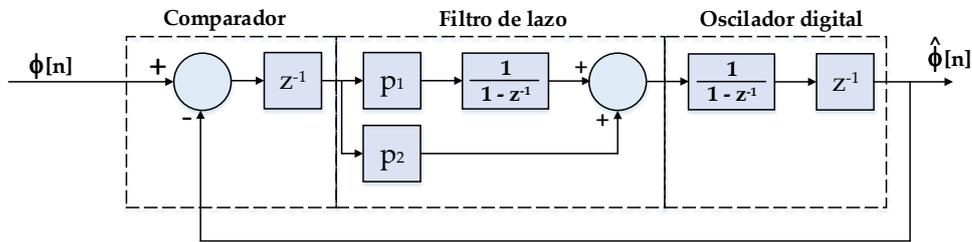


FIGURA 2.17: Modelo del lazo de portadora en el dominio Z

El bloque comparador calcula la diferencia entre la fase de la portadora de la señal de entrada $\phi[n]$ y la fase de la portadora local $\hat{\phi}[n]$. El retardo de una muestra z^{-1} se debe a que la comparación requiere la integración de un período completo, estando la estimación de la diferencia disponible retrasada una muestra. Para el filtro de lazo se propone una estructura de primer orden cuya salida contiene un término proporcional al error (p_2) y otro proporcional a su acumulación ($p_1 \cdot \frac{1}{1-z^{-1}}$). Finalmente la salida del lazo se aplica a la entrada del oscilador digital que puede modelarse como un integrador ($\frac{1}{1-z^{-1}}$) más un retardo z^{-1} . El comportamiento integrador del oscilador se debe a su implementación la cual se tratará en detalle en la sección 3.6. El retardo refleja el hecho que para cambiar la fase del oscilador local se requiere un tiempo adicional de una muestra para reconfigurar el dispositivo[12].

El análisis del modelo en el dominio de Z es imprescindible para comprender la respuesta del lazo, en particular, su comportamiento en régimen permanente ($n \rightarrow \infty$) ante diferentes tipos de referencia $\phi[n]$. La dependencia funcional de la referencia $\phi[n]$ respecto del tiempo depende esencialmente de la dinámica del sistema.

Por ejemplo, un lazo de seguimiento deberá ser capaz de sincronizarse con la frecuencia de portadora a partir de una referencia inicial. Usualmente, el valor inicial de frecuencia del lazo es cercano a la frecuencia real de la portadora, pero no idéntico. Esta diferencia se puede modelar como si el sistema fuese excitado por una función escalón de frecuencia. Siendo la fase, la integral de la frecuencia, esto implica que el lazo debe poder seguir sin error en estado estacionario una función rampa de fase. De la teoría de errores de los sistemas de control en estado estacionario [13] se deduce que para que el sistema presente error nulo en régimen permanente frente a una referencia de tipo rampa el sistema debe ser al menos de tipo 2, es decir, debe haber dos integradores en la cadena directa. Este es el caso del presente trabajo, donde se puede observar en la figura 2.17 que un integrador es aportado por el filtro del lazo y otro por la respuesta del oscilador digital.

En sistemas de alta dinámica, por ejemplo cohetes; surgen perturbaciones en el seguimiento de la frecuencia debido a la variación por efecto Doppler de la aceleración de estos móviles. En este tipo de casos, se requiere la implementación de sistemas de tipo 3 o superior [12]. El error en régimen permanente para el modelo propuesto se analizará a continuación.

El sistema de la figura 2.17 se puede representar de forma genérica según se indica en la figura 2.18, cuya función de transferencia queda expresada por la ecuación 2.42:

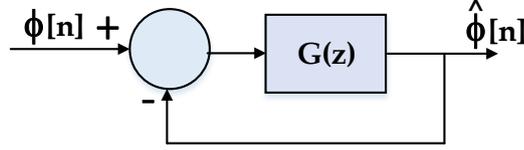


FIGURA 2.18: Estructura general del sistema realimentado

$$H(z) = \frac{\hat{\phi}(z)}{\phi(z)} = \frac{G(z)}{1 + G(z)} \quad (2.42)$$

donde:

- $H(z)$: Función de transferencia del sistema.
- $\hat{\phi}(z)$: Fase del oscilador digital local.
- $\phi(z)$: Fase de portadora de entrada.
- $G(z)$: Función de transferencia directa, incluyendo el comparador, filtro de lazo y oscilador digital.

El error de la referencia respecto de la señal de entrada $E(z)$ puede expresarse como:

$$E(z) = \hat{\phi}(z) - \phi(z) \quad (2.43)$$

Reemplazando $\hat{\phi}(z)$ de 2.42 en 2.43 se obtiene una expresión del error en función de $G(z)$ y la referencia $\phi(z)$:

$$E(z) = \frac{-1}{1 + G(z)} \phi(z) \quad (2.44)$$

Se desea hallar el error en régimen permanente $e[\infty]$, por lo que se aplica el teorema del valor final[8] a 2.44:

$$e[\infty] = \lim_{n \rightarrow \infty} e[n] = \lim_{z \rightarrow 0} (z - 1)e(z) = \lim_{z \rightarrow 0} -\frac{z - 1}{1 + G(z)} \phi(z) \quad (2.45)$$

donde:

- $e[n]$: Antitransformada Z de $E(z)$

La ecuación 2.45 se utiliza para calcular el error en régimen permanente $e[\infty]$ de forma general para cualquier tipo de referencia $\phi(z)$ y cualquier transferencia directa de lazo $G(z)$. Para la estructura propuesta según el diagrama 2.17, la función de transferencia es:

$$G(z) = p_2 \frac{z + \frac{p_1 - p_2}{p_2}}{(z - 1)^2} \quad (2.46)$$

La referencia $\phi(z)$ según se explicó corresponde a la transformada Z de una función rampa de fase[8]:

$$\phi(z) = A \frac{z}{(z - 1)^2} \quad (2.47)$$

donde:

- A : Pendiente de la rampa

Reemplazando 2.46 y 2.47 en 2.45 se obtiene que:

$$e[\infty] = \lim_{z \rightarrow 0} - \frac{z - 1}{1 + G(z)} \phi(z) = 0 \quad (2.48)$$

de donde se demuestra que la estructura de lazo propuesta es capaz de seguir la referencia rampa de fase (equivalente a un escalón de frecuencia) con error nulo en régimen.

Para simular el comportamiento del lazo de portadora, se creó un modelo en software (Octave) la estructura de la figura 2.16, implementando el filtro de lazo de la figura 2.17. La simulación permite generar una señal de entrada armónica la cual puede variarse en fase y frecuencia para generar diferentes condiciones de prueba. En las subsecciones siguientes, se presenta el comportamiento del lazo de seguimiento para dos situaciones distintas: excitación del lazo frente a escalón de frecuencia y escalón de fase.

Respuesta del sistema frente a escalón de frecuencia

Se realizó una simulación para comprobar la respuesta del sistema frente a una variación instantánea de la frecuencia de la señal de entrada (escalón de frecuencia) la cual puede observarse en la figura 2.19. El cambio abrupto de frecuencia se produce a los $50ms$ del inicio de la simulación y está indicado con una línea punteada vertical. El ensayo comienza con el lazo «enganchado» y el cambio de frecuencia genera un defasaje instantáneo. Esta perturbación puede observarse en la figura 2.20 en el aumento abrupto del error de fase ($\Delta\phi$) que luego es corregido por el sistema de control.

En las figuras 2.21 y 2.22 se indica la evolución temporal de $I[n]$ y $Q[n]$. El valor de $I[n]$ es máximo y constante hasta la perturbación, volviendo a su valor inicial luego de la corrección. De forma homóloga, el valor de $Q[n]$ es nulo al comienzo de la simulación (condición de «enganche») y luego de la perturbación vuelve a serlo. La variación de frecuencia del oscilador digital local que introduce el sistema re-alimentado se indica en la figura 2.23, donde se observa una sobrecompensación de la respuesta de frecuencia local para luego establecerse en el valor de régimen de $50Hz$.

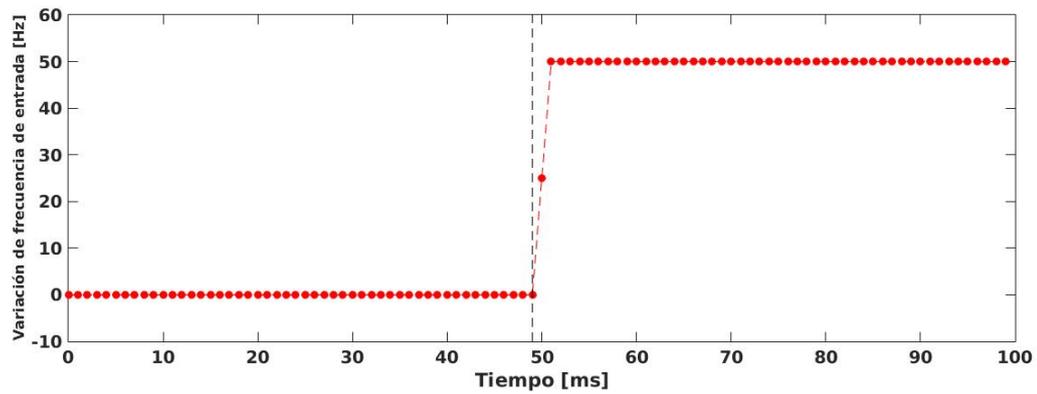


FIGURA 2.19: Cambio de la referencia de frecuencia de la señal de entrada

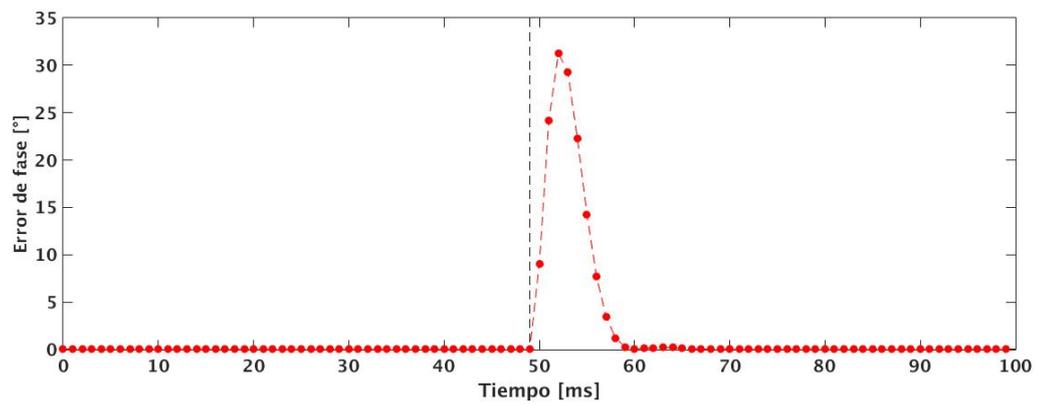
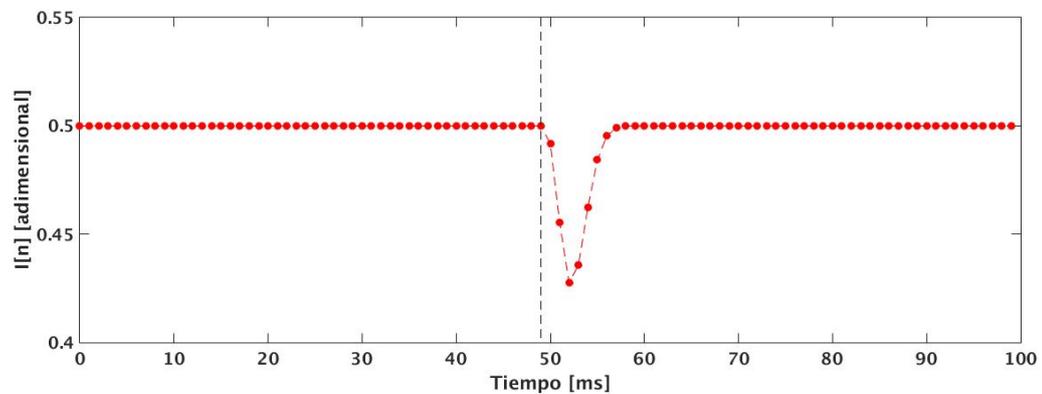
FIGURA 2.20: Error de fase instantáneo ($\Delta\phi$)

FIGURA 2.21: Correlación del canal en fase (I)

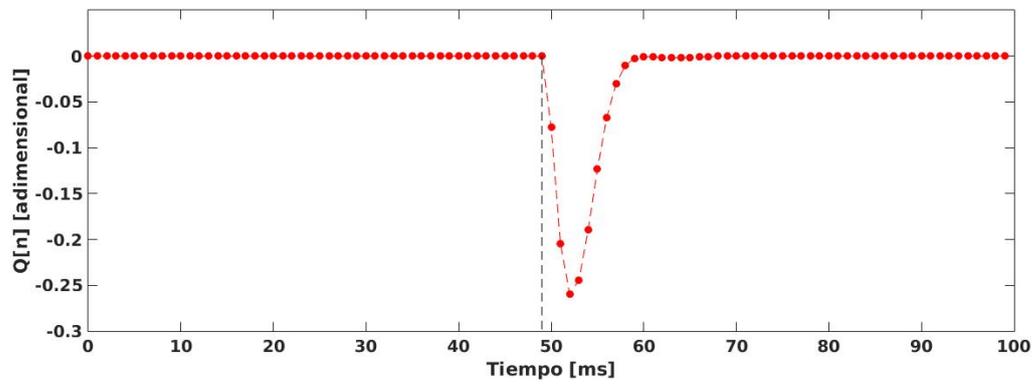
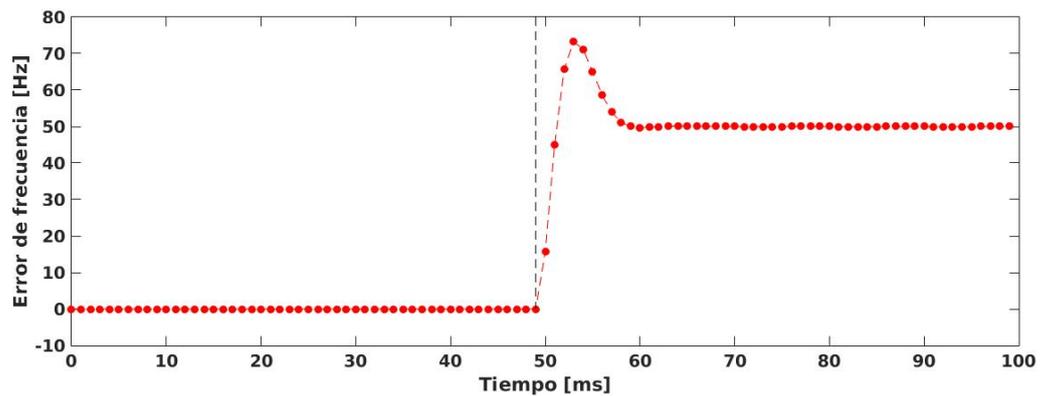


FIGURA 2.22: Correlación del canal en cuadratura (Q)

FIGURA 2.23: Variación de la frecuencia del oscilador local digital (Δf)

Respuesta del sistema frente a escalón de fase

Se realizó una simulación para comprobar la respuesta del lazo frente a una variación instantánea de la fase de la señal de entrada la cual puede observarse en la figura 2.24. El cambio abrupto de fase de 0° a -30° se produce a los 50ms del inicio de la simulación y está indicado con una línea punteada vertical. El ensayo comienza con el lazo «enganchado» y el cambio de fase genera una perturbación instantánea del error de fase $\Delta\phi$ el cual puede observarse en la figura 2.25 que luego es corregido por el sistema de control. En las figuras 2.21 y 2.27 se indica la evolución temporal de $I[n]$ y $Q[n]$. El valor de $I[n]$ es máximo y constante hasta la perturbación, volviendo a su valor inicial luego de la corrección. De forma homóloga, el valor de $Q[n]$ es nulo al comienzo de la simulación (condición de «enganche») y luego de la perturbación vuelve a serlo. La variación de frecuencia del oscilador digital local que introduce el sistema realimentado se indica en la figura 2.28, donde se observa que disminuye la frecuencia del oscilador local para hacer frente al retraso de fase de la señal de entrada. Finalmente, una vez producido el enganche la diferencia de frecuencia del oscilador respecto de la señal de entrada vuelve a 0Hz al producirse el «enganche» de fase.

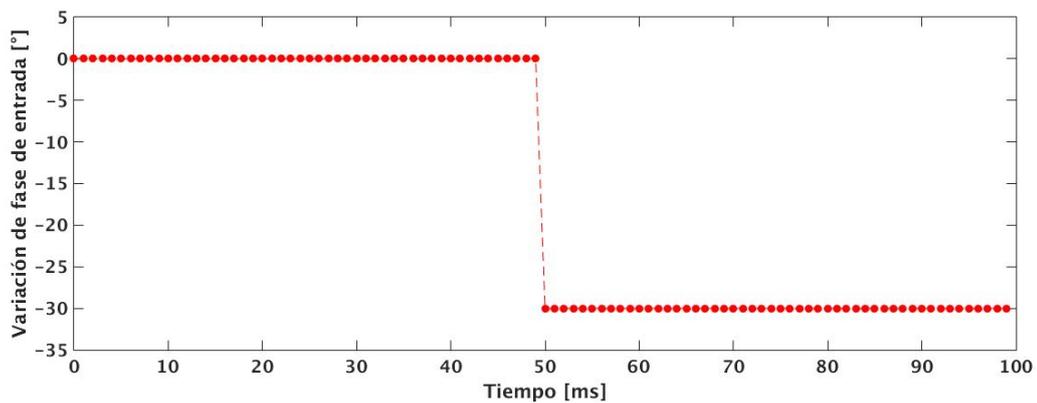


FIGURA 2.24: Cambio de la referencia de fase de la señal de entrada

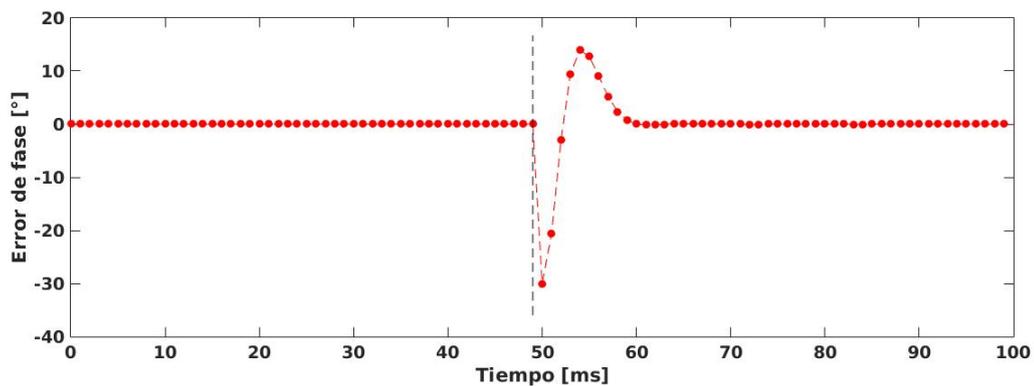


FIGURA 2.25: Error de fase instantáneo ($\Delta\phi$)

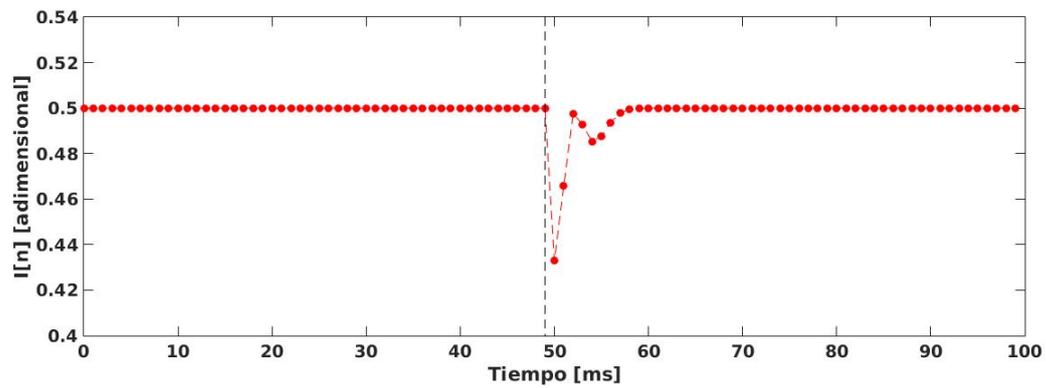


FIGURA 2.26: Correlación del canal en fase (I)

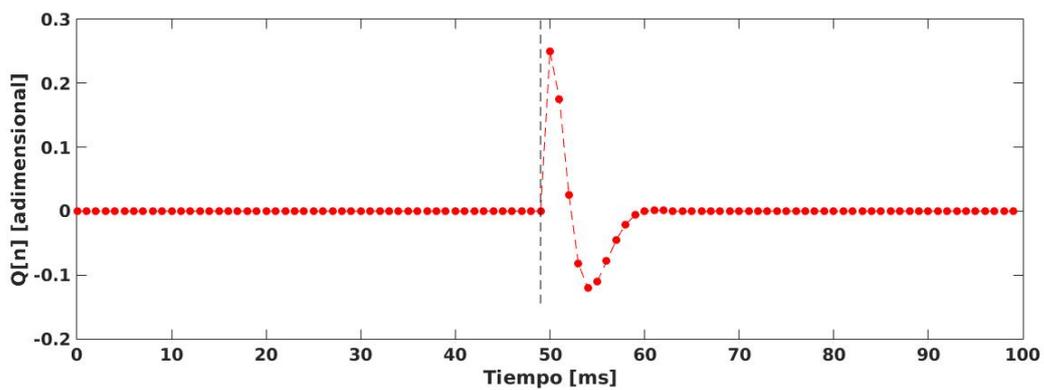
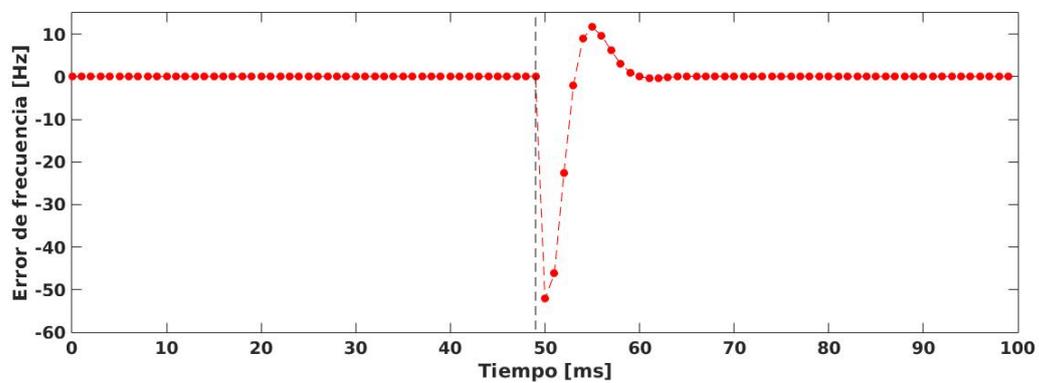


FIGURA 2.27: Correlación del canal en cuadratura (Q)

FIGURA 2.28: Variación de la frecuencia del oscilador local digital (Δf)

2.4. Conclusiones

En este capítulo, se presentaron las ecuaciones y métodos básicos para comprender la generación de las señales GPS y su detección por medio del método de búsqueda serial y el seguimiento de su frecuencia de portadora y fase de código C/A.

Las ecuaciones que plantean matemáticamente el método se desarrollaron detalladamente y se generaron *scripts* para reproducir señales ideales bajo diferentes condiciones (número de satélite, frecuencia central y fase de código C/A). Estos *scripts* permiten simular la operación de búsqueda para una señal o mezcla de señales de entrada arbitrarias. Por consiguiente, son de gran utilidad para las etapas posteriores del trabajo ya que se utilizarán tanto para el procesamiento de señales reales como para contrastar el procesamiento realizado sobre la lógica digital implementada.

Además, fue posible estudiar el comportamiento de una arquitectura de lazo de seguimiento ante diferentes perturbaciones de la señal de entrada permitiendo establecer una base sólida para el diseño de los módulos que se desarrollarán en el capítulo 3.

Capítulo 3

Diseño e Implementación

En este capítulo, se detalla el diseño e implementación de los módulos destinados a procesar la señal de salida de un *frontend* GPS, incluyendo sus interfaces, estructura, operación y señales generadas. Además, se incluye una fundamentación acerca del enfoque general empleado para la implementación de los módulos sobre lógica programable.

3.1. Introducción

3.1.1. Consideraciones generales

La función de un *frontend* consiste en acondicionar la señal L1 emitida por los satélites GPS ubicada en 1545.72MHz. Un esquema general de un *frontend* se indica en la figura 3.1.

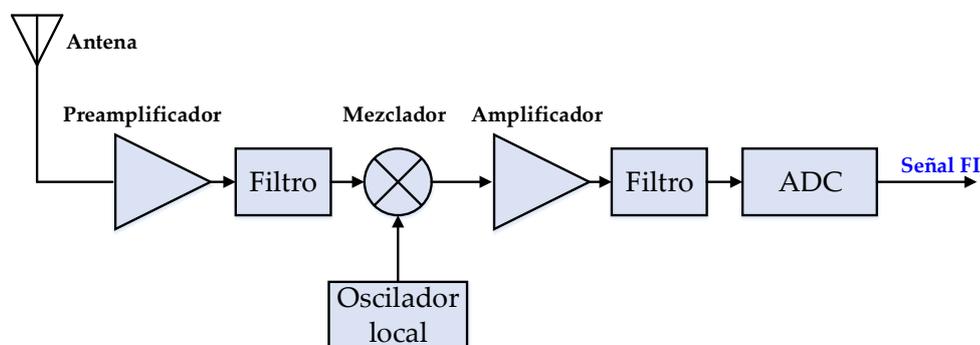


FIGURA 3.1: Diagrama en bloques del *frontend*

En la figura 3.1 se observan las diferentes etapas que componen un *frontend*. La señal de radiofrecuencia se recibe por medio de una antena activa o pasiva, para luego ser preamplificada (en general por un amplificador de bajo ruido). El filtro a continuación permite rechazar la frecuencia imagen previo a la mezcla con el oscilador local. Una vez convertida la señal a frecuencia intermedia (FI), se vuelve a filtrar para rechazar los productos de intermodulación de frecuencias superiores y evitar el *aliasing* previo a la etapa de conversión analógica a digital.

En este trabajo se utilizará consistentemente los parámetros generales indicados en la tabla 3.1 que son usuales en circuitos integrados de radiofrecuencia tales como el MAX 2769[14] de Maxim Integrated Inc. o el SE4150[15] de Skyworks Inc.

La utilización de circuitos integrados monolíticos en un *frontend* facilita su diseño, ya que no es necesario involucrarse con los pormenores de la implementación de los filtros, mezcladores y conversores A/D que se utilizan para operar la señal de radiofrecuencia.

TABLA 3.1: Parámetros generales del proyecto

Parámetro	Valor
Frecuencia intermedia	4,092 MHz
Frecuencia de muestreo	16,368 MHz
Bits de digitalización	1

La señal se digitaliza a un bit de resolución [16]¹ y luego se procesa en una FPGA (*Field Programmable Gate Array*) donde se realizan las operaciones más demandantes en tiempo real: la búsqueda (*search*) y el seguimiento (*tracking*).

La operación de búsqueda, consiste en correlacionar la señal de entrada con una réplica local correspondiente a un determinado satélite para diferentes fases del código C/A y de frecuencia intermedia. Si se encuentra un máximo de correlación consistente con la presencia de un satélite se informan los valores de frecuencia de código C/A y desplazamiento de frecuencia intermedia (debido al efecto Doppler) a la etapa de seguimiento la que se encargará de mantener una réplica local coherente con la señal entrante a fin de obtener el mensaje de navegación del satélite.

3.1.2. Consideraciones de diseño

En el diseño de los módulos se utilizó como referencia una FPGA Xilinx Spartan 3E 500, incluida en un kit de desarrollo Nexys 2 de la compañía Digilent Inc.[17] No obstante, se buscó generar un diseño lo más portable posible evitando la instanciación de módulos definidos por el fabricante (*IP Cores*), salvo para la definición de las memorias internas (*blockRAMs*).

Para los módulos de mayor complejidad se utilizó una implementación del tipo máquina de estados finita con camino de datos (FSMD, del inglés *Finite State Machine with Data path*) [18]. Una máquina de estados finita con camino de datos combina una máquina de estados finita (FSM, del inglés *Finite State Machine*) y circuitos secuenciales convencionales.

En la figura 3.2 se observan las dos partes principales de una FSMD. Por un lado, la máquina de estados (FSM), indicada como el «camino de control» examina los comandos externos (indicados a través de las entradas de control) y según su estado interno genera señales de control que operan sobre los datos de entrada a partir de los circuitos combinatoriales y secuenciales que conforman el «camino de datos». Usualmente, las FSMD implementan sistemas descritos a nivel de lógica de transferencia de registros (RTL, del inglés *Register Transfer Logic*) en donde las operaciones se realizan manipulando y transfiriendo datos entre una colección de registros internos del camino de datos.

¹Es común en los *frontend* de señales GPS la digitalización a 1, 2 o 3 bits. Ver referencia citada.

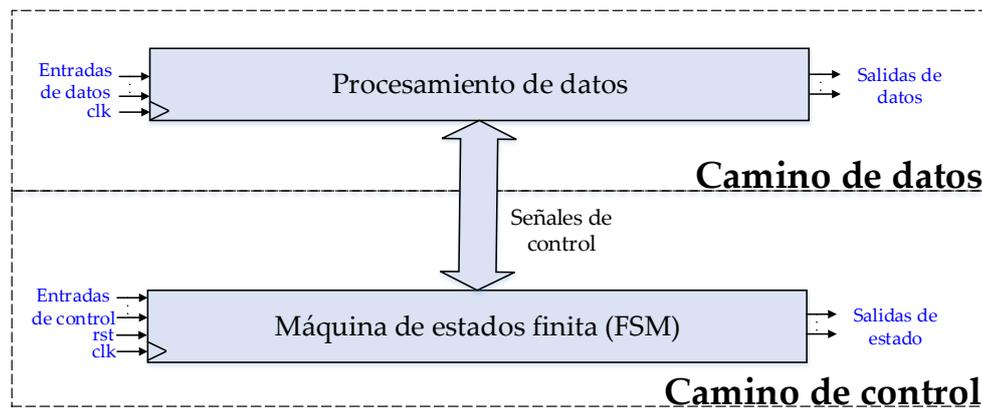


FIGURA 3.2: Diagrama en bloques general de una FSM D

El ciclo de trabajo utilizado para diseñar, implementar y simular las FSM D se describe a continuación:

1. Definir el algoritmo a implementar
2. Definir la interfaz (entradas y salidas) de la FSM D
3. Diseñar el camino de datos
4. Definir de la interfaz entre camino de datos y el camino de control
5. Definir los estados de la FSM D
6. Diseñar la FSM D
7. Codificar el diseño en VHDL
8. Diseñar y realizar las pruebas (*testbenches*)

La metodología descrita, se ha empleado consistentemente para la implementación de los módulos planteados en este capítulo. Este enfoque, si bien demanda un tiempo de desarrollo algo elevado ya que requiere implementar los circuitos en el más bajo nivel posible (bloques combinatoriales, registros, etc.) permite por otro lado tener máximo control sobre el diseño, favorece la portabilidad y reduce a un mínimo los recursos utilizados de la FPGA.

El sistema fue diseñado con un reloj único de 81.84 MHz, el cual es un múltiplo de la frecuencia de muestreo del front end como se indica en la tabla 3.1. Esta elección se realizó por dos razones:

- porque puede generarse con un error menor que 0,03% utilizando los módulos de control digitales de reloj (*DCM, Digital Clock Manager*[19]) de la FPGA a partir del reloj de entrada de la placa de desarrollo de 50 MHz (eligiendo para ello como multiplicador $M = 18$ y como divisor $N = 11$). De esta manera se facilitan las pruebas entre un *front end* desarrollado separadamente y el kit de desarrollo. Es importante aclarar además, que esta elección simplifica el diseño a futuro de un receptor integrado, ya que se podría utilizar una única fuente de reloj tanto para la FPGA como para el *frontend*.

- porque dicha frecuencia de reloj es un múltiplo de la tasa de bits del código C/A, permitiendo utilizar un mismo clock para todo el sistema, simplificando la generación de las señales internas, adaptando las frecuencias de algunos módulos por medio de circuitos de habilitación de reloj *preescalers*.

En las secciones subsiguientes se detallan las interfaces de los módulos construídos, su diagrama en bloques, la operación de los mismos y ensayos unitarios realizados.

3.2. Front end

Se diseñó y construyó una placa de circuito impreso la cual implementa un *frontend* GPS en base al circuito integrado Skyworks SE4150L [15]. Este circuito presenta como ventaja que es de fácil utilización, no requiere de una interfaz de programación y es relativamente económico respecto de los integrados en su segmento. En la figura 3.3 se indica el diagrama esquemático del circuito. En la figura 3.4 se muestra el PCB diseñado utilizando el programa KiCad[20].

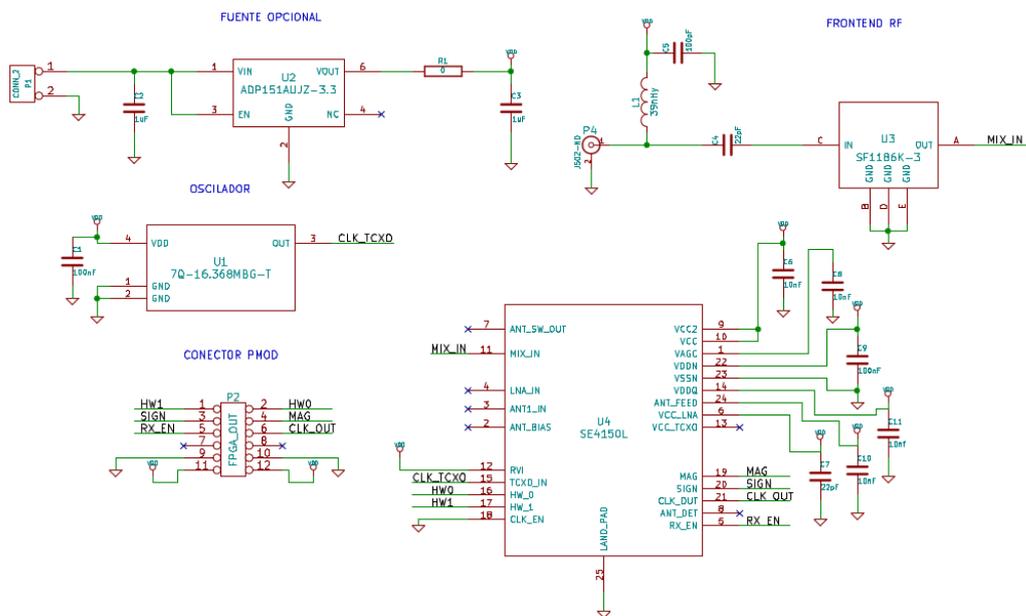
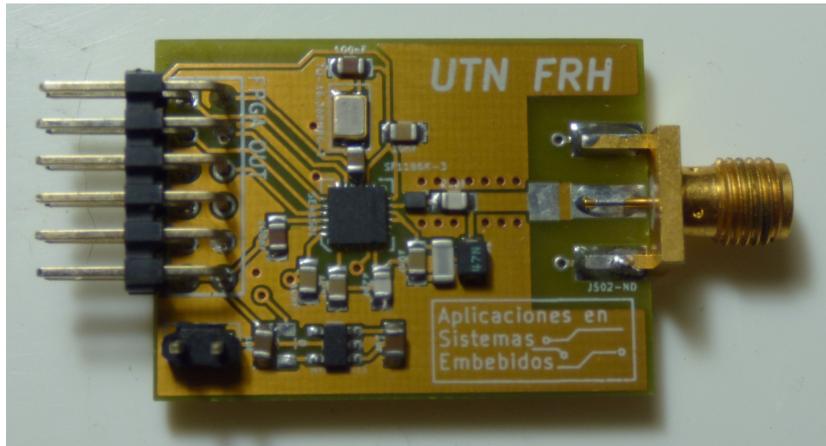
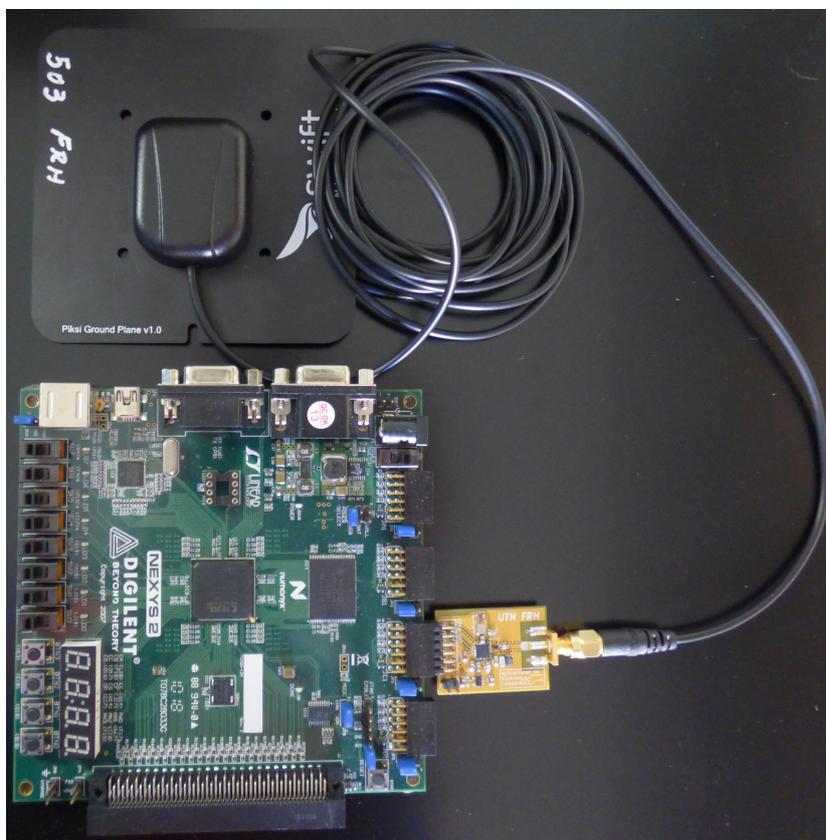


FIGURA 3.3: Diagrama esquemático del circuito del *frontend*

La placa del *frontend* fue diseñada específicamente para conectarse a un puerto PMOD [21] de las placas de desarrollo de FPGAs Xilinx de la empresa Digilent Inc. En la figura 3.5 se observa como el *frontend* con su antena activa se conecta directamente a la placa de desarrollo para procesar las muestras digitalizadas.

En la sección (3.3.1) se desarrolla el diseño de un módulo digital en la FPGA para adquirir una serie de muestras digitales del *frontend*. Dichas muestras serán analizadas posteriormente para caracterizar el funcionamiento así como validar los algoritmos de búsqueda y seguimiento que finalmente serán implementados sobre la FPGA.

FIGURA 3.4: Placa de circuito impreso (PCB) del *frontend*FIGURA 3.5: Fotografía del *frontend* con su antena activa conectado a la placa de desarrollo FPGA

3.3. Módulo de adquisición de señales digitales

El módulo de adquisición de señales digitales se utiliza para adquirir una serie temporal de muestras digitales provenientes del *frontend*, almacenándolas en una memoria RAM de la placa de desarrollo de la FPGA.

3.3.1. Interfaz

En la figura 3.6 se presenta la interfaz del módulo de adquisición de señales digitales y en la tabla 3.2 se detallan sus entradas y salidas.

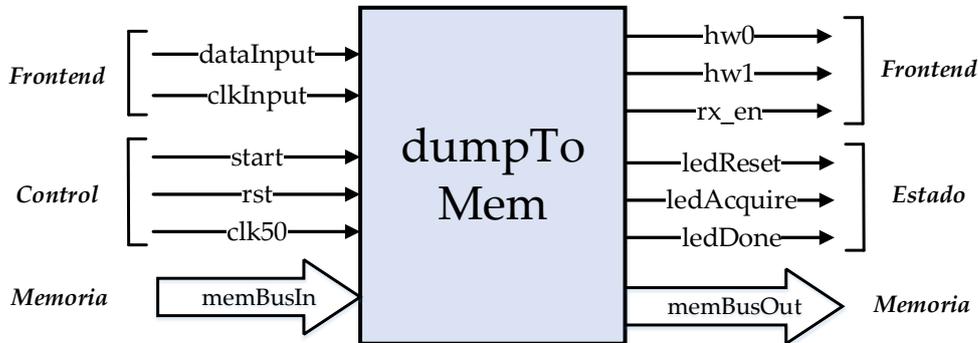


FIGURA 3.6: Interfaz del módulo de adquisición de señales digitales

TABLA 3.2: Descripción de la interfaz del Módulo de adquisición de señales digitales

Señal	Modo	Tipo	Descripción
dataInput	IN	std_logic	Entrada de datos <i>frontend</i>
clkInput	IN	std_logic	Reloj del <i>frontend frontend</i>
start	IN	std_logic	Inicio adquisición
rst	IN	std_logic	Reset
clk50	IN	std_logic	Reloj 50 MHz
memBusIn	IN	-	Bus de entrada memoria SRAM
hw0	OUT	std_logic	Configuración <i>frontend</i> (0)
hw1	OUT	std_logic	Configuración <i>frontend</i> (1)
rx_en	OUT	std_logic	Habilitación <i>frontend</i>
ledReset	OUT	std_logic	Led indicador (estado inicial)
ledAcquire	OUT	std_logic	Led indicador (adquisición)
ledDone	OUT	std_logic	Led indicador (fin adquisición)
memBusOut	IN	-	Bus de salida memoria SRAM

3.3.2. Operación

El *frontend* se conecta al módulo de adquisición a través de las señales de entrada de datos *dataInput* y su reloj *clkInput* el cual establece cuando hay disponible una nueva muestra. Ambas señales constituyen una interfaz sincrónica como se describe en la hoja de datos del circuito integrado Skyworks SE4150L[15]. Además, provee tres señales de control:

- *rx_en*: se utiliza para habilitar la operación del *frontend*

- $hw0, hw1$: se utilizan para configurar la ganancia del mezclador y del amplificador de bajo ruido.

En la figura 3.7 se observa un diagrama en bloques del módulo de adquisición de señales digitales. Las señales $dataInput$ y $dataReady$ se inyectan en el módulo de entrada (ver sección 3.7) que adapta los dominios de reloj del *frontend* al del módulo de adquisición, generando una salida de datos $dataOutput$ y una de habilitación $dataReady$. Como se ha descrito en la sección 3.1.1, la señal es de un bit, así que para escribir las muestras en la memoria se empaquetan en palabras de 16 bits a través de la lógica de empaquetamiento. Finalmente, el controlador de memoria se encarga de generar las señales de direcciones, datos y control de forma que las palabras entregadas por la lógica de empaquetamiento se almacenen en memoria.

La memoria que se utiliza, Micron Semi MT45W1MW16BDGB[22], viene integrada en el kit Nexys 2. Es una memoria RAM de 128 Mbit que puede utilizarse de forma asíncrona con características externas de funcionamiento similares a una SRAM. Se usó esta memoria ya que es la de mayor capacidad del kit, maximizando la longitud de secuencias de entrada que pueden almacenarse.

La máquina de estados «FSMAdquisición» inicia la grabación al detectar un uno en la entrada $start$ y va indicando a través de las salidas $ledReset$, $ledAcquire$ y $ledDone$ los diferentes estados del proceso de adquisición.

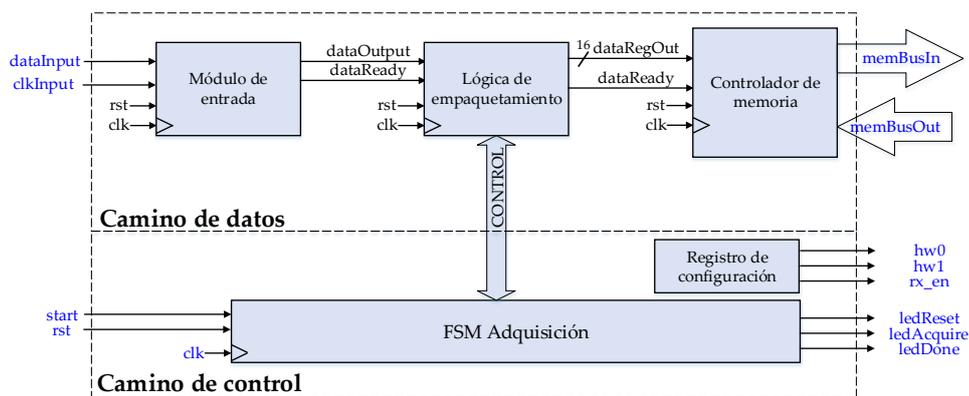


FIGURA 3.7: Diagrama en bloques del módulo de adquisición de señales digitales

3.3.3. Pruebas

Se generó un banco de pruebas (*testbench*) con el objeto de comprobar la correcta generación de las señales de salida y el empaquetamiento de muestras. En la figura 3.8, se muestra la prueba realizada para una señal de datos cuadrada, en consecuencia, los datos serán ceros y unos alternados. El módulo se encuentra en estado (*fsmstate*) 'reset'. Cuando el bit de $start$ pasa a uno, la máquina pasa a estado 'acquire' y comienza el empaquetamiento. Cuando se tiene una palabra de 16 bits completa, se genera la dirección de memoria y se coloca dicha palabra en el bus de datos.

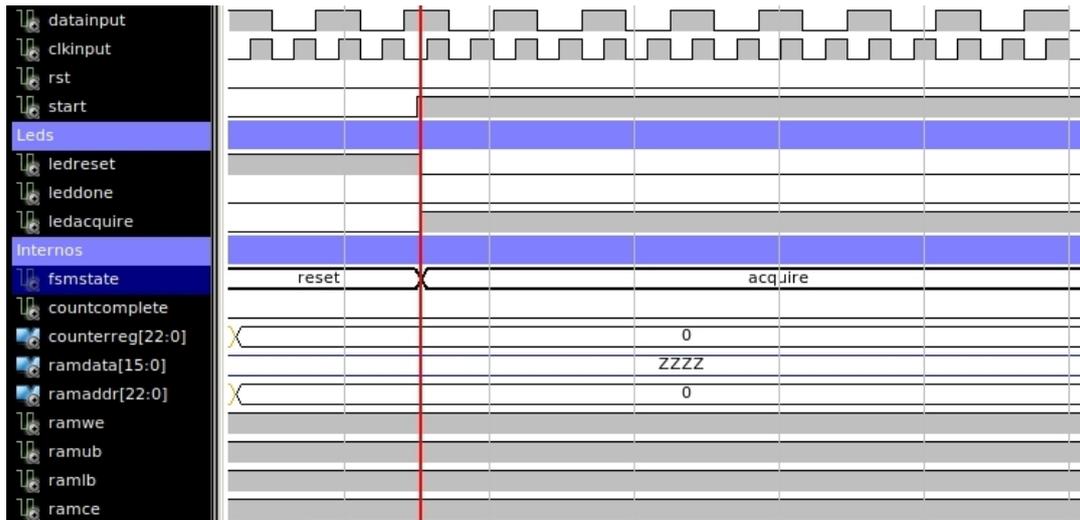


FIGURA 3.8: Señales correspondientes al inicio de una adquisición

En la figura 3.9 se observa que el bus de datos *ramdata* se carga con la palabra $AAAA_{16}$, donde el sufijo $_{16}$ indica 'base 16', correspondiente a una secuencia de ceros y unos. El bus de direcciones *ramaddr* toma el valor de la dirección correspondiente (en este caso 112_{10}). Las señales de control (*ramwe*, *ramub*, *ramlb* y *ramce*) se ponen en cero para efectuar la escritura.

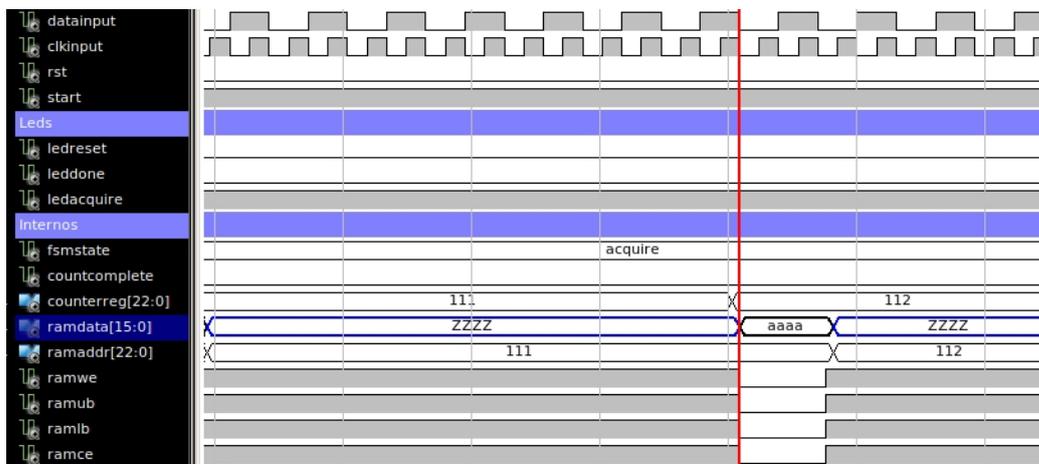


FIGURA 3.9: Señales correspondientes al proceso de adquisición

El proceso continúa de la misma manera hasta que un contador interno alcanza un determinado valor límite, el cual se declara definiendo una constante de programa y en la figura 3.10 se pone de manifiesto esta situación. Al alcanzar el contador *countReg* el valor final, en este caso igual a 10230_{10} , se detiene la adquisición. Los diferentes estados del proceso se indican por medio de las salidas *ledReset*, *ledAcquire* y *ledDone* que son mapeadas a los leds de la placa de desarrollo para dar indicaciones visuales al usuario de los diferentes estadios del proceso.

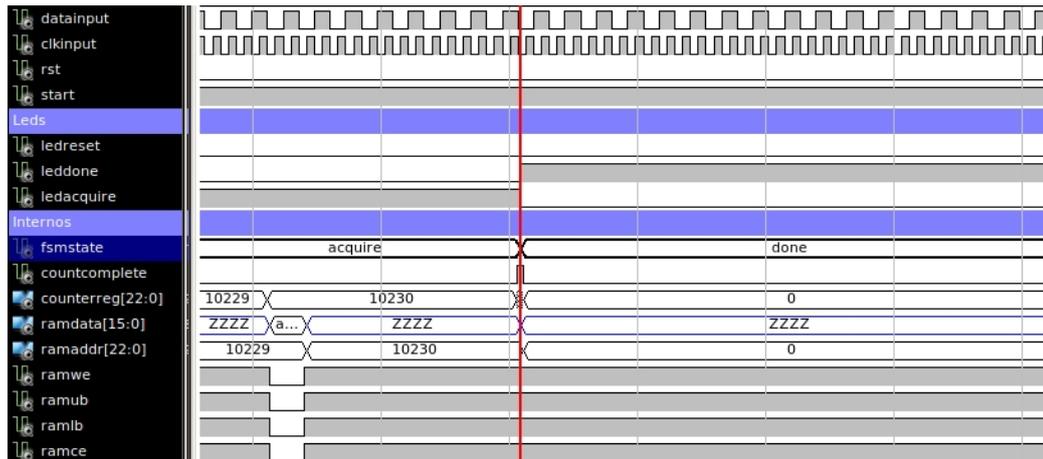


FIGURA 3.10: Señales correspondientes al fin del proceso de adquisición

3.4. Módulo de reproducción de señales digitales

Este módulo se utiliza para reproducir una serie temporal de muestras digitales cuyo formato es idéntico al generado por el módulo de entrada (ver sección 3.7). Su objetivo es utilizarlo como fuente de generación de datos para probar los otros módulos objeto de este trabajo (el módulo de búsqueda y el módulo de seguimiento) de forma flexible y conveniente.

3.4.1. Interfaz

En la figura 3.11 se indica la interfaz del módulo de reproducción de señales digitales, estando su descripción detallada en la tabla 3.3.

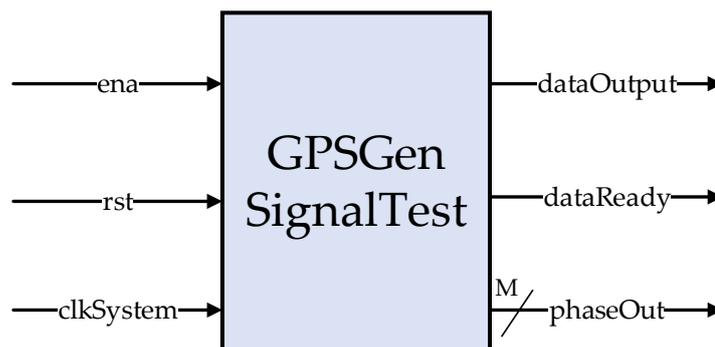


FIGURA 3.11: Interfaz del módulo de reproducción de señales digitales

En todas las figuras del capítulo donde se ilustren las interfaces y bloques constitutivos de cada módulo se representan las señales de más de un bit (*buses*) con la notación que se observa en la figura 3.11, donde M representa la cantidad de bits de la señal, expresada de forma genérica. Esta notación se utilizará consistentemente en todo el trabajo.

TABLA 3.3: Descripción de la interfaz del Módulo de reproducción de señales digitales

Señal	Modo	Tipo	Descripción
ena	IN	std_logic	Habilitación
rst	IN	std_logic	Reset
clkSystem	IN	std_logic	Reloj
dataOutput	OUT	std_logic	Salida de datos
dataReady	OUT	std_logic	Indicación dato válido
phaseOut	OUT	std_logic_vector(M)	Número de muestra

3.4.2. Operación

En la figura 3.12 se muestra un diagrama en bloques del módulo. La memoria se puede cargar con una serie temporal de muestras sintéticas o reales, las cuales se reproducen siguiendo el mismo esquema temporal que aquellas generadas por el *frontend*. Para definir los datos a reproducir se utiliza un archivo de datos en formato «.coe»[23].

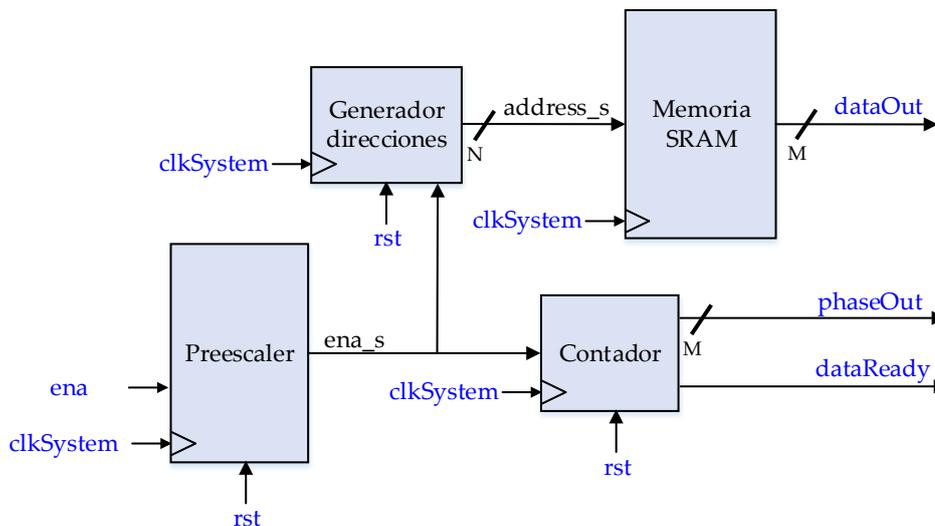


FIGURA 3.12: Diagrama en bloques del módulo de reproducción de señales digitales

3.4.3. Pruebas

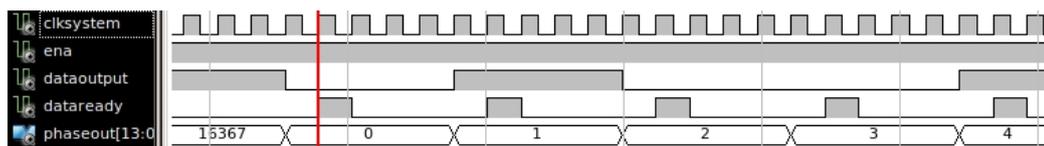
Se generó un *testbench* con el objeto de comprobar la correcta reproducción de las señales almacenadas en la memoria. En la figura 3.13, se presenta una imagen del archivo de datos COE que contiene el valor de las muestras a reproducir.

Al iniciar una adquisición, *ena* se pone en uno, comienza la reproducción de los datos cargados en el archivo a través de la salida *dataOutput*. La salida *dataReady*

Index	Value
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	0
8	0
9	0
10	1
11	1

FIGURA 3.13: Archivo de datos de la memoria (COE)

se pone en uno para indicar que el dato es válido y puede procesarse por los bloques que se conectan a continuación. La salida *phaseOut* indica el número de muestra de la secuencia. En la figura 3.14 se observa el comportamiento descrito.

FIGURA 3.14: Señales del módulo de reproducción (captura del *testbench*)

Se instanció el módulo descrito en esta sección en el kit de desarrollo Nexys 2 a fin de validar la correcta generación de las señales. Para ello, se midió por medio de un analizador lógico Zeroplus LAP-C[24] las señales de salida. En la figura 3.15 se muestra una captura de la interfaz gráfica del instrumento. Se observan las salidas *dataOutput*, *dataValid* y los cuatro bits menos significativos de *phaseOut*. Para facilitar el sincronismo se agregó una salida adicional *seqStart* (no forma parte del módulo).

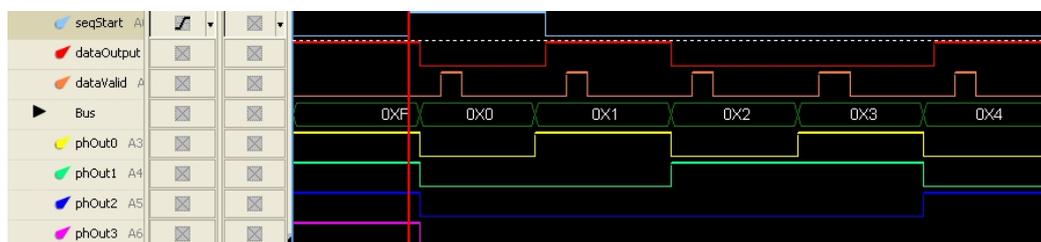


FIGURA 3.15: Señales del módulo de reproducción (captura del analizador lógico)

3.5. Generador de código C/A

Este módulo se utiliza para generar la secuencia de código C/A de un satélite determinado.

3.5.1. Interfaz

En la figura 3.16 se indica la interfaz del generador de código C/A, estando su descripción detallada en la tabla 3.4.

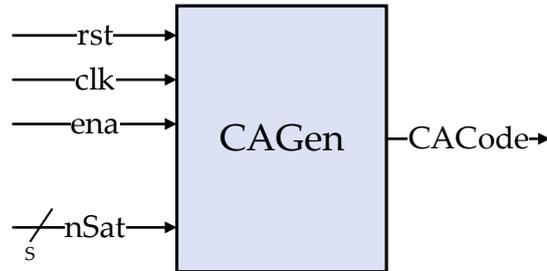


FIGURA 3.16: Interfaz del módulo de generación de código C/A

TABLA 3.4: Descripción de la interfaz del generador de código C/A

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
ena	IN	std_logic	Habilitación
nSat	IN	std_logic_vector (S)	Selección de satélite
CACode	OUT	std_logic	Código C/A del satélite seleccionado

3.5.2. Operación

Este módulo genera los códigos C/A del satélite que se indica en su entrada *nSat*, interpretada como un valor no signado de 0 a 31, correspondiendo unívocamente a los códigos de los satélites 1 a 32. La entrada de habilitación (*ena*) se conecta a un módulo externo (*preescaler*) que habilita selectivamente al módulo a una tasa tal que permite generar el código a la frecuencia adecuada (1,023MHz) siendo esta un submúltiplo del reloj del sistema.

En la figura 3.17 se muestra un diagrama en bloques del generador que responde a la implementación indicada en el capítulo anterior. El módulo consta de dos registros de desplazamiento realimentados (G1 y G2) y un bloque combinacional que permite seleccionar a partir de la entrada *nSat* el código C/A del satélite elegido.

3.5.3. Pruebas

Se realizó un *testbench* con el objeto de obtener las señales de salida. Las mismas fueron almacenadas en un archivo de texto y comparadas con señales de código

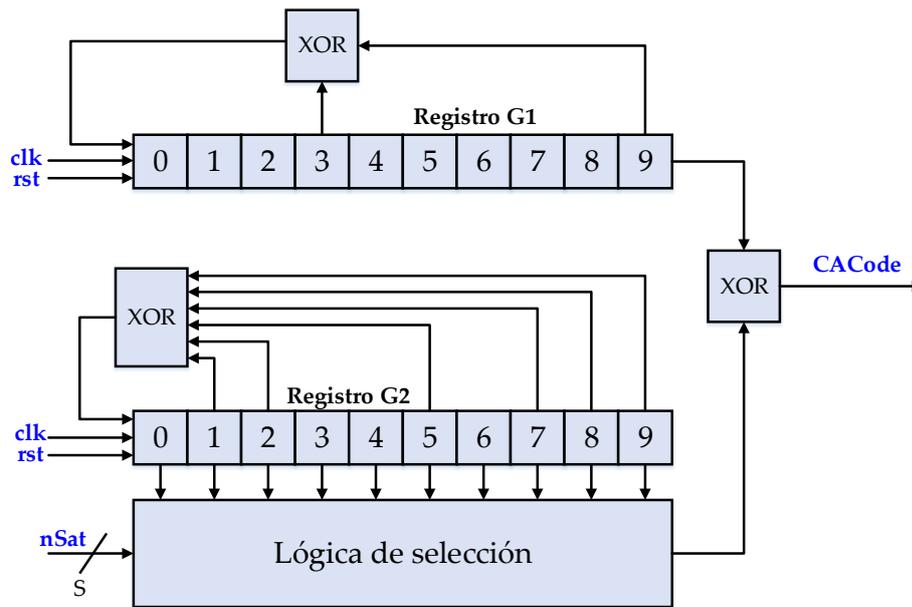
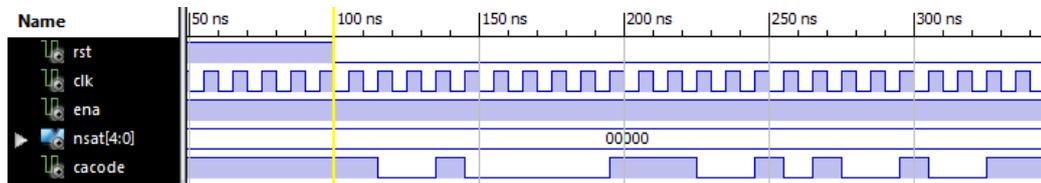


FIGURA 3.17: Diagrama en bloques del generador de código CA

C/A generadas separadamente por medio de un *script* de Octave verificándose su perfecta coincidencia. En la figura 3.18 se muestra a modo de ejemplo la generación de una señal de código C/A para el satélite 1 (salida *CACode*).

FIGURA 3.18: Generación de código C/A (salida *caCode* para el satélite 1)

3.6. Oscilador controlado numéricamente (NCO)

El oscilador controlado numéricamente (NCO, por sus siglas en inglés *Numerical Controlled Oscillator*) permite generar las réplicas locales de portadora en frecuencia intermedia tanto para el módulo de búsqueda como para el de seguimiento.

3.6.1. Interfaz

En la figura 3.19 se indica la interfaz del NCO, estando su descripción detallada en la tabla 3.5.

3.6.2. Operación

La frecuencia del NCO está dada por [25]:

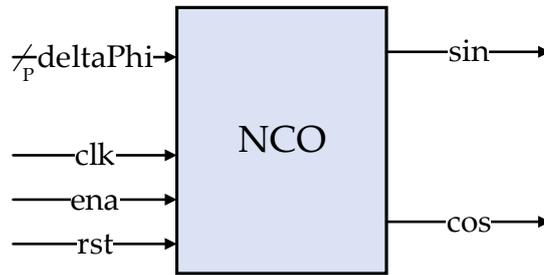


FIGURA 3.19: Interfaz del NCO

TABLA 3.5: Descripción de la interfaz del generador de código C/A

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
ena	IN	std_logic	Habilitación
deltaPhi	IN	std_logic_vector (P)	Incremento de fase
sin	OUT	std_logic	Salida seno
cos	OUT	std_logic	Salida coseno

$$f_o = \frac{\Delta\phi}{2^M} f_{clk} \quad (3.1)$$

donde:

- f_o : Frecuencia de salida del NCO
- $\Delta\phi$: Incremento de fase indicado a la entrada del NCO
- M : Cantidad de bits del registro del acumulador de fase
- f_{clk} : Reloj aplicado al NCO

En la figura 3.20 se muestra un diagrama en bloques del NCO. El bloque de truncamiento reduce la resolución en bits del registro acumulador previo a seleccionar la salida por medio de la tabla. En este proyecto, se truncó el registro a los dos bits más significativos ($L = 2$ en el esquema) utilizando un circuito combinacional como tabla de salida ya que las salidas son de un bit.

3.6.3. Pruebas

Se comprobó el correcto funcionamiento del NCO instanciándolo en la FPGA midiendo la frecuencia de salida por medio de un osciloscopio digital Hantek MSO5202D[26]. En la figura 3.21 se observa un oscilograma de las señales seno y coseno medidas.

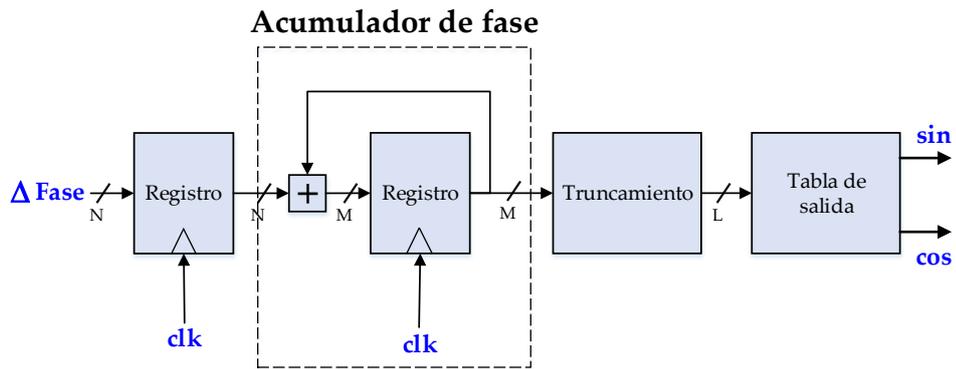


FIGURA 3.20: Diagrama en bloques del NCO

A fin de comprobar la correcta variación de la frecuencia de salida en función de la señal de control, se agregó a la entrada del NCO un bloque combinacional que en el kit de desarrollo se controla por medio de pulsadores y teclas. Este bloque introduce un número signado de 9 bits (-256 a 255) que permite variar la frecuencia de salida en torno al valor de referencia (la frecuencia de FI 4,092MHz) en pasos de 20Hz aproximadamente. Se relevó la frecuencia según la palabra de control a la entrada y se realizó el gráfico que se muestra en la figura 3.22, comprobándose la linealidad entre el incremento de fase $\Delta\phi$ y la frecuencia de salida f_o .

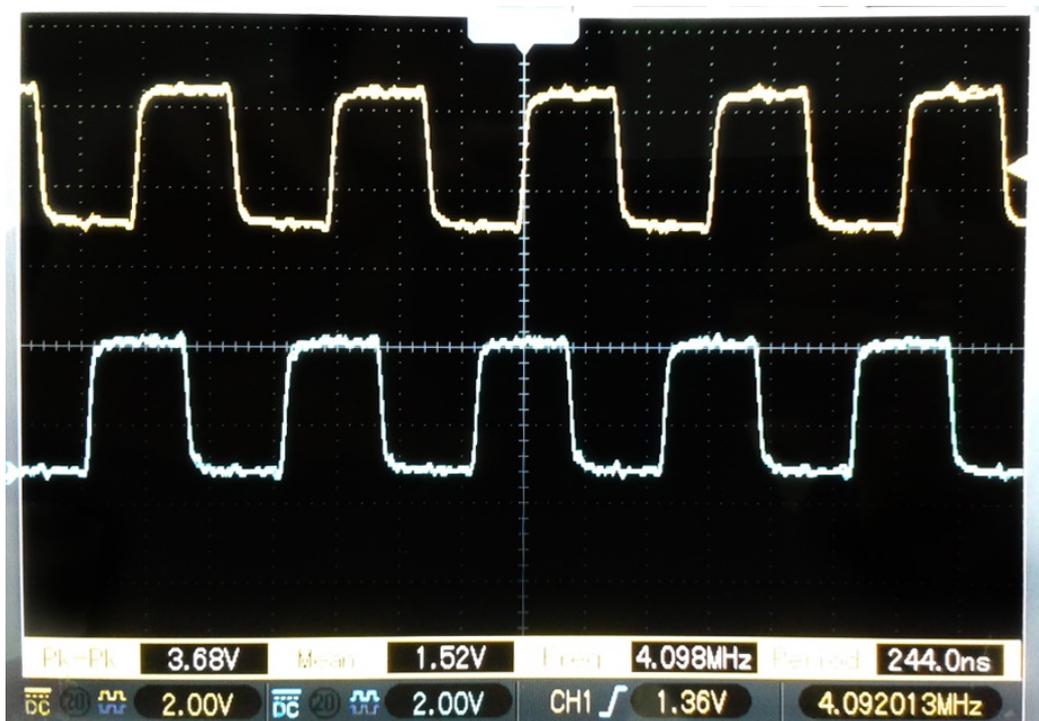


FIGURA 3.21: Oscilograma de las señales de salida del NCO: seno (superior) y coseno (inferior)

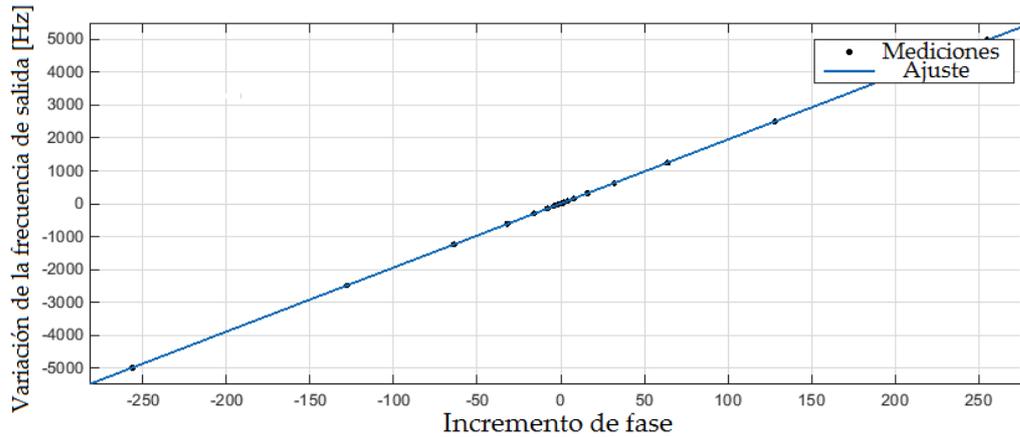


FIGURA 3.22: Valores medidos de frecuencia de salida en función del incremento de fase

3.7. Módulo de entrada

El módulo de entrada tiene por objeto adaptar las señales de datos provenientes del *frontend* para que puedan ser utilizadas por los diferentes módulos de procesamiento dentro de la FPGA. Este módulo actúa como interfaz de entrada, adaptando los dominios de reloj del *frontend* y los de la FPGA.

3.7.1. Interfaz

En la figura 3.23 se indica la interfaz del módulo de entrada, estando su descripción detallada en la tabla 3.6.

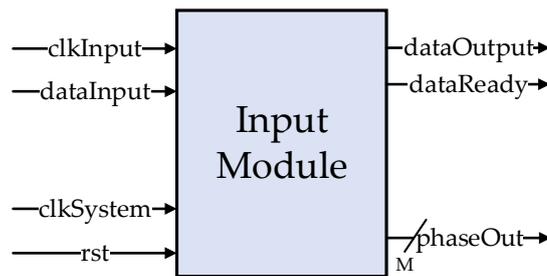


FIGURA 3.23: Interfaz del módulo de entrada

3.7.2. Operación

En la figura 3.24 se puede observar un diagrama en bloques del módulo de entrada. El primer flip flop se encuentra conectado al reloj del *frontend* denominado *clkInput*, mientras que el segundo en la cadena se conecta al reloj del sistema denominado *clkSystem*. Esta configuración tiene por objetivo sincronizar la señal de

TABLA 3.6: Descripción de la interfaz del módulo de memoria

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clkInput	IN	std_logic	Reloj <i>frontend</i>
dataInput	IN	std_logic	Entrada de datos
clkSystem	IN	std_logic	Reloj del sistema
dataOutput	OUT	std_logic	Salida de datos
dataReady	OUT	std_logic	Indicador datos válidos de salida
phaseOut	OUT	std_logic_vector (M)	Número de muestra de la señal de salida

entrada *dataInput* (en el dominio de *clkInput*) al dominio del *clkSystem*. El módulo de *prescaler* genera una señal de habilitación *ena* a partir del reloj del sistema de frecuencia $f_{clk} = 81,84MHz$ para comandar el contador y el flip flop de salida a una tasa de $f_s = 16,368MHz$. El contador numera las muestras obtenidas de 0 a 16367 de forma periódica, lo que corresponde a un período de código C/A. Como se verá, es importante conocer la fase de la señal de entrada en los módulos que se detallan en el resto del capítulo.

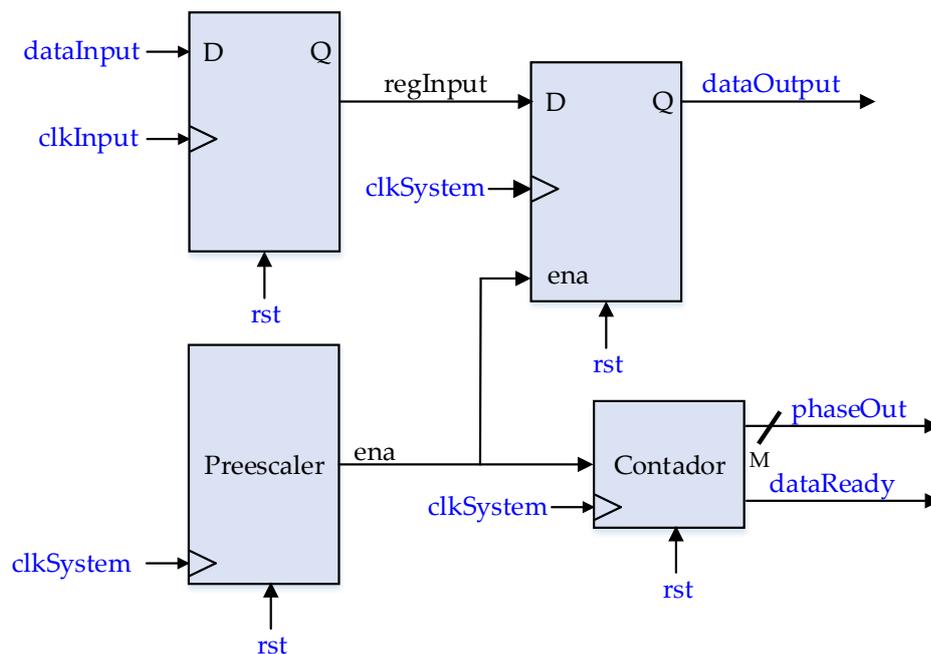


FIGURA 3.24: Diagrama en bloques del módulo de entrada

3.7.3. Pruebas

Con el objeto de comprobar el correcto funcionamiento del módulo de entrada se desarrolló un *testbench* en el que se alimentó al módulo de entrada con una señal de datos cuadrada. Los relojes *clkInput* y *clkSystem* se generaron desfasados para realizar el ensayo con la mayor generalidad posible.

En la figura 3.25 se muestra un diagrama de señales generado por el *testbench*. En él se puede observar que la entrada *dataInput* se almacena en la salida *inputReg* al presentarse el flanco ascendente del reloj de entrada *clkInput*.

Al ocurrir el flanco ascendente del reloj del sistema *clkSystem*, el dato pasa a la salida *dataOutput* y se actualizan las señales *dataReady* y *phaseOut*. La señal *dataReady* se pone en alto un ciclo de reloj indicando que hay un nuevo dato válido y *phaseOut* incrementa su valor.

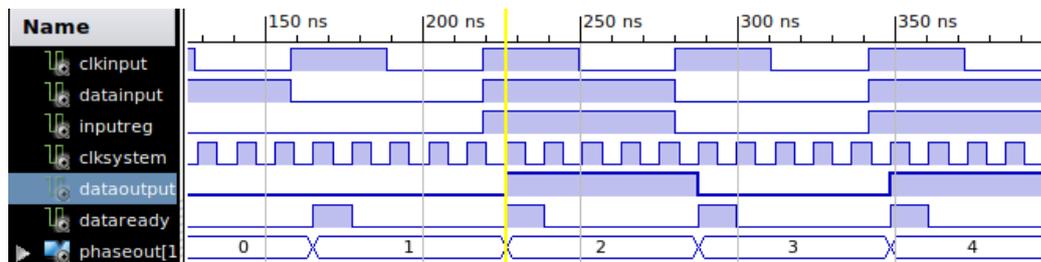


FIGURA 3.25: Diagrama de señales del módulo de entrada

3.8. Módulo de memoria

El módulo de memoria tiene por objeto grabar las muestras de las señales de entrada a la frecuencia de muestreo del *frontend* ($f_s = 16,368\text{MHz}$) en dos bancos de memoria alternativos para, a petición, reproducirlas a la frecuencia de reloj del sistema ($f_{clk} = 81,84\text{MHz}$). En la figura 3.26 se indica la interfaz del módulo de memoria, estando su descripción detallada en la tabla 3.7.

3.8.1. Interfaz

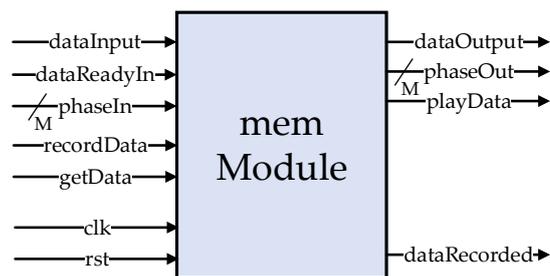


FIGURA 3.26: Interfaz del módulo de memoria

3.8.2. Operación

El módulo de memoria permite:

TABLA 3.7: Descripción de la interfaz del módulo de memoria

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
dataInput	IN	std_logic	Entrada de datos
dataReadyIn	IN	std_logic	Indicador de dato de entrada válido
recordData	IN	std_logic	Comenzar registro de datos
phaseIn	IN	std_logic_vector (M)	Número de muestra de la señal de entrada
getData	IN	std_logic	Reproducir datos
dataOutput	OUT	std_logic	Salida de datos
phaseOut	OUT	std_logic_vector (M)	Número de muestra de la señal de salida
playData	OUT	std_logic	Indicador de datos de salida válidos
dataRecorded	OUT	std_logic	Indicador de datos disponibles

- Desacoplar el flujo de datos de entrada del procesamiento posterior (ya que la fase de los datos de entrada es importante para el procesamiento, se obtiene mayor flexibilidad).
- Procesar los datos de entrada a la frecuencia del reloj del sistema ($f_{clk} = 5f_s$), aumentando la velocidad de procesamiento
- Permitir el escalamiento del diseño incluso a frecuencias superiores a $f_{clk} = 81,84MHz$ (siempre que la frecuencia del sistema sea un múltiplo entero de la frecuencia de muestreo f_s).

En la figura 3.27 se muestra un diagrama en bloques del módulo de memoria, donde se puede observar que el módulo está implementado como una FSMD según lo descrito en la sección 3.1.2.

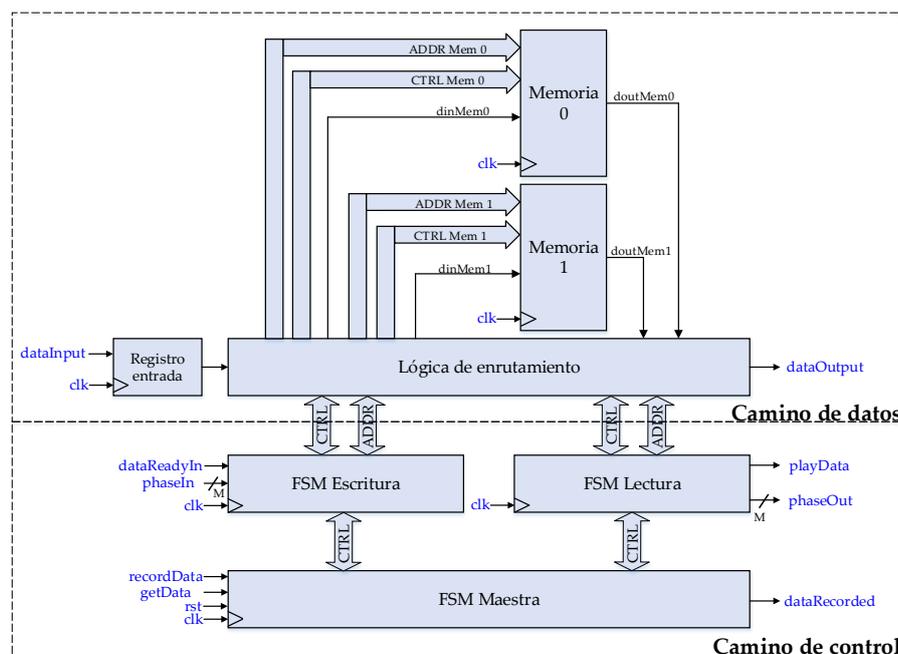


FIGURA 3.27: Diagrama en bloques del módulo de memoria

El camino de datos está compuesto por un registro de entrada el cual permite registrar una muestra de la señal de entrada (*dataInput*), dos bancos de memoria de 16Kbit cada una (denominados «memoria 0» y «memoria 1» e implementados utilizando una *blockRAM* de la FPGA) y lógica adicional. Las memorias 0 y 1 se alternan a fin de registrar la señal de entrada de forma tal que una vez en régimen siempre haya una de ellas disponible para la lectura permitiendo así un procesamiento continuo de la señal a la salida. El bloque denominado «lógica de enrutamiento» se controla por medio del camino de control y su función es la de guiar los datos según el estado interno de la máquina. El camino de control está compuesto por tres máquinas de estado (FSM) que interactúan entre sí: «FSM Maestra», «FSM Escritura» y «FSM Lectura».

La «FSM Maestra» regula el funcionamiento general del módulo. En el estado inicial (luego del *reset*), la «FSM maestra» aguarda la orden de inicio de grabación (entrada *recordData* = 1), de forma tal que los datos de entrada se graben de forma alineada (es decir, aguardando la condición *PhaseIn* = 0) a través de la «FSM Escritura». Cuando el primer banco de memoria («memoria 0») está completo (se han grabado 16.368 muestras equivalentes a un período del código C/A), conmuta la grabación al segundo banco («memoria 1»). En ausencia de lectura, la «FSM Maestra» graba alternativamente ambos bancos de forma alineada. Es importante aclarar, que la «FSM Escritura» se diseñó para abstraerse de que memoria está grabando ya que esto es regulado por la «FSM Maestra», quien controla la lógica de enrutamiento del camino de datos. Esto facilitaría el rediseño en caso de que se requiera aumentar la complejidad del módulo utilizando más de dos memorias.

Si la «FSM Maestra» recibe la orden de reproducir los datos guardados (*getData* = 1), reproducirá a través de la salida de datos *dataOutput* el contenido de la memoria activa para lectura, la que no se está escribiendo, a la frecuencia de reloj del sistema. A fin de ser utilizado por los módulos «clientes», la salida *playData* se pondrá en uno, un ciclo de reloj antes de comenzar la reproducción.

3.8.3. Pruebas

Se realizaron pruebas (*testbenches*) a fin de comprobar el funcionamiento del módulo de memoria en diferentes condiciones. De esta manera, se busca estudiar el comportamiento del módulo frente a diferentes condiciones de trabajo: escritura continua sin lectura de datos, lectura de un único juego de datos y lectura/escritura continua. Las mismas se describen a continuación.

Escritura continua

En esta prueba se le indicó al módulo que registre la señal de entrada, en este caso una señal cuadrada, por medio de la entrada de control (*recordData* = 1). Según se observa en la figura 3.28, una vez detectada la orden *recordData* = 1, la «FSM Escritura» aguarda la fase cero (*phaseIn* = 0) para alinearse, permaneciendo en el estado *detect_frame*.

Una vez detectada la fase cero, la «FSM Escritura» comienza a grabar las muestras de la señal de entrada en la memoria, activando la señal de control de escritura *weamemw* y actualizando las direcciones de memoria *addrmemw* (ver figura 3.29).

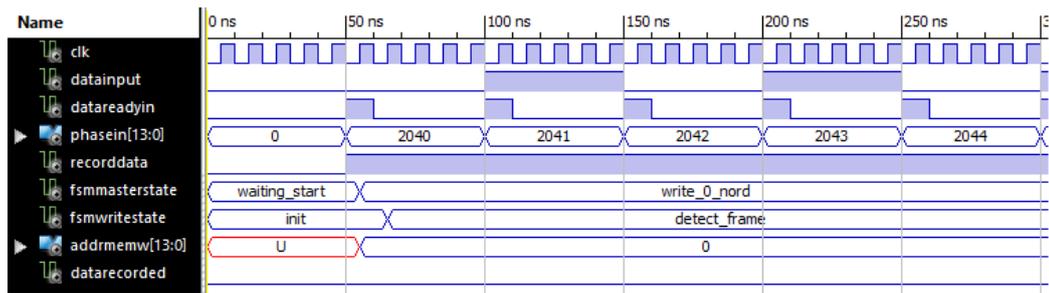


FIGURA 3.28: Comienzo de la prueba de escritura continua

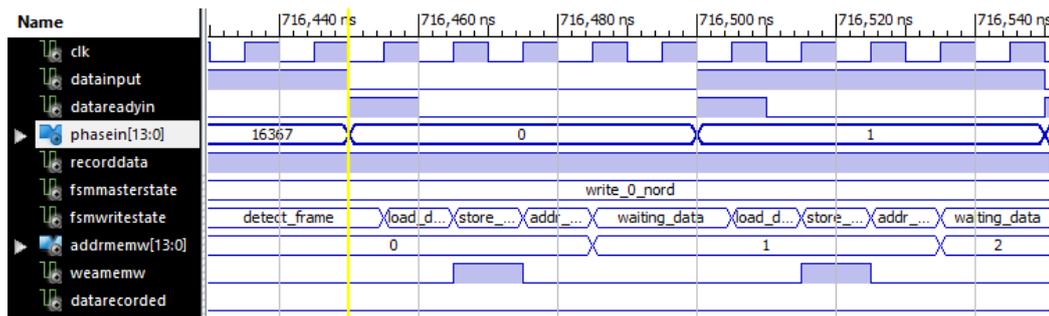


FIGURA 3.29: Comienzo del ciclo de grabación

Al finalizar la grabación de 16.368 muestras, se completa el ciclo por lo que se conmuta de la grabación de la memoria 0 a la memoria 1, pasando el estado de la «FSM Maestra» de *write_0_no_rd* a *write_1_no_rd* y poniendo en alto la salida *dataRecorded* (ver figura 3.30).

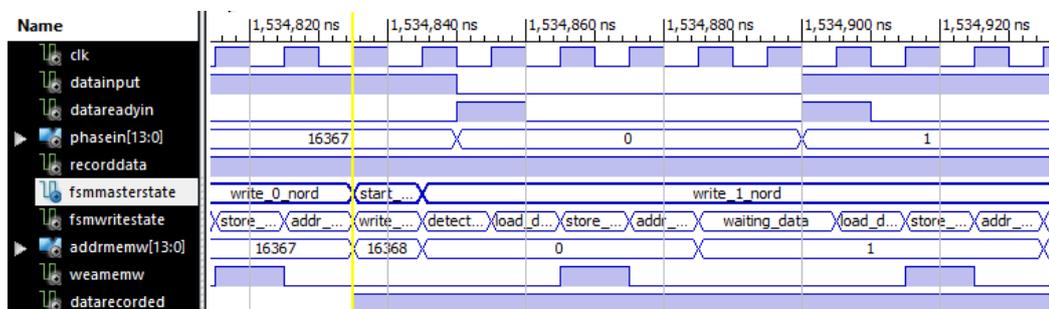


FIGURA 3.30: Fin del ciclo de grabación de la memoria 0, comienzo de grabación de la memoria 1

En ausencia de lectura, el módulo continúa grabando en cada una de las memorias alternadamente cada 1ms (el período de grabación) tal como se indica en la figura 3.31.

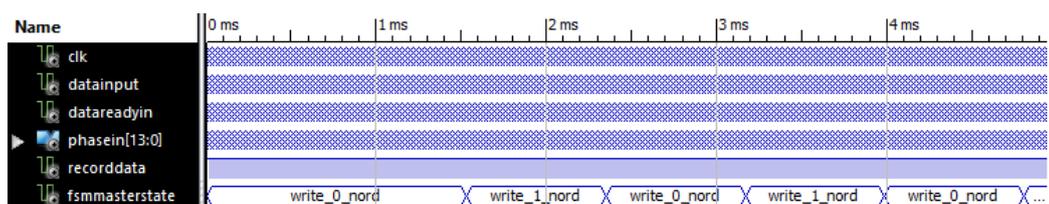


FIGURA 3.31: Escritura alternada de las memorias 0 y 1

Única lectura

En esta prueba se indicó al módulo de memoria que registre la señal de entrada, en este caso una señal cuadrada, por medio de la entrada de control (*recordData* = 1). Una vez detectado el fin de la grabación de la memoria 0 (*dataRecorded* = 1) se solicitó leer un bloque de muestras (*getData* = 1). Esto se puede observar en la figura 3.32.

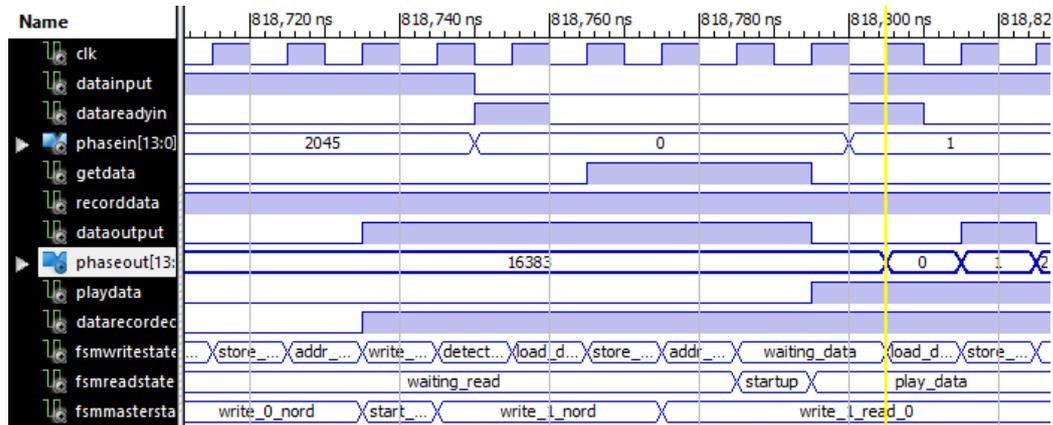


FIGURA 3.32: Fin de escritura de la memoria 1 y comienzo de lectura de memoria 0

En la figura 3.33 se observa un detalle de la reproducción de la señal de salida, en particular, la señal cuadrada reproducida a la frecuencia del clock del sistema indicando a través de *phaseOut* el número de muestra reproducida.

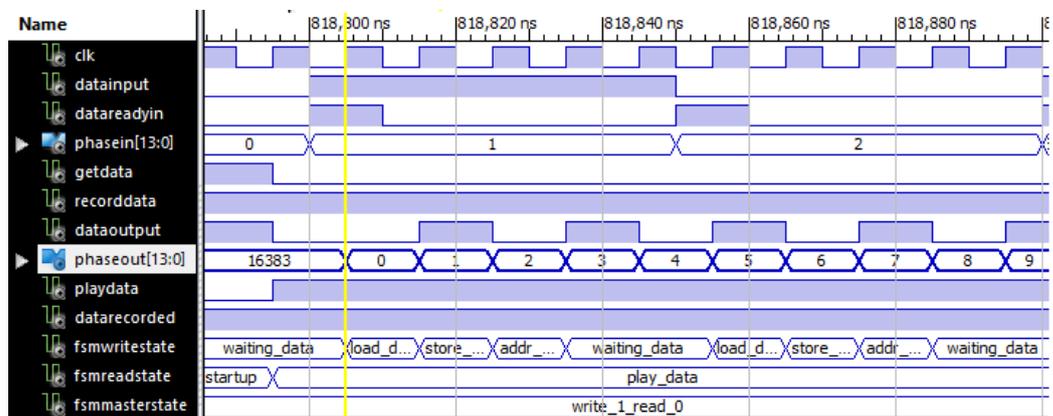


FIGURA 3.33: Lectura de memoria 0

En ausencia de comandos de lectura adicionales, el módulo de memoria permanece escribiendo los bancos de memoria alternadamente (ver figura 3.34).

Lectura/Escritura continua

En esta prueba se indicó al módulo que registre la señal de entrada, en este caso una señal cuadrada, por medio de la entrada de control (*recordData* = 1). Una vez detectado el fin de la grabación de la memoria 0 (*dataRecorded* = 1) se solicitó leer un bloque de muestras (*getData* = 1) una y otra vez.

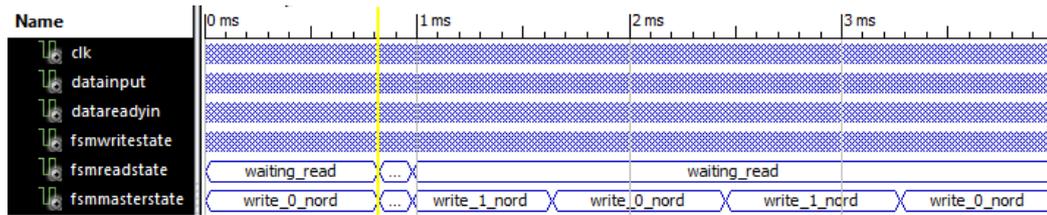


FIGURA 3.34: Escritura alternada de las memorias 0 y 1 en ausencia de nuevo comando de lectura

En la figura 3.35 se puede apreciar como ante sucesivos pedidos de lectura se procede a reproducir la memoria 0 mientras la memoria 1 se está grabando, permaneciendo el estado de la «FSM Maestra» en *write_1_read_0*.

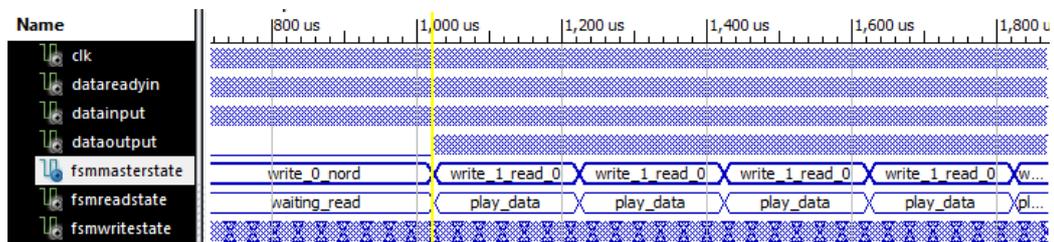


FIGURA 3.35: Lectura continua de la memoria 0 mientras la memoria 1 se está grabando

En la figura 3.36 se observa la conmutación de los bancos de memoria, intercambiando cual se usa para escritura y cual para lectura (la «FSM Maestra» cambia de estado de *write_1_read_0* a *write_0_read_1*). Nótese que la «FSM Escritura» pasa al estado *detect_frame* en espera de alinearse nuevamente en fase.

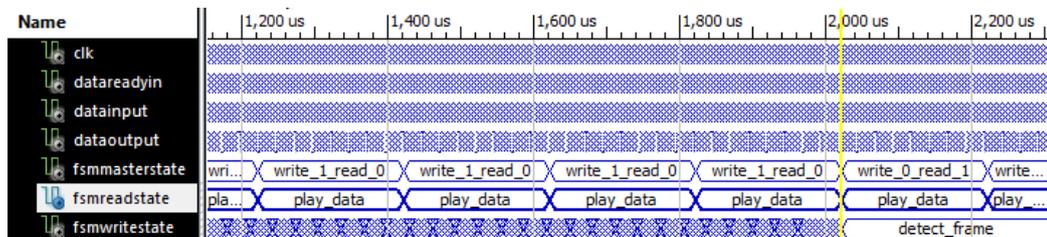


FIGURA 3.36: Detalle del cambio de escritura/lectura de la memoria 1 a la memoria 0

3.9. Módulo de generación de señal GPS local

El módulo de generación de señales GPS tiene por objeto generar, grabar internamente y reproducir réplicas de las señales GPS en cuadratura las cuales se utilizan para implementar el algoritmo de búsqueda.

3.9.1. Interfaz

En la figura 3.37 se indica la interfaz del módulo de generación de señal GPS local, estando su descripción detallada en la tabla 3.8.

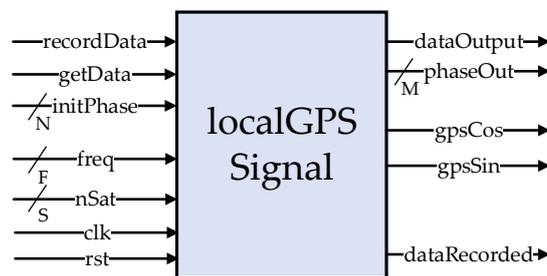


FIGURA 3.37: Interfaz del módulo de generación de señal GPS local

TABLA 3.8: Descripción de la interfaz del módulo de generación de señal GPS local

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
freq	IN	std_logic_vector (F)	Desplazamiento de frecuencia respecto de la FI
nSat	IN	std_logic_vector (S)	Selección de número de satélite
initPhase	IN	std_logic_vector (N)	Selección de fase inicial
recordData	IN	std_logic	Comenzar registro de datos
gpsCos	OUT	std_logic	Salida réplica coseno
gpsSin	OUT	std_logic	Salida réplica seno
dataOutput	OUT	std_logic	Salida de datos
phaseOut	OUT	std_logic_vector (M)	Número de muestra de la señal de salida
playData	OUT	std_logic	Indicador de datos de salida válidos
dataRecorded	OUT	std_logic	Indicador de datos disponibles

3.9.2. Operación

El módulo de generación de señales GPS genera, graba y reproducir réplicas de las señales GPS en cuadratura en dos memorias (*blockRAM*) internas de 16Kbit cada una. Las señales GPS consisten básicamente en el producto del código C/A

de un satélite; que se indica a través de la entrada $nSat$ y una señal portadora de FI seno o coseno cuya frecuencia está desplazada de la frecuencia de FI ($f_{FI} = 4,092MHz$) una cantidad indicada por la entrada $freq$. Una vez grabadas las señales, la salida $dataRecorded$ se pone en estado alto indicando que las muestras están disponibles, pudiendo ser extraídas mediante la entrada $getData$. Como consecuencia, las señales se reproducen a través de las salidas $gpsCos$ y $gpsSin$.

La implementación desarrollada permite reproducir las señales réplica en cualquier fase inicial lo cual se aplicará a la operación de búsqueda desarrollada en el Capítulo 2 a la frecuencia del reloj del sistema, aumentando la velocidad de procesamiento.

En la figura 3.38 se desarrolla un diagrama en bloques del módulo. Éste ha sido desarrollado como una FSMD tal como se ha explicado en la sección 3.1.2.

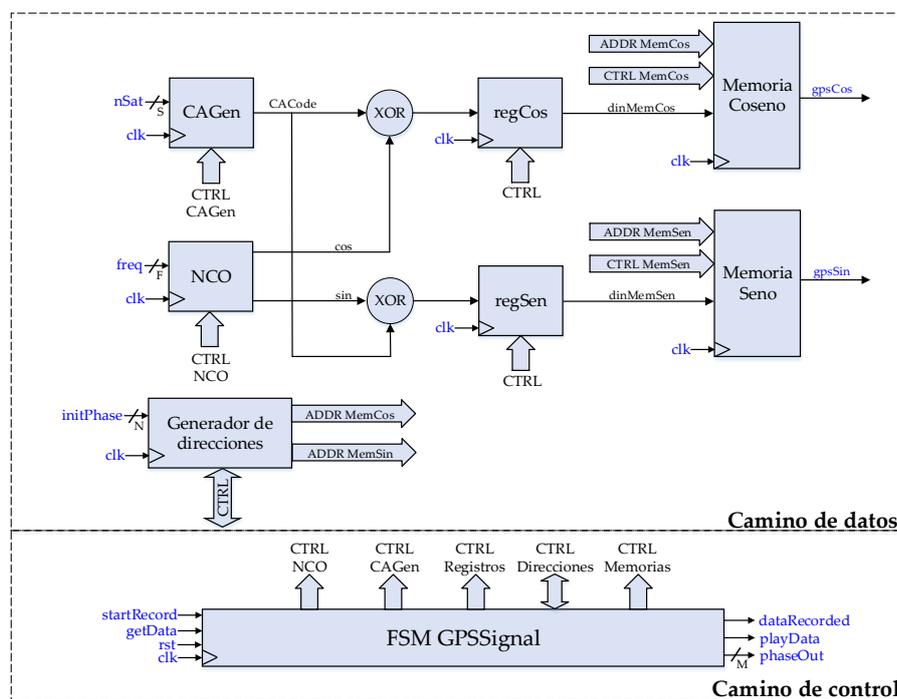


FIGURA 3.38: Diagrama en bloques del módulo de generación de señales local

El camino de control está compuesto por una máquina de estados finita «FSM GPSSignal» la cual controla los diferentes módulos del camino de datos, de acuerdo al estado interno de la misma y las entradas de comando $startRecord$ y $getData$.

El camino de datos está constituido por el generador de direcciones, los módulos de generación de señal, registros y dos memorias (denominadas «memoria Coseno» y «memoria Seno»). Los módulos de generación de código C/A y el oscilador controlado numéricamente (NCO) han sido descritos en secciones anteriores. La entrada del NCO ($freq$) se interpreta como un número no signado (con rango de 0 a 31), permitiendo variar la frecuencia central del NCO respecto de la frecuencia de la FI en pasos de $500Hz$ a ambos lados de la frecuencia central. La elección de $500Hz$ como paso de frecuencia se fundamenta en los resultados obtenidos en la sección 2.2.2, donde se concluyó que el máximo de correlación se

extingue en un rango de $\pm 1000\text{Hz}$ en torno a la frecuencia central. El rango de ajuste puede variarse en el intervalo $[-8000, 8000)\text{Hz}$.

Al recibir el comando $startRecord = 1$, la «FSM GPSSignal» configura los bloques de generación de señal (generador C/A y NCO) a su estado inicial e inicia la grabación de las señales, guardándolas previamente en los registros $regCos$ y $regSen$. Una vez finalizada la grabación, se indica la situación poniendo en alto la salida $dataRecorded$.

Una vez que la señal ha sido grabada, ésta puede reproducirse poniendo en alto la entrada de control $getData$. A fin de comunicar la reproducción a los «módulos cliente» la señal $playData$ se pone en alto dos ciclos de reloj antes del inicio de la secuencia de datos.

3.9.3. Pruebas

Con el fin de comprobar la correcta generación y reproducción de las señales réplica se realizó un *testbench* donde se solicitó la generación de una señal correspondiente a un satélite y a un desplazamiento dado. Dicha señal se grabó en el *testbench* por medio de los procedimientos *write* y *read* del paquete *textio*[27]. El archivo de texto generado se procesó utilizando los *scripts* desarrollados en la etapa de estudio de este trabajo con el objeto de validar el algoritmo de búsqueda serial (para mayor detalle sobre la validación de la señal réplica GPS ver [28]).

En la figura 3.39 se observa un diagrama de señales en función del tiempo para el inicio de grabación de las réplicas. En dicha figura, se observan las señales correspondientes al código C/A $CACode$ y las salidas en cuadratura del NCO (cos y sin), así como su mezcla ($regcosin$ y $regsinin$). El generador de direcciones va actualizando el registro de direcciones ($addrreg$) a medida que se graban las muestras, el cual alimenta el bus de direcciones de ambas memorias.

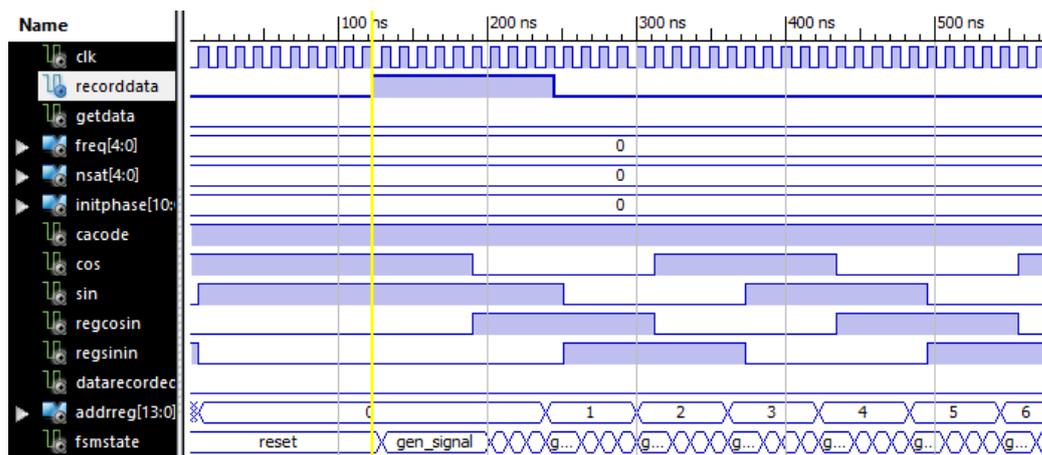


FIGURA 3.39: Inicio de la grabación de señales réplica

En la figura 3.40 se pone de manifiesto la inversión de fase de las señales réplica ($regcosin$ y $regsinin$) como consecuencia de la variación del código C/A (ver cursor amarillo en la transición de $cacode$).

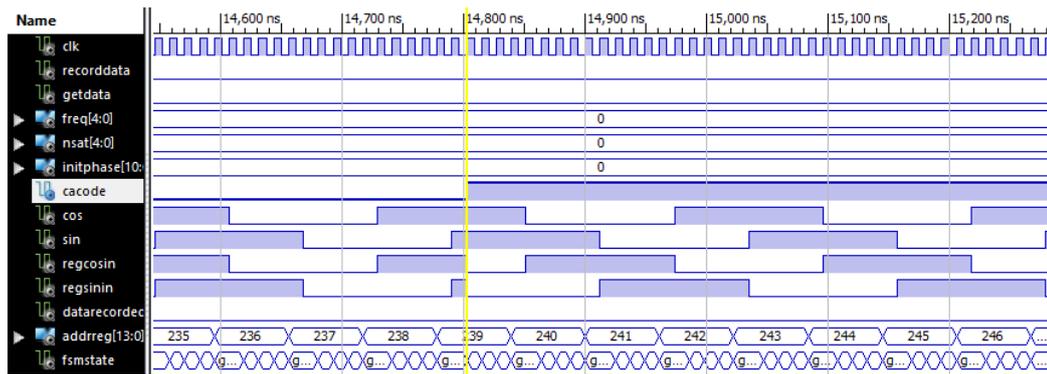


FIGURA 3.40: Detalle de la inversión de fase de las señales réplica

En la figura 3.41 se indica el fin de la grabación de un ciclo de la señal réplica. La salida *dataRecorded* se pone en alto indicando el fin de la grabación y que el módulo está disponible para reproducir las señales. El estado de la «FSM GPSSignal» pasa a *recordComplete*.

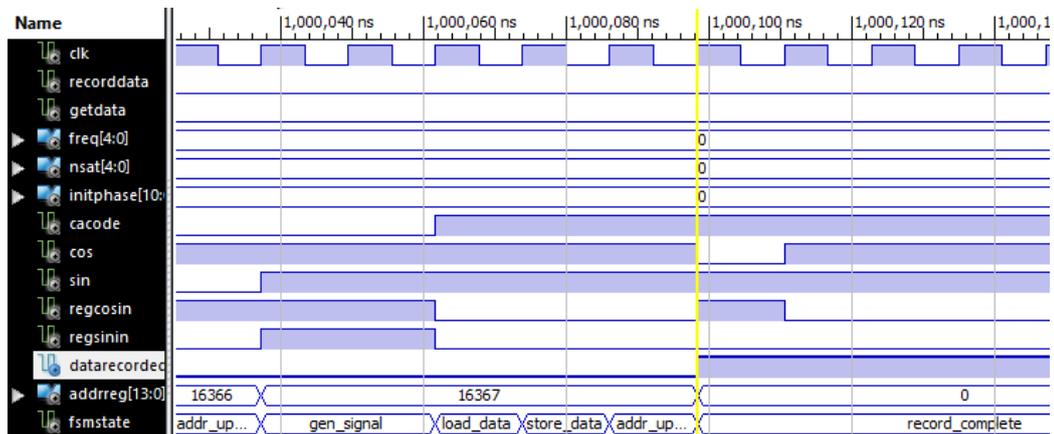


FIGURA 3.41: Fin de la grabación de las señales réplica

Para reproducir las señales réplica se le envía al módulo el comando *getData* = 1, en consecuencia, según la fase inicial requerida (*initPhase*), la «FSM GPSSignal» cargará al generador de direcciones con una dirección inicial distinta al comienzo de la reproducción. La fase inicial (*initPhase*) se interpreta como un número no signado en el intervalo $[0, 2046)$ y representa el desplazamiento en fase de la señal réplica en medio chip del código C/A. Teniendo en cuenta que la frecuencia de muestreo es $f_s = 16,368\text{MHz}$ y la tasa de datos del código C/A es de $1,023\text{MS/s}$ se toman 16 muestras por cada chip. Así, un desplazamiento de medio chip equivaldrá a un desplazamiento de 8 muestras. En la figura 3.42 se ilustra un diagrama de señales para *initPhase* = 0, lo que corresponde a reproducir las memorias a partir de la dirección *addrreg* = 0.

En la figura 3.43 se solicita la reproducción de las muestras desplazadas 512 chips (*initPhase* = 1024). Se observa que la dirección inicial vale *addrreg* = $1024 * 8 = 8192$.

Una vez finalizada la lectura de la última muestra de la memoria (*addrreg* = 16367, ver figura 3.44) se procede a leer a partir de la dirección inicial nuevamente (*addrreg* = 0), ya que la memoria se utiliza como un *buffer* circular.

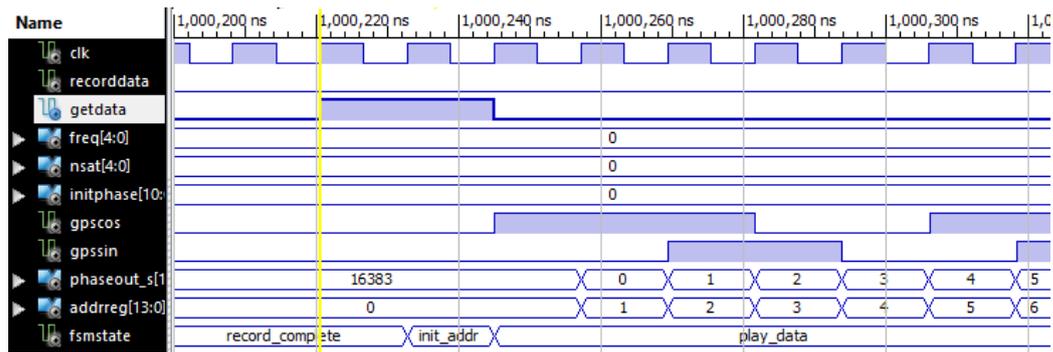
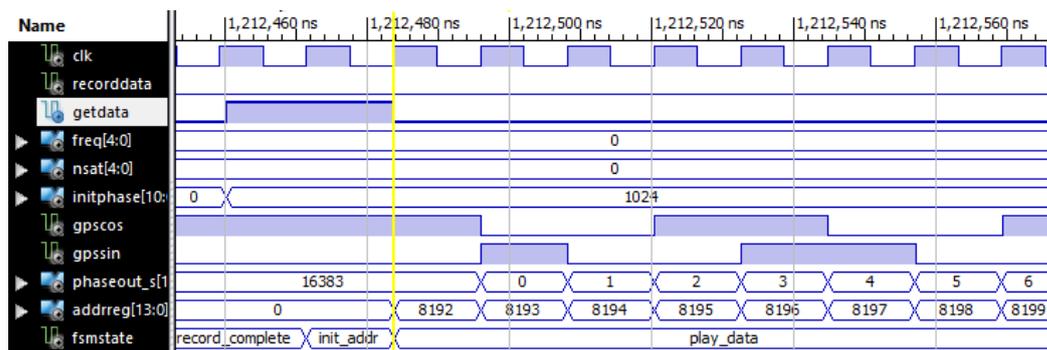
FIGURA 3.42: Lectura de las señales réplica (*initPhase=0*)FIGURA 3.43: Lectura de las señales réplica (*initPhase=1024*)

FIGURA 3.44: Detalle del cambio de dirección de lectura

3.10. Módulo de búsqueda

El módulo de búsqueda tiene por objeto realizar la correlación de la señal de entrada bajo diferentes condiciones para implementar en tiempo real el algoritmo de búsqueda descrito en la sección 2.2.

3.10.1. Interfaz (núcleo)

En la figura 3.45 se indica la interfaz del núcleo funcional del módulo de búsqueda, estando su descripción detallada en la tabla 3.9.

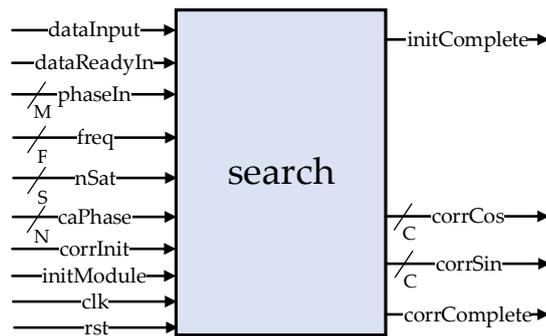


FIGURA 3.45: Interfaz del núcleo del módulo de búsqueda

TABLA 3.9: Descripción de la interfaz del núcleo del módulo de búsqueda

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
dataInput	IN	std_logic	Entrada de datos
dataReadyIn	IN	std_logic	Indicador de dato de entrada válido
phaseIn	IN	std_logic_vector (M)	Número de muestra de la señal de entrada
freq	IN	std_logic_vector (F)	Desplazamiento de frecuencia respecto de la FI
nSat	IN	std_logic_vector (S)	Selección de número de satélite
caPhase	IN	std_logic_vector (N)	Selección de fase inicial
corrInit	IN	std_logic	Iniciar correlación
initModule	IN	std_logic	Inicialización módulo
initComplete	OUT	std_logic	Inicialización completada
corrComplete	OUT	std_logic	Correlación finalizada
corrCos	OUT	std_logic_vector (C)	Correlación réplica coseno
corrSin	OUT	std_logic_vector (C)	Correlación réplica seno

3.10.2. Operación

El módulo de búsqueda tiene por función realizar la correlación entre la señal de entrada y las réplicas generadas localmente, para diferentes:

- Número de satélite (entrada *nSat*)
- Frecuencia de portadora (FI) (entrada *freq*)
- Fase de código C/A (entrada *caPhase*)

En la figura 3.38 se presenta un diagrama en bloques del módulo, el cual ha sido desarrollado como una FSMD tal como se ha explicado en la sección 3.1.2.

El camino de control está compuesto por una máquina de estados finita «FSM Search» la cual se encarga de generar las señales de control para los diferentes bloques del camino de datos (módulo de memoria, módulo de generación de señal GPS local, acumuladores y registros) a partir de su estado interno y las entradas de control (*initModule* y *corrInit*).

El camino de datos se constituye por:

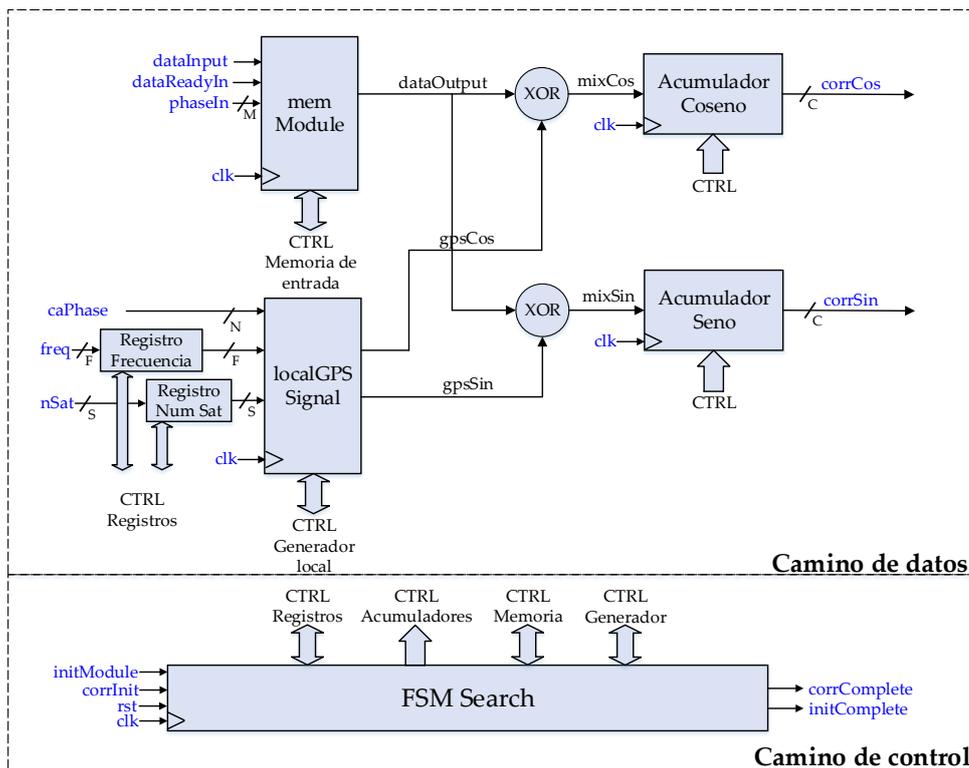


FIGURA 3.46: Diagrama en bloques del módulo de búsqueda

- **Módulo de memoria:** Su función es la de mantener un flujo de muestras continuo y actualizado de la señal de entrada (ver sección 3.8).
- **Generador de señal GPS local:** Su función es la de generar, almacenar y reproducir una réplica de una señal GPS con determinados parámetros iniciales ($freq$, $initPhase$ y $nSat$). Para mayor detalle ver sección 3.9.
- **Acumuladores:** Su función es ir incrementando / decrementando su cuenta interna en caso de que la señal de entrada concuerde / no concuerde con la señal generada localmente.
- **Registros:** Los registros de frecuencia y número de satélite guardan el último valor utilizado para las entradas $nSat$ y $freq$.

El módulo de búsqueda se inicializa poniendo en alto la entrada $initModule$, esto inicia la grabación de muestras de la señal de entrada. Cuando el módulo de memoria tiene un bloque de muestras disponible se pone en alto la salida $initComplete$. Una vez completada la inicialización, es posible correlacionar la señal de entrada con las señales de réplica locales poniendo en alto la entrada $corrInit$. No obstante, estas señales se deben generar y grabar previo a su reproducción por el módulo de generación de señales GPS local (ver sección 3.9).

Si se supone que el módulo se acaba de inicializar, se invertirá 1ms (un período de código C/A) adicional en generar las réplicas, luego la correlación se hará a la frecuencia de reloj del sistema ($f_{clk} = 81,84MHz$), tomando alrededor de $200\mu s$ cada correlación.

Si se cambia la fase inicial de código C/A (*caPhase*) no será necesario regenerar las señales réplicas locales ya que esta operación consiste únicamente en desplazar la dirección de memoria inicial del módulo de generación de señales locales (ver sección 3.8). En caso de que se cambie el número de satélite (entrada *nSat*) o la frecuencia (entrada *freq*), el módulo detectará automáticamente la condición y forzará la regeneración de las señales locales previo a realizar la correlación.

3.10.3. Pruebas

A fin de comprobar el correcto funcionamiento del módulo se planteó un *testbench* donde se inicializó el módulo y se solicitó la correlación de una señal de entrada arbitraria (una señal cuadrada) con la réplica local. A continuación, se desarrollan los resultados obtenidos.

En la figura 3.47, se indica la transición de la entrada *initModule*, de esta manera el estado de la «FSM Search» pasa a *record_input_data*, es decir, que inicializará el módulo de memoria a la espera de muestras por procesar.

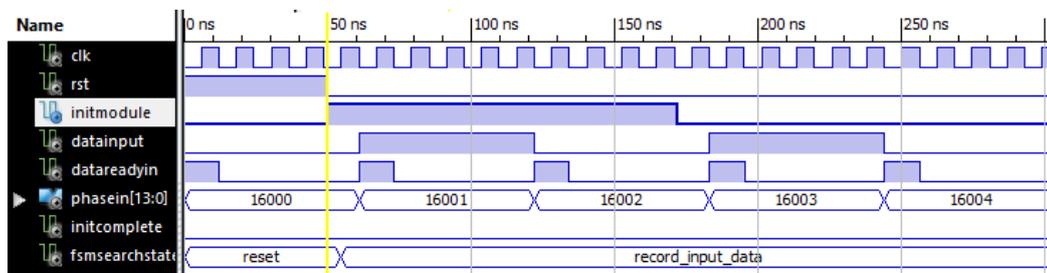


FIGURA 3.47: Comienzo de la inicialización del módulo de búsqueda

En la figura 3.48 se puede observar el fin de la inicialización, donde es importante notar ya se ha completado la grabación de un ciclo de muestras (la muestra final es la número 16367 indicada por el cursor amarillo).

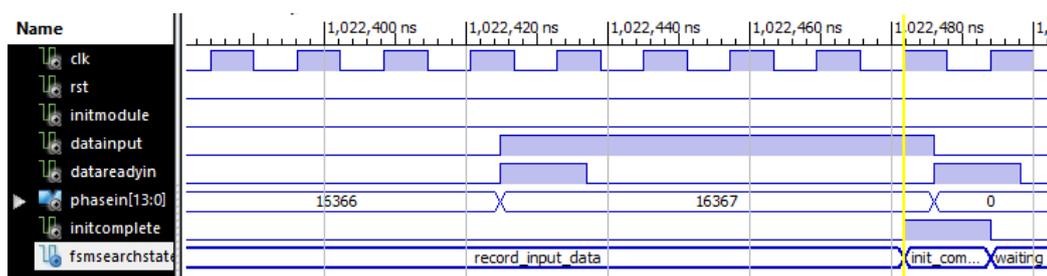


FIGURA 3.48: Fin de la inicialización del módulo de búsqueda

A continuación, se solicita el inicio de la correlación (*corrInit* = 1). Dado que es la primera vez que se utiliza el módulo, será necesario generar las señales réplica; la «FSM Search» pasa al estado *record_gps_freq_sat* (ver figura 3.49).

Una vez generada la señal de réplica se inicia la correlación. En la figura 3.50 se puede observar que el estado de la «FSM Search» (*fsmsearchstate*) pasa a *corr* y los acumuladores (*corrCos* y *corrSin*) comienzan a operar. Nótese que las señales correlacionadas están enfasadas, coincidiendo los valores de fase de ambas: *phaseout_s* (correspondiente al número de muestra del módulo de memoria) y

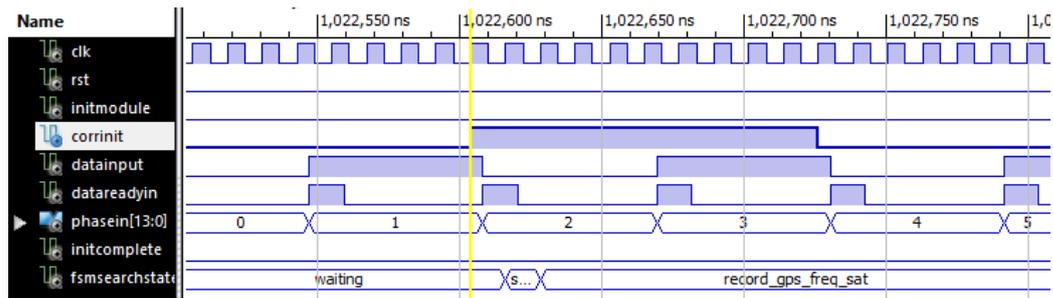


FIGURA 3.49: Inicio de generación de señal GPS local

phaseout (correspondiente al número de muestra del módulo de generación de réplicas).

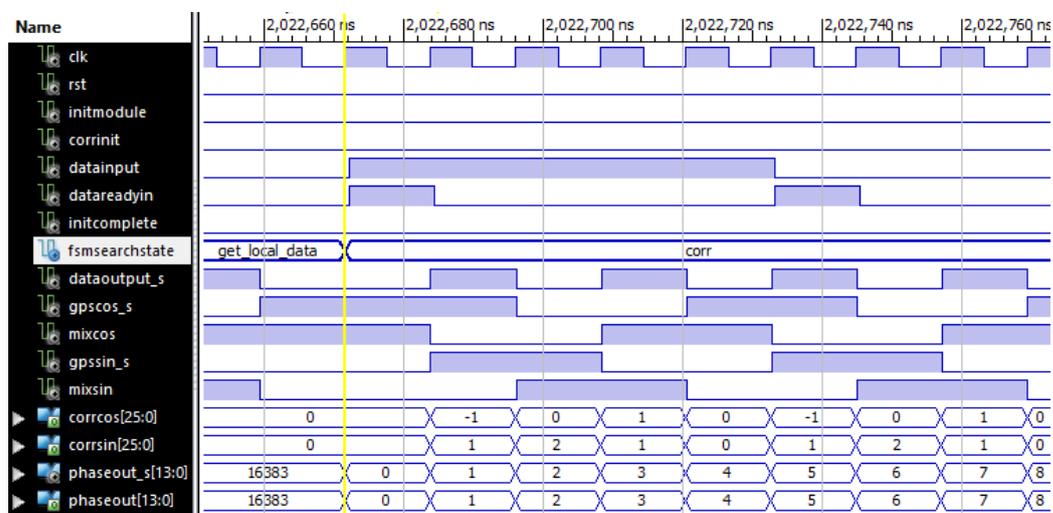


FIGURA 3.50: Inicio de la correlación de señal GPS local

Finalmente, cuando se ha completado la correlación, la salida *corrComplete* pasa a estado alto. Como se ha explicado anteriormente, en caso de que se inicie una nueva correlación variando sólo la fase de la señal inicial *caPhase*, ésta comenzará inmediatamente (ver figura 3.51). En caso de que se varíen otros parámetros (número de satélite y/o frecuencia), se necesitará pasar por el estado *record_gps_freq_sat* para regenerar las señales réplicas locales.

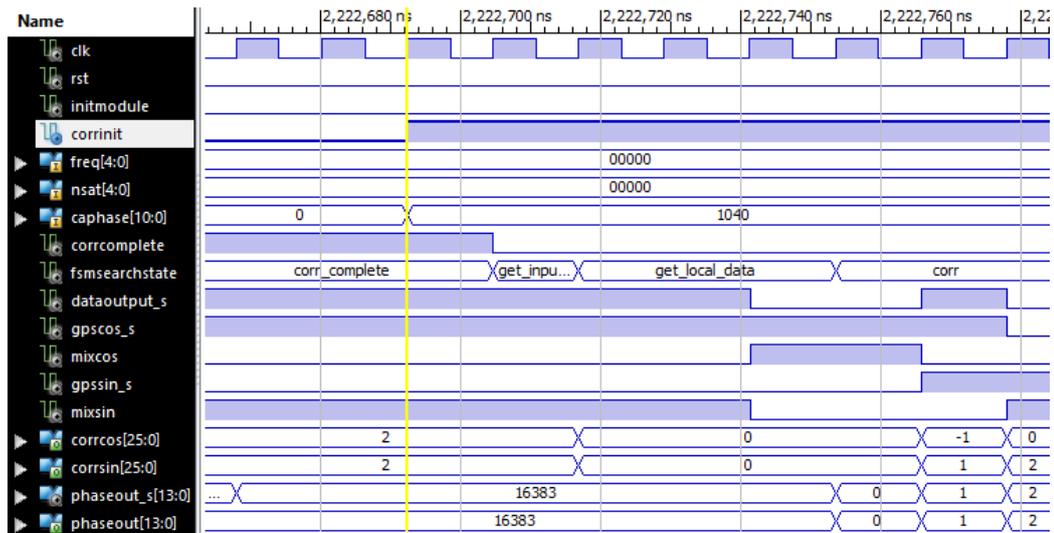


FIGURA 3.51: Fin de la correlación de señal GPS local

3.10.4. Interfaz (periférico)

El periférico de búsqueda se compone del módulo de búsqueda explicado en la sección interior al que se le agregó una interfaz SPI (del inglés, *Serial Peripheral Interface*) para efectuar el control de las diferentes operaciones del módulo.

En la figura 3.52 se indica la interfaz del módulo de búsqueda con su interfaz SPI, estando su descripción detallada en la tabla 3.10.

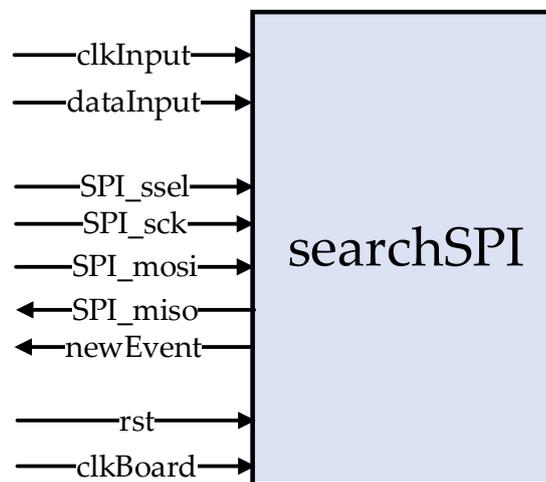


FIGURA 3.52: Interfaz del módulo de búsqueda

3.10.5. Operación

En la figura 3.53 se muestra un diagrama en bloques del módulo de búsqueda con su interfaz SPI.

TABLA 3.10: Descripción de la interfaz del módulo de búsqueda

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clkBoard	IN	std_logic	Reloj
dataInput	IN	std_logic	Entrada de datos del <i>frontend</i>
clkInput	IN	std_logic	Reloj del <i>frontend</i>
SPI_ssel	IN	std_logic	Entrada de selección (SPI)
SPI_sck	IN	std_logic	Entrada de reloj (SPI)
SPI_mosi	IN	std_logic	Entrada de datos maestro->esclavo (SPI)
SPI_miso	OUT	std_logic	Salida de datos esclavo->maestro (SPI)
newEvent	OUT	std_logic	Evento

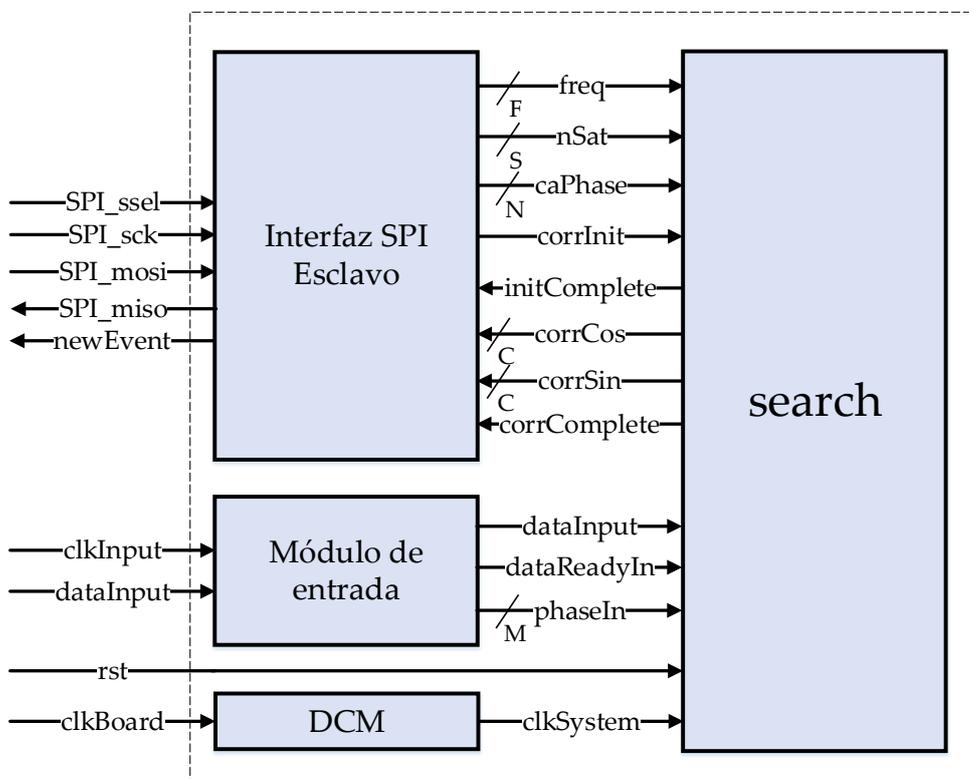


FIGURA 3.53: Diagrama en bloques del módulo de búsqueda con su interfaz SPI

El módulo se compone de:

- **Interfaz SPI esclavo:** este bloque provee el canal de comunicación entre el módulo de búsqueda y el exterior a través de una interfaz SPI de 16 bits.
- **Módulo de entrada:** este bloque sirve de interfaz entre el *frontend* y el módulo de búsqueda y su funcionamiento se describió en la sección 3.7.

- **DCM (Digital clock manager)[19]:** El gestor de reloj genera la señal de reloj de frecuencia $81,84MHz$ (aproximadamente) a partir del reloj del kit de desarrollo ($50MHz$).
- **Módulo de búsqueda:** este bloque es el núcleo del periférico y realiza la operación de búsqueda tal como se describió en la sección 3.10.1.

El protocolo de comunicación SPI es un protocolo de comunicación serial[29] ampliamente conocido. Fue desarrollado en los años 80 por Motorola Inc. y se ha convertido en un estándar de facto en el ámbito de los sistemas embebidos. Este protocolo cuenta con varias ventajas que fundamentan su elección para este trabajo:

- Se encuentra ampliamente distribuido y existe gran variedad de bibliografía existente, además de *cores* ya desarrollados en lenguaje VHDL, como el [30] utilizado como base para este trabajo.
- Puede funcionar en un amplio rango de frecuencias (se utilizó una frecuencia de $5MHz$ para el diseño de la interfaz) por lo que su utilización no ofrece una penalización temporal al controlar el módulo.
- Su implementación se puede describir por medio de lógica RTL (*Register Transfer Logic*) por lo que el uso de recursos de la FPGA es reducido.
- El protocolo puede analizarse exhaustivamente mediante un analizador de protocolos.

El conjunto del módulo de búsqueda, de entrada, DCM y la interfaz SPI, al que se denominará «periférico de búsqueda» se controla a través de cuatro registros de control: *CONTROL*, *FREQ*, *NSAT* y *CAPHASE*, con direcciones $0000|_{16}$, $0001|_{16}$, $0002|_{16}$ y $0003|_{16}$ respectivamente. Las operaciones se efectúan enviando en primer lugar la dirección del registro y en segundo lugar su valor. Dado que el protocolo SPI es del tipo «maestro esclavo», todas las transacciones, incluso las de lectura, son originadas en el maestro. De esta manera, resultó necesario agregar un pin adicional *newEvent* al periférico esclavo que genera un pulso positivo de $1\mu s$ de duración para brindarle la posibilidad de informar al maestro determinadas circunstancias. *newEvent* se utiliza para indicar al maestro la inicialización del módulo y cuando se ha finalizado una correlación para proceder a la lectura de su valor.

3.10.6. Pruebas

A fin de comprobar el correcto funcionamiento del módulo se planteó un *testbench* donde se simuló la operación del periférico de búsqueda a través de la interfaz de comunicación SPI. En primer lugar, se envió el comando $0000\ 0001|_{16}$ (*registro: CONTROL, comando: INICIALIZAR_MODULO*) tal como se observa en la figura 3.54. El registro *di_i_m* indica el valor de palabra enviado por el extremo maestro (haciéndose efectivo cuando la señal *wren_i_m* pasa a uno). Las señales *spissel*, *spisck*, *spimosi* y *spimiso* conforman el bus SPI. Se observa en la posición del cursor rojo como la señal *initModule* pasa de cero a uno un breve tiempo después de recibir el comando.

Tal como se describió en la sección 3.10.1, cuando el módulo completa su inicialización, pone en uno su salida *initComplete* tal como se muestra en la figura 3.55.



FIGURA 3.54: Señales de inicialización del periférico de búsqueda

A continuación la salida *newEvent* inicia el pulso de $1\mu s$ de duración para avisar al extremo maestro que el proceso de inicialización se completó y el módulo está listo para realizar una operación de correlación. El conjunto de operaciones de correlación conforman el proceso de búsqueda tal como se describirá en la sección 4.2.



FIGURA 3.55: Señales de inicialización completada

Antes de iniciar una operación de correlación es necesario fijar los parámetros: frecuencia *FREQ*, número de satélite *NSAT* y fase inicial *CAPHASE*. En la figura 3.56 se denota como se envían los comandos $0001\ 0014\ 0002\ 0000\ 0003\ 03FF|_{16}$ equivalentes a configurar $FREQ = 2000Hz$, $NSAT = 1$, $CAPHASE = 1023|_{10}$. Seguidamente, se envía el comando $0000\ 0002|_{16}$ (registro: *CONTROL*, comando: *INICIAR_CORRELACION*).

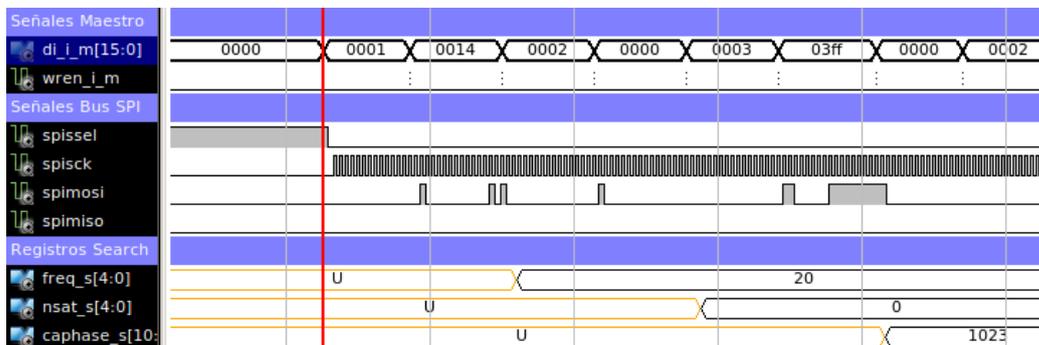


FIGURA 3.56: Señales de configuración de correlación

En la figura 3.57 se grafica en detalle como luego de recibir el comando de inicio de correlación, un breve tiempo después la entrada *corrInit* se pone en uno, iniciándose la operación de correlación.

Cuando la operación de correlación finaliza, se pone en uno la señal *corrComplete* y se activa la salida *newEvent* indicando al extremo maestro que los valores de correlación están disponibles y pueden leerse (ver figura 3.58).

En la figura 3.59 se denota la lectura de los registros de correlación (*corrCos* y *corrSin* de la sección 3.10.1). Como el bus SPI es dependiente de la iniciativa del extremo maestro, éste escribe dos valores *dummy* ($FFFF\ FFFF|_{16}$) en el bus para generar las señales de habilitación (*SPIssel*) y reloj (*SPIsck*). El periférico de

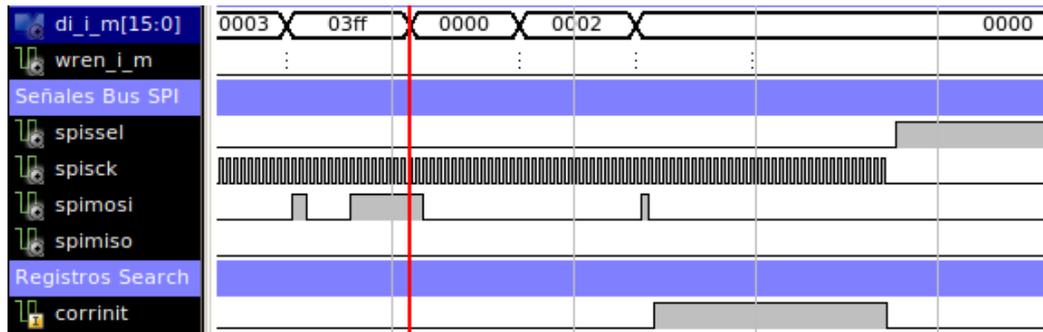


FIGURA 3.57: Señales de inicio de correlación



FIGURA 3.58: Señales de finalización de correlación

búsqueda (esclavo) introduce secuencialmente los valores de correlación calculados. En la señal *do_o_m_format* se aprecian los valores reportados ($-6750|_{10}$ y $6738|_{10}$) en sucesión.

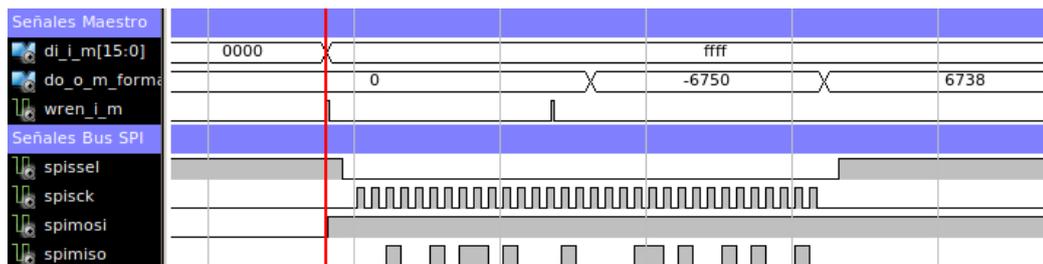


FIGURA 3.59: Señales de lectura de valores de correlación

3.11. Módulo de seguimiento

El módulo de seguimiento procesa la señal de entrada y genera datos que permiten mantener en fase la réplica local de portadora (en FI) y código C/A.

3.11.1. Interfaz

En la figura 3.60 se indica la interfaz del módulo de seguimiento, estando su descripción detallada en la tabla 3.9.

3.11.2. Operación

El objeto del módulo de seguimiento es obtener datos que permitan el seguimiento (tracking) de la señal de un satélite dado. En la figura 3.61 se muestra un diagrama en bloques del mismo, donde se puede observar que el módulo está implementado como una FSM según lo descrito en la sección 3.1.2.

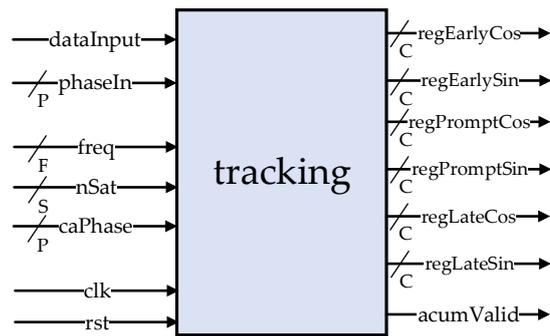


FIGURA 3.60: Interfaz del módulo de seguimiento

TABLA 3.11: Descripción de la interfaz del módulo de seguimiento

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clk	IN	std_logic	Reloj
dataInput	IN	std_logic	Entrada de datos
phaseIn	IN	std_logic_vector (P)	Número de muestra de la señal de entrada
freq	IN	std_logic_vector (F)	Desplazamiento de frecuencia respecto de la FI
nSat	IN	std_logic_vector (S)	Selección de número de satélite
caPhase	IN	std_logic_vector (P)	Selección de fase de señal C/A
setStatus	IN	std_logic	Seleccionar nuevos parámetros
acumValid	OUT	std_logic	Indicador de acumuladores válidos
regEarlyCos	OUT	std_logic_vector (C)	Correlación C/A Early Coseno
regEarlySin	OUT	std_logic_vector (C)	Correlación C/A Early Seno
regPromptCos	OUT	std_logic_vector (C)	Correlación C/A Prompt Coseno
regPromptSin	OUT	std_logic_vector (C)	Correlación C/A Prompt Seno
regLateCos	OUT	std_logic_vector (C)	Correlación C/A Late Coseno
regLateSin	OUT	std_logic_vector (C)	Correlación C/A Late Seno

El camino de control está compuesto por una máquina de estados finita «FSM Tracking» la cual se encarga de generar las señales de control para los diferentes bloques del camino de datos (NCO, generador de código C/A, acumuladores y registros) a partir de su estado interno y la entrada de control (*setStatus*).

El camino de datos está compuesto por:

- NCO:** Su función es la de generar una referencia de la señal de FI local cuya frecuencia se determina por medio de la entrada *freq*. Esta entrada se interpreta de forma signada y permite sintetizar señales seno y coseno con frecuencias en el rango $[f_I - 8kHz, f_I + 8kHz]$. En este caso, el registro interno del NCO se definió como de 32 bits, permitiendo en base al reloj del sistema $f_s = 81,84MHz$ obtener una precisión de frecuencia de $0,02Hz$ (ver sección 3.6).
- Generador de código C/A:** Su función es la de generar el código C/A del satélite indicado por *nSat*. El generador tiene asociado a su salida un desplazador que permite generar tres réplicas (*CAEarly*, *CAPrompt*, *CALate*) desplazadas 1/2 chip (8 muestras según se detalla en la tabla 3.1). Éstas se utilizan para generar señales de control que permiten predecir el desfase entre las réplicas locales y la señal de entrada

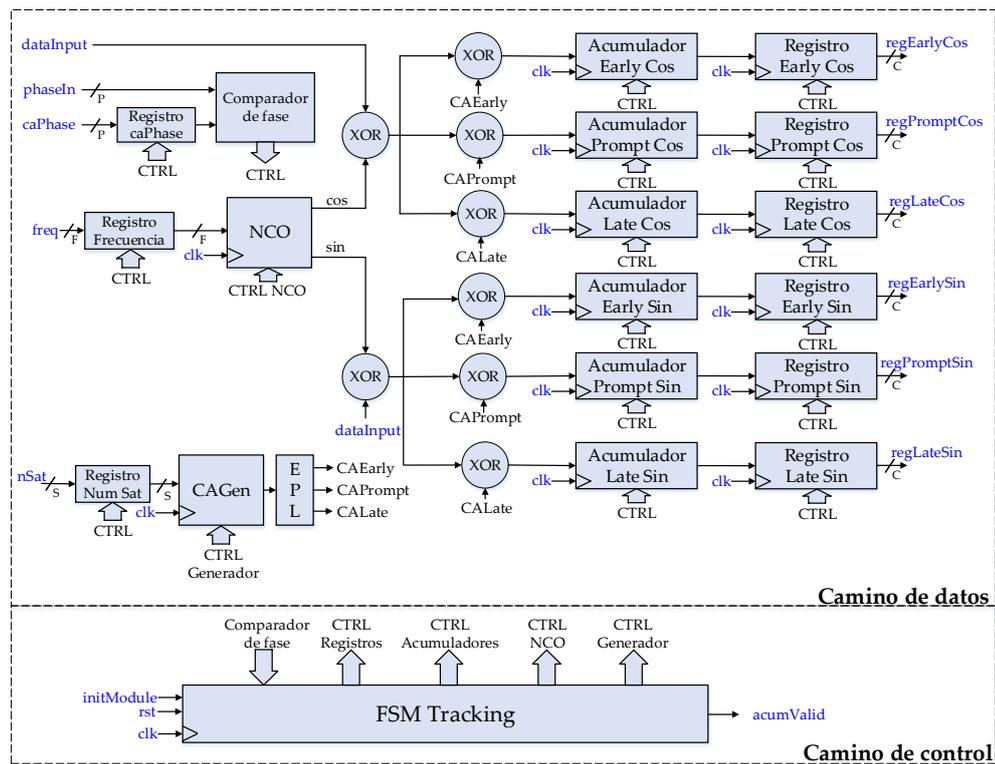


FIGURA 3.61: Interfaz del módulo de seguimiento

- **Acumuladores:** Su función es ir incrementando o decrementando su cuenta interna en caso de que la señal de entrada concuerde o no concuerde respectivamente con la señal generada localmente.
- **Registros:** Los registros de entrada (frecuencia, número de satélite y fase de código C/A) almacenan los parámetros de generación de señal que se utilizan en un momento dado. Los registros de salida (a continuación de los acumuladores) guardan el último dato válido generado por los acumuladores.

Las réplicas de los códigos C/A locales son idénticas salvo por su fase inicial:

- *C/A Prompt:* Es la réplica que se supone en fase con el código C/A de la señal de entrada.
- *C/A Early:* Está adelantada respecto de C/A Prompt en 1/2 chip, equivalente a 8 muestras.
- *C/A Late:* Está atrasada respecto de C/A Prompt en 1/2 chip, equivalente a 8 muestras.

La señal de entrada *dataInput* se mezcla en primer lugar con la réplica local de la señal de FI seno y coseno y luego con las tres versiones del código C/A (*CAEarly*, *CAPrompt*, *CALate*). Las seis señales generadas se acumulan durante un período de código C/A (1ms) y luego el valor obtenido se transfiere desde los acumuladores a los registros de salida. Los valores acumulados se utilizan para saber si la señal que está siendo recibida está en fase con las réplicas locales de la señal de FI y la señal de código C/A local.

Detección de fase de código

En esta subsección se enfocará la atención sobre la parte de la estructura que permite discriminar el desplazamiento del código C/A respecto de la réplica local. Para facilitar el análisis se partirá de la hipótesis de que la señal de entrada está a la misma frecuencia que la FI, evitando así que la diferencia de frecuencias degrade la correlación.

Si partimos de la suposición que la señal de entrada, en ausencia de ruido, está en fase con el código C/A *Prompt*, el valor de correlación para las señales C/A *Early* será igual y la mitad que para el valor de C/A *Prompt*, de acuerdo a las propiedades de autocorrelación del código (ver sección 2.1.2 y figura 2.3). Así, los valores de correlación para la señal C/A *Early* y C/A *Late* se pueden utilizar para discriminar si es necesario adelantar o atrasar la fase del código C/A local. Esta situación se ilustra en la figura 3.62, donde se grafica la autocorrelación normalizada.

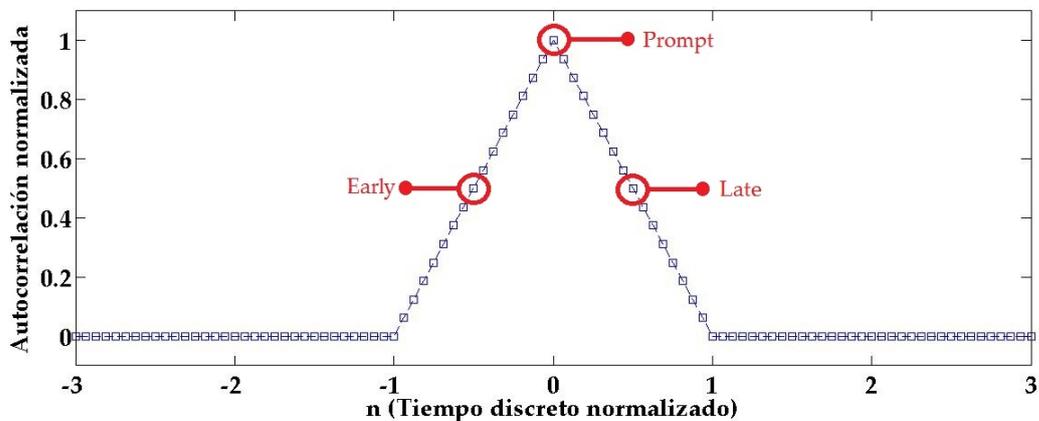


FIGURA 3.62: Valores de correlación normalizados para los códigos Early, Prompt y Late (señal en fase)

Si se supone que la señal de entrada empieza a atrasarse respecto de la señal C/A local, el valor para la señal C/A *Late* aumentará respecto del valor para la señal C/A *Early*. Esta situación debe compensarse supliendo un nuevo valor de fase de código C/A *caPhase* (ver figura 3.63).

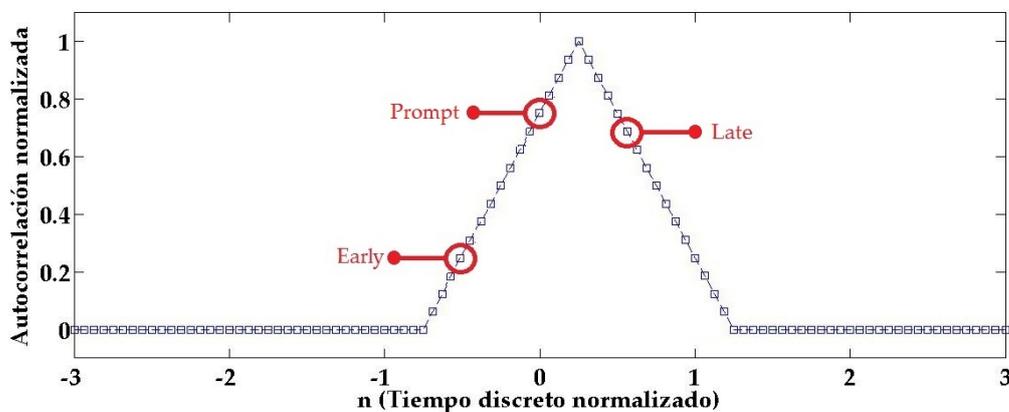


FIGURA 3.63: Valores de correlación normalizados para los códigos Early, Prompt y Late (señal atrasada)

En la figura 3.64 se ponen de manifiesto los bloques utilizados para mantener la sincronía del código C/A. El uso de dos ramas en cuadratura brinda a la estructura la capacidad de ser insensible a la fase de la señal de entrada [31].

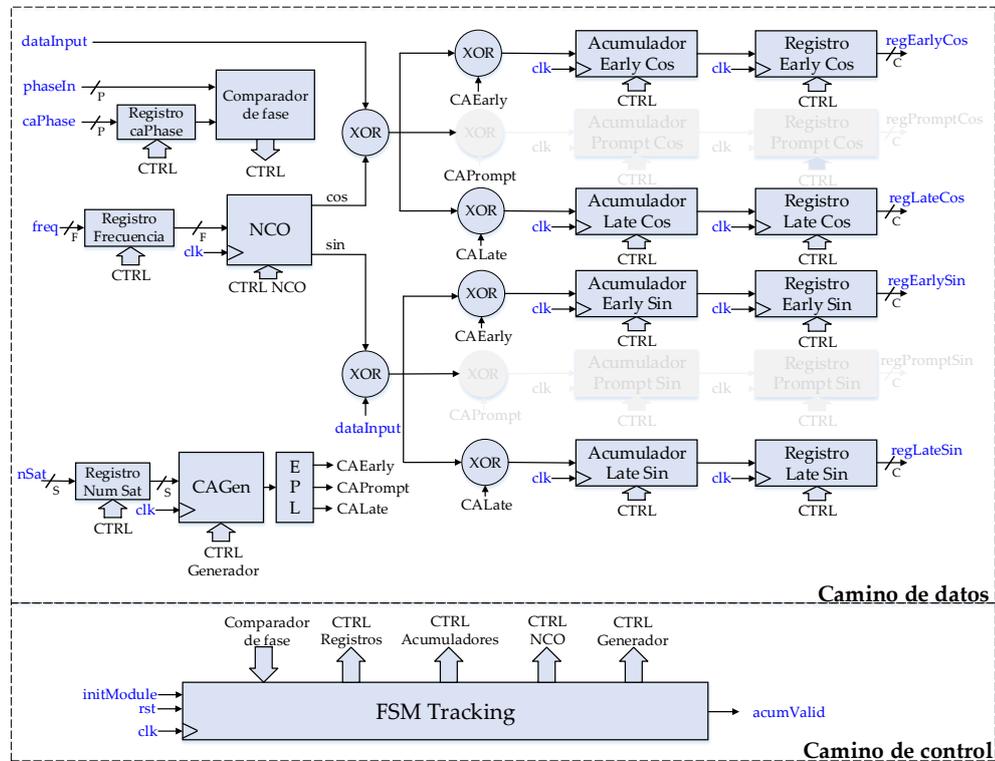


FIGURA 3.64: Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía del código C/A)

Detección de fase de portadora

En esta subsección se enfocará la atención sobre la parte de la estructura que permite discriminar el desplazamiento de la frecuencia de la señal de entrada respecto de la frecuencia de FI. Para facilitar el análisis se partirá de las mismas consideraciones anteriores, esto es, que el código C/A de la señal de entrada y el generado localmente están en fase, evitando así que la diferencia de fases degrade la correlación.

En la figura 3.65 se puede observar un diagrama en bloques de la estructura donde se ha puesto de manifiesto los bloques que se utilizan para detectar la sincronía fase/frecuencia de la señal de entrada respecto de la señal de FI. Técnicamente, el esquema de la figura representa un lazo de Costas [31, 32] sin la realimentación y es una estructura muy utilizada para la detección de las señales GPS [11], donde los acumuladores suplen los filtros pasabajos utilizados en la estructura tradicional del lazo.

En conclusión, el módulo de búsqueda permite generar valores de referencia capaces de discriminar el desfasaje de la señal de código C/A y portadora de la señal de entrada.

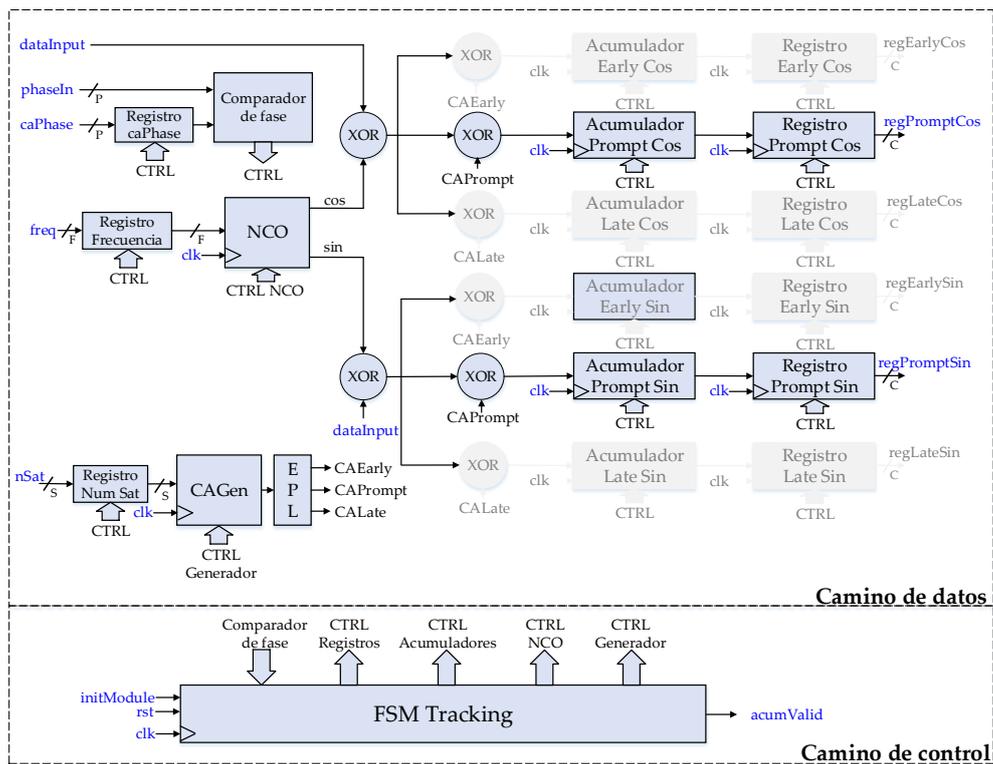


FIGURA 3.65: Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía de la señal de FI)

3.11.3. Pruebas

Con el objeto de probar el funcionamiento del módulo se diseñó un *testbench* en el que se inicializó el módulo con los parámetros de entrada: $freq = 0$, $caPhase = 160$ y $nSat = 1$.

Para inicializar el módulo, la entrada *setStatus* se pone en estado alto. En la figura 3.66 se puede observar que los parámetros de entrada (*freq*, *nSat* y *caPhase*) se guardan en los registros de entrada y la «FSM Tracking» pasa al estado *wait_ca_init*, en este estado aguardará para sincronizar el código C/A Prompt local con la muestra 160 de la señal de entrada.

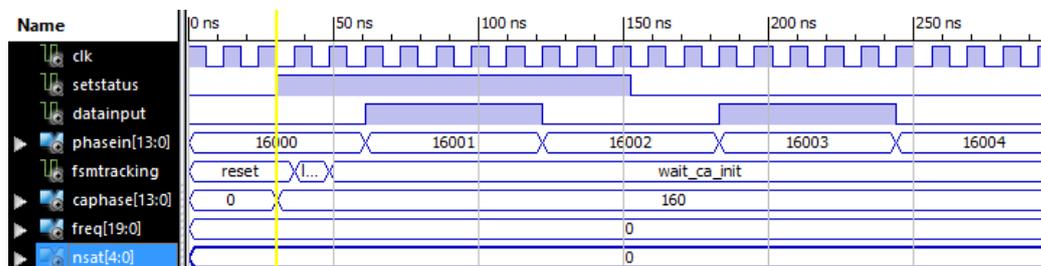


FIGURA 3.66: Inicialización módulo de seguimiento

En la figura 3.67 se puede notar como al llegar a la muestra 160, la «FSM Tracking» inicia la secuencia de código C/A. Estrictamente, el generador de código C/A sintetiza el código C/A Early mientras que el C/A Prompt y C/A Late se obtienen

desplazando el primero en 8 y 16 muestras respectivamente. En la figura 3.67, se observa un detalle de los códigos desplazados (el código C/A Prompt inicia en conjunto con la muestra 160 y el C/A Late con la 168). La «FSM Tracking» pasa al estado *wait_rollover* a la espera del inicio del *frame* (muestra 0).

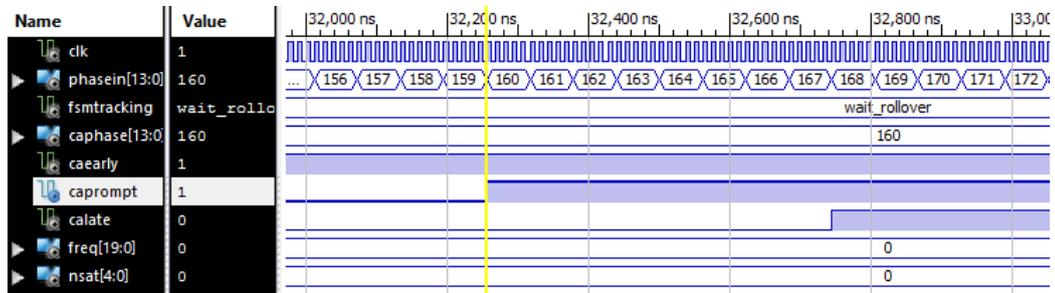


FIGURA 3.67: Sincronización del código C/A

En la figura 3.68 se puede observar al inicio del *frame* (muestra 0) las operaciones de acumulación actualizándose con cada nueva muestra. La «FSM Tracking» pasa al estado *corr*.

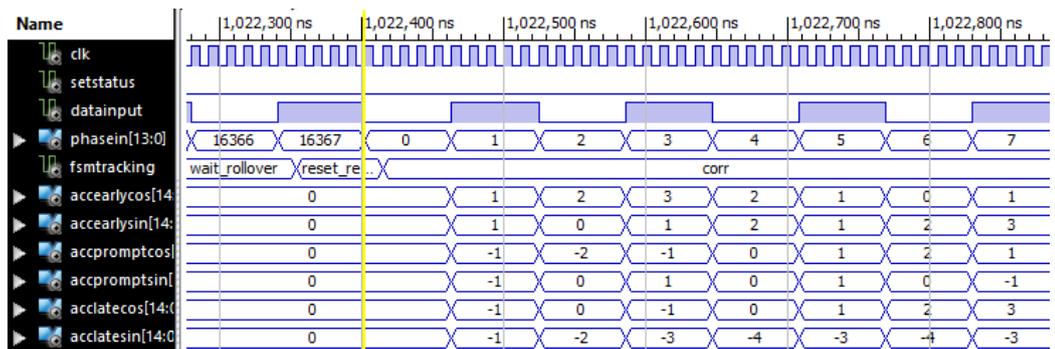


FIGURA 3.68: Inicio de la acumulación

En la figura 3.69 se puede observar al finalizar el *frame* (muestra 16367) la actualización de los registros de salida (*regEarlyCos*, *regEarlySin*, *regPromptCos*, *regPromptSin*, *regLateCos* y *regLateSin*). La salida *acumValid* pasa al estado alto indicando que se completó un ciclo de acumulación y los registros de salida pueden computarse. La «FSM Tracking» pasa brevemente al estado *reset_reg* a fin de poner en cero los acumuladores previo al inicio de un nuevo ciclo.

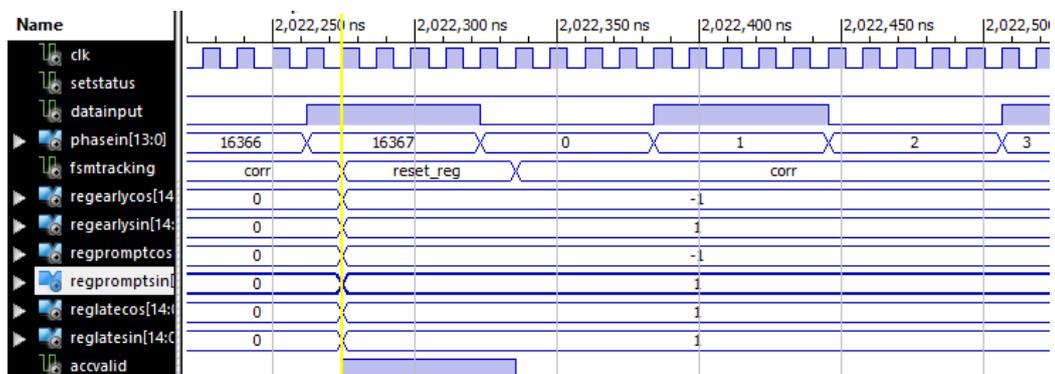


FIGURA 3.69: Fin de la acumulación

Las operaciones indicadas anteriormente continúan hasta que se establecen nuevos parámetros de entrada y se fuerza el refresco del módulo por medio de la entrada *setStatus*.

3.11.4. Interfaz (periférico)

El periférico de seguimiento se compone del módulo de seguimiento explicado en la sección anterior al que se le agregó una interfaz SPI para efectuar el control de las diferentes operaciones del módulo.

En la figura 3.70 se indica la interfaz del módulo de seguimiento con su comunicación SPI, estando su descripción detallada en la tabla 3.12.

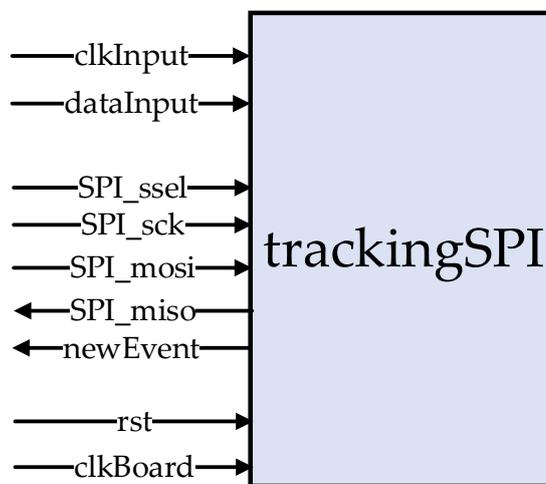


FIGURA 3.70: Interfaz del módulo de seguimiento

TABLA 3.12: Descripción de la interfaz del módulo de seguimiento

Señal	Modo	Tipo	Descripción
rst	IN	std_logic	Reset
clkBoard	IN	std_logic	Reloj
dataInput	IN	std_logic	Entrada de datos del <i>frontend</i>
clkInput	IN	std_logic	Reloj del <i>frontend</i>
SPI_ssel	IN	std_logic	Entrada de selección (SPI)
SPI_sck	IN	std_logic	Entrada de reloj (SPI)
SPI_mosi	IN	std_logic	Entrada de datos maestro->esclavo (SPI)
SPI_miso	OUT	std_logic	Salida de datos esclavo->maestro (SPI)
newEvent	OUT	std_logic	Evento

3.11.5. Operación

En la figura 3.71 se muestra un diagrama en bloques del módulo de seguimiento con su interfaz SPI.

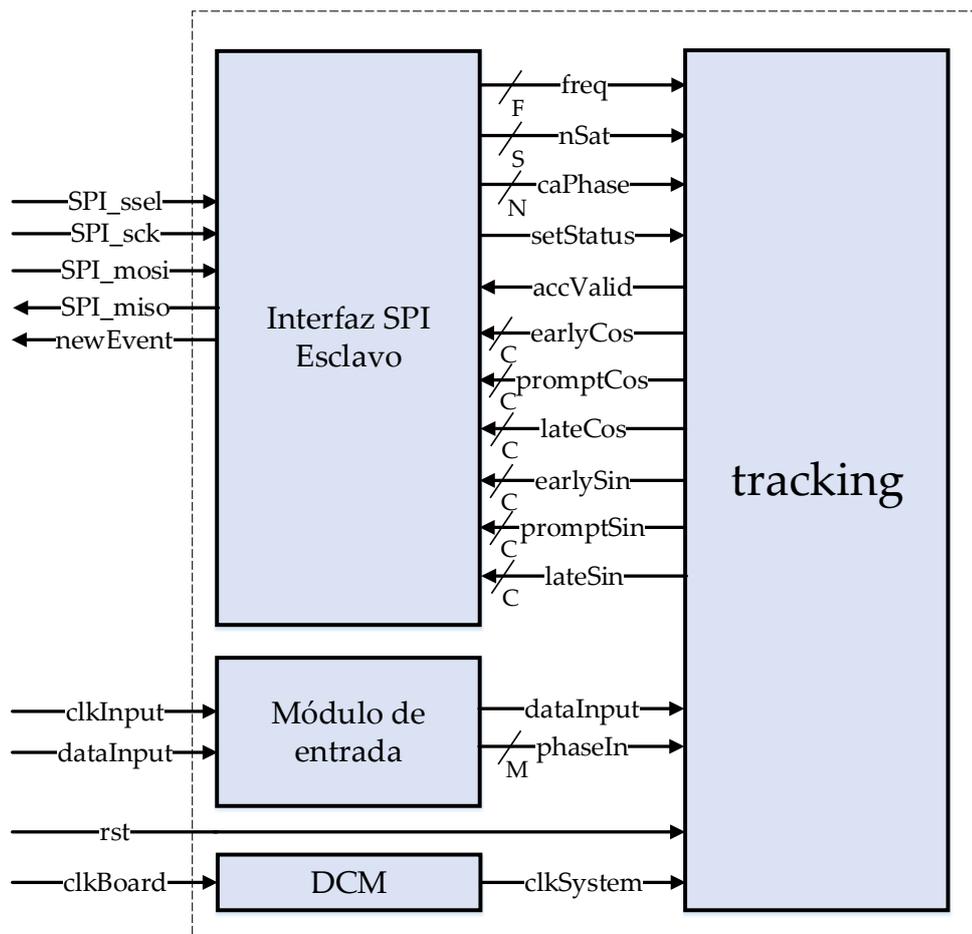


FIGURA 3.71: Diagrama en bloques del módulo de seguimiento con su interfaz SPI

El módulo se compone de:

- **Interfaz SPI esclavo:** este bloque provee el canal de comunicación entre el módulo de seguimiento y el exterior a través de una interfaz SPI de 16 bits.
- **Módulo de entrada:** este bloque sirve de interfaz entre el *frontend* y el módulo de seguimiento y su funcionamiento se describió en la sección 3.7.
- **DCM:** El gestor genera la señal de sincronismo de frecuencia $81,84\text{MHz}$ a partir del reloj del kit de desarrollo (50MHz).
- **Módulo de seguimiento:** este bloque es el núcleo del periférico y realiza la operación de seguimiento tal como se describió en la sección 3.11.1.

El conjunto del módulo de seguimiento, de entrada, DCM y la interfaz SPI, al que se denominará «periférico de seguimiento», se controla a través de cinco registros de control: *CONTROL*, *FREQ0*, *FREQ1*, *NSAT* y *CAPHASE*, con direcciones 0000_{16} , 0001_{16} , 0002_{16} , 0003_{16} y 0004_{16} , respectivamente. Las operaciones se efectúan enviando en primer lugar la dirección del registro y en segundo lugar su valor. El valor de frecuencia necesario para configurar el dispositivo es de 32 bits por lo que es necesario distribuir su valor en dos registros: la parte menos significativa en *FREQ0* y la más significativa en *FREQ1*. Dado que el protocolo

SPI es del tipo «maestro esclavo», todas las transacciones, incluso las de lectura, son originadas en el maestro. De esta manera, resultó necesario agregar un pin adicional *newEvent* al periférico esclavo que genera un pulso positivo de $1\mu s$ de duración para brindarle la posibilidad de informar al maestro determinados estados. La señal *newEvent* se utiliza para indicar al maestro que un período de correlación finalizó y los valores de los registros *earlyCos*, *promptCos*, *lateCos*, *earlySin*, *promptSin* y *lateSin* son válidos y pueden leerse.

3.11.6. Pruebas

A fin de comprobar el correcto funcionamiento del módulo se planteó un *testbench* donde se simuló la operación del periférico de seguimiento a través de la interfaz de comunicación SPI.

En primer lugar, se establecieron los valores iniciales de frecuencia, número de satélite y fase de código C/A. Para ello se enviaron los comandos:

- $0001\ 0001|_{16} : \text{FREQ0} = 0001|_{16}$
- $0002\ 0000|_{16} : \text{FREQ1} = 0000|_{16}$
- $0003\ 0002|_{16} : \text{NSAT} = 0002|_{16}$
- $0004\ 0003|_{16} : \text{CAPHASE} = 0003|_{16}$
- $0000\ 0001|_{16} : \text{CONTROL} = 0001|_{16}$
- $0000\ 0000|_{16} : \text{CONTROL} = 0000|_{16}$

Los primeros cuatro comandos transmiten a través de la interfaz SPI el valor de los parámetros de inicialización. Los últimos dos, generan el pulso de *setStatus* necesario para que el periférico grabe los valores en sus registros internos y se reconfigure en base a ellos. En la figura 3.72 se grafica esta situación donde el cursor rojo indica el cambio de valor del registro *FREQ*.

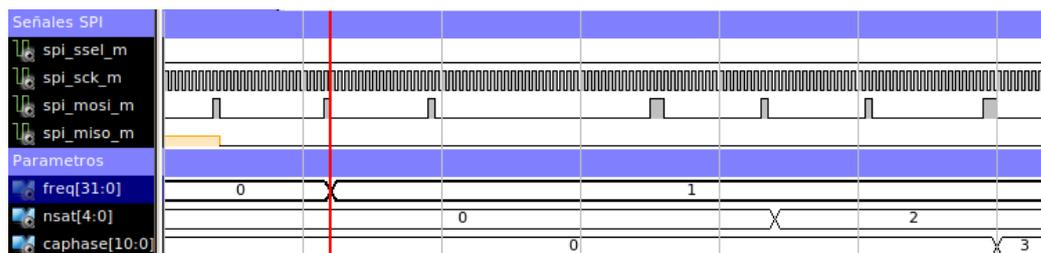


FIGURA 3.72: Inicialización de parámetros del periférico de seguimiento

Asimismo, en la figura 3.73 se muestra el pulso de *setStatus*, cuyo flanco de subida está indicado por el cursor vertical rojo. Como resultado de esto se habilita la configuración de los valores de frecuencia, número de satélite y fase de código C/A.

Al finalizar un período de correlación, el módulo presenta los valores de correlación (*EarlyCos*, *PromptCos*, *LateCos*, *EarlySin*, *PromptSin* y *LateSin*) y fija en alto la señal interna *accValid*. La interfaz SPI genera un pulso a través de la salida *newEvent* indicando al extremo maestro que el periférico tiene datos que reportar (ver figura 3.74).



FIGURA 3.73: Pulso *setStatus* que habilita la configuración del periférico

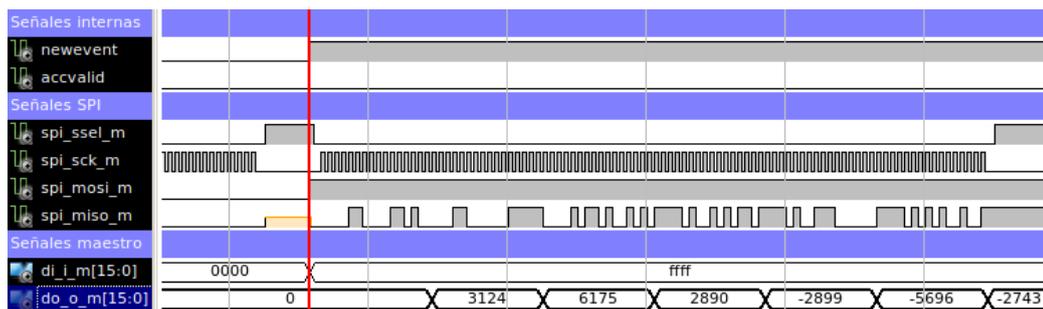


FIGURA 3.74: Lectura de los valores de correlación del periférico de seguimiento

Tal como se explicó en la sección 3.10.6, dado el funcionamiento del SPI que es dependiente de la iniciativa del extremo maestro, éste escribe seis valores *dummy* (FFF_{16}) en el bus para generar las señales de habilitación (*SPIssel*) y reloj (*SPIsck*); en respuesta, el periférico de seguimiento introduce secuencialmente los valores de correlación calculados. En la señal *do_o_m* de la figura 3.74 se aprecian los valores (3124_{10} , 6175_{10} , 2890_{10} , -2899_{10} , -5696_{10} y -2743_{10}) reportados en sucesión correspondientes en orden a *EarlyCos*, *PromptCos*, *LateCos*, *EarlySin*, *PromptSin* y *LateSin*.

Capítulo 4

Ensayos y Resultados

En este capítulo se describen diferentes validaciones y ensayos realizados para comprobar el correcto funcionamiento de los módulos planteados.

4.1. Validación de algoritmos de búsqueda utilizando señales reales

La validación del algoritmo de búsqueda planteado en la sección 2.2 es necesaria por dos razones:

- El algoritmo se utilizó como base para implementar el módulo de búsqueda planteado en la sección 3.10 por lo que resulta fundamental validar su funcionamiento con señales reales;
- A partir de la detección de los satélites en vista, es decir, aquellos en línea directa de vista con el receptor, se comprueba además el correcto funcionamiento del *frontend* presentado en la sección 3.2.

En esta sección se exponen diferentes pruebas, las cuales se realizaron a siguiendo la siguiente secuencia:

1. **Adquisición de señales reales:** se utilizó el módulo de adquisición detallado en la sección 3.3 conectado al *frontend* y a la memoria SRAM interna de la placa de desarrollo. En la figura 4.1 se indica el esquema de conexión y en la figura 4.2 se muestra una fotografía del sistema dispuesto para su uso. Al iniciarse la adquisición (*start=1*) se graba una secuencia de muestras provenientes del *frontend* en la memoria RAM. Luego, utilizando la aplicación *Digilent Adept 2*[33] se copian los datos binarios de la memoria a un archivo en la PC.
2. **Generación del vector de muestras:** a partir del archivo de disco que contiene el contenido binario de la memoria, se genera por medio de un *script* un vector de muestras que representa la salida del *frontend* en un formato nativo del software que realizará el procesamiento.
3. **Procesamiento:** el vector de muestras obtenido en el paso anterior se procesa por medio de diferentes *scripts* que modelan los algoritmos de búsqueda y seguimiento. Todos los *scripts* se implementaron utilizando Octave.

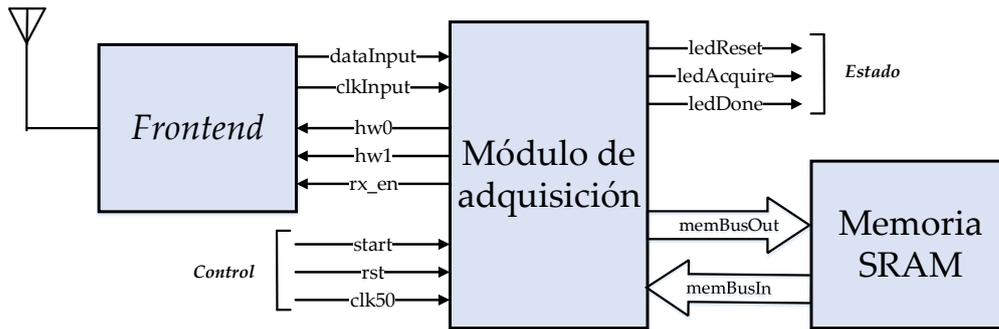


FIGURA 4.1: Esquema de conexión del *frontend* con el módulo de adquisición y la memoria SRAM

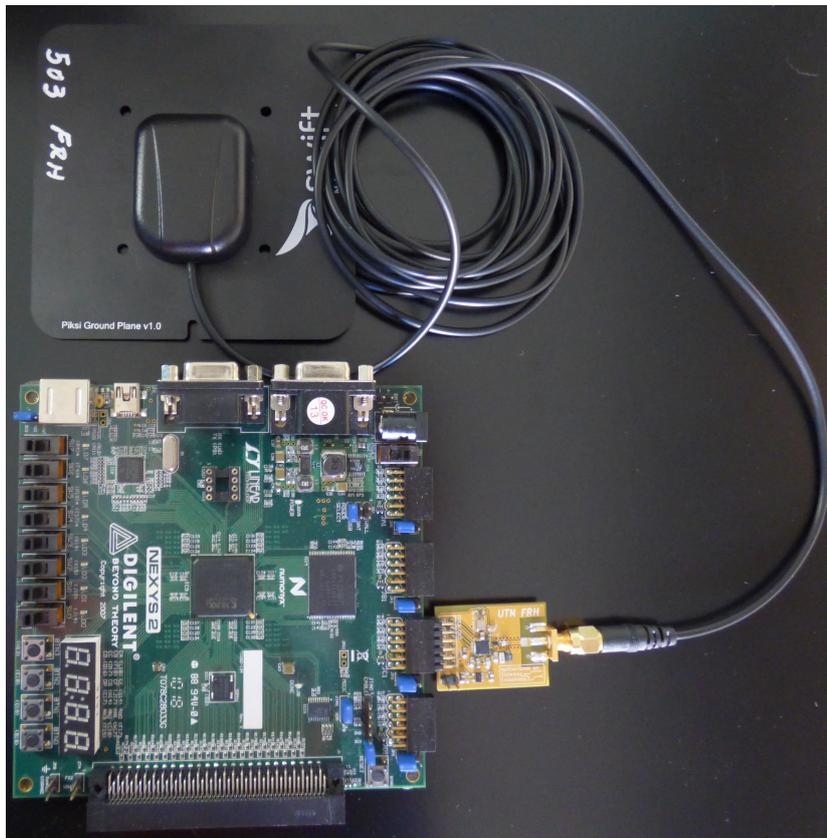


FIGURA 4.2: Fotografía del *frontend* con su antena activa conectado a la placa de desarrollo FPGA

4.1.1. Búsqueda de satélites en vista

En este ensayo se utiliza una función base denominada *Search* la cual recibe una señal GPS y devuelve la matriz correlación en el dominio $(n_{ol}, \Delta f_{dl})$ con una señal generada localmente (ver ecuación 2.25). Esta función se utilizó para graficar las matrices de correlación en el capítulo 2.

La interfaz de la función se define en Octave como:

```
[corrMatReplica]=SearchFFT(SenalGPS,nSat,fi,deltaDoppler
,stepDoppler,stepCA,blocal,length,samplesPerChip)
```

donde los distintos parámetros corresponden a:

- *SenalGPS*: Señal GPS a analizar
- *nSat*: Número de satélite con el cual se contrastará
- *fi*: Frecuencia intermedia [Hz]
- *deltaDoppler*: Radio del intervalo de desplazamiento Doppler [Hz] (el centro es cero)
- *stepDoppler*: Paso entre frecuencias de desplazamiento a analizar [Hz]
- *stepCA*: Paso de fase de código C/A a analizar [muestras]
- *blocal*: Cantidad de bits a utilizar para la generación de la señal réplica de búsqueda
- *length*: Longitud de la señal GPS (*SenalGPS*) a analizar
- *samplesPerChip*: Muestras por chip

Para hallar si un satélite está o no en vista analizando el vector de muestras provenientes del *frontend* resulta importante analizar las propiedades de las matrices de correlación en uno y otro caso. A modo de ejemplo, se muestran dos casos testigo. En la figura 4.3 se grafica una matriz de correlación resultante de aplicar el algoritmo de búsqueda sobre un satélite que no estaba a la vista. Se observa que no hay un máximo distintivo; esto acusa la situación de que el satélite no se encontró en el dominio de búsqueda. De forma contrapuesta, en la figura 4.4 se muestra una matriz de correlación para un satélite que sí estaba en vista, como puede apreciarse, la matriz posee un máximo distintivo, indicando la presencia del satélite. Las coordenadas del máximo en los ejes de búsqueda corresponden al desplazamiento de la frecuencia central y fase de código C/A instantánea del satélite al momento de aplicar el algoritmo.

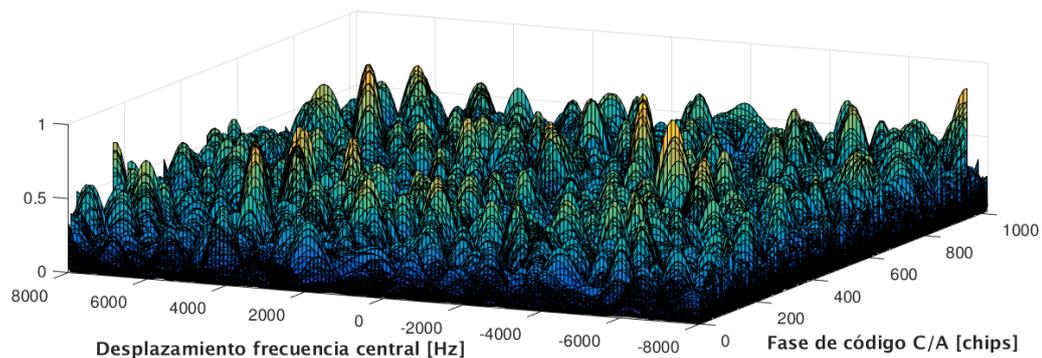


FIGURA 4.3: Matriz de correlación normalizada de una secuencia real (satélite no a la vista)

Para determinar qué satélites de la constelación están en vista y cuáles no, se debe repetir la operación de búsqueda para cada uno de ellos, considerando en qué casos se detectan los picos de correlación. Existen diferentes formas de hacer esta comprobación, pero la más sencilla de implementar consiste en relevar el máximo de cada matriz de correlación para cada uno de los satélites tal como se muestra en la figura 4.5.

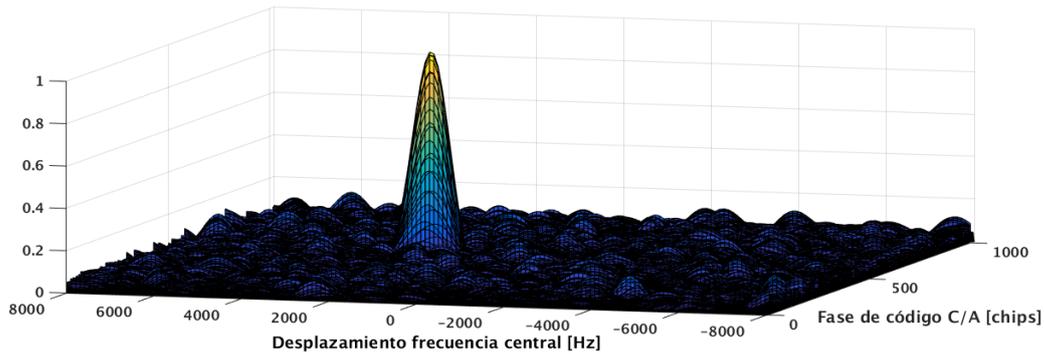


FIGURA 4.4: Matriz de correlación normalizada de una secuencia real (satélite a la vista)

Los satélites en vista, identificados como 5, 14, 15, 17, 19, 20, 28 y 30, se distinguen por su máximo de correlación de mayor valor. Alternativamente, se puede obtener un factor de mérito que resulta de dividir el máximo de correlación de la matriz respecto de su media. La media de la matriz de correlación tiende a adquirir un valor uniforme independientemente de que un satélite se encuentre presente o no[10]. En la figura 4.6 se grafica este factor de mérito para cada uno de los satélites. En ambos casos, es posible definir un umbral de detección a partir del cual se considera que un satélite se encuentra o no visible. El segundo método es más eficiente en entornos de peor relación señal ruido[10], no obstante, requiere de un mayor esfuerzo computacional. En este trabajo se utiliza el método de comparación del máximo de correlación respecto de un determinado umbral.

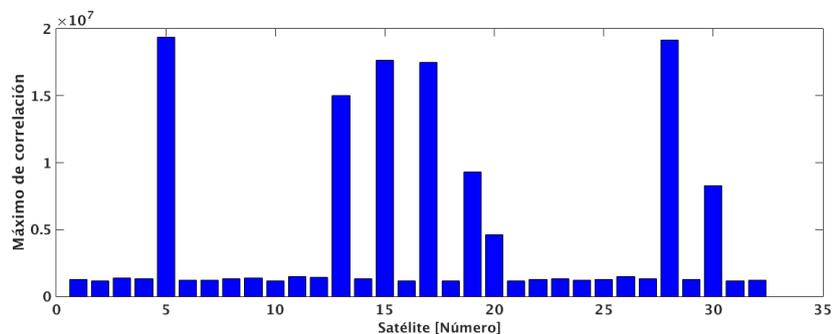


FIGURA 4.5: Máximos de correlación para la operación de búsqueda en función del número de satélite

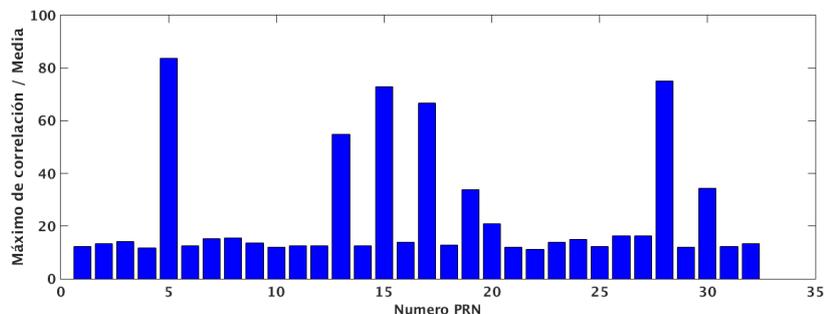


FIGURA 4.6: Máximos de correlación (divididos por la media de la matriz) para la operación de búsqueda en función del número de satélite

4.1.2. Evolución de frecuencia y fase de código C/A

En este ensayo se analizan períodos consecutivos de un vector de muestras a fin de estudiar como evoluciona la frecuencia central y la fase de código C/A en función del tiempo. Para ello, se divide el vector de muestras original en subvectores para cada uno de los cuáles se efectúa el proceso de búsqueda.

En la sección 4.1.1, se realizó la correlación en un período de código C/A ($1ms$) lo cual fue suficiente para determinar los satélites en vista. Para mejorar la inmunidad del método respecto del ruido es posible integrar varios períodos disminuyendo, de esta manera, el piso de ruido. En la figura 4.7 se grafica la matriz de correlación resultante de implementar el algoritmo de búsqueda con un período de integración de $5ms$. Se observa gráficamente la diferencia respecto de la figura 4.4 (tiempo de integración $1ms$) dado que en esta última los valores de correlación en la zona alrededor del pico disminuyen su valor relativo al máximo de correlación.

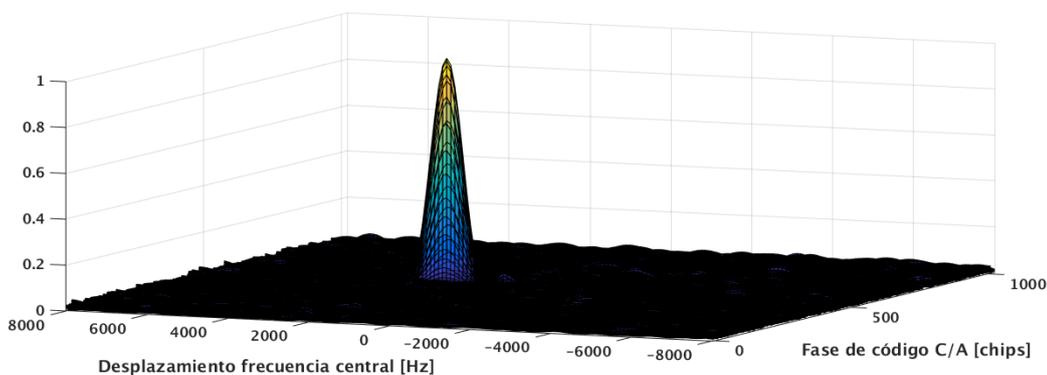


FIGURA 4.7: Matriz de correlación normalizada de una secuencia real (período de integración = $5ms$)

A modo de ejemplo, se estudió la evolución de la frecuencia y la fase de código C/A del satélite 5 en un intervalo de $500ms$, tomando como pasos de discretización en frecuencia $100Hz$ y en fase C/A de $\frac{1}{16}chip$. En la figura 4.8 y 4.9 se observa la variación de la frecuencia y la fase del código C/A respectivamente.

Según se desarrolló en la sección 2.1.3 la variación de la frecuencia en función del tiempo es lenta (menos de $1Hz/s$) por lo que aparece constante en el intervalo de estudio. Por otro lado, la variación de la fase del código C/A puede llegar a ser de algunos $chips/s$ [10] lo cual se aprecia como una rampa en la figura 4.9.

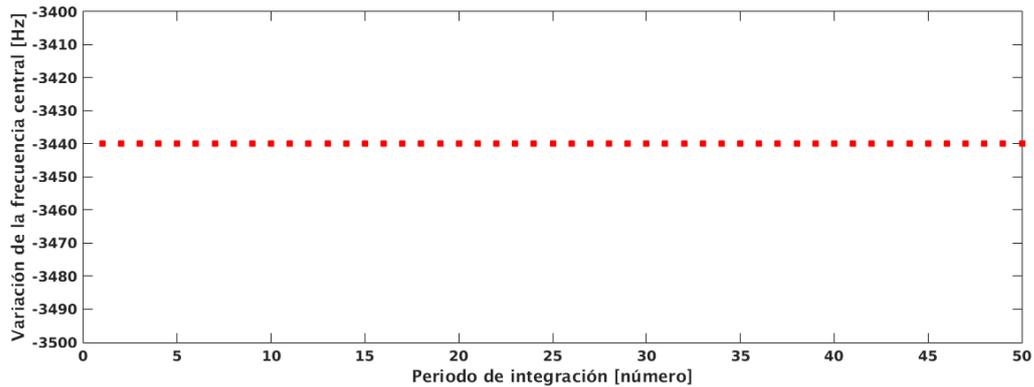


FIGURA 4.8: Evolución temporal del desplazamiento de la frecuencia central

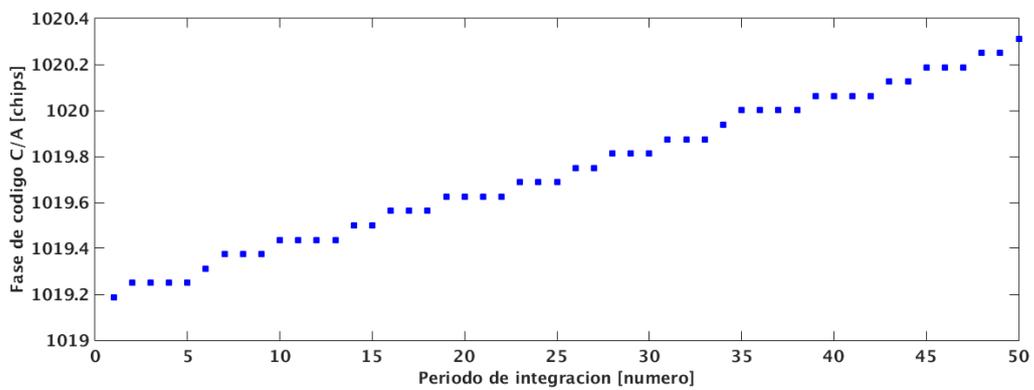


FIGURA 4.9: Evolución temporal de la fase del código C/A

4.2. Validación de módulo de búsqueda

En esta sección se describe el método desarrollado para validar el módulo de búsqueda descrito en detalle en la sección 3.10. Se instanció en la FPGA (Spartan 3E500) del kit de desarrollo Nexys 2 el periférico de búsqueda (sección 3.10.4) junto con el módulo de reproducción de señales (sección 3.4); cuyo esquema de conexión se muestra en la figura 4.10. Se puede observar que el módulo de reproducción de señales se conecta a la entrada del módulo de búsqueda, de esta forma, se puede reproducir a voluntad una secuencia temporal de muestras de una señal sintética o real según se desee. El módulo DCM genera el reloj de frecuencia $81,84MHz$ en base al reloj de la placa de desarrollo de $50MHz$.

Para controlar la estructura planteada a través de la interfaz SPI se utilizó una placa de desarrollo EDU-CIAA-NXP [34]. Esta placa posee un microcontrolador NXP LPC 4337, conformado por un núcleo Cortex M4F y un núcleo Cortex M0. El núcleo principal (M4F) cuenta con una unidad de punto flotante, extensiones de hardware para procesamiento de señales digitales, entre otros. Ambos núcleos operan de forma predeterminada a $204MHz$. La placa, según se muestra en la figura 4.11, dispone de dos puertos de expansión que permiten conectar los pines necesarios para intercomunicar el microcontrolador y la placa de desarrollo de la FPGA (Nexys 2).

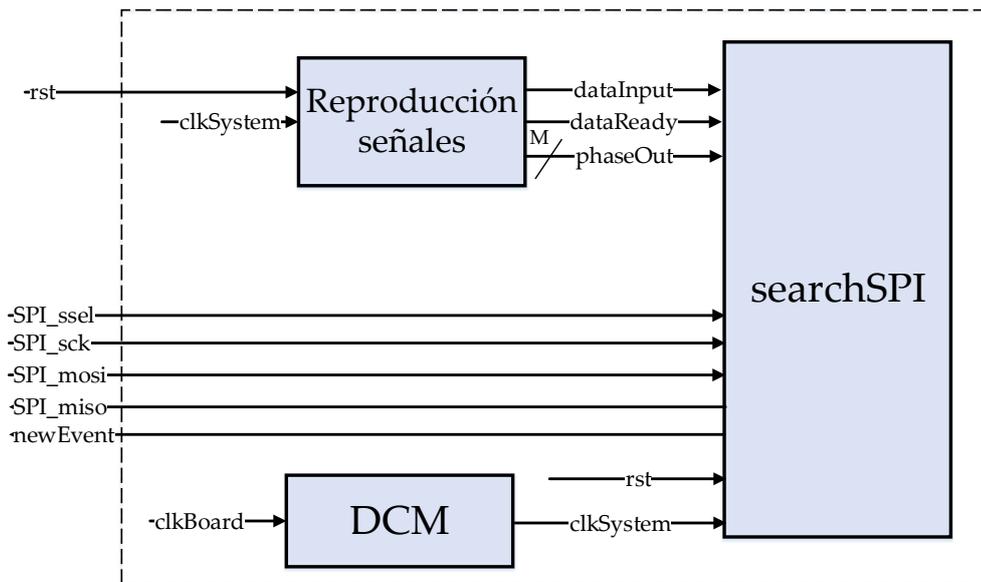


FIGURA 4.10: Conexión de módulos instanciados sobre la FPGA para validación del módulo de búsqueda



FIGURA 4.11: Placa de desarrollo EDU-CIAA-NXP

El esquema de conexión entre la placa EDU-CIAA-NXP y la placa Nexys 2 se muestra en la figura 4.12. Allí se observa:

- La conexión de la interfaz SPI del módulo de búsqueda instanciado en la Nexys 2 con la EDU-CIAA-NXP.
- La entrada de *reset*, que se conecta directamente a uno de los pulsadores de la placa Nexys 2.

Para controlar el periférico de búsqueda desde la placa EDU-CIAA-NXP se desarrolló una biblioteca de funciones en lenguaje C que permiten:

- Inicializar el módulo de búsqueda.
- Configurar la frecuencia central.
- Configurar la fase inicial de código C/A.
- Configurar el número de satélite.
- Iniciar una operación de correlación.

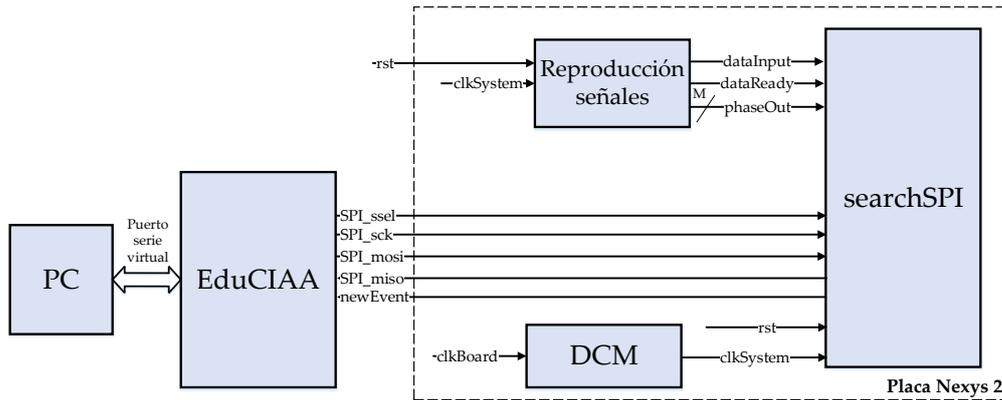


FIGURA 4.12: Esquema de conexión entre placa EDU-CIAA-NXP y placa Nexys 2

- Leer los valores de correlación (seno y coseno).

En base a la biblioteca desarrollada, se diseñó un ensayo utilizando el sistema operativo embebido FreeRTOS[35] que implementa:

- Inicialización del módulo de búsqueda.
- Configuración de la frecuencia central, fase inicial de código C/A y número de satélite.
- Inicio de la operación de correlación.
- Impresión de los valores de correlación (seno y coseno) a través de la interfaz serie de la placa de desarrollo.
- Repetición de los pasos 1 a 4 para diferentes valores de frecuencia central y fase de código C/A.

De esta manera, se obtienen los valores de correlación para diferentes valores de frecuencia central y fases de código C/A. Los resultados obtenidos se imprimen a una consola de texto a través de la interfaz serial de la EDU-CIAA-NXP y se capturan a través de un puerto serie virtual en la PC. Se eligió el formato CSV (del inglés, *Comma Separated Values*) para estructurar la salida de datos, de forma tal que puedan ser interpretados de forma directa por los *scripts* de procesamiento en Octave. Finalmente, se importan estos archivos de ensayo a través de un script de Octave que opera sobre los datos reportados y los presenta bajo la forma de las matrices de correlación que fueron presentadas ampliamente en este trabajo.

A modo de ejemplo, se presentan los resultados obtenidos al realizar la operación de búsqueda sobre una secuencia que contiene una señal GPS sintética ideal (sin ruido) originada en el satélite 1, con frecuencia central 2000Hz por sobre la frecuencia nominal de portadora y fase inicial de código CA igual a 512 chips.

En la figura 4.13 se observa la matriz de correlación normalizada obtenida al calcular los valores de correlación en el rango de frecuencias comprendido entre $[-8000, 8000]\text{Hz}$ en pasos de 500Hz y en el rango de fases de código C/A entre $[0, 1023]$ en pasos de 0,5 chip. Se puede notar el pico distintivo en torno al valor de frecuencia 2000Hz y fase C/A 512 chips.

En la figura 4.14 se muestra un detalle de la matriz para dos fases de código C/A fijas: una coincidente con la fase de la señal y otra no coincidente para facilitar su comparación. Se observa el máximo en torno a la frecuencia de 2000Hz para la fase coincidente.

En la figura 4.15 se muestra un detalle de la matriz para dos frecuencias fijas: una coincidente con la frecuencia de la señal y otra no coincidente para facilitar su comparación. Se observa el máximo en torno a la fase de código C/A de 512 chips para la frecuencia coincidente.

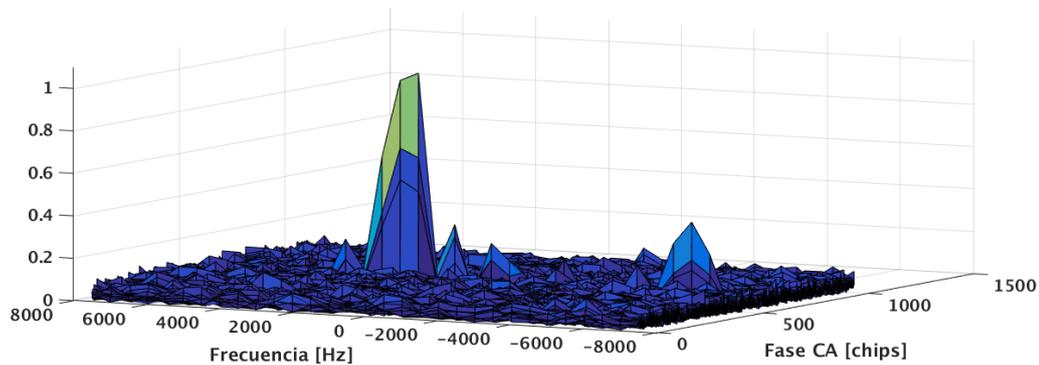


FIGURA 4.13: Matriz de correlación normalizada calculada por el periférico de búsqueda (señal sintética)

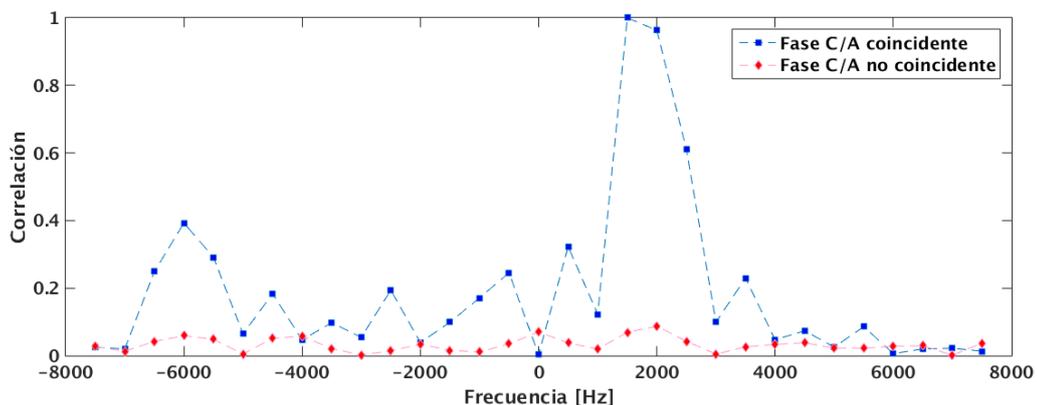


FIGURA 4.14: Valores de correlación para fase de código C/A fija y frecuencia variable (señal sintética)

Adicionalmente, se realizó la operación de búsqueda sobre una secuencia que contiene una señal GPS real. Esta secuencia es la misma que se analizó en la sección 4.1.1 y dió como resultado el hallazgo de los satélites número 5, 14, 15, 17, 19, 20, 28 y 30 (ver figuras 4.5 y 4.6). A modo de ejemplo, se presentan los resultados obtenidos al realizar la operación de búsqueda del satélite 14.

En la figura 4.16 se observa la matriz de correlación normalizada obtenida al calcular los valores de correlación en el rango de frecuencias comprendido entre $[-8000, 8000]\text{Hz}$ en pasos de 500Hz y en el rango de fases de código C/A entre $[0, 1023)$ en pasos de 0,5 chip. Se puede notar el pico distintivo en torno al valor de frecuencia 0Hz y fase C/A 777,5 chips.

En la figura 4.17 se muestra un detalle de la matriz para dos fases de código C/A fijas: una coincidente con la fase de la señal y otra no coincidente para facilitar su

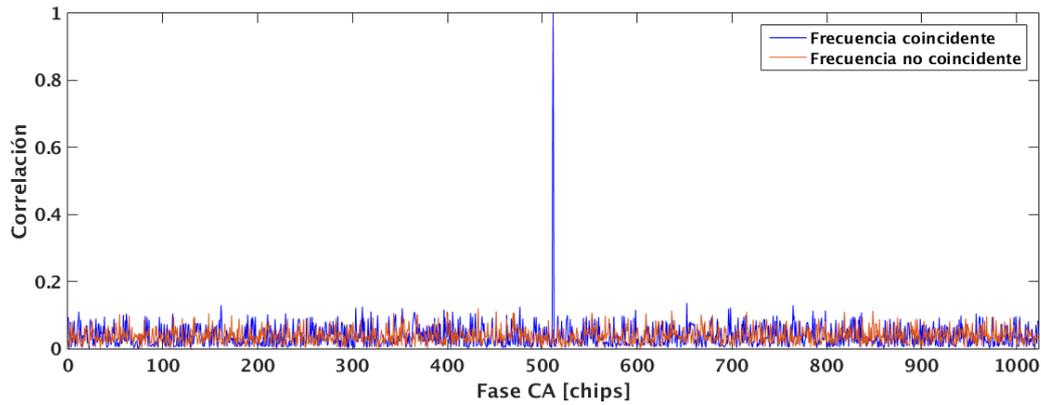


FIGURA 4.15: Valores de correlación para frecuencia fija y fase de código C/A variable (señal sintética)

comparación. Se observa el máximo en torno a la frecuencia de 0Hz para la fase coincidente.

En la figura 4.18 se muestra un detalle de la matriz para dos frecuencias fijas: una coincidente con la frecuencia de la señal y otra no coincidente para facilitar su comparación. Se observa el máximo en torno a la fase de código C/A de 777,5 chips para la frecuencia coincidente.

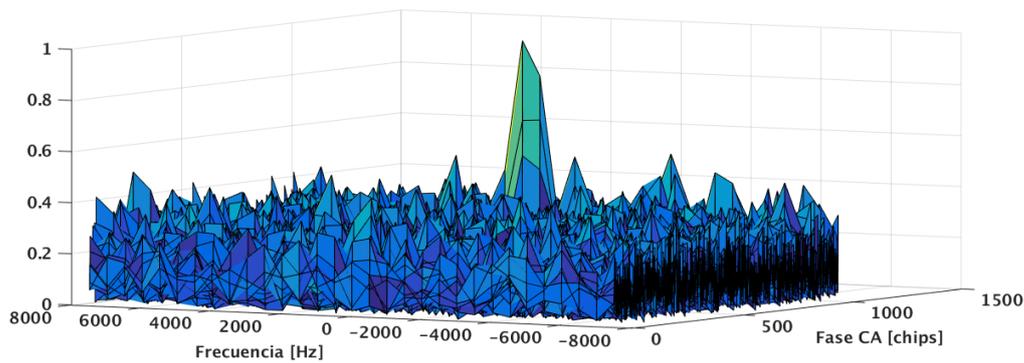


FIGURA 4.16: Matriz de correlación normalizada calculada por el periférico de búsqueda (señal real)

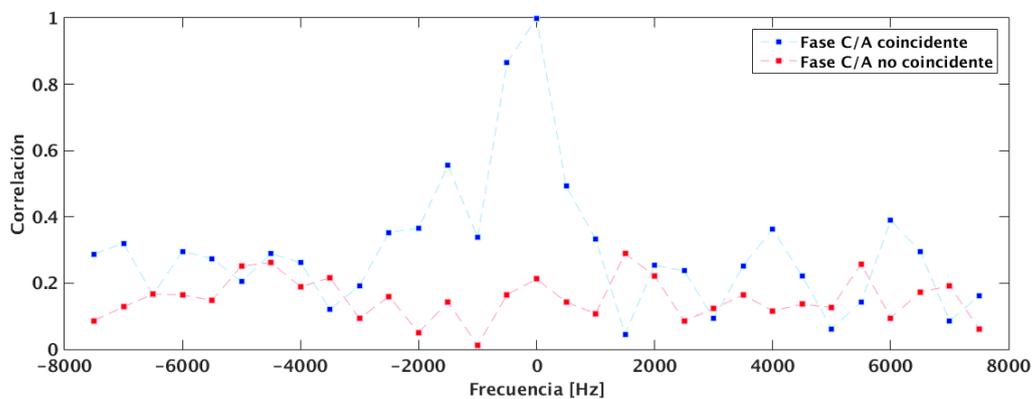


FIGURA 4.17: Valores de correlación para fase de código C/A fija y frecuencia variable (señal real)

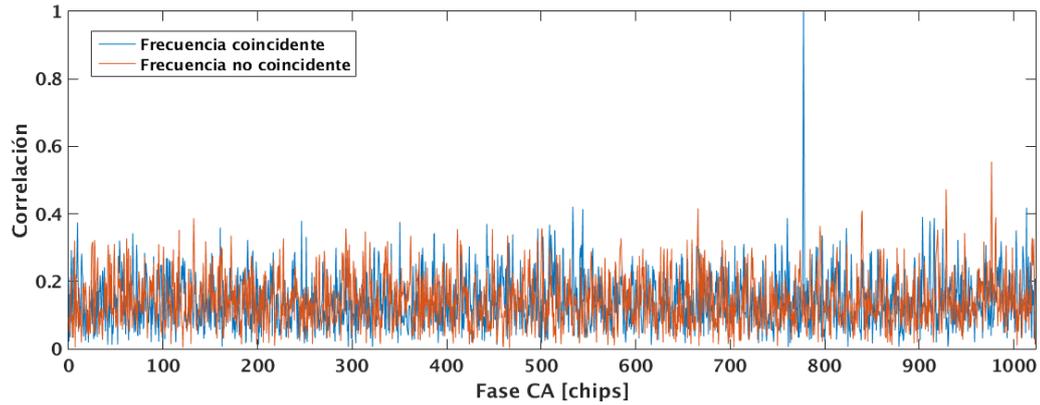


FIGURA 4.18: Valores de correlación para frecuencia fija y fase de código C/A variable (señal real)

4.3. Validación de algoritmo de seguimiento con señales reales

En esta sección se mostrarán los resultados de la validación de la estructura de seguimiento analizada en la sección 2.3 utilizando como entradas señales reales.

Por comodidad, se vuelve a reproducir la estructura general (figura 4.19), los diagramas de los lazos de control (figura 4.20) y del lazo de portadora (figura 4.21).

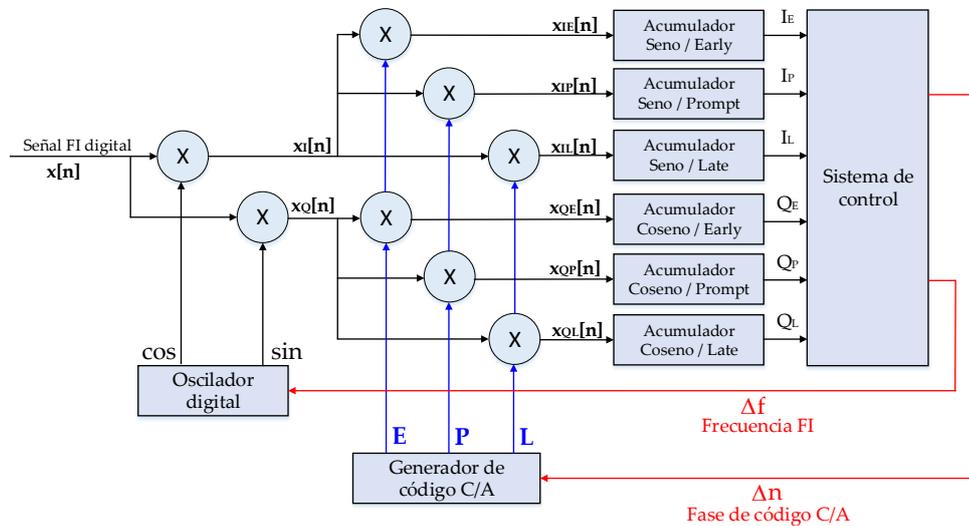


FIGURA 4.19: Estructura general del lazo de seguimiento

Se desarrolló un *script* en Octave que simula la estructura indicada en la figura 4.19 la cual fue alimentada con una serie de muestras tomadas del *frontend* de acuerdo al procedimiento descrito en la sección 4.1. Los valores iniciales para configurar los módulos del oscilador digital y el generador de código C/A se obtuvieron del algoritmo de búsqueda detallado en la sección 2.2. A fin de presentar un caso concreto se tomó el satélite 15 cuyos valores iniciales de desplazamiento de la frecuencia central es de 2200Hz y su valor inicial de fase de código C/A es de $926,125$ chips. Estos valores iniciales se cargaron en el oscilador digital y el

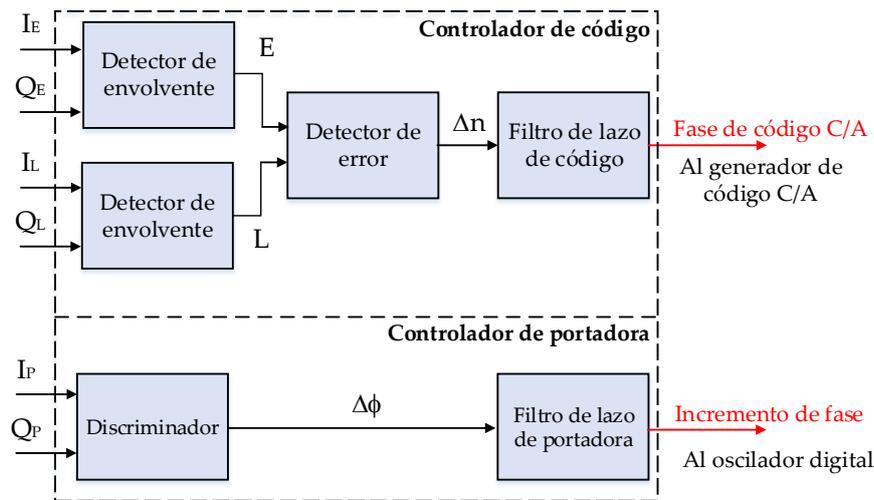


FIGURA 4.20: Estructura en bloques del controlador del lazo de seguimiento

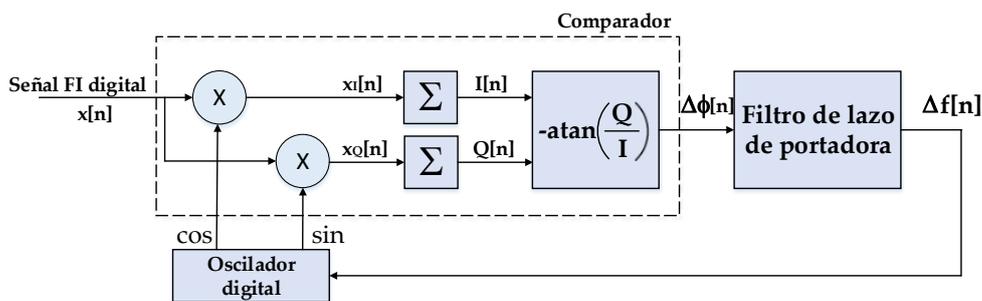


FIGURA 4.21: Estructura en bloques del lazo de portadora

generador de código C/A respectivamente para evaluar su comportamiento en función del tiempo, esto es, si el modelo propuesto es capaz de seguir tanto en frecuencia como en fase de código C/A una señal real.

El lazo de portadora se implementó utilizando el esquema de la figura 4.21 mientras que el lazo de código C/A se hizo siguiendo los lineamientos del diagrama de la figura 4.20 por medio de un lazo proporcional.

En primer lugar se analiza el resultado logrado por el lazo de código C/A, el cual pudo mantener adecuadamente la sincronía del código C/A según se muestra en la figura 4.22. En ella, se pueden apreciar los valores de las señales $E[n]$ (correlación con la réplica *Early*), $P[n]$ (correlación con la réplica *Prompt*) y $L[n]$ (correlación con la réplica *Late*). Según se ha planteado, el objetivo del lazo es maximizar el valor de $P[n]$, balanceando los valores de $E[n]$ y $L[n]$, lo cual se logra a pesar de la dispersión introducida por el ruido de la señal real siendo los resultados obtenidos comparables a los observados en modelos similares [10].

En segundo lugar, se analiza el comportamiento del lazo de portadora, el cual pudo mantener adecuadamente la sincronía de portadora lo que se evidencia en las figuras 4.23 y 4.24. En la figura 4.23 se grafica el valor del canal I que luego

del transitorio de «enganche» (en el período comprendido entre $0ms$ y $100ms$) muestra las transiciones propias del mensaje de navegación. Según se explicó en la sección 2.2 el mensaje de navegación digital se introduce en la portadora por medio de transiciones de fase de 180° resultando en la inversión del signo del canal I. Por otra parte, el canal en cuadratura Q (figura 4.24) mantiene un valor en torno a 0° como consecuencia del «enganche».

Finalmente, para facilitar la lectura de los bits del mensaje de navegación se indica en la figura 4.25 la secuencia del canal I normalizado, donde se ha graficado la envolvente que conforma el dato de navegación.

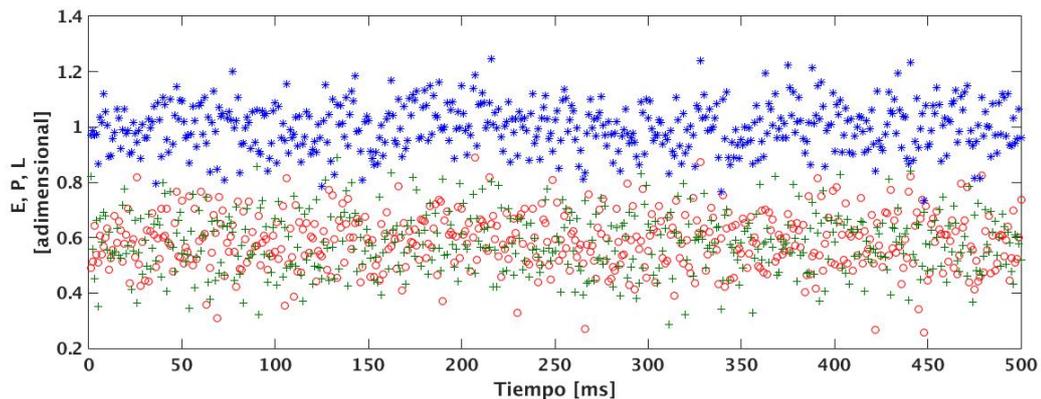


FIGURA 4.22: Valores de estimadores *Early*, *Prompt* y *Late* para algoritmo de seguimiento con señales reales

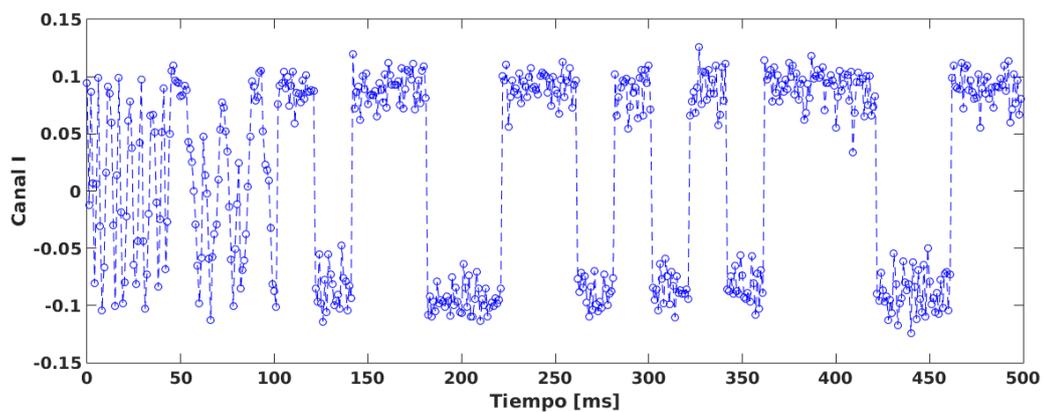


FIGURA 4.23: Valor del canal I (en fase) para algoritmo de seguimiento con señales reales

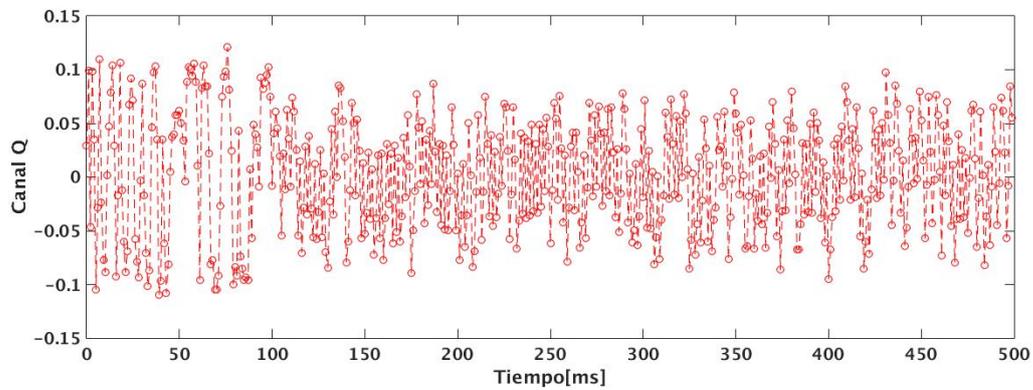


FIGURA 4.24: Valor del canal Q (en cuadratura) para algoritmo de seguimiento con señales reales

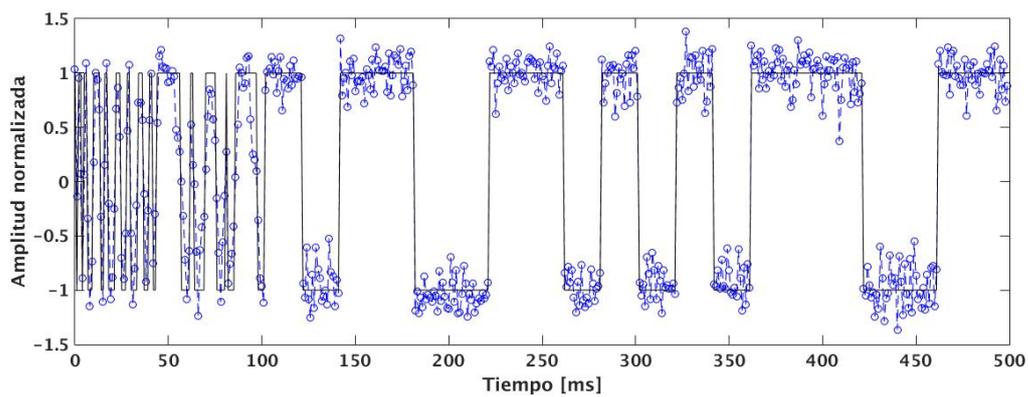


FIGURA 4.25: Valor del canal I (en fase) para algoritmo de seguimiento con señales reales (en negro, se resalta la envolvente de los bits del mensaje e navegación)

Capítulo 5

Conclusiones

5.1. Conclusiones

Se diseñaron, describieron y evaluaron módulos de búsqueda y seguimiento definidos en lógica programable (FPGA) para un receptor GPS.

Los receptores GPS comerciales sólo pueden utilizarse como periféricos, es decir, proveen una salida de datos que contiene la solución de navegación (latitud, longitud y altura) [36, 37] y, en menor cantidad de casos, efemérides y observables. Para la mayoría de las aplicaciones esto es suficiente, pero en algunos campos tales como: sistemas de navegación integrados para vehículos de alta dinámica, investigación sobre nuevas técnicas de posicionamiento, uso de receptores en entornos de baja relación señal ruido, entre otros, esto no es suficiente. Es deseable entonces contar con una arquitectura flexible que pueda adaptarse a cada situación y permita estudiar cambios en las estructuras internas del receptor tanto en la parte de señal mixta (*frontend*) como en la parte digital (módulos de búsqueda, seguimiento, procesamiento de efemérides y observables).

En ese sentido, este trabajo cumplió con el objetivo de diseñar, verificar y validar módulos definidos por lenguajes de descripción de hardware que pueden modificarse y adaptarse a diferentes condiciones. Aún más, se estudiaron exhaustivamente las señales del sistema GPS proveyendo una sólida base teórica y práctica para el desarrollo futuro en esta temática.

Asimismo se desarrollaron una serie de *scripts* que permiten la generación de señales GPS sintéticas (con y sin ruido) que se utilizaron para el ensayo de los módulos implementados. Los algoritmos de búsqueda y seguimiento se modelaron en software y se validaron utilizando señales sintéticas y reales. De esta forma, se estableció una base de conocimiento que permite el modelado de nuevas estructuras de búsqueda y seguimiento como paso previo y seguro para su descripción en hardware.

En el transcurso de este trabajo fue necesario generar herramientas de reproducción de señales digitales (3.4) para conducir diferentes pruebas sobre los módulos de hardware, tanto con señales sintéticas como reales. De esta manera, se consolidó un medio adecuado y flexible para las pruebas realizadas y para la continuación a futuro de la línea de trabajo.

Cabe destacar que el aprendizaje obtenido no se limita únicamente a la aplicación del sistema GPS. La modulación BPSK y las técnicas de DSSS/CDMA son de amplia utilización en los sistemas de comunicaciones satelitales[38, 39] por lo

que se podría adaptar las herramientas de hardware y software a otros sistemas satelitales.

Es importante destacar que el trabajo realizado sirvió de nexo para la aplicación e integración de las diferentes áreas de conocimiento que conforman el posgrado: diseño de placas de circuito impreso, análisis de señales y sistemas digitales (incluyendo su modelización por software e implementación en hardware), control digital, protocolos de comunicaciones, diseño y validación de lógica programable, sistemas operativos de tiempo real, entre otros. Además, se utilizaron herramientas específicas (GIT[40]) para versionar las diferentes etapas de diseño y validación de los módulos planteados, así como los *scripts* facilitando la trazabilidad del desarrollo.

5.2. Trabajo a futuro

Se plantea a futuro diseñar y construir una placa de circuito impreso que integre el *frontend*, la FPGA y el microcontrolador, así como todos los componentes necesarios para operar de forma integrada. Esta plataforma de hardware permitirá implementar un receptor integrado donde será posible instanciar diversas arquitecturas de módulos de búsqueda y seguimiento orientadas a diferentes escenarios (dinámica del sistema, relación señal ruido, entre otros). Adicionalmente, se integrarán las librerías de cálculo de solución de navegación, las cuales fueron desarrolladas y optimizadas[41, 42] por el grupo de investigación de «Aplicaciones en Sistemas Embebidos» de la UTN-FRH.

Bibliografía

- [1] *Federal Aviation Administration - GPS - Policy - Selective Availability*. URL: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/policy/availability/ (visitado 10-11-2016).
- [2] Department of Defense. *GPS Standard Positioning System Performance Standard*. 4.^a ed. 2008. URL: <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>.
- [3] *GPS - Control Segment*. URL: <http://www.gps.gov/systems/gps/control/> (visitado 10-11-2016).
- [4] R. Gold. *Optimal binary sequences for spread spectrum multiplexing*. IEEE Trans. on Information Theory vol. 13, pp. 619–621, 1967. ISBN: 0-7803-3777-8.
- [5] M. Insall. *Modular Arithmetic*. URL: <http://mathworld.wolfram.com/ModularArithmetic.html> (visitado 23-10-2017).
- [6] J. Bao-yen Tsui. *Fundamentals of global positioning system receivers : a software approach*. 2.^a ed. John Wiley & Sons, Inc., 2005. ISBN: 0-471-70647-7.
- [7] E. Weisstein. *Werner Formulas*. URL: <http://mathworld.wolfram.com/WernerFormulas.html> (visitado 10-11-2016).
- [8] S. H. Oppenheim A. , Willsky A. , Nawab. *Señales y Sistemas*. 2.^a ed. Prentice Hall, 1998. ISBN: 970-17-0116-X.
- [9] R. Lyons. *Quadrature Signals: Complex, but not complicated*. URL: <http://icrf.nl/LinkClick.aspx?fileticket=mt1qQJbpo7M\%3D&tabid=2544&mid=3732> (visitado 10-11-2016).
- [10] F. Johansson. *GPS Satellite Signal Acquisition and Tracking*. URL: <http://www.sm.luth.se/csee/courses/sms/019/1998/navstar/navstar.pdf> (visitado 10-11-2016).
- [11] Kaplan E. , Hegarty C. *Understanding GPS: principles and applications*. 2.^a ed. Artech House, Inc., 2006. ISBN: 1-58053-894-0.
- [12] A., Roncagliolo P. and G., García J. and H., Muravchik C. «Optimized Carrier Tracking Loop Design for Real-Time High-Dynamics GNSS Receivers». En: *International Journal of Navigation and Observation* 2012 (). Ed. por Hindawi Publishing Corporation. URL: <http://downloads.hindawi.com/journals/ijno/2012/651039.pdf>.
- [13] Faculty of Engineering and Applied Science, Memorial University of Newfoundland. *Unit 6 - Steady-state error*. URL: http://www.cs.mun.ca/av/old/teaching/cs/notes/steady_quad.pdf.
- [14] Maxim Integrated Inc. *Universal GPS Receiver*. Datasheet. URL: <https://datasheets.maximintegrated.com/en/ds/MAX2769.pdf>.
- [15] Skyworks Inc. *GPS Receiver IC Datasheet*. Datasheet. URL: <http://www.skyworksinc.com/uploads/documents/202445A.pdf>.

- [16] C. Hegarty. *Analytical Model for GNSS Receiver Implementation Losses*. The Mitre Corporation. URL: https://www.mitre.org/sites/default/files/pdf/09_3995.pdf.
- [17] Digilent Inc. *Spartan-3E FPGA Trainer Board*. Manual. URL: <https://reference.digilentinc.com/reference/programmable-logic/nexys-2/start>.
- [18] P. Chu. *FPGA Prototyping by VHDL Examples*. John Wiley & Sons, Inc., 2008. ISBN: 978-0-470-18531-5.
- [19] *Using Digital Clock Managers (DCMs) in Spartan 3 FPGAs*. Xilinx Inc. URL: <https://goo.gl/YJ6Bri>.
- [20] J.P. Charras. *KiCad EDA*. Página oficial. URL: <http://kicad-pcb.org/>.
- [21] *Digilent PMOD Interface Specification*. URL: <https://goo.gl/KsbcC3> (visitado 23-10-2017).
- [22] *Micron Async/Page/Burst CellularRAM™ Memory*. URL: http://www.stm32circle.com/resources/Datasheets/16mb_burst_cr1_0_p23z.pdf (visitado 23-10-2017).
- [23] *COE File Syntax*. URL: https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/cgn_r_coe_file_syntax.htm (visitado 23-10-2017).
- [24] Zeroplus Technology Co. Ltd. *Zeroplus LAP-C 16032 Logic Analyzer*. Manual. URL: <https://goo.gl/wqzYrz>.
- [25] *LogiCORE IP DDS Compiler v4.0*. Xilinx Inc. URL: https://www.xilinx.com/support/documentation/ip_documentation/dds_ds558.pdf.
- [26] Qingdao Hantek Ltd. *Hantek MSO5202D Digital Oscilloscope*. Manual. URL: <https://goo.gl/JuXbXj>.
- [27] Surf VHDL. *Write to File in VHDL using TextIO Library*. URL: <http://surf-vhdl.com/write-to-file-in-vhdl-using-textio-library/>.
- [28] F.S. Larosa. *Módulo de búsqueda, seguimiento y decorrelación para un sistema GPS sobre FPGA*. URL: <http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Trabajo-Final-CESE-Facundo-Larosa-2016.pdf>.
- [29] *Serial Peripheral Interface Bus*. Wikipedia. 2017. URL: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus.
- [30] Doin J. *SPI Master/Slave Core*. URL: https://opencores.org/project,spi_master_slave.
- [31] L. Couch. *Digital and Analog Communication Systems*. 8.ª ed. Pearson Education Ltd., 2013. ISBN: 0-273-77421-2.
- [32] J. P. Costas. *Synchronous Communications*. Proceedings of the IEEE , Volume 90 , pp. 1461-1466, 2002.
- [33] Digilent Inc. *Adept 2 Software*. Digilent Software. URL: <https://reference.digilentinc.com/reference/software/adept/start>.
- [34] Proyecto CIAA. *EduCIAA*. Página principal del proyecto EduCIAA. URL: <http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>.
- [35] *FreeRTOS*. Real Time Engineers Ltd. URL: <http://www.freertos.org/>.
- [36] *U-Blox Products*. U-Blox GMBH. URL: <https://www.u-blox.com/en/standard-precision-gnss-products>.
- [37] *Quectel Products*. URL: <http://www.quectel.com/product/list/GNSSIoTModule.htm>.
- [38] Monojit Mitra. *Satellite Communications*. Ed. por Prentice Hall. ISBN: 81-203-2786-1.
- [39] Bruce Elbert. *The Satellite Communication Application Handbook*. Ed. por Artech House. ISBN: 1-58053-490-2.

-
- [40] GIT. URL: <https://git-scm.com>.
- [41] F.S. Larosa y col. *Desarrollo y validación de algoritmos en C para cálculo de posición de usuario en el sistema GPS*. Ed. por Congreso Argentino de Ingeniería (CADI) 2016.
- [42] R. Ghignone y col. *Implementación y optimización de una librería embebida para receptor GPS*. Ed. por Congreso de Microelectrónica Aplicada (uEA) 2016.